

CA MICS[®] Resource Management Q&R Workstation

Query Library Guide

Release 12.9.00



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: Introduction 13

Purpose of this Guide	13
Audience	13
How to Use the CA MICS Query Library	14
Understanding Query Views	15

Chapter 2: CA MICS Analyzer Option for CICS (CIC) Queries 21

CICDAT – CICS Daily Top 10 Workloads	22
Overview	24
Uses	24
Data Extracts	25
Query Defaults	26
Modifications	26
CICDTR – CICS Daily Tran Count and Response Pct	27
Overview	28
Uses	28
Data Extracts	29
Query Defaults	29
Modifications	31
CICDTT – CICS Daily Top 10 Transactions	32
Overview	34
Uses	34
Data Extracts	35
Query Defaults	36
Modifications	36
CICMAT – CICS Monthly Top 10 Workloads	37
Overview	38
Data Extracts	39
Query Defaults	40
Modifications	40
CICMTR – CICS Monthly Tran Count and Response Pct	41
Overview	42
Uses	42
Data Extracts	43
Query Defaults	43
Modifications	45

CICMTT – CICS Monthly Top 10 Transactions	46
Overview	47
Data Extracts	48
Query Defaults	49
Modifications	49

Chapter 3: CA MICS Analyzer Option for DB2 (DB2) Queries 51

DB2BPH – DB2 Daily Buffer Pool Hit Percent	51
Overview	52
Uses	52
Data Extracts	52
Query Defaults	53
Modifications	53
DB2T10 – DB2 Daily Top Ten Plans by Thread Count	54
Overview	55
Uses	56
Data Extracts	56
Query Defaults	57
Modifications	57
Query Defaults	58
Modifications	58
DB2TCT – DB2 Daily Thread Count by Connection Type	59
Overview	59
Uses	60
Data Extracts	60
Query Defaults	60
Modifications	60

Chapter 4: CA MICS Analyzer Option for IDMS (IDM) Queries 61

IDMT10 - IDMS Daily Top Ten Programs Executed	61
Overview	63
Uses	65
Data Extracts	66
Query Defaults	68
Modifications	69
IDMTYP – IDMS Daily Transaction Type Analysis	70
Overview	71
Uses	73
Data Extracts	74
Query Defaults	78
Modifications	79

Chapter 5: CA MICS Analyzer Option for MQSeries (MQS) Queries 81

MQSBPA – MQ Daily Buffer Pool Analysis.....	82
Overview	83
Uses	84
Data Extracts	84
Query Defaults	85
Modifications	85
MQSCFS – MQ Daily Coupling Facility Structure Use	86
Overview	87
Uses	88
Data Extracts	88
Query Defaults	89
Modifications	89
MQSDB2 – MQ Daily DB2 Shared Queue Analysis	90
Overview	90
Uses	90
Data Extracts	91
Query Defaults	91
Modifications	92
MQSDMA – MQ Daily Data Manager Analysis	92
Overview	93
Uses	93
Data Extracts	93
Query Defaults	94
Modifications	94
MQSLMA – MQ Daily Log Manager Analysis	95
Overview	95
Uses	96
Data Extracts	96
Query Defaults	96
Modifications	97
MQSMMA – MQ Daily Message Manager Analysis	97
Overview	98
Uses	98
Data Extracts	98
Query Defaults	99
Modifications	99
MQST10 – MQ Daily Top 10 Queues	99
Overview	100
Uses	100
Data Extracts	100

Query Defaults	101
Modifications	101
MQSTSA – MQ Daily Task Suspend Call Analysis.....	102
Overview	103
Uses	103
Data Extracts	103
Query Defaults	104
Modifications	104

Chapter 6: CA MICS Hardware and SCP Analyzer (RMF) Queries 105

Mainframe CPC Configuration	105
RMFCFG – CPC LPAR Engine Configuration.....	106
RMFCFH – CPC Engine Configuration by Hour	110
RMFSTO – CPC LPAR Storage Allocation	114
RMFWGT – CPC LPAR Relative Weight by Engine Pool.....	116
Mainframe CPC Engine Utilization	120
RMFCPD – CPC and LPAR Daily Shared CP Engine Dispatch by Hour.....	121
RMFEHH – CPC Shared Engine Pools Percent Dispatch High Hours.....	127
RMFICD – CPC and LPAR Daily Shared ICF Engine Dispatch by Hour	133
RMFIFD – CPC and LPAR Daily Shared IFL Engine Dispatch by Hour	139
RMFZAD – CPC and LPAR Daily Shared zAAP Engine Dispatch by Hour	145
RMFZID – CPC and LPAR Daily Shared zIIP Engine Dispatch by Hour.....	151
Mainframe CPC Memory Use and Paging	156
RMFFRM – z/OS Central Storage Daily Frame Use by SYSID	157
RMFPAG – z/OS System Daily Paging Analysis	165
Mainframe CPC Specialty Engine Focus	170
RMFZAC – Daily zAAP Engine Use and Demand by CPC and SYSID.....	171
RMFZAS – Daily zAAP Engine Use and Demand by Service Class.....	176
RMFZIC – Daily zIIP Engine Use and Demand by CPC and SYSID.....	180
RMFZIS – Daily zIIP Engine Use and Demand by Service Class.....	185
Mainframe MSU Utilization and LPAR Capping.....	189
RMFMC4 – Daily z/OS LPAR MSU Use Cap and 4 Hour Avg.....	190
RMFMCL – Daily LPAR MSU Use and Capacity Limits	195
RMFMMC – Monthly LPAR MSU Use and Capacity Limits	200
RMFMSU – Daily CPC MSU Use with Stacked LPARs.....	204
WLM Service Class and Report Class Analysis	207
RMFRPT – z/OS Report Class Analysis by SYSPLEX, CPCID, and SYSID	208
RMFS10 – z/OS Daily Top 10 Service Classes by SYSPLEX	218
RMFSRV – z/OS Service Class Analysis by SYSPLEX, CPCID, and SYSID.....	222
Hardware Instrumentation Services CPU Measurement Facility	233
RMFCRL - CPU MF Cryptographic Coprocessor Activity LPAR Level.....	233

RMFCRT - CPU MF Cryptographic Coprocessor Activity CPUTYPE Level	240
RMFCRE - CPU MF Cryptographic Coprocessor Activity IPU Level	245
RMFWCL - CPU MF Workload Characterization LPAR Level	251
RMFWCT - CPU MF Workload Characterization CPUTYPE Level	273
RMFWCE - CPU MF Workload Characterization IPU Level	279

Chapter 7: CA MICS Batch and Operations Analyzer (SMF) Queries 285

Top 10 Resource Users	285
SMFAJT – SMF Daily Top 10 APPC Transaction Jobs	286
SMFBJT – SMF Daily Top 10 Batch Jobs	291
SMFBPT – SMF Daily Top 10 Batch Programs	302
SMFSAT – SMF Daily Top 10 System Address Spaces	313
SMFSJT – SMF Daily Top 10 Started Task Jobs	317
SMFTUT – SMF Daily Top 10 TSO Users	323
SMFUJT – SMF Daily Top 10 Unix SysSrv Jobs	329
Specialty Engine zIIP/zAAP Focus	333
SMFDZA – SMF Daily zAAP CPU Use and Demand	334
SMFDZI – SMF Daily zIIP CPU Use and Demand	339

Chapter 8: CA MICS Tape Analyzer Option (TAP) Queries 345

IBM TS77xx Virtual Tape Server Queries	345
TS7CDC – IBM TS77xx Daily Cluster Data Copy Activity	346
TS7HBA – IBM TS77xx Daily Host Channel I/O and HBA Compression Analysis	348
TS7VMA – IBM TS77xx Daily Virtual Tape Mount Activity	353
TS7VRA – IBM TS77xx Daily Virtual Recall Mount Analysis	356
StorageTek VSM Queries	358
STKCDA – STK VSM Daily VTV Mounts and Concurrent Drive Use	359
STKCTA – STK VSM Daily Channel Throughput Analysis	367
STKDBU – STK VSM Daily Disk Buffer Use	371
STKIOT – STK VSM Daily I/O Throughput	374
STKRMD – STK VSM Daily Recall Mount Time Distribution	378
STKVMA – STK VSM Daily Virtual Tape Mount Activity	381
STKVMD – STK VSM Daily VTV Mount Time Distribution	385

Chapter 9: CA MICS Analyzer Option for VM/CMS (VMC) Queries 389

VMCCHA – VM CPC Daily Physical Channel Utilization	390
Overview	393
Uses	394
Data Extracts	394
Query Defaults	396

Modifications	397
VMCT10 – VM System Daily Top 10 CPU Users.....	398
Overview	398
Uses	399
Data Extracts	399
Query Defaults	399
Modifications	400

Chapter 10: CA MICS WEB Analyzer Option (WEB) Queries 401

WEBDRA– Daily Web Server Request Analysis.....	402
Overview	403
Uses	408
Data Extracts	408
Query Defaults	414
Modifications	415
WEBMRA - 30 Day Web Server Request Activity	416
Overview	417
Uses	420
Data Extracts	420
Query Defaults	423
Modifications	424

Chapter 11: CA MICS Space Analyzer Option (VCA) Queries 425

VCACIS– Top 20 CI Splits by DS Type	425
Overview	425
Uses	426
Data Extracts	427
Query Defaults	428
Modifications	428
VCAT2C– VCA Daily top 20 STORCLAS Space Used	429
Overview	429
Uses	429
Data Extracts	430
Query Defaults	431
Modifications	431
VCAVOL – Volume Capacity and Use.....	432
Overview	432
Uses	432
Data Extracts	433
Query Defaults	434
Modifications	434

Chapter 12: CA MICS Analyzer Option for IMS (IMS) Queries 435

IMSDAT – IMS Daily Top 10 Workloads.....435

- Overview436
- Uses436
- Data Extracts437
- Query Defaults437
- Modifications438

IMSDTR – IMS Daily Tran Count and Response Pct.....438

IMSDTT – IMS Daily Top 10 Transactions442

- Overview443
- Uses443
- Data Extracts444
- Query Defaults444
- Modifications445

Chapter 1: Introduction

This section contains the following topics:

[Purpose of this Guide](#) (see page 13)

[Audience](#) (see page 13)

[How to Use the CA MICS Query Library](#) (see page 14)

[Understanding Query Views](#) (see page 15)

Purpose of this Guide

This guide illustrates and explains query library samples distributed with CA MICS Q&R Workstation. These sample queries provide "out-of-the-box" value for CA MICS customers. They generate useful graphics showing resource utilization, workload activity, processor configuration, and other metrics that are important to monitor at modern data centers. This guide explains the charts generated by each sample query and provides information on how to customize and extend the queries to meet individual customer requirements.

Audience

This guide is for systems analysts and administrators who want to use CA MICS Q&R to probe the CA MICS database and develop reports that are useful for problem analysis and long-term monitoring of data center resource utilization, workload performance, and capacity planning.

How to Use the CA MICS Query Library

When you find a documented query that interests you, locate the sample query *.QRQ file in the CA MICS Query Library directory. These can be found by clicking on the CA MICS QUERY LIBRARY in the FILE menu option. The directory structure and query names match the table of contents of this guide. Copy the query to your own directory, update the FILE SELECTION steps to point to the appropriate units and files in your CA MICS complex, and submit the query. View the output when complete, and customize your local copy *.QRQ file to meet your needs.

If you find a query that produces a chart similar to what you like, but the sample query is written for another component or file, you can build a similar query by following these steps:

- Open two instances of Q&R on your PC.
- Open the existing library query in one instance of Q&R.
- Select "Compose a New Query" in the other Q&R instance.
- Going back and forth between the two Q&R instances, build the new query by selecting the same steps and tasks used in the original query.

Understanding Query Views

When a Q&R query includes one or more data extract steps, each data extract step generates a Comma Separated Value (CSV) file. The CSV file is sent to the distributed server, and is available for charting. The charts are generated at the granularity specified by the data extract key selections.

For any given chart, there is a limit to the amount of information that can be usefully displayed. The Query Views feature helps to maximize the information value that is associated with a data extract CSV. Using multiple views, you can see different charts for each chart instance that meets the summarization granularity for the data extract CSV.

For example, if the data extract summarization keys result in a chart for each unique LPAR (SYSID), each chart in the initial chart view might display the amount of CPU time that the LPAR uses. A second chart view could show the LPARs memory use, and a third might show the I/O service units that the LPAR consumes.

Creating multiple views for a single data extract provides a powerful data analysis tool, ideal for exploiting the metric rich files in the CA MICS database. Many CA MICS files contain hundreds of data elements. By adding a large number of data elements to a data extract, using the data extract Element Selection task, a series of chart views can be designed that display information to help understand the root cause of resource usage spikes and other anomalous behavior.

As an example of the value that is provided by multiple query views, consider the RMF component WLM Service Class query:

RMFSRV – z/OS Service Class Analysis by SYSPLEX, CPCID, and SYSID

This query generates three output CSVs. Each output CSV has four Q&R views defined that let you analyze different aspects of each WLM Service Class Period participating in a SYSPLEX. The primary purpose of this query is to allow you to see how well your Service Classes are performing against their goal objectives. This information is provided with the initial chart view. If you find a Service Class Period (SCP) where the WLM Performance Index (PI) value indicates that the goal was not achieved, you can examine one or more of the additional query views to help understand the reasons behind it. These additional query views provide summary information about the sampled states for the SCP, and a breakdown of the various execution delays experienced by workloads belonging to the SCP. Understanding the types of delays that impeded the workloads can help you understand the reason why the goal was not met, and provide information that you can use to resolve the problem.

The RMFSRV query includes three data extract steps that generates the following output CSVs:

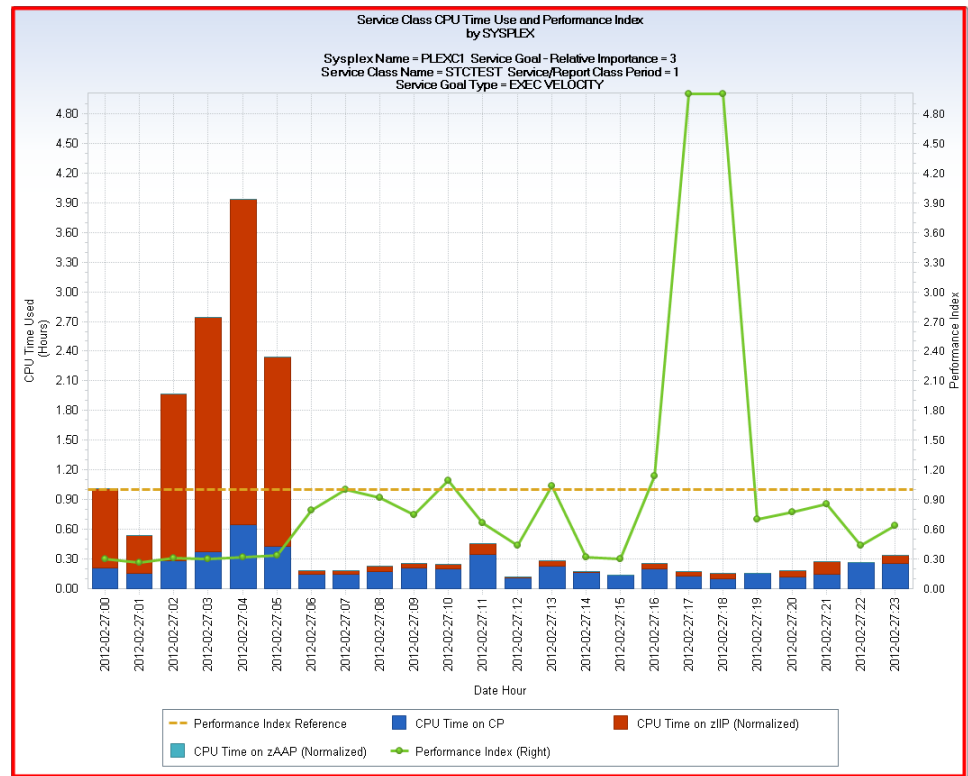
- Service Class by SYSPLEX
- Service Class by SYSPLEX CPCUID
- Service Class by SYSPLEX CPCUID SYSID

To show how multiple chart views support sophisticated problem analysis, we will examine the charts views for the data extract. Service Class by SYSPLEX. From the CSV file associated with this data extract, Q&R renders charts for each SCP in each SYSPLEX. At the SYSPLEX level, the SCP performance information is aggregated across all z/OS LPARs and Central Processing Complexes (CPCs) participating in each SYSPLEX. The following chart views are provided for the data extract:

1. WLMCLASS by SYSPLEX – CPU and PI
2. WLMCLASS by SYSPLEX – Sampling Summary
3. WLMCLASS by SYSPLEX – Delay States Group 1
4. WLMCLASS by SYSPLEX – Delay States Group 1

Note: The other two data extracts, Service Class by SYSPLEX CPCUID and Service Class by SYSPLEX CPCUID SYSID, provide similar chart views.

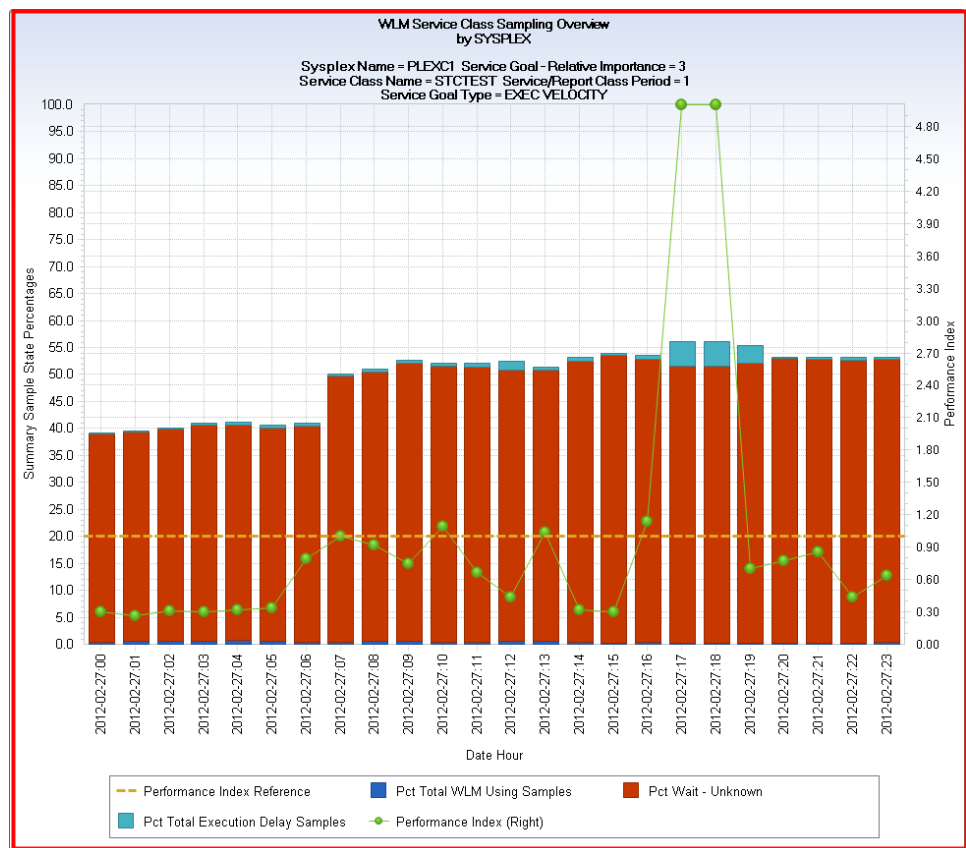
When multiple chart views are provided for a Q&R query, the first view is automatically displayed by Q&R when the data extract is charted. Following is a chart sample for the query RMFSRV View #1: WLMCLASS by SYSPLEX – CPU and PI.



Here you see the HOURLY chart for Period 1 of Service Class STCTEST, for February 27th 2012. The SYSPLEX name is PLEXC1. The SCP has a relative importance of 3 and an EXECUTION VELOCITY goal type.

As you can see, workloads running in this SCP execute on a mixture of CP and zIIP processors. The zIIP processors were heavily used in the early morning hours, but from about 6 AM until midnight, the CPU usage occurred primarily on CP processors—at the rate of about 0.20 CPU hours per hour. The Performance Index spikes well above a value of one between 4 PM and 7 PM. This means that the SCP goal was not met during those hours. To determine why the goal was not met, you can examine the other views that are provided by the query, using the VIEW drop-down button on the Q&R charting window.

Selecting View #2, WLMCLASS by SYSPLEX – Sampling Summary, results in the following new chart display for the same SCP:

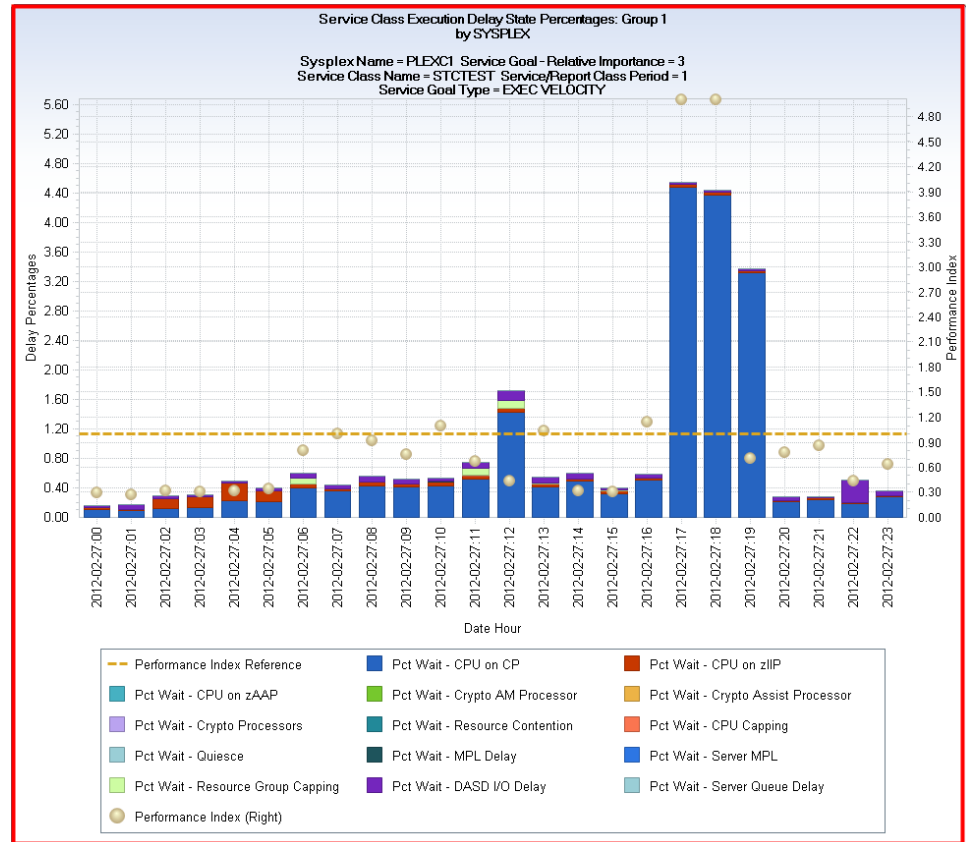


IBM’s Workload Manager (WLM) samples SCP workloads to determine their execution status. This Sampling Summary view displays the percentage of the samples that were found to be in one of three states: Using, Wait-Unknown, and Execution Delays. A fourth state not shown, which if added to the chart would bring the stacked bars up to the 100 percent line, is the percentage of samples where workloads in the SCP were idle.

The dark blue bar segments at the bottom of each stacked bar represent the percentage of samples where the workloads were using the CPU and, if I/O Priority is active, performing I/O. The red bar segments represent the percentage of workload delay states that WLM does not manage. These delays include time waiting on an I/O, DB2 latches, tape mount, enqueues, and others. The cyan colored bar segments represent the percentage of samples where the SCP workloads experienced execution delays for resources that WLM does manage. You can see that for the hours between 4 PM and 7 PM, where the SCP goal was not met (PI > 1), the percentage of Execution Delays increased.

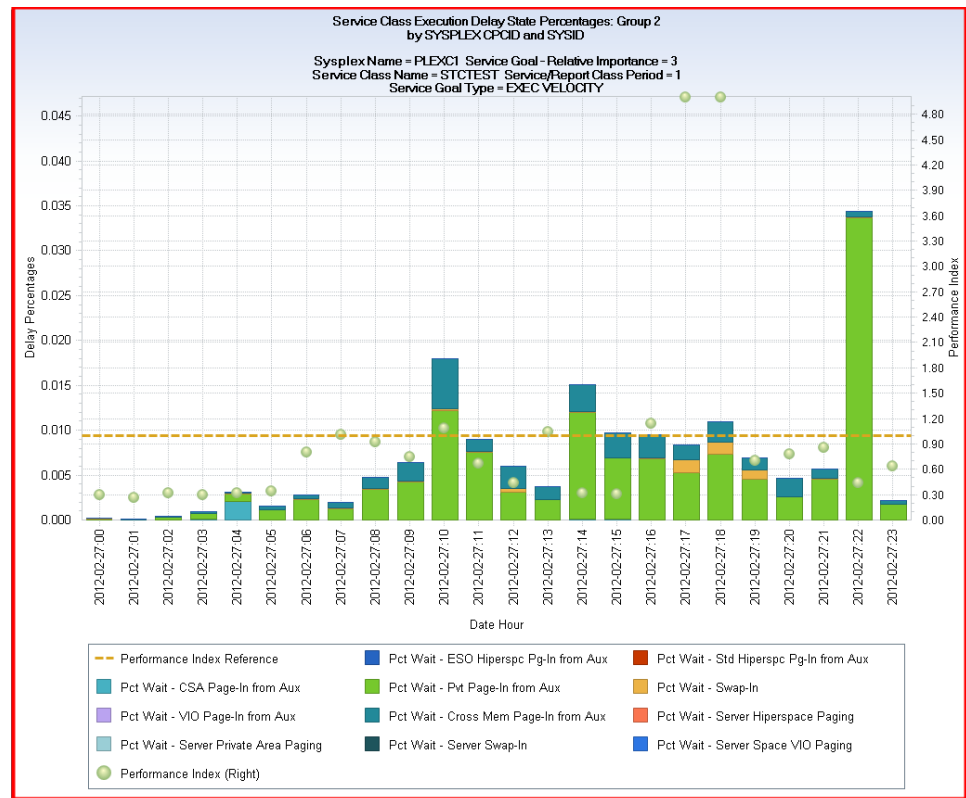
To determine the types of WLM managed execution delays experienced by the SCP workloads, you can examine views #3 and #4, Exec Delay States Group 1, and Exec Delay States Group 2, respectively. Two separate chart views are provided because there are almost 30 specific types of execution delays—too many to display effectively in a single chart.

Selecting View #3, Exec Delay States Group 1, results in the following chart display:



As you can see, there is a large spike in the Pct Wait – CPU on CP execution delay type for the hours between 4 PM and 7 PM when the execution velocity goal was missed. You now know the reason why this SCP failed to meet its execution velocity goal objective during those hours. About four percent of the time when the SCP workloads were ready to execute, they were waiting to be dispatched on CP processors—most likely because workloads belonging to higher priority Service Classes were competing for the same CP processors.

Selecting View #4, Exec Delay States Group 2, results in the following chart that shows the percentages of samples encountered for each of the remaining WLM managed execution delay wait states:



Make sure to examine the Left Y axis scale for these last two views, where the scale adjusts dynamically to match the values encountered in the charted data. The Left Y axis in the Group 1 view above ranges from about 0-5%, but only ranges 0-0.045% (less than 1/10th of 1 percent) for the 4th view.

With this example, you can see how the Q&R View feature can be used to create multiple chart views from a single data extract, allowing you to display different metrics from the same CSV file, simplifying sophisticated problem detection and analysis.

Chapter 2: CA MICS Analyzer Option for CICS (CIC) Queries

This section contains the following topics:

[CICDAT – CICS Daily Top 10 Workloads](#) (see page 22)

[CICDTR – CICS Daily Tran Count and Response Pct](#) (see page 27)

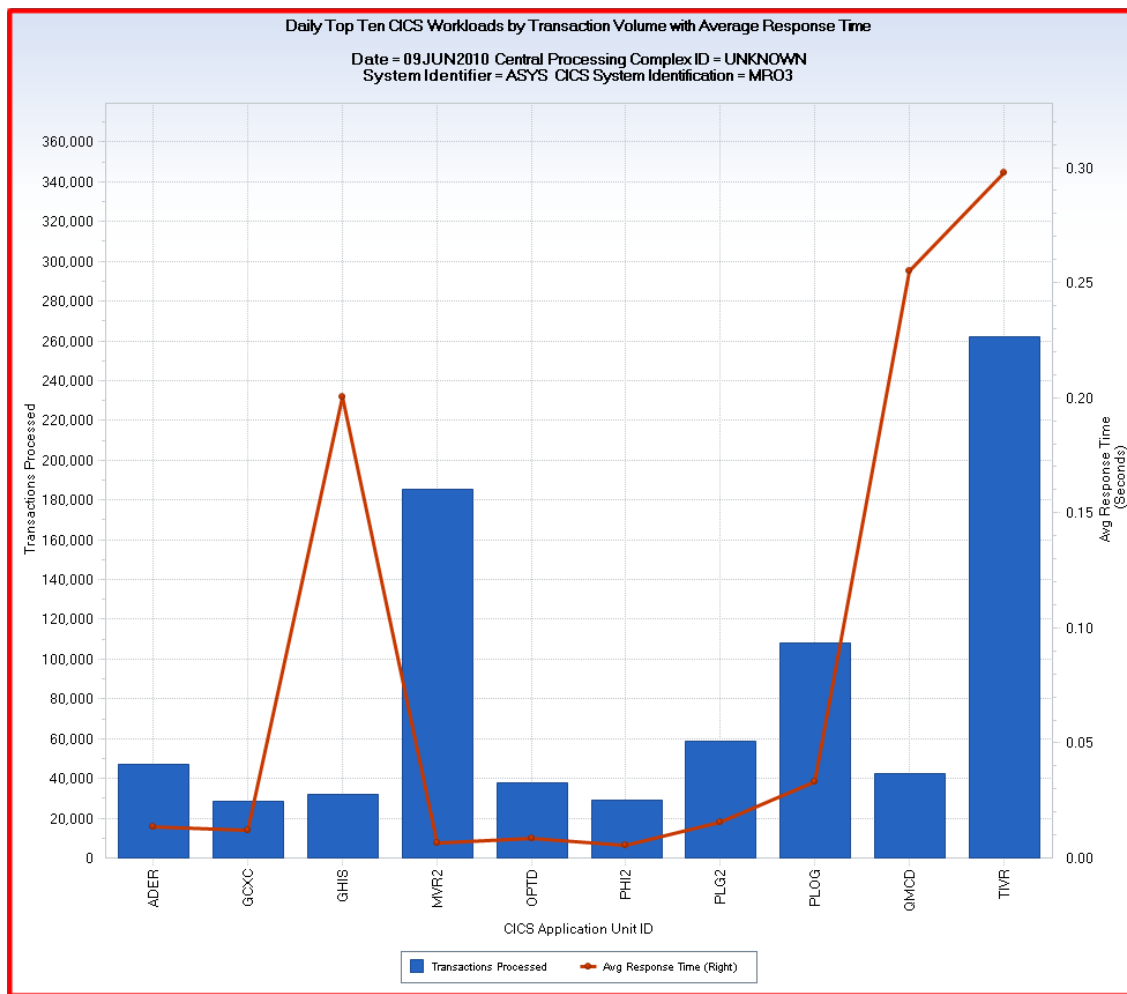
[CICDTT – CICS Daily Top 10 Transactions](#) (see page 32)

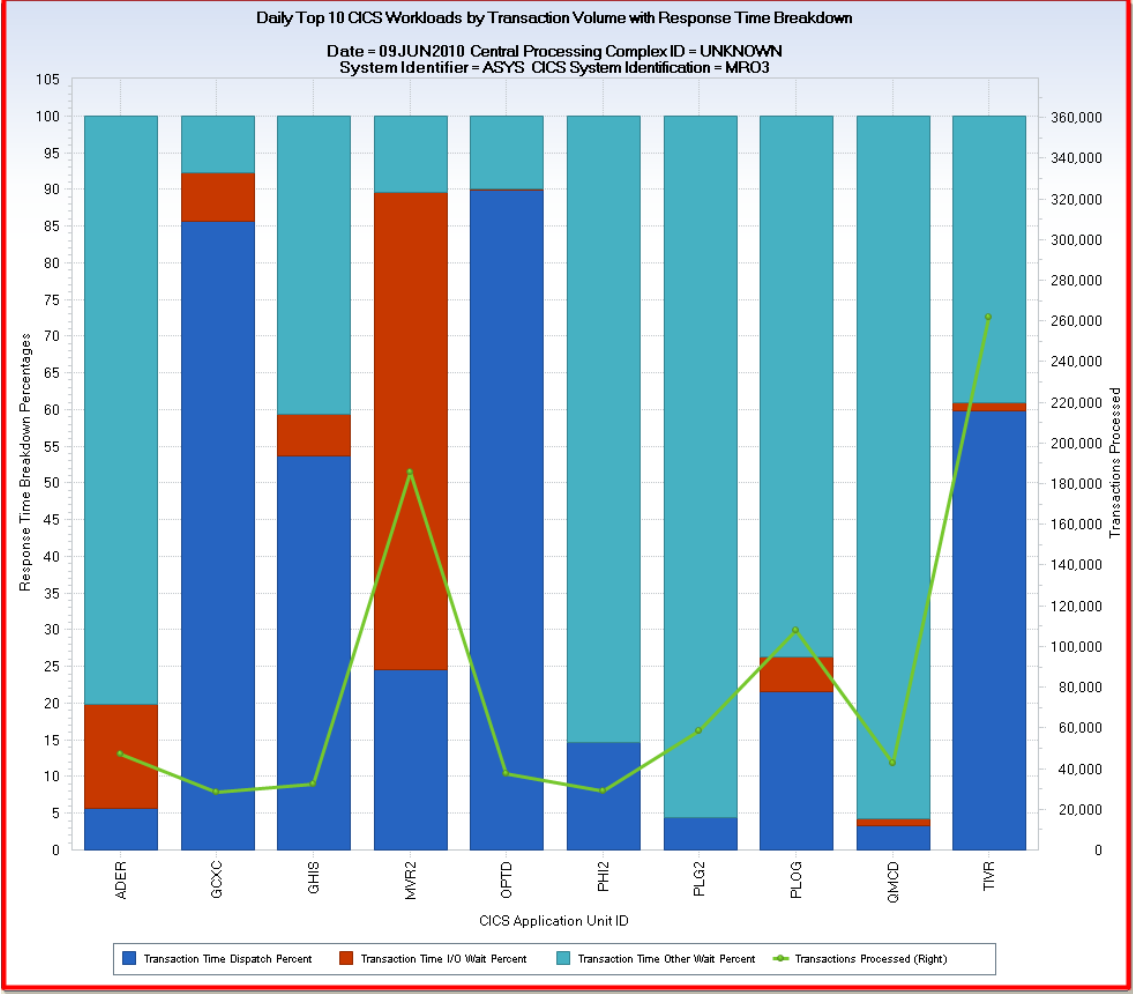
[CICMAT – CICS Monthly Top 10 Workloads](#) (see page 37)

[CICMTR – CICS Monthly Tran Count and Response Pct](#) (see page 41)

[CICMTT – CICS Monthly Top 10 Transactions](#) (see page 46)

CICDAT – CICS Daily Top 10 Workloads





Overview

The CICDAT CICS Daily Top 10 Workload Analysis by Region query generates two charts for each SYSID and CICS Region combination. The query uses the DAYS timespan CICS Application Unit Activity file as input, and reports on yesterday's top ten user-defined CICS Application Unit IDs (CICAPU). The CICAPU data element provides a means to categorize CICS transactions as belonging to specific groups or workloads. The CICCAU file uses CICAPU as a summarization key. The CICDAT query selects the top *nn* (default top 10) CICAPUs for each CICS region. The top ten are selected based on the highest number of CICS transactions. The top ten CICAPUs are graphically charted using two separate output views.

Note: The CICAPU routine, used in the CA MICS unit used to generate these charts, assigned TRANCODE to CICAPU. The resulting charts are actually showing the top ten transactions rather than the top ten workloads.

The first view shows the overall transaction count for the top ten CICAPUs using vertical bars, and the average CICS transaction response time for each CICAPU with a horizontal line.

The second view shows a response time breakdown for the same top ten workloads, using a stacked bar composed of the percentage of response time spent in dispatch, I/O wait, and other wait time. The total transaction count for each CICAPU is displayed with a horizontal line.

The examples shows the charts generated for a CICS region "MR03" executing on system "ASYS." Merging with data from the HARCPU file was not able to determine the Central Processing Complex (CPC) ID, hence the CPC ID is 'UNKNOWN.' The first chart shows that the top two transactions, by transaction count, were "TIVR" with 261,793 transactions and "MVR2" with 185,575 transactions. The average transaction response time was 0.30 seconds for workload "TIVR" and 0.01 seconds for workload "MVR2."

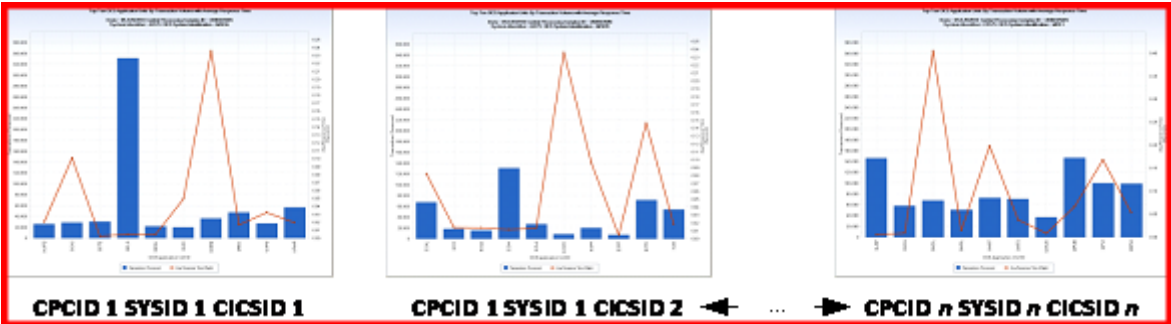
The second chart shows that workload "MVR2" spent 65 percent of the time waiting on I/O, and workload "TIVR" spent 39 percent of the time waiting for non-I/O related delays.

Uses

This query is useful to for analyzing workload-related transaction performance characteristics. By selecting the top ten workloads, based on transaction volume, you can focus your analysis on the workloads consuming the most CICS resources. Where transactions show significant percentages of time waiting on I/O or other delaying factors, further analysis may direct tuning efforts to improve overall response time.

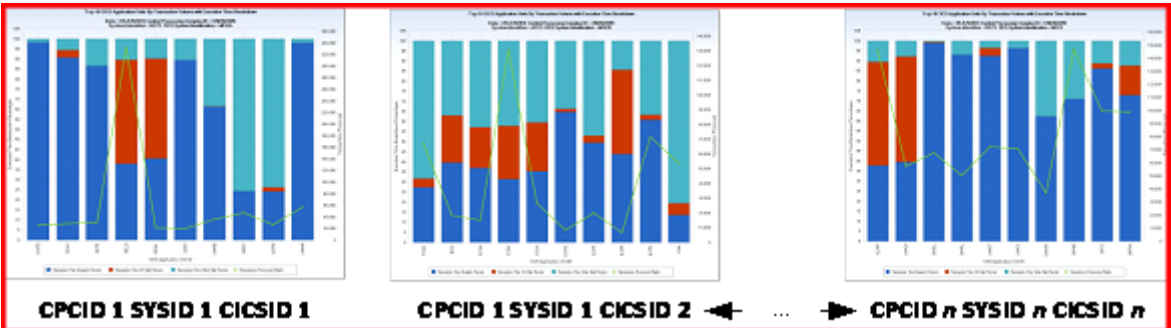
Data Extracts

CICAPU Daily Top 10 Response Time Avg



Left Y-axis
CAUTRANS - Transactions Processed
Right Y-axis
CAUAVTTM - Avg Response Time
X-axis
CICAPU – CICS Application Unit ID

CICAPU Daily Top 10 Resp Time Breakdown



Left Y-axis
DISP_PCT – Transaction Time Dispatch Percent
IOW_PCT – Transaction Time I/O Wait Percent
OTHW_PCT – Transaction Time Other Wait Percent
Right Y-axis
CAUTRANS – Transactions Processed
X-axis
CICAPU – CICS Application Unit ID

Query Defaults

- CICCAU DAYS timespan file: Cycle 01
- Derivations:
 - DATE
 - DATEPART(ENDTS)
 - CPCID
 - CPCID = TRIM(CPUMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5) ;
 - DISP_PCT
 - DISP_PCT = 100* (CAUPETTM / SUM(CAUPETTM,CAUIOWAT,CAUOTWAT)) ;
 - IOW_PCT
 - IOW_PCT = 100* (CAUIOWAT / SUM(CAUPETTM,CAUIOWAT,CAUOTWAT)) ;
 - OTHW_PCT
 - OTHW_PCT = 100* (CAUOTWAT / SUM(CAUPETTM,CAUIOWAT,CAUOTWAT)) ;

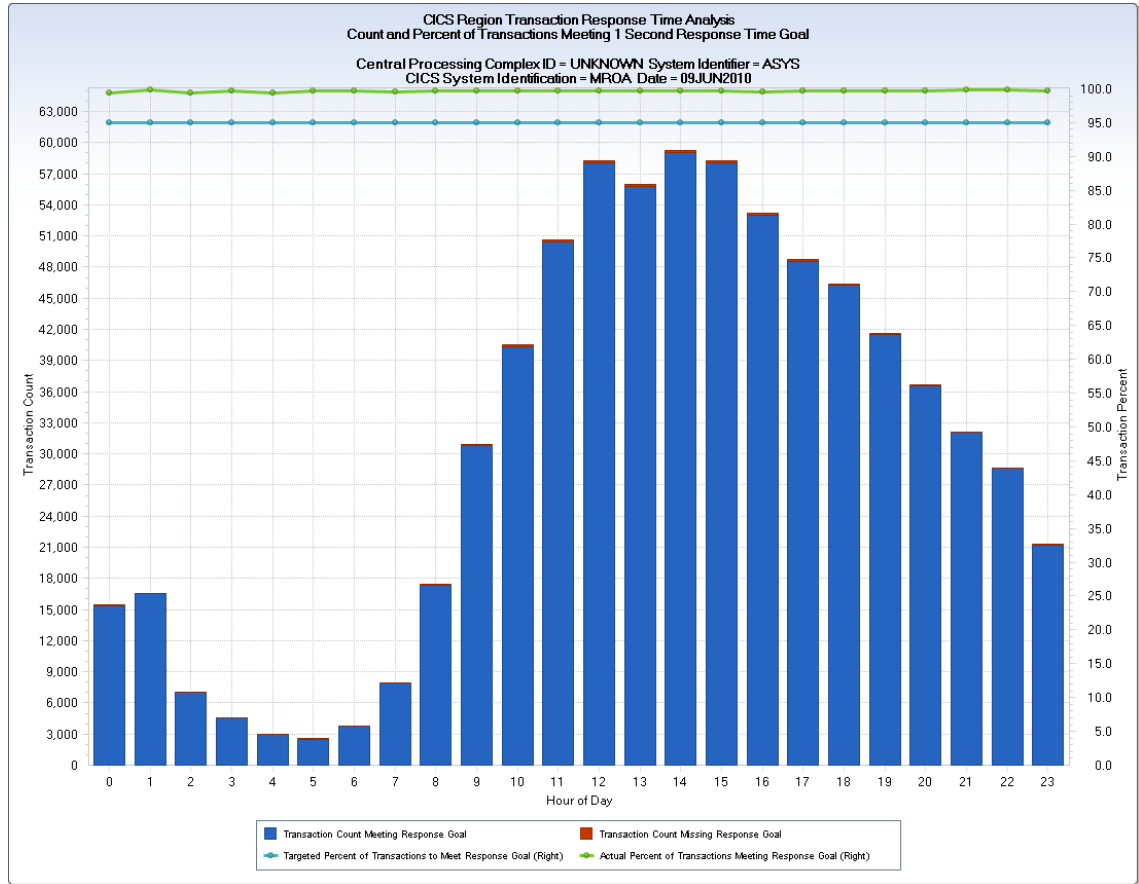
Modifications

The following modifications can be made to the query:

- You can adjust the TOP nn value to something other than the default of 10. You can add filter statements to select only a single SYSID or CICSID, or select only specific production CICS regions.
- You can increase the cycles input to create charts for multiple days, but comment out the user code in the CICAPU file input step that restricts processing to yesterday's data.
- Apply a format to the CICAPU variable to convert it from the 12 character workload identifier to a more meaningful workload description.

Note: If you change the TOP nn value (default TOP 10), modify the query chart title to reflect the changed value.

CICDTR – CICS Daily Tran Count and Response Pct



Overview

The CICDTR query generates daily charts showing overall transaction counts and response time metrics for individual CICS regions. A separate chart is produced for each z/OS SYSID and CICS region combination. The charts show the total transaction count for each hour using a stacked vertical bar—where the overall transaction count is broken down into two bar segments. One segment shows the transaction count that met a user-specified elapsed time goal using a blue bar (default one second), and the other segment shows the transaction count that missed the goal using a red bar.

In addition, the chart uses a horizontal line to show the percentage of transactions, per hour, that met the response time goal (green line). A user-specified desired percentage (default 95 percent) is plotted (blue line) to allow for easy visualization of the actual percentage of transactions, compared to the desired percentage of transactions, that met the response time goal.

The example shows the chart generated for a CICS region "MROA" running on a z/OS system "ASYS". This chart was produced from a CA MICS database built from CICS transaction records where we had not collected the associated RMF data. The Central Processing Complex ID (CPCID) was not available and set to "UNKNOWN." At a normal customer installation, one of the query steps examines the CA MICS RMF component HARCPU file to determine the CPCID for each z/OS system (SYSID) where the CICS regions executed. The CPCID is assigned a value like "2097-712-Serial AE000'.

The chart shows that transaction volume begins picking up at around 8:00 AM, peaks between noon and 3:00 PM, and then begins tapering off over the rest of the day. Close to 100 percent of the transactions, each hour, met the response time goal of completing in one second or less.

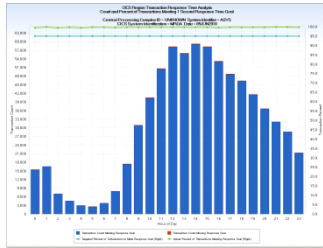
The second line of the title shows the response time target—and must be manually updated in the query output view definition if the response time target is changed from the default value of one (1) second.

Uses

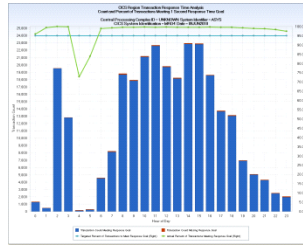
This query allows you to visualize easily the overall CICS transaction performance at the CICS region level. If you have established specific Service Level Agreements (SLAs) for CICS regions where *nn* percent of CICS transactions must complete in *mm* seconds or less, query parameters can be modified so that these charts provide fast visual assessment of actual performance compared to required or expected performance.

Data Extracts

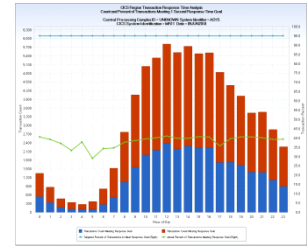
CICS Daily Transaction Response Goal Report



CPCID 1 SYSID 1 CICSID 1



CPCID 1 SYSID 1 CICSID 2 ← ... →



CPCID n SYSID n CICSID n

Left Y-axis

MEET_CNT: Transaction Count Meeting Response Goal

MISS_CNT: Transaction Count Missing Response Goal

Right Y-axis

GOAL_PCT: Targeted Percent of Transactions to Meet Response Goal

MEETL_PCT: Actual Percent of Transactions Meeting Response Goal

X-axis

Hour of Day

Query Defaults

- CICCSY DAYS timespan file: CYCLE 01
- HARCPU DAYS timespan file: CYCLE 01 (Used to determine CPCID<->SYSID mapping)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- User code defines response goal targets:
 - Response time goal: One (1) second – MACRO VAR &RESPVAL
 - Pct of transactions to meet Response Time Goal: 95% - MACRO VAR GOALPCT

■ Derivations:

DATE

DATEPART(ENDTS)

CPCID

```
CPCID = TRIM(CPUMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' ||  
SUBSTR(CPCSEQNB,12,5) ;
```

MEET_CNT

```
if cicrval7 <= &RESPVAL then do;  
  MEET_CNT=sum(CSYTDST7,CSYTDST6,CSYTDST5,CSYTDST4,  
    CSYTDST3,CSYTDST2,CSYTDST1);  
end;  
else if cicrval6 <= &RESPVAL then do;  
  MEET_CNT=sum(CSYTDST6,CSYTDST5,CSYTDST4,CSYTDST3,  
    CSYTDST2,CSYTDST1);  
end;  
else if cicrval5 <= &RESPVAL then do;  
  MEET_CNT = sum(CSYTDST5,CSYTDST4,CSYTDST3,CSYTDST2,CSYTDST1);  
end;  
else if cicrval4 <= &RESPVAL then do;  
  MEET_CNT = sum(CSYTDST4,CSYTDST3,CSYTDST2,CSYTDST1);  
end;  
else if cicrval3 <= &RESPVAL then do;  
  MEET_CNT = sum(CSYTDST3,CSYTDST2,CSYTDST1);  
end;  
else if cicrval2 <= &RESPVAL then do;  
  MEET_CNT = sum(CSYTDST2,CSYTDST1);  
end;  
else if cicrval1 <= &RESPVAL then MEET_CNT = CSYTDST1;
```

MISS_CNT

```
= SUM(CSYTRANS, -MEET_CNT) ;
```

GOAL_PCT

```
= &GOALPCT ;
```

MEET_PCT

```
= (MEET_CNT / CSYTRANS) * 100;
```

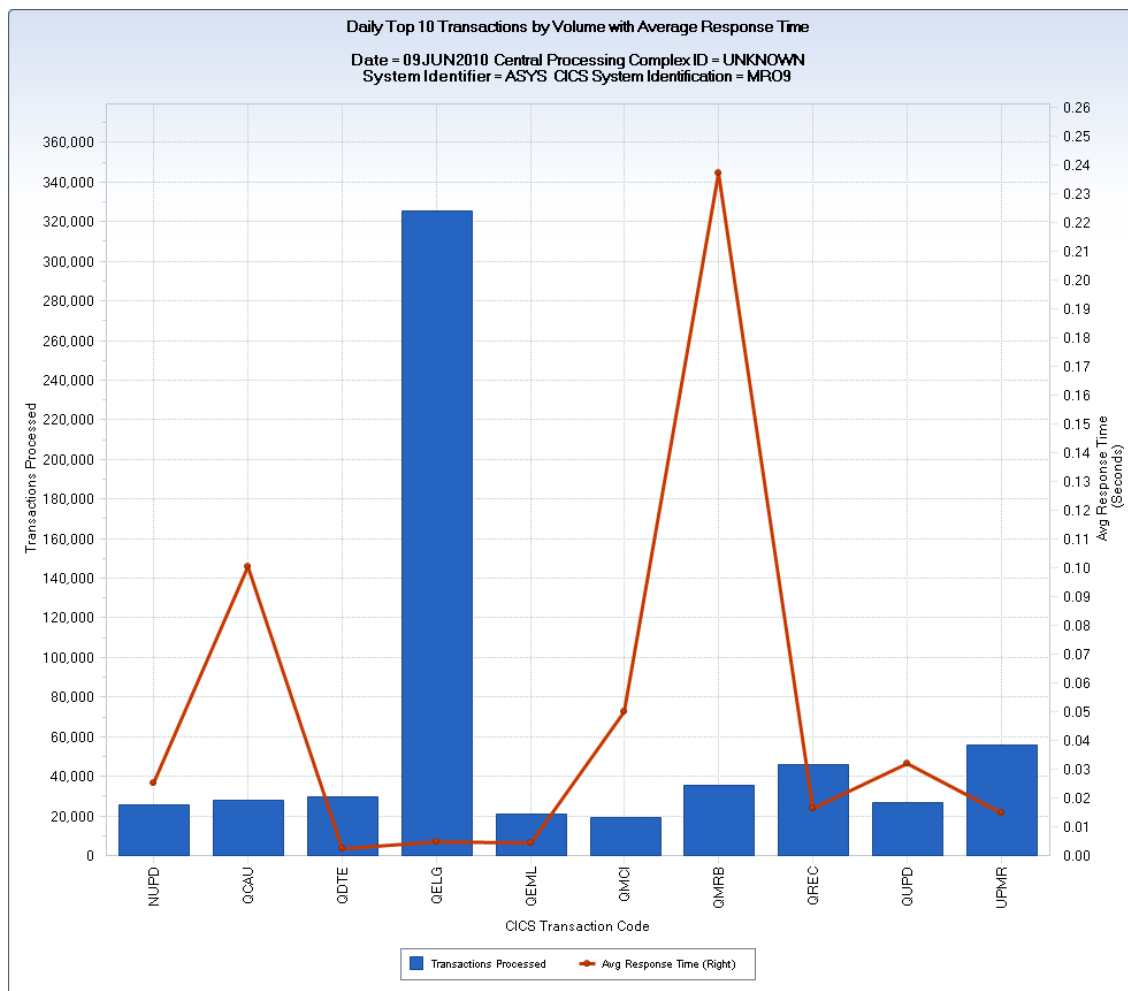
Modifications

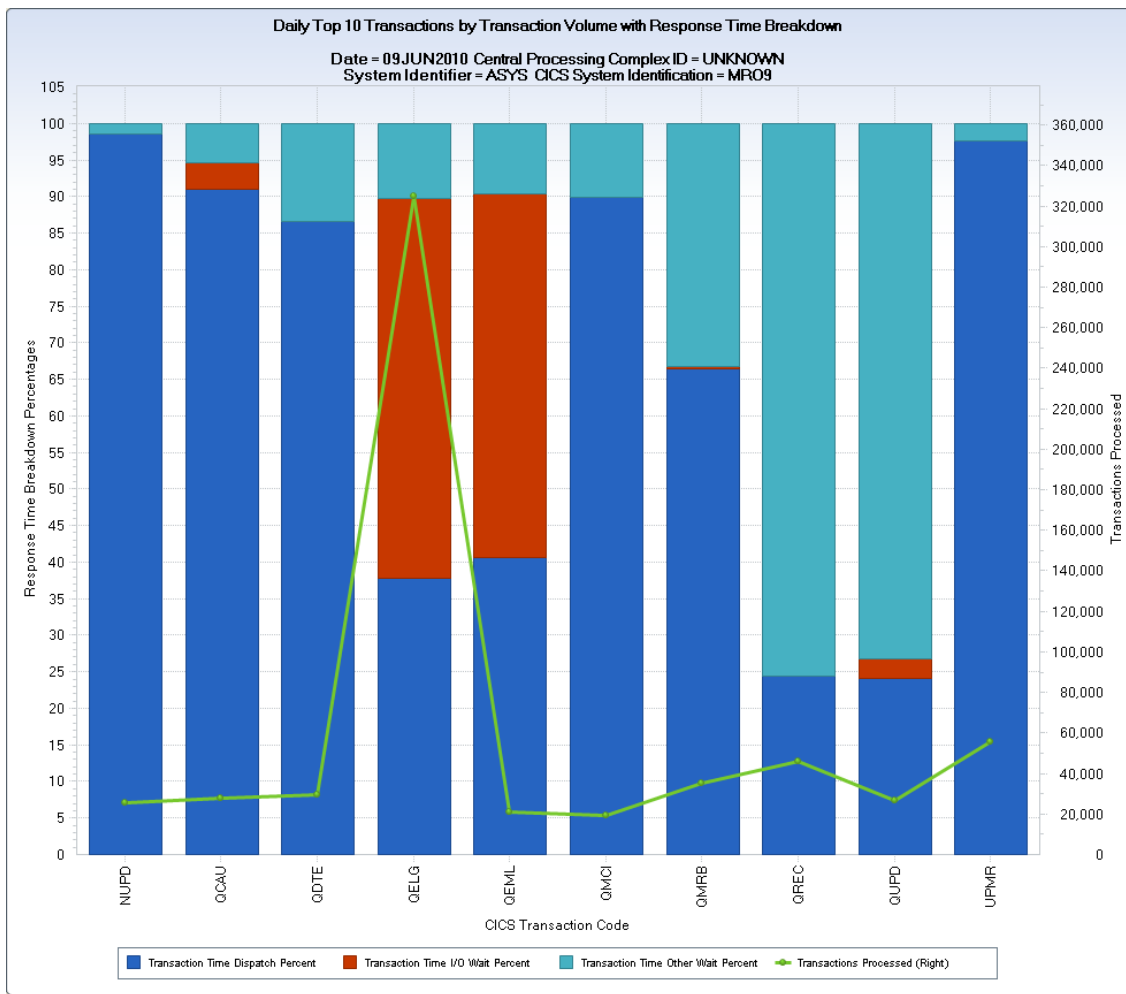
The following modifications can be made to the query:

- Add a Data Filter to the Input CICCYSY File step to select specific SYSIDs or CICS regions (CICSIDs).
- Update the SAS code in the Query Control Vars step to modify the response time goal (default 1 second) or percent of transactions to meet response time goal (default 95 percent).

Note: If you change the response time goal (default 1 second), modify the second line of the query chart title to reflect the changed value. The default value of 1 second is hardcoded, and you can modify it using the Edit View facility when viewing the charts.

CICDTT – CICS Daily Top 10 Transactions





Overview

The CICDTT query generates two charts for each SYSID and CICS region combination. The query uses the DAYS timespan CICS User Activity file as input, and reports on yesterday's top ten transactions for each CICS region. This query requires that one of the CA MICS CICTACT nn user-defined account codes is set equal to TRANCODE. Before you run the query, update the query TRANCODE derivation to use the CICTACT nn value assigned to TRANCODE in your CICALRT Account Routine.

The CICDTT query selects the top nn (default top 10) transactions, by volume, for each CICS region. The top ten transactions are graphically charted using two separate output views.

The first view shows the top ten transactions daily volume using vertical bars, and the average CICS transaction response time for each transaction with a horizontal line.

The second view shows a response time breakdown for the same top ten transactions. The view uses a stacked bar composed of the percentage of response time spent in dispatch, I/O wait, and other wait time. The total transaction count for each CICALPU is displayed with a horizontal line.

The examples show the charts generated for a CICS region "MR09" executing on system "ASYS." Merging with data from the HARCPU file was not able to determine the Central Processing Complex (CPC) ID, hence the CPC ID is 'UNKNOWN.' The first chart shows that the most frequently run transaction is "QELG" with 325,148 executed with an average transaction response time of 0.00474 seconds.

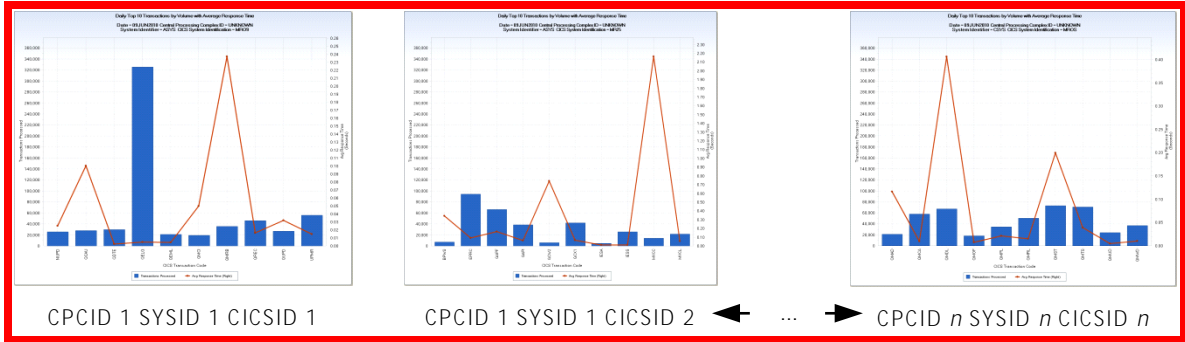
The second chart shows that the "QELG" transaction spent 52 percent of the time waiting on I/O and 10 percent of the time waiting for non-I/O related delays.

Uses

This query is useful to for analyzing transaction volume and performance characteristics. By selecting the top ten transactions, based on transaction volume, you can focus your analysis on the transactions most frequently executed by each CICS region. Where transactions show significant percentages of time waiting on I/O or other delaying factors, further analysis may direct tuning efforts to improve overall response time.

Data Extracts

CICS Daily Top 10 Trans Resp Time Average



Left Y-axis

CSUTRANS - Transactions Processed

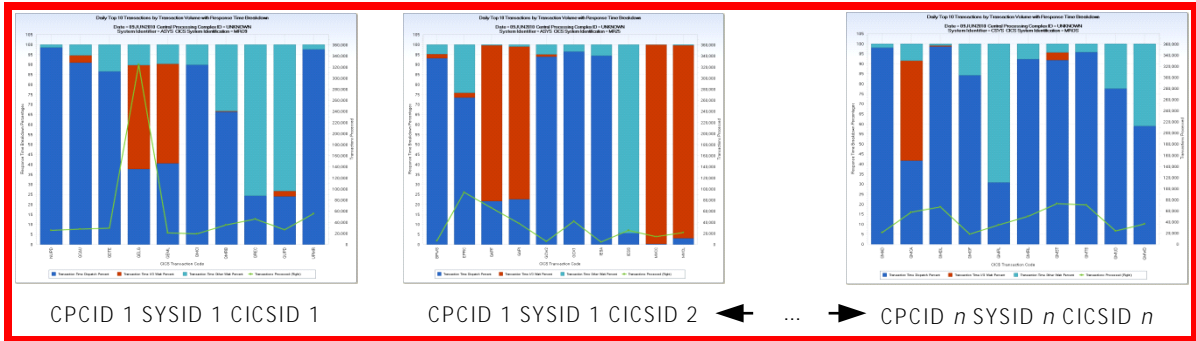
Right Y-axis

CSUAVTTM - Avg Response Time

X-axis

TRANCODE – CICS Transaction Code

CICS Daily Top 10 Trans Resp Tm Breakdown



Left Y-axis

DISP_PCT – Transaction Time Dispatch Percent

IOW_PCT – Transaction Time I/O Wait Percent

OTHW_PCT – Transaction Time Other Wait Percent

Right Y-axis

CSUTRANS – Transactions Processed

X-axis

TRANCODE – CICS Transaction Code

Query Defaults

- CICCSU DAYS timespan file: Cycle 01
- Derivations:
 - DATE
`DATEPART(ENDTS) ;`
 - TRANCODE
`TRANCODE=CICACT2 ; ⚠ Must modify to match your settings!`
 - CPCID
`CPCID = TRIM(CPUMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' ||
SUBSTR(CPCSEQNB,12,5) ;`
 - DISP_PCT
`DISP_PCT = 100* (CSUPETTM / SUM(CSUPETTM,CSUIOWAT,CSUOTWAT)) ;`
 - IOW_PCT
`IOW_PCT = 100* (CSUIOWAT / SUM(CSUPETTM,CSUIOWAT,CSUOTWAT)) ;`
 - OTHW_PCT
`OTHW_PCT = 100* (CSUOTWAT / SUM(CSUPETTM,CSUIOWAT,CSUOTWAT)) ;`

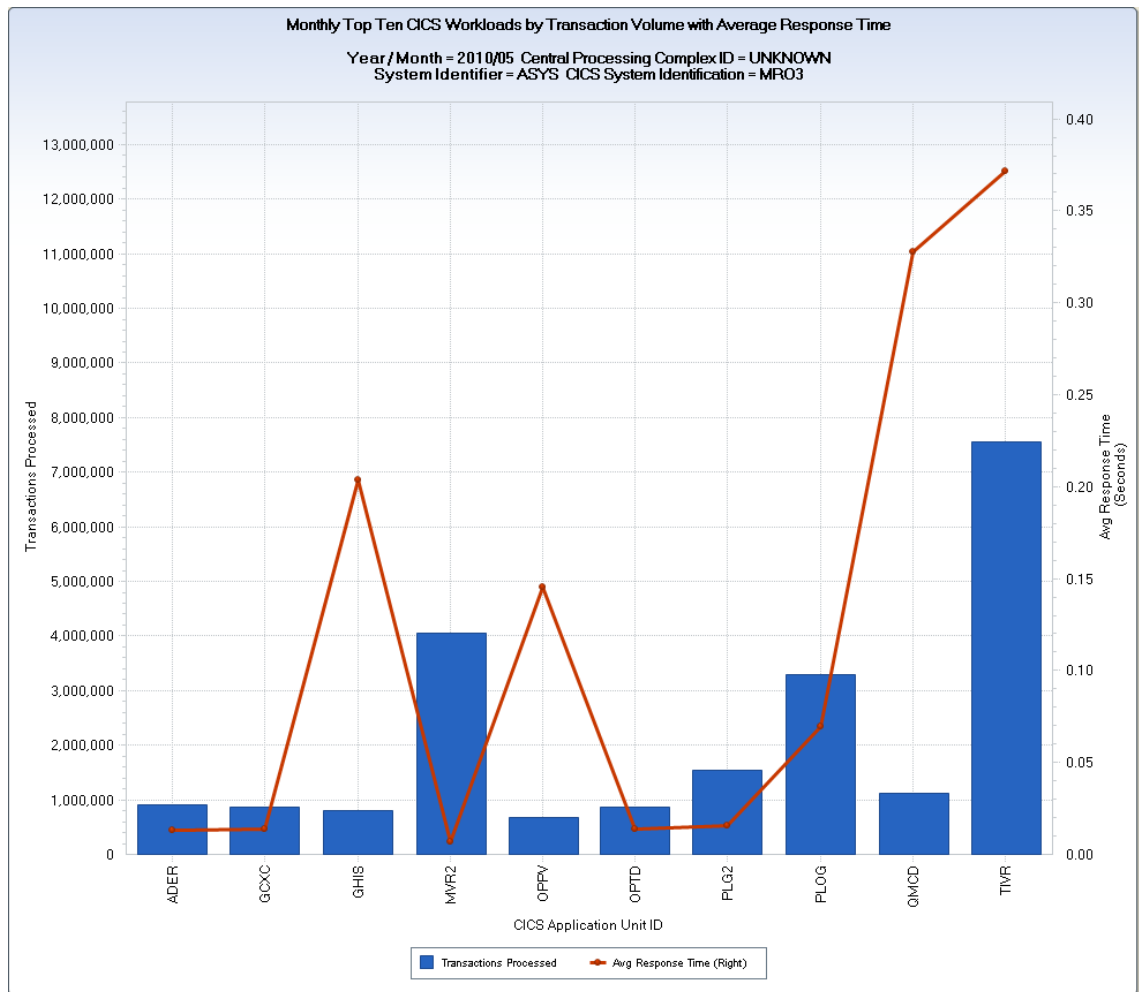
Modifications

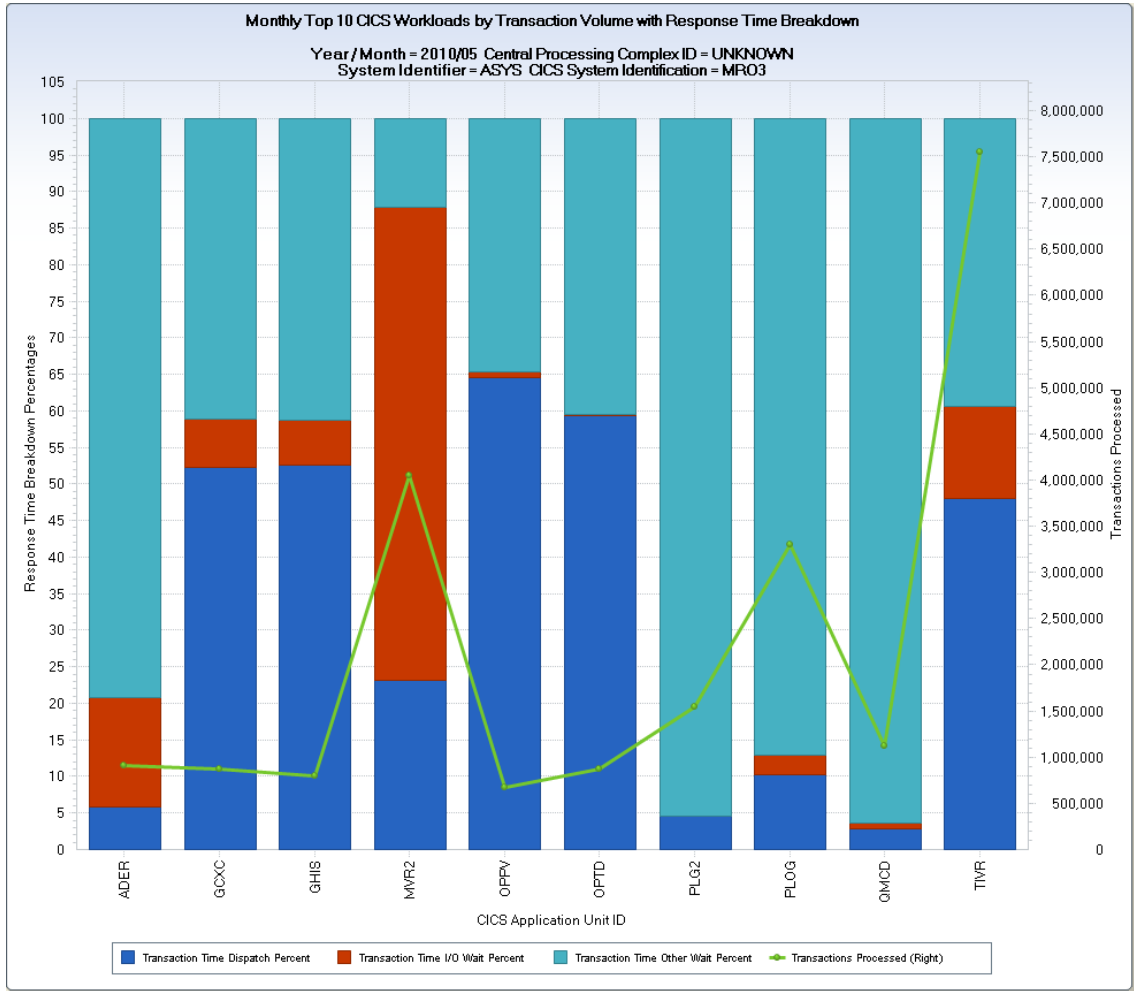
The following modifications can be made to the query:

- You can adjust the TOP nn value to something other than the default of 10.
- You can add filter statements to select only a single SYSID or CICSID, or select only specific production CICS regions.
- You can increase the cycles input to create charts for multiple days, but comment out the user code in the CICCSU file input step that restricts processing to yesterday's data.

Note: If you change the TOP nn value (default TOP 10), modify the query chart title to reflect the changed value.

CICMAT – CICS Monthly Top 10 Workloads





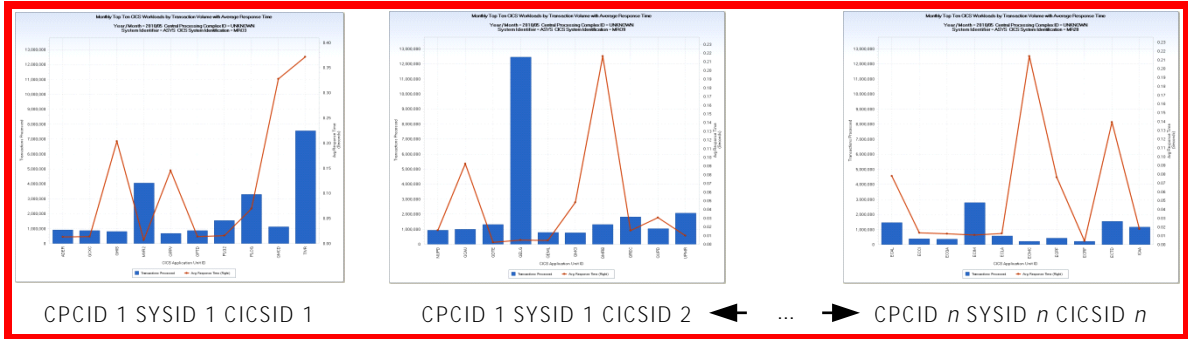
Overview

The CICMAT CICS Monthly Top 10 Workload Analysis by Region query generates two charts for each SYSID and CICS Region combination. The query uses the MONTHS timespan CICS Application Unit Activity file as input, and reports on last month’s top ten user-defined CICS Application Unit IDs (CICAPU).

Note: For a detailed explanation of the query uses and outputs, see the CICDAT CICS Daily Top 10 Workload Analysis by Region documentation. The only difference between the CICMAT and CICDAT queries is time period reported—last month as opposed to yesterday, respectively.

Data Extracts

CICAPU Month Top 10 Response Time Avg



Left Y-axis

CAUTRANS - Transactions Processed

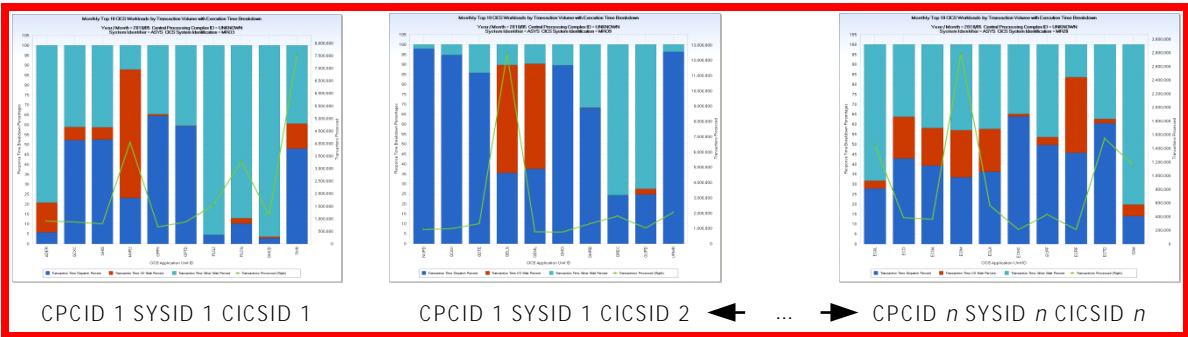
Right Y-axis

CAUAVTTM - Avg Response Time

X-axis

CICAPU – CICS Application Unit ID

CICAPU Month Top 10 Resp Time Breakdown



Left Y-axis

DISP_PCT – Transaction Time Dispatch Percent

IOW_PCT – Transaction Time I/O Wait Percent

OTHW_PCT – Transaction Time Other Wait Percent

Right Y-axis

CAUTRANS – Transactions Processed

X-axis

CICAPU – CICS Application Unit ID

Query Defaults

- CICCAU MONTHS timespan file: Cycle 01
- Derivations:
YRMONTH

```
IF DAY( DATEPART( ENDTs )) NE 1  
  THEN YRMONTH =DATEPART( ENDTs ) -DAY( DATEPART( ENDTs )) +1;  
  ELSE YRMONTH = DATEPART( ENDTs ) ;
```

CPCID

```
CPCID = TRIM( CPUMOD ) || '-' || TRIM( CPCMODID ) || '-Serial: ' ||  
SUBSTR( CPCSEQNB, 12, 5 ) ;
```

DISP_PCT

```
DISP_PCT = 100* ( CAUPETTM / SUM( CAUPETTM, CAUIOWAT, CAUOTWAT ) ) ;
```

IOW_PCT

```
IOW_PCT = 100* ( CAUIOWAT / SUM( CAUPETTM, CAUIOWAT, CAUOTWAT ) ) ;
```

OTHW_PCT

```
OTHW_PCT = 100* ( CAUOTWAT / SUM( CAUPETTM, CAUIOWAT, CAUOTWAT ) ) ;
```

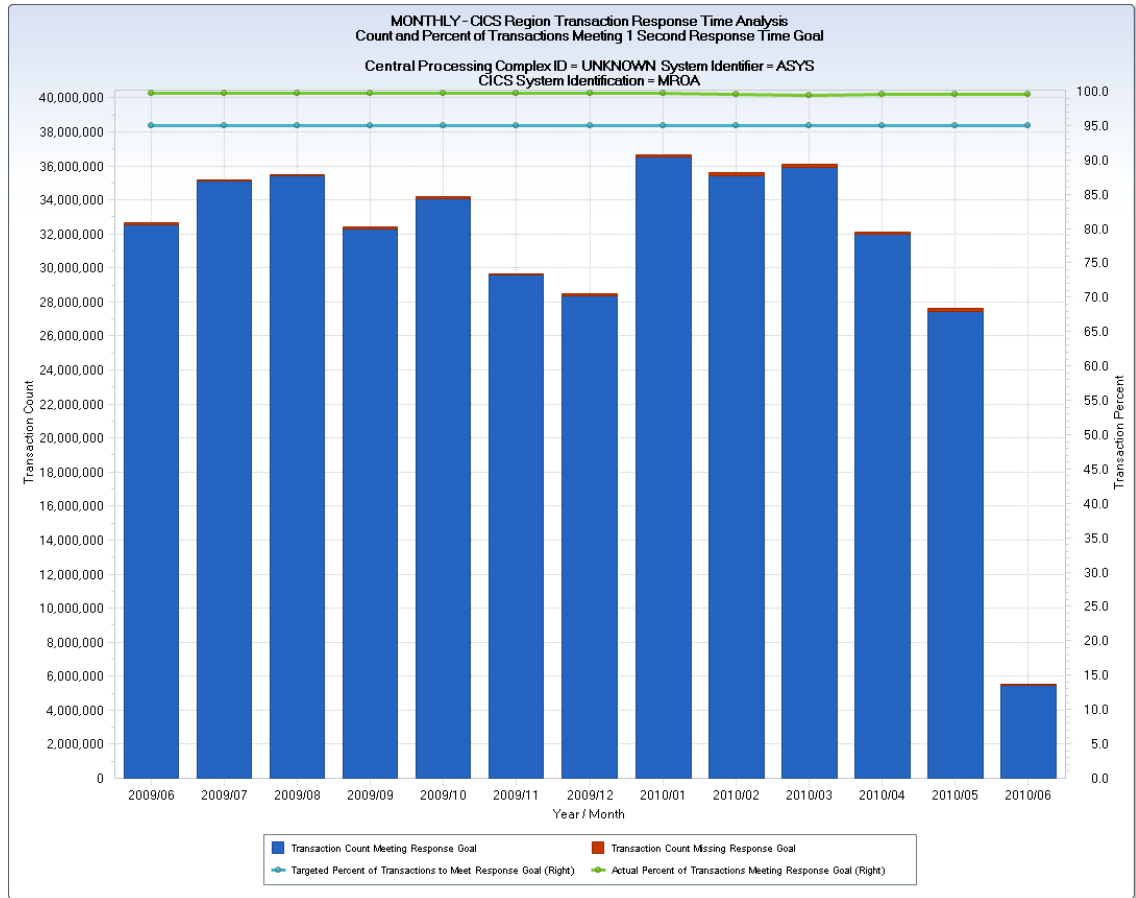
Modifications

The following modifications can be made to the query:

- You can adjust the `TOPnn` value to something other than the default of 10. You can add filter statements to select only a single `SYSID` or `CICSID`, or select only specific production CICS regions.
- You can increase the cycles input to create charts for multiple months.
- Apply a format to the `CICAPU` variable to convert it from the 12 character workload identifier to a more meaningful workload description.

Note: If you change the `TOPnn` value (default TOP 10), modify the query chart title to reflect the changed value.

CICMTR – CICS Monthly Tran Count and Response Pct



Overview

The CICMTR query generates charts showing overall transaction counts and response time metrics, by month, for CICS regions. A separate chart is produced for each z/OS SYSID and CICS region combination. The charts show the total transaction count for each month using a stacked vertical bar—where the overall transaction count is broken down into two bar segments. One segment shows the transaction count that met a user-specified elapsed time goal using a blue bar (default one second), and the other segment shows the transaction count that missed the goal using a red bar.

In addition, the chart uses a horizontal line to show the percentage of transactions, per month, that met the response time goal (green line). A user-specified desired percentage (default 95 percent) is plotted (blue line) to allow for easy visualization of the actual percentage of transactions, compared to the desired percentage of transactions, that met the response time goal.

The example shows the chart generated for a CICS region "MROA" running on a z/OS system "ASYS". Merging with data from the HARCPU file was not able to determine the Central Processing Complex (CPC) ID, hence the CPC ID is 'UNKNOWN.' The chart shows that transaction volume decreased in November and December 2009, and increased significantly for the months January through March 2010, falling off again in April and May. The final month, June 2010, use cycle 00 of the CICCYS file, representing a partial month. Close to 100 percent of the transactions, each month, met the response time goal of completing in one second or less.

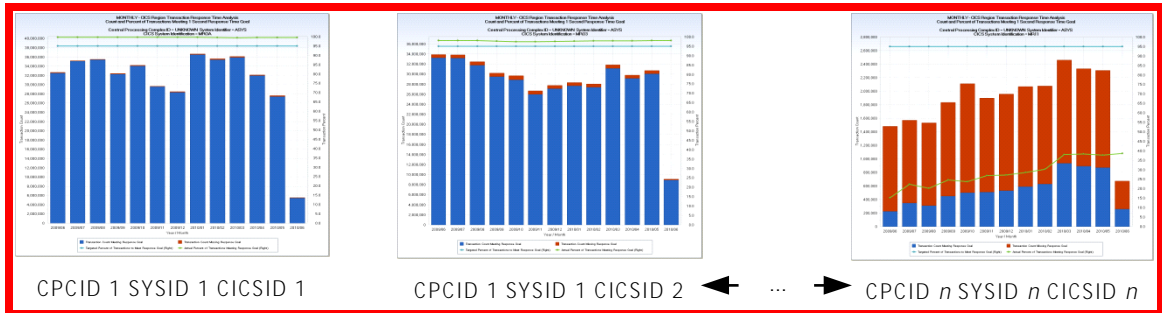
The second line of the title shows the response time target—and must be manually updated in the query output view definition if the response time value is changed from the default value of one (1) second.

Uses

This query allows you to evaluate the overall CICS transaction performance quickly at the CICS region level. If you have established specific Service Level Agreements (SLAs) for CICS regions where *nn* percent of CICS transactions must complete in *mm* seconds or less, query parameters can be modified so that these charts can easily identify regions and time periods where the goal was not met.

Data Extracts

CICS Monthly Transaction Response Goal Report



Left Y-axis

MEET_CNT: Transaction Count Meeting Response Goal

MISS_CNT: Transaction Count Missing Response Goal

Right Y-axis

GOAL_PCT: Targeted Percent of Transactions to Meet Response Goal

MEETL_PCT: Actual Percent of Transactions Meeting Response Goal

X-axis

Year / Month

Query Defaults

- CICCSY MONTHS timespan file: CYCLES 0 – 12
- HARCPU DAYS timespan file: CYCLE 01 (Used to determine CPCID<->SYSID mapping)
- User code defines response goal targets:
 - Response time goal: One (1) second – MACRO VAR &RESPVAL
 - Pct of transactions to meet Response Time Goal: 95% - MACRO VAR GOALPCT

■ Derivations:

YRMONTH

```
IF DAY(DATEPART(ENDTS)) NE 1
THEN YRMONTH =DATEPART(ENDTS)-DAY(DATEPART(ENDTS)) +1;
ELSE YRMONTH = DATEPART(ENDTS) ;
```

CPCID

```
CPCID = TRIM(CPUMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' ||
SUBSTR(CPCSEQNB,12,5) ;
```

MISS_CNT

```
= SUM(CSYTRANS, -MEET_CNT) ;
```

GOAL_PCT

```
= &GOALPCT ;
```

MEET_PCT

```
= (MEET_CNT / CSYTRANS) * 100;
```

MEET_CNT

```
if cicrval7 <= &RESPVAL then do;
  MEET_CNT=sum(CSYTDST7,CSYTDST6,CSYTDST5,CSYTDST4,
  CSYTDST3,CSYTDST2,CSYTDST1);
end;
else if cicrval6 <= &RESPVAL then do;
  MEET_CNT=sum(CSYTDST6,CSYTDST5,CSYTDST4,CSYTDST3,
  CSYTDST2,CSYTDST1);
end;
else if cicrval5 <= &RESPVAL then do;
  MEET_CNT = sum(CSYTDST5,CSYTDST4,CSYTDST3,CSYTDST2,CSYTDST1);
end;
else if cicrval4 <= &RESPVAL then do;
  MEET_CNT = sum(CSYTDST4,CSYTDST3,CSYTDST2,CSYTDST1);
end;
else if cicrval3 <= &RESPVAL then do;
  MEET_CNT = sum(CSYTDST3,CSYTDST2,CSYTDST1);
end;
else if cicrval2 <= &RESPVAL then do;
  MEET_CNT = sum(CSYTDST2,CSYTDST1);
end;
else if cicrval1 <= &RESPVAL then MEET_CNT = CSYTDST1;
```

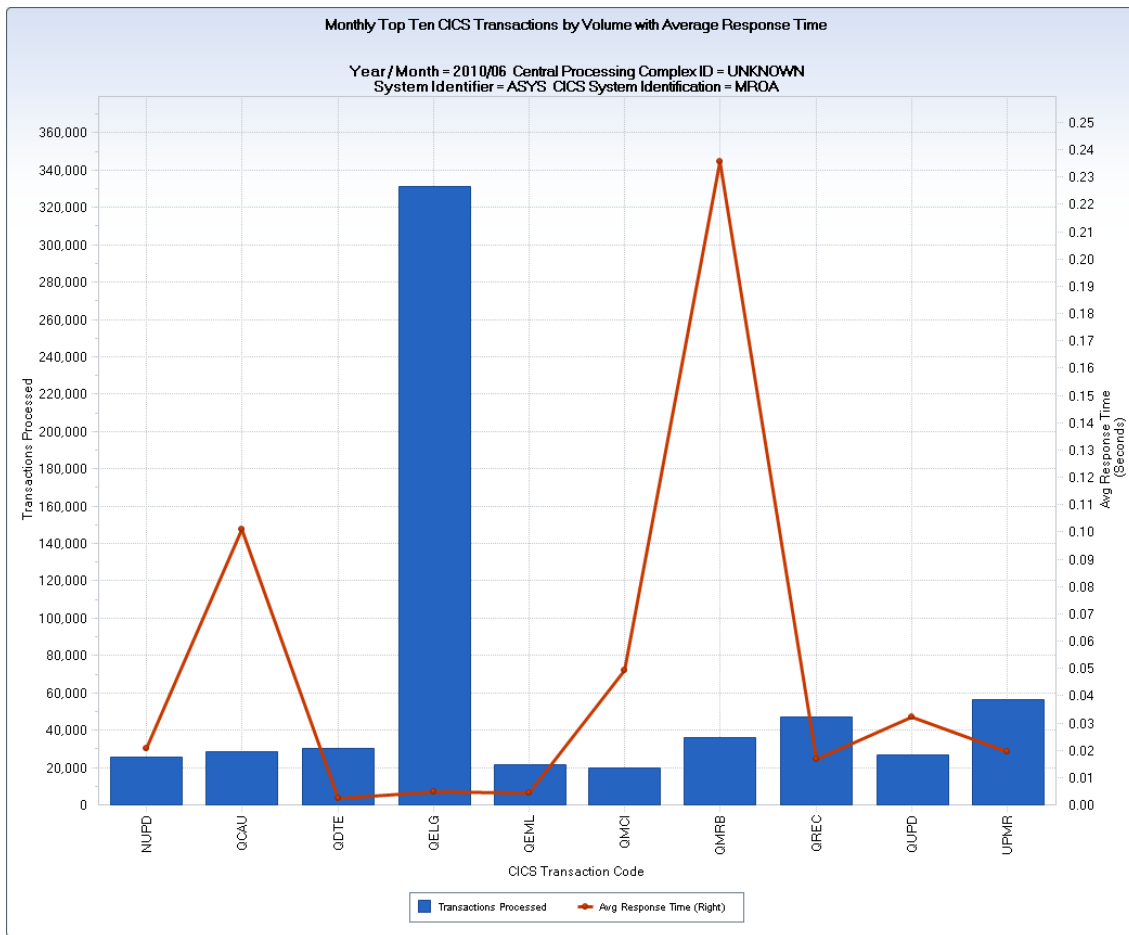
Modifications

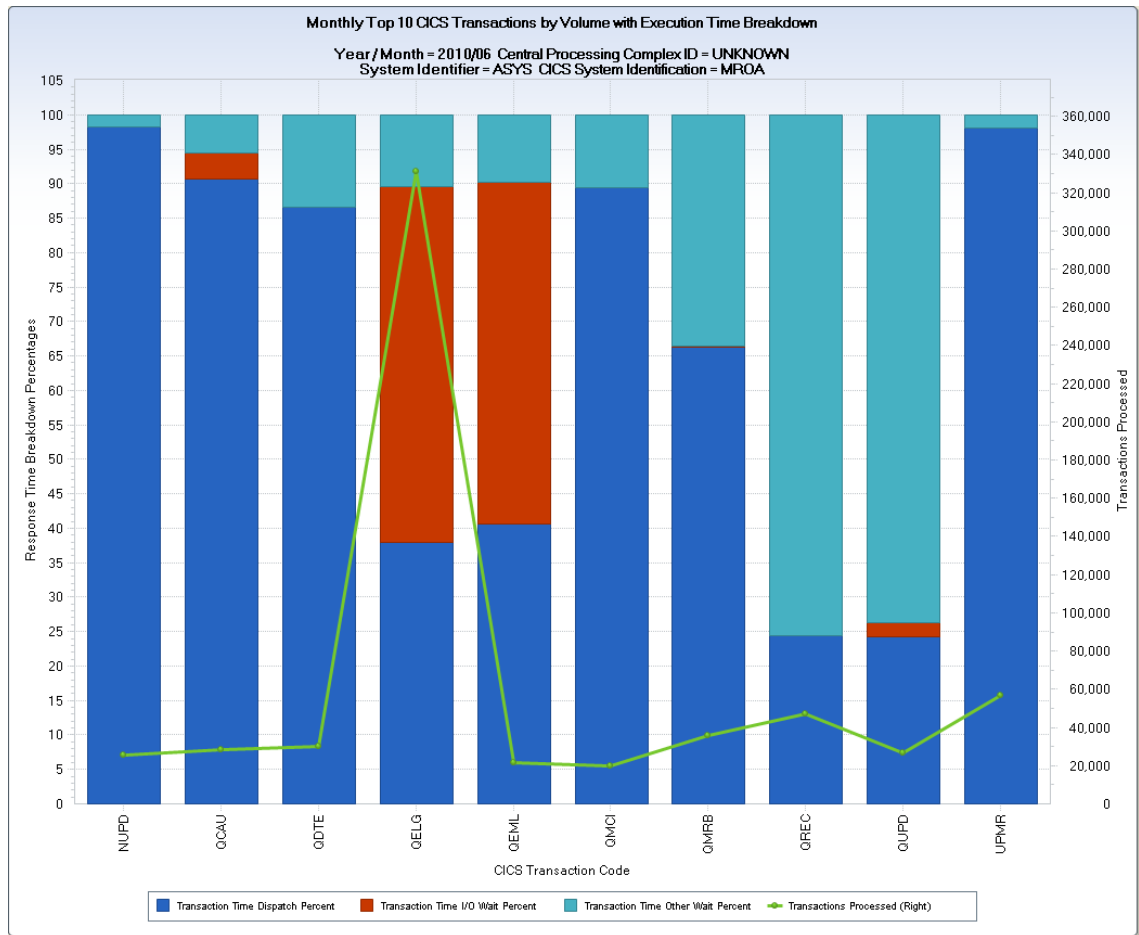
The following modifications can be made to the query:

- Add a Data Filter to the Input CICCYSY File step to select specific SYSIDs or CICS regions (CICSIDs).
- Update the SAS code in the Query Control Vars step to modify the response time goal (default 1 second) or percent of transactions to meet response time goal (default 95 percent).

Note: If you change the response time goal (default 1 second), modify the second line of the query chart title to reflect the changed value. The default value of 1 second is hardcoded, and you can modify it using the Edit View facility when viewing the charts.

CICMTT – CICS Monthly Top 10 Transactions





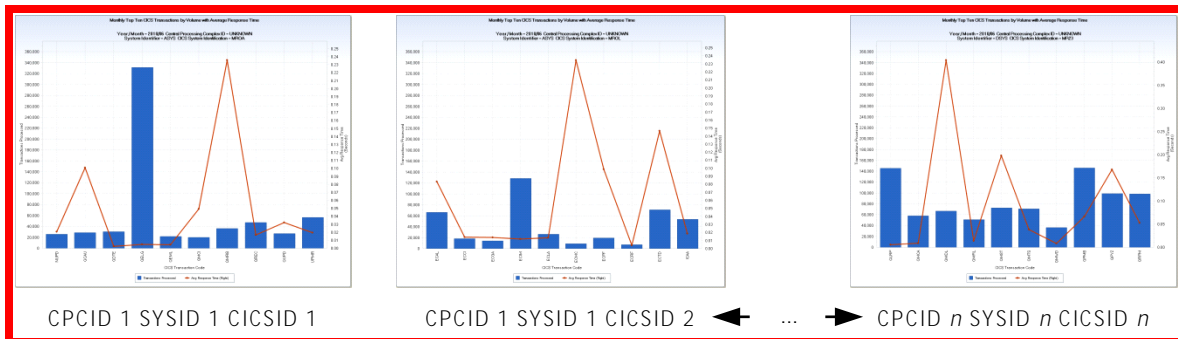
Overview

The CICMTT query generates two charts for each SYSID and CICS Region combination. The query uses the MONTHS timespan CICS User Activity file as input, and reports on the last month’s top ten transactions, by execution volume, for each CICS region.

Note: For a detailed explanation of the query uses and outputs, see the CIDTT CICS Daily Top 10 Transactions Analysis by Region documentation. The only difference between the CICMTT and CIDTT queries is the time period reported—last month as opposed to yesterday, respectively.

Data Extracts

CICS Monthly Top 10 Trans Resp Time Avg



Left Y-axis

CSUTRANS - Transactions Processed

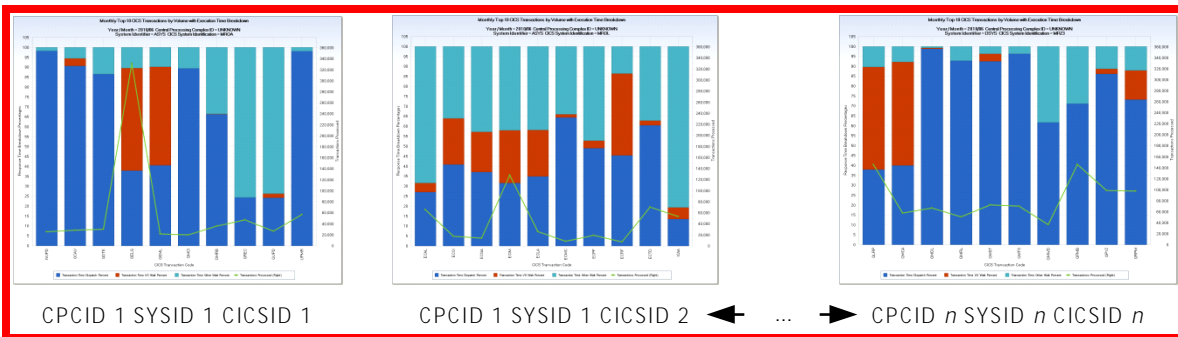
Right Y-axis

CSUAVTTM - Avg Response Time

X-axis

TRANCODE – CICS Transaction Code

CICS Monthly Top 10 Trans Resp Tm Breakdown



Left Y-axis

DISP_PCT – Transaction Time Dispatch Percent

IOW_PCT – Transaction Time I/O Wait Percent

OTHW_PCT – Transaction Time Other Wait Percent

Right Y-axis

CSUTRANS – Transactions Processed

X-axis

TRANCODE – CICS Transaction Code

Query Defaults

- CICCSU MONTHS timespan file: Cycle 01
- Derivations:
 - YRMONTH
 - IF DAY(DATEPART(ENDTS)) NE 1
 - THEN YRMONTH =DATEPART(ENDTS)-DAY(DATEPART(ENDTS)) +1;
 - ELSE YRMONTH = DATEPART(ENDTS) ;
 - TRANCODE
 - TRANCODE=CICACT2 ; ⚠ Must modify to match your settings!
 - CPCID
 - CPCID = TRIM(CPUMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' ||
 - SUBSTR(CPCSEQNB,12,5) ;
 - DISP_PCT
 - DISP_PCT = 100* (CSUPETTM / SUM(CSUPETTM, CSUIOWAT, CSUOTWAT)) ;
 - IOW_PCT
 - IOW_PCT = 100* (CSUIOWAT / SUM(CSUPETTM, CSUIOWAT, CSUOTWAT)) ;
 - OTHW_PCT
 - OTHW_PCT = 100* (CSUOTWAT / SUM(CSUPETTM, CSUIOWAT, CSUOTWAT)) ;

Modifications

The following modifications can be made to the query:

- You can adjust the TOP nn value to something other than the default of 10.
- You can add filter statements to select only a single SYSID or CICSID, or select only specific production CICS regions.
- You can increase the cycles input to create charts for multiple months

Note: If you change the TOP nn value (default TOP 10), modify the query chart title to reflect the changed value.

Chapter 3: CA MICS Analyzer Option for DB2 (DB2) Queries

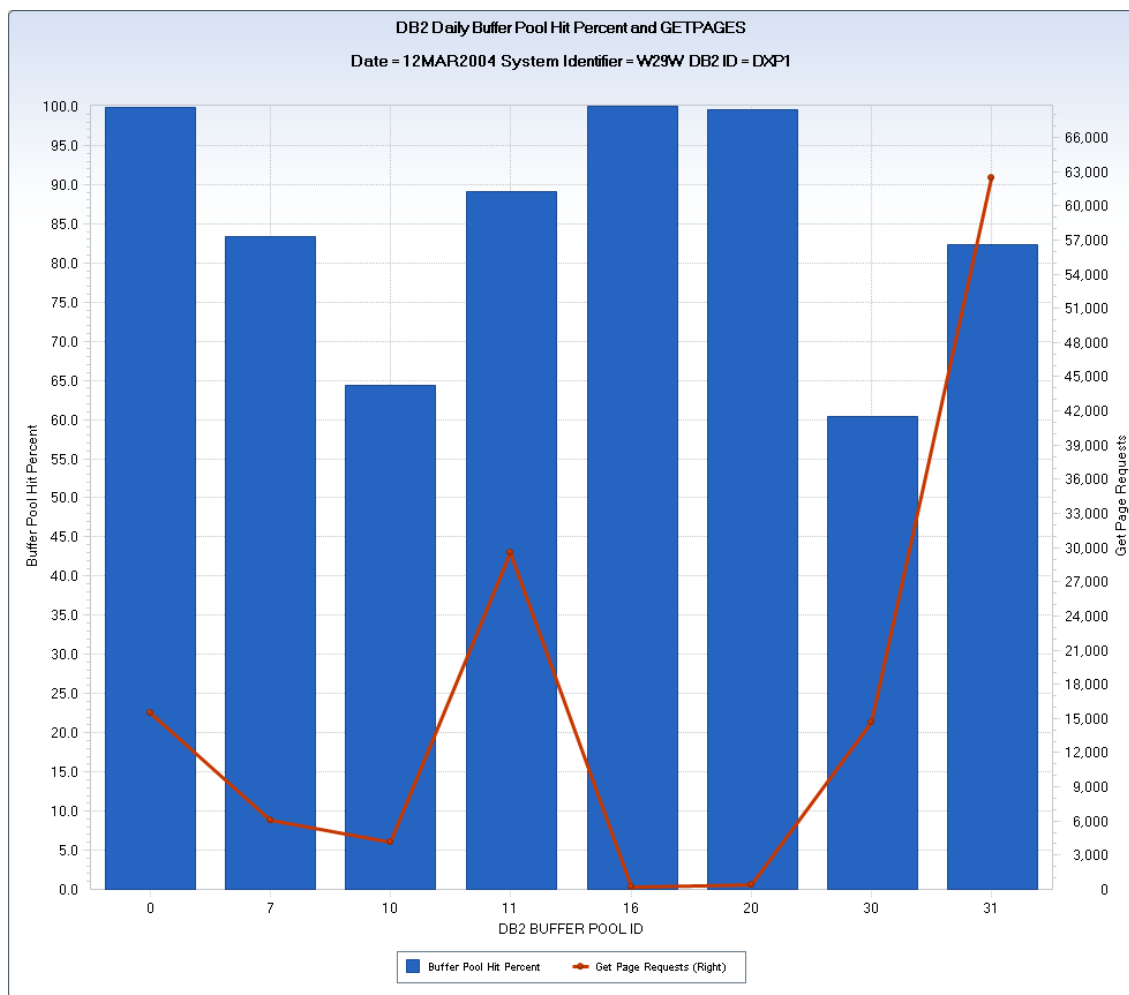
This section contains the following topics:

[DB2BPH – DB2 Daily Buffer Pool Hit Percent](#) (see page 51)

[DB2T10 – DB2 Daily Top Ten Plans by Thread Count](#) (see page 54)

[DB2TCT – DB2 Daily Thread Count by Connection Type](#) (see page 59)

DB2BPH – DB2 Daily Buffer Pool Hit Percent



Overview

The DB2BPH query produces a daily chart. The chart shows the total Get Page Requests for each buffer pool and the percentage of requests that were satisfied from the buffer pool (hits). The hit percentage is displayed using vertical bars and the total Get Page Requests are shown using a horizontal line.

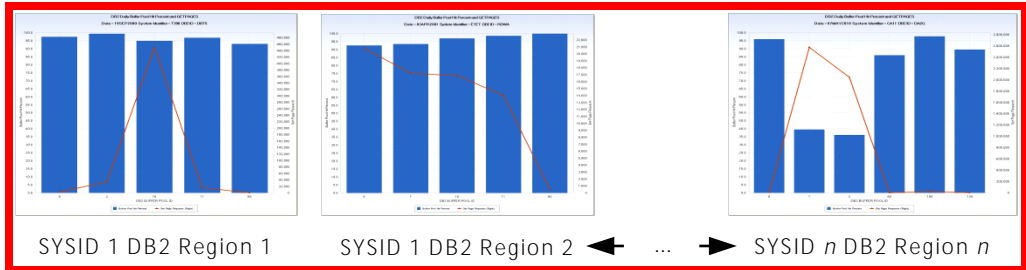
The example shows the daily chart generated for DB2 Region DXP1 running on SYSID W29W. Buffer pool 31 had the greatest number of Get Page Requests (62,489) with a buffer pool hit percentage of 82.3 percent.

Uses

This query is useful to identify buffer pools that are not large enough to hold frequently requested data. When requested data is not in the buffer pool, the data must be acquired by reading from DASD, degrading transaction response time. Analyze buffer pools with low hit percentages to determine if changes can be made to improve transaction performance.

Data Extracts

DB2 Daily Buffer Pool Hit Percent



Left Y-axis

BPHIT – Buffer Pool Hit Percent

Right Y-axis

DBUGPGR – Get Page Requests

X-axis

DB2PID – Buffer Pool ID

Query Defaults

- DBPDBU DAYS timespan file: Cycle 01
- Derivations:
DATE
DATEPART(ENDTS)
BPHIT
TEMPDPF=DBUDPF ;
TEMP=DBUGPGR-SUM(DBUPRFH, DBULPF, TEMPDPF, DBUSRIO);
IF DBUGPGR GT 0 AND TEMP GT 0 THEN DO ;
 BPHIT = (TEMP / DBUGPGR) * 100;
END ;

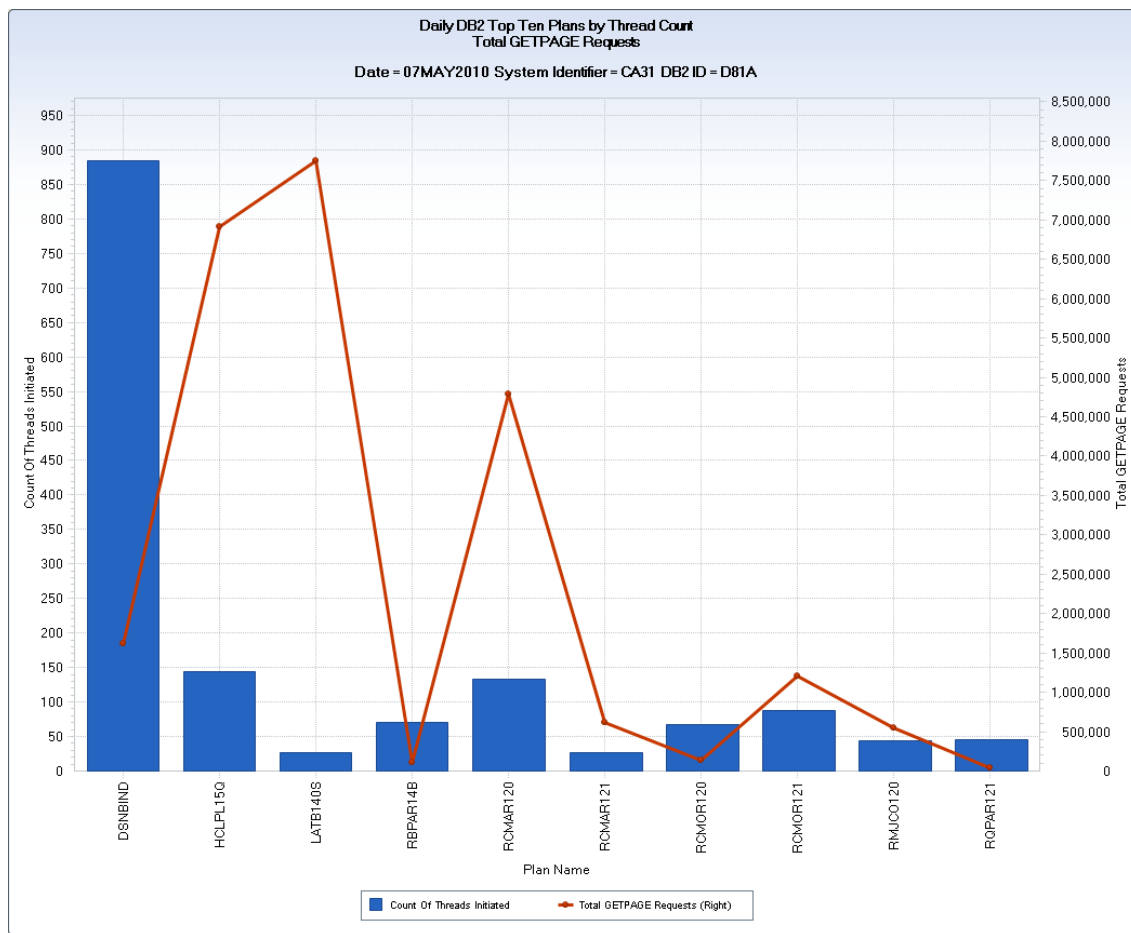
Modifications

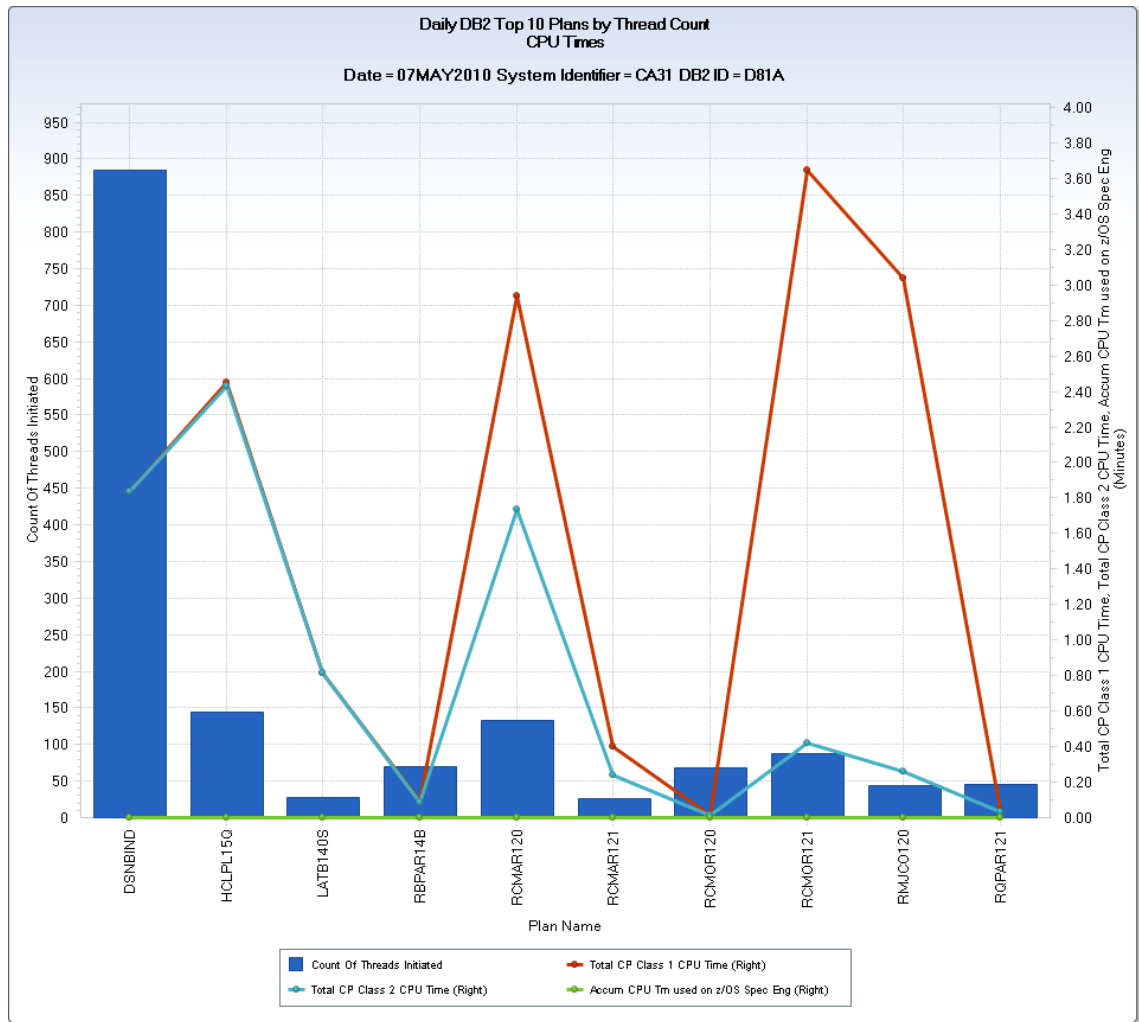
The following modifications can be made to the DB2BPH query:

- You can add filter statements to select only a single SYSID or DB2 Region (DB2ID).
- If you do not have the DAYS timespan DBPDBU file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

Note: This query charts a nonessential data element: DBUDPF (No. Times Dynamic Prefetch Req). If this data element is not active in your DBPDBU file, the query will not fail, but the computed value for buffer hit ratio may be understated. You can ask your CA MICS administrator to check and see if this data element is active. If not, it can be activated by changing the element definition in *sharedprefix*.MICS.GENLIB(DB2GENIN) from NAMX to NAME and then running the *sharedprefix*.MICS.CNTL(DB2CGEN) job. After this element is activated, refresh the Q&R Meta Database.

DB2T10 – DB2 Daily Top Ten Plans by Thread Count





Overview

The DB2T10 query generates two daily charts for each DB2 Region. The charts show resource usage for the top *nn* (default 10) DB2 Plans by count of threads initiated. Both charts show the number of threads initiated, using vertical bars. One of the charts shows the number of GETPAGE requests, and the other chart shows Class 1, Class 2, and Specialty engine CPU time use for the top ten plans.

The top example shows the daily GETPAGE and thread count for DB2 region D81A running on SYSID CA31. DB2 Plan "LATB140S" only had 27 threads initiated, but had almost eight million GETPAGE requests.

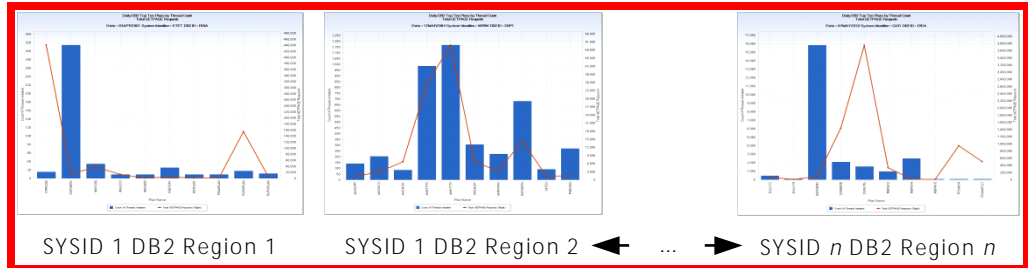
The second example shows the CPU time use for the same DB2 region. DB2 Plan "RCMOR121" used the most Class 1 CPU Time—3.65 minutes.

Uses

This query is useful to observe the resource utilization patterns for your top DB2 plans.

Data Extracts

DB2 Daily Top 10 Plans Threads and GPage



Left Y-axis

DSPNTHRD – Count of Threads Initiated

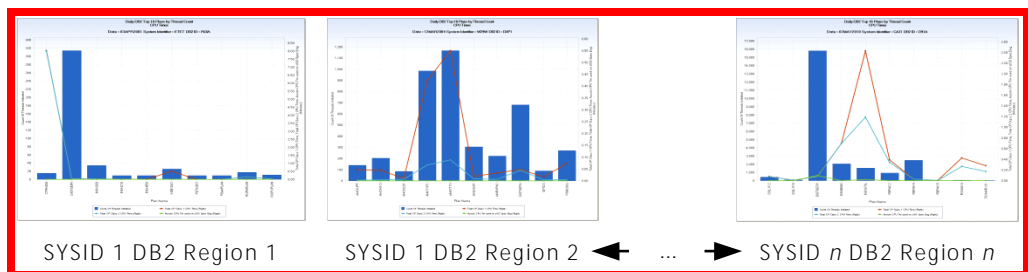
Right Y-axis

DSPSGPGR – Total GETPAGE Requests

X-axis

DB2PLAN – Plan Name

DB2 Daily Top 10 Plans Threads and CPU



Left Y-axis

DSPNTHRD – Count of Threads Initiated

Right Y-axis

DSPCP1TM – Total CP Class 1 CPU Time

DSPCP2TM – Total CP Class 2 CPU Time

DSPCPUZ1 – Accum CPU Tm used on z/OS Spec Eng

X-axis

DB2PLAN – Plan Name

Query Defaults

- DB2DSP DAYS timespan file: Cycle 01
- Derivations:
DATE
DATEPART(ENDTS)
COUNT
MAX(DSNPTHRD, SUM(DSPCOMM2,DSPABORT));

Modifications

The following modifications can be made to the query:

- You can add filter statements to select only a single SYSID or DB2ID, or select only specific production DB2 regions.
- Add additional chart views, updating the right Y-axis variable to show additional relationships for the top ten plan executions.

Query Defaults

- DB2DSP DAYS timespan file: Cycles 01 and 02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

DATE

DATEPART(ENDTS) ;

PLANEXEC

MAX(DSNPTHRD, SUM(DSPCOMM2, DSPABORT)) ;

AVTCB

DSPTCBTM / COUNT ;

AVELAPSE

DSPTLPT / COUNT ;

AVGETPG

DSPSGPGR / COUNT ;

AVSYNCRD

DSPSRIO / COUNT ;

AVSYQPF

DSPSPRFH / COUNT ;

AVMXPGLK

DSPMXPL / COUNT ;

AVUPDPG

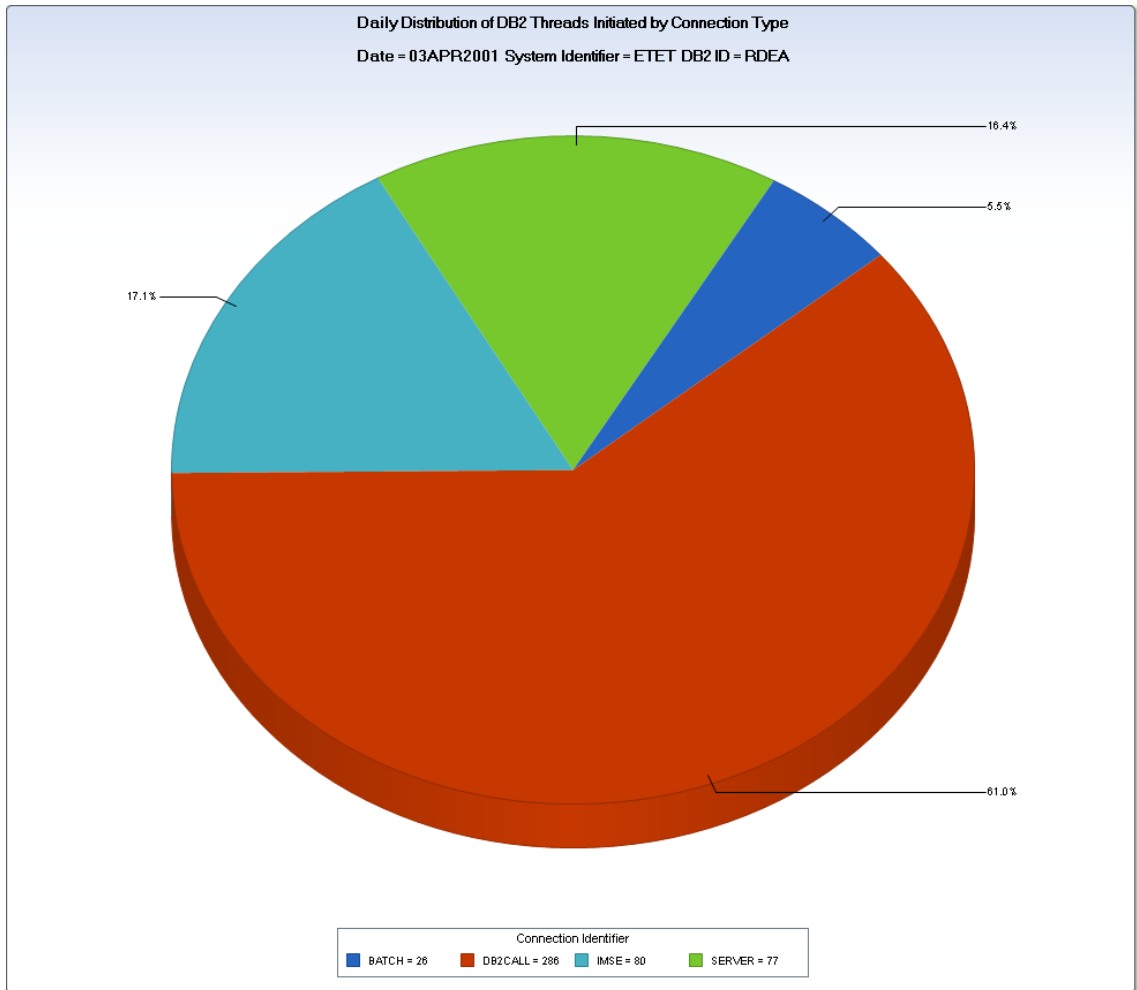
DSPSSPRU / COUNT ;

Modifications

The following modifications can be made to the DB2T10 query:

- You can add filter statements to select only a single SYSID or DB2 Region (DB2ID).
- You can adjust the TOPnn value to something other than the default of 10.
- If you do not have the DAYS timespan DB2DSP file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

DB2TCT – DB2 Daily Thread Count by Connection Type



Overview

The DB2TCT query generates a daily pie chart for each DB2 region showing the percentage of DB2 threads initiated by each connection type.

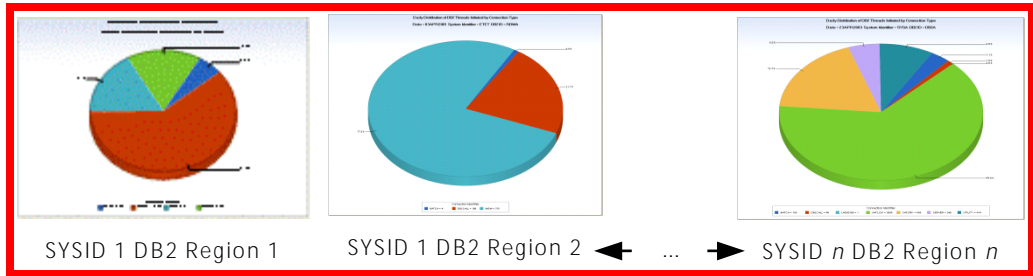
The example shows the pie chart for DB2 region RDEA running on SYSID ETET. Most threads, 61 percent, were initiated by DB2CALL. IMSE and SERVER accounted for 17.1 percent and 16.4 percent of the threads, respectively. BATCH accounted for 5.5 percent.

Uses

This query is useful for determining where DB2 activity is originating for each DB2 region. You can refer to the CA MICS data dictionary entry for DB2CONN – Connection Identifier, for complete information regarding interpretation of the connection type values.

Data Extracts

DB2 Daily Thread Count by Conn Type



X-axis (pie slices)

DSPNTHRD – Count of Threads Initiated

Query Defaults

- DB2DSP DAYS timespan file: Cycles 01 and 02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
DATE
DATEPART (ENDTS)

Modifications

The following modifications can be made to the DB2TCT query:

- You can add filter statements to select only a single SYSID or DB2 Region (DB2ID).
- If you do not have the DAYS timespan DB2DSP file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

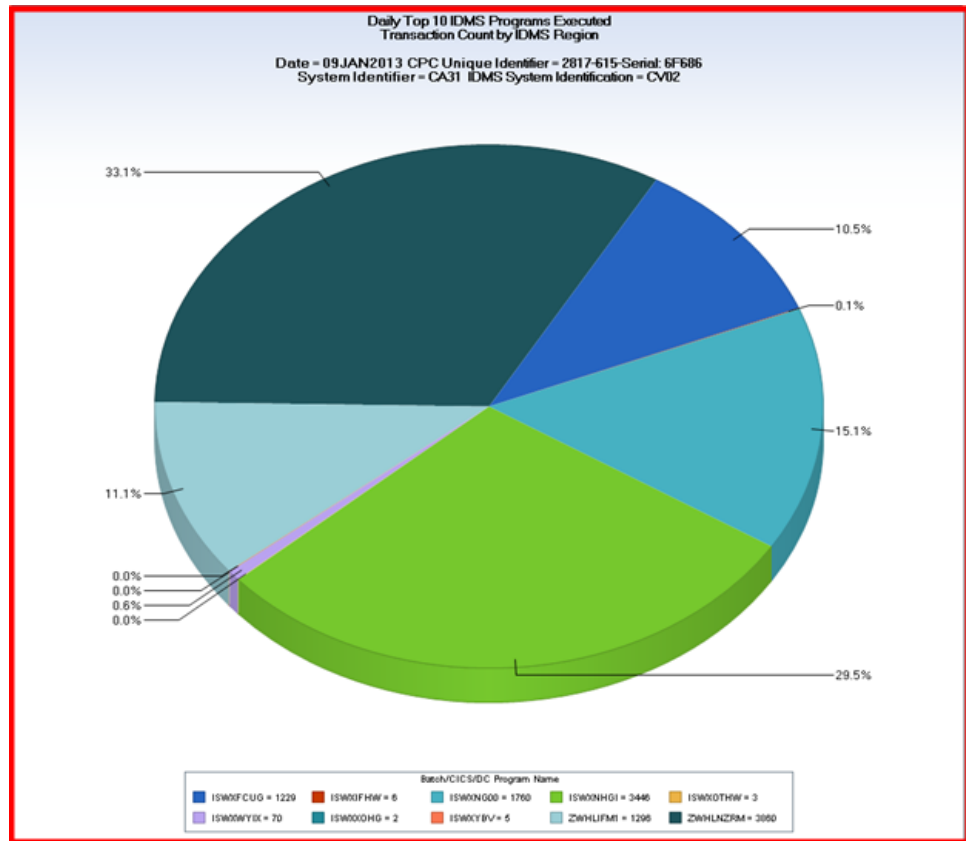
Chapter 4: CA MICS Analyzer Option for IDMS (IDM) Queries

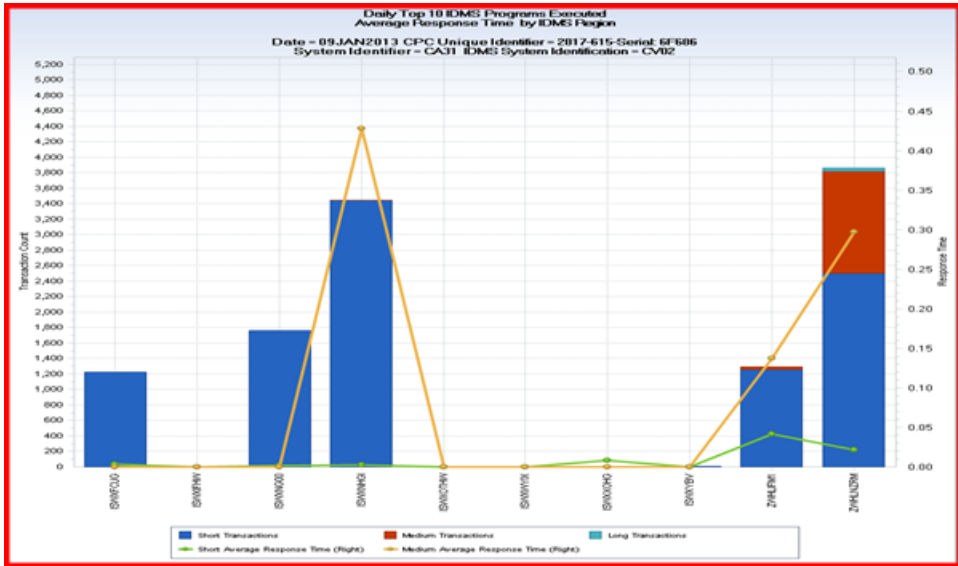
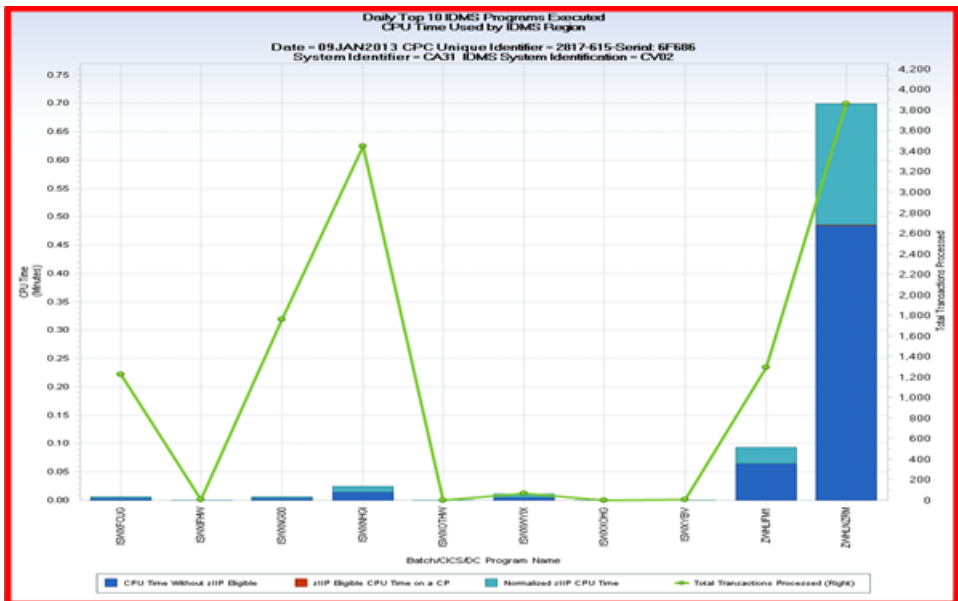
This section contains the following topics:

[IDMT10 - IDMS Daily Top Ten Programs Executed](#) (see page 61)

[IDMTYP – IDMS Daily Transaction Type Analysis](#) (see page 70)

IDMT10 - IDMS Daily Top Ten Programs Executed





Overview

The IDMT10 query produces charts that identify the programs that are the most frequently executed by IDMS transactions. The top nn programs are targeted, the default is 10. These charts display resource and performance-related metrics for these highly used programs.

The query generates two CSV output files that identify:

- The top 10 programs executed at the IDMS region level
- The top 10 programs executed at the z/OS SYSID level where the transaction data from multiple IDMS regions is consolidated

The data extract CSVs generated by the IDMT10 query are:

- Daily Top 10 IDMS Programs by Region
- Daily Top 10 IDMS Programs by SYSID

Each data extract generates the following three chart views:

- Top 10 Programs Transaction Count
- Top 10 Programs CPU Time
- Top 10 Programs Response Time

By default the query is executed against:

- DAYS timespan HARCPU file (to obtain the CPC identification for each SYSID)
- DAYS timespan IDMSUA file (to obtain the IDMS transaction information)

Each X axis value represents one of the top 10 program names. Data for the entire day is summarized to determine the most frequently executed program.

Considerations:

- By default, the SUAPGM (Batch/CICS/DC Program Name) data element is not located in the DAYS timespan IDMSUA file .
However, by convention, the sharedprefix.MICS.PARMS(IDMACRT) routine typically assigns SUAPGM to one of the IDMACTn Account Code summarization keys.
- You might see '*BLANKS*' as the name of one of the 10 most-executed programs. The IDMS transaction records sometimes contain blanks or binary zeros in the program name field. During the query execution, the query code replaces any blank or binary zero program names with the value '*BLANKS*' before it determines the 10 most-executed programs.

Chart view #1

View #1 uses pie charts to show the 10 most frequently executed programs. The number of transactions that executed the program determines the size of each pie slice.

The first chart sample (shown previously) is an example of the charts produced by View #1 from the "by Region" data extract CSV.

The chart sample shows IDMS region CV02 which is executing on SYSID CA31, running on an IBM 2817-615 (z196 CPC with 15 subcapacity CP engines). Notice that the two most frequently executed programs were ZWHLNZRM and ISWXNHGI. The two programs represent over 60% of the programs that were executed by transactions for this IDMS region (33.1% ZWHLNZRM and 29.5% for ISWXNHGI).

Chart View #2

View #2 uses stacked vertical bars to show:

- The CP processor CPU time
- CP processor CPU time that was eligible to run on a zIIP engine (but did not)
- Normalized zIIP processor CPU time used by each of the top 10 programs

Also, a horizontal line displays the transaction count for each of the programs.

The second chart sample (shown previously) is an example of the charts produced by View #2 from the "by Region" data extract CSV. It shows the top 10 programs executed by IDMS region CV02. From this chart sample you can see:

- About 1/3rd of the CPU time consumed by the most frequently executed program, ZWHLNZRM, (the right-most program on the X axis), executed on a zIIP processor engine (light blue bar segment).
- About 2/3rd of the CPU time was consumed on standard CP processor engines (dark blue bar segment)
- Only a small amount of zIIP-eligible CPU time was processed on standard CP engines (red bar segment).

Note: The second most frequently executed program, ISWXNHGI (fourth program from the left on the X axis) used much less CPU time than program ZWHLNZRM, even though the number of transactions executed was similar (3,446 for ISWXNHGI and 3,860 for ZWHLNZRM). This means that the average CPU time expended for each execution of ZWHLNZRM was much greater than for ISWXNHGI.

Chart view #3

View #3 uses stacked vertical bars to show the number SHORT, MEDIUM, and LONG transactions that executed each of the top 10 programs. In addition, the average SHORT and MEDIUM transaction response times are displayed using horizontal lines.

The third chart sample (shown previously) is an example of the charts produced by View #3 from the "by Region" data extract CSV. It shows the transaction counts and response times for the same IDMS region that is displayed in the first chart. Examining ZWHLNZRM, the most frequently executed program (the rightmost program on the X axis), you can see:

- About 1/3rd of the transactions that executed this program were MEDIUM (red bar segment) and 2/3rd of the transactions were SHORT (dark blue bar segment).
- The average transaction response times were:
 - SHORT transactions - about 0.02 seconds
 - MEDIUM transactions - about 0.30 seconds

Important! Read the [Query Defaults](#) topic for information about the updates you might have to make to execute this query successfully in your environment.

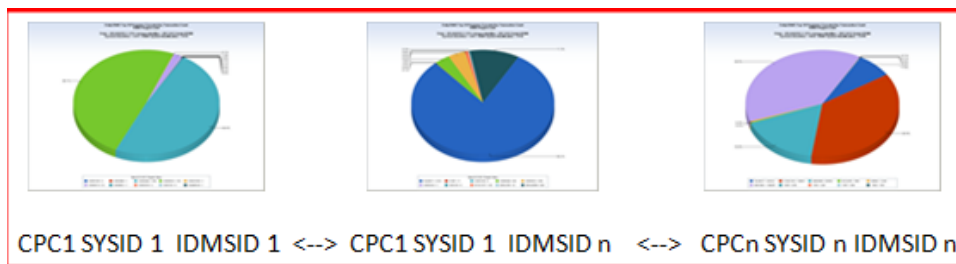
Uses

This query is useful for identifying the programs most frequently executed by your IDMS transactions. Because these programs are frequently executed, tuning efforts should yield significant dividends in reducing overall resource consumption and response time.

Data Extracts

This section includes the Daily Top 10 IDMS Programs by Region.

VIEW 1 of 3: IDMS Region Level: Top 10 Programs Executed



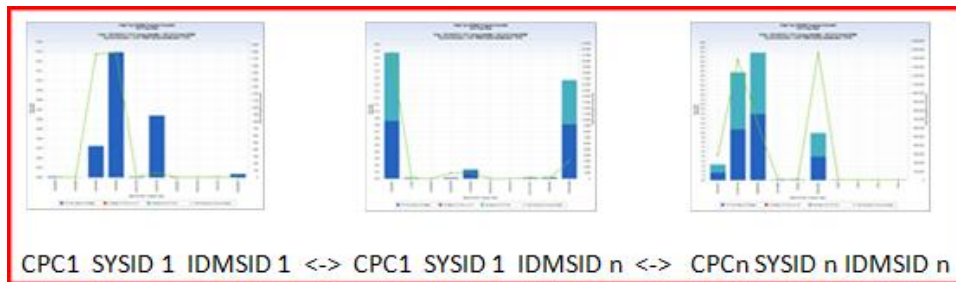
X axis

SUAPGM – Batch/CICS/DC Program Name

Note:

- Each pie slice shows the percentage of the total of all the program executions
- The legend shows the program name and execution count for each of the top 10 programs.

VIEW 2 of 3: IDMS Region Level: Top 10 Programs CPU Time



Left Y axis

SUAXCTM – CPU Time without zIIP Eligible

SUASUCTM – zIIP Eligible CPU Time on a CP

SUASPNTM – Normalized zIIP CPU Time

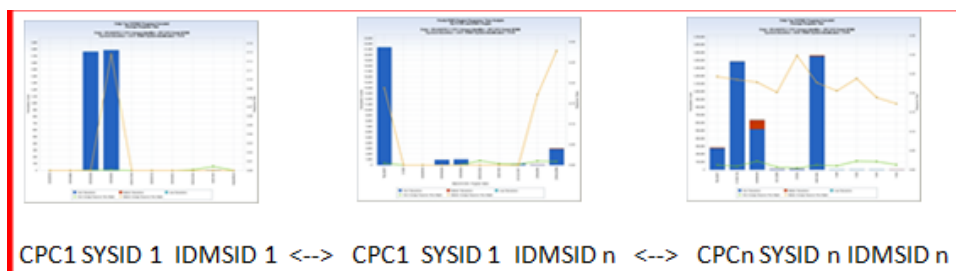
Right Y axis

SUATTRAN – Total Transactions Processed

X axis

SUAPGM – Batch/CICS/DC Program Name

VIEW 3 of 3: IDMS Region Level: Top 10 Programs Response Time



Left Y axis

SUASTRAN – Short Transactions

SUAMTRAN – Medium Transactions

SUALTRAN – Long Transactions

Right Y axis

SUAAVSTM – Short Average Response Time

SUAAVMTM – Medium Average Response Time

X axis

SUAPGM – Batch/CICS/DC Program Name

Daily Top 10 IDMS Programs by IDMS Region.

See the data extract described previously.

The charted information that is provided in this "by SYSID" level data extract is identical. The only difference is that charts are produced from data where information from multiple IDMS regions is aggregated to the SYSID level.

Therefore, instead of chart key values looking like this:

CPC1 SYSID 1 IDMSID 1 <--> CPC1 SYSID 1 IDMSID n <--> CPCn SYSID n IDMSID n

They will look like this:

CPC1 SYSID 1 <--> CPC1 SYSID 2 <--> CPCn SYSID n

Query Defaults

- HARCPU DAYS timespan file: Cycle 01
- IDMSUA DAYS timespan file: Cycles 01 and 02
- A User Code task selects data from yesterday only.

- Derivations:

DATE

DATEPART(ENDTS)

CPCUID

CPCUID = PUT(SYSID,\$CPCFMT.) ;

SUAPGM

By default, the query defaults to SUAPGM=IDMACT2.

You must examine your sp.PARMS(IDMACRT) to determine which IDMACTn element you populate with SUAPGM. Be sure this derivation task is updated to match your SUAPGM assignment.

If you do not save SUAPGM to an IDMACTn element, or you have not added SUAPGM to the DAYS timespan as a summarization key, this query can only be executed against the DETAIL timespan. If SUAPGM exists in the file, at the timespan you run with, remove this SUAPGM derivation task.

Modifications

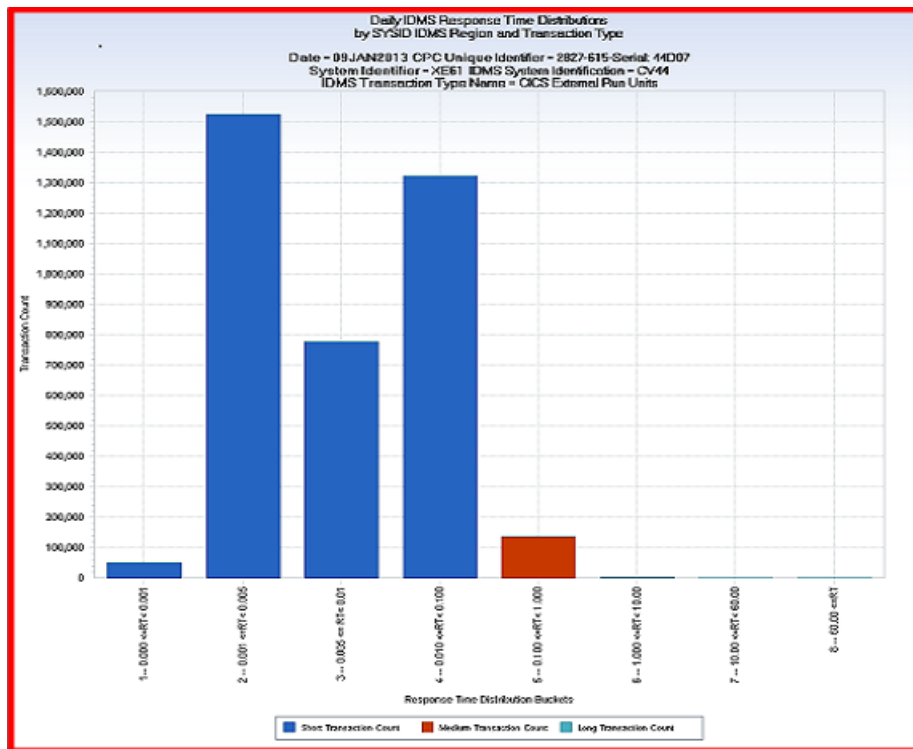
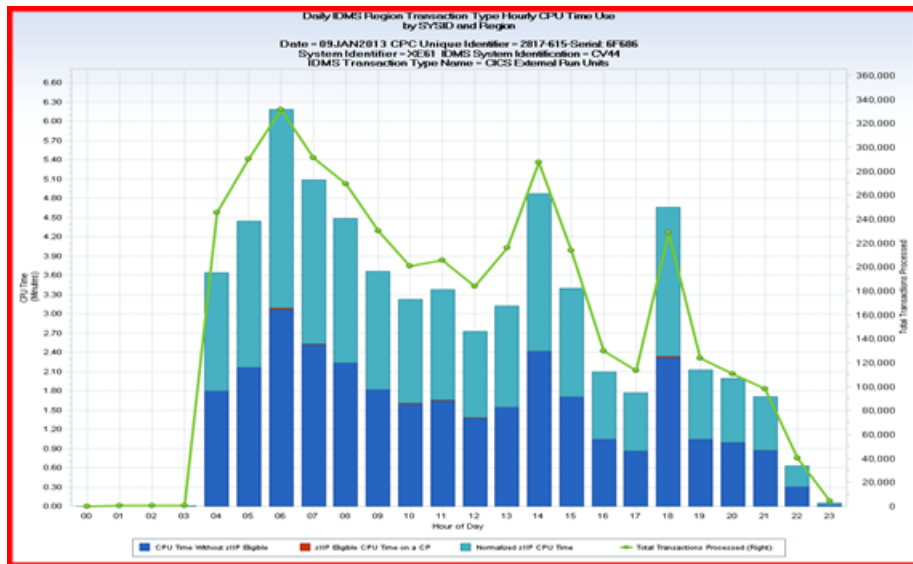
You can make the following modifications to the IDMT10 query:

- Unless your sharedprefix.MICS.PARMS(IDMACRT) member assigns SUAPGM to IDMACT2, you must modify the DERIVATION task for SUAPGM found in the query step that inputs the IDMSUA file.

See the SUAPGM derivation notes in the Query Defaults section for more information.

- Add filter statements to the IDMSUA File Selection step to select specific SYSID values or IDMS Regions (IDMSID)
- Adjust the TOPnn value to something other than the default of 10--update the assignment statement found in the first step of the query:
`%LET TOPNN = 10 ; /* <----- change if desired */`
- If the DAYS timespan IDMSUA file is not active, remove the DERIVATION task for SUAPGM and run the query against the DETAIL timespan, Cycles 01 and 02
- To produce charts for multiple dates:
 1. Uncomment the SAS statement in the USER CODE.
 2. Select the Yesterday task in the IDMSSY File Selection step.
 3. Input multiple file cycles.
- Use filter statements in the IDMSUA File Selection step to restrict data to particular time ranges (e.g., HOUR GE 8 AND HOUR LE 17) or particular time zones (e.g., ZONE EQ 1),
- Use the extensive Chart View Edit capabilities to modify Y axis value ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

IDMTYP – IDMS Daily Transaction Type Analysis



Overview

The IDMTYP query produces two CSV output files.

The first CSV output file produces several daily charts that display the hourly values of several resource and performance-related measurements for the various types of transactions executed by IDMS regions. The IDMS transaction type is stored in the MICS data element IDMTYPE.

The IDMTYPEs and their meanings are:

O	IDMS/DC Online
B	Batch external run units
C	CICS external run units
I	IDMS/DC/UCF external run units
S	IDMS/DC system task

The transaction type is determined from the raw data transaction records, and is included as a summarization key in the CA MICS database file that is used to generate these charts.

The second CSV output file generates the charts that show the daily transaction response time distribution for each IDMS transaction type. The charts from this second CSV rely on the transaction response time classifications (for example, SHORT, MEDIUM, LONG), and response time distribution buckets that are defined using CA MICS-provided user interfaces. The IDMS Transaction Type Exit in prefix.MISC.PARMS(IDMLRT) is used to assign a response time classification (for example, SHORT, MEDIUM, LONG) for each transaction. Response time bucket distribution values are specified with the RESP keyword in the IDMS Processing Options member, prefix.MICS.PARMS(IDMOPS). The value of these charts is enhanced significantly when these members are updated to reflect your IDMS environment.

Two CSV output files are generated by this query:

- Daily IDMS Region Tran Type by HOUR
- Daily IDMS Region Tran Type Resp Dist

The Daily IDMS Region Tran Type by HOUR data extract generates the following six chart views:

#1 - Daily IDMS Region Transaction Type: Hourly CPU Time Use

This view uses stacked vertical bars to show the hourly CPU time used by the transaction type displayed in the chart title area. A horizontal line shows the hourly transaction count.

The first chart sample (previously shown) is an example of the charts produced by view #1. This chart was produced from two cycles of the DAYS timespan IDMSSY file and shows:

- The hourly CPU time used and transaction count for IDMS transaction type 'CICS External Run Units' executed by IDMS Region CV44 running on z/OS LPAR XE61 on an IBM 2817-615 (z196 CPC with 15 sub-capacity CP engines).
- This IDMS transaction type began executing during HOUR 4:00 AM, and was heavily used until about 10:00 PM.
- The CPU time that was used by this transaction type was evenly split between standard CP and zIIP processors.

#2 - Daily IDMS Region Transaction Type: Hourly Response Time Analysis

This view uses stacked vertical bars to show the hourly number of SHORT, MEDIUM, and LONG transactions executed for the transaction type. The average SHORT and MEDIUM transaction response times are displayed using horizontal lines.

#3 - Daily IDMS Region Transaction Type: Hourly Database Activity

This view uses stacked vertical bars to show the hourly count of database calls, and pages requested, read, and written for the transaction type. A horizontal line shows the hourly transaction count.

#4 - Daily IDMS Region Transaction Type: Hourly Queue Activity

This view uses stacked vertical bars to show the hourly count of queue DELETES, GETS, and PUTS for the transaction type. A horizontal line shows the hourly transaction count.

#5 - Daily IDMS Region Transaction Type: Hourly Storage Pool Use

This view uses stacked vertical bars to show the hourly count of storage FREES and GETS for the transaction type. A horizontal line shows the hourly transaction count.

#6 - Daily IDMS Region Transaction Type: Hourly Scratch Activity

This view uses stacked vertical bars to show the hourly quantity of scratch DELETES, GETS, and PUTS for the transaction type. A horizontal line shows the hourly transaction count.

The Daily IDMS Region Resp Dist data extract generates a single chart view:

#1 - Daily IDMS Transaction Type Response Time Distributions

This view uses vertical stacked bars to display the count of SHORT, MEDIUM, and LONG transactions that completed in each of the eight user-defined response time distribution ranges for each transaction type where the transactions have been categorized.

Note: The query does not produce charts for transaction types where the CA MICS IDMS Transaction Type exit code did not assign response time categorizations.

The second chart sample (previously shown) is an example of a response time distribution chart that is produced by this view. This chart sample shows the daily transaction response time distributions for the same IDMS region and transaction type that is shown in the first chart sample. The chart shows:

- The majority of SHORT transactions (over 1.5 million) completed in between 0.001 and 0.005 seconds.
- Approximately 50,000 transactions completed in less than 0.001 seconds.
- Virtually all of the MEDIUM transactions completed in between 0.1 and 1 seconds
- A few of the LONG transactions (just over 3000) completed in between 1 and 10 seconds.

Uses

This query provides an excellent way to monitor the daily transaction volume, response times, and resource usage of the various IDMS transaction types that are run on your IDMS regions.

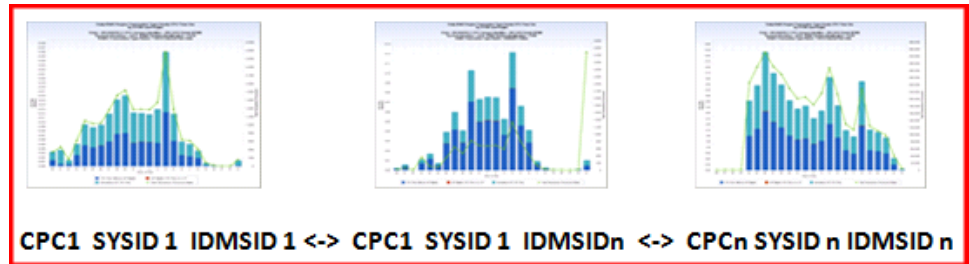
Execute this transaction on a daily basis to detect abnormal transaction volume or resource usage spikes (or valleys) by regularly viewing the charts generated by the first CSV data extract.

The second CSV data extract presents an ideal means to help you fine-tune both the response time values (specified with the RESP statement in prefix.MICS.PARMS(IDMOPS), and the transaction classification code (for example, SHORT, MEDIUM, LONG) found in prefix.MICS.PARMS(IDMRLRT). Once these user specifications are tuned to reflect the behavior of your IDMS transactions, the response time distribution charts can be used to help you validate tuning efforts and to provide the transaction response time information you require to set realistic service level agreements.

Data Extracts

DATA EXTRACT: Daily IDMS Region Tran Type by HOUR

VIEW 1 of 6: Daily IDMS Region Transaction Type: Hourly CPU Time Use



Transaction Type 1

Transaction Type n

Transaction Type n

Left Y axis

SSYXCSTM – CPU Time Without zIIP Eligible

SSYSUCTM – zIIP Eligible CPU Time on a CP

SSYSPNTM – Normalized zIIP CPU Time

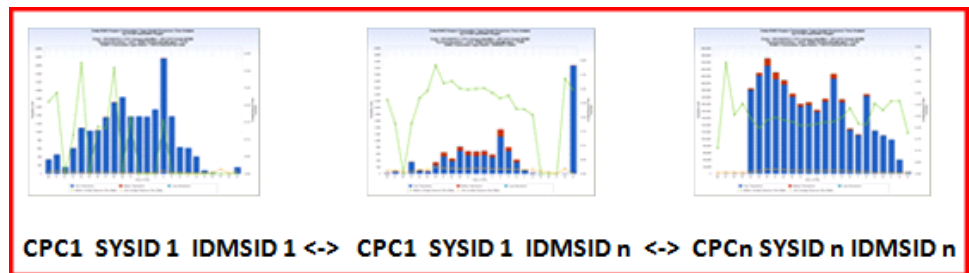
Right Y axis

SSYTTRAN – Total Transactions Processed

X axis

HOUR – Hour of Day

VIEW 2 of 6: Daily IDMS Region Transaction Type: Hourly Response Time Analysis



Transaction Type 1

Transaction Type n

Transaction Type n

Left Y axis

SSYSTRAN – Short Transactions

SSYMTRAN – Medium Transactions

SSYLTRAN – Long Transactions

Right Y axis

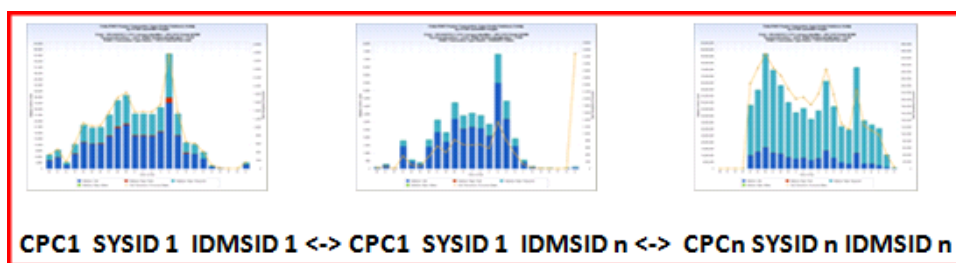
SSYAVSTM – Short Average Response Time

SSYAVMTM – Medium Average Response Time

X axis

HOUR – Hour of Day

VIEW 3 of 6: Daily IDMS Region Transaction Type: Hourly Database Activity



Transaction Type 1

Transaction Type n

Transaction Type n

Left Y axis

SSYDBCLS – Database Calls

SSYPREAD – Database Pages Read

SSYPREQ – Database Pages Requested

SSYPWRIT – Database Pages Written

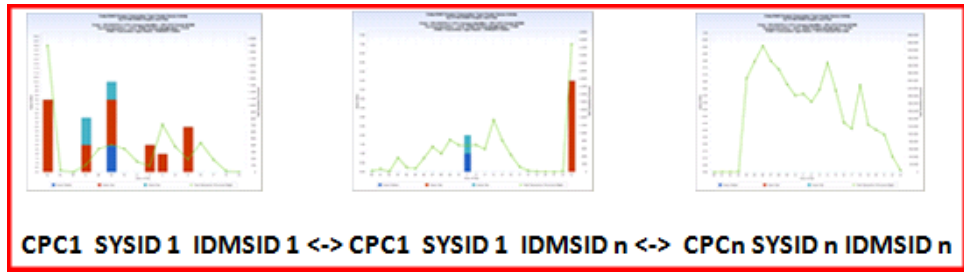
Right Y axis

SSYTTRAN – Total Transactions Processed

X axis

HOUR – Hour of Day

VIEW 4 of 6: Daily IDMS Region Transaction Type: Hourly Queue Activity



Transaction Type 1

Transaction Type n

Transaction Type n

Left Y axis

SSYQDEL – Queue Deletes

SSYQGETL – Queue Gets

SSYQPUT – Queue Puts

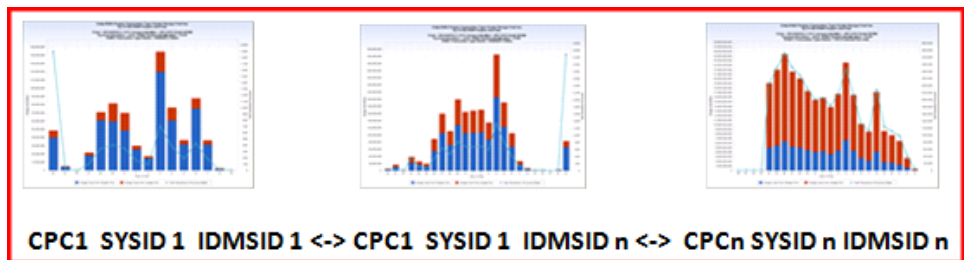
Right Y axis

SSYTTRAN – Total Transactions Processed

X axis

HOUR – Hour of Day

VIEW 5 of 6: Daily IDMS Region Transaction Type: Storage Pool Use



Transaction Type 1

Transaction Type n

Transaction Type n

Left Y axis

SSYUSDP – Storage Used from Program Pool

SSYUSDS – Storage Used from Storage Pool

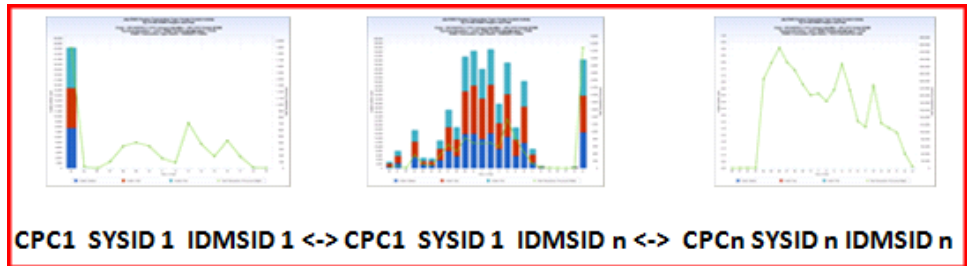
Right Y axis

SSYTTRAN – Total Transactions Processed

X axis

HOUR – Hour of Day

VIEW 6 of 6: Daily IDMS Region Transaction Type: Hourly Scratch Activity



Transaction Type 1

Transaction Type n

Transaction Type n

Left Y axis

SSYXDEL – Scratch Deletes

SSYXGET – Scratch Gets

SSYXPUT – Scratch Puts

Right Y axis

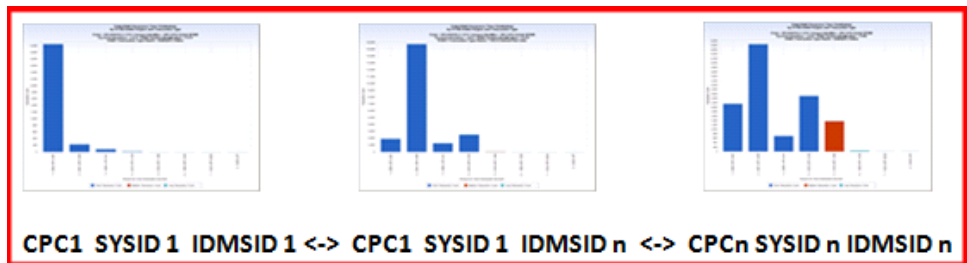
SSYTTRAN – Total Transactions Processed

X axis

HOUR – Hour of Day

DATA EXTRACT: Daily IDMS Region Tran Type Resp Dist

VIEW 1 of 1: Daily IDMS Transaction Type Response Time Distributions



Transaction Type 1

Transaction Type n

Transaction Type n

Left Y axis

SHORT – Short Transaction Count

MEDIUM – Medium Transaction Count

LONG – Long Transaction Count

X axis

BUCKET – Response Time Distribution Buckets

Query Defaults

- HARCPU DAYS timespan file: Cycle 01
- IDMSY DAYS timespan file: Cycles 01 and 02
- A User Code task selects data from yesterday only.

- Derivations:

DATE

DATEPART(ENDTS)

CPCUID

CPCUID = PUT(SYSID,\$CPCFMT.) ;

SHORT

SSYSDSTn – where n corresponds to the Bucket value (1-8)

MEDIUM

SSYMDSTn – where n corresponds to the Bucket value (1-8)

LONG

SSYLDSTn – where n corresponds to the Bucket value (1-8)

BUCKET

Text string constructed in query user step using the response distribution values found in data elements SSYRVAL1 – SSYRVAL7.

IDMTYPED

IDMTYPED = PUT(IDMTYPE,\$IDMTYPE.)

Modifications

You can make the following modifications to the IDMTYP query:

- Add filter statements to the IDMSY File Selection step to select specific SYSID values or IDMS Regions (IDMSID)
- Produce charts for multiple dates.

You can uncomment the SAS statement in the USER CODE-Select Yesterday task in the IDMSY File Selection step and input multiple file cycles.

- Use filter statements in the IDMSY File Selection step to restrict data to particular time ranges (for example, HOUR GE 8 AND HOUR LE 17) or particular time zones (for example, ZONE EQ 1),
- Use the extensive Chart View Edit capabilities to modify Y axis value ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

Chapter 5: CA MICS Analyzer Option for MQSeries (MQS) Queries

This section contains the following topics:

[MQSBPA – MQ Daily Buffer Pool Analysis](#) (see page 82)

[MQSCFS – MQ Daily Coupling Facility Structure Use](#) (see page 86)

[MQSDB2 – MQ Daily DB2 Shared Queue Analysis](#) (see page 90)

[MQSDMA – MQ Daily Data Manager Analysis](#) (see page 92)

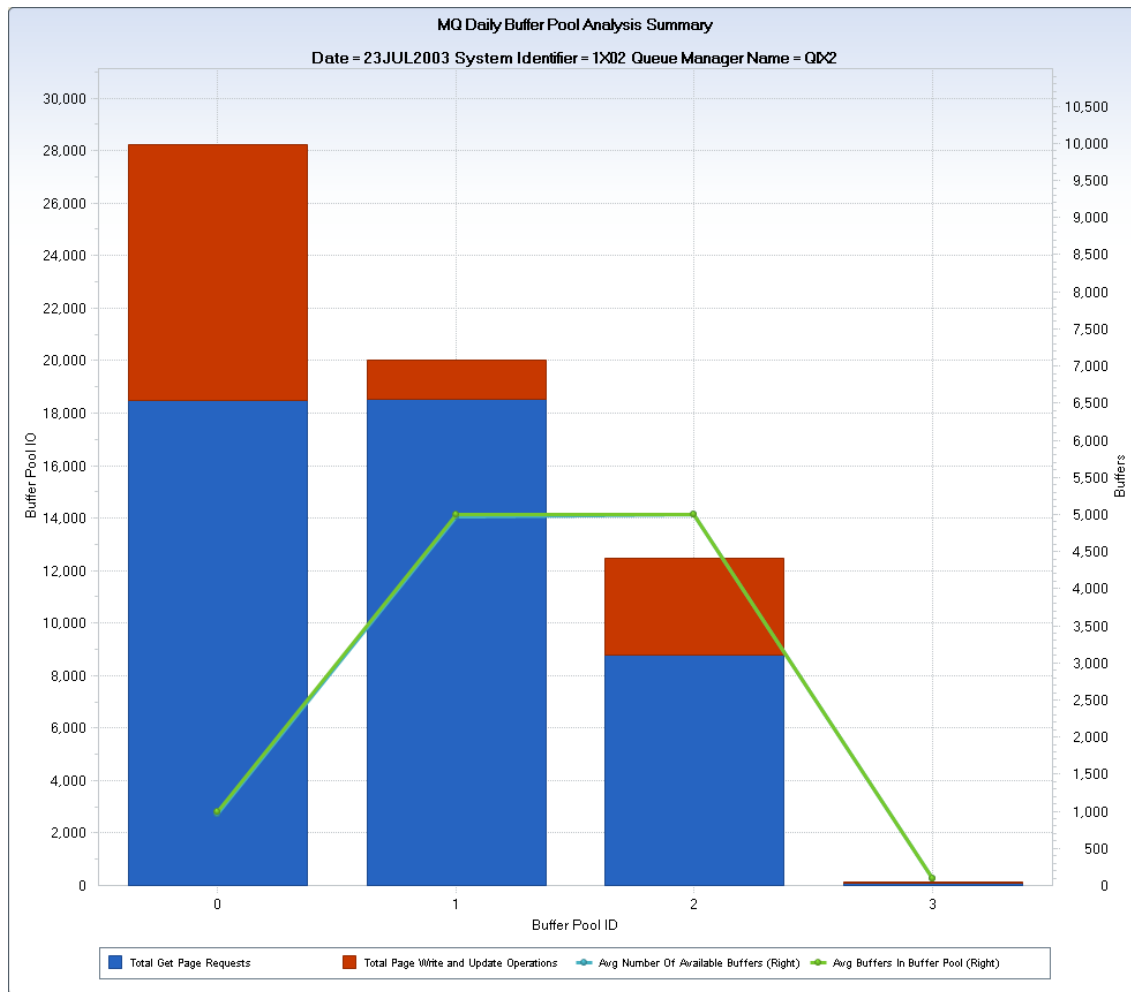
[MQSLMA – MQ Daily Log Manager Analysis](#) (see page 95)

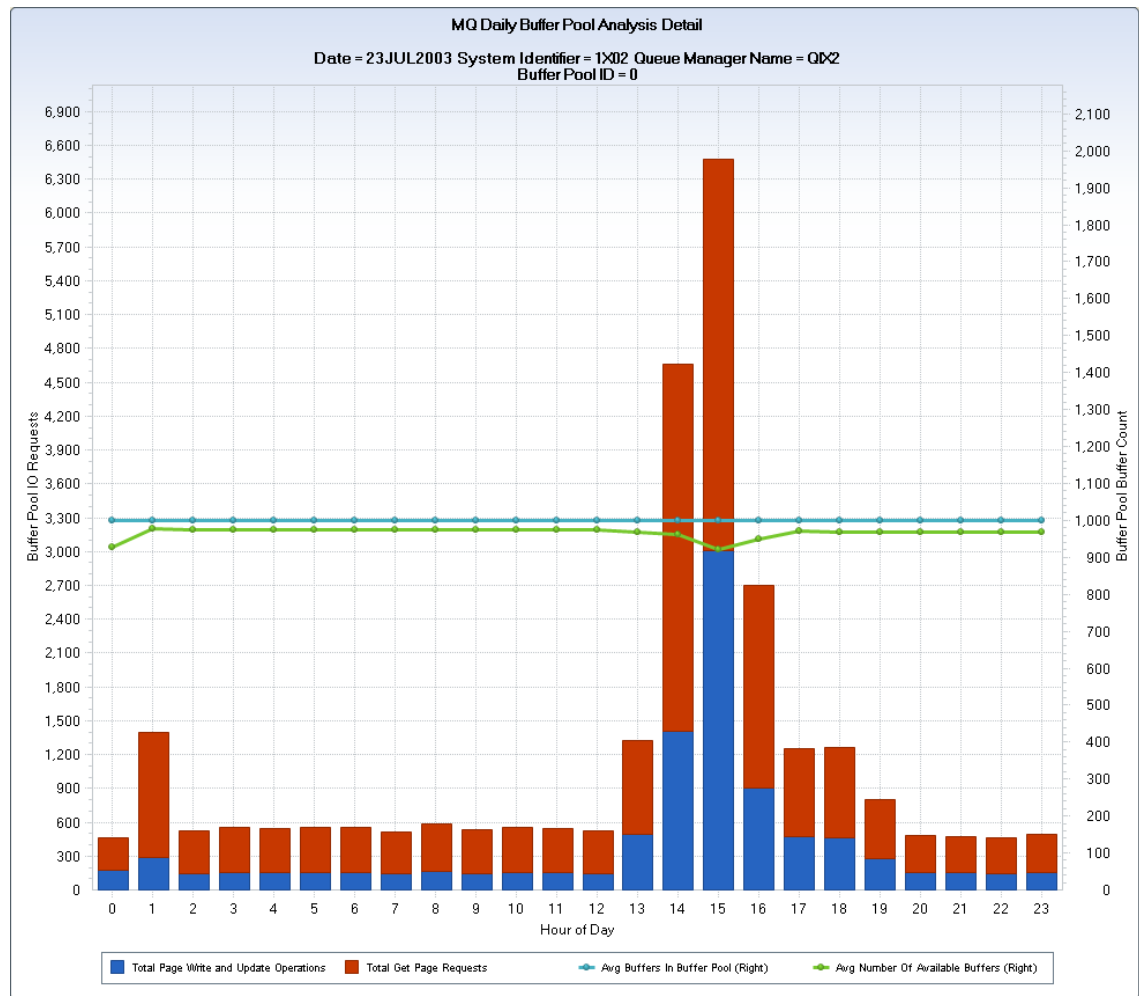
[MQSMMA – MQ Daily Message Manager Analysis](#) (see page 97)

[MQST10 – MQ Daily Top 10 Queues](#) (see page 99)

[MQSTSA – MQ Daily Task Suspend Call Analysis](#) (see page 102)

MQSBPA – MQ Daily Buffer Pool Analysis





Overview

The MQSBPA query produces two views of WebSphere MQ buffer pool activity and utilization. The summary view produces a chart for each queue manager, showing total daily activity and use of each active buffer pool. The detail view produces a chart for each buffer pool showing activity and use for each hour of the day. The query code eliminates buffer pools that have no activity.

The first example shows the summary chart generated for queue manager QIX2 running on SYSID 1X02. Four active buffer pools exist, with buffer pool 0 showing the greatest activity.

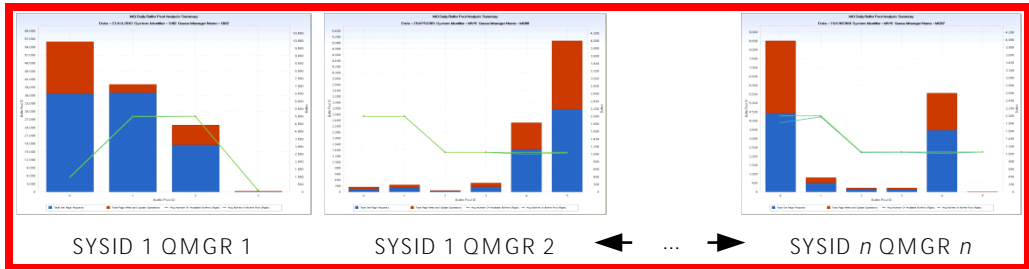
The second example shows hourly statistics for buffer pool 0 of the same queue manager. Hours 14 and 15 were the busiest, and the number of available buffer pools (green line) dipped slightly during these two hours.

Uses

Buffer pool allocation and management is key to a successful WebSphere MQ implementation. Message traffic handled by in-storage buffer pools provides the best response time and throughput. This query provides an excellent view of the use and availability of each buffer pool. Monitor the buffer pools serving business critical applications carefully to help ensure that sufficient buffers are available to handle workload, and to examine hourly usage patterns.

Data Extracts

MQ Daily Buffer Pool Analysis Summary



Left Y-axis

BMSTGETS – Total Get Page Requests

BMSWUO – Total Page Write and Update Operations

Right Y-axis

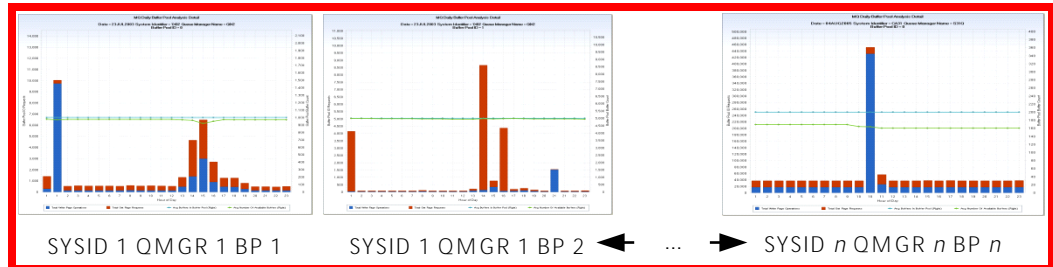
BMSAVBUF – Average Buffers in Buffer Pool

BMSAVVB – Average Number of Available Buffers

X-axis

BMSPool – Buffer Pool ID

MQ Daily Buffer Pool Analysis Detail



Left Y-axis

BMSTGETS – Total Get Page Requests

BMSWUO – Total Page Write and Update Operations

Right Y-axis

BMSAVBUF – Average Buffers in Buffer Pool

BMSAVAVB – Average Number of Available Buffers

X-axis

HOUR – Hour of Day

Query Defaults

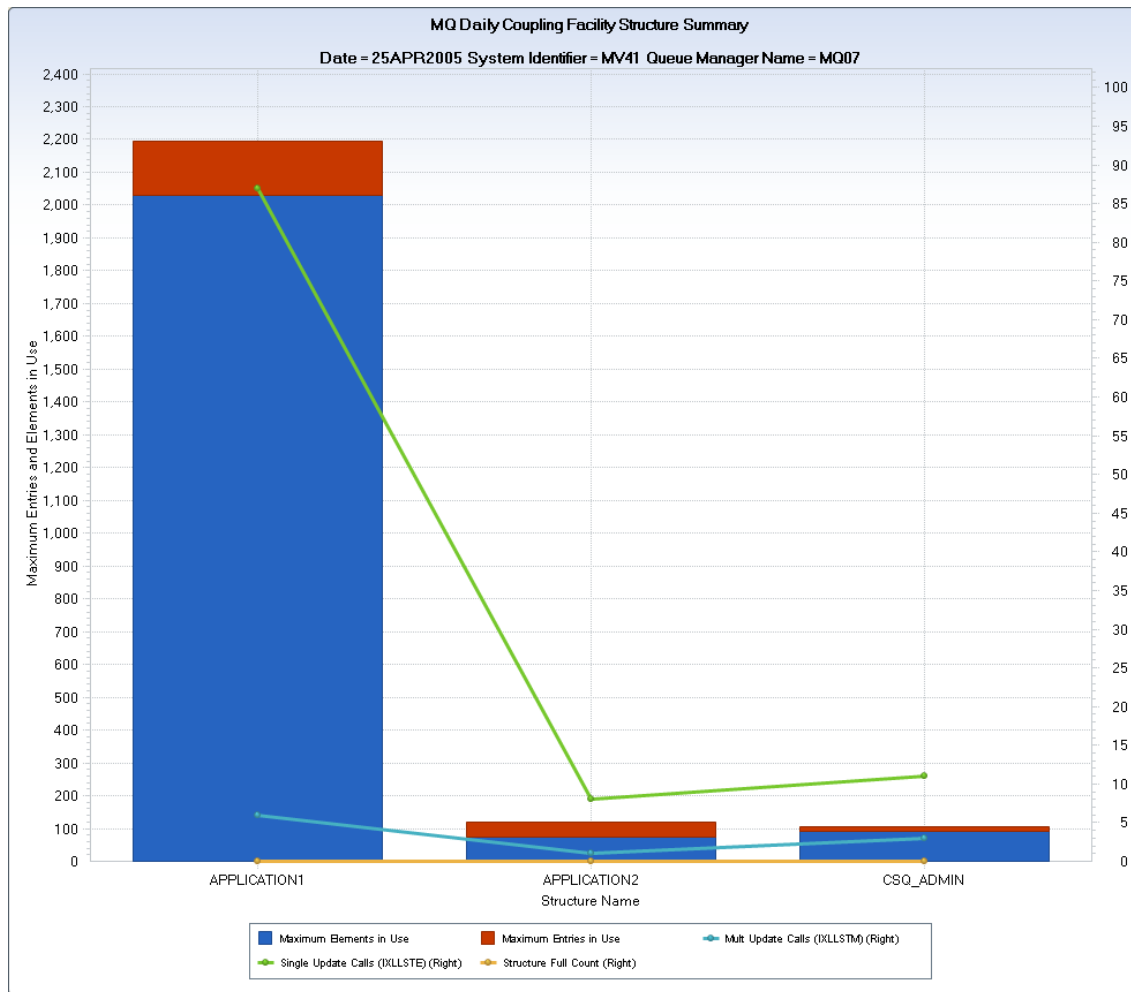
- MQSBMS DAYS timespan file: Cycles 01 and 02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
DATE
DATEPART(ENDTS)
BMSTGETS
SUM(BMSGETN, BMSGETP, BMSSTL);

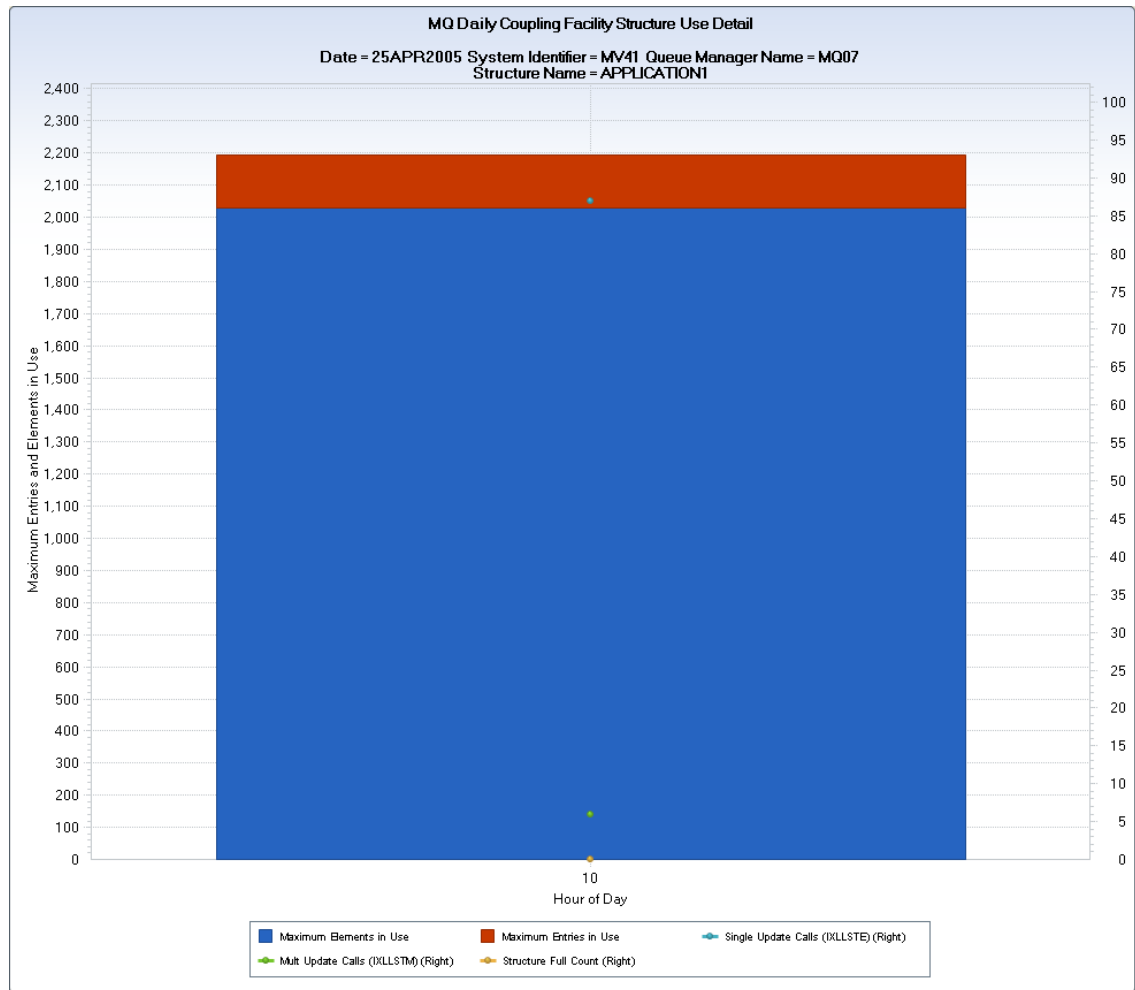
Modifications

The following modifications can be made to the MQSBMS query:

- You can add filter statements to select only a single SYSID or Queue Manager (MQMSSI).
- If you do not have the DAYS timespan MQSBMS file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

MQSCFS – MQ Daily Coupling Facility Structure Use





Overview

The MQSCFA query produces two views of WebSphere MQ Coupling Facility (CF) activity and utilization. The summary view produces a daily chart for each queue manager. The chart shows maximum entries and elements in use and various activity metrics for each MQ structure in the Coupling Facility. The detail view produces a daily chart for each structure showing the same metrics for each hour of the day.

The examples were produced from a limited data sample of one hour. The top example shows the summary chart generated for queue manager MQ07 running on SYSID MV451. Four Coupling Facility structures exist, with the structure named APPLICATION1 showing the greatest activity.

The second example shows hourly statistics for structure APPLICATION1. The data used to produce this chart only had one hour, HOUR 10, so the usage totals match the summary view. The structure had 87 single update calls 6 multiple update calls.

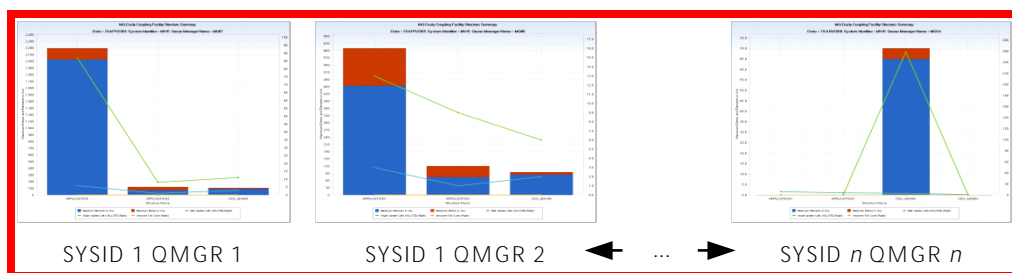
Uses

WebSphere MQ uses the SYSPLEX Coupling Facility to facilitate shared message queues. Multiple queue managers running on different systems in the SYSPLEX can process the messages arriving in the Coupling Facility structures. Applications using shared message queues benefit from load balancing across the SYSPLEX. This query can be used to monitor structure use—a key metric shown is the Structure Full Count. If this metric is ever greater than 0, the size of the structure may need to be increased.

This query is useful to identify the most used queues by applications.

Data Extracts

MQ Daily CF Structure Use Summary



Left Y-axis

CMSMXELE – Maximum Elements in Use

CMSMXENT – Maximum Entries in Use

Right Y-axis

CMSMSELE – Single Update Calls (IXLLSTE)

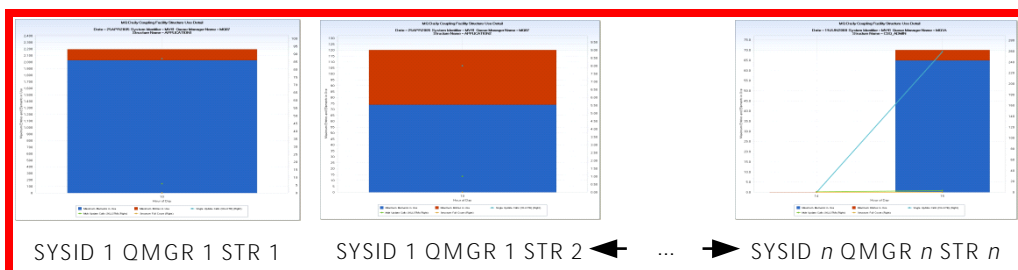
CMSMSENT – Multiple Update Calls (IXLLSTM)

CMSSTFUL – Structure Full Count

X-axis

CMSSTRNM – Structure Name

MQ Daily CF Structure Use Detail



Left Y-axis

 CMSMXELE – Maximum Elements in Use

 CMSMXENT – Maximum Entries in Use

Right Y-axis

 CMSCSEC – Single Update Calls (IXLLSTE)

 CMSCMEC – Multiple Update Calls (IXLLSTM)

 CMSSTFUL – Structure Full Count

X-axis

 HOUR – Hour of Day

Query Defaults

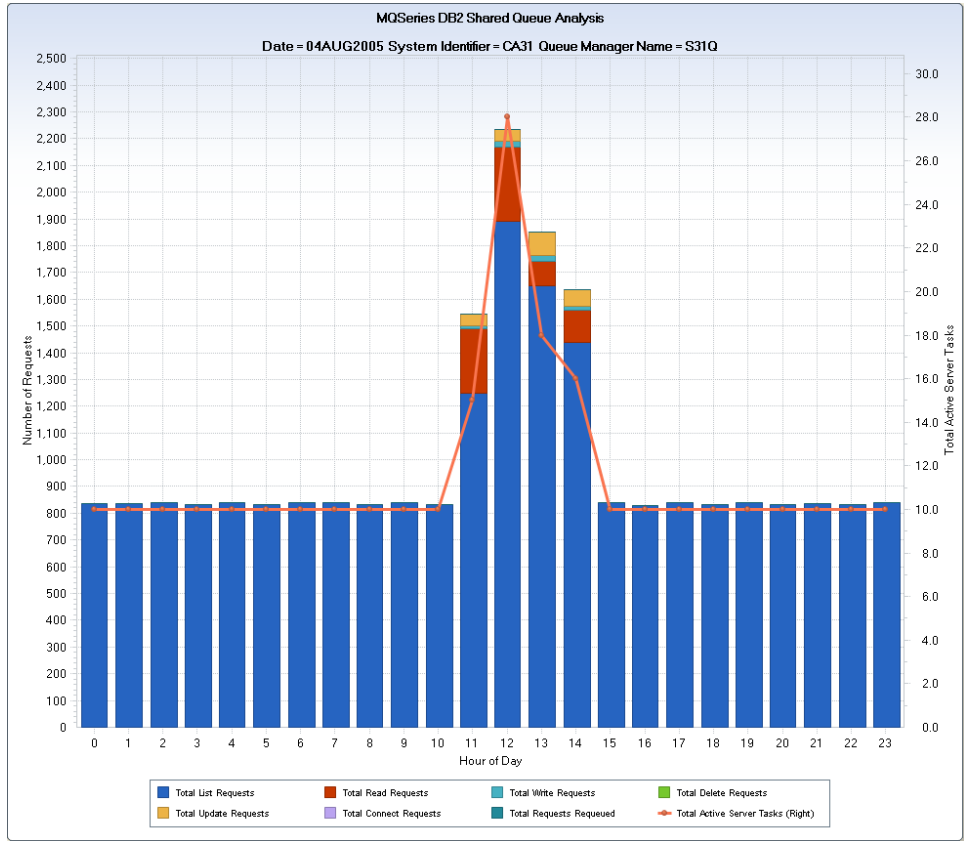
- MQSCMS DAYS timespan file: Cycles 01 and 02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART(ENDTS)

Modifications

The following modifications can be made to the MQSCFS query:

- You can add filter statements to select only a single SYSID or Queue Manager (MQMSSI).
- If you do not have the DAYS timespan MQSCMS file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

MQSDB2 – MQ Daily DB2 Shared Queue Analysis



Overview

The MQSDB2 query produces a daily chart showing the hourly request activity for the WebSphere MQ DB2 manager. The chart shows the number of requests of each type, using stacked bars, and the total number of active tasks using a horizontal line.

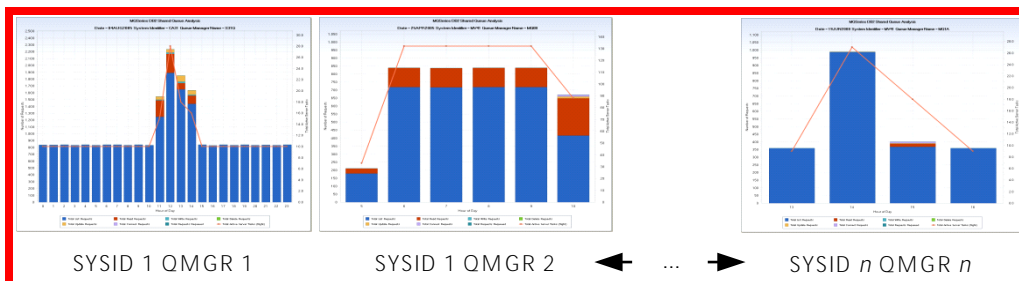
The example shows the daily chart generated for queue manager S31Q running on SYSID CA31. Only LIST requests occurred for hours 0-10, and 15-23. For hours 11-14, there were a number of READ, WRITE, and UPDATE requests.

Uses

WebSphere MQ uses an internal DB2 database to manage the object definitions and other information associated with shared message queues. This report helps you see how much activity and resources are used by this internal MQ subsystem to manage shared queues in a SYSPLEX environment.

Data Extracts

MQ Daily DB2 Shared Queue Analysis



Left Y-axis

- DBSLIST – Total List Requests
- DBSREAD – Total Read Requests
- DBSWRITE – Total Write Requests
- DBSDEL – Total Delete Requests
- DBSSUPD – Total Update Read Requests
- DBSCONN – Total Connect Requests
- DBSREQUE – Total Requests Requested

Right Y-axis

- DBSATASK – Total Active Server Tasks

X-axis

- HOUR – Hour of Day

Query Defaults

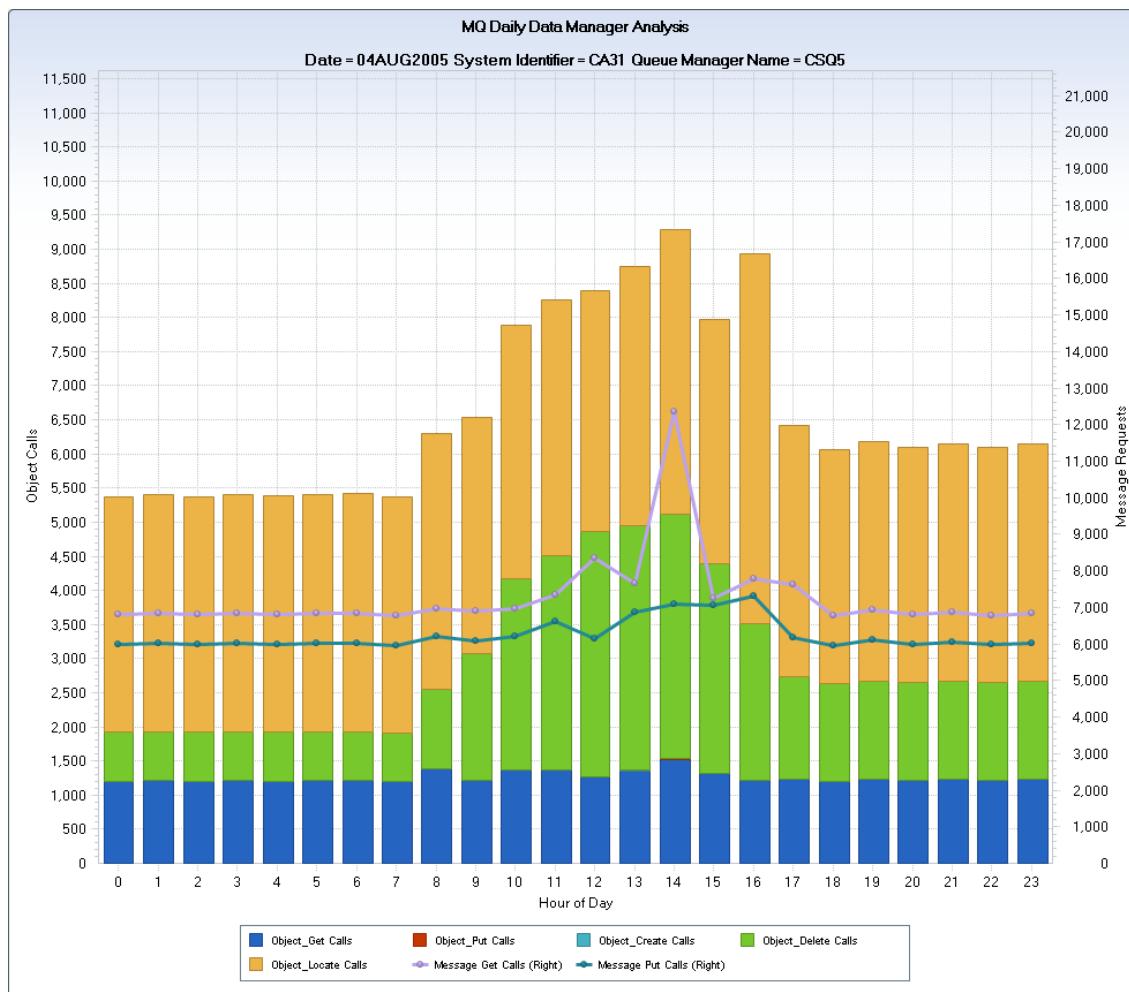
- MQSDBS DAYS timespan file: Cycles 01 and 02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART (ENDTS)

Modifications

The following modifications can be made to the MQSDB2 query:

- You can add filter statements to select only a single SYSID or Queue Manager (MQMSSI).
- If you do not have the DAYS timespan MQSDBS file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

MQSDMA – MQ Daily Data Manager Analysis



Overview

The MQSDMA query produces a daily chart showing the hourly activity handled by the WebSphere MQ Data manager. The chart shows the number of object call requests of each type, using stacked bars, and the number of message get and put calls using horizontal lines.

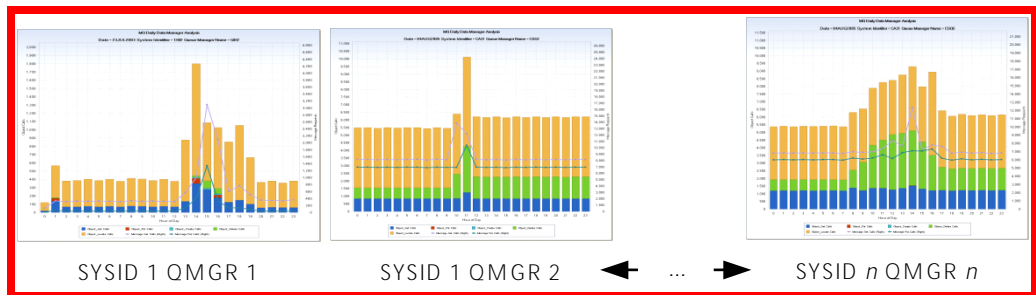
The example shows the daily chart generated for queue manager CSQ5 running on SYSID CA31. HOUR 14 was the peak activity hour with 12,357 message GET requests.

Uses

The WebSphere MQ data manager handles the links between messages and queues. This chart provides an excellent overview of the daily message traffic for each queue manager.

Data Extracts

MQSeries Daily Data Manager Analysis



Left Y-axis

- DMSDGET – Object_Get Calls
- DMSDPUT – Object_Put Calls
- DMSDCRE – Object_Create Calls
- DMSDDEL – Object_Delete Calls
- DMSDLOC – Object_Locate Calls

Right Y-axis

- DMSMGET – Message Get Calls
- DMSMPUT – Message Put Calls

X-axis

Hour of Day

Query Defaults

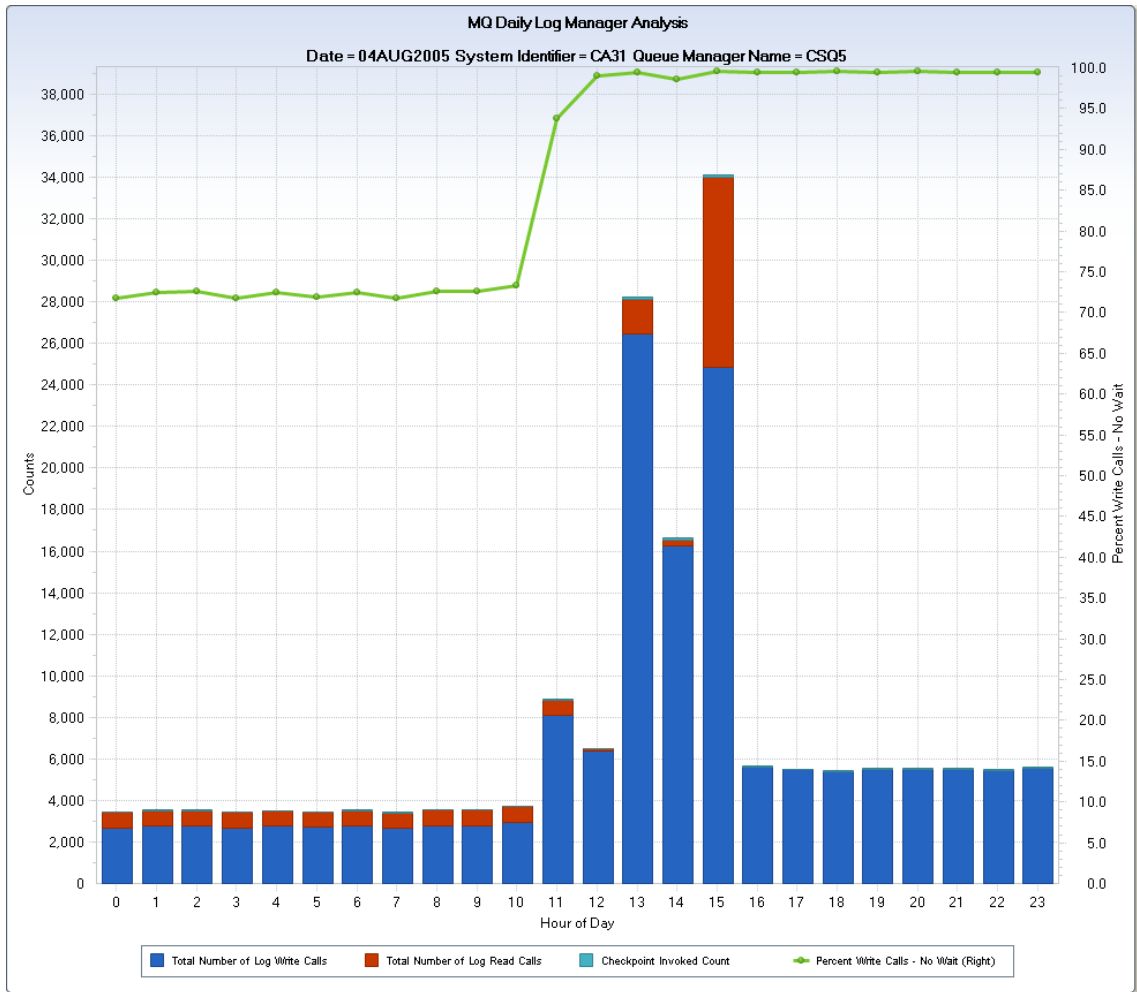
- MQSDMS DAYS timespan file: Cycles 01 and 02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART (ENDTS)

Modifications

The following modifications can be made to the MQSDB2 query:

- You can add filter statements to select only a single SYSID or Queue Manager (MQMSSI).
- If you do not have the DAYS timespan MQSDMS file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

MQSLMA – MQ Daily Log Manager Analysis



Overview

The MQSLMA query produces a daily chart showing the hourly activity of the WebSphere MQ Log manager. The chart shows the number of log reads, writes, and checkpoints taken using stacked bars, and percentage of undelayed log writes using a horizontal line.

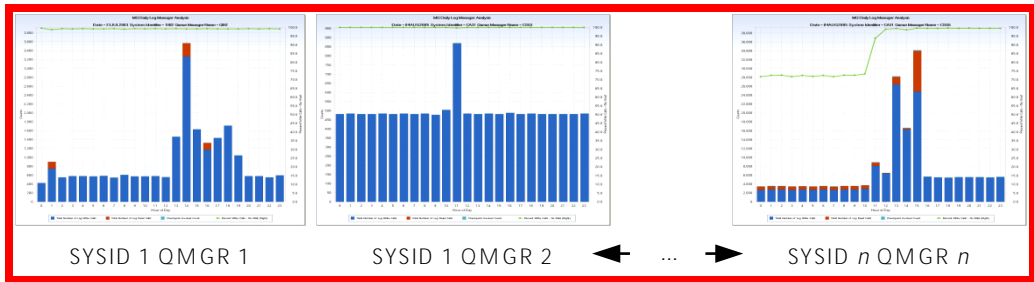
The example shows the daily chart generated for queue manager CSQ5 running on SYSID CA31. About 78 percent of log writes were delayed for HOURS 0-10. Hour 13 had the greatest number of log writes, and hour 15 had the greatest number of log reads.

Uses

The WebSphere MQ log manager helps ensure that important messages are backed up on DASD storage. While necessary for backup and recovery, message logging can be one of the biggest factors contributing to overall MQ message elapsed time. This query provides a good overview of the logging activity for each Queue Manager. Try to keep the percentage of no-wait log writes near 100 percent.

Data Extracts

MQ Daily Log Manager Analysis



Left Y-axis

LMSTNLWR – Total Number of Log Write Calls

LMSTNLRR – Total Number of Log Read Calls

LMSLLCP – Checkpoint Invoked Count

Right Y-axis

LMSPCWRN – Percent Write Calls - No Wait

X-axis

Hour of Day

Query Defaults

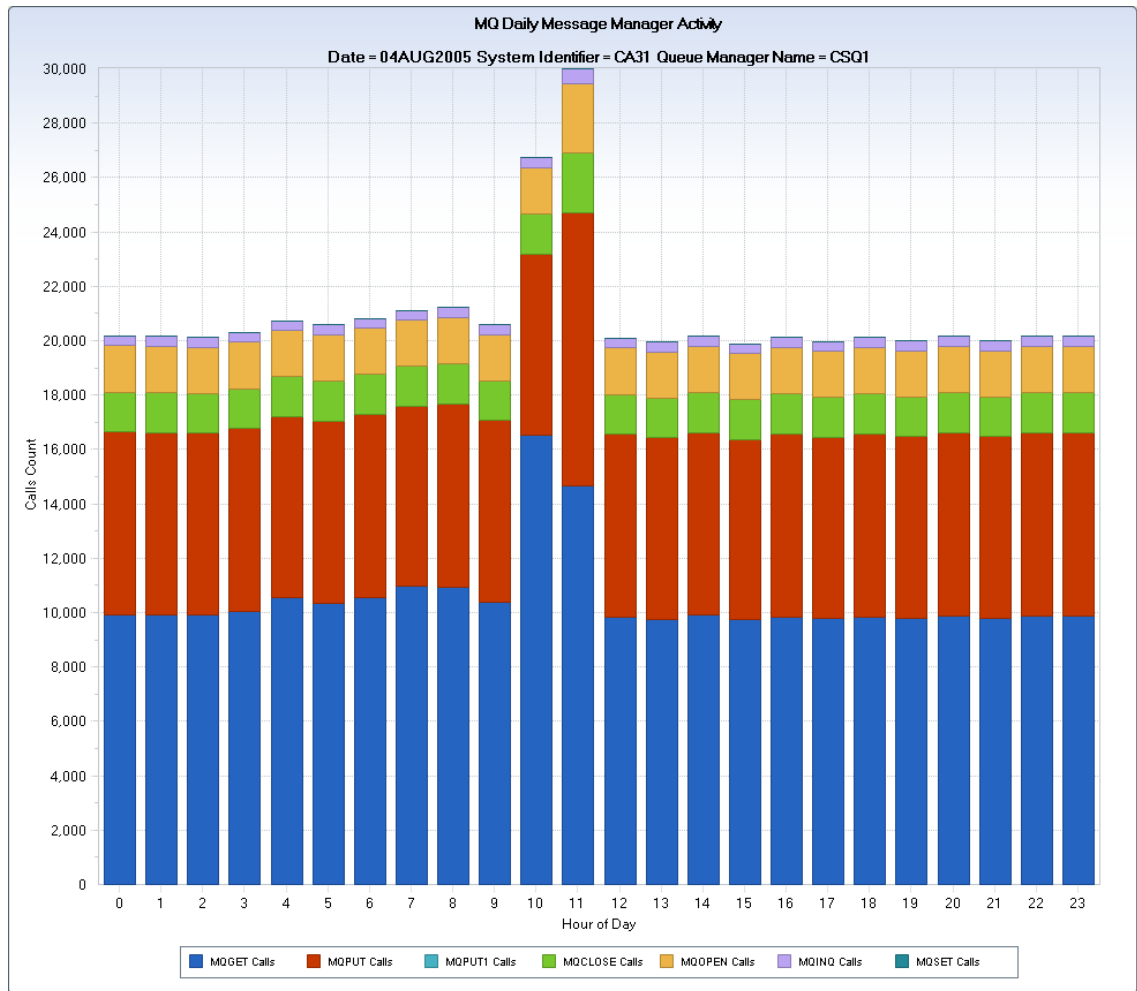
- MQSLMS DAYS timespan file: Cycles 01 and 02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART (ENDTS)

Modifications

The following modifications can be made to the MQSLMA query:

- You can add filter statements to select only a single SYSID or Queue Manager (MQMSSI).
- If you do not have the DAYS timespan MQSLMS file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

MQSMMA – MQ Daily Message Manager Analysis



Overview

The MQSMMMA query produces a daily chart showing the hourly activity handled by the WebSphere MQ Message manager. The chart shows the number of queue calls of each type, using stacked bars, made by the various applications using MQ.

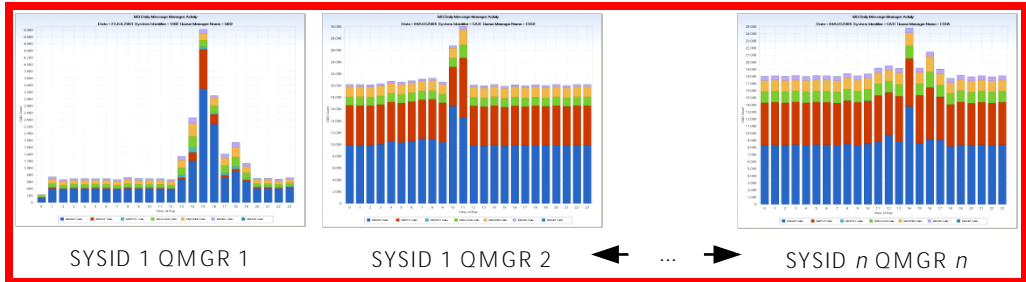
The example shows the daily chart generated for queue manager CSQ1 running on SYSID CA31. HOUR 10 was the peak activity hour for GETS, and HOUR 11 was the peak hour for PUTS.

Uses

This query provides an excellent way to view the overall activity of each WebSphere MQ data manager. Peak periods and the variety of message call activities can be easily visualized.

Data Extracts

MQ Daily Message Manager Analysis



Left Y-axis

- MMSGET – MQGET Calls
- MMSPUT – MQPUT Calls
- MMSPUT1 – MQPUT1 Calls
- MMSCLOS – MQCLOSE Calls
- MMSOPEN – MQOPEN Calls
- MMSINQ – MQINQ Calls
- MMSSET – MQSET Calls

X-axis

Hour of Day

Query Defaults

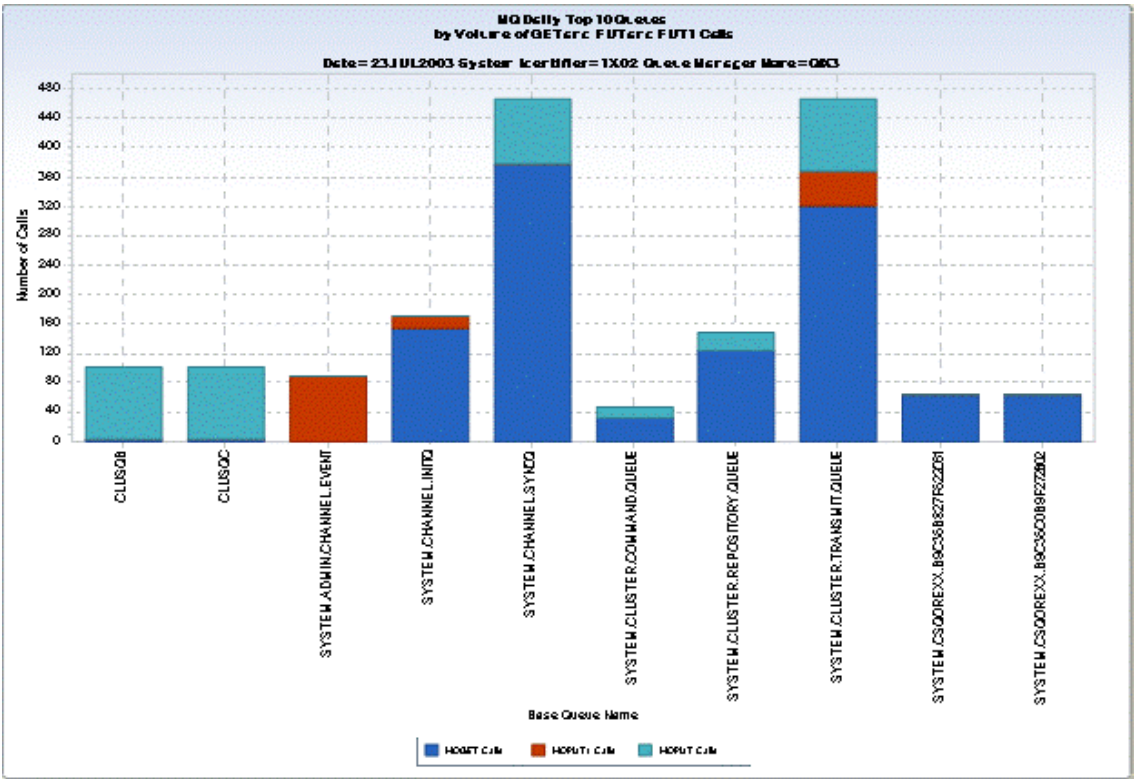
- MQSMMS DAYS timespan file: Cycles 01 and 02
- User code selects yesterday’s data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
DATE
DATEPART (ENDTS)

Modifications

The following modifications can be made to the MQSMMA query:

- You can add filter statements to select only a single SYSID or Queue Manager (MQMSSI).
- If you do not have the DAYS timespan MQSMMS file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

MQST10 – MQ Daily Top 10 Queues



Overview

The MQST10 query produces a daily chart for each WebSphere MQ queue manager. The chart shows the top *nn* (default 10) queues by volume of GET, PUT, and PUT1 calls using vertical stacked bars.

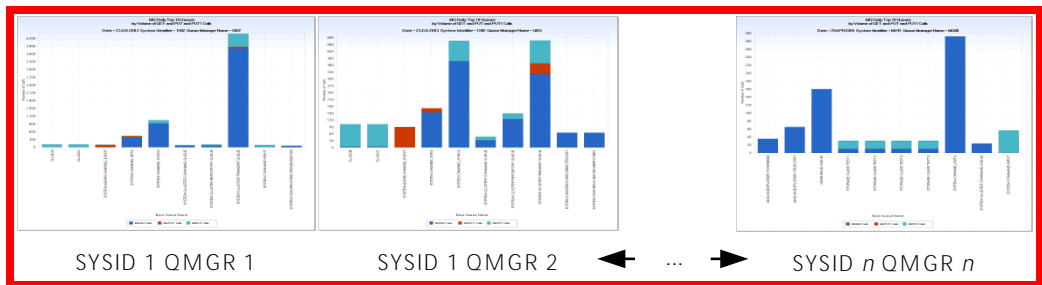
The example shows the daily chart generated for queue manager QIX3 running on SYSID 1X02. HOUR 14 had the greatest number of task call suspensions. The SYSTEM.CLUSTER.TRANSMIT.QUEUE and SYSTEM.CHANNEL.SYNQC queues were the most heavily used.

Uses

This query provides insight into which business applications are the heaviest users of each WebSphere MQ queue manager based on message traffic to and from the queues associated with the applications.

Data Extracts

MQ Daily Top 10 Queues



Left Y-axis

- QAAGN – MQGET Calls
- QAAP1N – MQPUT1 Calls
- QAAPN – MQPUT Calls

X-axis

QAABASNM – Base Queue Name

Query Defaults

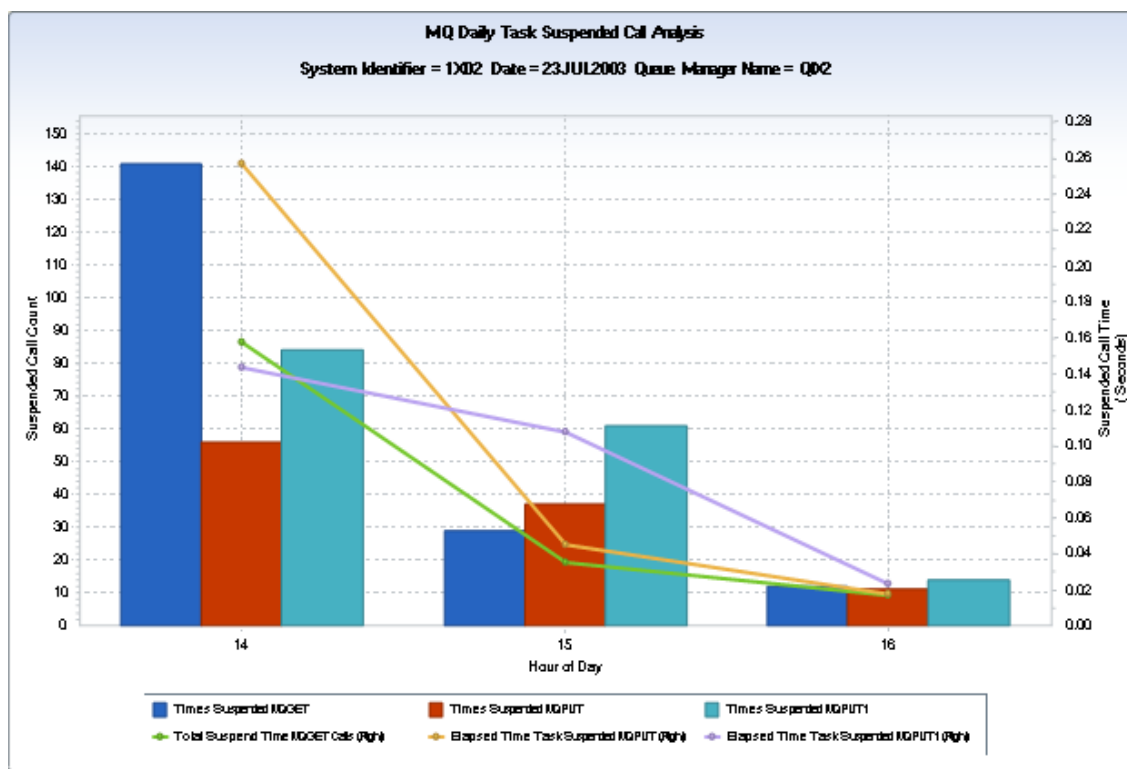
- MQAQAA DAYS timespan file: Cycle 01-02
- Derivations:
DATE
DATEPART (ENDTS)

Modifications

The following modifications can be made to the MQST10 query:

- You can add filter statements to select only a single SYSID or Queue Manager (MQMSSI).
- You can adjust the TOP nn value to something other than the default of 10.
- You can add filter statements to subset the queue selection to queues beginning with a particular character string (for example, QAABASNM =: 'AB').
- If you do not have the DAYS timespan MQATAA file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

MQSTSA – MQ Daily Task Suspend Call Analysis



MQ Daily Task Suspend Call Counts and Wait Times by Queue Manager Name								1
								10:35 Tuesday, February 15, 2011
----- System Identifier=CA31 Date=04AUG2005 -----								
Queue Manager Name	Wait Time	Total Suspend Time MQGET	Times Suspended MQGET	Elapsed Time Task Suspended MQPUT	Times Suspended MQPUT	Elapsed Time Task Suspended MQPUT1	Times Suspended MQPUT1	
CS05	0:00:00.0000	0:00:00.0000	0	0:00:02.2027	177	0:00:00.0000	0	
----- System Identifier=MV41 Date=25APR2005 -----								
Queue Manager Name	Wait Time	Total Suspend Time MQGET	Times Suspended MQGET	Elapsed Time Task Suspended MQPUT	Times Suspended MQPUT	Elapsed Time Task Suspended MQPUT1	Times Suspended MQPUT1	
MQ07	0:00:00.3218	0:00:18.1054	166	0:00:22.0877	232	0:00:00.0000	0	
MQ08	0:00:00.0540	0:00:05.5933	71	0:00:07.1841	111	0:00:00.0000	0	
----- System Identifier=MV41 Date=19JUN2008 -----								
Queue Manager Name	Wait Time	Total Suspend Time MQGET	Times Suspended MQGET	Elapsed Time Task Suspended MQPUT	Times Suspended MQPUT	Elapsed Time Task Suspended MQPUT1	Times Suspended MQPUT1	
MQ07	0:00:00.0000	0:00:00.0000	0	0:00:00.0000	0	0:00:00.0000	0	
MQ08	0:00:00.0000	0:00:00.0000	0	0:00:00.0000	0	0:00:00.0000	0	
MQ11	0:00:00.0405	0:00:00.0000	0	0:00:00.0000	0	0:00:00.0000	0	
MQ12	0:00:00.0000	0:00:00.0000	0	0:00:00.0000	0	0:00:00.0000	0	
----- System Identifier=1X02 Date=23JUL2003 -----								
Queue Manager Name	Wait Time	Total Suspend Time MQGET	Times Suspended MQGET	Elapsed Time Task Suspended MQPUT	Times Suspended MQPUT	Elapsed Time Task Suspended MQPUT1	Times Suspended MQPUT1	
QIX2	0:00:00.2029	0:00:00.2096	182	0:00:00.3198	104	0:00:00.2750	159	
QIX3	0:00:00.0938	0:00:00.1945	173	0:00:00.0884	57	0:00:00.2803	141	

Overview

The MQSTSA query produces a daily chart for each WebSphere MQ queue manager. The chart shows the number of task call suspensions (GET, PUT, and PUT) for each hour using vertical bars, and the elapsed time tasks were suspended for each call type using horizontal lines. The query also generates a tabular SASLIST report summarizing the charted information for each queue manager.

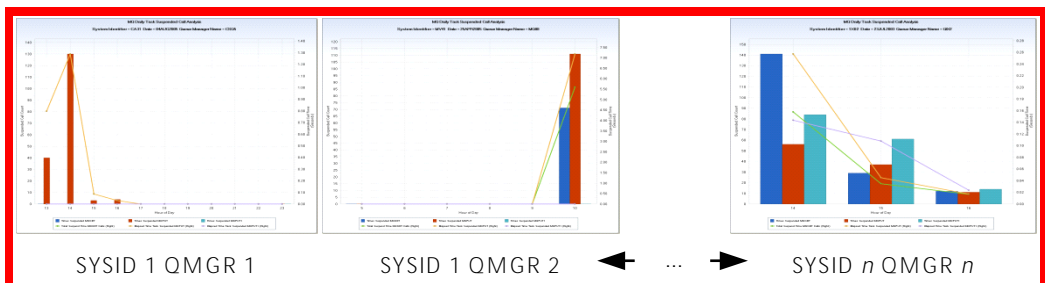
The example shows the daily chart generated for queue manager QIX2 running on SYSID 1X02. HOUR 14 had the greatest number of task call suspensions. This chart was generated from was the peak activity hour for GETS, and HOUR 11 was the peak hour for PUTS.

Uses

Task call suspensions can occur for various reasons, but the business impact is that they impede throughput. This query can be used to analyze call suspend patterns and volume. The MQATAA file can be processed to identify the specific tasks experiencing call suspensions and to determine root causes.

Data Extracts

MQ Daily Task Suspend Call Analysis



Left Y-axis

TAAGSUS – Times Suspended MQGET

TAAPSUS – Times Suspended MQPUT

TAAP1SUS – Times Suspended MQPUT1

Right Y-axis

TAAGSUSTM – Total Suspend Time MQGET Calls

TAAPSUSTM – Elapsed Time Task Suspend MQPUT

TAAP1STM – Elapsed Time Task Suspend MQPUT1

X-axis

Hour of Day

SASLIST

Report summarized by SYSID, Date, and Queue Manager Name. Reports Message IO wait time and suspend count statistics for the entire day.

Query Defaults

- MQATAA DAYS timespan file: Cycles 01 and 02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART (ENDTS)

Modifications

The following modifications can be made to the MQATSA query:

- You can add filter statements to select only a single SYSID or Queue Manager (MQMSSI).
- If you do not have the DAYS timespan MQATAA file active, you can run the query against the DETAIL timespan, Cycles 01 and 02.

Chapter 6: CA MICS Hardware and SCP Analyzer (RMF) Queries

This section contains the following topics:

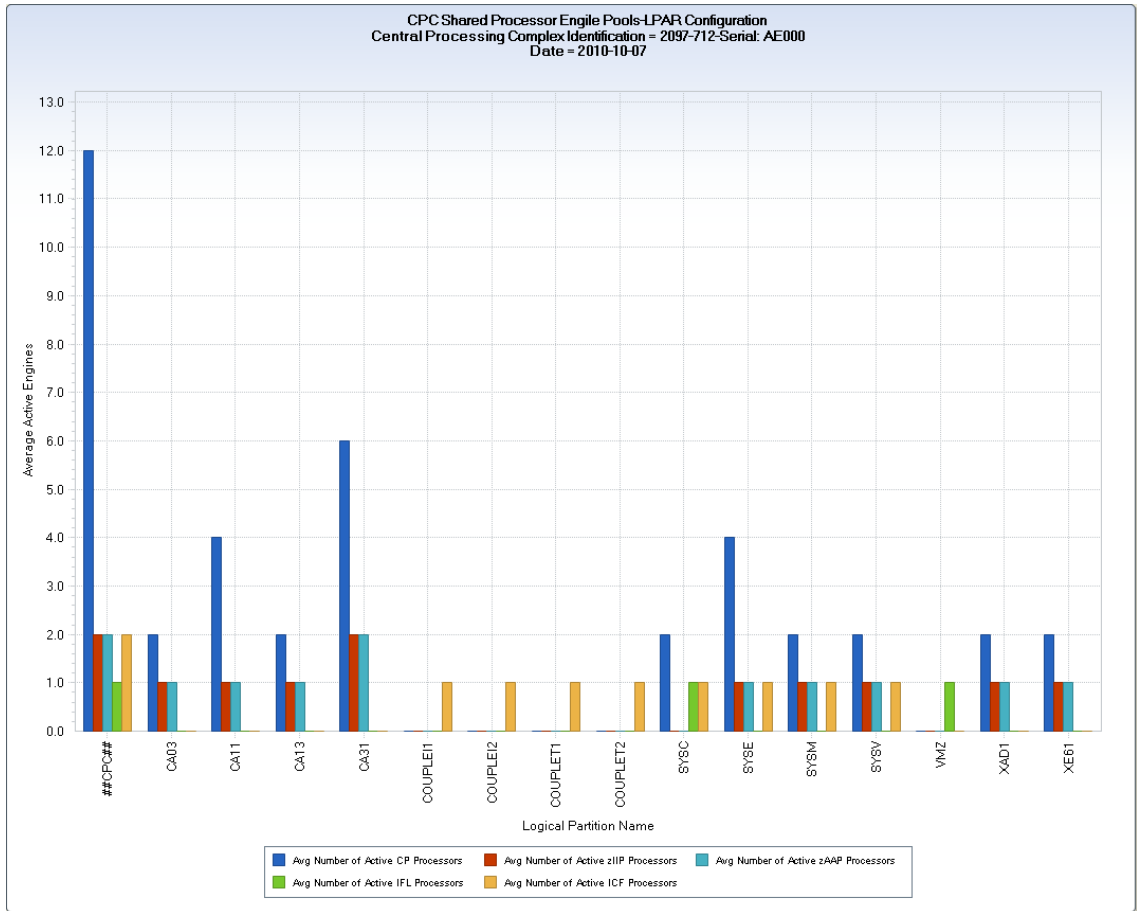
- [Mainframe CPC Configuration](#) (see page 105)
- [Mainframe CPC Engine Utilization](#) (see page 120)
- [Mainframe CPC Memory Use and Paging](#) (see page 156)
- [Mainframe CPC Specialty Engine Focus](#) (see page 170)
- [Mainframe MSU Utilization and LPAR Capping](#) (see page 189)
- [WLM Service Class and Report Class Analysis](#) (see page 207)
- [Hardware Instrumentation Services CPU Measurement Facility](#) (see page 233)

Mainframe CPC Configuration

This section contains the following topics:

- [RMFCFG - CPC LPAR Engine Configuration](#) (see page 106)
- [RMFCFH - CPC LPAR Engine Configuration by Hour](#) (see page 110)
- [RMFSTO - CPC LPAR Storage Allocation](#) (see page 114)
- [RMFWGT - CPC LPAR Relative Weight by Engine Pool](#) (see page 116)

RMFCFG – CPC LPAR Engine Configuration



Overview

The RMFCFG query generates a chart for each Central Processing Complex (CPC) that shows the physical engine configuration for the CPC and the logical engine assignments for each LPAR. One data extract shows the engine distribution for the shared engine pools, and the other for the dedicated engine pools.

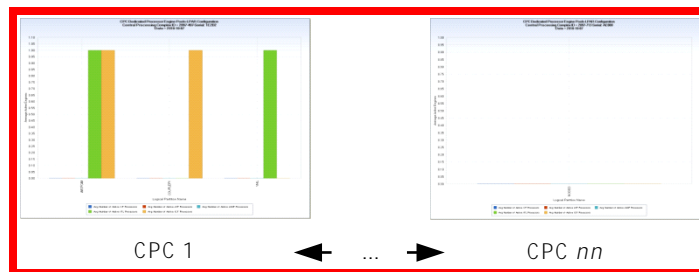
The example shows the chart generated for the *shared engine pools* of an IBM z10 2097-712 CPC. The five vertical bars for the leftmost LPAR, named '#CPC##', show the number of active *physical* processors (engines) of each type (CP, zAAP, zIIP, IFL, and ICF) in the shared engine pools. The *logical* processor assignments for the active LPARs in the CPC follow to the right. A similar chart is generated for the dedicated engine pools. Notice that Coupling Facility LPARs have only one engine type assigned (ICFs) while the z/OS LPARs (for example, 'CA11' and 'CA31') typically have three engine types assigned (CPs, zIIPs, and zAAPs). One of the VM LPARs 'VMZ' has IFL engines only, while another, 'SYSC', has CPs, IFLs, and ICFs.

Uses

This query helps you see the LPAR configuration at a glance—and serve as a basis to compare against some of the queries that show actual engine utilization values to validate that logical processor assignments make sense. To minimize multiprocessor management overhead, assign the minimum number of engines required to manage the workload to an LPAR.

Data Extracts

CPC Dedicated Engine Pool LPAR Configuration



Left Y-axis

LPAR ##CPC##

PRSMACDP - Avg CP Dedicated Processors

PRSMAIDP - Avg ICF Dedicated Processors

PRSMANDP - Avg IFL Dedicated Processors

PRSMAZDP - Avg zAAP Dedicated Processors

PRSMASDP - Avg zIIP Dedicated Processors

Real LPARs

LPCAVCPU - Avg Number of Active CP Processors

LPCAVICF - Avg Number of Active ICF Processors

LPCAVIFL - Avg Number of Active IFL Processors

LPCAVZAP - Avg Number of Active zAAP Processors

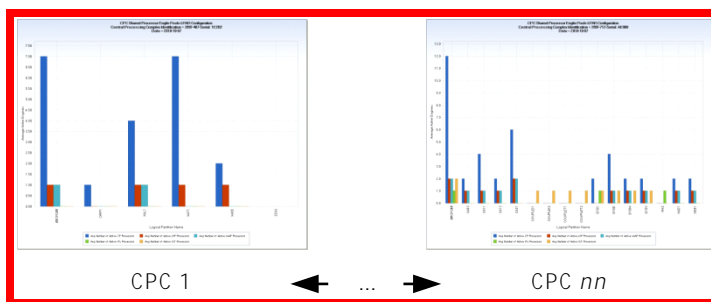
LPCAVSUP - Avg Number of Active zIIP Processors

X-axis

PRSMLPNM – Logical Partition Name

Note: If there are no dedicated engines for a CPC, a chart is produced showing a single LPAR, named "NODED".

CPC Shared Engine Pool LPAR Configuration



Left Y-axis

LPAR ##CPC##

PRSMACSP - Avg CP Shareable Processors
 PRSMAISP - Avg ICF Shareable Processors
 PRSMANSP - Avg IFL Shareable Processors
 PRSMAZSP - Avg zAAP Shareable Processors
 PRSMASSP - Avg zIIP Shareable Processors

Real LPARs

LPCAVCPU - Avg Number of Active CP Processors
 LPCAVICF - Avg Number of Active ICF Processors
 LPCAVIFL - Avg Number of Active IFL Processors
 LPCAVZAP - Avg Number of Active zAAP Processors
 LPCAVSUP - Avg Number of Active zIIP Processors

X-axis

PRSMLPNM – Logical Partition Name

Note: If there are no shared engines for a CPC, a chart is produced showing a single LPAR, named "NOSHARE".

Query Defaults

- HARLPC DAYS timespan file: Cycle 01
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- User code overwrites LPAR='PHYSICAL' to represent values for CPC level LPAR: '##CPC##'

- Derivations:

DATE

DATEPART(ENDTS)

CPCID

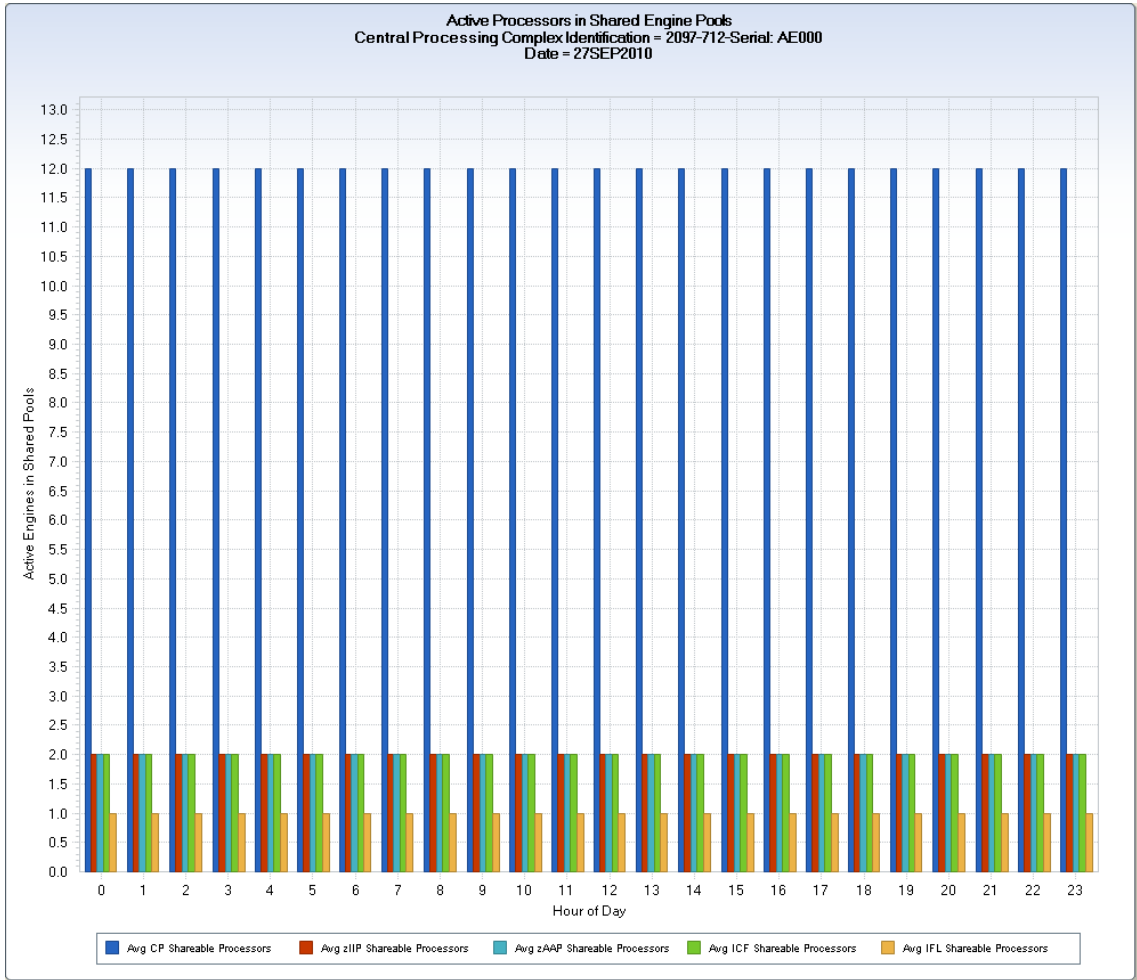
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);

Modifications

The following modifications can be made to the RMFCFG query:

- You can add filter statements to select only a single CPC, or set of CPCs using some of the identifying data elements shown in the CPCID derivation.
- If you do not have the DAYS timespan HARLPC file active, you can run the query against the DETAIL timespan, cycle 01.
- If you have no dedicated engines, you can delete the data extract step for the dedicated engine pool.

RMFCFH – CPC Engine Configuration by Hour



This query shows yesterday's active *physical* engine configuration, by hour, for the Central Processing Complex (CPC) shared and dedicated engine pools. The query generates two vertical bar charts for each CPC.

Five vertical bars show the number of active *physical* processors (engines) of each type (CP, zAAP, zIIP, IFL, and ICF) in the CPC for each hour of the day. The example shows the physical shared engine pool for an IBM z10 2097 model 712. In this example, the physical engine configuration was consistent for the entire day charted.

Overview

The RMFCFH query generates a chart for each Central Processing Complex (CPC) that shows the *physical* engine configuration for the CPC, by hour. One data extract shows the hourly physical engine configuration for the shared engine pools, and the other for the dedicated engine pools.

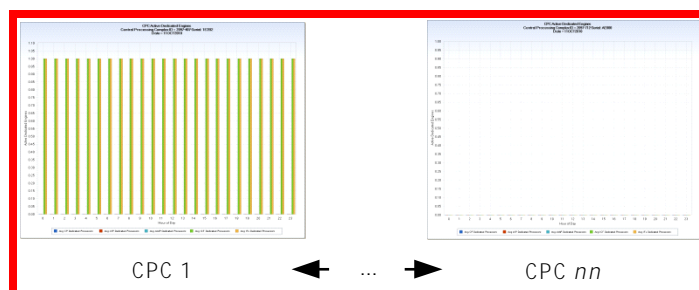
The example shows the physical shared engine pool for an IBM z10 2097 model 712. In this example, the physical engine configuration was consistent for the entire day charted.

Uses

This chart helps you see the impact of various features available with IBM mainframes to dynamically activate and deactivate physical engines to match variable workload activity. For example, IBM's Capacity Provisioning can provide rules to activate and deactivate engines based on various user-defined rules. These charts show any physical engine configuration changes at an hourly level.

Data Extracts

CPC Dedicated Eng Pools Config by Hour



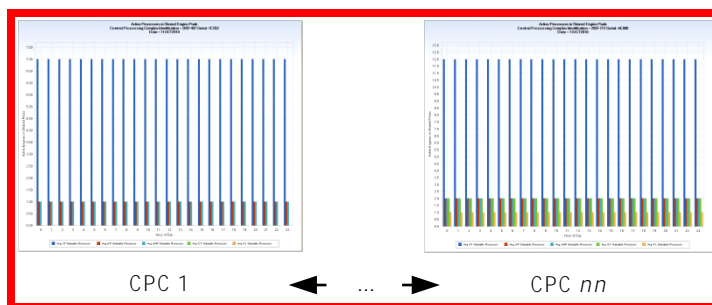
Left Y-axis

- PRSMACDP - Avg CP Dedicated Processors
- PRSMAIDP - Avg ICF Dedicated Processors
- PRSMANDP - Avg IFL Dedicated Processors
- PRSMAZDP - Avg zAAP Dedicated Processors
- PRSMASDP - Avg zIIP Dedicated Processors

X-axis

HOUR – Hour of Day

CPC Shared Engine Pools Config by Hour



Left Y-axis

- PRSMACSP - Avg CP Shareable Processors
- PRSMAISP - Avg ICF Shareable Processors
- PRSMANSP - Avg IFL Shareable Processors
- PRSMAZSP - Avg zAAP Shareable Processors
- PRSMASSP - Avg zIIP Shareable Processors

X-axis

HOUR – Hour of Day

Query Defaults

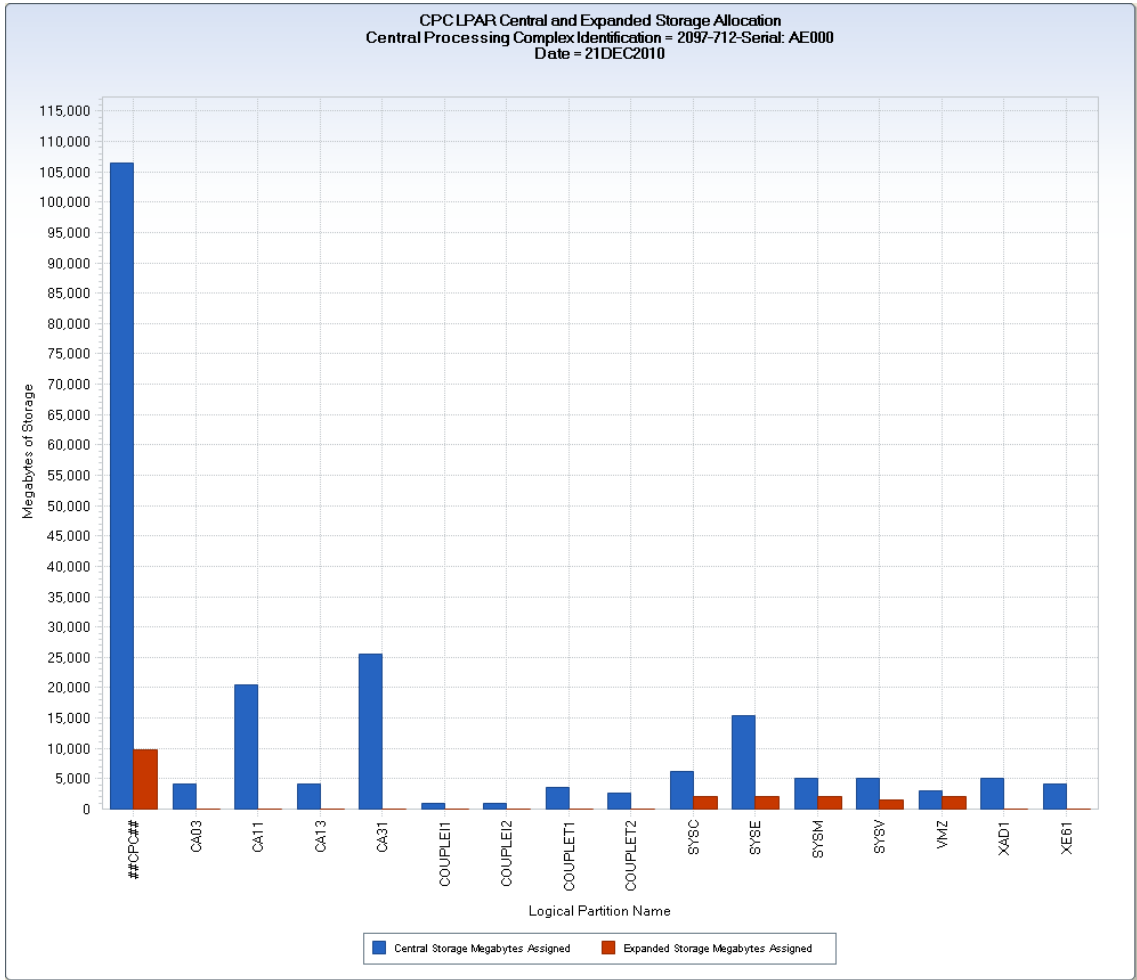
- HARLPC DAYS timespan file: Cycle 01
- Filters: PRSMNAME='PHYSICAL' (Select "PHYSICAL" LPAR only)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
DATE
DATEPART(ENDTS)
CPCID
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5) ;

Modifications

The following modifications can be made to the RMFCFH query:

- You can add filter statements to select only a single CPC, or set of CPCs using some of the identifying data elements shown in the CPCID derivation.
- If you have no dedicated engines, you can delete the data extract step for the dedicated engine pool.

RMFSTO – CPC LPAR Storage Allocation



Overview

The RMFSTO query generates a chart for each Central Processing Complex (CPC). The chart shows the central and expanded storage for the CPC, and how that storage is assigned to each LPAR.

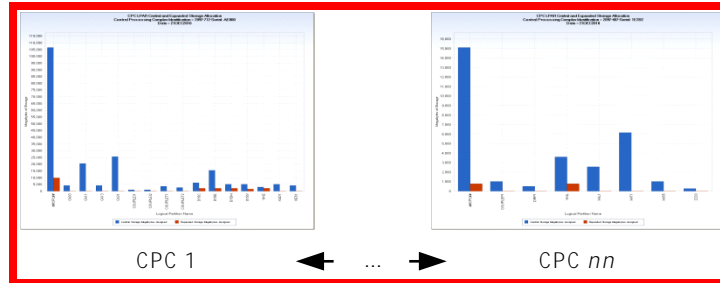
The example shows the chart generated for an IBM z10 2097-712 CPC. The two vertical bars for the leftmost LPAR, named ##CPC##, show the total amount of central and expanded storage, in megabytes, in the CPC. The distribution of this storage is shown for the active LPARs to the right. Notice that all LPARs are assigned some amount of central storage, but not all LPARs have expanded storage.

Uses

This query provides a useful view of CPC storage distribution at a glance. Analysis of central and expanded storage workload constraint may require better distribution of the available resource.

Data Extracts

CPC LPAR Storage Allocation



Left Y-axis

LPCCSF - Central Storage Megabytes Assigned

LPCEsf – Expanded Storage Megabytes Assigned

X-axis

PRSM LPNM – Logical Partition Name

Note: The leftmost LPAR, '##CPC##' storage values are the sum of all storage allocated to the active LPARs.

Query Defaults

- HARLPC DAYS timespan file: Cycle 01
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- User code overwrites LPAR='PHYSICAL' to represent values for CPC level LPAR: '##CPC##'

- Derivations:

DATE

DATEPART (ENDTS)

CPCID

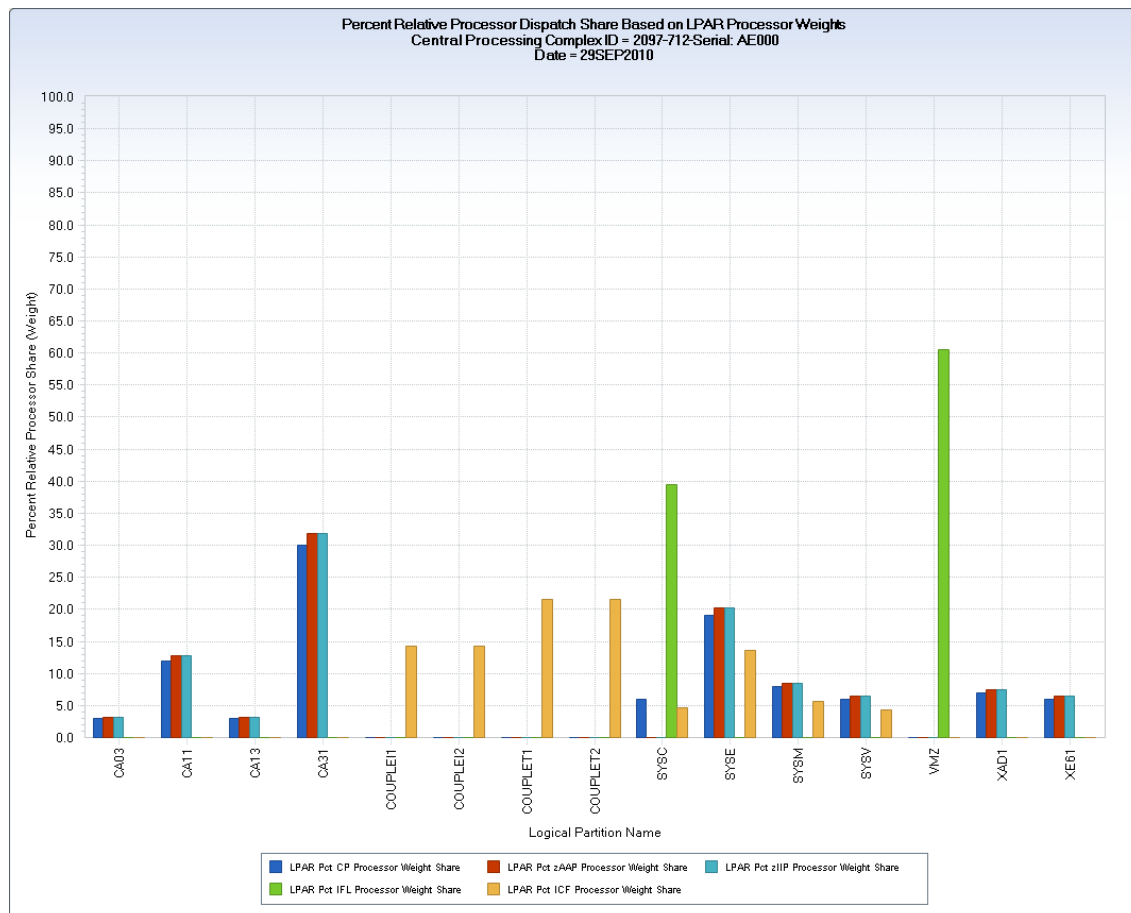
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);

Modifications

The following modifications can be made to the RMFSTO query:

- You can add filter statements to select only a single CPC, or set of CPCs using some of the identifying data elements shown in the CPCID derivation.

RMFWGT – CPC LPAR Relative Weight by Engine Pool



Overview

The RMFWGT query shows the relative dispatch hierarchy for the shared LPARs defined to a CPC, based on the system programmer specified weights for each LPAR. LPAR weights are designated for each logical processor engine type assigned to each LPAR.

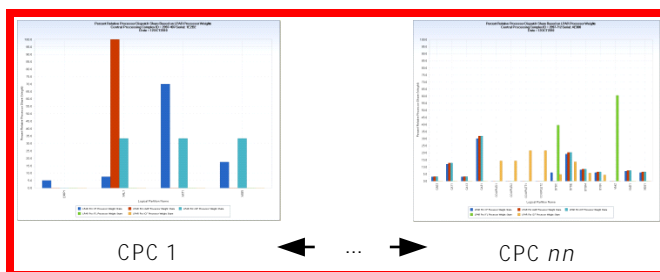
In the example, the fourth LPAR from the left, "CA31" has three logical processor engine types assigned—CPs, zIIPs, and zAAPs. Relative to the other LPARs defined to the CPC, the weights assigned to "CA31" tell PR/SM that when there is contention for the physical engines, "CA31" dispatches physical engines to handle its workloads about 30 percent of the time (for CPs) and 33 percent of the time (for zIIPs and zAAPs). For the two LPARs that use IFL engines (green vertical bars), "SYSC" and "VMZ", when there is contention for the IFL engines, VMZ gets about 60 percent of the dispatches and "SYSC" gets about 40 percent.

Uses

This query provides a useful overview of how PR/SM reports the various LPARs when there is contention for the physical engines in the shared engine pools. The dispatch priority hierarchy, defined by these relative weight values, should be in agreement with the importance of the business critical applications dispatched on the various LPARs. If the shared engine pool for a particular processor type is heavily used, an LPAR that primarily runs discretionary workloads should not receive as large a percentage of engine dispatches as an LPAR that executes business critical workloads.

Data Extracts

CPC LPAR Weights PCT by Engine Pool



Left Y-axis

PCTCPW - LPAR Pct CP Processor Weight Share

PCTZPAW - LPAR Pct zAAP Processor Weight Share

PCTZIPW - LPAR Pct zIIP Processor Weight Share

PCTIFLW - LPAR Pct IFL Processor Weight Share

PCTICFW - LPAR Pct ICF Processor Weight Share

X-axis

PRSMPLNM – Logical Partition Name

Query Defaults

- HARVPA DETAIL timespan file: Cycle 01
- Filters:
 - VPADEDCT EQ 0 – no dedicated LPARs
 - PRSMLPNM NE 'PHYSICAL' – eliminate physical (overhead) LPAR
 - PRSMLPTP EQ 'CP' or 'ICF' – only PR/SM managed LPARS
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

DATE

DATEPART(ENDTS)

CPCID

TRIM(LPCMOD)||'-'||TRIM(CPCMODID)||'-Serial: '||SUBSTR(CPCSEQNB,12,5);

PCTCPW

IF PRSMTWGT AND CP_W THEN PCTCPW=ROUND((CP_W / PRSMTWGT * 100),0.1) ; ELSE
PCTCPW=0 ;

PCTZAPW

IF PRSMZWGT AND ZAP_W THEN PCTZAPW=ROUND((ZAP_W / PRSMZWGT * 100),0.1) ; ELSE
PCTZAPW=0 ;

PCTZIPW

IF PRSMSWGT AND ZIP_W THEN PCTZIPW=ROUND((ZIP_W / PRSMSWGT * 100),0.1) ; ELSE
PCTZIPW=0 ;

PCTIFLW

IF PRSMNWGT AND IFL_W THEN PCTIFLW=ROUND((IFL_W / PRSMNWGT * 100),0.1) ; ELSE
PCTIFLW = 0

PCTICFW

IF PRSMIWGT AND ICF_W THEN PCTICFW=ROUND((ICF_W / PRSMIWGT * 100),0.1) ; ELSE
PCTICFW=0 ;

Note: A user step sets CP_W, ZAP_W, ZIP_W, IFL_W, and ICF_W to the VPABPS (Partition Processor Weight) value for each individual engine type, per LPAR, in the HARVPA file. If an engine type is not present for an LPAR, the corresponding xxx_W element is set to zero.

Modifications

The following modifications can be made to the RMFWGT query:

- You can add filter statements to select only a single CPC, or set of CPCs using identifying data elements shown in the CPCID derivation.

Mainframe CPC Engine Utilization

This section contains the following topics:

[RMFCPD - CPC and LPAR Daily Shared CP Engine Dispatch by Hour](#) (see page 121)

[RMFEHH - CPC Shared Engine Pools Percent Dispatch High Hours](#) (see page 127)

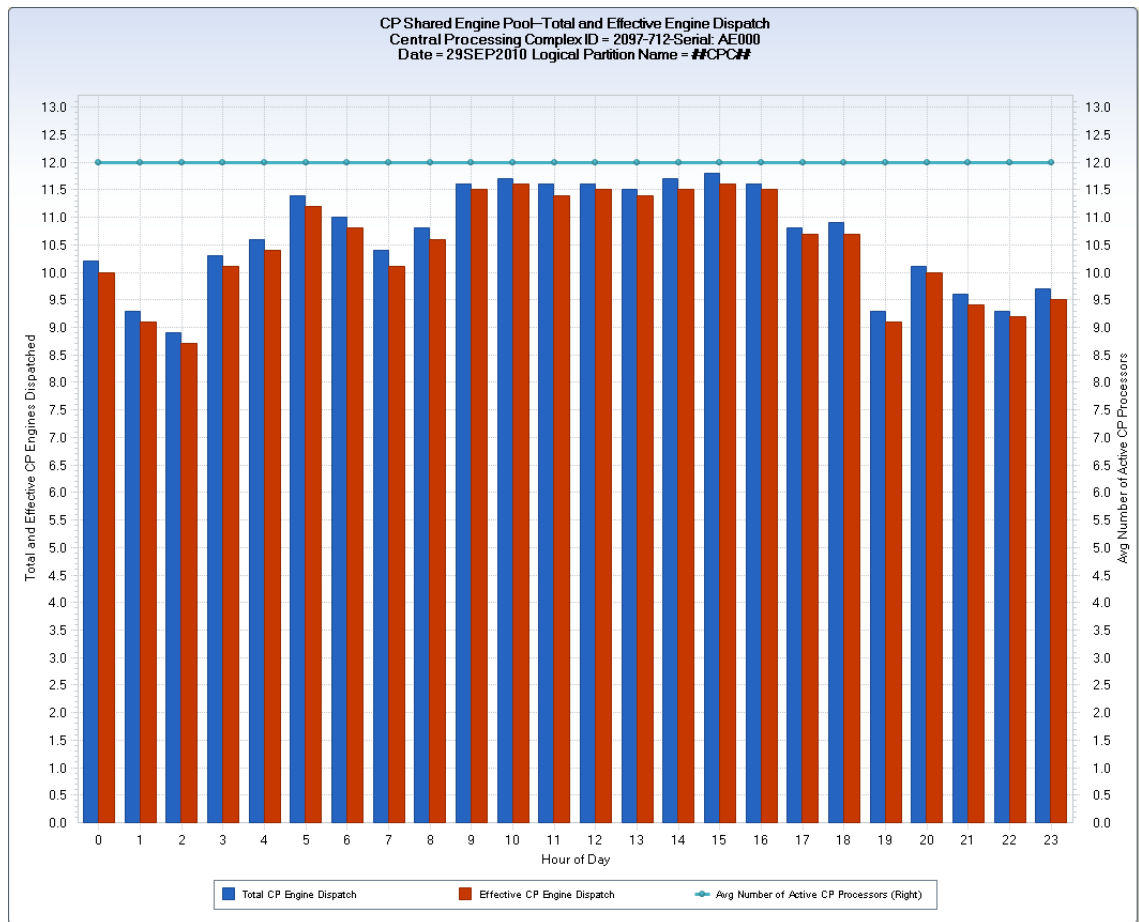
[RMFICD - CPC and LPAR Daily Shared ICF Engine Dispatch by Hour](#) (see page 133)

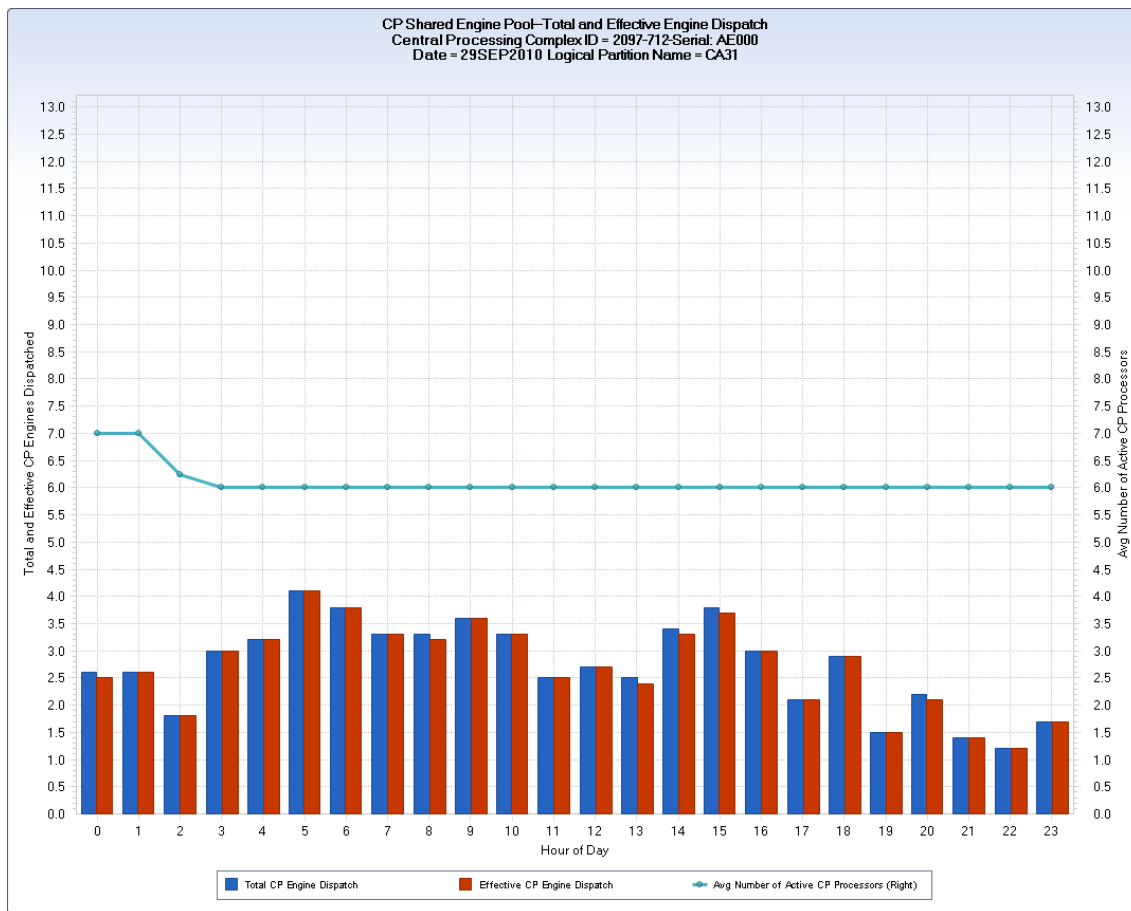
[RMFIFD - CPC and LPAR Daily Shared IFL Engine Dispatch by Hour](#) (see page 139)

[RMFZAD - CPC and LPAR Daily Shared zAAP Engine Dispatch by Hour](#) (see page 145)

[RMFZID - CPC and LPAR Daily Shared zIIP Engine Dispatch by Hour](#) (see page 151)

RMFCPD – CPC and LPAR Daily Shared CP Engine Dispatch by Hour





Overview

The LPCPDP query creates a single data extract CSV showing hourly CPC and LPAR dispatch percentages for the shared CP (Central Processor) engine pool. For each Central Processing Complex (CPC) encountered in the DAYS timespan HARLPC file, the query generates a CPC level chart. The chart uses vertical bars to show the total and effective dispatch percentage of the *physical* CP engines for each hour of the day—plus a horizontal reference line showing the total number of *physical* engines in the shared CP processor pool. Individual charts for each LPAR in the CPC follows the CPC level chart. These charts show the dispatch percentages of the *logical* CP engines assigned to the LPAR. For the LPAR charts, the horizontal reference line shows the number of *logical* CP engines active for the LPAR.

The difference between total and effective dispatch represents the amount of the engine resource pool required for multiprocessing management overhead.

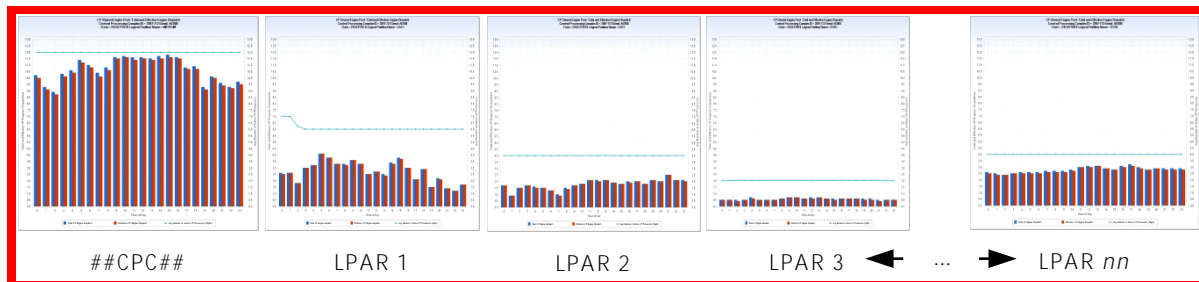
Notice in the second example (the LPAR level chart), LPAR CA31 shows that the number of logical CP engines assigned to the LPAR was reduced from seven, at 01:00 AM to six, by 03:00 AM. This reduction in the number of logical engines for LPAR CA31 can be effected with an operator command, or by a dynamic facility such as the Intelligent Resource Director (IRD).

Uses

This query is used to monitor the overall shared CP pool engine utilization of each CPC at an hour granularity. The individual LPAR views show to what extent each LPAR is using the logical CP engines available for its use. Even though the CPC level and LPAR level charts use the same chart variables, the actual values charted are different.

Data Extracts

CPC and LPAR Daily CP Engine Dispatch



This query generates a single data extract, but data manipulation is used to convert the "PHYSICAL" LPAR to a CPC level view of shared CP pool engine dispatch, where the LPAR name is changed to "##CPC##". The following data extract description is presented in two parts: one describing the element values plotted for the CPC level chart, and the other describing the element values plotted for the individual LPARs.

CPC level chart

For the CPC level chart, CPC level PR/SM data element values are substituted for the LPAR data elements:

Left Y-axis

ENGDISPT – Total CP Engine Dispatch (Set to PRSMTCDT – Total CP Dispatch Time / 3600)

ENGDISPE – Effective CP Engine Dispatch (Set to PRSMT CET – Total CP Dispatch Time – Effective / 3600)

This vertical bars, for the CPC level LPAR ##CPC##, show the total and effective dispatch of the physical processors in the entire shared CP engine pool. For this chart, a user data step takes the following actions before the ENGDISPT and ENGDISPE derivations:

- Overwrite the 'PHYSICAL' LPAR observation.
- Change the LPAR name to '##CPC##'.
- Substitute PRSMTCDT and PRSMT CET values for the LPAR equivalent data elements.

This process permits a CPC level view of the physical shared CP engine pool dispatch activity. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as physical engine dispatch hour equivalents. For example, three CP engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVCPU - Avg Number of Active CP Processors (Set to PRSMACSP – Avg CP Shareable Processors))

This horizontal reference line, for the CPC level LPAR ##CPC##, shows the average number of physical processors in the shared CP engine pool. While the data extract plots data element LPCAVCPU, which is the average number of logical CP engines assigned to an LPAR—for the ##CPC## chart, a user data step populates LPCAVCPU with the value for PRSMACSP—which is the average number of physical engines in the shared CP engine pool. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like capacity provisioning and on/off capacity on demand can activate and deactivate physical processors—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

LPAR level charts

The LPAR level charts use standard HARLPC file data elements to derive the dispatch variables:

Left Y-axis

ENGDISPT – Total CP Engine Dispatch (Set to LPCTODTM – Total CP Processors Dispatch Time / 3600)

ENGDISPE – Effective CP Engine Dispatch (Set to LPCTOEDT – Total CP Proc. Effective Dispatch Time / 3600)

The vertical bars, for the LPAR level charts, show the total and effective dispatch of the *logical* CP processors assigned to the LPAR. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as *logical* engine dispatch hour equivalents. For example, three logical CP engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVCPU - Avg Number of Active CP Processors

This horizontal reference line, for the LPAR level charts, show the average number of logical CP processors assigned to the LPAR. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like the Intelligent Resource Director (IRD) can activate and deactivate logical processors at the LPAR level—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

Query Defaults

- HARLPC DAYS timespan file: Cycles 01 and 02
- Filters:
 - LPCTOEDT or LPCTODTM GT 0 or PRSMLPNM="PHYSICAL" (Non-zero effective or total dispatch times or the "PHYSICAL" LPAR)
 - LPCDEDCT EQ 0 ; (no dedicated LPARS)
 - PRSMTSCP GT 0 ; (at least one active physical CP engine in shared engine pool)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

DATE

```
DATEPART(ENDTS)
```

CPCID

```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

ENGLDISPE

```
ROUND((LPCTOEDT/3600),.1) ;
```

ENGLDISPA

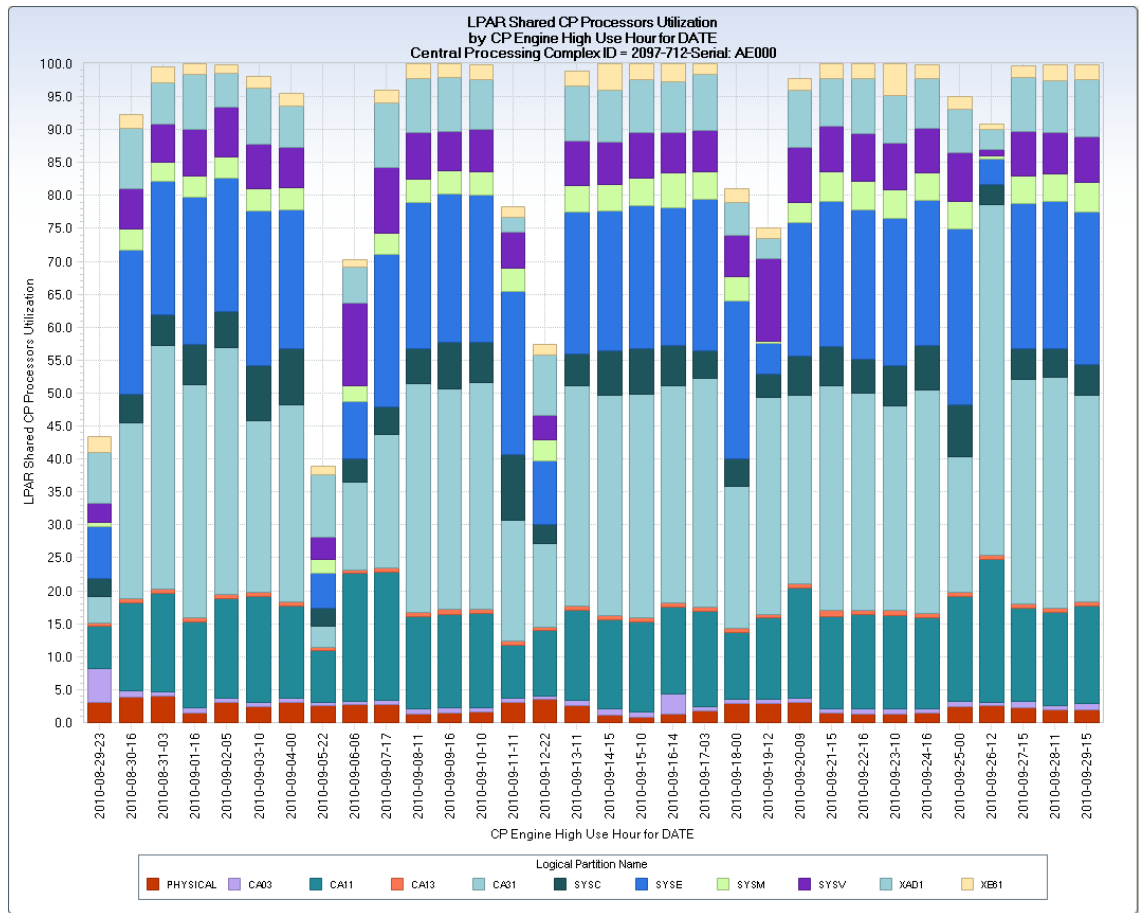
```
ROUND((LPCTODTM/3600),.1) ;
```

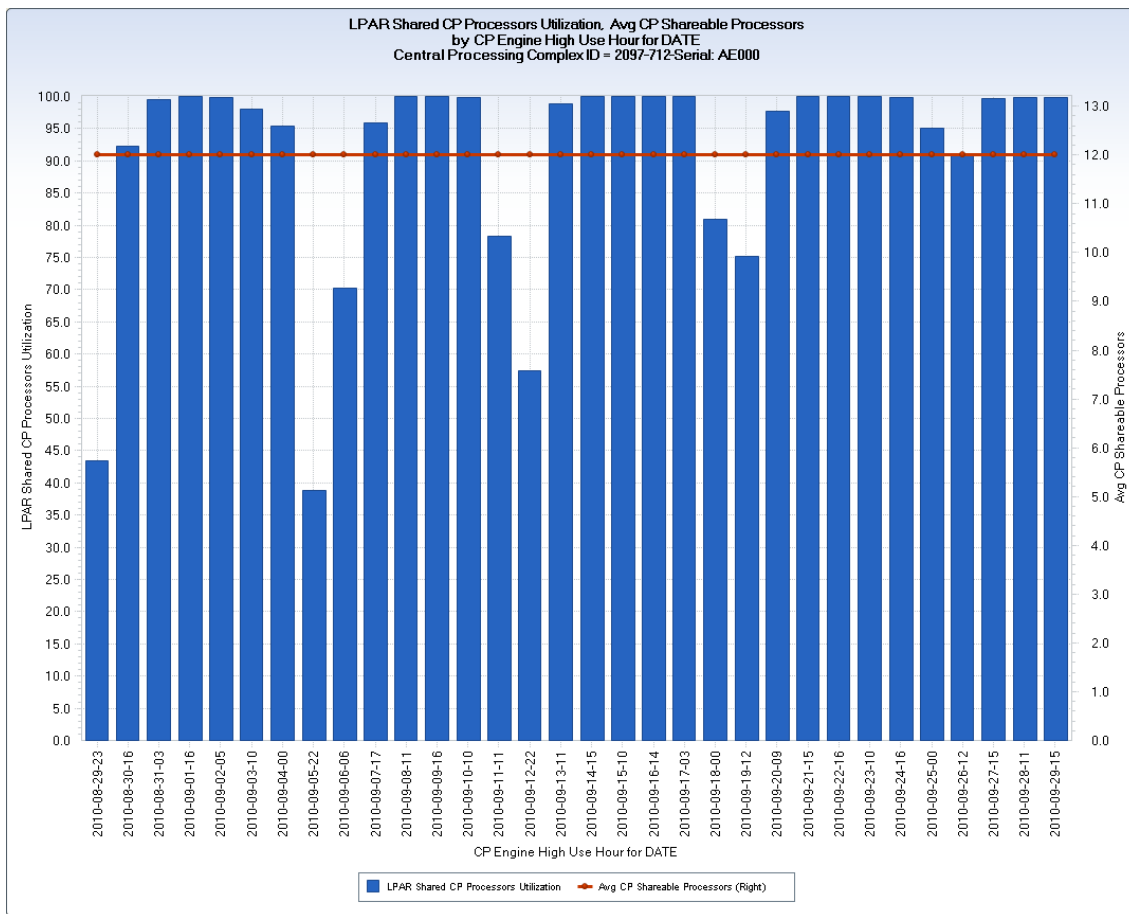
Modifications

The following modifications can be made to the LPCPDP query:

- You can comment out the user code in the initial file selection step that selects only yesterday's data. For example, you could read cycles 01-07 and generate charts for the past seven days.
- You can add filter statements to select only a single CPC, or set of CPCs using some of the identifying data elements shown in the CPCID derivation.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)

RMFEHH – CPC Shared Engine Pools Percent Dispatch High Hours





Overview

The RMFEHH query shows the peak hour of dispatch, over the past 32 days, for each engine type (CP, zIIP, zAAP, IFL, ICF) in the shared LPAR engine pools. The query generates two dispatch charts for each engine type in each Central Processing Complex (CPC).

The example shows the two charts generated for the shared CP engine pool of an IBM z10 2097-712 CPC. The top chart shows the peak hour, per day percent utilization of the CP engine pool, with the breakdown of the various LPARs that use CP engines. The second chart shows the same information, minus the LPAR breakdown. The chart includes a reference line (right Y-axis) that shows the number of engines in the CP engine pool (12 engines in this example). The X-axis shows the date concatenated to the high use hour. In the example, the first vertical bar shows that the peak CP engine dispatch occurred on August 29th, 2010 (a Sunday) during hour 23: "2010-08-29-23"—when the utilization of the CP engine pool was about 43 percent.

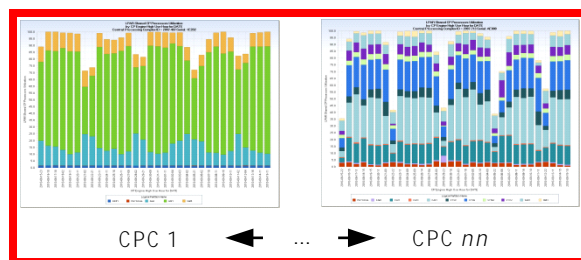
A similar set of charts is created for each engine type, one where the vertical bars show individual LPAR percent utilization, and the other showing percent utilization plus the physical engine count for the engine type.

Uses

This query is used to examine the peak utilization load for the CPC shared engine pools over the past month. Consistent peak hour utilization at, or near 100 percent means that, for at least 1 hour each day an engine pool has reached the utilization limit. It is likely that some workloads are being delayed, or receiving suboptimal attention. Consistent peak hour utilization significantly below 100 percent utilization means that you have invested more than necessary in a particular engine type.

Data Extracts

CPC Shared CP Pool Pct Use by LPAR HH



Left Y-axis

LPCPCSSU - LPAR Shared CP Processors Utilization

A user step analyzes HARLPC file data to determine the specific hour for each day where the Central Processing Complexes CP engines in the shared engine pool were most heavily dispatched. Once determined, the percent utilization of each LPAR using the CP engine pool during that peak hour is stacked in a vertical bar using data element LPCPCSSU.

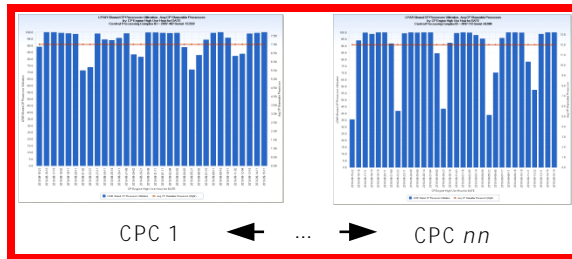
X-axis

CP_DDHH - CP Engine High Use Hour for DATE

Data element CP_DDHH is derived by appending the high use hour to the date, in the form:

YYYY-MM-DD-HH (for example, 2010-08-31:22, meaning hour 22 of August 31st, 2010).

CPC Shared CP Pool Pct Use High Hour



Left Y-axis

LPCPCSSU - LPAR Shared CP Processors Utilization

The same overall shared CP engine pool utilization as shown in the previous data extract, without the individual LPAR breakdown. LPCPCSSU is summarized to the CPC level.

Right Y-axis

PRSMACSP - Avg CP Shareable Processors

Data element PRSMACSP shows the average number of CP engines in the shared pool for the high use hour. If the number of physical engines in the shared CP pool changes, this value changes over the days plotted.

X-axis

CP_DDHH - CP Engine High Use Hour for DATE

Data element CP_DDHH is derived by appending the high use hour to the date, in the form:

YYYY-MM-DD-HH (for example, 2010-08-31:22, meaning hour 22 of August 31st, 2010).

Note: This query generates eight additional data extracts, two for each of the four other engine types, ICFs, IFLs, zAAPs, and zIIPs. The charts, and methods used to generate them are identical to those used to build the CP engine pool charts. The remaining data extract documentation lists the specific data elements used for the left and right Y-axis:

- **CPC Shared ICF Pool Pct Use by LPAR HH**
Left Y axis
LPCPCSIU - LPAR Shared ICF Processors Utilization
- **CPC Shared ICF Pool Pct Use High Hour**
Left Y axis
LPCPCSIU - LPAR Shared ICF Processors Utilization
Right Y axis
PRSMALSP – Avg ICF Sharable Processors
- **CPC Shared IFL Pool Pct Use by LPAR HH**
Left Y axis
LPCPCSNU - LPAR Shared ICF Processors Utilization
- **CPC Shared IFL Pool Pct Use High Hour**
Left Y axis
LPCPCSNU - LPAR Shared ICF Processors Utilization
Right Y axis
PRSMANSP – Avg ICF Sharable Processors
- **CPC Shared zAAP Pool Pct Use by LPAR HH**
Left Y axis
LPCPCSZU - LPAR Shared zAAP Processors Utilization
- **CPC Shared zAAP Pool Pct Use High Hour**
Left Y axis
LPCPCSZU - LPAR Shared zAAP Processors Utilization
Right Y axis
PRSMAZSP – Avg zAAP Sharable Processors
- **CPC Shared zIIP Pool Pct Use by LPAR HH**
Left Y axis
LPCPCSDU - LPAR Shared zIIP Processors Utilization

- CPC Shared zIIP ICF Pool Pct Use High Hour

Left Y-axis

LPCPCSDU - LPAR Shared zIIP Processors Utilization

Right Y-axis

PRSMASSP – Avg zIIP Sharable Processors

Query Defaults

- HARLPC DAYS timespan file: Cycles 01 - 32
- Filters: LPCPCSxU GT 0 selected for each engine type
- User code excludes today's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

xx_DDHH

```
xx_DDHH = PUT(YEAR4,Z4.) || '-' || PUT(MONTH,Z2.) || '-' ||  
PUT(DAY,Z2.) || '-' || PUT(MAXxxH,Z2.) ;DATEPART(ENDTS)
```

xx Engine High Hour for DATE element derived for each engine type (xx)

CPCID

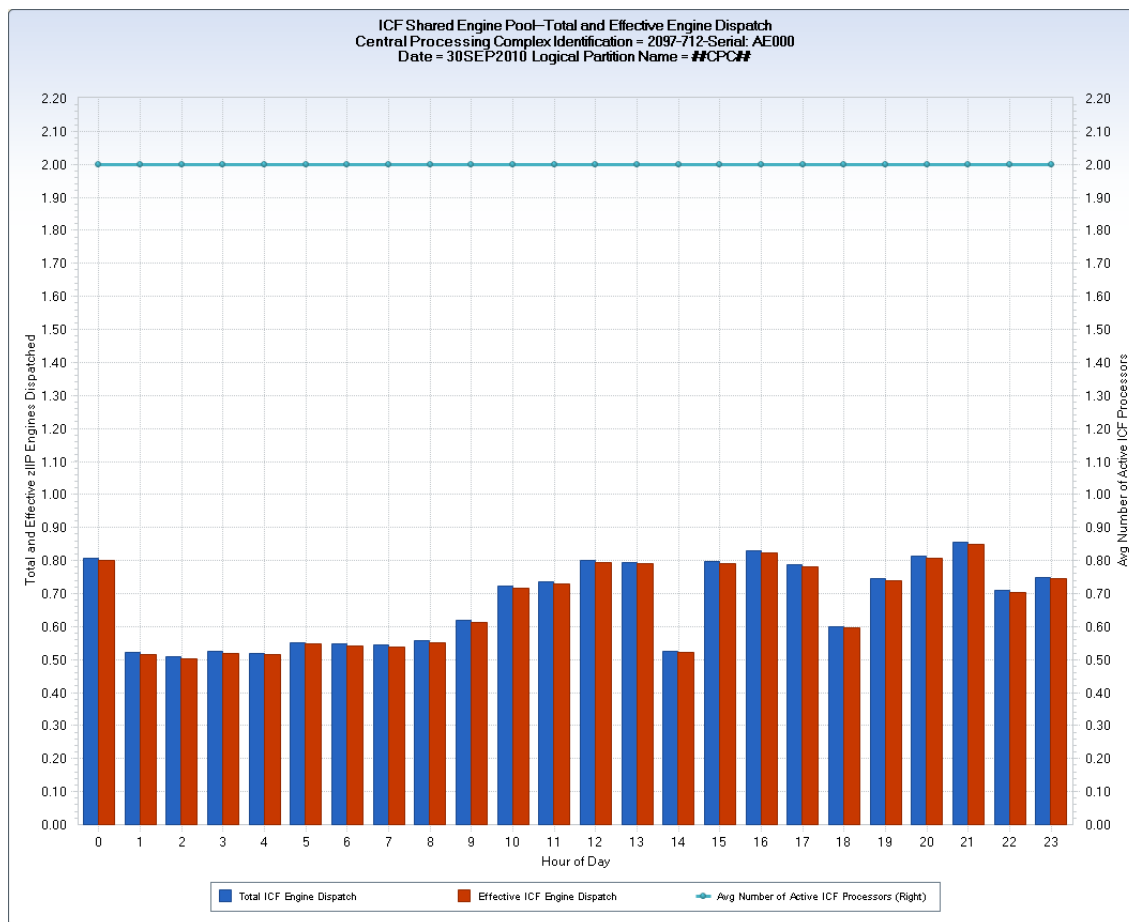
```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

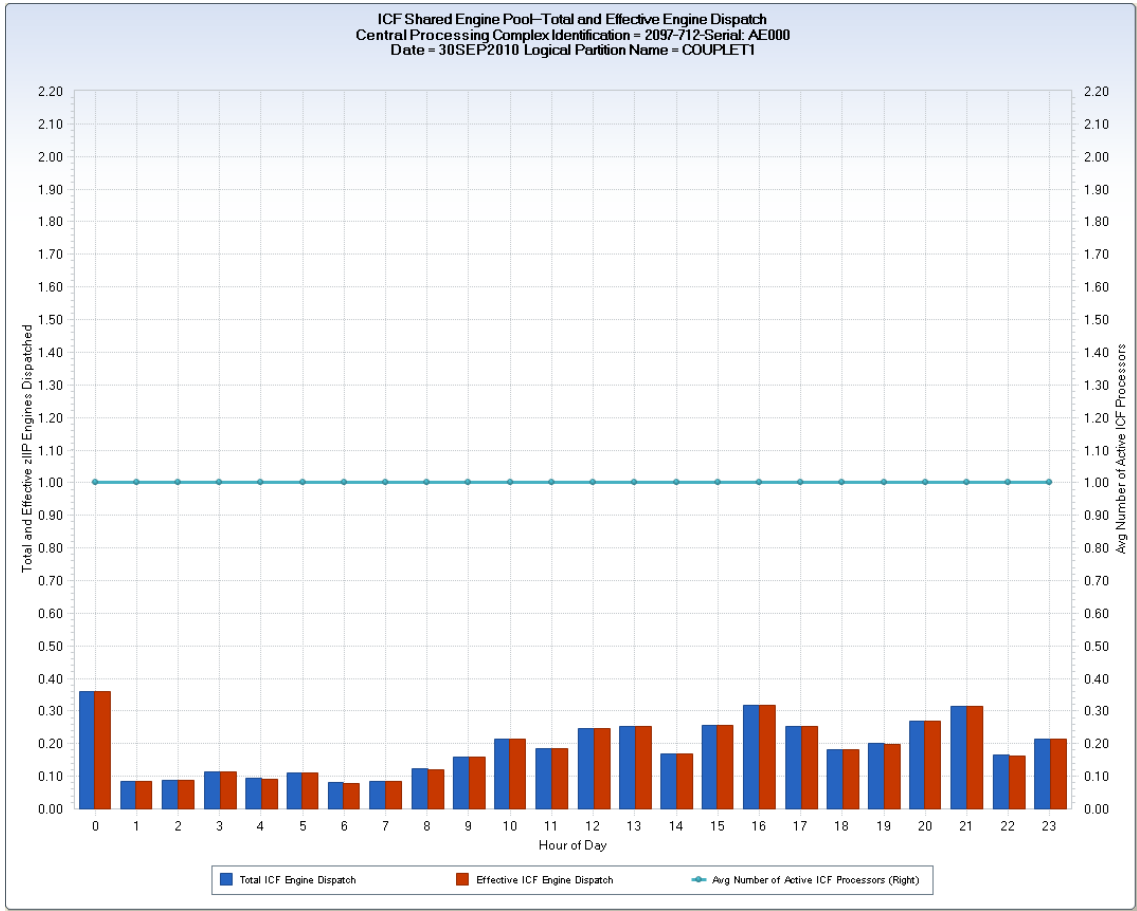
Modifications

The following modifications can be made to the RMFEHH query:

- You can change the HARLPC DAYS timespan cycles—defaults to cycles 1-32.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1).
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.
- You can delete data extract steps for engine types not found in your CPCs. For example, if you have no IFL engines, you can delete the two data extract steps for IFL engines.

RMFICD – CPC and LPAR Daily Shared ICF Engine Dispatch by Hour





Overview

The RMFICD query creates a single data extract CSV showing hourly CPC and LPAR dispatch percentages for the shared ICF (Integrated Coupling Facility) engine pool. For each Central Processing Complex (CPC) in the DAYS timespan HARLPC file, the query generates a CPC level chart. The chart uses vertical bars to show the total and effective dispatch percentage of the *physical* ICF engines for each hour of the day—plus a horizontal reference line showing the total number of *physical* engines in the shared ICF processor pool. The CPC level chart is followed by individual charts for each LPAR in the CPC, showing the dispatch percentages of the *logical* ICF engines assigned to the LPAR. For the LPAR charts, the horizontal reference line shows the number of *logical* ICF engines active for the LPAR.

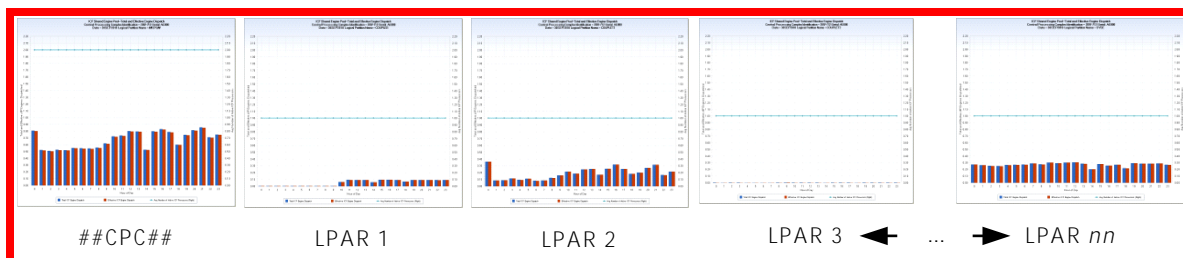
The difference between total and effective dispatch represents the amount of the engine resource pool required for multiprocessing management overhead.

Uses

This query is used to monitor the overall shared ICF pool engine utilization of each CPC at an hour granularity. The individual LPAR views show to what extent each LPAR is using the logical ICF engines available for its use. Even though the CPC level and LPAR level charts use the same chart variables, the actual values charted are different.

Data Extracts

CPC and LPAR Daily ICF Engine Dispatch



This query generates a single data extract, but data manipulation is used to convert the "PHYSICAL" LPAR to a CPC level view of shared ICF pool engine dispatch, where the LPAR name is changed to "##CPC##". The following data extract description is presented in two parts: one describing the element values plotted for the CPC level chart, and the other describing the element values plotted for the individual LPARs.

CPC level chart

For the CPC level chart, CPC level PR/SM data element values are substituted for the LPAR data elements:

Left Y-axis

ENGDISPT – Total ICF Engine Dispatch (Set to PRSMTIST – Total ICF Dispatch Time – Shared / 3600)

ENGDISPE – Effective ICF Engine Dispatch (Set to PRSMTIET – Total ICF Dispatch Time - Effective / 3600)

This vertical bars, for the CPC level LPAR **##CPC##**, show the total and effective dispatch of the physical processors in the entire shared ICF engine pool. For this chart, a user data step takes the following actions before the ENGDISPT and ENGDISPE derivations:

- Overwrite the 'PHYSICAL' LPAR observation.
- Change the LPAR name to '**##CPC##**'.
- Substitute PRSMTIST and PRSMTIET values for the LPAR equivalent data elements.

This process permits a CPC level view of the physical shared ICF engine pool dispatch activity. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as physical engine dispatch hour equivalents. For example, three ICF engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVICF - Avg Number of Active ICF Processors (Set to PRSMAISP – Avg ICF Sharable Processors)

This horizontal reference line, for the CPC level LPAR **##CPC##**, shows the average number of physical processors in the shared ICF engine pool. While the data extract plots data element LPCAVICF, which is the average number of logical ICF engines assigned to an LPAR—for the **##CPC##** chart, a user data step populates LPCAVICF with the value for PRSMAISP—which is the average number of physical engines in the shared ICF engine pool. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like capacity provisioning and on/off capacity on demand can activate and deactivate physical processors—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

LPAR level charts

The LPAR level charts use standard HARLPC file data elements to derive the dispatch variables:

Left Y-axis

ENGDISPT – Total ICF Engine Dispatch (Set to LPCSIDTM – Total shared ICF Proc. Dispatch Time / 3600)

ENGDISPE – Effective ICF Engine Dispatch (Set to LPCSIEDT – Total Shr ICF Proc. Effec. Dispatch Time / 3600)

The vertical bars, for the LPAR level charts, show the total and effective dispatch of the logical ICF processors assigned to the LPAR. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as logical engine dispatch hour equivalents. For example, three logical ICF engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVICF - Avg Number of Active ICF Processors

This horizontal reference line, for the LPAR level charts, show the average number of logical ICF processors assigned to the LPAR. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like the Intelligent Resource Director (IRD) can activate and deactivate logical processors at the LPAR level—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

Query Defaults

- HARLPC DAYS timespan file: Cycles 01 and 02
- Filters:
 - LPCSIDTM or LPCSIEDT GT 0 or PRSMLPNM="PHYSICAL" (Non-zero effective or total ICF engine dispatch times or the "PHYSICAL" LPAR)
 - LPCDEDCT EQ 0 ; (no dedicated LPARS)
 - PRSMTSIP GT 0 ; (at least one active physical ICF engine in shared engine pool)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

DATE

DATEPART(ENDTS)

CPCID

TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);

ENGDISPE

ROUND((LPCSIEDT/3600),.1) ;

ENGDISPA

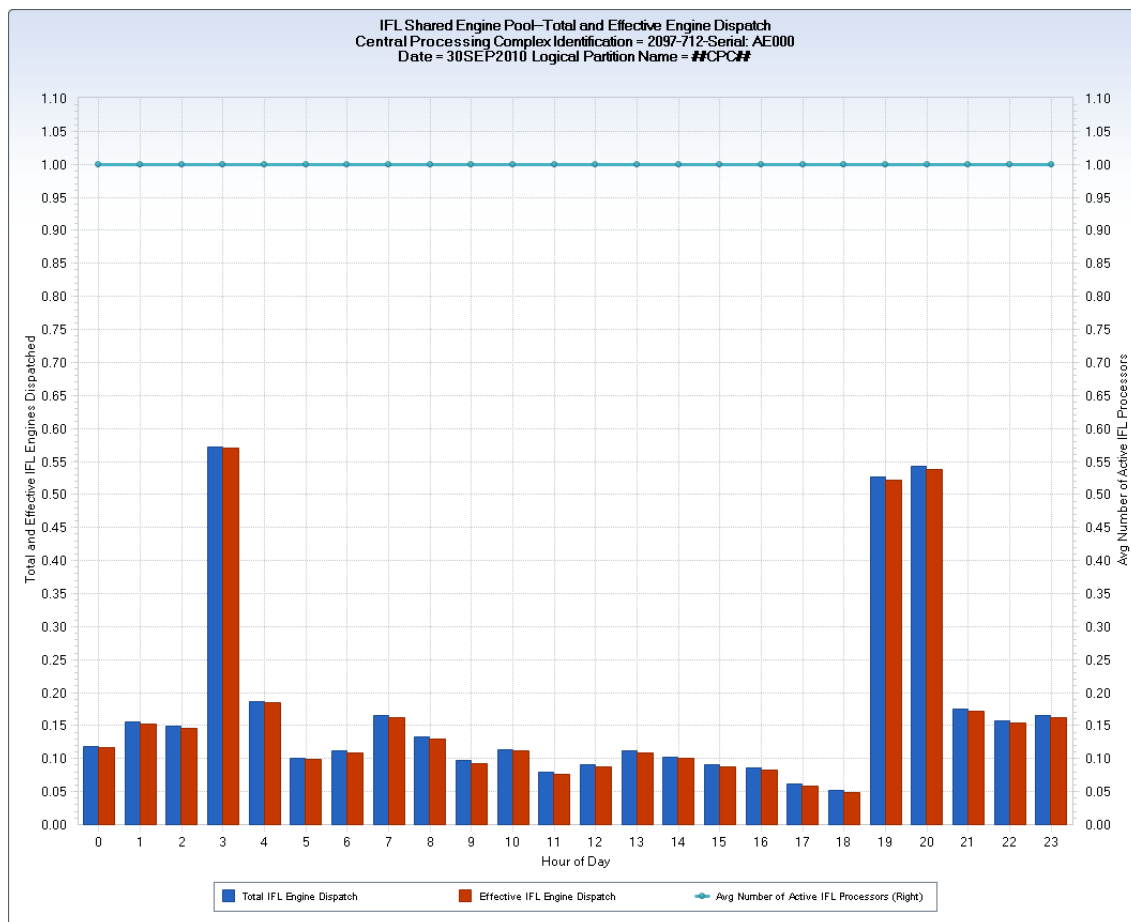
ROUND((LPCSIDTM/3600),.1) ;

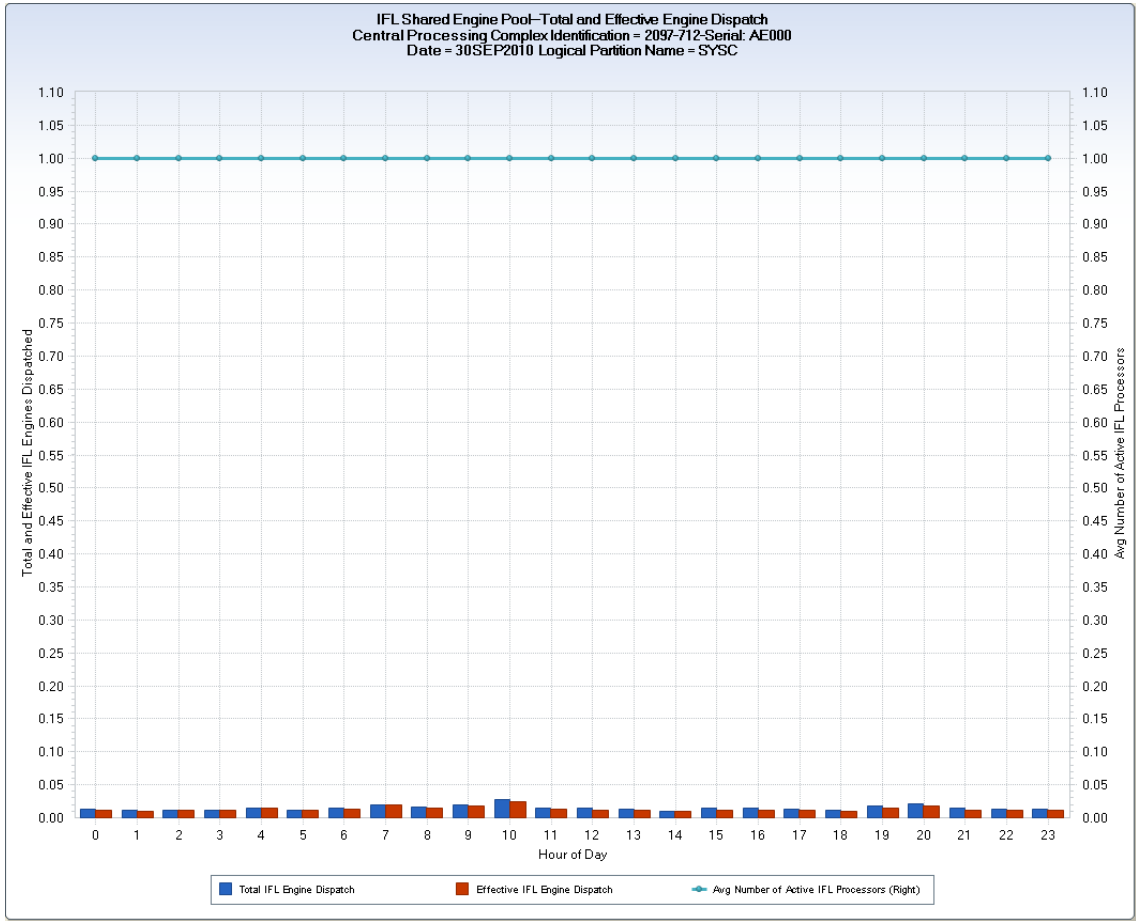
Modifications

The following modifications can be made to the RMFICD query:

- You can comment out the user code in the initial file selection step that selects only yesterday's data. For example, you could read cycles 01-07 and generate charts for the past seven days.
- You can add filter statements to select only a single CPC, or set of CPCs using some of the identifying data elements shown in the CPCID derivation.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or a particular time zone (for example, ZONE EQ 1)

RMFIFD – CPC and LPAR Daily Shared IFL Engine Dispatch by Hour





Overview

The RMFIFD query creates a single data extract CSV showing hourly CPC and LPAR dispatch percentages for the shared IFL (Integrated Facility for Linux) engine pool. For each Central Processing Complex (CPC) in the DAYS timespan HARLPC file, the query generates a CPC level chart. The chart uses vertical bars to show the total and effective dispatch percentage of the *physical* IFL engines for each hour of the day—plus a horizontal reference line showing the total number of *physical* engines in the shared IFL processor pool. The CPC level chart is followed by individual charts for each LPAR in the CPC, showing the dispatch percentages of the *logical* IFL engines assigned to the LPAR. For the LPAR charts, the horizontal reference line shows the number of *logical* IFL engines active for the LPAR.

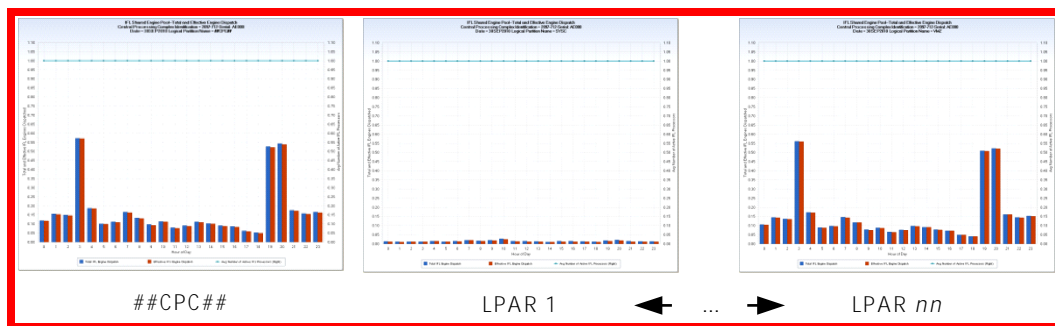
The difference between total and effective dispatch represents the amount of the engine resource pool required for multiprocessing management overhead.

Uses

This query is used to monitor the overall shared IFL pool engine utilization of each CPC at an hour granularity. The individual LPAR views show to what extent each LPAR is using the logical IFL engines available for its use. Even though the CPC level and LPAR level charts use the same chart variables, the actual values charted are different.

Data Extracts

CPC and LPAR Daily IFL Engine Dispatch



This query generates a single data extract, but data manipulation is used to convert the "PHYSICAL" LPAR to a CPC level view of shared IFL pool engine dispatch, where the LPAR name is changed to "##CPC##". The following data extract description is presented in two parts: one describing the element values plotted for the CPC level chart, and the other describing the element values plotted for the individual LPARs.

CPC level chart

For the CPC level chart, CPC level PR/SM data element values are substituted for the LPAR data elements:

Left Y-axis

ENGDISPT – Total IFL Engine Dispatch (Set to PRSMTNST – Total IFL Dispatch Time – Shared / 3600)

ENGDISPE – Effective IFL Engine Dispatch (Set to PRSMTNET – Total IFL Dispatch Time - Effective / 3600)

This vertical bars, for the CPC level LPAR ##CPC##, show the total and effective dispatch of the physical processors in the entire shared IFL engine pool. For this chart, a user data step takes the following actions before the ENGDISPT and ENGDISPE derivations:

- Overwrite the 'PHYSICAL' LPAR observation.
- Change the LPAR name to '##CPC##'.
- Substitute PRSMTNST and PRSMTNET values for the LPAR equivalent data elements.

This process permits a CPC level view of the physical shared IFL engine pool dispatch activity. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Since one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as physical engine dispatch hour equivalents. For example, three IFL engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVIFL - Avg Number of Active IFL Processors (Set to PRSMANSP – Avg IFL Sharable Processors)

This horizontal reference line, for the CPC level LPAR ##CPC##, shows the average number of physical processors in the shared IFL engine pool. While the data extract plots data element LPCAVIFL, which is the average number of logical CP engines assigned to an LPAR—for the ##CPC## chart, a user data step populates LPCAVIFL with the value for PRSMANSP—which is the average number of physical engines in the shared IFL engine pool. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like capacity provisioning and on/off capacity on demand can activate and deactivate physical processors—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

LPAR level charts

The LPAR level charts use standard HARLPC file data elements, to derive the dispatch variables:

Left Y-axis:

ENGDISPT – Total IFL Engine Dispatch (Set to LPCSNDTM – Total shared IFL Proc. Dispatch Time / 3600)

ENGDISPE – Effective IFL Engine Dispatch (Set to LPCSNETD – Total Shr IFL Proc. Effec. Dispatch Time / 3600)

The vertical bars, for the LPAR level charts, show the total and effective dispatch of the logical IFL processors assigned to the LPAR. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as logical engine dispatch hour equivalents. For example, three logical IFL engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVIFL - Avg Number of Active IFL Processors

This horizontal reference line, for the LPAR level charts, show the average number of logical IFL processors assigned to the LPAR. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like the Intelligent Resource Director (IRD) can activate and deactivate logical processors at the LPAR level—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

Query Defaults

- HARLPC DAYS timespan file: Cycles 01 and 02
- Filters:
 - LPCSNEDT or LPCSNDTM GT 0 or PRSMLPNM="PHYSICAL" (Non-zero dispatch times for IFL engines or the "PHYSICAL" LPAR)
 - LPCDEDCT EQ 0 ; (no dedicated LPARS)
 - PRSMTNSP GT 0 ; (at least one active physical IFL engine in shared engine pool)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

DATE

DATEPART(ENDTS)

CPCID

TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);

ENGDISPE

ROUND((LPCSNEDT/3600),.1) ;

ENGDISPA

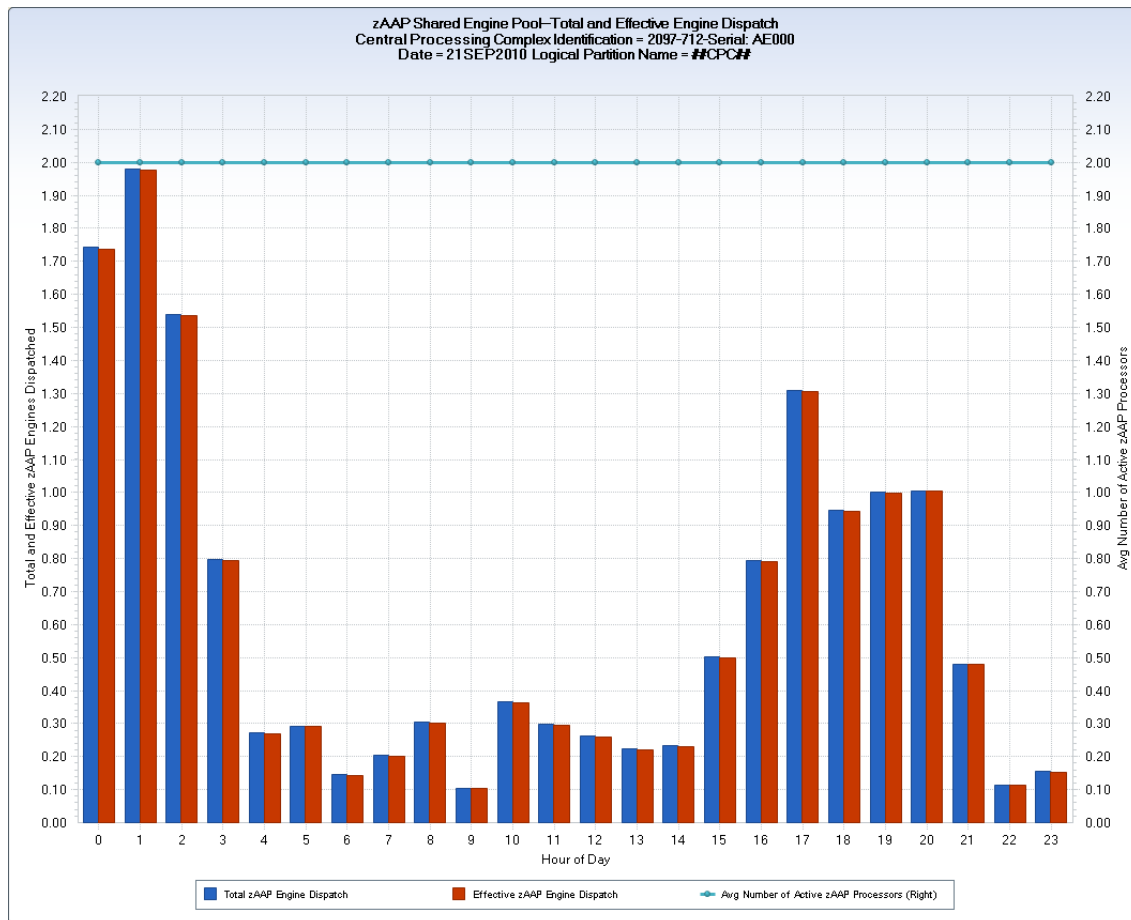
ROUND((LPCSNDTM/3600),.1) ;

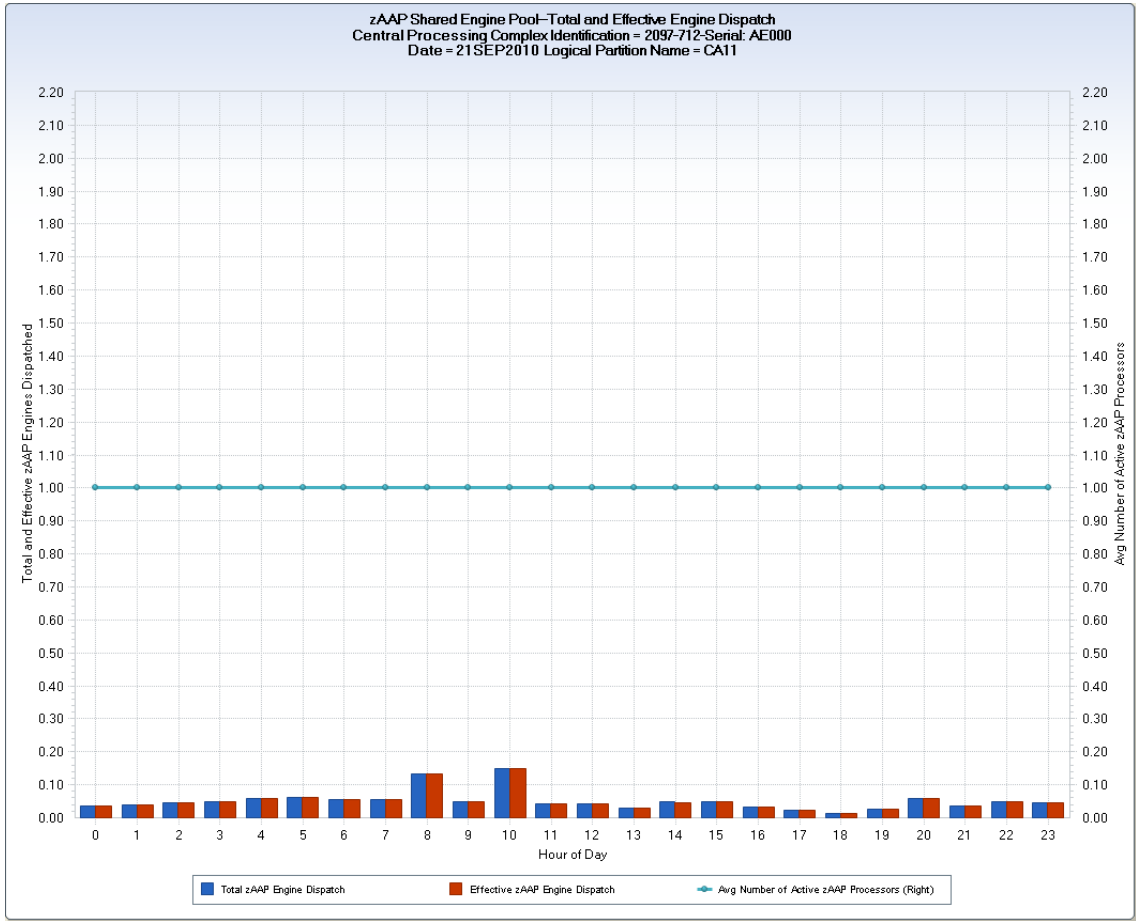
Modifications

The following modifications can be made to the RMFIFD query:

- You can comment out the user code in the initial file selection step that selects only yesterday's data. For example, you could read cycles 01-07 and generate charts for the past seven days.
- You can add filter statements to select only a single CPC, or set of CPCs using some of the identifying data elements shown in the CPCID derivation.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)

RMFZAD – CPC and LPAR Daily Shared zAAP Engine Dispatch by Hour





Overview

The RMFZAD query creates a single data extract CSV showing hourly CPC and LPAR dispatch percentages for the shared zAAP (zSeries Application Assist Processor) engine pool. For each Central Processing Complex (CPC) in the DAYS timespan HARLPC file, the query generates a CPC level chart. The chart uses vertical bars to show the total and effective dispatch percentage of the *physical* zAAP engines for each hour of the day—plus a horizontal reference line showing the total number of *physical* engines in the shared zAAP processor pool. The CPC level chart is followed by individual charts for each LPAR in the CPC, showing the dispatch percentages of the *logical* zAAP engines assigned to the LPAR. For the LPAR charts, the horizontal reference line shows the number of *logical* zAAP engines active for the LPAR.

The difference between total and effective dispatch represents the amount of the engine resource pool required for multiprocessing management overhead.

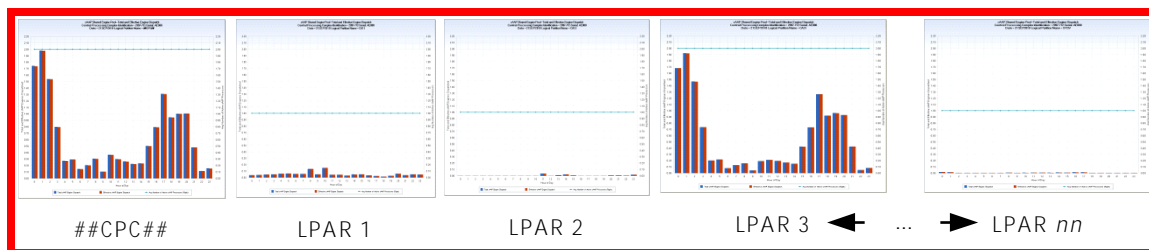
Uses

This query is used to monitor the overall shared zAAP pool engine utilization of each CPC at an hour granularity. The individual LPAR views show to what extent each LPAR is using the logical zAAP engines available for its use. Even though the CPC level and LPAR level charts use the same chart variables, the actual values charted are different.

Note: If a CPC has one or more zIIP engines and no zAAP engines, and has specified ZAAPZIIP=YES in SYS1.PARMLIB, zIIP engines process all specialty engine work—including work that would normally run on zAAP engines. In this environment, all SMF and RMF metrics treat zAAP and zIIP eligible workloads as zIIP eligible. No zAAP actual or eligible demand appears.

Data Extracts

CPC Total and Effective Daily zAAP Eng Dispatch



This query generates a single data extract, but data manipulation is used to convert the "PHYSICAL" LPAR to a CPC level view of shared zAAP pool engine dispatch, where the LPAR name is changed to "'##CPC##'". The following data extract description is presented in two parts: one describing the element values plotted for the CPC level chart, and the other describing the element values plotted for the individual LPARs.

CPC level chart

For the CPC level chart, CPC level PR/SM data element values are substituted for the LPAR data elements:

Left Y-axis

ENGDISPT – Total zAAP Engine Dispatch (Set to PRSMTZST – Total zAAP Dispatch Time – Shared / 3600)

ENGDISPE – Effective zAAP Engine Dispatch (Set to PRSMTZET – Total zAAP Dispatch Time - Effective / 3600)

This vertical bars, for the CPC level LPAR **##CPC##**, show the total and effective dispatch of the *physical* processors in the entire shared zAAP engine pool. For this chart, a user data step takes the following actions before the ENGDISPT and ENGDISPE derivations:

- Overwrite the 'PHYSICAL' LPAR observation.
- Change the LPAR name to '**##CPC##**'.
- Substitute PRSMTNST and PRSMTNET values for the LPAR equivalent data elements.

This process permits a CPC level view of the *physical* shared zAAP engine pool dispatch activity. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Since one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as physical engine dispatch hour equivalents. For example, three zAAP engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVZAP - Avg Number of Active zAAP Processors (Set to PRSMANSP – Avg zAAP Sharable Processors)

This horizontal reference line, for the CPC level LPAR **##CPC##**, shows the average number of physical processors in the shared zAAP engine pool. While the data extract plots data element LPCAVZAP, which is the average number of logical CP engines assigned to an LPAR—for the **##CPC##** chart, a user data step populates LPCAVZAP with the value for PRSMANSP—which is the average number of physical engines in the shared zAAP engine pool. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like capacity provisioning and on/off capacity on demand can activate and deactivate physical processors—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

LPAR level charts

The LPAR level charts use standard HARLPC file data elements, to derive the dispatch variables:

Left Y-axis

ENGDISPT – Total zAAP Engine Dispatch (Set to LPCSZDTM – Total Shared zAAP Proc. Dispatch Time / 3600)

ENGDISPE – Effective zAAP Engine Dispatch (Set to LPCSZEDT – Total Shr zAAP Proc. Effec. Dispatch Time / 3600)

The vertical bars, for the LPAR level charts, show the total and effective dispatch of the logical zAAP processors assigned to the LPAR. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as logical engine dispatch hour equivalents. For example, three logical zAAP engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVZAP - Avg Number of Active zAAP Processors

This horizontal reference line, for the LPAR level charts, show the average number of logical zAAP processors assigned to the LPAR. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like the Intelligent Resource Director (IRD) can activate and deactivate logical processors at the LPAR level—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

Query Defaults

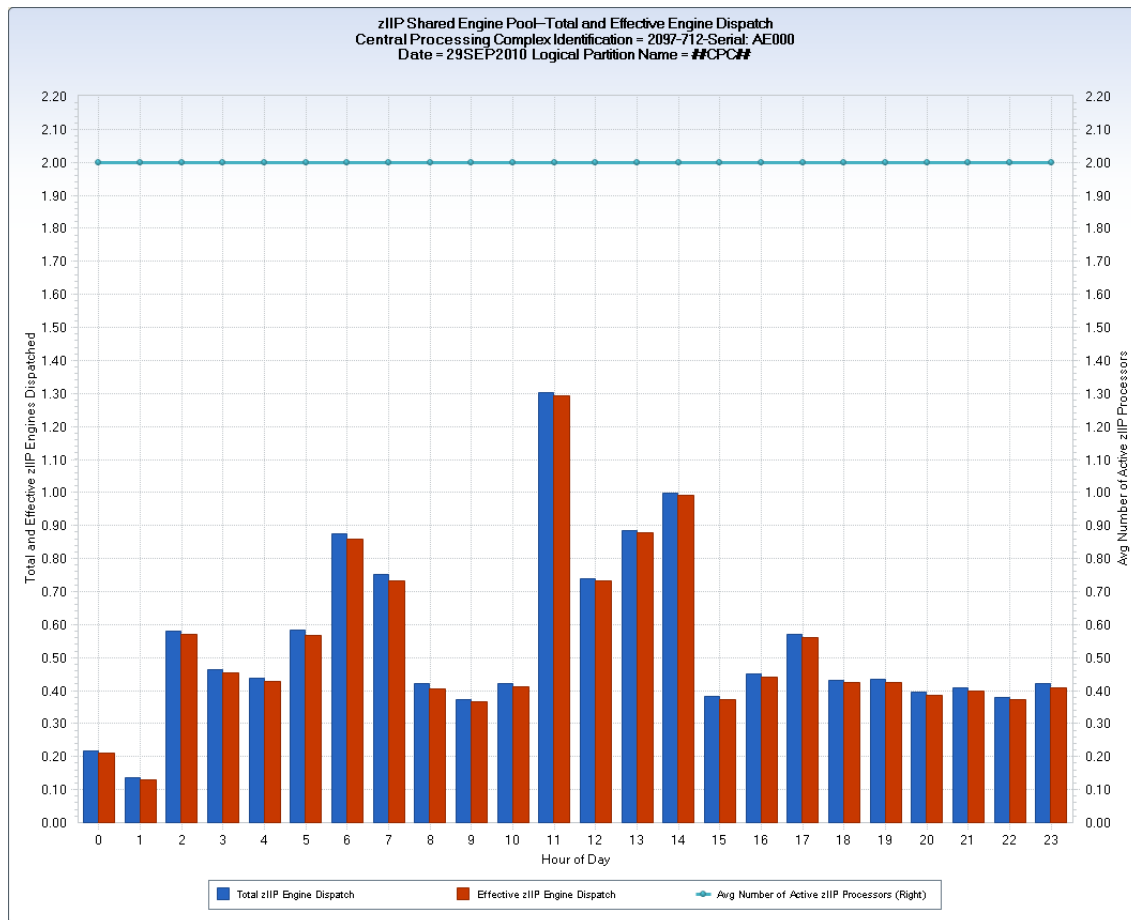
- HARLPC DAYS timespan file: Cycles 01 and 02
- Filters:
 - LPCSZEDT or LPCSZDTM GT 0 or PRSMLPNM="PHYSICAL" (Non-zero dispatch times for zAAP engines or the "PHYSICAL" LPAR)
 - LPCDEDCT EQ 0 ; (no dedicated LPARS)
 - PRSMTZSP GT 0 ; (at least one active physical zAAP engine in shared engine pool)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
DATEPART(ENDTS)
 - CPCID
TRIM(LPCMOD)||'-'||TRIM(CPCMODID)||'-Serial: '||SUBSTR(CPCSEQNB,12,5);
 - ENGDISPE
ROUND((LPCSZEDT/3600),.1) ;
 - ENGDISPA
ROUND((LPCSZDTM/3600),.1) ;

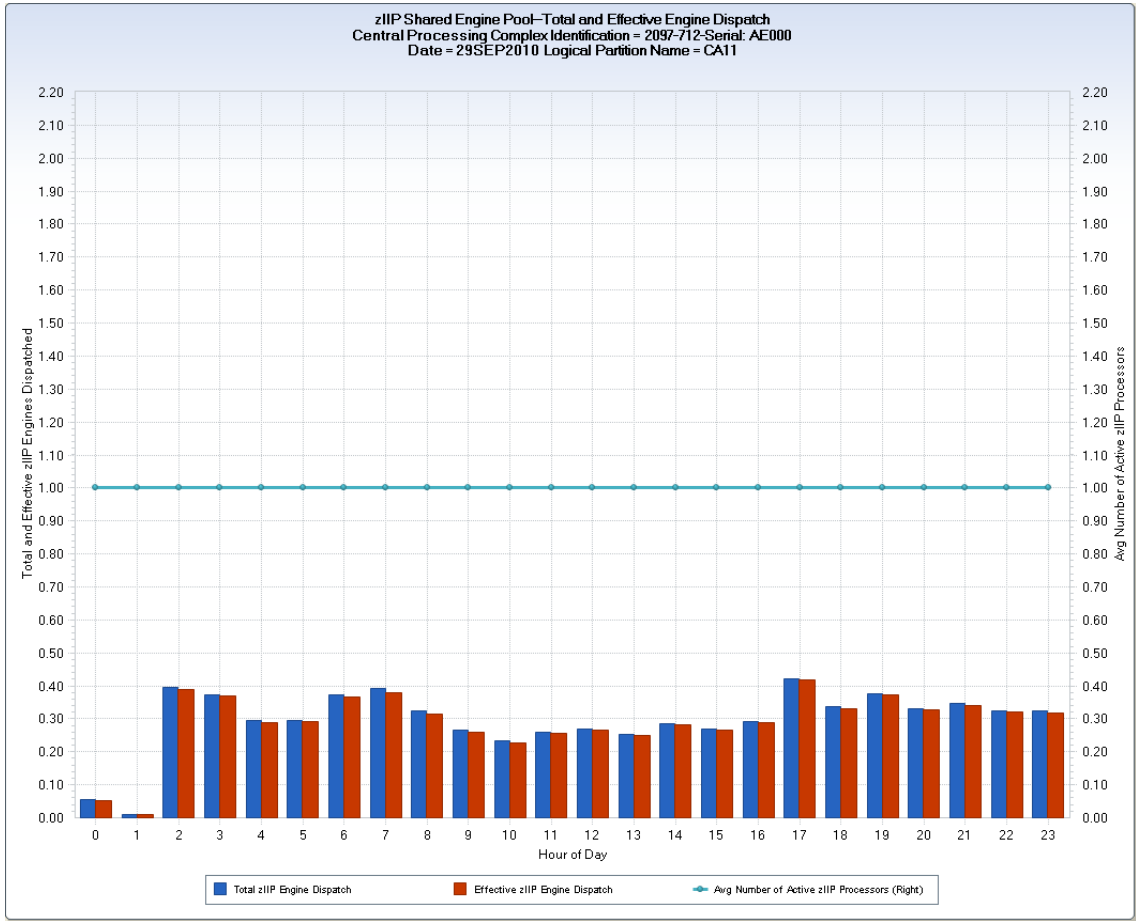
Modifications

The following modifications can be made to the RMFZAD query:

- You can comment out the user code in the initial file selection step that selects only yesterday's data. For example, you could read cycles 01-07 and generate charts for the past seven days.
- You can add filter statements to select only a single CPC, or set of CPCs using some of the identifying data elements shown in the CPCID derivation.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)

RMFZID – CPC and LPAR Daily Shared zIIP Engine Dispatch by Hour





Overview

The RMFZID query creates a single data extract CSV showing hourly CPC and LPAR dispatch percentages for the shared zIIP (zSeries Integrated Information Processor) engine pool. For each Central Processing Complex (CPC) in the DAYS timespan HARLPC file, the query generates a CPC level chart. The chart uses vertical bars to show the total and effective dispatch percentage of the *physical* zIIP engines for each hour of the day—plus a horizontal reference line showing the total number of *physical* engines in the shared zIIP processor pool. The CPC level chart is followed by individual charts for each LPAR in the CPC, showing the dispatch percentages of the *logical* zIIP engines assigned to the LPAR. For the LPAR charts, the horizontal reference line shows the number of *logical* zIIP engines active for the LPAR.

The difference between total and effective dispatch represents the amount of the engine resource pool required for multiprocessing management overhead.

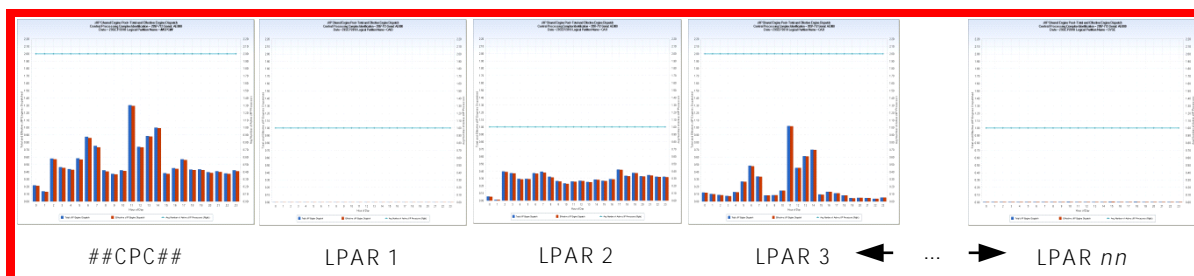
Uses

This query is used to monitor the overall shared zIIP pool engine utilization of each CPC at an hour granularity. The individual LPAR views show to what extent each LPAR is using the logical zIIP engines available for its use. Even though the CPC level and LPAR level charts use the same chart variables, the actual values charted are different.

Note: If a CPC has one or more zIIP engines and no zAAP engines, and has specified ZAAPZIIP=YES in SYS1.PARMLIB, zIIP engines process all specialty engine work—including work that would normally run on zAAP engines. In this environment, all SMF and RMF metrics treat zAAP and zIIP eligible workloads as zIIP eligible. No zAAP actual or eligible demand appears.

Data Extracts

CPC Total and Effective Daily zIIP Eng Dispatch



This query generates a single data extract, but data manipulation is used to convert the "PHYSICAL" LPAR to a CPC level view of shared zIIP pool engine dispatch, where the LPAR name is changed to "##CPC##". The following data extract description is presented in two parts: one describing the element values plotted for the CPC level chart, and the other describing the element values plotted for the individual LPARs.

CPC level chart

For the CPC level chart, CPC level PR/SM data element values are substituted for the LPAR data elements:

Left Y-axis

ENGDISPT – Total zIIP Engine Dispatch (Set to PRSMTSST – Total zIIP Dispatch Time – Shared / 3600)

ENGDISPE – Effective zIIP Engine Dispatch (Set to PRSMTSET – Total zIIP Dispatch Time - Effective / 3600)

The vertical bars, for the CPC level LPAR ##CPC##, show the total and effective dispatch of the physical processors in the entire shared zIIP engine pool. For this chart, a user data step takes the following actions before the ENGDISPT and ENGDISPE derivations:

- Overwrite the 'PHYSICAL' LPAR observation.
- Change the LPAR name to '##CPC##'.
- Substitute PRSMTNST and PRSMTNET values for the LPAR equivalent data elements.

This process permits a CPC level view of the physical shared zIIP engine pool dispatch activity. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Since one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as physical engine dispatch hour equivalents. For example, three zIIP engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVSUP - Avg Number of Active zIIP Processors (Set to PRSMANSP – Avg zIIP Sharable Processors)

This horizontal reference line, for the CPC level LPAR ##CPC##, shows the average number of physical processors in the shared zIIP engine pool. While the data extract plots data element LPCAVZAP, which is the average number of logical CP engines assigned to an LPAR—for the ##CPC## chart, a user data step populates LPCAVZAP with the value for PRSMANSP—which is the average number of physical engines in the shared zIIP engine pool. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like capacity provisioning and on/off capacity on demand can activate and deactivate physical processors—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

LPAR level charts

The LPAR level charts use standard HARLPC file data elements, to derive the dispatch variables:

Left Y-axis

ENGDISPT – Total zIIP Engine Dispatch (Set to LPCSSDTM – Total Shared zIIP Proc. Dispatch Time / 3600)

ENGDISPE – Effective zIIP Engine Dispatch (Set to LPCSSED – Total Shr zIIP Proc. Effec. Dispatch Time / 3600)

The vertical bars, for the LPAR level charts, show the total and effective dispatch of the logical zIIP processors assigned to the LPAR. The total and effective dispatch times are divided by 3600 to convert from seconds of dispatch to hours of dispatch. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the dispatch of the engine pool to be represented as logical engine dispatch hour equivalents. For example, three logical zIIP engines could, at full utilization, provide three hours of dispatch time in one hour.

Right Y-axis

LPCAVSUP - Avg Number of Active zIIP Processors

This horizontal reference line, for the LPAR level charts, show the average number of logical zIIP processors assigned to the LPAR. The actual dispatch engine equivalents, charted with the vertical bars, can reach, but should never exceed this line. IBM features like the Intelligent Resource Director (IRD) can activate and deactivate logical processors at the LPAR level—so this line may not reflect a constant value over the entire day.

X-axis

HOUR – Hour of Day

Query Defaults

- HARLPC DAYS timespan file: Cycles 01 and 02
- Filters:
 - LPCSSED or LPCSSDTM GT 0 or PRSMLPNM="PHYSICAL" (Non-zero dispatch times for zIIP engines or the "PHYSICAL" LPAR)
 - LPCDEDCT EQ 0 ; (no dedicated LPARS)
 - PRSMTSSP GT 0 ; (at least one active physical zIIP engine in shared engine pool)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART(ENDTS)
 - CPCID
 - TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
 - ENGDISPE
 - ROUND((LPCSSED/3600),.1) ;
 - ENGDISPA
 - ROUND((LPCSSDTM/3600),.1) ;

Modifications

The following modifications can be made to the RMFZID query:

- You can comment out the user code in the initial file selection step that selects only yesterday's data. For example, you could read cycles 01-07 and generate charts for the past seven days.
- You can add filter statements to select only a single CPC, or set of CPCs using some of the identifying data elements shown in the CPCID derivation.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)

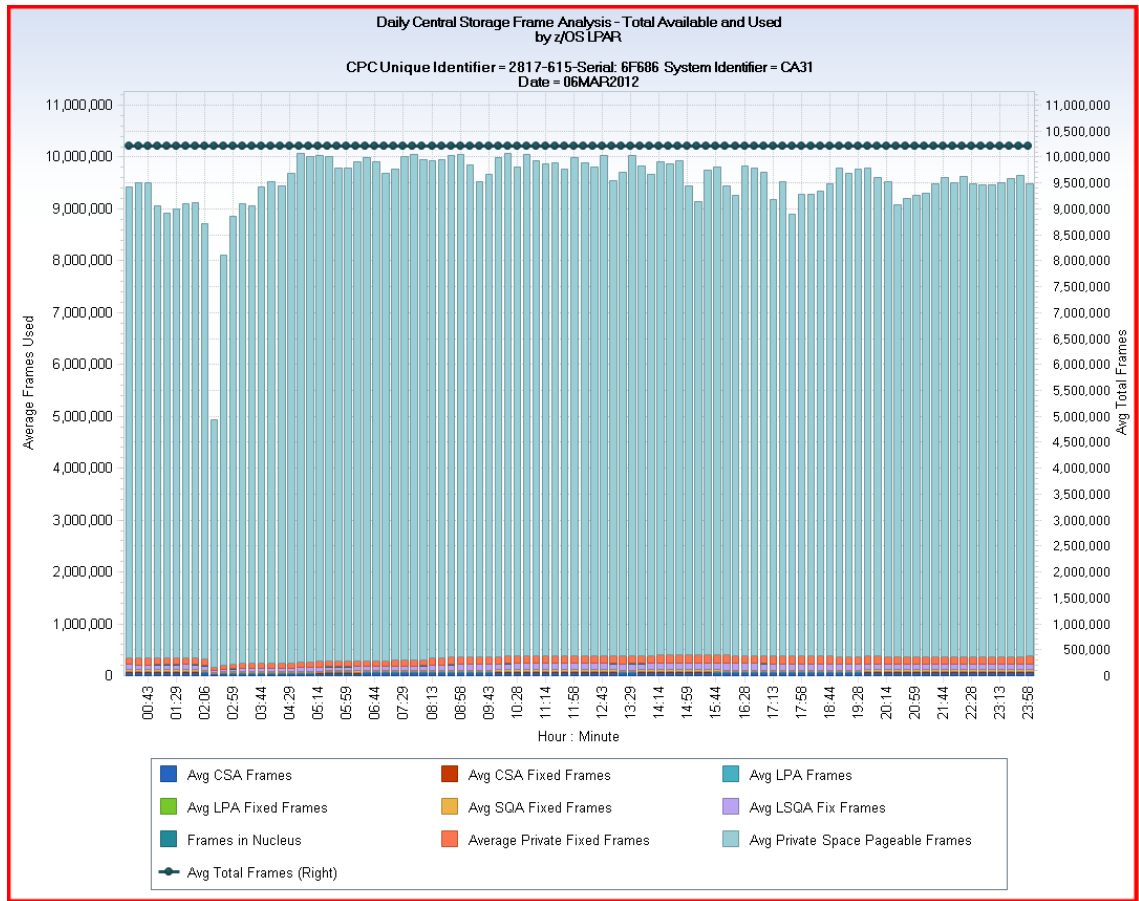
Mainframe CPC Memory Use and Paging

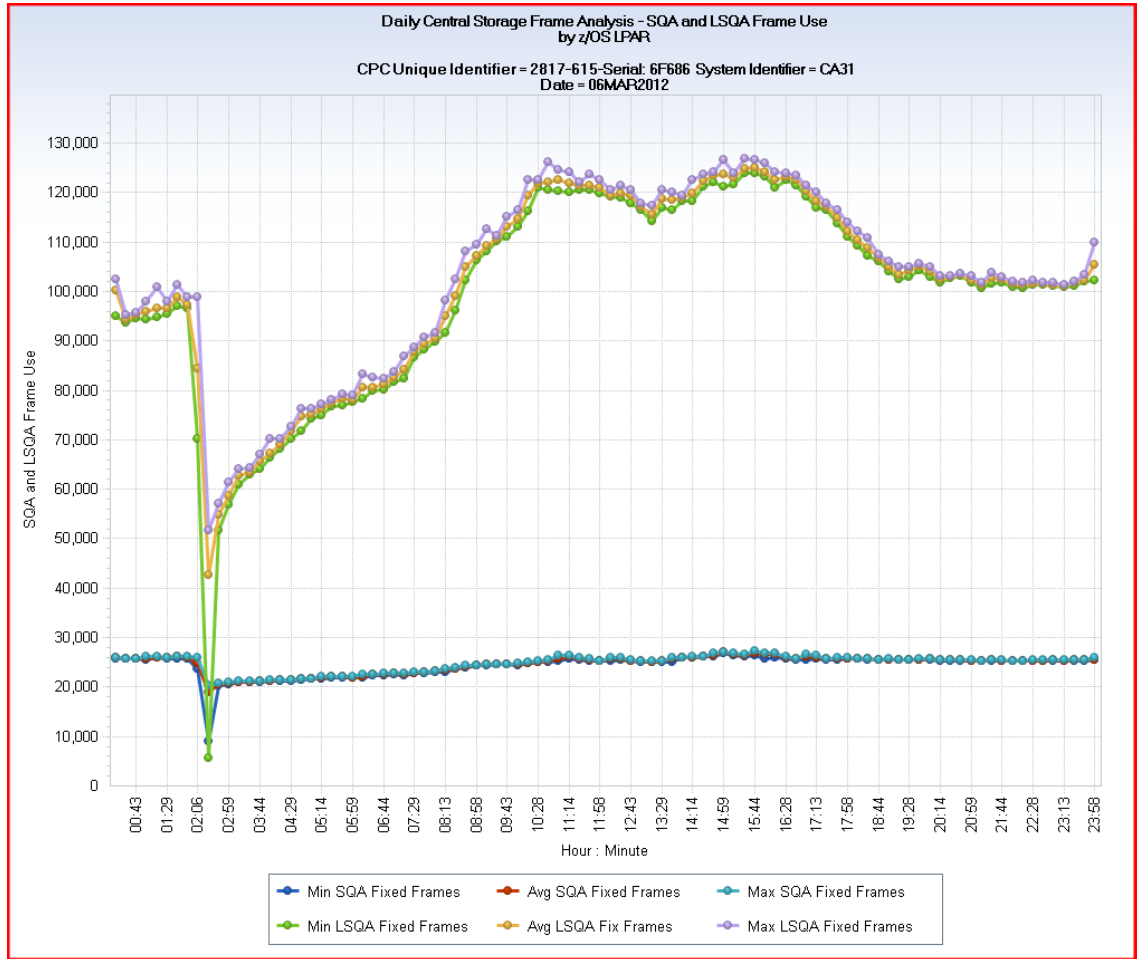
This section contains the following topics:

[RMFFRM - z/OS Central Storage Daily Frame Use by SYSID](#) (see page 157)

[RMFPAG - z/OS System Daily Paging Analysis](#) (see page 165)

RMFFRM – z/OS Central Storage Daily Frame Use by SYSID





Overview

The RMFFRM query shows how central storage 4k page frames are used for each z/OS LPAR in each Central Processing Complex (CPC). The query data extract step creates the following CSV output file:

- z/OS Central Storage Daily Page Frame Use

The data extract generates the following five chart views:

1. Daily Central Storage – Frames Available and Used
2. Daily Central Storage – Private Frame Use
3. Daily Central Storage – SQA and LSQA Frame Use
4. Daily Central Storage – LPA Frame Use
5. Daily Central Storage – CSA Frame Use

Chart view #1 displays the daily average central storage page frame usage by each category (private, SQA, CSA, etc.), for each RMF interval. This usage is shown using vertical bars, stacked by central storage frame category. A horizontal reference line shows the total central storage page frames available to the z/OS LPAR. The white space between the top of the bars and the horizontal reference line represents available central storage page frames. The first chart sample above is an example of chart view #1. It shows the daily page frame use for z/OS LPAR XAD1, running on an IBM z196 2817-615 CPC, at 15 minute intervals.

Chart view #2 displays the minimum, average, and maximum usage each interval for private fixed, and private pageable page frames using horizontal lines.

Chart view #3 displays the minimum, average, and maximum usage each interval for System Queue Area (SQA) and Local System Queue Area (LSQA) page frames using horizontal lines. The second chart sample above is an example of chart view #3, generated for the same LPAR displayed with the first chart. You can see that the SQA and LSQA frame use dropped dramatically at about 2 AM.

Chart view #4 displays the minimum, average, and maximum usage each interval for Link Pack Area (LPA) fixed and pageable page frames using horizontal lines.

Chart view #5 displays the minimum, average, and maximum usage each interval for Common Storage Area (CSA) fixed and pageable page frames using horizontal lines.

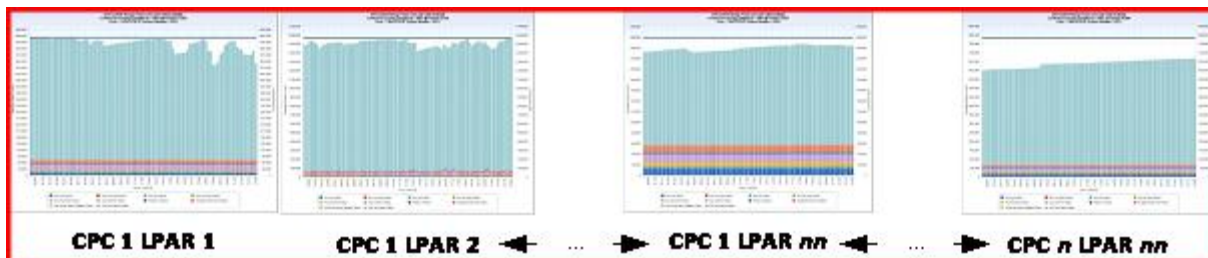
Uses

These charts show how z/OS LPARs use the real memory in central storage. Lack of sufficient real memory to handle workload requirements can result in excessive paging between central storage and auxiliary storage—which results in less than optimal performance. These charts help to understand how real memory is used, and can assist in identifying systems that are experiencing performance degradation due to insufficient central storage to satisfy workload requirements.

Data Extracts

[z/OS Central Storage Daily Page Frame Use](#)

VIEW 1 of 5: Daily Central Storage – Frames Available and Used



Left Y-axis

- PAGAVP – Avg CSA Frames
- PAGAVC – Avg CSA Fixed Frames
- PAGAVLP – Avg LPA Frames
- PAGAVLF – Avg LPA Fixed Frames
- PAGAVQ – Avg SQA Fixed Frames
- PAGAVLS – Avg LSQA Fixed Frames
- PAGFIN – Frames in Nucleus
- PAGAVR – Average Private Fixed Frames
- PAGAVS – Avg Private Space Pageable Frames

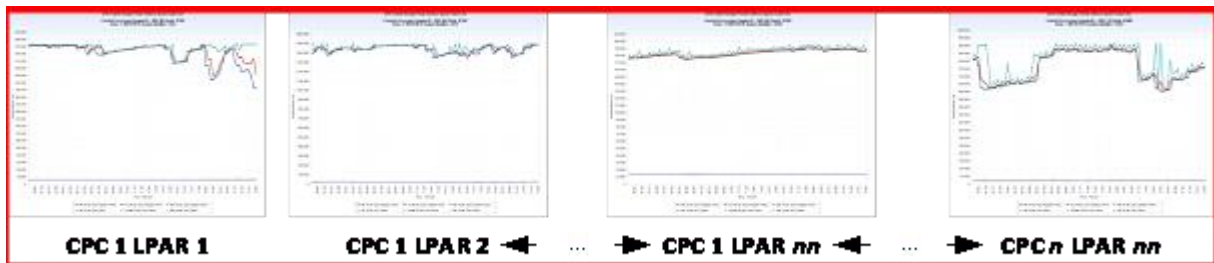
Right Y-axis

- PAGAVT – Avg Total Frames

X-axis

HHMM – Hour : Minute

VIEW 2 of 5: Daily Central Storage – Private Frames Use



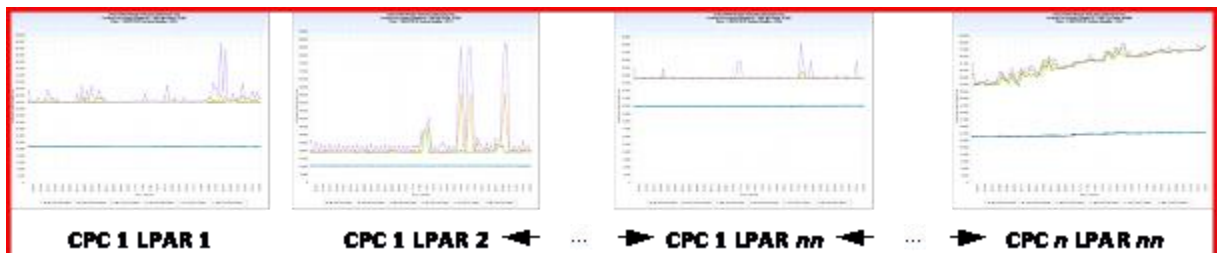
Left Y-axis

- PAGMNS – Min Private Space Pageable Frames
- PAGAVS – Avg Private Space Pageable Frames
- PAGMXS - Max Private Space Pageable Frames
- PAGMNR – Min Private Fixed Frames
- PAGAVR – Average Private Fixed Frames
- PAGMXR – Max Private Fixed Frames

X-axis

HHMM - Hour : Minute

VIEW 3 of 5: Daily Central Storage – SQA and LSQA Frame Use



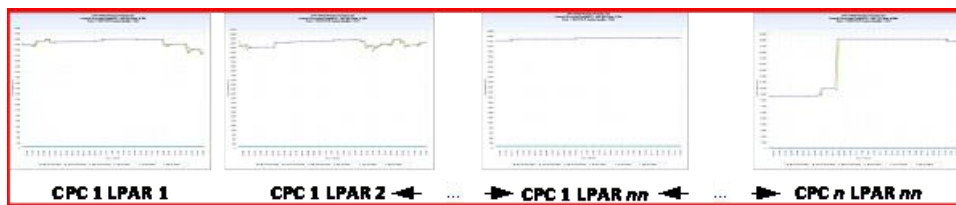
Left Y-axis

- PAGMNQ – Min SQA Fixed Frames
- PAGAVQ – Avg SQA Fixed Frames
- PAGMXQ – Max SQA Fixed Frames
- PAGMNLS - Min LSQA Frames
- PAGAVLS – Avg LSQA Frames
- PAGMXLS – Max LSQA Frames

X-axis

HHMM – Hour : Minute

VIEW 4 of 5: Daily Central Storage – LPA Frame Use



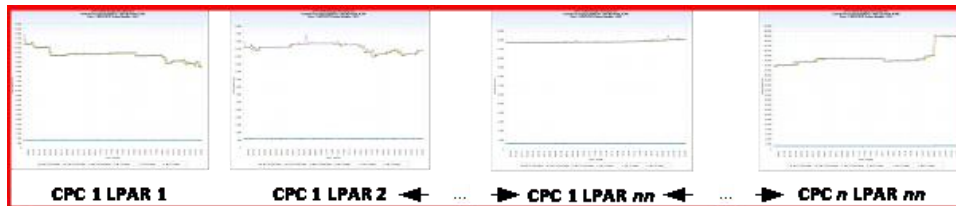
Left Y-axis

- PAGMNLF – Min LPA Fixed Frames
- PAGAVLF – Avg LPA Fixed Frames
- PAGMXLF – Max LPA Fixed Frames
- PAGMNLP – Min LPA Frames
- PAGAVLP – Avg LPA Frames
- PAGMXLP – Max LPA Frames

X-axis

HHMM – Hour : Minute

VIEW 5 of 5: Daily Central Storage – CSA Frame Use



Left Y-axis

- PAGMNC – Min CSA Fixed Frames
- PAGAVC – Avg CSA Fixed Frames
- PAGMXC – Max CSA Fixed Frames
- PAGMNP – Min CSA Frames
- PAGAVP – Avg CSA Frames
- PAGMXP – Max CSA Frames

X-axis

HHMM – Hour : Minute

Query Defaults

The following are RMFFRM Query Defaults:

- HARCPU DAYS timespan file: Cycle 01
- SCPPAG DETAIL timespan file: Cycles 01 - 02
- Filters: None
- User code excludes all but yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

CPCUID

```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

DATE

```
DATEPART(ENDTS);
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

DATEHHMM

```
PUT(DATEPART(ENDTS),YYMDDD10.) || ':' || HHMM ;
```

DATEHOUR

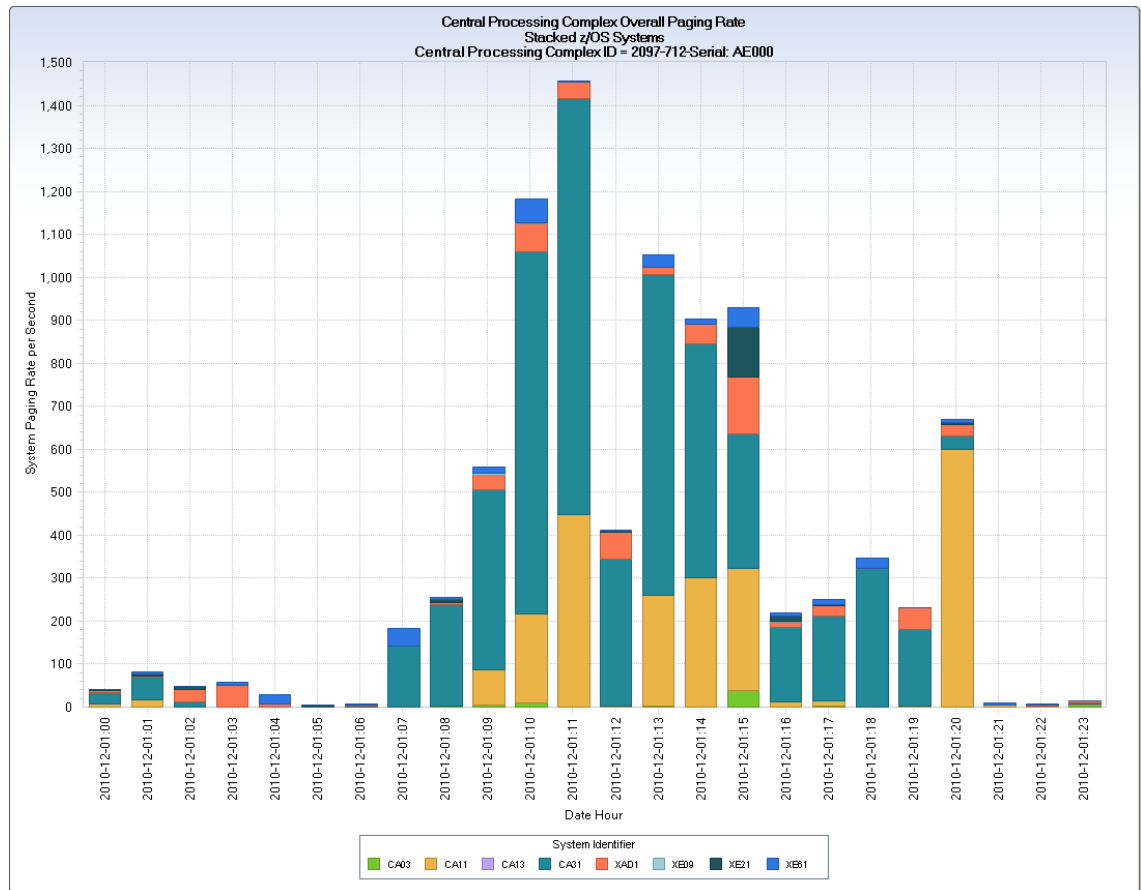
```
PUT(DATEPART(ENDTS),YYMDDD10.) || ':' || PUT(HOUR,Z2.) ;
```

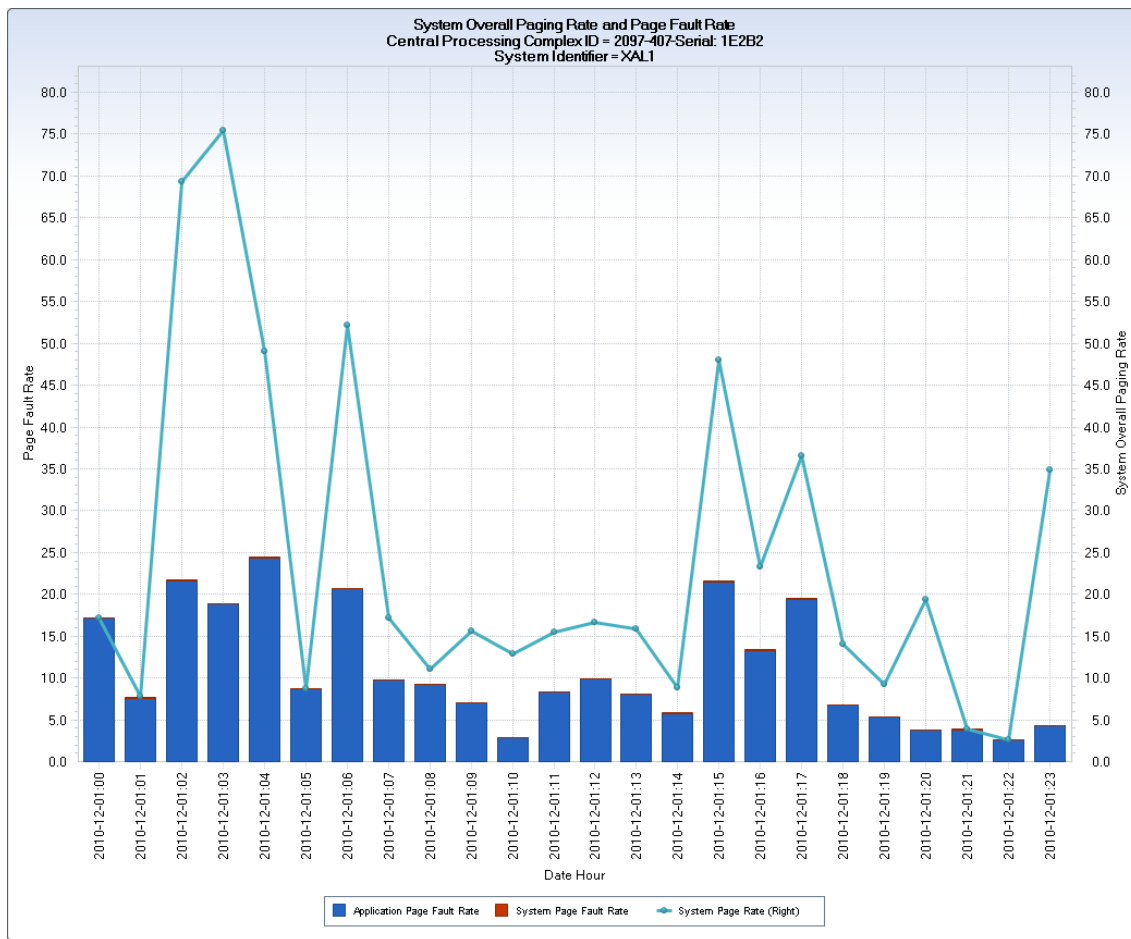
Modifications

The following modifications can be made to the RMFFRM query:

- If you want to see hourly, rather than interval level page frame use, you can use the SCPPAG DAYS timespan.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.
- Two additional temporal data elements, DATEHOUR - Date Hour, and DATEHHMM – Date Hour Minute, are provided for X axis use when creating single charts spanning multiple days. The key selection task for the data extract steps must be modified to substitute DATEHHMM for DATE and HHMM. Next, the user code in the initial file selection for the SCPPAG file must be commented out—it restricts processing to use yesterday's data only. Finally, the cycles selected for the SCPPAG file must be increased to read in the number of days to chart. Charts spanning multiple days to see changes in frame usage before and after changes to the amount of central storage assigned to z/OS LPARs.

RMFPAG – z/OS System Daily Paging Analysis





Overview

The RMFPAG query shows paging rates for z/OS LPARs in each Central Processing Complex (CPC). The query produces the following three CSV file data extracts:

- z/OS CPC Daily Paging Rate
- z/OS System Daily Page Fault Rate V1
- z/OS System Daily Page Fault Rate V2

The z/OS CPC Daily Paging Rate first data extract shows the hourly CPC paging rate of all z/OS LPARs in a single chart, with the contribution with the contribution of each z/OS system shown using stacked vertical bars. The first chart sample above is an example of the charts produced by this data extract. Here you see the hourly CPC paging rate incurred for the z/OS LPARs running on an IBM z10 2097-712 CPC.

The second data extract, z/OS System Daily Page Fault Rate V1, produces daily charts for each z/OS LPAR that show the hourly system, application, and overall system page rates. The system and application page fault rates are shown using stacked vertical bars, and the overall system page rate is shown using a horizontal line. The second chart sample above is an example of the charts produced by this data extract. Here you see the hourly system and application page fault rates, and the overall system paging rate for z/OS LPAR XAL1 running on an IBM z10 2097-712 CPC.

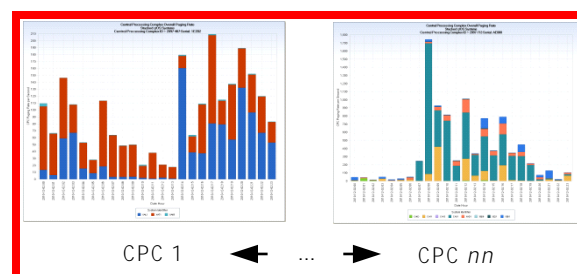
The third data extract, z/OS System Daily Page Fault Rate V2, produces daily charts for each z/OS LPAR that show the same information as the second data extract, but in a different way. The overall system page rate is shown using vertical bars, and the system and application page fault rates are displayed using horizontal lines.

Uses

Paging is essential and normal for the z/OS operating system. High per-second paging rates can indicate insufficient CPC central storage. Page faults result in workload delays. A program must wait for a page to be moved from auxiliary storage to central storage to continue executing. High page fault rates at the LPAR level can indicate insufficient central storage allocation for the LPAR.

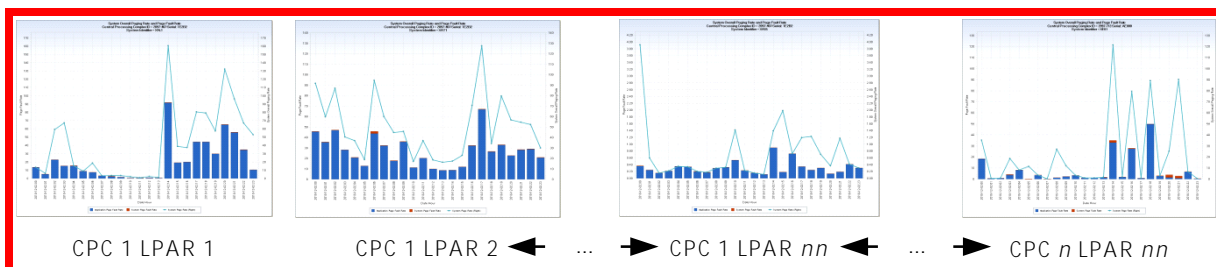
Data Extracts

z/OS CPC Daily Paging Rate



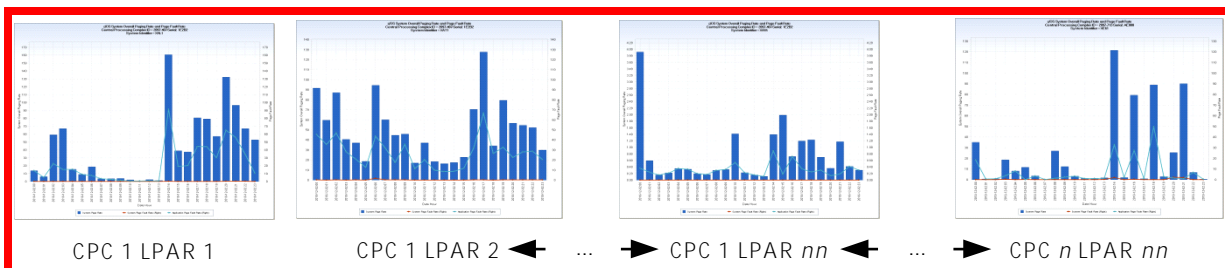
Left Y-axis
 PAGPSSPC – System Page Rate
 X-axis
 DATEHOUR – Date Hour

z/OS System Daily Page Rate and Faults V1



Left Y-axis
 PAGPSAPF – Application Page Fault Rate
 PAGPSSPF – System Page Fault Rate
 Right Y-axis
 PAGPSSPC – System Page Rate
 X-axis
 DATEHOUR – Date Hour

z/OS System Daily Page Rates and Faults V2



Left Y-axis
 PAGPSSPC – System Page Rate
 Right Y-axis
 PAGPSAPF – Application Page Fault Rate
 PAGPSSPF – System Page Fault Rate
 X-axis
 DATEHOUR – Date Hour

Query Defaults

The following are RMFPAG Query Defaults:

- HARCPU DAYS timespan file: Cycle 01
- SCPPAG DAYS timespan file: Cycles 01 - 02
- Filters: None
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

CPCID

```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

DATE

```
DATEPART(ENDTS);
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

Modifications

The following modifications can be made to the RMFPAG query:

- If you want to see interval, rather than hourly level paging and page fault statistics, you can use the HARCPU and SCPPAG DETAIL timespan.
- You can increase the number of cycles to show more than one day's data on a single chart, but comment out the user code for the SCPPAG initial file selection step that selects only yesterday's data.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.
- Various time-based derived data elements—HHMM, DATE, DATEHHMM, and DATEHOUR (default)—can be used to display the data in different ways. For example, if you want to display charts for multiple days, with each chart showing one day's activity, you can insert DATE and substitute HHMM for DATEHOUR in the data extract step KEY selection. The DATE appears in the chart title, and HHMM (Hour Minute) appears on the X-axis.
- The CSVs for the data extract steps include numerous other page rate and page count related data elements. These elements can be plotted if desired by editing the view and selecting the appropriate data elements for charting.

Mainframe CPC Specialty Engine Focus

This section contains the following topics:

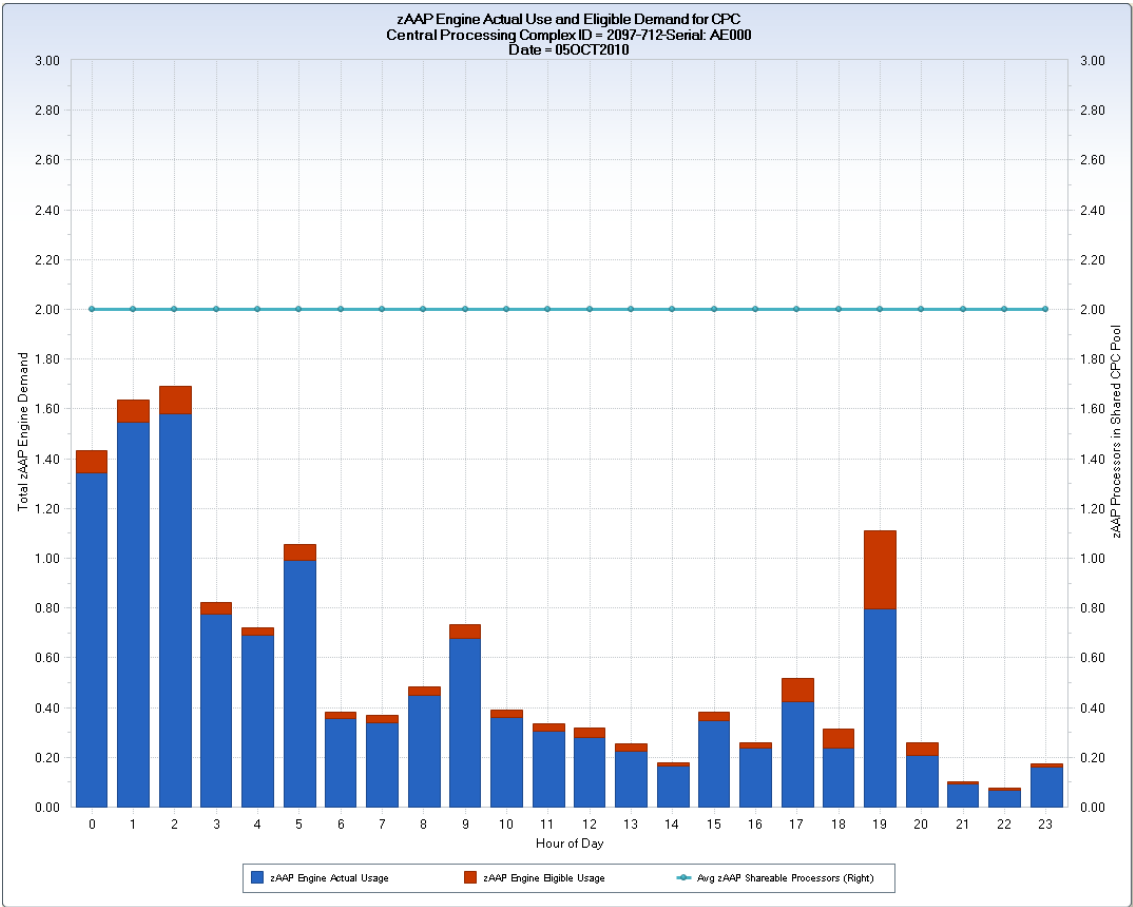
[RMFZAC - Daily zAAP Engine Use and Demand by CPC and SYSID](#) (see page 171)

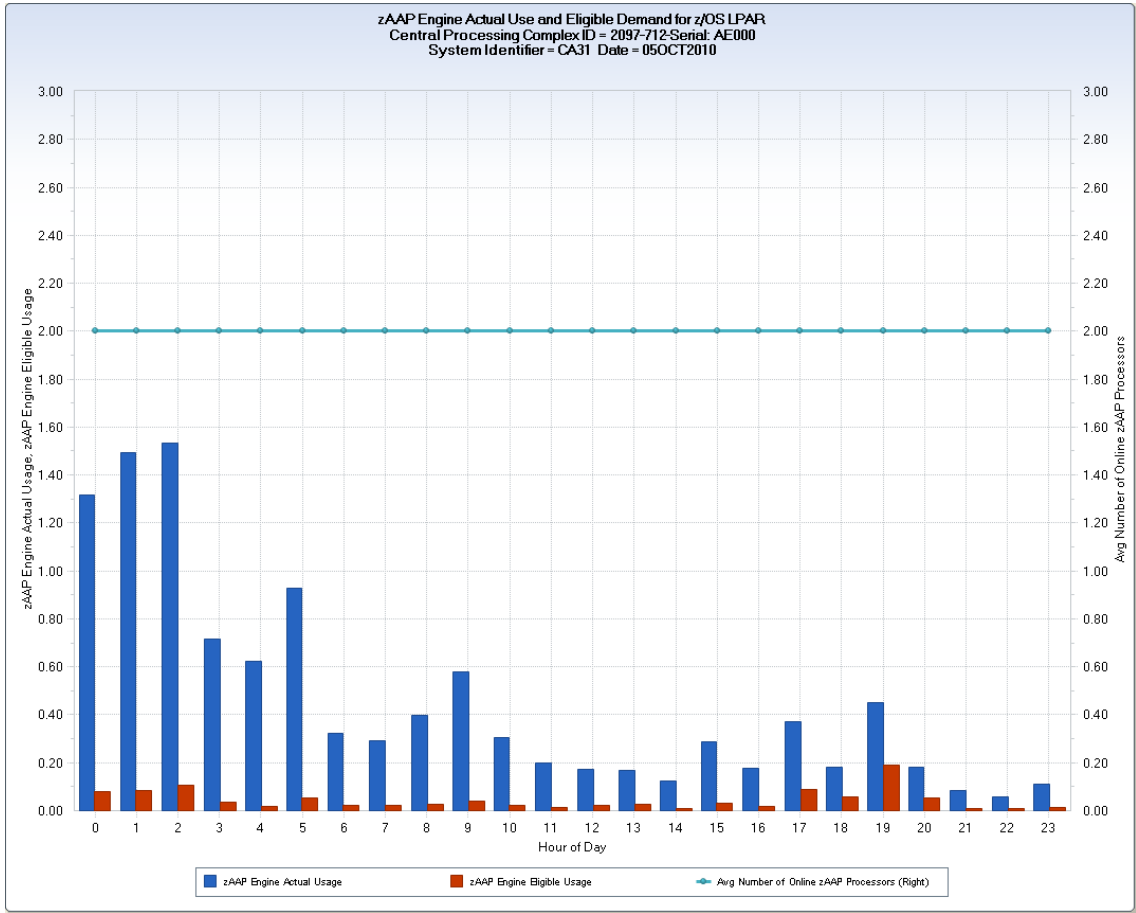
[RMFZAS - Daily zAAP Engine Use and Demand by Service Class](#) (see page 176)

[RMFZIC - Daily zIIP Engine Use and Demand by CPC and SYSID](#) (see page 180)

[RMFZIS - Daily zIIP Engine Use and Demand by Service Class](#) (see page 185)

RMFZAC – Daily zAAP Engine Use and Demand by CPC and SYSID





Overview

The RMFZAC query charts the hourly actual zAAP engine use, and zAAP engine eligible use for both the Central Processing Complex (CPC) and the z/OS LPARs using the shared engine pools of a CPC. Only PR/SM managed z/OS LPARs are charted. Non-z/OS LPARs and z/OS LPARs managed by VM are excluded from analysis because the data containing zAAP engine demand is not available.

The example shows the CPC level chart (top) for an IBM z10 2097-712 CPC, and one of the individual LPAR charts for an LPAR defined to that CPC.

The CPC level chart shows total zAAP engine demand, with the actual use and eligible demand stacked. The horizontal reference line in the CPC level chart shows the number of *physical* zAAP engines in the CPC shared engine pool.

The LPAR level chart shows zAAP engine use and zAAP engine demand as separate vertical bars for each hour of the day. The horizontal reference line in the LPAR level chart shows the number of *logical* zAAP engines assigned to the LPAR. One LPAR level chart is produced for each LPAR that executed zAAP eligible work.

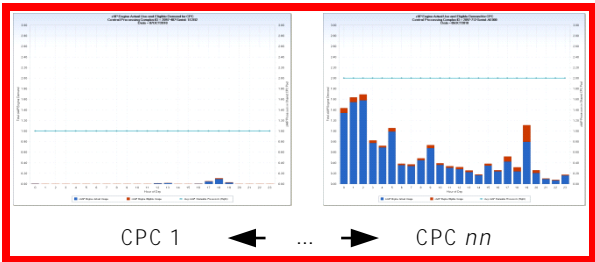
Uses

This query is used to determine if the zAAP engines in a CPC are sufficient to handle the zAAP eligible workload running on z/OS LPARs. Even if no zAAP engines are present in a CPC, the charts show z/OS LPAR zAAP eligible work that executed on CP engines. zAAP engines are less expensive than CP engines. Reducing CP engines in a CPC can reduce software licensing fees. In general, if significant zAAP eligible work is executing on CP engines, it is cost effective to add zAAP engines to the CPC.

Note: If a CPC has one or more zIIP engines and no zAAP engines, and has specified ZAAPZIIP=YES in SYS1.PARMLIB, zIIP engines process all specialty engine work—including work that would normally run on zAAP engines. In this environment, all SMF and RMF metrics treat zAAP and zIIP eligible workloads as zIIP eligible. No zAAP actual or eligible demand appears.

Data Extracts

Daily CPC zAAP Eng Use and Demand



Left Y-axis

ZAPENGA – zAAP Engine Actual Usage

ZAPENGE – zAAP Engine Eligible Usage

This extract shows the total z/OS LPAR zAAP usage at the CPC level. The vertical bars show total zAAP actual and zAAP eligible CPU time for all PR/SM managed z/OS LPARs at the CPC level. The CPU times are divided by 3600 to convert from seconds to hours. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the CPU demand to be represented as physical engine dispatch hour equivalents. For example, two zAAP engines could, at full utilization, provide two hours of CPU time in one hour.

Right Y-axis

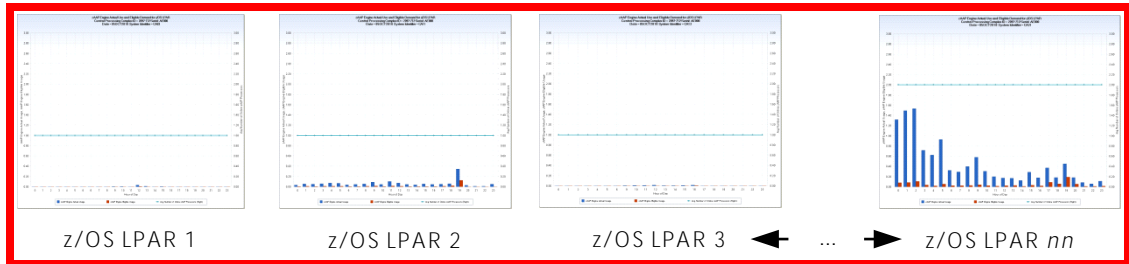
PRSMZSP - Avg zAAP Shareable Processors

Average number of physical zAAP engines online in the shared zAAP engine pool.

X-axis

HOUR – Hour of Day

Daily z/OS System zAAP Eng Use and Demand



Left Y-axis

ZAPENGA – zAAP Engine Actual Usage

ZAPENGE – zAAP Engine Eligible Usage

This extract shows the z/OS LPAR zAAP usage for the CPCs charted in the CPC level data extract. The vertical bars show total zAAP actual and zAAP eligible CPU time for all PR/SM managed z/OS LPARs at the LPAR level. The CPU times are divided by 3600 to convert from seconds to hours. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the CPU demand to be represented as physical engine dispatch hour equivalents. For example, two zAAP engines could, at full utilization, provide two hours of CPU time in one hour.

Right Y-axis

CPUAVOZP - Avg Number of Online zAAP Processors

Average number of logical zAAP processors online to the LPAR each hour.

X-axis

HOUR – Hour of Day

Query Defaults

- HARCPU DAYS timespan file: Cycles 01 -02
- WLMSEC DAYS timespan file: Cycles 01-02
- Filters: PRSMUSCT GT 0 (only PR/SM managed LPARs)
- User code selects yesterdays data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - CPCID


```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```
 - ZAPENGA


```
ROUND((SECZAPTM/3600),.001);
```
 - ZAPENGE

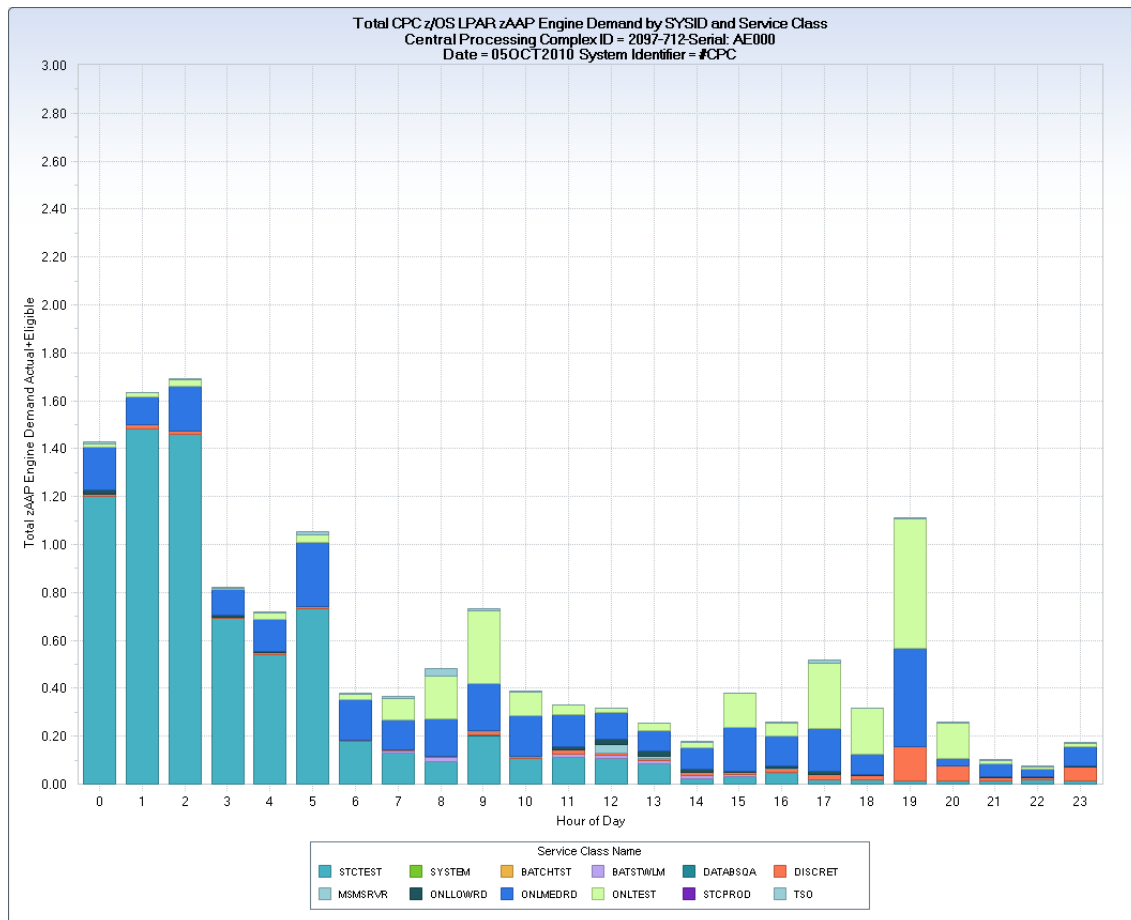

```
ROUND((SECZAPCT/3600),.001) ;
```

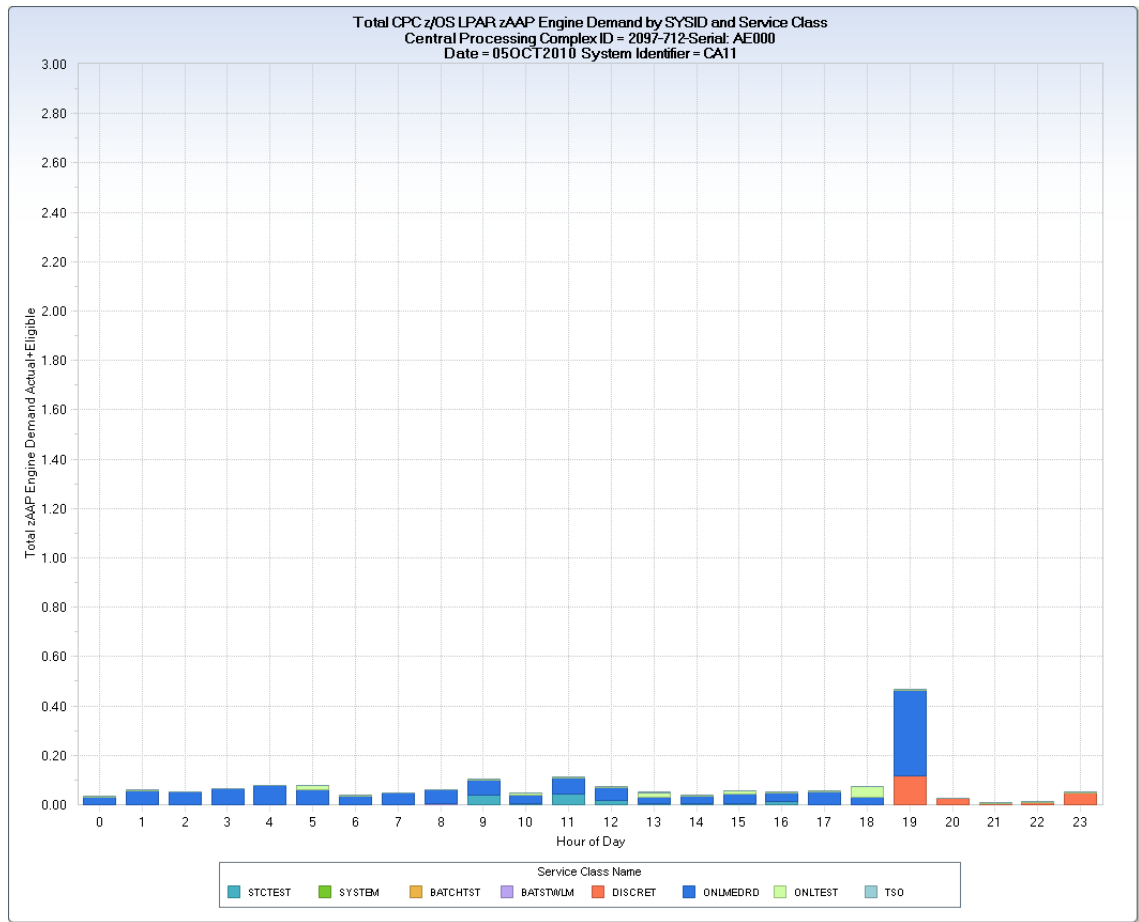
Modifications

The following modifications can be made to the RMFZAC query:

- You can increase the number HARCPU and WLMSEC DAYS timespan cycles input, but comment out the user code in each file selection step that selects yesterday's data.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.

RMFZAS – Daily zAAP Engine Use and Demand by Service Class





Overview

The RMFZAS query charts the hourly total zAAP engine demand (actual zAAP engine use plus zAAP engine eligible use), by WLM Report Class, for the Central Processing Complex (CPC) and individual z/OS LPARs using the shared engine pools of a CPC. Only PR/SM managed z/OS LPARs contribute to these charts because the data containing zAAP engine demand, extracted from the WLMSEC Report Class Resource Consumption file, is derived from RMF type 72-3 records, which only record zAAP engine demand for PR/SM managed z/OS LPARs.

The example shows the CPC level chart (top) for an IBM z10 2097-712 CPC, and one of the individual LPAR charts for an LPAR defined to that CPC.

Both the CPC level and SYSID level charts show total hourly zAAP engine demand, using vertical bars, stacked by WLM Report Class.

Note: A fake Report class named '*NO ZAP*' is created, showing no usage, for hours where there was no zAAP engine demand. This class prevents charts from being produced where hours are missing from the X-axis.

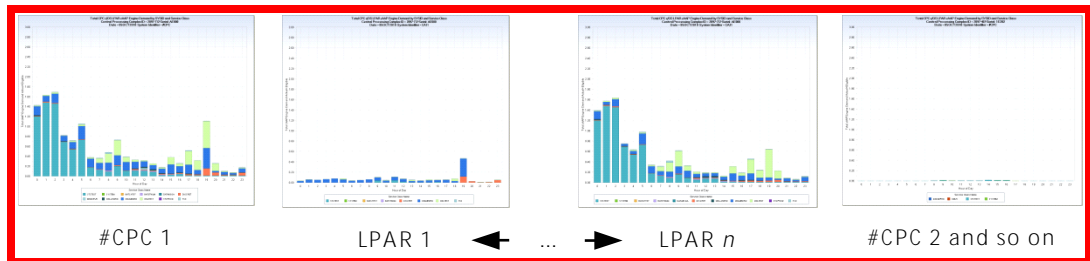
Uses

This query is used to examine the WLM Report Classes that execute zAAP engine eligible work. If zAAP usage appears for an LPAR that does not have zAAP engines assigned, then the work is actually executing on expensive CP engines. You may want to reconfigure the LPAR with one or more zAAP engines, or reroute the Report Class workload to execute on an LPAR with zAAP engines. The charts as delivered show total demand—which is the sum of zAAP engine actual use plus any zAAP engine eligible use that executed on CP engines. The two components—actual use and eligible use are available for charting, and can be selected as the Y-axis chart variables if desired. Additionally, you can review the RMFZAC - Daily zAAP Engine Use and Demand by CPC and SYSID query charts. These charts show the breakdown of actual and eligible zAAP engine demand in a single chart.

Note: If a CPC has one or more zIIP engines and no zAAP engines, and has specified ZAAPZIIP=YES in SYS1.PARMLIB, zIIP engines process all specialty engine work—including work that would normally run on zAAP engines. In this environment, all SMF and RMF metrics treat zAAP and zIIP eligible workloads as zIIP eligible. No zAAP actual or eligible demand appears.

Data Extracts

Daily Total zAAP Demand by Service Class



Left Y-axis

ZAPENGT – Total zAAP Engine Demand Actual+Eligible

The vertical bars show total zAAP usage (zAAP actual+eligible CPU time) at the CPC level and for each PR/SM managed z/OS LPAR. For each CPC, the CPC level chart is produced first, with a SYSID value of '#CPC', and is followed by the PR/SM managed z/OS LPARs for that CPC. The CPU times are divided by 3600 to convert from seconds to hours to create the chart variable ZAPENGT. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the zAAP engine demand to be represented as physical engine hour equivalents. For example, two zAAP engines could, at full utilization, provide two hours of CPU time in one hour.

X-axis

HOUR – Hour of Day

Query Defaults

- HARCPU DAYS timespan file: Cycles 01 -02
- WLMSEC DAYS timespan file: Cycles 01-02
- Filters: PRSMUSCT GT 0 (only PR/SM managed LPARs)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

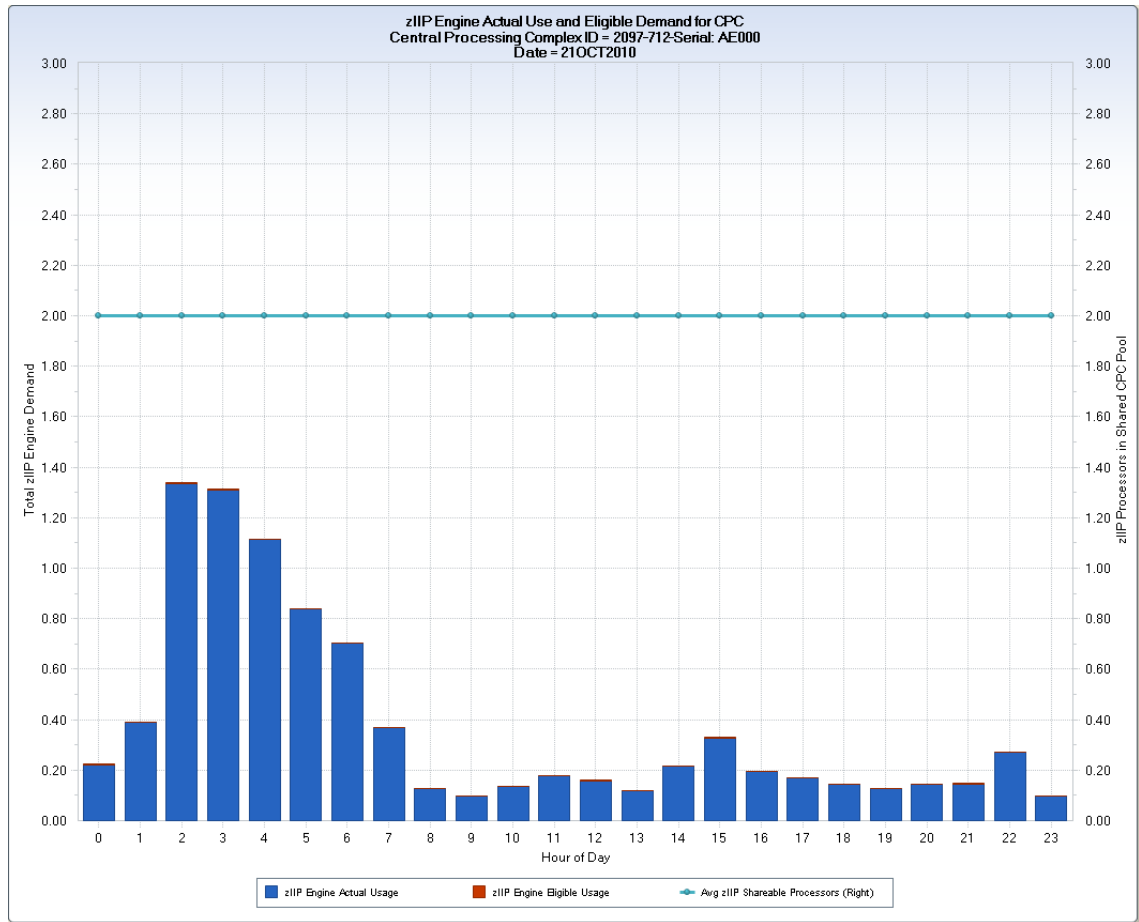

```
CPCID
      TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
ZAPENGT
      SUM(ZAPENGA,ZAPENGE) ;
ZAPENGA
      ROUND((SECZAPTM/3600),.001);
ZAPENGE
      ROUND((SECZAPCT/3600),.001) ;
```

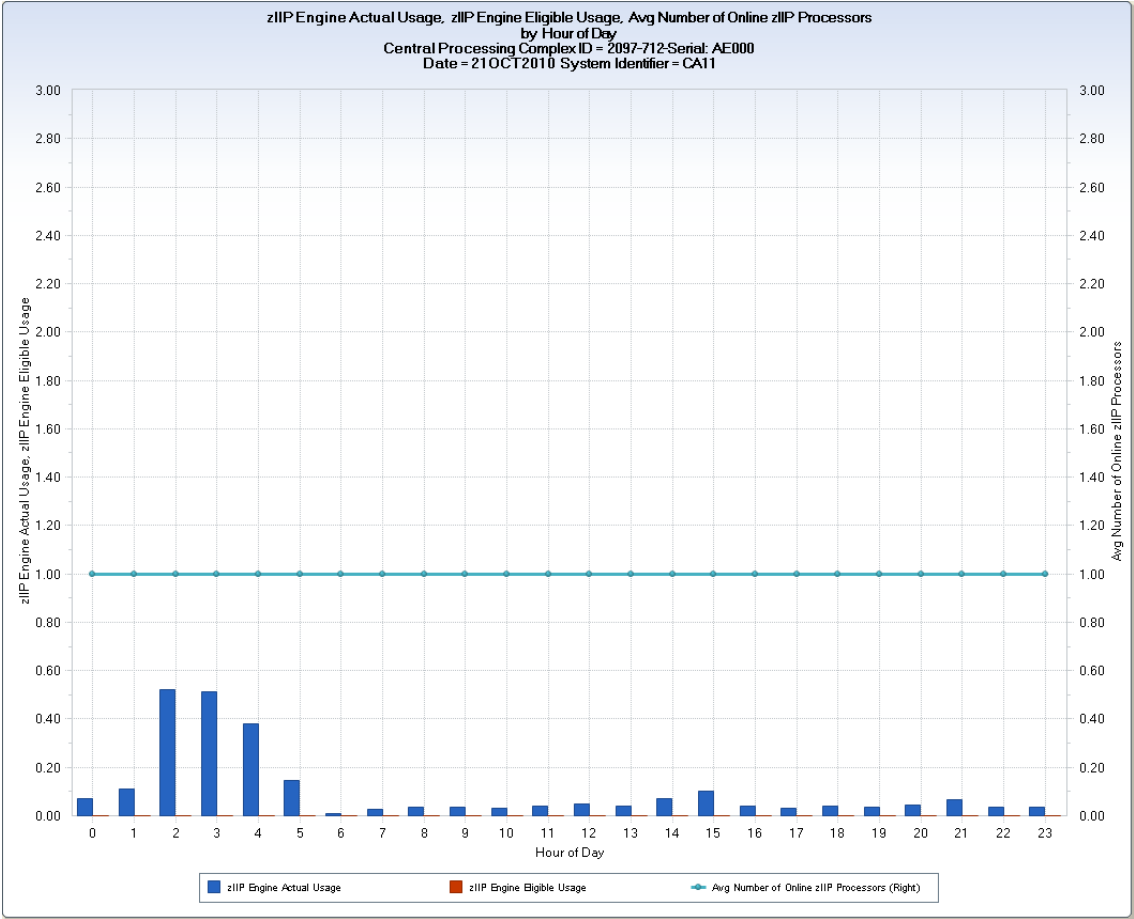
Modifications

The following modifications can be made to the RMFZAS query:

- You can increase the number HARCPU and WLMSEC DAYS timespan cycles input, but comment out the user code in each file selection step that selects yesterday's data.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.
- If you want to generate these charts to see either the zAAP engine actual use or zAAP engine eligible use, change the Y-axis variable charted. The default is ZAPENGT, but ZAPENGA or ZAPENGE can be selected.

RMFZIC – Daily zIIP Engine Use and Demand by CPC and SYSID





Overview

The RMFZIC query charts the hourly actual zIIP engine use and zIIP engine eligible use for both the Central Processing Complex (CPC) and the z/OS LPARs using the shared engine pools of a CPC. Only PR/SM managed z/OS LPARs are charted. Non-z/OS LPARs and z/OS LPARs managed by VM are excluded from analysis because the data containing zIIP engine demand is not available.

The example shows the following charts:

- CPC level chart (top) for an IBM z10 2097-712 CPC
- One of the individual LPAR charts for an LPAR defined to that CPC

The CPC level chart shows total zIIP engine demand, with the actual use and eligible demand stacked. The horizontal reference line in the CPC level chart shows the number of *physical* zIIP engines in the CPC shared engine pool.

The LPAR level chart shows zIIP engine use and zIIP engine demand as separate vertical bars for each hour of the day. The horizontal reference line in the LPAR level chart shows the number of *logical* zIIP engines assigned to the LPAR. One LPAR level chart is produced for each LPAR that executed zIIP eligible work.

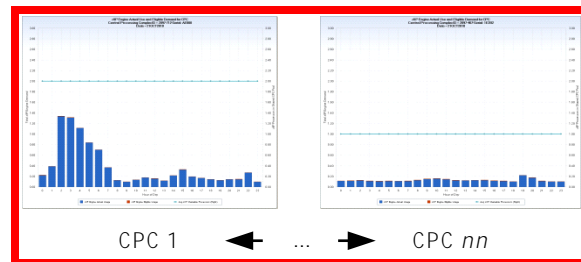
Uses

This query is used to determine if the zIIP engines in a CPC are sufficient to handle the zIIP eligible workload running on z/OS LPARs. Even if no zIIP engines are present in a CPC, the charts show z/OS LPAR zIIP eligible work that executed on CP engines. zIIP engines are less expensive than CP engines. Reducing CP engines in a CPC can reduce software licensing fees. In general, if significant zIIP eligible work is executing on CP engines, it is cost effective to add zIIP engines to the CPC.

Note: If a CPC has one or more zIIP engines and no zAAP engines, and has specified ZAAPZIIP=YES in SYS1.PARMLIB, zIIP engines process all specialty engine work—including work that would normally run on zAAP engines. In this environment, all SMF and RMF metrics treat zAAP and zIIP eligible workloads as zIIP eligible. No zAAP actual or eligible demand appears.

Data Extracts

Daily CPC zIIP Eng Use and Demand



Left Y-axis

ZIPENGA – zIIP Engine Actual Usage

ZIPENGE – zIIP Engine Eligible Usage

This extract shows the total z/OS LPAR zIIP usage at the CPC level. The vertical bars show total zIIP actual and zIIP eligible CPU time for all PR/SM managed z/OS LPARs at the CPC level. The CPU times are divided by 3600 to convert from seconds to hours. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the CPU demand to be represented as physical engine dispatch hour equivalents. For example, two zIIP engines could, at full utilization, provide two hours of CPU time in one hour.

Right Y-axis

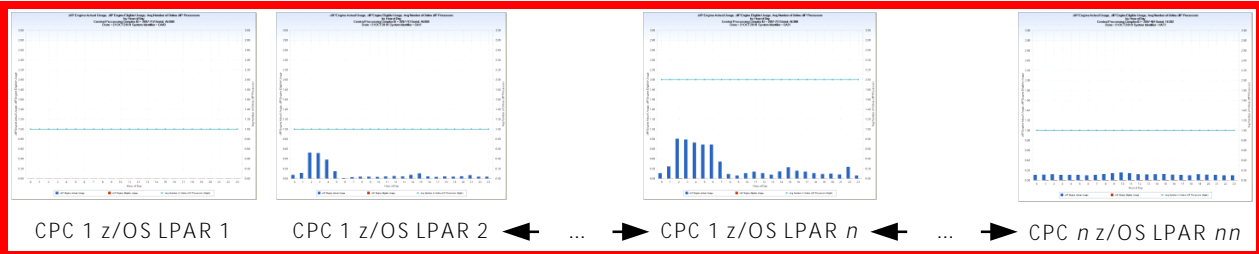
PRSMASPP - Avg zIIP Shareable Processors

Average number of physical zIIP engines online in the shared zIIP engine pool.

X-axis

HOUR – Hour of Day

Daily z/OS System zIIP Eng Use and Demand



Left Y-axis

ZIPENGA – zIIP Engine Actual Usage

ZIPENGE – zIIP Engine Eligible Usage

This extract shows the z/OS LPAR zIIP usage for the CPCs charted in the CPC level data extract. The vertical bars show total zIIP actual and zIIP eligible CPU time for all PR/SM managed z/OS LPARs at the LPAR level. The CPU times are divided by 3600 to convert from seconds to hours. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the CPU demand to be represented as physical engine dispatch hour equivalents. For example, two zIIP engines could, at full utilization, provide two hours of CPU time in one hour.

Right Y-axis

CPUAVOZP - Avg Number of Online zIIP Processors

Average number of logical zIIP processors online to the LPAR each hour.

X-axis

HOUR – Hour of Day

Query Defaults

- HARCPU DAYS timespan file: Cycles 01 -02
- WLMSEC DAYS timespan file: Cycles 01-02
- Filters: PRSMUSCT GT 0 (only PR/SM managed LPARs)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

CPCID

```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

ZAPENGA

```
ROUND((SECSUPTM/3600),.001);
```

ZAPENGE

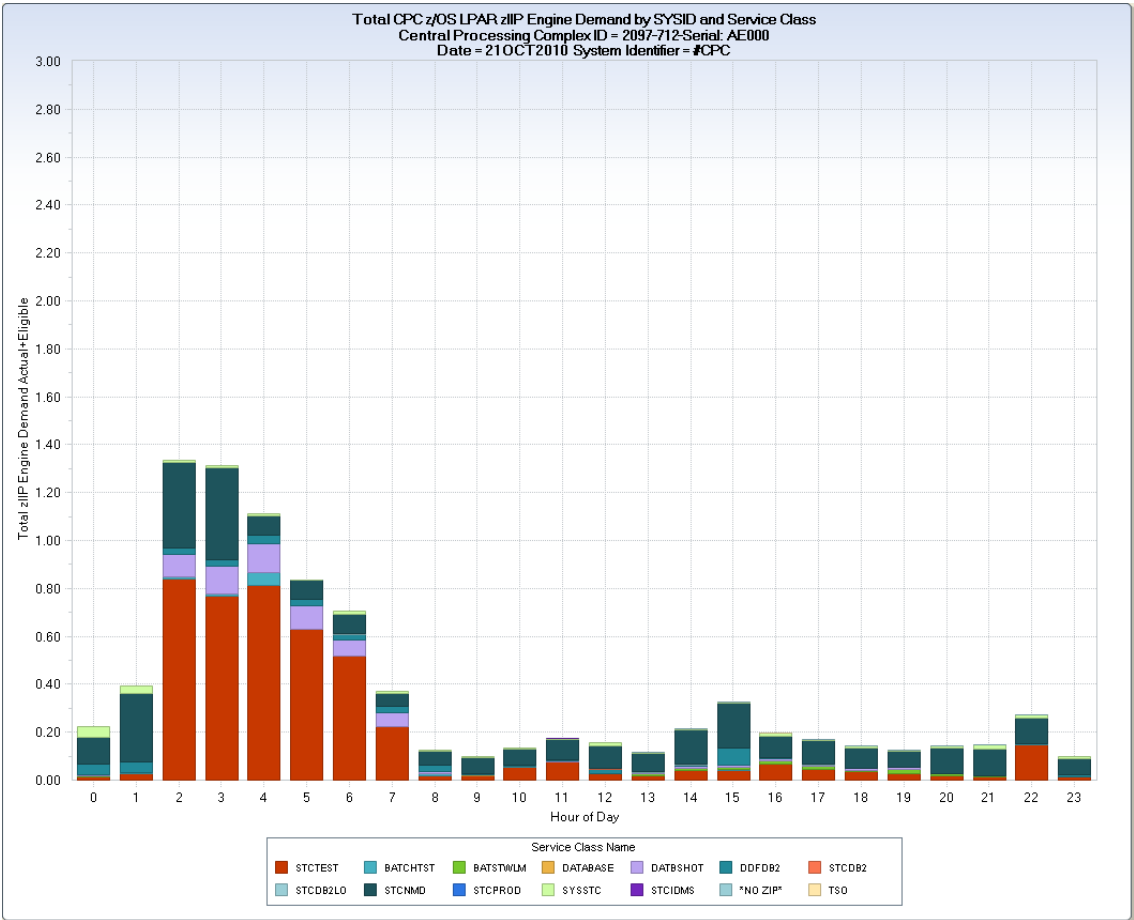
```
ROUND((SECSUPCT/3600),.001) ;
```

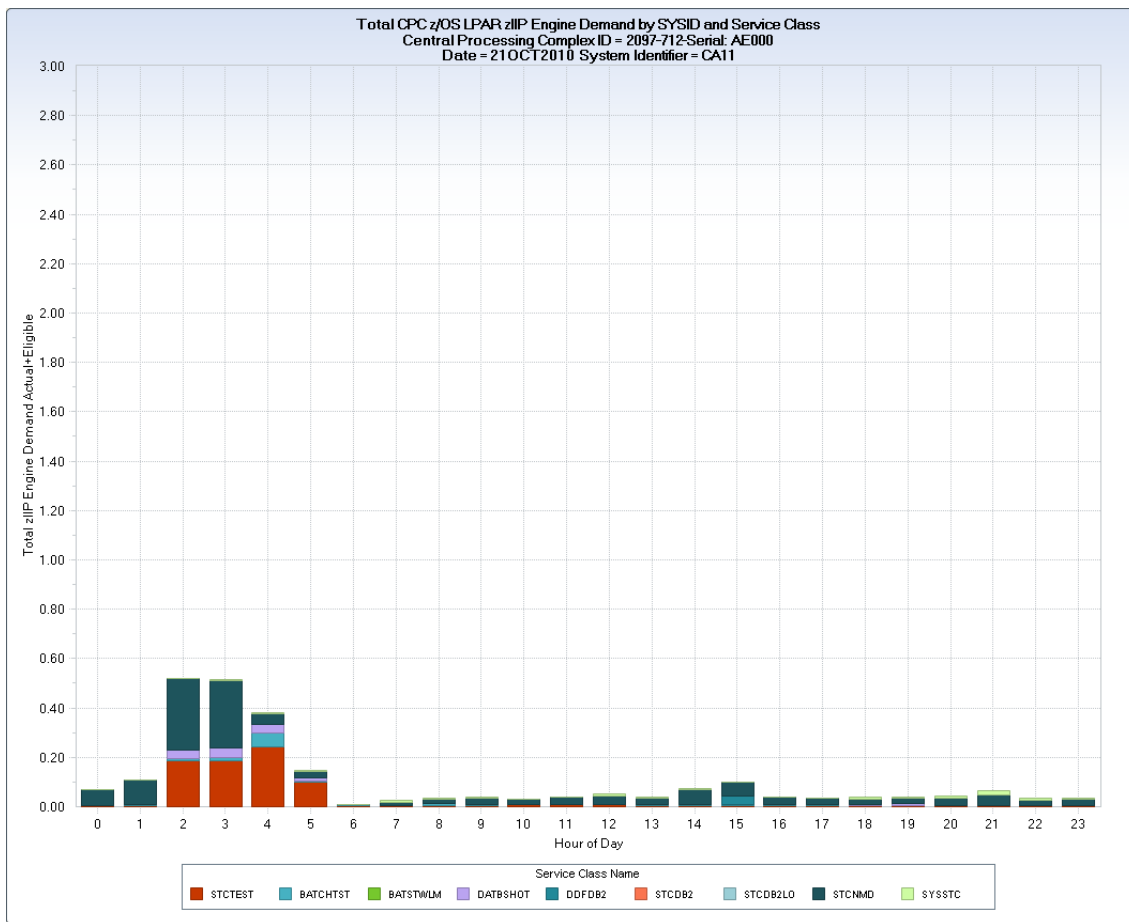
Modifications

The following modifications can be made to the RMFZIC query:

- You can increase the number HARCPU and WLMSEC DAYS timespan cycles input, but comment out the user code in each file selection step that selects yesterday's data.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.

RMFZIS – Daily zIIP Engine Use and Demand by Service Class





Overview

The RMFZIS query charts the hourly total zIIP engine demand (actual zIIP engine use plus zIIP engine eligible use), by WLM Service Class, for the Central Processing Complex (CPC) and individual z/OS LPARs using the shared engine pools of a CPC. Only PR/SM managed z/OS LPARs contribute to these charts because the data containing zIIP engine demand, extracted from the WLMSEC Service Class Resource Consumption file, is derived from RMF type 72-3 records, which only record zIIP engine demand for PR/SM managed z/OS LPARs.

The example shows the following charts:

- CPC level chart (top) for an IBM z10 2097-712 CPC
- One of the individual LPAR charts for an LPAR defined to that CPC

Both the CPC level and SYSID level charts show total hourly zIIP engine demand, using vertical bars, stacked by WLM Service Class.

Note: A fake Service class named `*NO ZAP*` is created, showing no usage, for hours where there was no zIIP engine demand. This class prevents charts from being produced where hours are missing from the X-axis.

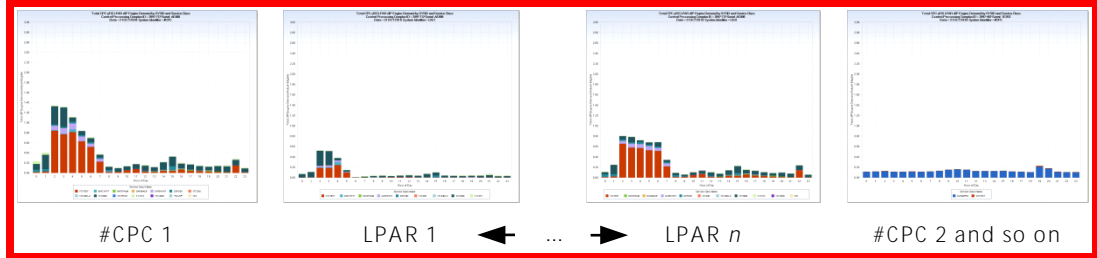
Uses

This query is used to examine the WLM Service Classes that execute zIIP engine eligible work. If zIIP usage appears for an LPAR that does not have zIIP engines assigned, then the work is actually executing on expensive CP engines. You may want to reconfigure the LPAR with one or more zIIP engines, or reroute the Service Class workload to execute on an LPAR with zIIP engines. The charts as delivered show total demand—which is the sum of zIIP engine actual use plus any zIIP engine eligible use that executed on CP engines. The two components—actual use and eligible use are available for charting, and can be selected as the Y-axis chart variables if desired. Additionally, you can review the RMFZIC - Daily zIIP Engine Use and Demand by CPC and SYSID query charts. These charts show the breakdown of actual and eligible zIIP engine demand in a single chart.

Note: If a CPC has one or more zIIP engines and no zAAP engines, and has specified `ZAAPZIIP=YES` in `SYS1.PARMLIB`, zIIP engines process all specialty engine work—including work that would normally run on zAAP engines. In this environment, all SMF and RMF metrics treat zAAP and zIIP eligible workloads as zIIP eligible. No zAAP actual or eligible demand appears.

Data Extracts

Daily Total zIIP Demand by Service Class



Left Y-axis

ZIPENGT – Total zIIP Engine Demand Actual+Eligible

The vertical bars show total zIIP usage (zIIP actual+eligible CPU time) at the CPC level and for each PR/SM managed z/OS LPAR. For each CPC, the CPC level chart is produced first, with a SYSID value of '#CPC', and is followed by the PR/SM managed z/OS LPARs for that CPC. The CPU times are divided by 3600 to convert from seconds to hours to create the chart variable ZIPENGT. Because one engine can be dispatched a maximum of one hour, per hour, this conversion allows the zIIP engine demand to be represented as physical engine hour equivalents. For example, two zIIP engines could, at full utilization, provide two hours of CPU time in one hour.

X-axis

HOUR – Hour of Day

Query Defaults

- HARCPU DAYS timespan file: Cycles 01 -02
- WLMSEC DAYS timespan file: Cycles 01-02
- Filters: PRSMUSCT GT 0 (only PR/SM managed LPARs)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

CPCID

```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

ZIPENGT

```
SUM(ZIPENGA, ZIPENGE) ;
```

ZIPENGA

```
ROUND((SECSUPTM/3600),.001);
```

ZIPENGE

```
ROUND((SECSUPCT/3600),.001) ;
```

Modifications

The following modifications can be made to the RMFZIS query:

- You can increase the number HARCPU and WLMSEC DAYS timespan cycles input, but comment out the user code in each file selection step that selects yesterday's data.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.
- If you want to generate these charts to see either the zIIP engine actual use or zIIP engine eligible use, change the Y-axis variable charted. The default is ZIPENGT, but ZIPENGA or ZIPENGE can be selected.

Mainframe MSU Utilization and LPAR Capping

This section contains the following topics:

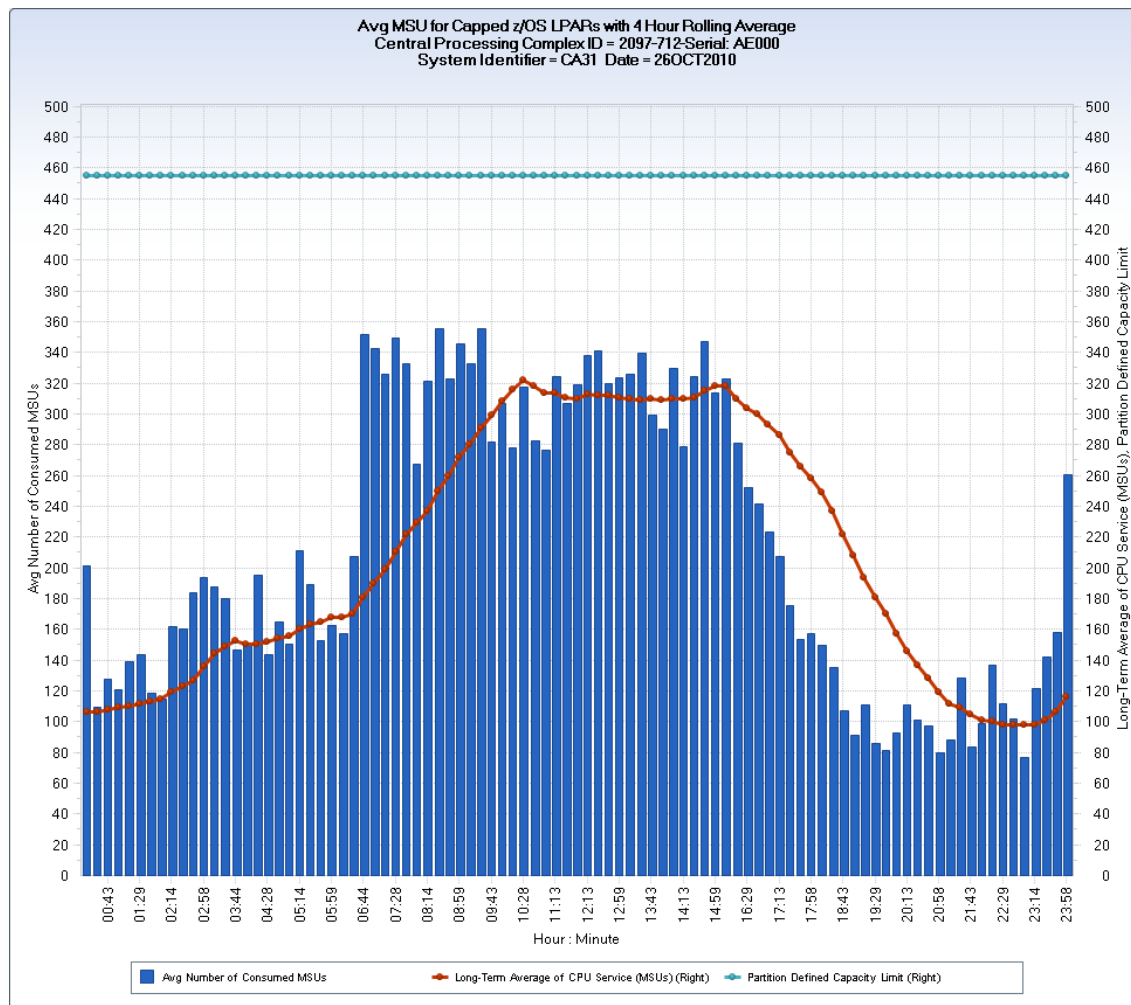
[RMFMC4 - Daily z/OS LPAR MSU Use Cap and 4 Hour Avg](#) (see page 190)

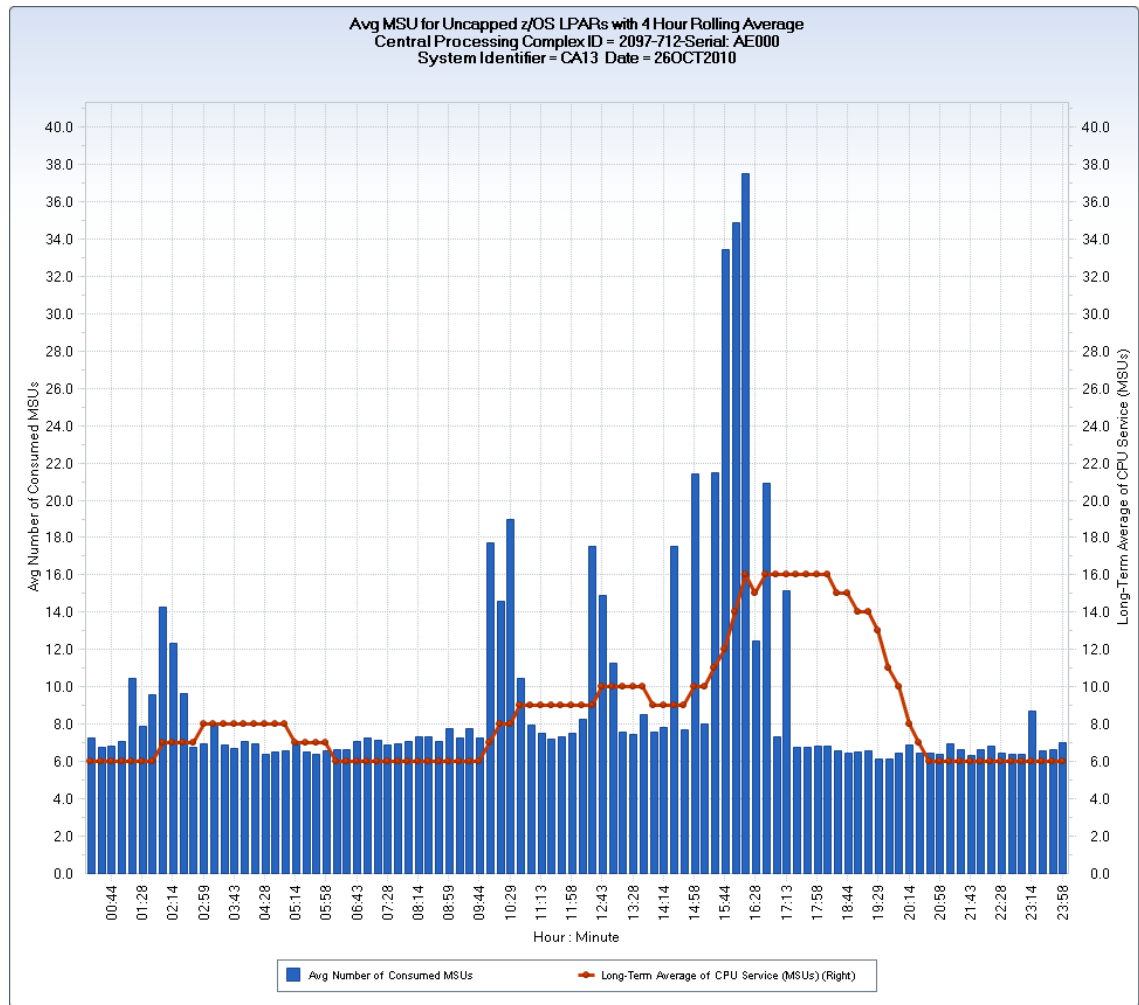
[RMFMCL - Daily LPAR MSU Use and Capacity Limits](#) (see page 195)

[RMFMMC - Monthly LPAR MSU Use and Capacity Limits](#) (see page 200)

[RMFMSU - Daily CPC MSU Use with Stacked LPARs](#) (see page 204)

RMFMC4 – Daily z/OS LPAR MSU Use Cap and 4 Hour Avg





Overview

The RMFMC4 query shows the average MSU usage of z/OS LPARs at the RMF interval level and includes the four hour rolling MSU average. All z/OS LPARs are accounted for. This query is similar to the RMFMCL query previously described, but has the additional benefit of showing the four hour rolling MSU average which is only available for z/OS LPARs. The query generates two separate data extracts. One for z/OS LPARs that have defined capacity limits, and the other for z/OS LPARs with no defined capacity limits. One of the uncapped LPARs for each CPC is the 'PHYSICAL' LPAR that captures PR/SM overhead.

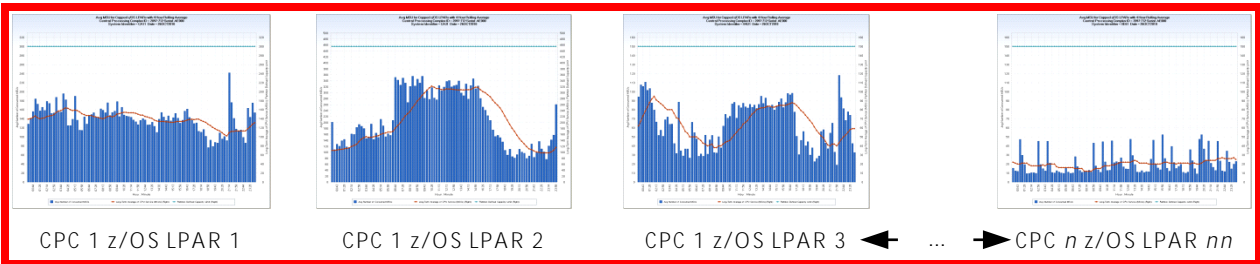
The top example shows the average MSU usage chart for a z/OS LPAR (PRSMPLPNM='CA31') running on an IBM z10 2097-712 CPC. The RMF intervals records were generated at 15-minute intervals, and the LPAR has a defined capacity limit of 455 MSUs. The second example shows the average MSU usage chart for another LPAR (PRSMPLPNM='CA13') on the same CPC. The RMF intervals records were also generated at 15-minute intervals. This second LPAR has no defined capacity limit.

Uses

The processing capacity of mainframe CPCs is expressed in MSUs (Millions of Service Units per Hour). One of the characteristics of a specific CPC model is its MSU capacity. This capacity is indicative of its workload processing capability and is directly related to the number of general-purpose CP engines configured. Each LPAR that is assigned logical CP processors uses some amount of the available CPC MSUs. The MSU capacity of a CPC and of individual LPARs can influence the cost of software licensed to run on LPARs. Some software licensing costs can be lowered by defining a capacity limit for an LPAR. The limit prevents the LPAR from using the full MSU capacity associated with the logical processors assigned to it. The RMFMC4 charts can be used to monitor the normal MSU consumption of LPARs that use general-purpose CP engines, and also to determine the appropriate value to use for a defined capacity limit. The four hour rolling average is helpful because LPARs are allowed to exceed the defined limit, without penalty, as long as the four hour rolling MSU average stays below the defined limit. For example, if the daily MSU usage shown for the LPAR in the top example (PRSM L PNM='CA31') is consistent, over time, it means that the limit of 455 MSUs is set too high. A defined capacity limit of about 330 to 340 MSUs is a more appropriate setting for this LPAR—given that the peak four-hour rolling average was 322 MSUs.

Data Extracts

Daily MSU Use for Cap z/OS LPARs 4H Avg



All LPARs in this data extract have defined capacity limits. If no LPARs in any CPCs have defined capacity limits, a single chart is produced with a CPCID of 'NO Capped LPARS' and an LPAR name of 'NADA.'

Left Y-axis

CPUAVMSU – Avg Number of Consumed MSUs

Right Y-axis #1

CPUAVLAC – Long-Term Average of CPU Service (MSUs)

Right Y-axis #2

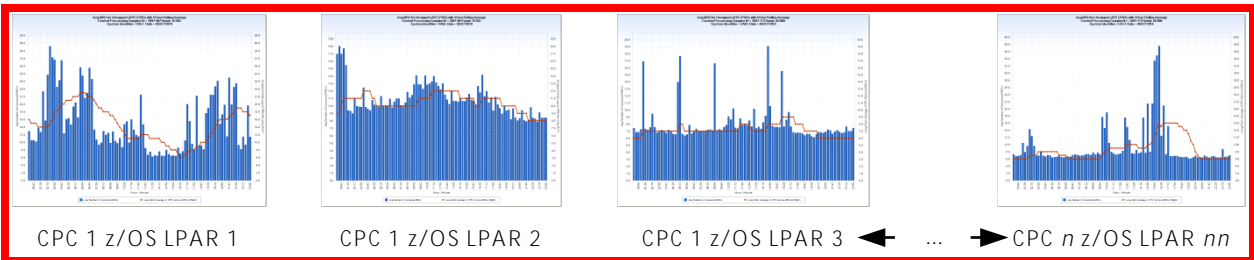
CPUDEFLM – Partition Defined Capacity Limit

X-axis

HHMM – Hour : Minute

Data element HHMM is derived by concatenating the HOUR and MINUTE of each RMF interval together, separated by a colon, HH:MM (for example, 14:59 meaning hour 14, 59th minute).

Daily MSU Use for Uncap z/OS LPARs 4H Avg



All LPARs in this data extract have no defined capacity limits. There is always at least one LPAR in this data extract for each CPC—the 'PHYSICAL' LPAR which shows PR/SM overhead.

Left Y-axis

CPUAVMSU – Avg Number of Consumed MSUs

Right Y-axis

CPUAVLAC – Long-Term Average of CPU Service (MSUs)

X-axis

HHMM – Hour : Minute

Data element HHMM is derived by concatenating the HOUR and MINUTE of each RMF interval together, separated by a colon, HH:MM (for example, 14:59 meaning hour 14, 59th minute).

Query Defaults

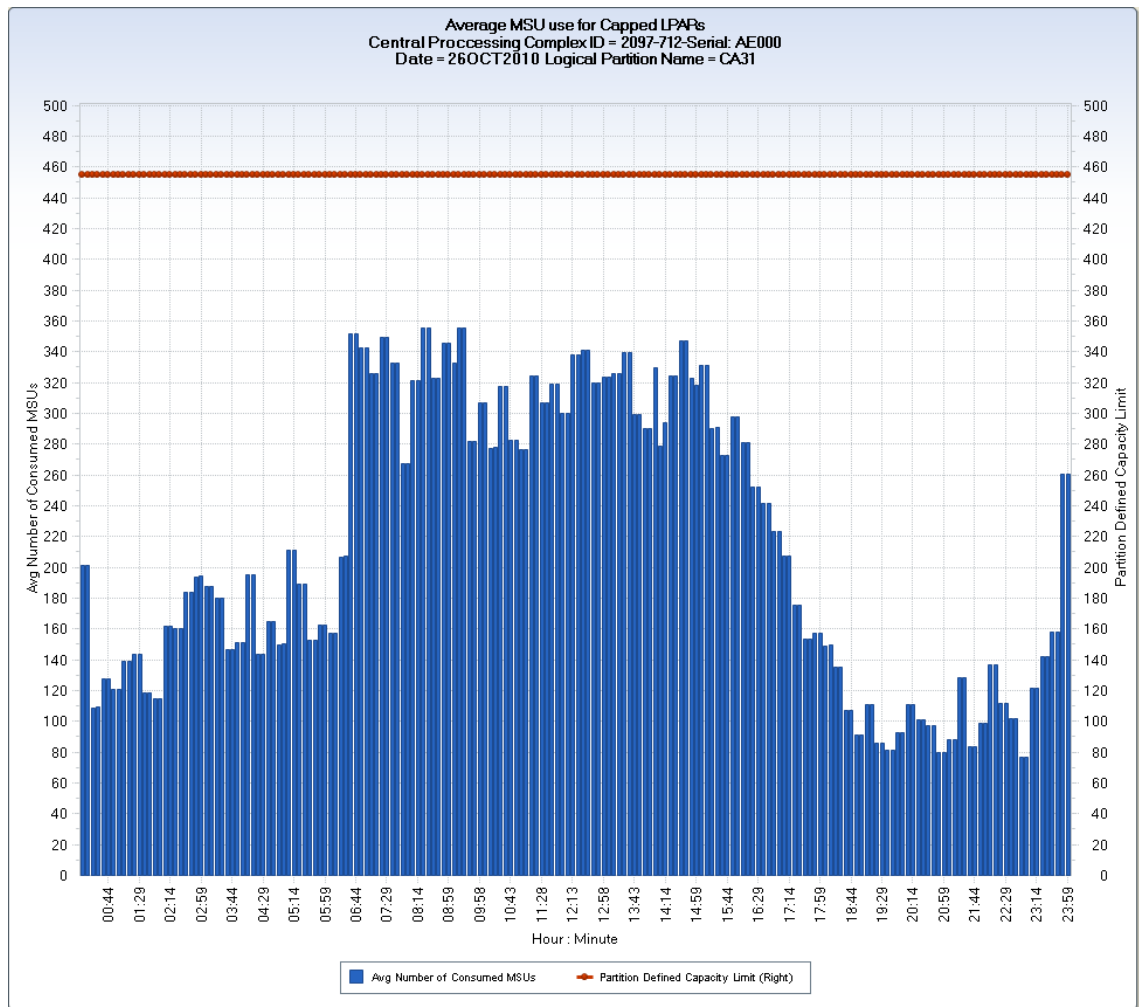
- HARCPU DETAIL timespan file: Cycles 01 - 02
- Filters: PRSMUSCT GT 0 (only PR/SM managed LPARs)
- User code excludes today's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
CPCID
`TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);`
DATE
`DATEPART(ENDTS);`
HHMM
`PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);`

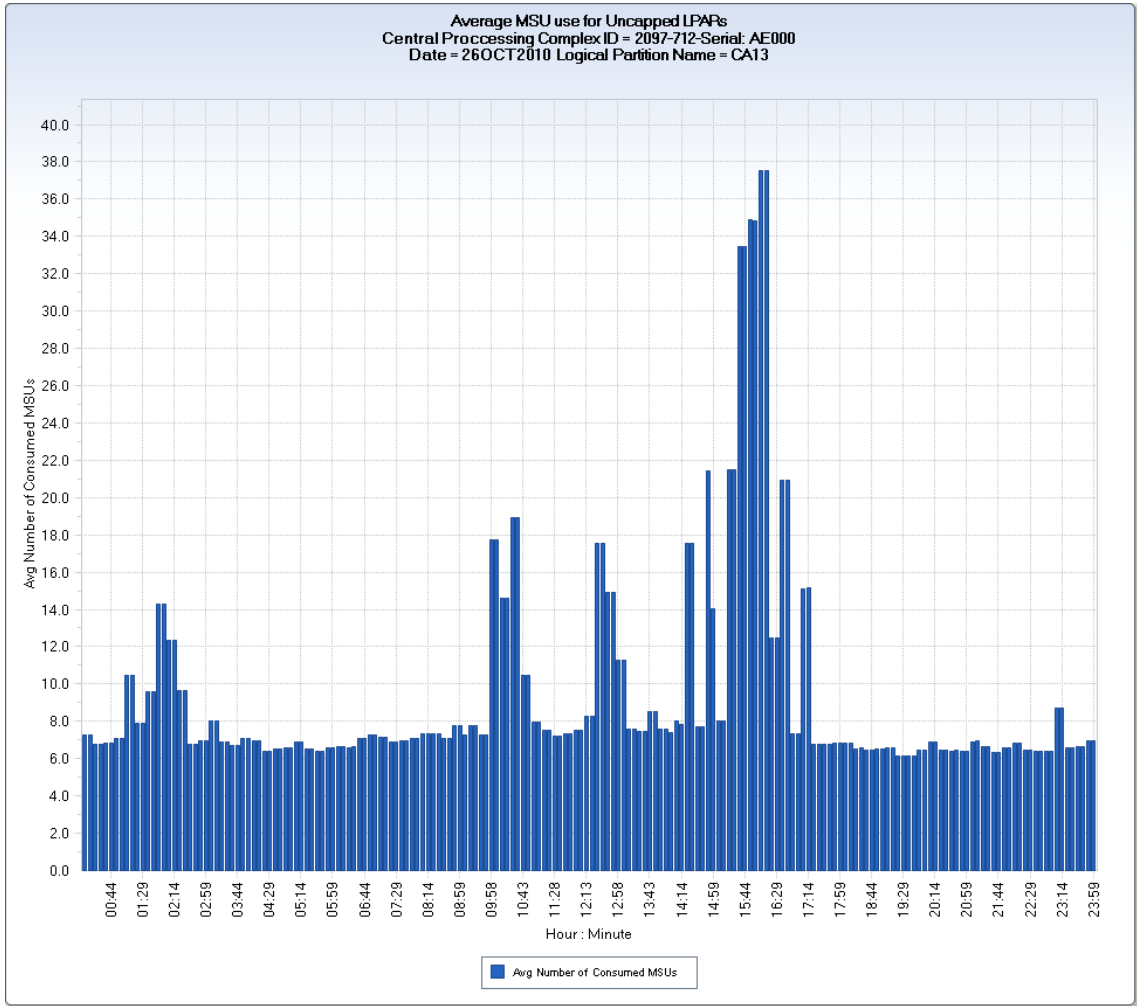
Modifications

The following modifications can be made to the RMFMC4 query:

- If you want to see hourly, rather than interval level MSU usage and four-hour rolling MSU averages, you can change use the HARLPC DAYS timespan.
- You can increase the number of cycles to create charts for multiple days, but comment out the user code in the initial file selection step that selects only yesterday's data.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.

RMFMCL – Daily LPAR MSU Use and Capacity Limits





Overview

The RMFMCL query shows the average MSU usage of LPARs at the RMF interval level. All LPARs are accounted for with the exception of LPARs that exclusively use specialty engines, such as Coupling Facility LPARs with ICF engines only or Linux LPARs with IFL engines only. If an LPAR uses at least one general-purpose CP engine, a chart is produced. The query generates two separate data extracts. One for LPARs that have defined capacity limits, and the other for LPARs with no defined capacity limits. One of the uncapped LPARs for each CPC is the 'PHYSICAL' LPAR that captures PR/SM overhead.

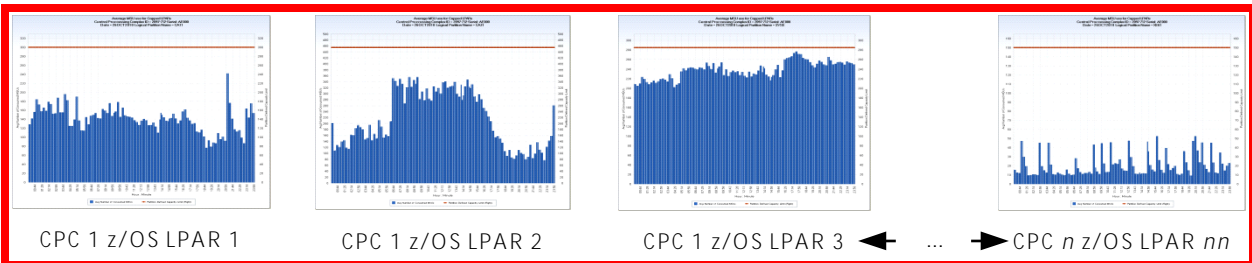
The top example shows the average MSU usage chart for an LPAR (PRSMPLNM='CA31') running on an IBM z10 2097-712 CPC. The RMF intervals records were generated at 15-minute intervals and the LPAR has a defined capacity limit of 455 MSUs. The second example shows the average MSU usage chart for another LPAR (PRSMPLNM='CA13') on the same CPC. The RMF intervals records were also generated at 15-minute intervals. This second LPAR has no defined capacity limit.

Uses

The processing capacity of mainframe CPCs is expressed in MSUs (Millions of Service Units per Hour). One of the characteristics of a specific CPC model is its MSU capacity. This capacity is indicative of its workload processing capability and is directly related to the number of general-purpose CP engines configured. Each LPAR that is assigned logical CP processors uses some amount of the available CPC MSUs. The MSU capacity of a CPC and of individual LPARs can influence the cost of software licensed to run on LPARs. Some software licensing costs can be lowered by defining a capacity limit for an LPAR. The limit prevents the LPAR from using the full MSU capacity associated with the logical processors assigned to it. These charts can be used to monitor the normal MSU consumption of LPARs that use general-purpose CP engines, and also to determine whether the values used as defined capacity limits are correctly set. For example, if the daily MSU usage shown for the LPAR in the top example (PRSM L PNM='CA31') is consistent, over time, it means that the limit of 455 MSUs is set too high. A defined capacity limit of about 360 to 380 MSUs is a more appropriate setting for this LPAR.

Data Extracts

Daily MSU Use for Capped LPARs



All LPARs in this data extract have defined capacity limits. If no LPARs in any CPCs have defined capacity limits, a single chart is produced with a CPCID of 'NO Capped LPARS' and an LPAR name of 'NADA.'

Left Y-axis

LPCAVMSU – Avg Number of Consumed MSUs

Right Y-axis

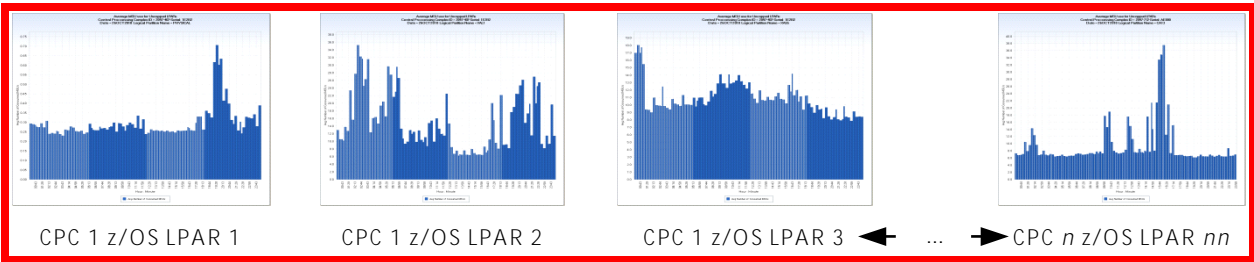
LPCDEFLM – Partition Defined Capacity Limit

X-axis

HHMM – Hour : Minute

Data element HHMM is derived by concatenating the HOUR and MINUTE of each RMF interval together, separated by a colon, HH:MM (for example, 14:59 meaning hour 14, 59th minute).

Daily MSU Use for Uncapped LPARs



All LPARs in this data extract have no defined capacity limits. There is always at least one LPAR in this data extract for each CPC—the 'PHYSICAL' LPAR which shows PR/SM overhead.

Left Y-axis

LPCAVMSU – Avg Number of Consumed MSUs

Right Y-axis

LPCDEFLM – Partition Defined Capacity Limit

X-axis

HHMM – Hour : Minute

Data element HHMM is derived by concatenating the HOUR and MINUTE of each RMF interval together, separated by a colon, HH:MM (for example, 14:59 meaning hour 14, 59th minute).

Query Defaults

- HARLPC DETAIL timespan file: Cycles 01 - 02
- Filters: PRSMLPTP EQ 'CP' or PRSMLPTP EQ 'PHY' (select LPARs using CP engines)
- User code excludes today's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

CPCID

```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

DATE

```
DATEPART(ENDTS);
```

HHMM

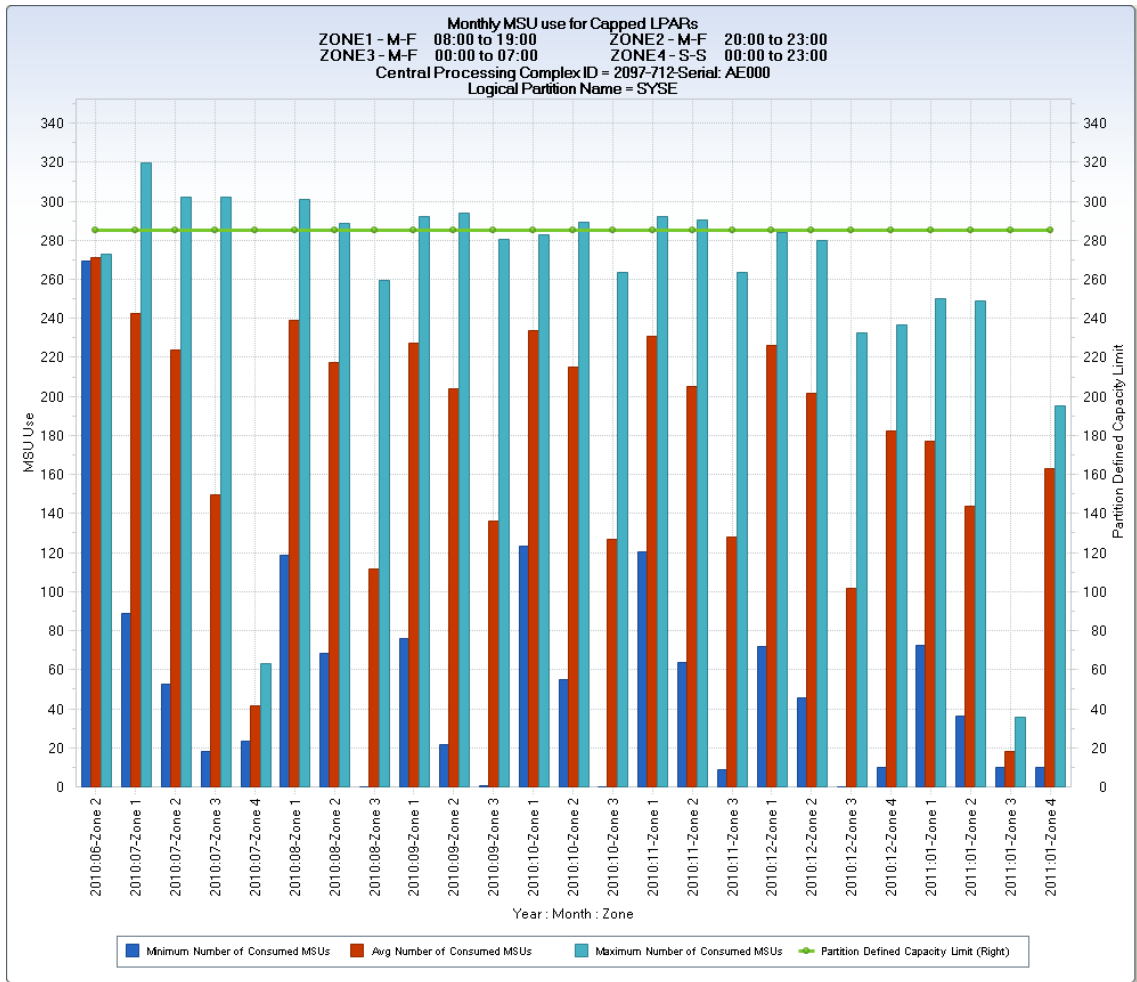
```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

Modifications

The following modifications can be made to the RMFMCL query:

- If you want to see hourly, rather than interval level MSU usage averages, you can change use the HARLPC DAYS timespan.
- You can increase the number of cycles to create charts for multiple days, but comment out the user code in the initial file selection step that selects only yesterday's data.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.

RMFMMC – Monthly LPAR MSU Use and Capacity Limits



Overview

The RMFMMC query shows the monthly z/OS LPAR MSU use at the ZONE (Time Zone) level. All z/OS LPARs are accounted for. For each ZONE within each month, three vertical bars show the minimum, average, and maximum MSU use in the time range. The query generates two separate data extracts. One for z/OS LPARs that have defined capacity limits, and the other for z/OS LPARs with no defined capacity limits. One of the uncapped LPARs for each CPC is the 'PHYSICAL' LPAR that captures PR/SM overhead.

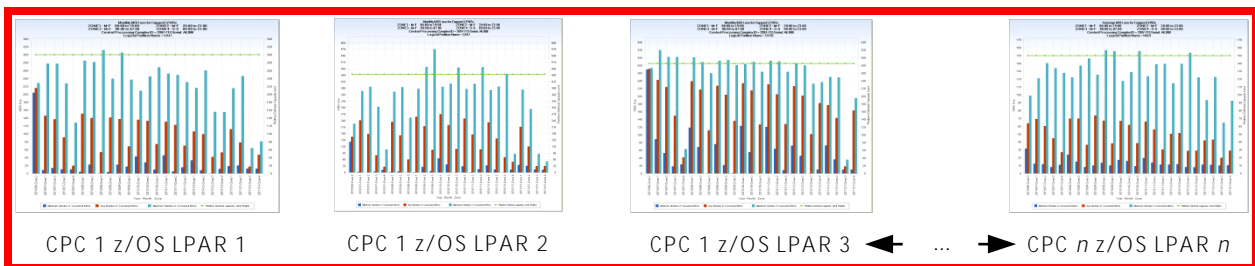
The example shows the minimum, average, and maximum MSU usage chart for a z/OS LPAR (PRSMPLNM='SYSE') running on an IBM z10 2097-712 CPC. Four ZONES have been defined to CA MICS. The title of the chart is manually customized using the Q&R View Edit facilities to specify the day and time ranges associated with each ZONE.

Uses

The processing capacity of mainframe CPCs is expressed in MSUs (Millions of Service Units per Hour). One of the characteristics of a specific CPC model is its MSU capacity. This capacity is indicative of its workload processing capability and is directly related to the number of general-purpose CP engines configured. Each LPAR that is assigned logical CP processors uses some amount of the available CPC MSUs. The MSU capacity of a CPC and of individual LPARs can influence the cost of software licensed to run on LPARs. Some software licensing costs can be lowered by defining a capacity limit for an LPAR. The limit prevents the LPAR from using the full MSU capacity associated with the logical processors assigned to it. The RMFMMC charts are useful for visualizing longer term resource usage trends for your z/OS LPARs. The average MSU usage bar is the most important. The MIN and MAX represent singular extremes, while the average MSU use is the average of all RMF intervals over each time period charted,

Data Extracts

Monthly MSU Use for Capped LPARs



All LPARs in this data extract have defined capacity limits. If no LPARs in any CPCs have defined capacity limits, a single chart is produced with a CPCID of 'NO Capped LPARS' and an LPAR name of 'NADA.'

Left Y-axis

LPCMNSMU – Avg Number of Consumed MSUs

LPCUAVMSU – Avg Number of Consumed MSUs

LPCMMSMU – Avg Number of Consumed MSUs

Right Y-axis

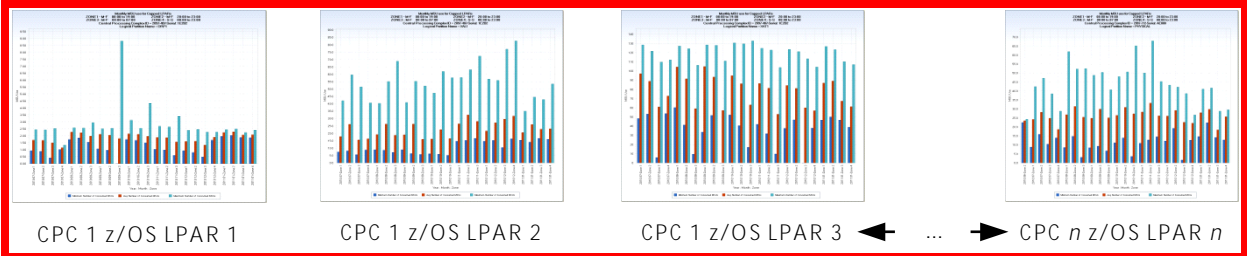
LPCDEFMLM – Partition Defined Capacity Limit

X-axis

YYYYMMZZ – Year : Month : Zone

Data element YYYYMMZZ is derived by concatenating the 4-digit year, colon ":", and 2-digit month, with the literal "-Zone ", and the 1-digit ZONE value (for example, 2010:08-Zone 1).

Monthly MSU Use for Uncapped LPARs



All LPARs in this data extract have no defined capacity limits. There is always at least one LPAR in this data extract for each CPC—the 'PHYSICAL' LPAR which shows PR/SM overhead.

Left Y-axis

LPCMNSMU – Avg Number of Consumed MSUs

LPCUAVMSU – Avg Number of Consumed MSUs

LPCMMSMU – Avg Number of Consumed MSUs

X-axis

HHMM – Hour : Minute

Data element YYYYMMZZ is derived by concatenating the 4-digit year, colon ":", and 2-digit month, with the literal "-Zone ", and the 1-digit ZONE value (for example: 2010:08-Zone 1).

Query Defaults

- HARLPC MONTHS timespan file: Cycles 00 - 06
- Filters: PRSMLPTP EQ 'CP' or 'PHYSICAL' (Only CP using LPARs or PR/SM overhead (PHYSICAL))
- Derivations:
 - CPCID


```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
YYYYMMZZ
PUT(YEAR4,Z4.) || ':' || PUT(MONTH,Z2.) || '-Zone ' || SUBSTR(ZONE,1,1) ;
```

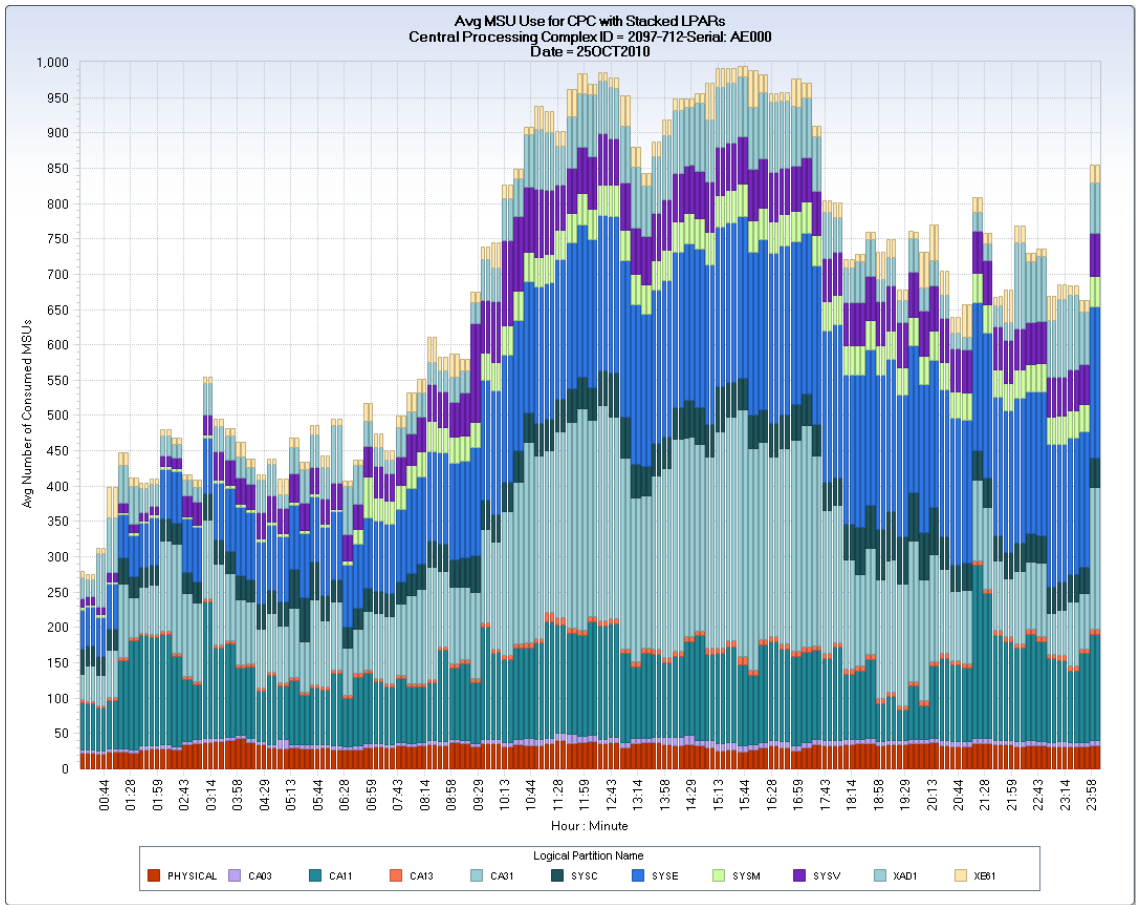
Modifications

The following modifications can be made to the RMFMMC query:

Important! Modify the chart titles, using the Q&R View Edit facility to change the ZONE definitions if they differ from the days/hours specified in the sample query.

- You can increase or decrease the number of cycles to look at more or few months.
- You can use a filter statement to restrict data to focus on a particular time zone (for example, ZONE EQ 1).
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.
- If you want to focus exclusively on the important Average MSU use value, you can remove the MIN and/or MAX MSU use data elements from the chart display.

RMFMSU – Daily CPC MSU Use with Stacked LPARs



Overview

The RMFMSU query shows the average MSU usage for the Central Processing Complex (CPC) at an hourly or RMF interval level, with the individual LPAR usage broken out in each vertical bar. All LPAR MSU usage is accounted for with the exception of LPARs that have only specialty engines, such as Coupling Facility LPARs with ICF engines only or Linux LPARs with IFL engines only. If an LPAR uses at least one general-purpose CP engine, it contributes to the chart. Two MSU utilization data extracts are provided. One extract generates daily charts for each CPC, and another extract generates a single chart for each CPC showing multiple days of activity if multiple HARLPC file cycles are input.

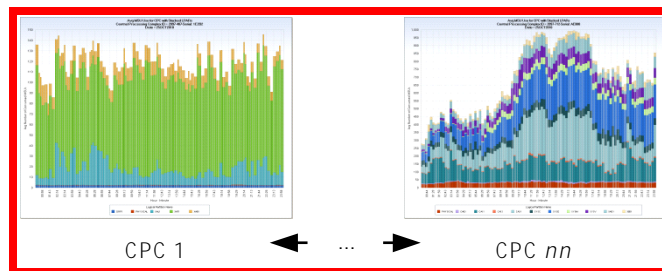
The example shows the daily chart for an IBM z10 2097-712 CPC, with the average MSU usage for each 15-minute RMF interval,

Uses

The processing capacity of mainframe CPCs is expressed in MSUs (Millions of Service Units per Hour). This query provides a CPC level view of MSU usage. The generated MSU charts have the same pattern as the daily CP engine utilization chart by hour generated by query RMFCPD—because MSU usage is a derivative of CP engine usage. These charts help you see LPAR usage patterns relative to each other throughout the day. These charts can be compared to the MSU capacity of the CPC to help determine when an upgrade (or downgrade) is considered.

Data Extracts

Daily MSU Use by CPC Stacked LPARs



Left Y-axis

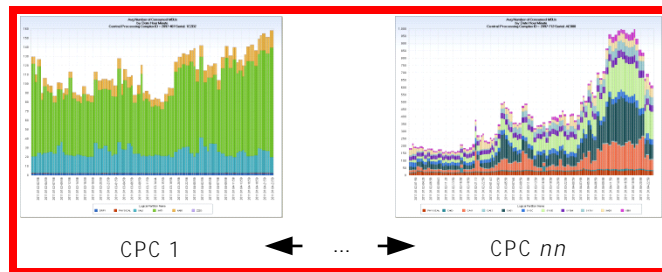
LPCAVMSU – Avg Number of Consumed MSUs

X-axis

HHMM – Hour : Minute

Data element HHMM is derived by concatenating the HOUR and MINUTE of each RMF interval together, separated by a colon, HH:MM (for example, 14:59 meaning hour 14, 59th minute).

Multiday MSU Use by CPC Stacked LPARs



Left Y-axis

LPCAVMSU – Avg Number of Consumed MSUs

X-axis

DATEHHMM – Date Hour Minute

Data element DATEHHMM is derived by concatenating the DATE, HOUR and MINUTE of each RMF interval together in the format YYYY.MM.DD-HH:MM.

Query Defaults

- HARLPC DAYS timespan file: Cycles 01 - 03
- Filters: PRSMLPTP EQ 'CP' or PRSMLPTP EQ 'PHY' (select LPARs using CP engines)
- Derivations:

CPCID

```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

DATE

```
DATEPART(ENDTS);
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

DATEHOUR

```
DATEHOUR = PUT(DATEPART(ENDTS),YMMDD10.) || ':' || PUT(HOUR,Z2.) ;
```

DATEHHMM

```
DATEHHMM = PUT(DATEPART(ENDTS),YMMDDP10.) || '-' || HHMM ;
```

Modifications

The following modifications can be made to the RMFMSU query:

- If you want to see hourly, rather than interval level MSU usage averages, you can change use the HARLPC DAYS timespan . If so, you may want to substitute the DATEHOUR element for the DATEHHMM element in the SUMMARIZATION and DATA EXTRACT steps.
- If you want to see only yesterday's data, uncomment the User Code in User Step task in the Input HARLPC File step.
- You can increase or decrease the number of HARLPC file cycles to see more or fewer days of average MSU utilization.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1)
- You can add filter statements to select only a single CPC, or set of CPCs using the identifying data elements shown in the CPCID derivation.

WLM Service Class and Report Class Analysis

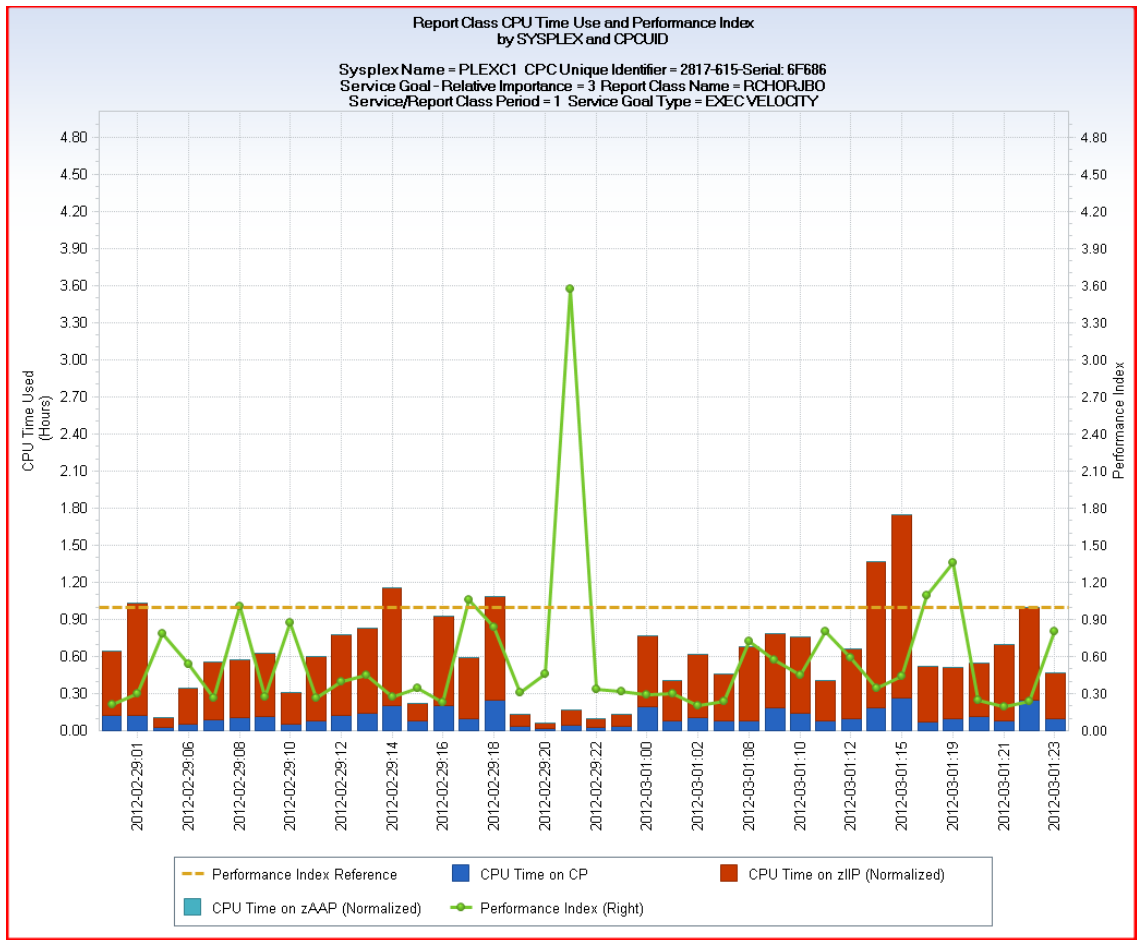
This section contains the following topics:

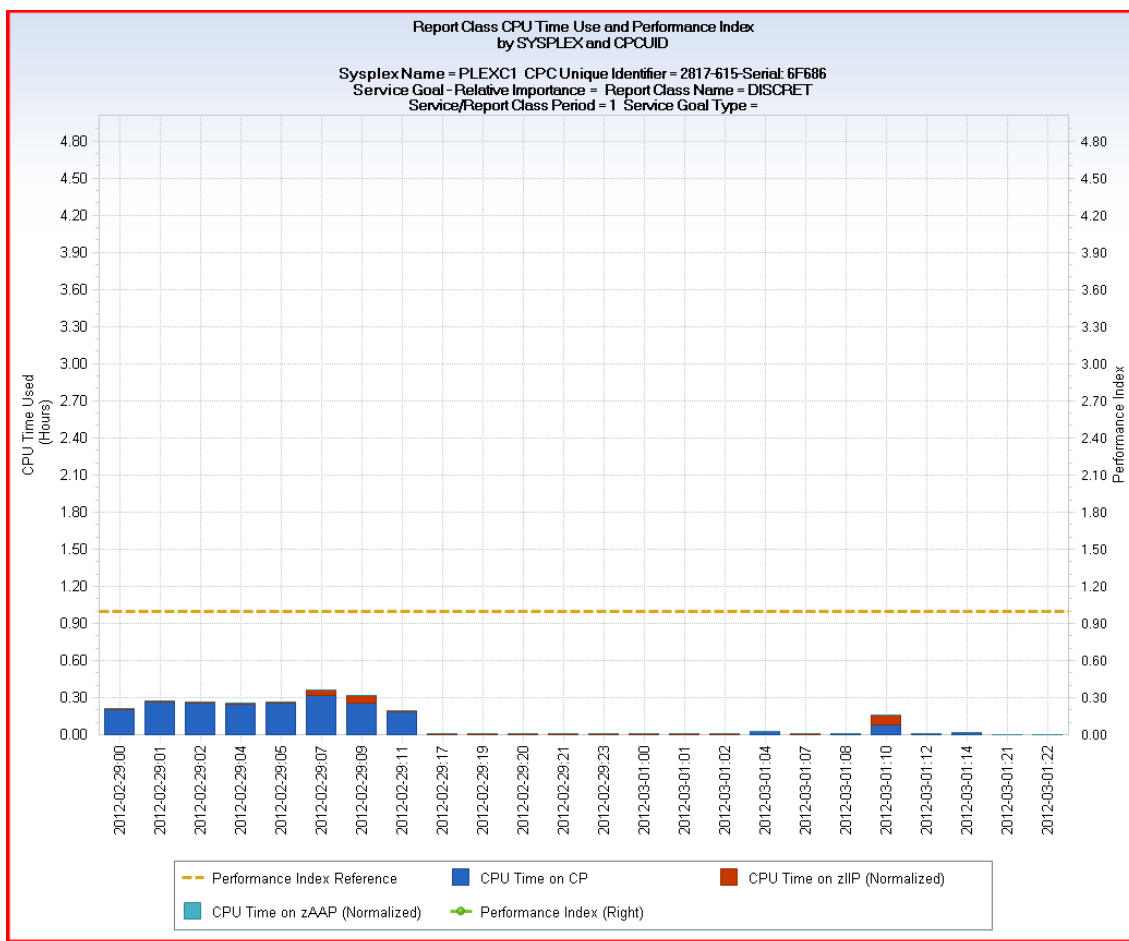
[RMFRPT - z/OS Report Class Analysis by SYSPLEX, CPCID, and SYSID](#) (see page 208)

[RMFS10 - z/OS Daily Top 10 Service Classes by SYSPLEX](#) (see page 218)

[RMFSRV - z/OS Service Class Analysis by SYSPLEX, CPCID, and SYSID](#) (see page 222)

RMFRPT – z/OS Report Class Analysis by SYSPLEX, CPCID, and SYSID





Overview

The RMFRPT query displays resource usage, performance indicators, summary sampled state percentages, and detailed execution delay state information for each Workload Manager (WLM) Report Class or Report Class Period. This information is provided by the following three data extract CSV's at the z/OS LPAR (SYSID), Central Processing Complex (CPC), and SYSPLEX levels:

- Report Class by SYSPLEX
- Report Class by SYSPLEX CPCUID
- Report Class by SYSPLEX CPCUID SYSID

Each data extract generates the following four chart views:

1. CPU and PI
2. Sampling Summary
3. Exec Delay States Group 1
4. Exec Delay States Group 2

Refer to the RMFSRV – z/OS Service Class Analysis by SYSPLEX, CPCID, and SYSID query description for information about each of these chart views. The chart views and individual chart content for the RMFSRV and this RMFRPT queries are almost identical. The only difference is in the title area and relates to the type of the charted Report Classes.

WLM Report Classes can be homogeneous or heterogeneous. Homogeneous Report Classes are defined to include workloads from a single Service Class. Heterogeneous Report Classes are defined to include workloads that span two or more Service Classes.

For both homogeneous and heterogeneous Report Classes, the title area of each chart shows the SYSPLEX name, Report Class Name, Relative Importance, Goal Type, and Report Class period charted. For homogeneous Report Classes, all of these fields will be populated.

For heterogeneous Report Classes, the Relative Importance and Goal Type values will be blank and the Report Class Period value will show a value of 1 (one). This is because heterogeneous Report Classes include workload information drawn from multiple Service Classes that could have different importance levels, goal types, and periods.

Both chart samples shown previously are charts generated from the data extract Report Class by SYSPLEX CPCUID.

The first chart shows an example of a homogeneous Report Class, Report Class RCHORJBO. As you can see, all values in the chart title area are populated.

The second chart sample above shows an example of a heterogeneous Report Class, Report Class DISCRET. You can see that the Relative Importance and Goal Type are blank, the period reported is 1, and the Performance Index is missing.

Uses

Refer to the RMFSRV – z/OS Service Class Analysis by SYSplex, CPCID, and SYSID query description for information about the use of this RMFRPT query. The chart views and individual chart content for the RMFSRV and this RMFRPT queries are almost identical.

While Service Classes must be defined, Report Classes are optional, user-defined reporting groups. Report Classes allow the analyst to report on a specific set of workloads. These workloads a sub-set of the workloads belonging to a Service Class, or may span two or more Service Classes.

To be efficient, the number of Service Classes defined for a specific WLM Policy must be limited. A typical Service Policy may have only 30 to 40 Service Classes defined. But up to 999 Report Classes may be defined, providing the user a powerful tool to track the performance of specific, mission critical applications.

When the workloads assigned to a Report Class are a subset of a single Service Class (for example, a Report Class that selects a single CICS region belonging the Service Class PRODCICS), the Report Class is homogeneous. Information about the Report Class data includes Importance Level, Period level records, Goal and Performance Index information.

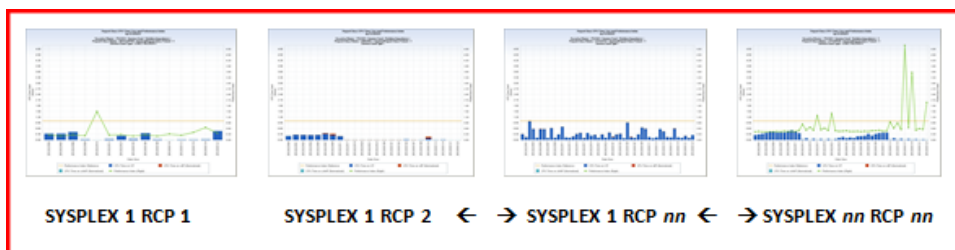
While CPU time is the default resource usage metric plotted in the CPU and PI view, the Data Extract CSV provides a number of other resource usage metrics that can be alternatively selected for charting. These metrics include DASD Service time, paging rates, and average storage used.

Note: By design, SYSplexES can span Central Processing Complexes. If your SYSplexES do not span CPCs, then the first two data extracts are identical, with the exception that the second data extract, Report Class by SYSplex CPCID, identifies the CPC in each chart title. If your SYSplexES do span CPCs, then the Report Class by SYSplex CPCID data extract displays multiple charts for a Report Class that ran on multiple CPCs.

Data Extracts

Report Class by SYSPLEX

VIEW 1 of 4: WLMRPTCL by SYSPLEX – CPU and PI



Note: RCP=Report Class Period

Left Y-axis

- SECCPUTM – CPU Time on CP
- SECSUPTM – CPU Time on zIIP (Normalized)
- SECZAPTM – CPU Time on zAAP (Normalized)

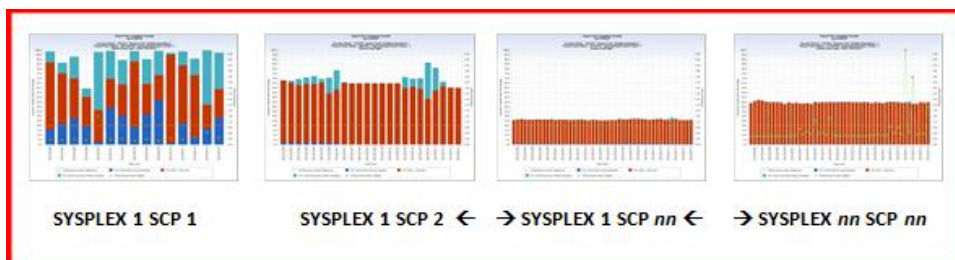
Right Y-axis

- SECPINDX – Performance Index
- Performance Index Reference – value set to 1

X-axis

HHMM – Hour: Minute

VIEW 2 of 4: WLMRPTCL by SYSPLEX – Sampling Summary



Left Y-axis

- SEDPCUSE – Pct Total WLM Using Samples (derived in query code)
- SEDPCDUN – Pct Wait – Unknown
- SEDPCSAM – Pct Total Execution Delay Samples (derived in query code)

Right Y-axis

- SECPINDX – Performance Index
- Performance Index Reference – value set to 1

X-axis

HHMM – Hour: Minute

VIEW 3 of 4: WLMRPTCL by SYSPLEX – Exec Delay States Group 1



Left Y-axis

- SEDPCDCP – Pct Wait – CPU on CP
- SEDPCSUD – Pct Wait – CPU on zIIP
- SEDPCZPD – Pct Wait – CPU on zAAP
- SEDPCAMD – Pct Wait – Crypto AM Processor
- SEDPCAPD – Pct Wait – Crypto Assist Processor
- SEDPCCRD – Pct Wait – Crypto Processors
- SEDPCCOD – Pct Wait – Resource Contention
- SEDPCDCC – Pct Wait – CPU Capping
- SEDPCDQU – Pct Wait – Quiesce
- SEDPCDMP – Pct Wait – MPL Delay
- SEDPCSMP – Pct Wait – Server MPL
- SEDPCDRG – Pct Wait – Resource Group Capping
- SEDPCDSD – Pct Wait – DASD I/O Delay
- SEDPCSQU – Pct Wait – Server Queue Delay

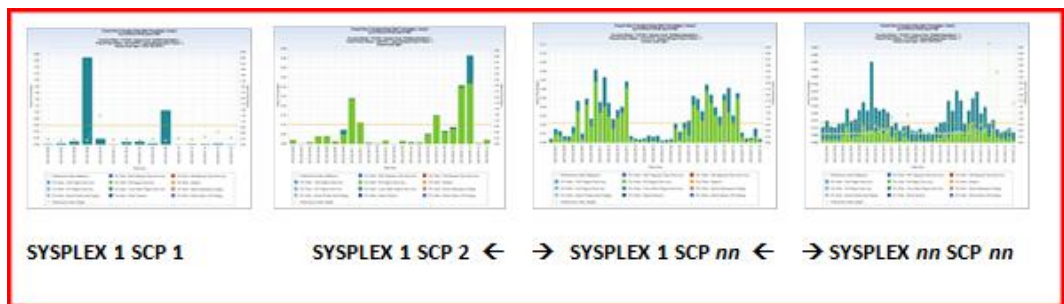
Right Y-axis

- SECPINDX – Performance Index
- Performance Index Reference – value set to 1

X-axis

HHMM – Hour: Minute

VIEW 4 of 4: WLMRPTCL by SYSPLEX – Exec Delay States Group 2



Left Y-axis

SEDPCDES – Pct Wait – ESO Hiperspc Pg-In from Aux
SEDPCDHS – Pct Wait – Std Hiperspc Pg-In from Aux
SEDPCDPC – Pct Wait – CSA Page-In from Aux
SEDPCDPP – Pct Wait – Pvt Page-In from Aux
SEDPCDSI – Pct Wait – Swap-In
SEDPCDVI – Pct Wait – VIO Page-In from Aux
SEDPCDXM – Pct Wait – Cross Mem Page-In from Aux
SEDPCSHP – Pct Wait – Server Hiperspace Paging
SEDPCSPP – Pct Wait – Server Private Area Paging
SEDPCSSI – Pct Wait – Server Swap-In
SEDPCSVI – Pct Wait – Server Space VIO Paging

Right Y-axis

SECPINDX – Performance Index
Performance Index Reference – value set to 1

X-axis

HHMM – Hour: Minute

Report Class by SYSPLEX CPCUID

Data extract Report Class by SYSPLEX CPCUID provides the same four Q&R chart views, with charts displaying the same Y and X axis charted data elements, as those described for data extract Report Class by SYSPLEX.

The only difference is that additional charts are generated if the SYSPLEX spans multiple CPCs. The chart granularity for the SYSPLEX level data extract followed this pattern:

SYSPLEX 1 RCP 1 SYSPLEX 1 RCP 2 ← ... → SYSPLEX 1 RCP nn ← ... → SYSPLEX nn RCP nn

The chart granularity for the Report Class by SYSPLEX CPCUID data extract follows this pattern:

SYSPLEX 1 CPC 1 SYSPLEX 1 CPC 1 ← ... → SYSPLEX 1 CPC 1 ← ... → SYSPLEX nn CPC nn
RCP 1 RCP 2 RCP nn RCP nn

If a SYSPLEX spans two or more CPCs, the the SYSPLEX CPCUID level data extract generates more unique charts that the SYSPLEX level data extract.

If a SYSPLEX does not span CPCs, however, the number of unique charts generated by the SYSPLEX level and SYSPLEX CPCUID level data extracts are identical.

Report Class by SYSPLEX CPCUID SYSID

Data extract Report Class by SYSPLEX CPCUID SYSID provides the same four Q&R chart views, with charts displaying the same Y and X axis charted data elements, as those described for data extract Report Class by SYSPLEX.

The only difference is that additional charts are generated if the SYSPLEX spans multiple CPCs, or multiple z/OS LPARs.

The chart granularity for the SYSPLEX level data extract followed this pattern:

SYSPLEX1 RCP 1 SYSPLEX1 RCP 2 ←... → SYSPLEX1 RCP nn ←... → SYSPLEXnn RCP nn

The chart granularity for the Report Class by SYSPLEX CPCUID SYSID data extract follows this pattern:

SYSPLEX1 CPC 1 SYSPLEX1 CPC 1 ←... → SYSPLEX1 CPC 1 ←... → SYSPLEXnn CPC nn
SYSID1 RCP 1 SYSID1 RCP 2 SYSID1 RCP nn SYSID nn RCP nn

With this data extract, you can examine the performance of each Report Class Period on each z/OS LPAR.

Query Defaults

The following are RMFRPT Query Defaults:

- HARCPU DAYS time span file: Cycle 01 (used to determine CPCID)
- WLM_EC DAYS timespan file: Cycles 01 – 02
- WLM_ED DAYS timespan file: Cycles 01 – 02
- Filters: Empty Filter tasks provided as a placeholder for user WLM_EC and WLM_ED file filtering to exclude/select specific SYSPLEXs, SYSIDs, ZONES, and so on.
- Commented user code for WLM_EC and WLM_ED file steps only accepts yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

CPCUID

```
CPCUID = PUT(SYSID,$CPCFMT.) ;
```

DATE

```
DATEPART(ENDTS);
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

DATEHOUR

```
PUT(DATEPART(ENDTS),YMMDDD10.) || ':' || PUT(HOUR,Z2.) ;
```

DATEHHMM

```
PUT(DATEPART(ENDTS),YMMDDP10.) || '-' || HHMM ;
```

SECFILE

```
1 ; Used after MERGE of WLM_EC and WLM_ED files to limit charting to observations that contained WLM_EC file data.
```

SEDPCSAM

```
100 * SEDTOSAM / SEDTOSST;
```

SEDPCUSE

```
100 * SEDTOUSE / SEDTOSST;
```

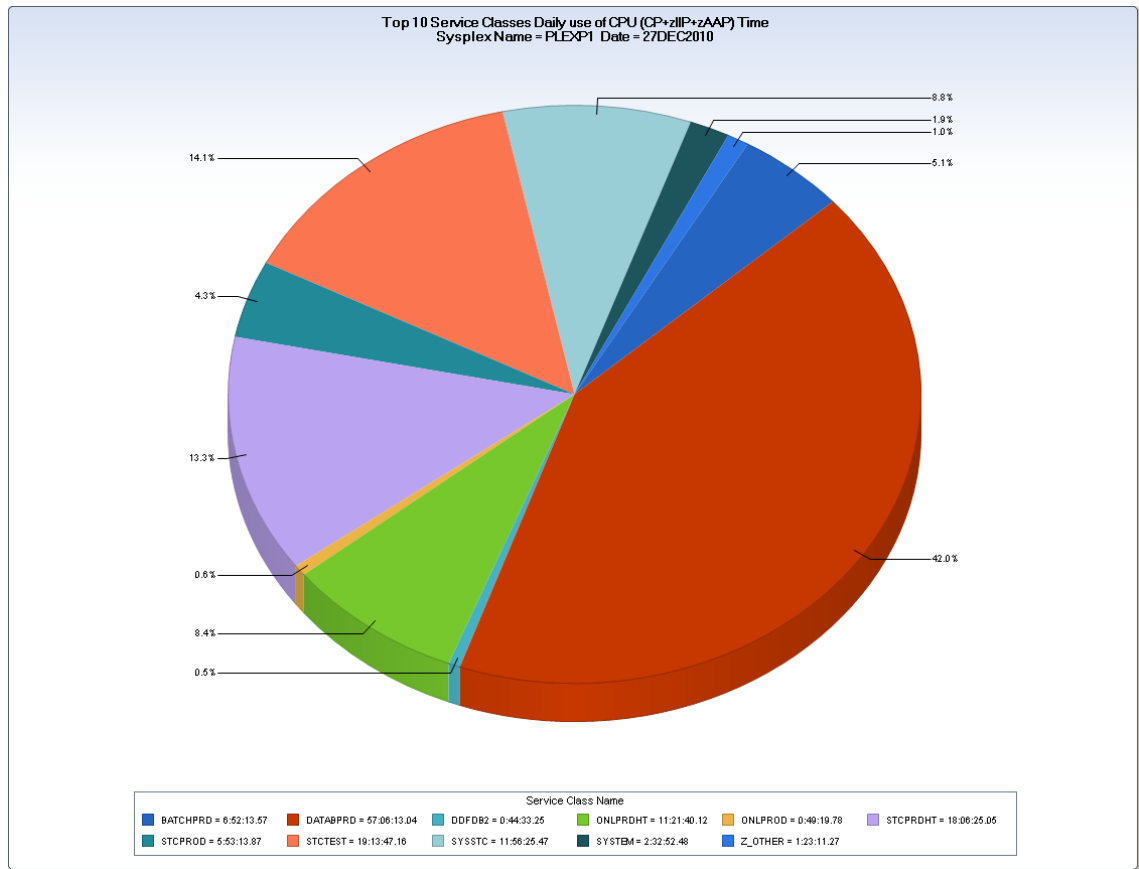
Modifications

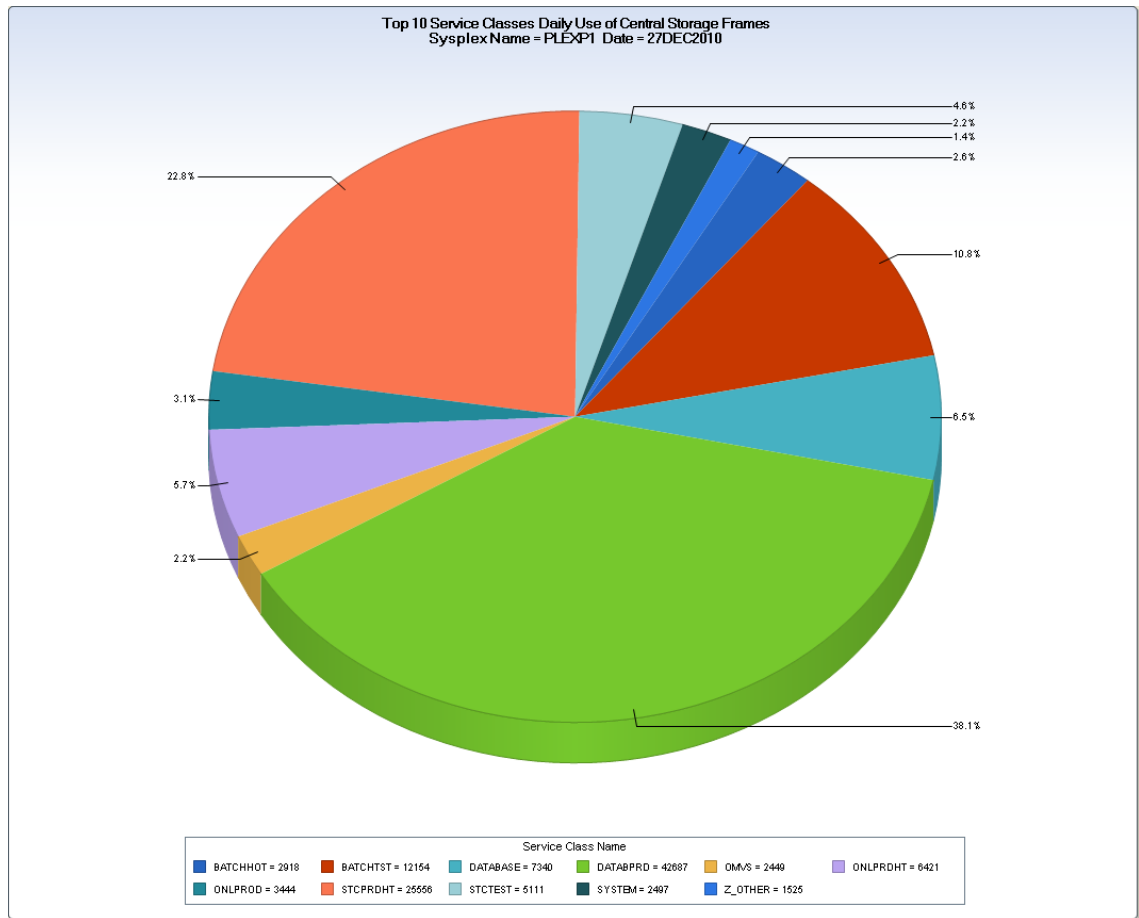
The following modifications can be made to the RMFSRV query:

- You can use the WLM_EC and WLM_ED DETAIL time spans to see interval level Service Class Period activity and performance. If DETAIL is selected, it should be selected for both files. If DETAIL is selected, replace the X-axis variable DATEHOUR with DATEHHMM in the Key Selection of all three SUMMARIZATION and DATA EXTRACT steps.
- You can increase or decrease the number of WLM_EC and WLM_ED file cycles to create charts that show more or fewer days.
- If you want to restrict the charts to show data from the day before exclusively, uncomment the SAS statement in the USER CODE task in both the WLM_EC and WLM_ED File Selection steps.
- You can use filter statements in the WLM_EC and WLM_ED File Selection steps to restrict data to particular time ranges (for example, HOUR GE 8 AND HOUR LE 17) or particular time zones (for example, ZONE EQ 1). If you do provide FILTER statements, they should be made in both the WLM_EC and WLM_ED File Selection steps.
- You can add filter statements in the WLM_EC and WLM_ED File Selection steps to select a particular SYSPLEX, CPCID, and/or WLMCLASS. If you do provide FILTER statements, they should be made in both the WLM_EC and WLM_ED File Selection steps.
- If your SYSPLEXES do not span CPCs, you can delete one of the first two data extract steps. If you want the CPC identified in each chart heading, delete the Service Class by SYSPLEX data extract.
- For the first data view, CPU and PI, the left Y-axis is scaled for five hours of CPU time. You can adjust this scale higher or lower to match the actual CPU usage of your Service Class Periods. You can also allow the axis to adjust dynamically for each chart by un-checking the FIXED button for the left Y-axis in the EDIT VIEW window.

You can use the extensive Chart View Edit capabilities to modify Y axis ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

RMFS10 – z/OS Daily Top 10 Service Classes by SYSPLEX





Overview

The RMFSVT query displays several pie charts showing a daily view of the top ten Service Class users of several important resources—CPU, I/O, and Central Storage. For the CPU usage pie chart, CP, zIIP, and zAAP CPU times are aggregated into a single metric. The metric shows total CPU utilization across all engine types.

The chart heading information shows the SYSPLEX Name, DATE, and resource unit (CPU, I/O, or Central Storage) charted. Three different data extract views are provided, each one showing the top ten users of a resource. An artificial 11th Service Class, Z_OTHER, is created and used to display the aggregated resource use for the remaining (11th through *n*th) Service Classes.

Note: If you want to see the top *nn* resource users, where *nn* is a number other than 10, see the Modifications documentation.

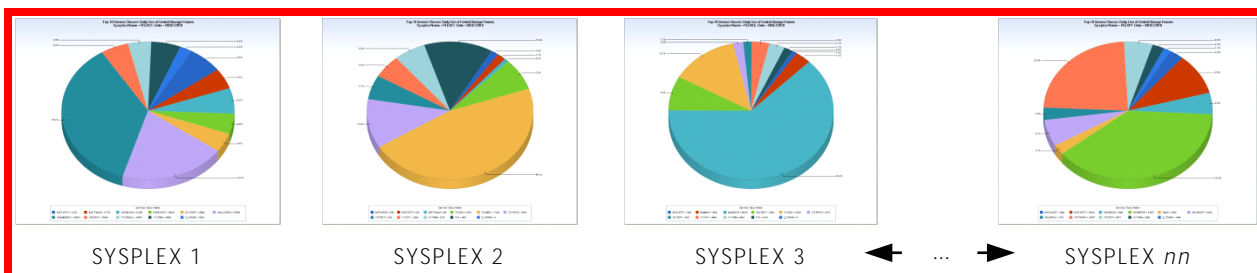
The first chart shows the top ten Service Classes consuming CPU time for the day in SYSPLEX PLEXP1. The second chart shows the top ten Service Class highest average consumers of central storage page frames.

Uses

The pie charts generated by the RMFSVT query provide a good way to identify quickly the Service Classes that dominate resource utilization across your WLM SYSplexes. Identifying these high use resource consumers allows you to focus performance improvement efforts where they provide the greatest return.

Data Extracts

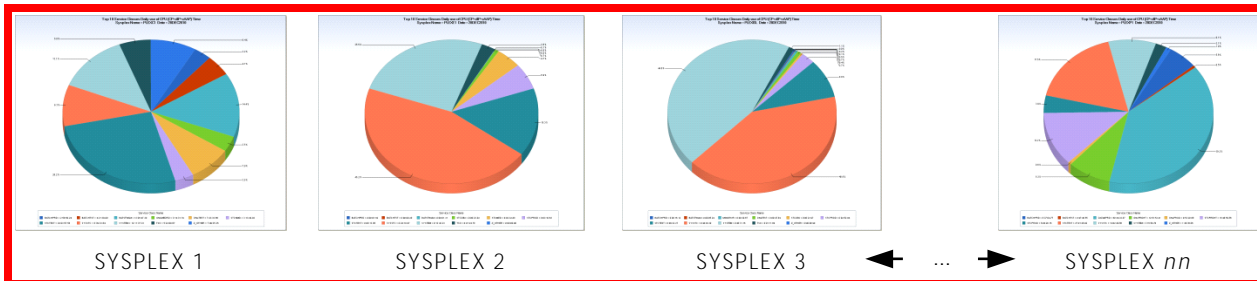
zOS Daily Top 10 Service Class Storage



Pie Slices

SECAVCRS – Average Storage Used – CS

zOS Daily Top 10 Service Class CPU Time

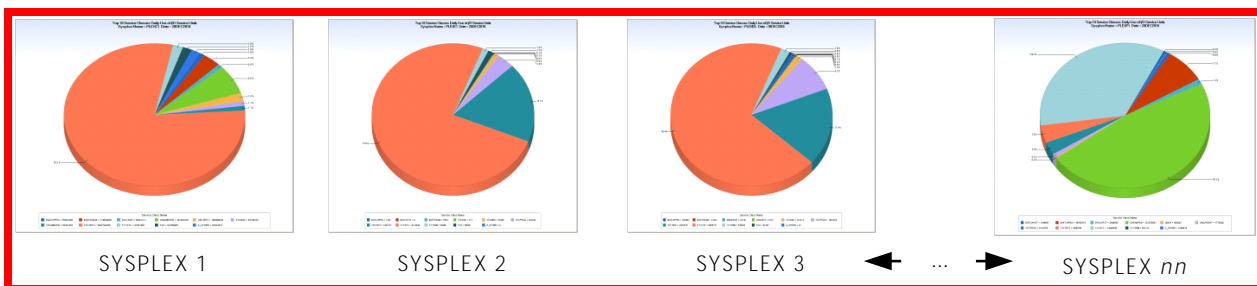


Pie Slices

TOTCPU – Total CP zIIP and zAAP CPU Time

Note: TOTCPU is the sum of CP, Normalized zIIP, and Normalized zAAP CPU Times.

zOS Daily Top 10 Service Class IO SUs



Pie Slices

SECTOIOC – I/O Service Units

Query Defaults

The following are RMFS10 Query Defaults:

- HARCPU DAYS timespan file: Cycle 01 (used to determine CPCID)
- WLMSEC DAYS timespan file: Cycle 02
- User code only accepts data from the day before (IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

CPCID

```
TRIM(CPUMOD) || ' - ' || TRIM(CPCMODID) || ' - Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

DATE

```
DATEPART(ENDTS);
```

TOTCPU

```
TOTCPU = SUM(SECTCPTM,SECZAPT,SECSUPTM) ;
```

TOTNN

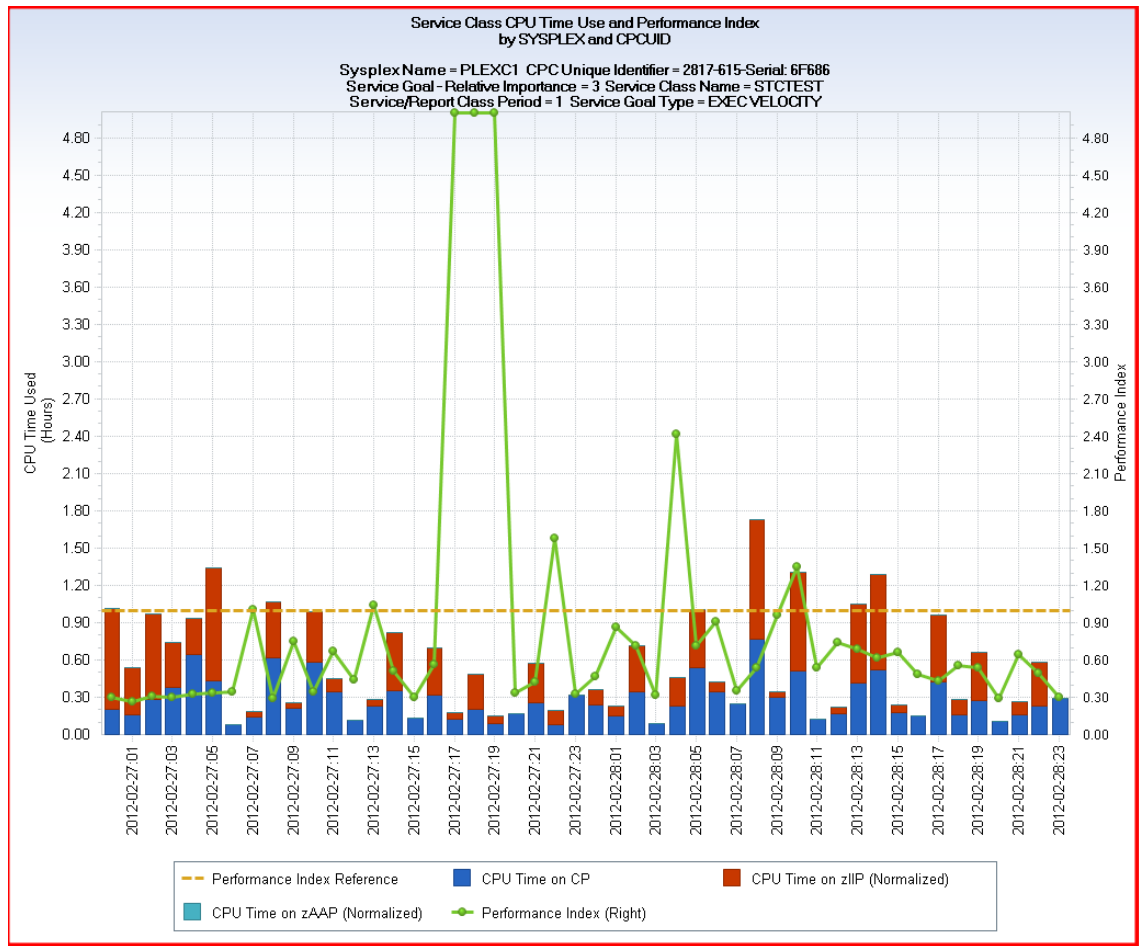
```
TOPNN = 10 ;
```

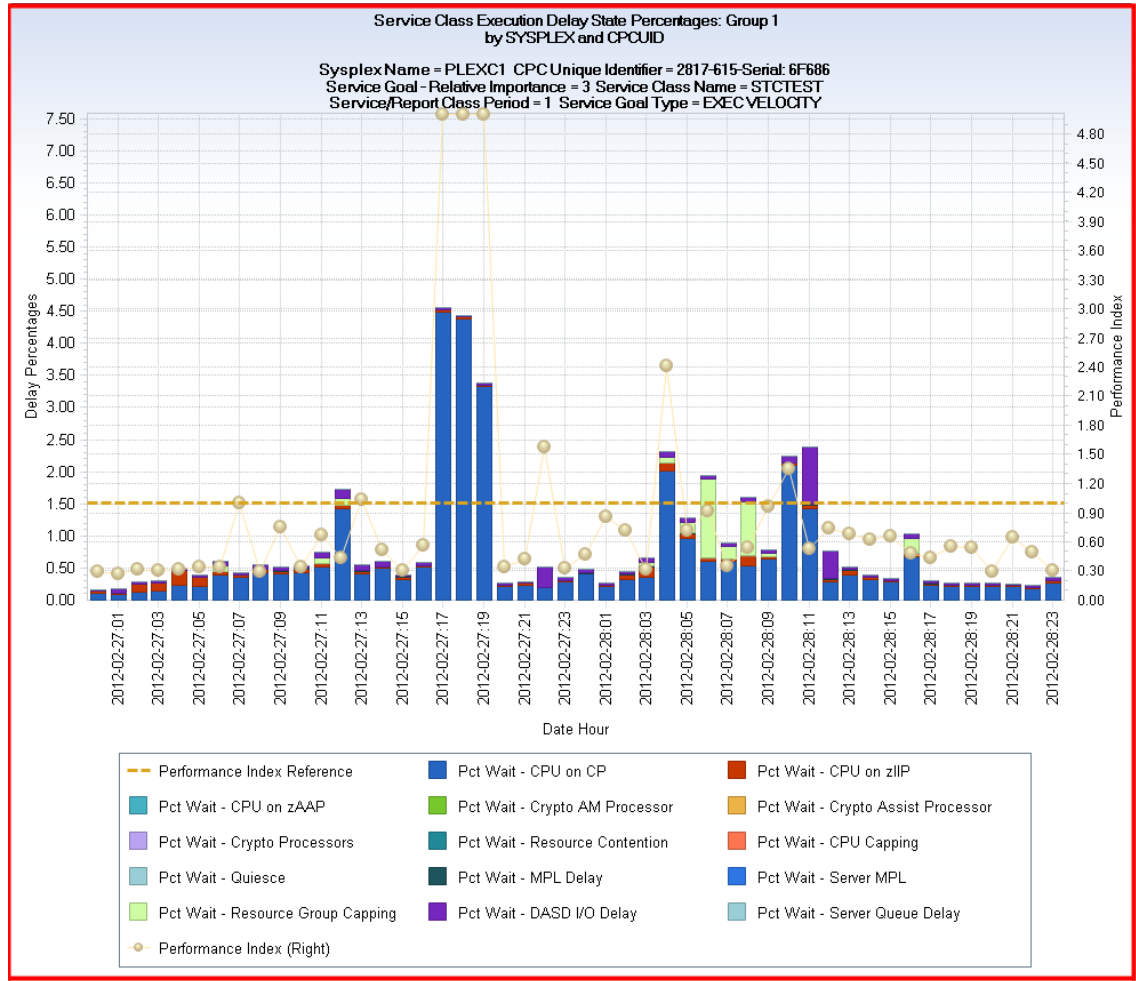
Modifications

The following modifications can be made to the RMFSVT query:

- You can show the top *nn* resource users, where *nn* is an integer number other than 10, by changing the value that is assigned to derived element TOPNN in the Input WLMSEC File step. For example, changing TOPNN=10 to TOPNN=5 creates Pie Charts that show the top five resource consuming Service Classes.
- You can input multiple cycles of the WLMSEC DAYS timespan file to see Top 10 Pie Charts for multiple days. The top ten resource users are determined independently for each unique DATE. If you do increase the number of WLMSEC DAYS timespan cycles, comment out the User Code – Select Yesterday in the Input WLMSEC File step.
- You can use filter statements in the WLMSEC File Selection step to restrict data to particular time ranges (for example, HOUR GE 8 AND HOUR LE 17) or particular time zones (for example, ZONE EQ 1).
- You can add filter statements in the WLMSEC File Selection step to select particular SYSPLEXs, CPCIDs, and/or WLMCLASSs.

RMFSRV – z/OS Service Class Analysis by SYSPLEX, CPCID, and SYSID





Overview

The RMFSRV query displays resource usage performance indicators, summary sampled state percentages, and detailed execution delay state information for each Workload Manager (WLM) Service Class Period. This information is provided by the following three data extract CSV's at the z/OS LPAR (SYSID), Central Processing Complex (CPC), and SYSPLEX levels:

- Service Class by SYSPLEX
- Service Class by SYSPLEX CPCUID
- Service Class by SYSPLEX CPCUID SYSID

Each data extract generates the following four chart views:

1. CPU and PI
2. Sampling Summary
3. Exec Delay States Group 1
4. Exec Delay States Group 2

The title area for all RMFSRV query charts shows the SYSPLEX name, Service Class Name, Relative Importance, Goal Type, and Service Class period charted. As delivered, each RMFSRV query chart shows the hourly activity for the past two days. For each SYSPLEX, the Service Class Periods are presented in Relative Importance order, with the most important Service Classes presented first.

Chart view #1, CPU and PI, the general processor CP, zIIP, and zAAP CPU time used by the Service Class Period (SCP). The CPU times used for each processor engine type are stacked and are scaled using the left Y-axis. Two lines are plotted against the right Y-axis scale - the Performance Index (PI) for the SCP, and a Performance Index "goal met" reference line, set to 1.

To prevent scaling issues, PI values greater than 5 are set to 5. PI values between zero (0) and one (1) mean that the Service Class exceeded the specified performance goal. A value equal to 1 means that the goal was met exactly. Values greater than 1 means that the specified performance goal was not met. The PI cannot be computed for certain goal types such as DISCRETIONARY and SYSTEM SPECIFIED.

The first chart sample was charted from the data extract: Service Class by SYSPLEX CPCUID. Shown is a two day hourly chart for Period 1 of Service Class STCTEST. This Service Class executed in SYSPLEX PLEXC1, and the information displayed occurred on z/OS LPARS executing on an IBM 2817-615 CPC.

Workloads in the STCTEST Service Class use both CP and zIIP processor engines, as shown by the blue and red stacked bars. Service Class STCTEST has an execution velocity goal type. The actual performance index value, shown with the beige line, is generally below the reference value of 1, meaning that the actual performance of workloads running in this Service Class generally exceeded the specified goal. But from 5 PM to 7 PM on the first day, the PI value was 5, meaning the goal was severely missed.

Chart view #2, Sampling Summary, displays the percentage of times workloads for the SCP were found to be in one of three states: Using, Wait-Unknown, and Execution Delays. The percentage of each state is stacked using vertical bars and are scaled using the left Y-axis fixed to show values from 0 to 100 percent. A fourth state which is not shown, if added to the chart would bring the stacked bars up to the 100% line, is the percentage of samples where workloads in the SCP were idle.

The Using state represents the percentage of samples where the workloads were using the CPU, and if I/O Priority is active, performing I/O. The Wait-Unknown state represents the percentage of workload delay states that WLM does not manage. These delays include time waiting on an I/O, DB2 latches, tape mount, enqueues, and others. The Execution Delay state represents the percentage of samples where the SCP workloads experienced execution delays for resources that WLM does manage.

Two lines are plotted against the right Y-axis scale—the performance index (PI) for the Service Class period, and a performance index "goal met" reference line, set to 1.

Chart views #3 and #4, Exec Delay States Group 1 and Exec Delay States Group 2, show the percentage of samples belonging to the various categories of execution delays. There are too many categories to effectively display using a single chart, so two chart views are provided, Group 1 and Group 2. The sample percentages of each delay category are stacked using vertical bars against the left Y axis. The axis scale is dynamically adjusted based on the percentages encountered in the data. As with the other chart views, two lines are plotted against the right Y-axis scale—the performance index (PI) for the Service Class period, and a performance index "goal met" reference line, set to 1.

The second chart sample above was displayed by selecting view #3, Exec Delay States Group 1. This view shows the first set of execution delay category percentages for Period 1 of the Service Class STCTEST. You can see that the reason the SCP goal was missed for hours 5 PM through 7 PM on the first day was due to an increase in delays for the STCTEST workloads while waiting to be dispatched on a CP processor.

Uses

All workloads running on z/OS are assigned to a Service Class, and each Service Class is defined with a level of importance. Most Service Classes are also assigned a goal type (for example, Execution Velocity, Response Time Percentile, and so on) and a performance goal to meet. The z/OS Workload Manager (WLM) manages physical resources (for example, CPU) and uses workload dispatching strategies to meet Service Class goals. Service classes with a higher relative importance value (0 high <--> 5 low) are given preferential treatment.

All three data extracts generated by the RMFSRV query are organized by SYSPLEX, and charts are produced for each Service Class Period. View # 1, CPU and PI show the CPU utilization and Performance Index (PI) for each HOUR. The PI indicates how well the workload performed against the designated goal.

This chart view provides the information required to understand the performance and resource consumption of z/OS workloads. Using this data view, you can monitor the performance of each Service Class Period defined in your WLM Service Policies and Definitions. If a Service Class is routinely exceeding the assigned goal (for example, the PI is consistently 0.8 or lower), or missing the assigned goal (for example, the PI is consistently higher than 1), you can adjust the goal or modify other Service Classes to free up additional resources for the troubled Service Class.

While CPU time is the default resource usage metric plotted, the Data Extract CSV provides a number of other resources usage metrics that can be alternatively selected for charting. These metrics include DASD Service time, paging rates, and average storage used.

View #2, Sampling Summary, allows you to quickly determine whether poorly performing SCPs are being delayed by WLM managed resources, or for other reasons. If you see increased WLM execution delays at times when the SCP goals are missed, you can examine Views #3 and #4 to determine the specific types of execution delays that impacted performance. The resources represented by these delay categories are managed by WLM. The delays experienced by Service Classes can be alleviated by adjusting Service Policy Service Class goals or by adding additional capacity of the resource type indicated.

Note: By design, SYSPLEXES can span Central Processing Complexes. If your SYSPLEXES do not span CPCs, then the first two data extracts are identical, with the exception that the second data extract, Service Class by SYSPLEX CPCID, identifies the CPC in each chart title. If your SYSPLEXES do span CPCs, then the Service Class by SYSPLEX CPCID data extract displays multiple charts for a Service Class that ran on multiple CPCs.

Query Defaults

The following are RMFSRV Query Defaults:

- HARCPU DAYS timespan file: Cycle 01 (used to determine CPCID)
- WLMSEC DAYS timespan file: Cycles 01 - 02
- WLMSED DAYS timespan file: Cycles 01 – 02
- Filters: Empty Filter tasks provided as placeholder for user WLMSEC and WLMSED file filtering to exclude/select specific SYSPLEXs, SYSIDs, ZONES, and so on.
- Commented user code for WLMSEC and WLMSED file steps only accepts data from the day before (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

CPCUID

```
CPCUID = PUT(SYSID,$CPCFMT.) ;
```

DATE

```
DATEPART(ENDTS);
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

DATEHOUR

```
PUT(DATEPART(ENDTS),YMMDD10.) || ':' || PUT(HOUR,Z2.) ;
```

DATEHHMM

```
PUT(DATEPART(ENDTS),YMMDDP10.) || '-' || HHMM ;
```

SECFILE

```
1 ; Used after MERGE of WLMSEC and WLMSED files to limit charting to observations that contained WLMSEC file data.
```

SEDPCSAM

```
100 * SEDTOSAM / SEDTOSST;
```

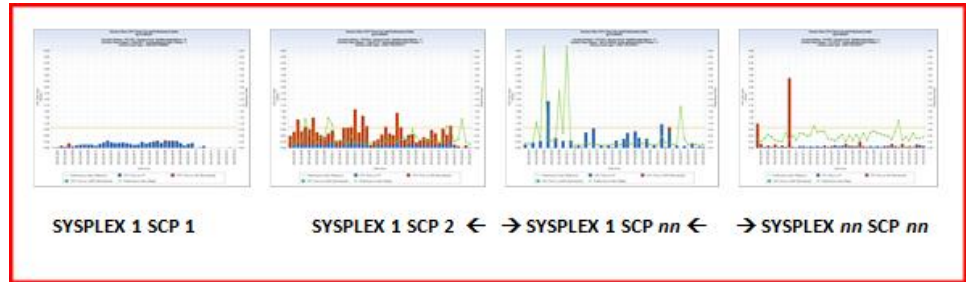
SEDPCUSE

```
100 * SEDTOUSE / SEDTOSST;
```

Data Extracts

Service Class by SYSplex

VIEW 1 of 4: WLMCLASS by SYSplex – CPU and PI



Note: RCP=Report Class Period

Left Y-axis

- SECCPUTM – CPU Time on CP
- SECSUPTM – CPU Time on zIIP (Normalized)
- SEZAPTM – CPU Time on zAAP (Normalized)

Right Y-axis

- SECPINDX – Performance Index
- Performance Index
- Reference (value that is set at 1)

X-axis

HHMM – Hour: Minute

VIEW 2 of 4: WLMCLASS by SYSplex – Sampling Summary



Left Y-axis

- SEDCUSE – Pct Total WLM Using Samples (derived in query code)
- SEDPCDUN – Pct Wait – Unknown
- SEDCSAM – Pct Total Execution Delay Samples (derived in query code)

Right Y-axis

- SECPINDX – Performance Index
- Performance Index Reference – value set to 1

X-axis

HHMM – Hour: Minute

VIEW 3 of 4: WLMCLASS by SYSPLEX – Exec Delay States Group 1



Left Y-axis

- SEDPCDCP – Pct Wait – CPU on CP
- SEDPCSUD – Pct Wait – CPU on zIIP
- SEDPCZPD – Pct Wait – CPU on zAAP
- SEDPCAMD – Pct Wait – Crypto AM Processor
- SEDPCAPD – Pct Wait – Crypto Assist Processor
- SEDPCCRD – Pct Wait – Crypto Processors
- SEDPCCOD – Pct Wait – Resource Contention
- SEDPCDCC – Pct Wait – CPU Capping
- SEDPCDQU – Pct Wait – Quiesce
- SEDPCDMP – Pct Wait – MPL Delay
- SEDPCSMP – Pct Wait – Server MPL
- SEDPCDRG – Pct Wait – Resource Group Capping
- SEDPCDSD – Pct Wait – DASD I/O Delay
- SEDPCSQU – Pct Wait – Server Queue Delay

Right Y-axis

- SECPINDX – Performance Index
- Performance Index Reference – value set to 1

X-axis

HHMM – Hour: Minute

VIEW 4 of 4: WLMCLASS by SYSPLEX – Exec Delay States Group 2



Left Y-axis

- SEDPCDES – Pct Wait – ESO Hiperspc Pg-In from Aux
- SEDPCDHS – Pct Wait – Std Hiperspc Pg-In from Aux
- SEDPCDPC – Pct Wait – CSA Page-In from Aux
- SEDPCDPP – Pct Wait – Pvt Page-In from Aux
- SEDPCDSI – Pct Wait – Swap-In
- SEDPCDVI – Pct Wait – VIO Page-In from Aux
- SEDPCDXM – Pct Wait – Cross Mem Page-In from Aux
- SEDPCSHP – Pct Wait – Server Hiperspace Paging
- SEDPCSP – Pct Wait – Server Private Area Paging
- SEDPCSSI – Pct Wait – Server Swap-In
- SEDPCSV – Pct Wait – Server Space VIO Paging

Right Y-axis

- SECPINDX – Performance Index
- Performance Index Reference – value set to 1

X-axis

- HHMM – Hour: Minute

Service Class by SYSPLEX CPCUID

Data extract Service Class by SYSPLEX CPCUID provides the same four Q&R chart views, with charts displaying the same Y and X axis charted data elements, as those described for data extract Service Class by SYSPLEX.

The only difference is that additional charts are generated if the SYSPLEX spans multiple CPCs. Where the chart granularity for the SYSPLEX level data extract followed this pattern:

SYSPLEX1 SCP 1 SYSPLEX1 SCP 2 ← → SYSPLEX1 SCP nn ← → SYSPLEXnn SCP nn

The chart granularity for the Service Class by SYSPLEX CPCUID data extract follows this pattern:

SYSPLEX1 CPC 1 SYSPLEX 1 CPC 1 ←... → SYSPLEX1 CPC 1 ←... → SYSPLEXnn CPC nn
SCP 1 SCP 2 SCP nn SCP nn

If a SYSPLEX spans two or more CPCs, the the SYSPLEX CPCUID level data extract will generate more unique charts that the SYSPLEX level data extract.

If a SYSPLEX does not span CPCs, however, the number of unique charts generated by the SYSPLEX level and SYSPLEX CPCUID level data extracts will be identical.

Service Class by SYSPLEX CPCUID SYSID

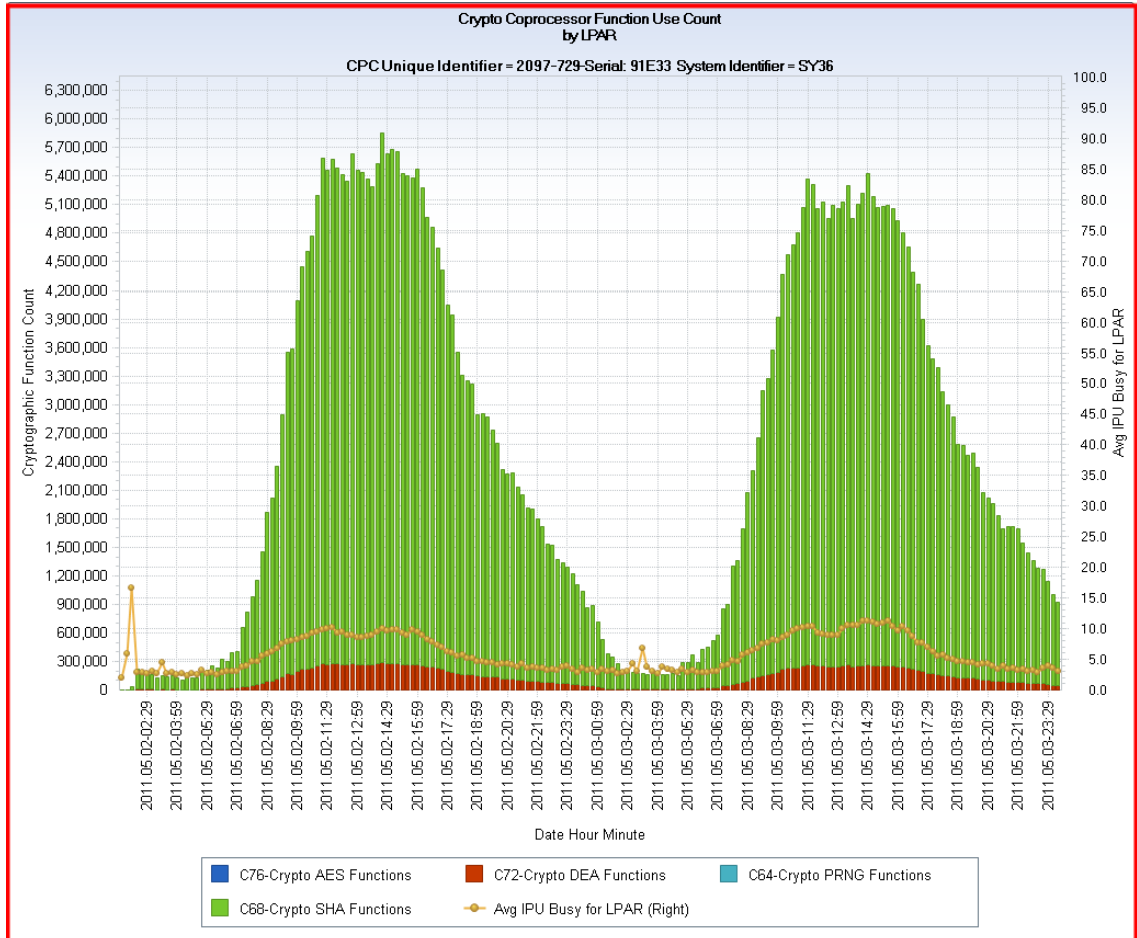
Modifications

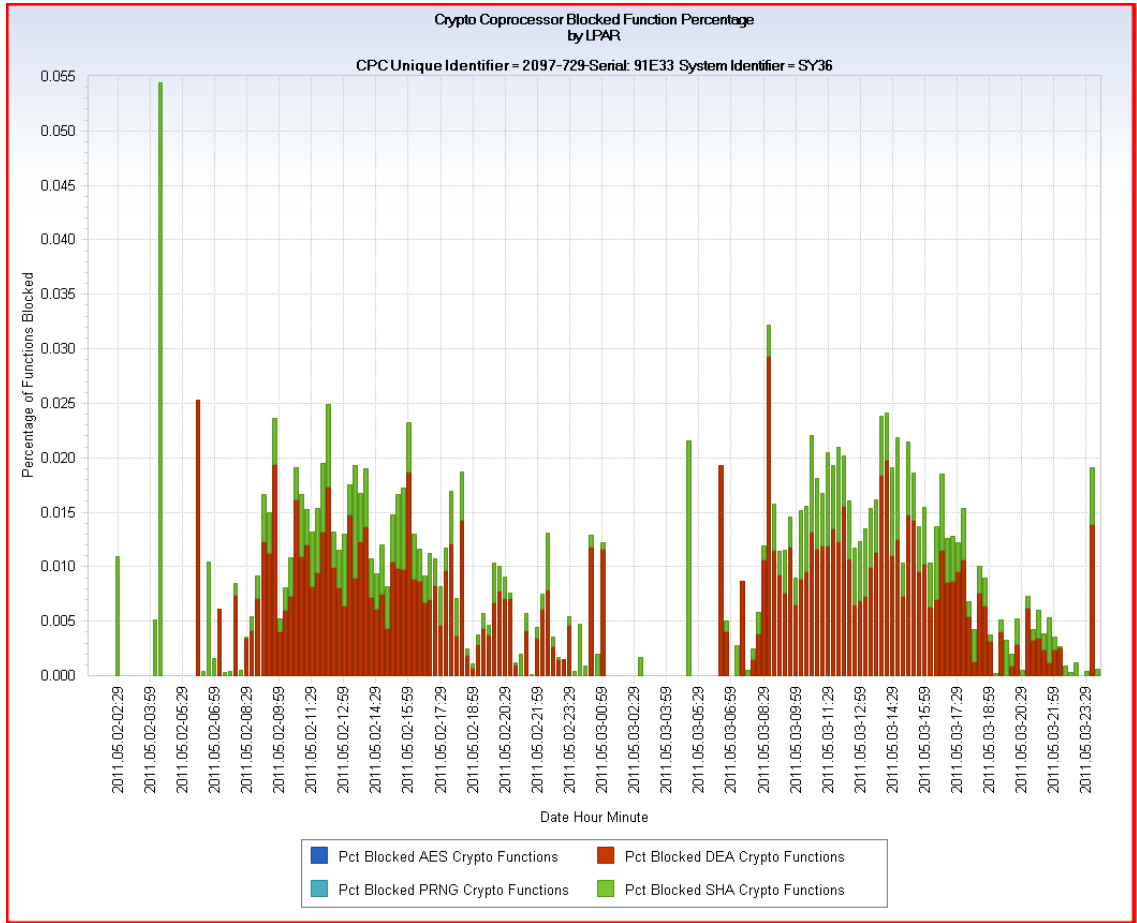
The following modifications can be made to the RMFSRV query:

- You can use the WLMSEC and WLMSED DETAIL time spans to see interval level Service Class Period activity and performance. If DETAIL is selected, it should be selected for both files. If DETAIL is selected, replace the X-axis variable DATEHOUR with DATEHHMM in the Key Selection of all three SUMMARIZATION and DATA EXTRACT steps.
- You can increase or decrease the number of WLMSEC and WLMSED file cycles to create charts that show more or fewer days.
- If you want to restrict the charts to show the data from the day before exclusively, uncomment the SAS statement in the USER CODE task in both the WLMSEC and WLMSED File Selection steps.
- You can use filter statements in the WLMSEC and WLMSED File Selection step to restrict data to particular time ranges (for example, HOUR GE 8 AND HOUR LE 17) or particular time zones (for example, ZONE EQ 1). If you do provide FILTER statements, they should be made in both the WLMSEC and WLMSED File Selection steps.
- You can add filter statements in the WLMSEC and WLMSED File Selection steps to select a particular SYSPLEX, CPCID, and/or WLMCLASS. If you do provide FILTER statements, they should be made in both the WLMSEC and WLMSED File Selection steps.
- If your SYSPLEXES do not span CPCs, you can delete one of the first two data extract steps. If you want the CPC identified in each chart heading, delete the Service Class by SYSPLEX data extract.
- For the first data view, CPU and PI, the left Y-axis is scaled for five hours of CPU time. You can adjust this scale higher or lower to match the actual CPU usage of your Service Class Periods. You can also allow the axis to adjust dynamically for each chart by unchecking the FIXED button for the left Y-axis in the EDIT VIEW window.
- You can use the extensive Chart View Edit capabilities to modify Y axis ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

Hardware Instrumentation Services CPU Measurement Facility

RMFCRL - CPU MF Cryptographic Coprocessor Activity LPAR Level





Overview

The RMFCRL query produces charts that show cryptographic coprocessor use by workloads executing on z/OS LPARs running on IBM z10 and later generation Central Processing Complexes (CPCs), such as zEC12. In addition to showing cryptographic function use, these charts quantify delays incurred because coprocessors were busy, and the average number of wasted machine cycles lost due to these delays. The RMFCRL query creates the following CSV output file:

Crypto Coprocessor LPAR Level Activity

The data extract generates the following three chart views:

1. Crypto Coprocessor LPAR – Function Use Count
2. Crypto Coprocessor LPAR – Blocked Function Pct
3. Crypto Coprocessor LPAR – Avg Penalty Cycles per Blocked Function

Chart view #1 displays overall cryptographic function use by type:

AES

Advanced Encryption Standard

DEA

Data Encryption Algorithm

PRNG

Pseudo-Random Number Generator

SHA

Secure Hash Algorithm

The overall count of each function type executed is displayed using stacked vertical bars and the left Y axis. The average LPAR Individual Processor Unit (IPU) busy is displayed using a horizontal line and the right Y axis. The first chart sample above was produced by inputting two cycles (days) of the DETAIL timespan HARCML file. This chart, for LPAR 'SY36' running on an IBM 2097-729 (z10 CPC with 29 CP engines), shows that for each of the two days, cryptographic coprocessor function execution starts ramping up about 7 AM, peaks between 11 AM and 4 PM, then decreases through the late afternoon and evening hours. Two cryptographic functions are executed by the workloads running on LPAR 'SY36', DEA and SHA, with the vast majority being SHA (green bar) functions. The average logical IPU busy percentage, which represents the percentage of available machine cycles used by dispatched logical processors, never exceeds 12 percent.

Chart view #2 displays the percentage of each function type that was blocked from execution, using stacked vertical bars and the left Y axis. Crypto functions blocking occurs when the cryptographic coprocessor is busy. The second chart sample above, shows that for each of the two days, the blocked function percentage was very low, far less than 1/10th of 1 percent. Even though the first chart showed that the majority of crypto functions executed were SHA functions, the majority of blocked functions were DEA

Chart view #3 displays the average number of penalty cycles incurred per function block, for each function type using vertical (non-stacked) bars and the left Y axis. Penalty cycles are wasted machine cycles expended while the function is waiting for the cryptographic coprocessor.

By default, the query is executed against the DETAIL timespan HARCML file, and each X axis value represents data from an RMF interval. The X axis metric, DATE HOUR MINUTE, is constructed by concatenating the DATE, HOUR, and MINUTE from the end time boundary of each interval.

Uses

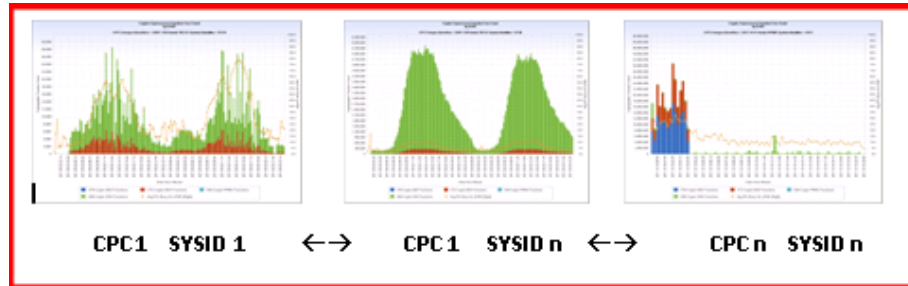
The processor chip design of IBM z10 and later generation CPCs includes cryptographic coprocessor chips that service cryptographic function requests issued by workloads executing on the IPU. Many modern applications, particularly those servicing web based transactions, require data encryption. To avoid delays, it is important that encryption requests do not experience high blockage rates.

The charts generated by this RMFCRL query provide the information necessary to monitor the data encryption demand of each LPAR, and to identify LPARs that are experiencing unacceptable percentages of blocked requests. Shifting workloads to other LPARs or adding additional logical processors IPUs to an LPAR can help alleviate elevated blocked function percentages.

Data Extracts

Crypto Coprocessor LPAR Level Activity

VIEW 1 of 3: Crypto Coprocessor LPAR – Function Use Count



Left Y axis

- CMLC76 – C76 – Crypto AES Functions
- CMLC72 – C72 – Crypto DEA Functions
- CMLC64 – C64 – Crypto PRNG Functions
- CMLC68 – C68 – Crypto SHA Functions

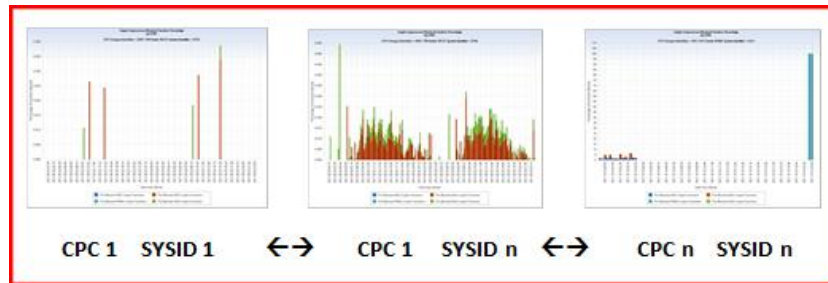
Right Y axis

AVGPCBSY – Avg IPU Busy for LPAR

X axis

DATEHHMM – Date: Hour : Minute

VIEW 2 of 3: Crypto Coprocessor LPAR – Blocked Function Pct



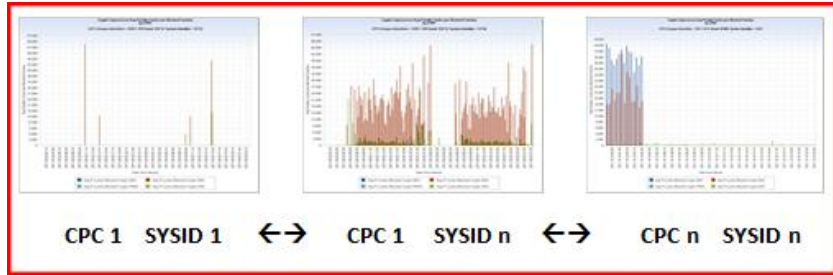
Left Y axis

- CMLPCCBA – Pct Blocked AES Crypto Functions
- CMLPCCBD – Pct Blocked DEA Crypto Functions
- CMLPCCBP – Pct Blocked PRNG Crypto Functions
- CMLPCCBS – Pct Blocked SHA Crypto Functions

X axis

DATEHHMM – Date: Hour: Minute

VIEW 3 of 3: Crypto Coprocessor LPAR – Avg Penalty Cycles per Blocked Function



Left Y axis

- CMLAVCBA – Avg P-Cycles Blocked Crypto AES
- CMLAVCBD – Avg P-Cycles Blocked Crypto DEA
- CMLAVCBP – Avg P-Cycles Blocked Crypto PRNG
- CMLAVCBS – Avg P-Cycles Blocked Crypto SHA

X axis

DATEHHMM – Date: Hour: Minute

Query Defaults

The following are RMFCRL Query Defaults

- HARCML DETAIL time span file: Cycle 01
- Filters: Empty Filter task provided as placeholder for user HARCML file filtering to exclude/select specific LPARs
- Commented user code , that if uncommented will only accept yesterday's data
 - *IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

DATE

DATEPART(ENDTS) ;

HHMM

PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);

DATEHOUR

PUT(DATEPART(ENDTS),YYMMDDD10.) || ':' || PUT(HOUR,Z2.) ;

DATEHHMM

PUT(DATEPART(ENDTS),YYMMDDP10.) || '-' || HHMM ;

AVGPCBSY

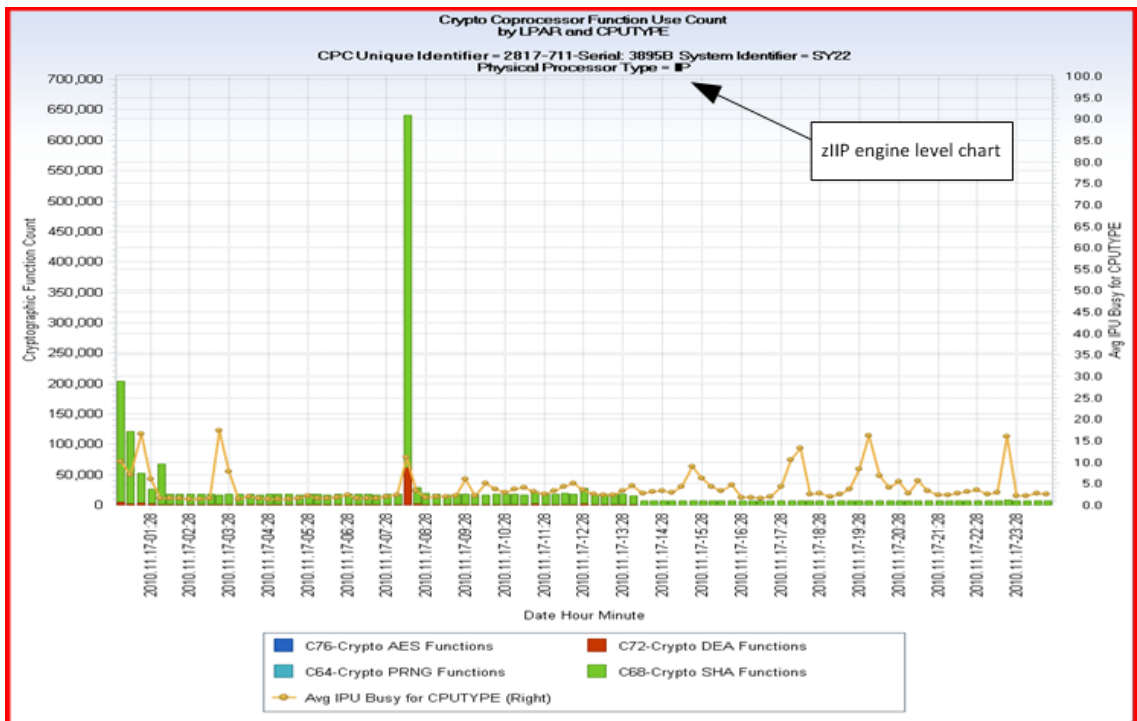
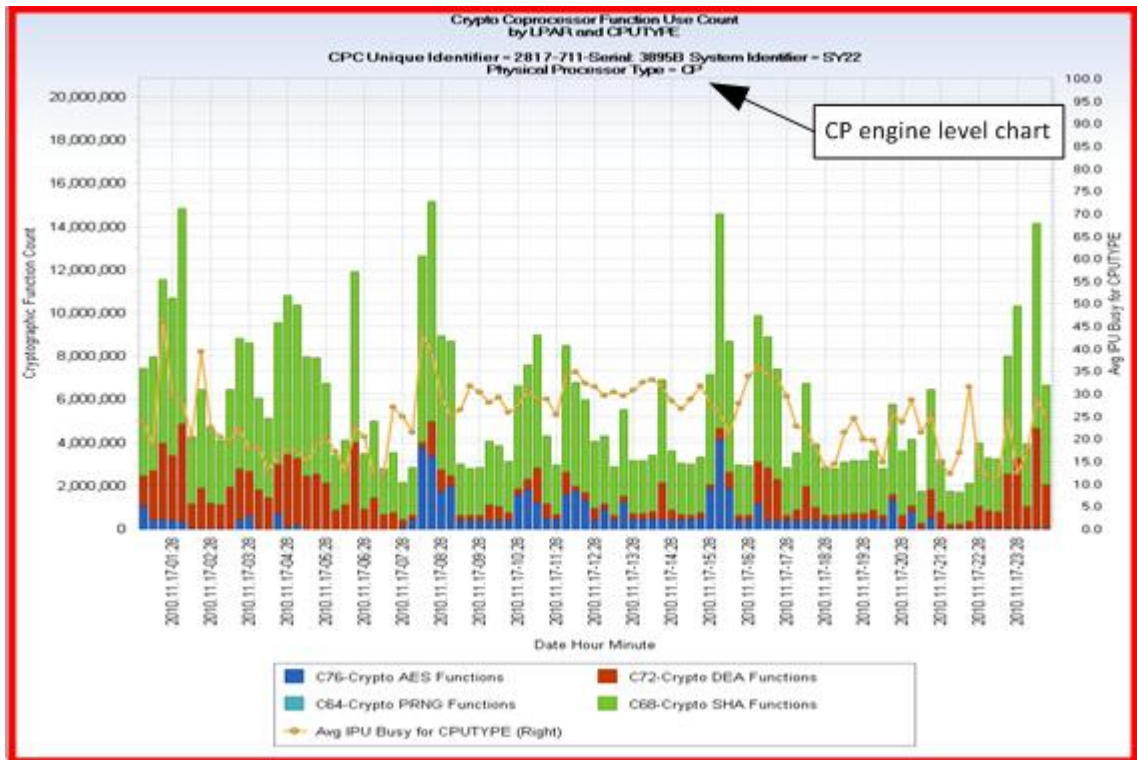
IF CMLAVIPU THEN AVGPCBSY=CMLPCBSY / CMLAVIPU ;

Modifications

The following modifications can be made to the RMFCRL query:

- You can input multiple cycles of DETAIL timespan HARCML data to create longer duration charts.
- You can input the DAYS, WEEKS, or higher timespan HARCML files, but will need to modify the Key Selection data elements in the Data Extract Steps. For example, if you input the DAYS timespan HARCML file, you will want to change DATEHHMM to DATEHOUR.
- If you want to restrict the charts to always show yesterday's data exclusively, uncomment the SAS statement in the USER CODE-Select Yesterday task in the HARCML File Selection step.
- You can use filter statements in the HARCML File Selection step to restrict data to particular time ranges (e.g., HOUR GE 8 AND HOUR LE 17) or particular time zones (e.g., ZONE EQ 1),
- You can add filter statements in the HARCML File Selection step to select a particular CPCID or LPAR (SYSID).
- You can use the extensive Chart View Edit capabilities to modify Y axis value ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

RMFCRT - CPU MF Cryptographic Coprocessor Activity CPUTYPE Level



Overview

The RMFCRT query produces charts that show cryptographic coprocessor use by workloads executing on different types of IPU (Individual Processor Units) assigned to z/OS LPARs running on IBM z10 and later generation Central Processing Complexes (CPCs), such as zEC12. z/OS LPARs can be assigned three different IPU types:

CP

Central Processors – General processing unit

zAAP

System z Application Assist Processor – specialty engine for JAVA workloads

zIIP

System z Integrated Information Processor – specialty engine for enclave exploiting workloads

In addition to cryptographic function use, these charts quantify delays incurred because the coprocessors were busy, and the average number of machine cycles wasted due to these delays. The RMFCRT query creates the following CSV output file:

[Crypto Coprocessor CPUTYPE Level Activity](#)

The data extract generates the following three chart views:

1. Crypto Coprocessor CPUTYPE – Function Use Count
2. Crypto Coprocessor CPUTYPE – Blocked Function Pct
3. Crypto Coprocessor CPUTYPE – Avg Penalty Cycles per Blocked Function

For information about each of these chart views see the RMFCRL – CPU MF Cryptographic Coprocessor Activity LPAR Level query description. The difference between the charts created by the RMFCRL LPAR level query and the RMFCRT CPUTYPE level query is in the chart granularity. The RMFCRL query views generate a separate chart for each LPAR, but the RMFCRT CPUTYPE level query views generate a separate chart for each IPU type (as for example, CP, zAAP, or zIIP) assigned to each LPAR.

The two chart samples above previous two chart samples show charts generated by chart view #1 – Crypto Coprocessor CPUTYPE – Function Use Count for a single LPAR, 'SY22.' LPAR 'SY22' is a z/OS LPAR running on an IBM 2817-711 (z196 CPC with 11 CP engines). LPAR 'SY22' is defined with eleven logical CP engines and two logical zIIP engines.

Chart view #1 displays overall cryptographic function use by type:

AES

Advanced Encryption Standard

DEA

Data Encryption Algorithm

PRNG

Pseudo-Random Number Generator

SHA

Secure Hash Algorithm

The first chart sample shows that workloads running on the 11 logical CP processors execute SHA cryptographic functions (green bars) throughout the day. DEA function (red bars) execution is highest in the early morning and late evening hours, and AES function execution is highest between 8 AM and 8 PM. The second chart sample generated for the 2 logical zIIP processors, shows much lower cryptographic function execution except for a large spike at about 8 AM.

Uses

To understand the value provided by these cryptographic coprocessor utilization queries, see the query use information provided for the RMFCRL query. The RMFWCT query provides additional granularity that allows you to further refine your understanding of how your z/OS workloads make use of the cryptographic coprocessor chips in your z10, z114, or z196 CPCs.

If a particular CPUTYPE is experiencing a problem, adding additional logical processors of that type could help alleviate elevated blocked function percentages.

Data Extracts:

For a complete description of the data extracts generated by the RMFCRT query, see the RMFCRL – CPU MF Cryptographic Coprocessor Activity LPAR Level query information. The data extracts and views are identical in all aspects. The only differences are in the granularity of the chart images.

For the RMFCRL LPAR level charts, only one chart is generated for each LPAR, but for the RMFCRT query, one chart is generated for each CPUTYPE (e.g., CP, zAAP, zIIP) assigned to each LPAR.

Query Defaults

The following are RMFCRT Query Defaults:

- HARCMT DETAIL time span file: Cycle 01
- Filters: Empty Filter task provided as placeholder for user HARCMT file filtering to exclude/select specific LPARs
- Commented user code, that if uncommented will only accept yesterday's data.
 - *IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

DATE

```
DATEPART(ENDTS) ;
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

DATEHOUR

```
PUT(DATEPART(ENDTS),YYMMDDD10.) || ':' || PUT(HOUR,Z2.) ;
```

DATEHHMM

```
PUT(DATEPART(ENDTS),YYMMDDP10.) || '-' || HHMM ;
```

AVGPCBSY

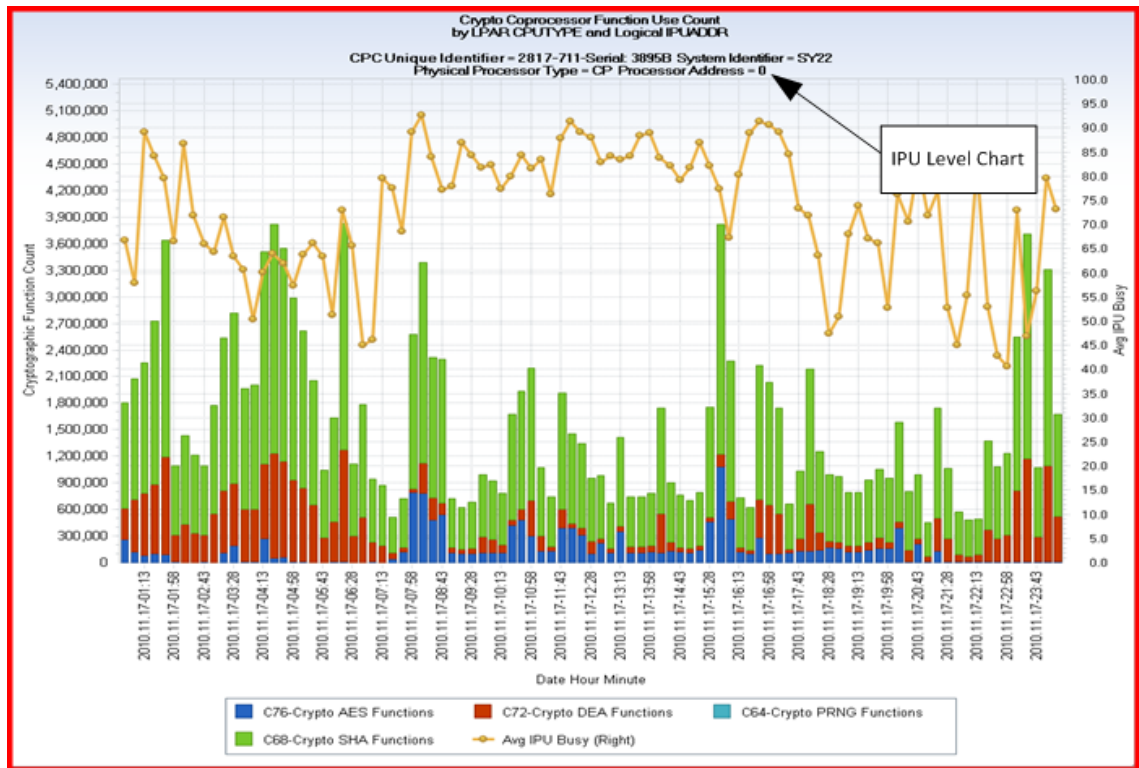
```
IF CMTAVIPU THEN AVGPCBSY=CMTPCBSY / CMTAVIPU ;
```

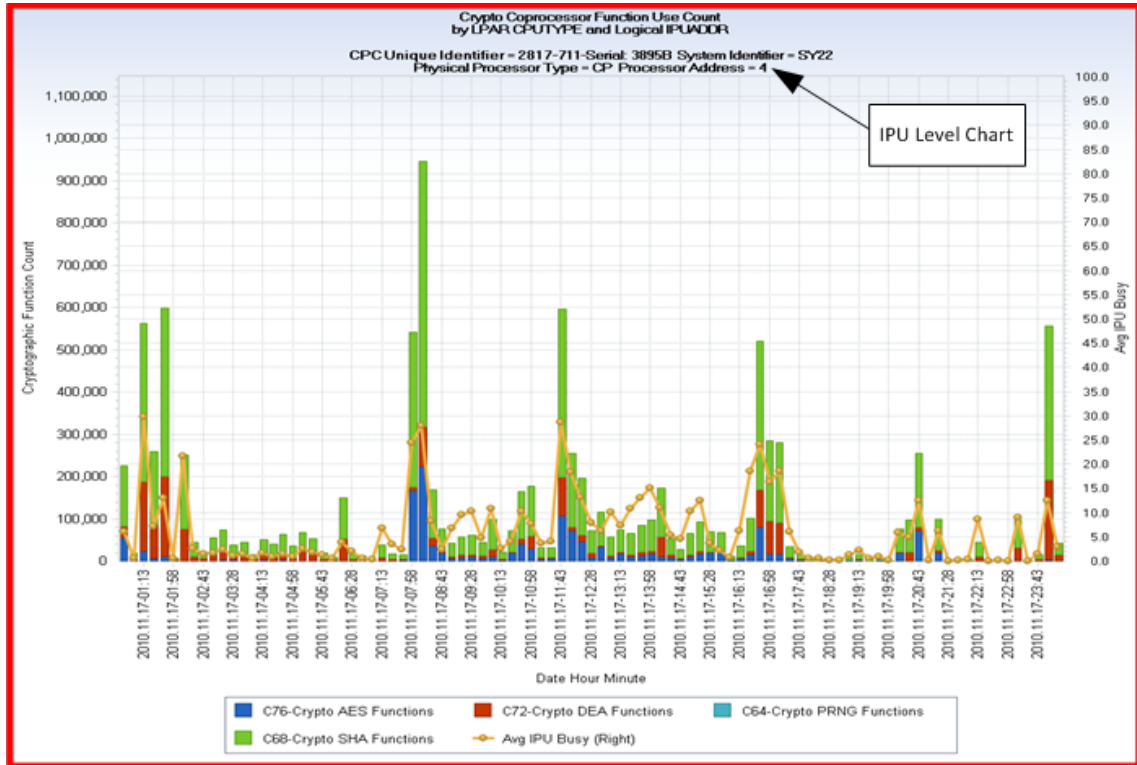
Modifications

The following modifications can be made to the RMFCRT query:

- You can input multiple cycles of DETAIL timespan HARCMT data to create longer duration charts.
- You can input the DAYS, WEEKS, or higher timespan HARCMT files, but will need to modify the Key Selection data elements in the Data Extract Steps. For example, if you input the DAYS timespan HARCMT file, you will want to change DATEHHMM to DATEHOUR.
- If you want to restrict the charts to always show yesterday's data exclusively, uncomment the SAS statement in the USER CODE-Select Yesterday task in the HARCMT File Selection step.
- You can use filter statements in the HARCMT File Selection step to restrict data to particular time ranges (e.g., HOUR GE 8 AND HOUR LE 17) or particular time zones (e.g., ZONE EQ 1),
- You can add filter statements in the HARCMT File Selection step to select a particular CPCID or LPAR (SYSID).
- You can use the extensive Chart View Edit capabilities to modify Y axis value ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

RMFCRE - CPU MF Cryptographic Coprocessor Activity IPU Level





Overview

The RMFCRE query produces charts that show cryptographic coprocessor use by workloads executing on each active logical IPU (Individual Processor Unit) assigned to z/OS LPARs running on IBM z10 and later generation Central Processing Complexes (CPCs), such as zEC12.

The RMFCRT query creates the following CSV output file:

Crypto Coprocessor IPU Level Activity

The data extract generates the following three chart views:

1. Crypto Coprocessor IPU – Function Use Count
2. Crypto Coprocessor IPU – Blocked Function Pct
3. Crypto Coprocessor IPU – Avg Penalty Cycles per Blocked Function

Refer to the RMFCRL – CPU MF Cryptographic Coprocessor Activity LPAR Level query description for information about each of these chart views. The only difference between the charts created by the RMFCRL LPAR level query and this RMFCRE IPU level query is one of granularity. Where the RMFCRL query views generate a single chart for each LPAR, the RMFCRE IPU level query views generate a separate chart for each active logical IPU (engine) assigned to each LPAR.

The two chart samples above show charts generated by chart view #1 – Crypto Coprocessor IPU – Function Use Count for two different logical IPU's assigned to LPAR, 'SY22.' LPAR 'SY22' is a z/OS LPAR running on an IBM 2817-711 (z196 CPC with 11 physical CP engines). LPAR 'SY22' is defined with eleven logical CP engines and two logical zIIP engines.

Chart view #1 displays overall cryptographic function use by type:

AES

Advanced Encryption Standard

DEA

Data Encryption Algorithm

PRNG

Pseudo-Random Number Generator

SHA

Secure Hash Algorithm

The top chart sample above shows the cryptographic coprocessor function use by the logical CP engine with IPU address 0. The average IPU busy is above 50% for most of the day, with heavy cryptographic function use.

The second chart sample above, generated for the logical CP engine with IPU address 5, however, shows that this logical engine is not heavily dispatched, and its four highest dispatch intervals (1 AM, 8 AM, 11 AM, and 4 PM) correspond to the highest busy times of the much busier logical engine shown in the top chart.

Uses

Refer to the query use information provided for the RMFCRL query to gain an understanding of the value provided by these cryptographic coprocessor utilization queries. This RMFWCE query provides additional granularity that can help you to understand the cryptographic function requirements of each logical processor assigned to your z/OS LPARs.

Data Extracts

Refer to the RMFCRL – CPU MF Cryptographic Coprocessor Activity LPAR Level query information for a complete description of the data extracts generated by the RMFCRT query. The data extracts and views are identical in all aspects. The only differences are in the granularity of the chart images.

Where, with the RMFCRL LPAR level charts, only one chart is generated for each LPAR, for this RMFCRE query, one chart is generated for each active logical IPU assigned to the LPAR.

Query Defaults

The following are RMFCRE Query Defaults:

- HARCME DETAIL time span file: Cycle 01
- Filters: Empty Filter task provided as placeholder for user HARCME file filtering to exclude/select specific LPARs
- Commented user code, that if uncommented will only accept yesterday's data
 - *IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

DATE

```
DATEPART(ENDTS) ;
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

DATEHOUR

```
PUT(DATEPART(ENDTS),YYMMDDD10.) || ':' || PUT(HOUR,Z2.) ;
```

DATEHHMM

```
PUT(DATEPART(ENDTS),YYMMDDP10.) || '-' || HHMM ;
```

AVGPCBSY

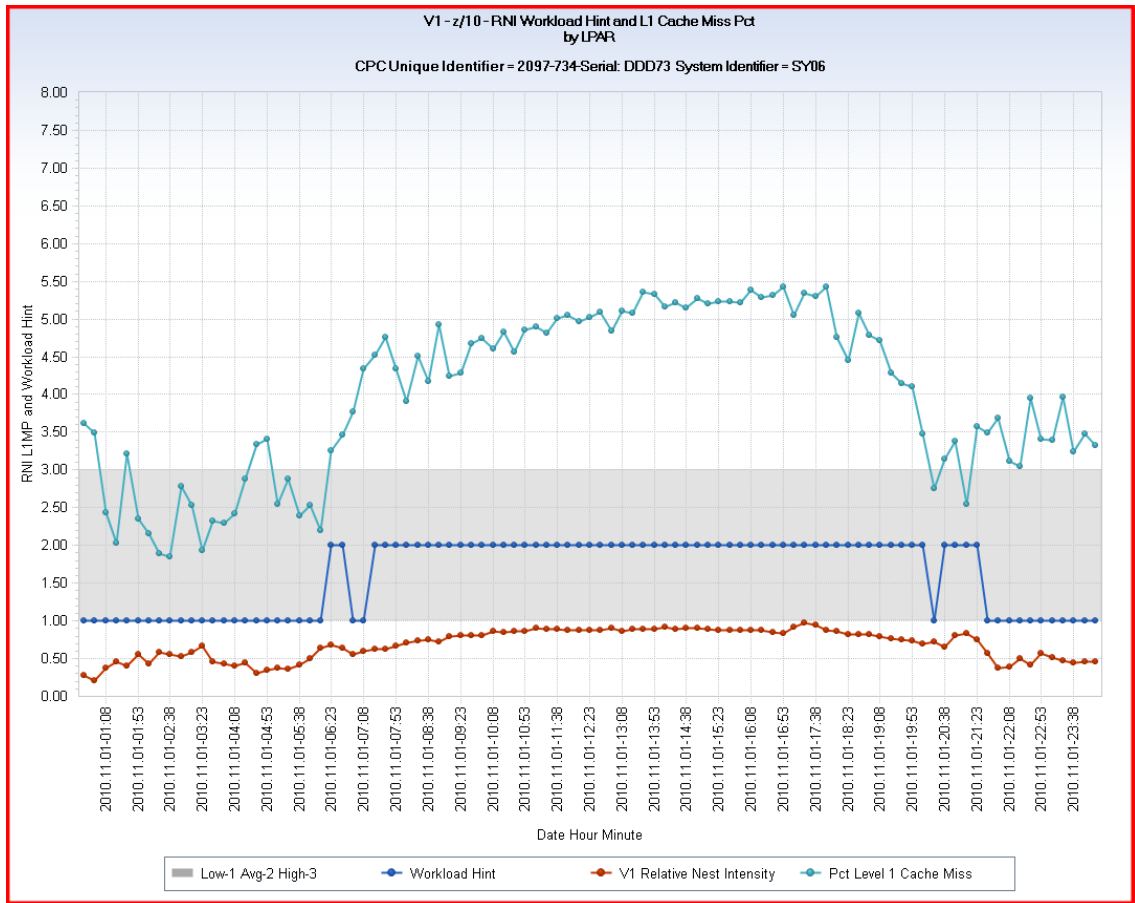
```
IF CMEAVIPU THEN AVGPCBSY=CMEPCBSY / CMEAVIPU ;
```

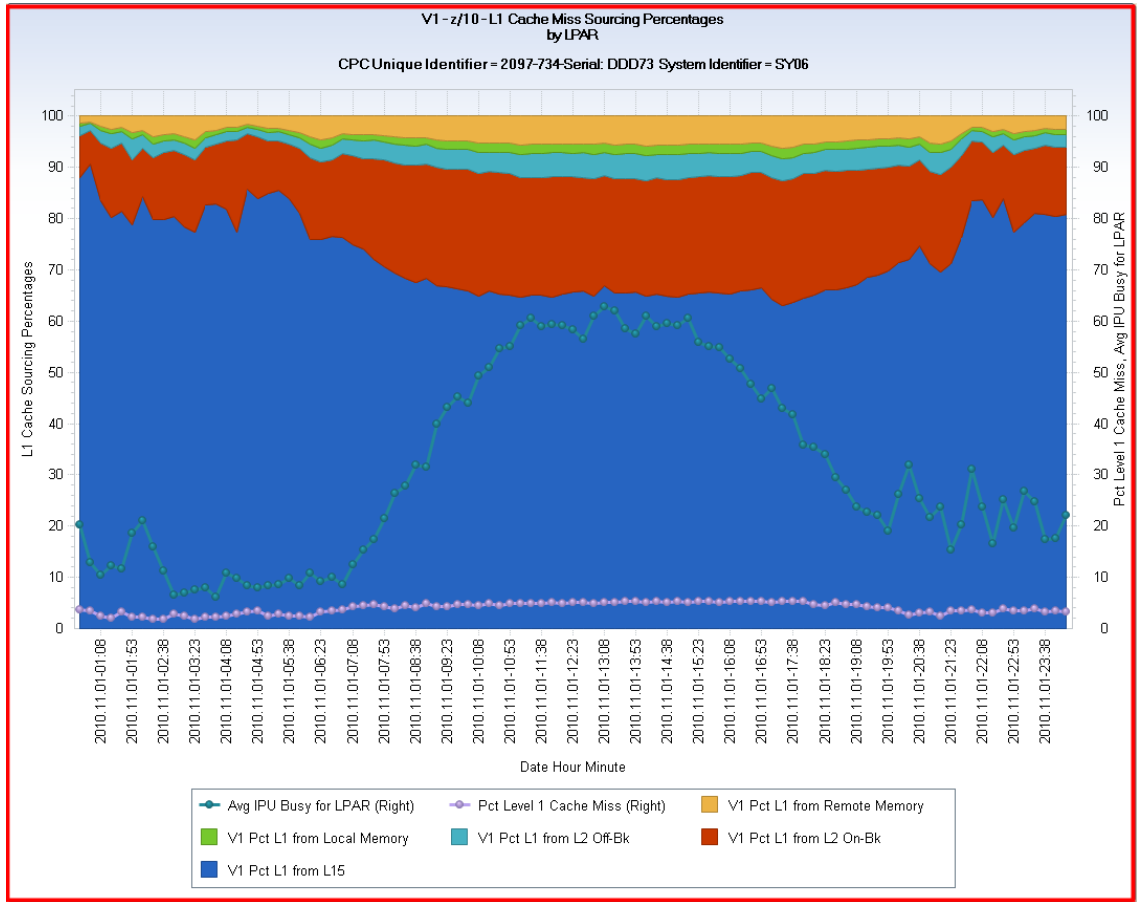
Modifications

The following modifications can be made to the RMFCRE query:

- You can input multiple cycles of DETAIL timespan HARCME data to create longer duration charts.
- You can input the DAYS, WEEKS, or higher timespan HARCME files, but will need to modify the Key Selection data elements in the Data Extract Steps. For example, if you input the DAYS timespan HARCME file, you will want to change DATEHHMM to DATEHOUR.
- If you want to restrict the charts to always show yesterday's data exclusively, uncomment the SAS statement in the USER CODE-Select Yesterday task in the HARCME File Selection step.
- You can use filter statements in the HARCME File Selection step to restrict data to particular time ranges (e.g., HOUR GE 8 AND HOUR LE 17) or particular time zones (e.g., ZONE EQ 1),
- You can add filter statements in the HARCME File Selection step to select a particular CPCID or LPAR (SYSID).
- You can use the extensive Chart View Edit capabilities to modify Y axis value ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

RMFWCL - CPU MF Workload Characterization LPAR Level





Overview

The RMFWCL query produces charts that show how efficiently LPAR workloads make use of the cache and memory hierarchy of IBM z10 and later generation Central Processing Complexes (CPC). Seven separate chart views are produced from each of the two CSV output files. One CSV output file is used to generate charts for LPARs executing on IBM z10 CPCs, and the other for LPARs executing on IBM z114 and z196 CPCs. The seven chart views in each CSV output file provide similar information for each CPC type, but the specific data element values charted are different because the cache architecture of different CPC models is different. IBM's CPU Measurement Facility collects information about LPAR logical IPU (Individual Processor Unit) interaction with the CPC cache architecture in three formats, Version 1 for the z10, Version 2 for the z114 and z196, and Version 3 for the zEC12 CPCs.

The three data extract CSV's generated by the RMFWCL query are:

- V1 z10 CPU MF LPAR RNI and Cache
- V2 z114 z196 CPU MF LPAR RNI and Cache
- V3 zEC12 CPU MF LPAR RNI and Cache

Each data extract generates the following seven chart views:

1. LPAR Level – RNI Workload Hint and L1 Cache Miss Pct
2. LPAR Level – L1 Cache Miss Sourcing Percentages
3. LPAR Level – L1 Data Cache Miss Sourcing Percentages
4. LPAR Level – L1 Instruction Cache Miss Sourcing Percentages
5. LPAR Level – Penalty Cycles and Avg Penalty per L1 Miss
6. LPAR Level – Realized MIPS and Avg IPU Count
7. LPAR Level – Instruction Count and CPI

Chart view #1 displays the "Workload Hint", Relative Nest Intensity (RNI), and Level 1 cache miss percentage. The Workload Hint is derived from an IBM formula that uses the RNI and Level 1 cache miss percentages to classify workloads as 'LOW', 'MEDIUM', and 'HIGH.' This workload classification methodology is required to accurately predict how workloads behave when upgrading from z10 or later CPC models. The first chart sample above was produced from a single cycle of the DETAIL timespan HARCML file. This chart, for LPAR 'SY06' running on an IBM 2097-734 (z10CPC with 34 CP engines), shows that from about 7 AM through 9 PM, the workload hint was 'MEDIUM', and from about 9 PM through 7 AM, the workload hint was 'LOW.' As the workload hint value moves from 'LOW' to 'HIGH', the workloads experience more Level 1 cache misses, and/or when retrieving the required item, find it in the slower cache levels and memory.

Chart view #2 uses a stacked area chart to show the percentages of where data or instructions were retrieved, when not found in the Level 1 Cache. In addition, the Level 1 cache miss percentage and average LPAR IPU busy are shown. The stacked area percentages are displayed with the least expensive (from a resource utilization perspective) retrieval source at the bottom to the most expensive (remote memory) at the top. In the example above, generated for the same LPAR as the first example, you can see that workloads running in the early morning (midnight to 7 AM), and late night (9 PM to midnight) find data and instructions in the z10 Level 1.5 cache more frequently than workloads running from 7 AM to 9 PM when more Level 1 cache misses occur.

Chart views #3 and #4 show the same information as the second view, but for Level 1 data cache misses and instruction cache misses, respectively. When examining these chart views, you will see that instruction cache misses tend to be resolved less expensively than data cache misses.

Chart view #5 displays the overall average number of penalty cycles incurred per Level 1 cache miss, as well as the average penalty cycles incurred for Level 1 misses in the data and instruction caches. Penalty cycles are wasted machine cycles that occur while the CPC is working to retrieve a datum or instruction from the cache/memory hierarchy.

Chart view #6 shows the average number of active logical IPUs (engines) for the LPAR, and the actual Millions of Instructions per Second (MIPS) rate realized by the logical engines while dispatched on the CPC.

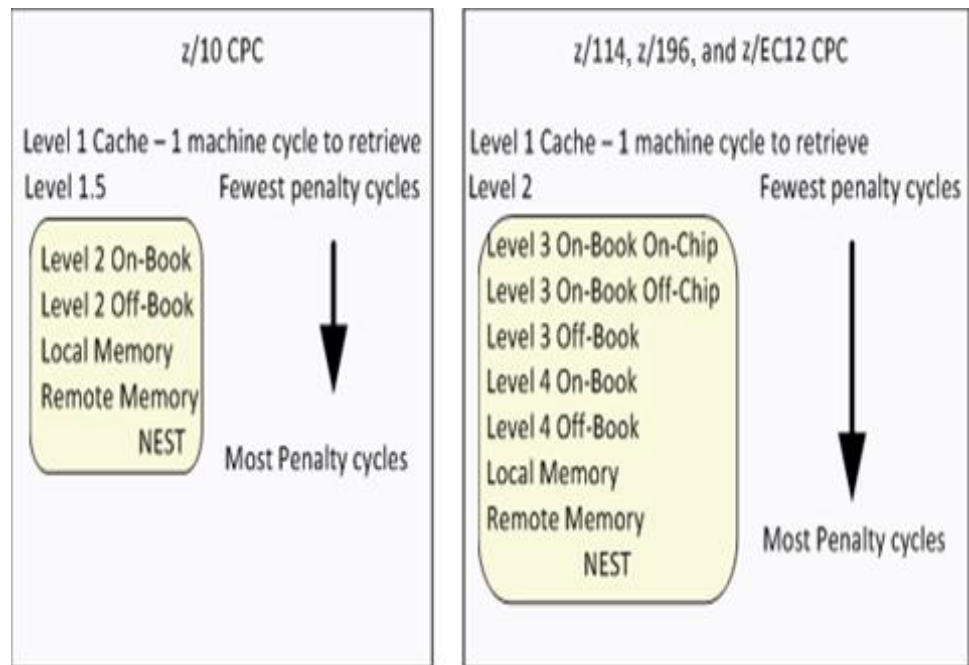
Chart view #7 shows the number of problem and supervisor state instructions executed, the average number of cycles per instruction for each of these states, as well as the overall CPI rate.

By default, the query is executed against the DETAIL timespan HARCML file, and each X axis value represents data from an RMF interval. The X axis metric, DATE HOUR MINUTE, is constructed by concatenating the DATE, HOUR, and MINUTE from the end time boundary of each interval.

Uses

A design objective of IBM mainframe CPC's is that when an instruction or datum is fetched from the IPU level 1 cache, the retrieval completes in a single machine cycle. With the introduction of the IBM z10 CPC, the CPC cycle rate increased dramatically from the 1.7 GHz rate of the previous z/9 CPC to 4.4 GHz. At this significantly higher cycle rate, the Level 1 data and instruction caches had to be reduced in size to continue to meet the one cycle retrieval objective.

Beginning with the z10, and continuing with the z196 (5.2 GHz), and zEC12 (5.5 GHz), to maximize CPC performance, IBM introduced additional layers of cache between the Level 1 cache and real memory. The number of machine cycles required to retrieve an item from these intermediate cache levels increases from level to level, with the worst performance occurring when the item must be retrieved from remote memory, where it can take several hundred machine cycles for retrieval. The machine cycles expended while attempting to find and retrieve a required datum or instruction are referred to as penalty cycles:



When preparing to add additional capacity to a z10 or later generation CPC, or to upgrade to a newer generation CPC, it is essential to understand the cache use characteristics of the workloads running on your z/OS LPARs in order to understand how much capacity is required to meet your performance objectives. LPARs whose workloads experience high rates of Level 1 cache misses for data and instructions will experience high numbers of penalty cycles. If the required items are frequently found in the deeper levels of cache or real memory, an area referred to as the “nest”, the number of penalty cycles increases. The IBM metric Relative Nest Intensity (RNI) provides a means to quantify this behavior. Higher RNI values mean greater access of the “nest” when retrieving required items not found in the Level 1 cache.

High numbers of penalty cycles incurred by LPAR workloads effectively degrade CPC capacity. When adding additional IPU's to a z10 or later generation CPC, you must be aware of the Level 1 cache miss percentages and RNI values to understand how much capacity is actually realized with the additional engines. Similarly, when upgrading from a z10 or later generation CPC, the cache and memory use characteristics of your LPAR workloads must be understood to accurately configure the new CPC.

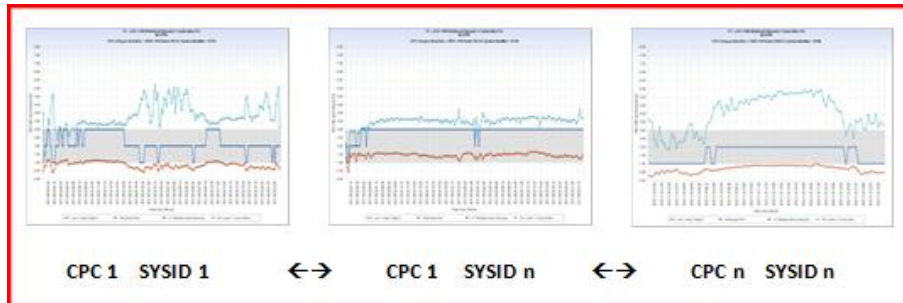
These charts can also be used to validate improved Level 1 cache hit percentages when activating an IBM option such as HIPERDISPATCH, where the objective is to direct specific LPAR logical engines to be repeatedly dispatched on the same physical engine, so that needed data and instructions are found in the low-level caches.

Finally, these charts can be used to identify LPARs whose workloads incur abnormally high numbers of penalty cycles. Application upgrades, designed to improve data and instruction Level 1 cache hit percentages and to reduce penalty cycles, can be evaluated using this query. Because the data processed by this query is LPAR level, in order to evaluate individual application behavior you may need to set up controlled environments where specific applications are executed in relative isolation at specific times.

Data Extracts

V1 z10 MF LPAR RNI and Cache

VIEW 1 of 7: V1 z10 LPAR – RNI Workload Hint and L1 Cache Miss Pct



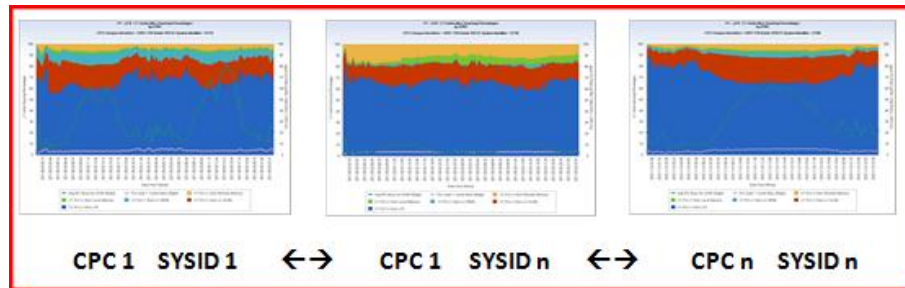
Left Y axis

- Workload Hint (Low-1 Avg-2 High-3)
- CML1RNI – V1 Relative Nest Intensity
- CMLPCL1M – Pct Level 1 Cache Miss
- Workload Hint Reference Area – shaded gray from 1-3

X axis

DATEHHMM – Date: Hour : Minute

VIEW 2 of 7: V1 z10 LPAR – L1 Cache Miss Sourcing Percentages



Left Y axis

- CML1PC15 – V1 PCT L1 from L1.5
- CML1PC2A – V1 PCT L1 from L2 On-Bk
- CML1PC2B – V1 PCT L1 from L2 Off-Bk
- CML1PC3A – V1 PCT L1 from Local Memory
- CML1PC3B – V1 PCT L1 from Remote Memory

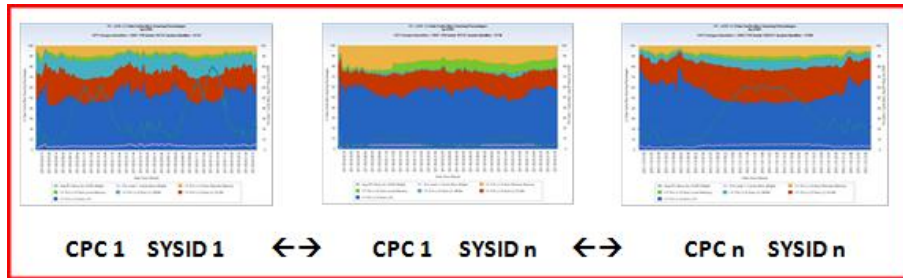
Right Y axis

- AVGPCBSY – Avg IPU Busy for LPAR
- CMLPCL1M – Pct Level 1 Cache Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 3 of 7: V1 z10 LPAR – L1 Data Cache Miss Sourcing Percentages



Left Y axis

- CML1PD15 – V1 PCT L1-D from L1.5
- CML1PD2A – V1 PCT L1-D from L2 On-Bk
- CML1PD2B – V1 PCT L1-D from L2 Off-Bk
- CML1PD3A – V1 PCT L1-D from Local Memory
- CML1PD3B – V1 PCT L1-D from Remote Memory

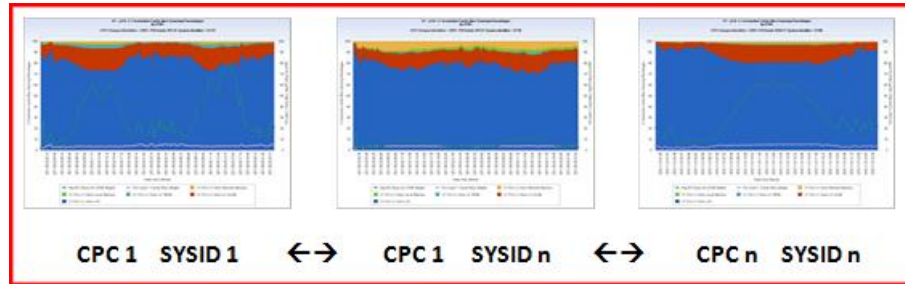
Right Y axis

- AVGPCBSY – Avg IPU Busy for LPAR
- CMLPCL1M – Pct Level 1 Cache Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 4 of 7: V1 z10 LPAR – L1 Instruction Cache Miss Sourcing Percentages



Left Y axis

- CML1PI15 – V1 PCT L1-I from L1.5
- CML1PI2A – V1 PCT L1-I from L2 On-Bk
- CML1PI2B – V1 PCT L1-I from L2 Off-Bk
- CML1PI3A – V1 PCT L1-I from Local Memory
- CML1PI3B – V1 PCT L1-I from Remote Memory

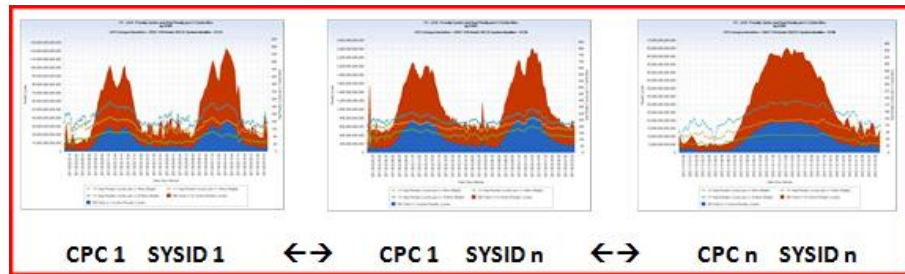
Right Y axis

- AVGPCBSY – Avg IPU Busy for LPAR
- CMLPCL1M – Pct Level 1 Cache Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 5 of 7: V1 z10 LPAR Penalty Cycles and Avg Penalty per L1 Miss



Left Y axis

- CMLB3 – Total L1 I-Cache Penalty Cycles
- CMLB5 – Total L1 D-Cache Penalty Cycles

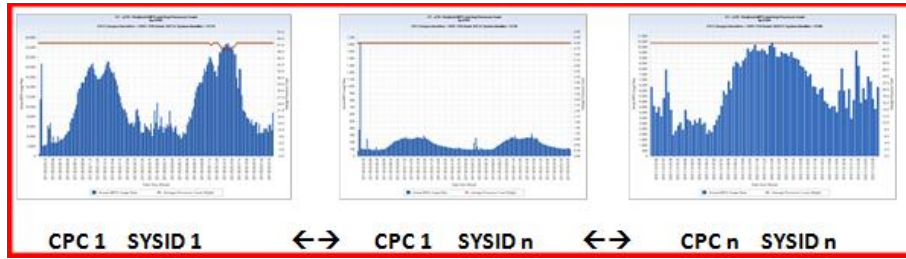
Right Y axis

- CML1AVPM – V1 Avg Penalty Cycles per L1 Miss
- CML1AVPD – V1 Avg Penalty Cycles per L1 D-Miss
- CML1AVPI – V1 Avg Penalty Cycles per L1 I-Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 6 of 7: V1 z10 LPAR – Realized MIPS and Avg IPU Count



Left Y axis

CMLMIPS – Actual MIPS Usage Rate

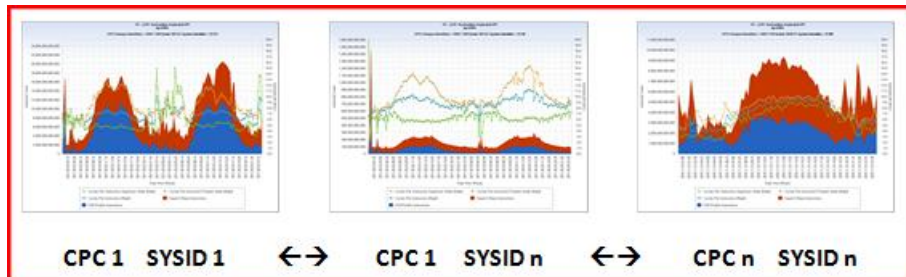
Right Y axis

CMLAVIPU – Average Processor Count

X axis

DATEHHMM – Date: Hour : Minute

VIEW 7 of 7: V1 z10 LPAR – Instruction Count and CPI



Left Y axis

CMLP33 – P33-ProbSt Instructions

CMLS33 – SuperV-State Instructions

Right Y axis

CMLCPI – Cycles per Instrucion

CPI_P – Cycles per Instruction Problem State

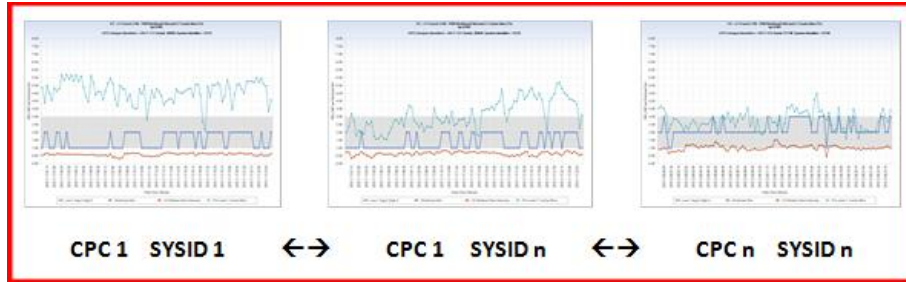
CPI_S – Cycles per Instruction Supervisor State

X axis

DATEHHMM – Date: Hour : Minute

V2 z114 z196 LPAR MF LPAR RNI and Cache

VIEW 1 of 7: V2 z114 z196 LPAR – RNI Workload Hint and L1 Cache Miss Pct



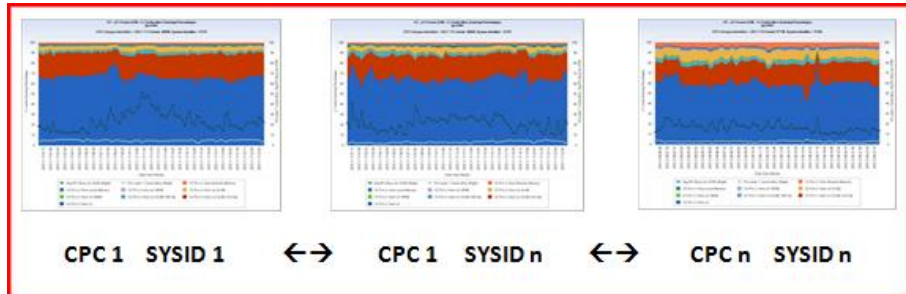
Left Y axis

- Workload Hint (Low-1 Avg-2 High-3)
- CML2RNI – V1 Relative Nest Intensity
- CMLPCL1M – Pct Level 1 Cache Miss
- Workload Hint Reference Area – shaded gray from 1-3

X axis

DATEHHMM – Date: Hour : Minute

VIEW 2 of 7: V2 z114 z196 LPAR – L1 Cache Miss Sourcing Percentages



Left Y axis

- CML2PC15 – V2 PCT L1 from L2
- CML2PC3A – V2 PCT L1 from L3 On-Bk On-Chp
- CML2PC3B – V2 PCT L1 from L3 On-Bk Off-Chp
- CML2PC4A – V2 PCT L1 from L4 On-Bk
- CML2PC4B – V2 PCT L1 from L4 Off-Bk
- CML2PC5A – V2 PCT L1 from Local Memory
- CML2PC5B – V2 PCT L1 from Remote Memory

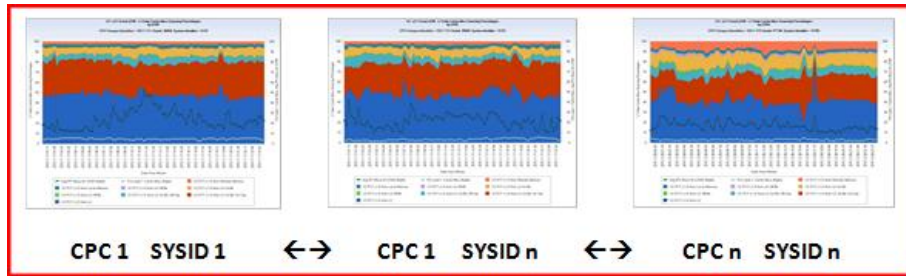
Right Y axis

- AVGPCBSY – Avg IPU Busy for LPAR
- CMLPCL1M – Pct Level 1 Cache Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 3 of 7: V2 z114 z196 LPAR – L1 Data Cache Miss Sourcing Percentage



Left Y axis

- CML2PD15 – V2 PCT L1-D from L2
- CML2PD3A – V2 PCT L1-D from L3 On-Bk On-Chp
- CML2PD3B – V2 PCT L1-D from L3 On-Bk Off-Chp
- CML2PD4A – V2 PCT L1-D from L4 On-Bk
- CML2PD4B – V2 PCT L1-D from L4 Off-Bk
- CML2PD5A – V2 PCT L1-D from Local Memory
- CML2PD5B – V2 PCT L1-D from Remote Memory

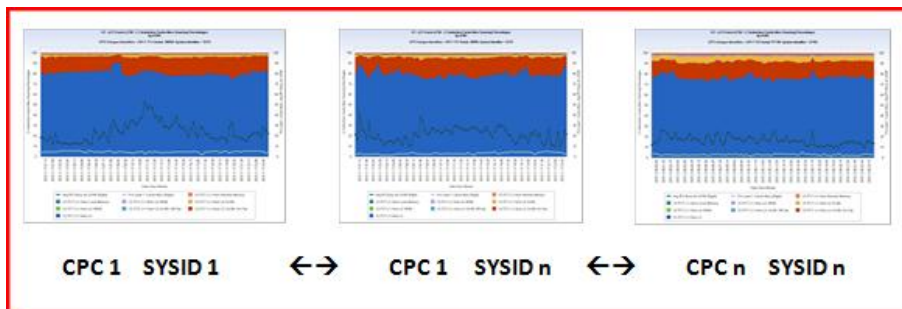
Right Y axis

- AVGPCBSY – Avg IPU Busy for LPAR
- CMLPCL1M – Pct Level 1 Cache Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 4 of 7: V2 z114 z196 LPAR – L1 Instruction Cache Miss Sourcing Percentages



Left Y axis

- CML2PI15 – V2 PCT L1-I from L2
- CML2PI3A – V2 PCT L1-I from L3 On-Bk On-Chp
- CML2PI3B – V2 PCT L1-I from L3 On-Bk Off-Chp
- CML2PI4A – V2 PCT L1-I from L4 On-Bk
- CML2PI4B – V2 PCT L1-I from L4 Off-Bk
- CML2PI5A – V2 PCT L1-I from Local Memory
- CML2PI5B – V2 PCT L1-I from Remote Memory

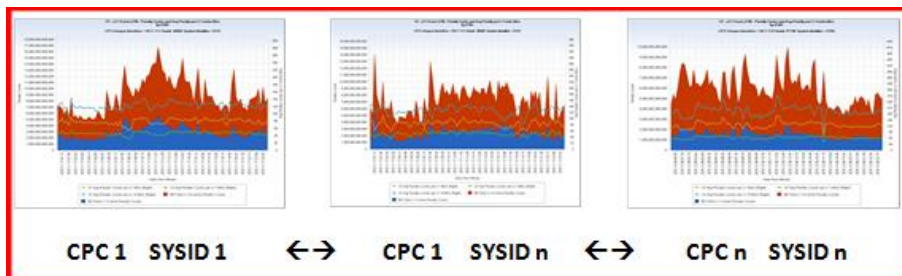
Right Y axis

- AVGPCBSY – Avg IPU Busy for LPAR
- CMLPCL1M – Pct Level 1 Cache Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 5 of 7: V2 z114 z196 LPAR Penalty Cycles and Avg Penalty per L1 Miss



Left Y axis

- CMLB3 – Total L1 I-Cache Penalty Cycles
- CMLB5 – Total L1 D-Cache Penalty Cycles

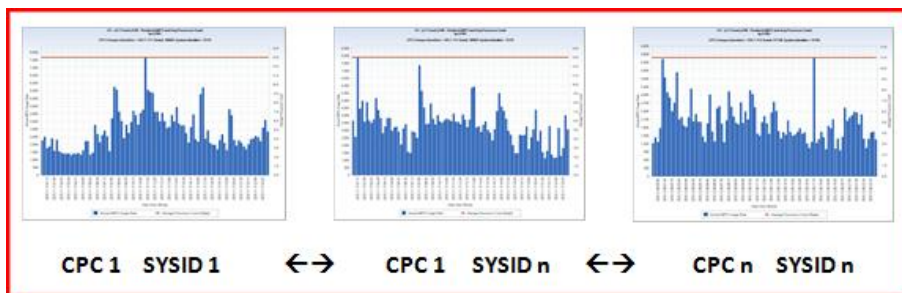
Right Y axis

- CML2AVPM – V2 Avg Penalty Cycles per L1 Miss
- CML2AVPD – V2 Avg Penalty Cycles per L1 D-Miss
- CML2AVPI – V2 Avg Penalty Cycles per L1 I-Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 6 of 7: V2 z114 z196 LPAR – Realized MIPS and Avg IPU Count



Left Y axis

CMLMIPS – Actual MIPS Usage Rate

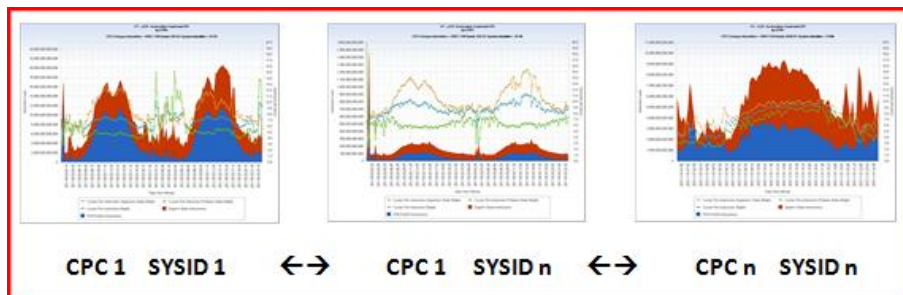
Right Y axis

CMLAVIPU – Average Processor Count

X axis

DATEHHMM – Date: Hour : Minute

VIEW 7 of 7: V2 z114 z196 LPAR – Instruction Count and CPI



Left Y axis

- CMLP33 – P33-ProbSt Instructions
- CMLS33 – SuperV-State Instructions

Right Y axis

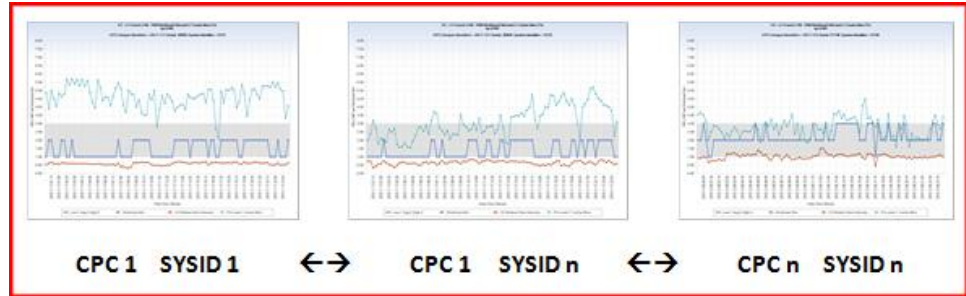
- CMLCPI – Cycles per Instrucion
- CPI_P – Cycles per Instruction Problem State
- CPI_S – Cycles per Instruction Supervisor State

X axis

DATEHHMM – Date: Hour : Minute

V3 zEC12 MF LPAR RNI and Cache

VIEW 1 of 7: V3 zEC12 LPAR – RNI Workload Hint and L1 Cache Miss Pct



Left Y axis

Vx Workload Hint

CMLxRNI – Vx Relative Nest Intensity

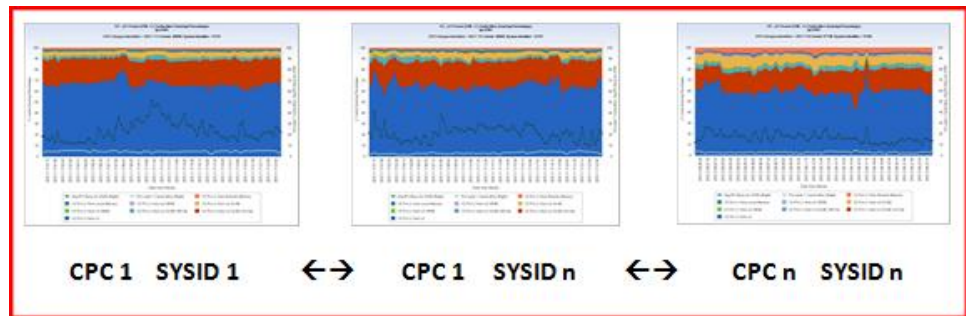
CMLPCL1M – Pct Level 1 Cache Miss

Workload Hint Reference Area – (Low-1 Avg-2 High-3) shaded gray from 1-3 where “x” is “2” for V2 (z114, z196) charts, “3” for V3 (zEC12) charts

X axis

DATEHHMM – Date: Hour : Minute

VIEW 2 of 7: V3 zEC12 LPAR – L1 Cache Miss Sourcing Percentages



Left Y axis

CMLxPC15 – Vx PCT L1 from L2

CMLxPC3A – Vx PCT L1 from L3 On-Bk On-Chp

CMLxPC3B – Vx PCT L1 from L3 On-Bk Off-Chp

CMLxPC4A – Vx PCT L1 from L4 On-Bk

CMLxPC4B – Vx PCT L1 from L4 Off-Bk

CMLxPC5A – Vx PCT L1 from Local Memory

CMLxPC5B – Vx PCT L1 from Remote Memory

where “x” is “2” for V2 (z114, z196) charts, “3” for V3 (zEC12) charts

Right Y axis

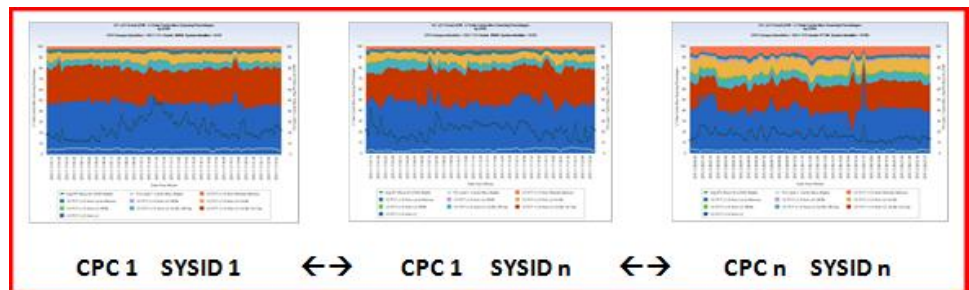
AVGPCBSY – Avg IPU Busy for LPAR

CMLPCL1M – Pct Level 1 Cache Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 3 of 7: V3 zEC12 LPAR – L1 Data Cache Miss Sourcing Percentage



Left Y axis

CMLxPD15 – Vx PCT L1-D from L2

CMLxPD3A – Vx PCT L1-D from L3 On-Bk On-Chp

CMLxPD3B – Vx PCT L1-D from L3 On-Bk Off-Chp

CMLxPD4A – Vx PCT L1-D from L4 On-Bk

CMLxPD4B – Vx PCT L1-D from L4 Off-Bk

CMLxPD5A – Vx PCT L1-D from Local Memory

CMLxPD5B – Vx PCT L1-D from Remote Memory

where “x” is “2” for V2 (z114, z196) charts, “3” for V3 (zEC12) charts

Right Y axis

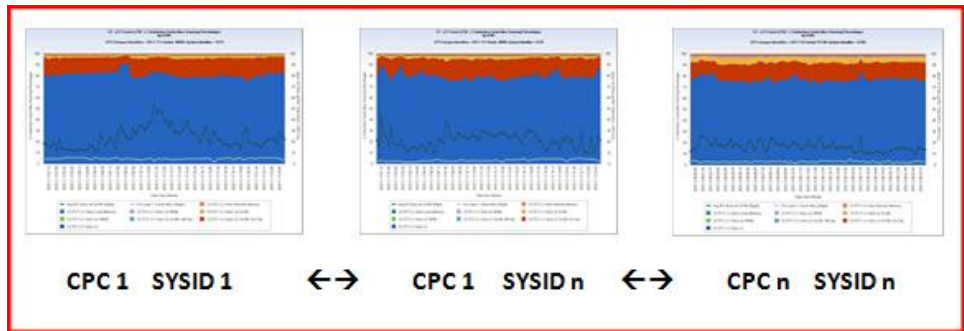
AVGPCBSY – Avg IPU Busy for LPAR

CMLPCL1M – Pct Level 1 Cache Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 4 of 7: V3 zEC12 LPAR – L1 Instruction Cache Miss Sourcing Percentages



Left Y axis

CMLxPI15 – Vx PCT L1-I from L2

CMLxPI3A – Vx PCT L1-I from L3 On-Bk On-Chp

CMLxPI3B – Vx PCT L1-I from L3 On-Bk Off-Chp

CMLxPI4A – Vx PCT L1-I from L4 On-Bk

CMLxPI4B – Vx PCT L1-I from L4 Off-Bk

CMLxPI5A – Vx PCT L1-I from Local Memory

CMLxPI5B – Vx PCT L1-I from Remote Memory

where “x” is “2” for V2 (z114, z196) charts, “3” for V3 (zEC12) charts

Right Y axis

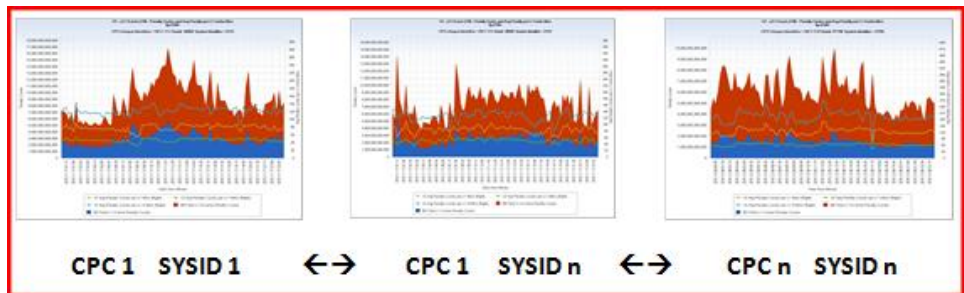
AVGPCBSY – Avg IPU Busy for LPAR

CMLPCL1M – Pct Level 1 Cache Miss

X axis

DATEHHMM – Date: Hour : Minute

VIEW 5 of 7: V3 zEC12 LPAR Penalty Cycles and Avg Penalty per L1 Miss



Left Y axis

CMLB3 – Total L1 I-Cache Penalty Cycles

CMLB5 – Total L1 D-Cache Penalty Cycles

Right Y axis

CMLxAVPM – Vx Avg Penalty Cycles per L1 Miss

CMLxAVPD – Vx Avg Penalty Cycles per L1 D-Miss

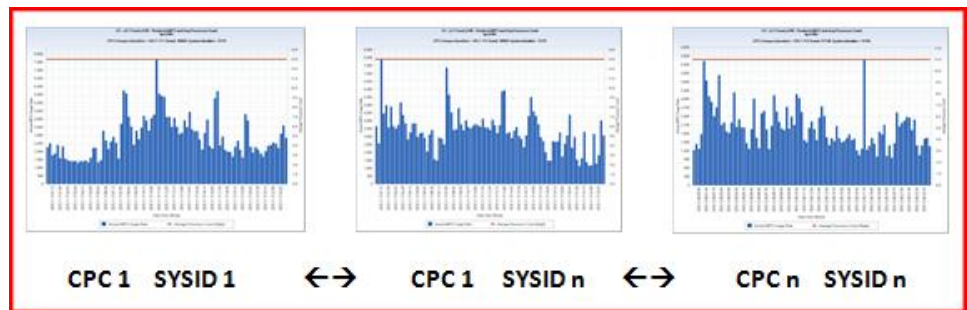
CMLxAVPI – Vx Avg Penalty Cycles per L1 I-Miss

where “x” is “2” for V2 (z114, z196) charts, “3” for V3 (zEC12) charts

X axis

DATEHHMM – Date: Hour : Minute

VIEW 6 of 7: V3 zEC12 LPAR – Realized MIPS and Avg IPU Count



Left Y axis

CMLMIPS – Actual MIPS Usage Rate

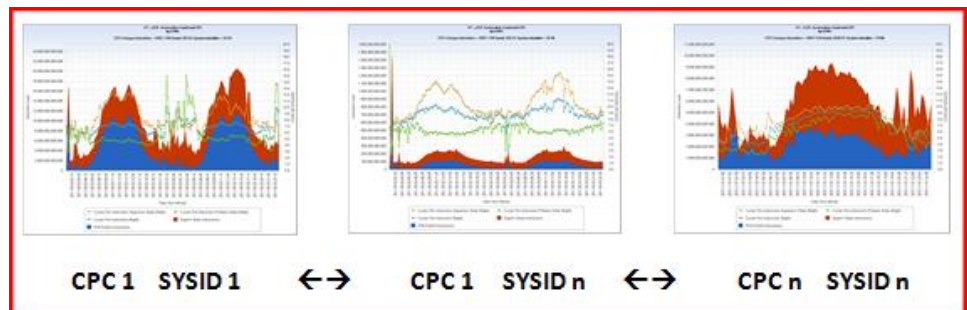
Right Y axis

CMLAVIPU – Average Processor Count

X axis

DATEHHMM – Date: Hour : Minute

VIEW 7 of 7: V3 zEC12 LPAR – Instruction Count and CPI



Left Y axis

CMLP33 – P33-ProbSt Instructions

CMLS33 – SuperV-State Instructions

Right Y axis

CMLCPI – Cycles per Instrucion

CPI_P – Cycles per Instruction Problem State

CPI_S – Cycles per Instruction Supervisor State

X axis

DATEHHMM – Date: Hour : Minute

Query Defaults

The following are RMFWCL Query Defaults:

- HARCML DETAIL time span file: Cycle 01
- Filters: Empty Filter task provided as placeholder for user HARCML file filtering to exclude/select specific LPARs
- Commented user code, that if uncommented will only accept yesterday's data
 - *IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

DATE

```
DATEPART(ENDTS) ;
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

DATEHOUR

```
PUT(DATEPART(ENDTS),YYMMDDD10.) || ':' || PUT(HOUR,Z2.) ;
```

DATEHHMM

```
PUT(DATEPART(ENDTS),YYMMDDP10.) || '-' || HHMM ;
```

WKLDHINT

```
IF HISCVER2=1 THEN RNI=CML1RNI ;
ELSE IF HISCVER2=2 THEN RNI=CML2RNI ;
IF CMLPCL1M LT 3 THEN DO ;
  IF RNI GE .75 THEN WKLDHINT=2 ;
  ELSE WKLDHINT=1 ;
END ;
ELSE IF 3 LE CMLPCL1M LE 6 THEN DO ;
  IF RNI GT 1.0 THEN WKLDHINT=3 ;
  ELSE IF 0.6 LE RNI LE 1.0 THEN WKLDHINT=2 ;
  ELSE WKLDHINT=1 ;
END ;
ELSE IF CMLPCL1M GT 6 THEN DO ;
  IF RNI GE 0.75 THEN WKLDHINT=3 ;
  ELSE WKLDHINT=2 ;
END ;
```

CPI_P

```
IF CMLP33 THEN CPI_P = CMLP32 / CMLP33;
```

CPI_S

```
IF CMLS33 THEN CPI_S = CMLS32 / CMLS33 ;
```

AVGPCBSY

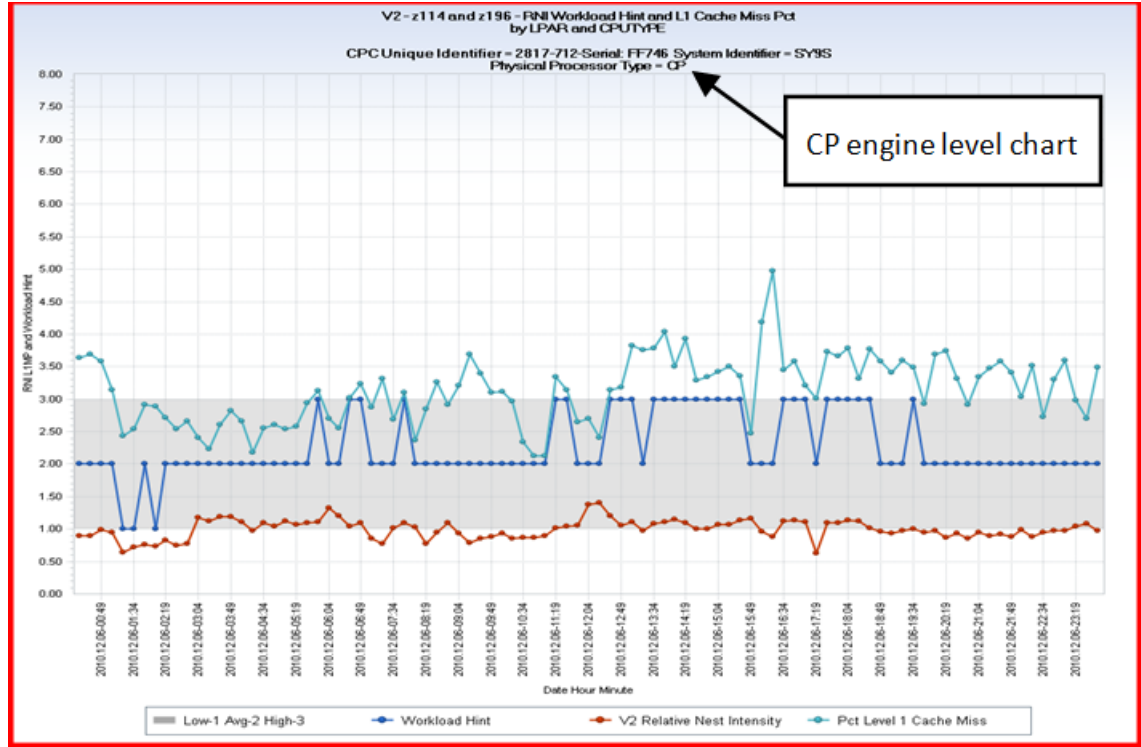
```
IF CMLAVIPU THEN AVGPCBSY=CMLPCBSY / CMLAVIPU ;
```

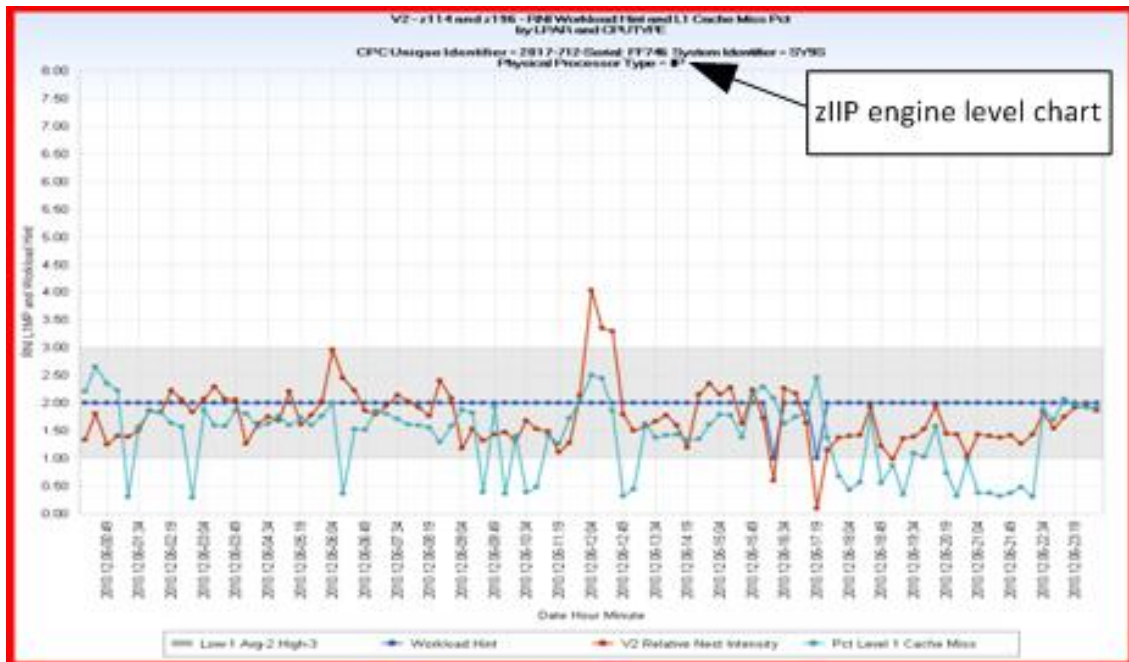
Modifications

The following modifications can be made to the RMFWCL query:

- You can input multiple cycles of DETAIL timespan HARCML data to create longer duration charts.
- You can input the DAYS, WEEKS, or higher timespan HARCML files, but will need to modify the Key Selection data elements in the Data Extract Steps. For example, if you input the DAYS timespan HARCML file, you will want to change DATEHHMM to DATEHOUR.
- If you want to restrict the charts to always show yesterday's data exclusively, uncomment the SAS statement in the USER CODE-Select Yesterday task in the HARCML File Selection step.
- You can use filter statements in the HARCML File Selection step to restrict data to particular time ranges (e.g., HOUR GE 8 AND HOUR LE 17) or particular time zones (e.g., ZONE EQ 1),
- You can add filter statements in the HARCML File Selection step to select a particular CPCID or LPAR (SYSID).
- If you only have IBM z114 and z196 CPCs, you can delete the Version 1 Data Extract step that generates the output CSV for IBM z10 CPCs.
- You can use the extensive Chart View Edit capabilities to modify X axis ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

RMFWCT - CPU MF Workload Characterization CPUTYPE Level





Overview

The RMFWCT query produces charts that reveal how efficiently workloads executing on the different types of IPU (Individual Processor Units) assigned to z/OS LPARs make use of the cache and memory hierarchy of IBM z10 and later generation Central Processing Complexes (CPCs). z/OS LPARs can be assigned three different IPU types:

CP

Central Processors – General processing unit

zAAP

System z Application Assist Processor – specialty engine for JAVA workloads

zIIP

System z Integrated Information Processor – specialty engine for enclave exploiting workloads

Note: zIIP specialty engines accept and process both zIIP and zAAP eligible workloads on a CPC if the CPC has no zAAP engines installed and the IEASYSxx PARMLIB member has ZAAPZIIP=YES specified.

The three data extract CSV's generated by the RMFWCT query are:

- V1 z10 CPU MF CPUTYPE RNI and Cache
- V2 z114 z196 CPU MF CPUTYPE RNI and Cache
- V3 zEC12 CPU MF CPUTYPE RNI and Cache

Each data extract generates the following seven chart views:

1. CPUTYPE Level – RNI Workload Hint and L1 Cache Miss Pct
2. CPUTYPE Level – L1 Cache Miss Sourcing Percentages
3. CPUTYPE Level – L1 Data Cache Miss Sourcing Percentages
4. CPUTYPE Level – L1 Instruction Cache Miss Sourcing Percentages
5. CPUTYPE Level – Penalty Cycles and Avg Penalty per L1 Miss
6. CPUTYPE Level – Realized MIPS and Avg IPU Count
7. CPUTYPE Level – Instruction Count and CPI

Refer to the RMFWCL – CPU MF Workload Characterization LPAR Level query description for information about each of these chart views. The only difference between the charts created by the RMFWCL LPAR level query and this RMFWCT CPUTYPE level query is one of granularity. Where the RMFWCL query views generate a single chart for each z/OS LPAR, the RMFWCT CPUTYPE level query views generate a separate chart for each IPU type (e.g., CP, zAAP, or zIIP) assigned to each LPAR.

The two chart samples above show charts generated for view 1 – RNI Workload Hint and L1 Cache Miss Pct, for a single LPAR, 'SY9S.' LPAR 'SY9S' is a z/OS LPAR running on an IBM 2817-712 (z196 CPC with 12 CP engines). LPAR 'SY9S' was defined with eight logical CP engines and three logical zIIP engines. The workload hint, Level 1 cache miss percent, and Relative Nest Intensity (RNI) for the eight logical CP engines are displayed in the first chart above. The second chart above shows the same metrics for the three zIIP engines. To see the number of logical engines of each type assigned to the LPAR, you can examine the 6th chart view – Realized MIPS and Avg IPU Count.

It is apparent from viewing these two charts that the Level 1 cache miss percentage (cyan colored line), and the RNI (red line) are quite different for workloads when executing on the CP engines versus the zIIP engines.

Uses

Refer to the query use information provided for the RMFWCL query to gain an understanding of the value provided by these workload characterization queries. This RMFWCT query provides additional insight that allows you to further refine decisions regarding capacity requirements in an existing z10 or z196 CPC, or when planning to upgrade to a z196 or later generation CPC.

For example, if the workloads driving the need for additional capacity primarily execute on zIIP engines, then you should closely examine the cache utilization characteristics at the zIIP engine level to help determine how much capacity is needed to meet your requirements.

You may discover that for a particular LPAR, workloads running on zIIP or zAAP processors have better Level 1 cache hit ratios. When a miss occurs, they incur fewer penalty cycles in retrieving the required item than workloads running on CP processors. This knowledge will help you to determine the appropriate number and types of IPU engines necessary to meet your capacity requirements.

Finally, when analyzing application performance, where applications tend to exploit particular IPU types, such as the CP or specialty engines, these CPUTYPE level charts provide much better cache usage analysis than the LPAR level charts where all IPU performance information is summarized to the LPAR level.

Data Extracts

Refer to the RMFWCL – CPU MF Workload Characterization LPAR Level query information for a complete description of the data extracts generated by the RMFWCT query. The data extracts and views are identical in all aspects. The only differences are in the granularity of the chart images.

Where, with the RMFWCL LPAR level charts, only one chart is generated for each LPAR, for this RMFWCT query, one chart is generated for each CPUTYPE (e.g., CP, zAAP, zIIP) assigned to each LPAR.

Query Defaults

The following are RMFCWT Query Defaults:

- HARCMT DETAIL time span file: Cycle 01
- Filters: Empty Filter task provided as placeholder for user HARCMT file filtering to exclude/select specific LPARs
- Commented user code, that if uncommented will only accept yesterday's data
 - *IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

DATE

```
DATEPART(ENDTS) ;
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

DATEHOUR

```
PUT(DATEPART(ENDTS),YYMMDDD10.) || ':' || PUT(HOUR,Z2.) ;
```

DATEHHMM

```
PUT(DATEPART(ENDTS),YYMMDDP10.) || '-' || HHMM ;
```

WKLDHINT

```
IF HISCV2=1 THEN RNI=CMT1RNI ;
ELSE IF HISCV2=2 THEN RNI=CMT2RNI ;
IF CMTPL1M LT 3 THEN DO ;
  IF RNI GE .75 THEN WKLDHINT=2 ;
  ELSE WKLDHINT=1 ;
END ;
ELSE IF 3 LE CMTPL1M LE 6 THEN DO ;
  IF RNI GT 1.0 THEN WKLDHINT=3 ;
  ELSE IF 0.6 LE RNI LE 1.0 THEN WKLDHINT=2 ;
  ELSE WKLDHINT=1 ;
END ;
ELSE IF CMTPL1M GT 6 THEN DO ;
  IF RNI GE 0.75 THEN WKLDHINT=3 ;
  ELSE WKLDHINT=2 ;
END ;
```

CPI_P

```
IF CMT33 THEN CPI_P = CMT32 / CMT33;
```

CPI_S

```
IF CMTS33 THEN CPI_S = CMTS32 / CMTS33 ;
```

AVGPCBSY

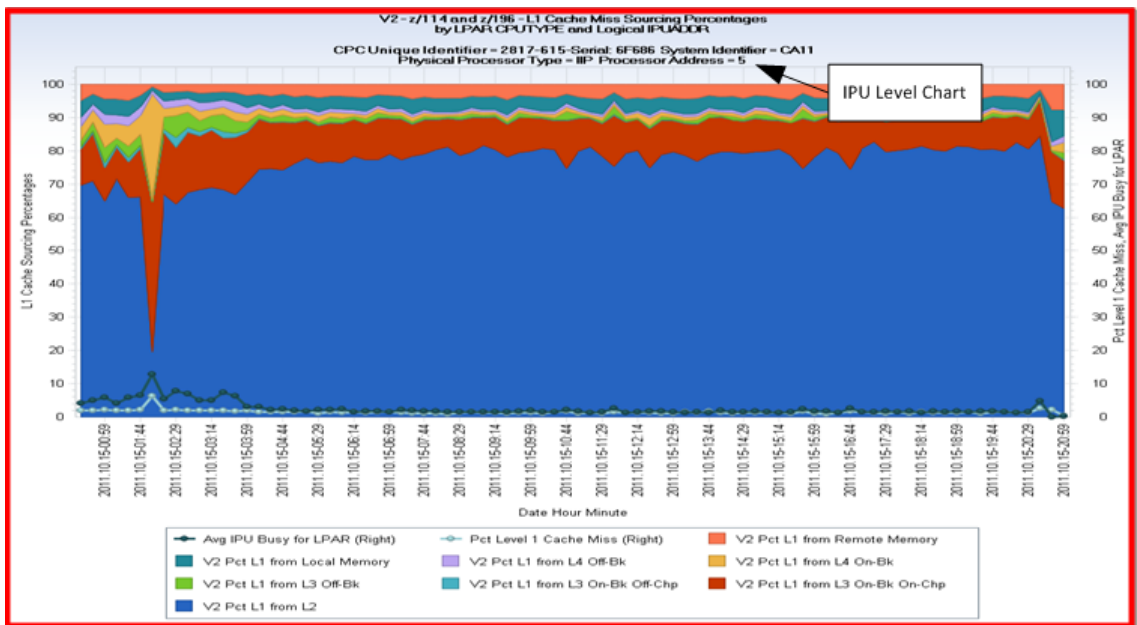
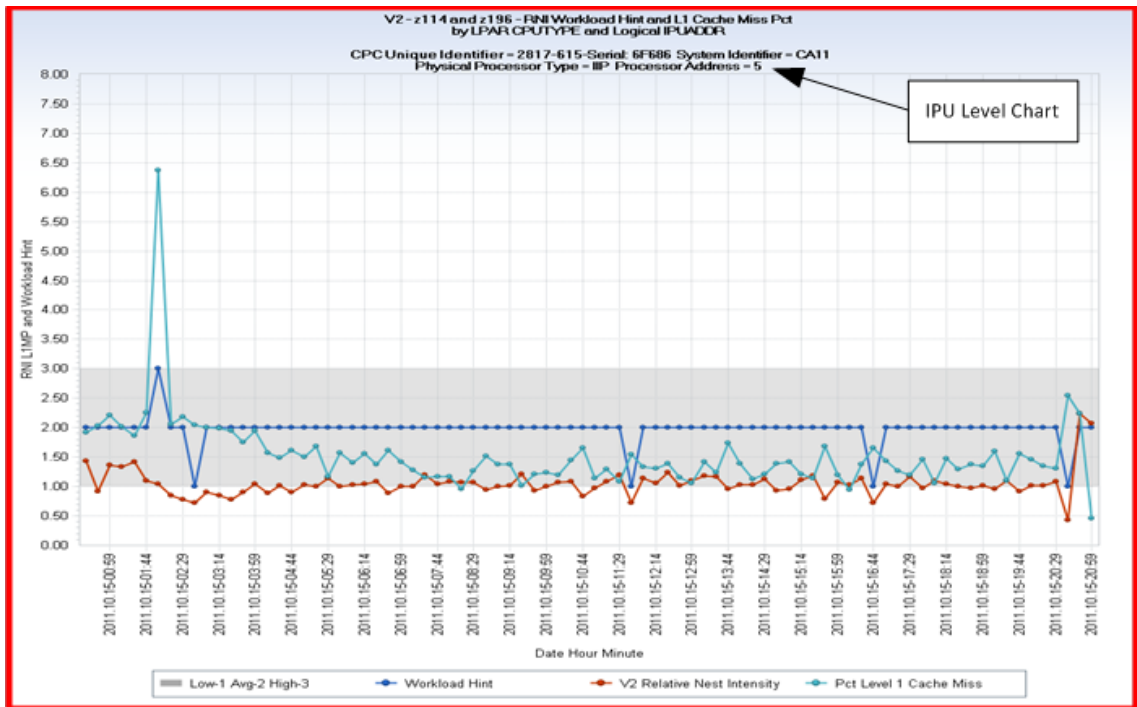
IF CMTAVIPU THEN AVGPCBSY=CMTPCBSY / CMTAVIPU ;

Modifications

The following modifications can be made to the RMFWCT query:

- You can input multiple cycles of DETAIL timespan HARCMT data to create longer duration charts.
- You can input the DAYS, WEEKS, or higher timespan HARCMT files, but will need to modify the Key Selection data elements in the Data Extract Steps. For example, if you input the DAYS timespan HARCMT file, you will want to change DATEHHMM to DATEHOUR.
- If you want to restrict the charts to always show yesterday's data exclusively, uncomment the SAS statement in the USER CODE-Select Yesterday task in the HARCMT File Selection step.
- You can use filter statements in the HARCMT File Selection step to restrict data to particular time ranges (e.g., HOUR GE 8 AND HOUR LE 17) or particular time zones (e.g., ZONE EQ 1),
- You can add filter statements in the HARCMT File Selection step to select a particular CPCID or LPAR (SYSID).
- If you only have IBM z114 and z196 CPCs, you can delete the Version 1 Data Extract step that generates the output CSV for IBM z10 CPCs.
- You can use the extensive Chart View Edit capabilities to modify X axis ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

RMFWCE - CPU MF Workload Characterization IPU Level



Overview

The RMFWCE query produces charts that reveal how efficiently workloads executed on each active logical IPU (Individual Processor Unit), assigned to z/OS LPARs, make use of the cache and memory hierarchy of IBM z10 and later generation Central Processing Complexes (CPCs).

The three data extract CSV's generated by the RMFWCE query are:

- V1 z10 CPU MF IPU RNI and Cache
- V2 z114 z196 CPU MF IPU RNI and Cache
- V3 zEC12 CPU MF IPU RNI and Cache

Each data extract generates the following seven chart views:

1. IPU Level – RNI Workload Hint and L1 Cache Miss Pct
2. IPU Level – L1 Cache Miss Sourcing Percentages
3. IPU Level – L1 Data Cache Miss Sourcing Percentages
4. IPU Level – L1 Instruction Cache Miss Sourcing Percentages
5. IPU Level – Penalty Cycles and Avg Penalty per L1 Miss
6. IPU Level – Realized MIPS and Avg IPU Count
7. IPU Level – Instruction Count and CPI

Refer to the RMFWCL – CPU MF Workload Characterization LPAR Level query description for information about each of these chart views. The only difference between the charts created by the RMFWCL LPAR level query and this RMFWCE IPU level query is one of granularity. Where the RMFWCL query views generate a single chart for each LPAR, the RMFWCE IPU level query views generate a separate chart for each active logical IPU (engine) assigned to each LPAR.

The first chart sample above shows the chart generated for view 1 – RNI Workload Hint and L1 Cache Miss Pct, for a single logical processor assigned to LPAR, 'CA11.' LPAR 'CA11' is a z/OS LPAR running on an IBM 2817-615 (z196 CPC with 15 CP engines). LPAR 'CA11' is defined with four logical CP engines and two logical zIIP engines. The workload hint, Level 1 cache miss percent, and Relative Nest Intensity (RNI) for the zIIP engine with logical IPU address 5 is shown with this chart. There is a large spike in the Level 1 cache miss percentage, and workload hint at about 2 AM.

The second chart sample above shows View 2 – L1 Cache Miss Sourcing Percentages for the same logical engine. You can see that around 2 AM, the IPU busy percentage spiked, and a much greater percentage of Level 1 cache misses were satisfied from the Level 3 On-Book On-Chip cache than normal.

Uses

Refer to the query use information provided for the RMFWCL query to gain an understanding of the value provided by these workload characterization queries. This RMFWCE query provides individual logical engine level insights that can help to decide if the numbers of logical engines, of each CPU type (CP, zIIP, and zAAP) assigned to each LPAR are appropriate.

In some cases, you may find that certain logical engines are rarely dispatched, and because of this, initially incur large numbers of penalty cycles because the data and instructions required for workloads is not in the fast, low level caches.

Data Extracts

Refer to the RMFWCL – CPU MF Workload Characterization LPAR Level query information for a complete description of the data extracts generated by the RMFWCE query. The data extracts and views are identical in all aspects. The only differences are in the granularity of the chart images.

Where, with the RMFWCL LPAR level charts, only one chart is generated for each LPAR, for this RMFWCE query, one chart is generated for each active logical IPU assigned to the LPAR.

Query Defaults

The following are RMFWCE Query Defaults:

- HARCME DETAIL time span file: Cycle 01
- Filters: Empty Filter task provided as placeholder for user HARCME file filtering to exclude/select specific LPARs
- Commented user code, that if uncommented will only accept yesterday's data
 - *IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

DATE

```
DATEPART(ENDTS) ;
```

HHMM

```
PUT(HOUR(ENDTS),Z2.) || ':' || PUT(MINUTE(ENDTS),Z2.);
```

DATEHOUR

```
PUT(DATEPART(ENDTS),YYMMDDD10.) || ':' || PUT(HOUR,Z2.) ;
```

DATEHHMM

```
PUT(DATEPART(ENDTS),YYMMDDP10.) || '-' || HHMM ;
```

WKLDHINT

```
IF HISCVER2=1 THEN RNI=CME1RNI ;  
ELSE IF HISCVER2=2 THEN RNI=CME2RNI ;  
IF CMEPCL1M LT 3 THEN DO ;  
  IF RNI GE .75 THEN WKLDHINT=2 ;  
  ELSE WKLDHINT=1 ;  
END ;  
ELSE IF 3 LE CMEPCL1M LE 6 THEN DO ;  
  IF RNI GT 1.0 THEN WKLDHINT=3 ;  
  ELSE IF 0.6 LE RNI LE 1.0 THEN WKLDHINT=2 ;  
  ELSE WKLDHINT=1 ;  
END ;  
ELSE IF CMEPCL1M GT 6 THEN DO ;  
  IF RNI GE 0.75 THEN WKLDHINT=3 ;  
  ELSE WKLDHINT=2 ;  
END ;
```

CPI_P

```
IF CMEP33 THEN CPI_P = CMEP32 / CMEP33;
```

CPI_S

```
IF CMES33 THEN CPI_S = CMES32 / CMES33 ;
```

AVGPCBSY

```
IF CMEAVIPU THEN AVGPCBSY=CMEPCBSY / CMEAVIPU ;
```

Modifications

The following modifications can be made to the RMFWCE query:

- You can input multiple cycles of DETAIL timespan HARCME data to create longer duration charts.
- You can input the DAYS, WEEKS, or higher timespan HARCME files, but will need to modify the Key Selection data elements in the Data Extract Steps. For example, if you input the DAYS timespan HARCME file, you will want to change DATEHHMM to DATEHOUR.
- If you want to restrict the charts to always show yesterday's data exclusively, uncomment the SAS statement in the USER CODE-Select Yesterday task in the HARCME File Selection step.
- You can use filter statements in the HARCME File Selection step to restrict data to particular time ranges (e.g., HOUR GE 8 AND HOUR LE 17) or particular time zones (e.g., ZONE EQ 1),
- You can add filter statements in the HARCME File Selection step to select a particular CPCID or LPAR (SYSID).
- If you only have IBM z114 and z196 CPCs, you can delete the Version 1 Data Extract step that generates the output CSV for IBM z10 CPCs.
- You can use the extensive Chart View Edit capabilities to modify X axis ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.

Chapter 7: CA MICS Batch and Operations Analyzer (SMF) Queries

This section contains the following topics:

[Top 10 Resource Users](#) (see page 285)

[Specialty Engine zIIP/zAAP Focus](#) (see page 333)

Top 10 Resource Users

This section contains the following topics:

[SMFAJT - SMF Daily Top 10 APPC Transaction Jobs](#) (see page 286)

[SMFBJT - SMF Daily Top 10 Batch Jobs](#) (see page 291)

[SMFBPT - SMF Daily Top 10 Batch Programs](#) (see page 302)

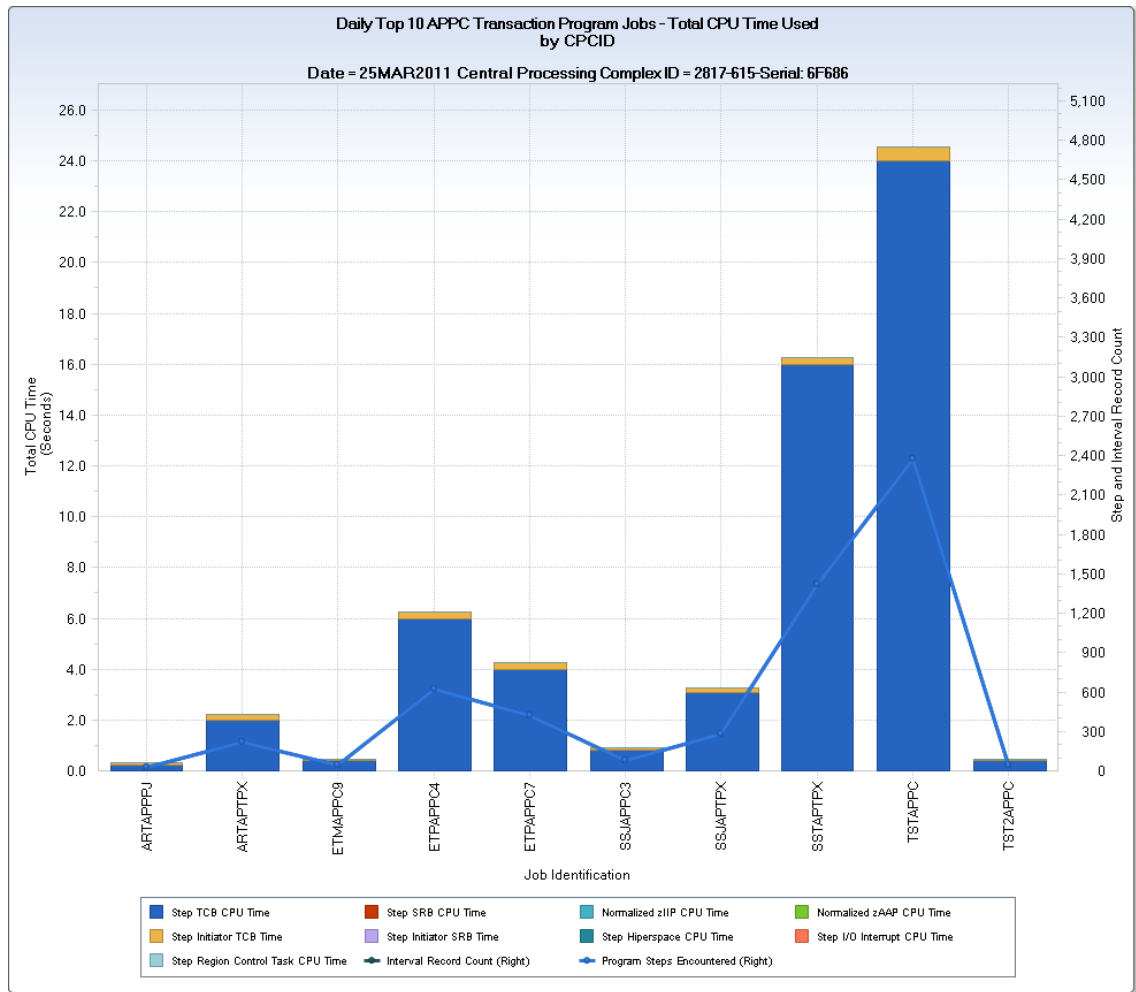
[SMFSAT - SMF Daily Top 10 System Address Spaces](#) (see page 313)

[SMFSJT - SMF Daily Top 10 Started Task Jobs](#) (see page 317)

[SMFTUT - SMF Daily Top 10 TSO Users](#) (see page 323)

[SMFUJT - SMF Daily Top 10 Unix SysSrv Jobs](#) (see page 329)

SMFAJT – SMF Daily Top 10 APPC Transaction Jobs



Overview

The SMFAJT query generates several chart sets. These chart sets show the daily top *nn* (default 10) APPC Transaction Programs (TPs) using CPU, I/O, and central storage (memory) resources on your z/OS systems. For each resource type, one chart set shows the top ten APPC TPs using the resource over the entire Central Processing Complex (CPC), and another that shows the top ten APPC TPs for each z/OS system. The query reads detail data from the BAT_TP (APPC/MVS TP Activity) file, and summarizes the data by job name executed. The charted data reflects summarized interval and step record information from all job steps for the top ten APPC TPs.

Note: While the chart samples displayed in this guide present the Top *nn* APPC TPs alphabetically, from left to right on the x-AXIS, the query has been updated to present the Top *nn* APPC Ts in order of descending resource use. The name of each APPC TP is prefixed with the resource usage rank. (for example, 01- jobnamea, 02- jobnameb, and so on).

For CPU, the CPU time used by each of the top ten APPC TPs is displayed using a vertical bar. The bar is segmented by CPU time type (for example, TCB, SRB, I/O Interrupt, zIIP, zAAP, and so on). Two horizontal lines show the program count (number of unique program executions) and interval count (number of interval records) encountered for the APPC TP. For short running steps, the interval count and program count values have similar values. For long running programs, the interval count exceeds the program count.

Note: The program count data element (PGMCOUNT) is only set to one (1) for the last interval record for a long running program. For APPC TPs with a long running program step that has not yet ended, PGMCOUNT is zero.

For I/O, I/O Service Units are used to determine the top ten APPC TPs. Three different chart views are available to display I/O related resource utilization. The first view shows I/O service units for the top ten using vertical bars. The second view displays device connect time for the top ten by device class (for example, DASD, TAPE, Unit Record, and so on). The device connect time is displayed using a vertical bar, with the bar segmented by the time connected to each of the device classes. The final view shows EXCPs by device class, again using a segmented vertical bar. As with the CPU chart, the program and interval record count are shown using a horizontal line.

For central storage, the top ten APPC TPs with the largest average working set size are displayed. Average working set size is the average number page frames in central storage used by the program. The page frame count is shown using a vertical bar and the program and interval record count are shown using a horizontal line.

Note: The average working set size represents the average for all step and interval records encountered for each top ten APPC TP.

The example shows the CPC level chart generated for APPC TSs executing on an IBM z196 2817-615 CPC. The largest user of CPU time was the APPC TP 'TSTAPPC'. The chart shows that for the day depicted, the job had 2374 program step executions, recorded with 2374 interval records, and consumed just over 24 seconds of CPU time. Most of this time was Step TCB Time.

Uses

It is important to understand which applications are the biggest consumers of limited and expensive IT resources. This query shows the largest APPC TP consumers running on z/OS systems—at both the CPC and individual z/OS system level. Knowing not only the amount of resource consumed, but also the frequency of program execution, can help you to focus tuning efforts where potential payback is greatest. Tracking utilization by job name may or may not be useful at your installation—and is largely dependent on job naming standards.

Data Extracts

Note: The following SMFAJT query data extracts display the same X- and Y-axis data elements as those elements displayed by the SMFBJT query. The only difference is that this SMFAJT query shows the top *nn* APPC TP jobs, while the SMFBJT query output shows the top *nn* batch jobs. For the data extract chart samples and axis values, see the SMFBJT query description.

[SMF Daily Top 10 APPC Jobs CPU CPCID](#)

[SMF Daily Top 10 APPC Jobs CPU SYSID](#)

[SMF Daily Top 10 APPC Jobs IO CPCID](#)

VIEW 1 – SMF Daily Top 10 APPC Jobs I/O – Service Units CPCID

VIEW 2 – SMF Daily Top 10 APPC Jobs I/O – Connect Time CPCID

VIEW 3 – SMF Daily Top 10 APPC Jobs I/O – EXCPs CPCID

[SMF Daily Top 10 APPC Jobs IO SYSID](#)

VIEW 1 – SMF Daily Top 10 APPC Jobs I/O – Service Units SYSID

VIEW 2 – SMF Daily Top 10 APPC Jobs I/O – Connect Time SYSID

VIEW 3 – SMF Daily Top 10 APPC Jobs I/O – EXCPs SYSID

[SMF Daily Top 10 APPC Jobs Memory CPCID](#)

[SMF Daily Top 10 APPC Jobs Memory SYSID](#)

Query Defaults

- HARCPU DETAIL timespan file: Cycle 01
- BAT_TP DETAIL timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

DATE

DATEPART(ENDTS) ;

CPCID

CPCID = PUT(SYSID,\$CPCFMT.) ;

JOB RANK

PUT(COUNT,Z2.) || ' - ' || JOB ;

where COUNT is the rank of resource use (for example, 01 is largest resource user)

TOTCPU

SUM(PGMTCBTM,PGMSRBTM,PGMHIP TM,PGMRCTTM,PGMIOITM,
PGMSPNTM,PGMZPNTM,PGMISR TM,PGMITCTM) ;

INTVLCNT

INTVLCNT=1 ;

(uses existing data element PGMCOACT=1 in the initial input user code, then after summarization sets INTVLCNT=PGMCOACT to deal with Q&R default of treating Q&R derived elements as retained values in summarization).

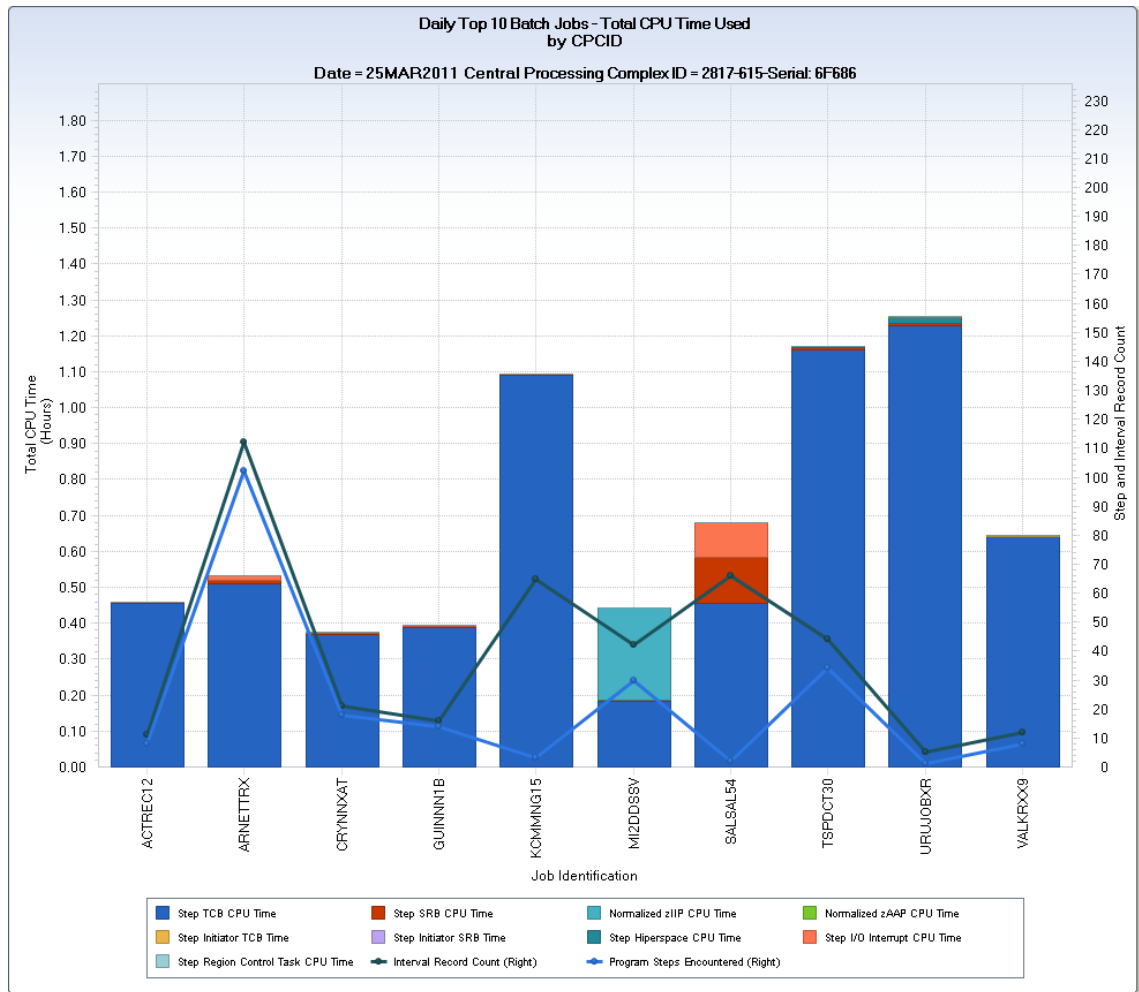
Modifications

The following modifications can be made to the SMFAJT query:

- You can add filter statements to the INPUT BAT_TP step to subset data to restrict analysis to jobs that begin specific characters, specific SYSIDs, and so on.
- You can adjust the TOP nn value to something other than the default of 10.
- You can generate charts for multiple days by reading in more than two cycles, but comment out the Select Yesterday user code in the Input BAT_TP step.
- To minimize SAS WORK space, this query uses a KEEP statement coded as a user code task in the INPUT BAT_TP step. If you want to chart data elements other than those elements in the query, you can either comment out the KEEP statement, or add the data elements you are interested in to the list of elements in the KEEP statement.

Note: This query charts two nonessential data elements: PGMITCTM and PGMISRTM (Initiator TCB and SRB time, respectively). If these data elements are not active in your step level files, the query fails to execute. To run the query, you can remove these two data elements from the ELEMENT SELECTION task in the two CPU time Data Extract steps. You can also ask your CA MICS administrator to activate these two elements in the step level files by changing the element definition in *sharedprefix*.MICS.GENLIB(SMFGENIN) from NAMX to NAME and then running the *sharedprefix*.MICS.CNTL(SMFCGEN) job. After these elements are activated, refresh the Q&R Meta Database.

SMFBJT – SMF Daily Top 10 Batch Jobs



Overview

The SMFBJT query generates several chart sets. The chart sets show the daily top *nn* (default 10) batch jobs using CPU, I/O, and central storage (memory) resources on your z/OS systems. For each resource type, one chart set shows the top ten batch jobs using the resource over the entire Central Processing Complex (CPC), and another that shows the top ten batch jobs for each z/OS system. The query reads detail data from the BATPGM (Batch User Program Activity) file, and summarizes the data by job name. The charted data reflects summarized interval and step record information from all job steps for the top ten batch jobs.

Note: While the chart samples displayed in this guide present the top *nn* batch job names alphabetically, from left to right on the x-AXIS, the query has been updated to present the top *nn* batch jobs in order of descending resource use. The name of each job is prefixed with the resource usage rank. (for example, 01- jobnamea, 02- jobnameb, and so on).

For CPU, the CPU time used by each of the top ten jobs is displayed using a vertical bar. The bar is segmented by CPU time type (for example, TCB, SRB, I/O Interrupt, zIIP, zAAP, and so on). Two horizontal lines show the program count (number of unique program executions) and interval count (number of interval records) encountered for the job. For short running steps, the interval count and program count values have similar values. For long running programs, the interval count exceeds the program count.

Note: The program count data element (PGMCOUNT) is only set to one (1) for the last interval record for a long running program. For jobs with a long running program step that has not yet ended, PGMCOUNT is zero.

For I/O, I/O Service Units are used to determine the top ten jobs. Three different chart views are available to display I/O related resource utilization. The first view shows I/O service units for the top ten using vertical bars. The second view displays device connect time for the top 10 by device class (for example, DASD, TAPE, Unit Record, and so on). The device connect time is displayed using a vertical bar, with the bar segmented by the time connected to each of the device classes. The final view shows EXCPs by device class, again using a segmented vertical bar. As with the CPU chart, the program and interval record count are shown using a horizontal line.

For central storage, the top ten jobs with the largest average working set size are displayed. Average working set size is the average number page frames in central storage used by the program. The page frame count is shown using a vertical bar and the program and interval record count are shown using a horizontal line.

Note: The average working set size represents the average for all step and interval records encountered for each top ten batch job. For this query, average working set size may be of limited value because many batch jobs contain several execution steps with dissimilar execution characteristics.

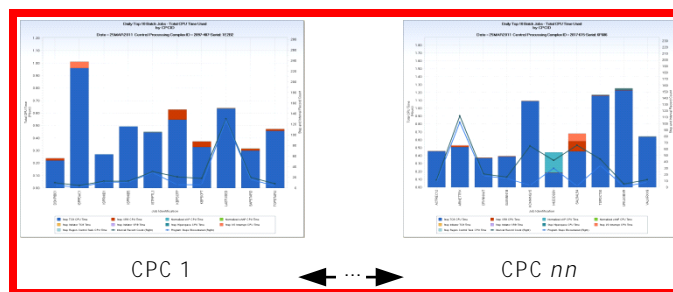
The example shows the CPC level chart generated for jobs executing on an IBM z196 2817-615 CPC. The largest user of CPU time was the job 'URUJOBXR'. The chart shows that for the day depicted, the job only had one step completion spanning five interval records, and consumed just over 1.2 hours of CPU time. Most of this time was Step TCB Time. Job 'SALSAL54' consumed significant SRB and I/O Interrupt CPU time, indicating significant I/O activity.

Uses

It is important to understand which applications are the biggest consumers of limited and expensive IT resources. This query shows the largest batch job consumers running on z/OS systems—at both the CPC and individual z/OS system level. Knowing not only the amount of resource consumed, but also the frequency of program execution, can help you to focus tuning efforts where potential payback is greatest. Tracking utilization by job name may or may not be useful at your installation—and is largely dependent on job naming standards.

Data Extracts:

SMF Daily Top 10 Batch Jobs CPU CPCID



Left Y-axis

- PGMTCBTM – Step TCB CPU Time
- PGMSRBTM – Step SRB CPU Time
- PGMSPNTM – Normalized zIIP CPU Time
- PGMZPNTM – Normalized zAAP CPU Time
- PGMTCTM – Step Initiator TCB Time
- PGMISRTM – Step Initiator SRB Time
- PGMHIPTM – Step Hiperspace CPU Time
- PGMIOITM – Step I/O Interrupt CPU Time
- PGMRCTTM – Step Region Control Task CPU Time

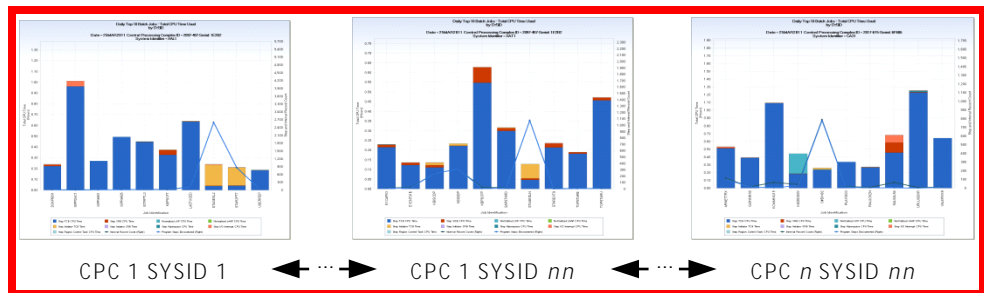
Right Y-axis

- PGMCOUNT – Program Steps Encountered
- INTVLCNT – Interval Record Count

X-axis

JOBRANK – "rank" - Job Identification (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Jobs CPU SYSID



Left Y-axis

- PGMTCBTM – Step TCB CPU Time
- PGMSRBTM – Step SRB CPU Time
- PGMSPNTM – Normalized zIIP CPU Time
- PGMZPNTM – Normalized zAAP CPU Time
- PGMTCTM – Step Initiator TCB Time
- PGMISRTM – Step Initiator SRB Time
- PGMHIPTM – Step Hiperspace CPU Time
- PGMIOITM – Step I/O Interrupt CPU Time
- PGMRCTTM – Step Region Control Task CPU Time

Right Y-axis

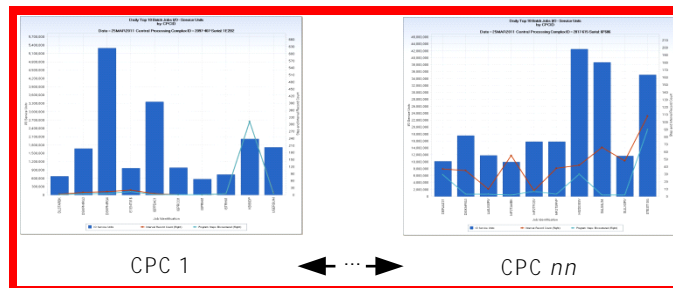
- PGMCOUNT – Program Steps Encountered
- INTVLCNT – Interval Record Count

X-axis

JOB RANK – “rank” - Job Identification (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Jobs IO CPCID

VIEW 1 – SMF Daily Top 10 Batch Jobs I/O – Service Units CPCID



Left Y-axis

PGMIOSU – I/O Service Units

Right Y-axis

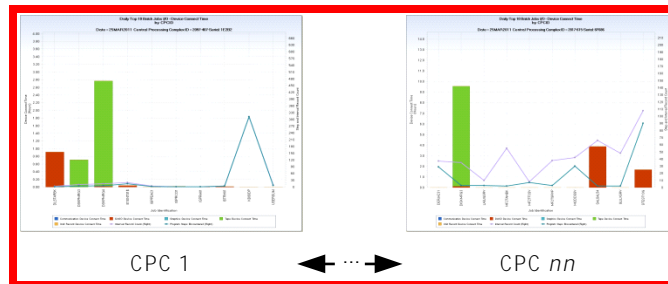
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

JOBRANK – “rank” - Job Identification (where rank is 01, 02, and so on)

VIEW 2 – SMF Daily Top 10 Batch Jobs I/O – Connect Time CPCID



Left Y-axis

PGMCCOMM – Communication Device Connect Time

PGMCDASD – DASD Device Connect Time

PGMCGRAF – Graphics Device Connect Time

PGMCTAPE – Tape Device Connect Time

PGMCUREC – Unit Record Device Connect Time

Right Y-axis

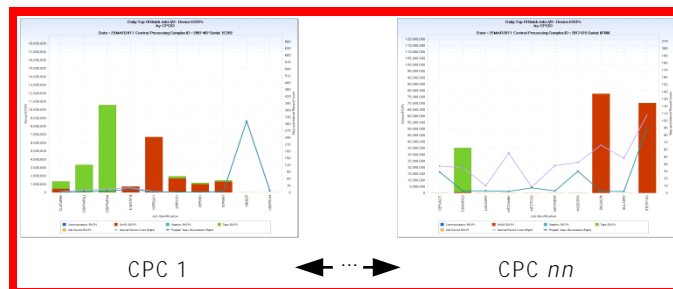
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

JOBRANK – “rank” - Job Identification (where rank is 01, 02, and so on)

VIEW 3 – SMF Daily Top 10 Batch Jobs I/O – EXCPs CPCID



Left Y-axis

PGMCCOMM – Communication Device EXCPS

PGMCDASD – DASD Device EXCPS

PGMCGRAF – Graphics Device EXCPS

PGMCTAPE – Tape Device EXCPS

PGMCUREC – Unit Record Device EXCPS

Right Y-axis

PGMCOUNT – Program Steps Encountered

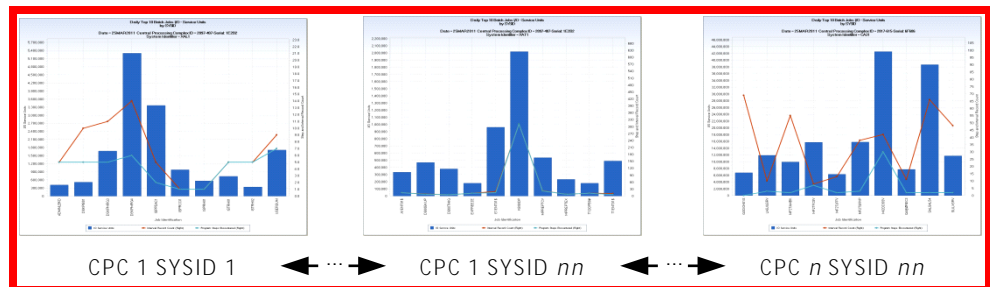
INTVLCNT – Interval Record Count

X-axis

JOB RANK – “rank” - Job Identification (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Jobs IO SYSID

VIEW 1 – SMF Daily Top 10 Batch Jobs I/O – Service Units SYSID



Left Y-axis

PGMIOSU – I/O Service Units

Right Y-axis

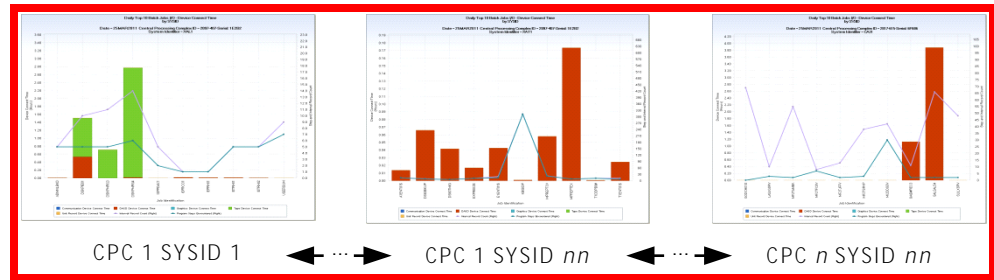
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

JOB RANK – “rank” - Job Identification (where rank is 01, 02, and so on)

VIEW 2 – SMF Daily Top 10 Batch Programs I/O – Connect Time SYSID



Left Y-axis

- PGMCCOMM – Communication Device Connect Time
- PGMCDASD – DASD Device Connect Time
- PGMCGRAF – Graphics Device Connect Time
- PGMCTAPE – Tape Device Connect Time
- PGMCUREC – Unit Record Device Connect Time

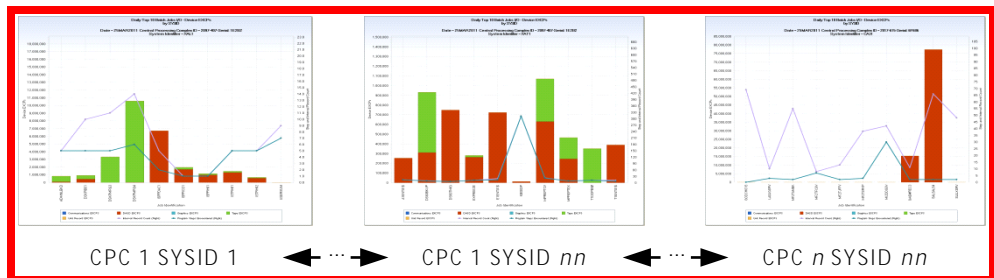
Right Y-axis

- PGMCOUNT – Program Steps Encountered
- INTVLCNT – Interval Record Count

X-axis

JOBIRANK – “rank” - Job Identification (where rank is 01, 02, and so on)

VIEW 3 – SMF Daily Top 10 Batch Programs I/O – EXCPS SYSID



Left Y-axis

- PGMCCOMM – Communication Device EXCPS
- PGMCDASD – DASD Device EXCPS
- PGMCGRAF – Graphics Device EXCPS
- PGMCTAPE – Tape Device EXCPS
- PGMCUREC – Unit Record Device EXCPS

Right Y-axis

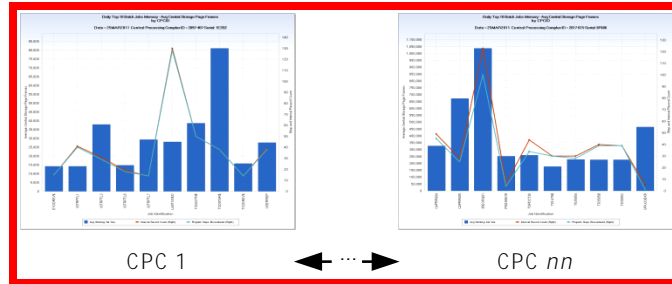
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

JOBRANK – “rank” - Job Identification (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Jobs Memory CPCID



Left Y-axis

PGMAVWSS – Avg Working Set Size

Right Y-axis

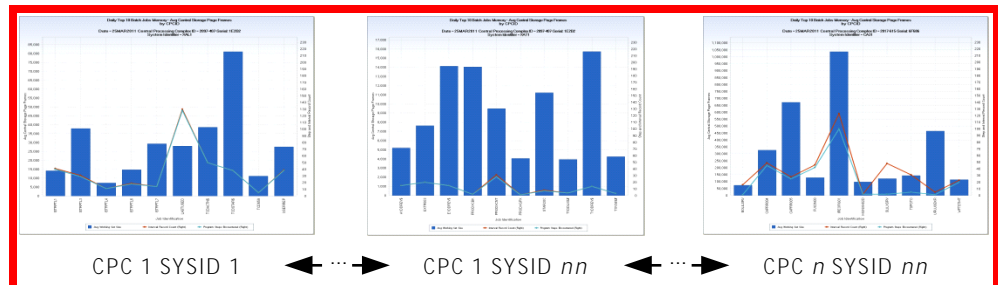
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

JOBRANK – “rank” - Job Identification (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Jobs Memory SYSID



Left Y-axis

PGMAVWSS – Avg Working Set Size

Right Y-axis

PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

JOB RANK – “rank” - Job Identification (where rank is 01, 02, and so on)

Query Defaults

- HARCPU DETAIL timespan file: Cycle 01
- BATPGM DETAIL timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

DATE

DATEPART (ENDTS)

CPCID

CPCID = PUT (SYSID, \$CPCFMT.) ;

JOB RANK

PUT (COUNT, Z2.) || ' - ' || JOB ;

where COUNT is the rank of resource use (for example, 01 is largest resource user)

TOTCPU

SUM (PGMTCBTM, PGMSRBTM, PGMHIPTM, PGMRCTTM, PGMI0ITM,
PGMSPNTM, PGMZPNTM, PGMISRTM, PGMITCTM) ;

INTVLCNT

INTVLCNT=1 ;

(uses existing data element PGMCOACT=1 in the initial input user code, then after summarization sets INTVLCNT=PGMCOACT to deal with Q&R default of treating Q&R derived elements as retained values in summarization).

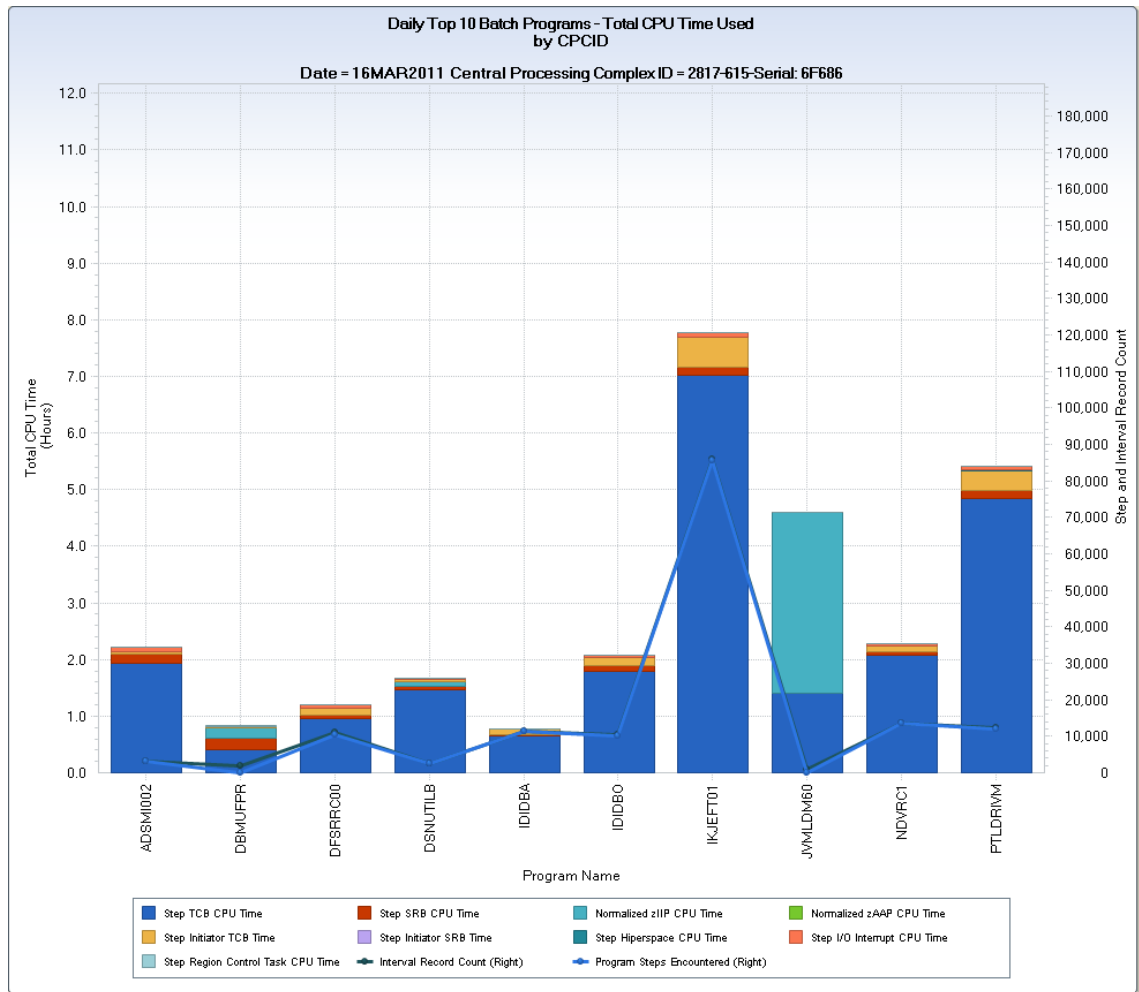
Modifications

The following modifications can be made to the SMFBJT query:

- You can add filter statements to the INPUT BATPGM step to subset data to restrict analysis to jobs that begin specific characters, specific SYSIDs, and so on.
- You can adjust the *TOPnn* value to something other than the default of 10.
- You can generate charts for multiple days by reading in more than two cycles, but comment out the Select Yesterday user code in the Input BATPGM step.
- To minimize SAS WORK space, this query uses a KEEP statement coded as a user code task in the INPUT BATPGM step. If you want to chart data elements other than those elements in the query, you can either comment out the KEEP statement, or add the data elements you are interested in to the list of elements in the KEEP statement.

Note: This query charts two nonessential data elements: PGMITCTM and PGMISRTM (Initiator TCB and SRB time, respectively). If these data elements are not active in your step level files, the query fails to execute. To run the query, you can remove these two data elements from the ELEMENT SELECTION task in the two CPU time Data Extract steps. You can also ask your CA MICS administrator to activate these two elements in the step level files by changing the element definition in *sharedprefix.MICS.GENLIB(SMFGENIN)* from NAMX to NAME and then running the *sharedprefix.MICS.CNTL(SMFCGEN)* job. After these elements are activated, refresh the Q&R Meta Database.

SMFBPT – SMF Daily Top 10 Batch Programs



Overview

The SMFBPT query generates several chart sets. The chart sets show the daily top *nn* (default 10) batch programs using CPU, I/O, and central storage (memory) resources on your z/OS systems. For each resource type, one chart set shows the top ten batch programs using the resource over the entire Central Processing Complex (CPC), and another that shows the top ten batch programs for each z/OS system. The query reads detail data from the BATPGM (Batch User Program Activity) file, and summarizes the data by program name executed. The charted data reflects interval and step record information from all batch job steps that executed the top ten programs over all batch jobs.

Note: While the chart samples displayed in this guide present the top *nn* batch program names alphabetically, from left to right on the x-AXIS, the query has been updated to present the top *nn* batch programs in order of descending resource use. The name of each program is prefixed with the resource usage rank. (for example, 01- programa, 02-programb, and so on.

For CPU, the CPU time used by each of the top ten programs is displayed using a vertical bar. The bar is segmented by CPU time type (for example, TCB, SRB, I/O Interrupt, zIIP, zAAP, and so on). Two horizontal lines show the program count (number of unique program executions) and interval count (number of interval records) encountered for the program. For short running steps, the interval count and program count values have similar values. For long running programs, the interval count exceeds the program count.

Note: The program count data element (PGMCOUNT) is only set to one (1) for the last interval record for a long running program. For programs that have not yet ended, PGMCOUNT is zero.

For I/O, I/O Service Units are used to determine the top ten programs. Three different chart views are available to display I/O related resource utilization. The first view shows I/O service units for the top ten using vertical bars. The second view displays device connect time for the top ten by device class (for example, DASD, TAPE, Unit Record, and so on). The device connect time is displayed using a vertical bar, with the bar segmented by the time connected to each of the device classes. The final view shows EXCPs by device class, again using a segmented vertical bar. As with the CPU chart, the program and interval record count are shown using a horizontal line.

For central storage, the top ten programs with the largest average working set size are displayed. Average working set size is the average number page frames in central storage used by the program. The page frame count is shown using a vertical bar and the program and interval record count are shown using a horizontal line.

Note: The average working set size represents the average for all step and interval records encountered for instance of the top ten batch programs.

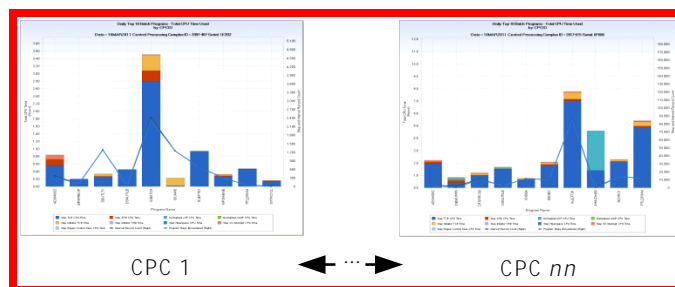
The example shows the CPC level chart generated for batch programs executing on an IBM z196 2817-615 CPC. The largest user of CPU time is the program 'IKJEFT01', the program name for TSO, meaning that many batch job program steps invoke TSO on this CPC. The chart shows that for the day depicted, about 86,000 batch job program steps executed TSO in batch and, in total, consumed almost 8 hours of CPU time. Most of this time was Step TCB Time. Another top ten program, 'JMMLDM60', only executed about 100 times, but consumed about 1.5 hours of TCB time and 3.2 hours of normalized zIIP CPU time. The normalization of zIIP CPU time is important for this CPC model because the CP engines run slower than the specialty (zIIP/zAAP) engines. When zIIP or zAAP CPU time is normalized, the value is expressed as the CP engine equivalent by multiplying by a normalization factor. For the 2817-615 CPC model, the normalization factor is 1.52734.

Uses

It is important to understand which applications are the biggest consumers of limited and expensive IT resources. This query shows the largest batch program consumers running on z/OS systems—at both the CPC and individual z/OS system level. Knowing not only the amount of resource consumed, but also the frequency of program execution, can help you to focus tuning efforts where potential payback is greatest.

Data Extracts

SMF Daily Top 10 Batch Pgms CPU CPCID



Left Y-axis

PGMTCBTM – Step TCB CPU Time

PGMSRBTM – Step SRB CPU Time

PGMSPNTM – Normalized zIIP CPU Time

PGMZPNTM – Normalized zAAP CPU Time

PGMTCTM – Step Initiator TCB Time

PGMISRTM – Step Initiator SRB Time

PGMHPTM – Step Hiperspace CPU Time

PGMIOITM – Step I/O Interrupt CPU Time

PGMRCTTM – Step Region Control Task CPU Time

Right Y-axis

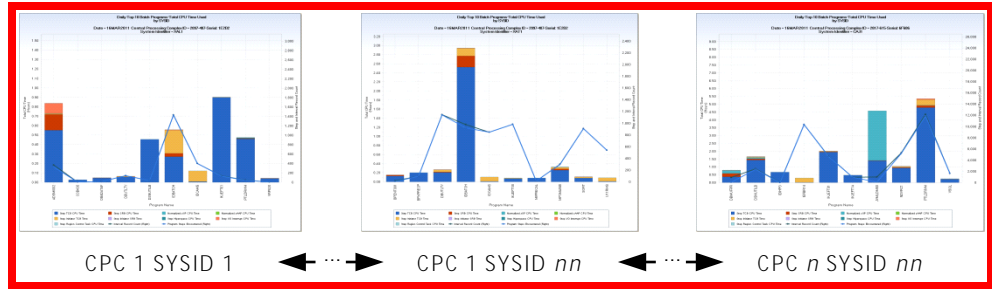
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Pgms CPU SYSID



Left Y-axis

- PGMTCBTM – Step TCB CPU Time
- PGMSRBTM – Step SRB CPU Time
- PGMSPNTM – Normalized zIIP CPU Time
- PGMZPNTM – Normalized zAAP CPU Time
- PGMTCTM – Step Initiator TCB Time
- PGMISRTM – Step Initiator SRB Time
- PGMHPTM – Step Hiperspace CPU Time
- PGMIOITM – Step I/O Interrupt CPU Time
- PGMRCTTM – Step Region Control Task CPU Time

Right Y-axis

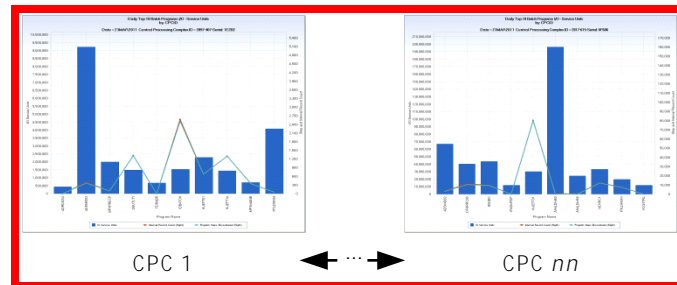
- PGMCOUNT – Program Steps Encountered
- INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Pgms IO CPCID

VIEW 1 – SMF Daily Top 10 Batch Programs I/O – Service Units CPCID



Left Y-axis

PGMIOSU – I/O Service Units

Right Y-axis

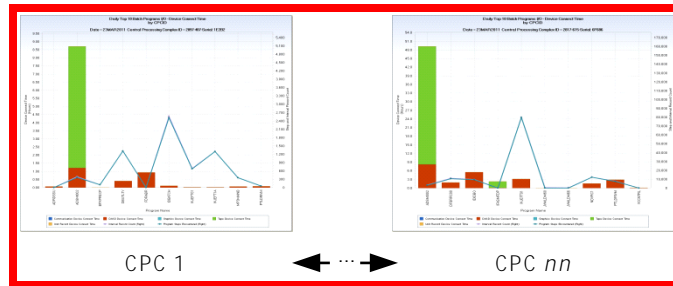
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

VIEW 2 – SMF Daily Top 10 Batch Programs I/O – Connect Time CPCID



Left Y-axis

PGMCCOMM – Communication Device Connect Time

PGMCDASD – DASD Device Connect Time

PGMCGRAF – Graphics Device Connect Time

PGMCTAPE – Tape Device Connect Time

PGMCUREC – Unit Record Device Connect Time

Right Y-axis

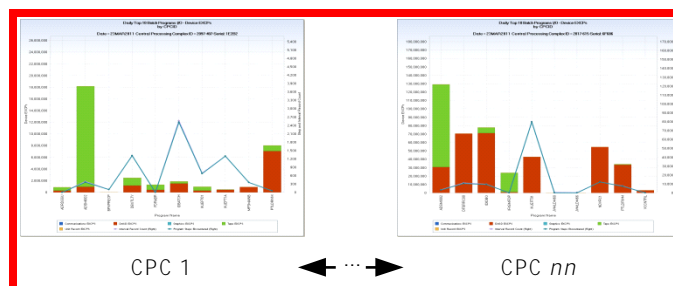
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

VIEW 3 – SMF Daily Top 10 Batch Programs I/O – EXCPs CPCID



Left Y-axis

PGMCCOMM – Communication Device EXCPS

PGMCDASD – DASD Device EXCPS

PGMCGRAF – Graphics Device EXCPS

PGMCTAPE – Tape Device EXCPS

PGMCUREC – Unit Record Device EXCPS

Right Y-axis

PGMCOUNT – Program Steps Encountered

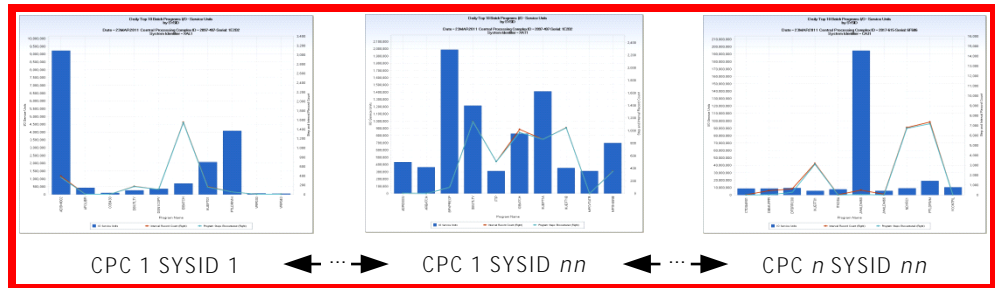
INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Pgms IO SYSID

VIEW 1 – SMF Daily Top 10 Batch Programs I/O – Service Units SYSID



Left Y-axis

PGMIOSU – I/O Service Units

Right Y-axis

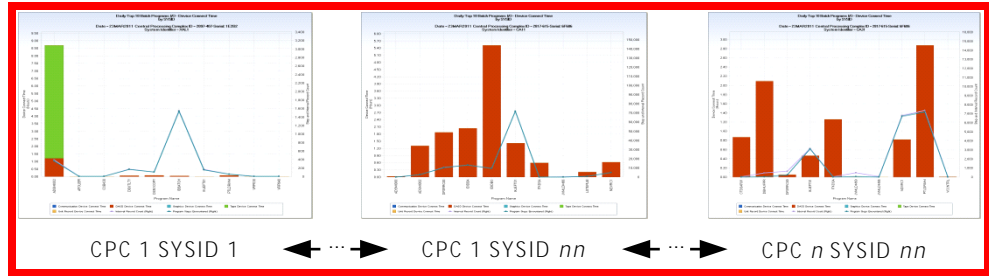
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

VIEW 2 – SMF Daily Top 10 Batch Programs I/O – Connect Time SYSID



Left Y-axis

- PGMCCOMM – Communication Device Connect Time
- PGMCDASD – DASD Device Connect Time
- PGMCGRAF – Graphics Device Connect Time
- PGMCTAPE – Tape Device Connect Time
- PGMCUREC – Unit Record Device Connect Time

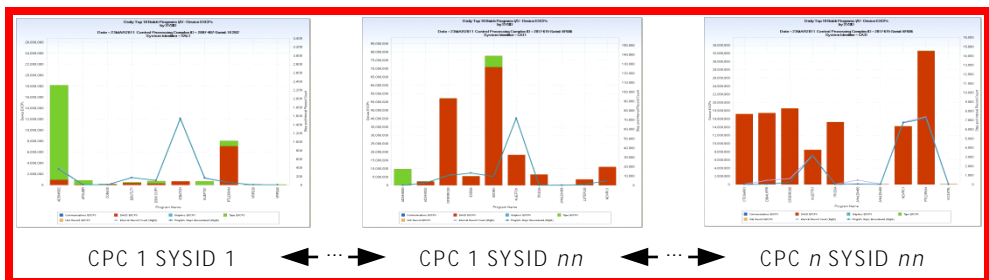
Right Y-axis

- PGMCOUNT – Program Steps Encountered
- INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

VIEW 3 – SMF Daily Top 10 Batch Programs I/O – EXCPS SYSID



Left Y-axis

- PGMCCOMM – Communication Device EXCPS
- PGMCDASD – DASD Device EXCPS
- PGMCGRAF – Graphics Device EXCPS
- PGMCTAPE – Tape Device EXCPS
- PGMCUREC – Unit Record Device EXCPS

Right Y-axis

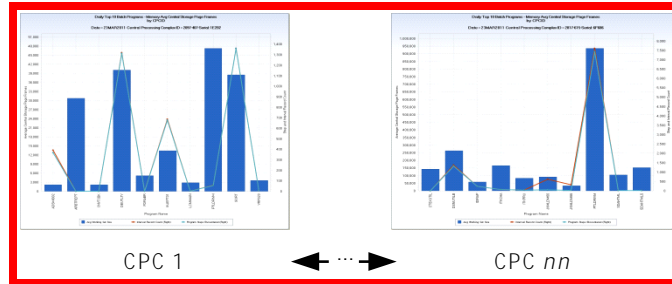
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Pgms Memory CPCID



Left Y-axis

PGMAVWSS – Avg Working Set Size

Right Y-axis

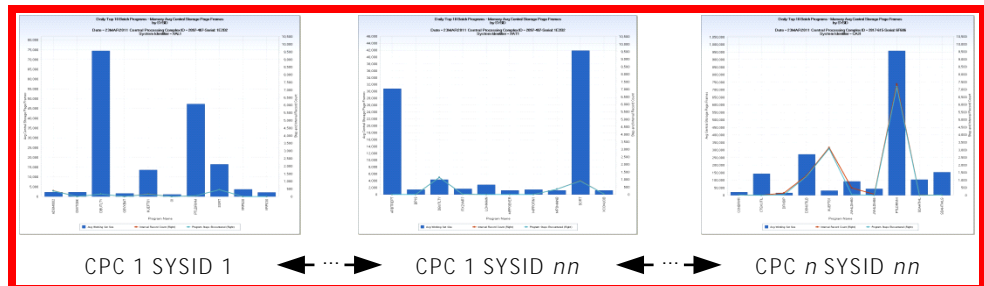
PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

SMF Daily Top 10 Batch Pgms Memory SYSID



Left Y-axis

PGMAVWSS – Avg Working Set Size

Right Y-axis

PGMCOUNT – Program Steps Encountered

INTVLCNT – Interval Record Count

X-axis

PGMRANK – “rank” – Program Name (where rank is 01, 02, and so on)

Query Defaults

- HARCPU DETAIL timespan file: Cycle 01
- BATPGM DETAIL timespan file: Cycles 01-02
- User code selects yesterday’s data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

DATE

DATEPART(ENDTS)

CPCID

CPCID = PUT(SYSID,\$CPCFMT.) ;

PGMRANK

PUT(COUNT,Z2.) || ' - ' || PROGRAM ;

where COUNT is the rank of resource use (for example, 01 is largest resource user)

TOTCPU

SUM(PGMTCBTM,PGMSRBTM,PGMHIPTM,PGMRCTTM,PGMIOITM,
PGMSPNTM,PGMZPNTM,PGMISRTM,PGMITCTM) ;

INTVLCNT

INTVLCNT=1 ;

(uses existing data element PGMCOACT=1 in the initial input user code, then after summarization sets INTVLCNT=PGMCOACT to deal with Q&R default of treating Q&R derived elements as retained values in summarization).

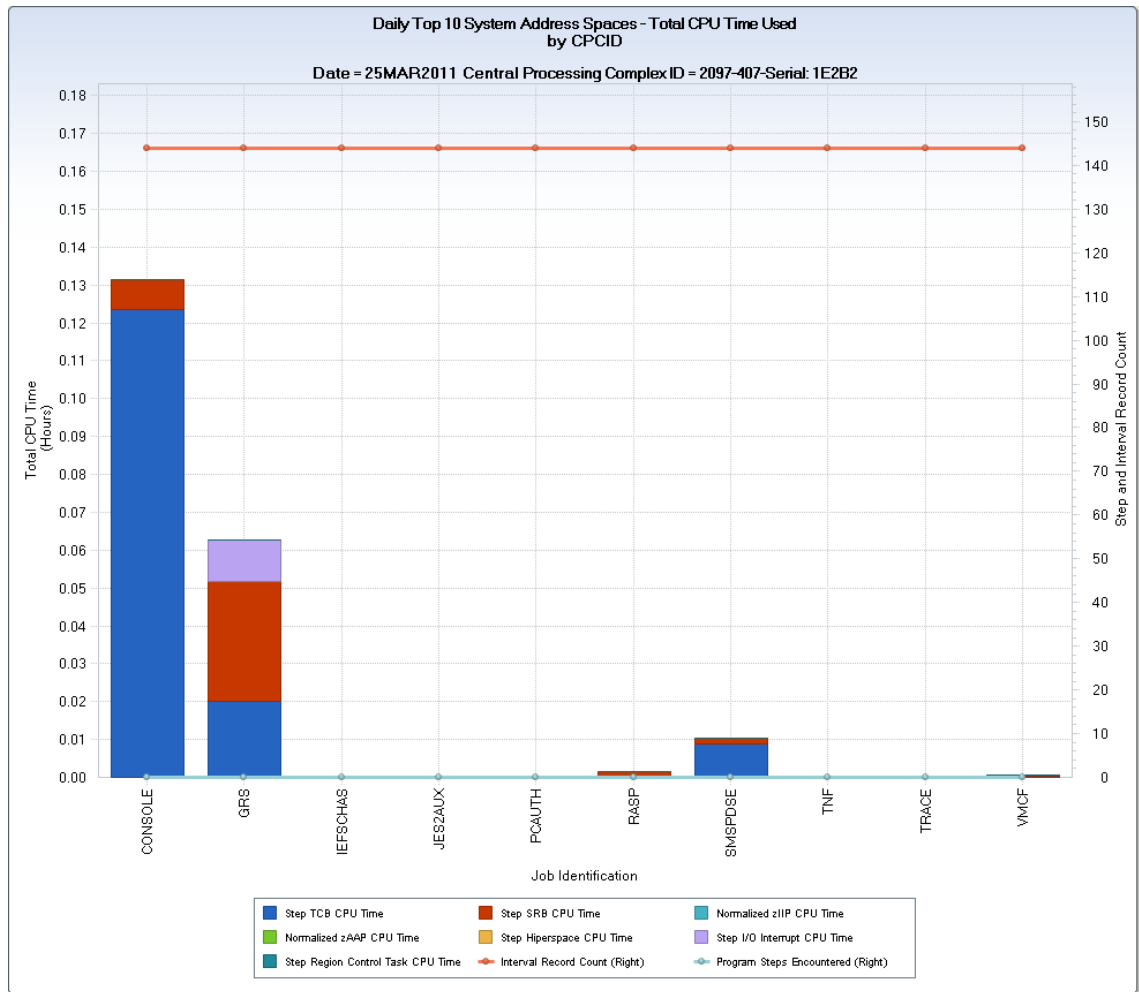
Modifications

The following modifications can be made to the SMFBPT query:

- You can add filter statements to the INPUT BATPGM step to subset data to restrict analysis to programs that begin specific characters, specific SYSIDs, and so on.
- You can adjust the *TOPnn* value to something other than the default of 10.
- You can generate charts for multiple days by reading in more than two cycles, but comment out the Select Yesterday user code in the Input BATPGM step.
- To minimize SAS WORK space, this query uses a KEEP statement coded as a user code task in the INPUT BATPGM step. If you want to chart data elements other than those elements in the query, you can either comment out the KEEP statement, or add the data elements you are interested in to the list of elements in the KEEP statement.

Note: This query charts two nonessential data elements: PGMITCTM and PGMISRTM (Initiator TCB and SRB time, respectively). These data elements are defined as 'nonessential' in the CA MICS database and may not be active in your step level files based on how CA MICS is generated. If not active, the query fails to execute. To run the query, you can remove these two data elements from the ELEMENT SELECTION task in the two CPU time Data Extract steps. You can also ask your CA MICS administrator to activate these two elements in the step level files by changing the element definition in *sharedprefix*.MICS.GENLIB(SMFGENIN) from NAMX to NAME and then running the *sharedprefix*.MICS.CNTL(SMFCGEN) component generation job. After these elements are activated and populated in your CA MICS database, refresh the Q&R Meta Database.

SMFSAT – SMF Daily Top 10 System Address Spaces



Overview

The SMFBJT query generates several chart sets. The chart sets show the daily top *nn* (default 10) system address spaces (SASs) using CPU, I/O, and central storage (memory) resources on your z/OS systems. For each resource type, one chart set shows the top ten SASs using the resource over the entire Central Processing Complex (CPC), and another that shows the top ten SASs for each z/OS system. The query reads detail data from the BAT_SA (System Address Space Activity) file, and summarizes the data by job name. The charted data reflects summarized interval and step record information from all job steps for the top ten system address spaces.

Note: While the chart samples displayed in this guide present the Top *nn* system address spaces alphabetically, from left to right on the x-AXIS, the query has been updated to present the Top *nn* system address spaces in order of descending resource use. The name of each system address space is prefixed with the resource usage rank. (for example, 01- jobnamea, 02- jobnameb, and so on.)

For CPU, the CPU time used by each of the top ten SASs is displayed using a vertical bar. The bar is segmented by CPU time type (for example, TCB, SRB, I/O Interrupt, zIIP, zAAP, and so on). Two horizontal lines show the program count (number of unique program executions) and interval count (number of interval records) encountered for the SAS. System address spaces are the first address spaces that initiate on a z/OS system. These address spaces tend to remain up until the system is brought down. In general, the only data processed is interval records. The program count is zero and the interval count identifies the number of individual detail records processed.

Note: The program count data element (PGMCOUNT) is only set to one (1) for the last interval record for a long running system address space.

For I/O, I/O Service Units are used to determine the top ten SASs. Unlike the other address space types (for example, batch, TSO, and so on), there is no device-specific information available for SASs. The only I/O related information is I/O service units. The device connect time and device EXCP views generated for the other Top 10 queries are not available for this SMFSAT query. The I/O service units for the top ten SASs are displayed using vertical bars. As with the CPU chart, the program and interval record count are shown using a horizontal line.

For central storage, the top ten SASs with the largest average working set size are displayed. Average working set size is the average number page frames in central storage used by the program. The page frame count is shown using a vertical bar and the program and interval record count are shown using a horizontal line.

Note: The average working set size represents the average for all step and interval records encountered for each top ten system address space.

The example shows the CPC level chart generated for the system address spaces executing on an IBM z10 2097-407 CPC. The two largest users of CPU time were 'CONSOLE' and 'GRS.' 'CONSOLE' primarily consumed TCB CPU time. The 'GRS' system address space consumed more SRB and I/O Interrupt CPU time than TCB time, indicating significant I/O activity.

Uses

It is important to understand which applications are the biggest consumers of limited and expensive IT resources. This query shows the largest system address space consumers running on z/OS systems—at both the CPC and individual z/OS system level. Knowing not only the amount of resource consumed, but also the frequency of program execution, can help you to focus tuning efforts where potential payback is greatest.

Data Extracts

Note: The following SMFSAT query data extracts display the same X- and Y-axis data elements as those elements displayed by the SMFBJT query. The only difference is that this SMFSAT query shows the top *nn* system address spaces, while the SMFBJT query output shows the top *nn* batch jobs. For the data extract chart samples and axis values, see the SMFBJT query description.

[SMF Daily Top 10 Sys Address CPU CPCID](#)

Note: Initiator TCB and SRB CPU times (PGMITCTM and PGMISRTM) are not available for system address spaces.

[SMF Daily Top 10 Sys Address CPU SYSID](#)

Note: Initiator TCB and SRB CPU times (PGMITCTM and PGMISRTM) are not available for system address spaces.

[SMF Daily Top 10 Sys Address IO CPCID](#)

Note: No multiple views—only I/O service units are available for system address spaces.

[SMF Daily Top 10 Sys Address IO SYSID](#)

Note: No multiple views—only I/O service units are available for system address spaces.

[SMF Daily Top 10 Sys Address Memory CPCID](#)

[SMF Daily Top 10 Sys Address Memory SYSID](#)

Query Defaults

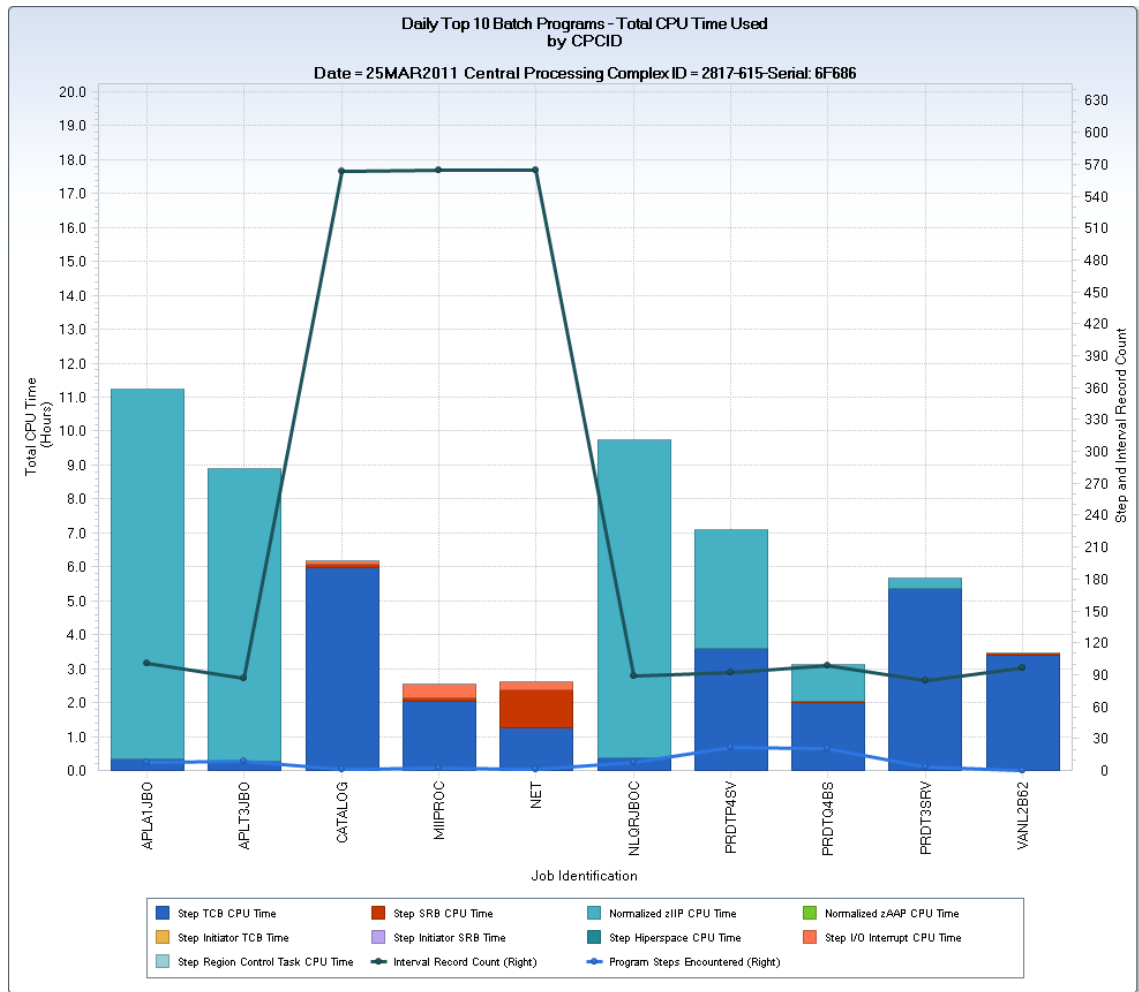
- HARCPU DETAIL timespan file: Cycle 01
- BAT_SA DETAIL timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
DATE
DATEPART(ENDTS)
CPCID
CPCID = PUT(SYSID,\$CPCFMT.) ;
JOB RANK
PUT(COUNT,Z2.) || ' - ' || JOB ;
where COUNT is the rank of resource use (e.g., 01 is largest resource user)
TOTCPU
SUM(PGMTCBTM,PGMSRBTM,PGMHIPTM,PGMRCTTM,PGMIOITM,
PGMSPNTM,PGMZPNTM,PGMISRTM,PGMITCTM) ;
INTVLCNT
INTVLCNT=1 ;
(uses existing data element PGMCOACT=1 in the initial input user code, then after summarization sets INTVLCNT=PGMCOACT to deal with Q&R default of treating Q&R derived elements as retained values in summarization).

Modifications

The following modifications can be made to the SMFSAT query:

- You can add filter statements to the INPUT BAT_SA step to subset data to restrict analysis to system address spaces that begin specific characters, specific SYSIDs, and so on.
- You can adjust the TOP nn value to something other than the default of 10.
- You can generate charts for multiple days by reading in more than two cycles, but comment out the Select Yesterday user code in the Input BAT_SA step.
- To minimize SAS WORK space, this query uses a KEEP statement coded as a user code task in the INPUT BAT_SA step. If you want to chart data elements other than those elements in the query, you can either comment out the KEEP statement, or add the data elements you are interested in to the list of elements in the KEEP statement.

SMFSJT – SMF Daily Top 10 Started Task Jobs



Overview

The SMFSJT query generates several chart sets. The chart sets show the daily top *nn* (default 10) started tasks, by job name, using CPU, I/O, and central storage (memory) resources on your z/OS systems. For each resource type, one chart set shows the top ten started tasks using the resource over the entire Central Processing Complex (CPC), and another that shows the top ten started tasks for each z/OS system. The query reads detail data from the BAT_ST (System Task Program Activity) file, and summarizes the data by job name. The charted data reflects summarized interval and step record information from all job steps for the top ten started tasks.

Note: While the chart samples displayed in this guide present the Top *nn* started tasks alphabetically, from left to right on the x-AXIS, the query has been updated to present the Top *nn* started tasks in order of descending resource use. The name of each started task is prefixed with the resource usage rank. (for example, 01- jobnamea, 02-jobnameb, and so on.

For CPU, the CPU time used by each of the top ten started tasks is displayed using a vertical bar. The bar is segmented by CPU time type (for example, TCB, SRB, I/O Interrupt, zIIP, zAAP, and so on). Two horizontal lines show the program count (number of unique program executions) and interval count (number of interval records) encountered for the started task. For short running steps, the interval count and program count values have similar values. For long running programs, the interval count exceeds the program count.

Note: The program count data element (PGMCOUNT) is only set to one (1) for the last interval record for a long running program. For started tasks with a long running program step that has not yet ended, PGMCOUNT is zero.

For I/O, I/O Service Units are used to determine the top ten started tasks. Three different chart views are available to display I/O related resource utilization. The first view shows I/O service units for the top ten using vertical bars. The second view displays device connect time for the top 10 by device class (for example, DASD, TAPE, Unit Record, and so on). The device connect time is displayed using a vertical bar, with the bar segmented by the time connected to each of the device classes. The final view shows EXCPs by device class, again using a segmented vertical bar. As with the CPU chart, the program and interval record count are shown using a horizontal line.

For central storage, the top ten started tasks with the largest average working set size are displayed. Average working set size is the average number page frames in central storage used by the program. The page frame count is shown using a vertical bar and the program and interval record count are shown using a horizontal line.

Note: The average working set size represents the average for all step and interval records encountered for each top ten started task.

The example shows the CPC level chart generated for started tasks executing on an IBM z196 2817-615 CPC. The largest user of CPU time was the started task 'APLA1JBO'. The chart shows that for the day depicted, the started task only had 8 step completions spanning 101 interval records, and consumed just over 11 hours of CPU time. Most of this time was Normalized zIIP CPU time. The normalization of zIIP CPU time is important for this CPC model because the CP engines run slower than the specialty (zIIP/zAAP) engines. When zIIP or zAAP CPU time is normalized, the value is expressed as the CP engine equivalent by multiplying by a normalization factor. For the 2817-615 CPC model, the normalization factor is 1.52734.

Uses

It is important to understand which applications are the biggest consumers of limited and expensive IT resources. This query shows the largest started task consumers running on z/OS systems—at both the CPC and individual z/OS system level. Knowing not only the amount of resource consumed, but also the frequency of program execution, can help you to focus tuning efforts where potential payback is greatest. Tracking started task utilization by job name may or may not be useful at your installation—and is largely dependent on job naming standards.

Data Extracts

Note: The following SMFSJT query data extracts display the same X- and Y-axis data elements as those elements displayed by the SMFBJT query. The only difference is that this SMFSJT query shows the top *nn* started task jobs, while the SMFBJT query output shows the top *nn* batch jobs. For the data extract chart samples and axis values, see the SMFBJT query description.

[SMF Daily Top 10 STask Jobs CPU CPCID](#)

[SMF Daily Top 10 STask Jobs CPU SYSID](#)

[SMF Daily Top 10 STask Jobs IO CPCID](#)

[VIEW 1 – SMF Daily Top 10 STask Jobs I/O – Service Units CPCID](#)

[VIEW 2 – SMF Daily Top 10 STask Jobs I/O – Connect Time CPCID](#)

[VIEW 3 – SMF Daily Top 10 STask Jobs I/O – EXCPs CPCID](#)

[SMF Daily Top 10 STask Jobs IO SYSID](#)

[VIEW 1 – SMF Daily Top 10 STask Jobs I/O – Service Units SYSID](#)

[VIEW 2 – SMF Daily Top 10 STask Jobs I/O – Connect Time SYSID](#)

[VIEW 3 – SMF Daily Top 10 STask Jobs I/O – EXCPs SYSID](#)

[SMF Daily Top 10 STask Jobs Memory CPCID](#)

[SMF Daily Top 10 STask Jobs Memory SYSID](#)

Query Defaults

- HARCPU DETAIL timespan file: Cycle 01
- BAT_ST DETAIL timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART(ENDTS)
 - CPCID
 - CPCID = PUT(SYSID,\$CPCFMT.) ;
 - JOBRANK
 - PUT(COUNT,Z2.) || ' - ' || JOB ;
 - where COUNT is the rank of resource use (e.g., 01 is largest resource user)
 - TOTCPU
 - SUM(PGMTCBTM,PGMSRBTM,PGMHIPTM,PGMRCTTM,PGMIOITM,
PGMSPNTM,PGMZPNTM,PGMISRTM,PGMITCTM) ;
 - INTVLCNT
 - INTVLCNT=1 ;
 - (uses existing data element PGMCOACT=1 in the initial input user code, then after summarization sets INTVLCNT=PGMCOACT to deal with Q&R default of treating Q&R derived elements as retained values in summarization).

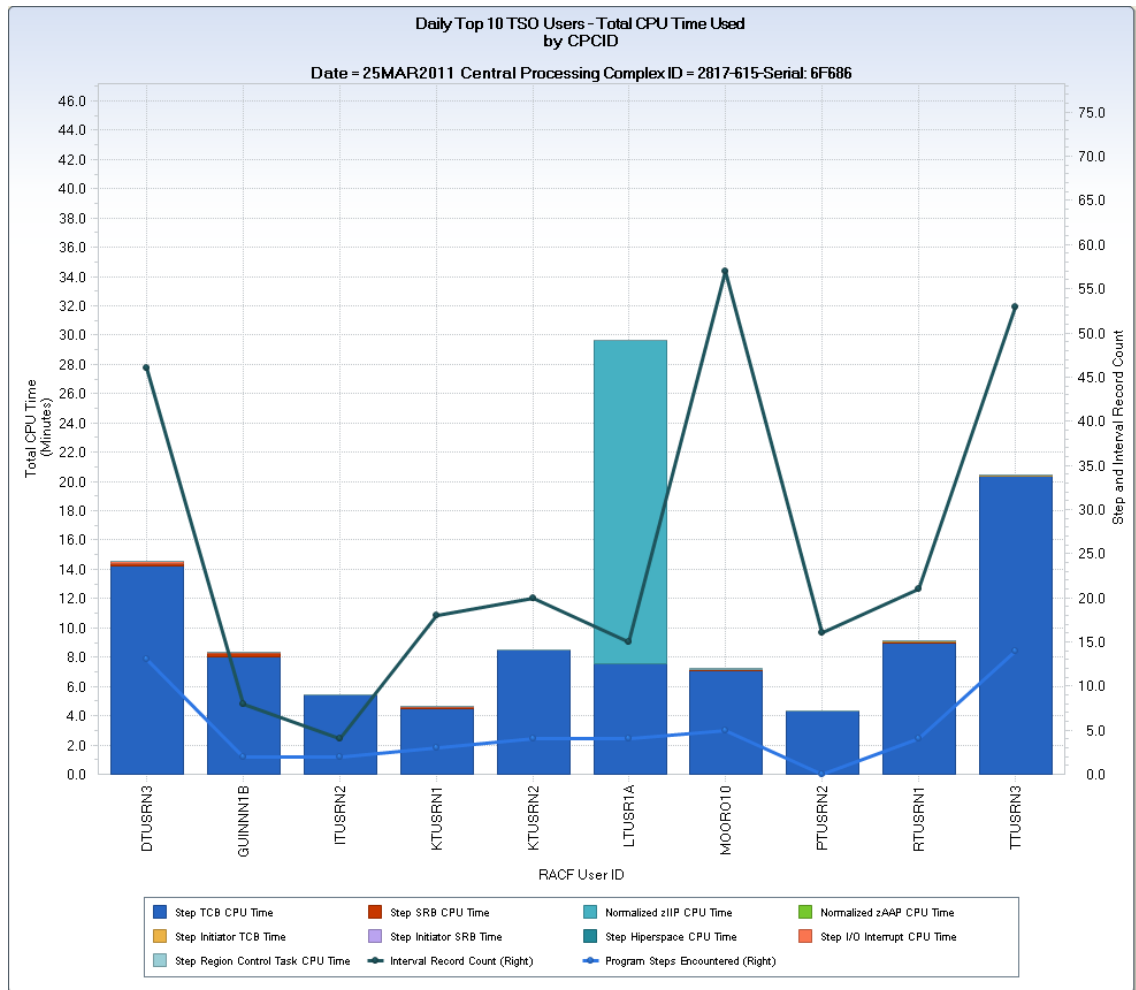
Modifications

The following modifications can be made to the SMFSJT query:

- You can add filter statements to the INPUT BAT_ST step to subset data to restrict analysis to jobs that begin specific characters, specific SYSIDs, and so on.
- You can adjust the TOP nn value to something other than the default of 10.
- You can generate charts for multiple days by reading in more than two cycles, but comment out the Select Yesterday user code in the Input BAT_ST step.
- To minimize SAS WORK space, this query uses a KEEP statement coded as a user code task in the INPUT BAT_ST step. If you want to chart data elements other than those elements in the query, you can either comment out the KEEP statement, or add the data elements you are interested in to the list of elements in the KEEP statement.

Note: This query charts two nonessential data elements: PGMITCTM and PGMISRTM (Initiator TCB and SRB time, respectively). If these data elements are not active in your step level files, the query fails to execute. To run the query, you can remove these two data elements from the ELEMENT SELECTION task in the two CPU time Data Extract steps. You can also ask your CA MICS administrator to activate these two elements in the step level files by changing the element definition in *sharedprefix*.MICS.GENLIB(SMFGENIN) from NAMX to NAME and then running the *sharedprefix*.MICS.CNTL(SMFCGEN) job. After these elements are activated, refresh the Q&R Meta Database.

SMFTUT – SMF Daily Top 10 TSO Users



Overview

The SMFTUT query generates several chart sets. The chart sets show the daily top *nn* (default 10) TSO users based on use of CPU, I/O, and central storage (memory) resources on your z/OS systems. For each resource type, one chart set shows the top ten TSO users using the resource over the entire Central Processing Complex (CPC), and another that shows the top ten TSO users for each z/OS system. The query reads detail data from the BAT_TS (SMF User TSO Activity) file, and summarizes the data by User Identification. The charted data reflects summarized interval and step record information from all TSO sessions for the top ten TSO users.

Note: While the chart samples displayed in this guide present the Top *nn* TSO users alphabetically, from left to right on the x-AXIS, the query has been updated to present the Top *nn* TSO users in order of descending resource use. The name of each TSO user is prefixed with the resource usage rank. (for example, 01- jobnamea, 02- jobnameb, and so on.

Note: The user identification field, RACFUSID, implies RACF, but the user ID information is present even if other non-RACF security systems are used.

For CPU, the CPU time used by each of the top ten TSO users is displayed using a vertical bar. The bar is segmented by CPU time type (for example, TCB, SRB, I/O Interrupt, zIIP, zAAP, and so on). Two horizontal lines show the program count (number of unique TSO sessions) and interval count (number of interval records) encountered for the user. For short running TSO sessions, the interval count and program count values have similar values. For long running sessions, the interval count exceeds the program count.

Note: The program count data element (PGMCOUNT) is only set to one (1) for the last interval record for a long running TSO session. For long running sessions that has not yet ended, PGMCOUNT is zero.

For I/O, I/O Service Units are used to determine the top 10 TSO users. Three different chart views are available to display I/O related resource utilization. The first view shows I/O service units for the top ten using vertical bars. The second view displays device connect time for the top 10 by device class (for example, DASD, TAPE, Unit Record, and so on). The device connect time is displayed using a vertical bar, with the bar segmented by the time connected to each of the device classes. The final view shows EXCPs by device class, again using a segmented vertical bar. As with the CPU chart, the program and interval record count are shown using a horizontal line.

For central storage, the top ten TSO users with the largest average working set size are displayed. Average working set size is the average number page frames in central storage used during the TSO session. The page frame count is shown using a vertical bar and the session and interval record count are shown using a horizontal line.

Note: The average working set size represents the average for all step and interval records encountered for each top ten TSO user.

The example shows the CPC level chart generated for jobs executing on an IBM z196 2817-615 CPC. The largest TSO user of CPU time was 'LTUSR1A'. The chart shows that for the day depicted, the user had 4 TSO sessions completed spanning 15 interval records, and consumed almost 30 minutes of CPU time. Most of this time was Normalized zIIP CPU Time. The normalization of zIIP CPU time is important for this CPC model because the CP engines run slower than the specialty (zIIP/zAAP) engines. When zIIP or zAAP CPU time is normalized, the value is expressed as the CP engine equivalent by multiplying by a normalization factor. For the 2817-615 CPC model, the normalization factor is 1.52734.

Uses

It is important to understand which applications and users are the biggest consumers of limited and expensive IT resources. This query shows the largest TSO user consumers running on z/OS systems—at both the CPC and individual z/OS system level. Knowing not only the amount of resource consumed, but also the frequency of TSO sessions, can help you to focus tuning efforts where potential payback is greatest.

Data Extracts

Note: The following SMFTUT query data extracts display the same Y-axis data elements as those elements displayed by the SMFBJT query. The X-axis, however, displays the top *nn* TSO user IDs instead of the top ten job names. For the data extract chart samples and axis values, see the SMFBJT query description.

[SMF Daily Top 10 TSO Users CPU CPCID](#)

[SMF Daily Top 10 TSO Users CPU SYSID](#)

[SMF Daily Top 10 TSO Users IO CPCID](#)

[VIEW 1 – SMF Daily Top 10 TSO Users I/O – Service Units CPCID](#)

[VIEW 2 – SMF Daily Top 10 TSO Users I/O – Connect Time CPCID](#)

[VIEW 3 – SMF Daily Top 10 TSO Users I/O – EXCPs CPCID](#)

[SMF Daily Top 10 TSO Users IO SYSID](#)

[VIEW 1 – SMF Daily Top 10 TSO Users I/O – Service Units SYSID](#)

[VIEW 2 – SMF Daily Top 10 TSO Users I/O – Connect Time SYSID](#)

[VIEW 3 – SMF Daily Top 10 TSO Users I/O – EXCPs SYSID](#)

[SMF Daily Top 10 TSO Users Memory CPCID](#)

[SMF Daily Top 10 TSO Users Memory SYSID](#)

Query Defaults

- HARCPU DETAIL timespan file: Cycle 01
- BAT_TS DETAIL timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART(ENDTS)
 - CPCID
 - CPCID = PUT(SYSID,\$CPCFMT.) ;
 - UIDRANK
 - PUT(COUNT,Z2.) || ' - ' || RACFUSID ;
 - where COUNT is the rank of resource use (e.g., 01 is largest resource user)
 - TOTCPU
 - SUM(PGMTCBTM,PGMSRBTM,PGMHIPTM,PGMRCTTM,PGMIOITM,
PGMSPNTM,PGMZPNTM,PGMISRTM,PGMITCTM) ;
 - INTVLCNT
 - INTVLCNT=1 ;
 - (uses existing data element PGMCOACT=1 in the initial input user code, then after summarization sets INTVLCNT=PGMCOACT to deal with Q&R default of treating Q&R derived elements as retained values in summarization).

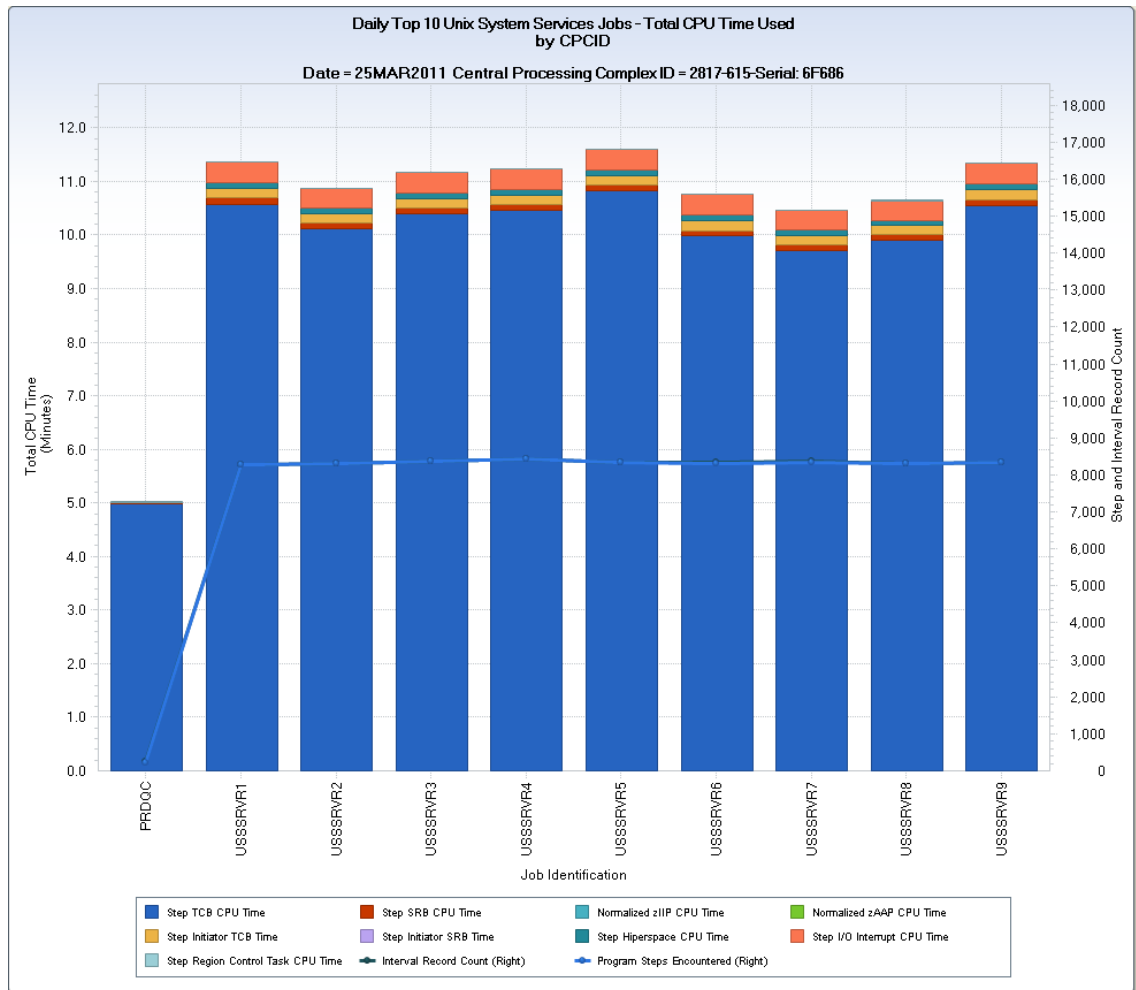
Modifications

The following modifications can be made to the SMFTUT query:

- You can add filter statements to the INPUT BAT_TS step to subset data to restrict analysis to jobs that begin specific characters, specific SYSIDs, and so on.
- You can adjust the TOP nn value to something other than the default of 10.
- You can generate charts for multiple days by reading in more than two cycles, but comment out the Select Yesterday user code in the Input BAT_TS step.
- To minimize SAS WORK space, this query uses a KEEP statement coded as a user code task in the INPUT BAT_TS step. If you want to chart data elements other than those elements in the query, you can either comment out the KEEP statement, or add the data elements you are interested in to the list of elements in the KEEP statement.

Note: This query charts two nonessential data elements: PGMITCTM and PGMISRTM (Initiator TCB and SRB time, respectively). If these data elements are not active in your step level files, the query fails to execute. To run the query, you can remove these two data elements from the ELEMENT SELECTION task in the two CPU time Data Extract steps. You can also ask your CA MICS administrator to activate these two elements in the step level files by changing the element definition in *sharedprefix*.MICS.GENLIB(SMFGENIN) from NAMX to NAME and then running the *sharedprefix*.MICS.CNTL(SMFCGEN) job. After these elements are activated, refresh the Q&R Meta Database.

SMFUJT – SMF Daily Top 10 Unix SysSrv Jobs



Overview

The SMFUJT query generates several chart sets. The chart sets show the daily top *nn* (default 10) UNIX System Services (USS) jobs, by job name, using CPU, I/O, and central storage (memory) resources on your z/OS systems. For each resource type, one chart set shows the top ten USS jobs using the resource over the entire Central Processing Complex (CPC), and another that shows the top ten USS jobs for each z/OS system. The query reads detail data from the BAT_OE (Open Edition/MVS Program) file, and summarizes the data by job name. The charted data reflects summarized interval and step record information from all job steps for the top ten USS jobs.

Note: While the chart samples displayed in this guide present the Top *nn* USS jobs alphabetically, from left to right on the x-AXIS, the query has been updated to present the Top *nn* USS jobs in order of descending resource use. The name of each USS job is prefixed with the resource usage rank. (for example, 01-jobnamea, 02- jobnameb, and so on.

For CPU, the CPU time used by each of the top ten USS jobs is displayed using a vertical bar. The bar is segmented by CPU time type (for example, TCB, SRB, I/O Interrupt, zIIP, zAAP, and so on). Two horizontal lines show the program count (number of unique program executions) and interval count (number of interval records) encountered for the USS job. For short running steps, the interval count and program count values have similar values. For long running programs, the interval count exceeds the program count.

Note: The program count data element (PGMCOUNT) is only set to one (1) for the last interval record for a long running program. For USS jobs with a long running program step that has not yet ended, PGMCOUNT is zero.

For I/O, I/O Service Units are used to determine the top ten USS jobs. Three different chart views are available to display I/O related resource utilization. The first view shows I/O service units for the top ten using vertical bars. The second view displays device connect time for the top ten by device class (for example, DASD, TAPE, Unit Record, and so on). The device connect time is displayed using a vertical bar, with the bar segmented by the time connected to each of the device classes. The final view shows EXCPs by device class, again using a segmented vertical bar. As with the CPU chart, the program and interval record count are shown using a horizontal line.

For central storage, the top ten USS jobs with the largest average working set size are displayed. Average working set size is the average number page frames in central storage used by the program. The page frame count is shown using a vertical bar and the program and interval record count are shown using a horizontal line.

Note: The average working set size represents the average for all step and interval records encountered for each top ten USS job.

The example shows the CPC level chart generated for USS jobs executing on an IBM z196 2817-615 CPC. The nine largest users of CPU time are the USSSRVRX jobs, which all have similar CPU consumption characteristics. The chart shows that for the day depicted, each of these USS jobs used about 11 minutes of CPU time. Most of this time was TCB time. These USS jobs appear to be frequently executed (about 8500 times each). The interval count and program count values are identical, meaning that they represent quick running, one-step USS job executions.

Uses

It is important to understand which applications are the biggest consumers of limited and expensive IT resources. This query shows the largest started task consumers running on z/OS systems—at both the CPC and individual z/OS system level. Knowing not only the amount of resource consumed, but also the frequency of program execution, can help you to focus tuning efforts where potential payback is greatest.

Data Extracts

Note: The following SMFUJT query data extracts display the same X- and Y-axis data elements as those elements displayed the SMFBJT query. The only difference is that this SMFUJT query shows the top *nn* UNIX System Services jobs, while the SMFBJT query output shows the top *nn* batch jobs. For the data extract chart samples and axis values, see the SMFBJT query description.

[SMF Daily Top 10 USS Jobs CPU CPCID](#)

[SMF Daily Top 10 USS Jobs CPU SYSID](#)

[SMF Daily Top 10 USS Jobs IO CPCID](#)

VIEW 1 – SMF Daily Top 10 USS Jobs I/O – Service Units CPCID

VIEW 2 – SMF Daily Top 10 USS Jobs I/O – Connect Time CPCID

VIEW 3 – SMF Daily Top 10 USS Jobs I/O – EXCPs CPCID

[SMF Daily Top 10 USS Jobs IO SYSID](#)

VIEW 1 – SMF Daily Top 10 USS Jobs I/O – Service Units SYSID

VIEW 2 – SMF Daily Top 10 USS Jobs I/O – Connect Time SYSID

VIEW 3 – SMF Daily Top 10 USS Jobs I/O – EXCPs SYSID

[SMF Daily Top 10 USS Jobs Memory CPCID](#)

[SMF Daily Top 10 USS Jobs Memory SYSID](#)

Query Defaults

- HARCPU DETAIL timespan file: Cycle 01
- BAT_OE DETAIL timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
DATEPART(ENDTS)
 - CPCID
CPCID = PUT(SYSID,\$CPCFMT.) ;
 - JOBRANK
PUT(COUNT,Z2.) || ' - ' || JOB ;
where COUNT is the rank of resource use (e.g., 01 is largest resource user)
 - TOTCPU
SUM(PGMTCBTM,PGMSRBTM,PGMHIPTM,PGMRCTTM,PGMIOITM,
PGMSPNTM,PGMZPNTM,PGMISRTM,PGMITCTM) ;
 - INTVLCNT
INTVLCNT=1 ;
(uses existing data element PGMCOACT=1 in the initial input user code, then after summarization sets INTVLCNT=PGMCOACT to deal with Q&R default of treating Q&R derived elements as retained values in summarization).

Modifications

The following modifications can be made to the SMFUJT query:

- You can add filter statements to the INPUT BAT_OE step to subset data to restrict analysis to jobs that begin specific characters, specific SYSIDs, and so on.
- You can adjust the TOP nn value to something other than the default of 10.
- You can generate charts for multiple days by reading in more than two cycles, but comment out the Select Yesterday user code in the Input BAT_OE step.
- To minimize SAS WORK space, this query uses a KEEP statement coded as a user code task in the INPUT BAT_OE step. If you want to chart data elements other than those elements in the query, you can either comment out the KEEP statement, or add the data elements you are interested in to the list of elements in the KEEP statement.

Note: This query charts two nonessential data elements: PGMITCTM and PGMISRTM (Initiator TCB and SRB time, respectively). If these data elements are not active in your step level files, the query fails to execute. To run the query, you can remove these two data elements from the ELEMENT SELECTION task in the two CPU time Data Extract steps. You can also ask your CA MICS administrator to activate these two elements in the step level files by changing the element definition in *sharedprefix*.MICS.GENLIB(SMFGENIN) from NAMX to NAME and then running the *sharedprefix*.MICS.CNTL(SMFCGEN) job. After these elements are activated, refresh the Q&R Meta Database.

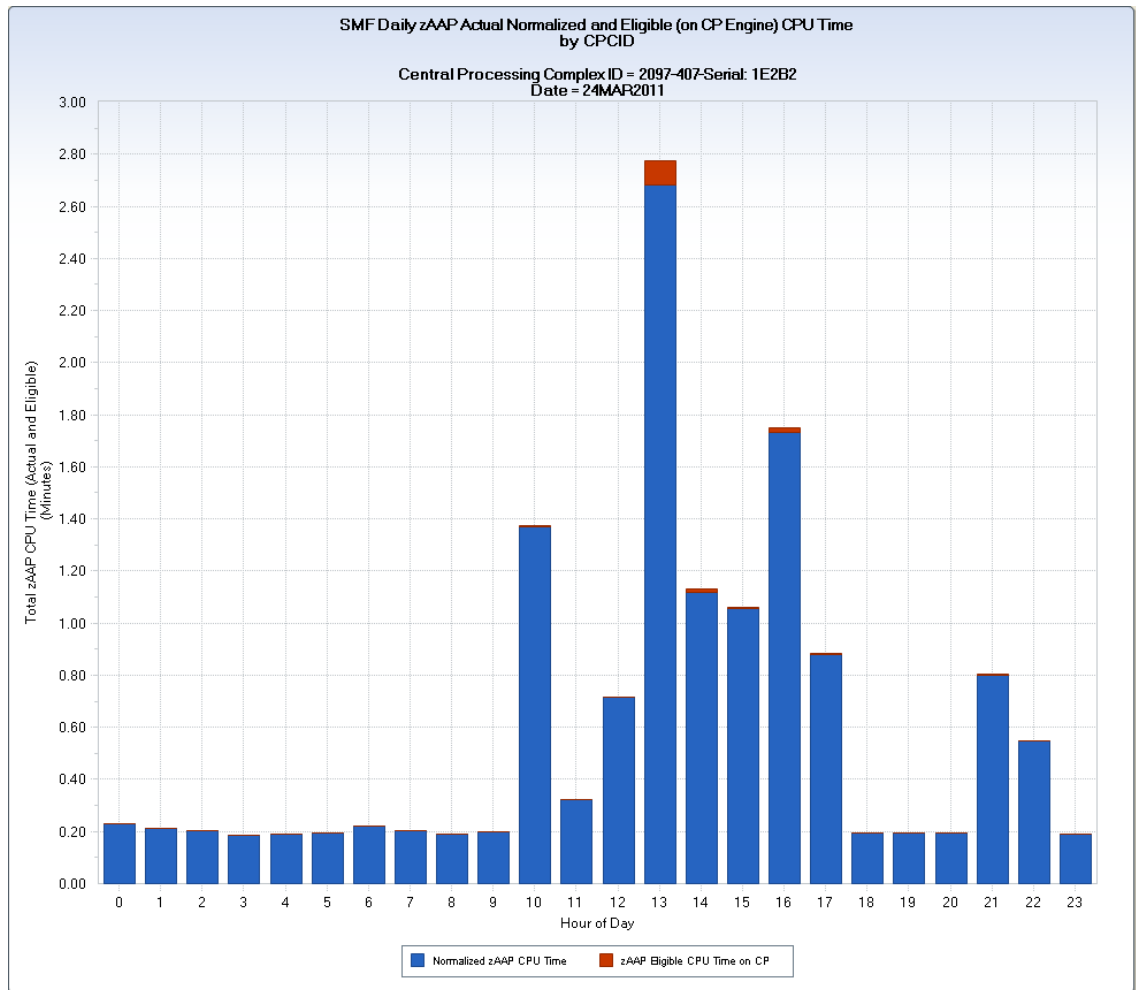
Specialty Engine zIIP/zAAP Focus

This section contains the following topics:

[SMFDZA - SMF Daily zAAP CPU Use and Demand](#) (see page 334)

[SMFDZI - SMF Daily zIIP CPU Use and Demand](#) (see page 339)

SMFDZA – SMF Daily zAAP CPU Use and Demand



Overview

The SMFDZA query charts the hourly actual zAAP engine use per day, and eligible use, for both the Central Processing Complex (CPC) and the z/OS LPARs. In addition, the query shows zAAP engine actual and eligible use by address space type (batch, TSO, started tasks, and so on).

The examples show CPC level zAAP engine demand for an IBM z10 2097-407.

The first chart shows the total hourly zAAP engine CPU time demand with actual zAAP engine use (blue) and eligible (red) using stacked vertical bars. The peak zAAP engine demand occurred during hour 13. At that hour, 2.68 hours of normalized zAAP engine CPU time were used. The normalization of zAAP CPU time is important for this CPC model because the CP engines run slower than the specialty (zIIP/zAAP) engines. When zIIP or zAAP CPU time is normalized, the value is expressed as the CP engine equivalent by multiplying by a normalization factor. For the 2097-407 CPC model, the normalization factor is 4.210938.

The second chart shows the actual zAAP CPU time use, broken down by address space type. All zAAP engine demand on the CPC shown came from started task address spaces. The second Q&R view for the address space level chart would show the breakdown for any eligible zAAP CPU time that executed on CP engines.

Uses

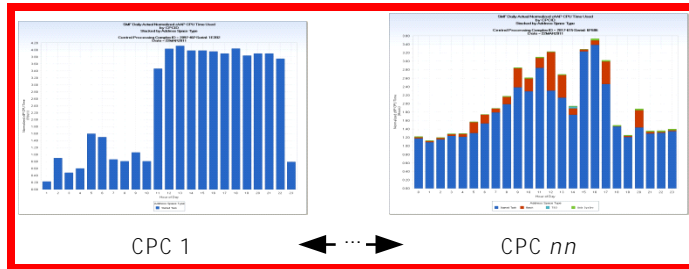
This query is used to determine if the zAAP engines in a CPC are sufficient to handle the zAAP eligible workload running on z/OS LPARs. The query also shows you what address space types in your workload mix execute zAAP engine eligible code. Even if no zAAP engines are present in a CPC, the charts show z/OS LPAR zAAP eligible work that executed on CP engines. zAAP engines are less expensive than CP engines. Reducing CP engines in a CPC can reduce software licensing fees. In general, if significant zAAP eligible work is executing on CP engines, it is cost effective to add zAAP engines to the CPC.

Note: If a CPC has one or more zIIP engines and no zAAP engines, and has specified ZAAPZIIP=YES in SYS1.PARMLIB, zIIP engines process all specialty engine work—including work that would normally run on zAAP engines. In this environment, all SMF and RMF metrics treat zAAP and zIIP eligible workloads as zIIP eligible. No zAAP actual or eligible demand appears.

Data Extracts

SMF Daily zAAP CPU Use and Demand Adrtyp

VIEW 1 – SMF Daily Actual Normalized zAAP CPU Time Used



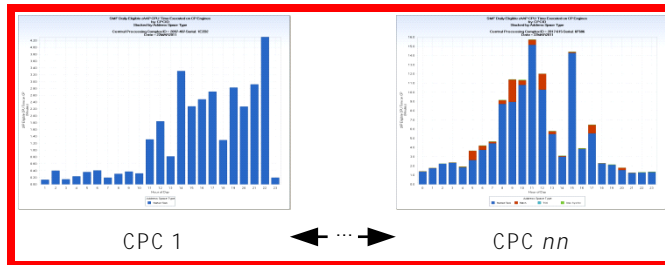
Left Y-axis

PGMZPNTM – Normalized zAAP CPU Time

X-axis

HOUR – Hour of Day

VIEW 2 – SMF Daily Eligible zAAP CPU Time Executed on CP Engines



Left Y-axis

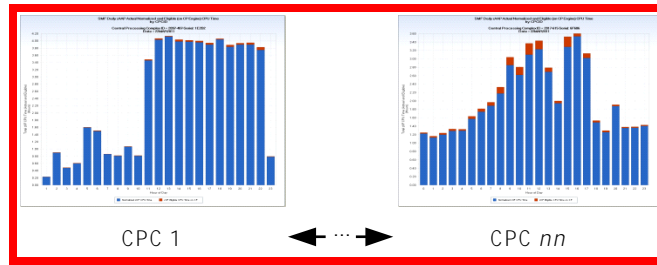
PGMZACTM – zAAP Eligible CPU Time on a CP

X-axis

HOUR – Hour of Day

Note: The two chart thumbnails for VIEW 2 are related to the thumbnails for VIEW 1. While the zAAP Eligible CPU Time on a CP values are much smaller than those values shown in VIEW 1, the vertical bar heights are similar because the Y-axis scale used for VIEW 1 shows hours of CPU time and VIEW 2 shows minutes.

SMF Daily zAAP CPU Use and Demand CPCID



Left Y-axis

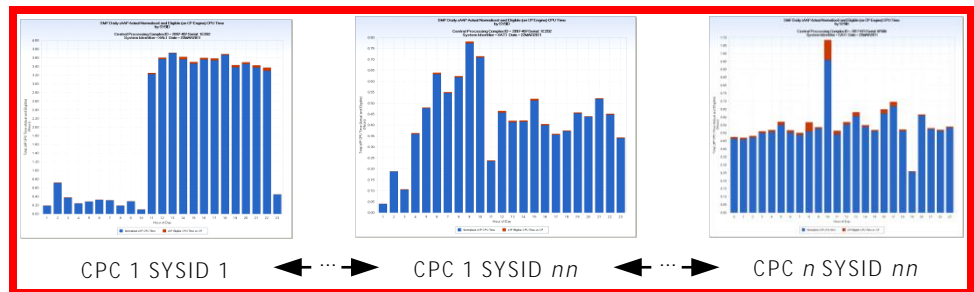
PGMZPNTM – Normalized zAAP CPU Time

PGMZACTM – zAAP Eligible CPU Time on a CP

X-axis

HOUR – Hour of Day

SMF Daily zAAP CPU Use and Demand SYSID



Query Defaults

- HARCPU DETAIL timespan file: Cycles 01
- BATPGM, BAT_OE, BAT_SA, BAT_ST, BAT_TS, and BAT_TP DETAIL timespan files: Cycles 01-02
- Filters: PGMZAPTM GT 0 OR PGMZACTM GT 0 (only zAAP actual or eligible CPU time)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

CPCID

```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

DATE

```
DATEPART(ENDTS) ;
```

ADDRTYP

```
IF PGMTYPE=1 THEN ADDRTP='Batch' ;
ELSE IF PGMTYPE=2 THEN ADDRTP='TSO' ;
ELSE IF PGMTYPE=3 THEN ADDRTP='Started Task' ;
ELSE IF PGMTYPE=4 THEN ADDRTP='APPC' ;
ELSE IF PGMTYPE=5 THEN ADDRTP='Unix SysSrv' ;
ELSE IF PGMTYPE=6 THEN ADDRTP='System Addr' ;
ELSE ADDRTP= 'UNKNOWN' ;
```

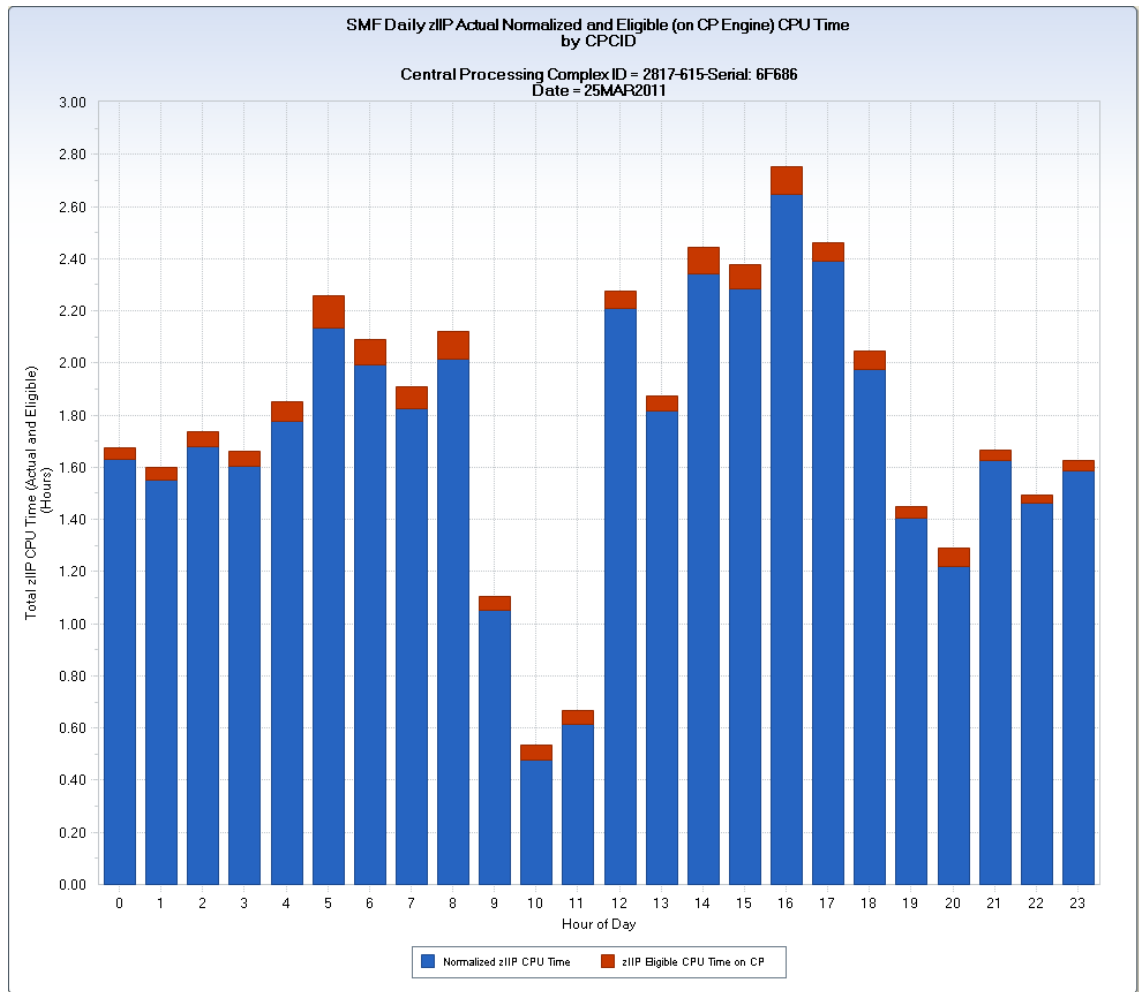
Modifications

The following modifications can be made to the SMFDZA query:

- You can increase the number step level files input, but comment out the user code in the FILE SELECTION < combine step that selects yesterday's data.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1).
- You can add filter statements to select a single CPC, SYSID, or address space type (using the PGMTYPE data element (see ADDRTP derivation).

Note: The query step shows a *red X* next to the Key Selection task of the Concatenation step. You can ignore the validation error—keys are not required because the six temporary files are not interleaved.

SMFDZI – SMF Daily zIIP CPU Use and Demand



Overview

The SMFDZI query charts the hourly actual zIIP engine use per day, and eligible use, for both the Central Processing Complex (CPC) and the z/OS LPARs. In addition, the query shows zIIP engine actual and eligible use by address space type (batch, TSO, started tasks, and so on).

The examples show CPC level zIIP engine demand for an IBM z196 2817-615.

The first chart shows the total hourly zIIP engine CPU time demand with actual zIIP engine use (blue) and eligible (red) using stacked vertical bars. The peak zIIP engine demand occurred during hour 16. At that hour, 2.65 hours of normalized zIIP engine CPU time were used, and 0.11 hours of CP engine time was used to execute zIIP eligible work. The normalization of zIIP CPU time is important for this CPC model because the CP engines run slower than the specialty (zIIP/zIIP) engines. When zIIP or zIIP CPU time is normalized, the value is expressed as the CP engine equivalent by multiplying by a normalization factor. For the 2817-615 CPC model, the normalization factor is 1.52734.

The second chart shows the actual zIIP CPU time use, broken down by address space type. Most zIIP engine demand on the CPC shown came from started task address spaces (dark blue), with TSO sessions (cyan), batch (red), and USS workloads (green) accounting for the additional demand. The second Q&R view for the address space type chart would show the breakdown for the eligible zIIP CPU time that executed on CP engines.

Uses

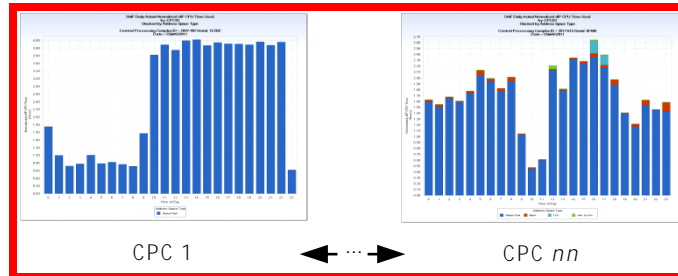
This query is used to determine if the zIIP engines in a CPC are sufficient to handle the zIIP eligible workload running on z/OS LPARs. The query also shows you what address space types in your workload mix execute zIIP engine eligible code. Even if no zIIP engines are present in a CPC, the charts show z/OS LPAR zIIP eligible work that executed on CP engines. zIIP engines are less expensive than CP engines. Reducing CP engines in a CPC can reduce software licensing fees. In general, if significant zIIP eligible work is executing on CP engines, it is cost effective to add zIIP engines to the CPC.

Note: The sample charts were generated for a CPC that is configured with several zIIP engines, no zAAP engines, and has ZIIPZIIIP=YES specified in SYS1.PARMLIB. With this configuration, zIIP engines process all specialty engine work—including work that would normally run on zAAP engines. In this environment, the operating system treats both zIIP and zAAP eligible workloads as zIIP eligible. No zAAP actual or eligible information is recorded in SMF and RMF workload activity records. Therefore, some portion of the zIIP actual and eligible demand shown in the sample charts may be associated with zAAP eligible workloads.

Data Extracts

SMF Daily zIIP CPU Use and Demand Adrtyp

VIEW 1 – SMF Daily Actual Normalized zIIP CPU Time Used



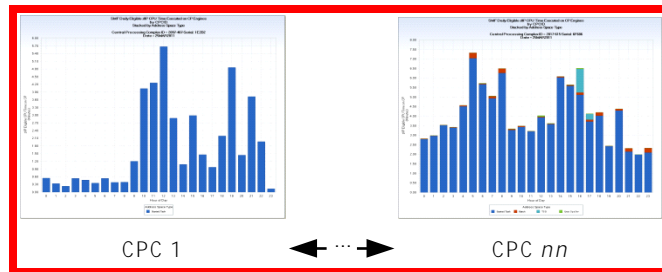
Left Y-axis

PGMSPNTM – Normalized zIIP CPU Time

X-axis

HOUR – Hour of Day

VIEW 2 – SMF Daily Eligible zIIP CPU Time Executed on CP Engines



Left Y-axis

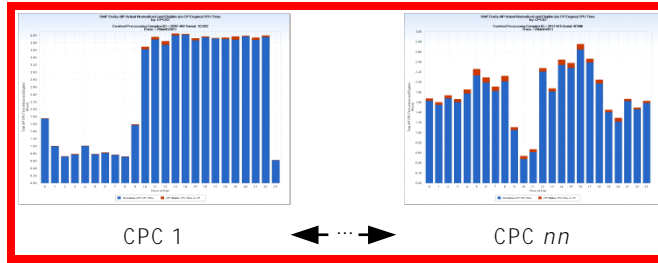
PGMSUCTM – zIIP Eligible CPU Time on a CP

X-axis

HOUR – Hour of Day

Note: The two chart thumbnails for VIEW 2 are associated with the thumbnails for VIEW 1. While the zIIP Eligible CPU Time on a CP values are much smaller than those values shown in VIEW 1, the vertical bar heights are similar because the Y-axis scale used for VIEW 1 shows hours of CPU time and VIEW 2 shows minutes.

SMF Daily zIIP CPU Use and Demand CPCID



Left Y-axis

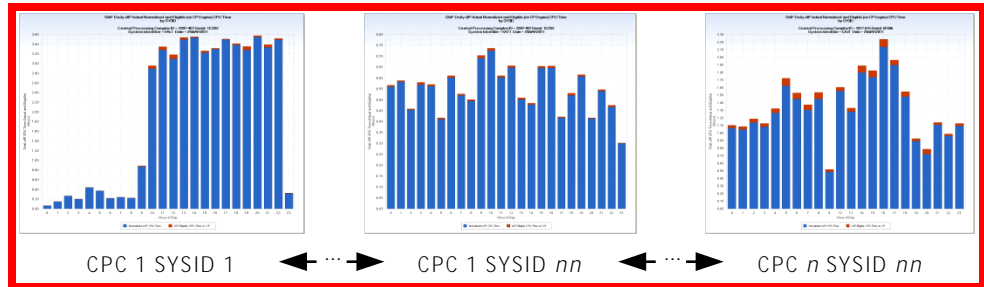
PGMSPNTM – Normalized zIIP CPU Time

PGMSUCTM – zIIP Eligible CPU Time on a CP

X-axis

HOUR – Hour of Day

SMF Daily zIIP CPU Use and Demand SYSID



Query Defaults

- HARCPU DETAIL timespan file: Cycles 01
- BATPGM, BAT_OE, BAT_SA, BAT_ST, BAT_TS, and BAT_TP DETAIL timespan files: Cycles 01-02
- Filters: PGMSUPTM GT 0 OR PGMSUCTM GT 0 (only zIIP actual or eligible CPU time)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:

CPCID

```
TRIM(LPCMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5);
```

DATE

```
DATEPART(ENDTS) ;
```

ADDRTYP

```
IF PGMTYPE=1 THEN ADDRTP='Batch' ;
ELSE IF PGMTYPE=2 THEN ADDRTP='TSO' ;
ELSE IF PGMTYPE=3 THEN ADDRTP='Started Task' ;
ELSE IF PGMTYPE=4 THEN ADDRTP='APPC' ;
ELSE IF PGMTYPE=5 THEN ADDRTP='Unix SysSrv' ;
ELSE IF PGMTYPE=6 THEN ADDRTP='System Addr' ;
ELSE ADDRTP= 'UNKNOWN' ;
```

Modifications

The following modifications can be made to the SMFDZI query:

- You can increase the number step level files input, but comment out the user code in the FILE SELECTION < combine step that selects yesterday's data.
- You can use a filter statement to restrict data to focus on a particular time range (for example, HOUR GE 8 AND HOUR LE 17) or particular time zone (for example, ZONE EQ 1).
- You can add filter statements to select a single CPC, SYSID, or address space type (using the PGMTYPE data element (see ADDRTP derivation).

Note: The query step shows a *red X* next to the Key Selection task of the Concatenation step. You can ignore the validation error—keys are not required because the six temporary files are not interleaved.

Chapter 8: CA MICS Tape Analyzer Option (TAP) Queries

This section contains the following topics:

[IBM TS77xx Virtual Tape Server Queries](#) (see page 345)

[StorageTek VSM Queries](#) (see page 358)

IBM TS77xx Virtual Tape Server Queries

This section contains the following topics:

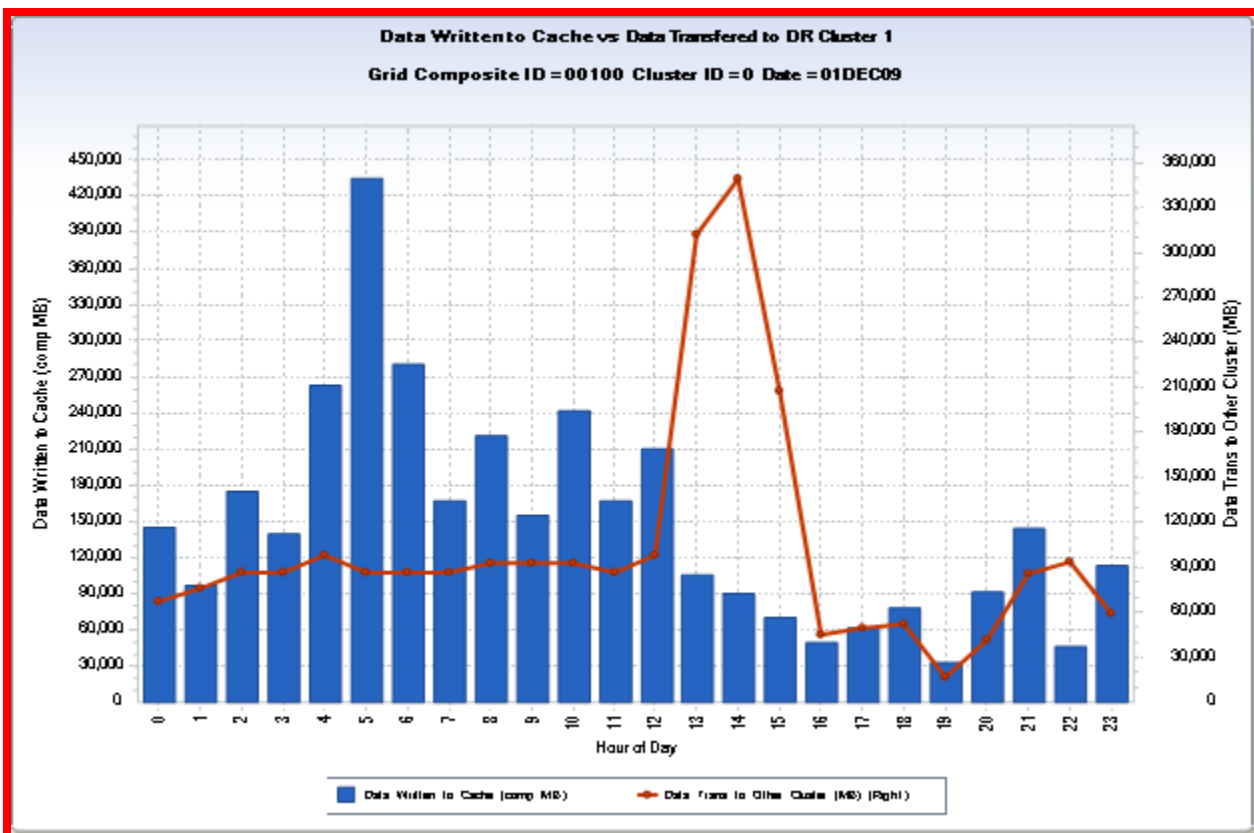
[TS7CDC - IBM TS77xx Daily Cluster Data Copy Activity](#) (see page 346)

[TS7HBA - IBM TS77xx Daily Host Channel I/O and HBA Compression Analysis](#) (see page 348)

[TS7VMA - IBM TS77xx Daily Virtual Tape Mount Activity](#) (see page 353)

[TS7VRA - IBM TS77xx Daily Virtual Recall Mount Analysis](#) (see page 356)

TS7CDC – IBM TS77xx Daily Cluster Data Copy Activity



Overview

The TS7CDT query generates a daily chart for Cluster 0 of each IBM TS7700 Grid. The chart shows, for each hour, the amount of data written by workloads to cache, and the amount of data copied to the high availability Cluster 1. The chart shows megabytes written to cache using a vertical bar and the data copied to Cluster 1 using a horizontal line. Both values represent data that was compressed by the Host Bus Adaptors (HBAs).

The example shows the chart generated for Grid 00100, Cluster 0 of an IBM TS7700. For hours 0 through 12, more data was written to the cache than was copied to Cluster 1. As the incoming data rate decreased, in hours 13 through 23, the data copy volume exceeded that written to cache.

Uses

This query is useful to see patterns where the copy activity cannot keep up with new data being written to cache. If copying is significantly delayed, performance can be impacted. If you detect periods where very high volumes of data requiring copy are overwhelming the copy function, you can shift some workloads to other time periods with lower incoming data volumes.

Data Extracts

IBM TS77xx Daily Cluster Data Copy



Left Y-axis

VDAVDW - Data Written to Cache (comp MB)

Right Y-axis

GCCDTR – Data Trans to Other Cluster (MB)

X-axis

Hour of Day

Query Defaults

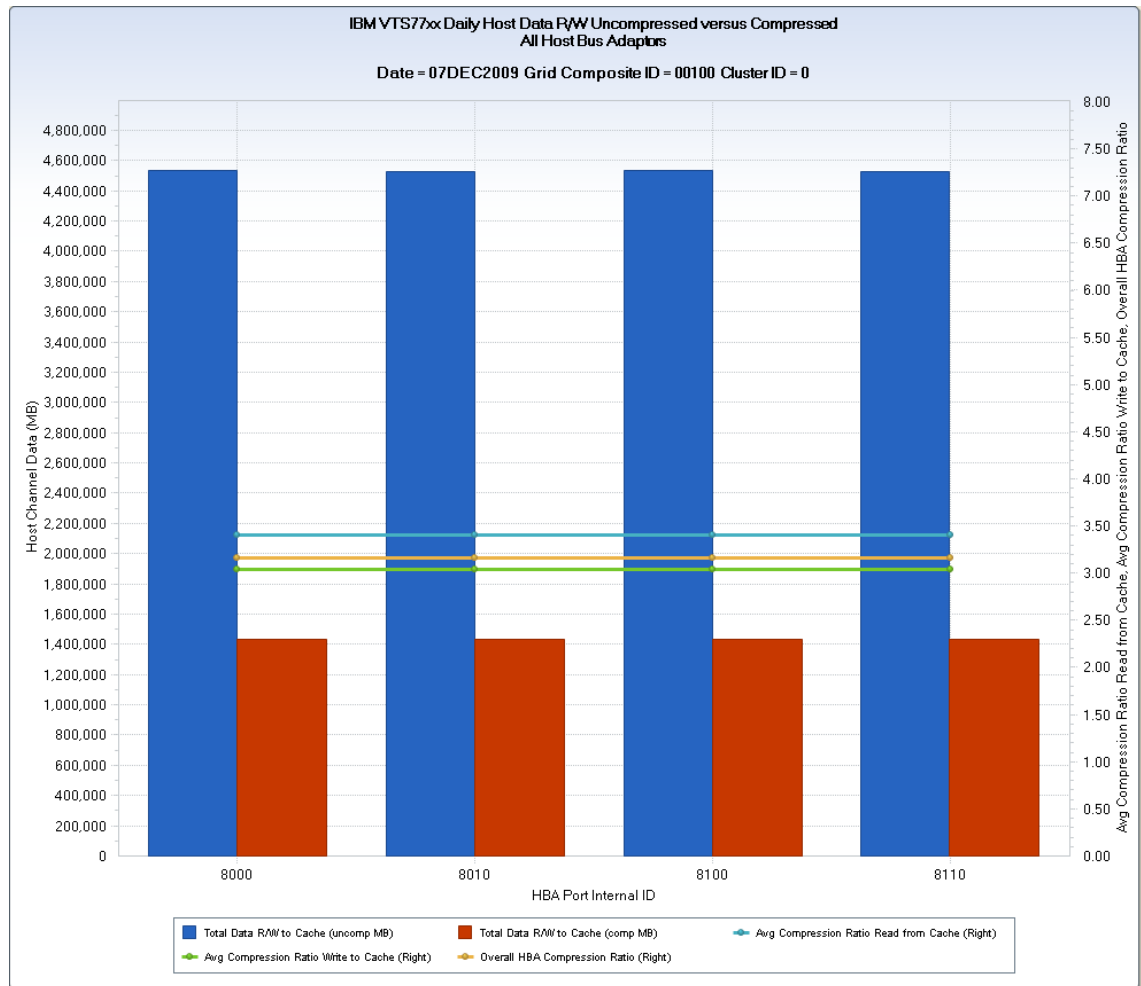
- HVTVDA DAYS timespan file: Cycles 01-02
 - HVTVDA file filtered to select CLUSTER 0 only
- HVTGCC DAYS timespan file: Cycles 01-02
 - HVTGCC file filtered to select only CLUSTER 0 and Distributed Cluster 1
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
 - Selection code present for both VDA and GCC File Selection steps
- Derivations:
 - DATE
 - DATEPART (ENDTS)

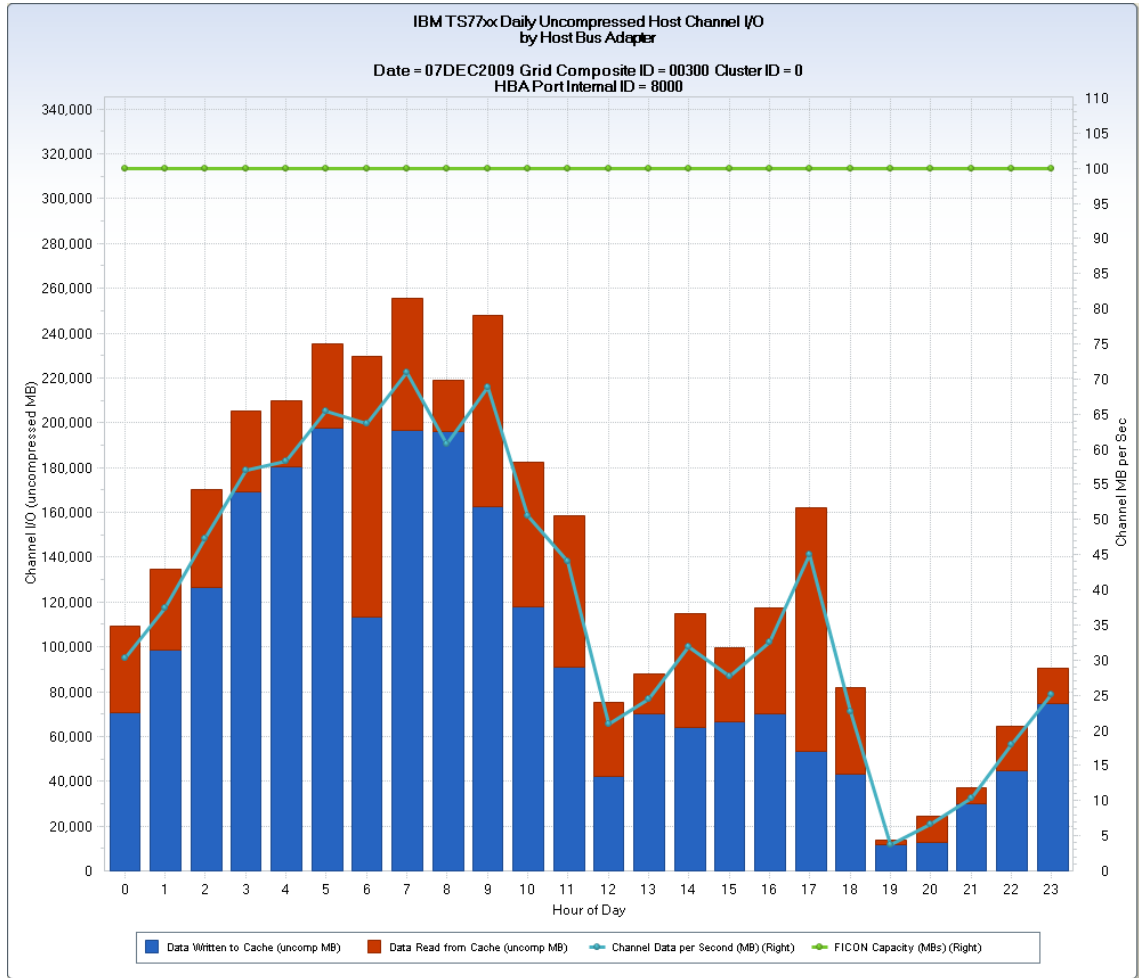
Modifications

The following modifications can be made to the TS7CDC query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in both the VDA and GCC File Selection steps.
- You can add Data Filters to the VDA and GCC File Selection steps to select a particular Grid.

TS7HBA – IBM TS77xx Daily Host Channel I/O and HBA Compression Analysis





Overview

The TS7HBA query generates two daily charts for Cluster 0 of each TS77 XX grid. The charts show host channel I/O data volume, data rates, and Host Bus Adaptors (HBA) data compression ratios achieved for data written to the TS77 XX cache.

A detail chart produces a daily chart for each HBA. The chart shows the hourly host to HBA FICON channel data volume (uncompressed MB) written to and read from the cache using stacked vertical bars. The hourly MB/sec HBA throughput rate and FICON channel MB/sec capacity threshold (hardcoded at 100 MB/sec) are displayed using horizontal lines. The summary chart displays the daily total MB of data volume processed by each HBA. The data volume values are shown both as uncompressed and compressed MB of data using vertical bars. The read, write, and overall HBA compression ratios are displayed using horizontal lines.

The top example shows the summary chart for Grid 00100, Cluster 0 of an IBM TS7700. Four active HBAs and the data processing load is evenly distributed across the HBAs. For the day shown, the compression ratio achieved for data read from cache by host applications was 3.4 to 1—better than the compression ratio achieved for new data written to cache at 3.04 to 1. Data read from cache is uncompressed by the HBAs, so the read compression ratio actually reflects the compression ratio achieved when the data was earlier written to cache.

The bottom example shows the detailed hourly activity for HBA 8000 in the same grid and cluster. The vertical bars show that for most hours of the day more data is written to cache than read. Hour 10 was the hour with the highest data volume. The FICON channel data rate for that hour averaged 93 MB/sec—close to the maximum data rate of 100 MB/sec.

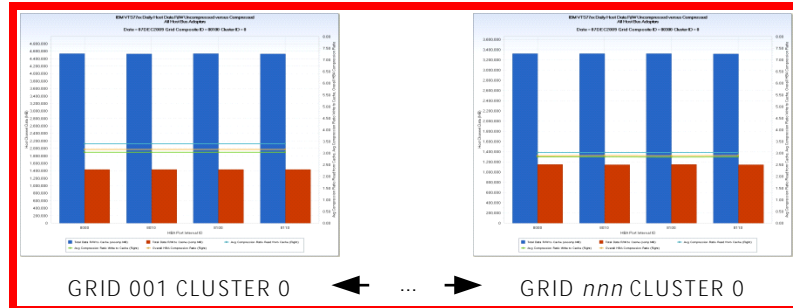
Investigate either end of the channel to avoid possible IO throughput issues during peak processing. In addition, the report shows the amount of compressed I/O data on the cache side of the HBA, which is a visual view of the data compression achieved by the HBA. In this case, it looks to be between a 3:1 and 4:1 ratio. The green reference line on the right axis is the megabytes per second capacity of a FICON channel, 100 MBs. The blue line indicates that for the overall day, in this case, the overall megabytes per second processed averaged 50 MBs. To view the hourly processing rates, and identify the peak hours, review the Channel IO to HBA Interfaces charts in this same query.

Uses

This query is useful for monitoring the capacity limitations for the hardware interfaces between the z/OS host and the IBM TS77 XX cache. The FICON channels have data rate limitations. If the data rate routinely approaches the channel throughput limit, you can install additional HBAs (up to four), or install an additional TS77 XX grid. Uneven load balancing across the HBAs may indicate a problem. The compression ratio is useful in determining the real virtual volume storage capacity of the TS77 XX cache.

Data Extracts

IBM TS77xx Daily HBA Compression Summary



Left Y-axis

VAPCRW – Total Data R/W to Cache (uncomp MB)

VAPVRW- Total Data R/W to Cache (comp MB)

Right Y-axis

VAPAVCRC – Avg Compression Ratio Read from Cache

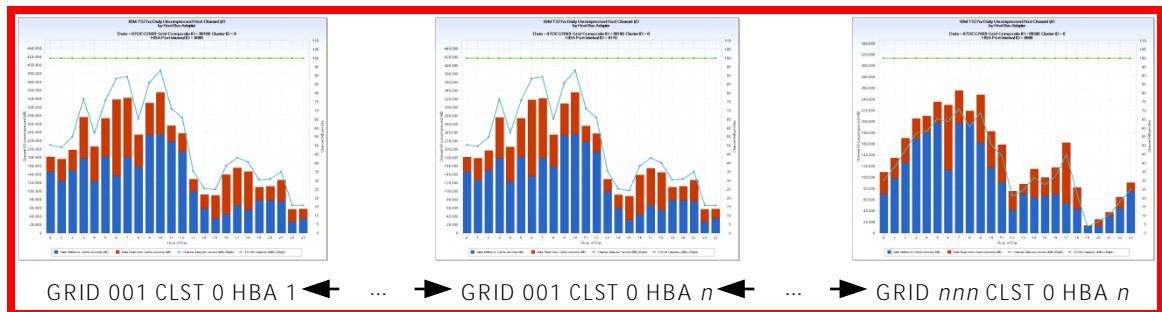
VAPAVCWC – Avg Compression Ratio Write to Cache

TOTRATIO – Overall HBA Compression Ratio

X-axis

HVTHBAID – HBA Port Internal ID (Displayed with \$HEX4. FORMAT)

IBM TS77xx Daily Host Channel HBA I/O Detail



Left Y-axis

VAPCW – Data Written to Cache (uncomp MB)

VAPVCR – Data Read from Cache (uncomp MB)

Right Y-axis

VAPPSCRW – Channel Data per Second (MB)

FICONMBS – FICON Capacity (MB/sec)

X-axis

HOUR – Hour of Day

Query Defaults

- HVTVAP DAYS timespan file: Cycles 01-02
 - HVTVAP file filtered to select CLUSTER 0 only
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;

- Derivations:

DATE

DATEPART (ENDTS)

FICONMBS

100

TOTRATIO

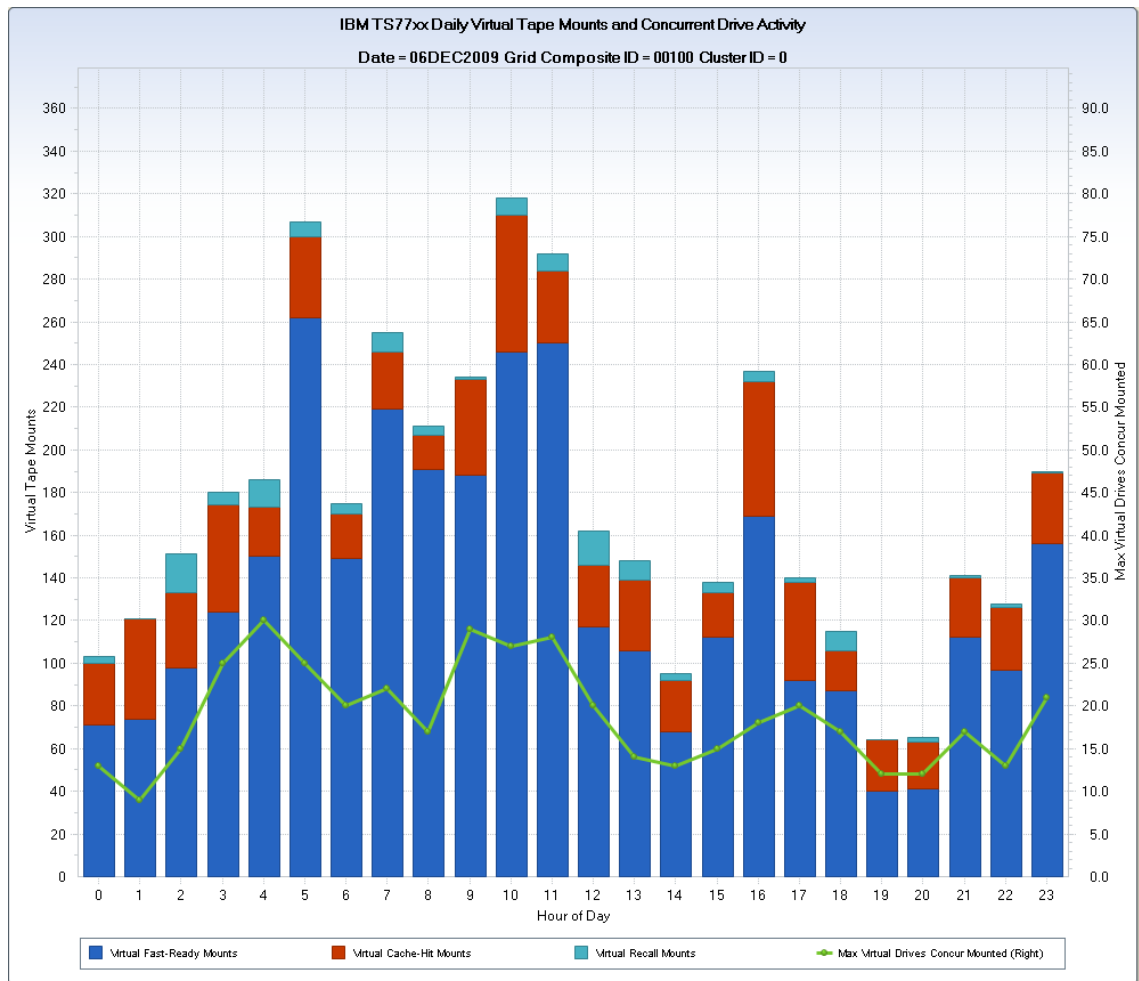
```
IF VAPVRW THEN TOTRATIO=ROUND( (VAPCRW/VAPVRW) ,0.01) ;  
ELSE TOTRATIO=0 ;
```

Modifications

The following modifications can be made to the TS7HBA query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in the VAP File Selection step.
- You can add Data Filters to the VAP File Selection step to select a particular Grid or Cluster.

TS7VMA – IBM TS77xx Daily Virtual Tape Mount Activity



Overview

The TS7VMA query generates a daily chart for each IBM TS7700 Grid and Cluster combination. The chart shows the hourly virtual tape mount activity and maximum concurrent virtual tape drive use. The number of virtual tape mounts are shown, for each hour, using a stacked vertical bar composed of fast ready (scratch), cache hit, and recall (cache miss) mounts. The maximum number of concurrently mounted virtual tape drives for each hour is charted using a horizontal line.

The example shows a daily chart generated for an IBM TS7700 Grid 00100 and Cluster 0. Hour 10 was the peak hour for virtual tape mounts. The maximum number of concurrently mounted tape drives was 30. This number occurred at hour 4. The maximum number of recall (cache miss) mounts was 18, and occurred at hour 2.

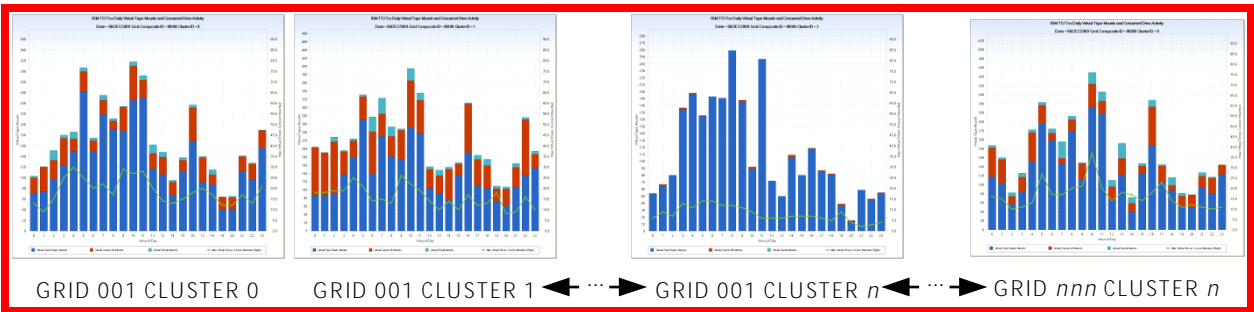
Uses

This query provides an excellent overview of the hourly usage patterns for your TS77XX grid clusters. With virtual tape drives, fast ready mount (scratch tape) requests are satisfied almost instantly. When a workload requests a specific virtual tape volume, the objective is to have the volume resident in the cache where the time to mount is fast. Recall mounts, on the other hand, result in significant delays for the requesting workload. The virtual tape must be accessed in the physical tape repository and copied into cache before the mount request can be satisfied. If significant numbers of recall mounts occur, particularly in recognizable patterns, review your archiving policies.

Another key metric to watch is the maximum number of concurrently mounted virtual tape drives. If this metric reaches the number of defined virtual tape drives, generally 128 or 256, workloads requesting mounts are delayed. If this delay occurs during particular hours, shifting some workloads to a lower use time period may solve the problem.

Data Extracts

IBM TS77xx Daily Virt Tape Mount Activity



Left Y-axis

HHCFRM – Virtual Fast Ready Mounts

HCCCHM – Virtual Cache-Hit Mounts

HCCCMM – Virtual Recall Mounts

Right Y-axis

VDAMXVCM – Max Virtual Drives Concur Mounted

X-axis

Hour of Day

Query Defaults

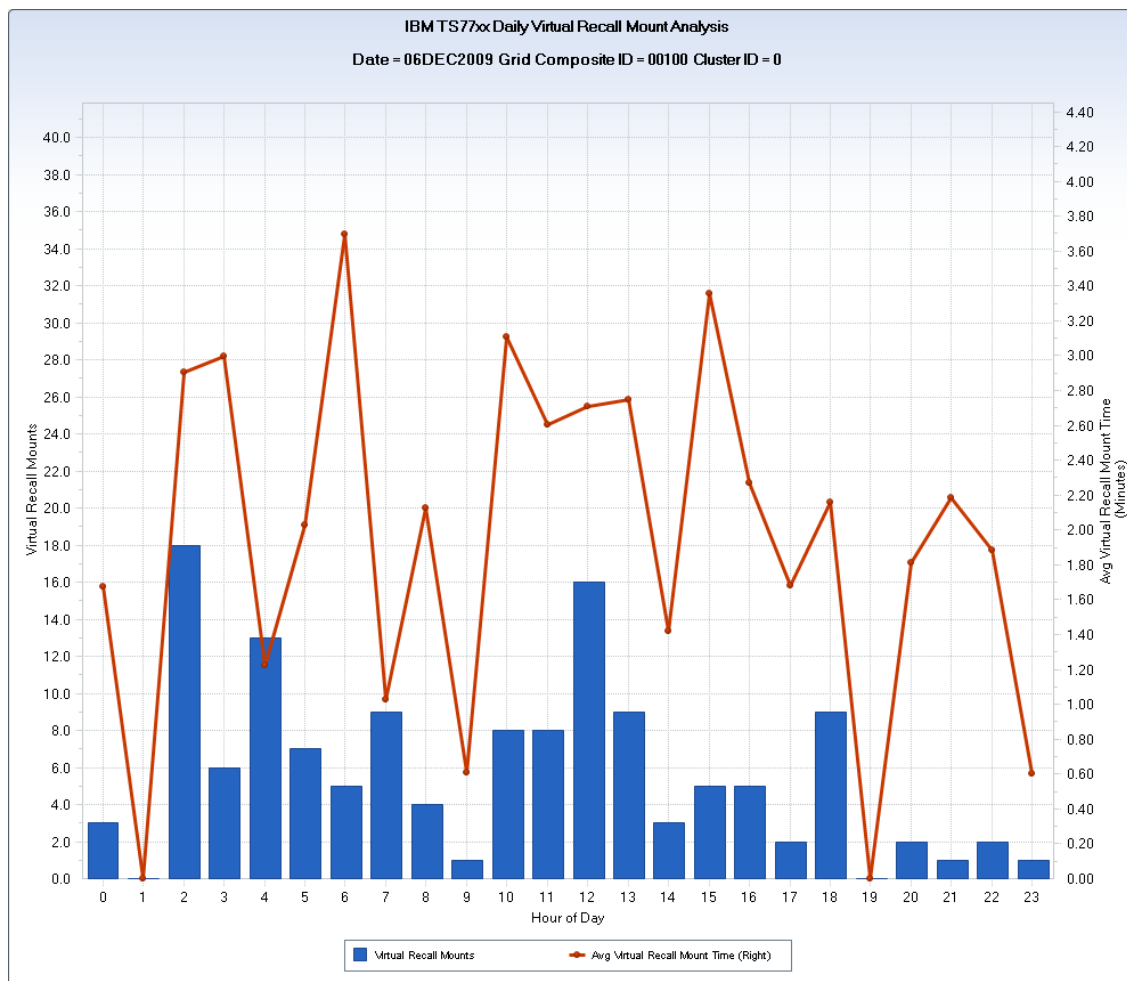
- HVTHHC DAYS timespan file: Cycles 01-02
- HVTVDA DAYS timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
 - Selection code present for both HCC and VDA File Selection steps
- Derivations:
 - DATE
 - DATEPART (ENDTS)

Modifications

The following modifications can be made to the TSAVMA query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in both the HCC and VDA File Selection step.
- You can add Data Filters to the HCC and VDA File Selection step to select particular Grids or Clusters.

TS7VRA – IBM TS77xx Daily Virtual Recall Mount Analysis



Overview

The TS7VRA query generates a daily chart for each IBM TS7700 Cluster. The chart shows the hourly number of, and average time to satisfy, recall (cache miss) mounts.

The example shows the chart generated for Grid 00100, Cluster 0 of an IBM TS7700. The average time to satisfy recall mount requests ranges from 0.60 minutes in hour 23, to 3.70 minutes in hour 6. The maximum number of recall mounts was 18 in hour 2.

Uses

This query is useful for observing when recall mounts occur and to see the impact of virtual volume recalls on workload throughput. When a requested volume is in cache, the time to satisfy the mount is less than 1 second. A cache miss means that the requesting workload must wait for the virtual volume to be recalled from the physical cartridges backing the tape server and copied to cache before the mount request can be satisfied. While recalls cannot be eliminated, minimize them. If excessive recalls are observed, the storage administrator can determine the characteristics of the volumes being recalled and make configuration changes to try to increase the chance that virtual volumes are in cache when requested.

Data Extracts

IBM TS77xx Daily Recall Mount Analysis



Left Y-axis

HHCCMM – Virtual Recall Mounts

Right Y-axis

HHCAVRTM - Avg Virtual Recall Mount Time

X-axis

Hour of Day

Query Defaults

- HVTHHC DAYS timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
DATE
DATEPART (ENDTS)

Modifications

The following modifications can be made to the TS7VRA query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in the initial HVTHCC File Selection step.
- You can add Data Filters to the HVTHCC File Selection step to select particular Grids or Cluster. For example, you only want to report on the host attached Cluster 0.

StorageTek VSM Queries

This section contains the following topics:

[STKCDA - STK VSM Daily VTV Mounts and Concurrent Drive Use](#) (see page 359)

[STKCTA - STK VSM Daily Channel Throughput Analysis](#) (see page 367)

[STKDBU - STK VSM Daily Disk Buffer Use](#) (see page 371)

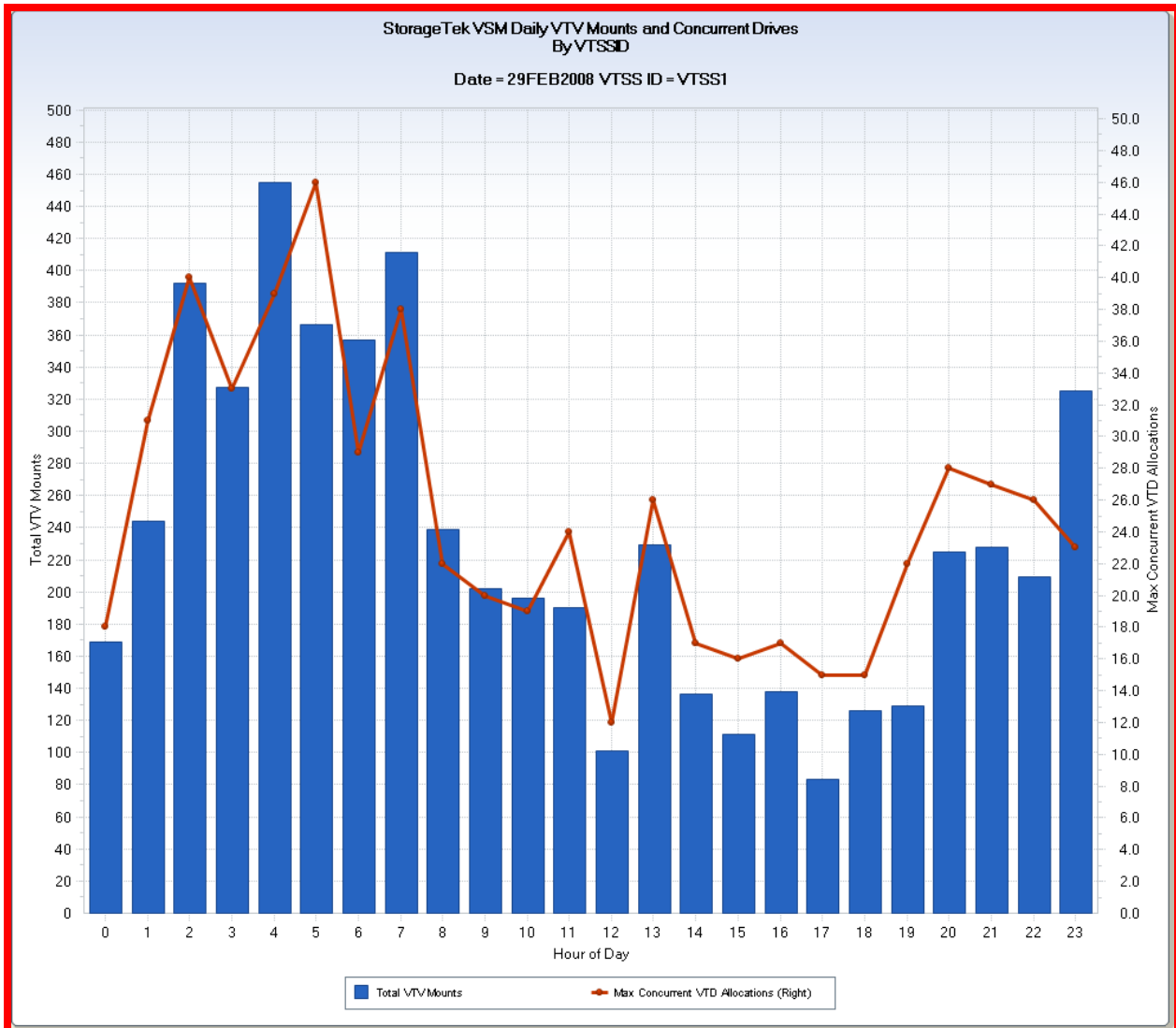
[STKIOT - STK VSM Daily I/O Throughput](#) (see page 374)

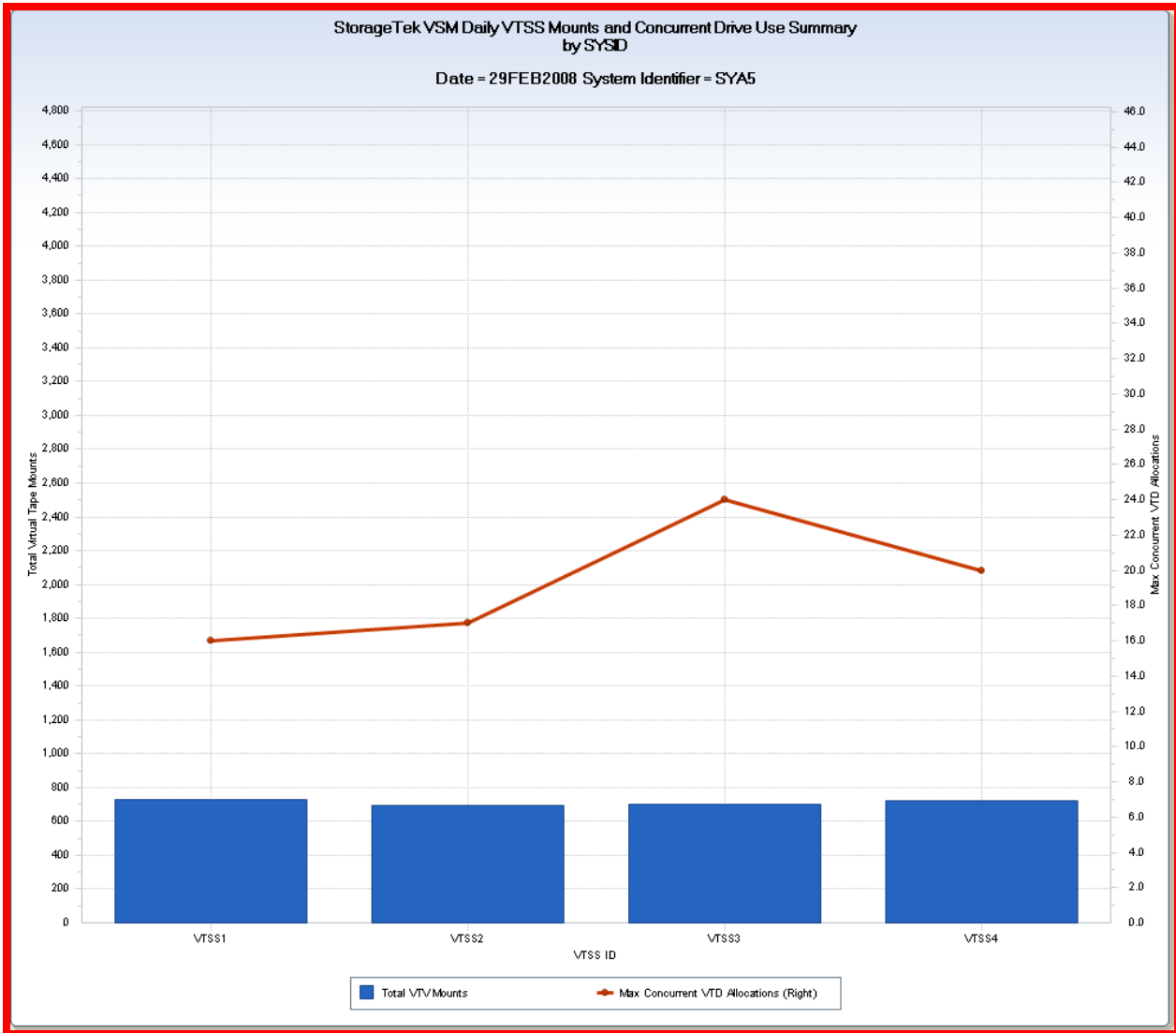
[STKRMD - STK VSM Daily Recall Mount Time Distribution](#) (see page 378)

[STKVMA - STK VSM Daily Virtual Tape Mount Activity](#) (see page 381)

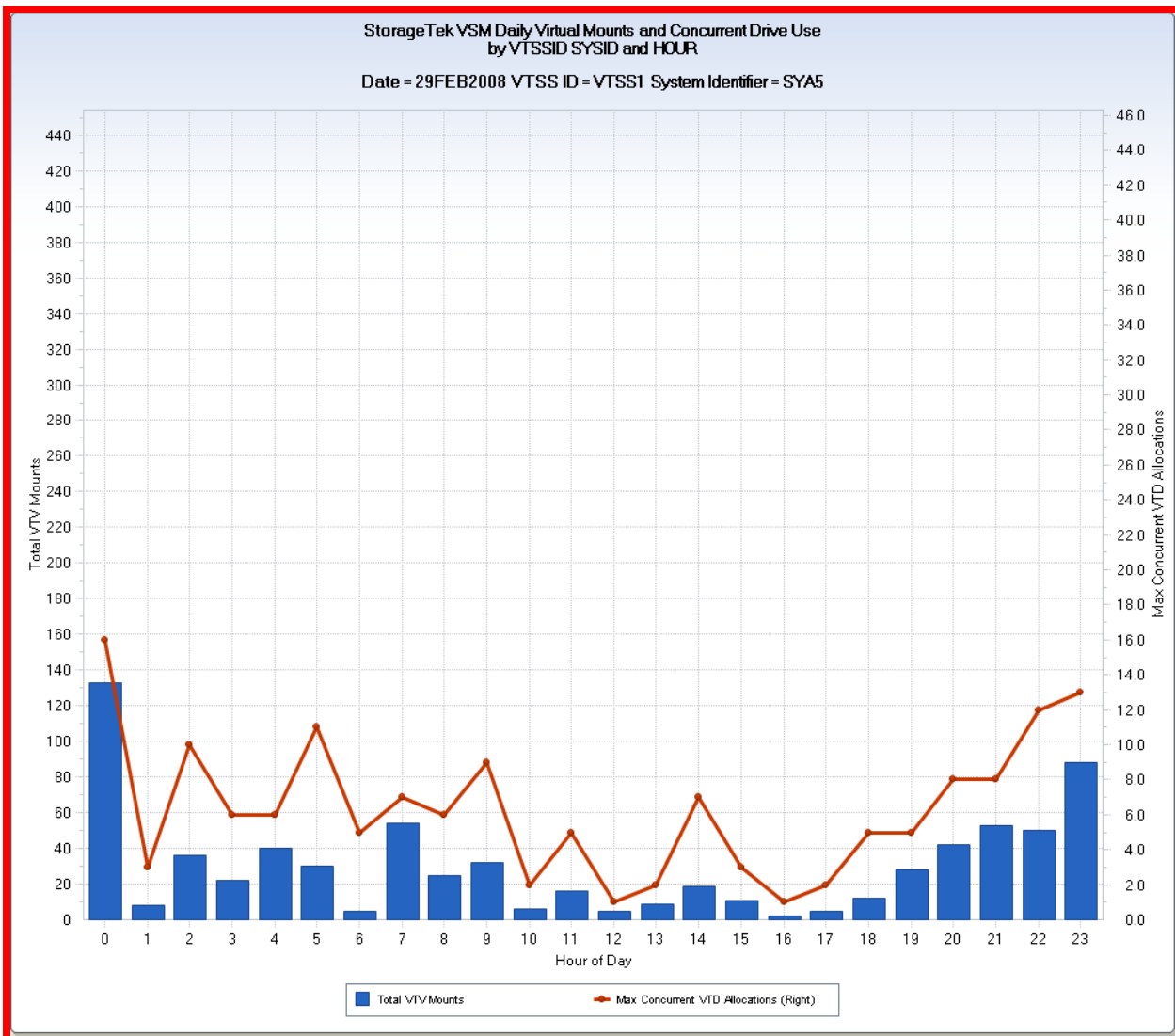
[STKVMD - STK VSM Daily VTV Mount Time Distribution](#) (see page 385)

STKCDA – STK VSM Daily VTV Mounts and Concurrent Drive Use

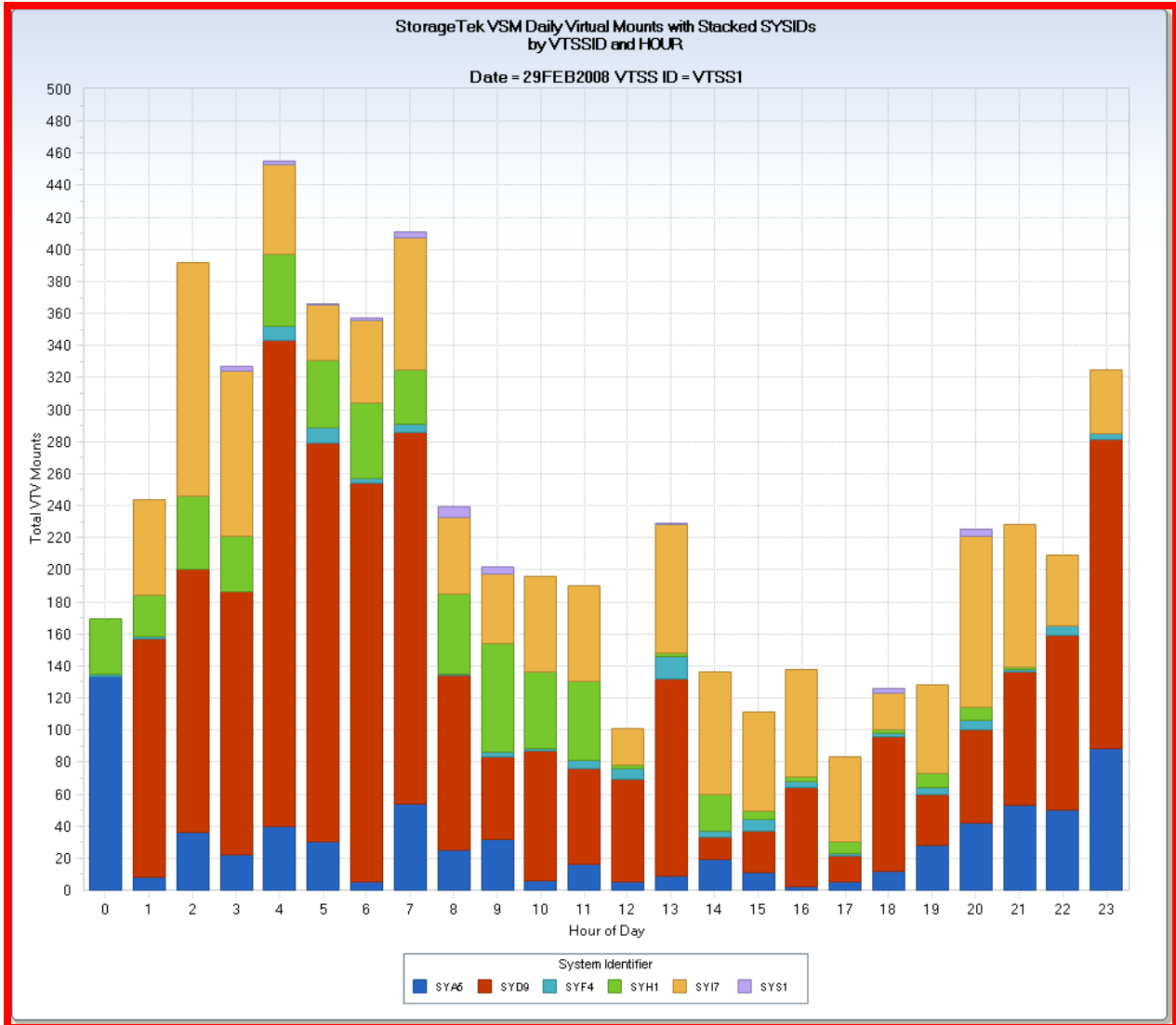




View 1 – Detailed VTSS by SYSID View



View 2 – VTSS with stacked SYSIDs View



Overview

The STKCDA query produces three data extracts, one with two separate views. The views show daily StorageTek Virtual Tape Storage Subsystem (VTSS) virtual tape volume mounts and concurrent use of virtual tape drives. The three data extracts display the daily use statistics at different levels of granularity and from different perspectives (VTSS views and SYSID views).

The first chart shows a VTSS level daily chart showing total Virtual Tape Volume (VTV) mounts and maximum concurrently mounted VTV drives, per hour, for all SYSIDs using the VTSSID. The example shows the statistics for VTSSID "VTSS1". Hour 3 had the most VTV mounts (455) and hour 4 had the greatest number of concurrently mounted VTV tape drives (46).

The second chart shows a SYSID level view with daily totals for each VTSSID accessed. A single vertical bar chart shows the total VTV mounts, and a horizontal line shows the maximum number of concurrently mounted VTV drives for each VTSSID. The example shows the totals for SYSID "SYA5". The use by the system of four VTSSs was evenly distributed. The greatest number of VTV mounts was on VTSSID "VTSS1" (731), and the maximum number of concurrently mounted VTV drives was 24 on VTSSID "VTSS3."

The third and fourth charts represent two separate views provided by a single data extract.

The third chart shows daily VTSS VTV mount and concurrent drive use, by hour, for each SYSID accessing the VTSS. The example shows the hourly statistics for SYSID "SYA5" use of VTSSID "VTSS1." Hour 0 had both the greatest number of VTV mounts (133) and the maximum number of concurrently mounted VTV drives (16).

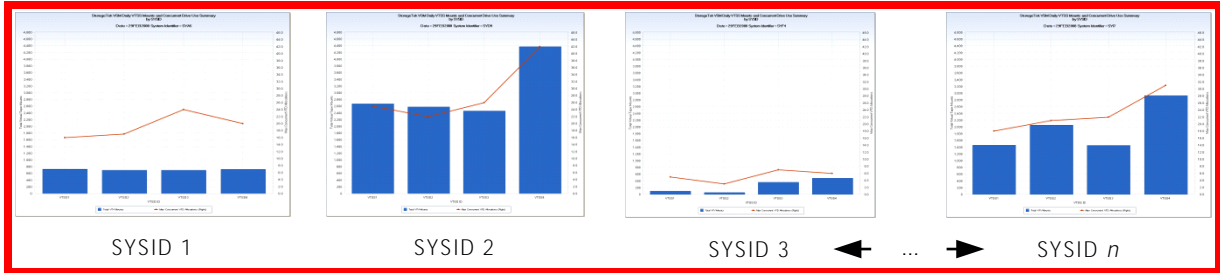
The fourth chart shows a daily view of VTSS VTV mounts with each SYSID's VTV mounts stacked using vertical bars. The example shows VTV mounts for VTSSID "VTSS1." This view shows the same VTV mount data provided in the first chart sample, but the individual SYSID contributions to total VTV mounts is provided. "SYD9" was responsible for 303 of the 455 total VTV mounts in hour 3.

Uses

This query provides insight into the hourly and overall VTV mount demand for your StorageTek VTSSs from a number of different perspectives. If concurrent VTV mount demand exceeds the number of virtual tape drives defined for a SYSID, workloads are delayed. The usage patterns seen with the variety of views provided by this query allow you to identify high and low periods—with the potential for workload shifting.

Data Extracts

STK Daily SYSID VTSS Mount Summary



Left Y-axis

VMRCOUNT – Total VTV Mounts

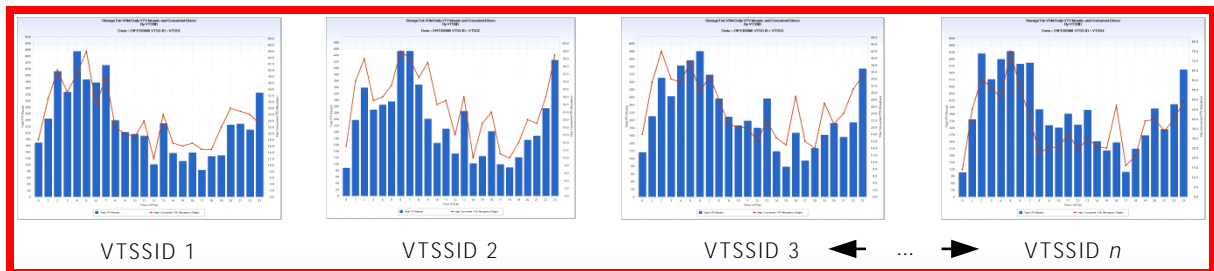
Right Y-axis

SVTMXVTD – Max Concurrent VTD Allocations

X-axis

Hour of Day

STK Daily VTSS Mounts and Drives by Hour



Left Y-axis

VMRCOUNT – Total VTV Mounts

Right Y-axis

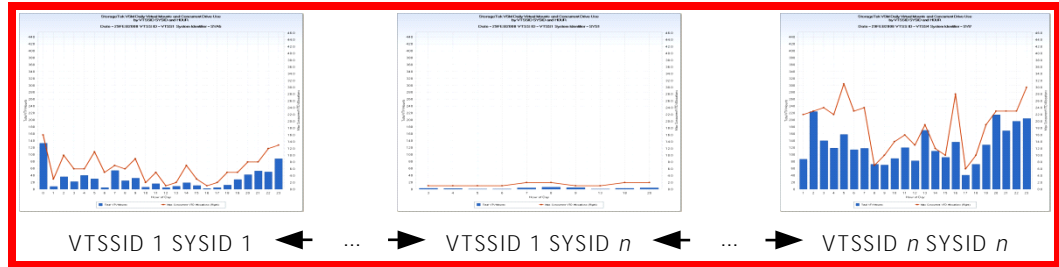
SVTMXVTD – Max Concurrent VTD Allocations

X-axis

Hour of Day

STK Daily VTSS Mounts and Drives Detail

VIEW 1 - Daily VTV Mounts and Concurrent Drives SYSID Chart



Left Y-axis

VMRCOUNT – Total VTV Mounts

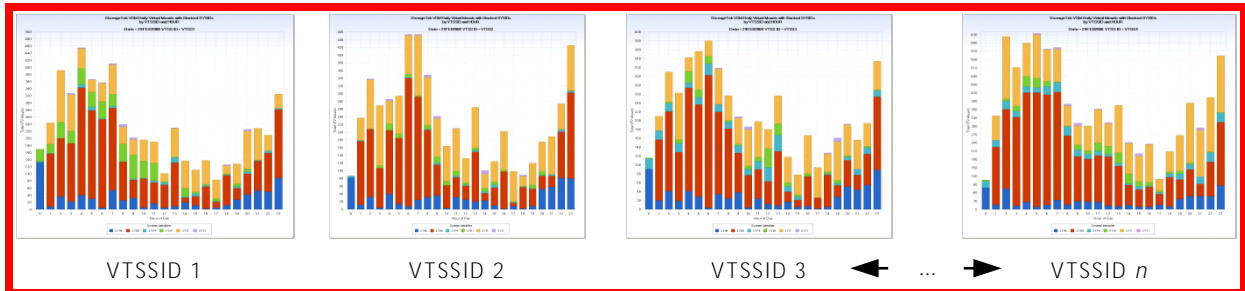
Right Y-axis

SVTMXVTD – Max Concurrent VTD Allocations

X-axis

Hour of Day

VIEW 2: Daily VTV Mounts VTSS View with Stacked SYSIDs



Left Y-axis

VMRCOUNT – Total VTV Mounts

X-axis

Hour of Day

Query Defaults

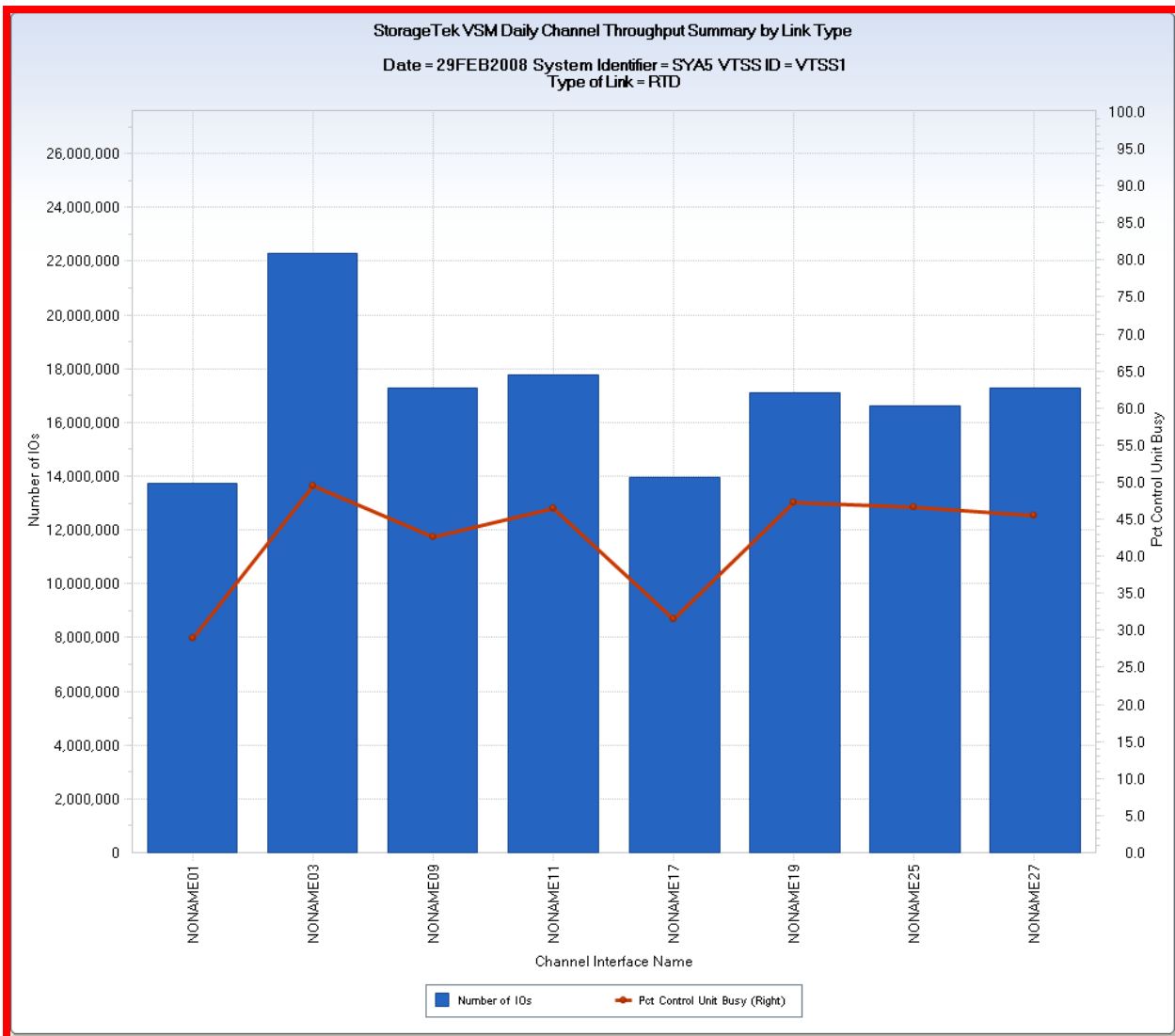
- SVTVMR DETAIL timespan file: Cycles 01-02
- SVTVDR DETAIL timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ; (Both steps)
- Derivations:
 - DATE
 - DATEPART (ENDTS)
 - SEQ
 - =1 for VMR file, =2 for VDR file
 - VTDMNTM, VTDAVTM, VTDMXTM, SVTVTDCT, SVTMNVTD, SVTAVVTD, SVTMXVTD, SVTVTDTM, and SVTVTD
 - Complex calculations in User Code step analyzing VMR and VDR records

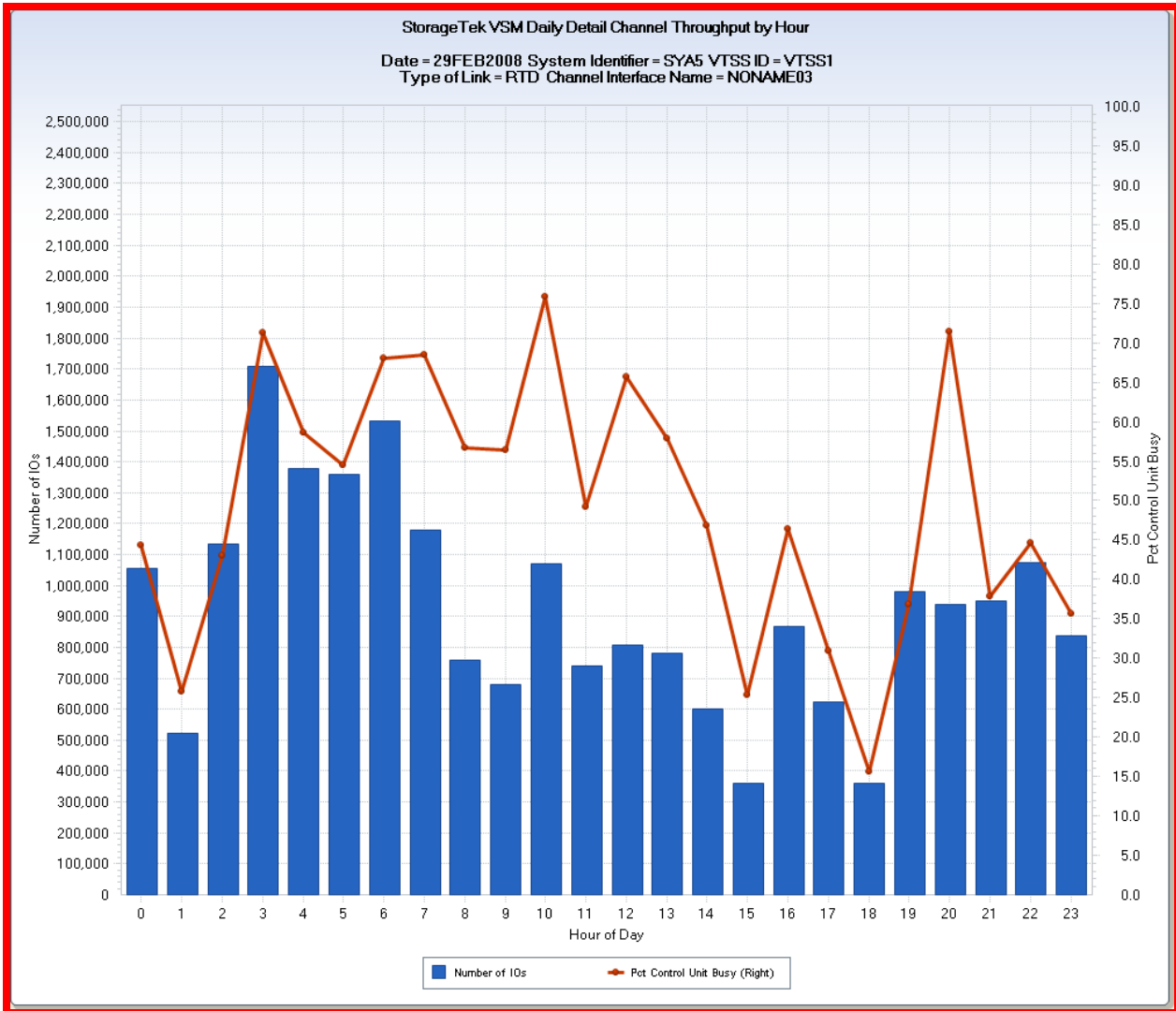
Modifications

The following modifications can be made to the STKCDA query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in both the VMR and VDR File Selection steps.
- You can add Data Filters to select particular VTSSIDs or SYSIDs, but do so in both the VMR and VDR File selection steps.
- You can plot some of the different average, minimum, and maximum elements available in the CSV files used in the Data Extracts.

STKCTA – STK VSM Daily Channel Throughput Analysis





Overview

The STKCTA query produces two sets of daily charts. The charts show the StorageTek Virtual Tape Storage Subsystem (VTSS) channel activity for the data links servicing both host applications and the real tape drives managing the physical tape back store.

A summary chart set produces two daily charts for each SYSID and VTSSID combination. The first summary chart shows the total daily channel I/O for the HOST channel links. The second summary chart shows the total daily channel I/O for the Real Tape Drive (RTD) channel links. Both charts use vertical bars to show the channel I/O and horizontal lines to show the channel links control unit busy.

The detail chart set shows the same information at the hourly level for each individual channel link.

The top example shows the summary chart for the RTD links servicing VTSSID "VTSS1" for activity originating from SYSID "SYA5." Link "NONAME03" had the highest I/O volume of all the RTD links at just over 22 million I/Os for the day.

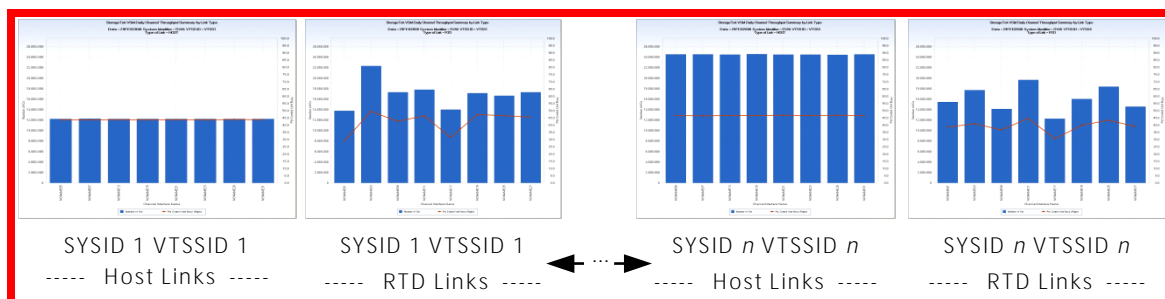
The bottom example shows the detail chart for link "NONAME03"—the high volume channel link in the shown in the summary chart example. This detail chart shows that the highest I/O volume occurred in HOUR 3 at just over 1.7 million I/O's. The channel control unit busy rate was just over 70 percent at HOUR 3.

Uses

This query is useful for determining if additional channels are required to handle I/O volume and to identify time periods where the I/O activity is approaching capacity thresholds. If control unit busy is approaching 100 percent, it is likely that some workloads are being delayed. If high I/O volume routinely occurs at specific times, you can shift workloads to different time periods to achieve greater throughput.

Data Extracts

STK Daily Channel Throughput Summary



Left Y-axis

CIPNO – Number of IOs

Right Y-axis

CIPPCUB – Pct Control Unit Busy

X-axis

CIPINM - Channel Interface Name

STK Daily Channel Throughput Detail



Left Y-axis

CIPNO – Number of IOs

Right Y-axis

CIPPCUB – Percent Control Unit Busy

X-axis

Hour of day

Query Defaults

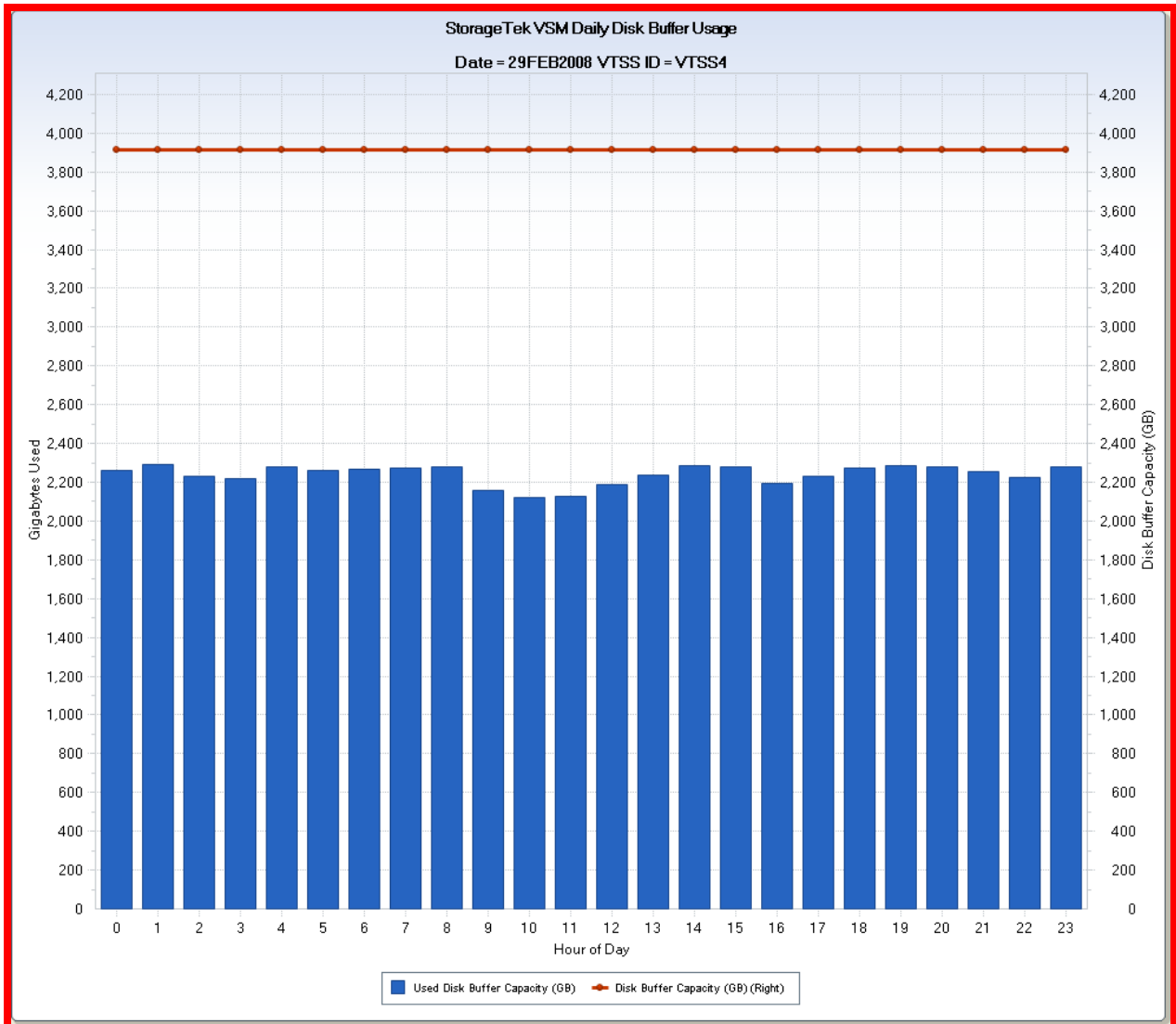
- SVTCIP DAYS timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- A SAS code step selects only active channels
- Derivations:
 - DATE
 - DATEPART (ENDTS)

Modifications

The following modifications can be made to the STKCTA query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in the CIP File Selection step.
- You can add Data Filters to the CIP File Selection step to select a particular SYSID or VTSSID.
- You can add Data Filters to the CIP File Selection step to select only HOST or RTD channel links using the CIPTOL-Type of Link data element.

STKDBU – STK VSM Daily Disk Buffer Use



Overview

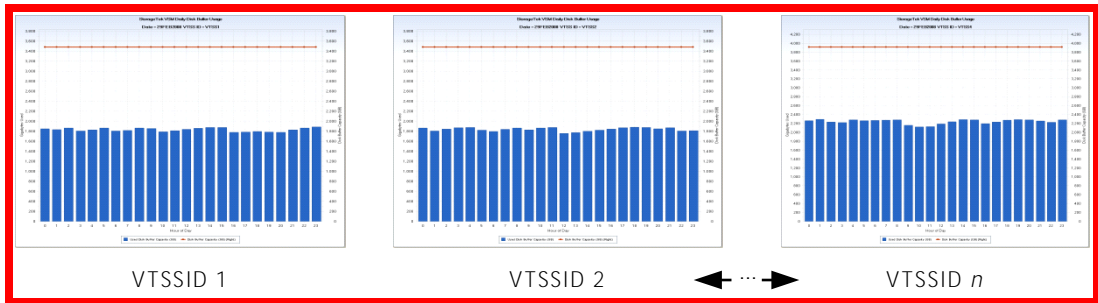
The STKDBU query produces a daily chart for each StorageTek Virtual Tape Storage Subsystem (VTSS) VTSSID. The chart shows the hourly capacity and use of the disk buffer (cache) used to hold virtual tape volumes.

Uses

This query is useful to determine if the disk buffer capacity is sufficient for workload requirements or is approaching capacity limitations. The hourly view allows you to see if capacity is constrained at particular time periods—if so you can consider workload shifting. As disk buffer use approaches capacity limits, workload throughput and performance suffer.

Data Extracts

StorageTek VSM Daily Disk Buffer Usage



Left Y-axis

SSPAVUCP – Used Disk Buffer Capacity (GB)

Right Y-axis

SSPAVTCP – Disk Buffer Capacity (GB)

X-axis

Hour of day

Query Defaults

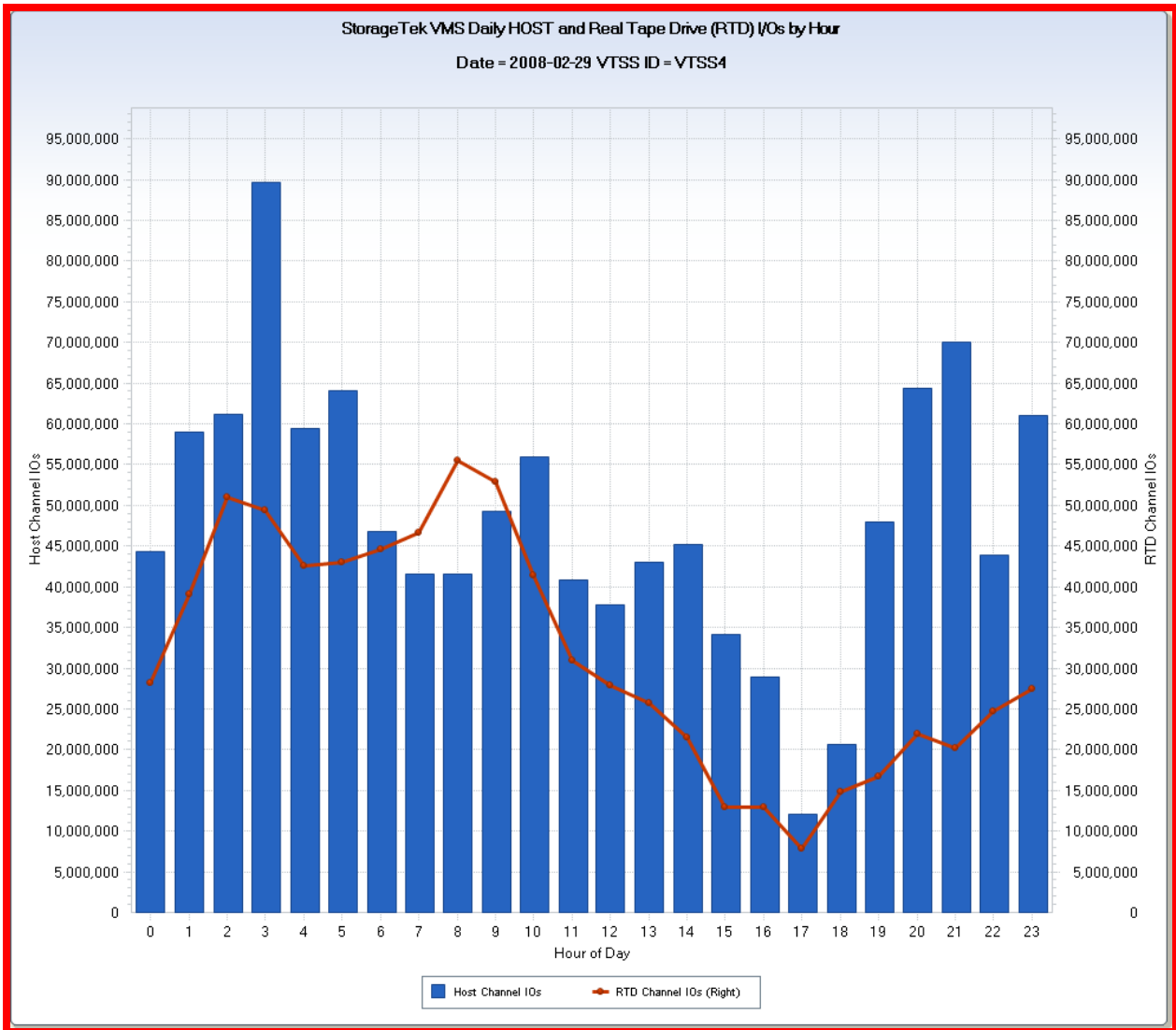
- SVTSSP DAYS timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART (ENDTS)

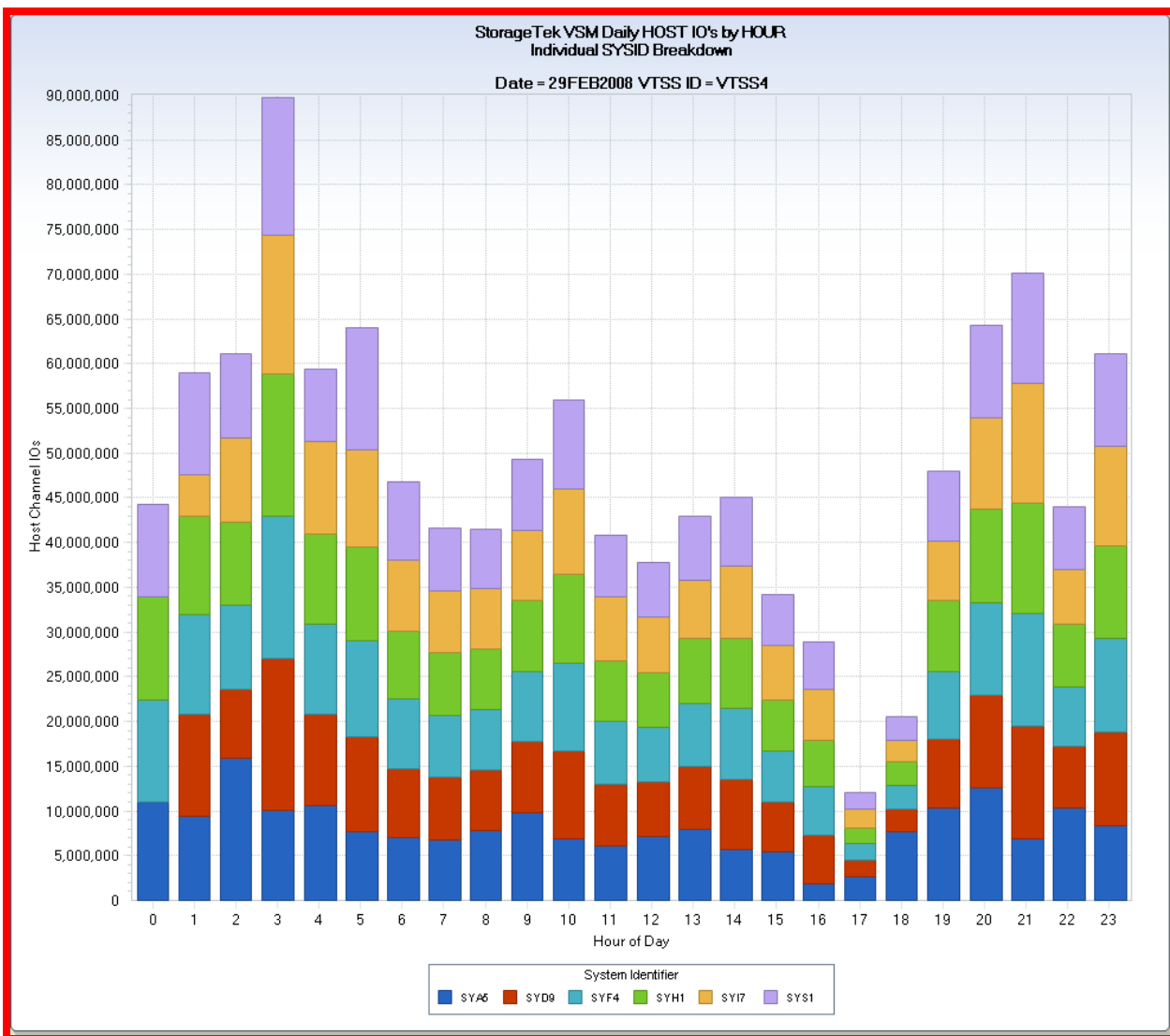
Modifications

The following modifications can be made to the STKDBU query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in the CIP File Selection step.
- You can add Data Filters to the SSP File Selection step to select a particular VTSSID.

STKIOT – STK VSM Daily I/O Throughput





Overview

The STKIOT query produces two sets of daily charts. The charts show daily host and Real Tape Drive (RTD) channel I/O for each StorageTek Virtual Tape Storage Subsystem (VTSS). One chart set shows the hourly I/O totals, with the host I/O displayed using vertical bars, and the RTD I/O shown with a vertical line. The second chart set shows the hourly host I/O for each VTSS with the I/O count for each SYSID shown with stacked vertical bars.

The top example shows the daily chart for VTSSID "VTSS4." The highest volume of host I/O activity occurred during HOUR 3, with 89.7 million channel I/Os. The peak hour for RTD I/O activity was HOUR 8, with just over 55 million I/Os.

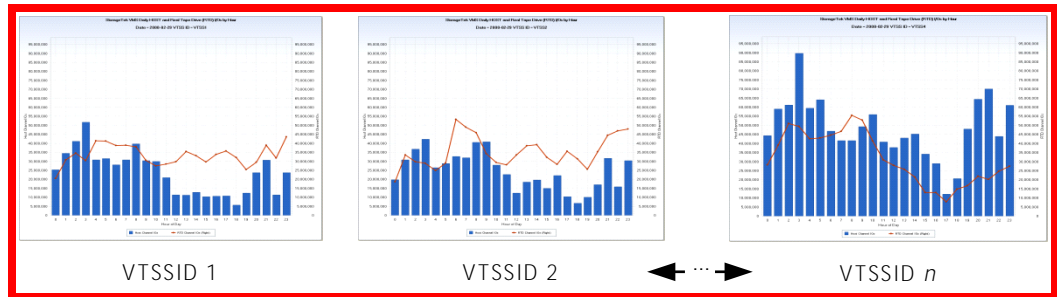
The bottom example shows the hourly host channel I/O with the SYSID breakdown for the same VTSSID. Over most hours, VTSS activity is fairly evenly spread over the attached SYSID hosts.

Uses

This query is useful for monitoring overall hourly throughput on both the front end (host) and back end (RTD) sides of your VTSSs. High use hours can be identified as potential targets for workload shifting. In addition, the SYSID breakdown allows you to see to what extent each host system is driving VTSS use—potentially exposing opportunities for workload balancing.

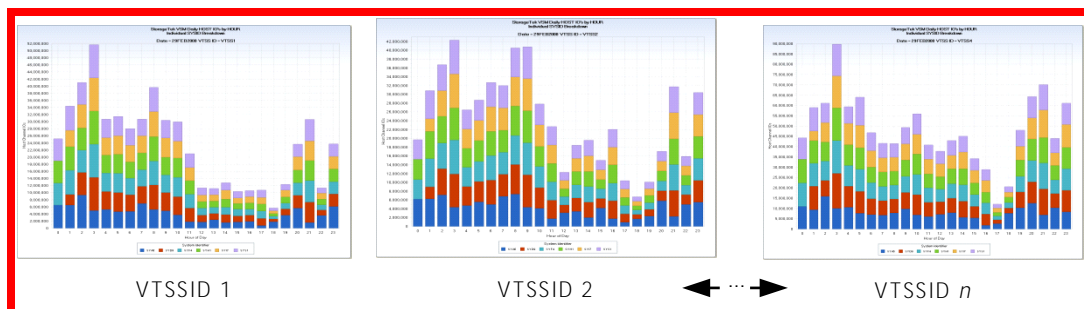
Data Extracts

STK Daily HOST and RTD IO Throughput



- Left Y-axis
 - CIPNIOH – Host Channel IOs
- Right Y-axis
 - CIPNIOR – RTD Channel IOs
- X-axis
 - Hour – Hour of day

STK Daily HOST IO Thruput Stacked SYSIDs



Left Y-axis

CIPNIOH – Host Channel IOs

X-axis

HOUR – Hour of day

Query Defaults

- SVTCIP DAYS timespan file: Cycles 01-02 (two separate File Selection steps)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ; (Both steps)
- Data Filters: First CIP File Selection Step: Selects only enabled HOST channels
- Data Filters: Second CIP First File Selection Step: Selects only enabled RTD channels
- Derivations:

DATE

DATEPART(ENDTS) ;

CIPNIOH/R

CIPNIO

HST/RTDAVCUB

CIPAVCUB

HST/RTDCUB

CIPCUB

HST/RTDCSP

CIPCSP

HST/RTDPCCUB

CIPPCUB

HST/RTDMXNIO

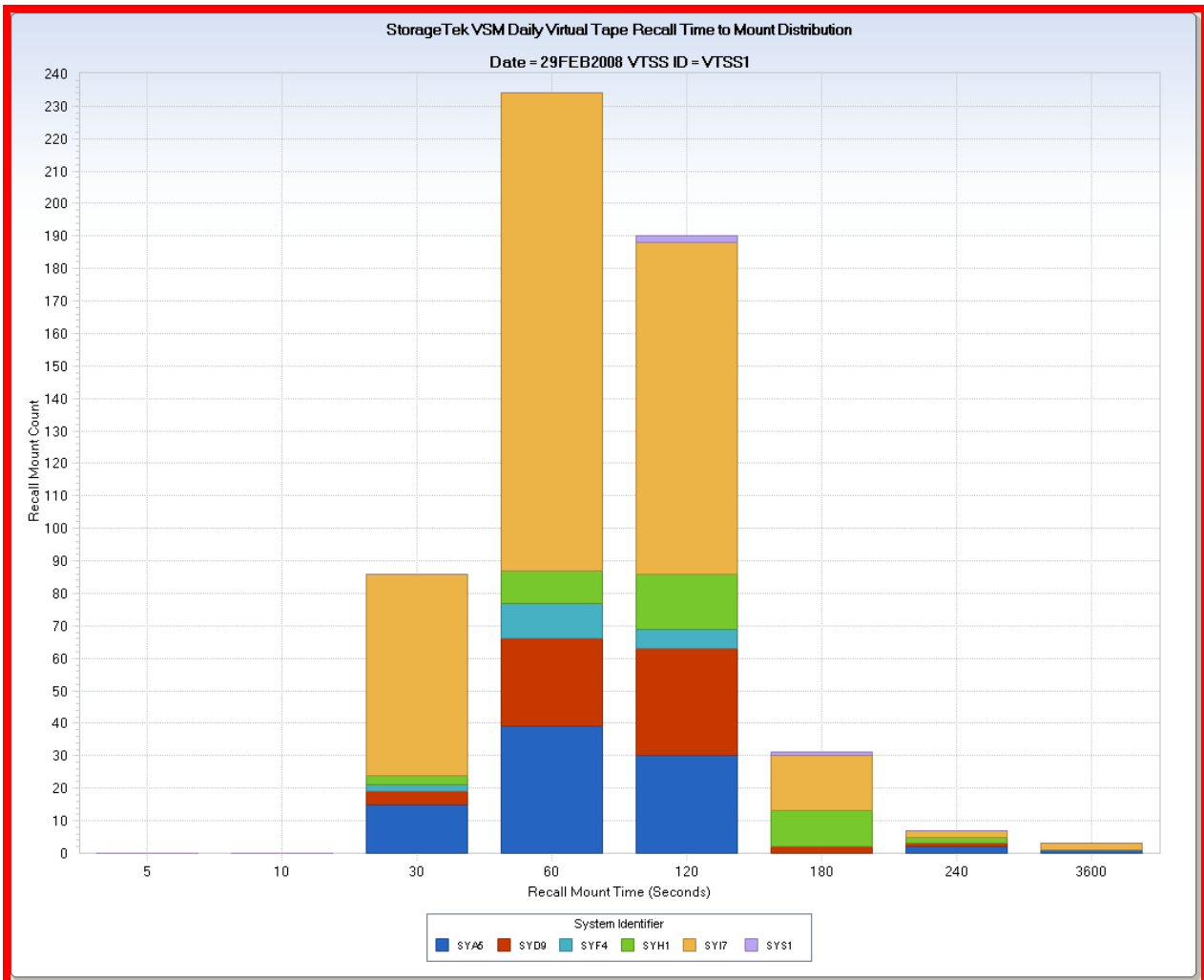
CIPMXNIO

Modifications

The following modifications can be made to the STKIOT query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in both CIP File Selection step.
- You can add Data Filters to the CIP File Selection step to select a particular SYSID or VTSSID.

STKRMD – STK VSM Daily Recall Mount Time Distribution



Overview

The STKRMD query generates a daily chart for each Virtual Tape Storage System (VTSS). The chart shows the distribution of VTV mount times required to satisfy VTV mount requests that resulted in volume recall from the physical tape back store.

The distribution buckets are set at 5, 10, 30, 60, 120, 180, 240, and 3600 seconds. If a recall mount request is satisfied between 0 and 5 seconds, it is counted in the 5-second bucket. If the mount required between 5 and 10 seconds, it is counted in the 10-second bucket, and so forth.

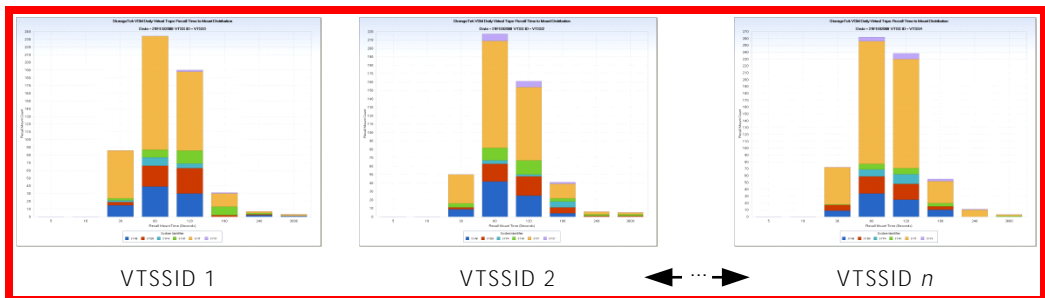
The example shows that most mounts requiring recall originated from SYSID "SYS7" and required between 30 and 60 seconds to satisfy the virtual tape mount requests.

Uses

This query is useful for determining the workload response time costs associated with virtual tape recalls. The query can be used to identify SYSIDs whose workloads generate the most recalls, which prompt investigations and strategies to make it more likely that requested volumes are in cache when required.

Data Extracts

STK Daily Recall Mount Time Distribution



Left Y-axis

MNTCNT – Recall Mount Count

X-axis

MNTTIME – Recall Mount Time (Seconds)

Query Defaults

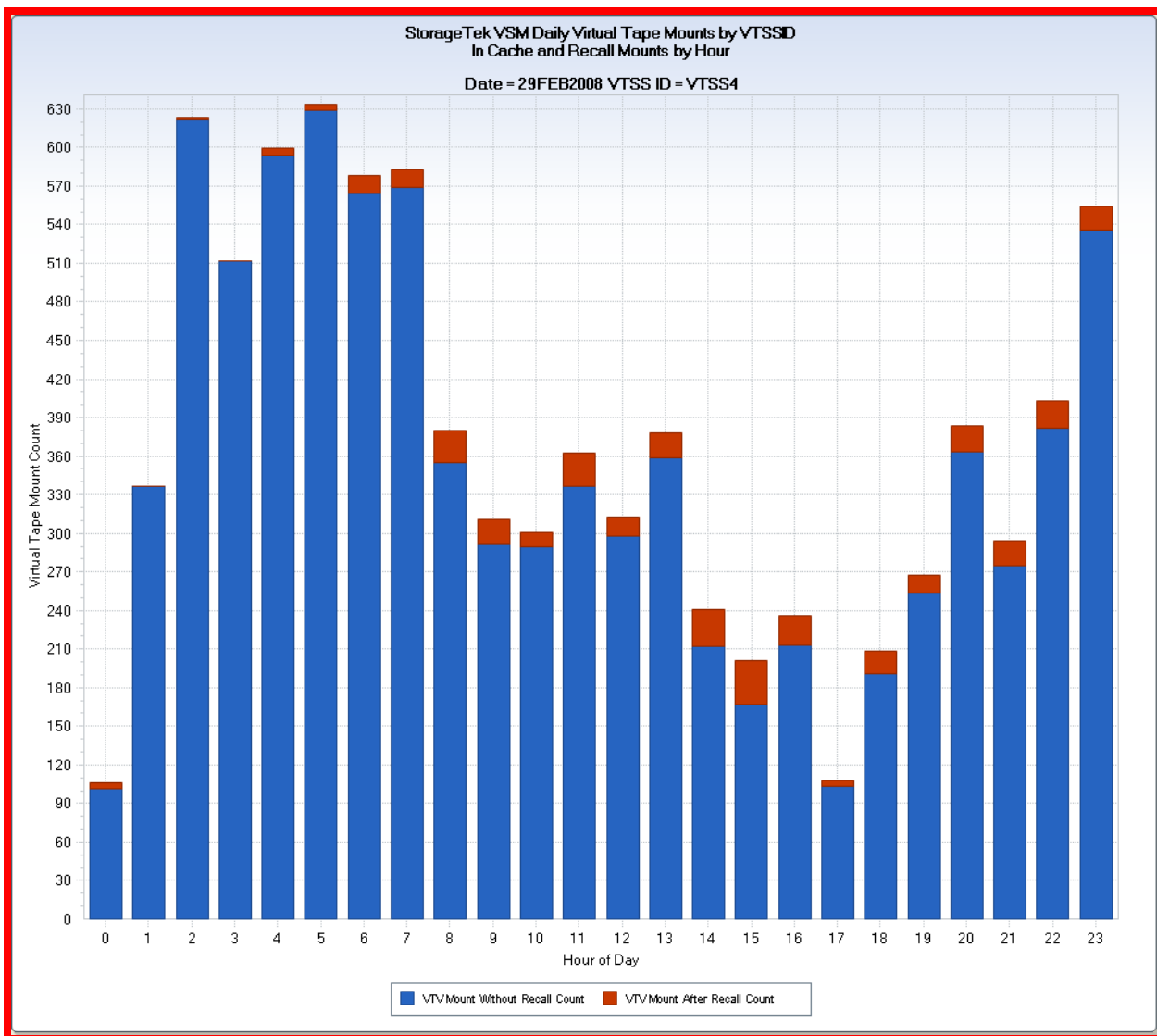
- SVTRMR DAYS timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ; (Both steps)
- Derivations:
 - DATE
 - DATEPART (ENDTS)
 - MTNCNT
 - SVTDST1 through SVTDST8
 - MNTTIME
 - SVTVAL1 through SVTVAL8

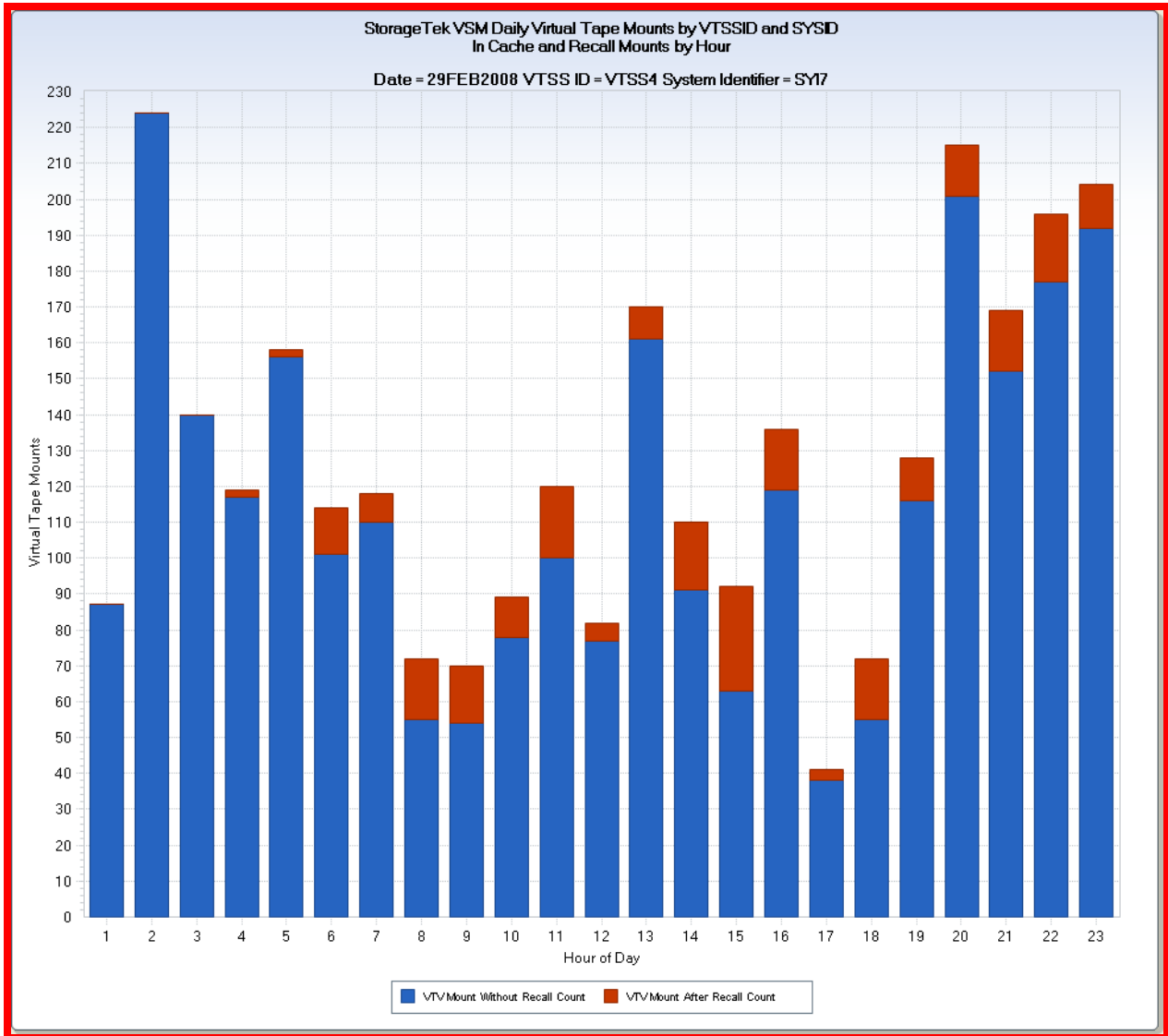
Modifications

The following modifications can be made to the STKRMD query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in the RMR File Selection step.
- You can add Data Filters to the RMR File Selection step to select a particular SYSID or VTSSID.

STKVMA – STK VSM Daily Virtual Tape Mount Activity





Overview

The STKVMA query produces two sets of daily charts. The charts show the amount of Virtual Tape Volume (VTV) mounts for each StorageTek Virtual Tape Storage Subsystem (VTSS). One chart set shows the hourly VTV mounts at the VTSSID level, and the other shows the hourly VTV mounts at the SYSID/VTSSID level. Both chart sets use vertical stacked bars to show the number of mounts satisfied in the DASD cache buffer, and the number of mounts requiring recall each hour.

The top example shows the daily summary chart for VTSSID "VTSS4." The highest number of mounts occurred during HOUR 5, with 629 mounts satisfied in cache and 5 mounts requiring recall. The highest number of mounts requiring recall was 34, and occurred during hour 15.

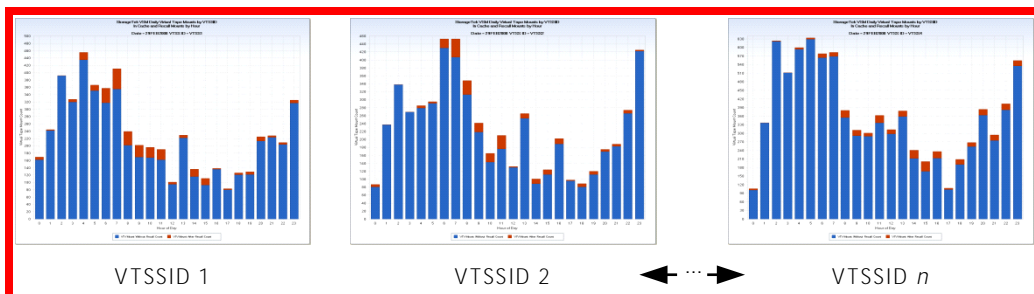
The bottom example shows the virtual tape activity for one of the systems, SYSID "SYI7", accessing VTSSID "VTSS4." SYSID "SYI7" accounted for 29 of 34 mounts recalling recall during hour 15.

Uses

This query is useful for understanding the virtual tape access activity of your VTSSs over the course of the day. Using both the VTSSID view and the SYSID level views, you can determine times where the system is both over and underutilized, offering opportunities for potential workload shifting. Investigate high levels of mounts requiring recall. Recall mounts require more time to satisfy than mounts satisfied from the DASD buffer.

Data Extracts

STK Daily Virtual Tape Mounts by VTSSID



Left Y-axis

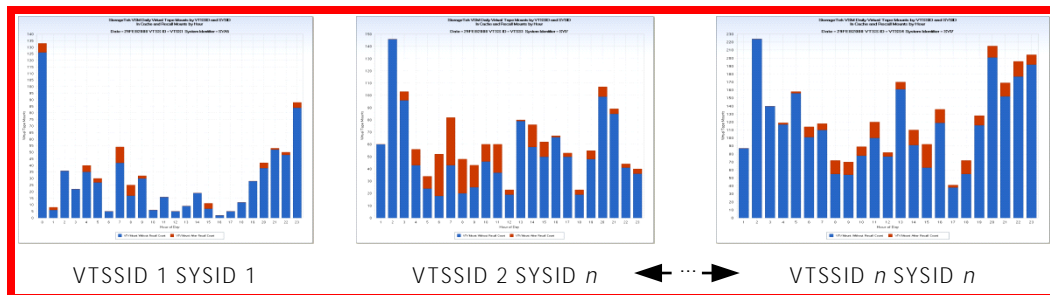
SVSMNR – VTV Mount Without Recall Count

SVSMRC – VTV Mount After Recall Count

X-axis

Hour of day

STK Daily Virtual Tape Mounts by SYSID



Left Y-axis

SVSMNR – VTV Mount Without Recall Count

SVSMRC – VTV Mount After Recall Count

X-axis

Hour of day

Query Defaults

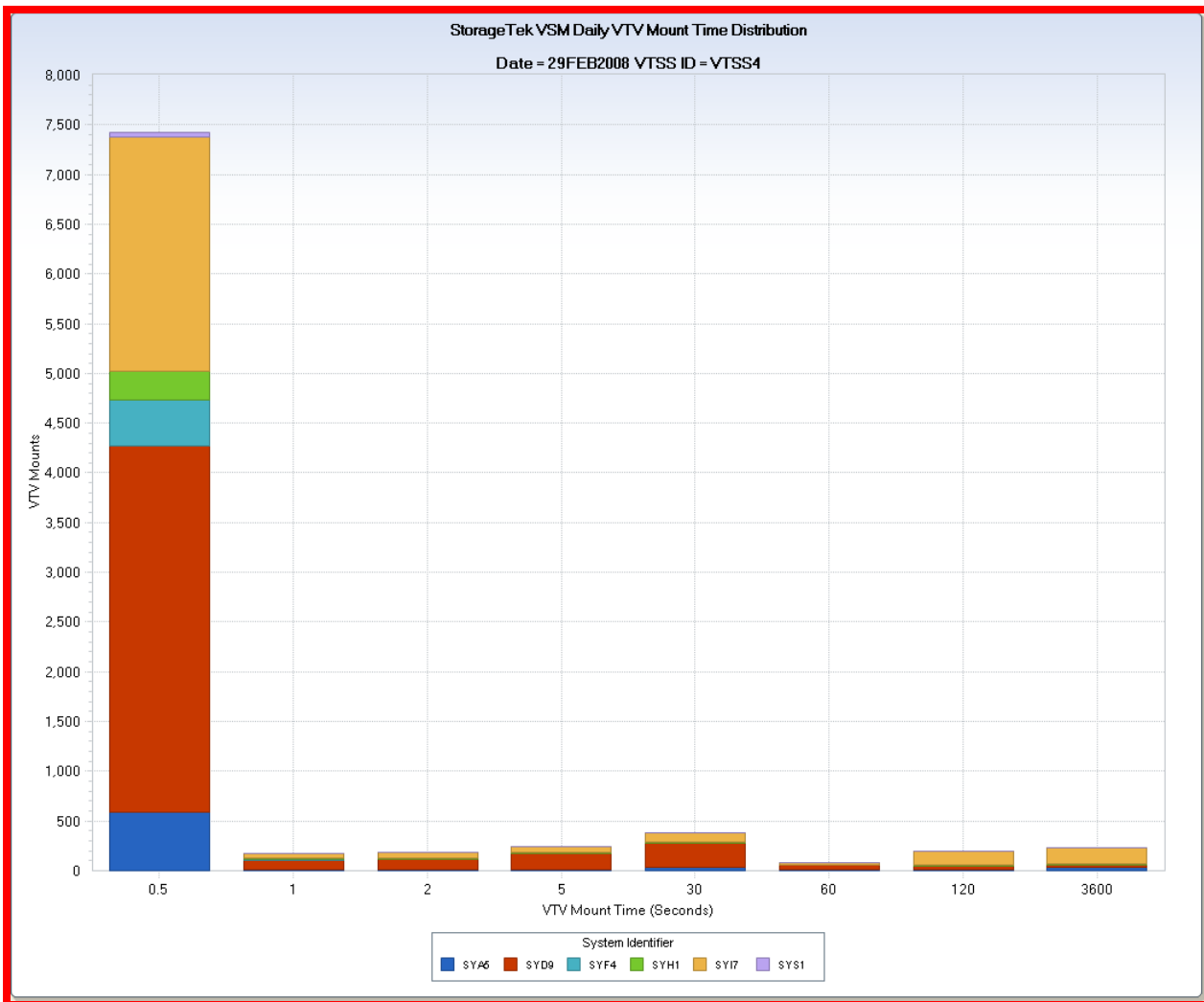
- SVTSVS DAYS timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ; (Both steps)
- Derivations:
 - DATE
 - DATEPART (ENDTS)

Modifications

The following modifications can be made to the STKVMA query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in the SVS File Selection step.
- You can add Data Filters to the SVS File Selection step to select a particular SYSID or VTSSID.

STKVMD – STK VSM Daily VTV Mount Time Distribution



Overview

The STKVMD query generates a daily chart for each Virtual Tape Storage System (VTSS). The chart shows the distribution of mount times required to satisfy all Virtual Tape Volume (VTV) mount requests—requests that were satisfied in the DASD buffer and requests requiring recall from the physical tape back store.

The distribution buckets are set at 0.5, 1, 2, 5, 30, 60, 120, and 3600 seconds. If a VTV mount request is satisfied between 0 and 0.5 seconds, it is counted in the 0.5-second bucket. If the mount required between 0.5 and 1 second, it is counted in the 1-second bucket, and so forth. The number of mounts are shown at the SYSID granularity, with each systems count contributing to a stacked vertical bar.

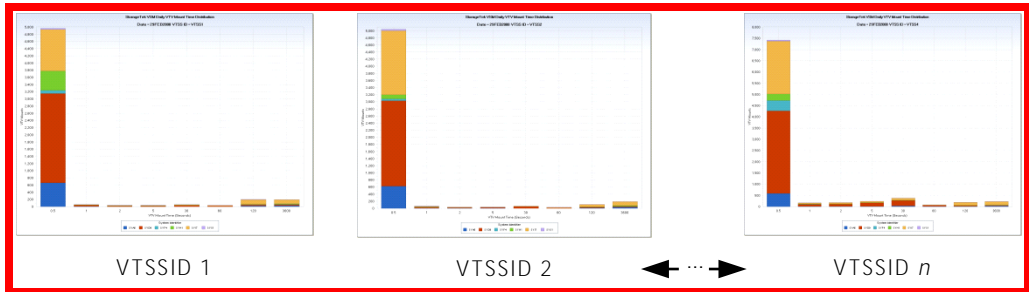
The example shows the daily VTV mount time distribution for StorageTek VTSSID "VTSS4." Most VTV mounts were satisfied in less that ½ second, and SYSID "SYD9" workloads requested most VTV mounts (2,488).

Uses

This query shows the importance of maximizing the number of DASD buffer hits for VTV mount requests. When the requested volume is in the DASD buffer, access for the requesting workload is quick. But when the requested volume is not in the buffer, it must be recalled from physical tape—resulting in long delays. Execute the STKRMD – STK VSM Daily Recall Mount Time Distribution for a deeper look at the mount time distributions specifically for VTV requests requiring recall.

Data Extracts

STK Daily VTV Mount Time Distribution



Left Y-axis

MNTCNT – VTV Mounts

X-axis

MNTTIME – VTV Mount Time (Seconds)

Query Defaults

- SVTVMR DAYS timespan file: Cycles 01-02
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- Derivations:
 - DATE
 - DATEPART (ENDTS)
 - MTNCNT
 - SVTDST1 through SVTDST8
 - MNTTIME
 - SVTVAL1 through SVTVAL8

Modifications

The following modifications can be made to the STKVMD query:

- You can increase the CYCLES input to report on multiple days, but comment out the user code that selects yesterday's data in the VMR File Selection step.
- You can add Data Filters to the VMR File Selection step to select a particular SYSID or VTSSID.

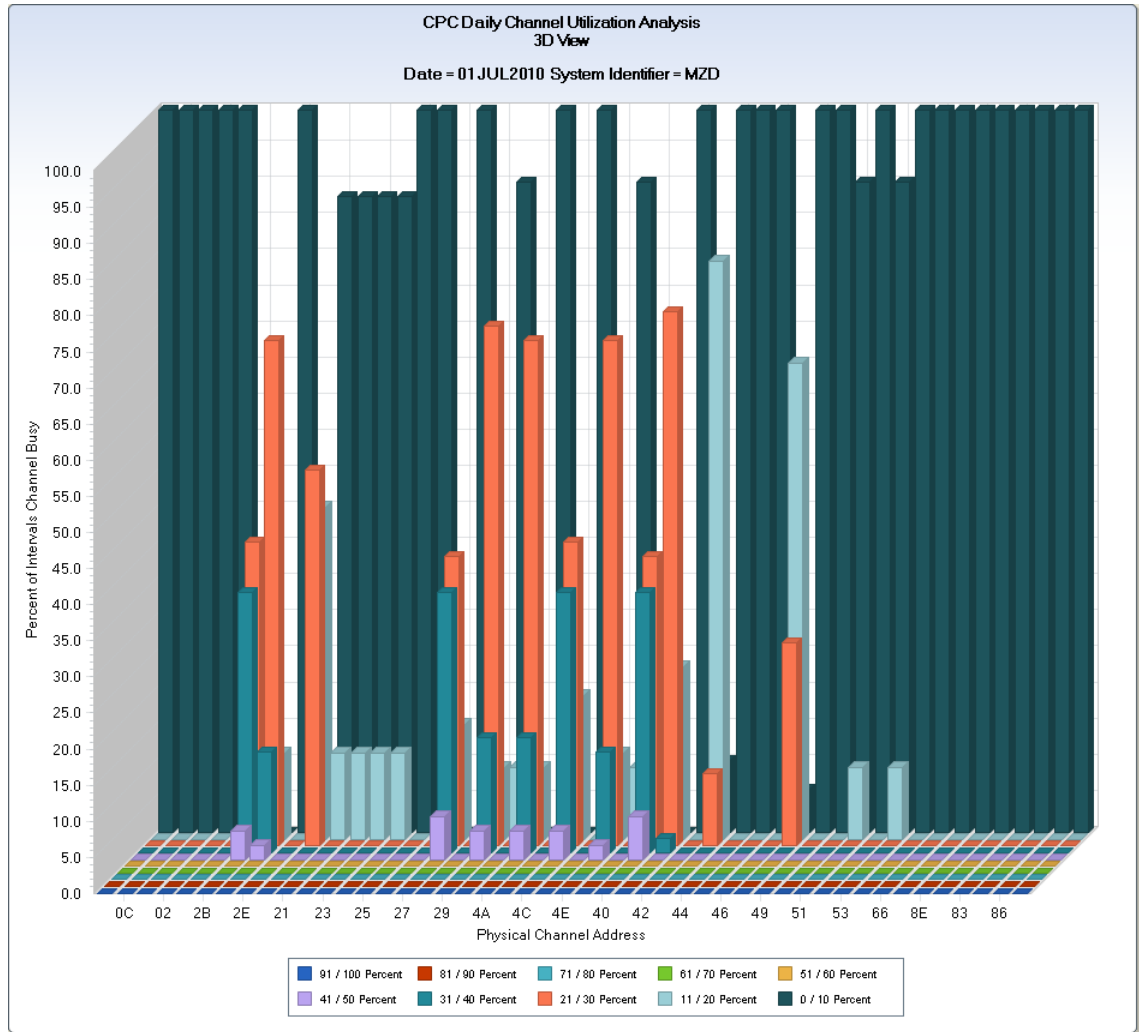
Chapter 9: CA MICS Analyzer Option for VM/CMS (VMC) Queries

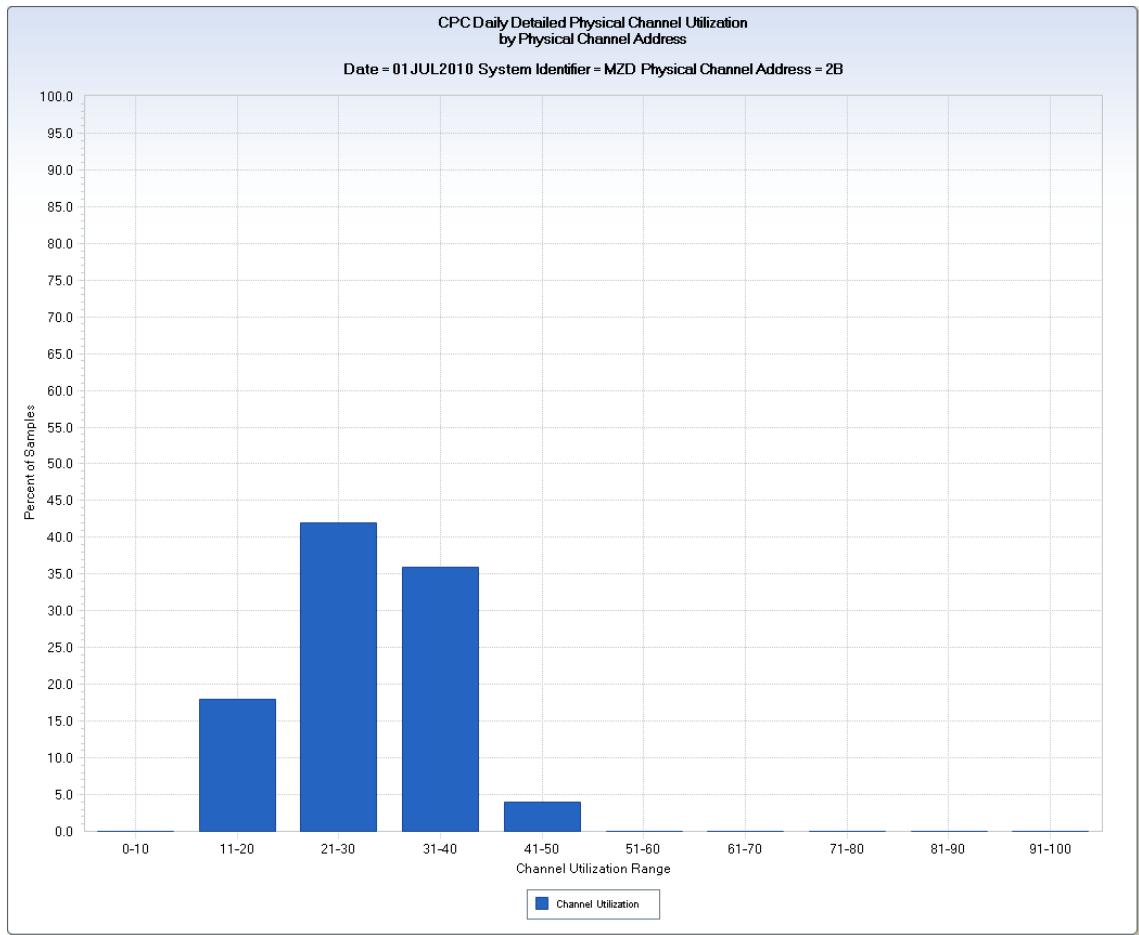
This section contains the following topics:

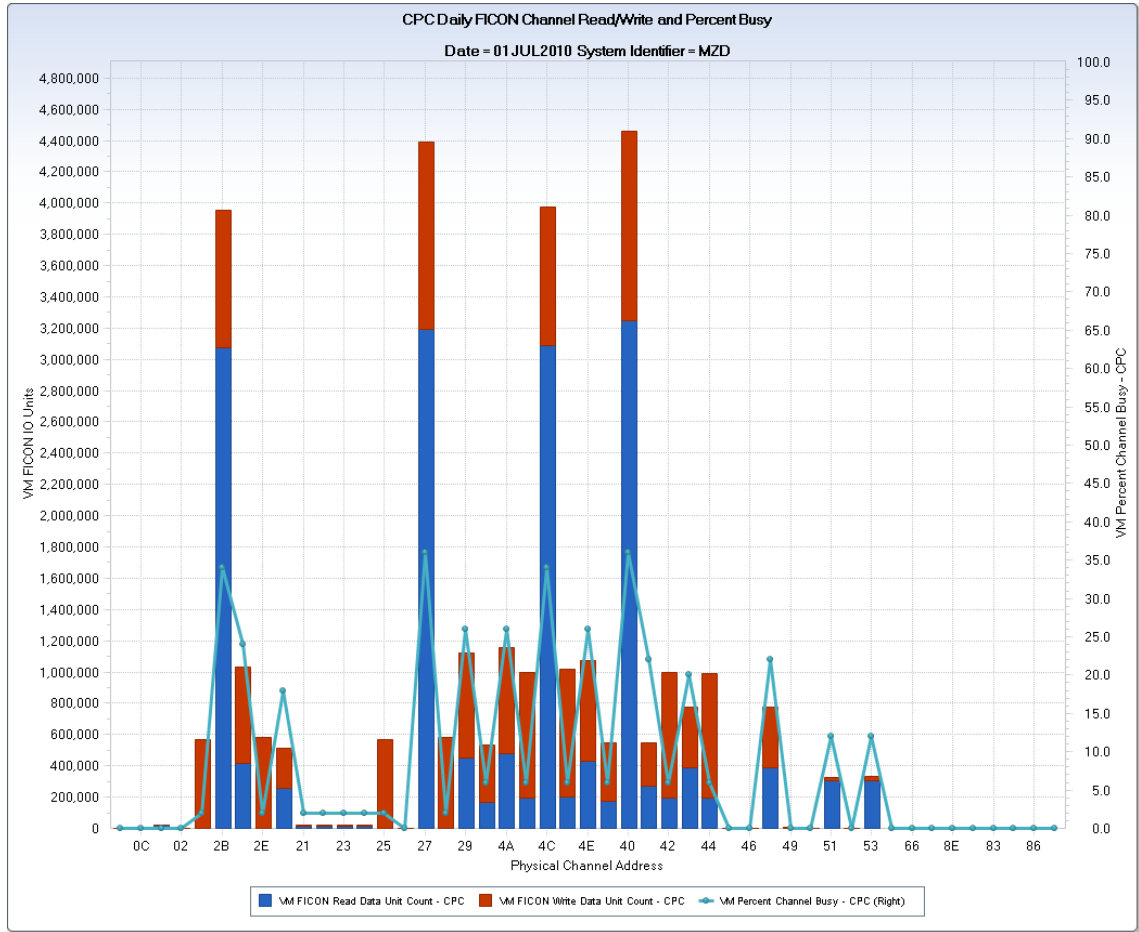
[VMCCHA – VM CPC Daily Physical Channel Utilization](#) (see page 390)

[VMCT10 – VM System Daily Top 10 CPU Users](#) (see page 398)

VMCCHA – VM CPC Daily Physical Channel Utilization







CPC Overall Channel Load Analysis													18:28 Tuesday, February 22, 2011
DATE RANGE 30JUN10:23:53:11.44 TO 01JUL10:11:59:11.44													
----- System Identifier=MZD -----													
Physical Channel Address	VM Channel Measurement Group	Avg Channel Busy	Percent										
			0 / 10	11 / 20	21 / 30	31 / 40	41 / 50	41 / 50	61 / 70	71 / 80	81 / 90	91 / 100	
27	FICON	28	0	16	40	36	6	6	0	0	0	0	
40	FICON	28	0	18	40	36	6	6	0	0	0	0	
2B	FICON	28	0	18	42	36	4	4	0	0	0	0	
4C	FICON	28	0	20	42	36	4	4	0	0	0	0	
29	FICON	26	0	10	72	16	4	4	0	0	0	0	
4A	FICON	26	0	10	70	16	4	4	0	0	0	0	
4E	FICON	26	0	12	70	14	2	2	0	0	0	0	
2D	FICON	26	0	12	70	14	2	2	0	0	0	0	
41	FICON	22	0	24	74	2	0	0	0	0	0	0	
48	FICON	18	6	66	28	0	0	0	0	0	0	0	
43	FICON	16	10	80	10	0	0	0	0	0	0	0	
20	FICON	20	0	46	52	0	0	0	0	0	0	0	
51	FICON	6	90	10	0	0	0	0	0	0	0	0	
53	FICON	6	90	10	0	0	0	0	0	0	0	0	
31	FICON	4	90	10	0	0	0	0	0	0	0	0	
4B	FICON	2	100	0	0	0	0	0	0	0	0	0	
4D	FICON	2	100	0	0	0	0	0	0	0	0	0	
4F	FICON	4	90	10	0	0	0	0	0	0	0	0	
42	FICON	2	100	0	0	0	0	0	0	0	0	0	
44	FICON	2	100	0	0	0	0	0	0	0	0	0	
2A	FICON	2	100	0	0	0	0	0	0	0	0	0	
2E	FICON	2	100	0	0	0	0	0	0	0	0	0	
21	FICON	4	88	12	0	0	0	0	0	0	0	0	
22	FICON	4	88	12	0	0	0	0	0	0	0	0	
23	FICON	4	88	12	0	0	0	0	0	0	0	0	
24	FICON	4	88	12	0	0	0	0	0	0	0	0	
49	FICON	0	100	0	0	0	0	0	0	0	0	0	
50	FICON	0	100	0	0	0	0	0	0	0	0	0	
52	FICON	0	100	0	0	0	0	0	0	0	0	0	
60	ESCON	0	100	0	0	0	0	0	0	0	0	0	
66	ESCON	0	100	0	0	0	0	0	0	0	0	0	
8B	ESCON	0	100	0	0	0	0	0	0	0	0	0	

Overview

The VMCCHA query provides a tabular report and several charts showing Central Processing Complex (CPC) level physical channel utilization. Three separate data extracts show physical channel utilization:

- Overall daily use in a single 3D chart
- Daily use chart for each physical channel
- Daily FICON channel use with a breakdown of read and write volume

In addition, the query produces a SASLIST daily report showing utilization for each physical channel. The charts and reports are produced by VM SYSID, but each SYSID shows the exact same utilization numbers. This is because the data reported is for overall CPC level utilization. As mentioned in Query Modifications, you can filter the data to select a single VM SYSID to cut down on redundant charts.

Channel utilization is shown using percent busy ranges, 0-10 percent, 11-20 percent, 21-30 percent utilization, and so on. As an illustration, for a given channel 'X', you see that the channel 'X' was 0-10 percent busy 12 percent of the day, 11-20 percent busy 18 percent of the day, and so forth.

The top example shows the 3D chart. The channel addresses run along the X-axis, from left to right. Channel 40 was 41-50 percent busy (lilac bar) 6 percent of the day.

The next example shows the detail level chart, where a separate daily chart is produced for each physical channel address. The example shows that channel "2B" was 21-30 percent busy, 42 percent of the day, and it was 41-50 percent busy 4 percent of the day.

The final chart example shows overall daily statistics for FICON channels only. The average percent busy for each channel is shown using a horizontal line. The daily totals for read and write FICON I/O units are shown using stacked vertical bars.

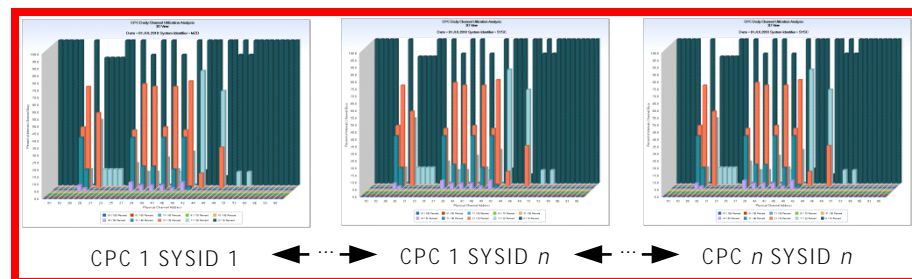
The SASLIST report generates a line for each physical channel, showing the channel type (for example, FICON or ESCON) and the same daily percent utilization metrics displayed with the charts.

Uses

This query provides several views to understand the daily physical channel utilization of your CPCs. If channel utilization is uneven, certain workloads can be delayed. This chart provides an easy way to detect over utilized channels that demand deeper analysis.

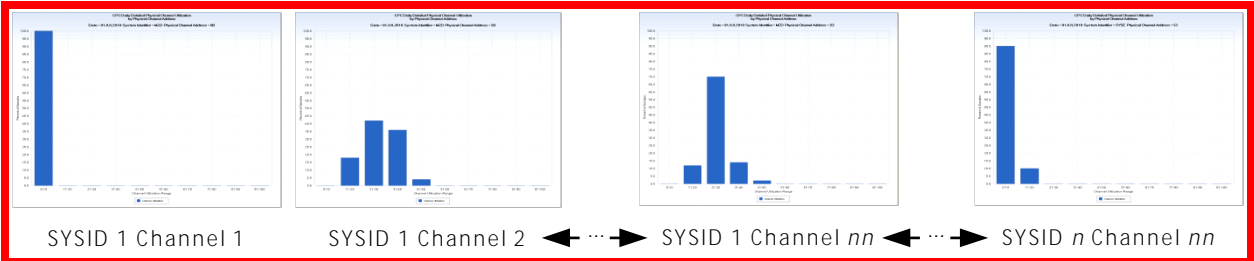
Data Extracts

CPC Daily 3D Channel Utilization Chart



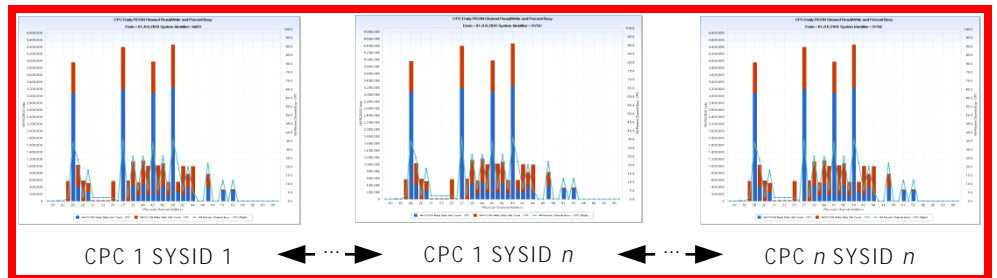
Left Y-axis
mm-nn Percent Utilization
 X-axis
 Physical Channel Address

CPC Daily Detailed Physical Channel Use



Left Y-axis
mm-nn Percent Utilization
 X-axis
 Channel Utilization Range

CPC Daily FICON Channel Reads and Writes



Left Y-axis
 VM FICON Read Data Unit Count – CPC
 VM FICON Write Data Unit Count – CPC
 Right Y-axis
 CPMPCBSY – VM Percent Channel Busy - CPC
 X-axis
 Channel – Physical Channel Address

Query Defaults

- VMXCPM DETAIL timespan file: Cycles 01-02
- Filters: CMPPCBSY GT 0 ; (Only active channels)
- Derivations:
 - PCmm_nn
 - IF CMPPCBSY LE nn THEN PCmm_nn +1 ; (over all observations by Channel)
 - TOTSAMP
 - Interval count by Channel
 - TOTPCT
 - CMPPCBSY (summed over all observations by Channel)
 - PCTnn
 - PCTnn=ROUND((PCmm_nn/TOTSAMP*100),2);
 - DATE
 - DATEPART(ENDTS)
 - TSSTART
 - MIN(STARTTS) over all observations by SYSID
 - TSEND
 - MAX(ENDTS) over all observations by SYSID
 - RANGE
 - '0-10', '11-20, ... '91-100'
 - RVALUE
 - PCTnn

Modifications

The following modifications can be made to the VMCCHA query:

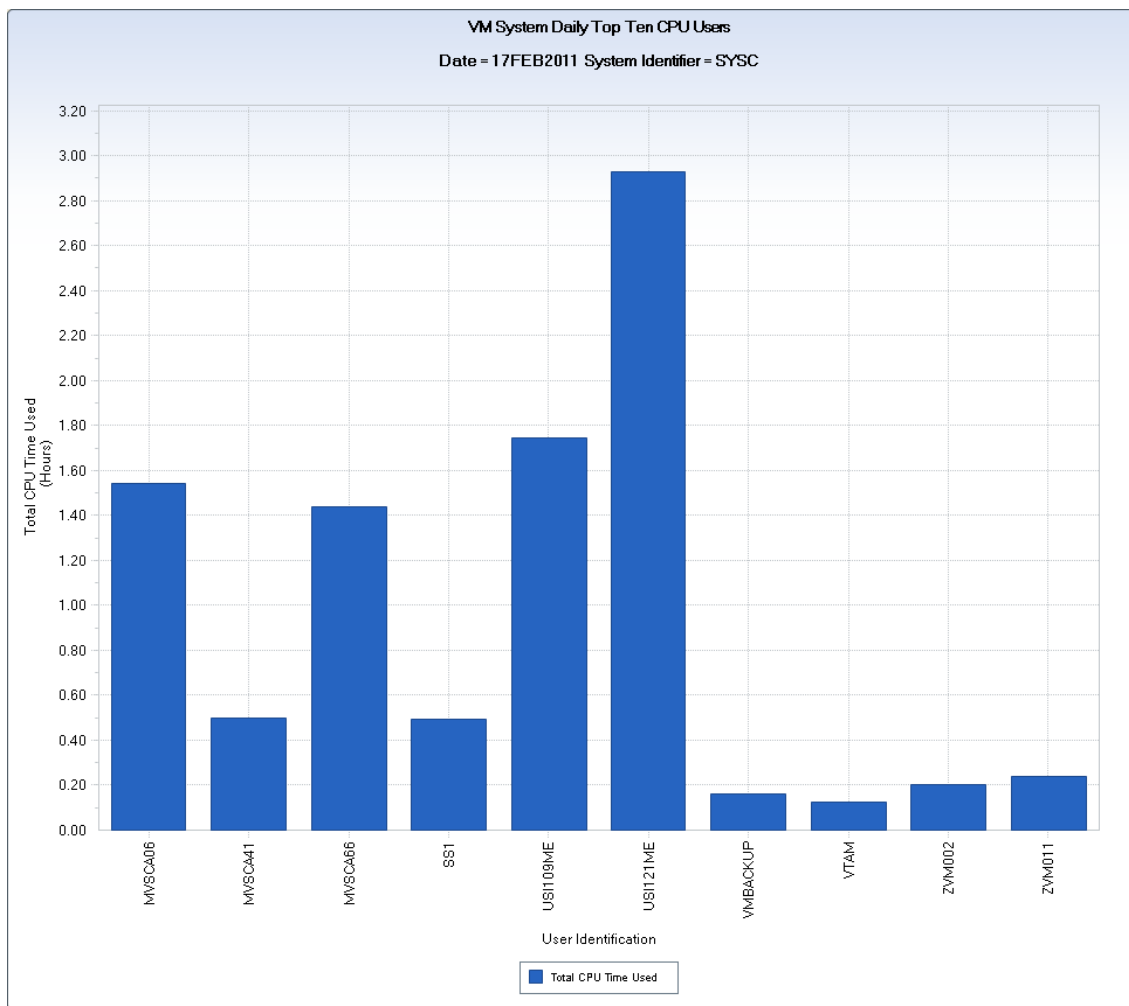
- Each VM system running on a given CPC produces the exact same set of channel charts. If you have multiple VM LPARs running on a CPC, add a FILTER statement to the CPM File Selection step that selects a single VM system from each CPC. For example, if "SYV1" is a VM LPAR on CPC 1, and "PYV1" is a VM LPAR on CPC 2, a FILTER statement:

```
SYSID EQ "SYV1" AND  
SYSID EQ "PYV1" ;
```

would eliminate redundant charts from the other VM systems.

- You can increase the cycles input to create charts for multiple days, but comment out the user code in the VMXCPM file input step that restricts processing to yesterday's data.

VMCT10 – VM System Daily Top 10 CPU Users



Overview

The VMCT10 query generates a daily chart for each VM System showing the top ten users based on total CP CPU time use. Total CPU time includes both system and virtual time.

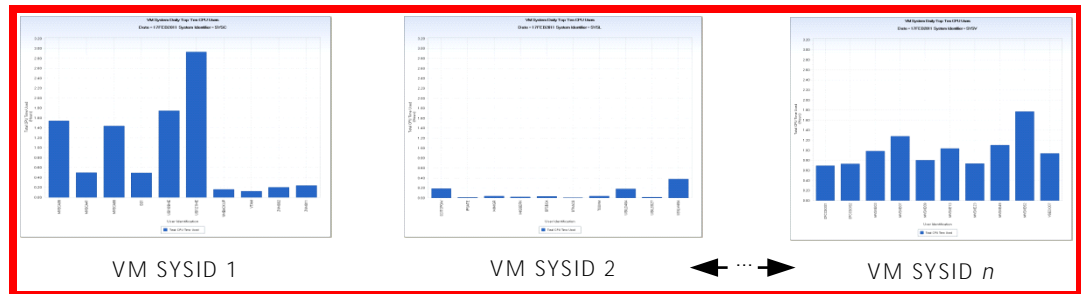
The example shows the top ten users for SYSID "SYSC." User "US1121ME" was the top user and consumed almost three hours of total CPU time.

Uses

This query is useful to determine the major CPU time users on your VM systems. Analyze these top resource users to see if they are running as effectively as possible.

Data Extracts

VM System Daily Top 10 CPU Users



Left Y-axis

SYSID : VMXTTIME – Total CPU Time Used

X-axis

User – User Identification

Query Defaults

- VMXVXU DAYS timespan file: Cycle 01
- Derivations:
 - DATE
 - DATEPART(ENDTS) ;
 - CPURATE
 - VXUTIME / DURATION ;
 - VIOTOTAL
 - SUM(VXUCONIO, VXUDASIO, VXUVURIO, VXUCTCIO, VXUMSCIO) ;
 - VIORATE
 - VIOTOTAL / DURATION;
 - OTHW_PCT
 - OTHW_PCT = 100* (CSUOTWAT / SUM(CSUPETTM, CSUIOWAT, CSUOTWAT)) ;

Modifications

The following modifications can be made to the query:

- You can adjust the `TOPnn` value to something other than the default of 10.
- You can add filter statements to select only a single `SYSID` or select a subset of users (for example, `USER EQ: "A"` would select only users beginning with the letter "A.")
- You can increase the `cycles` input to create charts for multiple days, but comment out the user code in the `VMXVXU` file input step that restricts processing to yesterday's data.
- Save the query as a new name and change the input to the `MONTHS` timespan to get the top ten users for the month. Make appropriate query chart title changes.

Note: If you change the `TOPnn` value (default TOP 10), make sure that you modify the query chart title to reflect the changed value.

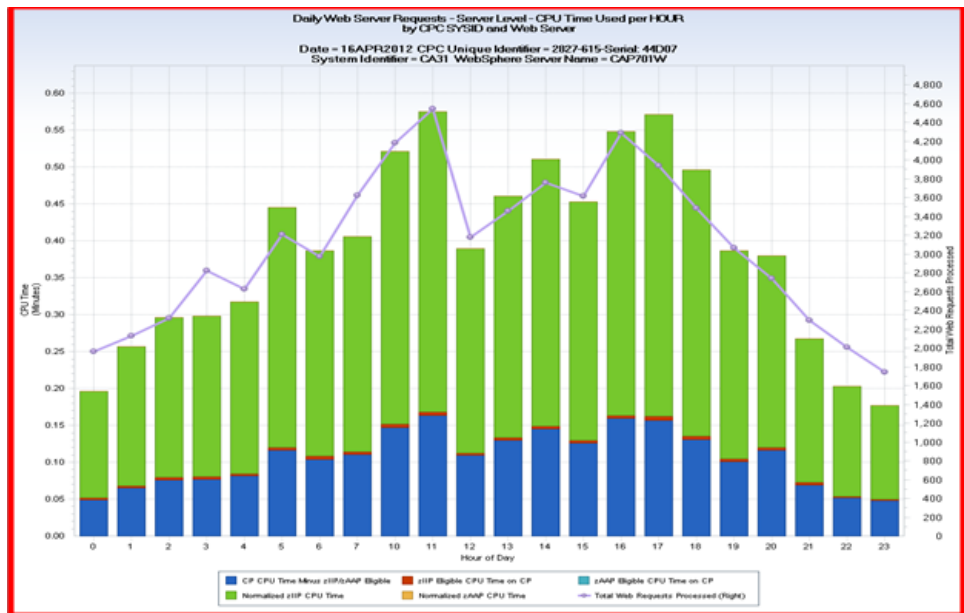
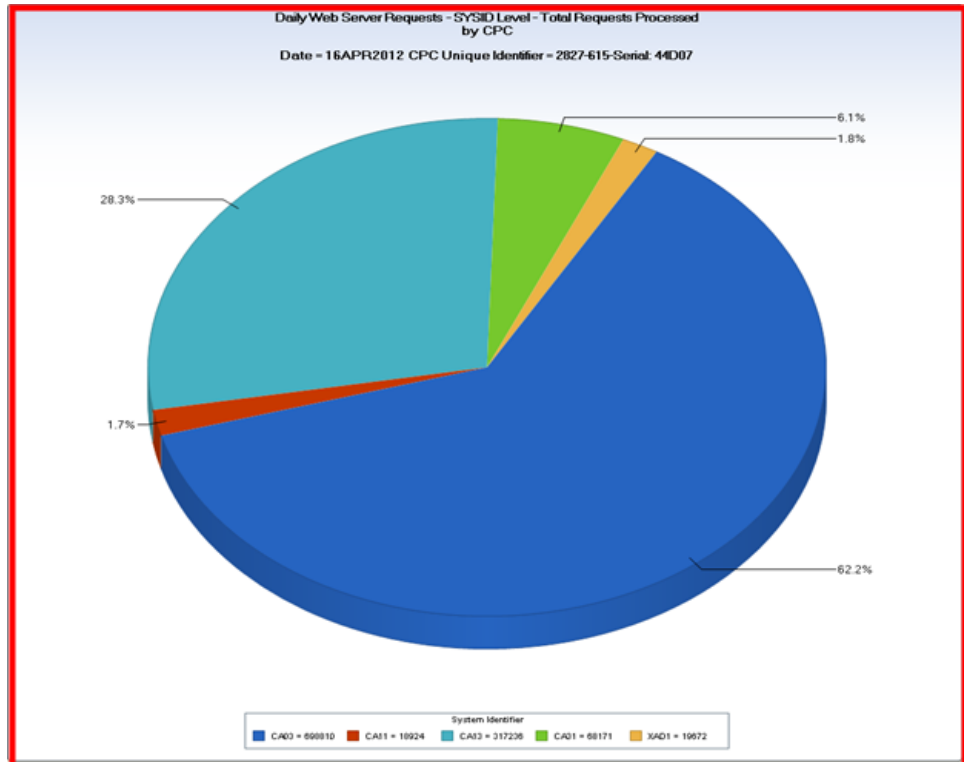
Chapter 10: CA MICS WEB Analyzer Option (WEB) Queries

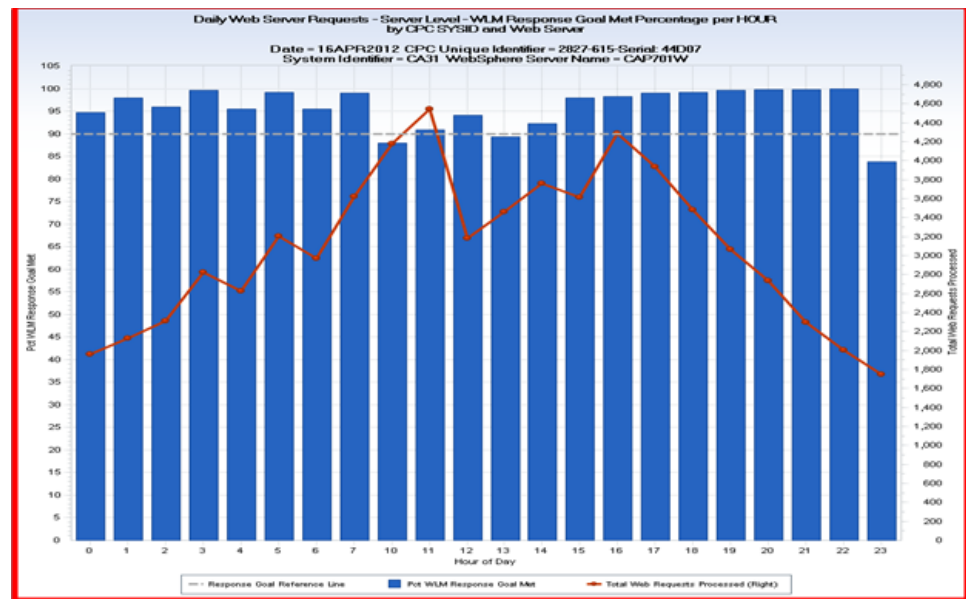
This section contains the following topics:

[WEBDRA– Daily Web Server Request Analysis](#) (see page 402)

[WEBMRA - 30 Day Web Server Request Activity](#) (see page 416)

WEBDRA- Daily Web Server Request Analysis





Overview

The WEBDRA query generates a series of data extracts that support the daily analysis of synchronous web server requests, processed by z/OS hosted web servers, at a variety of summarization levels. At the highest level, the overall daily resource use and performance information for each z/OS system in a Central Processing Complex (CPC) can be viewed in a single chart. At an intermediary level, individual charts display the same information at the web server level.

And at the lowest level, individual charts are generated for each type of synchronous request (HTTP, INTERNAL, MBEAN) processed by individual web server for each hour of the day.

Six CSV output files are generated by the query.

- The first, third, and fifth CSV output files generate charts showing the overall daily resource usage and performance information for synchronous web requests at the z/OS system (SYSID), web server, and request type summarization levels, respectively.
- The second, fourth, and sixth CSV output files generate charts at similar summarization levels, but show the resource usage and performance information for each hour of the day.

The six CSV data extracts generated by the WEBWRS query are:

- (1) Daily Web Server Analysis SYSID Summary
- (2) Daily Web Server Analysis SYSID by HOUR
- (3) Daily Web Server Analysis Web Server Summary
- (4) Daily Web Server Analysis Web Server by HOUR
- (5) Daily Web Server Analysis Request Type Summary
- (6) Daily Web Server Analysis Request Type by HOUR

Each summary CSV data extract (1, 3, and 5) generates the following four chart views:

- Daily Web Server Summary – *granularity* Level Requests Processed
- Daily Web Server Summary – *granularity* Level CPU Use
- Daily Web Server Summary – *granularity* Level WLM Response Goal Met
- Daily Web Server Summary – *granularity* Level Response Time
where *granularity* is SYSID (CSV–1), Server (CSV–3), and ReqType (CSV–5)

Each "by HOUR" CSV data extract (2, 4, and 6) generates the following four chart views:

- Daily Web Server by HOUR – *granularity* Level Requests Processed
- Daily Web Server by HOUR – *granularity* Level CPU Use
- Daily Web Server by HOUR – *granularity* Level WLM Response Goal Met
- Daily Web Server by HOUR – *granularity* Level Response Time
where *granularity* is SYSID (CSV–2), Server (CSV–4), and ReqType (CSV–6)

Chart View #1 - Requests Processed

Each chart image generated with view #1 displays the total number of synchronous requests processed at:

- The SYSID level (CSV data extracts 1 and 2)
- The web server level (CSV data extracts 3 and 4)
- The request type level (CSV data extracts 5 and 6)

Chart view #1 displays the following:

For the summary CSV data extracts (1, 3, and 5), pie charts are generated, where the pie slice size is dependent on the number of synchronous requests processed.

- In the charts generated from the SYSID level CSV data extract (1), the pie represents a Central processing complex and each pie slice represents the number of requests processed on each z/OS SYSID.
- In the charts generated from the Web Server level CSV data extract (3), the pie represents a z/OS SYSID, and each pie slice represents the number of requests processed by an individual web server.
- In the charts generated from the Request Type (ReqType) level CSV data extract (6), the pie represents a z/OS hosted web server, and each pie slice represents the number of requests processed by each request type (HTTP, INTERNAL, or MBEAN) on the web server.

For the by HOUR CSV data extracts (2, 4, and 6), vertical bar charts are generated, where the height of each bar represents the number of synchronous requests processed each hour of the day.

- In the charts generated from the SYSID level CSV data extract (2), each chart represents the daily activity for a z/OS SYSID, and the vertical bars show the number of requests processed, each hour of the day, on that LPAR.
- In the charts generated from the web server level CSV data extract (4), each chart represents the daily activity for a z/OS hosted web server, and the vertical bars show the number of requests processed each hour of the day for that web server.
- In the charts generated from the Request Type (ReqType) Web Server level CSV data extract (6), each chart represents the daily activity for a z/OS hosted web server, with stacked vertical bars showing the hourly number of requests processed for each request type (HTTP, INTERNAL, or MBEAN) on the web server.

The first chart sample previously displayed is an example of the pie chart generated with Chart View #1 for the SYSID Summary CSV data extract (1). This chart sample shows the daily count of synchronous web requests executed by web servers hosted on each of six z/OS LPARs on an IBM 2827-615 (zEC12 CPC with 15 sub-capacity CP engines). You can see that the majority of requests (about 62 percent) were processed by web servers hosted on SYSID CA03 (dark blue pie slice). SYSIDs CA13 (light blue – 28 percent) and CA31 (green – 6 percent) were the next busiest LPARs.

Chart View #2 - CPU Time

Each chart image generated with view #2 displays the CPU time used by synchronous web requests at the SYSID level (CSV data extracts 1 and 2), Web Server level (CSV data extracts 3 and 4), and Request Type level (CSV data extracts 5 and 6).

CPU time is displayed using stacked vertical bars, with each bar segment indicating CPU use for a specific processor type (CP, zIIP, or zAAP), and if on a CP processor, the portion, if any, that was zIIP or zAAP processor eligible. The count of synchronous requests processed is displayed using a horizontal line.

The level of detail provided in each of the six CSV data extracts is similar to that described above for Chart View #1. For more detailed information about the data elements displayed with Chart View #2, refer to the DATA EXTRACTS section below.

The second sample above is an example of Chart View #2, generated by the Daily Web Sync Reqs – Web Server by HOUR CSV data extract (4). This chart shows the hourly CPU use for web server CAP701W hosted on SYSID CA31 on an IBM 2827-615 (zEC12 CPC with 15 sub-capacity CP engines) for one day. You can see that the majority of CPU time used by synchronous requests occurs on zIIP processors (green bar segments). About 1/4th to 1/3rd of CPU use occurs on CP processors (blue), and access to zIIP processors is very good—because very little zIIP eligible CPU time was executed on CP processor engines (red bar segments). The number of synchronous requests processed each hour is shown with a horizontal line, and shows that CPU use per request is consistent—evident from the strong correlation observed between the number of requests processed and the CPU time used.

Chart View #3 - WLM Response Goal Met

Each chart image generated with view #3 displays the percentage of web requests that met or exceeded their WLM response time goal at the SYSID level (CSV data extracts 1 and 2), Web Server level (CSV data extracts 3 and 4), and Request Type level (CSV data extracts 5 and 6).

In all cases, the percentage met is shown using vertical bars. The left y axis is scaled from 0 to 105 so that the 100% WLM Response goal met percentage bar height ends near the top of the chart viewing area. The chart view includes a customizable horizontal reference line, with a default value showing a 90 percent response goal met value. The count of synchronous requests processed is displayed using a horizontal line.

The level of detail provided in each of the six CSV data extracts is similar to that described for the preceding chart views. Refer to the DATA EXTRACTS section below for more information about chart view #3.

The third sample above is an example of Chart View #3, generated by the Daily Web Sync Reqs – Web Server by HOUR CSV data extract (4). This chart shows the hourly WLM response time goal met percentage for synchronous transactions processed by web server CAP701W hosted on SYSID CA31 on an IBM 2827-615 (zEC12 CPC with 15 sub-capacity CP engines) for one day. You can see that close to 100 percent of the transactions met their WLM response time goal most hours of the day. Hours 10 and 23 showed the worst performance. Only about 88 percent of the transactions met their goal during hour 10, and about 84 percent met the response time goal during hour 23.

Chart View #4 - Response Time

Each chart image generated with view #4 displays the average synchronous request response time broken down into two components; queue time and dispatch time. These response time components are displayed at the SYSID level (CSV data extracts 1 and 2), Web Server level (CSV data extracts 3 and 4), and Request Type level (CSV data extracts 5 and 6).

In all cases, the two response time components are shown using stacked vertical bars. The count of synchronous requests processed is displayed using a horizontal line.

The level of detail provided in each of the six CSV data extracts is similar to that described above for the previously described chart views. Refer to the DATA EXTRACTS section below for more information about chart view #4.

Uses

This query generates charts showing daily resource use and performance information about synchronous web requests processed by web servers hosted on z/OS LPARs. At most installations, the use of synchronous requests is far greater than that for asynchronous requests, so it is critical to monitor their quantity, resource use, and performance.

The six CSV data extracts provided by this query provide the analyst with both highly summarized and very detailed insight into z/OS LPAR hosted web server request activity. The SYSID level charts allow daily activity to be viewed with just a few charts. If unusual behavior is noted, the more granular level charts can be used to isolate the specific web server or synchronous request type responsible.

These charts can be used to evaluate specialty processor (zIIP/zAAP) use.

- If excessive CP processor time is being used to process zIIP or zAAP eligible work, it might indicate the need for additional specialty processors.
- If excessive queue time is observed, it might indicate a need for workload balancing. Or specific web servers might need additional resources to help reduce the time waiting for dispatch.

The WLM Response Goal Met charts can assist in fine tuning your WLM Service Policy to ensure that goals and levels of importance are set appropriately.

Data Extracts

DATA EXTRACT: Daily Web Server Analysis SYSID Summary

VIEW 1 of 4: Daily Web Server Summary – SYSID Level Requests Processed



Y axis

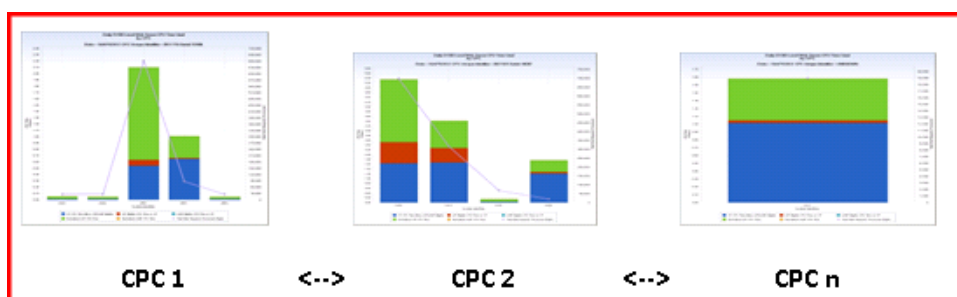
WRAREQST – Total Web Requests Processed

X axis

SYSID – System Identifier

Note that each pie slice shows the percentage of synchronous web requests processed by a particular z/OS SYSID. The legend shows SYSID name and count of web requests for each pie slice.

VIEW 2 of 4: Daily Web Server Summary – SYSID Level CPU Use



Left Y axis

WRAXCSTM – CP CPU Time Minus zIIP/zAAP Eligible

WRASUCTM – zIIP Eligible CPU Time on a CP

WRAZACTM – zAAP Eligible CPU Time on a CP

WRASPNTM – Normalized zIIP CPU Time W

WRAZPNTM – Normalized zAAP CPU Time

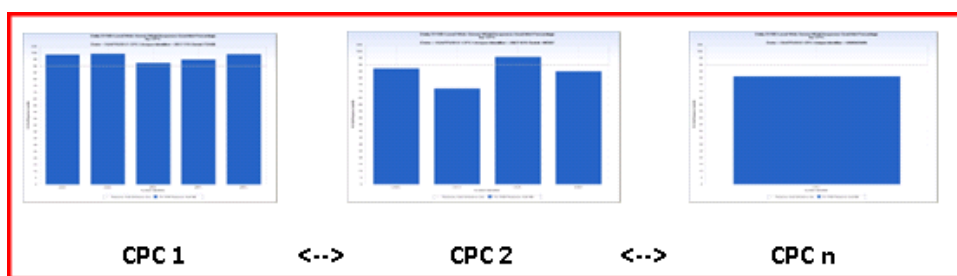
Right Y axis

WRAREQST – Total Web Requests Processed

X axis

SYSID – System Identifier

VIEW 3 of 4: Daily Web Server Summary – SYSID Level WLM Response Goal Met



Left Y axis

WRAPCGMT – Pct WLM Response Goal Met

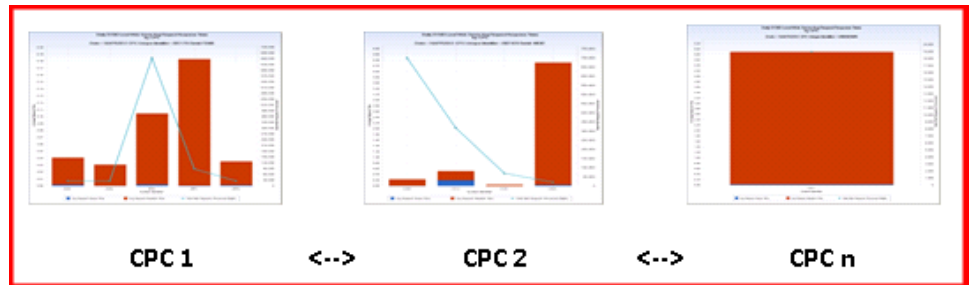
Right Y axis

WRAREQST – Total Web Requests Processed

X axis

SYSID – System Identifier

VIEW 4 of 4: Daily Web Server Summary – SYSID Level Response Time



Left Y axis

WRAAVQTM – Avg Request Queue Time

WRAAVDTM – Avg Request Dispatch Time

Right Y axis

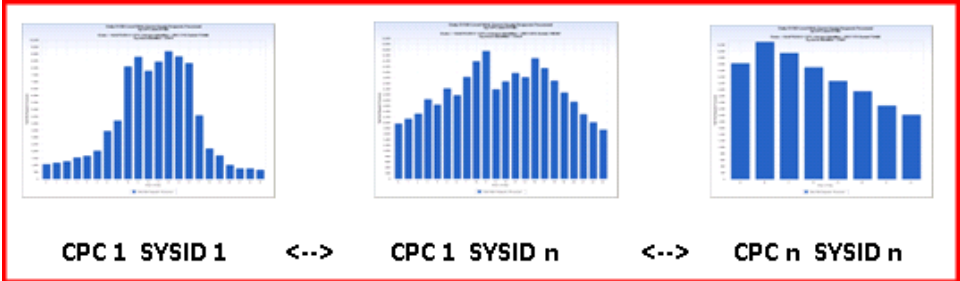
WRAREQST – Total Web Requests Processed

X axis

SYSID – System Identifier

DATA EXTRACT: Daily Web Server Analysis SYSID by HOUR

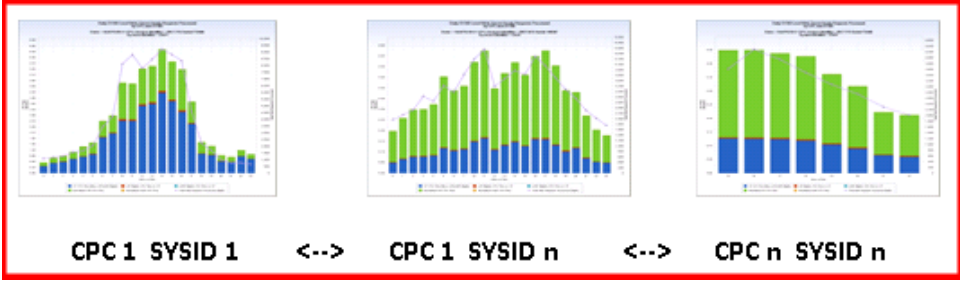
VIEW 1 of 4: Daily Web Server by HOUR – SYSID Level Requests Processed



Y axis
WRAREQST – Total Web Requests Processed

X axis
HOUR – Hour of Day

VIEW 2 of 4: Daily Web Server by HOUR – SYSID Level CPU Time



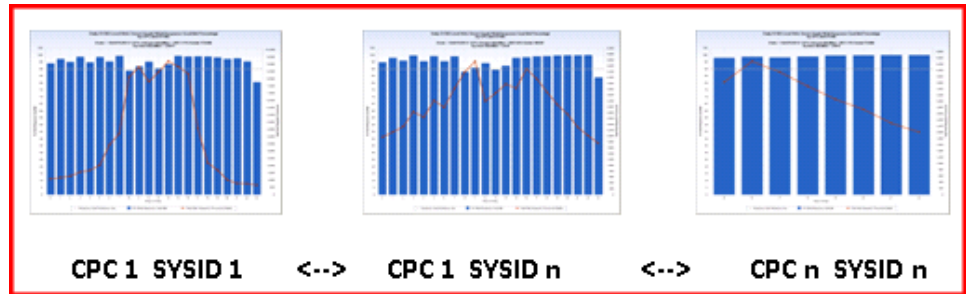
Left Y axis

- WRAXCSTM – CP CPU Time Minus zIIP/zAAP Eligible
- WRASUCTM – zIIP Eligible CPU Time on a CP
- WRAZACTM – zAAP Eligible CPU Time on a CP
- WRASPNTM – Normalized zIIP CPU Time W
- WRAZPNTM – Normalized zAAP CPU Time

Right Y axis
WRAREQST – Total Web Requests Processed

X axis
HOUR – Hour of Day

VIEW 3 of 4: Daily Web Server by HOUR – SYSID Level WLM Response Goal Met



Left Y axis

WRAPCGMT – Pct WLM Response Goal Met

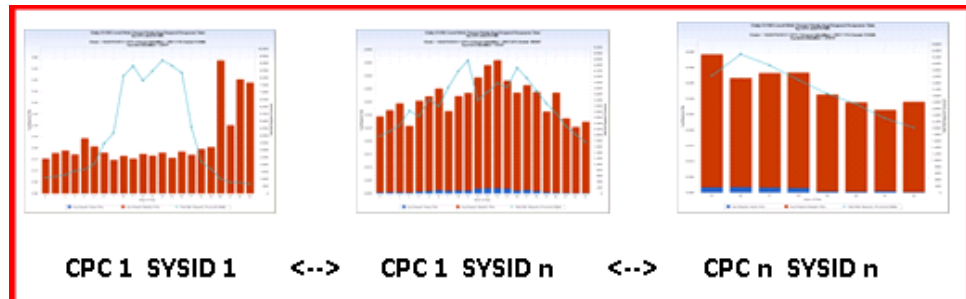
Right Y axis

WRAREQST – Total Web Requests Processed

X axis

HOUR – Hour of Day

VIEW 4 of 4: Daily Web Server by HOUR – SYSID Level Response Time



Left Y axis

WRAAVQTM – Avg Request Queue Time

WRAAVDTM – Avg Request Dispatch Time

Right Y axis

WRAREQST – Total Web Requests Processed

X axis

HOUR – Hour of Day

DATA EXTRACT: Daily Web Server Analysis Web Server Summary

The four data views in this data extract are similar to the data extracts shown above for Data Extract DAILY Web Server Analysis SYSID Summary.

The only difference is in individual chart scope. Where in the SYSID summary, each chart represents a CPC, and individual SYSIDs are shown along the x axis, in the Web Server summary, each chart represents a SYSID and individual web servers are shown along the x axis.

DATA EXTRACT: Daily Web Server Analysis Server by HOUR

The four chart views in this data extract are similar to the chart views described previously for the data extract: DAILY Web Server Analysis SYSID by HOUR.

The only difference is in individual chart scope. Where in the SYSID by HOUR, each chart represents a SYSID, in the Web Server summary data extract each chart represents a web server.

DATA EXTRACT: Daily Web Server Analysis ReqType Summary

The four chart views in this data extract are similar to the chart views described previously for the data extract: DAILY Web Server Analysis SYSID Summary.

The only difference is in individual chart scope. Where in the SYSID summary, each chart represents a CPC, and individual SYSIDs are shown along the x axis, in the ReqType Summary, each chart represents a web server and the three synchronous request types (HTTP, INTERNAL, MBEAN) are shown along the x-axis.

DATA EXTRACT: Daily Web Server Analysis ReqType by HOUR

The four chart views in this data extract are similar to the chart views described previously for the data extract: DAILY Web Server Analysis SYSID by HOUR.

The only difference is in individual chart scope. Where in the SYSID by HOUR, each chart represents a SYSID, in the ReqType by HOUR data extract, the four chart views represent the following:

- Chart View #1 – Daily Web Server by HOUR ReqType Level Requests Processed: Each chart represents a web server and the number of requests processed, of each type (HTTP, INTERNAL, or MBEAN) are shown for each hour using stacked vertical bars.
- Chart View #2 – #4: Each chart represents a specific synchronous request type (HTTP, INTERNAL, or MBEAN) for an individual web server.

Query Defaults

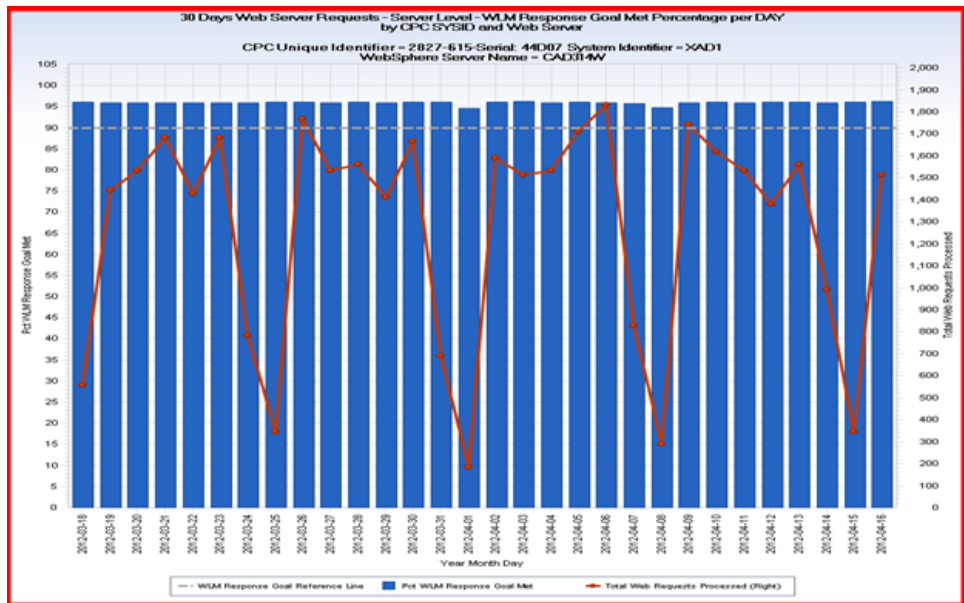
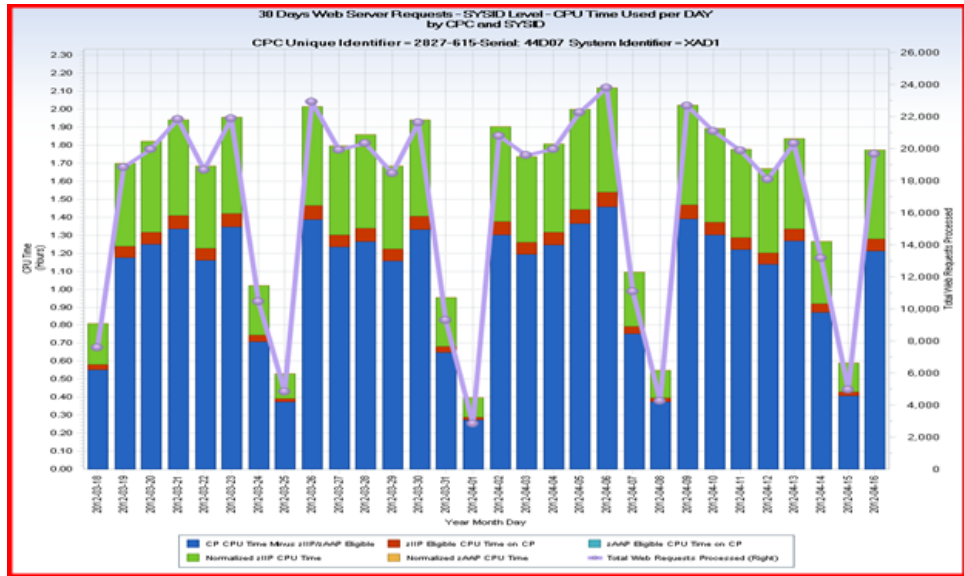
- HARCPU DAYS timespan file: Cycle 01
- WEBWRA DAYS timespan file: Cycles 01 and 02
- A User Code task selects data from yesterday only
- Derivations:
 - DATE
 - DATEPART(ENDTS)
 - CPCUID
 - TRIM(CPUMOD)||'-'||TRIM(CPCMODID)||'-Serial: '||SUBSTR(CPCSEQNB,12,5)

Modifications

You can make the following modifications to the WEBDRA query:

- Add filter statements to the WEBWRA File Selection step to select specific SYSID values, web servers (WEBSRVNM), Request types (WEBRTYPE), or shifts (ZONE).
- Use filter statements in the WEBWRA File Selection step to restrict data to particular time ranges (e.g., HOUR GE 8 AND HOUR LE 17) or particular time zones (e.g., ZONE EQ 1),
- Use the extensive Chart View Edit capabilities to modify Y axis value ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.
- Modify the WLM Goal Met Percentage reference line (default 90%) in the chart views to a value that is meaningful in your environment.

WEBMRA - 30 Day Web Server Request Activity



Overview

The WEBMRA query generates several data extracts that show the 30 most recent days of synchronous z/OS web server request activity at a variety of summarization levels.

At the highest level, a CSV data extract provides charts for each z/OS SYSID, showing 30 days of requests processed, CPU use, percentage of requests that met their WLM response time goal, and average request response time. Another CSV data extract generates similar 30 day charts for each z/OS hosted web server. A final CSV data extract generates 30 day charts showing the same information for each synchronous request type (HTTP, INTERNAL, and MBEAN) executed on each z/OS hosted web server.

This series of CSV data extracts allows an analyst to easily compare day-to-day request volume, resource use, and performance at different levels of granularity (SYSID, Web Server, and Request Type). With thirty days of activity displayed in each chart, anomalous behavior is easily identified.

Three CSV output files are generated by the query:

- 30 Day Web Server Analysis SYSID by DAY (1)
- 30 Day Web Server Analysis Web Server by DAY (2)
- 30 Day Web Server Analysis Request Type by DAY (3)

Each CSV data extract generates the following four chart views:

- 30 Days Web Server Analysis - *granularity* Level Requests Processed
- 30 Days Web Server Analysis - *granularity* Level CPU Use
- 30 Days Web Server Analysis – *granularity* Level WLM Response Goal Met
- 30 Days Web Server Analysis – *granularity* Level Response Time

Where *granularity* is SYSID (CSV–1), Server (CSV–2), and ReqType (CSV–3)

Chart View #1 – Requests Processed

Each chart image generated with view #1 displays 30 days of synchronous requests processed at the SYSID level (CSV data extract 1), web server level (CSV data extract 2), and request type level (CSV data extract 3).

- Each chart generated from the SYSID level (CSV data extract 1) displays the number of requests processed for a z/OS SYSID, and each vertical bar shows the number of requests processed on that LPAR for a day.
- Each chart generated from the Web Server level (CSV data extract 2) displays the requests processed for a z/OS hosted web server, and each vertical bar shows the number of requests processed by that web server for a day.
- Each chart generated from the Request Type (ReqType) Web Server level (CSV data extract 3) displays the requests processed for a z/OS hosted web server, where each vertical bar is stacked, showing the number of requests processed of each type (HTTP, INTERNAL, or MBEAN) by the server for a day.

Chart View #2 - CPU Time

Each chart image generated with view #2 displays 30 days of CPU time use by synchronous web requests at the SYSID level (CSV data extract 1), Web Server level (CSV data extract 2), and Request Type level (CSV data extract 3).

CPU time is broken down by where it was consumed (CP, zIIP, or zAAP processors), and if on a CP processor, the portion, if any that was zIIP or zAAP processor eligible. Additionally, the request count is displayed using a horizontal line and the right y axis.

The level of detail provided by Chart View #2 for each of the three CSV data extracts is similar to that described above for Chart View #1. For the specific information provided with Chart View #2 charts in each CSV data extract, refer to the DATA EXTRACTS section below.

The first chart sample above is an example of Chart View #2, generated by the 30 Day Web Server Analysis SYSID by DAY CSV data extract (1). This chart shows the hourly CPU use for synchronous requests by all web servers hosted on SYSID XAD1 on an IBM 2827-615 (zEC12 CPC with 15 sub-capacity CP engines). You can see that the majority of CPU use for synchronous requests occurs on CP processors (dark blue bar segments). About 1/3rd of CPU use occurs on zIIP processors (green bar segments), and access to zIIP processors should be examined, because the zIIP eligible CPU time executed on CP processor engines in fairly significant (red bar segments).

Chart view #3 – WLM Response Goal Met

Each chart image generated with Chart View #3 displays the percentage of web requests that met or exceeded their WLM response time goal at the SYSID level (CSV data extract 1), the Web Server level (CSV data extract 2), and the Request Type level (CSV data extract 3).

In all cases, the percentage met is shown using vertical bars. The left y axis is scaled from 0 to 105 so that a 100% WLM Response goal met percentage bar height ends near the top of the chart viewing area. Additionally, a horizontal reference line is displayed indicating a 90 percent response goal met value, and the request count is displayed using a horizontal line and the right y axis.

The level of detail provided by Chart View #3 for each of the three CSV data extracts is similar to that described above for Chart View #1. For the specific information provided with Chart View #3 charts in each CSV data extract, refer to the DATA EXTRACTS section below.

The second chart sample above is an example of Chart View #3, generated by the 30 Day Web Server Analysis Web Server by DAY CSV data extract (2). This chart sample shows 30 days of the WLM response time goal met percentage for synchronous transactions processed by web server CAD314W. This server is hosted on SYSID XAD1, a z/OS LPAR running on an IBM 2827-615 (zEC12 CPC with 15 sub-capacity CP engines). You can see that consistently, about 95 percent of the synchronous requests processed by this web server met their WLM response time goal.

Chart View #4 - Response Time

Each chart image generated with Chart View #4 displays the average synchronous request response time broken down into two components; queue time and dispatch time. These response time components are displayed at the SYSID level (CSV data extract 1), the Web Server level (CSV data extract 2), and the Request Type level (CSV data extract 3).

In all cases, the two response time components are shown using stacked vertical bars. Additionally, the request count is displayed using a horizontal line and the right y axis.

The level of detail provided by Chart View #4 for each of the three CSV data extracts is similar to that described above for Chart View #1. For the specific information provided with Chart View #4 charts in each CSV data extract, refer to the DATA EXTRACTS section below.

Uses

This query generates charts showing the most recent 30 days of resource use and performance information for synchronous web requests processed by web servers hosted on z/OS LPARs. At most sites, the use of synchronous requests is far greater than that for asynchronous requests, so it is critical to monitor their quantity, resource use, and performance.

The three CSV data extracts provided by this query provide the analyst with both highly summarized and very detailed insight into z/OS LPAR hosted web server request activity. The SYSID level charts allow 30 days of activity to be viewed for each LPAR with a single chart. If unusual behavior is noted, the more granular level charts can be used to isolate the specific web server or synchronous request type responsible.

For a more detailed analysis, the Daily Web Server Request Activity query (WEBDRA) can be used to closely examine request performance and behavior at the hourly level for specific days where anomalous behavior is noted using this 30 day query.

These charts can be used to evaluate specialty processor (zIIP/zAAP) use.

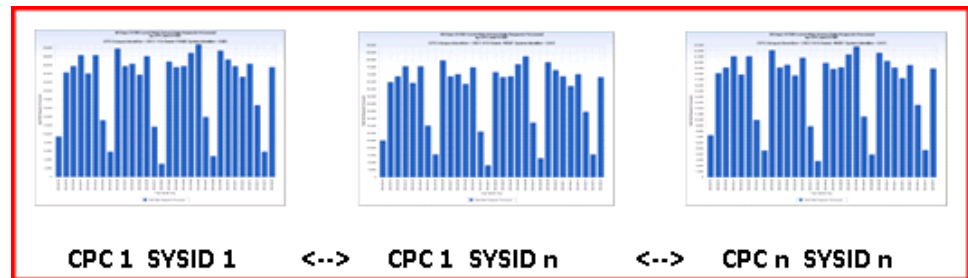
- If excessive CP processor time is being used to process zIIP or zAAP eligible work, it might indicate the need for additional specialty processors.
- If excessive queue time is observed, it might indicate a need for workload balancing. Or specific web servers might need additional resources to help reduce the time waiting for dispatch.

The WLM Response Goal Met charts can assist in fine tuning your WLM Service Policy to ensure that goals and levels of importance are set appropriately.

Data Extracts

DATA EXTRACT: 30 Day Web Activity SYSID by DAY

VIEW 1 of 4: 30 Day Web Server Activity - SYSID Level Requests Processed



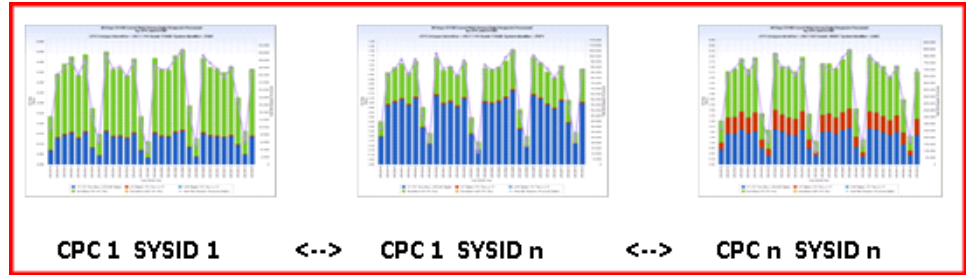
Y axis

WRAREQST – Total Web Requests Processed

X axis

YYMMDD – Year Month Day

VIEW 2 of 4: 30 Day Web Server Activity - SYSID Level CPU Use



Left Y axis

WRAXCSTM – CP CPU Time Minus zIIP/zAAP Eligible

WRASUCTM – zIIP Eligible CPU Time on a CP

WRAZACTM – zAAP Eligible CPU Time on a CP

WRASPNTM – Normalized zIIP CPU Time W

WRAZPNTM – Normalized zAAP CPU Time

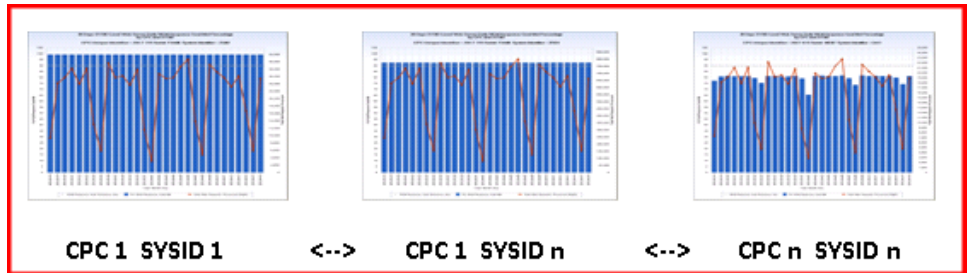
Right Y axis

WRAREQST – Total Web Requests Processed

X axis

YYMMDD – Year Month Day

VIEW 3 of 4: 30 Day Web Server Activity - SYSID Level WLM Response Goal Met



Left Y axis

WRAPCGMT – Pct WLM Response Goal Met

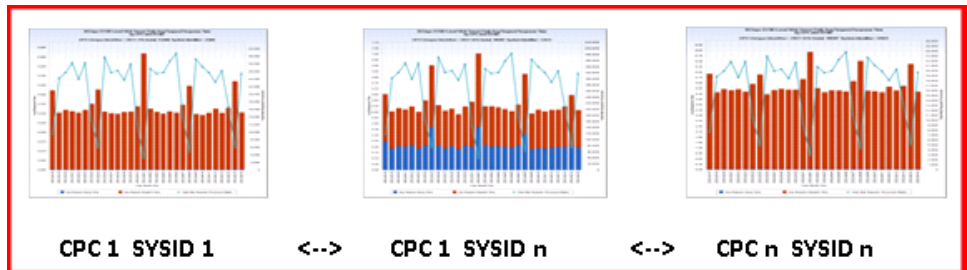
Right Y axis

WRAREQST – Total Web Requests Processed

X axis

YYMMDD – Year Month Day

VIEW 4 of 4: 30 Day Web Server Activity - SYSID Level Response Time



Left Y axis

WRAAVQTM – Avg Request Queue Time

WRAAVDTM – Avg Request Dispatch Time

Right Y axis

WRAREQST – Total Web Requests Processed

X axis

YYMMDD – Year Month Day

DATA EXTRACT: 30 Day Web Activity Web Server by DAY

The four chart views in this data extract are similar to the chart views described previously for the data extract: 30 Day Web Activity SYSID by DAY.

The only difference is in individual chart scope. Where in the SYSID by DAY, each chart represents a SYSID, in the Web Server by DAY data extract, each chart represents a web server.

DATA EXTRACT: 30 Day Web Activity Request Type by DAY

The four chart views in this data extract are similar to the chart views described previously for the data extract: 30 Day Web Activity SYSID by DAY.

The only difference is in individual chart scope. Where in the SYSID summary, each chart represents a SYSID, in the Request Type by DAY data extract, each chart represents the following:

- Chart View #1 – Each chart in the first chart view represents a web server, and the number of requests processed, of each type (HTTP, INTERNAL, or MBEAN) are shown for each day using stacked vertical bars.
- Chart Views #2 – #4: Each chart represents a specific synchronous request type (HTTP, INTERNAL, or MBEAN) for an individual web server.

Query Defaults

- HARCPU DAYS timespan file: Cycle 01
- WEBWRA DAYS timespan file: Cycles 01 – 30
- Derivations:
 - DATE
 - DATEPART(ENDTS)
 - YYYYMMDD
 - PUT(DATE, YEAR4.) || '-' || PUT(MONTH, Z2.) || '-' || PUT(DAY, Z2.)
 - CPCUID
 - TRIM(CPUMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB, 12, 5)

Modifications

The following modifications can be made to the WEBMRA query:

- You can increase or decrease the number of CYCLES input to report on more or fewer than 30 days. If so, make sure you modify the chart titles to reflect your modification.
- You can add filter statements to the WEBWRA File Selection step to select specific SYSID values, web servers (WEBSRVNM), Request types (WEBRTYPE).
- You can use filter statements in the WEBWRA File Selection step to restrict data to particular time ranges (for example, HOUR GE 8 AND HOUR LE 17) or particular time zones (for example, ZONE EQ 1),
- You can use the extensive Chart View Edit capabilities to modify Y axis value ranges to better visualize data ranges found in your environment, and to change chart visualization choices such as line thickness and color schemes.
- You can modify the WLM Goal Met Percentage reference line (default 90%) in the chart views to a value that is meaningful in your environment.

Chapter 11: CA MICS Space Analyzer Option (VCA) Queries

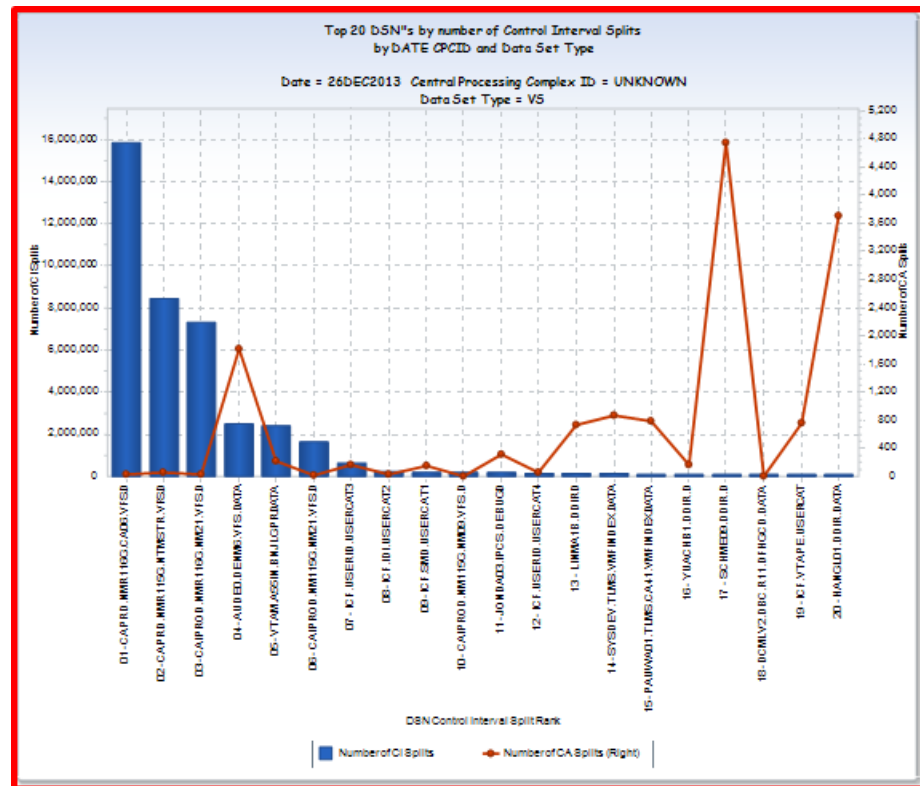
This section contains the following topics:

[VCACIS– Top 20 CI Splits by DS Type](#) (see page 425)

[VCAT2C– VCA Daily top 20 STORCLAS Space Used](#) (see page 429)

[VCAVOL – Volume Capacity and Use](#) (see page 432)

VCACIS– Top 20 CI Splits by DS Type



Overview

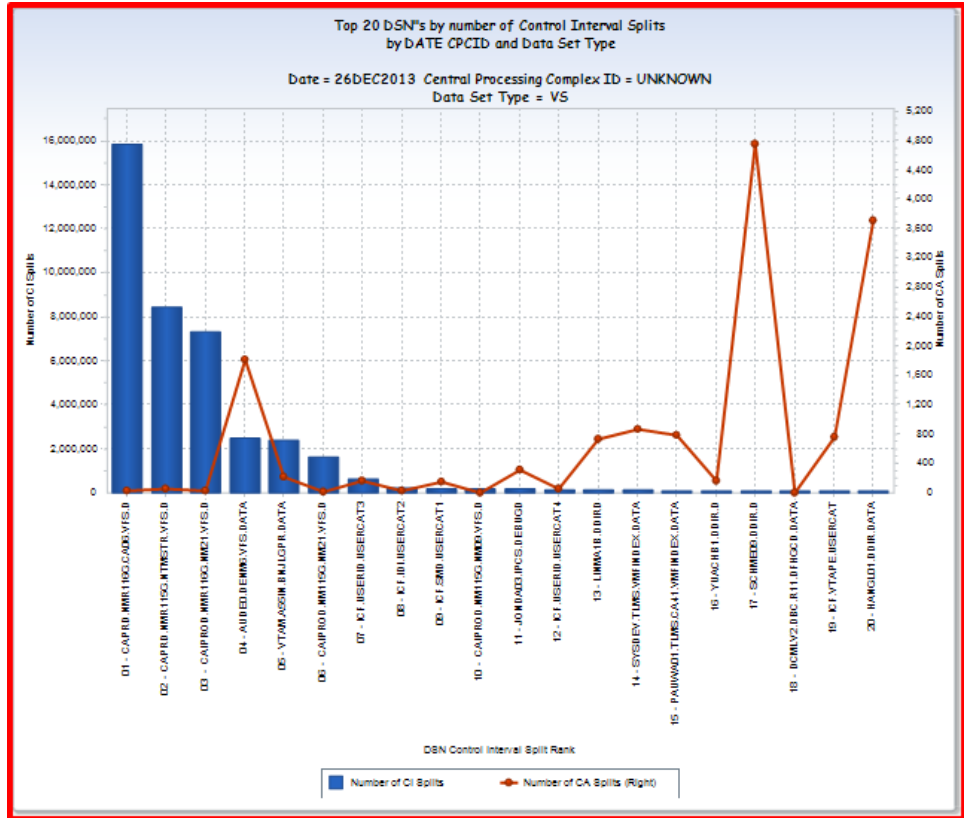
The VCACIS query produces a daily Pareto style chart showing the top 20 VSAM data sets with the largest number of control interval splits in descending order, left to right. By default, the query shows the top 20 VSAM data sets. The query can be easily modified to show a different number of data sets.

Uses

This query is useful to identify VSAM data sets that might need REORG maintenance to eliminate the volume of control interval splits. Control interval splits can degrade read/write performance when accessing and updating the data set.

Data Extracts

Top 20 DSN CI Splits by DS Type



Left Y-axis

DAANCIS
Number of CI Splits

Right Y-axis

DAANCAS
Number of CA Splits

X-axis

DSN_CIRK
DSN Control Interval Split Rank

Query Defaults

The query defaults are:

- VCA_VS DETAIL timespan file: Cycle 01
- Filters:
DAANCIS GE 30
- Derivations:
DATE

DATEPART(ENDTS)
TOPNN

%LET TOPNN = 20 ; /* <----- change if desired */
DSN_CIRK

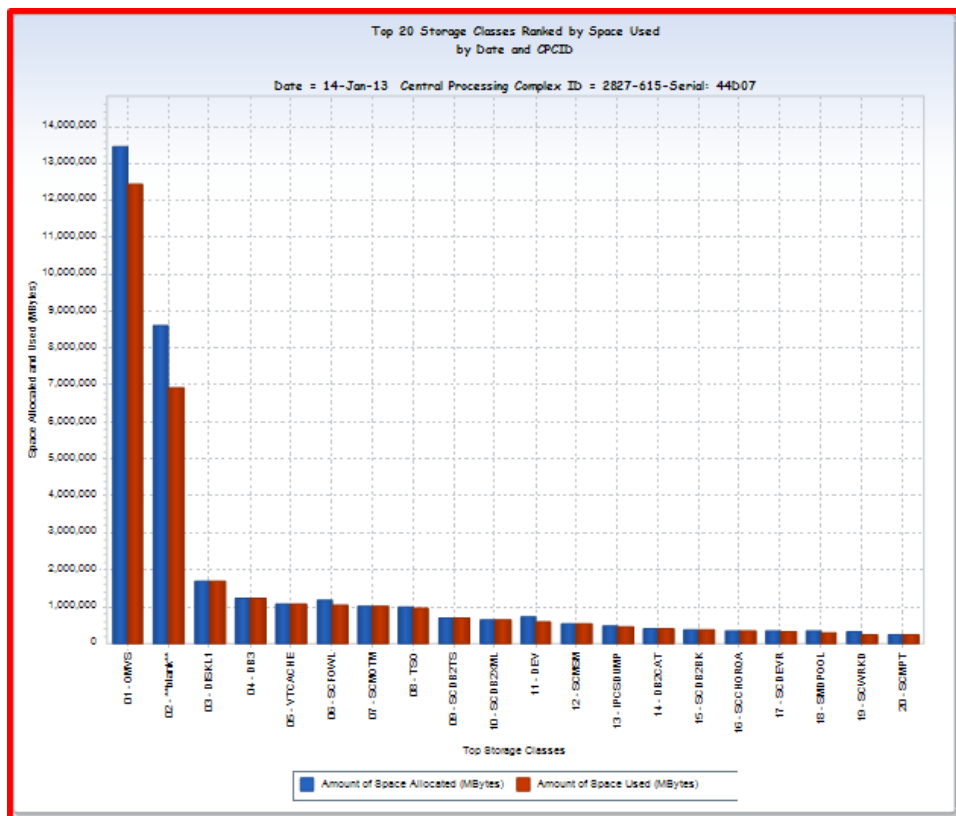
DSN_CIRK = DSNAME ;
/* after input file is sorted by descending DAASCIS the rank is prefixed to
the DSNAME and the TOPNN are output */
RETAIN TOPNN &TOPNN ;
RETAIN COUNT 0 ;
IF FIRST.CPCID THEN COUNT=1 ;
IF COUNT LE TOPNN THEN DO ;
DSN_CIRK=PUT(COUNT,Z2.) || ' - ' || LEFT(TRIM(DSN_CIRK)) ;
OUTPUT FILE2 ;
COUNT+1 ;
END ;

Modifications

The following modifications can be made to the VCACIS query:

- You can add filter statements in the 'INPUT VCA_VS File' step focus results other specific data set criteria.
- You can change the number of data sets displayed (default 30) by modifying the TOPNN parameter in the first step of the query.

VCAT2C- VCA Daily top 20 STORCLAS Space Used



Overview

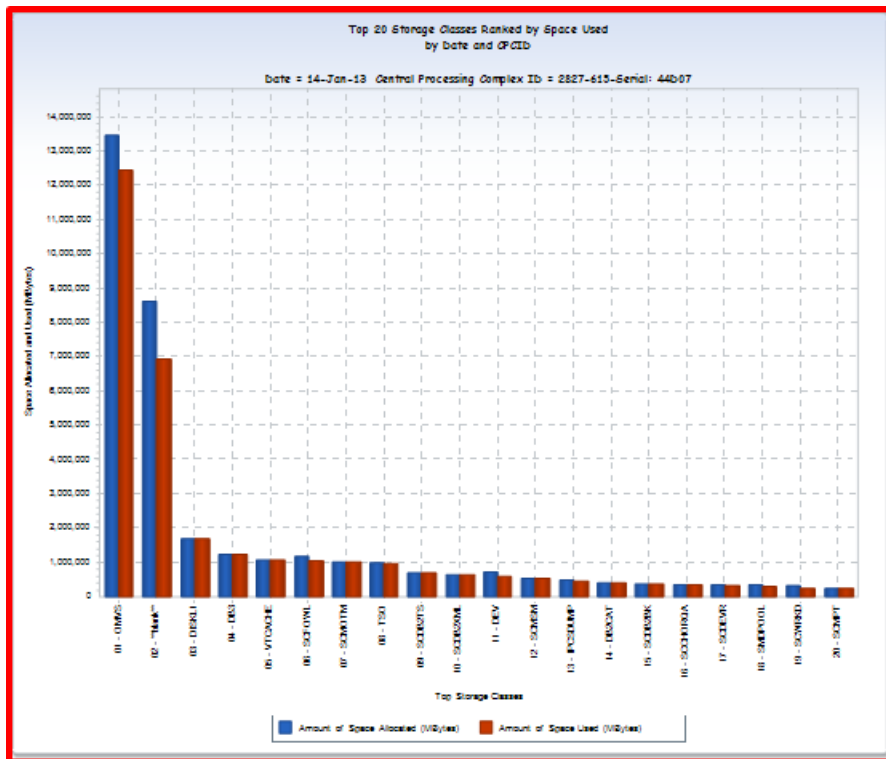
The VCAT2C query generates a daily Pareto style chart identifying the top 20 Storage Classes that are based on DASD space that is allocated in descending order, left to right.

Uses

This query is useful in helping you to identify the largest users of DASD space, which is based on your storage class definitions.

Data Extracts

Top 20 Storage classes Ranked by Spaced Used



Left Y-axis

DAAALLSP

Amount of Space Allocated (Mbytes)

DAAUSESP

Amount of Space Used (Mbytes)

X-axis

STCRANK

Top Storage Classes

Query Defaults

The query defaults are:

- VCADAA DETAIL timespan file: Cycle 01
- Derivations:

DATE

DATEPART(ENDTS)

TOPNN

```
%LET TOPNN = 20 ;      /* <----- change if desired */
```

STCRANK

```
STCRANK = STORSTGC ;
```

```
IF STCRANK=' ' THEN STCRANK='**blank**';
```

```
/* after descending sort by space used, the top 20 are selected for charting
*/
```

```
RETAIN TOPNN &TOPNN ;
```

```
RETAIN COUNT 0 ;
```

```
IF FIRST.CPCID THEN COUNT=1 ;
```

```
IF COUNT LE TOPNN THEN DO ;
```

```
STCRANK=PUT(COUNT,Z2.) || ' - ' || LEFT(TRIM(STCRANK)) ;
```

```
OUTPUT FILE3 ;
```

```
COUNT+1 ;
```

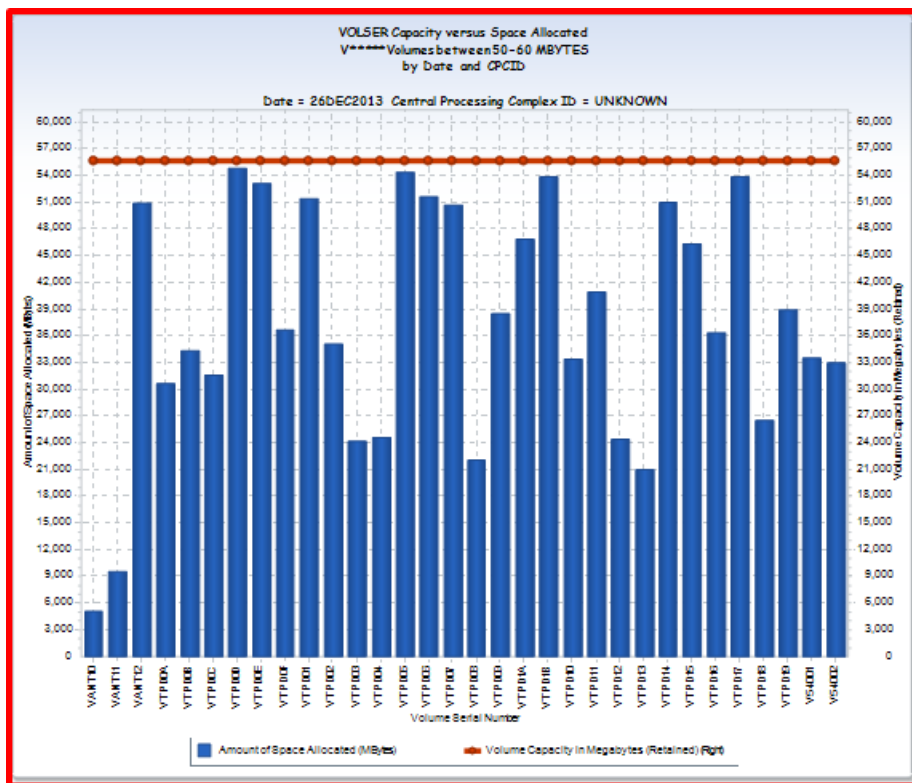
```
END ;
```

Modifications

The following modifications can be made to the VCAT2C query:

- You can add filter statements to select or reject storage volumes or classes as needed for analytical study.
- You can change the number of storage classes displayed (default 20) by modifying the TOPNN parameter in the first step of the query.

VCAVOL – Volume Capacity and Use



Overview

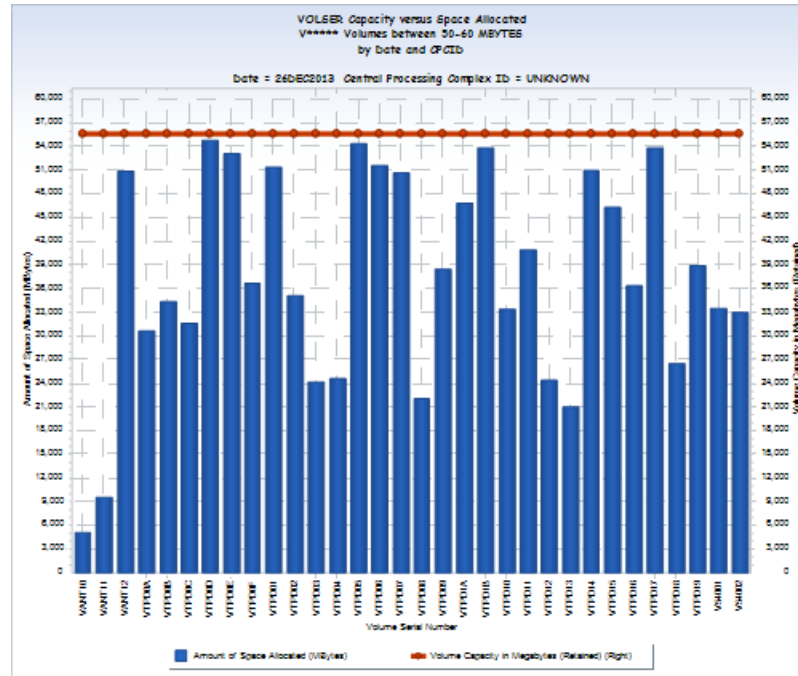
The VCAVOL query generates a chart showing the capacity and space that is allocated for DASD volumes you specify using query that is specified filtering criteria. The previous example shows the capacity and allocation status of DASD volumes with VOLSER names beginning with the letter 'V'.

Uses

This query is useful for studying the allocation patterns of specific volume subsets. Multiple queries can be run with different filter criteria, allowing you to generate charts for various specific DASD volume subsets.

Data Extracts

VOLSER Capacity compared to Space Allocated



Left-axis

VOAALLSP

Amount of Space Allocated (Mbytes)

Right-axis

VOACAPMB

Volume Capacity (Mbytes)

X-axis

Volser

Volume Serial Number

Query Defaults

The query defaults are:

- VCAVOA DETAIL timespan file: Cycle 01
- Filters:
 - VOLSER EQ 'V'**
 - VOACAPMB GT 50000 and VOACAPMB LT 60000**
- Derivations:
 - DATE
 - DATEPART (ENDTS)

Modifications

The following modifications can be made to the VCAVOL query:

- You can add or modify filter statements to select specific VOLSER ranges, storage capacity, or device types.

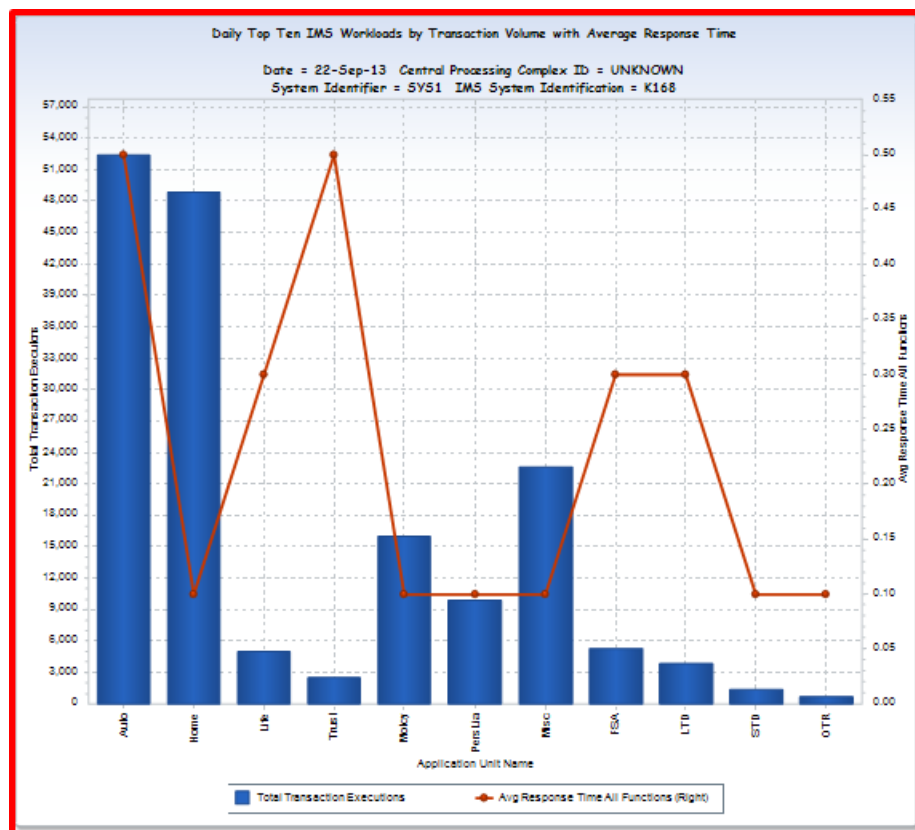
Chapter 12: CA MICS Analyzer Option for IMS (IMS) Queries

This section contains the following topics:

[IMSDAT – IMS Daily Top 10 Workloads](#) (see page 435)

[IMSDTT – IMS Daily Top 10 Transactions](#) (see page 442)

IMSDAT – IMS Daily Top 10 Workloads



Overview

The IMSDAT IMS Daily Top 10 Workload Analysis by Region query generates a chart for each SYSID and IMS Region combination. The query uses the DAYS timespan IMS Application Unit Activity file as input, and reports on yesterday's top ten user-defined IMS Application Unit IDs (IMSAPU). The IMSAPU data element provides a means to categorize IMS transactions as belonging to specific groups or workloads. The IMSIAU file uses IMSAPU as a summarization key. The IMSDAT query selects the top nn (default top 10) IMSAPUs for each IMS region. The top ten are selected based on the highest number of IMS transactions. The top ten IMSAPUs are graphically charted using two separate output views.

Note: The IMSAPU routine which is used in the CA MICS unit that is used to generate these charts, assigned TRANCODE to IMSAPU. The resulting charts are actually showing the top ten transactions rather than the top ten workloads.

The first view shows the overall transaction count for the top ten IMSAPUs using vertical bars, and the average IMS transaction response time for each IMSAPU with a horizontal line.

The example shows the chart that is generated for an IMS region "K168" executing on system "SYS1". Merging with data from the HARCPU file was not able to determine the Central Processing Complex (CPC) ID, hence the CPC ID is 'UNKNOWN.' The first chart shows that the top two workloads, by transaction count, were "Auto" with 52,793 transactions and "Home" with 48,575 transactions. The average transaction response time was 0.50 seconds for workload "Auto" and 0.1 seconds for workload "Home."

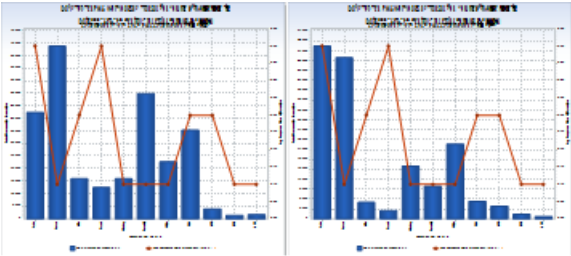
Uses

This query is useful to for analyzing workload-related transaction performance characteristics. By selecting the top ten workloads which are based on transaction volume, you can focus your analysis on the workloads consuming the most IMS resources. Where transactions show significant percentages of time waiting on I/O or other delaying factors, further analysis may direct tuning efforts to improve overall response time.

Data Extracts

IMSDAT Daily Top 10 Response Time Avg

Figure 1: Shows the IMSDAT Daily Top 10 Response Time Avg



CPC 1 IMS 1 <---> CPC nn IMS nn

Left Y-axis

IAUTRANS

Transactions Processed

Right Y-axis

IAUAVTTM

Avg Response Time

X-axis

IMSAPU

IMS Application Unit ID

Query Defaults

- IMSIAU DAYS timespan file: Cycle 01
- Derivations:
 - DATE
 - DATEPART(ENDTS)
 - CPCID
 - CPCID = TRIM(CPUMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' || SUBSTR(CPCSEQNB,12,5) ;

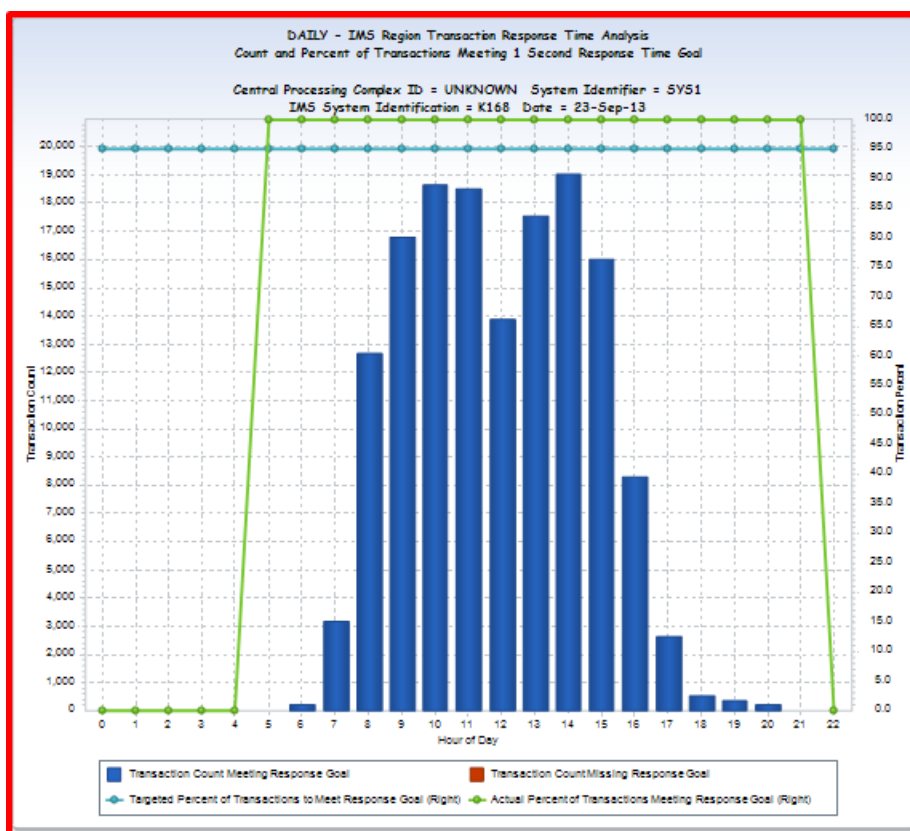
Modifications

The following modifications can be made to the query:

- You can adjust the TOPnn value to something other than the default of 10. You can add filter statements to select only a single SYSID or IMSID, or select only specific production IMS regions.
- You can increase the cycles input to create charts for multiple days, but comment out the user code in the IMSAPU file input step that restricts processing to yesterday's data.
- Apply a format to the IMSAPU variable to convert it from the 12 character workload identifier to a more meaningful workload description.

Note: If you change the TOPnn value (default TOP 10), modify the query chart title to reflect the changed value.

IMSDTR – IMS Daily Tran Count and Response Pct



Overview

The IMSDTR query generates daily charts showing overall transaction counts and response time metrics for individual IMS regions. A separate chart is produced for each z/OS SYSID and IMS region combination. The charts show the total transaction count for each hour using a stacked vertical bar—where the overall transaction count is broken down into stacked-bar segments. One segment shows the transaction count that met a user-specified elapsed time goal using a blue bar (default one second), and the other segment shows the transaction count that missed the goal using a red bar.

In addition, the chart uses a horizontal line to show the percentage of transactions, per hour, that met the response time goal (green line). A user-specified desired percentage (default 95 percent) is plotted (blue line) to allow for easy visualization of the actual percentage of transactions which are compared to the desired percentage of transactions, that met the response time goal.

The example shows the chart that is generated for an IMS region "K168" running on a z/OS system "K168". This chart was produced from a CA MICS database that is built from IMS transaction records where we had not collected the associated RMF data. The Central Processing Complex ID (CPCID) was not available and set to "UNKNOWN." At a normal customer installation, one of the query steps examines the CA MICS RMF component HARCPU file to determine the CPCID for each z/OS system (SYSID) where the IMS regions executed. The CPCID is assigned a value like "2097-712-Serial AE000'.

The chart shows that transaction volume begins picking up at around 8:00 AM, peaks between 10 AM and 3:00 PM, and then begins tapering off over the rest of the day. Close to 100 percent of the transactions, each hour, met the response time goal of completing in one second or less.

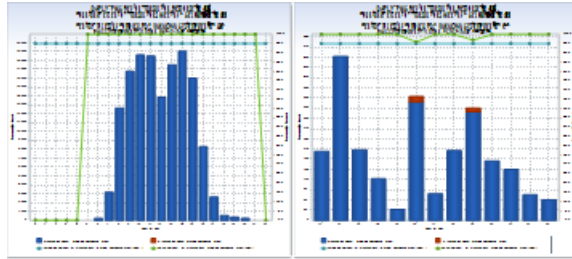
The second line of the title shows the response time target—and must be manually updated in the query output view definition if the response time target is changed from the default value of one (1) second.

Uses

This query allows you to visualize easily the overall IMS transaction performance at the IM region level. If you have established specific Service Level Agreements (SLAs) for the IMS regions where nn percent of IMS transactions must complete in mm seconds or less, query parameters can be modified so that these charts provide fast visual assessment of actual performance that is compared to required or expected performance.

Data Extracts

IMS Daily Transaction Response Goal Report



CPC 1 IMS 1 <---> CPC nn IMS nn

Left Y-axis

MEET_CNT: Transaction Count Meeting Response Goal

MISS_CNT: Transaction Count Missing Response Goal

Right Y-axis

GOAL_PCT: Targeted Percent of Transactions to Meet Response Goal

MEETL_PCT: Actual Percent of Transactions Meeting Response Goal

X-axis

Hour of Day

Query Defaults

- IMSISY DAYS timespan file: CYCLE 01
- HARCPU DAYS timespan file: CYCLE 01 (Used to determine CPCID<->SYSID mapping)
- User code selects yesterday's data (IF DATEPART(ENDTS) EQ TODAY()-1 ;
- User code defines response goal targets:
 - Response time goal: One (1) second – MACRO VAR &RESPVAL
 - Pct of transactions to meet Response Time Goal: 95% - MACRO VAR GOALPCT

Derivations:

DATE

DATEPART(ENDTS)

CPCID

```
CPCID = TRIM(CPUMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' ||
SUBSTR(CPCSEQNB,12,5) ;
```

MEET_CNT

```
if imsrval7 <= &RESPVAL then do;
  MEET_CNT=sum(ISYTDST7,ISYTDST6,ISYTDST5,ISYTDST4,
    ISYTDST3,ISYTDST2,ISYTDST1);
end;
else if imsrval6 <= &RESPVAL then do;
  MEET_CNT=sum(ISYTDST6,ISYTDST5,ISYTDST4,ISYTDST3,
    ISYTDST2,ISYTDST1);
end;
else if imsrval5 <= &RESPVAL then do;
  MEET_CNT = sum(ISYTDST5,ISYTDST4,ISYTDST3,ISYTDST2,ISYTDST1);
end;
else if imsrval4 <= &RESPVAL then do;
  MEET_CNT = sum(ISYTDST4,ISYTDST3,ISYTDST2,ISYTDST1);
end;
else if imsrval3 <= &RESPVAL then do;
  MEET_CNT = sum(ISYTDST3,ISYTDST2,ISYTDST1);
end;
else if imsrval2 <= &RESPVAL then do;
  MEET_CNT = sum(ISYTDST2,ISYTDST1);
end;
else if imsrval1 <= &RESPVAL then MEET_CNT = ISYTDST1;
```

MISS_CNT

```
= SUM(ISYTRANS, -MEET_CNT) ;
```

GOAL_PCT

```
= &GOALPCT ;
```

MEET_PCT

$$= (\text{MEET_CNT} / \text{ISYTRANS}) * 100$$

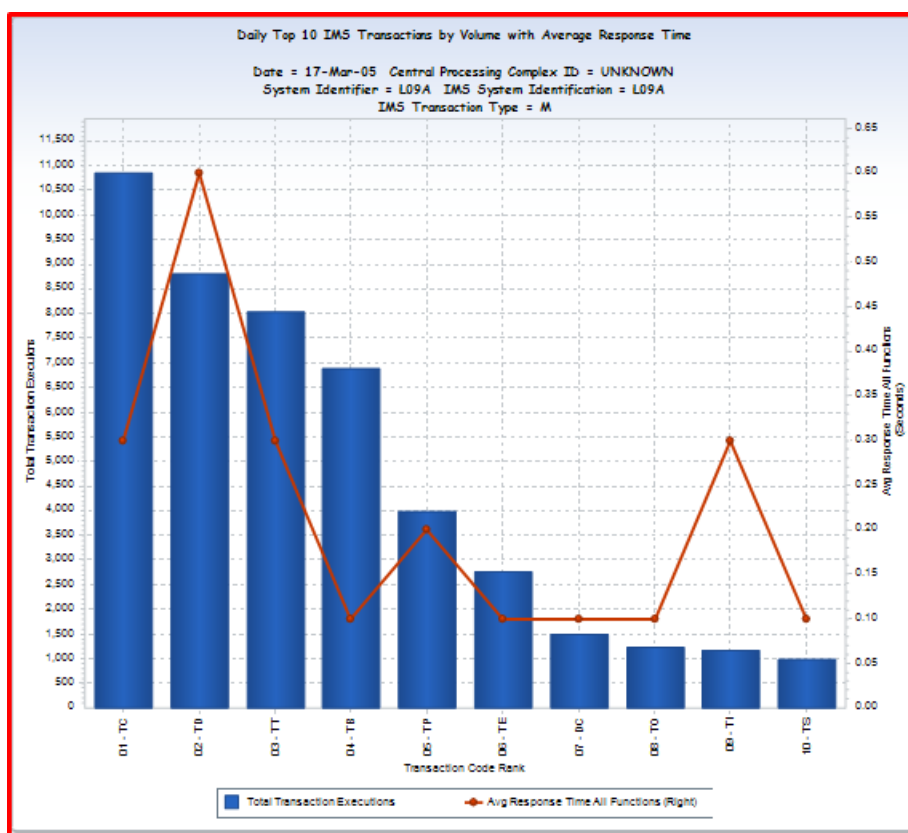
Modifications

The following modifications can be made to the query:

- To select specific SYSIDs or IMS regions (IMSDIDs), add a Data Filter to the Input IMSCSY File step
- To modify the response time goal (default 1 second) or percent of transactions to meet response time goal (default 95 percent), update the SAS code in the Query Control Vars step.

Note: If you change the response time goal (default 1 second), modify the second line of the query chart title to reflect the changed value. The default value of 1 second is hardcoded, and you can modify it using the Edit View facility when viewing the charts.

IMSDTT – IMS Daily Top 10 Transactions



Overview

The IMSDTT query generates a Pareto style chart for each SYSID and IMS region combination. The query uses the DAYS timespan IMS User Activity file as input, and reports on yesterday's top ten transactions for each IMS region. This query requires that one of the CA MICS IMSACTnn user-defined account codes is set equal to TRANCODE. Before you run the query, update the query TRANCODE derivation to use the IMSACTnn value that is assigned to TRANCODE in your IMSACRT Account Routine.

The IMSDTT query selects the top nn (default top 10) transactions, by volume, for each IMS region. The top ten transactions are graphically charted using two separate output views.

The first view shows the top ten transactions daily volume using vertical bars, and the average IMS transaction response time for each transaction with a horizontal line.

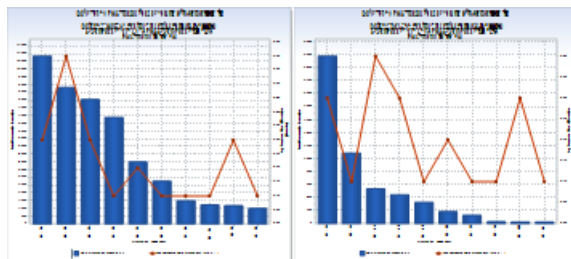
The example shows the charts that are generated for an IMS region "L09A" executing on system "SYS1". Merging with data from the HARCPU file was not able to determine the Central Processing Complex (CPC) ID, hence the CPC ID is 'UNKNOWN.' The first chart shows that the most frequently run transaction is "TC" with 10,854 executed with an average transaction response time of 0.14 seconds.

Uses

This query is useful to for analyzing transaction volume and performance characteristics. By selecting the top ten transactions, which are based on transaction volume, you can focus your analysis on the transactions that are most frequently executed by each IMS region.

Data Extracts

IMS Daily Top 10 Trans Resp Time Average



CPC 1 IMS 1 <---> CPC nn IMS nn

Left Y-axis

ISUTRANS

Transactions Processed

Right Y-axis

ISUAVTTM

Avg Response Time

X-axis

TRAN_RK

IMS Transaction Code Rank

Query Defaults

- IMSISU DAYS timespan file: Cycle 01

- Derivations:

DATE

DATEPART(ENDTS) ;

TRANCODE

TRANCODE=IMSACT2 ; ß Must modify to match your settings!

CPCID

CPCID = TRIM(CPUMOD) || '-' || TRIM(CPCMODID) || '-Serial: ' ||
SUBSTR(CPCSEQNB,12,5) ;

Modifications

The following modifications can be made to the query:

- You can adjust the TOPnn value to something other than the default of 10.
- You can add filter statements to select only a single SYSID or IMSID, or select only specific production IMS regions.
- You can increase the cycles input to create charts for multiple days, but comment out the user code in the IMSISU file input step that restricts processing to yesterday's data.

Note: If you change the TOPnn value (default TOP 10), modify the query chart title to reflect the changed value.