z/OS Performance Spotlight – Some Top Things You May Not Know

aka Peter and Scott's Tips and Tidbits

z/OS Performance Education, Software, and Managed Service Providers

Creators of Pivotor®

Peter Enrico & Scott Chapman

Enterprise Performance Strategies, Inc.

performance.questions@epstrategies.com

Abstract

- During this session, Peter Enrico and Scott Chapman will discuss a variety of z/OS performance measurement, analysis, and tuning techniques that may not be commonly known or are not often discussed.

- The key objective of this presentation is to provide the attendee with information they can bring back to their shop and conduct some analysis or tuning exercises. A secondary objective of this session is to help the attendee learn more about the z/OS environment, and how things work. This session is sure to be highly educational!

Contact, Copyright, and Trademarks

Questions?

Send email to performance.questions@EPStrategies.com, or visit our website at https://www.epstrategies.com or http://www.pivotor.com.

Copyright Notice:

Trademarks:

Enterprise Performance Strategies, Inc. presentation materials contain trademarks and registered trademarks of several companies.

The following are trademarks of Enterprise Performance Strategies, Inc.: **Health Check®, Reductions®, Pivotor®**

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries: IBM®, z/OS®, zSeries®, WebSphere®, CICS®, DB2®, S390®, WebSphere Application Server®, and many others.

Other trademarks and registered trademarks may exist in this presentation

Today's Agenda

- Who we are / what we do (Peter)
- Emerging Areas of Interest (we will get through this list as best as possible)
  - Some questions we have been getting lately
  - z16 Migrations (Scott)
  - CPENABLE and z/OS 3.1 (Peter)
  - Implicit CPU Protection in z/OS 3.1 (Peter)
  - New enclave measurements in response to DB2 HiPerf DBATs
  - First Reference Page Faults (Peter)
  - IXGCNFxx Keep local buffers (Peter)
  - SDC Coefficients and Reevaluating Durations for z/OS 2.5 (Peter)
- Short Reminders from Continuing Questions and Opportunities (not presented)
  - Large memory should mean less I/O? (Scott)
  - Scott's current AI thoughts
  - XCF transport class simplification (Peter)
  - SRB Update and SMF 30 data (Scott)
  - SuperPAV (Scott)
  - I/O Priority Management (Scott)
  - Record the 98s and 99s (Scott)
  - SMT (Scott)
- Prize drawings! (Jamie)

EPS: We do z/OS performance…

- We are z/OS performance!

- Pivotor
  - Performance reporting and analysis of your z/OS measurements
  - Example: SMF, DCOLLECT, other, etc.
  - Not just reporting, but cost-effective analysis-based reporting based on our expertise

- Performance Educational Workshops (while analyzing your own data)
  - Essential z/OS Performance Tuning
  - Parallel Sysplex and z/OS Performance Tuning
  - WLM Performance and Re-evaluating Goals

- Performance War Rooms
  - Concentrated, highly productive group discussions and analysis

- MSU reductions
  - Application and MSU reduction

z/OS Performance workshops available

During these workshops you will be analyzing your own data!

- WLM Performance and Re-evaluating Goals
  - February 19-23, 2024
- Parallel Sysplex and z/OS Performance Tuning
  - August 20-21, 2024
- Essential z/OS Performance Tuning
  - October 7-11, 2024
- Also… please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

Like what you hear today?

- Free z/OS Performance Educational webinars!
  - Have been on hiatus for a couple of months but should be coming back soon
  - Let us know if you want to be on our mailing list for these webinars

- If you want a free cursory review of your environment, let us know!
  - We're always happy to process a day's worth of data and show you the results
  - See also: http://pivotor.com/cursoryReview.html

EPS presentations this week

What	Who	When	Where
60 Years of Pushing Performance Boundaries with the Mainframe	Scott Chapman	Sun 17:00	Neptune D
Introduction to Parallel Sysplex and Data Sharing	Peter Enrico	Mon 13:15	Pomona
Macro to Micro: Understanding z/OS Performance Moment by Moment	Scott Chapman	Mon 15:45	Neptune D
WLM Turns 30! : A Retrospective and Lessons Learned	Peter Enrico	Tue 10:30	Neptune D
PSP: z/OS Performance Spotlight: Some Top Things You May Not Know	Peter Enrico Scott Chapman	Tue 13:00	Pomona
More/Slower vs. Fewer/Faster CPUs: Practical Considerations in 2024	Scott Chapman	Tue 14:15	Neptune D
z16 SMF 113s – Understanding Processor Cache Counters	Peter Enrico	Wed 13:15	Pomona

Like what you see?

- Free z/OS Performance Educational webinars!
  - The titles for our Summer / Fall 2024 webinars are as follows:
    - ✓ *What a z/OS Guy Learned About AWS in 10 Years*
    - ✓ *Advantages of Multiple Period Service Classes*
    - ✓ *Understanding z/OS Connect Measurements*
    - *WLM and SMF 99.1 – System Measurements Deeper Dive*
    - *WLM and SMF 99.2 – Service Class Period Measurements Deeper Dive*
    - *Optimizing Performance at the Speed of Light: Why I/O Avoidance is Even More Important Today*
    - *Understanding MVS Busy % versus LPAR Busy % versus Physical Busy %*
    - *Rethinking IBM Software Cost Management Under Tailored Fit Pricing*
    - *Understanding Page Faults and Their Influence on Uncaptured Time*
    - *Response Time Goals: Average or Percentiles?*
    - *Understanding and Using Enclave*

- If you want a free cursory review of your environment, let us know!
  - We're always happy to process a day's worth of data and show you the results
  - See also: http://pivotor.com/cursoryReview.html

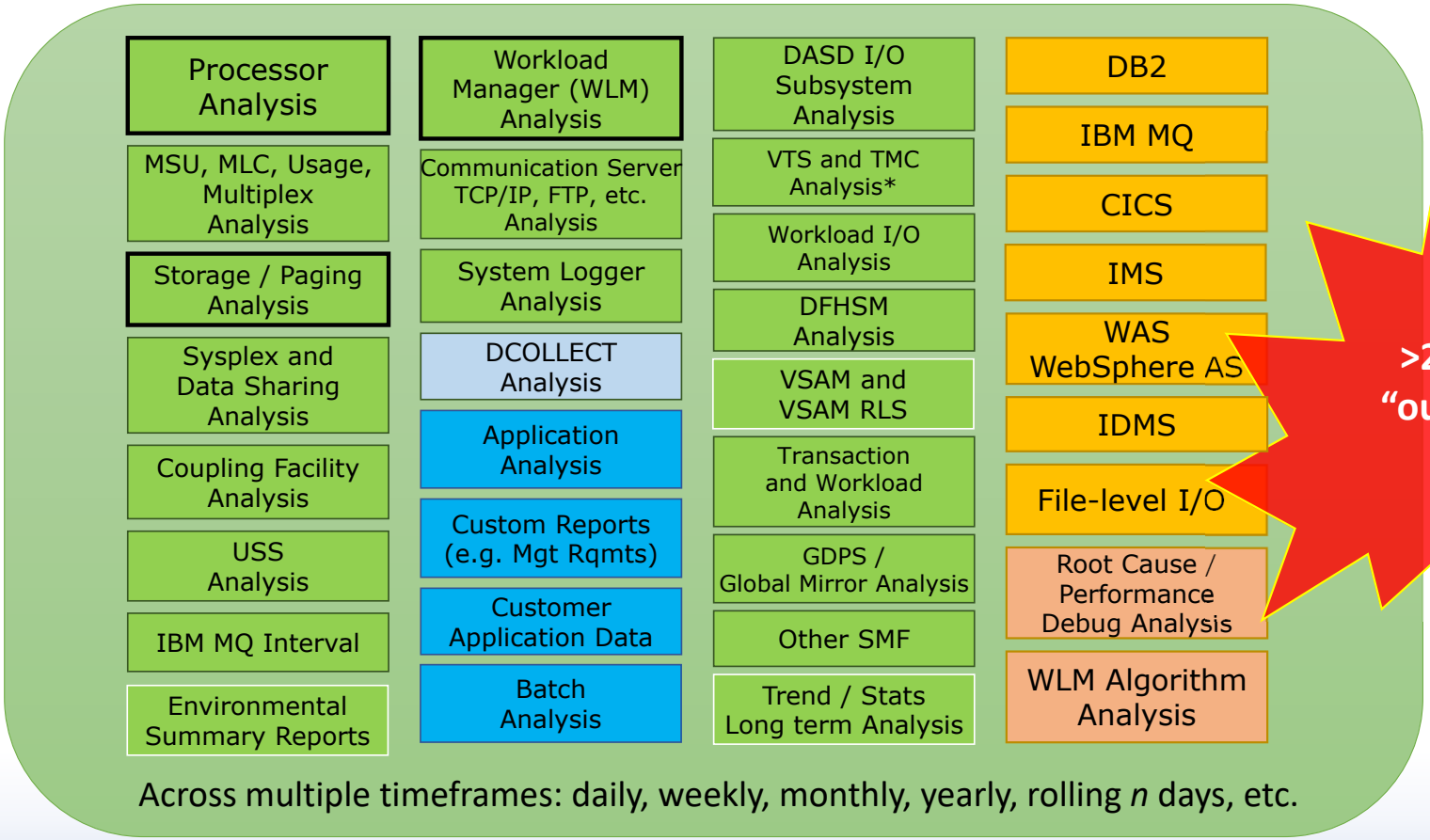
Like what you see?

- The z/OS Performance Graphs you see here come from Pivotor™
- If you just a free cursory review of your environment, let us know!
  - We're always happy to process a day's worth of data and show you the results
  - See also: http://pivotor.com/cursoryReview.html
- We also have a free Pivotor offering available as well
  - 1 System, SMF 70-72 only, 7 Day retention
  - That still encompasses over 100 reports!

Pivotor – Intelligent Reporting

- Pivotor is our data reporting tool & service designed specifically for z/OS performance reporting
  - Designed and used by z/OS performance experts
  - Processes data from SMF, DCOLLECT, and customer sources
  - Contains hundreds of z/OS performance reports “out of the box”
  - Designed to be easy to use and manage
  - Reports are organized into logical and searchable report sets
  - Features include intelligent exceptions, drill down, search, canned analysis, and so much more
  - Built in expanded helps to help foster education

Comprehensive Report Sets for Immediate Performance Analysis

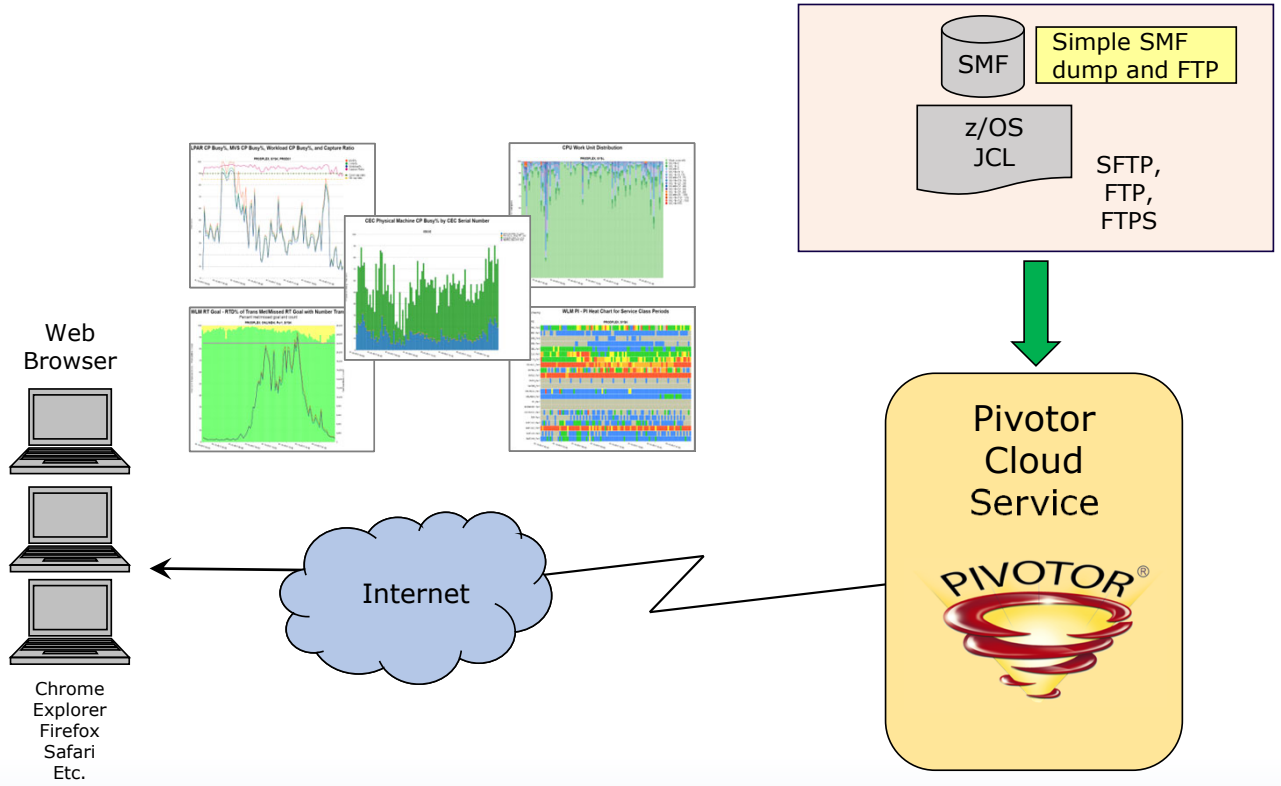


Pivotor Software as a Solution (SaaS)

- Pivotor is offered as both a SaaS or local install
- When SaaS:

SaaS Includes:

- Formal yearly cursory review / discussion
- Ability to ask us performance questions, or for us to look at a particular problem or concern. (support@epstrategies.com)
- We can occasionally look in on your data and performance
- We can participate in performance debug with IBM, or other vendors

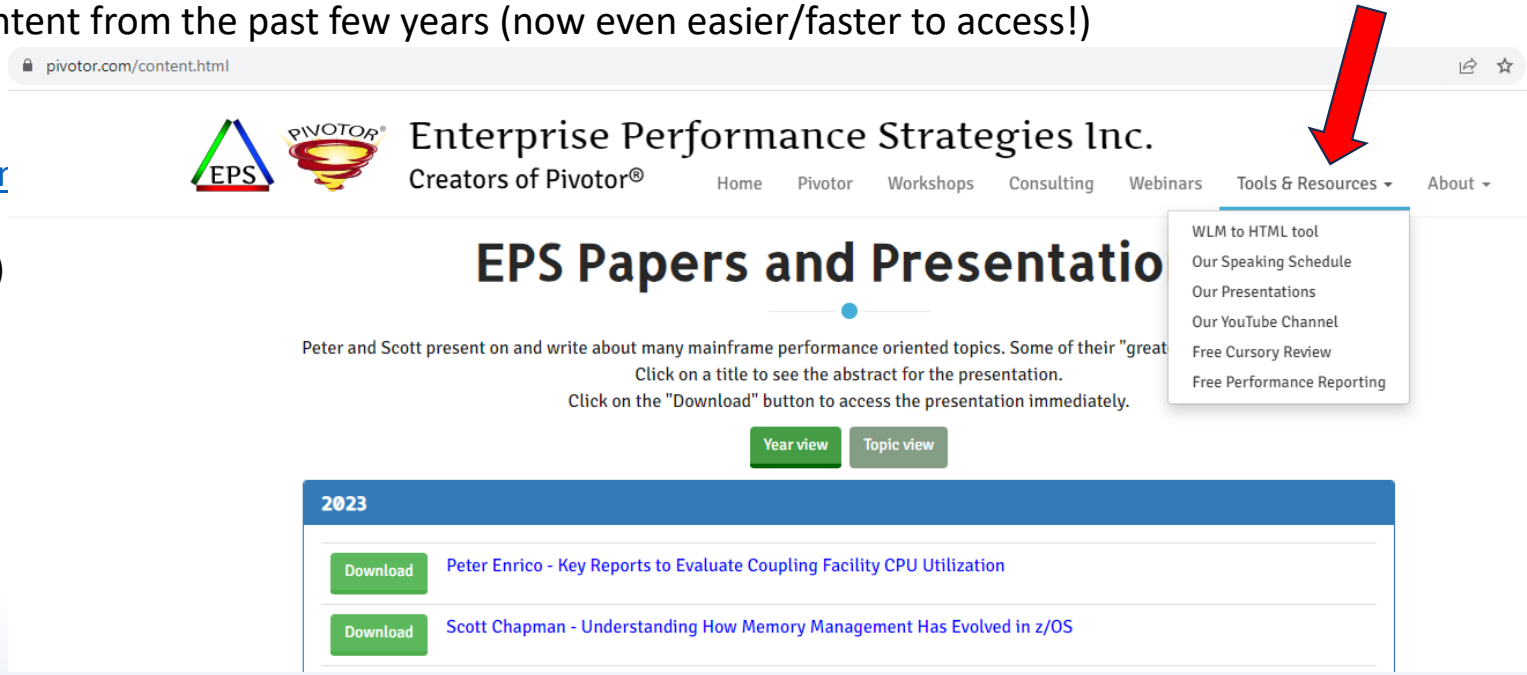


- Pivotor pricing is clear and affordable

More Free Things!

- On our web site click on Tools & Resources to access:
  - WLM to HTML Tool
    - Get your WLM policy in a useful and usable HTML format
  - Our Presentations
    - Lots of great content from the past few years (now even easier/faster to access!)

https://www.epstrategies.com
https://www.pivotor.com/
(Same site behind both URLs)



Questions we get…

CPU Critical

Q Is this importance *n* service class suffering because other service classes at the same importance have CPU critical set?

A Not directly: CPU critical only ensures that the service class will never have a dispatching priority below a lower importance service class

- Of course, CPU critical might be elevating an undeserving SCs above both lower importance and equal importance workloads and while adding CPU Crit to the equal importance SCs might fix it, the real issue might be the original CPU Crit

Q Should I add CPU Critical to this service class?

A Maybe, probably not. Primarily of use where work absolutely cannot wait for the 10 second WLM policy intervals to react to changing conditions

LPAR Management Time

Q Why does the "Physical" CPU consumption seem "high" or higher than it was before?

A Not sure, but probably related to having a large number of medium and low pool processors relative to the number of physicals not reserved to highs, especially if/when those lows are busy.

Background

- The CPU consumption attributed to "Physical" is PR/SM LPAR management overhead that can't be tied directly to a particular LPAR.
- Generally expect this to be less than 1% of the machine, but have recently seen examples where it's higher
  - Is 1% really the right metric? 1% of a 508 is very different than 1% of a 724
- Considering/investigating some new metrics and reporting around a modern logical to physical analysis

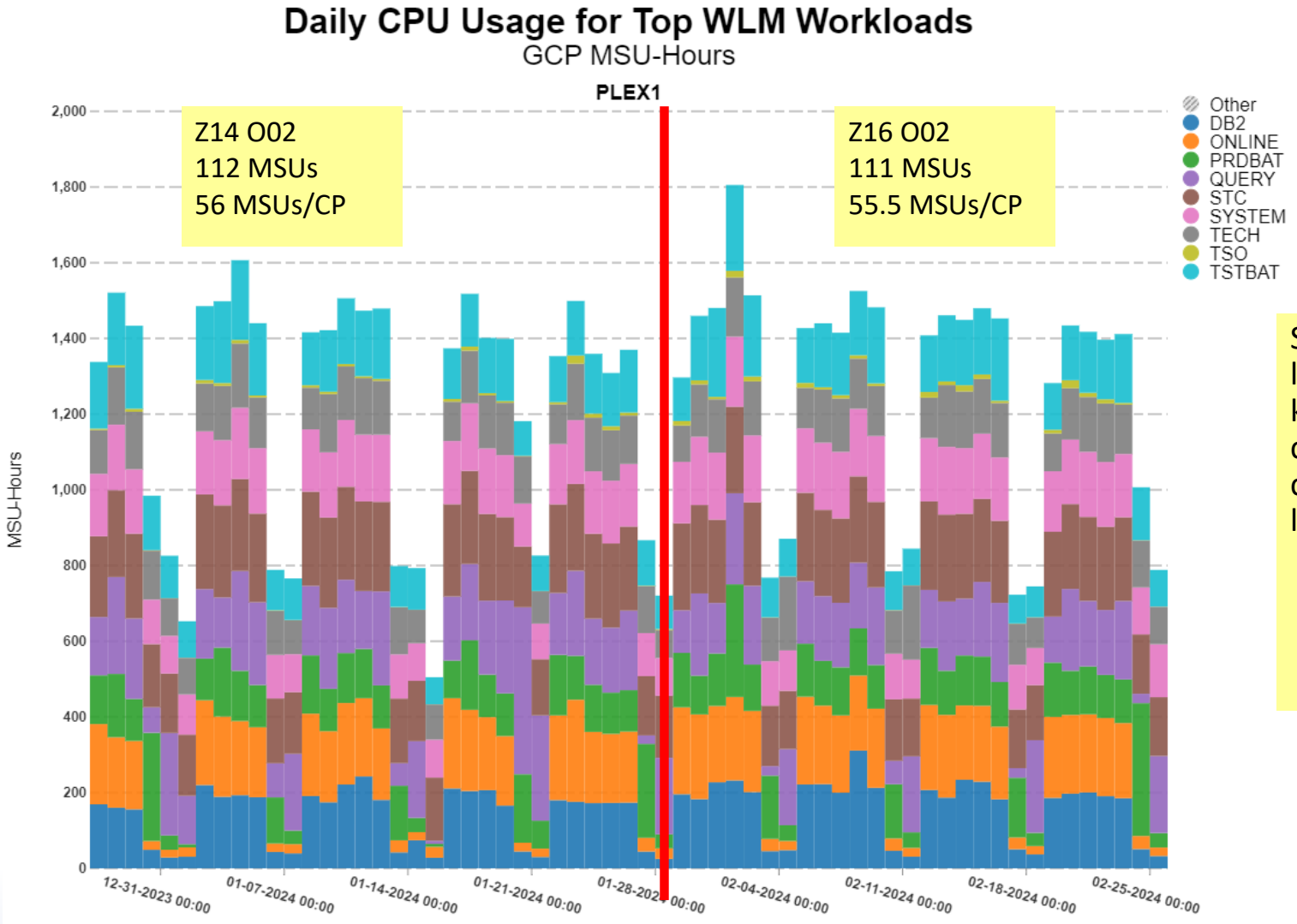
Emerging Areas of Interest

New things coming and things we're actively keeping an eye on

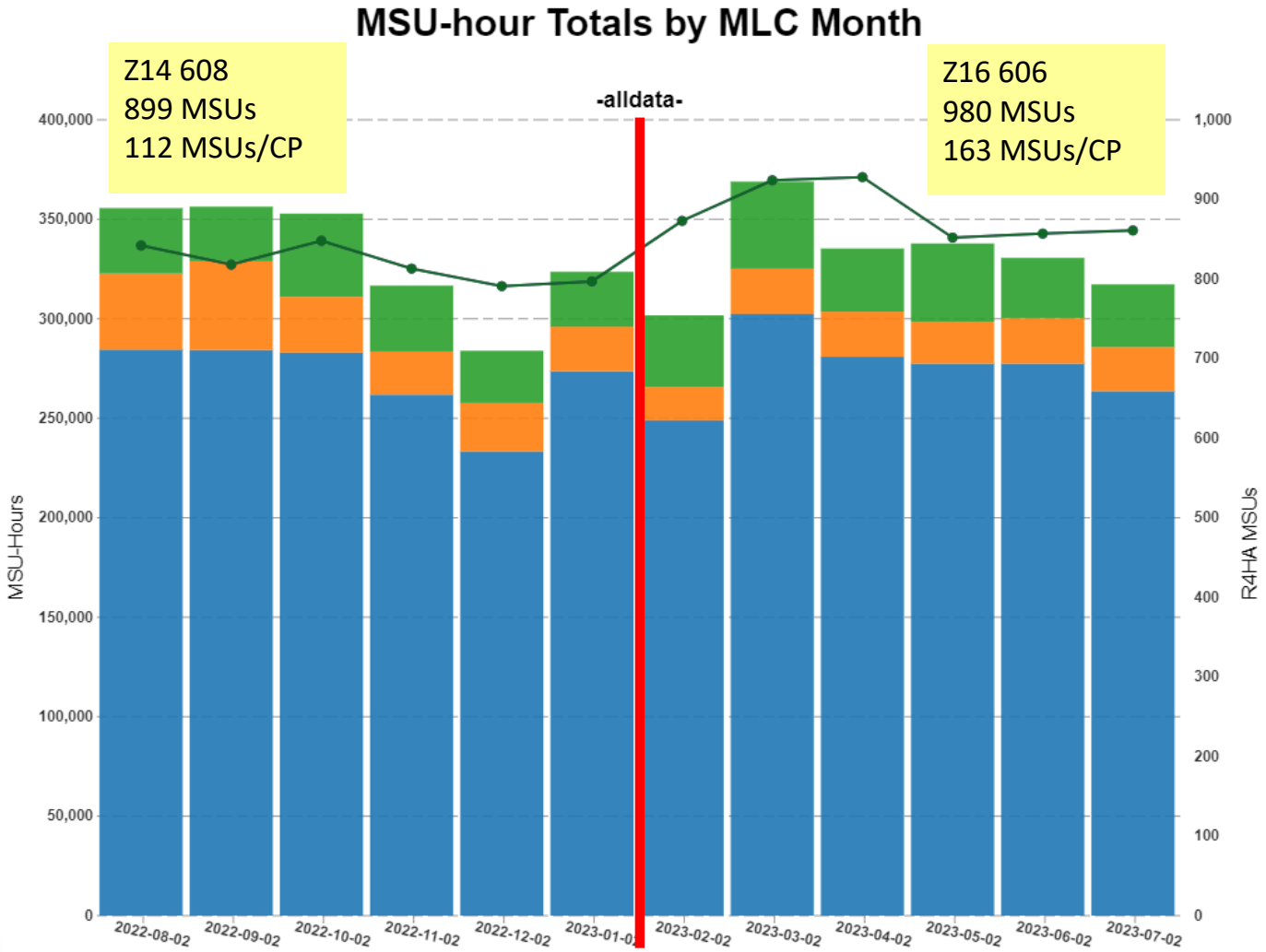
z16 Migrations

TLDR: Mostly going as expected

- There was some questions about how the significant cache design change would behave in real life
- For the migrations we've seen, it seems that migrations to the z16 have been pretty much along (our) expectations
  - Except for the one customer that did contact us that saw higher MSU consumption, but the had moved to fewer/faster CPs
- In general, fewer/faster CPs are likely to be worse for overall system efficiency
  - Thought the larger L2 cache size might mitigate this, but… maybe not
- More/slower (or more/faster!) better for efficiency
- Staying with same number of CPs is the conservative approach



Simple migration, very low risk because they kept the same engine count and overall capacity rating. Looks like it went fine.



Migration to fewer/faster.

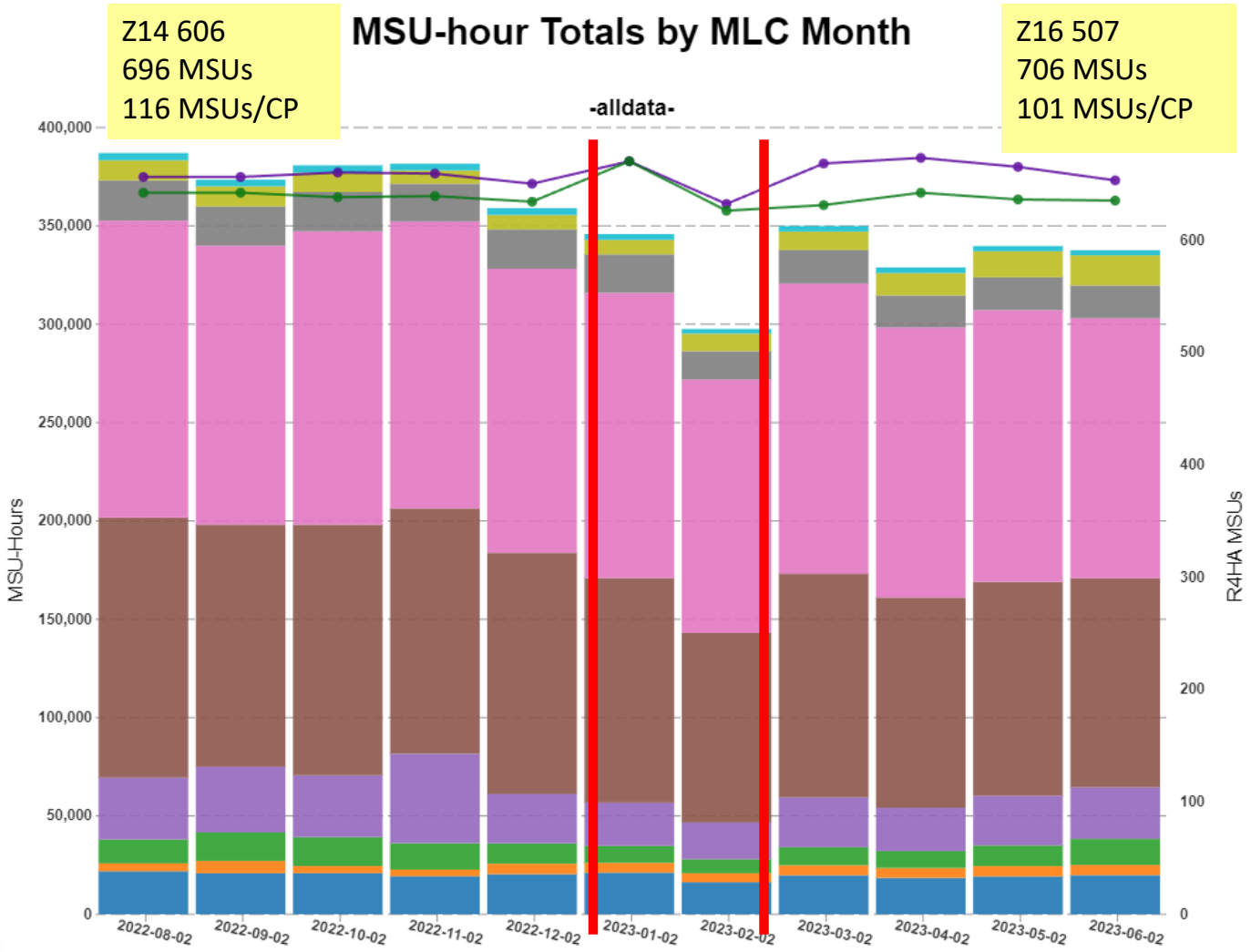
Customer had some concerns after migrating because of increased CPU consumption.

RNI had gone up, CPI stayed similar.

May have been workload related as effects seemed reduced in later months. Also, recompiling after moving to new architecture can help.

Z14 606
696 MSUs
116 MSUs/CP

Z16 507
706 MSUs
101 MSUs/CP



This was a phased migration over about a month.

Extra engine only slightly slower.

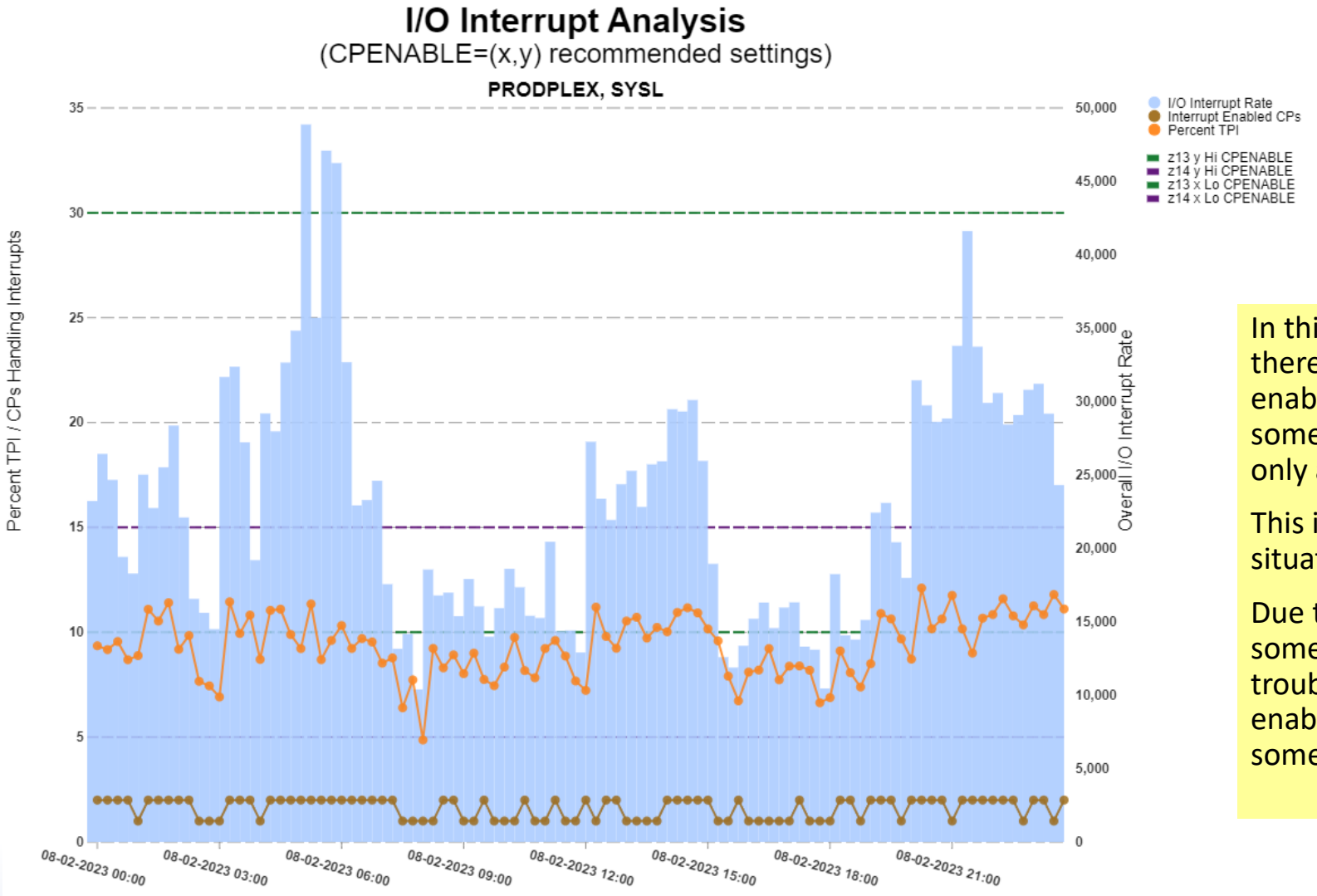
Peak R4HA did go up slightly, but total MSU-hours is down noticeably.

CPI of larger systems did improve, RNI stayed about the same.

CPENABLE in z/OS 3.1

CPENABLE

- CPENABLE in IEAOPTxx sets the low and high threshold for disabling / enabling processors for handling I/O interrupts
- z13 and below recommendation is (10,30)
- On z/14 and above the recommendation is (5,15)
  - Prior to z/14 all no-work wait CPs were enabled for interrupts
  - z/14+ rely solely on WLM/SRM to set the number of CPs enabled for interrupts
- The goal of this change was to better ensure 2 CPs are enabled for handling I/O interrupts
  - Single CP enabled for I/O interrupts puts LPAR at greater risk of delaying I/O
  - Sometimes with quite problematic results – having 2 is partly risk mitigation
- We've sometimes recommended even more aggressive settings (e.g. 3,10)



In this case, sometimes there were 2 CPs enabled for interrupts, sometimes there was only a single CP.

This is a fairly common situation.

Due to arrival patterns, some systems have trouble getting a second enabled even with something like (3,10).

CPENABLE in z/OS 3.1

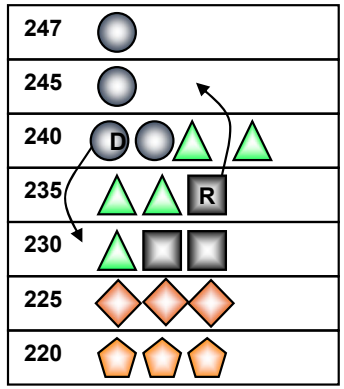
- In z/OS 3.1 minimum CPs enabled will raise from 1 to 2
  - The only z/OS 3.1 LPAR we've seen data for only had 1 online processor ☹
- New CPENABLE option of SYSTEM will take IBM's recommendation for the generation of hardware the system is running on
- Evaluation of enabled CPs will change from 20 seconds to 2 seconds

- We think this is a great change!
  - Will be able to specify CPENABLE=SYSTEM and probably not worry about it
  - A lot of I/O can happen in 20 seconds so changing to every 2 seconds (same as HiperDispatch cycle) makes sense
    - Extra path length seems like it would be pretty minimal

Implicit CPU Protection in z/OS 3.1

CPU Critical aka Long-term CPU Protection

- Long-time option in your WLM service definition
- Enabled by setting YES for CPU Critical on a Service Class
  - Must be a single-period SC and cannot be discretionary
- Ensures that the CPU Critical SC always has a CPU DP that's greater than the DP of lower importance service class periods



Note: To make the point, just a few priorities between DP 203 and DP247 are shown.

- Note some small amount of lower-importance work may still get higher DP:
  - Due to promotion for locks, resource contention, etc.
  - Small consumers
- General recommendation has been to avoid this option
  - Allows WLM to make better decisions about balancing overall work throughput to best meet the goals of all work

> **Important:** The use of these options limits WLM's ability to manage the system. This may affect system performance and/or reduce the system's overall throughput.

New IBM Defaults in z/OS 3.1

- New option for “Implicit” Long-Term CPU Protection
  - In other words, CPU Critical without having to specify it on every SC definition
- Default is “On” for importance 1 service classes
  - Optional, but “Off” for importance 2 service classes
- We think “On” for importance 1 workloads is a bad default
  - Could significantly change the dispatching priority of work in the system
  - Goes against historical practices of not changing defaults that change behavior
- DP/Importance inversions are common
  - I.E. Lower Importance work running with a DP above higher importance work
  - Not all such inversions are problematic
  - Not all importance 1 work really should be importance 1

Our thoughts

- We don't see the need for this change
  - A significant part of the premise of WLM was that it would manage dispatching priorities and could intelligently move them in possibly counter-intuitive ways to better balance throughput for diverse workloads
  - If you want, you can make all importance 1 work CPU Critical today
- We'd recommend turning this off for z/OS 3.1 and wish that was the default
- If you want to go to z/OS 3.1 with it on, we might suggest
  1. Evaluate which workloads are at risk
  2. Before 3.1, incrementally add CPU Critical to importance 1 workloads
     - If something goes wrong, you can back out your change and z/OS 3.1 doesn't get the blame
- We do sometimes recommend CPU Critical, but it's an exception, not the rule
- Emerging area of study, we might refine our recommendations over time

New WLM Enclave Measurements

Geared towards ‘enclave bundled transactions’ in response to High Performance DBATs

Upending >25 Years of DDF Management

Enclaves allow the *management of individual transactions* flowing through address spaces, something that simply has never been possible before. Since MVS is aware of and has access to each transaction, they can be classified individually and most importantly *each transaction is subject to period switch*. ~~This means that you can separate out the long-running CPU killers from the shorter requests in the same manner that most installations already employ to control batch, by period-level controls.~~

Each enclave is a single transaction, which starts when the enclave is created and ends when the enclave is deleted. ~~DDF creates an enclave for an incoming request when it detects the first SQL statement and deletes the enclave at SQL COMMIT, thus a DDF enclave transaction consists of a single SQL COMMIT scope.~~

~~In WLM goal mode, all goal types are valid for enclaves.~~

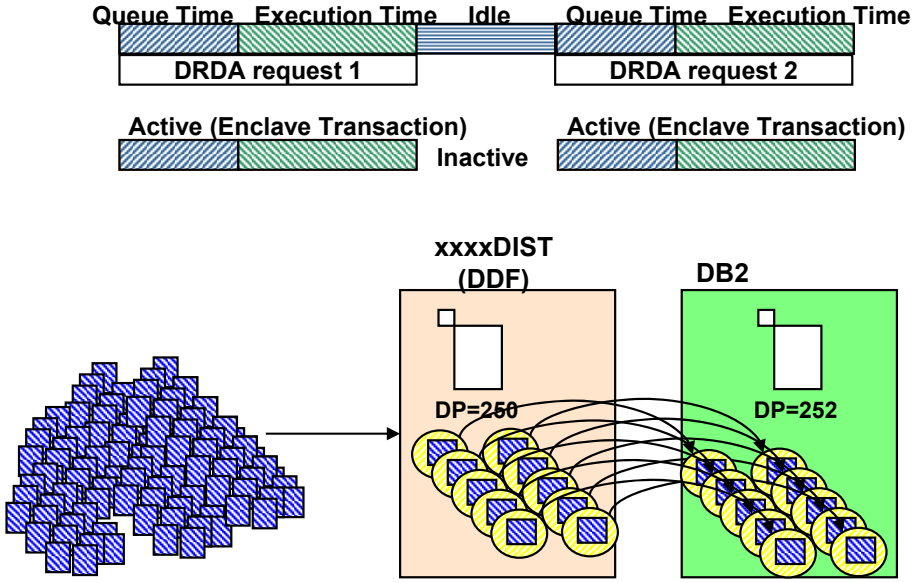
This is now the situation if your DDF work uses High Performance DBATs!
(With DB2 APAR PH34378)

Technically: multi-period service classes and RT goals still are allowed but they will not work as expected!
This could be a surprise when you apply the DB2 maintenance that made this change.

Traditional / Typical DB2 DDF Enclaves

- DDF transactions are managed by WLM as independent enclaves
  - The life of the enclave is the life of the transaction (i.e. 1 tran = 1 ended enclave)

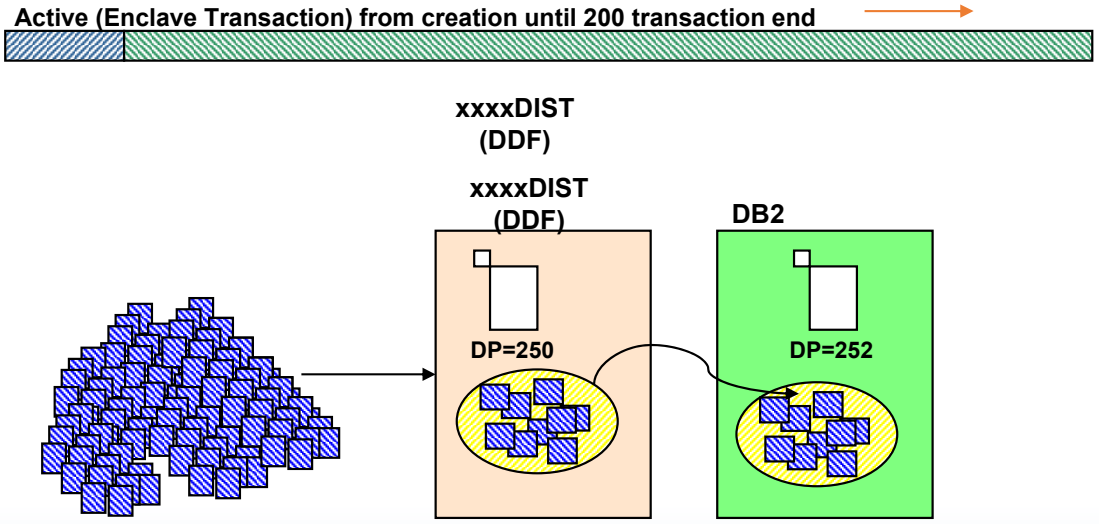
- In this case, 200 transactions come in, and for each transaction an enclave is created
- When each transaction ends, the enclave for that transaction is deleted, and an ended transaction and response time are posted to the WLM Service Class period
- WLM response time goals are best since each transaction is measured and considered by WLM



DBAT DB2 DDF Enclaves w/ PH34378

- The DBAT transactions still run in independent enclaves
  - But now the life of the enclave is for as long as up to 200 transactions in the enclave
    - The transactions in the enclave are known as 'bundled transactions'.
  - Thus, you may have 200 ended bundled transactions, but from a WLM point-of-view, the 200 bundled transactions are part of a single (i.e. 1) longer running enclave.

- In this case, when a new thread is needed, a new tread is created and associated with a new enclave.
- That thread and associated enclave remain active for up to 200 transactions come in, and all are associated with a single enclave.
- After the last of the 200 transactions end, enclave is deleted, and one ended transaction is posted to the WLM service class period. Response time is from start of first transaction to end of last ended transaction
- The many transactions running in the enclave are known as 'bundled transactions'

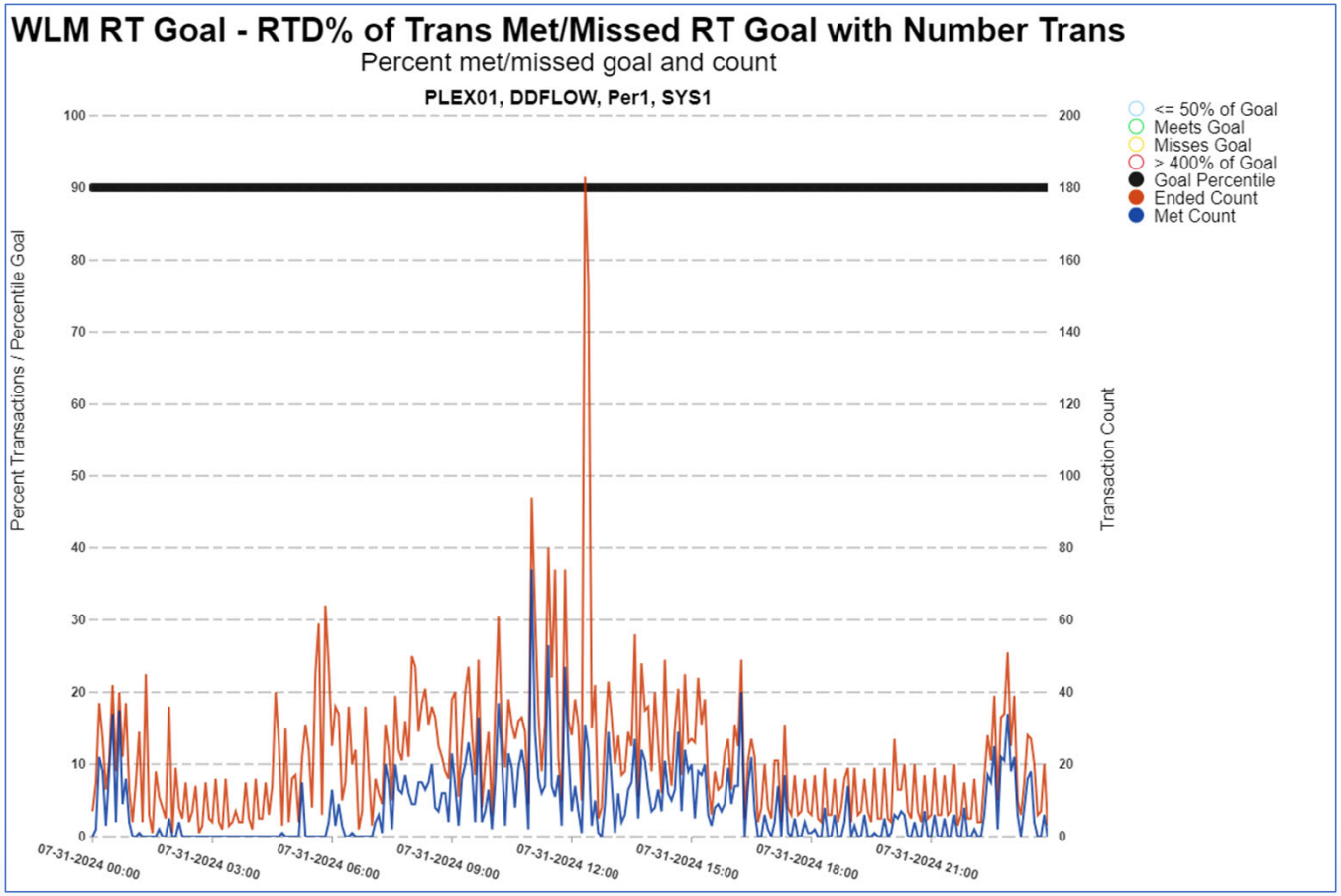


New SMF fields in support of Enclaves

- New SMF 72, subtype 3 SMF fields in support of enclaves
  - Fields are useful to help investigate if DB2 High Performance DBATs are being used

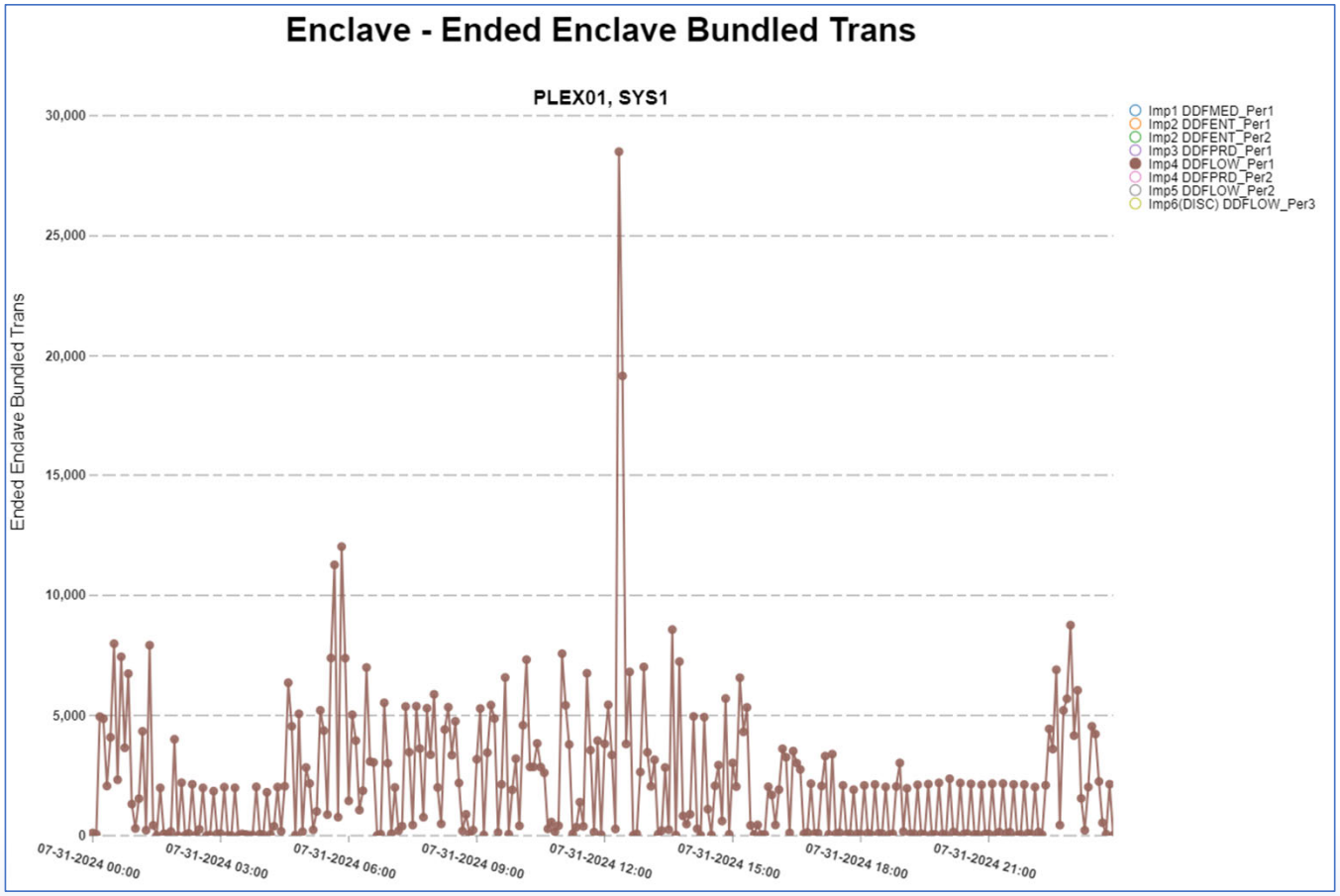
Field Name	Description
R723ENCTRXNUM	Number of subsystem transactions processed within enclaves.
R723ENCTRXCALLS	Number of times transaction data has been reported by subsystem work managers when deleting an enclave. When zero, no transaction data for enclaves has been provided by the subsystem work manager.
R723ENCTRXET	Total execution time, in microseconds, for all subsystem transactions reported in R723ENCTRXNUM
R723ENCTRXETS	Sum of squared execution times, in microseconds, for all subsystem transactions reported in R723ENCTRXNUM.

Example: DDFLOW Response Time Distribution



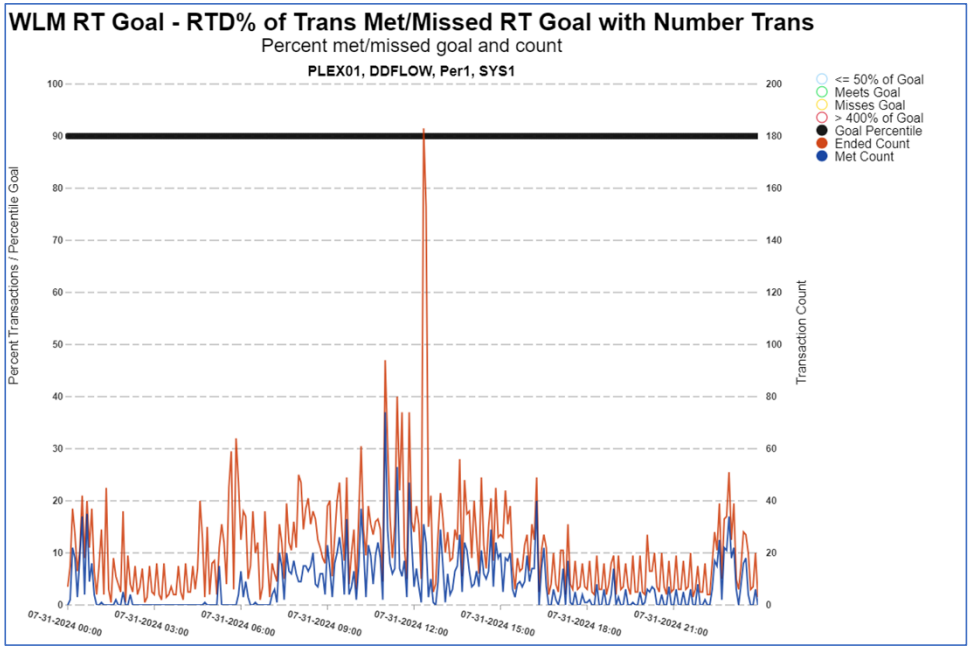
- DDFLOW has a response time goal of 90% complete in 0.3 seconds
- Notice that we have a fair low number of ended transactions
- Notice the large gap between number of ended transactions and the number that met the goal value.

Example DDFLOW Ended Bundled Transactions

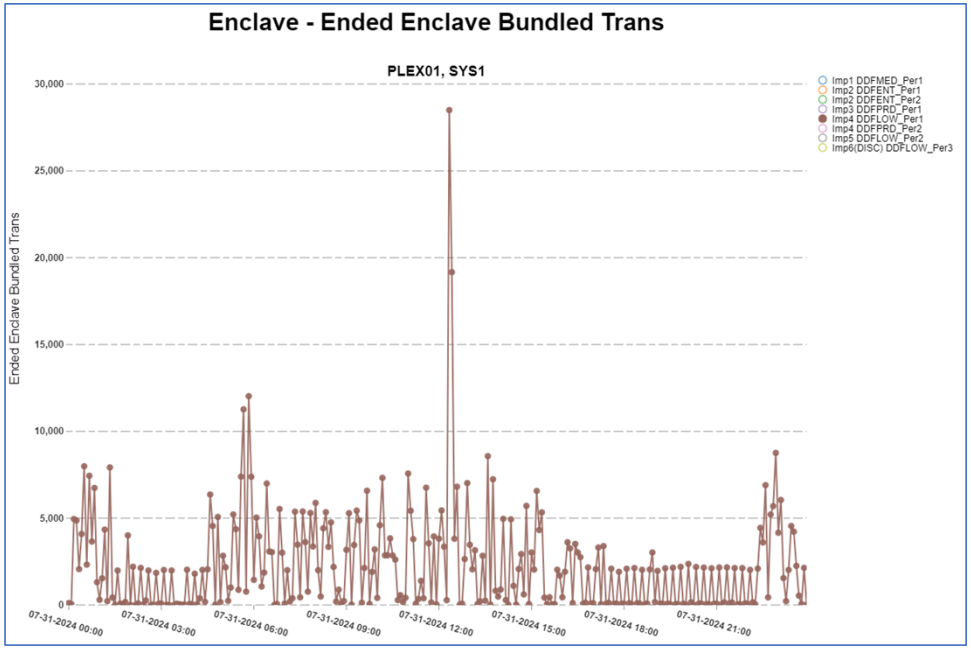


- DDFLOW has a response time goal of 90% complete in 0.3 seconds
- Notice there were thousands of ended bundled transactions

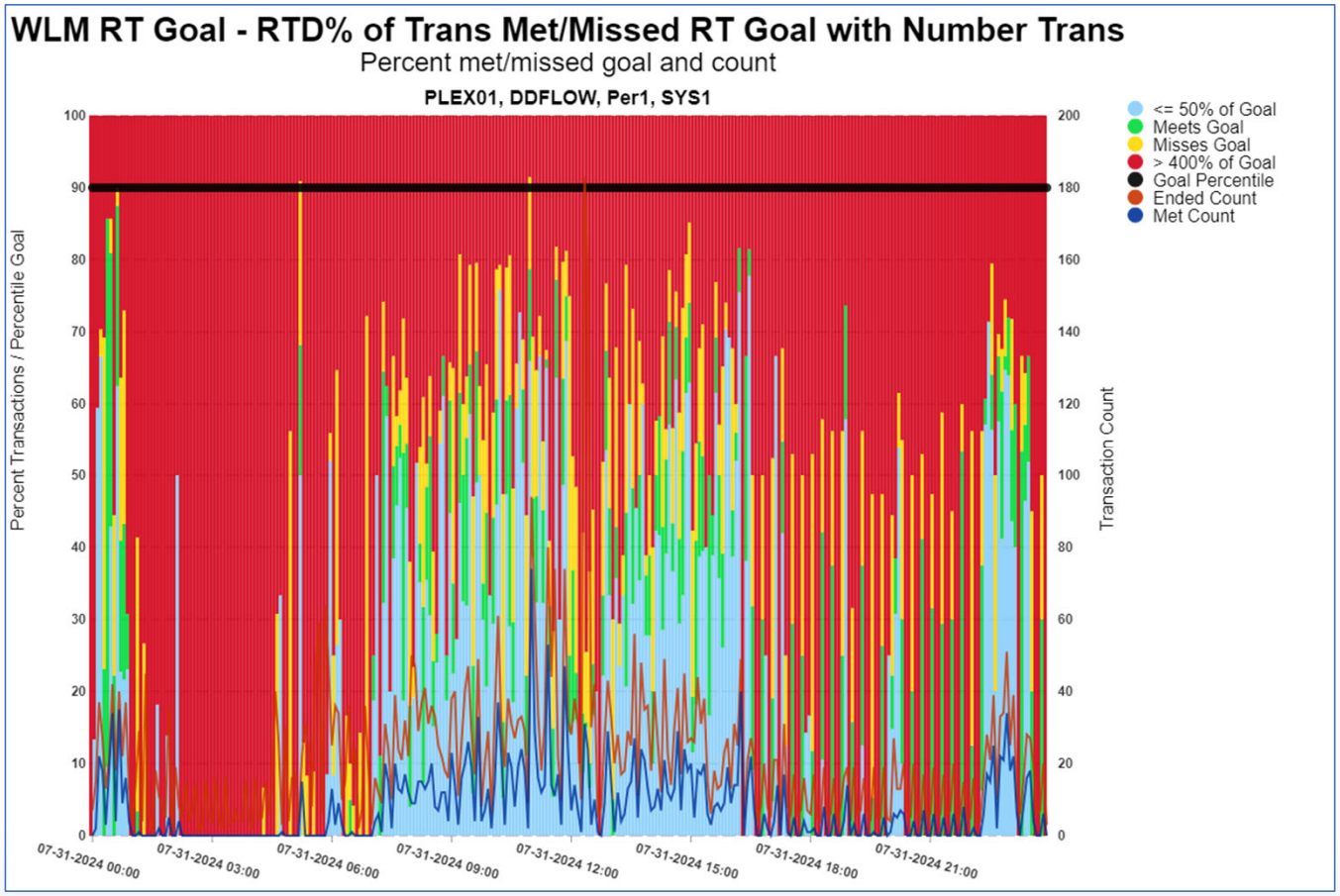
Ended WLM Transactions (Enclaves)



Ended Bundled Transactions



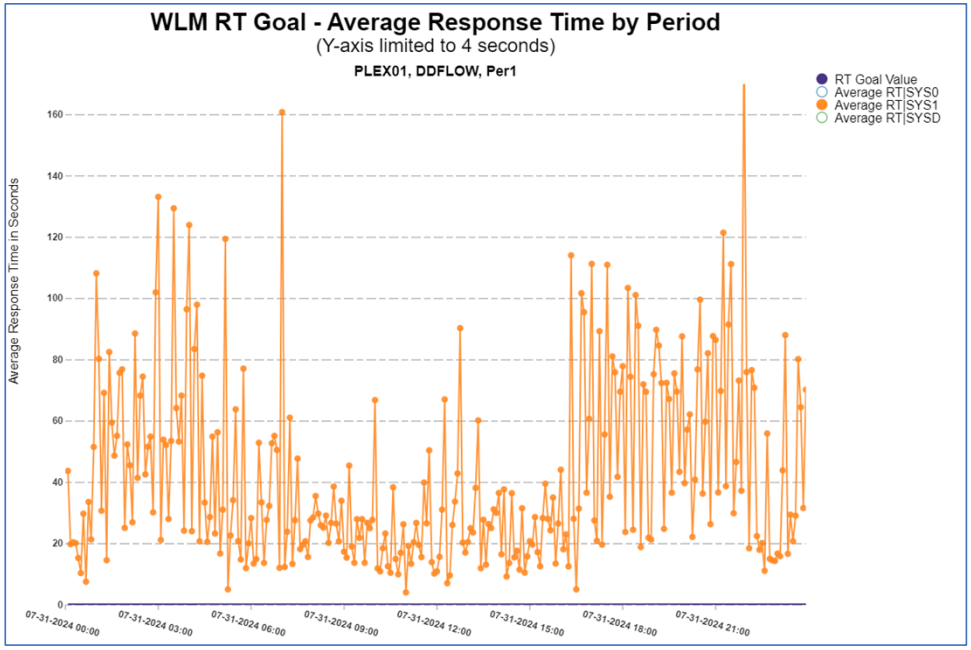
Example: DDFLOW Response Time Distribution



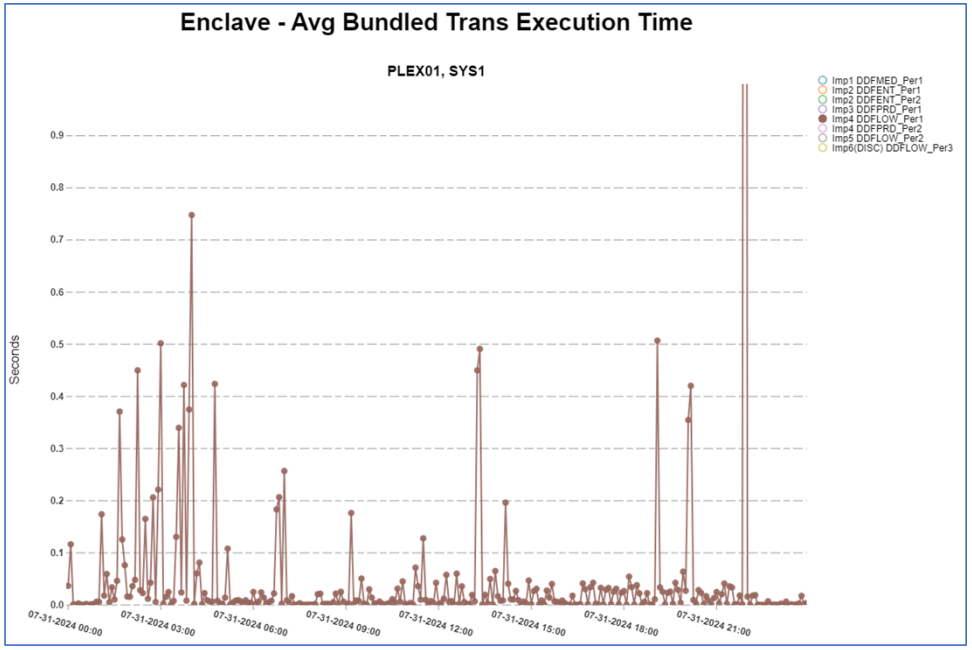
- DDFLOW has a response time goal of 90% complete in 0.3 seconds
- Added to this chart is the percentage of transactions that met and missed the goals
- Not that a large percentage of transactions missed the goal value of 0.3 seconds

Example: DDFLOW Response Time

Average Response Time of Enclaves (Used for goal management)



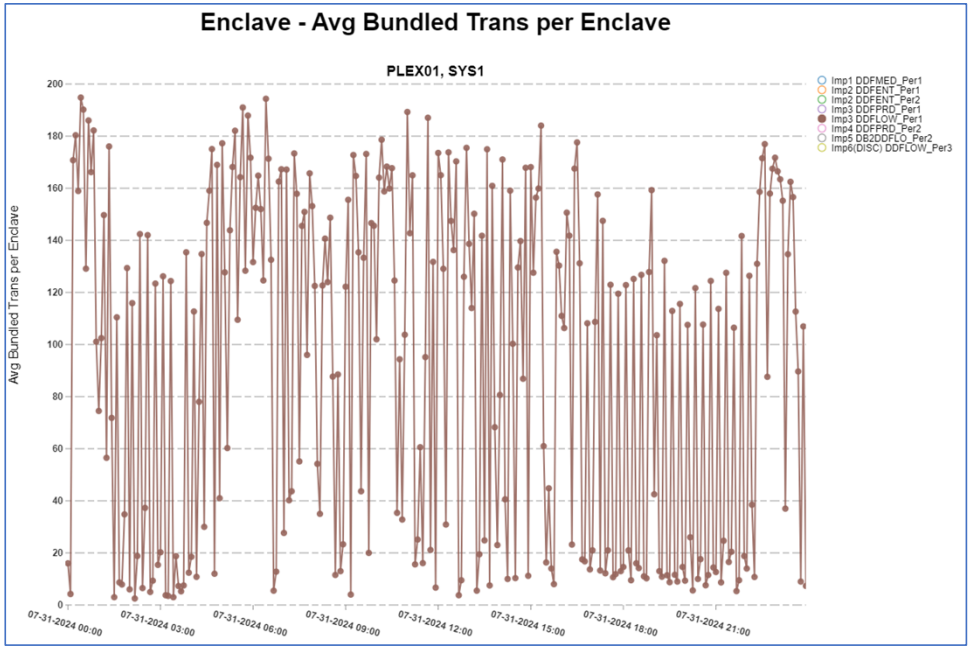
Average Response Time of Bundled Transactions (Not used for goal management)



Example: DDFLOW Enclaves for Bundled Transactions

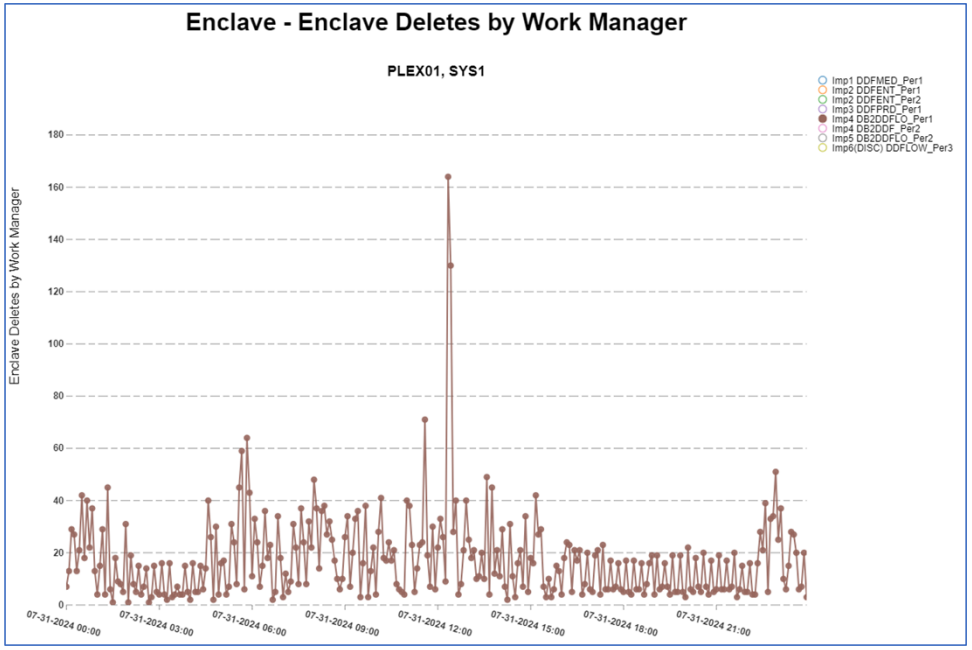
Average Bundled Trans per Enclave

(Note the 200 limit)



Deleted Enclaves for Bundled Trans

(An ended enclave is one WLM ended transaction even though up to 200 DBAT transactions ran in each enclave)



Possible Approaches to DDF Service Classes

- Understand your applications usage of DDF and HiPerf DBATS
  - Are enclaves being used to run bundled transactions?
    - Use the new measurements listed. If non-zero than you are using bundled transactions
- If running HiPerf DBATS, update your WLM setup for DDF
  - Best solution:
    - Work with the DB2 folks to segregate HiPerf DBATs away from other DDF transactions into separate service classes. DBAT transactions should be assigned a velocity goal
  - Mediocre solution:
    - In mixed environments, covert all your DDF service class period goals to velocity
  - Worst solution
    - Do nothing and just use your existing response time goals for the mixed DDF.
    - Adjust goals accordingly
  - Peter's solution
    - Convert the service class running both regular DDF transactions and HiPerf DBATs transactions into a multiple period service class with both periods at the same importance
      - Period 1: Importance x with short duration with a response time goal
      - Period 2: Importance x, with velocity goal

First Reference Page Faults Decrease Capture Ratios

What is a first reference page fault?

- Demand Page Faults
  - Typically, virtual frames are backed by real storage
  - If there is stress on storage, a real frame could be paged out to auxiliary storage
  - When that frame is re-referenced, this is known as a demand page fault
  - Demand Page Fault:
    - When a referenced page of virtual storage is not backed by a frame in central storage, a page fault occurs. This requires z/OS to retrieve the page from auxiliary storage and bring it into central storage.

- First Reference Page Fault
  - When a referenced page of virtual storage is not ***YET*** backed by a real frame in central storage, a first reference page fault occurs
  - It is the 1st reference page fault that drives Dynamic Address Translation (DAT), and the real frame is associated with the virtual address

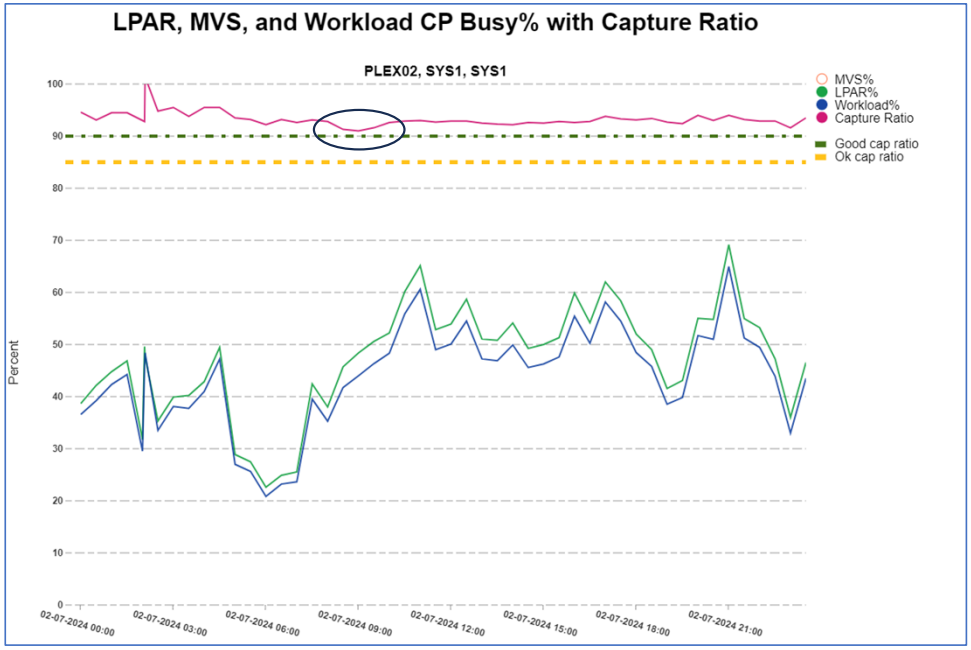
Capture Ratios and 1st Referenced Page Faults

- IBM WCS says that 1st Reference Page Faults contribute to uncaptured times
  - And that 1st Reference Page Fault rates above 100,000 per second should be considered problematic

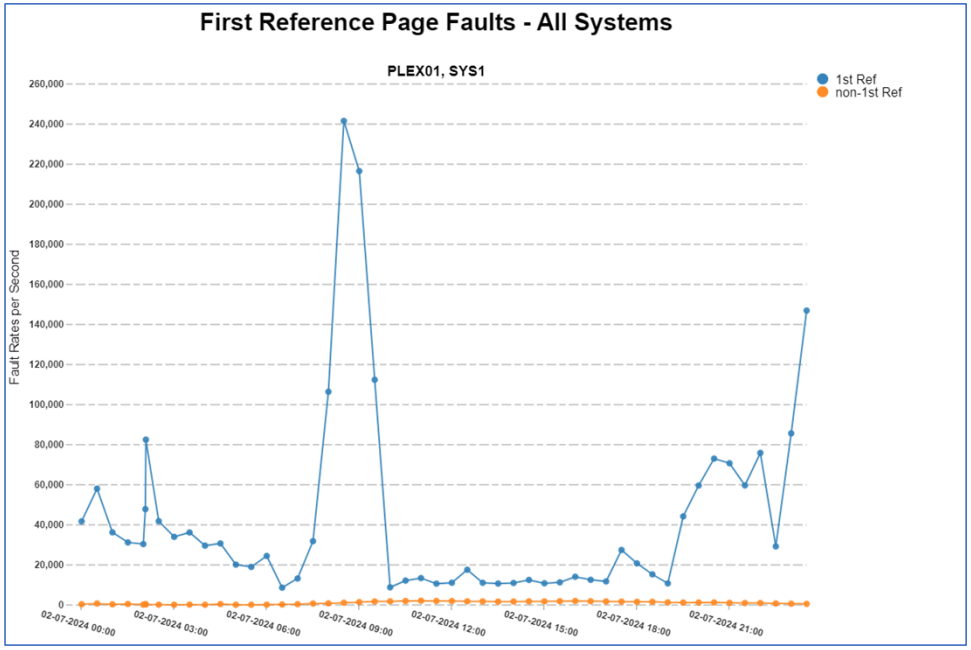
- Comments:
  - There is not much that can be done by customers to alleviate 1st Reference Page Faults
    - Perhaps recode applications to get less storage?
    - However, correlating them to capture ratios can be helpful to explain some of the uncaptured times
  - So many things contribute to uncaptured times, that is tough to see the direct correlation
  - Just understand this, and if investigating low capture ratios, then consider analyzing your 1st reference page faults to *maybe* help explain.

Example : Tough to see any correlation

Capture Ratios for System

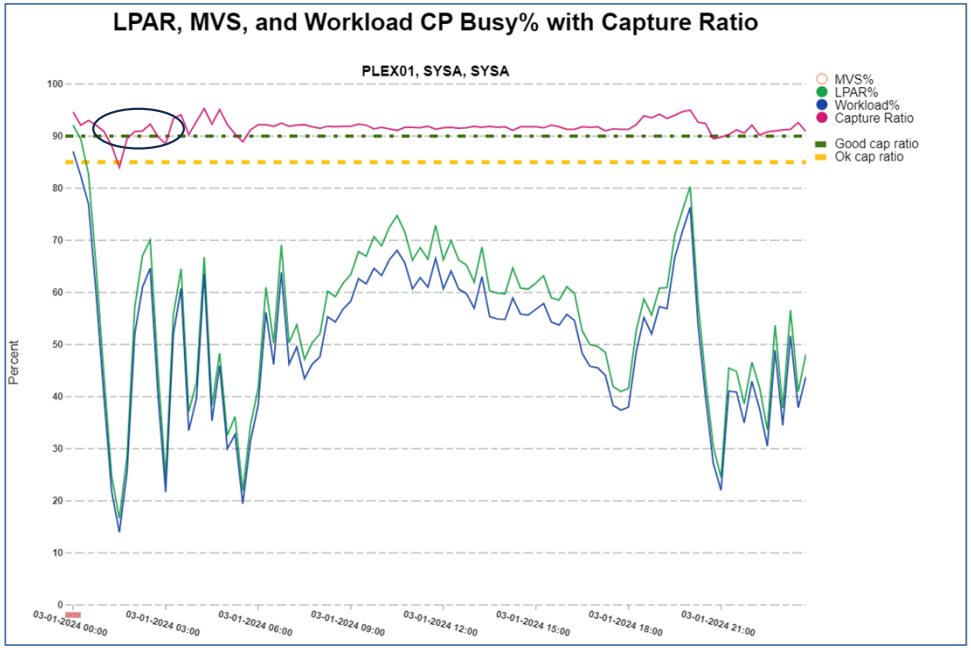


1st Reference Page Fault Rates

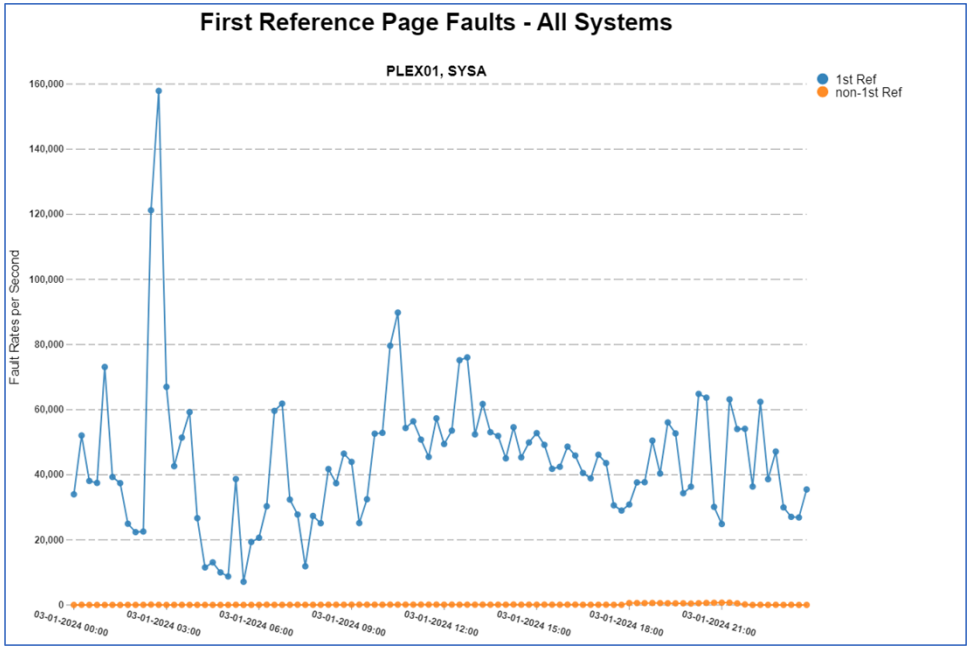


Example : Tough to see any correlation

Capture Ratios for System



1st Reference Page Fault Rates



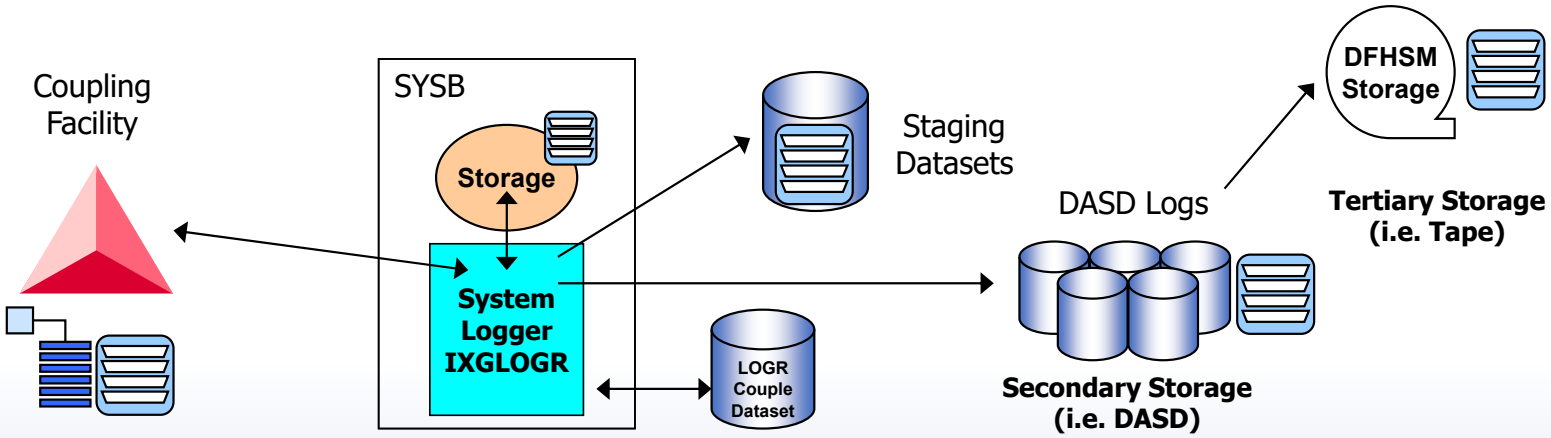
New z/OS System Logger IXGCNFxx Parameter

KEEPLOCALBUFFERS(NO | YES)

Targeted to alleviate the uncaptured time due to 1st reference page faults

Introduction to z/OS System Logger

- z/OS System Logger - Component of z/OS that provides logging services
  - IXGLOGR – key system address space for logger functions
  - Interim Storage - Primary storage used to hold the log data that has not yet been offloaded
    - What 'interim storage' is depends on how the log stream has been setup
    - Examples of include central storage (via a data space), Coupling Facility, Staging data sets
  - Secondary Storage - generally DASD
  - Tertiary Storage – generally Tape medium



New system Logger IXGCNFxx Parm

(APAR OA63551)

- PROBLEM DESCRIPTION:
  - New function to reduce page faults caused by IXGWRITE requests that were submitted after a log stream offload occurred.
- RECOMMENDATION:
  - Delays in completing IXGWRITE requests can occur as a result of page faults associated with system logger local buffers used by IXGWRITE processing.
- Comments
  - A new IXGCNFxx parmlib option will be introduced to keep the real frames that back the local buffers when the storage for the local buffers are freed after a log stream offload.
  - Keeping the real frames reduces page faults that will occur when the local buffers are reused during subsequent IXGWRITE requests. This will result in an increase of real storage associated with the System Logger address space.

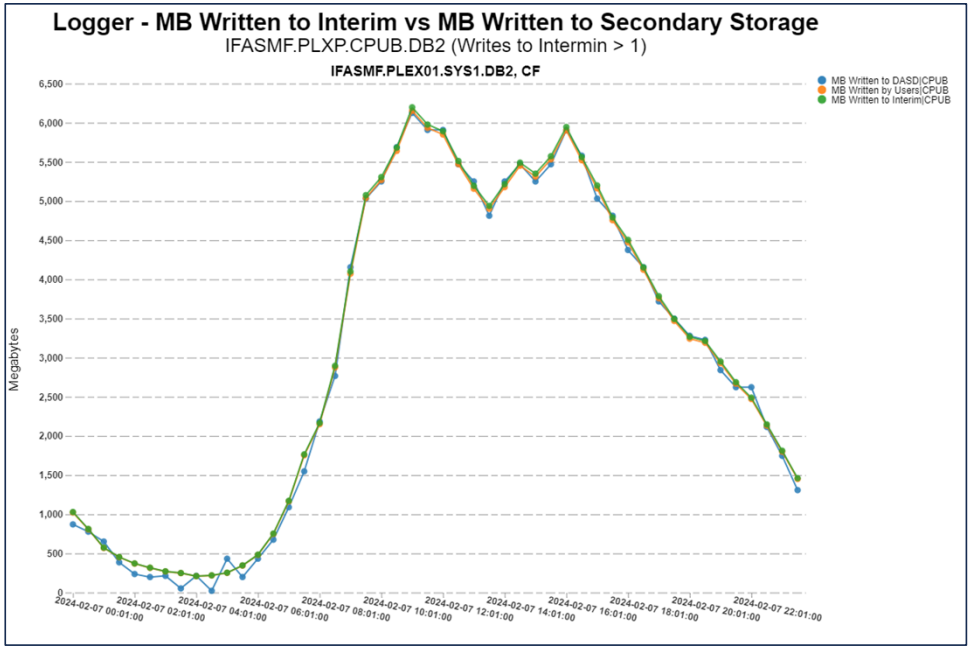
New IXGCNFxx KEEPLOCALBUFFERS Parm

KEEPLOCALBUFFERS(NO | YES)

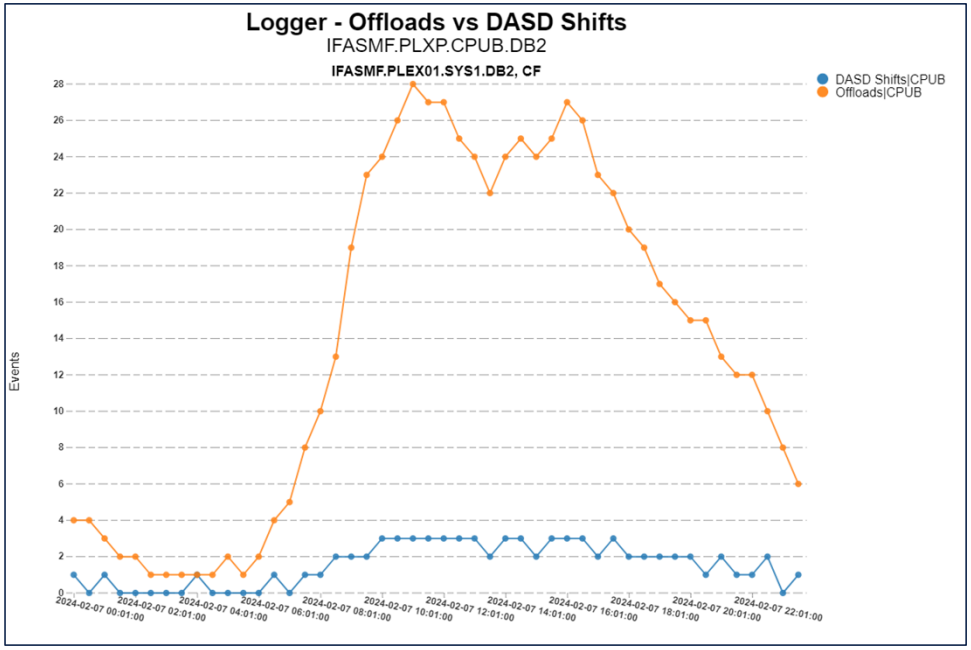
- Specifies whether the system will request to keep the real frames backing the local buffers used as interim storage when it is freed. Keeping the real frames reduces page faults that will occur when the local buffers are reused during subsequent IXGWRITE requests.
- Note: Local buffers are data space areas associated with the system logger address space, IXGLOGR. Specifying KEEPLOCALBUFFERS(YES) may result in systems experiencing increased paging.
- Evaluate your real memory requirements to ensure unacceptable paging does not occur by reviewing the amount of real memory consumed by the system logger address space, IXGLOGR.
- The following options are possible:
  - NO - Indicates that the system will not keep the real frame used to back local buffers when the buffer storage is freed.
  - YES - Indicates that the system will request to keep the real frame used to back local buffers when the buffer storage is freed.
- You can use the DISPLAY LOGGER,IXGCNF,MANAGE command to view the parameter settings for configuring the system logger.
- Default: NO

Example: Logger Offloads of SMF

MBs of SMF offloaded

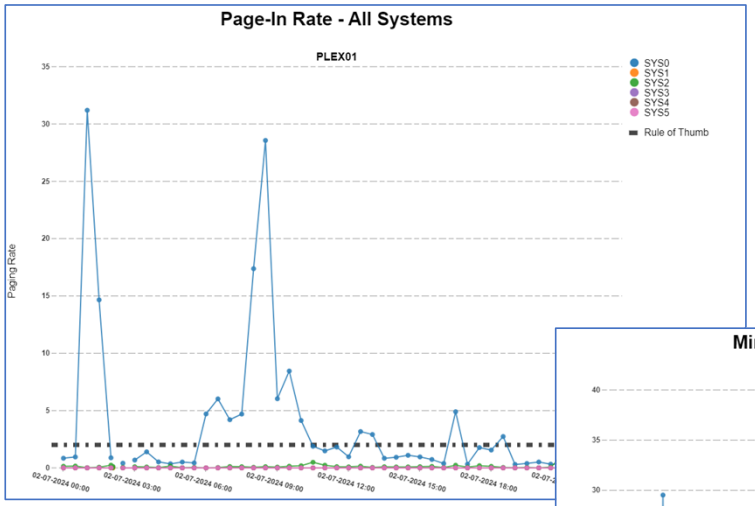


Number of offloads

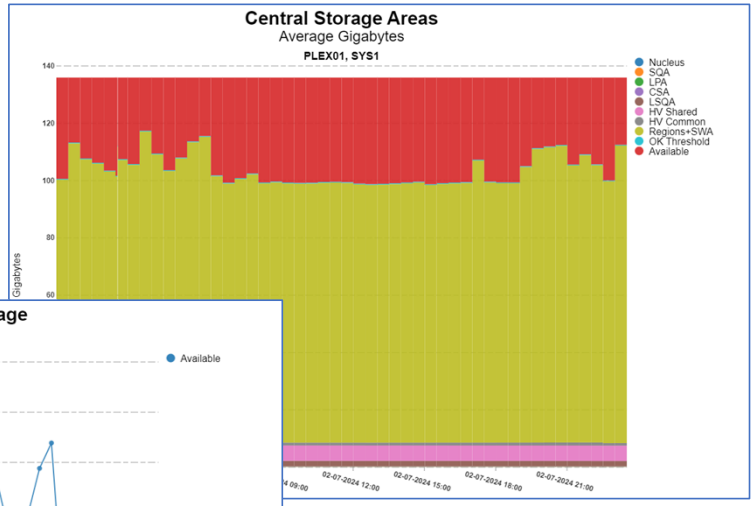


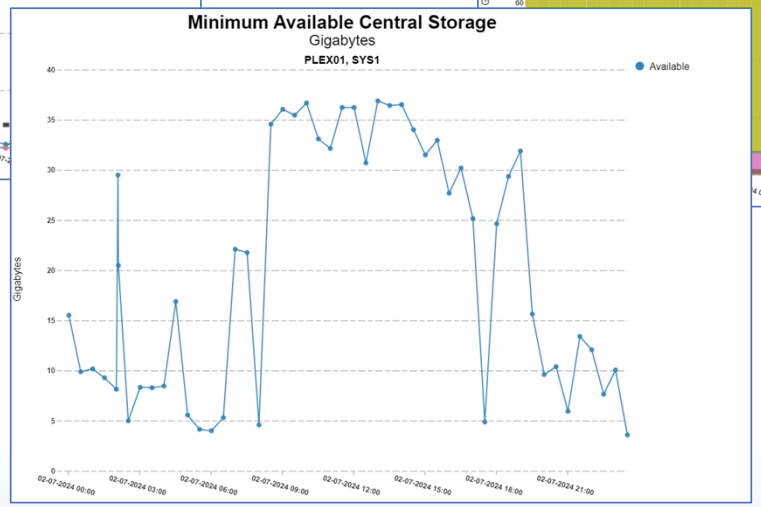
Example: Logger Offloads of SMF

Paging



Average and Min Storage Available

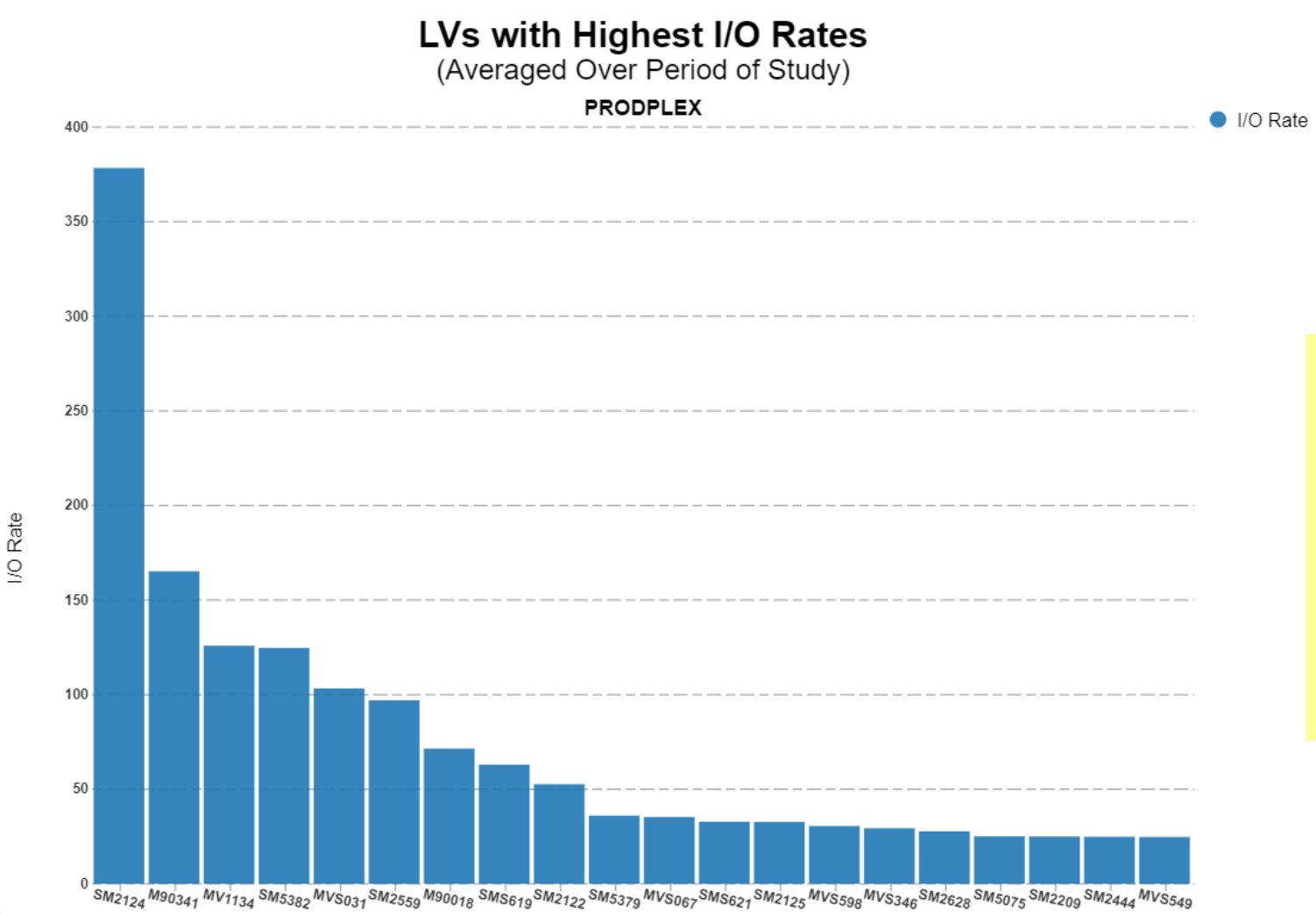




Large memory should mean less I/O

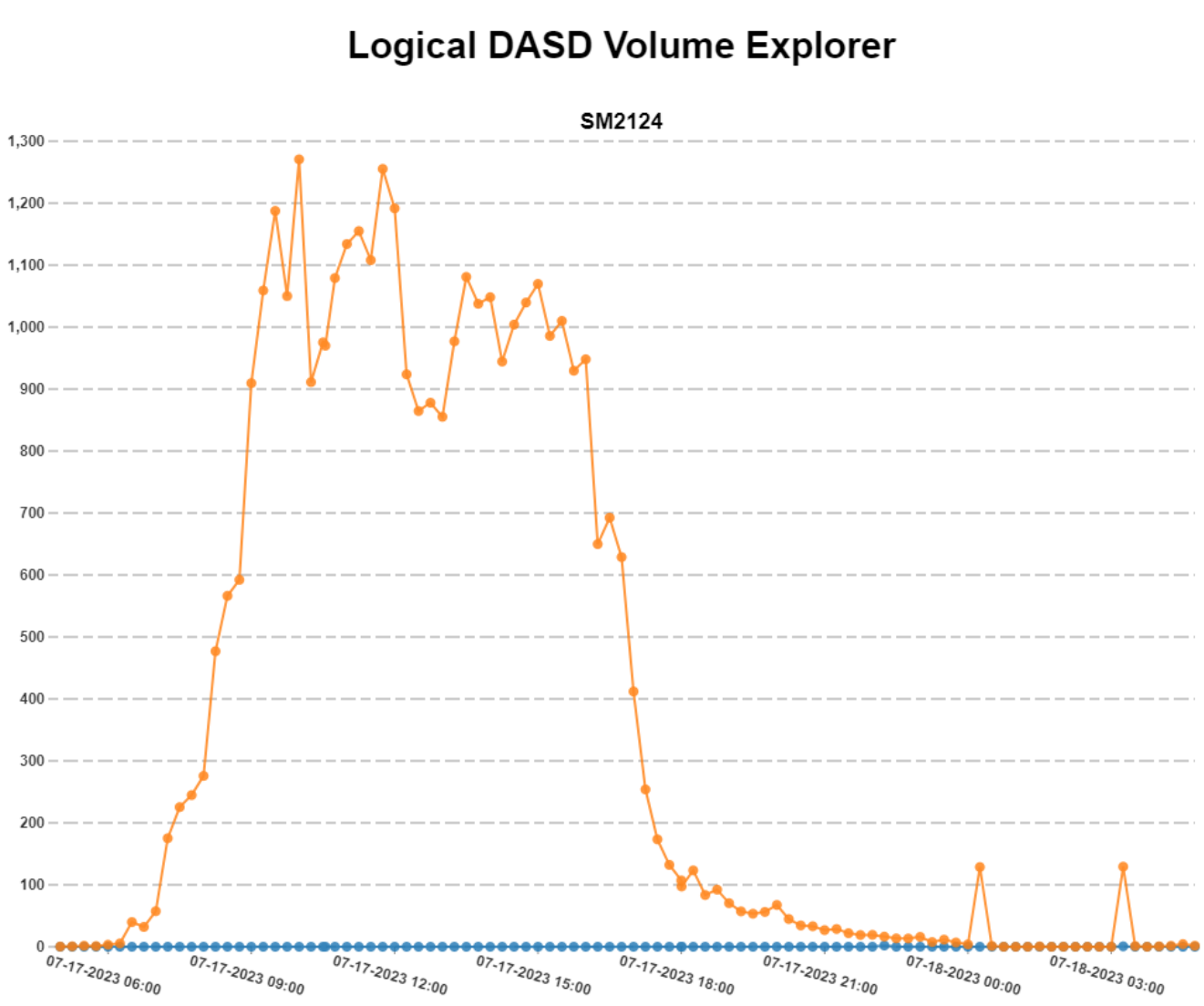
Memory and I/O

- We see systems with lots of memory free and yet they're doing significant amounts of I/O
- We've been saying for a long while things like "make your BPs bigger"
- But lately we've been trying to look deeper to point out opportunities
  - How much data is really on those busy volumes?
  - Which specific datasets are getting lots of read I/O
- So come this afternoon to session 7F wherein I'll talk about this in detail



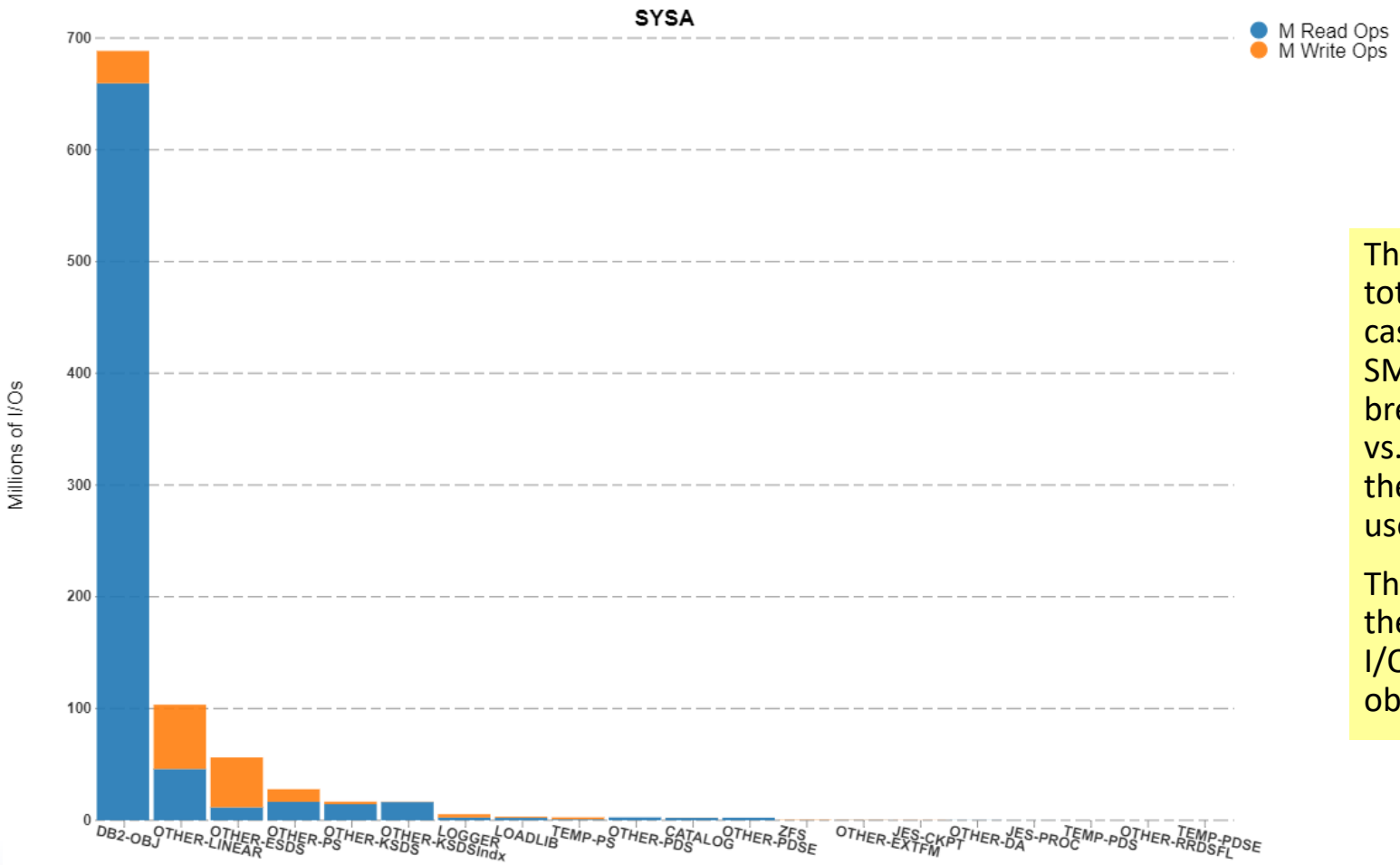
This Pivotor report shows the top volumes by I/O rate over the day.

375 IOPS doesn't sound too interesting but note that is an average I/O rate over 24 hours.



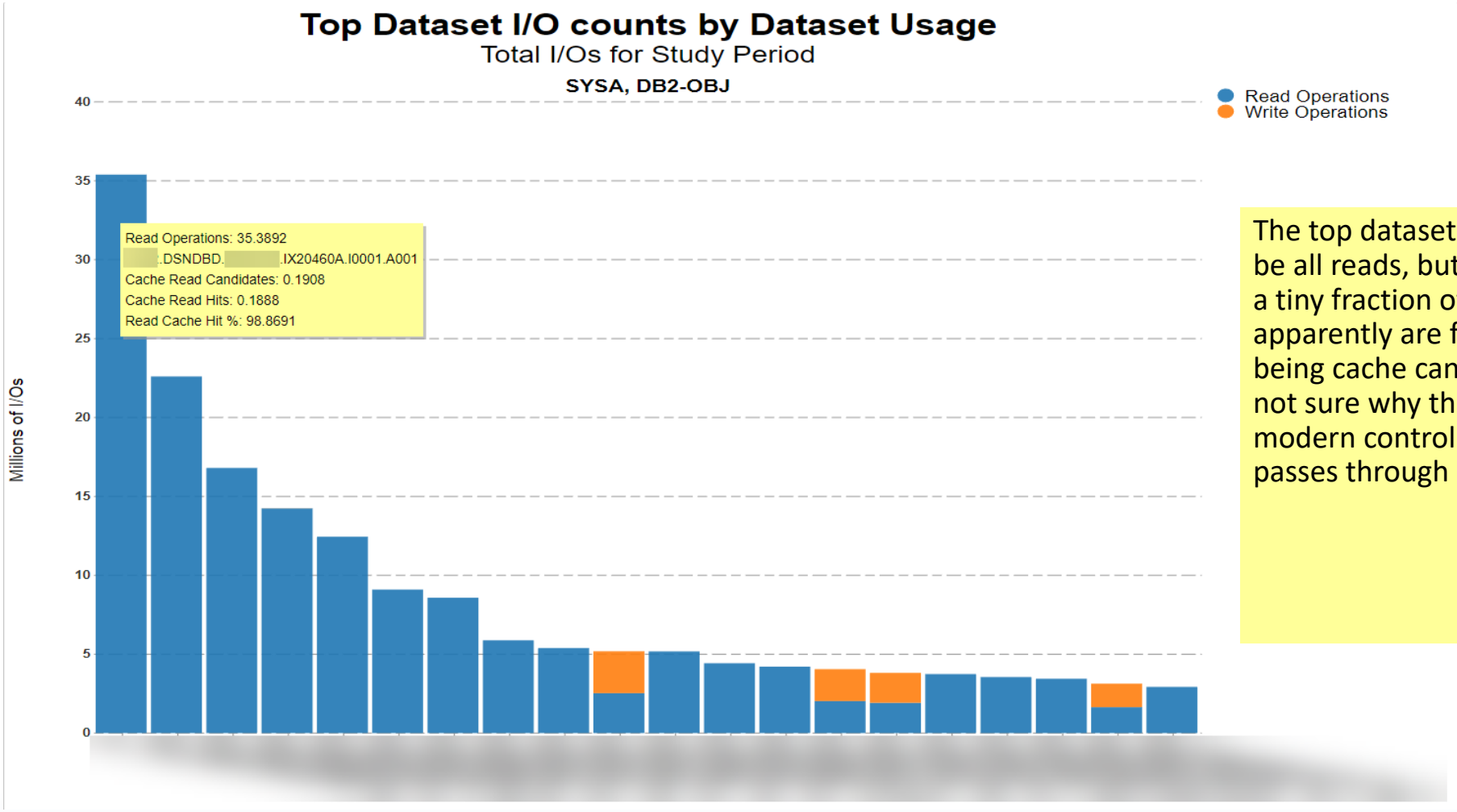
Here's the read and write rate for a particular volume over time. Virtually all the I/O is read I/O, and during the day it is doing over 1000 IOPS.

The kicker: this volume only has 1.5 GB of data stored on it!



This reports looks at the total I/O over (in this case) a day from the SMF 42 records and breaks it down by reads vs. writes and by what the dataset is (probably) used for.

This site is not unusual: the vast majority of the I/O is reading from DB2 objects.



The top dataset appears to be all reads, but oddly, only a tiny fraction of those apparently are flagged as being cache candidates. I'm not sure why that is. But in modern control units all I/O passes through cache.

Top Datasets by Cache Read Hits

2023-07-17

Date	Usage	DS Name	M Cache Read Hits	Cache Hit Pct	Read MiB	Allocated MiB	Read-Allocated Ratio	Read O...	Write Ops	42.6 Records	Volume
2023-07-17	DB2-OBJ	.DSNDBD. .IX20460A.I0001.A001	0.189	98.869	4,201,991.004	569.841	7,373.973	35.389	0.000	54.000	2.0
2023-07-17	DB2-OBJ	.DSNDBD. IX08956B.I0001.A001	20.108	97.188	664,607.695	2,361.233	281.466	22.601	0.001	251.000	8.0
2023-07-17	DB2-OBJ	.DSNDBD. /02.TS06435.J0001.A001	1.450	98.796	1,880,165.453	569.841	3,299.457	16.798	0.000	245.000	5.0
2023-07-17	DB2-OBJ	.DSNDBD. IX00854E.I0001.A001	14.170	99.960	69,569.031	1,339.897	51.921	14.244	0.006	10.000	4.0
2023-07-17	DB2-OBJ	.DSNDBD. IX08956B.I0001.A002	11.436	97.651	288,683.320	1,159.947	248.876	12.444	0.004	241.000	8.0
2023-07-17	DB2-OBJ	.DSNDBD. .TS01452.J0001.A001	8.968	99.152	71,634.121	1,203.719	59.511	9.097			
2023-07-17	DB2-OBJ	.DSNDBD. TS07315.I0001.A001	7.206	98.403	696,717.207	4,175.324	166.865	8.580			
2023-07-17	DB2-OBJ	.DSNDBD. .TS20310.I0001.A001	5.076	96.649	135,041.418	4,720.846	28.605	5.882			
2023-07-17	DB2-OBJ	.DSNDBD. TS08957.J0001.A001	4.991	99.302	228,975.305	4,721.657	48.495	5.390			
2023-07-17	DB2-OBJ	.DSNDBD. TS00854.I0001.A014	3.724	89.910	198,572.117	1,957.563	101.438	5.179			
2023-07-17	DB2-OBJ	.DSNDBD. .TS01451.J0001.A001	1.365	94.870	332,869.598	306.401	1,086.384	4.428			
2023-07-17	DB2-OBJ	.DSNDBD. TS07315.I0001.A001	3.588	99.894	499,237.086	4,377.969	114.034	4.210			
2023-07-17	DB2-OBJ	.DSNDBD. TS07315.J0001.A001	3.121	99.978	417,840.731	4,341.493	96.244	3.551			
2023-07-17	DB2-OBJ	.DSNDBD. .TS07292.I0001.A001	2.965	99.891	77,346.273	4,722.468	16.378	3.444			
2023-07-17	DB2-OBJ	.DSNDBD. .TS17613.I0001.A001	0.156	94.461	345,553.938	284.516	1,214.534	2.930			
2023-07-17	DB2-OBJ	.DSNDBD. .TS07292.J0001.A002	2.255	99.791	71,282.477	4,721.657	15.097	2.738			
2023-07-17	DB2-OBJ	.DSNDBD. TS02809.J0001.A009	2.170	89.721	77,474.367	3,465.251	22.358	2.688			
2023-07-17	DB2-OBJ	.DSNDBD. TS02813.J0001.A004	2.085	95.299	63,996.520	3,465.252	18.468	2.609			
2023-07-17	DB2-OBJ	.DSNDBD. .TS06562.I0001.A001	2.316	96.014	37,477.902	1,738.704	21.555	2.499			
2023-07-17	DB2-OBJ	.DSNDBD. TS17820.I0001.A001	1.968	92.237	73,247.258	2,691.953	27.210	2.451			
2023-07-17	DB2-OBJ	.DSNDBD. .TS17613.J0001.A001	0.062	93.959	295,867.445	284.516	1,039.899	2.446			

This table report joins the SMF 42 data with the DCOLLECT data to get the total allocated size (summed across multiple volumes if necessary) of the datasets.

Note there's little write activity and a number of these datasets are only a few GB.

Even if they can't all go into memory, probably some can, saving 10s of millions of I/Os.

z/OS 2.5 Service Definition Coefficients

Like goals, durations need to be periodically re-evaluated
(but many haven't!)

Service Definition Coefficients Updates

- Recommended values by EPS since about 2018 (maybe earlier)
  - CPU=1, SRB=1, IOC=0 MSO=0
  - Summary of reasoning: Aging a transaction based on I/O no longer made much sense since I/O priority management mattered much less due to advent of PAVs, and most I/O processing is also outside the z/OS operating system. So why age a workload based on its I/O characteristics. It is CPU that matters.
- z/OS 2.5 the SDCs go away, and the values will default as follows
  - CPU=1, SRB=1, IOC=0. MSO=0
  - Basically, it is durations are now based on CPU and SRB service units, and no longer based on the concept of 'service'.
- Most customers are using 1,1,0,0
  - If you haven't made the transition yet, read next slides…

IBM's z/OS 2.5 Migration Step

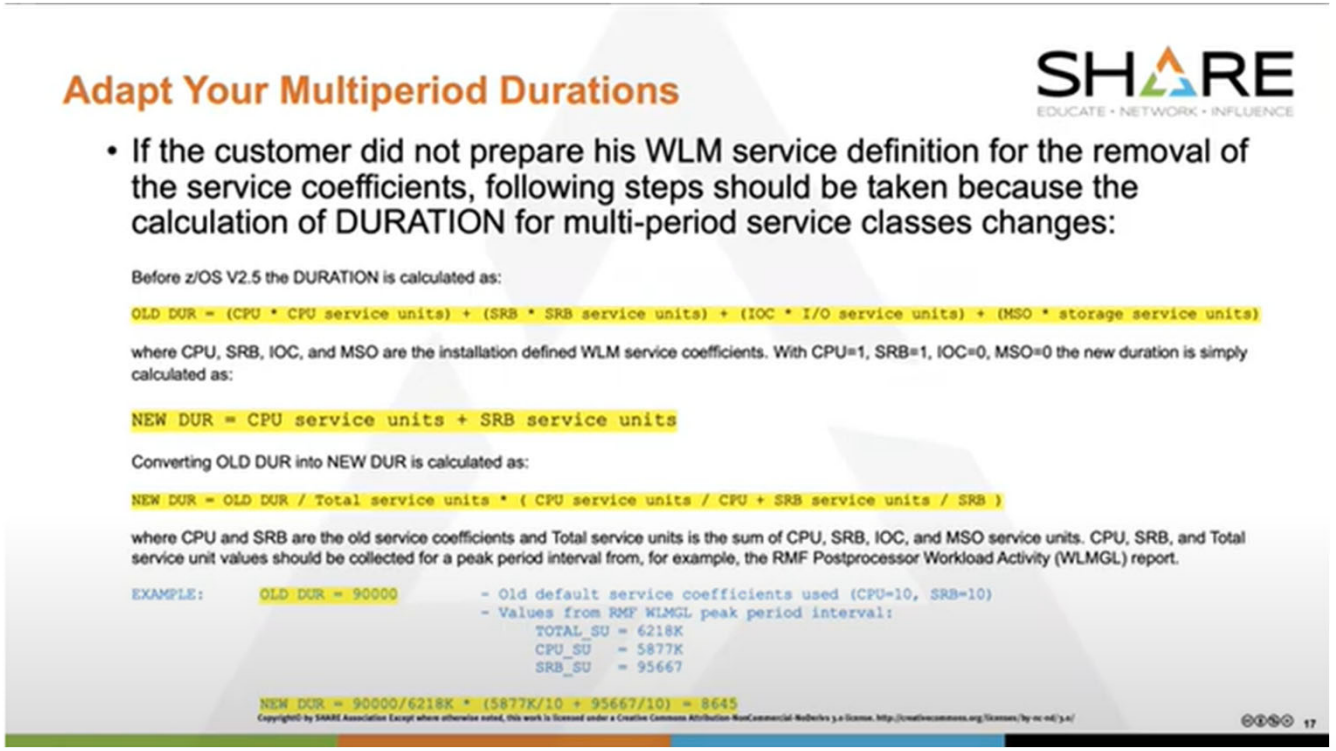
The following is an excerpt from SHARE presentation:
PERFORMANCE INFRASTRUCTURE IMPROVEMENTS IN Z/OS V2.5 WLM
Presenter:
ANDREAS HENICKE (IBM WLM)

Presentation discusses the z/OS 2.5 migration steps suggested to migrate your period durations prior to migrating to z/OS 2.5.

Basically, IBM is suggesting to take CPU and SRB 'service', divide by your current SDCs to convert to 'service units'. Then take the sum of those values and multiple them by the ratio of current duration to service consumed.

Or put a little simpler…
Blah, blah, blah…

Feel free to take this approach, but a bit to complicated for me.



Adapt Your Multiperiod Durations

SHARE
EDUCATE • NETWORK • INFLUENCE

- If the customer did not prepare his WLM service definition for the removal of the service coefficients, following steps should be taken because the calculation of DURATION for multi-period service classes changes:

Before z/OS V2.5 the DURATION is calculated as:

```
OLD DUR = (CPU * CPU service units) + (SRB * SRB service units) + (IOC * I/O service units) + (MSO * storage service units)
```

where CPU, SRB, IOC, and MSO are the installation defined WLM service coefficients. With CPU=1, SRB=1, IOC=0, MSO=0 the new duration is simply calculated as:

```
NEW DUR = CPU service units + SRB service units
```

Converting OLD DUR into NEW DUR is calculated as:

```
NEW DUR = OLD DUR / Total service units * ( CPU service units / CPU + SRB service units / SRB )
```

where CPU and SRB are the old service coefficients and Total service units is the sum of CPU, SRB, IOC, and MSO service units. CPU, SRB, and Total service unit values should be collected for a peak period interval from, for example, the RMF Postprocessor Workload Activity (WLMGL) report.

```
EXAMPLE:    OLD DUR = 90000      - Old default service coefficients used (CPU=10, SRB=10)
                                 - Values from RMF WLMGL peak period interval:
                                      TOTAL_SU = 6218K
                                      CPU_SU   = 5877K
                                      SRB_SU   = 95667

NEW DUR = 90000/6218K * (5877K/10 + 95667/10) = 8645
```


17

Peter's Approach to Migrating SDCs to New z/OS 2.5

- Understand that most durations for multiple periods are usually wrong to begin with.
  - If you feel yours are correct, then do this exercise
- My general approach is a follows:
  1. Determine your current SDCs
  2. Remember the reason you are defining a multiple period service class
  3. Determine your current multiple period service classes
     - Most likely multiple periods are only being used for the following interactive workloads or certain batch
       - TSO, Interactive OMVS, DDF, WAS CB, Batch (sometimes)
  4. Determine which multiple period service classes are consuming I/O service and how much
  5. Then ignore any sort of duration migration exercise for the following enclave workload types since these enclave workloads do not consider I/O service
     - DDF
     - WAS CB
     - So will be left with workloads such as eft with only TSO, interactive OMVS, and Batch,
  6. Revisit duration
     - Either start fresh (which should be done for many periods regardless of this change)
     - Ignore and accept
     - Tweak

Wrap-up

- We hope you enjoyed this and that you've learned something
- Let us know if you like this potpourri of topics format
- We'll be around now and all week for questions

- Questions?

- Please visit our website: www.epstrategies.com
  - Past presentations
  - WLM to HTML tool
  - More information about Pivotor
  - Future educational webinars