

z/OS Performance Topics: The Ones That Divide Us



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



Creators of Pivotor®

**SHARE EPS
Webinar
September 2026**

Peter Enrico
Email: Peter.Enrico@EPStrategies.com

Scott Chapman
Email: Scott.Chapman@EPStrategies.com

Enterprise Performance Strategies, Inc.
3457-53rd Avenue North, #145
Bradenton, FL 34210
<http://www.epstrategies.com>
<http://www.pivotor.com>



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:

© Enterprise Performance Strategies, Inc. All rights reserved. No part of this material may be reproduced, distributed, stored in a retrieval system, transmitted, displayed, published or broadcast in any form or by any means, electronic, mechanical, photocopy, recording, or otherwise, without the prior written permission of Enterprise Performance Strategies. To obtain written permission please contact Enterprise Performance Strategies, Inc. Contact information can be obtained by visiting <http://www.epstrategies.com>.

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

Controversial z/OS Performance Topics

Not all performance topics and recommendations simply cut and dry. Many are controversial. These are the recommendations that tend to generate discussion amongst peers, need careful consideration, or may depend on which 'performance camp' you belong to.

During this presentation Peter Enrico and Scott Chapman will explore some of these recommendations. If you attend this session you are sure to learn something new. The goal of this presentation is to provide you a deeper understanding of these performance topics that don't always have a simple answer.

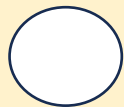
AI Disclosure



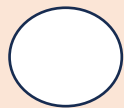
AI Level 0 – AI was not involved in the creation of this content, other than possibly for spelling or grammar checking.



AI Level 1 – AI contributed to non-materially parts of this content, such as transcription, formatting, or clip art. All analysis and conclusions were human-authored.



AI Level 2 – AI assisted in the drafting and analysis. All findings and figures reviewed and confirmed by a human.



AI Level 3 – This was substantially produced by AI and reviewed by a human.

EPS: We do z/OS performance...



- Pivotor - Reporting and analysis software and services
 - Not just reporting, but analysis-based reporting based on our expertise
- Education and instruction
 - We have taught our z/OS performance workshops all over the world
- Consulting
 - Performance war rooms: concentrated, highly productive group discussions and analysis
- Information
 - We present around the world and participate in online forums
<https://www.pivotor.com/content.html>

Like what you hear today?



- Free z/OS Performance Educational webinars!
 - Let us know if you want to be on our mailing list for these webinars
 - Email: contact@epstrategies.com
- 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>

z/OS Performance workshops available



During these workshops you will be analyzing your own data!

- Essential z/OS Performance Tuning
 - March 30 – April 3, 2026 (4 days, excl Wednesday the 1st)
- WLM Performance and Re-evaluating Goals
 - July 27 – 31, 2026 (4 days, excl Wednesday the 29th)
- Parallel Sysplex and z/OS Performance Tuning
 - May 12-13 2026
- Also... please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

Like what you see?



- The z/OS Performance Graphs you see here come from Pivotor™ but should be in most of the major reporting products
- If not, or you just 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>
- 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!

All Charts (132 reports, 258 charts)

All charts in this reportset.

Charts Warranting Investigation Due to Exception Counts (2 reports, 6 charts, [more details](#))

Charts containing more than the threshold number of exceptions

All Charts with Exceptions (2 reports, 8 charts, [more details](#))

Charts containing any number of exceptions

Evaluating WLM Velocity Goals (4 reports, 35 charts, [more details](#))

This playlist walks through several reports that will be useful in while conducting a WLM velocity goal an.

Introduction



- We work with dozens of z/OS shops each year, and we regularly examine the performance at hundreds of systems
- While many performance measurement, analysis, and tuning recommendations are straightforward
 - Each one typically has a 'it depends' escape hatch
 - This is not a presentation about these recommendations
- What we are concentrating on in this presentation are those topics that tend to spark heated discussions, disagreements, and endless forum discussions
 - While there are many of these, this presentation only highlights a few

Should you always tune all your WLM goals?

- Why it matters:
 - Is it worth the exercise to fine tune a goal if an installation's transactions are meeting their business objective?

Point / Counter-Point



● Point

- WLM goals should be well tuned such that the goal is not too hard nor too easy
- The result should be the workloads are assigned a CPU dispatching priority order that reflects the business importance of the workloads

● Counter-Point

- Workloads should be assigned accurate WLM goals based on business requirements
- If the workloads are meeting their business objectives, then why worry about the CPU dispatching priority of the workloads?
- Besides, isn't it a bad practice to for set goals to force certain resource allocation conditions?

Question: Are goals which match actual performance best?



- For many years it has been recommended to set realistic goals for effective WLM management such that the goal is just about 'met'
 - Even if means the work will perform better than what the business needs
- But if the workloads are performing well, does over-performing actually provide additional business value?
 - Why not let WLM manage to goals that match what the business needs?
 - Ensuring unnecessarily good performance for one thing may preclude other work from reaching its needed performance

Question: Aren't easy goals bad?



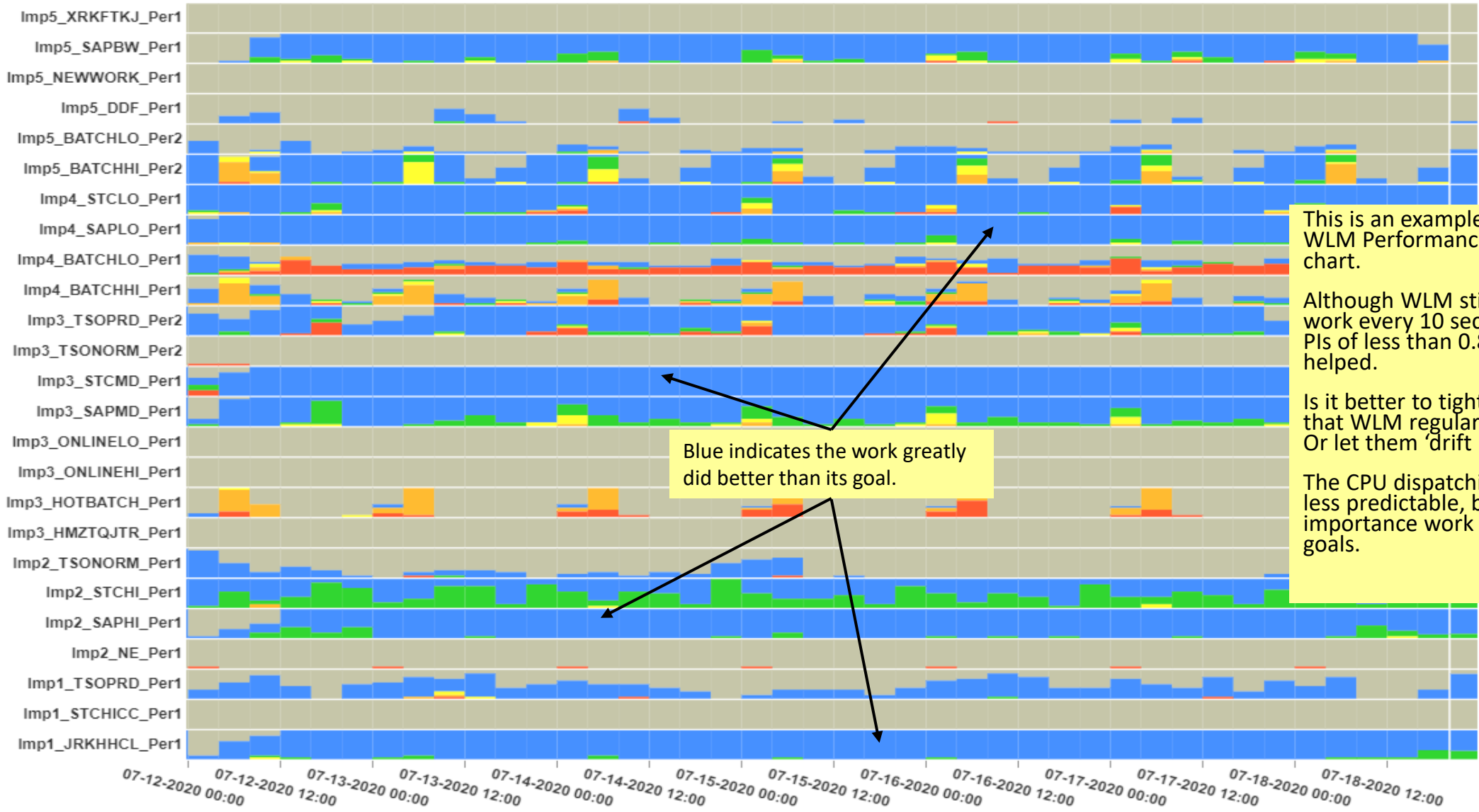
- WLM does not help workloads doing better than 90% of their goal value
 - In other words, if the workload is doing much better than its goal value, WLM does not attempt to help the work, and is more likely to take resources from the work
 - Thus, goals that are too easy will effectively run unmanaged
- But remember: WLM only helps one service class period per 10 second interval
 - Thus, it is very possible that a workload that is “drifting in the wind” may perform poorly for some period of time before WLM notices it needs help
 - “Poorly” is relative: and users accustomed to 0.1s response time may not be happy if WLM lets it degrade to the goal of 1s.

WLM PI - PI Heat Chart for Service Class Periods



- ≤ 0: Zero
- ≤ 0.81: Over Achieving
- ≤ 1.1: Met
- ≤ 1.4: Fair
- ≤ 1.99: Warning
- higher: Severe

PRODPLEX, SYSL



This is an example of a one-week WLM Performance Index (PI) heat chart.

Although WLM still looks at all work every 10 seconds, goals with PIs of less than 0.81 will not be helped.

Is it better to tighten the goals so that WLM regularly helps them? Or let them 'drift in the wind'?

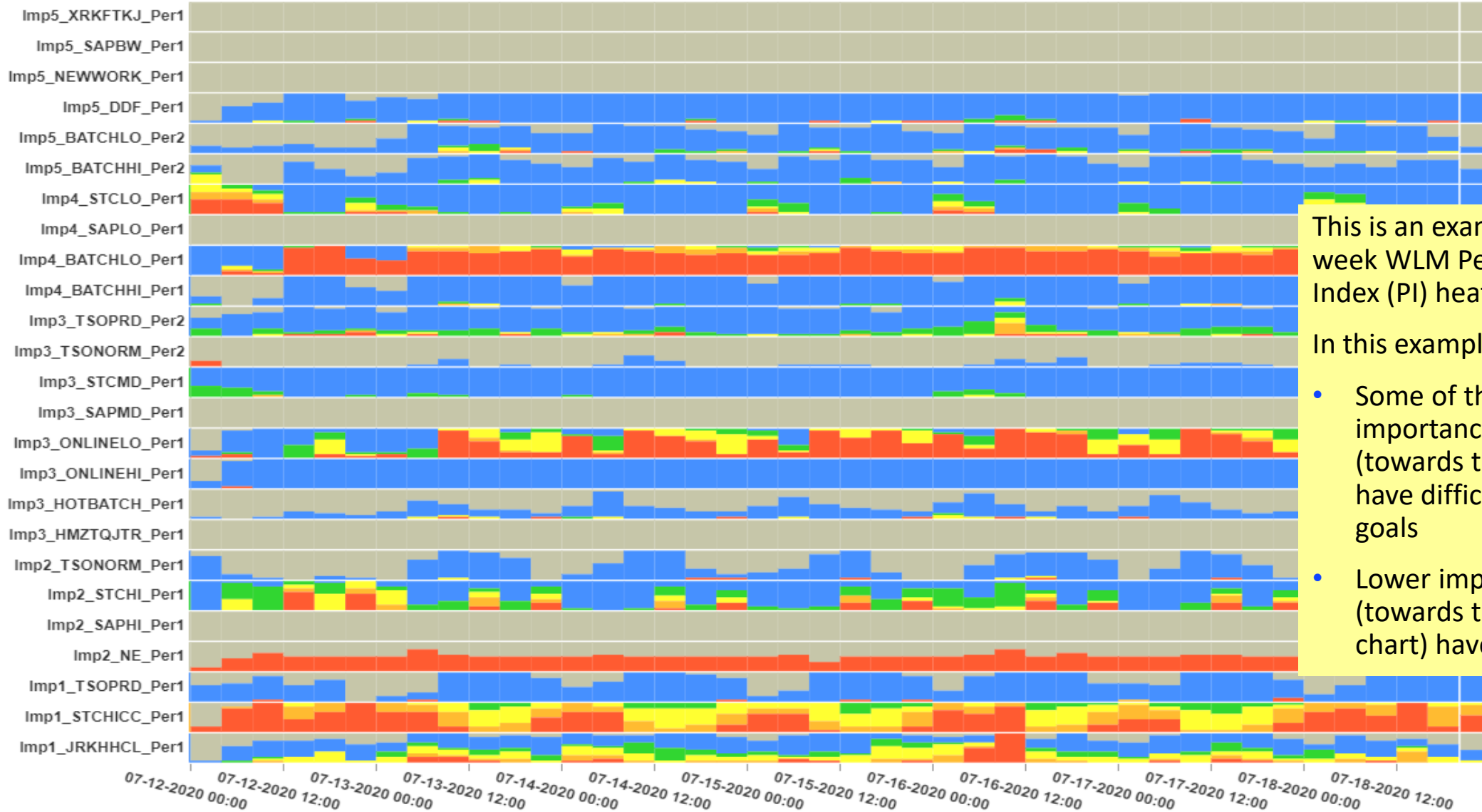
The CPU dispatching priorities are less predictable, but hey... high importance work is meeting its goals.

WLM PI - PI Heat Chart for Service Class Periods



● ≤ 0: Zero
● ≤ 0.81: Over Achieving
● ≤ 1.1: Met
● ≤ 1.4: Fair
● ≤ 1.99: Warning
● higher: Severe

PRODPLEX, SYSK



This is an example of a one-week WLM Performance Index (PI) heat chart.

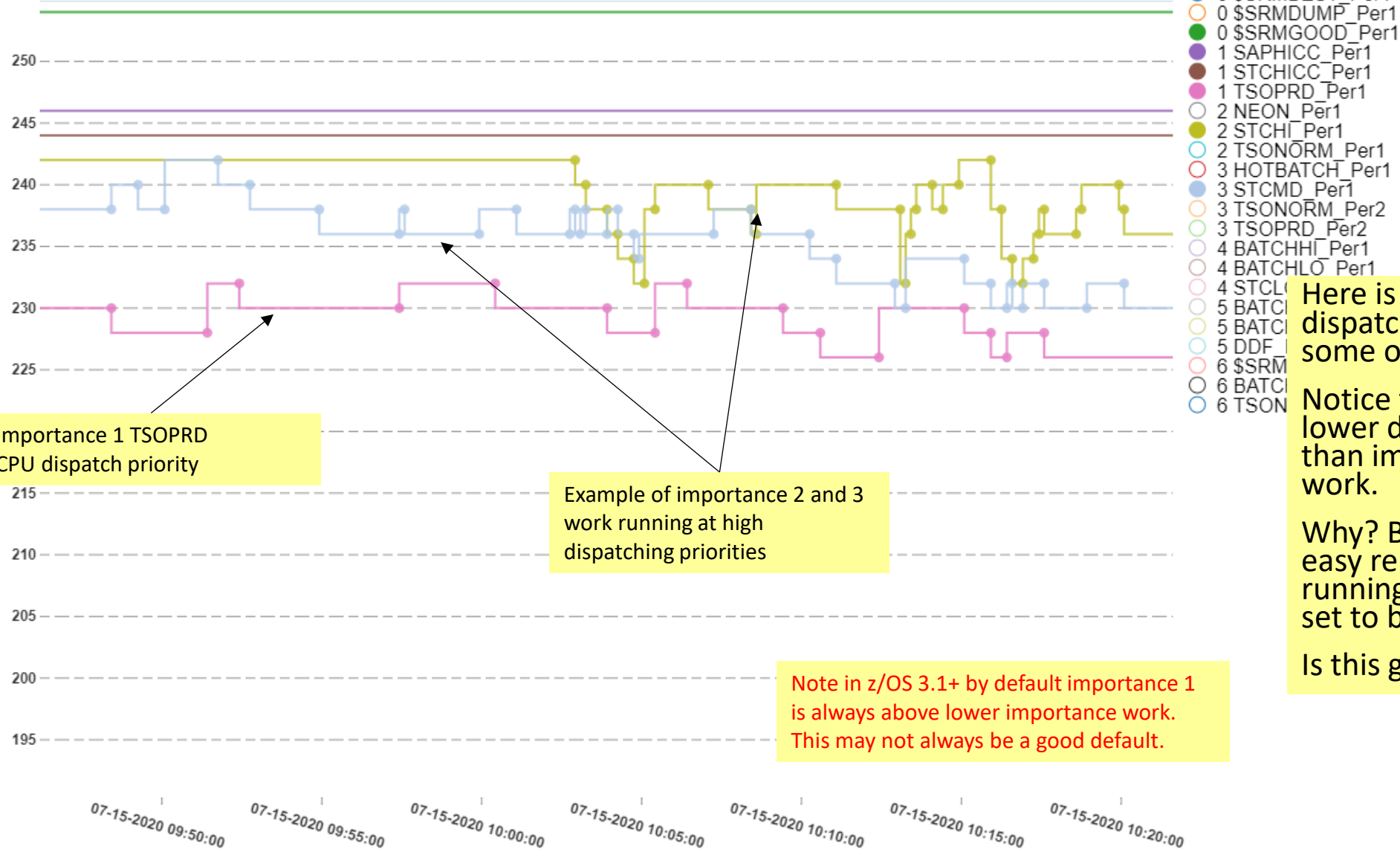
In this example we see:

- Some of the higher importance workloads (towards the bottom) have difficult to achieve goals
- Lower importance goals (towards the top of the chart) have easier goals

WLM SMF 99.6 - CPU Dispatching Priority



SYSK, External



Here is an example of CPU dispatching priorities of some of the workloads.

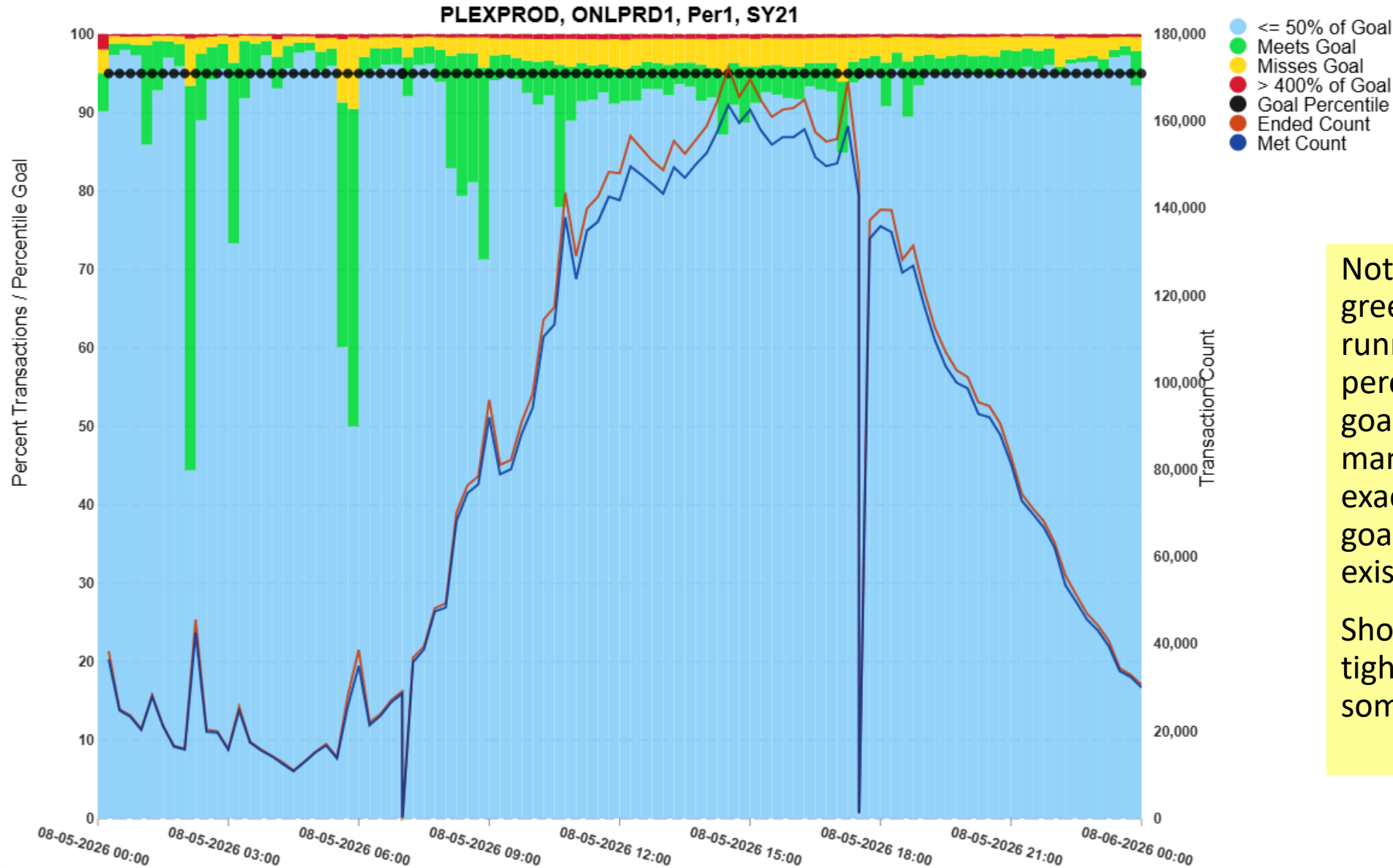
Notice that TSOPRD has a lower dispatching priority than importance 2 and 3 work.

Why? Because its goals is easy relative to the work running, but the goal is set to business objectives.

Is this good or bad?

WLM RT Goal - RTD% of Trans Met/Missed RT Goal with Number Trans

Percent met/missed goal and count



Note the top of the green ("meets goal") is running right at the goal percentage for this RT goal. Either WLM is managing this work exactly to goal or the goal was set to match existing performance.

Should this goal be tightened? We sometimes disagree...

How Should You Use Response Time Goals?

- Why it matters:
 - The defined goals fundamentally affects how WLM manages the work
 - Since the beginning of WLM there has been a bias towards using percentiles for RT goals: does that still make sense?
 - You can now set RT goals down to 1 ms, but should you?

Point / Counter-Point



- Point

- RT goals should be percentiles to limit the impact of long running outliers
- Goals that are too easy can be stolen from and given low CPU dispatching priorities
- Accurately set those single-digit ms RT goals based on what the work can achieve to preserve the work's performance during times of stress

- Counter-Point

- Are the long-running transactions really outliers, or are they the important ones?
- With more systems doing less I/O and faster, do we have as many true outliers as we once did?
- If a workload is regularly and very easily getting a response time of less than 15 milliseconds, will an end user notice a difference of a few milliseconds?

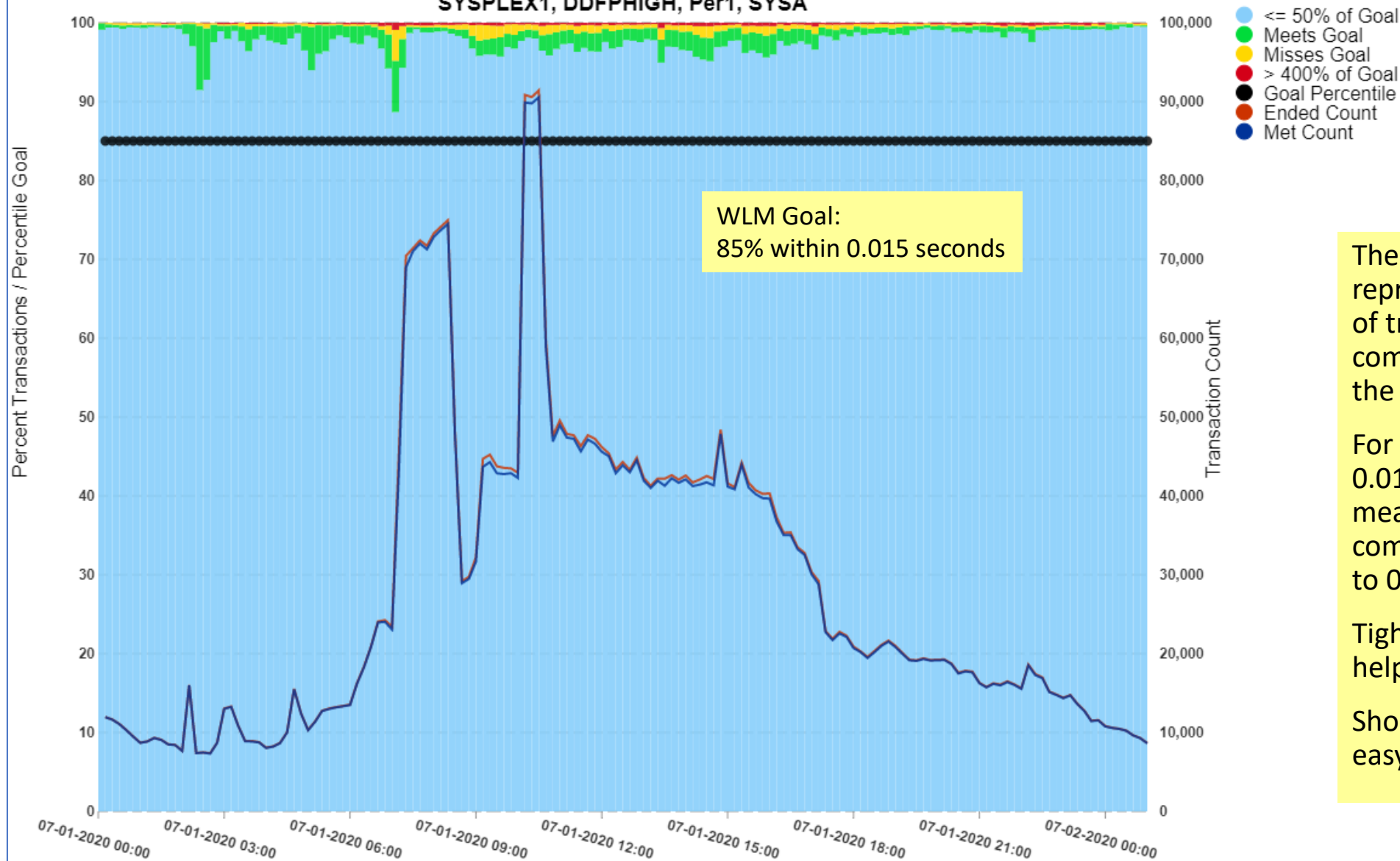
- Counter-counter point

- How many transactions are needed to make up a user interaction?

WLM RT Goal - RTD% of Trans Met/Missed RT Goal with Number Trans

Percent met/missed goal and count

SYSPLEX1, DDFPHIGH, Per1, SYSA



The light blue in this chart represents the percentage of transactions that completed in less than half the goal value.

For a response time goal of 0.015 seconds, this would mean any transaction completing in less or equal to 0.0075 seconds.

Tightening percentile will help a bit, but not much

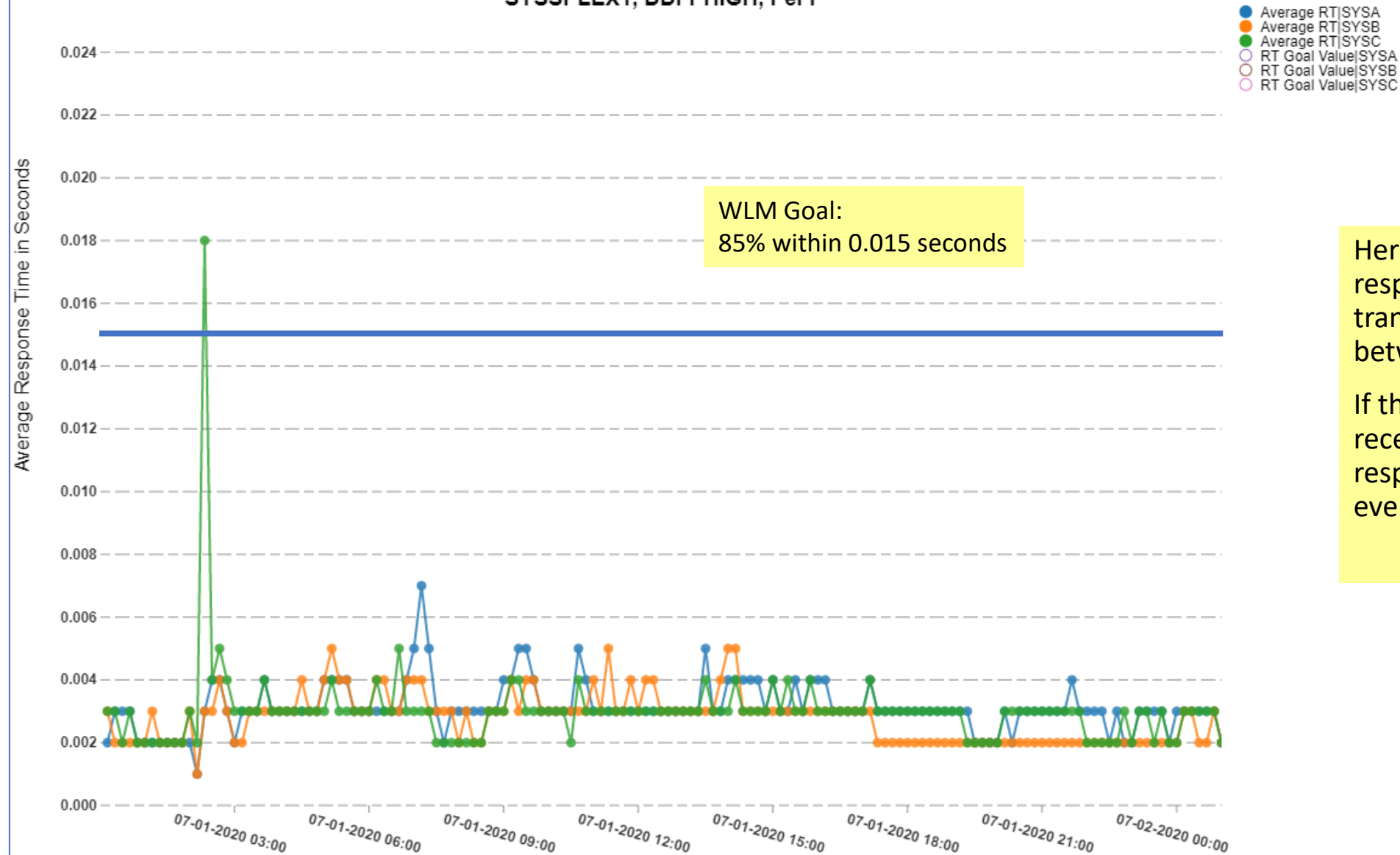
Should we tighten this very easy response time goal?



WLM RT Goal - Average Response Time by Period

(Y-axis limited to 4 seconds)

SYSSPLEX1, DDFPHIGH, Per1



Here we see the average response time for these transactions is usually between 0.002 and 0.004.

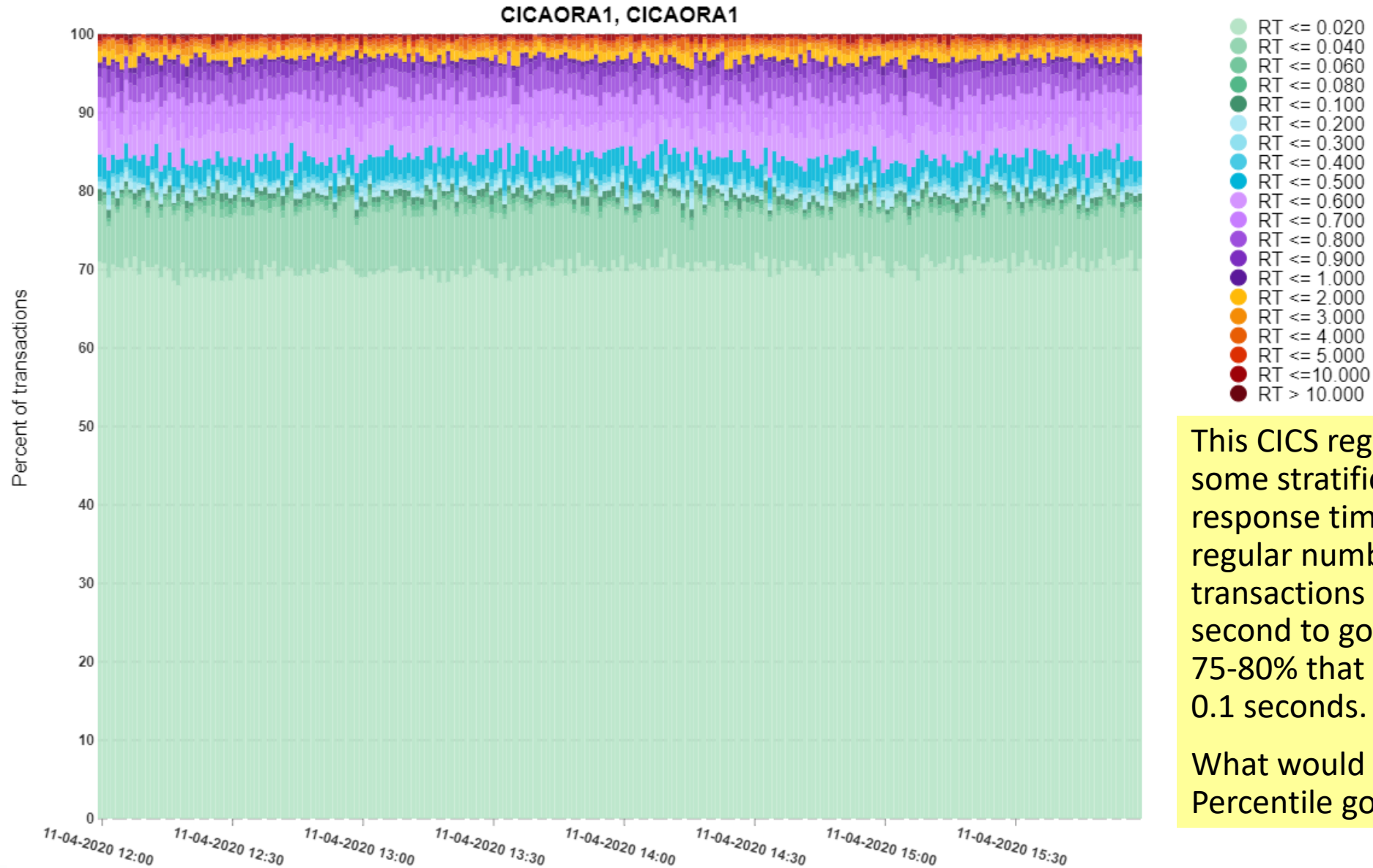
If the users start actually receiving 0.015 second response time, will they even notice?

Question: Who needs single-digit ms RTs?



- If transactions can achieve single-digit millisecond response time, should that be their goal?
 - Nobody notices when their response time changes by even 10s of milliseconds: average human reaction time for visual stimulus is 200-250ms
 - See <https://humanbenchmark.com> to try for yourself
 - When you start managing response times down to single-digit ms, there's a good chance that the network time is going to be longer than the in-host response time
 - Forcing WLM to optimize to save a few milliseconds responses time may result in foregone optimizations elsewhere that would have been more noticeable
- But... user interactions very likely may be made up of multiple transactions
 - The longest-running transactions may be the most important!
 - E.G. "return immediate" transactions in CICS are trivially short and do effectively no work

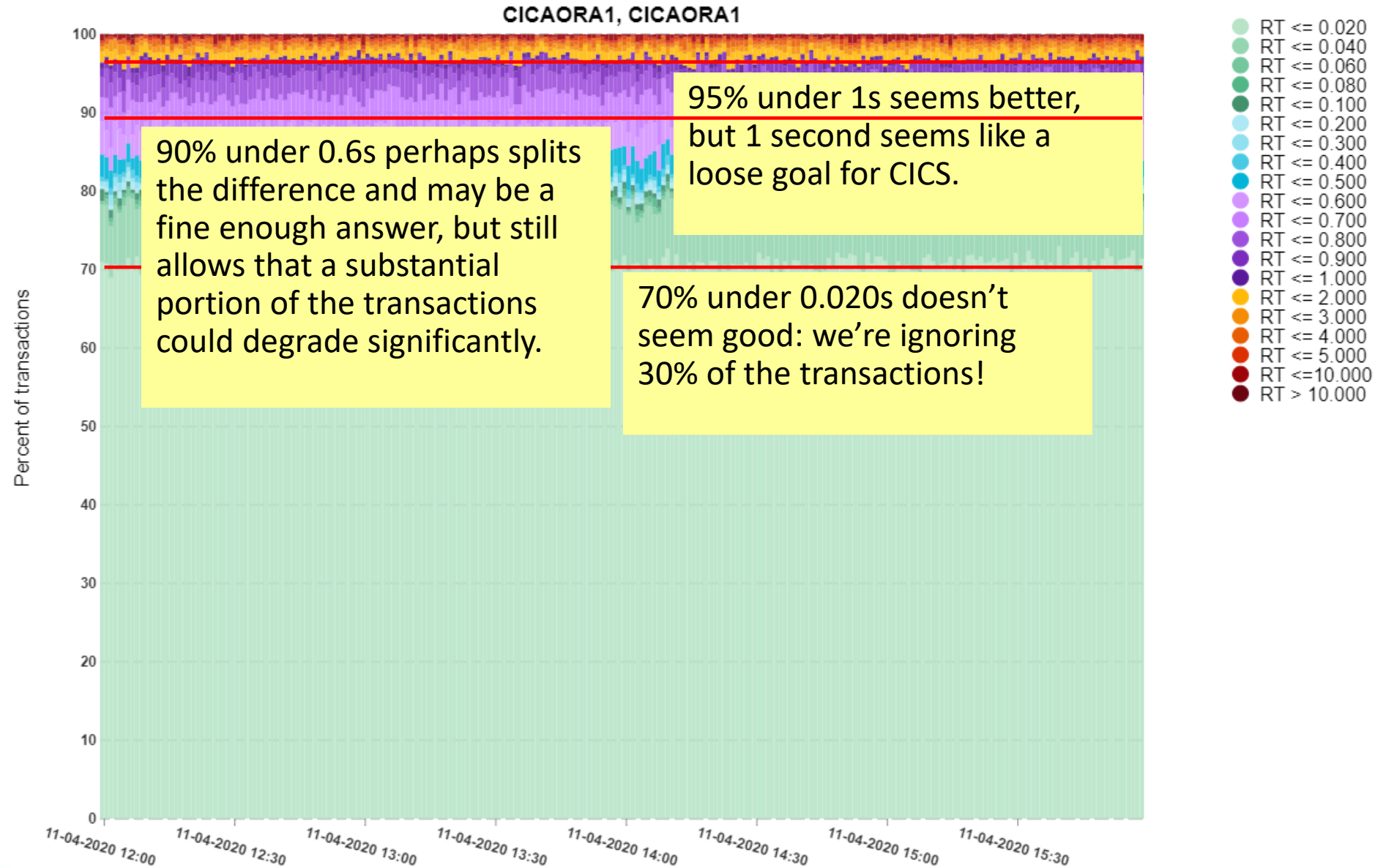
CICS Region Response Time Distribution



This CICS region shows some stratification of response times; there's a regular number of transactions over 1 second to go with the 75-80% that are under 0.1 seconds.

What would be a good Percentile goal here?

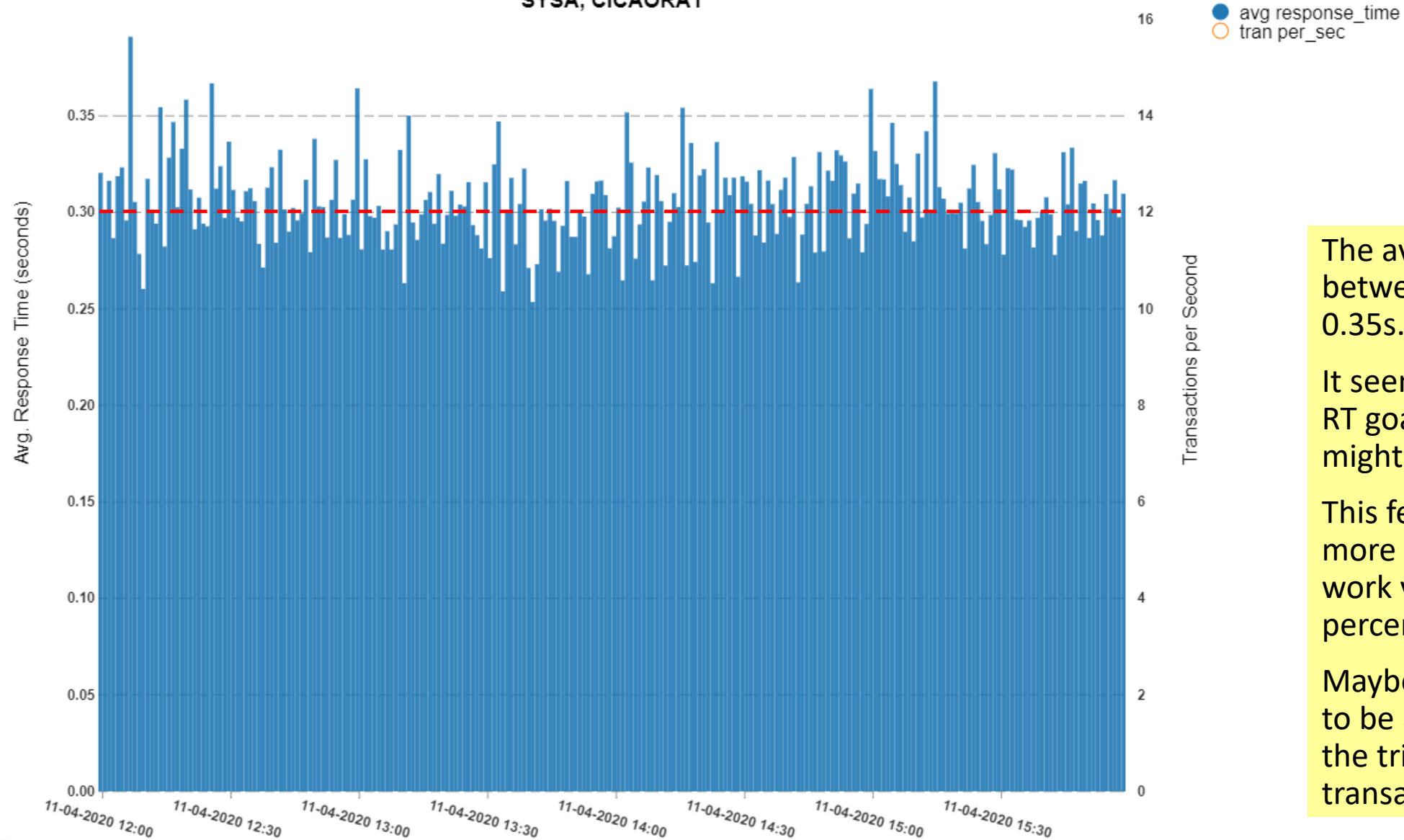
CICS Region Response Time Distribution



Avg Response Time and Rate

(Excluding system transactions)

SYSA, CICAORA1



The average RT runs between about 0.25 and 0.35s.

It seems like an average RT goal of around 0.3s might be good.

This feels like it would more closely control the work vs. using a percentile goal.

Maybe it would be nice to be able to ignore all the trivially short transactions! 😊

How little memory is too little?

- Why it matters:
 - Reserving memory for infrequent activities means that you can't leverage it during the frequent peak times

Point / Counter-Point



● Point

- Maintain generous free memory to avoid paging
- Certain activities might suddenly need a lot of memory, especially during dumps
- Possible recommendation: 1.2x Dump MAXSPACE
 - Db2 MAXSPACE recommendation based on Db2 memory usage but could be 10s of GBs
 - Even though the BPs are generally not dumped (because those can be 100s of GBs)

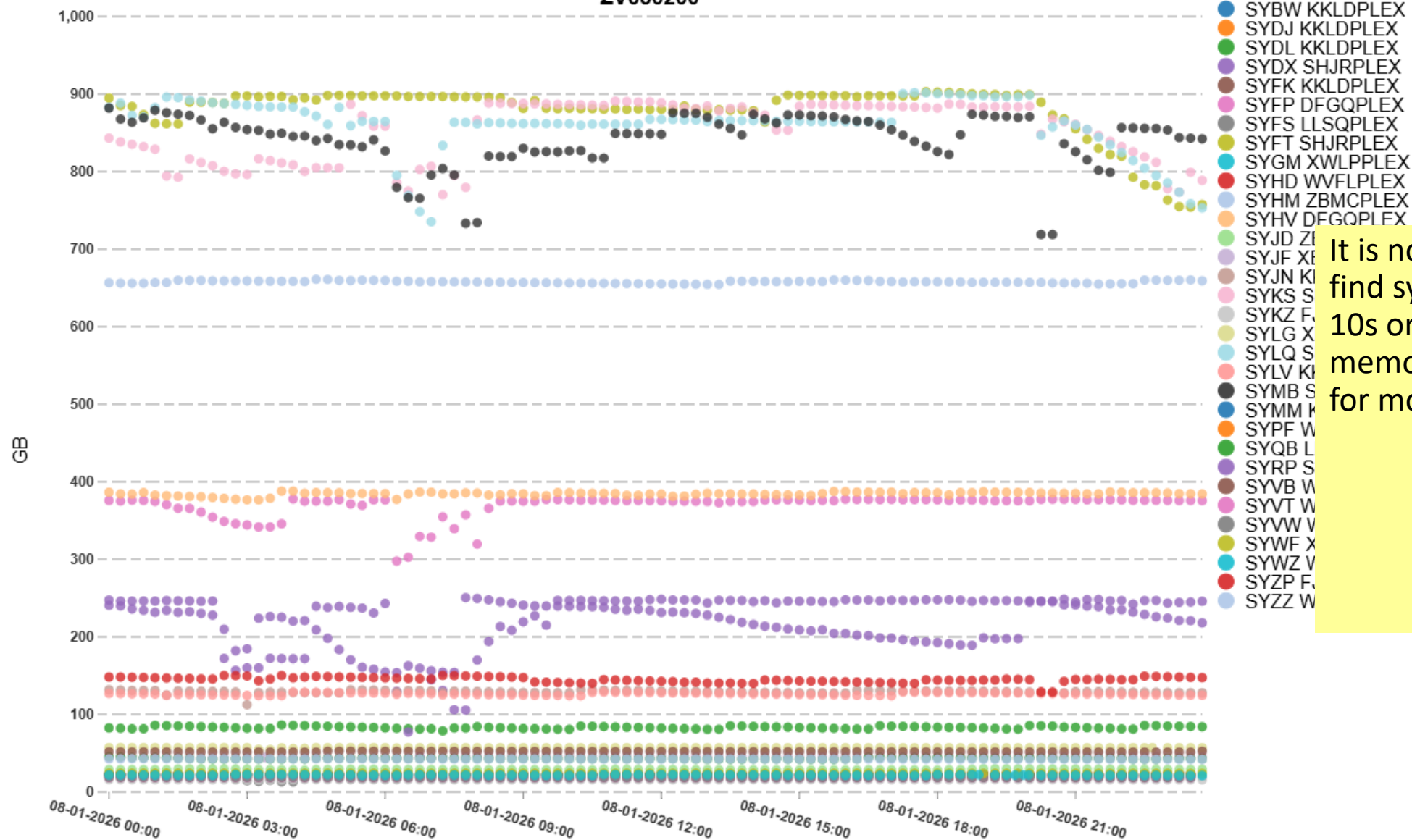
● Counter-Point

- Leaving 10s or even 100s of GBs free “just in case” has a real opportunity cost
 - Can we reduce I/O to drive performance improvements and reduce CPU?
 - How often are you taking dumps of Db2?
 - Pro tip if you are taking large dumps regularly: use System Recovery Boost (req. z16+)
- Virtual Flash Memory reduces paging overhead by paging to memory
 - Similarly speeds up dumps

Central Storage Minimum Available

In GB

ZV030200



It is not uncommon to find systems today with 10s or 100s of GBs of memory that are unused for most or all of the day.

The only good I/O is no I/O

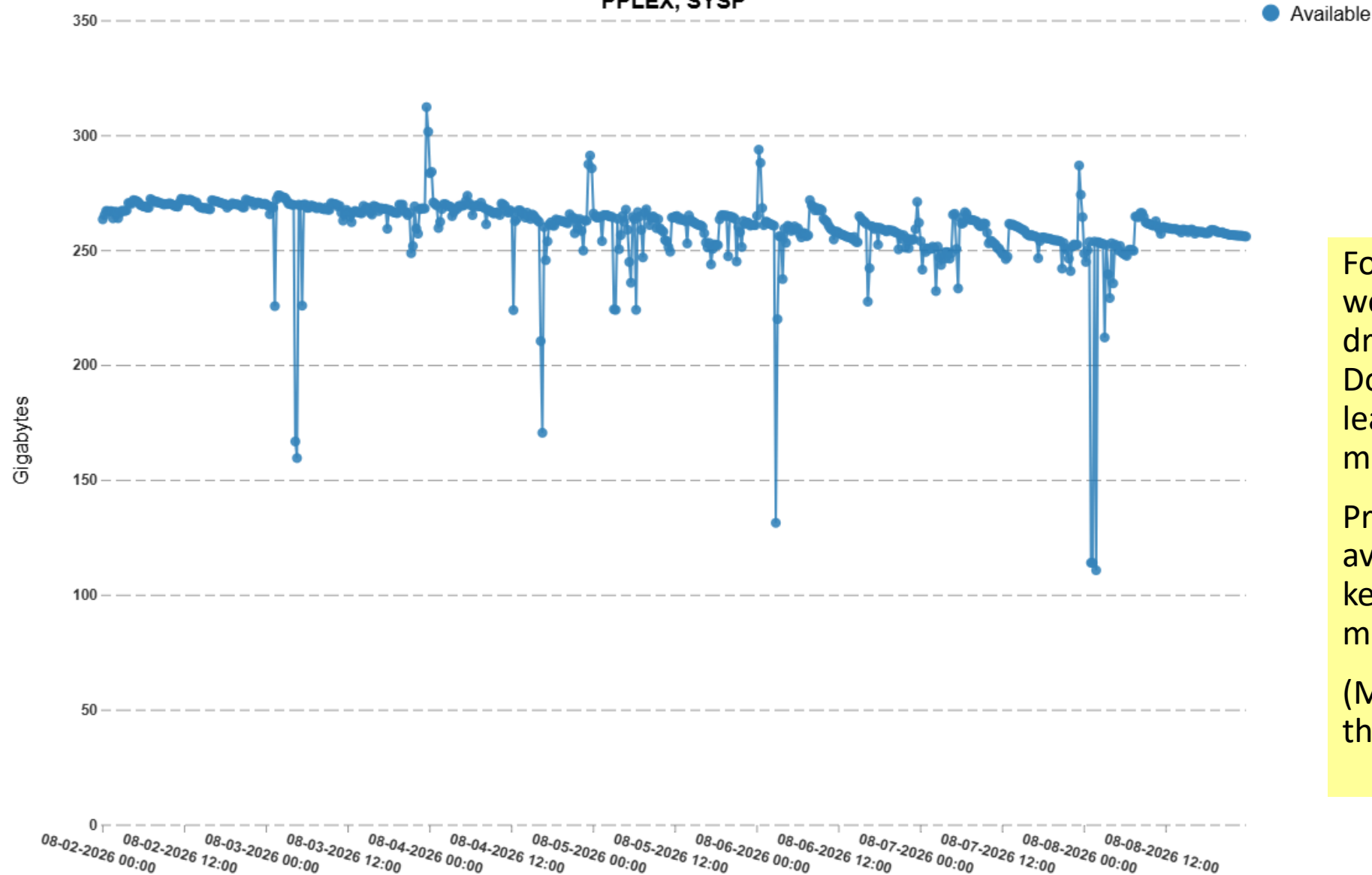


- Yes, I/O can be really fast today, but it still takes time
 - But memory is extremely fast
 - I/O: hundreds of microseconds
 - z/Hyperlink: up to tens of microseconds
 - Memory: fraction of a microsecond
- I/O still takes CPU
 - Giving up the CPU while waiting for the I/O to complete means that when redispached, the work likely won't have its data and instructions in L1 cache
 - zHyperLink spins on CPU while waiting for completion
- Software cost driven by CPU utilization
 - Usually: Software Cost > Hardware Cost
- Performance gated by bottlenecks
 - I/O not always the bottleneck, but is a common one

Minimum Available Central Storage

Gigabytes

PPLEX, SYSP



For this system for this week, min available never dropped below 100GB. Does that mean we should leave 100-200GB of memory unused?

Probably this system could avoid a lot of I/O by keeping more data in memory.

(May want to investigate those dips though.)

zHperLink: Worth implementing?

- Why it matters:
 - Faster I/O is always better, but is it worth the cost and complexity? Or is there a better answer?

Point / Counter-Point



● Point

- Sub 20-microsecond RT is an order of magnitude improvement over traditional I/O
- No redispach means:
 - No I/O interrupt delay
 - Potentially better CPU cache usage

● Counter-Point

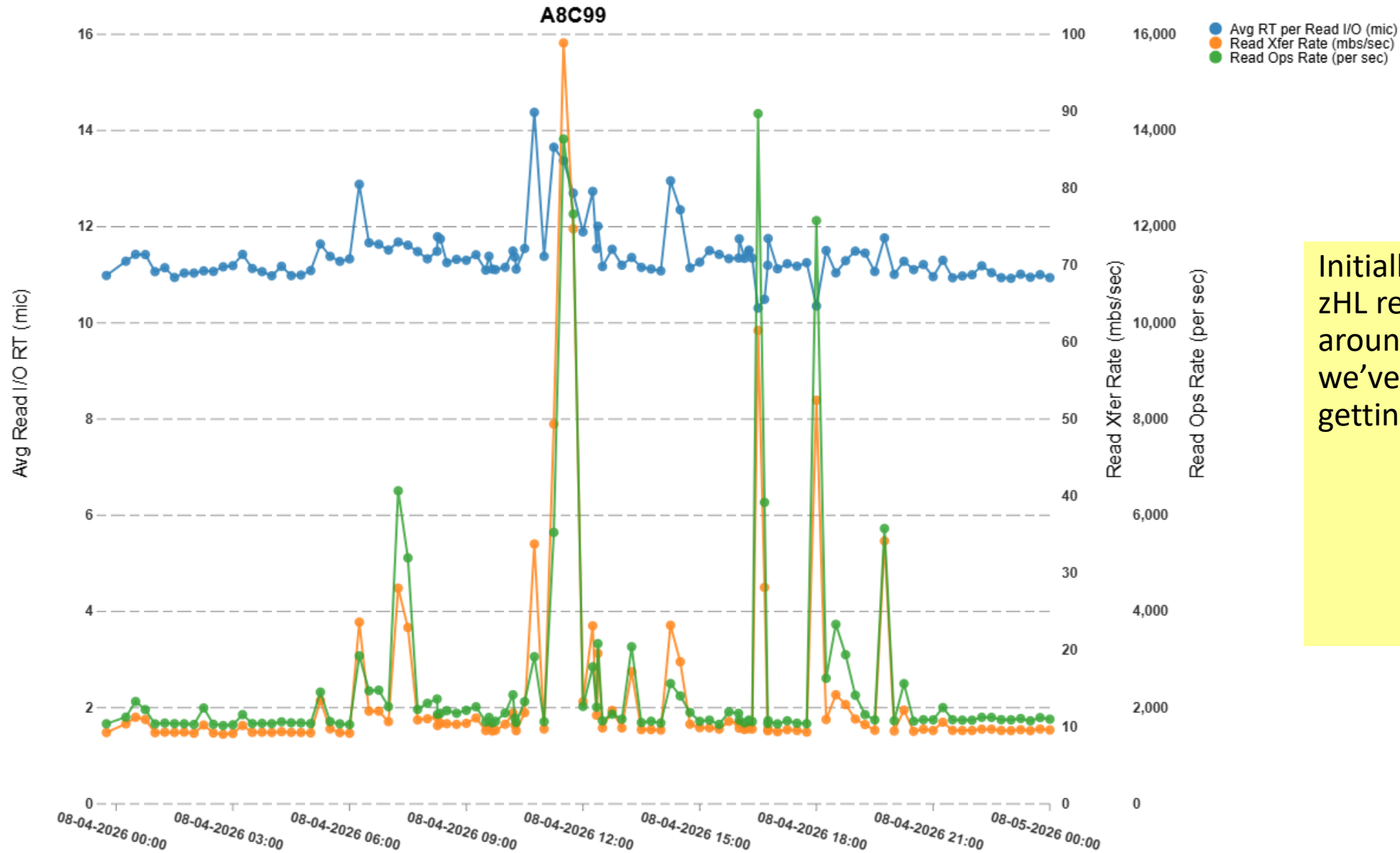
- CPU spins while waiting on the I/O, potentially offsetting CPU cache improvements
- Implementation takes some amount of work + hardware
- Which of your I/Os are actually performance sensitive?
 - E.G. limited impact on performance for making Db2 async prefetch faster

What about memory?



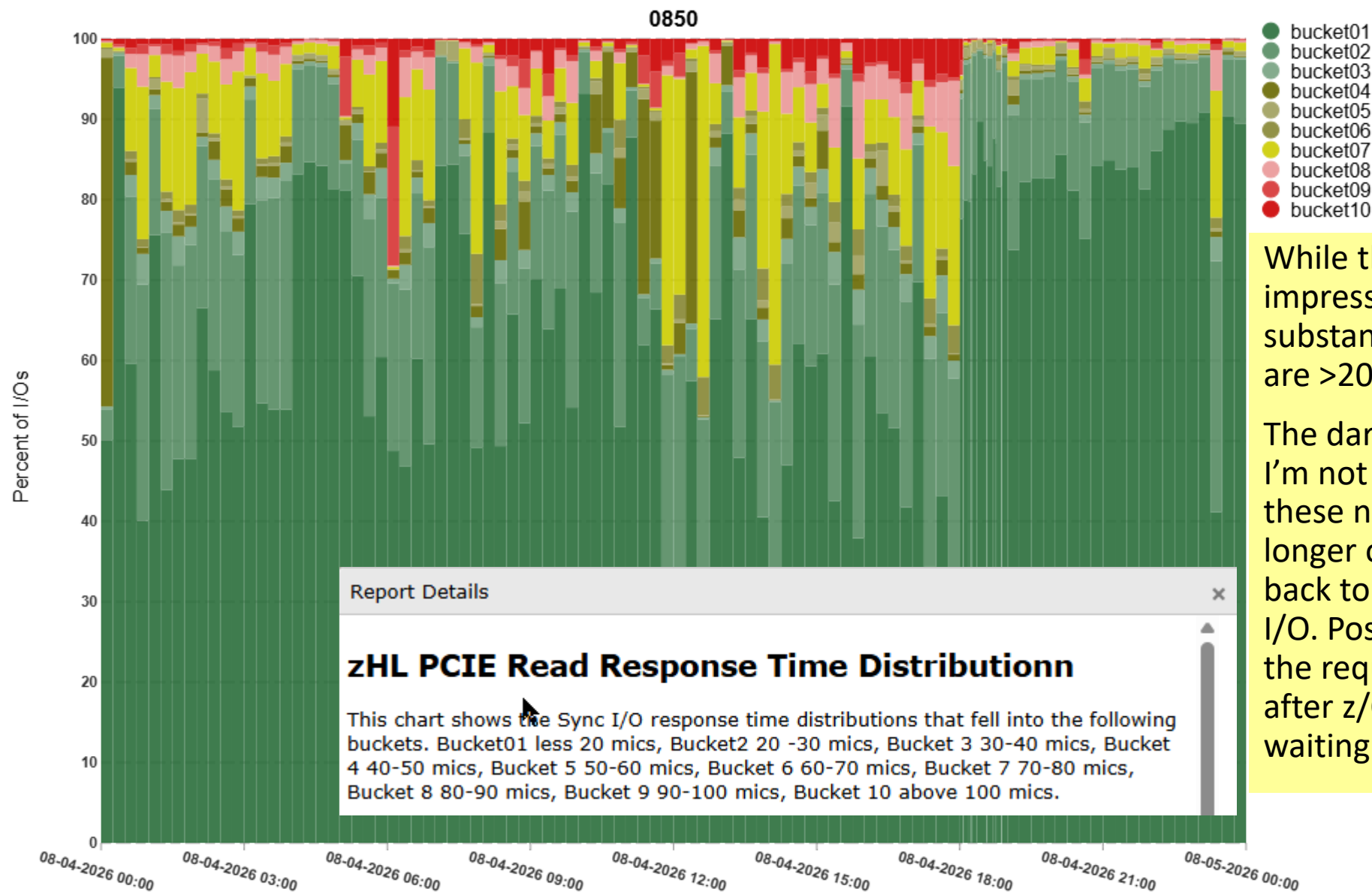
- Not all I/O eligible to be converted to zHyperLink
- zHyperLink promises response times on order of 10-20 μ s *for cache hits*
 - Also, no I/O interrupt delay because processor spins while waiting on the I/O
 - Some of overhead of spinning offset by improved CPU L1/L2 cache hits
 - Unsuccessful zHyperLink I/O will result in driving a FICON I/O to complete the I/O
- This is still much slower than main memory
 - Extrapolating from some published numbers, z16 main memory access might be on the order of 162ns (0.162 μ s)
 - Rough estimate, for memory on the same book
 - L3 would of course be even faster—supposedly on the order of 12ns
- Reading 4K from memory probably 10-100x faster than zHyperlink
 - Means corresponding reduction in the CPU time impact too!

zHL Reads by CEC



Initially we saw average zHL response times around 16-18 μ s but we've now seen some getting down to 11-12 μ s.

zHL PCIE Read Response Time Distributionn



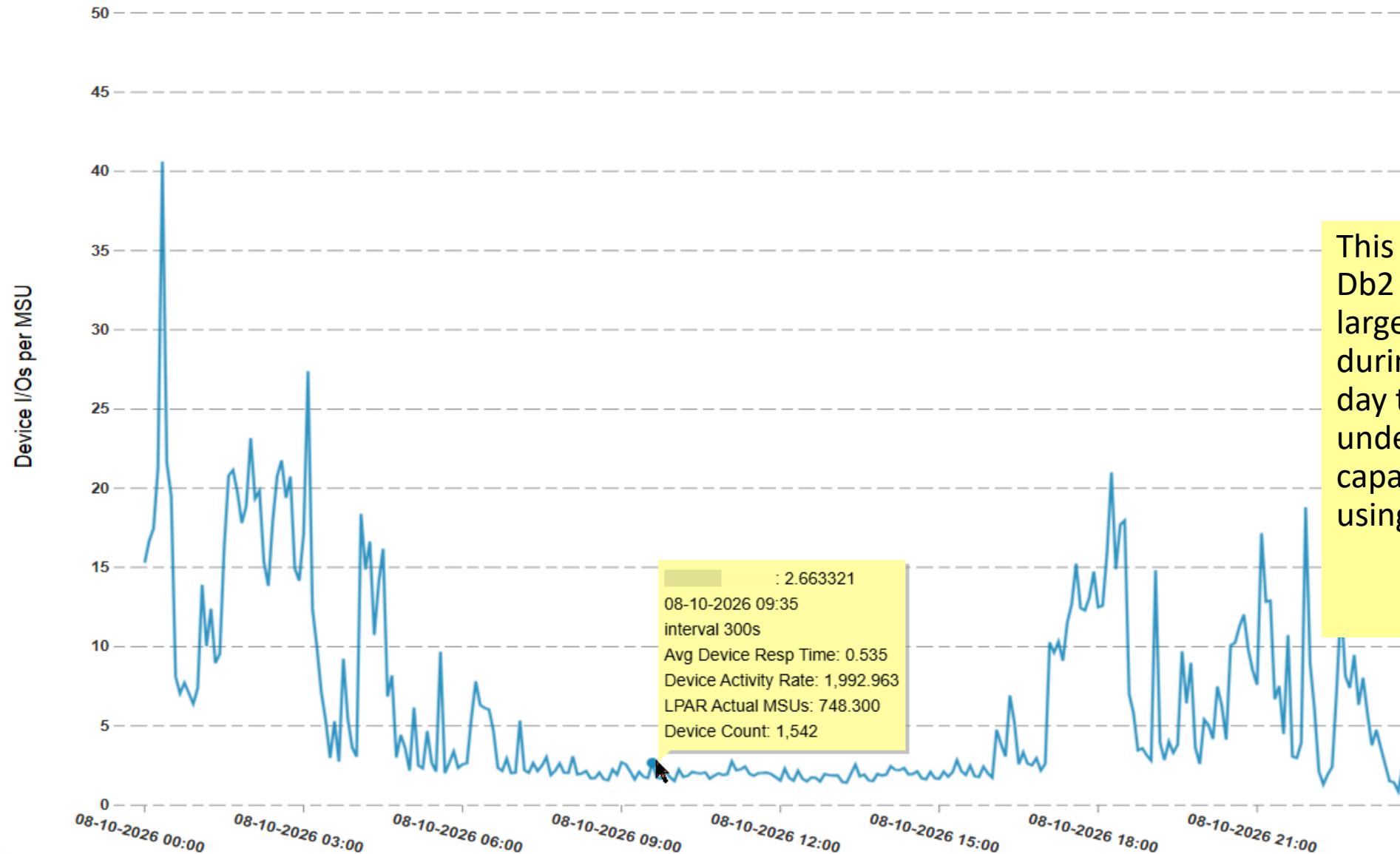
While those averages are impressive, there can still be substantial number of I/Os that are $>20\mu s$.

The dark green here is $<20\mu s$. I'm not sure how RMF captures these numbers given that those longer ones should have fallen back to drive a regular FICON I/O. Possibly the PCIE captures the request completion even after z/OS has given up on waiting for it.

Device I/Os per MSU Consumed by System

All systems, ratio per interval

-alldata-



This system is primarily Db2 DDF activity and has large Buffer Pools and during their busy time of day they're running under 5 IOPS per MSU of capacity that they're using.

Physical or Logical Utilizations?

- Why it matters:
 - Utilization percentages are commonly used, but do you know what they're a percentage of??

Point / Counter-Point



● Point

- Logical CPU utilization represents the consumption of the logical CPU capacity consumed which is different than physical CPU utilization which represents the amount of the physical CPU capacity of the CEC consumed.
 - If number of logical CPUs equals the number of physical CPU, the processor capacity is the same

● Counter-Point

- Logical utilization important if you're constrained on logical CPs, but with HiperDispatch people are no longer micromanaging the number of logical CPs
 - Probably more systems have too many logical CPs vs. too few
- Measurement that represent how much of the machine capacity the work is using can be more intuitively useful and allows comparisons across LPARs

Workload Utilization



- Workload Utilization = $\text{CPU Time} \div \text{Interval Length} \div \text{Online Logical CPs}$
 - Some might suggest “online and unparked” instead of “online”
- This expresses the usage as the utilization of the *logical* CPs
 - The LPAR can't use more than the sum of the logical CPs, so this may be useful limit to be aware of
- If logical CPs = physical CPs, then this is also a utilization of the physical CPs
 - I.E. percentage of the machine capacity
- If logical CPs $\neq$ physical CPs, then it represents a utilization of a subset of the machine capacity
 - And you can't sum workload utilization across multiple LPARs because the total logical CPs across the LPARs is probably greater than the number of physical CPs

Physical consumption

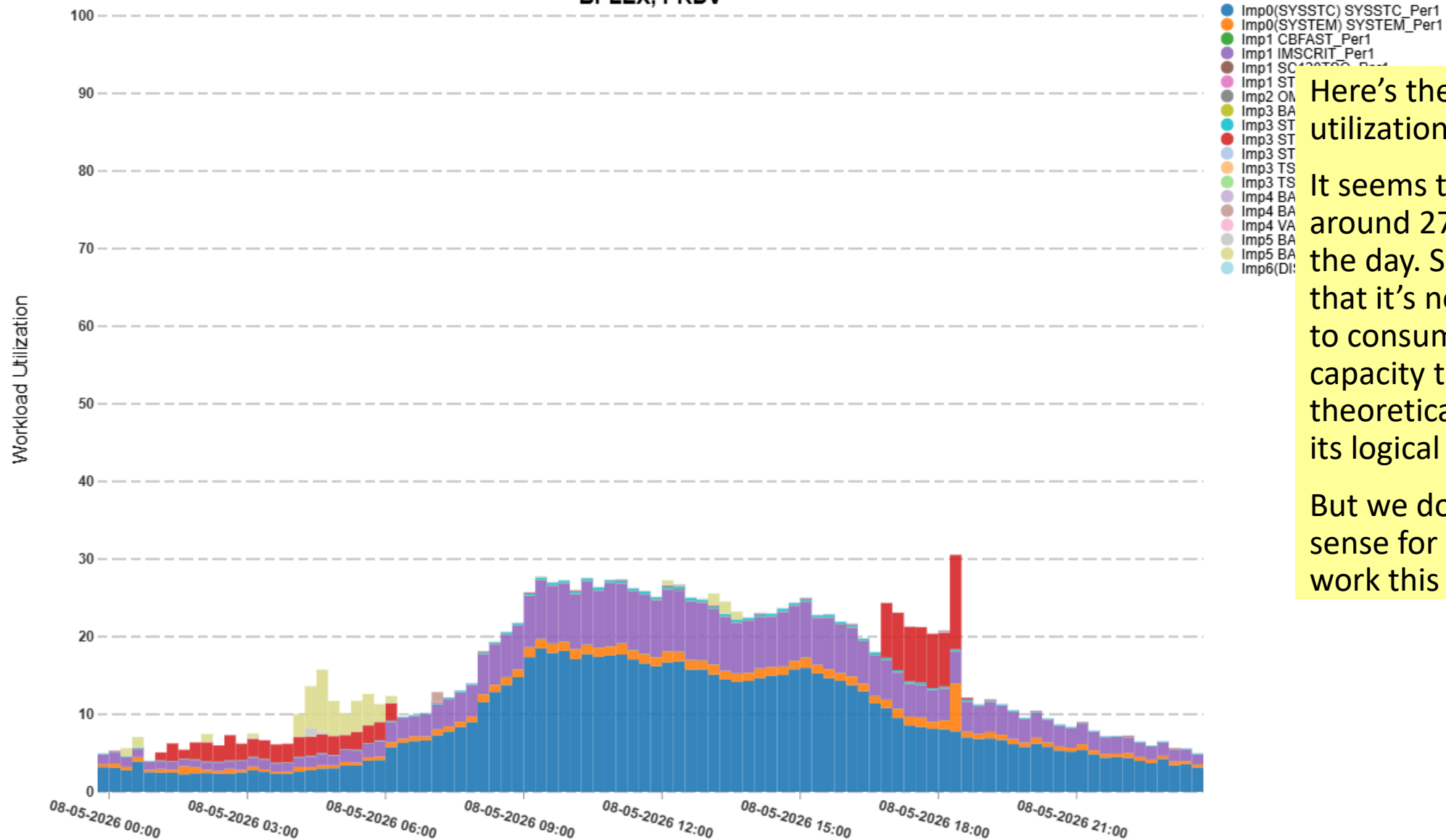


- CPU time
 - zArchitecture has had accurate measuring of CPU time for decades
- $APPL\% = CPU\ Time \div Interval\ Length$
 - Percentage of a single physical CP
- Measures based on ratings:
 - $SUs = CPU\ Time * SU/sec\ rating$
 - $MSUs / MIPS = APPL\% * MSUs\ or\ MIPS\ per\ physical\ CPU$
 - $MSU\text{-}hours\ (for\ TFP) = CPU\ Time\ (in\ hours) * MSUs\ per\ physical\ CPU$

WLM CPU - CP CPU Workload% by Service Class

(CP + zIIP on CP)

BPLEX, PRDV

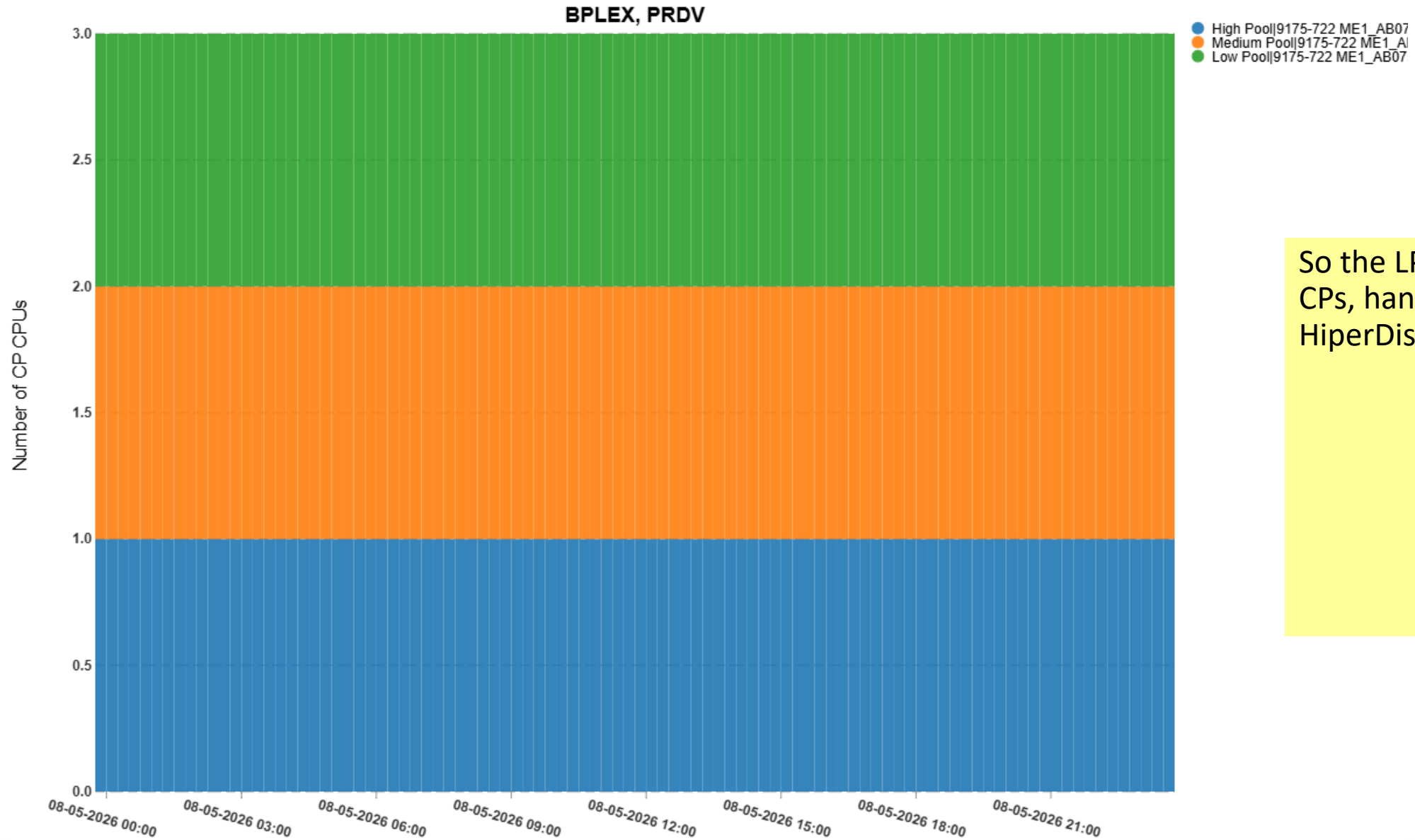


Here's the workload% utilization for a system.

It seems to be running around 27% busy during the day. So we can say that it's not coming close to consuming the capacity that it could theoretically get to with its logical CPs.

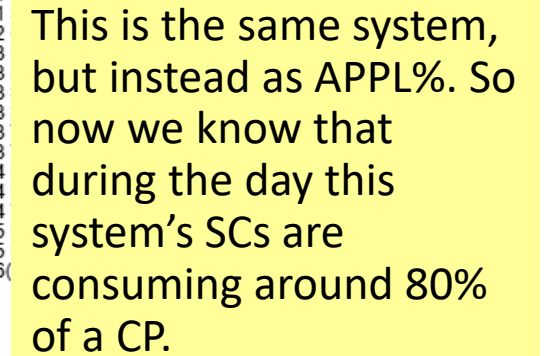
But we don't have any sense for how much work this really is.

HiperDispatch CP CPU Pooling at End of Interval



So the LPAR has 3 logical CPs, handily one of each HiperDispatch polarity.

(CP + zllP on CP)



Note that it's the same profile as the prior chart.

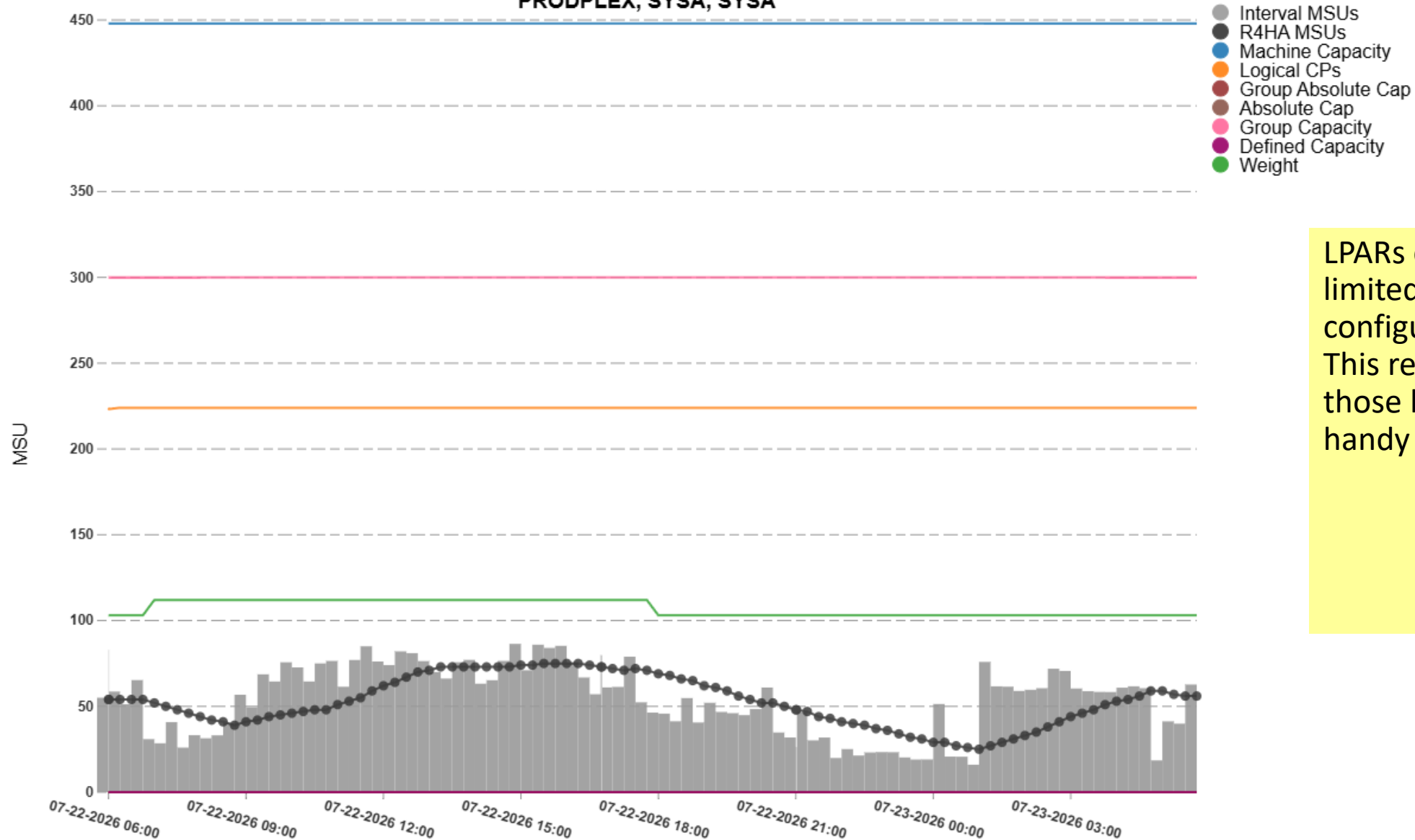
Also, $.27 * 3 = .81$

If this is your system, you probably know how many physical CPs you have. I don't of course, but we can look at the overall CEC utilization.

LPAR Limits and Utilization

Expressed as MSUs

PRODPLEX, SYSA, SYSA



LPARs can actually be limited by multiple configuration settings. This report summarizes those limits in one handy place.

Is conflating zIIP and GCP measurements a good practice?

- Why it matters:
 - Conflating the zIIP and GCP measurements can misrepresent the overall efficiency delivered by each processor pool.

Point / Counter-Point



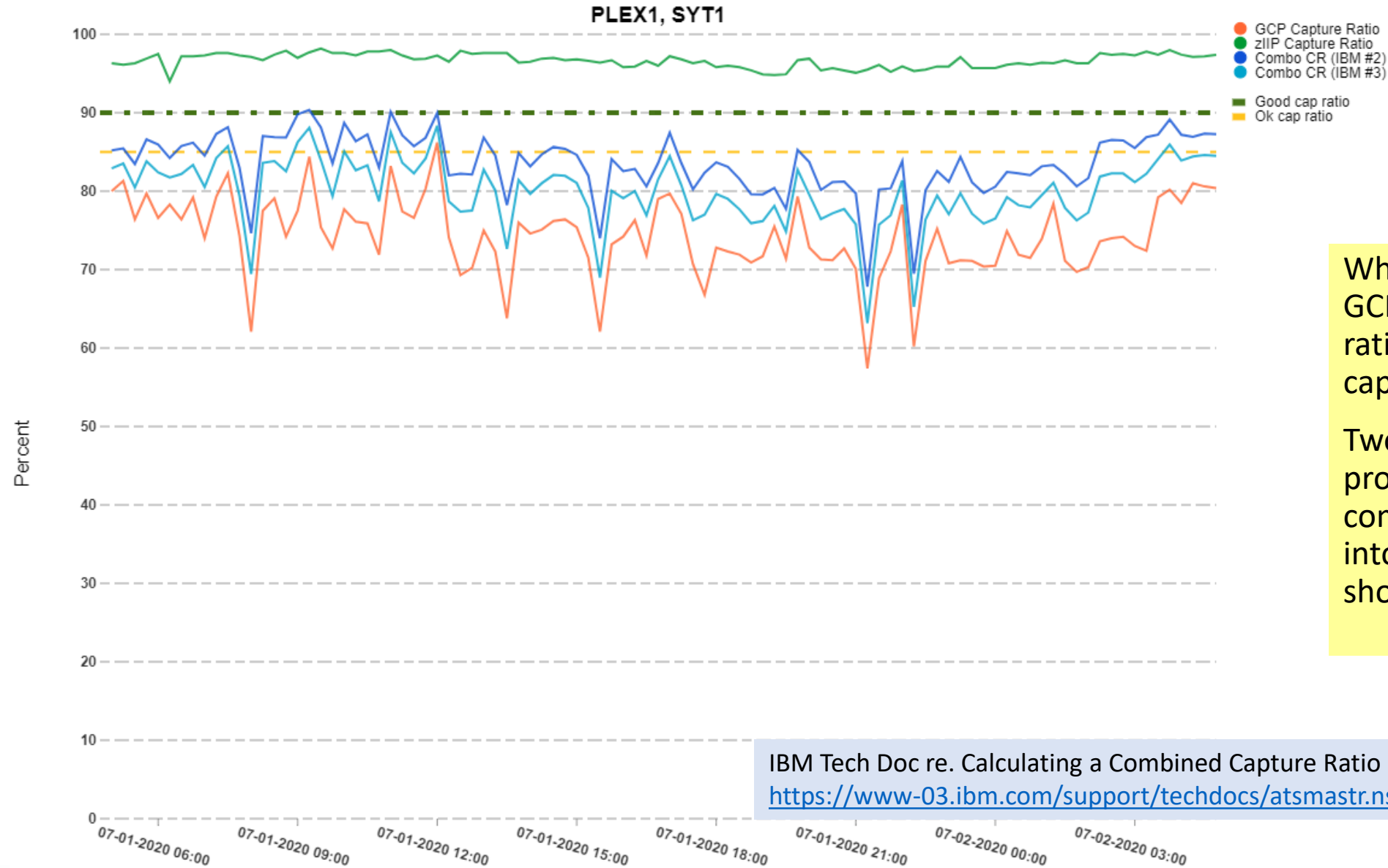
● Point

- Always separate your processor measurements by processor type (GCPs, zIIPs)
 - Not all work can run on a zIIP so combining capacity measures is meaningless
 - The zIIP and GCP pools are managed separately
- Performance of the zIIPs and GCPs is often different
- Combining zIIP and GCP times sometimes involves combining unlike capacities or making equivalency estimates

● Counter-Point

- Sometimes it is nice to get a viewpoint of your overall processor capacity.
 - Combine the measurements from zIIPs and GCPs because you have work running across both processor types
- Looking at them independently misrepresents your overall efficiency of the processors

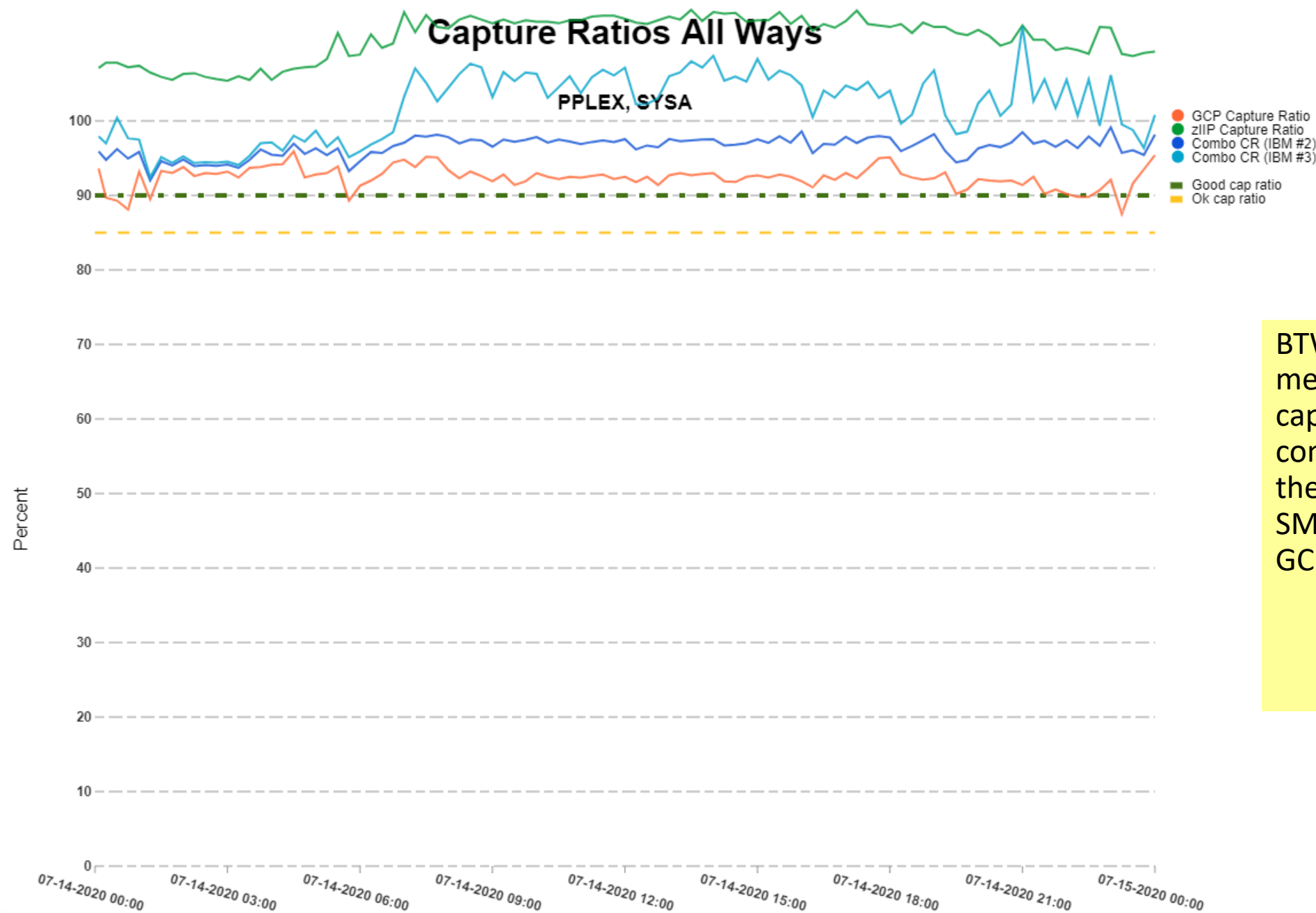
Capture Ratios All Ways



What if we compare the GCP and zIIP capture ratio to a combined capture ratio?

Two different methods proposed by IBM for combining GCP and zIIP into one capture ratio shown here.

IBM Tech Doc re. Calculating a Combined Capture Ratio
<https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102717>

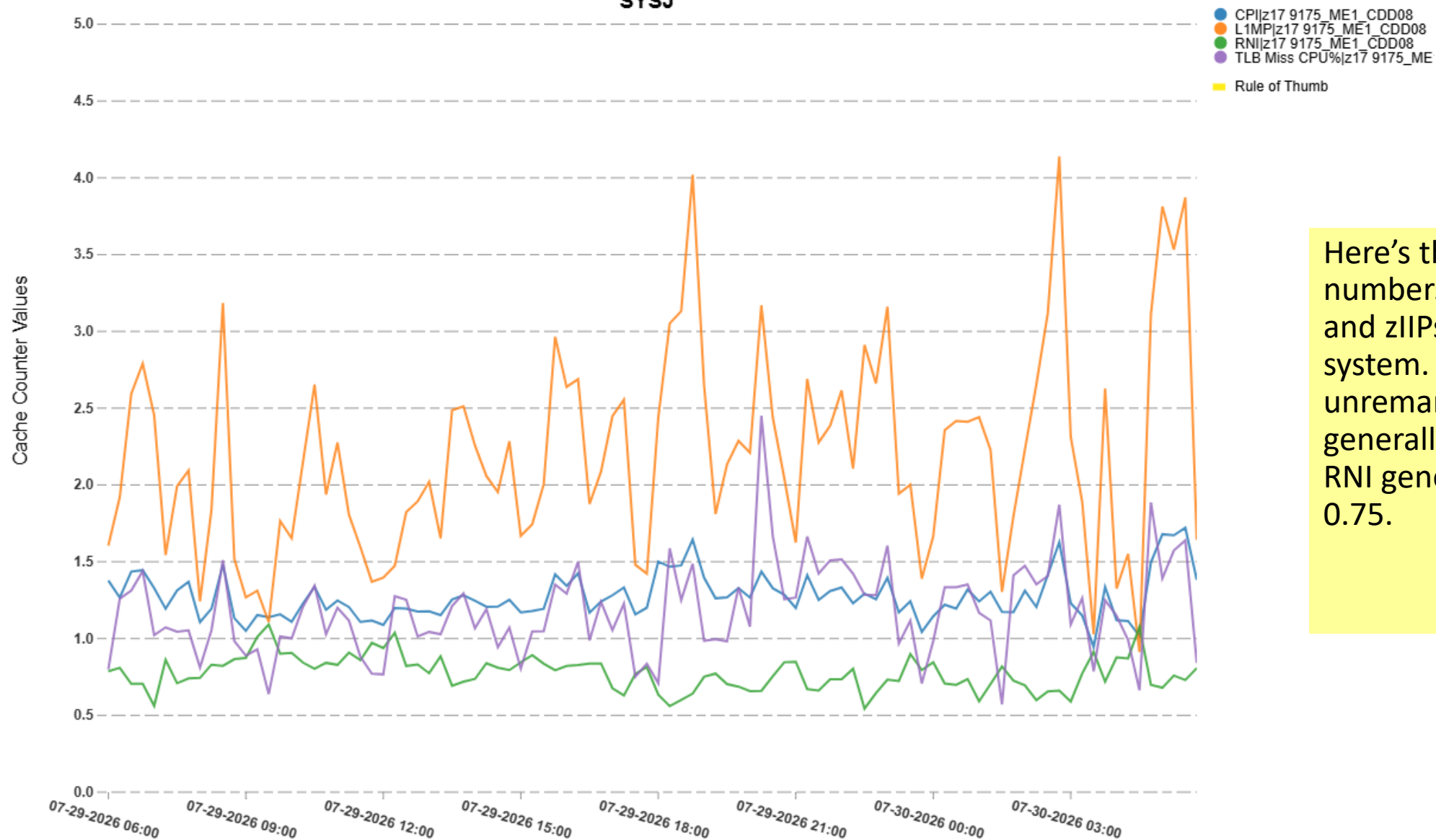


BTW, SMT can really mess with the zIIP capture ratios. If you combine zIIP and GCP then you've allowed SMT to mess with your GCP capture ratio too.

Processor Caches - Key Measurements for Processors

SMF 113

SYSJ

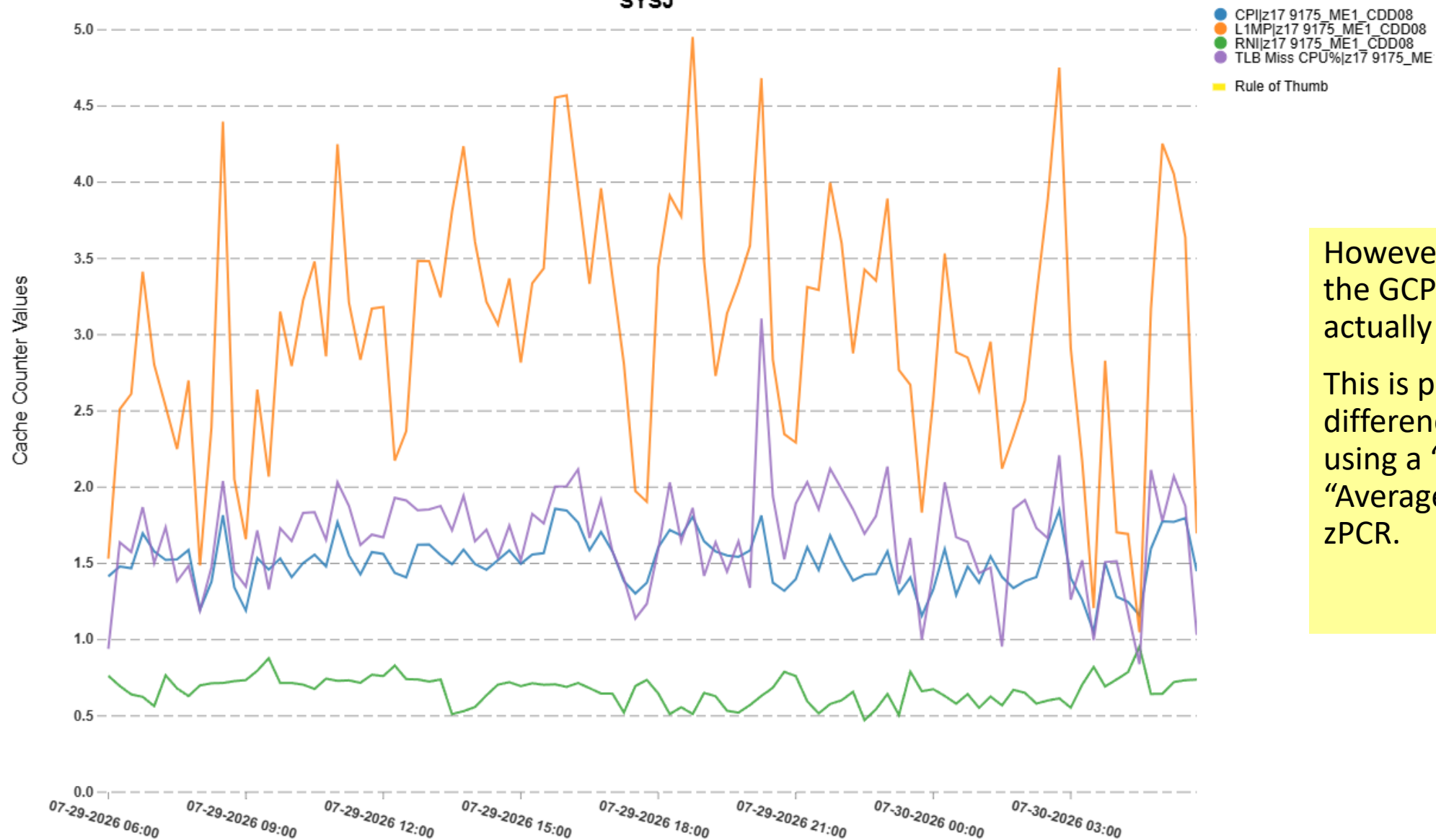


Here's the major HIS numbers for both GCPs and zIIPs combined for a system. Seems mostly unremarkable. L1MP generally under 3 and RNI generally around 0.75.

Processor Caches - CP CPU Key Measurements

SMF 113

SYSJ



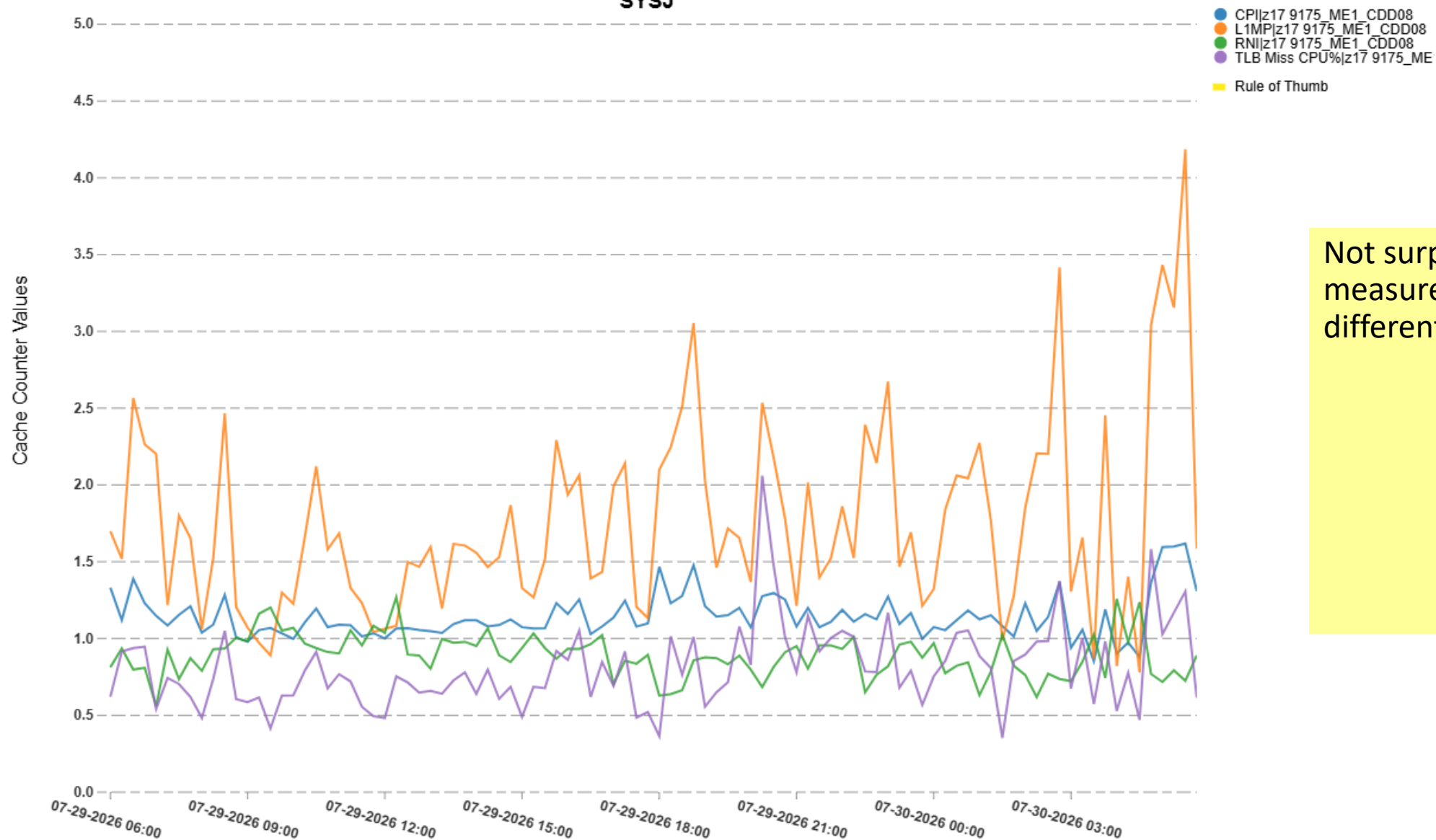
However, looking at just the GCPs, that L1MP is actually regularly over 3!

This is potentially a difference between using a “Low” or “Average” workload in zPCR.

Processor Caches - zIIP CPU Key Measurements

SMF 113

SYSJ



Not surprisingly, the zIIP measurements are different.

Should CPU Critical Control be used? (Knowing it will take some control away from WLM)

- Why it matters:
 - The CPU critical control may inhibit and limit WLM's optimal CPU dispatching decisions.

Point / Counter-Point



● Point

- The WLM CPU adjustment algorithms were developed to optimal assign CPU dispatching priorities based on goals.
- Set the goals correctly, and then let WLM do its thing
- There may be some select cases for limited use of CPU Critical

● Counter-Point

- Many installations do not want to take the risk of a workload getting a less than predictable CPU dispatching priority
- This is especially true on low n-way machines and LPARs

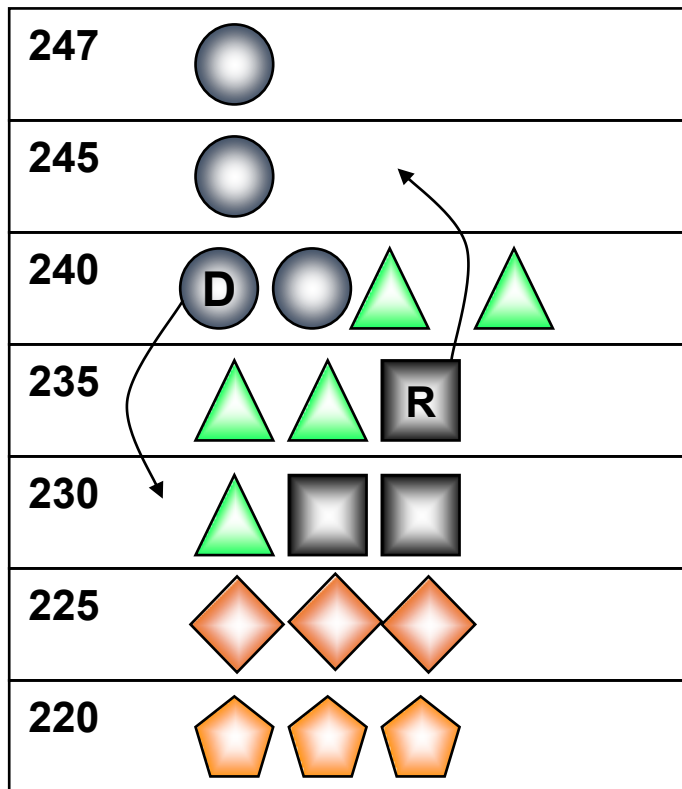
● Counter-Counter Point

- In z/OS 3.1, by default, all importance 1 workload (that is the first period) is treated as CPU critical(!)

CPU Critical Control- Background



- Control created many years ago when some installations would not migrate to WLM for fear of poor CPU dispatching priority decisions



- With well set predictable goals, DPs tend to be ordered by importance
- If work is missing its goal WLM may decide to adjust its DP equal or above a higher importance period
- The problem occurs when this lower importance period starts to consume more CPU and causes the higher importance period to miss its goal
- WLM will recognize this condition and fix it ... but it can be slow to react

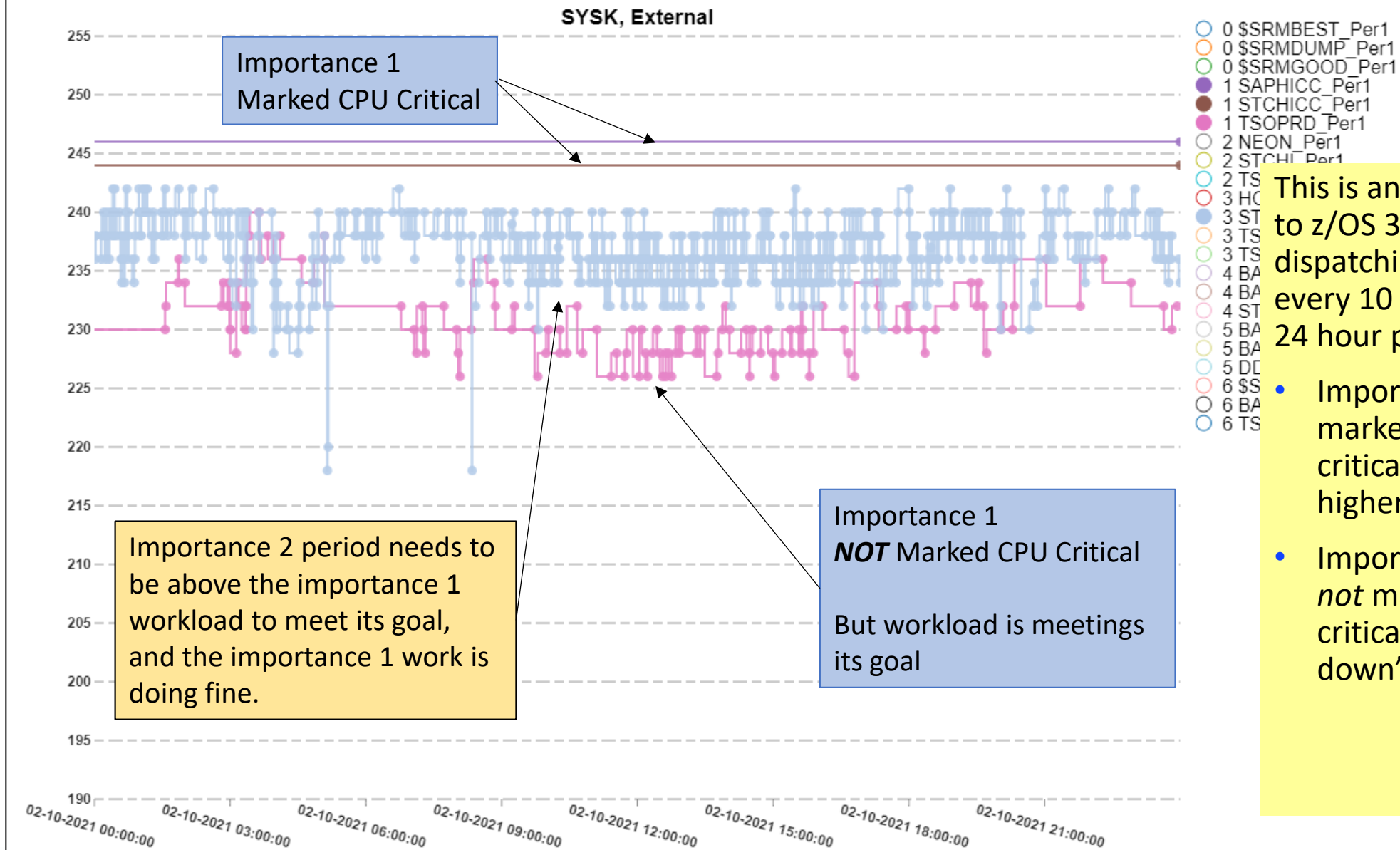
Note: To make the point, just a few priorities between DP 203 and DP247 are shown.

There is usually no need for using CPU Critical



- Over the last 26 years, the CPU dispatching algorithms have proven to be robust and responsive
- It is best to set both importance levels and goals properly
 - Let WLM figure out the optimal dispatching priorities to meet goals
 - Should it matter if work 'drifts down' in CPU DP if WLM adjusts it up when needed?
 - Less important on today's higher n-way and MIPS LPARs and CECs
 - Blocking less likely
- Occasionally, maybe use of CPU critical to provide 'peace of mind' for the management of certain workloads
 - Example: Financial institution that wants to make sure DB2 always has the highest CPU dispatching priority before the stock market open
- Regardless, set importance levels and goals correctly, and let WLM do the rest

WLM SMF 99.6 - CPU Dispatching Priority



This is an example (prior to z/OS 3.1) of CPU dispatching priorities every 10 seconds over a 24 hour period of time.

- Importance 1 periods marked as CPU critical always have a higher CPU DP
- Importance 1 period *not* marked as CPU critical can 'drift down'

Use CPU Critical for Predictability

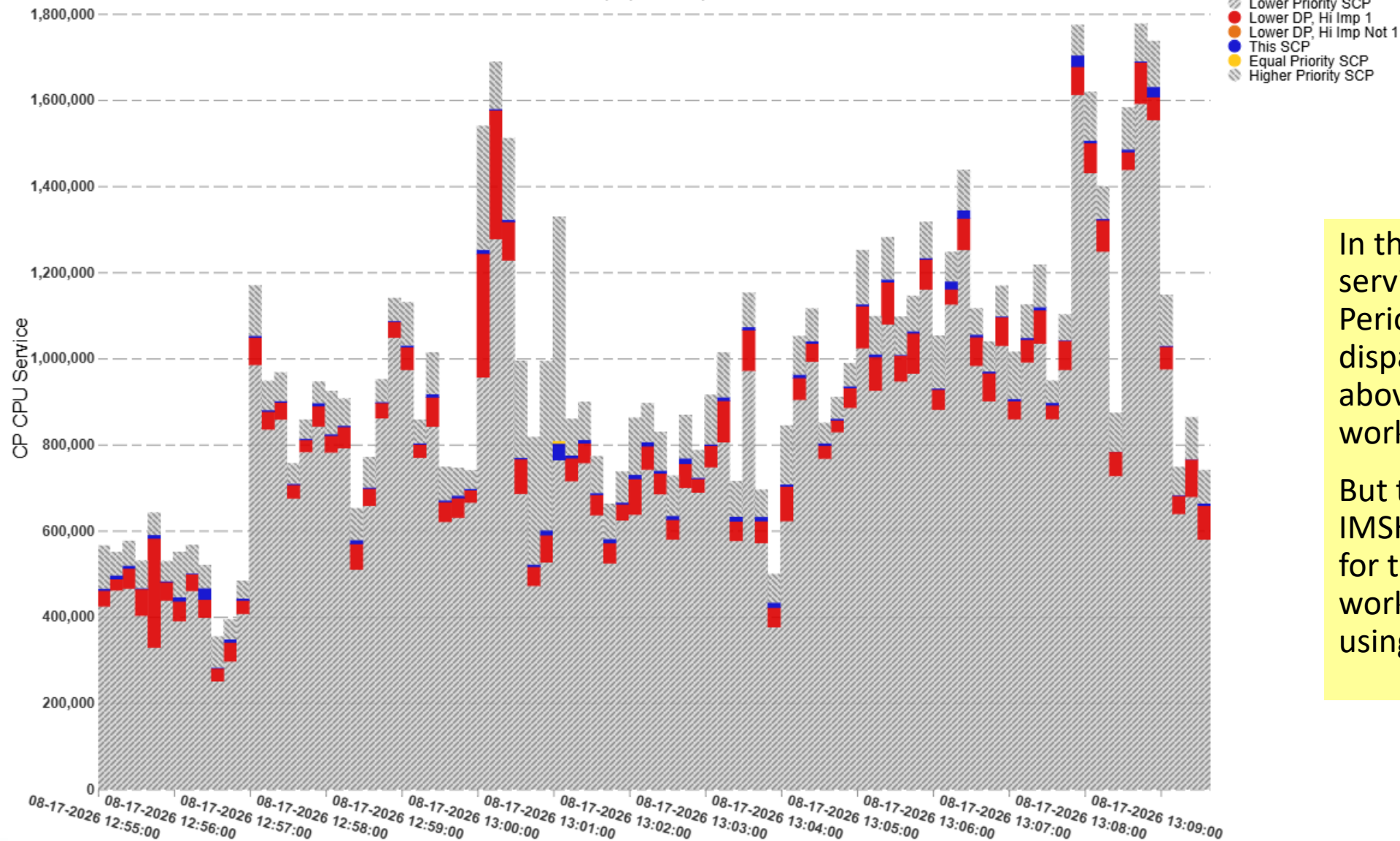


- When predictable access to the CPU is paramount, why not take advantage of the WLM CPU critical control?
- Many customers have high latent demand or are running lower n-way LPARs
 - Why not just take the loved ones, help ensure a high CPU DP. Why always worry if WLM is doing what is expected?
- CPU critical control is also great since requires less tuning of goals
 - These are my loved one, put up high, and let everything else be managed below them
 - In fact, in theory, if everything is CPU critical, then work will run in CPU dispatching 'bands' based on WLM importance level
- Just remember, usage of CPU critical demands an understanding of the workload
 - Does the period marked CPU critical have a predictable usage of CPU?
 - Is it possible for a large consuming workload to 'block' everything below it?

CP CPU Service Accumulated Above / Below SCP

From SMF 99.6

SYSJ, 2, IMSHI, Per1



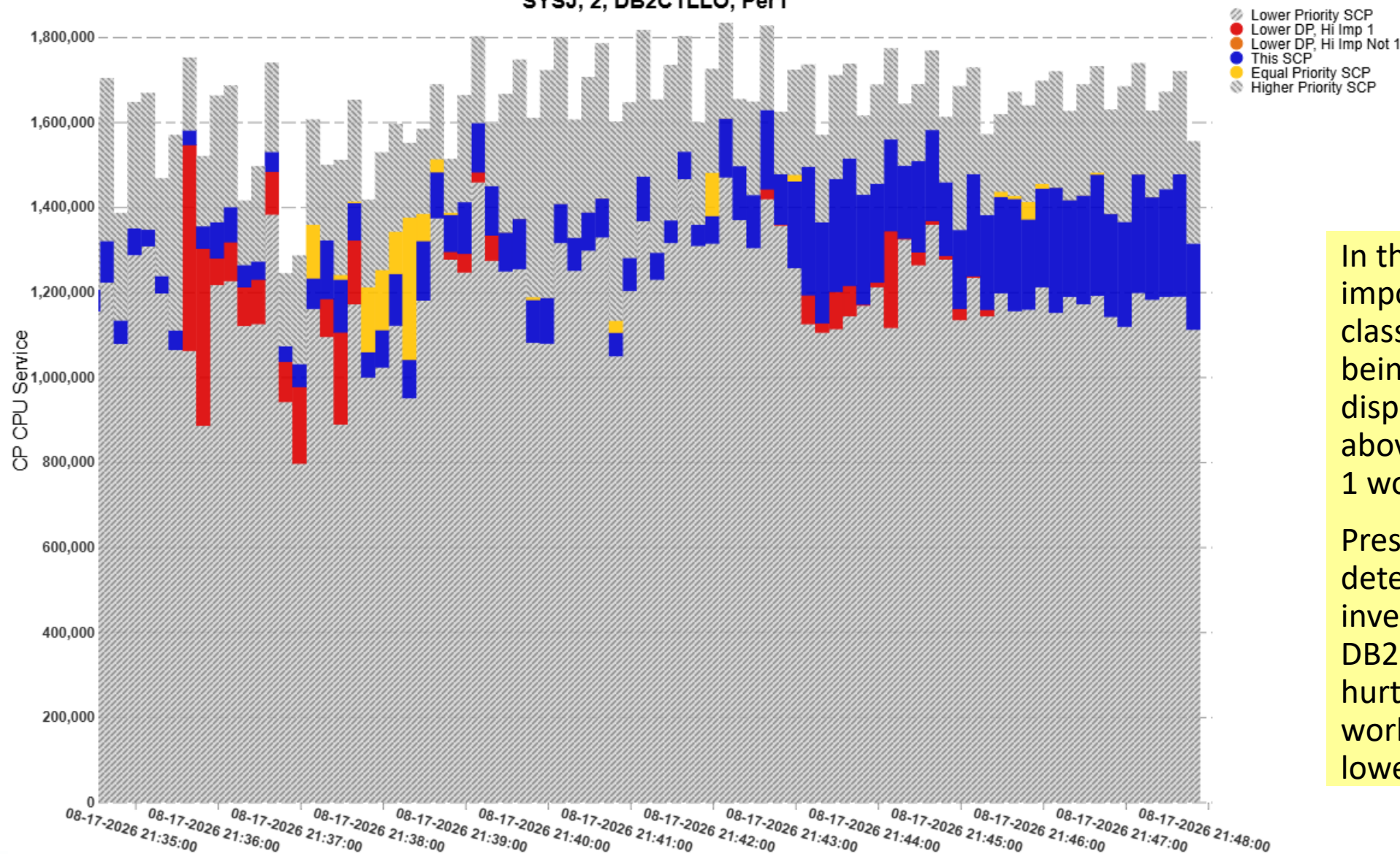
In this case Importance 2 service class IMSHI Period 1 is running at a dispatching priority above importance 1 workload.

But this is likely good for IMSHI and immaterial for the importance 1 work because IMSHI is using so little CPU.

CP CPU Service Accumulated Above / Below SCP

From SMF 99.6

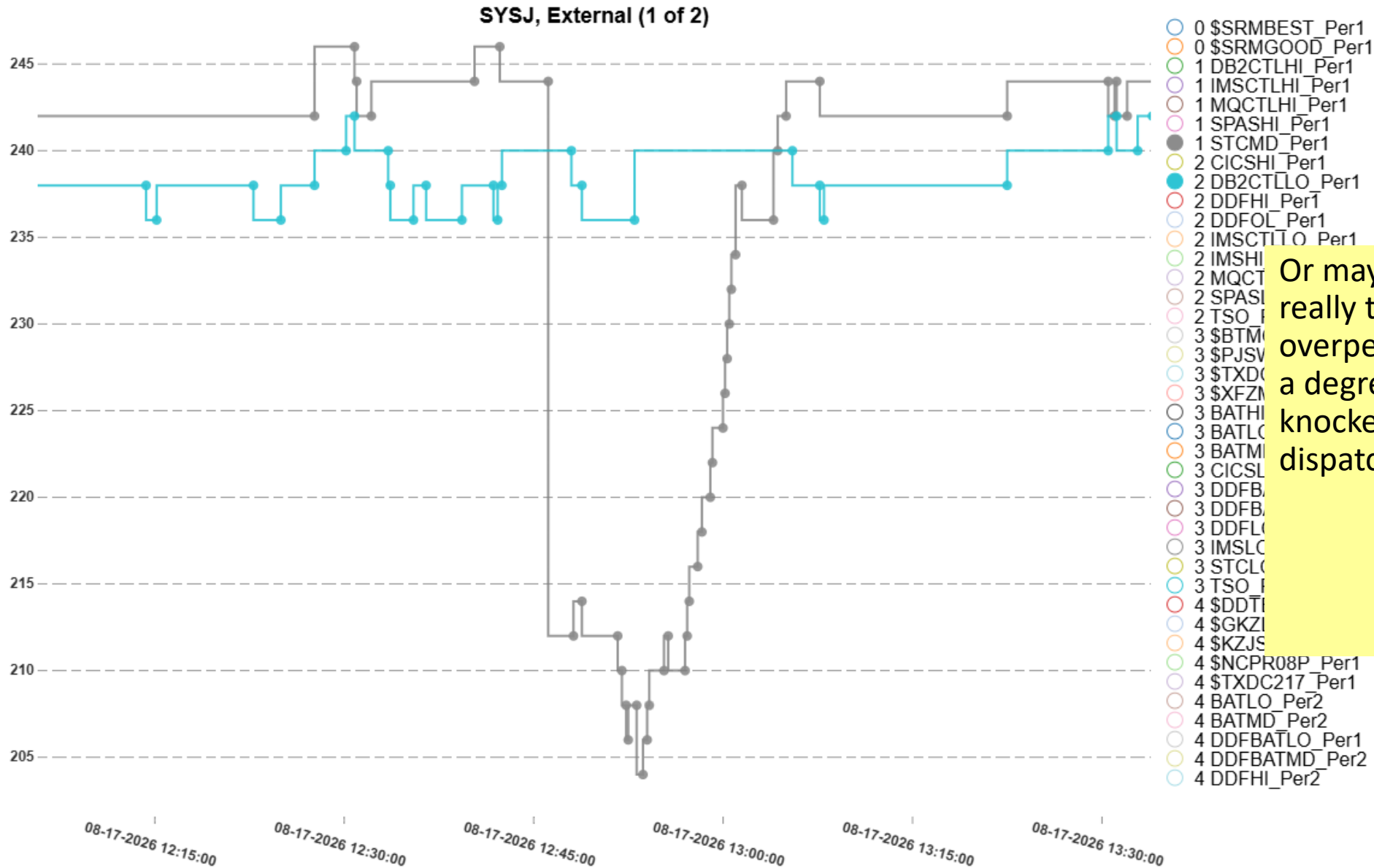
SYSJ, 2, DB2CTLLO, Per1



In this example, importance 2 service class DB2CTLLO (blue) is being run at a dispatching priority above some importance 1 workload.

Presumably WLM determined that this inversion helped DB2CTLLO without hurting the importance 1 workload that was at a lower priority.

WLM SMF 99.6 - CPU Dispatching Priority



Or maybe the issue was really that STCMD was overperforming to such a degree that WLM knocked down its dispatching priority.

Summary: what we really think



- Goals that are impossibly difficult or way too easy get limited management by WLM
 - For too easy: this work is at risk of degrading to the goal during times of stress. Is that ok?
- Response time goals can be either percentiles or averages
- Reserving free memory for infrequent use seems a missed opportunity
- Use memory to avoid I/O before investing in zHyperLink
- Logical measurements have value, but physical may be easier to interpret
- Reporting GCPs and zIIPs separately is almost always the right answer
- Use CPU critical control carefully for very few, and select, workloads with predictable CPU usage
 - For z/OS 3.1 determine whether it makes sense to let importance 1 run as CPU critical
- We don't understand IBM naming: Db2, HiperDispatch, zHyperLink, z/OS



Questions?

Thanks for attending!

