Controversial z/OS Performance Topics

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

Creators of Pivotor®

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

Voice: 813-435-2297
Mobile: 941-685-6789

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

Abstract

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.

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 your all your WLM goals?

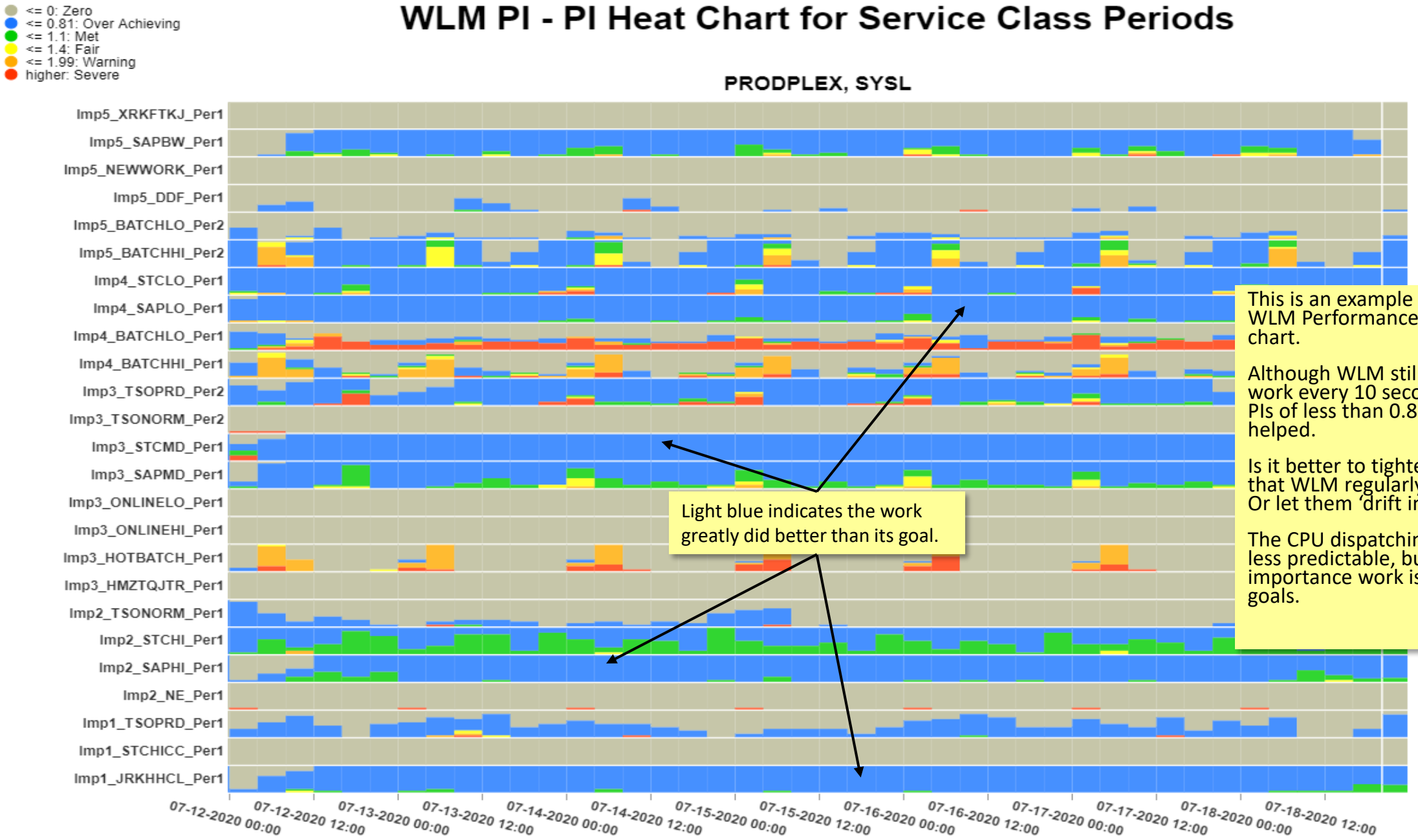
- Why it matters:
  - Is it worth the exercise to fine tune a goal if an installation's transactions are meeting their business objective?

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 WLM goals based on their business importance
  - 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?

WLM PI - PI Heat Chart for Service Class Periods



Light blue indicates the work greatly did better than its goal.

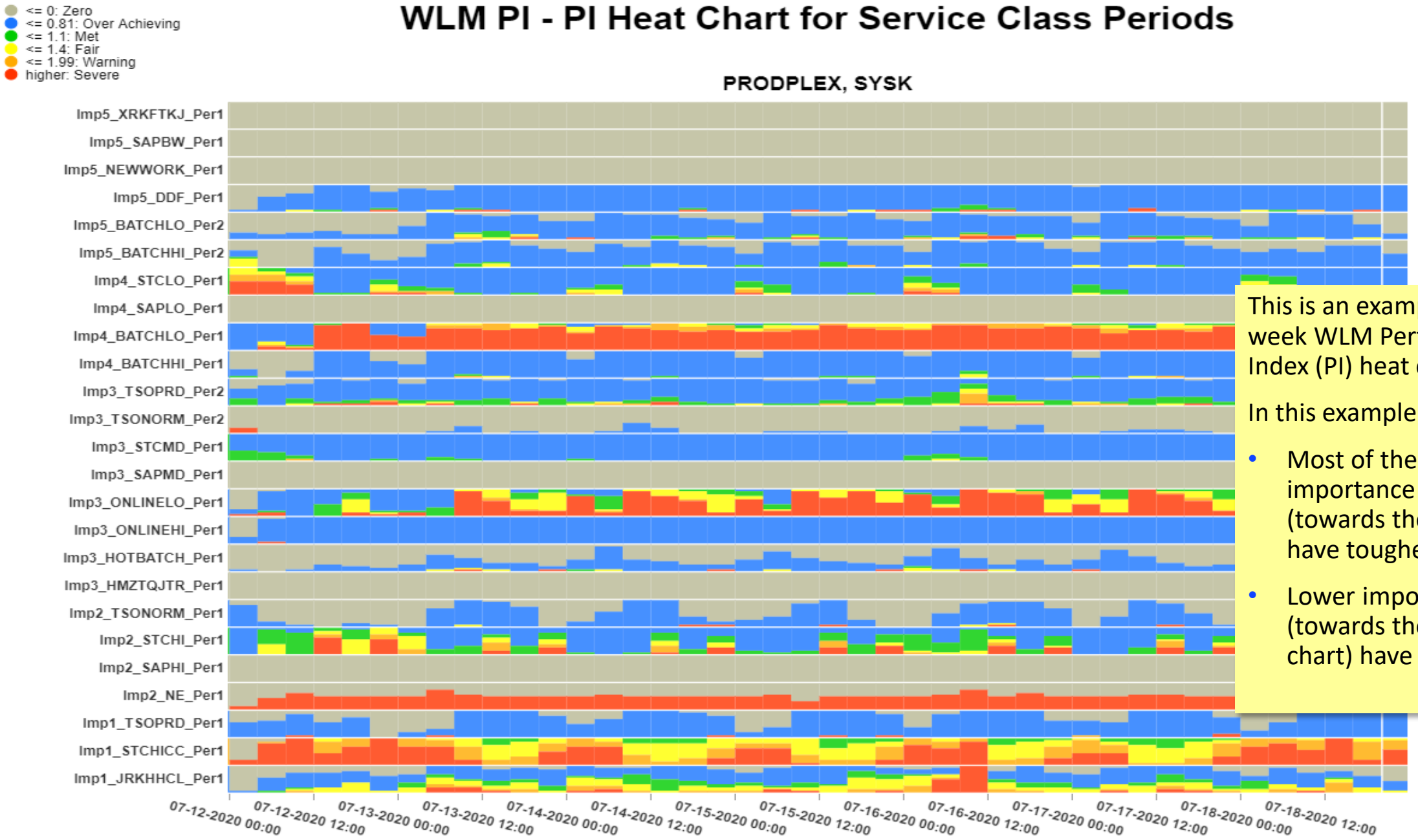
This is an example of a one-week WLM Performance Index (PI) heat chart.

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

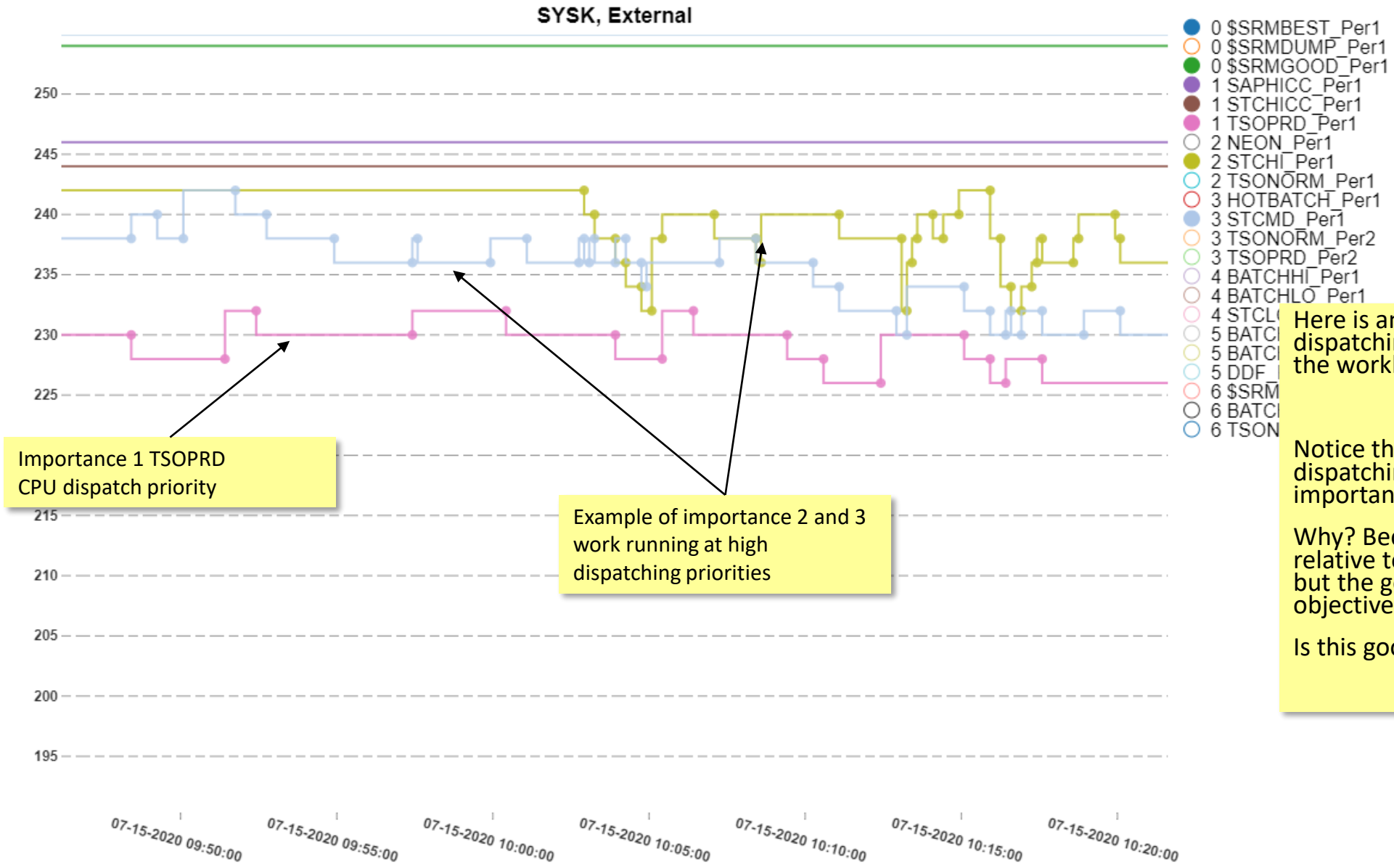


This is an example of a one-week WLM Performance Index (PI) heat chart.

In this example we see:

- Most of the higher importance workloads (towards the bottom) have tougher goals
- Lower importance goals (towards the top of the chart) have easier goals

WLM SMF 99.6 - CPU Dispatching Priority



Importance 1 TSOPRD CPU dispatch priority

Example of importance 2 and 3 work running at high dispatching priorities

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?

Does anybody need <15ms response time?

- Why it matters:
  - As processor environment become constrained it may be better to let very short response time transactions suffer if the end-user is less likely to notice the impact.

Point / Counter-Point

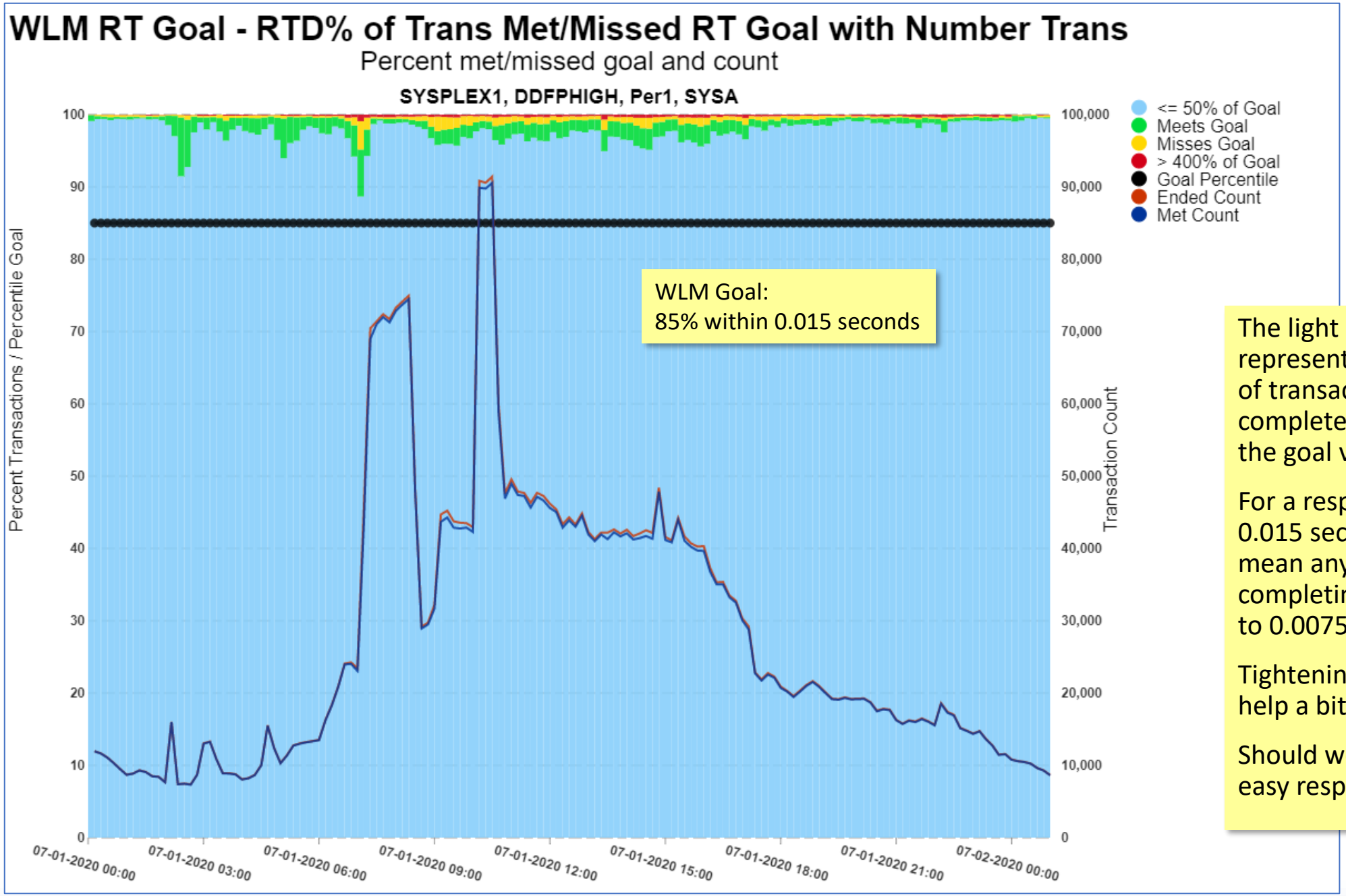
- Point
  - Use response time goals of < 15ms to encourage WLM manage short work
  - Some user interactions consists of multiple WLM transactions
  - Remember, goals that are too easy can be stolen from and given low CPU dispatching priorities

- Counter-Point
  - 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?

Understanding WLM's Response Time Distributions

- The value of each bucket is based on percentage of goal value
  - The below example is a distribution for a 0.015 second response time goal
  - Notice the first bucket contains nearly all the transactions and
  - On today's high-speed processors running lighter transactions, 0.015 second goal may be too 'easy' of a goal for some workloads

Bucket	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Width	<=50%	60%	70%	80%	90%	100%	110%	120%	130%	140%	150%	200%	400%	>400%
Value	<= .0075sec	0.009 sec	0.0105 sec	0.012 sec	0.0135 sec	0.015 sec	0.099 sec	0.108 sec	0.117 sec	0.126 sec	0.135 sec	0.180 sec	0.360 sec	>0.360 sec
Trans Count	50000	50	25	10	4	0	0	0	25	00	25	0	0	1

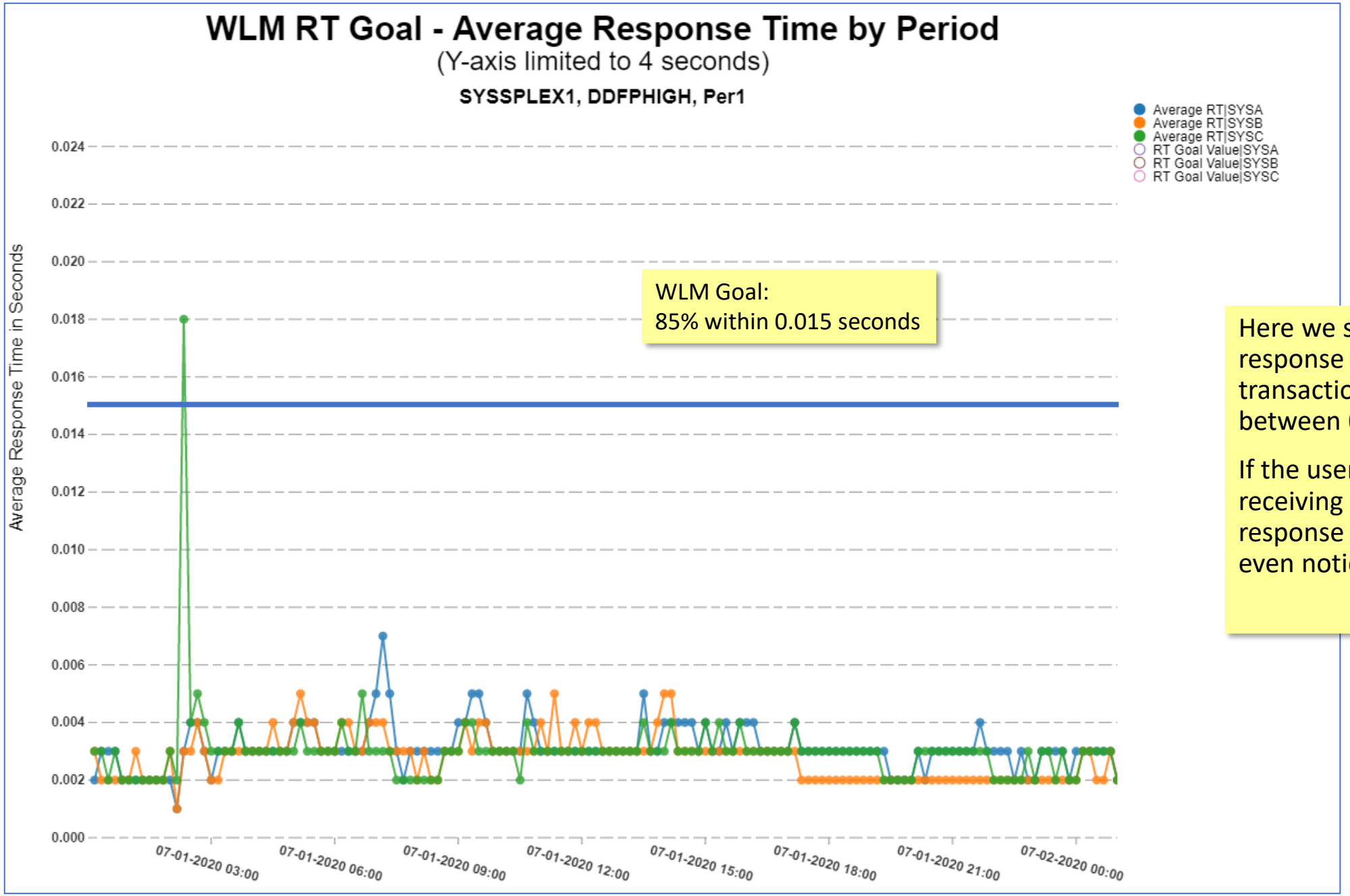


The light blue in this chart represents the percentage of transactions that completed in less than half the goal value.

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?



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?

- While transactions can achieve single-digit millisecond response time, should such goals be set
  - 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 below 15ms, 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

Accounting for Parked Time when calculating LPAR % Busy

- Why it matters:
  - Impacts the fundamental way we analyze and evaluate the logical processor constraints and utilizations of an LPAR

Point / Counter-Point

- Point
  - LPAR % Busy is a measure of utilization of the configured logical processor capacity
  - Will match the profile of MSU usage
  - This is the way LPAR % Busy has been calculated and used since PR/SM was first introduced

- Counter-Point
  - LPAR % Busy should only include the unparked processor capacity
    - That is, how busy is the unparked capacity
  - Allows for a more accurate assessment of latent demand

Traditional LPAR % BUSY Formula

- Traditional formula currently used by RMF and CMF and Pivotor

  For a non-dedicated partition when Wait Completion is NO (which is 99.8% of all z/OS partitions)

$$\text{LPAR BUSY TIME \%} = \frac{\text{Partition Dispatch Time}}{\text{Online Time}} * 100$$

  The partition dispatch time is the sum elapsed time that PR/SM dispatched the logical cores during the interval.

- How is LPAR % Busy used?
  - It is the primary measurement used since the introduction of PR/SM to evaluate items such as
    - As a utilization measure of the logical processors to evaluate the logical processor constraints of an LPAR
    - As a base measure when calculating Capture Ratios
    - As a contextual measure when evaluating Latent Demand
    - and more…
  - It is probably the most widely used measurement on most monitors
    - Basically, it is what we use to see how busy an individual system is

RMF CPU Activity Report Example

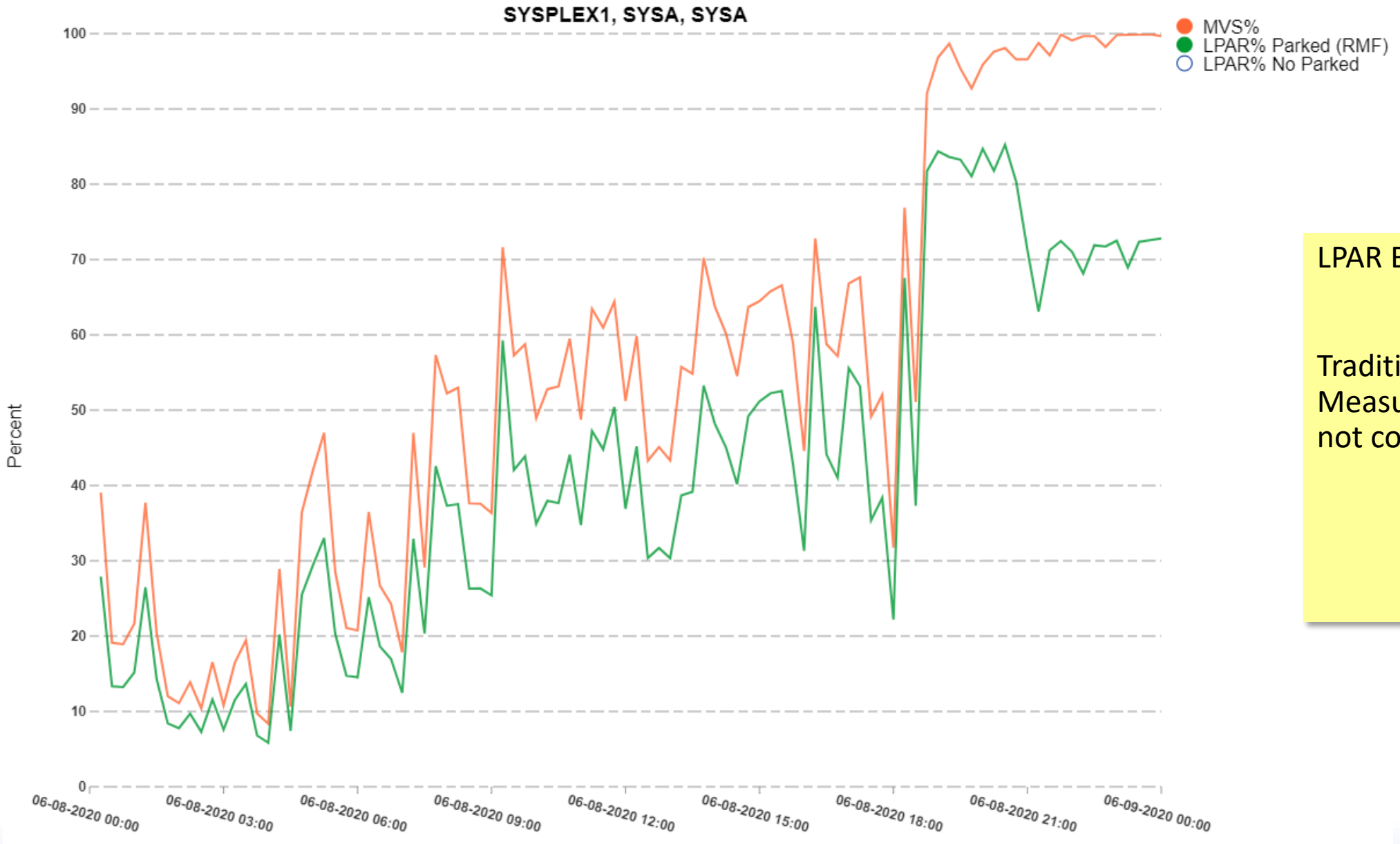
- The following is an example of an RMF report
  - Note the (in this case) slight difference in LPAR BUSY and MVS BUSY

```
                                                C P U  A C T I V I T Y

            z/OS V2R3                    SYSTEM ID MVSD           DATE 11/30/2020          INTERVAL 09.25.950
                                         RPT VERSION V2R3 RMF     TIME 15.00.00            CYCLE 0.500 SECONDS

-CPU         2964    CPC CAPACITY    874         SEQUENCE CODE 00000000000F8D14
 MODEL       608     CHANGE REASON=NONE          HIPERDISPATCH=YES
 H/W MODEL   N63
0---CPU---    ---------------- TIME % ----------------  --- MT % ----   LOG PROC      --I/O INTERRUPTS--
 NUM   TYPE   ONLINE    LPAR BUSY    MVS BUSY   PARKED  PROD   UTIL     SHARE %       RATE     % VIA TPI
  0     CP    100.00    71.82        71.86        0.00  100.00  71.82   100.0  HIGH    3721      9.80
  1     CP    100.00    73.94        75.51        0.00  100.00  73.94    63.7  MED    94.78     44.83
  2     CP    100.00    67.19        68.68        0.00  100.00  67.19    63.7  MED    92.21     44.11
  3     CP    100.00    36.67        52.15       27.48  100.00  36.67     0.0  LOW     0.00      0.00
  4     CP    100.00     0.06        -----      100.00  100.00   0.06     0.0  LOW     0.00      0.00
  5     CP    100.00     0.06        -----      100.00  100.00   0.06     0.0  LOW     0.00      0.00
 TOTAL/AVERAGE           41.62        68.16              100.00  41.62   227.4          3908     11.46
```

MVS CP Busy%, LPAR CP Busy% (with / without Parked)



LPAR Busy

Traditional Measurement that does not consider parked time

A new formula that seems to be gaining traction

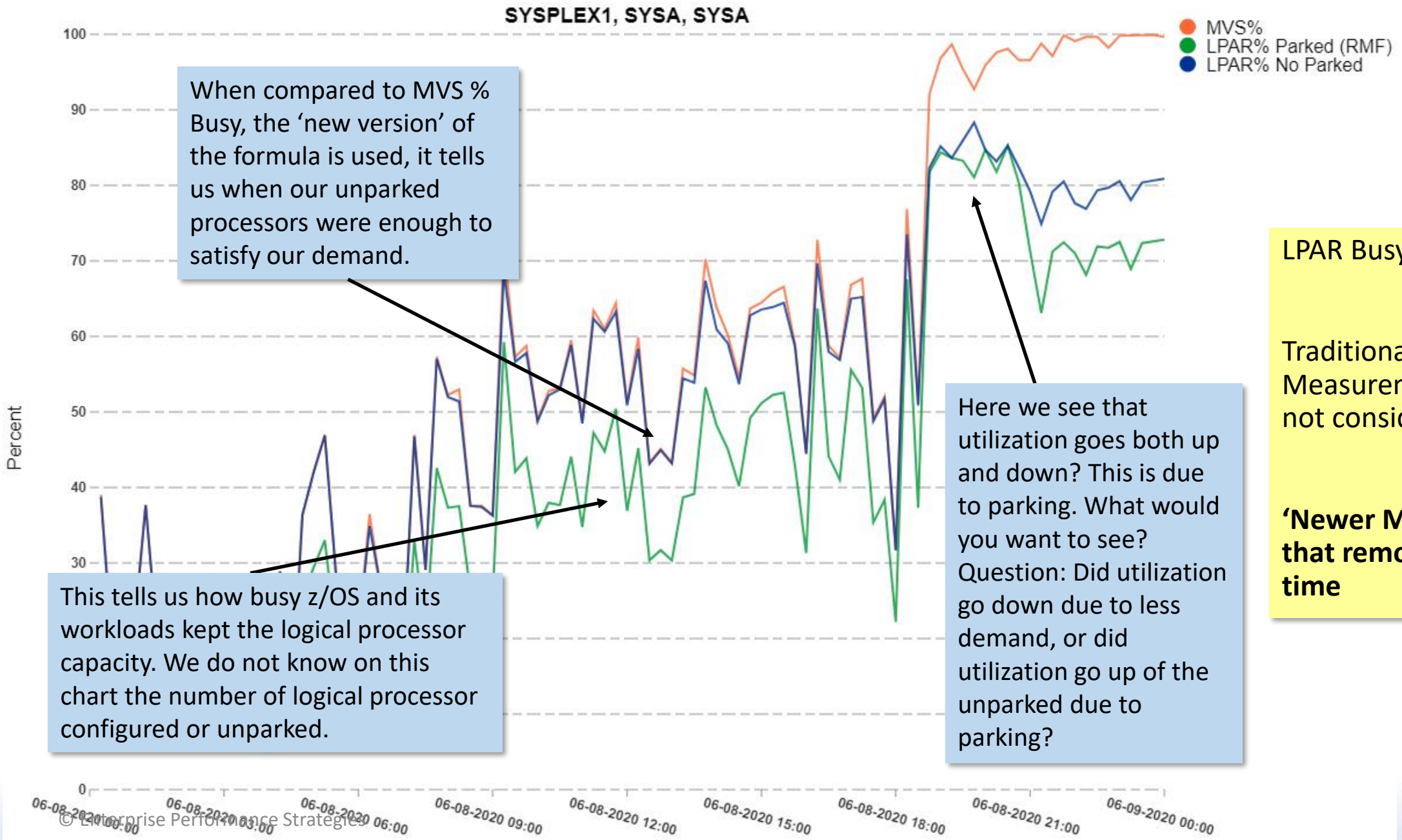
- Competing formula which results in a higher than expected LPAR BUSY

For a non-dedicated partition when Wait Completion is NO (which is 99.8% of all z/OS partitions)

$$\text{LPAR BUSY TIME \%} = \frac{\text{Partition Dispatch Time}}{\text{Online Time} - \text{Parked Time}} * 100$$

- Thus:
  - When all processors are unparked, the old and new values are the same
  - But when there are parked processors the formula becomes ‘how busy unparked logical processors are’ rather than ‘how busy the configured logical processors are’
  - Probably a better measure to understand current latent demand

MVS CP Busy%, LPAR CP Busy% (with / without Parked)



When compared to MVS % Busy, the 'new version' of the formula is used, it tells us when our unparked processors were enough to satisfy our demand.

This tells us how busy z/OS and its workloads kept the logical processor capacity. We do not know on this chart the number of logical processor configured or unparked.

Here we see that utilization goes both up and down? This is due to parking. What would you want to see? Question: Did utilization go down due to less demand, or did utilization go up of the unparked due to parking?

LPAR Busy

Traditional Measurement that does not consider parked time

'Newer Measurement' that removes parked time

Is conflating zIIP and GCP measurements a good practice?

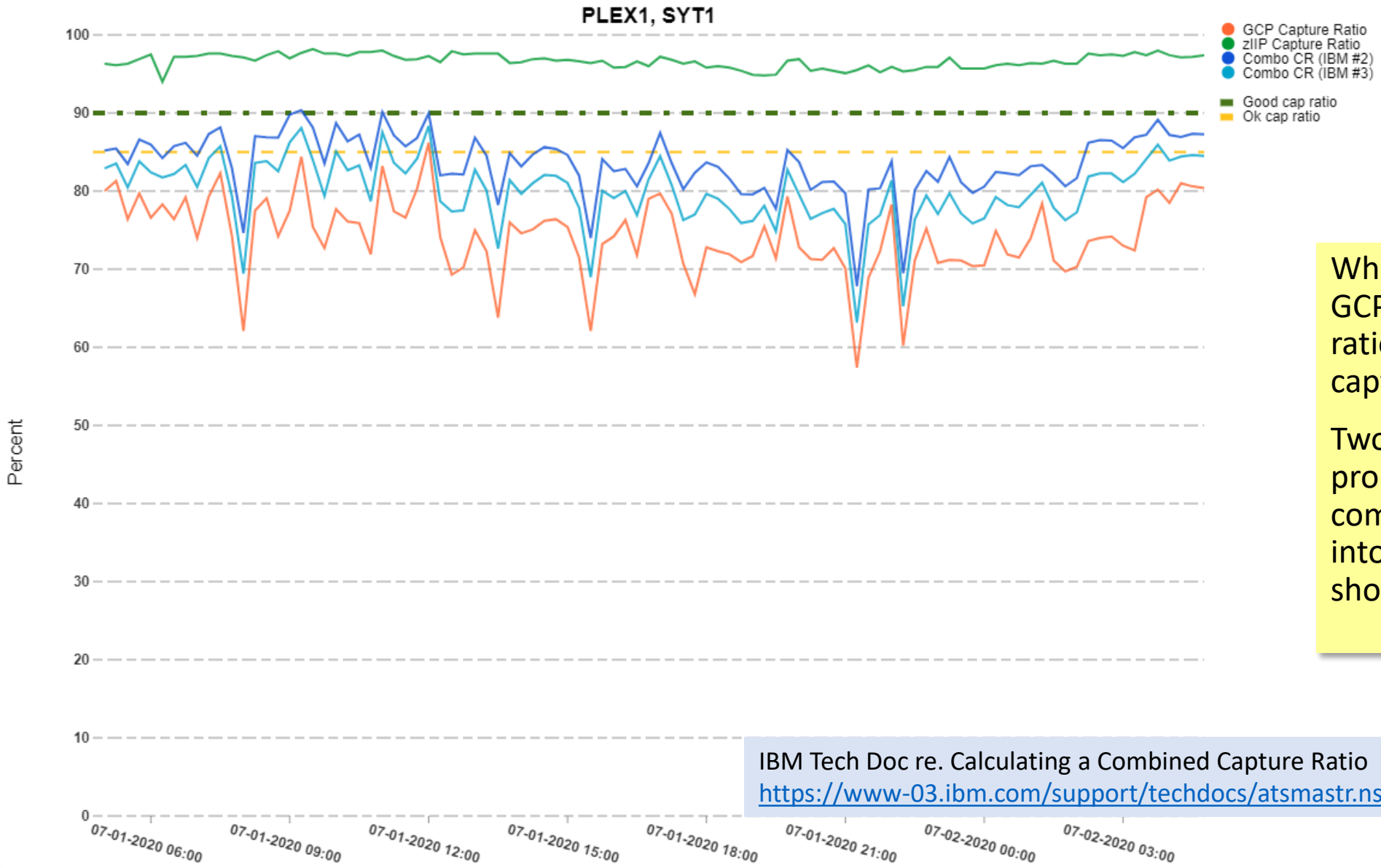
- Why it matters:
  - Conflating the zIIP and GCP measurements can misrepresent the overall efficiency delivered by each processor pool.

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

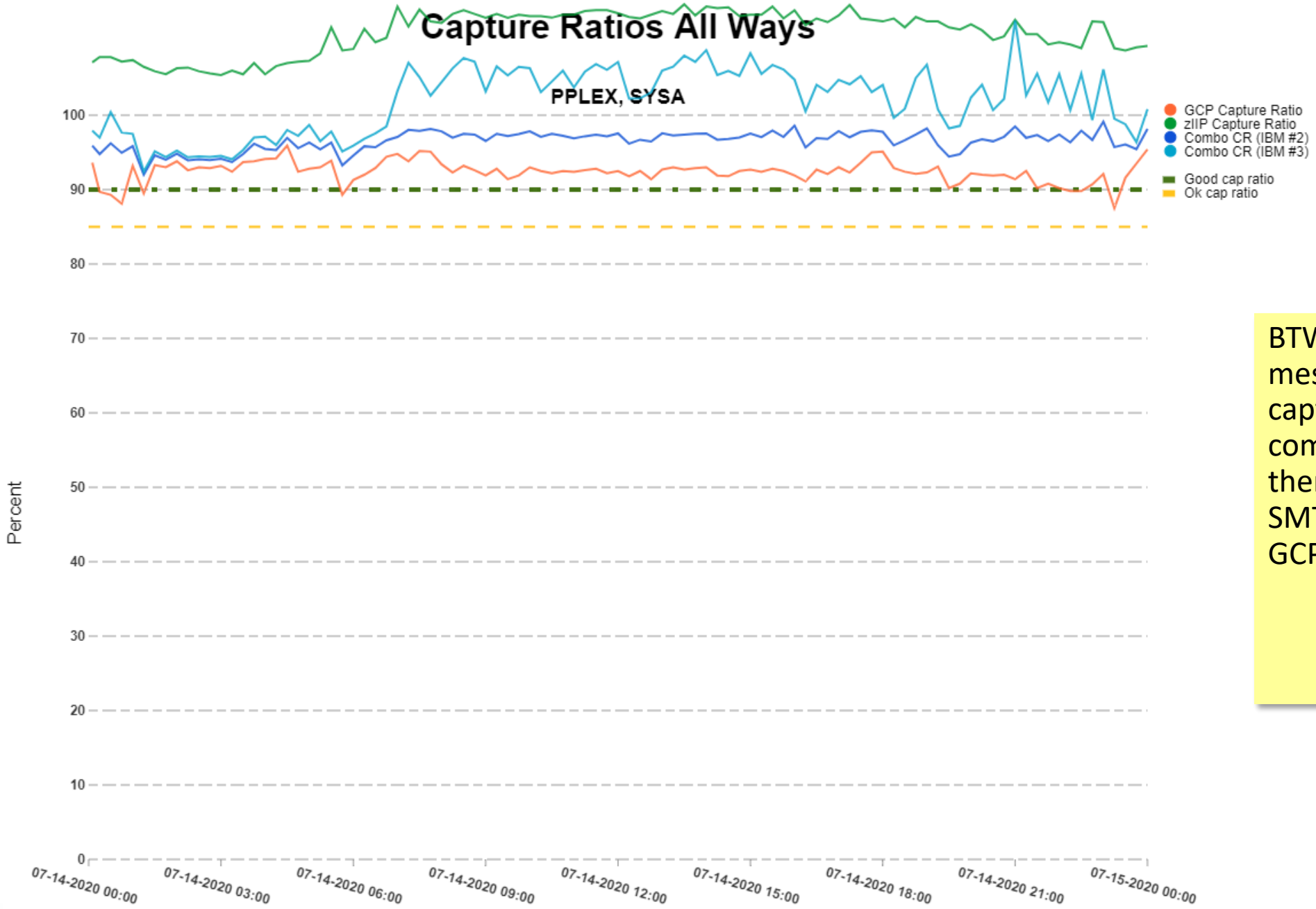
Capture Ratios All Ways



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.

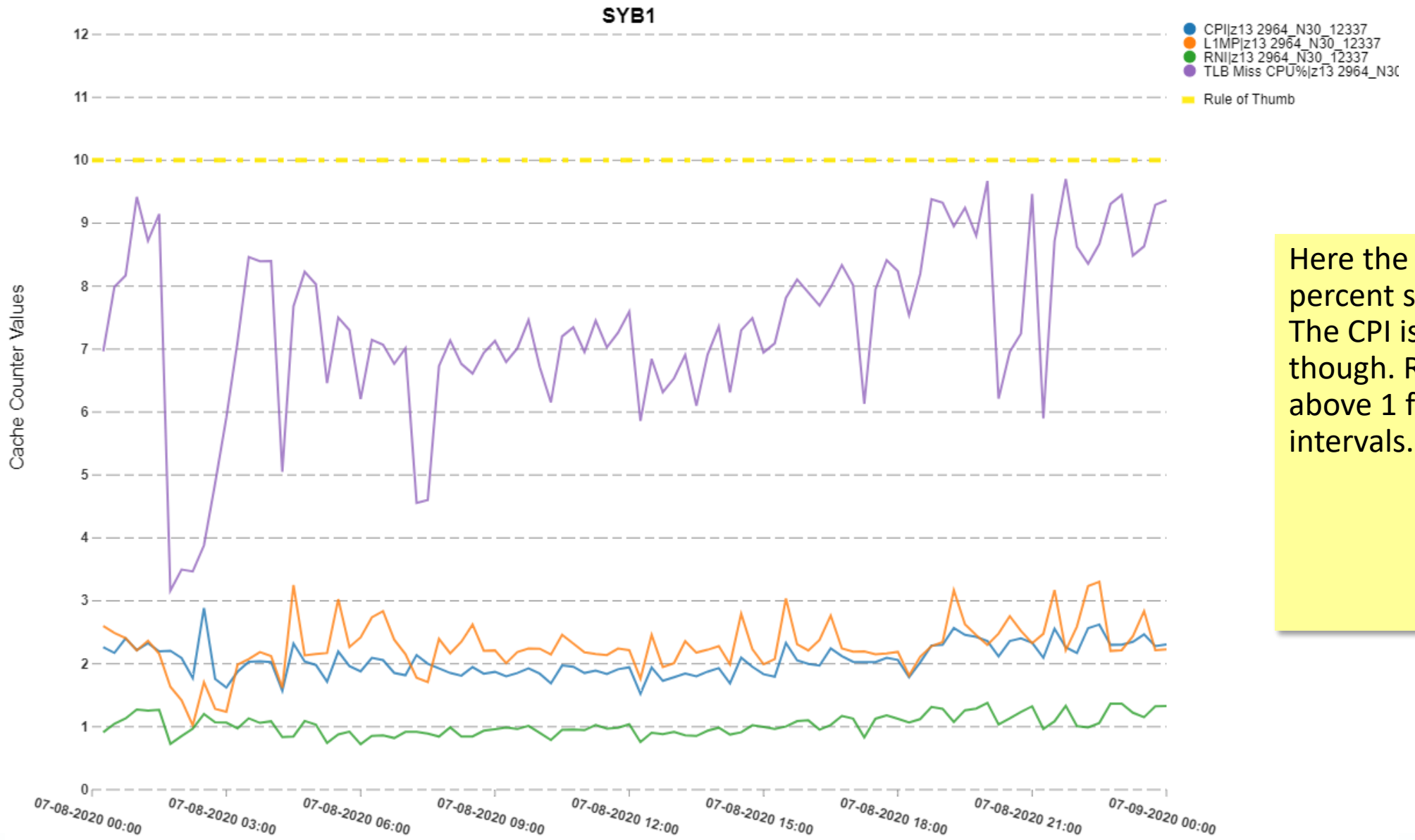
IBM Tech Doc re. Calculating a Combined Capture Ratio
https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102717



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

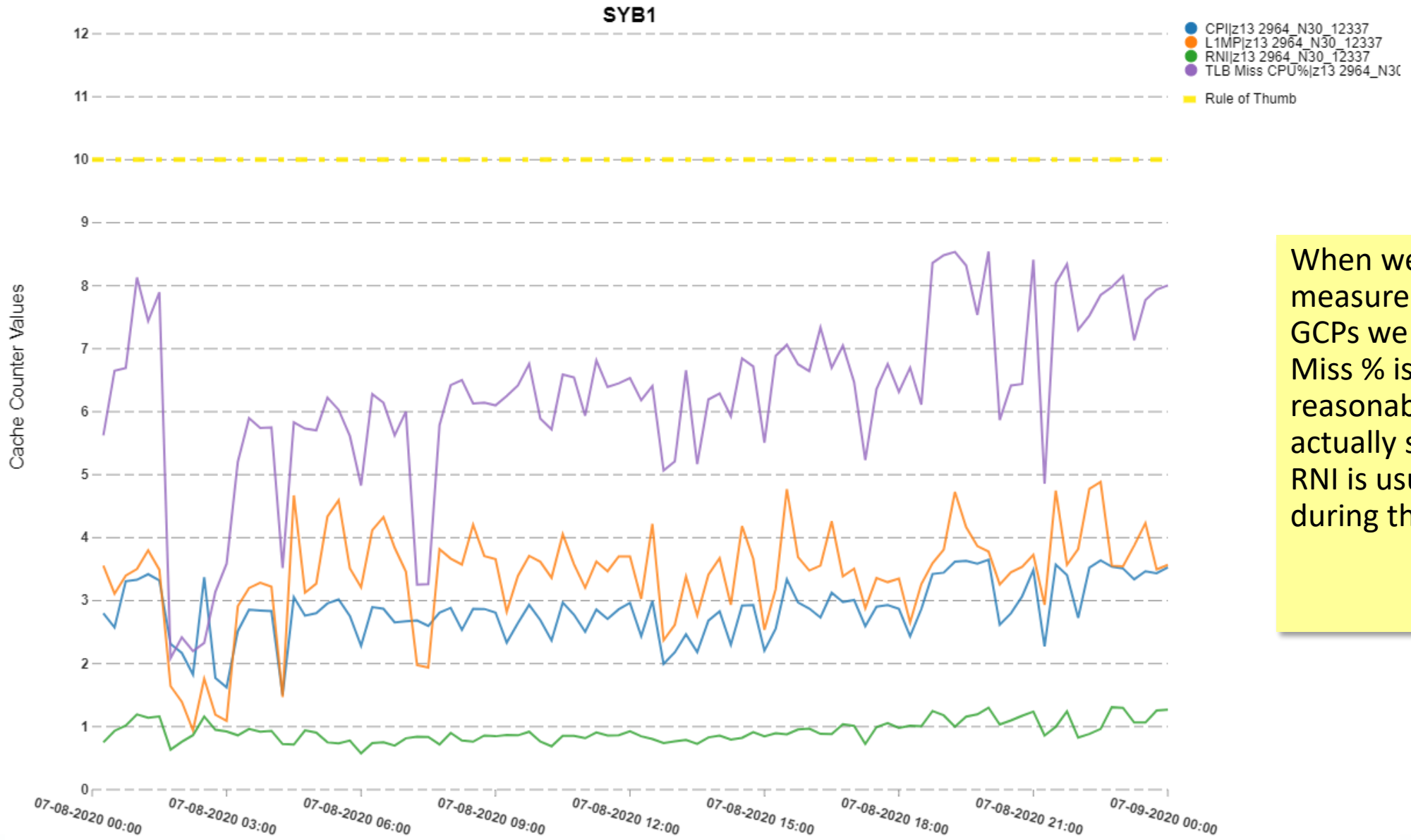
SMF 113



Here the TLB miss percent seems a bit high. The CPI is pretty good though. RNI is at or above 1 for many intervals.

Processor Caches - CP CPU Key Measurements

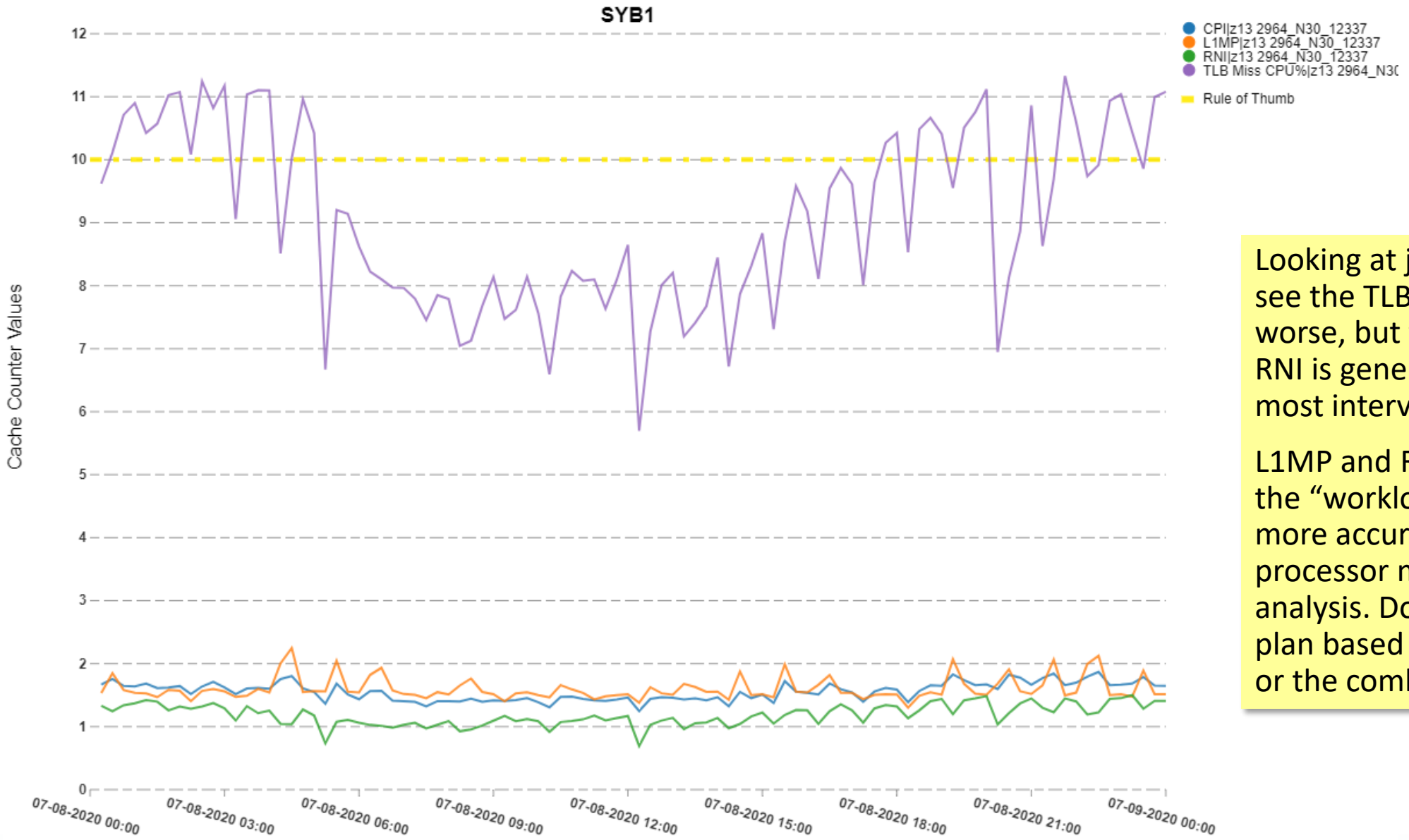
SMF 113



When we look at just the measurements for the GCPs we see the TLB Miss % is more reasonable, but CPI is actually slightly worse. RNI is usually below 1 during the day.

Processor Caches - zIIP CPU Key Measurements

SMF 113



Looking at just the zIIPs we see the TLB Miss % is worse, but the CPI is better. RNI is generally above 1 for most intervals.

L1MP and RNI determines the “workload hint” for more accurate zPCR processor migration analysis. Do you want to plan based on GCPs, zIIPs, or the combination?

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.

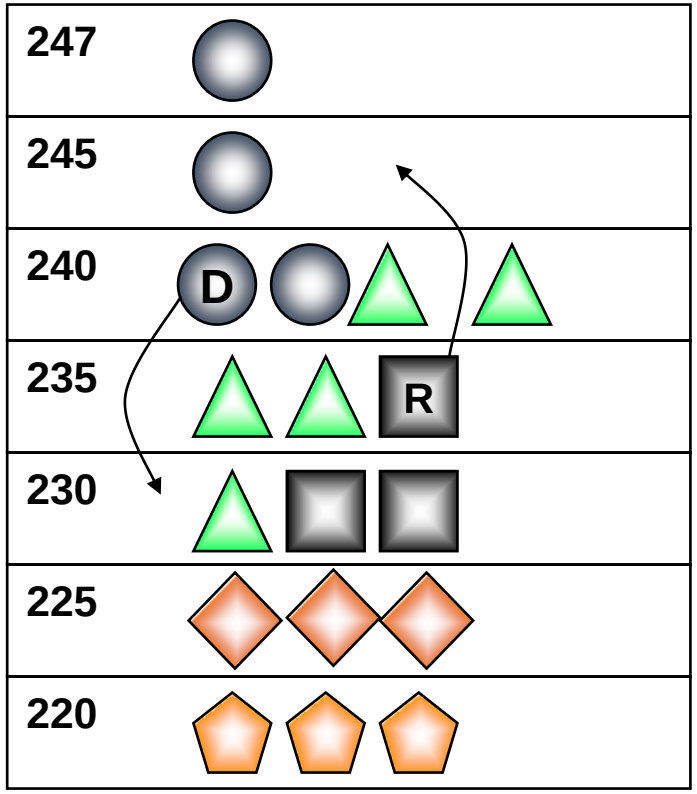
Point / Counter-Point

- 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

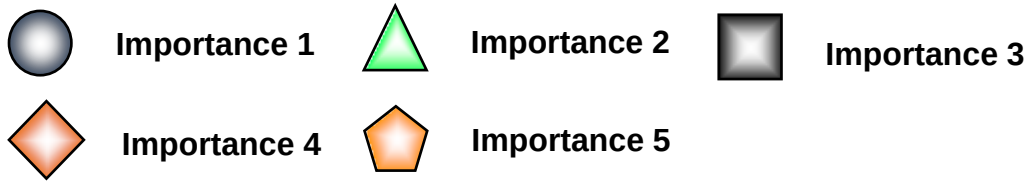
- Counter-Point
  - Many installations do not want to take the risk of a workload getting a less than predictable CPU dispatching priority
  - This is especially true on low n-way machines and LPARs

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



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



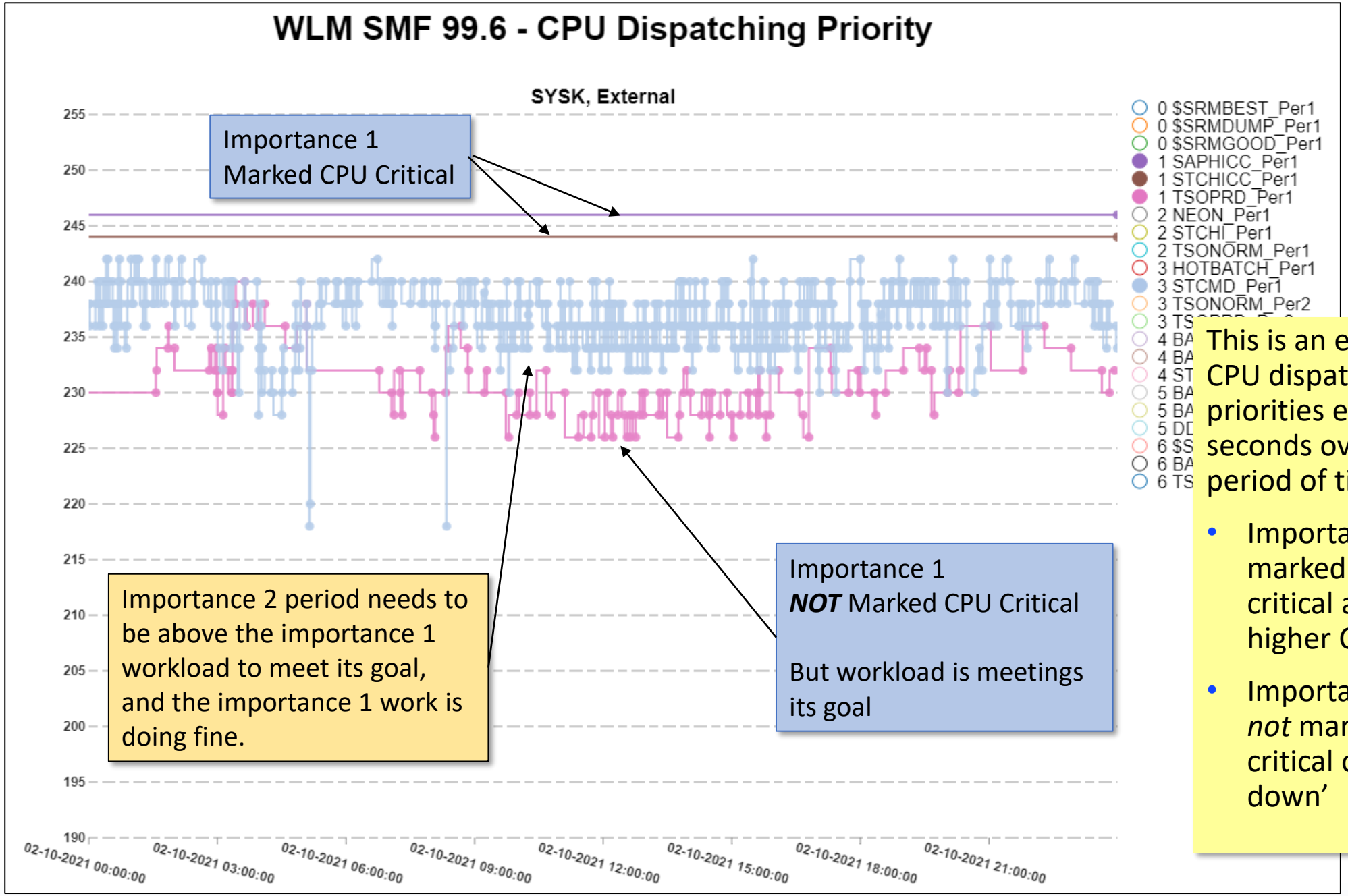
- With well set predictable goals, DPs tend to be ordered by importance
- If work is missing its goal WLM may decide to adjust its DP equal or above a higher importance period
- The problem occurs when this lower importance period starts to consume more CPU and causes the higher importance period to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

Overview - CPU Critical Control

- Objective
  - Ensure that worked marked as CPU critical always has a CPU dispatching priority above lower importance work
    - Option set at the Service Class level
  - WLM still manages the priority within its importance level and the importance level of any higher priority work
  - Example: Importance level 2 service class period marked as CPU critical
    - Will have CPU dispatching priorities above importance levels 3, 4, and 5.
    - Can still be equal to or above any importance level 1 work not marked as CPU critical
- It does limit WLM's management of the workloads

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
  - Especially true on todays 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



This is an example of CPU dispatching priorities every 10 seconds over a 24 hour period of time.

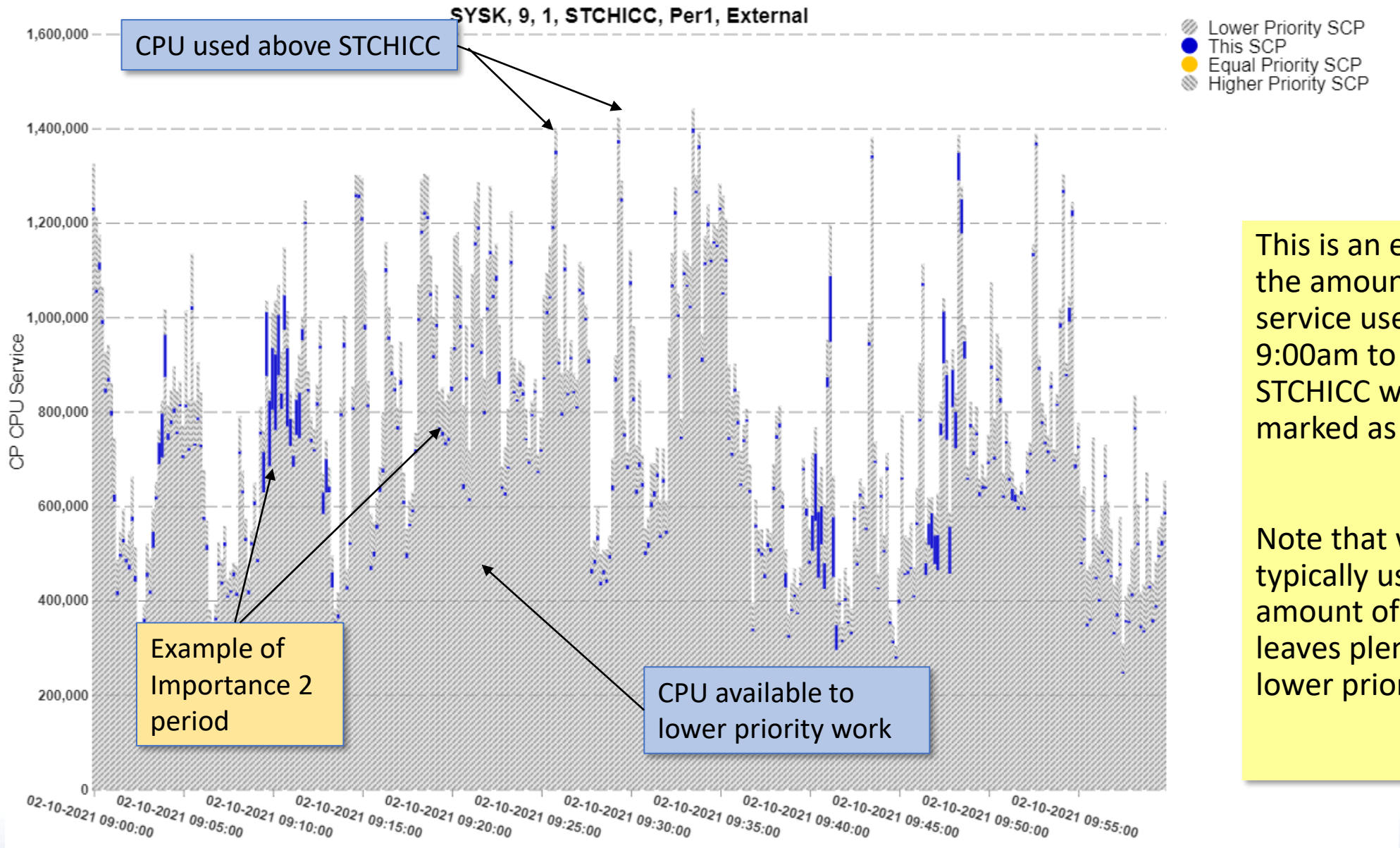
- Importance 1 periods marked as CPU critical always have a higher CPU DP
- Importance 1 period *not* marked as CPU critical can 'drift down'

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

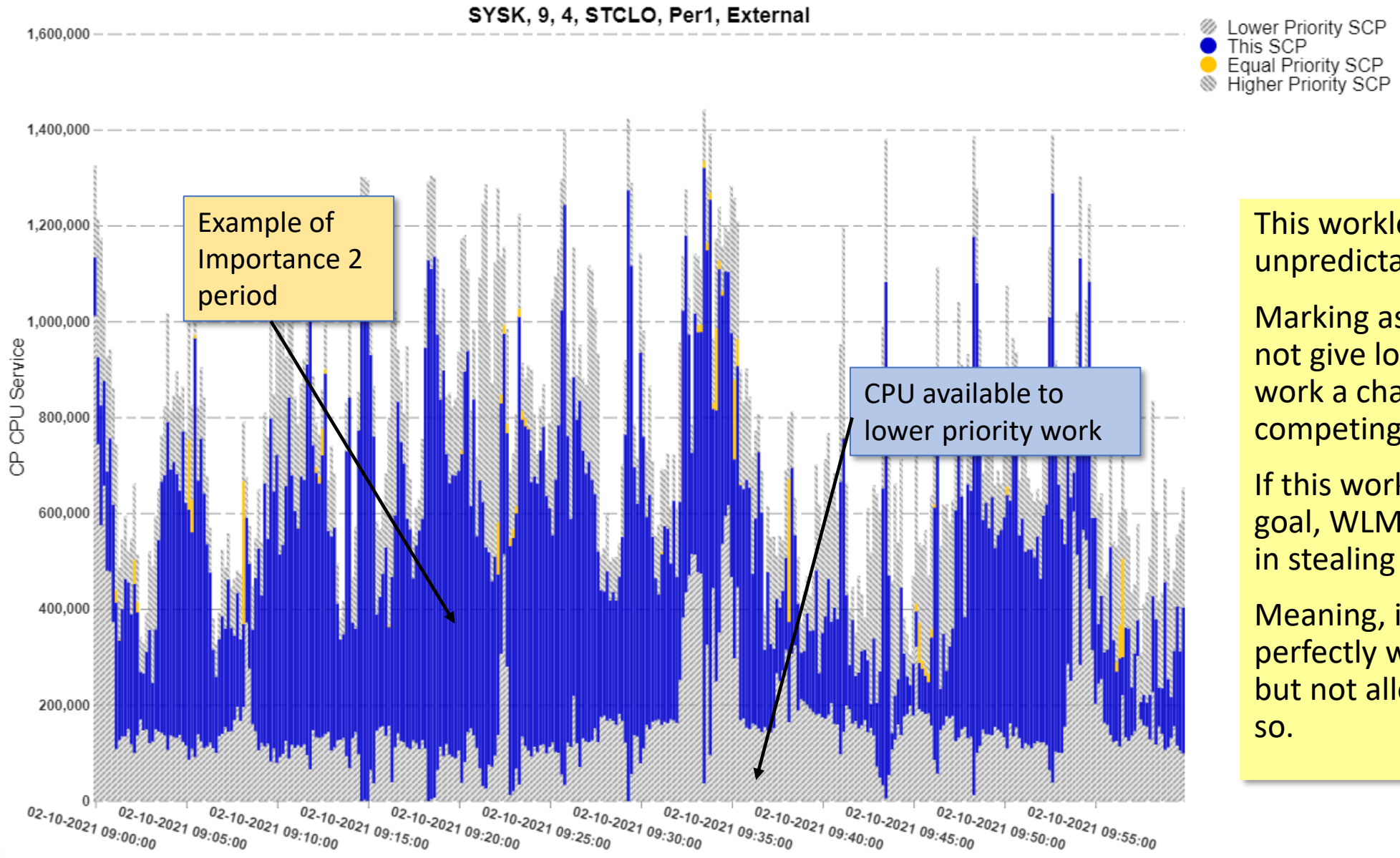


This is an example of the amount of CPU service used for the 9:00am to 9:59am for STCHICC which is marked as CPU critical.

Note that workload typically uses a small amount of service and leaves plenty of CPU for lower priority work.

CP CPU Service Accumulated Above / Below SCP

From SMF 99.6



This workload has an unpredictable usage of CPU.

Marking as CPU critical may not give lower importance work a chance of competing.

If this workload had an easy goal, WLM would be limited in stealing from it.

Meaning, it could do perfectly well at a lower DP, but not allowing WLM to do so.

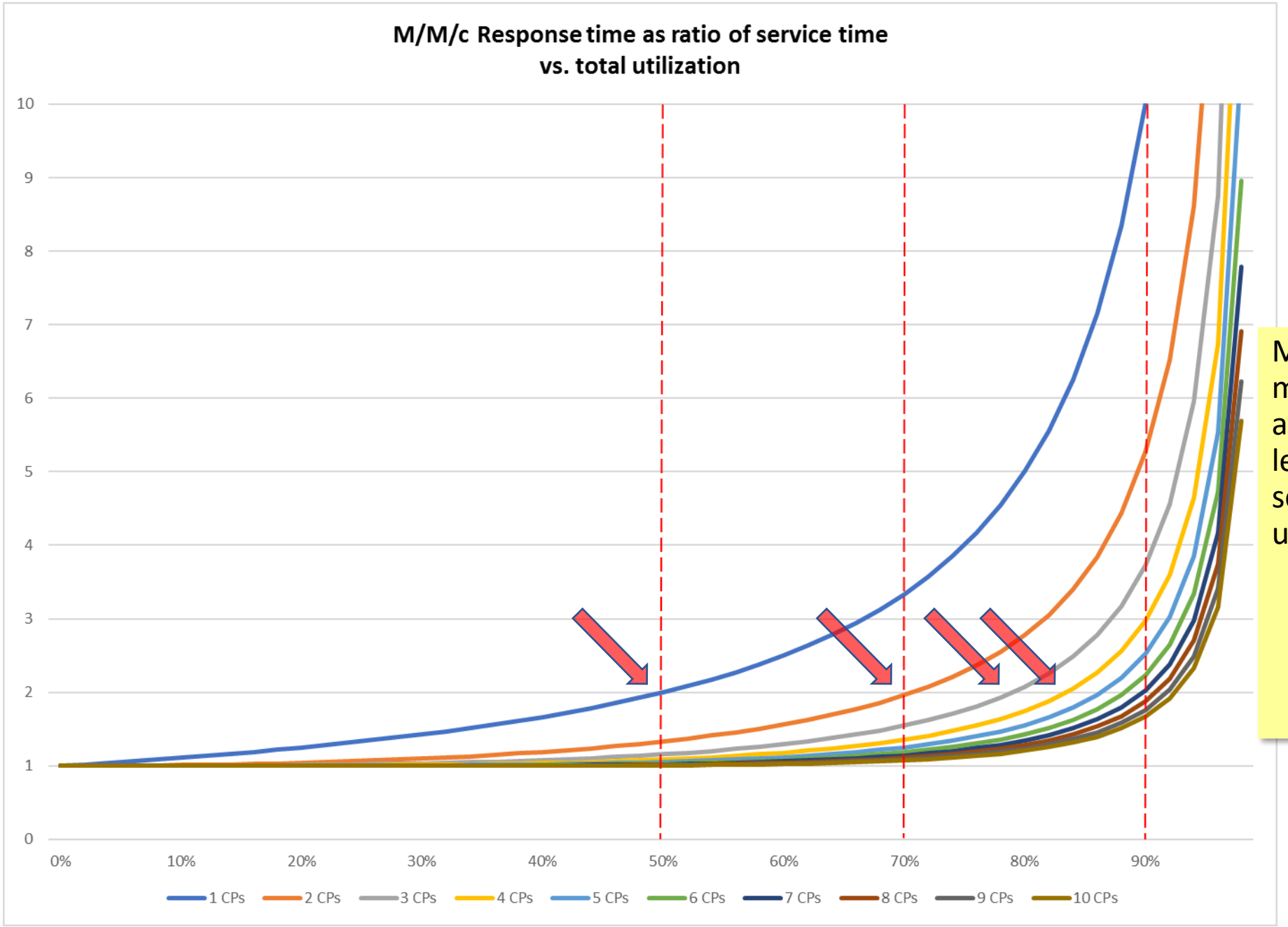
When is zIIP utilization too high?

- Why it matters:
  - How do you decide when to purchase more zIIPs?

Point / Counter-Point

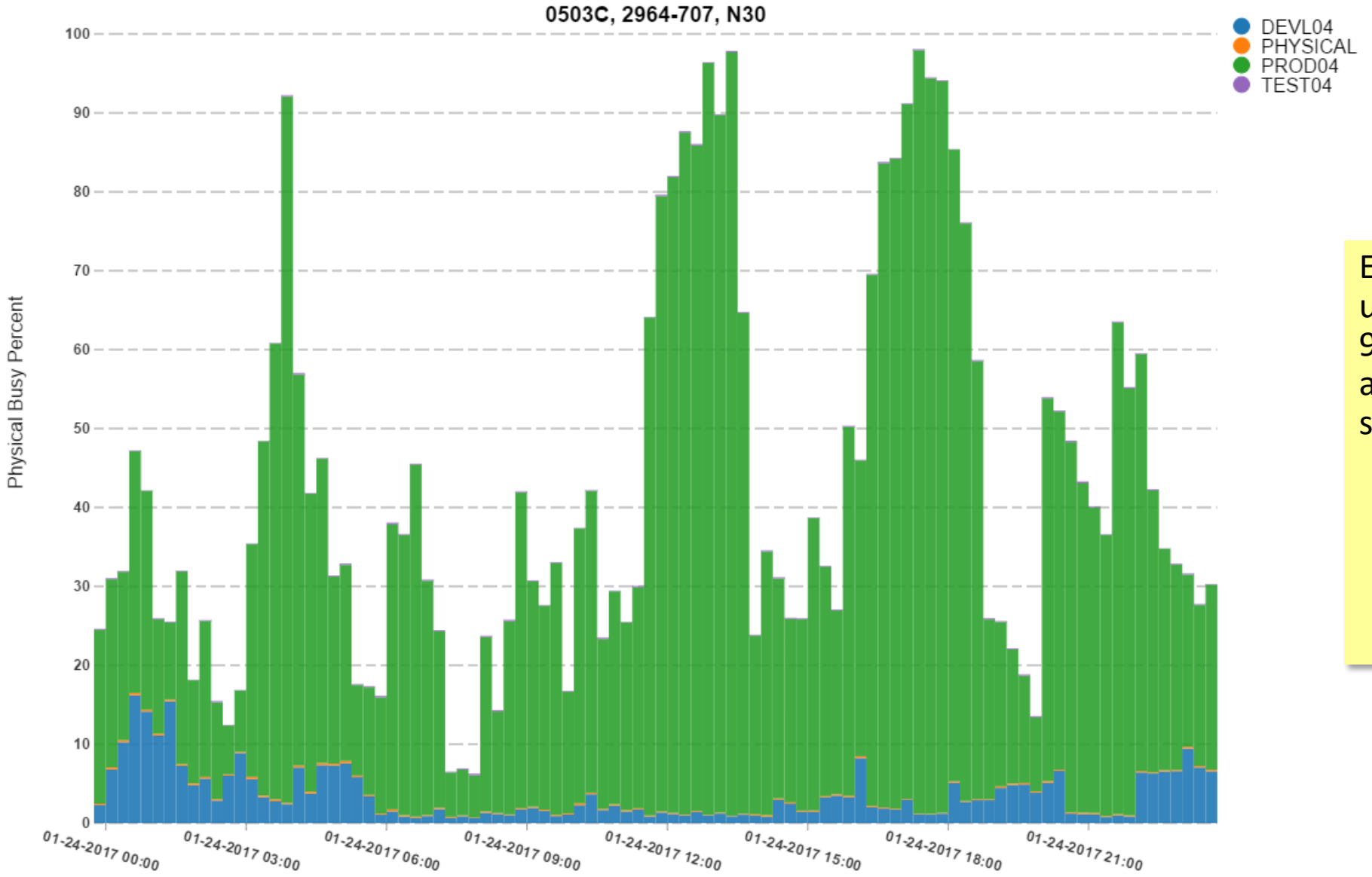
- Point
  - Don’t run your zIIPs too busy… keep at 70% or less
  - Regardless, it’s more “dangerous” to overload the zIIPs
  - zIIPs are cheap, buy more

- Counter-Point
  - We run GCPs at upwards of 100%, why not zIIPs?
  - Buying more zIIPs is great for performance but not always an option
  - The whole point of zIIPs is to keep work from running on the CPs
    - Does it matter how busy the zIIPs are if they’re saving GCP capacity?



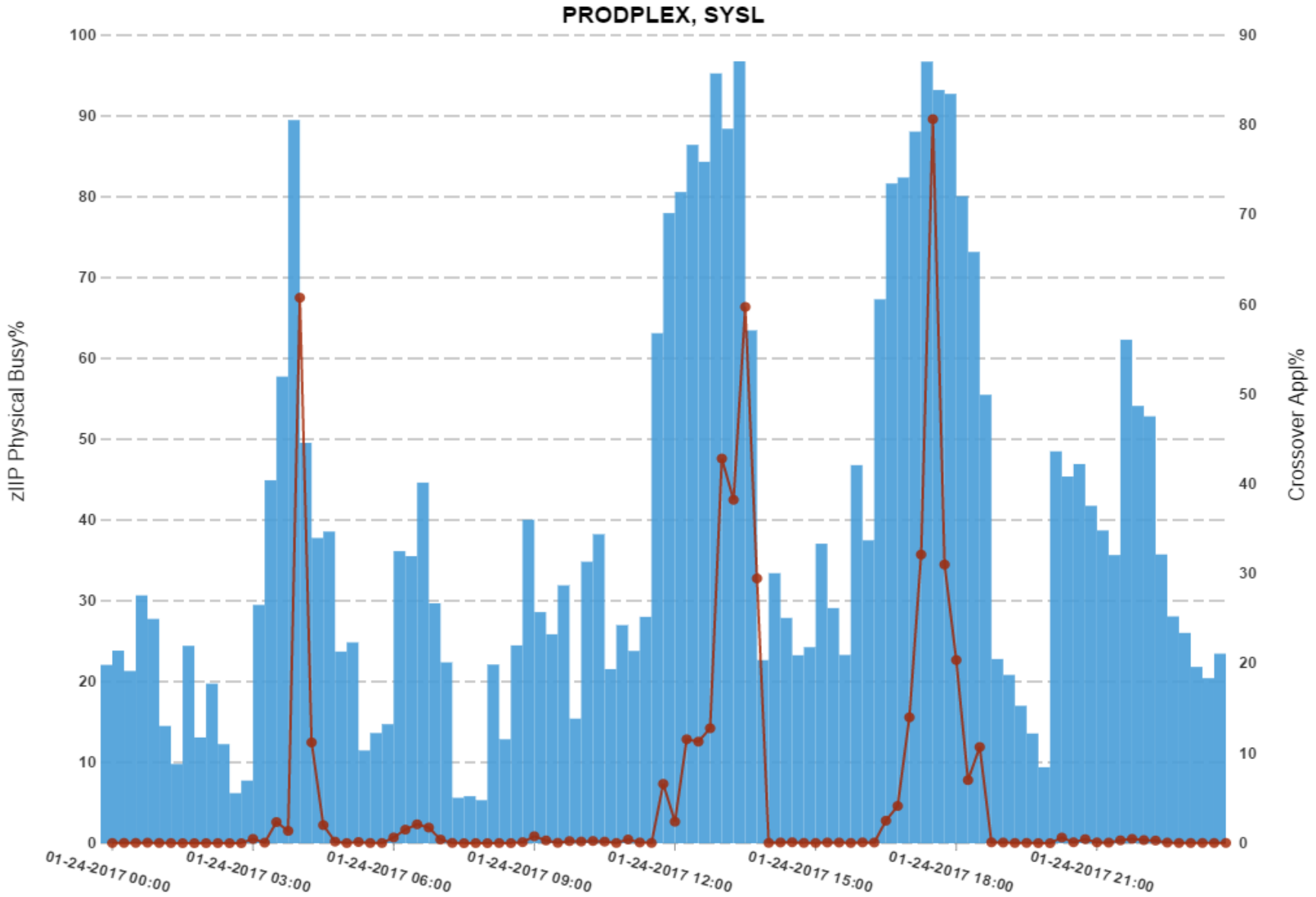
More CPs (or zIIPs) means less queueing at a given utilization level and better scaling to higher utilizations.

CEC Physical Machine zIIP Busy%



Example case with zIIP utilization running above 90% for some intervals, and above 80% for several.

zIIP APPL% Crossover CPU vs. Physical Busy



But the amount of cross-over to the GPs is not necessarily well-correlated to the total zIIP utilization.

Summary: what we really think

- Keep goals “tight” for work you need to protect
  - But it is probably ok to relax this guideline for lower importance workloads
  - But also give the workload the goal it needs based on business needs
- Use sub-15ms response times where it makes sense
  - But understand the impact on the user experience
- Calculate LPAR % Busy the traditional way (as RMF and CMF do)
  - But… nothing… Just use the traditional measurement
- Reporting GCPs and zIIPs separately is almost always the right answer
  - But understand that IBM sizing tools sometimes conflate GCP and zIIP measurements
- Use CPU critical control carefully for very few, and select, workloads with predictable CPU usage
- zIIP utilization is not as important as cross-over, and zIIPs can be pushed to high utilizations just as GCPs can.
  - But justification for buying more zIIPs is definitely a lower hurdle than GCPs
  - Also remember the original intention of zIIPs is to lower CPU usage on GCPs (so avoid crossover)

Questions?

Thanks for attending!