IMPACT
CMG - YOUR TECHNOLOGY COMMUNITY

z/OS Performance Risk Management: Easy Things To Do To Reduce the Risk of Bad Performance

Scott Chapman

CIO, Enterprise Performance Strategies

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

- Performance management for z/OS systems is a well-established field, and z/OS has a wealth of performance data and tools to help with that. However, sometimes organizations make decisions or have practices that limit their ability to effectively manage z/OS performance. In this session Scott Chapman will explore some of those anti-patterns, from the perspective of the easy things you can and should do to make it easier for you to understand and manage z/OS performance.

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

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!

Risk #10: Not investing in people

Performance Management is Important!

- Somebody should be responsible for system (and application) performance
- That person (people in larger organizations) need:
  - Training
    - Conferences, webinars, training classes, reading
  - Tools
    - Including both reporting and real-time monitoring
  - Time
    - To practices all of the above and regularly review performance
- In many orgs, performance management is not a full-time role
  - But the less time dedicated to ongoing performance management, likely the longer it takes to get to a bottom of a performance issue when one does occur

Risk #9: Inappropriate SMF/RMF intervals

RMF/SMF Intervals shouldn't be too long

- 90% of sites are using 15 minute RMF/SMF intervals, but we still come across sites using 30 minutes or longer!
- Long intervals can make performance analysis more difficult
  - More problems can hide in longer intervals
- 5 or 10 minute intervals are also good choices
- Use same settings across all systems
- Set your SMF interval to 15 minutes, set RMF to sync with SMF

```
                In SMFPRMxx:

INTVAL(15)  - 15 minute SMF intervals
SYNCVAL(15) - Sync at 15 minutes after hour
```

Some products may require SYNCVAL(59)

```
                In ERBRMFxx:
SYNC(SMF) - Sync with and use SMF intervals

                In CMFCPMxx:
On REPORT statement:
… SYNC=SMF
```

Key Sync Problem

- There are usually subsystem-specific options in your SMFPRMxx and those need to be set correctly too.
  - Sometimes there will be a different INTERVAL set there or NOINTERVAL
  - Default is NOINTERVAL which (I think) overrides the global interval
  - Easy answer: specify INTERVAL(SMF,SYNC) on the SYS and SUBSYS statements

```
SYNCVAL(00)                    /* SYNCRONIZE ON THE HOUR          */
INTVAL(15)                     /* STANDARD RECORDING INTERVAL     */
…
SYS(TYPE(0:125,127:255),INTERVAL(SMF,SYNC),DETAIL)
SUBSYS(STC,EXITS(IEFU29,IEFU83,IEFU84,IEFUJP,IEFUSI,IEFUSO),INTERVAL(SMF,SYNC))
```

If you aren't syncing your SMF intervals, you won't get new interval records coming in/out of system recovery boost, making those records that include boost periods problematic!

Risk #8: Not recording useful data

Finding transient performance problems

- Today you should be recording the SMF 98 and most SMF 99 records
- These records record in sub-minute intervals (e.g. 2, 5, or 10 seconds)
- Not as much detailed data as in RMF/CMF, but very useful for zeroing in on transient performance problems and evaluating performance on those short intervals

- Many sites haven't enabled SMF 98
- Many sites have 99s excluded due to IBM recommendations from 1995
  - May have been some validity to those recommendations then, but times and hardware capacity have changed!

SMF Records to Include

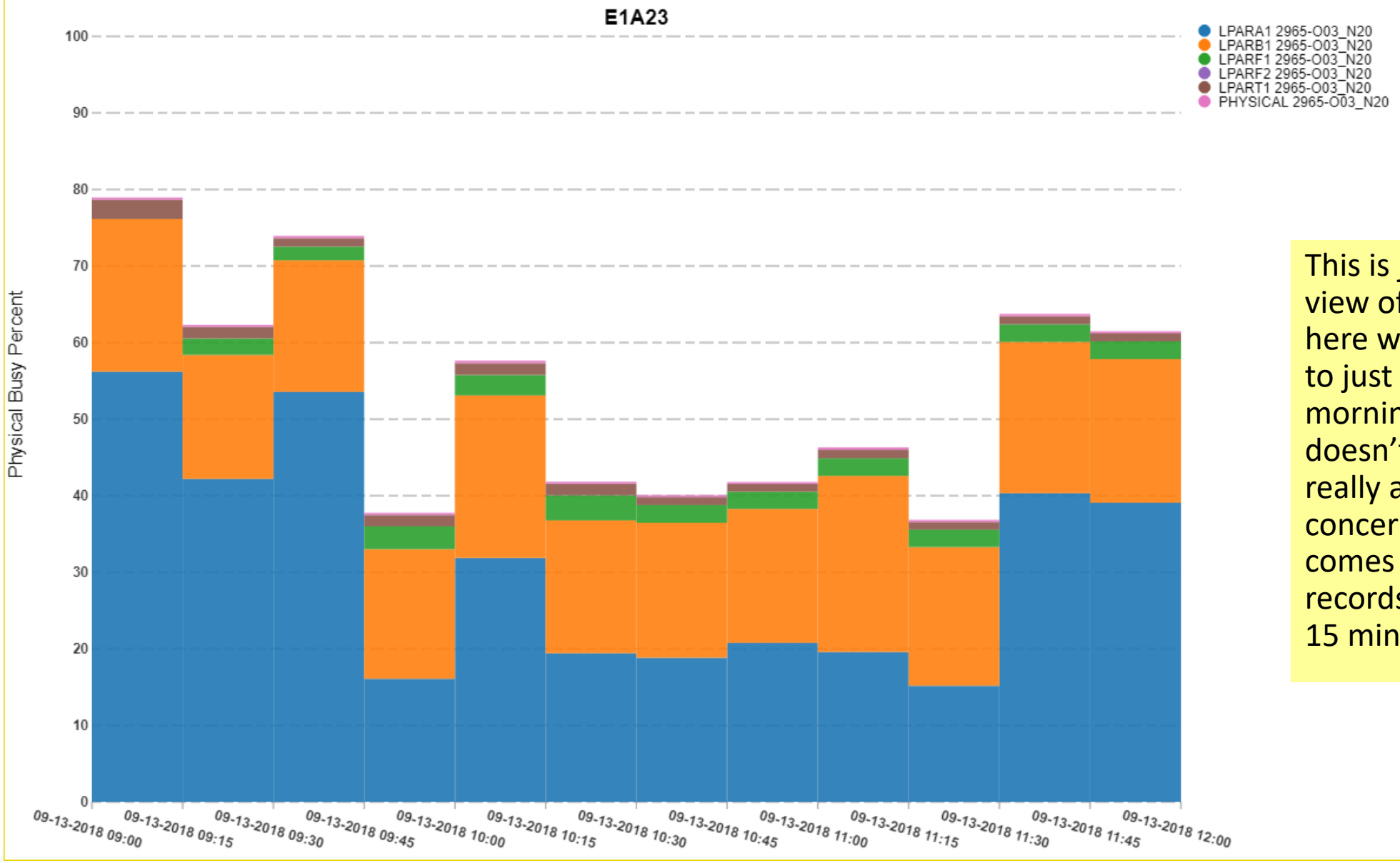
- New 98 High-frequency throughput statistics
  - IBM recommendation is to record on 5 second interval
    - Can use 5, 10, 15, 20, 30 or 60 seconds
    - 5 second interval is about 400MB-500MB/system/day
- SMF 99 SRM/WLM details
  - Our minimum recommended subtypes: 6, 10, 11, 12, 14
    - These will total around 50-150MB/system/day
  - Subtype 1, 2, and 3 can be quite useful, but can be more voluminous
    - These can be 1-1.5GB/system/day
  - Pivotor customers: send them if you're collecting them!
  - Subtype 13 is fairly voluminous and is undocumented "IBM use only"
    - 150-200MB/system/day
- SMF 113 - HIS
  - Most sites have enabled this, but if you haven't: do so now

```
In SMFPRMxx:

HFTSINTVL(15)
```

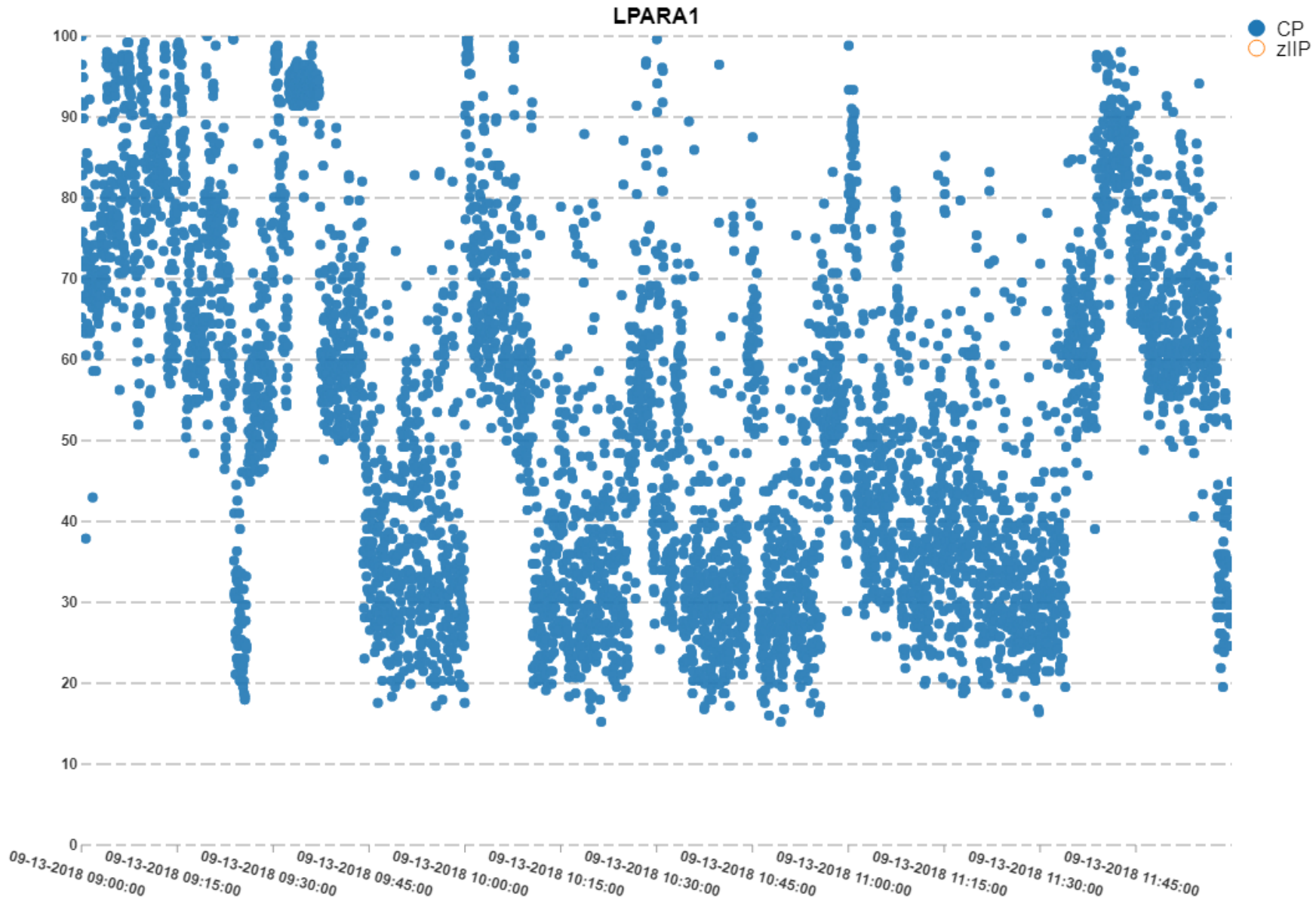
None of these records represent data you'll look at every day, but it's nice to have them available when you need them!

CEC Physical Machine CP Busy% by CEC Serial Number

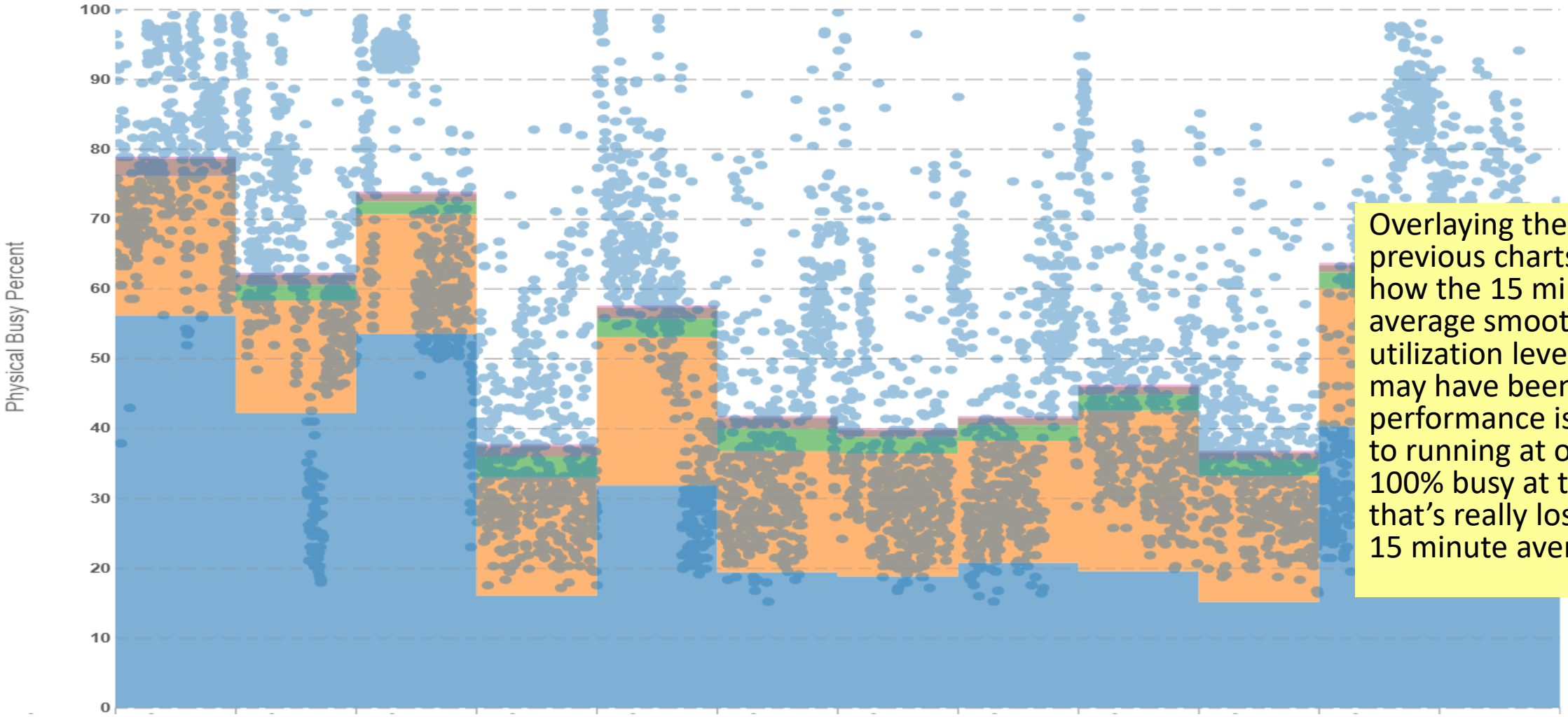


This is just a standard view of CEC Utilization, here we've narrowed in to just 3 hours in the morning, where it doesn't appear there's really any capacity concerns. This data comes from the SMF 70 records, in this case on 15 minute intervals.

HiperDispatch CEC Utilization



This data comes from the 99.12 HyperDispatch records and shows the CEC utilization at 2 second(!) intervals. Note that this tells a different story than the 15 minute RMF intervals.

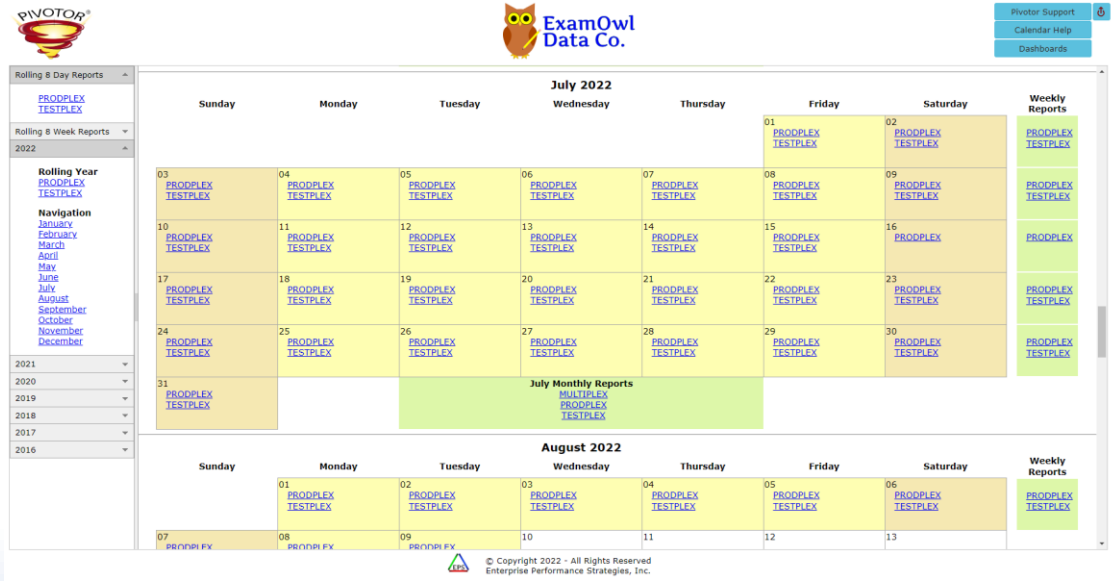


Overlaying the two previous charts shows how the 15 minute average smooths out the utilization levels. There may have been performance issues due to running at or nearly 100% busy at times, but that's really lost in the 15 minute averages.

Risk #7: Lack of usable & useful tooling

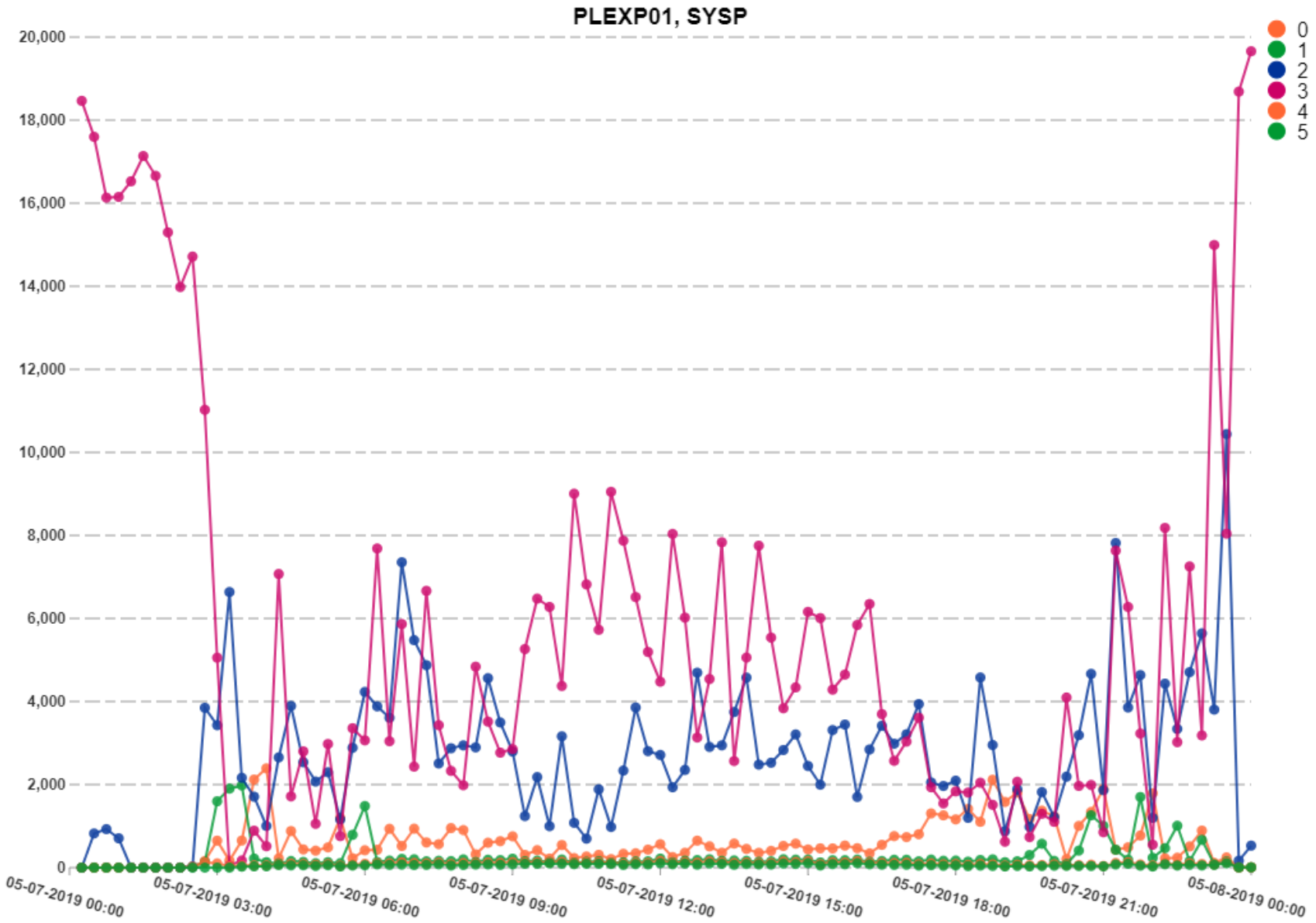
You have a wealth of data, is it usable?

- The mainframe is blessed with a wealth of performance data
- Not every site has easy, convenient, and useful access to it
- Build or buy a process/product such that you have ready access to performance data on a regular basis
  - Most already have tooling, but understanding that tooling is sometimes lacking
  - Make sure your performance people have time to learn how to use their tools



Risk #6: Old CPENABLE setting on z14/z15

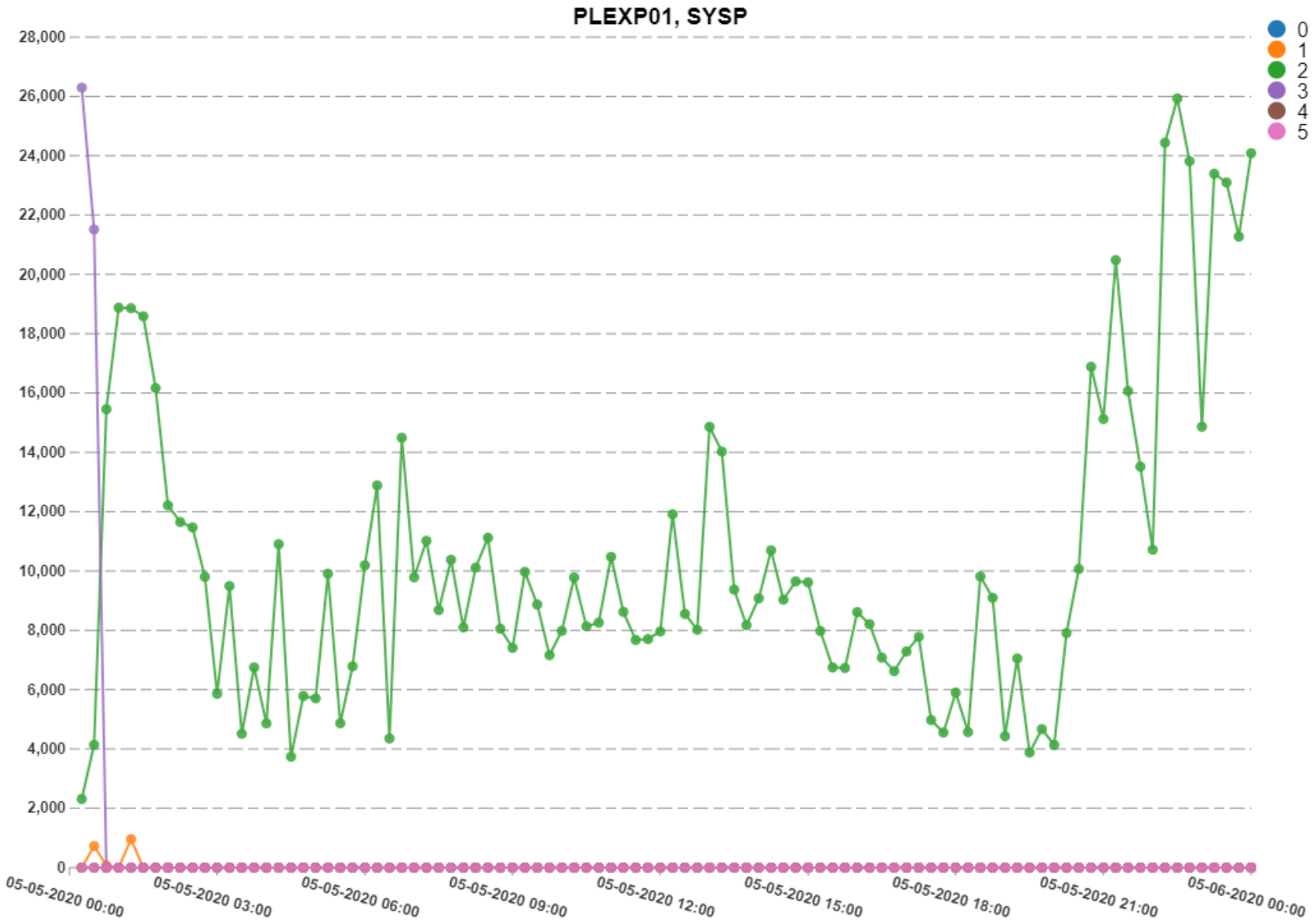
LPAR - I/O Interrupt Rate by Processor



This shows a typical system on a z13 (or earlier) processor with CPENABLE=(10,30). A limited subset of processors are handling I/O interrupts.

Allowing all processors to handle interrupts is less efficient than having a limited subset handle them.

LPAR - I/O Interrupt Rate by Processor

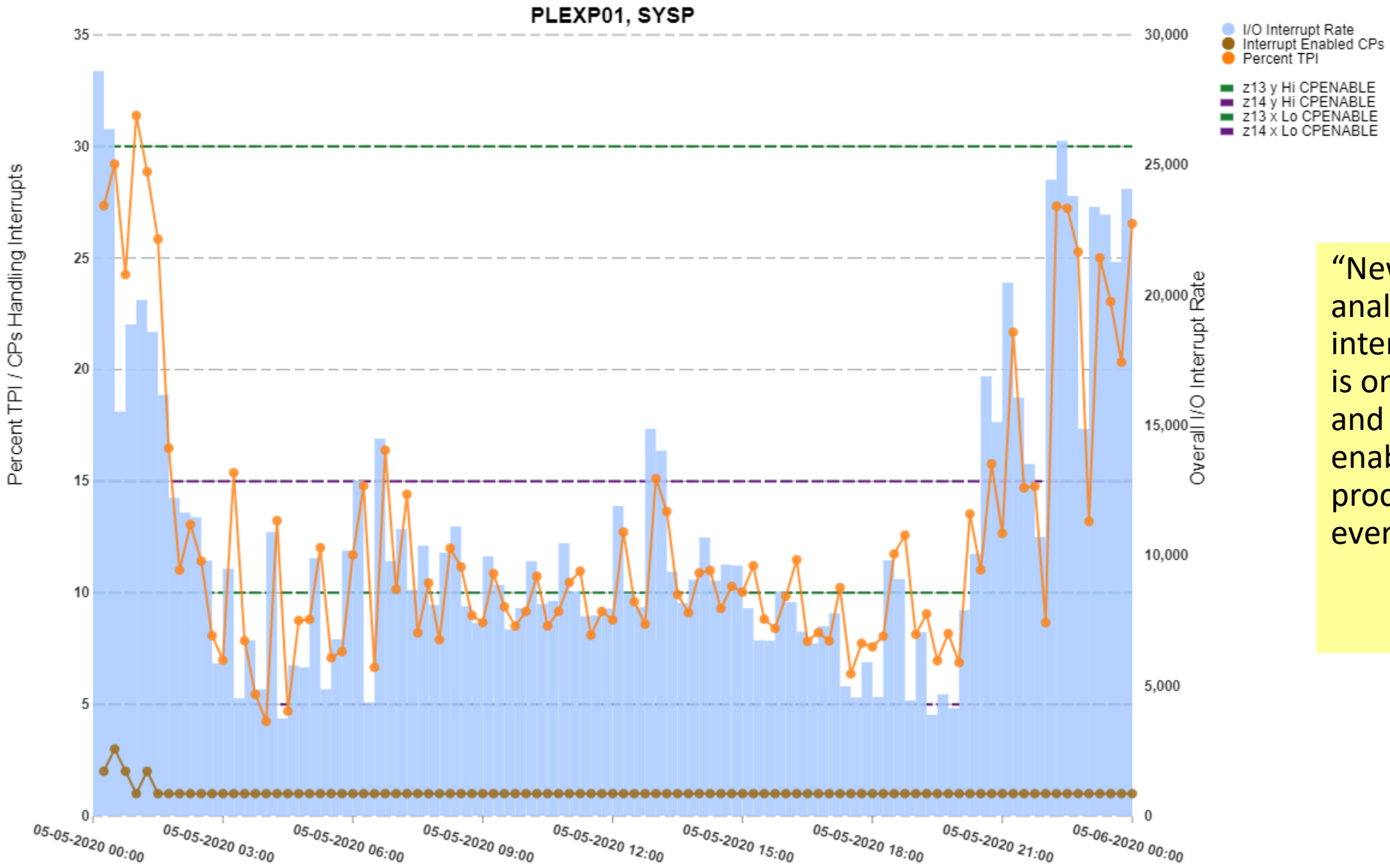


That same system after migration to a z14. Now only 1 CPU is enabled for I/O interrupts.

That is more efficient, but the risk is that I/Os could be delayed if that CP is busy such that it can't handle the interrupts.

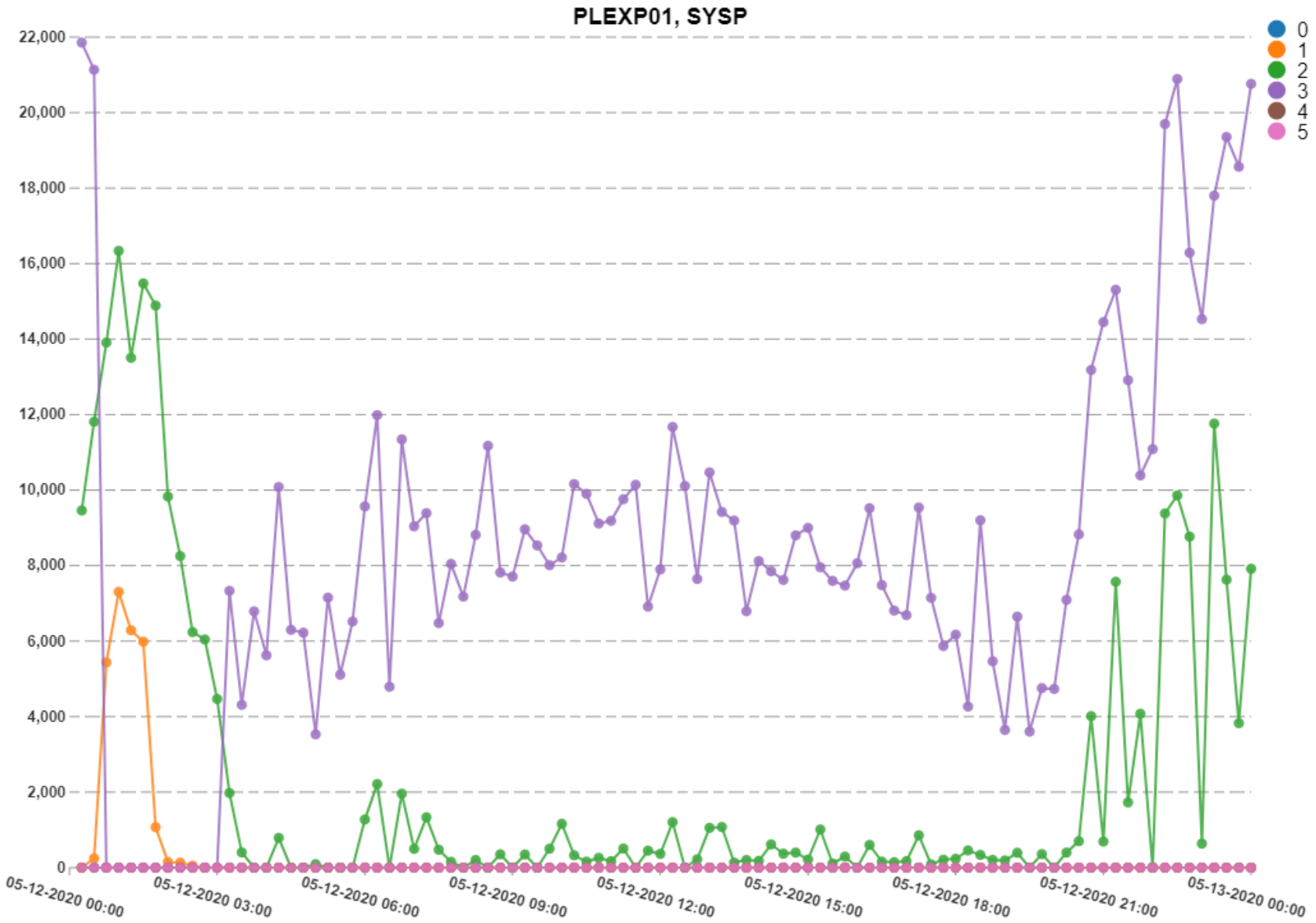
I/O Interrupt Analysis

(CPENABLE=(x,y) recommended settings)



"New" report to help analyze the I/O interrupts. But note this is on 15 minute intervals and decisions to enable/disable processors are made every 20 seconds.

LPAR - I/O Interrupt Rate by Processor

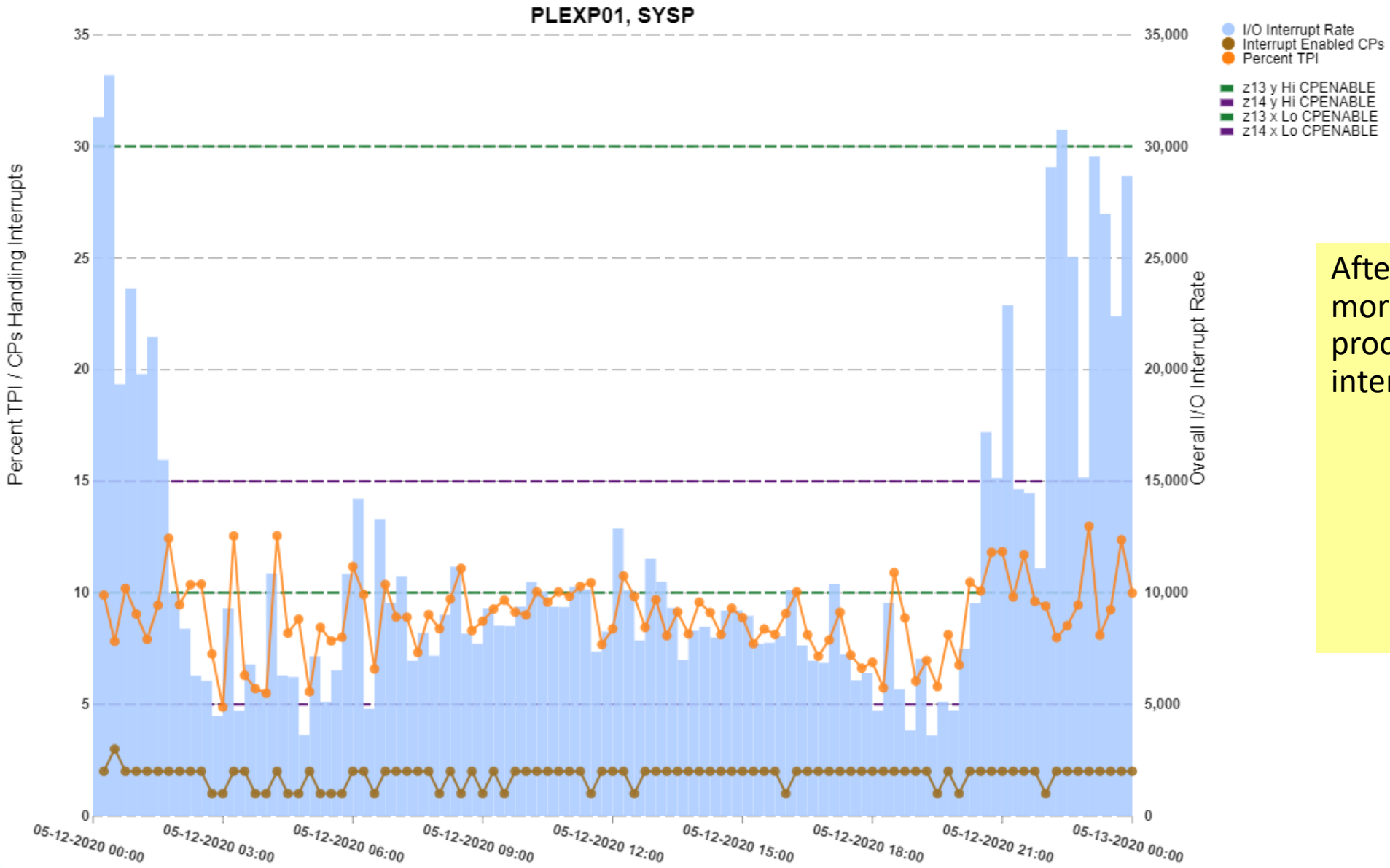


Z14 ZR1 after changing to new IBM recommendation of CPENABLE=(5,15).

Note that there's now usually two CPs enabled for interrupts, even though one is still primarily servicing the I/Os.

I/O Interrupt Analysis

(CPENABLE=(x,y) recommended settings)



After CPENABLE(5,15) more commonly have 2 processors handling interrupts.

Risks and Mitigations

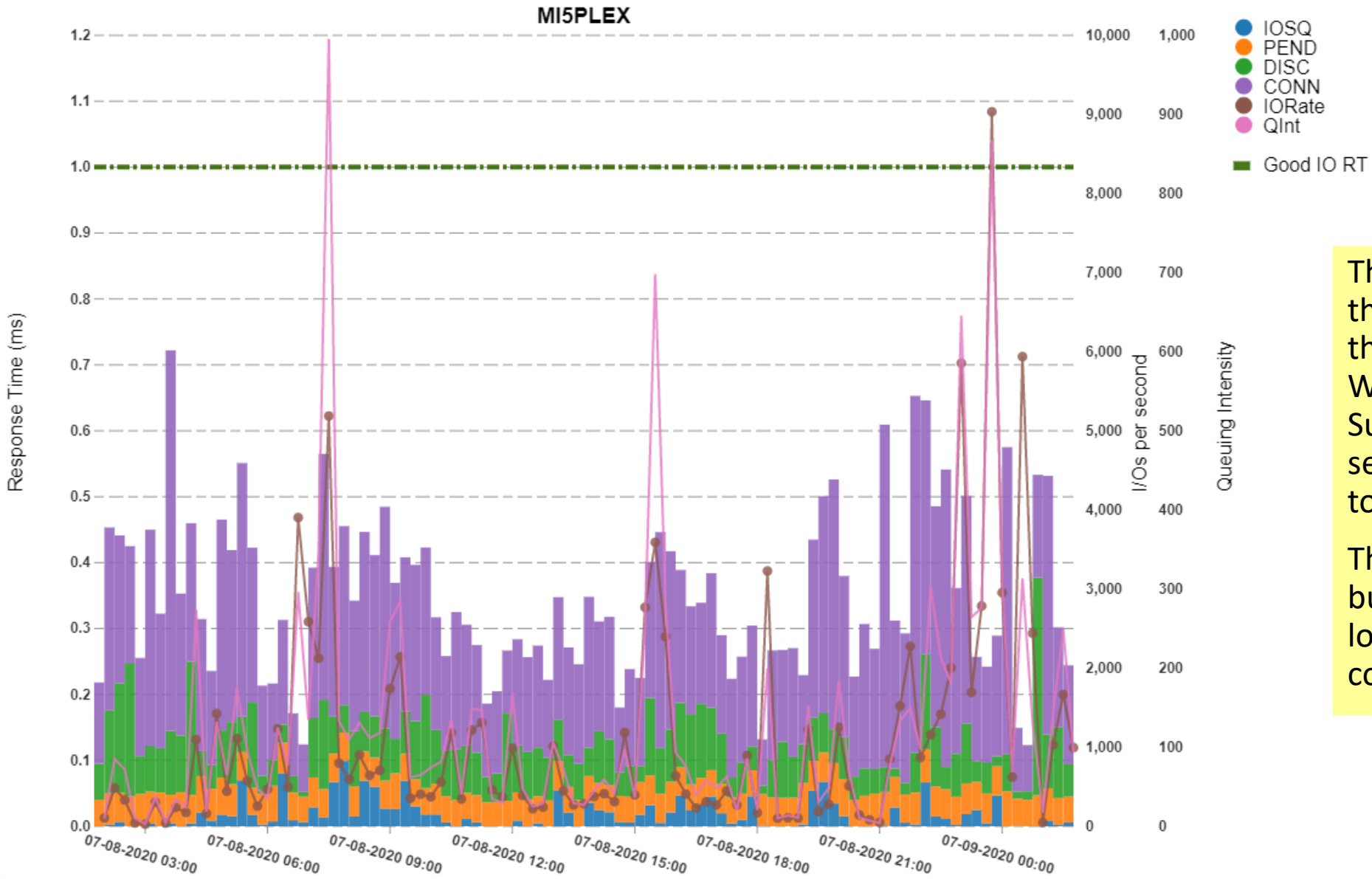
- Risk is that I/Os could be delayed (perhaps severely) if the CPU to handle the interrupts is busy in such a way that it can't handle the interrupts
  - Could be especially problematic if those I/Os are critical
- Risk is less on LPARs with more CPs, greater on LPARs with fewer CPs
- On z14 (and above) processors, new IBM recommendation for CPENABLE is (5,15) not the old (10,30)
  - This is probably a good starting position, but some environments might need to tweak the settings to help ensure most of the time there's two CPs enabled
- If you're going to DR on a z13 or older, probably ok to run with (5,15) during a DR test, or probably even a real DR
  - Risk is that with (5,15) the z13 might enable one more CP than is absolutely needed, a relatively minor efficiency hit

Risk #5: Too few PAVs defined

(Or possibly SuperPAV not used)

DASDplex RT Components

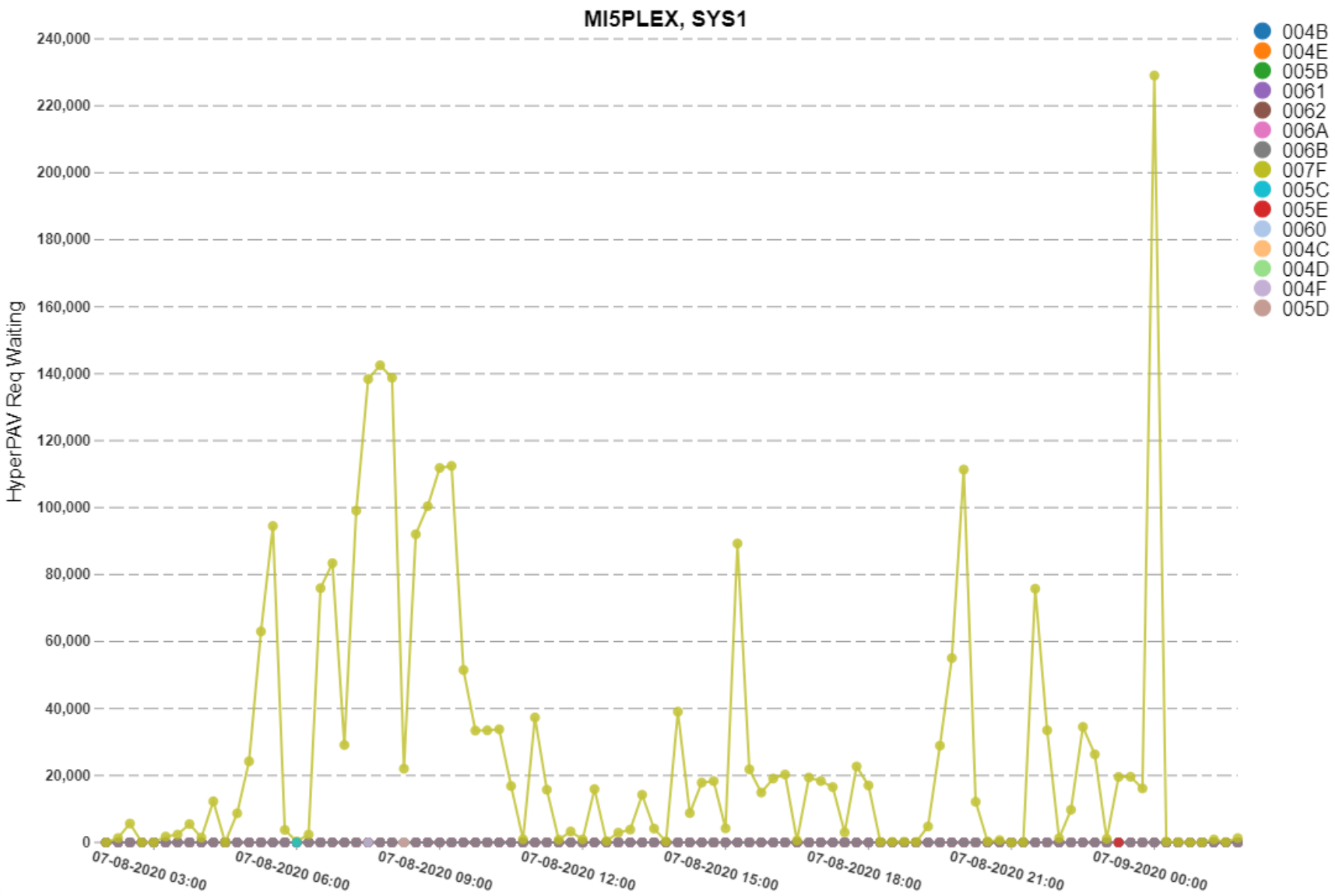
Including I/O rate, Queuing Intensity



The blue on this chart is the IOSQ component of the DASD response time. With HyperPAV and SuperPAV, we expect to see very little IOSQ time today.

This looks small here, but it's actually quite a lot for a modern configuration.

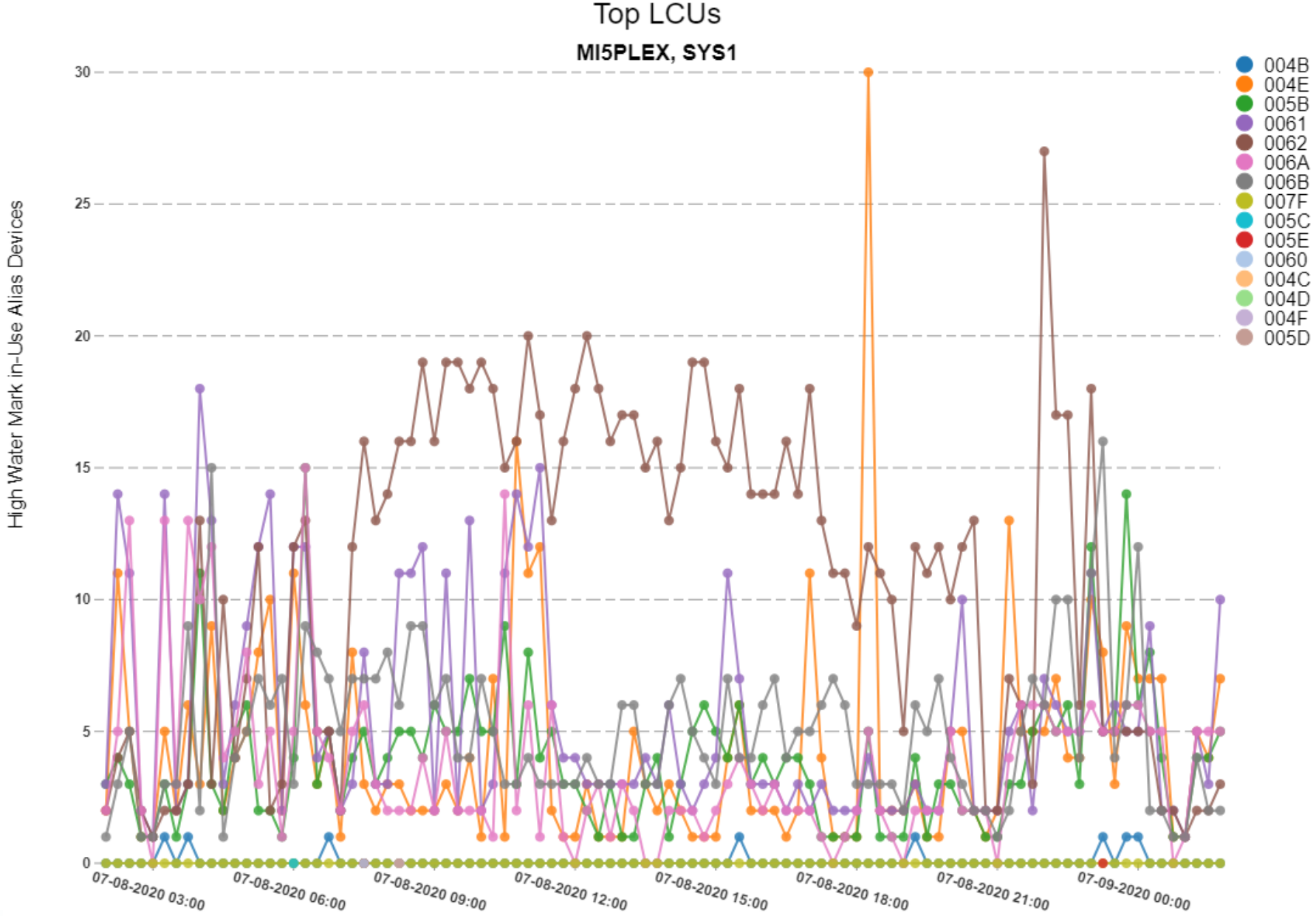
HyperPAV - I/O Not Started Because No HyperPAV-aliases Were Available



How many I/Os are not being immediately started because of lack of a PAV?

Often multiple LCUs are impacted, but here it's only a single LCU.

HyperPAV - Top LCU High Water Mark of In-Use HyperPAV-alias Devices



Often the HWM of in-use PAVs will flat-line at the number of defined PAVs in the LCU (at least for HyperPAV).

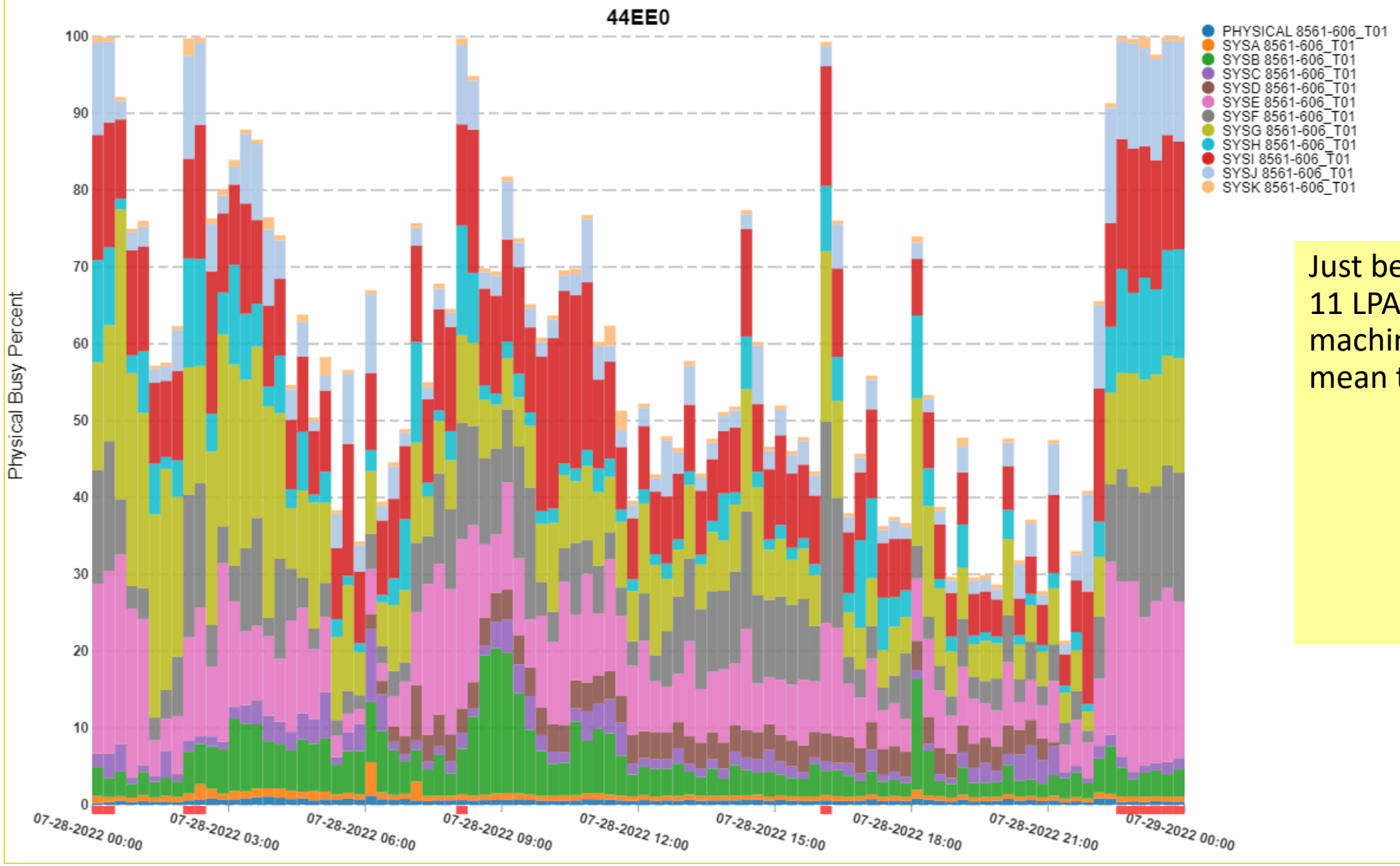
Here the problematic LCU apparently has no PAVs defined!

Risks and Mitigations

- IOSQ time should be almost non-existent in a well-configured environment with Hyper- or (especially) Super-PAV
- Regular indications of some IOSQ time may indicate a potential limit for increasing your I/O workload
  - May be caused by bursts of I/Os, but still: you may not be able to grow those bursts
- Adding PAVs for affected LCUs is often difficult to impractical
- Next best answer is to rebalance busy logical volumes between LCUs
- Note that SuperPAV allows LCUs to borrow PAVs from other LCUs and so is much less susceptible to these balance issues
  - If your control unit supports SuperPAV make sure IECIOSxx contains HYPERPAV=XPAV
  - Pro Tip: your control unit almost certainly supports SuperPAV

Risk #4: Too few CPs

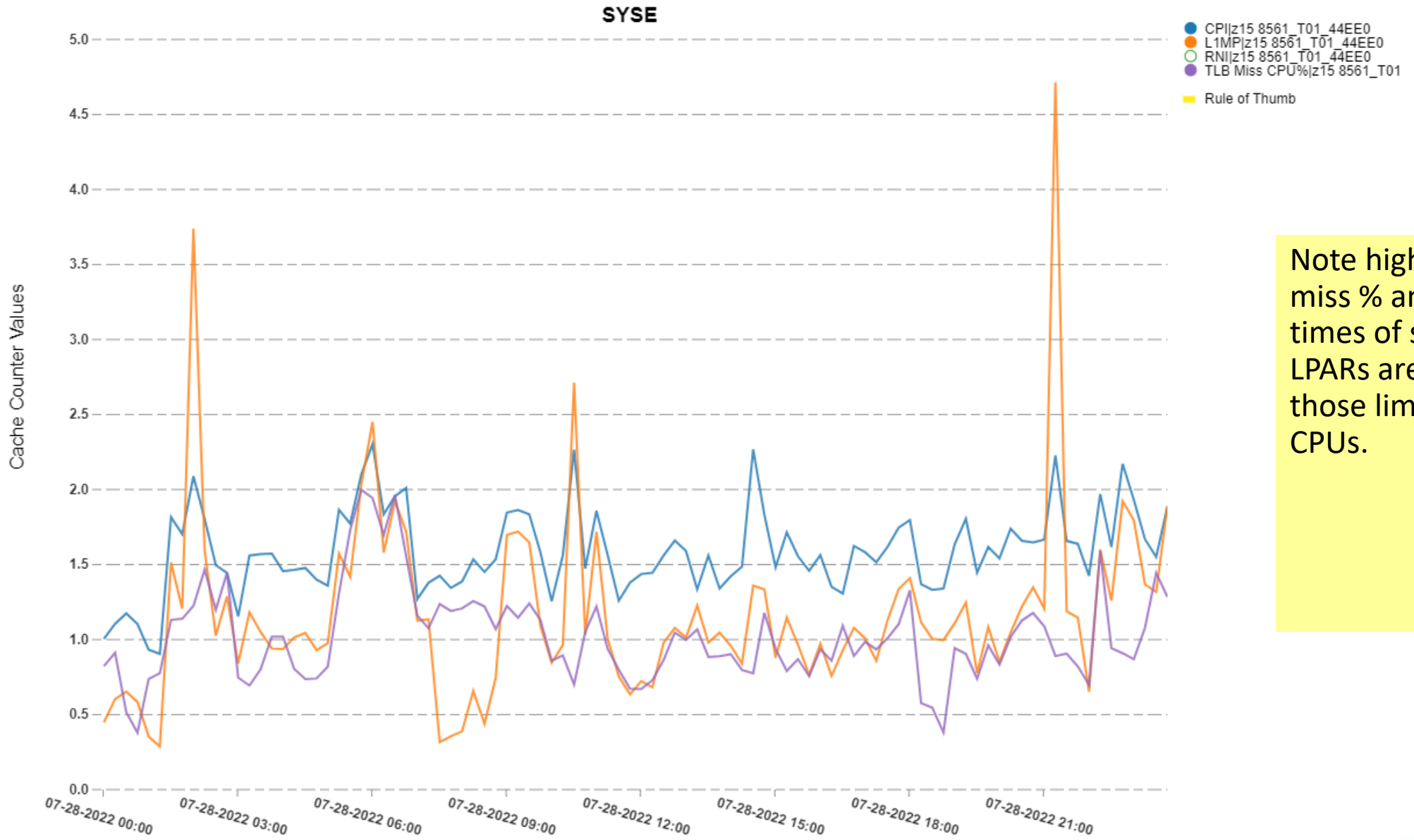
CEC Physical Machine CP Busy% by CEC Serial Number



Just because you can run 11 LPARs on a 6-way machine, that doesn't mean that you *should*!

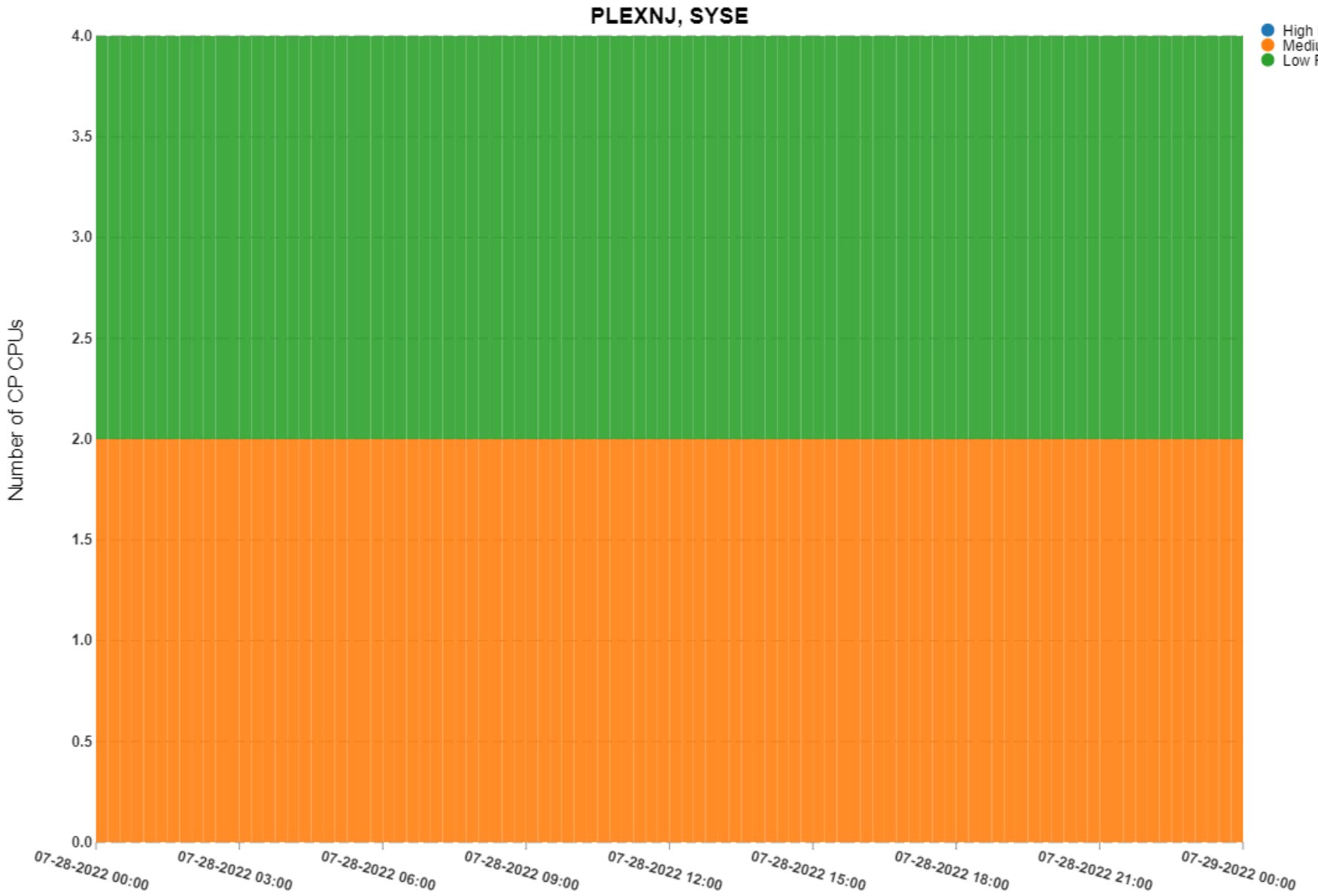
Processor Caches - Key Measurements for Processors

SMF 113



Note higher CPI, TLB miss % and L1MP during times of stress as more LPARs are contending for those limited physical CPUs.

HiperDispatch CP CPU Pooling at End of Interval



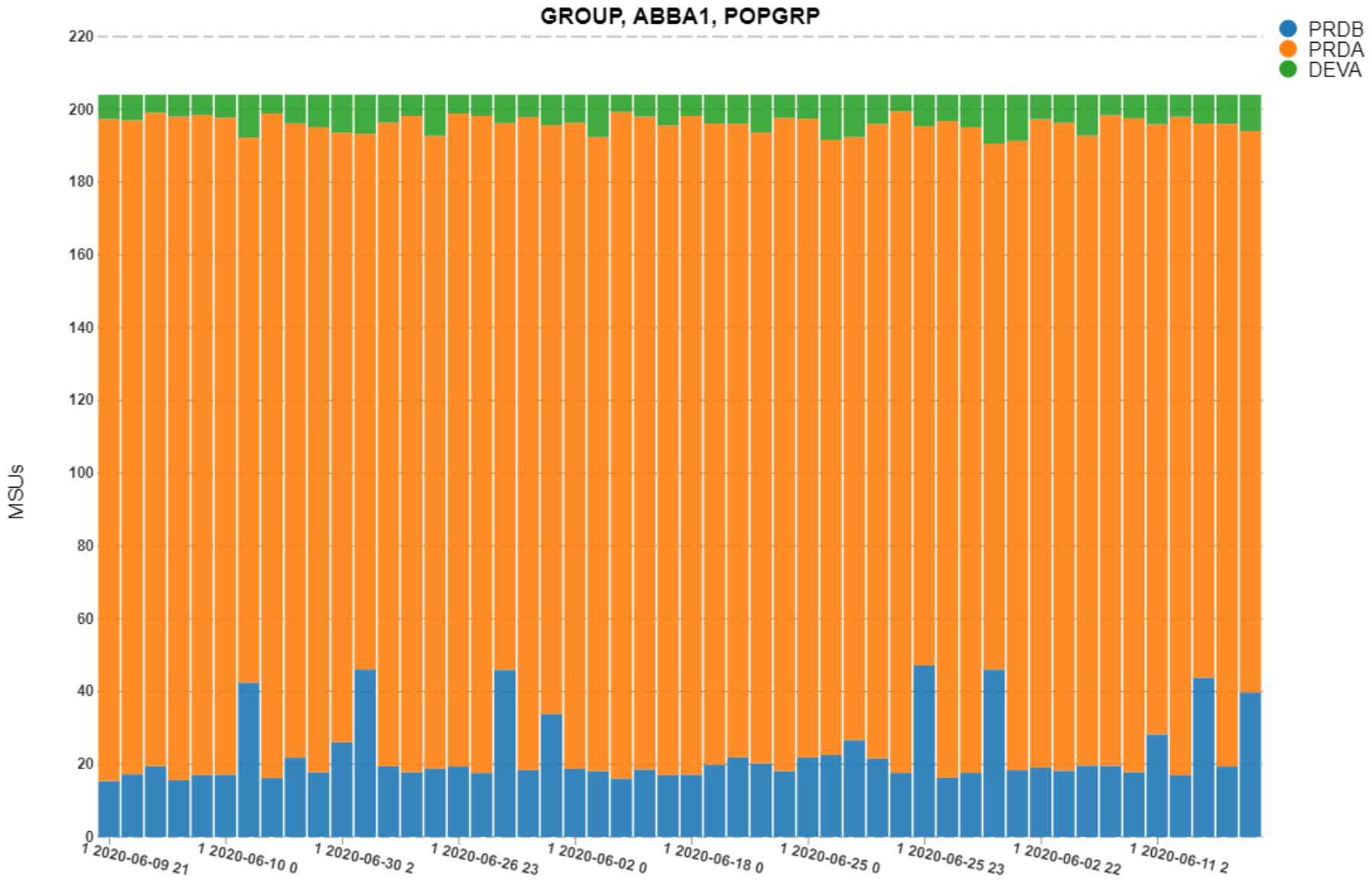
This LPAR never gets to enjoy a high pool processor.

Risks and Mitigations

- This is really about efficiency and more/slower vs. fewer/faster
- In many cases, more/slower CPs will be more efficient than fewer/faster
- Especially when you have so many LPARs sharing so few CPs
  - Extreme sharing is more plausible if most of the LPARs are small
  - Or the usage is separated in time
  - But when a significant number all get busy at the same time, efficiency will suffer
- Use zPCR while planning for your processor to find an ideal configuration
  - In this case a 512 probably would have been a better choice than a 606
    - And is rated for fewer MSUs, making the software bill cheaper, while likely delivering better overall performance
  - However, do your due diligence about slower processors
    - Single-threaded workloads running at non-busy times to be the most likely impacted
    - CICS regions heavily reliant on the QR TCB could be problematic

Risk #3: Dev/test LPARs in capacity group with prod

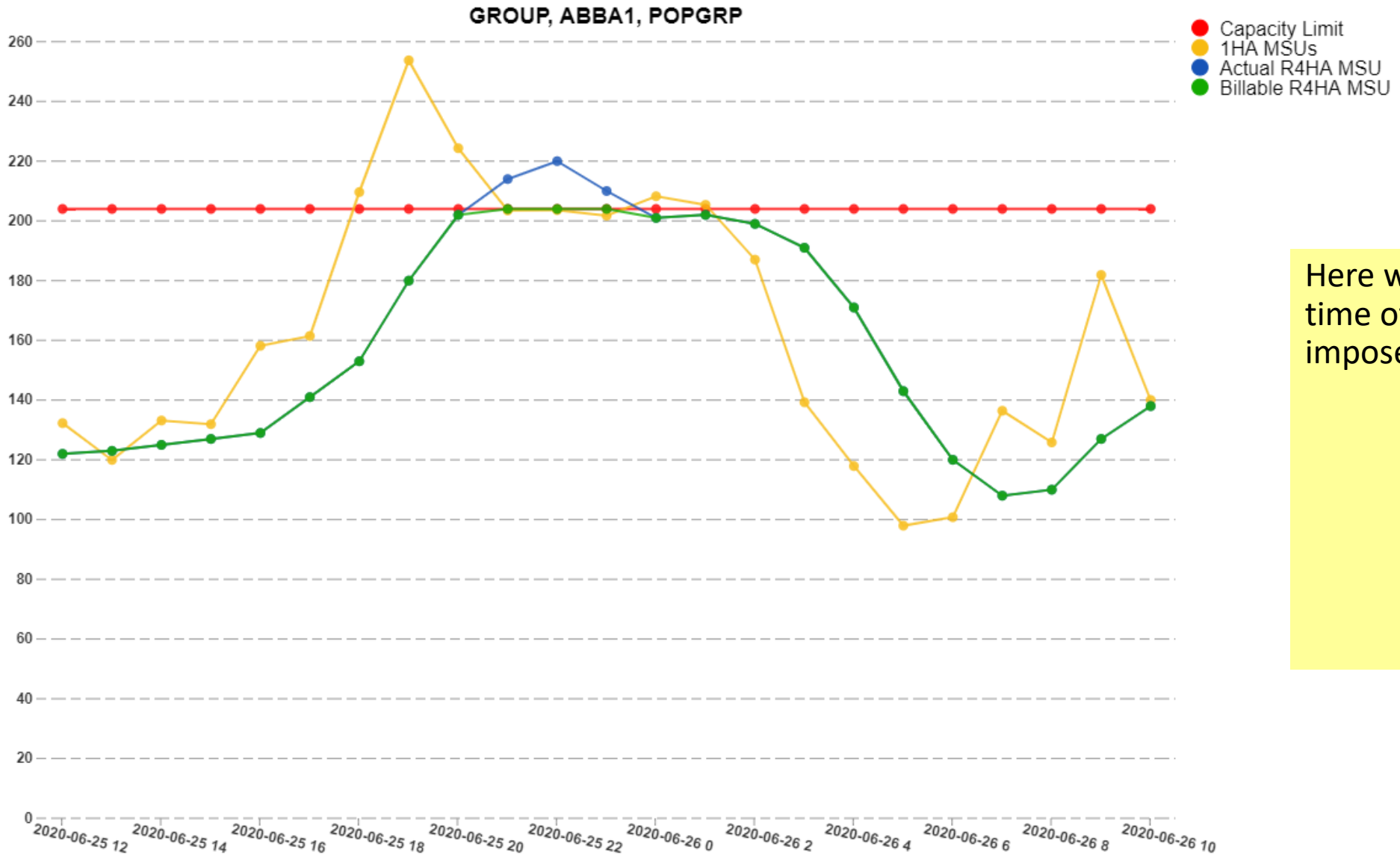
Top R4HA Intervals by System



In this case, we have a capacity group with three members, one of which appears to be non-production. The rolling 4 hour average of the combination of all 3 LPARs determines when the group will be capped.

For these top 50 intervals, DEVA is not a significant contributor (maybe 10-15 MSUs), but it could be in other intervals.

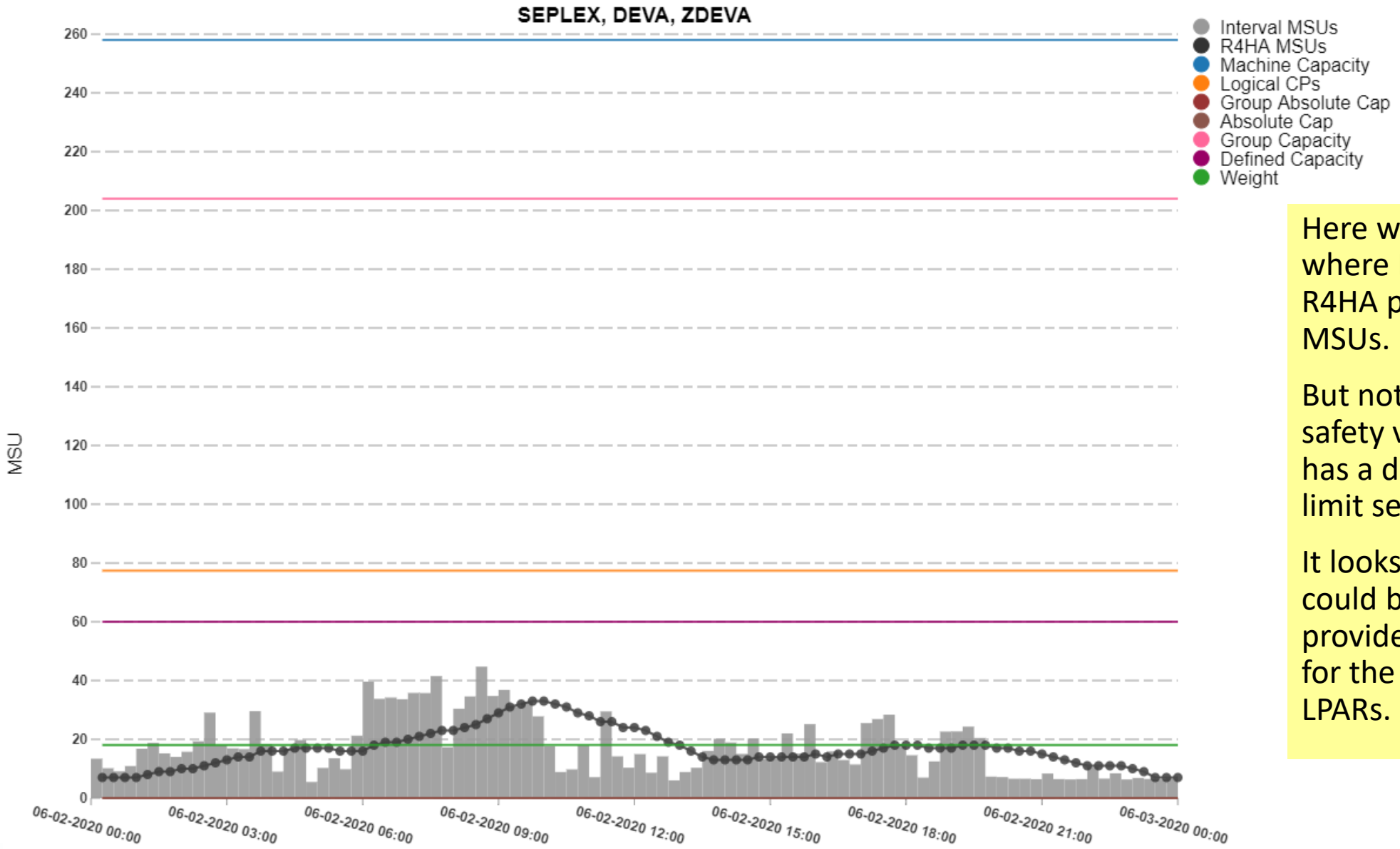
MSU Averages Comparisons



Here we see an example time of the cap being imposed.

LPAR Limits and Utilization

Expressed as MSUs



Here we see an example where DEVA reaches a R4HA peak of about 35 MSUs.

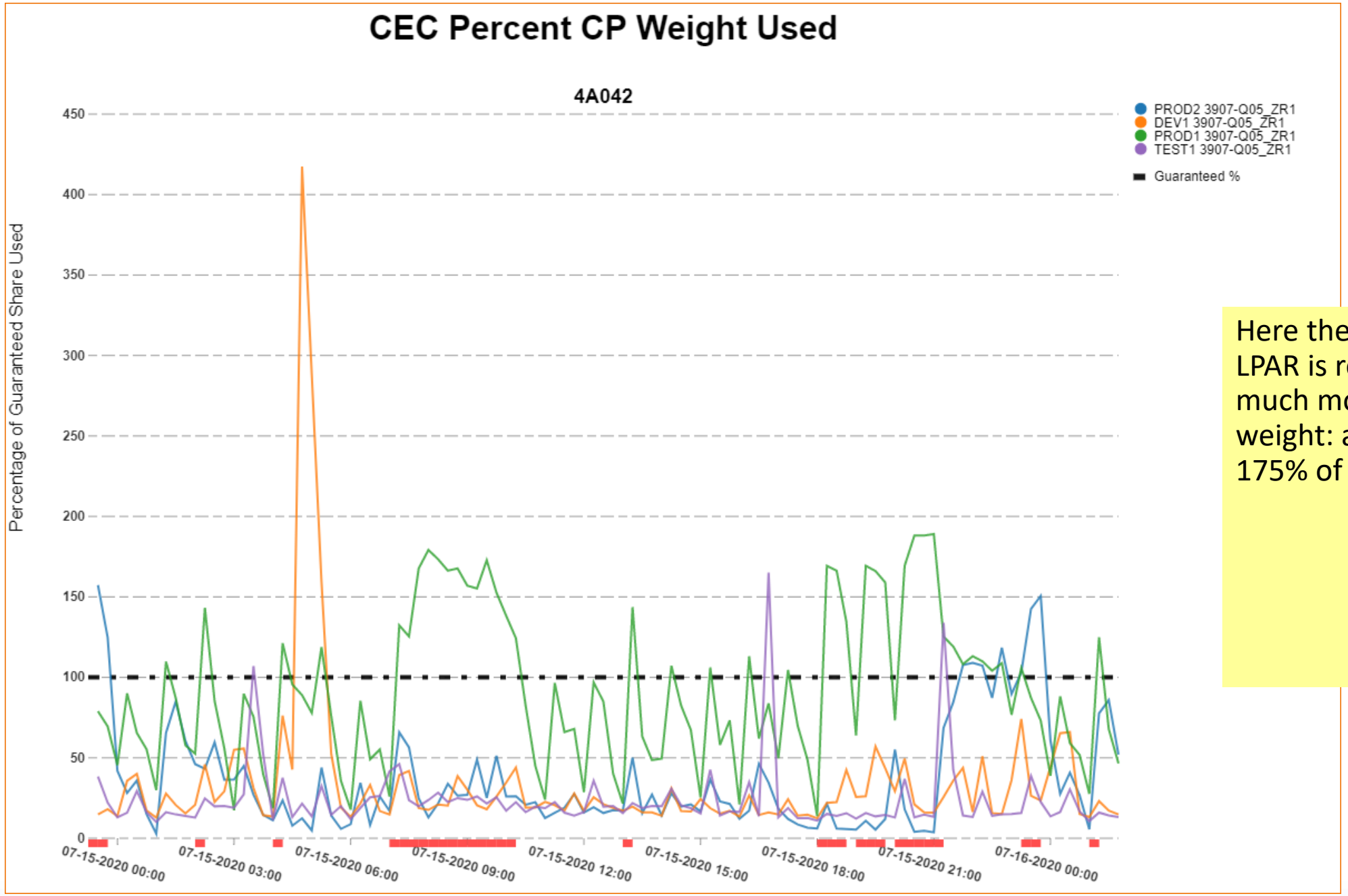
But note the important safety valve: DEVA also has a defined capacity limit set at 60 MSU.

It looks like maybe that could be lowered to provide more protection for the production LPARs.

Risks and Mitigations

- Having all LPARS (production and not) in a single capacity group for the CEC provides the most cost protection for the organization
  - In general, I'm in favor of this
  - But even better would be to use a dev/test container (talk to your IBM MLC rep)
- Risk is that an increase in work on the non-production LPARs could cause the cap to be enforced, potentially impacting production work
- Not including non-production LPARs can greatly reduce the ability to guarantee the peak R4HA for the CEC
- Potential compromise is to include a defined capacity limit for the non-production LPARs that will stop them from running away with too much capacity
  - Note you can use BCPii to change capacity limits dynamically too

Risk #2: LPARs using more than their weight



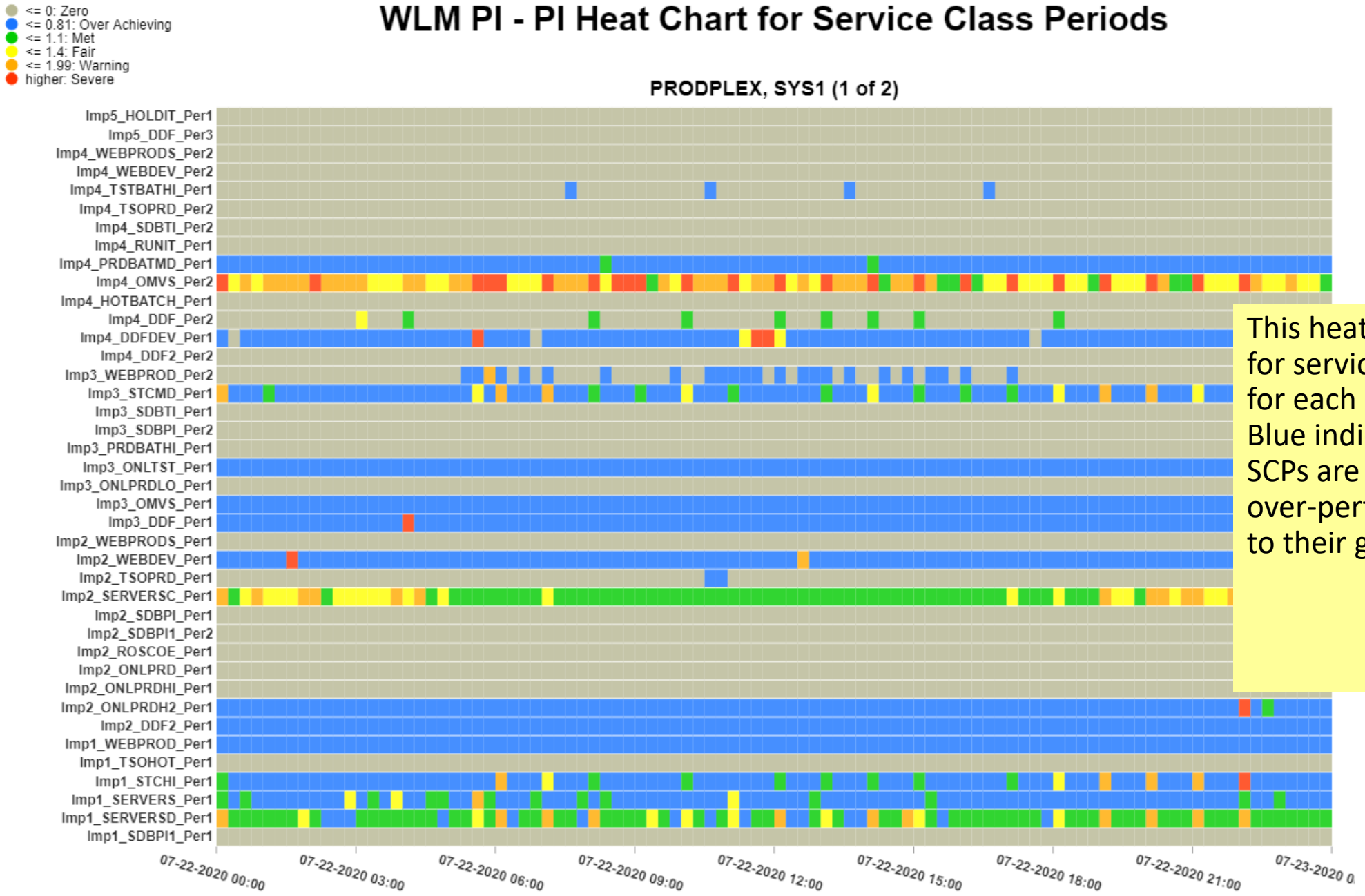
Here the green PROD LPAR is regularly using much more than its weight: at times around 175% of its weight.

Risks and Mitigations

- Important LPARs regularly consuming their weight are at risk of not being able to access that capacity if the other LPARs are busy
  - Classic case: increase in activity on test LPAR causes production LPAR to be limited to its weight
- In the previous slide, PROD was at risk of losing access to about 40% of the capacity it was consuming
  - That would elongate work running on that LPAR
- Make sure your weights are such that they give your important LPARs enough weight to satisfy their capacity needs
- If workload balance shifts at different times, consider using automation to change the LPAR weights
  - Can do that with BCPii which can be driven by REXX scripts

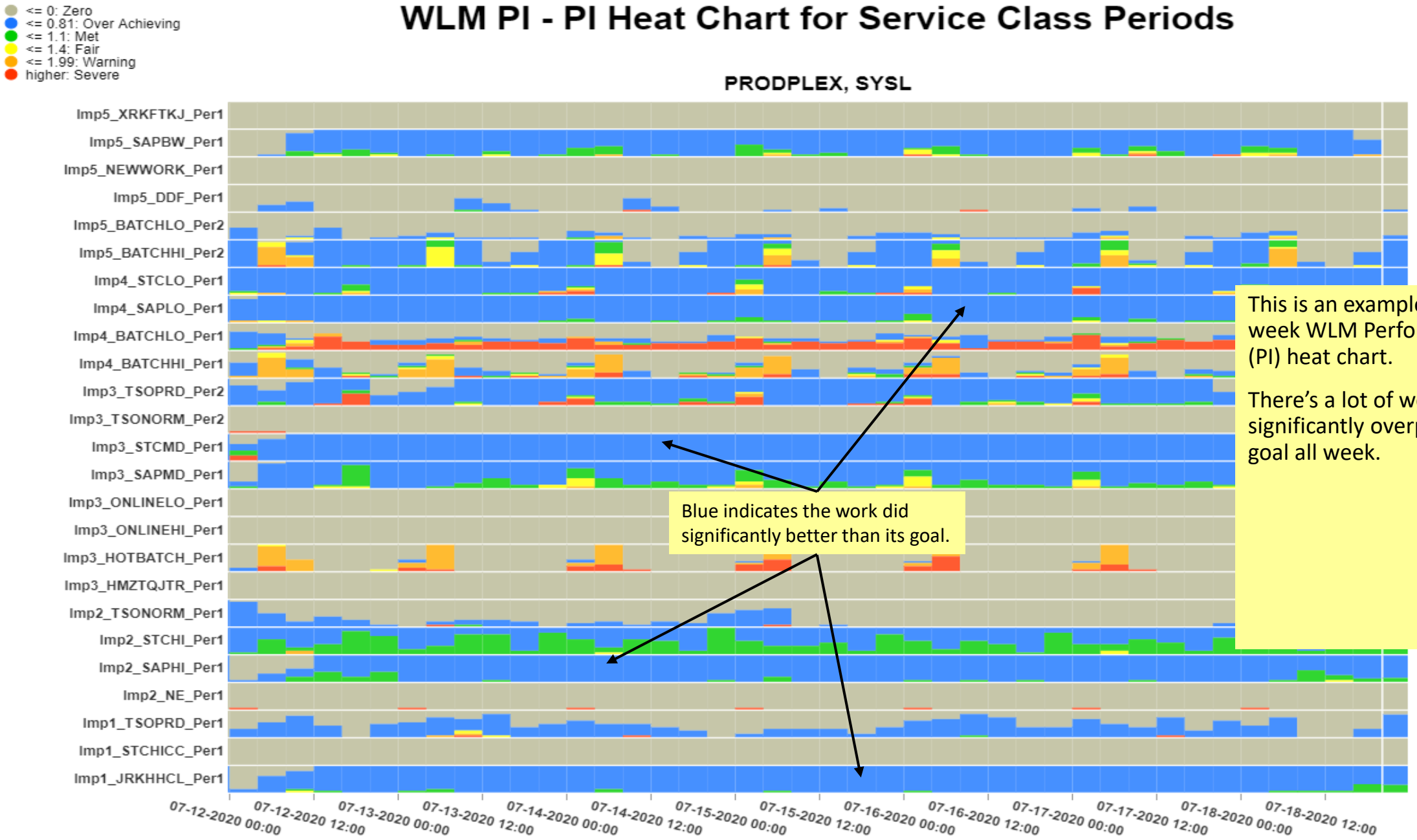
Risk #1: Too easy goals

WLM PI - PI Heat Chart for Service Class Periods



This heat chart shows PIs for service class periods for each RMF interval. Blue indicates that the SCPs are significantly over-performing relative to their goal.

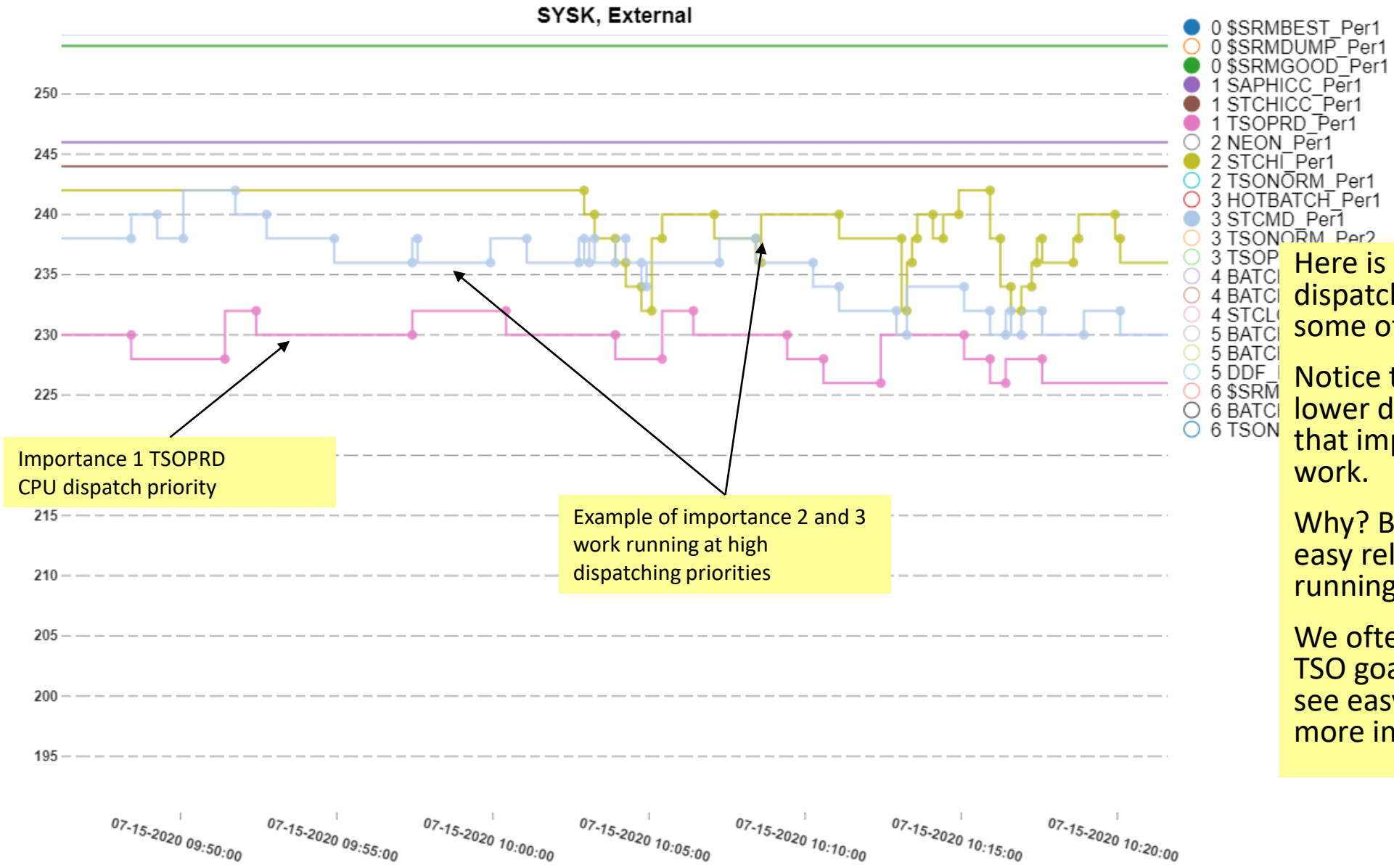
WLM PI - PI Heat Chart for Service Class Periods



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

There's a lot of work here that's significantly overperforming its goal all week.

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 that importance 2 and 3 work.

Why? Because its goals is easy relative to the work running.

We often see very easy TSO goals. Sometimes we see easy goals for even more important work.

Risks and Mitigations

- We see easy goals all the time—sometimes extremely easy goals
  - E.G. TSO period 1 with 80% less than 1 second, but actually achieving something like 99% < 1 second, with an average of 0.1 seconds
- When resources become constrained work that's running with a very easy goal may degrade to its goal
  - So if the above TSO users start to see lots of transactions in the 1 second range, do you think they'll notice? Will they be satisfied?
- For important work that you need to protect: tighten up the goals
  - If the goal is already reasonable compared to user expectations, then maybe it's ok
  - I.E. if the above goal was 95% under 0.2 seconds, then maybe that's ok, even though it's over-performing
    - Although it may still be susceptible to brief degradations in changing situations

Summary Highlights

- For goals that are easy, make sure you won't mind if the work degrades to goal
- If important LPARs are regularly using more than their weight, consider what will happen if they lose access to that capacity
- Consider defined capacity limits for dev/test LPARs in a mixed capacity group for a little extra protection for production
- Consider more/slower instead of fewer/faster processor
- You shouldn't see much IOSQ: check your defined PAVs and/or rebalance LCU activity if it is more than a tiny sliver of response time
- On a z14 or later processor, set CPENABLE to 5,15 instead of 10,30
- Record useful data at useful intervals
- Invest in your people!