High, Medium, Low: Understanding how HiperDispatch Influences Performance in z/OS

Scott Chapman

Enterprise Performance Strategies, Inc.

Scott.Chapman@EPStrategies.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:

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

How do HiperDispatch High, Medium, and Low pool processors influence the performance in a z/OS environment? What does it mean when a logical processor is designated as a high pool processor, and how does a high pool processor impact the performance of a z/OS LPAR differently that a medium pool processor? In this webinar Scott Chapman will discuss HiperDispatch and how various HiperDispatch pooling influences the 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

z/OS Performance workshops available

During these workshops you will be analyzing your own data!

- Essential z/OS Performance Tuning
  - March 20-24, 2023
- WLM Performance and Re-evaluating Goals
  - September 11-15, 2023
- Parallel Sysplex and z/OS Performance Tuning
  - May 2-3, 2023
- 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™

- If you don't see them in your performance reporting tool, or you just want a free cursory performance 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!

Agenda

- HiperDispatch Discussion
- Measurements
- Recommendations

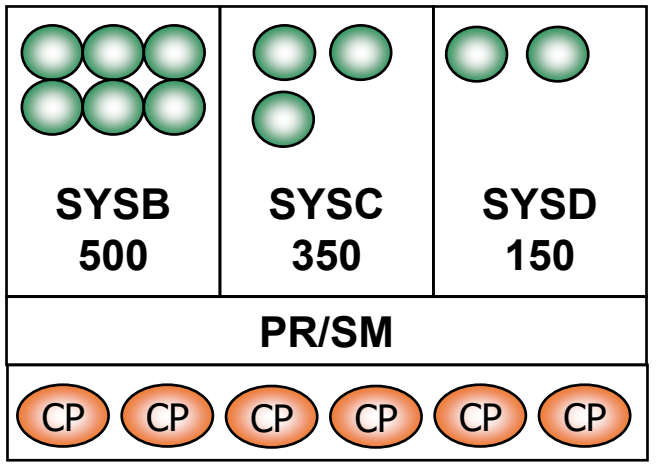
What is HiperDispatch?

Some important things to remember

- A CP can only be in use by 1 LPAR at a time!
  - PR/SM dispatches CPs to LPARs
- LPARs' relative weights determine their relative capacity "fair share"
  - In most environments, LPARs are allowed to use more than their fair share if the other LPARs are not using their capacity allocation
  - All LPARs guaranteed to get at least its fair share
    - Absent capping of course!
  - But if all LPARs have demand for their weight, they'll be limited to their fair share

Weights and logical CPs

- Each LPAR is guaranteed to get at least its share
  - $LPAR\ Share = 100\ * \frac{LPAR\ Weight}{\sum Weight\ of\ activated\ LPARS}$
- In below example:
  - SYSB – guaranteed 50% of capacity of the 6 CPs (3 CPs worth of capacity)
  - SYSC – guaranteed 35% of capacity of the 6 CPs (2.1 CPs worth of capacity)
  - SYSD – guaranteed 15% of capacity of the 6 CPs (0.9 CPs worth of capacity)



Each system has some number of logical CPs

For ease of use, try to make weights add up to 1000 (like they do here).

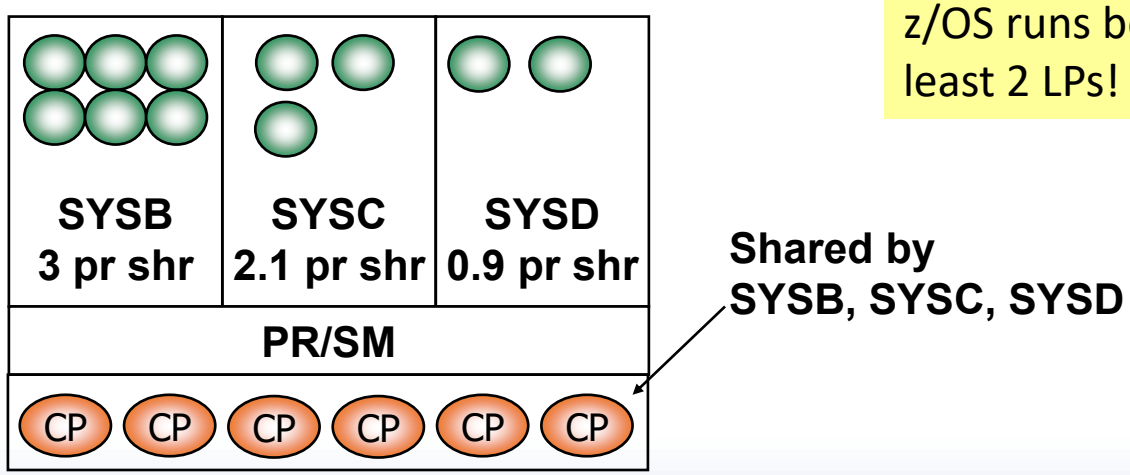
Physical CPs shared by SYSB, SYSC, SYSD

Horizontal CP Management

- Cache effectiveness will be better when a unit of work is redispatched on the same physical CPU that it was last on
- Prior to HiperDispatch, PR/SM would split each logical CPU evenly based on its average share of a processor
  - SYSB gets 6 LPs, each effectively 50% of a physical (3 / 6)
  - SYSC gets 3 LPs, each effectively 70% of a physical (2.1 / 3)
  - SYSD gets 2 LPs, each effectively 45% of a physical (0.9 / 2)

Can lead to what's called "short CPs": Note SYSB has "shorter" CPs than SYSC!

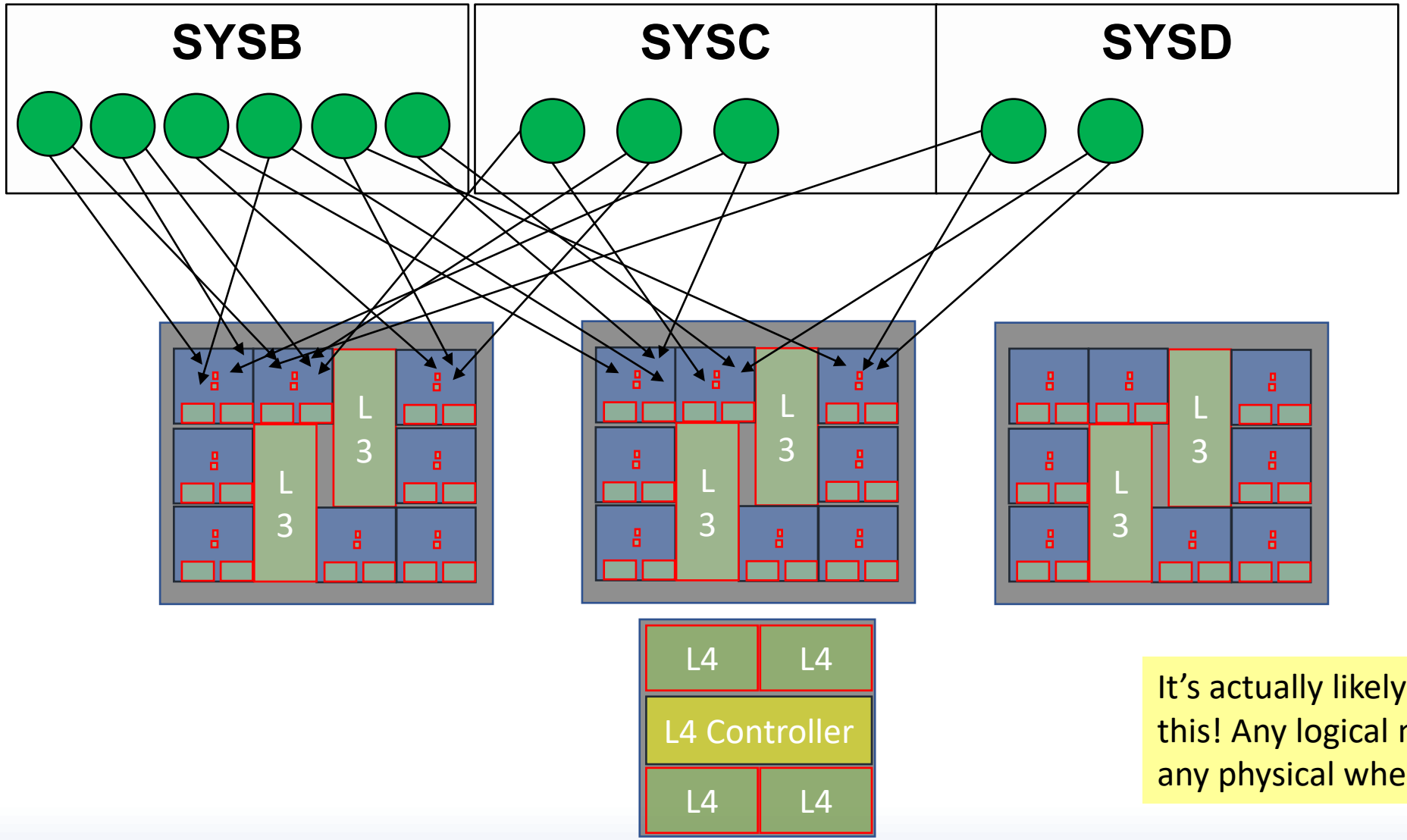
z/OS runs better with at least 2 LPs!



Vertical CP Management

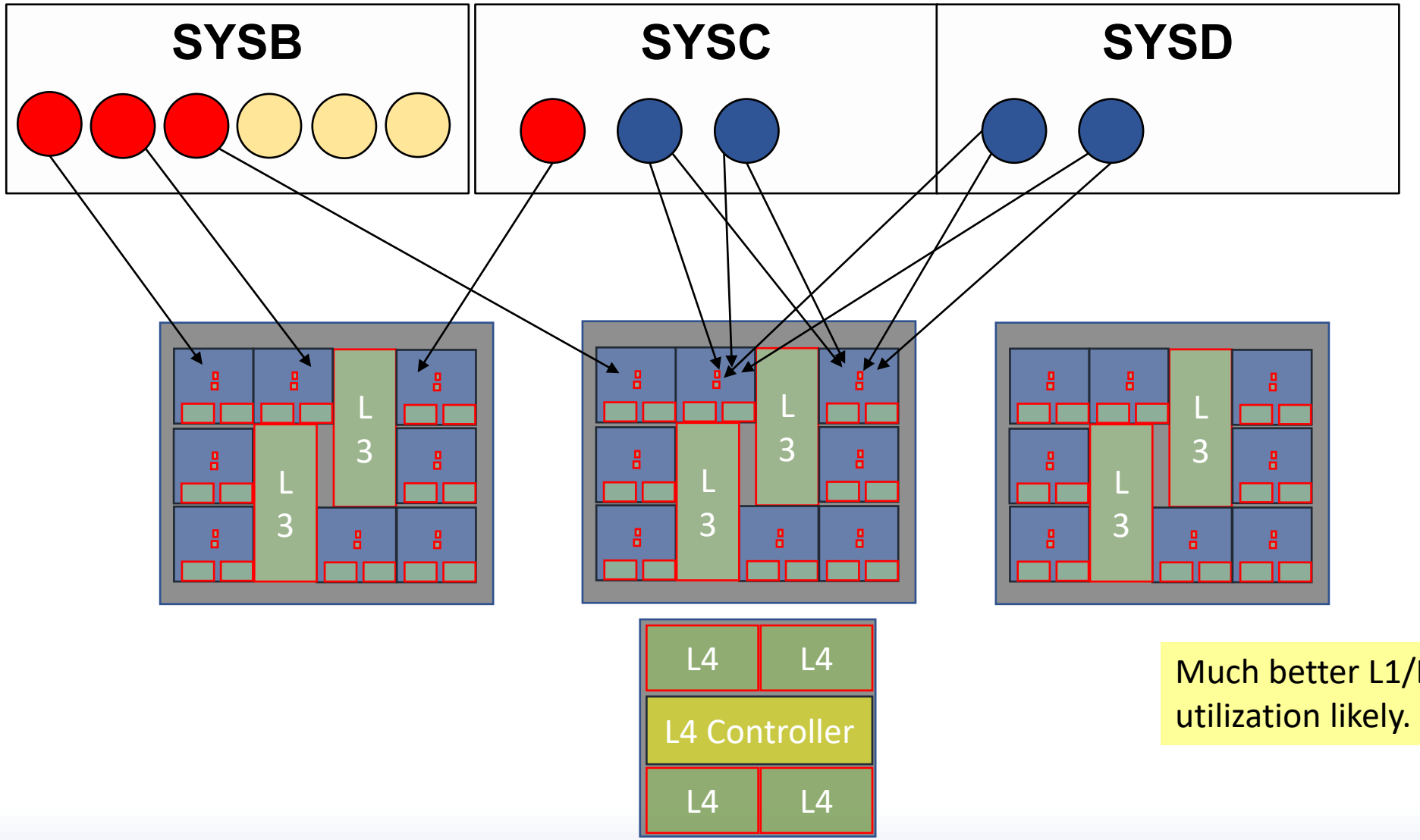
- HiperDispatch manages CPs “vertically”, meaning it endeavors to make the logical CPs a larger percentage of a physical
- Logical processors classified as:
  - High – The processor is essentially dedicated to the LPAR (100% share)
  - Medium – Share between 0% and 100%
  - Low – Unneeded to satisfy LPAR’s weight
- This processor classification is sometimes referred to as “vertical” or “polarity” or “pool”
  - E.G. Vertical High = VH = High Polarity = High Pool = HP
- Parked / Unparked
  - Initially, VL processors are “parked”: work is not dispatched to them
  - VL processors may become unparked (eligible for work) if there is demand and available capacity

HiperDispatch Off



It's actually likely worse than this! Any logical might end up on any physical when redispatched.

HiperDispatch On

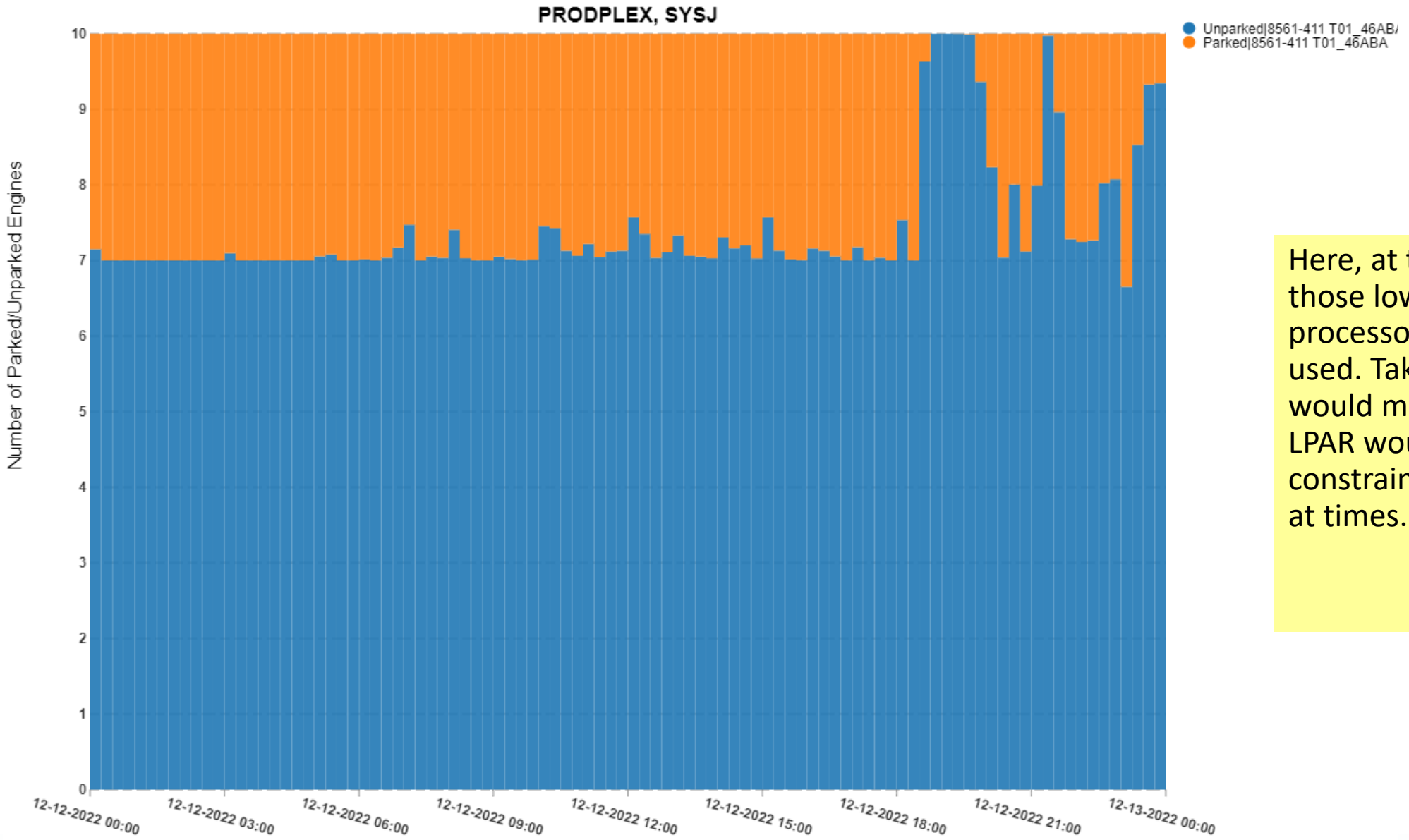


Much better L1/L2 cache utilization likely.

Should you limit low pool processors?

- IBM advice: don't have more than 2 low pool processors on an LPAR
  - I mostly agree with that, but maybe not in the way you might think
- 3 (or even more) parked low pool processors aren't causing any problem
- When a processor is unparked, its CPU efficiency will be quite low
  - This likely be relatively brief until the low pool CP "warms up"
  - In some cases, a low pool CP may be on the "wrong" book: this could lead to bad performance even after the CP "warms up"
  - In most cases, after a very brief warm-up, a busy low pool CP likely won't be substantially less efficient than a busy medium pool CP
- If a processor is being unparked, it's because it can be put to good use!
  - Don't take away low pool CPs that are being unparked and used just to meet some arbitrary rule about having too many low pool processors!

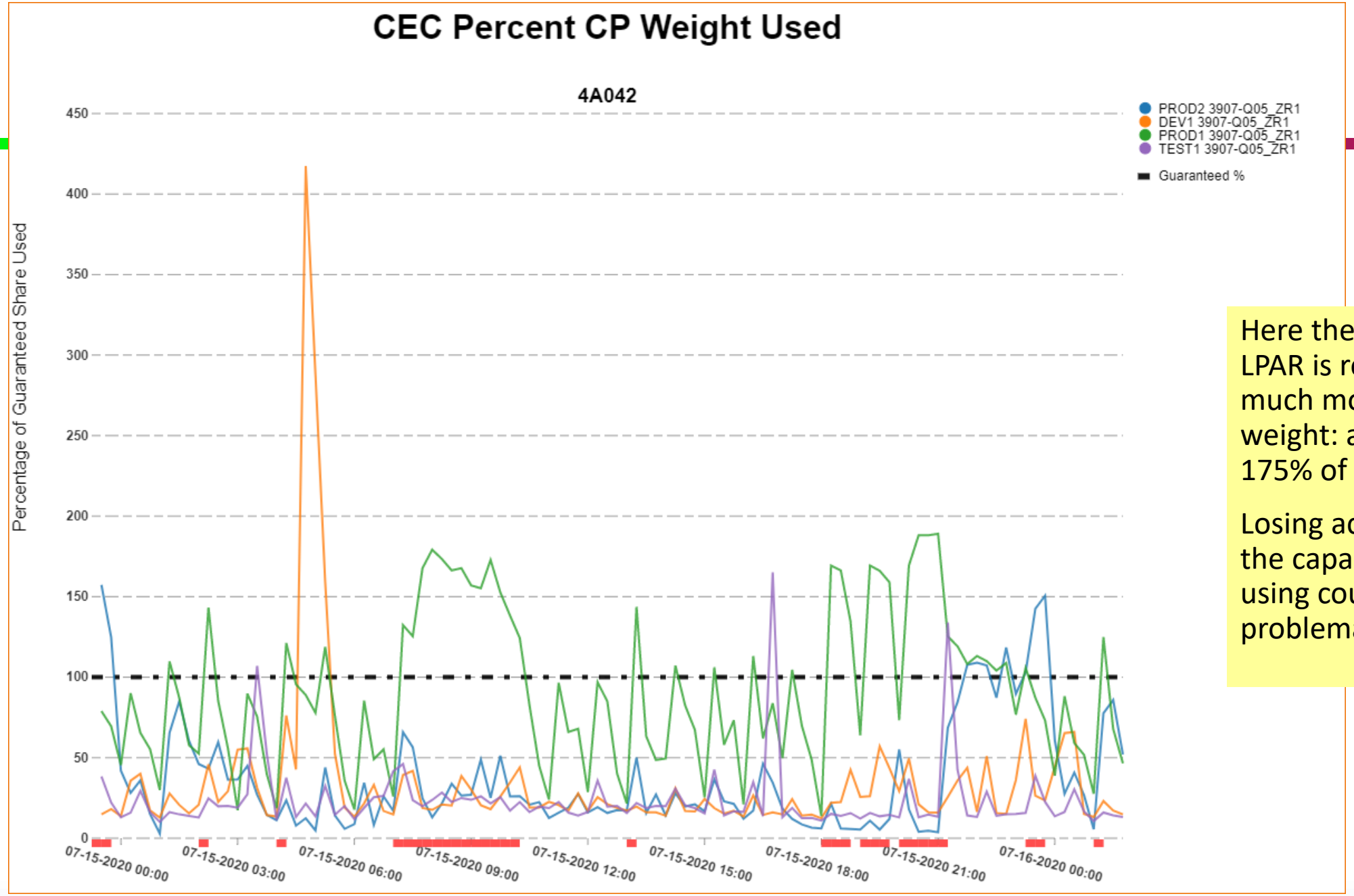
HiperDispatch - Parked / Unparked CPs



Here, at times, all 3 of those low pool processors are being used. Taking one away would mean that this LPAR would be more constrained for capacity at times.

So why do I mostly agree with the advice?

- The real issue with having too many low pool processors is that that implies that your weights are not matching your actual workloads
  - If you're unparking low pool CPs to get work done, that means that the LPAR is borrowing weight (capacity) from other LPARs
  - The borrower may lose access to that capacity if the other LPARs get busy, which is a risk point that should be addressed
  - The LPAR could potentially have more high pool processors, which generally are expected to out-perform medium and low pool processors
- So ideally… adjust the weights to reflect what the LPARs need
  - Most especially for production LPARs!
  - BCPii can be used to change weights dynamically via REXX scripts
  - Consider changing weights before batch processing or other known events



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

Losing access to ~40% of the capacity that it's using could be problematic!

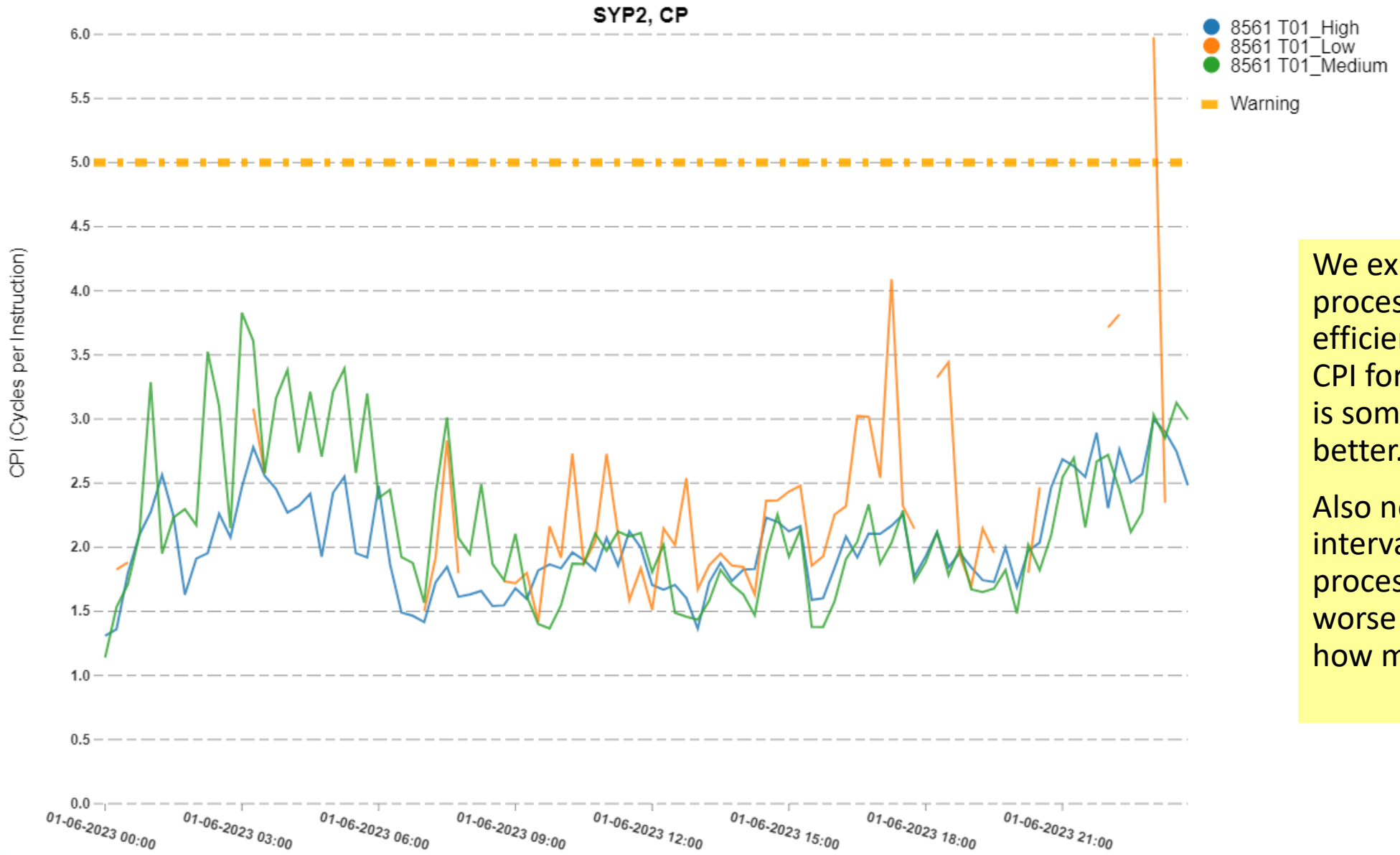
Interesting Measurements

Some SMF 113 Measurements

- CPI – Cycles Per Instruction
  - Lower is better, can approach 1-2 on latest processors in best case scenario
  - A primary measure of processor “efficiency”, but also impacted by instruction mix
- L1MP – Level 1 Misses Per 100 Instructions
  - How often does the processor have to go beyond the level 1 cache to get data
  - More cycles taken to access data further out in the cache hierarchy
  - Lower is better, often around 2-4 range on modern processors
  - Lower L1MP = Lower CPI
- RNI – Relative Nest Intensity
  - Relative measure of how intensively the workload is using the cache hierarchy
  - Less a measure of “efficiency” than a characterization of the workload
  - Although decreasing cache contention can lower it
  - Lower is “better”, generally is < 1

CPI for System by Engine Type and Polarity

SMF 113

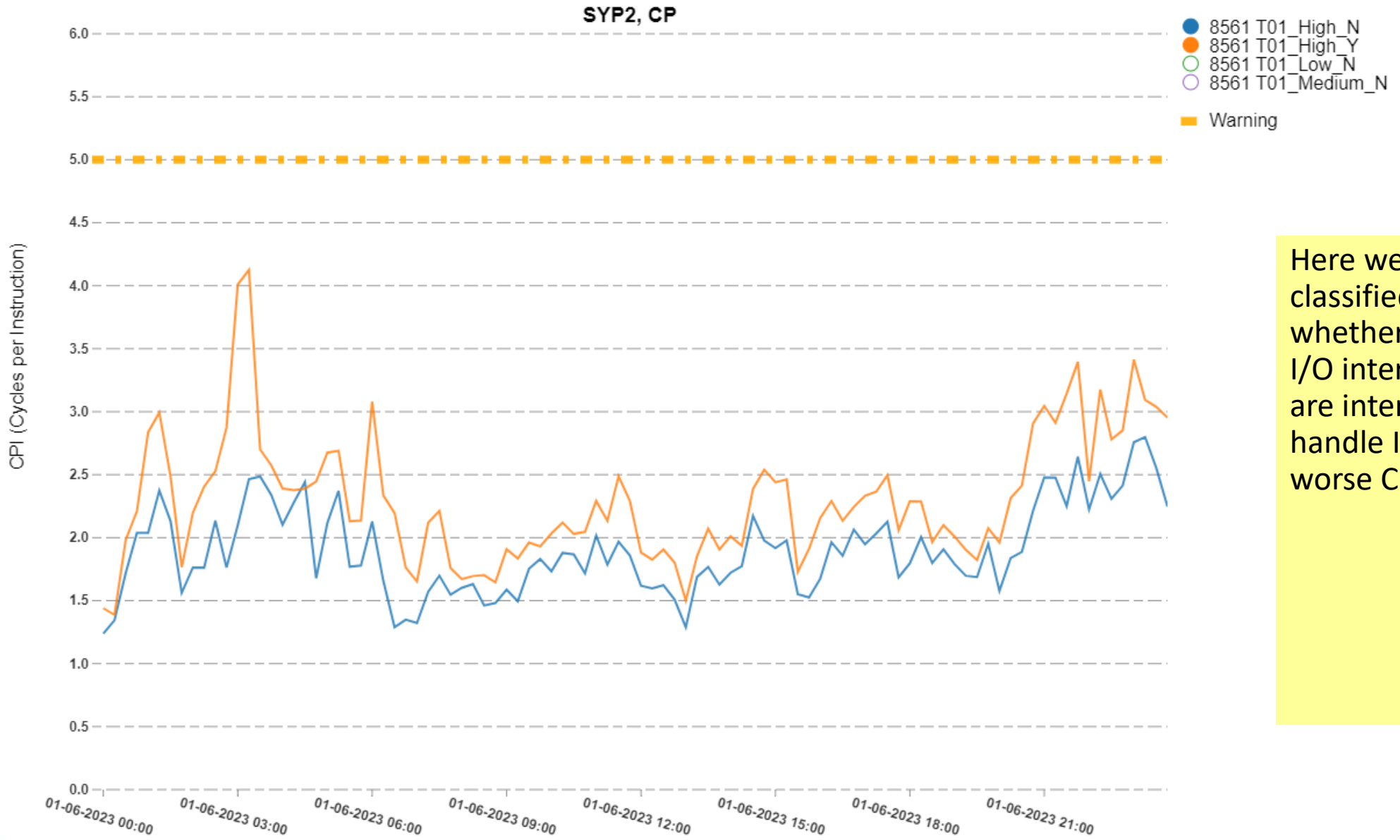


We expect to High pool processors to be more efficient, but here the CPI for Medium pool CPs is sometimes slightly better.

Also note that in many intervals, Low pool processors not much worse (but depends on how much they're used).

CPI for System by Engine Type, Polarity, and I/O Interrupts

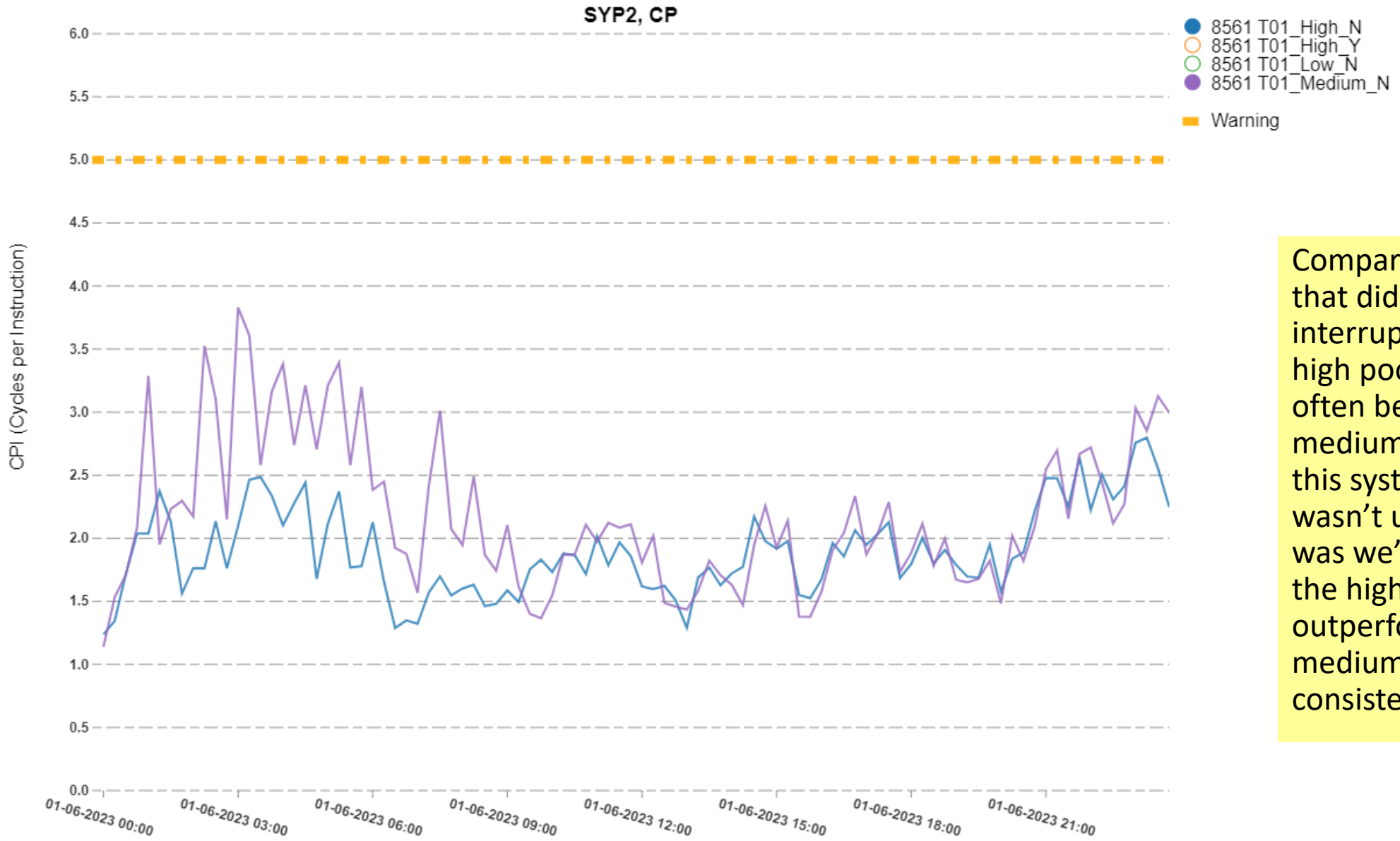
SMF 113



Here we've further classified the CPs by whether they handled I/O interrupts. CPs that are interrupted to handle I/O tend to have worse CPIs.

CPI for System by Engine Type, Polarity, and I/O Interrupts

SMF 113



Comparing just the CPs that didn't handle interrupts shows the high pool CPs to be more often better than the mediums. But note that this system generally wasn't under stress, if it was we'd expect to see the high pool processors outperform the mediums more consistently.

Summary / Recommendations

- High pool processors generally most efficient
- Low pool processors perform same as medium pool when actively used
- Large number of Low pool processors imply a possible weight issue
- Adjust weights to make sure you don't have a risk of important LPARs losing access to capacity that they need
- Record SMF 113 data
  - And sync it to SMF! (SMFINTVAL=SYNC on the modify command to start HIS recording)
- Doing I/O clearly affects processor efficiency
  - The only good I/O is still no I/O!

Questions??