# Per CPU Utilizations

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 (why you're here!)

When CPU utilizations are reported and analyzed, it is most common that the utilizations reported are for the pool of processors configured to a machine or to an LPAR. Rather than reporting the utilization of each processor, usually we just look at the average across all the online processors. Why is this? Is there any value to analyzing the utilizations of each individual processor? How are the measurements for an individual processor affected by HiperDispatch? During this presentation, Scott Chapman will explore reporting and analyzing individual CPU measurements. It will be interesting to see how the measurements for individual processors differ from reporting the utilization of the pool of processors.

# 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

# 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
  - September 16-20, 2024
- 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!

**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.

# Agenda

- A brief reminder of HiperDispatch
- Looking at processor utilization:
  - How busy is the machine and what do we mean by that?
  - A look at utilization at shorter timeframes
  - A look at utilization by processor
  - Why do we see the patterns we see?
  - Do we care about utilization by processor?

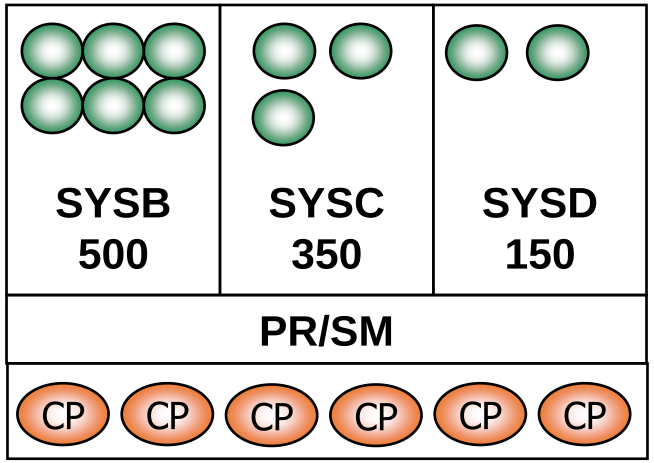
# HiperDispatch Reminder

# 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"
  - Weights assigned on the HMC by type of processor (GP, zIIP, ICF, IFL)
  - 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).

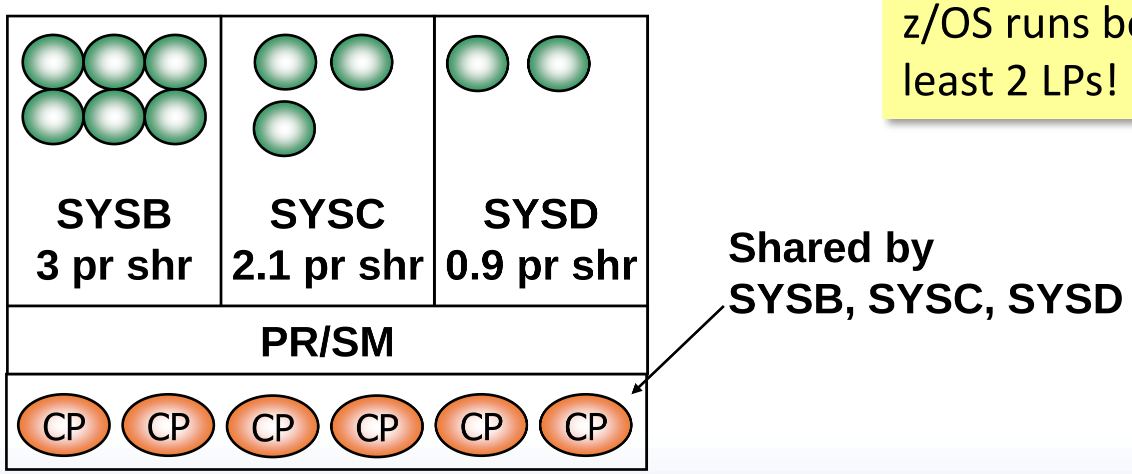
**Physical CPs shared by SYSB, SYSC, SYSD**

# 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!

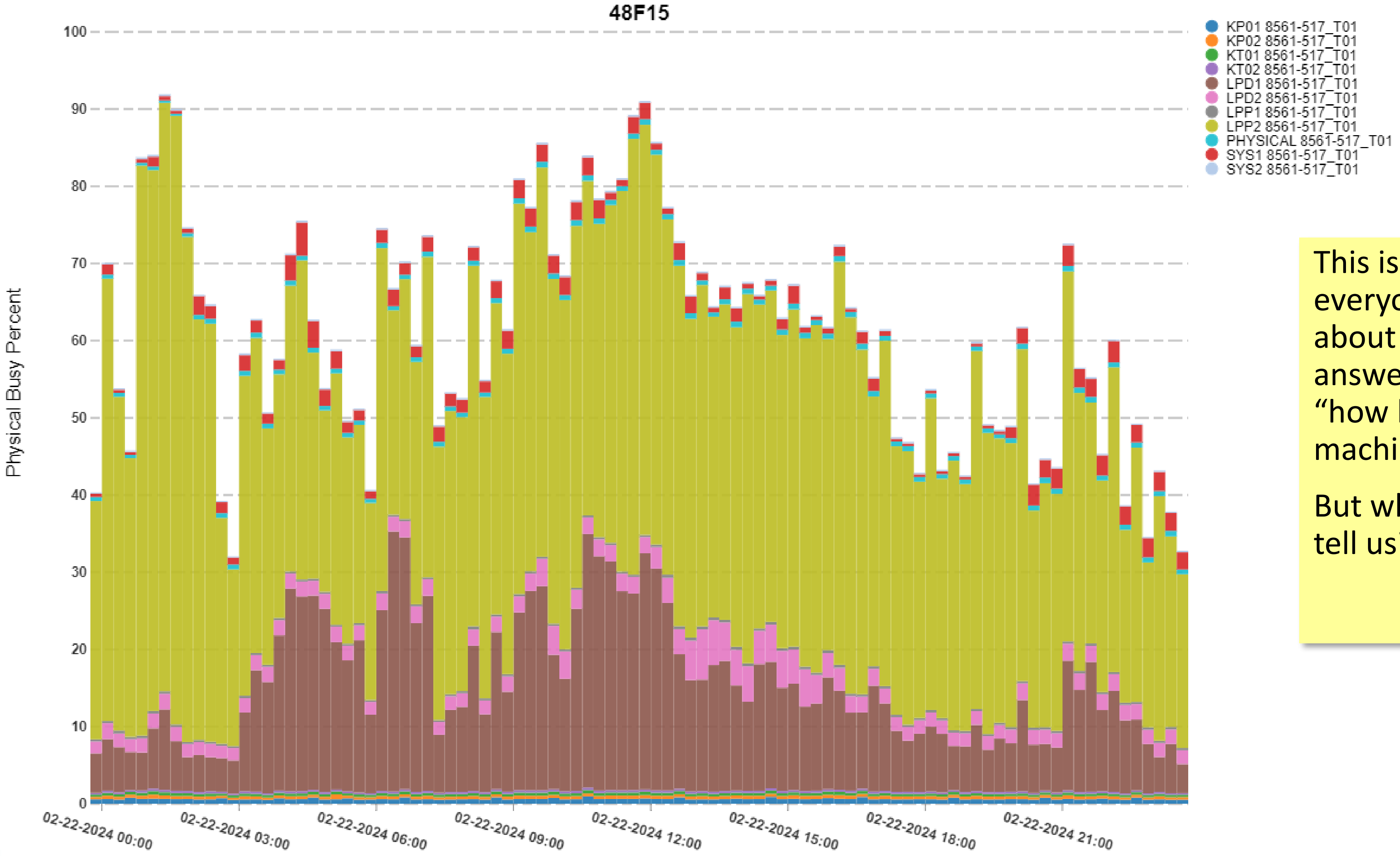


# Vertical CP Management

- 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% (often 50-100% unless small LPAR)
  - 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

# Looking at processor utilization

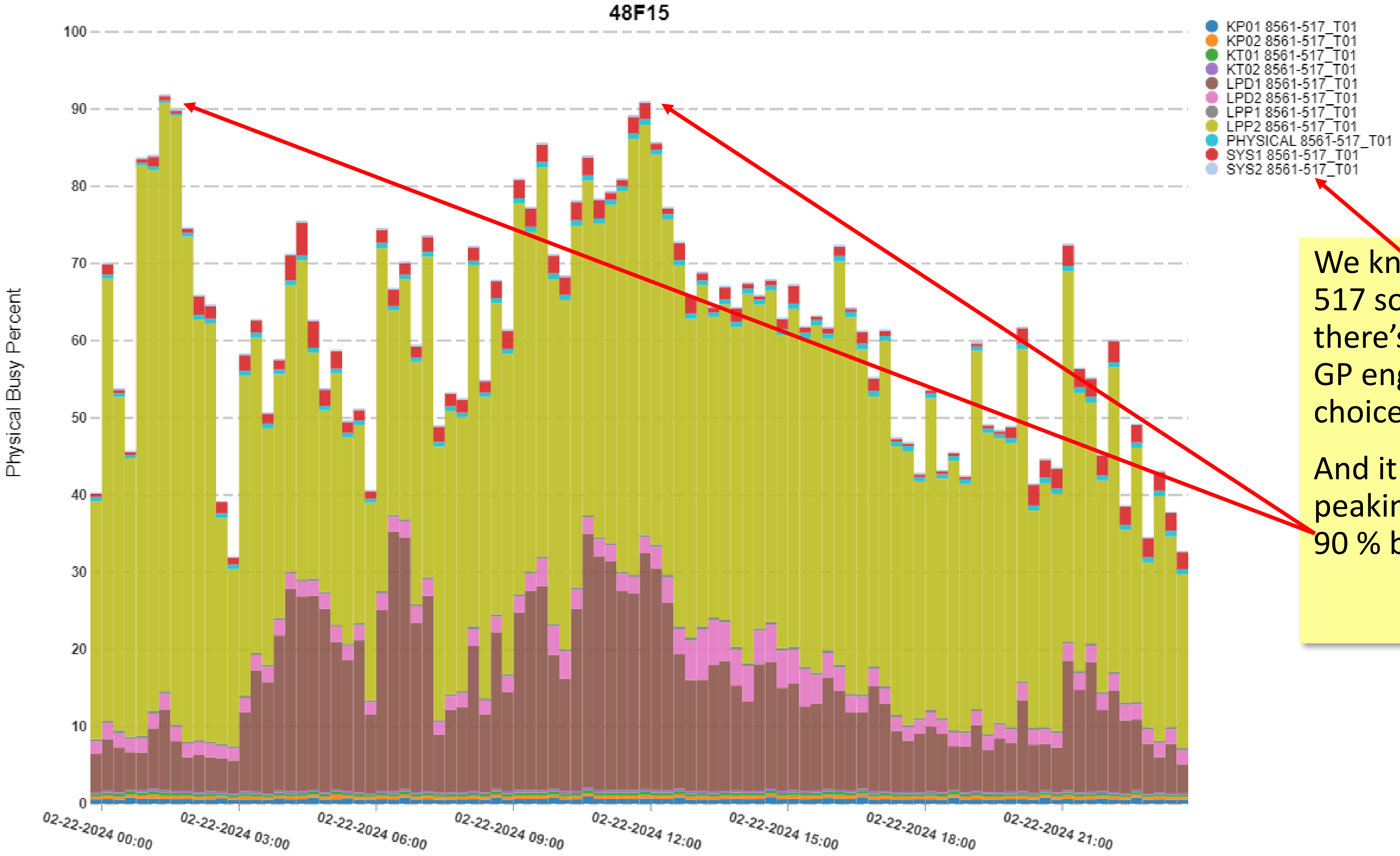
# CEC Physical Machine CP Busy% by CEC Serial Number




This is the chart everyone cares most about because it answers the question "how busy is my machine?"

But what does it really tell us?

# CEC Physical Machine CP Busy% by CEC Serial Number



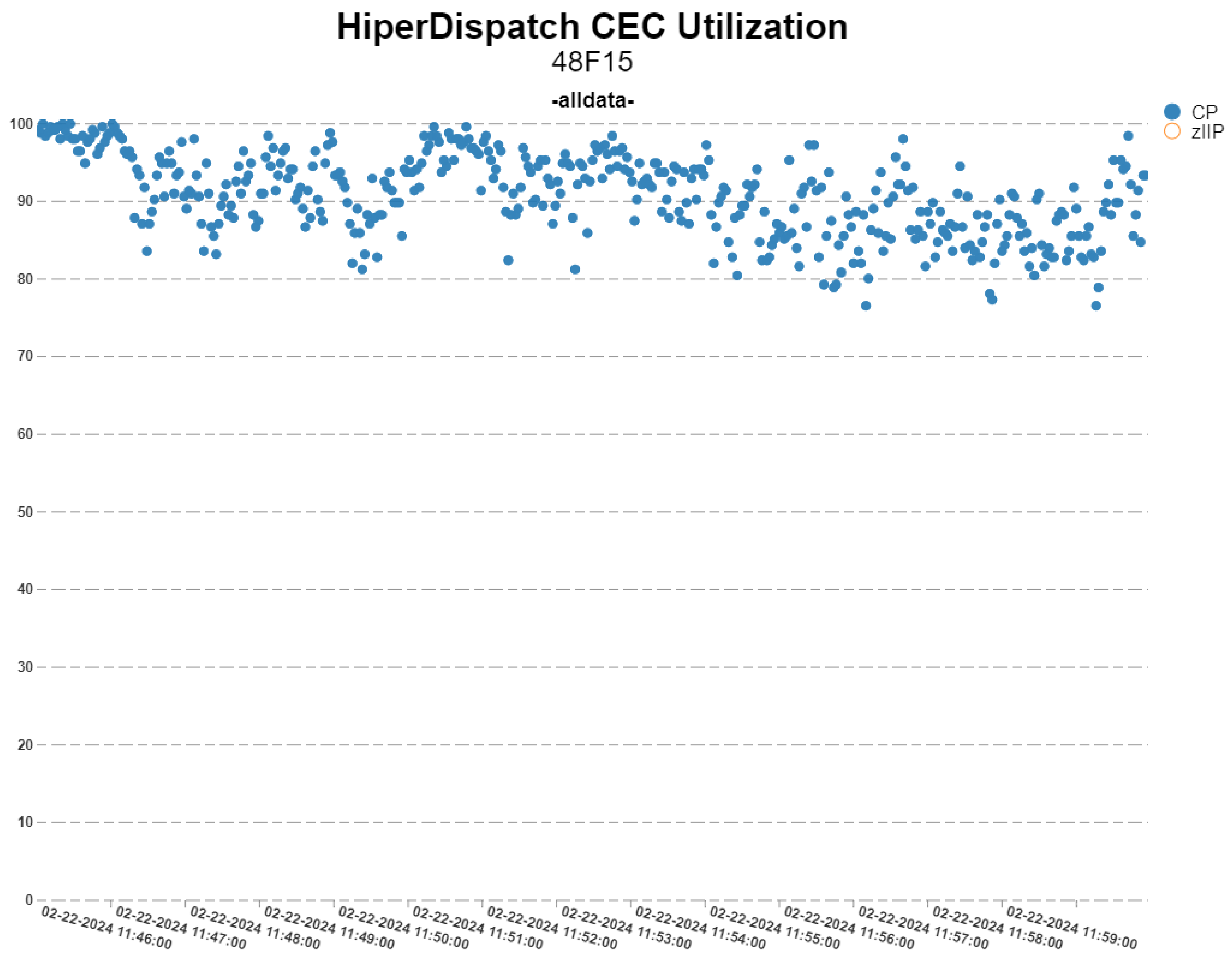

We know this is a z15 517 so we know that there's 17 sub-capacity GP engines. (Good choice!)

And it looks like it's peaking out at just over 90 % busy.

# What does the 517 is 90% busy *mean*?

- Effectively that's an average utilization of the 17 GP engines over the course of the 15 minute (900 second) interval
  - So averaged over space (engines) and time (seconds)
- Really, it's total CPU time / CPUs * interval
  - E.G. 13770 / (900 * 17) = 0.9 = 90%

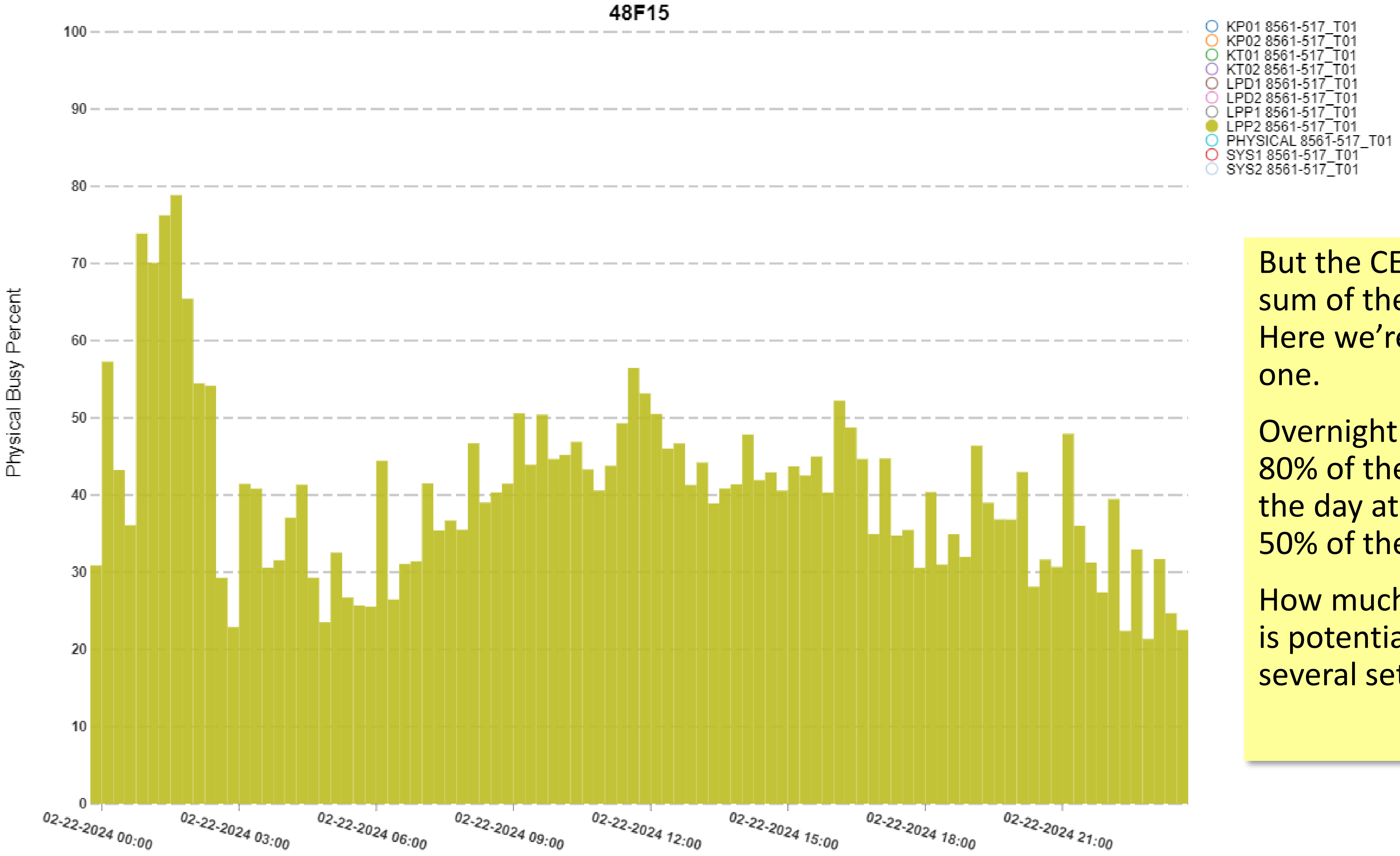
- Important notes:
  - At any given moment a CPU is either being used (CPU time) or is not being used
  - Averages can hide peaks within the interval




Here's that peak 15 minute interval that was showing just over 90% busy, but with observations every 2 seconds.

You see how the average was ~90%, but there were a few minutes where the utilization was more like 99%+.

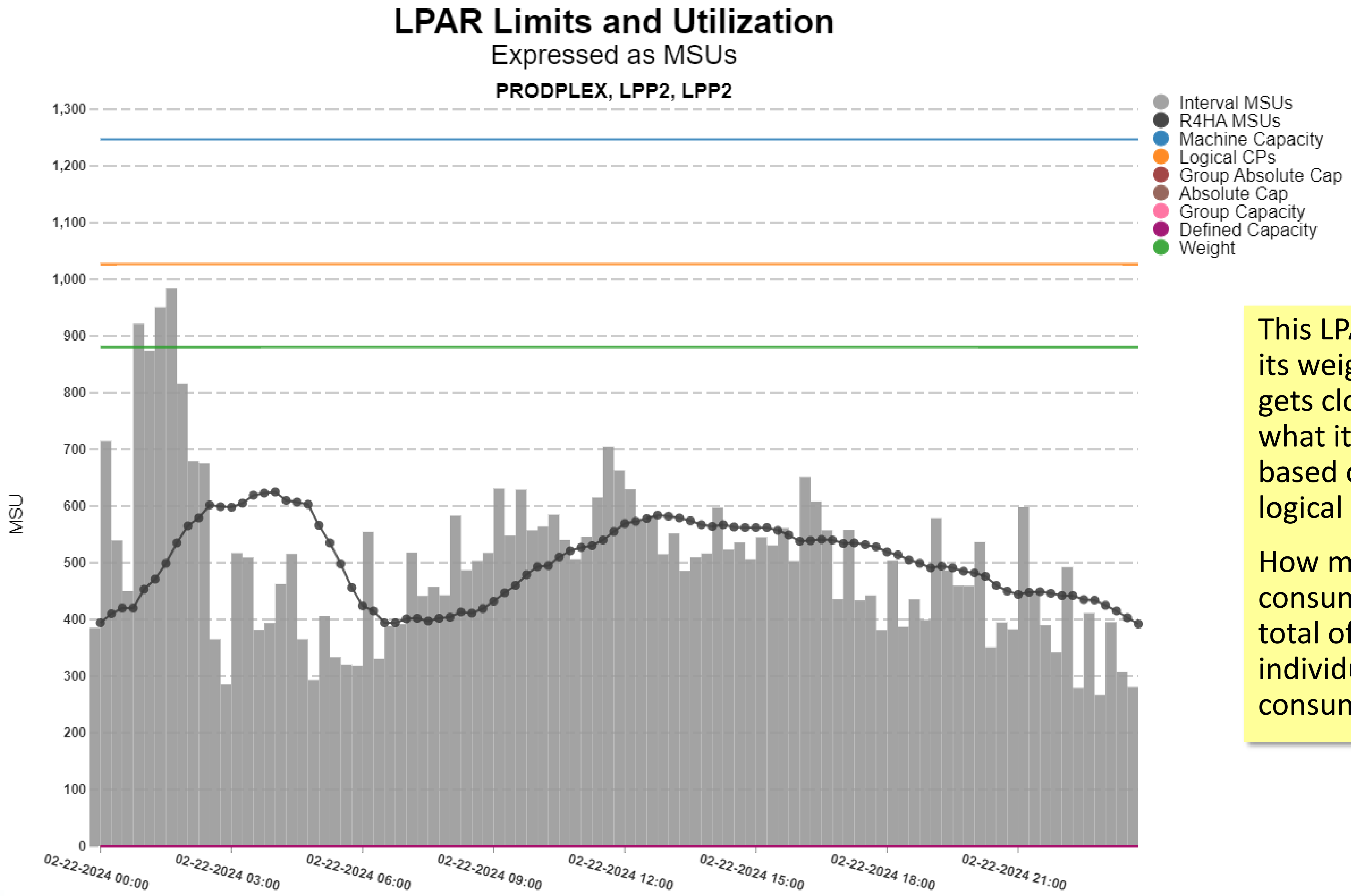
# CEC Physical Machine CP Busy% by CEC Serial Number




But the CEC utilization is the sum of the LPAR utilizations. Here we're focusing on just one.

Overnight it peaks at almost 80% of the CEC, and during the day at times uses around 50% of the CEC.

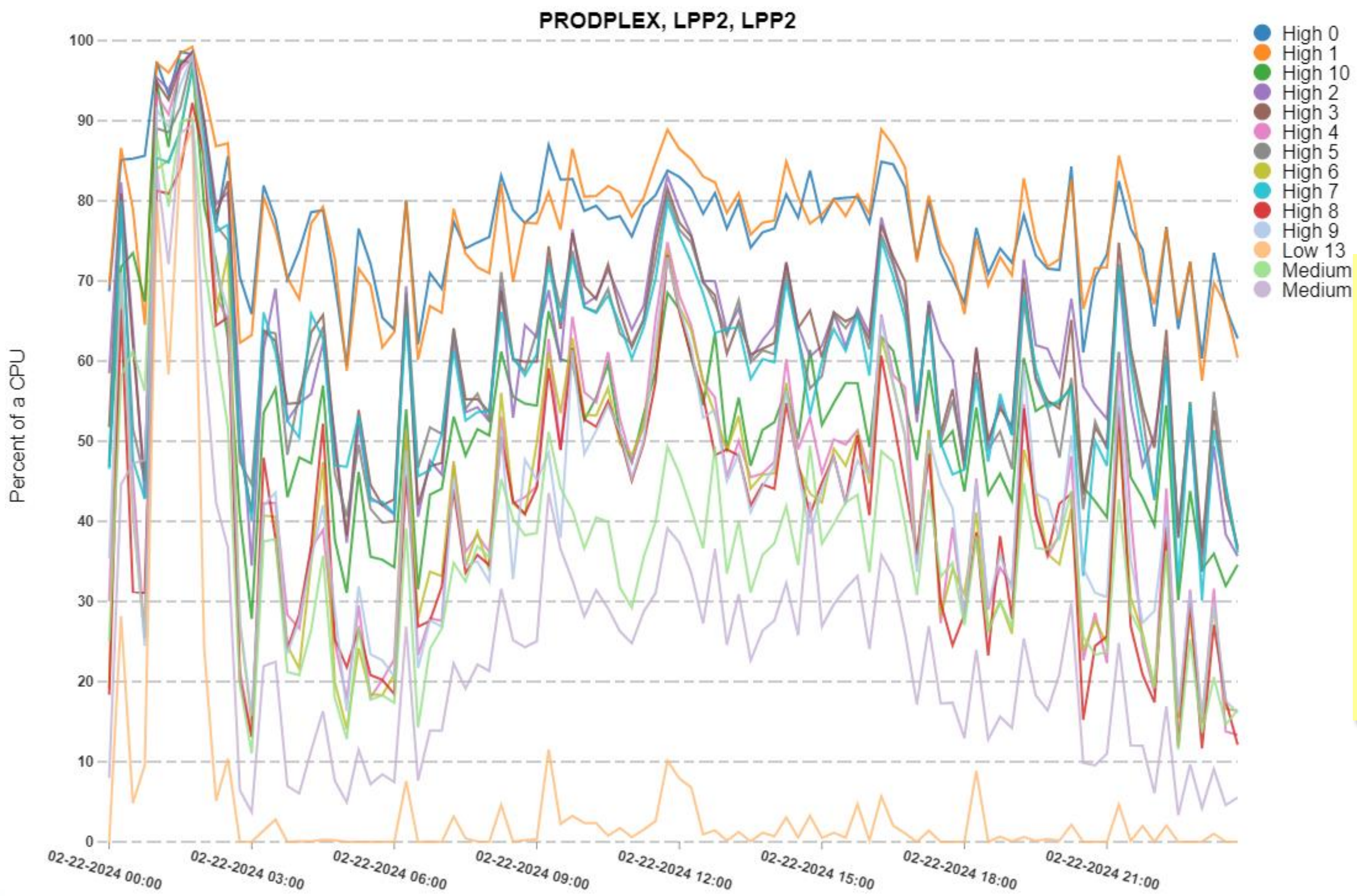
How much the LPAR can use is potentially limited by several settings.




This LPAR does exceed its weight overnight and gets close to the limit of what it could consume based on the number of logical CPs it has.

How much the LPAR consumes is really the total of how much each individual logical CP consumes.

# LPAR Per-CPU CP Busy%

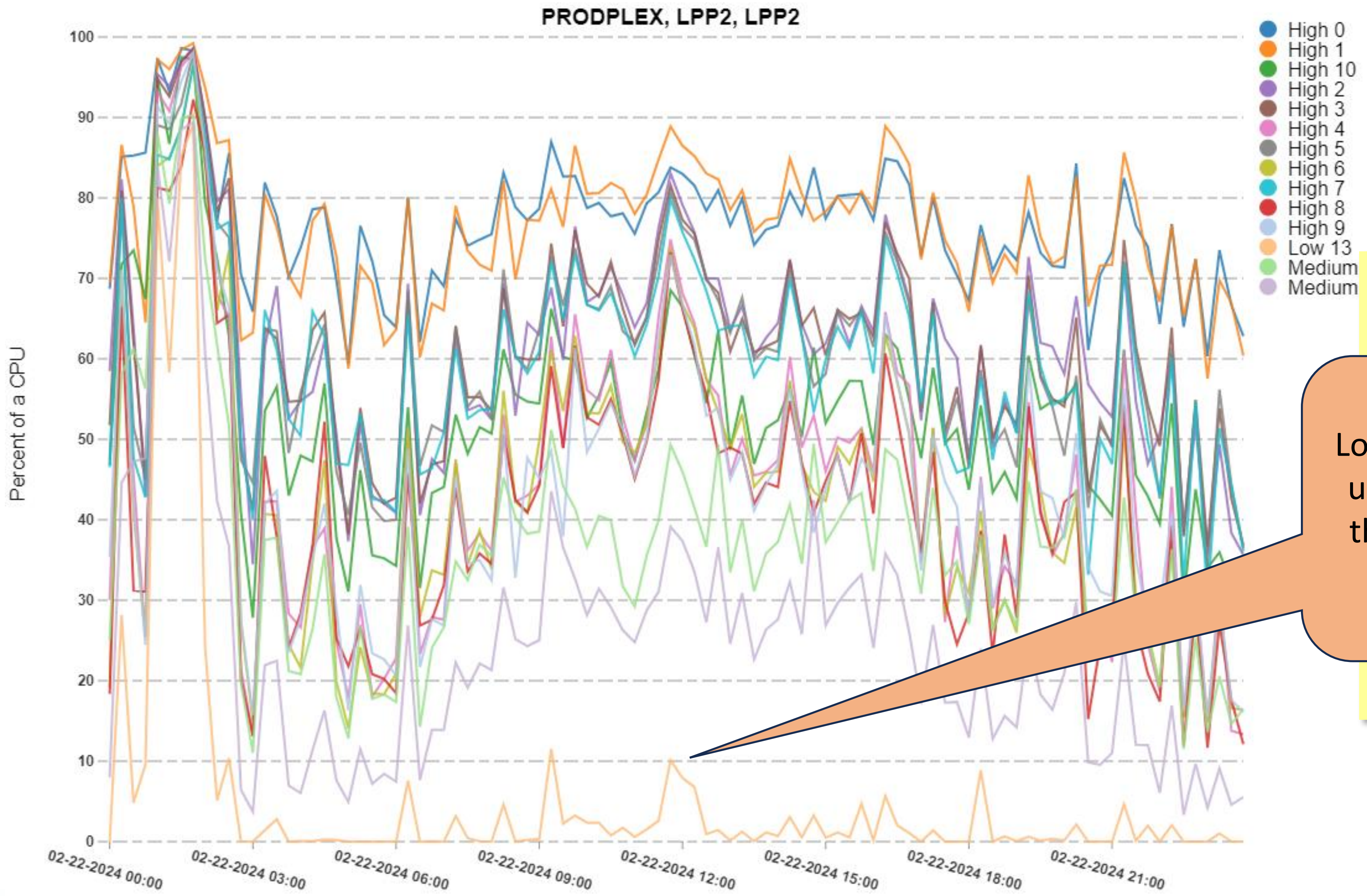



This chart shows how busy each GP CPU was on the LPAR.

Note that there seems to be bands of processor utilizations when the LPAR isn't trying to consume all its possible capacity.

Is this surprising?

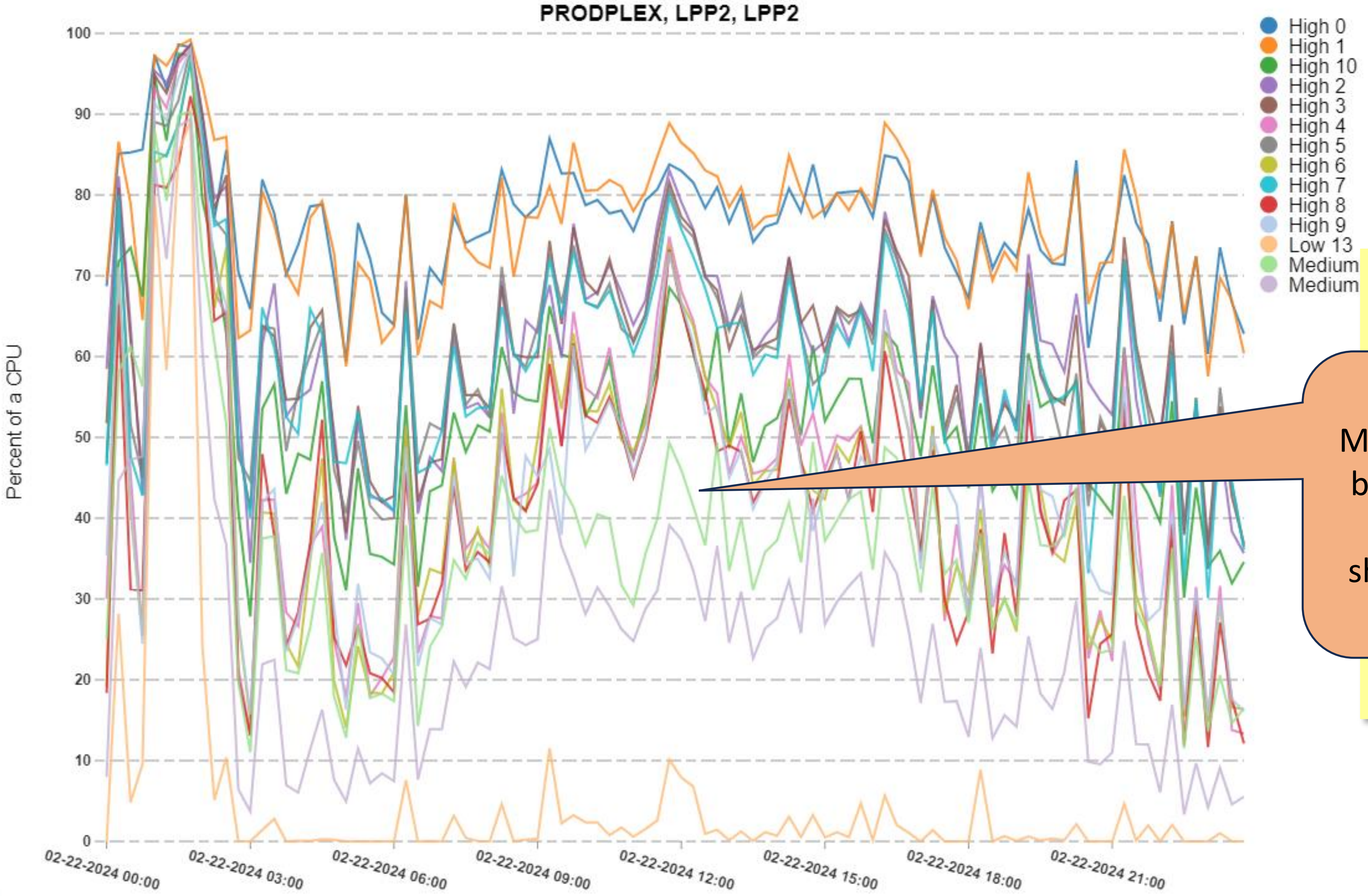
# LPAR Per-CPU CP Busy%




This pattern of utilization is not at all surprising!

Low pool CPs will naturally use less, especially when they're not unparked for the entire interval.

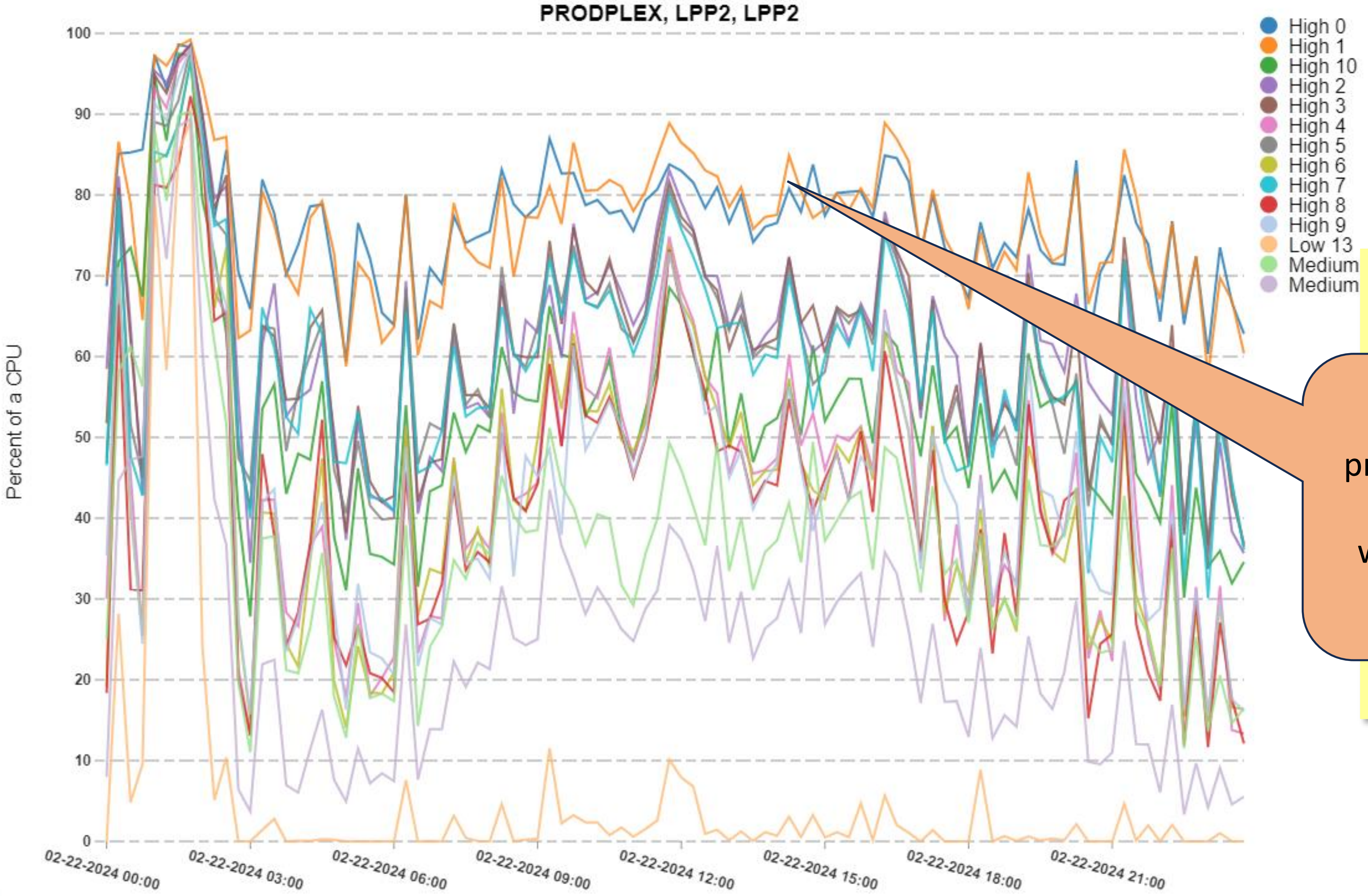
# LPAR Per-CPU CP Busy%




This pattern of utilization is not at all surprising!

Medium pool CPs will also be expected to consume less because they're shared with other LPARs.

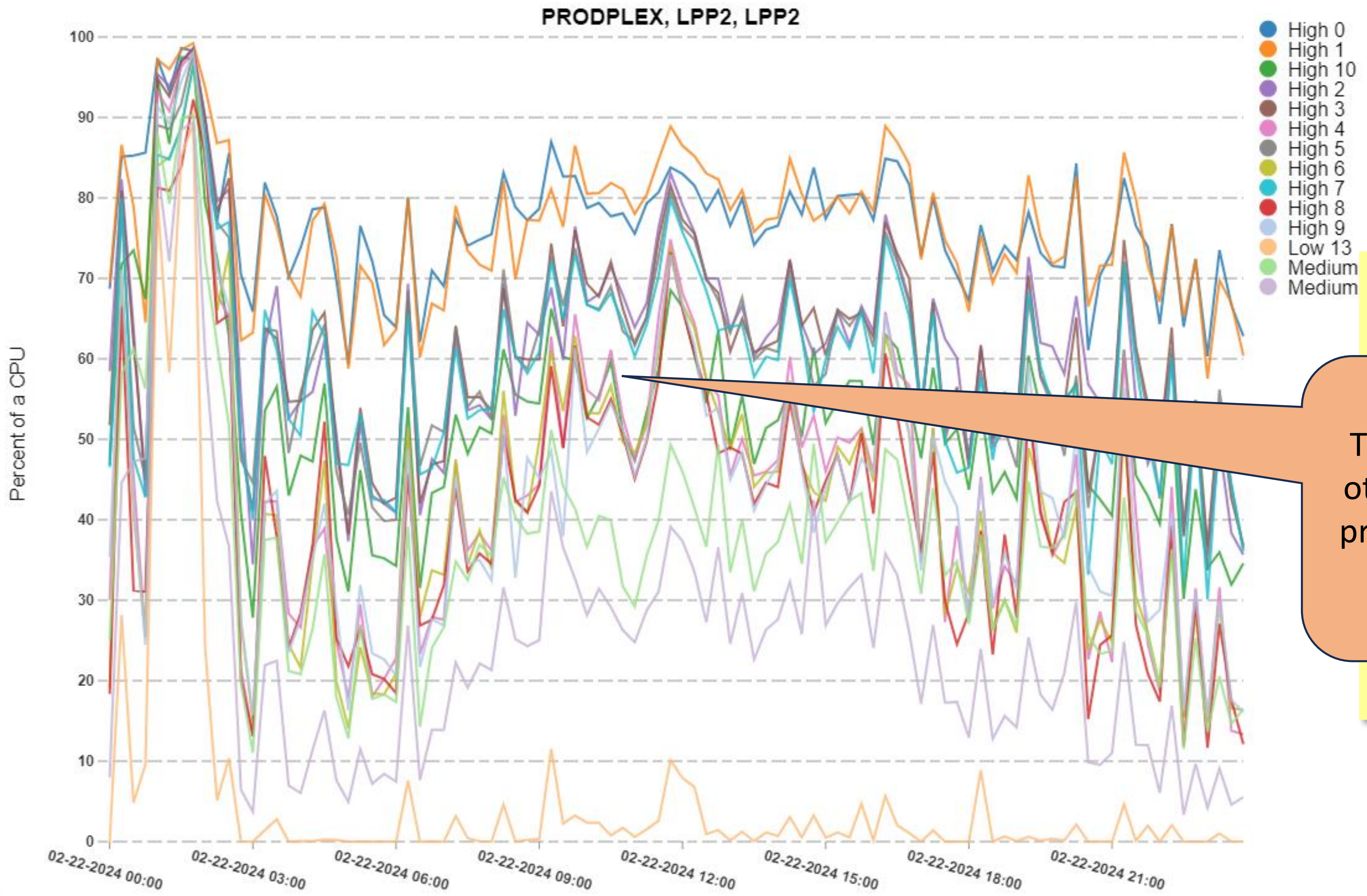
# LPAR Per-CPU CP Busy%




This pattern of utilization is not at all surprising!

These high pool processors were handling I/O interrupts so that would explain why they were using more.

# LPAR Per-CPU CP Busy%




This pattern of utilization is not at all surprising!

The fact that there's two other groups of high pool processors is explained by affinity nodes.

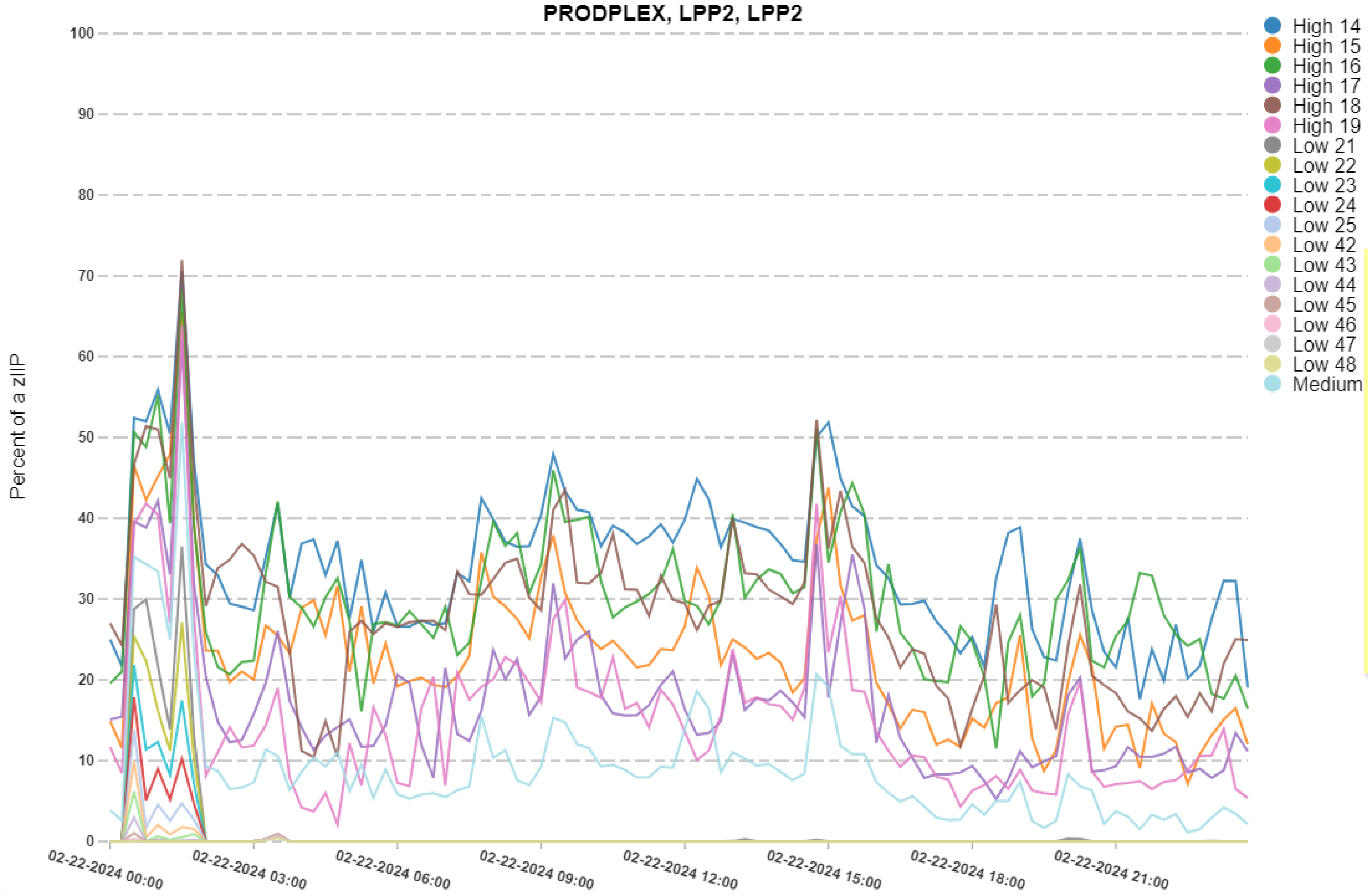
# z/OS Dispatcher Affinity Nodes

- System creates nodes of logical processors
  - Originally said to be “ideally 4 high-pool processors”
  - But on recent machines, 2-3 high pool processors seems quite common
    - This makes more sense to me!
  - May have many low pool processors in one node
- Each node gets its own queue
  - Work units assigned to a particular node
  - Separate high performance work unit queue for SYSSTC/SYSTEM SRBs crosses nodes
- Nodes have list of helper nodes
  - Node needs help when it can’t run all the work assigned to it
    - Low pool processor in the node used before signaling another node
  - “Needs help” frequency controlled in part by CCCAWMT and ZIIPAWMT in IEAOPTxx

# PR/SM Affinity

- PR/SM also enforces affinity
  - High Pool logical CPs have very strong affinity to a particular physical CP
  - Mediums will try to stay in the same area in the nest (especially at book level)
  - Low pool CPs have little affinity as their capacity is not guaranteed by their weight
- See “The Highs and Lows: How Does Hyperdispatch Really Impact CPU Efficiency?” at https://www.pivotor.com/content.html
  - While tweaking weights to convert 1 medium to 1 high probably won’t have a significant impact, choosing more/slower CPs so you have a number of high pool processors instead of all mediums can be significant

# LPAR Per-CPU zIIP Busy%




Here's the zIIPs on the earlier system. Less obvious bands.

Turns out that the 6 high pool zIIPs were assigned to 3 affinity nodes of 2 highs each. (One node also had the medium and all lows.)

# Affinity Nodes Makeup


Here's what those affinity nodes looked lie for that system.

First for each CP type has the medium and lows as well as 1-2 highs. The remainder have 2-3 highs.

Note: LPARs seem to need at least 3 high pool processors to get more than a single affinity node (per CPU type).

# Summary: How much do we care?

- Other than as an interesting academic discussion: not much
  - I always think it's useful to understand how things are working at a fairly low level
  - Having these details in your mental model of how things work can help you understand other measurements
    - E.G. Why do I still sometimes see CPU delay samples for high-importance workloads when the machine is not busy?
  - Does show another reason why more/slower with more high pool CPs can be good
- There's no externalization of what workload is assigned to what affinity node
  - And workloads may shift between affinity nodes
- Only tuning opportunity is ZIIPAWMT/CCCAWMT
  - Tuning ZIIPAWMT to avoid crossover makes some sense
  - Trying to tune CCCAWMT for some useful outcome seems… questionable
- Knowing one affinity node or CPU is more or less busy than the others doesn't really highlight any tuning opportunities

# Questions?