Understanding and Comparing z/Architecture Processor Topologies

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!)

Mainframe processor design has evolved over the various generations of machines. In this webinar Scott will explore how the core of the mainframe has evolved over the past several generations, with a particular emphasis on how processor and cache designs influence both the performance and capacity of modern mainframes. Understanding these impacts can be useful for understanding why a workload might over- or under- perform on a new machine.

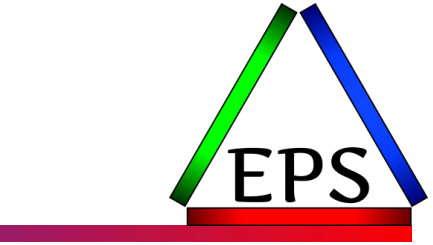
After discussing the physical designs, Scott will talk about the relationship between logical (what an LPAR sees) and the physical (the actual hardware) processors. That relationship can also impact performance and the effective capacity of the machine. Of course, there will be a discussion about the measurements used to help understand how efficiently your systems are utilizing the hardware.

Whew, that sounds like a lot, will it all fit in half an hour? Probably not: expect this to be closer to an hour than a half hour. But it will be fun to geek out over processor details!

Agenda

- History
- Processor Design
- Logical Processors
- Measurements and Comparisons

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
    https://www.pivotor.com/webinar.html

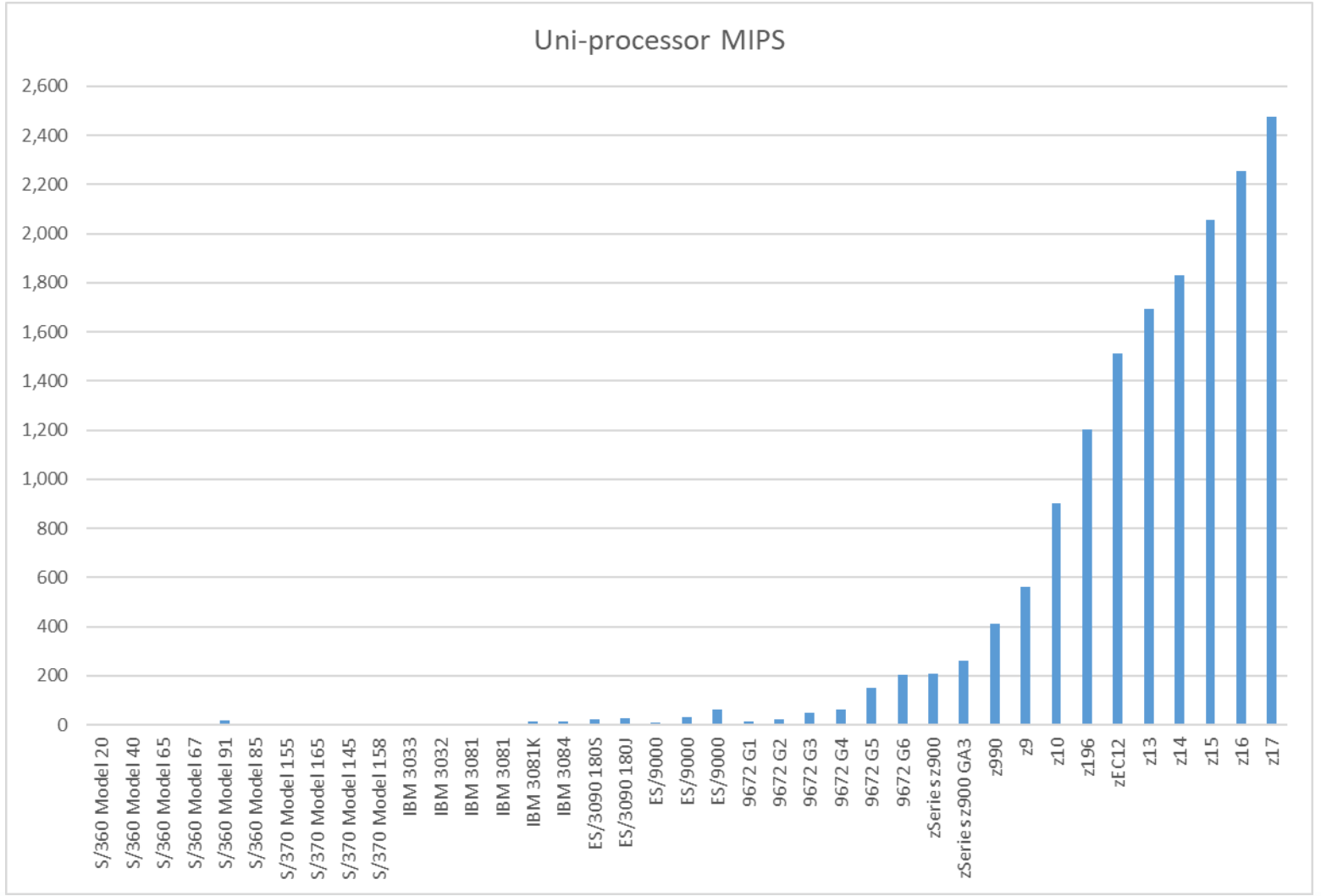
z/OS Performance workshops available

During these workshops you will be analyzing your own data!

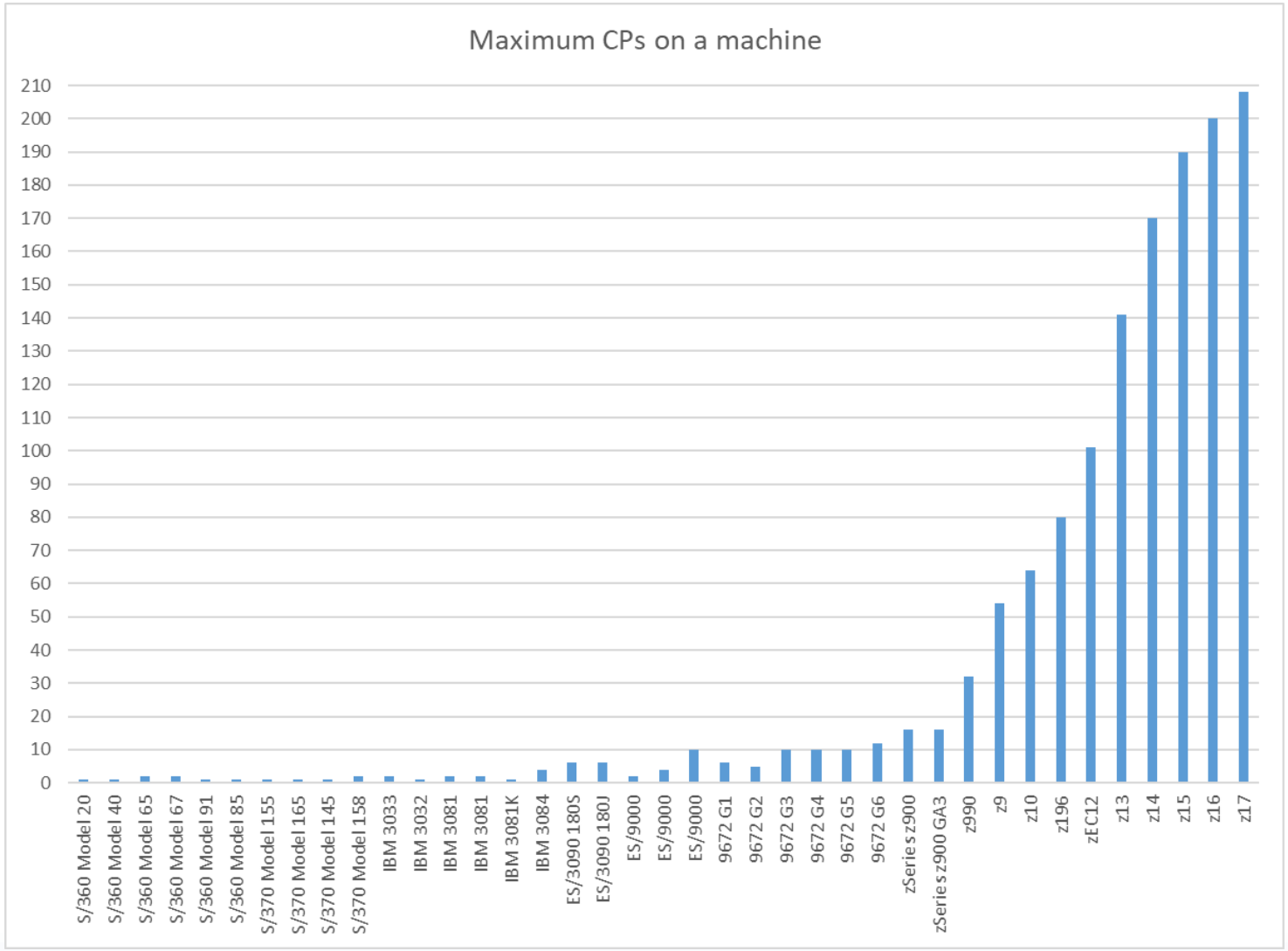
- WLM Performance and Re-evaluating Goals
  - May 12 – May 16, 2025 (4 days)
- Parallel Sysplex and z/OS Performance Tuning
  - October 21-22, 2025 (2 days)
- Essential z/OS Performance Tuning
  - September 22-26, 2025 (4 days)
- Also… please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

We have history

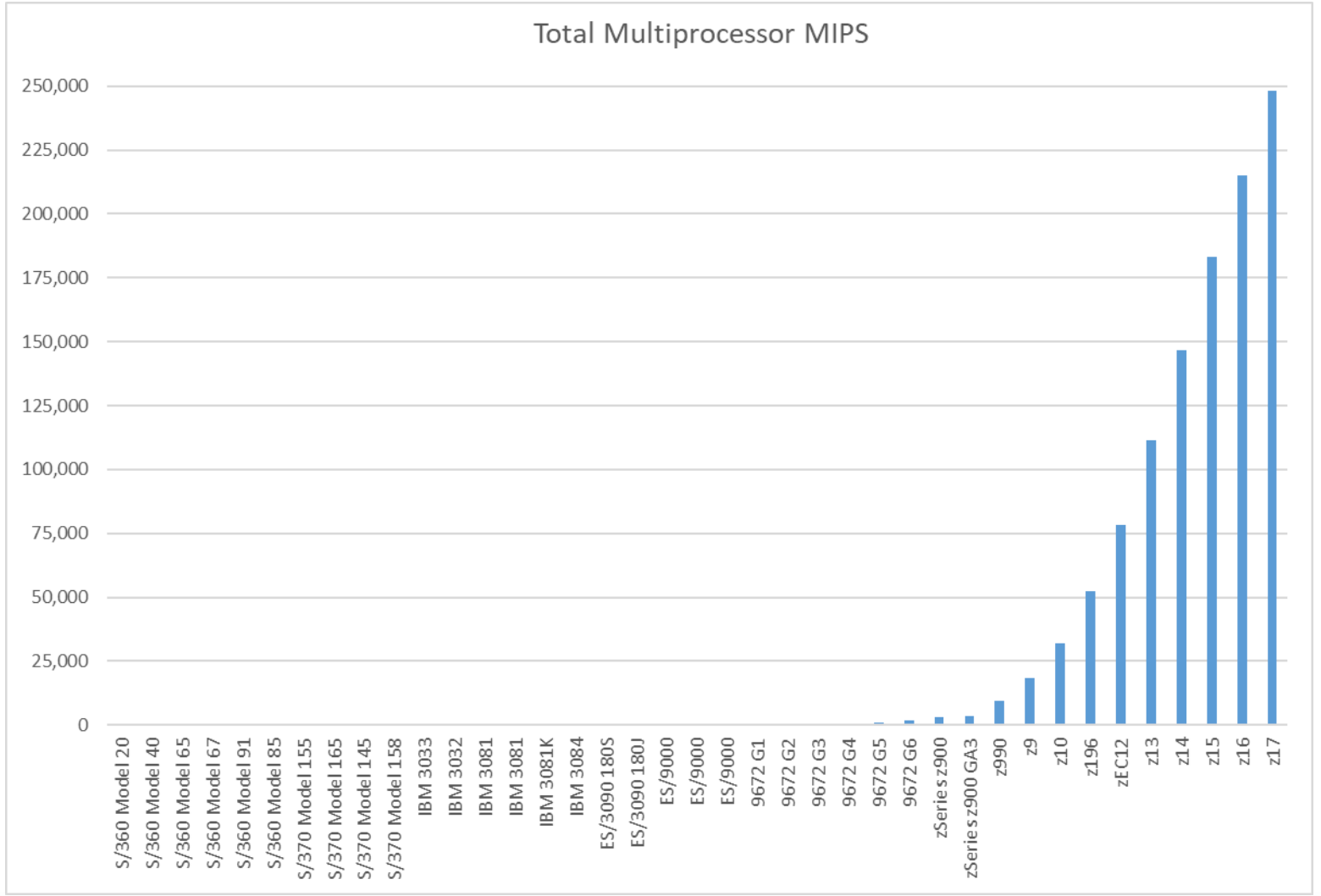
Historic Uni-processor capacity



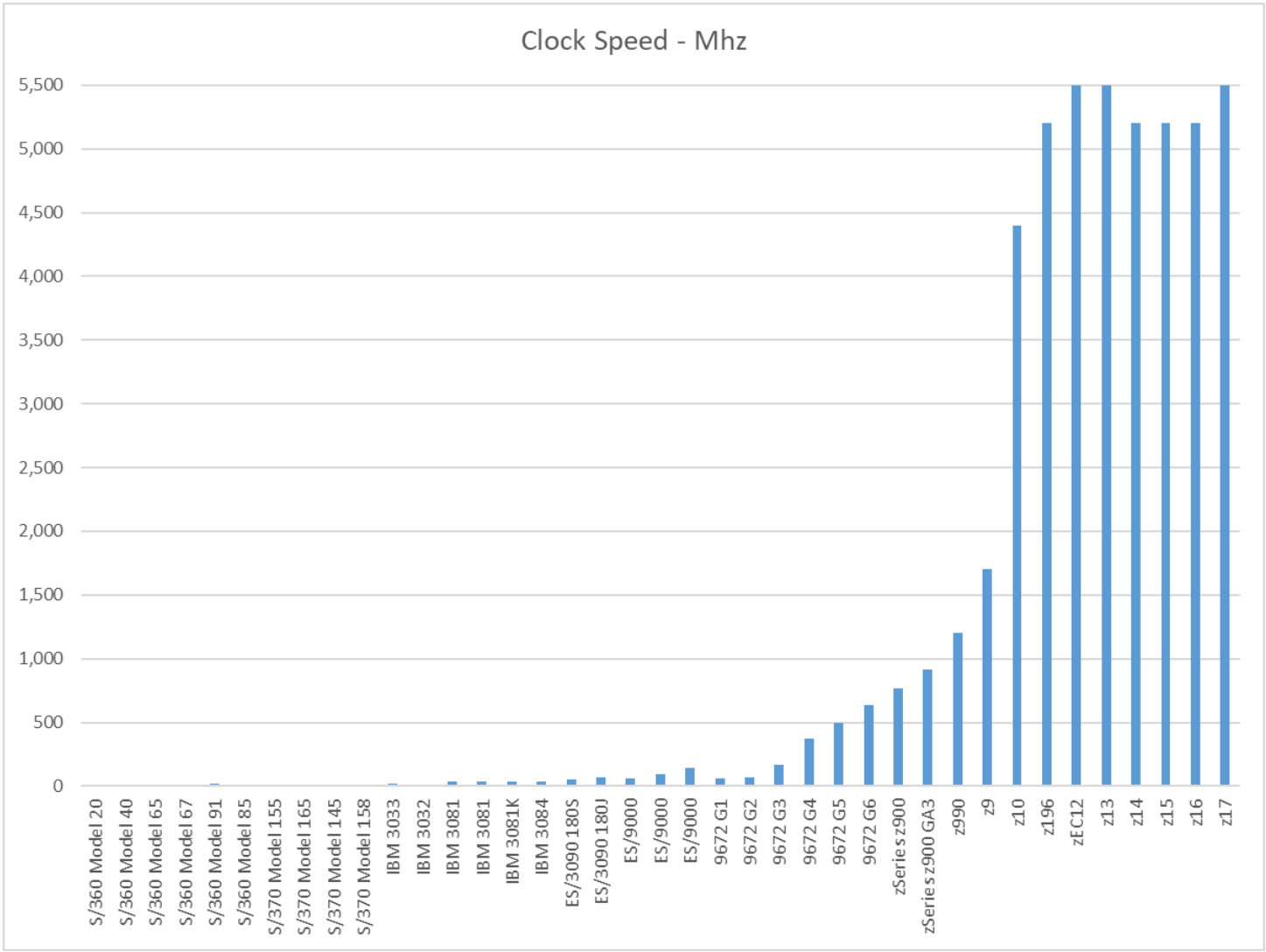
Historic CP counts



Historic max machine capacity



Historic clock speeds



Processor Design

About that clock speed…

- "Clock speed" or "Cycle time" or "Clock cycle" basically refers to a quantum of time that is used to control the flow of work through the processor
  - Think of it as a metronome for the processor
  - Often referred to as a frequency, e.g. 5.5Ghz = 0.18 nanoseconds
    - Also: 54.51mm (distance light can cover in a vacuum in 180 picoseconds)
  - Represents a commit point for in-flight operations
    - Electrical signals take time to propagate around the chip so need a point in time of truth
  - Faster clock speed generally means more work done per unit of time
    - Because we have shorter quanta of time
- Note that the clock speed was mostly flat recently despite continual increases in the capacity of the processors
  - Higher clock speeds can require more power = more heat = more problems
  - Also, hard to get the necessary things done, especially considering distances

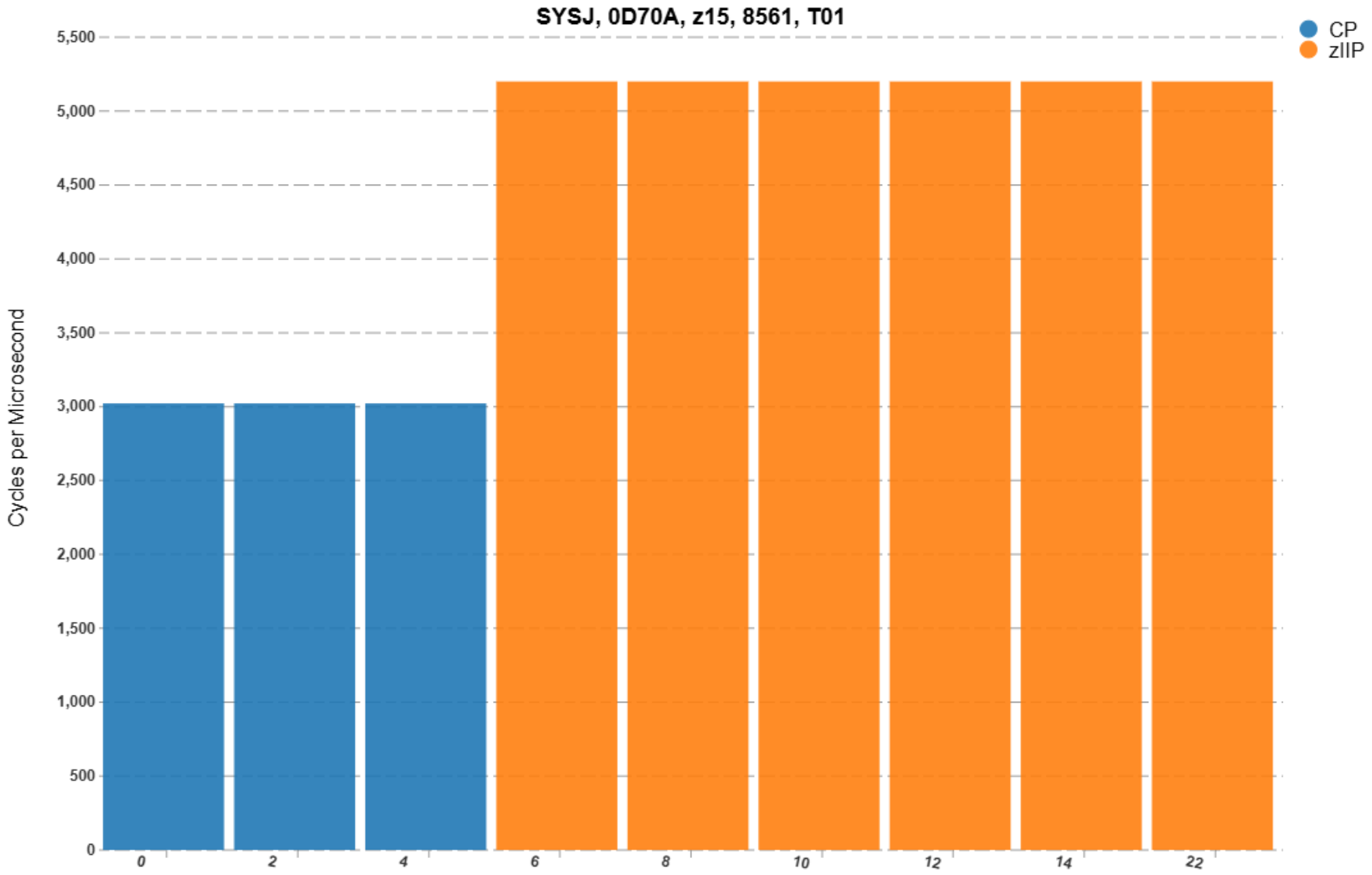
Physical distance to the data matters!

Speed and capacity

- All of the (e.g.) z17 processors run at the same clock speed
- But some GPs may be “sub-capacity” engines, which we often say are “slower” CPs
  - “slower” = “get less work done per unit of time” (make sense)
  - But the physical clock speed is not any slower
    - “Virtually”, in some measurements, it may appear to be
  - Notionally, think about the no-ops being injected into the instruction stream
- Or we say a new machine has “faster” CPs when the clock speed hasn’t changed
- I.E. we often talk about the “speed” of the CPs when we really are referring to the capacity of the individual CPs
  - I’m mostly ok with this ambiguity, but feel compelled to point it out

Processor Speed (in Cycles per Microsecond)

SMF 113

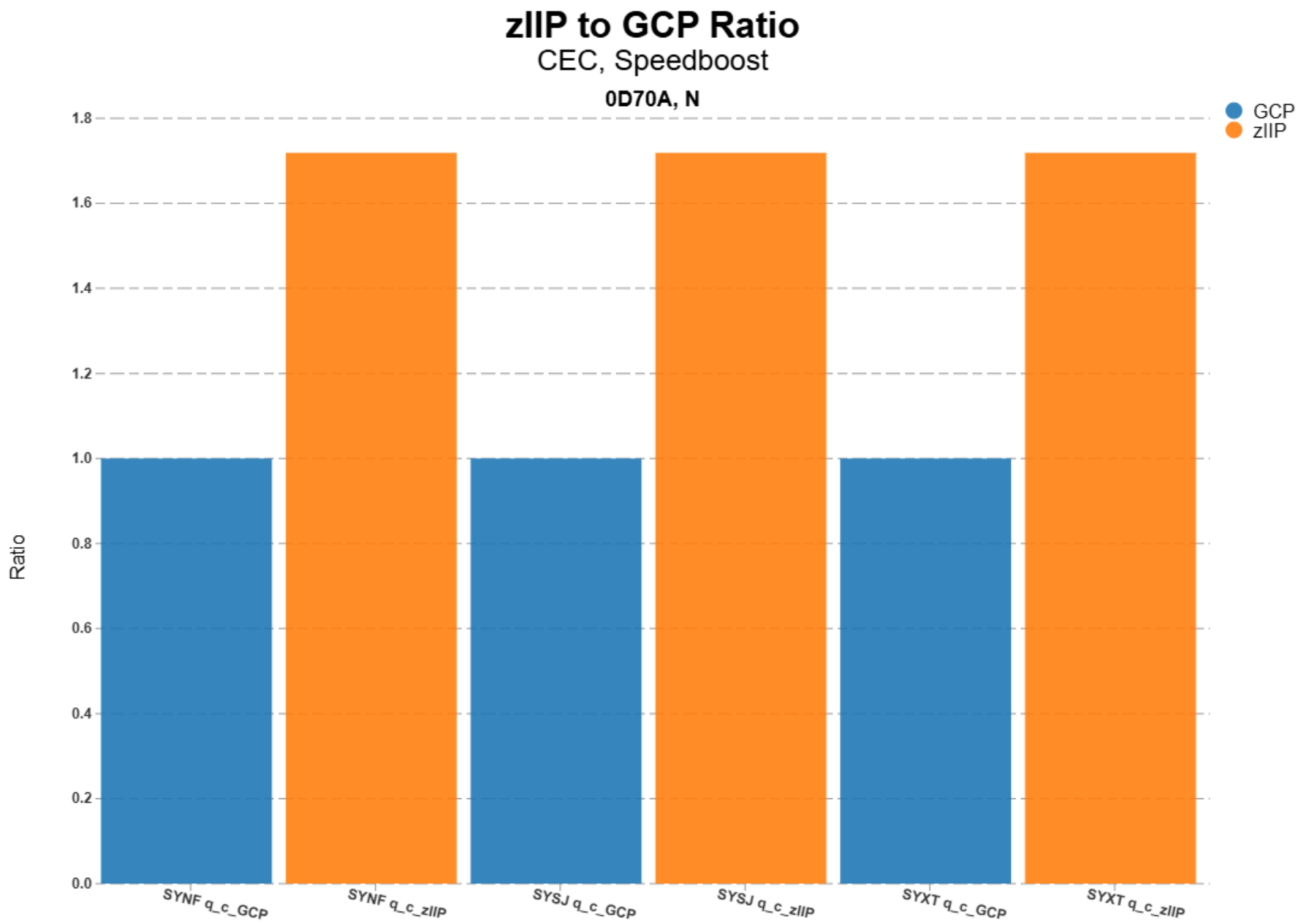


Virtual clock speed difference from the 113s.

zIIP = 5200

CP = 3022

Ratio = 1.7207

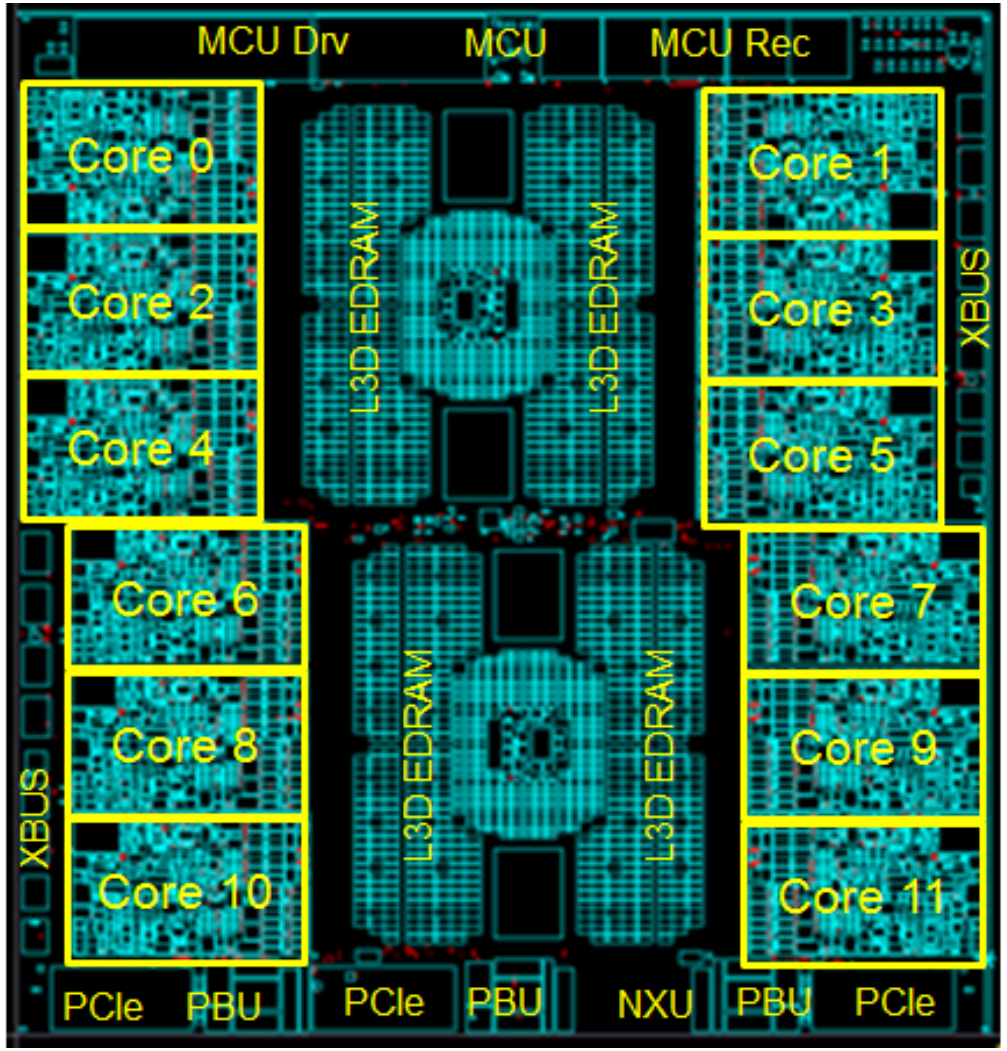


Ratio used by system in CPU calculations to normalize CPs to zIIPs.

Ratio = 1.7188

z15 PU Chip

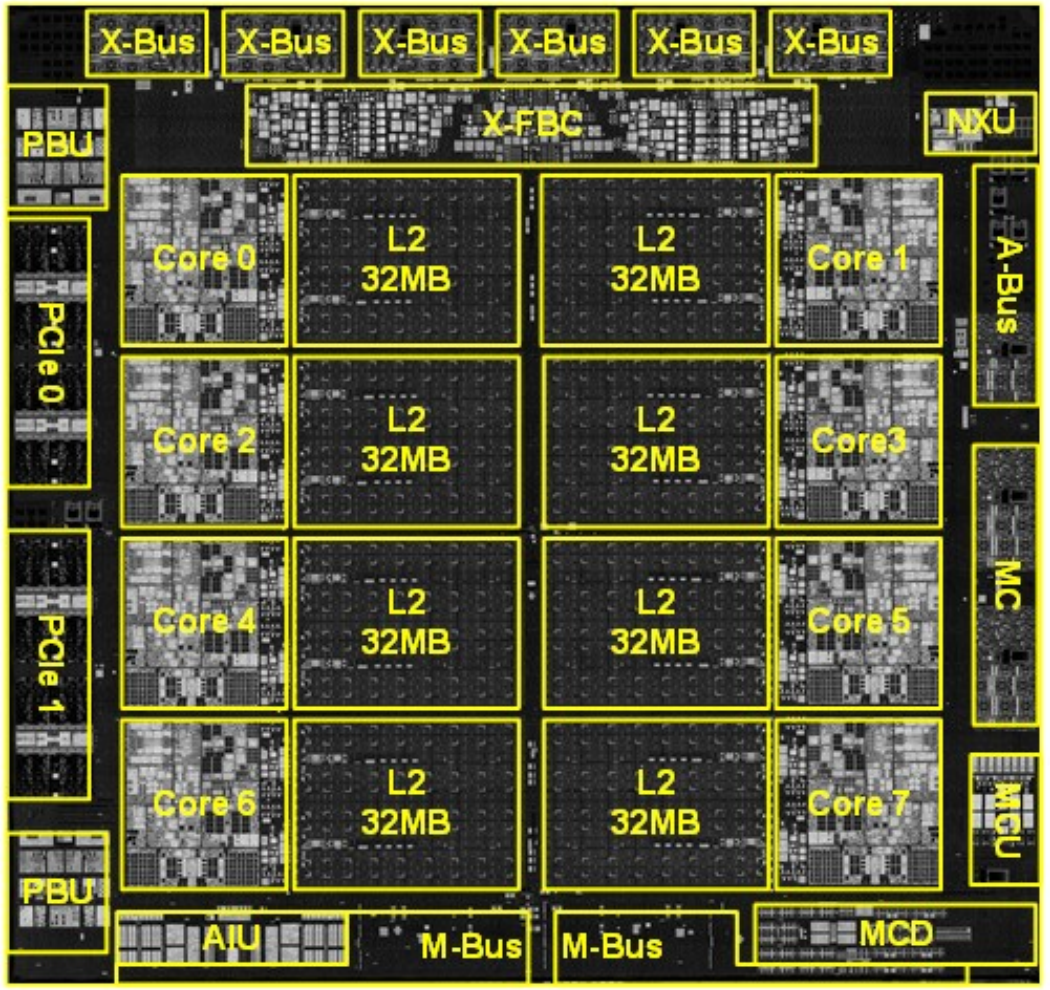
- This is one z15 PU (Processor Unit) Chip
  - About 1” square (25.3mmx27.5mm)
  - 9.2B transistors
- 4 chips per drawer (each on SCM)
- 12 cores (9, 10, or 11 “active”) per chip
  - 41 active cores per drawer < Max190
  - 43 active cores per drawer Max190
  - Wafer yields improved by utilizing chips that have some cores disabled
- Notice amount of chip area for L3 cache
  - Note cores rotated to orient L2 near L3
  - Distance matters!
- L4 is a separate chip in the drawer



© IBM

z16 PU Chip - Telum

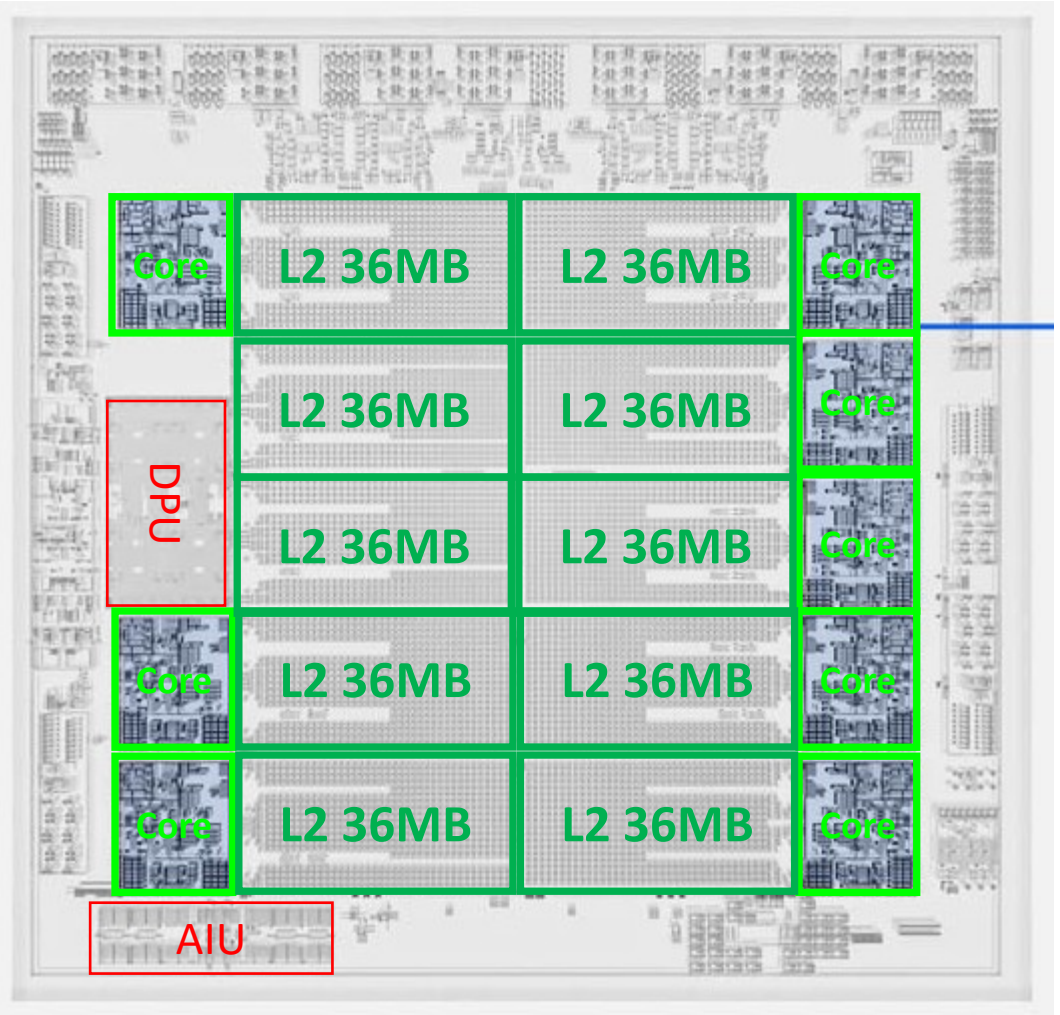
- This is one z16 PU (Processor Unit) Chip
  - A bit under 1” square (530 mm^2)
  - 22.5B transistors
- 2 chips per DCM, 4 DCMs per drawer
- 8 cores per PU (not all may be active)
  - 48 active cores per drawer < Max200
  - 57 active cores per drawer Max200
  - Wafer yields improved by utilizing chips that have some cores disabled
- Note large L2 and no specific L3/L4
  - Virtual L3/L4 from sharing L2 between cores



© IBM

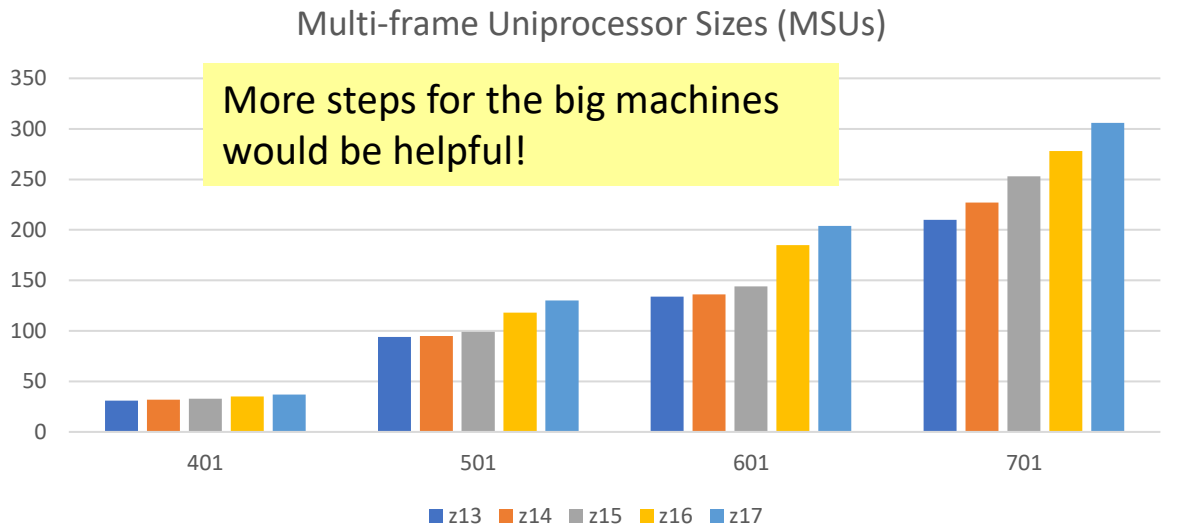
z17 – Telum II Chip

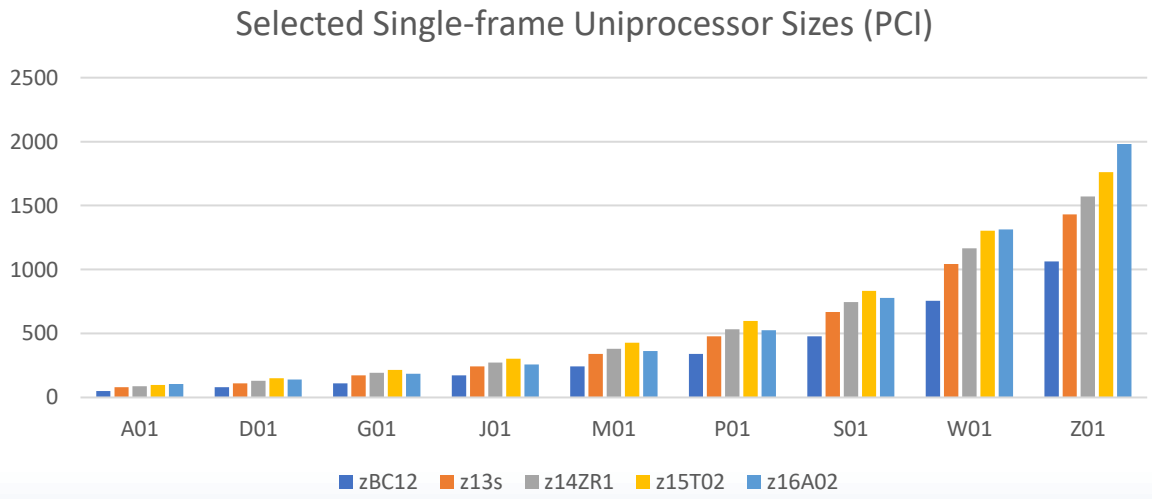
- This is one z17 PU (Processor Unit) Chip
  - A bit under 1” square (566 mm^2)
  - 43B transistors
- 2 chips per DCM, 4 DCMs per drawer
- Still 8 z Cores, but 10 L2 cache areas
  - 50 active cores per drawer < Max208
  - 60 active cores per drawer Max208
  - Wafer yields improved by utilizing chips that have some cores disabled
- Similar L2/L3/L4 cache design, but more of it
- DPU (Data Processing Unit) core takes up space from two of the z cores & replaces custom ASICs on FICON cards



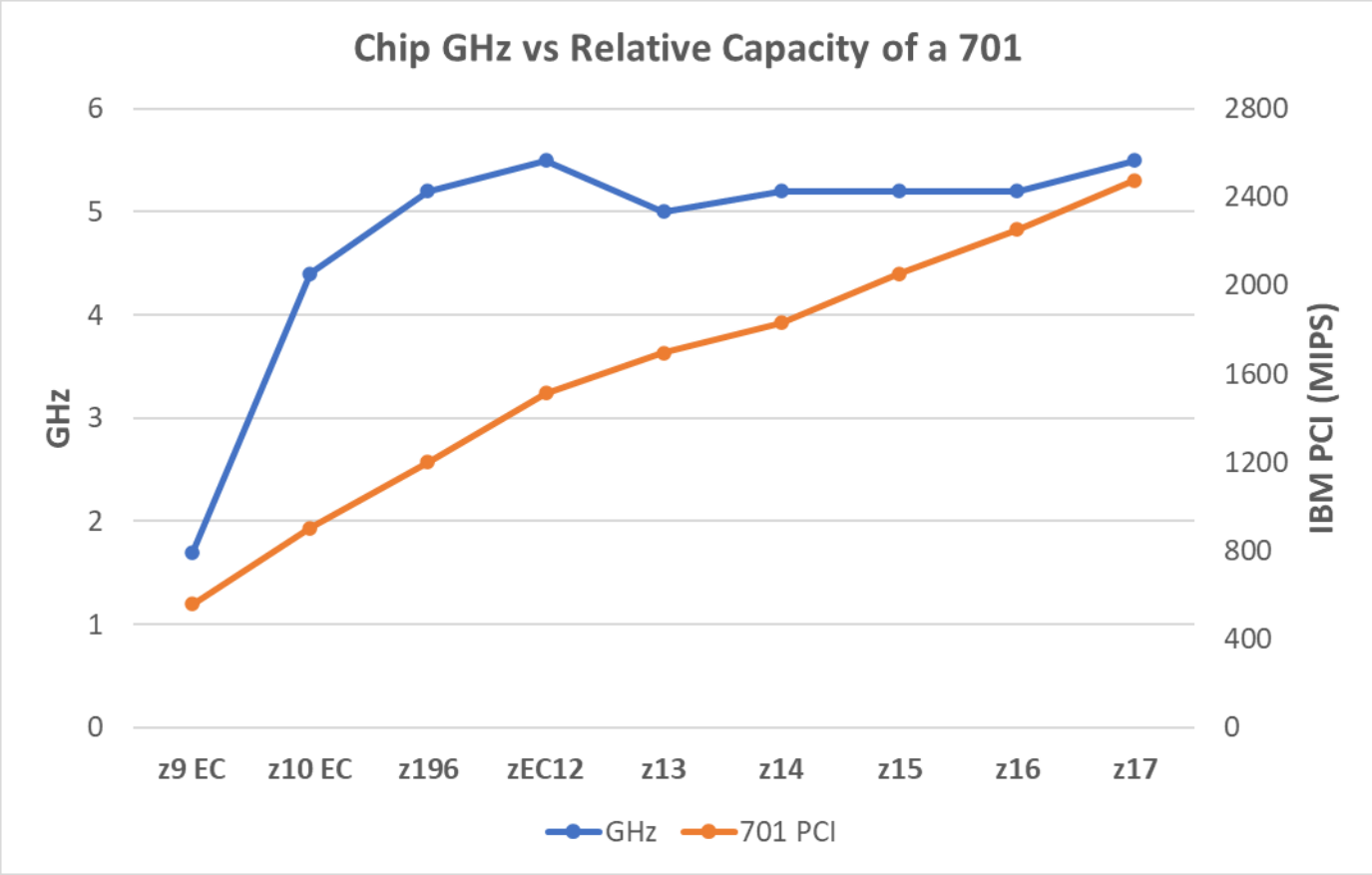
Sub-capacity Capacity Increases (or not)

- IBM sets the capacity of the sub-capacity models
- Sub-capacity models may not see the same per-processor capacity/performance increase that the full-speed machines see
  - z16 started adding capacity to the sub-capacity models after IBM held them mostly steady for 3 generations
  - Interesting that for some z16 A02 capacity settings, they dialed capacity *down* from the z15 T02 level for the same step
- Whether this is good or bad depends on your specific situation
  - Always use zPCR to model your proposed upgrade!

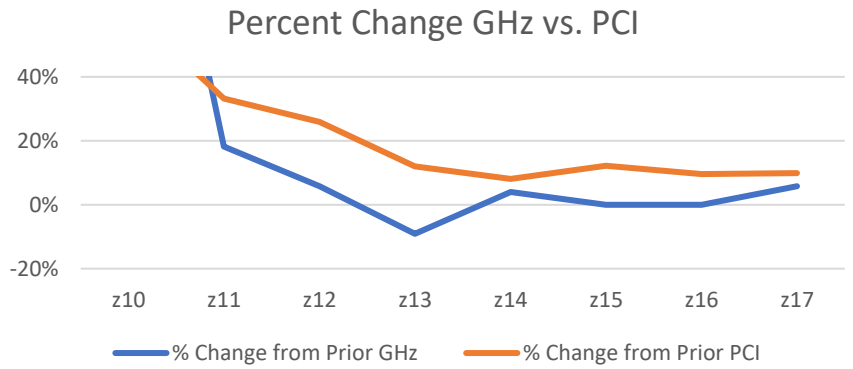




Increased speed without clock speed



- *Architectural* improvements represent most (sometimes all) of the capacity/speed improvements over the past 10+ years
  - Cache changes
  - New instructions
  - Micro/milli/pico-code changes



Changes Detailed

							Max per first book-drawer				Processor Cache					
											Core-level				Chip	Book-dwr
zGen	Name	Year	Mach Type	GHz	701 PCI	701 MSUs	Memory	CPs	PU Chips	Cores/chip	L1-Data	L1-Instr	L2-Data	L2-Instr	L3/chip	L4/bk-dwr
z9	z9 EC	2005	2094	1.7	560	81	128G	8	8	2	256K	256K	n/a	n/a	n/a	40M
z10	z10 EC	2008	2097	4.4	902	115	384G	12	5	4	128K	64K	3M		n/a	48M
z11	z196	2010	2817	5.2	1202	150	704G	15	6	4	128K	64K	1.5M		24M	192M
z12	zEC12	2012	2827	5.5	1514	188	704G	20	6	6	96K	64K	1M	1M	48M	348M
z13	z13	2015	2964	5	1695	210	2464G	30	6	8	128K	96K	2M	2M	64M	960M
z14	z14	2017	3906	5.2	1832	227	8000G	33	6	10	128K	128K	4M	2M	128M	672M
z15	z15	2019	8561	5.2	2055	253	8000G	34	4	12	128K	128K	4M	4M	256M	960M
z16	z16	2022	3931	5.2	2253	278	9984G	39	4x2	8	128K	128K	up to 32M		up to 224M	up to 1.75G
z17	z17	2025	9175	5.5	2477	306	16TB	43	4x2	8	128K	128K	up to 36M		up to 324M	up to 2.88G

- Other measures go up or down, but there's always a cache size that goes up
  - Fast access to data is critical for increasing performance
  - L1 cache size limited by clock frequency
- z17 got clock speed increase and all cache levels stayed same or increased
  - At z12 and z14 clock speed bumps, cache changes were mixed

Notable changes by generation

Gen	Changes
z13	SMT 3-for-1 memory deal leads to more affordable, larger memory sizes CFCC Levels 20 & 21 bring larger memory support and async CF duplexing
z14	DAT changed from pico-code to multiple hardware engines (a big part of the MIPS increase) zHyperLink SMT enhancements and enabled for IOPs (SAPs) Clock speed increase
z15	System Recover Boost (SRB) zEDC on chip replaced zEDC Express PCIE cards (keeping data closer to the core) SORTL instruction (although of questionable value)
z16	Cache restructuring to virtual L3 and L4 and is now all faster SRAM instead of eDRAM RAIM moved to the DIMMs AI Unit SRB for Middleware Recovery
z17	DPU (Data Processing Unit) moves FICON functionality from I/O card to the processor chip DDR5 memory (increased bandwidth) Significantly enhanced AI Unit plus available AI accelerator cards (Spyre) Clock speed increase

Performance impact by workload

- Clock speed increase can improve all workloads
  - Note though that cache misses still take time so improvement may not be entirely uniform across all workloads
- Architectural changes will impact some workloads more than others, E.G.
  - If a workload fits all within L3 cache, increasing L3/L4 cache won't help
    - But increasing L2 probably would be helpful
  - z14 DAT improvement was a significant improvement for many systems
  - SORTL has shown limited benefits for customers I've talked to
  - Cache-unfriendly workloads may benefit more from faster memory
- Understanding your workloads can help you understand how a potential migration might affect those workloads
  - zPCR will help with this, providing better impact estimates than just using the MIPS/MSU ratings

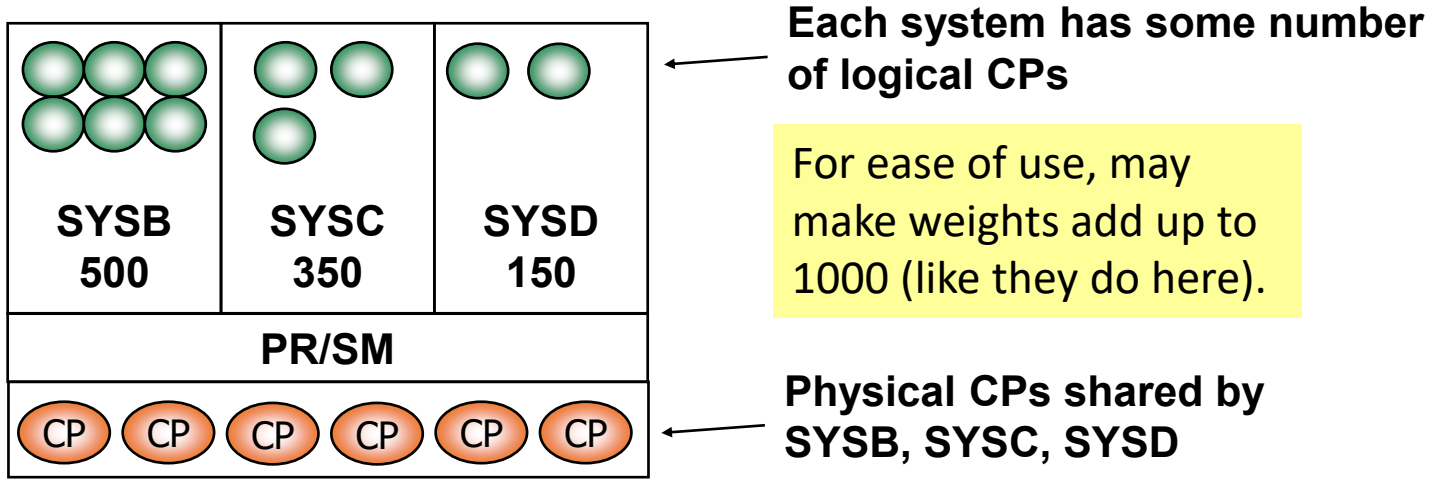
Logical processors

Logical and Physical CPUs

- Processor = CP = CPU = GCP or zIIP or any other processor type
  - All the same bit of silicon: a core on a physical chip
- You pay for a certain number of physical processors (CPs)
  - A processor can only be processing one stream of instructions at a time
    - Absent SMT, which doesn't apply to GCPs and which we're not going to discuss here
- You define LPARs, each with a certain number of logical, shared CPs
  - For each LPAR Logical CPs <= physical CPs, although can have reserved CPs
  - Most machines have multiple LPARs
- z/OS dispatches work to its (logical) CPs
- PR/SM dispatches logical CPs to physical CPs
  - A logical CP can't do any work when it's not dispatched to a physical CP
  - If you only have 1 physical CP, only 1 LPAR is doing anything at any given instant

Weights and logical CPs

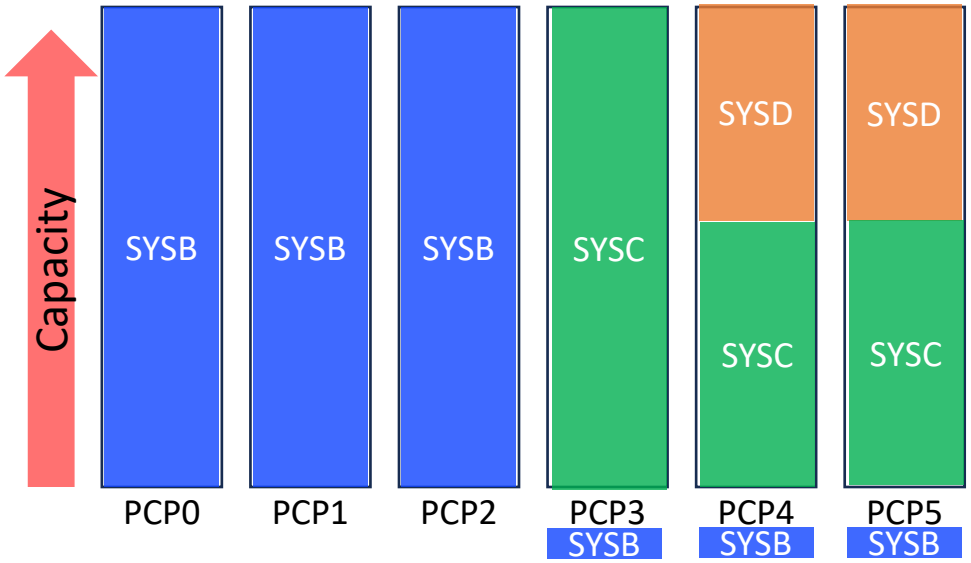
- Each LPAR is guaranteed to get at least its share
  - $LPAR\ Share = 100\ * \frac{LPAR\ Weight}{\sum Weight\ of\ activated\ LPARS}$
- In below example:
  - SYSB – guaranteed 50% of capacity of the 6 CPs (3 CPs worth of capacity)
  - SYSC – guaranteed 35% of capacity of the 6 CPs (2.1 CPs worth of capacity)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



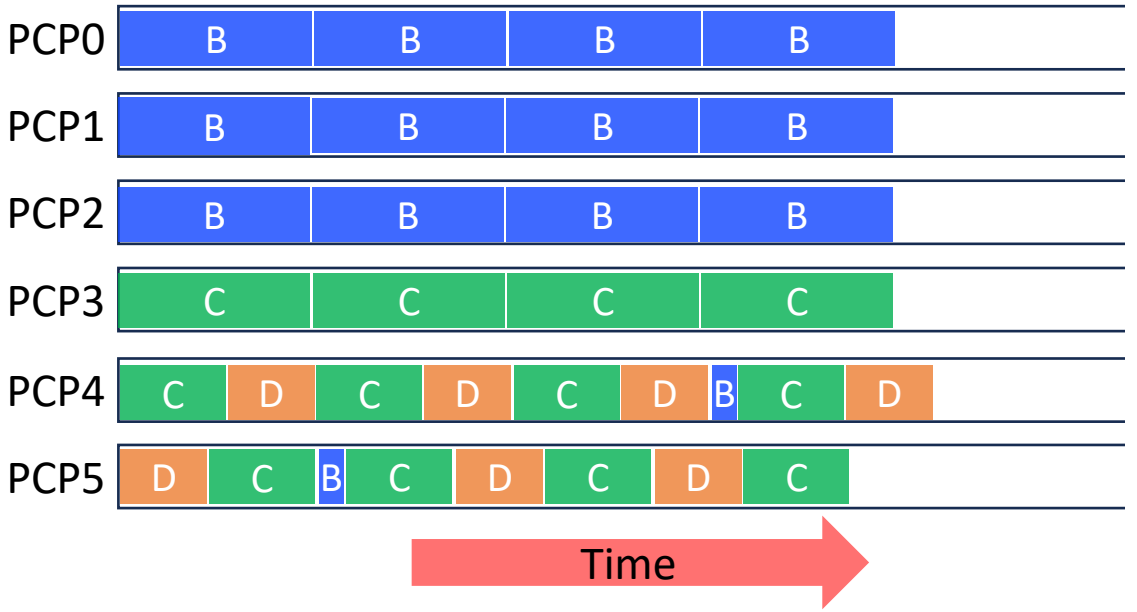
HiperDispatch CP Management

- 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

Physical to Logical: Vertical Mgt



With HiperDispatch, vertical high CPs are quasi-dedicated to an LPAR. Note that SYSB's VLs will only come into play when there's both demand from SYSB and the other LPARs aren't using the capacity.



Note that while reality may be a bit messier, vertical CPU management does greatly reduce the movement of logicals to different physicals. Also note VH CPs get longer dispatch intervals.

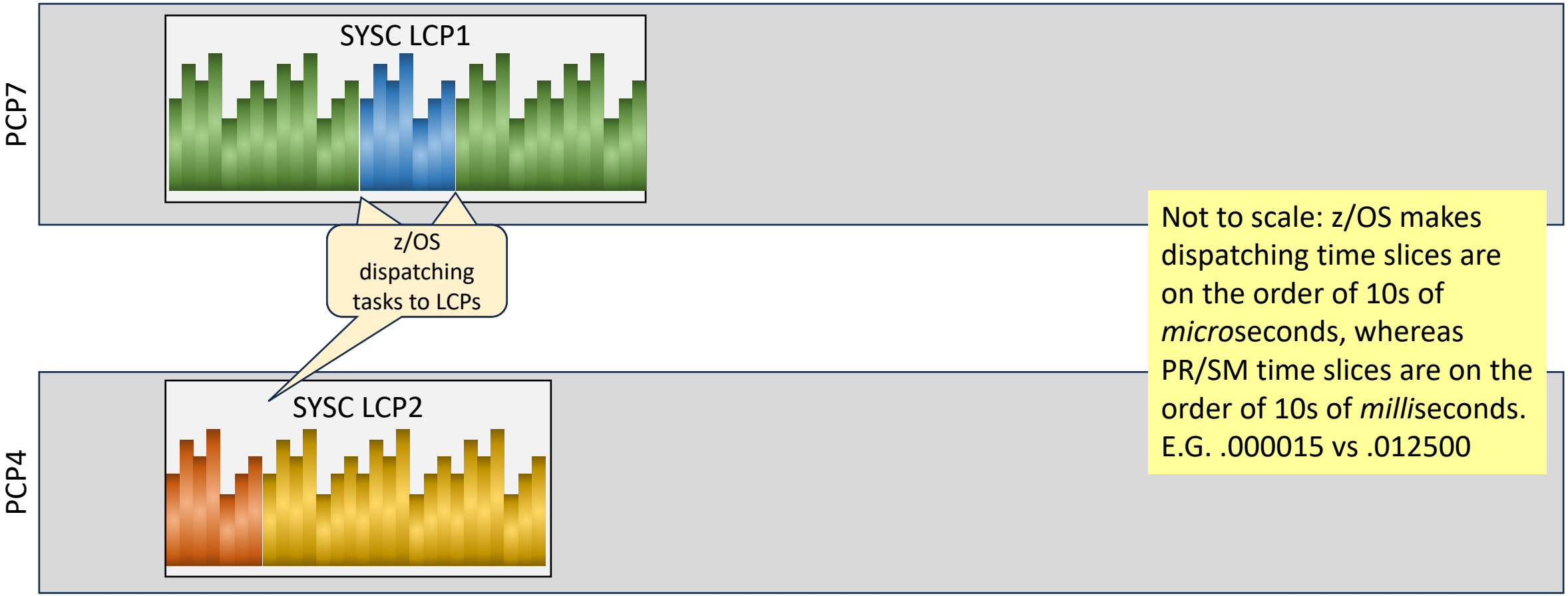
z/OS Dispatcher Affinity Nodes

- 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

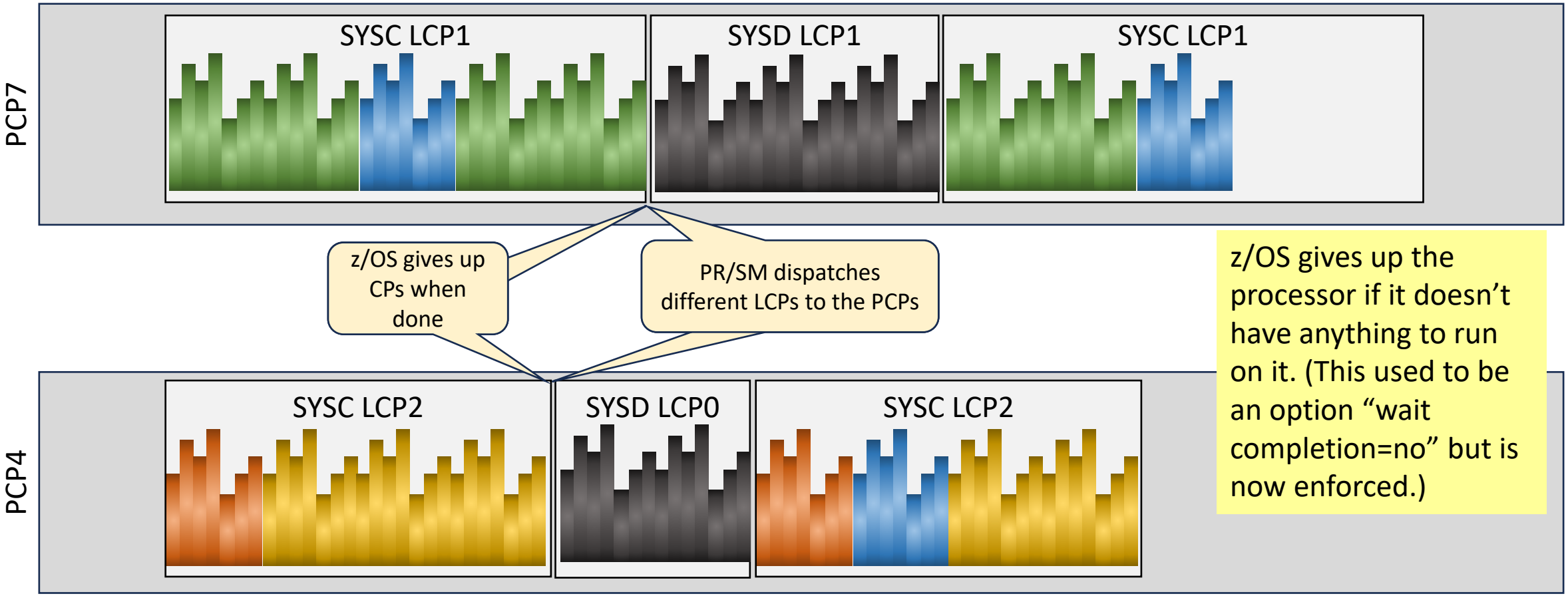
- PR/SM also enforces affinity
  - High Pool logical CPs have very strong affinity to a particular physical CP
  - Mediums will try to stay in the same area in the nest (especially at drawer level)
  - Low pool CPs have little affinity as their capacity is not guaranteed by their weight
- We care about this because we'd like the CPs to be close to the data
  - E.G. the caches (core/chip/drawer) and memory (drawer)
- 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

PR/SM Dispatching LCPs



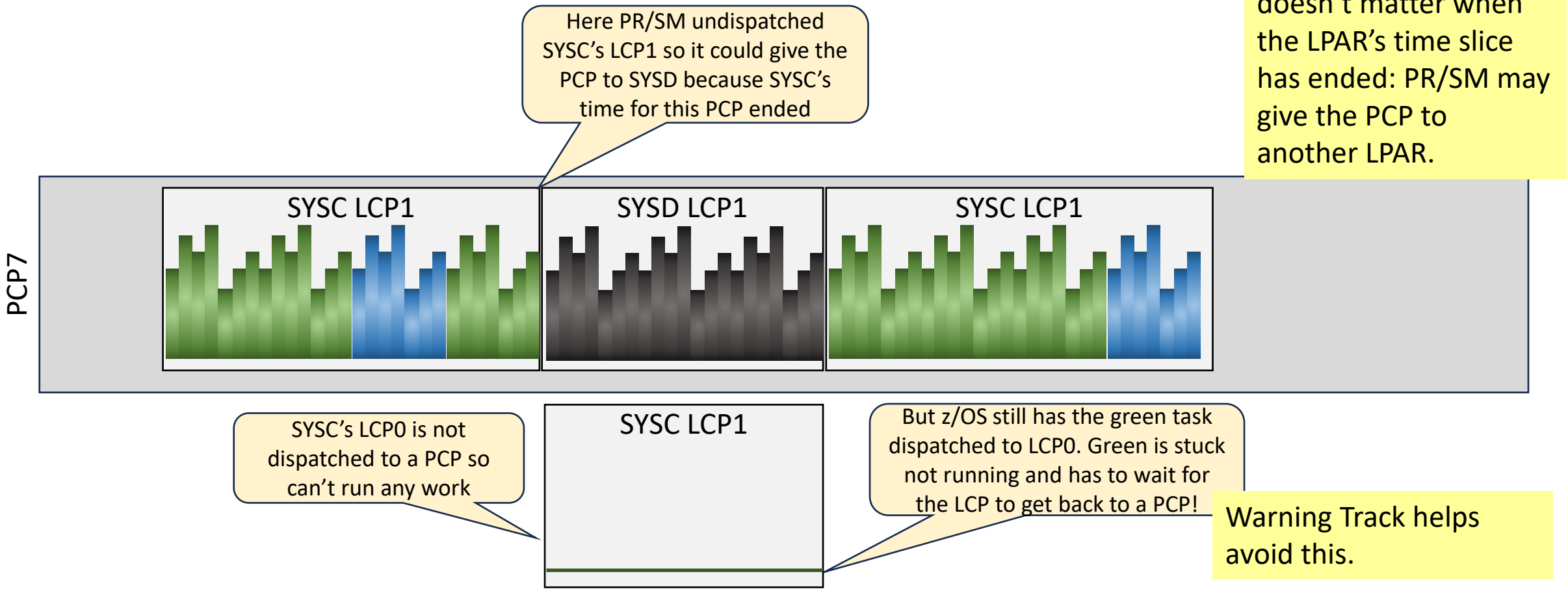
Not to scale: z/OS makes dispatching time slices are on the order of 10s of *micro*seconds, whereas PR/SM time slices are on the order of 10s of *milli*seconds. E.G. .000015 vs .012500

PR/SM Dispatching LCPs



z/OS gives up the processor if it doesn't have anything to run on it. (This used to be an option "wait completion=no" but is now enforced.)

What if z/OS task wasn't done?



Wait completion doesn't matter when the LPAR's time slice has ended: PR/SM may give the PCP to another LPAR.

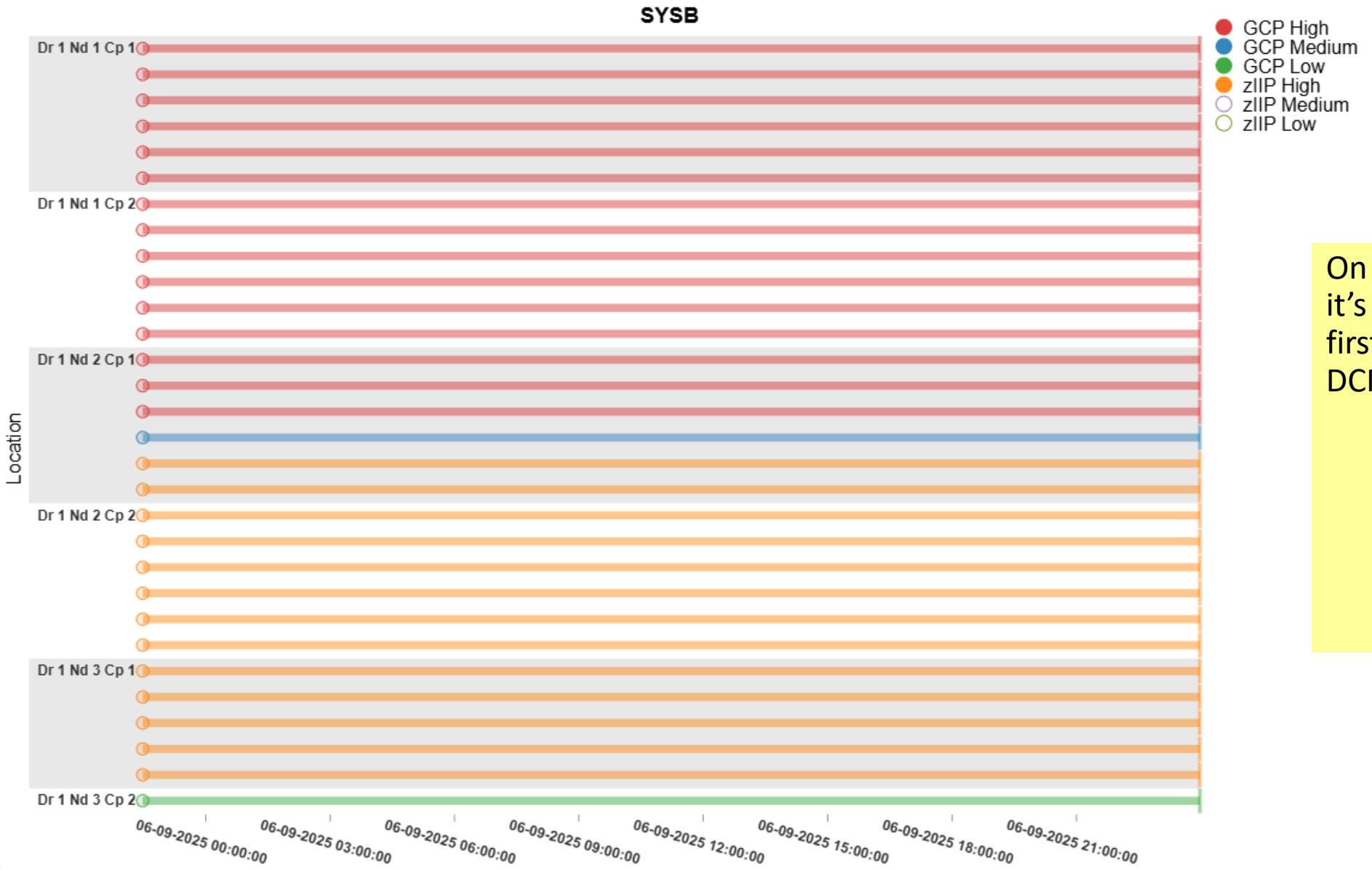
Warning Track helps avoid this.

See also my "Macro to Micro" presentation at https://www.epstrategies.com/content.html

LPAR Sizes

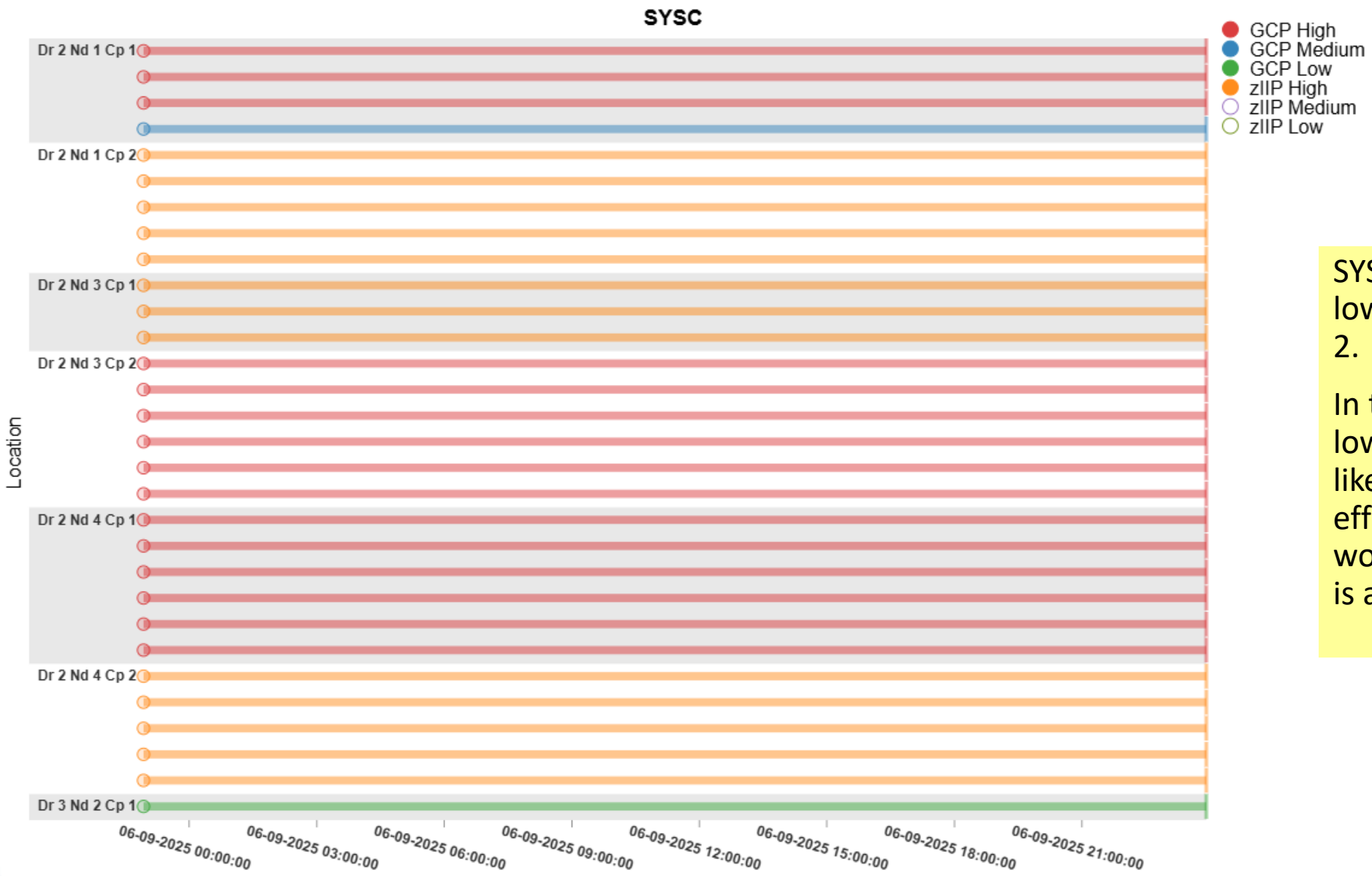
- If you have very large LPARs and are considering a multiple drawer machine
  - Large = dozens of CPs and zIIPs for an LPAR and or multiple TBs of memory
- Ideally keep an individual LPAR “small” enough to fit into a single drawer
  - CPs and zIIPs total count <= max per drawer
    - Generally easy to plan for
  - Memory <= drawer max
    - May be harder to plan for, discuss with IBM during configuration planning
    - Probably somewhat less important than CPs/zIIPs since it only affects L4 cache misses

Processor Location Assignments



On this z16, SYSB has all it's GCPs and zIIPs in the first drawer on 3 of the 4 DCMs.

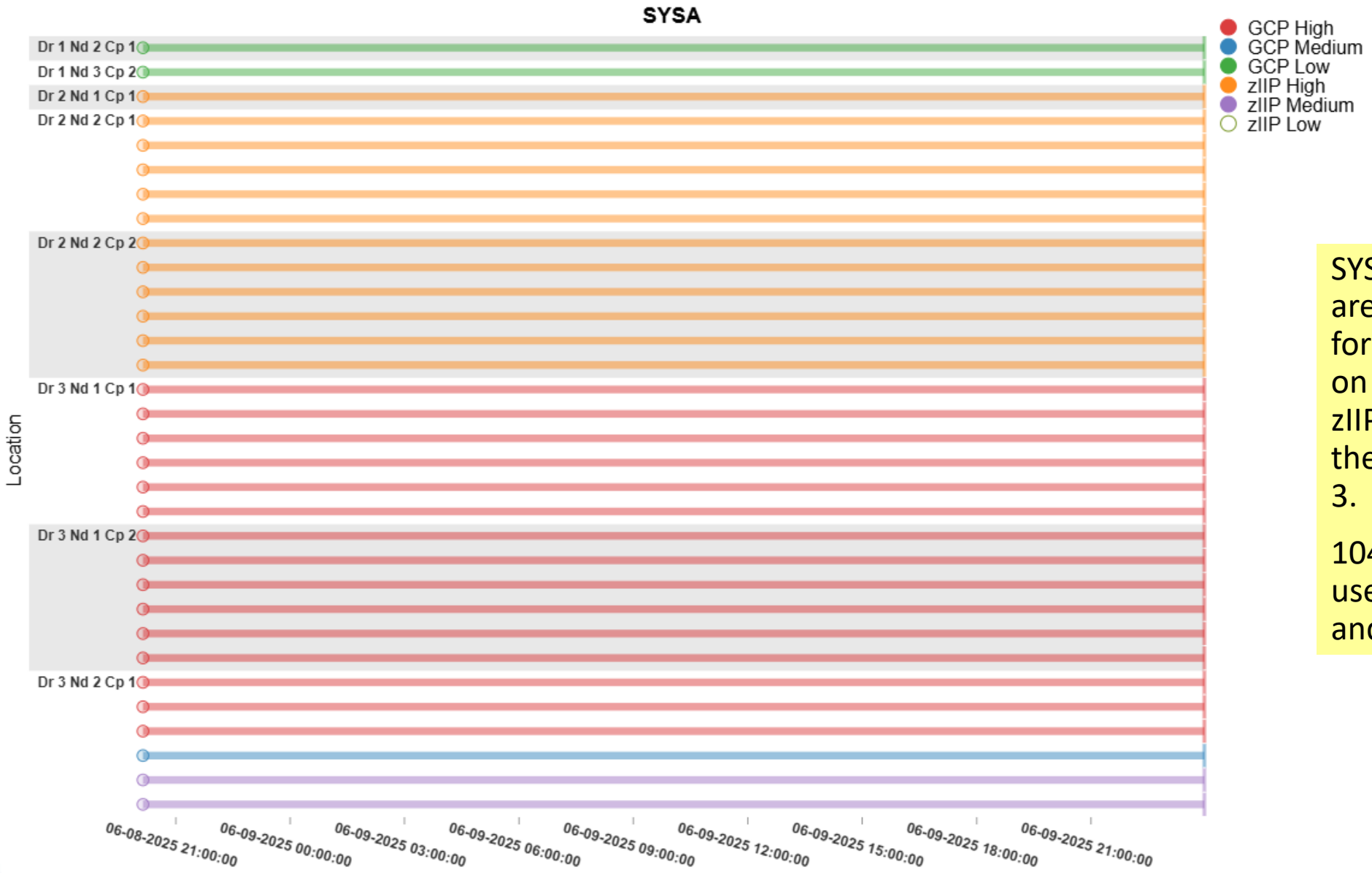
Processor Location Assignments



SYSC has all but it's one low-pool GCP on drawer 2.

In this situation, that low-pool processor will likely run noticeably less efficiently (when it runs work) because the data is all in the other drawer.

Processor Location Assignments



SYSA is scattered: GCPs are on drawer 3, except for the lows which are on drawer 1. The high zIIPs are on drawer 2 but the medium zIIPs are on 3.

104 (of 125) CPs are in use, including 12 ICFs and IFLs.

Measurements & Comparisons

Workload Impact on CPU Efficiency

- In most business use cases, we use computers to transform data
- Accessing data takes time
  - Data closer to the core running the instructions will be accessed faster
    - Even if that “closer” is just a fraction of an inch further away in the higher cache level
    - Instruction streams (i.e. programs) have to be read too and have the same issue
- PR/SM and z/OS affinities attempt to dispatch work to near its data
  - More work more closely located to its data = less time waiting to access data
- Less time waiting for data = more CPU efficiency
  - I.E. more productive work done per unit of time
- Hardware Instrumentation Services (HIS) records processor efficiency metrics in SMF 113 records
  - Be sure to record these
- SMF 99.14 (and now 70.1) records record mapping of logical to physical cores
  - Of particular interest for multi-book machines to make sure LPARs aren’t crossing books

HIS Metrics of Interest

- CPI – Cycles Per Instruction
  - Simply calculated as number of cycles in interval / instructions completed
  - Estimated Finite CPI – CPI due to the fact that not all memory references are satisfied in L1 (i.e. because the L1 cache is finite)
    - Calculated via IBM formula (more directly on latest processors)
  - Instruction Complexity CPI – CPI due to the fact that some instructions simply take longer than others to execute
    - Calculated as CPI – Estimated Finite CPI
- Relative Nest Intensity
  - IBM formula, changes occasionally as new information becomes available about how the processors are actually performing in the field
  - See http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TC000066

More HIS Metrics of Interest

- L1MP – Level 1 Misses per 100 Instructions
  - Gives you an indication of how well you're leveraging L1 cache
  - Generally expect to be under 5 in most cases
- TLB CPU Miss Percent of CPU
  - Total percent of the CPU consumed by the LPAR that goes to dynamic address translation (DAT) due to a translation look-aside buffer miss
  - DAT is more costly than you might imagine: hope for it to be less than 5%, but not unusual for it to be more (before z14)
  - For z14+: TLB redesign basically includes the DAT for every L1 cache line
    - 1-3% seems to be common for z14 and later
- For all these metrics, best to look at the metrics on a GCP vs. zIIP basis
  - Workload and utilization differences between the processor types result in differences in the metrics, averaging them together skews the metrics

What can you do about these metrics?

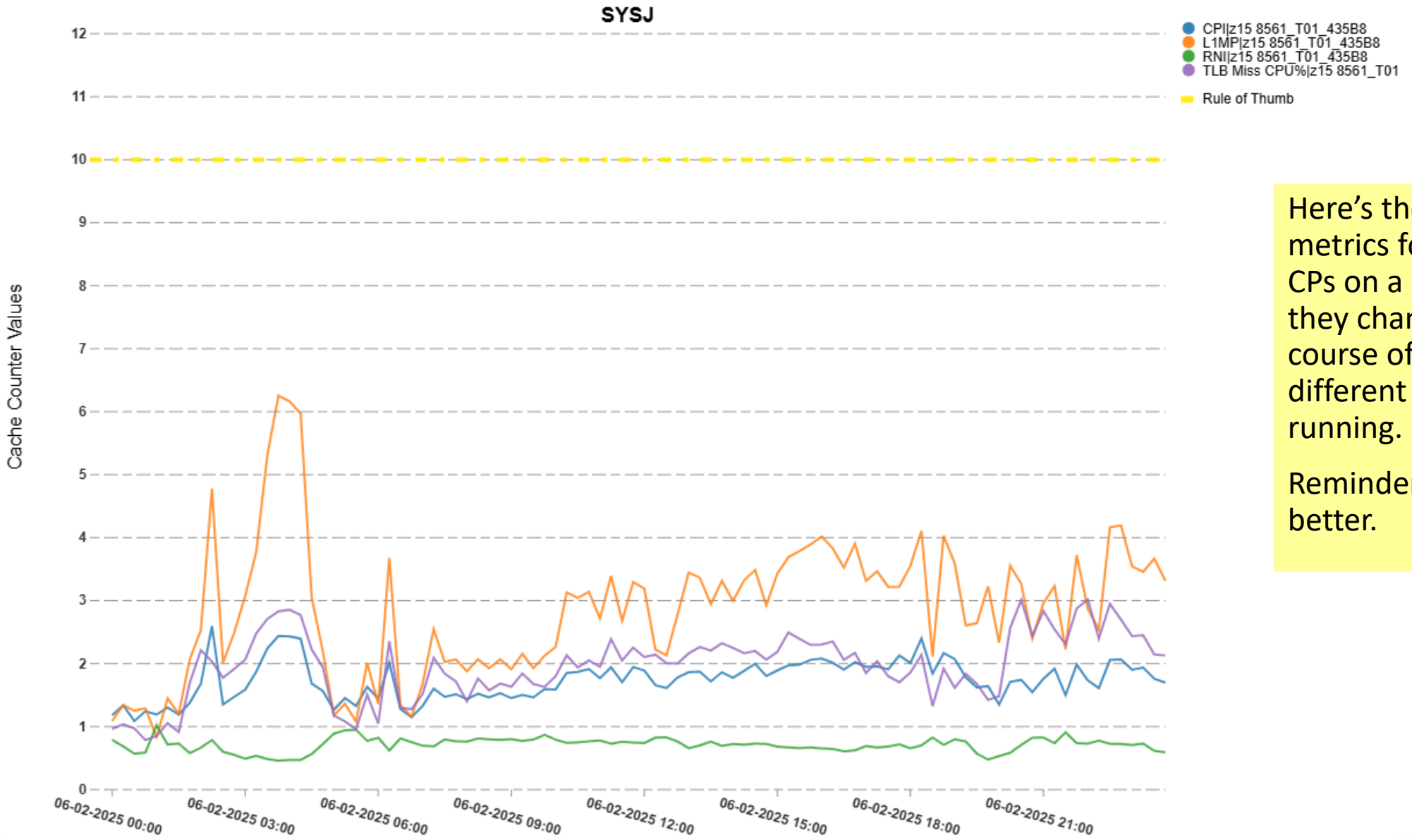
- All of these are at least partially driven by the workload characteristics, so to some degree they are what they are
  - But some values may be impacted by certain configuration choices
- Cache utilization can be impacted by the number of CPs that you have configured
  - More CPs = more L1/L2 cache
- Cache utilization can be impacted by HiperDispatch configuration
  - More vertical high processors = better L1/L2 cache utilization
- TLB effectiveness can impacted by use of large pages
  - One 1 MB page table entry covers 256x as much storage as 4K pages
    - DB2 buffer pools, JVMs, others…
  - Consider 2GB pages where appropriate
    - Large DB2 buffer pools or very large JVMs
  - z14 architecture makes this less important than on prior generations
    - Still can gain TLB2 benefits from using larger pages

Fewer/Faster vs. More/Slower

- Although dependent on the LPAR configuration and software particulars, I'm often a fan of more/slower vs. fewer faster CPs
  - E.G. a 410 vs. a 503 or 620 vs. 710
- More/slower can get you more:
  - L1/L2 cache
  - More TLB
  - More vertical high CPs
- All of the above can result in a more efficient overall system when you have more than 1 significant LPAR on the machine
- Multiple LPARs sharing a few fast CPs, each end up getting a small time slice, resulting in them processing much like slower CPs, albeit with less total cache

Processor Caches - CP CPU Key Measurements

SMF 113

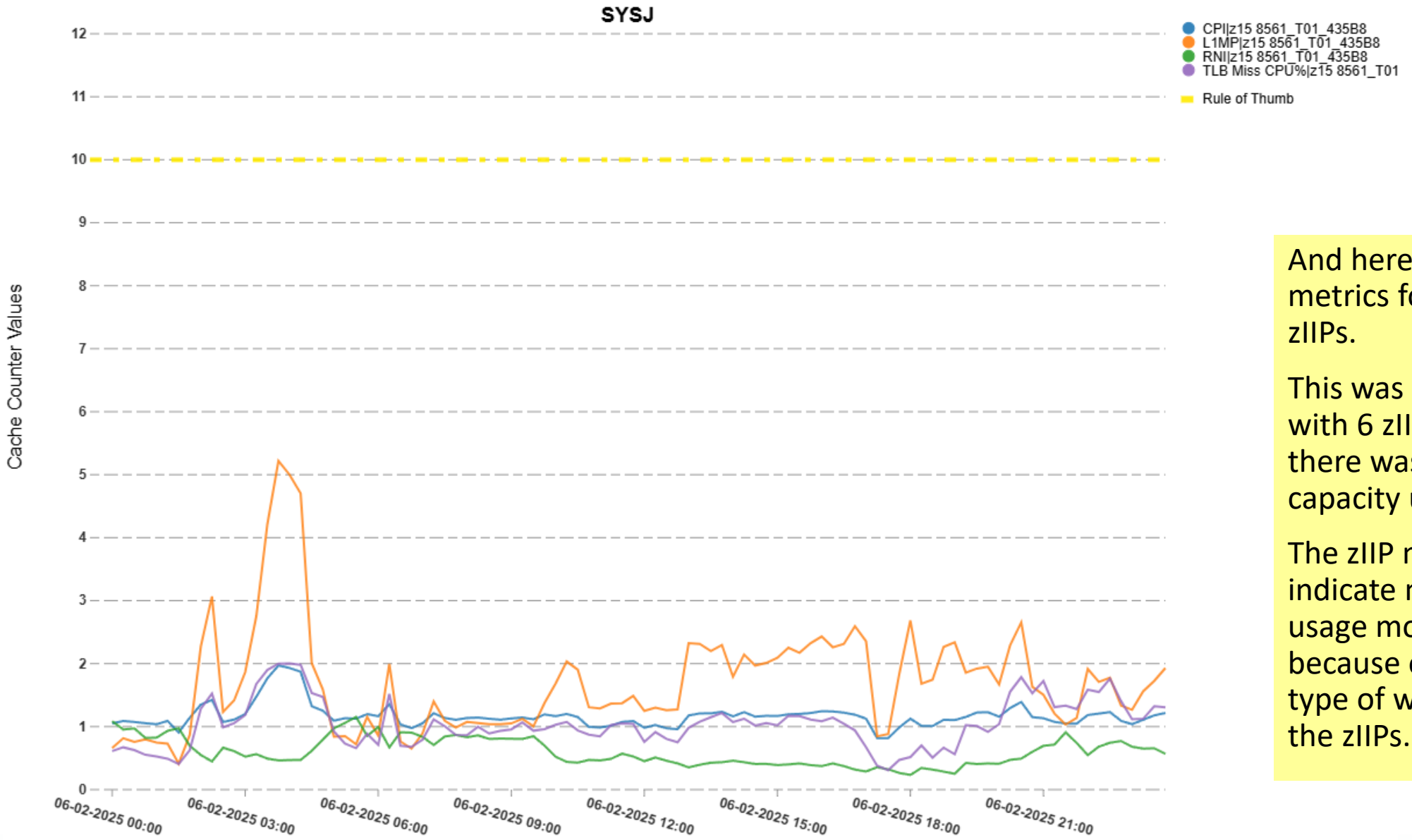


Here's those 4 primary metrics for an LPAR's GP CPs on a z15. Note how they change over the course of the day when different workloads are running.

Reminder: lower is better.

Processor Caches - zIIP CPU Key Measurements

SMF 113



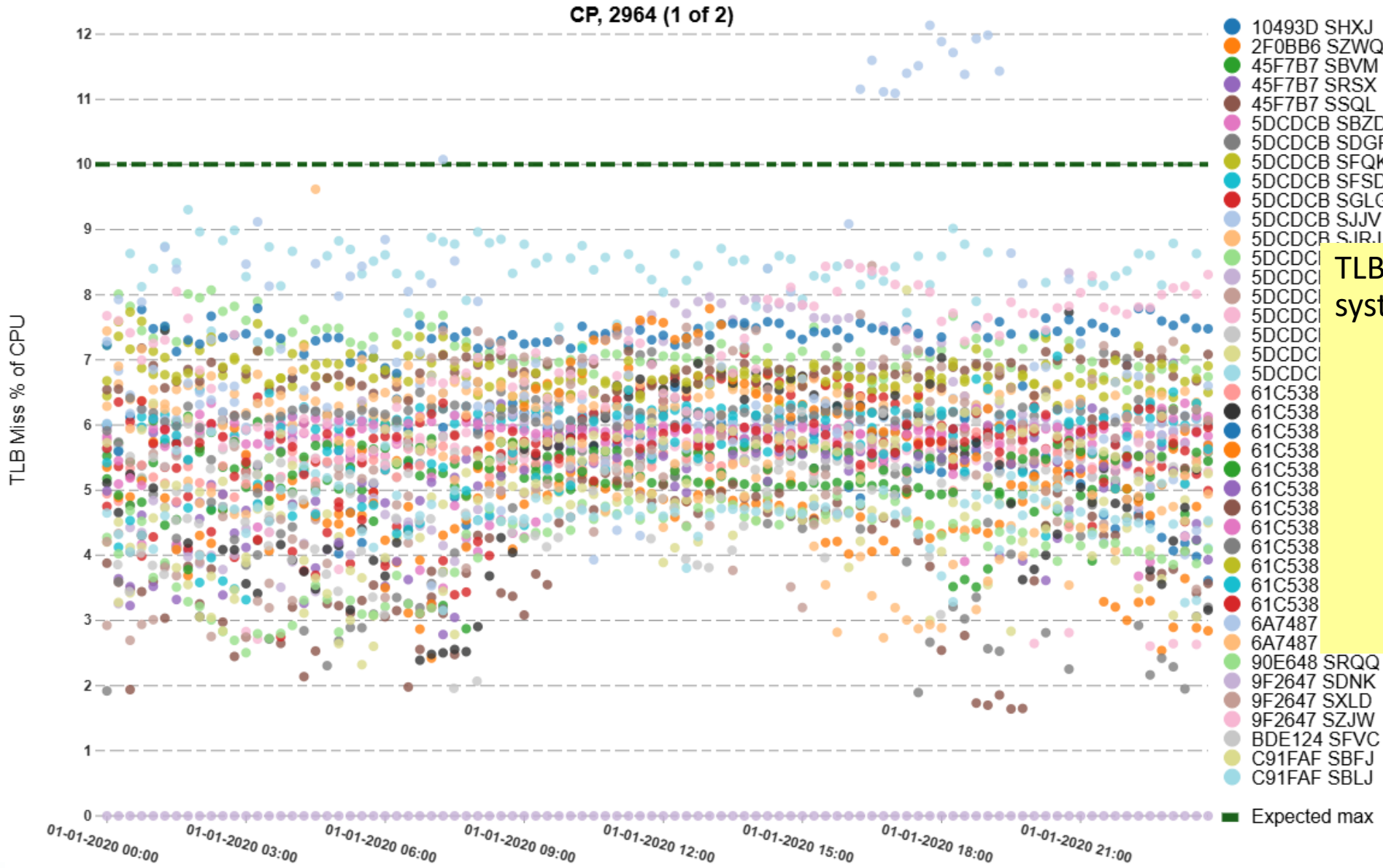
And here are the same metrics for that LPAR's zIIPs.

This was a 603 (3 GPs) with 6 zIIPs. At times there was more zIIP capacity used than GP.

The zIIP measurements indicate more efficient usage most likely because of the limited type of work running on the zIIPs.

TLB Miss CPU %

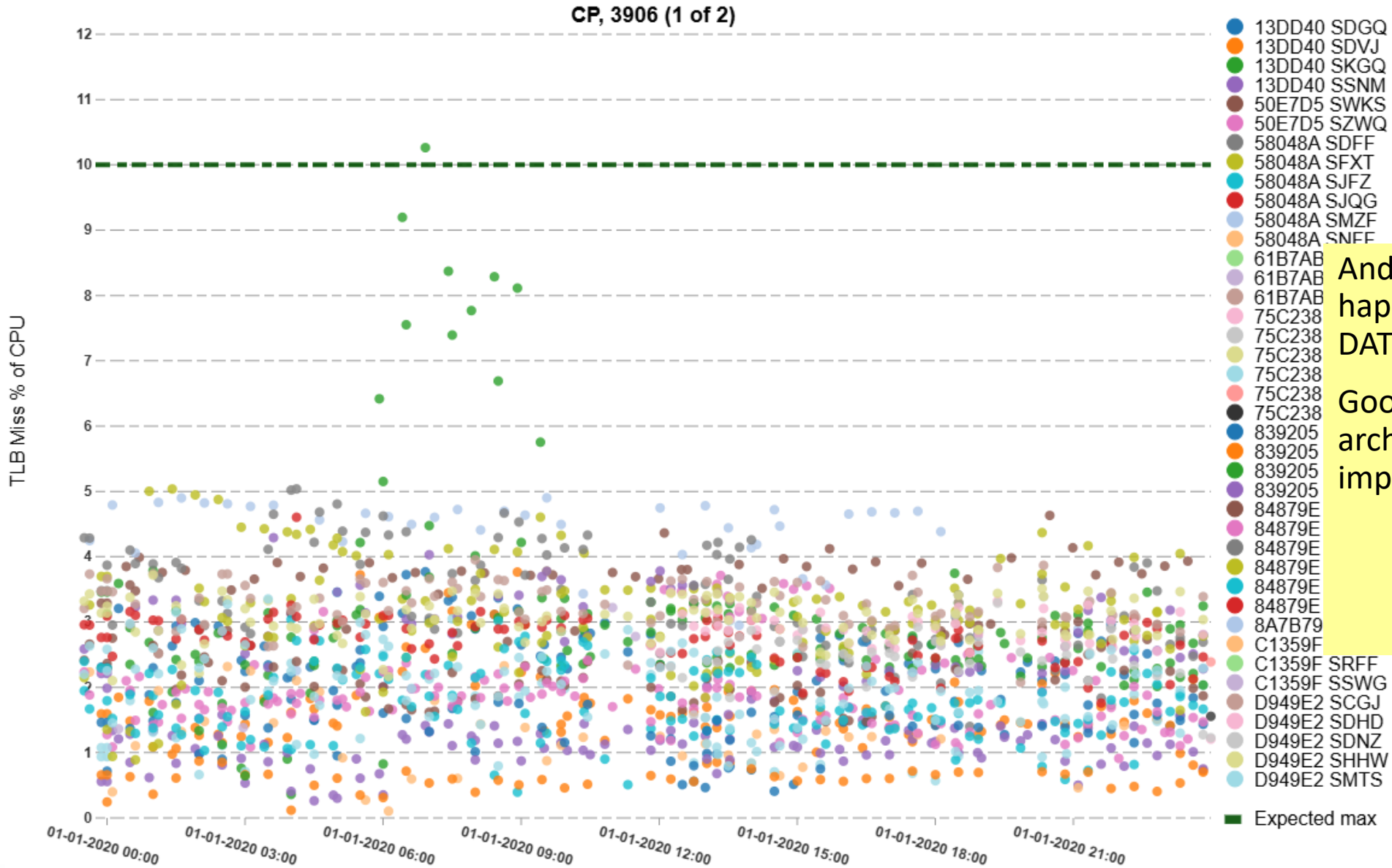
By z/OS Hardware Model



TLB Miss% for some z13 systems

TLB Miss CPU %

By z/OS Hardware Model

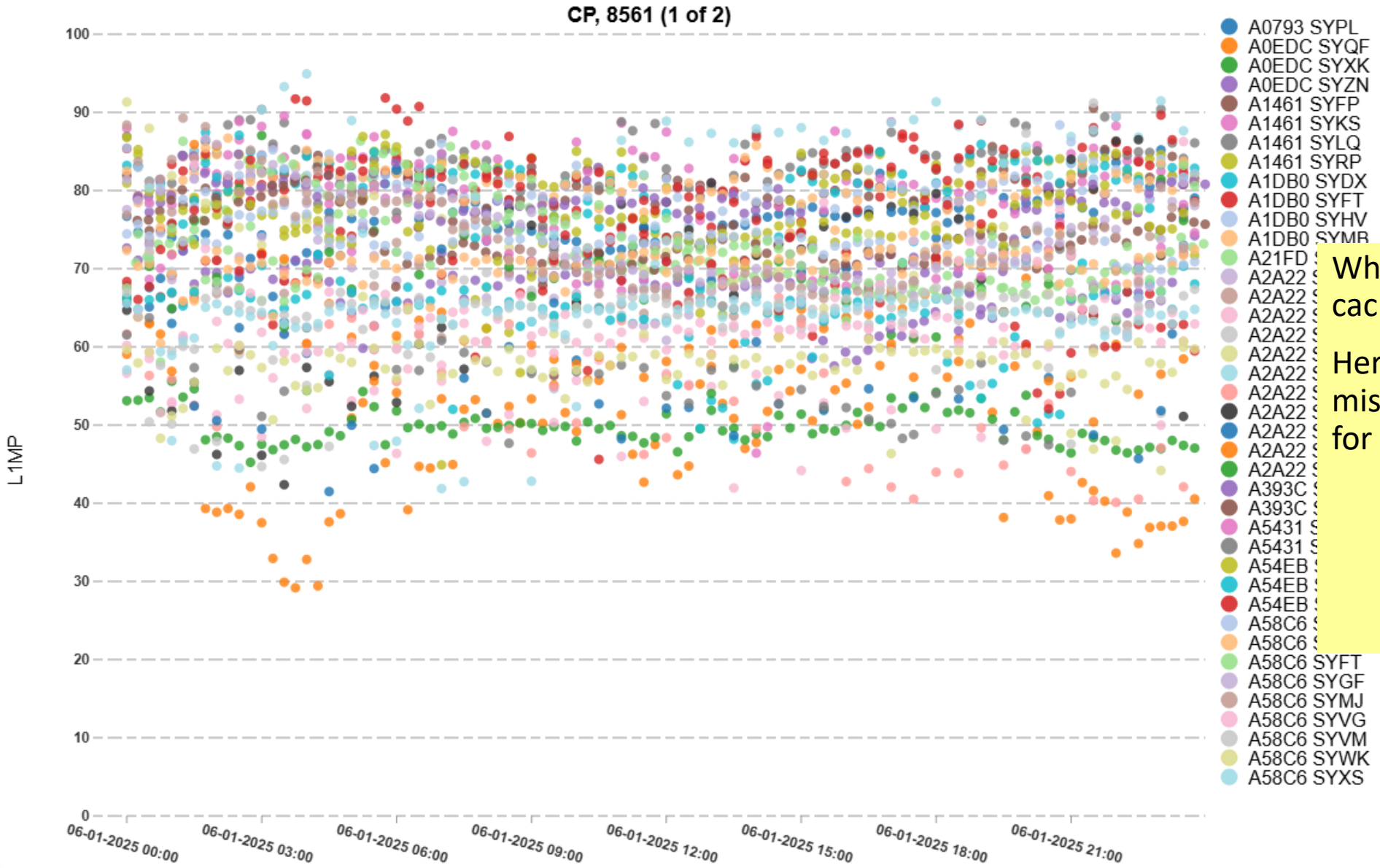


And this is what happened with the z14 DAT change!

Good example of architectural change improving performance.

Percent of L1 Misses Satisfied in L2

By z/OS Hardware Model

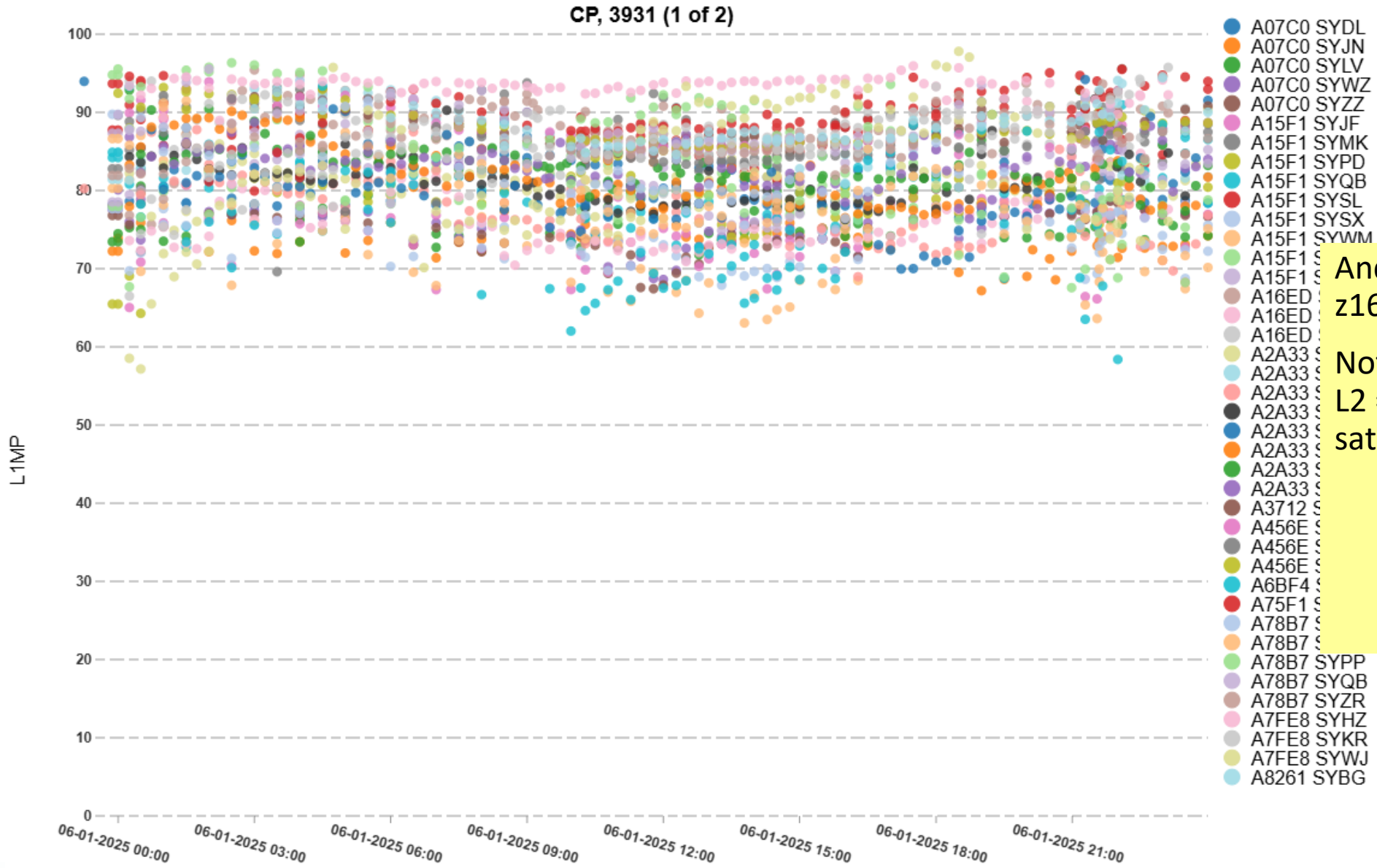


What about the major cache change in the z16?

Here's the percent of L1 misses satisfied out of L2 for some z15 systems.

Percent of L1 Misses Satisfied in L2

By z/OS Hardware Model

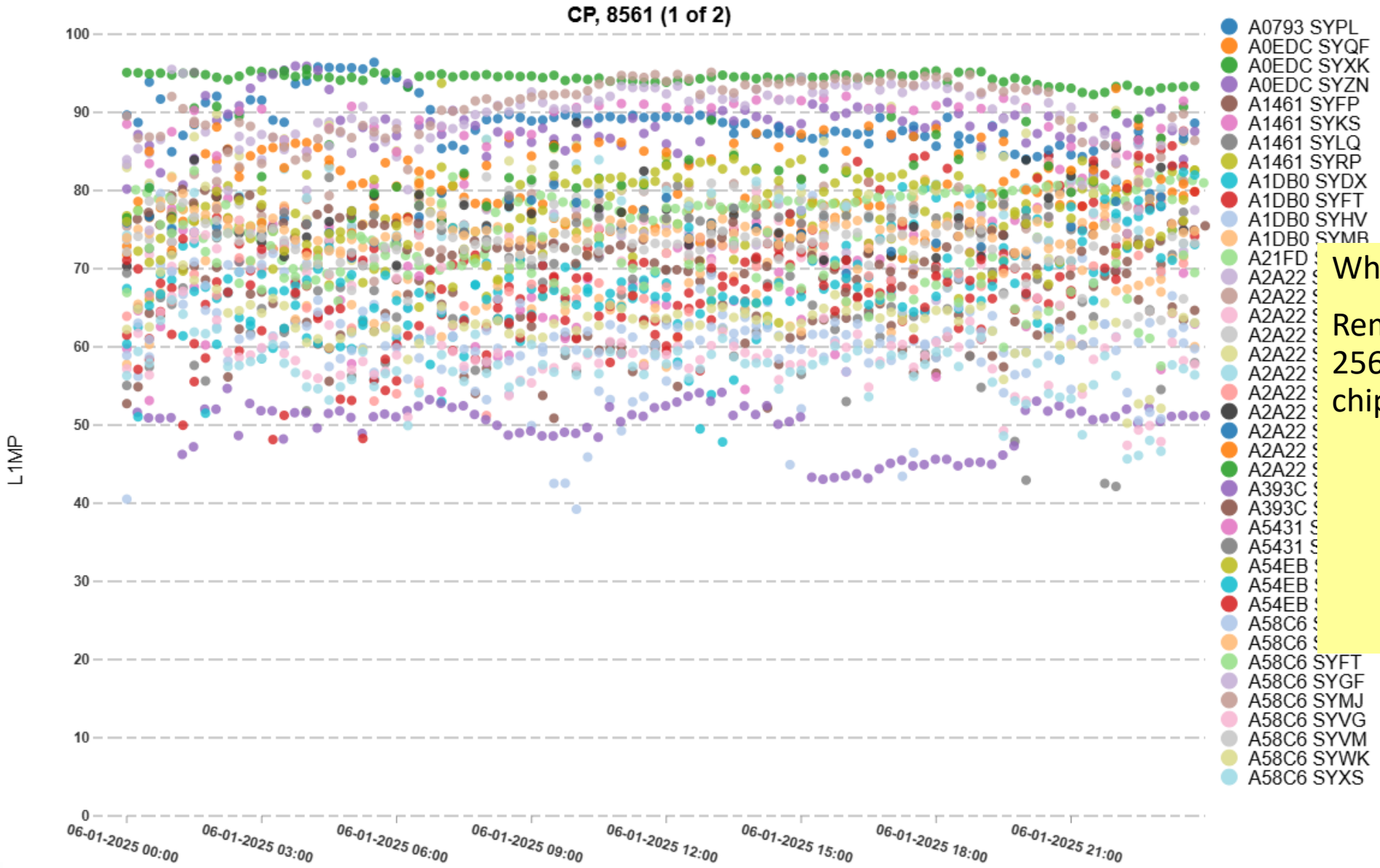


And here are a bunch of z16 systems!

Not surprising that larger L2 = more L1 misses satisfied in L2

Percent of L1 & L2 Misses Satisfied in L3

By z/OS Hardware Model

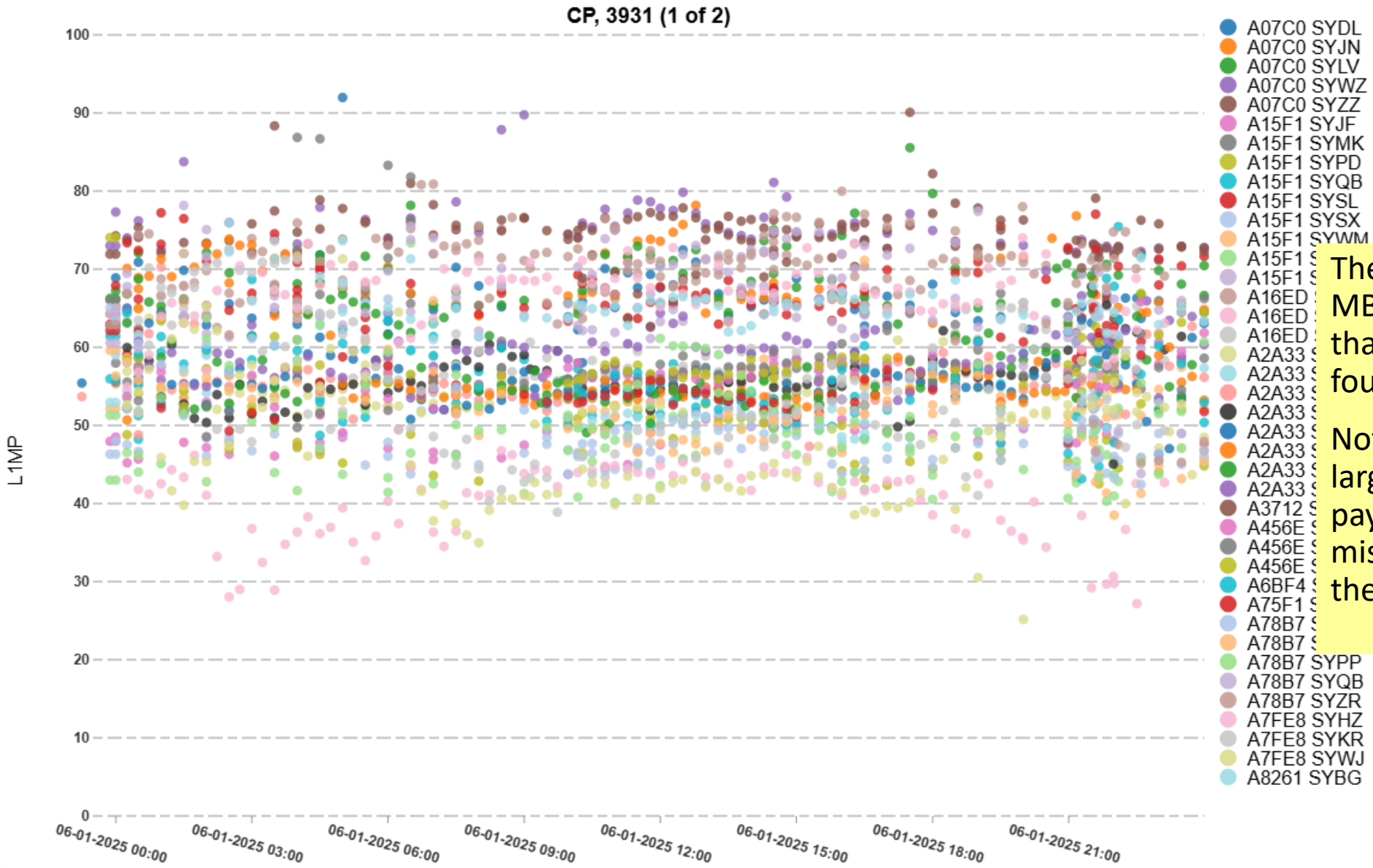


What about L2 misses?

Remember the z15 had 256 MB of L3 cache per chip.

Percent of L1 & L2 Misses Satisfied in L3

By z/OS Hardware Model



The z16 L3 is up to 224 MB so not surprising that fewer L2 misses are found in a smaller area.

Not shown here, but the larger L4 in the z16 does pay off with more L3 misses being satisfied there.

Summary

- Processors get faster over time
  - “Faster” = more useful work done per unit of time
  - “Faster” ≠ faster clock speed
- Architectural changes often more important than clock speed changes
  - At least for last several generations
  - Likely for the next ones too
- Data closer to processor = better performance
- Understanding these details is useful for understanding why some workloads may over/under perform relative to machine rating