Real-World Mainframe Performance Metrics and Observations (2026 Edition)

Scott Chapman
Enterprise Performance Strategies, Inc.
Scott.chapman@EPStrategies.com

SHARE Pittsburgh
August 2026
Session Tech158

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!)

There is a plethora of z/OS performance metrics available. Some are easily understood and their applicability is well-known. Others are not so obvious. Sometimes answering the question of "what is a good value for this metric?" can be difficult. For some metrics, there are no "right" or "wrong" values as they are workload-dependent, but it is still interesting to understand the range of values commonly seen as situations that generate metrics outside the normal range might be worth investigating. We get data from hundreds of systems every day, giving us a unique insight into the metric values achieved by real-world systems. Join Scott Chapman for a tour through some interesting z/OS performance metrics, backed by anonymized data from systems spanning multiple machine generations and z/OS versions. You may know your metrics, but where do your systems fall in the range of values that other systems experience? Come to this session and get a sense for the distribution of values seen for things like capture ratio, CPI, disk response time, and many more. Whether you find that inspiring, comforting, or just confirming, you'll leave with a clearer perspective on where your systems stand.

Agenda

- Introduction
- Discuss several interesting measurements as seen across hundreds of systems

Introduction

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

Like what you hear today?

- Free z/OS Performance Educational webinars!
  - Let us know if you want to be on our mailing list for these webinars
    - Email: contact@epstrategies.com

- If you 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

z/OS Performance workshops available

During these workshops you will be analyzing your own data!

- Essential z/OS Performance Tuning
  - March 30 – April 3, 2026 (4 days, excl Wednesday the 1st)
- WLM Performance and Re-evaluating Goals
  - July 27 – 31, 2026 (4 days, excl Wednesday the 29th)
- Parallel Sysplex and z/OS Performance Tuning
  - May 12-13 2026
- Also… please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

EPS presentations this week

What	Who	When	Where
Taming the Beast: WLM Optimization Strategies for z/OS Performance Analysts	Peter Enrico	Mon 14:30	Room 302
Real-World Mainframe Performance Metrics and Observations (2026 Edition)	Scott Chapman	Tue 09:15	Room 302
PSP: z/OS Performance Spotlight: Some Top Things You May Not Know	Peter Enrico Scott Chapman	Tue 13:15	Room 304
Db2: Understanding Db2 Usage and Measurement of WLM Enclaves	Peter Enrico	Tue 15:45	Room 318
What a z/OS Guy Learned about AWS in 12 Years	Scott Chapman	Wed 08:00	Room 309
z/OS Performance Topics: The Ones That Divide Us	Peter Enrico Scott Chapman	Thu 13:15	Room 302

What is this all about?

- We regularly make recommendations about what is expected for various measurement values but…
- For most values “good” is really a broad range of values
  - And certainly some systems run outside what we consider “good”
- While we have a sense of that range because we see data from lots of customers, customers rarely get that perspective
  - Sometimes it would be nice for customers to get a sense for where they are on the continuum
- All data herein is anonymized
  - Any resemblance to any actual names or serial numbers is coincidental (and unlikely!)

Red words are key points to pay attention to

Notes on the data

- Date on the charts will show July 1 or August 1, 2026, but that's just a normative date: not all data may have come from exactly that date
- For various reasons, the data shown herein is a subset of the data that we've seen
  - Selection of systems may not even be exactly the same between reports
- Some (many) of the reports are very "busy"
  - The intent is not to be able to read what any particular system's measurements are, but rather get a sense for the range from all the presented systems

- This is not to be considered a statistically valid sampling!
- "Your mileage will vary" (That should be very clear!)
- While some reports are broken down by z/OS version or machine type, that's mostly for reporting convenience and in most cases you can't compare the different categorization because there's vastly different workloads represented

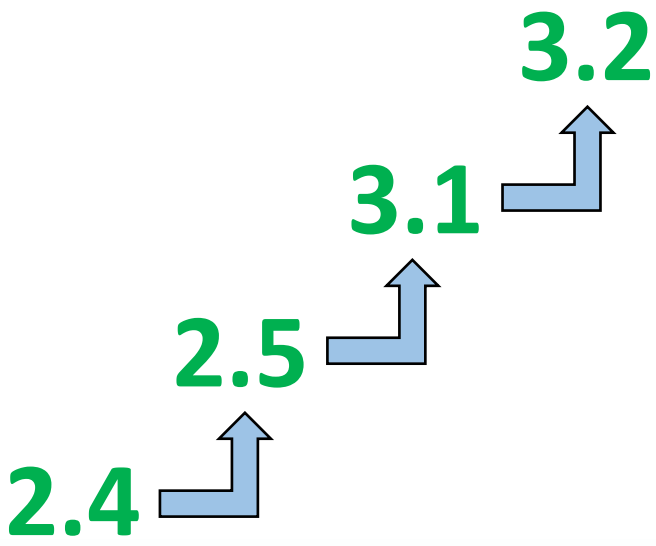
How This Presentation is Arranged

- For each measurement or group of measurements:
  - Why you care: why would you be interested in this measurement
  - The measurements themselves: usually from dozens of systems
    - The charts aren't meant to be read precisely, rather more as a point cloud to see overall trends
  - What you should do: actions you might want to take if your systems are outside of the norms

- My hope is that seeing what other systems are doing might:
  - Inspire you: to try to improve your own systems
  - Console you: you are not alone in not achieving "optimal" numbers

How current are you?

Machine and z/OS versions

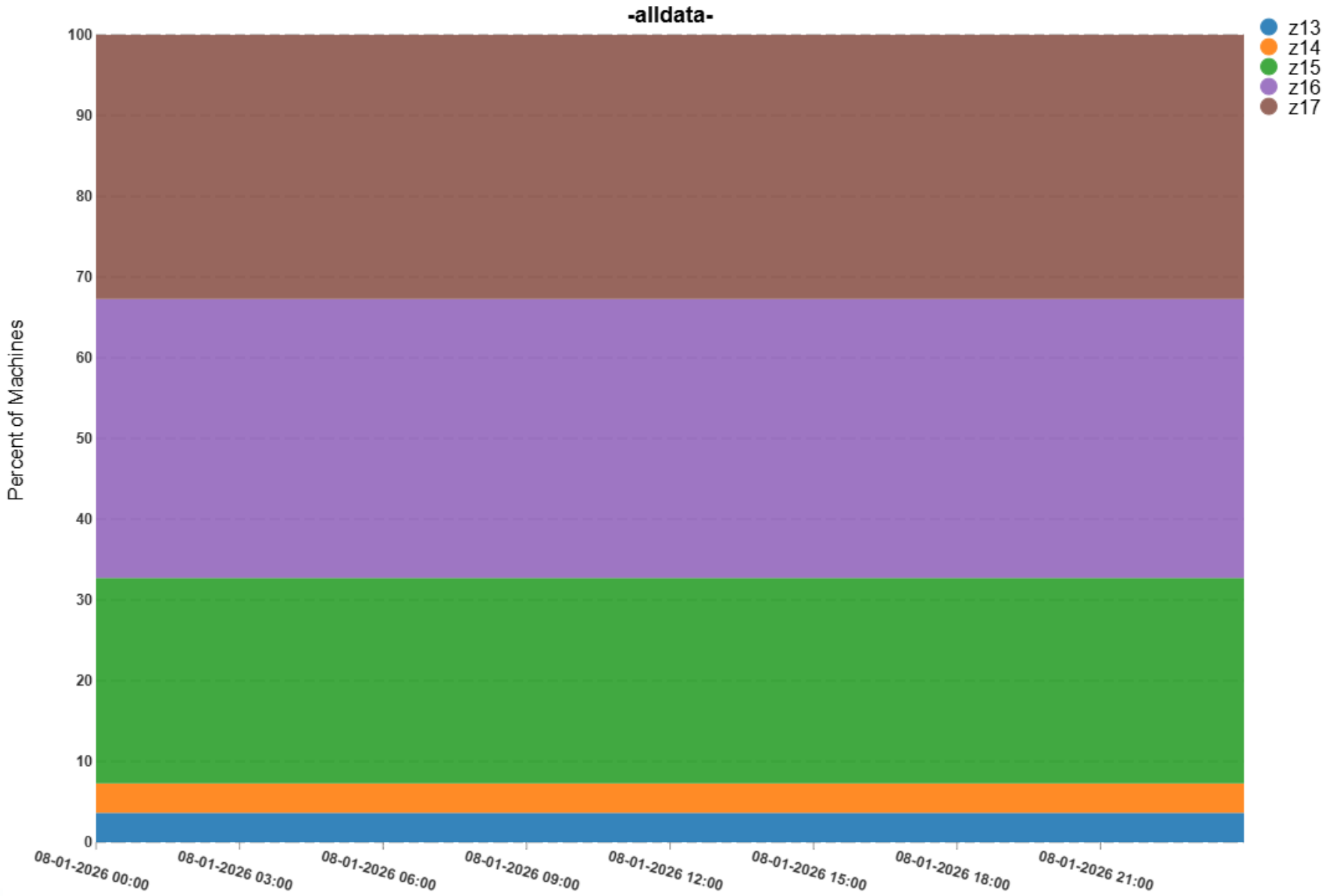


Machine Generations and z/OS versions: Why you care

- Maintenance on older hardware can be expensive
  - But you can always go time and material, so this may be simply a risk point
- IBM software discounts between machine generations can be significant
- Advances in the hardware may make applications/system more efficient, possibly lowering consumed MSUs
- z/OS more aggressively requiring more recent hardware
  - z/OS 2.2 requires at least z10 hardware (Extended Support date: 2020-09-30)
  - z/OS 2.3 and 2.4 requires at least z12 hardware (ES: 2022-09-30, 2024-09-30)
  - z/OS 2.5 requires at least z13 hardware (ES: 2026-09-30)
  - z/OS 3.1 requires at least z14 hardware
  - z/OS 3.2 requires at least z15 hardware
  - z/OS goes to extended support ("extra cost support") basically 5 years after release

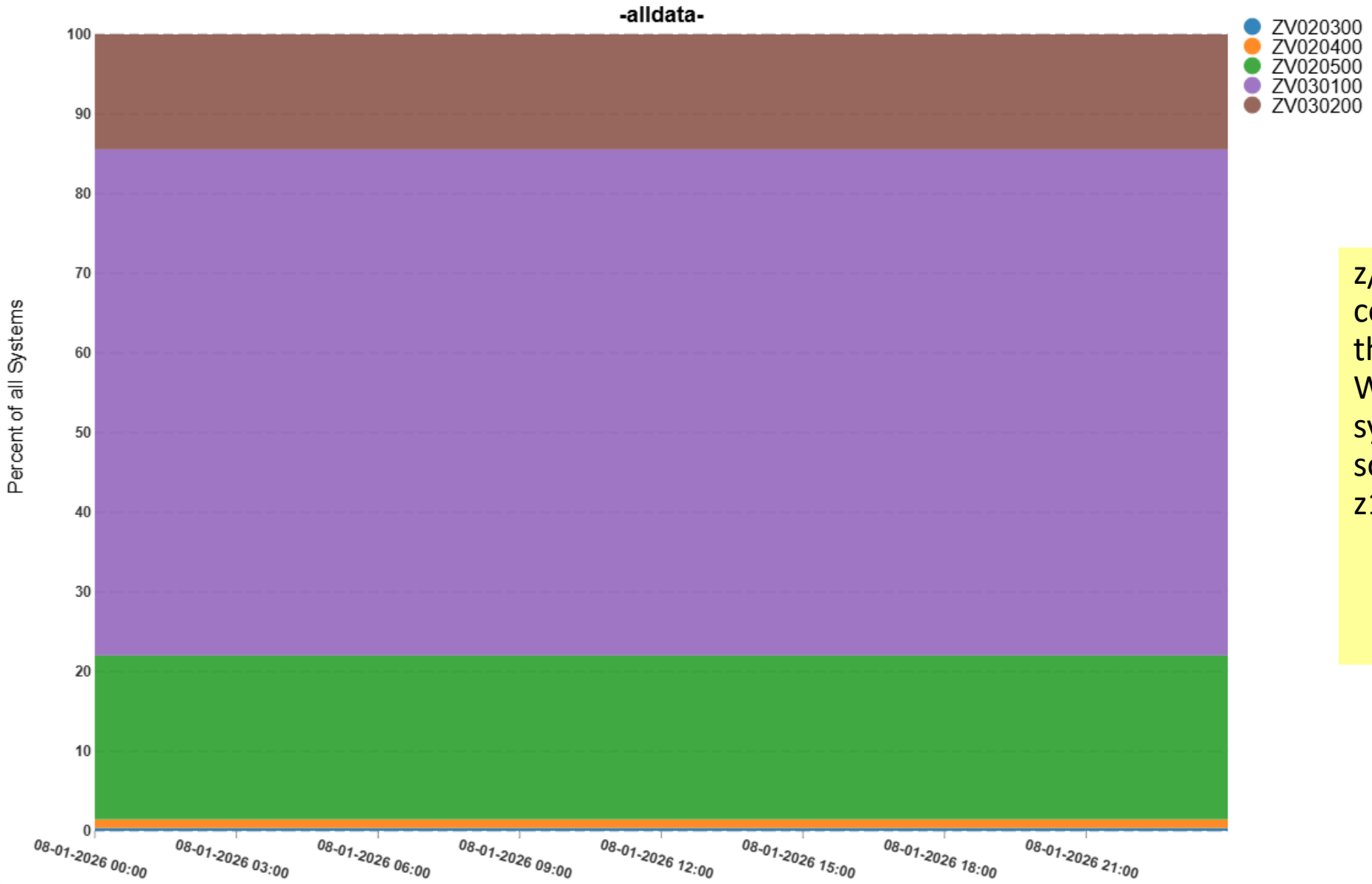
There may be a pattern emerging here!

Machine Generations



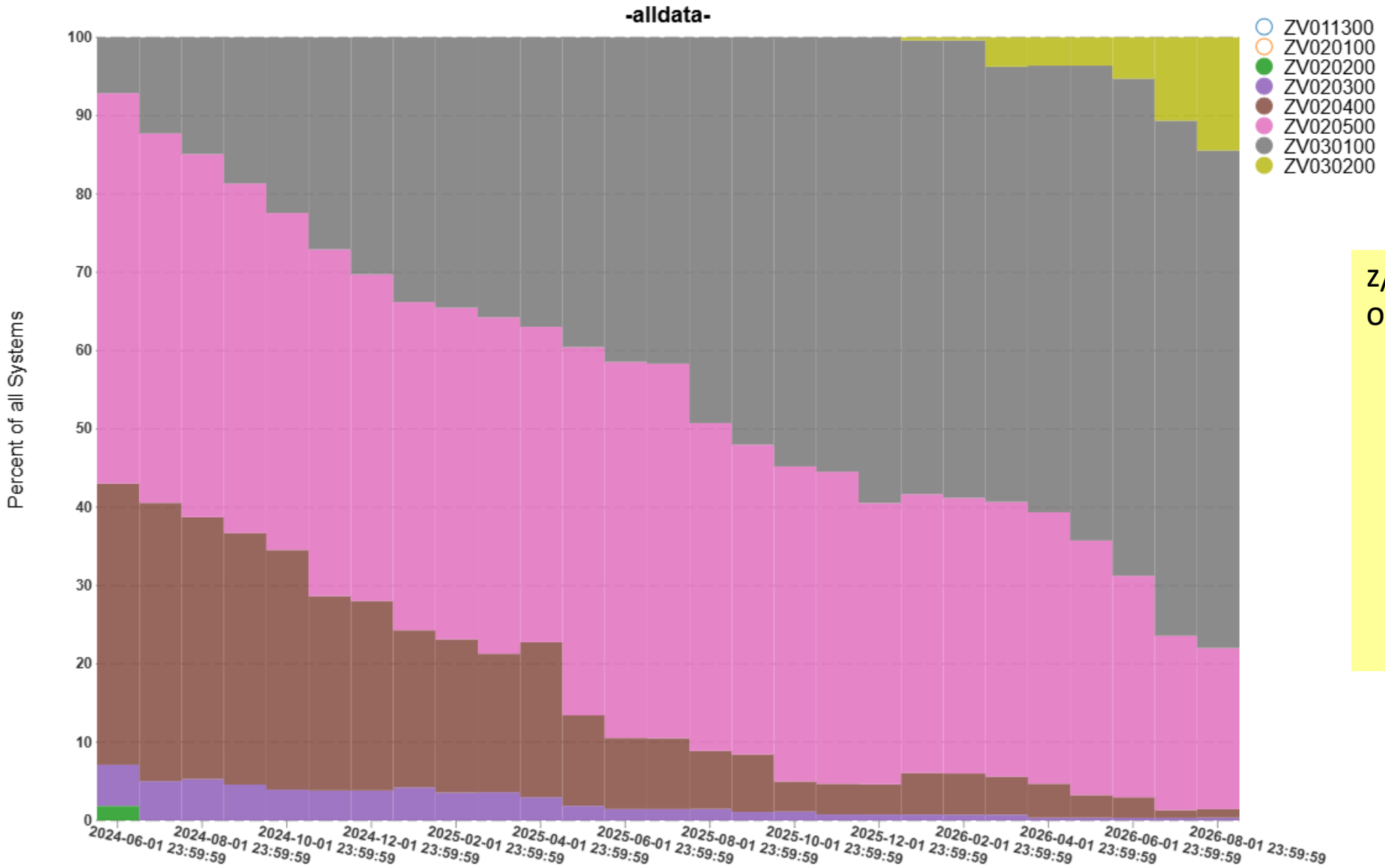
z17 adoption continuing, most customers on z15 or later

z/OS Versions



z/OS 3.1 by far the most common z/OS version that we see right now. Would expect those 2.5 systems to get to 3.2 soon unless they're on a z13 or z14.

z/OS Versions



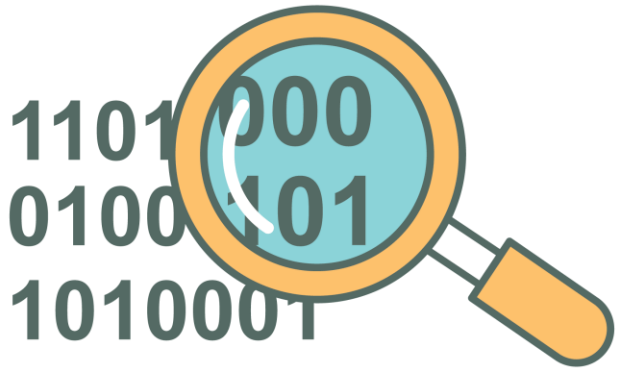
z/OS version trajectory over the past 2 years

Machine Generations and z/OS versions: What should you do?

- If you're not on z/OS 3.x, hopefully you're transitioning there soon
- Start making plans for getting to at least z15 levels of hardware if you're not already
  - Most customers already there
- Consider the "skip a generation" upgrade pattern to optimize software discounts and availability vs. hardware costs
  - Most customers are within 2 generations of current
  - Old machines may not save you as much money as you think, despite being pretty darn cheap
  - Software generally costs more than hardware so optimizing for hardware may be questionable

Wherein Scott preaches about SMF records

SMF & RMF Details

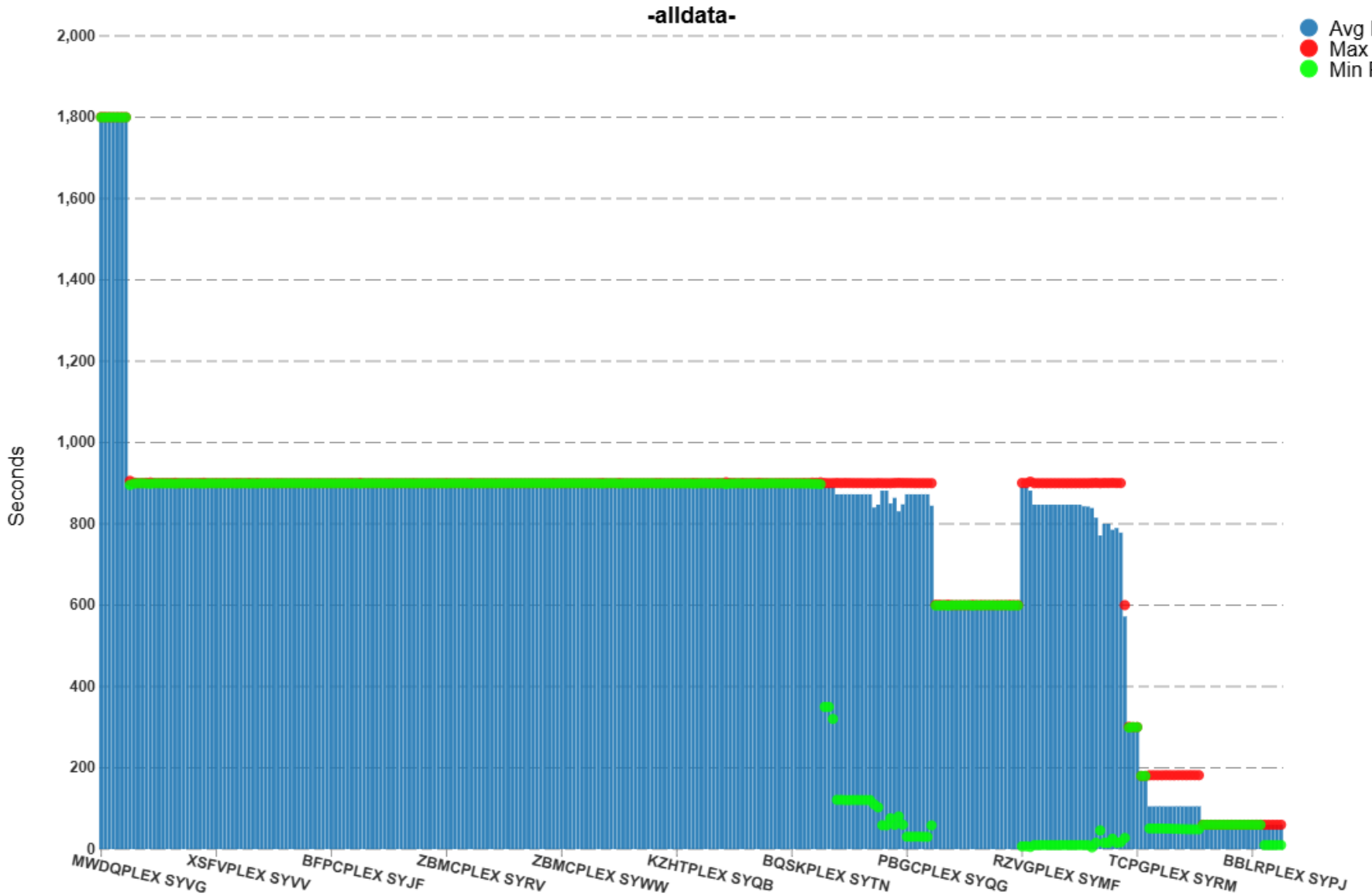


SMF Data & RMF/CMF Intervals: Why you care

- SMF data, particularly that recorded by RMF/CMF is, key to managing your system
- Having the data readily available after an incident makes problem diagnosis easier
- But some worry about the size of the recorded data
  - Don't just look at record counts: look at total bytes (e.g. avg record length * records)
  - Mainframes are really good at processing data!
- Old concerns about data size lead some to not record certain SMF records
  - SMF type 99s in particular
- Interval length should be based on your sensitivity to performance problems
  - 15 minutes (900 seconds) is the maximum we recommend

RMF/CMF Interval

Average/Max/Min For Day



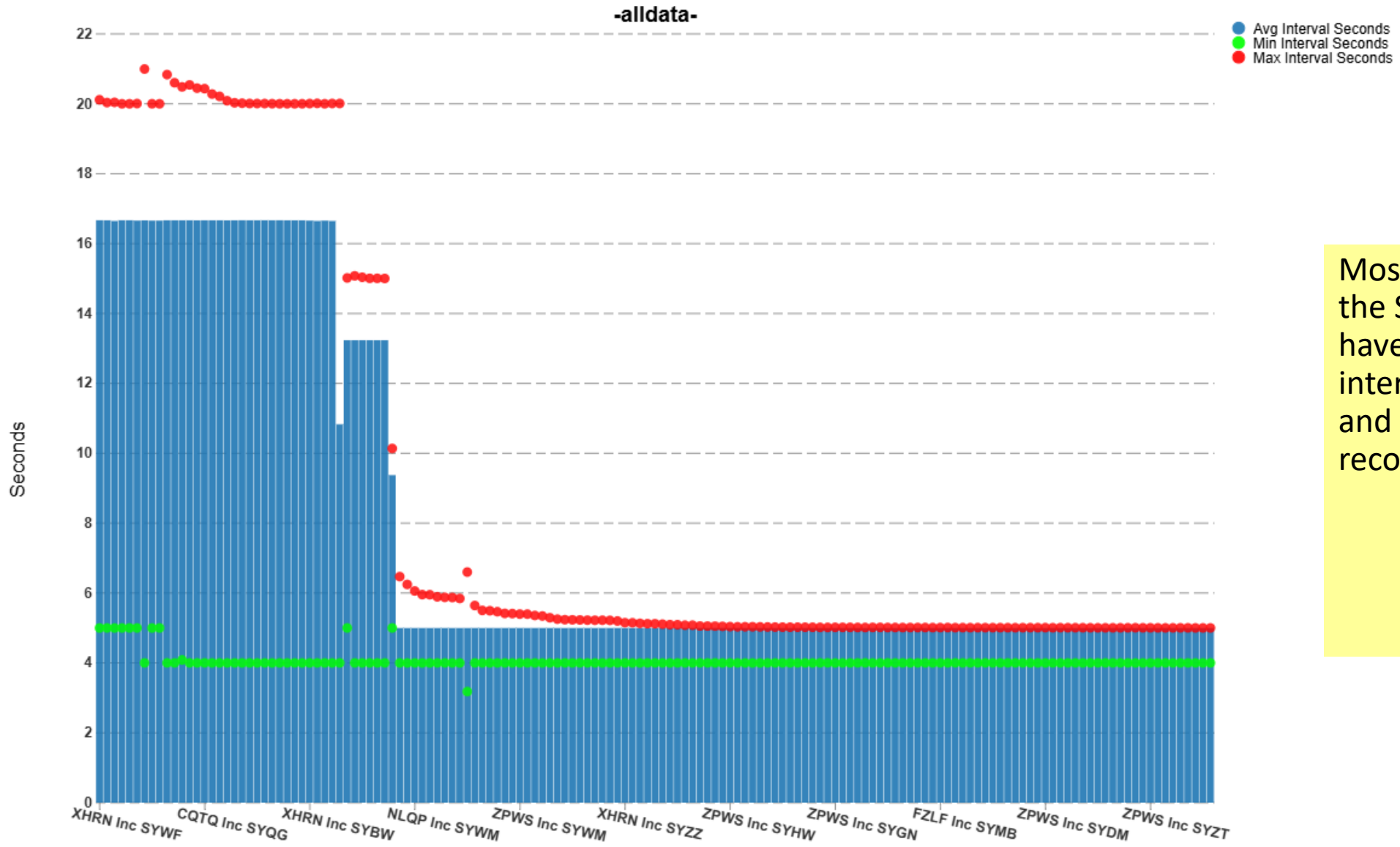
About 80% of systems are on 15 minute intervals, with smaller numbers on 5 or 10 minute intervals.

We would generally not recommend over 15 or under 5.

Sub-minute details

- SMF 99 have some really interesting data in them
  - Minimally, record subtypes 6, 10, 11, 12, 14
  - Ideally, record *all* subtypes except 13 (undocumented, ~150MB/sys/day)
- SMF 98 also can be useful in some situations
  - Some data is poorly documented
  - Some useful data is recorded in a confusing manner (Address space level data)
- See also:
  - https://www.pivotor.com/library/content/Enrico.Using.HF.SMF.98.99.SHARE.S2022.pdf
  - https://www.pivotor.com/library/content/Chapman_PinpointingPerformanceSMF9899.pdf
  - https://www.pivotor.com/library/content/Chapman-UnderstandingSMF98_AddressSpaces.pdf

SMF 98 Interval Seconds



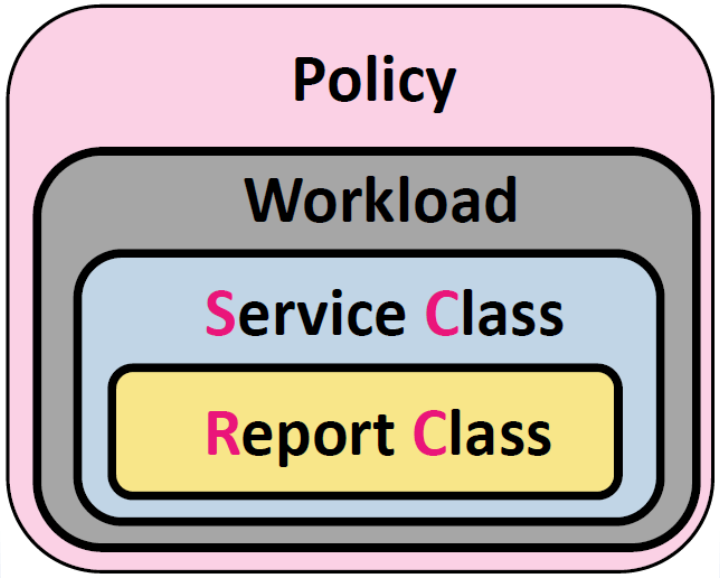
Most systems recording the SMF 98 HTS data do have it on a 5 second interval. (That is IBM's and our current recommendation.)

SMF Data & RMF/CMF Intervals : What you should do

- If your RMF interval is > 900 seconds, change to 900 seconds
  - Possibly consider shorter intervals
- If you're RMF and SMF intervals are not synced, make them so
  - Makes it easier (possible) to compare the non-RMF records to the RMF records
  - Unless you're using very short RMF intervals
- Record the SMF 98 records
  - Ideally at 5 second interval (see HFTSINTVL in SMFPRMxx)
- Record at least these 99 subtypes:
  - 6: WLM Service Class Period Summary (10 seconds)
  - 10: Dynamic speed change (should be zero of these, so why not record them?)
  - 11: Group Capacity Limits (300 seconds)
  - 12: HiperDispatch Intervals (2 seconds)
  - 14: HiperDispatch Topology (300 seconds or whenever a change occurs)
  - All of these might total an extra 100-150MB of data per system per day but can be very interesting or important when doing a performance analysis

Worry less about your SC count and define more RCs

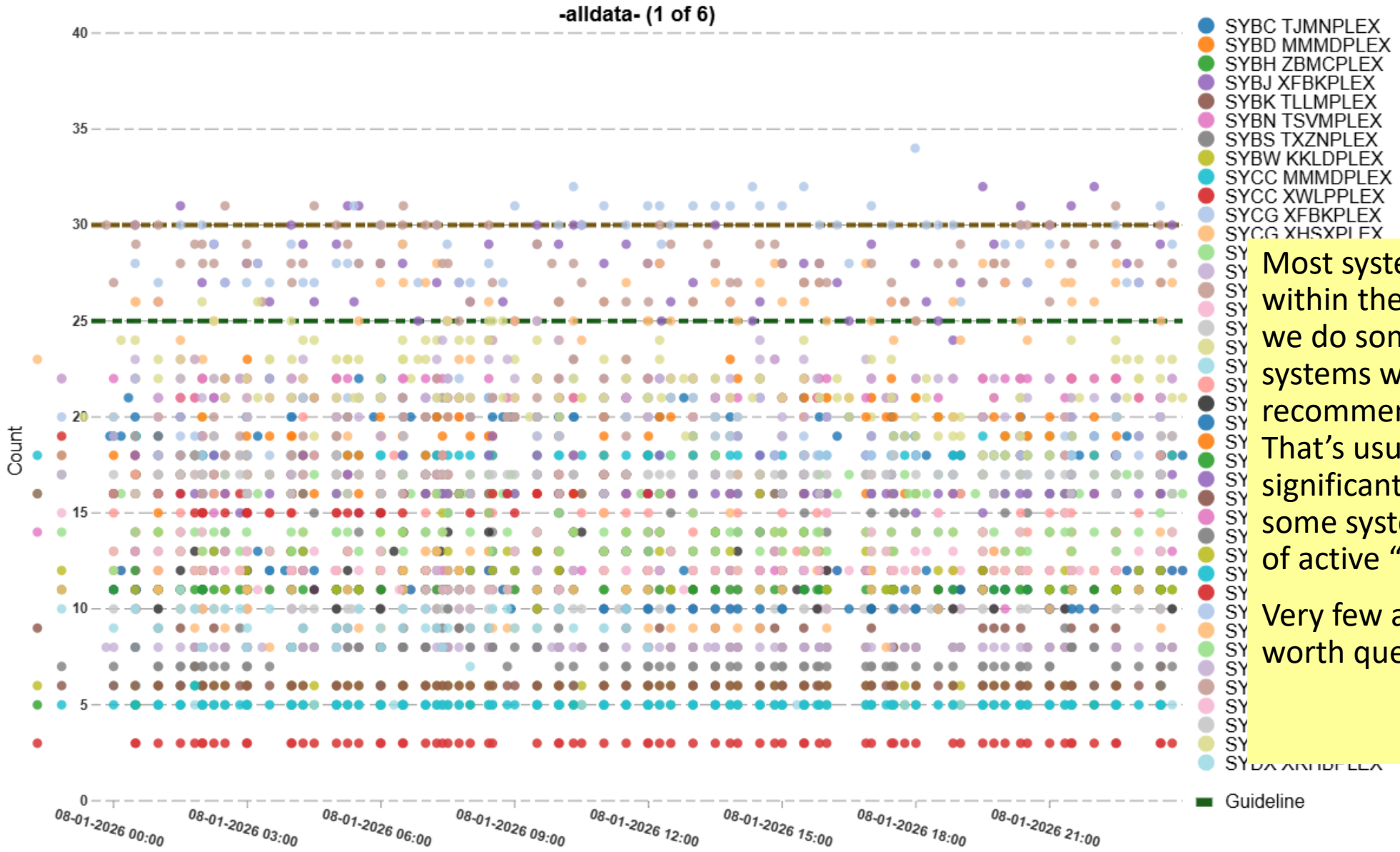
WLM Service and Report Classes



Active Service Class Periods: Why you care

- General recommendation: Have no more than 25-30 active service class periods per system during periods of interest
- The primary concern is WLM's ability to respond to a changing environment in a timely fashion
  - Remember that WLM only attempts to help a single SCP every 10 seconds
- In some cases, breaking the work down into too many SCPs makes the work less manageable too due to activity rates being too low
  - Too few using / delay samples make velocities sporadic
  - For RT goals: need to have transactions completing, ideally at least multiple/minute

Active Service Class Periods



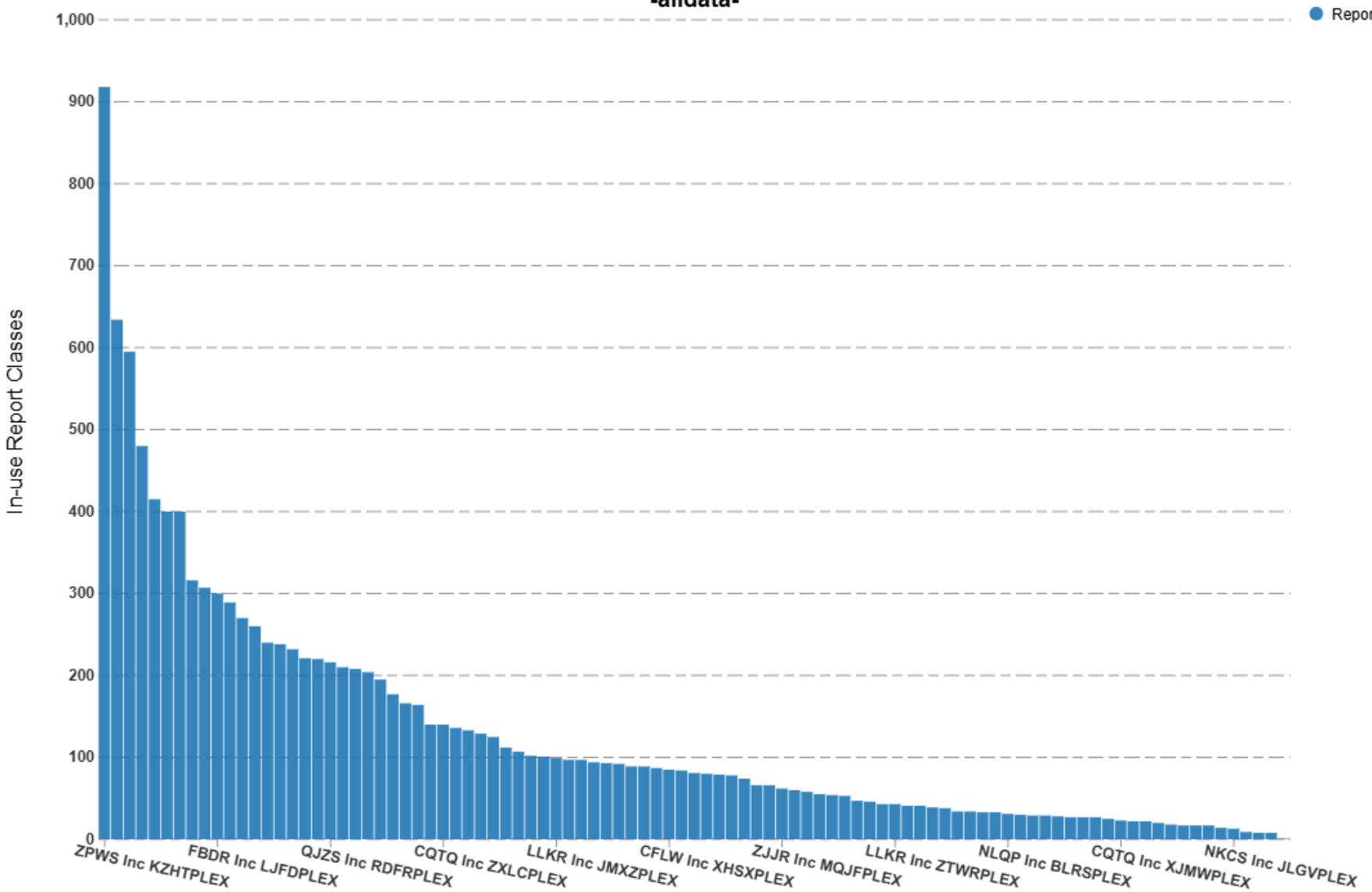
Most systems are well within the guidelines, but we do sometimes see systems with more than the recommended threshold. That's usually not a significant problem. (Indeed some systems have dozens of active "internal" SCPs!)

Very few active SCPs is also worth questioning.

WLM Report Classes

By Sysplex

-alldata-



Having hundreds of report classes lets you understand the work at a more granular level, but less than half of the plexes use more than 100 report classes.

Active Service Class Periods : What you should do

- If you have more than 30 active SCPs, then you may want to re-evaluate your policy to see if you're breaking things up too finely
- OTOH, if you have less than 10 active SCPs, I might question that as well
  - It's possible that the system is simply dedicated to one kind of production work only
  - But if you're trying to be overly conservative with your number of SCPs, that's probably not good either
- Most systems seem to be able to run fine within the recommended SCP guideline, and occasionally running above is not a huge problem

Report classes

- Use them!
- Can be very useful for reporting purposes
  - Break workload up by application / business unit / whatever you like
  - Very useful for DDF, for example
  - Could theoretically make chargeback run off from just SMF 72 records(!)
    - (With new support that gets CICS transaction CPU time into those records)
- No measurable penalty for having many report classes, other than the size of the RMF data produced
  - But this RMF data is probably small compared to DB2, CICS, Websphere transactional SMF data
- Corollary: consider using workloads too, for similar benefits

How busy is too busy?

CEC Busy

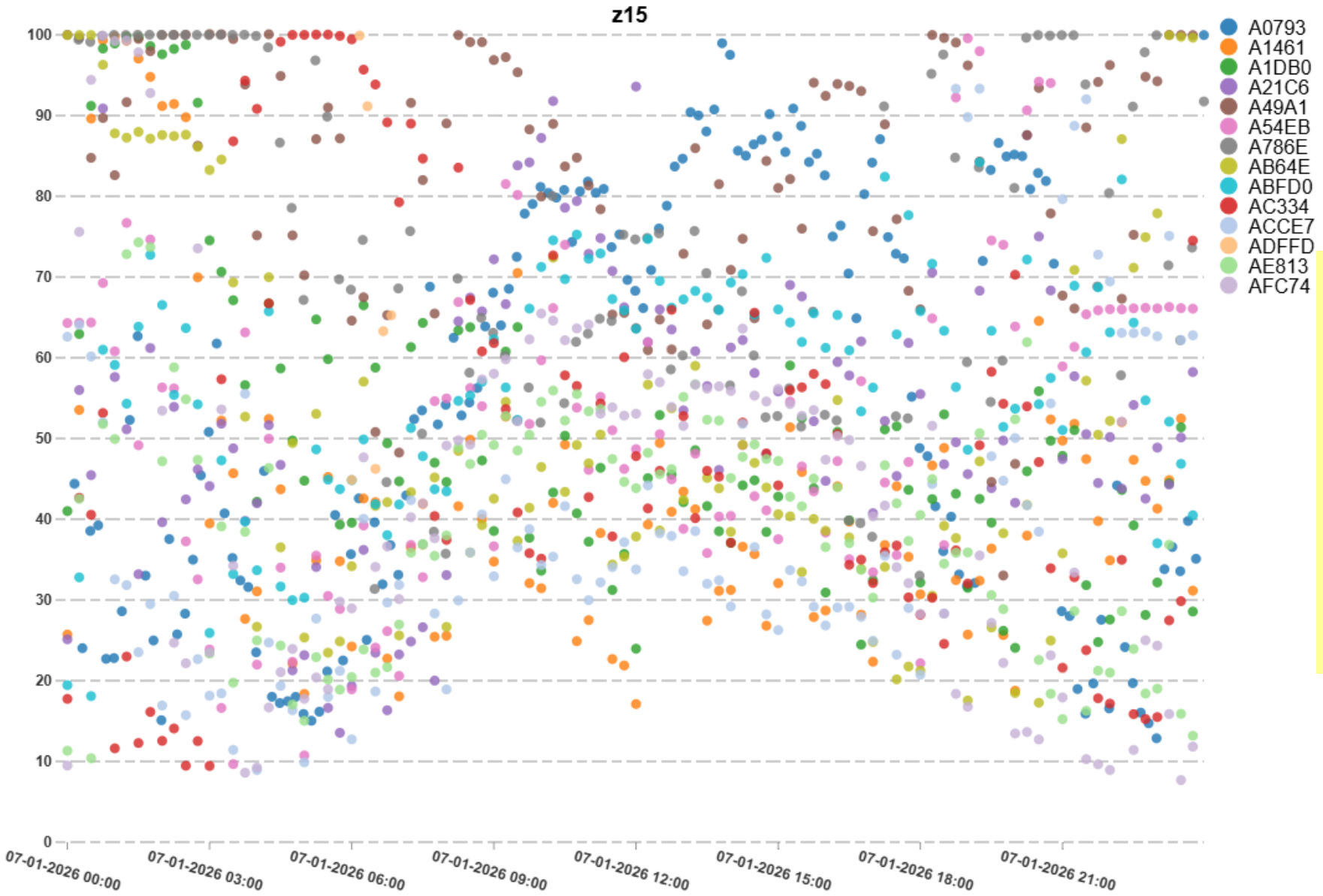
CEC Busy: Why you care

- Can you run your z/OS machine at 100% busy
  - (yes)
- Should you run your z/OS machine at 100% busy
  - (probably not)
- What about zIIPs? Should you keep those below xx%?
  - (see above)

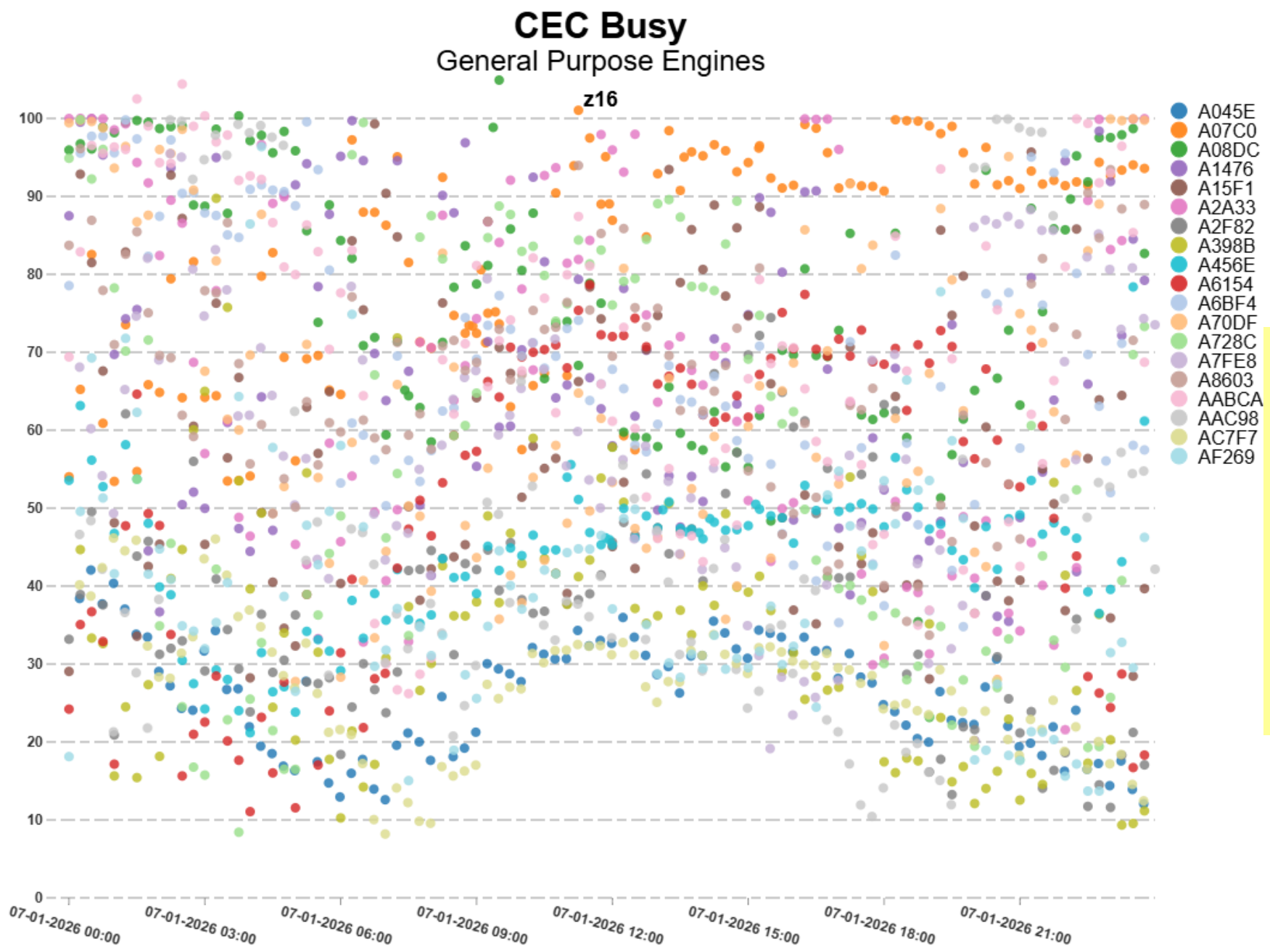
- So what are sites doing?

CEC Busy

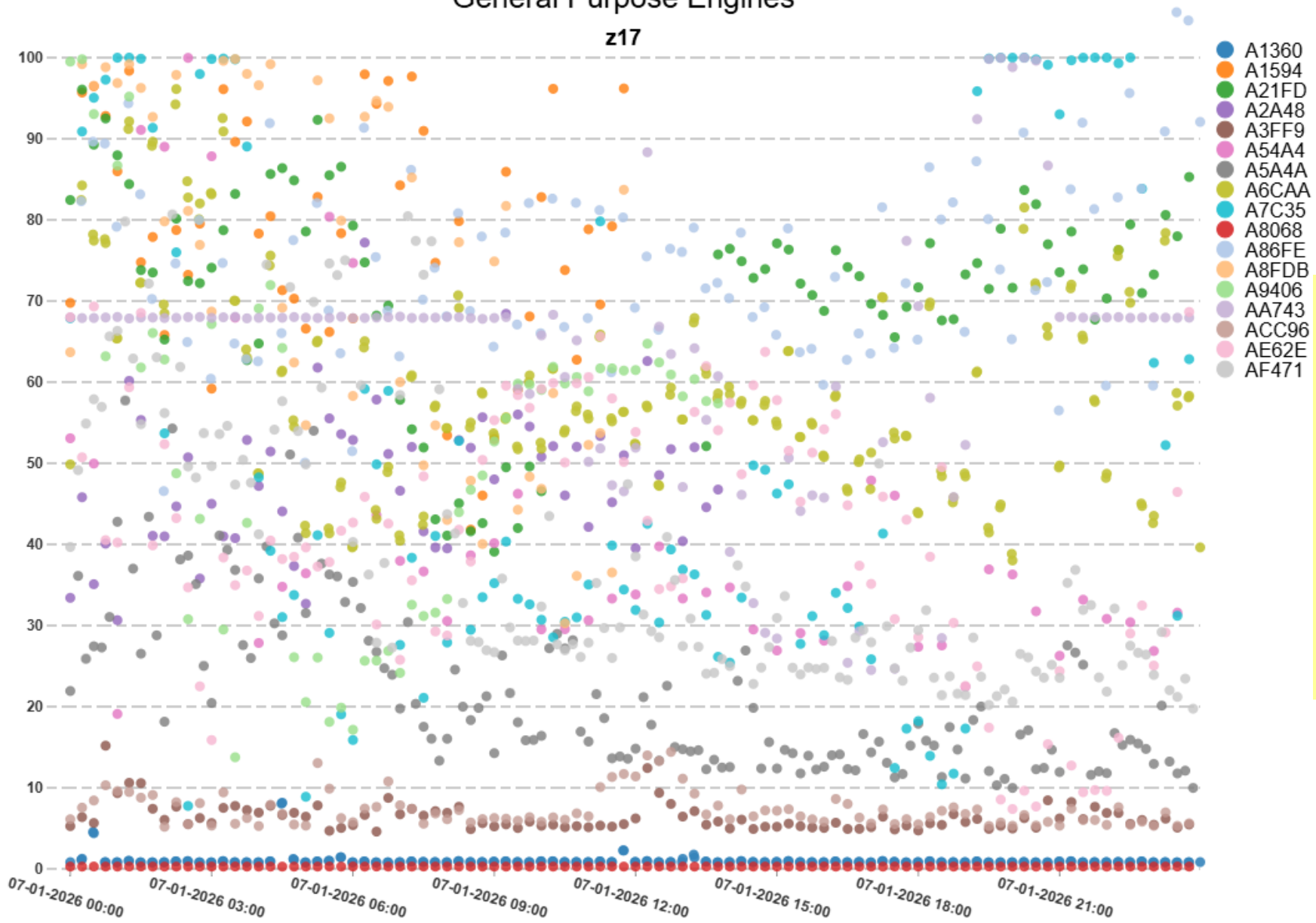
General Purpose Engines



Quite a variety of utilization levels on these z15s

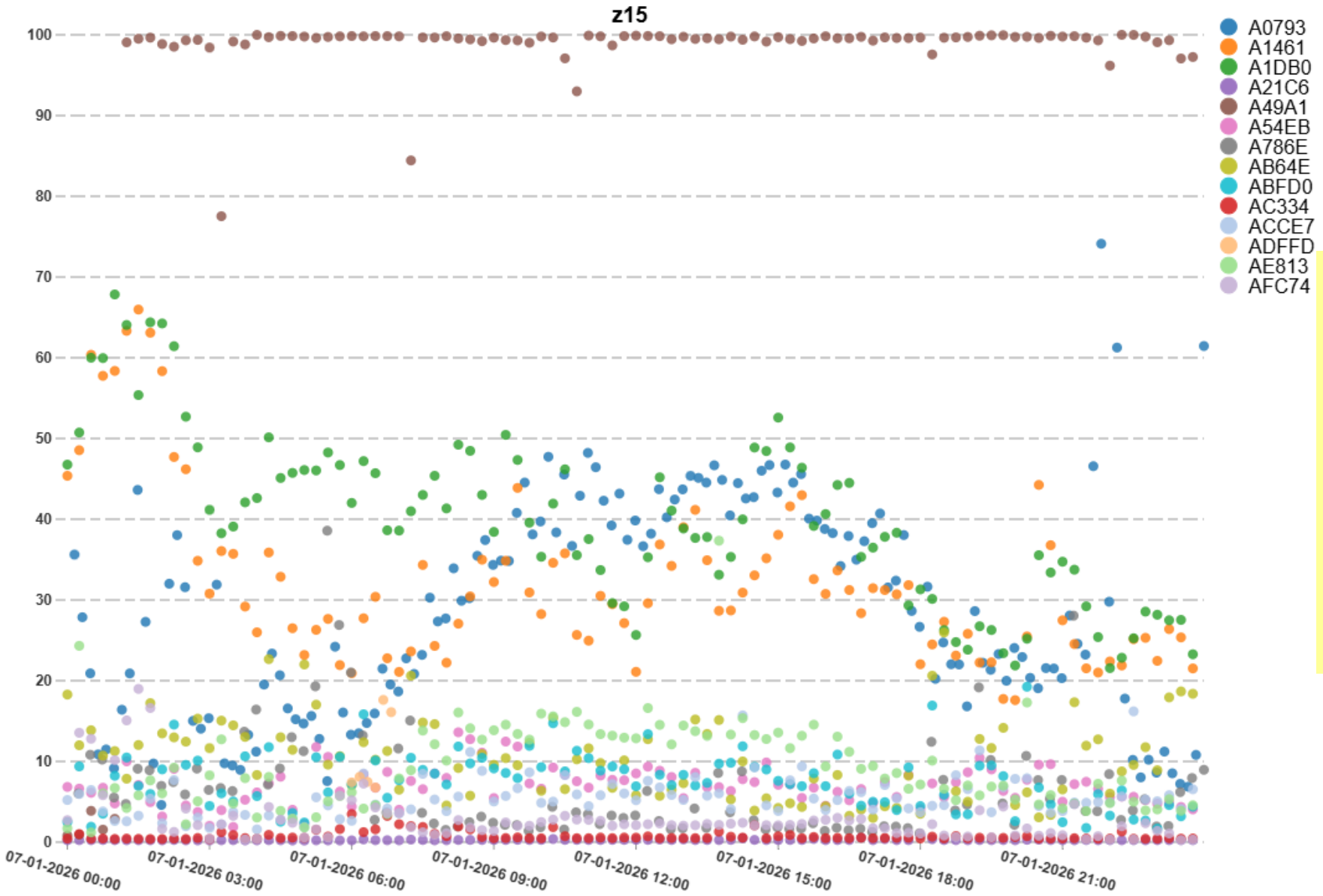


These z16s are even more varied



Wonder if those almost unused z17s are in the process of being migrated to.

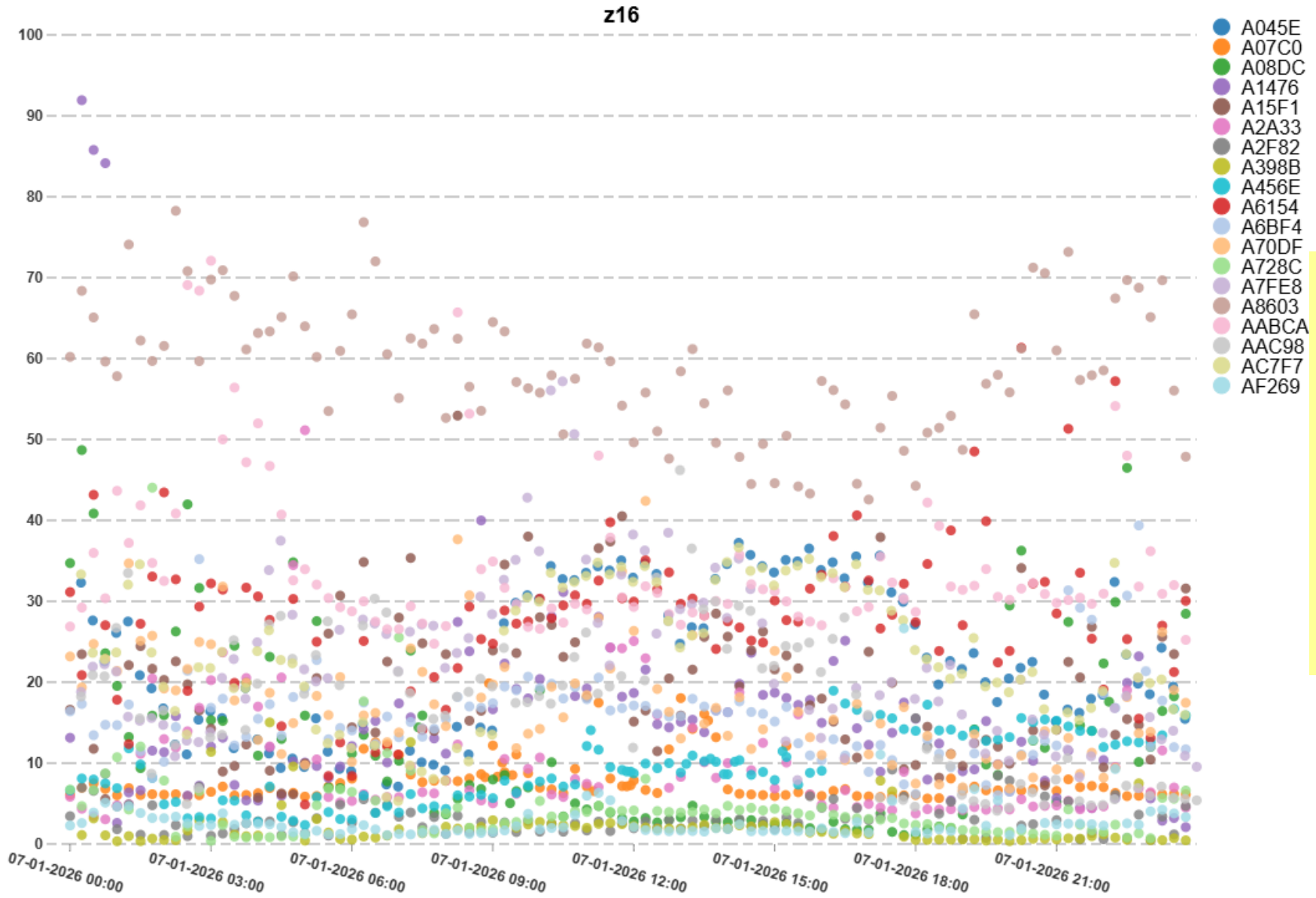
CEC Busy - zIIPs



z15 zIIP utilization

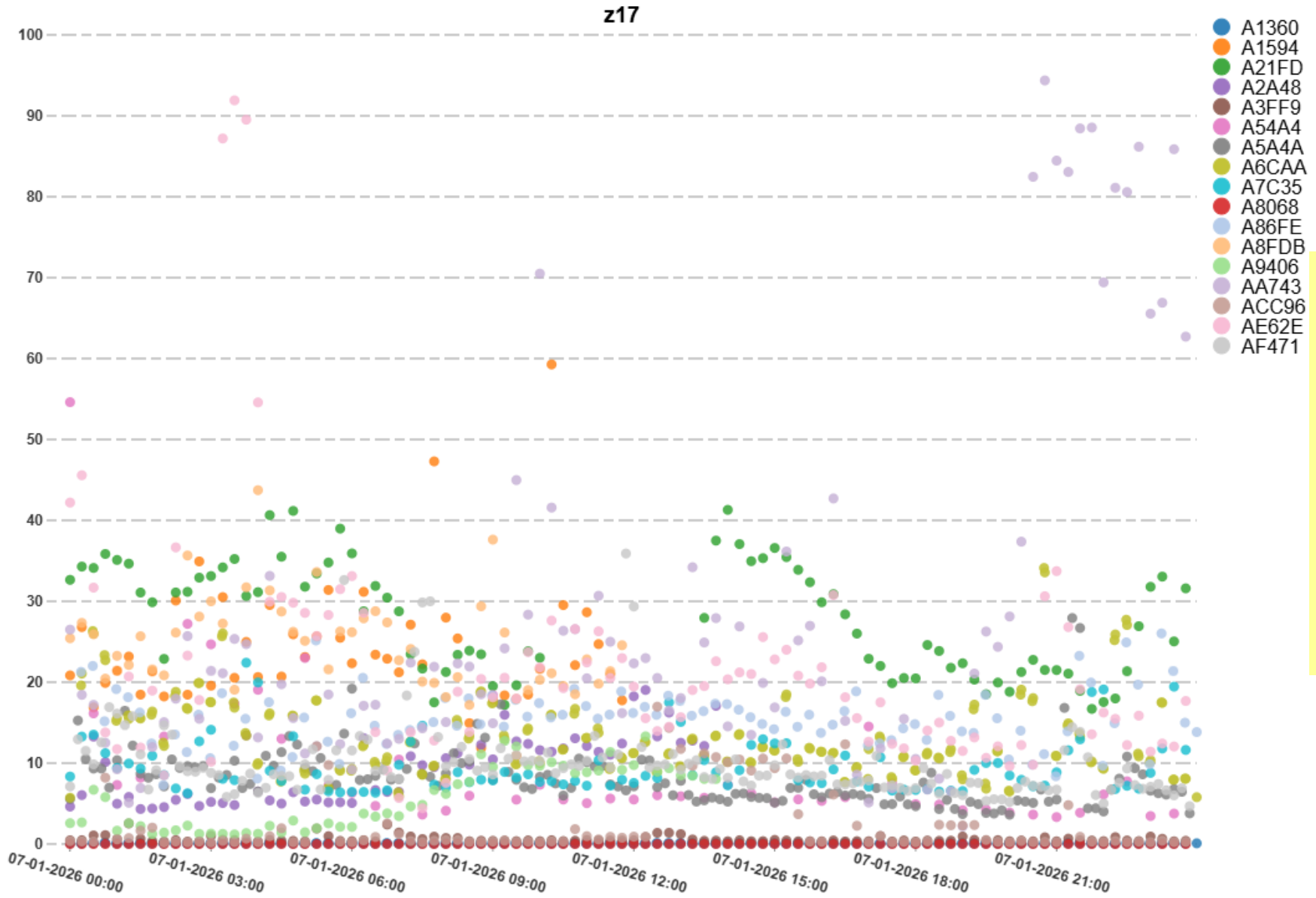
One machine clearly is driving it's zIIPs hard! But most aren't.

CEC Busy - zIIPs



z16 zIIPs

CEC Busy - zIIPs



z17 zIIPs

It may be that systems are tending to run their zIIPs less busy over time. Remember zIIPs run full speed and so generally get more capacity each upgrade regardless of specific need.

CEC Busy: What you should do

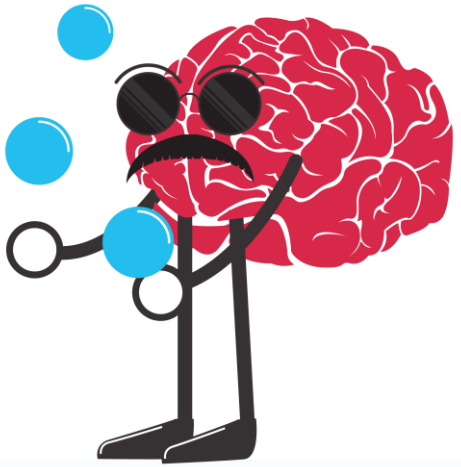
- Don’t be afraid to run less than 100% busy
  - There are efficiency and performance benefits from running less than 100% busy
  - When you’re looking at 15 minute intervals, don’t forget those are averages over the 15 minutes: there will be time periods within that 15 minute interval where the machine is busier
- zIIPs are often run less busy than GCPs, but they also can be pushed to high utilization levels
  - There is of course the potential for work to cross over and run on the GPs
  - Controls do exist to help limit this if need be
- Remember queuing theory though: performance impact of being busy is greater when you have fewer CPs
  - Another good reason to consider more/slower GCPs!

More thoughts about running at 100% busy

- Just because you can do something doesn't mean you should!
- Running 100% busy made some financial sense when your software charges were based on the installed machine capacity
  - Today most customers are on some sort of sub-capacity agreement
  - Get your ISVs on sub-capacity agreements if they aren't already
- Today focus should be on consuming least amount of MSUs while getting the work done
  - This needs to be a peak R4HA analysis
  - For TFP sites all time periods count
  - Cache contention at higher utilization levels may mean more net MSUs consumed than if you installed more capacity and ran at lower utilization levels
    - CPU time of "stable" workload increases at higher utilization levels
  - More/slower processors are almost always more efficient (in total) than fewer/faster!

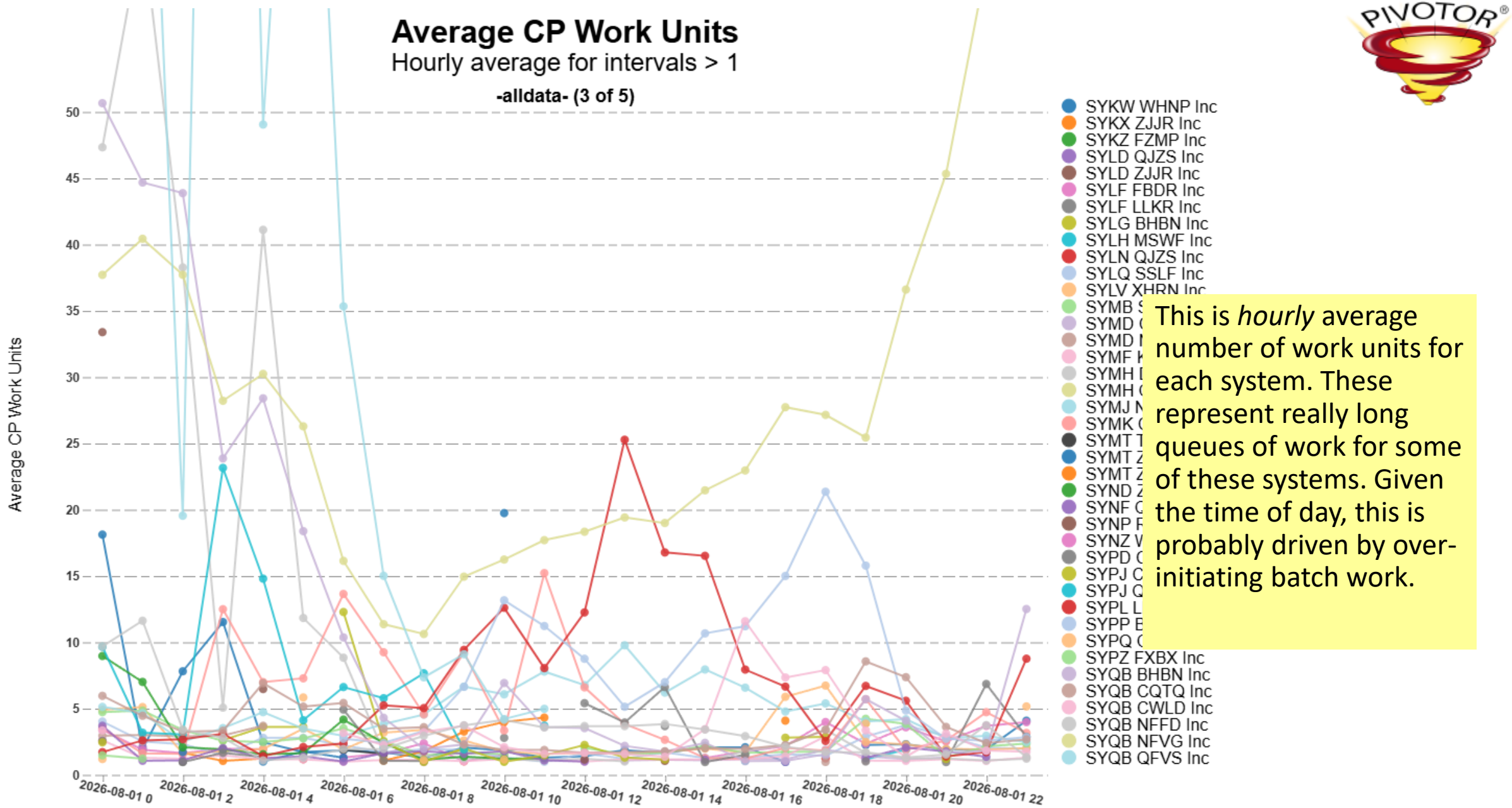
How many balls can you juggle?

Work Units



Work Units: Why do you care

- Absent SMT (not available for GPs) a CPU can only be servicing one task at a time
- Note that this is a physical thing: more logical processors does not mean that your physical processors can run more tasks!
- Just like your coworker (or cat?) interrupting you, task switching hurts efficiency
- Trying to do too many things at once means everything gets short changed



This is *hourly* average number of work units for each system. These represent really long queues of work for some of these systems. Given the time of day, this is probably driven by over-initiating batch work.

Work Units: What you should do

- Don’t over-initiate work!
  - This is a relatively common problem we see in batch windows
- Consider more/slower CPs
  - Fast CPs shared among many LPARs means at busy times, they’re really slow CPs from the perspective of the individual LPARs
  - More/faster is always better for performance, maybe not for financials
- Make sure your WLM policy is well thought-out so your loved ones suffer least

Numbers for geeks

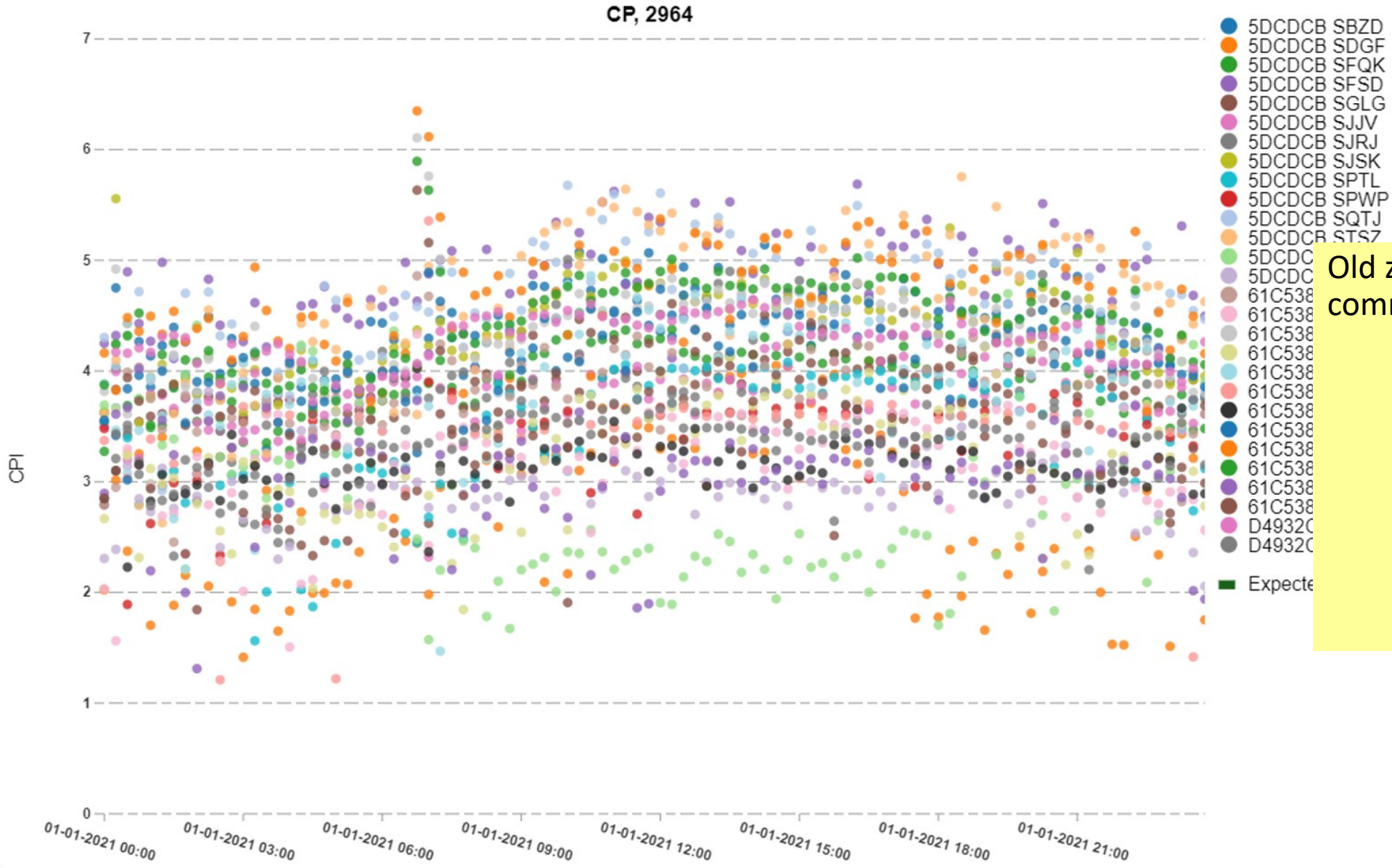
Hardware Instrumentation Services

Hardware Instrumentation Services: Why you care

- Collecting the SMF 113 HIS records is mandatory for doing proper planning for a new processor
- The information is interesting to understand the characteristics of your workload and how efficiently it's utilizing the hardware resources
- The primary numbers of interest (lower=better in all cases):
  - CPI: Cycles Per Instruction
  - L1MP: Level 1 Cache Misses per Hundred Instructions
  - RNI: Relative Nest Intensity
  - TLB Miss CPU%
    - This one is the one you're most likely to be able to (easily) influence
- These numbers are very workload dependent

Cycles Per Instruction

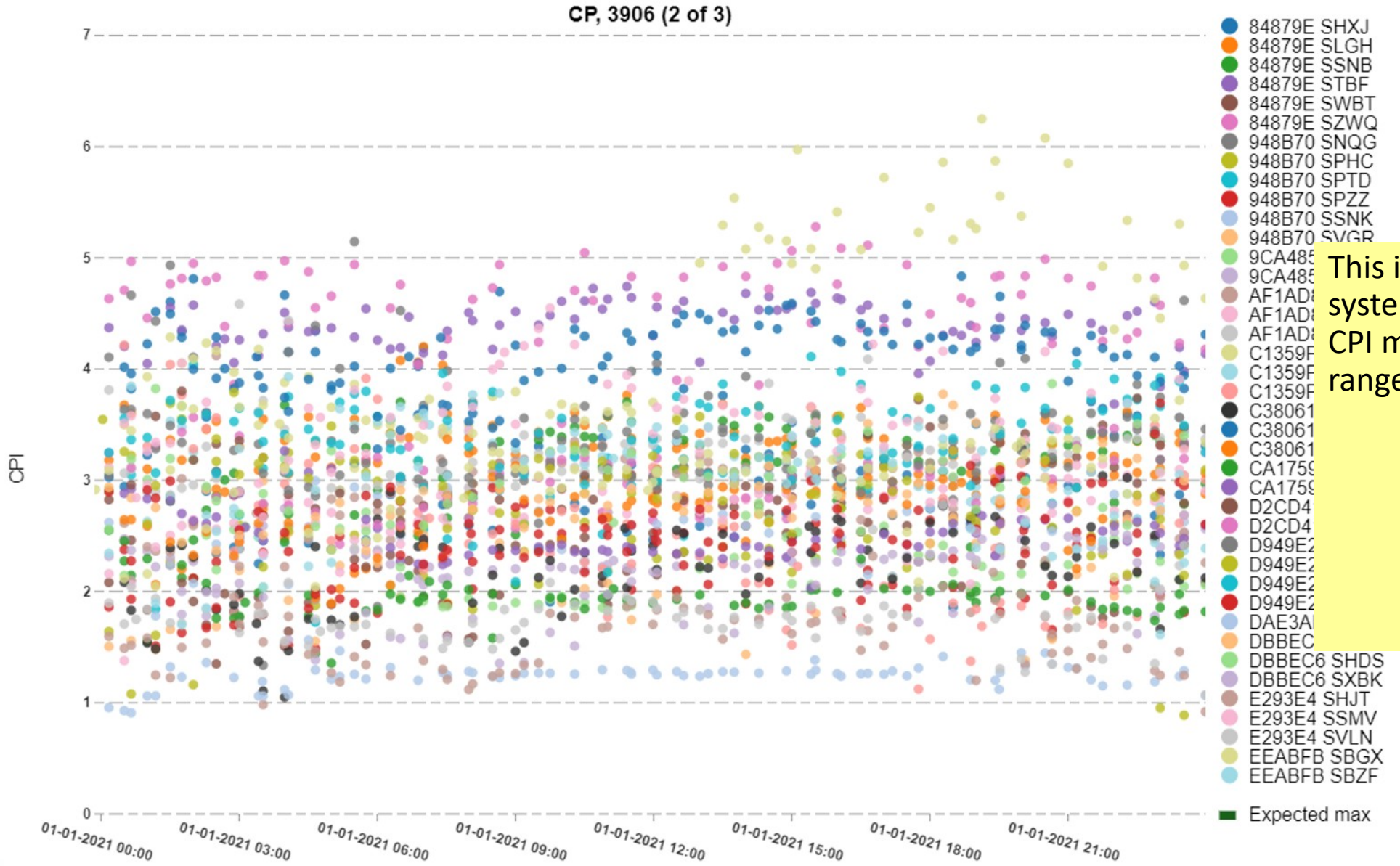
By z/OS Hardware Model



Old z13 CPI was commonly above 3

Cycles Per Instruction

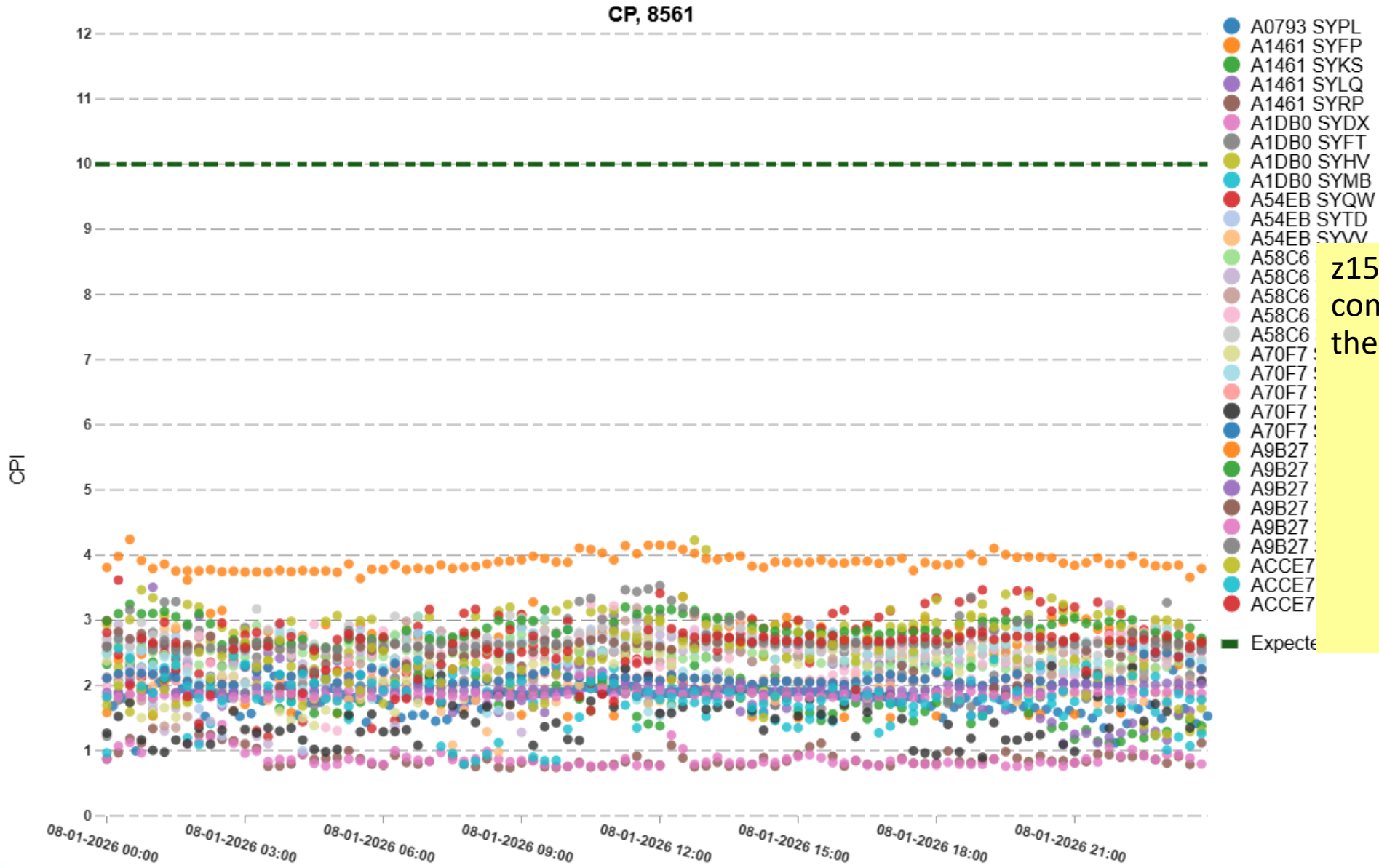
By z/OS Hardware Model



This is old data from z14 systems that showed the CPI mostly spread over a range of 2 to 4.

Cycles Per Instruction

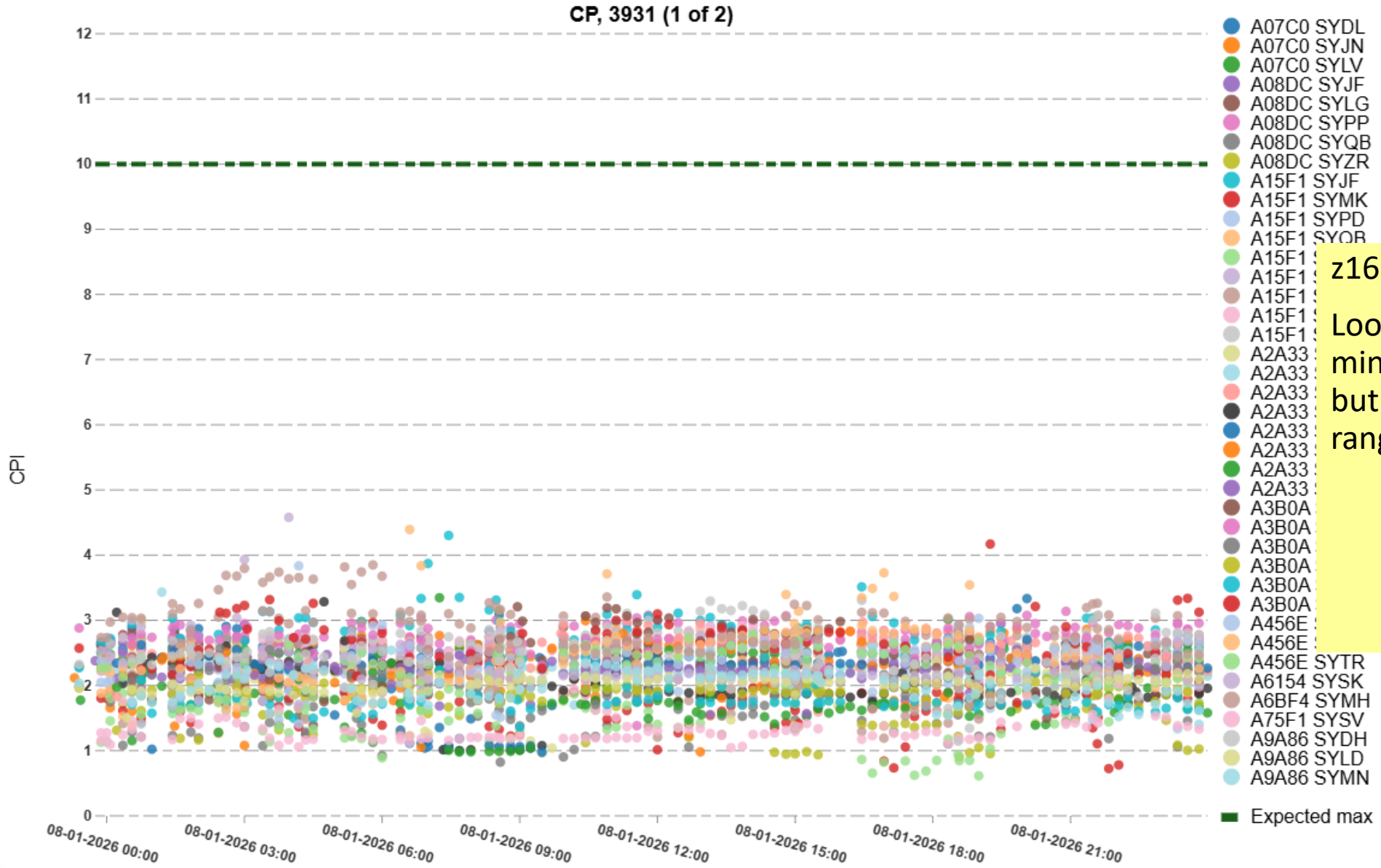
By z/OS Hardware Model



z15 systems now commonly show CPI in the 2-3 range

Cycles Per Instruction

By z/OS Hardware Model

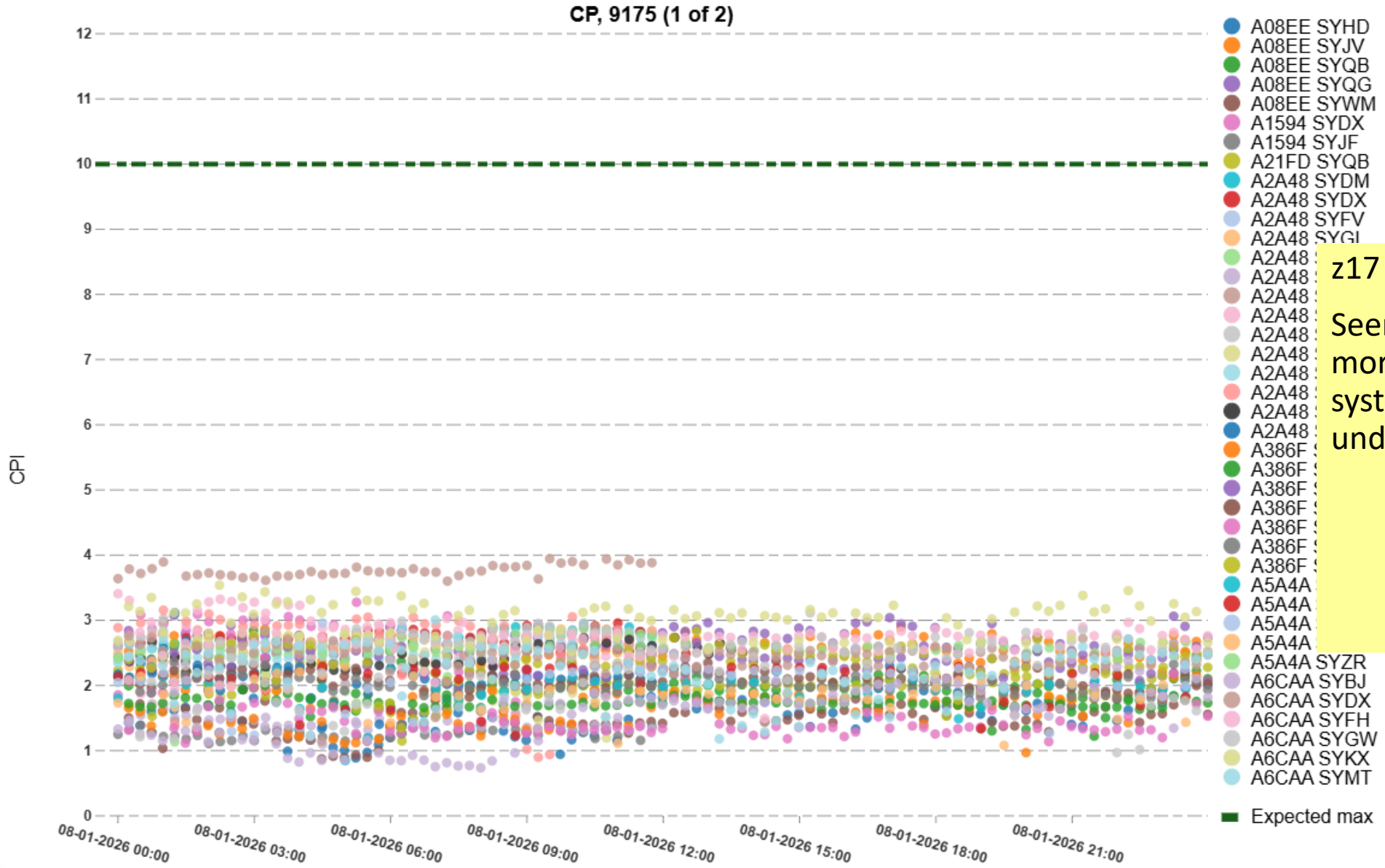


z16 systems

Looks like maybe a minor downwards shift, but still mostly in the 2-3 range.

Cycles Per Instruction

By z/OS Hardware Model

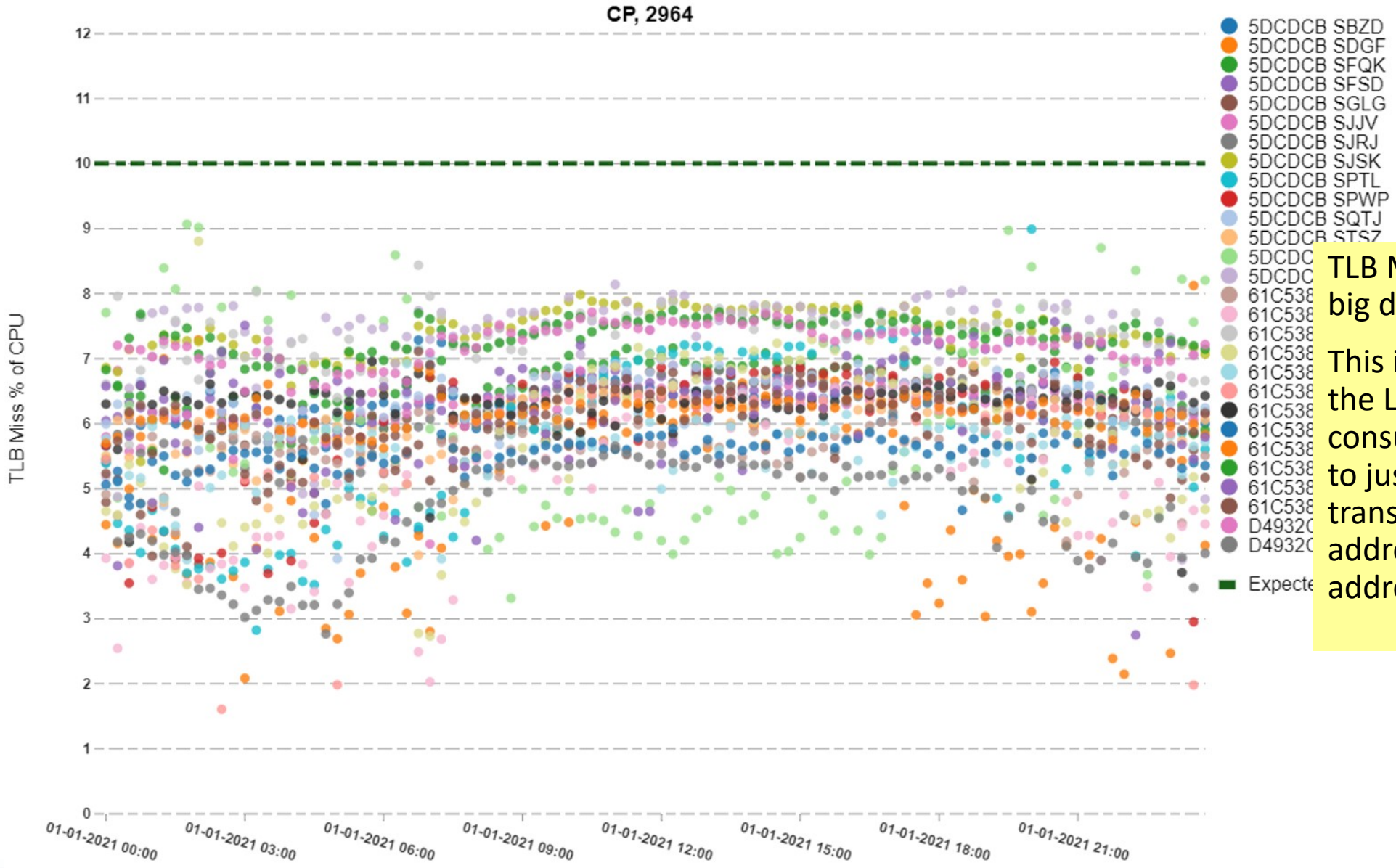


z17 systems

Seems like there are more instances of systems achieving a CPI under 2.

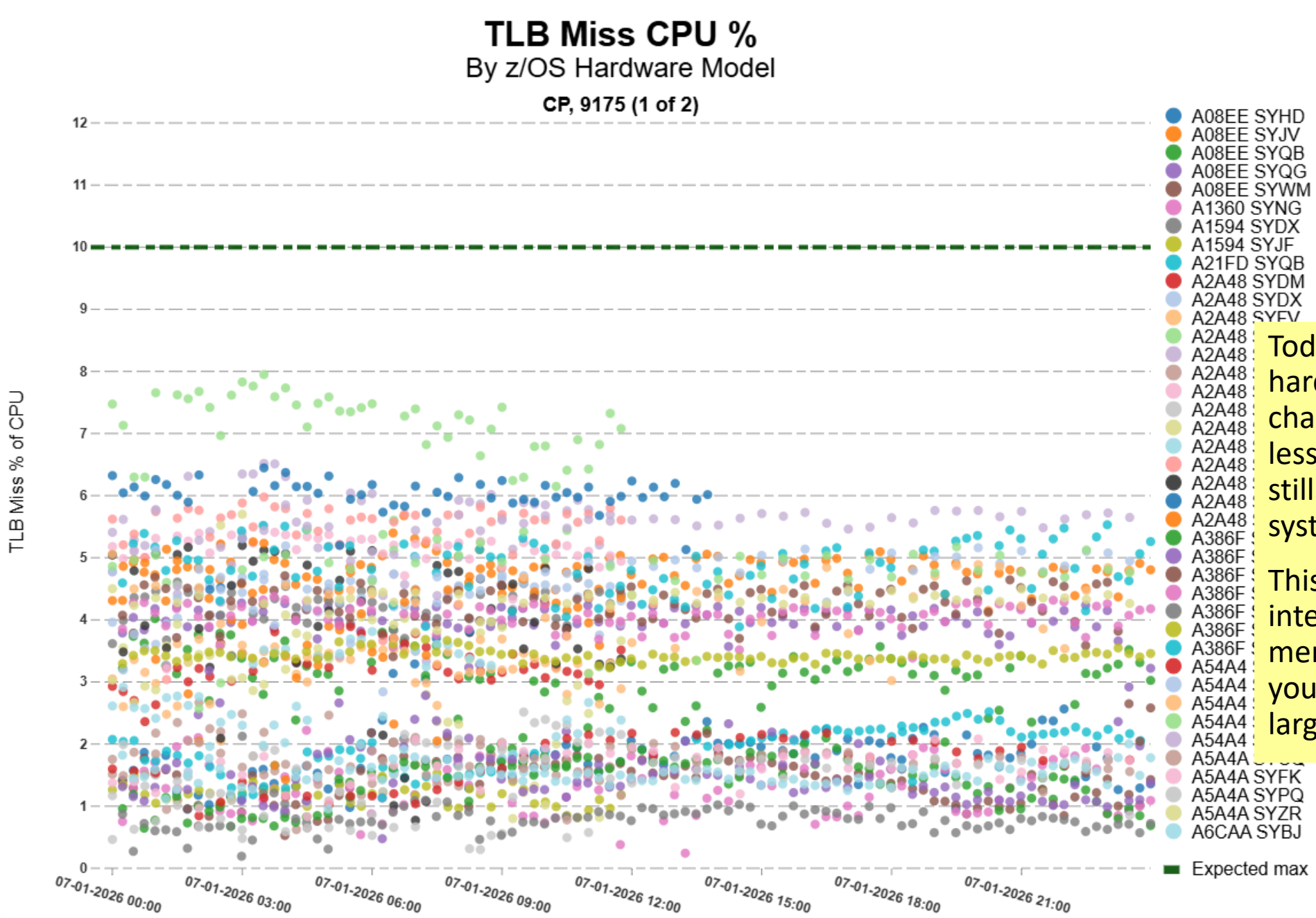
TLB Miss CPU %

By z/OS Hardware Model



TLB Miss % used to be a big deal before the z14.

This is the percentage of the LPAR's CPU consumption that goes to just handling the translation of virtual addresses to real addresses.



Today, after the z14 hardware DAT engine change, that's usually less of a concern, but it still exists for some systems.

This depends on how intensively you're using memory and how much you are or are not using large pages.

HIS: What you should do

- Note that a change from e.g. 2.5 to 2 may seem small, but percentage-wise that's significant
- If your numbers seem high, that may be the nature of your workload, but there are a few configuration choices that do affect these numbers
- Consider more/slower vs. fewer/faster CPs
  - More CPs = more L1/L2 cache = lower L1MP = lower CPI
- Use large pages where you can
  - Can reduce the TLB Miss % CPU
  - z14 greatly reduces this TLB Miss impact, but above ~3% still worth looking at
  - Also, 1MB limit now just sets the HWM limit and doesn't reserve frames
    - 2 GB frames are still managed separately

Do you really want to jump into that stream?

Store Into Instruction Stream

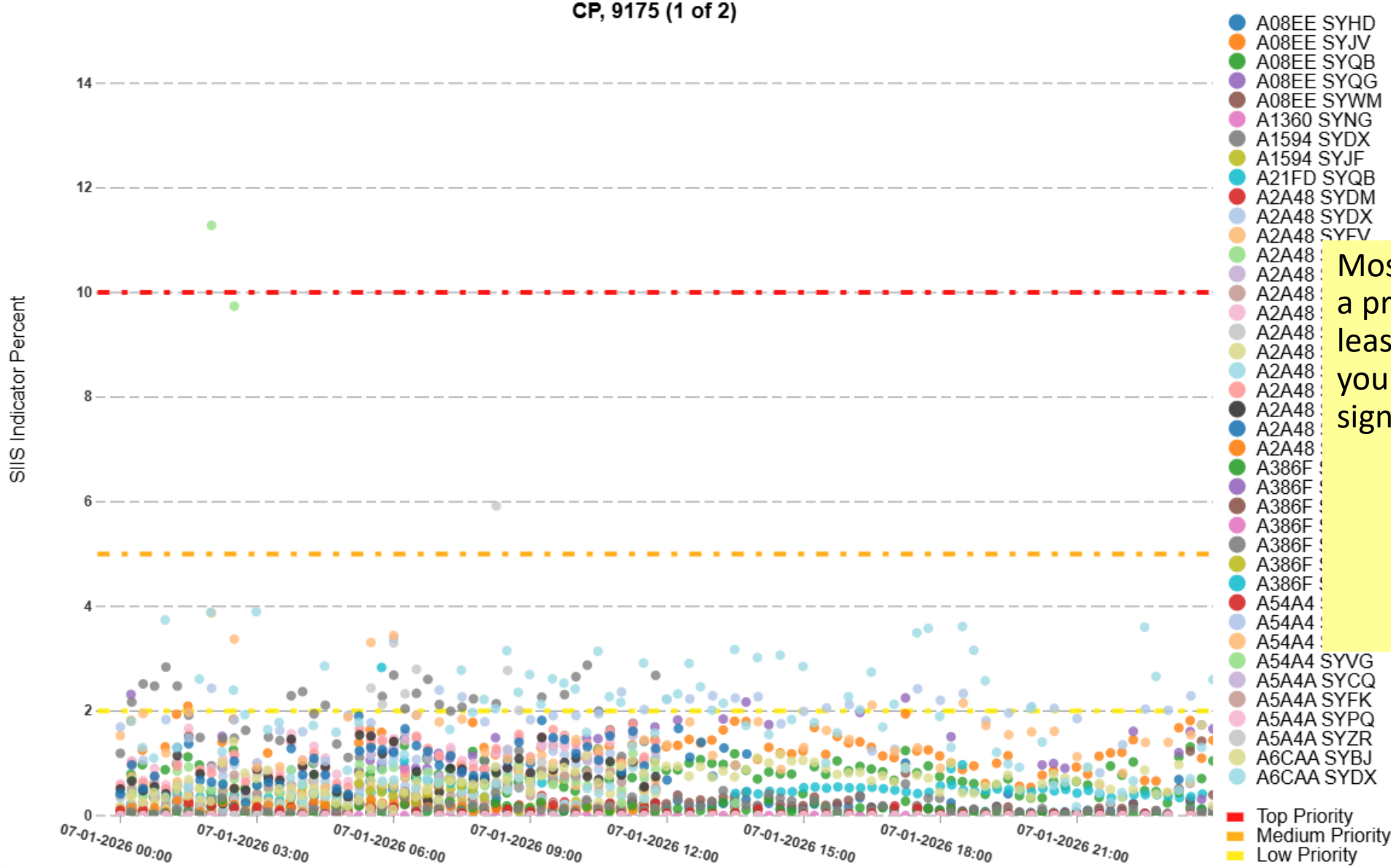
SIIS: Why you care

- Store Into Instruction Stream (SIIS) describes a situation where code writes to memory that is within 1 cache line (256 bytes) of the executing instructions
- That update will trigger a flush of that cache line from the L1 Instruction cache
- This can be a significant performance hit!
  - Pipeline has to be flushed
  - Obviously doing this once is not a noticeable problem, but doing it repeatedly can be
- This is not a new problem, but relatively recently IBM came up with a formula to give an indication of the relative impact of SIIS
  - Threshold for action at 5% with 10% said to indicate “likely significant impact”

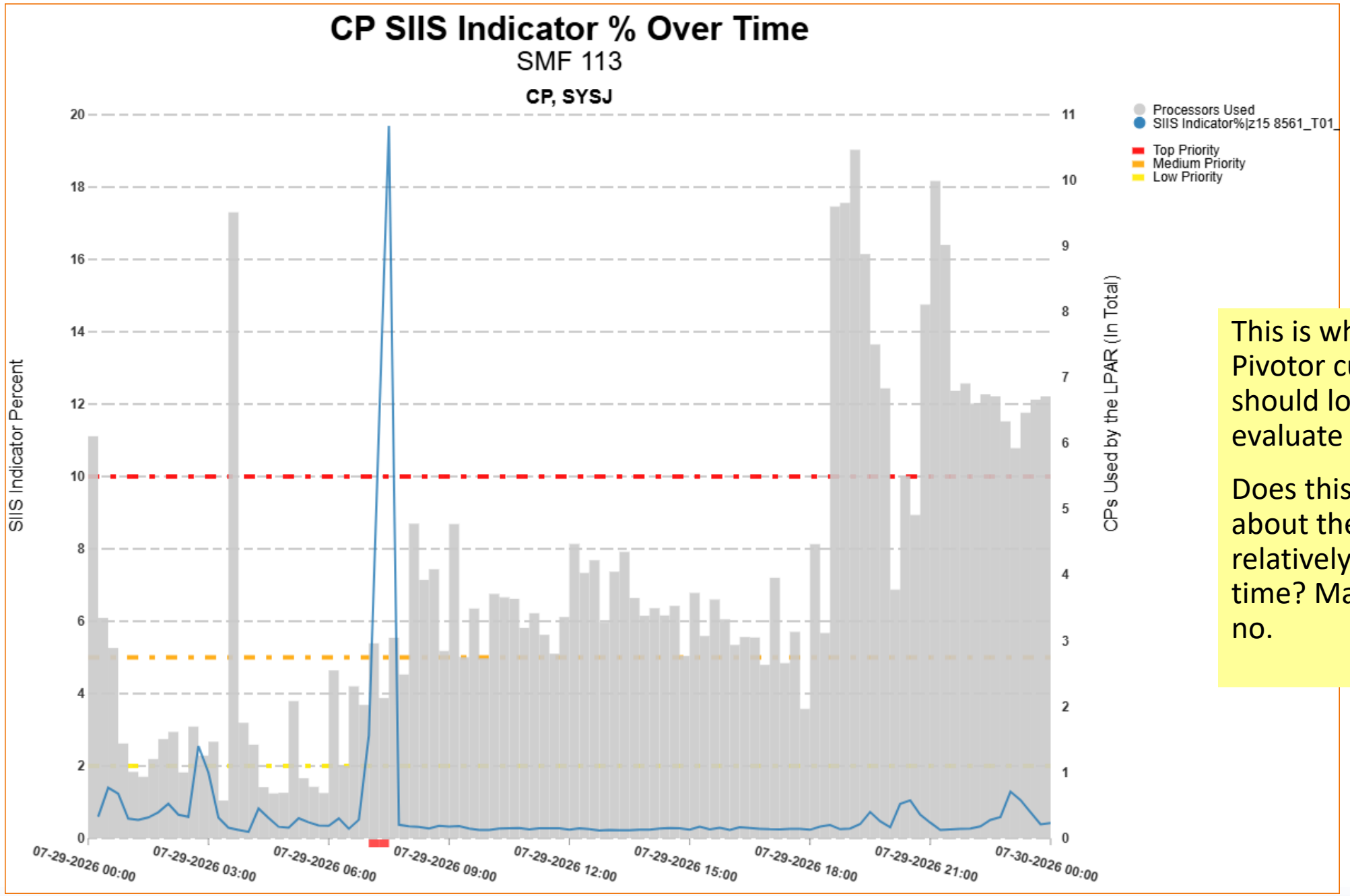
Store Into Instruction Stream Indicator

Active LPARs (>50 SMF 113 MIPS)

CP, 9175 (1 of 2)



Most systems don't have a problem with this, at least consistently. But if you do, it can be significant.



This is what reports Pivotor customers should look at to evaluate the SIIS impact.

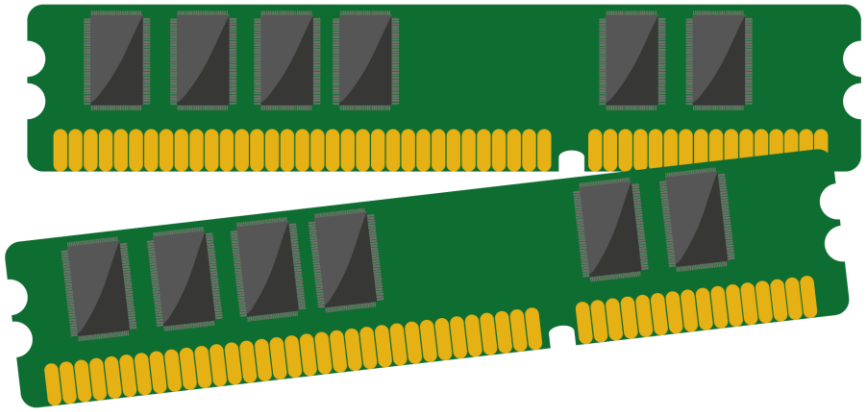
Does this customer care about the one spike at a relatively low utilization time? Maybe so, maybe no.

SIIS: What you should do

- If you have systems that:
  - Are regularly showing SIIS above 10% (or maybe 5%)
  - And are not just mostly idle during the high SIIS intervals
  - And utilization in those SIIS intervals are contributing to your software costs
- Then try to find the offenders
  - What's running during that timeframe?
  - Almost without fail, the offender will be code written in assembly language
    - Or high level languages compiled with a really, really old compiler
  - The SMF 30 instruction count fields *may* be of help here
    - But I'd start by looking first at commonalities between what was running in the SIIS intervals
  - See also: https://www.pivotor.com/library/content/Chapman_SIIS_SearchingForCulprits_2021_11_GSEUK.pdf

If only I could remember…

Memory

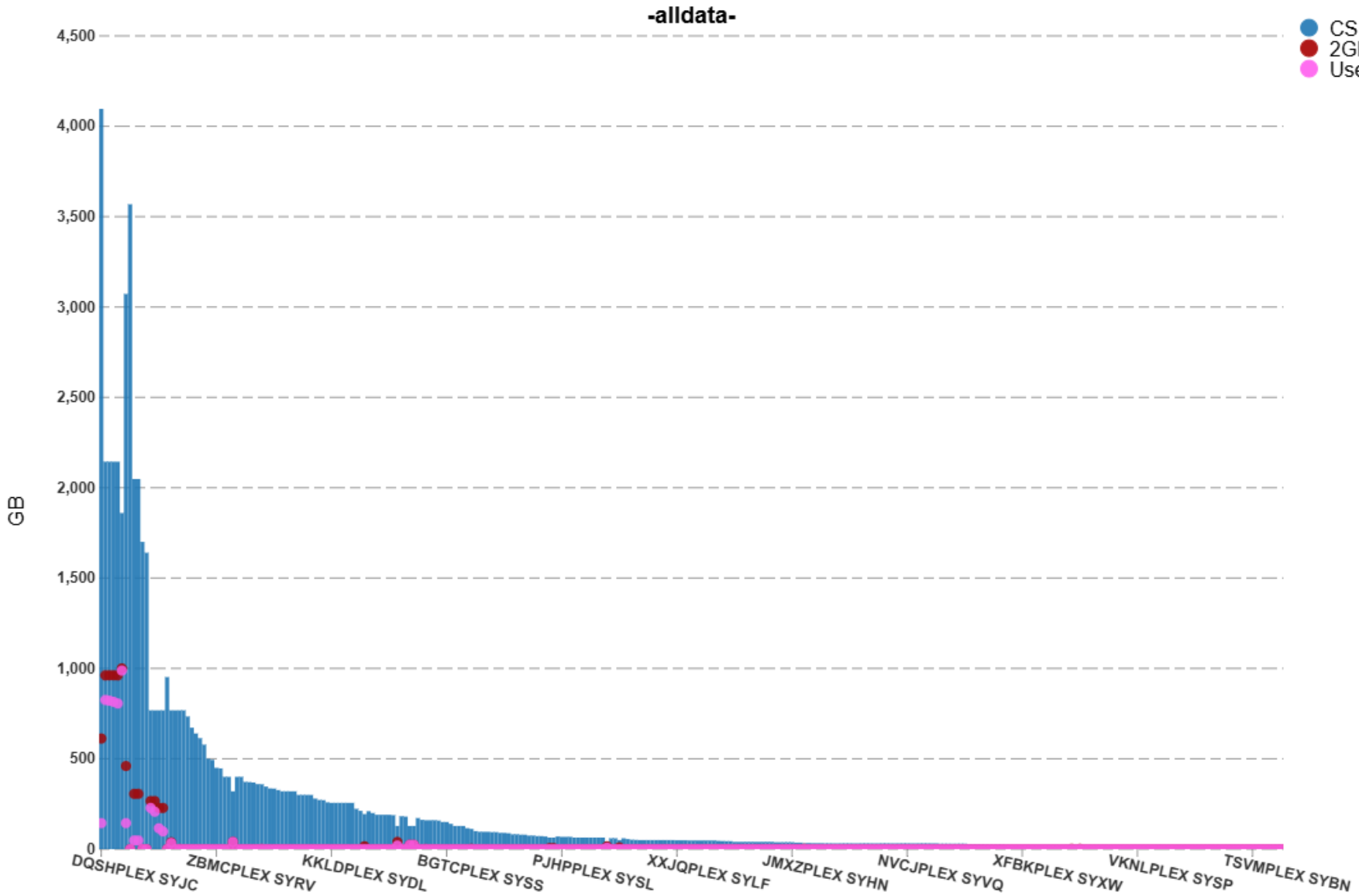


Memory: Why you care

- Keeping data as close to the processor as possible is the key to optimizing performance
  - Larger memory sizes means it's more plausible to keep more data in memory
- Use of large 1MB frames can help lower TLB Miss % CPU
- Paging has a significant performance impact
  - FlashExpress can mitigate *some* of the cost of paging
  - New Virtual Flash Memory even better: page to a RAM drive instead of a flash drive
- Large memory can offset IO and CPU
  - Use large pages though!
- You pay for memory once, you pay for software every month

Total Storage vs 2GB Frames

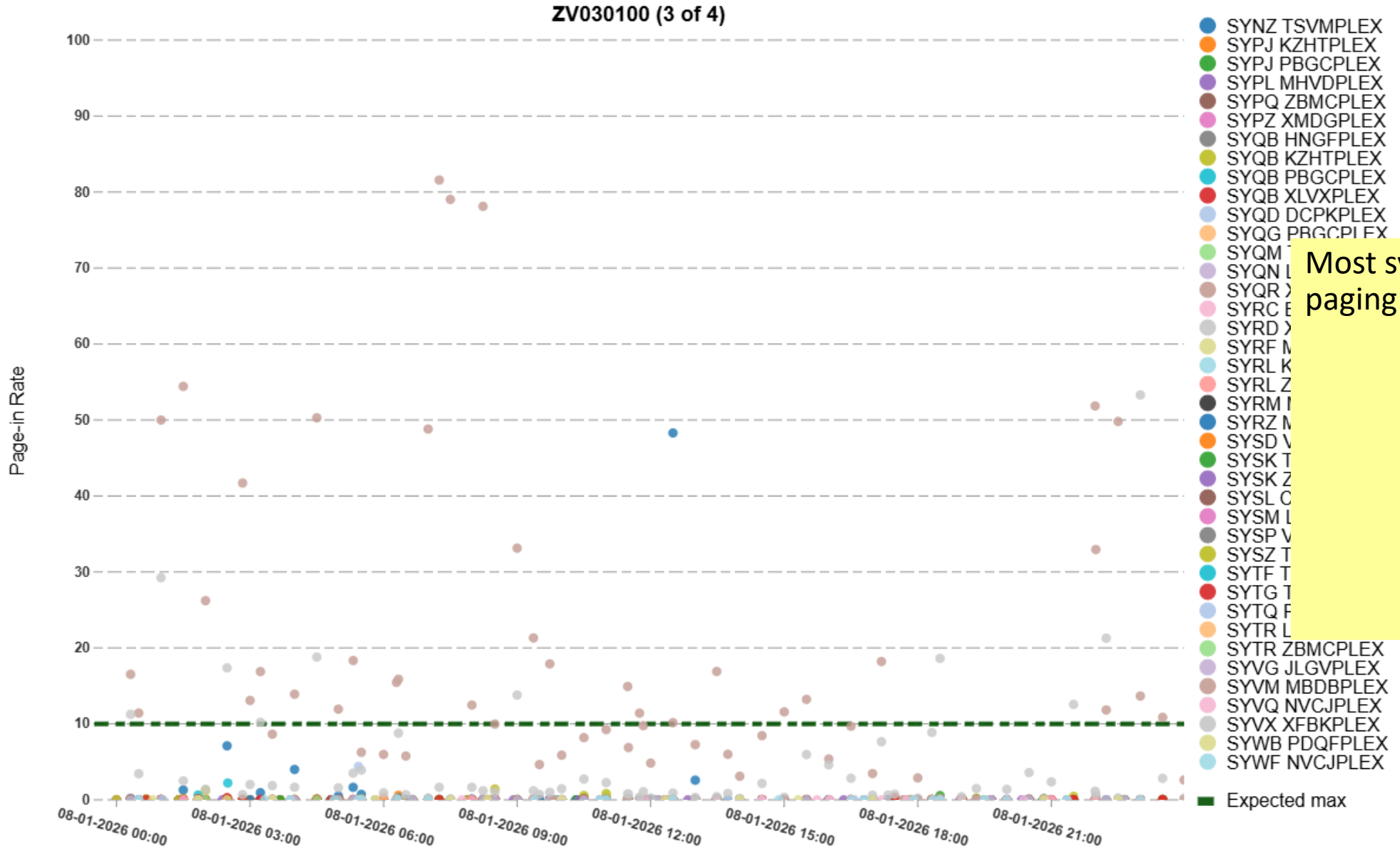
In GB



Yup, there's systems out there with multiple TBs of memory. Somewhat surprising to me that so few of the pages on those large systems are 2GB pages.

But over half of the systems have 64GB or less.

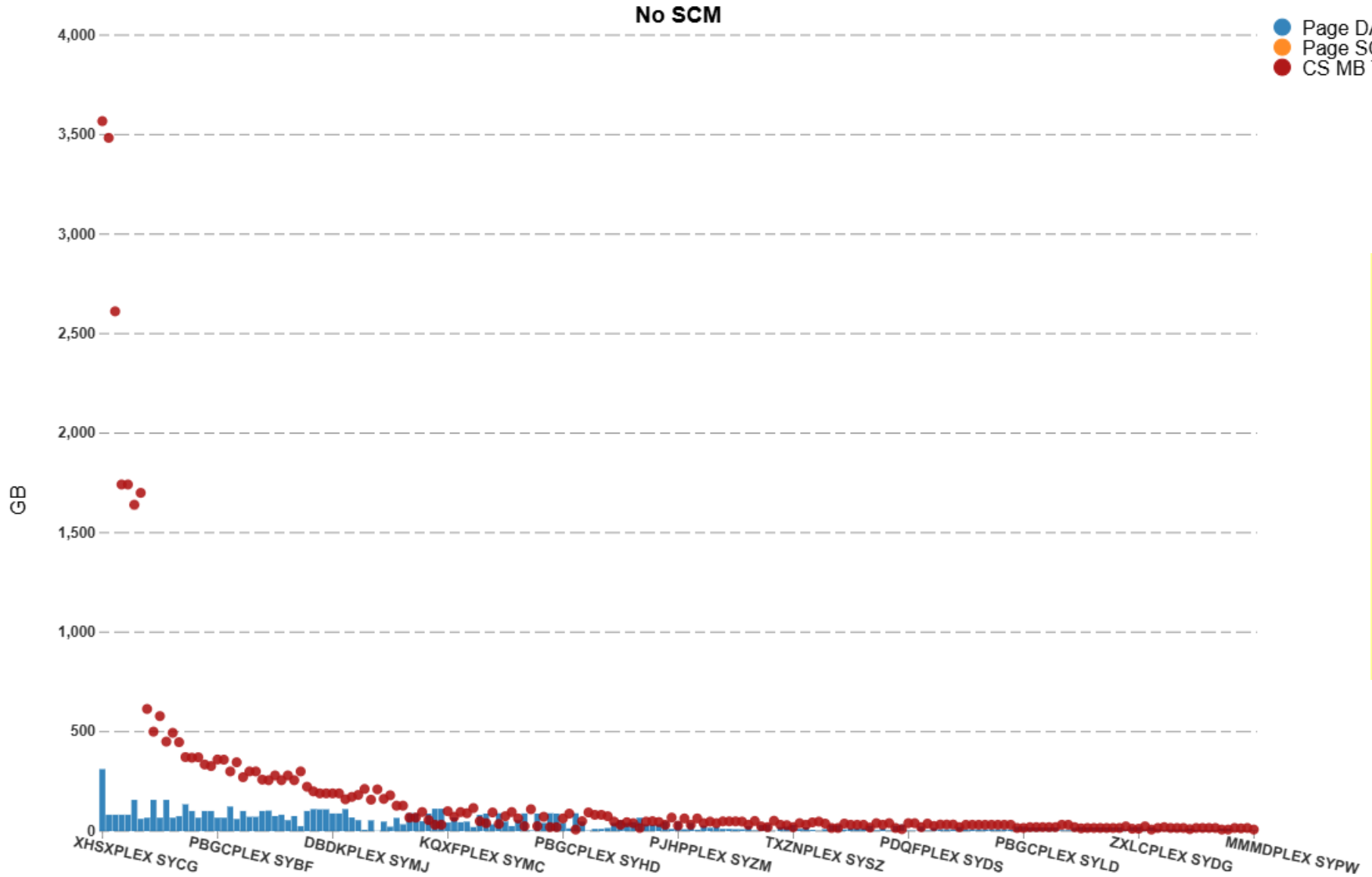
Page-In Rate



Most systems aren't paging.

Total Page Space Relative to Central Storage

In GB

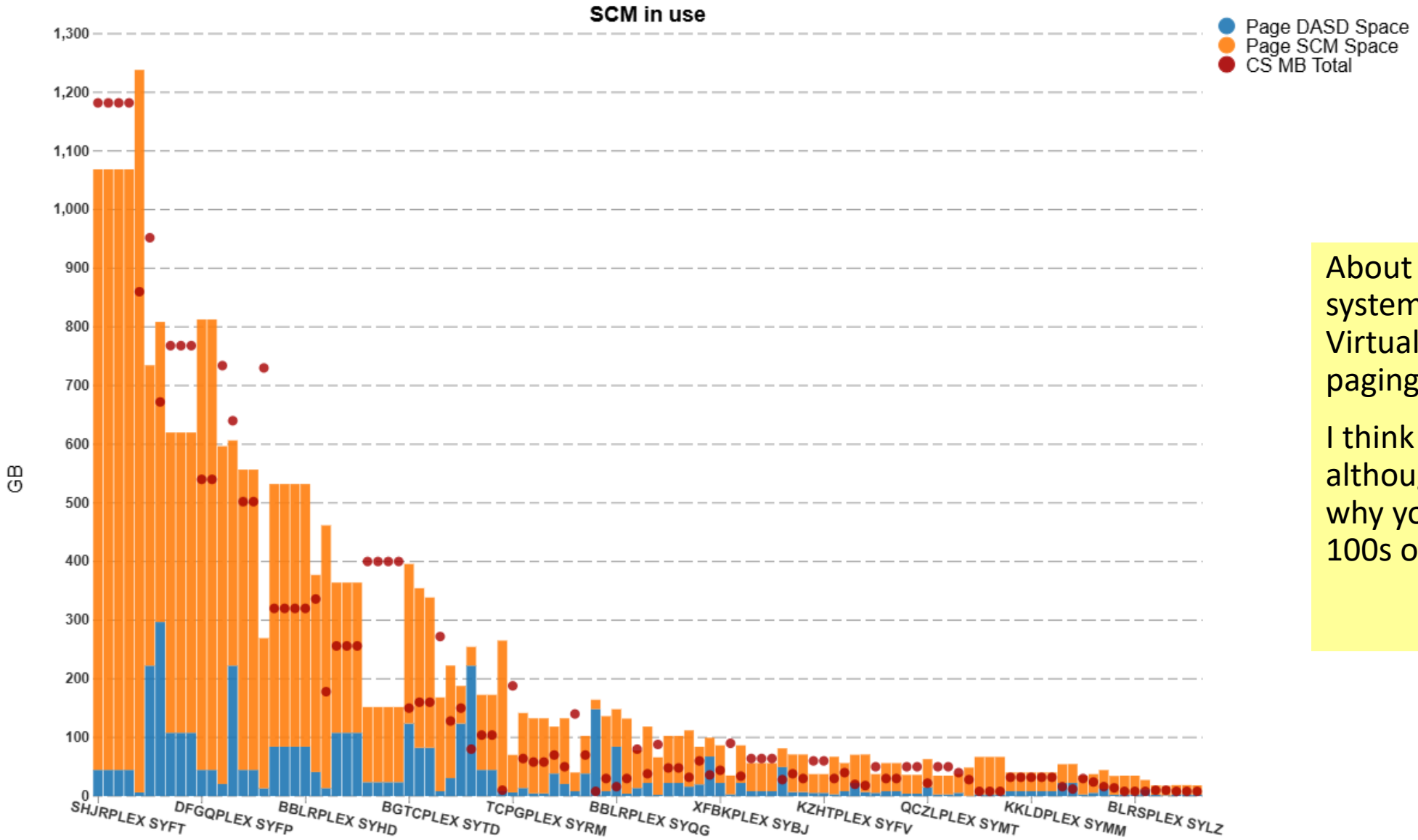


It makes sense that page space does not have to grow with the size of memory today.

It's not like the old days when we were assigning memory to LPARs in MBs.

Total Page Space Relative to Central Storage

In GB



About a third of the systems we see have Virtual Flash Memory for paging.

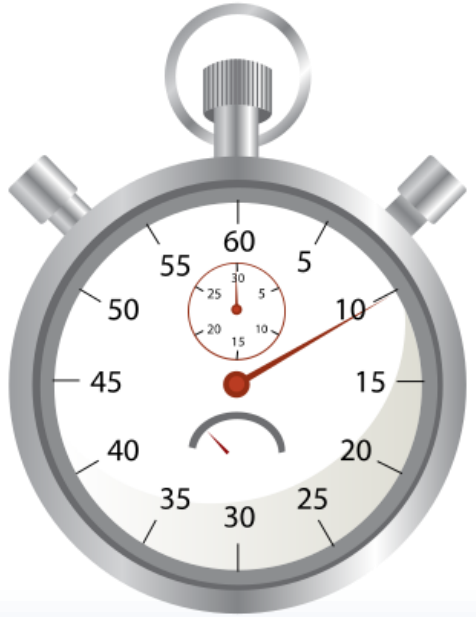
I think VFE/SCM is great, although I do wonder why you might need 100s of GBs per LPAR.

Memory: What you should do

- 1MB Pages are commonly in use and are easier to configure these days
  - 2GB pages still fairly uncommon, but I think more should be using them, especially once you start having Db2 BPs that are several 10s of GBs.
- Avoid paging: if you're regularly paging in more than a couple of pages a second, you're certainly in the small minority
  - Dumps may be causing some of that paging
  - Even with VFM, a system low on real storage may run differently (read: probably worse)
- Especially for large memory configurations, you probably don't need page datasets equal to the size of real memory
  - OTOH, FlashExpress or Virtual Flash Memory can give you very large paging spaces
    - Which might be very good if you are in fact paging or if dumps are being disruptive
  - If you have SCM/VFM in use for paging, you can probably shrink your DASD paging space
    - Don't forget about DR

Microseconds matter

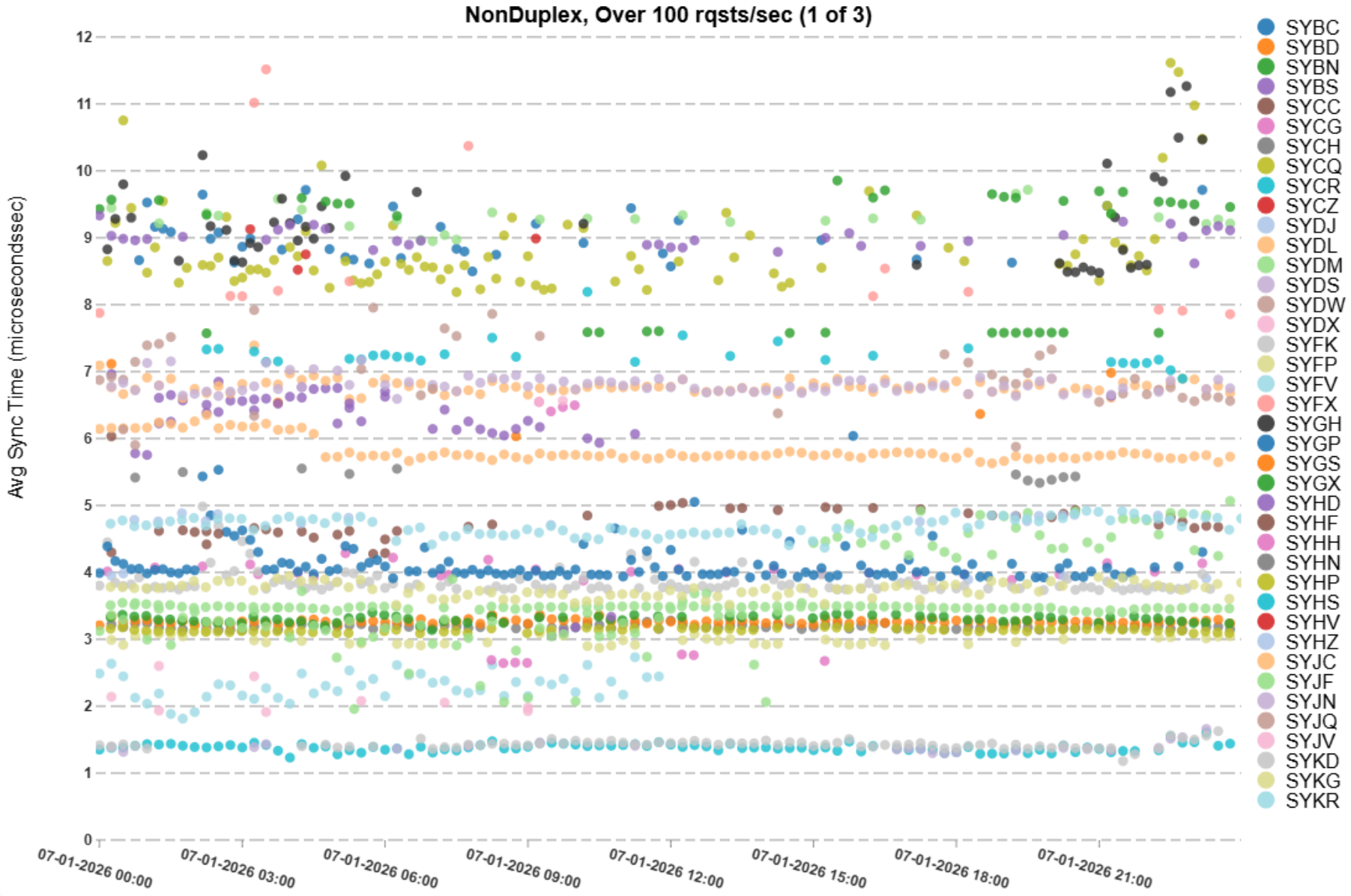
CF Sync Response Time



CF Sync Request Response Time: Why you care

- The processor spins while executing a synchronous request the CF
  - I.E. the CPU just loops waiting for the request to come back
  - Because sync requests should be very quick, this is generally the more efficient than an async request where the CPU is released to go do other work then has to be redispatched later
  - But this can also be looked at as overhead
  - Quicker CF requests = less overhead
- CF response time depends on multiple things, including:
  - Type of request (lock requests generally the shortest sync times we see)
  - Structure duplexing (generally don't duplex lock structures)
  - CF hardware generation
  - CF Busy
- Many customers have ISGLOCK structures, so that's a nice one to compare as a "best possible case"

ISGLOCK Avg Sync RT



Generally today I'm hoping to see under 5 microseconds for sync lock requests, and best in class can approach 1, but 3-4 is more common for a "good" response time.

CF Sync Request Response Time: What you should do

- Keep your CF hardware at the same level as your general purpose processors
  - Less raw performance delta between generations than in the past, but latest generation gives you access to latest CFCC code and features, which may be more important
  - Vast majority of sites are using internal CFs today, so this is mostly a moot point
- If your lock sync times are over 10 microseconds investigate why
  - High CF busy?
  - Long distance?
  - Inefficient sharing of CF engines?
  - Duplexing?
- Don’t buy long-reach cards if you don’t actually need to go long distances!

Forget spinning

DASD Response Time

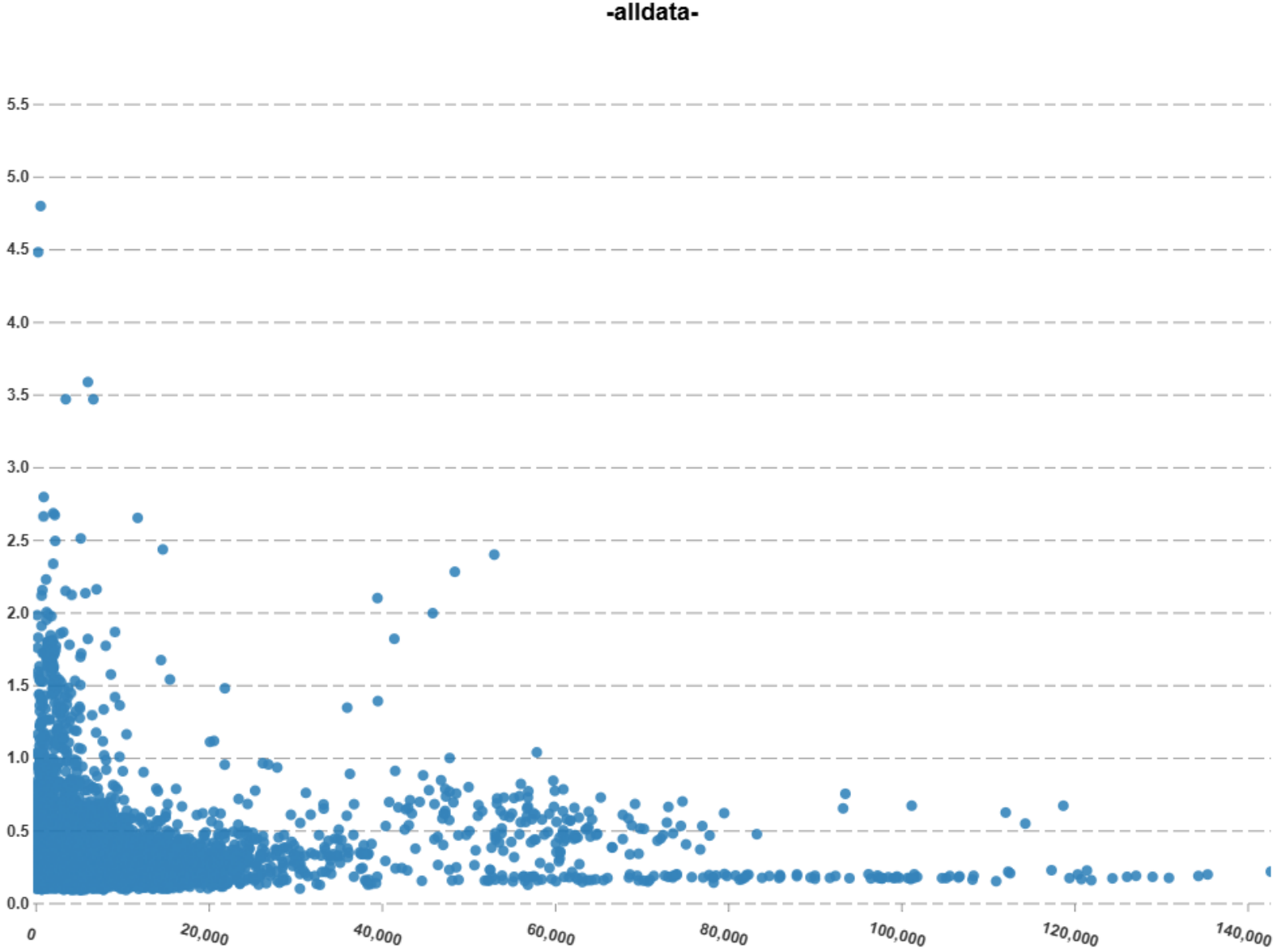
DASD Response time components: Why you care

- Because we see so much SSD storage, I/O response time is somewhat less of a concern than in ages gone by
  - Larger central storage sizes can also be leveraged to reduce I/Os
- But even in the age of sub-millisecond I/Os, the only good I/O is no I/O
  - A 1 millisecond response time is a whole lot of CPU cycles!
- Some components of I/O response time might be able to be addressed
  - “High” average response time may mean hot spots in the controller or a need to rebalance what’s on SSD vs HDD
  - IOSQ time is largely addressed with various flavors of PAV
  - High initial command response time can be indicative of bottlenecks in the controller
- The follow numbers are based on the weighted averages by physical control unit

DASD Avg RT vs Activity Rate

PCU Intervals with >100 IOPS

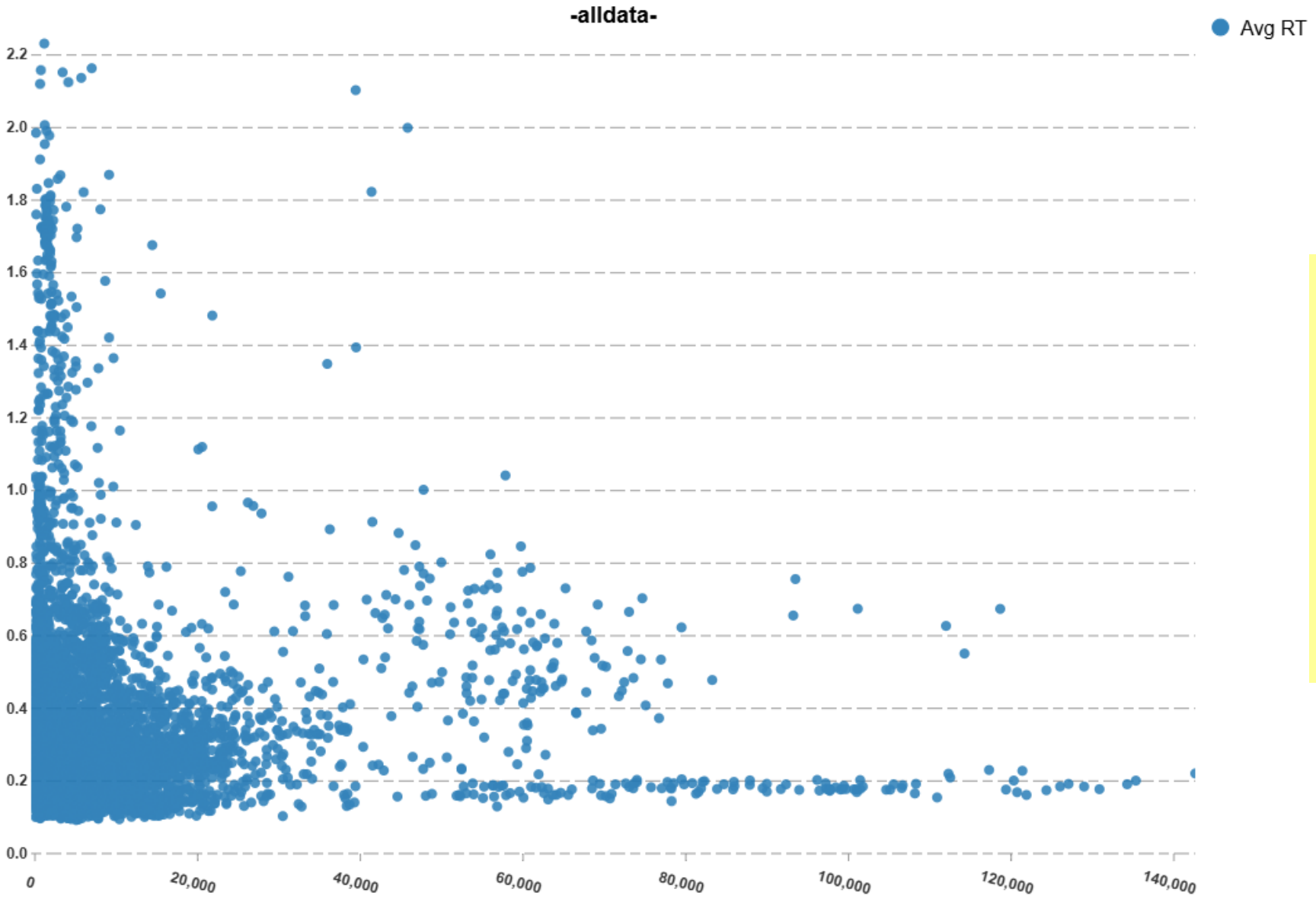
-alldata-



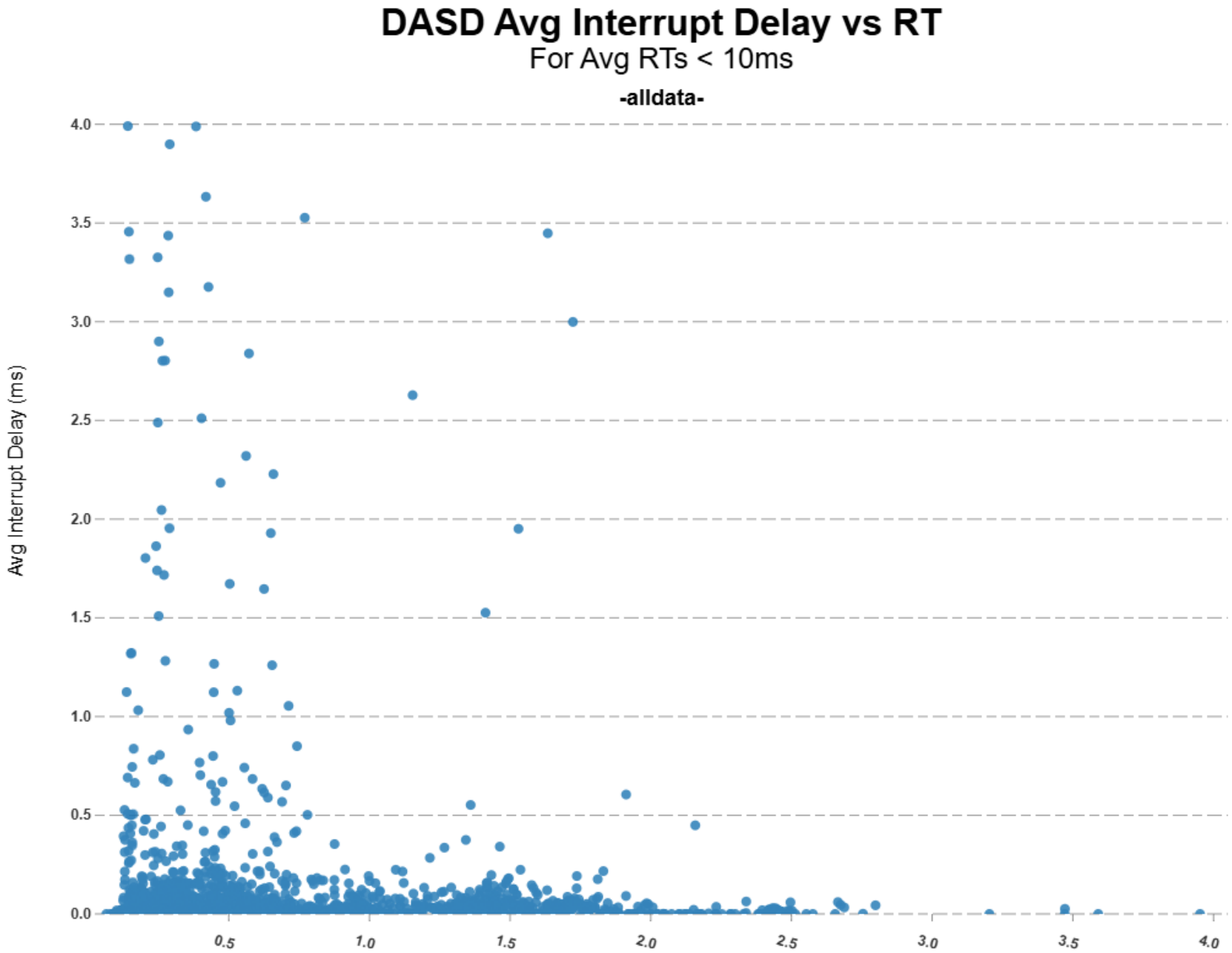
Vast majority (but not all!) of response time averages are under 0.5ms and many getting down near to the 0.125ms resolution of the RMF device RT measurement.

DASD Avg RT vs Activity Rate

PCU Intervals with >100 IOPS



Just zoomed in even further.

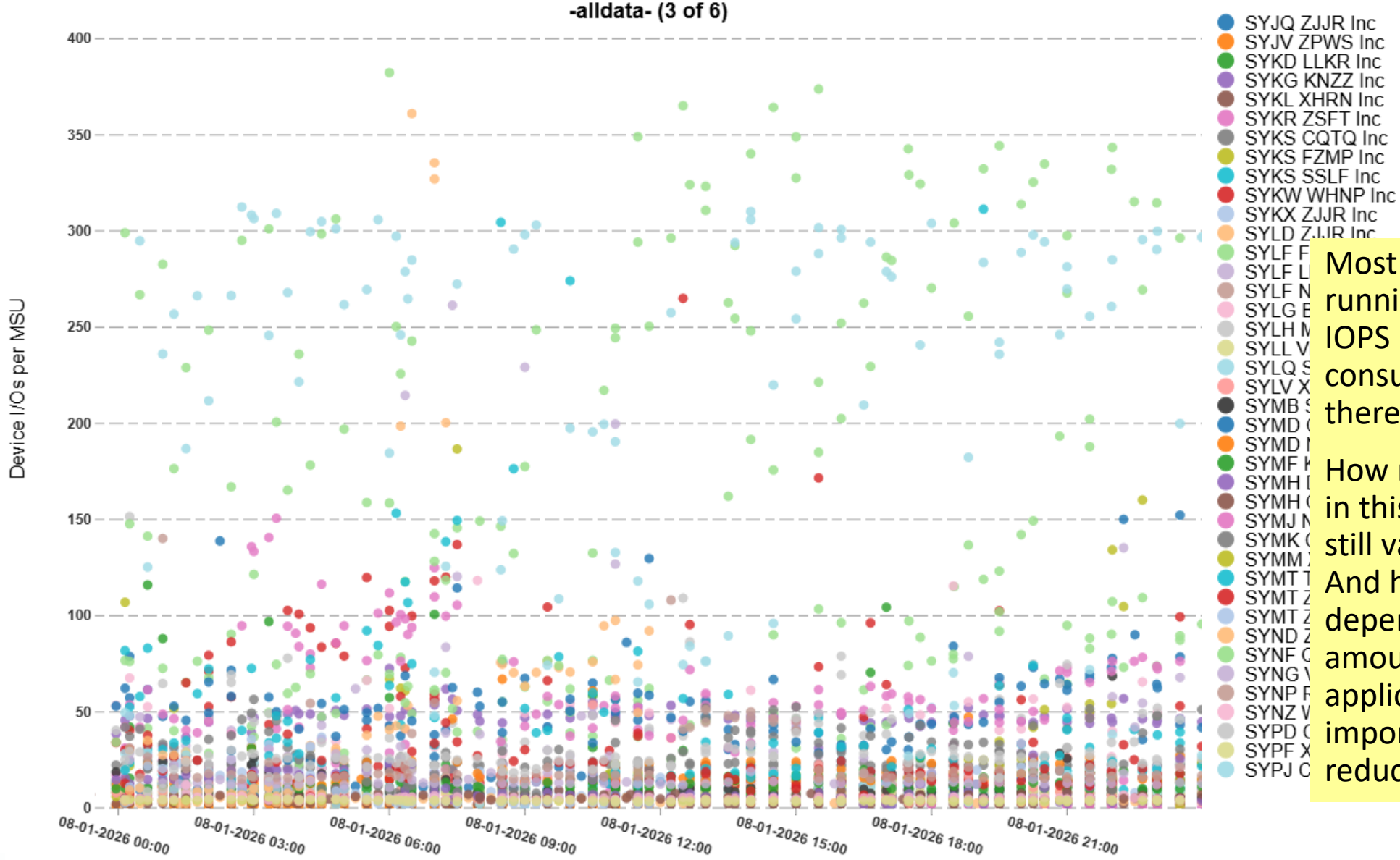


Interrupt delay is usually most significant for lower weight LPARs, but it can be a significant "hidden" extra delay to I/O response time.

Device I/Os per MSU Consumed

By System Over Time

-alldata- (3 of 6)



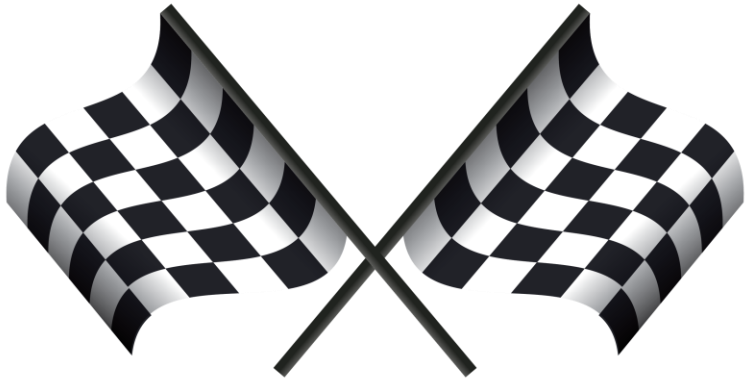
Most systems are running less than 50 IOPS per MSU consumed, although there is a wide range.

How much value is there in this? IDK… but there's still value in avoiding I/O. And how much value depends on the absolute amount of I/O, application performance importance, CPU reduction goals, etc.

DASD Response time components: What you should do

- Don’t ignore I/O reporting just because you have all SSD
  - Although you probably don’t need to look at it as often as you did in the past
- If you regularly see IOSQ time you might need to investigate your PAV configuration
  - If you have a new controller, make sure you have SuperPAV and it’s enabled
    - HYPERPAV=XPAV in IECIOSxx
- zHiperLink can make I/O faster, but…
  - Even better is getting the data out of memory via better buffering, which will both be much faster that zHL and also result in much less CPU overhead
- If you’re seeing more than 50 IOPS per MSU (capacity) consumed then maybe that’s an indication you could do better buffering?

Conclusion



Summary and Future Work

- Hopefully getting a glimpse for the range of real values that other sites are seeing for some measurements has either:
  - Confirmed that your values are not out of the ordinary
  - Encouraged you to investigate to see if you can do better

- Future work for me: more questions came up than I had time to explore today
  - Will likely update the samples in the future to capture a more things
  - Will likely add reports to explore some more relationships

- Future work for you: if you have a curiosity about particular values, send me an email and maybe I'll include it in a future version of this presentation

Your feedback is important!

SHARE
EDUCATE • NETWORK • INFLUENCE

Submit a session evaluation for each session you attend:

www.share.org/evaluation

SHARE Pittsburgh
August 2026
Session Tech158