Mainframe Metric Observations 2020-2021 Data Edition

Scott Chapman

Email: Scott.Chapman@EPStrategies.com

Enterprise Performance Strategies, Inc.
3457-53rd Avenue North, #145
Bradenton, FL 34210
http://www.epstrategies.com
http://www.pivotor.com

z/OS Performance
Education, Software, and
Managed Service Providers

Creators of Pivotor®

Contact, Copyright, and Trademark Notices

Questions?

Send email to Scott at scott.chapman@EPStrategies.com, or visit our website at http://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.

Introduction

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 January 1, 2021, but data was sampled from various times around the end of 2020 into the beginning of 2021
- 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

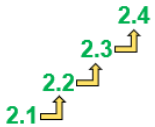
How This Presentation is Arranged

- 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

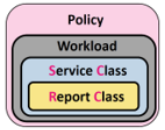


Wherein Scott preaches about SMF records

SMF & RMF Details

Worry less about your SC count and define more RDs

WLM Service and Report Classes



How busy is too busy?

CEC Busy

How many balls can you juggle?

Work Units

Do more, slower

SMT

Numbers for geeks

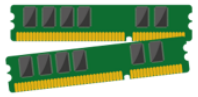
Hardware Instrumentation Services

Do you really want to jump into that stream?

Store Into Instruction Stream

If only I could remember…

Memory

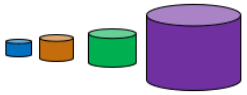


Forget spinning

DASD Response Time

More/smaller or fewer/larger?

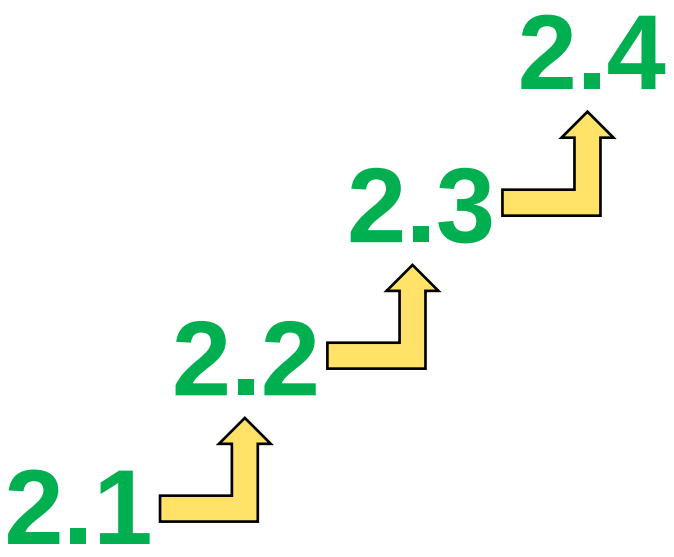
Volume Size



Conclusion

How current are you?

Machine and z/OS versions

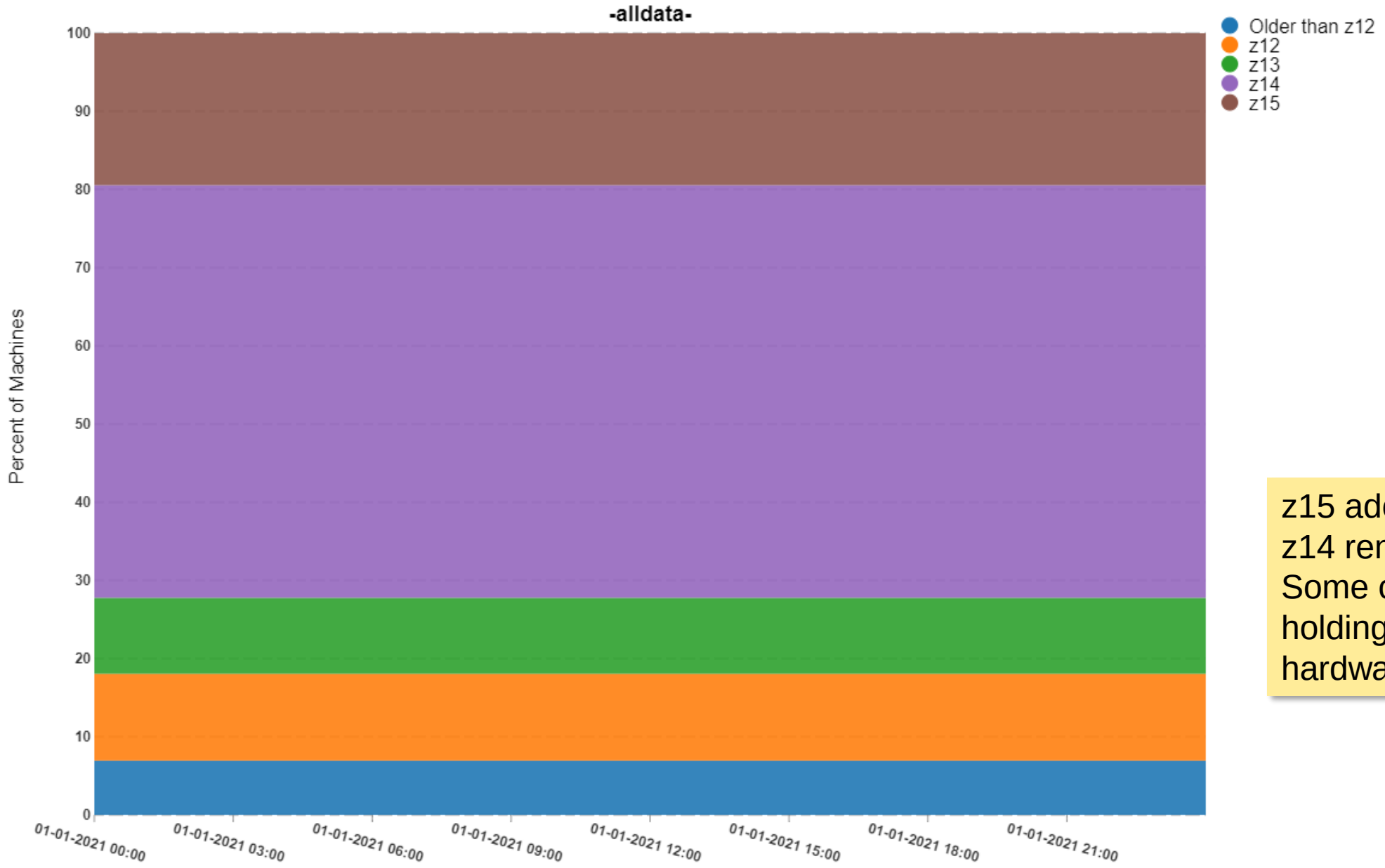


Machine Generations and z/OS versions: Why you care

- 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 2.3 requires at least z12 hardware

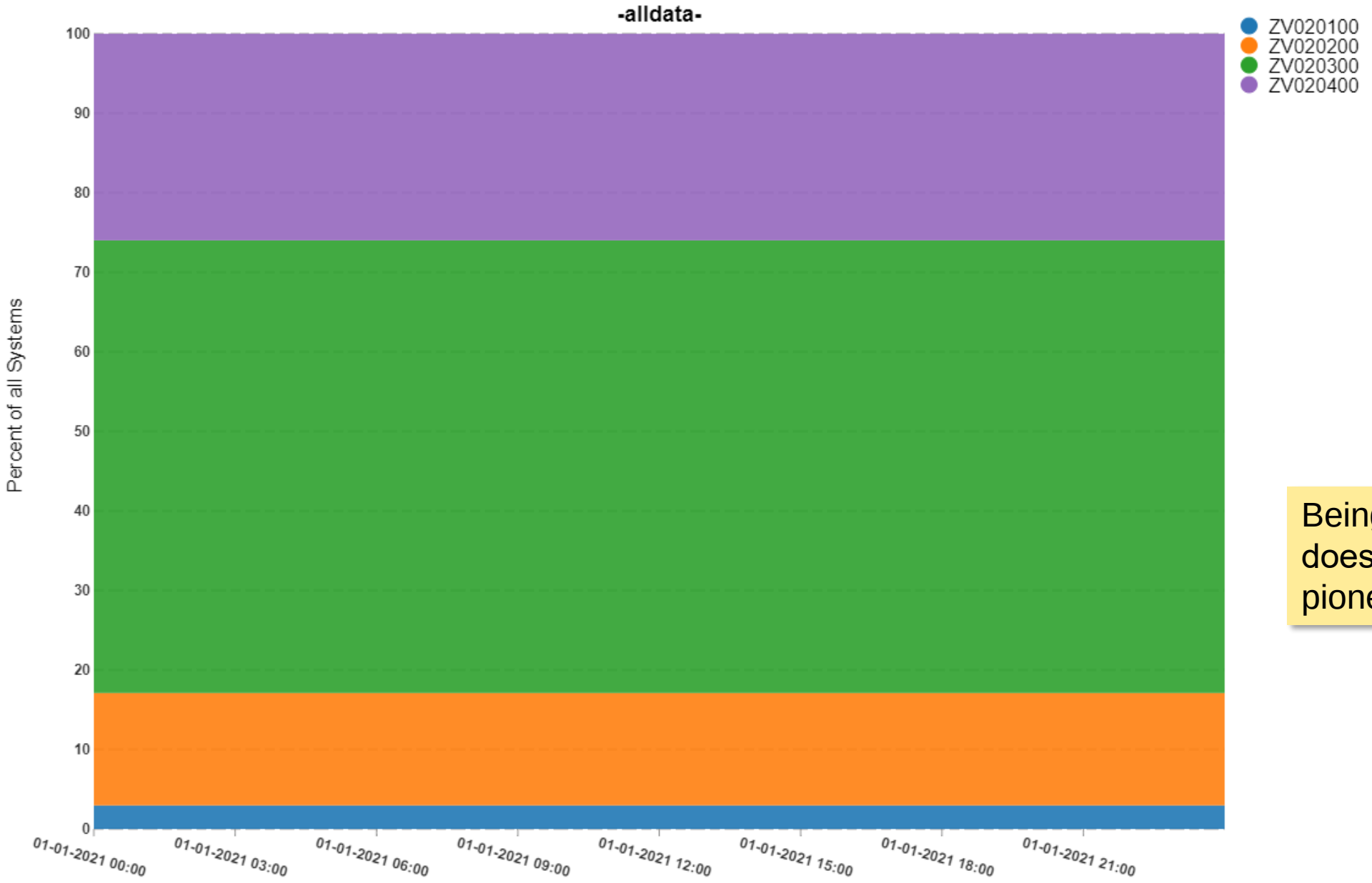
- z/OS 2.1 went out of support Sept 2018, z/OS 2.2 went out September 2020
  - Extended support may be available at an extra cost

Machine Generations



z15 adoption good, z14 remains popular. Some companies like holding onto old hardware though!

z/OS Versions



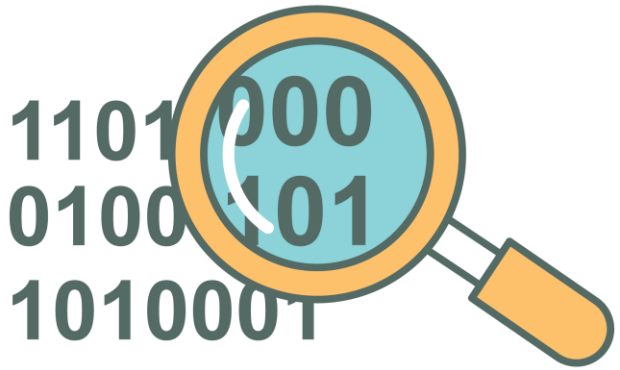
Being on 2.4 certainly doesn't make you a pioneer in 2021!

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

- If you're not on z/OS 2.3, hopefully you're transitioning there soon
- Start making plans for getting to at least z14 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

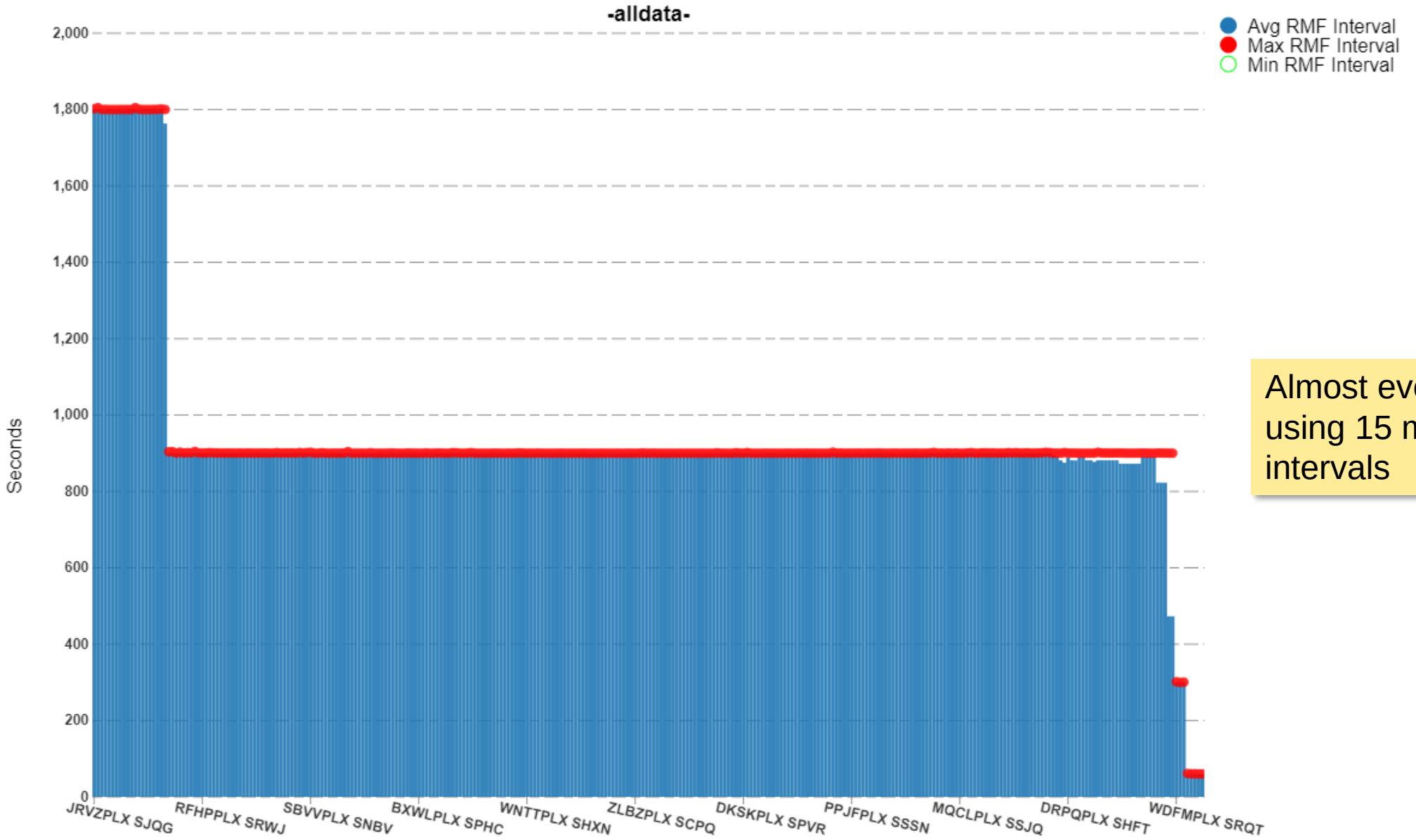


SMF Data & RMF/CMF Intervals: Why you care

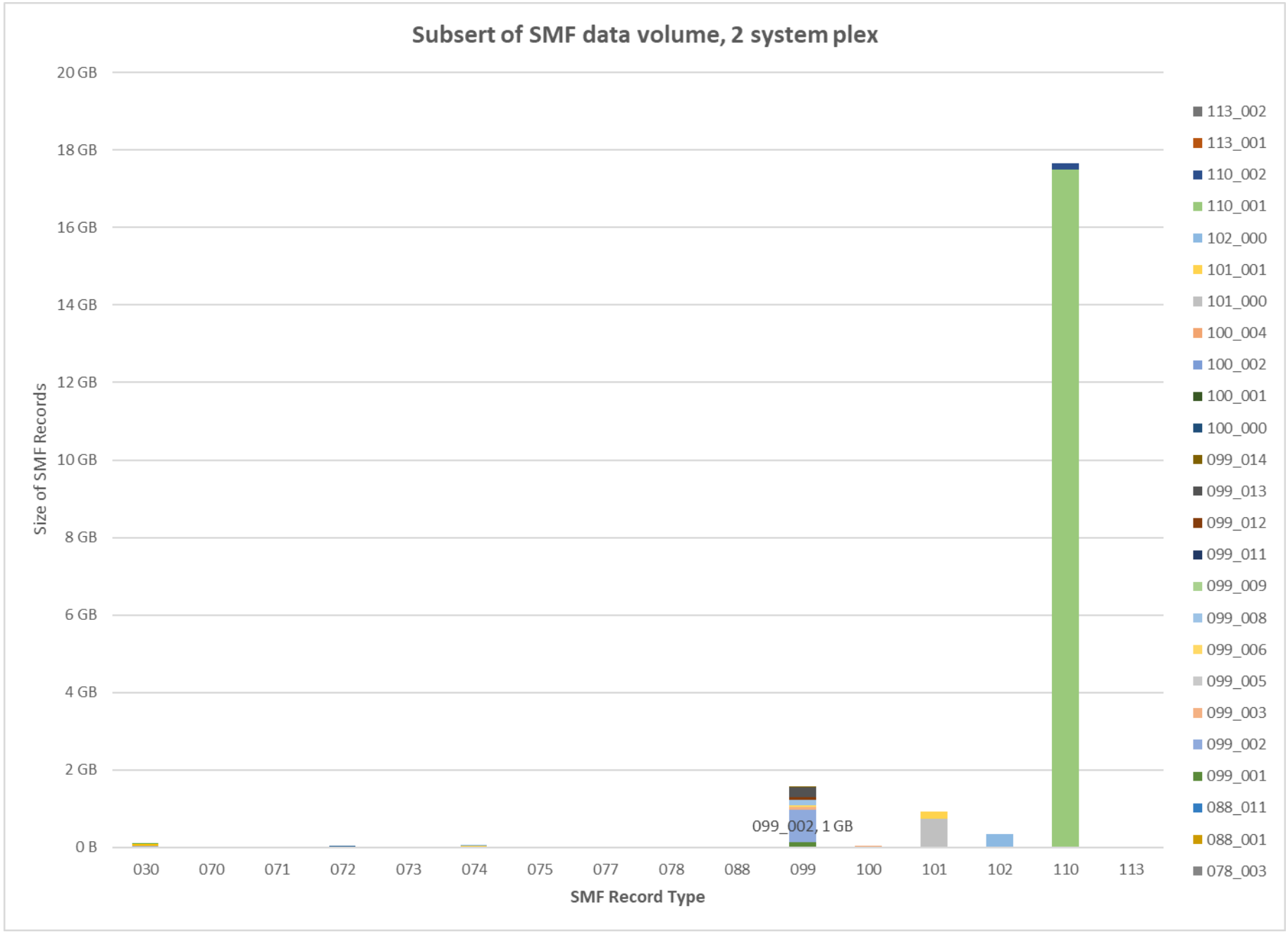
- SMF data, particularly that recorded by RMF/CMF is, key to managing your system
- Having the data readily available after an incident makes problem diagnosis easier
- But some worry about the size of the recorded data
  - Don’t just look at record counts: look at total bytes (e.g. avg record length * 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



Almost everyone is using 15 minute intervals



This just shows how CICS and DB2 records can completely dwarf the 99s.

(This from a relatively small 2-system plex with relatively little DB2 work.)

Note majority of the 99s was subtype 2.

Type	%Recs	%Bytes	Recs	Bytes
030	1.21	0.45	64,849	93,592,224
070	0.01	0.01	288	1,213,248
071	0.00	0.00	144	353,088
072	0.17	0.08	9,337	15,698,760
073	0.00	0.02	144	4,207,104
074	0.04	0.18	2,160	37,016,664
075	0.02	0.00	1,200	326,400
077	0.00	0.00	144	942,336
078	0.01	0.01	288	2,374,272
088	0.09	0.01	4,965	1,463,952
099	19.28	7.52	1,036,965	1,554,239,616
100	0.21	0.16	11,520	32,561,280
101	9.28	4.49	499,027	928,100,864
102	14.36	1.61	772,144	333,011,616
110	55.28	85.45	2,973,126	17,662,175,232
113	0.04	0.01	1,920	2,981,760

SubType	%Recs	%Bytes	Recs	Bytes
099_001	0.32	0.66	17,392	135,834,448
099_002	6.98	4.09	375,595	845,156,224
099_003	2.69	0.28	144,780	57,867,128
099_005	0.04	0.00	2,383	453,386
099_006	0.32	0.20	17,274	40,910,264
099_008	0.64	0.68	34,579	139,830,112
099_009	0.24	0.01	12,821	2,615,484
099_011	0.01	0.01	576	2,643,264
099_012	1.60	0.39	86,198	81,198,512
099_013	6.41	1.20	344,792	247,455,936
099_014	0.01	0.00	575	274,850

Details for the previous 2 systems' SMF data comparison.

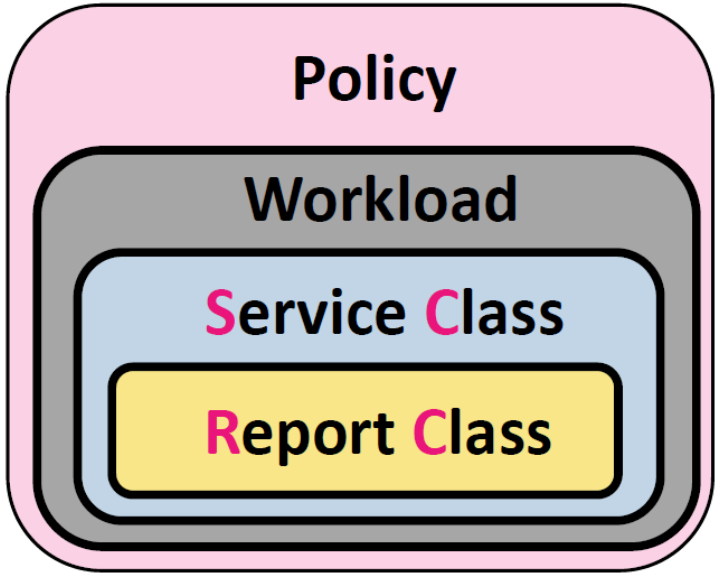
SMF Data & RMF/CMF Intervals : What you should do

- 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 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



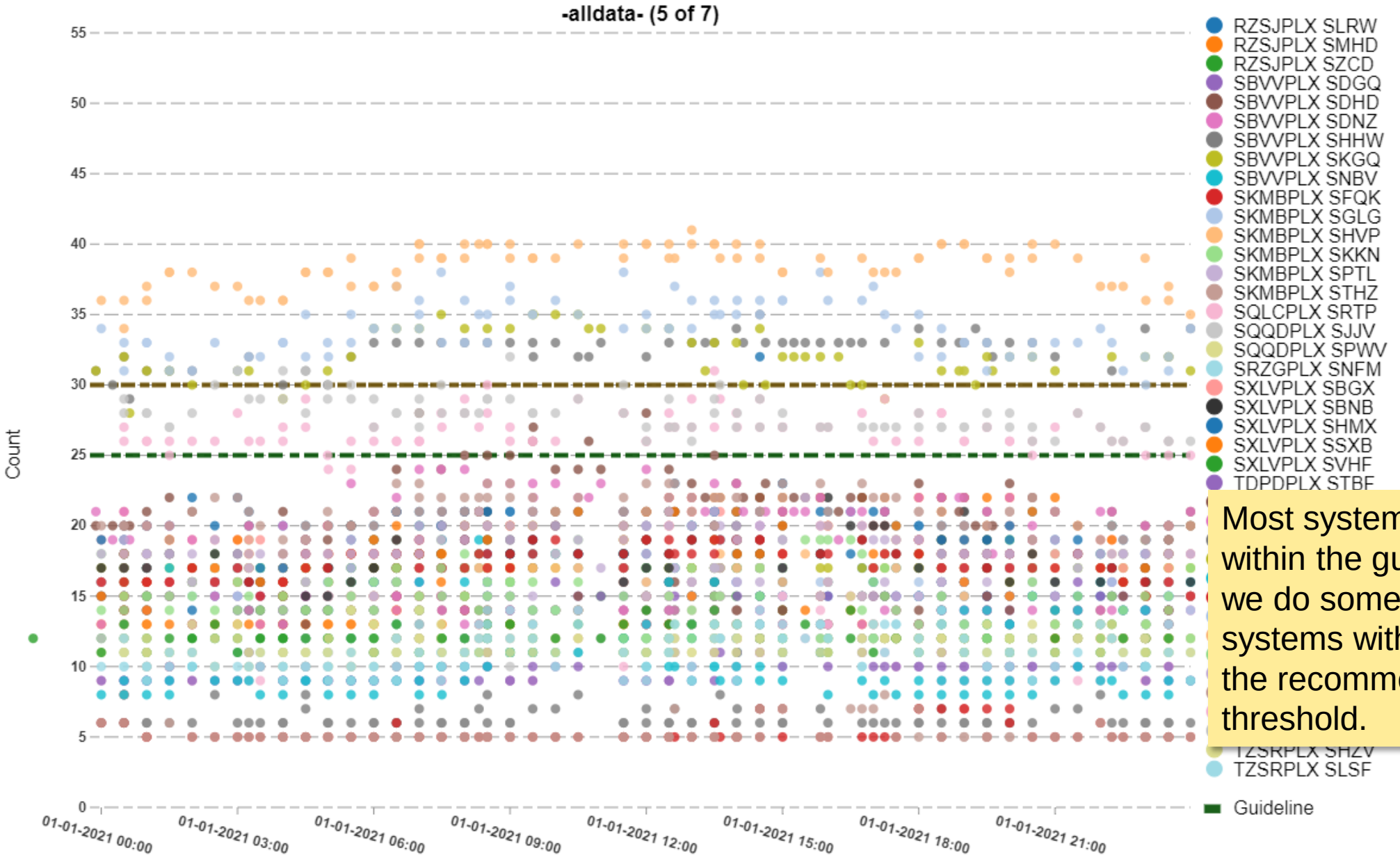
Active Service Class Periods: Why you care

- 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 work completion rates

Active Service Class Periods

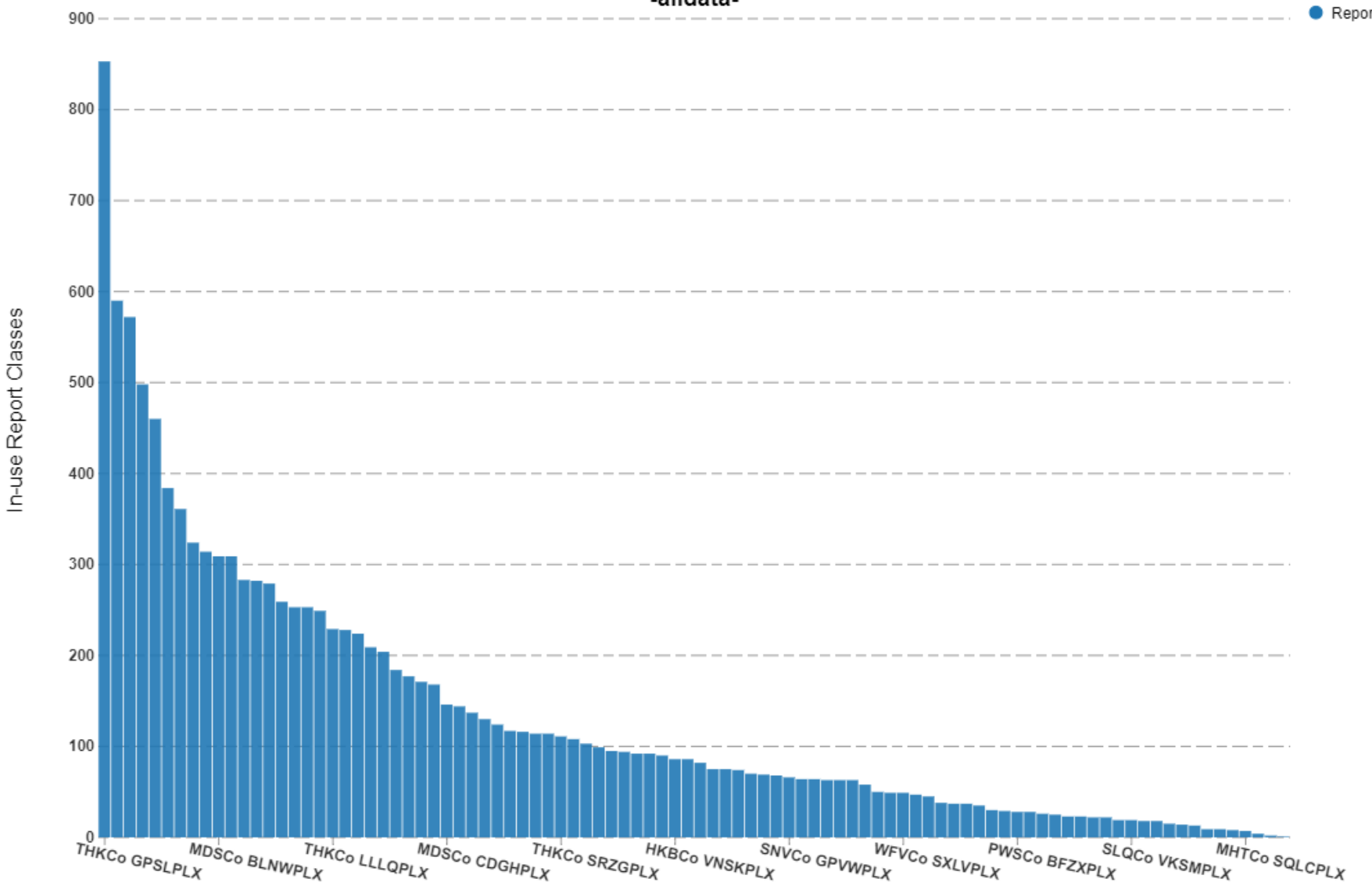


Most systems are well within the guidelines, but we do sometimes see systems with more than the recommended threshold.

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

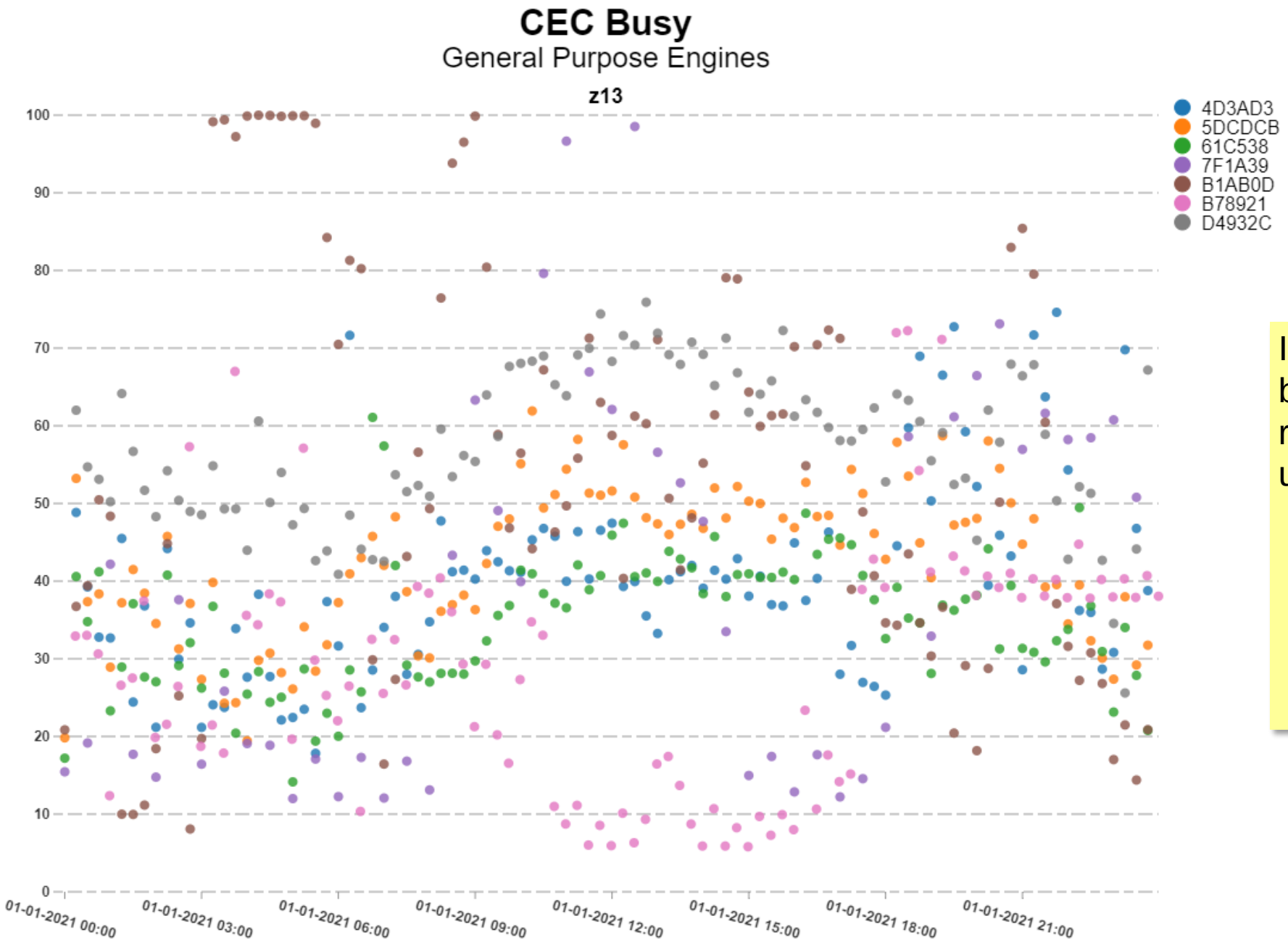
How busy is too busy?

CEC Busy

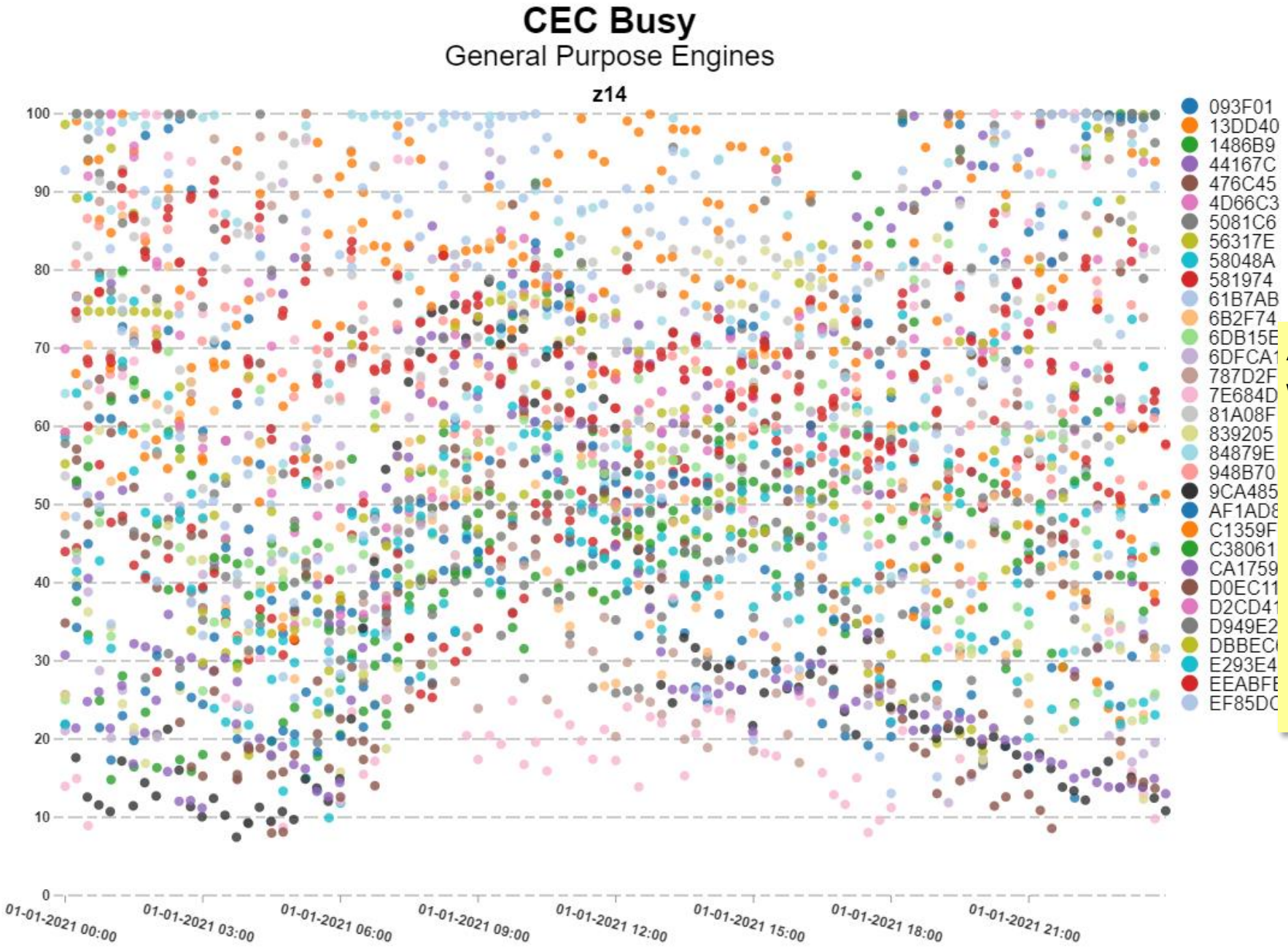
CEC Busy: Why you care

- 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?



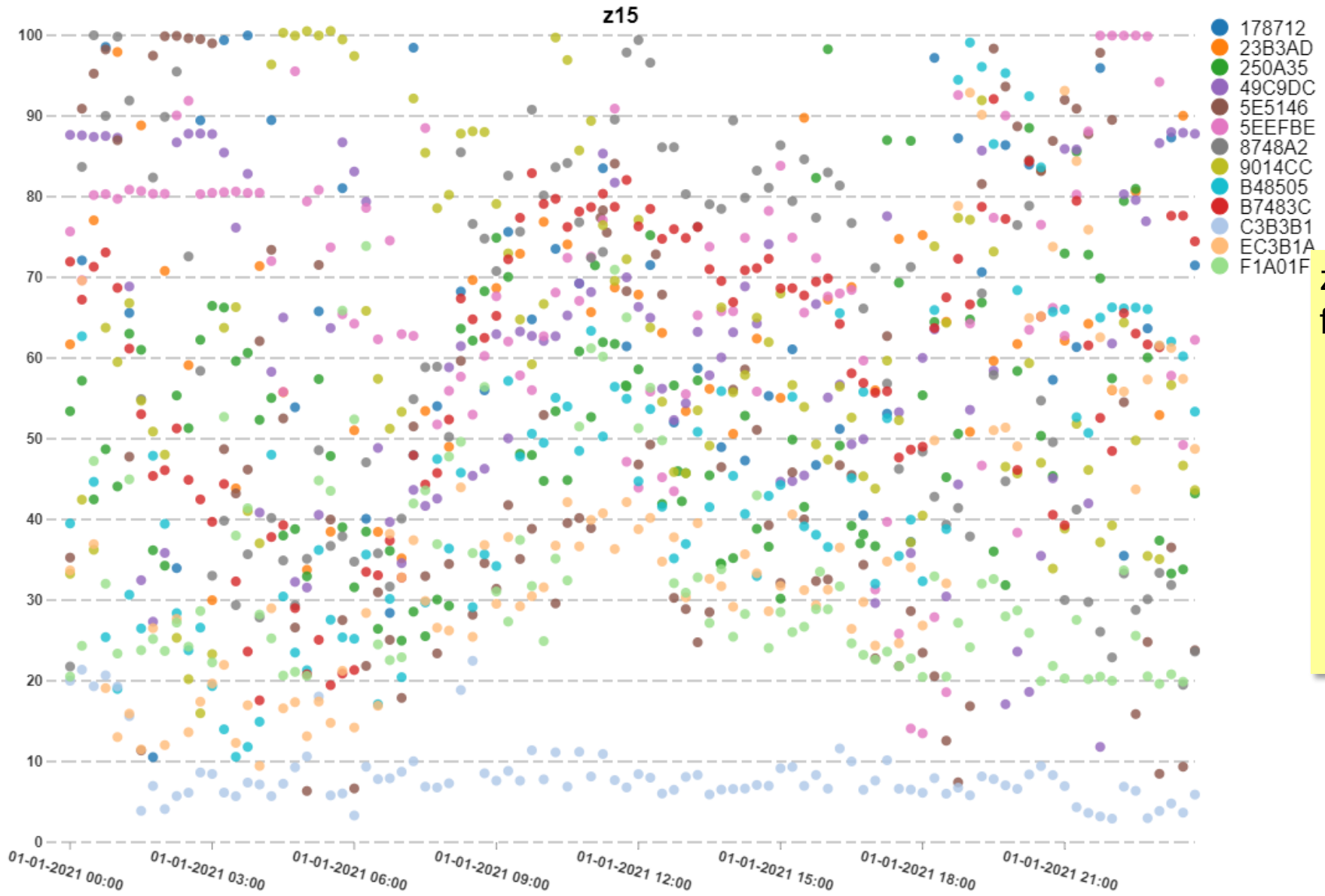
Interesting that the z13 boxes seem mostly running at comfortable utilization levels



A bunch of z14 CECs with a bunch of different utilizations, but mostly healthy during the day.

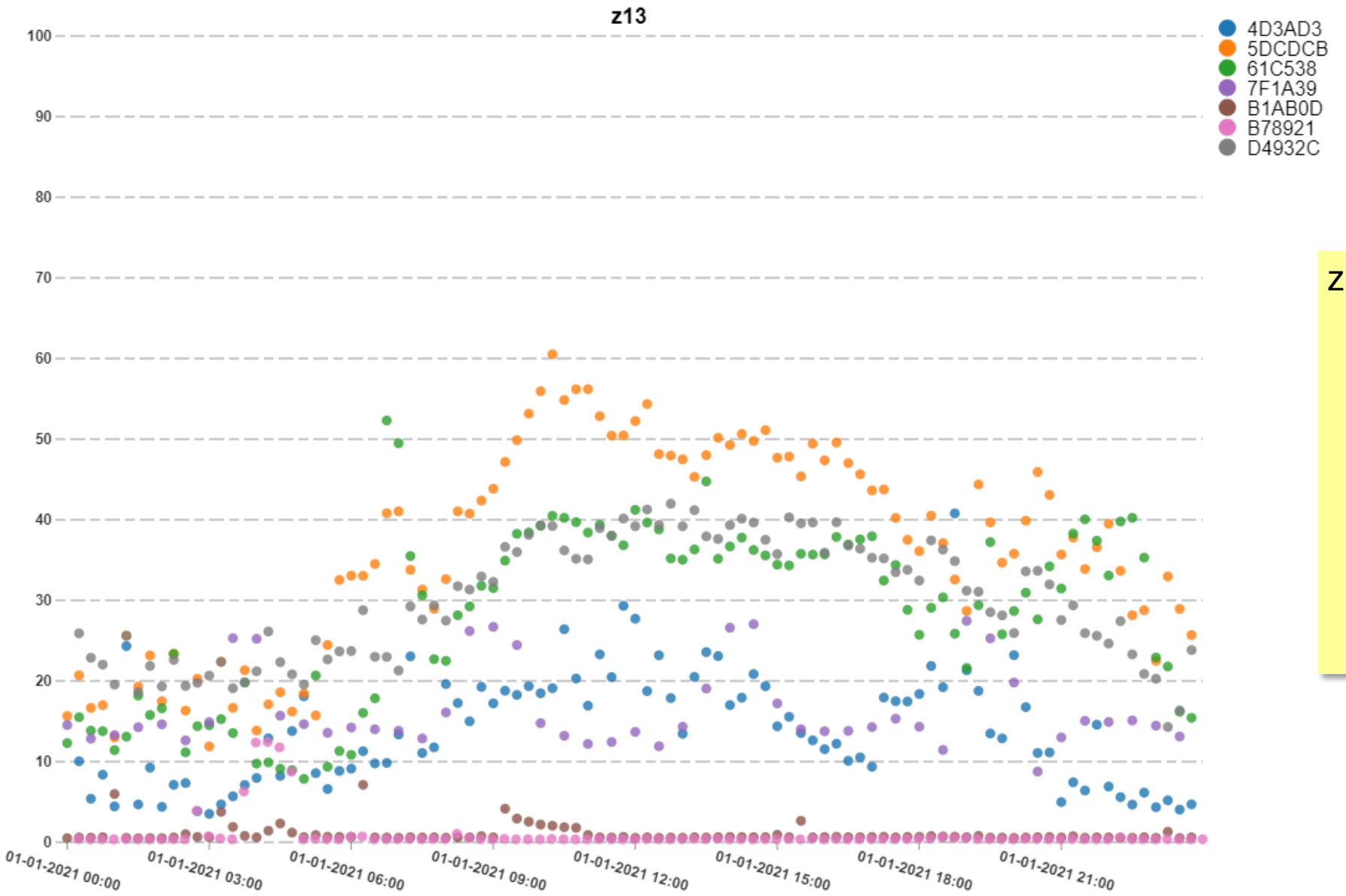
CEC Busy

General Purpose Engines



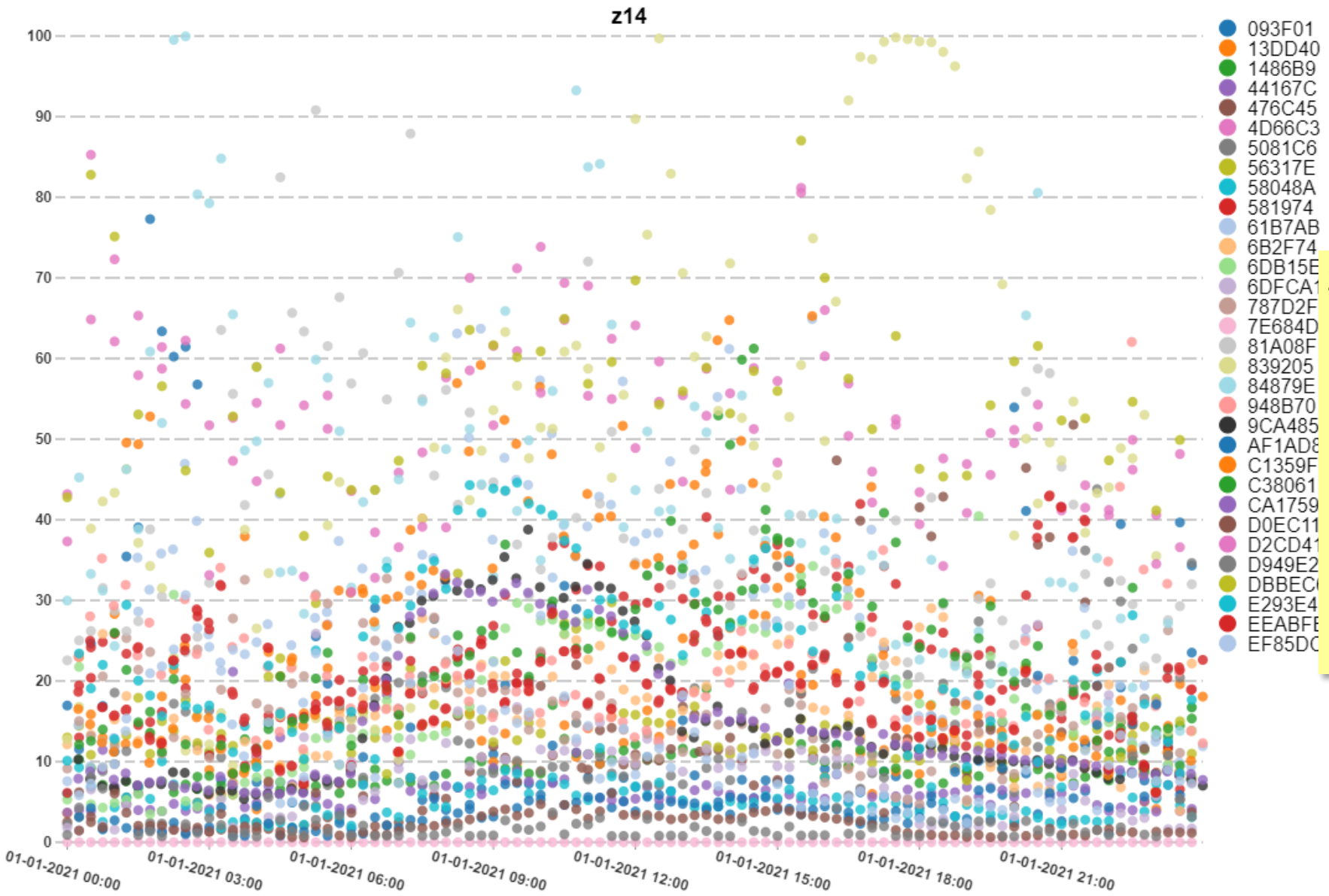
z15 CECs seem to follow the same trend.

CEC Busy - zIIPs



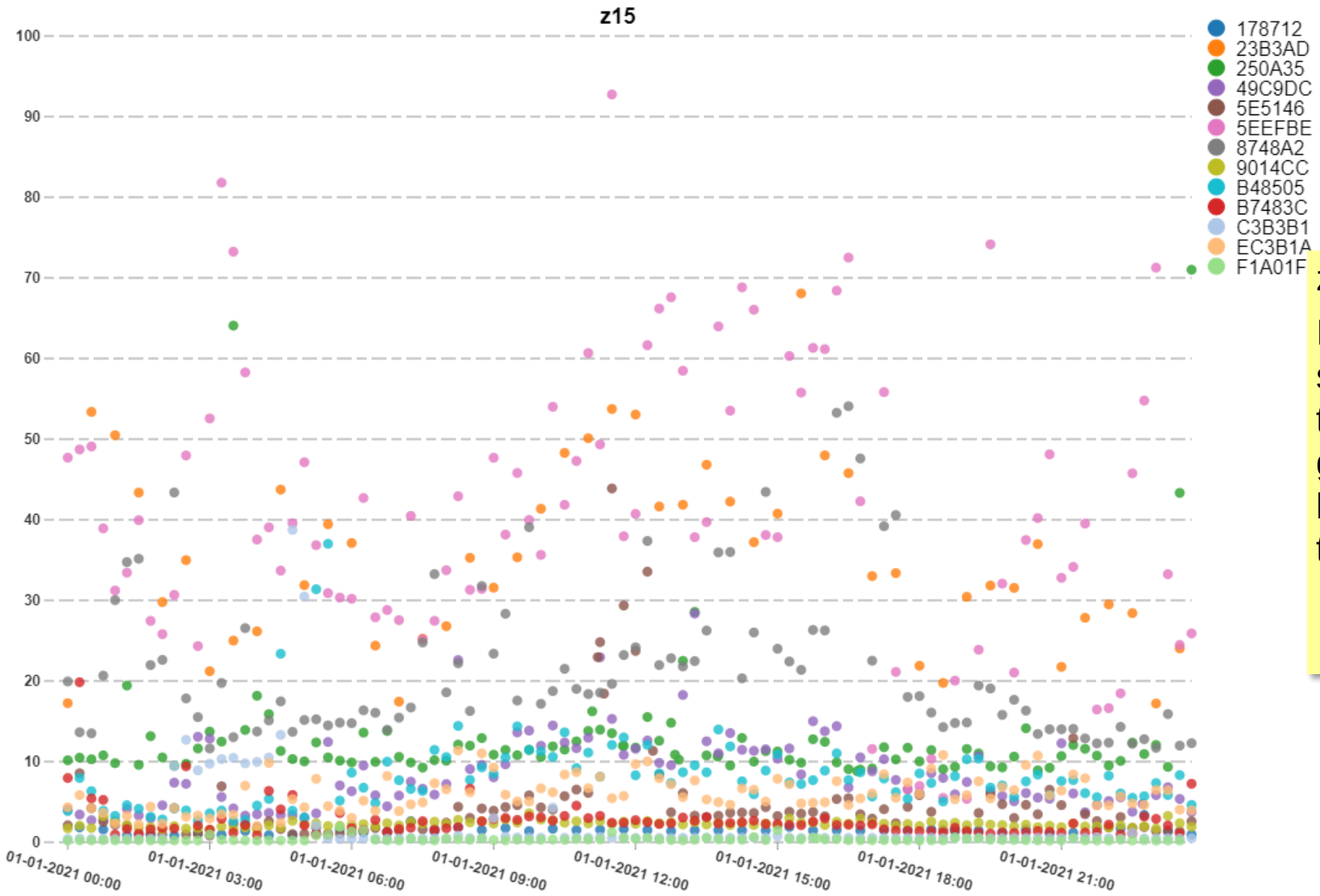
z13 zIIPs

CEC Busy - zIIPs



z14 zIIPs

CEC Busy - zIIPs



z15 zIIPs

Interesting that these seem somewhat lower than the prior two generations. They do keep getting faster though.

CEC Busy: What you should do

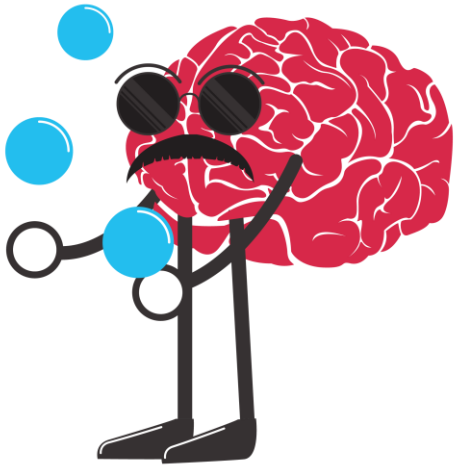
- Don’t be afraid to run less than 100% busy
  - In general, there are efficiency and performance benefits from running less than 100% busy
  - When you’re looking at 15 minute intervals, don’t forget those are averages over the 15 minutes: there will be time periods within that 15 minute interval where the machine is busier
- zIIPs are often run less busy than GCPs, but they also can be pushed to high utilization levels
- 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
  - Usually this needs to be a peak R4HA analysis
    - Except for TFP sites, where 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 may be better than fewer/faster!

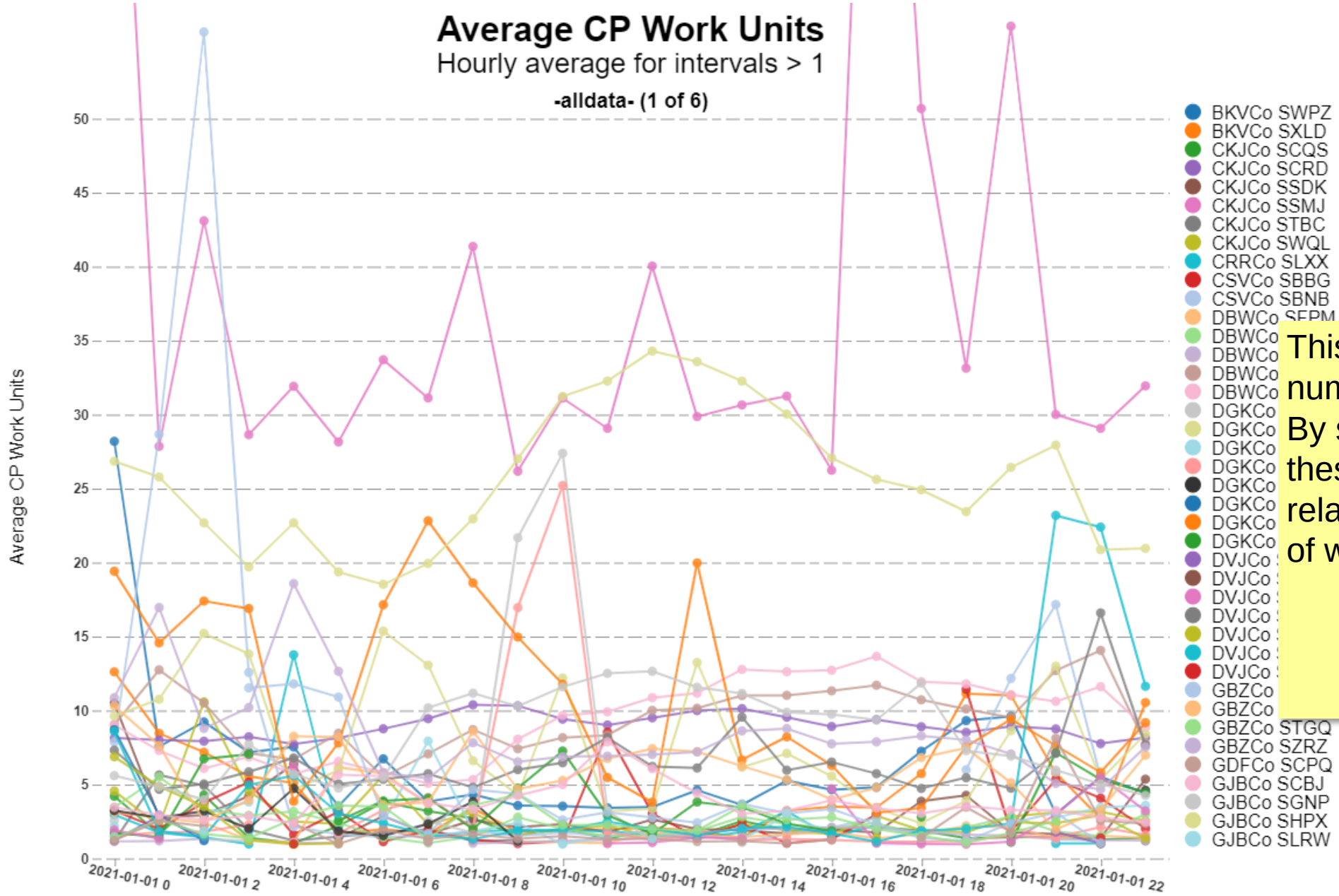
How many balls can you juggle?

Work Units

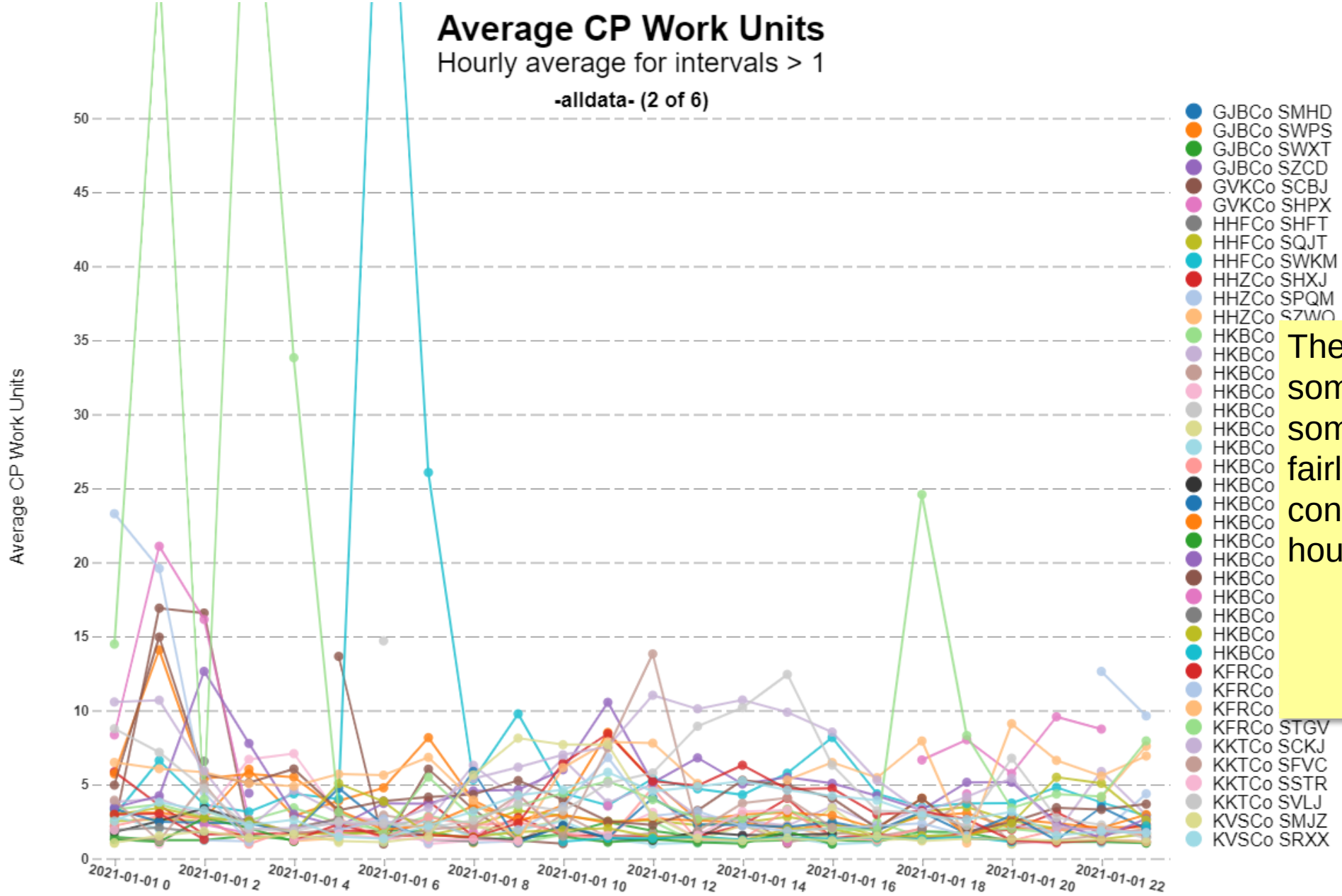


Work Units: Why do you care

- 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. By system. So some of these systems have relatively long queues of work.



These systems are somewhat better, but some of them still have fairly high averages consider these are hourly averages.

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

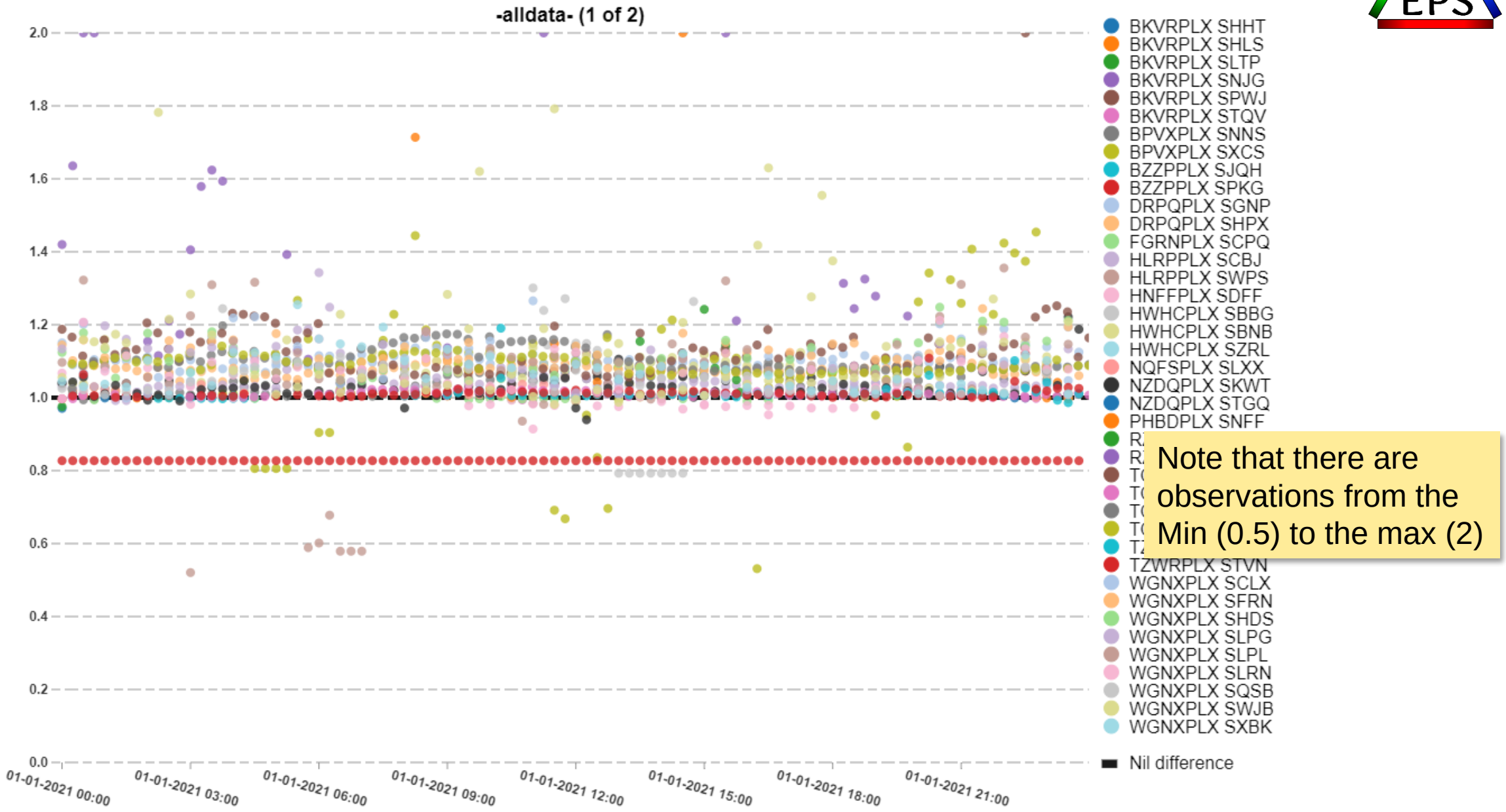
Do more, slower

SMT

SMT: Why do you care

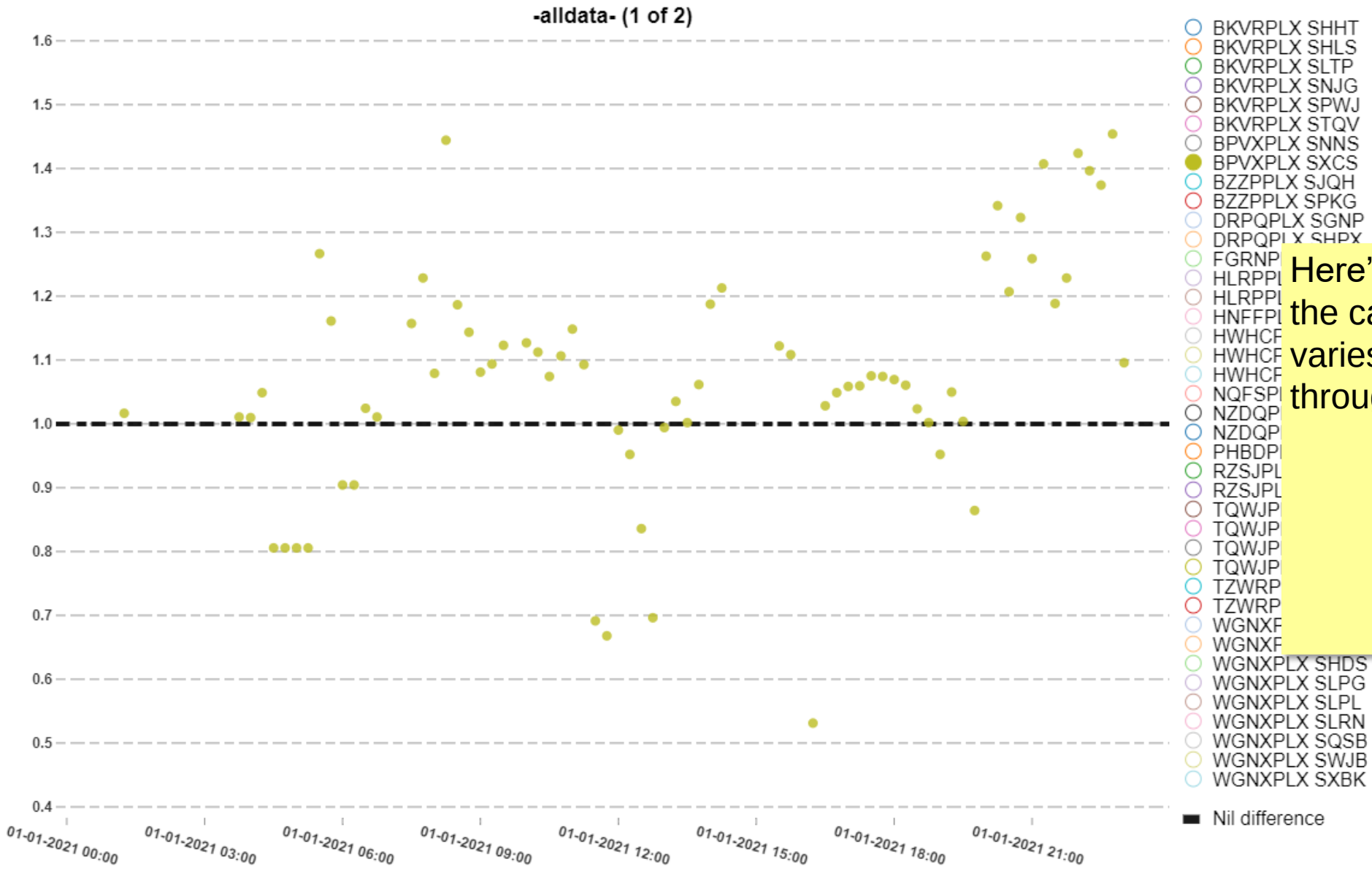
- SMT may be something useful to investigate in particular situations
  - See my SMT presentation tomorrow for more details!
- These SMT measurements are notoriously variable, so treat these charts with even more caution than the others herein
  - Your results will vary
- Capacity Factor is generally held up as the metric to say whether you're getting net benefit from SMT or not
  - Estimate of ratio of total work done with SMT vs. total work without SMT
  - CF will be between 0.5 and 2.0

SMT: Capacity Factor



Note that there are observations from the Min (0.5) to the max (2)

SMT: Capacity Factor



Here's a system where the capacity factor varies a fair bit throughout the day.

SMT: What you should do

- If you have a good reason, try it
  - Along with SMT measurements, be sure to check your application responsiveness
  - Be wary of the SMT measurements, most especially at non-busy times
- If you don’t have a defined reason, don’t worry about enabling it
- SMT makes zIIP capacity planning more difficult
- zIIP consumption measurements change with SMT enabled

Numbers for geeks

Hardware Instrumentation Services

Hardware Instrumentation Services: Why you care

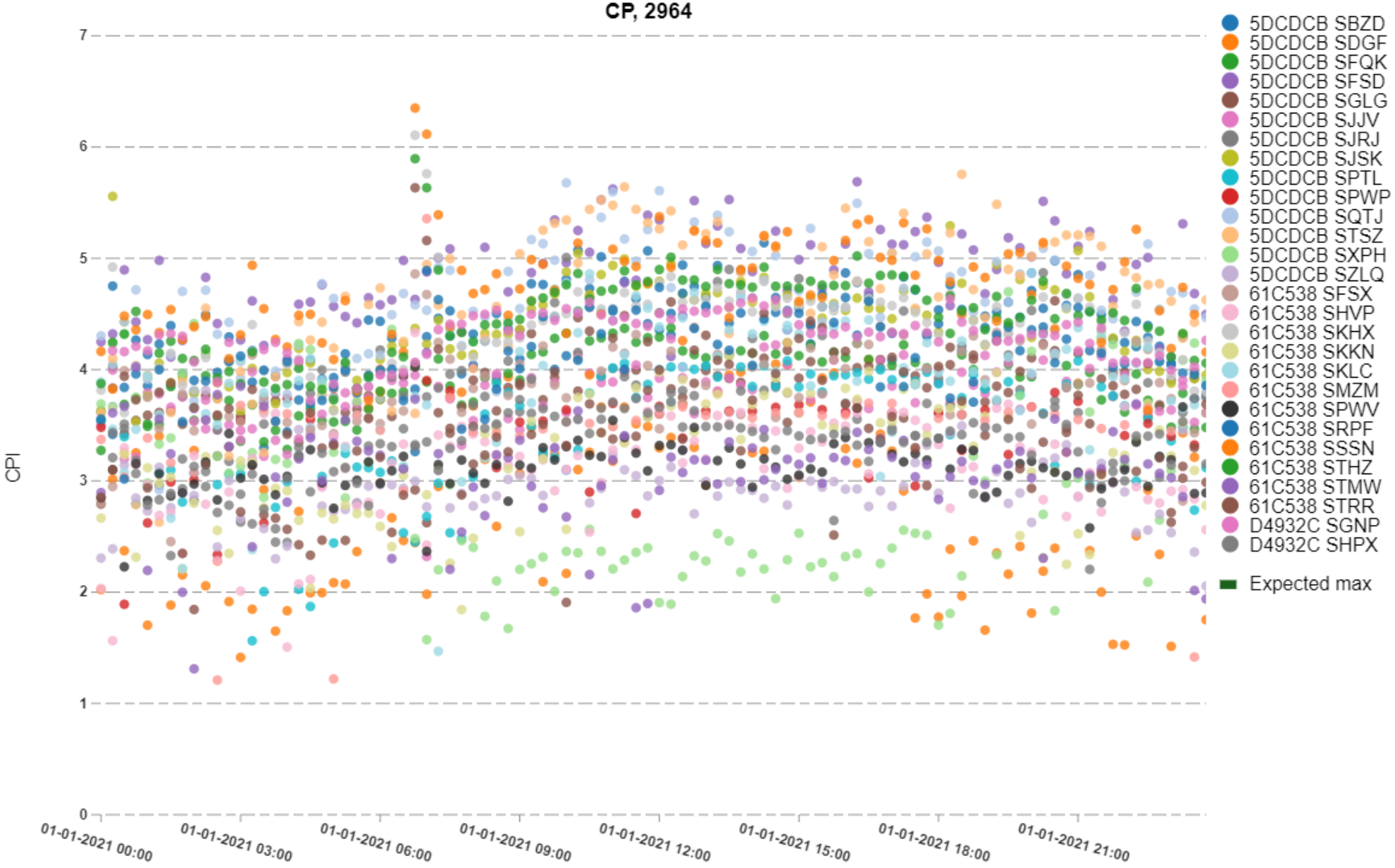
- Collecting the SMF 113 HIS records is mandatory for doing proper planning for a new processor
- The information is interesting to understand the characteristics of your workload and how efficiently it's utilizing the hardware resources
- The primary numbers of interest (lower=better in all cases):
  - 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

CP, 2964

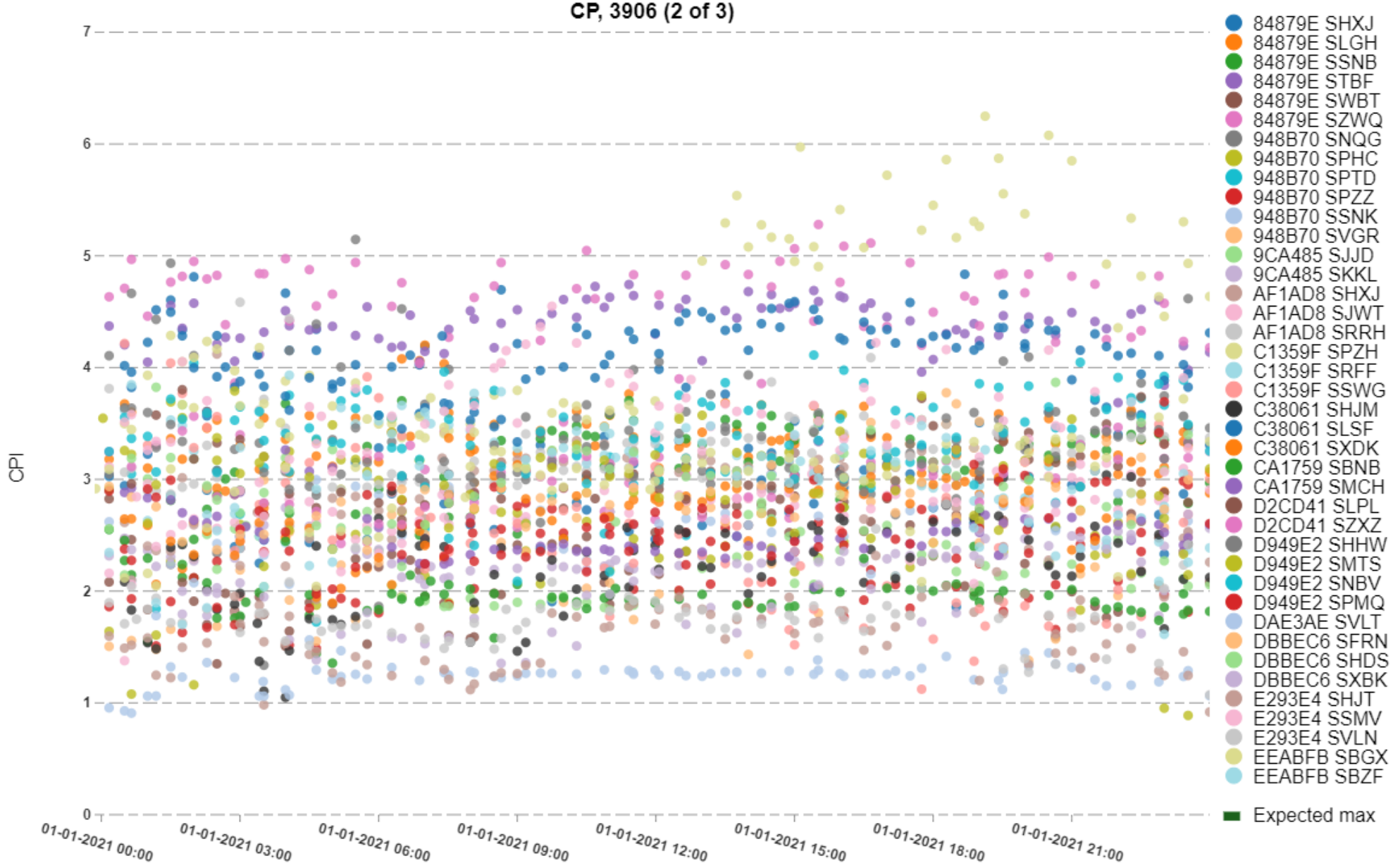


Z13 GCP

Cycles Per Instruction

By z/OS Hardware Model

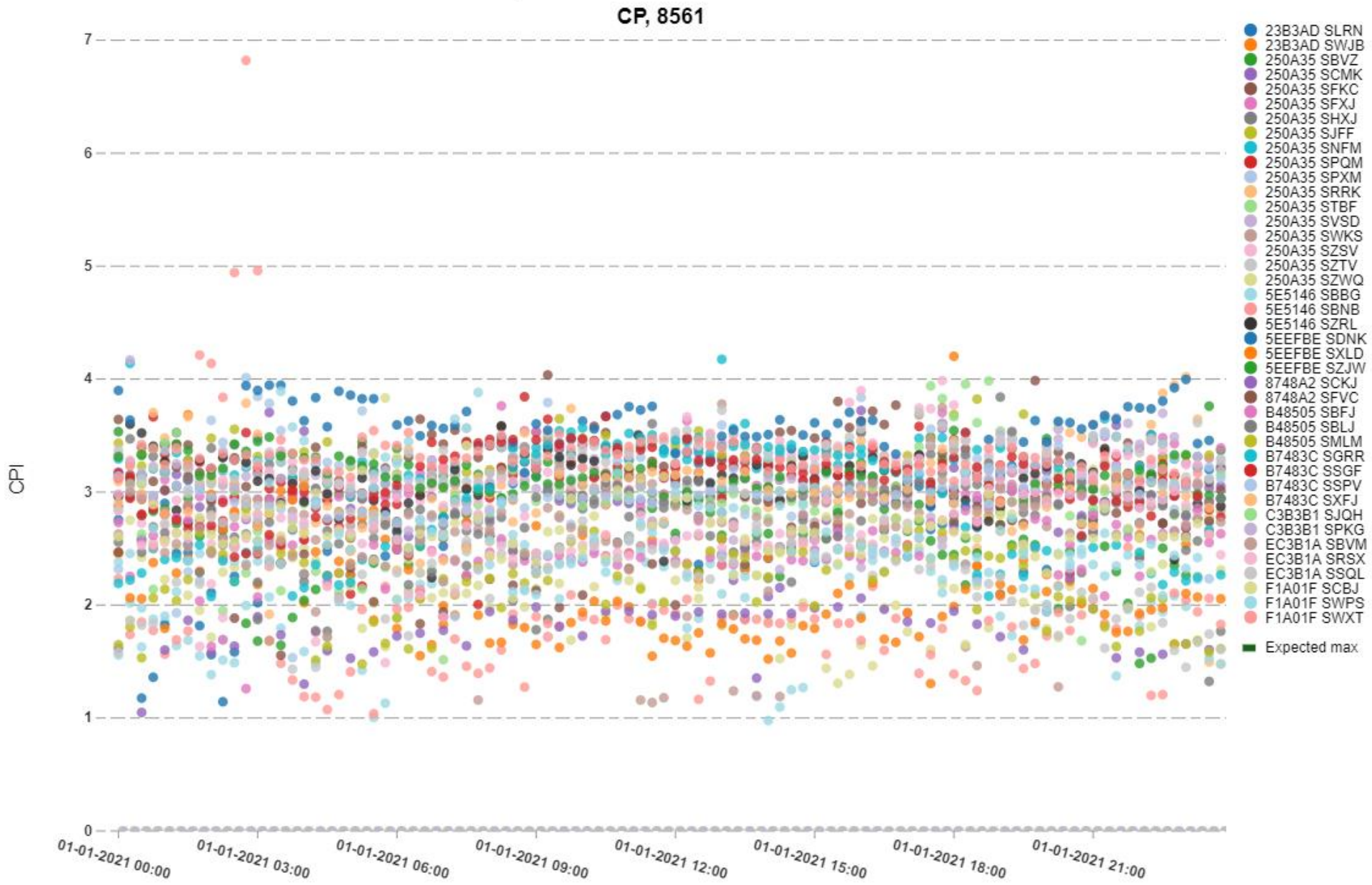
CP, 3906 (2 of 3)



z14 GCP

Cycles Per Instruction

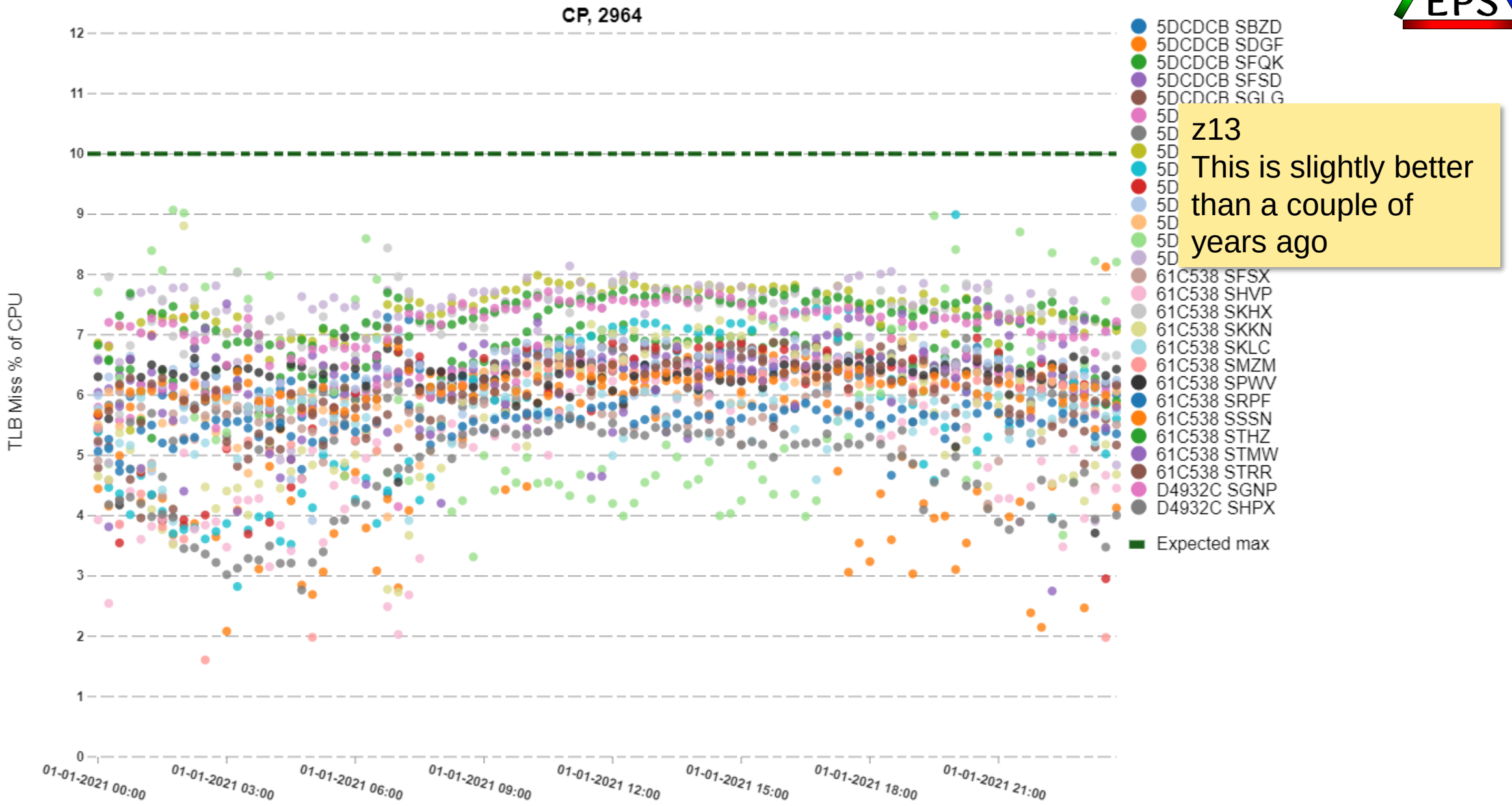
By z/OS Hardware Model



z15 GCP

TLB Miss CPU %

By z/OS Hardware Model

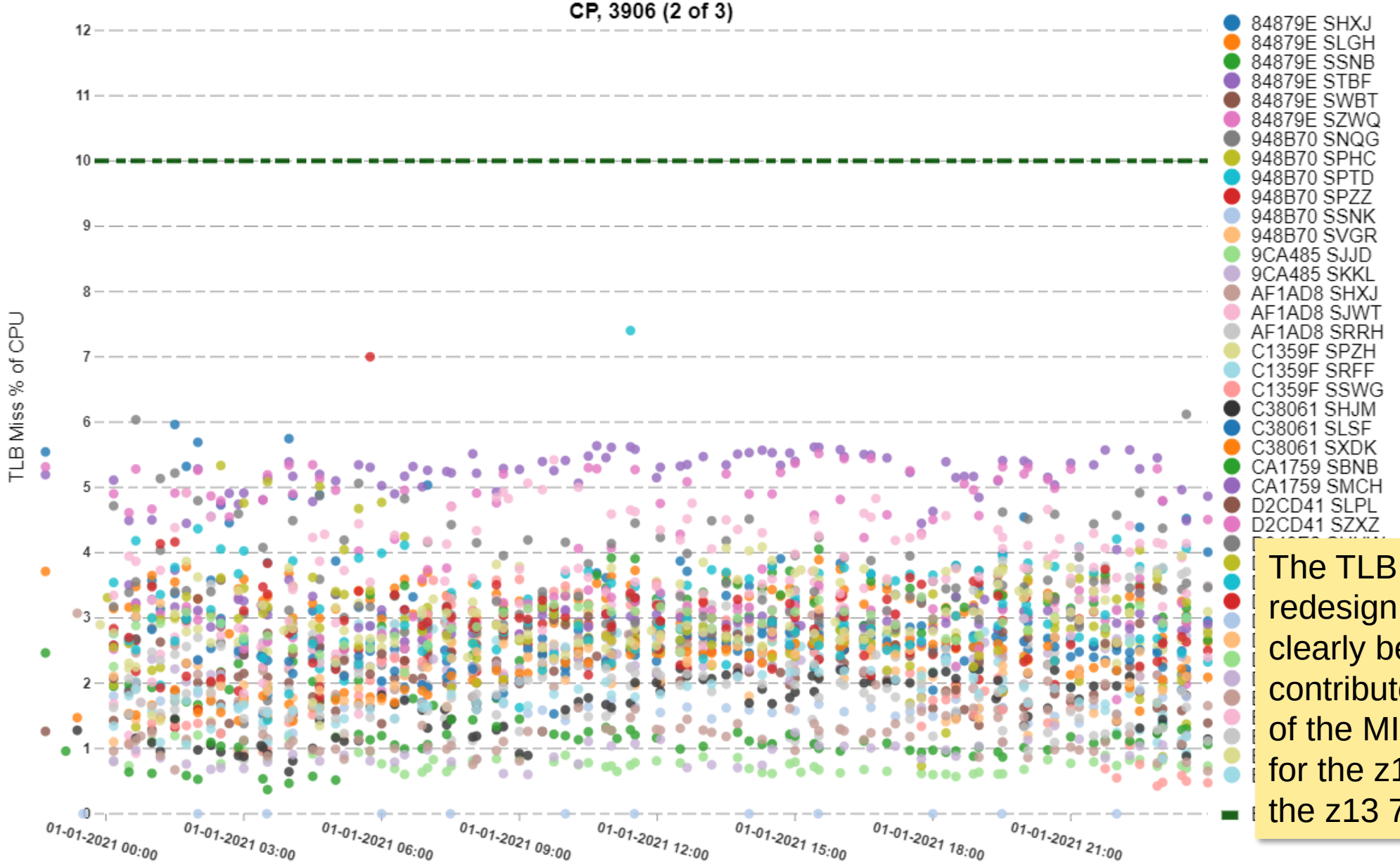


z13
This is slightly better than a couple of years ago

TLB Miss CPU %

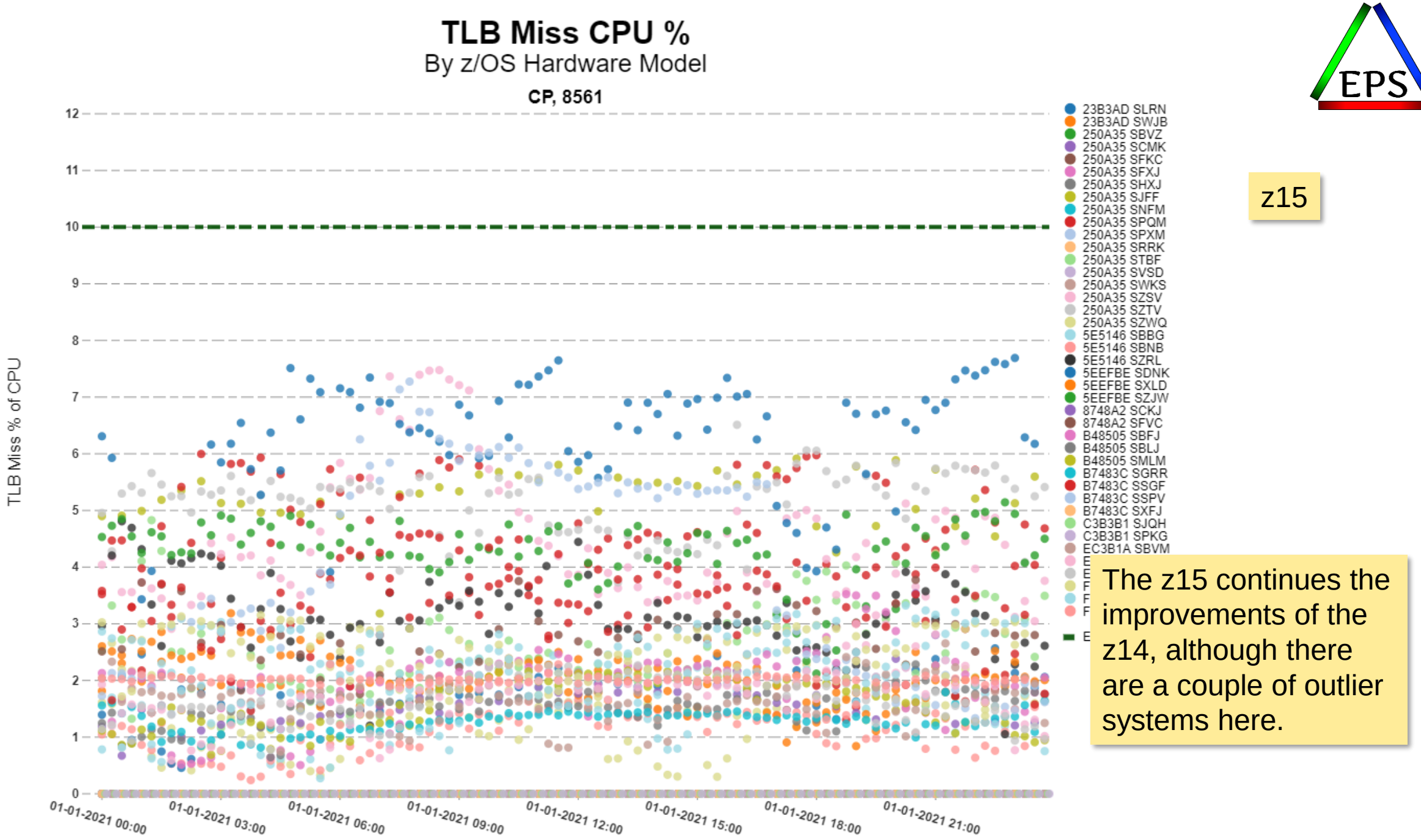
By z/OS Hardware Model

CP, 3906 (2 of 3)



z14

The TLB and DAT redesign on the z14 is clearly beneficial! This contributes about half of the MIPS increase for the z14 7xx over the z13 7xx.



z15

The z15 continues the improvements of the z14, although there are a couple of outlier systems here.

HIS: What you should do

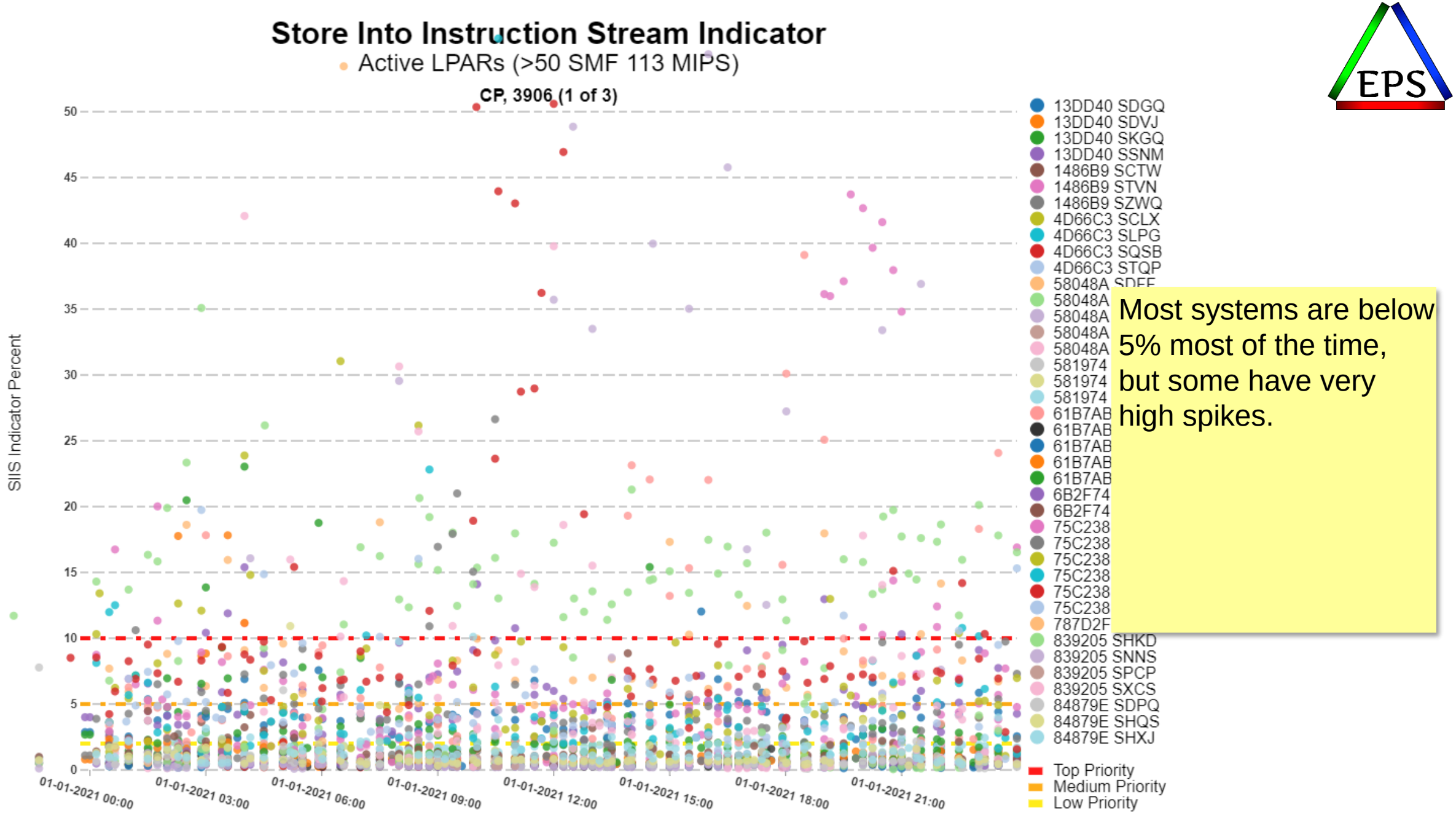
- If your numbers seem high, that may be the nature of your workload, but there are a few configuration choices that do affect these numbers
- HiperDispatch should be enabled in about 95%+ of the cases
  - This can help L1MP and CPI
- Consider more/slower vs. fewer/faster CPs
  - More CPs = more L1/L2 cache = lower L1MP = lower CPI
- Use large pages where you can
  - Can reduce the TLB Miss % CPU
  - z14 greatly reduces this TLB Miss
- If you have lots of large JVMs, choice of heap sizes and garbage collection configuration may influence cache effectiveness (RNI) and TLB Miss % of CPU
- Evaluate zIIPs and CPs separately when calculating your workload hint for zPCR

Do you really want to jump into that stream?

Store Into Instruction Stream

SIIS: Why you care

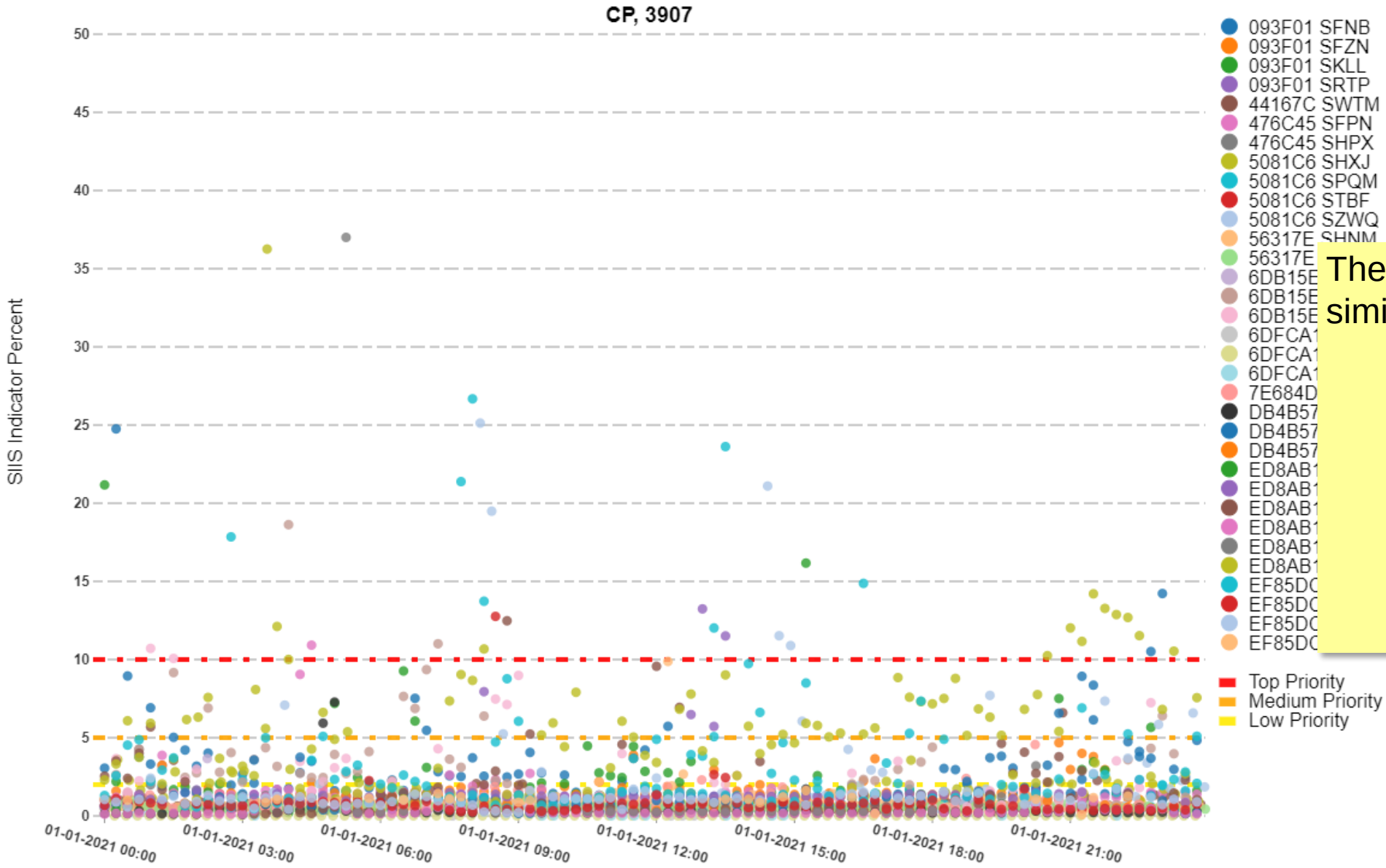
- Store Into Instruction Stream (SIIS) describes a situation where code writes to memory that is within 1 cache line (256 bytes) of the executing instructions
- That update will trigger a flush of that cache line from the L1 Instruction cache
- This can be a significant performance hit!
  - Pipeline 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”



Most systems are below 5% most of the time, but some have very high spikes.

Store Into Instruction Stream Indicator

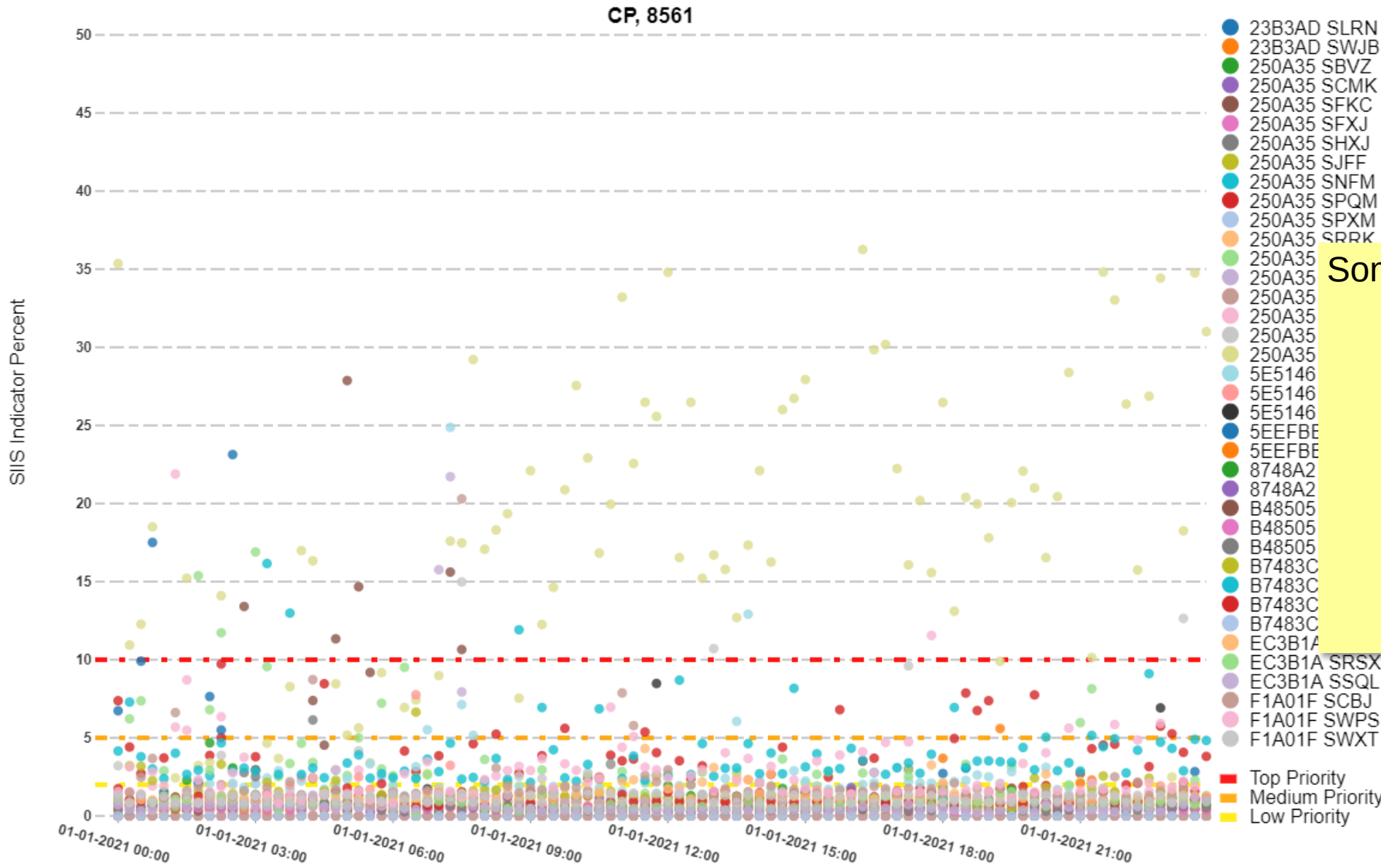
Active LPARs (>50 SMF 113 MIPS)



These systems follow a similar pattern.

Store Into Instruction Stream Indicator

Active LPARs (>50 SMF 113 MIPS)



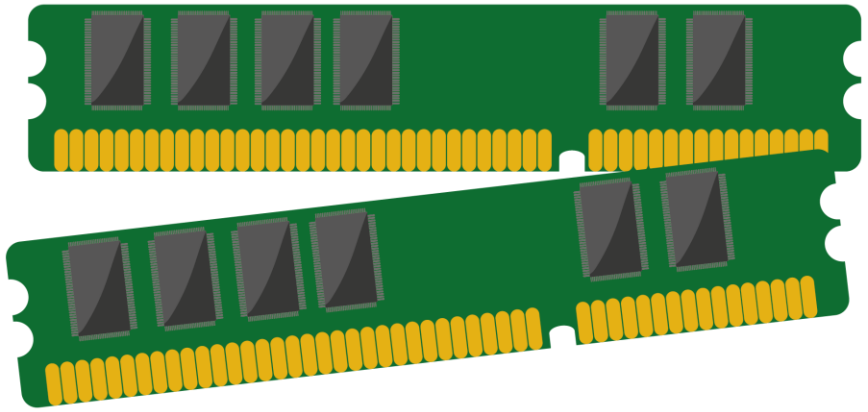
Some z15 systems

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
    - I'm scheduled to explore this in an upcoming webinar

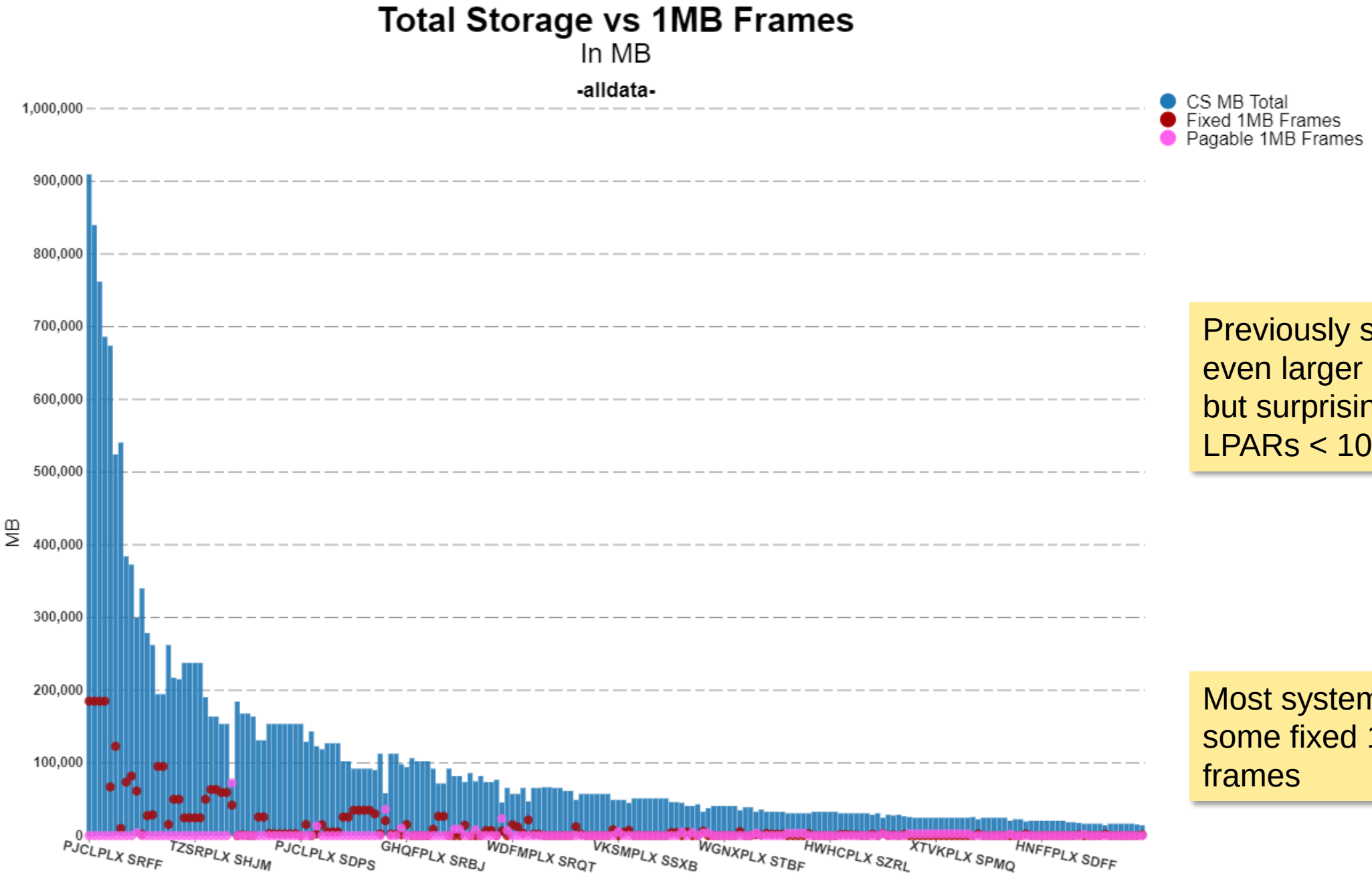
If only I could remember…

Memory



Memory: Why you care

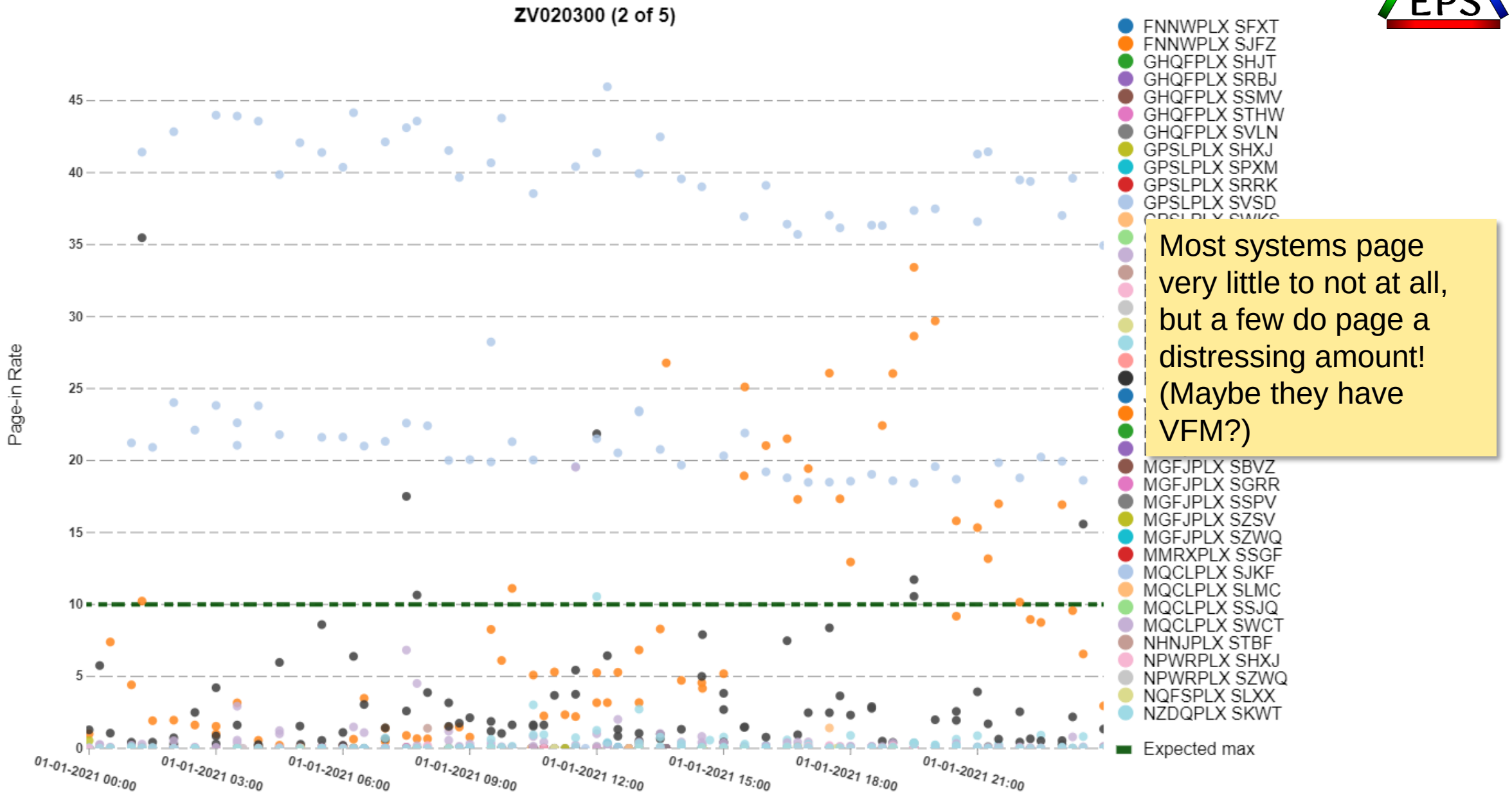
- Keeping data as close to the processor as possible is the key to optimizing performance
  - Larger memory sizes means it's more plausible to keep more data in memory
- Use of large 1MB frames can help lower TLB Miss % CPU
- Paging has a significant performance impact
  - FlashExpress can mitigate *some* 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



Previously saw some even larger LPARs, but surprisingly many LPARs < 100GB

Most systems using some fixed 1MB frames

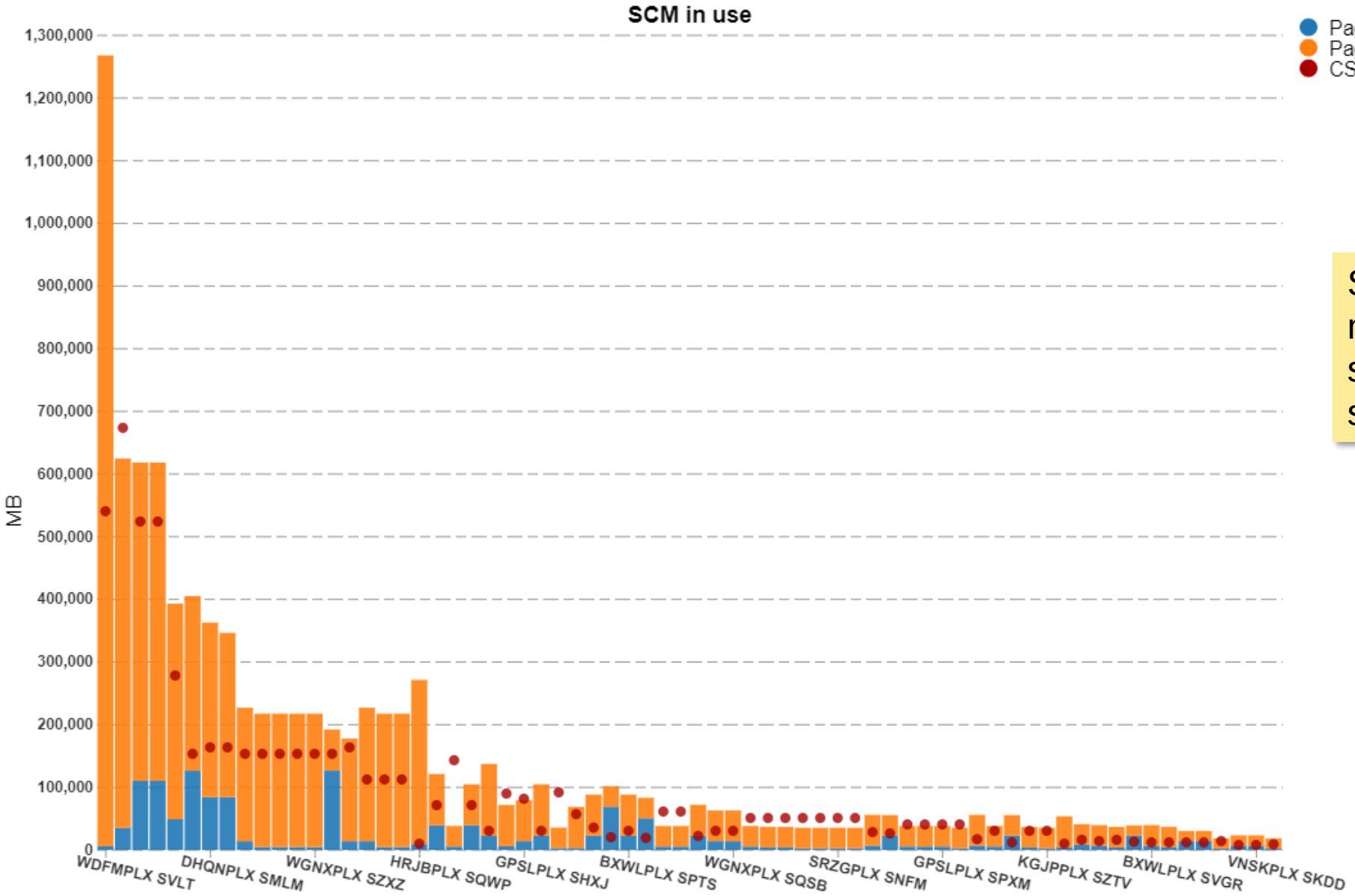
Page-In Rate



Most systems page very little to not at all, but a few do page a distressing amount! (Maybe they have VFM?)

Total Page Space Relative to Central Storage

In MB



Systems with SCM may have more paging space than central storage.

Memory: What you should do

- 1MB Pages are commonly in use, if you've been trying to avoid being the pioneer, you're now safe to forge ahead
  - 2GB pages not yet common, but I suspect we'll start to see more of that with these LPARs that are hundreds of GBs
- Avoid paging: if you're regularly paging in more than a couple of pages a second, you're certainly in the 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

Forget spinning

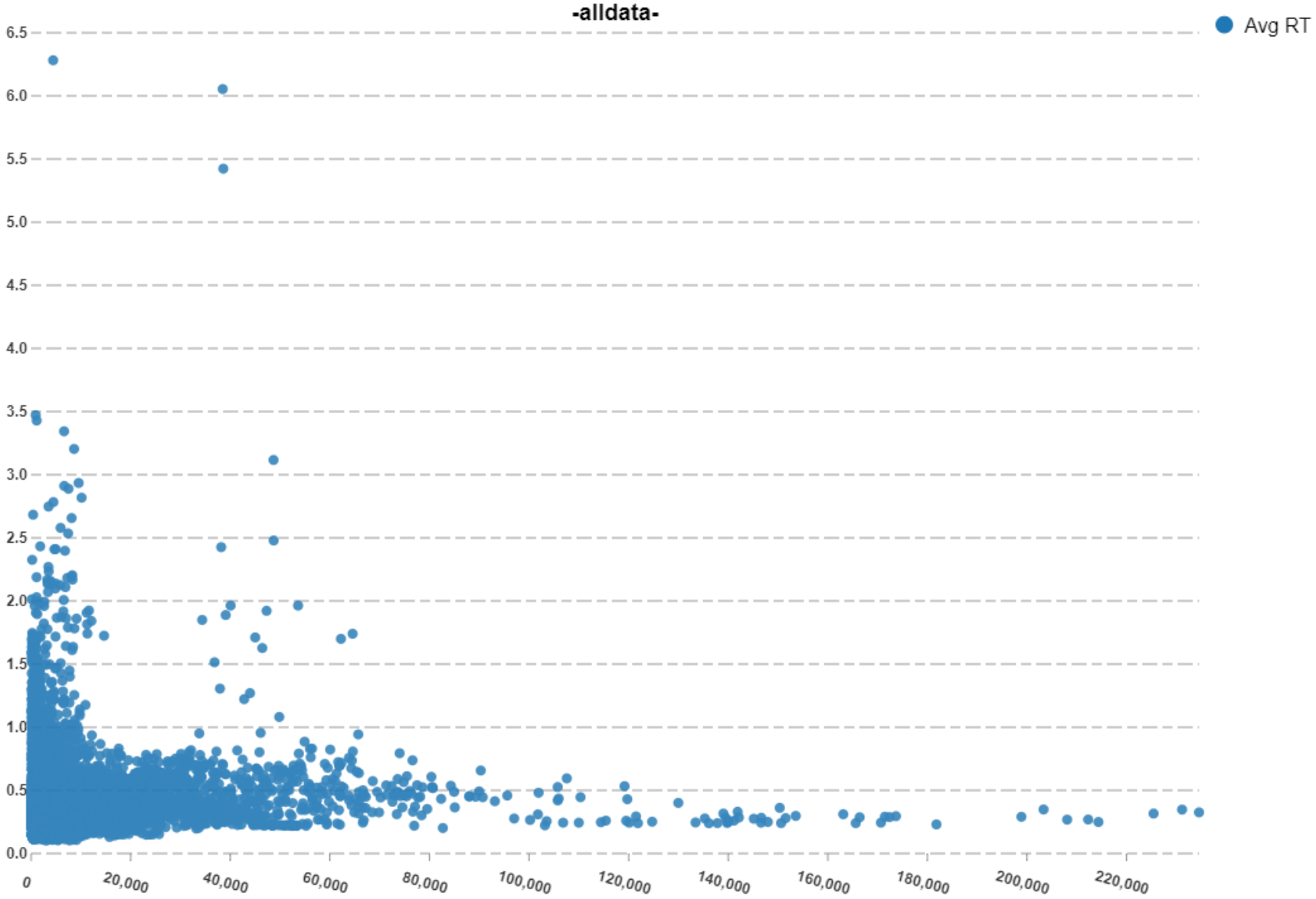
DASD Response Time

DASD Response time components: Why you care

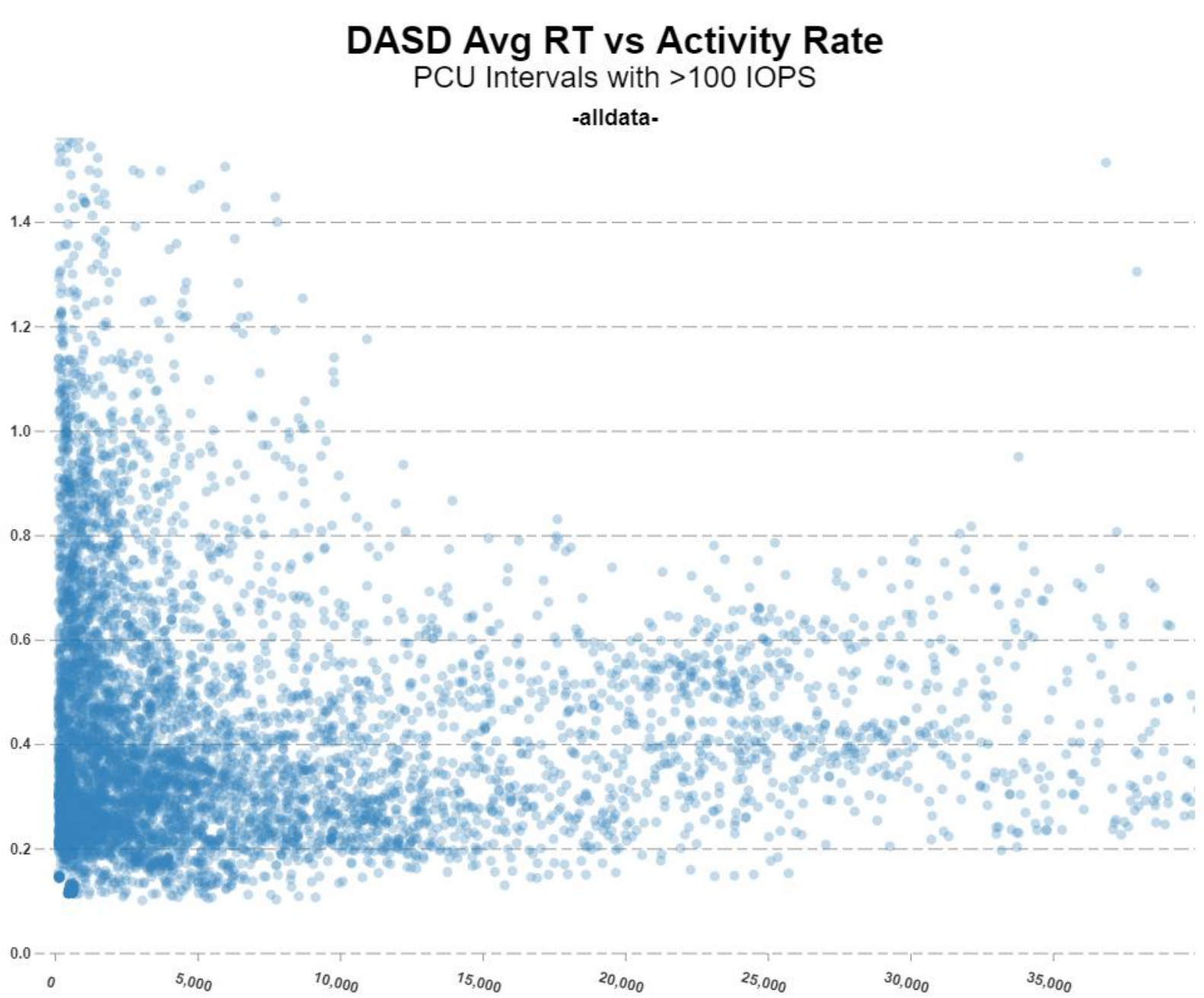
- Because we see so much SSD storage, I/O response time is somewhat less of a concern than in ages gone by
  - Larger central storage sizes can also be leveraged to reduce I/Os
- But even in the age of sub-millisecond I/Os, the only good I/O is no I/O
  - A 1 millisecond response time is a whole lot 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

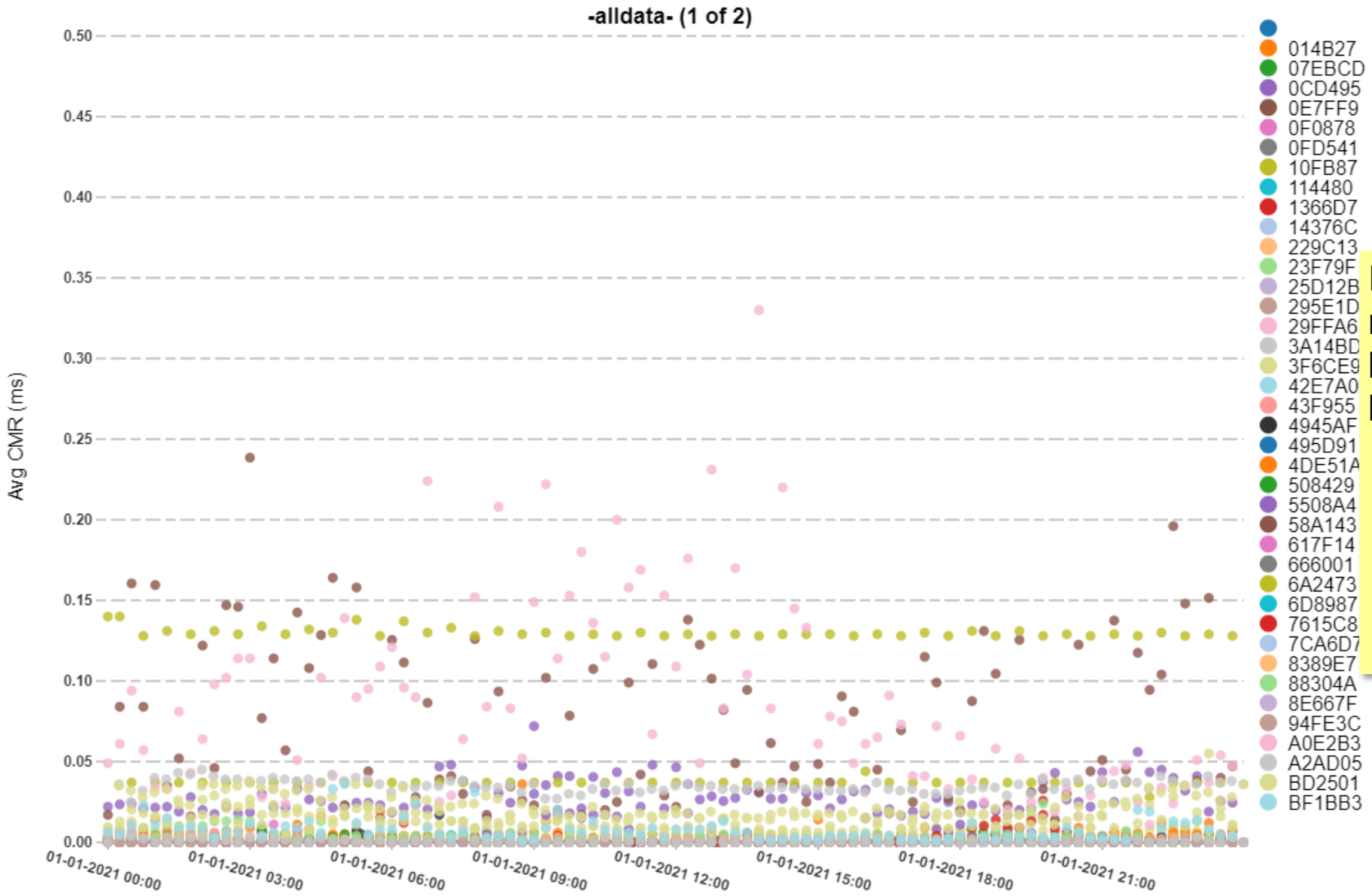


Long tails on both axes, but vast majority of response times are <1 ms, even under 0.5ms may be a majority.



Zoomed in, we see indeed that it does look like the majority of the cases are indeed under 0.5ms.

DASD Avg Command Response Time over Time

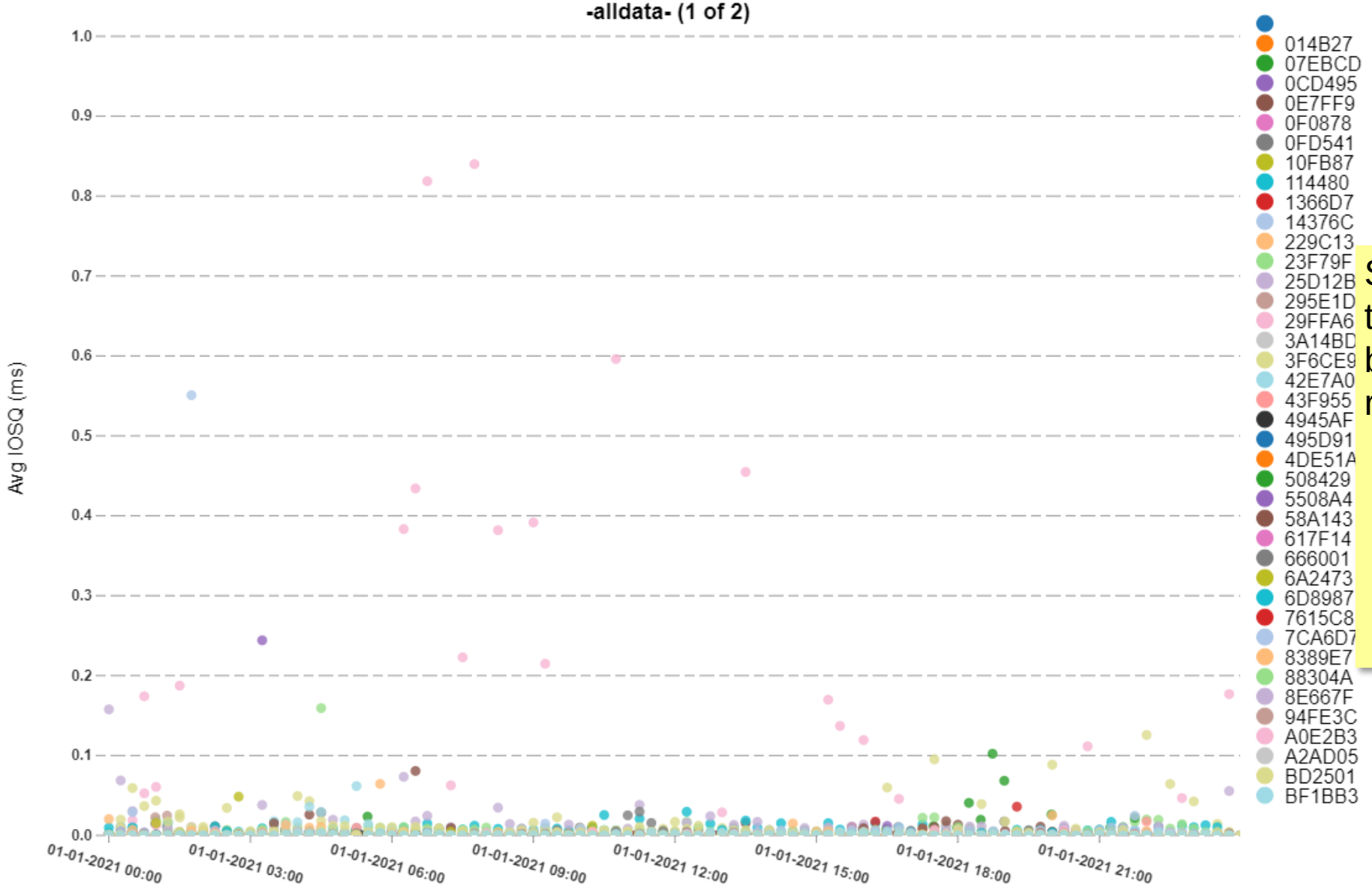


Initial command response time now largely immaterial: below 0.05ms (50**µs**)

DASD Avg IOSQ over Time

For Avg RTs < 10ms

-alldata- (1 of 2)



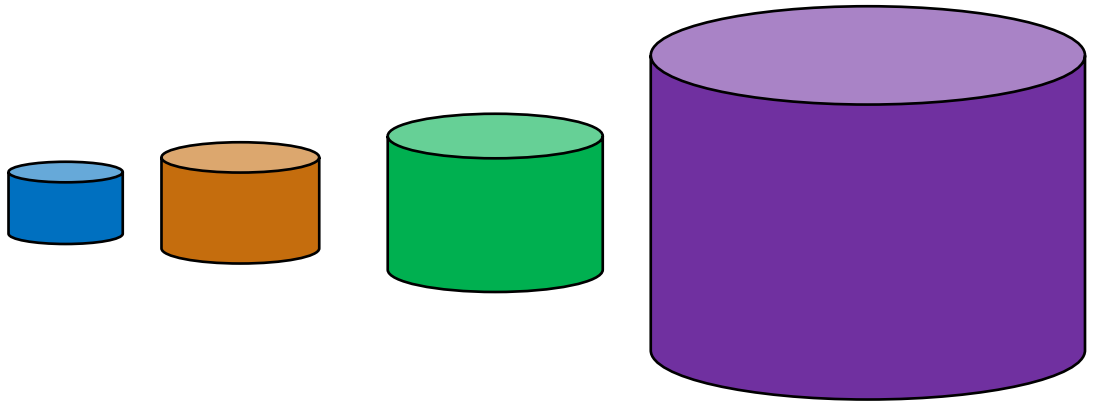
Seeing even less IOSQ today, probably because of SuperPAV rolling out.

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 IOSQ time is regularly more than 0.1ms 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
- If Initial Command Response is high, is it an older controller?
  - Possibly look for overloading/balance issues on the host adapters
- If overall response time is high
  - Investigate logical volume placement in the PCU (e.g. SSD vs HDD)
  - Is the tiering software working as expected?

More/smaller or fewer/larger?

Volume Size



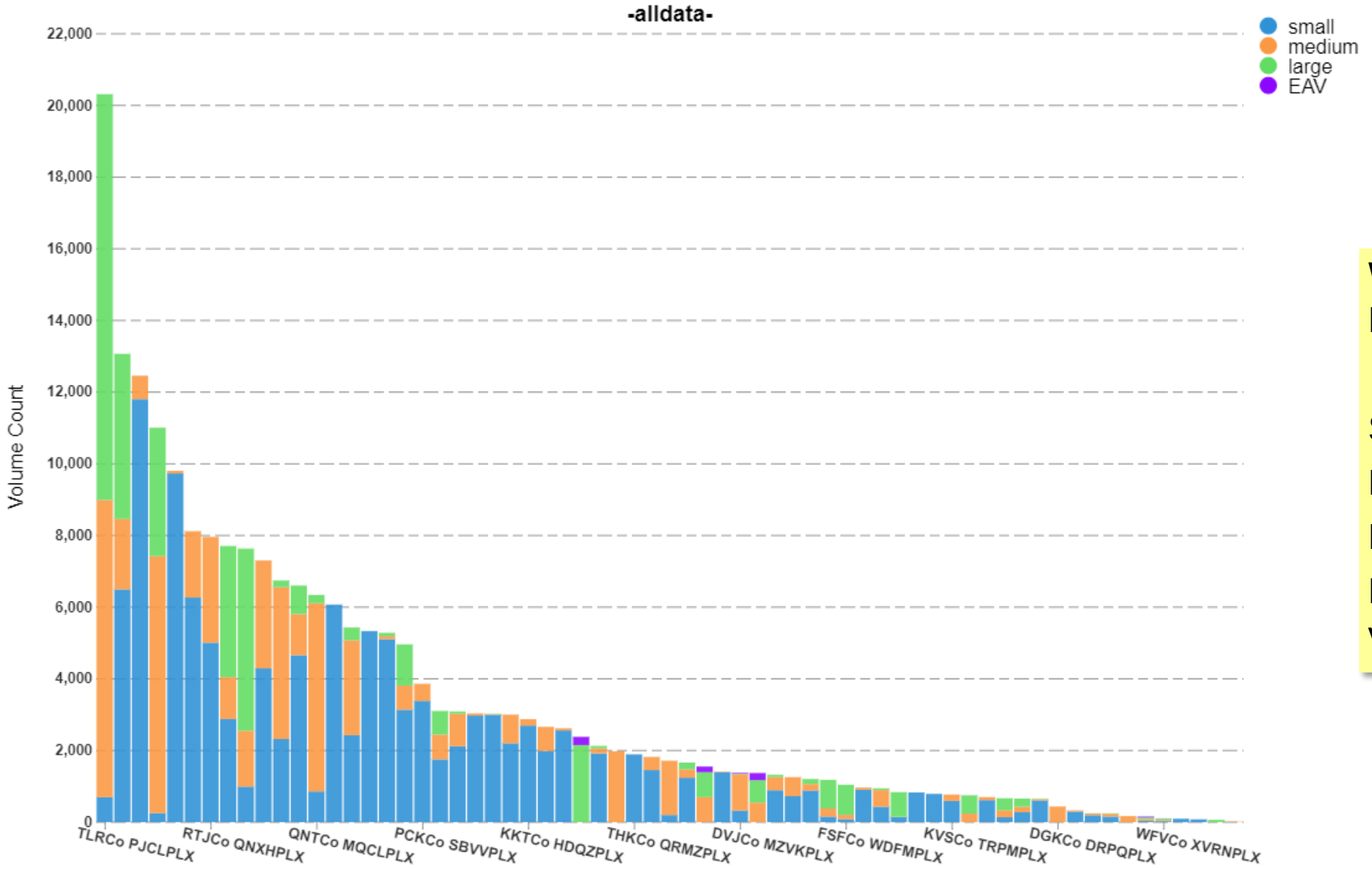
Volume Size: Why you care

- This is *mostly* a systems management issue
  - A collection of fewer things generally easier to manage than lots of things

- But there may be a (generally small) hidden performance issue
  - Allocation times may be faster on pools with a few large volumes vs many small volumes
  - Depending on the free space on the volumes and the size of the allocations
  - Allocation CPU consumption can reduce your capture ratio

DASD Volume Counts by Size

By Sysplex



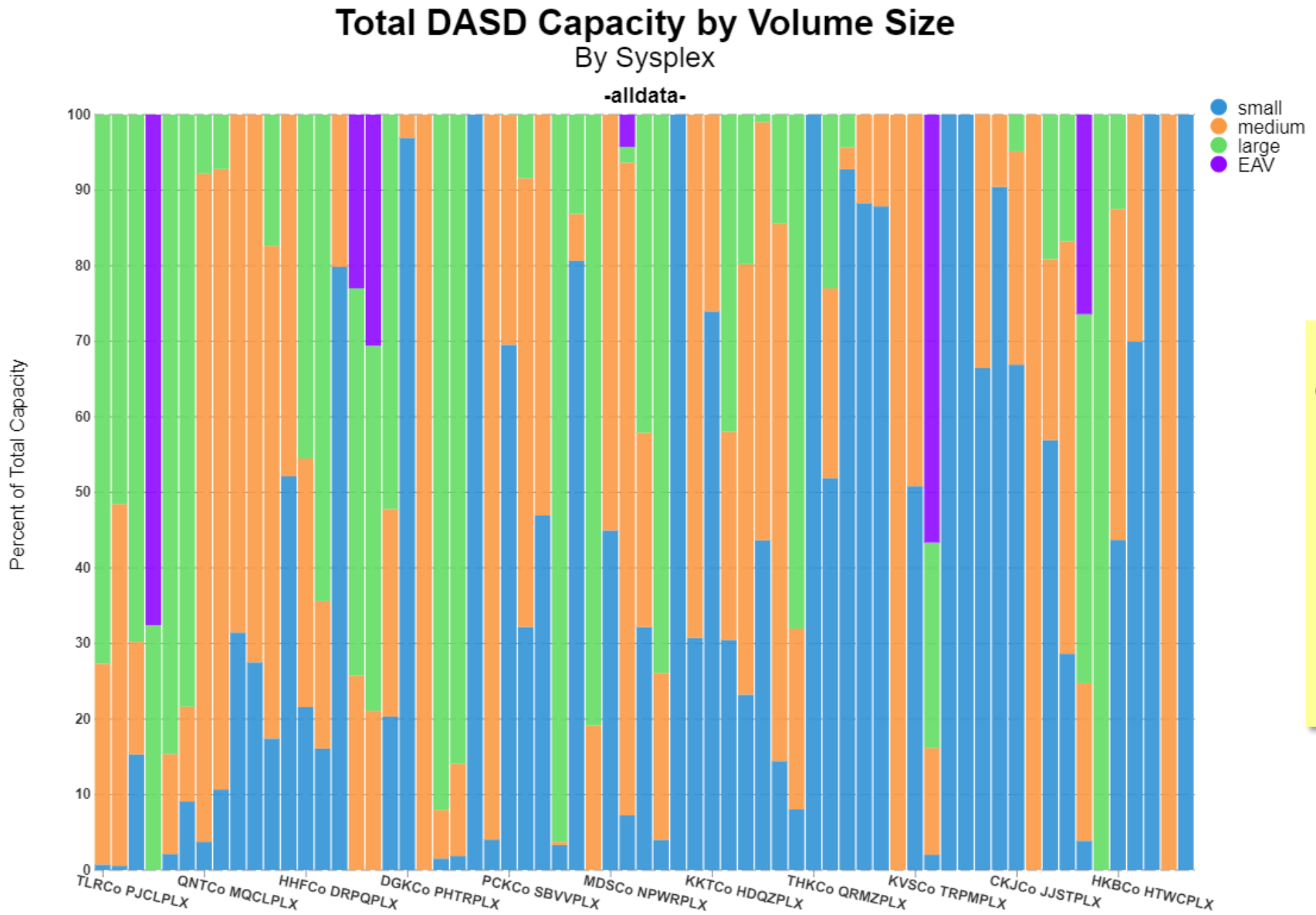
We're starting to see EAVs used!

Small: <= ~Mod-9

Medium: <= ~Mod27

Large: Up to 64K cyls

EAV: Extended Address volumes

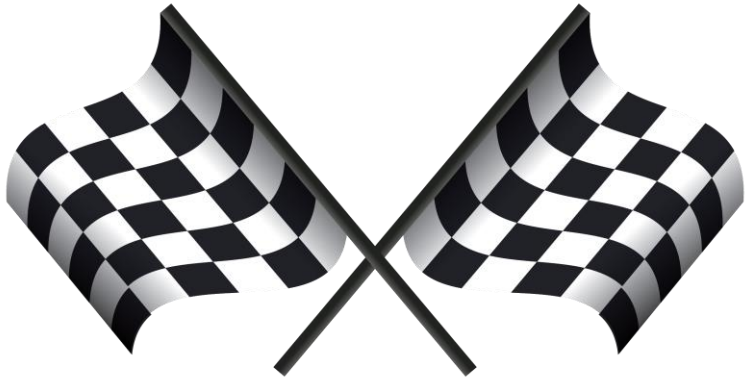


For some plexes, EAVs comprise a significant portion of their storage.

Volume Sizes: What you should do

- With proper use of SuperPAV (and, mostly, even HyperPAV) large volumes avoid the performance problems they had in the past
- Fewer/larger volumes generally will be better for systems management and allocation
- If you have thousands of mod-9 and smaller volumes, consider making those larger the next time you do a DASD migration
  - Or possibly sooner, depending on your controller and how easy it is to change sizes
  - Generally this is a non-trivial migration, but … may be worth it
- Consider trying out EAVs, probably starting in dev/test environments
  - Have seen EAVs up to 1TB each, but around 200GB seems to be a more common size
  - Managing things like backup/recovery for 1TB volumes may be interesting
    - OTOH, lots of non-mainframe systems have volumes >1 TB

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