

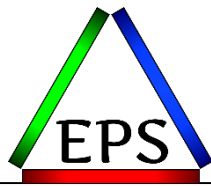
Mainframe Performance Metrics and Observations in the Real World

Scott Chapman
Enterprise Performance Strategies

November 2018
Session **LE**



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:

© Enterprise Performance Strategies, Inc. All rights reserved. No part of this material may be reproduced, distributed, stored in a retrieval system, transmitted, displayed, published or broadcast in any form or by any means, electronic, mechanical, photocopy, recording, or otherwise, without the prior written permission of Enterprise Performance Strategies. To obtain written permission please contact Enterprise Performance Strategies, Inc. Contact information can be obtained by visiting <http://www.epstrategies.com>.

Trademarks:

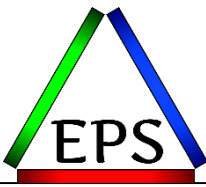
Enterprise Performance Strategies, Inc. presentation materials contain trademarks and registered trademarks of several companies.

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.

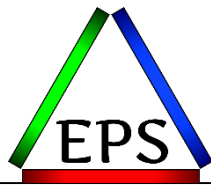
Other trademarks and registered trademarks may exist in this presentation.

Like what you see?



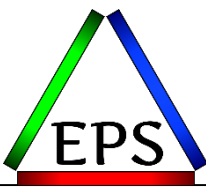
- The z/OS Performance Graphs you see here come from Pivotor™ (although most are not our standard reports)
- If you want a free cursory review of your environment, let us know!
 - We're always happy to process a day's worth of data and show you the results
 - See also: <http://pivotor.com/cursoryReview.html>

EPS presentations this week



What	Who	When	Where
Mainframe Performance Metrics and Observations in the Real World	Scott Chapman	Tue 10:30	Woodcote
z/OS SMT: Understanding the Measurements and Applicability	Scott Chapman	Wed 12:00	Woodcote

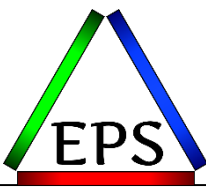
What is this all about?



- 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

Red words are key points to pay attention to

Notes on the data



- Date on the charts will show June 30th 2018, but data was sampled from various times around the middle of 2018
- 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: you shouldn't try comparing the different categorizations by these entirely different workloads

How This Presentation is Arranged

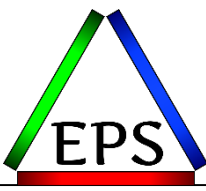


- For each measurement or group of measurements:
 - Why you care: why would you be interested in this measurement
 - The measurements themselves: usually from dozens of systems
 - The charts aren't meant to be read precisely, rather more as a point cloud to see overall trends
 - What you should do: actions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



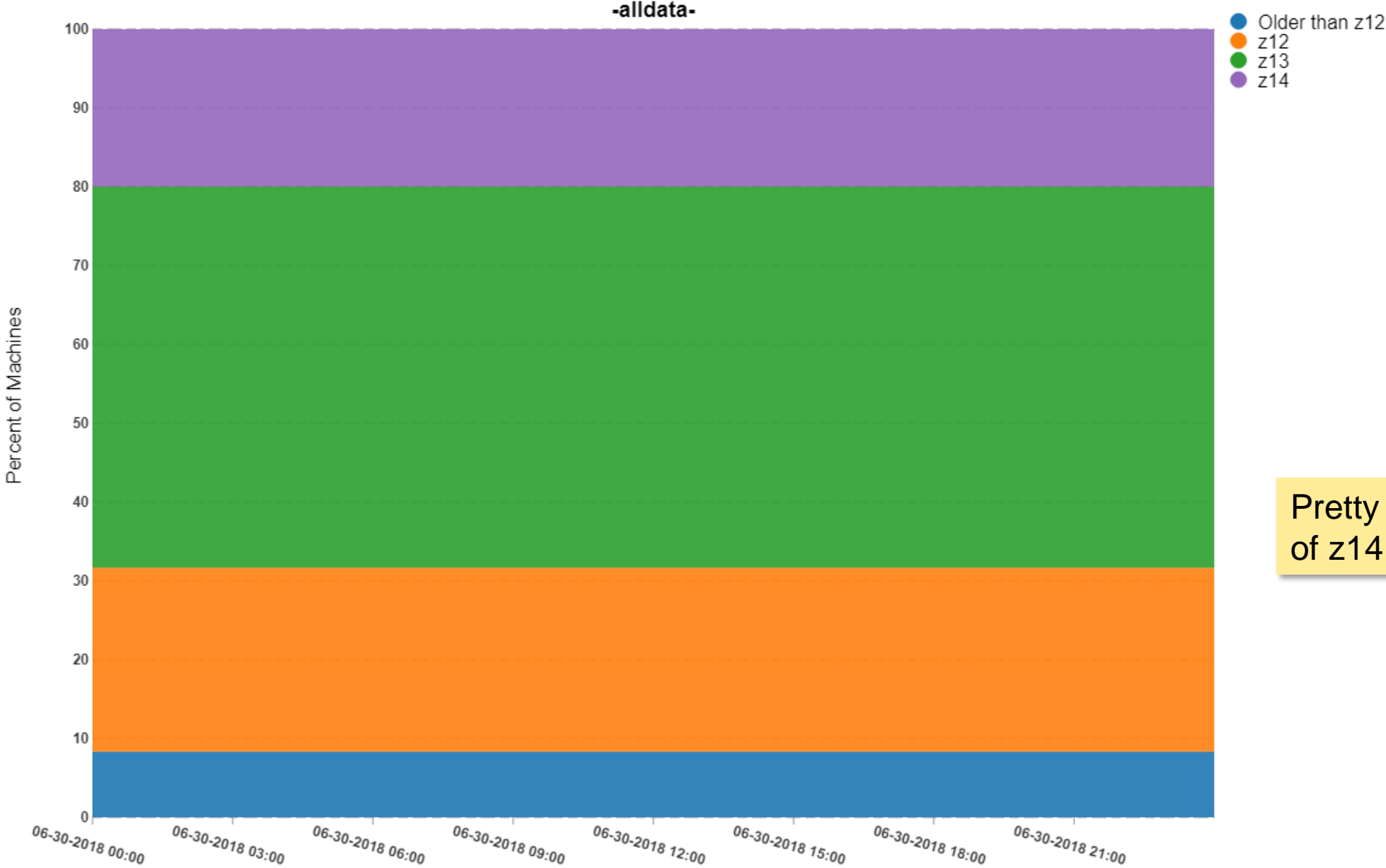
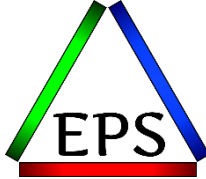
Let's Go!

Machine Generations and z/OS versions: Why you care



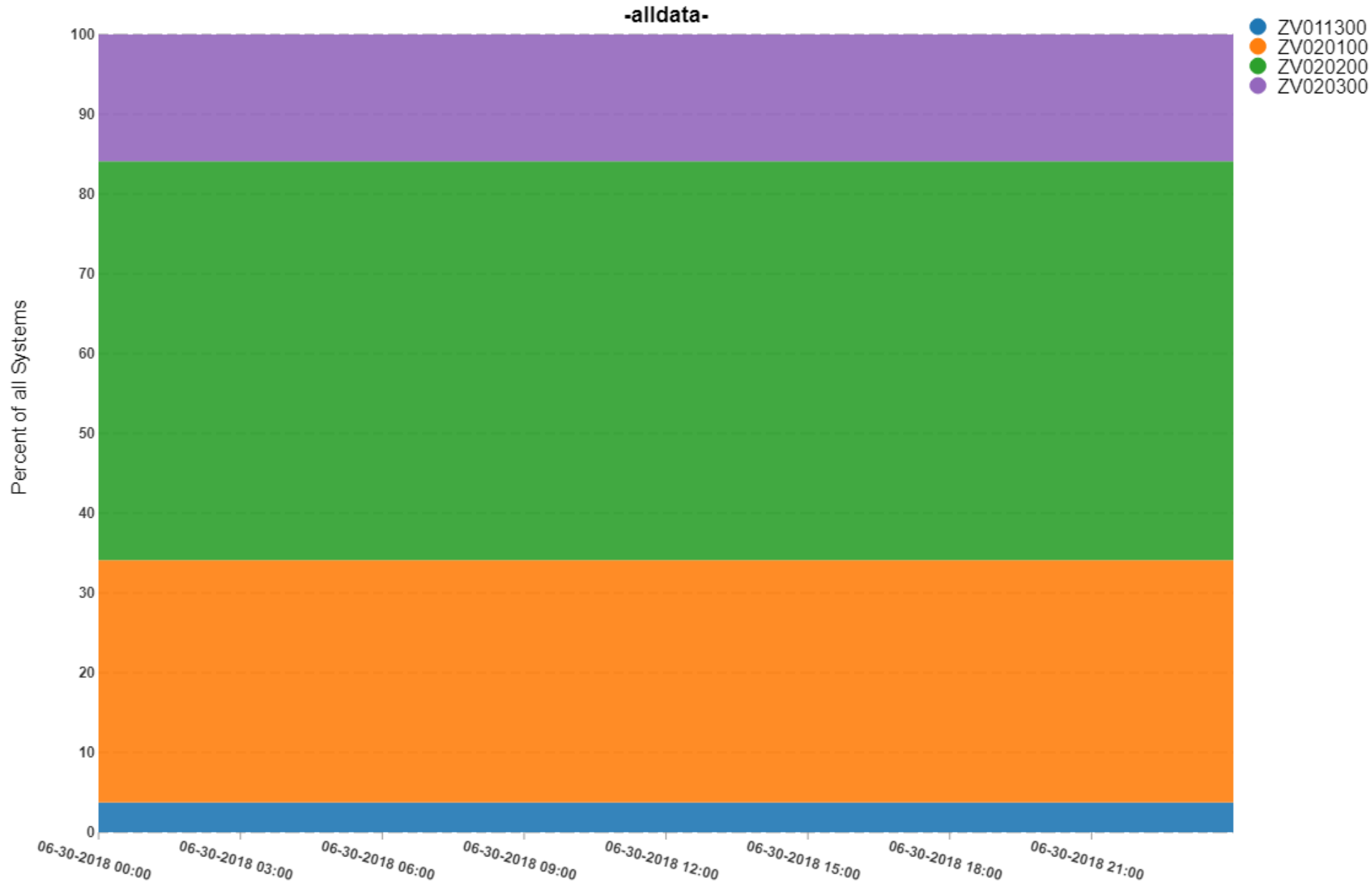
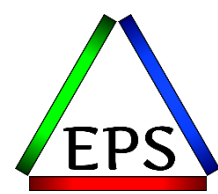
- Maintenance on older hardware can be expensive
 - But you can always go time and material, so this may be simply a risk point
- IBM software discounts between machine generations can be significant
- Advances in the hardware may make applications/system more efficient, possibly lowering consumed MSUs
- z/OS more aggressively requiring more recent hardware
 - z/OS 2.3 requires at least z12 hardware
- z/OS 1.13 out of support since Sept 2016, 2.1 went out of support Sept 2018
 - Extended support may be available at a cost

Machine Generations

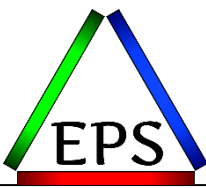


Pretty good adoption of z14!

z/OS Versions



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



- Start planning for your next z/OS migration if you're not on at least z/OS 2.2
- Start making plans for getting to at least z12 (preferably z13) levels of hardware if you're not already
 - Most customers already there (but we did see data from at least one z9 in 2017)
- 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
 - OTOH, some of those old machines are available pretty darn cheap

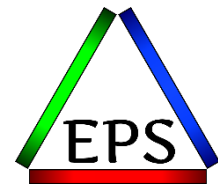
SMF Data & RMF/CMF Intervals: Why you care



- 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 I would generally support

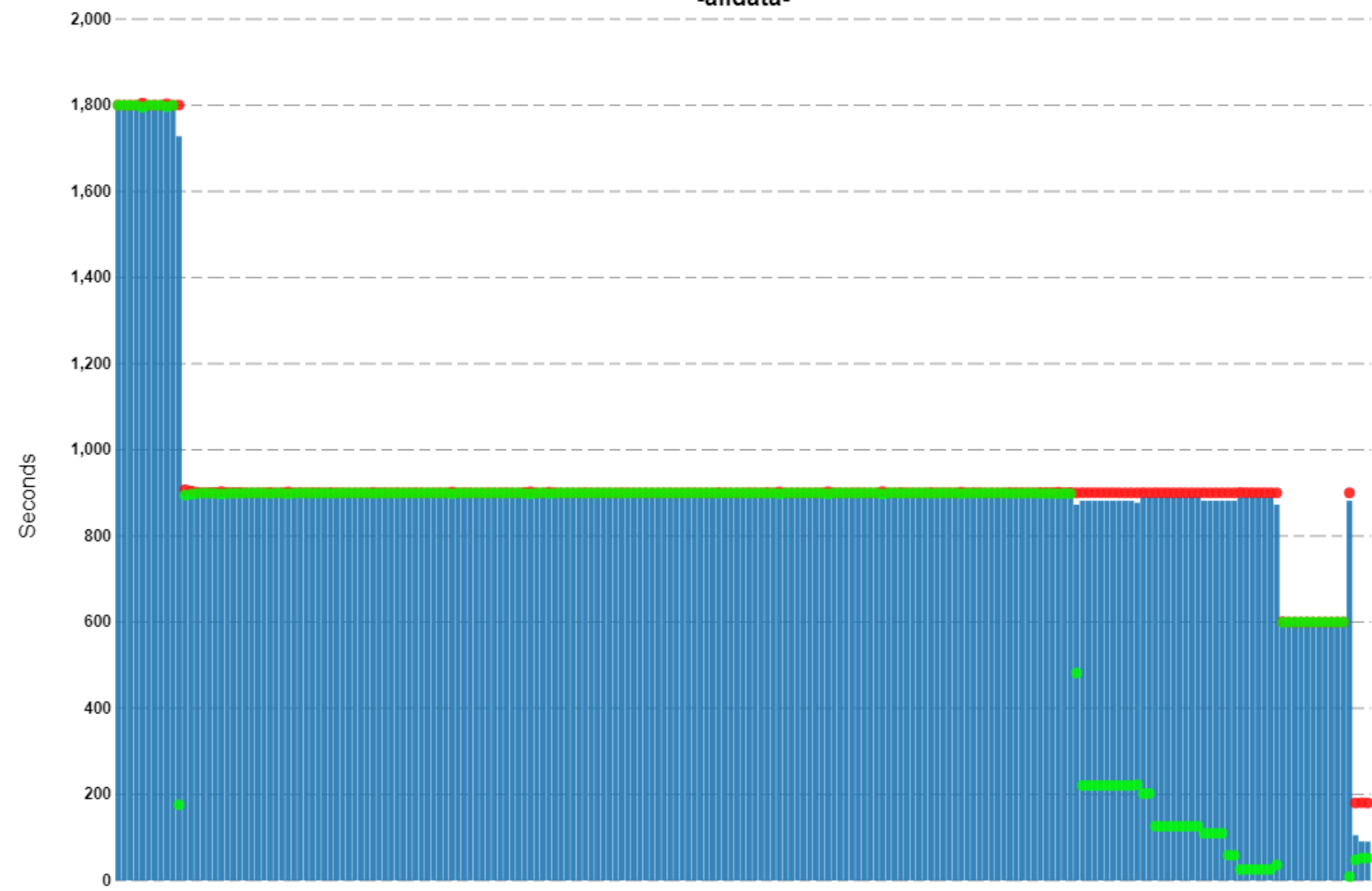
RMF/CMF Interval

Average/Max/Min For Day

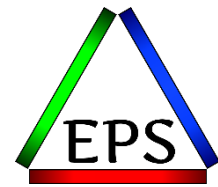


-alldata-

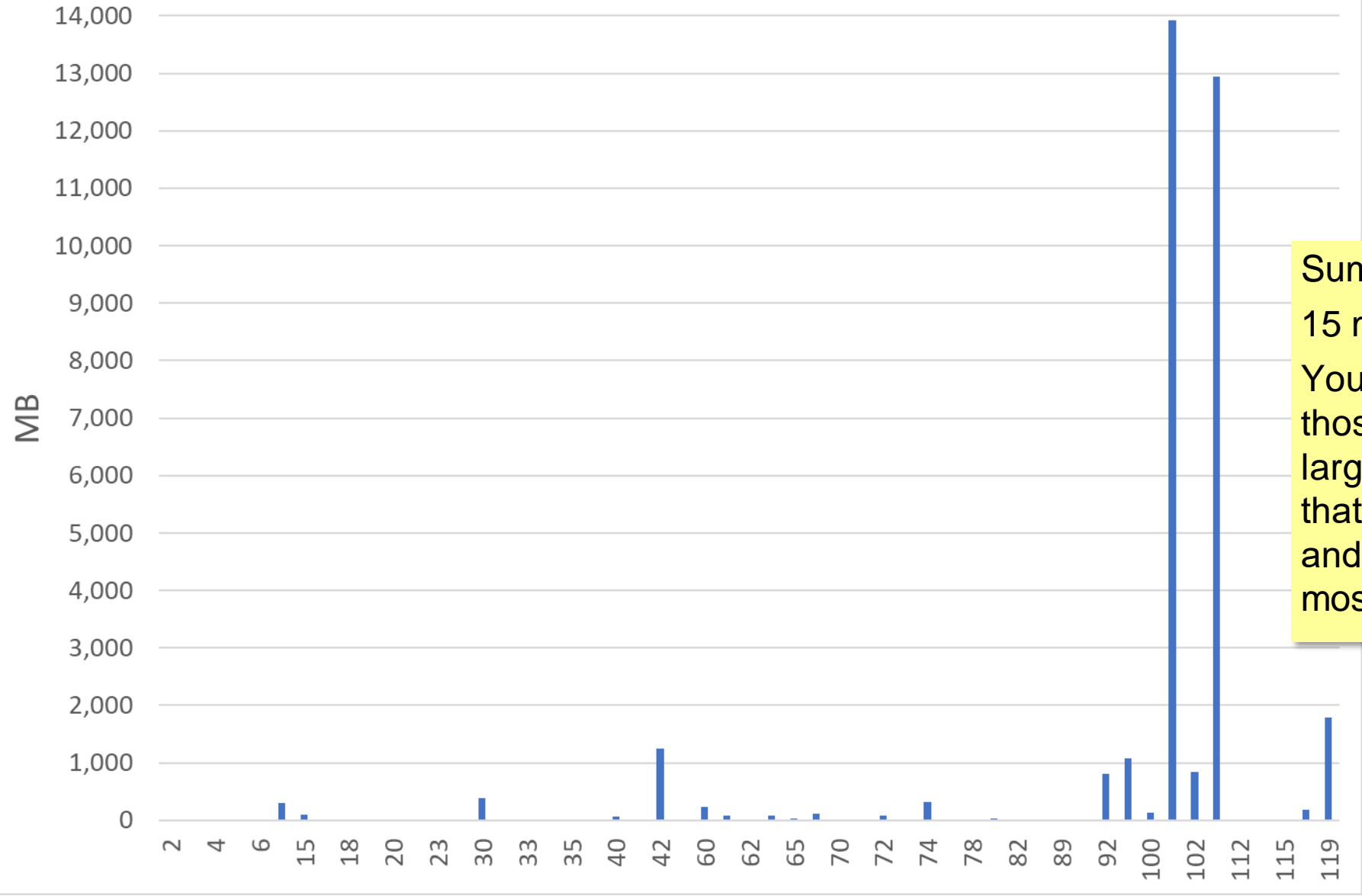
- Avg RMF Interval
- Max RMF Interval
- Min RMF Interval



Almost everyone is using 15 minute intervals

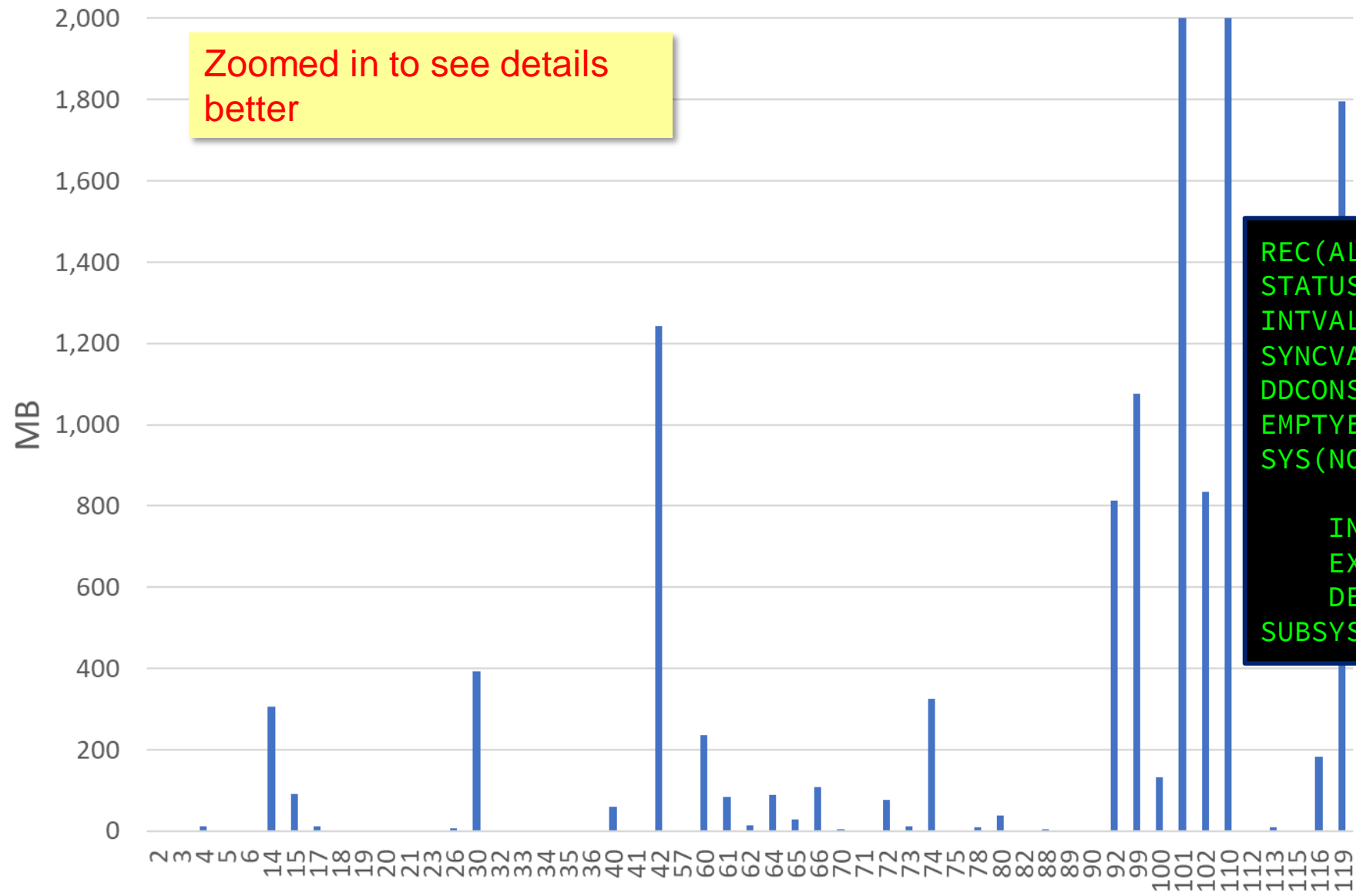


One Particular 4-System Sysplex By Record Type



Sum of 4 systems
15 minute intervals
You might be thinking
those 99s are pretty
large at ~1GB/day, but
that's from 4 systems,
and they're recording
most subtypes.

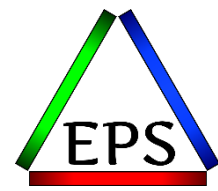
One Particular 4-System Sysplex By Record Type



Zoomed in to see details better

```

REC(ALL)
STATUS(003000)
INTVAL(15)
SYNCVAL(15)
DDCONS(NO)
EMPTYEXCPSEC(SUPPRESS)
SYS(NOTYPE(92(10,11),
          99(1,2,3)),
     INTERVAL(SMF,SYNC),
     EXITS(...),
     DETAIL)
SUBSYS(STC,NODETAIL)
  
```

99s Breakdown by subtype

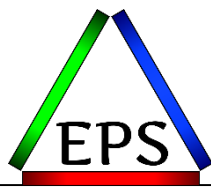
SubType	%Recs	%Bytes	Recs	Bytes
subtype 099_004	16.77	3.98	240,976	75,927,072
subtype 099_006	2.40	9.43	34,487	179,773,376
subtype 099_008	2.40	1.80	34,487	34,317,668
subtype 099_011	0.08	0.19	1,150	3,543,150
subtype 099_012	11.88	8.24	170,761	157,161,280
subtype 099_013	47.54	32.72	683,031	624,146,176
subtype 099_014	0.12	0.06	1,698	1,103,291

The recommended 99 subtypes were about 340MB for 4 systems. (85MB/system)

Subtype 10 are typically non-existent (processor speeds don't usually change).

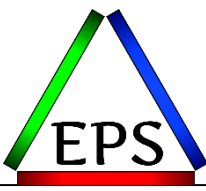
Subtype 11 gets written every 5 minutes only when there are group capacity limits in place.

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



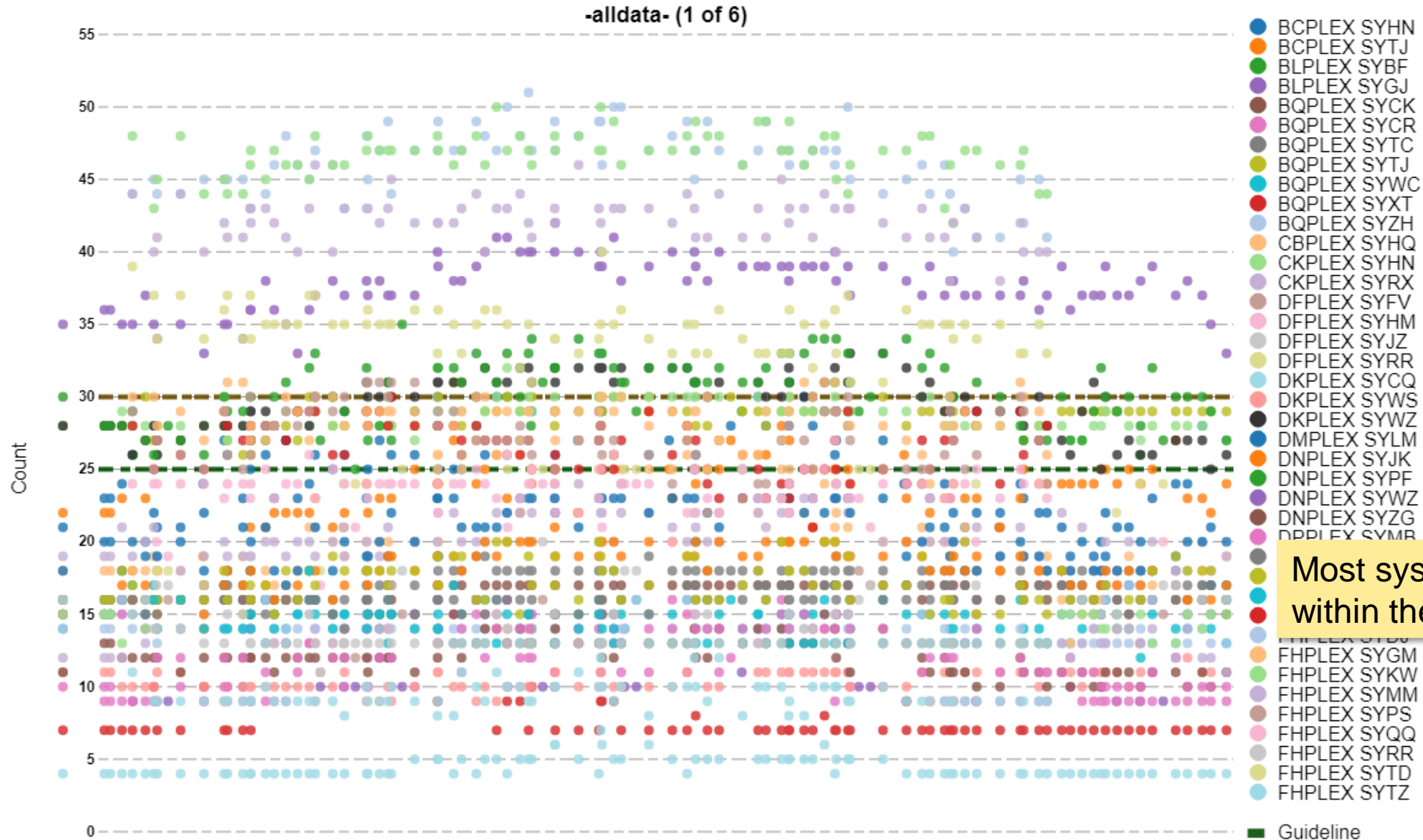
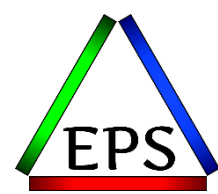
- If your RMF interval is > 900 seconds, change to 900 seconds
 - Possibly consider shorter intervals
- If you're RMF and SMF intervals are not synced, make them so
 - Makes it easier (possible) to compare the non-RMF records to the RMF records
 - Unless you're using very short RMF intervals
- Record at least some 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

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

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 guideline

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

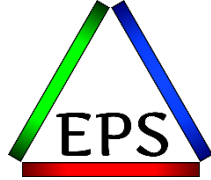
CEC Busy: Why you care



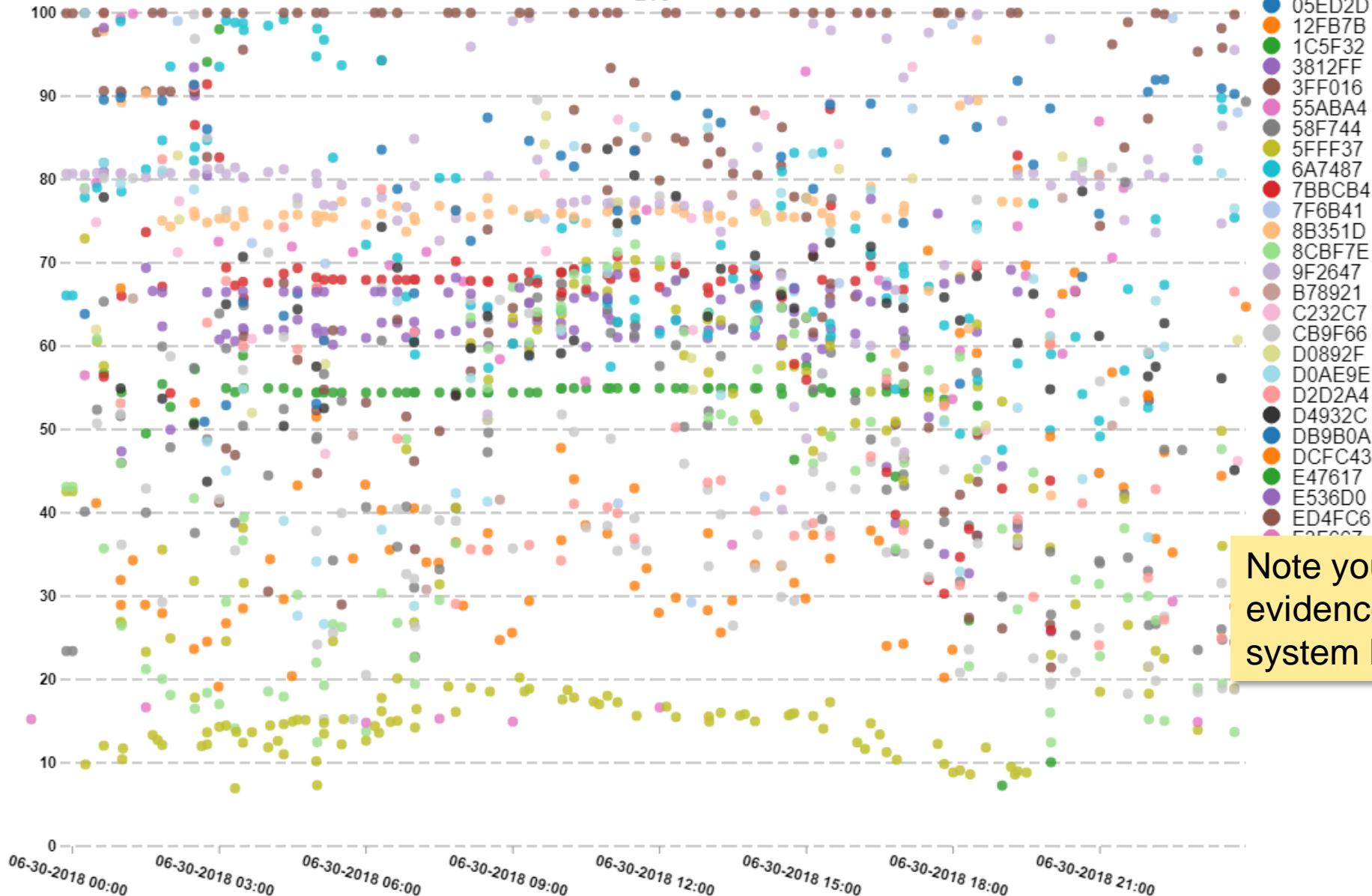
- Can you run your z/OS machine at 100% busy
 - (yes)
- Should you run your z/OS machine at 100% busy
 - (probably not)
- What about zIIPs? Should you keep those below xx%?
 - (see above)

- So what are sites doing?

CEC Busy General Purpose Engines



z13

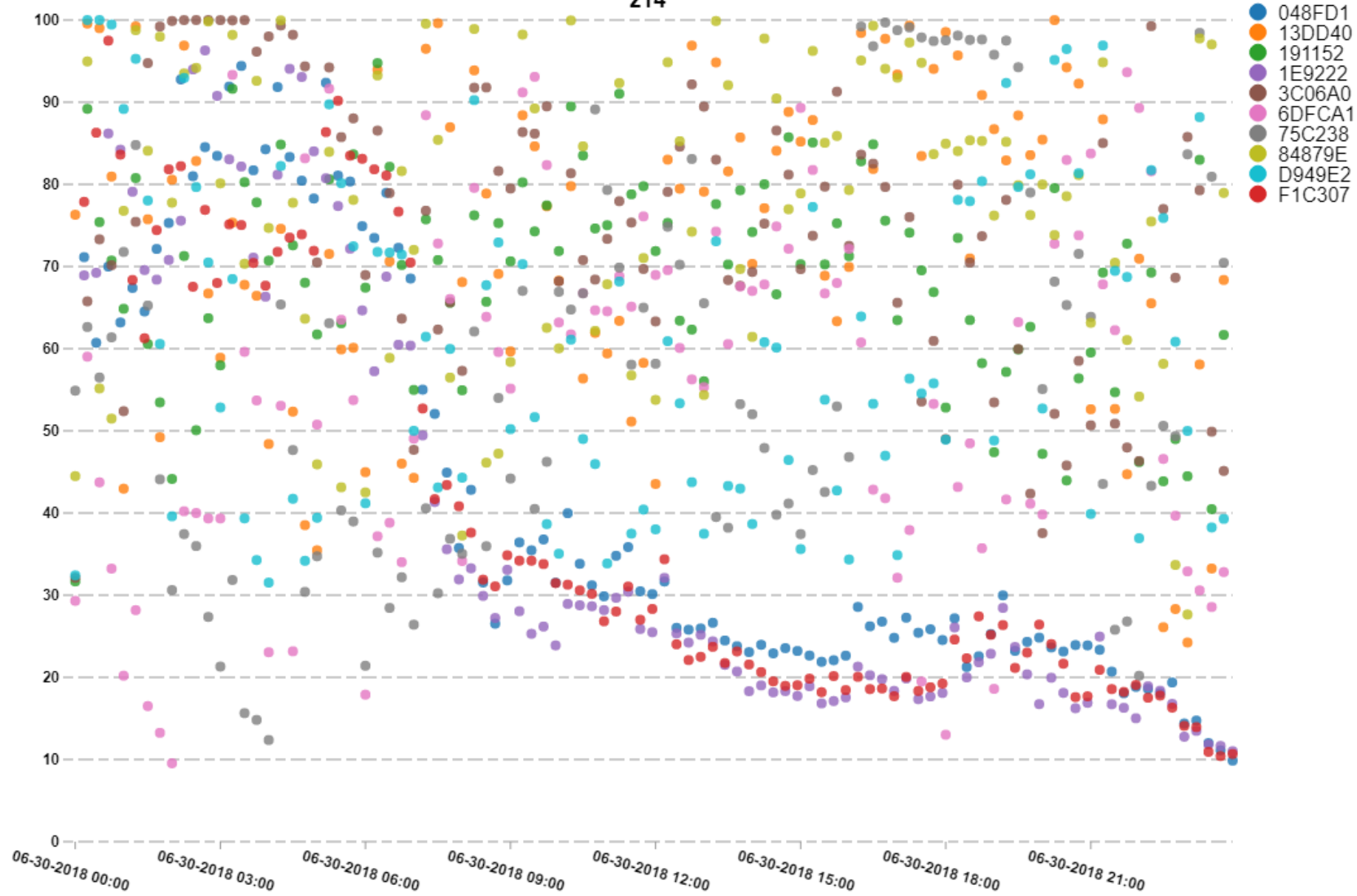
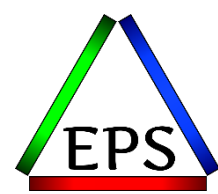


Note you can see some evidence of at least one system likely being capped

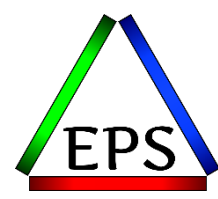
CEC Busy

General Purpose Engines

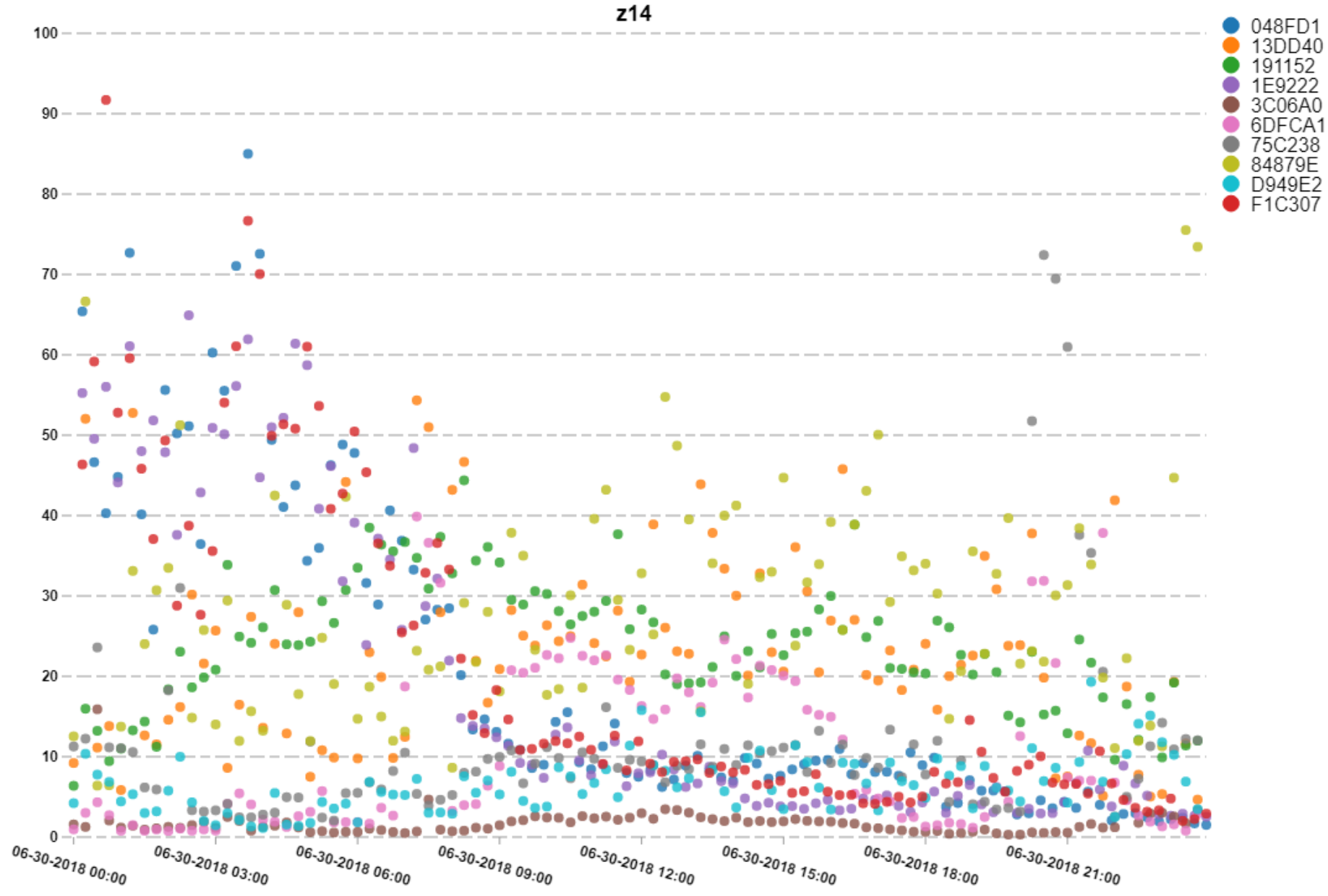
z14



CEC Busy - zIIPs



CEC Busy - zIIPs



CEC Busy: What you should do

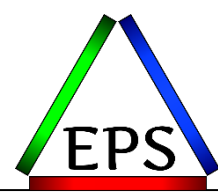


- Don't be afraid to run less than 100% busy
 - In general, there are efficiency and performance benefits from running less than 100% busy
 - When you're looking at 15 minute intervals, don't forget those are averages over the 15 minutes: there will be time periods within that 15 minute interval where 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
- Today focus should be on consuming least amount of MSUs while getting the work done
 - Usually this needs to be a peak R4HA analysis
 - 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!



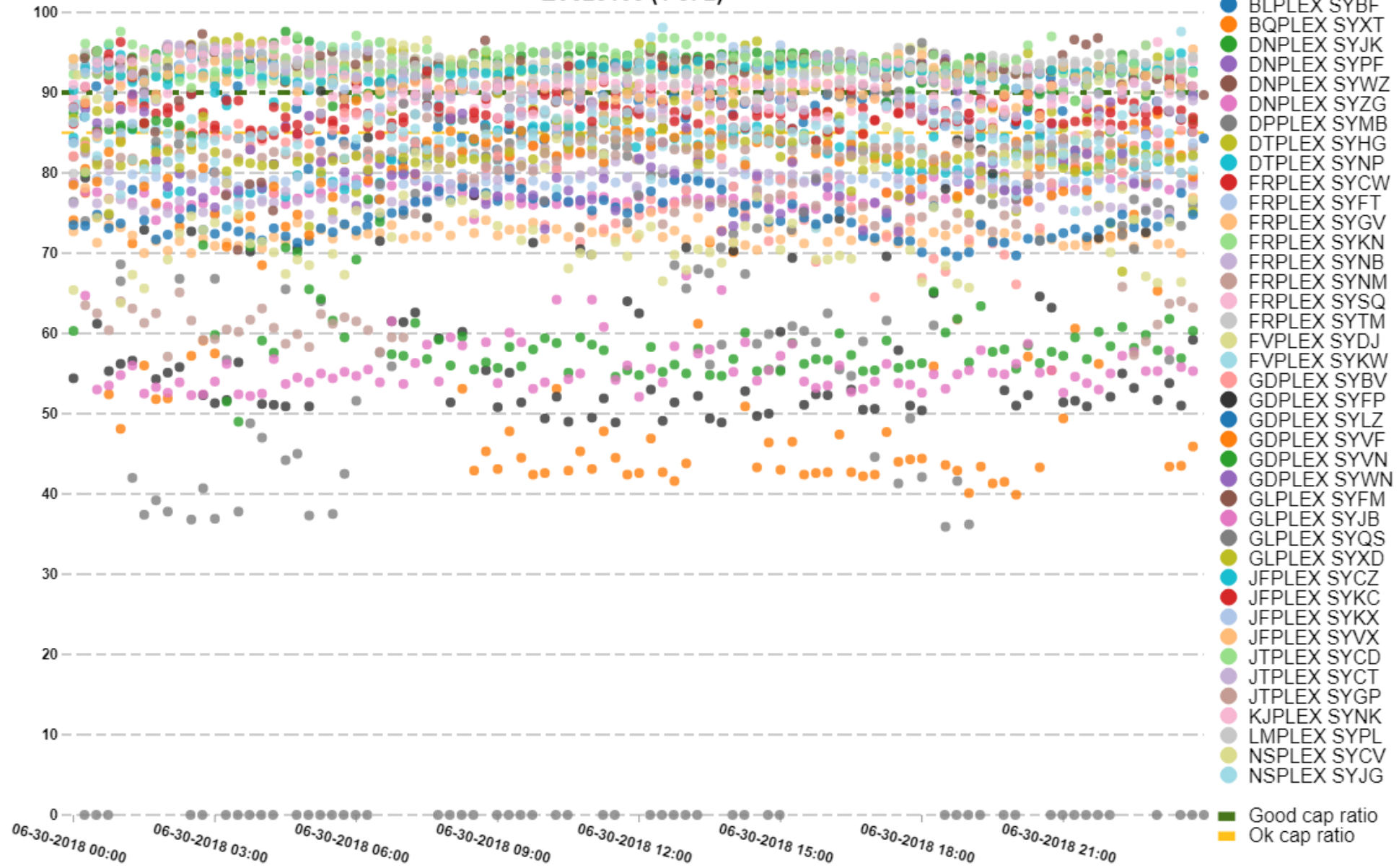
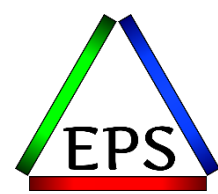
Capture Ratio: Why you care

- Capture Ratio refers to how much of the CPU time consumed by the LPAR is “captured”, i.e. the operating system has attributed it to a particular workload
 - “Captured” time is the CPU time recorded in the SMF 72 or 30 records
 - LPAR CPU time is recorded in SMF 70 records
- Uncaptured CPU time can include:
 - Interrupt processing
 - Paging
 - Certain traces
 - Certain types of contention
 - Affinity processing
 - Work queue processing
 - Allocation (fragmented storage pools, inefficient ACS routines)
- We expect to see capture ratios above 90%, at least at relatively busy times
 - There will always be some uncaptured time

Capture Ratio

All Intervals

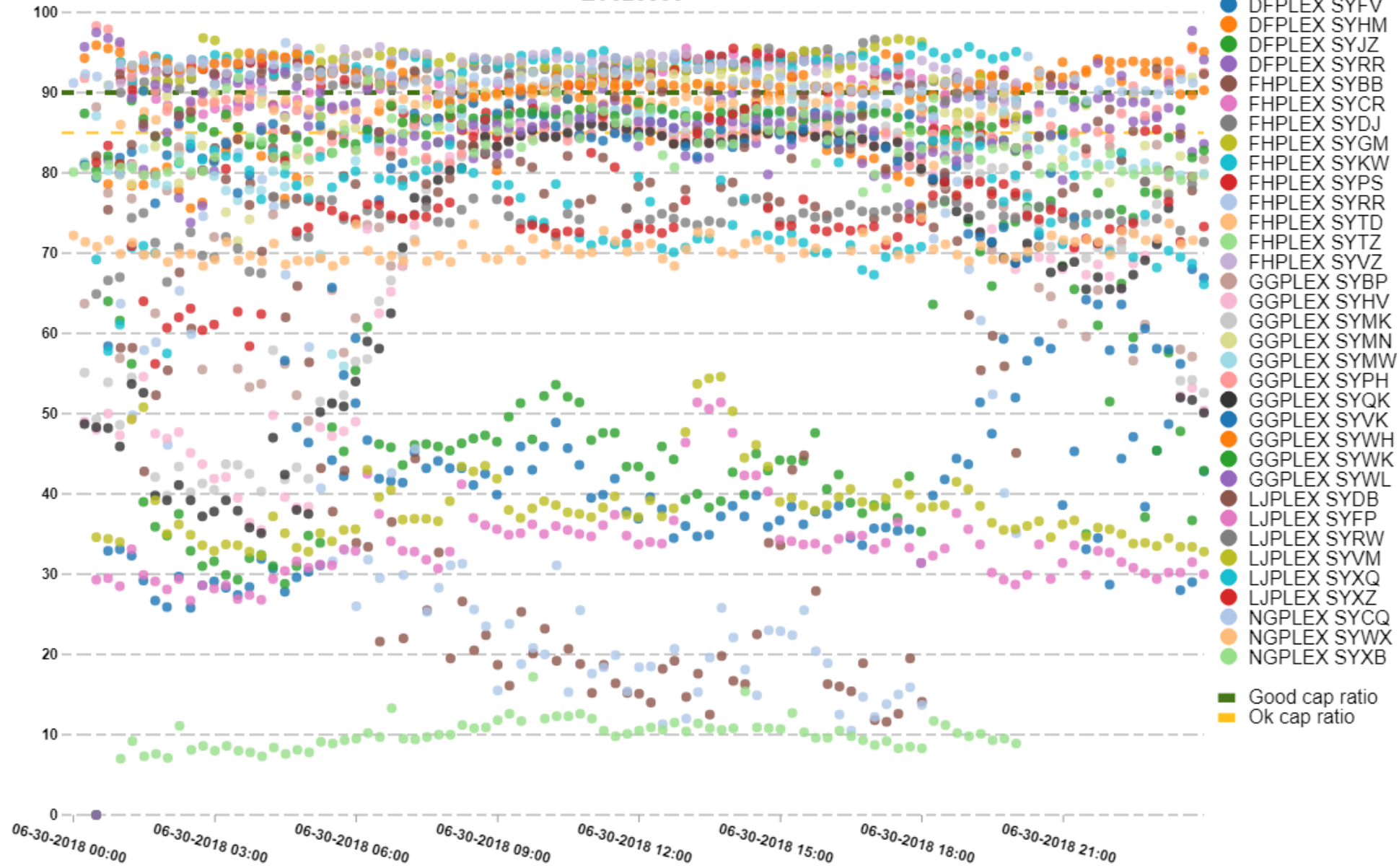
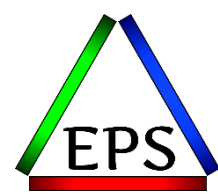
ZV020100 (1 of 2)



Capture Ratio

All Intervals

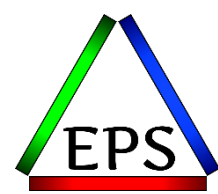
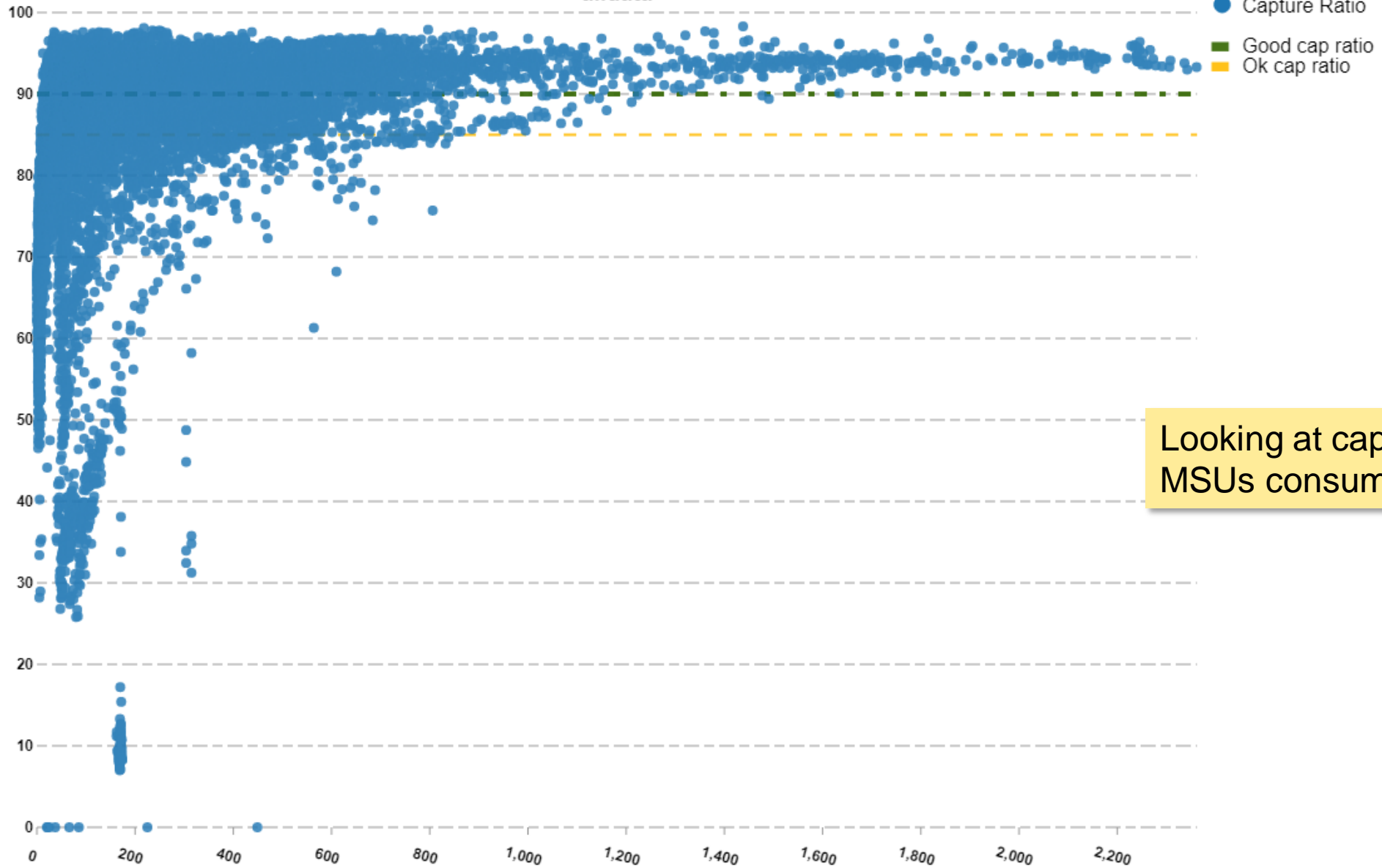
ZV020300



Capture Ratio vs. MSUs

All intervals

-alldata-

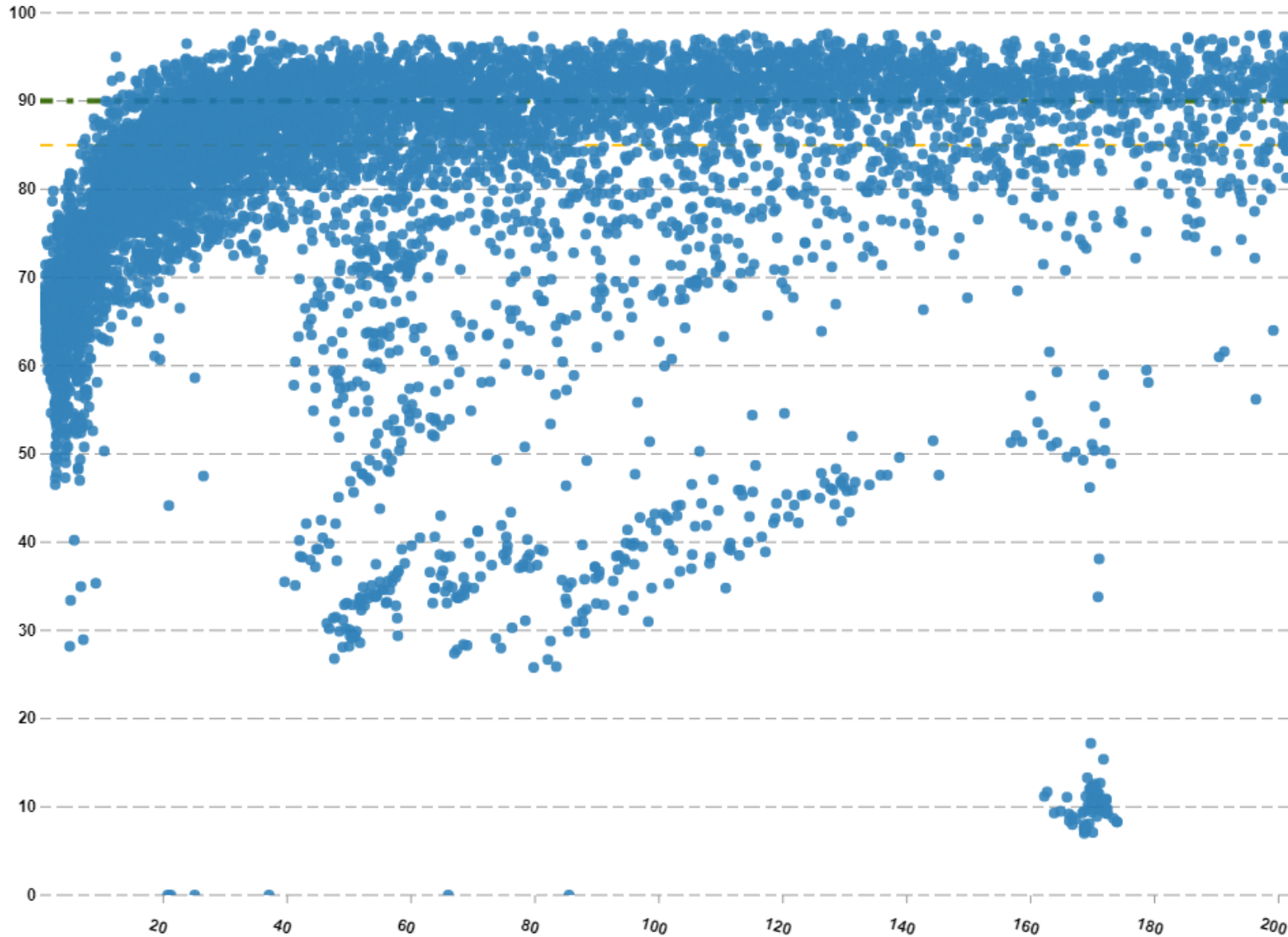


Looking at capture ratio vs. MSUs consumed in the interval

Capture Ratio vs. MSUs

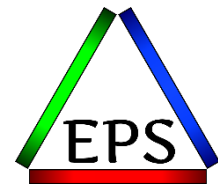
All intervals

-alldata-

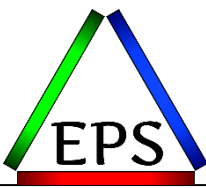


- Capture Ratio
- Good cap ratio
- Ok cap ratio

Zoomed in to see the smaller systems better



Capture Ratio : What you should do



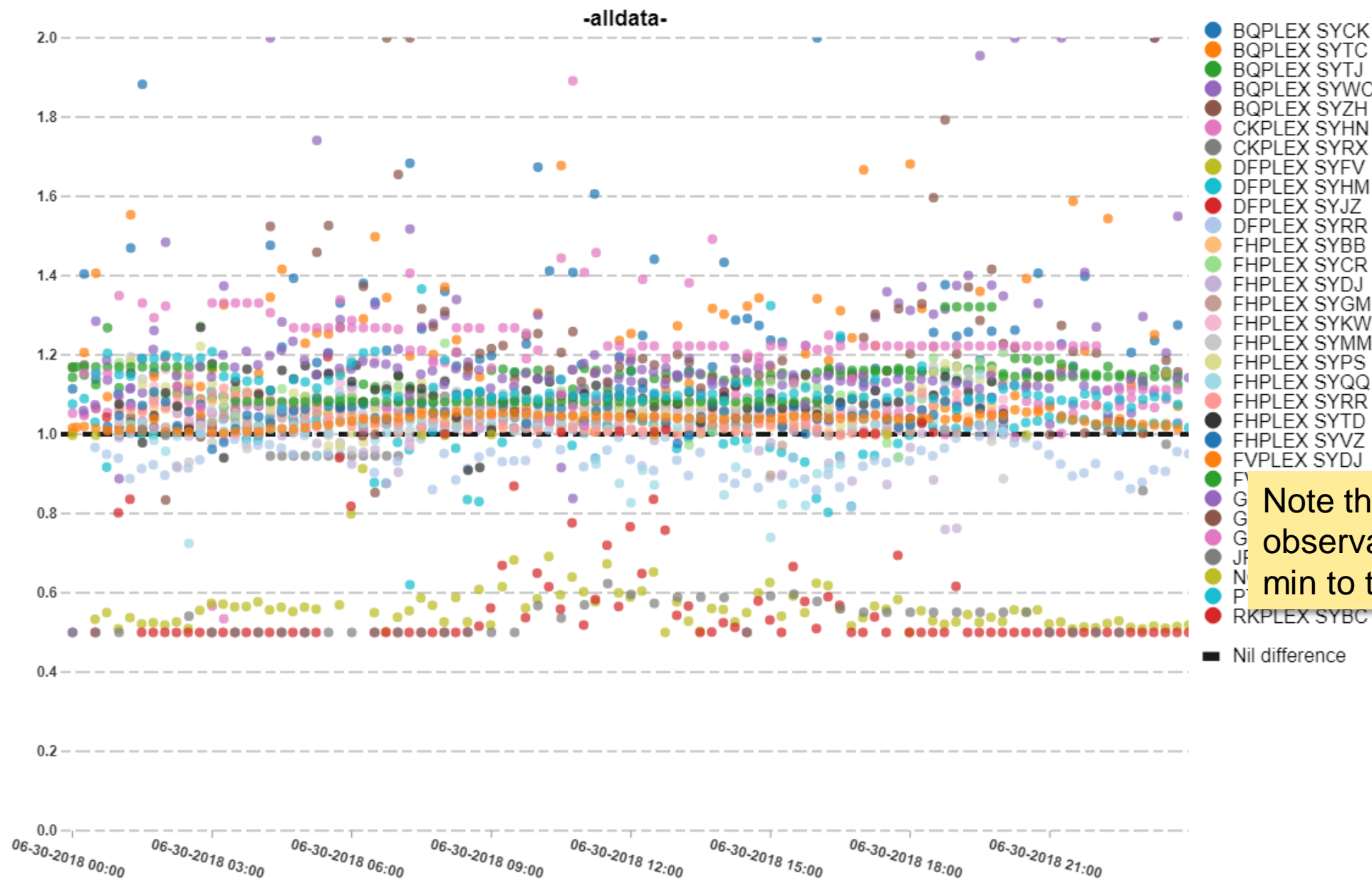
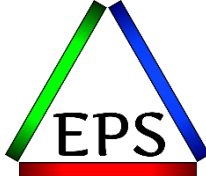
- If capture ratio is above 90%: no concern, probably little tuning opportunity
- Some systems clearly run relatively low capture ratios all the time
 - These seem to be mostly lightly loaded systems
- If a capture ratio is regularly below 85% and the system is a significant contributor to your software costs, I would definitely consider looking into it
 - Look for old traces or SLIP traps that may have been left enabled
 - Do you have RMPTTOM set too low in IEAOPTxx?
 - May be able to increase it above default for lightly used systems
 - Look for paging activity
 - Look for contention
 - Look for highly fragmented storage pools or overly complicated ACS routines
 - Unfortunately, tracking down uncaptured time is usually difficult
 - Because it is “uncaptured”

SMT: Why do you care



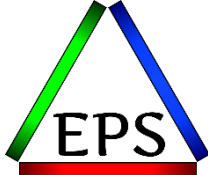
- SMT may be something useful to investigate in particular situations
 - See my SMT presentation tomorrow for more details!
- These SMT measurements are notoriously variable, so treat these charts with even more caution than the others herein
 - Your results will vary
- Capacity Factor is generally held 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

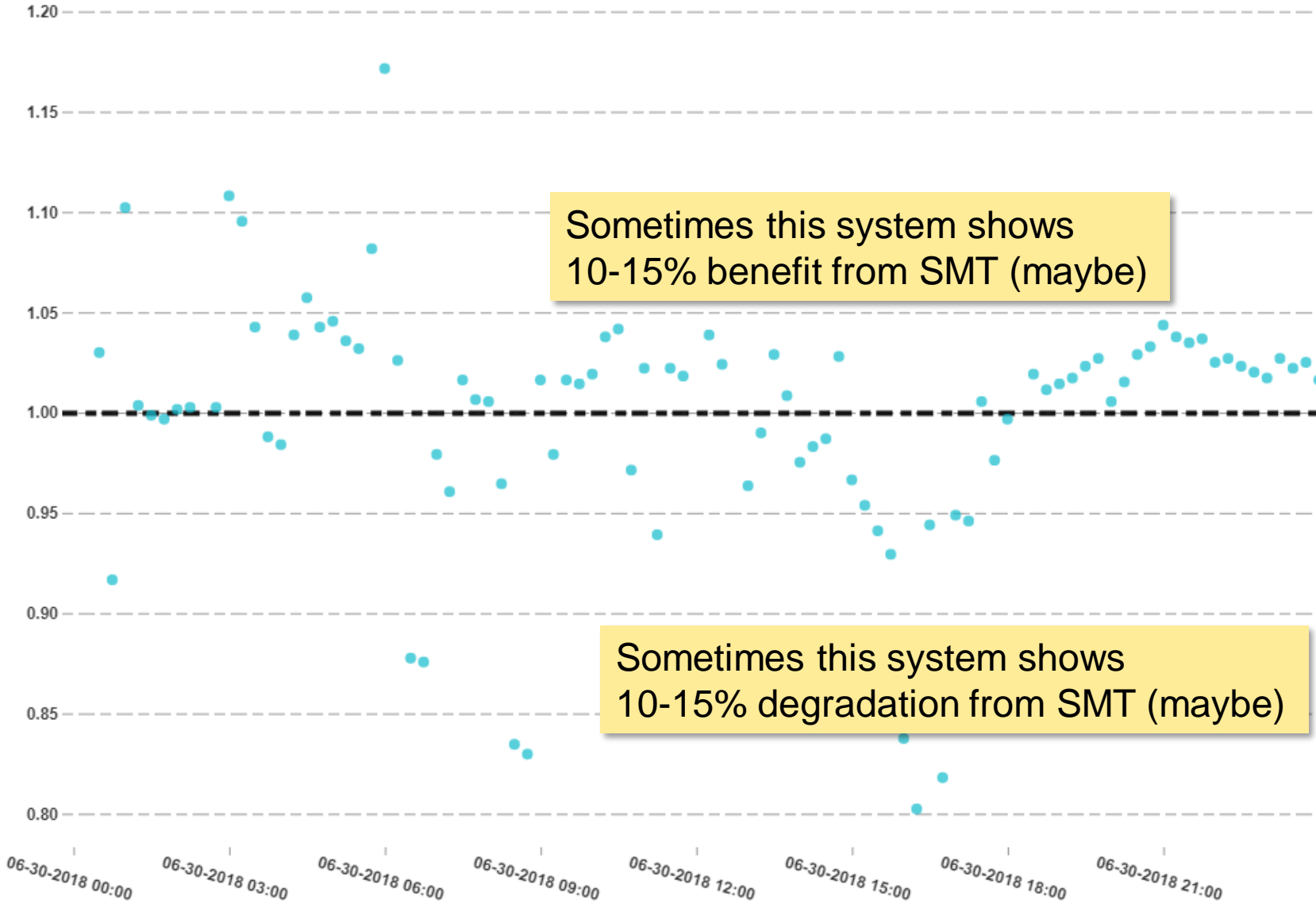


Note that there are observations from the min to the max

SMT: Capacity Factor



-alldata-



- BQPLEX SYCK
- BQPLEX SYTC
- BQPLEX SYTJ
- BQPLEX SYWC
- BQPLEX SYZH
- CKPLEX SYHN
- CKPLEX SYRX
- DFPLEX SYFV
- DFPLEX SYHM
- DFPLEX SYJZ
- DFPLEX SYRR
- FHPLEX SYBB
- FHPLEX SYCR
- FHPLEX SYDJ
- FHPLEX SYGM
- FHPLEX SYKW
- FHPLEX SYMM
- FHPLEX SYPS
- FHPLEX SYQQ
- FHPLEX SYRR
- FHPLEX SYTD
- FHPLEX SYVZ
- FVPLEX SYDJ
- FVPLEX SYKW
- GLPLEX SYFM
- GLPLEX SYQS
- GLPLEX SYXD
- JPPLEX SYJD
- NGPLEX SYGK
- PTPLEX SYDV
- RKPLEX SYBC

■ Nil difference

SMT: What you should do

- If you have a good reason, try it
 - Along with SMT measurements, be sure to check your application responsiveness
- 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

Hardware Instrumentation Services: Why you care

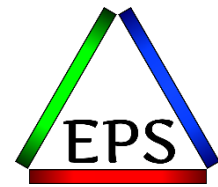


- Collecting the SMF 113 HIS records is mandatory for doing proper planning for a new processor
- The information is interesting to understand the characteristics of your workload and how efficiently it's utilizing the hardware resources
- The primary numbers of interest (lower=better in all cases):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

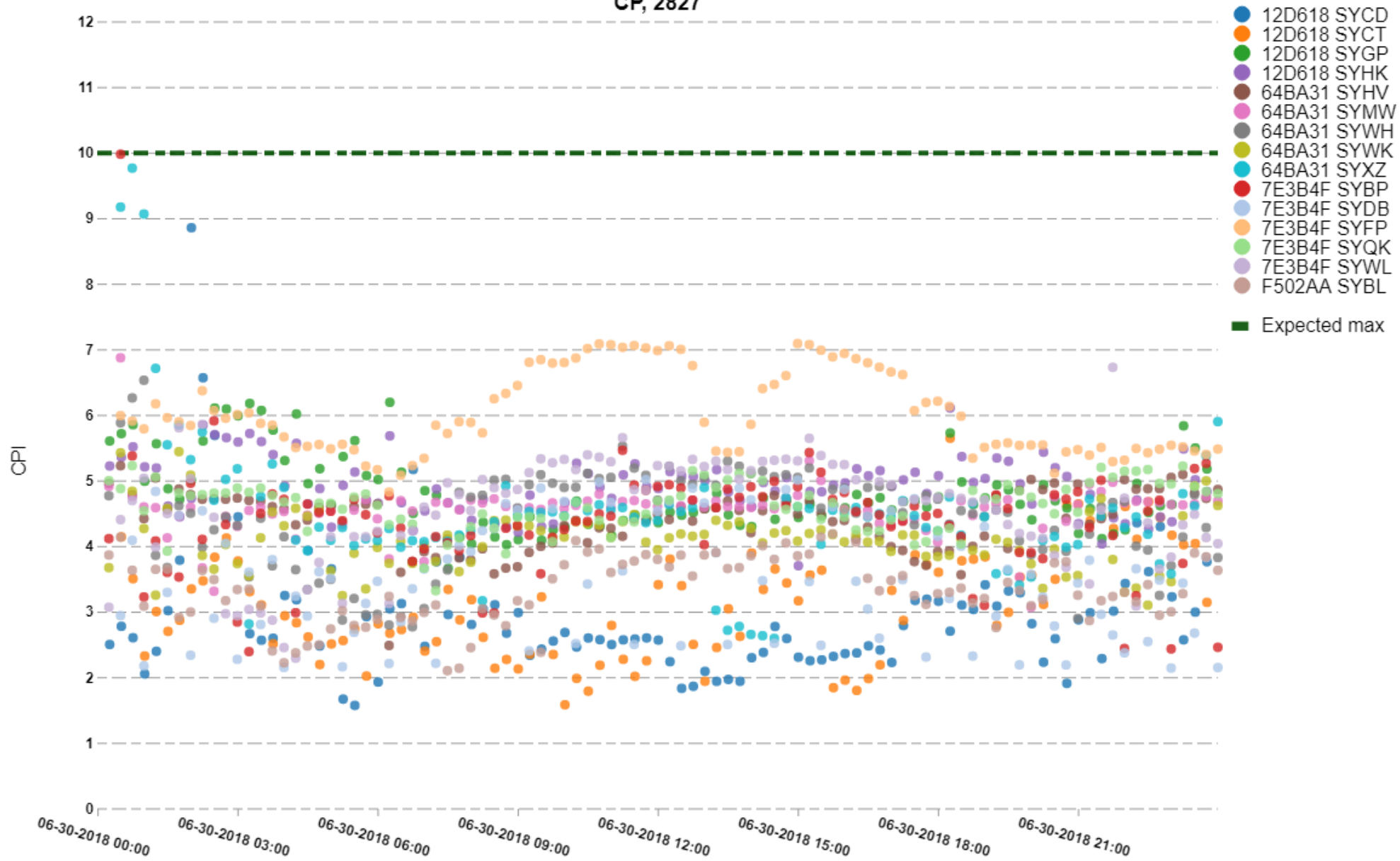
Cycles Per Instruction

By z/OS Hardware Model

CP, 2827



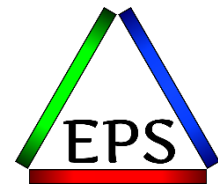
zEC12 GCP



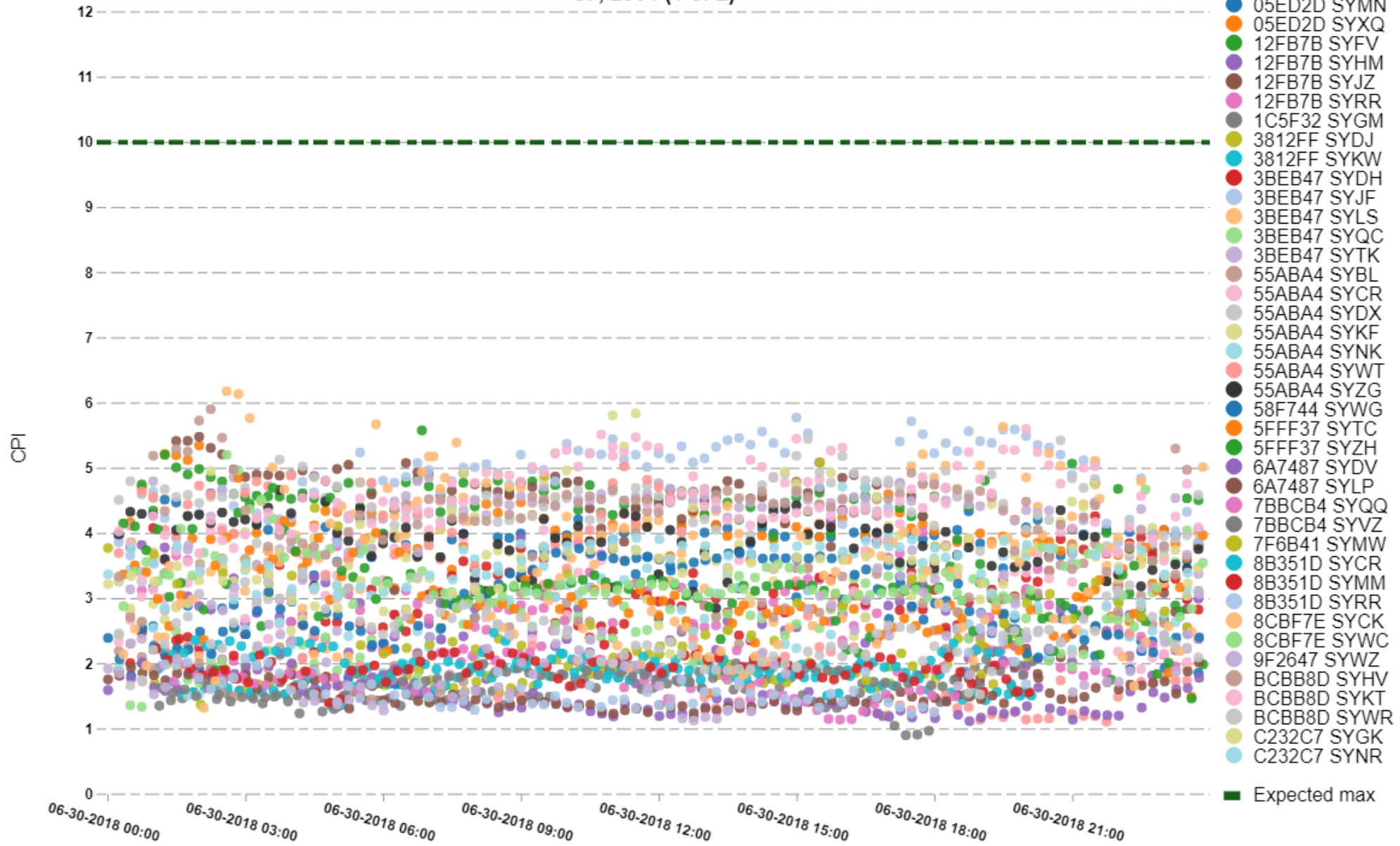
Cycles Per Instruction

By z/OS Hardware Model

CP, 2964 (1 of 2)



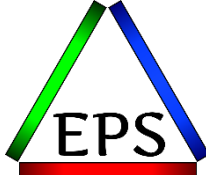
Z13 GCP



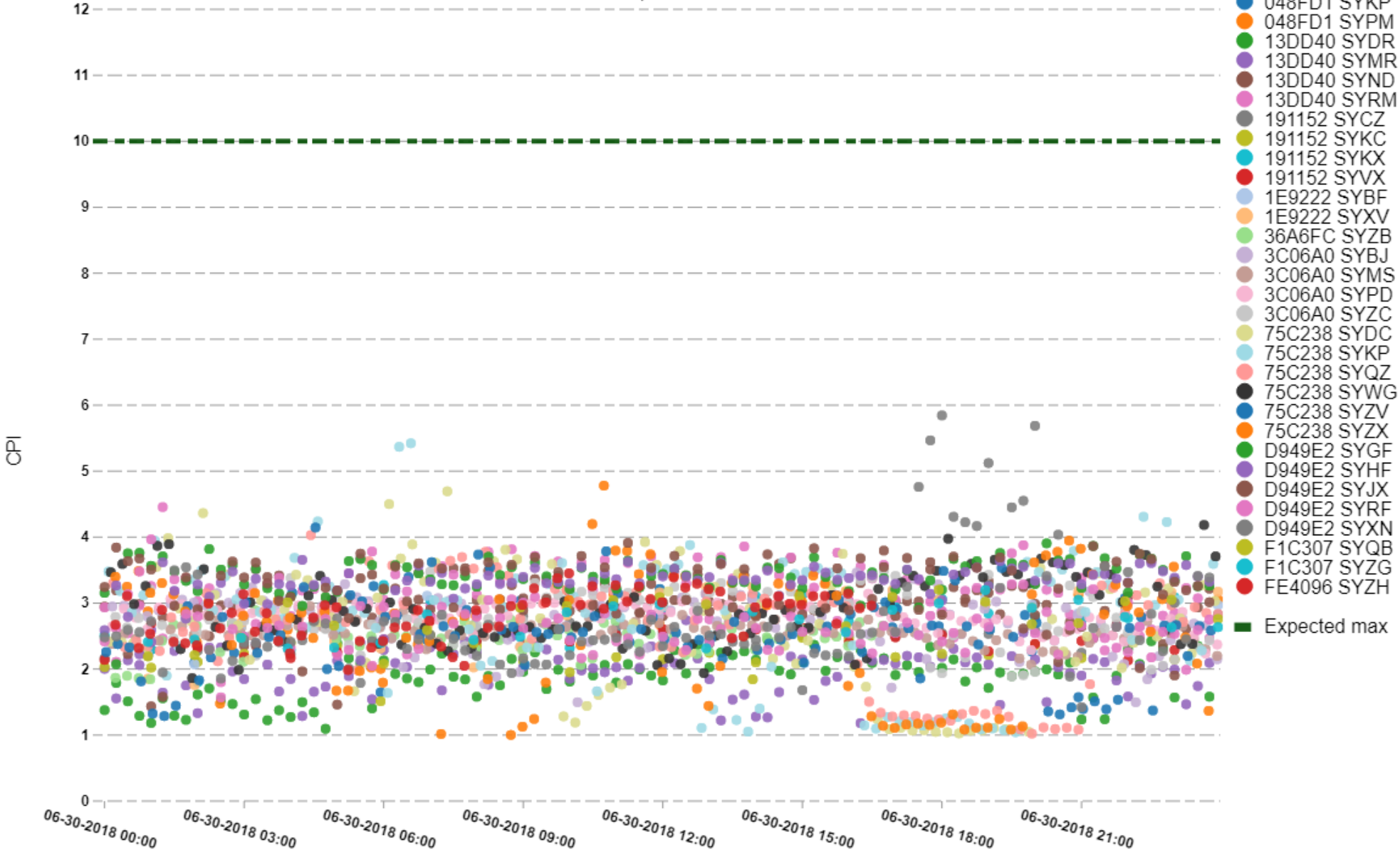
Cycles Per Instruction

By z/OS Hardware Model

CP, 3906



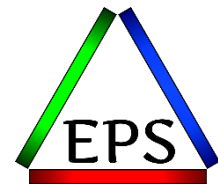
z14 GCP



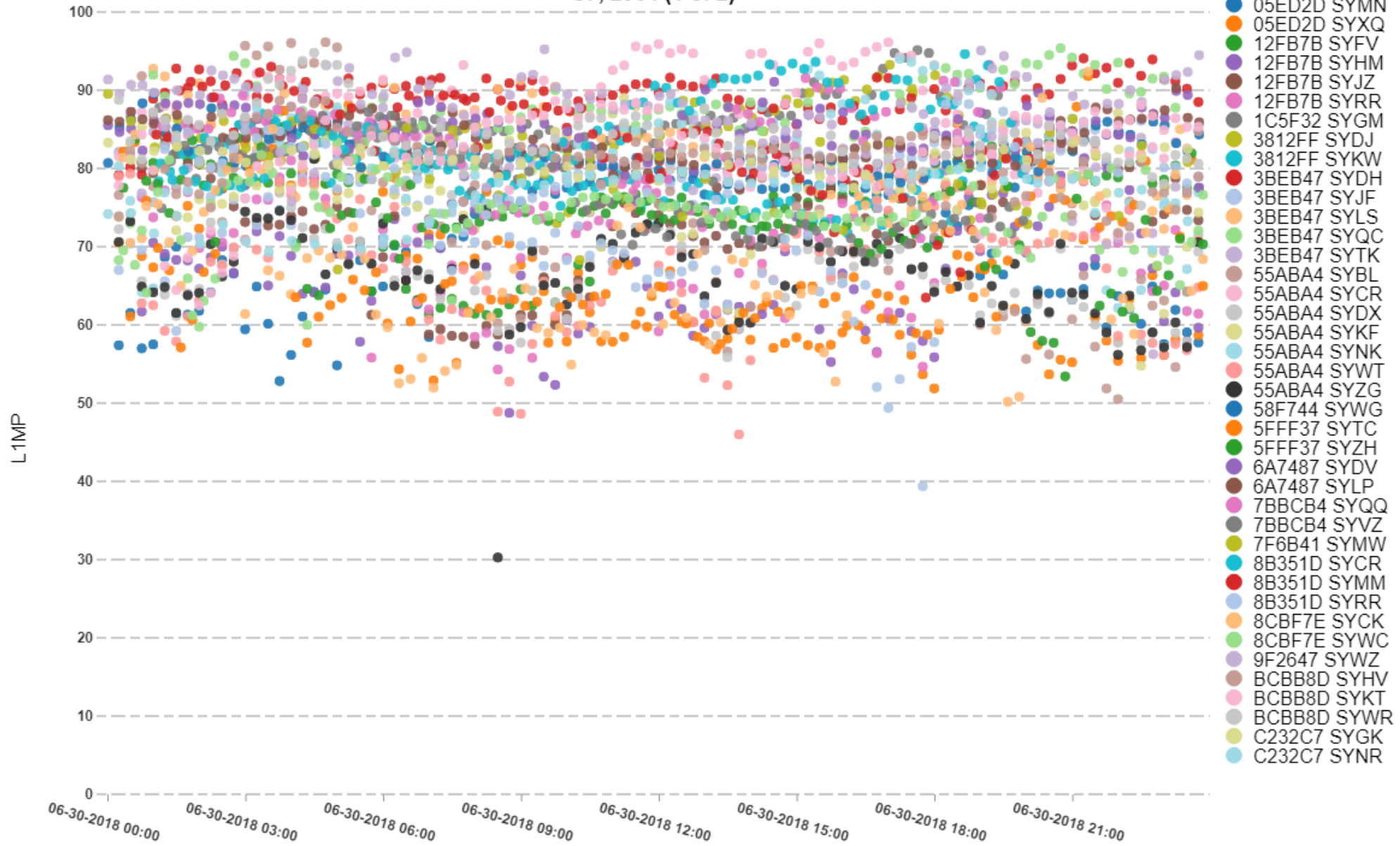
Percent of L1, L2, & L3 Misses Satisfied in L4

By z/OS Hardware Model

CP, 2964 (1 of 2)



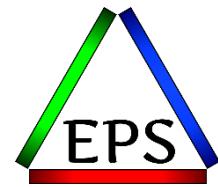
z13



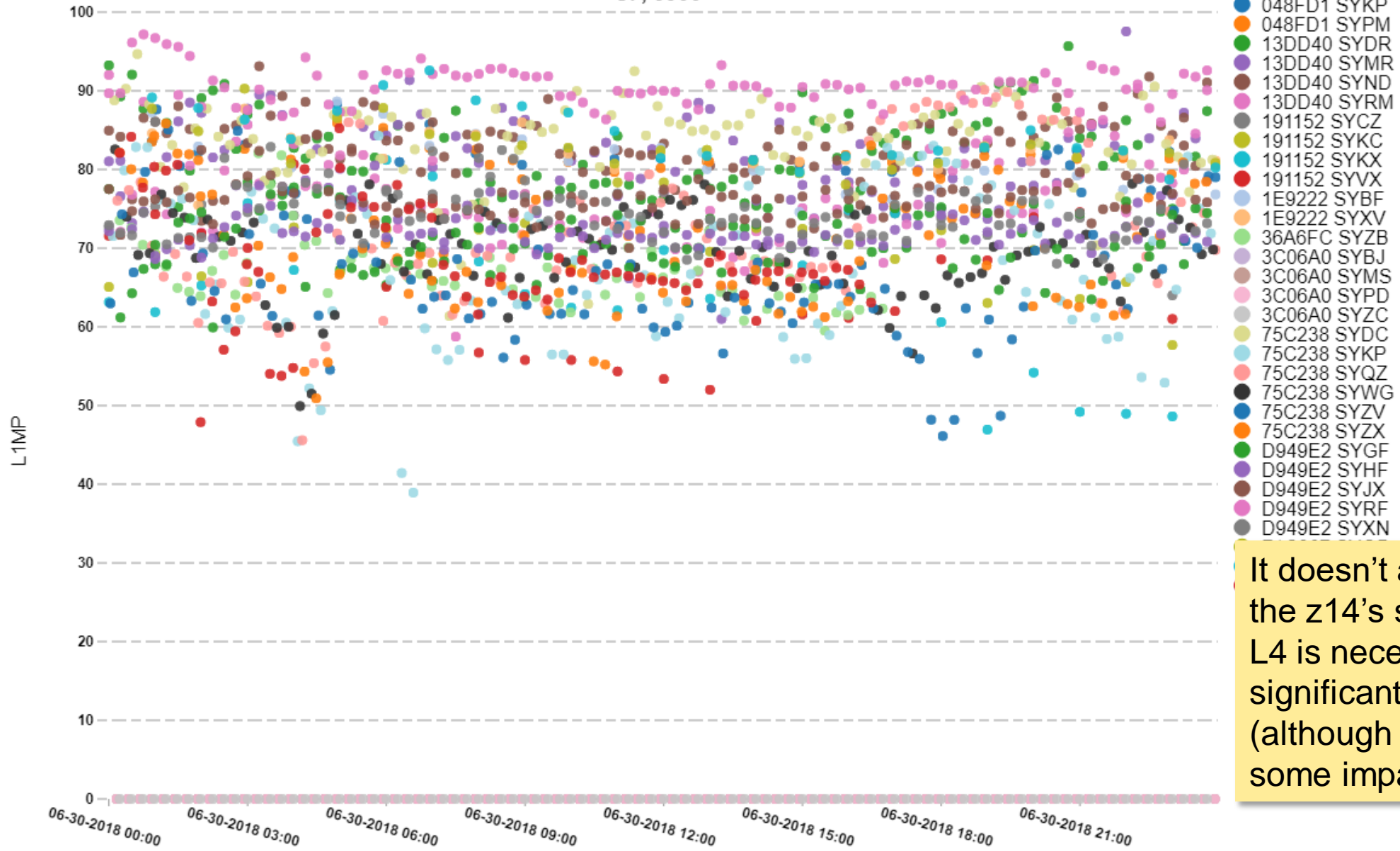
Percent of L1, L2, & L3 Misses Satisfied in L4

By z/OS Hardware Model

CP, 3906



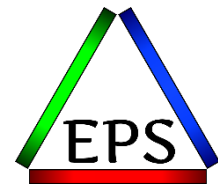
z14



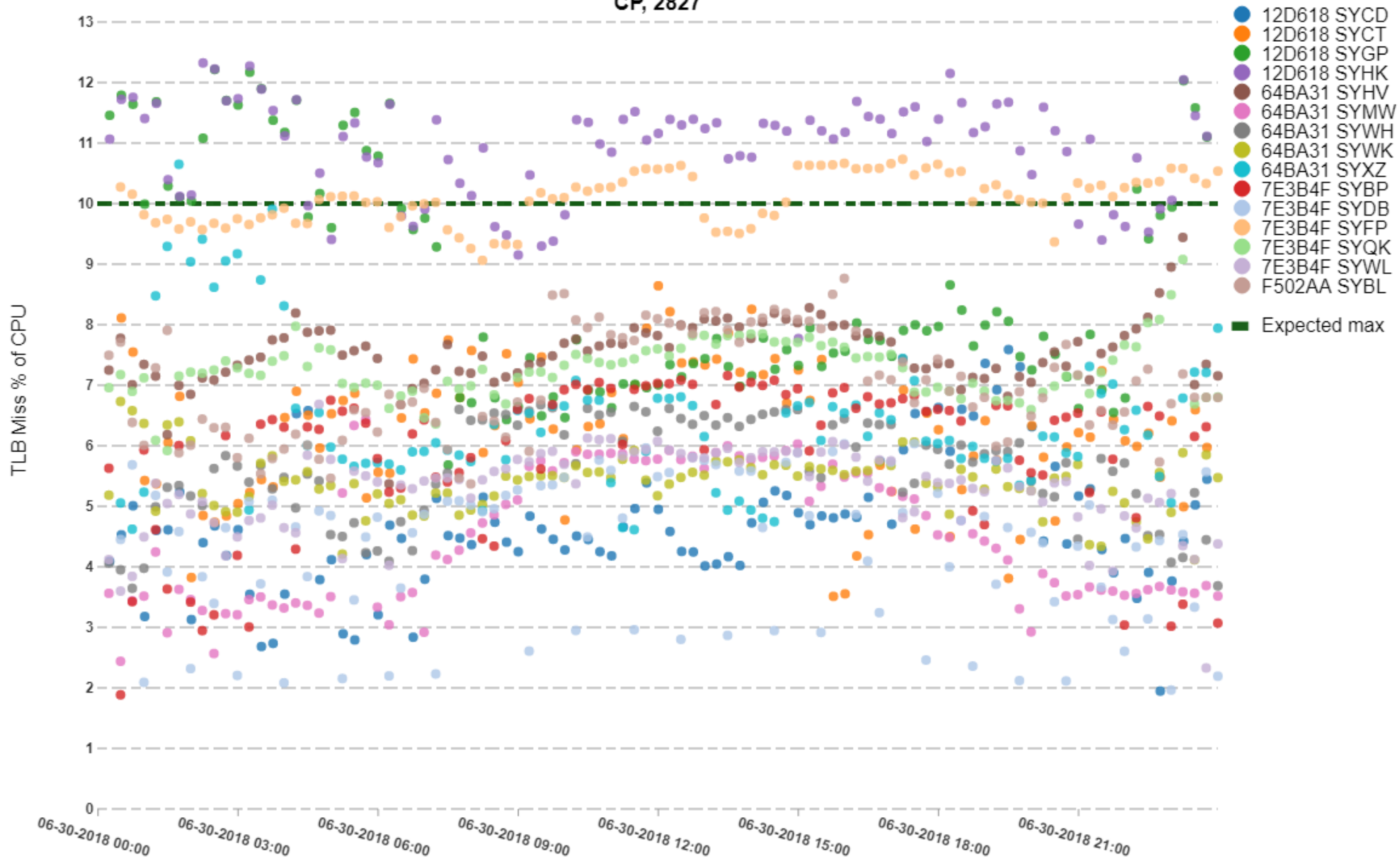
It doesn't appear that the z14's smaller total L4 is necessarily a significant problem (although there is some impact)

TLB Miss CPU % By z/OS Hardware Model

CP, 2827

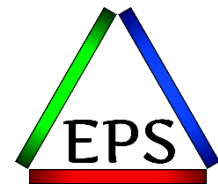


zEC12

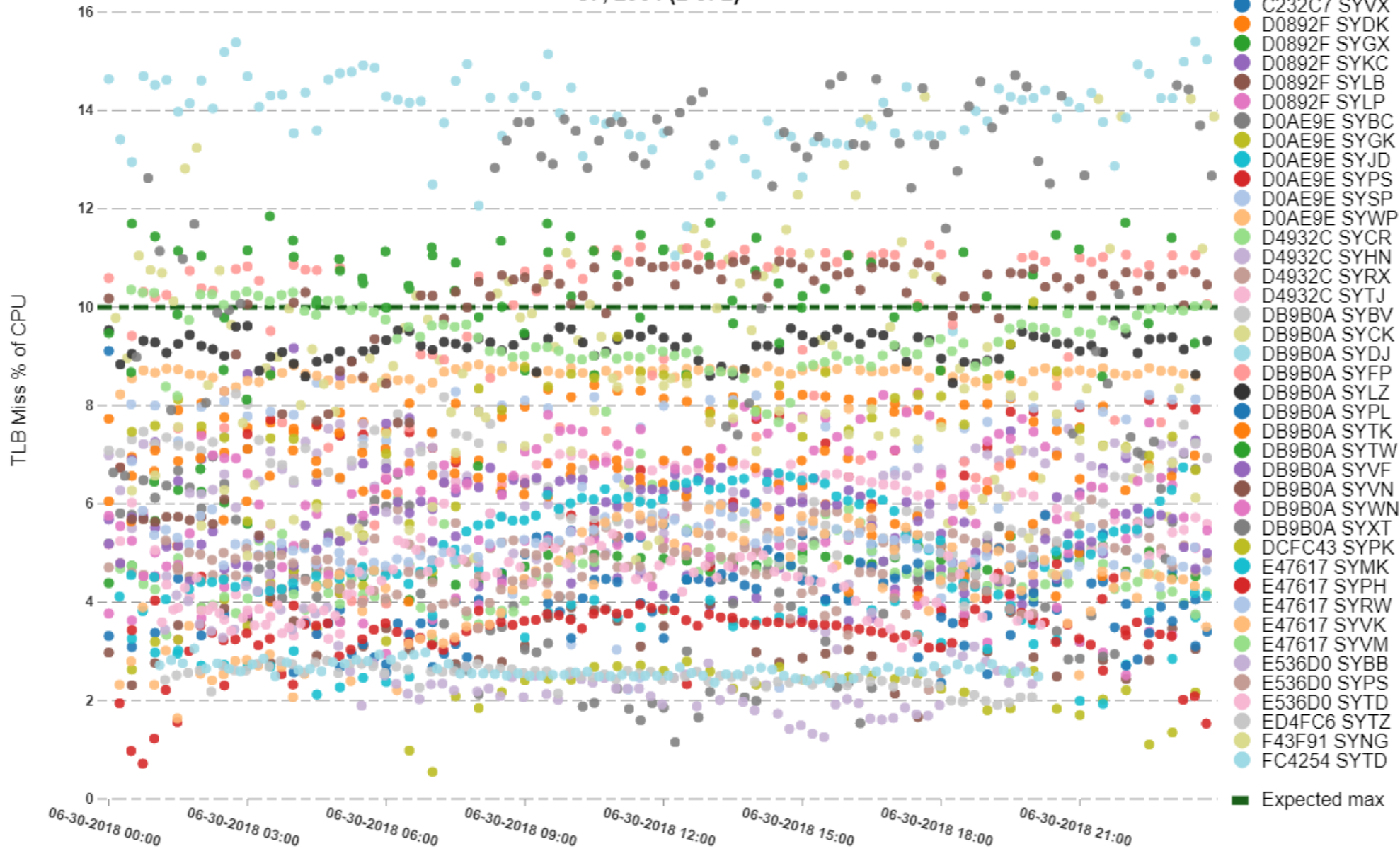


TLB Miss CPU % By z/OS Hardware Model

CP, 2964 (2 of 2)

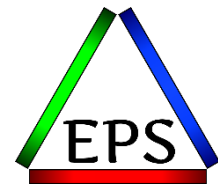


z13

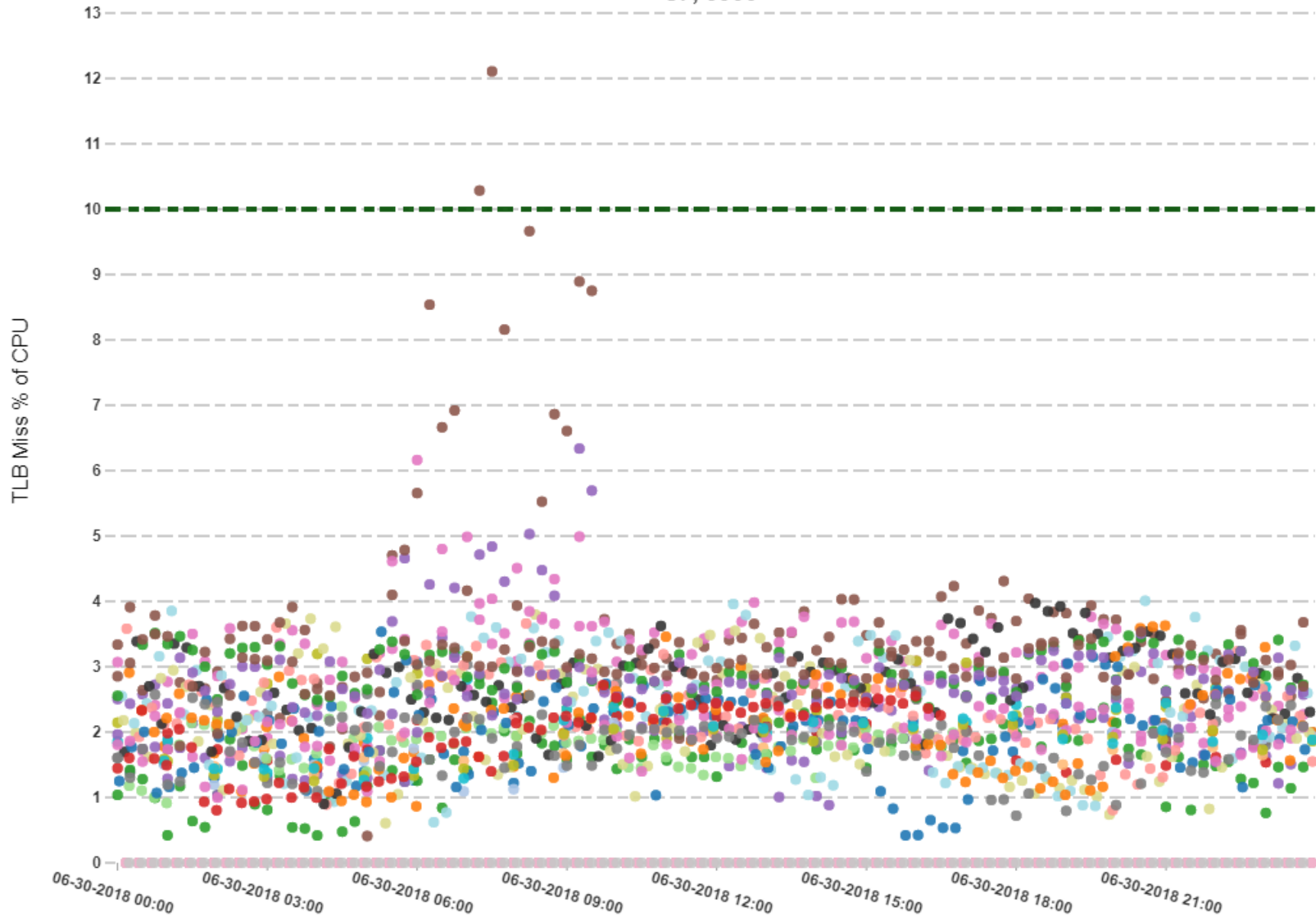


TLB Miss CPU % By z/OS Hardware Model

CP, 3906



z14



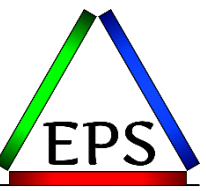
- 048FD1 SYKP
- 048FD1 SYPM
- 13DD40 SYDR
- 13DD40 SYMR
- 13DD40 SYND
- 13DD40 SYRM
- 191152 SYCZ
- 191152 SYKC
- 191152 SYKX
- 191152 SYVX
- 1E9222 SYBF
- 1E9222 SYXV
- 36A6FC SYZB
- 3C06A0 SYBJ
- 3C06A0 SYMS
- 3C06A0 SYPD
- 3C06A0 SYZC
- 75C238 SYDC
- 75C238 SYKP
- 75C238 SYQZ
- 75C238 SYWG
- 75C238 SYZV
- 75C238 SYZX
- D949E2 SYGF
- D949E2 SYHF
- D949E2 SYJX
- D949E2 SYRF

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.

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
- Use large pages where you can
 - Can reduce the TLB Miss % CPU
 - z14 greatly reduces this TLB Miss—this is about half the increase in MIPS/engine
- 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



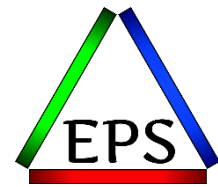
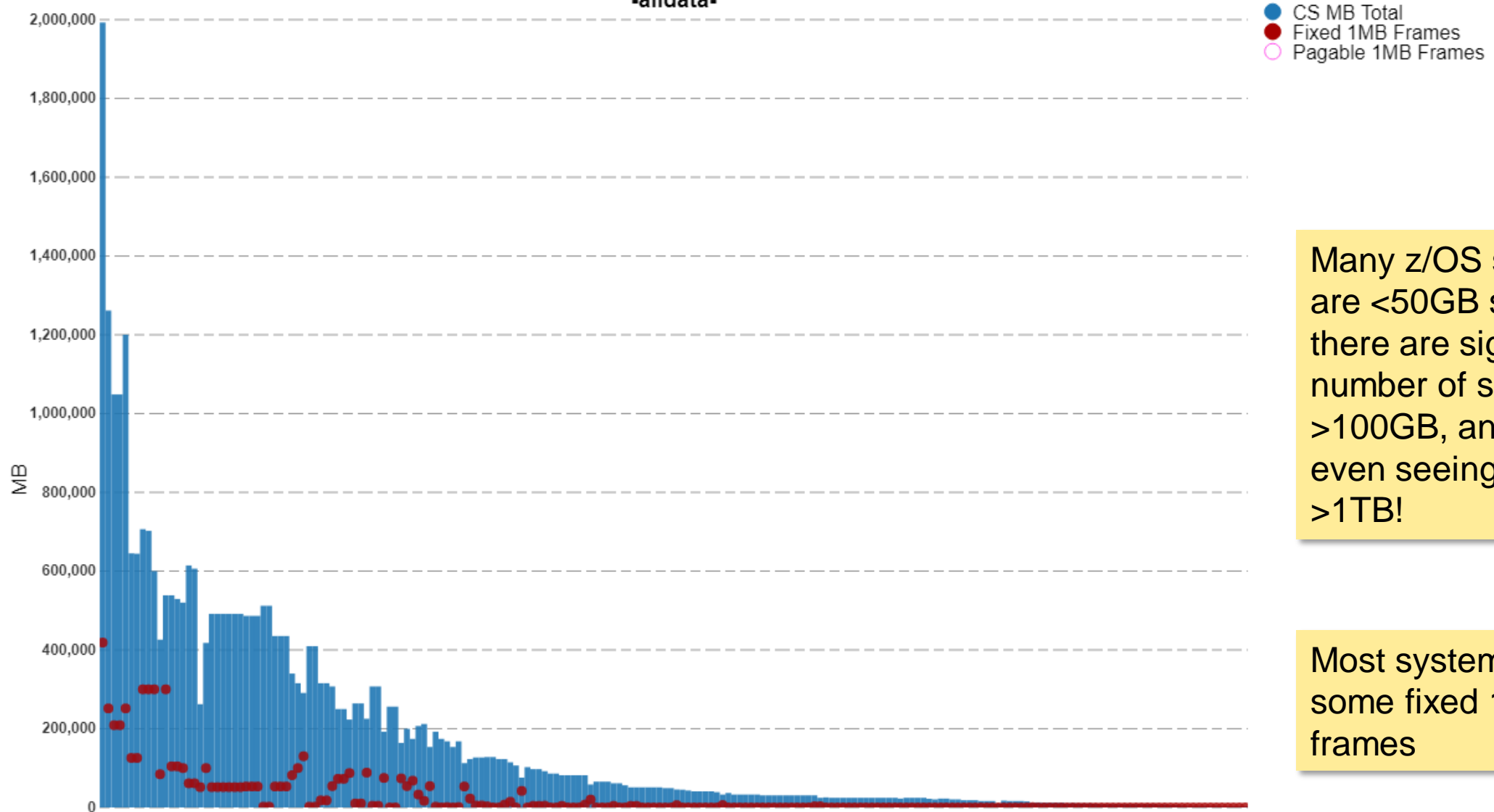
Memory: Why you care

- 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 1MB Frames

In MB

-alldata-

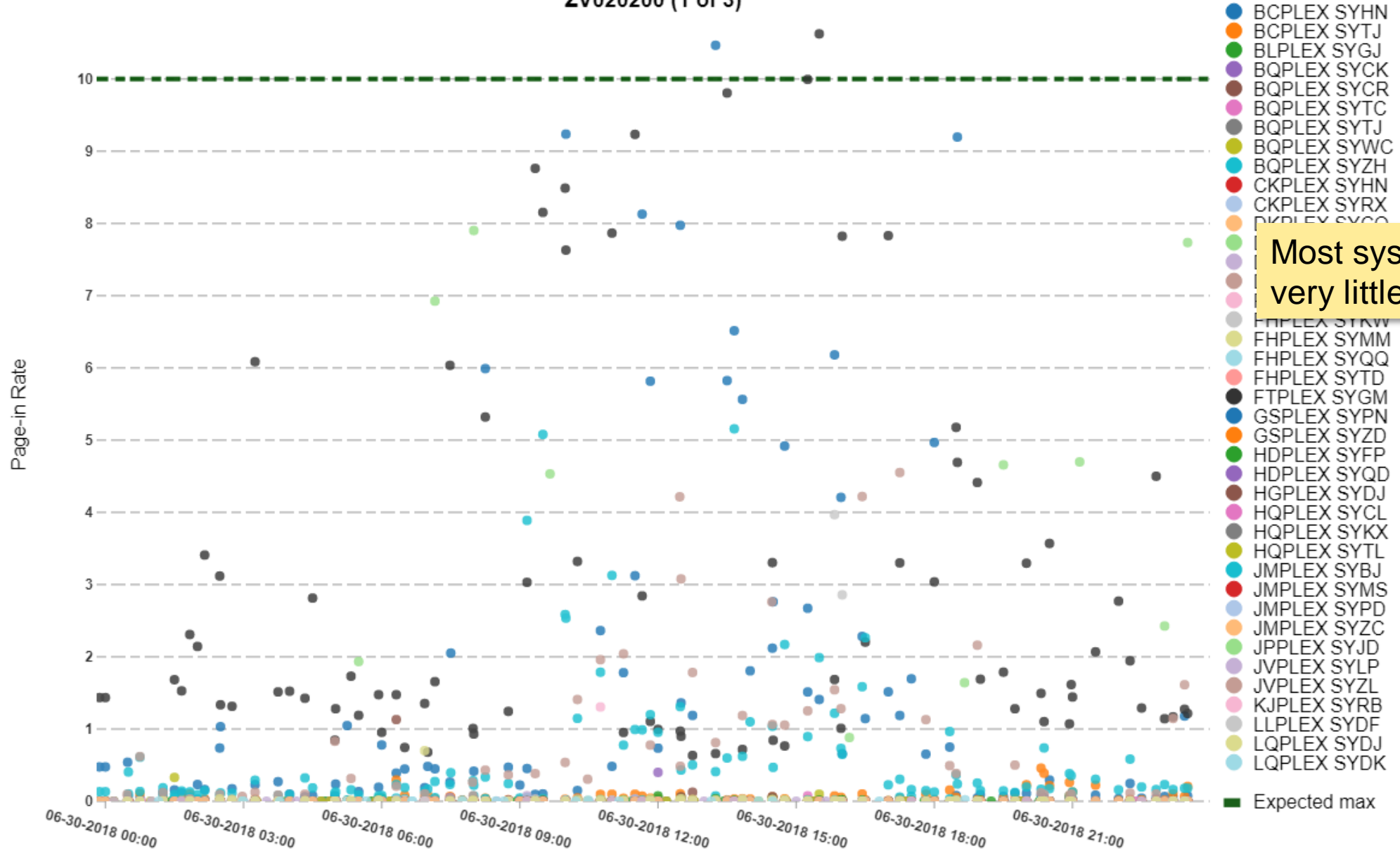
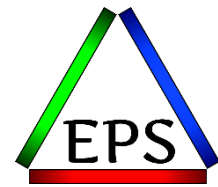


Many z/OS systems are <50GB still, but there are significant number of systems >100GB, and now even seeing a few >1TB!

Most systems using some fixed 1MB frames

Page-In Rate

ZV020200 (1 of 3)

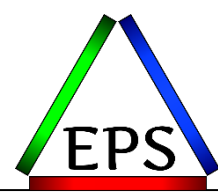


Most systems page very little to not at all

Memory: What you should do



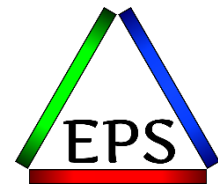
- 1MB Pages are commonly in use, if you've been trying to avoid being the pioneer, you're now safe to forge ahead
 - 2GB pages not yet common, but I suspect we'll start to see more of that with these LPARs that are hundreds of GBs
- Avoid paging: if you're regularly paging in more than a couple of pages a second, you're certainly in the minority
 - But beware of the impact of dump processing
- 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 in use for paging, you can probably shrink your DASD paging space
 - Don't forget about DR



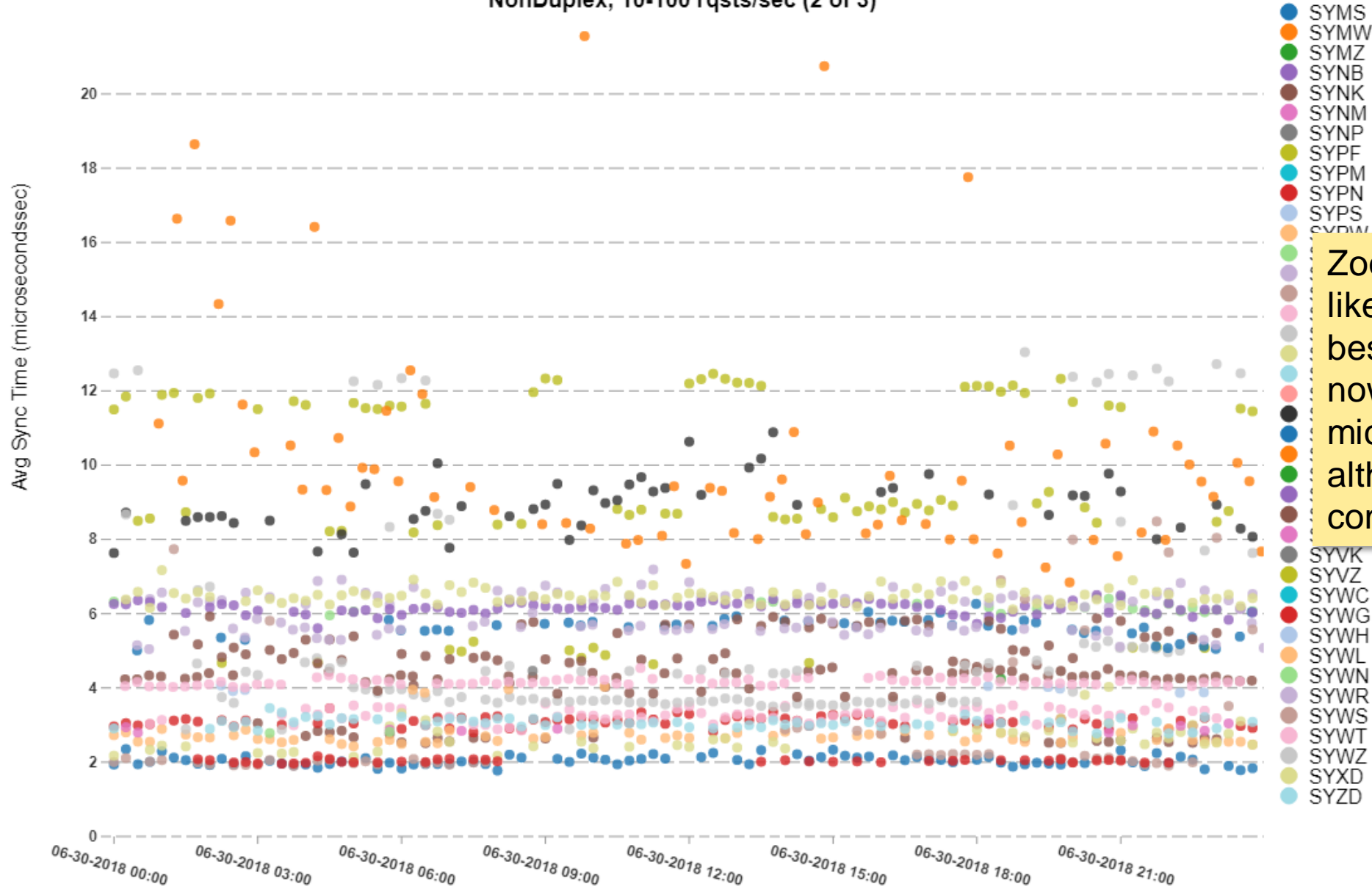
CF Sync Request Response Time: Why you care

- 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 redispached 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



NonDuplex, 10-100 rqsts/sec (2 of 3)



Zoomed in to see just likely local CFs. Note best case scenario is now down to around 2 microseconds, although 3-6 is more common.

CF Sync Request Response Time: What you should do



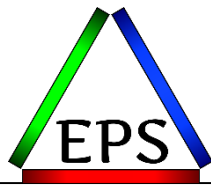
- Keep your CF hardware at the same level as your general purpose processors
 - Or at least no more than N-1
 - Less raw performance delta between generations than in the past, but latest generation gives you access to latest CFCC code and features, which may be more important
- If your lock sync times are over 10 microseconds investigate why
 - High CF busy?
 - Long distance?
 - Inefficient sharing of CF engines?
 - Duplexing?

DASD Response time components: Why you care

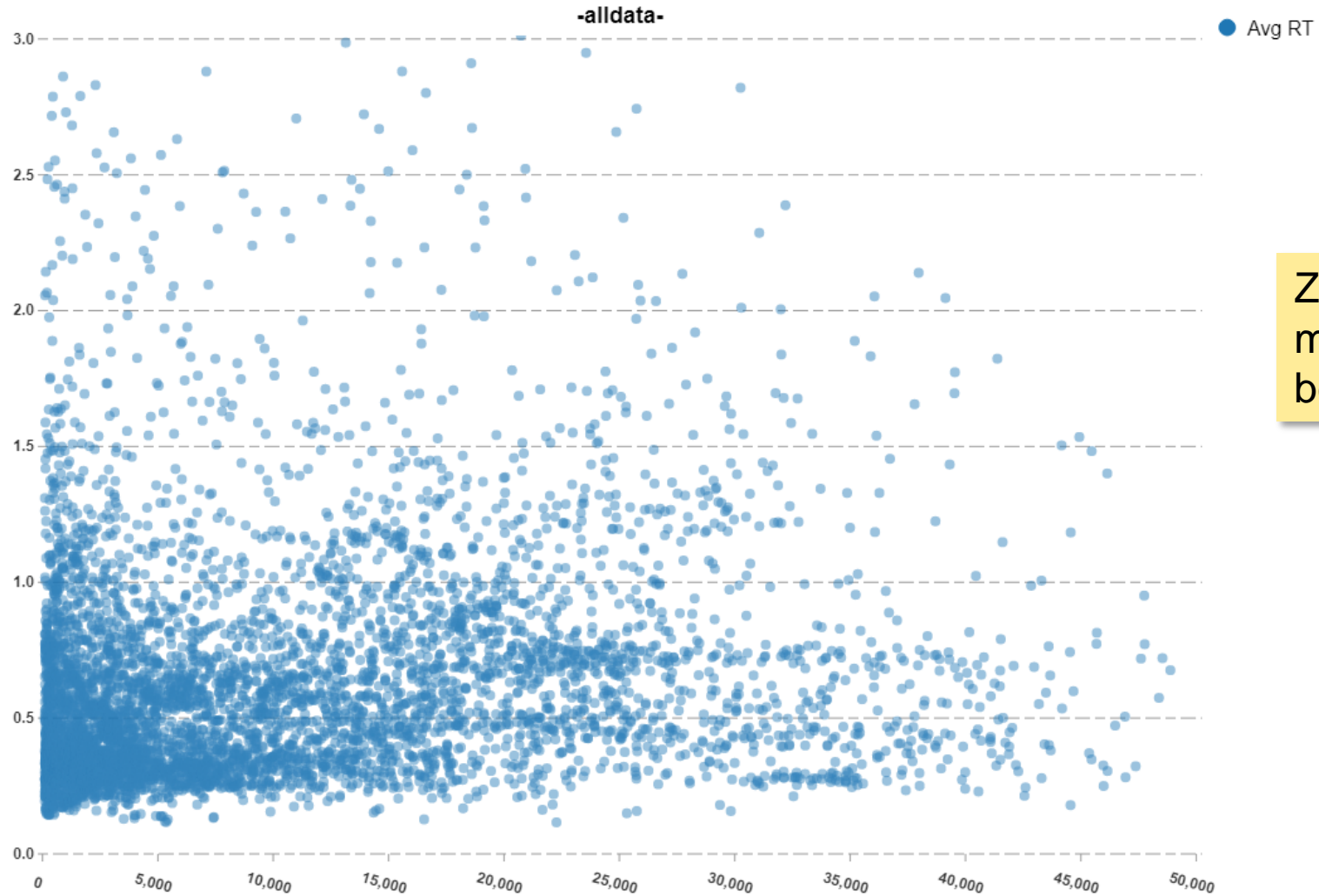


- Because we see so much SSD storage, I/O response time is somewhat less of a concern than in ages gone by
 - Larger central storage sizes can also be leveraged to reduce I/Os
- But even in the age of sub-millisecond I/Os, the only good I/O is no I/O
 - A 1 millisecond response time is a whole lot of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

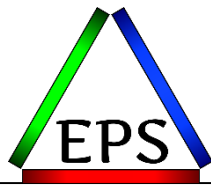


DASD Avg RT vs Activity Rate
PCU Intervals with >100 IOPS



Zoomed in to see the most usual cases better

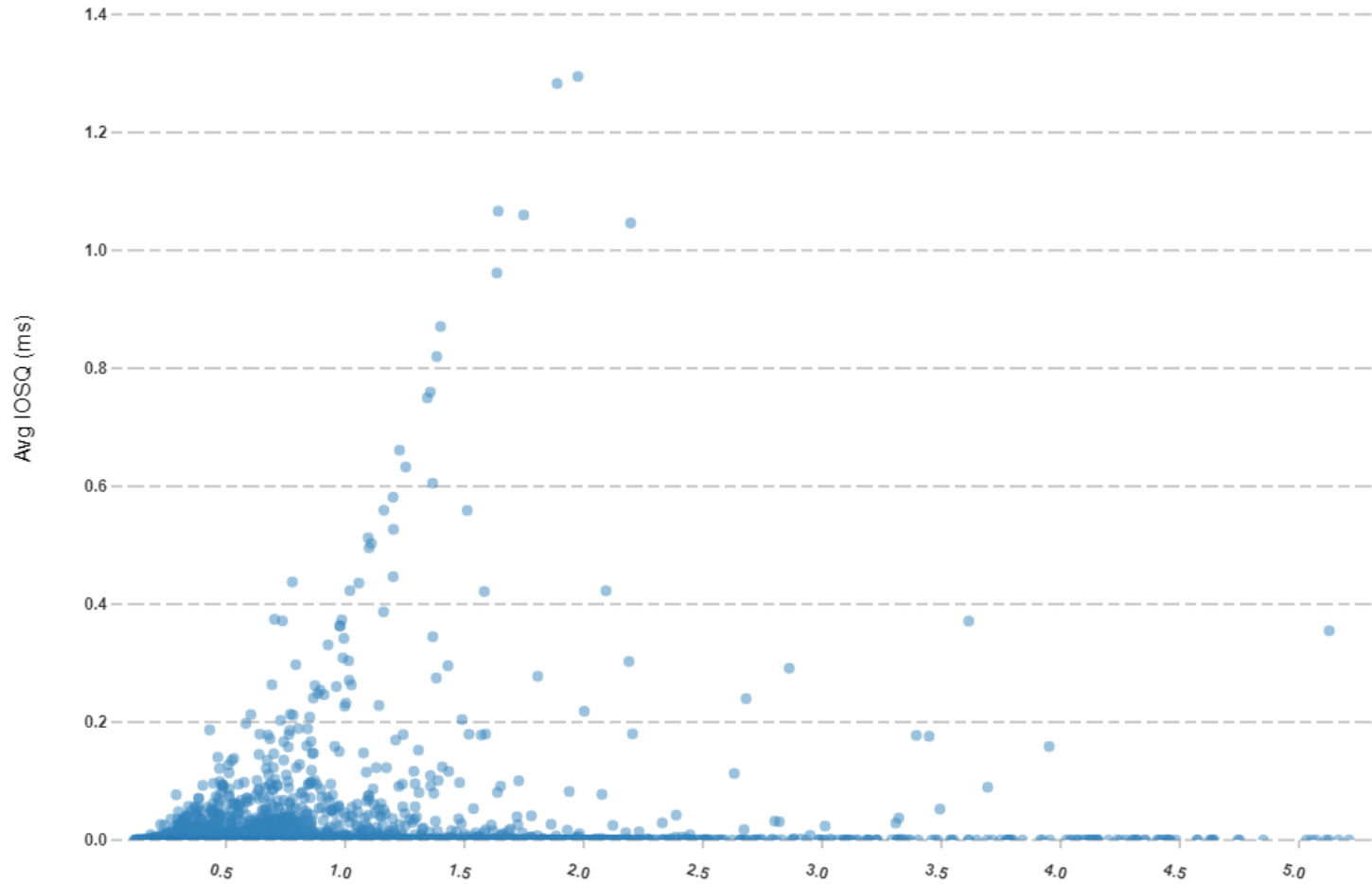
Average IOSQ vs Average RT



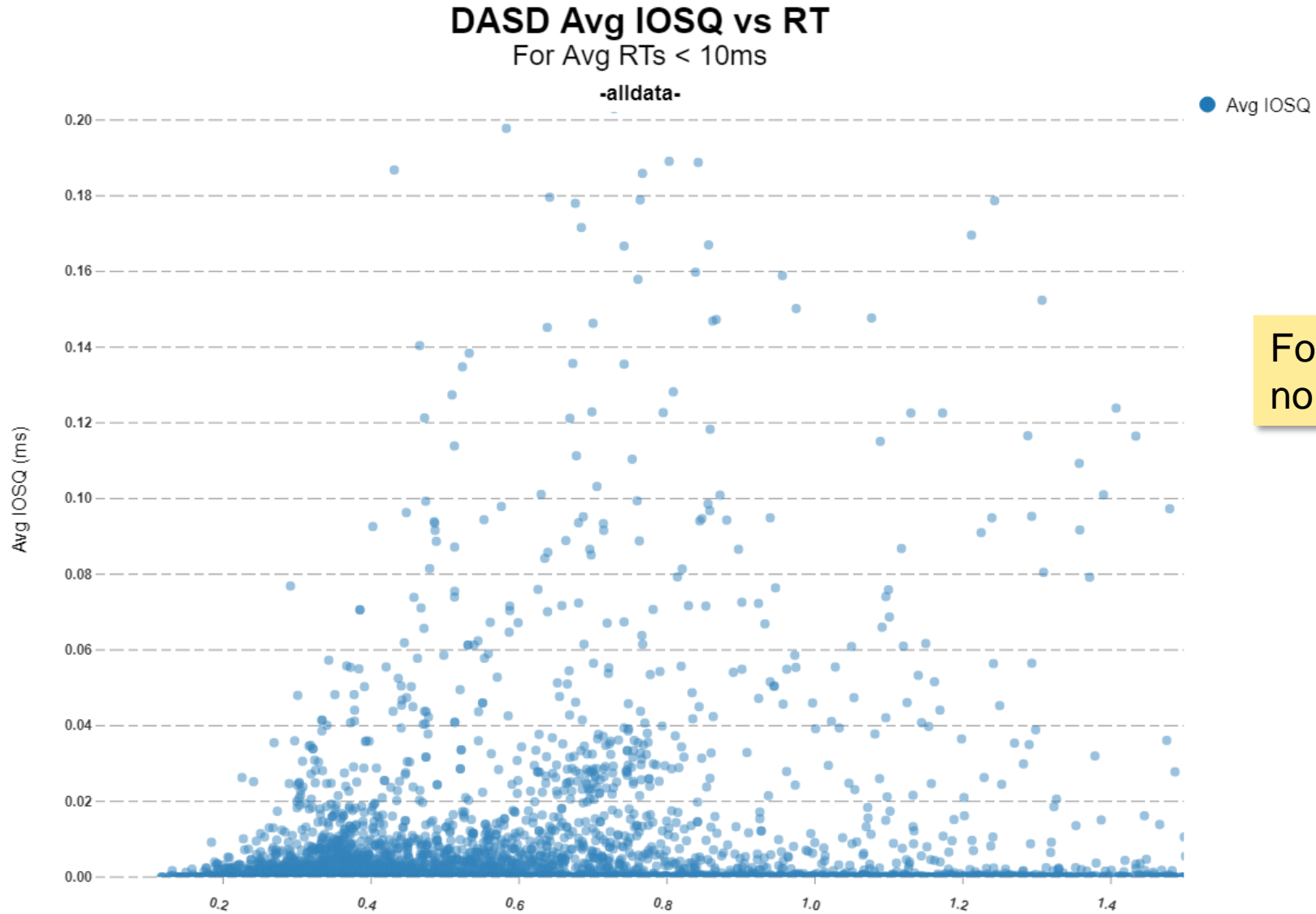
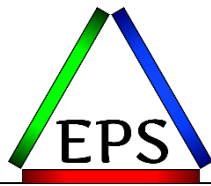
DASD Avg IOSQ vs RT
For Avg RTs < 10ms

-alldata-

● Avg IOSQ

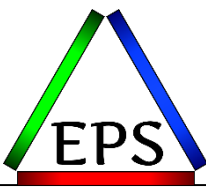


Average IOSQ vs Average RT (zoomed in)



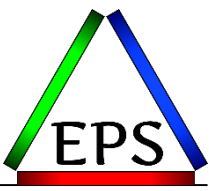
For most I/Os, there's no IOSQ

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 0.2ms you might need to investigate your PAV configuration
 - Or consider adding PAV!
- If Initial Command Response is high, look for balance issues on the controller
 - E.G. are you overloading one host adapter card?
- If overall response time is high
 - Investigate logical volume placement in the PCU (e.g. SSD vs HDD)

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 to explore in this iteration
 - Will likely update the samples in 2019 to try to capture a few 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

Please complete the session survey!

