

Measuring z/OS Performance (for Mainframe Rookies)

Scott Chapman

Enterprise Performance Strategies, Inc.

Scott.chapman@EPStrategies.com



Contact, Copyright, and Trademarks



Questions?

Send email to performance.questions@EPStrategies.com, or visit our website at <https://www.epstrategies.com> or <http://www.pivotor.com>.

Copyright Notice:

© 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

Abstract



Are you new to the mainframe and need to understand the performance of your z/OS systems? Have you seen some basic z/OS performance measurements but don't understand the source of those measurements? Or maybe you've seen some measurements that don't sound like measurements that you're familiar with on other platforms?

If you've answered "yes" to any of those questions, come to this session wherein Scott Chapman will lead you through some of the more important z/OS performance metrics and explain where they come from and why they're important.

Agenda



- Measurement basics
- A few best practices
- A review of important measurements
- Wrap-up



Measurements Basics to Understand

Measurements Lie!



- Well, maybe not exactly... but it is very easy to be misled
- You have to understand the basis of the measurements to understand what they're telling you
 - Assumptions can get you in trouble
 - Sometimes the data visualization functions more as data obfuscation
- Good news: z/OS measurements are generally well documented
 - Except when they aren't
- Bad news: not all tools avoid obfuscation issues

- Let's start with a bad (non-z/OS) example

Peak of 90%
last Weds

CPU Utilization Details

30m

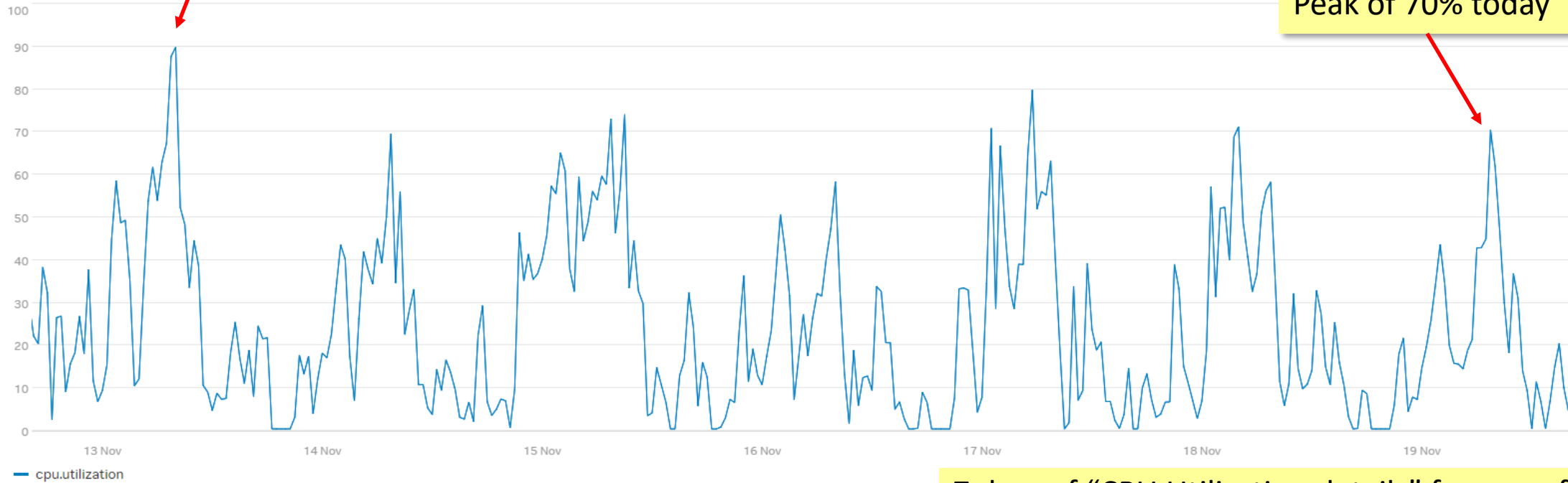


Close

Save And Close



Chart description



Peak of 70% today

7 days of “CPU Utilization details” for one of our Linux AWS servers from the “real-time” monitor we use
Not from the AWS tools (which are generally even more problematic than this!)

CPU Utilization Details

5m



Close

Save And Close




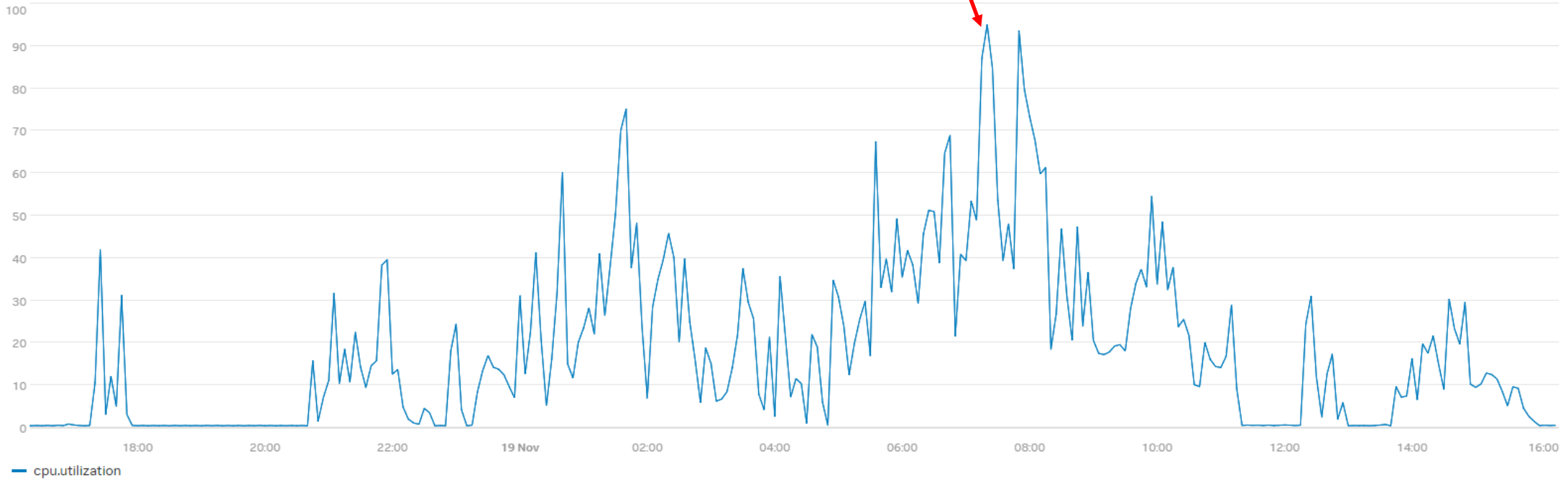
Peak of 95% today!?!


Chart description



Zoomed into the last 24 hours.
Why is my peak higher?

CPU Utilization Details

1m



Close

Save And Close



Chart description



Zoomed into those morning peaks today and the peak appears even higher! Why?

Even more important: how is "CPU utilization" measured?

What does the value represent in the interval



- In computer performance analysis, most measurements are recorded for a particular interval
- It is important to recognize what the value represents relative to the interval
 - Cumulative value – total consumption during the interval
 - May be derived from samples or comparing values at start and end of interval (delta value)
 - e.g. the job consumed 90 CPU seconds or 2000 transactions ended
 - Aggregated value – a single value representing many values from the interval
 - Commonly an average value, but may be min, max, standard deviation, etc.
 - May be calculated from samples, cumulative values, or events
 - It can be important to understand the difference!
 - e.g. average CPU utilization of the machine or maximum active work units
 - Snapshot value – the current value of a measurement
 - Captured from a single examination (sample) of the value (often at end of an interval)
 - Although often associated with an interval, these are really point in time values
 - e.g. memory configured online to an LPAR or cache structure directory entries in use

Beware the averages!!

Samples vs. Deltas



- Samples: a process periodically examines the value of interest and records the result
 - Possibly simply added to an accumulator
 - Possibly additional checks/aggregations are done
- Deltas: the starting value is subtracted from the beginning value to derive the change in value over the interval
 - Usually taken from (hopefully) monotonically increasing values
 - Although sometimes values will “wrap” back to zero resulting in a negative delta
 - Negative values may occur in some systems if the end value could not be determined and so zero was used for the ending value
 - SMF113 values are an interesting example:
 - Original subtype 2 only included the end-of-interval snapshots, requiring the user to delta the values
 - “New” subtype 1 records the values as deltas
- CPU time measurements on z/OS are based on long-term deltas of CPU timers
 - On other platforms, they may not be so clear

Sampling



- Statistically valid results need surprisingly few samples
 - Assuming the samples are captured correctly, e.g. not exact timing between samples
 - Unless doing analysis of code hotspots, probably don't need to sample 1000/sec
 - And maybe not even then
- SRM samples the system 4 times / second (250ms)
- RMF cycle time of 1000ms is the default
 - Reducing the cycle time will increase RMF CPU somewhat
 - Very short cycle times are not recommended, but somewhat shorter cycle times might better catch shorter transient conditions
 - Use CYCLE time in range of 250 to 1000

Top SMF Records for Performance

This is not an
exhaustive list!



- 30 – Batch job and started task details, including totals and intervals
- 42 – I/O details at a dataset level
- 70-79 – various measurements from RMF
 - 70 – Processor/LPAR details
 - 71 – Paging
 - 72 – WLM SC/RC details
 - 73 – Channel Path details
 - 74 – I/O, XCF, CF, etc.
- 88 – System Logger
- 98 – High Frequency Throughput Statistics
- 99 – SRM/WLM Decisions



Best Practices

z/OS Interval Configuration



- Most customers use an RMF interval length of 15 minutes (900 seconds)
 - 15 minutes is the *maximum* interval that we recommend
 - 5 or 10 minutes can be beneficial
 - Changing RMF interval length only changes the amount of data written out: collection overhead is same
 - We have seen customers with 1 minute RMF I intervals (but we **don't** recommend this)
 - Biggest issue with this is related to the type 74 records
- Generally, synchronize your RMF and SMF intervals (eases certain analyses)
 - Set SMFPRMxx INTVAL(15) (or possibly less)
 - Set ERBRMFxx SYNC(SMF) (ignores RMF option INTERVAL)
 - May want longer SMF intervals though if you're using really short RMF intervals
- RMF III intervals of 60 or 100 seconds are common
 - MINTIME(100) is default, MINTIME(60) is my preference
- Use the same **CYCLE, MINTIME, and INTERVAL** on all systems

High Frequency Measurements



- Sometimes when analyzing a problem the RMF interval will be too long
 - See previous slide
- Your online monitors may be of use, in particular if they've captured some history
 - What is that historical interval that they capture?
- But there is SMF data that can help too, if you capture it
 - SMF 99.6 – 10 second interval data on a service class basis
 - SMF 99.12 – HiperDispatch utilization values on a 2 second basis (CEC utilization and MVS busy)
 - SMF 98 – High Frequency Throughput Statistics (z/OS 2.3)
 - Many different measurements, primarily on a system basis, or CPU type basis
 - Interval can be 5, 10, 15, 20, 30, or 60 seconds
 - Seriously consider recording these records
 - If you have a problem, it's too late to enable recording
 - If you record the data and never have a problem, consider that a small price to pay!

SMF 99 Recording Recommendation



- 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 50-150MB of data per system per day but can be very interesting or important when doing a performance analysis
- The other subtypes are more rarely used, but I'm not going to dissuade you from recording them
 - Subtypes 1 and 2 would be the next set I'd recommend (2s can be voluminous)
 - Subtype 13 are "IBM use only" and voluminous: disable unless IBM requests

SMF 98 Recording Recommendations



- Record them if you're on z/OS 2.3 or above
 - Set HFTSINTVL in SMFPRMxx
- SMF Manual recommends HFTSINTVL(20)
 - I've heard other IBMers recommend 5 second intervals
- Regardless of interval, if you have them enabled, 4 times an hour you get 5 second intervals for 1 minute
 - For minutes 00, 15, 30, and 45
- Data volume will depend on interval and system activity
 - 20 second intervals: 60-150 MB/system/day ($56 * 3 + 4 * 12 = 216$ intervals / hour)
 - 5 second intervals: 200-500 MB/system/day ($60 * 12 = 720$ intervals / hour)
- It would be nice to have consistent 5 second intervals, but 20 seconds may not be a bad starting place

Data Visualization Tidbits to Remember



- Line graphs can hide the individual points
 - Bar (column) graphs can make it easier to discern the individual intervals and the interval duration
- Y-axis should almost always start at zero
- Avoid “chart junk” (visual clutter that detracts from understanding the data)
 - Useless & confusing 3-D effects
 - Unnecessary gradients and backgrounds
- Pie charts are inherently difficult to understand
 - Angles and areas harder to evaluate than length
- Color blindness affects about 10% of male population



Measurements Review

CPU Busy %



Source: SMF 70

Type: Cumulative Delta of CPU timers

Interval: RMF (typically 900s)

Scope: Particular CPU pool for a CEC (usually shown as the sum of all the LPARs)

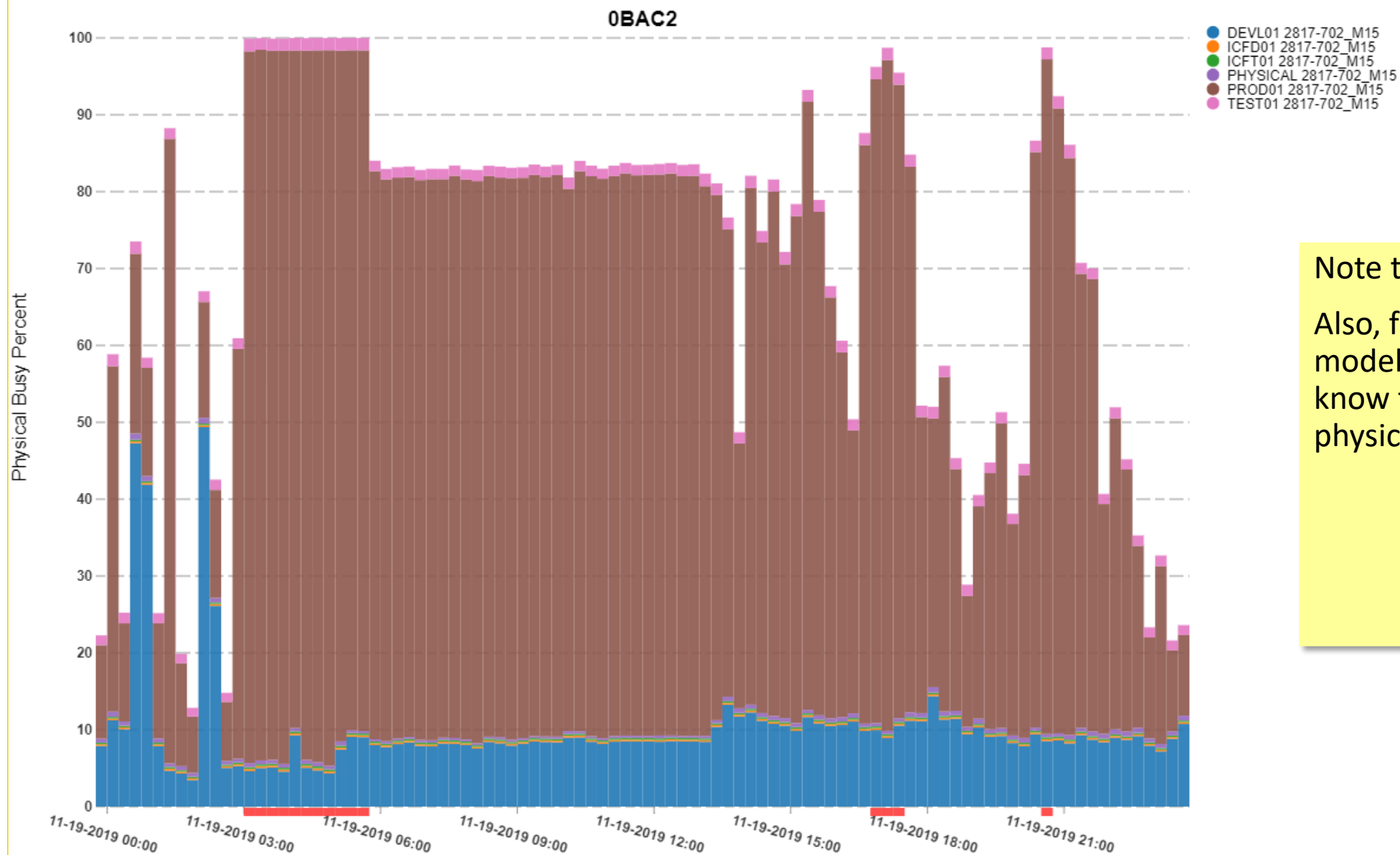
Meaning: This is the percent busy (sometimes, “utilization”) expressed as a percentage of the machine capacity. $\text{CPU time} / (\text{CPUs} * \text{Interval duration})$

Look for: High values aren't necessarily a problem per se, but do increase the chance that lower importance work is waiting. Generally speaking, efficiency is better at lower utilizations.

Usually Busy % and Utilization % are interchangeable, but for zIIP pool, if you have SMT enabled busy % is almost always greater than utilization %. But in that scenario, utilization % is just an estimate.

Plateaus may indicate capping enforced.

CEC Physical Machine CP Busy% by CEC Serial Number



Note the plateau here.

Also, from the machine model (2817-702), we know this machine has 2 physical CPs.

CEC Busy % (2 seconds)



Source: SMF 99

Type: Cumulative Delta of CPU timers

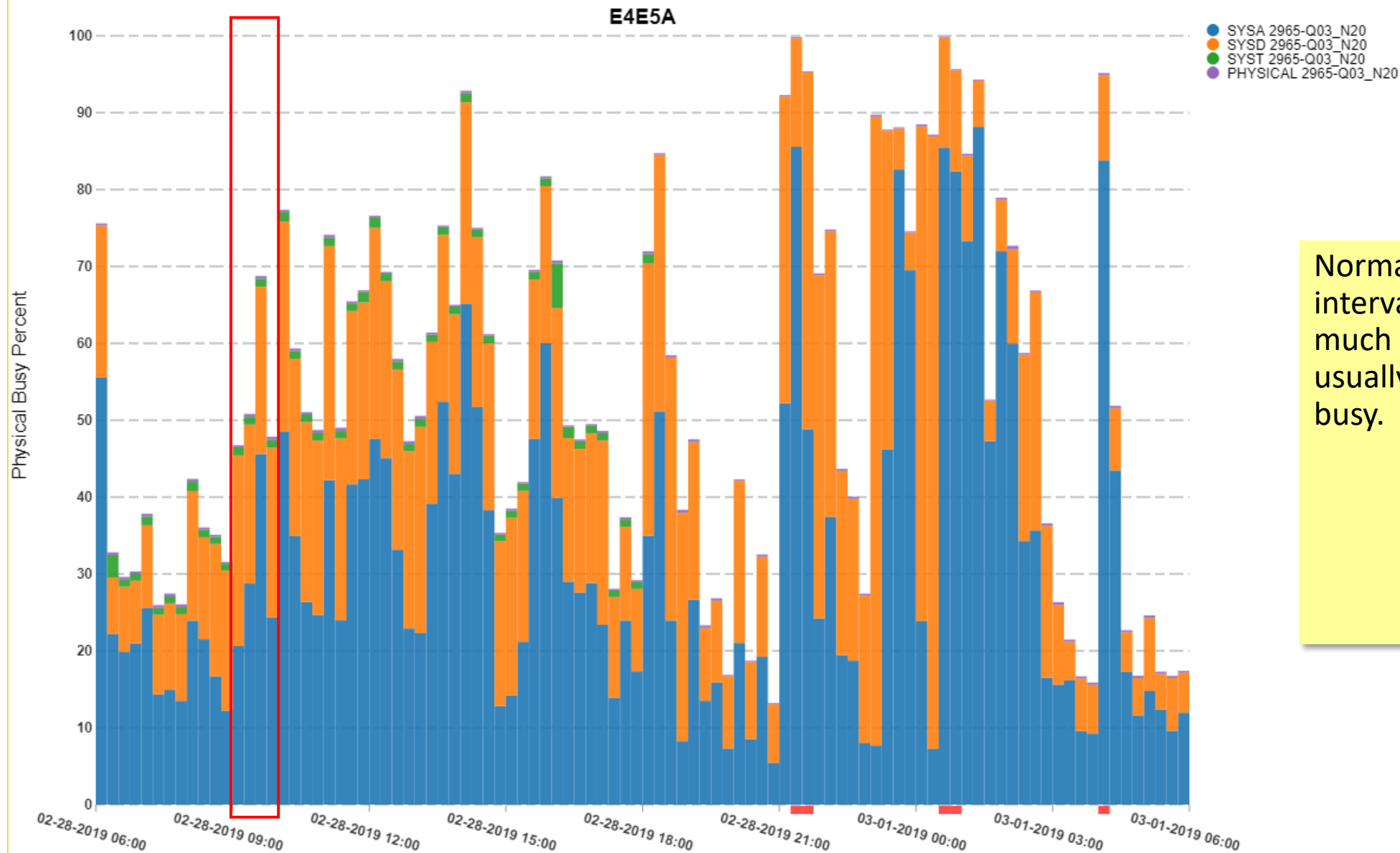
Interval: 2 Seconds

Scope: Particular CPU pool for a CEC

Meaning: This is the percent busy (sometimes, “utilization”) expressed as a percentage of the machine capacity. $\text{CPU time} / (\text{CPUs} * \text{Interval duration})$

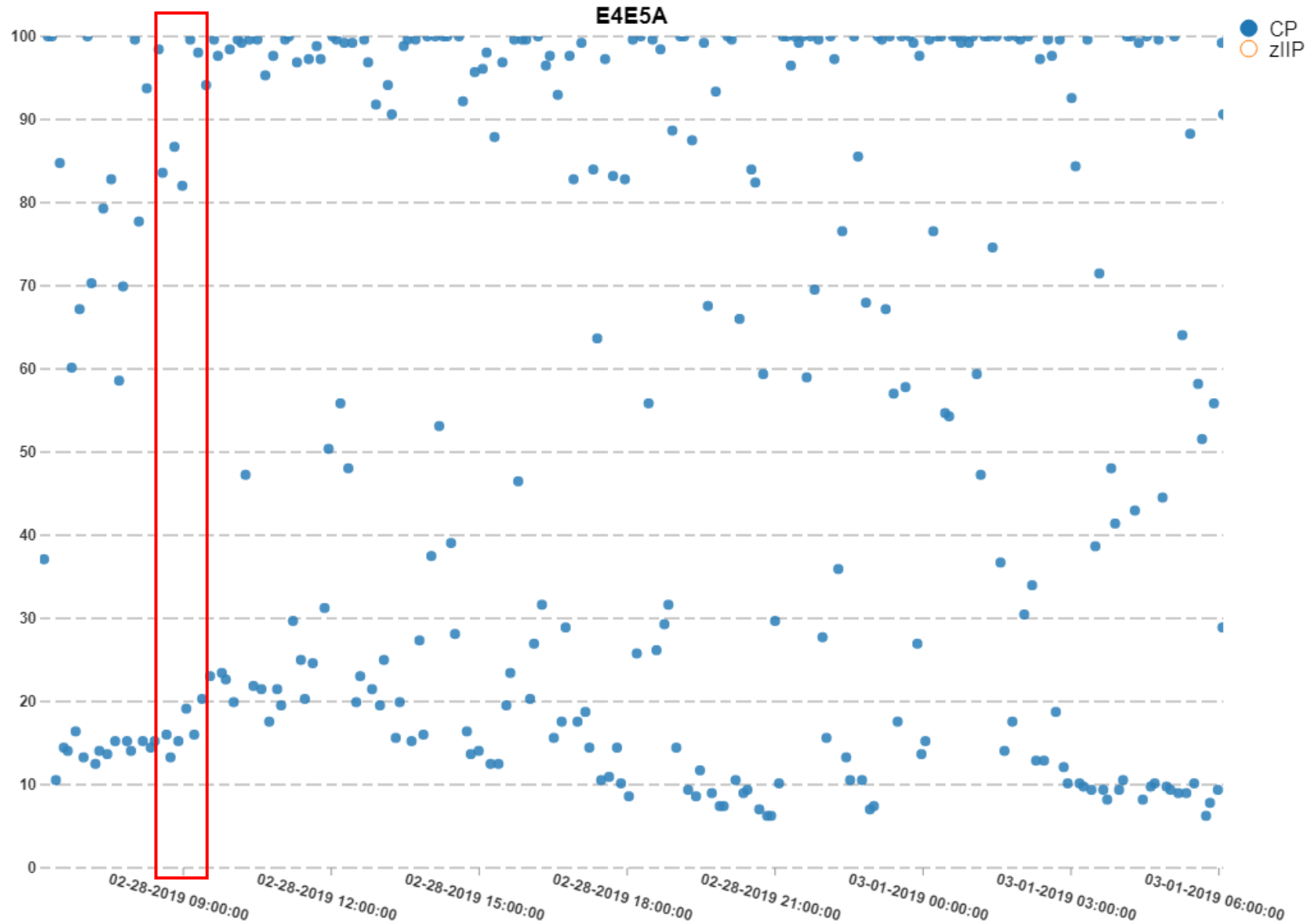
Look for: High values aren't necessarily a problem per se, but do increase the chance that lower importance work is waiting. Generally speaking, efficiency is better at lower utilizations. Usually Busy % and Utilization % are interchangeable, but for zIIP pool, if you have SMT enabled busy % is almost always greater than utilization %. But in that scenario, utilization % is just an estimate. Plateaus may indicate capping enforced.

CEC Physical Machine CP Busy% by CEC Serial Number



Normal 15-minute intervals, below 80% for much of the daytime, usually even below 70% busy.

HiperDispatch CEC Utilization



This is a min-max aggregation of the 99.12 CEC Busy 2-second data which shows that there are at least some 2-second intervals that are 100% busy.

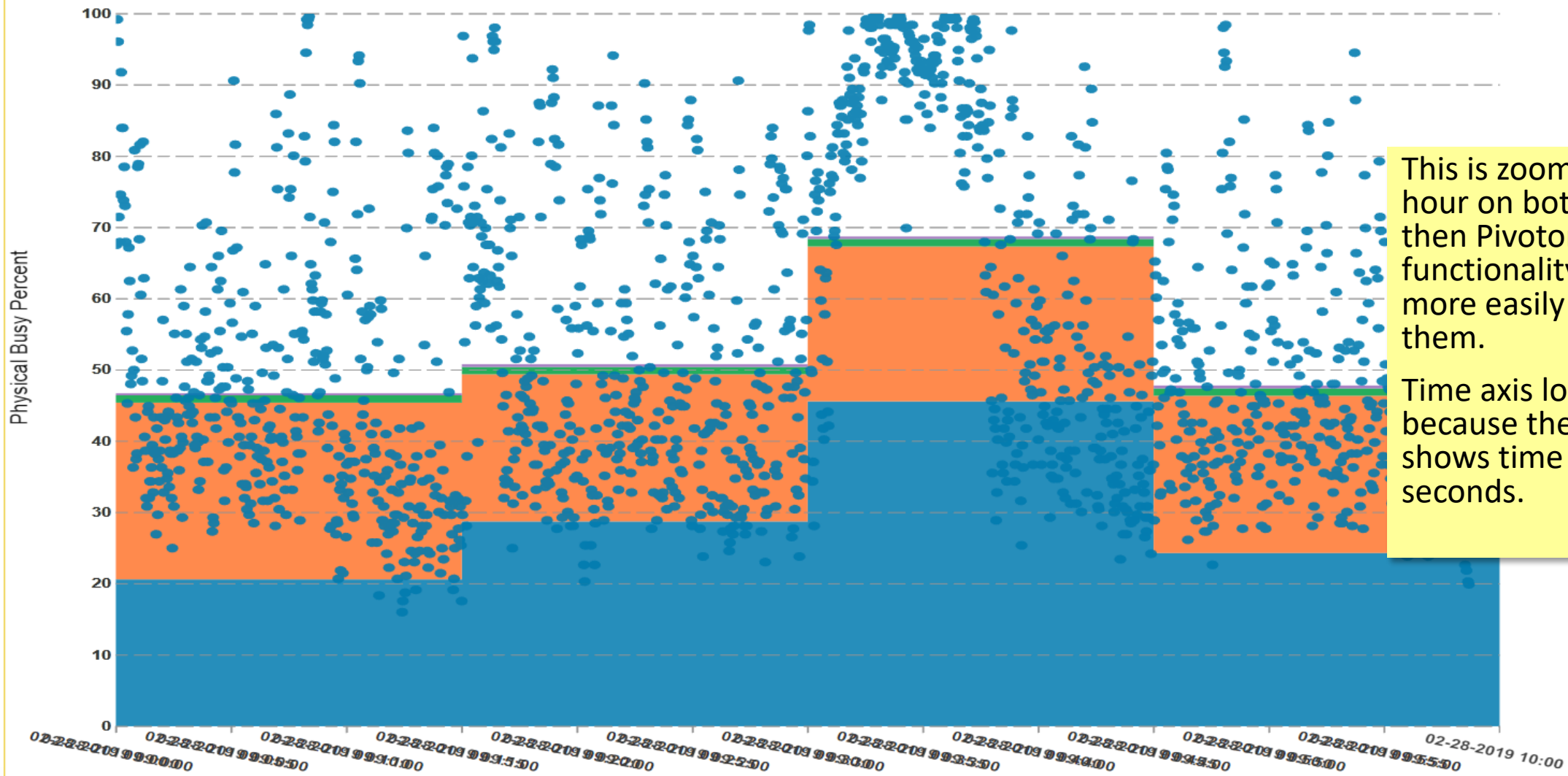


Opacity

Titles Legend Grid

Chart

CEC Physical Machine CP Busy% by CEC Serial Number
 HiperDispatch CEC Utilization



This is zoomed into 1 hour on both charts and then Pivotor's overlay functionality was used to more easily compare them.

Time axis looks messy because the 99.12 chart shows time with seconds.

CP APPL%



Source: SMF 72

Type: Accumulated deltas

Interval: RMF (typically 900s)

Scope: Service Class Period or Report Class

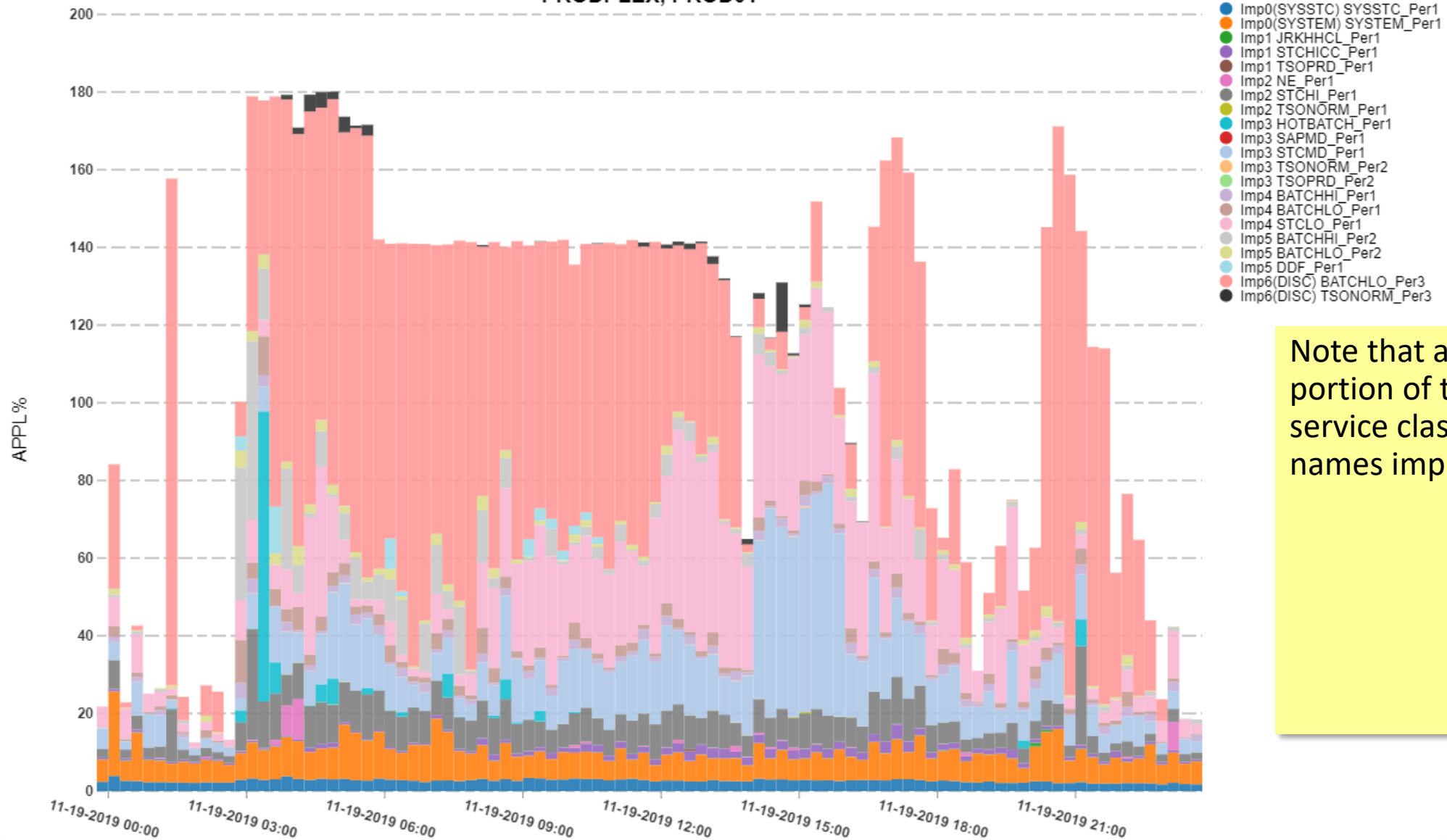
Meaning: APPL % (standing for application percentage) is the percent of a single physical CPU (or zIIP if looking at zIIP APPL%) that the work is consuming. There are multiple ways representing the CPU consumed by a workload, but this is one of my favorites because I find it more intuitive.

Look for: Unexpectedly high/low numbers.

WLM CPU - CP APPL% by Service Class

(CP + zAAP on CP + zIIP on CP)

PRODPLEX, PROD01

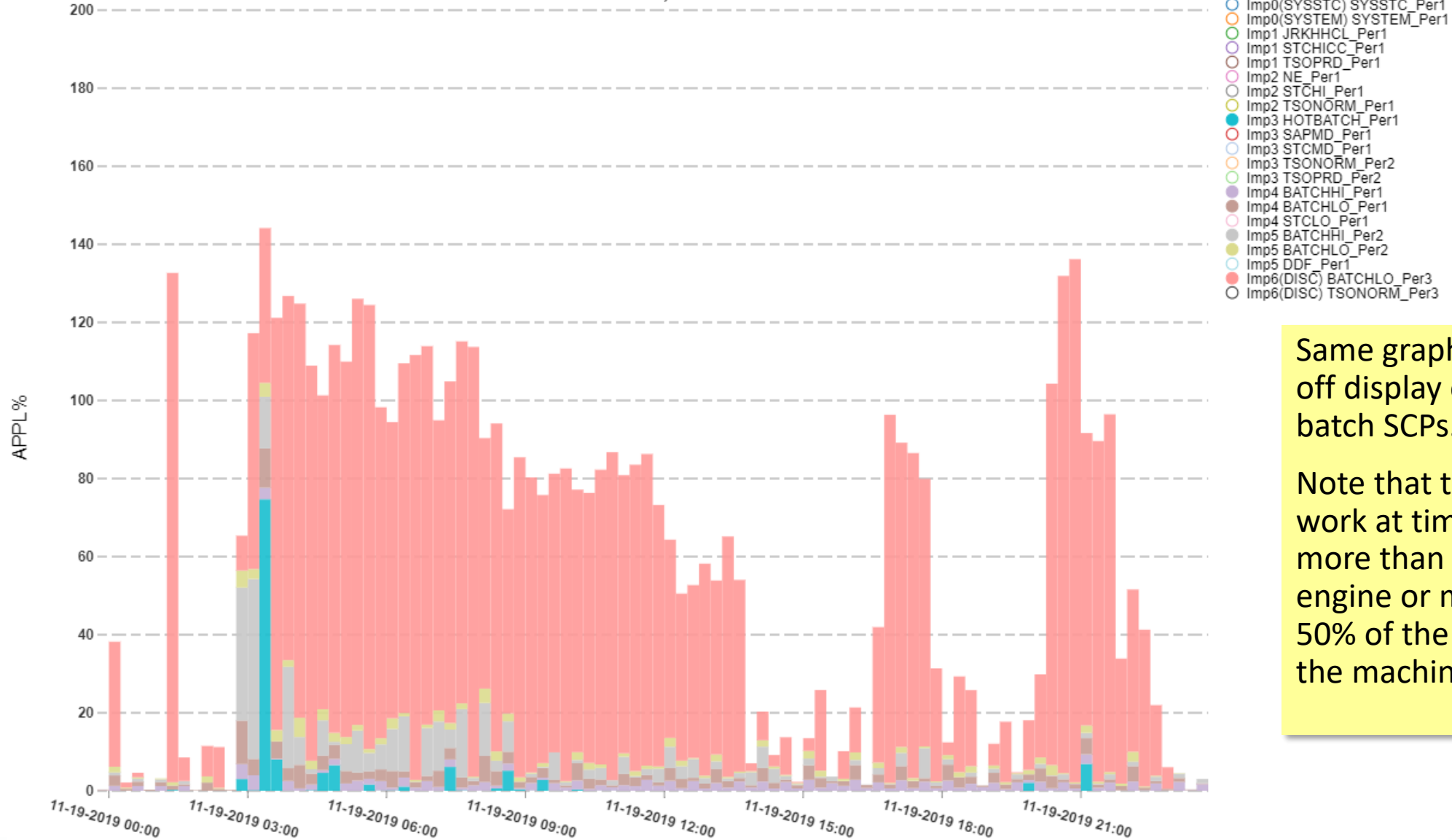


Note that a significant portion of the work is in service classes whose names imply batch work.

WLM CPU - CP APPL% by Service Class

(CP + zAAP on CP + zIIP on CP)

PRODPLEX, PROD01



Same graph, just turned off display of all but batch SCPs.

Note that the batch work at times was using more than 100% of one engine or more than 50% of the capacity of the machine.

Capture Ratio



Source: SMF 70 and 72

Type: Calculated

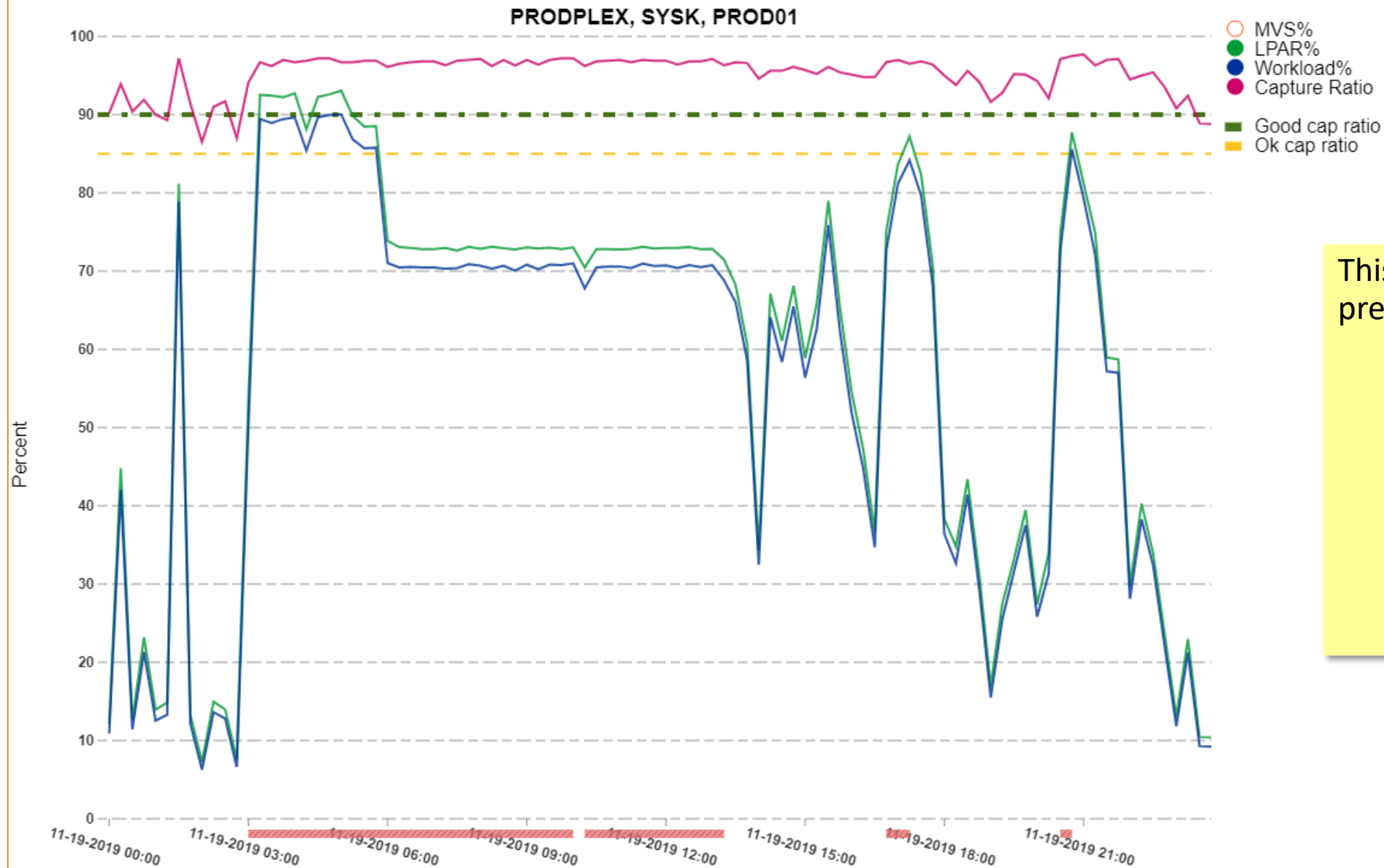
Interval: RMF (typically 900s)

Scope: LPAR

Meaning: Not all of the CPU time consumed by the LPAR is “captured” on the SMF 72 records. By comparing the CPU consumed by the LPAR (SMF 70) to the CPU consumed by the service classes (SMF 72) we can calculate the capture ratio.

Look for: There will always be some uncaptured time because there are system tasks that consume CPU that are not done directly on behalf of any one service class. We expect to see capture ratios above 90% and best in class will be around 95%. Significant LPARs with regular capture ratios less than 85% are worthy some further investigation.

LPAR CP Busy%, MVS CP Busy%, Workload CP Busy%, and Capture Ratio



This capture ratio is pretty much ideal.

Finding uncaptured time



- Some uncaptured time is normal—will never get to 100% capture ratio
 - Unless you use the IBM practice of combining CP and zIIP consumption for capture ratio (don't do this) and turn on SMT (maybe do this) in which case capture ratio can appear to exceed 100%!
- Smaller or idle systems will generally have lower capture ratios
- A few things to look for:
 - Contention
 - Paging
 - SLIP traps and system traces
 - Inefficient allocation process

LPAR Capping Percent



Source: SMF 70

Type: Averaged sampled values

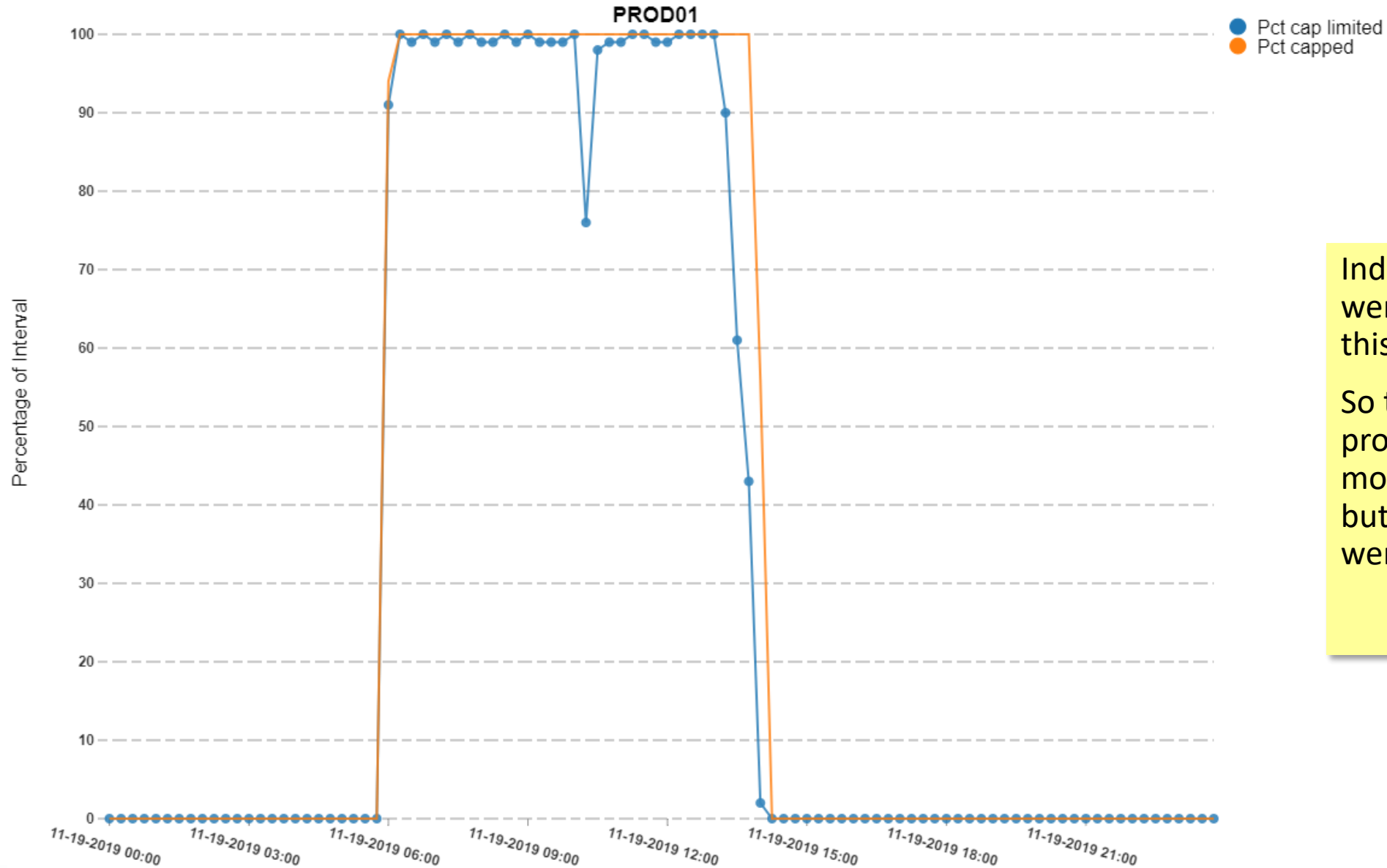
Interval: RMF (typically 900s)

Scope: LPAR

Meaning: There's two capping percentages: Percent considered capped is the percentage of the interval the LPAR was considered capped. The more useful percent capping actually limited, indicates what percentage of the interval was the LPAR demanding more capacity than the cap would allow.

Look for: If the “actually limited” value is low, then the cap is not significantly degrading work. If the “actually limited” percentage is high, then some work is suffering to at least some degree.

CEC Capping Actually Limited Percentage vs. Percent Considered Capped



Indeed, the systems were being capped in this example.

So the customer probably saved some money vs not capping but what workloads were actually suffering?

CPU Delay Samples



Source: SMF 72

Type: Accumulated sample counts (at $\frac{1}{4}$ second intervals)

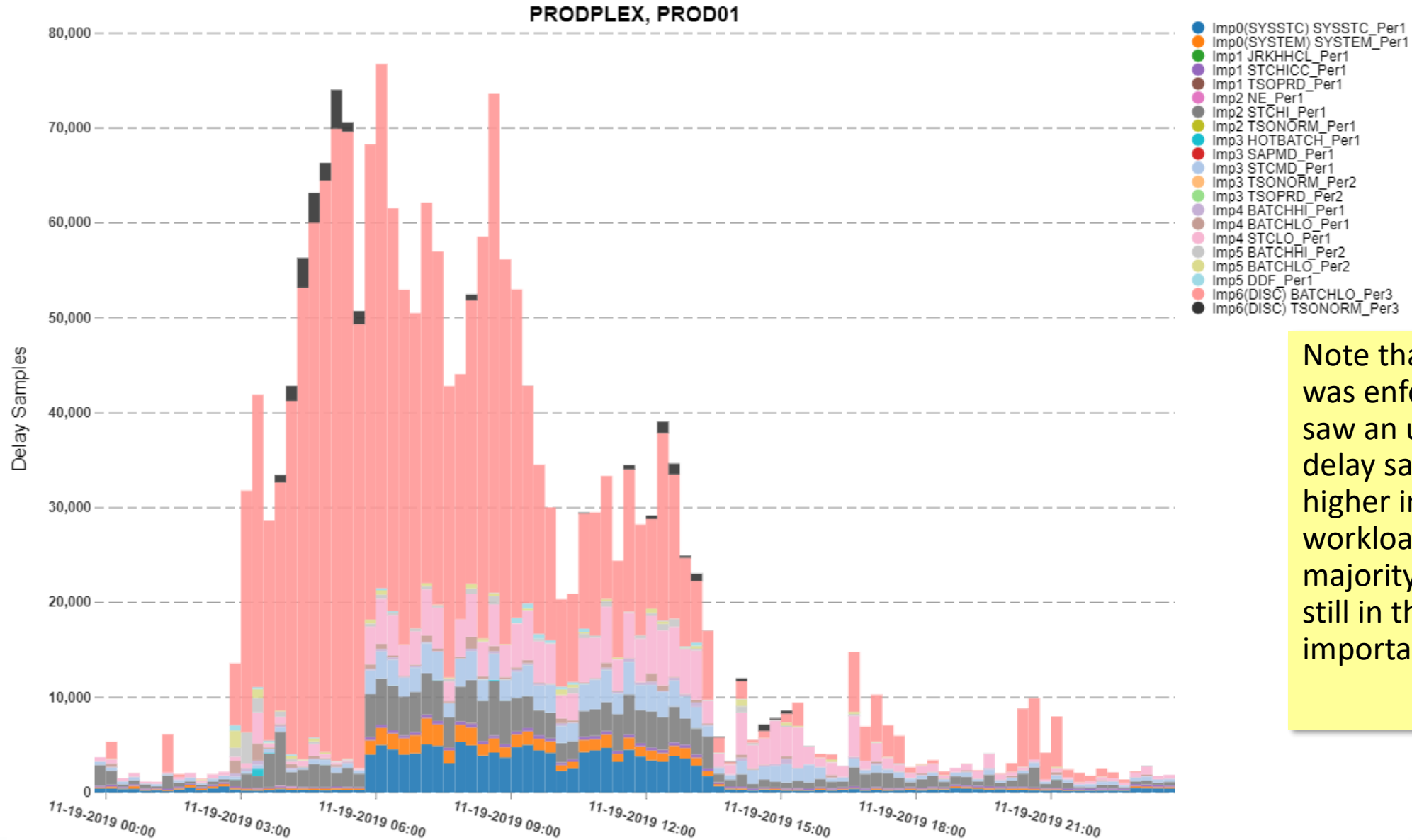
Interval: RMF (typically 900s)

Scope: Service Class Period

Meaning: Work in the systems is sampled every $\frac{1}{4}$ second to determine its state. The delay samples are accumulated and externalized in the SMF 72 records by service class period and can be used to understand the level of CPU contention in the system.

Look for: Sudden increases delay samples for a given workload likely means that something changed such that it is suffering more. Possibly this could be due to an increase in work units in the system, or could be due to an increase in utilization, capping, weight enforcement, etc.

WLM CPU - CP CPU Delay Samples By Period

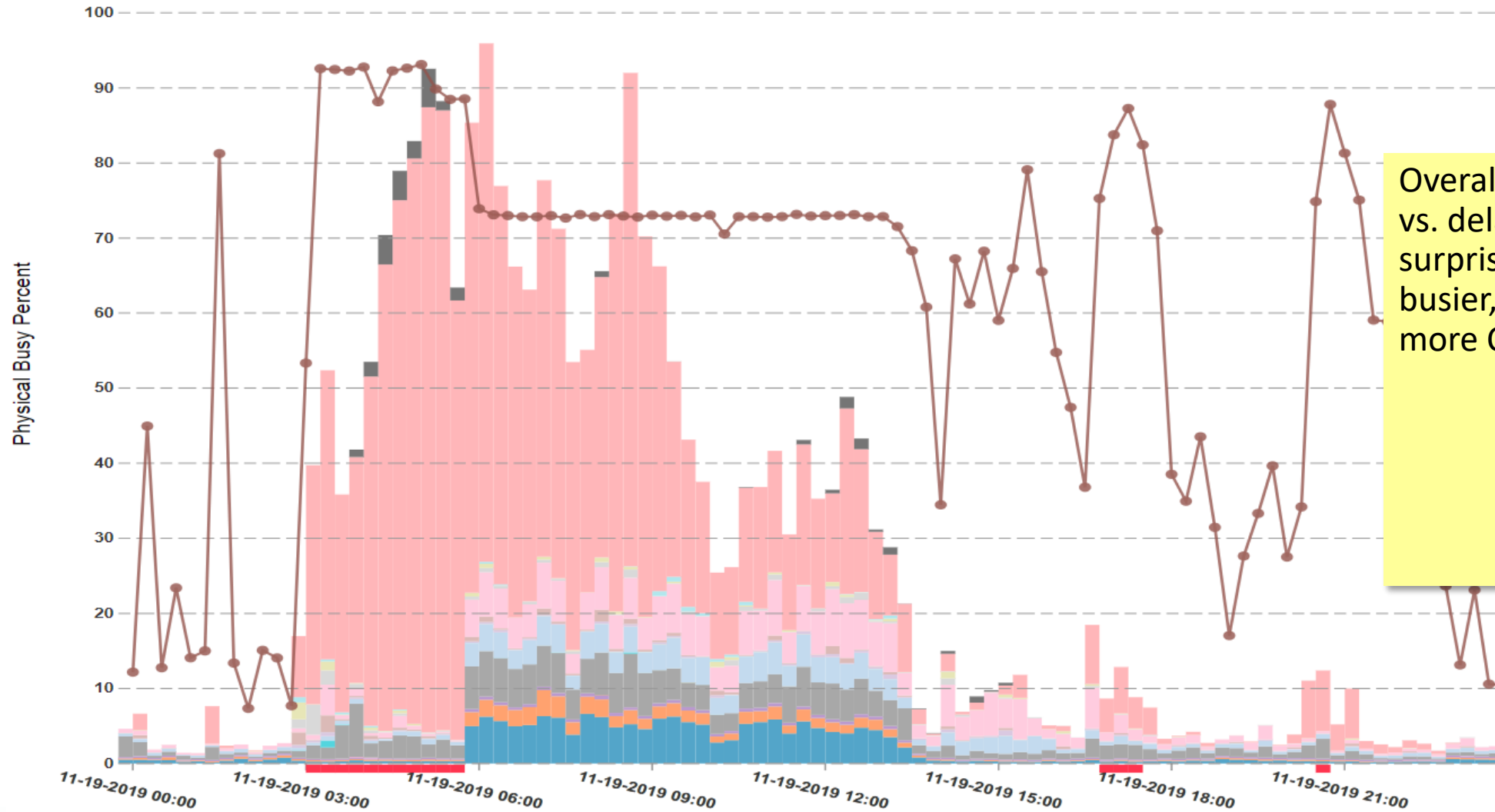


Note that when the cap was enforced this system saw an uptick in the delay samples of the higher importance workloads, although the majority of the delay is still in the lower importance workloads.



Opacity Titles Legend Grid Chart

WLM CPU - CP CPU Delay Samples By Period
CEC Physical Machine CP Busy% by CEC Serial Number



Work Units (min, max, avg)



Source: SMF 70

Type: Aggregated sample values (at ¼ second intervals)

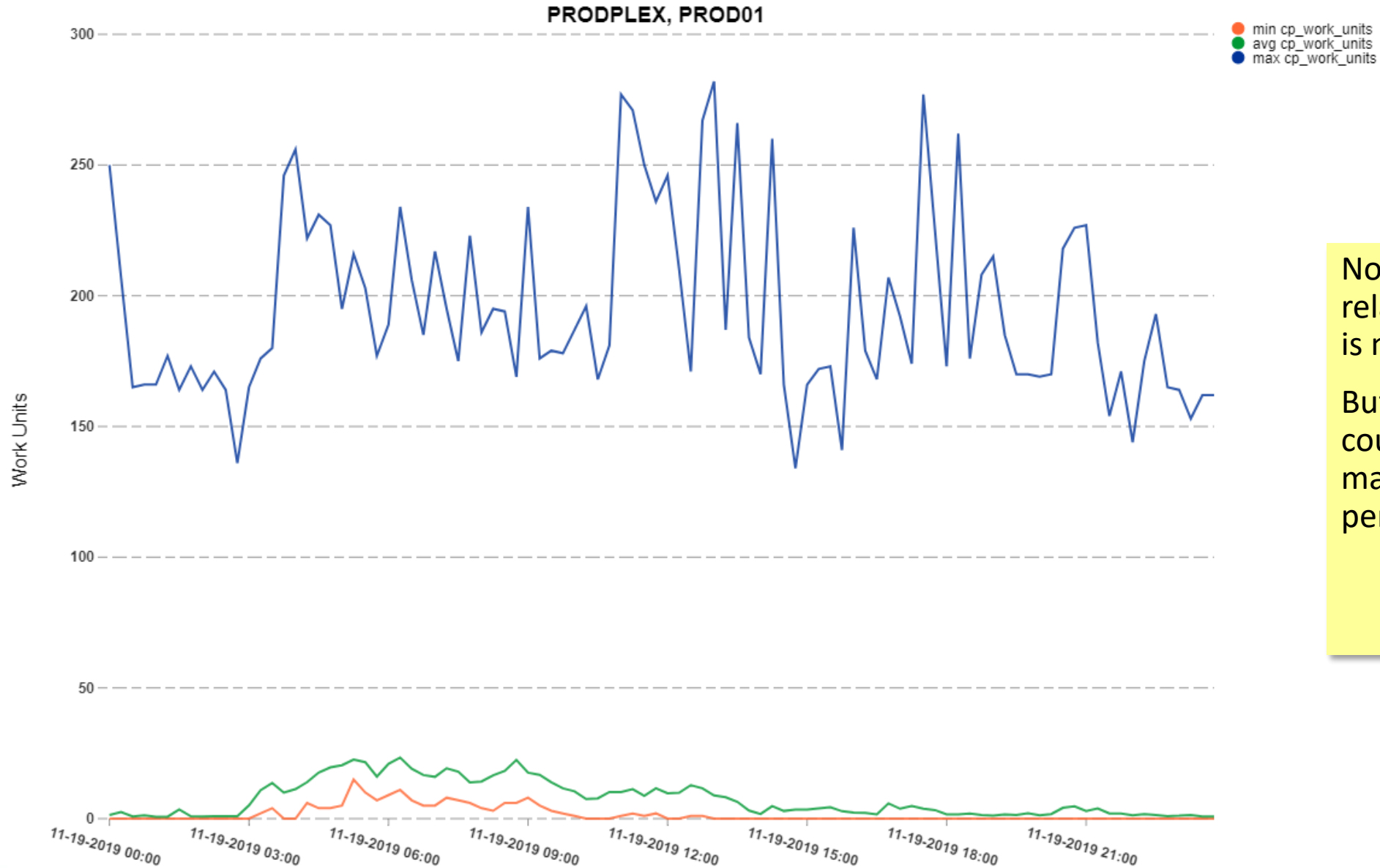
Interval: RMF (typically 900s)

Scope: LPAR

Meaning: This is the number of work units running or waiting on CPU.

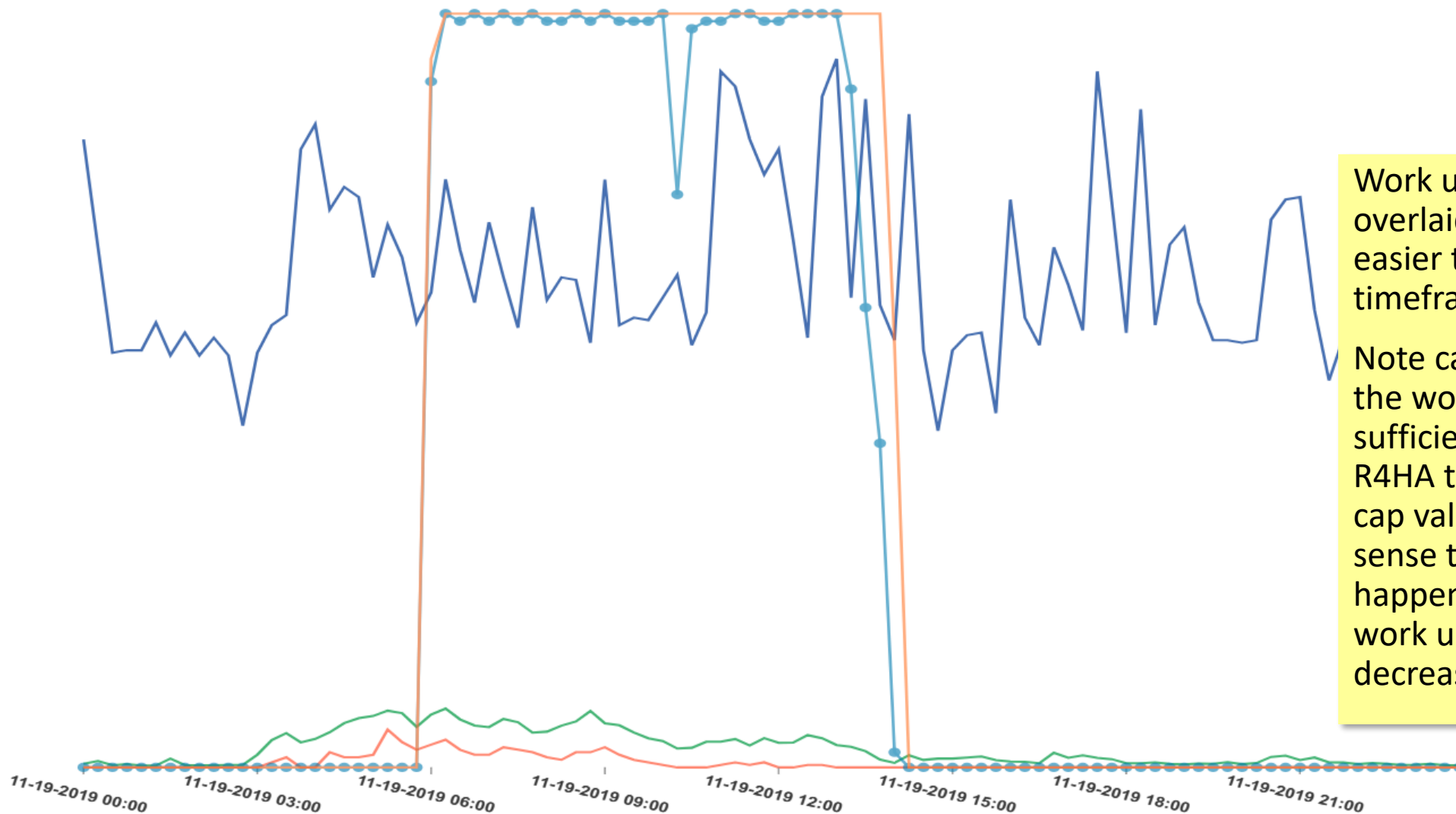
Look for: Very high maximums in an interval is not necessarily significant. The average value is more indicative of potential issues. Work unit counts > the number of online & unparked CPUs means there are work units waiting to get to a CPU.

CP Work Units - Min, Avg, Max



Note high maximum relative to average. This is not unusual.

But an average workunit count of around 25 on a machine with 2 CPUs is perhaps concerning.



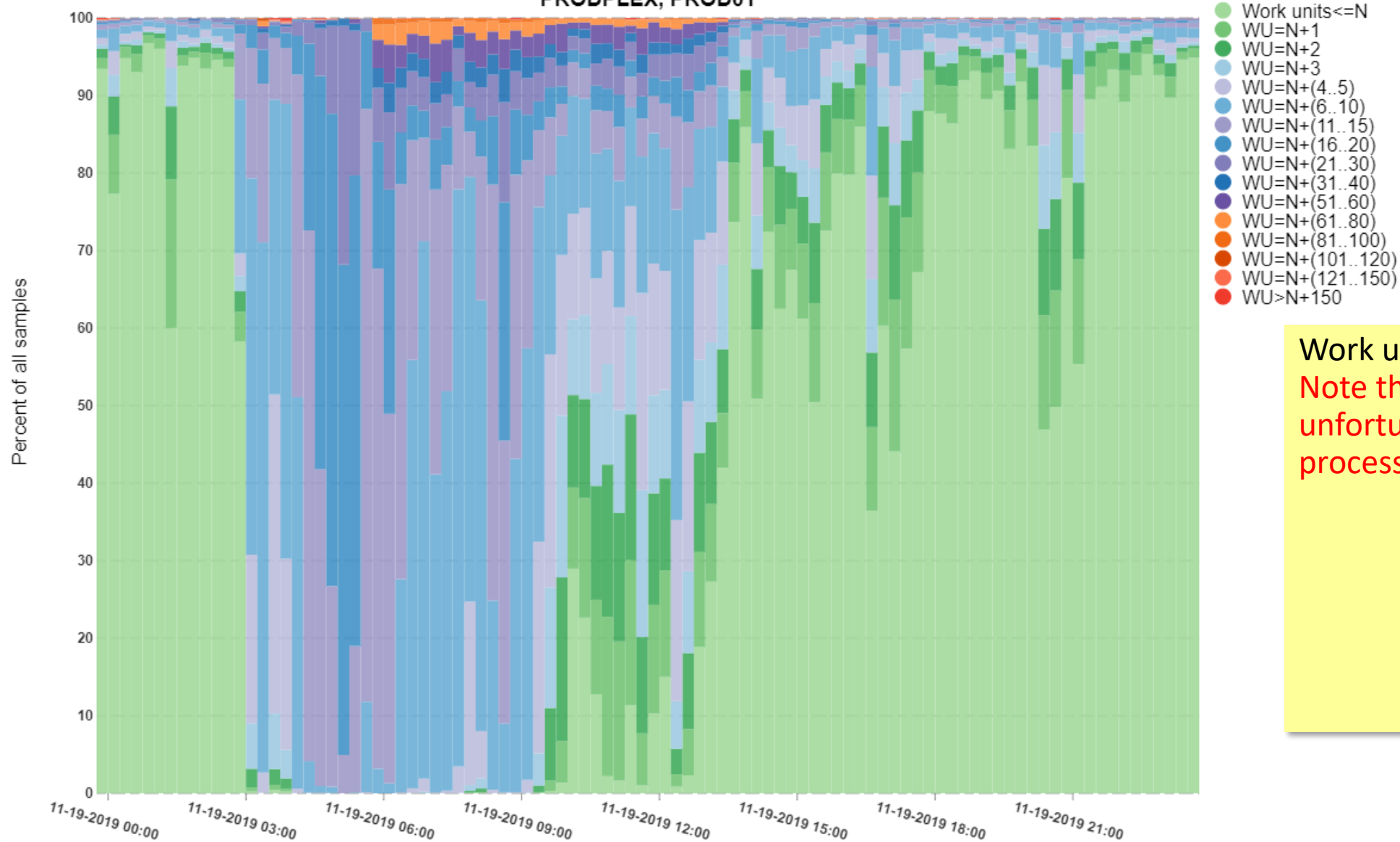
Work units and capping overlaid to make it easier to see the timeframe.

Note caps come off after the work decreases sufficiently to allow the R4HA to come below the cap value, so it makes sense that would happen after/as the work unit counts decrease.

CPU Work Unit Distribution

(N = Average of Unparked Engines Regardless of Engine Type)

PRODPLEX, PROD01

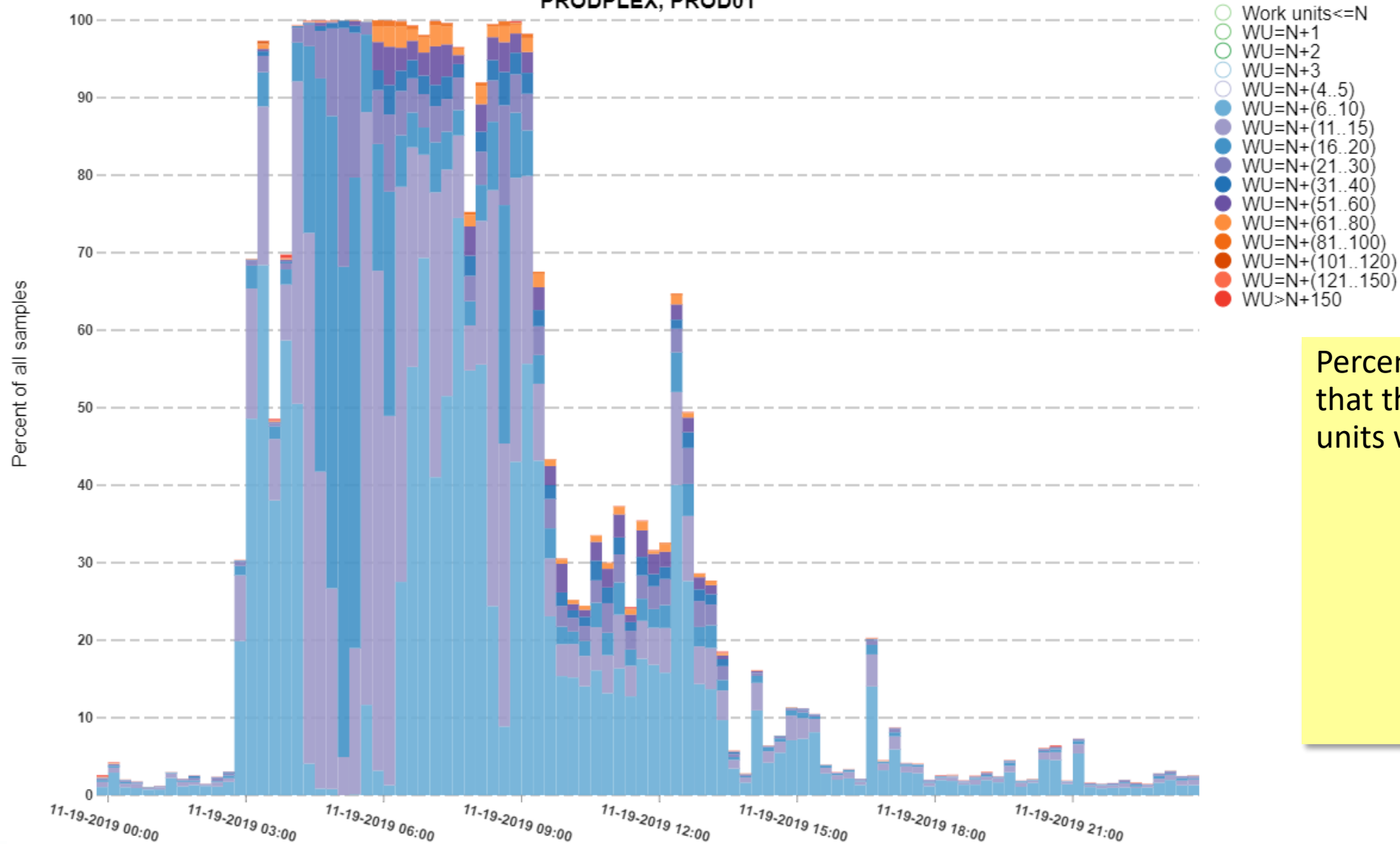


Work unit distribution.
Note that this is unfortunately across all processor types.

CPU Work Unit Distribution

(N = Average of Unparked Engines Regardless of Engine Type)

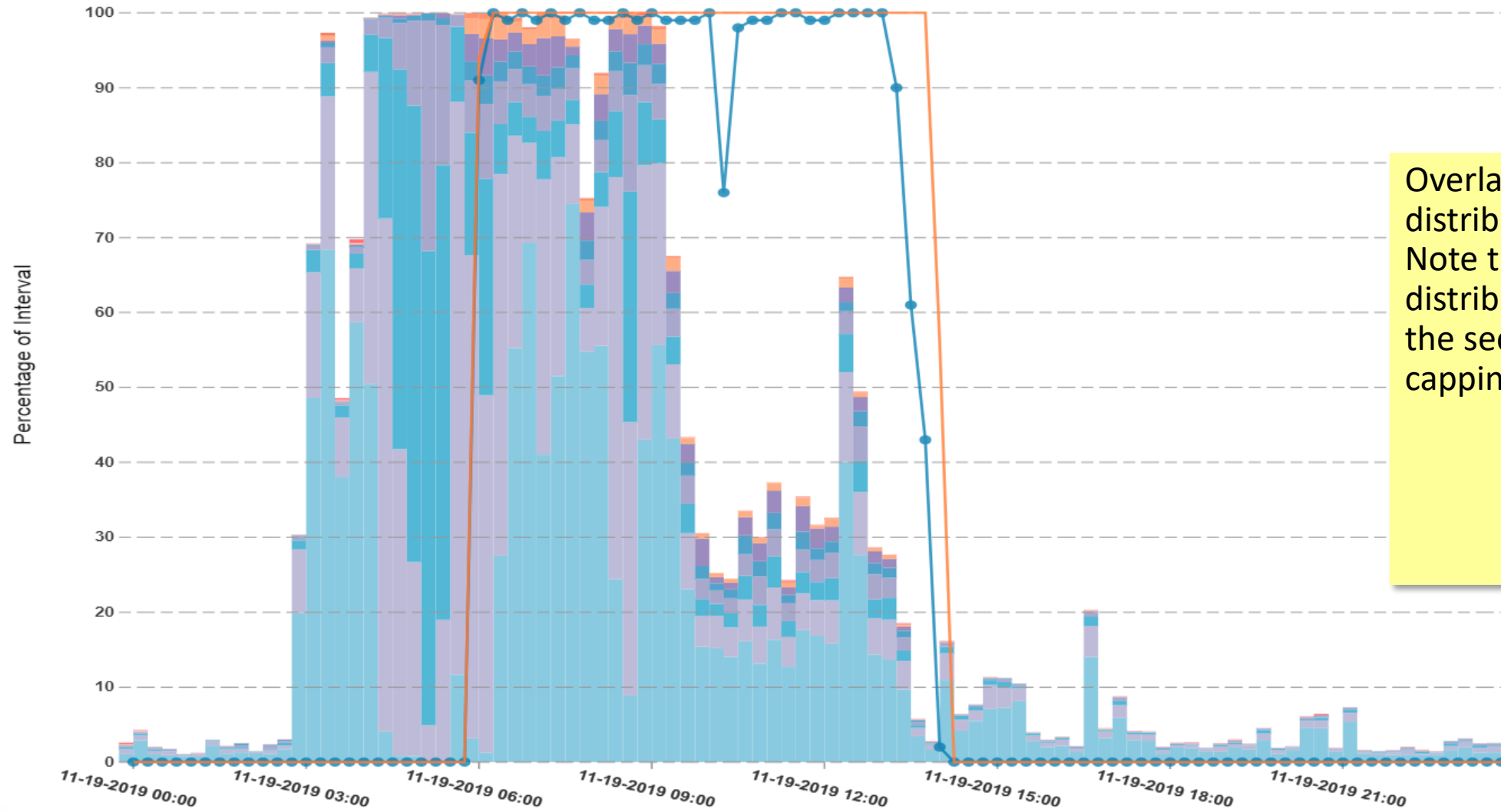
PRODPLEX, PROD01



Percent of the interval that there were >5 work units waiting.

Opacity
 Titles
 Legend
 Grid
 Chart

CPU Work Unit Distribution
 CEC Capping Actually Limited Percentage



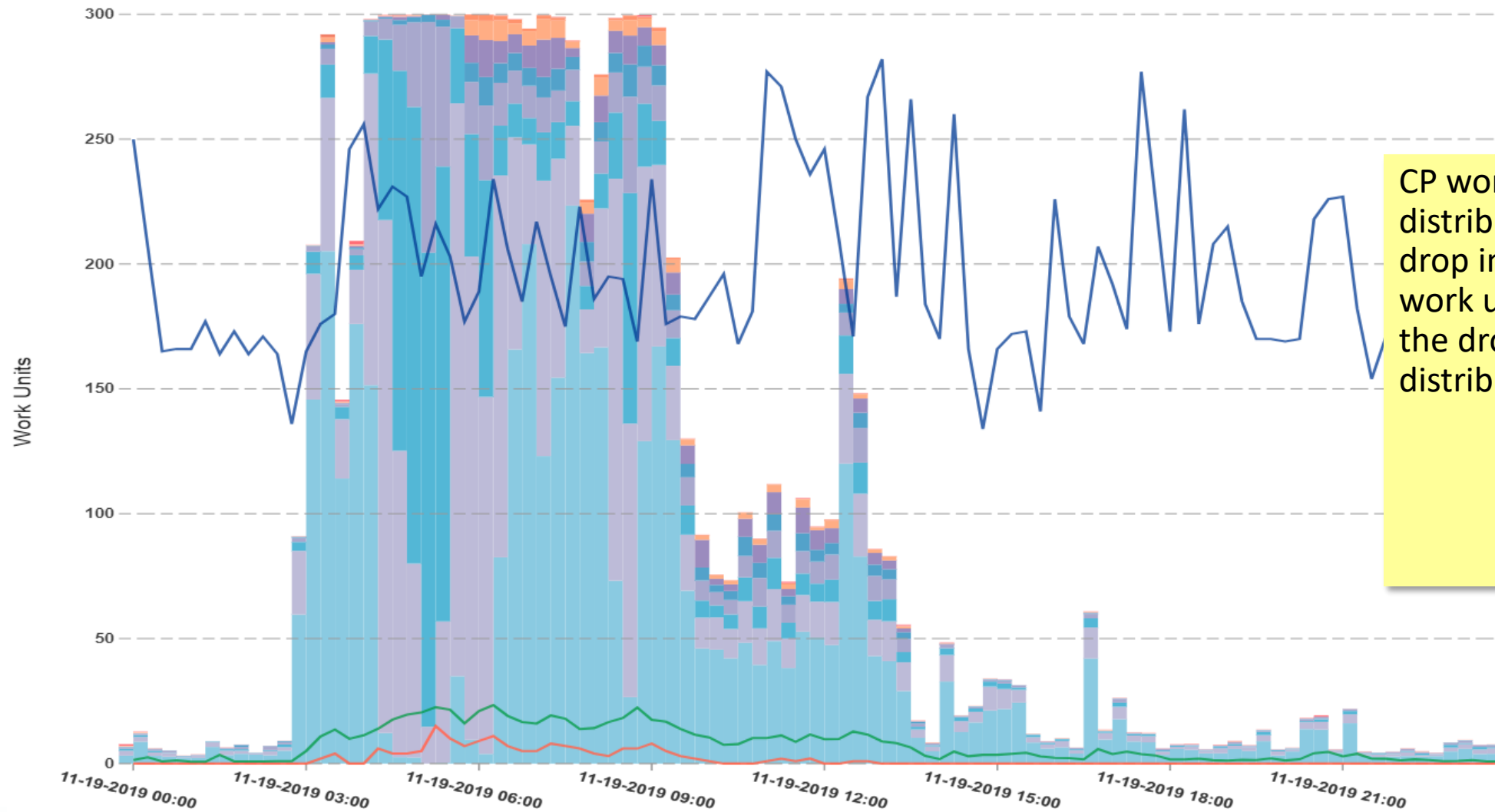
Overlay of the queue distribution vs. capping. Note that the queue distribution got better in the second half of the capping interval.



Opacity Titles Legend Grid Chart

☐ ☐ ☐ CPU Work Unit Distribution

☑ CP Work Units - Min, Avg, Max



Percent Weight Used



Source: SMF 70

Type: Calculated (from accumulated deltas and snapshot values)

Interval: RMF (typically 900s)

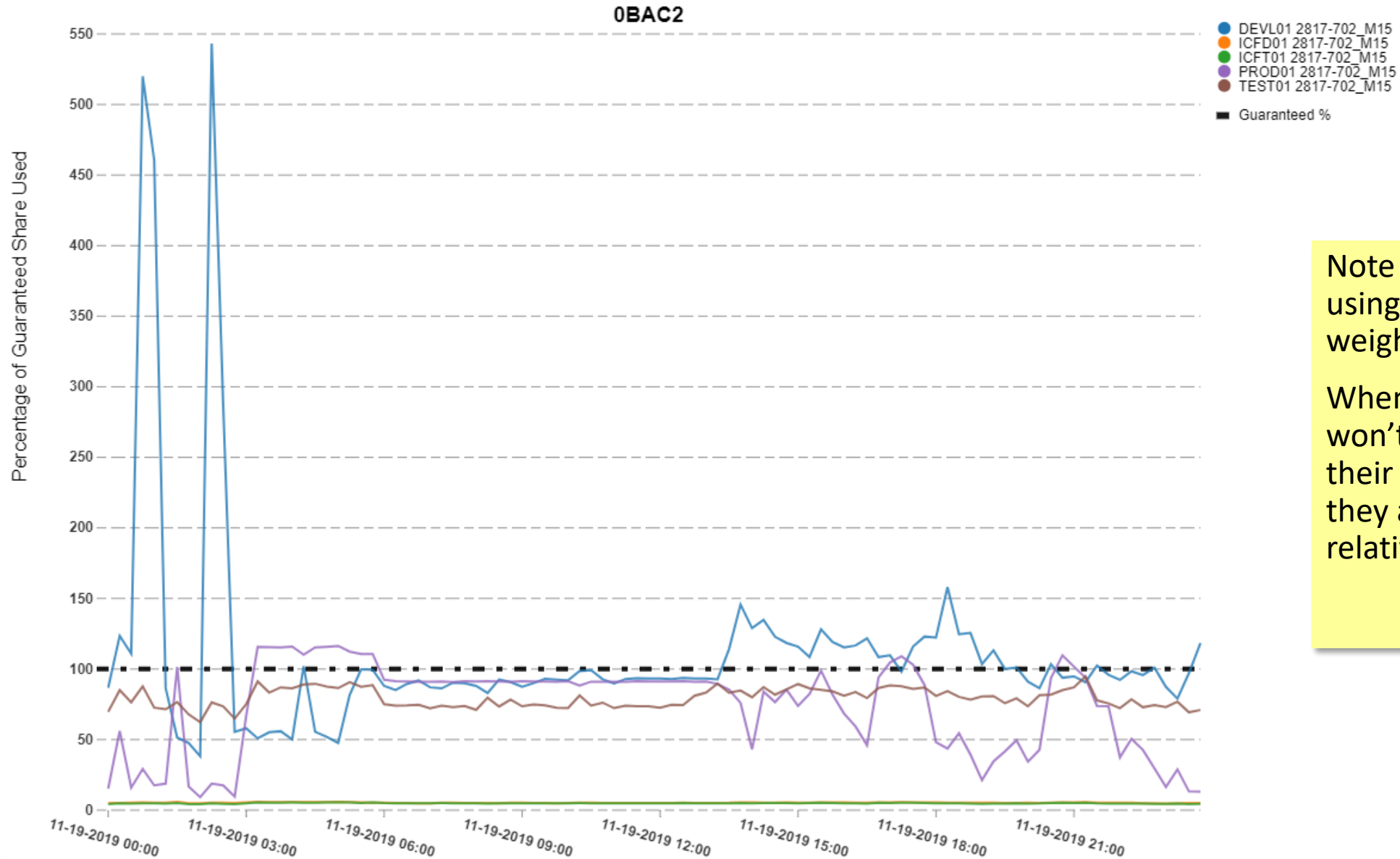
Scope: LPAR

Meaning: This is the percentage of the LPAR's weight that the LPAR consumed.

Look for: Values > 100% means that the LPAR is “borrowing” from other LPARs and may find itself restricted to its weight if the other LPARs start consuming more capacity. If your important LPARs are regularly running more than 100% of their weight, you might want to consider if they need more weight.

Optimizing weights can result in more efficient system operation as well. If need be, consider using automation to change LPAR weights at different times for different expected workload mixes between the LPARs.

CEC Percent CP Weight Used



Note how PROD01 was using more than its weight before capping.

When capped, the LPARs won't be able to achieve their total weight, but they are impacted relative to their weights.

Application Response Time



Source: SMF 72

Type: Aggregated Events

Interval: RMF (typically 900s)

Scope: Service Class Period or Report Class

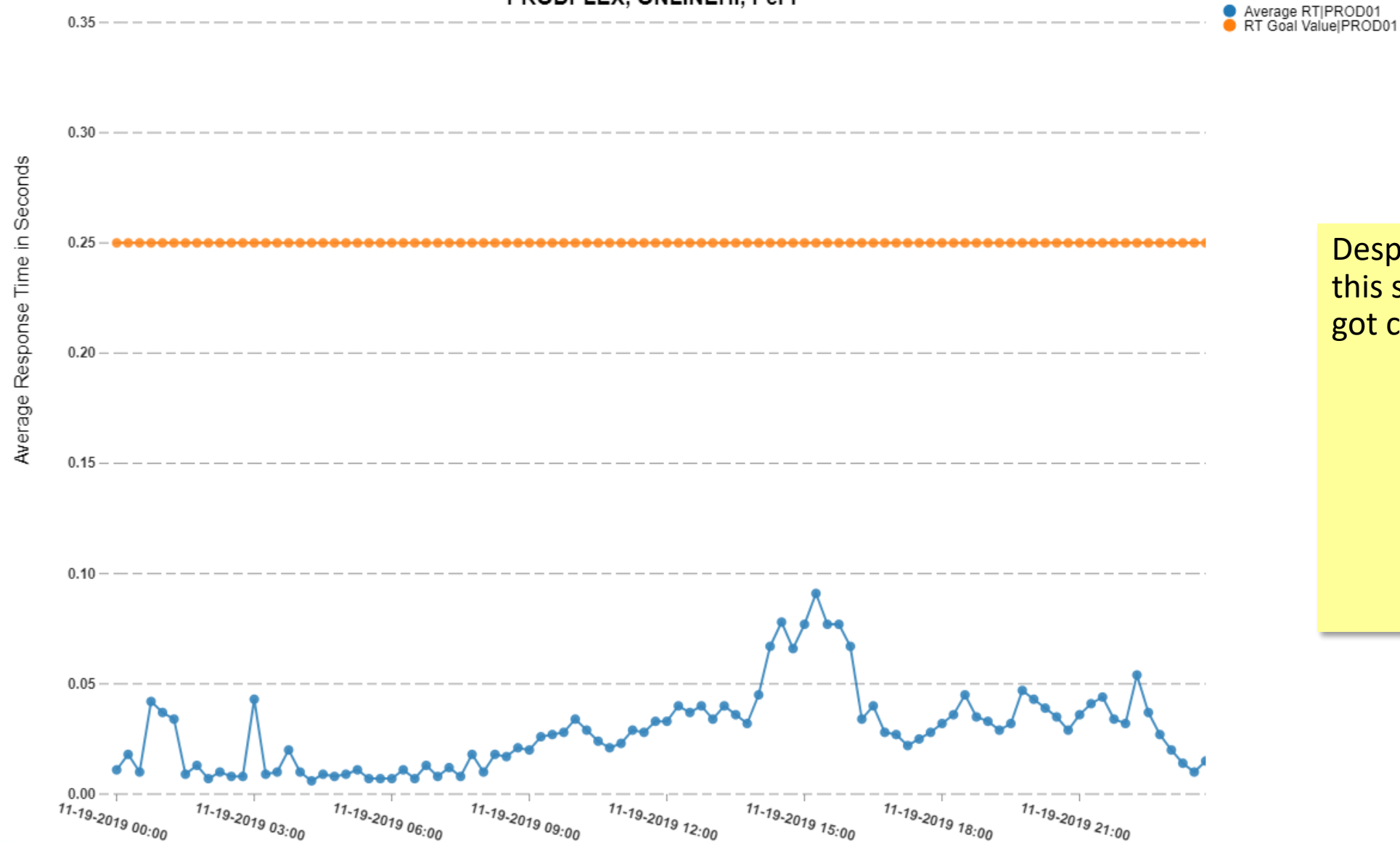
Meaning: WLM tracks transaction response time by service class and report class. With some intelligent definition of your WLM policy you can find application response time statistics. Application response time is probably the most important metric: is the business work getting done effectively?

Look for: High values relative to your SLAs.

WLM RT Goal - Average Response Time by Period

(Y-axis limited to 4 seconds)

PRODPLEX, ONLINEHI, Per1

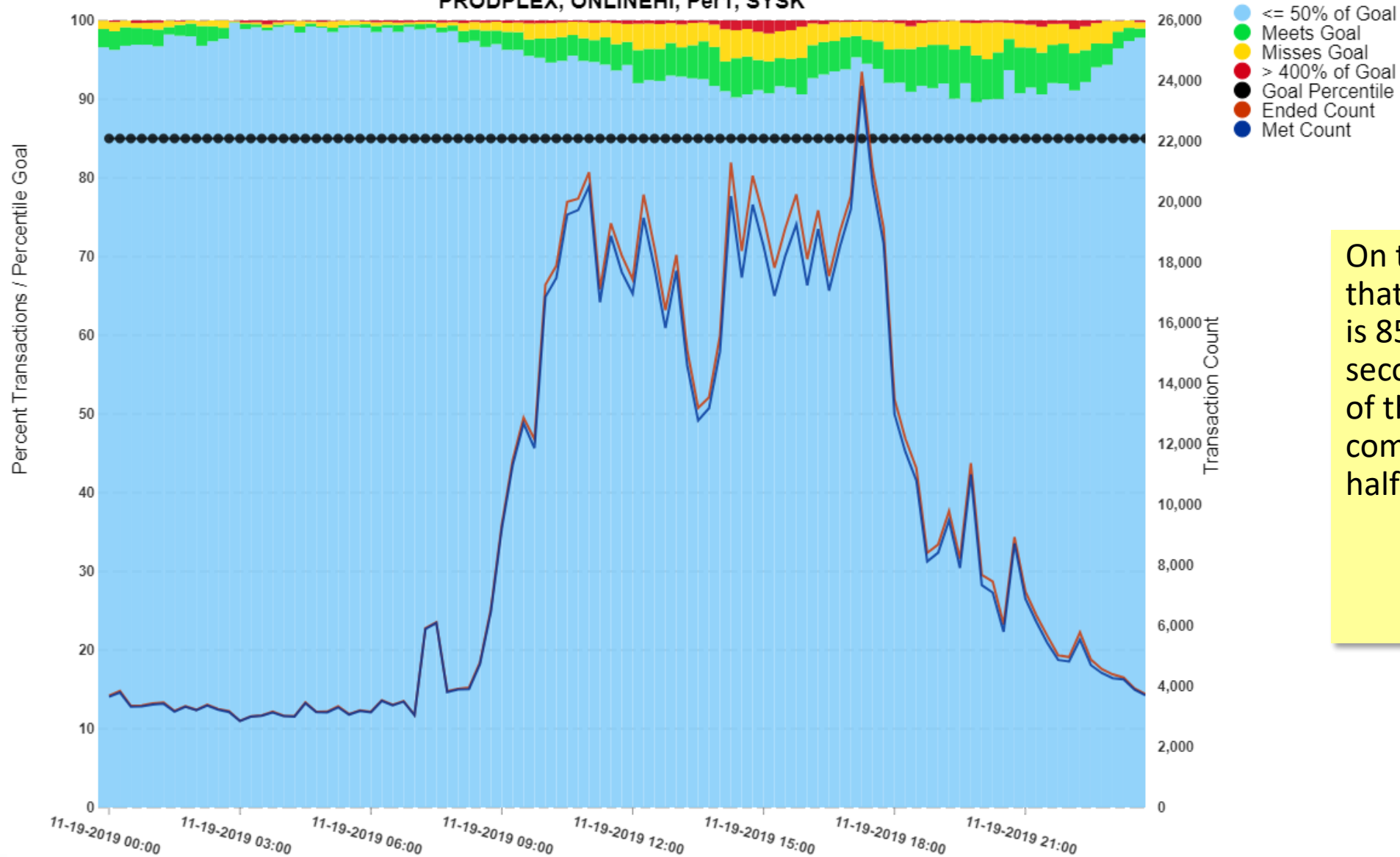


Despite being capped, this service class never got close to its goal.

WLM RT Goal - RTD% of Trans Met/Missed RT Goal with Number Trans

Percent met/missed goal and count

PRODPLEX, ONLINEHI, Per1, SYSK



On this chart we can see that the goal for that SCP is 85% less than 0.25 seconds, and over 90% of the transactions are completing in less than half that time.

I/O Response Time



Source: SMF 74

Type: Summed Events

Interval: RMF (typically 900s)

Scope: Device

Meaning: I/O response time is measured by z/OS at the device level. Typically this is reported as an average per I/O and broken down into different components.

Look for: Overall average response times of < 1ms now common, but be aware that cache misses to real spinning disks will be significantly higher!
IOSQ time should be near zero in pretty much all cases. Significant IOSQ indicates insufficient or mis-configured PAVs.

I/O Response Time Components

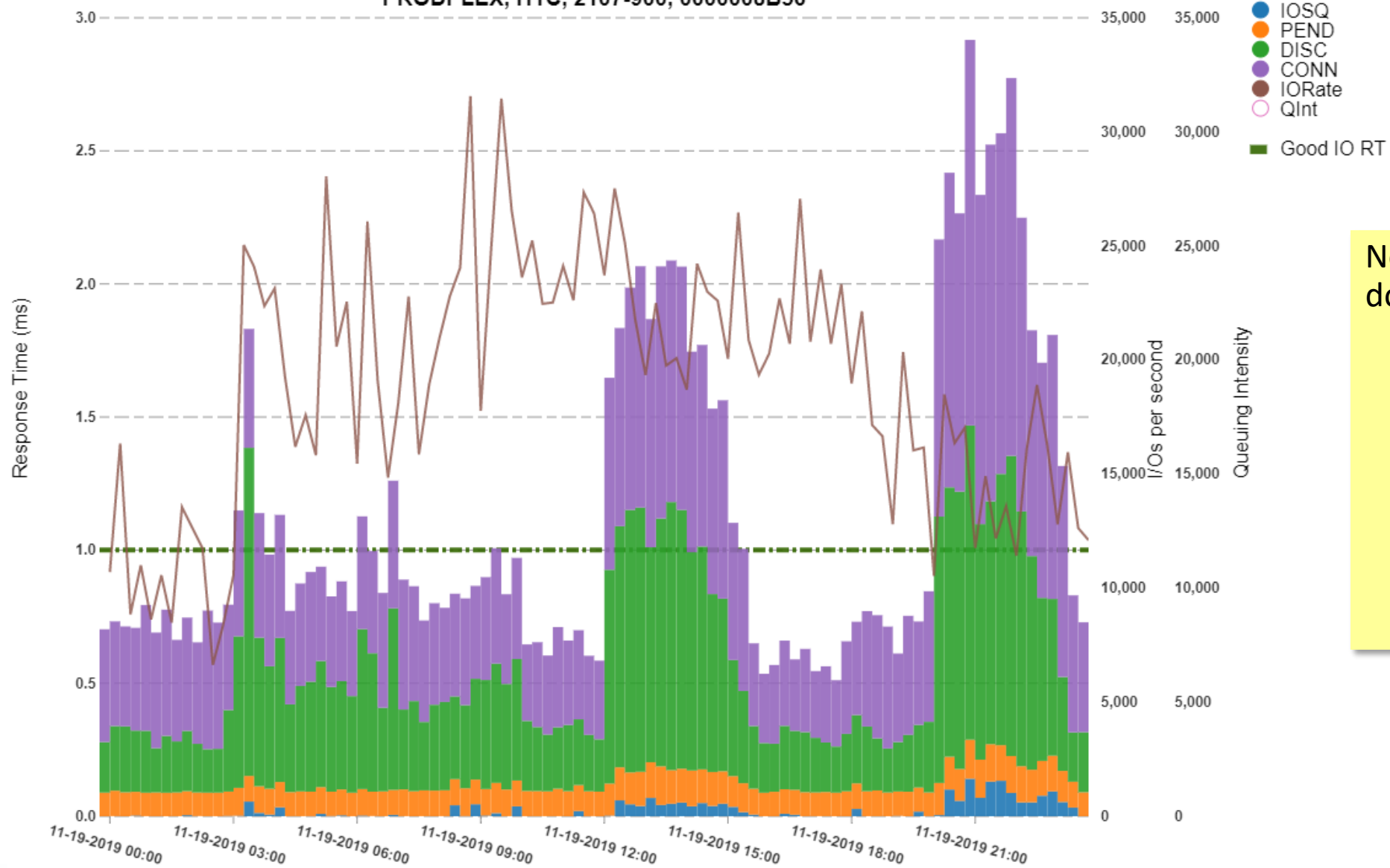


- IOSQ = I/O Supervisor Queue Time
 - Waiting on a UCB inside of z/OS
 - Virtually eliminated with PAV: especially HyperPAV and SuperPAV
- Pend = Pending, waiting for something outside z/OS UCB
 - Channel, FICON Director, Control Unit, Logical Volume in CU
- Connect = Connected transfer time
 - Time it takes to transfer the data (for the most part)
- Disconnect = Channel disconnected, I/O in progress
 - Usually due to waiting for data due to a controller cache miss

PCU RT Components

Including I/O rate, Queuing Intensity

PRODPLEX, HTC, 2107-900, 0000008B56

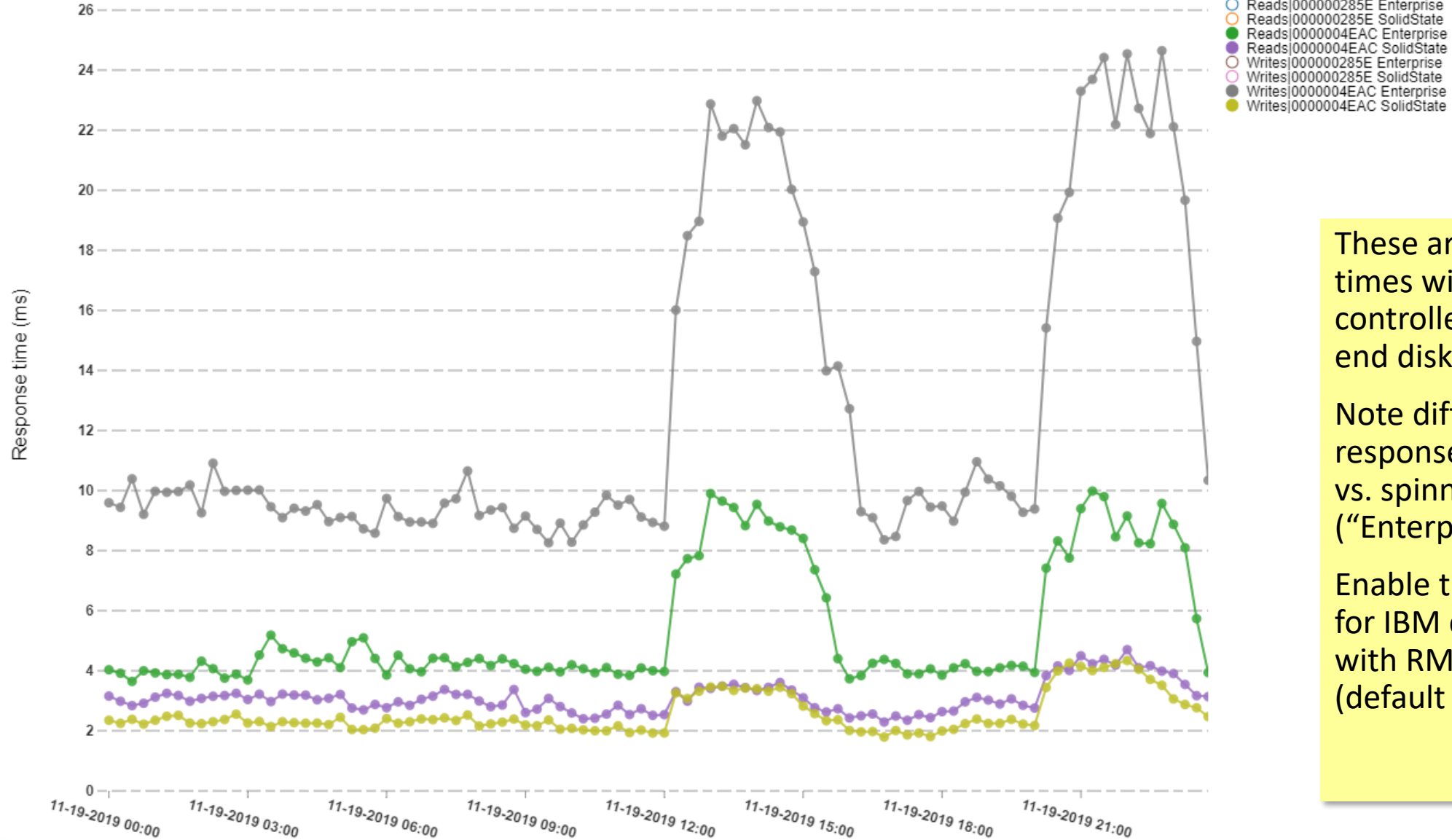


Note I/O response time does seem high at times.

PCU Rank SSD vs HDD Response Time

(Back-end I/O)

PRODPLEX



These are response times within the controller to the back-end disks.

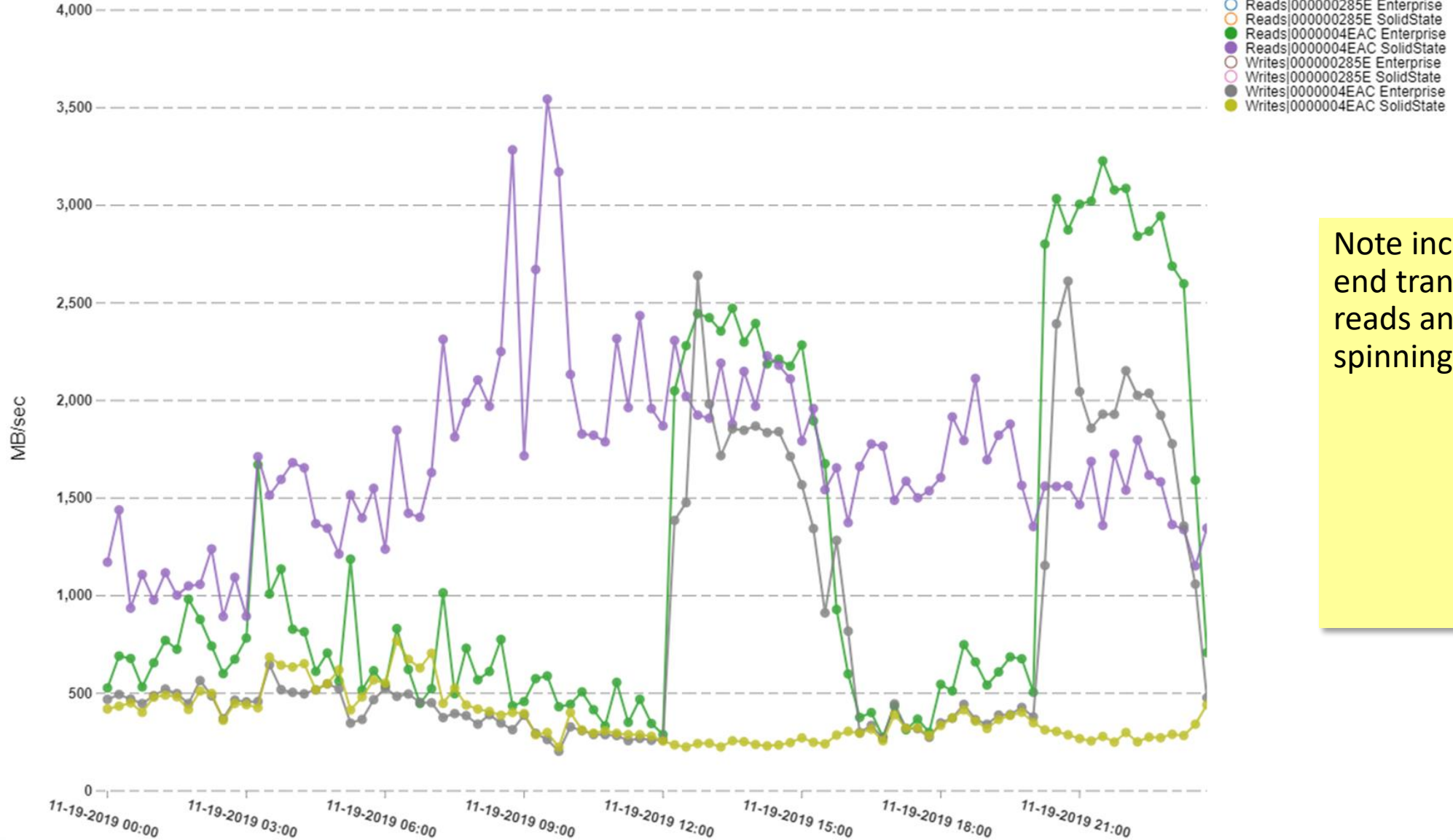
Note difference in response times for SSDs vs. spinning (“Enterprise”) disks.

Enable these statistics for IBM control units with RMF option ESS (default is NOESS).

PCU Rank SSD vs HDD I/O MB/Sec

(Back-end I/O)

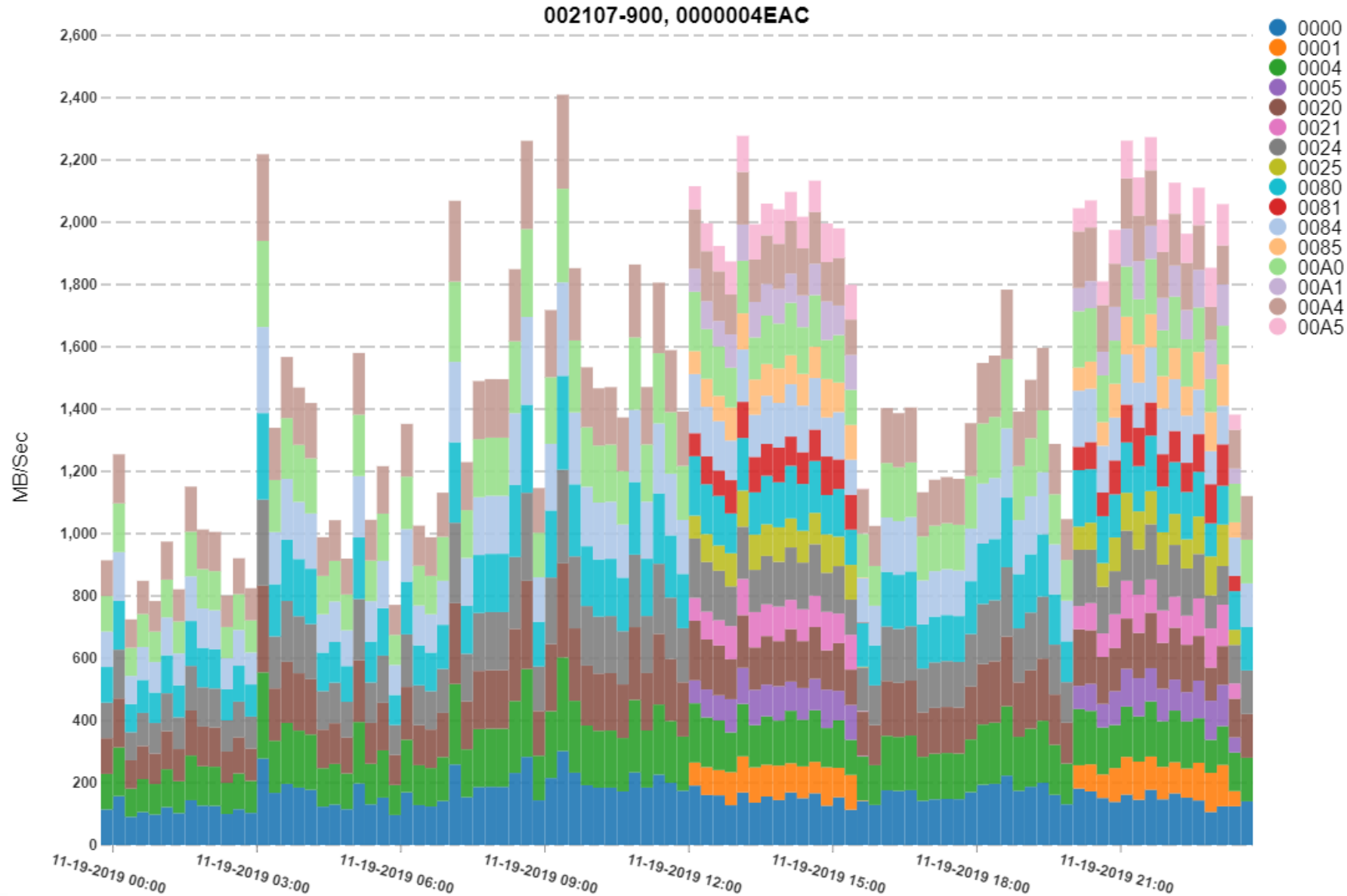
PRODPLEX



Note increase in back-end transfers for both reads and writes for the spinning disks.

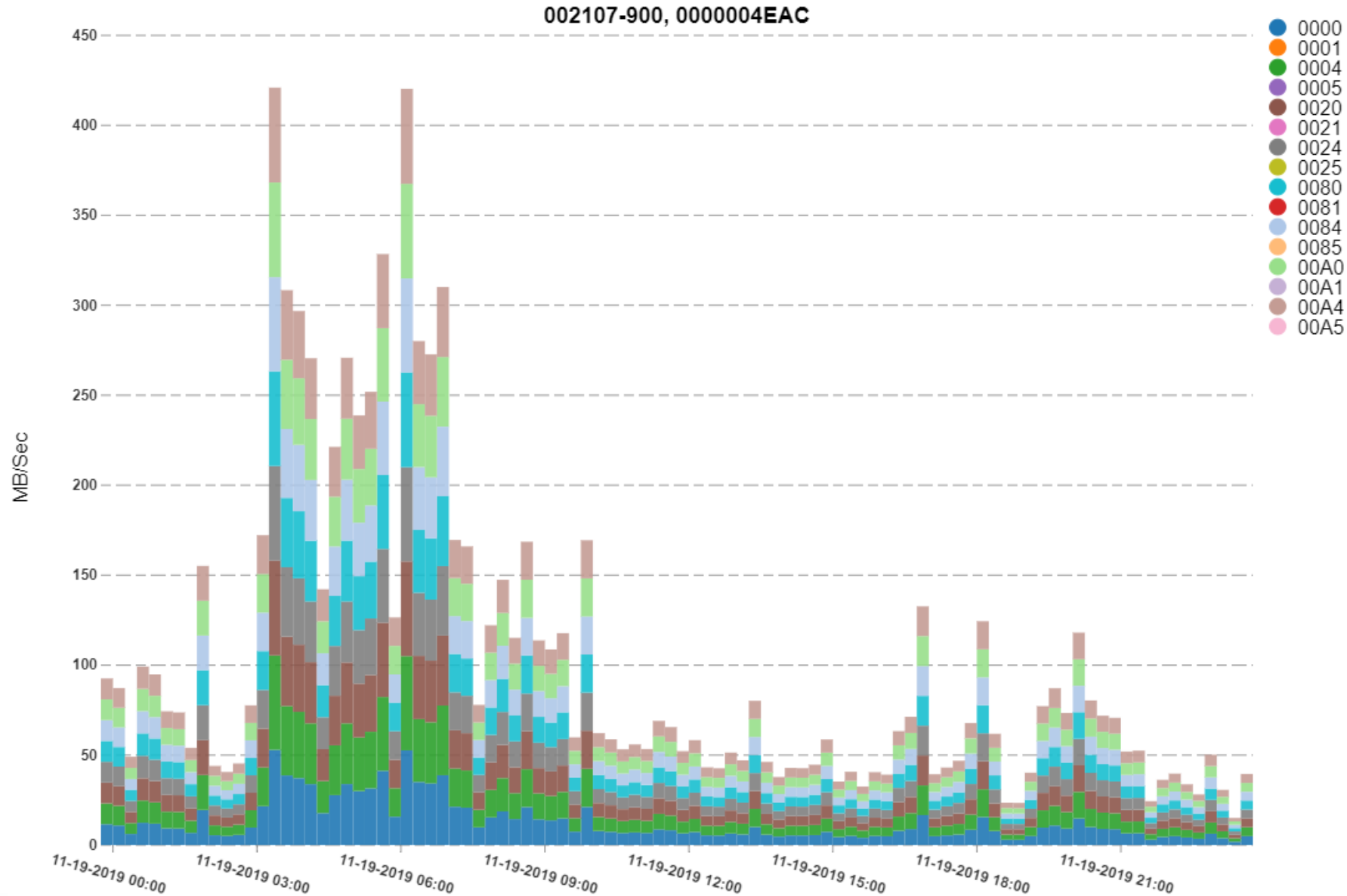


ECKD (FICON) Read MB/sec by port



The FICON channels show increase in read activity that correspond to the problem intervals.

ECKD (FICON) Write MB/sec by port



Host write activity increased some too. In absolute terms, much less than read activity.

Theory: snapshots taken and then backed up.



Wrap-up

Important Things to Remember



- SMF & RMF intervals of 15 minutes good compromise
- Record the 98s and 99s
- Understand the sources and the meaning of your data
- Being able to correlate measurements across time can be a valuable part of performance analysis

Questions?