z/OS I/O Performance: Do You Have a Problem?

Scott Chapman

Enterprise Performance Strategies, Inc.

Scott.chapman@EPStrategies.com

TechChannel Webinar
October 15, 2025

Contact, Copyright, and Trademarks

Questions?

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

Copyright Notice:

Trademarks:

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

The following are trademarks of Enterprise Performance Strategies, Inc.: **Health Check®, Reductions®, Pivotor®**

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries: IBM®, z/OS®, zSeries®, WebSphere®, CICS®, DB2®, S390®, WebSphere Application Server®, and many others.

Other trademarks and registered trademarks may exist in this presentation

No rebroadcast

For the avoidance of doubt, your attendance at this webinar ***does not*** constitute permission to record and rebroadcast this webinar without prior express written permission of Enterprise Performance Strategies, Inc.

Abstract (why you're here!)

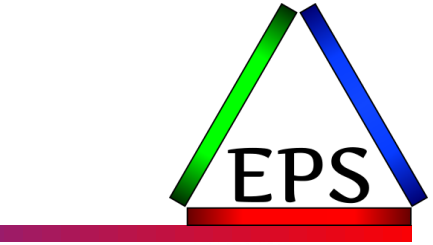
Here we are living in the future, where I/O is blazingly fast, at least compared to back in the day when we were waiting for platters to spin. And we have lots of memory so we can define large buffers and avoid a lot of I/O. And we have zHyperLink that can make disk I/O perform like coupling facility requests. So nobody has an I/O problem anymore. Right? Well… that depends. You may think your I/O response times are fine but there may be I/Os that are not. Conversely, there may be a lot of I/O that really doesn't impact application performance. And what sort of I/O response times should you expect anyways? According to which measurements?

In this session Scott Chapman will explain some of the limitations of z/OS I/O measurements today, the difficulties of identifying what really matters to application performance, and ideas for how to better investigate and reason about I/O performance.

Agenda

- Introduction
- Do we have a problem?
- Understanding the measurements
- What do we care about?
- Can we find hidden problems?
- I/O Reduction

EPS™: We do z/OS performance…



- Pivotor - Reporting and analysis software and services
  - Not just reporting, but analysis-based reporting based on our expertise
- Education and instruction
  - We have taught our z/OS performance workshops all over the world
- Consulting
  - Performance war rooms: concentrated, highly productive group discussions and analysis
- Information
  - We present around the world and participate in online forums
    https://www.pivotor.com/content.html
    https://www.pivotor.com/webinar.html

Like what you see?

- The z/OS Performance Graphs you see here come from Pivotor

- If you don't see them in your performance reporting tool, or you just want a free cursory performance 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

- We also have a free Pivotor offering available as well
  - 1 System, SMF 70-72 only, 7 Day retention
  - That still encompasses over 100 reports!

Like what you hear today?

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

- If you want a free cursory review of your environment, let us know!
  - We're always happy to process a day's worth of data and show you the results
  - See also: http://pivotor.com/cursoryReview.html

z/OS Performance workshops available

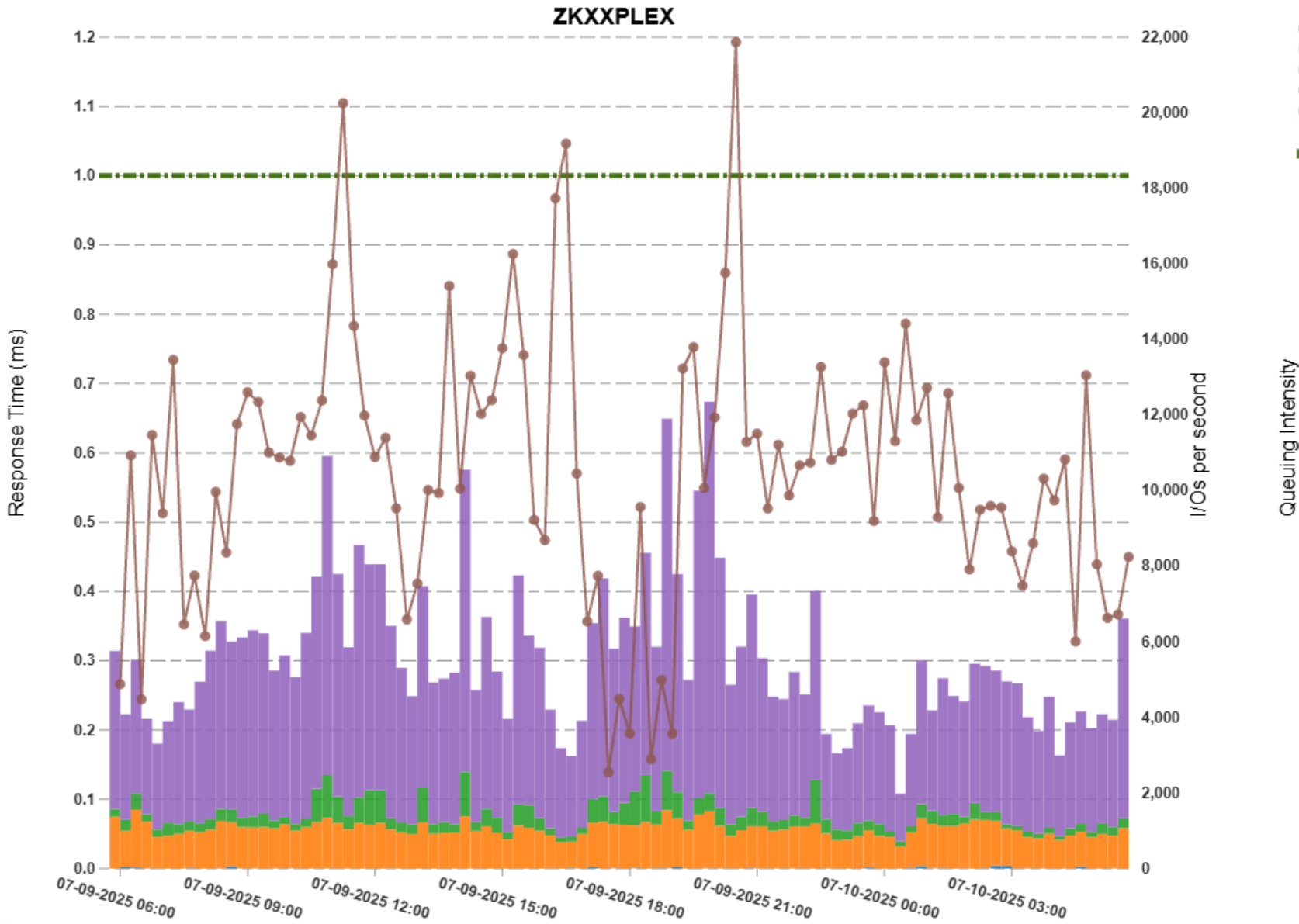
During these workshops you will be analyzing your own data!

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

Do we have a problem?

DASDplex RT Components

Including I/O rate

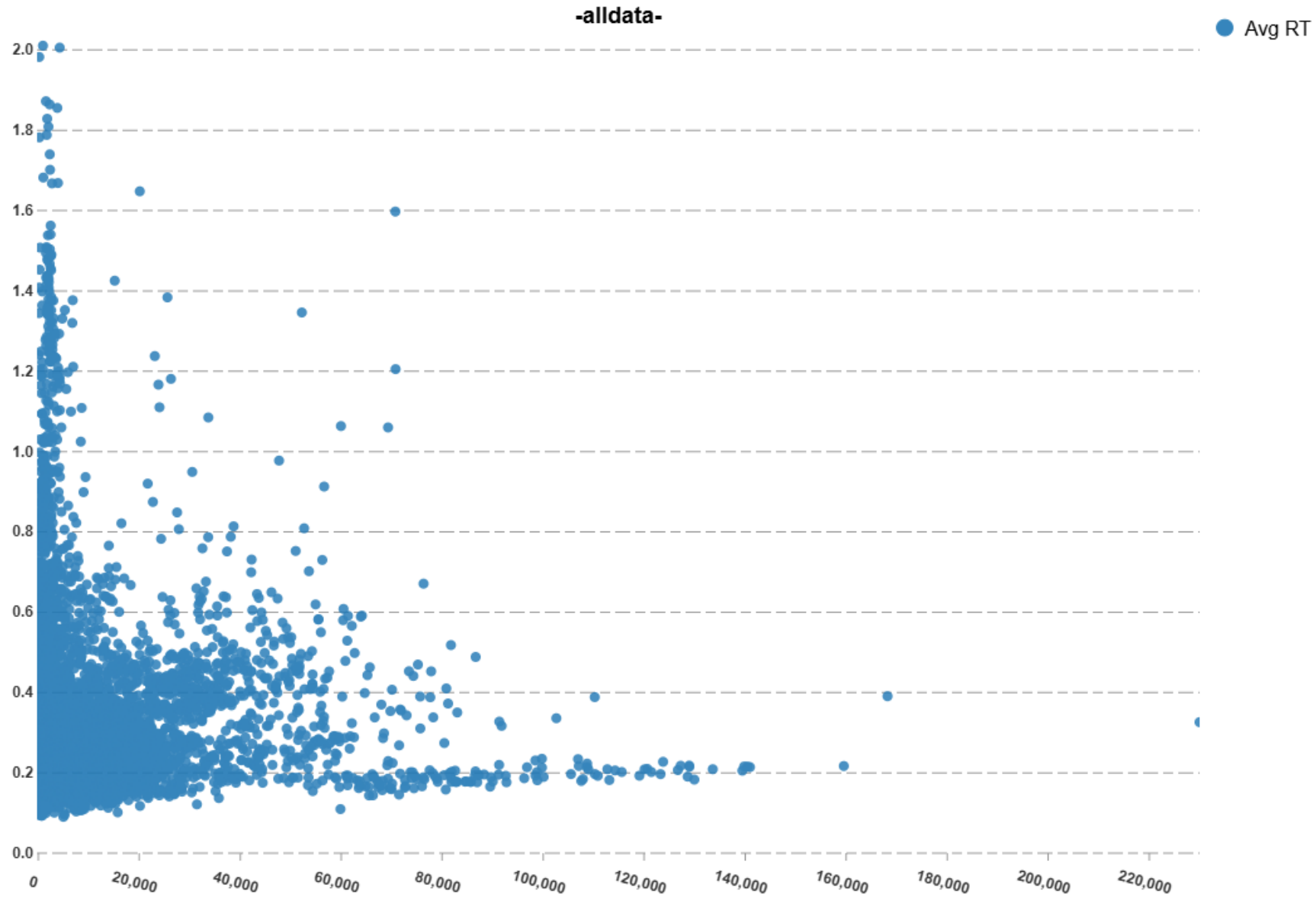


Very common to see systems like this: 10-20,000 IOPS with average response time mostly under 0.5ms.

There is a variability of course but appears to be doing quite well.

DASD Avg RT vs Activity Rate

PCU Intervals with >100 IOPS



System average DASD I/O Response Time for a day of data from scores of systems.

Vast majority of intervals are <0.5ms average RT.

So… everything is good?

- Today's average RTs are quite good
  - I remember average RTs ~10ms
- Most DASD subsystems use all-flash storage now
  - Which has different issues than spinning disk, but is generally much faster
- SuperPAV has eliminated vast majority of IOSQ time
  - IOSQ time = wait for a UCB in z/OS
  - May still sometimes see some tiny IOSQ due to various replication things
- In short, we're living in the future and we don't have the I/O performance problems of the past

So… why zHyperLink?

- zHyperLink introduced back in 2017 to improve response time for I/O satisfied out of the DS8K cache
  - Targets ~20 microsecond (0.02 milliseconds) response time
  - Processor “spins” waiting for response, much like CF sync request
  - Various technical requirements, initially not write, only recently for >4K I/Os
- Potential benefits:
  - Even faster I/O
  - CPU optimization because don’t have to redispatch after I/O
- Potential cost:
  - CPU overhead due to “spinning” waiting for the I/O to complete
- We’re starting to see more customers experiment with zHL
  - Long time coming for multiple reasons including hardware lifecycle timelines

So… should you zHyperLink?

- Maybe… do you have a I/O problems?
  - If you think so, can you articulate the business cost of that problem?
  - If you don't or can't: why do you want to complicate your life?
- Can you solve your problems with memory?
  - Memory access on the order of 15-100x faster than zHL
  - You may have memory on your LPARs/CEC that you're not using
  - You can put more memory on a CEC than on a DS8K
    - 512GB = Max DS8A10 cache = Min z17 ME1 orderable memory

Max DS8K is ~4.5 TB
Max z17 is 64 TB

- If you can't… are your problematic I/Os zHL eligible?
  - Must be 4K prior to latest DS8K (gen 10)
  - Various details around replication
- Uniquely addressable with zHL: Db2 log write bottlenecks

Four problems I believe we have

Measurements can be confusing

We don’t know what I/Os we care about

Some I/O problems are hidden by aggregation

I/O reduction efforts are undervalued

Understanding the measurements

You can’t manage what you can’t (or don’t) measure

What do we care about measuring?

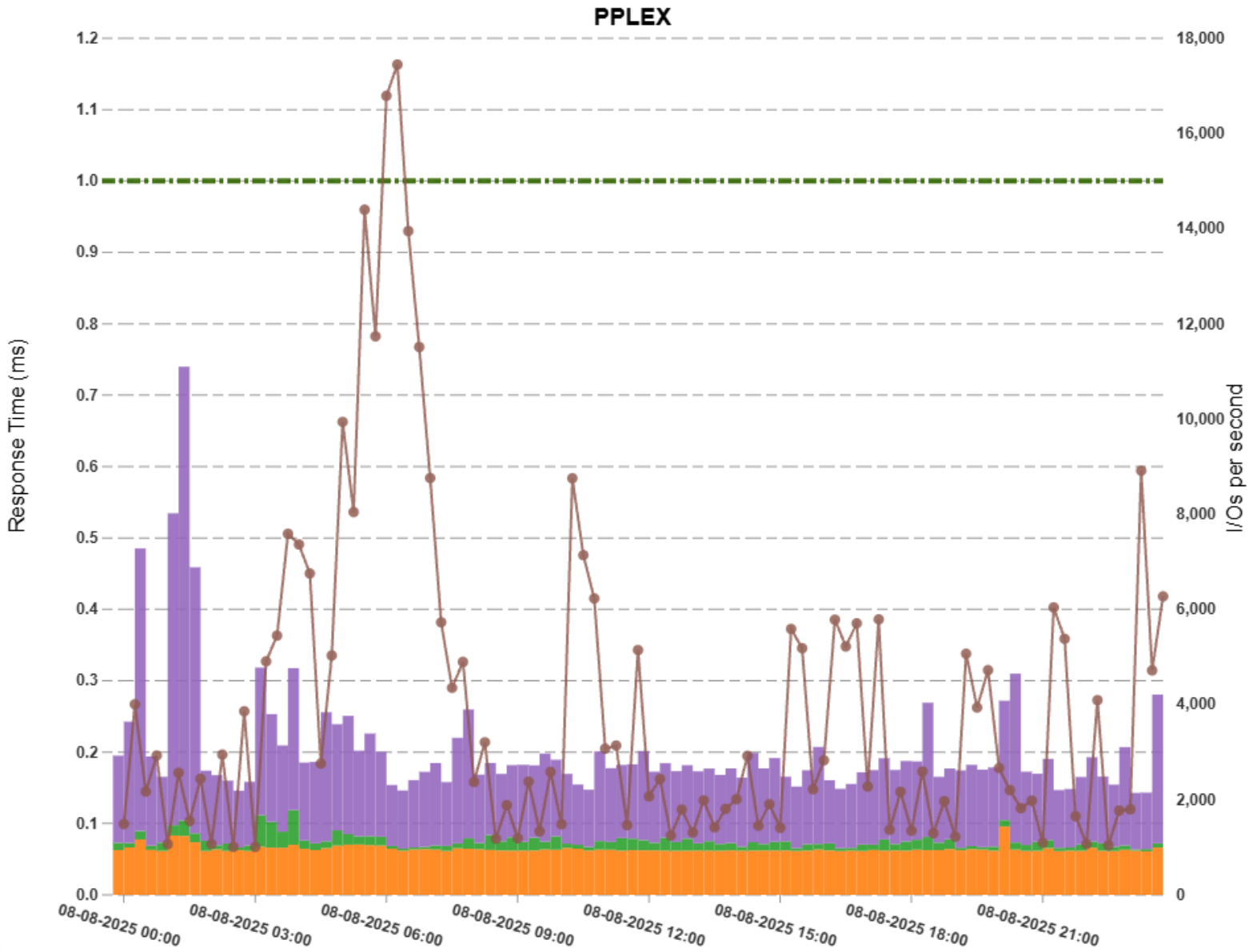
- In the z/OS world we generally care about two things:
  - How many I/Os are being done? (Usually IOPS – I/Os per second)
  - How long do they take? (Usually an average response time, but may be total time)
- Note that there's generally less emphasis on MB/sec rates
- There's of course plenty of other measurements as well
  - Help explain the above measurements or diagnose problems
- Today we expect to see average response times to be < 0.5ms
  - Under 0.2ms is not terribly unusual
- I/O rates vary widely based on lots of things
  - <10,000 IOPS for the sysplex is relatively low
  - >100,000 IOPS for the sysplex would be relatively high (and/or a large plex)

Response Time components

- Response Time = IOSQ + PEND + DISCONNECT + CONNECT
  - IOSQ: IO Supervisor Queue time, wait for UCB in z/OS (should be almost always 0)
  - PEND: Wait outside of z/OS (channel -> director -> control unit)
    - Command Response
    - Device Busy
  - DISCONNECT: Wait for processing data in the controller
  - CONNECT: Protocol + data transfer time
- Not (usually) included in the response time value: Interrupt delay time
  - This is delay that it takes for the interrupt to be processed by z/OS
  - On larger LPARs generally rounds to zero
  - On small LPARs with low weights and only 1 or 2 CPs, can be quite significant
    - Seen small systems with fast DASD where interrupt delay is >= RT
    - Could be a risk point for migrating to shared outsourcer system

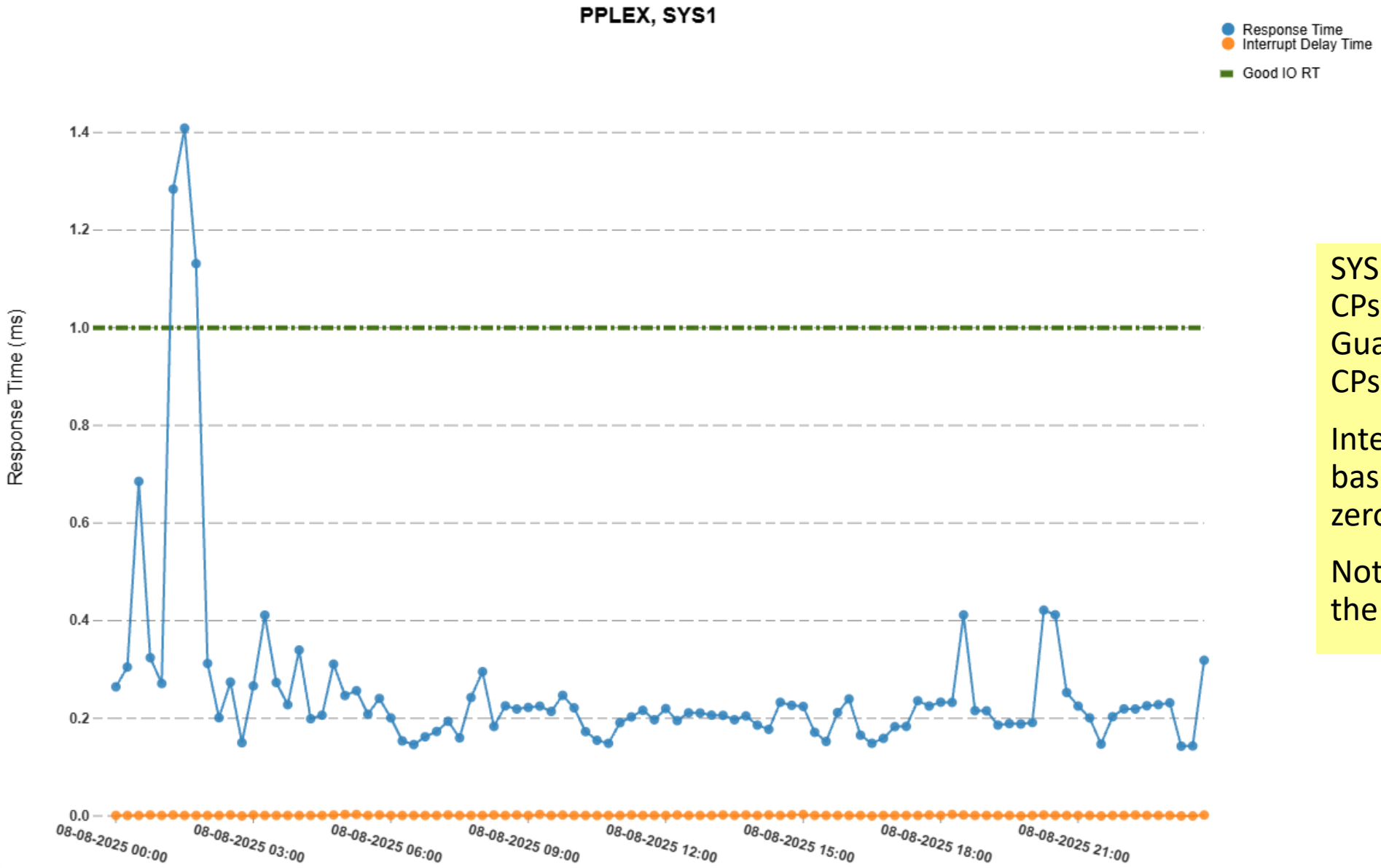
DASDplex RT Components

Including I/O rate



This all looks very uninteresting: I/O rate is generally very low to low and I/O average response time is almost always below 0.3ms. And there doesn't appear to be any significant IOS Queue time.

Average Response Time versus Interrupt Delay Time

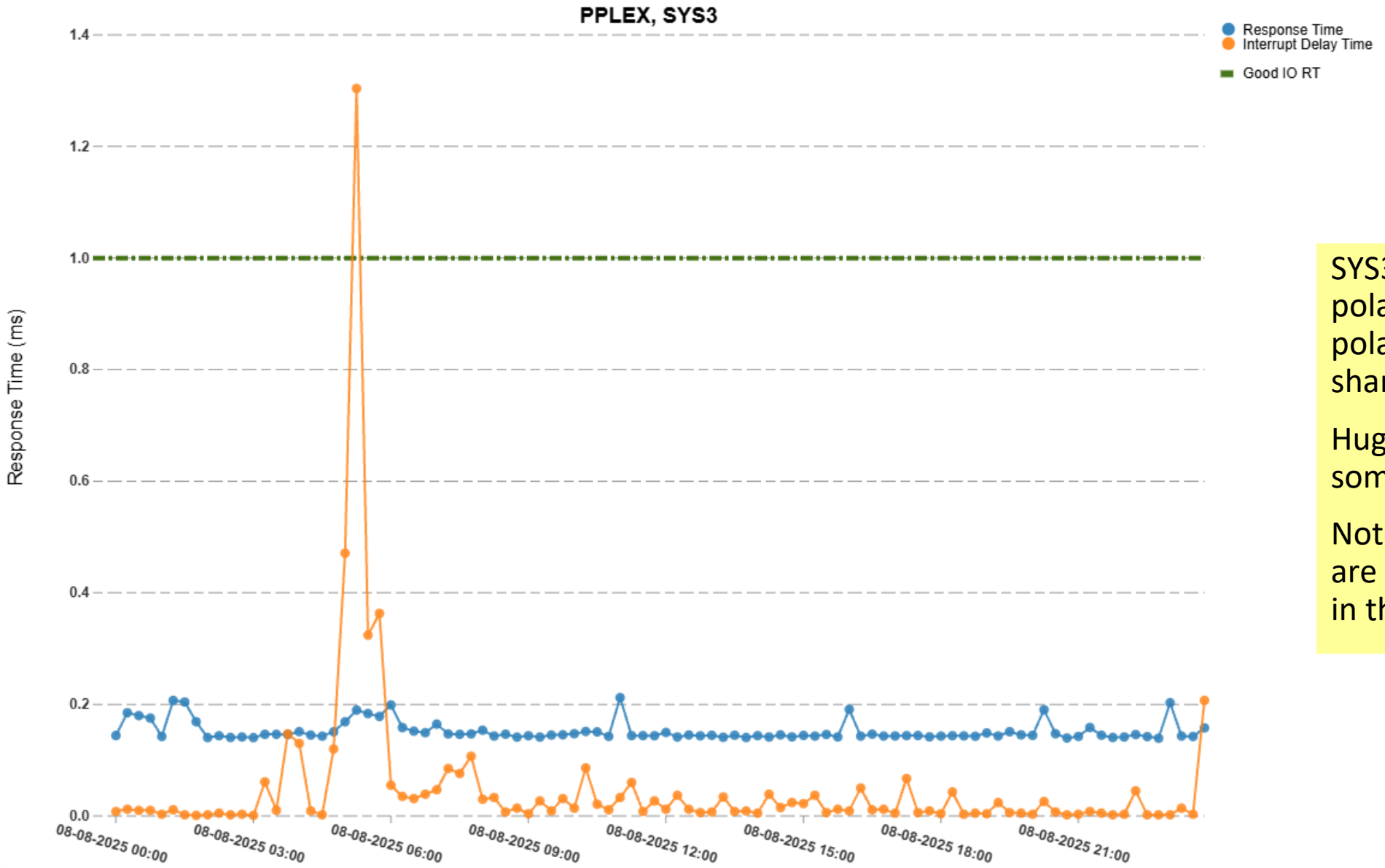


SYS1 has 2 high polarity CPs and 2 mediums. Guaranteed share is ~3.2 CPs capacity.

Interrupt delay is basically rounding to zero.

Note this comes from the SMF 74 data.

Average Response Time versus Interrupt Delay Time



SYS3 has 1 medium polarity CPs and 3 low polarity CPs. Guaranteed share is only 0.16 CPs.

Huge interrupt delay in some intervals!

Note these two systems are on the same CEC and in the same sysplex.

What to do about Interrupt Delay?

- Make sure you have 2 CPs enabled for interrupts
  - Default in z/OS 3.1
  - Prior to 3.1 may need to tune CPENABLE to try to keep 2 enabled during busy times
    - Small systems with only 2 CPs: CPENABLE=(0,0) may be appropriate
- Less likely to see interrupt delay if the LPAR has 2+ high pool CPs
  - Of course this is not always possible
  - “More/slower” CPs is almost always better than “fewer/faster” CPs
- Do less I/O
  - Less I/O = fewer interrupts to handle
  - Won’t necessarily help the situation for low weighted LPARs with limited I/O

I/O Measurements in SMF data

- I/O related information is all over in the SMF records
  - Type 14/15 – “Old” DD-related I/O
  - Type 30 – Summary I/O at job/step
  - Type 42 – Volume and dataset level I/O
  - Type 64 – VSAM Status
  - Type 72 – I/O by service class/report class
  - Type 74 – Volume level I/O details
  - Type 75 – I/O by page dataset
  - Type 100-102 – Various Db2 details, including Db2 I/O (depending on config)
  - Type 110 – CICS details, including CICS I/O
  - Others – Sort, Vendor-specific DASD measurements, etc.

Interesting Measurement Details: 74

- Most of the RMF measurements in the 74s is in “128-microsecond units”
  - These are totals though so not *necessarily* as bad as it seems
  - Long ago (z12?) there as an announcement that channel measurements would be available in 1 microsecond units: apparently, that’s what’s being totaled
- 74.1 measurements are activity as seen by the LPAR
  - Includes RT breakdown but no details on how many were reads or writes
- 74.5 measurements are activity as seen by the control unit itself
  - All systems will report “same” measurements (so some only collect on 1 system)
  - Contains cache hit/miss and read vs write details
  - Contains details (inc. RT) for “back-end” storage for IBM control units

Interesting Measurement Details: 30

- SMF 30 has both summary at address space and DD/device levels
  - I/O section has I/O at address space level
    - Has counts for blocks and I/Os, plus some response time components
    - Some details by independent enclaves and the address space + dependent enclaves
  - EXCP section has I/O info for each combination of DD name and device address pair
    - Contains count of blocks issued (wraps at 4B) and total device connect time
  - EXCP Section impacted by options in SMFPRMxx
    - DDCONS(YES|NO) – Consolidate duplicated EXCP entries (may elongate AS shutdown)
    - NODETAIL – specify on SUBSYS to exclude EXCP sections for STCs to speed STC shutdown
    - EMPTYEXCPSEC(NOSUPPRESS|SUPPRESS) – empty EXCP sections for candidate vols

Interesting Measurement Details: 42

- SMF 42.5 has Storage Class level measurements
  - Includes combination of RTs, read/write, and cache details
  - Some data in 1µs units, some in 128µs units
- SMF 42.6 has dataset level measurements
  - Written both at interval and dataset close
    - Interval records not written if no activity
  - Has average timings in both 128-microsecond and 1-microsecond precision
    - Use the latter which are reasonably close to averages calculated from 74.1 totals
- SMF 42.15-19 has RLS measurements
  - Sysplex-wide measurements collected by one system
  - Has total times in milliseconds, but records averages RTs as integers (i.e. 0) 🤦

What I/O do we care about?

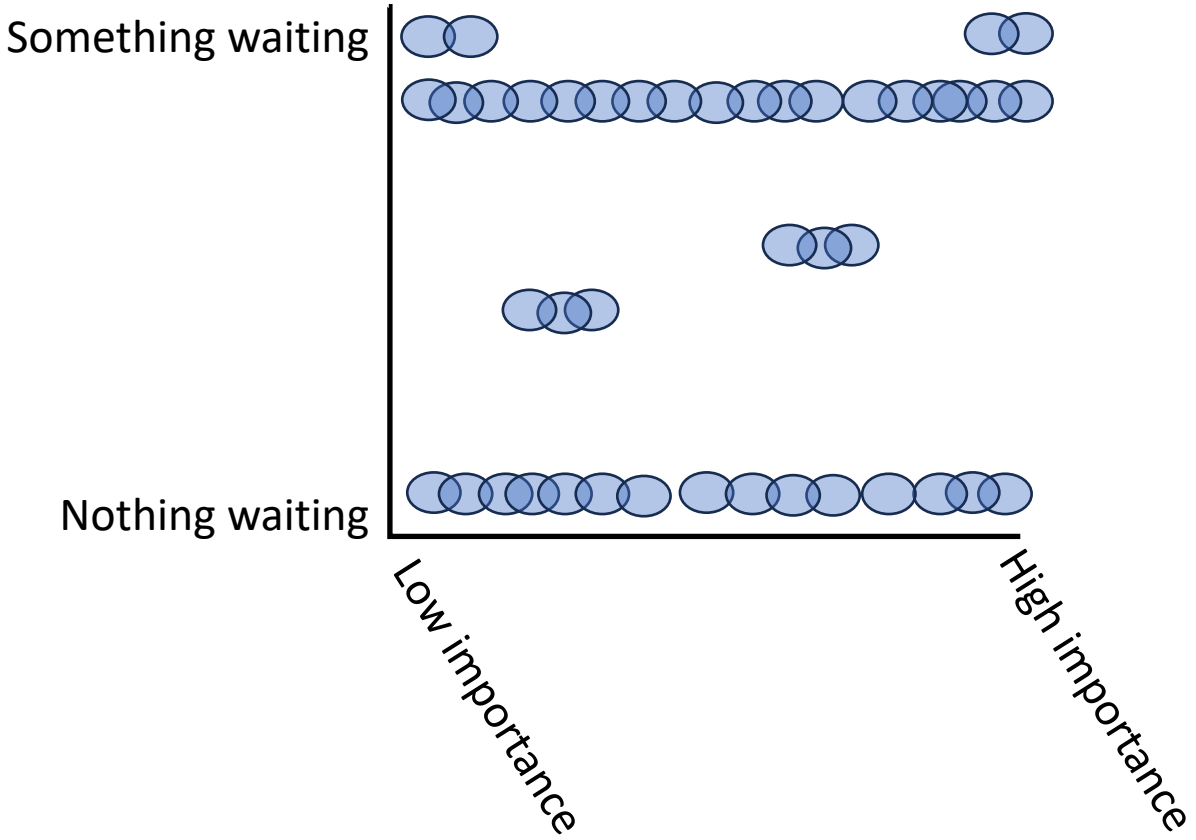
Waiting continuum

- Some I/O is issued *asynchronously* to the application processing
  - Writes are often buffered and may be flushed without the application waiting
  - Reads may be read-aheads for data that's not immediately needed
- Some I/O is *synchronous* to the application processing
  - Application can't continue until the data is read/written
- Some async I/O may become a delay if it's not completed in time
  - E.G. read-ahead I/Os that don't complete before the data is needed
- Some I/O may delay multiple things
  - May need to finish an I/O to release a lock on something larger than just that I/O

In this presentation, "Sync I/O" is an I/O that needs to complete before something can continue.
Async I/O is an I/O that does not directly impact application response time.
zHyperLink I/Os are often referred to as "Sync I/O", but that's not what we're talking about here.

I/O Significance

- I/Os do have a priority but that is mostly moot
  - Very rarely are I/Os queued anywhere such that the priority would matter
  - With I/O priority management off, I/O priority = CPU priority of the work
- I/Os also have different importances to the system and business
- Not all I/Os have something actively waiting for them to complete



The significance of the response time for a particular I/O is the combination of how important the I/O is to the business/system and whether or not something is actively waiting on that I/O completion

Significance categories & examples

- Significant: Important / Sync
  - Db2 sync I/O for a transaction a customer is waiting on
- Possibly Significant: Important / Async
  - Db2 async prefetch (may effectively become “sync” if too slow)
- Mostly Insignificant: Unimportant / Sync
  - Db2 sync I/O for batch not on the critical path
- Insignificant: Unimportant / Async
  - Reads for backups

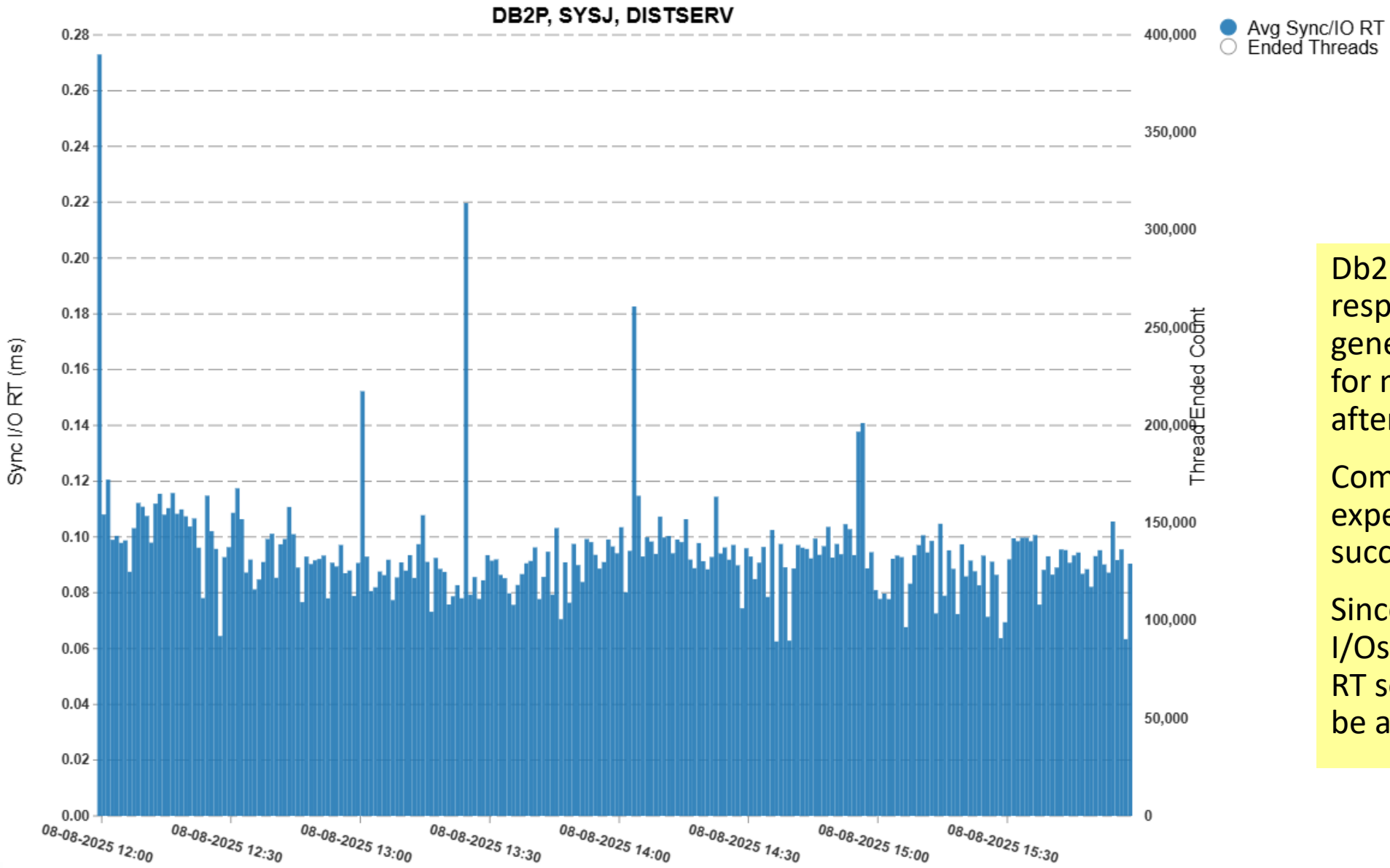
- Remember: these are really continuums!

Determining I/O significance

- Determining the significance of the I/Os is very difficult
- Volume and dataset naming conventions can determine if something is production or not
  - But not all production is of utmost importance
  - Some I/O to a given volume/dataset may be sync, others may be async
- Db2 does capture stats around sync/async I/Os
  - I haven't seen that in other subsystems
  - Possibly because they don't prefetch as aggressively / intelligently

Plans Average Sync I/O RT

Plans with non-negligible CP CPU



Db2 is recording sync I/O response times of generally under 100µs for much of the afternoon.

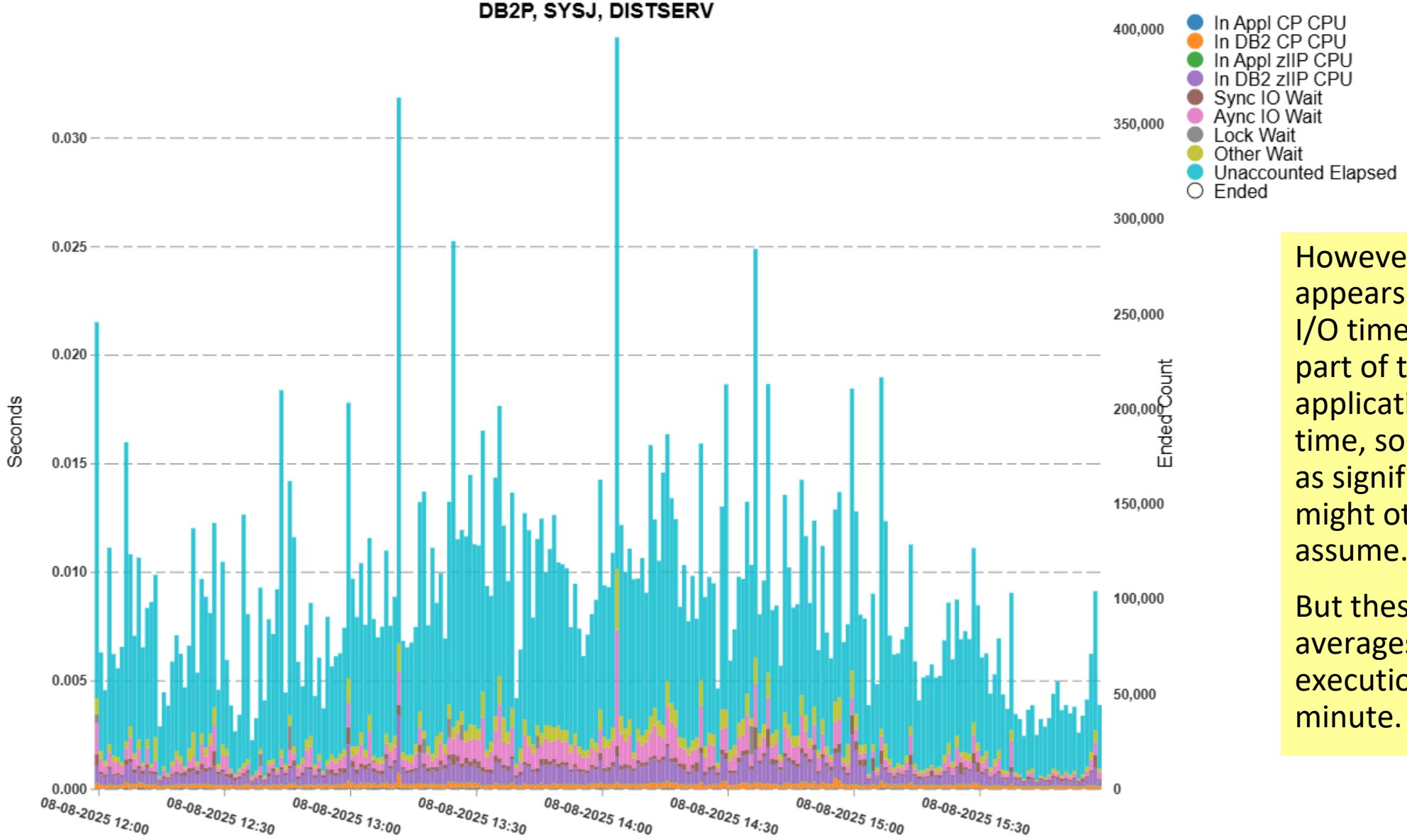
Compare to an expectation of 17µs for successful zHyperLink.

Since these are sync I/Os, saving ~80% of the RT sounds like this might be a good thing.

Plan Avg RT Component Breakdown

Plans with non-negligible CP CPU

DB2P, SYSJ, DISTSERV



However, in this case it appears that the Sync I/O time is only a small part of the overall application elapsed time, so maybe it's not as significant as we might otherwise assume.

But these are also averages across all plan executions ended in that minute.

Can we find “hidden” problems?

Or at least determine if we might have outliers?

Aggregation aggravation

- It is common for mainframe systems to have thousands of DASD volumes
- This is too much to report on in detail, so the data is typically aggregated
  - E.G. by storage group, volser prefix, system, sysplex
  - May report “top” volumes for the reporting period (e.g. day, shift, hour)
  - Usually this aggregation produces an average RT across many volumes / intervals
- And even the detail RT for a volser in an interval is an average of many I/Os
- But averages can hide outliers
  - Do we care about those outliers?
    - Probably depends on the significance of those outlier I/Os
- Hypothetically: a volser does 10 IO/sec over a 900 second interval
  - 8990 I/Os took 0.1ms, but 10 I/O took 500ms
  - That’s an average RT of 0.66ms. Would you think twice about that?
  - But if those 10 I/Os were “significant” that might have impacted something

But… I/Os don't take half a second!

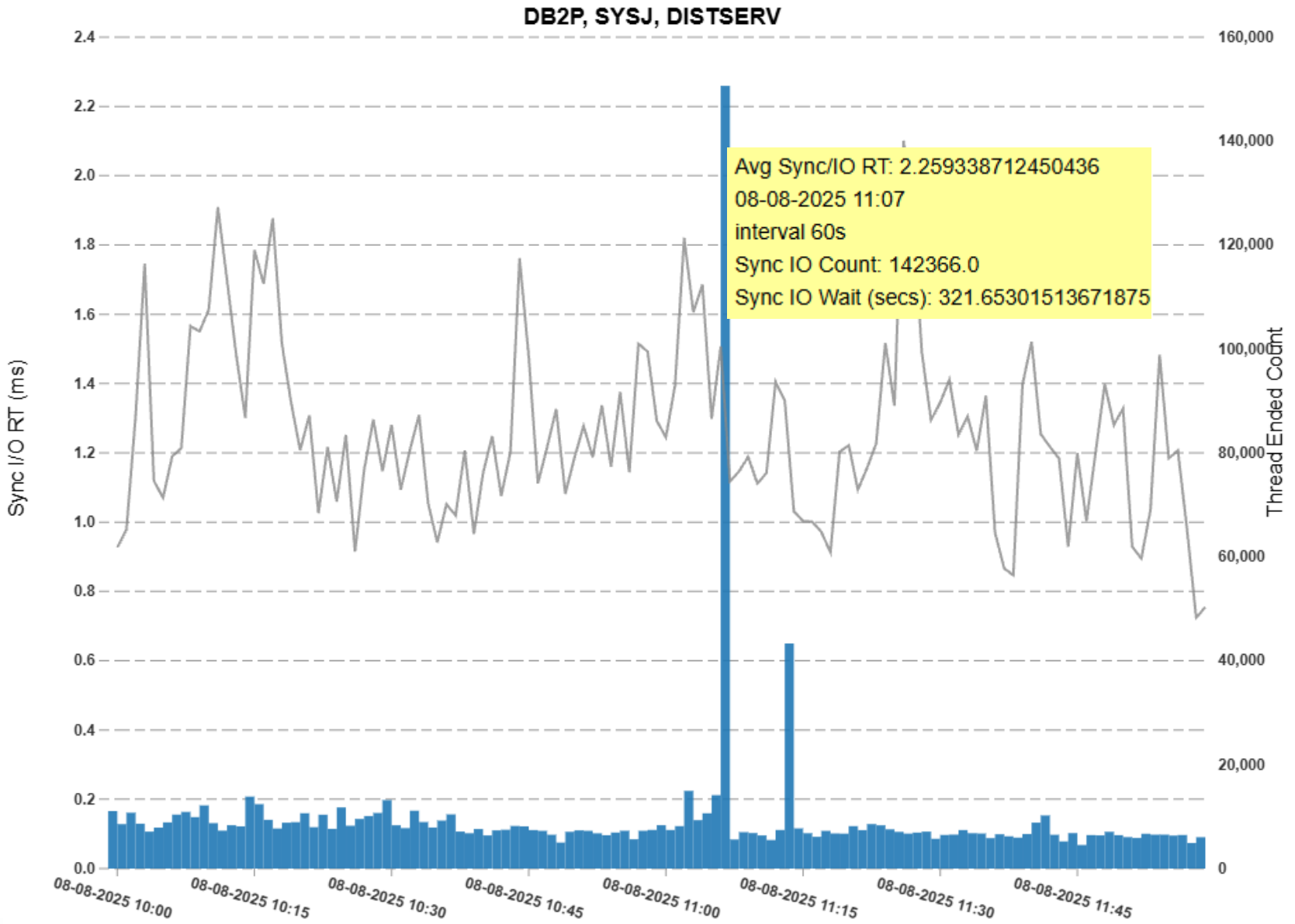
- 500ms is over 1000x longer than the expected average RT today
- So that's ridiculous, right?

Freeze: 8 Filter: sysplex = Value Apply Clear All Toggle Sums 509 Initial rows

sysplex	Usage	DS Name	service class	wkld name	volser	Job Name	Record Type	Int Start Date	Int Start Time	Int End Date	Int end Time	Max RT ms	Avg RT	Total RT
509												143624.7	2251.2	7535522.1
PLEX00	OTHER-KSDS							2025-07-31	00:17:21	2025-07-31	00:22:10	1,132.032	0.288	34,912.79
PLEX00	OTHER-EXTFM							2025-07-31	00:42:41	2025-07-31	00:43:43	991.488	3.769	1,741.27
PLEX00	OTHER-PS							2025-07-31	01:43:51	2025-07-31	01:43:56	974.592	4.046	3,046.63
PLEX00	TEMP-PS							2025-07-31	00:23:42	2025-07-31	00:24:49	951.936	17.517	9,984.69
PLEX00	TEMP-PS							2025-07-31	00:23:43	2025-07-31	00:24:49	951.680	17.426	9,793.41
PLEX00	OTHER-KSDS							2025-07-30	05:00:00	2025-07-30	05:15:00	943.872	0.106	201,028.05
PLEX00	OTHER-DA							2025-07-31	02:15:00	2025-07-31	02:30:00	935.168	7.018	1,045.68
PLEX00	LOADLIB							2025-07-31	02:29:03	2025-07-31	02:29:06	927.872	19.967	978.38
PLEX00	OTHER-DA							2025-07-31	00:45:00	2025-07-31	01:00:00	910.592	2.388	1,554.58
PLEX00	TEMP-PS							2025-07-31	00:48:16	2025-07-31	00:50:49	898.560	3.101	24,928.94
PLEX00	DB2-OBJ							2025-07-30	05:15:00	2025-07-30	05:30:00	880.000	1.241	90,434.16
PLEX00	OTHER-LINE…							2025-07-31	03:15:00	2025-07-31	03:30:00	863.744	1.523	12,051.49
PLEX00	DB2-OBJ							2025-07-31	00:45:00	2025-07-31	01:00:00	863.488	0.244	150,894.23
PLEX00	DB2-OBJ							2025-07-31	03:15:00	2025-07-31	03:30:00	856.064	3.653	19,766.38
PLEX00	DB2-OBJ							2025-07-31	03:00:00	2025-07-31	03:15:00	844.032	1.766	31,842.74
PLEX00	DB2-OBJ							2025-07-31	03:00:00	2025-07-31	03:15:00	843.392	1.885	17,021.55

Plans Average Sync I/O RT

Plans with non-negligible CP CPU

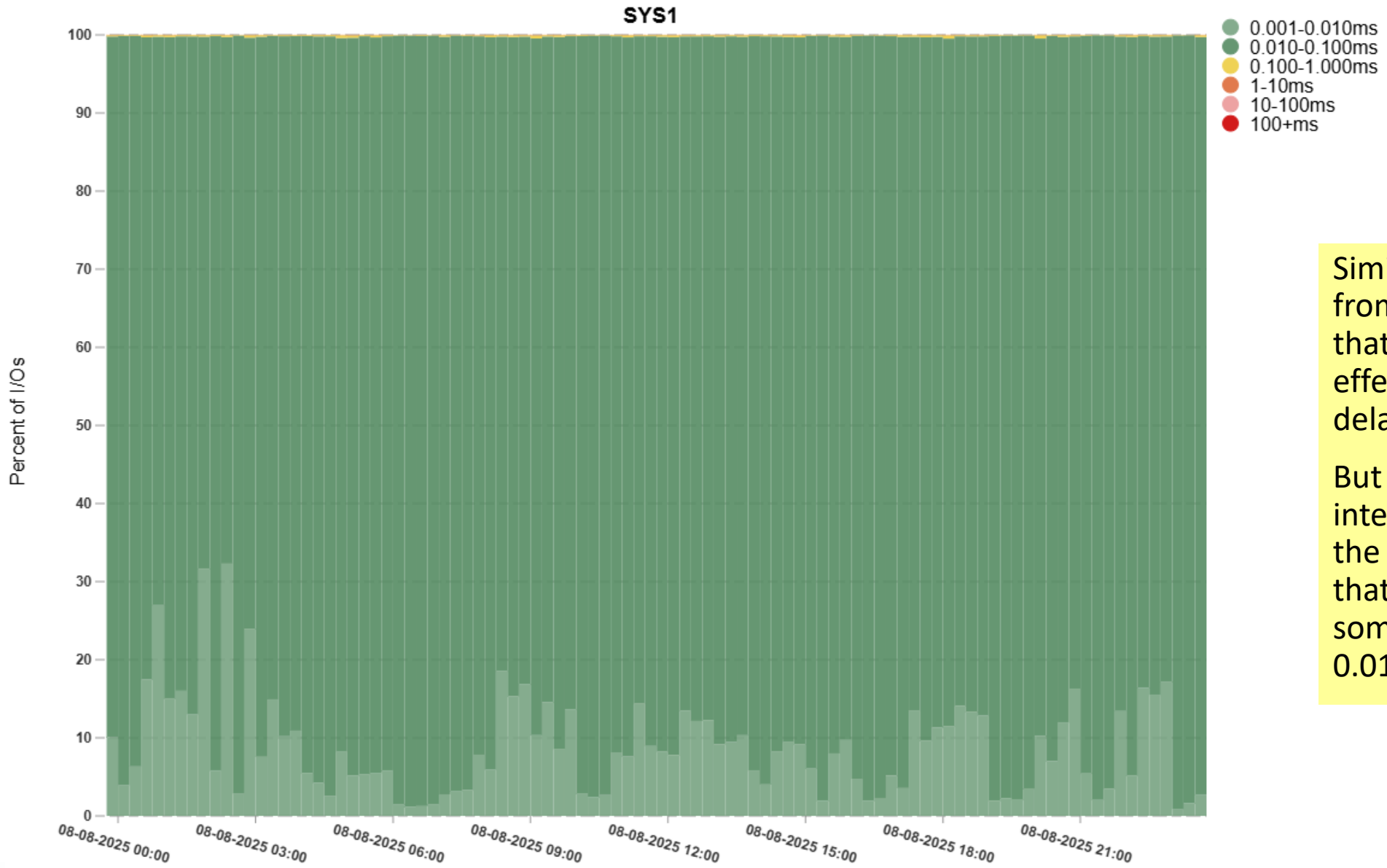


I cheated earlier and didn't show the 11:00 hour for the Db2 Sync I/O response times.

At 11:07 we saw an average Sync I/O RT spike to over 2.2ms!

Had a total Sync I/O wait of 321 seconds for that minute. (Note though that this is for threads ending in the minute, and some may have run for some time.)

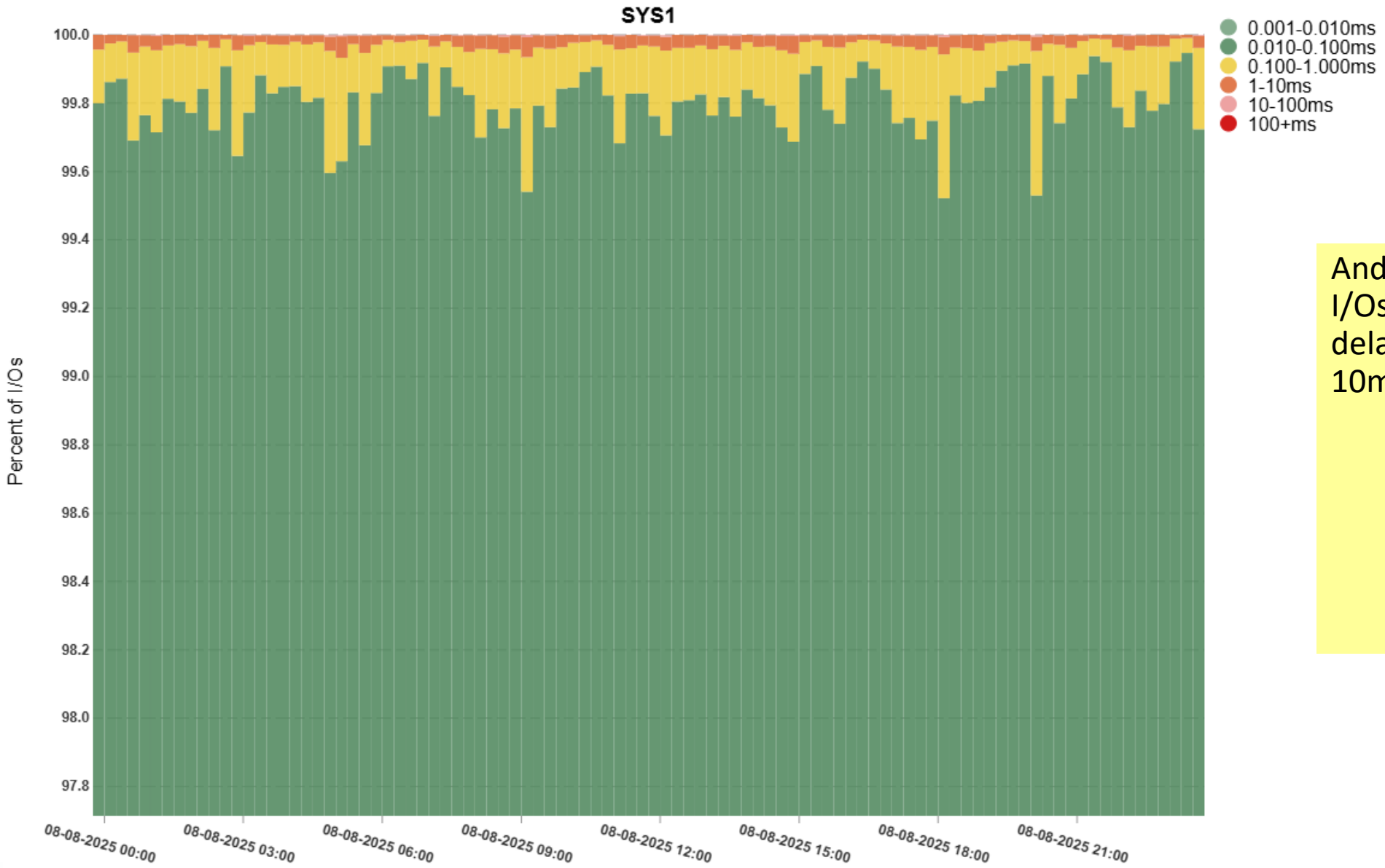
System I/O Interrupt Delay Distribution



Similarly, earlier we saw from the SMF 74 data that this system had effectively 0 interrupt delay.

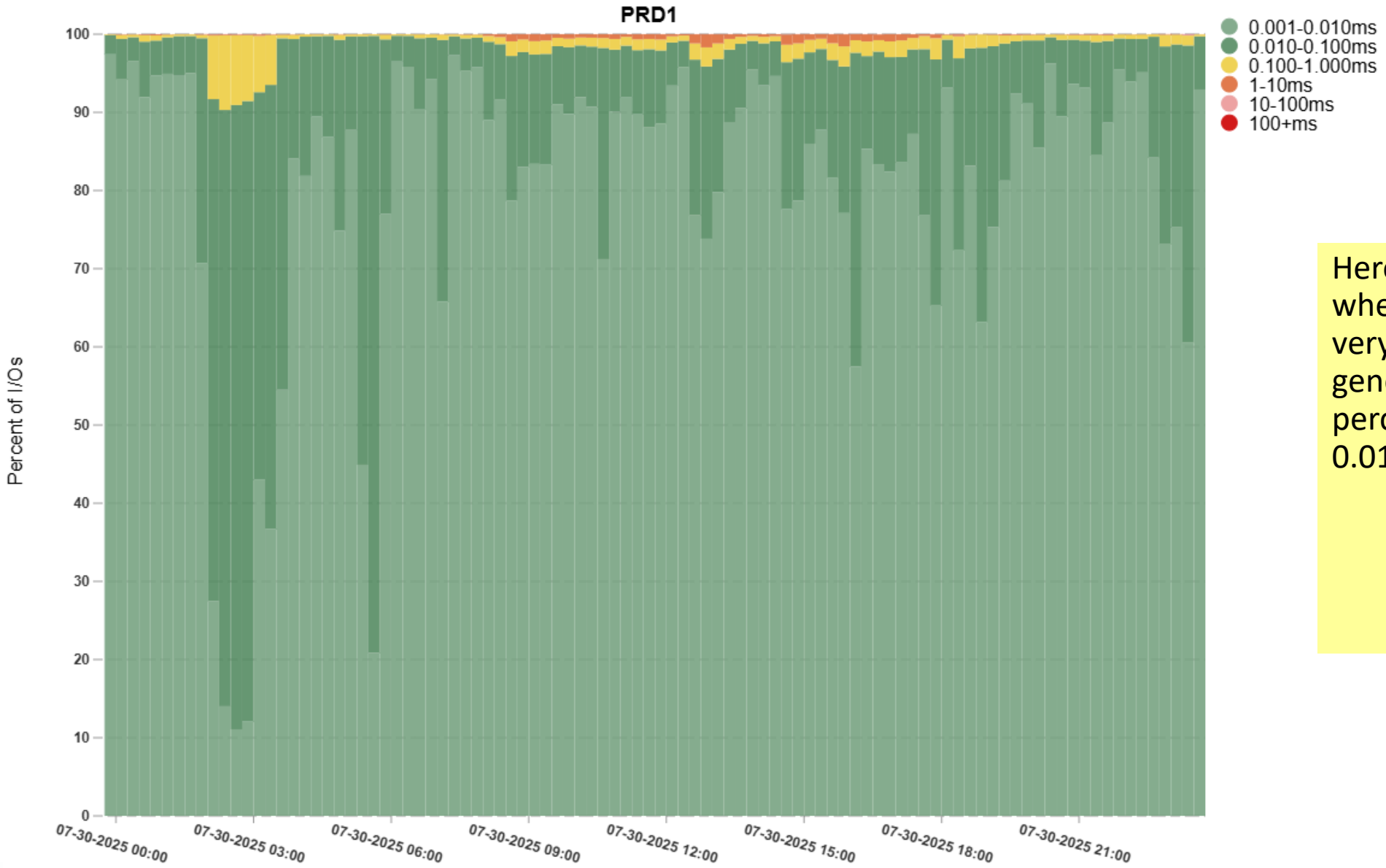
But this histogram of interrupt delays from the SMF 42 data shows that most I/Os suffered some interrupt delays of 0.010 to 0.100ms.

System I/O Interrupt Delay Distribution

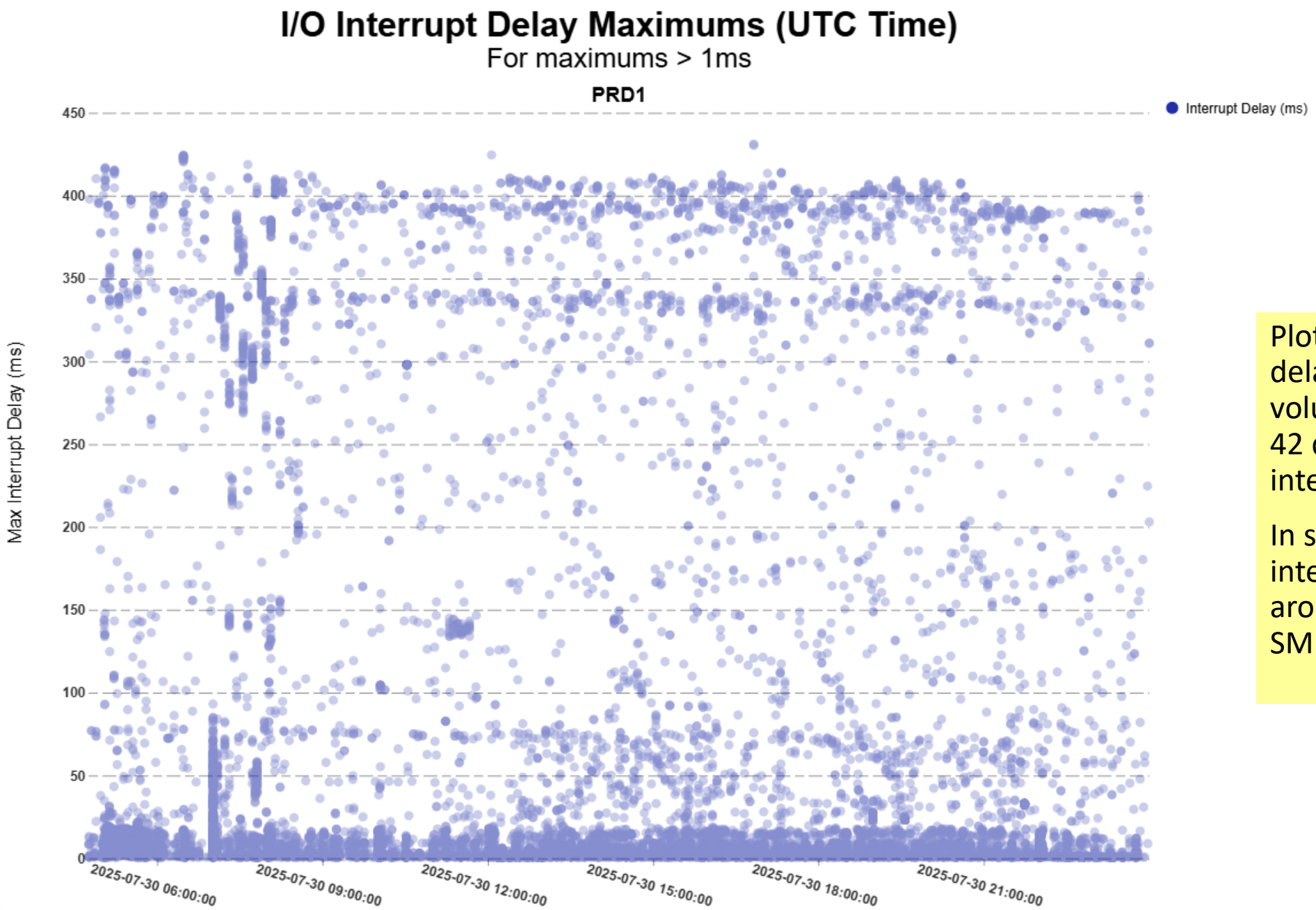


And a tiny portion of the I/Os suffered interrupt delays between 1 and 10ms.

System I/O Interrupt Delay Distribution



Here's a different system where the distribution is very much different, but generally has a larger percentage of I/Os under 0.010ms .

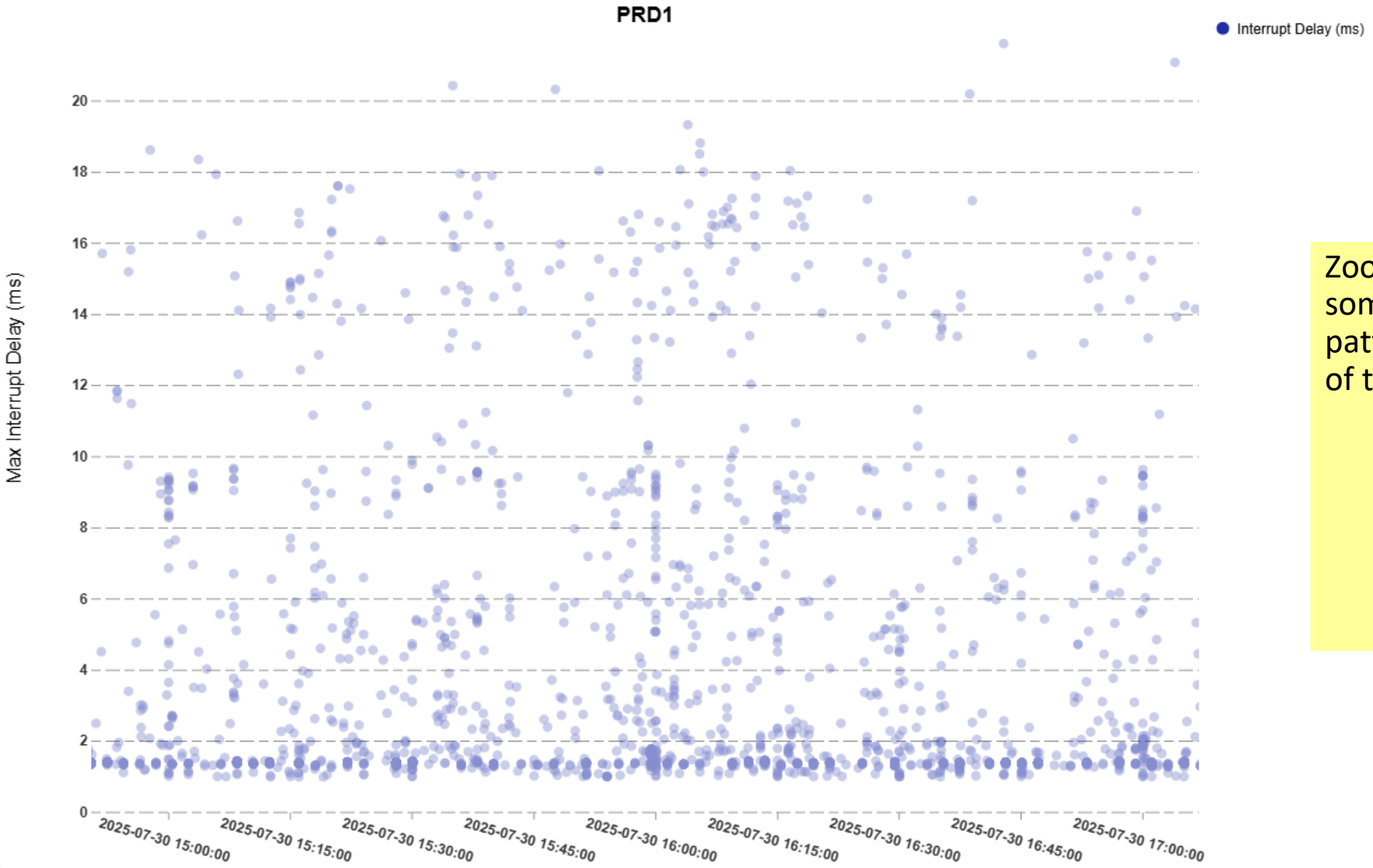


Plotting the interrupt delay maximums (by volume) from the SMF 42 data shows some interesting patterns.

In some cases, I've seen interesting patterns around the end of the SMF interval.

I/O Interrupt Delay Maximums (UTC Time)

For maximums > 1ms



Zoomed in to show some interesting patterns around the top of the hours.

Do we care?

- I don’t know: this gets back to the significance of the individual I/Os
  - And we can’t know the significance of any particular I/O
  - So the max response time may very well be immaterial
- A large maximum may inflate the average RT for no good reason
- Average RT number may or may not reflect the RTs of the significant I/Os
- Interrupt delay can sometimes extend the I/O response time significantly

Some I/Os take much longer than what we’d expect from average RT numbers.
We have very limited knowledge of the significance of those I/Os.

I/O Reduction

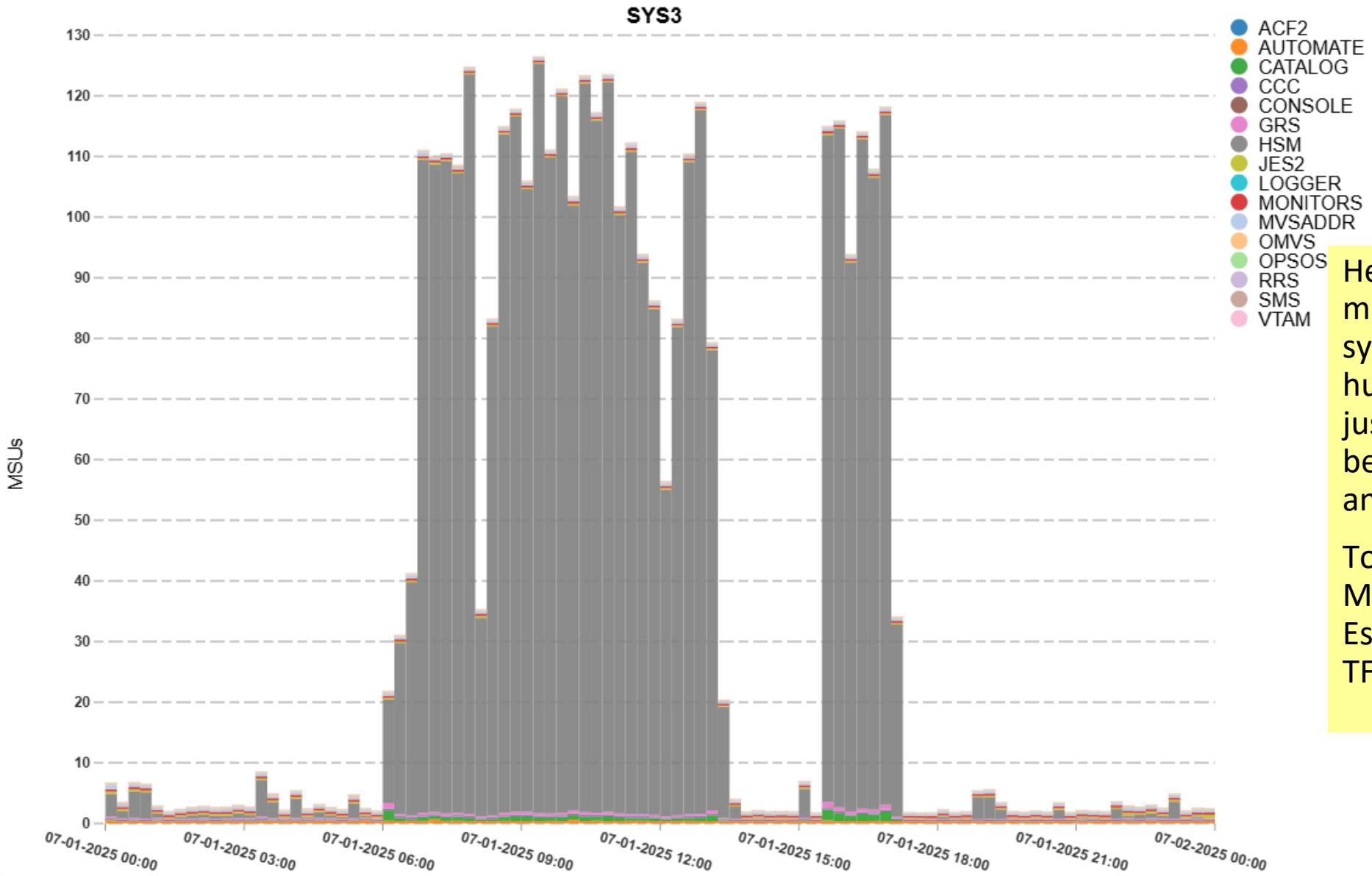
Repeat the mantra of the past

- "The only good I/O is no I/O"
  - I'm not sure who first said this, but this was a commonly heard 25-30 years ago
  - As I/O got faster we all stopped worrying about it as much
- I/O has gotten faster, but CPU capacity has increased even more
  - And CPU consumption drives software costs
  - Software costs are typically more than hardware costs
  - In last 20-25 years CPU reduction has really been the focus
- But reducing I/O still has important benefits:
  - Performance improvements
    - Average RTs may not represent performance impacts for significant I/Os
  - Small CPU reductions (it takes CPU to issue/handle I/O)
  - Possible performance improvement for remaining I/O
- Also: memory has gotten cheaper and larger
  - No better time to keep data in memory!

I/O Reduction Candidates

- Unnecessary work
  - Biggest savings always come from turning off unnecessary work
  - HSM configured for expensive DASD from decades ago should be examined
- Repeated read I/O
  - Re-reading the same data continually implies that data should live in memory
- Sort Work I/O
  - Can you give sort more memory to SORT to avoid using disk-based sort work?
- Use compression
  - This can save disk storage and reduce the number of I/Os required
  - Despite hardware acceleration, will cost some (probably small) amount of CPU
  - Net benefit here is probably more borderline today, but can be worth looking at
  - Reduces writes as well as reads

Interval MSUs for Top Report Classes by LPAR

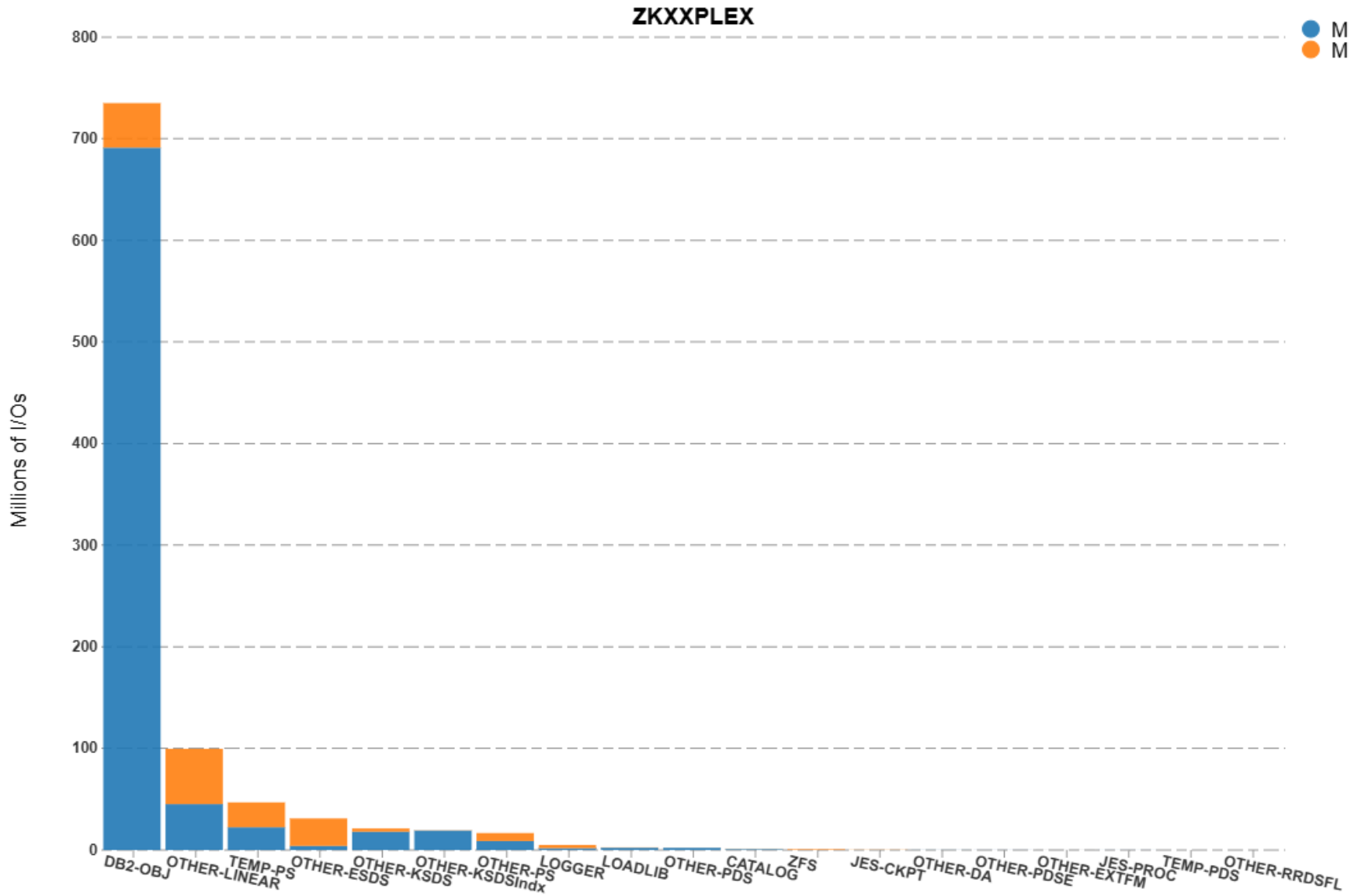


Here we have an LPAR managing HSM for the sysplex consuming hundreds of MSU-hours just moving data between ML0 and ML2 and recycling ML2 tapes.

Today, migration from ML0 should be very lazy. Especially if you're on TFP instead of R4HA.

Top Dataset I/O Counts by Dataset Usage

Total I/Os for Study Period



This is a very common pattern for Db2-using systems: Db2 does the most I/O. And most of it is read I/O.

DCOLLECT data has information about the allocated size of the datasets. Comparing that to the amount of data read as recorded on the SMF 42 indicates that Db2 is re-reading some of these datasets a whole lot!

Top 5 datasets here are just 5 GB. Top 10 is 26 GB.

Top Datasets by Cache Read Hits

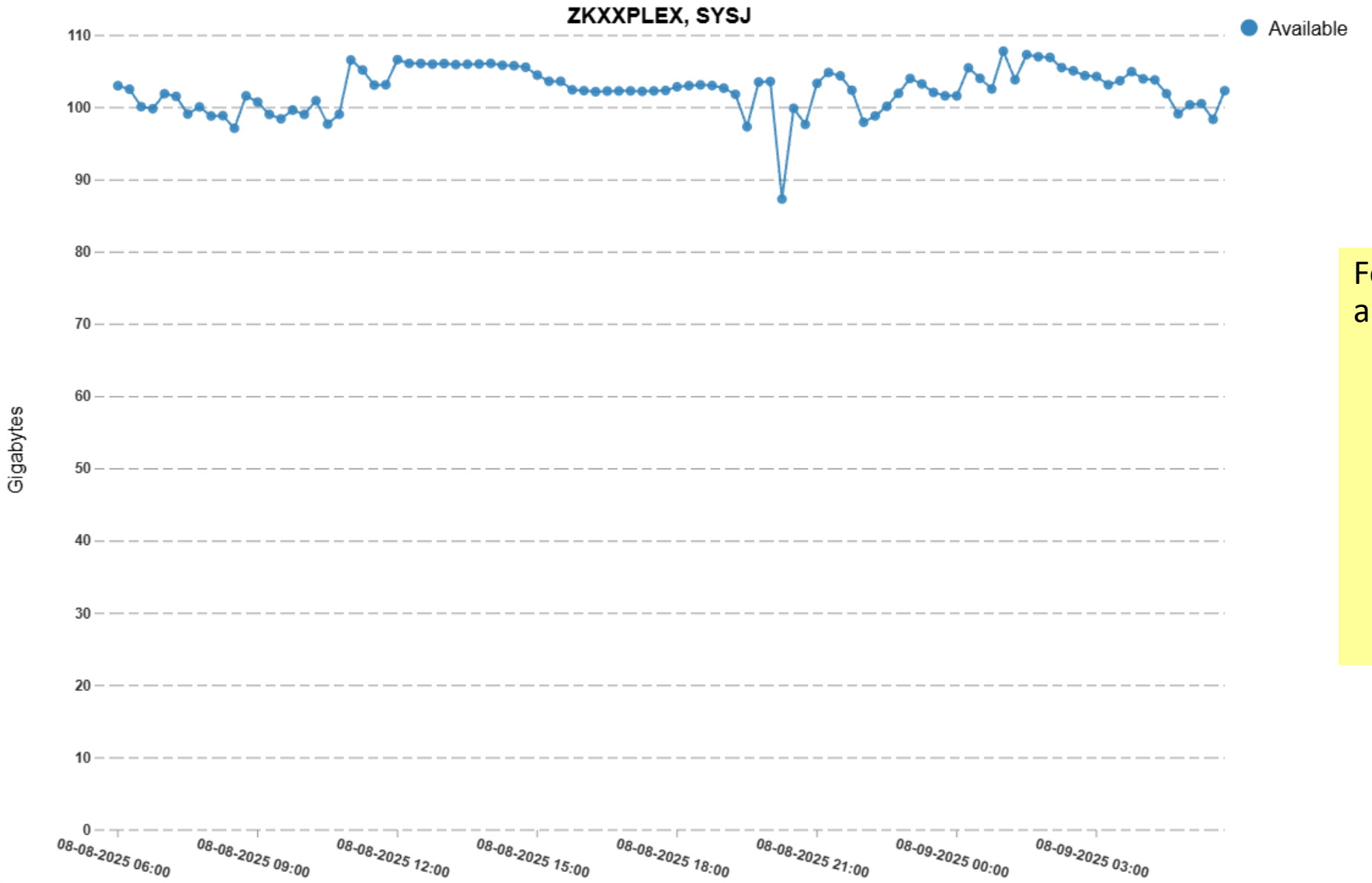
ZKXXPLEX

Freeze: 4 Filter: sysplex = Value

s...	Us...	DS Name	M Read Ops	M Cache Read Hits	Allocated MiB	Read MiB	Read-Allocated Ratio	M Write Ops	Cache Hit Pct	Read Hit RT secs	42.6 Records
200			366	174	686968	31000319	101028	13	18580	35469	31545
ZK...	DB2-OBJ	VS62359ABD.DBZCPA.IX20460A....	65.97	0.58	2,361	8,121,757	3,439.6	0.00	98.4	80.7	147
ZK...	DB2-OBJ	VS50359ABD.DBZABW02.TS064...	32.48	1.98	732	3,812,513	5,208.6	0.00	92.3	635.9	97
ZK...	DB2-OBJ	VS75359ABD.DBZAAD.TS17613....	13.47	0.05	428	1,651,740	3,859.3	0.00	93.6	18.8	17
ZK...	DB2-OBJ	VS80359ABD.DBZAHT.TS02813.J...	9.17	9.00	1,010	47,737	47.3	0.00	98.7	1,087.1	48
ZK...	DB2-OBJ	VS78359ABD.DBZAAD.TS17613....	9.16	4.33	402	1,257,392	3,127.5	0.00	99.9	515.8	13
ZK...	DB2-OBJ	VS50359ABD.DBZAEG.TS07292.I...	9.03	8.47	4,722	114,618	24.3	0.00	99.9	1,177.5	782
ZK...	DB2-OBJ	VS80359ABD.DBZARB.TS20703.I...	8.57	7.90	4,721	72,319	15.3	0.01	94.3	794.3	403
ZK...	DB2-OBJ	VS62359ABD.DBZMEW.TS20191....	7.95	6.24	4,829	659,480	136.6	0.00	98.6	1,722.6	90
ZK...	DB2-OBJ	VS80359ABD.DBZARB.TS20703.I...	6.72	6.03	4,721	64,648	13.7	0.01	92.9	595.8	307
ZK...	DB2-OBJ	VS46359ABD.DBZAEL.IX17820E....	6.36	6.27	2,189	24,987	11.4	0.00	98.7	573.8	53
ZK...	DB2-OBJ	VS77359ABD.DBZAEG.IX09188E...	5.75	0.49	482	614,302	1,273.7	0.00	98.4	305.5	12
ZK...	DB2-OBJ	VS50359ABD.DBZAUM.TS03167....	5.75	0.01	684	674,805	986.4	0.00	97.9	4.2	47
ZK...	DB2-OBJ	VS80359ABD.DBZARB.TS20703.I...	5.36	5.12	2,249	41,063	18.3	0.00	97.1	496.7	257
ZK...	DB2-OBJ	VS80359ABD.DBZAHT.TS02813.J...	5.26	5.22	344	27,326	79.3	0.00	99.8	628.9	74
ZK...	DB2-OBJ	VS50359ABD.DBZAHT.TS02813.J...	5.14	3.70	3,952	176,096	44.6	0.00	94.5	429.1	251
ZK...	DB2-OBJ	VS50359ABD.DBZAHT.TS02809.I...	4.99	4.52	4,134	67,911	16.4	0.00	95.6	556.9	854
ZK...	DB2-OBJ	VS50359ABD.DBZAEG.TS07292....	4.90	4.63	2,310	63,762	27.6	0.00	99.9	652.7	423
ZK...	DB2-OBJ	VS49359ABD.DBZAET.TS07315.I...	4.55	4.25	3,393	525,469	154.9	0.00	100.0	1,921.6	73
ZK...	DB2-OBJ	VS80359ABD.DBZARB.IX20703B...	4.47	0.02	1,117	554,821	496.7	0.00	97.4	4.8	96

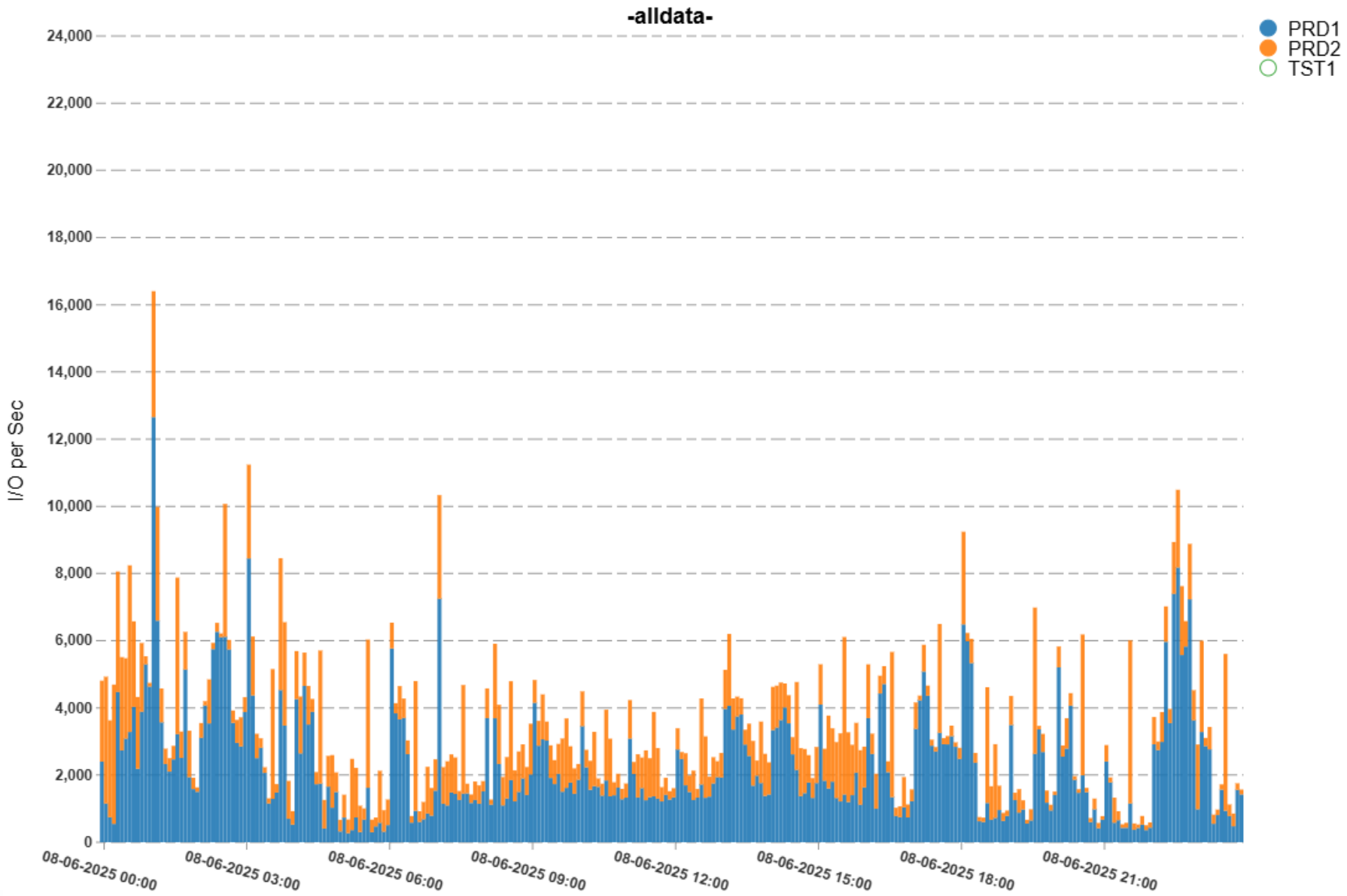
Minimum Available Central Storage

Gigabytes



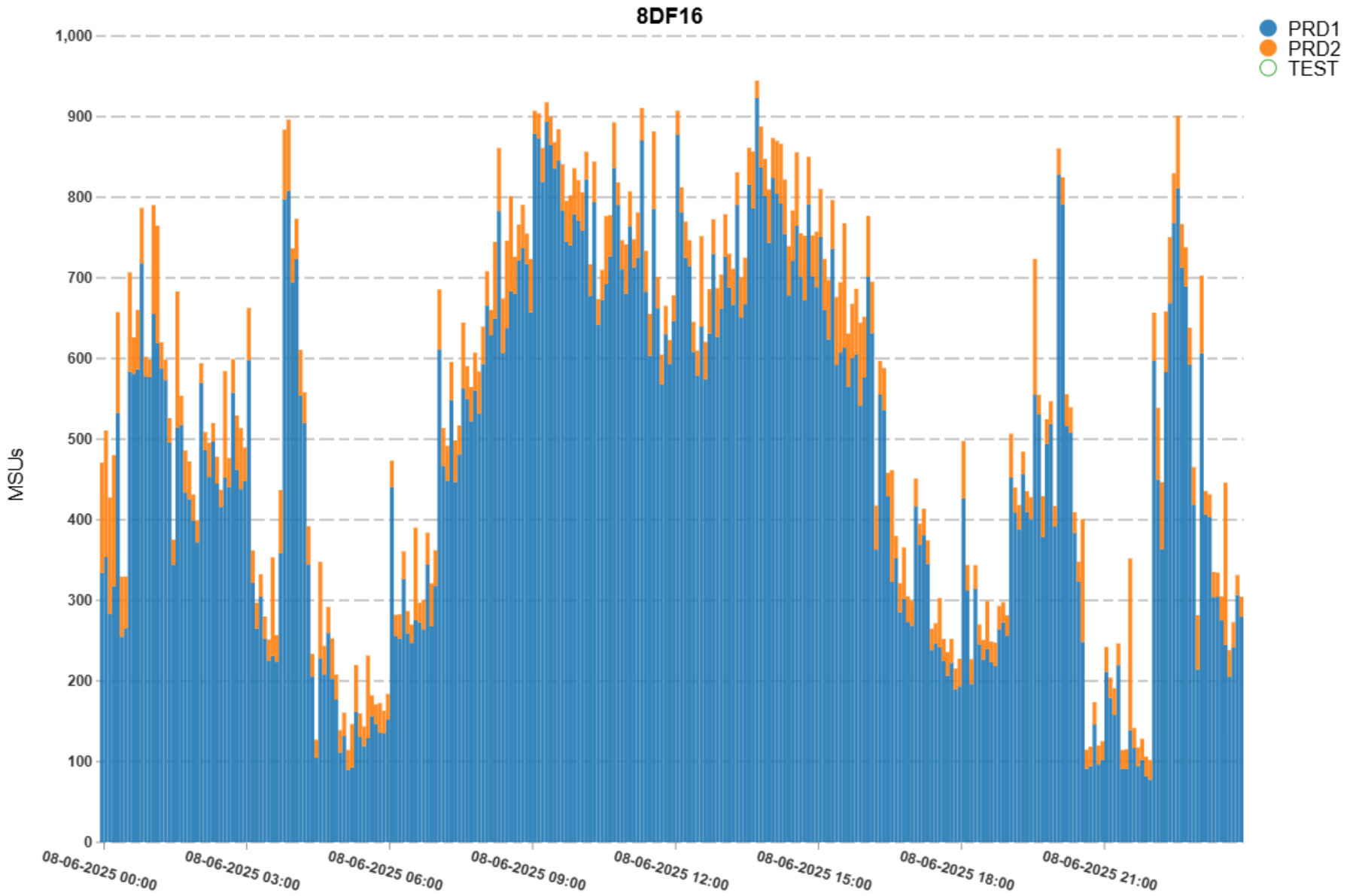
For context, 26 GB is not a lot memory today.

Device I/O Rate

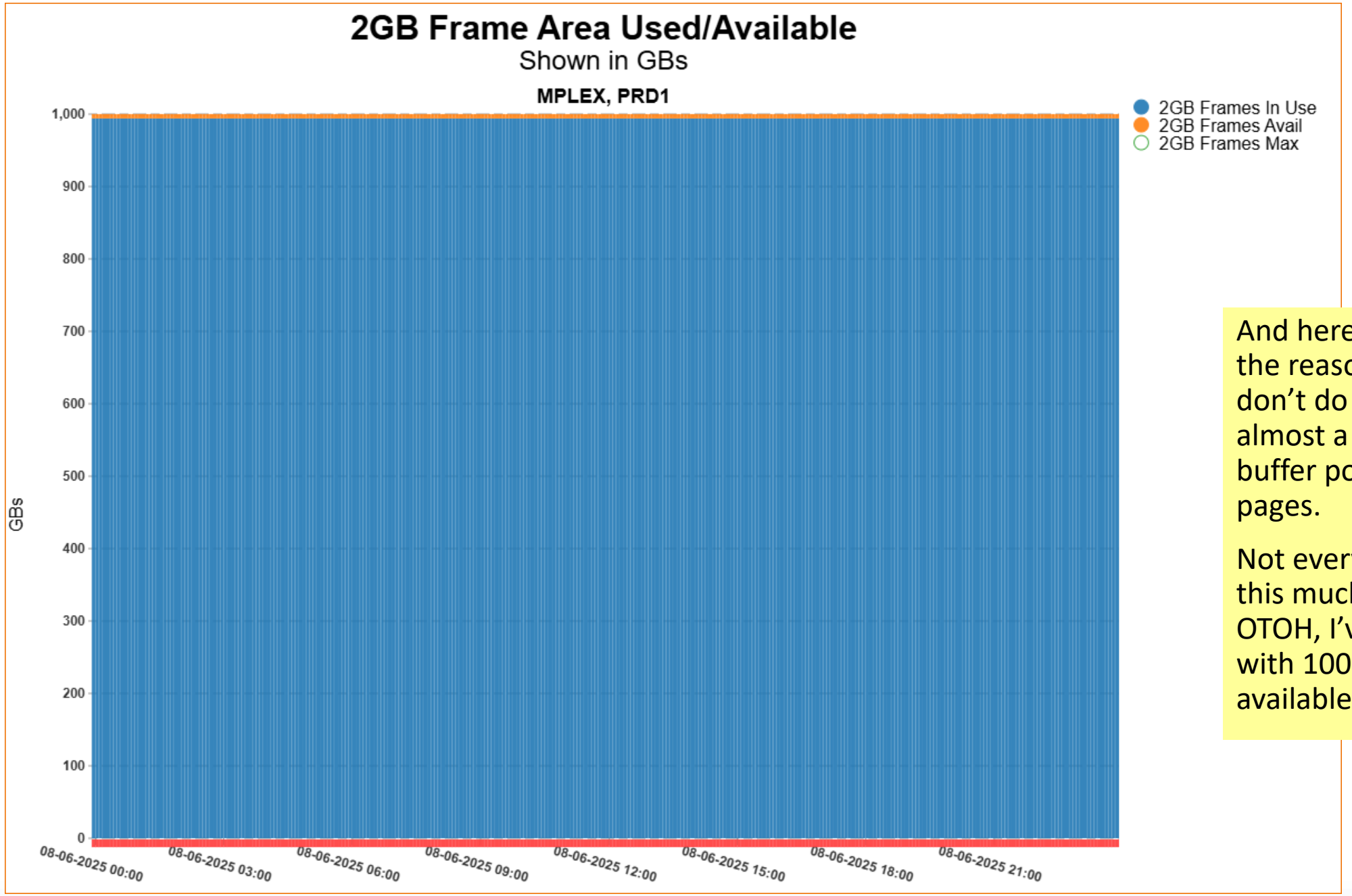


How little I/O can you do? In some cases: very little.

CEC Actual MSUs for LPARs



These were not trivial systems.



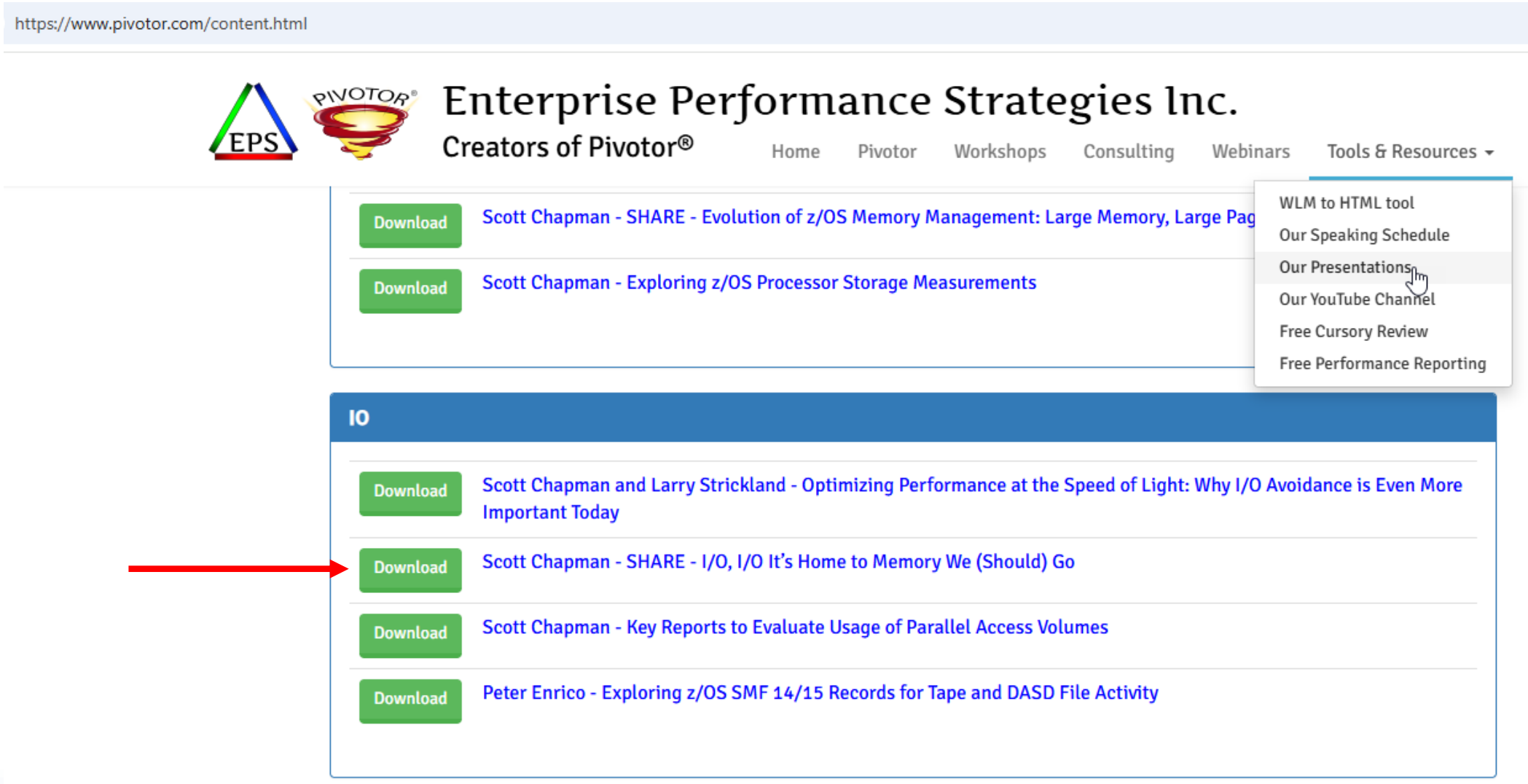
And here's a big part of the reason why they don't do much I/O: almost a TB of Db2 buffer pools. Using 2GB pages.

Not every system needs this much of course. OTOH, I've seen LPARs with 100s of GBs of available memory.

I/O Reduction: Things to think about

- Involve the application team to help understand what I/Os are significant
  - You won't of course get a complete picture, but even partial information is useful
  - Reduce things that are more significant to the application first
- Don't be afraid to use memory
  - In some cases, worsening some overnight sorts to help day-time processing is a beneficial trade-off
  - Do you have reserve storage the LPARs aren't using?
  - Paging to SCM much less problematic than paging to disk
    - Even paging to disk for rare events may be "ok" if it offsets more I/O over more hours
- Some metrics may get worse as you become more memory-intensive
  - L1MP and TLB CPU Miss Percent may increase
    - Remember you're offsetting even worse metrics!
    - Use large (1MB or 2GB) fixed pages for Db2 bufferpools
    - CPI might improve

See also https://www.pivotor.com/content.html



Summary & Best Practices

- z/OS has a lot of I/O measurements, in lots of SMF records
  - Those measurements may be in different units
  - 128µs units are larger than some I/Os take today
- Not all I/Os are of equal significance
  - But we don't know which individual I/Os are significant
- Averages can hide outliers that much larger than the average
- Interrupt delay can significantly lengthen I/O times for some LPARs
- It's hard to know how much I/O response time is impacting the business
- Reducing I/O by keeping more data in memory is (still) valuable
  - Improve performance
  - Improve performance of remaining I/Os
  - Reduce CPU (probably only slightly, but more than if you do zHL)

Questions?

Thanks for attending!