Store Into Instruction Stream: Searching for the Culprits to Save CPU

Scott Chapman

scott.chapman@epstrategies.com

Enterprise Performance Strategies

November 2021

Session 6BH

GSE UK Conference 2021 Charity Raffle

- The GSE UK Region team hope that you find this presentation and others that follow useful and help to expand your knowledge of z Systems.
- Please consider showing your appreciation by kindly donating to our charities this year, Royal National Lifeboat Institution (RNLI) & Guide Dogs for the Blind. Then follow the link on your receipt to enter your receipt number & amount donated into the GSE Raffle. You will get a raffle entry for every pound donated.
- Follow the link below or scan the QR Code:

http://uk.virginmoneygiving.com/GuideShareEuropeUKRegion

Supporting

Contact, Copyright, and Trademarks

Questions?

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

Copyright Notice:

Trademarks:

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

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

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

Other trademarks and registered trademarks may exist in this presentation

Abstract

- CPU overhead caused by the programming anti-pattern of updating data that's too close to the instruction stream has long been a known problem. IBM came out with a nice formula that highlights time intervals where that may practice may be causing significant overhead. But that's only part of the battle: the next question is how do you find the possible culprits so you can remediate them?

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

z/OS Performance workshops available

During these workshops you will be analyzing your own data!

- Essential z/OS Performance Tuning
  - Via Zoom, June 21-25, 2021
- WLM Performance and Re-evaluating Goals
  - Via Zoom, September 20-24, 2021
- Parallel Sysplex and z/OS Performance Tuning
  - Via Zoom, November 16-17, 2021
- Also… please make sure you are signed up for our free monthly z/OS educational webinars!

Like what you see?

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

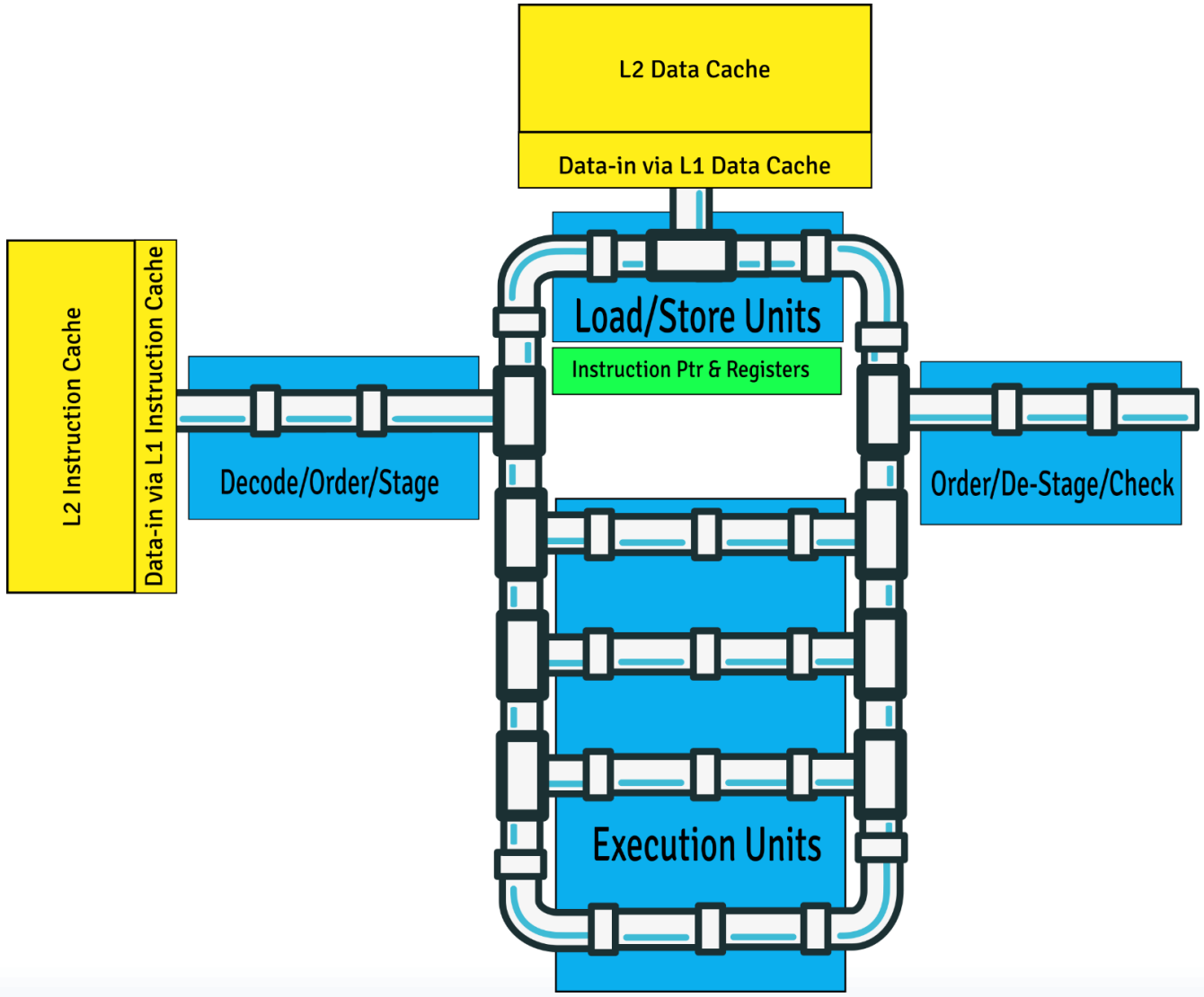
- If you don't see them in your performance reporting tool, or you just want a free cursory performance review of your environment, let us know!
  - We're always happy to process a day's worth of data and show you the results
  - See also: http://pivotor.com/cursoryReview.html

- 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!

Store Into Instruction Stream

Modern Superscalar Processors

- Modern processors are complicated
- The appearance that program instructions happen in order as written is a fiction maintained by the processor
  - Many instructions will execute out of order
- Multiple instructions in-flight at any moment
- Aim: 1+ instruction finished / cycle
- Lots of things can cause pipeline stalls
  - L1 / TLB misses
  - Branch prediction misses
  - Data dependencies
  - Long instructions
- When stalled, there are parts of the chip idle (not doing work)
  - Out-of-order execution helps avoid pipeline stalls



SIIS

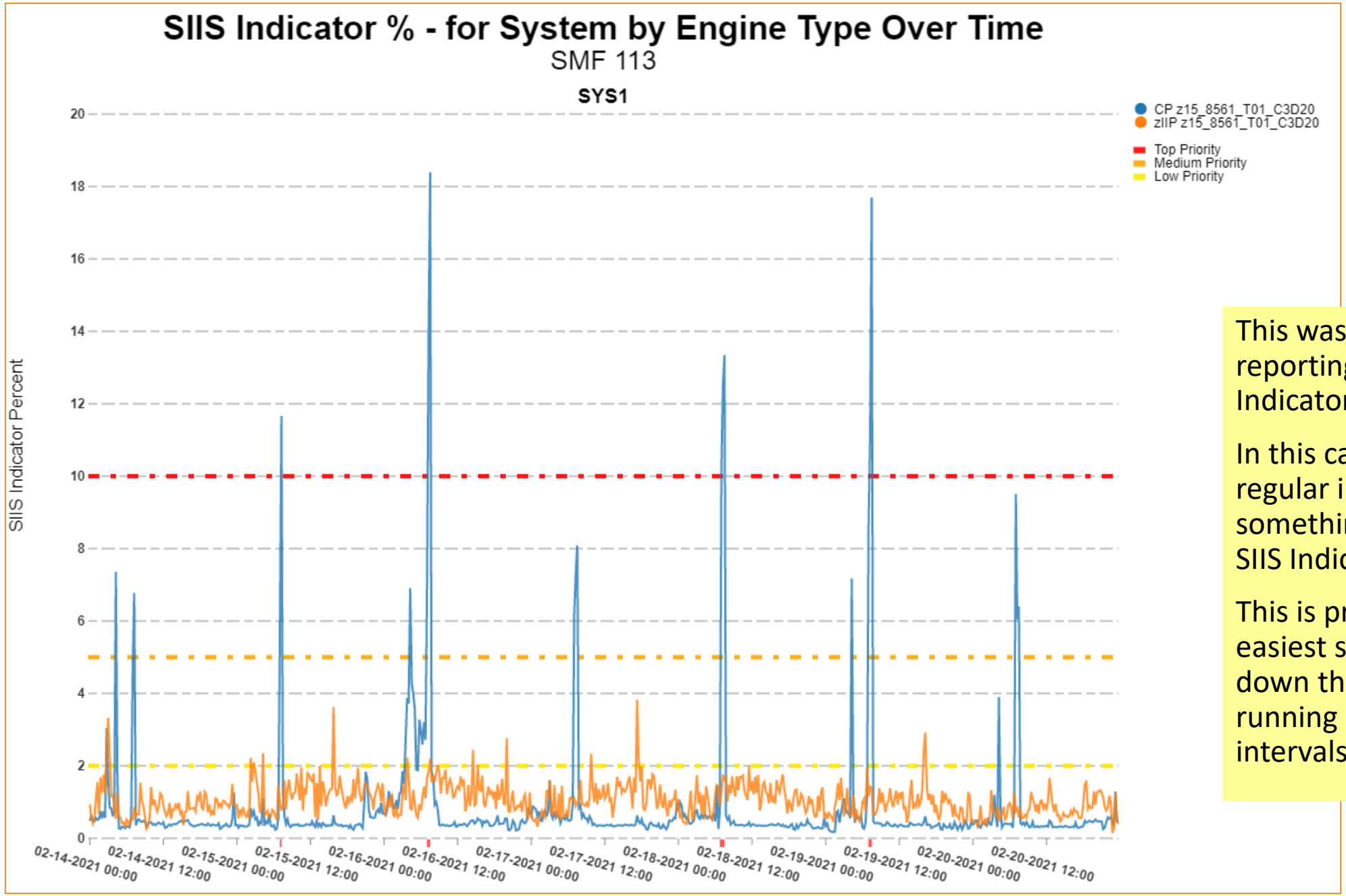
- Store Into Instruction Stream (SIIS) describes a situation where code writes to memory that is within 1 cache line (256 bytes) of the executing instructions
  - That update will trigger a flush of that cache line from the L1 Instruction cache, which will trigger a pipeline flush
  - This can be a significant performance hit!
  - Obviously doing this once is not a noticeable problem, but doing it repeatedly can be
- Primarily is seen in programs written in Assembler
  - FORTRAN protected against this since about 2002
  - COBOL v4.2 issue only when compiled with NOOPT and/or TEST options, which should not be the case for production code
    - If you have production code with NOOPT/TEST, SIIS is probably the least of your inefficiencies

Do you have a SIIS problem?

- See: https://www.ibm.com/support/pages/node/6355777
- The IBM formula is effectively the percentage of instruction stream updates that required L3 intervention
- IBM thresholds for recommended action:

SIIS Description	SIIS Indicator %	Action
Noise – it will never be 0%	< 2%	None
Minimal SIIS impact	2% < 5%	Low Priority but potential MSU savings
Noteworthy SIIS impact	5% < 10%	Medium Priority – Investigate and Remediate
Considerable SIIS impact	>= 10%	Top Priority – Investigate and Remediate

- Note that this doesn't estimate the actual savings
  - In some cases that can be significant
  - In some cases you can have high SIIS % during low-utilization times
- How do you find the culprit(s)???



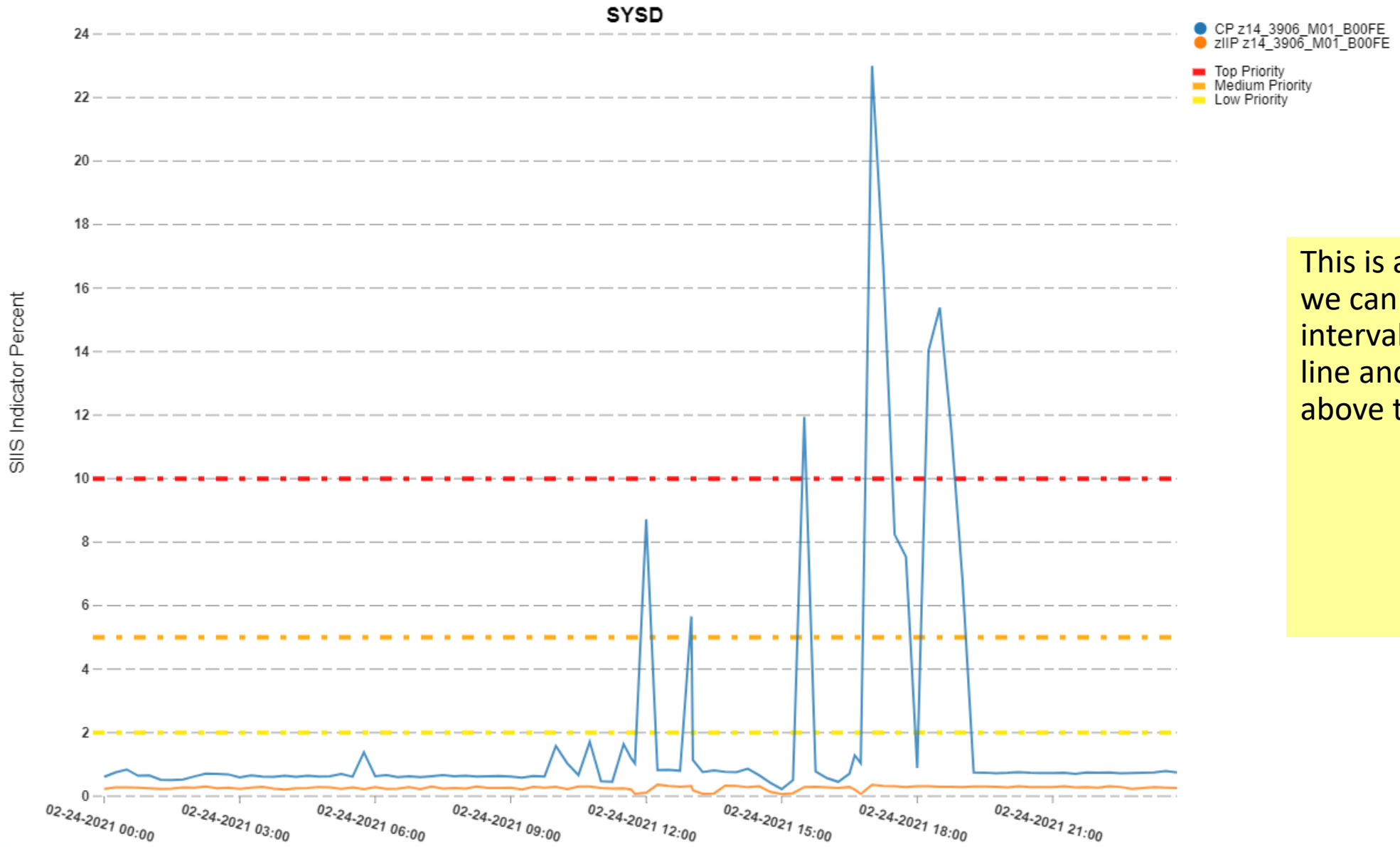
This was our initial reporting of SIIS Indicator %.

In this case there's a regular instance of something triggering the SIIS Indicator high.

This is probably the easiest scenario to track down the culprit: what's running during those intervals?

SIIS Indicator % - for System by Engine Type Over Time

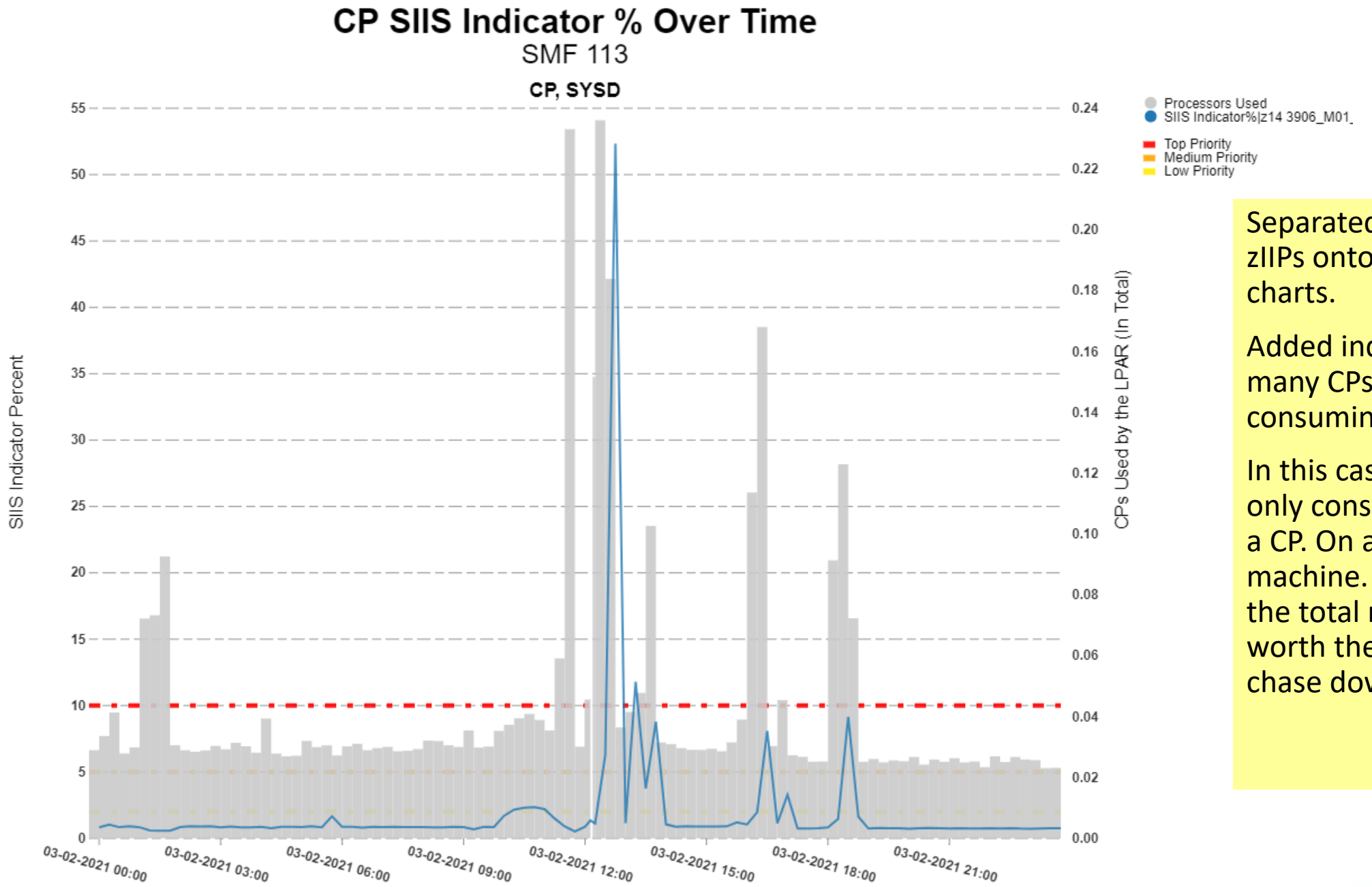
SMF 113



This is a single day but we can see multiple intervals above the 10% line and some more above the 5% line

SIIS Reporting Issues

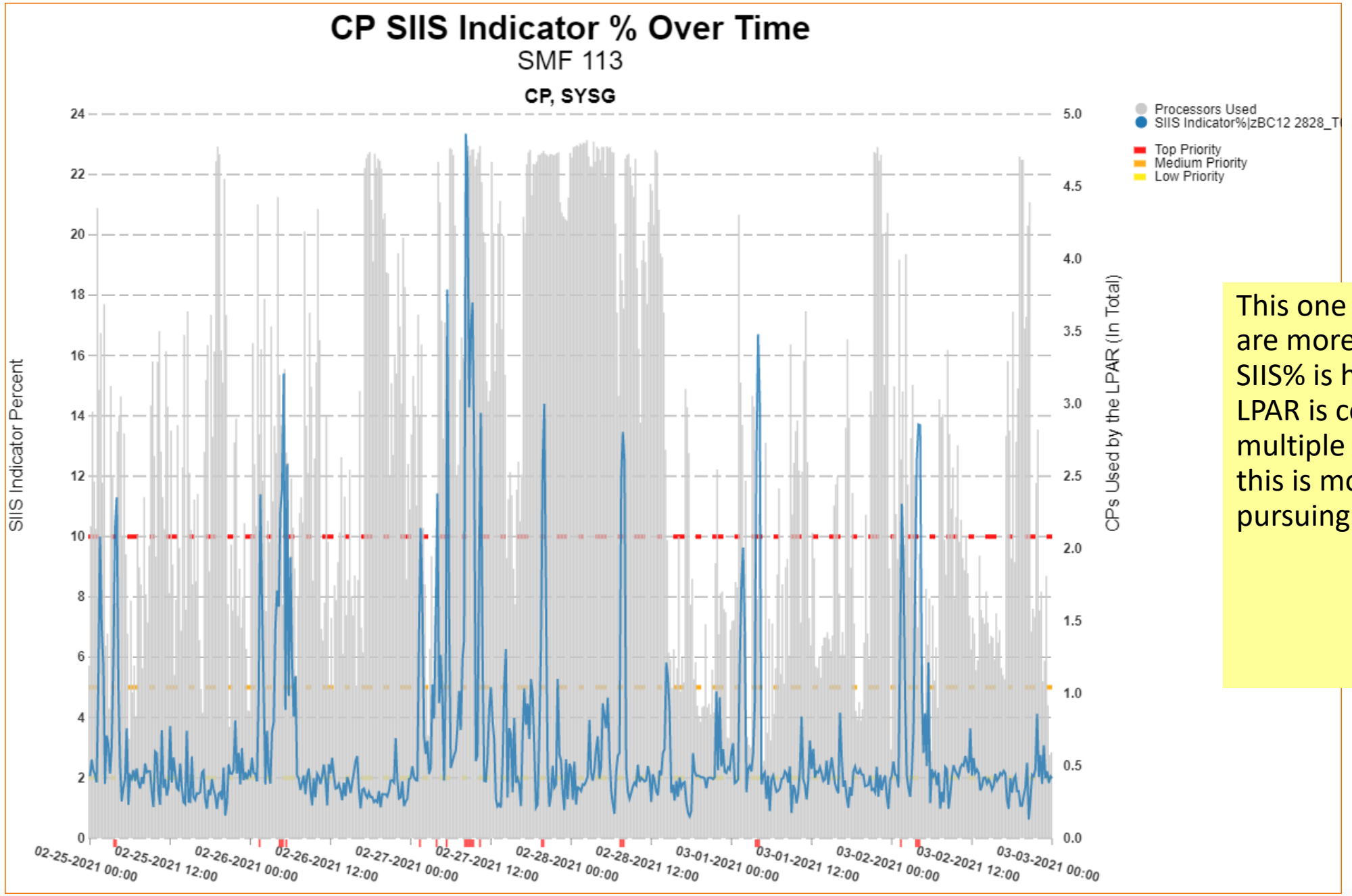
- How much savings are we really talking about?
  - Is it going to be worth the time and trouble of tracking down the culprits?
- Do we really care about SIIS on zIIP engines?
  - Probably not since zIIPs are cheaper and generally more abundant
  - And any Assembler code running on zIIPs is going to be coming from ISVs
- How can we find the culprits?
  - Look for commonly running assembler-based programs in problem times
  - Look for high CPIs in problem times? (Using the SMF 30 instruction counts)
- So we made some reporting changes…



Separated out CPs and zIIPs onto their own charts.

Added indicator for how many CPs the LPAR is consuming (as a whole).

In this case the LPAR is only consuming <20% of a CP. On a 6-way machine. So about 3% of the total machine. Is it worth the trouble to chase down?

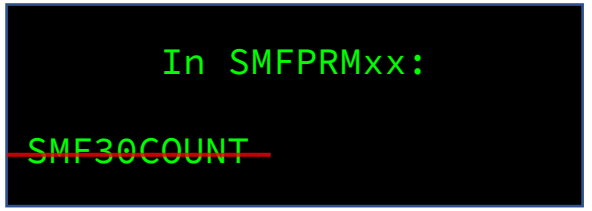


This one looks like there are more intervals where SIIS% is high and the LPAR is consuming multiple CPs so maybe this is more worth pursuing.

SMF 30 Instruction Counter

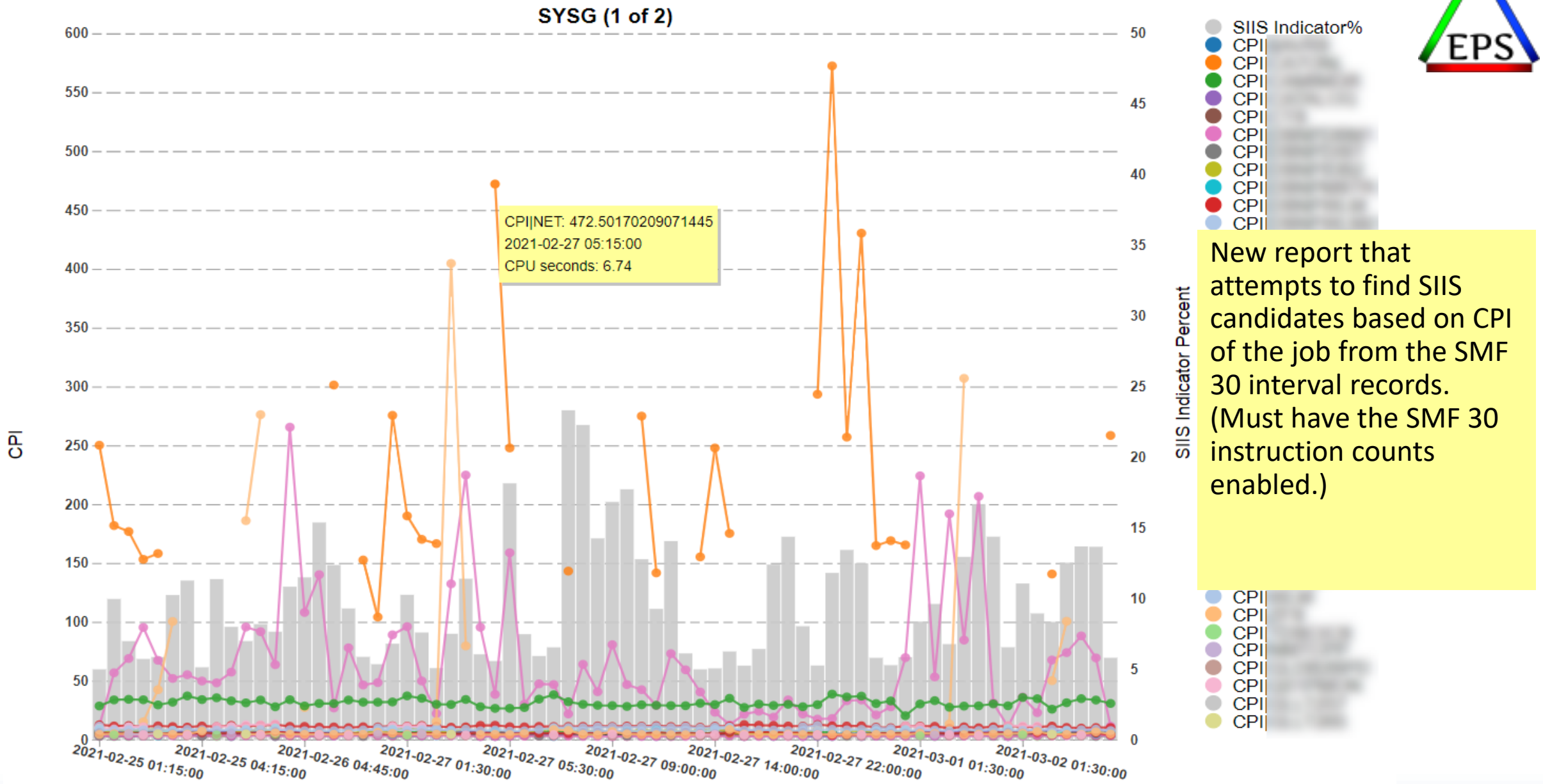
- SMFPRMxx option SMF30COUNT enables the SMF 30 Counter Data Section
  - Default is NOSMF30COUNT
- The idea for these counters was that while CPU time is variable due to things like cache contention, the number of instructions being executed should be stable, so maybe that would be a better measurement to use
- Except it ended up not being stable
  - CPU timers subtract out interrupt handling time
  - There's no similar mechanism for backing out interrupt handling instructions
  - So the instruction counts are potentially more variable than CPU time
- So even though section is relatively small, why bother?
  - Recommendation: don't use this

Time to revisit this?!?



SIIS - Job Name Candidates

(Job Name from Step EXEC=)



New report that attempts to find SIIS candidates based on CPI of the job from the SMF 30 interval records. (Must have the SMF 30 instruction counts enabled.)

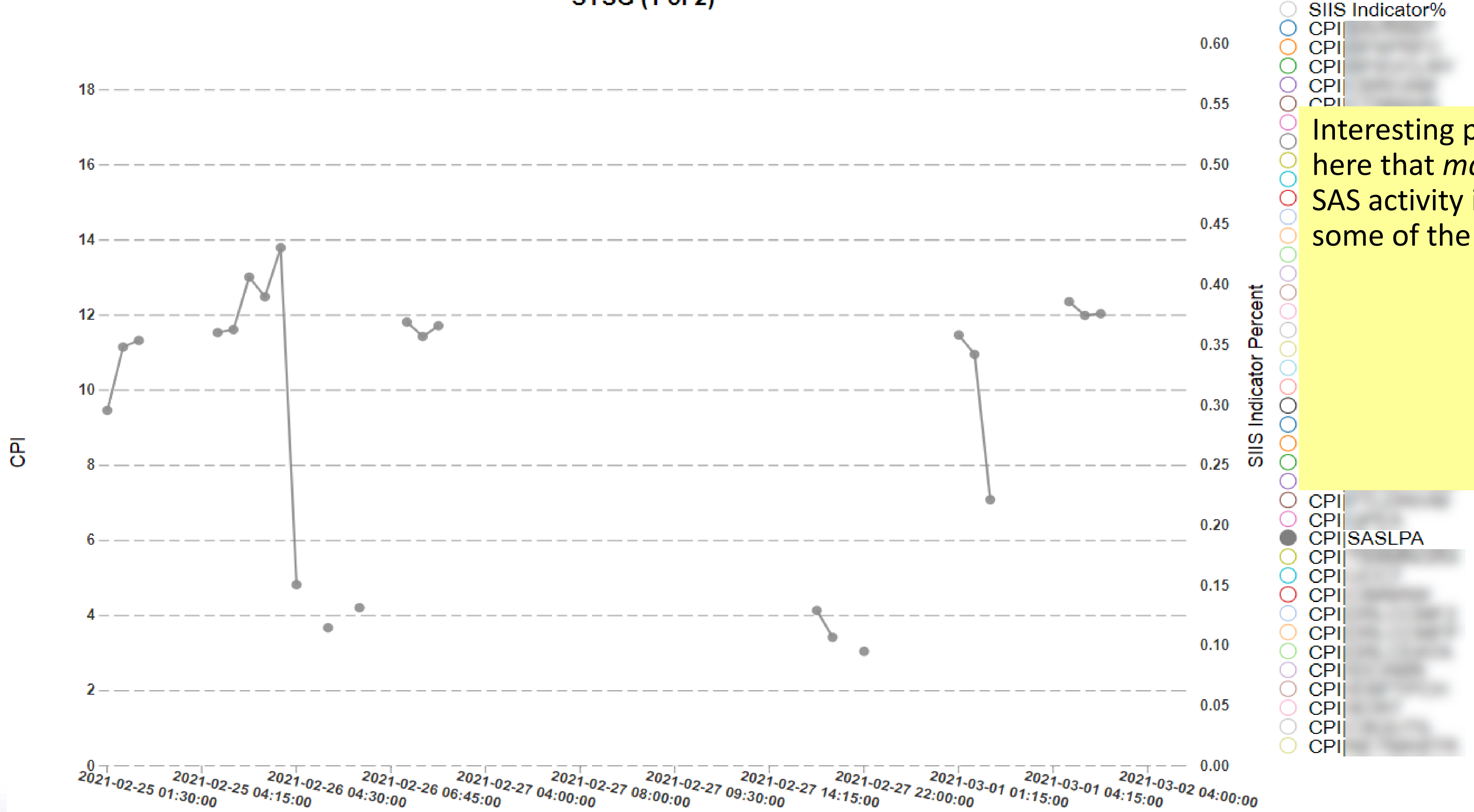
SIIS Culprits report

- Note that there were several address spaces that had really high CPIs!
  - NET, *DBM1, OAM
  - Commonality: compression and encryption
  - Compression and encryption instructions will naturally take many cycles to complete so address spaces making heavy use of these will have high CPIs
  - This is not indicative of a problem!
- Note that if you hover over a point you get the CPU time consumed by that address space in that interval to help you determine if you want to pursue
  - High CPI with low CPU usage = immaterial (low usage could cause high CPI)
- Look for things that appear in multiple intervals
  - And are written in Assembler
- Also have the report by PGM=

SIIS - Program Name Candidates

(Program Name from Step EXEC=)

SYSG (1 of 2)



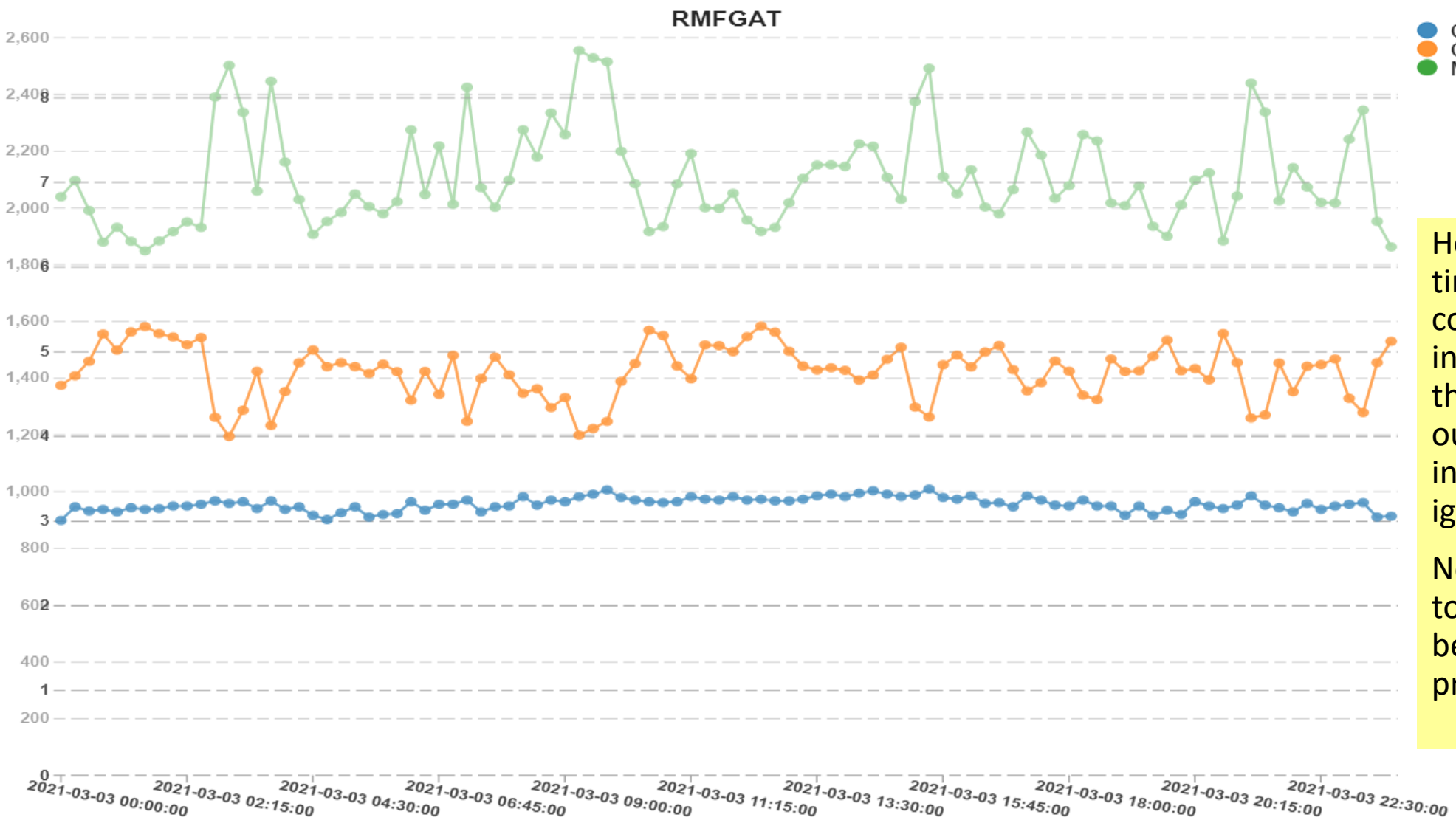
Interesting possibility here that *maybe* some SAS activity is driving some of the high SIIS %.

Can we trust SMF 30 CPI?

- Short answer: “no”
- Long answer: “no, but it *may* still be useful for finding SIIS culprits”

- Address space CPI depends on the cycle counts in the SMF 30 records
- Already mentioned the variability problem
- One thought has been “ignore the obviously bad measurements”
  - But if some are obviously bad, are there others that are less obviously bad?
  - I think the answer has to be yes (in most systems)
- Even given the variability, address spaces/programs that *consistently* show higher than expected CPI during times of high SIIS% might be worth investigating

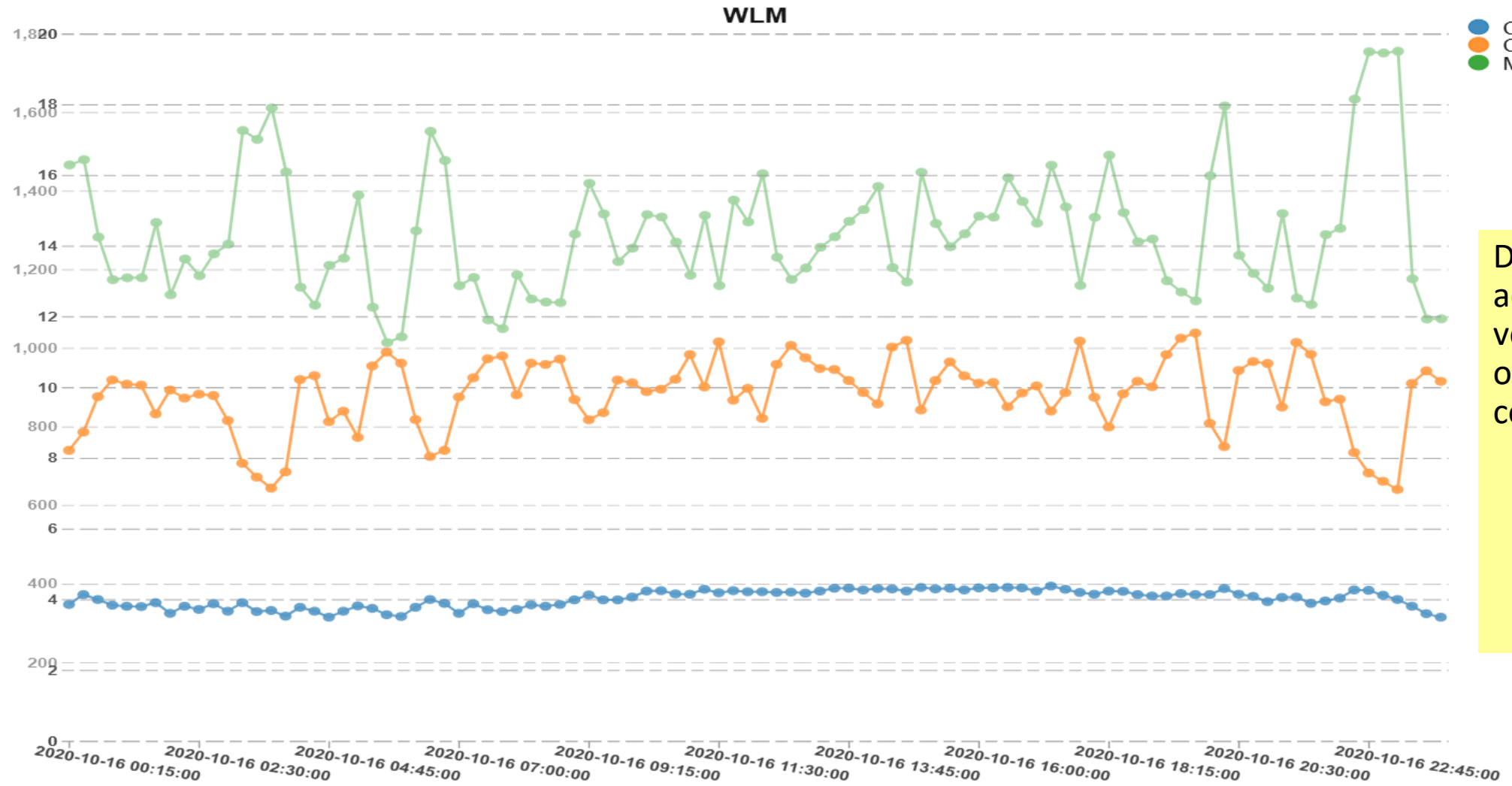
Address Space Instruction Details



Here RMFGAT's CPU time is definitely more consistent than its instruction count and there's no real obvious outliers in terms of the instruction counts to ignore.

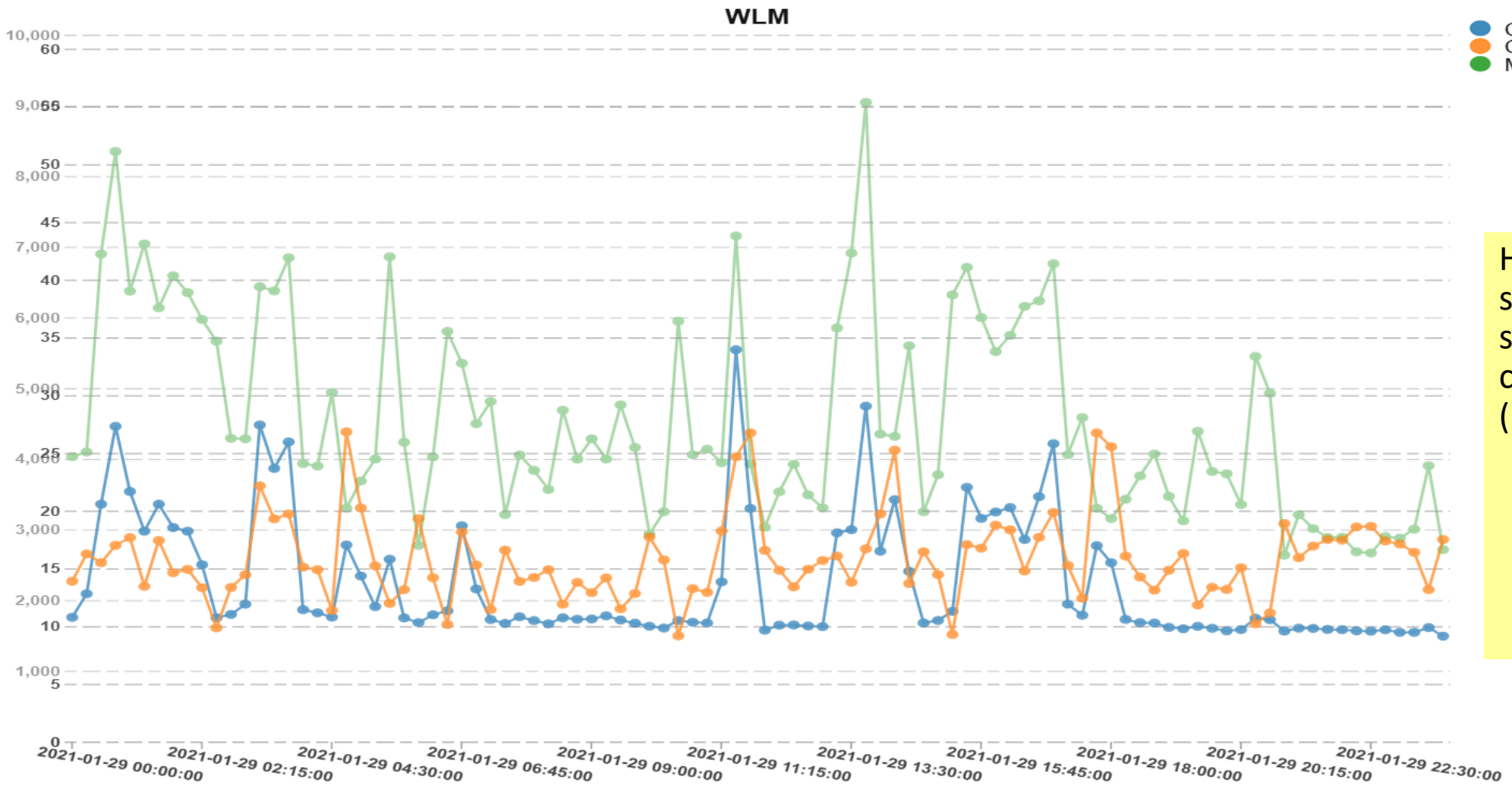
Note CPI varies inversely to instruction count because CPU time is pretty consistent.

Address Space Instruction Details



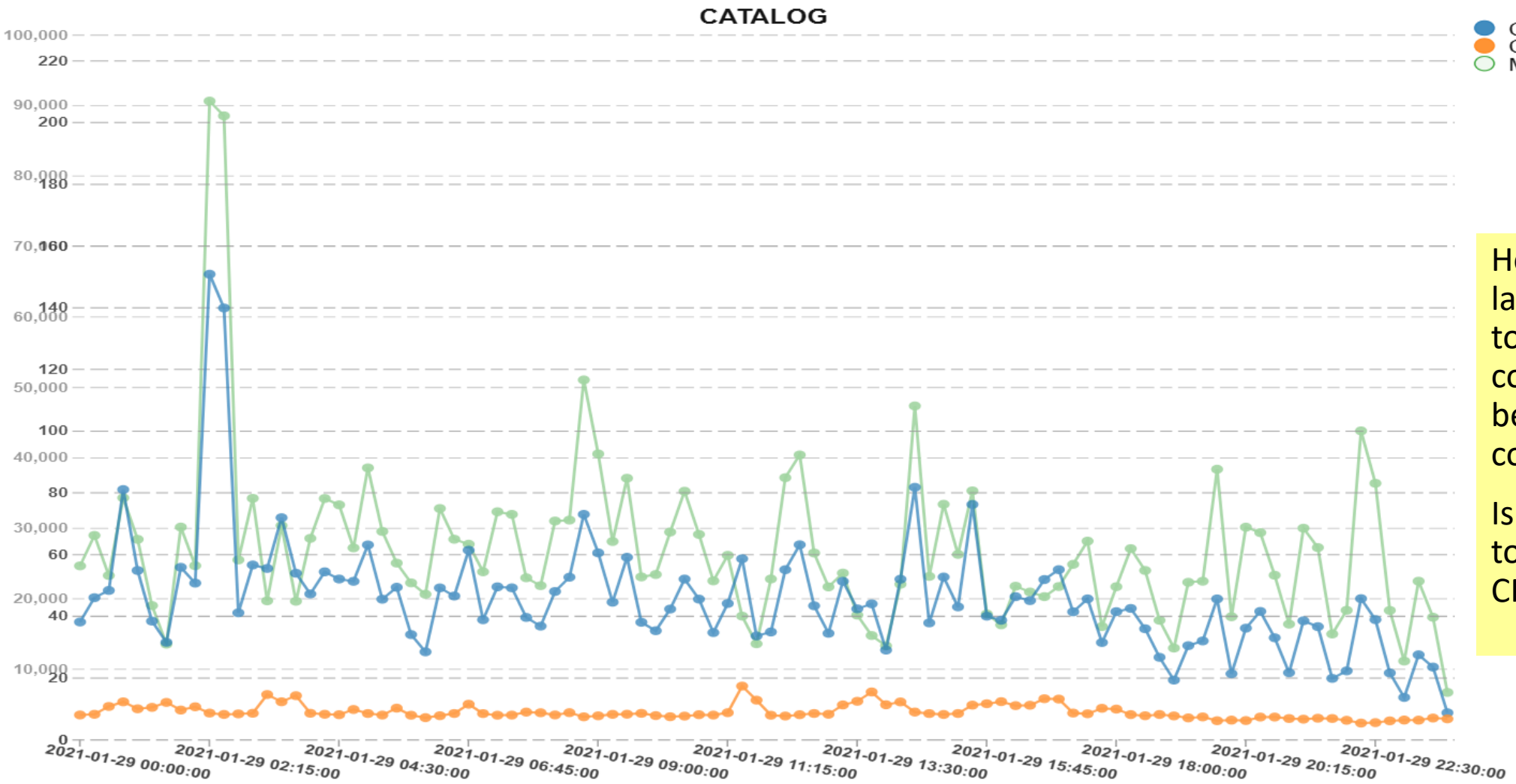
Different system, WLM address space consumes very consistent amount of CPU but instruction count varies significantly.

Address Space Instruction Details



Here's WLM on a larger system. There does seem to be better correlation here in most (but not all!) intervals.

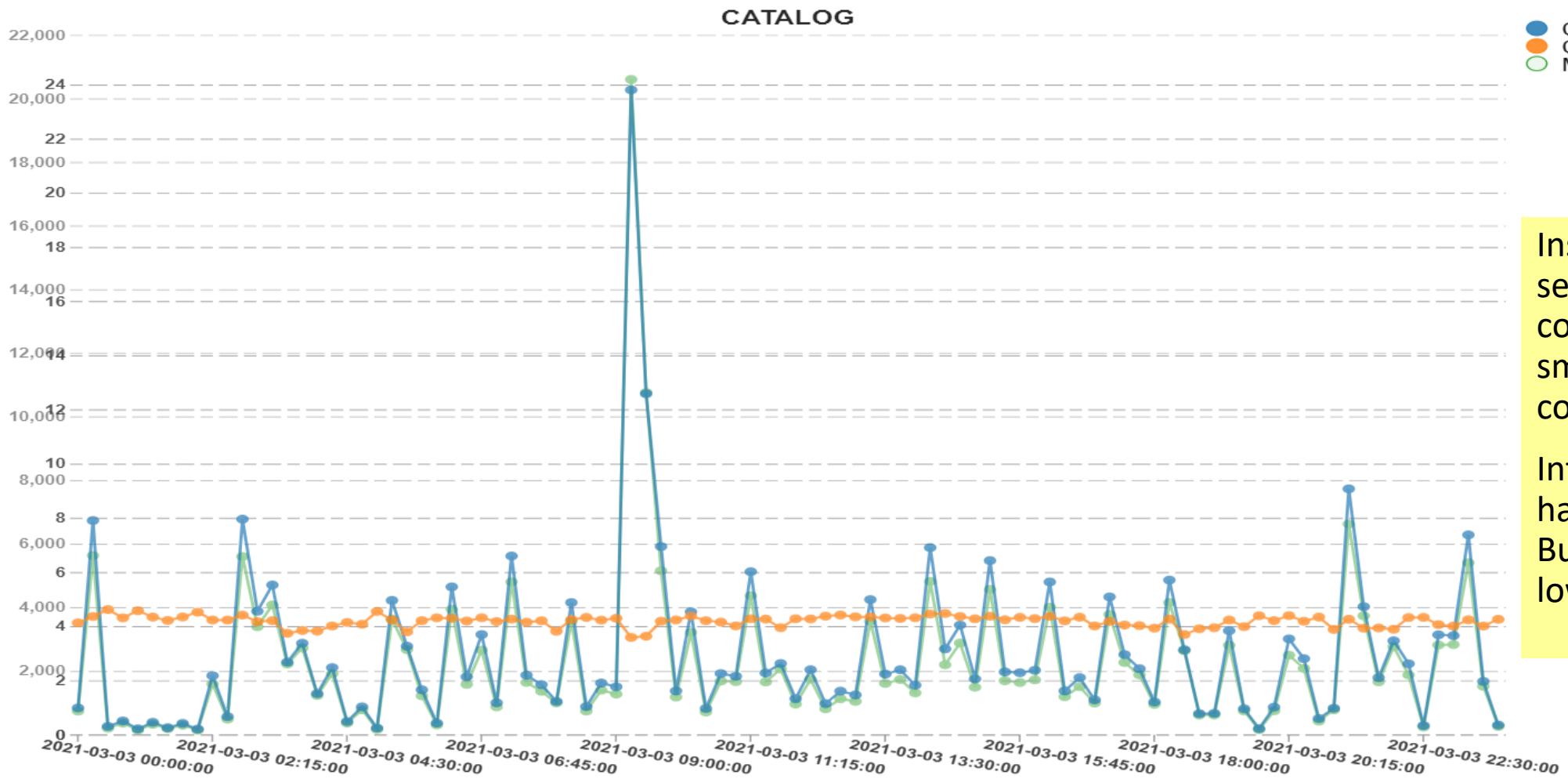
Address Space Instruction Details



Here's CATALOG on that larger system: it seems to have a more consistent relationship between instruction counts and CPU time.

Is this good luck or due to consuming more CPU?

Address Space Instruction Details



Instructions and CPU seconds extremely well correlated on this smaller system: best correlation I've found!

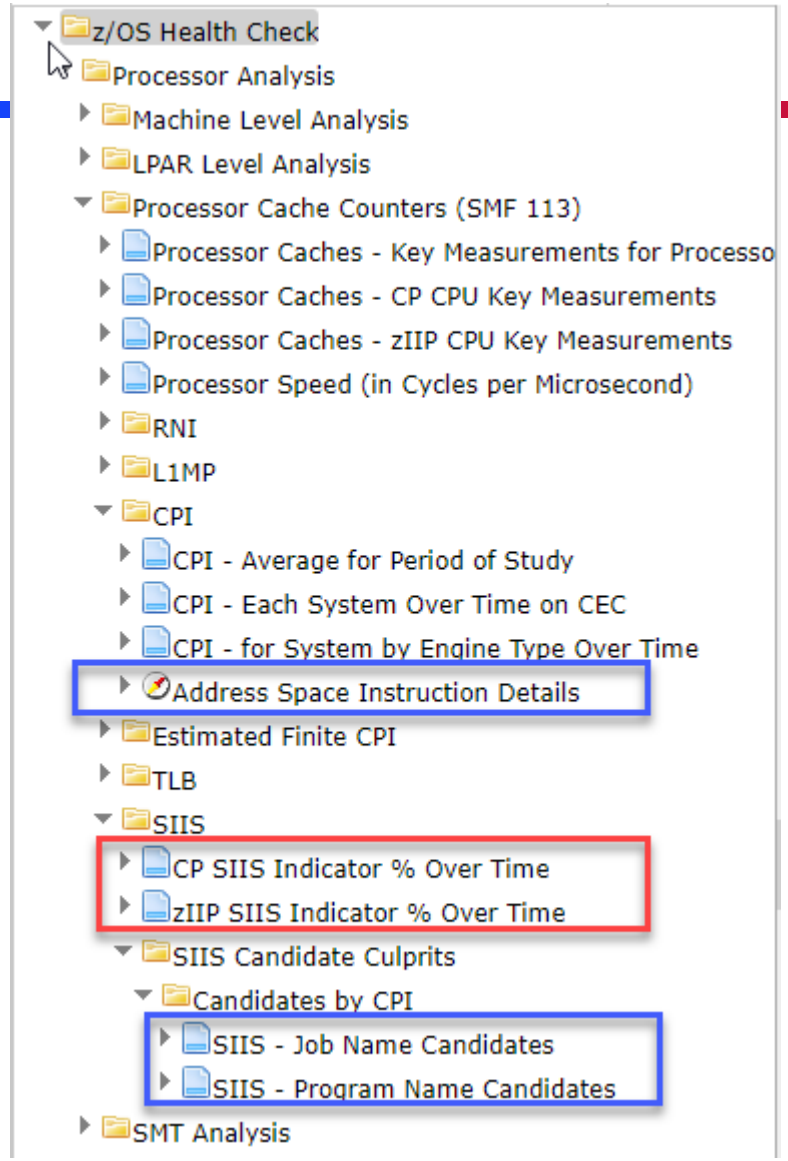
Interestingly(?) this LPAR had a single logical CP. But also had relatively low interrupt rate.

Conclusions

- Only worry about SIIS % Indicator in intervals with significant CPU usage
  - Also consider the type and diversity of work running in the interval
- Look first for what's running during those intervals
  - Customers without Assembler application code generally have few SIIS problems
    - Most ISV code has been fixed
- SMF 30 instruction counts *might* help you find the culprits that are driving up the SIIS % Indicator
  - SMF 30 instruction counts are variable to degrees that are not always obvious
  - LPARs with low I/O rates *may* be less susceptible to this issue
  - Don't assume high CPIs based on the SMF 30 data is indicative of a problem
  - If you aren't having a SIIS % problem, there still seems to be little need to turn on the SMF 30 instruction counts

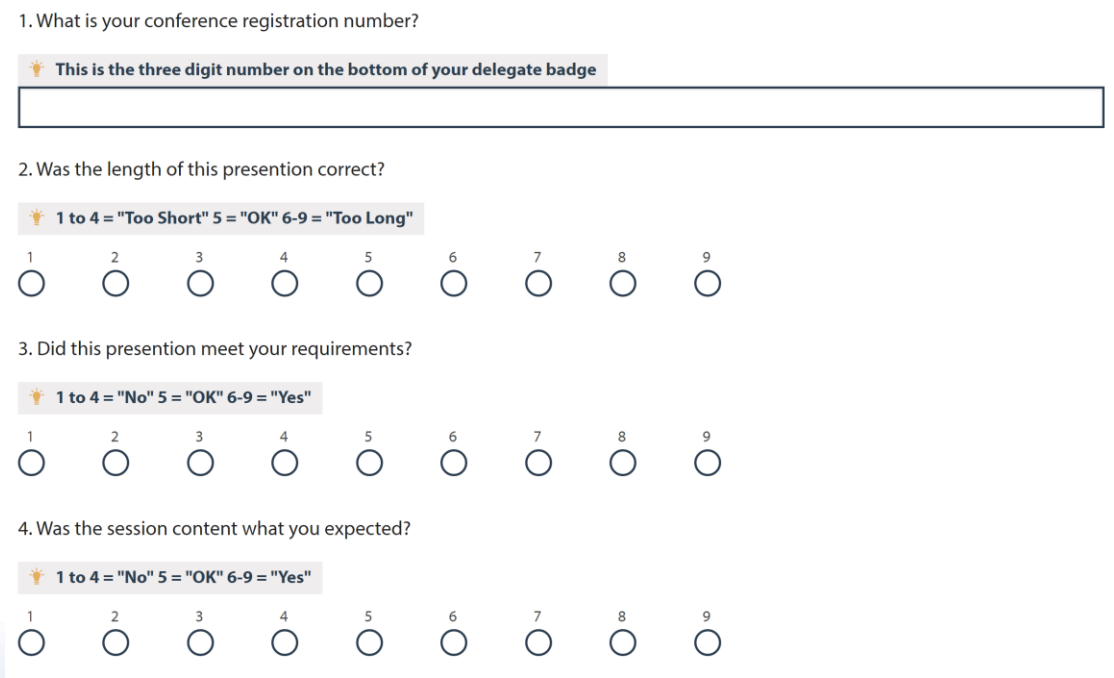
For Pivotor Customers…

- Find those new reports under the Processor Cache Counter section
- Note the reports outlined in blue are dependent on the SMF 30 instruction counts and won't appear if you don't have those enabled
- If you have any questions, don't hesitate to reach out!



Please submit your session feedback!

- Do it online at https://conferences.gse.org.uk/2021/feedback/6BH
- This session is 6BH



1. What is your conference registration number?

This is the three digit number on the bottom of your delegate badge

2. Was the length of this presention correct?

1 to 4 = "Too Short" 5 = "OK" 6-9 = "Too Long"

1 2 3 4 5 6 7 8 9

3. Did this presention meet your requirements?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

4. Was the session content what you expected?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

Become a member of GSE UK

- Company or individual membership available
- Benefits include:
  - GSE Annual Conference: Receive 5 free places + 2 free places for trainees
  - 20% discount on fees for IBM Technical Conferences
  - 20% on IBM Training Courses in Europe
  - 15% discount for IBM STG Technical Conferences in the USA
  - 20% discount on the fee for taking the Mainframe Technology Professional (MTP) exams
  - European events – via GSE HQ

- Contact membership@gse.org.uk for details

Thanks!

Questions/comments:
Scott.Chapman@epstrategies.com