Counting Instructions: Valuable Insights or More Noise?

Scott Chapman

Enterprise Performance Strategies

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

- Several years ago, IBM added an option to record instruction counts in the SMF 30 data. They almost immediately realized that the instruction counts were skewed by I/O interrupts and so the recommendation was to not use them. More recently, IBM has started to say that maybe you can derive useful information from the instruction counts if you just discard the bad measurements. Does this make sense? Are there insights to be gleaned from those instruction counts? Can you use them to find inefficiencies? Or does the natural variability introduce too much noise to make sense of any signal that may be lurking in the data? In this session, we will explore what we have discovered by looking through data from multiple customers that have enabled the instruction counters in the SMF 30 data.

EPS: We do z/OS performance…

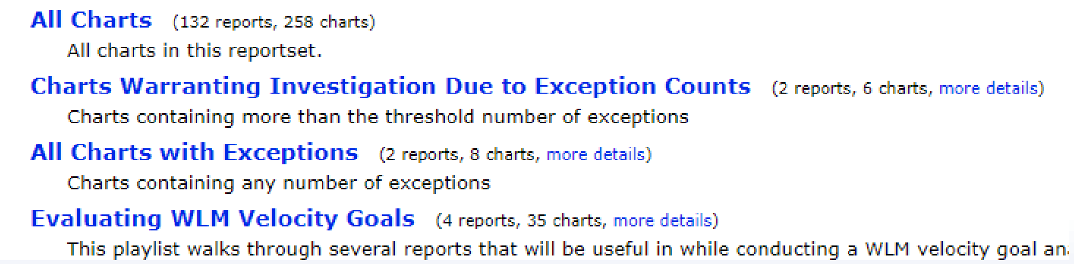
- Pivotor - Reporting and analysis software and services
  - Not just reporting, but analysis-based reporting based on our expertise
- Education and instruction
  - We have taught our z/OS performance workshops all over the world
- Consulting
  - Performance war rooms: concentrated, highly productive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

Like what you see?

- The z/OS Performance Graphs you see here come from Pivotor™ but should be in most of the major reporting products

- If not, or you just 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

- 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!



All Charts (132 reports, 258 charts)
All charts in this reportset.

Charts Warranting Investigation Due to Exception Counts (2 reports, 6 charts, more details)
Charts containing more than the threshold number of exceptions

All Charts with Exceptions (2 reports, 8 charts, more details)
Charts containing any number of exceptions

Evaluating WLM Velocity Goals (4 reports, 35 charts, more details)
This playlist walks through several reports that will be useful in while conducting a WLM velocity goal an

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!

What instruction counts are we talking abou?

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?!?

In SMFPRMxx:

~~SMF30COUNT~~

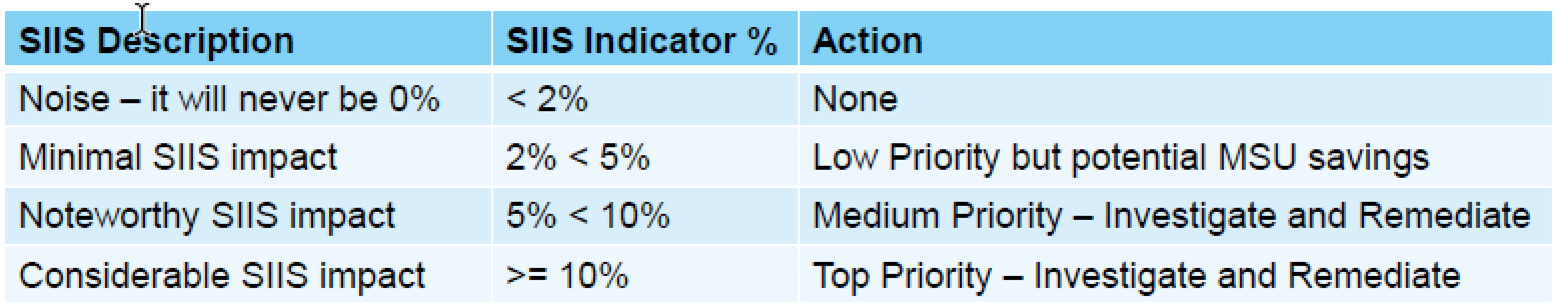
Why now??? SIIS

- 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
  - This can be a significant performance hit!
    - Pipeline has to be flushed
  - Obviously doing this once is not a noticeable problem, but doing it repeatedly can be
- Almost always this is triggered out of Assembler Code, not HLL
- Relatively recently IBM came up with a formula to give an indication of the relative impact of SIIS
  - Threshold for action at 5% with 10% said to indicate “likely significant impact”
- How do you find the culprit(s)???

Store Into Instruction Stream

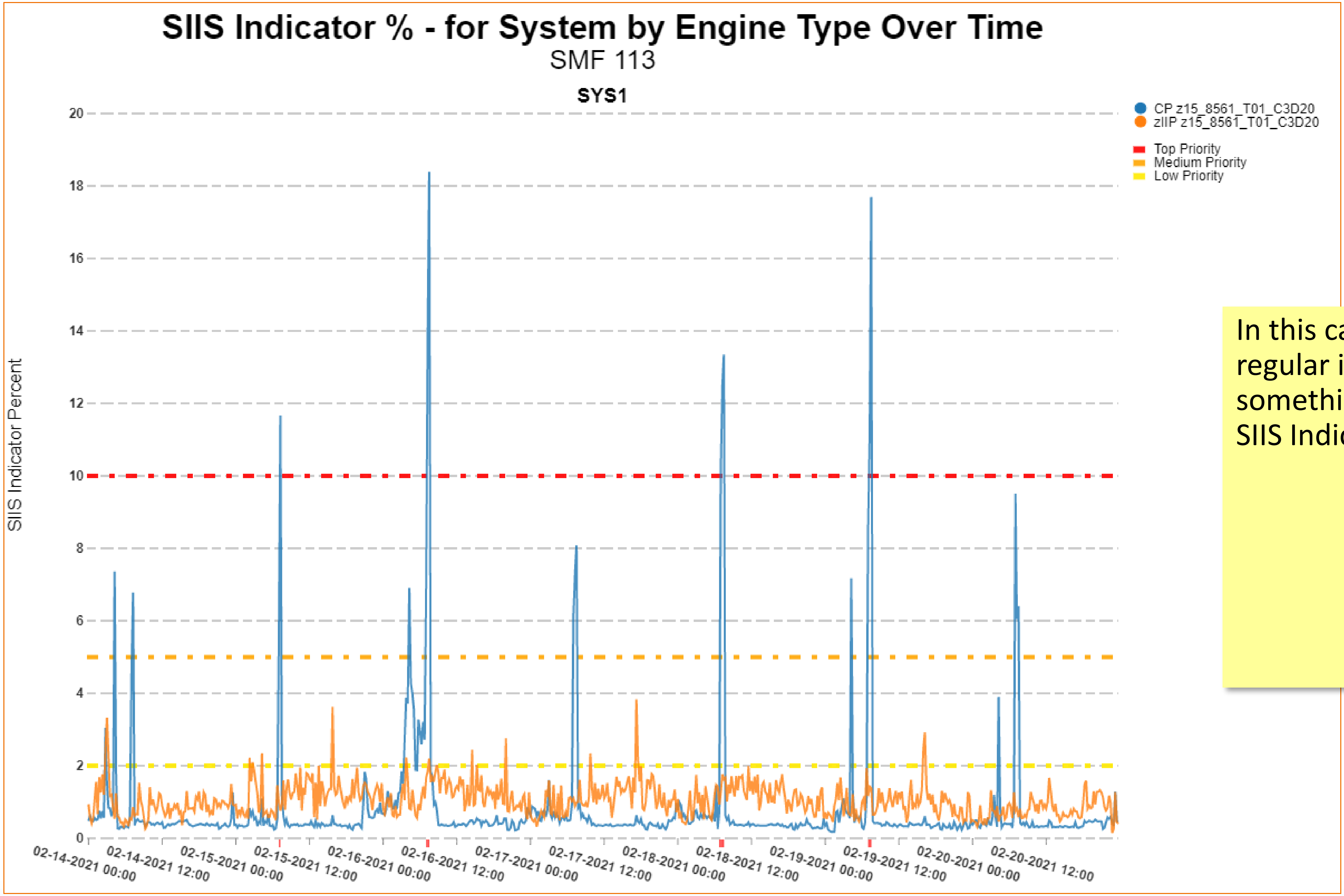
SIIS Details

- 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

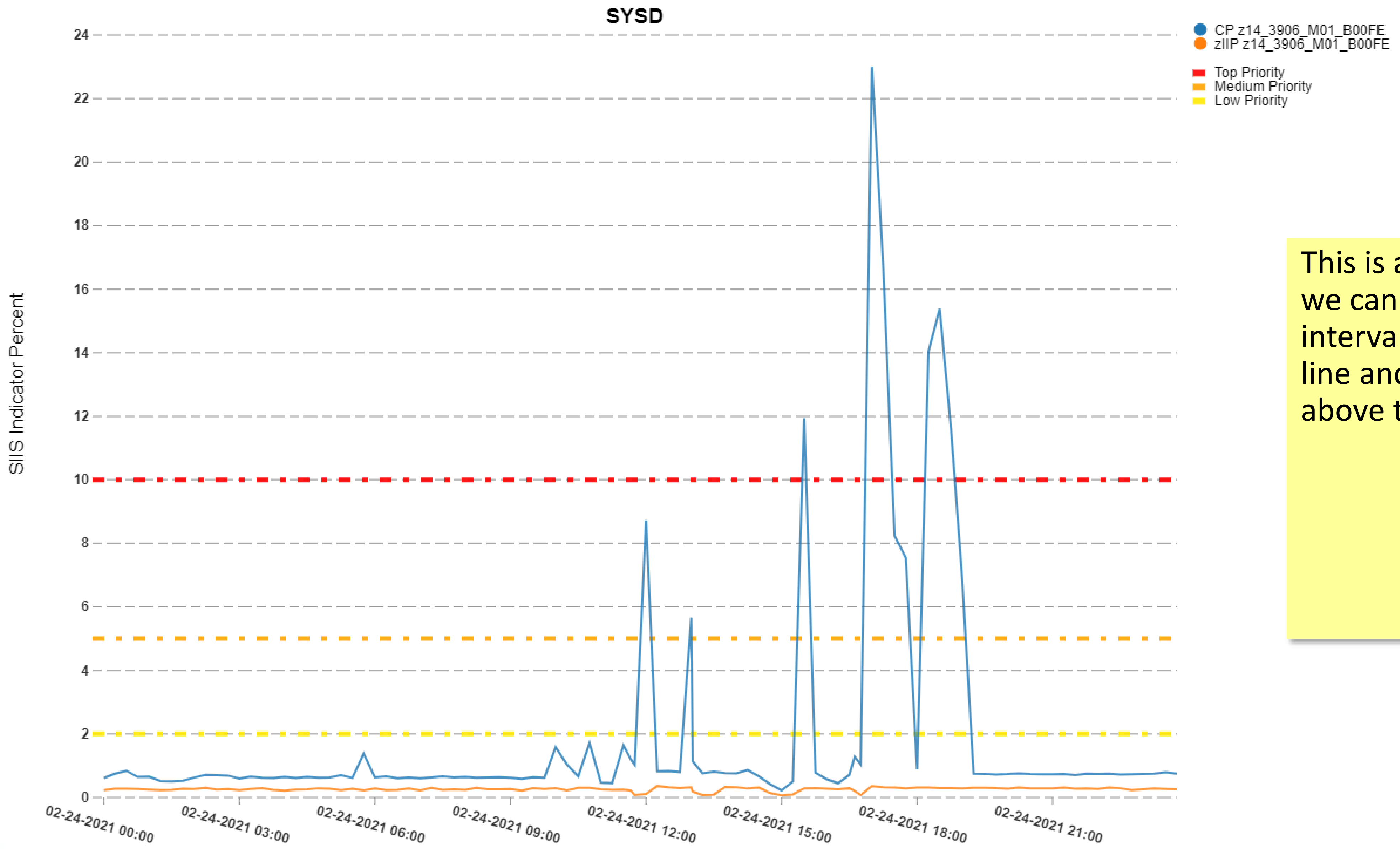
- Note that this doesn’t estimate the real savings
  - In some cases that can be significant
  - In some cases you can have high SIIS % during low-utilization times



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

SIIS Indicator % - for System by Engine Type Over Time

SMF 113

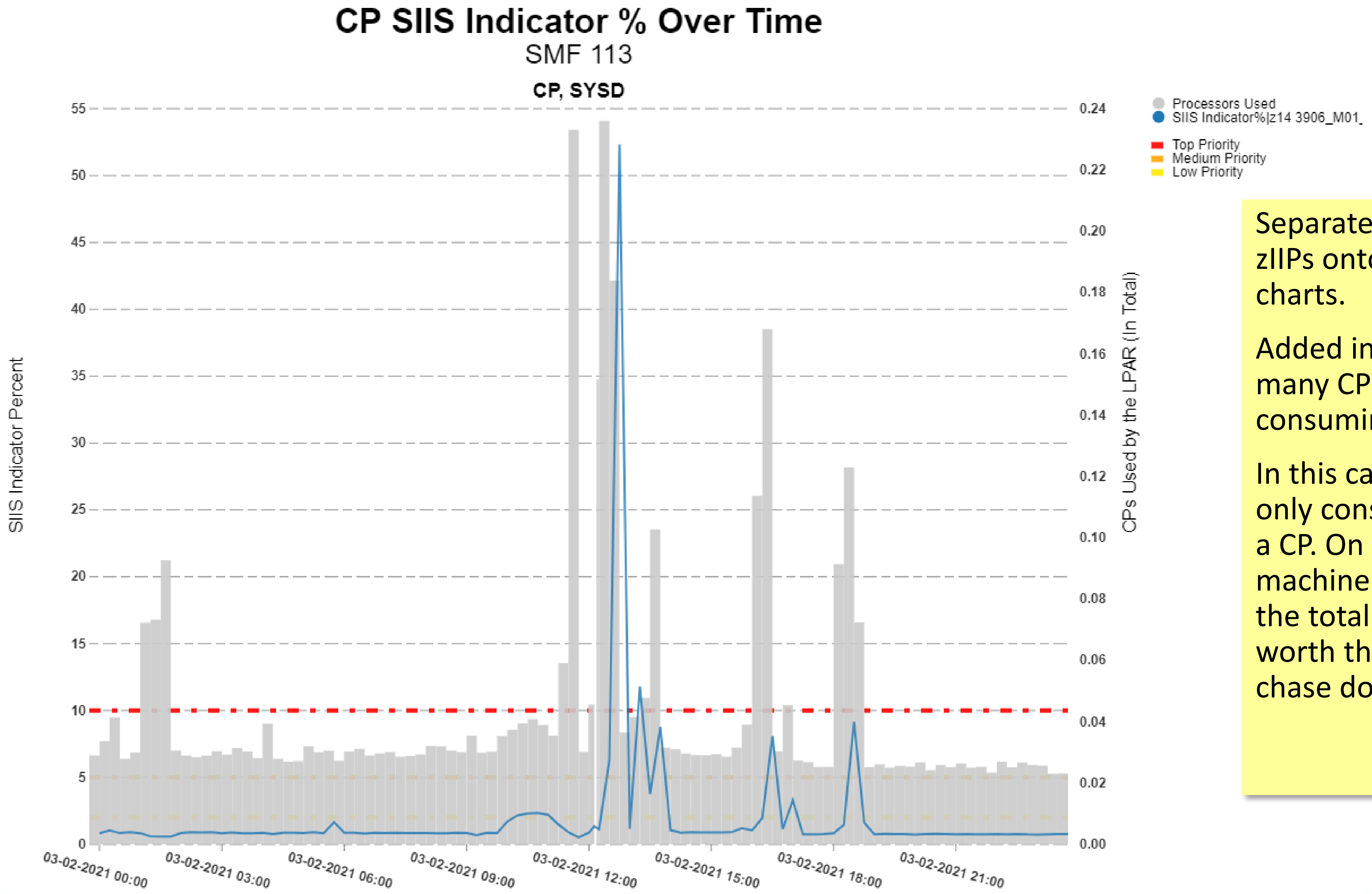


This is a single day but we can see multiple intervals above the 10% line and some more above the 5% line

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
- 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)

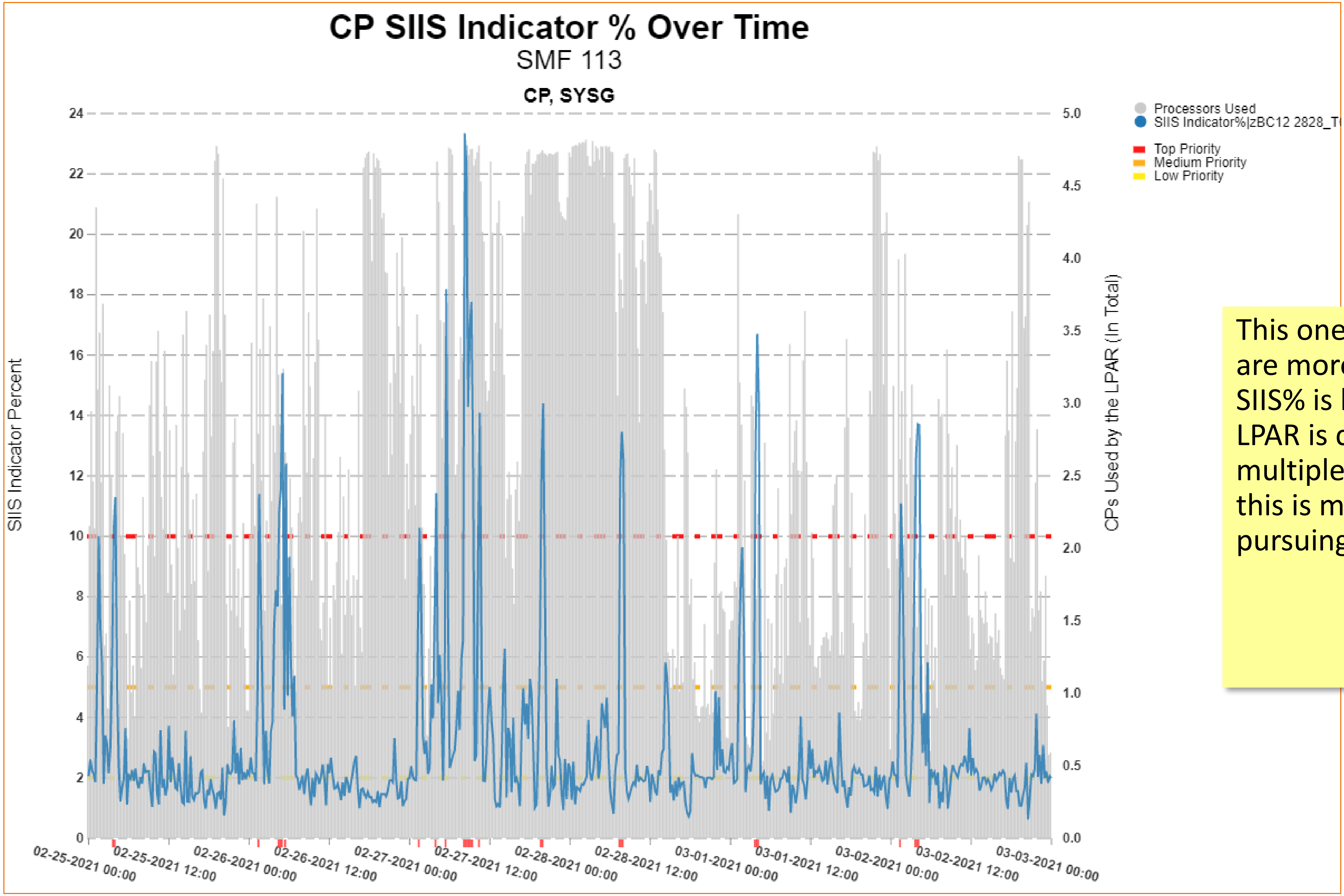
- So I made some reporting changes…



Separated out CPs and zIIPs onto their own charts.

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

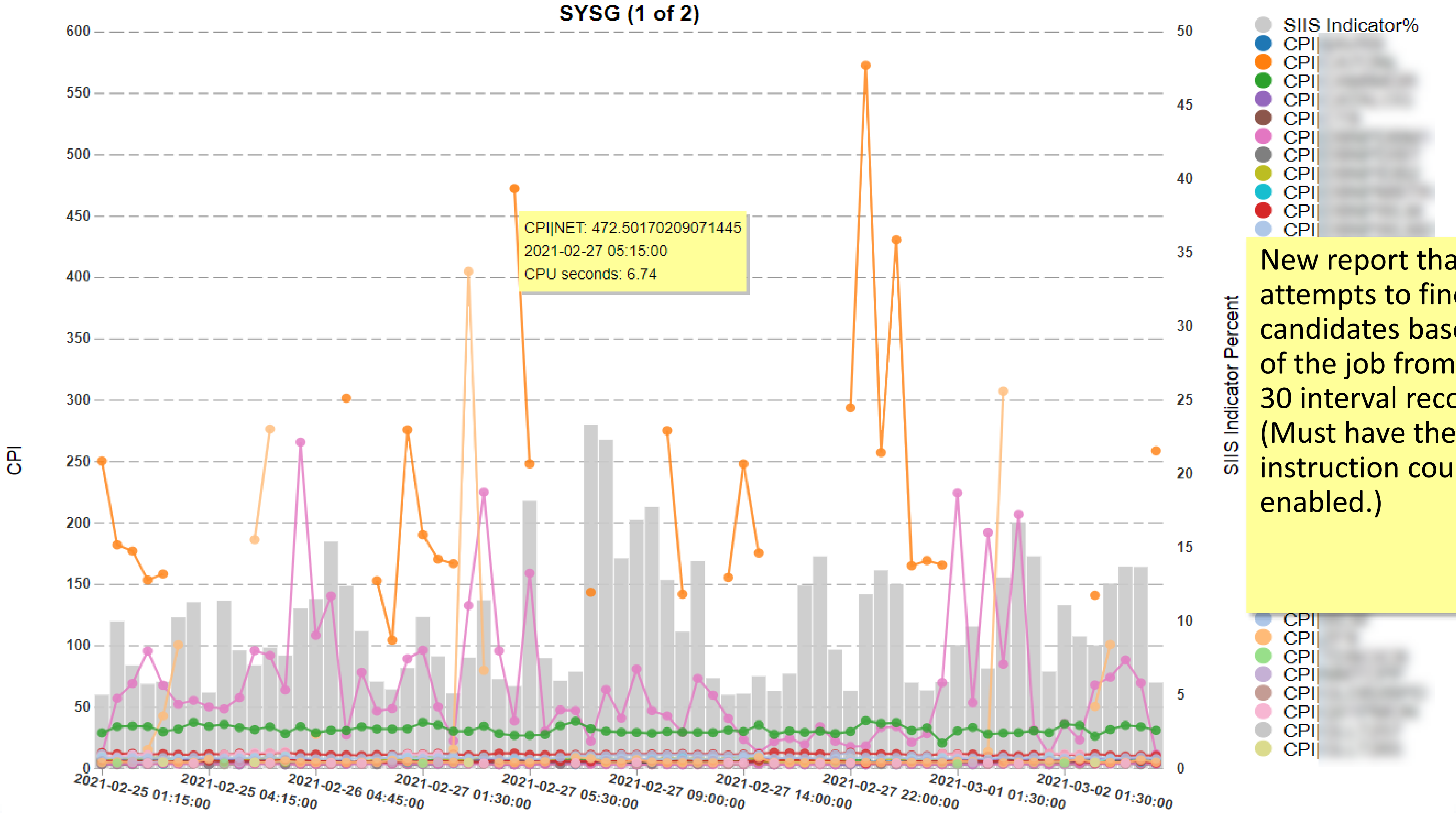
In this case the LPAR is only consuming <20% of a CP. On a 6-way machine. So about 3% of the total machine. Is it worth the trouble to chase down?



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.

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.)

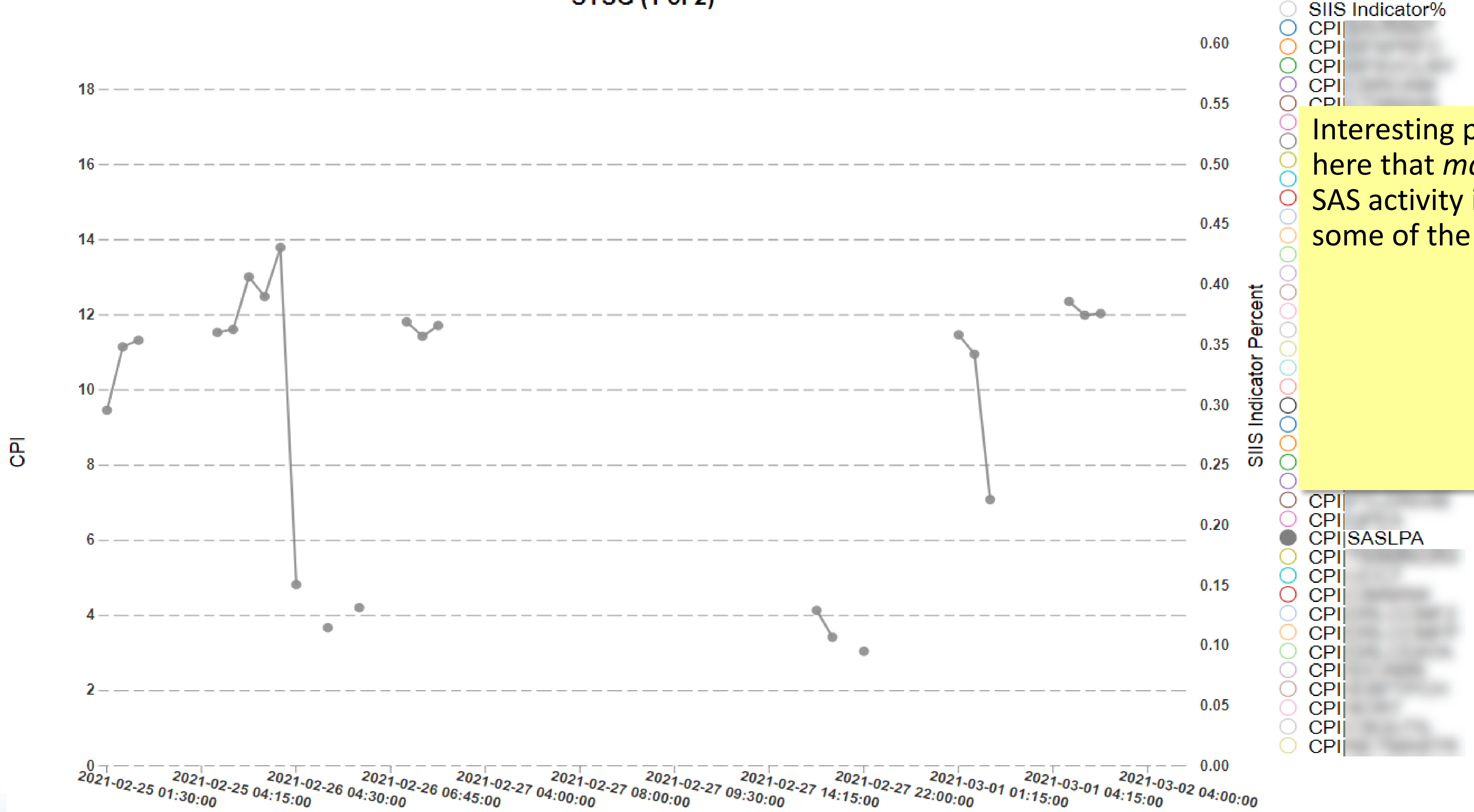
SIIS Culprits report

- 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)



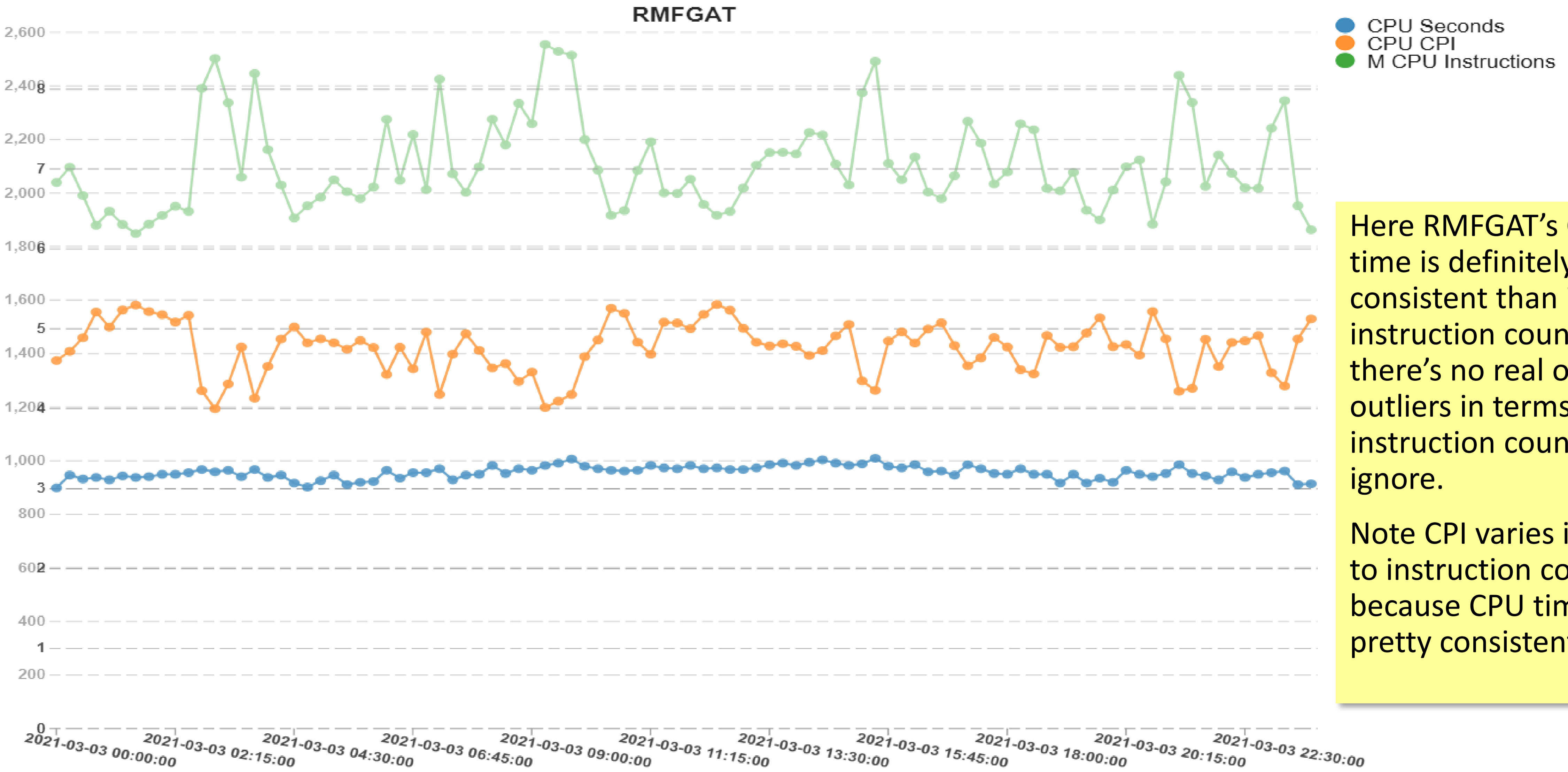
Interesting possibility here that *maybe* some SAS activity is driving some of the high SIIS %.

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
- Earlier we 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

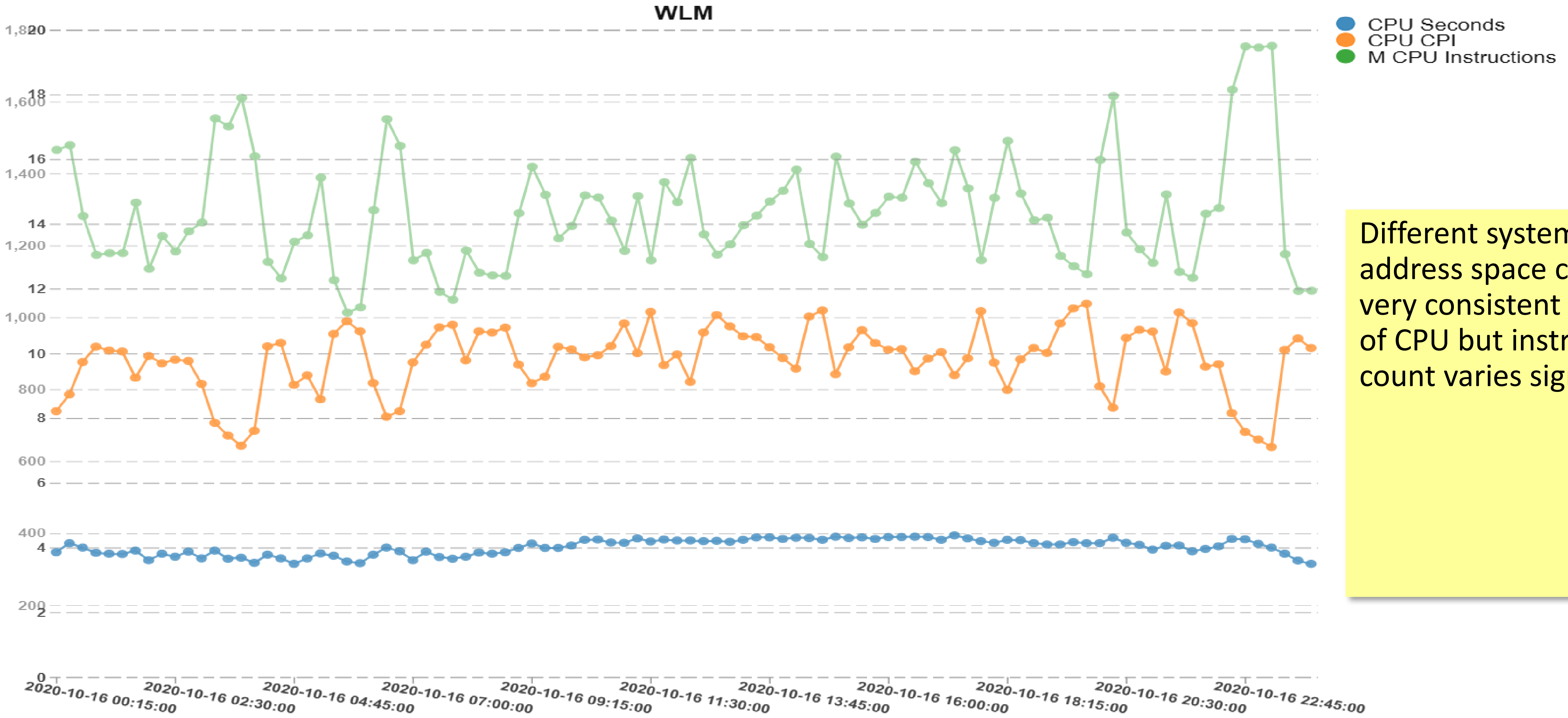
Address Space Instruction Details



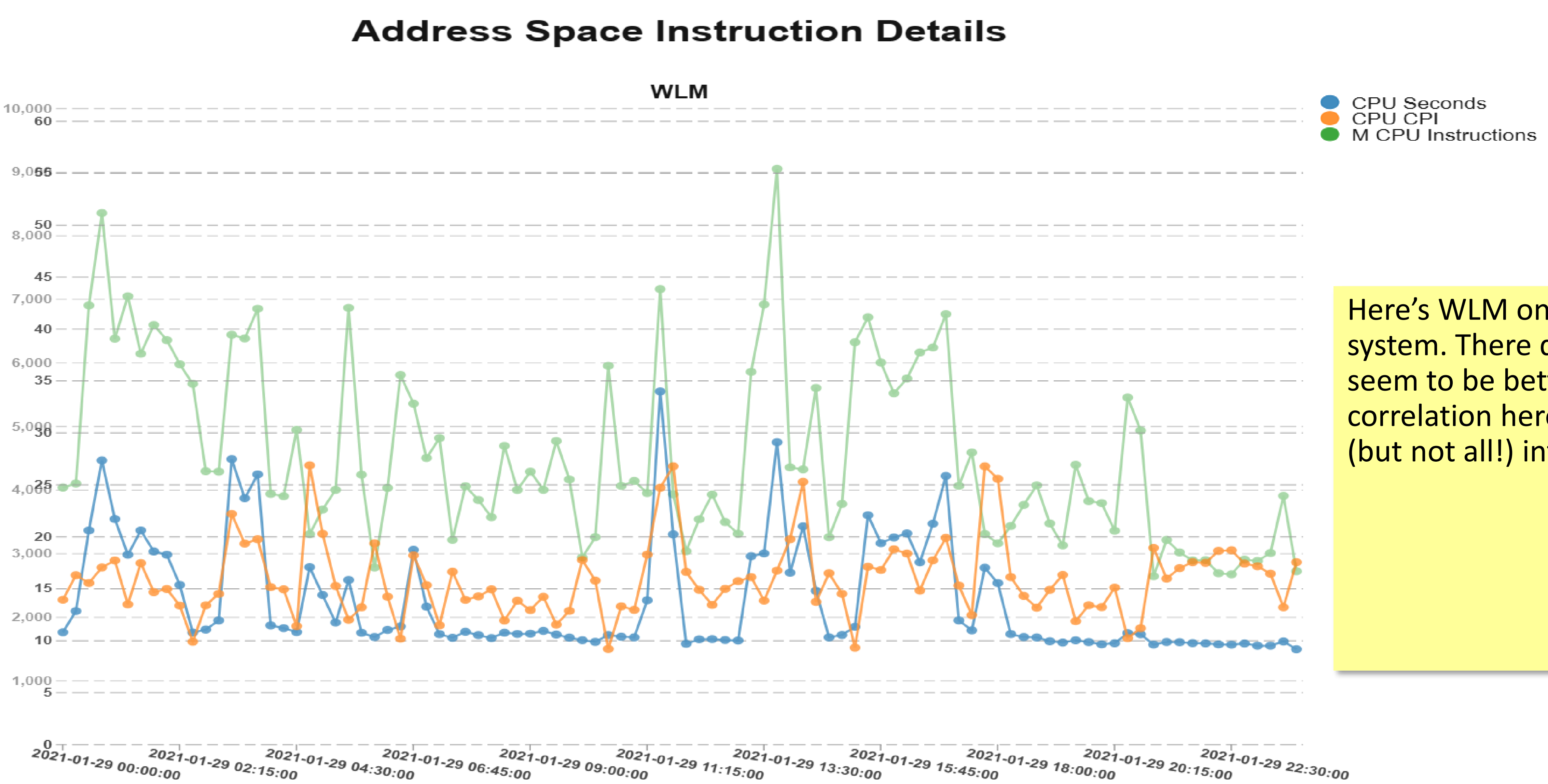
Here RMFGAT's CPU time is definitely more consistent than its instruction count and there's no real obvious outliers in terms of the instruction counts to ignore.

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

Address Space Instruction Details

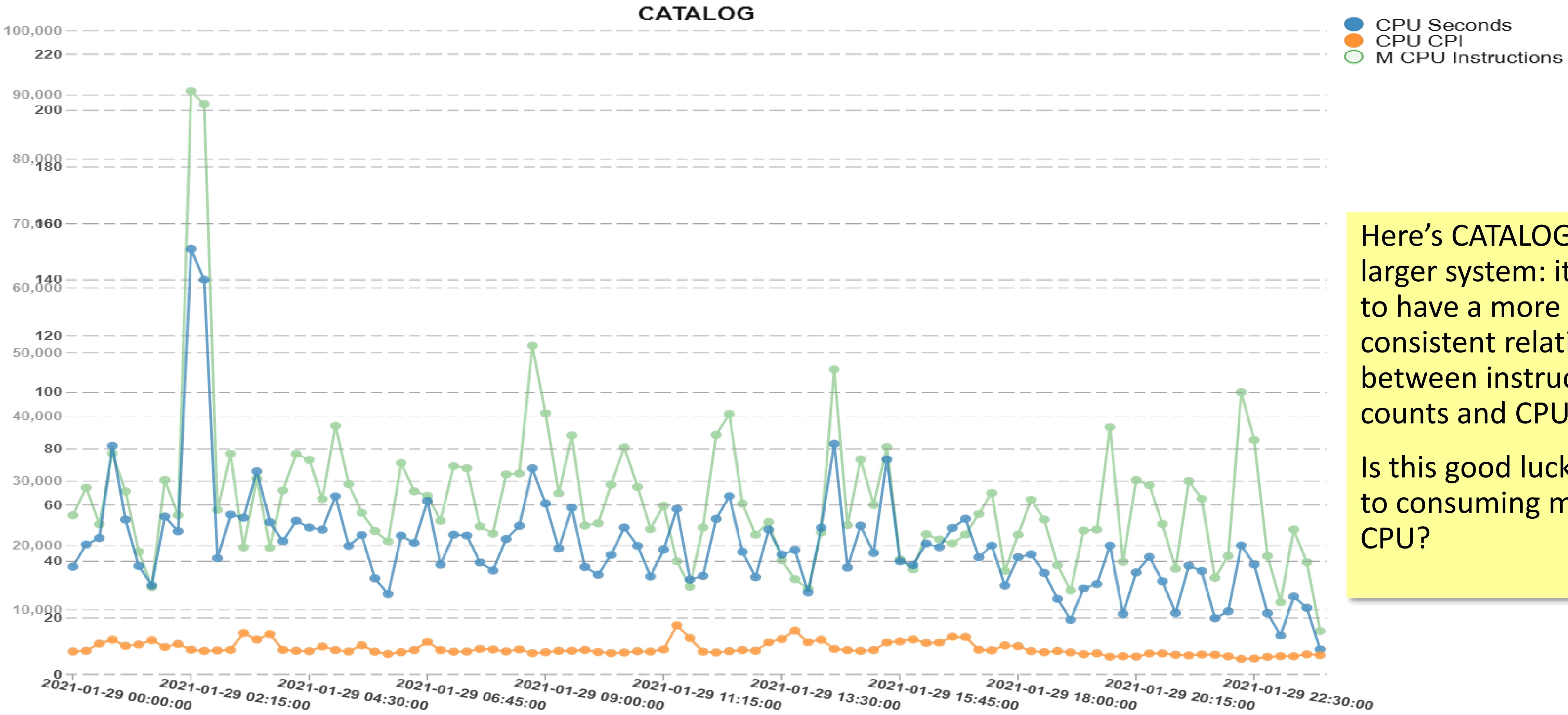


Different system, WLM address space consumes very consistent amount of CPU but instruction count varies significantly.



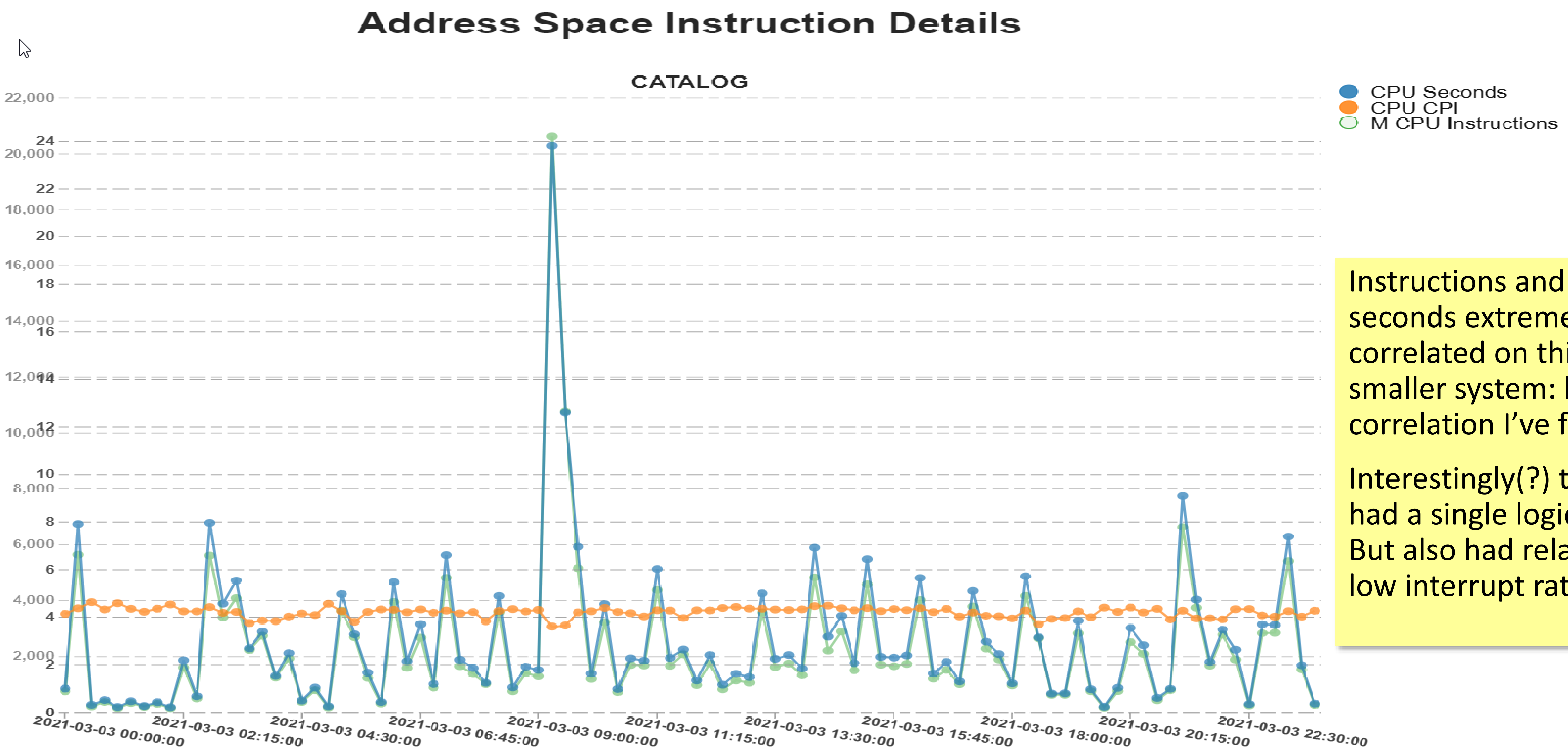
Here's WLM on a larger system. There does seem to be better correlation here in most (but not all!) intervals.

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?



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

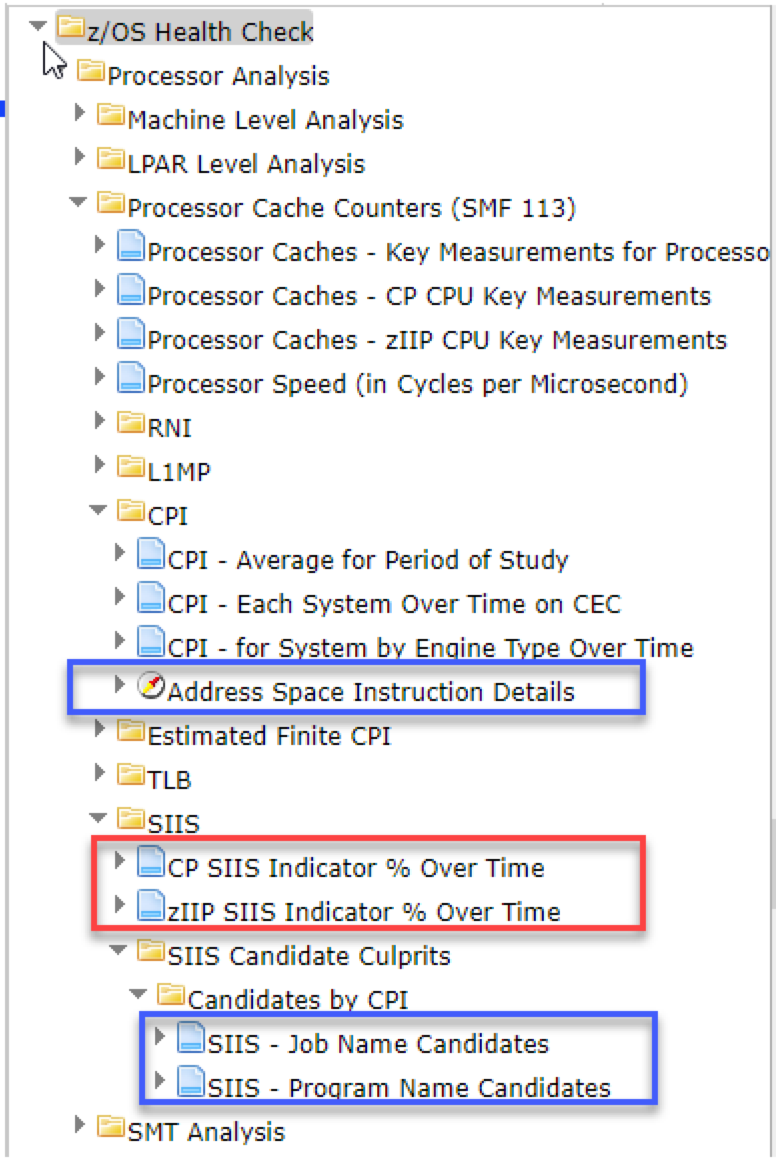
Interestingly(?) this LPAR had a single logical CP. But also had relatively low interrupt rate.

Conclusions

- Only worry about SIIS % Indicator in intervals with significant CPU usage
- 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
  - Drawing conclusions from the SMF 30 instruction counts should be done carefully
- If you aren’t having a SIIS % problem you need to track down, there seems to be little need to go turn on the SMF 30 instruction counts
  - If you have a proposed use for these numbers, I’d love to hear it!

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!



- z/OS Health Check
  - Processor Analysis
    - Machine Level Analysis
    - LPAR Level Analysis
    - Processor Cache Counters (SMF 113)
      - Processor Caches - Key Measurements for Processo
      - Processor Caches - CP CPU Key Measurements
      - Processor Caches - zIIP CPU Key Measurements
      - Processor Speed (in Cycles per Microsecond)
      - RNI
      - L1MP
      - CPI
        - CPI - Average for Period of Study
        - CPI - Each System Over Time on CEC
        - CPI - for System by Engine Type Over Time
        - Address Space Instruction Details
      - Estimated Finite CPI
      - TLB
      - SIIS
        - CP SIIS Indicator % Over Time
        - zIIP SIIS Indicator % Over Time
        - SIIS Candidate Culprits
          - Candidates by CPI
            - SIIS - Job Name Candidates
            - SIIS - Program Name Candidates
    - SMT Analysis

Questions??