



Exploring z/OS Processor Storage Measurements

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



Today, not all, but a great many z/OS environments have plenty of processor memory (i.e., storage). Whether your systems are storage rich, or teetering on the edge of paging, it is always important to occasionally look in on your processor storage usage. In this webinar, Scott Chapman will explore some key processor storage measurements to help you understand what types of storage frames your systems have configured and are using. Also discussed will be some key measurements you should regularly monitor, regardless of if you are z/OS processor storage rich or poor.

Agenda



- Basics
- Some common things to watch out for
- Reminder for those of you still on 2.3

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
 - October 3-7, 2022
- WLM Performance and Re-evaluating Goals
 - September 12-16, 2022
- Parallel Sysplex and z/OS Performance Tuning
 - August 8-12, 2022
- Also... please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

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!

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.

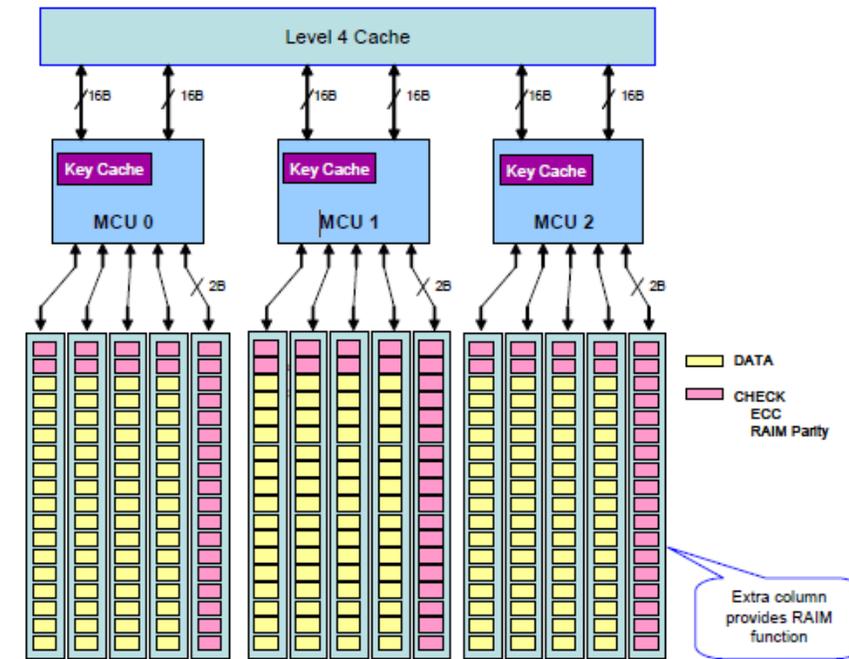


z/OS Memory Basics

Memory Introduction



- Just to confuse things, mainframers sometimes call memory “storage”
- Memory is expensive compared to other platforms
 - But getting much less so: 3x upgrade deal on z13/z14
 - LPARs with 100s of GBs are no longer unusual
 - Minimum orderable z14 memory is 256 GB
 - Minimum orderable z15 memory is 512 GB
 - Both: minimum granularity is 64 GB
 - Minimum z/OS 2.3 LPAR is 8 GB
- Memory is rigorously protected
 - Main memory, L4, and L3 all protected by ECC
 - Main memory is RAIM: Redundant Array of Independent Memory
 - Error detection and correction at sockets, channels, or DIMM level
 - Supposedly no disruptive memory failure since z196 (z11)



Virtual Memory, Not Logical



- Can't assign more online memory to LPARs than the CEC has (PR/SM)
 - However, you can have more reserved memory defined than is available
 - Reserved storage not online to the LPAR but can be brought online
 - Useful if you find the LPAR short of storage and need a quick fix
 - Consider holding some memory back as reserved for that reason
 - And consider setting IEASYSxx parameter RSU to "OFFLINE" to keep fixed pages out of the reserved storage units so you can take it offline again (maybe)
 - Generally RSU=OFFLINE or RSU=0 is most performant as z/OS doesn't have to work as hard to manage fixed pages
- But z/OS address spaces can use more memory than the LPAR has
 - Each address space gets its own virtual address range
 - System can page out non-fixed memory as demand increases
 - Paging is a performance hit as well as a CPU overhead

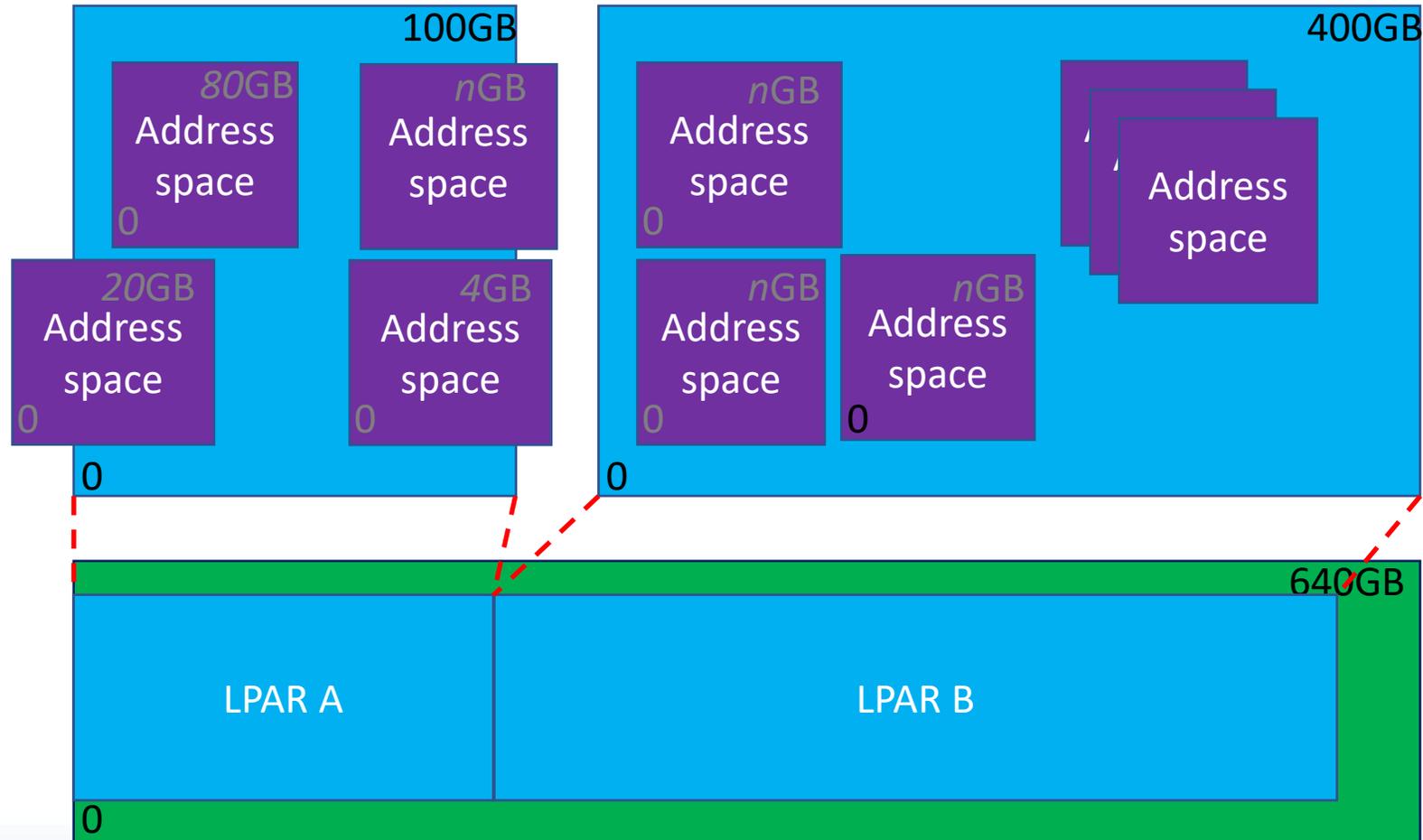
LPARS use
Real Memory

Address Spaces use
Virtual Memory

Memory address translation



Dynamic Address Translation & Prefixing: Virtual -> Real -> Absolute



Page Sizes



- Default page size is 4 KB
- “Large” 1 MB pages available since z10 and z/OS 1.9
 - Since z12, 1 MB Pages can be pageable
- 2 GB pages available since z12 (also officially “large”)
 - Always fixed (never paged out)
- Fixed 1MB/2GB page area specified on LFAREA parm in IEASYSxx
 - 1MB Pageable large area (PLAREA) allocated automatically by system
 - If 1M paged out to disk, will lose their 1MB attribute (retained for SCM)
 - Other events can cause this “demotion” to 4K pages as well

Fixed storage threshold optimizations



- z/OS 2.2 optimized some settings for LPARs with large memory
- Default (set with MCCFXTPR) is that system should always have at least 20% of the storage pageable
 - This could become a problem with LPARs now being hundreds of GBs
 - In z/OS 2.2, for LPARs > 320GB, requirement is 64GB of pageable storage
 - So now on your 1TB LPAR you could have 900GB of page fixed BPs and not run into issues with IRA400E (which stops new address space creation)
- Similarly, RCCFXTT and RCFXET default to “Auto” on LPARs with more than 64GB to better manage the MPL
 - Shouldn't have to be as proactive about MPL management with large memory
 - Initially this default didn't work, but has since been fixed

Storage Areas (pre-z/OS 2.3)



Storage Map

1M/2G LFArea
Quad
PLArea
RSU
4K Pref
V=R

- LFArea – Fixed 1M/2G pages
- Quad frames – 1/8th of online storage
 - 4 contiguous (in real) 4K pages on particular boundary
- PLArea – Pageable 1M (1/8th of online – Quad)
 - 10.9375% of memory (sometimes erroneously said to be 1/8th)
 - Created if running on a zEC12 or later machine
- RSU – Reconfigurable Storage
 - Probably want this 0 or effectively 0 with “offline”
- 4K Pref – Preferred area for 4K frames
- V=R reserved from IEASYSxx REAL parm
 - Recommended to be 0

Storage Areas (z/OS 2.3)



Storage Map

2G LFArea
Combined 1M/4K
RSU
V=R

- LFArea – Fixed 2G pages only
 - LFAREA 1M parm now just specifies the maximum number of 1M fixed frames that are allowed to be allocated (don't need to manage this quite as carefully now)
- No PLArea
 - No arbitrary limit of 10.9375% for 1M pageable frames
 - Allows potentially many more pageable 1M frames
- 1M & 4K frames allocated from same area
 - Reducing pools avoids overhead of managing pools

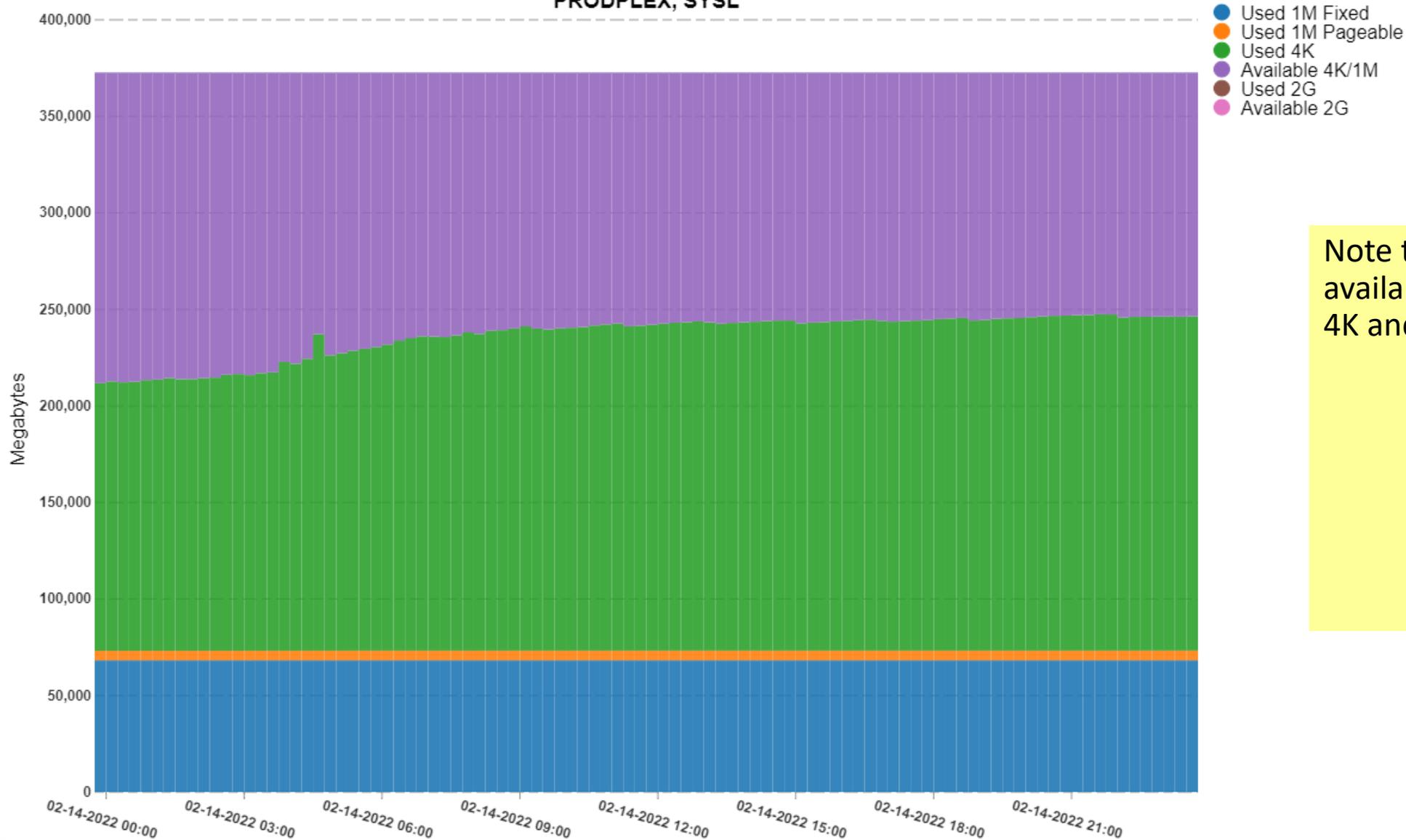
Pageable 1M allocations now recorded in SMF71PL* instead of SMF71L6* (* = M, X, or A)



Central Storage Areas by Page Size

Average Megabytes

PRODPLEX, SYSL



Note the purple available area is for both 4K and 1M frames



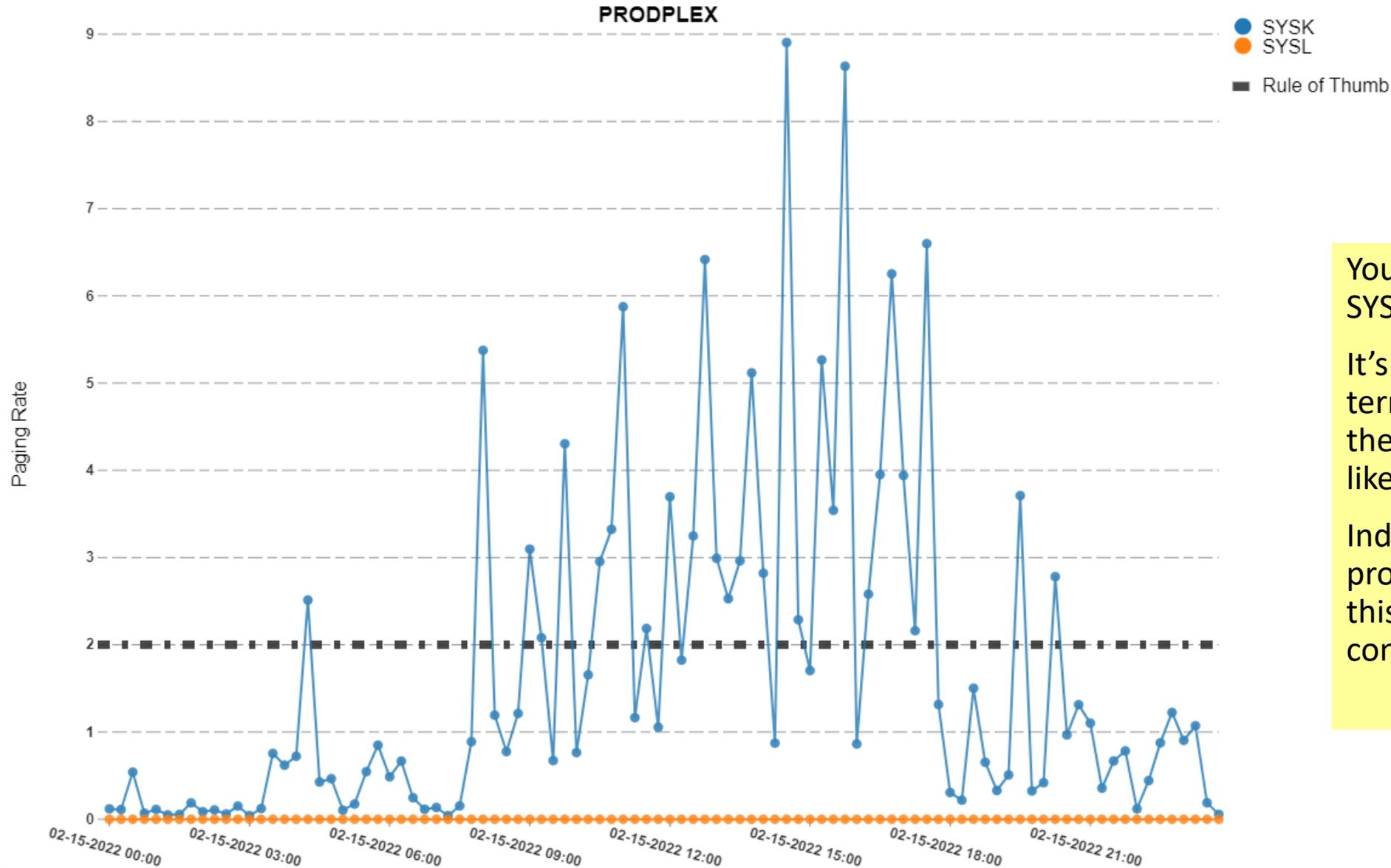
What do you need to keep an eye on?

Paging



- Today, the general expectation is that there should be almost zero paging
 - As well as being a performance problem, it costs CPU as well
 - Paging to Virtual Flash Express is less of a concern, but why not have more real memory?

Page-In Rate - All Systems



You want to be SYSL, not SYSK.

It's not like SYSK has a terrible problem, but... the expectation is really like SYSL.

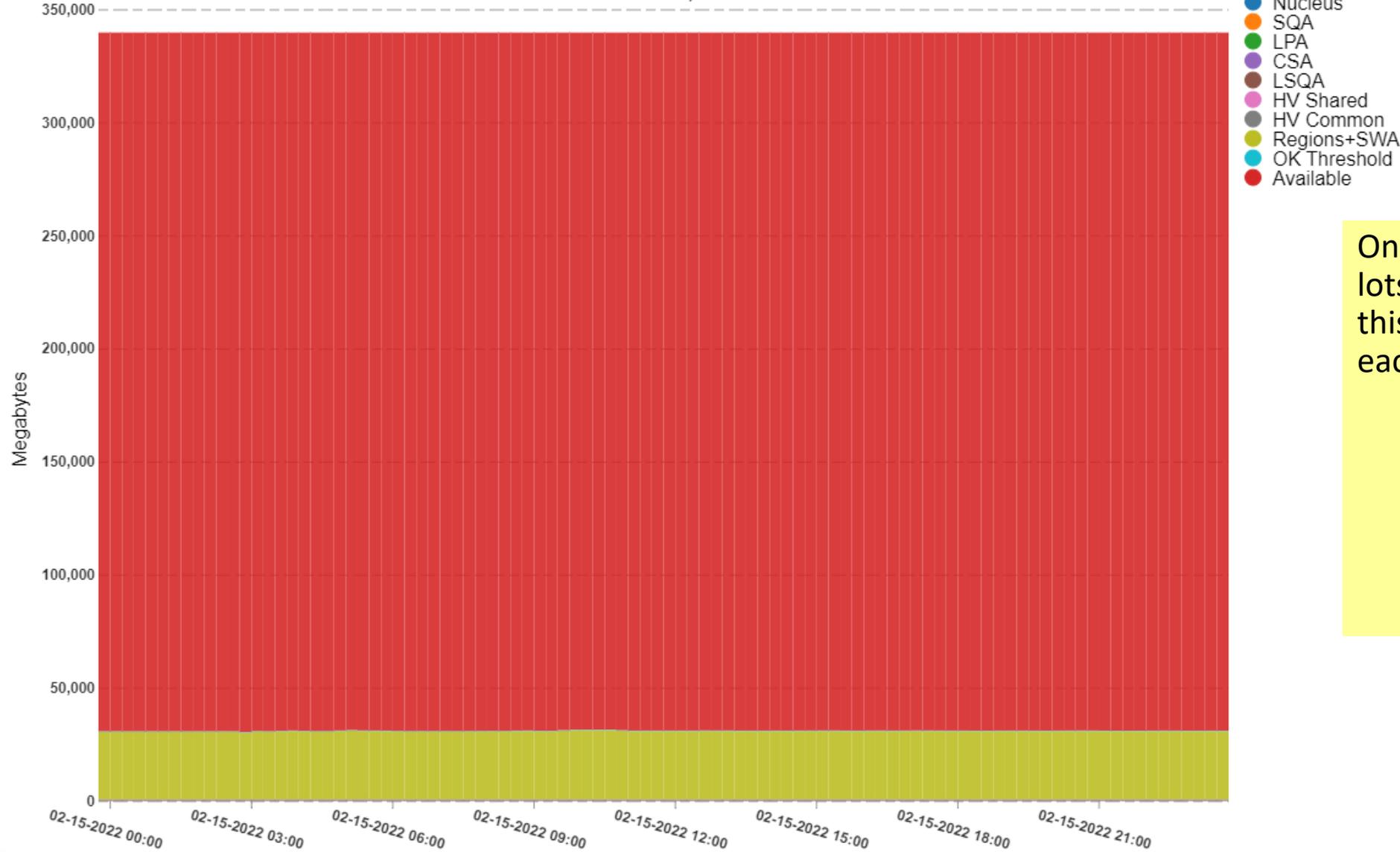
Individual spikes are probably just dumps, but this looks more consistent.



Central Storage Areas

Average Megabytes

PRODPLEX, SYSK



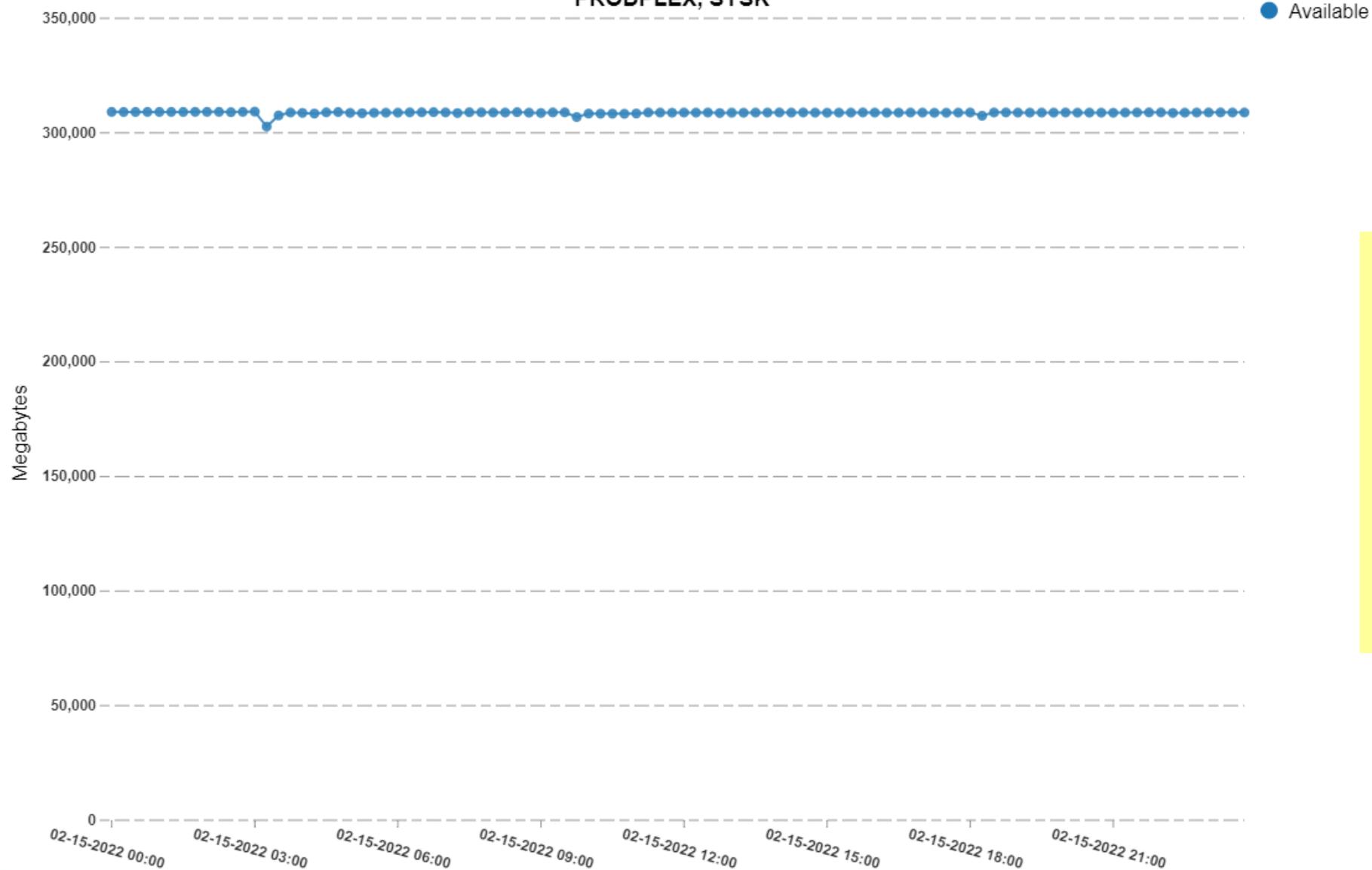
On average, SYSK has lots of free storage, but this is an average over each interval.



Minimum Available Central Storage

Megabytes

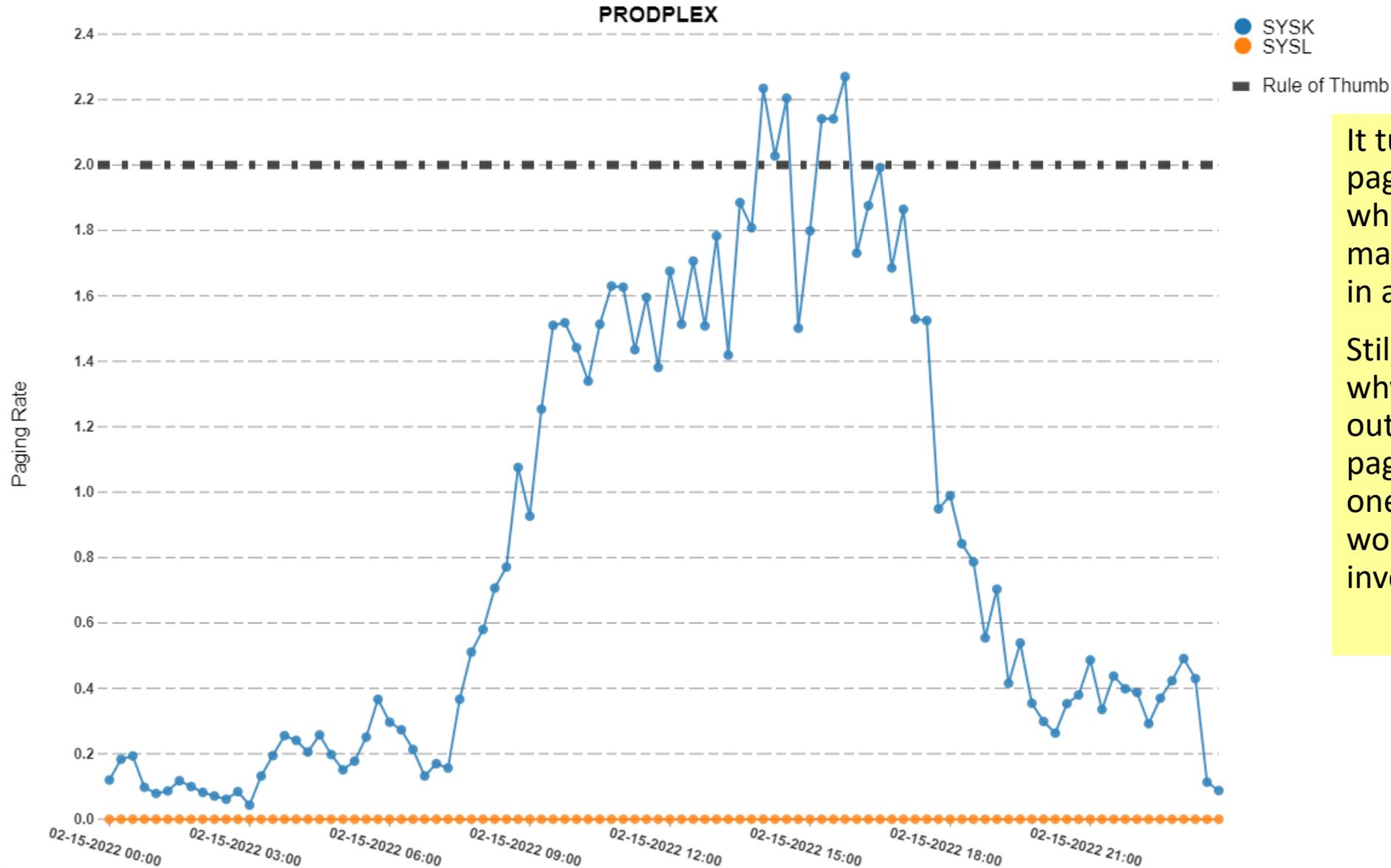
PRODPLEX, SYSK



Looking at the minimum available per interval is usually a better indication of running out of free memory as those events may be somewhat transitory.

But this leaves the question: why is SYSK paging?

Page-Out Rate - All Systems



It turns out that SYSK is paging data out. And what goes out very well may have to come back in at some point.

Still leaves the question why is it paging data out? In this case, the paging is all limited to one product, which would need to be investigated.

Large Pages



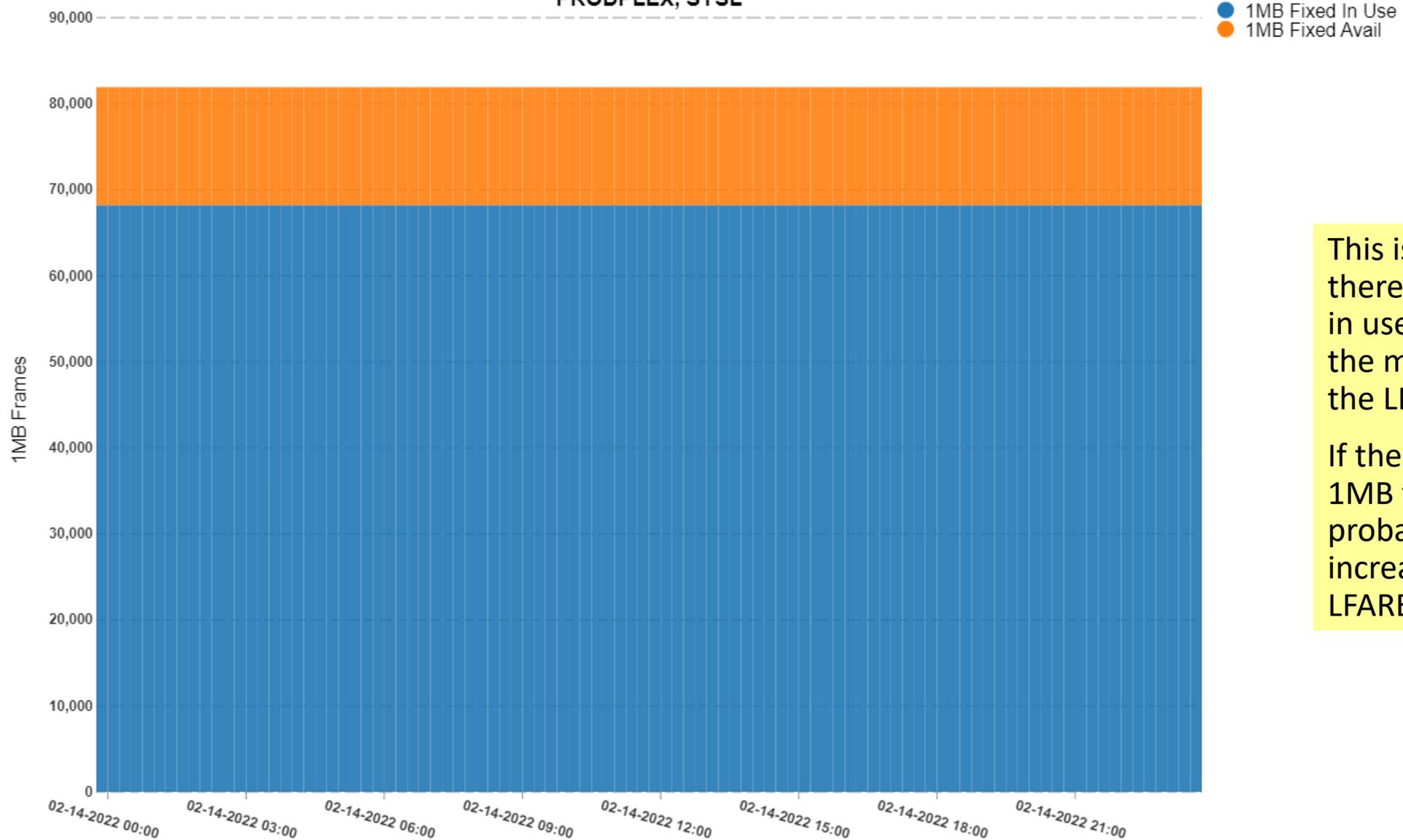
- Using 1MB and 2GB pages is good practice
 - Reduced TLB overhead (although this is minor on z14 and above)
 - Reduced pages for z/OS to track
- Primary candidates:
 - DB2 buffer pools
 - Always use fixed pages for production (and ideally for dev/test too)
 - IMS log buffers
 - Presumably should have these fixed too
 - Large JVM heaps
 - Be wary of boundary condition if using 2GB pages though (allocate heaps just below 2GB increments)
- We are seeing more customers use 2GB pages, but still relatively rare
 - But we also haven't seen any good reason to avoid them
 - If you have DB2 BPs that are 10s to 100s of GBs, why not?



Large 1MB Fixed Frame Averages

1MB Frames

PRODPLEX, SYSL



This is a good sign: there's 1MB fixed frames in use and they're not at the maximum defined in the LFAREA.

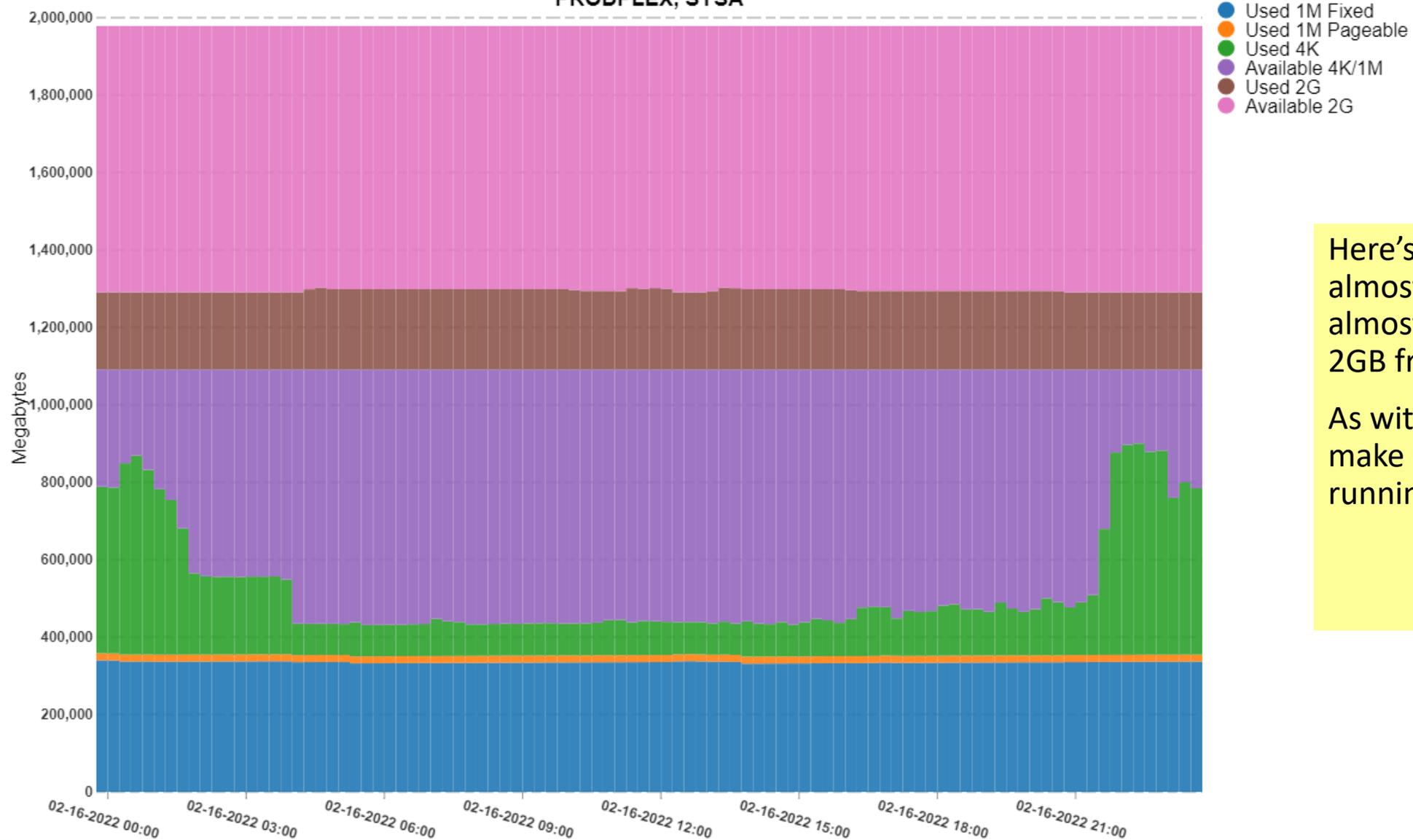
If there's no available 1MB fixed frames, then probably you need to increase the size of LFAREA.



Central Storage Areas by Page Size

Average Megabytes

PRODPLEX, SYSA



Here's an LPAR that has almost 2TB of memory, almost half of which is 2GB frames.

As with 1MB frames, make sure you're not running out.

Aux storage



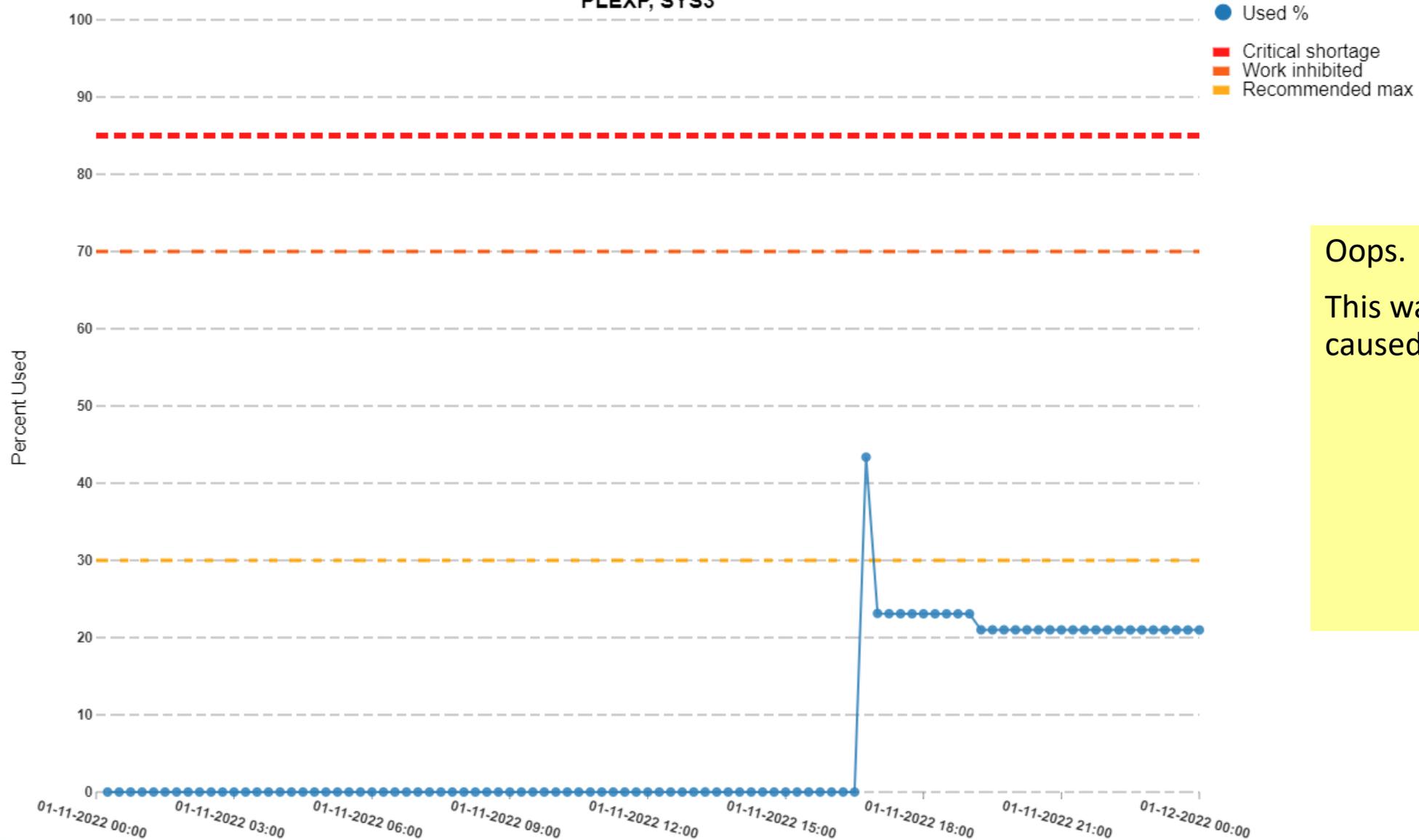
- Still should keep your aux storage below the 30% threshold
 - Should be easy if you're not paging!
 - We very rarely see an issue with this today
- Flash Express or Virtual Flash Memory give you lots of aux storage
 - Does cost money
 - Does help with system responsiveness when dumps occur
 - Does make you worry less about some paging, possibly particularly useful in large dev/test environments



Auxiliary Storage Utilization

Interval Average

PLEXP, SYS3



Oops.
This was probably caused by a dump.

Reminder: User key CSA goes away in 2.4

See also: “USERKEY COMMON Migration” at
<https://www.pivotor.com/content.html>

What is changing?



● User Key CSA

- The ability to have user key storage in CSA/ECSA
- This will no longer be allowed in z/OS 2.4
- If your programs are doing this
 - Then you either need to either change programs to have them not do this
 - Or use RUCSA

● RUCSA

- Restricted Use Common Service Area
- New function via APAR OA56180 added in z/OS V2.1, V2.2, and V2.3
 - <http://publibz.boulder.ibm.com/zoslib/pdf/OA56180.pdf>
- Free function in z/OS V2.1, V2.2, and V2.3
- **Priced feature if you continue to use in z/OS 2.4**
- IBM encourages you not to continue using RUCSA
 - But you may need to if you do not have the source code to stop using User Key CSA

What is changing?



● User Key Change Key

- Programs are changing common ESQA storage to use a user key (with CHANGKEY)
- Basically programs are attempting to CHANGKEY subpool 247 or 248 common storage to a user key (8 to 15)
- These programs must be fixed
- RUCSA will not help here

● User Key CADS

- CADS stands for 'common area data spaces'
- Programs are using user key SCOPE=COMMON data spaces
- These programs must be fixed
- RUCSA will not help here

Useful Measurements



- SMF 30 records
 - SMF30_USERKEYCOMMONAUDITENABLE
 - Enabled auditing to get the SMF 30 measurements
 - SMF30_USERKEYCSAUSAGE
 - Successful or unsuccessful attempts made to *obtain* user-key CSA or RUCSA storage for this step/job
 - SMF30_USERKEYCADSUSAGE
 - successful or unsuccessful attempts were made to *create* a user-key CADS for this step/job.
 - SMF30_USERKEYCHANGKEYUSAGE
 - successful or unsuccessful attempts were made to *change* the key of common ESQA storage to a user key (via CHANGKEY) for this step/job
 - SMF30_USERKEYRUCSAUSAGE
 - Successful or unsuccessful attempts were made to *obtain, reference, free, or change* the state of RUCSA storage for this step

Search Reports Exact

Titles
Tags
Values
All

Playlist 711/952 View Day 20220309 unzoom Hold No series Alter Chart Show All series Report Help

z/OS Health Check

- Processor Analysis
- MSU Usage Analysis
- Storage / Memory Analysis
- SMF 30 Address Space Analysis
 - Address Space - Classification and CPU
 - Avg CPU Dispatching Priorities
 - Started Tasks Over Time
 - Various Address Space Measurements
 - Address Space CPU Summary
 - CPU Intensity
 - User Key Common Analysis
 - Interval - User Key Common Attempts
 - Step End - User Key Common Attempts
 - Job End - User Key Common Attempts
 - WLM Analysis
 - Virtual Storage Activity Analysis
 - Environment Summary

User Key Common Analysis

Interval - User Key Common Attempts

No data found



No data found

Step End - User Key Common Attempts

No data found



No data found

Job End - User Key Common Attempts

No data found



No data found

This is the ideal view of these reports!

Summary



- Use large pages where you can
- z/OS 2.2 and 2.3 made changes to improve performance for large memory
- Basics still apply though:
 - Don't page
 - Keep aux storage relatively unused
 - Make sure your large page limits are sufficient
- User key common goes away in 2.4



Thanks!
Questions?

Email: scott.chapman@epstrategies.com