z/VM Paging Configuration Options

See https://www.vm.ibm.com/library/presentations/ for latest version of this presentation.

Walter Church
z/VM Development Lab: Endicott, NY
wchurch@us.ibm.com

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

Db2*	FlashCopy*	IBM eserver	OMEGAMON*	XIV*	z10 BC	zSecure
DirMaint	FlashSystem	IBM (logo)*	PR/SM	z13*	z10EC	zSeries*
DS8000*	GDPS*	IBM Z*	RACF*	z13s	z/Architecture*	z/VM*
ECKD	ibm.com	LinuxONE*	System z10*	z14	zEnterprise*	z Systems*
FICON*	IBM Cloud*	LinuxONE Emperor	System 390*	z15	zPDT	
		LinuxONE Rockhopper	WebSphere*		z/OS*	

* Registered trademarks of IBM Corporation

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.
ITIL is a Registered Trade Mark of AXELOS Limited.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
UNIX is a registered trademark of The Open Group in the United States and other countries.
VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.
Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Abstract

In configuring z/VM paging, you are faced with many choices, and the number of choices has increased as additional enhancements have been made. This presentation walks through the choices and gives guidance for making and verifying decisions. In order to discuss these choices, an introduction to the paging subsystem is given. This presentation focuses on paging to ECKD 3390 devices, but briefly mentions paging to EDEV SCSI devices. Topics include use of EAV, HyperPAV, number of volumes, etc.

First

What is paging?

Then

How much page space do I need?

Followed by

What types of paging space can I have?

And finally

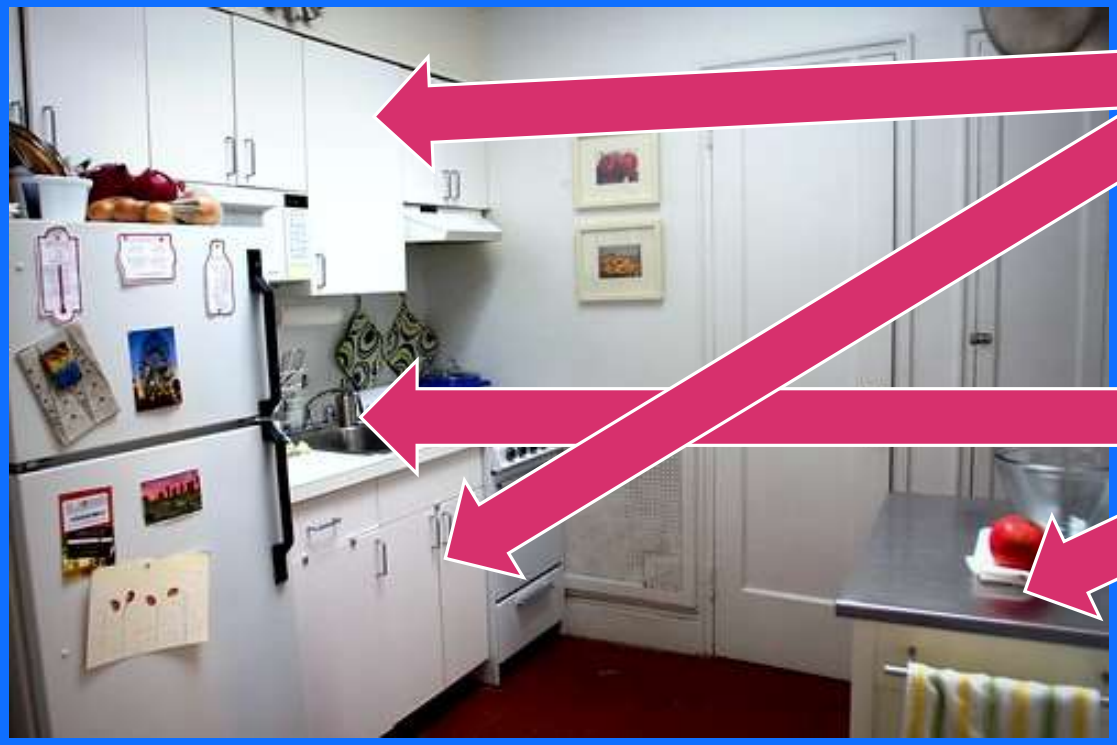
Paging subsystem monitoring and troubleshooting.

Why do we page?

"Because disk is less expensive than memory."
- anonymous IBMer

Allows the consolidation of memory white space across multiple virtual machines.

Why do we page?



Cabinets – more space available, but more work to get to things – similar to page space

Countertop – most commonly used but least space available – similar to central storage

Who pages?

- Multiple levels possible
  - Host (z/VM) paging
  - Guest (Linux, z/OS, etc)
- z/VM designed to handle high levels of paging activity
- Paging space vs. Paging activity
- Page read is what causes delays to guest

- Linux historically configured not to actively page
  - Typically referred to as "swapping"
  - Handshakes with z/VM (Asynchronous Page Fault)
- Double paging
  - z/VM pages in a page so guest can page out the contents
  - Goal is to avoid this unnecessary overhead
- z/OS doesn't have a handshake for paging

Memory Overcommitment

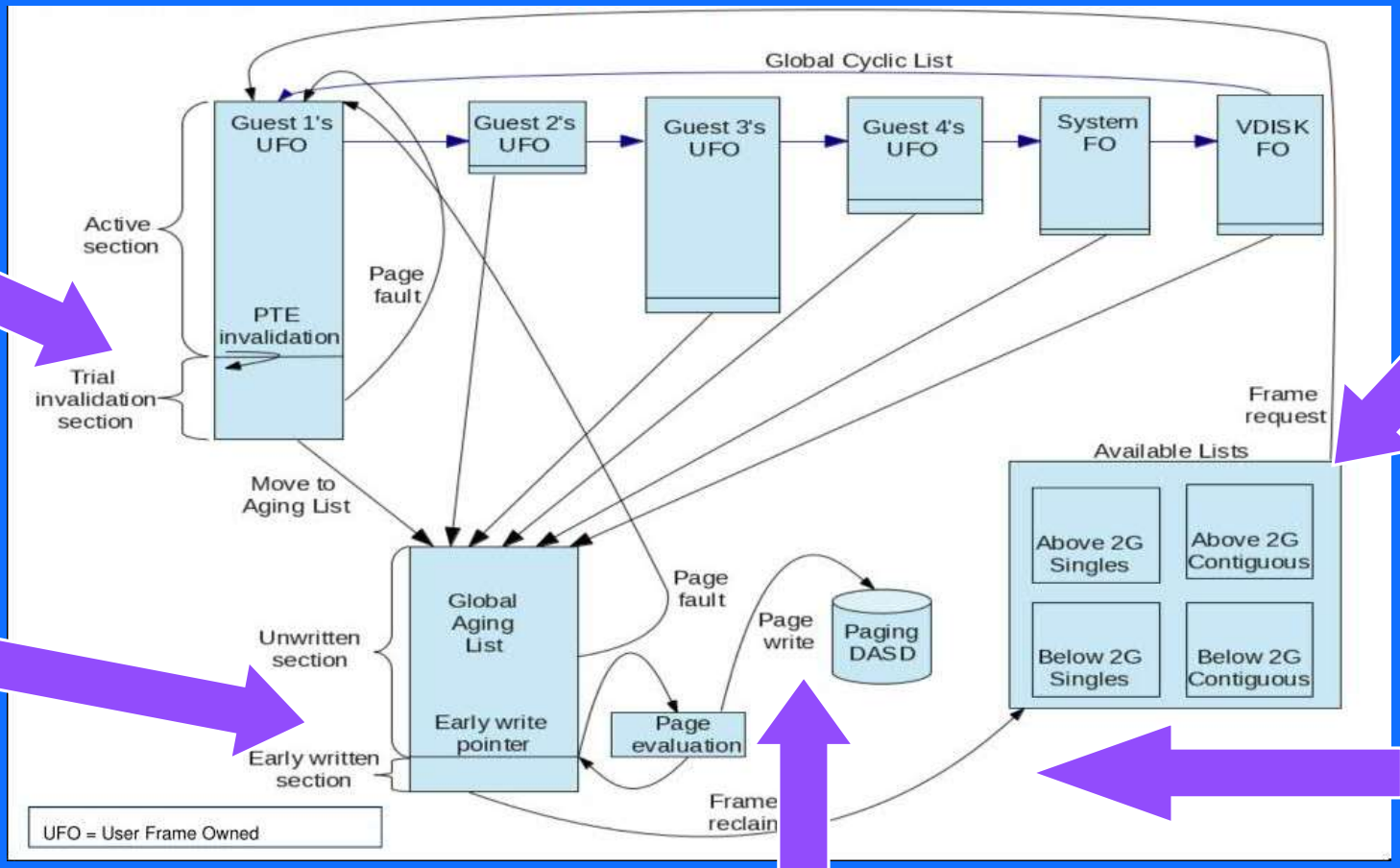
- Do you overcommit?
  - No? Your salesperson loves you!
- By how much?
  - 1.5, 2, 2.5, 3, etc
- Always have the necessary paging space

- Not sure how to figure out your level of overcommitment?
  - It’s easy!
  - Total amount of virtual memory (including guest, host, NSS/DCSS, VDISK, etc)
  - Divided by
  - Total amount of real memory available to your LPAR
  - Too much maths? Use VIR2REAL EXEC!

How pages move through the system

Guests don't touch all their memory all the time, pages slowly age down the list if they're not touched

We don't want to take pages from guests and immediately page them out, much like fine alcohols, pages have a multistep aging process



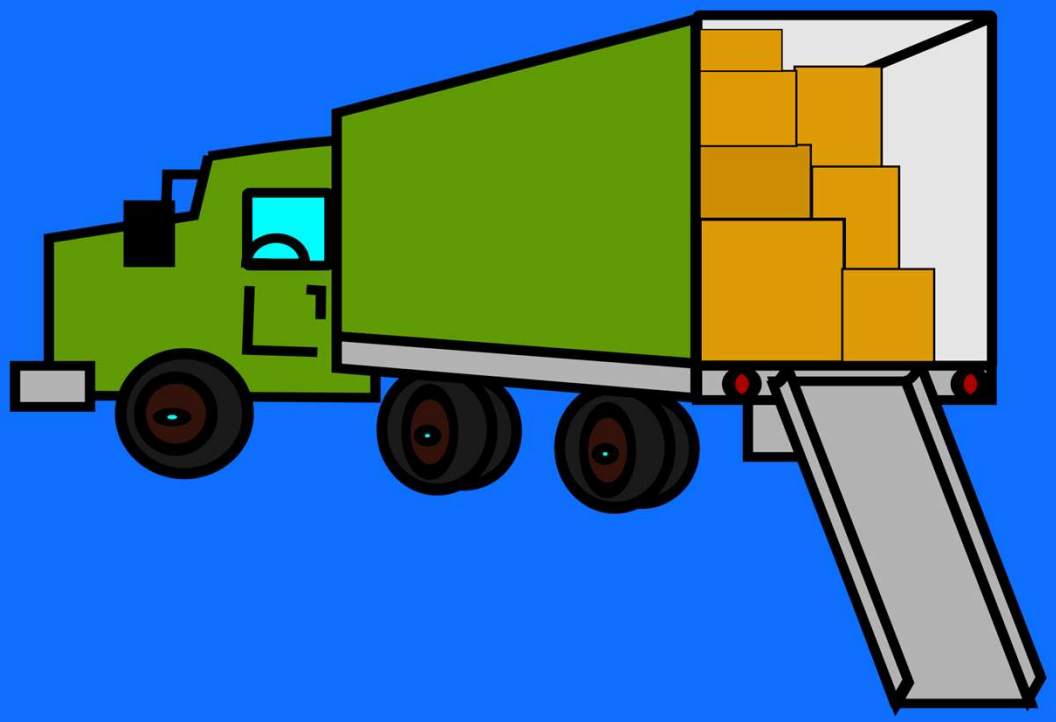
Different types of requests have different memory needs, so we keep a variety of buckets of available frames to back guest pages

Don't necessarily have to write out a page when we need to replenish available lists!

Writes are system overhead, reads (to resolve page faults) take priority

What is Pageable?

- Guest Memory
- PGMBKs (host DAT structures)
- VDISKs
- Some SYSTEM address spaces



What is not Pageable?

- Frame Table – proportional to size of real memory
- CP Nucleus (Kernel) and data areas
- Various free storage control blocks
- SXS Page Table
- Other structures and allocations
  - Trace frames
  - Prefix pages
  - Etc
- Amount of non-pageable memory varies based on configuration and workload

Don't want a piece of memory to get paged out?

- SET RESERVED
  - Always good to reserve your MONDCSS and MONWRITE user
  - Promise group of pages amount
  - Users and segments

- ~~Can’t I use LOCK to lock a range of pages in storage?~~ **NO!**

How much PAGE space do I need?
Or
How I learned to stop worrying and love the PGT004

Does a PGT004 always mean I need to add more paging space?

Yes, you should always follow the guidelines about paging space

Maybe it is also a sign that workload adjustment is needed

- For example – a group of batch jobs that all happened to kick off at the same time

Maybe it's indicative of a problem with a user, but this brings us to our next point.....

NO CHEATING

Because of guest memory touch patterns, you might observe less paging space in use than you calculated. Do not allow this to lead you into paging iniquities!

Paging Space Considerations

1. Maximum total virtual machine size
2. Maximum total size of virtual disks in storage (VDISKs)
3. NSS/DCSS
4. System overhead
5. System growth

Maximum total virtual machine size

- Consider the logon and maximum allowed storage sizes for each virtual machine
- Potential to instantiate all of the pages within its base address space
- Any data spaces it is allowed to create

Maximum total size of virtual disks in storage (VDISKs)

- Shared and private VDISKs
- Number of pages that can be instantiated for each

NSS/DCSS

- Total number of shared pages within all NSS/DCSSs
- Number of exclusive pages multiplied by the number of virtual machines loading them

System overhead

- Number of CP object directory pages as reported by the DIRECTXA utility
  - Additional amount for future directory growth, possibly 2x
- Allow approximately 1% of previous calculations for pageable PGMBKs
- Number of system virtual pages can vary based on your workload
  - MIN(10% real memory size, 4G)

System Growth

- Consider any changes to the CP Directory
  - Defining more users
  - Increasing virtual storage sizes for existing users
  - Increasing the number of allowable VDISKs
- Consider your SSI cluster
  - Size of guests that could relocate
  - size of private VDISKs allocated to those guests

Total

For ECKD devices, to calculate the number of cylinders of that device type needed:

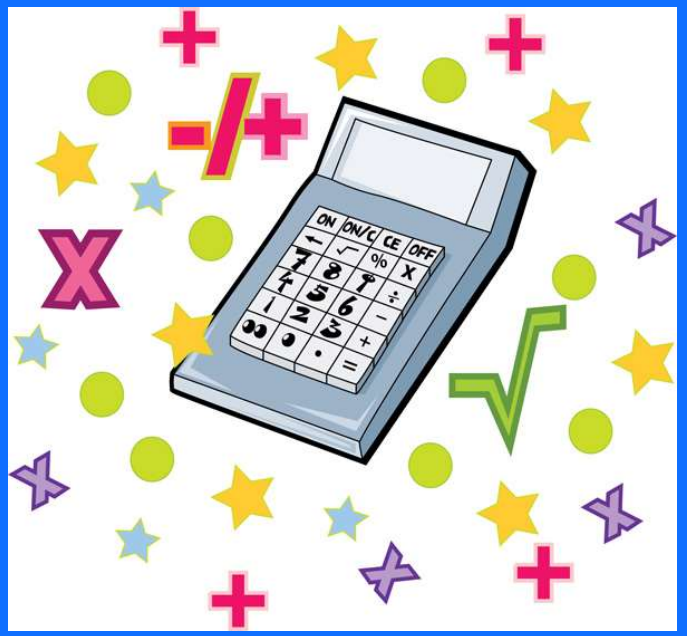
- Convert PAGE space need (bytes) to 4K pages by dividing by 4096 (bytes/page)
- Convert 4K pages to cylinders by dividing by number of pages/cylinder (180 pages/cylinder on 3390 models)

For FBA devices, if you have to determine the number of blocks needed for paging space:

- Convert PAGE space need (bytes) to 4K pages by dividing by 4096 (bytes/page)
- Convert 4K pages to blocks by multiplying by 8 (Eight 512-byte blocks per 4K page)

Tools to help with this

- VIR2REAL - http://www.vm.ibm.com/DOWNLOAD/packages/descript.cgi?VIR2REAL
- Performance monitoring application of your choice
- Pen, paper, calculator
- Planning spreadsheet (you have this, right?)



Now that I know how much PAGE space I need, how do I get it?

IT DEPENDS

This section gives you some ideas, considerations, and our experiences, YMMV

Two paging packs diverged in a yellow wood

ECKD

- Extended Count Key Data
- Extended Address Volumes (EAVs) allow 1,182,006 cylinders (~811GB)
  - PAGE space can go up to EAV limit
  - SPOOL still at old limit 0-65519 cylinders
- Features of DS8000
  - HPF
  - HYPERPAV

FBA-SCSI

- Fixed Block Architecture (FBA)
  - 512 Bytes
  - Defined as SCSI LUN to Linux
    - WWPN
  - Defined as edev (9336) to z/VM
- Small Computer System Interface (SCSI)
- 16,777,215 pages (64 GB minus 1 page) of the volume

And I took the path lined with FICON Channels.

A Loose Comparison for z/VM Paging: ECKD vs SCSI

ECKD	SCSI
Exploits HyperPAV concurrent I/O where available	Exploits concurrent I/O for CP paging
Supported by GDPS	No GDPS
Less host CPU utilization per start	High host CPU utilization per start for CP managed volumes
Fewer midrange storage solutions available	Supports V7000 or other midrange storage through an SVC
Supports solid state DASD options	Supports solid state DASD options & IBM Flash Systems (z/VM 6.4)
If running SSI, you already have ECKD requirement	Cannot have all SCSI environment for SSI
Eliminates WWPN Management, but consumes space for Count/Keys	WWPN Management Issues

Don’t mix types

- Load balancing algorithm optimized for uniform devices for paging
  - Same type, size, speed
- Calculations are done very differently for ECKD and EDEVs
- Mixing could result in unpredictable, and potentially poor, performance

Single-Purpose Volumes

- Do not define PAGE space on the same device as non-PAGE space
  - Use whole volumes for PAGE areas. A volume should be either all PAGE or contain no PAGE at all
  - Likewise, a volume should be either all SPOL or contain no SPOL at all
  - Avoid contention in I/O scheduler with other allocation types

Betcha can’t have just one!

- Assume we have an infinitely fast "pipe" to the paging packs.
- Why not just have 1 really large page pack?
- Only one is less setup and only takes one CPOWNED slot.
- Redundancy is good. However with striping you could still end up with pieces of memory from most of your guests on any given volume, so losing even 1 volume is going to hurt.
- How friendly are you with your storage folks? Do you want to ask for a different volume size for your paging space than for your z/VM system and guest disks?

EAV (Extended Address Volumes) for ECKD

- CP paging space can be allocated anywhere on an Extended Address Volume (EAV)
- Can be up to 1,182,006 cylinders (1 TB)
- Previously limited to a maximum of 65,520 cylinders
- Requires below service for z/VM 7.1
  - In base of z/VM 7.2

Service details	CP	CMS	PerfKit
APAR	VM66263	VM66297	VM66293
PTF	R710 UM35475	R710 UM35483	UM35484

Mr. Owl, how many PAGE volumes to the center of a Tootsie Roll pop?

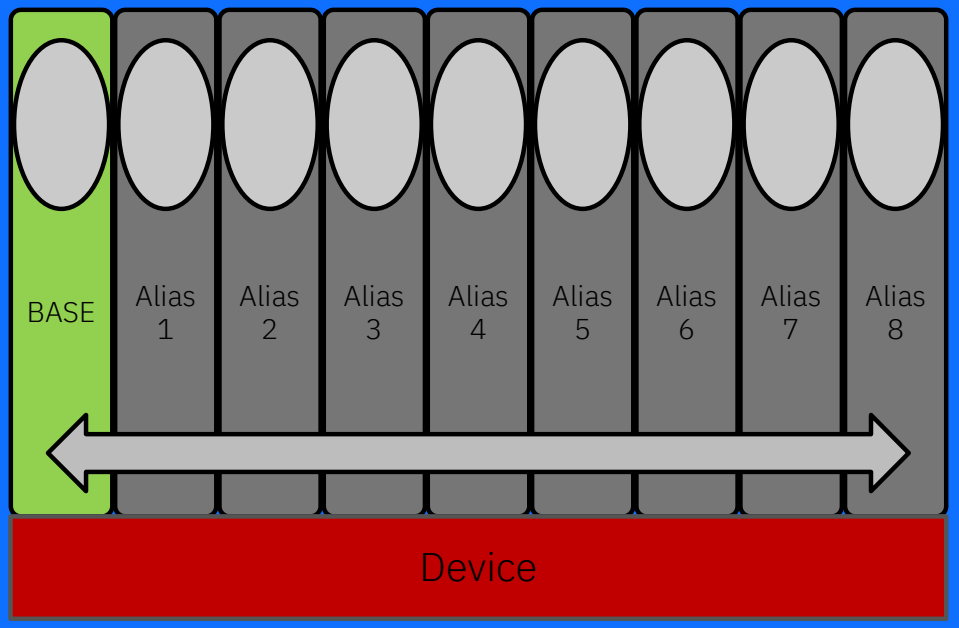
- Allocate PAGE volume(s) large enough to meet total need
- Fewer large PAGE volumes might be easier to manage, but reduces potential paging I/O rates
- Improve paging performance by reducing I/O contention
  - Single-purpose volumes
  - Reduce amount of other I/O activity on the channel path (or dedicate channel path to paging)
  - More real storage (potentially decreases page fault occurrence)

Channels matter!

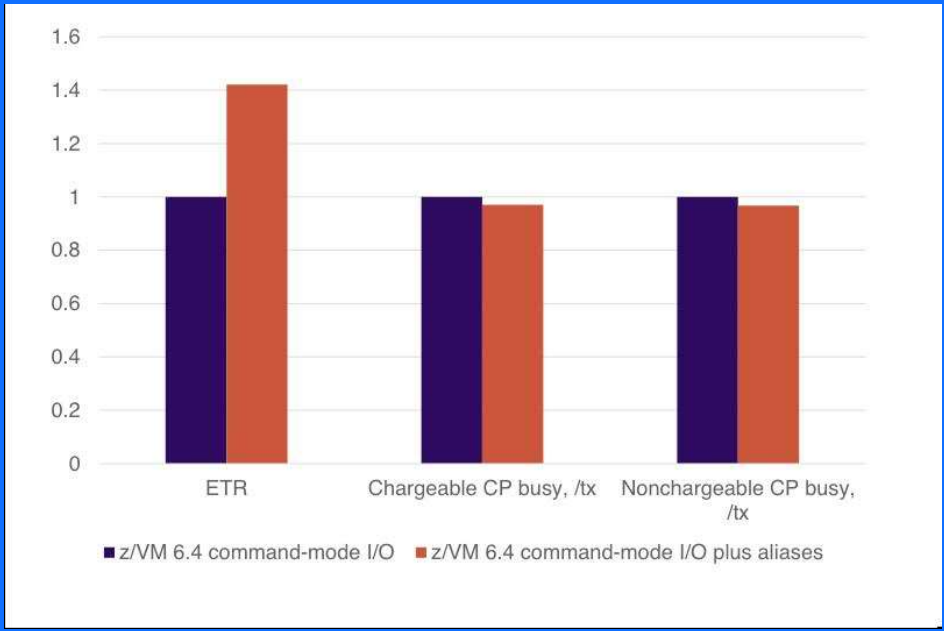
- You can have the best airport in the world, but if it has a dirt road to it, no one will fly!
- Consider not only the type of device, but how you get to it
  - ESCON (shudder) or FICON?
  - What bandwidth?
  - How many channels?
  - Competition?
  - Distance?

HyperPAV

- Allows greater bandwidth via multiple paths to the same volume (1 base + 8 aliases)
- Paid feature
- If the base volume is busy, z/VM selects a free alias device from a pool, binds the alias to the base device, and starts the I/O.
- HyperPAV paging enables the management of fewer and larger CPOWNED volumes



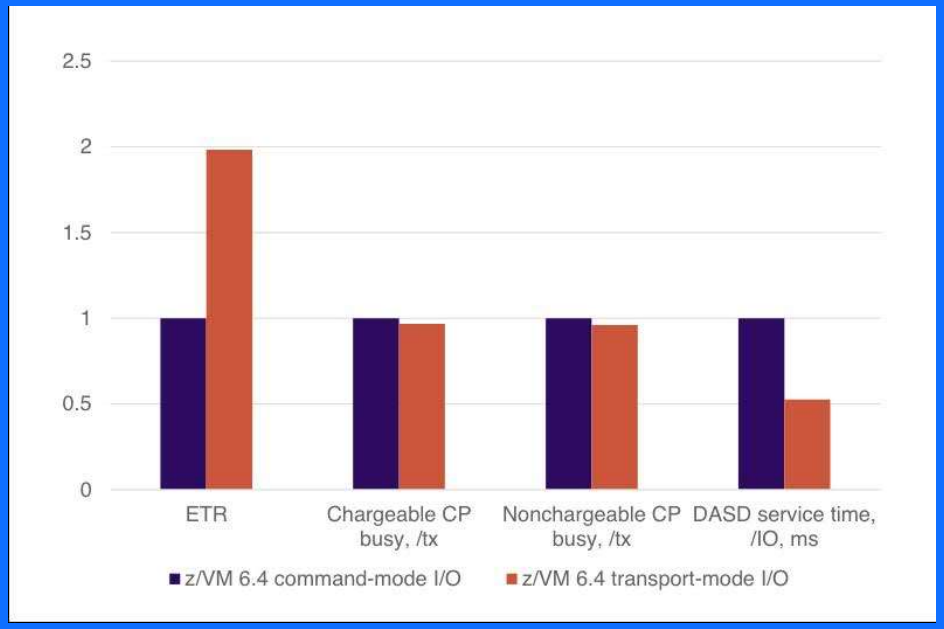
HyperPAV



- Controlled in the configuration file or via CP command.
- HyperPAV is exploited by the z/VM hypervisor for:
  - The SYSRES volume, and volumes containing checkpoint and warm start data
  - Volumes used for paging, spooling, and the z/VM user directory
  - Minidisk pools, as defined by a guest's use of the MAPMDISK IDENTIFY macro

High-Performance Ficon (HPF)

- Designed to help reduce the FICON channel overhead and therefore may improve latency
- Useful for I/O workloads that transfer small (4 KB) blocks of fixed-sized data
- The supported FICON Express adapters support the FICON architecture, FICON channel-to-channel (CTC), and the zHPF architecture simultaneously.



Some ECKD is more equal than others

- DS8K optional feature - Easy Tier
  - Hierarchical organization for storage efficiency
  - Moving frequently accessed info to a flash disk

- Flash? What flash do you mean?
  - IBM Flash Express -> Virtual Flash Memory
    - Processor feature build on Storage Class Memory architecture
  - FlashSystems (storage server) -> IBM FlashSystem A9000 Storage Server
  - Flash memory in any of the IBM Storage Servers, including the DS8900F announced September 2019

What limits do we have?

Limits

ECKD
- Extended Address Volumes (EAVs) allow 1,182,006 cylinders (~811GB paging space)

FBA-SCSI
- 16,777,215 pages (64 GB minus 1 page) of the volume

- Number of aliases = (8 aliases and 1 base)
- Number of CPOWNED volumes = 255
  - SSI systems have to share some slots (paging isn't shared, but others are)
- Number of frames in a single channel program (ECKD transport mode) 136 4K pages

What's your limiting factor?

Page rate?

Page space?

Channel bandwidth?

CPU?

Size of channel program?

Time to format your paging volume?

Doctor, my paging hurts!

“My system is running slowly, is it a paging problem?”

- Are guests in page wait?
  - INDICATE QUEUES EXPANDED
  - INDICATE PAGING
    - Tells whether guests are in page wait
  - INDICATE PAGING ALL
    - Tells which guests have pages out on DASD
- Is your system paging?
  - QUERY ALLOC PAGE
  - INDICATE USER <user> EXPANDED
    - Tells paging reads and writes for a particular user

“My system is running slowly, is it a paging problem?”

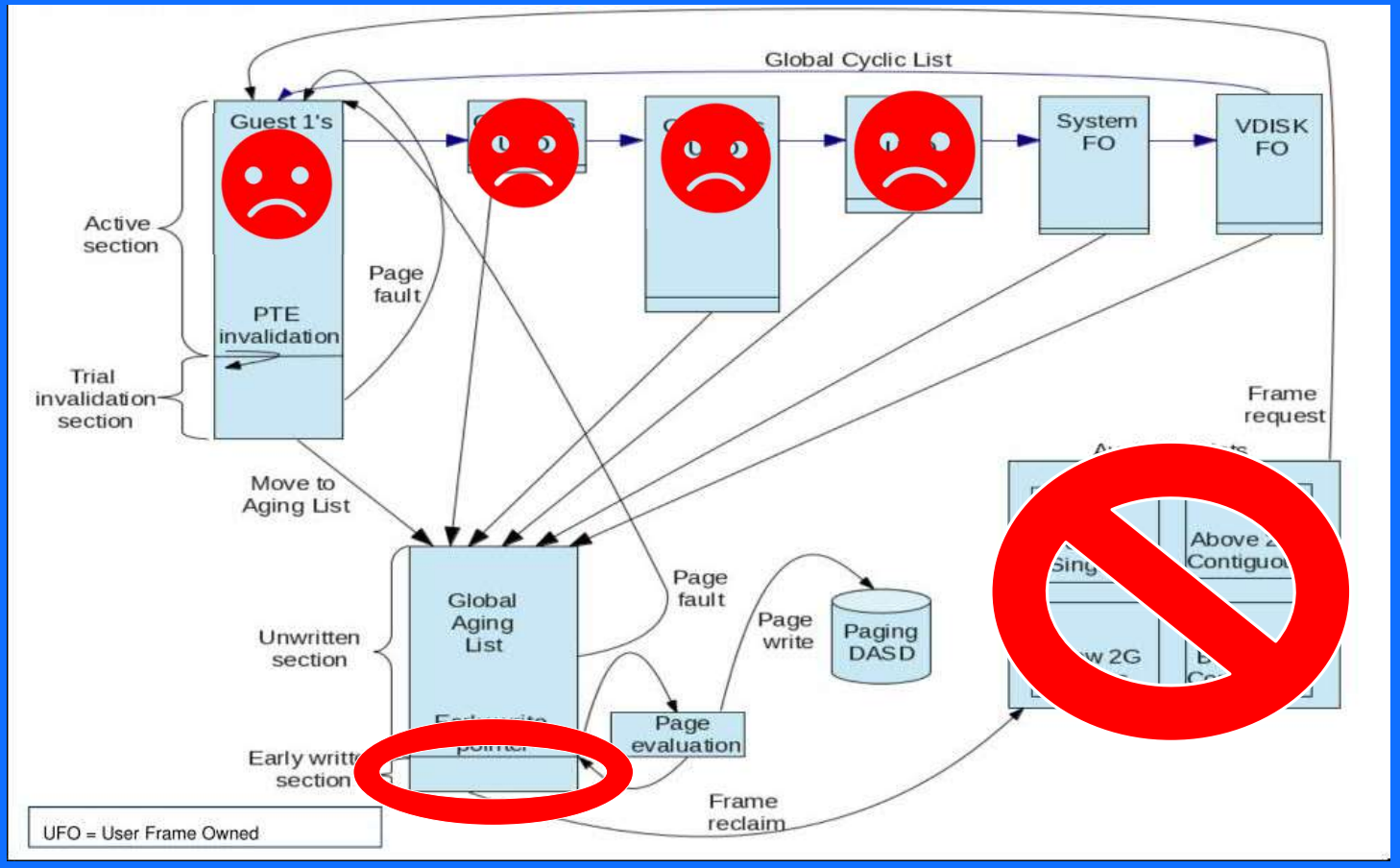
- Just because the answer is yes here does not necessarily mean it is a paging problem…
  - Could also be an I/O problem
  - Could be a CPU problem
- Having historical monitoring data is important, allows you to compare and contrast today vs. yesterday.

“I see a lot of time charged to the z/VM system, is it a paging problem?”

- INDICATE LOAD –
  - a smoothed paging rate over a long period of time, doesn’t respond minute to minute with changes
  - Combined paging rate here is combined reads and writes
- Monitor data or your favorite performance monitoring tool
  - The gold standard

Empty lists are sad lists

In a healthy system, there are pages in all lists.



And then our guests will not be able to get new pages, which makes them sad

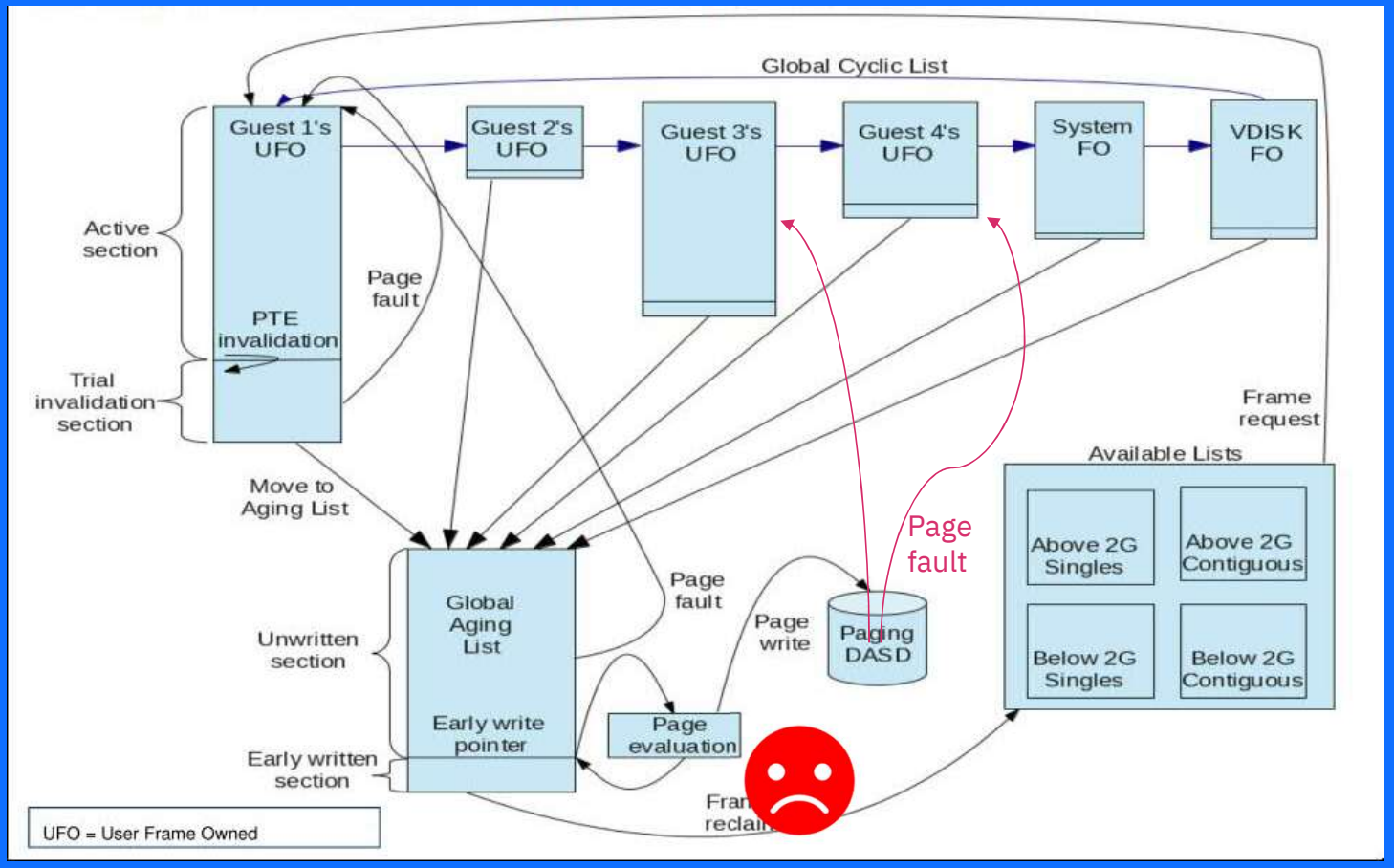
Eventually the available lists will empty

What if the early write list becomes empty? Then we have nowhere to get pages from to replenish our available lists

Write Rate Higher Than Read Rate

In a system where page reads are getting backed up, you might see more page faults satisfied from revalidations of pages on the aging list or trial invalidation section.

The black page faults make up the majority of our page faults, not the red ones



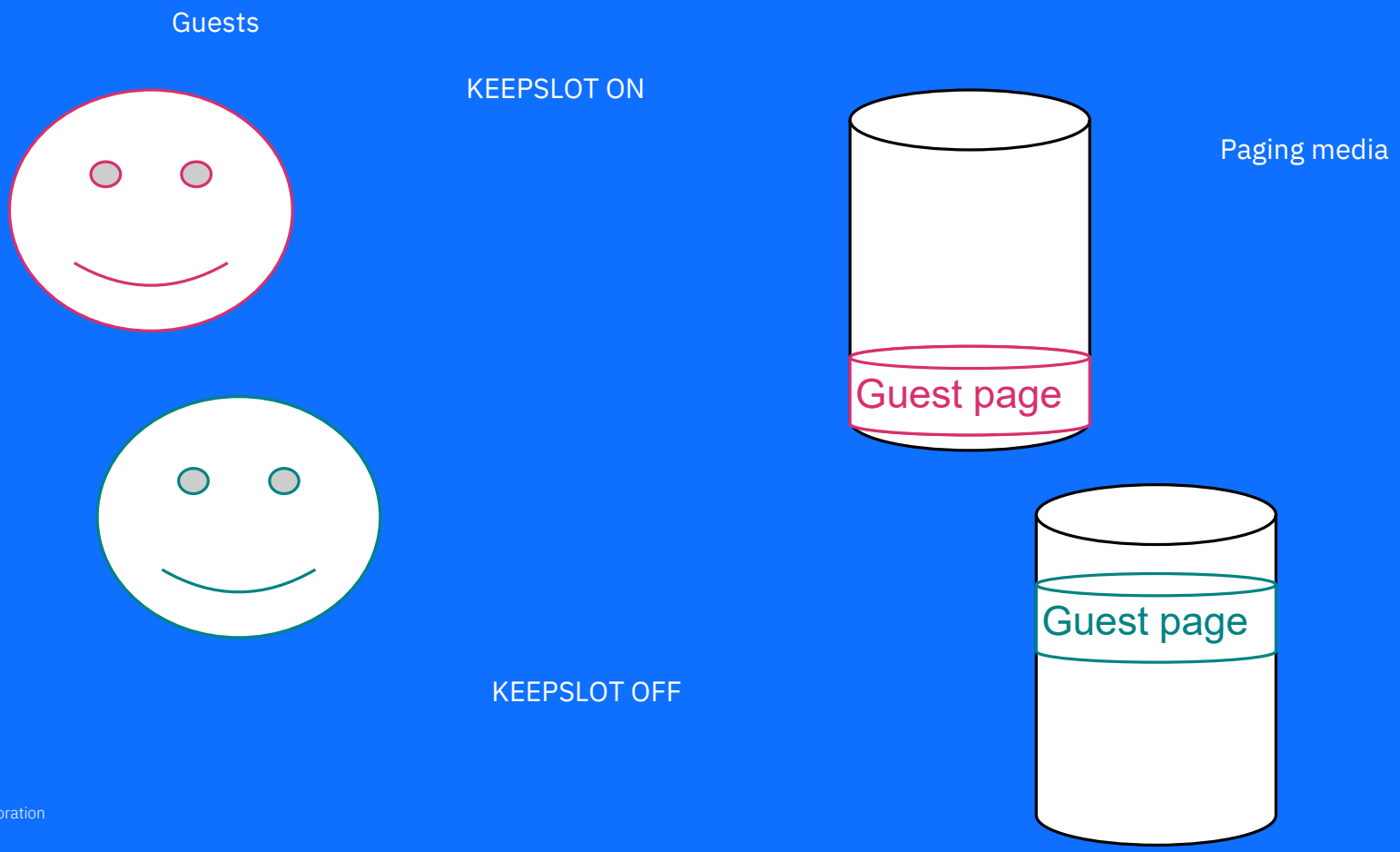
You could turn off early writes in this case to prevent the overhead of writing pages you're probably going to need before we could reclaim the frame.

Why are we spending time writing pages out to DASD if we rarely need to read them in? The system is spending time in an unhappy place!

Example problems – Paging rate

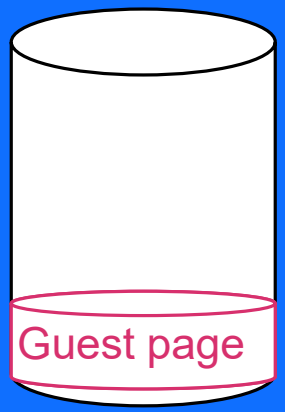
- If we can't read pages fast enough, we won't be able to satisfy page faults in a timely fashion
  - Shows up as many guests in page wait in queries and performance data
  - Diagnostic data
    - INDICATE PAGING
    - Monitor records – D4R4
    - Performance Toolkit screens
      - FCX143 – Pagelog
      - FCX297 – Agellog
- If we can't write out pages fast enough, you'll have a long aging list and available lists will be empty, so getting frames could take awhile
  - Diagnostic data
    - Monitor records – D3R1
    - Performance Toolkit screens
      - FCX294 – Avlb2glg
      - FCX295 – Avla2glg
      - FCX296 - Steallog
      - FCX297 – Agellog

Example problem – KEEPSLOT setting



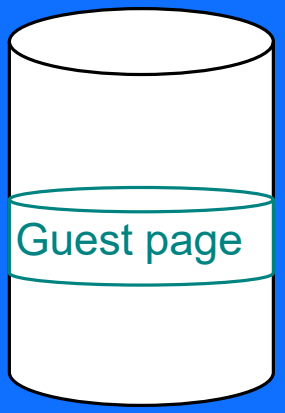
Why would I want KEEPSLOT on?

KEEPSLOT ON



Keeping that slot on DASD means that if the guest doesn't change the page, we don't have to write it out again.
We saved ourself some work!
But it cost us paging space!

KEEPSLOT OFF



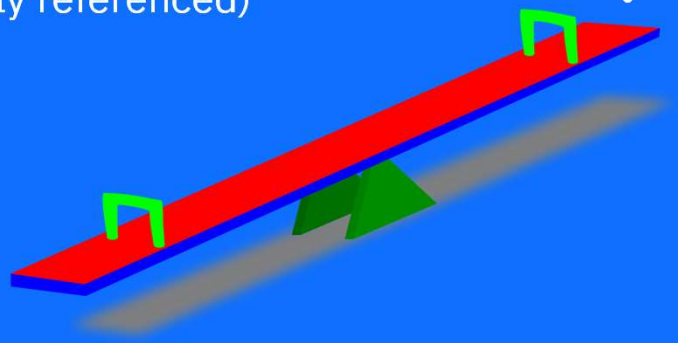
Not keeping the slot means we'll have to write it back out no matter what! But then we saved some page space.

If you're a cheater on page space, maybe this option is for you.
But you're not a cheater, right?

KEEPSLOT is a balancing act

- On one hand, we don’t want to waste a lot of system time writing back out unchanged pages
  - Diagnostic data
    - Monitor records
      - D3R1 (KEEPSLOT setting)
    - Performance Toolkit screens
      - FCX297 – AGELLOG (shows you pages written that were only referenced)

- On the other hand, we don’t want to use up paging space for pages that are likely to get changed after we read them in
  - Diagnostic data to watch your paging space
    - QUERY ALLOC PAGE
    - Monitor records – D0R6
    - Performance Toolkit screens
      - FCX146 - AUXLOG

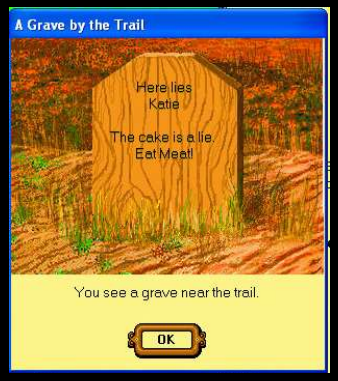


What can go wrong?

A paging error occurred reading a pageable page table

Six continuous paging errors occurred..

PGT004 – all paging space is exhausted



Paging space is filling up, hitting paging warning threshold (default 90%)

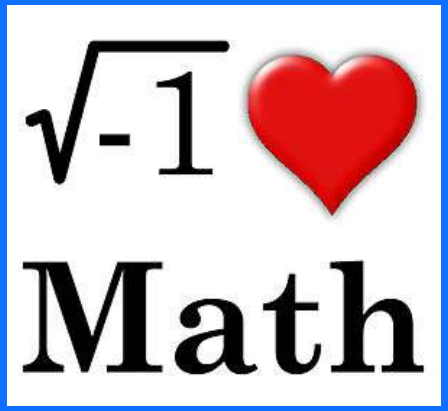
Conclusion

Paging is a strength of z/VM - paging is what makes the magic of overcommitment work

There is a formula to follow for how much page space you need – don't cheat!

There are lots of options for paging space – choose what works best for your shop!

Because paging is a complex system, it needs monitoring - always use a performance monitoring tool!



BACKUP

What does my current PAGE layout look like?

```
FCX109      CPU nnnn  SER nnnnn  Interval HH:MM:SS - HH:MM:SS    Perf. Monitor

Page / SPOOL Allocation Summary
PAGE slots available        2642m          SPOOL slots available      8411760
PAGE slot utilization          0%          SPOOL slot utilization          0%
T-Disk space avail. (MB)    14152          DUMP slots available        23587k
T-Disk space utilization       0%          DUMP slot utilization           0%

___ .          .                      .     .     .     .     .     .     .     .     .     .     .     .
< Device Descr. ->                    <------------ Rate/s ------------>   I/O       Serv MLOAD Block %Used I
           Volume Area   Area     Used  <--Page---> <--Spool-->         SSCH Inter Queue  Time  Resp  Page   for O
Addr Devtyp Serial Type   Extent     %  P-Rds P-Wrt S-Rds S-Wrt Total  +RSCH feres Lngth /Page  Time  Size Alloc M
BE00 3390-9 ATP000 PAGE   11793420    0   34.1  13.6   ...   ...  47.7    9.1     1     0    .0    .0     1   100 C
BE01 3390-9 ATP001 PAGE   11793420    0   33.8  13.7   ...   ...  47.5    9.0     1     0    .0    .0     1   100 T
BE02 3390-9 ATP002 PAGE   11793420    0   34.7  13.8   ...   ...  48.5    9.6     1     0    .0    .0     1   100 T
BF0A 3390-9 ATP024 PAGE   11793420    0   35.5  13.8   ...   ...  49.3   10.6     1     0    .0    .0     1   100 T
C008 3390-9 ATP036 PAGE   11793420    0   34.8  13.6   ...   ...  48.5   10.1     1     0    .0    .0     1   100 C
C009 3390-9 ATP037 PAGE   11793420    0   34.1  13.8   ...   ...  47.8    9.1     1     0    .0    .0     1   100 C
C00A 3390-9 ATP038 PAGE   11793420    0   33.8  13.8   ...   ...  47.6    9.5     1     0    .0    .0     1   100 C
C00B 3390-9 ATP039 PAGE   11793420    0   32.9  13.6   ...   ...  46.5    8.9     1     0    .0    .0     1   100 C
C00C 9336   ATP040 PAGE   11793420    0   34.1  13.8   ...   ...  47.9     .0     1     0    .0    .0     1   100 E
Select a device for I/O device details
Command ===> _
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

Size of volumes
How much used
I/O rates (R/W)

Queue can indicate volume unable to keep up with requests

Tips for Paging

Aliases to devices

A disk volume should be either all paging (cylinders 1 to END) or no paging at all.

When you decide where to place paging volumes, take the DASD subsystems' capabilities and existing loads into account

avoid ESCON chpids, go or FICON for multiple IO

multiple chpids to each DASD controller that holds paging volumes

If you have FCP chpids and SCSI DASD controllers, you might consider exploiting them for paging.
A SCSI LUN defined to the z/VM system as an EDEV and ATTACHed to SYSTEM for paging has the very nice property that the z/VM Control Program can overlap I/Os to it.
This lets you achieve paging I/O concurrency without needing multiple volumes.
However, don't run this configuration if you are CPU-constrained.
It takes more CPU cycles per I/O to do EDEV I/O than it does to do classic ECKD I/O

Encryption

- Encrypts pages so that they are not readable if your paging media is accessed outside the system (for legitimate or illegitimate purposes)
- Requires z14 or greater with CPACF
- Despite the extra cost of encryption, the z14 with encrypted paging enabled performed better when compared back to a z13 (measured one test case).
- CPU cost of encrypted paging is a function of the paging rate rather than the LPAR size
- When first enabling, allows you to choose algorithm (AES 128, 192, 256)
- Remember to keep a back-up system configuration file available and specify that on your SALIPL screen in case of emergencies

Adding and Removing Paging Space

Is this dynamic?

Adding – yes

- Provided you have a RESERVED CPOwned slot

Removing – unlikely

- Have to drain first
- Drain will not actively remove pages from the volume, but will keep us from using it for any new paging
- Eventually the workload might page in all those pages and the volume would be empty, but that's not guaranteed
- Could still have system pages paged out, preventing remov

http://www.vm.ibm.com/newfunction/#active-drain

Adding a page volume

Add the volume to your I/O configuration (via updating the IOCDS or via DPM)

Attach the volume to a z/VM guest and use CPFMTXA

DEFINE CPOWNED slot (remember to update your CPOWNED list in the SYSTEM CONFIG file)

DETACH from your guest and ATTACH to system

Removing a page volume

DRAIN the volume of all pages

Remove from the SYSTEM CONFIG

Probably re-IPL your system to finish the drain

Remove from I/O configuration

How do I know if it worked?

QUERY ALLOC PAGE

QUERY CPOWNED

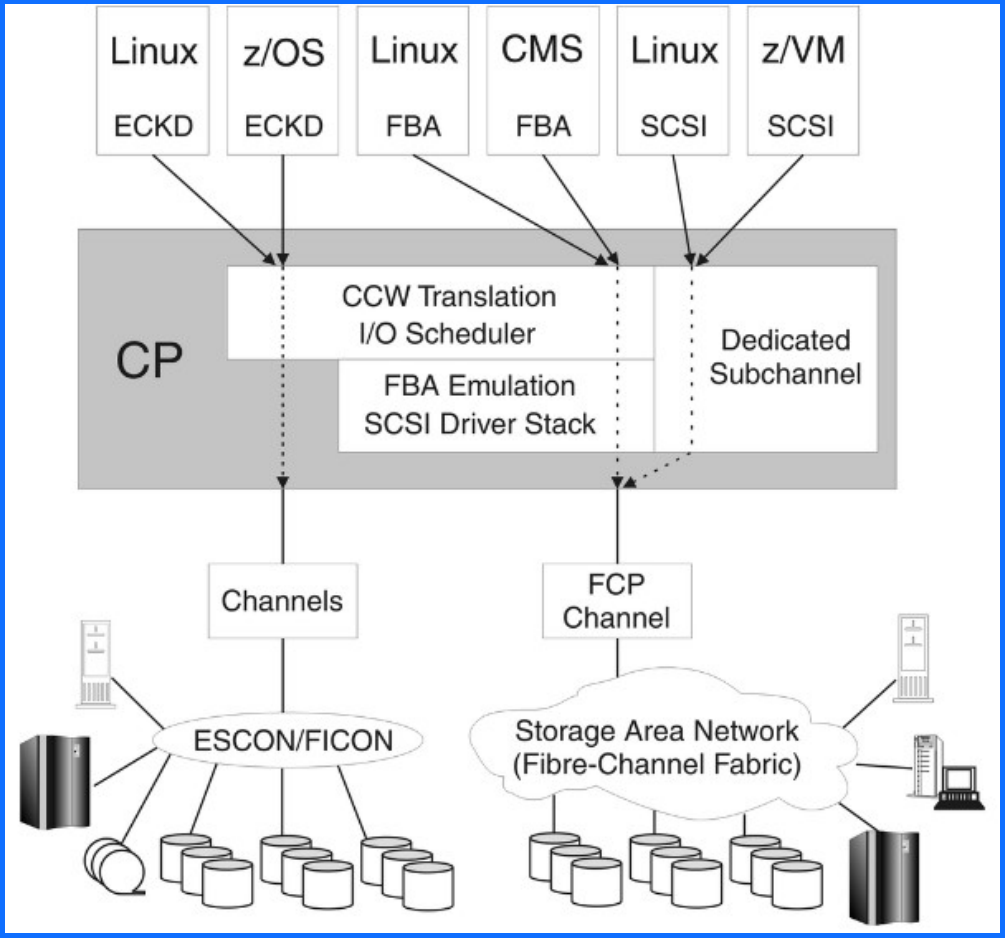
Perfkit screen?

Look and Emulation

ECKD
FBA SCSI

AND NOW ALL ECKD TALKING POINTS
Because that's what we use most

Hold this slide for Version 2



https://www.ibm.com/support/knowledgecenter/en/SSB27U_7.1.0/com.ibm.zvm.v710.hcpa5/scsiover.htm