What's New in the z/VM 6.3 Hypervisor

Linux on System z Live Virtual Class
August 27, 2014

John Franciscovich
francisj@us.ibm.com

Acknowledgements

- Bill Bitner
- Brian Wade
- Alan Altmark
- Emily Hugenbruch
- Mark Lorenc
- Kevin Adams
- Romney White

- … and anyone else who contributed to this presentation that I may have omitted

Topics

- Recent Enhancements
- Scalability
  - Large Memory Support
  - Enhanced Dump Support
- HiperDispatch
- Virtual Networking
- Appendices
  - Technology Exploitation
  - Miscellaneous Enhancements

z/VM 6.3 Themes

- Reduce the number of z/VM systems you need to manage
  - Expand z/VM systems constrained by memory up to four times
    - Increase the number of Linux virtual servers in a single z/VM system
  - Exploit HiperDispatch to improve processor efficiency
    - Allow more work to be done per IFL
    - Support more virtual servers per IFL
  - Expand real memory available in a Single System Image Cluster up to 4 TB
- Improved memory management flexibility and efficiency
  - Benefits for z/VM systems of all memory sizes
  - More effective prioritization of virtual server use of real memory
  - Improved management of memory on systems with diverse virtual server processor and memory use patterns

z/VM 6.3 – Recent Enhancements

- **Environment Information Interface**
  - Available with APAR VM65419 / PTF UM34348

- **CPU Pooling**
  - Available with APAR VM65418 / PTF UM34348

- **PCIe / 10GbE RoCE Express Feature / zEDC Express Feature**
  - Available with:
    - IBM zEC12 or zBC12, driver 15, bundle 21

    - VM CP - APAR VM65417 / PTF UM34343
    - VM CMS - APAR VM65437 / PTF UM34401
    - VM TCP/IP - APAR PI20509 / PTF UI19055
    - VM DVF - APAR VM65572 / PTF UM34342
    - z/OS 2.1 - APAR OA43256 / PTF UA72717
    - z/OS 2.1 - APAR OA44482 / PTF UA73687

  - Fullfills 2013 Statement of Direction

Environment Information Interface

- New programming interface allows guests to capture execution environment
  - Configuration and Capacity information
  - Various Levels:
    - Machine, logical partition, hypervisor, virtual machine, CPU pools

- New problem state instruction STore HYpervisor Information (STHYI)
  - Supported by z/VM 6.3
  - Tolerated by z/VM 6.2 ("function not supported")

- Used by IBM License Metric Tool (ILMT)
  - New ILMT 9.0.1 includes the ability to track CPU pools

CPU Pooling

CPU Pooling

- Define and limit the aggregate amount of CPU resources that a group of z/VM guests is allowed to consume
  - Allows capping of CPU utilization for a set of guests to better balance resource utilization

- Define one or more named pools in which a limit of CPU resources is set
  - No restrictions on number of pools or aggregate capacity (can overcommit)

- CPU pools coexist with individual share limits
  - More restrictive limit applies

- CPU pools in SSI clusters
  - Pool capacities are independent and enforced separately on each member
  - Live Guest Relocation
    - Destination member must have an identically named pool with same **TYPE** attribute
    - If limit is not required on destination, remove guest from pool before relocating
  - Recommend defining pools with identical names and types on all members of cluster

CPU Pooling: Use Cases

- Department resource requirements
  - Assign each department's guests to CPU pool with contracted capacity

- Grow workloads without affecting existing requirements and limits
  - Add New Workload
  - Add Capacity
  - Combine LPARs
  - Handle fractional workload requirements

- Prevent resource over-consumption
  - Limit aggressive workloads

CPU Pooling: Defining and Managing

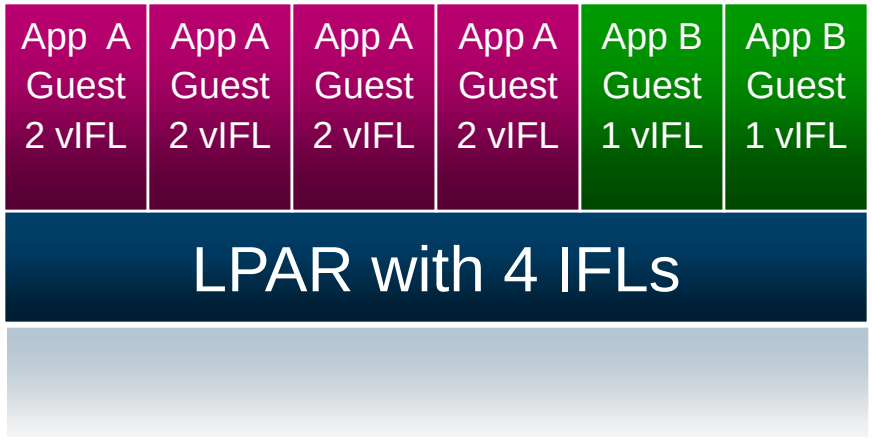
- Use the **DEFINE CPUPOOL** command to define named pools
  - **LIMITHARD** - % of system CPU resources
  - **CAPACITY** – number of CPUs
  - Define for a particular **TYPE** of CPU (**CP** or **IFL**)
- Limits can be changed with the **SET CPUPOOL** command
- Assign and remove guests to/from a CPU pool with the **SCHEDULE** command
- Use **QUERY CPUPOOL** to see information about the pools that are defined on your system

```
query cpupool all
CPU pool  Limit      Type       Members
LINUXP2   8.0 CPUs    IFL             0
CPPOOL10   12 %        CP             8
LINUXP3    30 %       IFL            20
LINUXP1   2.5 CPUs    IFL             6

query cpupool linuxp1 members
CPU pool Limit      Type       Members
LINUXP1 2.5 CPUs     IFL             6
The following users are members of CPU pool LINUXP1:
D70LIN12 D79LIN03 D79ADM D79LIN10 D79LIN07
D79LIN04
```

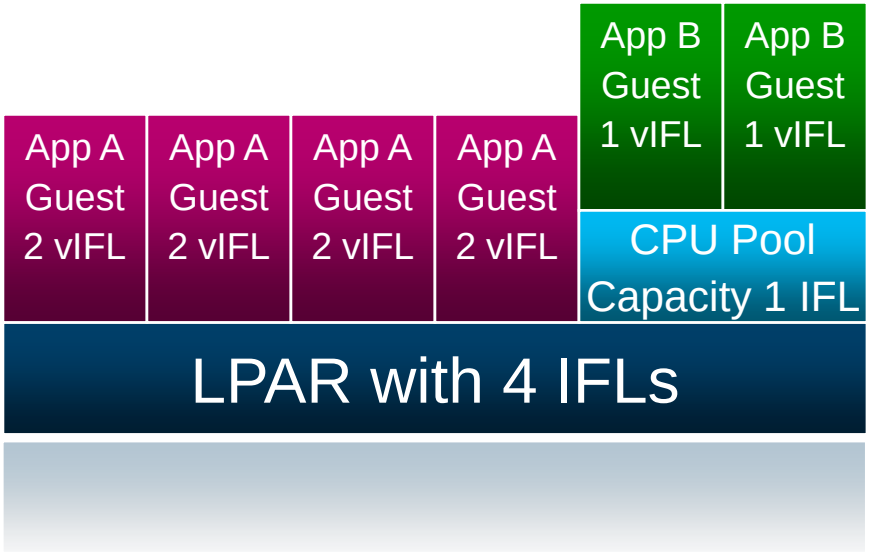
Add New Workload: Without CPU Pooling

- **4 production guests for App A**
  - **May consume up to 4 engines**
- **Add 2 production guests for App B**
  - **May consume up to 2 engines**



Add New Workload: With CPU Pooling

- 4 production guests for App A
  - May consume up to 4 engines
- Create a 1-IFL pool
- Put the 2 App B production guests in the pool
  - App B is limited to 1 engine instead of 2



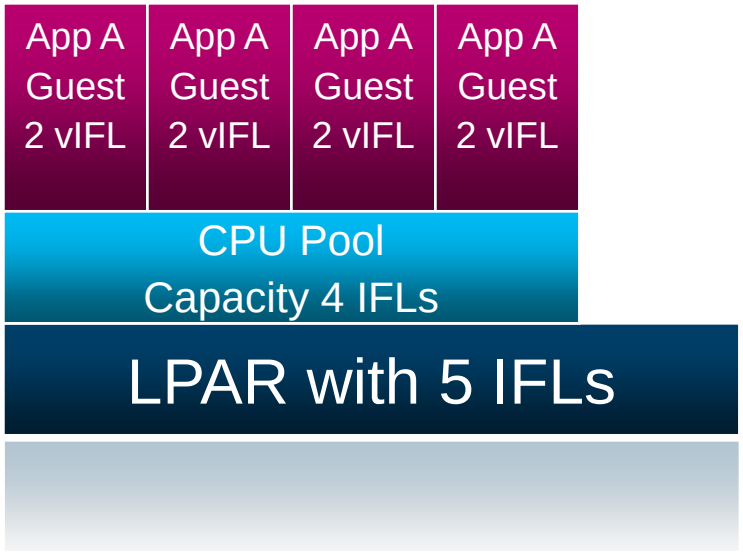
- Allows new workloads to be added cost effectively
- Encourages additional workload consolidation after initial success

Add Capacity: Without CPU Pooling

- **4 production guests for App A**
  - **May consume up to 4 engines**
- **Add another IFL to the LPAR**
  - **Limit for App A increases to 5 engines**

Add Capacity: With CPU Pooling

- LPAR with 4 IFLs
- Set up CPU Pooling for 4 IFLs
  - Limits guests for App A to 4 engines
- Add another IFL to the LPAR
  - App A remains limited to 4 engines
    - Allows capacity to be added for new workload without increasing consumption of existing workloads

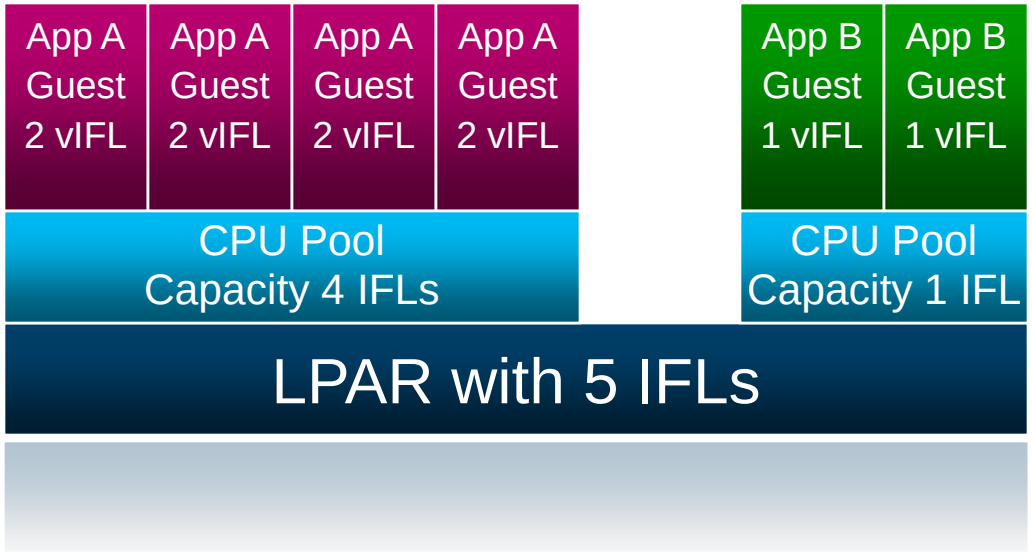


Combine LPARs: Without CPU Pooling

- **LPAR with 4 IFLs and 4 production guests for App A**
  - **May consume up to 4 engines**
- **LPAR with 1 IFL and 2 production guests for App B**
  - **May consume up to 1 engine**
- **LPARs merge to one LPAR with 5 IFLs**
  - **Limit for App A increases to 5 engines**
  - **Limit for App B increases to 2 engines**

Combine LPARs: With CPU Pooling

- LPAR with 5 IFLs
- Create 2 Pools – one with 4 IFLs and one with 1 IFL
- Place the four App A guests in the 4 IFL pool and the two App B guests in the 1 IFL pool
  - App A remains limited to 4 engines
  - App B remains limited to 1 engine



- Avoids increase in software license requirements (and costs)
- Reduces z/VM system management and maintenance workload
- Consolidates resources (memory, paging, network) for greater efficiency

PCIe

PCIe Support: Overview

- Basis for support for guest exploitation of
  - 10GbE RoCE Express Feature
  - zEDC Express Feature
- Allows guests with PCIe drivers to access PCI "functions" (devices)
- PCI functions can be dedicated to a guest
  - Guest must have PCI driver supporting specific function

Defining and Managing PCI Functions

- PCI functions are defined in the IOCP
  - May also be defined, modified, and deleted dynamically with new commands
    - **DEFINE PCIFUNCTION**
    - **MODIFY PCIFUNCTION**
    - **DELETE PCIFUNCTION**

    ⟶ Update IOCP so you don't lose your dynamic definitions

- New or enhanced commands to manage PCI functions
  - **VARY PCIFUNCTION**
  - **ATTACH** (PCIFUNCTION operand)
  - **DETACH PCIFUNCTION**
  - **QUERY PCIFUNCTION**

    - *Sample query response:*

```
PCIF 00000003 ATTACHED TO USER01 00000001 DISABLED 10GbE RoCE
PCIF 00000004 FREE                        DISABLED 10GbE RoCE
PCIF 00000021 NOT CONFIGURED              STANDBY 10GbE RoCE
PCIF 00000026 NOT CONFIGURED              STANDBY 10GbE RoCE
PCIF 00000029 FREE                        DISABLED 10GbE RoCE
PCIF 00000032 ATTACHED TO USER02 00000032 ENABLED 10GbE RoCE
PCIF 00000033 FREE                        ERROR 10GbE RoCE
```

Enabling PCIe Support

- Make sure you have required hardware
  - IBM zEC12 or zBC12, driver 15, bundle 21

- System configuration flie
  - Enable new **PCI** feature on **FEATURES** statement
  - Define size of **IOAT** subpool (in megabytes) on **STORAGE** statement
    - Specify warning threshold percentage for usage

```
STORAGE IOAT 2 Megabytes WARN 80 Percent
```

  - Use **LOCKING** operand to define limits of available storage to be used by PCIe functions
    - Specify percentages to issue warning message and to fail lock request

```
STORAGE LOCKING WARN 50 Percent FAIL 80 Percent
```

  - **QUERY FRAMES** shows **IOAT** and **LOCKING** settings and usage

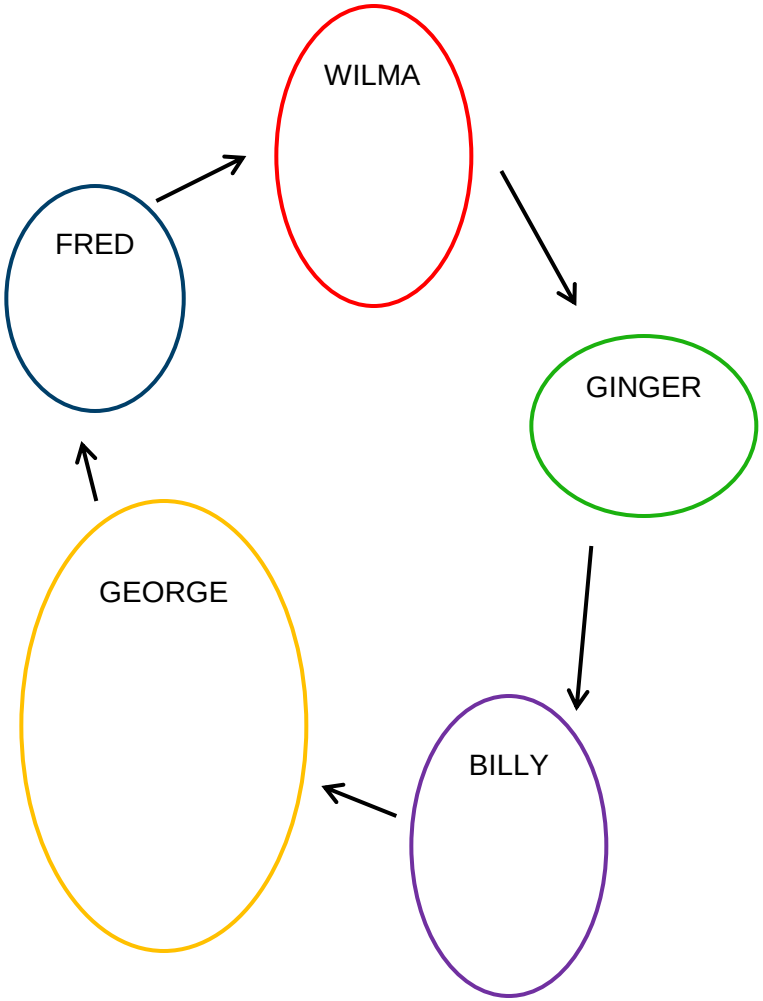
- Review "Using PCIe Functions for z/VM Guests"
  - Chapter 16 (new) in CP Planning and Administration

Scalability – Large Memory Support

Large Memory Support

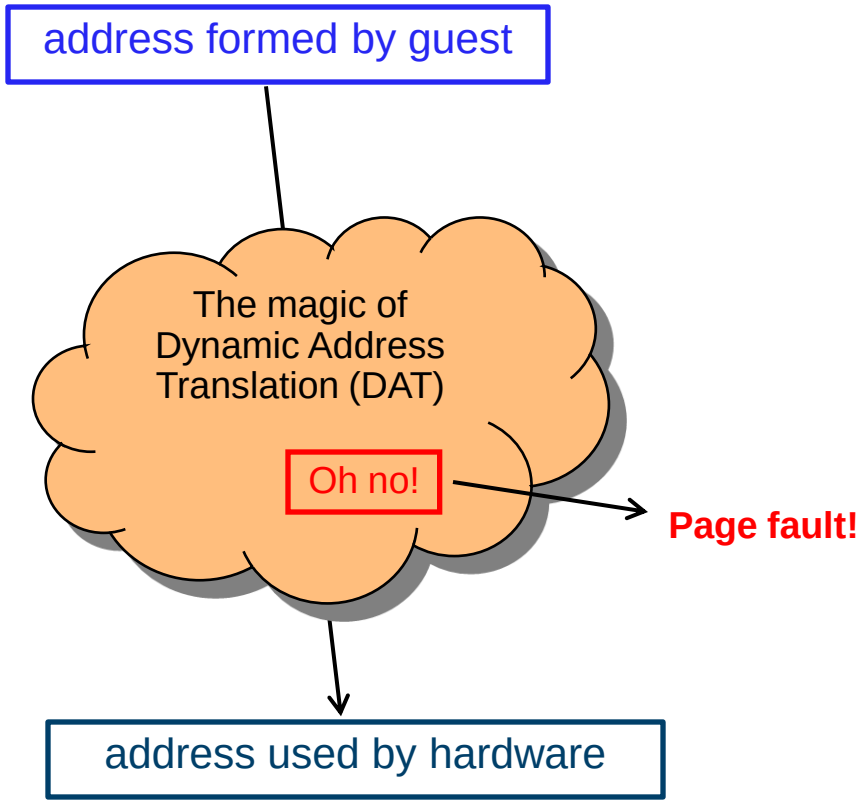
- Support for up to **1TB** of real memory (increased from 256GB)
  - Proportionately increases total virtual memory
  - Individual virtual machine limit of **1TB** is unchanged
- Improved efficiency of memory over-commitment
  - Better performance for large virtual machines
  - More virtual machines can be run on a single z/VM image (depending on workload)
- Paging DASD utilization and requirements have changed
  - No longer need to double the paging space on DASD
  - Paging algorithm changes increase the need for a properly configured paging subsystem
- Recommend converting all Expanded Storage to Central Storage
  - Expanded Storage will be used if configured

New Approach: New Demand Scan Design



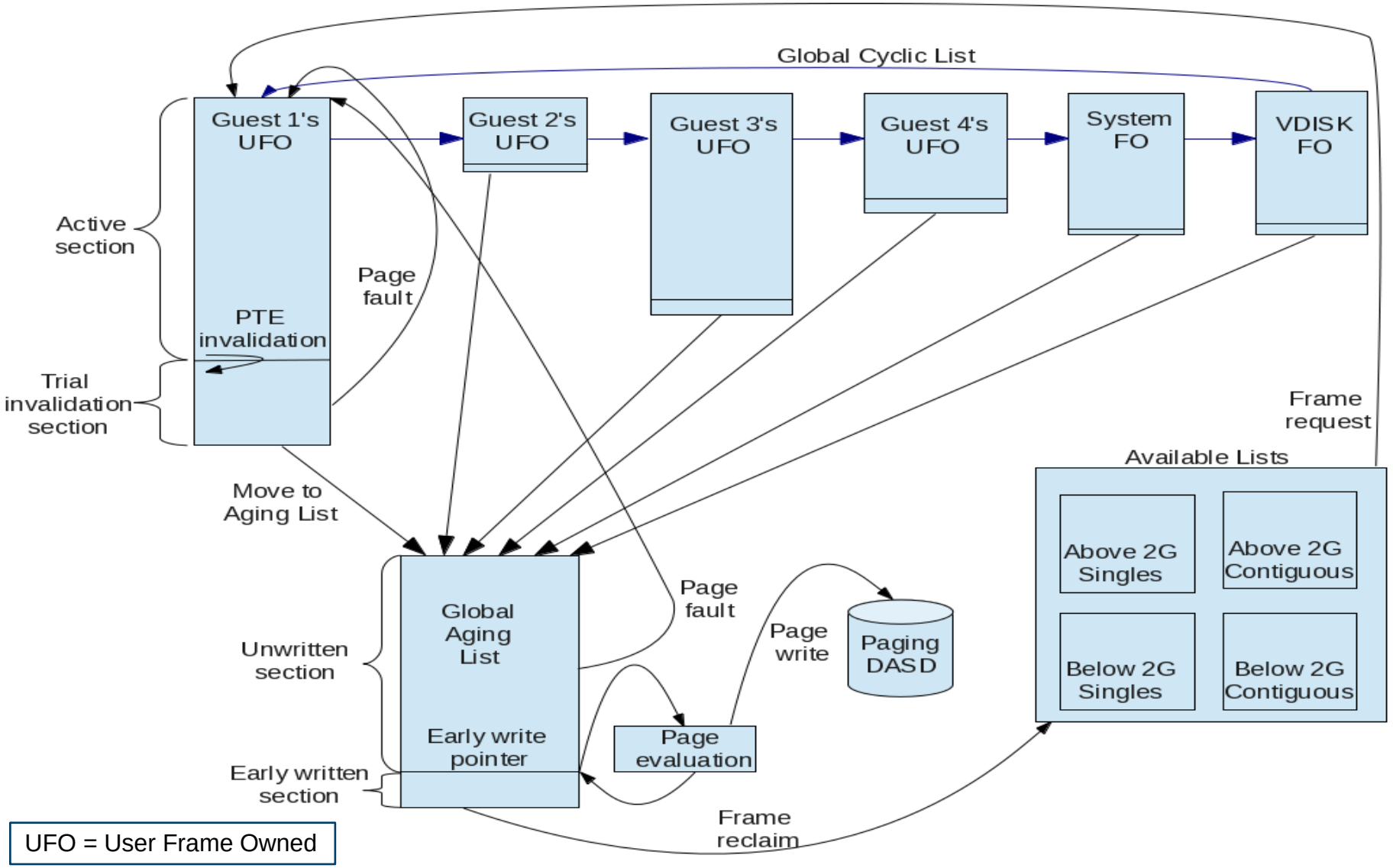
- **Used to:**
  - Visit according to scheduler lists
  - Take heavily at each visited guest
  - Start over at list tops every pass
  - Take from private VDISKs nearly last
  - A “take” was truly a ***reclaim*** of a frame
- **Now:**
  - Cyclically visits the logged-on guests
  - Keeps a visit cursor so it can resume
  - Takes a little and then moves to next
  - Takes from private VDISKs much earlier
  - A “take” is now just a push of in-use frames down toward ***eventual reclaim***
- **Effects**
  - Better equalizing in the face of storage constraint
  - Better equalizing on the notion of “hot” vs. “cold” pages

New Approach: Trial Invalidation



- Page table entry (PTE) contains an “invalid” bit
- What if we:
  - Keep the PTE intact but set the “invalid” bit
  - Leave the frame contents intact
  - Wait for the guest to touch the page
- A touch will cause a page fault, but…
- On a fault, there is nothing really to do except:
  - Clear the “invalid” bit
  - Move the frame to the front of the frame list to show that it was recently referenced
- We call this **trial invalidation**.

Memory Management Algorithm Visualization



UFO = User Frame Owned

Large Memory Support: Reserved Storage

- Reserved processing is improved
  - More effective at keeping specified amount of reserved storage in memory

- Pages can be now be reserved for NSS and DCSS as well as virtual machines
  - Set ***after*** **CP SAVESYS** or **SAVESEG** of NSS or DCSS
    - Segment does not need to be loaded in order to reserve it
    - Recommend reserving monitor segment (**MONDCSS**)

- Reserved settings do not survive IPL
  - Recommend automating during system startup

Large Memory Support: Reorder

- Reorder processing has been removed
  - Could cause "stalling" of large virtual machines
  - No longer required with new paging algorithms

- Reorder commands remain for compatibility but have no impact
  - **CP SET REORDER** command gives RC=6005, “not supported”.
  - **CP QUERY REORDER** command says it’s OFF.

- Monitor data is no longer recorded for Reorder

Large Memory Support: New and Changed Commands

- New commands to **SET** and **QUERY AGELIST** attributes
  - Size
  - Early Writes

- Enhanced **SET RESERVED** command
  - Reserve pages for NSS and DCSS
  - Define ***number of frames*** or ***storage size*** to be reserved
  - Define maximum amount of storage that can be reserved for system

  - **QUERY RESERVED** command enhanced to show information about above

- **STORAGE** config statement enhanced to set AGELIST and maximum reserved storage

- **INDICATE** commands
  - New "instantiated" pages count where appropriate

Large Memory Support: Planning DASD Paging Space

- Calculate the sum of:
  - Logged-on virtual machines' primary address spaces, plus…
  - Any data spaces they create, plus…
  - Any VDISKs they use, plus…
  - Total number of shared NSS or DCSS pages, … and then …
  - Multiply this sum by 1.01 to allow for PGMBKs and friends

- Add to that sum:
  - Total number of CP directory pages (reported by DIRECTXA), plus…
  - Min (10% of central, 4 GB) to allow for system-owned virtual pages

- Then multiply by some safety factor (1.25?) to allow for growth or uncertainty

- Remember that your system will take a PGT004 if you run out of paging space
  - Consider using something that alerts on page space, such as Operations Manager for z/VM

Enhanced Dump Support

Enhanced Dump: Scalability

- Create dumps of real memory configurations up to 1 TB
  - Hard abend dump
  - SNAPDUMP
  - Stand-alone dump

- Performance improvement for hard abend dumps
  - Writes multiple pages of CP Frame Table per I/O
    - CP Frame Table accounts for significant portion of the dump
    - Previously wrote one page per I/O
  - Also improves time required for SNAPDUMPs and Stand-alone dumps

Enhanced Dump: Utilities

- New Stand-Alone Dump utility
  - Dump is written to disk – either ECKD or SCSI
    - Type of all dump disks must match IPL disk type
    - Dump disks for first level systems must be entire ECKD volumes or SCSI LUNs
    - Dump disks for second level systems may be minidisk "volumes"
  - Creates a CP hard abend format dump
    - Reduces space and time required for stand-alone dump
- **DUMPLD2** utility can now process stand-alone dumps written to disk
- VM Dump Tool supports increased memory size in dumps

Enhanced Dump: Allocating Disk Space for Dumps

- Dumps are written to disk space allocated for spool
  - Kept there until processed with DUMPLD2 (or DUMPLOAD)
- Recommend allocating enough spool space for 3 dumps
  - See "Allocating Space for CP Hard Abend Dumps" in CP Planning and Administration manual
- CPOWNED statement
  - Recommend use of **DUMP** option to reserve spool volumes for dump space only
- **SET DUMP rdev**
  - Can specify up to 32 real device numbers of CP_Owned DASD
  - Order specified is the order in which they are searched for available space

Enhanced Dump: New Stand-Alone Dump Utility

- **SDINST EXEC** (new)
  - Used to create new stand-alone dump utility
  - For details:
    - Chapter 12, "The Stand-Alone Dump Facility", in CP Planning and Administration manual

- APAR VM65126 required to run **SDINST** second-level on z/VM 5.4 – 6.2 systems
  - PTF UM33687 for z/VM 5.4
  - PTF UM33688 for z/VM 6.1
  - PTF UM33689 for z/VM 6.2

Enhanced Dump: What is not Changed for Large Memory Dumps

- Old (pre-z/VM 6.3) stand-alone dump utility (HCPSADMP)
- DUMPLOAD
- VMDUMP

HiperDispatch

HiperDispatch

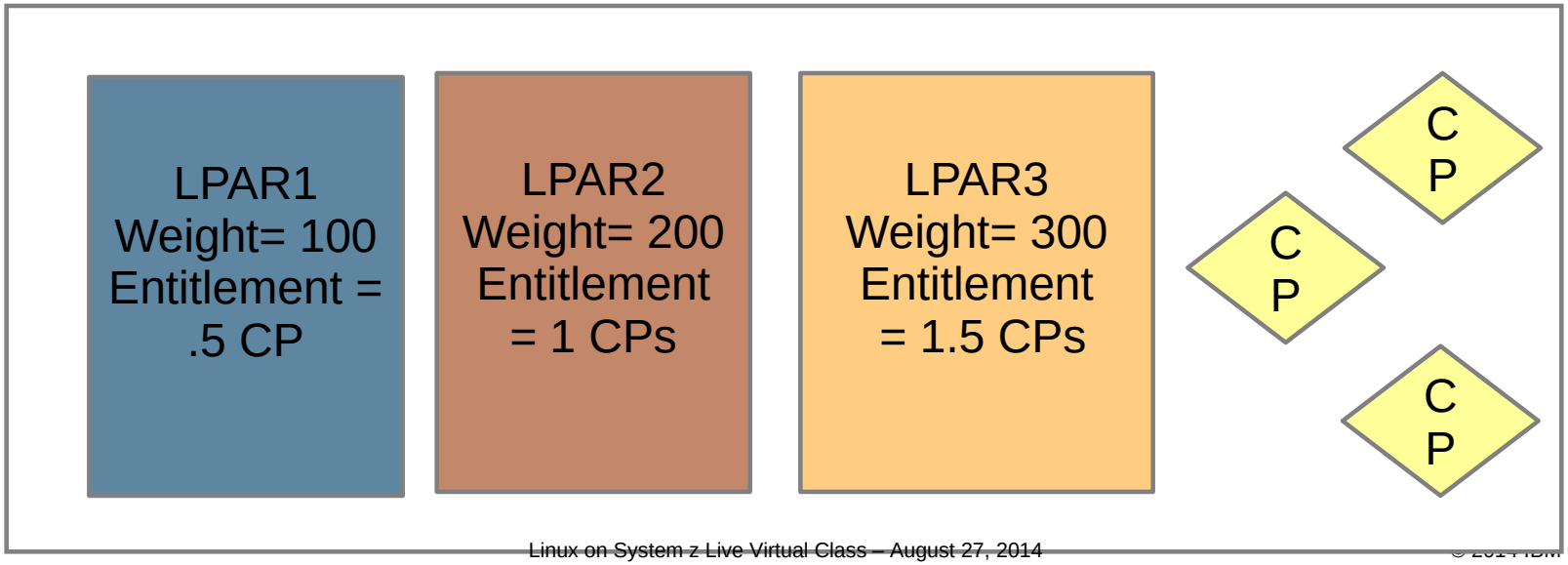
- Objective: Improve performance of guest workloads
  - z/VM 6.3 communicates with PR/SM to maintain awareness of its partition's topology
    - Partition Entitlement and excess CPU availability
    - Exploit cache-rich system design of System z10 and later machines
  - z/VM polls for topology information/changes every 2 seconds
- Two components
  - Dispatching Affinity
  - Vertical CPU Management
- For most benefit, Global Performance Data (GPD) should be on for the partition
  - Default is ON

HiperDispatch: System z Partition Entitlement

- The allotment of CPU time for a partition

- Function of
  - Partition's weight
  - Weights for all other shared partitions
  - Total number of shared CPUs

- Dedicated partitions
  - Entitlement for each logical CPU = 100% of one real CPU



HiperDispatch: Horizontal Partitions

- Horizontal Polarization Mode
  - Distributes a partition's entitlement evenly across all of its logical CPUs
  - Minimal effort to dispatch logical CPUs on the same (or nearby) real CPUs ("soft" affinity)
    - Affects caches
    - Increases time required to execute a set of related instructions
  - z/VM releases prior to 6.3 always run in this mode

HiperDispatch: Vertical Partitions

- Vertical Polarization Mode
  - Consolidates a partition's entitlement onto a subset of logical CPUs
  - Places logical CPUs topologically near one another
  - Three types of logical CPUs
    - Vertical High (Vh)
    - Vertical Medium (Vm)
    - Vertical Low (Vl)
  - z/VM 6.3 runs in vertical mode by default
    - First level only
    - Mode can be switched between vertical and horizontal
    - Dedicated CPUs are not allowed in vertical mode

HiperDispatch: Partition Entitlement vs. Logical CPU Count

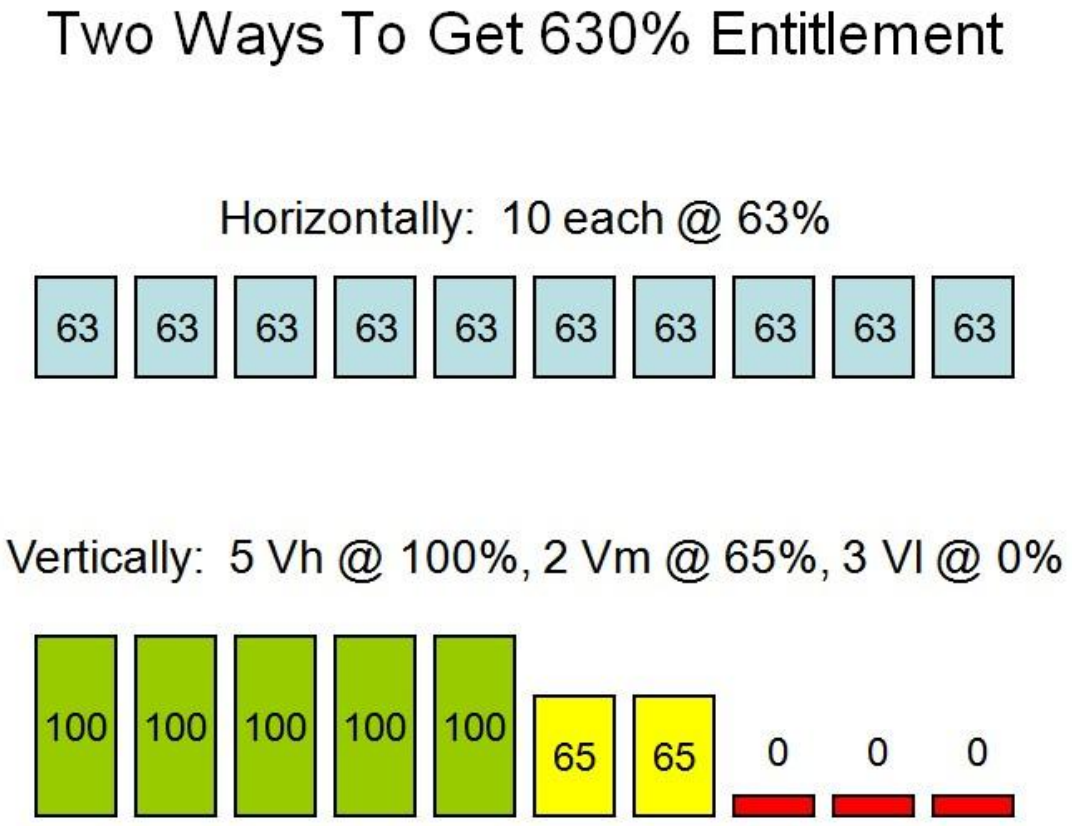
Suppose we have 10 IFLs shared by partitions FRED and BARNEY:

Partition	Weight	Weight Sum	Weight Fraction	Physical Capacity	Entitlement Calculation	Entitlement	Maximum Achievable Utilization
FRED, a logical **10-way**	**63**	100	63/100	1000%	1000% x (63/100)	**630%**	1000%
BARNEY, a logical **8-way**	**37**	100	37/100	1000%	1000% x (37/100)	**370%**	800%

For FRED to run *beyond* **630%** busy, BARNEY has to leave some of its entitlement *unconsumed.*

(CEC's excess power XP) = (total power TP) - (consumed entitled power EP).

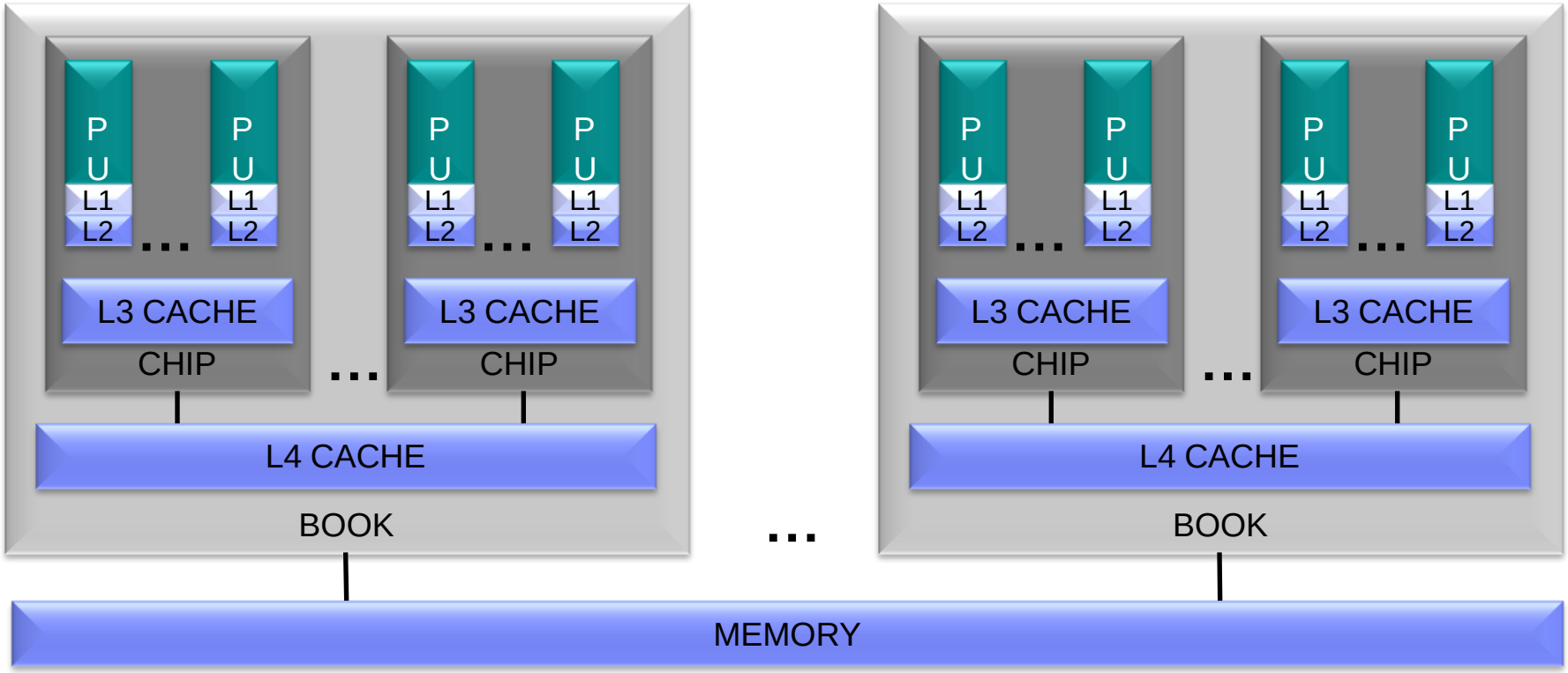
HiperDispatch: Horizontal and Vertical Partitions



In vertical partitions:

- Entitlement is distributed unequally among LPUs.
- Unentitled LPUs are useful only when other partitions are not using their entitlements.
- PR/SM tries very hard not to move Vh LPUs.
- PR/SM tries very hard to put the Vh LPUs close to one another.
- Partition consumes its XPF on its Vm and Vl LPUs.

HiperDispatch: Dispatching Affinity



- Processor cache structures have become increasingly complex and critical to performance
- z/VM 6.3 groups together the virtual CPUs of n-way guests
  - Dispatches guests on logical CPUs and in turn real CPUs that share cache
  - Goal is to re-dispatch guest CPUs on same logical CPUs to maximize cache benefits
  - Better use of cache can reduce the execution time of a set of related instructions

HiperDispatch: Parked Logical CPUs

- z/VM automatically parks and unparks logical CPUs
  - Based on usage and topology information
  - Only in vertical mode
- Parked CPUs remain in wait state
  - Still varied on
- Parking/Unparking is faster than **VARY OFF/ON**

HiperDispatch: Checking Parked CPUs and Topology

- **QUERY PROCESSORS** shows PARKED CPUs

```
PROCESSOR nn MASTER type
PROCESSOR nn ALTERNATE type
PROCESSOR nn PARKED type
PROCESSOR nn STANDBY type
```

- **QUERY PROCESSORS TOPOLOGY** shows the partition topology

```
q proc topology
13:14:59 TOPOLOGY
13:14:59   NESTING LEVEL: 02  ID: 01
13:14:59     NESTING LEVEL: 01  ID: 01
13:14:59       PROCESSOR 00  PARKED     CP    VH  0000
13:14:59       PROCESSOR 01  PARKED     CP    VH  0001
13:14:59       PROCESSOR 12  PARKED     CP    VH  0018
13:14:59     NESTING LEVEL: 01  ID: 02
13:14:59       PROCESSOR 0E  MASTER     CP    VH  0014
13:14:59       PROCESSOR 0F  ALTERNATE  CP    VH  0015
13:14:59       PROCESSOR 10  PARKED     CP    VH  0016
13:14:59       PROCESSOR 11  PARKED     CP    VH  0017
                     .
                     .
                     .
13:14:59   NESTING LEVEL: 02  ID: 02
13:14:59     NESTING LEVEL: 01  ID: 02
13:14:59       PROCESSOR 14  PARKED     CP    VM  0020
13:14:59     NESTING LEVEL: 01  ID: 04
13:14:59       PROCESSOR 15  PARKED     CP    VM  0021
13:14:59       PROCESSOR 16  PARKED     CP    VL  0022
13:14:59       PROCESSOR 17  PARKED     CP    VL  0023
```

HiperDispatch: New and Changed Commands

- **INDICATE LOAD**
  - AVGPROC now represents average value of the portion of a real CPU that each logical CPU has consumed

- **SET SRM** command can be used to change default settings for attributes related to HiperDispatch
  - Review monitor data and/or performance reports before changing

Virtual Networking

Virtual Networking: Live Guest Relocation Enhancements

- Live Guest Relocation supports port-based virtual switches
  - New eligibility checks allow safe relocation of a guest with a port-based VSwitch interface
  - Prevents relocation of an interface that will be unable to establish proper network connectivity
  - Adjusts the destination virtual switch configuration, when possible, by inheriting virtual switch authorization from the origin

Virtual Networking: VSwitch Recovery and Stall Prevention

- Initiate controlled port change or failover to a configured OSA backup port
  - Minimal network disruption

- SET VSWITCH UPLINK **SWITCHOVER** command
  - Switch to first available configured backup device
  - Switch to specified backup device
    - Specified RDEV and port number must already be configured as a backup device
  - If backup network connection cannot be established, original connection is reestablished
  - Not valid for a link aggregation or GROUP configured uplink port

Virtual Networking: VSwitch Support for VEPA Mode

- Virtual Edge Port Aggregator (VEPA)
  - IEEE 802.1Qbg standard
  - Provides capability to send all virtual machine traffic to the network switch
  - Moves all frame switching from CP to external switch
  - Relaxes "no reflection" rule
  - Supported on OSA-Express3 and later on zEC12 and later
- Enables switch to monitor and/or control data flow
- z/VM 6.3 support
  - New **VEPA OFF/ON** operand on SET VSWITCH command

Additional Information

- z/VM 6.3 resources
  - http://www.vm.ibm.com/zvm630/
  - http://www.vm.ibm.com/zvm630/apars.html
  - http://www.vm.ibm.com/events/

- z/VM 6.3 Performance Report
  - http://www.vm.ibm.com/perf/reports/zvm/html/index.html

- z/VM Library
  - http://www.vm.ibm.com/library/

- Licensing
  - IBM License Metric Tool 9.0.1
    - https://ibm.biz/cpupoolilmt
  - z/VM Software
    - http://www-03.ibm.com/systems/z/resources/swprice/zipla/zvm.html
  - Linux on System z Middleware
    - http://www-03.ibm.com/systems/z/resources/swprice/subcap/linux.html

- Live Virtual Classes for z/VM and Linux
  - http://www.vm.ibm.com/education/lvc/

Thanks!

John Franciscovich
z/VM Design and Development
Endicott, NY

francisj@us.ibm.com

Appendix A: Technology Exploitation

Crypto Express4S

- Available on zEC12 and zBC12
- Supported for z/Architecture guests
  - Authorized in directory (CRYPTO statement)
- Shared or Dedicated access when configured as
  - IBM Common Cryptographic Architecture (CCA) coprocessor
  - Accelerator
- Dedicated access only when configured as
  - IBM Enterprise Public Key Cryptographic Standards (PKCS) #11 (EP11) coprocessor

FCP Data Router (QEBSM)

- Allows guest exploitation of the Data Router facility
  - Provides direct memory access (DMA) between an FCP adapter's SCSI interface and real memory
  - Guest must enable the Multiple Buffer Streaming Facility when establishing its QDIO queues

- **QUERY VIRTUAL FCP** command indicates whether
  - Device is eligible to use Data Router facility
    - **DATA ROUTER ELIGIBLE**
  - Guest requested use of Data Router facility when transferring data
    - **DATA ROUTER ACTIVE**

- Monitor record updated:
  - Domain 1 Record 19 – MRMTRQDC – QDIO Device Configuration Record

FICON DS8000 and MSS Support

- FICON DS8000 Series New Functions
  - Storage Controller Health message
    - New attention message from HW providing more details for conditions in past reflected as Equipment Check.
    - Intended to reduce the number of false HyperSwap events.
  - Peer-to-Peer Remote Copy (PPRC) Summary Unit Check
    - Replaces a series of state change interrupts for individual DASD volumes with a single interrupt per LSS
    - Intended to avoid timeouts in GDPS environments that resulted from the time to process a large number of state change interrupts
- Multiple Subchannel Set (MSS) support for mirrored DASD
  - Support to use MSS facility to allow use of an alternate subchannel set for Peer-to-Peer Remote Copy (PPRC) secondary volumes
  - New **QUERY MSS** command
  - New MSS support cannot be mixed with older z/VM releases in an SSI cluster

Satisfies SODs from October 12, 2011

Appendix B: Miscellaneous Enhancements

IPL Changes for NSS in a Linux Dump

- Allows contents of NSS to be included in dumps created by stand-alone dump tools such as Linux Disk Dump utility
  - New **NSSDATA** operand on IPL command

- **NSSDATA** can only be used if the NSS:
  - is fully contained within the first extent of guest memory
  - does not contain SW, SN or SC pages
  - is not a VMGROUP NSS

- See http://www.vm.ibm.com/perf/tips/vmdump.html for information on differences between VMDUMP and Linux Disk Dump utility

Specify RDEV for System Volumes

- Prevents wrong volume from being attached when there are multiple volumes with the same volid
- Optionally specify RDEV along with volid in system configuration file
  - **CP_OWNED** statement
  - **USER_VOLUME_RDEV** statement (new)
- If specified, disk volume must match both in order to be brought online
- No volume with specified volid is brought online when
  - Volume at RDEV address has a different volid than specified
  - There is no volume at specified RDEV address

Cross System Extensions (CSE) Withdrawn in z/VM 6.3

- Function has been replaced by z/VM Single System Image (VMSSI) feature
  - **XSPOOL** … commands no longer accepted
  - **XSPOOL_** … configuration statements not processed (tolerated)
- CSE cross-system link function is still supported
  - **XLINK** … commands
  - **XLINK_** … configuration statements
- CSE XLINK and SSI shared minidisk cannot be used in same cluster
- Satisfies Statement of Direction (October 12, 2011)

OVERRIDE Utility and UCR Function Withdrawn

- "Very OLD" method for redefining privilege classes for
  - CP Commands
  - Diagnose codes
  - other CP functions

- To redefine privilege classes, use
  - **MODIFY COMMAND** command and configuration statement
  - **MODIFY PRIV_CLASSES** command and configuration statement

- Satisfies Statement of Direction (October 12, 2011)