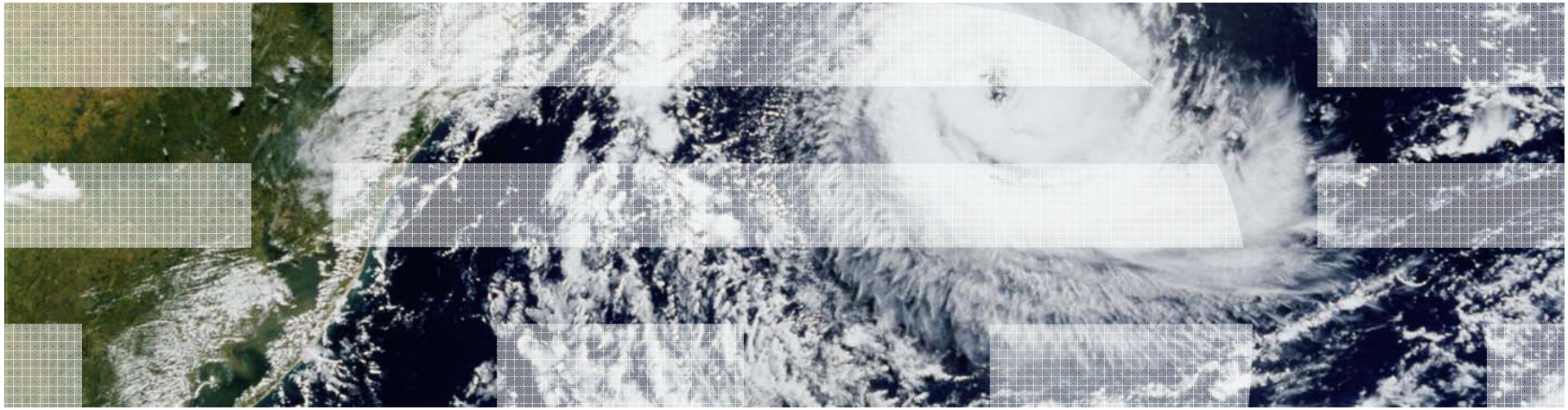
Romney White
IBM System z Architecture and Technology
romneyw@us.ibm.com

z/VM 6.3 Scalability

GSE Hamburg – October, 2013



Topics

- **z/VM 6.3 Themes**
- **Scalability and Performance**
  - Large Memory Support
  - Enhanced Dump Support
  - HiperDispatch
- **References**
- **Q&A**

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 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

Large Memory Support

Large Memory Support

- **Support for up to 1TB of real memory (increased from 256 GB)**
  - Proportionally increases total virtual memory
  - Individual virtual machine limit of 1TB 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

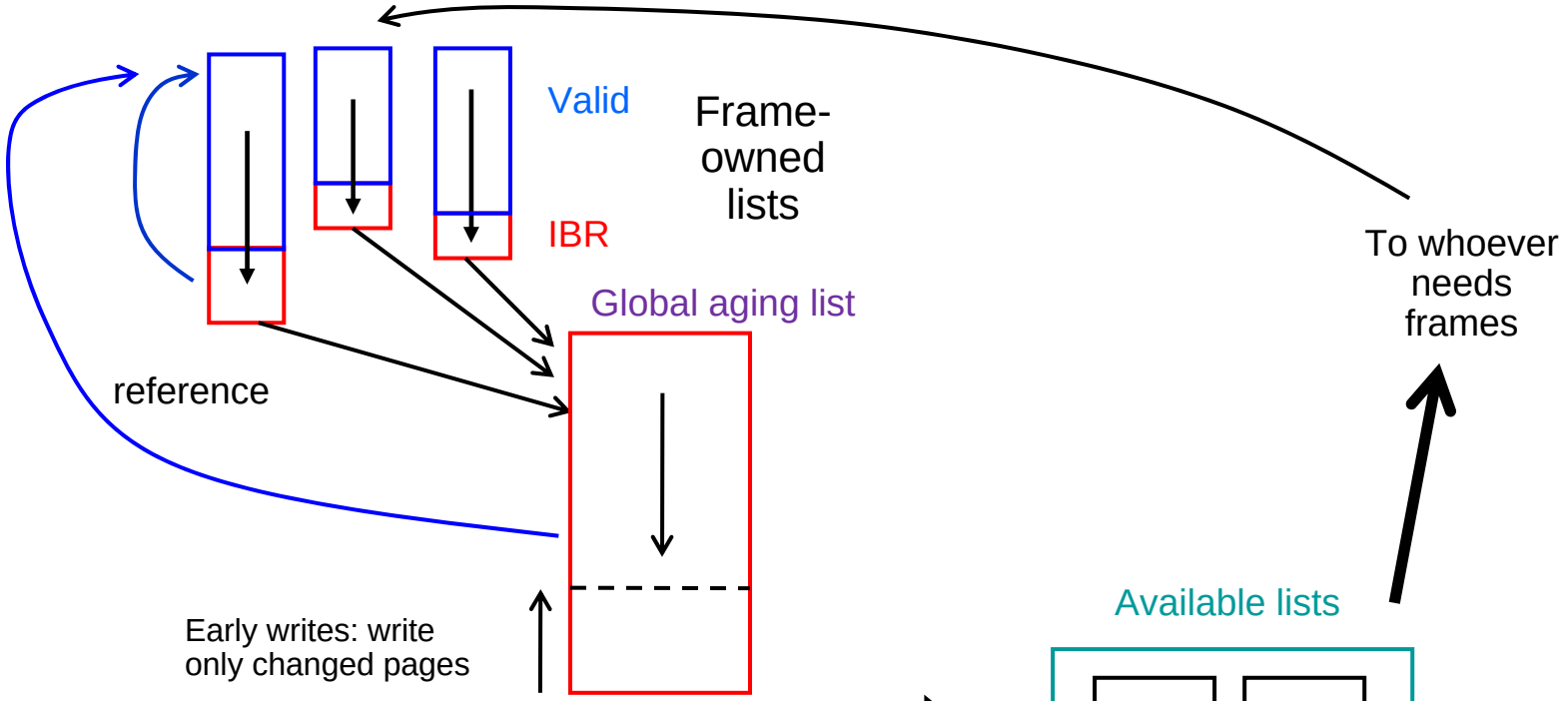
Large Memory Support: Reserved Storage

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

- SET RESERVED **command is enhanced**
  - 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
    - A segment does not need to be loaded in order to SET RESERVED for it
    - Can be used for monitor segment (MONDCSS)
  - Can define number of frames or storage size to be reserved
  - **SYSMAX** operand defines maximum amount of storage that can be reserved for system
    - CP SET RESERVED command or STORAGE RESERVED config statement

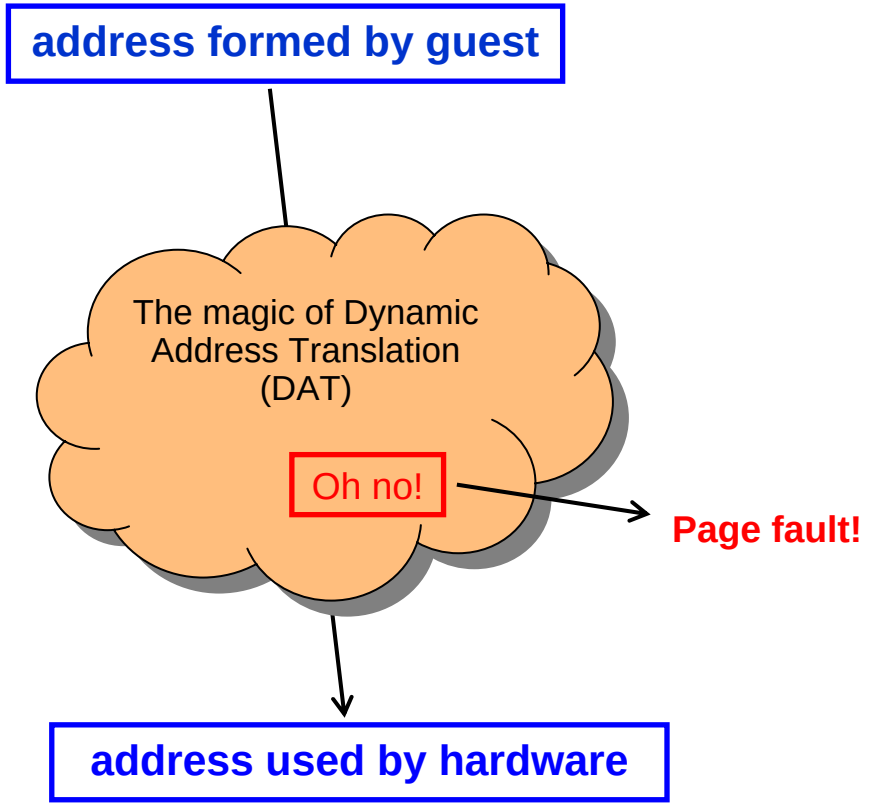
- **Reserved settings do not survive IPL**

Large Memory Support: The Big State Diagram



Demand scan pushes frames
- ***From Frame-owned valid sections, down to…***
- ***Framed-owned IBR sections, then down to…***
- ***Global aging list, then back to…***
- ***Frame-owned lists, or over to …***
- ***Available lists, from which they...***
- ***are used to satisfy requests for frames***

Large Memory Support: Trial Invalidation

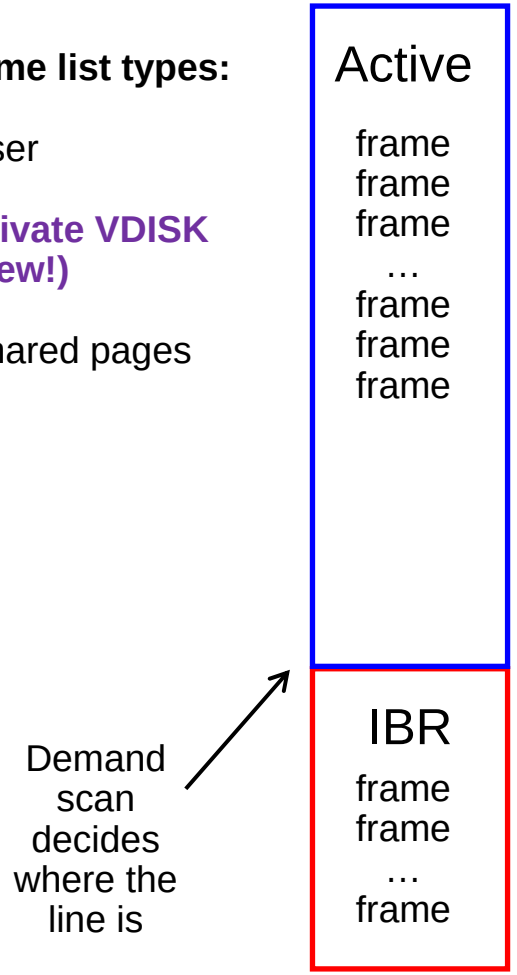


- **Page table entry (PTE) contains an "invalid" bit**
- **What if we**
  - Keep the PTE intact but set the "invalid" bit
  - Leave the frame contents intact
  - Wait for the guest to touch the page
- **A touch will cause a page fault, but…**
- **On a fault, there is nothing really to do except**
  - Clear the "invalid" bit
- **We call this** trial invalidation

Large Memory Support: Two-Section Frame-Owned Lists

Frame list types:

- User
- **Private VDISK (new!)**
- Shared pages

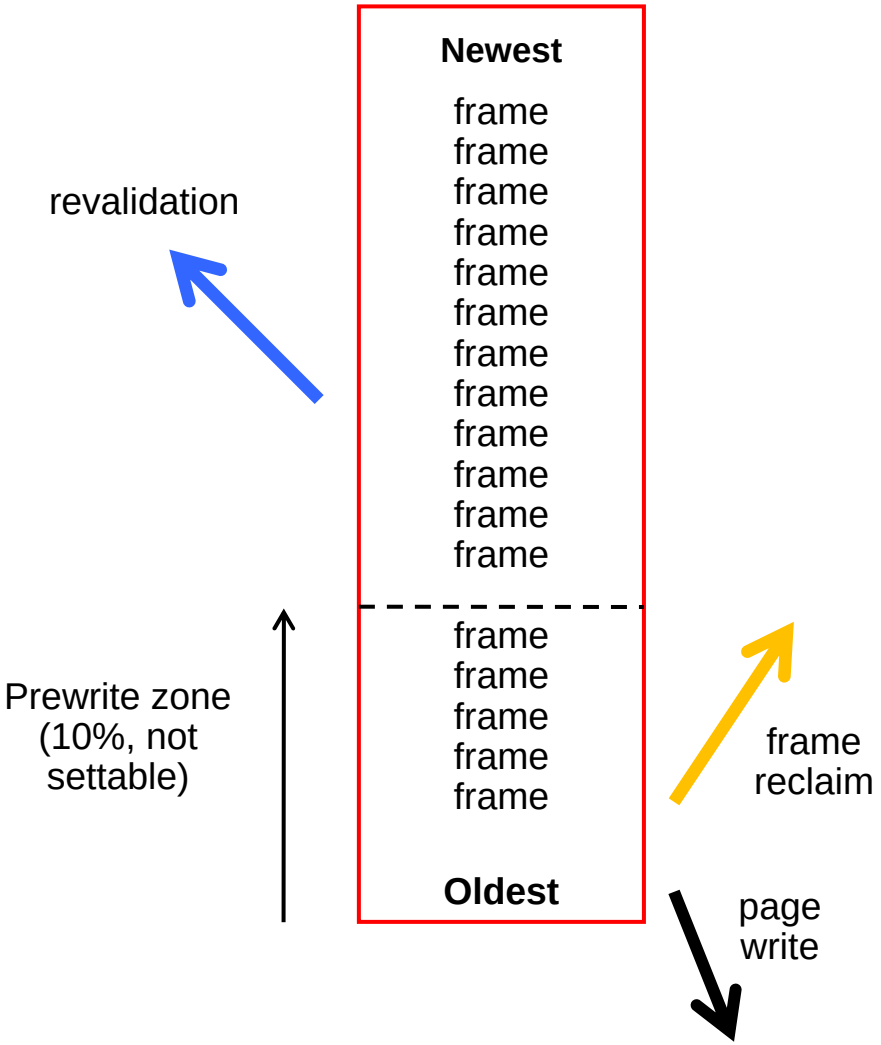


Active: Frames that are in use
- Roughly in order by when they became valid

A frame list is no longer ever searched, sorted, or reordered

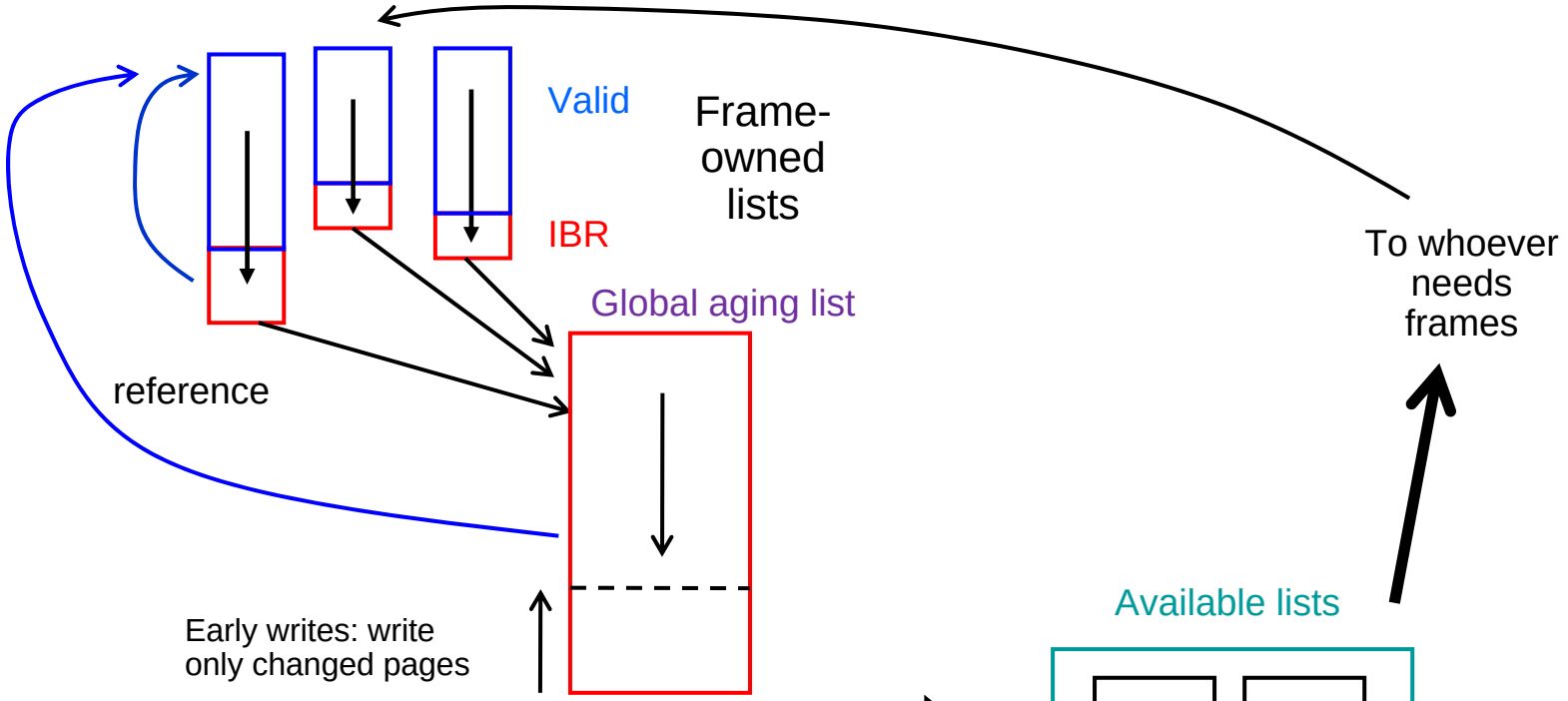
IBR: invalid but resident
- Marked invalid in page table entry
- If referenced, moves to top of active section
- Allows detecting lack of reference
- “We try to keep [it] rather small”
- Influenced by revalidation rate

Large Memory Support: Global Aging List



- Size of global aging list can be specified but is best left to the system to manage
- All pages here are IBR
- Demand scan fills from the top
- Revalidated pages return to their owned-lists
- Changed pages are pre-written up from bottom of list
- Global aging list accomplishes age-filtering process that XSTORE used to provide
- No longer suggest XSTORE for paging, but will use it if configured

Large Memory Support: The Big State Diagram



Demand scan pushes frames
- ***From Frame-owned valid sections, down to…***
- ***Framed-owned IBR sections, then down to…***
- ***Global aging list, then back to…***
- ***Frame-owned lists, or over to …***
- ***Available lists, from which they...***
- ***are used to satisfy requests for frames***

Large Memory Support: Reorder

- **Reorder processing has been removed**
  - Commands remain for compatibility but have no effect
    - **SET REORDER** command gives RC=6005, “not supported”
    - **QUERY REORDER** command says it’s OFF
  - Monitor data no longer recorded

Large Memory Support: New/Changed Commands

Concept	Command	Comments
Size of the global aging list **Early writes allowed**	**Command:** **SET AGELIST …** **Config file:** **STORAGE AGELIST …** **Lookup:** **QUERY AGELIST**	**Sets size of global aging list:** **- A fixed amount (e.g., GB)** **- A percentage of DPA** **Default is 2% of DPA** **Determines if early writes allowed (if storage-rich, say NO)**
Amount of storage reserved for a user or for a DCSS	**Command:** **SET RESERVED …** **Config file:** **STORAGE RESERVED …** **Lookup:** **QUERY RESERVED …**	**You can set RESERVED for:** **- A user, a NSS, or a DCSS** **You can also set a SYSMAX on total RESERVED storage** **Config file can set only SYSMAX**

Large Memory Support: INDICATE Command Changes

Command	Comments
INDICATE LOAD	**STEAL-nnn% field no longer appears in output**
INDICATE NSS	**Includes a new “instantiated” count (number of pages that exist)** **Sum of locus counts might add to more than “instantiated”**
INDICATE USER	**Includes a new “instantiated” count** **Sum of locus counts might add to more than “instantiated”**
INDICATE SPACES	**Includes a new “instantiated” count**

Large Memory Support: Planning DASD Paging Space

- **Calculate the sum of**
  - Logged-on virtual machine primary address spaces
  - Any data spaces they create
  - Any VDISKs they use
  - Total number of shared NSS or DCSS pages
- **Multiply by 1.01 to allow for PGMBKs and friends**
- **Add to that sum**
  - Total number of CP directory pages (reported by DIRECTXA)
  - Min (10% of central, 4 GB) to allow for system-owned virtual pages
- **Multiply by safety factor (e.g., 1.25) to allow for growth or uncertainty**
- **Remember that your system will abend (PGT004) if you run out of paging space**
  - Consider using something that alerts on page space utilization, such as Operations Manager for z/VM

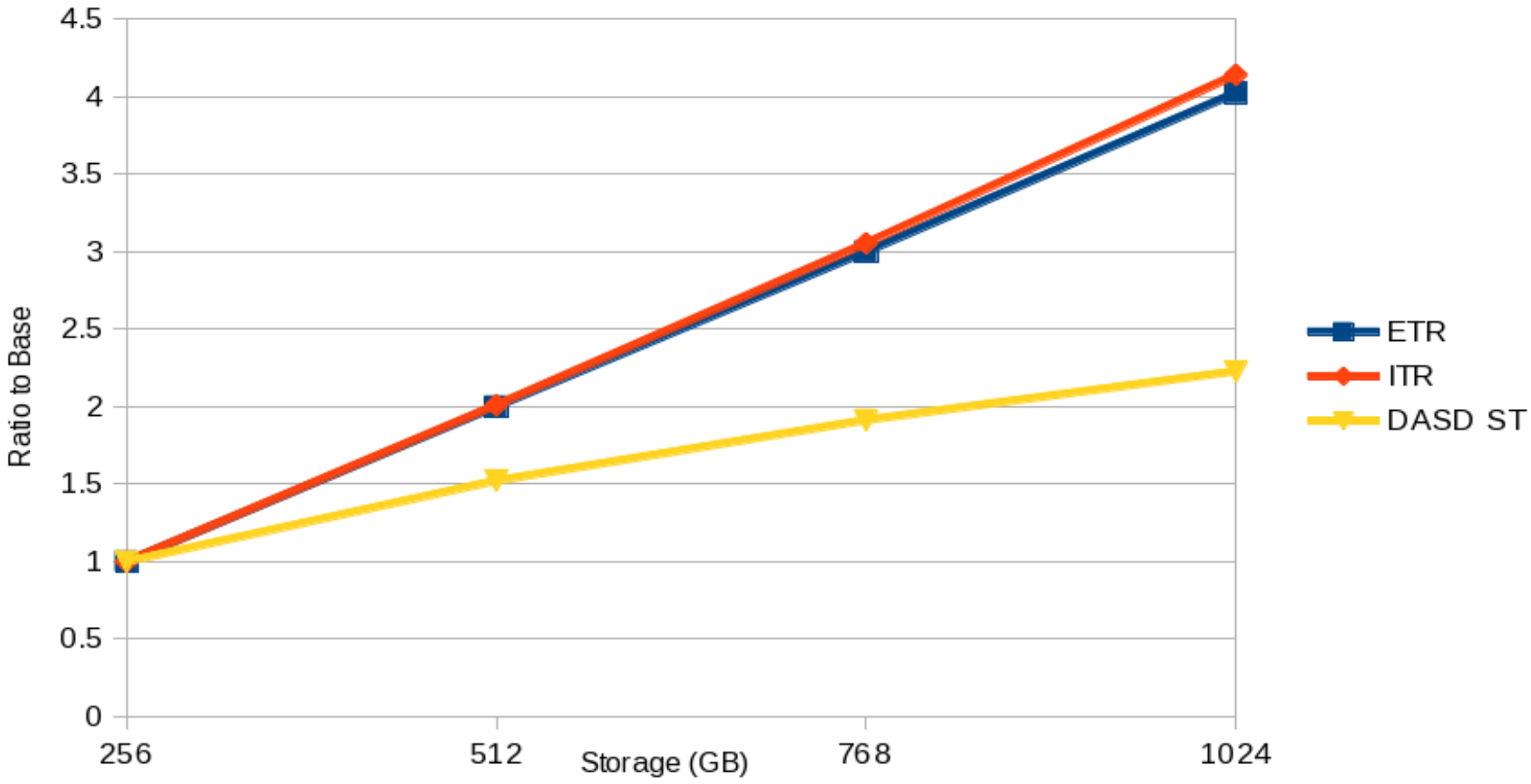
The “Sweet Spot” Workload

A synthetic workload called *Sweet Spot* imitates behaviors we have seen in customer-supplied MONWRITE data

	z/VM 6.2	z/VM 6.3	Delta	Pct. Delta
Cstore	**256**	**384**	**128**	
Xstore	**128**	**0**	**-128**	
External Throughput (ETR)	**0.0746**	**0.0968**	**0.0222**	**29.8%**
Internal Throughput (ITR)	**77.77**	**105.60**	**27.83**	**35.8%**
System Util/Proc	**31.4**	**4.7**	**-26.7**	**-85.0%**
T/V Ratio	**1.51**	**1.08**	**-0.43**	**-28.5**

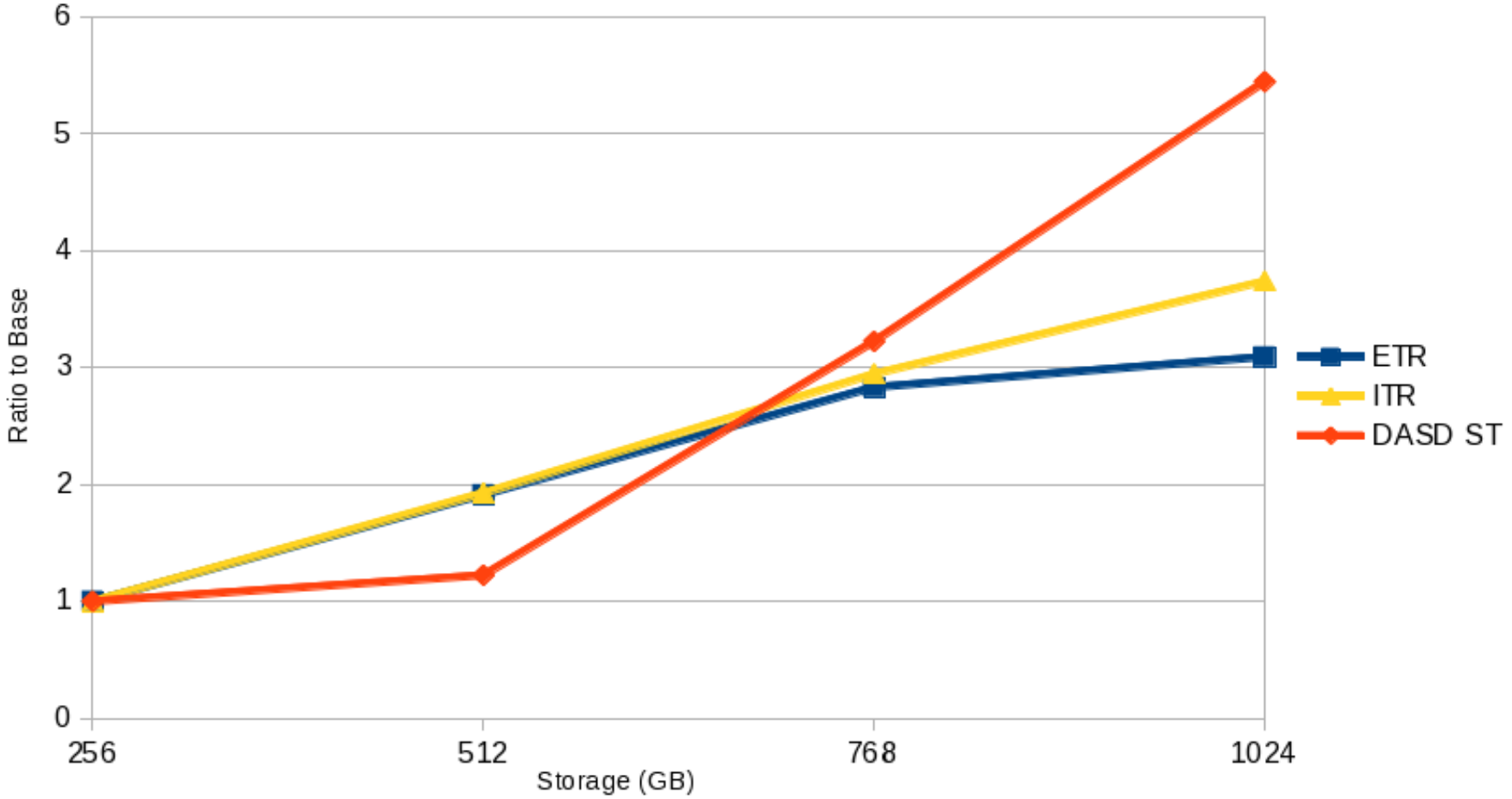
By getting rid of both reorder and spin lock contention we achieved huge drops in %CPU and T/V

VIRSTOR Workload in Overcommitted Environment



ETR = External Throughput; ITR = Internal Throughput; DASD ST = DASD Service Time

Apache Workload in Overcommitted Environment



ETR = External Throughput; ITR = Internal Throughput; DASD ST = DASD Service Time

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 three dumps**
  - See "Allocating Space for CP Hard Abend Dumps" in CP Planning and Administration manual
  - http://www.vm.ibm.com/service/zvmpladm.pdf
- **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 Unchanged for Large Memory Dumps

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

HiperDispatch

HiperDispatch

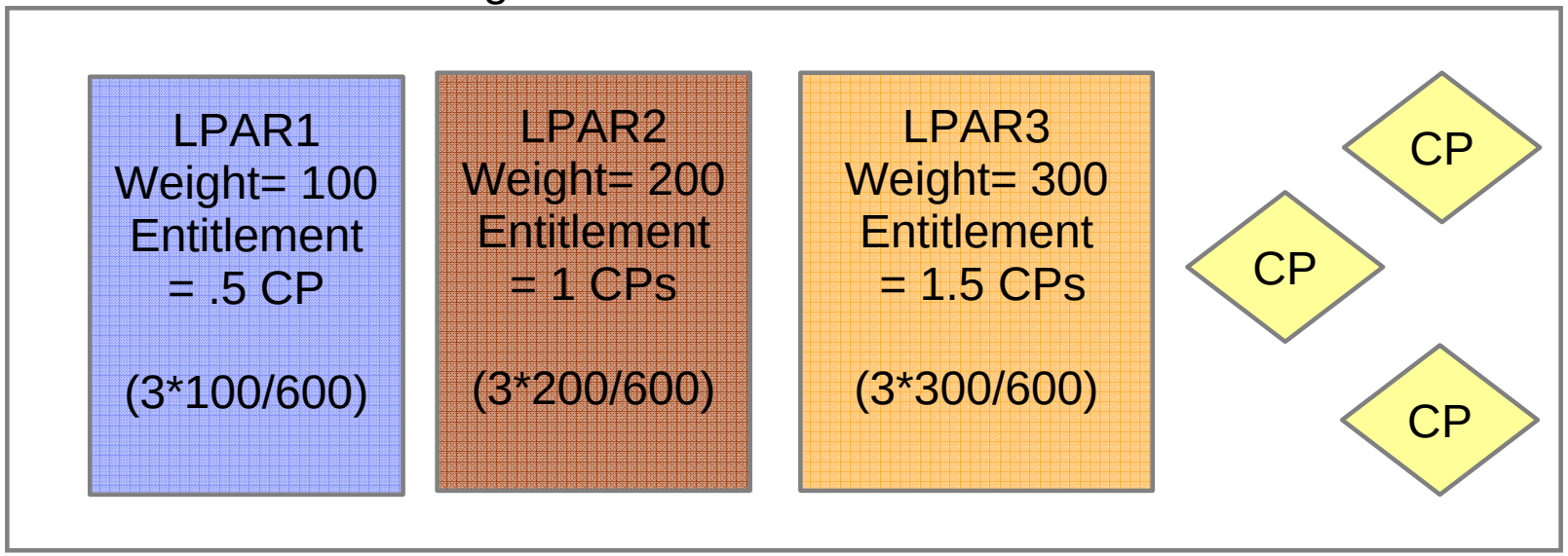
- **Objective: Improve performance of guest workloads**
  - z/VM 6.3 communicates with PR/SM to maintain awareness of its partition topology
    - Partition Entitlement and excess CPU availability
    - Exploit cache-rich system design of System z10 and later machines
  - z/VM polls for topology information/changes every 2 seconds

- **Two components**
  - Dispatching Affinity
  - Vertical CPU Management

- **For most benefit, Global Performance Data (GPD) should be on for the partition in its Activation Profile**
  - Default is ON

HiperDispatch: System z LPAR Entitlement

- **The allotment of CPU time for an LPAR**
- **Function of**
  - LPAR's weight
  - Weights for all other shared LPARs
  - Total number of shared CPUs
- **Dedicated CPU partitions**
  - Entitlement for each logical CPU = 100% of one real CPU



HiperDispatch: Partition Entitlement vs. Logical CPU Count

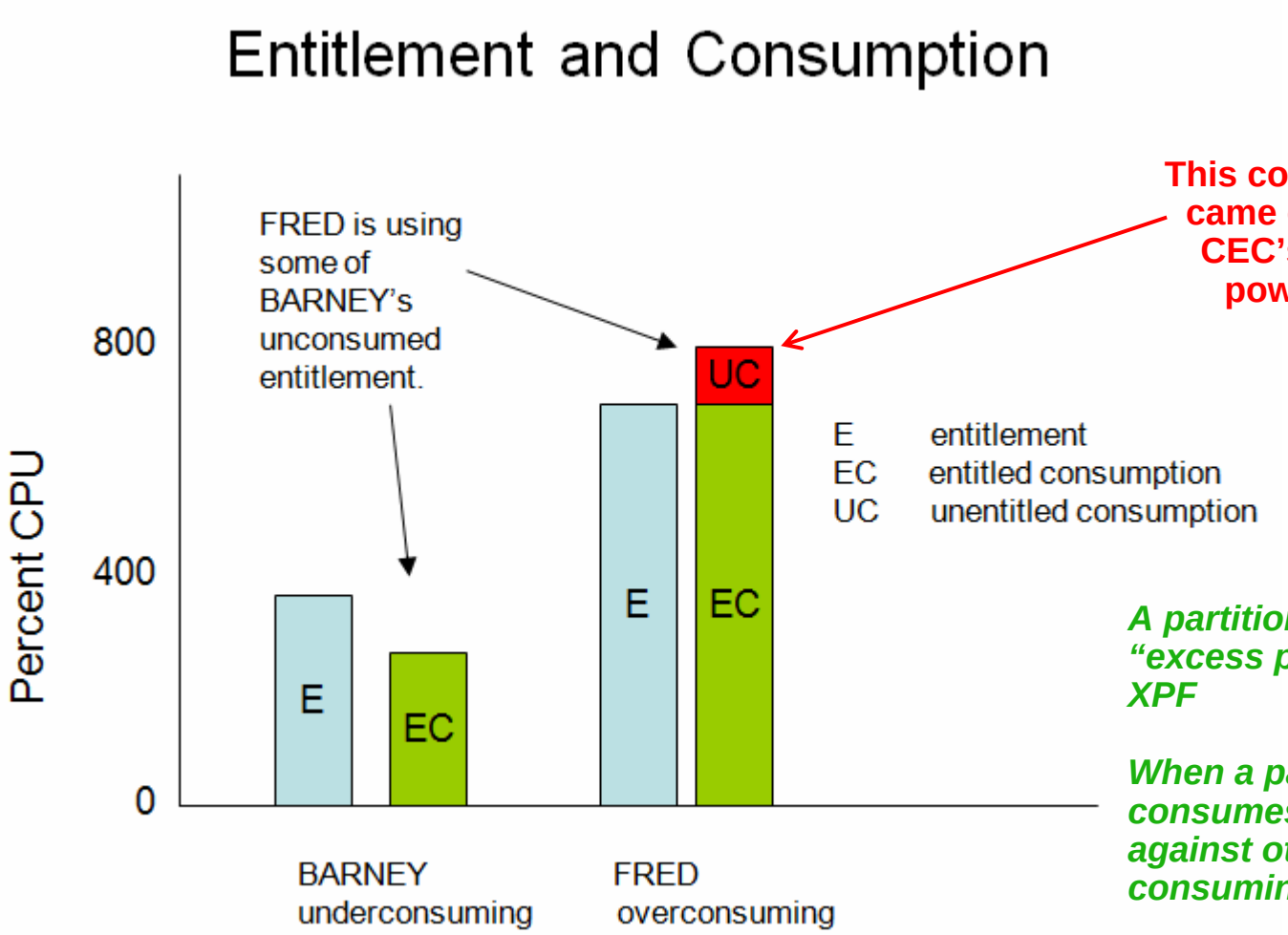
Suppose we have 10 IFLs shared by partitions FRED and BARNEY:

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

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

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

HiperDispatch: Entitlement and Consumption



A partition's share of XP is its "excess power fraction" or XPF

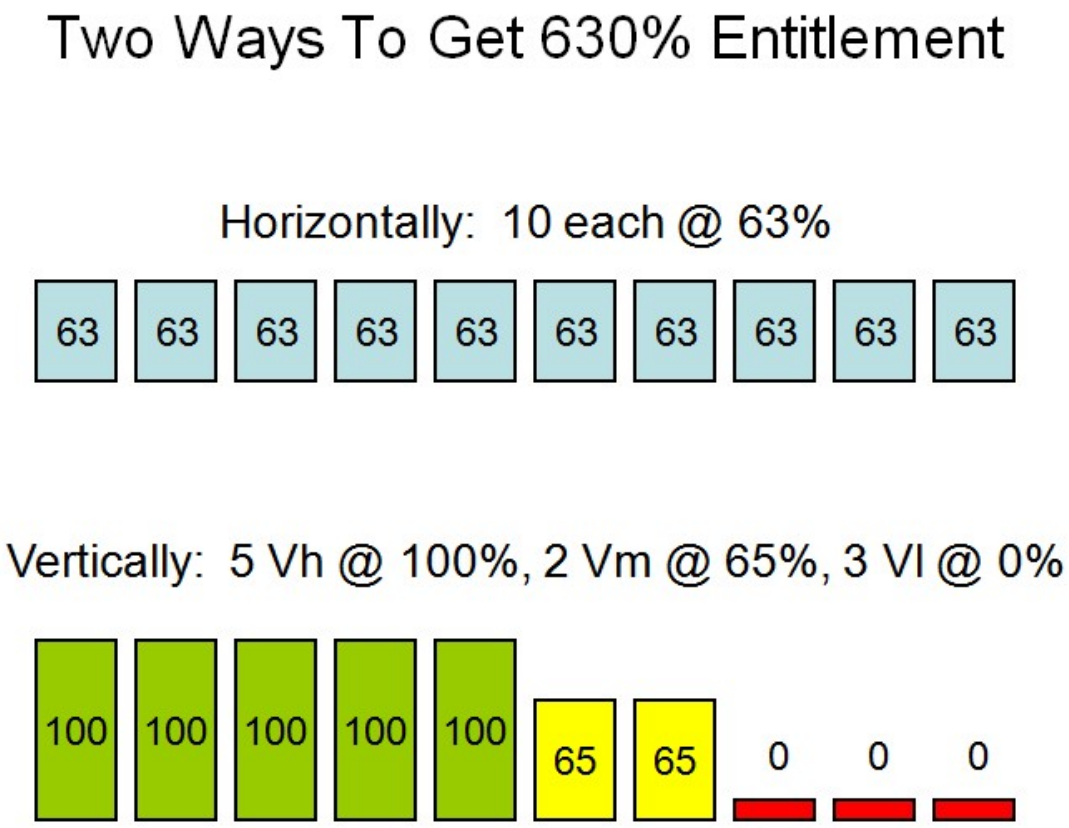
When a partition over consumes, it competes against other over-consuming partitions for XP

HiperDispatch: Horizontal and Vertical 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 cache effectiveness
    - Can increase time required to execute a set of related instructions
  - z/VM releases prior to 6.3 always run in this mode

- **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)

HiperDispatch: Horizontal and Vertical Partitions



In vertical partitions:

- Entitlement is distributed unequally among LPUs

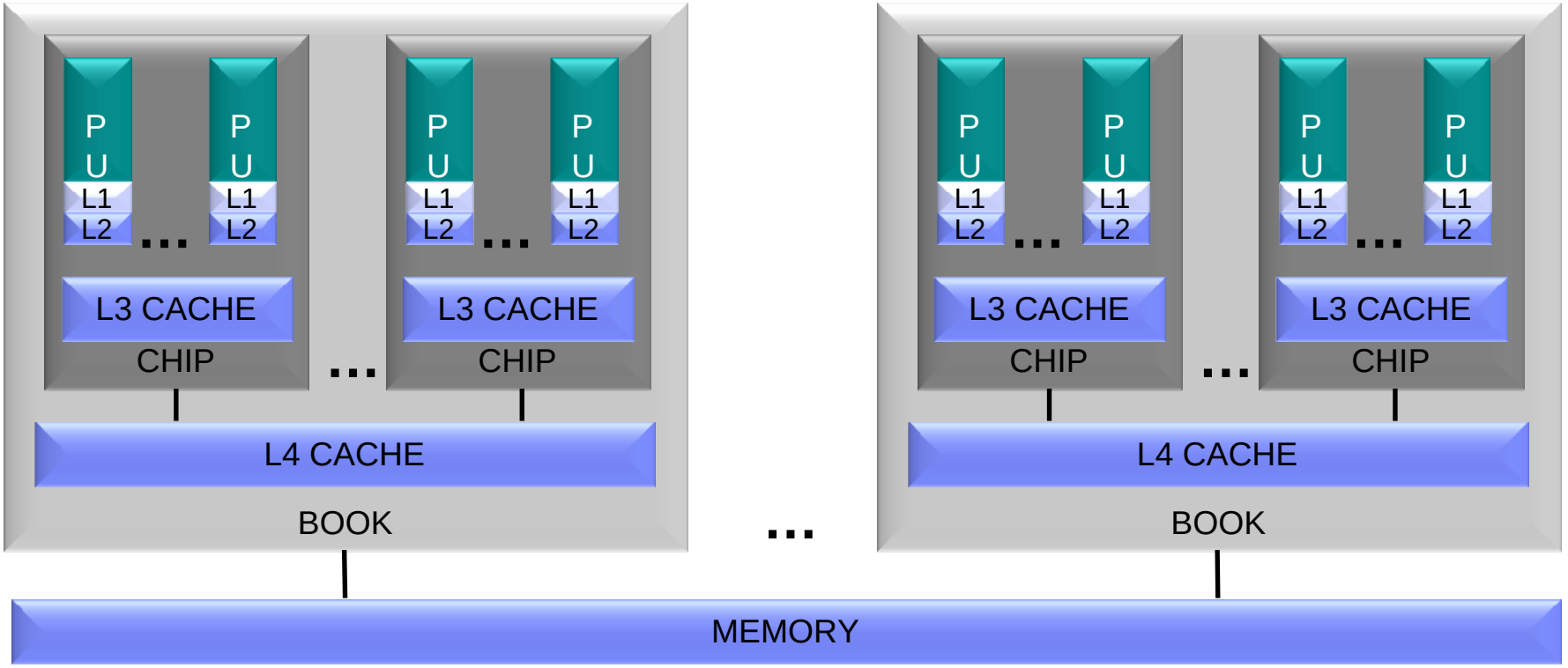
- Unentitled LPUs are useful only when other partitions are not using their entitlements

- 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: Vertical Polarization Mode

- **z/VM monitors CPU use in its LPAR as well as others to predict CPU demand and project whether excess CPU power will be available**
  - Determines the best number of CPUs for consuming the available power
  - Determines which logical CPUs should be in use
    - Unnecessary CPUs are put into new "parked" state

- **z/VM 6.3 runs in vertical mode by default**
  - Mode can be switched between vertical and horizontal
    - New **POLARIZATION** option of SET SRM command and SRM statement
  - Vertical mode is not permitted for second-level z/VM systems

- DEDICATE **command or directory statement not allowed in vertical mode**
  - Cannot switch to vertical mode if there are dedicated CPUs

HiperDispatch: Parked Logical CPUs

- **z/VM automatically parks and unparks logical CPUs**
  - Based on use 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

- QUERY PROCESSORS **shows parked CPUs**

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

HiperDispatch: Checking Topology

- QUERY PROCESSORS TOPOLOGY **shows 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: Other Changes

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

- **Monitor records – new and updated**

- **z/VM Performance Toolkit – new and updated reports**

HiperDispatch: Knobs

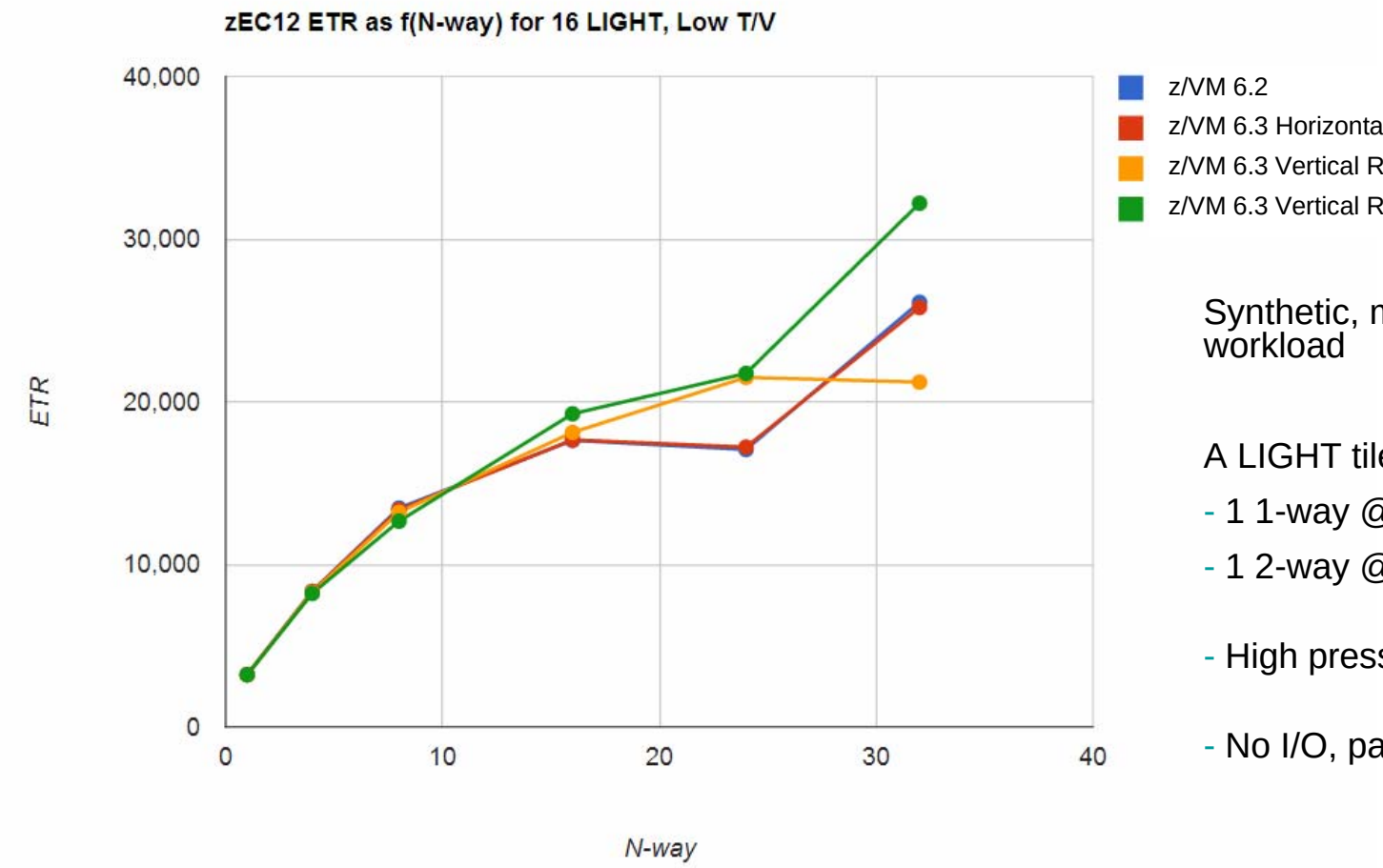
Concept	Knob
Horizontal or vertical	`SET SRM POLARIZATION { HORIZONTAL | VERTICAL }`
How optimistically to predict XPF floors	`SET SRM [TYPE cpu_type] EXCESSUSE { HIGH | MED | LOW }`
How much CPUPAD safety margin to allow when parking below available power	`SET SRM [TYPE cpu_type] CPUPAD nnnn%`
Reshuffle or rebalance	`SET SRM DSPWDMETHOD { RESHUFFLE | REBALANCE }`

Defaults

- Vertical mode
- EXCESSUSE MEDIUM (70%-confident floor)
- CPUPAD 100%
- Reshuffle

CP Monitor has been updated to report changes to these new SRM settings

Memory-Touching Workload, Light Edition



Synthetic, memory-touching workload

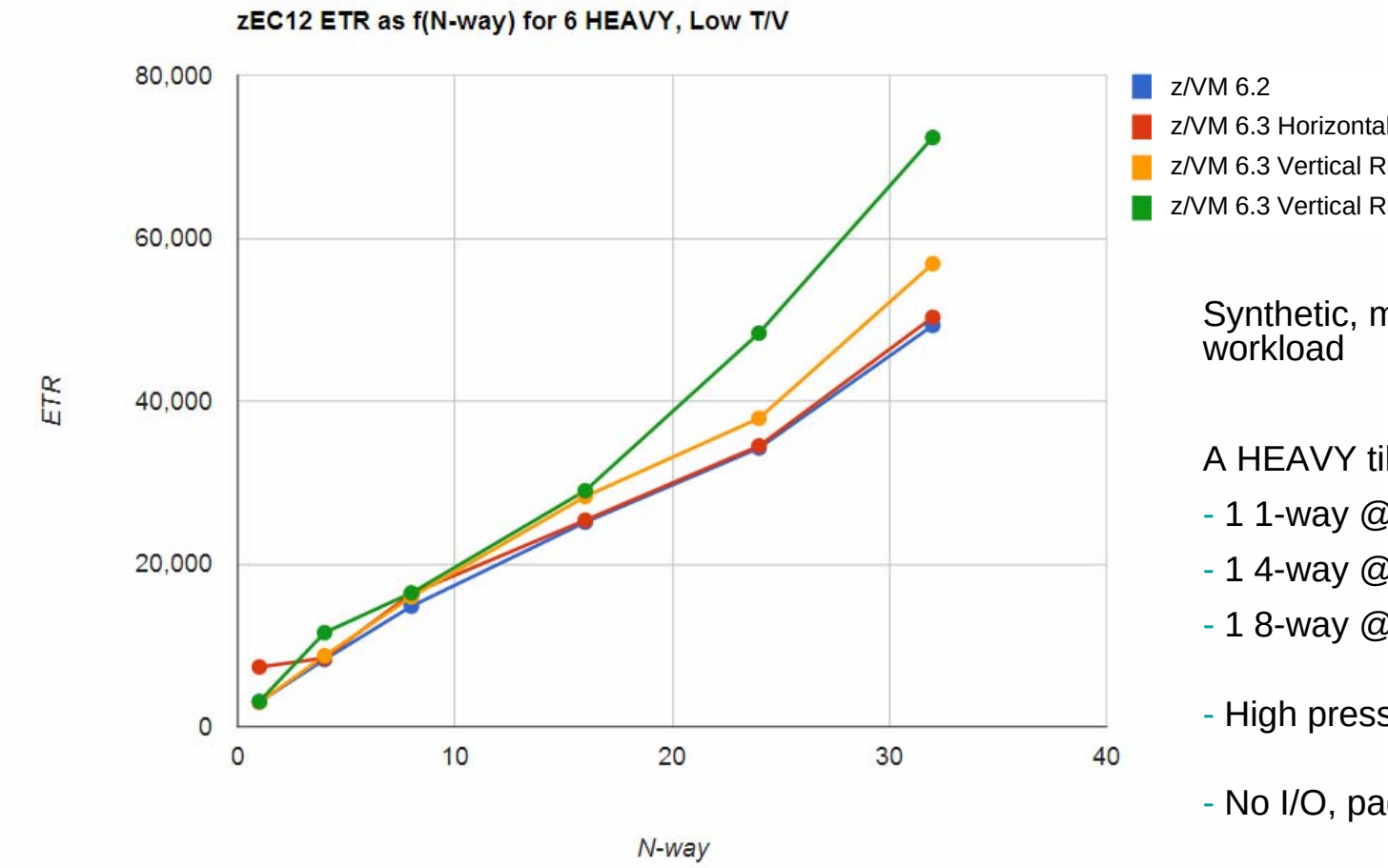
A LIGHT tile is 81% busy:

- 1 1-way @ 15% each
- 1 2-way @ 33% each

- High pressure on nest

- No I/O, paging, etc.

Memory-Touching Workload, Heavy Edition



Synthetic, memory-touching workload

A HEAVY tile is 540% busy:

- 1 1-way @ 15% each
- 1 4-way @ 31% each
- 1 8-way @ 50% each

- High pressure on nest

- No I/O, paging, etc.

Comments on Workloads

- **Workloads amenable to z/VM HiperDispatch**
  - High-CPU, CPU-constrained workloads (CPI)
  - Active VCPU:LCPU ratio not too large (context switches)
  - Runs in a partition with multiple topology containers (affinity)
- **Workloads indifferent to z/VM HiperDispatch**
  - Constrained by something else (e.g., I/O)
  - Memory-overcommitted
  - High Virtual:Logical processor ratio with all low activity virtual CPUs
  - Workloads with poor memory access habits
- **Remember that vertical mode isolates your partition**

More Information

z/VM 6.3 resources

http://www.vm.ibm.com/zvm630/
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/

Live Virtual Classes for z/VM and Linux

http://www.vm.ibm.com/education/lvc/


Efficiency of One. Flexibility of Many.
40 Years of Virtualization.