

Brian K. Wade, Ph.D.
Senior Software Engineer
IBM Corporation



z/VM and the IBM z13

2015
IBM z Systems
Technical University
18-22 May | Dublin, Ireland



© Copyright IBM Corporation 2015

Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Acknowledgements

- Kevin Adams
- Bill Bitner
- Sue Farrell
- John Franciscovich
- Emily Hugenbruch
- Mark Lorenc
- Angelo Macchiano
- Xenia Tkatschow
- Brian Wade
- Romney White

- ... and any contributor I might have omitted

Topics

- Overview of z/VM support for the IBM z13™

- z/VM enhancements to exploit z13 features
 - Simultaneous multithreading (SMT)
 - Increased processor scalability
 - Multi-VSWITCH link aggregation
 - Crypto enhancements

z/VM Support for the IBM z13

Expanding the Horizon of Virtualization

- Release for announcement – the IBM z13™
 - January 14, 2015
 - [Announcement link](#)

- z/VM compatibility support
 - PTFs available February 13, 2015
 - Also includes crypto enhanced domain support
 - z/VM 6.2 and z/VM 6.3
 - No z/VM 5.4 support
 - [Refer to bucket for full list](#)

- Enhancements and exploitation support on only z/VM 6.3
 - IBM z13 Simultaneous Multithreading
 - Increased processor scalability
 - Multi-VSWITCH link aggregation support (link aggregation with shared OSAs)



z/VM Support for IBM z13

- **Updates for z/VM 6.2 and 6.3**
 - Many components affected

- No z/VM 5.4 support

- No z/VM 6.1 support even if you have extended support contract.

- **PSP Bucket**
 - Upgrade **2964DEVICE**
 - Subset **2964/ZVM**

- **If running Linux, please also check for required updates prior to migration.**



z/VM Service Required for the IBM z13

<http://www.vm.ibm.com/service/vmreqz13.html>

United States [change]

Home Solutions Services Products Support & downloads My IBM

Welcome [IBM Sign in] [Register]

IBM Systems > System z > z/VM >

z/VM service required to run on the IBM z13

Last updated: January 14, 2015

The table below provides you with a list of service required for z/VM V6.3 and V6.2 to run on the IBM z13.

Note: Refer to the the 2964ZVM subset of the 2964DFV1CF bucket.

z/VM service required to run on the IBM z13		
APAR Number	z/VM Releases	Description
VM65577	z/VM V6.3 z/VM V6.2	Provides z/VM support that will enable guests to exploit IBM zEnterprise EC12 function on the IBM z13
VM65577	z/VM V6.3 z/VM V6.2	Provides support for the new Crypto Express5S adapter and enhanced domain support for Crypto Express4S and Crypto Express5S
VM65586	z/VM V6.3	Provides host exploitation support for SMT on IBM z13, which will enable z/VM to dispatch work on up to two threads (logical CPUs) of an IFL processor core
VM65676 VM65677	z/VM V6.3	Provides SMT stand-alone dump support
VM65586	z/VM V6.3	Provides support for up to 64 logical processors on IBM z13
VM65583 PI21053	z/VM V6.3	Provides Multi-VSwitch Link Aggregation Support, allowing a port group of OSA-Express features to span multiple virtual switches within a single z/VM system or between multiple z/VM systems
VM65670	z/VM V6.3	Provides SMAPI support for Multi-VSwitch Link Aggregation
VM65568	z/VM V6.3 z/VM V6.2	z/VM IOCP support for z13
VM65527	z/VM V6.3 z/VM V6.2	Performance ToolKit compatibility support for z13
VM65528	z/VM V6.3	Performance ToolKit support for simultaneous multithreading on z13
VM65529	z/VM V6.3	Performance ToolKit support for Multi-VSwitch Aggregation on z13
VM65588	z/VM V6.3 z/VM V6.2	DirMaint support for enhanced crypto domain support on z13
VM65489	z/VM V6.3 z/VM V6.2	VMHCD support for z13
VM65658	z/VM V5.4	VMHCD toleration support for z13 IODF
VM64437	z/VM V6.3 z/VM V6.2	VMHCM support for z13
VM64659	z/VM V5.4	VMHCM toleration support for z13 IODF
VM65495	z/VM V6.3 z/VM V6.2	VM EREP support for z13
PM79901	z/VM V6.3 z/VM V6.2	HLASM support for z13

About IBM Privacy Contact Terms of use IBM Feeds Jobs

Tested Linux Platforms

<http://www.ibm.com/systems/z/os/linux/resources/testedplatforms.html>

Distribution	z13	zEnterprise - zBC12 and zEC12	zEnterprise - z114 and z196	System z10 and System z9
RHEL 7	✓ (1,3)	✓ (4)	✓ (4)	✗
RHEL 6	✓ (1,3)	✓ (5)	✓	✓
RHEL 5	✓ (1,3)	✓ (6)	✓	✓
RHEL 4 (*)	✗	✗	✓ (9)	✓
SLES 12	✓ (2,3)	✓	✓	✗
SLES 11	✓ (2,3)	✓ (7)	✓	✓
SLES 10 (*)	✗	✓ (8)	✓	✓
SLES 9 (*)	✗	✗	✓ (10)	✓

z13 Compatibility

z/VM z13 Compatibility

- Guest support for the following new hardware facilities:
 - Load/Store-On-Condition Facility 2
 - Load-and-Zero-Rightmost-Byte Facility
 - Decimal-Floating-Point Packed Conversion Facility
 - PCIe: Extended-I/O Address-Translation Facility guest exploitation
- Integer or binary floating point for capability numbers
 - Accounting / Monitor will now report integer and binary floating point numbers for capability values
 - CP QUERY CAPABILITY will now report decimal numbers for capability values.
- Removal of guest zAAP support
- Toleration of:
 - STP Hardware-Based TOD-Clock Steering
 - SMT feature on machine; stand-alone dump compatibility
 - z/Architecture vector registers; mask from guests
- Dynamic I/O support for HiperSockets VCHID and CS5 Coupling

Simultaneous Multithreading (SMT)

What is Simultaneous Multithreading (SMT)?

- It is the ability of a single physical processor, or **core**, to run more than one stream of instructions at a time
- Each stream of instructions is called a **thread**
- The threads **share** the hardware assets on the core
 - Sometimes they collide or have to take turns...
 - ... but sometimes **they don't**
- When the core cannot make progress on one thread, perhaps it can **keep making progress** on the other one
 - Cache miss is a really good example of this
- This can **increase overall core capacity** to complete instructions even though the individual threads might run:
 - More slowly than they would if they used the core alone
 - More slowly than they did on the previous generation

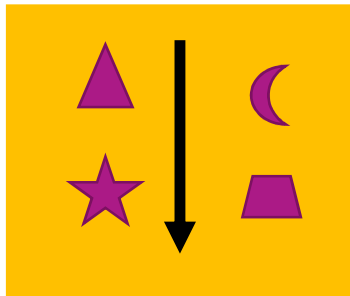


Which approach is designed for the higher volume of traffic? Which road is faster?

** Illustrative numbers only*

Hardware: Cores and Threads

zEC12



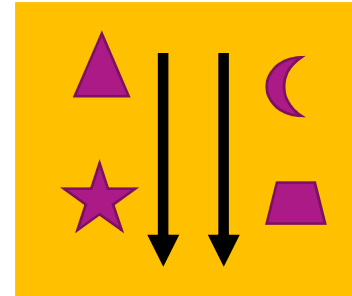
Core: L1, L2, address translator, ...

Thread: PSW, registers, address translations, timers, ... *execution context*

zEC12 had **one** thread per core.

The thread **did not share any** core facilities.

z13



Core: L1, L2, address translator, ...

Thread: PSW, registers, address translations, timers, ... *execution context*

z13 has **two** threads per core for IFLs and zIIPs. The rest have **one**.

The threads **must share some** of the core's facilities.

Two Threads – Example 1

Thread 0

```
L  R3,FIELD A  
L  R6,FIELD C
```

Thread 1

```
LLGC R5,FIELD B  
LGR  R3,R5  
LGHI R0,1  
SLLG R3,R0,0(R3)
```

Let's say:

- FIELD A is in the L3 cache
- FIELD B is in the L1 cache
- FIELD C is in L4 cache

*Note that this is a contrived example, not necessarily representative of the real amount of time these instructions take.

Two Threads – Example 1

<i>Thread 0</i>	<i>Thread 1</i>
Resolving FIELDA	Resolving FIELDB
	LLGC R5,FIELDB
L R3,FIELDA	
Resolving FIELDC	LGR R3,R5
	LGR R3,R5
	LGHI R0,1
	SLLG R3,R0,0(R3)
L R6,FIELDC	

While thread 0 is waiting for its memory references to be resolved, thread 1 can keep running, and so the core keeps making progress.

Because each thread keeps its own registers, their instructions can be interleaved.

Two Threads – Example 2

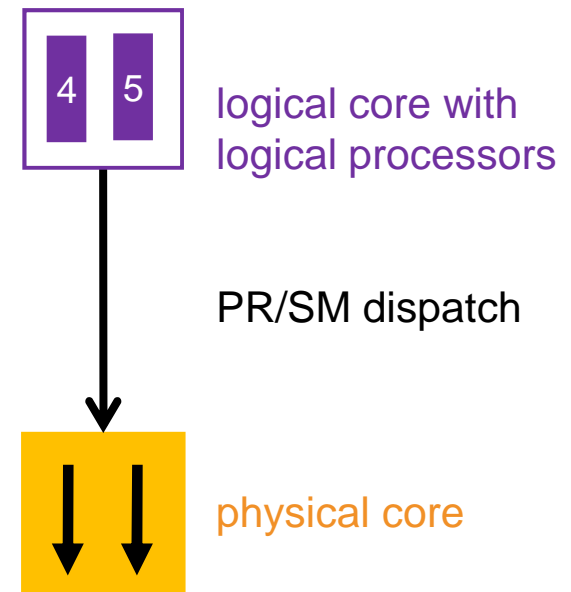
<i>Thread 0</i>	<i>Thread 1</i>
	AR R0,R1
AR R3,R4	
	AR R3,R5
	AR R3,R5
	AR R0,R1
AR R3,R1	
AR R9,R4	
	AR R2,R1

Of course this doesn't work well all the time - what if both threads just have instructions with no memory references?

In this case, each thread will run more slowly than it would if it had its own core.

PR/SM: Logical Processors and Logical Cores

- These are **logical assets** PR/SM provides to partitions
- Logical processor (aka “logical CPU” or sometimes “LPU”)
 - A dispatching context for a stream of instructions
 - When your partition has 14 logical processors...
 - ... it can run 14 instruction streams concurrently
- Logical core
 - A **pair** of logical processors,
 - ...owned by the same partition, and...
 - **always dispatched together on the same physical core**
- Your partition’s activation profile defines **how many logical cores** it has
- PR/SM adjusts the number of logical processors according to **whether the partition enables, or opts in for, SMT**



z13 Vocabulary Cleanup

- Words I try no longer to say
 - CPU *What is this?*
 - IFL, CP, zIIP, ... *as nouns*
 - “The LPAR has three CPs and four IFLs.” *What does this mean?*
 - N-way
 - N cores? N threads? N logical CPUs? *What does this mean?*

- What I now try to say instead
 - Core
 - “This machine contains 36 physical IFL cores.”
 - IFL core, CP core, zIIP core
 - Thread
 - Logical core
 - “In the activation profile for VM1 I defined 14 logical IFL cores.”
 - Logical processor
 - “My partition has 16 logical processors on 8 logical cores.”
 - Logical CPU
 - Same as logical processor
 - N-core

z/VM: SMT

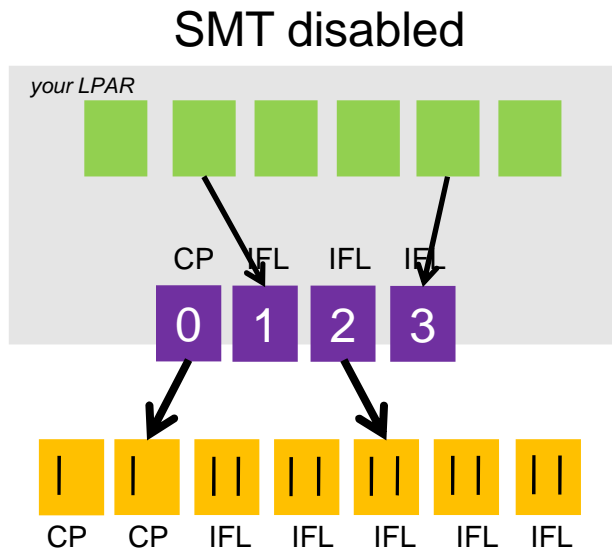
- Objective is to improve system capacity, not the speed of a single instruction stream.
- Lets z/VM dispatch work on up to two threads of a z13 IFL core
 - Up to 32 cores supported
- VM65586 for z/VM 6.3 **only**
 - PTF UM34552 became available March 13, 2015
- Transparent to virtual machine
 - Guest does not need to be SMT-aware
 - SMT is not virtualized to the guest
- z13 bundle 11
- z/VM exploitation is for IFL cores only
- SMT is disabled by default
 - Requires a system configuration setting and re-IPL
 - Once enabled, applies to the entire z/VM system
- Potential to increase the overall capacity of the system
 - Workload-dependent



Which approach is designed for the higher volume of traffic? Which road is faster?

** Illustrative numbers only*

Dispatching: Compare and Contrast



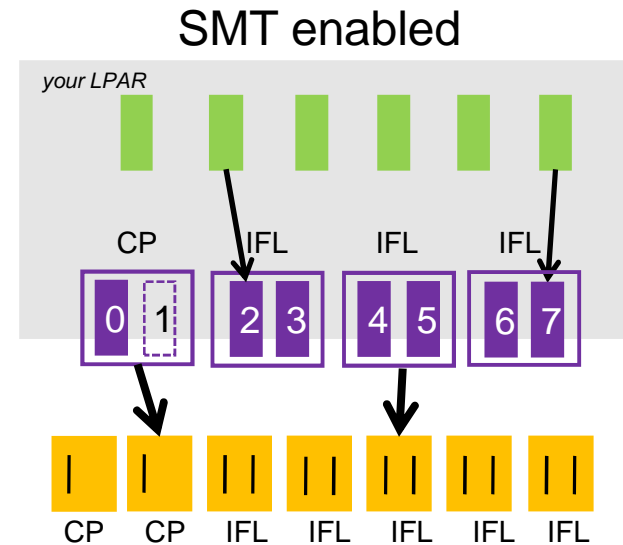
Each physical IFL core can run *two* streams of instructions at a time. We say each one has *two threads*.

When the partition does not *opt in*, PR/SM provides *logical CPUs* for the partition.

PR/SM dispatches *one* logical CPU on a physical core at a time. The other thread goes unused.

z/VM provides *virtual CPUs* for guests.

z/VM dispatches virtual CPUs on logical CPUs.



Each physical IFL core can run *two* streams of instructions at a time. We say each one has *two threads*.

When the partition *opts in*, PR/SM provides logical CPUs for the partition and groups them into *logical cores*.

PR/SM dispatches *one* logical core on a physical core at a time.

z/VM still provides virtual CPUs for guests.

z/VM still dispatches virtual CPUs on logical CPUs.

SMT in Mixed-Engine Environments

- z/VM exploits multithreading on only IFL cores

- A VM-mode partition will have both
 - Logical IFL cores embodying two logical CPUs each
 - Logical non-IFL cores embodying one logical CPU each (CPs, zIIPs)

- Guests in a VM-mode partition must have virtual IFLs defined in order to use IFL cores
 - Virtual IFLs are used only when you **SET VCONFIG MODE LINUX** or **VM**.

 - If you **SET VCONFIG MODE LINUX** you have to choose all virtual IFLs

 - Resetting your VCONFIG settings or redefining the type of CPUs will cause a **SYSTEM RESET** that will kill your guest operating system.

How do I enable SMT on my z/VM system?

- Add the **MULTITHreading ENAbLe** statement to your system configuration file.

- The system must be in *vertical polarization mode* (this is the default)
 - Make sure you **don't** have an **SRM POLARIZATION HORIZONTAL** statement in your system configuration file.

- The system must be using the *reshuffle* dispatcher method (this is the default)
 - Make sure you **don't** have an **SRM DSPWDMMethod REBALANCE** statement in your system configuration file.

- Re-IPL your system!

Enabling SMT – New MULTITHreading statement

- **MULTITHreading** configuration statement lets you specify either
 - Maximum number of threads for all core types
 - Different number of threads for each type
 - z/VM exploits multithreading on only IFL cores

- **CPSYNTAX** has been updated to verify:
 - Are there multiple **MULTITHreading** statements?
 - Is the maximum activated thread value less than the number of threads specified for any type?
 - Is **MULTITHreading ENABLE** specified with any incompatible **SRM** statements?

I believe I enabled SMT, but how do I know it's on?

- New command – **Query MULTITHread (MT)**
- Compares what you requested in the system configuration statement to what was actually activated, given the hardware and software levels.

```
query multithread
Multithreading is enabled.
           Requested      Activated
           Threads        Threads
MAX_THREADS      MAX          2
CP core          2            1
IFL core         2            2
ICF core         2            1
zIIP core        2            1
Ready; T=0.01/0.01 11:51:29
```


QUERY PROCessors with SMT

- Shows which logical core each logical CPU is on:

```
query processors
PROCESSOR 00 MASTER CP      CORE 0000
PROCESSOR 02 ALTERNATE CP   CORE 0001
PROCESSOR 04 ALTERNATE IFL  CORE 0002
PROCESSOR 05 ALTERNATE IFL  CORE 0002
PROCESSOR 06 PARKED IFL    CORE 0003
PROCESSOR 07 PARKED IFL    CORE 0003
PROCESSOR 08 ALTERNATE IFL  CORE 0004
PROCESSOR 09 ALTERNATE IFL  CORE 0004
PROCESSOR 0A ALTERNATE IFL  CORE 0005
PROCESSOR 0B ALTERNATE IFL  CORE 0005
PROCESSOR 0C ALTERNATE IFL  CORE 0006
PROCESSOR 0D ALTERNATE IFL  CORE 0006
PROCESSOR 0E PARKED IFL    CORE 0007
PROCESSOR 0F PARKED IFL    CORE 0007
PROCESSOR 10 ALTERNATE IFL  CORE 0008
PROCESSOR 11 ALTERNATE IFL  CORE 0008
PROCESSOR 12 ALTERNATE IFL  CORE 0009
PROCESSOR 13 ALTERNATE IFL  CORE 0009
PROCESSOR 14 ALTERNATE ZIIP CORE 000A
PROCESSOR 16 ALTERNATE ZIIP CORE 000B
Ready; T=0.01/0.01 11:55:52
```

Vary On and Off

- When SMT is enabled
 - Use **VARY CORE** to vary off or on an entire logical core
 - Multithread or single-thread cores
 - **VARY PROCESSOR** isn't allowed
 - Because one cannot vary a single thread of a core.

- When SMT is not installed or not enabled
 - **VARY CORE** is the same as **VARY PROCESSOR**

```
vary off processor a
HCPCPS1321E VARY PROCESSOR is not valid because multithreading is enabled.
Ready(01321);
vary off core 5
Command accepted
Ready;
Core 0005 offline Proc 000A-000B
vary on core 5
Command accepted
Core 0005 online Proc 000A-000B
Ready;
```

SMT is enabled - how do I see what's going on with my cores?

- **Indicate Load** will still show information by processor, which means by individual thread on multithreaded cores.
 - The percent-busy is thread-busy aka logical CPU-busy
- A new command, **INDicate MULTITHread (MT)** will show you the per-type information, giving you an idea of how much capacity you have left for each type. The utilization shown is an average of the utilization of the cores of that type.

```
indicate multith
Multithreading is enabled.
Statistics from the interval 12:00:53 - 12:01:23
Core Type CP      Busy    8%    TD  1.00 of 1    Prod 100%    Util    8%
   CF  100%    MaxCF  100%
Core Type IFL    Busy    1%    TD  1.50 of 2    Prod  90%    Util    1%
   CF  113%    MaxCF  125%
Core Type ZIIP   Busy    0%    TD  1.00 of 1    Prod 100%    Util    0%
   CF  100%    MaxCF  100%
Ready;
```

What are all those other numbers on Indicate MT?

- Busy time – percent of time at least one thread of the core was busy (aka *core utilization*)
- Thread density - how often the core was able to run both threads at once, while the core was in use at all
- Productivity – work completed while core non-idle, compared to work that could have been completed if all non-idle time were two-threads-busy time
- Utilization (MT utilization) - how much of the maximum core capacity was used
- Capacity factor – total work rate for the core while busy, compared to its work rate when it was running with one-thread-busy (the “*SMT benefit*”)
- Maximum capacity factor – work rate for two-threads-busy, compared to the work rate for one-thread-busy

```

indicate multith
Multithreading is enabled.
Statistics from the interval 12:00:53 - 12:01:23
Core Type CP      Busy   8%   TD  1.00 of  1   Prod 100%   Util   8%
   CF  100%   MaxCF  100%
Core Type IFL     Busy   1%   TD  1.50 of  2   Prod  90%   Util   1%
   CF  113%   MaxCF  125%
Core Type ZIIP    Busy   0%   TD  1.00 of  1   Prod 100%   Util   0%
   CF  100%   MaxCF  100%
Ready;

```

Processor Time Reporting

- **Raw time** (the old way, but with new implications)
 - Amount of time each virtual CPU is run on a thread
 - This is the only kind of time measurement available when SMT is disabled
 - Used to compute dispatcher time slice and scheduler priority

- **MT-1 equivalent time** (new)
 - Used when SMT is enabled
 - Approximates what the raw time would have been if the virtual CPU had run on the core all by itself
 - Adjusted downward (decreased) from raw time
 - Intended to be used for chargeback

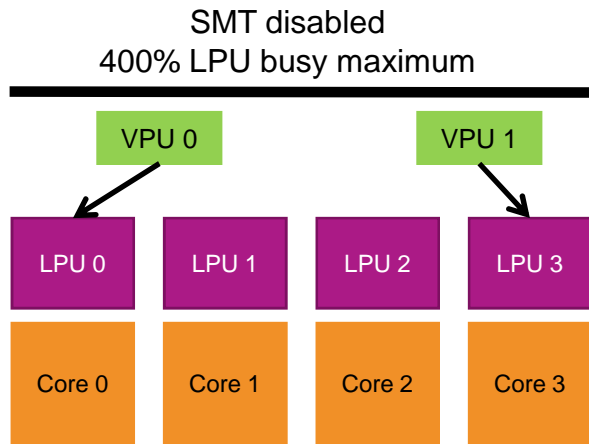
Processor Time Reporting

	Raw Time	MT-1 Equivalent time
INDICATE USER		X
QUERY TIME		X
LOGOFF		X
TYPE 1 Accounting record		X
TYPE F Accounting record	X	
Diag x'0c'	X	
Diag x'70'	X	
Diag x'270'	X	
Diag x'2FC'	X	
Monitor Records	X	X

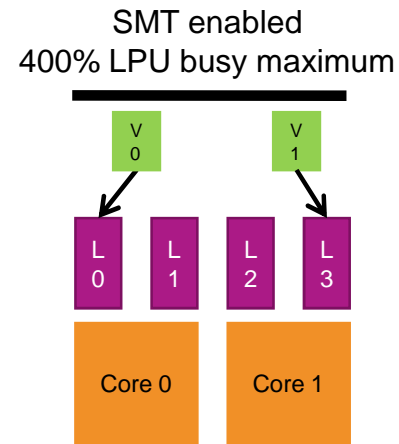
Note: "CONNECT" time displayed by commands represents wall-clock time and is not changed

CPU Pooling and Simultaneous Multithreading

CPU Pooling always regulates *total virtual CPU utilization of the pool's guests*.



more work per virtual CPU-second

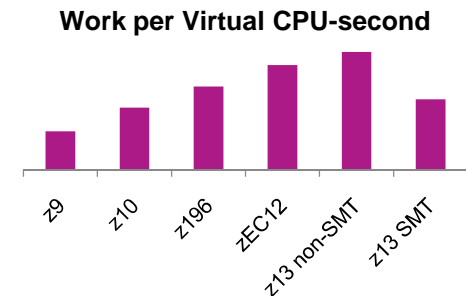


less work per virtual CPU-second

DEFINE CPUPOOL ... (CAPACITY nnn
DEFINE CPUPOOL ... (LIMITHARD nnn%

cap expressed in absolute consumption
 cap expressed in percent of LPU capacity

You might have to adjust your limits to account for how much work a virtual CPU can accomplish in a CPU-second.



Prorated Core Time (availability TBD)

- Prorated core time will divide core dispatch time evenly among the threads dispatched in that interval
 - CPU pool capacity consumed as if by cores
 - Suitable for core-based software licensing

- When SMT is enabled, prorated core time will be calculated for users who are
 - In a CPU pool limited by the **CAPACITY** option
 - Limited by the **SET SHARE LIMITHARD** command
(currently raw time is used; raw time will continue to be used when SMT is disabled)

- **QUERY CPUPOOL** will show capacity in cores instead of CPUs

- Prorated core time will be reported in monitor records and the new Type F accounting record.

- Watch for APAR VM65680

SMT Performance Summary

- Performance varies from workload to workload
- Workload throughput and response time are the best ways to determine whether SMT is providing value
- Throughput for measured workloads varied from 0.64x to 1.34x
- BEST results were observed in applications that consist of highly parallel activity
- WORST results were observed in applications that consist of highly serial activity
- It might be necessary to adjust the workload or configuration to overcome serial behavior
 - For example, adding virtual CPUs to a guest
- See updated z/VM Performance Report
 - <http://www.vm.ibm.com/perf/reports/zvm/html/>

Increased CPU Scalability

Increased CPU Scalability

- Various improvements to help z/VM to run more efficiently when large numbers of processors are present, thereby improving the N-way curve

- APAR VM65586 for z/VM 6.3 **only**
 - PTF UM34552 available March 12, 2015

- For z13
 - With SMT disabled, increases logical processors supported from 32 to 64
 - With SMT enabled, the limit is 32 logical cores (yields at most 64 logical processors)

- For machines prior to z13
 - Limit remains at 32 logical processors
 - Might still benefit from improved N-way curves

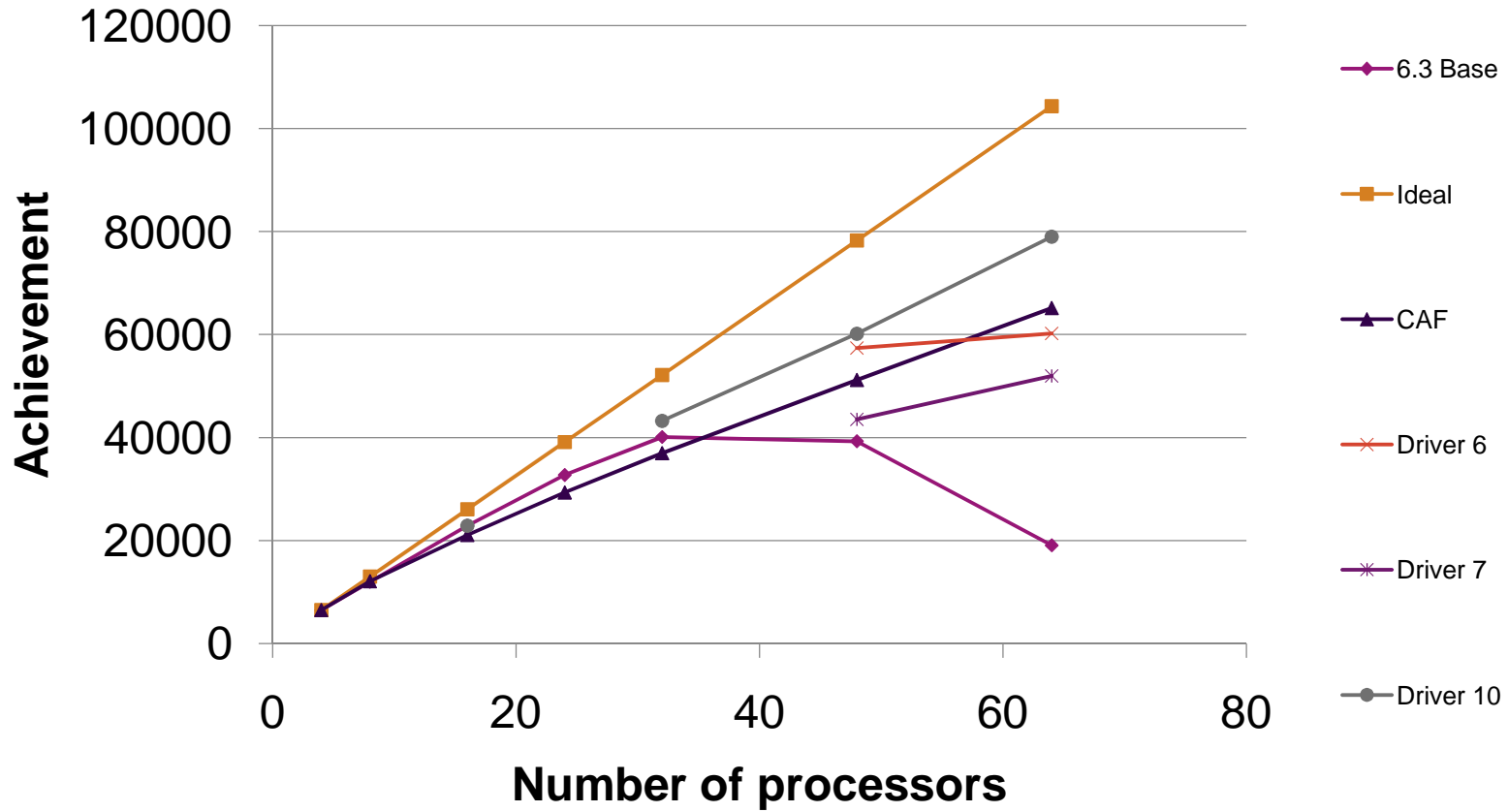


Areas Improved to Increase CPU Scalability

- Improvements were made to the following areas to improve efficiency and reduce contention:
 - Scheduler lock
 - VSWITCH data transfer buffers
 - Serialization and processing of VDISK I/Os
 - Memory management
- Some areas needing improvement were known – others required thorough investigation and experimentation
- All tested workloads showed acceptable scaling up to...
 - ... 64 logical processors when SMT is enabled
 - ... 32 logical processors when SMT is not enabled
- Benefits are workload-dependent

Creating a Scalability Enhancement

Scalability of Test Workload



CPU Scalability: Scheduler Lock

- Reduced contention for scheduler lock
 - Known to cause overhead for many workloads

- Eliminated unnecessary exclusive holds of scheduler lock
 - Only a share of the lock needed
 - Lock not needed at all

- Improved efficiency of processes that use the scheduler lock
 - Work stacking improvements
 - Test-idle changes

CPU Scalability: VSWITCH Data Transfer Buffers

- Frequently obtained in systems with high VSWITCH activity
 - Caused spin lock overhead

- Added processor-local queues to existing global queues

- Modified structure of global queues

- Improved efficiency
 - Dequeueing batches of buffers
 - Moving batches of buffers from global queues to local queues

- Reduced lock hold time

CPU Scalability: VDISK I/O

- Reduced attaches and detaches of VDISK address spaces
 - Previously attached and detached after each CCW was processed
 - Now done only once per channel program

- Reduced serialization contention when updating CP-use access list (used to read and write VDISK data)
 - Removed lock usage for some list processing
 - Reduced likelihood of concurrent tasks using the same section of the list
 - Starting index of search for available list entries is derived from processor address

CPU Scalability: Memory Management

- Improved management of frame lists
 - Local frame available list
 - Processed frame list

- Improved page table serialization
 - Mainly affects memory allocation and deallocation

- Most benefit seen during system initialization and when memory is very constrained

CPU Scalability: Processor Parking

- Base z/VM 6.3
 - Control Program used T/V ratio as an indicator of MP overhead
 - Used in processor parking decision algorithm

- z/VM 6.3 with CPU scalability enhancements
 - Control Program overhead due to MP is reduced
 - T/V ratio is a less accurate indicator of MP overhead
 - Removed from parking decision

- Control Program no longer parks entitled engines (vertical highs or mediums)
 - Base z/VM 6.3 would have parked entitled engines only when T/V was high

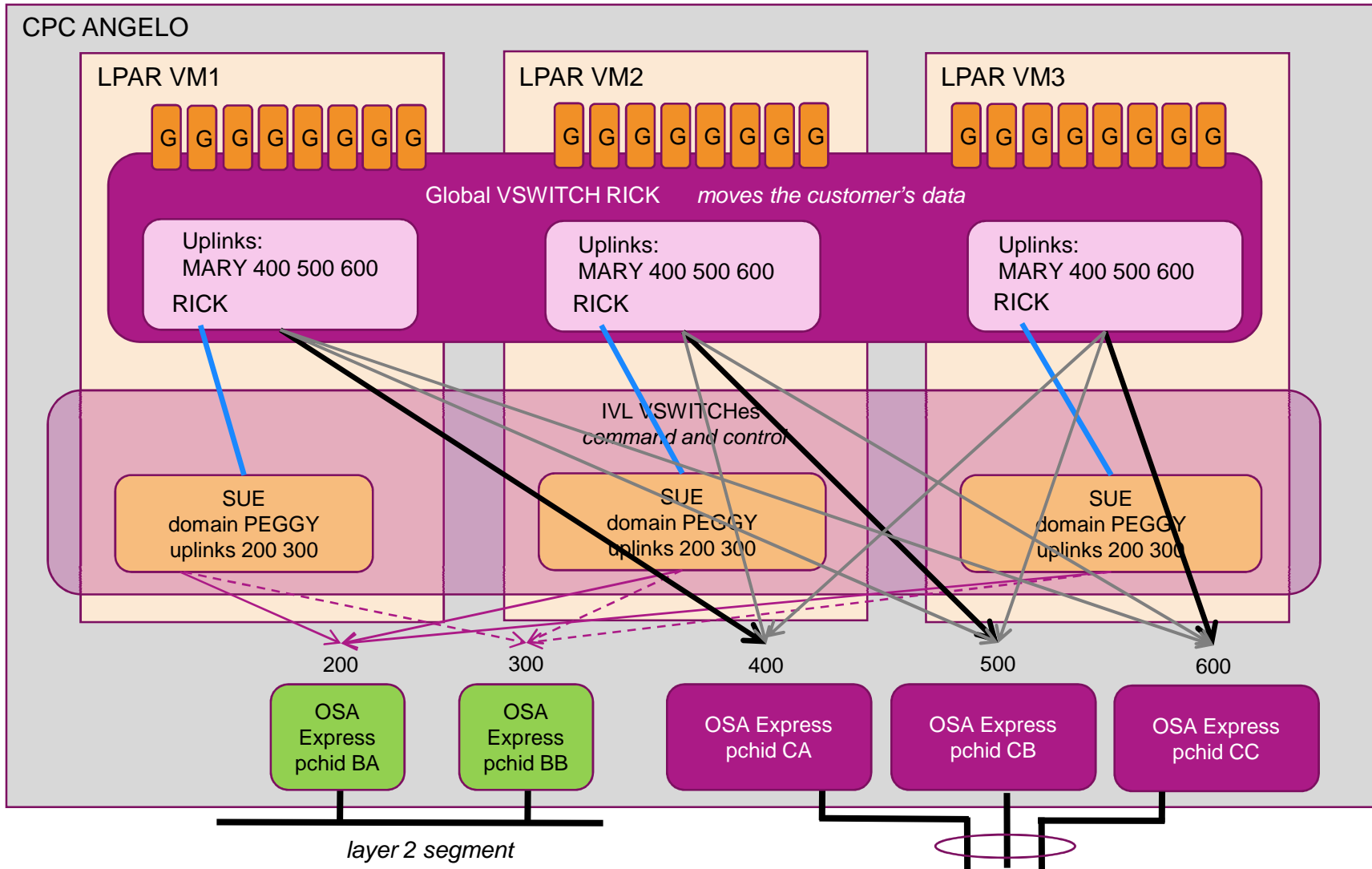
Multi-VSWITCH Link Aggregation

Overview

- Makes it possible to do link aggregation with VSWITCHes without the requirement for dedicated OSAs
- Lets a port group of OSA-Express features span VSWITCHes within a single or multiple z/VM systems in same CPC
 - Cannot be shared with non-z/VM logical partitions or z/VM systems without the enhancement
- Available on only z13
 - Requires OSA enhancements introduced with the z13
- Allows better consolidation and availability while improving TCO
- APARs VM65583 and PI21053 for z/VM 6.3 **only**
 - PTFs planned to be available June 26, 2015



Multi-VSWITCH LAG Configuration



MV-LAG: Command Highlights

- On each system, [create its IVL VSWITCH](#)

```
DEFINE VSWITCH SUE TYPE IVL DOMAIN PEGGY UPLINK RDEV 200 300
```

- On each system, [declare the shared port group](#)

```
SET PORT GROUP MARY LACP ACTIVE SHARED
```

```
SET PORT GROUP MARY JOIN 400 500 600
```

- On each system, [create the global VSWITCH member](#)

```
DEFINE VSWITCH RICK TYPE QDIO GLOBAL ETHERNET UPLINK GROUP MARY
```

New Attributes

- **Inter-VSWITCH Link (IVL) network**
 - Provides management and data communication among multiple z/VM systems

- **IVL virtual switch**
 - One per z/VM system
 - Communicates with other systems in the same IVL domain
 - Carries all inter-LPAR control traffic among systems

- **IVL domain**
 - Group of up to 16 z/VM systems
 - Each system has an IVL virtual switch defined
 - All systems' IVL VSWITCHes' UPLINK ports are connected to the same external LAN segment
 - Each IVL domain is assigned a separate, reserved multicast MAC address
 - A z/VM system can belong to only one IVL domain
 - The domain name is set when the IVL VSWITCH is created

```
DEFINE VSWITCH SUE TYPE IVL DOMAIN PEGGY UPLINK RDEV 200 300
```

New Attributes...

- **Shared port group**

- Set of OSA-Express features (chpids) configured to be shared among members of an IVL domain
- Can be used with only this new style of VSWITCH

SET PORT GROUP MARY LACP ACTIVE SHARED

SET PORT GROUP MARY JOIN 400 500 600

New Attributes...

▪ Global VSWITCH

- Collection of VSWITCHes that share the same networking characteristics
 - Same name and attributes
 - Reside on systems that all belong to the same IVL domain

- To create a global VSWITCH "member"
 - The system's IVL virtual switch must be defined and its UPLINK port connected
 - Form the DEFINE VSWITCH command to match name and attributes

- Created when first member is defined, deleted when last member is detached

```
DEFINE VSWITCH RICK TYPE QDIO GLOBAL ETHERNET UPLINK GROUP MARY
```

Crypto Enhancements

z Systems Crypto Enhancements

- z Systems *crypto architecture* itself grew
 - Maximum number of crypto features increased from 64 to 256
 - Maximum number of domains per AP increased from 16 to 256
 - The new architecture applies to Crypto Express4S and Crypto Express5S on z13

- z13 uses the new architecture to support more domains
 - Up to 16 APs and up to 85 domains per AP

- New Crypto Express5S card
 - Increased performance over Crypto Express4S
 - New functions: VISA Formatted Preserving Encryption, Elliptic Curve Cryptography

- Read more about it:
 - [Ultimate Security with the IBM z13](#), an IBM Redbooks publication

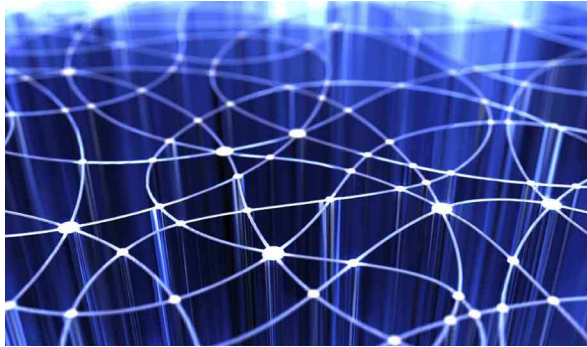
z/VM Crypto Enhancements

- z/VM can handle the expanded z Systems architecture
- z/VM supports the new Crypto Express5S card
 - When configured as: (same as Express4S, by the way)

Config mode	Shared?	Dedicated?
IBM Common Cryptographic Architecture (CCA) coprocessor	yes	yes
IBM Enterprise PKCS #11 (EP11) coprocessor		yes
Accelerator	yes	yes

- Only z/Architecture guests can use the new Crypto Express5S card
- Authorized via **CRYPTO** statement in guest's CP directory entry
- New z/VM system configuration statement lets the administrator specify which domains should be used to fulfill shared crypto
 - Stops all the nondedicated domains from defaulting into shared mode
 - If no statement specified, only two domains get reserved for shared
 - **CRYPTO APVIRTUAL AP n1-n2 DOMAIN p1-p2**

Summary



Leadership

z/VM continues to provide additional value to the platform as the strategic virtualization solution for z Systems. Virtual Switch technology in z/VM is industry-leading.



Innovation

z/VM 6.3 added HiperDispatch, allowing greater efficiencies to be realized. Now adding SMT with topology awareness raises the bar again.



Growth

z/VM 6.3 increases the vertical scalability and efficiency to complement the horizontal scaling introduced in z/VM 6.2, because we know our customers' systems continue to grow. This year we continue to extend the limits with processor scalability improvements.

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/>
 - <http://www.vm.ibm.com/service/vmreqz13.html>

- 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/>

Thank You

Brian Wade
bkw@us.ibm.com

Continue growing your IBM skills



ibm.com/training provides a comprehensive portfolio of skills and career accelerators that are designed to meet all your training needs.

- **Training in cities local to you - where and when you need it, and in the format you want**
 - Use [IBM Training Search](#) to locate public training classes near to you with our five Global Training Providers
 - Private training is also available with our Global Training Providers
- **Demanding a high standard of quality – view the paths to success**
 - Browse [Training Paths](#) and [Certifications](#) to find the course that is right for you
- **If you can't find the training that is right for you with our Global Training Providers, we can help.**
 - Contact IBM Training at dpmc@us.ibm.com



Global Skills Initiative