SHARE in Orlando 2015

SHARE Educate · Network · Influence

z/VM and the IBM z13

Session 17526

John Franciscovich

IBM: z/VM Development

Endicott, NY

#SHAREorg

CELEBRATING 60 YEARS OF SHARE Influencing IT Since 1955

SHARE is an independent volunteer-run information technology association that provides **education**, professional **networking** and industry **influence.**

Trademarks

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

BladeCenter*
DB2*
DS6000*
DS8000*
ECKD
FICON*
GDPS*
HiperSockets
HyperSwap
IBM z13*
OMEGAMON*
Performance Toolkit for VM
Power*
PowerVM
PR/SM
RACF*
Storwize*
System Storage*
System x*
System z*
System z9*
System z10*
Tivoli*
zEnterprise*
z/OS*
zSecure
z/VM*
z Systems*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

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.
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.
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.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
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
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.
TEALEAF is a registered trademark of Tealeaf, an IBM Company.
Windows Server and the Windows logo are trademarks of the Microsoft group of countries.
Worklight is a trademark or registered trademark of Worklight, an IBM Company.
UNIX is a registered trademark of The Open Group in the United States and other countries.

* 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.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads 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). 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 SEs 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.

Acknowledgements

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

- … and anyone else who contributed to this presentation that 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

z/VM Support for the IBM z13

Expanding the Horizon of Virtualization

- Release for Announcement – 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 only on z/VM 6.3
  - IBM z13 Simultaneous Multithreading
  - Increased processor scalability

z/VM Support for IBM z13

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

- **No z/VM 5.4 support**

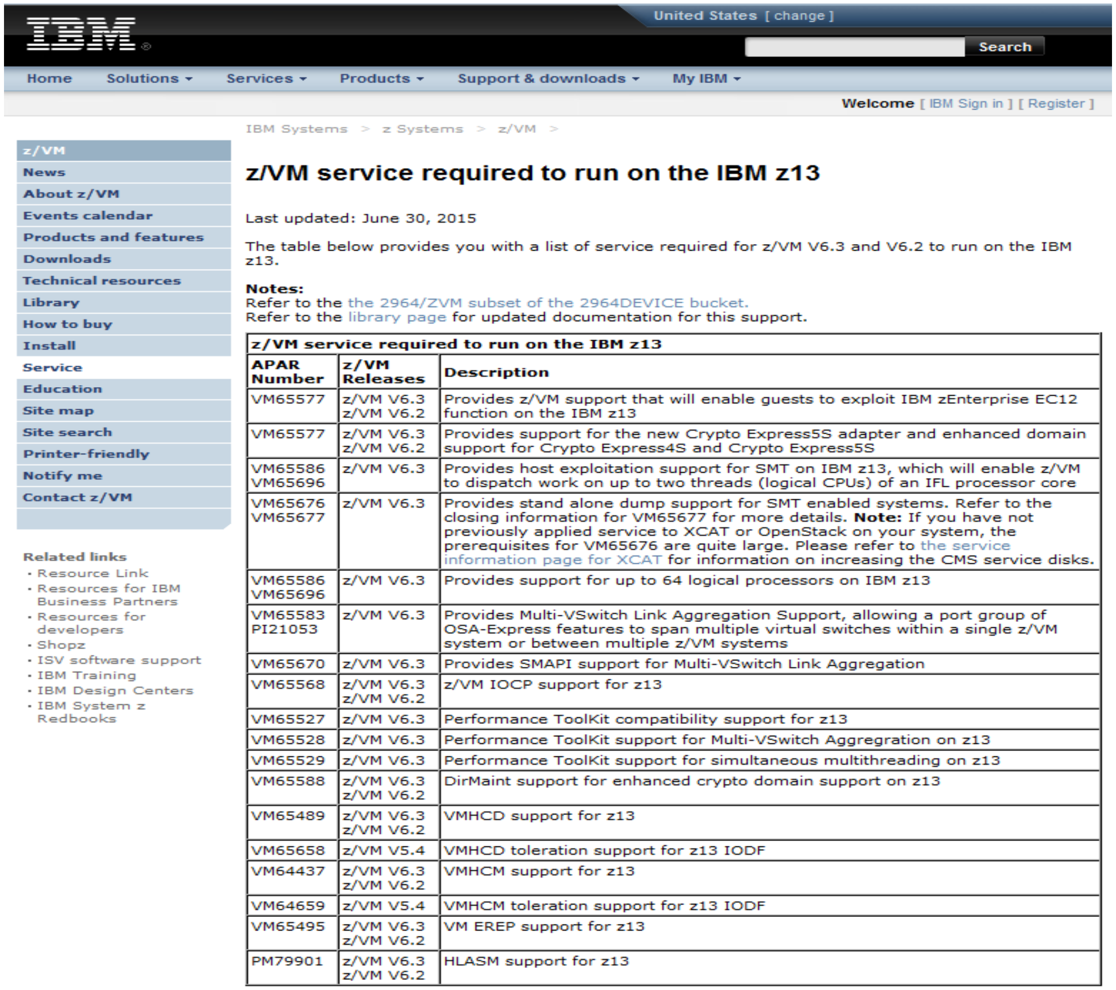
- **No z/VM 6.1 support even if you have extended support contract.**

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

- **If running Linux, check for required updates prior to migration!!**

z/VM Service Required for the IBM z13

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



z/VM service required to run on the IBM z13

Last updated: June 30, 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.

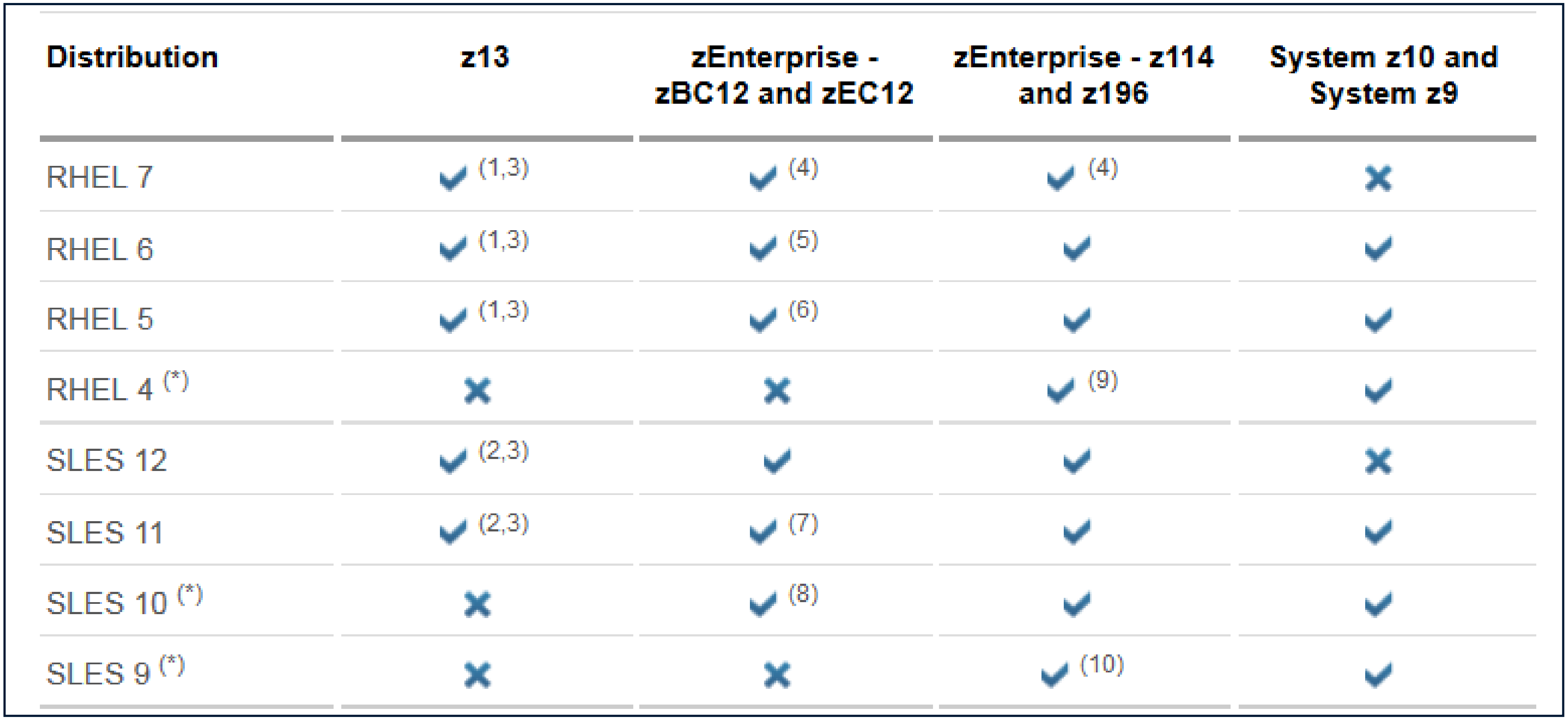
Notes:
Refer to the the 2964/ZVM subset of the 2964DEVICE bucket.
Refer to the library page for updated documentation for this support.

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 VM65696	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 stand alone dump support for SMT enabled systems. Refer to the closing information for VM65677 for more details. **Note:** If you have not previously applied service to XCAT or OpenStack on your system, the prerequisites for VM65676 are quite large. Please refer to the service information page for XCAT for information on increasing the CMS service disks.
VM65586 VM65696	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	Performance ToolKit compatibility support for z13
VM65528	z/VM V6.3	Performance ToolKit support for Multi-VSwitch Aggregration on z13
VM65529	z/VM V6.3	Performance ToolKit support for simultaneous multithreading 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

Tested Linux Platforms

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

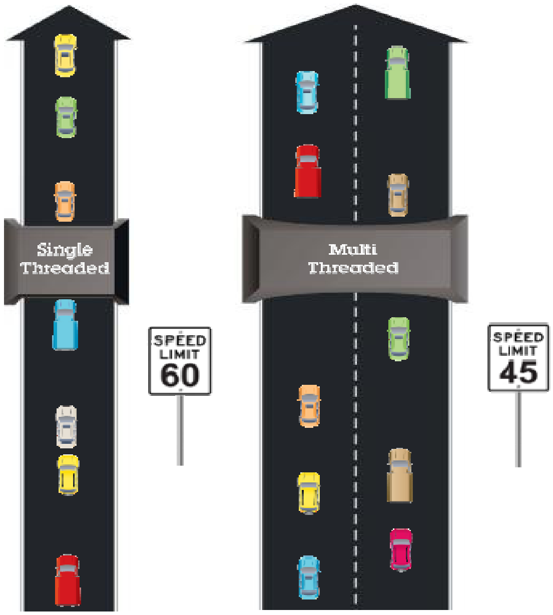


Distribution	z13	zEnterprise - zBC12 and zEC12	zEnterprise - z114 and z196	System z10 and System z9
RHEL 7	✓ (1,3)	✓ (4)	✓ (4)	✗
RHEL 6	✓ (1,3)	✓ (5)	✓	✓
RHEL 5	✓ (1,3)	✓ (6)	✓	✓
RHEL 4 (*)	✗	✗	✓ (9)	✓
SLES 12	✓ (2,3)	✓	✓	✗
SLES 11	✓ (2,3)	✓ (7)	✓	✓
SLES 10 (*)	✗	✓ (8)	✓	✓
SLES 9 (*)	✗	✗	✓ (10)	✓

Simultaneous multithreading (SMT)

What is Simultaneous Multithreading (SMT)?

- 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 slower.
  - Amount of benefit for different workloads **will** vary

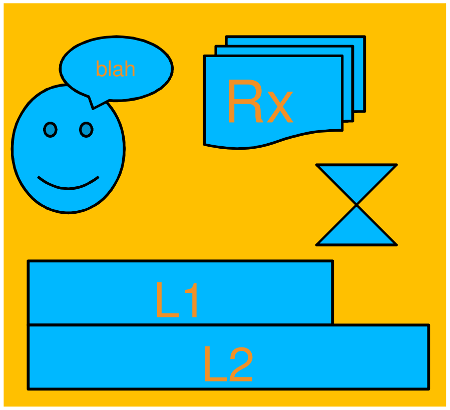


Which approach is designed for the higher volume of traffic? Which road is faster?

**Illustrative numbers only*

Cores and Threads

Single threaded cores



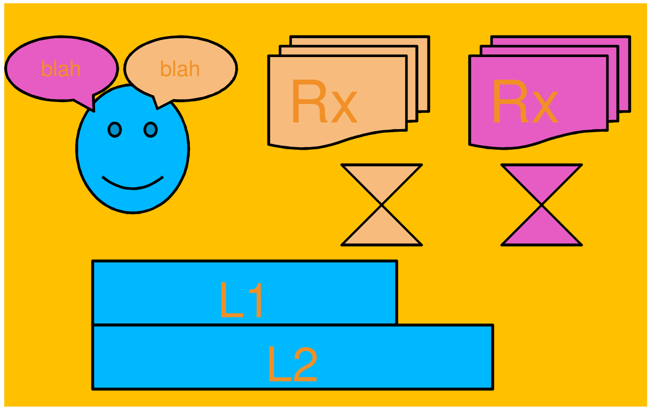
Core: L1, L2, address translator, …

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

zEC12 had **one** thread per core.

What's mine is mine, no sharing!

Multithreaded cores



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 core facilities!

How would this look, in a perfect world?

Thread 0

```
L     R3,FIELDA
L     R6,FIELDC
```

Thread 1

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

Let's say:

FIELDA is in the L3 cache
FIELDB is in the L1 cache
FIELDC is in L4 cache

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

A happy marriage

Thread 0	*Thread 1*
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 has its own registers, the threads can run absolutely concurrently.

A fight for shared resources

Thread 0	*Thread 1*
	AR R0,R1
AR R3,R4	
	AR R3,R5
	AR R3,R5
	AR R0,R1
AR R3,R1	
AR R9,R4	
	AR R2,R1

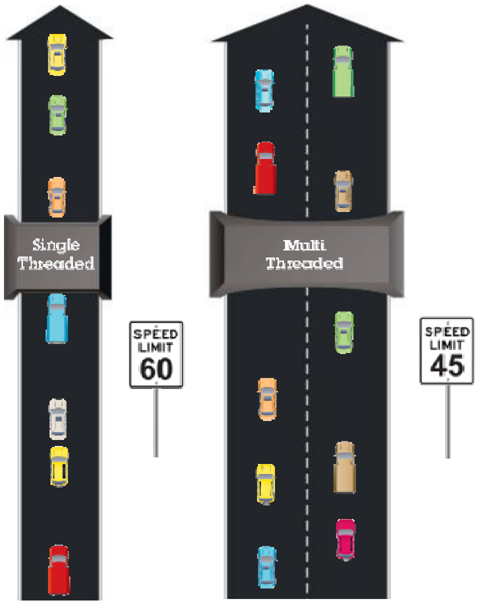
Of course this doesn't work well all the time - what if both threads just have instructions with no memory references?

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

Simultaneous multithreading (SMT) and z/VM

SMT on z/VM

- Objective is to improve capacity, not the speed of a single instruction stream
- z/VM can now 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
- Requires z13 millicode bundle 11
- Transparent to virtual machines
  - Guests do not need to be SMT-aware
  - SMT is not virtualized to the guest
- z13 exploitation is for IFL cores only
- SMT is disabled by default
  - Requires a system configuration setting and re-IPL
  - When enabled, applies to the entire 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*

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

1. Install your IBM z13 mainframe
2. Install service for APAR **VM65586**
3. Set up an LPAR with at least some IFL engines
   - Could be a Linux-only LPAR with all IFLs
   - Could be a VM-mode LPAR with some IFLs
4. 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 CONFIG.
5. The system must be using the *reshuffle dispatcher method* (this is the default)
   Make sure you ***don't*** have an **SRM DSPWDMethod REBALANCE** statement in your SYSTEM CONFIG.
6. Add the **MULTITHreading ENAble** statement to your SYSTEM CONFIG
7. Re-IPL your system!

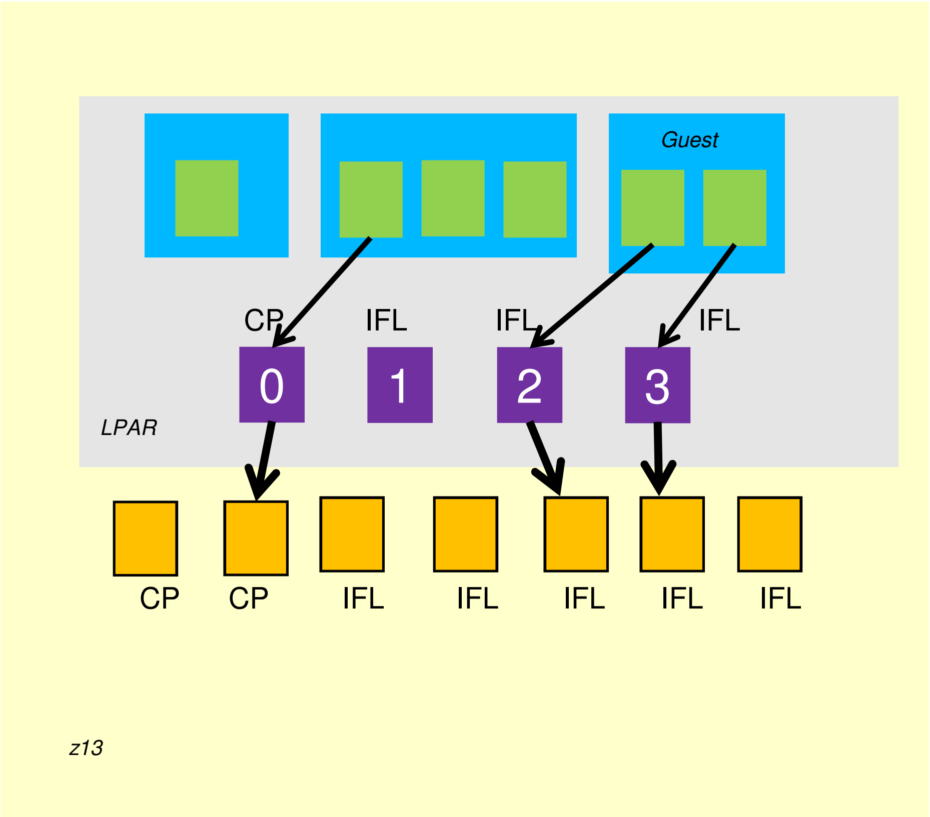
Enabling SMT – MULTIThreading statement

- **MULTIThreading** configuration statement allows you to specify either
  - maximum number of threads for all core types
  - different number of threads for each type
    - z/VM supports 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 enabled SMT; what does that mean for guests?

SMT disabled



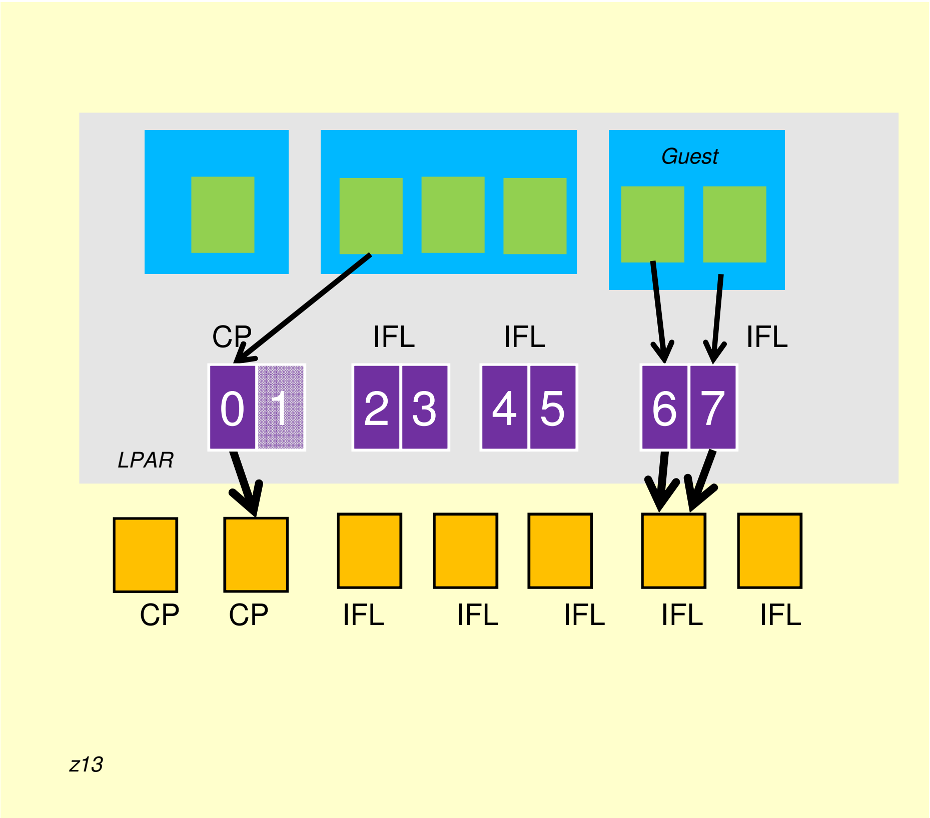
z/VM provides *virtual CPUs* for guests.
z/VM dispatches virtual CPUs on logical CPUs.

When the partition does not *opt in to SMT*, PR/SM provides *logical CPUs* for the partition.
PR/SM dispatches *one* logical CPU on a physical core at a time.

Each physical IFL core can run *two* streams of instructions at a time.
We say each one has two *threads.*
In this case for IFL cores, one thread goes unused.

I enabled SMT; what does that mean for guests?

SMT enabled



z/VM still provides virtual CPUs for guests.

z/VM still dispatches virtual CPUs on logical CPUs.

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

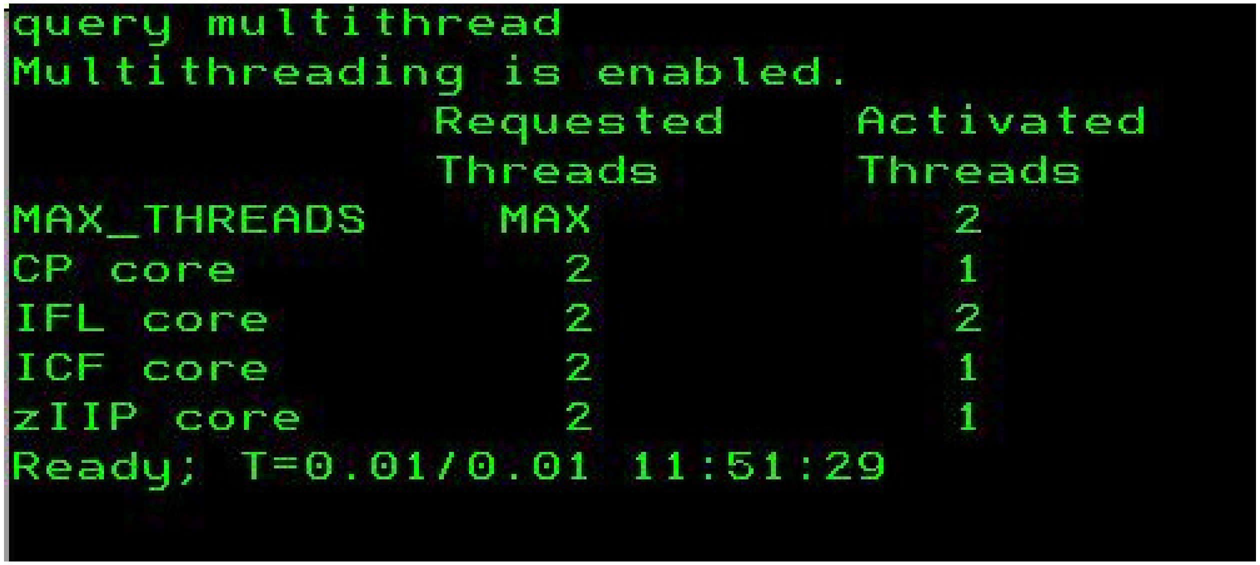
PR/SM dispatches *one* logical core on a physical core at a time.

Each physical IFL core can run *two* streams of instructions at a time. We say each one has two *threads*. In this case for IFL cores, both threads are used.

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

- New command – **Query MULTIThread (Query MT)**

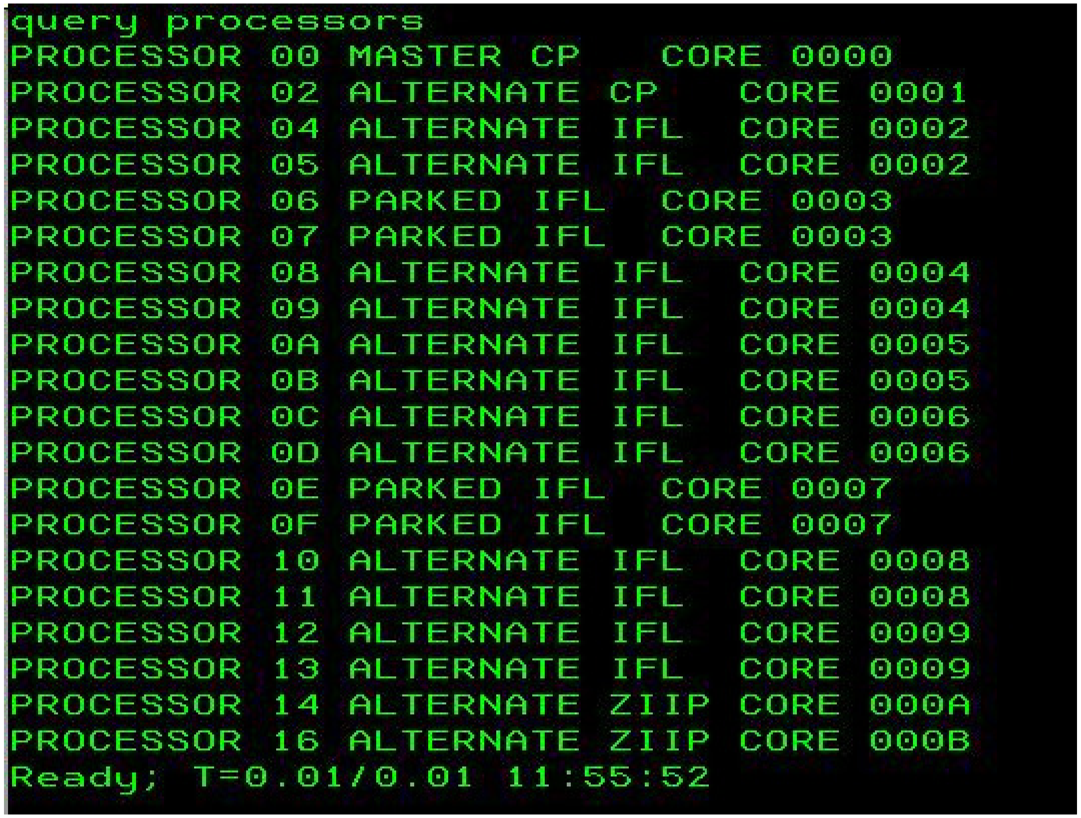
- Compares what you requested in the system configuration statement to what was actually available to be activated, given the hardware and software levels.



```
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:

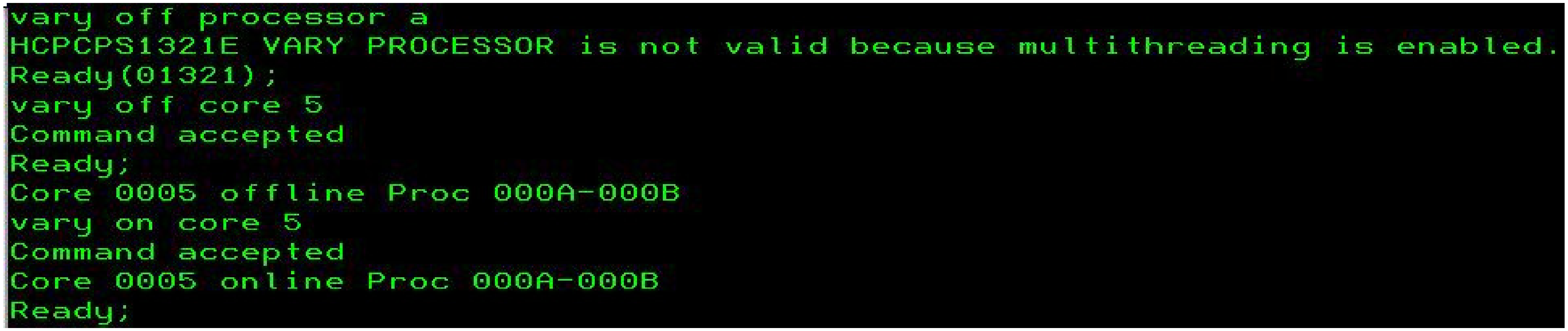


```
query processors
PROCESSOR 00 MASTER CP    CORE 0000
PROCESSOR 02 ALTERNATE CP    CORE 0001
PROCESSOR 04 ALTERNATE IFL   CORE 0002
PROCESSOR 05 ALTERNATE IFL   CORE 0002
PROCESSOR 06 PARKED IFL   CORE 0003
PROCESSOR 07 PARKED IFL   CORE 0003
PROCESSOR 08 ALTERNATE IFL   CORE 0004
PROCESSOR 09 ALTERNATE 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 core
    - Multithread or single thread cores
  - **VARY PROCESSOR** isn't allowed
    - Cannot vary a single thread of a core.

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

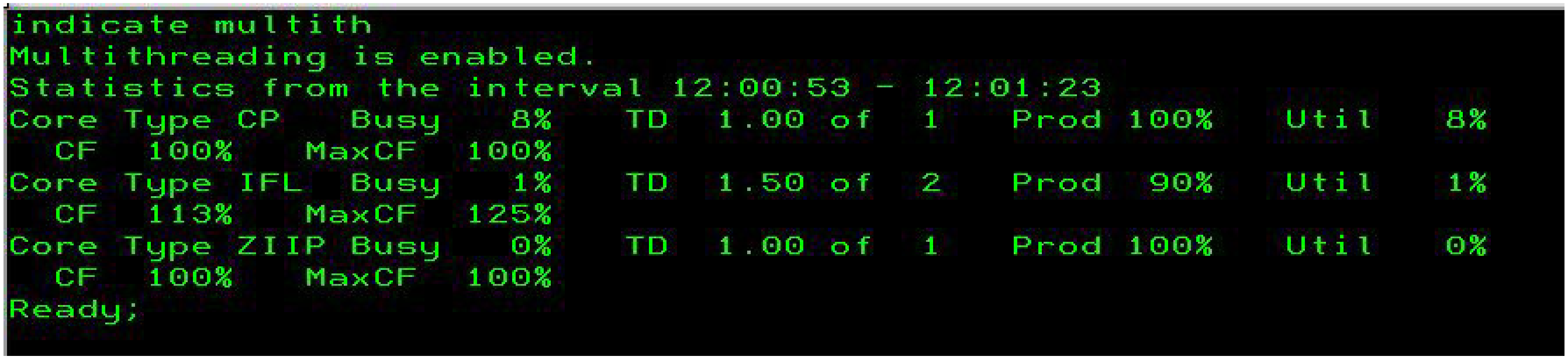


```
vary off processor a
HCPCPS1321E VARY PROCESSOR is not valid because multithreading is enabled.
Ready(01321);
vary off core 5
Command accepted
Ready;
Core 0005 offline Proc 000A-000B
vary on core 5
Command accepted
Core 0005 online Proc 000A-000B
Ready;
```

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

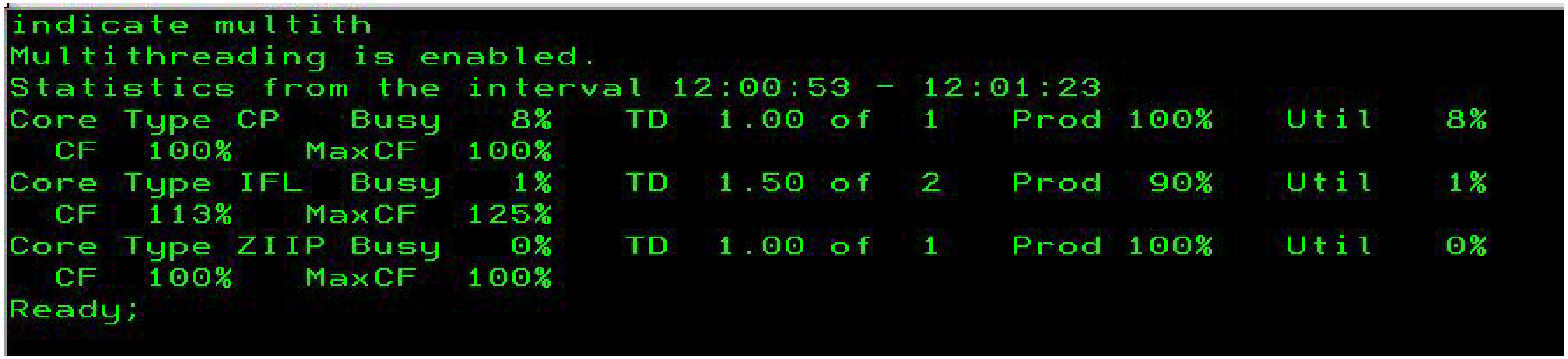
- A new command, **INDicate MULTITHread (MT)** will show you the per type information, giving you an idea of how much capacity you have left for each type. The utilization shown is an average of the utilization of the cores of that type.



```
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

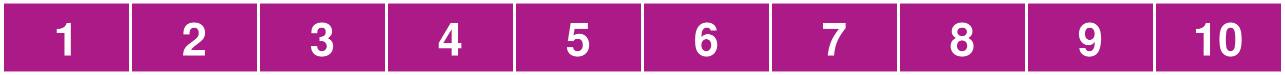


```
indicate multith
Multithreading is enabled.
Statistics from the interval 12:00:53 - 12:01:23
Core Type CP   Busy   8%   TD  1.00 of  1   Prod 100%   Util   8%
  CF   100%   MaxCF  100%
Core Type IFL  Busy   1%   TD  1.50 of  2   Prod  90%   Util   1%
  CF   113%   MaxCF  125%
Core Type ZIIP Busy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

Does multithreading affect the space-time continuum in any way?

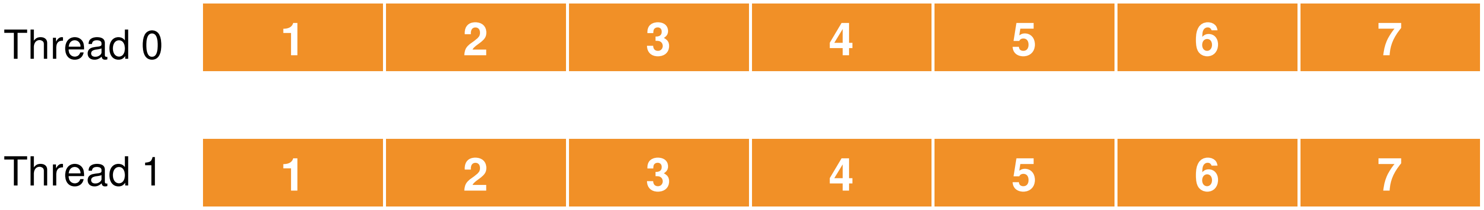
Additional Work Capacity

IFL (SMT disabled) – Instruction Execution Rate: 10



1 2 3 4 5 6 7 8 9 10

IFL (SMT enabled) – Instruction Execution Rate: 7

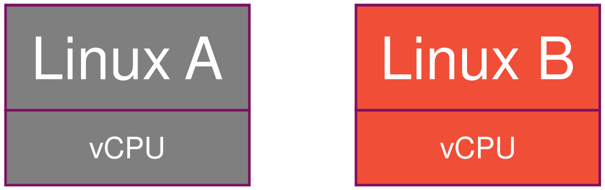


Thread 0 1 2 3 4 5 6 7

Thread 1 1 2 3 4 5 6 7

- Numbers are just for illustrative purposes
- Without SMT, 10 / second
- With SMT, 7 / second but two threads yields capacity of 14 / second

Interleaving Virtual CPUs of Guests



Linux A

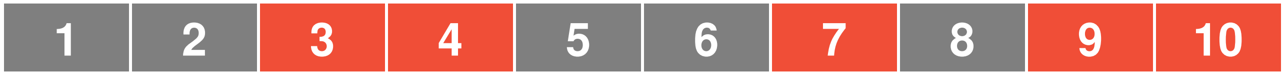
vCPU

Linux B

vCPU

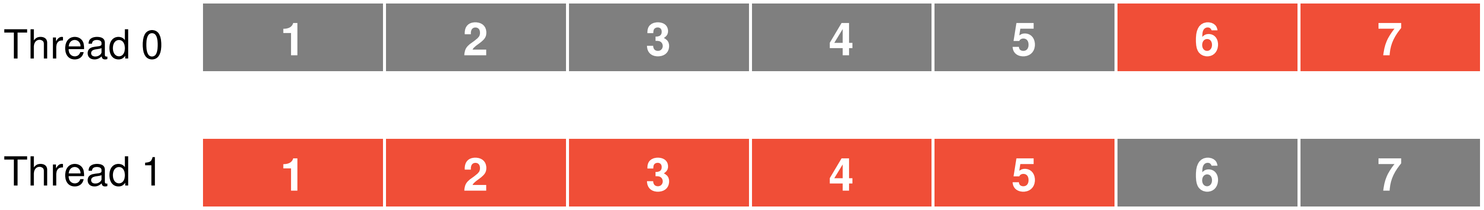
- In single core, we time slice access with each guest getting 5 ops completed.
- With SMT, each guest gets 7 ops completed for total of 14

IFL (SMT disabled) – Instruction Execution Rate: 10



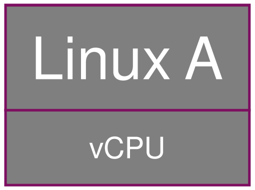
1	2	3	4	5	6	7	8	9	10

IFL (SMT enabled) – Instruction Execution Rate: 7



Thread 0	1	2	3	4	5	6	7
Thread 1	1	2	3	4	5	6	7

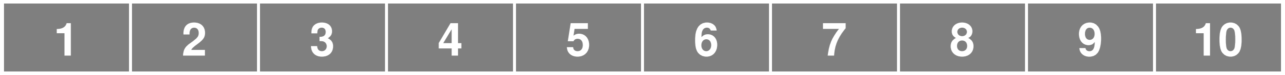
Potential Need to Increase Virtual CPUs


Linux A

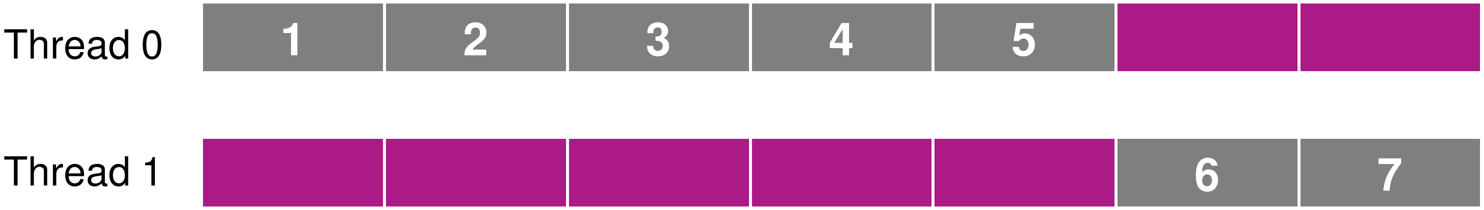
vCPU

- Lets look at a single guest that hits maximum of its virtual resources
- In single core, it can execute 10 ops, but only 7 with SMT as there is only one virtual CPU to dispatch.

IFL (SMT disabled) – Instruction Execution Rate: 10

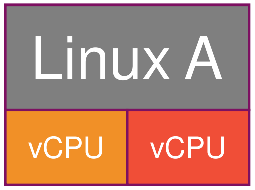

1 2 3 4 5 6 7 8 9 10

IFL (SMT enabled) – Instruction Execution Rate: 7


Thread 0 1 2 3 4 5

Thread 1 6 7

Potential Need to Increase Virtual CPUs



- Taking that guest and giving it a second virtual CPU allows additional work to be completed
  (if guest can exploit multiple virtual CPUs)

IFL (SMT disabled) – Instruction Execution Rate: 10

1	2	3	4	5	6	7	8	9	10

IFL (SMT enabled) – Instruction Execution Rate: 7

Thread 0	1	2	3	4	5	6	7
Thread 1	1	2	3	4	5	6	7

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

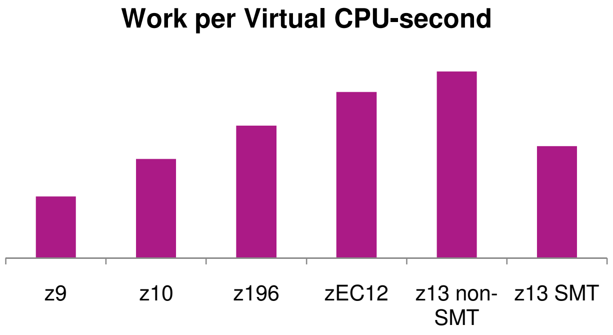
Processor Time Reporting

	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

SMT - CPU Pooling Implications

- With SMT enabled
  - **CAPACITY** limit for CPU pools is defined as processing power of a number of IFL cores …. but limit enforcement is based on thread utilization (raw time)
  - In some cases, guests in a CPU pool will not be able to complete the same amount of work as before SMT with the same capacity limit
    - Capacity limits for CPU pools might need to be increased
    - More problematic when trying to match experience from zEC12 processor than older, slower processors



Prorated Core Time (availability TBD)

- Prorated core time will divide the time a core is dispatched proportionally among the threads dispatched in that interval
  - Full time charged while a vCPU runs alongside an idle thread
  - Half time charged while a vCPU is dispatched beside another active thread
- Therefore:
  - 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** or **LIMITHARD** option
  - Limited by the **SET SHARE LIMITHARD** command

    (currently raw time is used; raw time will continue to be used when SMT is disabled)
- Only CAPACITY-based CPU pools meet requirements for sub-capacity pricing
- **QUERY CPUPOOL** will report capacity in terms of cores' worth of processing power instead of CPUs'
- Prorated core time will be reported in monitor records and the new Type F accounting record.
- Watch for APAR VM65680

How about other effects?

- Live Guest Relocation
  - Guests are allowed to relocate between SMT enabled and disabled z/VM systems because SMT is transparent to guests.
  - However, because of the above-noted differences in time, they may see their CPU time advance at different rates.
  - Their time will never go backward though!

SMT Performance on z/VM

CPUMF Display Tool

- An exec that extracts z System CPU records; internal performance experience; metrics as instructions completed, clock cycles used, and cache misses
- Available on z/VM download library: http://www.vm.ibm.com/download/packages/
- CPUMF Documentation
  - http://www.vm.ibm.com/perf/tips/cpumf.html

The process for reducing the CPUMF counters is the following:

- Start with a MONWRITE file that contains Domain 5 Record 13 records.
- Command Syntax

  EXEC CPUMFINT *filename* MONDATA *filemode*
- Resultant file:

  filename CPUMFINT filemode 🡐 *Interim file*
- Command Syntax

  EXEC CPUMFLOG *filename* CPUMFINT *filemode*
- Resultant file:

  filename $CPUMFLG *filemode* 🡐 Final report file

Sample $CPUMFLG Output

```
 _IntEnd_ LPU Typ    ___L1MP___ ___L2P____ ___L3P____ ___L4LP__
>>Mean>>    0 IFL        2.05      87.22      12.71       0.0
>>Mean>>    1 IFL        2.01      87.27      12.66       0.0
>>Mean>>    2 IFL        2.02      87.13      12.80       0.0
>>Mean>>    3 IFL        2.04      87.06      12.86       0.0
>>Mean>>    4 IFL        2.01      87.25      12.68       0.0
>>Mean>>    5 IFL        2.01      87.21      12.72       0.0
>>MofM>>                 2.02      87.19      12.74       0.0
>>AllP>>

00:46:02    0 IFL        1.99      87.00      12.93       0.0
00:46:02    1 IFL        1.99      87.04      12.91       0.0
00:46:02    2 IFL        1.96      87.01      12.93       0.0
00:46:02    3 IFL        1.96      86.93      13.01       0.0
00:46:02    4 IFL        1.97      86.95      12.98       0.0
00:46:02    5 IFL        1.99      86.96      12.96       0.0
```

Memory Footprint within the Cache

- 2% of the instructions incur an L1 cache miss. (L1MP)
- 87% of the L1 misses are sourced from the L2 cache (L2P)
- 13% of the L1 misses are sourced from the L3 cache (L3P)

SMTMET Display Tool

- An EXEC that extracts MT metrics from Domain 0 Record 2.
- Available on z/VM download library: http://www.vm.ibm.com/download/packages/
- SMTMET Documentation: http://www.vm.ibm.com/perf/tips/smtmet.html

The process for reducing the SMTMET counters is the following:

1. Start with a MONWRITE file that contains D0 R2 records.

–2. Command Syntax from CMS prompt:
SMTMET *filename* MONDATA *filemode*

–3. Resultant file from CMS prompt:
filename $SMTMET *filemode*

What are all those numbers in SMTMET output file

SMT Parameters	Definition	Range of Value
Core Busy	Percent of time at least one thread of the core was busy (aka core utilization)	0 – 100 %
Thread Density	average number of threads running on the core during the times the core was running at least one thread	1.00 – 2.00
Productivity	the ratio of the amount of work the core accomplished to the estimated amount of work the core would have accomplished if all of its busy time had been spent with all threads busy	0 – 100 %
MT Utilization	estimated core capacity in use	0 – 100 %
Capacity Factor	core's work rate compared to its work rate when running with one thread busy	100 – 200% Sometimes CF < 100%; implies workload not MT - friendly
Maximum Capacity Factor	core's work rate when running with all threads busy compared to its work rate when running with one thread busy	100 – 200% Sometimes MCF < 100%; implies workload not MT-friendly

SMTMET Resultant File Sample: Per-Core Report

```
D0R2 Per-Core Report for file: AMPDGLD1 MONDATA

Interval Core Core             Pct Core   Pct MT     Average    Pct Core
__Ended_ _ID_ Type ___Secs___ Prodctvity Utilztion_ Thread Den ___Busy___
>>Mean>>   00 IFL        30.0       93.6       86.0       1.83      92.06
>>Mean>>   01 IFL        30.0       93.5       86.0       1.83      91.92
>>Mean>>   02 IFL        30.0       93.7       86.3       1.83      92.20
>>Mean>>   03 IFL        30.0       93.6       85.9       1.84      91.86

21:32:02   00 IFL        30.0       93.4       74.0       1.84      79.26
21:32:02   01 IFL        30.0       93.1       74.4       1.83      79.91
21:32:02   02 IFL        30.0       93.8       74.2       1.85      79.11
21:32:02   03 IFL        30.0       93.8       75.4       1.85      80.39

21:32:32   00 IFL        30.0       94.1       91.2       1.86      96.86
21:32:32   01 IFL        30.0       93.3       88.8       1.84      95.16
21:32:32   02 IFL        30.0       92.7       88.3       1.82      95.28
21:32:32   03 IFL        30.0       94.0       90.7       1.86      96.42
```

SMTMET Resultant File Sample: Per-Core-type Report

```
D0R2 Per-Core-type Report for file: AMPDGLD1 MONDATA

Interval Core              Sampled   Pct Core   Pct Cap    Pct Max    Pct MT     Average
__Ended_ Type ___Secs___ ___Cores__ Prodctvity __Factor__ _Cap Fct__ Utilztion_ Thread Den
>>Mean>> IFL       120.0        4.0       93.6      156.4      167.1       86.0       1.83

21:32:02 IFL       120.0        4.0       93.6      159.2      170.1       74.5       1.84
21:32:32 IFL       120.0        4.0       93.6      158.7      169.6       89.7       1.84
21:33:02 IFL       120.0        4.0       93.2      157.7      169.2       89.3       1.83
21:33:32 IFL       119.6        4.0       93.4      158.9      170.1       89.3       1.84
21:34:02 IFL       120.0        4.0       93.3      159.2      170.5       89.4       1.84
21:34:32 IFL       120.0        4.0       93.5      158.9      169.9       89.8       1.84
21:35:02 IFL       120.0        4.0       94.1      161.1      171.1       91.2       1.86
21:35:32 IFL       120.0        4.0       93.4      159.0      170.1       89.8       1.84
21:36:02 IFL       120.0        4.0       93.8      159.5      170.0       90.5       1.85
21:36:32 IFL       120.0        4.0       93.0      158.5      170.4       88.7       1.83
21:37:02 IFL       120.0        4.0       93.4      159.1      170.3       89.7       1.84
21:37:32 IFL       120.0        4.0       93.7      159.4      170.1       90.2       1.85
21:38:02 IFL       120.0        4.0       93.7      159.0      169.6       90.4       1.85
```

SMT Performance Measurements: Conclusions

- Results in measured workloads varied widely.

- Best results were observed for applications having highly parallel activity and no single point of serialization.

- No improvements were observed for applications having a single point of serialization.

- To overcome serialization, workload adjustment should be done where possible.

- Workloads that have a heavy dependency on the z/VM master processor are not good candidates for SMT-2. In z/VM Performance Toolkit, the master processor can be identified from FCX100 CPU and FCX180 SYSCONF.

- The multithreading metrics (provided by the SMTMET tool) provide information about how well the cores perform when SMT is enabled. There is no direct relationship with workload performance (ETR, transaction response time)

- Measuring workload throughput and response time is the best way to know whether SMT is providing value to the workload

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

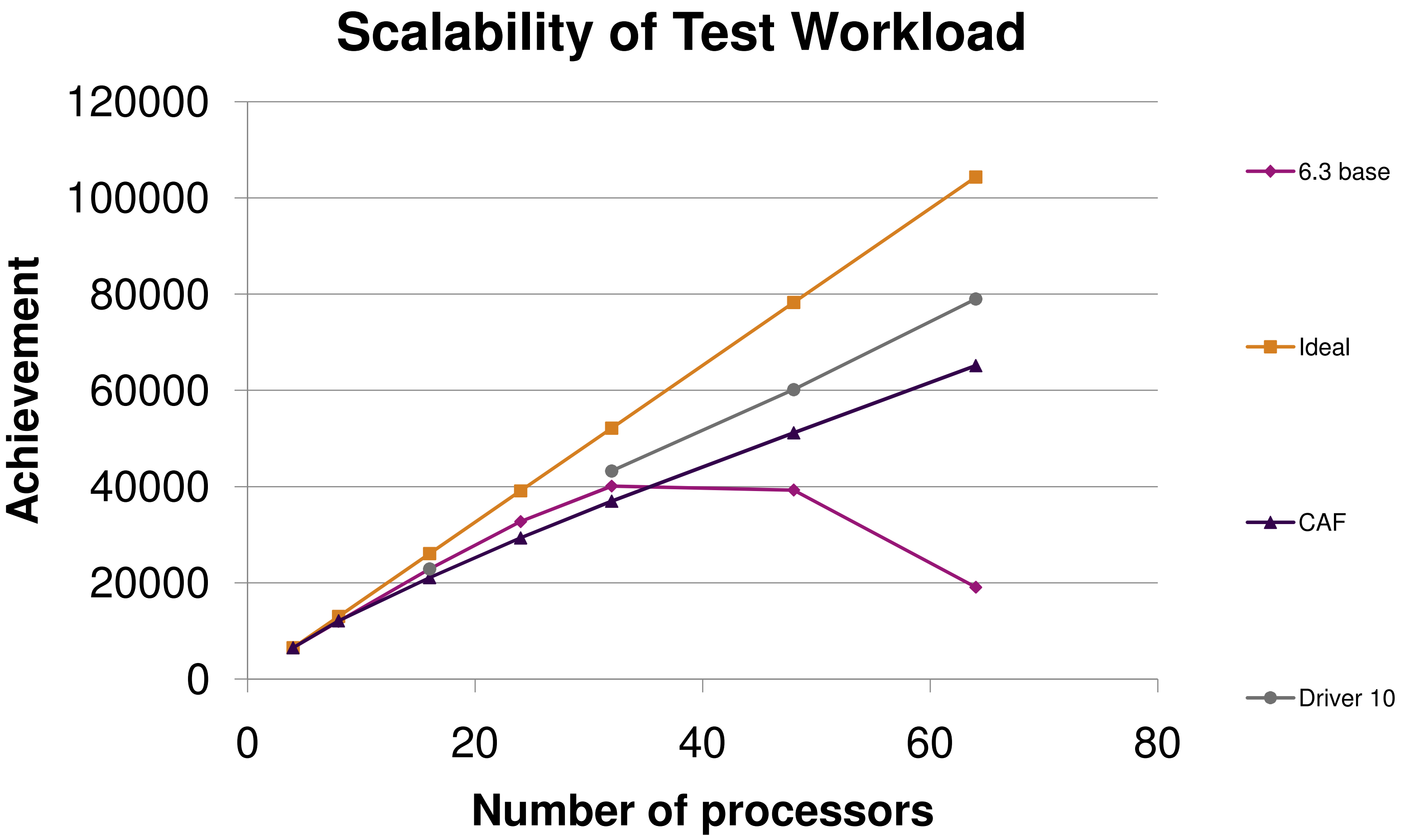
- For z13
  - With SMT disabled, increases logical processors supported from 32 to 64
  - With SMT enabled, the limit is 32 logical cores (yields at most 64 logical processors)

- 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



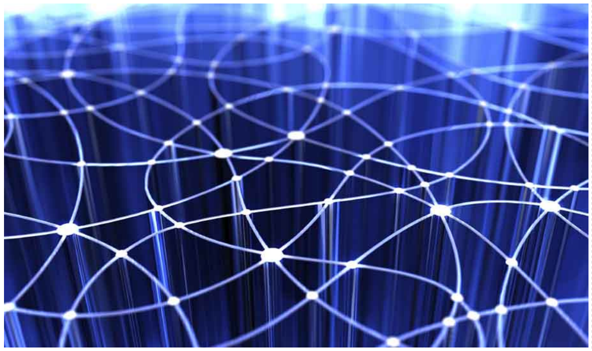
Did the CPU scalability work affect HiperDispatch at all?

- Yes!

- We no longer park entitled engines (vertical highs or mediums).
  - More efficient use of resources means that the **CPUPAD** option on the **SET SRM** command and **SRM** configuration statement is used only when global performance data is off.

  - Global performance data is a setting in the LPAR activation profile on the HMC/SE. By default, this is on.

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 the 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/

Thanks!

John Franciscovich
IBM
z/VM Design and Development
Endicott, NY
francisj@us.ibm.com

Session 17526