Performance Update for Linux on System z10 – Hints and Tips

2nd European IBM / GSE Conference for z/VSE, z/VM and Linux on System z, Leipzig, 29. Oct 2008

Hans-Joachim Picht
IBM Linux Technology Center, Germany
hans@linux.vnet.ibm.com

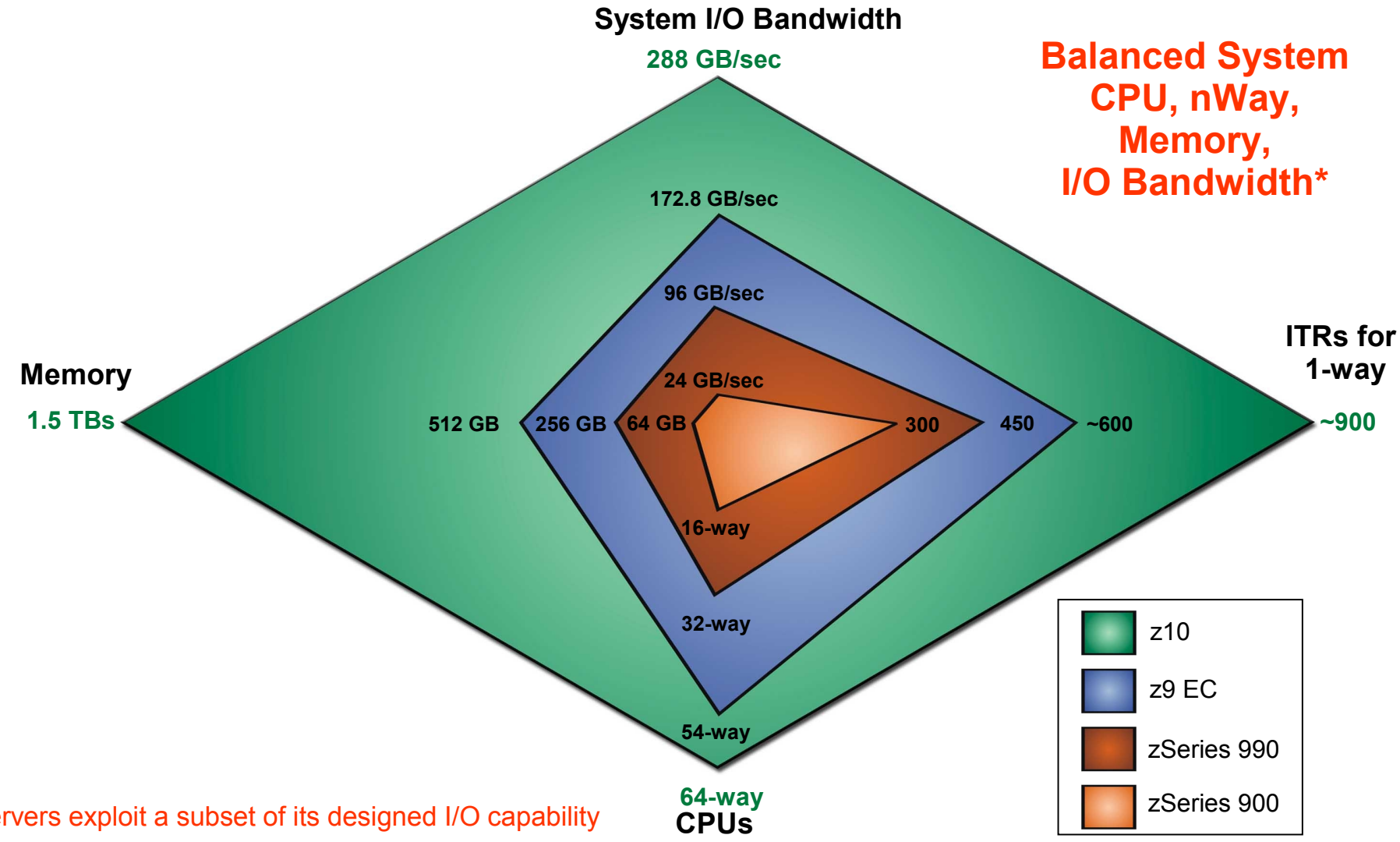
Acknowledgments

* The performance results shown in the upcoming slides have been generated by the Linux on System z performance Team at the IBM Lab in Boeblingen, Germany
* The slides were made by Martin Kammerer and Steffen Thoss (I only changed the order and the presentation template)
* I'm here today to present their results

Agenda

* Performance Update
  - z10 performance and support
  - Disk I/O
  - Cryptographic support
  - Networking
* Hints and Tips
  - Application
  - Java
  - Networking
  - Disk Performance

IBM System z – system design comparison



Balanced System CPU, nWay, Memory, I/O Bandwidth*

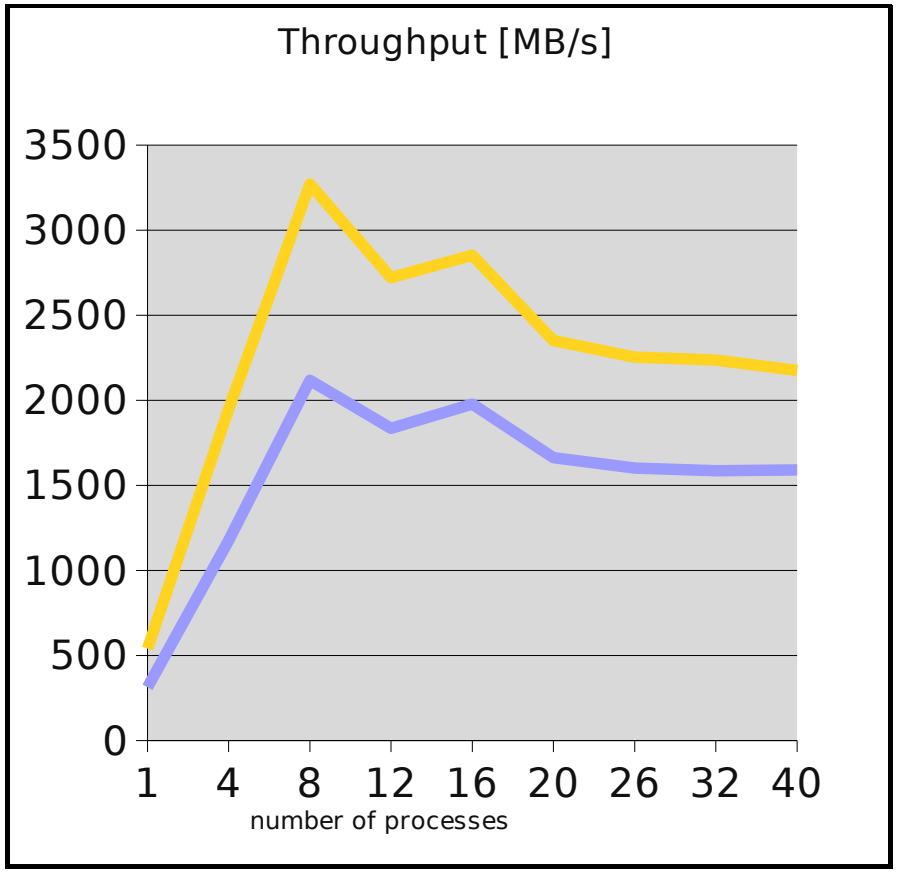
*Servers exploit a subset of its designed I/O capability

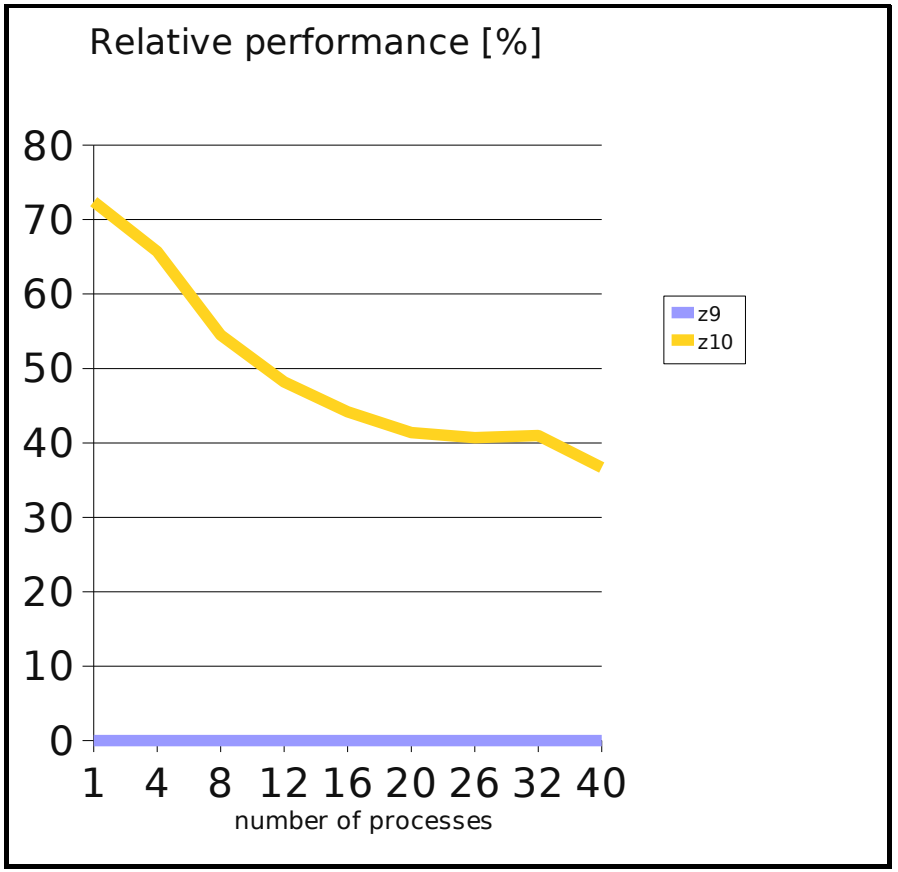
File server benchmark description

* DBench 3
  - Emulation of Netbench benchmark, rates windows file servers
  - Mainly memory operations
  - Mixed file operations workload for each process: create, write, read, append, delete
  - 8 CPUs and 1, 4, 8, 12, 16, 20, 26, 32, 40 processes
  - 2 GB memory

z10 Performance: DBench 3

* Improvement z10 versus z9:
  - Measured with 8 CPUs: average improvement is 50%



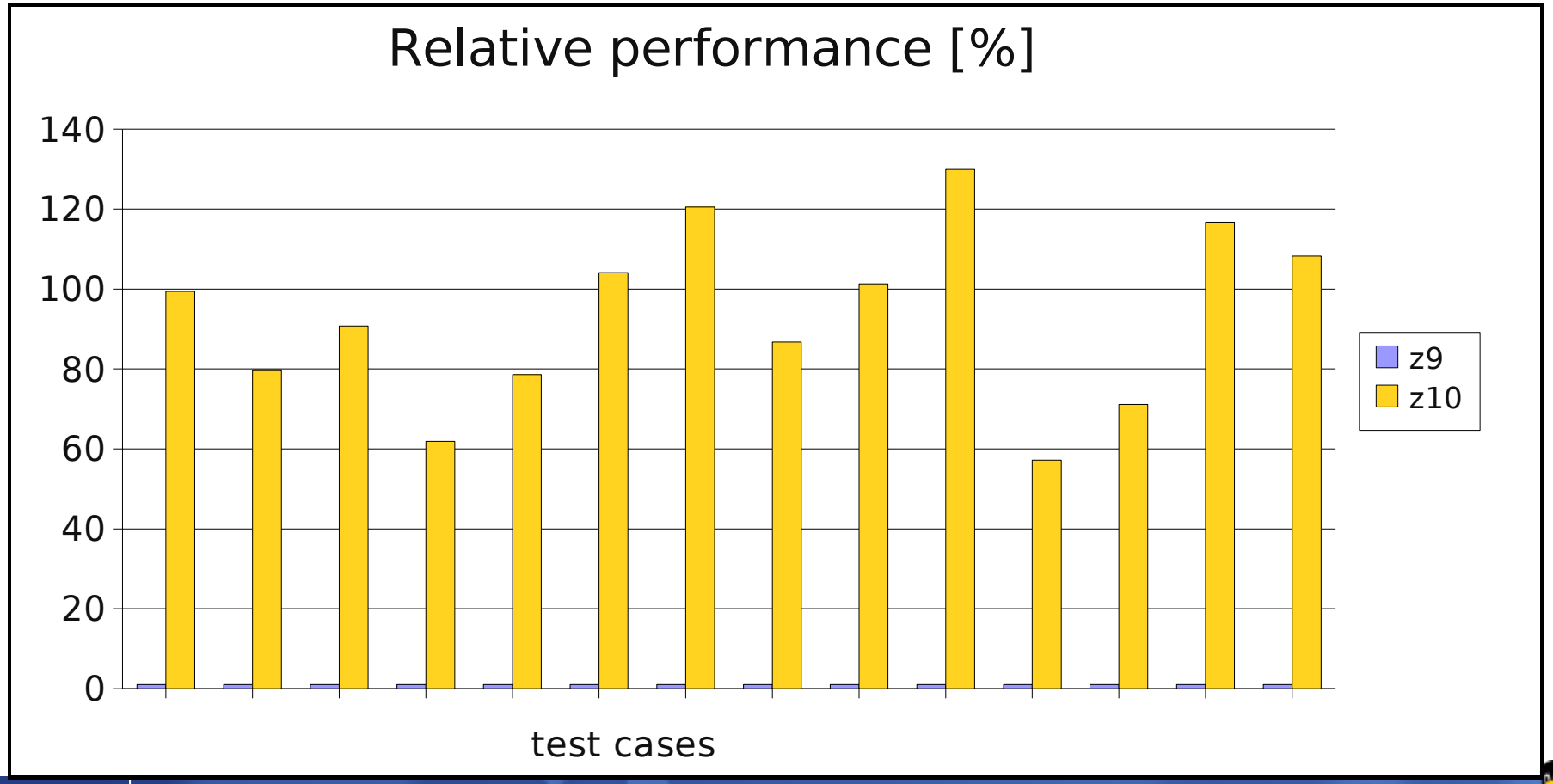


Compiler - System z features

* System z9 109 and z9 ec | bc processor support (gcc-4.1)
  - Exploit instructions provided by the extended immediate facility
  - Selected via -march=z9-109 / -mtune=z9-109
* System z10 processor support (>= gcc-4.3)
  - Exploit instructions new to z10
  - Selected via -march=z10 / -mtune=z10
* Overall integer performance enhancement on z9
  - 8% comparing gcc-3.4 and gcc-4.1 on System z
  - 5.9% comparing gcc-4.1 and gcc-4.2 on System z
  - gcc-4.3 is work in progress
* Decimal floating point support - DFP
  - Software DFP support (gcc-4.2) for older machines without hardware DFP support
  - Hardware DFP support for newer machines (gcc-4.3)

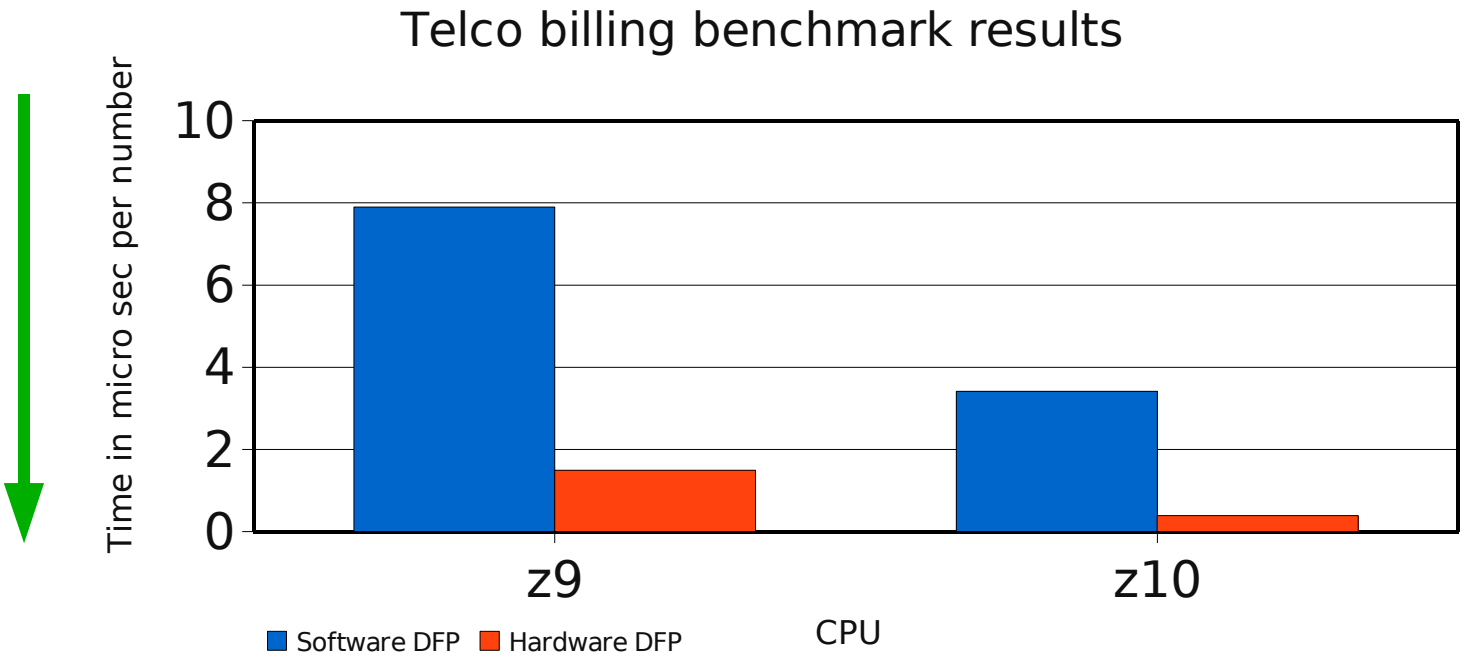
z10 performance: CPU intensive workloads

* Overall improvement with z10 versus z9: 1.9x
* Work in progress with gcc-4.3 compiler using -march=z10 option



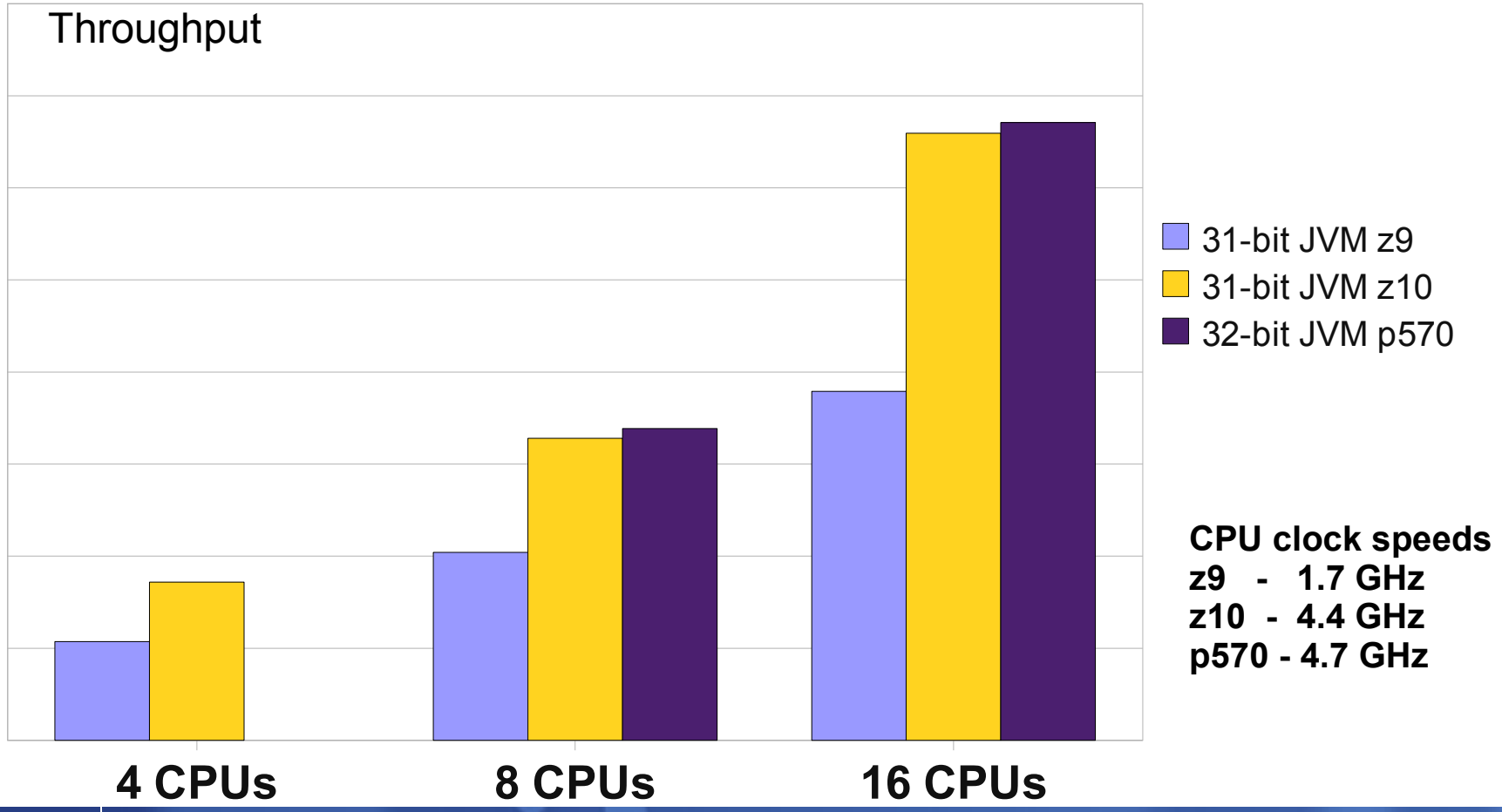
DFP - decimal floating point performance on z10

* Testcase: 1 million telephone bills
  - On z9: hardware DFP needs 1/5 of the runtime of software DFP
  - On z10: hardware DFP needs 1/9 of the runtime of software DFP
  - On z10 the test runs 2.3x/3.8x faster than on z9 (software DFP/hardware DFP)



z10 Performance: Java workload

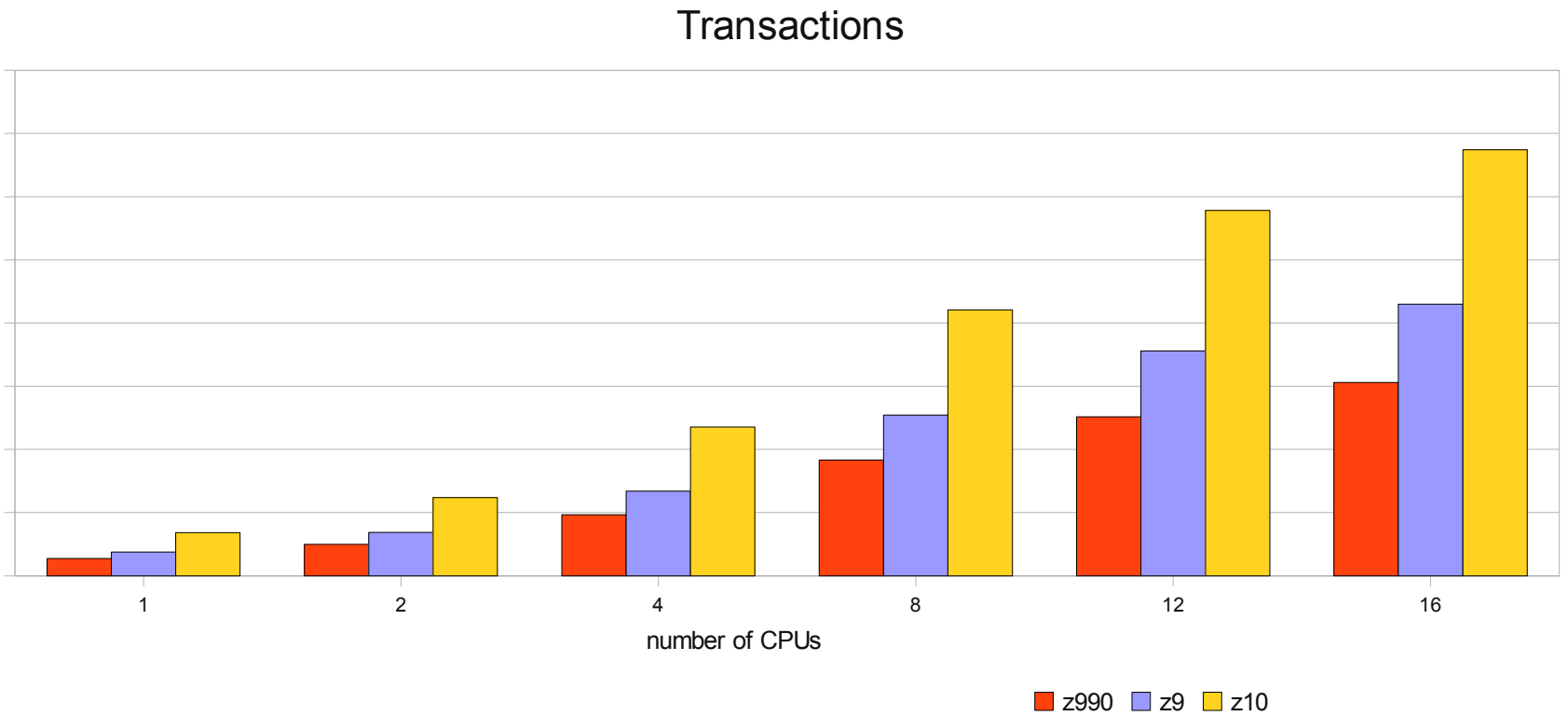
* System z versus System p



CPU clock speeds
z9 - 1.7 GHz
z10 - 4.4 GHz
p570 - 4.7 GHz

z10 with Informix IDS 11 OLTP workload

* Throughput improvements
  - z9 to z10: 65% - 82%
  - x numbers of z10 CPUs can do the same work as 2x z9 CPUs



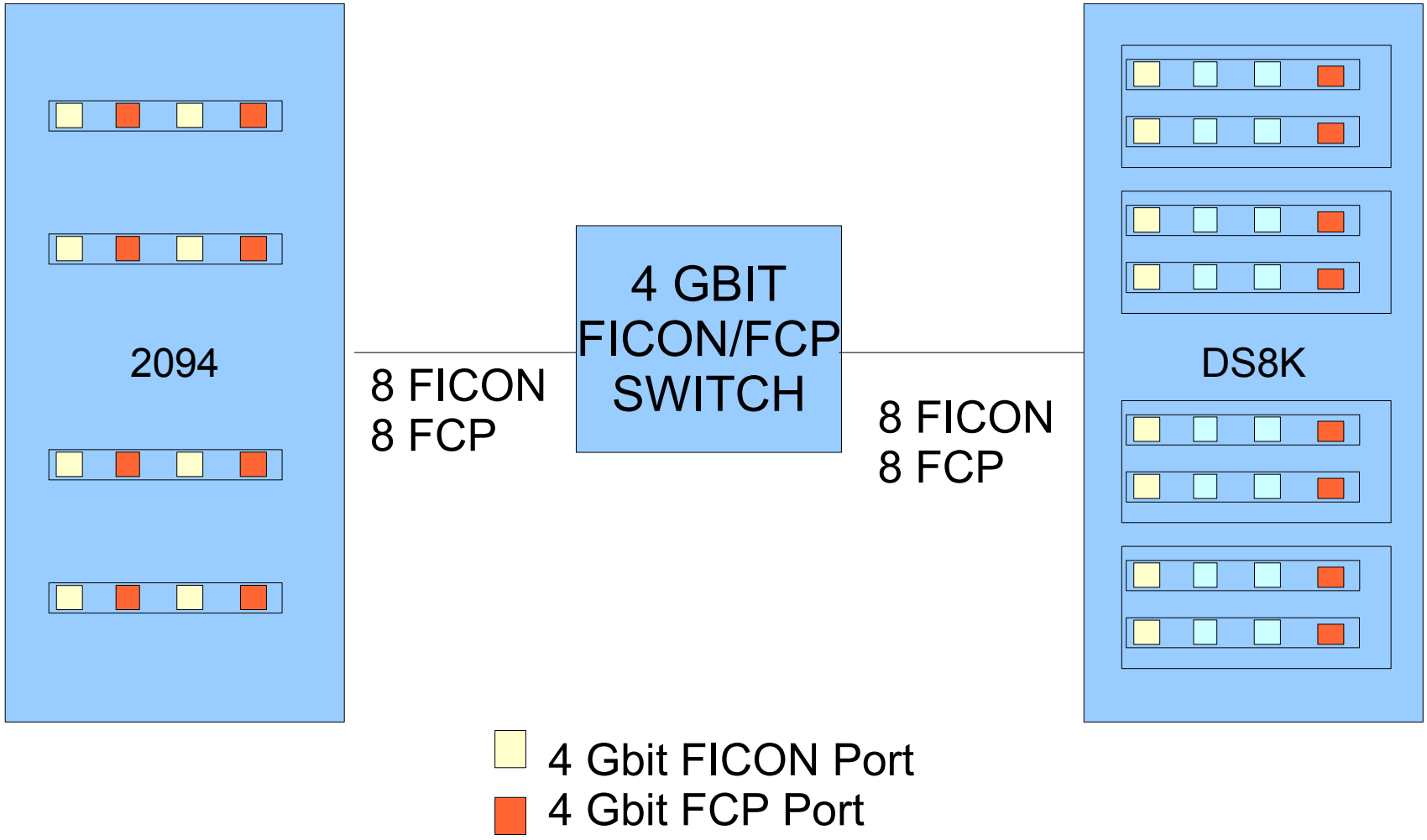
z10 performance summary

* System z evolution continues
* Performance boost from z9 to z10
* Balanced System
* Excellent on compute intensive and Java workloads
* **More to come with gcc 4.3**

Agenda

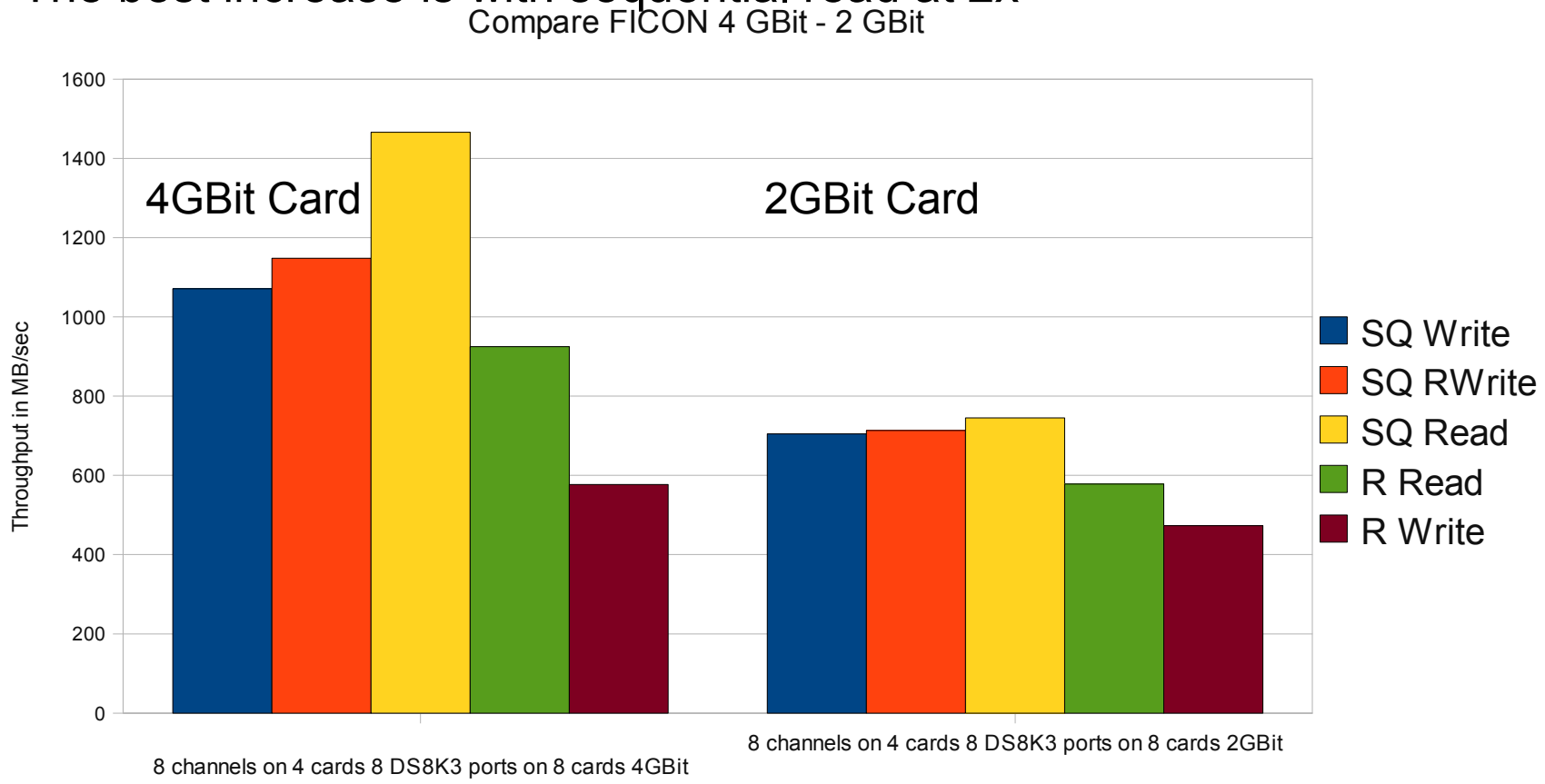
* Performance Update
  - z10 performance and support
  - Disk I/O
  - Cryptographic support
  - Networking
* Hints and Tips
  - Application
  - Java
  - Networking
  - Disk Performance
  - Tools

Configuration for 4Gbps disk I/O measurements



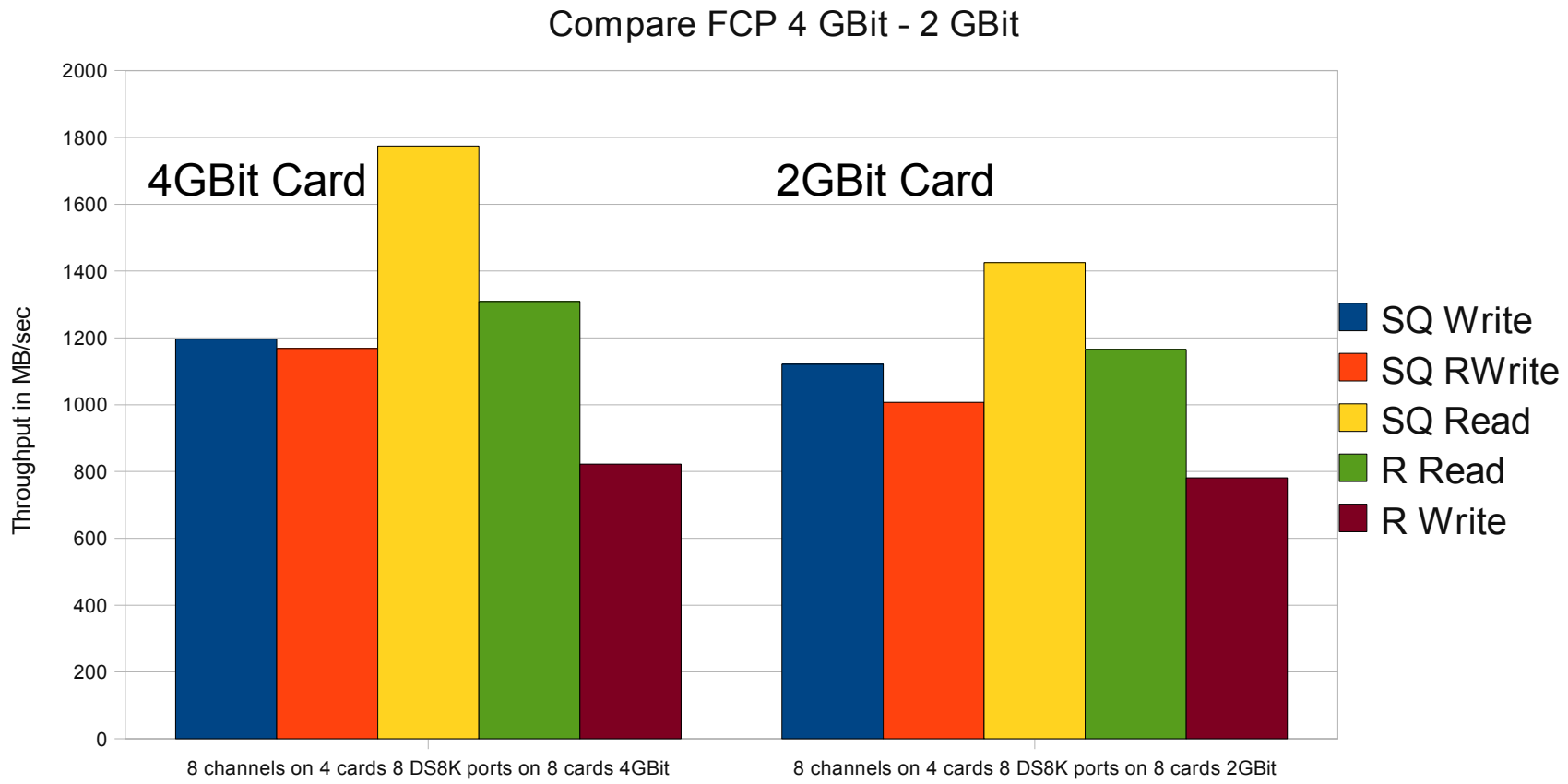
Disk I/O performance with 4Gbps links - FICON

* Strong throughput increase (average 1.6x)
* The best increase is with sequential read at 2x



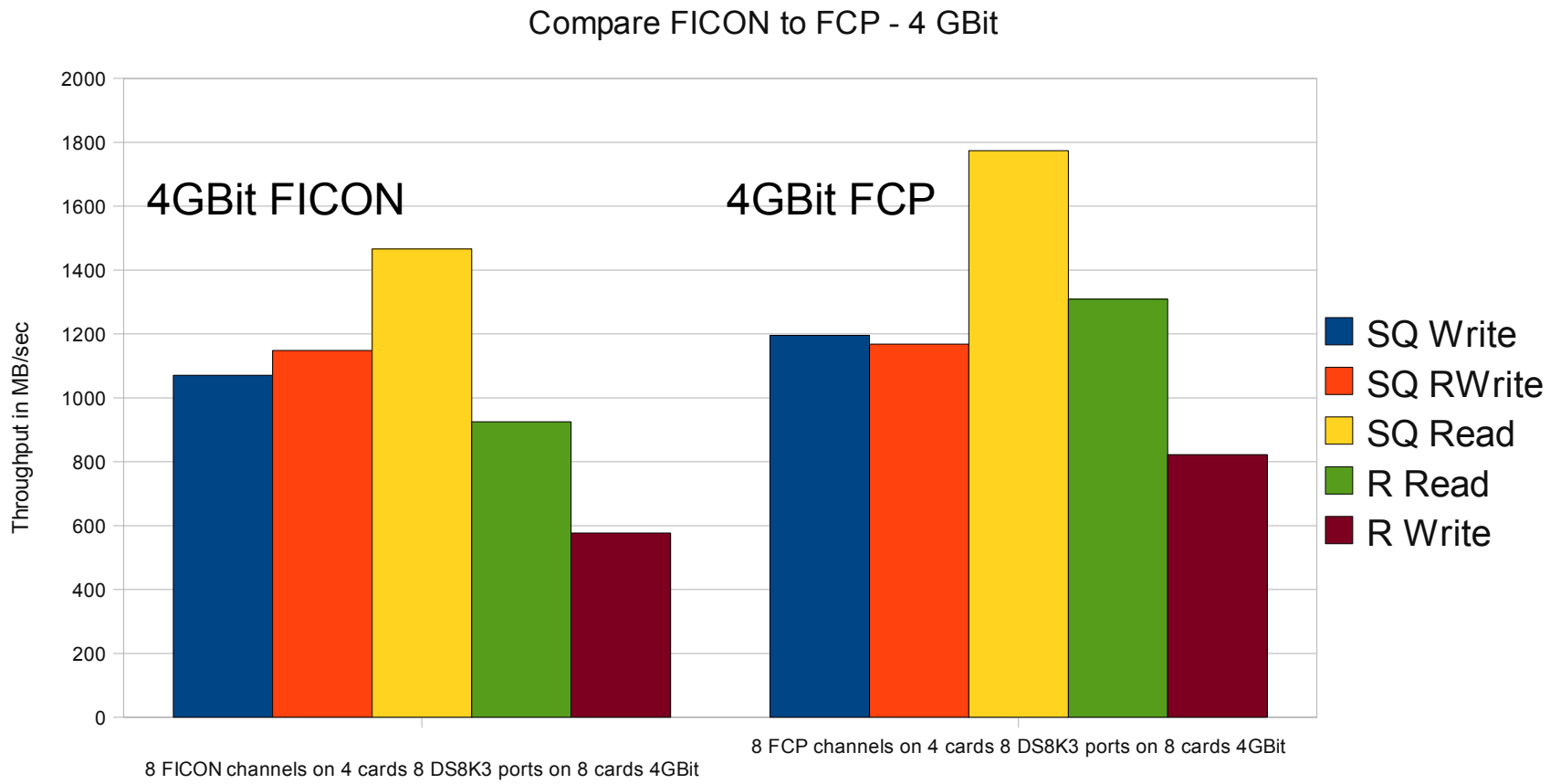
Disk I/O performance with 4Gbps links - FCP

* Moderate throughput increase
* Best improvement with sequential read at 1.25x



Disk I/O performance with 4Gbps links – FICON / FCP

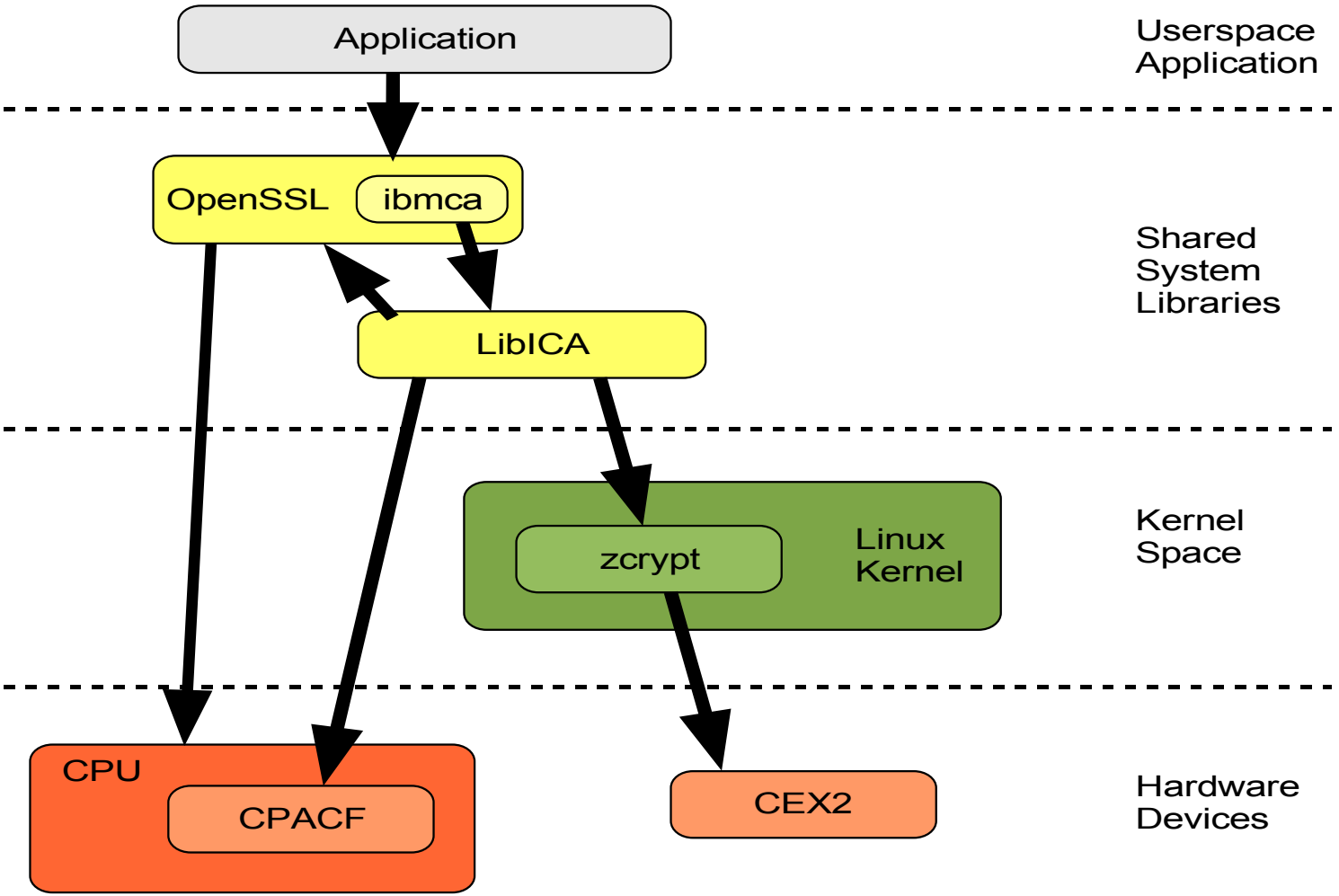
* Throughput for sequential write is similar
* FCP throughput for random I/O is 40% higher



Agenda

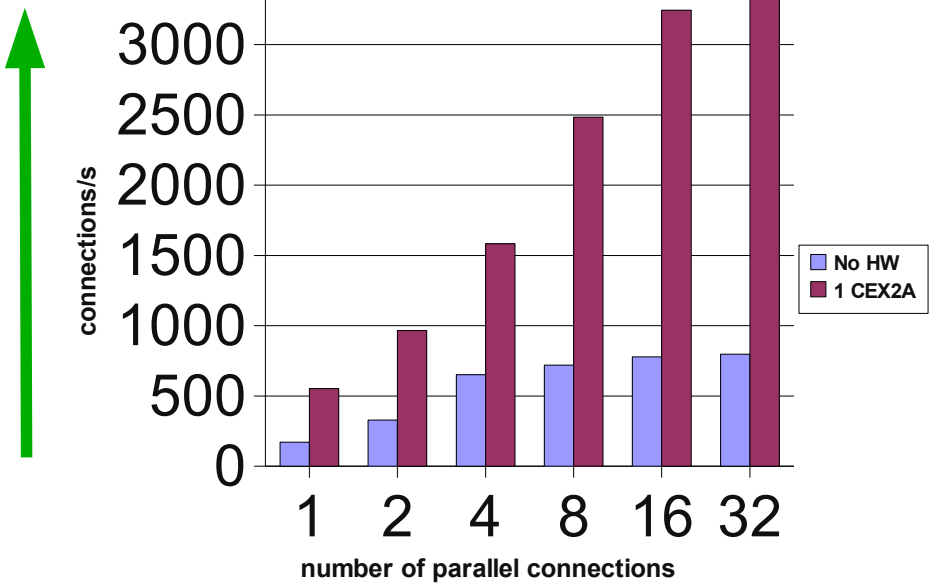
- Performance Update
  - System z hardware
  - z10 performance and support
  - Disk I/O
  - Cryptographic support
  - Networking
- Hints and Tips
  - Application
  - Java
  - Networking
  - Disk Performance
  - Tools

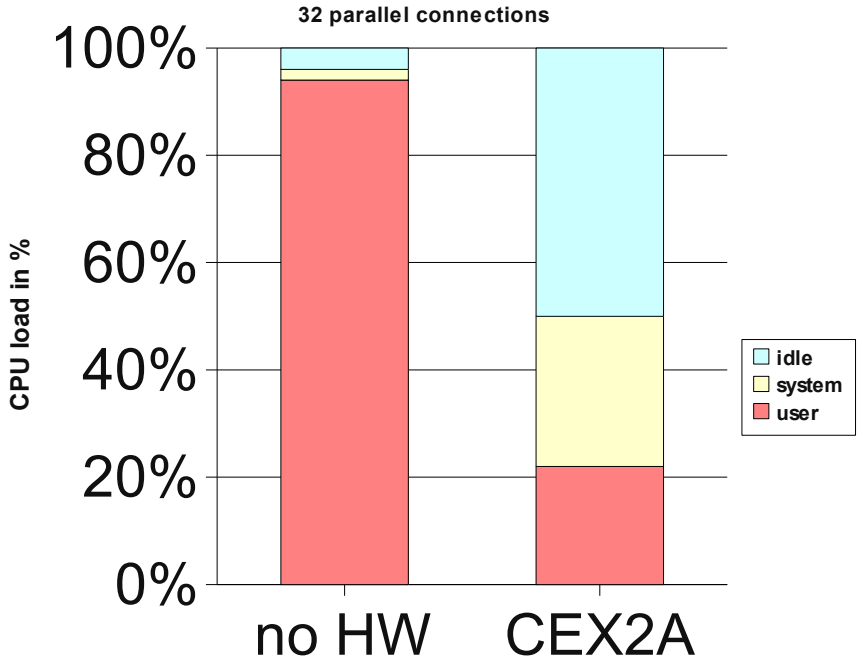
Cryptographic hardware support - SSL stack



Crypto Express2 accelerator (CEX2A) - SSL handshakes

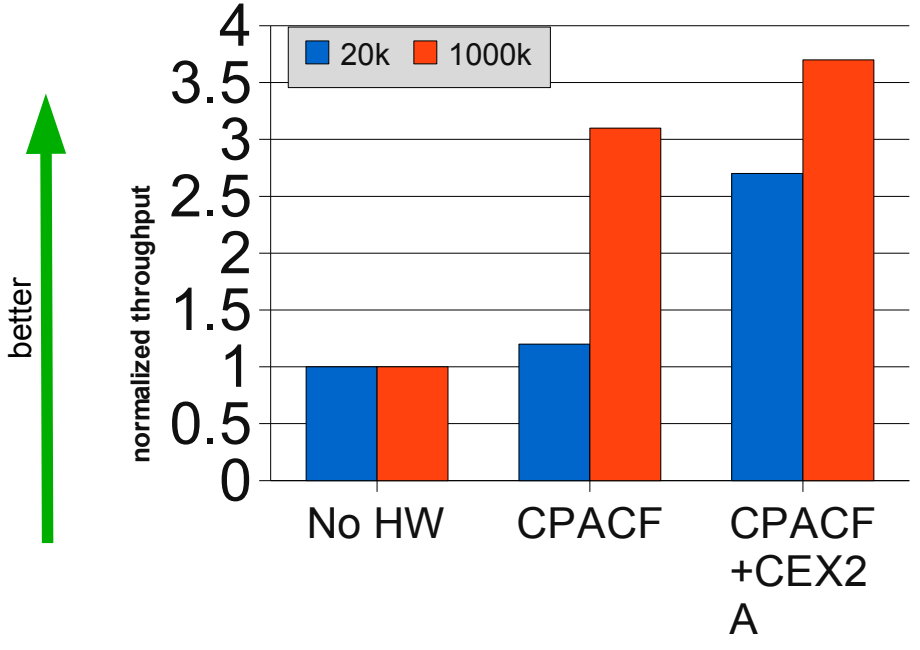
* The number of handshakes is up to 4x higher with HW support
* In the 32 connections case we save about 50% of the CPU resources

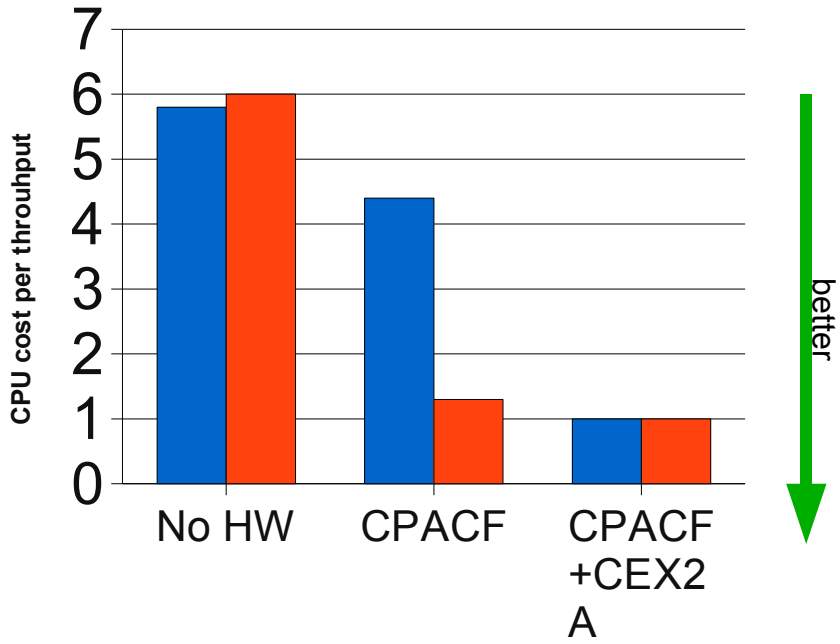




Crypto Express2 Accelerator (CEX2A) and CPACF

* The use of both hardware features leads to 3.5x more throughput
* Using software encryption costs about 6x more CPU





Agenda

* Performance Update
  - System z hardware
  - z10 performance and support
  - Disk I/O
  - Cryptographic support
  - Networking
* Hints and Tips
  - Application
  - Java
  - Networking
  - Disk Performance
  - Tools

Networking performance

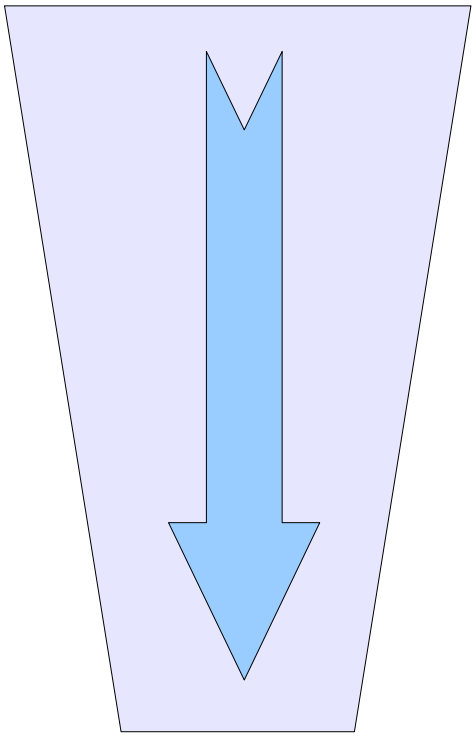
* Which connectivity to use:
  - External connectivity:
    - Use new 10 GbE cards with MTU 8992
    - Attach OSA directly to Linux guest image
  - Internal connectivity:
    - Hipersockets for LPAR-LPAR communication
    - VSwitch for guest-guest communication
* For really busy network devices consider to
  - use channel bonding
  - Increase the number of inbound buffers in the qeth driver
    - Device has to be offline
    - **# echo <number> > /sys/bus/ccwgroup/drivers/qeth/<device_bus_id>/buffer_count**

Hits and Tips

- Application
- Java
- Networking
- Disk Performance
- Tools

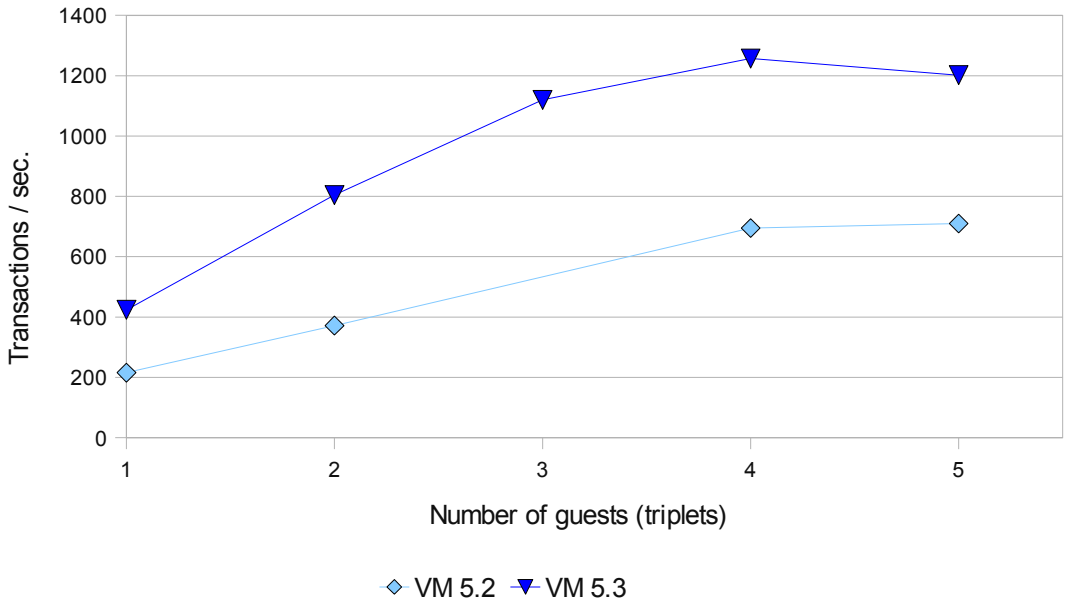
Optimize your stack in the right direction

* Diminishing effect of tuning efforts
  - Application design
  - Application implementation
  - Middleware
  - Operating system
  - Virtualization layer
  - Hardware



Impact of newer software releases

Virtualization Performance - Throughput Comparison



- Hardware:
  System z9(TM) 2094-S18
  8-way 1.65 GHz

- Software upgraded
  - z/VM 5.2 → 5.3
  - Java 1.4 → 1.5
  - WebSphere Application server 6.0.2 → 6.1.0.11
  - DB28.2 → 9.1

Keep your system current!

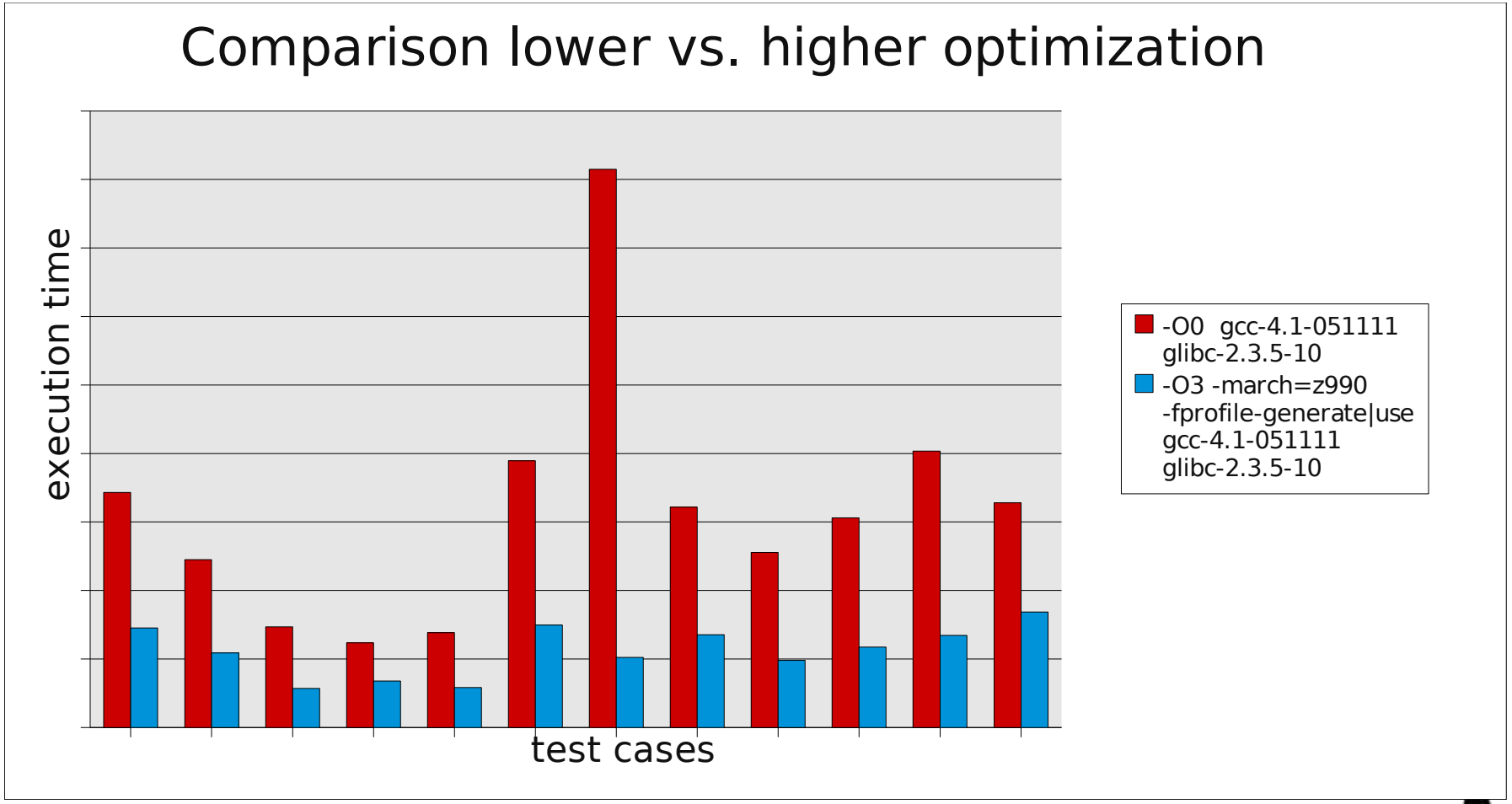
The newer software levels provides a significant improvement in throughput!

Optimizing C/C++ code

- **Use -O3 optimization as default**
  - no debugging options
    Further optimization:
  - architecture dependent options
    - **-march**=values <G5,z900,z990> <z9-109 with gcc-4.1> <z10 with patched gcc 4.3>
    - **-mtune**=values <G5,z900,z990> <z9-109 with gcc-4.1> <z10 with patched gcc 4.3>
  - inline assembler functions
- Next step: application design
  - dynamic or static linking
  - Avoid –fPIC for executables
  - right use of inlined C / C++ functions
- Fine Tuning: additional general options on a file by file basis
  - -funroll-loops -ffast-math

Results of changing compiler options

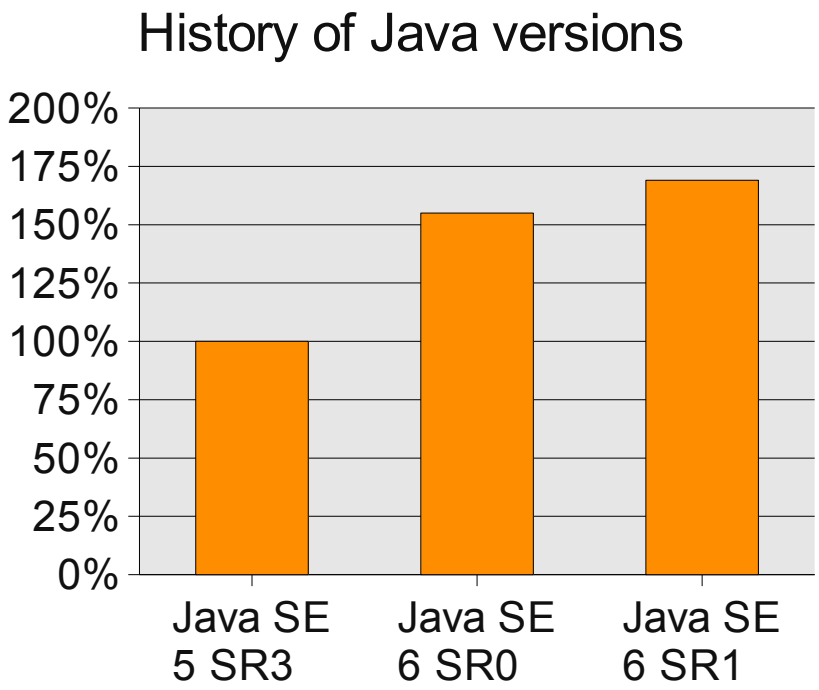
* Using -O3 instead of no optimization reduces execution time

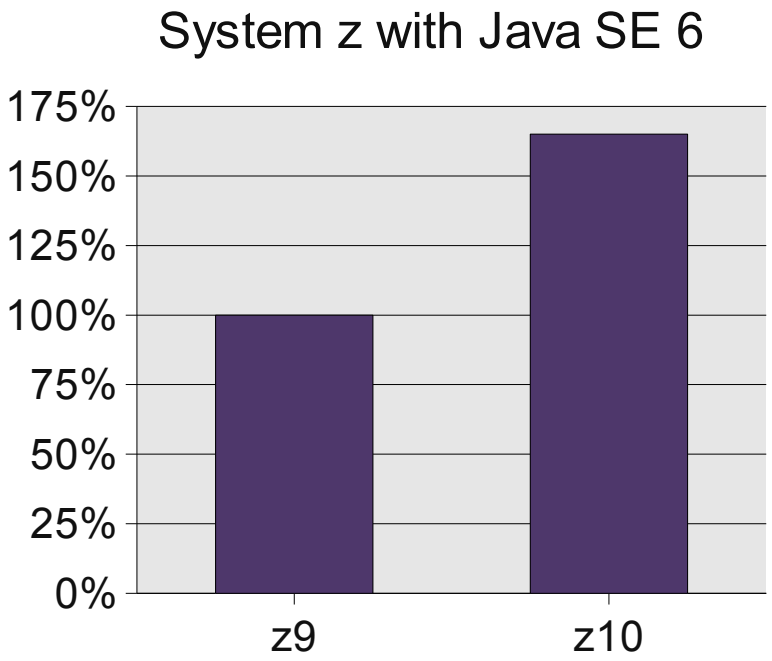


Java on servers: Workload

- **evaluates server side Java**
  - emulates 3-tier system
    - random input from user
    - middle tier business logic implemented in Java
    - no explicit database --> emulated by Java objects
- **stressed components**
  - **Java**
    - Virtual Machine (VM)
    - Just-In-Time compiler (JIT)
    - Garbage Collection (GC)
  - **Linux operating system**
    - Threads
    - CPUs
    - Caches and Memory

Java on servers: Performance Improvements





* better virtual machines (VMs) and just-in-time (JIT) compilers
* better garbage collection (GC) technologies
* improvements through new hardware

Java on servers: Heap size

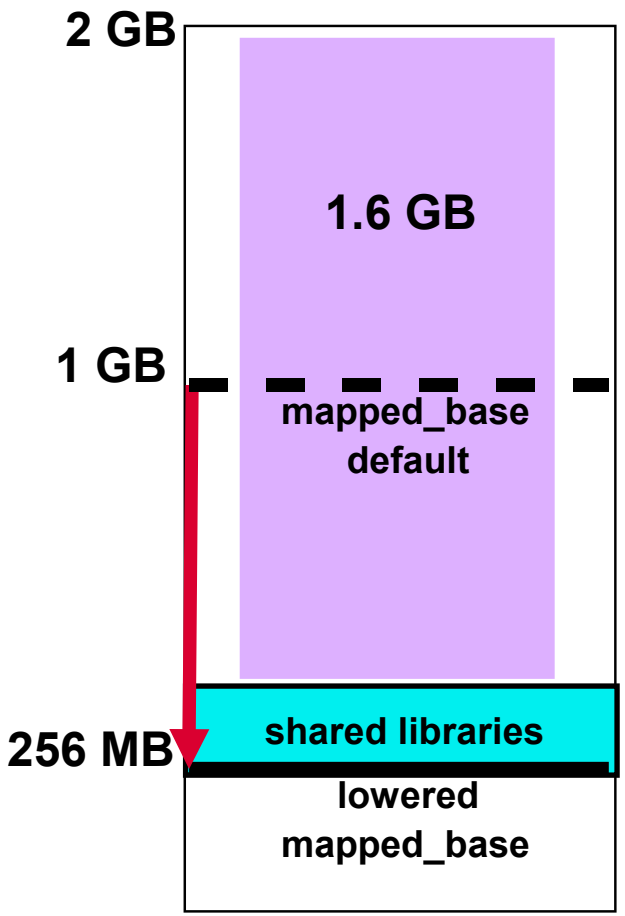
- **Heap size needs to be sized adequately**
  - maximum heap size <= available memory
    - avoids paging in Linux and z/VM
  - Heap too small: frequent garbage collection and OutOfMemoryErrors
  - Heap too big: infrequently garbage collection; Linux starts swapping
  - 31-bit Java kits: larger heap sizes up to 1.6 GB (modify memory layout)
    - also true for 31-bit Java kits in a 64-bit Linux environment
- **useful Java interpreter parameters for fine tuning**
  - setting a fixed heap size: **-Xms** (initial), **-Xmx** (maximum), when initial==maximum
  - monitor garbage collection (GC) **-verbose:gc**
  - control GC behaviour **-Xgcpolicy:[...]**
    - Gencon instead of optthruput
    - Requests the combined use of concurrent and generational GC to help minimize the time that is spent in any garbage collection pause.

Java on servers: larger heaps for 31-bit Java kits (1)

* modify Linux memory layout
  - reorder mapped base for shared libraries
* 31-bit emulation mode for Novell SLES 9,10

HOWTO:

* PID is the process ID of the process you want to change the layout (usually the bash shell)
  - **$$** gives the current shell PID, **/proc/self/**... works as well
* display memory map of any PID by
  `cat /proc/<PID>/maps`
* check the mapped base value by
  `cat /proc/<PID>/mapped_base`
* lower the value to e.g. 256 MB by
  `echo 268435456 >/proc/<PID>/mapped_base`



Java on servers: Summary & Hints

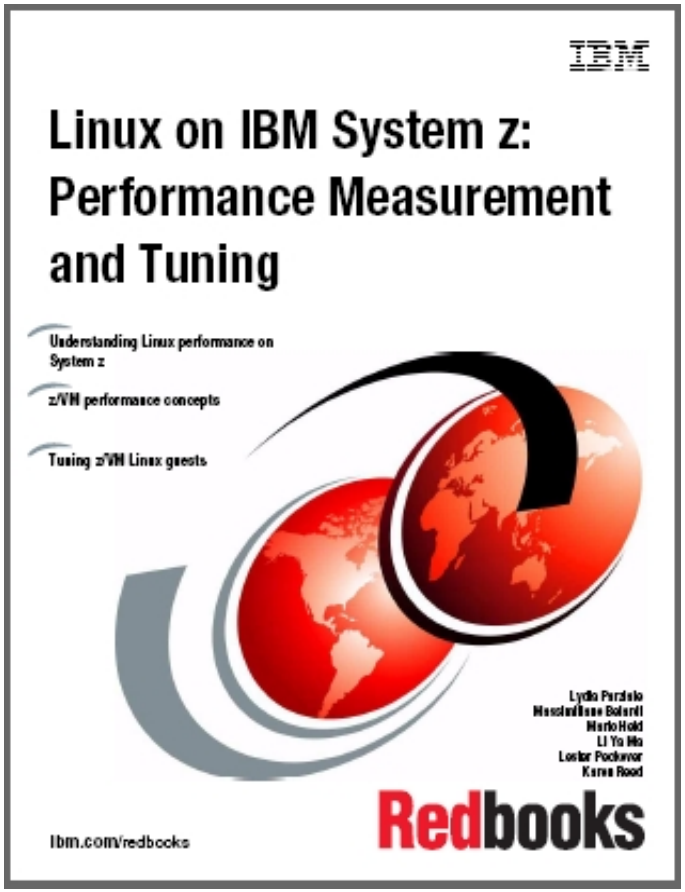
* try to use the **latest Java version**
  - up to 60% release to release improvements
  - up to 15% with newer service releases (SR) for a release
  - middleware applications often bring their own Java Kit
* make sure that you've got **JIT enabled**
  - command 'java -version' says “JIT enabled/disabled”
* lots of java interpreter -X... parameters for fine tuning
  - to get an idea type 'java -X'
* provide an optimal heap size to your application
* don't use the java interpreter in batch mode – call x-times 'java Myprog'
  - try to put the loop logic into your Java application

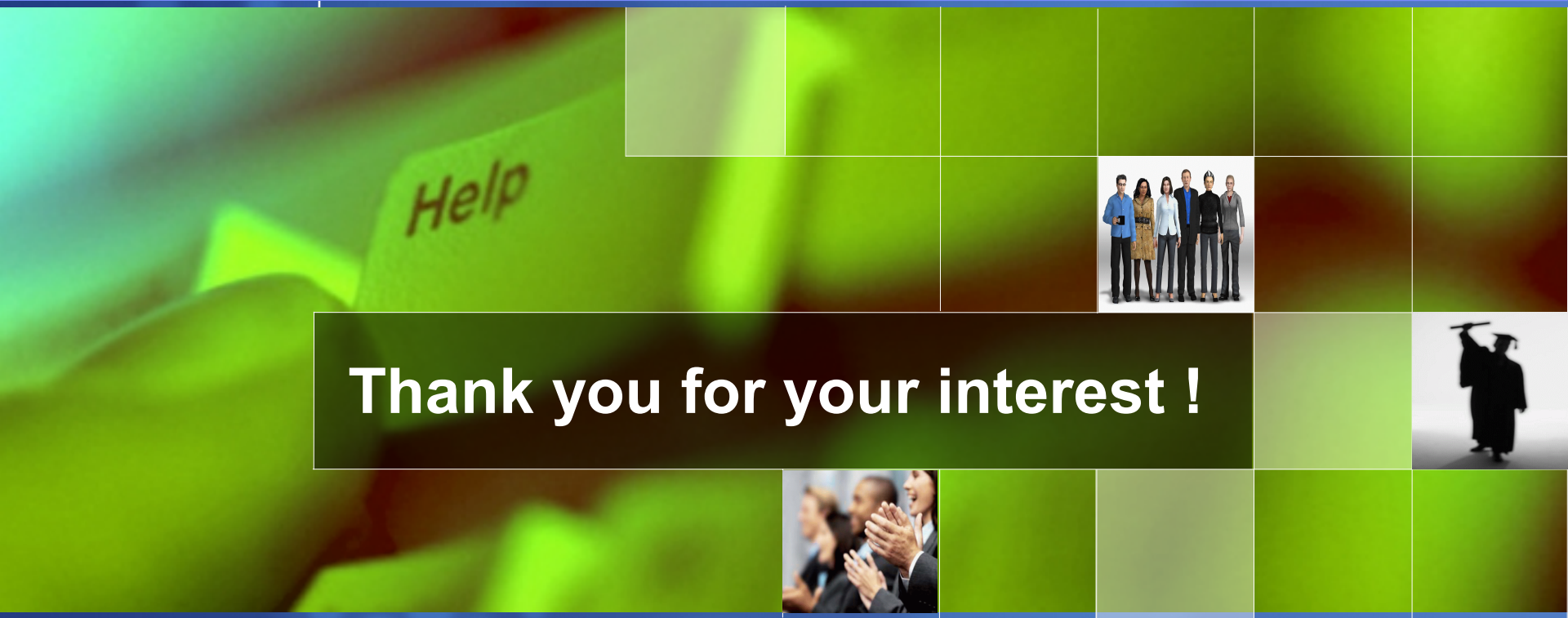
How to improve disk performance

- Hardware choices
  - Use SCSI instead of ECKD
  - Use FICON instead of ESCON
    - 4Gbps FICON > 2Gbps FICON > 1Gbps FICON
- Utilize your hardware
  - Use “striped” logical volumes from different ranks
  - Consider using HyperPAV
  - Carefully set up your storage system
  - With DS8000 – new option to stripe on storage server (see Session zLP02)
  - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_rec_dasd_optimizedisk.shtml

Visit us !

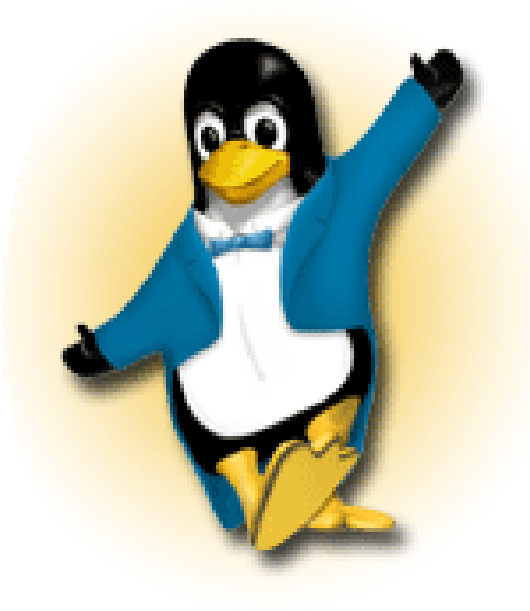
- Linux on System z: Tuning Hints & Tips
  - http://www.ibm.com/developerworks/linux/linux390/perf/
- Linux-VM Performance Website:
  - http://www.vm.ibm.com/perf/tips/linuxper.html
- IBM Redbooks
  - http://www.redbooks.ibm.com/





Thank you for your interest !

Questions?



Hans-Joachim Picht

Linux Technology Center

Linux on System z Kernel Development & Red Hat Liaision

IBM Deutschland Research & Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-1810
Mobile +49 (0)175 - 1629201
hans@linux.vnet.ibm.com

Disclaimer / Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

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.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.
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.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
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.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective 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.

Backup Slides

Linux command 'top' – the snapshot tool

* Adds new field “CPU steal time”
  - Is time Linux wanted to run, but the hypervisor was not able to schedule CPU
  - Is included in SLES10 and RHEL5

```
top - 09:50:20 up 11 min,  3 users,  load average: 8.94, 7.17, 3.82
Tasks:  78 total,   8 running,  70 sleeping,   0 stopped,   0 zombie
 Cpu0 : 38.7%us,  4.2%sy,  0.0%ni,  0.0%id,  2.4%wa,  1.8%hi,  0.0%si, 53.0%st
 Cpu1 : 38.5%us,  0.6%sy,  0.0%ni,  5.1%id,  1.3%wa,  1.9%hi,  0.0%si, 52.6%st
 Cpu2 : 54.0%us,  0.6%sy,  0.0%ni,  0.6%id,  4.9%wa,  1.2%hi,  0.0%si, 38.7%st
 Cpu3 : 49.1%us,  0.6%sy,  0.0%ni,  1.2%id,  0.0%wa,  0.0%hi,  0.0%si, 49.1%st
 Cpu4 : 35.9%us,  1.2%sy,  0.0%ni, 15.0%id,  0.6%wa,  1.8%hi,  0.0%si, 45.5%st
 Cpu5 : 43.0%us,  2.1%sy,  0.7%ni,  0.0%id,  4.2%wa,  1.4%hi,  0.0%si, 48.6%st
Mem:    251832k total,   155448k used,    96384k free,     1212k buffers
Swap:   524248k total,    17716k used,   506532k free,    18096k cached
```

Sysstat – the 'long' term data collection

- Contains four parts
  - sadc: data gatherer - stores data in binary file
  - Sar: reporting tool - reads binary file and converts it to readable output
  - Mpstat: processor utilization
  - Iostat: I/O utilization
- “steal time” included starting version 7.0.0
- Install the sysstat package and configure it depending on your distribution (crontab)
  - by default data is collected in /var/log/sa
- More info at: http://perso.orange.fr/sebastien.godard and with “man sar” on your system

oprofile – the Open Source sampling tool

* oprofile offers profiling of all running code on Linux systems, providing a variety of statistics.
  - By default, kernel mode and user mode information is gathered for configurable events
* System z hardware currently does not have support for hardware performance counters, instead timer interrupt is used
  - Enable the hz_timer(!)
* The timer is set to whatever the jiffy rate is and is not user-settable
* Novell / SUSE: oprofile is on the SDK CDs
* More info at:
  - http://oprofile.sourceforge.net/docs/
  - http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/sysadmin-guide/ch-o

oprofile – short HowTo

```
sysctl -w kernel.hz_timer=1

gunzip /boot/vmlinux-2.6.16.46-0.4-default.gz
```

- specify the kernel level of `uname -r`

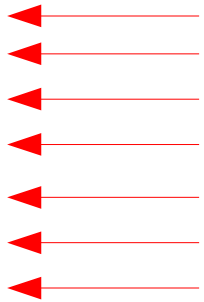
```
opcontrol --vmlinux=/boot/vmlinux-2.6.16.46-0.4-
  default

opcontrol --start

<DO TEST>

opcontrol --shutdown

opreport
```

any next test to run? If yes

```
opcontrol --reset
```

opreport

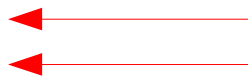
```
>opreport
CPU: CPU with timer interrupt, speed 0 MHz (estimated)
Profiling through timer interrupt
          TIMER:0|
  samples|      %|
------------------
   140642 94.0617 vmlinux-2.6.16.46-0.4-default
     3071  2.0539 libc-2.4.so
     1925  1.2874 dbench
     1922  1.2854 ext3
     1442  0.9644 jbd
      349  0.2334 dasd_mod
      152  0.1017 apparmor
        6  0.0040 oprofiled
        5  0.0033 bash
        5  0.0033 ld-2.4.so
        1 6.7e-04 dasd_eckd_mod
        1 6.7e-04 oprofile
```



Kernel
glibc
application
file system
journaling
dasd driver
security
...

opreport -l

```
>opreport -l
warning: /apparmor could not be found.
warning: /dasd_eckd_mod could not be found.
warning: /dasd_mod could not be found.
warning: /ext3 could not be found.
warning: /jbd could not be found.
warning: /oprofile could not be found.
CPU: CPU with timer interrupt, speed 0 MHz (estimated)
Profiling through timer interrupt
samples  %        app name                 symbol name
130852   87.5141  vmlinux-2.6.16.46-0.4-default cpu_idle
1922      1.2854  ext3                     (no symbols)
1442      0.9644  jbd                      (no symbols)
734       0.4909  vmlinux-2.6.16.46-0.4-default memcpy
662       0.4427  libc-2.4.so              strchr
619       0.4140  dbench                   next_token
567       0.3792  vmlinux-2.6.16.46-0.4-default do_gettimeofday
536       0.3585  vmlinux-2.6.16.46-0.4-default __link_path_walk
525       0.3511  vmlinux-2.6.16.46-0.4-default copy_to_user_std
435       0.2909  libc-2.4.so              strstr
413       0.2762  dbench                   child_run
349       0.2334  dasd_mod                 (no symbols)
347       0.2321  vmlinux-2.6.16.46-0.4-default _spin_lock
328       0.2194  vmlinux-2.6.16.46-0.4-default sysc_do_svc
285       0.1906  dbench                   all_string_sub
283       0.1893  vmlinux-2.6.16.46-0.4-default __d_lookup
251       0.1679  vmlinux-2.6.16.46-0.4-default __find_get_block
231       0.1545  libc-2.4.so              ____strtol_l_internal
216       0.1445  dbench                   vsnprintf
209       0.1398  vmlinux-2.6.16.46-0.4-default filldir64
205       0.1371  vmlinux-2.6.16.46-0.4-default memset
196       0.1311  vmlinux-2.6.16.46-0.4-default _atomic_dec_and_lock
166       0.1110  vmlinux-2.6.16.46-0.4-default strchr
155       0.1037  libc-2.4.so              memmove
152       0.1017  apparmor                 (no symbols)
148       0.0990  libc-2.4.so              readdir
147       0.0983  vmlinux-2.6.16.46-0.4-default __brelse
146       0.0976  vmlinux-2.6.16.46-0.4-default generic_file_buffered_write
144       0.0963  vmlinux-2.6.16.46-0.4-default generic_permission
140       0.0936  vmlinux-2.6.16.46-0.4-default __getblk
140       0.0936  vmlinux-2.6.16.46-0.4-default kmem_cache_free
```



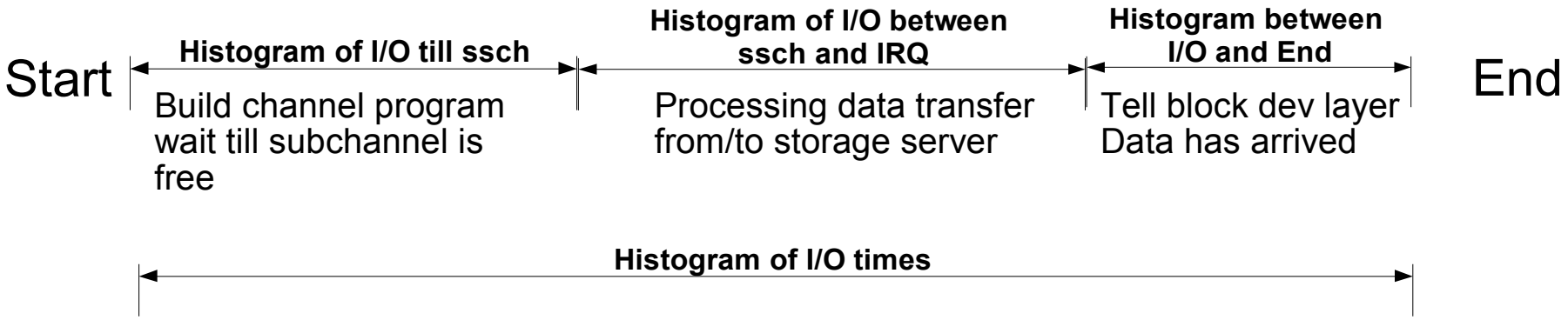
almost idle
unresolved symbols

opreport -l --image-path=...

```
>opreport -l --image-path=/lib/modules/2.6.16.46-0.4-default/kernel/fs/ext3/,/lib/modules/2.6.16.46-0.4-
default/kernel/fs/jbd/,/lib/modules/2.6.16.46-0.4-
default/kernel/drivers/s390/block/,/lib/modules/2.6.16.46-0.4-
default/kernel/security/apparmor/,/lib/modules/2.6.16.46-0.4-default/kernel/arch/s390/oprofile
CPU: CPU with timer interrupt, speed 0 MHz (estimated)
Profiling through timer interrupt
samples  %        image name               app name                 symbol name
130852   87.5141  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default cpu_idle
734       0.4909  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default memcpy
662       0.4427  libc-2.4.so              libc-2.4.so              strchr
619       0.4140  dbench                   dbench                   next_token
567       0.3792  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default do_gettimeofday
536       0.3585  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default __link_path_walk
525       0.3511  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default copy_to_user_std
435       0.2909  libc-2.4.so              libc-2.4.so              strstr
413       0.2762  dbench                   dbench                   child_run
361       0.2414  ext3.ko                  ext3                     ext3_get_block_handle
347       0.2321  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default _spin_lock
328       0.2194  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default sysc_do_svc
285       0.1906  dbench                   dbench                   all_string_sub
283       0.1893  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default __d_lookup
251       0.1679  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default __find_get_block
231       0.1545  libc-2.4.so              libc-2.4.so              ____strtol_l_internal
226       0.1511  ext3.ko                  ext3                     ext3_try_to_allocate
223       0.1491  dasd_mod.ko              dasd_mod                 dasd_smalloc_request
216       0.1445  dbench                   dbench                   vsnprintf
209       0.1398  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default filldir64
205       0.1371  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default memset
196       0.1311  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default _atomic_dec_and_lock
188       0.1257  ext3.ko                  ext3                     ext3_new_inode
166       0.1110  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default strchr
157       0.1050  jbd.ko                   jbd                      journal_init_dev
155       0.1037  libc-2.4.so              libc-2.4.so              memmove
148       0.0990  libc-2.4.so              libc-2.4.so              readdir
147       0.0983  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default __brelse
146       0.0976  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default generic_file_buffered_write
144       0.0963  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default generic_permission
140       0.0936  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default __getblk
140       0.0936  vmlinux-2.6.16.46-0.4-default vmlinux-2.6.16.46-0.4-default kmem_cache_free
```

dasd statistics – data collection

* Collects statistics (mostly processing times) of IO operations
* Each line represents a histogram of times for a certain operation
* Operations split up into the following :



dasd statistics - controls

* Linux can collect performance stats on DASD activity as seen by Linux(!)
* Turn on with

```
echo on > /proc/dasd/statistics
```

* Turn off with

```
echo off > /proc/dasd/statistics
```

* To reset: turn off and then on again
* Can be read for the whole system by
  - `cat /proc/dasd/statistics`
* Can be read for individual DASDs by
  - `tunedasd -P /dev/dasda`

dasd statistics - report

request size <= 8KB

response time <= 2ms

```
21155901 dasd I/O requests
with 433275376 sectors(512B each)
   __<4    ___8    __16    __32    __64    _128    _256    _512    __1k    __2k    __4k    __8k    _16k    _32k    _64k    128k
   _256    _512    __1M    __2M    __4M    __8M    _16M    _32M    _64M    128M    256M    512M    __1G    __2G    __4G    _>4G
Histogram of sizes (512B secs)
      0       0 3774298  838941  352193  232188   43222   30563   16163    1403       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O times (microseconds)
      0       0       0       0       0       0       0       2 3005329  352056  726353  671293  355198  147238   29245    2201
     51       3       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O times per sector
      0       0   24686  204678  524222 2803252  500319  537993  249088  316175  111592   15932    1005      26       3       0
      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O time till ssch
3498191   51615   86168   21601    2756    1927    4348   22793  177758  138465  955964  214188   61200   42284    9075     621
     14       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O time between ssch and irq
      0       0       0       0       0       0       0       4 4252115  408592   78374  122000  309317  108290    9848     416
     13       3       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O time between ssch and irq per sector
      0       0   41819  517428  890743 3323127   21897   23329  103966  280533   79777    6056     282      10       2       0
      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Histogram of I/O time between irq and end
4531949  633301   75411   41903    4984     791     516      48      40       3       3      20       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
# of req in chanq at enqueuing (1..32)
      0 3658672  277906  128989   97542 1125789      27       0       0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
```