Linux for System z performance update

Session number 2590

Martin Kammerer
kammerer@de.ibm.com

Feb 28, 2008 8:00 - 9:00

© Disney

Trademarks

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

DB2*
DB2 Connect
DB2 Universal Database
e-business logo
IBM*
IBM eServer
IBM logo*
Informix®

System z
Tivoli*
WebSphere*
z/VM*
zSeries*
z/OS*

ECKD
Enterprise Storage Server®
FICON
FICON Express
HiperSocket
OSA
OSA Express

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* 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.

Agenda

- **System z hardware**
- Hardware improvements
  - Processor
  - Networking
  - Disk / Tape
  - Cryptography
- Software improvements
  - Compiler
  - Java
  - WebSEAL
  - Tivoli Storage Manager
- Distribution improvements
  - Red Hat
  - Novell SUSE

Our hardware for measurements

SHARE
Technology · Connections · Results

2084-B16 (z990)
0.83ns (1.2 GHz)
2 Books, 16 CPUs
2 * 32 MB L2 Cache
80 GB
FICON-Express2

2094-S18 (z9-109)
0.58ns (1.7GHz)
2 Books, 18 CPUs
2*40 MB L2 Cache
128 GB
FICON-Express4

HiperSockets
OSA-Express2 (10)GbE

2105-800 (Shark)
32 GB Cache
1 GB NVS
128 * 72 GB disks
15.000 RPM
FCP (2 Gbps)
FICON (2 Gbps)

2107-922 (DS8000)
256 GB Cache
8 GB NVS
256 * 72 GB disks
15.000 RPM
FCP (4 Gbps)
FICON (4 Gbps)

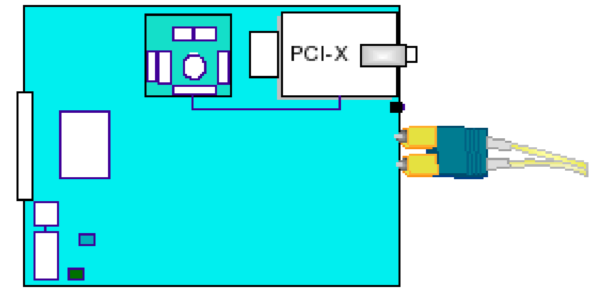
Agenda

- System z hardware
- **Hardware improvements**
  - **Processor**
  - **Networking**
  - **Disk / Tape**
  - **Cryptography**
- Software improvements
  - Compiler
  - Java
  - WebSEAL
  - Tivoli Storage Manager
- Distribution improvements
  - Red Hat
  - Novell/SUSE

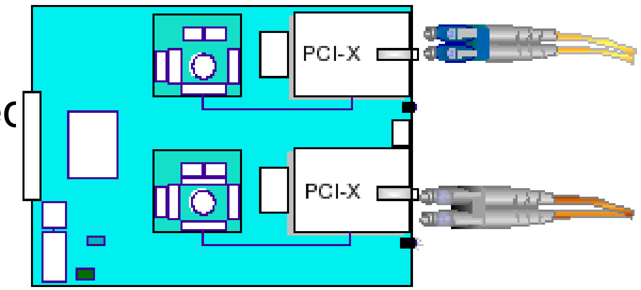
OSA-Express2

- Newest member – 10 Gb Ethernet LR (long reach)
  - One port per feature
- New – Gb Ethernet features
  - Gigabit Ethernet LX (long wavelength)
  - Gigabit Ethernet SX (short wavelength
- Support offered by both 10 GbE and 1 GbE
  - Layer 2 support
  - Up to 1920 TCP/IP stacks for improved virtualization
  - Large send for CPU efficiency

10 Gigabit Ethernet Feature
3368

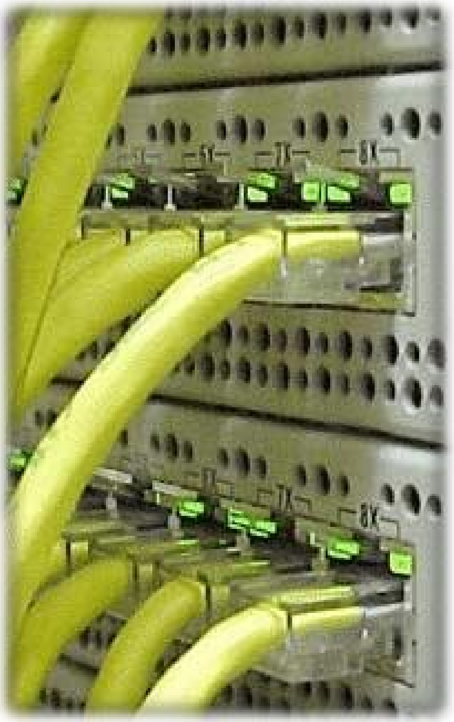


Gigabit Ethernet Features
3364 (LX), 3365 (SX)

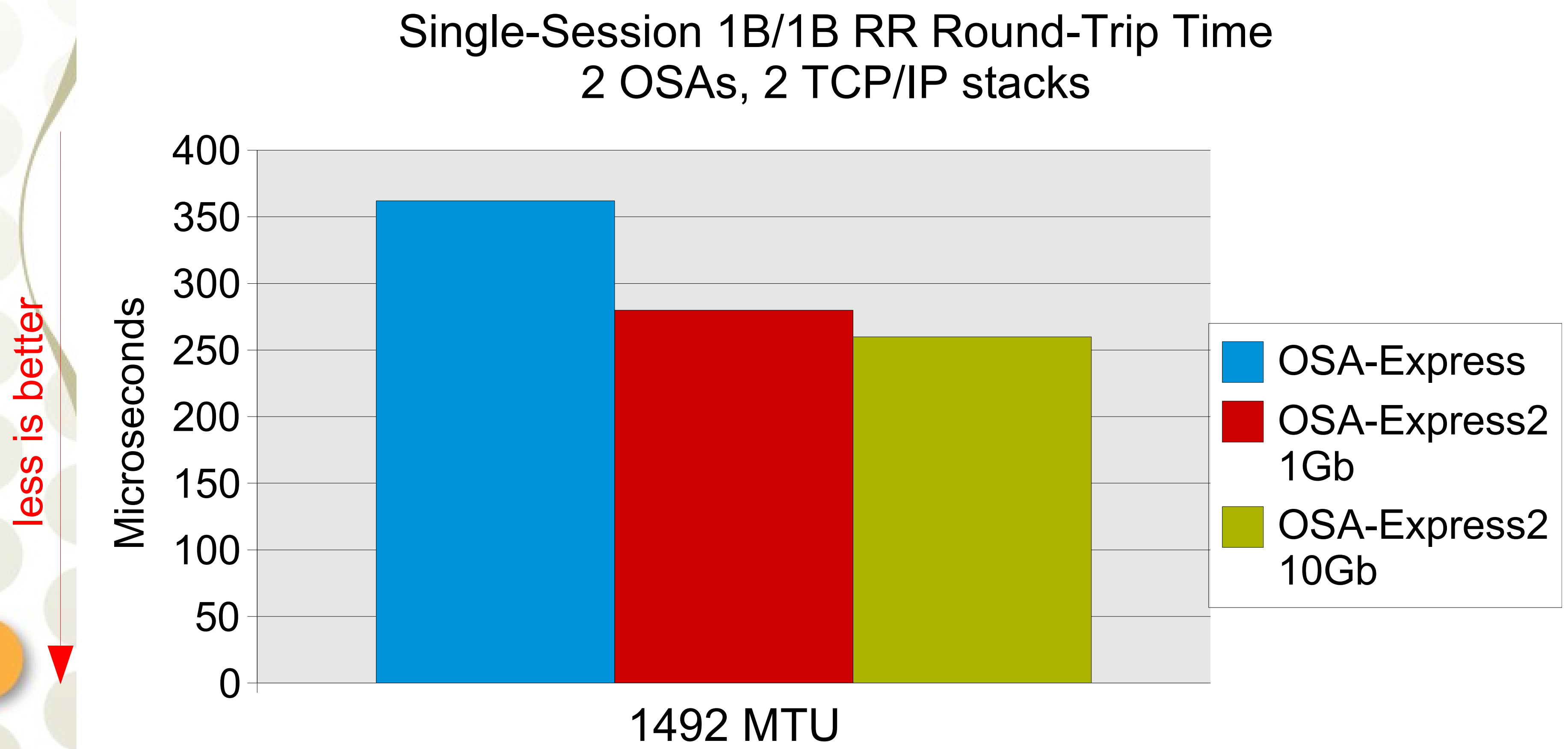


Networking benchmark

- AWM
- Several workload models
  - transactional workload
  - streaming workload
  - mixed workload
- Measured with GbE (QDIO), Hipersockets, and virtual connections in z/VM
- Throughput and cost (CPU) measurements

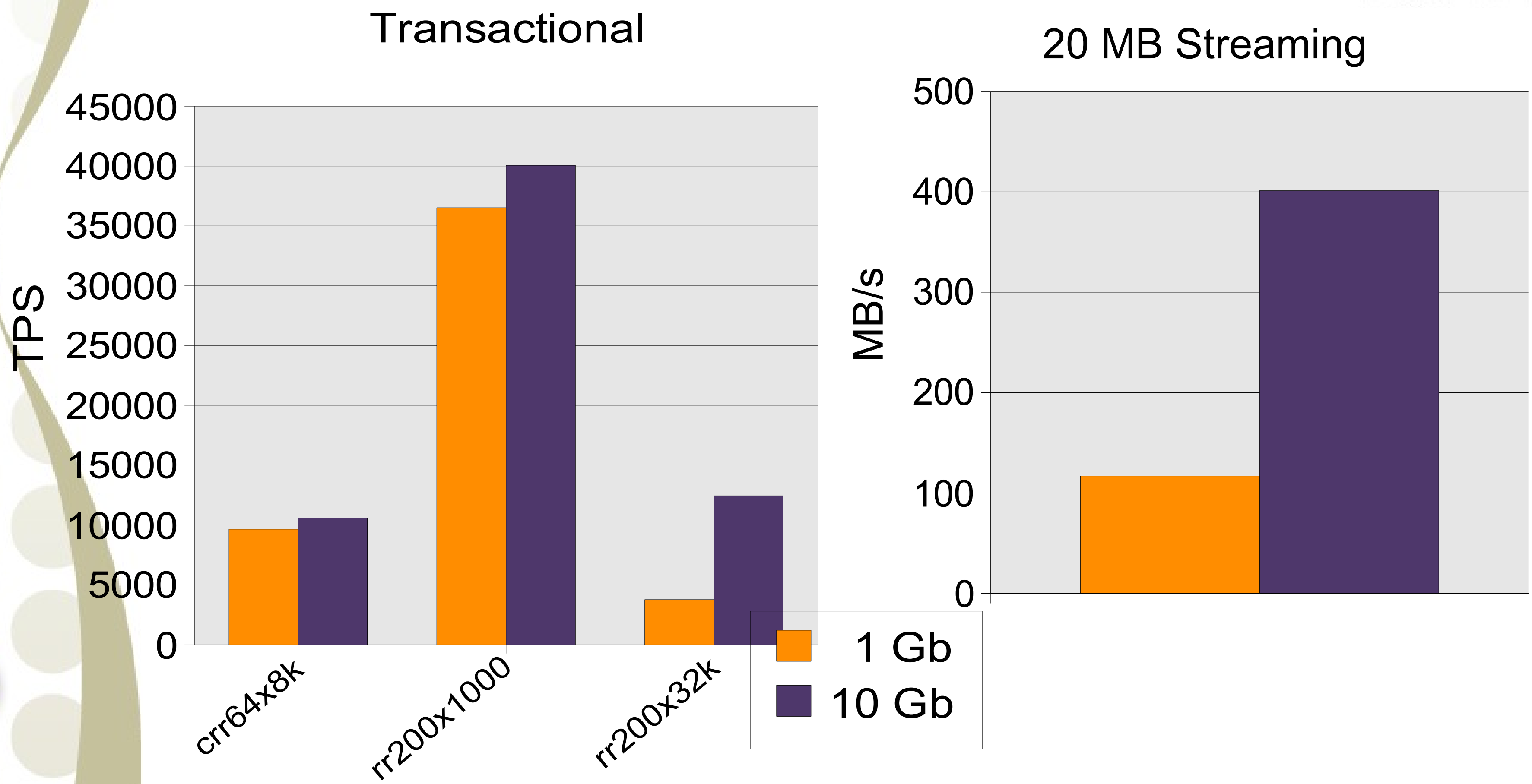


Response times



- More than 20% improvement with OSA-Express2

OSA-Express2, 1Gb / 10Gb, MTU 8992

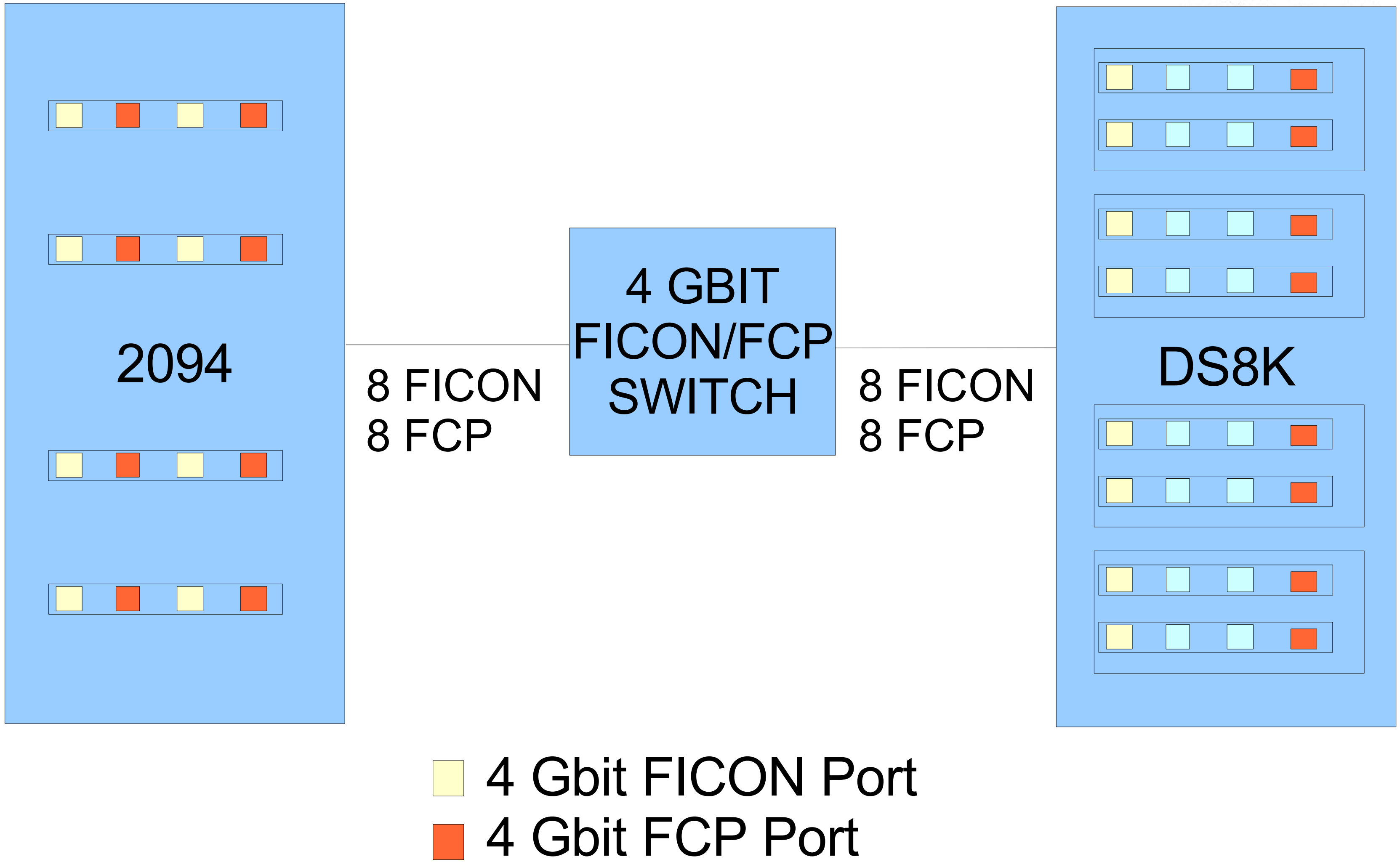


- Advantage for 10 Gb over 1 Gb is increasing with data size
- Improvements up to 3.4x

Disk I/O benchmark

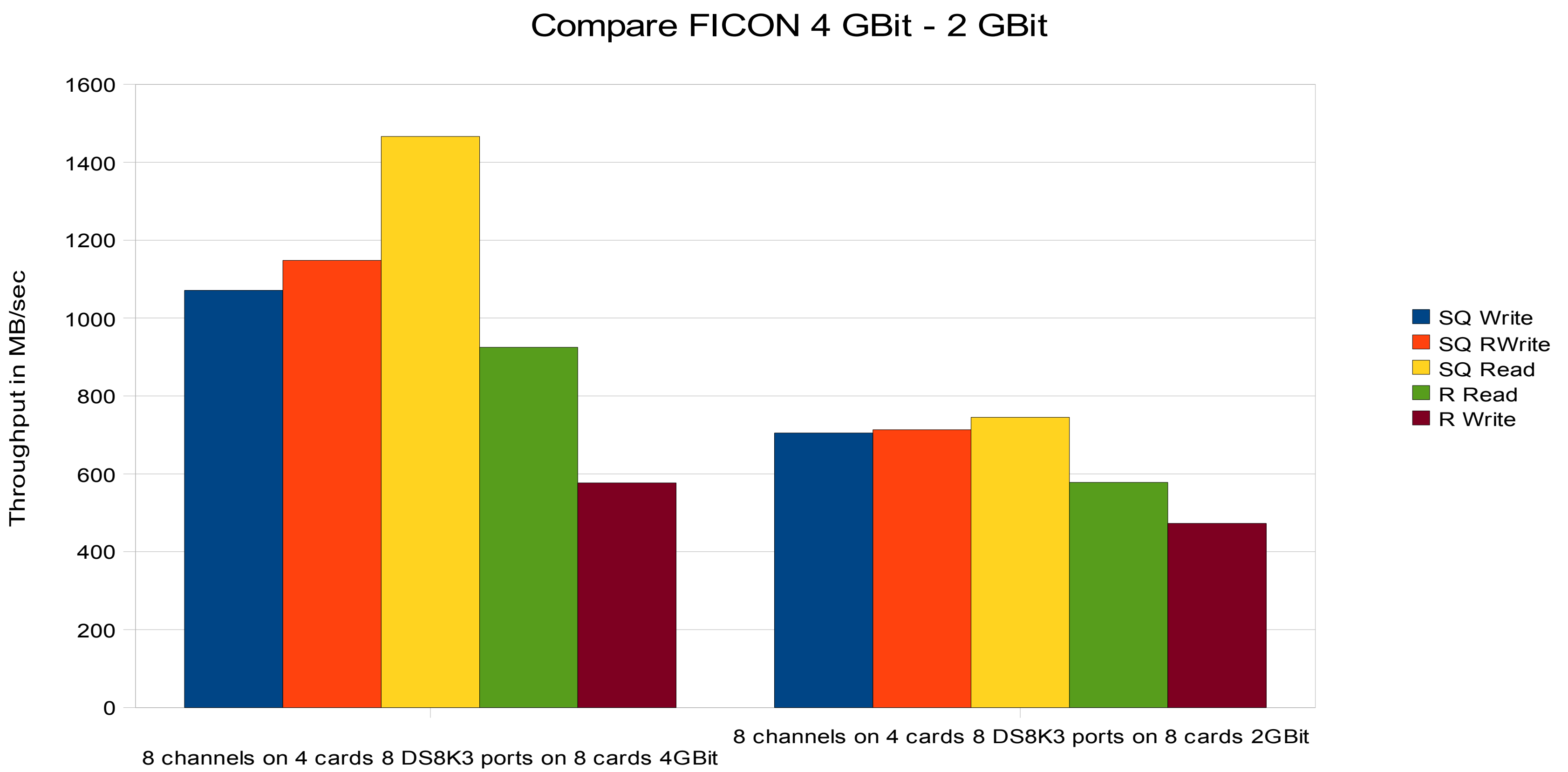
- IOzone
- Threaded file system benchmark used to measure synchronous I/O
- Sequential/random write, rewrite, read of a large enough file (700MB = almost 3x of memory size)
- Main memory was restricted to 256MB
- 1, 2, 4, 8, 16, 32, 64 threads, each operating on its private disk or using a Logical Volume
- Used on FICON and SCSI disks

Configuration for 4Gbps disk I/O measurements



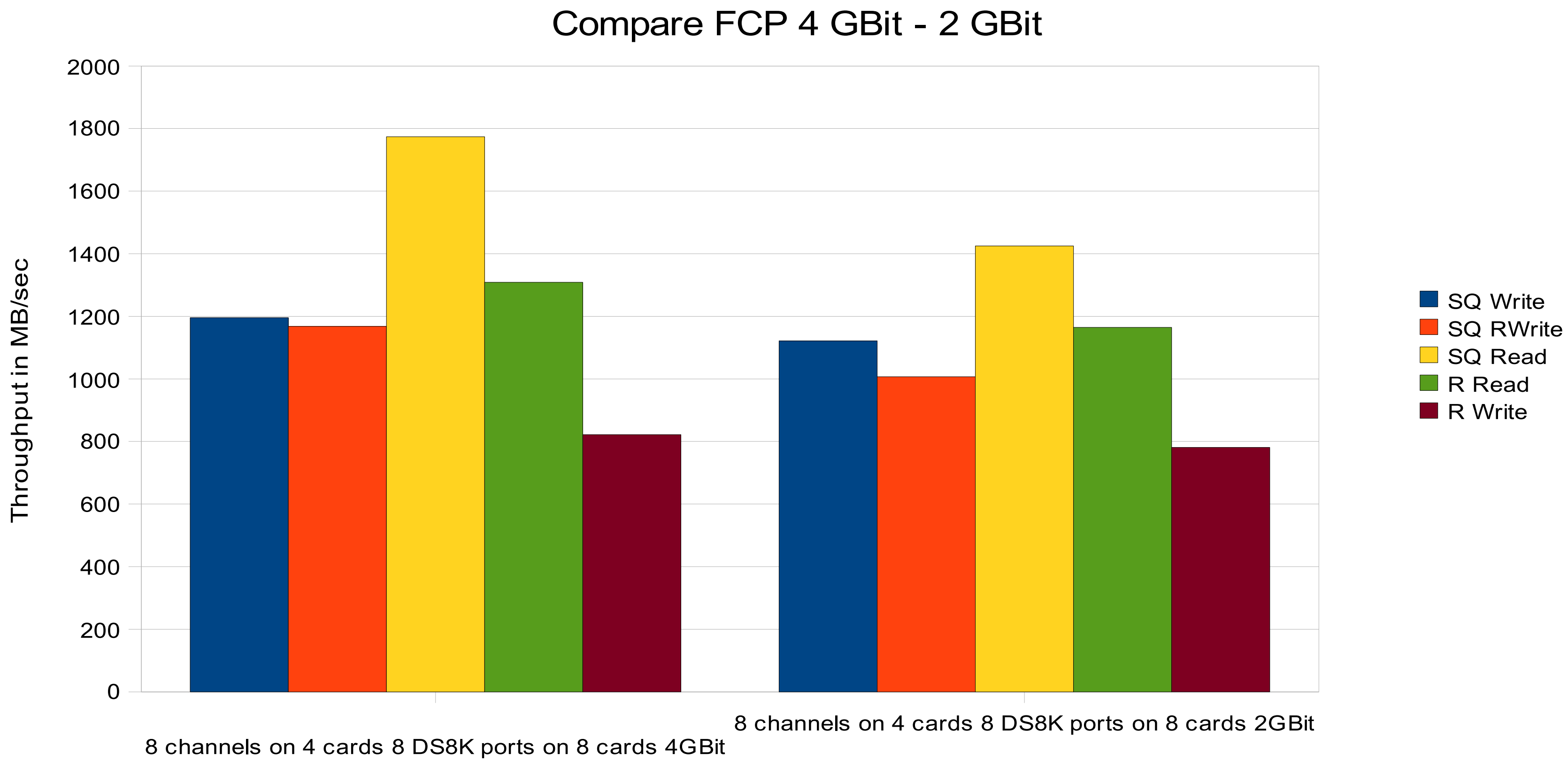
Disk I/O performance with 4Gbps links - FICON

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



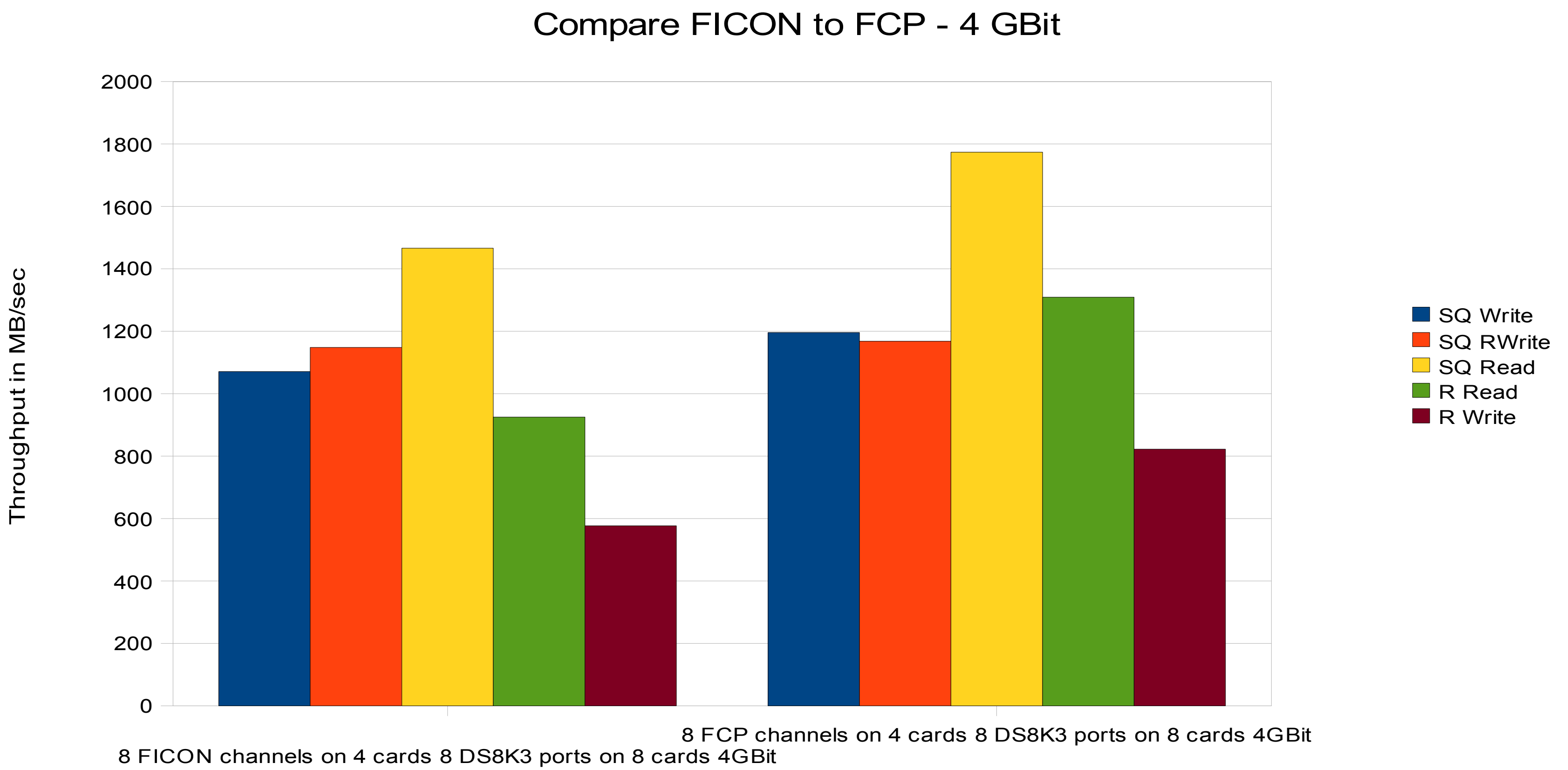
Disk I/O performance with 4Gbps links - FCP

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



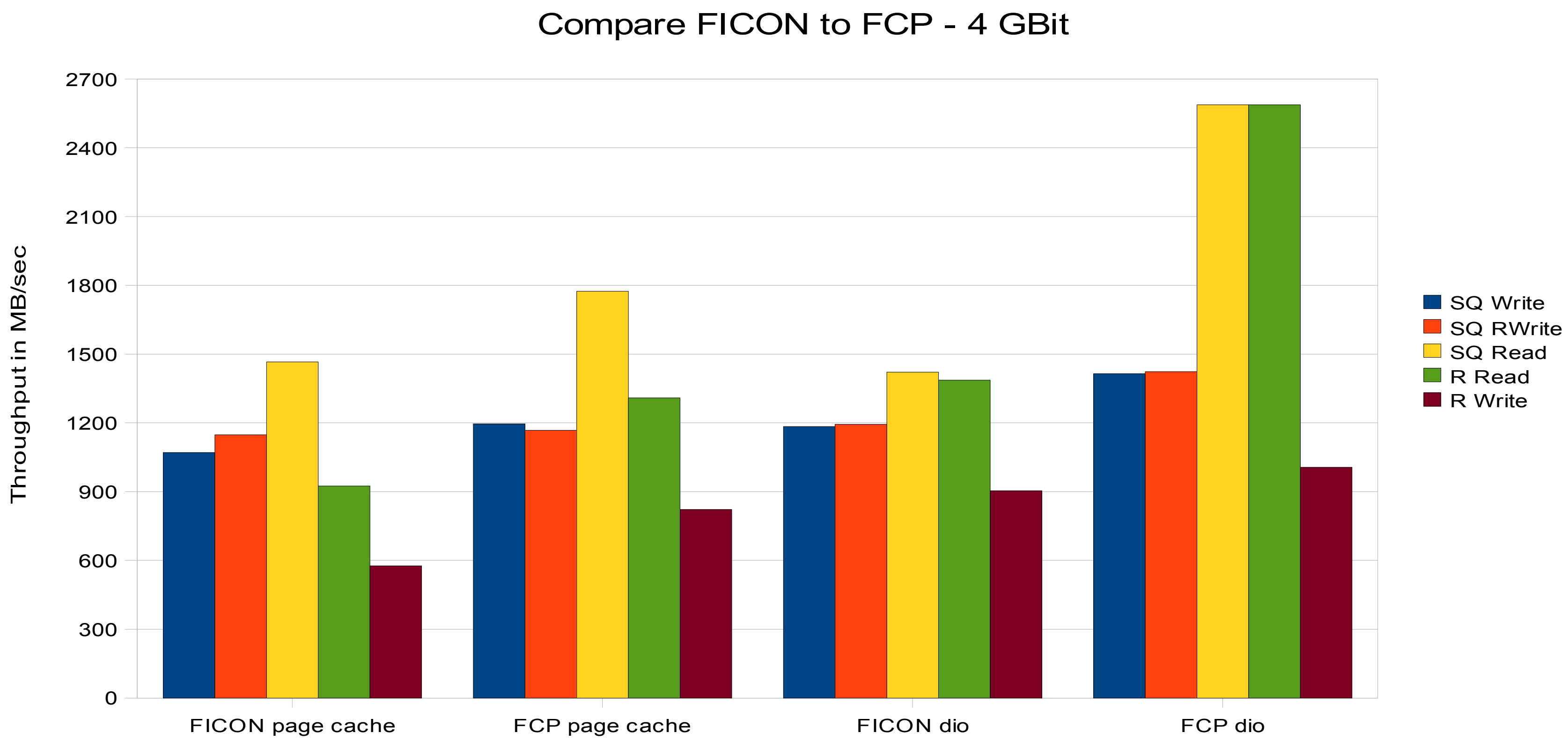
Disk I/O performance with 4Gbps links – FICON versus FCP

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



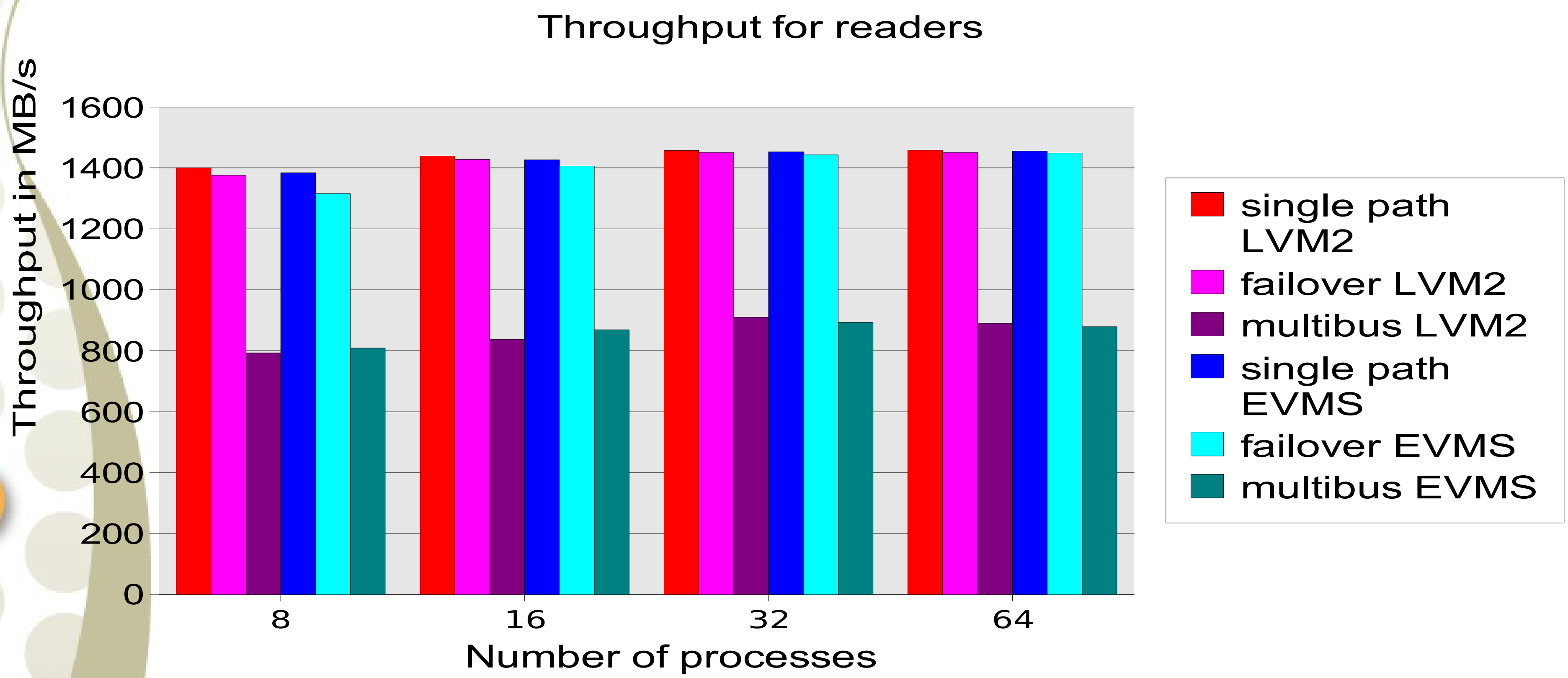
Disk I/O performance with 4Gbps links – FICON versus FCP / direct I/O

- Bypassing the Linux page cache improves throughput for FCP up to 2x, for FICON up to 1.6x.
- Read operations are much faster on FCP



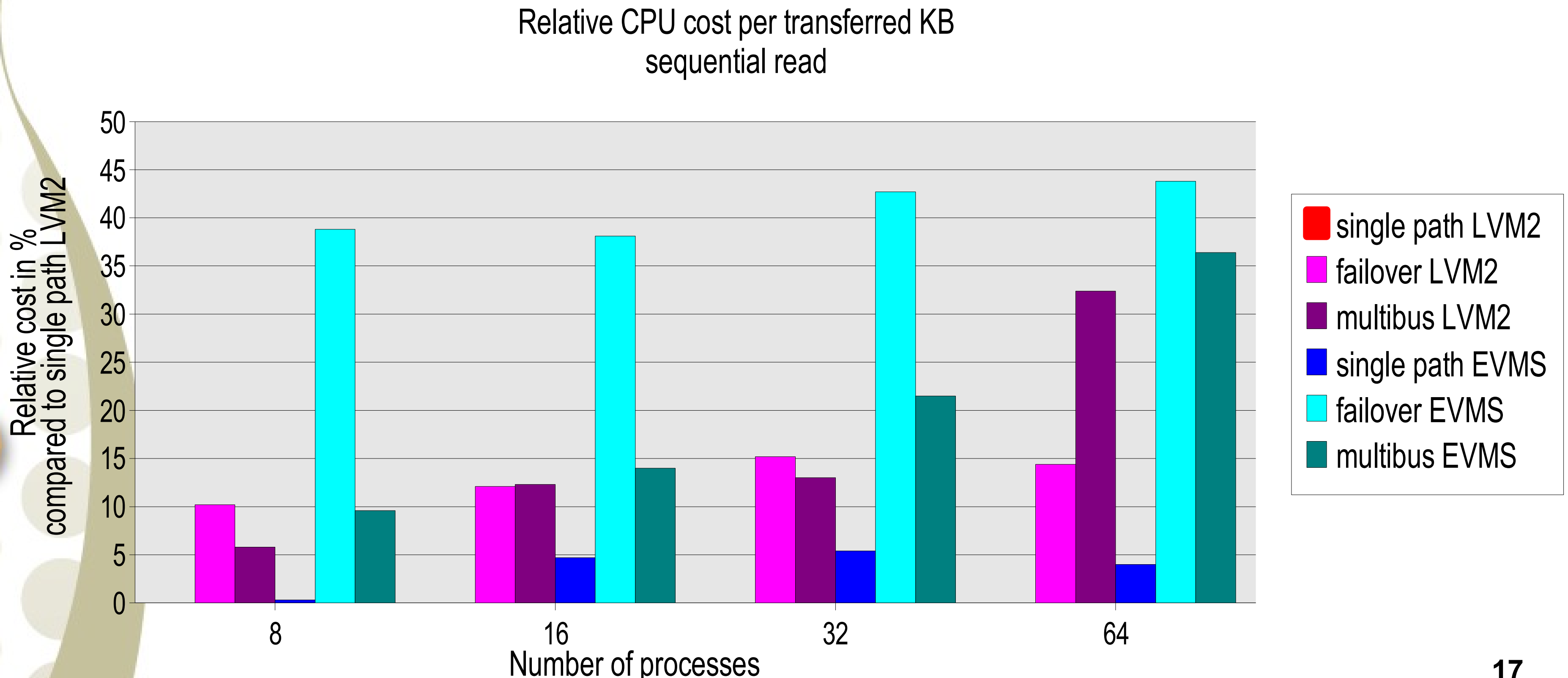
FCP/SCSI single path versus multipath

- Use failover instead of multibus



FCP/SCSI single path versus multipath (2)

- Use LVM2 instead of EVMS
- Costs for multipathing are about 10%

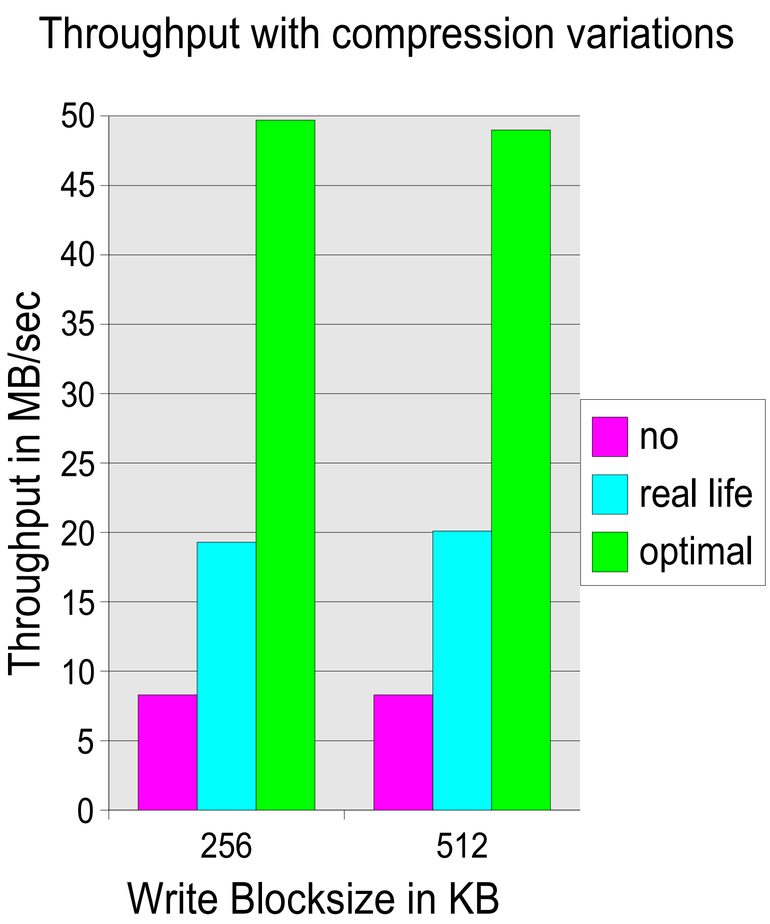


Disk I/O considerations

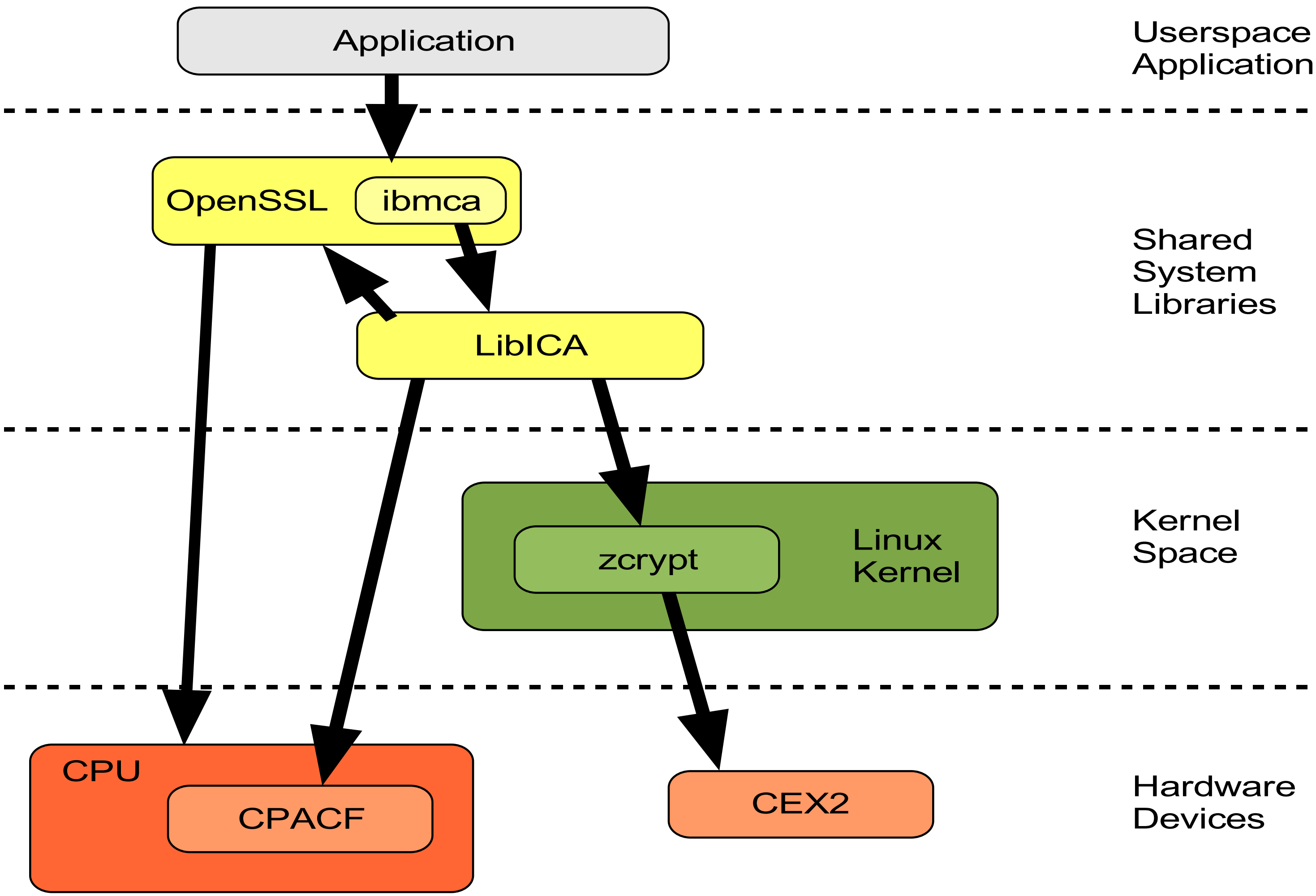
- Higher throughput rates with the new storage server generation require also higher CPU utilization
- This applies also to FCP/SCSI I/O when there is a throughput win versus FICON/ECKD I/O
- Take care that any specific path assignments for FCP/SCSI disks are still valid after re-IPL.
- `http://www.ibm.com/developerworks/linux/linux390/perf/tuning_how_dasd_multipath.html`

SCSI tape performance

- Measurements on IBM 3590 with optimal compression, compression of real life data (Linux source code), without compression
- Tests were done with dd, 1 FCP channel to the tape.
- Select a large blocksize for the tape, e.g. 256 KB

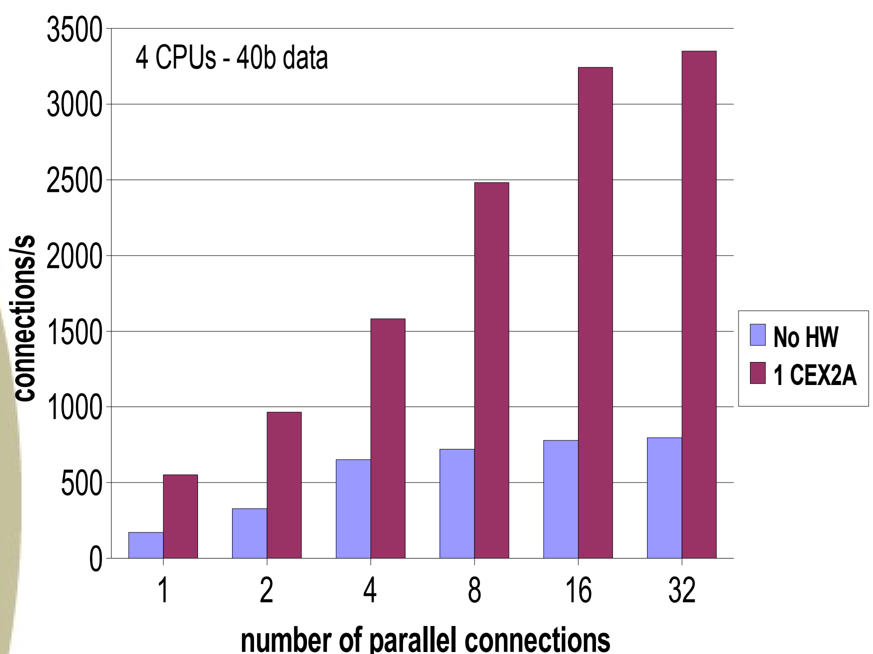


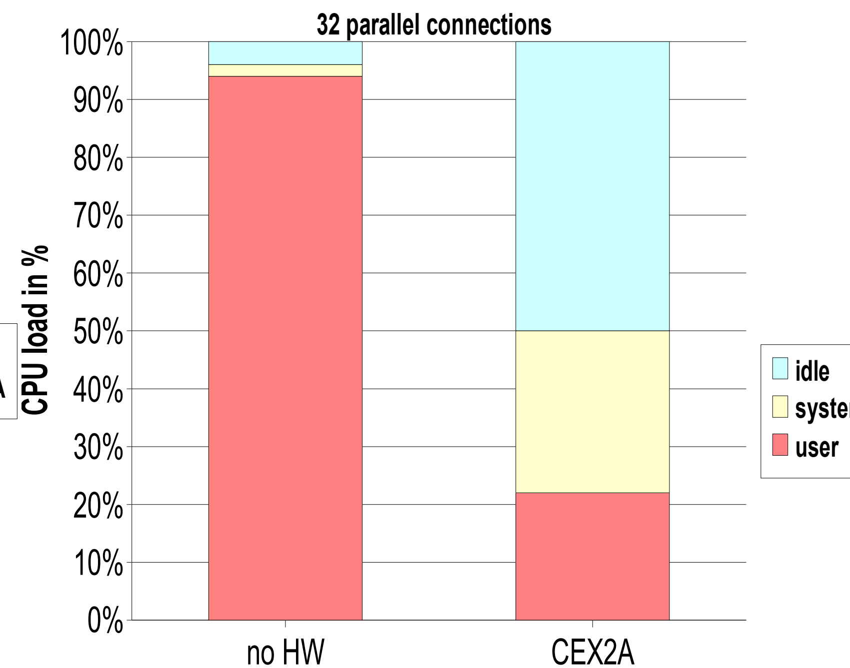
Linux software SSL stack



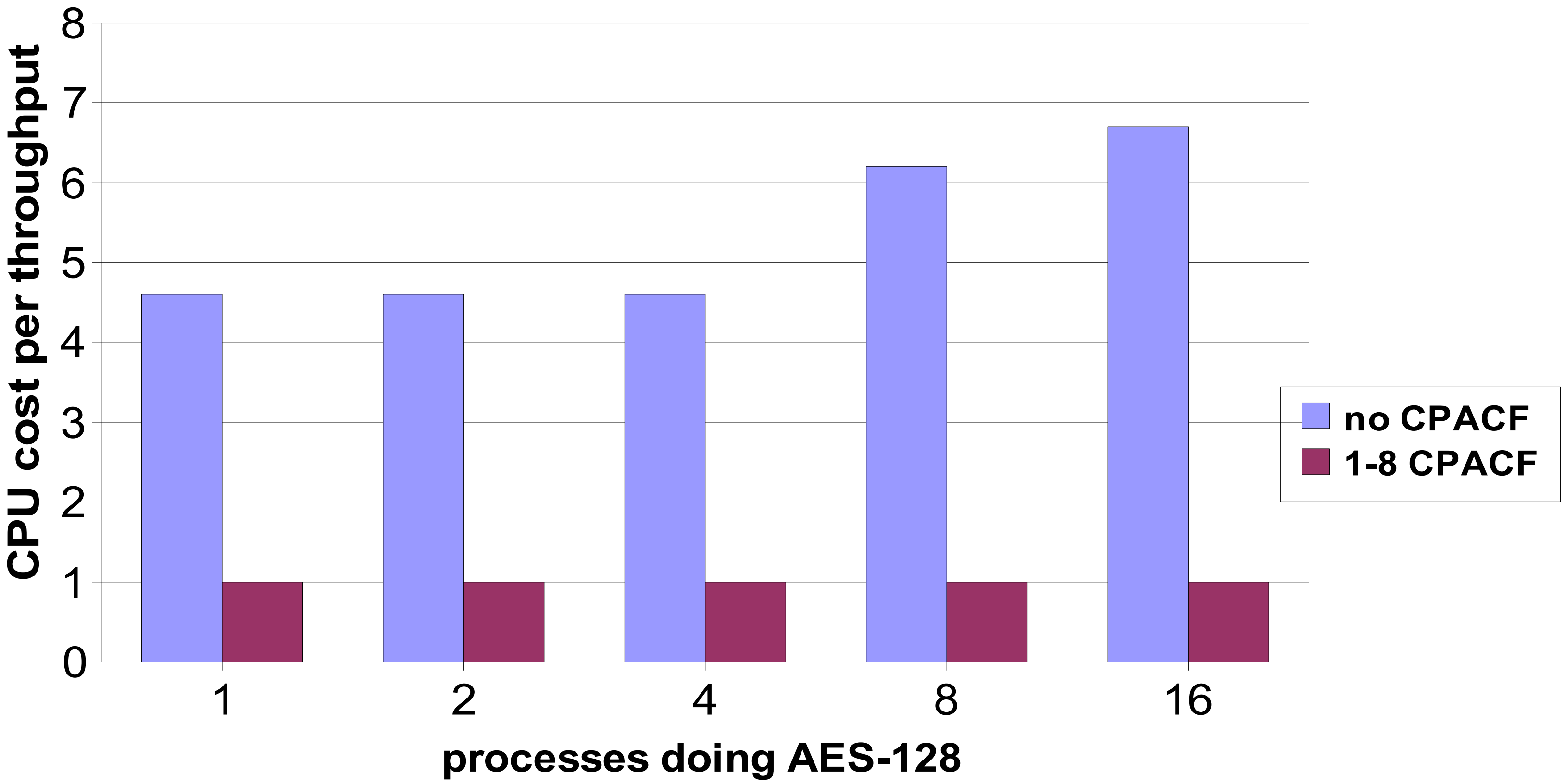
Crypto Express2 - SSL handshakes

- 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



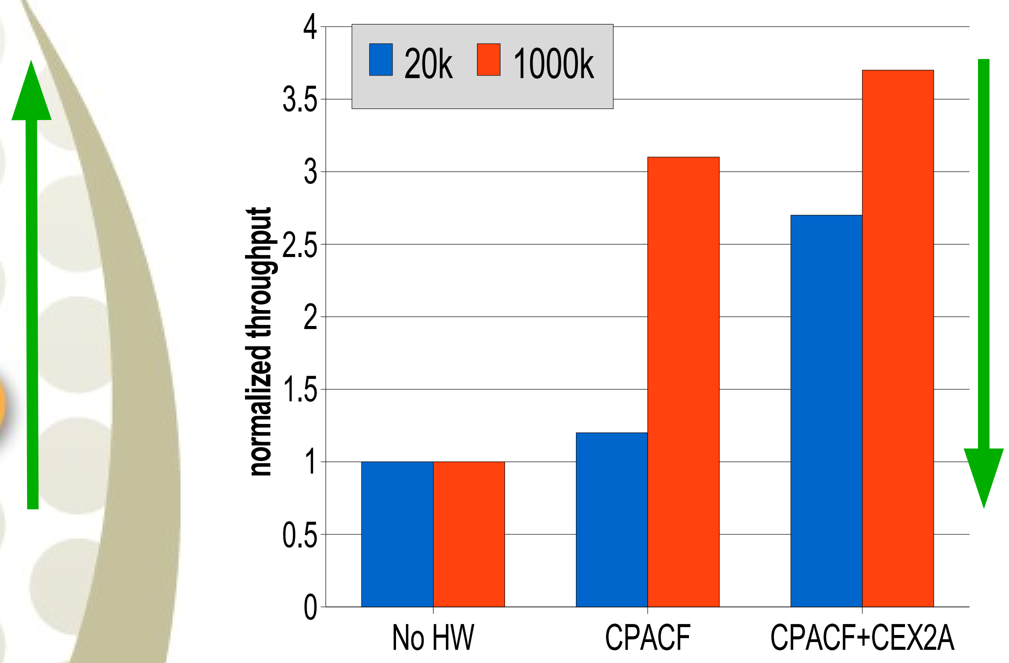


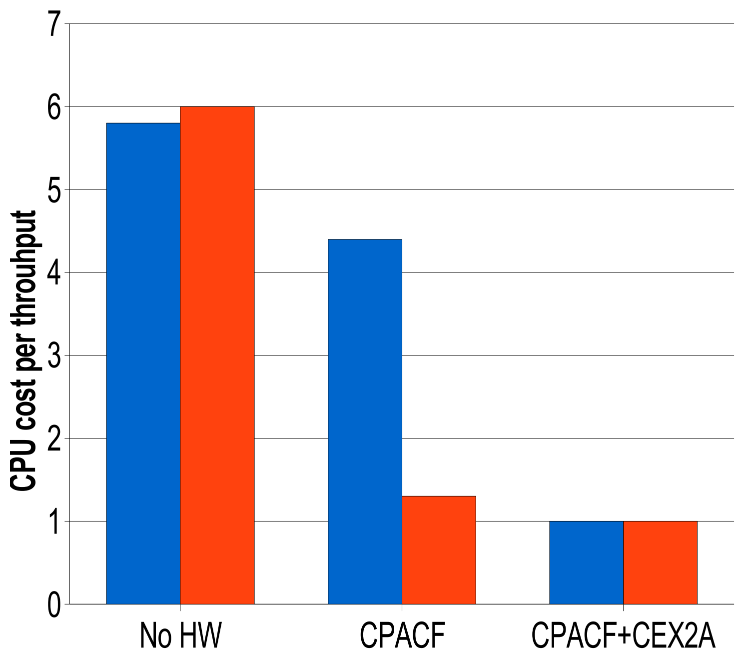
System z9 CPACF feature



Crypto Express2 – CPACF and CEX2

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





Agenda

- System z hardware
- Hardware improvements
  - Processor
  - Networking
  - Disk / Tape
  - Cryptography
- **Software improvements**
  - **Compiler**
  - **Java**
  - **WebSEAL**
  - **Tivoli Storage Manager**
- Distribution improvements
  - Red Hat
  - Novell SUSE

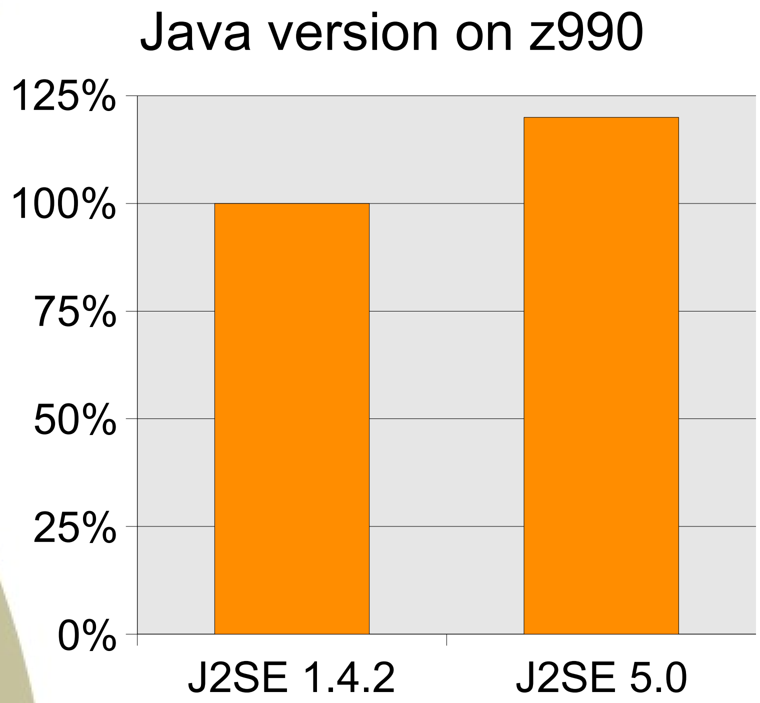
gcc 64bit compiler – SLES9 (gcc-3.3.3) versus SLES10 (gcc-4.1.0)

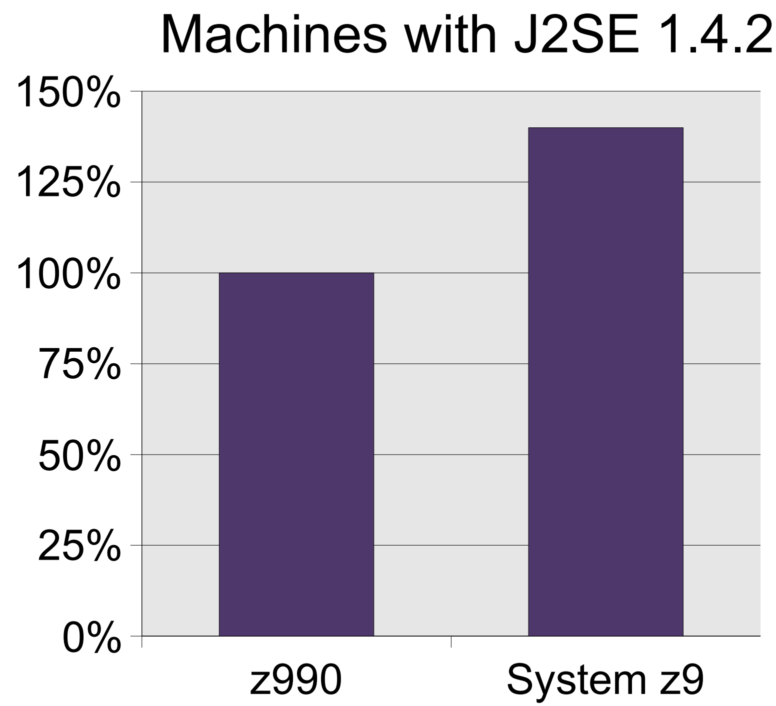
- gcc 4.1 supports -mtune=z9-109 and -march=z9-109

Compiler - why isn't 64-bit for free?

- Hardware effects
  - Primarily D-cache "pressure"
  - z/Architecture supports both 31-bit and 64-bit addressability
    - Data cache is fixed size for machine
    - 64-bit pointers "twice" as large as 31-bit pointers
  - Also impacts I-cache performance
    - 64-bit instructions tend to be 6-byte instead of 2 or 4
- Software effects
  - some 31-bit instructions have no 64-bit equivalent
    - must be implemented with series of 64-bit opcodes
    - = additional pathlength for same function
  - increased cost for entry/exit linkage
    - registers are twice as wide

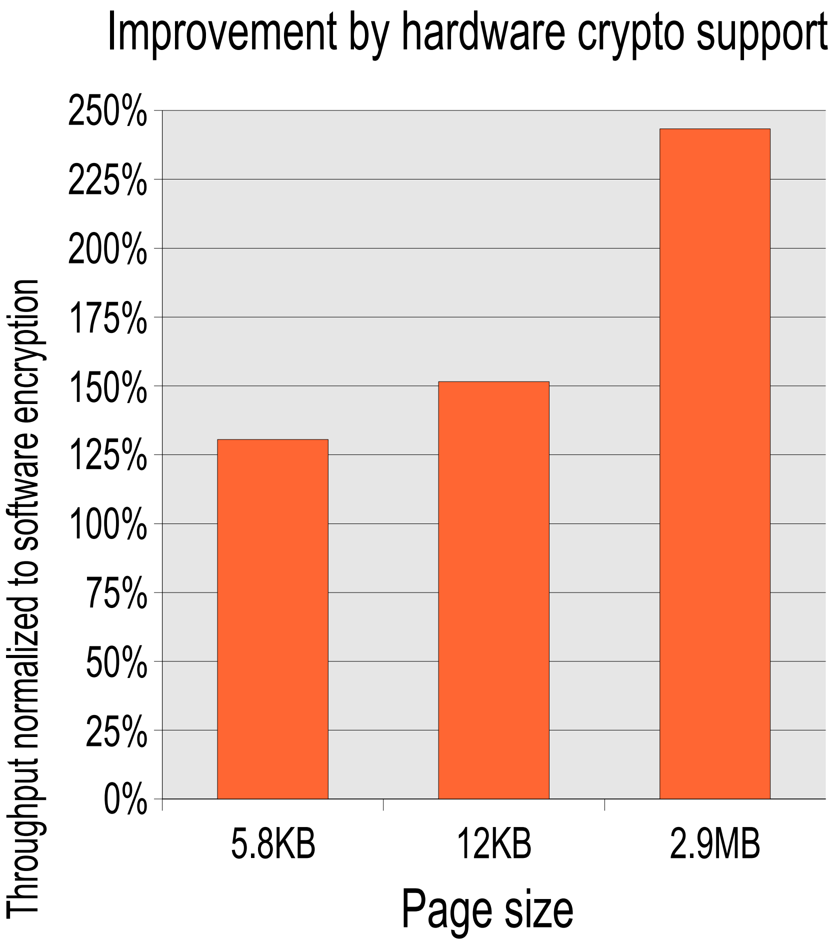
Java Results 64-bit





- Improvements through Java (JVM and JIT)
- Improvements through new hardware
- 64-bit Java is production ready

Crypto performance – WebSEAL SSL access



- The connection from the client to the WebSEAL server runs encrypted using SSL (AES-128)
- Scaling the size of the requested page
- uses mostly CPACF
- Improvement up to factor 2.4 for hardware encryption versus software encryption

Special study with Tivoli Storage Manager

- ECKD versus SCSI
- Configured and measured on our system together with TSM performance specialist
- Entry statement from TSM, based on their tests in 2005 for backing up 70 GB data:
  - *“execution time with SCSI is 25% shorter than with ECKD”*
  - *“average CPU consumption with SCSI is 67% more than with ECKD”*
- Common exit statement from after the tests:
  - *“execution time with SCSI is 50% shorter than with ECKD”*
  - *“costs were almost equal (more CPU resources need to be provided for SCSI)”*

Agenda

- System z hardware
- Hardware improvements
  - Processor
  - Networking
  - Disk / Tape
  - Cryptography
- Software improvements
  - Compiler
  - Java
  - WebSEAL
  - Tivoli Storage Manager
- **Distribution improvements**
  - **Red Hat**
  - **Novell SUSE**

Comparison SLES10 / RHEL5

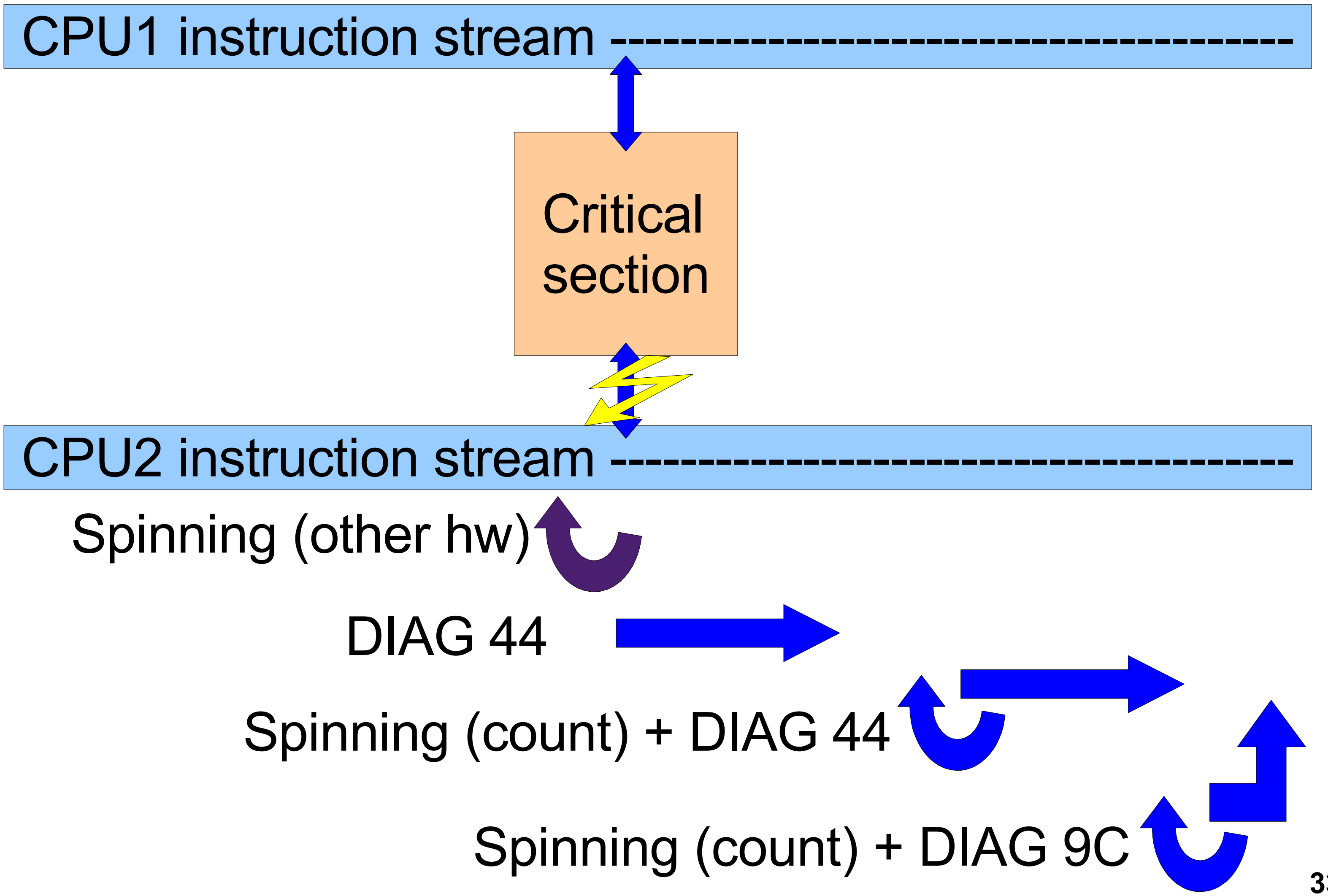
measurement portfolio SLES10 GA versus RHEL5 GA	LPAR 64	LPAR 31 (emu)	z/VM 64	z/VM 31 (emu)
Scaling	equal	n/a	n/a	n/a
Mixed I/O ECKD	worse	n/a	equal	n/a
Mixed I/O SCSI	worse	n/a	worse	n/a
Kernel	equal	n/a	better	better
Compiler INT	equal	n/a	n/a	n/a
Compiler FP	worse	n/a	n/a	n/a
Seq. I/O ECKD	better	n/a	better	n/a
Seq. I/O SCSI	better	n/a	better	n/a
Rnd I/O ECKD	equal	n/a	equal	n/a
Rnd I/O SCSI	equal	n/a	equal	n/a
Network 1000Base-T QDIO	equal	n/a	worse	n/a
Network 1GbE QDIO	equal	n/a	worse	n/a
Network 10GbE QDIO	equal	n/a	worse	n/a
Network HiperSockets	equal	n/a	n/a	n/a
Java	equal	equal	n/a	n/a

Legend	*n/a*	*better*	*equal*	*worse*

SLES9 improved resource usage

- The Linux kernel uses spin locks to wait for exclusive use of kernel resources
- On System z it is not desirable to use processors for active waiting
- The old solution was to issue a DIAG 44 to the LPAR hypervisor or to z/VM to give the CPU back instead of looping on the lock, to allow other more useful work to be done.
- 2 new features:
  - spin_retry counter in Linux to avoid excessive use of diagnose instructions
  - use of DIAG 9C to pass information along with the instruction, who should get the processor

Avoiding spin locks on System z



SLES10 virtual CPU time accounting

- The standard Linux implementation is based on a heuristic model with a 10 ms timer interrupt
  - The complete time slice is accounted to the interrupted context
- On systems with virtual CPUs this approach is too inaccurate
- The new implementation is based on the System z virtual timer
  - CPU times get now accounted whenever the execution context changes
  - A new category of Linux wait state is showing, how often the Linux system is waiting for CPU (current sysstat version required)
  - The feature is enabled by default in SLES10 and RHEL5

Linux command 'top' – the snapshot tool

- Adds new field “CPU steal time”
  - Is time Linux wanted to run, but the hipervisor 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
```

Visit us !

- Linux on zSeries Tuning Hints and Tips
  `http://www.ibm.com/developerworks/linux/linux390/perf/index.html`
- Linux-z/VM Performance Website
  `http://www.vm.ibm.com/perf/tips/linuxper.html`

Questions

SHARE
Technology · Connections · Results