Linux on System z performance update

Speaker Names: Dr. Eberhard Pasch

Speakers Company: IBM

Date of Presentation: **Tuesday, February 5, 2013 (11:00am)**

Yosemite C, Ballroom Level

Session Number: 13102

http://linuxmain.blogspot.com/

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:

*BladeCenter®, DB2®, e business(logo)®, DataPower®, ESCON, eServer, FICON, IBM®, IBM (logo)®, MVS, OS/390®, POWER6®, POWER6+, POWER7®, Power Architecture®, PowerVM®, S/390®, System p®, System p5, System x®, System z®, System z9®, System z10®, WebSphere®, X-Architecture®, zEnterprise, z9®, z10, z/Architecture®, z/OS®, z/VM®, z/VSE®, zSeries®

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 registered 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.

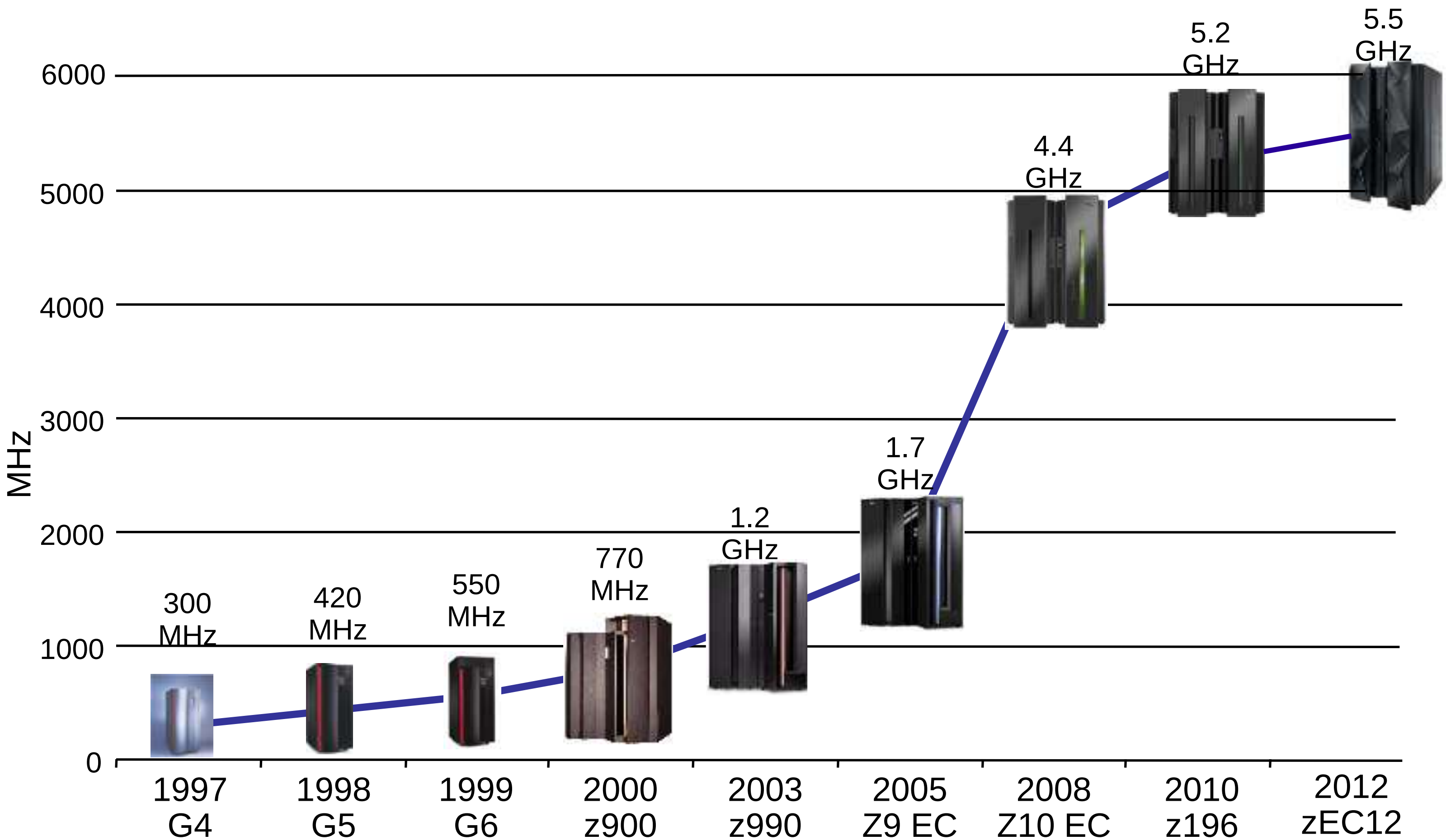
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products 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.

Agenda

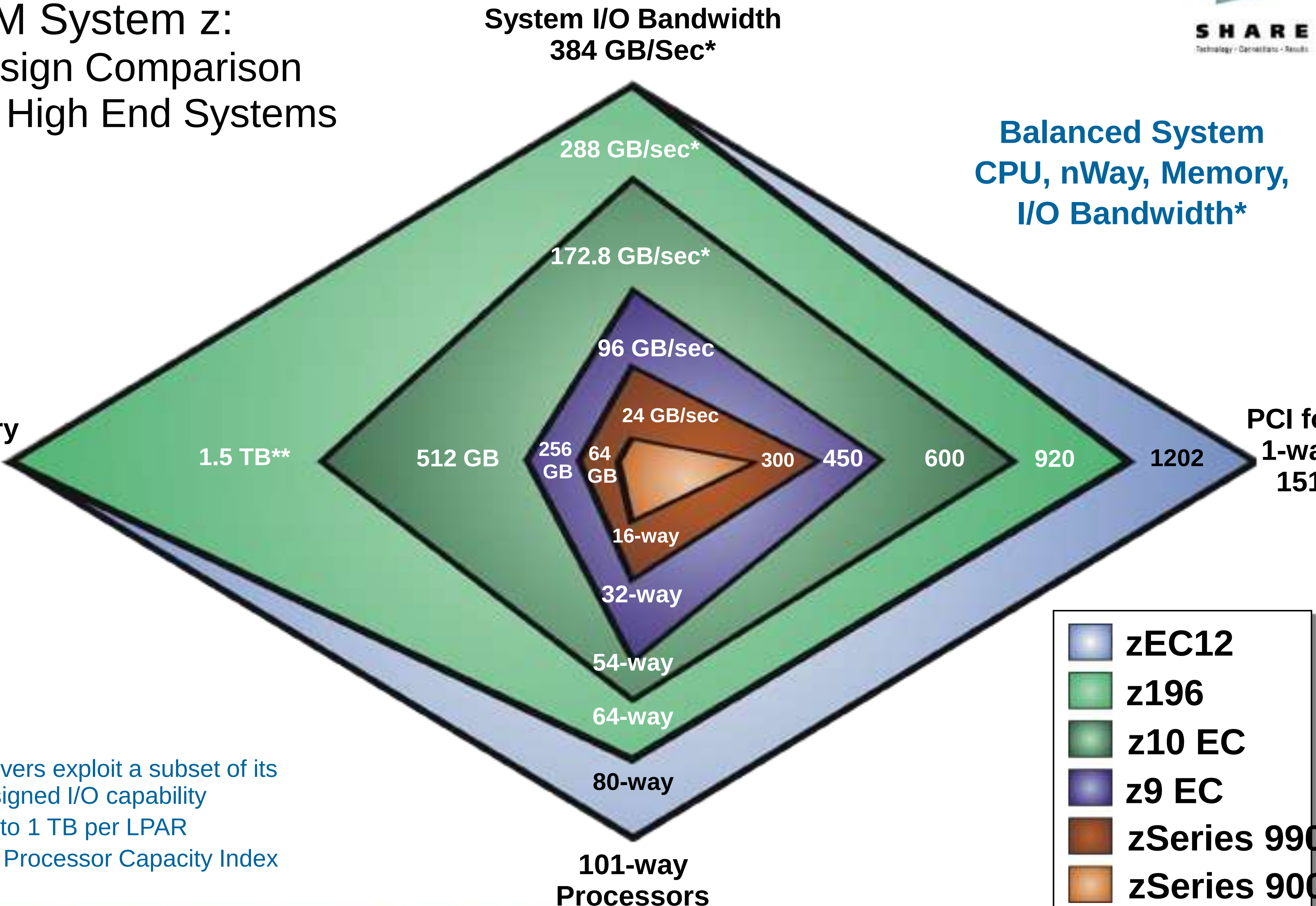
- zEC12 – hardware design
- zEC12 – performance comparison with z196
- OSA Express4S and FICON Express 8S results
- Miscellaneous
  - RHEL 5.9
  - Large Pages
  - Java / WAS

zEC12 Continues the Mainframe Heritage



IBM System z:
Design Comparison for High End Systems

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



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

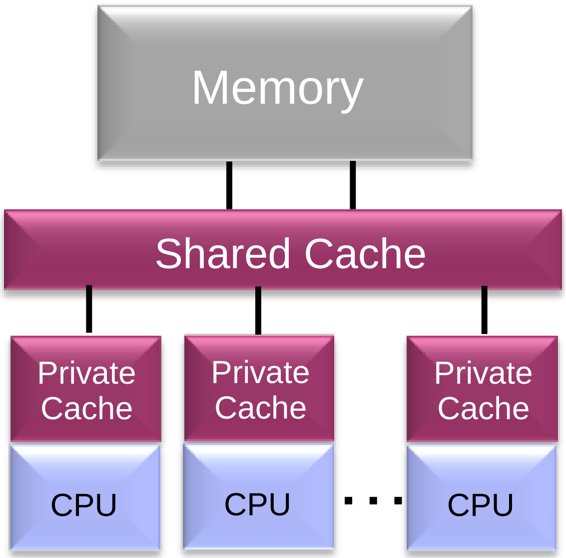
** Up to 1 TB per LPAR

PCI – Processor Capacity Index

Processor Design Basics

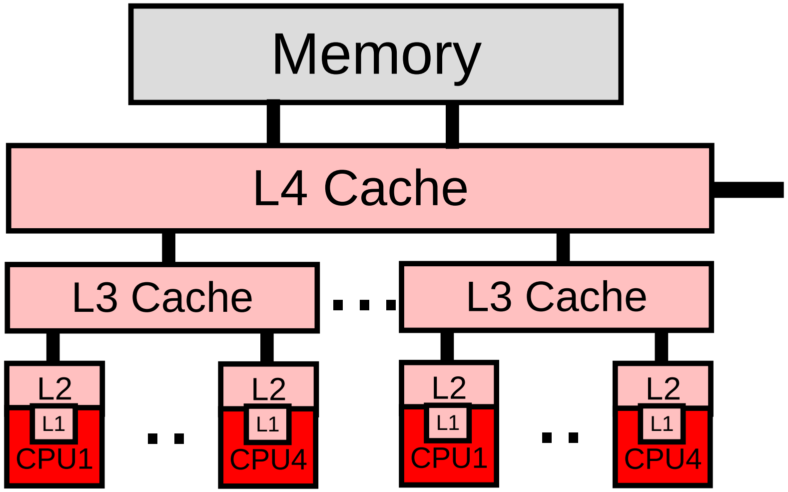
- CPU (core)
  - Cycle time
  - Pipeline, execution order
  - Branch prediction
  - Hardware versus millicode
- Memory subsystem
  - High speed buffers (caches)
    - On chip, on book
    - Private, shared
    - Coherency required
  - Buses
    - Number, Bandwidth
  - Limits
    - Distance + speed of light, space

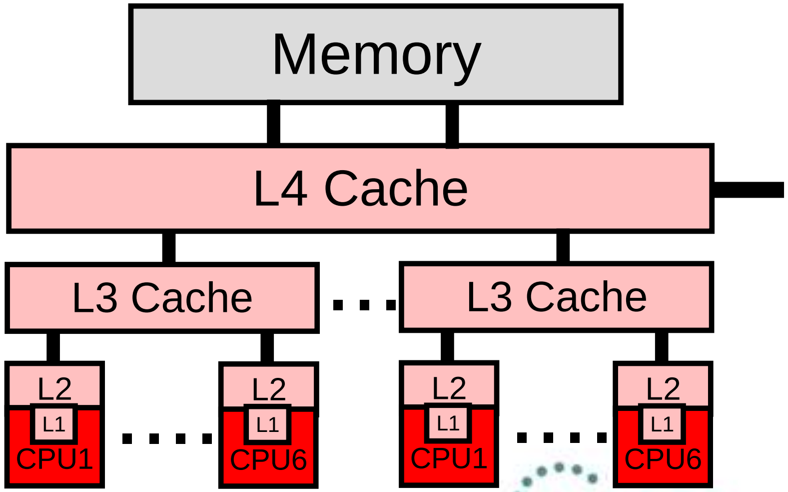
Generic Hierarchy example



zEC12 versus z196 hardware comparison

- z196
  - CPU
    - 5.2 GHz
    - Out-Of-Order execution
  - Caches
    - L1 private 64k I, 128k D
    - L2 private 1.5 MB
    - L3 shared 24 MB / chip
    - L4 shared 192 MB / book
- zEC12
  - CPU
    - 5.5 GHz
    - Enhanced Out-Of-Order
  - Caches
    - L1 private 64k I, 96k D
    - L2 private 1 MB I + 1 MB D
    - L3 shared 48 MB / chip
    - L4 shared 384 MB / book

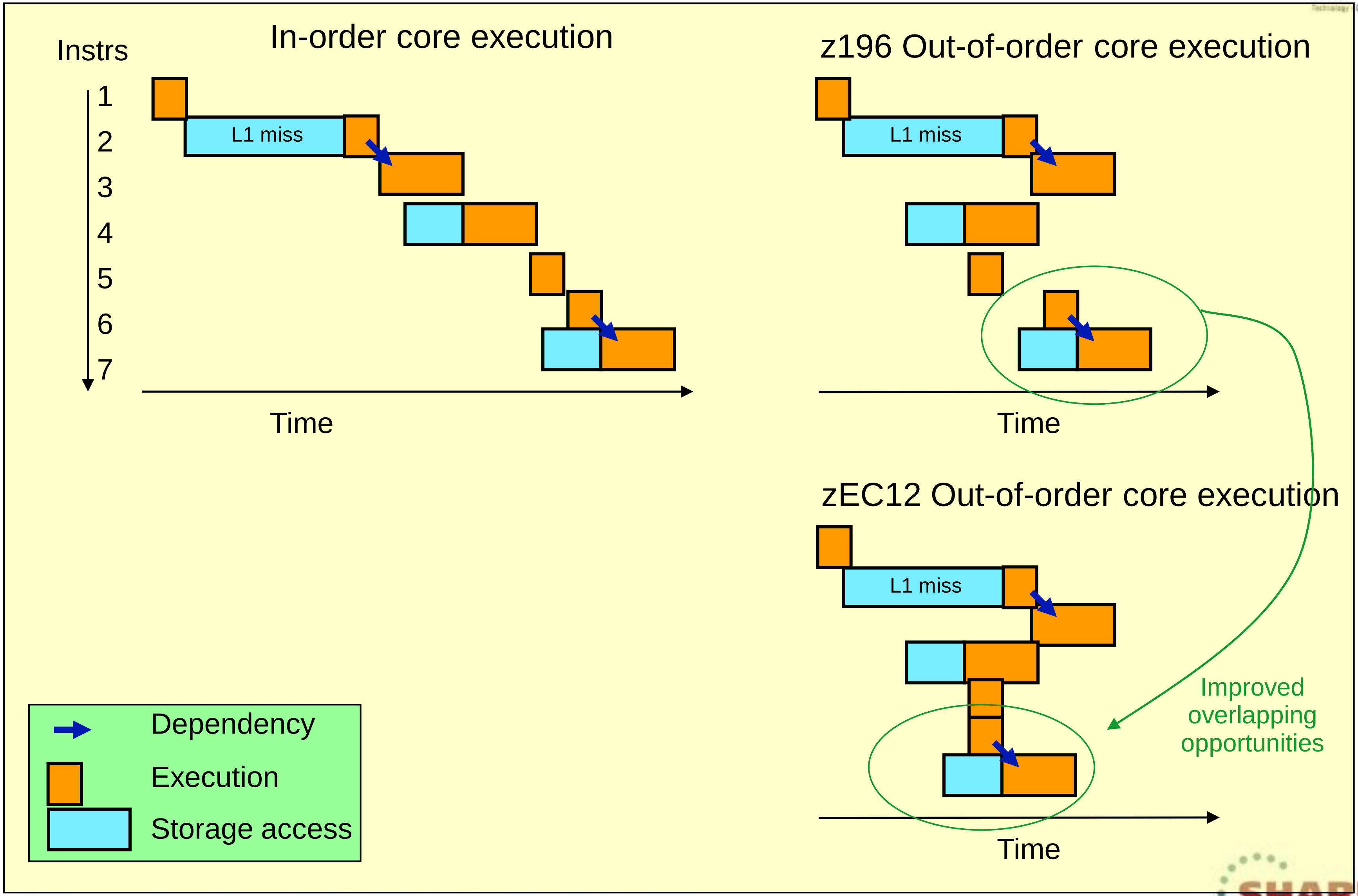




zEC12 Out of Order – why?

- Out of order yields significant performance benefit through
  - Re-ordering instruction execution
    - Instructions stall in a pipeline because they are waiting for results from a previous instruction or the execution resource they require is busy
    - In an in-order core, this stalled instruction stalls all later instructions in the code stream
    - In an out-of-order core, later instructions are allowed to execute ahead of the stalled instruction
  - Re-ordering storage accesses
    - Instructions which access storage can stall because they are waiting on results needed to compute storage address
    - In an in-order core, later instructions are stalled
    - In an out-of-order core, later storage-accessing instructions which can compute their storage address are allowed to execute
  - Hiding storage access latency
    - Many instructions access data from storage
    - Storage accesses can miss the L1 and require 10 to 500 additional cycles to retrieve the storage data
    - In an in-order core, later instructions in the code stream are stalled
    - In an out-of-order core, later instructions which are not dependent on this storage data are allowed to execute

Out of Order Execution – z196 vs zEC12



Agenda

- zEC12 – hardware design
- zEC12 – performance comparison with z196
- OSA Express4S and FICON Express 8S results
- Miscellaneous
  - RHEL 5.9
  - Large Pages
  - Java / WAS

zEC12 vs z196 comparison Environment

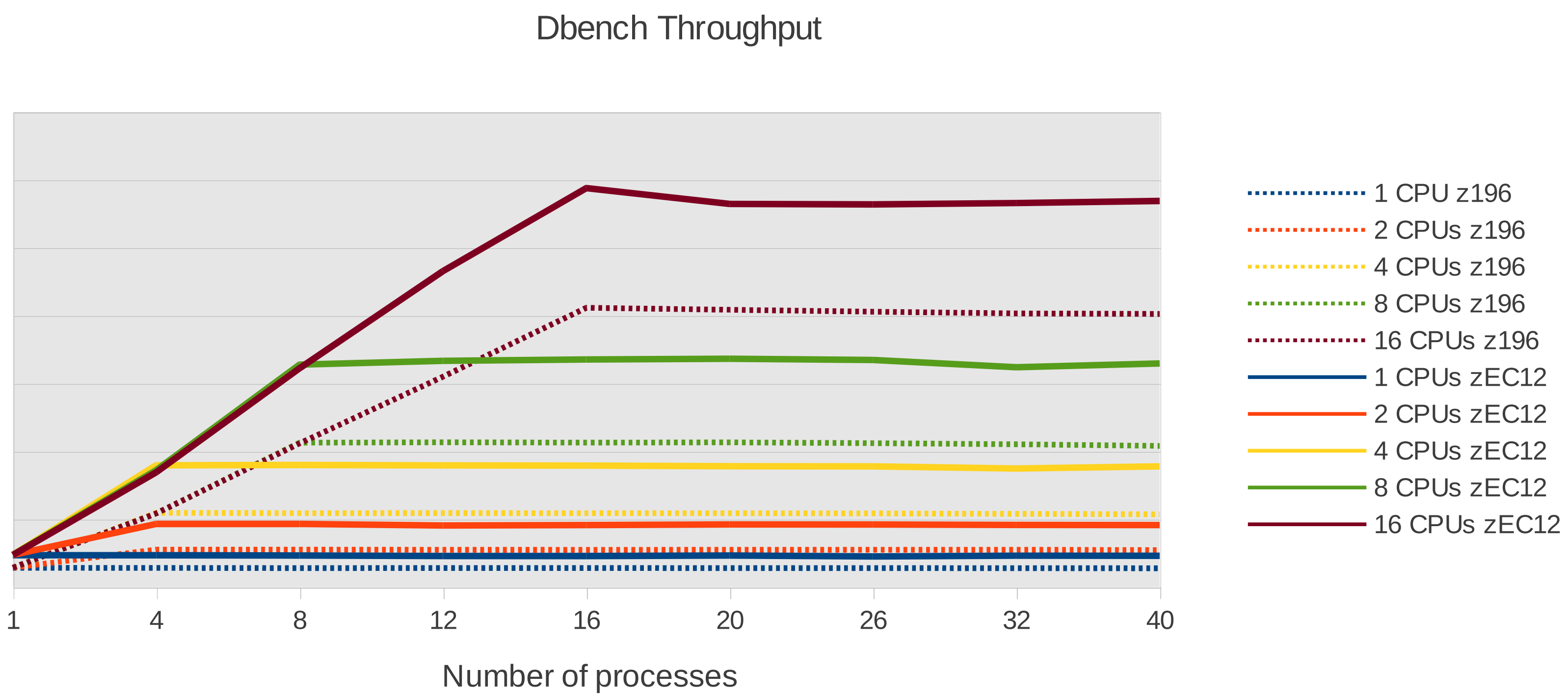
- Hardware
  - z12EC : 2827-708 H66 with **pre-GA microcode, pre-GA hardware**
  - z196 : 2817-766 M66
  - (z10 : 2097-726 E26)
- Linux distribution with recent kernel
  - SLES11 SP2: 3.0.13-0.27-default
  - Linux in LPAR
  - Shared processors
  - Other LPARs deactivated

File server benchmark description

- Dbench 3
  - Emulation of Netbench benchmark
  - Generates file system load on the Linux VFS
  - Does the same I/O calls like the smbd server in Samba (without networking calls)
  - Mixed file operations workload for each process: create, write, read, append, delete
  - Measures throughput of transferred data
- Configuration
  - 2 GiB memory, mainly memory operations
  - Scaling processors 1, 2, 4, 8, 16
  - For each processor configuration scaling processes 1, 4, 8, 12, 16, 20, 26, 32, 40

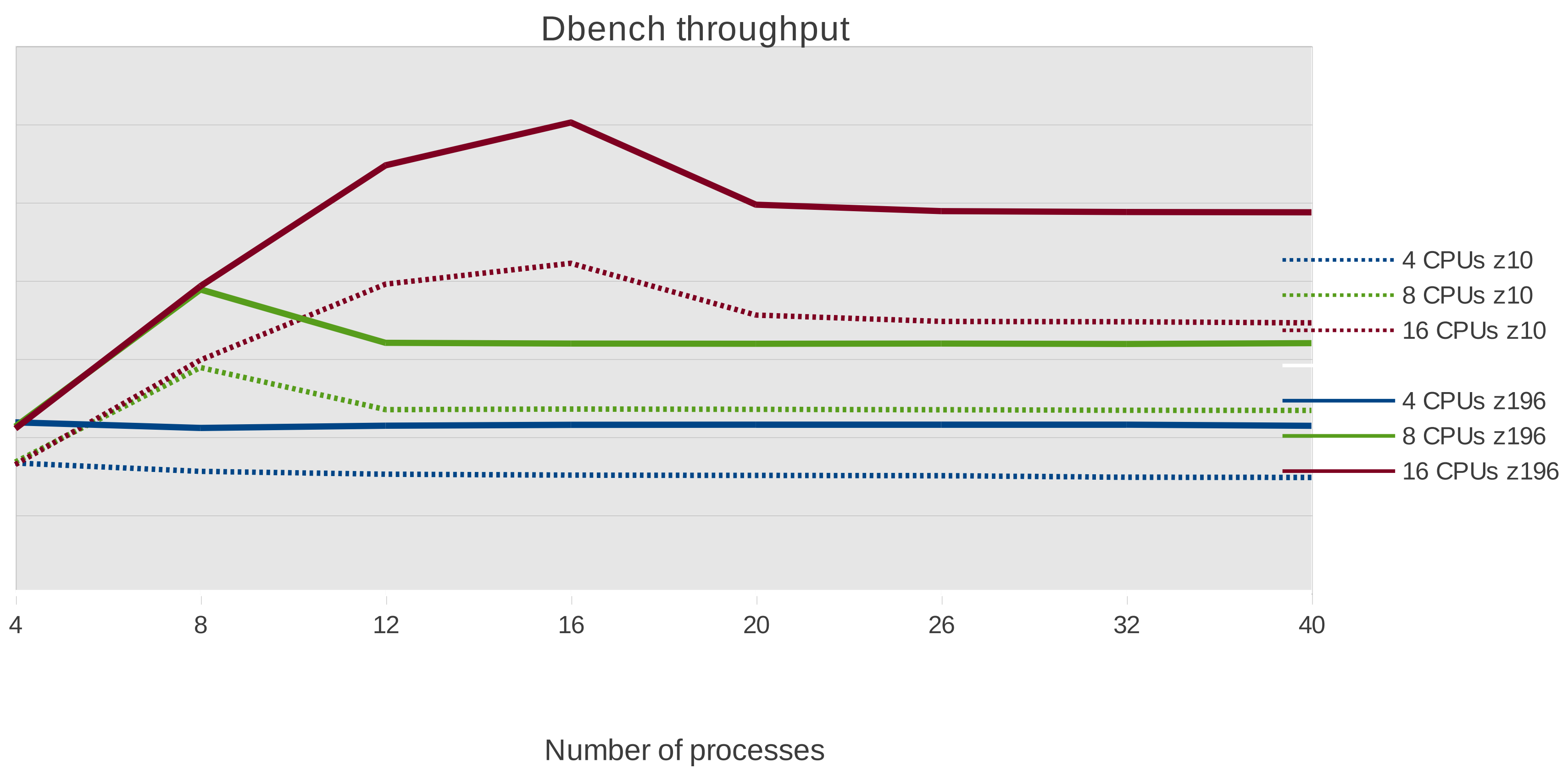
Dbench3

Throughput improves by 38 to 68 percent in this scaling experiment comparing zEC12 to z196



Dbench3

Throughput improves by 40 percent in this scaling experiment comparing z196 to z10



Kernel benchmark description

Lmbench 3

- Suite of operating system micro-benchmarks
- Focuses on interactions between the operating system and the hardware architecture
- Latency measurements for process handling and communication
- Latency measurements for basic system calls
- Bandwidth measurements for memory and file access, operations and movement
- Configuration
  - 2 GB memory
  - 4 processors

Lmbench3

Benefits seen in the very most operations, average at 45%

Measured operation	Deviation zEC12 to z196 in %
simple syscall	52
simple read/write	46 /43
select of file descriptors	32
signal handler	55
process fork	25
libc bcopy aligned L1 / L2 / L3 / L4 cache / main memory	0 / 12 / 25 / 10 / n/a
libc bcopy unaligned L1 / L2 / L3 / L4 cache / main memory	0 / 26 / 25 / 35 / n/a
memory bzero L1 / L2 / L3 / L4 cache / main memory	40 / 13 / 20 / 45 / n/a
memory partial read L1 / L2 / L3 / L4 cache / main memory	-10 / 25 / 45 / 105 / n/a
memory partial read/write L1 / L2 / L3 / L4 cache / main memory	75 / 75 / 90 / 180 / n/a
memory partial write L1 / L2 / L3 / L4 cache / main memory	45 / 50 / 62 / 165 / n/a
memory read L1 / L2 / L3 / L4 cache / main memory	5 / 10 / 45 / 120 / n/a
memory write L1 / L2 / L3 / L4 cache / main memory	80 / 92 / 120 / 250 / n/a
Mmap read L1 / L2 / L3 / L4 cache / main memory	0 / 13 / 35 / 110 / n/a
Mmap read open2close L1 / L2 / L3 / L4 cache / main memory	23 / 18 / 19 / 55 / n/a
Read L1 / L2 / L3 / L4 cache / main memory	60 / 30 / 35 / 50 / n/a
Read open2close L1 / L2 / L3 / L4 cache / main memory	27 / 30 / 35 / 60 / n/a
Unrolled bcopy unaligned L1 / L2 / L3 / L4 cache / main memory	35 / 28 / 60 / 35 / n/a
Unrolled partial bcopy unaligned L1 / L2 / L3 / L4 cache / main memory	35 / 13 / 45 / 20 / n/a
mappings	34-41

Lmbench3

Most benefits in L3 and L4 cache, overall +40%

Measured operation	Deviation z196 to z10 in %
simple syscall	-30
simple read/write	0
select of file descriptors	35
signal handler	-22
process fork	25
libc bcopy aligned L1 / L2 / L3 / L4 cache / main memory	0 / 20 / 100 / 300 / n/a
libc bcopy unaligned L1 / L2 / L3 / L4 cache / main memory	15 / 0 / 0 / 40 / n/a
memory bzero L1 / L2 / L3 / L4 cache / main memory	35 / 90 / 300 / 800 / n/a
memory partial read L1 / L2 / L3 / L4 cache / main memory	45 / 25 / 130 / 500 / n/a
memory partial read/write L1 / L2 / L3 / L4 cache / main memory	15 / 15 / 10 / 120 / n/a
memory partial write L1 / L2 / L3 / L4 cache / main memory	80 / 30 / 60 / 300 / n/a
memory read L1 / L2 / L3 / L4 cache / main memory	10 / 30 / 40 / 300 / n/a
memory write L1 / L2 / L3 / L4 cache / main memory	50 / 30 / 30 / 180 / n/a
Mmap read L1 / L2 / L3 / L4 cache / main memory	50 / 35 / 85 / 300 / n/a
Mmap read open2close L1 / L2 / L3 / L4 cache / main memory	40 / 35 / 50 / 200 / n/a
Read L1 / L2 / L3 / L4 cache / main memory	20 / 40 / 90 / 300 / n/a
Read open2close L1 / L2 / L3 / L4 cache / main memory	25 / 35 / 90 / 300 / n/a
Unrolled bcopy unaligned L1 / L2 / L3 / L4 cache / main memory	100 / 75 / 75 / 200 / n/a
memory	70 / 0 / 80 / 300 / n/a
mappings	40

CPU-intensive benchmark suite

Stressing a system's processor, memory subsystem and compiler

Workloads developed from real user applications

Exercising integer and floating point in C, C++, and Fortran programs

Can be used to evaluate compile options

Can be used to optimize the compiler's code generation for a given target system

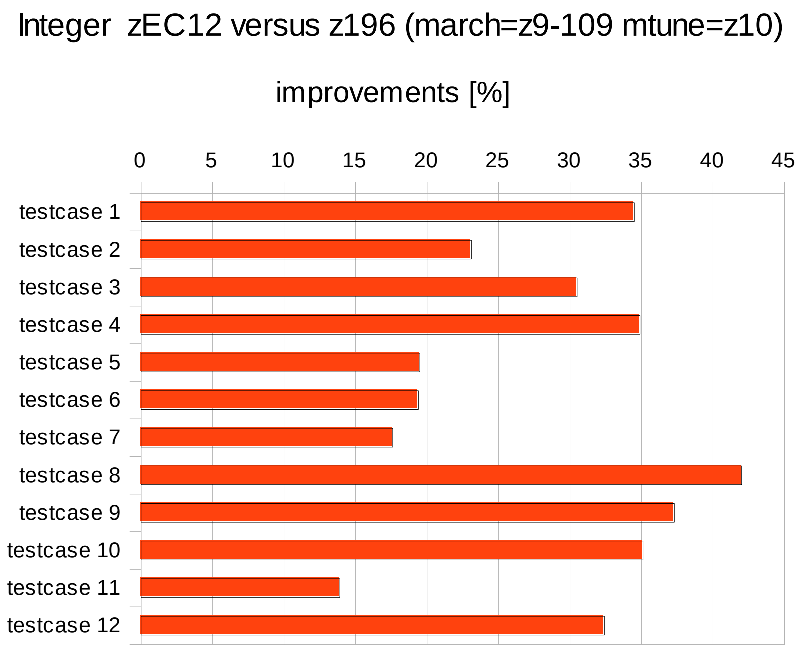
Configuration

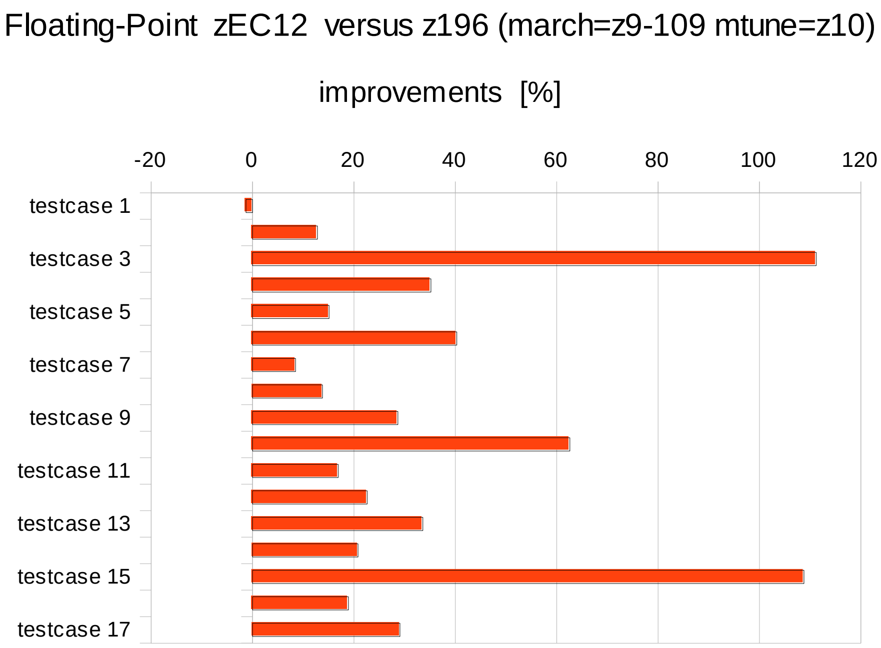
- 1 CPU, 2 GiB memory, executing one test case at a time
- N CPUs, executing N same test cases at a time

Single-threaded, compute-intense workload

SLES11 SP2 GA, gcc-4.3-62.198, glibc-2.11.3-17.31.1 using default machine optimization options as in gcc-4.3 s390x

- Integer suite improves by 28% (geometric mean)
- Floating Point suite improves by 31% (geometric mean)



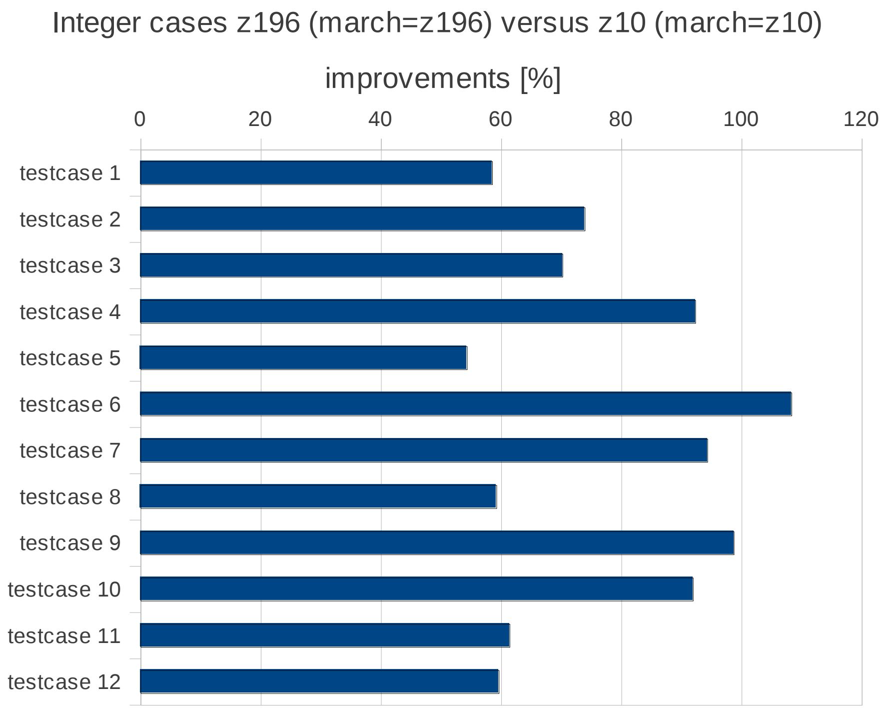


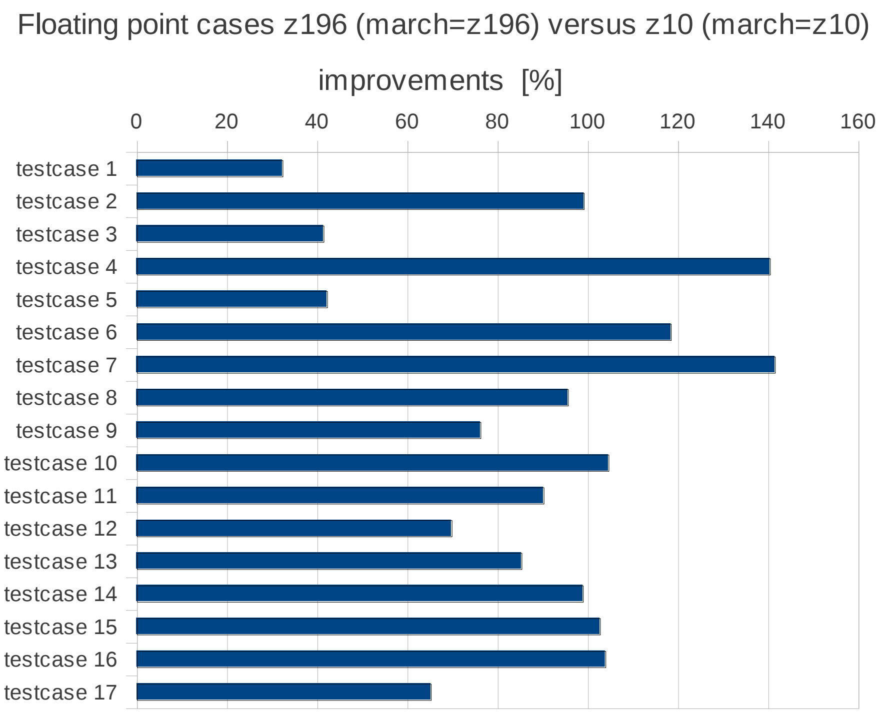
2010

Single-threaded, compute-intense workload

Linux: Internal driver (kernel 2.6.29) gcc 4.5, glibc 2.9.3

- Integer suite improves by 76% (geometric mean)
- Floating Point suite improves by 86% (geometric mean)





Benchmark description – Network

Network Benchmark which simulates several workloads

Transactional Workloads

- 2 types
  - RR – A connection to the server is opened once for a 5 minute time frame
  - CRR – A connection is opened and closed for every request/response
- 4 sizes
  - RR 1x1 – Simulating low latency keepalives
  - RR 200x1000 – Simulating online transactions
  - RR 200x32k – Simulating database query
  - CRR 64x8k – Simulating website access

Streaming Workloads – 2 types

- STRP/STRG – Simulating incoming/outgoing large file transfers (20mx20)

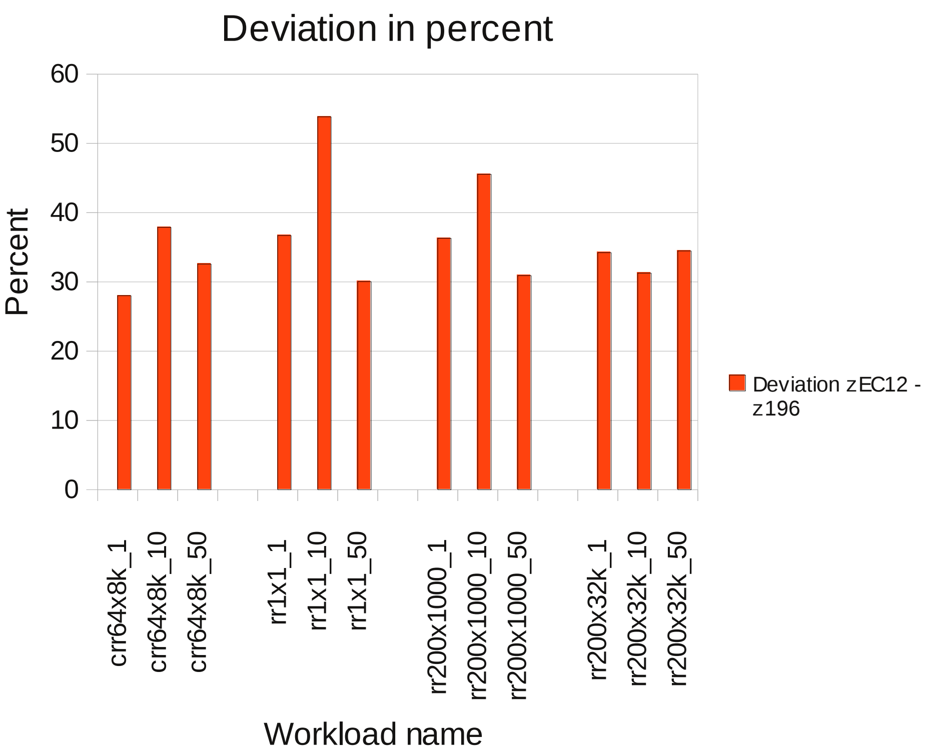
All tests are done with 1, 10 and 50 simultaneous connections on multiple connection types (different cards and MTU configurations)

AWM Hipersockets MTU-32k IPv4 LPAR-LPAR

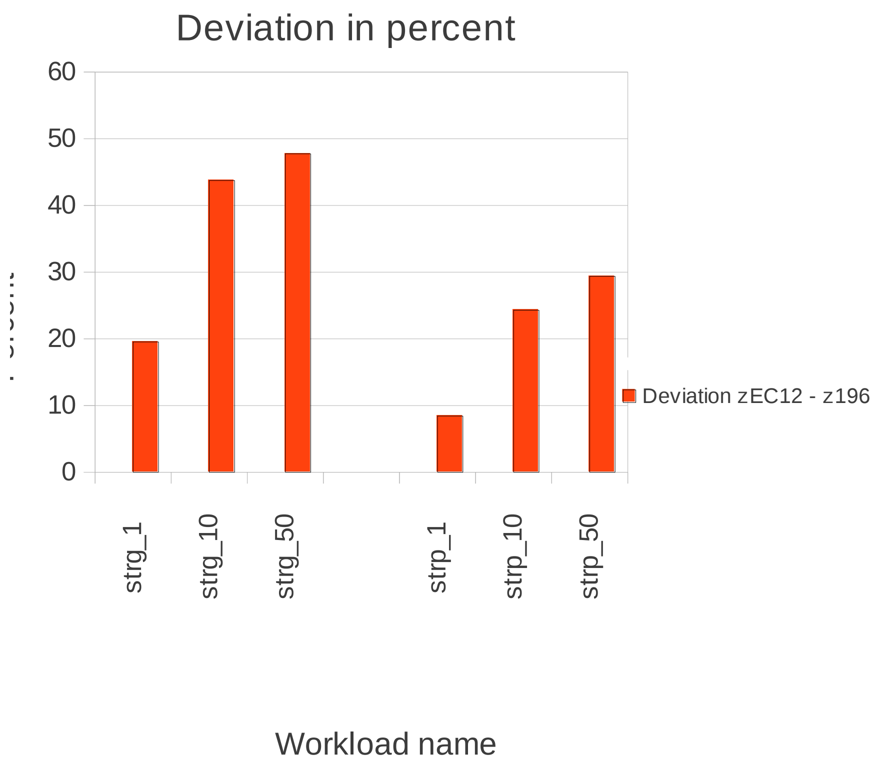
More transactions / throughput with 1, 10 and 50 connections

More data transferred at 20 to 30 percent lower CPU consumption

RR/CRR Transactions per second



STREAM throughput



Benchmark description – Re-Aim 7

Scalability benchmark Re-Aim-7

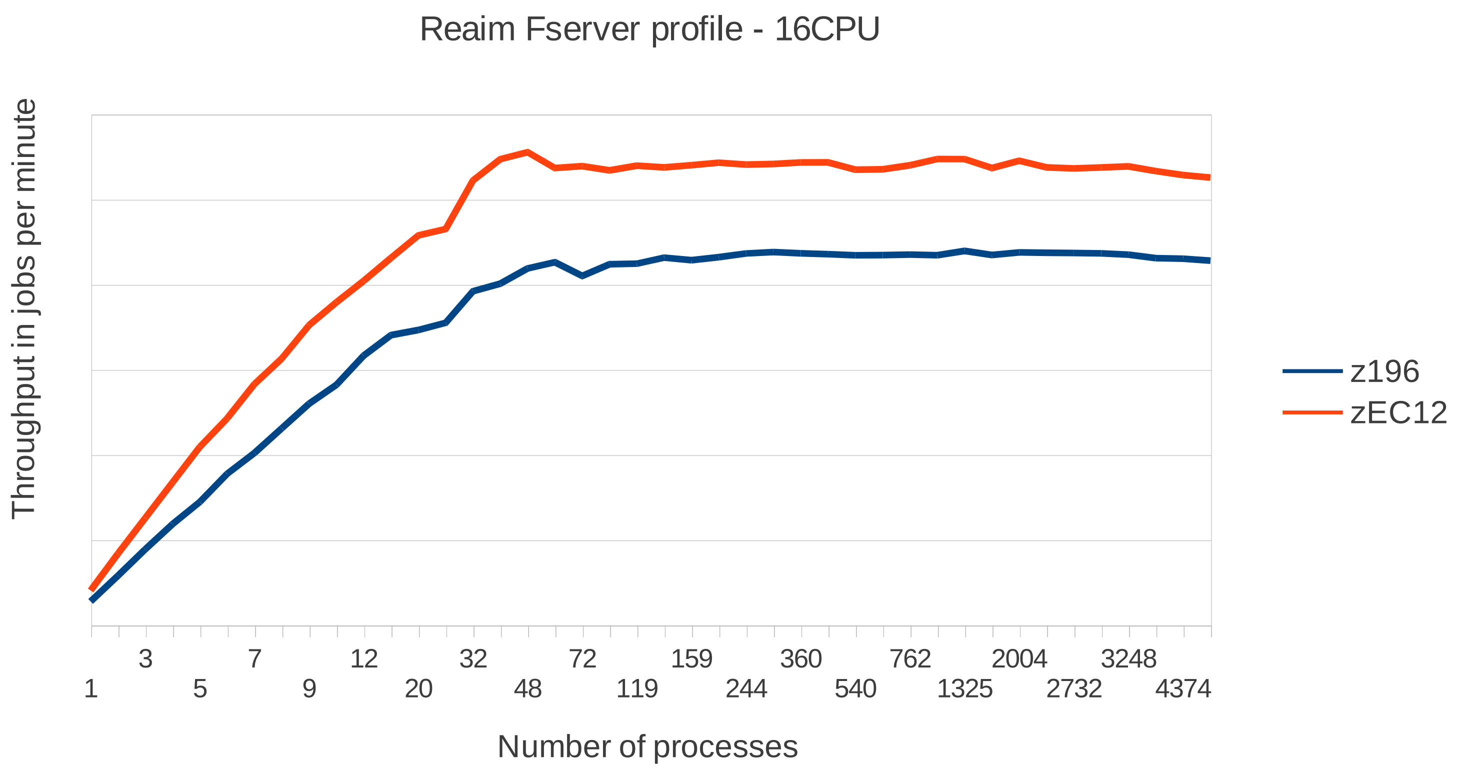
- Open Source equivalent to the AIM Multiuser benchmark
- Workload patterns describe system call ratios (patterns can be more ipc, disk or calculation intensive)
- The benchmark then scales concurrent jobs until the overall throughput drops
  - Starts with one job, continuously increases that number
  - Overall throughput usually increases until #threads ≈ #CPUs
  - Then threads are further increased until a drop in throughput occurs
  - Scales up to thousands of concurrent threads stressing the same components
- Often a good check for non-scaling interfaces
  - Some interfaces don't scale at all (1 Job throughput ≈ multiple jobs throughput, despite >1 CPUs)
  - Some interfaces only scale in certain ranges (throughput suddenly drops earlier as expected)
- Measures the amount of jobs per minute a single thread and all the threads can achieve

Our Setup

- 2, 8, 16 CPUs, 4 GiB memory, scaling until overall performance drops
- Using a journaled file system on an xpram device (stress FS code, but not be I/O bound)
- Using fserver, new-db and compute workload patterns

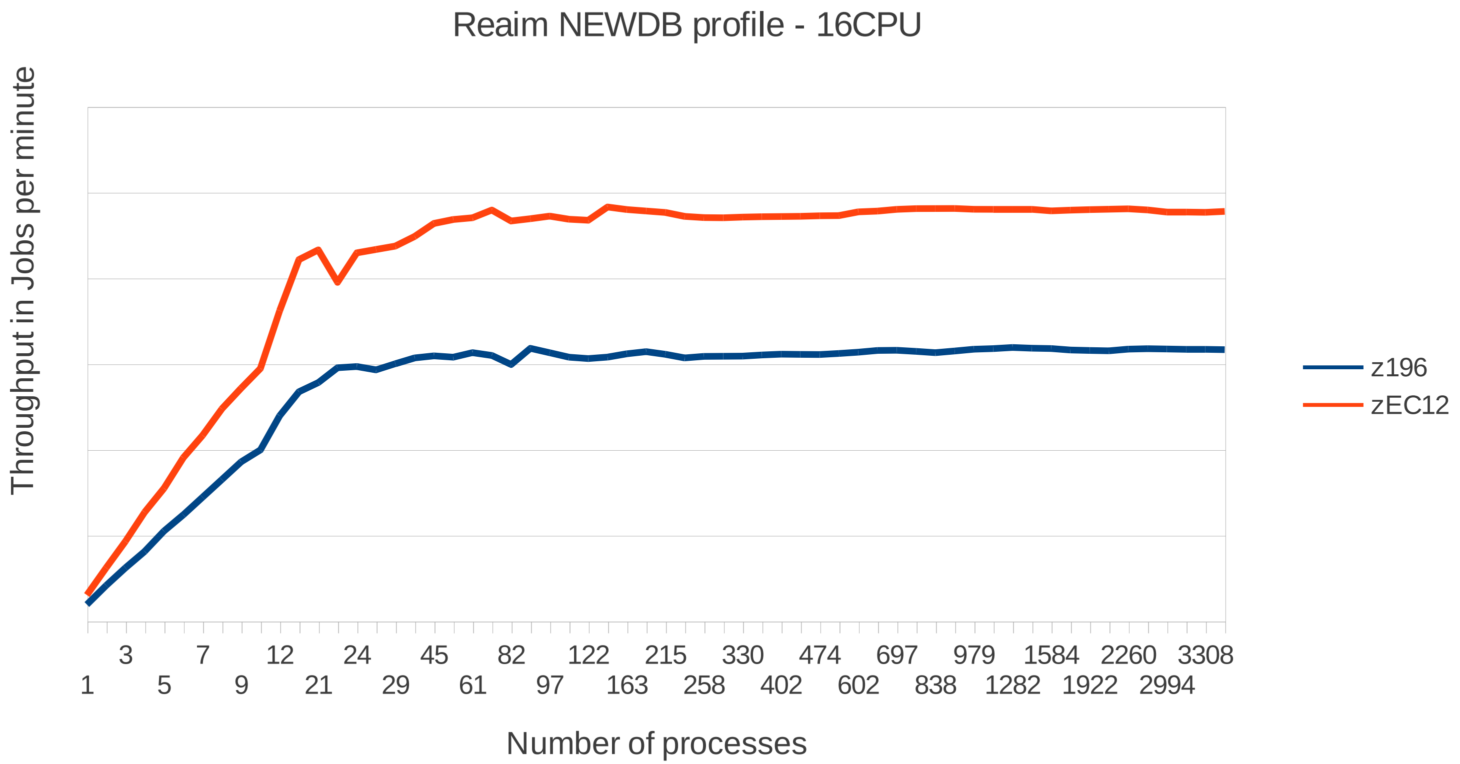
Re-Aim Fserver

Higher throughput with 4, 8, and 16 PUs (25% to 40% percent) at 30% lower processor consumption



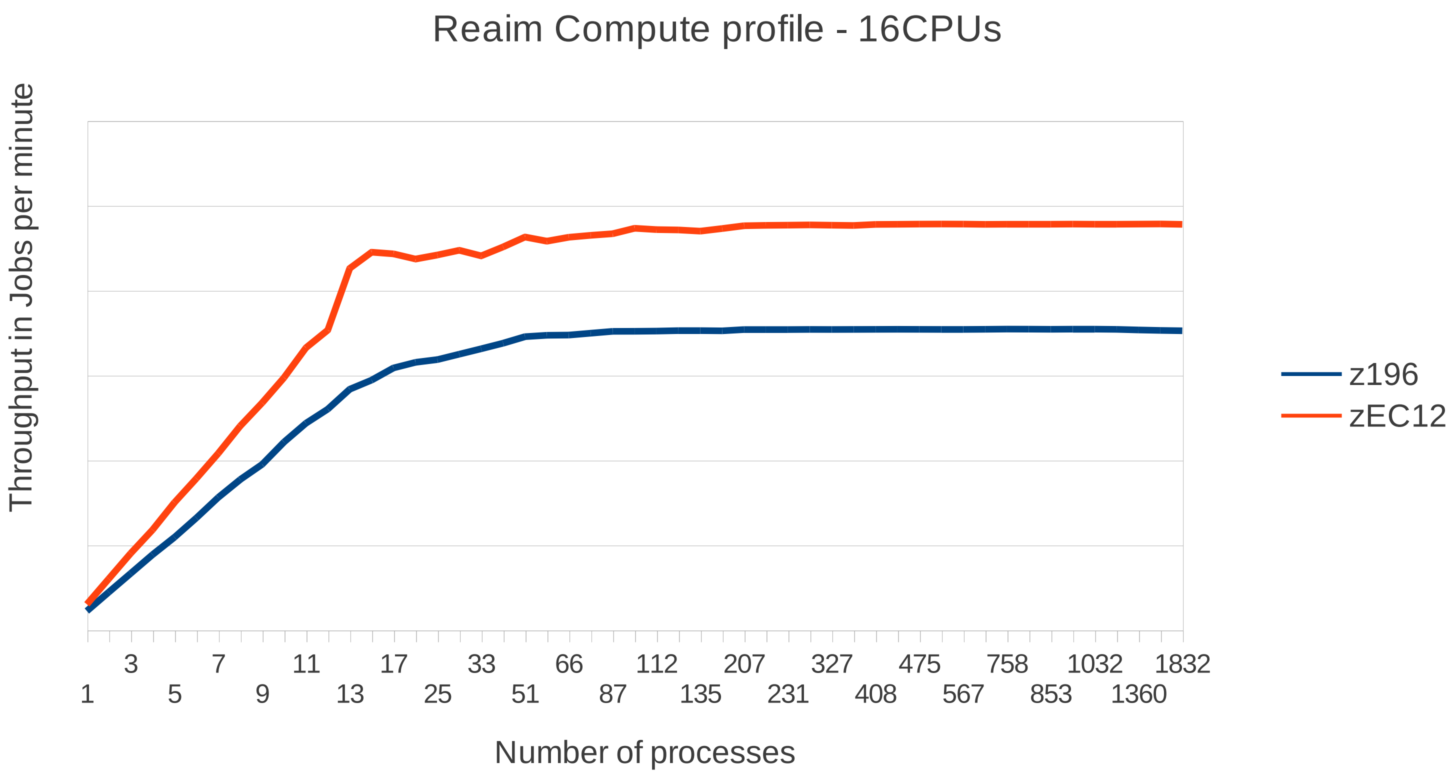
Re-Aim Newdb

Higher throughput with 4, 8, and 16 CPUs (average 55%) at 35% lower processor consumption



Re-Aim Compute

Higher throughput with 4, 8, and 16 CPUs (average 35%) at 20 to 30% lower processor consumption

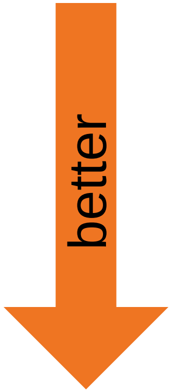


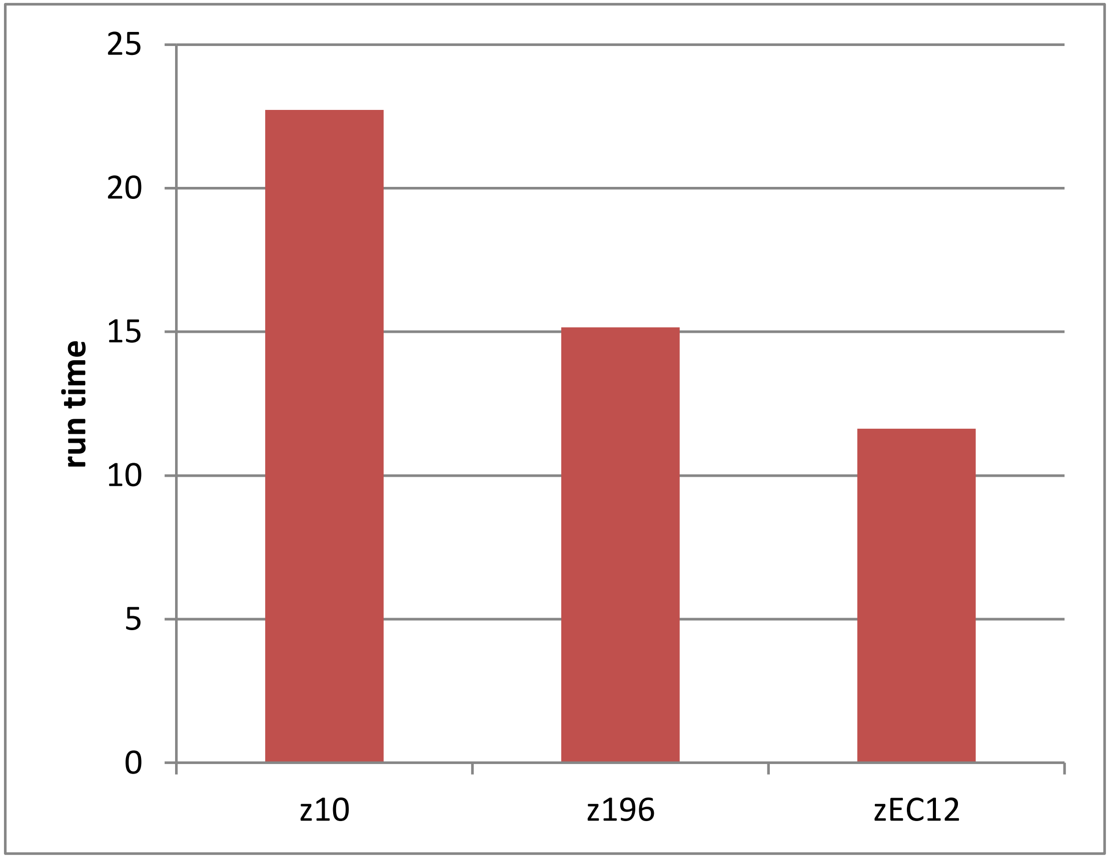
DB2 database BI workload

- complex database warehouse database
- Using 128 GB memory
- 16 CPUs
- No I/O constraint

DB2 workload – hardware comparison

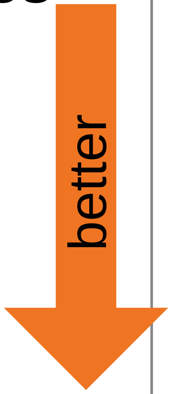
- z10 → zEC12 provides ~factor 2
- z196 → zEC12 ~30%

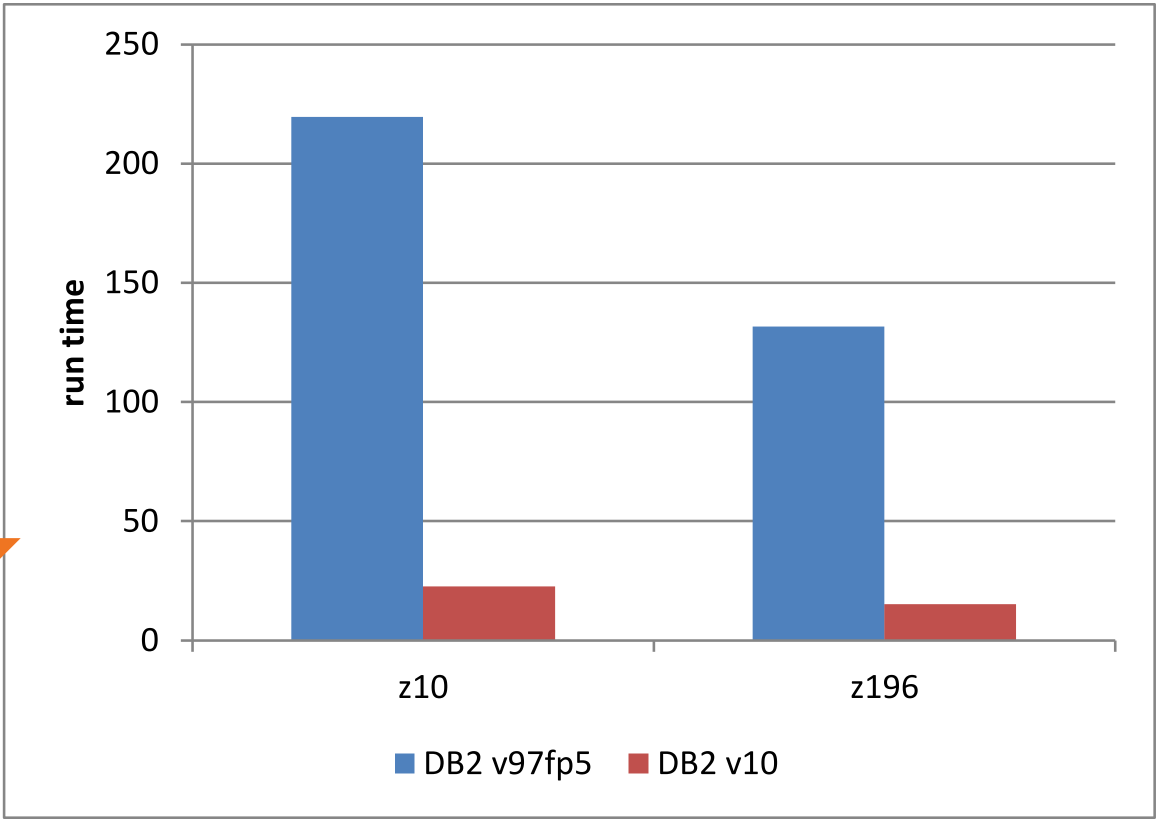




DB2 workload – version 9.5 / 10 comparison

- ~ 9x more throughput
- z196 is 1.5 times as fast as z10



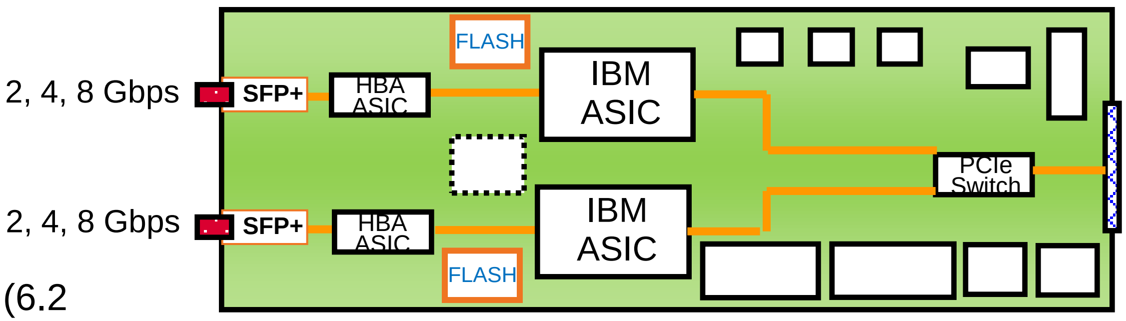


Agenda

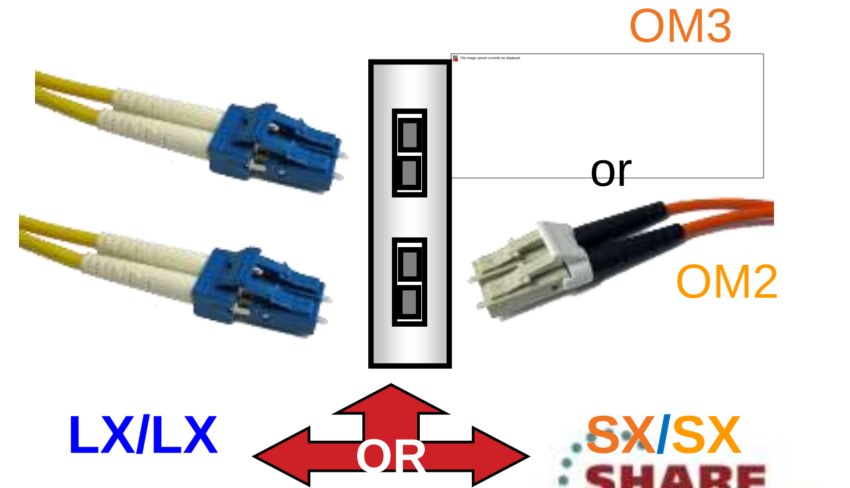
- zEC12 – hardware design
- zEC12 – performance comparison with z196
- OSA Express4S and FICON Express 8S results
- Miscellaneous
  - RHEL 5.9
  - Large Pages
  - Java / WAS

FICON Express8S – SX and 10KM LX in the PCIe I/O drawer

- For FICON, zHPF, and FCP environments
  - CHPID types: FC and FCP
    - 2 PCHIDs/CHPIDs
- Auto-negotiates to 2, 4, or 8 Gbps
- Increased performance compared to FICON Express8
- 10KM LX - 9 micron single mode fiber
  - Unrepeated distance - 10 kilometers (6.2 miles)
  - Receiving device must also be LX
- SX - 50 or 62.5 micron multimode fiber
  - Distance variable with link data rate and fiber type
  - Receiving device must also be SX
- 2 channels of LX or SX (no mix)
- Small form factor pluggable (SFP) optics
  - Concurrent repair/replace action for each SFP



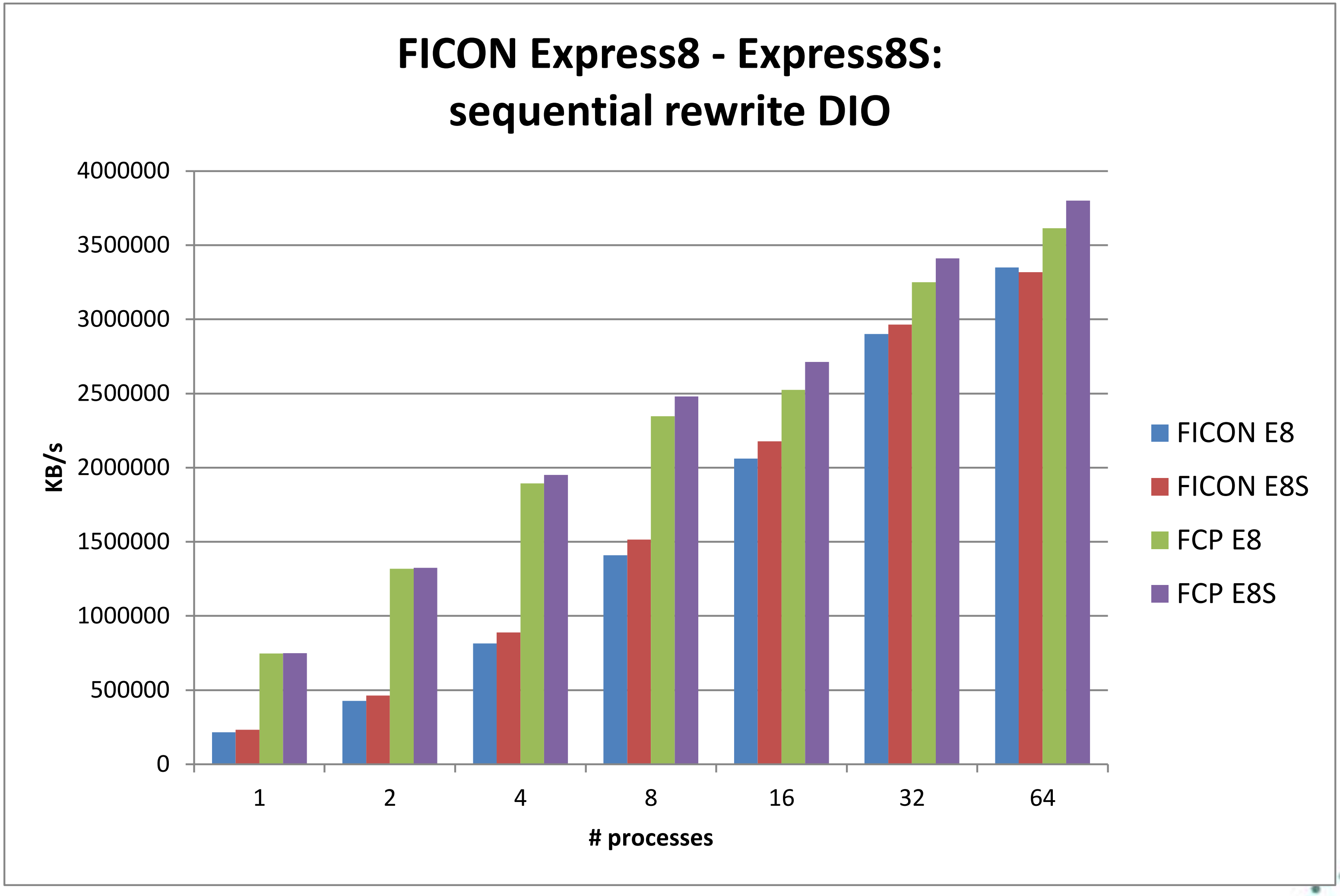
FC 0409 – 10KM LX, FC 0410 – SX



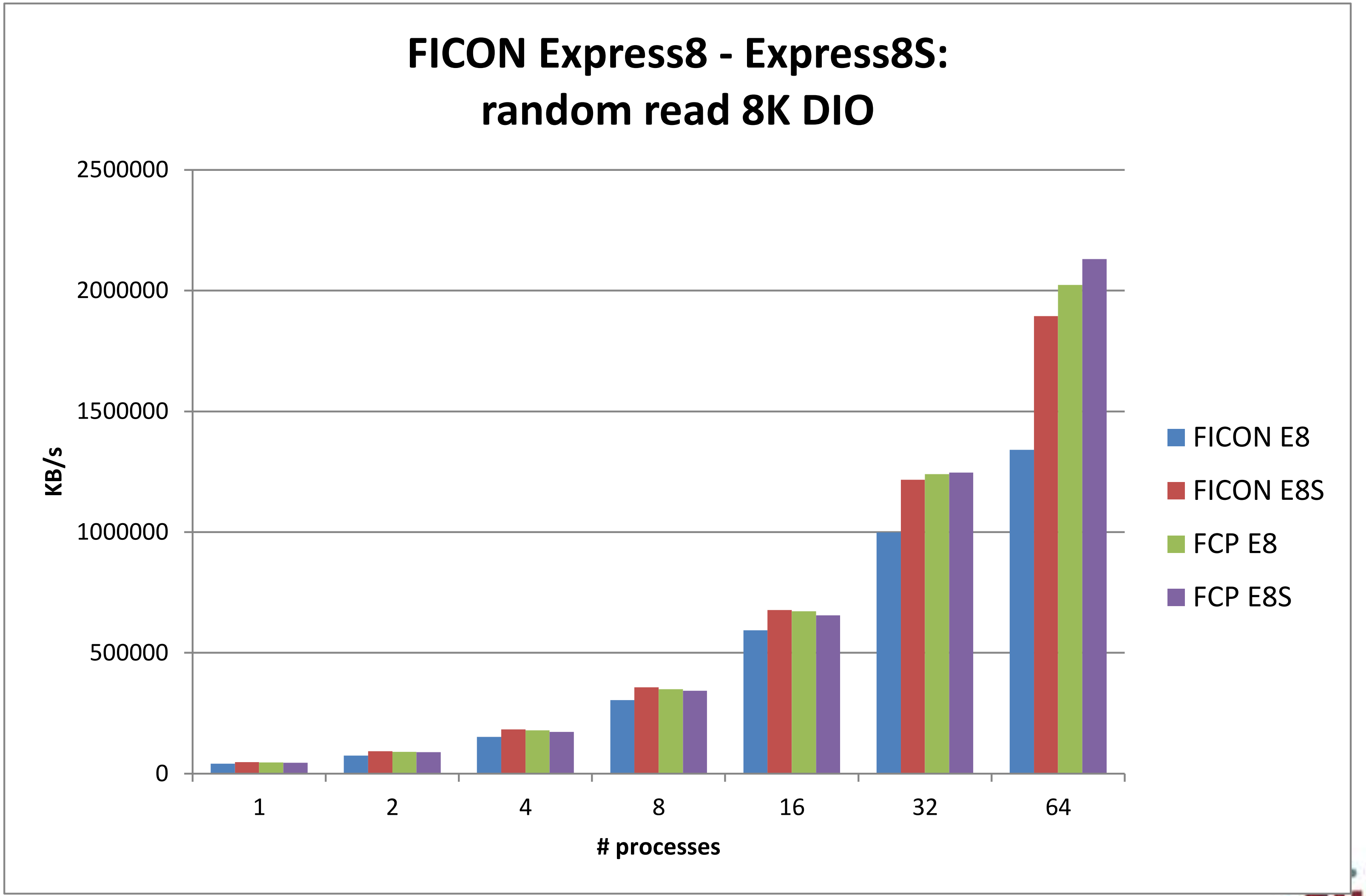
FICON Express 8S - overview

- Available since z196 GA2
- All measurements on z196 with SLES11 SP2
- Benchmark description
  - Multiple processes - each process writes or reads to a single file, volume or disk
  - Can be configured to run with and without page cache (direct I/O), operating modes: Sequential write/rewrite/read + Random write/read
  - Setup: 256 MiB, file size 2 GiB
  - Scaling over 1, 2, 4, 8, 16, 32, 64 processes
  - Sync and Drop Caches prior to every invocation

FICON Express 8S – results (1)

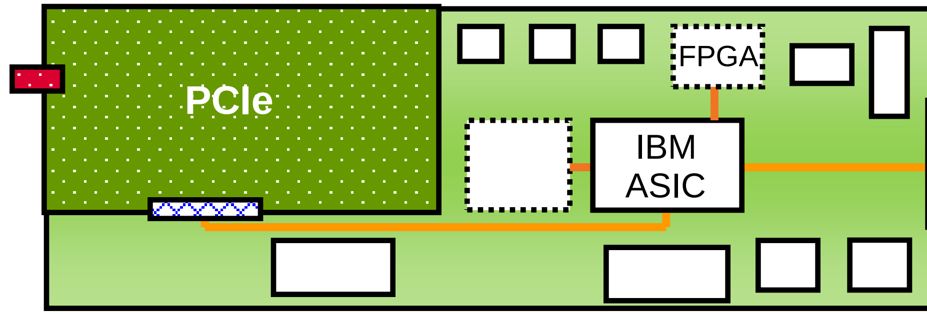


FICON Express 8S – results (2)



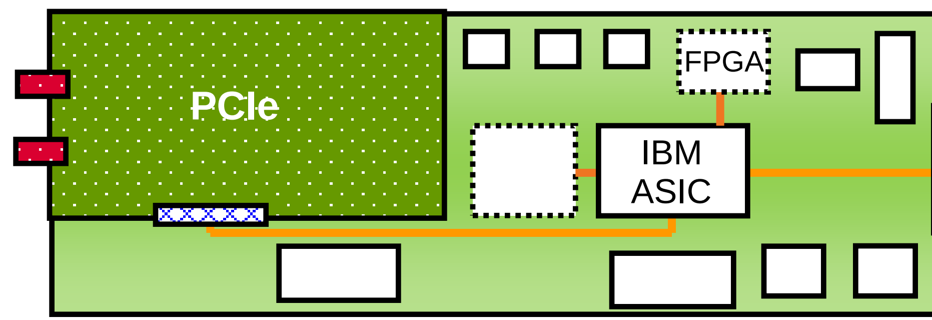
OSA-Express4S GbE and 10 GbE fiber for the PCIe I/O drawer

- 10 Gigabit Ethernet (10 GbE)
  - CHPID types: OSD, OSX
  - Single mode (LR) or multimode (SR) fiber
  - One port of LR or one port of SR
    - 1 PCHID/CHPID



FC 0406 – 10 GbE LR, FC 0407 – 10 GbE SR

- Gigabit Ethernet (GbE)
  - CHPID types: OSD (CHPID OSN not supported)
  - Single mode (LX) or multimode (SX) fiber
  - Two ports of LX or two ports of SX
    - 1 PCHID/CHPID



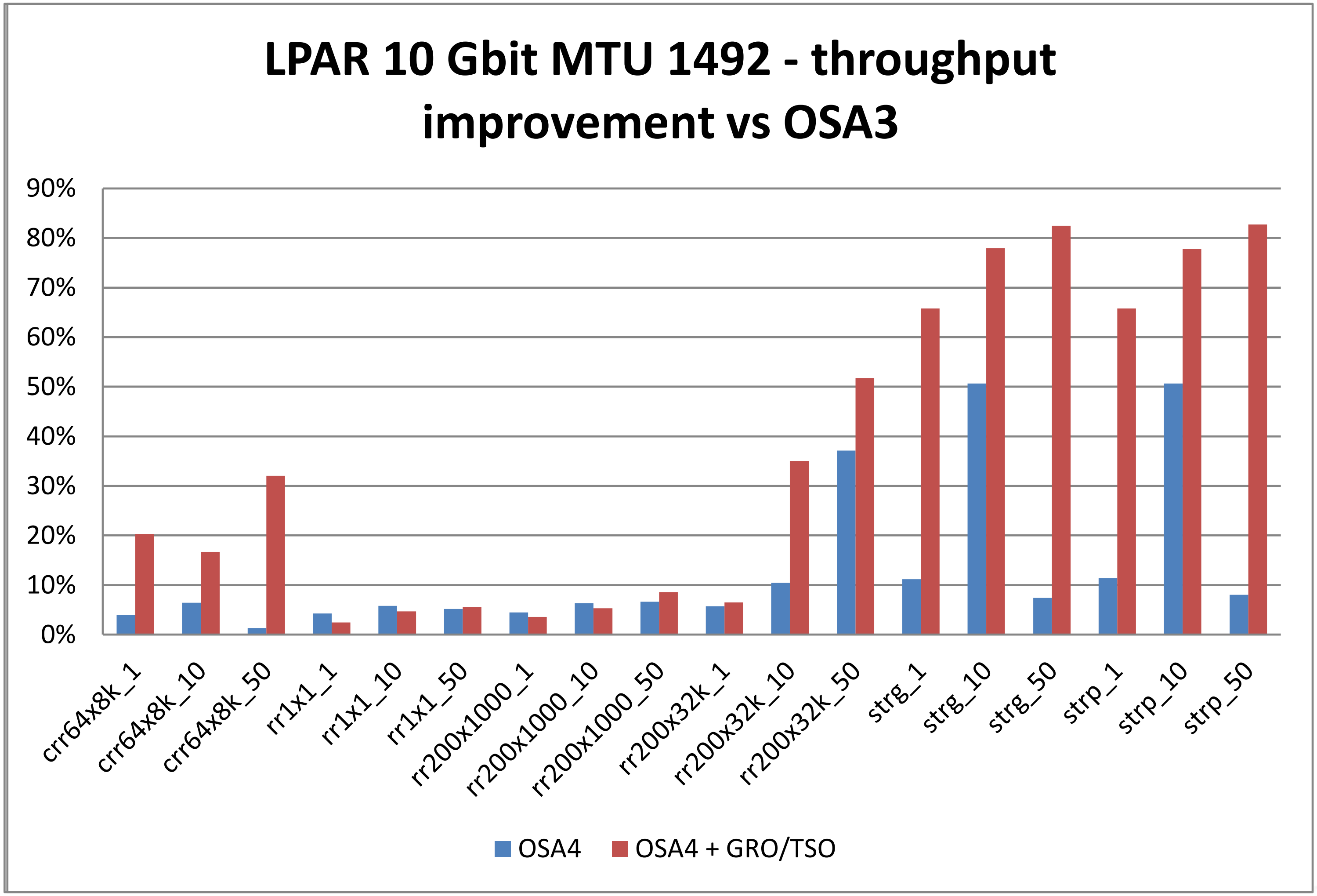
FC 0404 – GbE LX, FC 0405 – GbE SX

- Small form factor optics – LC Duplex

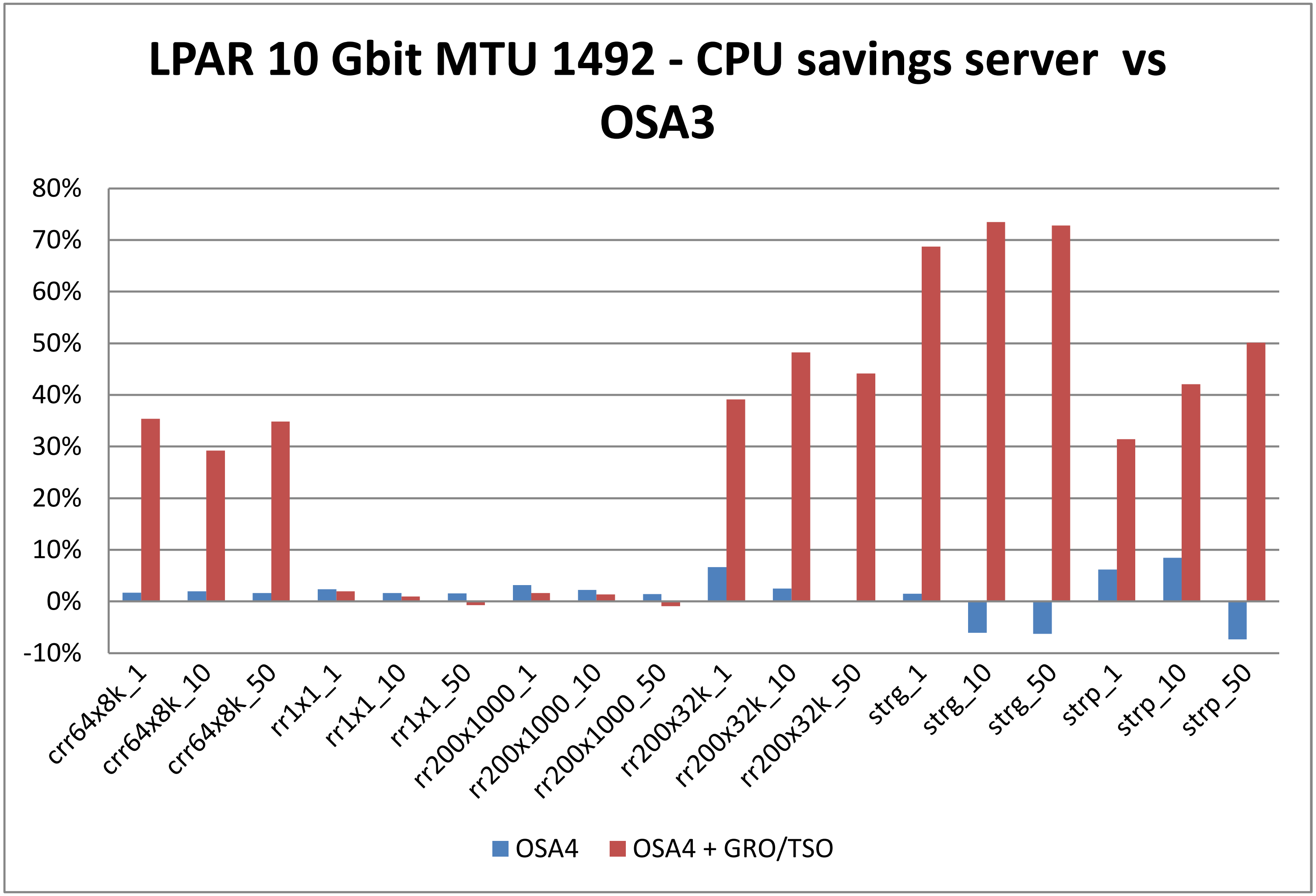
OSA Express 4S - overview

- Available since z196 GA2
- All measurements on z196 with SLES11 SP2
- Benchmark description see above

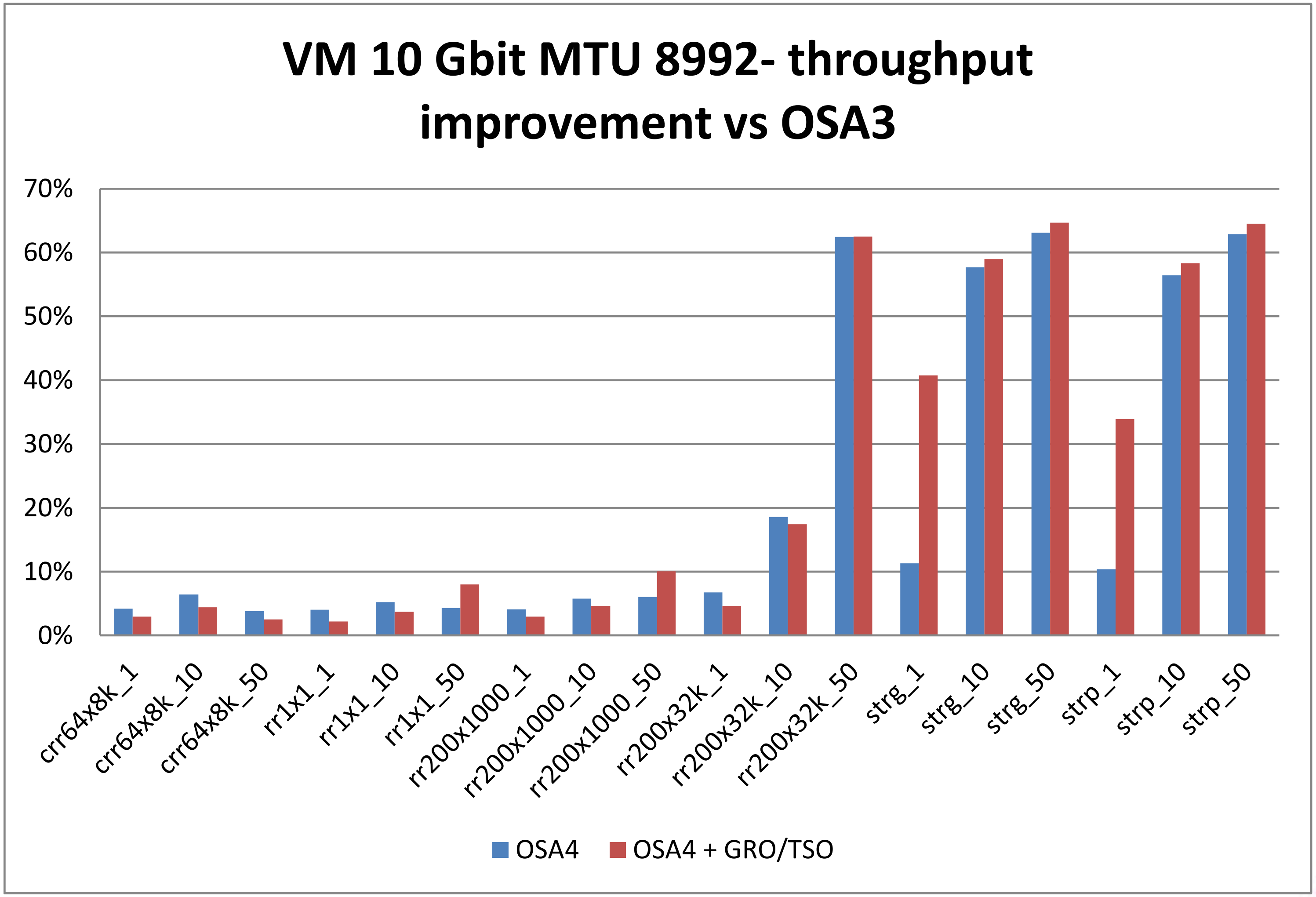
OSA-Express 4S – LPAR (1)



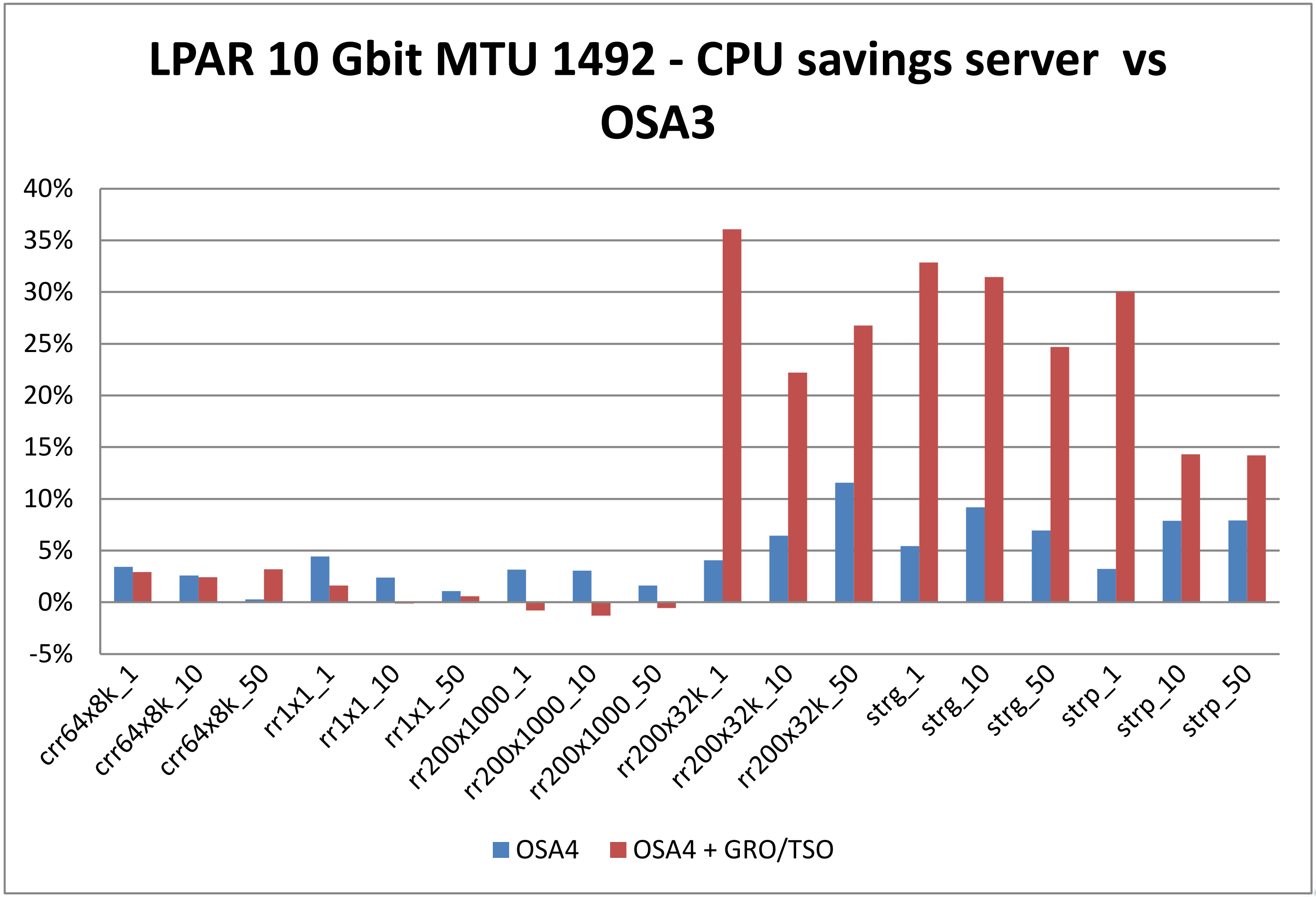
OSA-Express 4S – LPAR (2)



OSA-Express 4S – z/VM (1)



OSA-Express 4S – z/VM (2)



Agenda

- zEC12 – hardware design
- zEC12 – performance comparison with z196
- OSA Express4S and FICON Express 8S results
- Miscellaneous
  - RHEL 5.9
  - Large Pages
  - Java / WAS

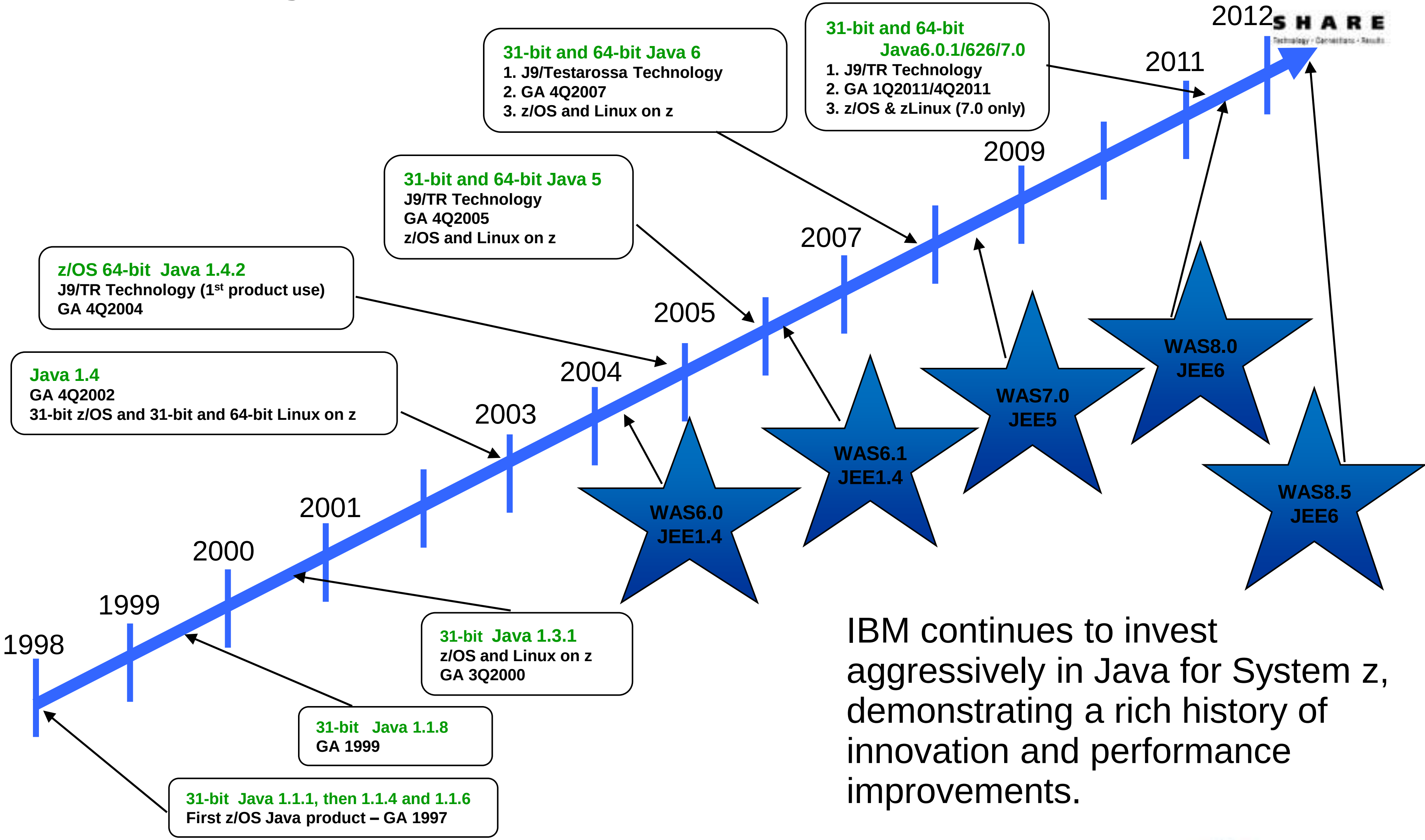
RHEL 5.9

- Two performance features delivered
  - VDSO
  - HyperPAV
- Summary see my blog post
  - linuxmain.blogspot.com/2013/01/red-hat-enterprise-linux-59-released.html

Large Pages – performance advances

- **Explicit use of large pages**
  - used directly from applications, e.g. Java –Xlp
  - Available in all distos
- **Kernel pages** mapped automatically with 1 MB pages
  - SLES11 SP2 and later distros
- **Libhugetlbfs**
  - preload library for not yet enabled applications and lib for relinking applications
  - next distro updates
- **Transparent Huge Pages**
  - pageable
  - Next major distro update
- **Hardware benefit:** ~5% (TLB savings) for CPU intense, more for memory intense workload
- **Software benefit:** it depends

Java on System z – 15 Years of Innovation



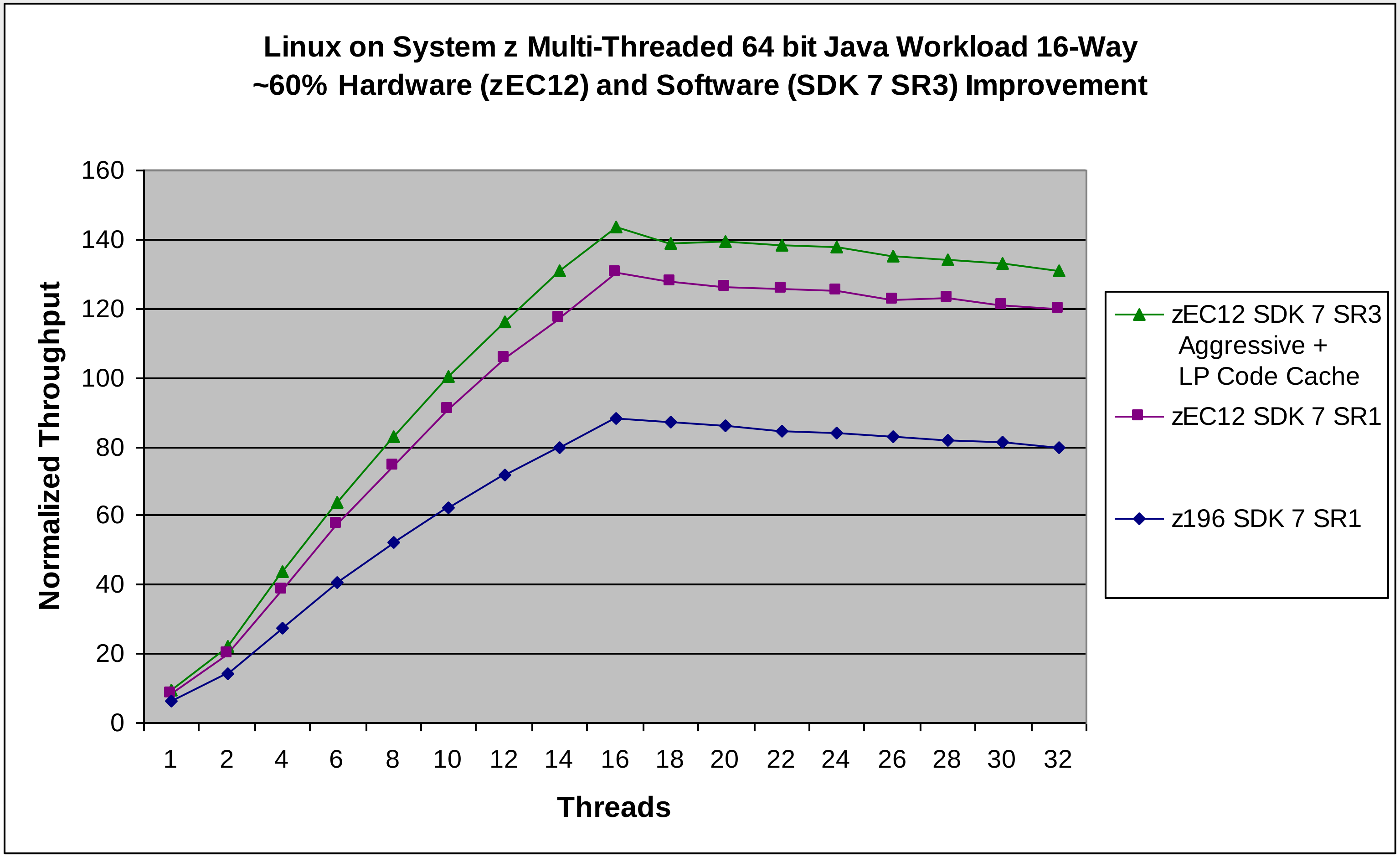
IBM continues to invest aggressively in Java for System z, demonstrating a rich history of innovation and performance improvements.

Testimonials: http://www-01.ibm.com/software/os/systemz/testimonials/
http://www.centerline.net/review/#/3332_B

Timelines and deliveries are subject to change.

Linux on System z and Java7SR3 on zEC12:

64-Bit Java Multi-threaded Benchmark on 16-Way

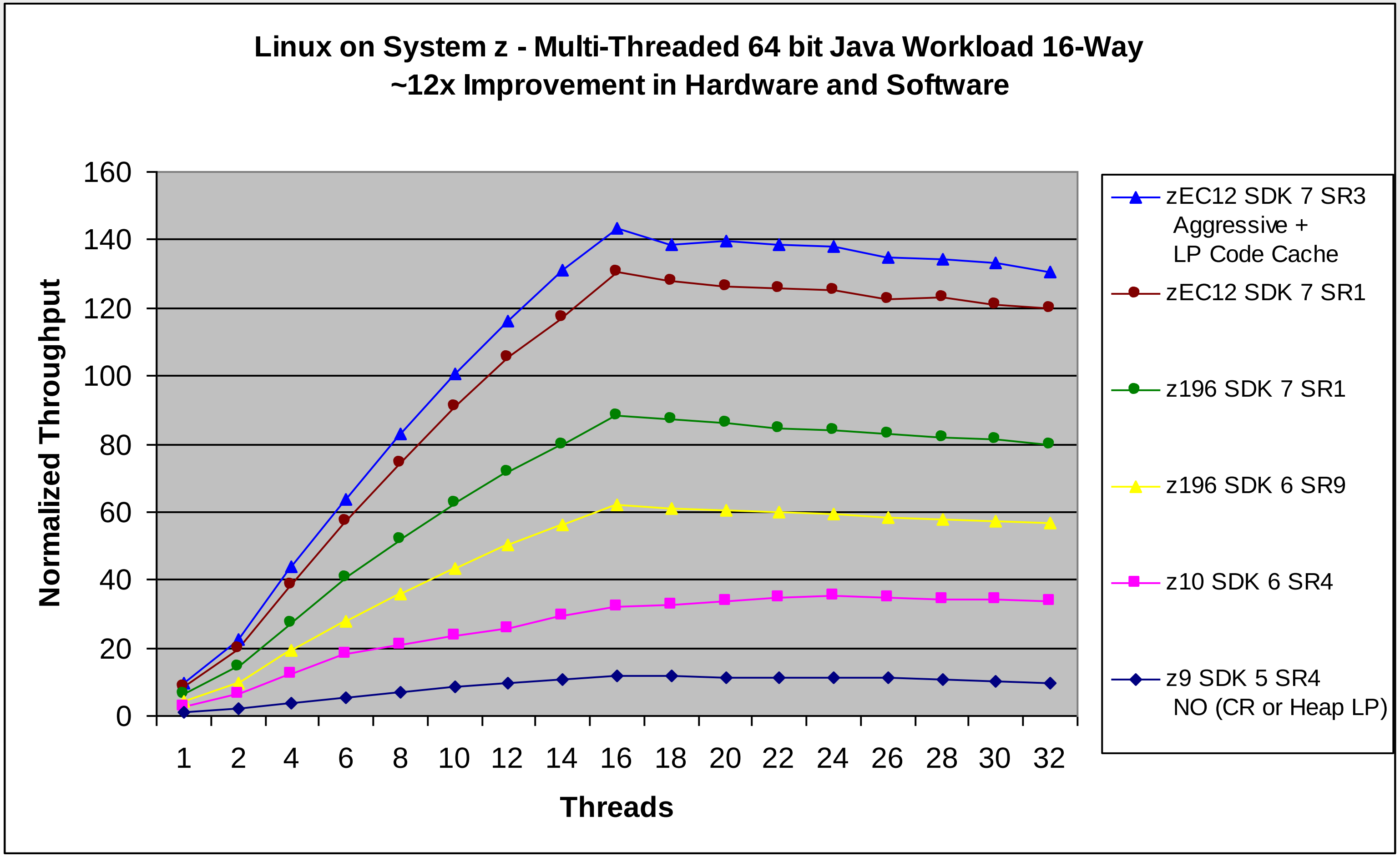


Aggregate 60% improvement from zEC12 and Java7SR3

- **zEC12 offers a ~45% improvement over z196 running the Java Multi-Threaded Benchmark**
- **Java7SR3 offers an additional ~10% improvement** **(-Xaggressive)**

Linux on System z and Java7SR3 on zEC12:

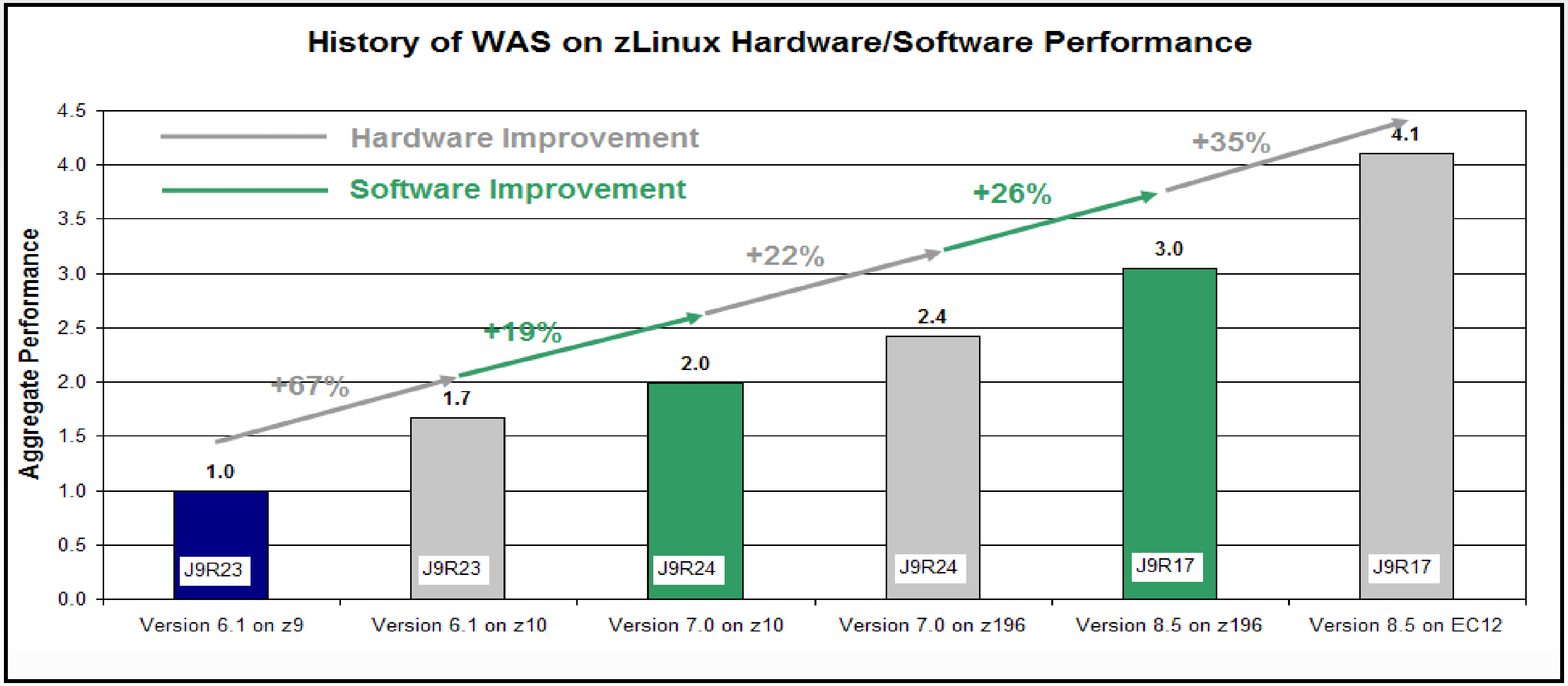
64-Bit Java Multi-threaded Benchmark on 16-Way



~12x aggregate hardware and software improvement comparing Java5SR4 on z9 to Java7SR3 on zEC12
LP=Large Pages for Java heap   CR= Java compressed references
Java7SR3 using -Xaggressive + 1Meg large pages

WAS on zLinux –

Aggregate HW, SDK and WAS Improvement: WAS 6.1 (Java 5) on z9 to WAS 8.5 (Java 7) on zEC12



~4x aggregate hardware and software improvement comparing WAS 6.1 Java5 on z9 to WAS 8.5 Java7 on zEC12

Summary

- zEC12 offers solid performance gains
  - Performance improvement seen in nearly all areas measured
  - More improvement than just from higher rate to expect
    - Rate is up from 5.2 GHz to 5.5 GHz which means close to 6 percent higher
    - New cache setup with much bigger caches
    - Out-of-order execution of the second generation
- Also improvements in network and I/O
  - Enable TSR+GRO for OSA Expresss 4S
- More improvements outside the hardware
  - Large pages, Java, DB2, WAS …..

Dr. Eberhard Pasch
epasch@de.ibm.com

Linux on System z – Tuning hints and tips:
http://www.ibm.com/developerworks/linux/linux390/perf/index.html

Mainframe Linux blog: http://linuxmain.blogspot.com

Other Linux performance sessions at SHARE

- 12378: Running Java on Linux on System z
- 12477: z/VM Performance Update for 2012
- 13109 / 13110: Tips Learned Implementing Oracle Solutions With Linux for IBM System z
- ….


Thank
You

Linux on System z performance update

Speaker Names: Dr. Eberhard Pasch

Speakers Company: IBM

Date of Presentation: **Tuesday, February 5, 2013 (11:00am)**

Yosemite C, Ballroom Level

Session Number: 13102

http://linuxmain.blogspot.com/

SHARE in San Francisco 2013