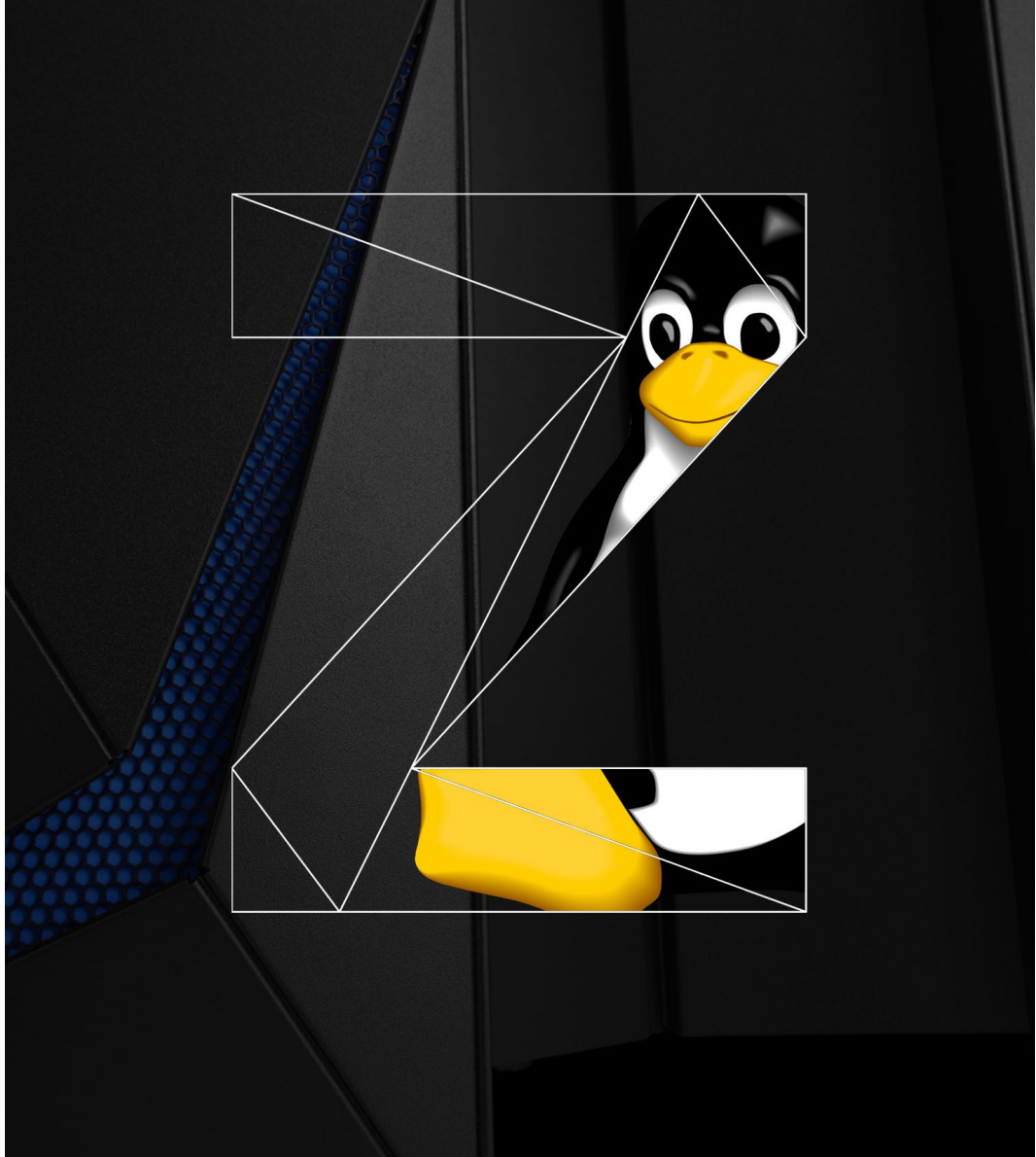
Linux on IBM z14 (z14) Performance Evaluation

What You Get With Machine Improvements Only

—

Barbara Mundle
Jens Markwardt

Linux Performance Analysts

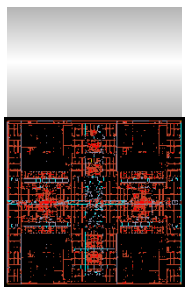


Agenda

- z14 hardware introduction
- Performance evaluation summary
- Test environment details
- Benchmark descriptions and typical results

IBM z Processor Roadmap

65 nm



Workload Consolidation and Integration Engine for CPU Intensive Workloads

Decimal FP

Infiniband

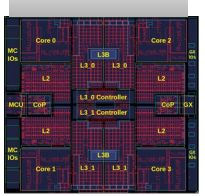
64-CP Image

Large Pages

Shared Memory

45 nm

z196
9/2010



Top Tier Single Thread Performance,System Capacity

Accelerator Integration

Out of Order Execution

Water Cooling

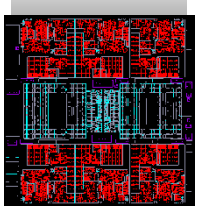
PCIe I/O Fabric

RAIM

Enhanced Energy Management

32 nm

zEC12
8/2012



Leadership Single Thread, Enhanced Throughput

Improved out-of-order

Transactional Memory

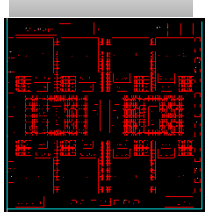
Dynamic Optimization

2 GB page support

Step Function in System Capacity

22 nm

z13
1/2015



Leadership System Capacity and Performance

Modularity & Scalability

Dynamic SMT

Supports two instruction threads

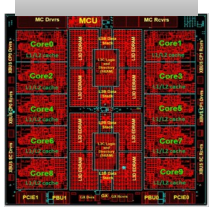
SIMD

PCIe attached accelerators

Business Analytics Optimized

14 nm

z14
7/2017



Pervasive encryption

Low latency I/O for acceleration of transaction processing for DB2 on z/OS

Pause-less garbage collection for enterprise scale JAVA applications

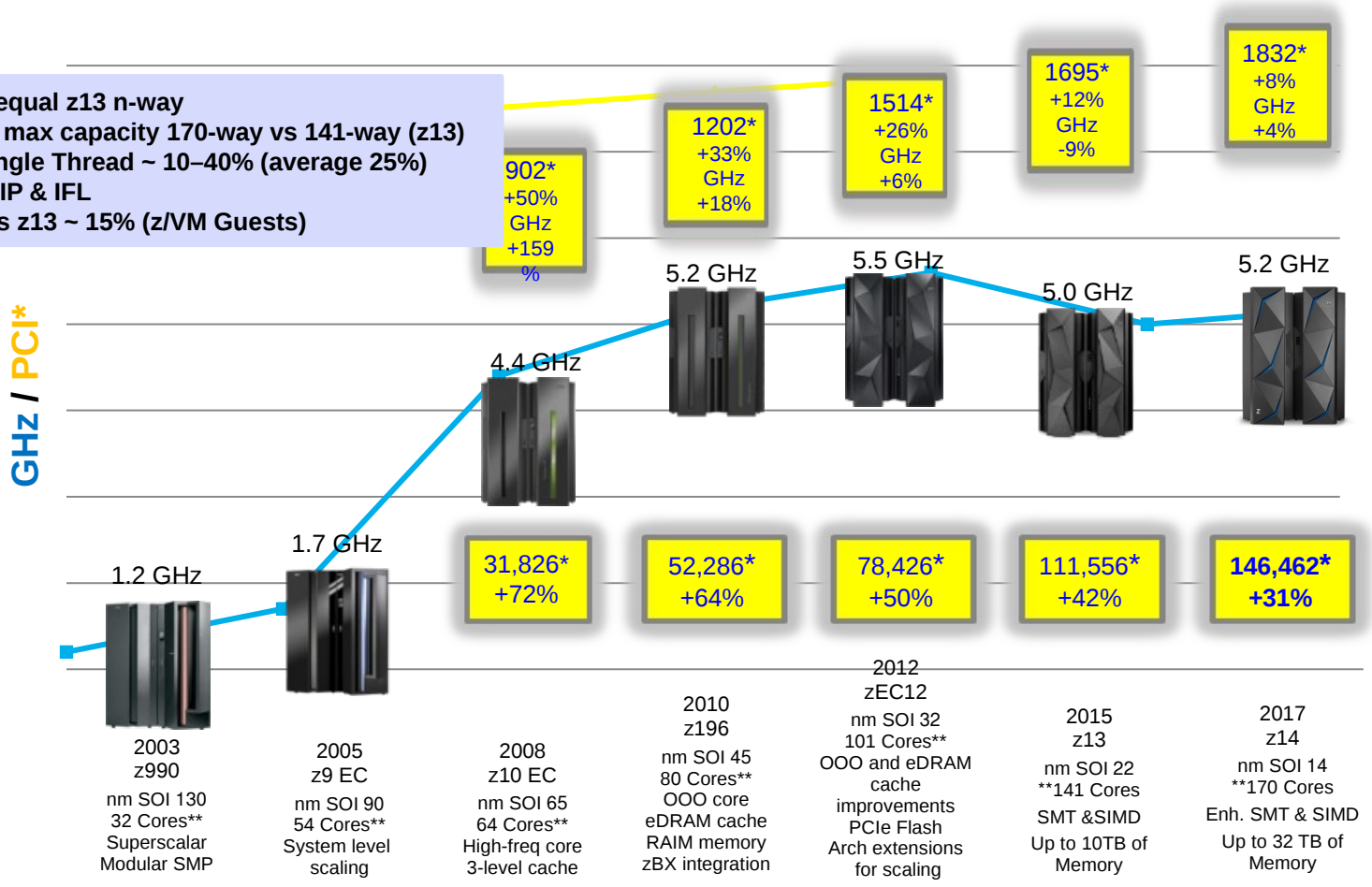
New SIMD instructions

Optimized pipeline and enhanced SMT

Virtual Flash Memory

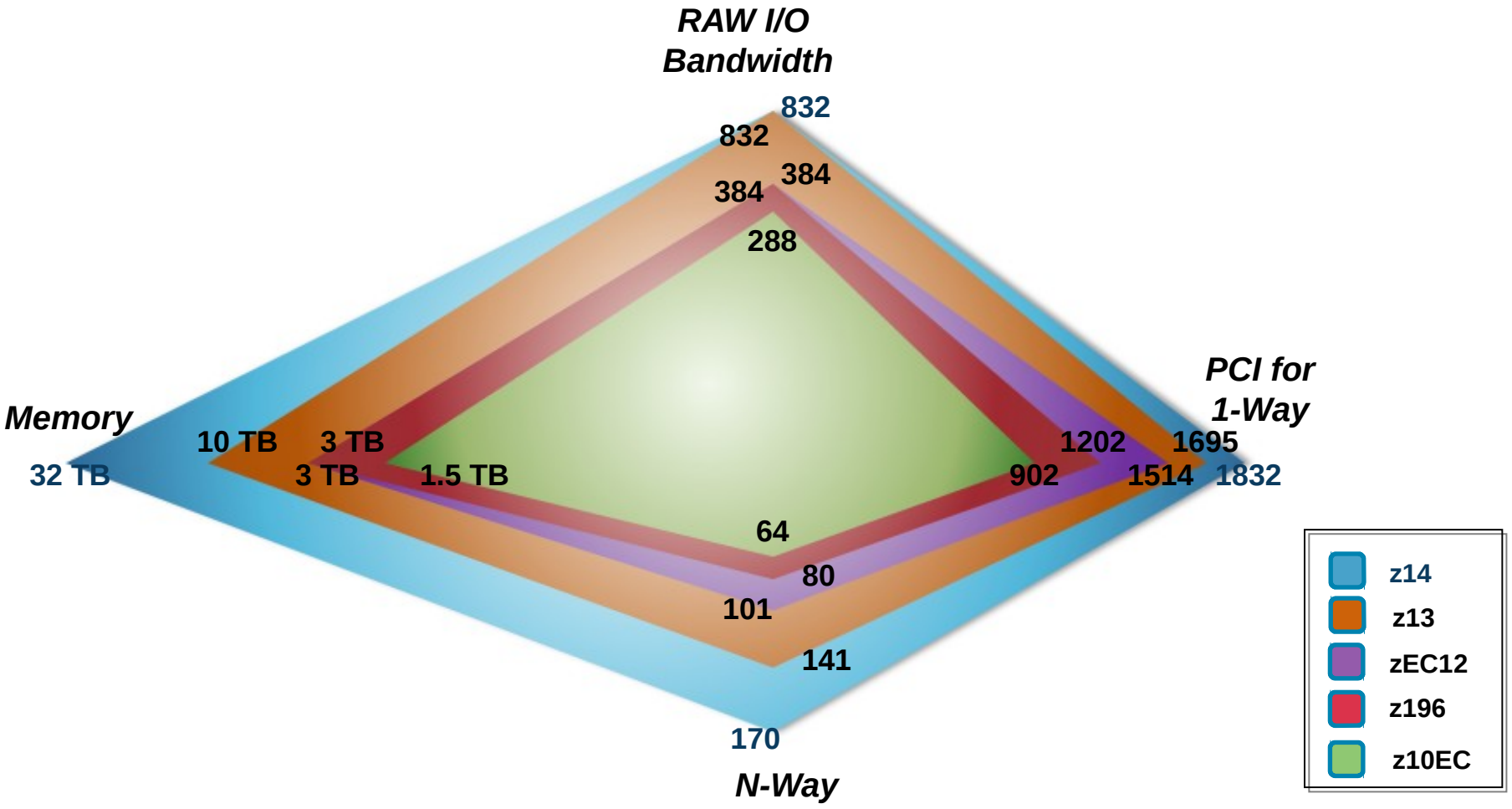
z14 Continues the CMOS Mainframe Heritage

~ 10% for equal z13 n-way
Up to 35% max capacity 170-way vs 141-way (z13)
SMT vs Single Thread ~ 10–40% (average 25%)
- both zIIP & IFL
SMT z14 vs z13 ~ 15% (z/VM Guests)



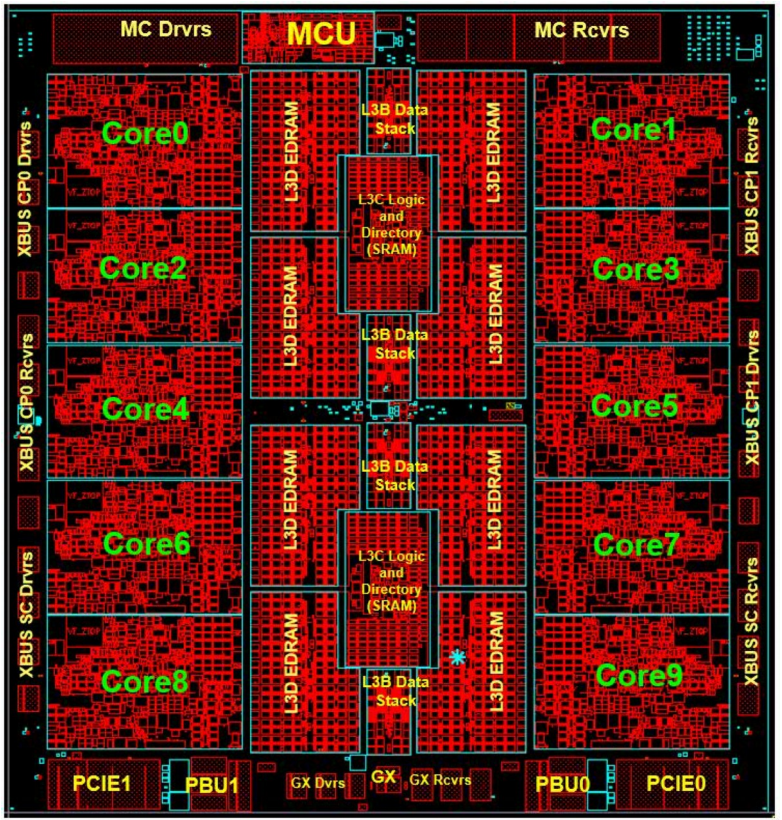
* MIPS Tables are NOT adequate for making comparisons of IBM Z processors. Additional capacity planning required
** Number of PU cores for customer use

Balanced System Design



Processor / Memory (1)

- z14 versus z13
  - Larger caches improve throughput, CPU utilization and program runtime
    - 33% larger L1 for instructions (128 KiB)
    - 2x larger L2 for data (4 MiB)
    - 2x larger L3 cache (128 MiB)
  - New translation / TLB2 design improve address translation
    - 4 concurrent translations
    - Reduced latency
    - Lookup integrated into L2 access pipe
    - New 64 entry TLB2 for 2 GiB pages
    - Optimized 2nd generation SMT2



Performance evaluation summary

- Applications benefit from improvements in these areas
  - Webservers and Java programs
  - Database servers
  - Backup jobs
  - High availability configurations
  - Analytics
  - Blockchain
  - Mobile

Legend

CPU

Memory

Scalability

Network

Disk

Crypto

Our hardware for measurements

2964-701 NC9 (z13)

0.2ns (5.0 GHz)
4 Drawer with a total of 128 IFLs (1 CP)
L1 private 96 KiB instr, 128 KiB data
L2 private 2 MiB instr, 2 MiB data
L3 shared 64 MiB per chip,
shared by all cores
L4 shared 2 x 480 MiB
=> 960 MiB per drawer
2.5 TB RAM per drawer, 10 TB available
Driver 22H Bundle #21 / #30
8 FICON Express16S
HiperSockets
OSA-Express4s & 5s 10 GiB and 1 GiB

3906-701 M04 (z14)

0.19ns (5.2 GHz)
4 Drawer with a total of 141 IFLs (1 CP)
L1 private 128 KiB instr, 128 KiB data
L2 private 2 MiB instr, 4 MiB data
L3 shared 128 MiB per chip,
shared by all cores
L4 shared 4 x 672 MiB
(672 MiB per drawer)
8 TB RAM per drawer, 32 TB available
Driver D32L Bundle #8
8 FICON Express16S+
HiperSockets
OSA-Express5s & 6s 10 GiB and 1 GiB

2831-985 (DS8886)
1945.6 GiB Cache
2038 GiB Processor memory total
64 GiB NVS
384 * 600 GB disks
15.000 RPM
8 FCP ports (16 Gbps)
8 FICON ports (16 Gbps)

General remarks

- All measurements conducted with
  - SLES 12 SP2 (4.4.21-69-default), no z14 feature exploitation available
  - Java-version 8.0.5.0 - pxz6480sr5-20170905_01(SR5)
  - Transparent huge pages disabled
  - SMT-2

- Platforms
  - Linux in LPAR
  - Linux under z/VM (Version 6 Release 4.0, Service Level 1702)
  - Linux under KVM (Host Versions SLES 12 SP2 GM and SLES 12 SP3 RC1)

- Results are summarized covering
  - Throughput
  - CPU consumption

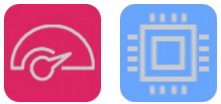
z14 measurement matrix overview

- Generally, all measurements were conducted without any z14 specific tunings
- Most workloads benefit of bigger cache sizes on z14

area	LPAR	z/VM guest	KVM guest
Kernel functions	X	X	X
Java	X		
Processor intense	X		
Scalability processor intensive	X		
Scalability file server	X	X	X
Scalability database server	X	X	
Networking OSA-Express5s			X
Networking OSA-Express6s	X		
Networking RoCE Express2	X		
Networking HiperSockets	X	X	
Networking Open vSwitch			X
Disk I/O FICON Express16s+ ECKD	X	X	X
Disk I/O FICON Express16s+ SCSI	X	X	X
Crypto CPACF	X		
Database	X		

Kernel functions benchmark description

Benchmark characteristics



- 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
  - 4 processors, 4 GiB memory

Kernel functions benchmark results

- Almost all performance data have improved. A lot show big improvements in high double-digit percentages

Measured operation	% improvement z14 to z13
simple syscall	32
simple read/write	21/27
select of file descriptors	19
signal handler	17
process fork	9
libc bcopy aligned L1 / L2 / L3 / L4 cache	46 / 11 / 11 / 63
libc bcopy unaligned L1 / L2 / L3 / L4 cache	52/ 51 / 55 / 23
memory bzero L1 / L2 / L3 / L4 cache	100 / 111 / 37 / 0
memory partial read L1 / L2 / L3 / L4 cache	20 / 37 / 13 / 14
memory partial read/write L1 / L2 / L3 / L4 cache	154 / 86 / 51 / 11
memory partial write L1 / L2 / L3 / L4 cache	398 / 317 / 133 / 16
memory read L1 / L2 / L3 / L4 cache	21 / 17 / 18 / 8
memory write L1 / L2 / L3 / L4 cache	243 / 222 / 176 / 29
Mmap read L1 / L2 / L3 / L4 cache	26 / 20 / 26 / 21
Mmap read open2close L1 / L2 / L3 / L4 cache	39 / 234 / 54 / 37
Read L1 / L2 / L3 / L4 cache	12 / 10 / 11 / -3
Read open2close L1 / L2 / L3 / L4 cache	11 / 11 / 13 / -2
Unrolled bcopy unaligned L1 / L2 / L3 / L4 cache	77 / 77 / 55 / -9
Unrolled partial bcopy unaligned L1 / L2 / L3 / L4 cache	145 / 39 / 26 / -3
Mappings L2 / L3 / L4 cache	24 / 54 / 38

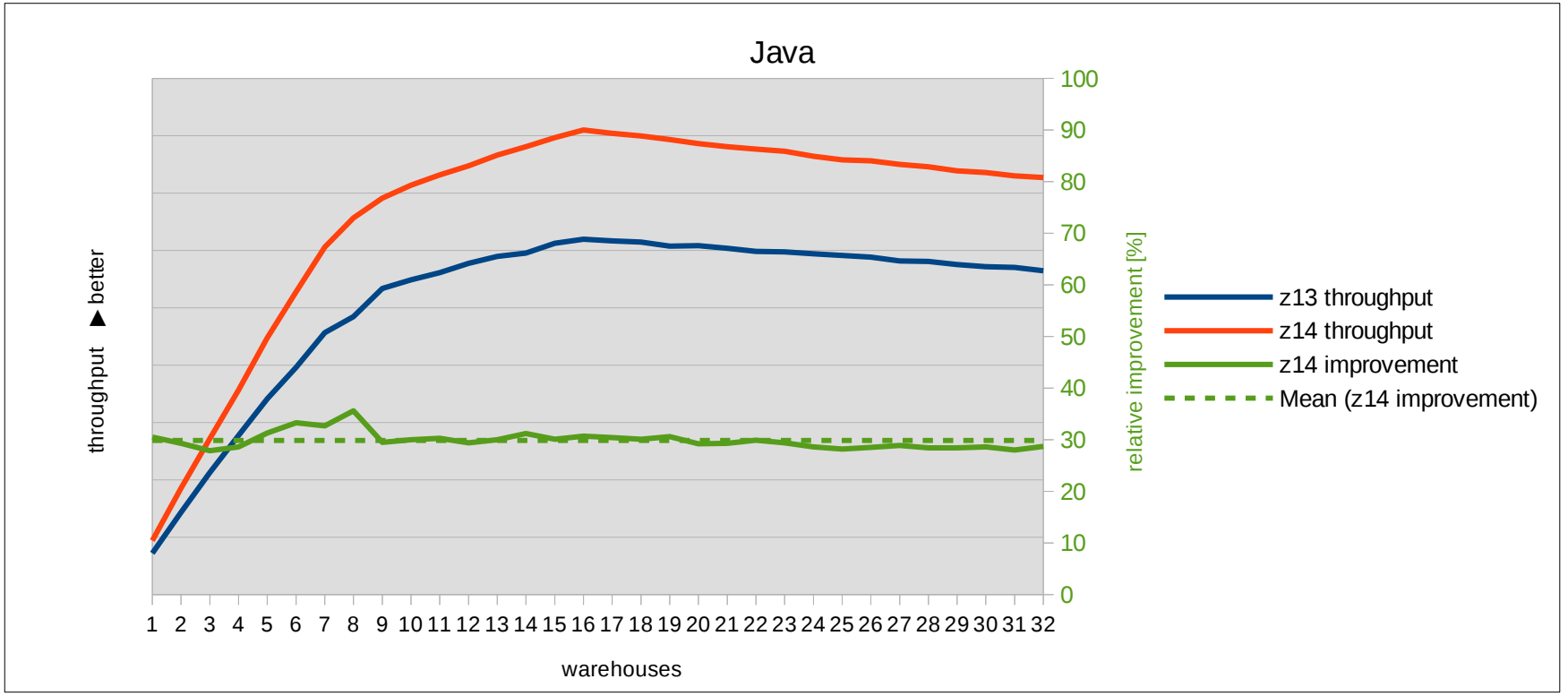
Java benchmark description

Benchmark characteristics

- Industry standard benchmark
  - Evaluates the performance of server-side Java
  - Exercises
    - Java Virtual Machine (JVM)
    - Just-In-Time compiler (JIT)
    - Garbage collection
    - Multiple threads
    - Simulates real-world applications including XML processing or floating point operations
  - Can be used to measure performance of processors, memory hierarchy and scalability
- Configuration
  - 16 processors, 4 GiB memory, 1 JVM, 2 GiB max. heap size

Java benchmark results

- Java measurements show about 30% throughput increase at lower CPU consumption

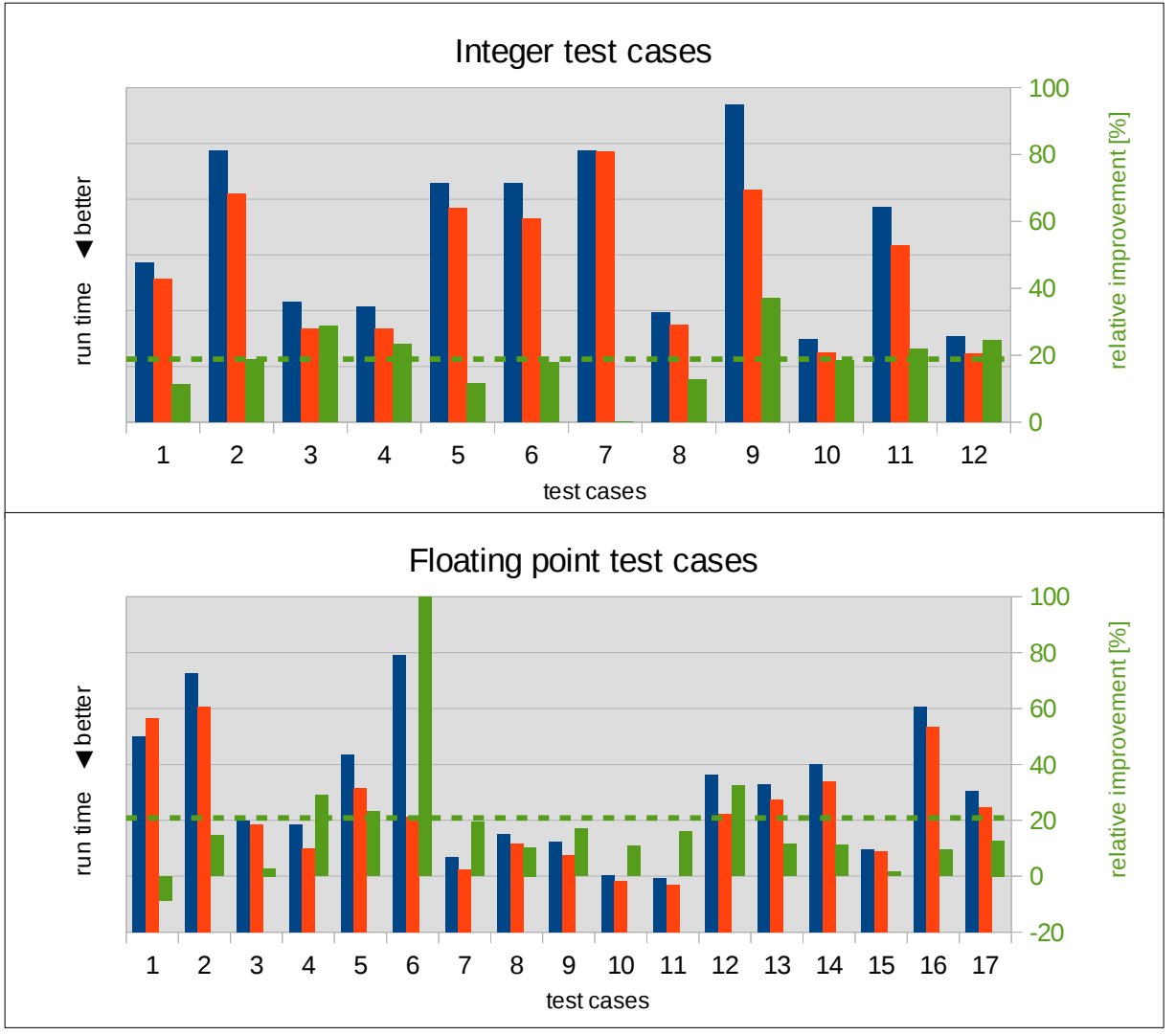


Processor benchmark description

- Industry standard benchmark
  - 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 single-thread performance
    - evaluate compile options
    - optimize the compiler's code generation for a given target system
- Configuration
  - 1 processor, 2 GiB memory, executing one test case at a time

Processor benchmark results

- Compiler benchmarks show about 20% throughput increase
- Same for both floating point and integer test cases (single-threaded workloads)



Scalability benchmark description

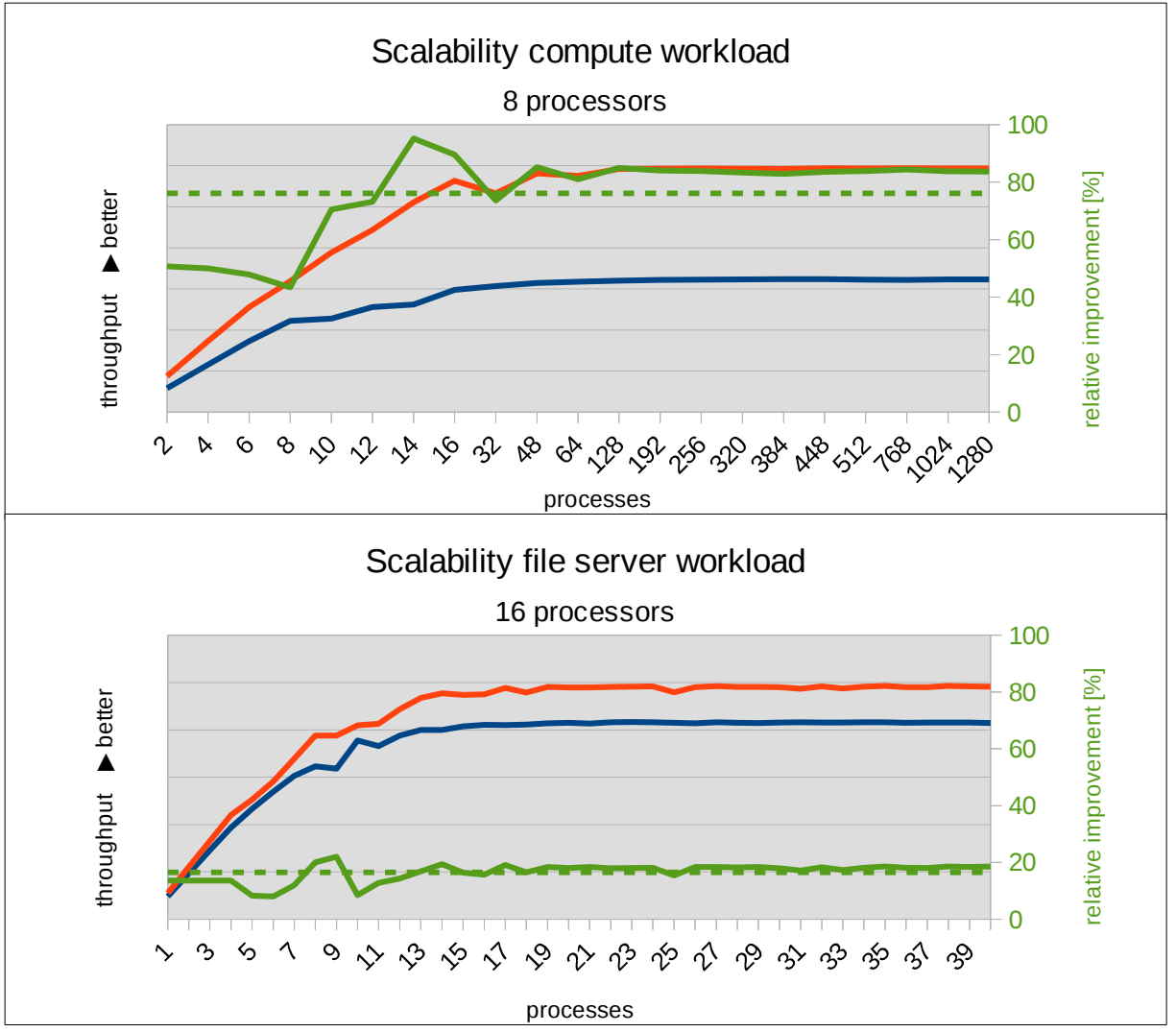
Benchmark characteristics

- Simulates multi-user scenarios
- Workload patterns describe system call ratios (patterns can be more inter-process-communication, disk or calculation intensive)
- The benchmark run
  - Starts with one job, continuously increases that number
  - Overall throughput usually increases until #threads ≈ #processors
  - Then threads are further increased until a drop in throughput occurs
  - Scales up to thousands of concurrent threads stressing the same components

- Measures the amount of jobs per minute a single thread and all the threads can achieve
- Configuration
  - 2, 8, 16 processors, 4 GiB memory
  - Using a journaled file system on an xpram device (not I/O bound)
  - Using fserver, new-db and compute workload patterns

Scalability benchmark results

- Computational workload shows significant performance gains of about 75%
- Database and file server workloads show solid improvements in the range of 20%



Network benchmark description

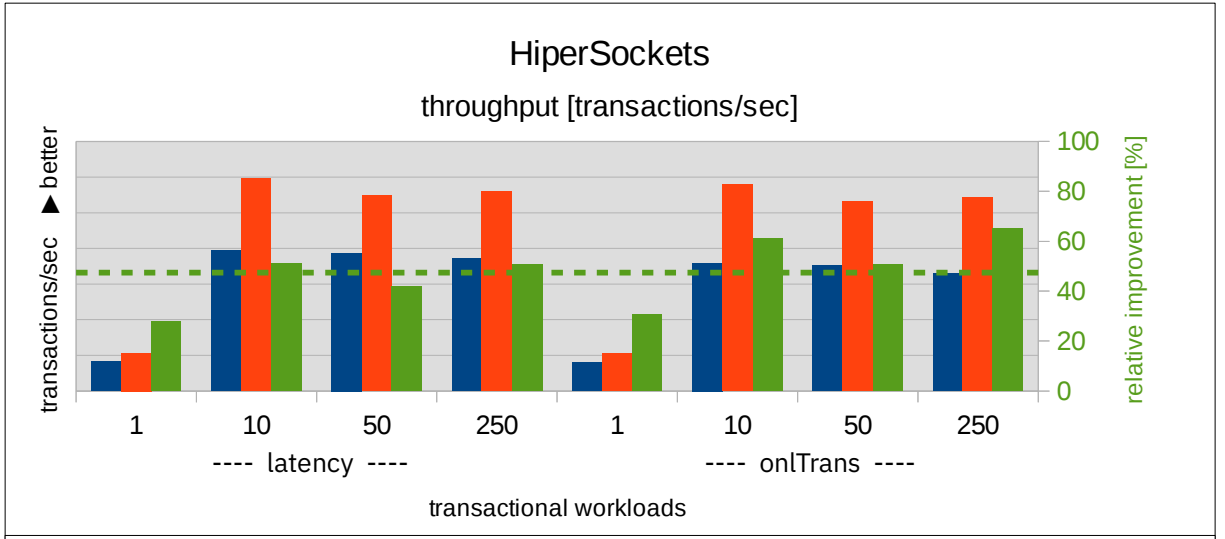
Benchmark characteristics

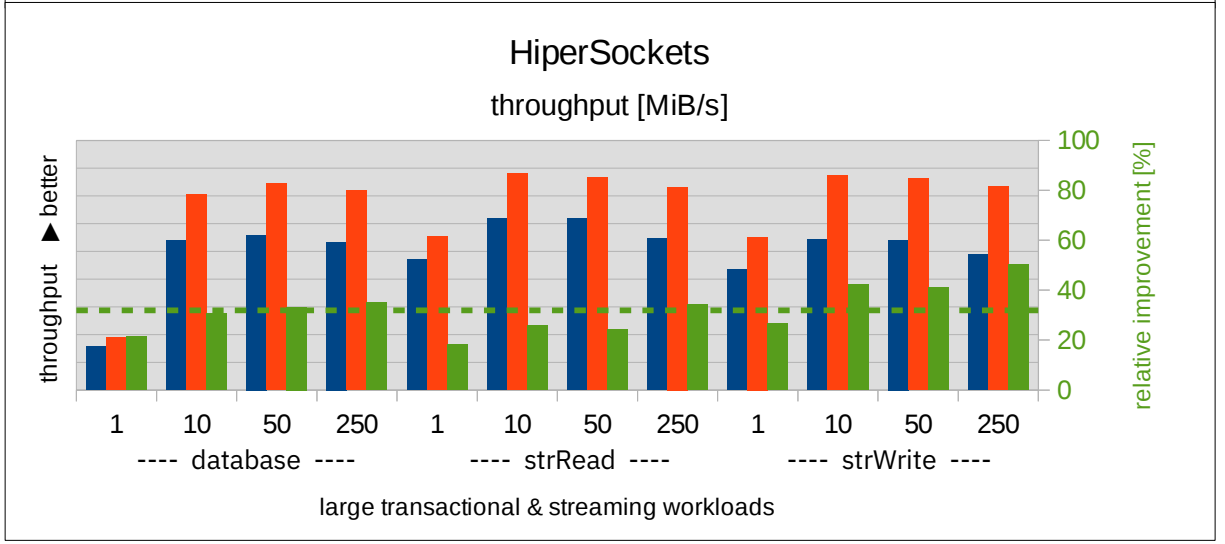
- Network performance tool that supports modeling
- Transactional workloads
  - “latency” Simulating low latency keep-alives (request 1 Byte, response 1 Byte)
  - “onlTrans” Simulating online transactions (request 200 Bytes, response 1000 Bytes)
- Large transactional & streaming workloads
  - “database“ Simulating database query (request 200 Bytes, response 30 KiB)
  - “strRead” Simulating incoming file transfers
  - “strWrite” Simulating outgoing file transfers
  - Simulated file size of 30 KiB for both streaming read and write
- All tests are done with 1, 10, 50, and 250 simultaneous connections

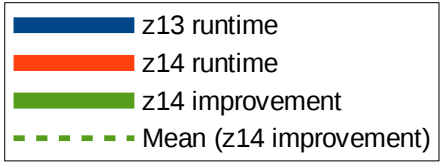
- All that across on multiple connection types
  - LPAR - LPAR
  - LPAR - z/VM Guest / z/VM Guest - z/VM Guest
  - LPAR - KVM Guest / KVM Guest - KVM Guest
  - Physical and virtual connections
  - HiperSockets, OSA-Express, RoCE Express, Open vSwitch (KVM)
  - MTU sizes 1500 and 9000
- Configurations
  - 4 processors, 4 GiB memory on each side

Network benchmark results (1)

- HiperSockets
  - With z14, throughput noticeably improves about 40% in average







Network benchmark results (2)

- OSA-Express5s vs. OSA-Express6s
  - Slight improvements for OSA-Express6s (z14)
- RoCE Express vs. RoCE Express2
  - RoCE Express2 (z14) outperforms RoCE Express (z13) in all streaming workloads and transactional workloads with huge data sizes
  - Processor consumption for RoCE Express2 is significantly higher than for the RoCE Express. This is partially the cost of increased performance

Disk benchmark description

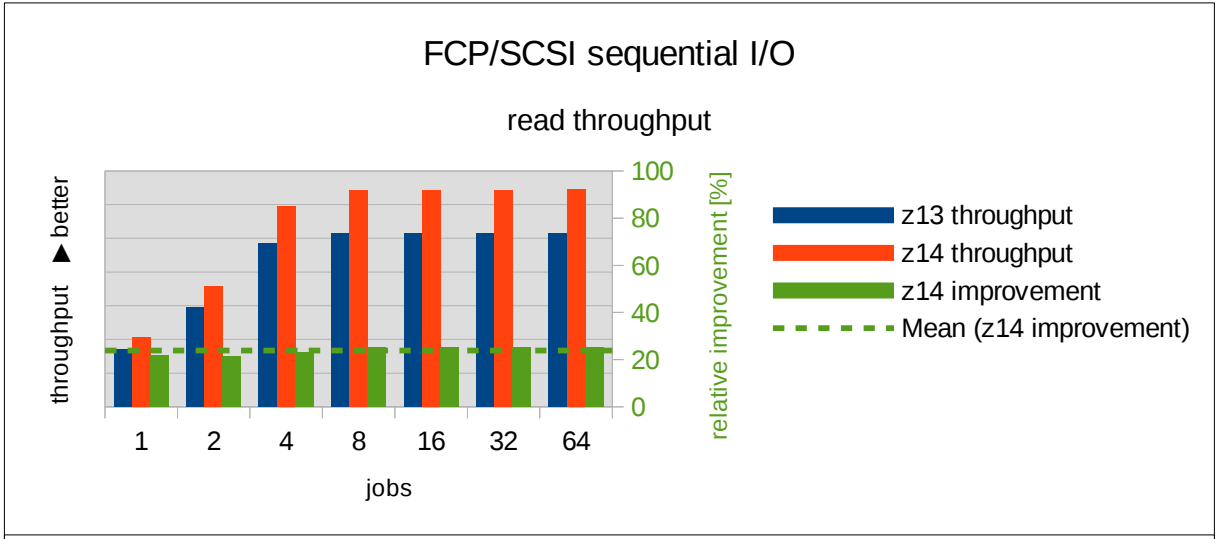
Benchmark characteristics

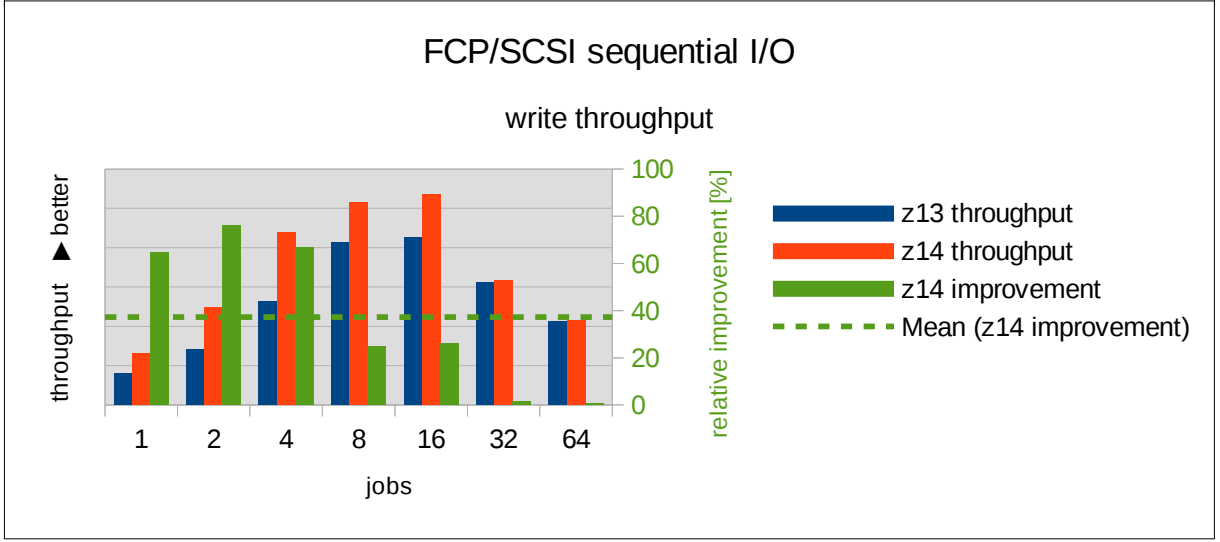
- I/O testing tool for benchmarking and hardware verification / stress I/O devices
  - Provides information regarding throughput, latencies, system utilization and much more
- Configuration
  - 8 processors, 2 GiB memory, 8 FICON channels

- Scenarios
  - FICON/ECKD
  - FCP/SCSI multipath
  - Scaling processes and disks: 1, 2, 4, 8, 16, 32, 64
  - Sequential read/write with 128 KiB record size
  - Random read/write with 8 KiB record size
  - Standard I/O using page cache
  - Direct I/O
  - Async I/O
  - Striped logical volume

Disk benchmark results (1)

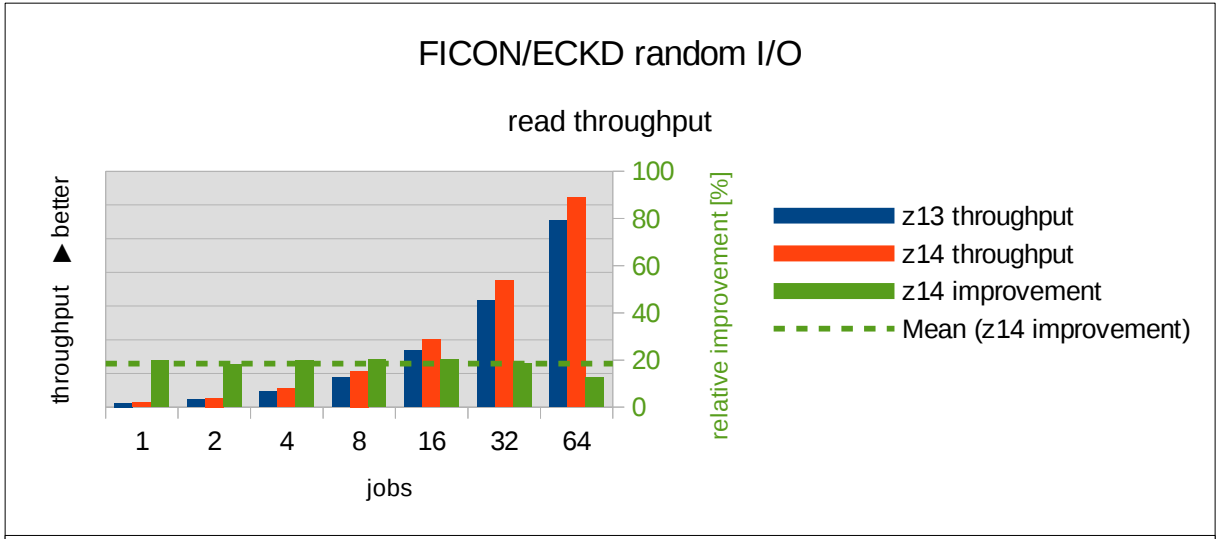
- FCP/SCSI
  - Significant improvements of 20-40% in average for sequential I/O workloads
  - Random I/O workloads about 10% improvement [no chart]
  - Significant CPU consumption savings
  - Full 16G FC line speed can be utilized

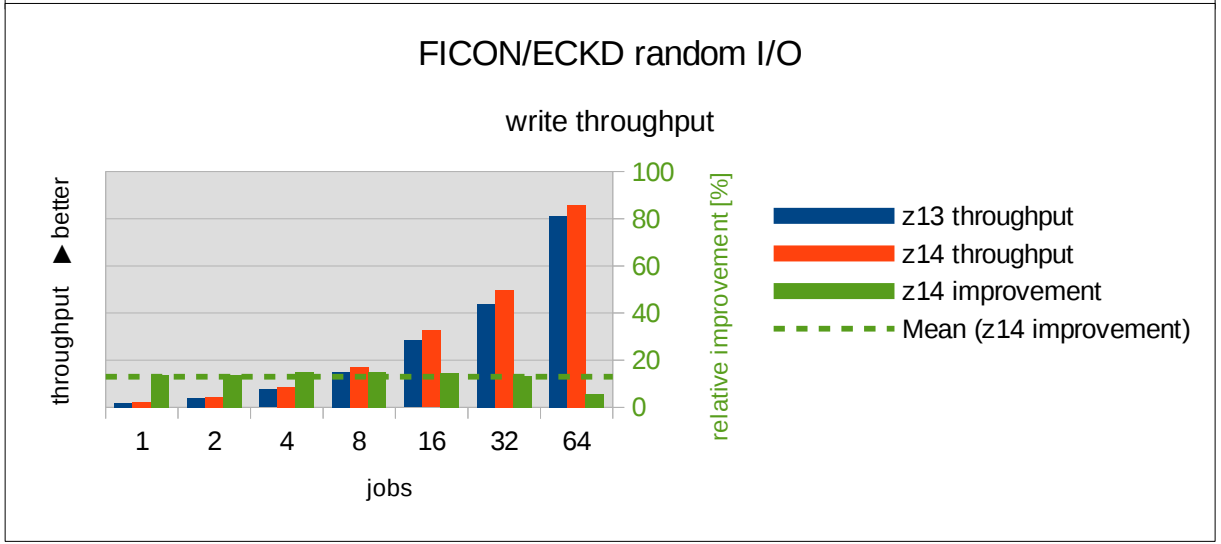




Disk benchmark results (2)

- FICON/ECKD
  - About 15-20% average improvement for random I/O workloads
  - Some improvements for sequential I/O workloads [no chart]
  - Significant CPU consumption savings





Crypto benchmark description

Benchmark characteristics

- Industry standard test
  - Using libcrypto – the crypto library of OpenSSL, used by many open source projects (e.g. OpenSSH, Apache mod_ssl, Node.js, PHP, PostgreSQL, MongoDB EE, Ruby)
  - Encrypting a maximum possible amount of data in a given time slot
  - Calculating throughput and normalized CPU consumption
- Focus on symmetric cryptography
  - Exploiting the CPACF facility
  - AES standard, 3 modes selected: GCM, XTS, CBC
  - Various key sizes (128, 256) and payload sizes (256 B, 1024 B, 8192 B) measured
- Configuration
  - 2 processors, 2 GiB memory

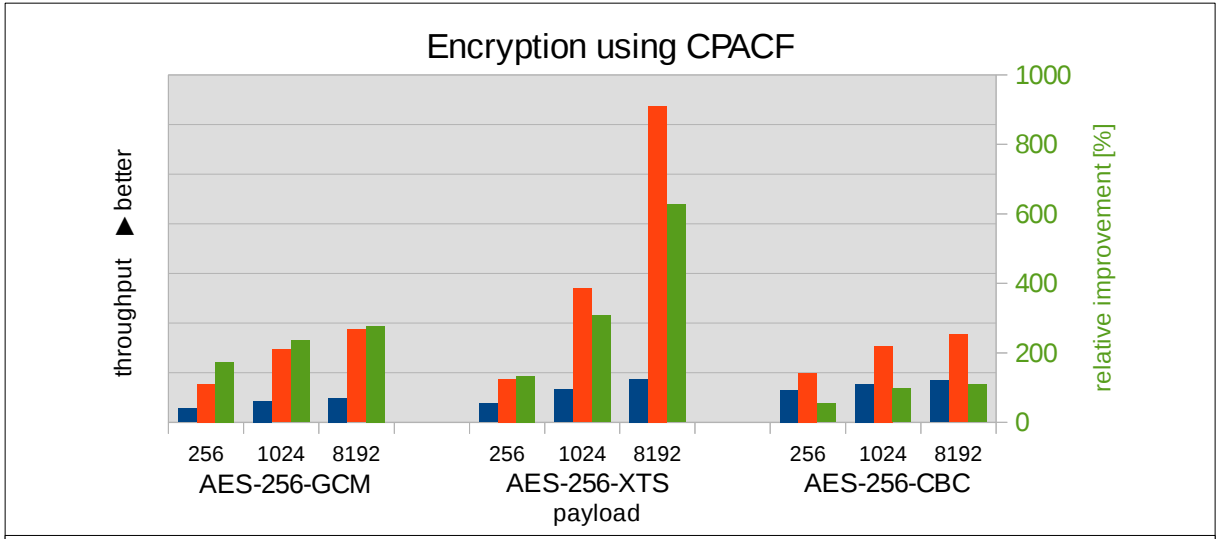
Crypto benchmark results (1)

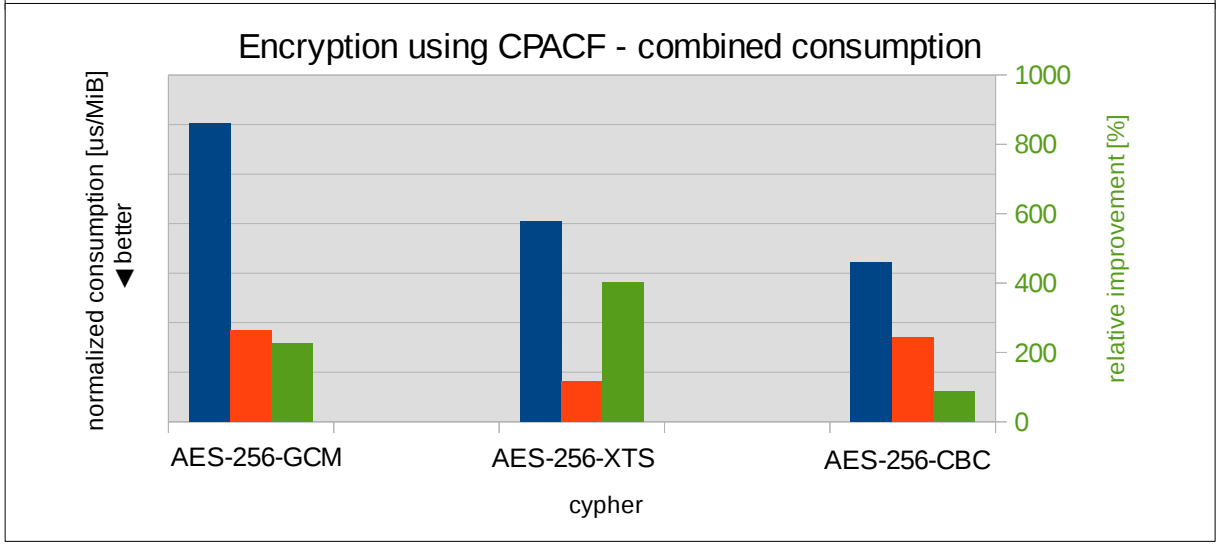
- Generally, CPACF exploitation introduces significant throughput and consumption improvements, especially with larger payload sizes
- Crypto benchmarks
  - Symmetric ciphers show significant improvements for both throughput and CPU consumption
    - Data in flight (with GCM, CBC) shows a solid performance gain
    - Data at rest/disk encryption (with XTS) benefits significantly with up to 600% improvement
    - Recent OpenSSL versions contain similar performance improvements for GCM
  - Asymmetric ciphers and SSL/TLS show slight improvements for both throughput and CPU consumption, mostly in software
  - Crypto Express6S toleration mode to come with SLES 12 SP3
  - Hash functions: slight performance gains

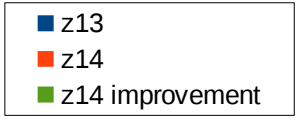
Crypto benchmark results (2)

- Detailed results for symmetric encryption

  - Up to 600% relative improvement with XTS, major cipher for disk encryption!







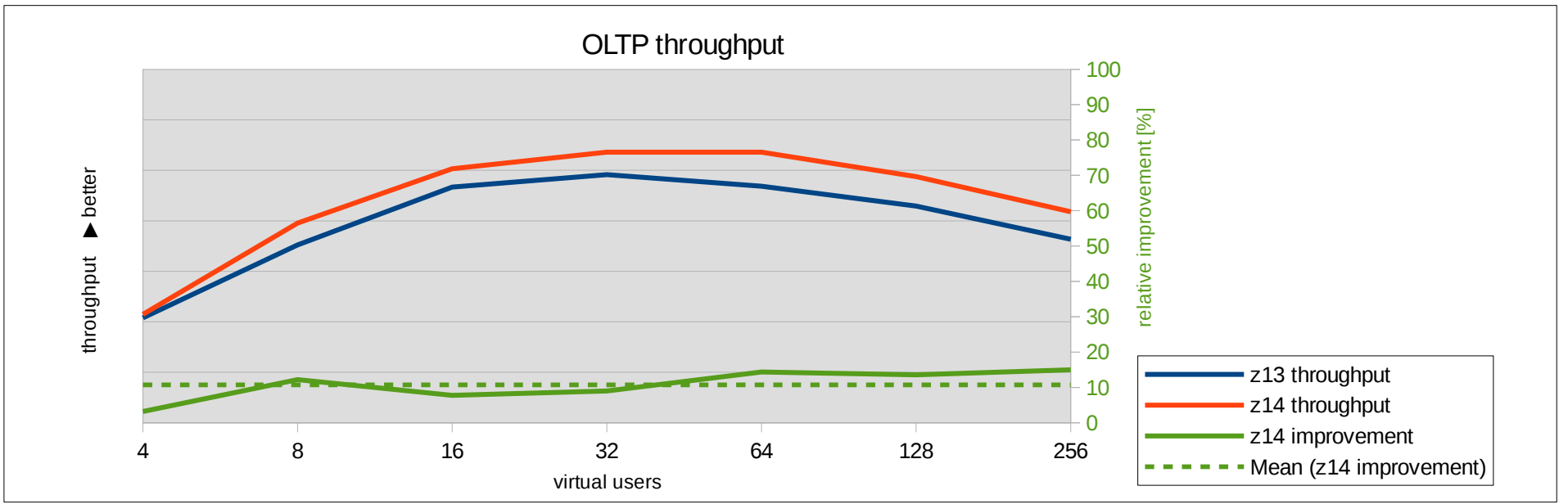
Database benchmark description

Benchmark characteristics

- Database benchmark executing online transaction processing
  - User-scaling tests with up to 256 parallel virtual database users
  - Using customer-like environment with significant disk I/O and CPU load
    - Database disk I/O for test: 180 MiB/s read, 200 MiB/s write
    - CPU consumption up to 100% utilization
- Configuration
  - 4 processors, 64 GiB memory
  - Database running in a Docker container
    - 256 GiB database
  - LUKS-encrypted FCP/SCSI data pool
    - CPACF supports strong encryption with AES-256 XTS and SHA-256

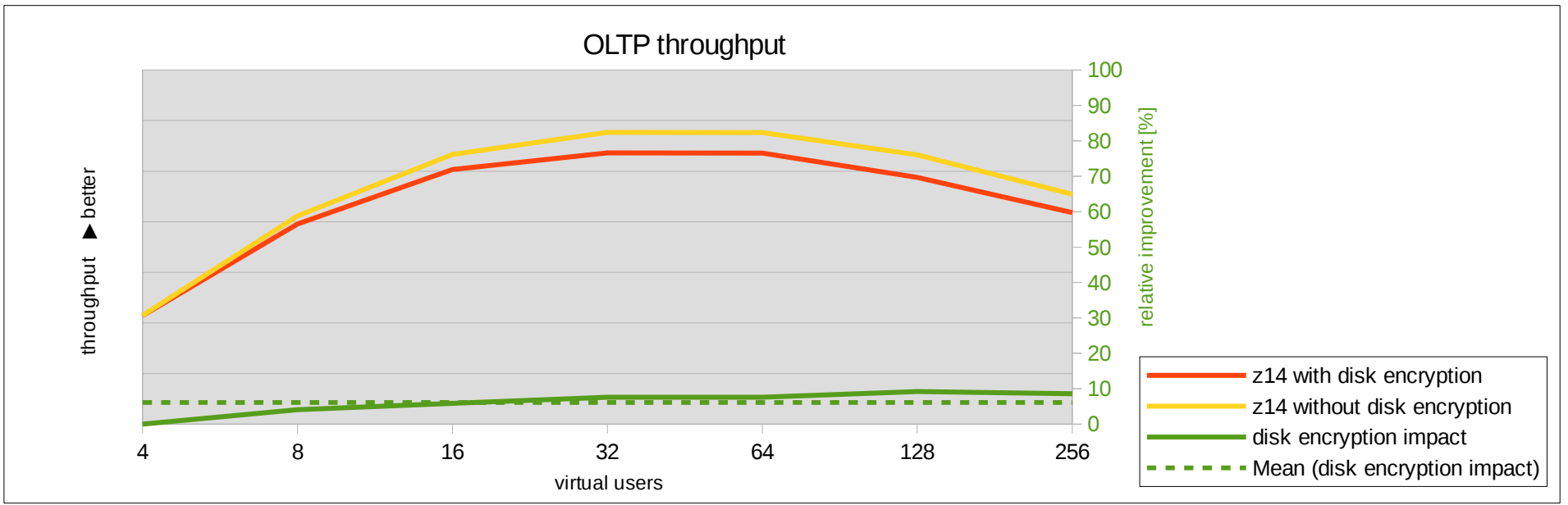
Database benchmark results (1)

- Workload scales well towards very high database load levels
- Up to 15% database throughput increase



Database benchmark results (2)

- Mean disk encryption impact on database throughput is about 7%
- CPACF greatly assists Linux disk encryption with LUKS



z/VM guest performance results

- Measurements here comprise a subset of those used in LPAR
  - Kernel benchmark
  - Scalability: File server workload only
  - Network
    - OSA-Express5s vs. OSA-Express6s
    - HiperSockets
    - VSWITCH
  - Disk I/O
- Results are comparable to those in LPAR with only slight differences in some detailed cases

KVM guest performance results (1) – Kernel & file server benchmarks

- Kernel benchmark
  - KVM results compare well to LPAR
    - Almost all performance data have improved
    - A lot show a big performance gain in double-digit percentages
    - Significant improvements in memory-write benchmarks
- File server workload scalability benchmark
  - More than 15% improvements in throughput (nearly as good as LPAR)
  - About 15% reduction in CPU consumption

KVM guest performance results (2) – Network benchmark (KVM-specific description)

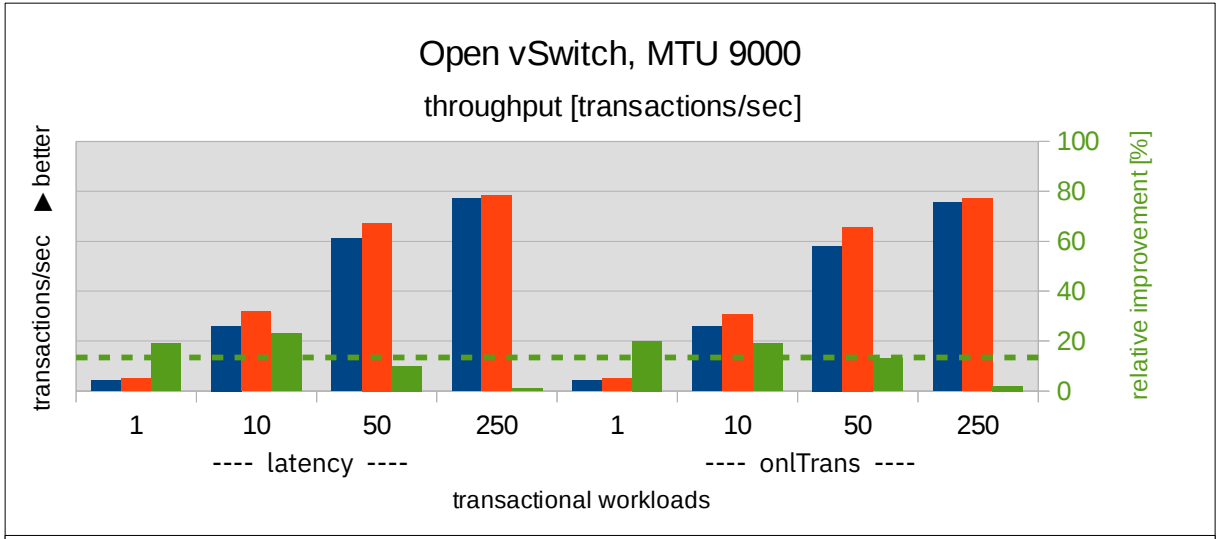
- LPAR - KVM Guest
  - Single KVM guest, connection via OSA-Express5s
  - 2 different setups for KVM guest interface attachment
    - MacVTap
    - Open vSwitch
- KVM Guest - KVM Guest
  - 2 guests in same KVM host, connected via Open vSwitch
- Configurations
  - 8 processors (for both LPAR and KVM guest)
  - 4 GiB memory on each side

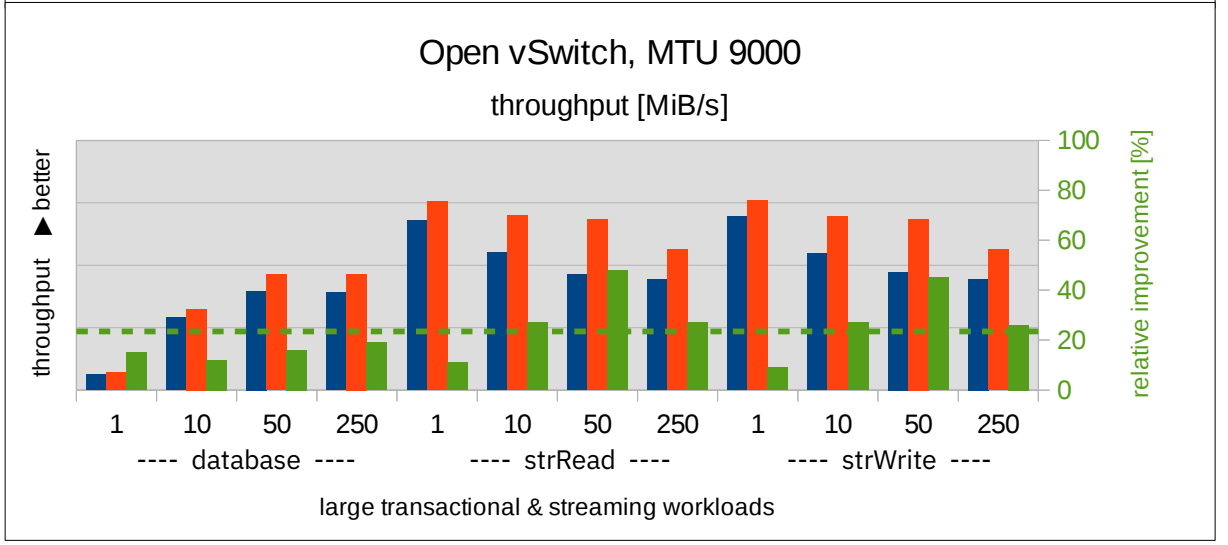
KVM guest performance results (3) – Network benchmark results

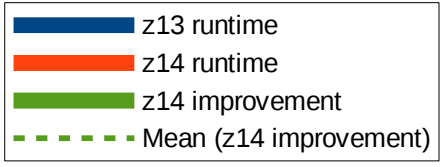
- LPAR - KVM Guest
  - Nearly 10% average throughput improvement for all transactional workloads for both MacVTap and Open vSwitch (more than 15% for single-connection scenario)
  - About 10% throughput improvement for streaming and large transactional workloads with small MTU size for both MacVTap and Open vSwitch (no difference for large MTU size: OSA-Express5s line speed was reached on z13 already)
- KVM Guest - KVM Guest
  - Nearly 15% average throughput improvement for all transactional workloads (20% for single-connection scenario)
  - More than 20% average throughput improvement for streaming and large transactional workloads and both MTU sizes

KVM guest performance results (4) – Network benchmark results

- KVM Guest - KVM Guest
  - About 20% improvement in streaming workloads



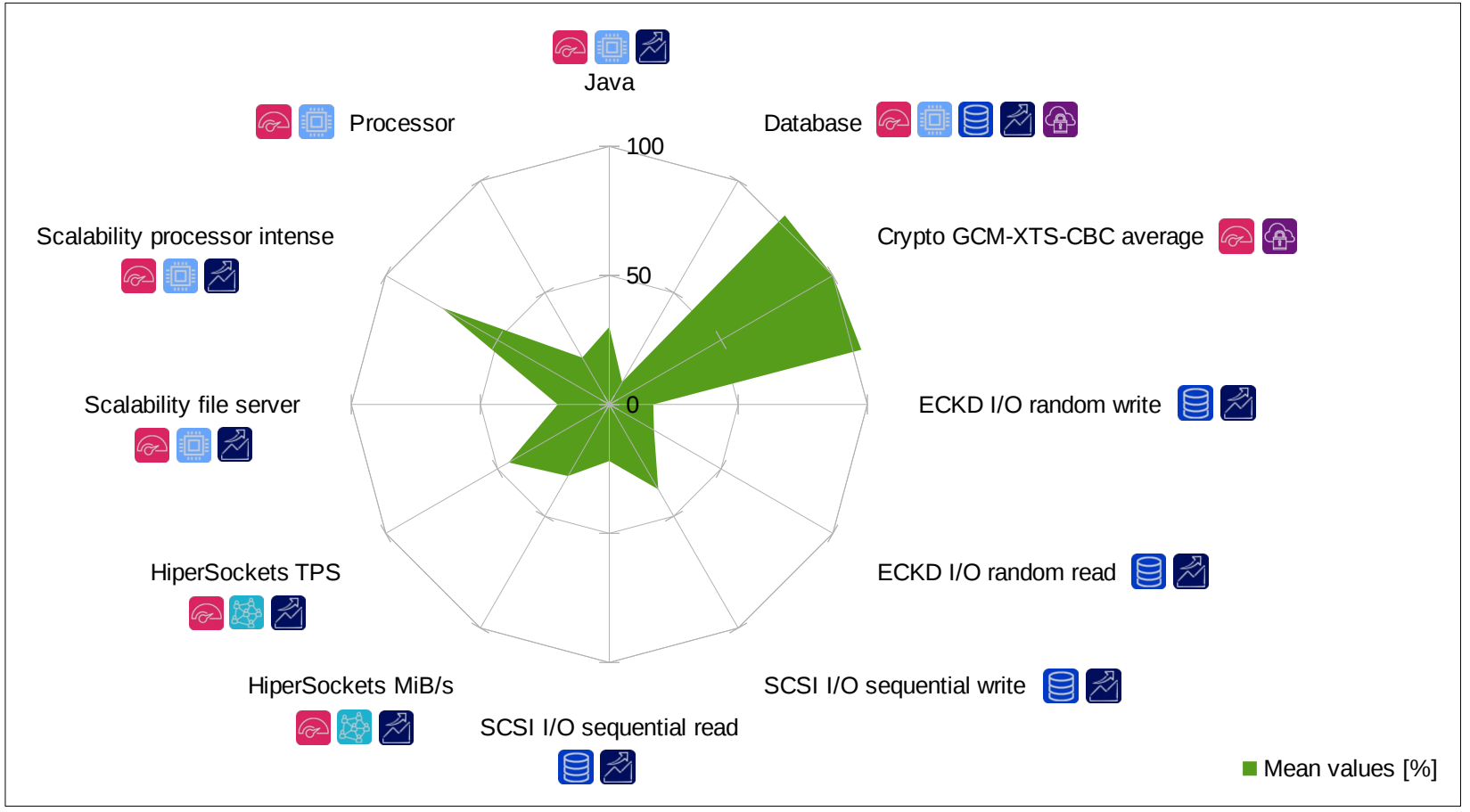




KVM guest performance results (5) – Disk I/O benchmark

- Disk I/O
  - Guest disk attachment for FICON/ECKD and FCP/SCSI via s390-ccw-virtio
  - Using qemu raw, non-host-cached I/O, 64 disks and 8 I/O threads
  - KVM throughput improvement with z14 comparable to LPAR
    - FCP/SCSI
      - Most significant throughput improvements for sequential I/O workloads, especially for sequential write
    - FICON/ECKD
      - Most significant throughput gain for random I/O workloads, in particular for random read
  - Overall about 10% reduction in CPU consumption

Summary of shown relative performance improvements with z14 vs. z13



Thank You

Barbara Mundle
Performance Analyst

Linux on Z Performance Evaluation

Research & Development
Schönaicher Strasse 220
71032 Böblingen, Germany

barbara.mundle@de.ibm.com

Jens Markwardt
Performance Analyst

Linux on Z Performance Evaluation

Research & Development
Schönaicher Strasse 220
71032 Böblingen, Germany

jens.markwardt@de.ibm.com

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

Live Virtual Classes for z/VM and Linux http://www.vm.ibm.com/education/lvc/

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

Acknowledgements

- Marc Beyerle
- Dr. Juergen Doelle
- Horst Hartmann
- Mario Held
- Nils Hoppmann
- Martin Kammerer
- Dominik Klein
- Dr. Stefan Reimbold
- Danijel Soldo
- Thomas Weber
- IBM z14 Technical Leadership Library, July 17, 2017 Announcement
- z14 Imagery, July 18, 2017
- IBM z Systems and DS8880 – Integrated by Design: Why it Matters

Trademarks