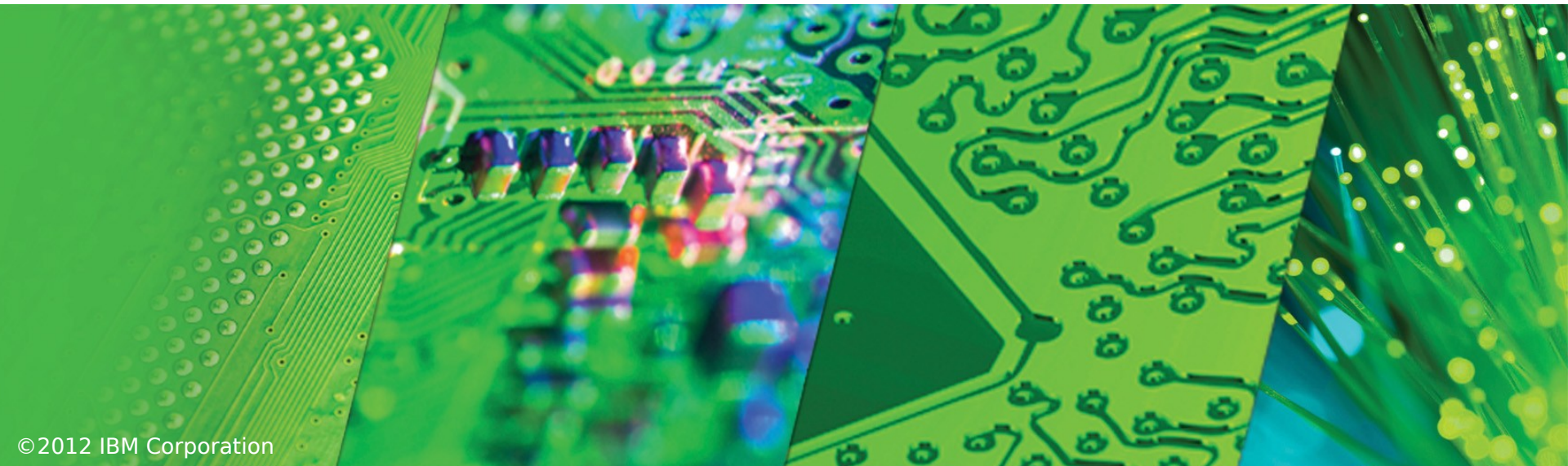
2012
IBM System z Technical University

Enabling the infrastructure for smarter computing

# SLES11 SP2 performance evaluation

**zLG15**

Mario Held, Christian Ehrhardt



# Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates in the United States, other countries, or both.

Red Hat, Red Hat Enterprise Linux (RHEL) are registered trademarks of Red Hat, Inc. in the United States and other countries.

SLES and SUSE are registered trademarks of Novell, Inc., in the United States and other countries.

SPEC and the "performance chart" SPEC logo are registered trademarks of the Standard Performance Evaluation Corporation.

Other product and service names might be trademarks of IBM or other companies.

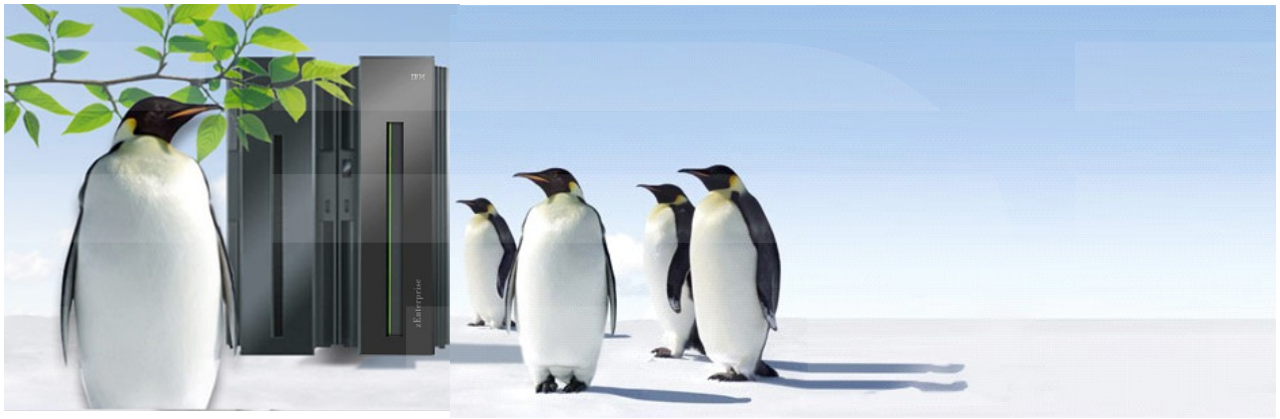
# Agenda

- Performance evaluation
  - ✓ System under test (SUT) description
  - ✓ Changes one should be aware of
- Performance evaluation summary
  - ✓ Improvements and degradations per area
  - ✓ Summarized comparison

# System under test (SUT)

- Hardware platform – System z10
  - ✓ FICON 8 Gbps
  - ✓ FCP 8 Gbps
  - ✓ HiperSockets
  - ✓ OSA Express 3 1GbE + 10GbE
- Software platform
  - ✓ z/VM 5.4
  - ✓ LPAR
- Storage – DS8300 (2107-922 )
  - ✓ FICON 8 Gbps
  - ✓ FCP 8 Gbps

- Hardware platform – System z196
  - ✓ FICON 8 Gbps
  - ✓ FCP 8 Gbps
  - ✓ HiperSockets
  - ✓ OSA Express 3 1GbE + 10GbE
- Software platform
  - ✓ z/VM 6.1
  - ✓ LPAR
- Storage – DS8800
  - ✓ FICON 8 Gbps
  - ✓ FCP 8 Gbps



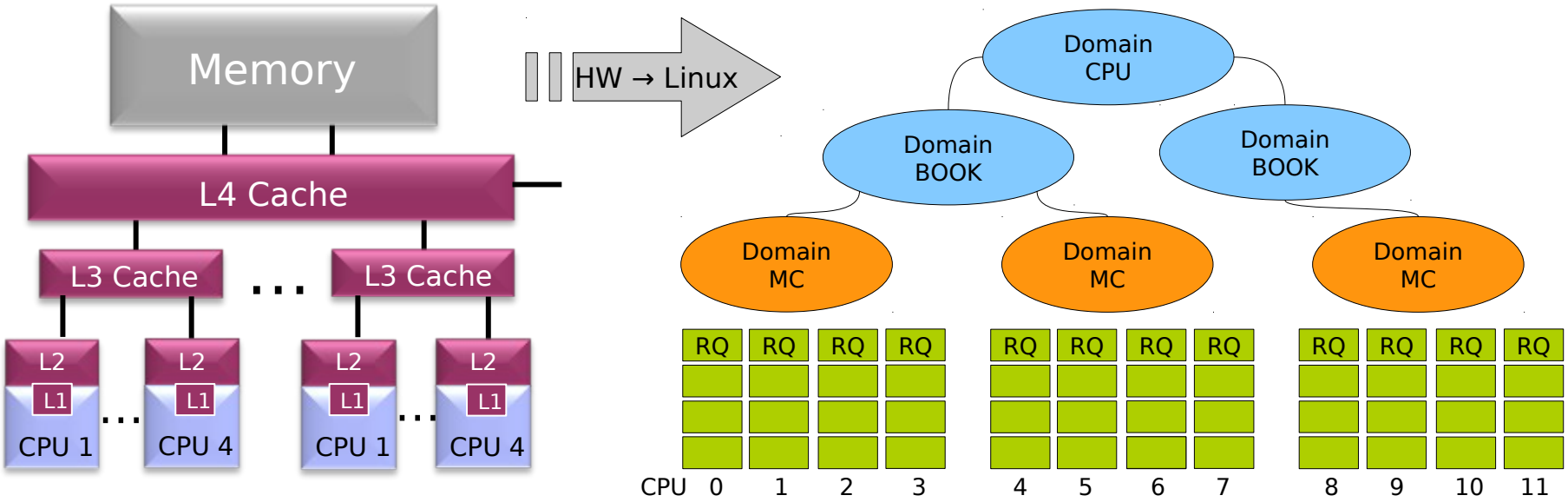
# Compared Distribution Levels

- Compared Distribution Levels
  - ✓ SLES 11 SP1 (2.6.32.12-0.6-default)
  - ✓ SLES 11 SP2 (3.0.13-0.27-default)
- Measurements
  - ✓ Base regression set covering most customer use cases as good as possible
  - ✓ Focus on areas where performance issues are more likely
  - ✓ Just the top level summary, based on thousands of comparisons
  - ✓ Special case studies for non-common features and setups
- Terminology
  - ✓ Throughput – “How much could I transfer once?”
  - ✓ Latency – “How long do I have to wait for event X?”
  - ✓ Normalized cpu consumption - “How much cpu per byte do I need?”

# Completely Fair Scheduler (CFS)

- Goals of CFS
  - ✓ Models “ideal, precise multi-tasking CPU”
  - ✓ Fair scheduling based on virtual runtime
- Changes you might notice when switching from O(1) to CFS
  - \+ Lower response times for I/O, signals, …
  - \+ Balanced distribution of process time-slices
  - \+ Improved distribution across processors
  - \- Shorter consecutive time-slices
  - \- More context switches
- Improved balancing
  - ✓ Topology support is now on by default
    - ▫ Can be switched via the topology=on|off kernel parameter
  - ✓ This makes the scheduler aware of the cpu hierarchy
- You really get something from fairness as well
  - ✓ Improved worst case latency and throughput
  - ✓ By that CFS can ease QoS commitments

# Topology of a zEnterprise System

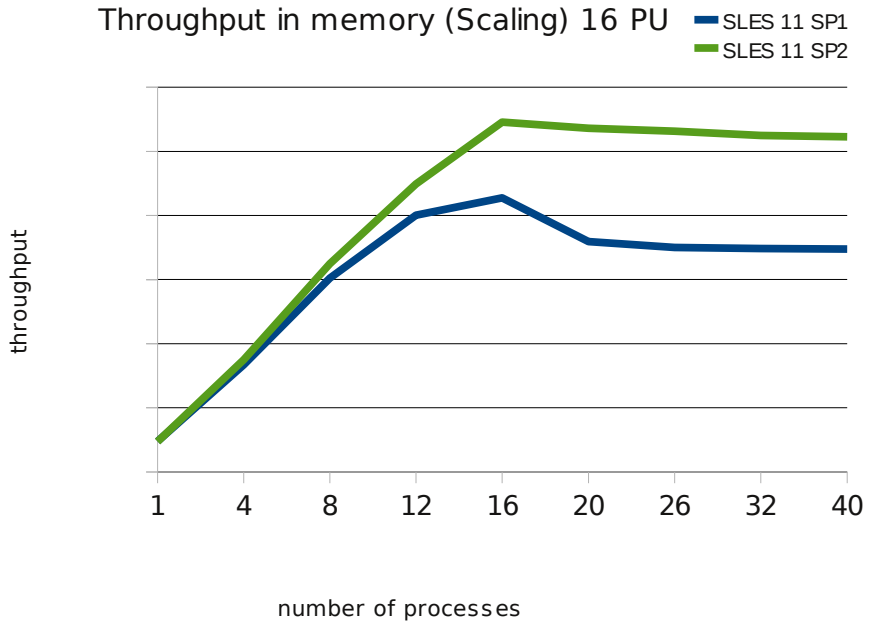



- Topology feature
  - ✓ Ability to group (rec. ipc heavy loads) or spread (rec. cache hungry) loads
  - ✓ Unintended asymmetries now known to the system
  - ✓ **Off in z/VM Guests, since there is no virtual topology information**
- Tunable, but complex
  - ✓ /proc/sys/kernel/sched_* files contains tunables for decisions regarding request queues (▮)
  - ✓ /proc/sys/kernel/sched_domain/... provides options for the scheduling domains (▮/▮)

# Benchmark description - File system / LVM / Scaling

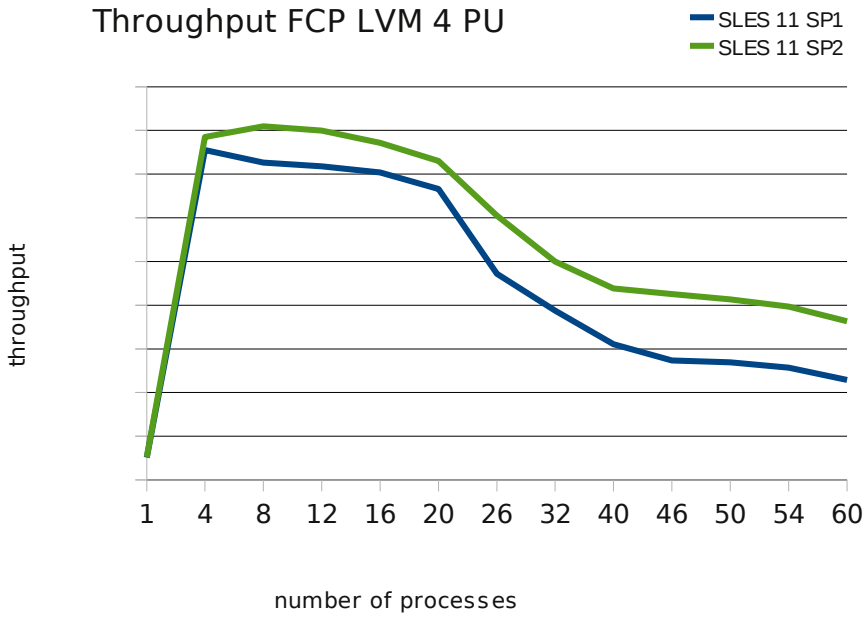
- Filesystem benchmark Dbench 3.0
  - ✓ Emulation of Netbench benchmark
  - ✓ Generates file system load on the Linux VFS
  - ✓ Does the same I/O calls like smbd server in Samba (without networking calls)
- Simulation
  - ✓ Workload simulates client and server (Emulation of Netbench benchmark)
  - ✓ Mixed file operations workload for each process: create, write, read, append, delete
  - ✓ Measures throughput of transferred data
  - ✓ Two setup scenarios
    - Scaling – Loads fits in cache, so mainly memory operations for scaling 2,4,8,16 CPUs, 8Gib Memory and scaling from 1 to 40 processes
    - Low main memory and LVM setup for mixed I/O LVM performance 8 CPUs, 2 GiB memory and scaling from 4 to 62 processes

# File System benchmark – Scaling Scenario




- Improved scalability for page cache operations
  - ✓ Especially improves large workloads
    - Saves cache misses of the load that runs primarily in memory
    - Lower cross process deviation improves QoS

# File system benchmark – LVM Scenario

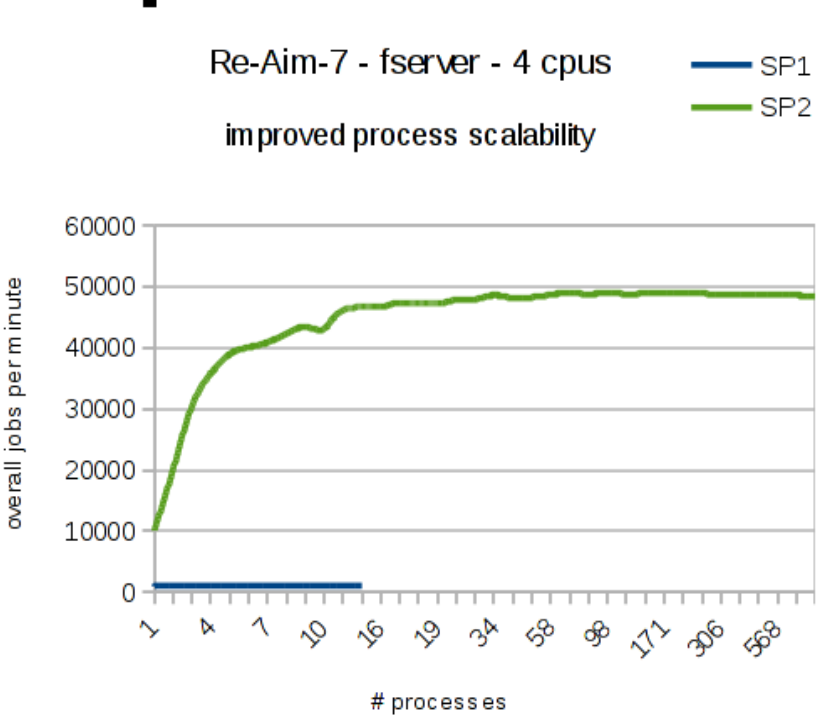


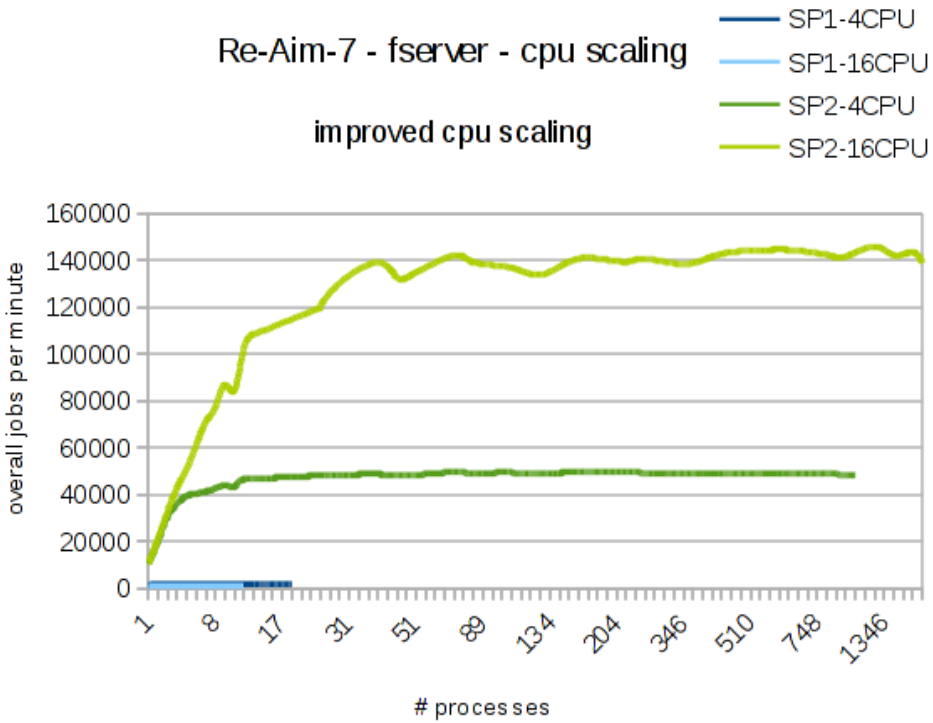

- Improved throughput for disk bound LVM setups as well
  - ✓ Especially improves heavily concurrent workloads
    - Fix in device mapper included which especially helps workloads with small I/Os
    - Saves a lot of CPU costs

# 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

# Improvements to file-system sync



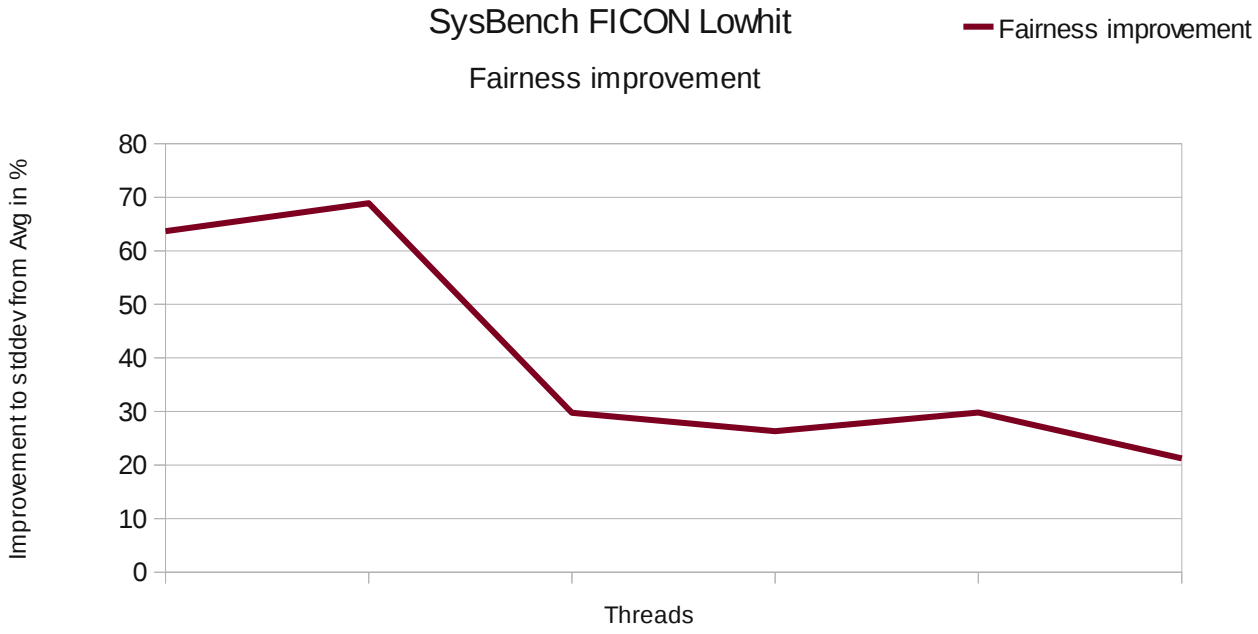




- The issue blocked process scaling (left) and CPU scaling (right)
- The sync call was broken, so scaling relying on it was almost non existent
  - ✓ Scales well in SP2 now with increasing number of processes
  - ✓ Fortunately for SP1 this system call is not one of the most frequently called ones

# Benchmark description – SysBench

- Scalability benchmark SysBench
  - ✓ SysBench is a multi-threaded benchmark tool for (among others) OLTP- database loads
  - ✓ Can be run read-only and read-write
  - ✓ Clients can connect locally or via network to the database
  - ✓ Database level and tuning is important
    - We use Postgres 9.0.4 with configuration tuned for this workload in our test
  - ✓ High/Low Hit cases resemble different real world setup cases with high or low cache hit ratios
- Our List of Setups
  - ✓ Scaling – read-only load with 2, 8, 16 CPUs, 8 GiB memory, 4GiB DB (High-Hit)
  - ✓ Scaling Net – read-only load with 2, 8, 16 CPUs, 8 GiB memory, 4GiB DB (High-Hit)
  - ✓ Scaling FCP/FICON High Hit ratio – read-write load with 8 CPUs, 8 GiB memory, 4GiB DB
    - RW loads still need to maintain the transaction log, so I/O is still important despite DB<MEM
  - ✓ Scaling FCP/FICON Low Hit ratio – read-write load with 8 CPUs, 4 GiB memory, 64GiB DB
    - This is also I/O bound to get the Data into cache TODO
  - ✓ All setups use
    - HyperPAV (FICON) / Mulitpathing (FCP)
    - Disk spread over the Storage Server as recommended + Storage Pool Striping
    - Extra Set of disks for the WAL (Transaction Protocol)

# SysBench – improved thread fairness

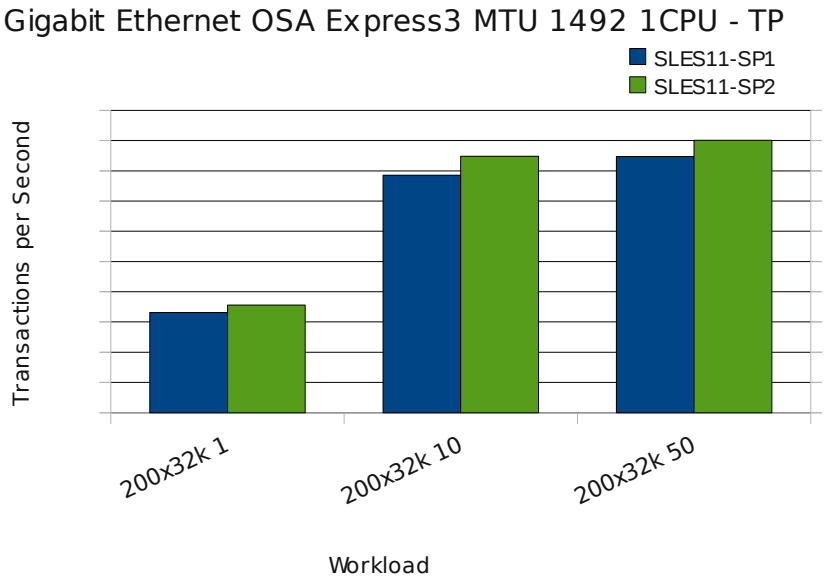


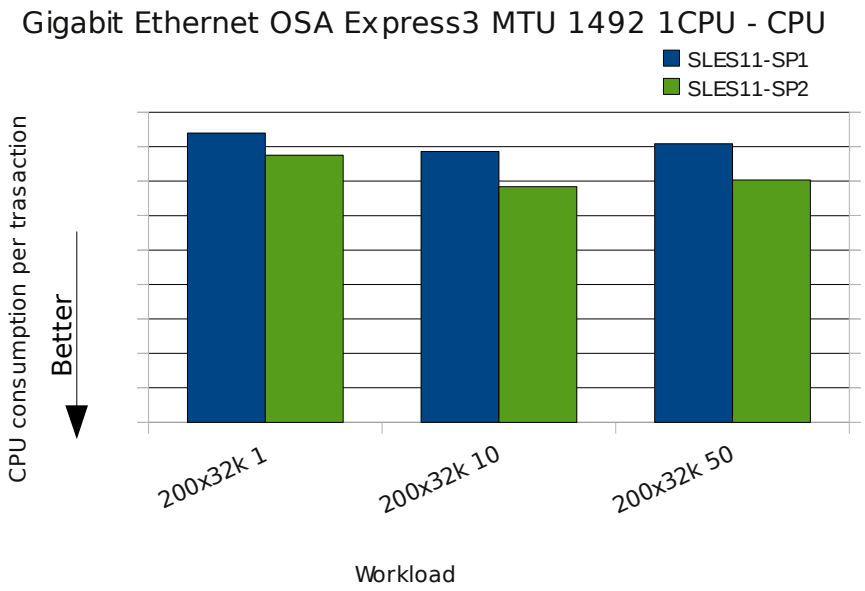

- Overall throughput stayed comparable
- But the fairness across the concurrent threads improved
  - ✓ Good to improve fair resource sharing without enforced limits in shared environments
  - ✓ Effect especially visible when the Database really has to go to disk (low hit scenario)
  - ✓ Can ease fulfilling QoS commitments

# 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
- All that across on multiple connection types (different cards and MTU configurations)

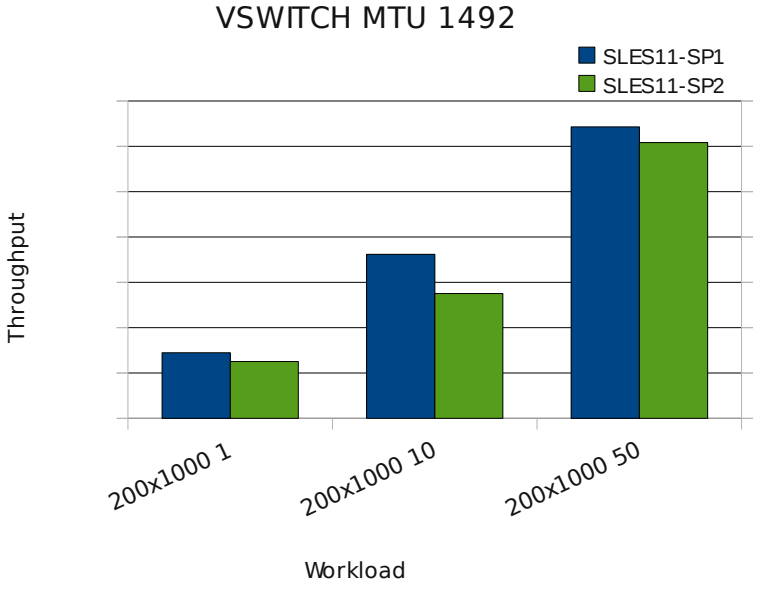
# Network I

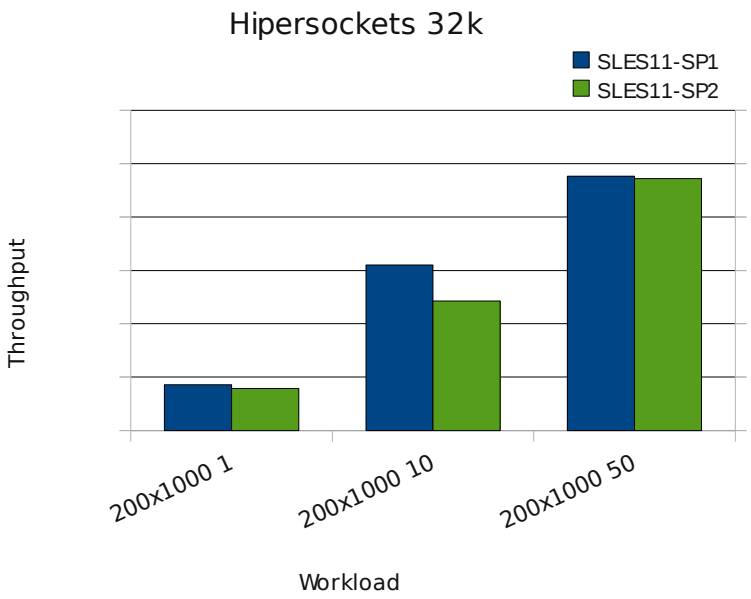



- Small systems gain an improvement in streaming throughput and cpu consumption
  - ✓ Systems being cpu-oversized always had to pay a price in terms of cpu consumption
  - ✓ Sometimes dynamic adjustment of your sizing can be an option, check out cpuplugd
    - A soon published paper about that can be found at http://www.ibm.com/developerworks/linux/linux390/perf/index.html
- Generic receive offload is now on by default
  - ✓ Further improves cpu consumption, especially for streaming workloads

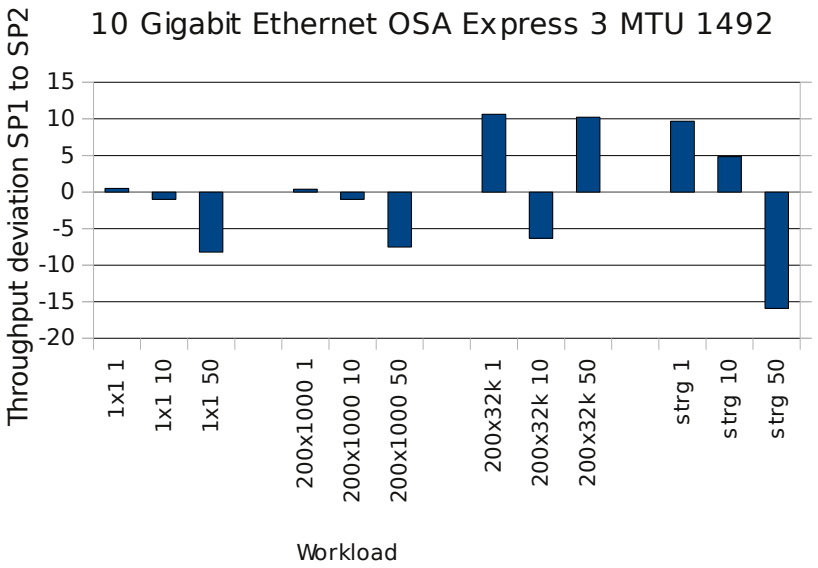
# Network II

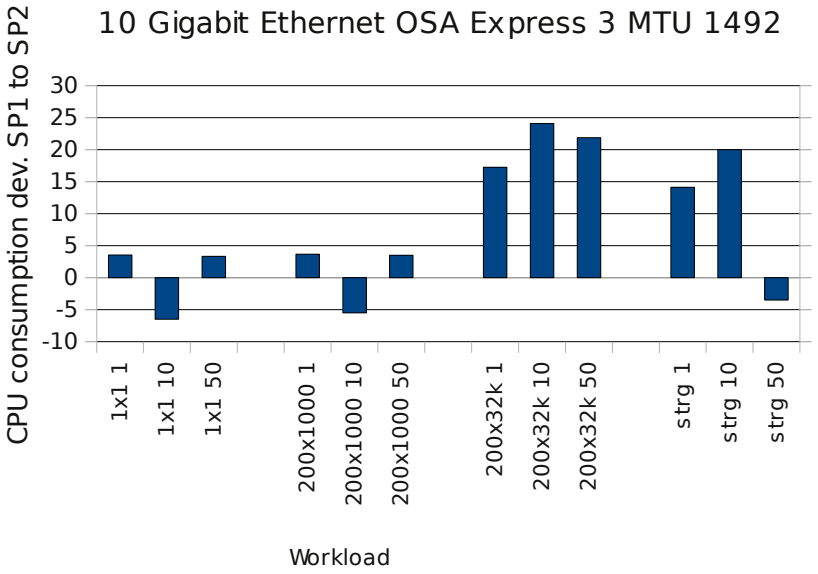



- Pure virtual connections degraded by 5 to 20%
  - ✓ Affects approximately half of the workload scenarios (smaller payloads are more in trouble)
  - ✓ Affects virtual VSWITCH and Hipersocket connections
- Some good messages mitigating that degradations
  - ✓ The reported overhead caused in the virtualization layers improved, so scaling will be better
  - ✓ Smaller degradations with larger mtu sizes
  - ✓ Effect smaller on z196 than on z10

# Network III



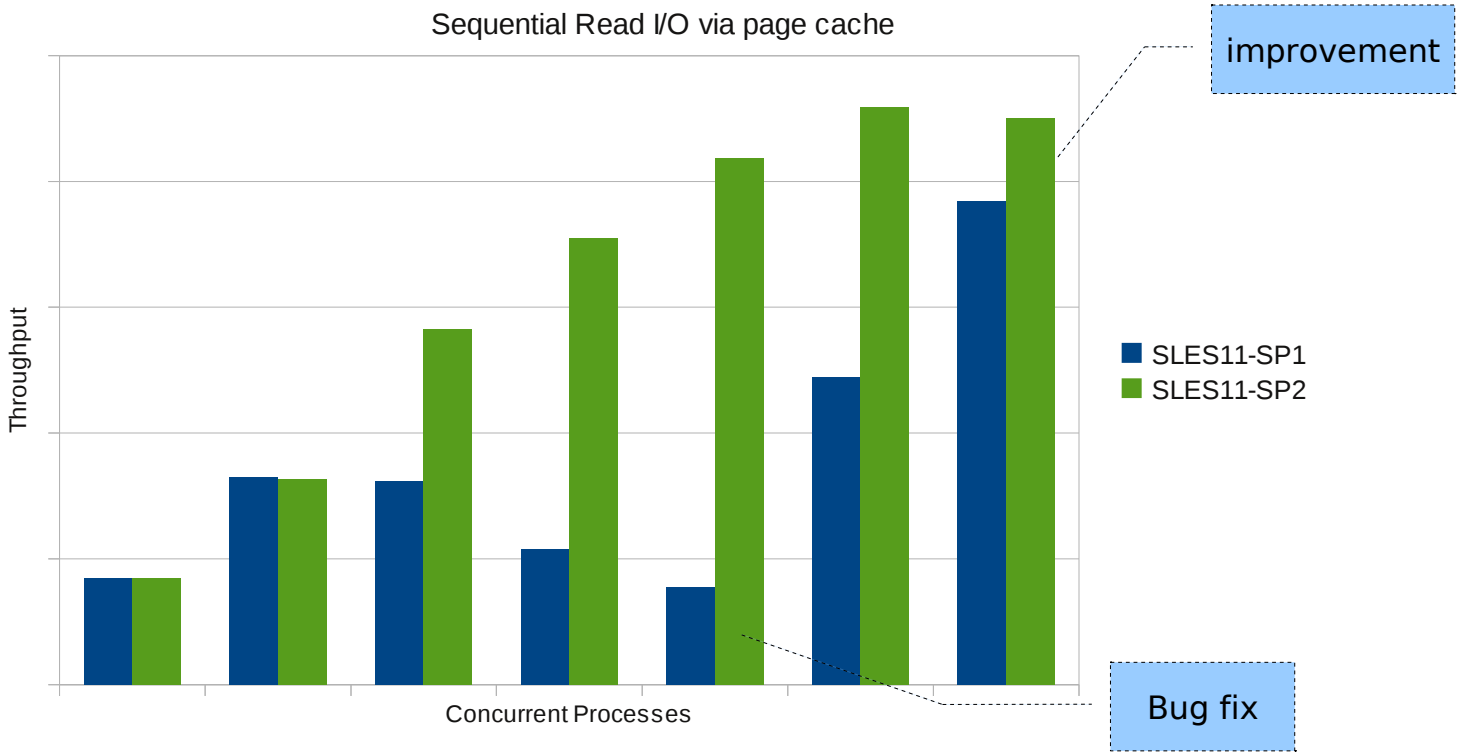

- Degradations and Improvements often show no clear line to stay away from
  - ✓ Overall we rated most of the network changes as acceptable tradeoff
    - If your workload matches exactly one of the degrading spots it might be not acceptable for you
    - On the other hand if your load is in one of the sweets spots your load can improve a lot
  - ✓ No solid recommendations what will surely improve or degrade in a migration
    - While visible in pure network benchmarks, our Application benchmarks didn't show TODO
    - Streaming like workloads improve in most, but not all cases

# Benchmark description - Disk I/O

- Workload
  - ✓ Threaded I/O benchmark
  - ✓ 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
  - ✓ Main memory was restricted to 256 MiB
  - ✓ File size (overall): 2 GiB, Record size: 64KiB
  - ✓ Scaling over 1, 2, 4, 8, 16, 32, 64 processes
  - ✓ Sequential run: write, rewrite, read
  - ✓ Random run: write, read (with previous sequential write)
  - ✓ Once using bypassing the page cache
  - ✓ Sync and Drop Caches prior to every invocation

# Page cache based read

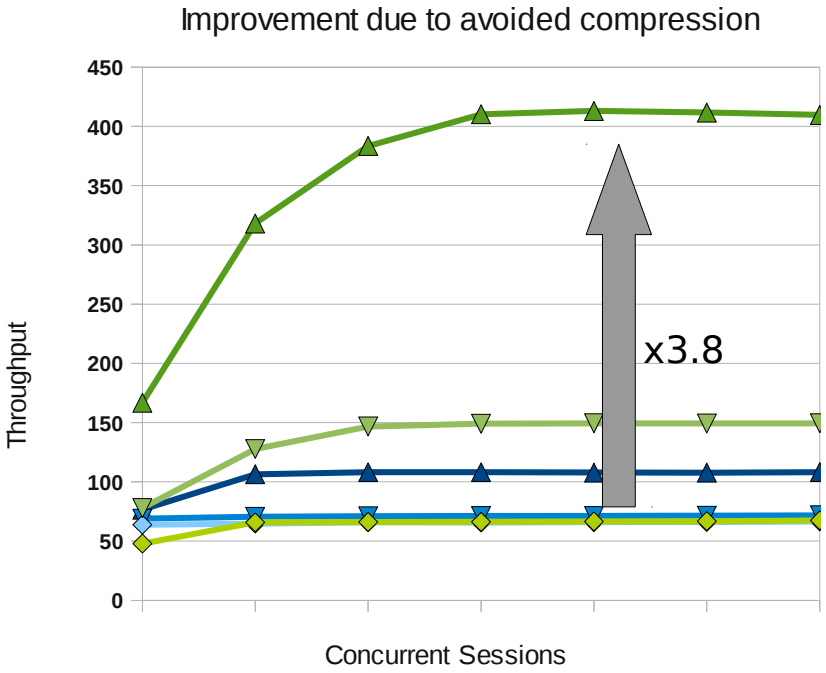
- Huge improvement for read throughput
  - ✓ Most of the impressive numbers are caused by a fixed bug included in older releases
  - ✓ Occurred if a lot of concurrent read streams ran on a small (memory) system
    - Last Distribution releases only had a partial mitigation of the issue
  - ✓ The improvements for other loads are within a range from 0 to 15%

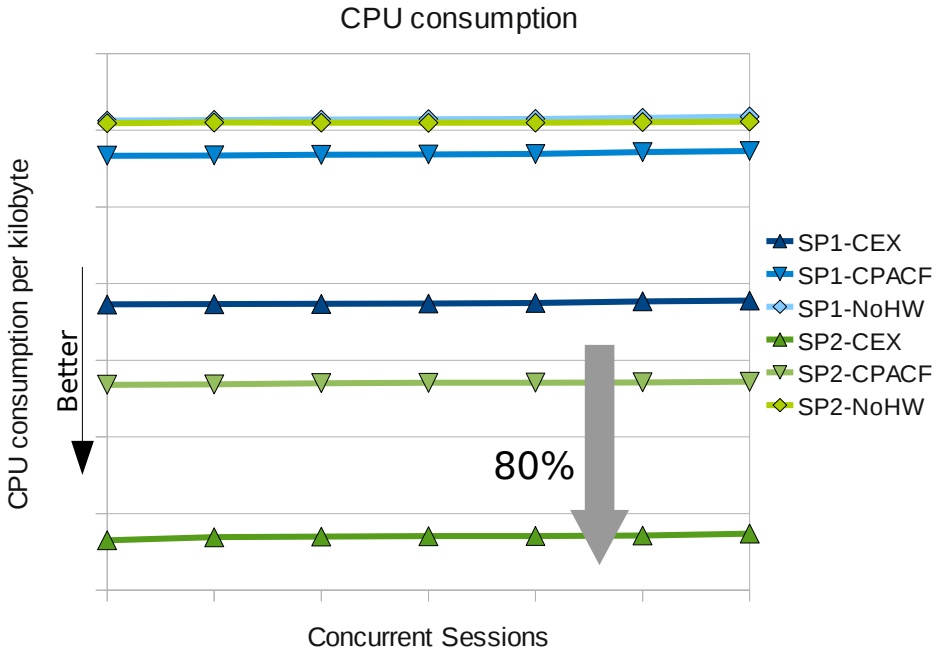


# OpenSSL based cryptography

- OpenSSL test suite
  - ✓ Part of the openssl suite
  - ✓ Able to compare different Ciphers
  - ✓ Able to compare different payload sizes
  - ✓ contains a local and distributed (via network) test tools
  - ✓ Can pass handshaking to crypto cards using the ibmca openssl engine
  - ✓ Can pass en-/decryption to accelerated CPACF commands using the ibmca openssl engine
- Our Setups
  - ✓ Scale concurrent connections to find bottlenecks
  - ✓ Iterate over different Ciphers like AES, DES
  - ✓ Run the workload with different payload sizes
  - ✓ Run SW only, CPACF assisted and CPACF + CEX3 Card assisted modes
    - CEX cards in accelerator and co-processor mode
  - ✓ We use distributed clients as workload driver
    - Evaluate overall throughput and fairness of throughput distribution
    - Evaluate the CPU consumption caused by the load

# OpenSSL based cryptography




- Compressing the data to save cryptographic effort was the default for a while
  - ✓ Counter-productive on System z as CPACF/CEX is so fast (and CEX account as off-loaded)
- Now it is possible to deactivate compression via an Environment variable `OPENSSL_NO_DEFAULT_ZLIB=Y`
  - ✓ 1000k payload cases with CPACF and cards x3.8 times faster now, still x2.3 without CEX cards
  - ✓ Even 40b payload cases still show 15% throughput improvement
  - ✓ At the same time depending on the setup 50% to 80% less cpu per transferred kilobyte

# Agenda

- Performance Evaluation
  - ✓ System under test (SUT) description
  - ✓ Changes one should be aware of
- Performance evaluation Summary
  - ✓ Improvements and degradations per area
  - ✓ Summarized comparison

# SLES 11 SP2 Improvements & Degradations per area

**SLES 11 SP2 vs. SLES 11 SP1**

| Improvements/Degradations | Especially affects, but not limited to the following workloads |
|---|---|
| Process scaling | Websphere Family, large scale Databases |
| Filesystem Scaling | File serving |
| Network Streaming | TSM, replication tasks (DB2 HADR, Domino) |
| Disk I/O via page cache | Clearcase, DB2 on ECKD disks, File serving, Datastage |
| Disk I/O | TSM, Databases |
| Cryptography | Secure Serving/Communication in general |
| Pure Virtual Networks (vswitch G2G, HS) | Common Hipersocket setups: SAP enqueue server, Websphere to z/OS, Cognos to z/OS |

# Summary for SLES 11 SP2 vs. SP1

- SLES 11 SP2 performance is good
  - ✓ Improved compared to the already good SP1 release
    - Beneficial effects slightly bigger on newer System zEnterprise systems
  - ✓ Generally recommendable
    - Except environments focusing on pure virtual networks
- Improvements and degradations

| Level | On HW | Improved | No difference or Trade-off | Degraded |
|---|---|---|---|---|
| SLES 11 SP2 | z10 | 30 | 67 | 8 |
| SLES 11 SP2 | z196 | 33 | 64 | 3 |

# Questions

- Further information is located at
  - ✓ 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/

***Mario Held***

*Linux on System z*
*System Software Performance Engineer*

*IBM Deutschland Research & Development*
*Schoenaicher Strasse 220*
*71032 Boeblingen, Germany*

*Phone +49 (0)7031–16–4257*
*Email mario.held@de.ibm.com*