Linux on System z Performance Update
Part 1: z10 CPU, Compiler, Java, Linux Kernel

Thomas Weber (tweber@de.ibm.com)

WAVV Conference 2009

Orlando, FL, May 15-19

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.

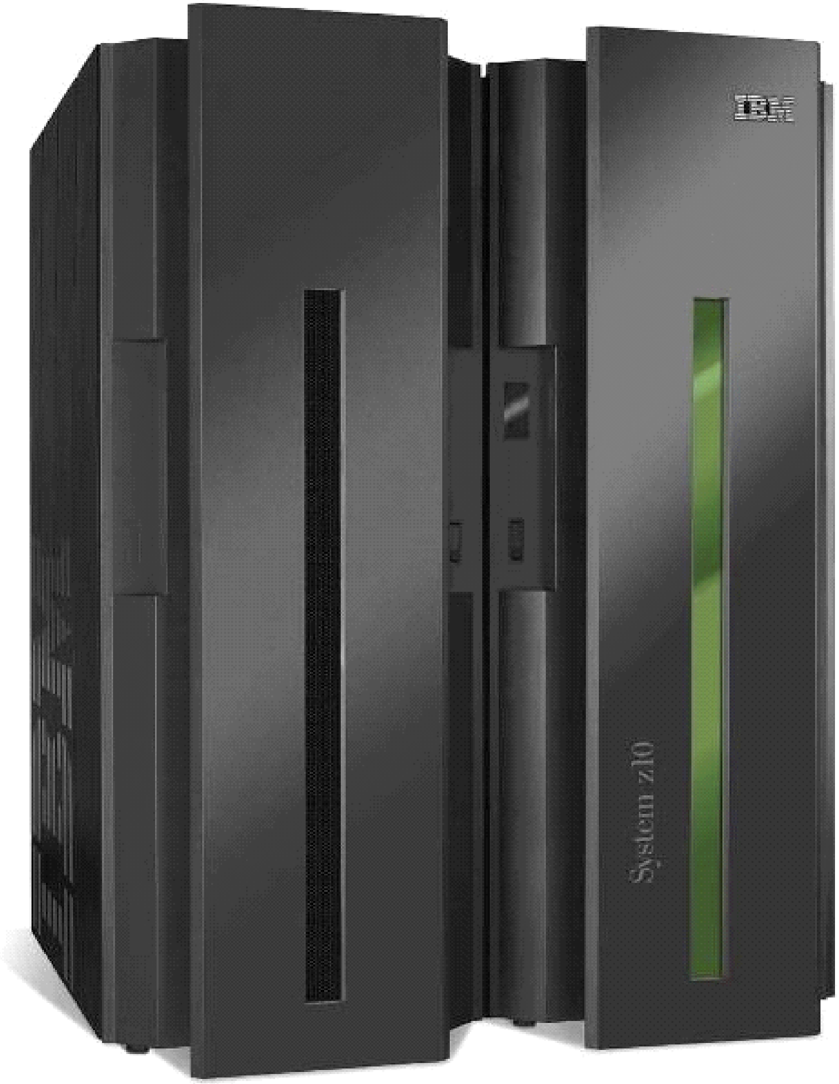
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

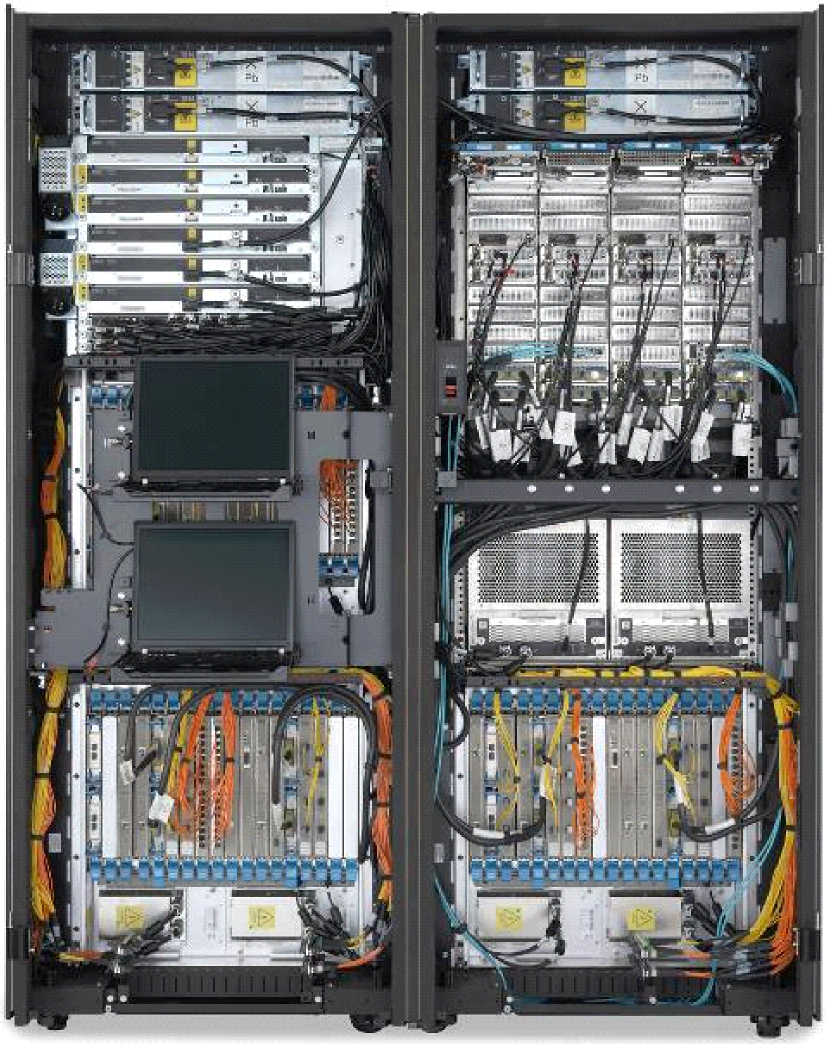
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products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

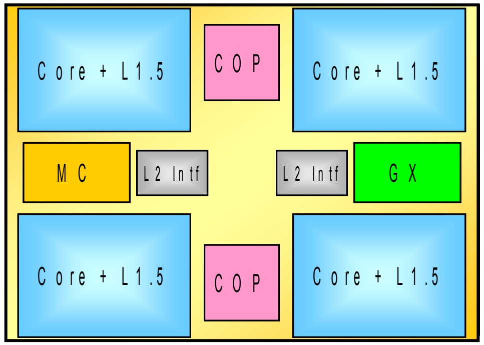
- z10 CPU, Compiler, Java, Linux Kernel
  - short System z HW overview
  - z10 Performance
  - GCC compiler
  - Java server performance
  - CPU hotplug function


IBM
System z10

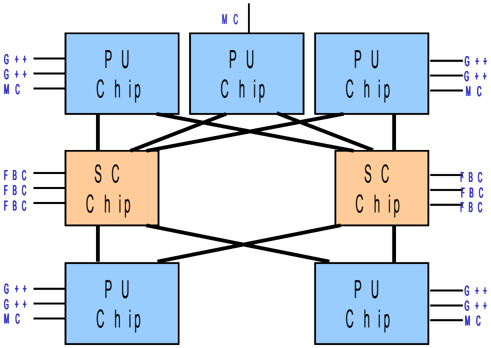


z10 Technology

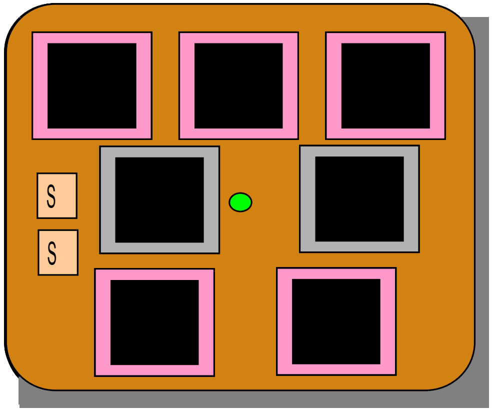
CP Chip



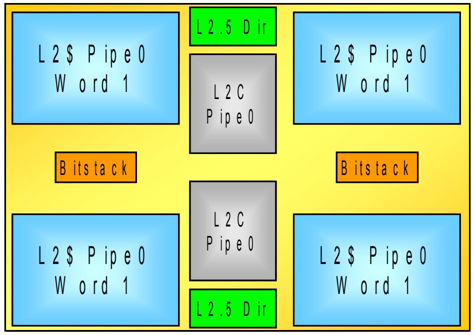
20w Node



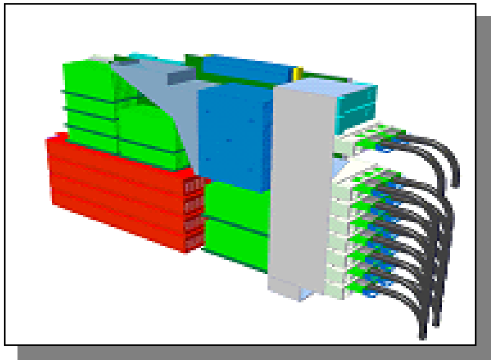
MCM



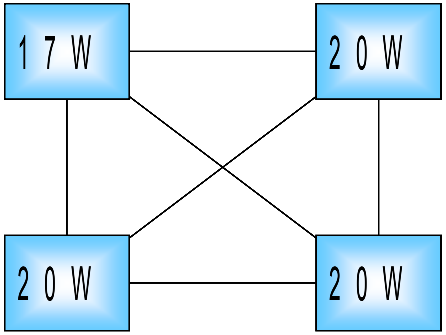
SC Chip



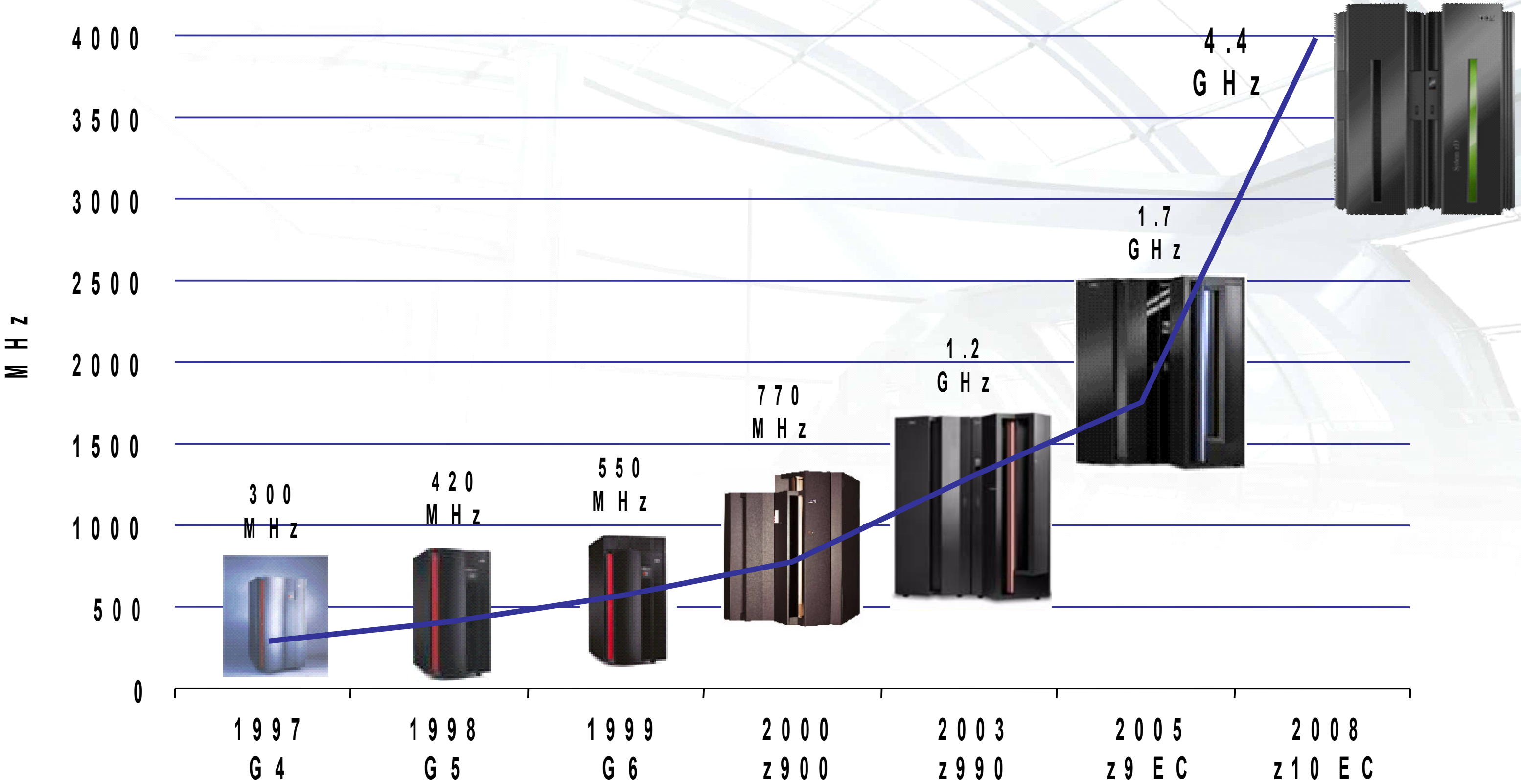
Node package



77w CEC



IBM z10 EC Continues the CMOS Mainframe Heritage

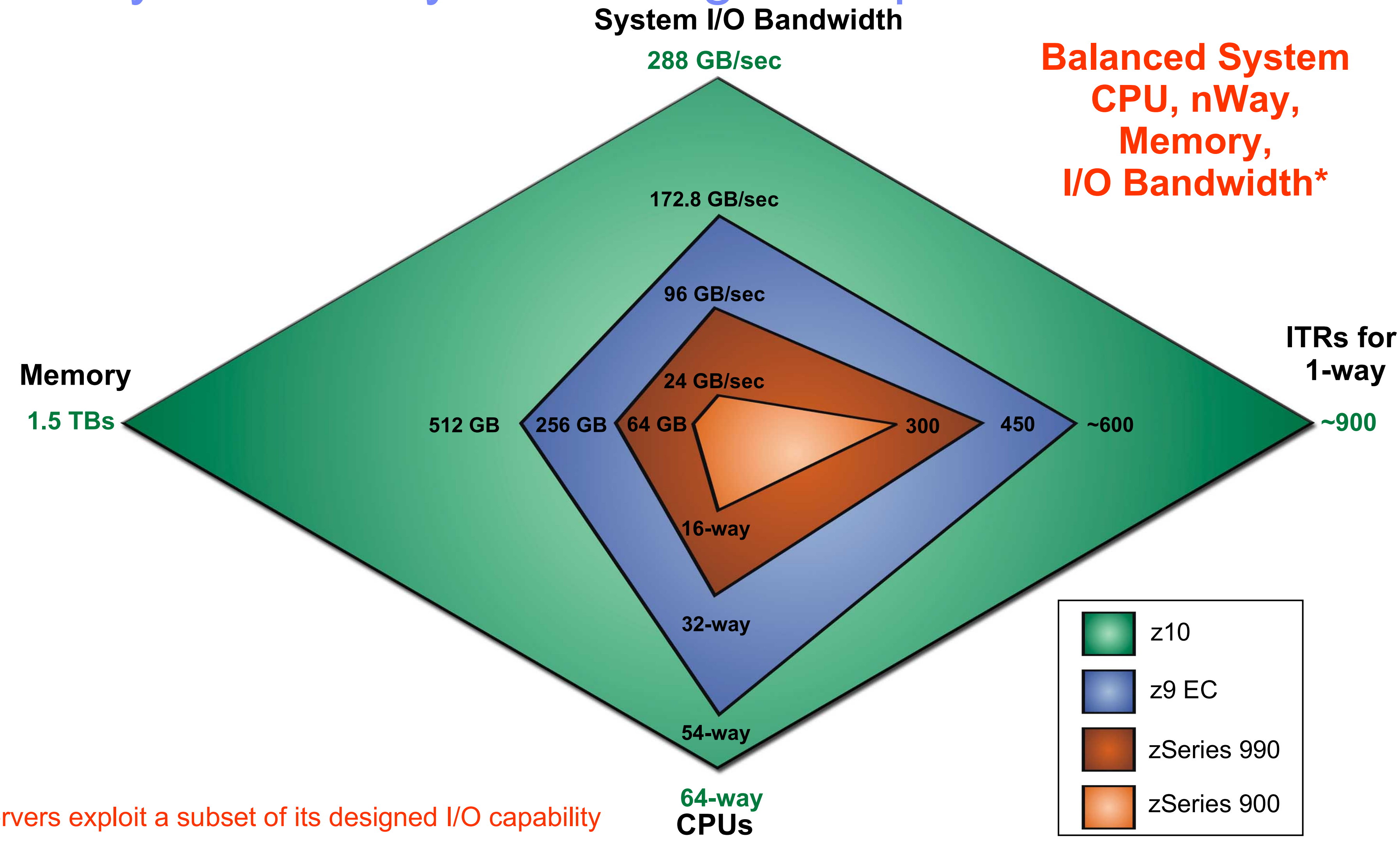


- G4 - 1st full-custom CMOS S/390®
- G5 - IEEE-standard BFP; branch target prediction
- G6 - Cu BEOL

- IBM eServer zSeries 900 (z900) - Full 64-bit z/Architecture®
- IBM eServer zSeries 990 (z990) - Superscalar CISC pipeline
- z9 EC - System level scaling

- z10 EC - Architectural extensions

IBM System z – system design comparison



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

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

LSPR Mixed Workload for System z10 EC

	z10 EC to z9 EC Ratios
LSPR Mixed workload average, multi-image for z/OS 1.8 with HiperDispatch active on z10 EC!	
Uni-processor	1.62
16-way z10 EC to 16-way z9 EC	1.49
32-way z10 EC to 32-way z9 EC	1.49
56-way z10 EC to 54-way z9 EC	1.54
64-way z10 EC to 54-way z9 EC	1.70

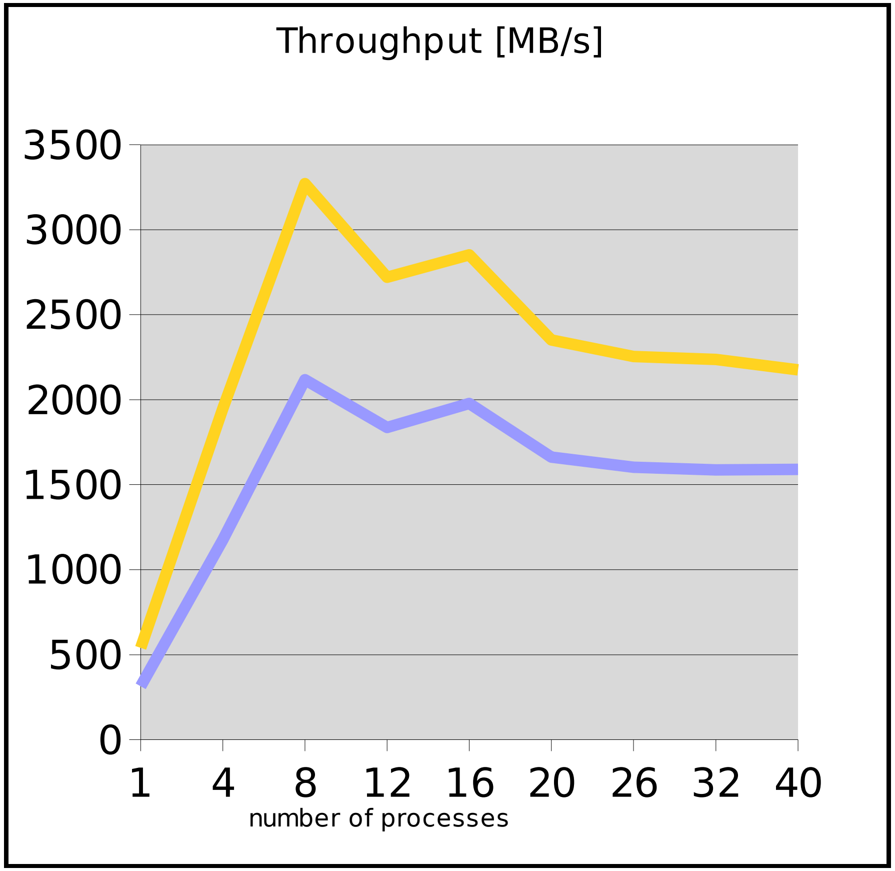
IBM Large System Performance Reference (LSPR)
http://www.ibm.com/systems/z/advantages/management/lspr/lsprwork.html

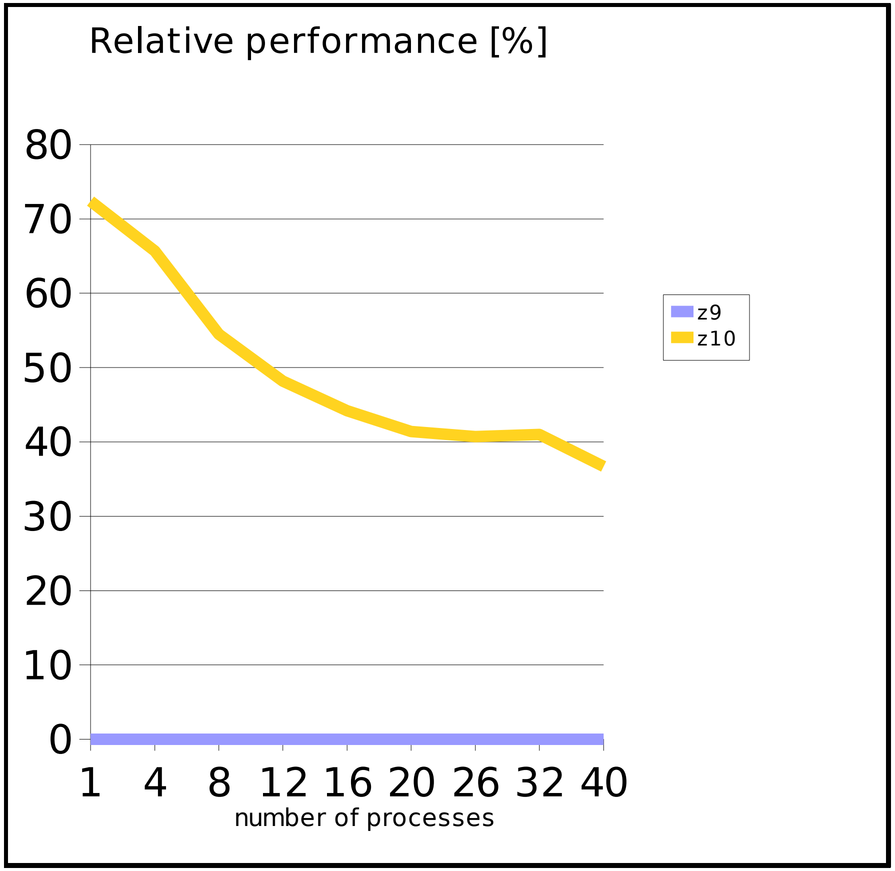
File server benchmark description

- dbench 3
  - Emulation of Netbench benchmark, rates windows file servers
  - Mainly memory operations (large page cache)
  - 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

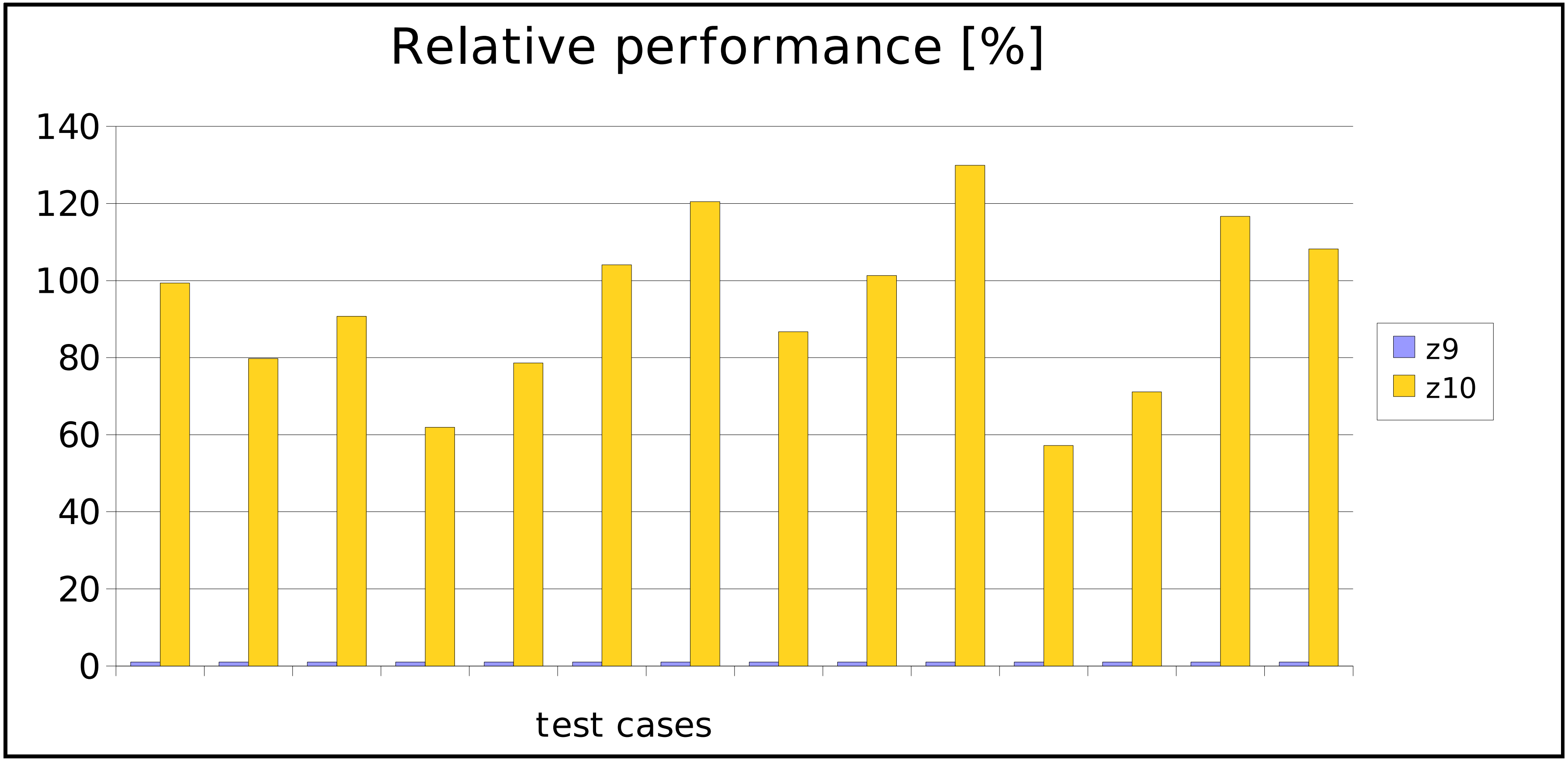
- Improvement z10 versus z9:
  - Measured with 8 CPUs: average improvement is 50%





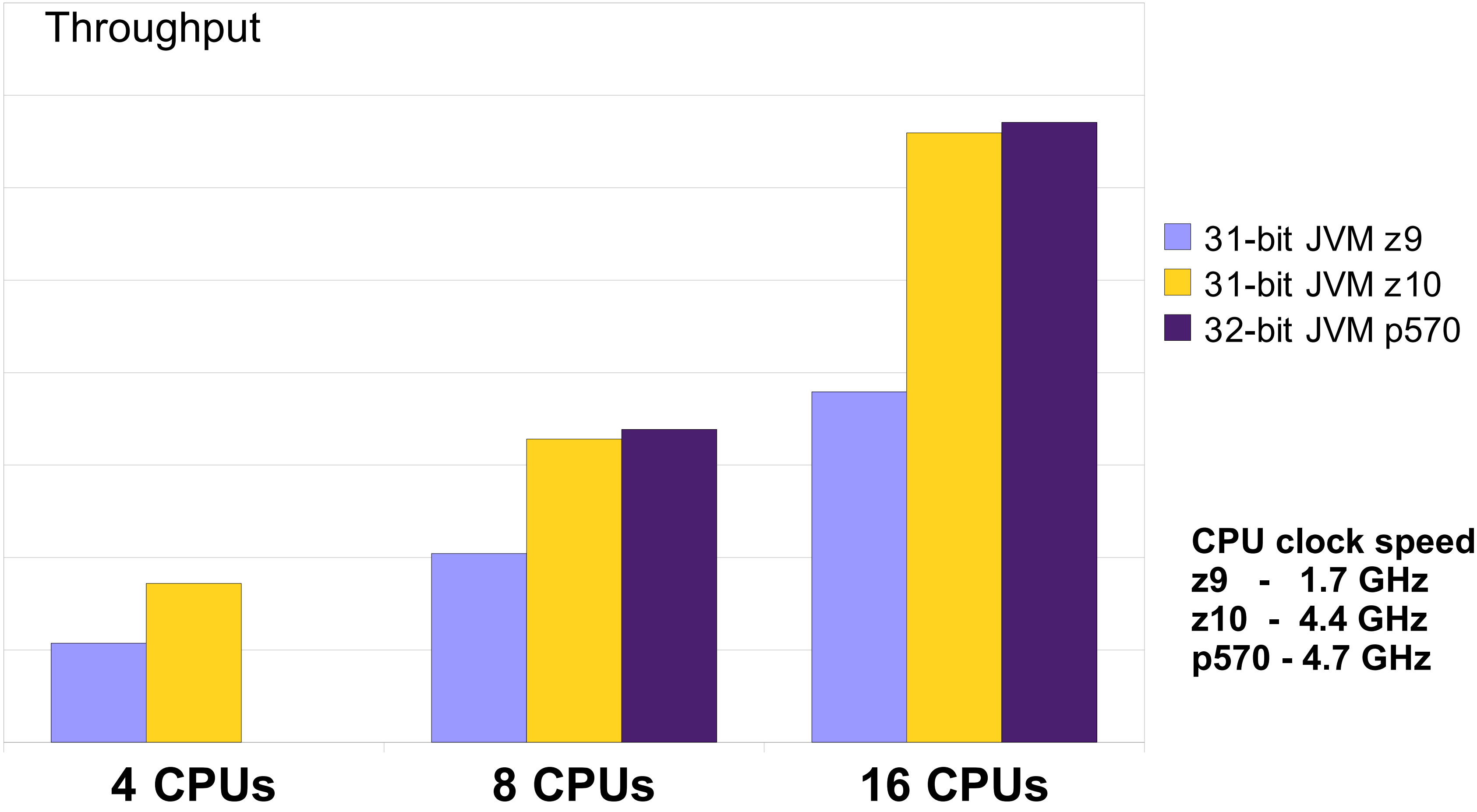
z10 Performance: CPU intensive workloads

- Overall improvement with z10 versus z9: 1.9x
- GCC-4.3 compiler can use -march=z10 option



z10 Performance: Java workload

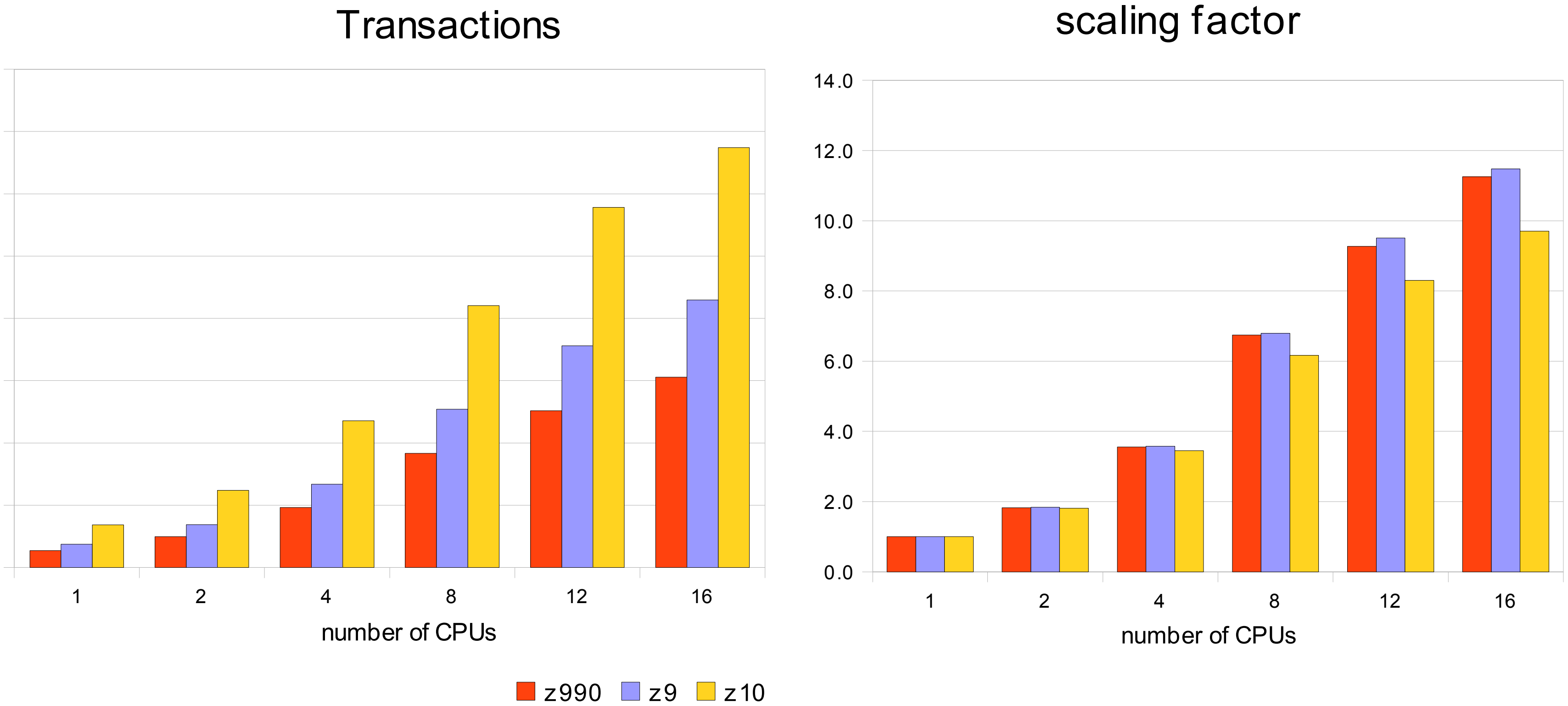
- System z versus System p



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

z10 with Informix IDS 11 OLTP workload

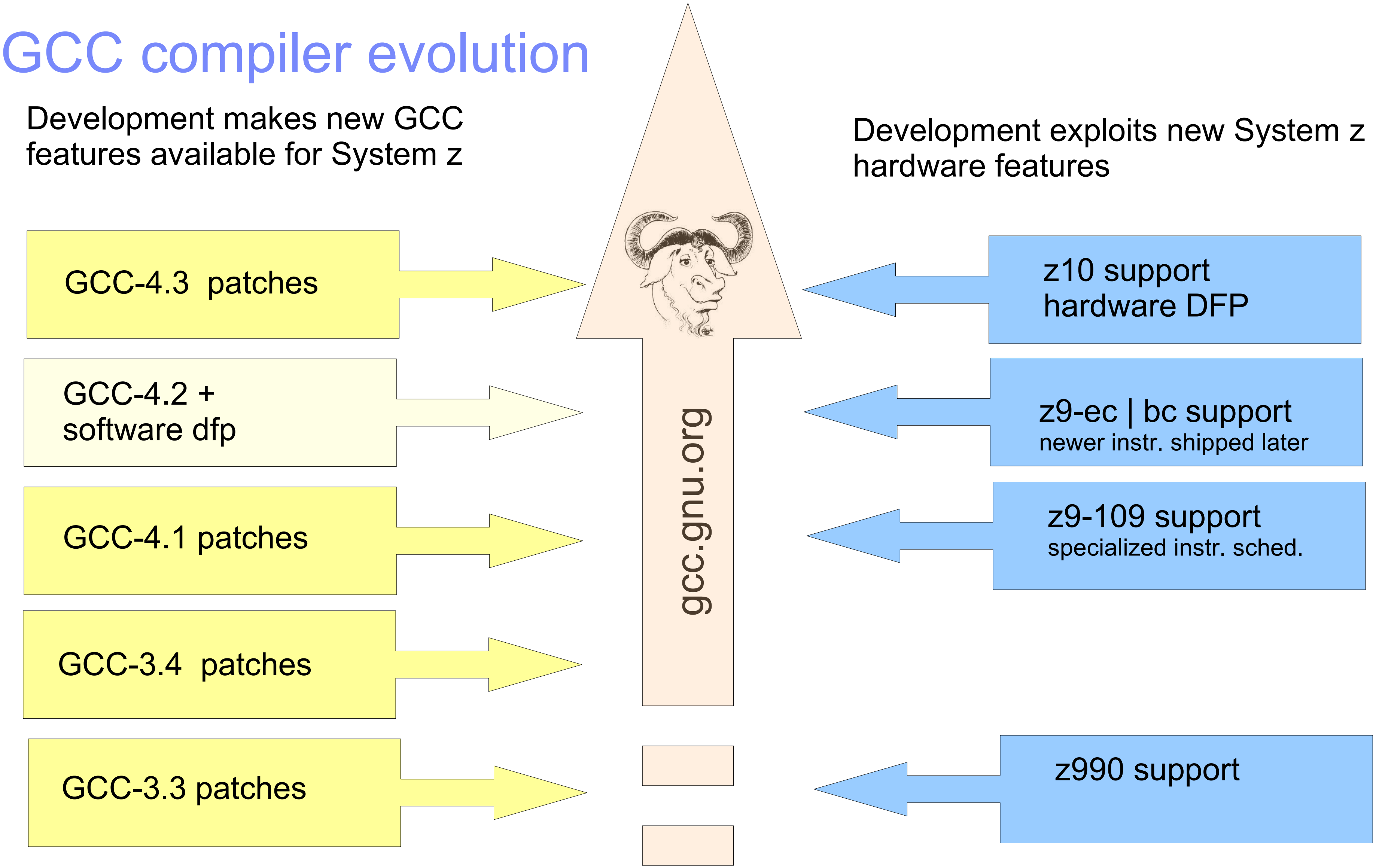
- Throughput improvements
  - z9 to z10: 65% - 82%
  - x numbers of z10 CPUs can do the same work as 2x z9 CPUs



z10 Performance summary

- System z evolution continues
- Performance boost from z9 to z10
- Balanced System
- LSPR expectations met
- Excellent on compute intensive and Java workloads
- GCC 4.3 available with SLES11

GCC compiler evolution



Development makes new GCC features available for System z

Development exploits new System z hardware features

GCC-4.3 patches

GCC-4.2 + software dfp

GCC-4.1 patches

GCC-3.4 patches

GCC-3.3 patches

z10 support
hardware DFP

z9-ec | bc support
newer instr. shipped later

z9-109 support
specialized instr. sched.

z990 support

GCC versions supported on System z

GCC version	Used in SUSE distribution	Used in Red Hat distribution
GCC-3.3	SLES9	
GCC-3.4		RHEL4
GCC-4.0		
GCC-4.1	SLES10	RHEL5
GCC-4.2		
GCC-4.3	SLES11	
GCC-4.4		

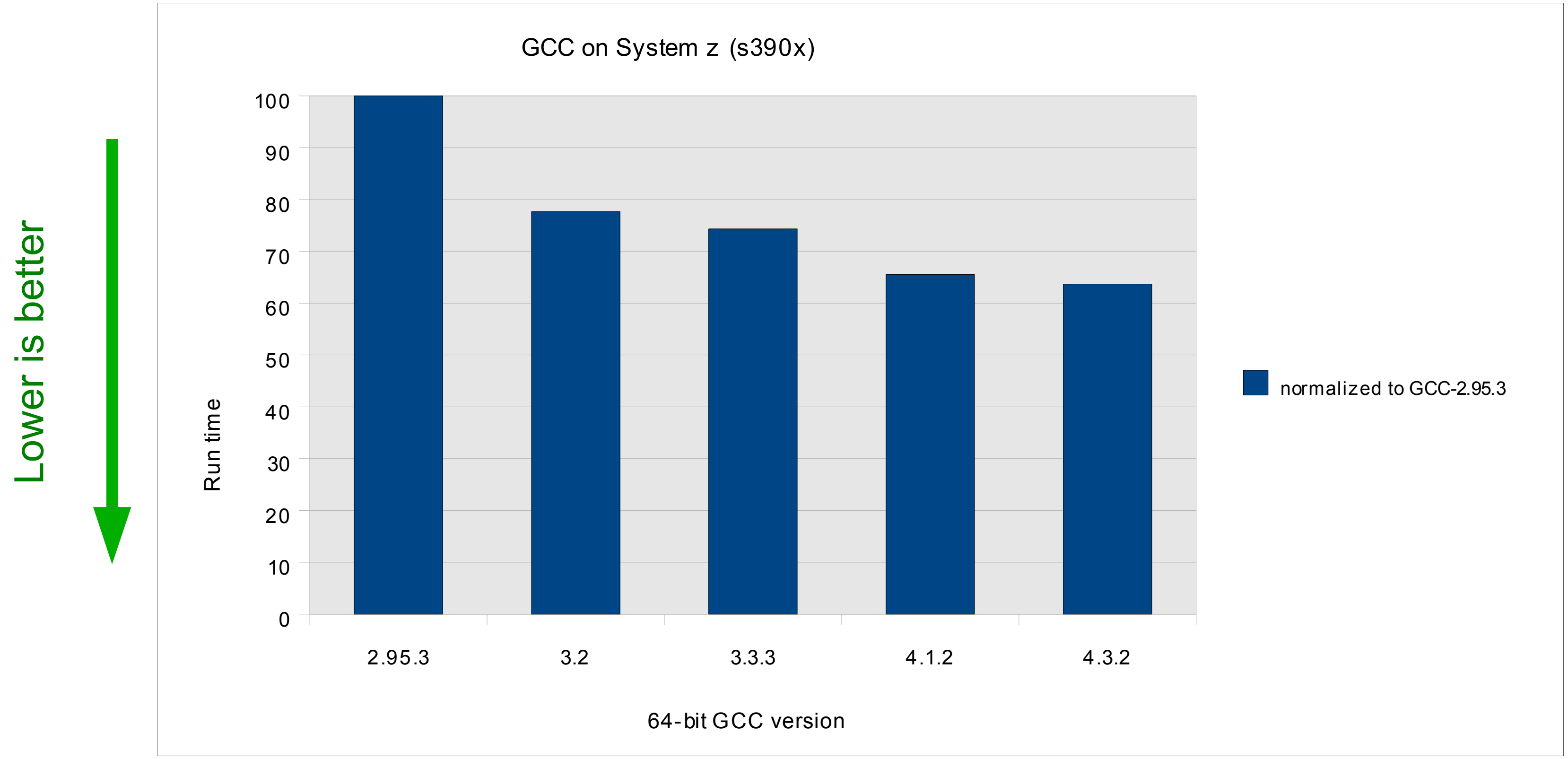
Optimizing C/C++ code

- Produce optimized code
  - Options -O3 or -O2 (often found in delivered Makefiles) are a good starting points
  - Optimize GCC instruction scheduling with the performance critical target machine in mind
    - -mtune=values <z900,z990>, <z9-109 with GCC-4.1>,
      <z9-ec | bc with GCC-4.2 and GCC-4.3>, <z10 with SLES11 GCC-4.3>
  - If you know the target machine exploit improved machine instruction set
    - -march=values <z900,z990>, <z9-109 with GCC-4.1>,
      <z9-ec | bc with GCC-4.2 and GCC-4.3>, <z10 with SLES11 GCC-4.3>
    - -march build binaries are only upward compatible!

- Fine Tuning: additional general options on a file by file basis
  - Use inline assembler for performance critical functions may have advantages
  - Avoid -fPIC for executables (gen. position independent code)
  - -funroll-loops often has advantages
  - -ffast-math speeds up calculations (if not exact implementation of IEEE or ISO rules/specifications for math functions is needed)
  - Don't use debugging options in the final executable

GCC performance evolution on System z

- Run time of industry standard benchmark applications with newer GCC versions is much shorter



DFP – Decimal Floating Point
Limitations of binary numbers in economy

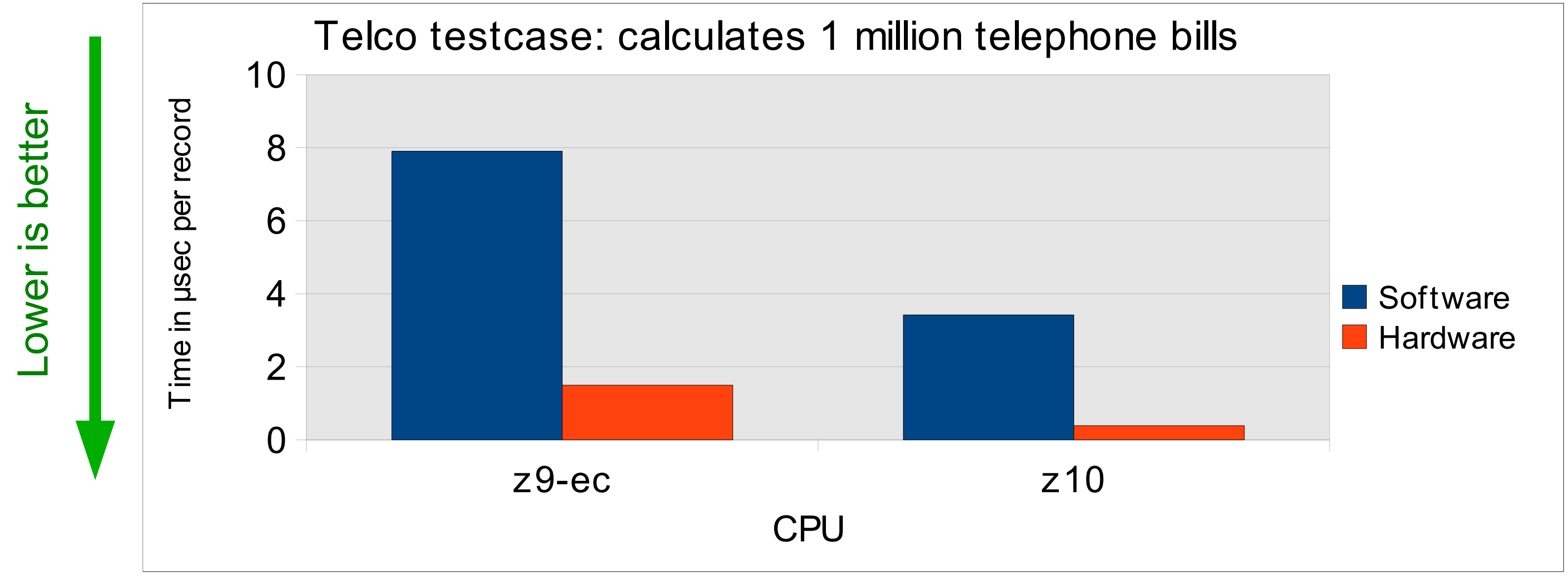
- Trading goods and amounts of money cannot be calculated or represented exactly by <u>binary floating point</u> numbers
  - Many numbers cannot be represented properly (1/5, 1/10)
  - People who are used to decimal numbers expect results and calculations to be available with full precision
  - The traditional binary representation is not suitable for usual calculations
    - $ 0.70 x 1.05 = $ 0.73499999999999998667732370449812151491641998291015625
    - Rounding to two digital places gives $ 0.73
    - Expected is $ 0.70 x 1.05 = $ 0.735 => rounded to $ 0.74
- If you rely on correctly calculated results without DFP you have to add many lines of code to your program
  - Example: troublesome binary floating point rounding mechanisms
    - Sometimes more than 50 times the number of lines than in an DFP implementation
  - Additional code is error-prone
  - Depending on the amount of calculations a performance degradation is to expect
  - TCO is higher due to service, maintenance, run time

DFP - support added in GCC

- Front end support (C, C++, Fortran, Java):
  - Support for the 3 new data types: _Decimal{32|64|128}
  - Support for DFP constants written with DF suffix
- Middle end support:
  - Complete DFP arithmetic layer for constant folding
  - Support for integer or IEEE floating point conversion routines
- GCC - versions
  - Software DFP support in GCC-4.2 added
  - Hardware DFP support in GCC-4.3 added (usable with z9-EC, z10)
- GCC-4.3.2 is available in SLES11
  - offers DFP in a supported environment on Linux on System z
  - The usage of DFP arithmetics in applications requires the explicit use of DFP data types
  - If GCC is used with -march=z9-ec and -march=z10 the HW DFP support is used by default (-mhard-dfp/-mno-hard-dfp options)

DFP - decimal floating point performance

- Telco benchmark models a telephone company's billing system
  - Billing of one million telephone calls including tax using DFP arithmetics
- Big advantage if DFP hardware support is exploited
  - z9-EC DFP hardware support in millicode
  - z10 DFP hardware support by real hardware -> much faster



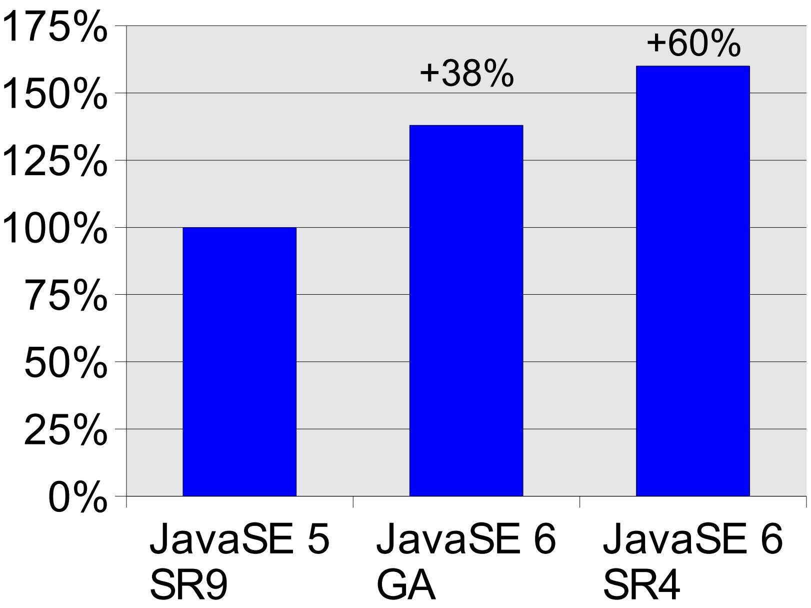
Java on servers: Workload

- 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

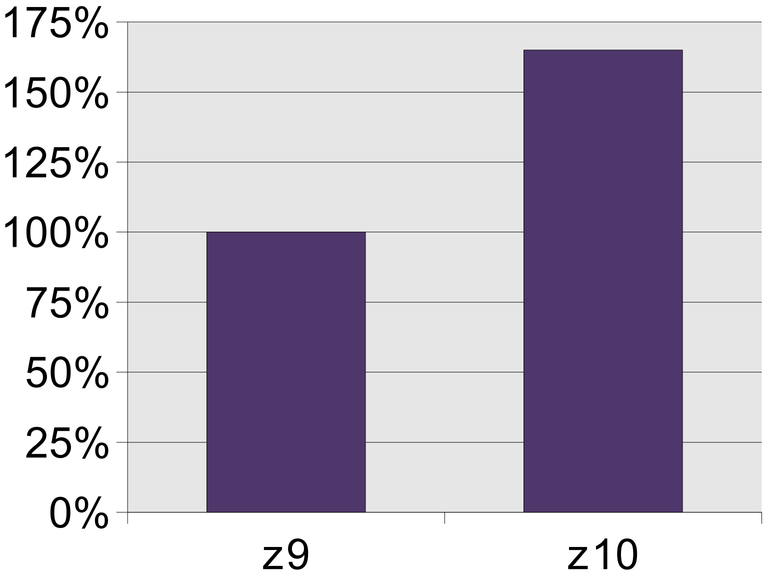
Java on servers: Performance Improvements

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

History of Java versions



System z with Java SE 6 GA

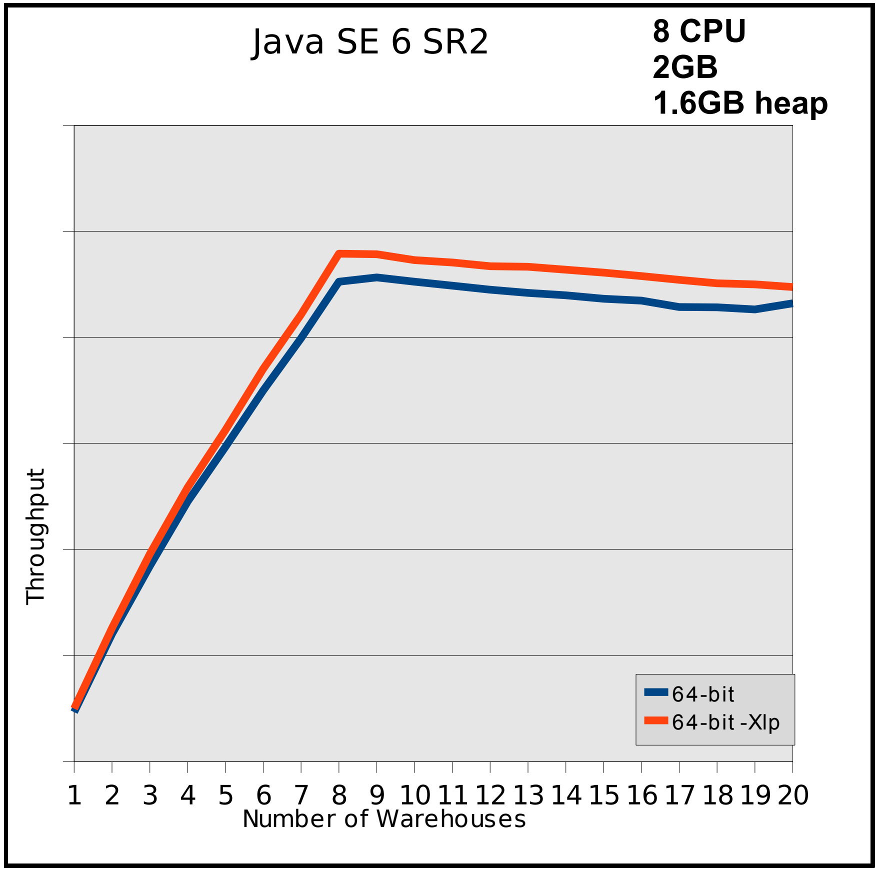


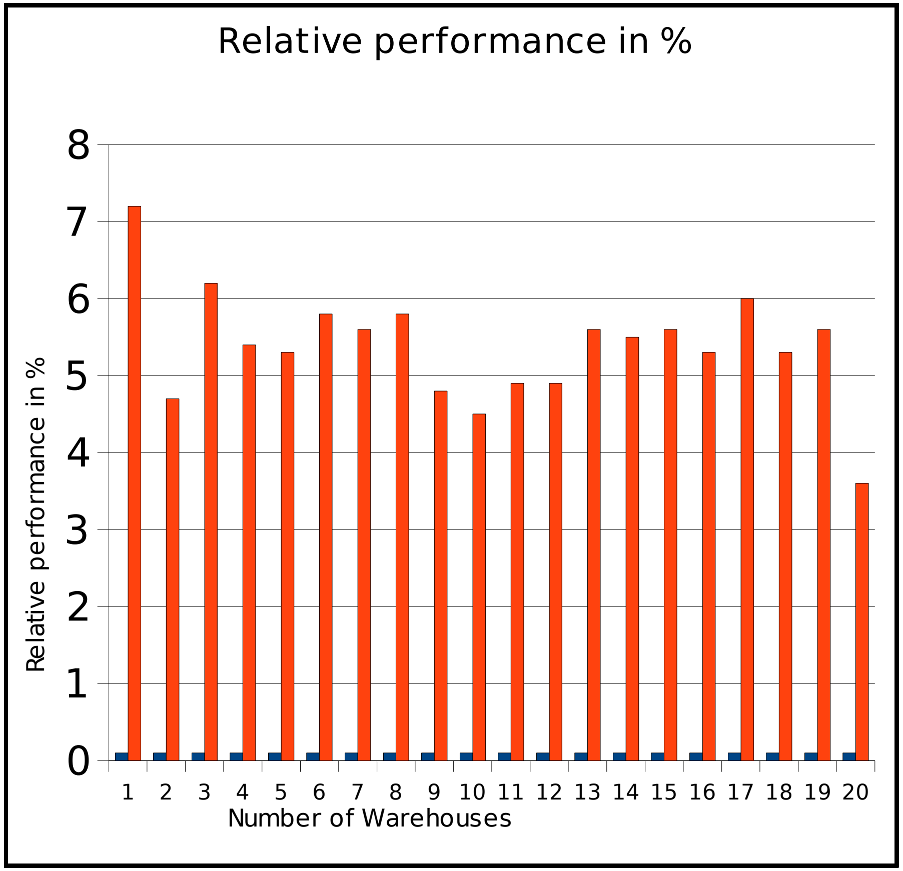
Java on servers: Heap size

- 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: infrequent 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 – workload dependent
  - setting a fixed heap size: **-Xms** (initial), **-Xmx** (maximum), when initial==maximum
  - monitor garbage collection (GC): **-verbose:gc**
  - **-Xlp** tries to allocate large pages for the heap
    - prereq: Linux kernel needs to be setup for large pages (vm.nr_hugepages)
  - control GC behavior: **-Xgcpolicy:[optthruput, optavgpause, gencon]**
  - 64-bit: smaller size of heap objects: **-Xcompressedrefs**

Java on servers: Large page support (z10 feature)

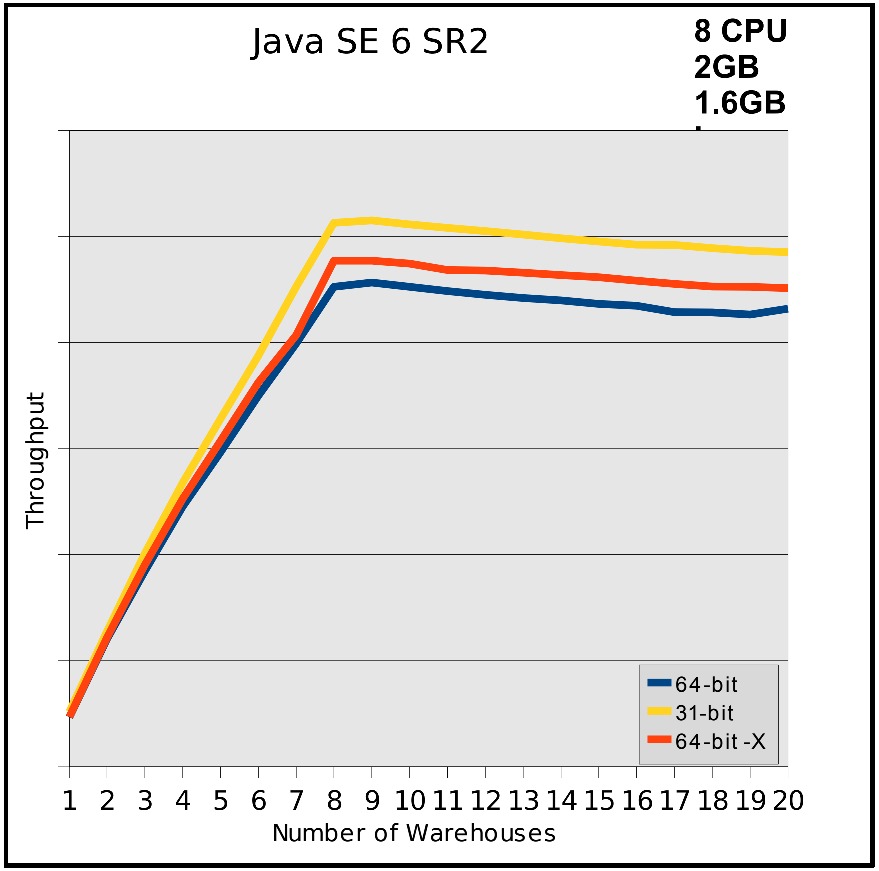
- use of **-Xlp** improves throughput
- large page size was 2 MB (default for SLES10 and RHEL5)

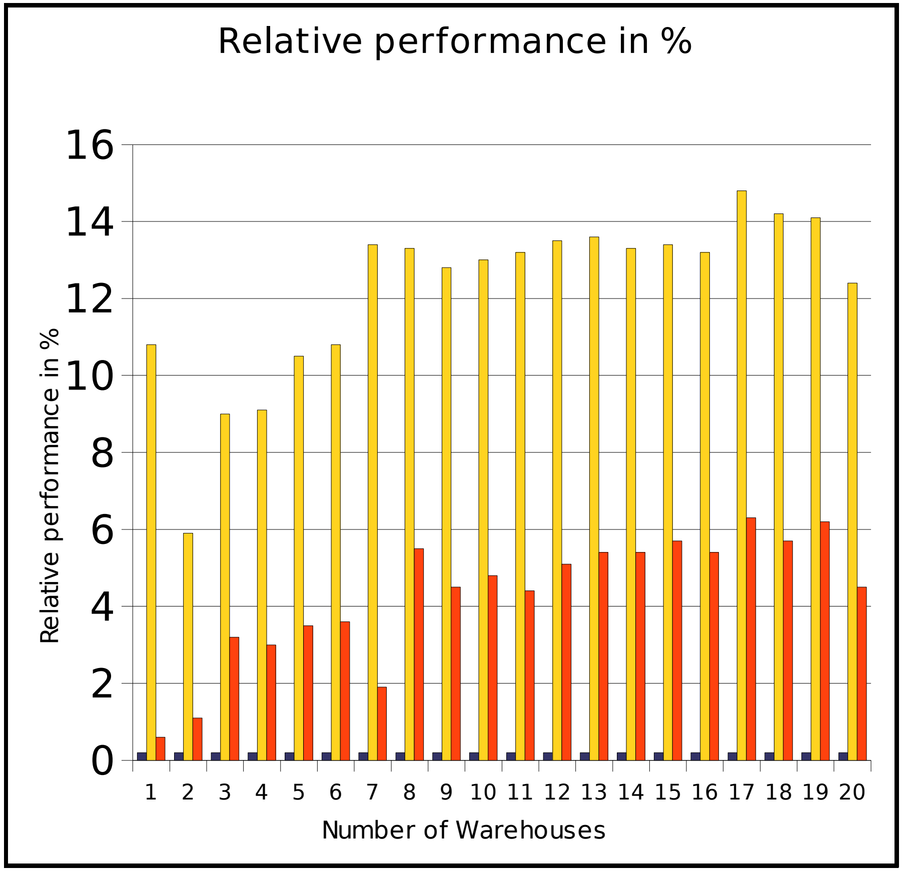




Java on servers: 31-bit vs. 64-bit

- use of **-Xcompressedrefs** provides relief for 64-bit (new in Java SE 6 SR2)
- smaller size of 64-bit heap objects





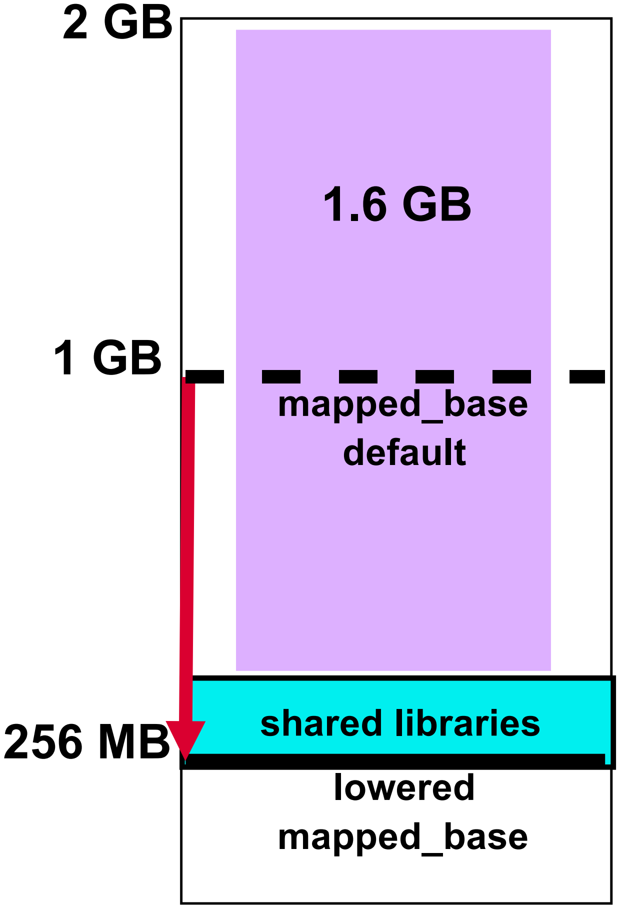
Java on servers: larger heaps for 31-bit Java kits (1)

- 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**

==> retry to allocate a larger heap size



Java on servers: larger heaps for 31-bit Java kits (2)

- modify Linux memory layout
  - RHEL includes flex-mmap patch; turn off Linux prelinking
- applies RHEL 4,5 distributions (31-bit emulation mode)

HOWTO:

- show state of flex-mmap patch
  - **cat /proc/sys/vm/legacy_va_layout**
  - 0 means flex-mmap is enabled; 1 means old memory layout
- enable flex-mmap if disabled
  - **echo 0 > /proc/sys/vm/legacy_va_layout**
- disable Linux prelinking
  - in **/etc/sysconfig/prelink** set PRELINKING=no
- apply setting by running the daily cron prelink job immediately
  - **# /etc/cron.daily/prelink** <ENTER>

==> retry to allocate a larger heap size

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

CPU hotplug function

- Changes the number of used processors on the fly, depending on the current overall utilization and load
- available with SLES10 SP2
- Expectation:
  - **Increases the performance of single threaded applications within a z/VM or LPAR environment with multiple CPUs**
- Enables or disables CPUs based on a set of rules
- Is enabled in the kernel configuration by setting

```
Base setup --->
--- Processor type and features ---
...
Symmetric multi-processing support (CONFIG_SMP)
 └─Support for hot-pluggable CPUs (CONFIG_HOTPLUG_CPU)
```

CPU hotplug parameters

- The configuration information is stored at `/etc/sysconfig/cpuplugd`
- Minimum number of CPUs is set with `cpu_min="<number>"`
- Maximum number of CPUs is set with `cpu_max="<number>"`
  - `0 means number of detected CPUs`
- The update interval is set with `update="<value in seconds>"`
  - `Default is 10 seconds`
- Consider the effect of kernel "cpu" parameters:
  - `maxcpus=<n>` sets the number of processors which will be active after system boot
  - `possible_cpus=<n>` is the upper limit for hotpluggable CPUs
  - If `possible_cpus` is not specified but `maxcpus` is, then `maxcpus` is the upper limit for hotpluggable CPUs

CPU hotplug rules

- The default rule for increasing the number of CPUs is

  ```
  HOTPLUG="(loadavg > onumcpus + 0.75) & (idle < 10.0)”
  ```

  - An additional CPU is enabled, if the loadaverage is greater than the number of active (online) CPUs plus 0.75 and the current idle percentage is less than 10 percent.
- The default rule for decreasing the number of CPUs is

  ```
  HOTUNPLUG="(loadavg < onumcpus - 0.25) | (idle > 50)”
  ```

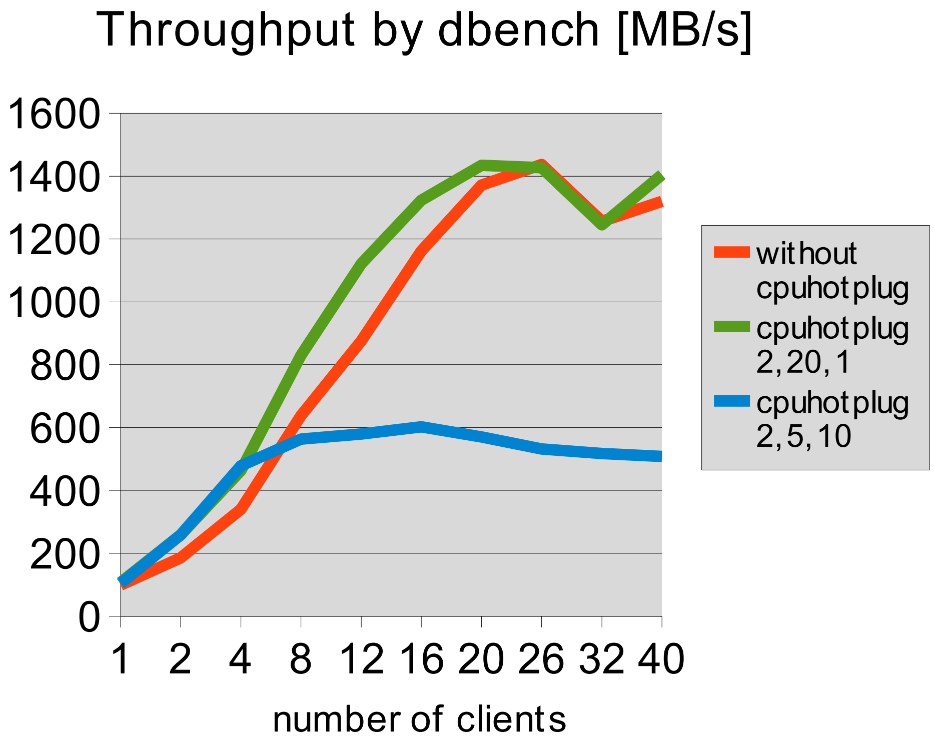
  - A CPU is disabled, either if the current load is below the number of active CPUs minus 0.25 or if the idle percentage is greater than 50%.
- The formulas for these rules can be modified. See “Device Drivers, Features and Commands” for valid expressions.
- Note:
  - `loadavg` is a value that changes slowly
  - `idle` changes fast
  - Increments and decrements of active CPUs are done in steps of 1 every time when the rules are checked.

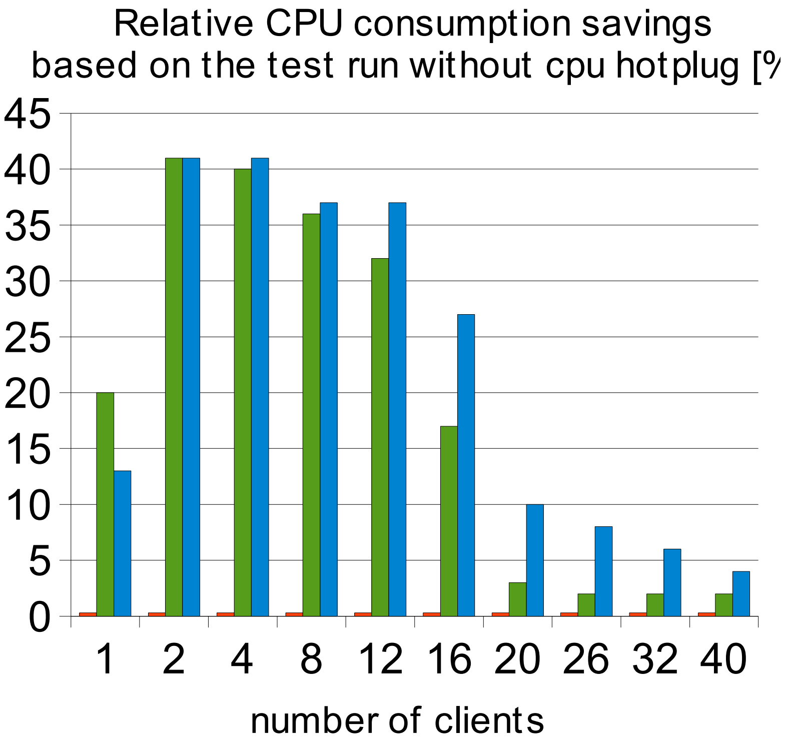
CPU hotplug test workload

- dbench 3
  - Emulation of Netbench benchmark, rates windows file servers
  - Mainly memory operations (large page cache)
  - Mixed file operations workload for each process: create, write, read, append, delete
  - Scaling with 1,2,4,8,16 CPUs and 1,4,8,12,16,20,26,32 and 40 clients
  - 20 CPUs available, 2 GB memory
- Modification to the standard code:
  - Purpose:
    - Need more interaction between clients
  - Create two processes per client and communicate with POSIX message queues
  - First process:
    - Read the I/O commands from the control file
    - Pass this information to the second process
  - Second process:
    - Performs the execution of this command
    - Reports the end of the operation back to the first process

CPU hotplug performance results

- Improvements in case where the default (high) number of CPUs is not needed
- Up to 40% more throughput, up to 40% CPU cost savings



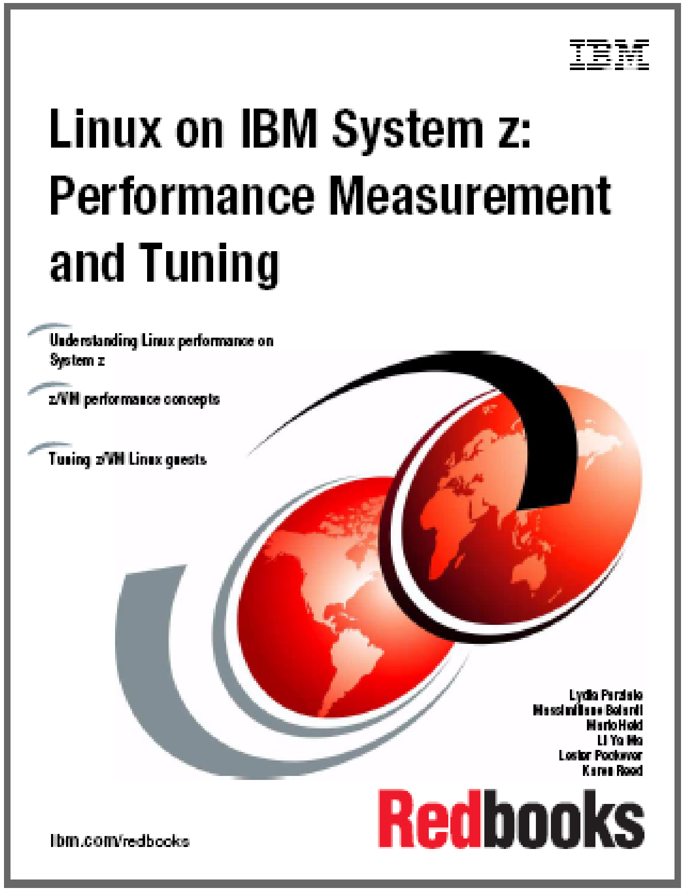


CPU hotplug summary

- This feature improves the performance by
  - sizing the correct amount of processors for a Linux system depending on its current load
  - avoiding the Linux scheduler queue balancing in partial load situations
- Set the minimum and maximum number of CPUs to values which apply to the real workload:
  - Setting `cpu_min` to 2 may be too high
  - `cpu_max` should be set so that it really covers the peaks
- Linux guests under z/VM: use z/VM 5.4
  - Guarantees that stopped processors are no longer included in virtual processor prioritization calculations
  - Ensures share redistribution

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/



IBM

Linux on IBM System z: Performance Measurement and Tuning

Understanding Linux performance on System z

z/VM performance concepts

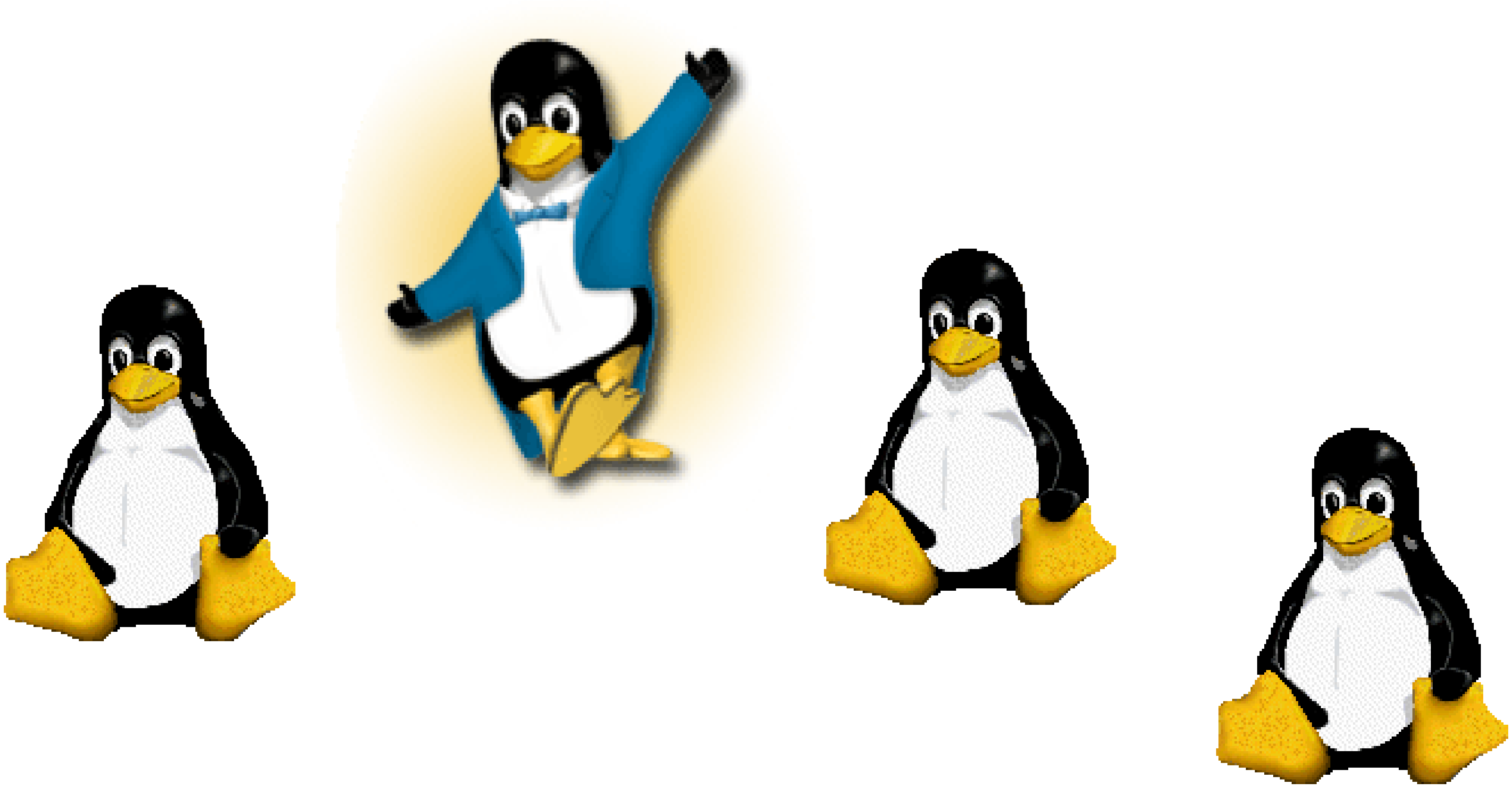
Tuning z/VM Linux guests

Lydia Parziale
Massimiliano Belardi
Mario Held
Li Ya Mo
Lester Peckover
Karen Reed

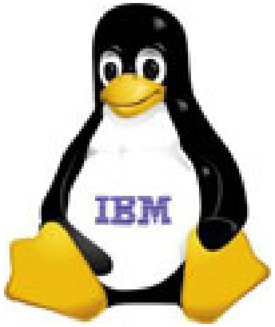
ibm.com/redbooks

Redbooks

Questions



BACKUP



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-guid

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`

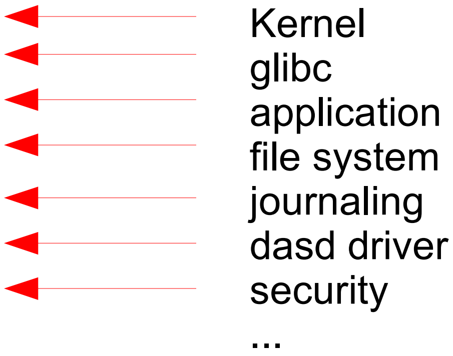
```
opcontrol --vmlinux=/boot/vmlinux-2.6.16.46-0.4-default
opcontrol --start

<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
```