



SHARE

Technology • Connections • Results

Linux on System z performance hints and tips

Session 2591

Martin Kammerer
kammerer@de.ibm.com

Feb 25, 2008 11:00 - 12:00



Trademarks



S H A R E

Technology • Connections • Results

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.

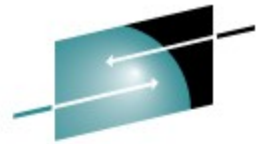
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available 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



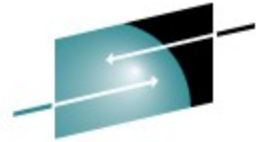
S H A R E

Technology • Connections • Results

- Tuning
 - Application
 - C/C++
 - Middleware
 - Java
 - Linux
 - Networking
 - Virtualization
 - Hardware / Setup

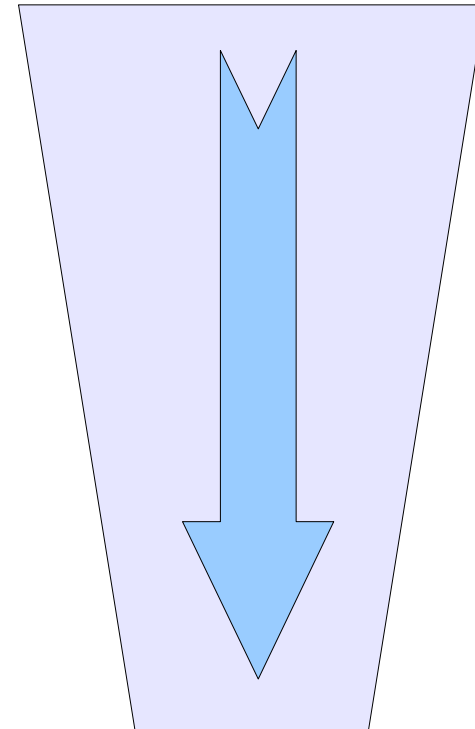
- Monitoring
 - Linux
 - z/VM

Optimize your stack in the right direction



SHARE
Technology • Connections • Results

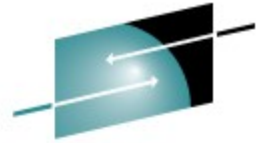
- Diminishing effect of tuning efforts
 - Application design
 - Application implementation
 - Middleware
 - Operating system
 - Virtualization layer
 - Hardware



Optimizing C/C++ code

- **Use -O3 optimization as default**
 - no debugging options
Further optimization:
 - architecture dependent options
 - **-march=values** <G5,z900,z990> <z9-109 with gcc-4.1>
 - **-mtune=values** <G5,z900,z990> <z9-109 with gcc-4.1>
 - inline assembler functions
- **Next step: application design**
 - dynamic or static linking
 - Avoid `-fPIC` for executables
 - right use of inlined C / C++ functions
- **Fine Tuning: additional general options on a file by file basis**
 - `-funroll-loops -ffast-math`

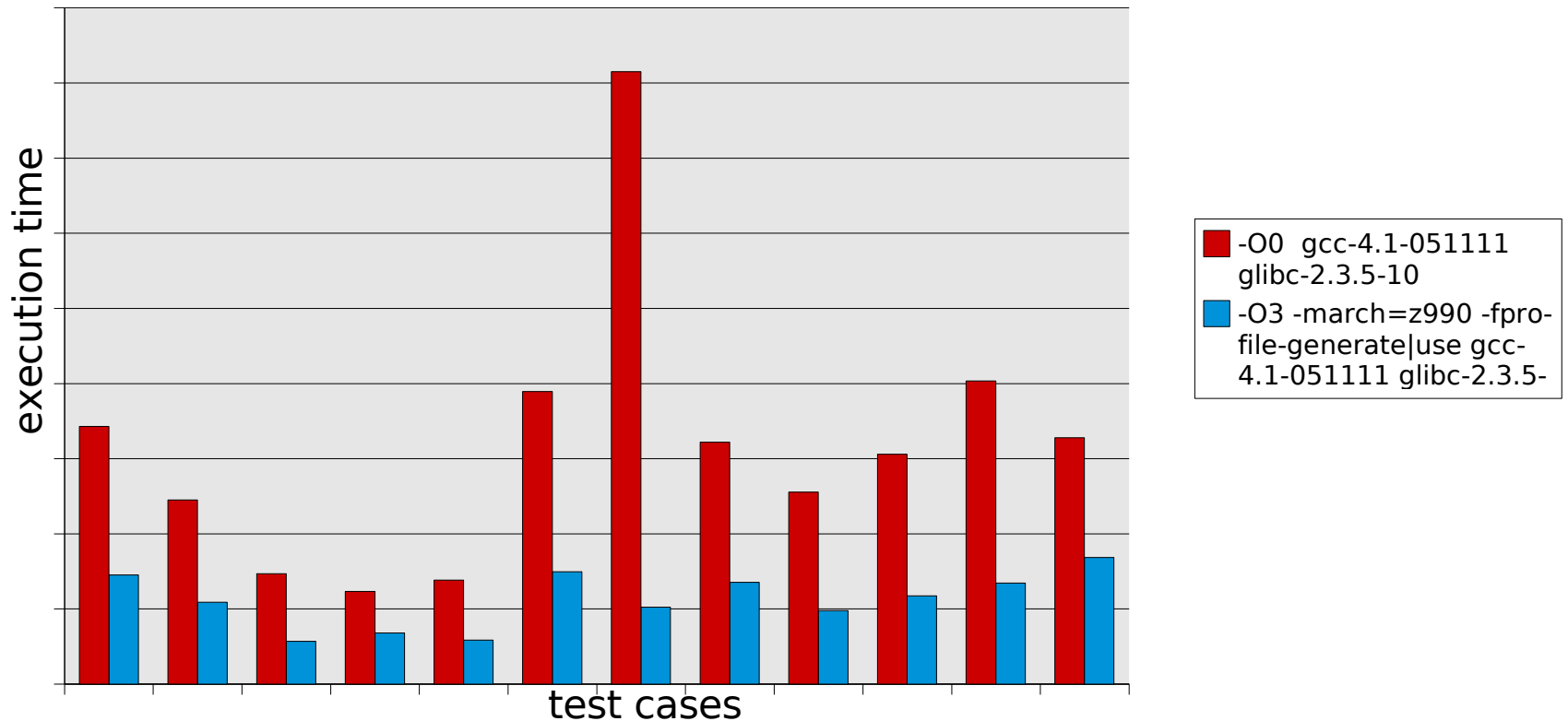
Results of changing compiler options



SHARE
Technology • Connections • Results

- Using -O3 instead of no optimization cuts runtime up to 50%

Comparison lower vs. higher optimization



Java basics

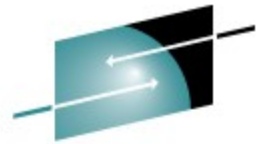


- Try to use latest Java version
 - Up to 20% release to release improvements
 - True as well for newer service releases (SR)
- Make sure that you've got enabled JIT
 - Verify Java output and look for “JIT enabled: jitc”
- Don't use Java in batch mode:
 - If you do 100 calls “java -jar myprogram.jar” you compile myprogram 100 times
 - can take more CPU power than the program itself
 - the JIT compiler can't do its optimization work
 - Instead pull the loop inside the Java program and call “java -jar myprogram100.jar” once

Java heap size

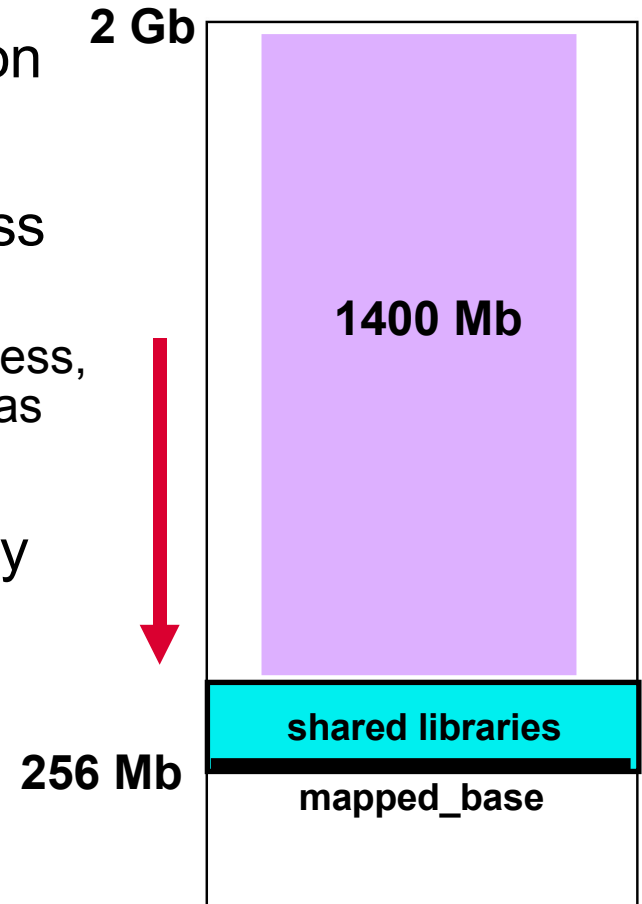
- Useful parameters
 - Setting heap size: -Xms (minimal), -Xmx (maximal), use min=max
 - -verbose:gc -- monitor GC
- Max heap \leq available memory
 - Avoid paging - Linux and VM
 - remember: heap memory will be used eventually!
- Larger heap size usually implies better performance
 - in 31bit SLES8, SLES9 & SLES10 use /proc/<pid>/mapped_base to define heaps up to 1.7 GB
 - In 31bit RHEL4 environments use flex-mmap mechanism
 - Watch out for prelinked applications!
 - Works also in 31bit emulation on 64 bit distros

Mapped_base HowTo



S H A R E
Technology • Connections • Results

- Only available for Novell distribution SLES8,9,10 (31 bit)
- PID is the process ID of the process you want to change
 - In bash \$\$ gives you the current process, from any process /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`
- Change value to e.g. 256 Mb by
`echo 268435456`
`>/proc/PID/mapped_base`



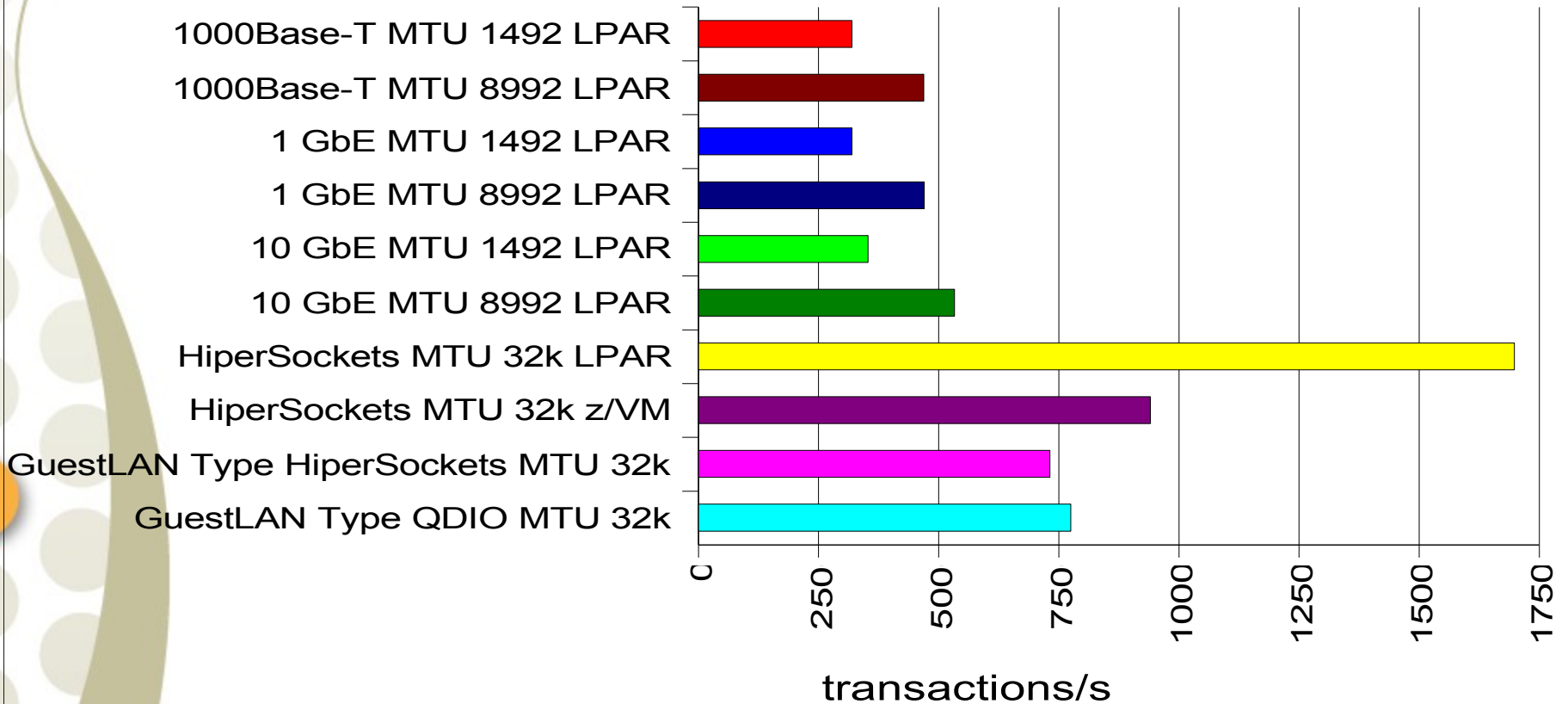
Networking performance



- Which connectivity to use:
 - External connectivity:
 - Use new 10 GbE cards with MTU 8992
 - Attach OSA directly to Linux guest image
 - Internal connectivity:
 - Hipersockets for LPAR-LPAR communication
 - Guest LAN for guest-guest communication
- For really busy network devices consider to
 - use channel bonding
 - Increase the number of inbound buffers in the qeth driver
 - Device has to be offline
 - `# echo <number> > /sys/bus/ccwgroup/drivers/qeth/<device_bus_id>/buffer_count`
- Channel bonding for HA creates only a small overhead
- Choose your MTU size carefully
 - Avoid fragmentation, lots of small packages can drive up CPU utilization

Networking throughput for various connection types

Website access example, 10 simultaneous connections (crr64x8k)
Linux to Linux, SLES10

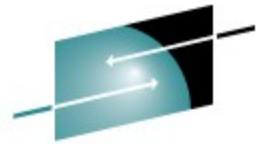


Networking throughput overview (SLES10)



	Online transaction (rr200x1000)	Database query (rr200x32k)	Website access (crr64x8k)	File transfer (strp, strg 20Mx20)
Advantage of large MTU size over default MTU size	equal	1.2x (1 GbE), 2.1x (10 GbE)	1.5x	3.4x (only 10 GbE)
Advantage of 10 GbE over 1 GbE	1.1x	1.9x (large MTU)	1.1x	3.3x (large MTU)
Advantage of GuestLAN over OSA	1.8x	3.4x (1 GbE), 1.7x (10 GbE)	1.4x	4.5x (1 GbE), 1.3x (10 GbE)
Fastest connection	HiperSockets LPAR	HiperSockets LPAR	HiperSockets LPAR	HiperSockets LPAR

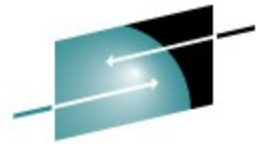
How to improve disk performance



S H A R E
Technology • Connections • Results

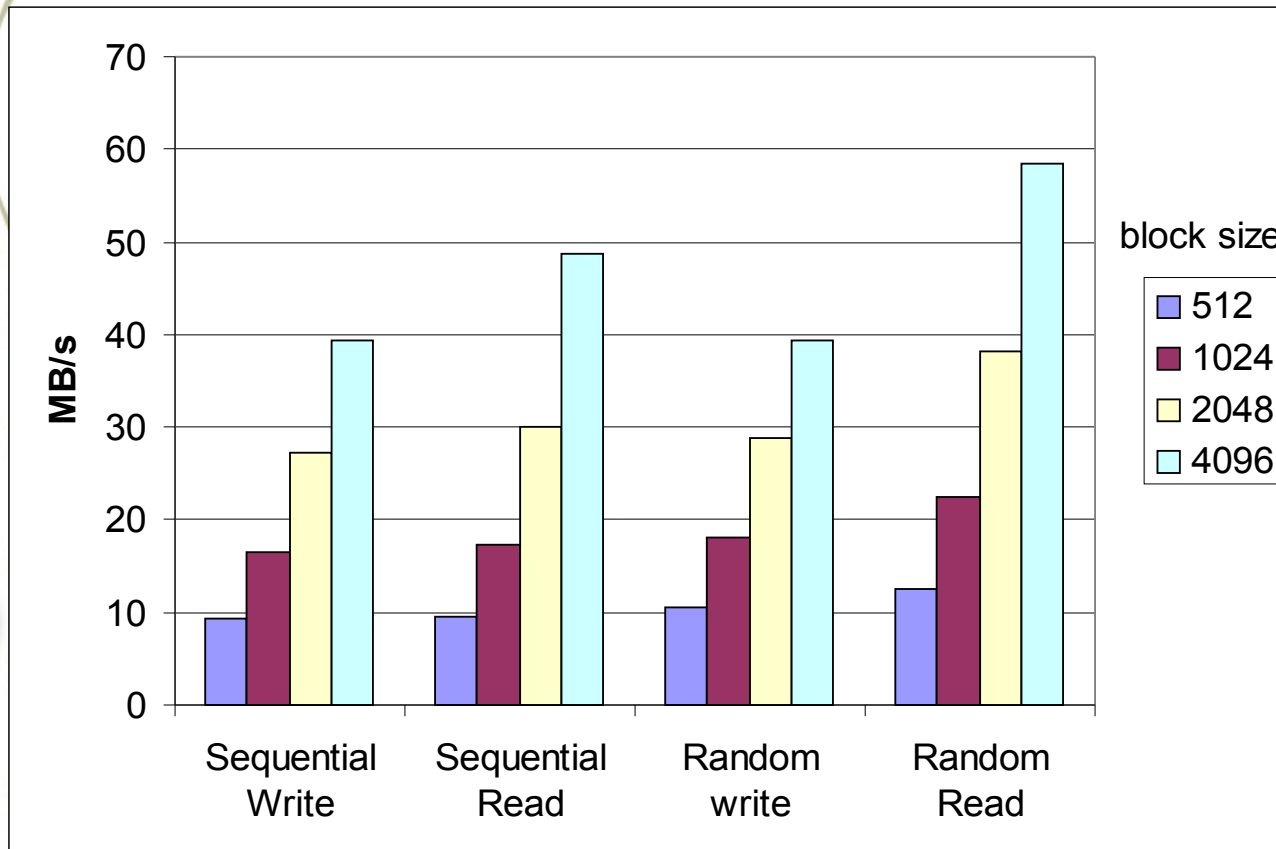
- Hardware choices
 - Use SCSI instead of ECKD
 - Use FICON instead of ESCON
 - 4Gb FICON > 2Gb FICON > 1Gb FICON
- Utilize your hardware
 - Use “striped” logical volumes from different ranks
 - Consider using PAV
 - Carefully set up your storage system
 - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_rec_dasd_optimizedisk.shtml

Effect of dasdfmt block size on throughput and capacity



SHARE
Technology • Connections • Results

- Use 4k block size on ECKD DASDs whenever possible !



dasdfmt blocksize	disk space
512b	3.5G
1024b	4.7G
2048b	6G
4096b	6.8G

z/VM 2 GB considerations

- **Solution:** upgrade z/VM to 5.2 or 5.3 level
- Read at
 - <http://www.vm.ibm.com/perf/tips/2gstorag.html>
 - <http://www.vm.ibm.com/perf/reports/zvm/html/64bit.html>
 - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_rec_fixed_io_t
- Old workarounds
 - **Cooperative Memory Management**
 - fixed I/O buffers with kernel 2.6 and ECKD
 - distribute your guests to multiple z/VMs
 - Move large guest to LPAR

“On Demand Timer” patch

- Linux uses HZ based timer interrupts
- Timer interrupts for idle guests create unnecessary overhead
- Starting with SLES8: enable & disable on the fly
 - `/proc/sys/kernel/hz_timer`
 - 1 = timer interrupts occurring every 10 ms
 - 0 = timer interrupts generated on demand only
- Included in SLES9, SLES10 and RHEL4, RHEL5 s390/s390x distributions

- **Problem:**
 - with many guests in z/VM it can happen that CP is busy executing diagnose instructions for the guest
- **What's behind it:**
 - in a so-called spin lock, Linux guests give their CPU share back to the hipervisor using DIAG 44
 - Hipervisor can be overloaded
- **Solution:**
 - Linux tries to get a lock n times before issuing a DIAG
 - Value of n is adjustable in `/proc/sys/kernel/spin_retry` (default 1000)
 - Included in latest SLES9 + SLES10 + RHEL4 + RHEL5

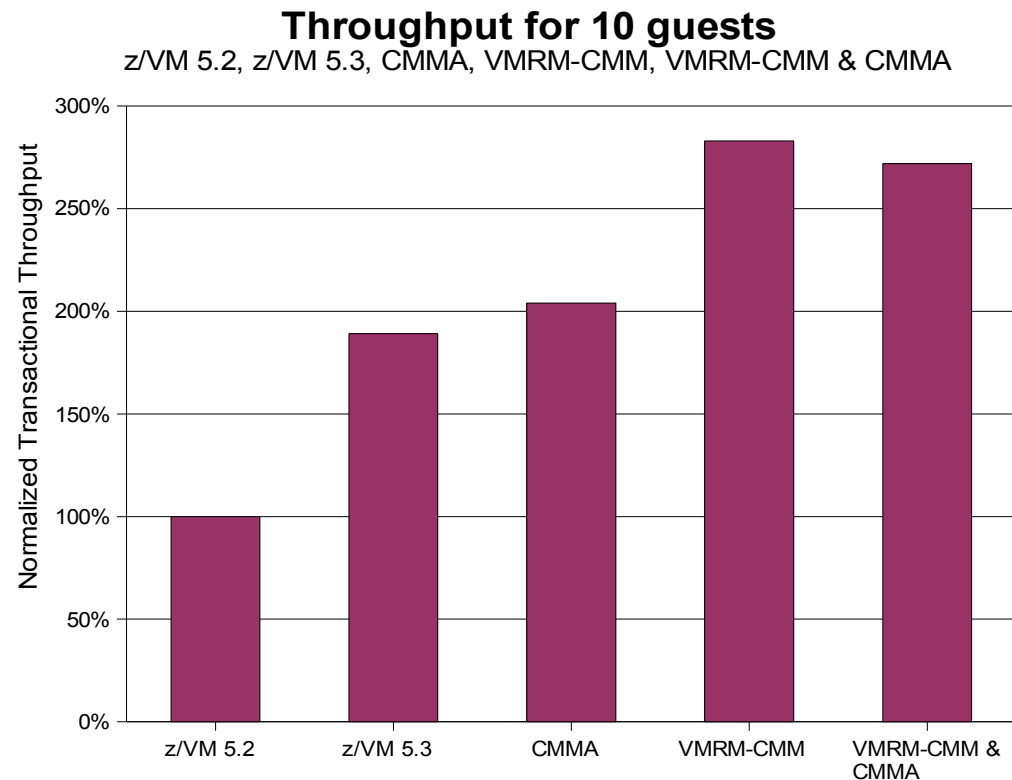
CMM



- 2 methods available:
 - VMRM-CMM (VM Resource Manager – Cooperative Memory Management) aka CMM1
 - Resource manager controls the size of the guests
 - CMMA (Collaborative Memory Management Assist) aka CMM2
 - Linux indicates which pages don't need to be saved
- Both methods show performance improvements when z/VM hits a system memory constraint.

CMM1 scenario

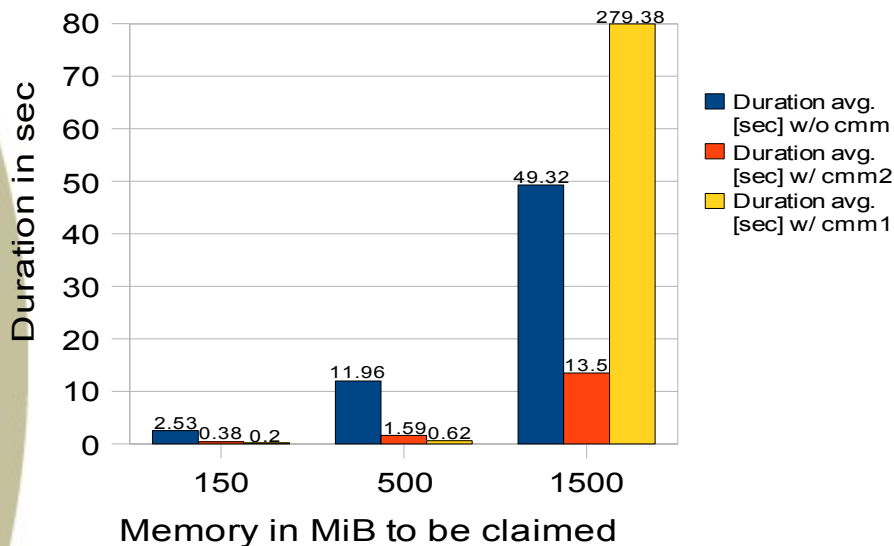
- Large Oracle guests, total used Linux memory = 2x of z/VM central storage, OLTP workload
- Advantages with CMM1
- Guests did not suffer from less page cache



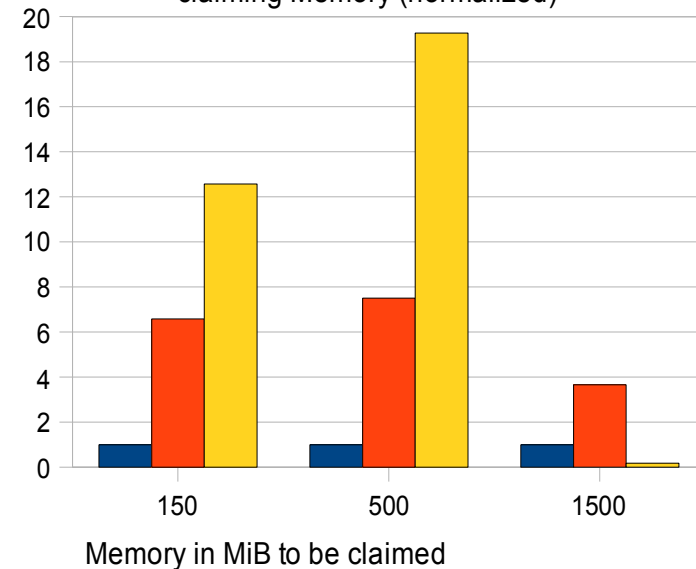
CMM2 scenario

- **Workload**
 - 15 guests, touching all their memory, all z/VM storage used. A guest orders now 150MB, 500MB, 1.5GB of memory. We measure the duration of this operation
- **Result**
 - In case of sudden memory claims CMM2 is the best choice

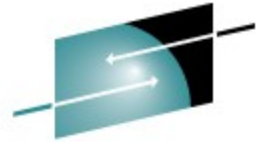
Duration of claiming Memory



Improvement factor for claiming Memory (normalized)



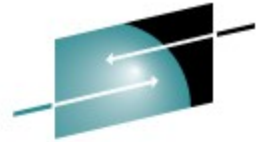
of CPUs per Linux image



SHARE
Technology • Connections • Results

- Use as few virtual CPUs as possible
- For LPAR definitions:
 - # all virtual CPUs : # real CPUs \leq 4:1
- For z/VM:
 - #of guest CPUs \leq #of CPUs for VM (LPAR)
- You don't get done more by defining more CPUs!

Agenda



SHARE
Technology • Connections • Results

- Tuning
 - Application
 - C/C++
 - Middleware
 - Java
 - Linux
 - Disk I/O
 - Networking
 - Virtualization
 - Hardware / Setup
- Monitoring
 - Linux
 - z/VM

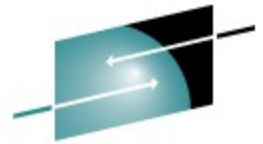
Linux command 'top' – the snapshot tool



- Adds new field “CPU steal time”
 - Is time Linux wanted to run, but the hipervisor was not able to schedule CPU
 - Is included in SLES10 and RHEL5

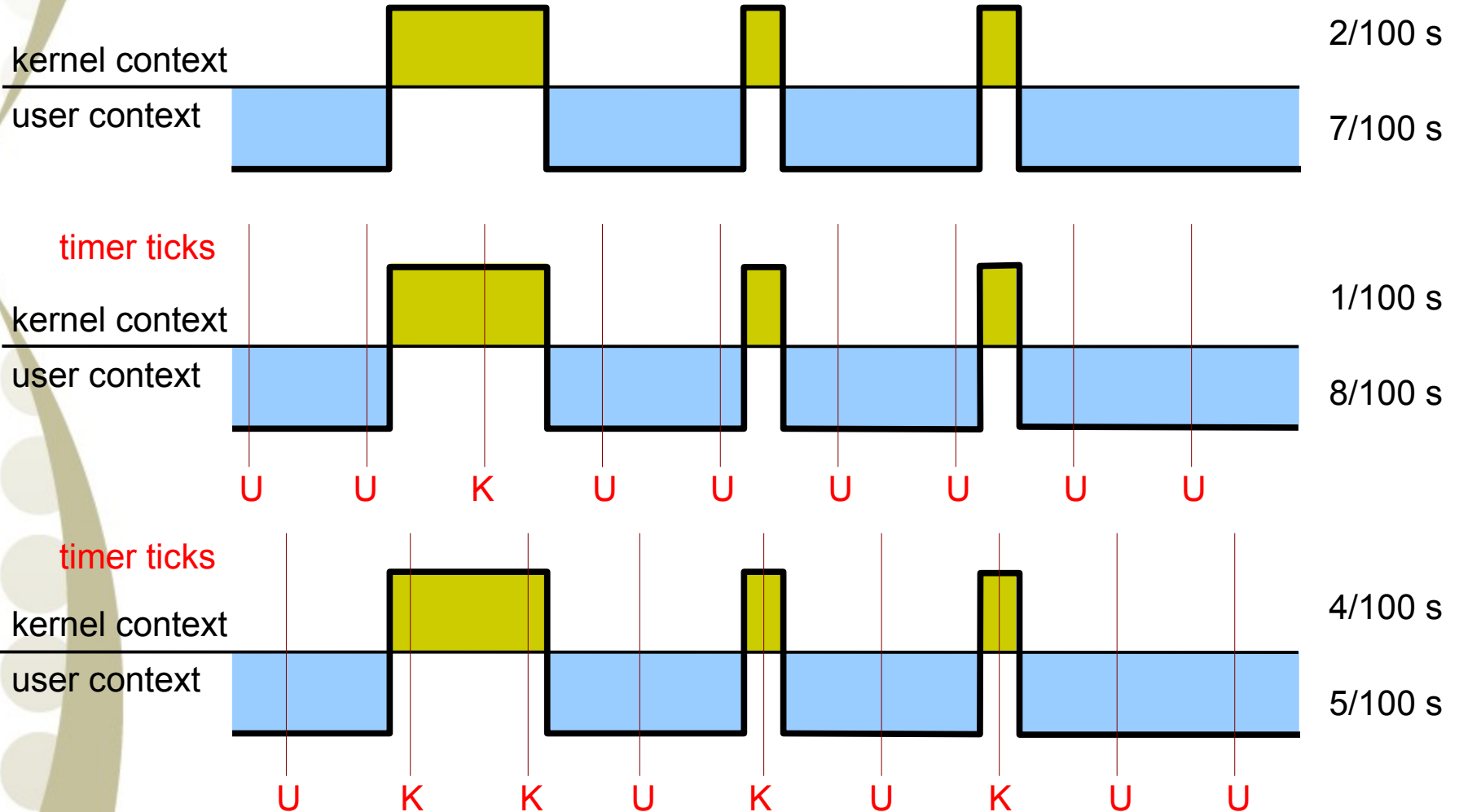
```
top - 09:50:20 up 11 min, 3 users, load average: 8.94, 7.17, 3.82
Tasks: 78 total, 8 running, 70 sleeping, 0 stopped, 0 zombie
Cpu0 : 38.7%us, 4.2%sy, 0.0%ni, 0.0%id, 2.4%wa, 1.8%hi, 0.0%si, 53.0%st
Cpu1 : 38.5%us, 0.6%sy, 0.0%ni, 5.1%id, 1.3%wa, 1.9%hi, 0.0%si, 52.6%st
Cpu2 : 54.0%us, 0.6%sy, 0.0%ni, 0.6%id, 4.9%wa, 1.2%hi, 0.0%si, 38.7%st
Cpu3 : 49.1%us, 0.6%sy, 0.0%ni, 1.2%id, 0.0%wa, 0.0%hi, 0.0%si, 49.1%st
Cpu4 : 35.9%us, 1.2%sy, 0.0%ni, 15.0%id, 0.6%wa, 1.8%hi, 0.0%si, 45.5%st
Cpu5 : 43.0%us, 2.1%sy, 0.7%ni, 0.0%id, 4.2%wa, 1.4%hi, 0.0%si, 48.6%st
Mem: 251832k total, 155448k used, 96384k free, 1212k buffers
Swap: 524248k total, 17716k used, 506532k free, 18096k cached
```

Tick based CPU Time inaccuracy

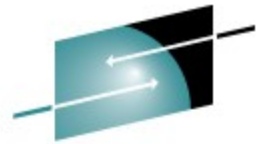


SHARE

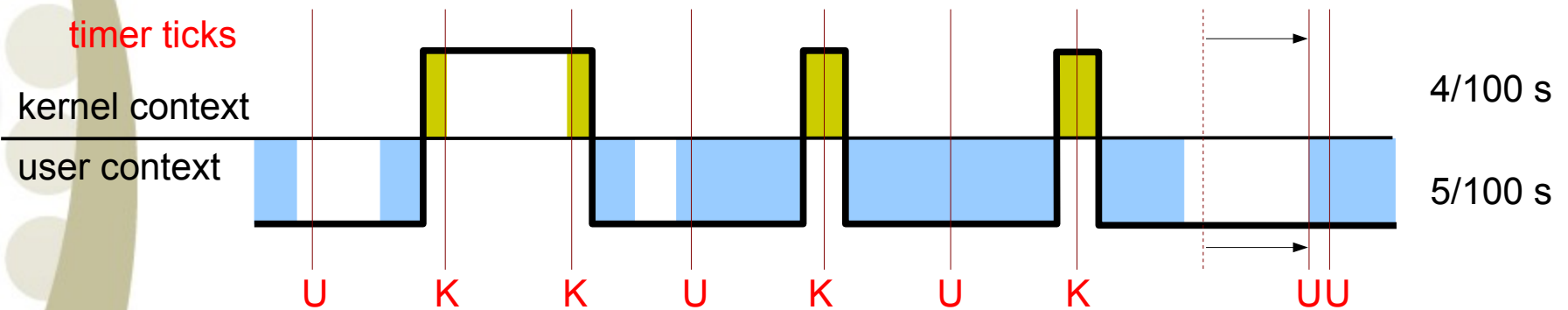
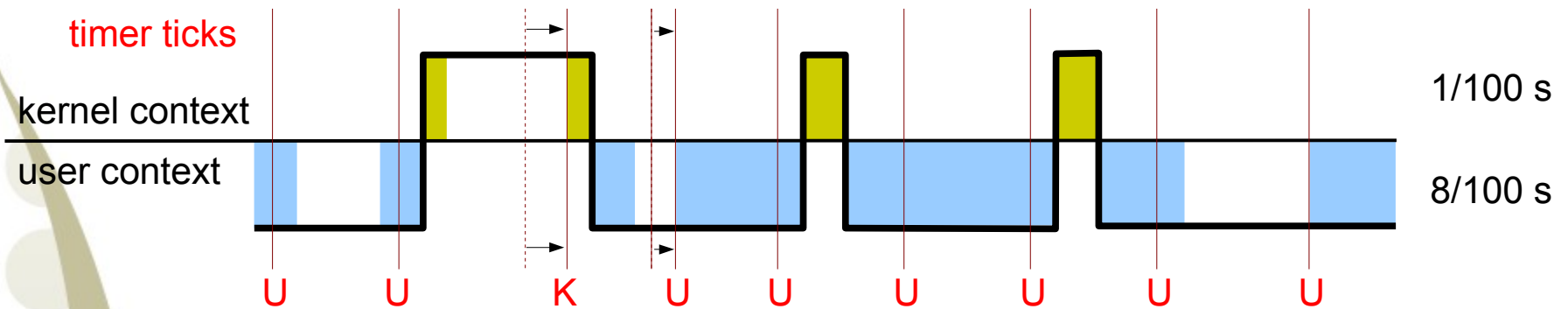
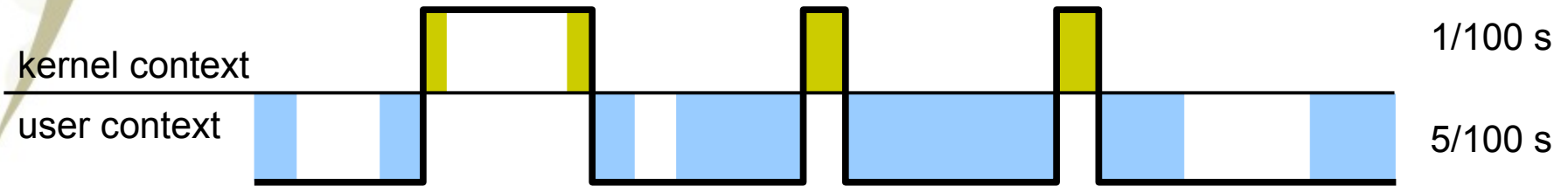
Technology • Connections • Results



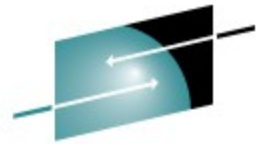
Tick based CPU accounting on virtual systems



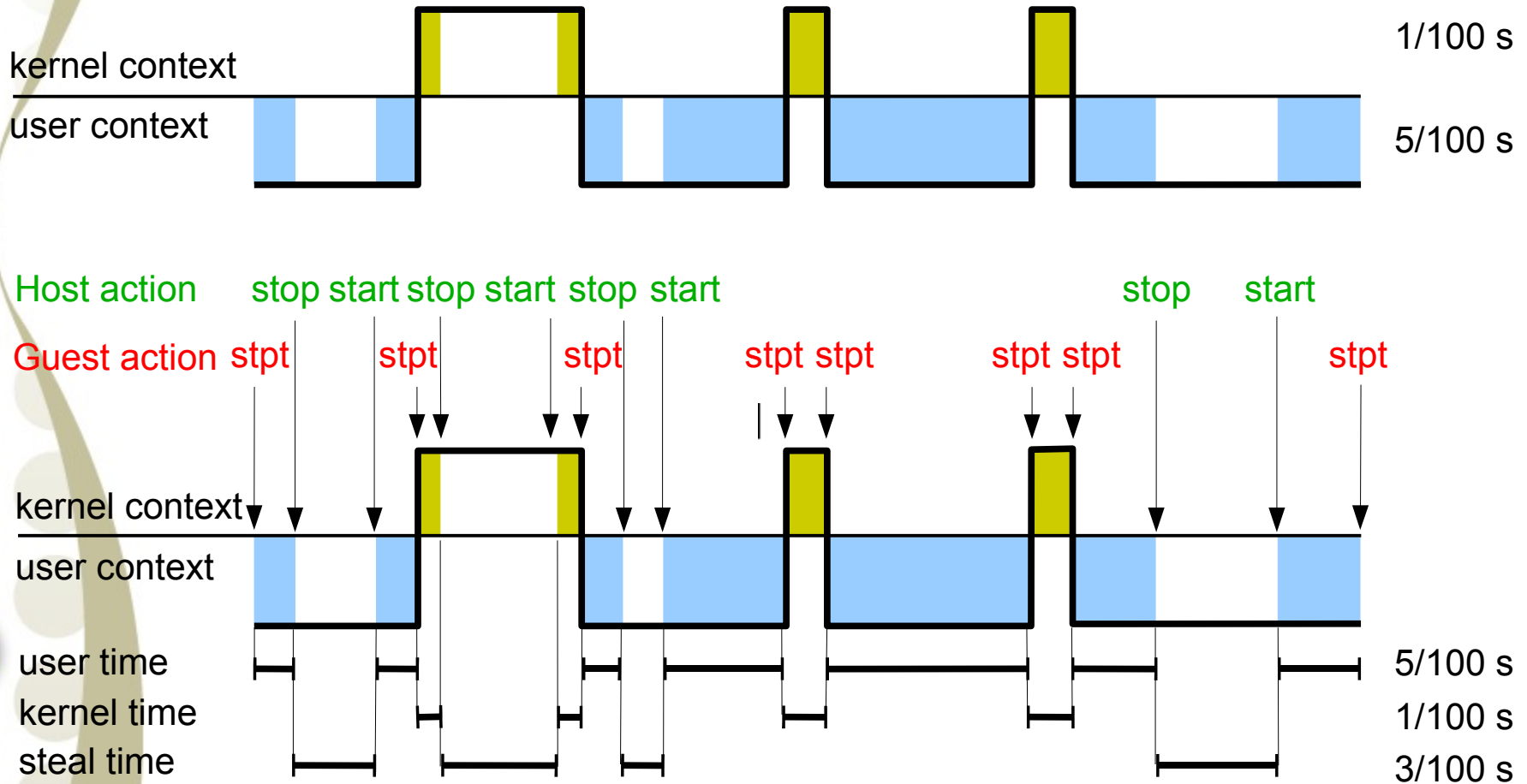
SHARE
Technology • Connections • Results



New Virtual CPU time accounting



SHARE
Technology • Connections • Results



stpt = Store CPU Timer

Sysstat – the 'long' term data collection



- Contains four parts
 - sadc: data gatherer - stores data in binary file
 - Sar: reporting tool - reads binary file and converts it to readable output
 - mpstat processor utilization
 - iostat I/O utilization
- “steal time” included starting version 7.0.0
- Install the sysstat package and configure it depending on your distribution (crontab)
 - by default data is collected in /var/log/sa
- More info at: <http://perso.orange.fr/sebastien.godard> and with “man sar” on your system

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-guide/ch>

Oprofile – short how-to

- Example from http://www.ibm.com/developerworks/linux/linux390/perf/tuning_how_tools.html#op

```
sysctl -w kernel.hz_timer=1
```

```
gunzip /boot/vmlinux-2.6.5-7.201-s390x.gz
```

```
opcontrol --vmlinux=/boot/vmlinux-2.6.5-7.201-s390x
```

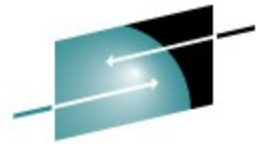
```
opcontrol --start
```

```
<DO TEST>
```

```
opcontrol --shutdown
```

```
opreport
```

Oprofile – output example



SHARE

Technology • Connections • Results

CPU: CPU with timer interrupt, speed 0 MHz (estimated)

Profiling through timer interrupt

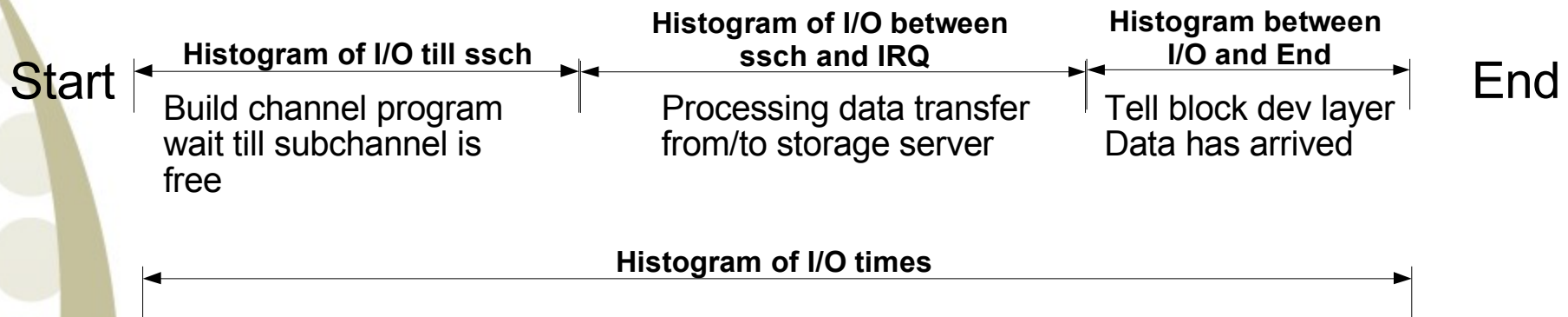
vma	samples	%	app name	symbol name
80002840	5862	34.8970	mcf_base.z_Linux	price_out_impl
800012c8	5221	31.0811	mcf_base.z_Linux	refresh_potential
80003cb4	4398	26.1817	mcf_base.z_Linux	primal_bea_mpp
80003b60	408	2.4289	mcf_base.z_Linux	sort_basket
0001a67c	345	2.0538	vmlinux	default_idle
800013d8	138	0.8215	mcf_base.z_Linux	flow_cost
800033bc	98	0.5834	mcf_base.z_Linux	update_tree
800020f8	88	0.5239	mcf_base.z_Linux	dual_feasible
800036a4	72	0.4286	mcf_base.z_Linux	primal_iminus
8000323c	40	0.2381	mcf_base.z_Linux	write_circulations
80002720	24	0.1429	mcf_base.z_Linux	insert_new_arc

/proc/dasd/statistics (1)

- Linux can collect performance stats on DASD activity as seen by Linux(!)
- Turn on with
`echo on > /proc/dasd/statistics`
- Turn off with
`echo off > /proc/dasd/statistics`
- To reset: turn off and then on again
- Can be read for the whole system by
`cat /proc/dasd/statistics`
- Can be read for individual DASDs by
`tunedasd -P /dev/dasda`

/proc/dasd/statistics (2)

- Collects statistics (mostly processing times) of IO operations
- Each line represents a histogram of times for a certain operation
- Operations split up into the following :



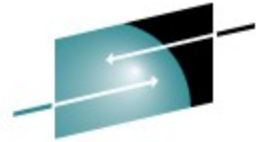
SCSI statistics (1)

- In SLES9 and SLES10 SCSI statistics can be collected
- The parameter `CONFIG_STATISTICS=y` must be set in the kernel config file
- If `debugfs` is mounted at `/sys/kernel/debug/` , all the statistics data collected can be found at `/sys/kernel/debug/statistics/`
- The names of these subdirectories consist of `zfcplib-<device-bus-id>` for an adapter and `zfcplib-<device-bus-id>-<WWPN>-<LUN>` for a LUN.

SCSI statistics (2)

- Each subdirectory contains two files, a data and a definition file.
- Using
`echo on=1 > definition`
the data gathering can be switched on for each device,
- With
`echo on=0 > definition`
the gathering is switched off again. It defaults to data gathering being turned off.
- The command
`echo data=reset > definition`
enables you to reset the collected data to 0.

SCSI statistics example



S H A R E

Technology • Connections • Results

```
cat /sys/kernel/debug/statistics/zfcp-0.0.1700-0x5005076303010482-0x4014400500000000/data
```

```
...  
request_sizes_scsi_read 0x1000 1163  
request_sizes_scsi_read 0x80000 805  
request_sizes_scsi_read 0x54000 47  
request_sizes_scsi_read 0x2d000 44  
request_sizes_scsi_read 0x2a000 26  
request_sizes_scsi_read 0x57000 25  
request_sizes_scsi_read 0x1e000 25  
request_sizes_scsi_read 0x63000 24  
request_sizes_scsi_read 0x6f000 19  
request_sizes_scsi_read 0x12000 19
```

```
...  
latencies_scsi_read <=1 1076  
latencies_scsi_read <=2 205  
latencies_scsi_read <=4 575  
latencies_scsi_read <=8 368  
latencies_scsi_read <=16 0
```

```
...  
channel_latency_read <=16000 0  
channel_latency_read <=32000 983  
channel_latency_read <=64000 99  
channel_latency_read <=128000 115  
channel_latency_read <=256000 753  
channel_latency_read <=512000 106  
channel_latency_read <=1024000 141  
channel_latency_read <=2048000 27  
channel_latency_read <=4096000 0
```

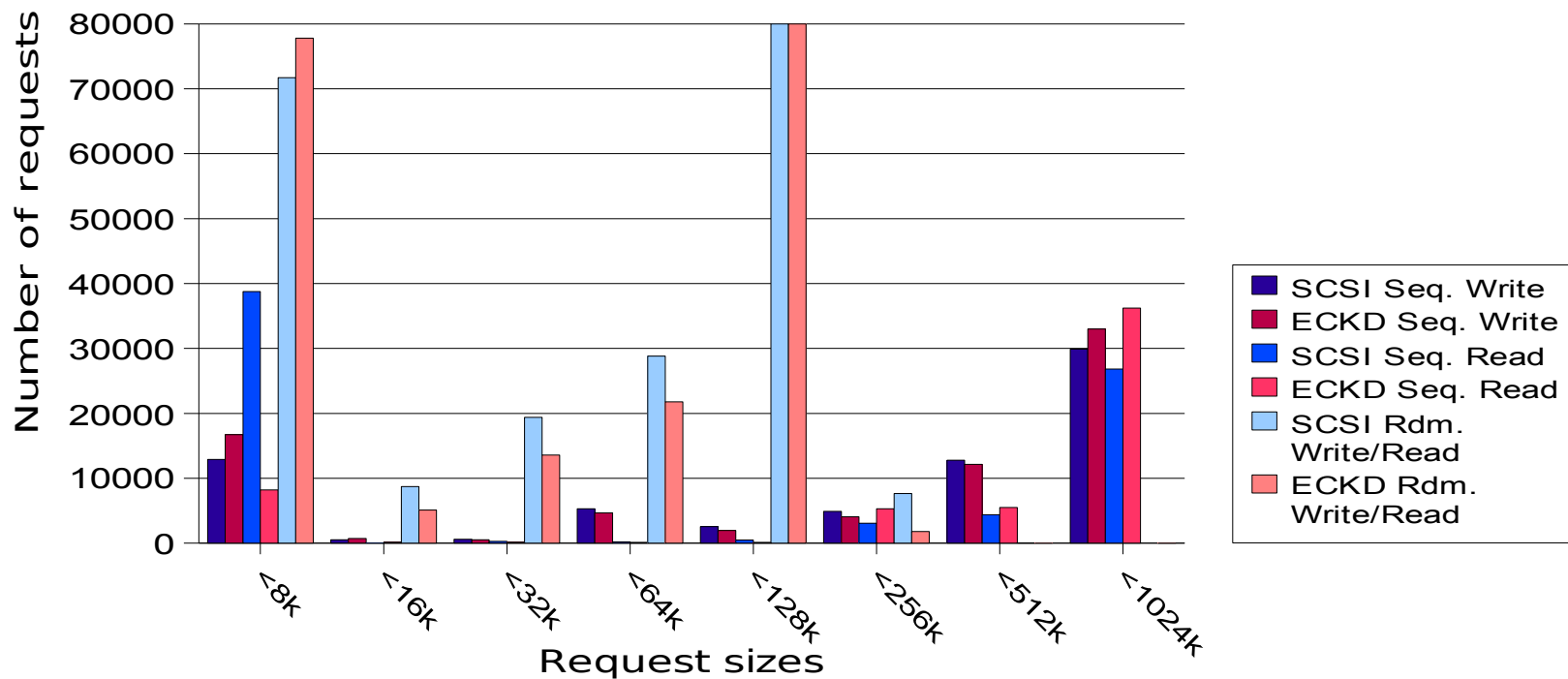
```
...  
fabric_latency_read <=1000000 1238  
fabric_latency_read <=2000000 328  
fabric_latency_read <=4000000 522  
fabric_latency_read <=8000000 136  
fabric_latency_read <=16000000 0
```

```
...
```

Comparing SCSI and ECKD request sizes

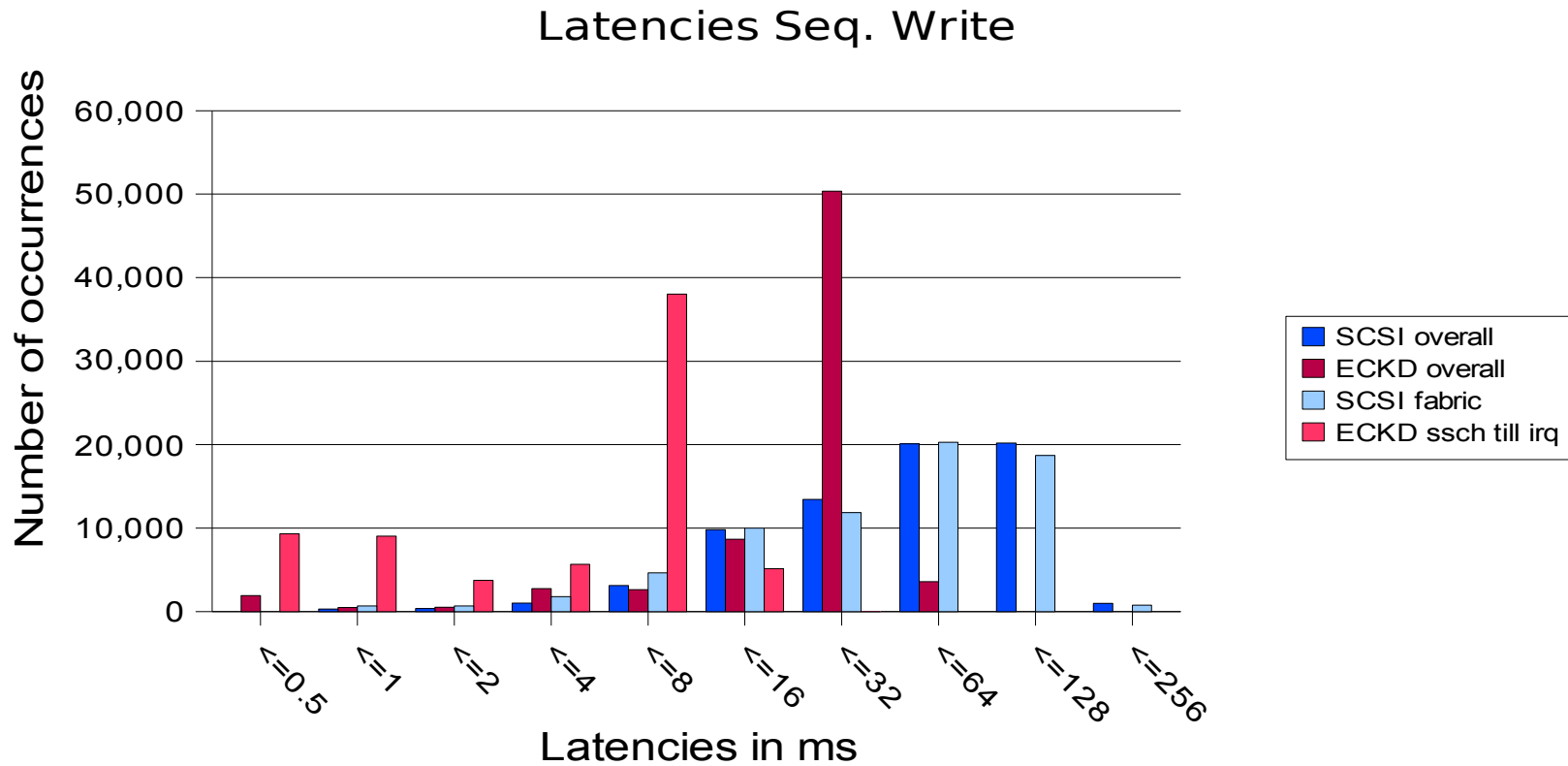
- Similar request sizes for sequential and random I/O

Request sizes (IOzone 16 processes)

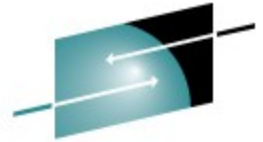


Comparing SCSI and ECKD latencies (1)

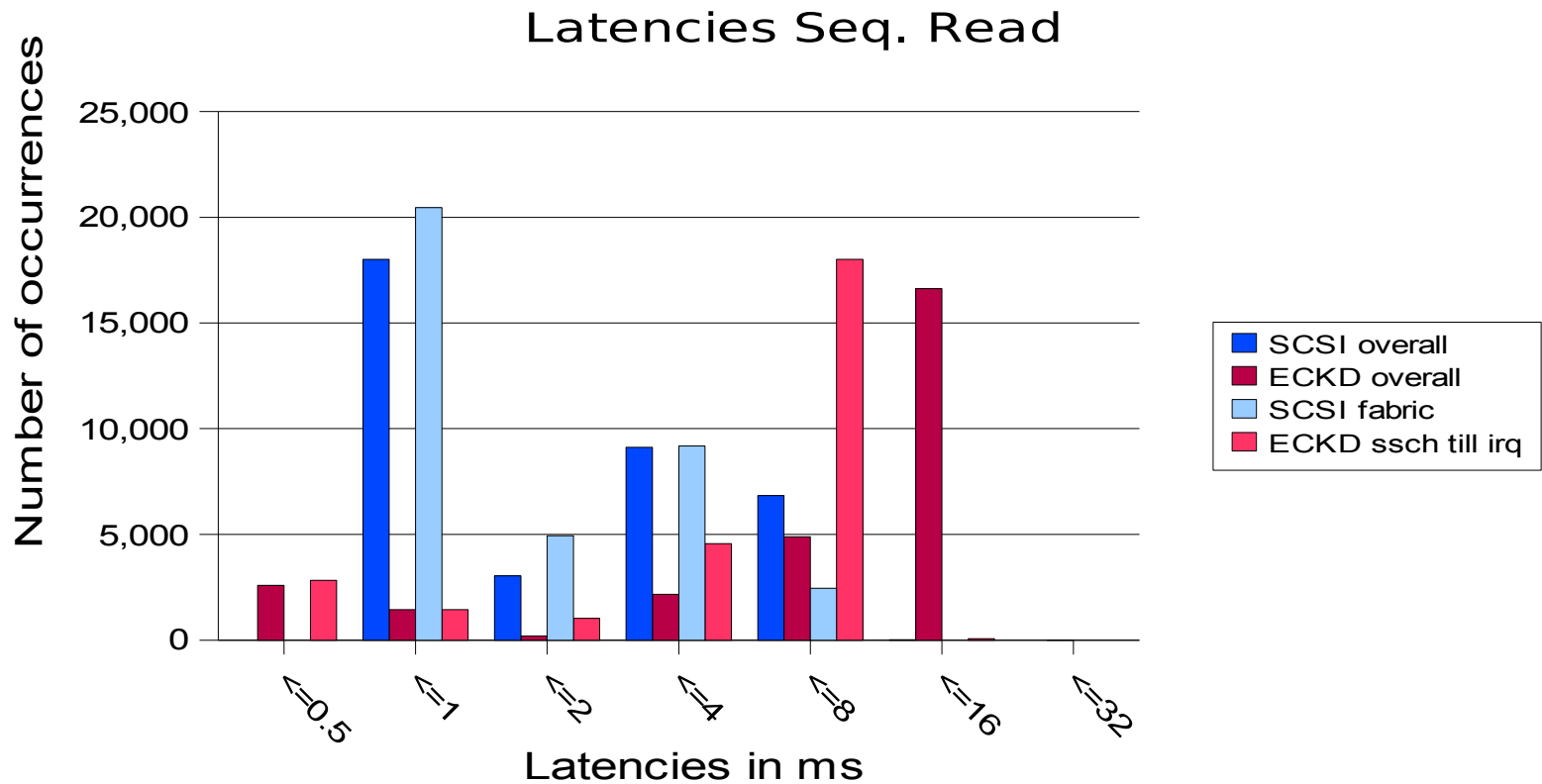
- SCSI sequential write latencies are longer



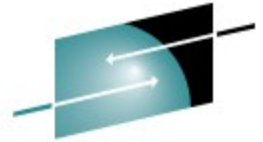
Comparing SCSI and ECKD latencies (2)



- SCSI sequential read latencies are shorter



How to collect z/VM monitor data



SHARE
Technology • Connections • Results

- Cheat Sheet at: <http://www.vm.ibm.com/perf/tips/collect.html>
- 5 basic steps
 - Create monitor DCSS
 - Setup userid to issue monwrite command
 - Start and configure monitor
 - Start monwrite
 - Stop monwrite and save data

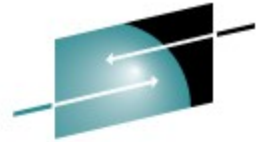
How to insert Linux data in z/VM monitor stream



- Enable your guest for inserting data into the monitor stream
 - set APPLMON option to user direct
- Insert Linux modules
 - `modprobe appldata_mem`
 - `modprobe appdata_os`
 - `modprobe appldata_net_sum`
- Turn on monitoring
 - `echo 1 > /proc/sys/appldata/timer`
 - `echo 1 > /proc/sys/appldata/mem`
 - `echo 1 > /proc/sys/appldata/os`
 - `echo 1 > /proc/sys/appldata/net_sum`
- Details can be found in chapter 15 of Device Drivers, Features, and Commands (SC33-8281-02)

<http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/docu/I26bdd02.pdf>

Visit us !



S H A R E
Technology • Connections • Results

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

Questions



SHARE

Technology • Connections • Results

