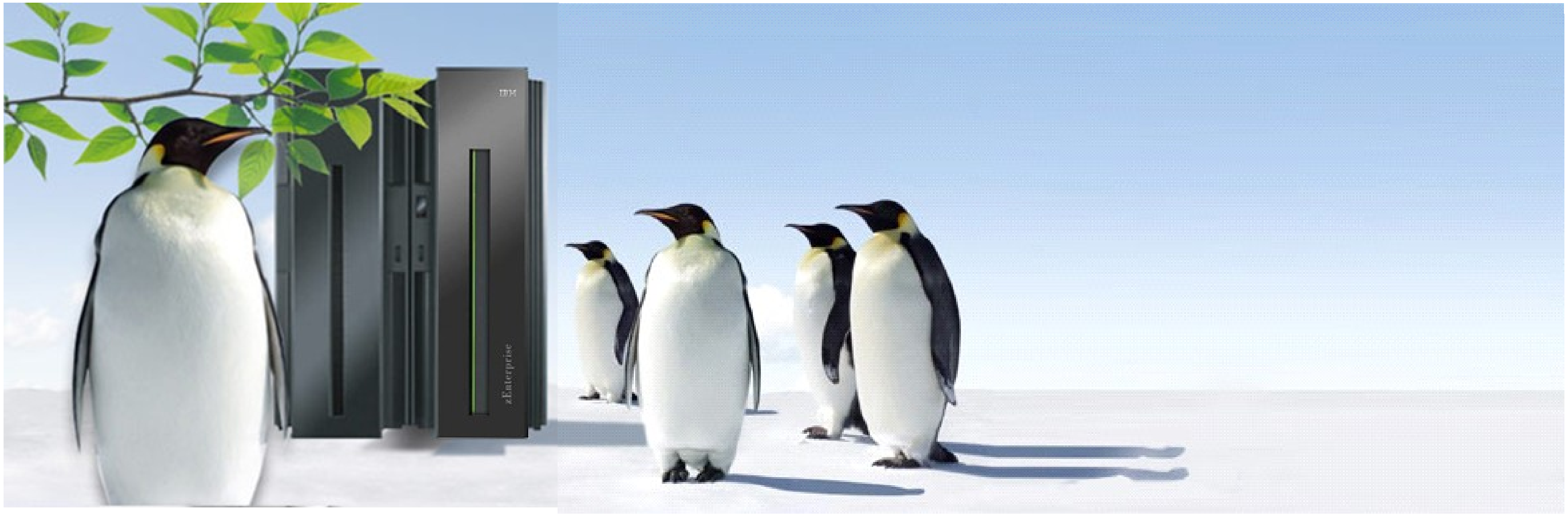
Susanne Wintenberger – Certified IT Specialist Linux on System z <swinten@de.ibm.com>
15./16. May 2013

Hints & Tips for Solving Linux on System z Problems with Customer Cases



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

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

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

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

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

- Introduction
- How to help us to help you
- System monitoring
- How to dump a Linux on System z
- Some real customer cases

Introductory Remarks

- Looks straight forward on the charts, ...
  - But a problem does not necessarily show up on the place of origin
  - Analysis can take weeks
    - Starts to look simple once you know the solution
  - Memory overwrites as an example
    - Can cause symptoms anywhere

- More information → faster problem resolution
  - Gathering and submitting additional information introduces delays.
  - Having a structured process for yourself eases a service request if needed

Trouble Shooting First Aid Kit – be prepared

- Install some packages required for debugging
  - s390-tools/s390-utils
    - dbginfo.sh
  - sysstat
    - sadc/sar
    - iostat
  - dump tools crash / lcrash
    - lcrash (lkcdutils) available with SLES10
    - crash available on SLES11
    - crash in all RHEL distributions
  - Use these pro-actively in healthy system as well

dbginfo script

- It collects various system-related files for debugging purposes.
  - It captures the current system environment and generates a tar file, which can be attached to PMRs / Bugzilla entries
- part of the s390-tools package in SUSE and s390-utils package in recent Red Hat distributions
  - dbginfo.sh gets continuously improved by service and development
  - Check out: http://www.ibm.com/developerworks/linux/linux390/s390-tools.html
- In order to run the script properly
  - Ensure that it is run as root user.
  - Under z/VM, the appropriate privilege classes help to be authorized for some used commands (e.g. privilege class B)
- It is similar to the Red Hat tool sosreport or to the SUSE tool supportconfig

```
root@larsson:~>  dbginfo.sh
Create target directory /tmp/DBGINFO-2009-04-15-22-06-20-t6345057
Change to target directory /tmp/DBGINFO-2009-04-15-22-06-20-t6345057
[...]
```

dbginfo script (cont'd)

- dbginfo.sh captures the following information:
  - /proc/[version, cpu, meminfo, slabinfo, modules, partitions, devices ...]
  - System z specific device driver information: /sys/kernel/debug/s390dbf
  - Kernel messages /var/log/messages
  - Reads configuration files in directory /etc/ [ccwgroup.conf, fstab ...]
  - Uses several commands: ps, dmesg
  - Query setup scripts: lscss, lsdasd, lsqeth, lszfcp, lstape, ...
  - And much more
- If the Linux system runs as z/VM guest operating system, dbginfo collects information about the z/VM guest setup:
  - Release and service Level: q cplevel
  - Network setup: q [lan, nic, vswitch, v osa, ...]
  - Storage setup: q [set, v dasd, v fcp, q pav ...]
  - Configuration/memory setup: q [stor, v stor, xstore, cpus...]

Describe the system

- Describe the software setup
  - What is the System/Workload intended to do ?
  - What software (versions) are used for that ?
    - System (Distribution)
    - Middle-ware components

- Describe the hardware setup
  - Machine and Storage type
  - Storage and Network attachments

- Describe the infrastructure setup
  - Clients
  - Network topology (firewalls, devices, vswitches, vlans, …)
  - Disk configuration (multipath, lvm, storage server setup, …)

Trouble Shooting First Aid Kit - emergency

- General
  - Collect dbginfo.sh output then compare with healthy systems log
  - increase log level in /sys/kernel/debug/s390dbf for affected subsystems
- In case of a performance problem
  - Always archive syslog (/var/log/messages)
  - Start sadc (System Activity Data Collection) and provide sar files
  - If running as guest under z/VM, collect z/VM MONWRITE data
  - Periodically, collect and archive some data during your peak periods, so that you have a historical record
    - Peak loads
    - month-end processing
    - Significant changes (e.g. moving from z10 to z196, refreshing level of application code)

Trouble Shooting First Aid Kit – emergency (cont'd)

- In case of a disk problem
  - Enable disk statistics
- In case of a network problems
  - Provide a diagram of your network setup
  - Run lsqeth (part of s390-tools package)
- In case of a system hangs
  - Take a kernel dump
    - Include System.map, Kerntypes (if available) and vmlinux file
  - See “Using the dump tools” book on http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/docu/l26ddt02.pdf

System z debug feature (s390dbf traces)

- System z specific driver tracing environment
  - Uses ring buffers
  - Available in live system and in system dumps
- Must be mounted for live view:
  - '`mount -t debugfs /sys/debug /sys/kernel/debug`'
- Each component has these control interfaces
  - level – controlling the trace detail between 0 <-> 6 (lowest-highest) default: 2
    - Increase pages when logging with high levels: '`echo 6 > level`'
  - pages – shows and defines the preallocated space: '`echo 20 > pages`'
  - flush – cleans the ring buffer: '`echo 1 > flush'`
- And one of these output files
  - hex_ascii – output is not that human readable, but very useful for debugging
  - sprintf – human readable output, usually an event log

```
cat /sys/kernel/debug/s390dbf/qeth_msg/sprintf
00 01289399222:389736 5 - 01 000003c01956f346  IPA: delipm(xB5) for eth1 succeeded
00 01289399222:390166 5 - 01 000003c01956f346  IPA: destroy_addr(xC4) for eth1 succeeded
00 01289399224:977051 5 - 01 000003c01956f346  IPA: qipassist(xB2) for eth1 succeeded
```

Describe the problem

- What is the symptom ?
  - When did it happen ?
    - Date and time, important to dig into logs
    - How frequently does it occur ?
    - Is there any pattern ?
  - Is this a first time occurrence ?
    - Was anything changed recently ?
    - Diffs of dbginfo can save your day
  - Where did it happen ?
    - One or more systems, production or test environment ?
  - Is the problem reproducible ?

- Write down as much as possible information about the problem !

Trouble Shooting First Aid Kit - report

- Problem report
  - Provide your problem and environment description
  - Attach the output file of dbginfo.sh, any (performance) reports or logs
  - Upload dump data
  - Use meaningful names for the output files (e.g. tool_test_case_date_and_time)
  - z/VM MONWRITE data
    - Binary format, make sure, record size settings are correct.
    - For details see http://www.vm.ibm.com/perf/tips/collect.html
- When opening a PMR
  - Upload comprehensive documentation to directory associated to your PMR at
    - ftp://ecurep.ibm.com/, or ftp://testcase.boulder.ibm.com/
  - See Instructions: http://www.ibm.com/de/support/ecurep/other.html
- If opening multiple partner tickets, let them know about each other
- When opening a Bugzilla (bug tracker web application) at distribution partner attach documentation to Bugzilla

System Monitoring

sadc/sar

- Capture Linux performance data with sadc/sar
  - CPU utilization
  - Disk I/O overview and on device level
  - Network I/O and errors on device level
  - Memory usage/swapping
  - Reports statistics data over time and creates average values for each item
- sadc example (for more see man sadc)
  - System Activity Data Collector (sadc) --> data gatherer
  - **`/usr/lib64/sa/sadc [options] [interval [count]] [binary outfile]`**
  - `/usr/lib64/sa/sadc 10 20 sadc_outfile`
  - `/usr/lib64/sa/sadc -d 10 sadc_outfile`
  - -d option: collects disk statistics
  - Choosing the right interval can be important
    - Too small → too much data & overhead, can mask the issue
    - Too large → values are too “averaged”, peaks no more visible

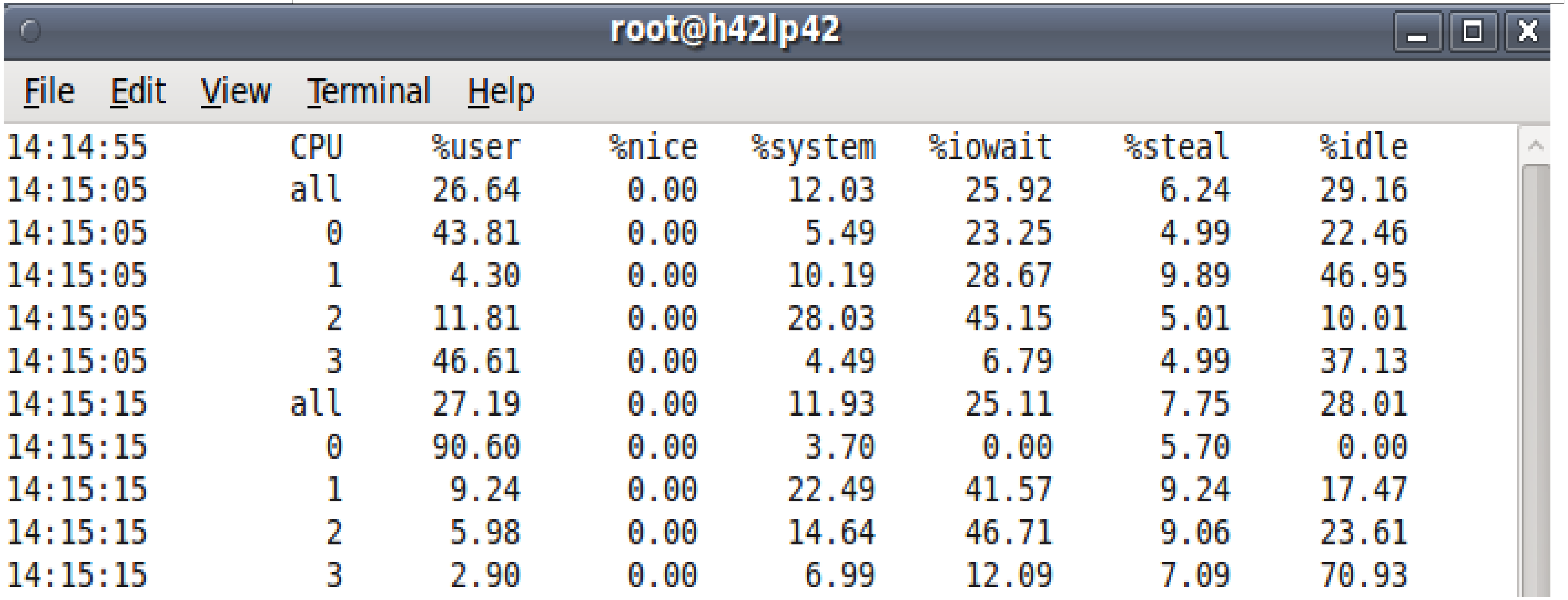
sadc/sar (cont'd)

- sar example (for more see man sar)
  - System Activity Report (sar) command --> reporting tool
  - **`sar [options] sadc_outfile > [sar outfile]`**
  - `sar -A -f sadc_outfile > sar_outfile`
  - -A option: reports all the collected statistics
  - -f option: specifies the binary sadc output file
  - enables the creation of item specific reports e.g. network
  - enables the specification of a start and end time → averages are created for the time of interest
- Should be started as a service during system start e.g.

  `'service sysstat start'`
- Please always include both the sadc and the 'sar -A' files when submitting SAR information to IBM support
  - This often allows to verify/falsify conclusions seen in other parts of the report

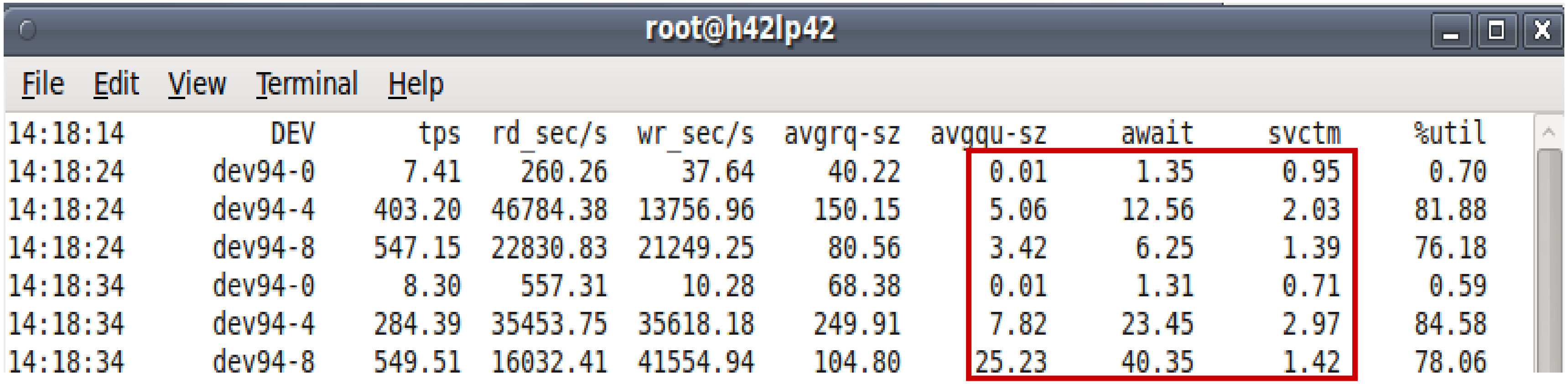
CPU utilization

```
Per CPU values:
watch out for
       system time (kernel time)
       iowait time (runnable, but waiting for I/O)
       steal time (runnable, but time taken by
other guests)
```



```
14:14:55    CPU   %user   %nice  %system  %iowait   %steal   %idle
14:15:05    all   26.64    0.00    12.03    25.92     6.24   29.16
14:15:05      0   43.81    0.00     5.49    23.25     4.99   22.46
14:15:05      1    4.30    0.00    10.19    28.67     9.89   46.95
14:15:05      2   11.81    0.00    28.03    45.15     5.01   10.01
14:15:05      3   46.61    0.00     4.49     6.79     4.99   37.13
14:15:15    all   27.19    0.00    11.93    25.11     7.75   28.01
14:15:15      0   90.60    0.00     3.70     0.00     5.70    0.00
14:15:15      1    9.24    0.00    22.49    41.57     9.24   17.47
14:15:15      2    5.98    0.00    14.64    46.71     9.06   23.61
14:15:15      3    2.90    0.00     6.99    12.09     7.09   70.93
```

Disk I/O I – per device



```
14:18:14         DEV      tps   rd_sec/s   wr_sec/s   avgrq-sz   avgqu-sz     await     svctm     %util
14:18:24     dev94-0     7.41     260.26      37.64      40.22       0.01      1.35      0.95      0.70
14:18:24     dev94-4   403.20   46784.38   13756.96     150.15       5.06     12.56      2.03     81.88
14:18:24     dev94-8   547.15   22830.83   21249.25      80.56       3.42      6.25      1.39     76.18
14:18:34     dev94-0     8.30     557.31      10.28      68.38       0.01      1.31      0.71      0.59
14:18:34     dev94-4   284.39   35453.75   35618.18     249.91       7.82     23.45      2.97     84.58
14:18:34     dev94-8   549.51   16032.41   41554.94     104.80      25.23     40.35      1.42     78.06
```

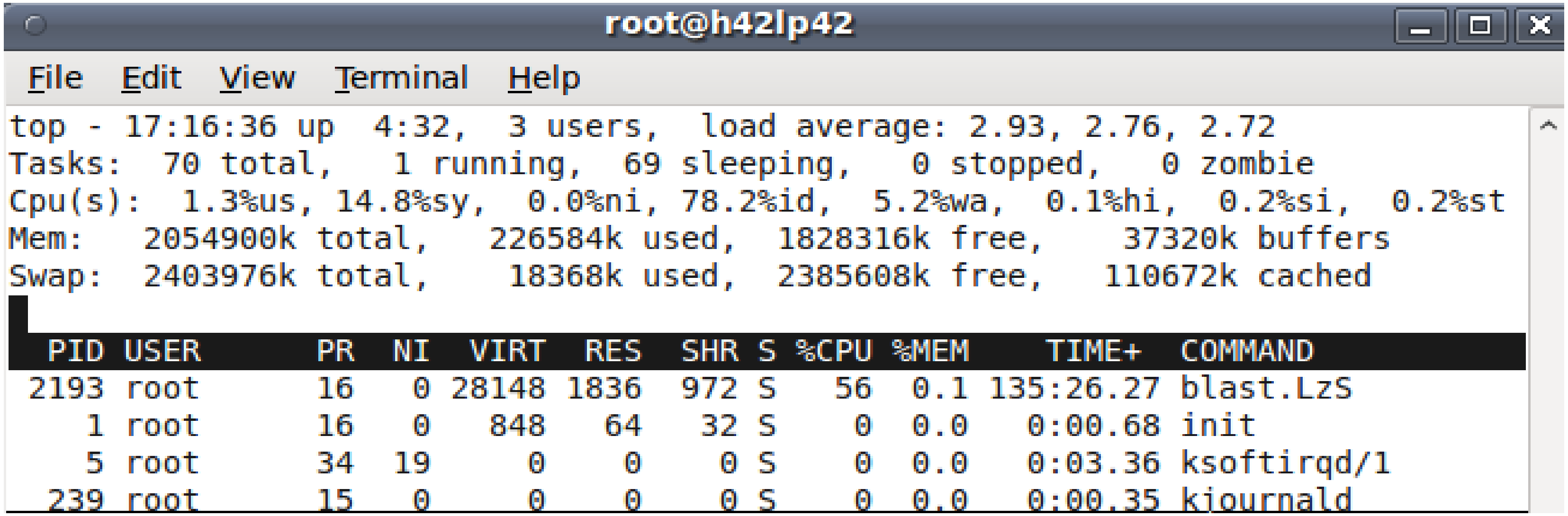
avgqu-sz: average length of queue, how many i/o requests are not dispatched

await: average time (ms) for i/o requests issued to the device to be serviced (includes the time spent by the requests in queue and the time spent servicing them).

svctm: average service time (ms) for i/o requests that were issued to the device. (time spent outside linux)

top

- The top command shows resource usage on process thread level
- top example (for more see man top)
  - **`top [options] -d [delay] -n [iterations] -p [pid, [pid]]`**
  - *`top -d 1`*
  - `top -b -d 1 -n 180  >top.log 2>&1 &` => batch mode, 3 minutes
  - Customize interactively, “`W`” writes to ~/.toprc (default config)

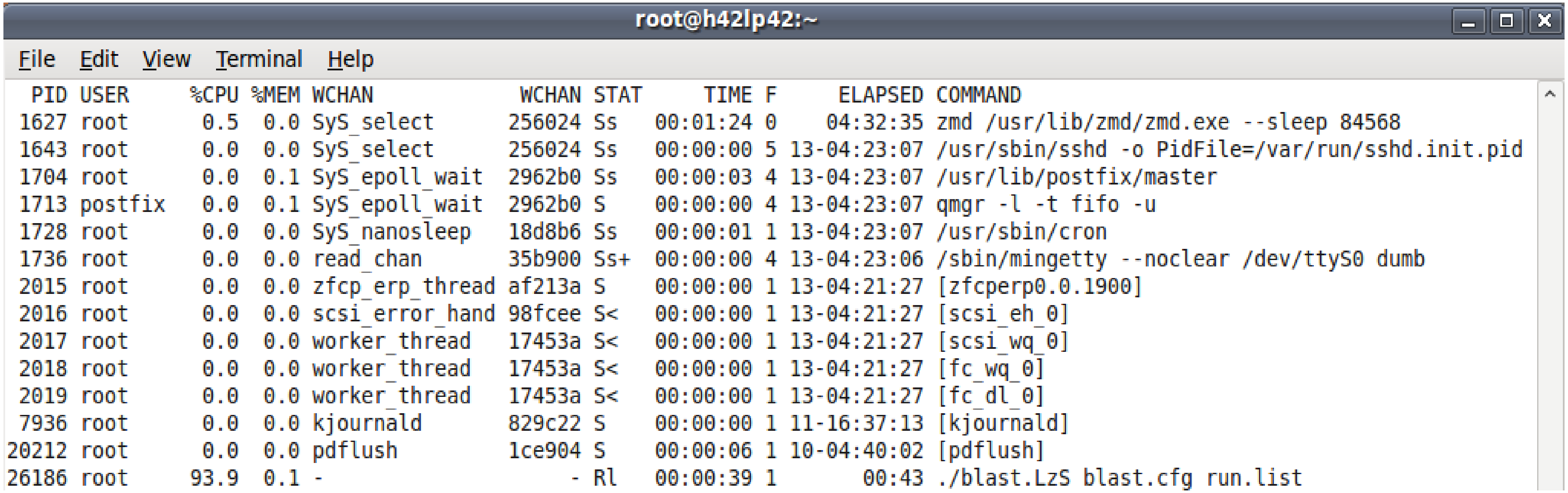


```
top - 17:16:36 up  4:32,  3 users,  load average: 2.93, 2.76, 2.72
Tasks:  70 total,   1 running,  69 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.3%us, 14.8%sy,  0.0%ni, 78.2%id,  5.2%wa,  0.1%hi,  0.2%si,  0.2%st
Mem:   2054900k total,   226584k used,  1828316k free,    37320k buffers
Swap:  2403976k total,    18368k used,  2385608k free,   110672k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 2193 root      16   0 28148 1836  972 S   56  0.1 135:26.27 blast.LzS
    1 root      16   0   848   64   32 S    0  0.0   0:00.68 init
    5 root      34  19     0    0    0 S    0  0.0   0:03.36 ksoftirqd/1
  239 root      15   0     0    0    0 S    0  0.0   0:00.35 kjournald
```

ps

- The ps command reports a snapshot of the current processes
- ps example (for more see man ps)
  - to see every process with a user-defined format
  - *ps -eLo pid,user,%cpu, %mem,wchan:15,nwchan,stat,time,flags,etime,command:50*

```
wchan/stat to search stalls/serialization
Time is accumulated
```



```
  PID USER     %CPU %MEM WCHAN            WCHAN STAT     TIME F     ELAPSED COMMAND
 1627 root      0.5  0.0 SyS_select      256024 Ss   00:01:24 0    04:32:35 zmd /usr/lib/zmd/zmd.exe --sleep 84568
 1643 root      0.0  0.0 SyS_select      256024 Ss   00:00:00 5 13-04:23:07 /usr/sbin/sshd -o PidFile=/var/run/sshd.init.pid
 1704 root      0.0  0.1 SyS_epoll_wait  2962b0 Ss   00:00:03 4 13-04:23:07 /usr/lib/postfix/master
 1713 postfix   0.0  0.1 SyS_epoll_wait  2962b0 S    00:00:00 4 13-04:23:07 qmgr -l -t fifo -u
 1728 root      0.0  0.0 SyS_nanosleep   18d8b6 Ss   00:00:01 1 13-04:23:07 /usr/sbin/cron
 1736 root      0.0  0.0 read_chan       35b900 Ss+  00:00:00 4 13-04:23:06 /sbin/mingetty --noclear /dev/ttyS0 dumb
 2015 root      0.0  0.0 zfcp_erp_thread af213a S    00:00:00 1 13-04:21:27 [zfcperp0.0.1900]
 2016 root      0.0  0.0 scsi_error_hand 98fcee S<   00:00:00 1 13-04:21:27 [scsi_eh_0]
 2017 root      0.0  0.0 worker_thread   17453a S<   00:00:00 1 13-04:21:27 [scsi_wq_0]
 2018 root      0.0  0.0 worker_thread   17453a S<   00:00:00 1 13-04:21:27 [fc_wq_0]
 2019 root      0.0  0.0 worker_thread   17453a S<   00:00:00 1 13-04:21:27 [fc_dl_0]
 7936 root      0.0  0.0 kjournald       829c22 S    00:00:00 1 11-16:37:13 [kjournald]
20212 root      0.0  0.0 pdflush         1ce904 S    00:00:06 1 10-04:40:02 [pdflush]
26186 root     93.9  0.1 -                    - Rl   00:00:39 1       00:43 ./blast.LzS blast.cfg run.list
```

Creating dumps

Linux on System z Dumps - General Principles

- Goal
  - store all CPU states and all of main memory
- Procedure
  - preparation
    - write dump tool as IPL program to dump device (using zipl)
  - dumping
    - stop all CPUs and store CPU state (into some hidden space)
    - IPL dump tool (possibly with special dump option)
    - dump tool saves (while running in main memory) the stored CPU states and original contents of main memory to dump space
    - a Linux is IPLed and used to read dump from dump space (zgetdump)

Linux on System z dump tools

- DASD dump tool:
  - Writes dump directly on DASD partition
  - Uses s390 standalone dump format
  - ECKD and FBA DASDs supported
  - Single volume and multiple volume (for large systems) dump possible
  - Works in z/VM and in LPAR
- SCSI dump tool
  - Writes dump into filesystem
  - Uses lckd dump format
  - Works in z/VM and in LPAR
- VMDUMP:
  - Writes dump to vm spool space (VM reader)
  - z/VM specific dump format, dump must be converted
  - Only available when running under z/VM
- Tape dump tool:
  - Writes dump directly on Escon/Ficon Tape device
  - Uses s390 standalone dump format

DASD dump tool – general usage

- Format and partition dump device

```
root@larsson:~>  dasdfmt -f /dev/dasd<x> -b 4096
root@larsson:~>  fdasd /dev/dasd<x>
```

- Prepare dump device in Linux

```
root@larsson:~>  zipl -d /dev/dasd<x1>
```

- Stop all CPUs
- Store Status
- IPL dump device
- Copy dump to Linux

```
root@larsson:~>  zgetdump /dev/dasd<x1> > dump_file
```

DASD dump under z/VM

- Prepare dump device under Linux:

```
root@larsson:~>  zipl -d /dev/dasd<x1>
```

- After Linux crash issue these commands on 3270 console:

```
#cp cpu all stop
#cp cpu 0 store status
#cp i <dasd_devno>
```

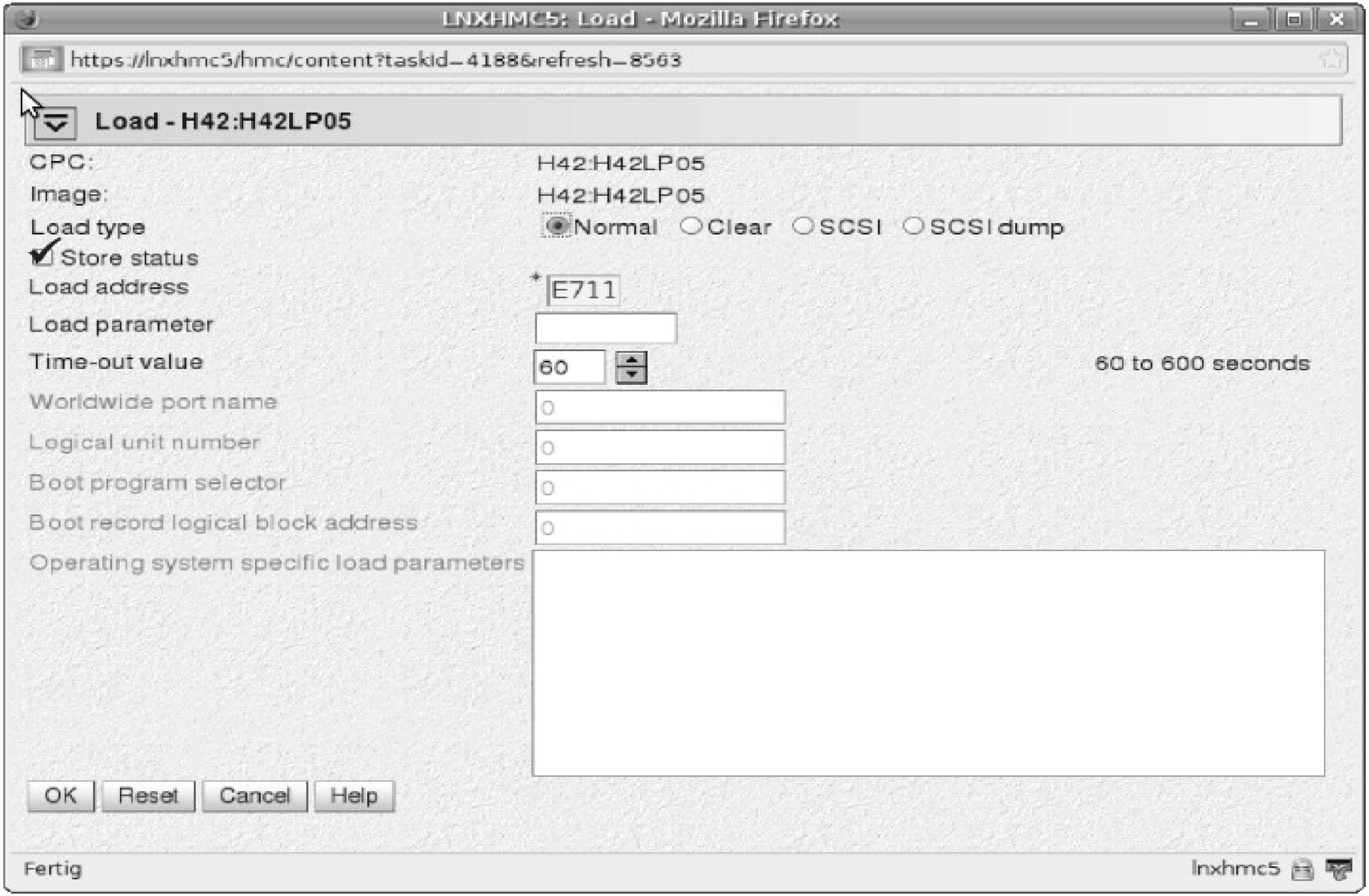
- Wait until dump is saved on device:

```
00: zIPL v1.6.0 dump tool (64 bit)
00: Dumping 64 bit OS
00: 00000087 / 00000700 MB 0
...
00: Dump successful
```

  – Only disabled wait PSW on older Distributions

- Attach dump device to a linux system with dump tools installed
- Store dump to linux file system from dump device (e.g. zgetdump)

DASD dump on LPAR



LNXHMC5: Load - Mozilla Firefox

https://lnxhmc5/hmc/content?taskid=4188&refresh=8563

Load - H42:H42LP05

CPC:	H42:H42LP05
Image:	H42:H42LP05
Load type	● Normal ○ Clear ○ SCSI ○ SCSI dump
✓ Store status	
Load address	* E711
Load parameter	
Time-out value	60 — 60 to 600 seconds
Worldwide port name	0
Logical unit number	0
Boot program selector	0
Boot record logical block address	0
Operating system specific load parameters	

How to obtain information about a dump

- Display information of the involved volume:

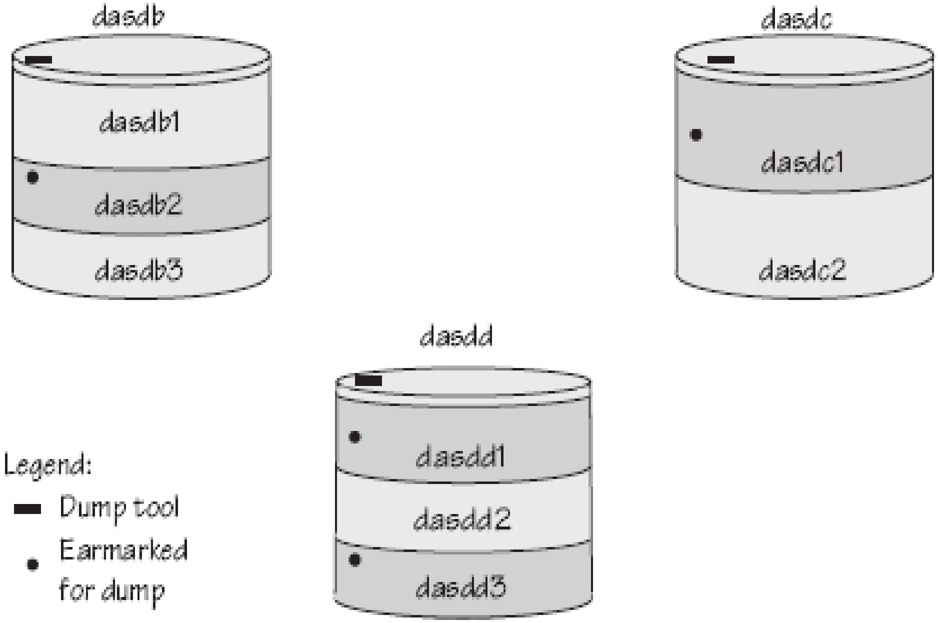
```
root@larsson:~>  zgetdump -d /dev/dasdb
'/dev/dasdb' is Version 0 dump device.
Dump size limit: none
```

- Display information about the dump itself:

```
root@larsson:~>  zgetdump -i /dev/dasdb1
Dump device: /dev/dasdb1

Dump created on: Thu Oct  8 15:44:49 2009

Magic number:        0xa8190173618f23fd
Version number:      3
Header size:         4096
Page size:           4096
Dumped memory:       1073741824
Dumped pages:        262144
Real memory:         1073741824
cpu id:              0xff00012320978000
System Arch:         s390x (ESAME)
Build Arch:          s390x (ESAME)
>>>  End of Dump header  <<<

Dump ended on:       Thu Oct  8 15:45:01 2009
Dump End Marker found: this dump is valid.
```

Multi volume dump

- zipl can now dump to multiple DASDs. It is now possible to dump system images, which are larger than a single DASD.
  - You can specify up to 32 ECKD DASD partitions for a multi-volume dump
- **What are dumps good for?**
  - Full snapshot of system state taken at any point in time (e.g. after a system has crashed, of or a running system)
  - Can be used to analyse system state beyond messages written to the syslog
  - Internal data structures not exported to anywhere



Obtain messages, which have not been written to the syslog due to a crash

Multi volume dump (cont'd)

- How to prepare a set of ECKD DASD devices for a multi-volume dump? (64-bit systems only)
  - We use two DASDs in this example:

```
root@larsson:~>  dasdfmt -f /dev/dasdc -b 4096
root@larsson:~>  dasdfmt -f /dev/dasdd -b 4096
```

  - Create the partitions with fdasd. The sum of the partition sizes must be sufficiently large (the memory size + 10 MB):

```
root@larsson:~>  fdasd /dev/dasdc
root@larsson:~>  fdasd /dev/dasdd
```

  - Create a file called sample_dump_conf containing the device nodes (e.g. /dev/dasdc1) of the two partitions, separated by one or more line feed characters
  - Prepare the volumes using the zipl command.

```
root@larsson:~>  zipl -M sample_dump_conf
[...]
```

Multi volume dump (cont'd)

- To obtain a dump with the multi-volume DASD dump tool, perform the following steps:
  - Stop all CPUs, Store status on the IPL CPU.
  - IPL the dump tool using one of the prepared volumes, either 4711 or 4712.
  - After the dump tool is IPLed, you'll see a messages that indicates the progress of the dump. Then you can IPL Linux again

```
#cp cpu all stop
#cp cpu 0 store status
#cp ipl 4711
```

- Copying a multi-volume dump to a file
  - Use zgetdump without any option to copy the dump parts to a file:

```
root@larsson:~>  zgetdump /dev/dasdc > mv_dump_file
```

Multi volume dump (cont'd)

- Display information of the involved volumes:

```
root@larsson:~>  zgetdump -d /dev/dasdc
'/dev/dasdc' is part of Version 1 multi-volume dump,which is
spread along the following DASD volumes:
0.0.4711 (online, valid)
0.0.4712 (online, valid)
[...]
```

- Display information about the dump itself:

```
root@larsson:~>  zgetdump -i /dev/dasdc
Dump device: /dev/dasdc
>>>  Dump header information  <<<
Dump created on: Fri Aug  7 15:12:41 2009  [...]
Multi-volume dump: Disk 1 (of 2)
Reading dump contents from
0.0.4711.................................
Dump ended on:   Fri Aug  7 15:12:52 2009
Dump End Marker found: this dump is valid.
```

SCSI dump tool – general usage

- Create partition with PCBIOS disk-layout (fdisk)
- Format partition with ext2 or ext3 filesystem
- Install dump tool:
  - mount and prepare disk :

```
root@larsson:~>  mount /dev/sda1 /dumps
root@larsson:~>  zipl -D /dev/sda1 -t dumps
```

  - Optional: /etc/zipl.conf:

```
[scsidump]
dumptofs=/dev/sda1
target=/dumps
```

- Stop all CPUs
- Store Status
- IPL dump device

Dump tools creates dumps directly in filesystem

SCSI dump supported for LPARs and as of z/VM 5.4

SCSI dump under z/VM

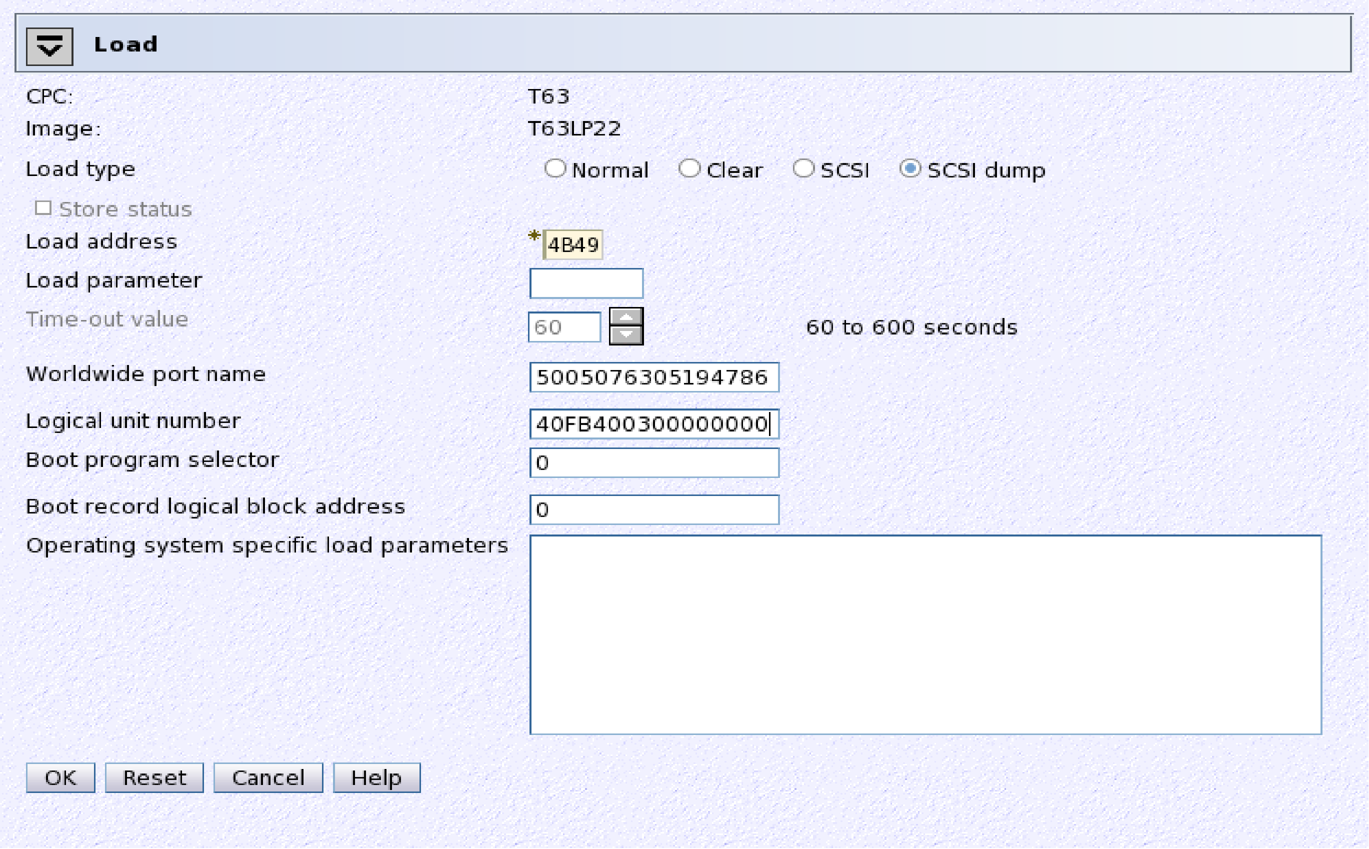
- SCSI dump from z/VM is supported as of z/VM 5.4
- Issue SCSI dump

```
#cp cpu all stop
#cp cpu 0 store status
#cp set dumpdev portname 47120763 00ce93a7 lun 47120000
00000000 bootprog 0
#cp ipl 4b49 dump
```

- To access the dump, mount the dump partition

SCSI dump on LPAR

- Select CPC image for LPAR to dump
- Goto Load panel
- Issue SCSI dump
  - FCP device
  - WWPN
  - LUN



Load

CPC: T63

Image: T63LP22

Load type ○ Normal ○ Clear ○ SCSI ● SCSI dump

□ Store status

Load address * 4B49

Load parameter

Time-out value 60 60 to 600 seconds

Worldwide port name 5005076305194786

Logical unit number 40FB400300000000

Boot program selector 0

Boot record logical block address 0

Operating system specific load parameters

Get dump and send it to service organization

- DASD/Tape:
  - Store dump to Linux file system from dump device:

```
root@larsson:~>  zgetdump /dev/<device node> > dump_file
```

- SCSI:
  - Get dump from filesystem
- Additional files needed for dump analysis:
  - SUSE (lcrash tool): */boot/System.map-xxx* and */boot/Kerntypes-xxx*
  - Redhat & SUSE (crash tool): vmlinux file (kernel with debug info) contained in debug kernel rpms:
    - RedHat: kernel-debuginfo-xxx.rpm and kernel-debuginfo-common-xxx.rpm
    - SUSE: kernel-default-debuginfo-xxx.rpm

Handling Large Dumps

- Dumps of large images are large
  - e.g. an image of 0.5 TB leads to a dump of approx. 0.5TB
  - transferring large dumps may be a problem

- Solutions
  - compress & split the dump
    - no dump data gets lost
    - SCSI dump tool has a compress option (dump_compress=gzip)
  - filter the dump
    - only dump data relevant to kernel operation is preserved

Compressing and Splitting Large Dumps

- Compress the dump and split it into parts of 1 GB

```
root@larsson:~>  zgetdump /dev/dasdc1 | gzip | split -b 1G
```

- Several compressed files such as xaa, xab, xac, ... are created
- Create md5 sums of the compressed files

```
root@larsson:~>  md5sum xa* > dump.md5
```

- Upload all parts together with the md5 information

- Verification of the parts for a receiver

```
root@larsson:~>  md5sum -c dump.md5
xaa: OK
[....]
```

- Merge the parts and uncompress the dump

```
root@larsson:~>  cat xa* | gunzip -c > dump
```

Transferring dumps

- Transferring single volume dumps with ssh

```
root@larsson:~>  zgetdump /dev/dasdc1 | ssh user@host "cat >
dump_file_on_target_host"
```

- Transferring multi-volume dumps with ssh

```
root@larsson:~>  zgetdump /dev/dasdc | ssh user@host "cat >
multi_volume_dump_file_on_target_host"
```

- Transferring a dump with ftp
  - Establish an ftp session with the target host, login and set the transfer mode to binary

```
root@larsson:~>  ftp> put |"zgetdump /dev/dasdc1"
<dump_file_on_target_host>
```

Makedumpfile tool

- Can be used to compress s390 dumps and exclude memory pages that are not needed for analysis e.g. user space pages, (file) cache pages, free pages, zero pages
- Expects as input dumps in the ELF format
- Transform your s390-format dump into ELF format by mounting the dump from partition
  - create virtual elf dump in /mnt/dump.elf from dump partition /dev/dasdb1

```
root@larsson:~>  zgetdump -m -f elf /dev/dasdb1 /mnt
```

  - or from SCSI dump file dump.0

```
root@larsson:~>  zgetdump -m -f elf dump.0 /mnt
```

- Now the dump is available in the file /mnt/dump.elf

Makedumpfile tool (cont'd)

- In order to use the makedumpfile you need the vmlinux file that contains necessary debug information
- Extract the vmlinux debug file from the kernel rpm for your kernel version xyz
  - SLES 11 SP2

```
root@larsson:~>  rpm -qlp kernel-default-debuginfo-xyz.rpm | grep vmlinux
/usr/lib/debug/boot/vmlinux-xyz-default.debug
root@larsson:~>  rpm2cpio kernel-default-debuginfo-xyz.rpm | cpio -idv
*vmlinux*
./usr/lib/debug/boot/vmlinux-xyz-default.debug
1224646 blocks
```

  - RHEL 6

```
root@larsson:~>  rpm -qlp kernel-debuginfo-xyz.rpm | grep vmlinux
/usr/lib/debug/lib/modules/2.6.32-131.0.15.el6.s390x/vmlinux
root@larsson:~>  rpm2cpio kernel-debuginfo-xyz.rpm | cpio -idv *vmlinux*
./usr/lib/debug/lib/modules/2.6.32-131.0.15.el6.s390x/vmlinux
1082264 blocks
```

Makedumpfile tool (cont'd)

- Use the makedumpfile tool to exclude pages and compress the dump
  - Use **-d <dump_level>** to indicate which pages are excluded
  - Use **-c** to compress the dump

```
root@larsson:~>  makedumpfile -c -d 31 -x
usr/lib/debug/lib/modules/2.6.32-131.0.15.el6.s390x/vmlinux
/mnt/dump.elf dump.kdump
Copying data                               : [100 %]

The dumpfile is saved to dump.kdump.
makedumpfile Completed.
```

- For initial problem analysis, extract kernel log

```
root@larsson:~>  makedumpfile --dump-dmesg -x
usr/lib/debug/lib/modules/2.6.32-131.0.15.el6.s390x/vmlinux
/mnt/dump.elf kernel.log

The dmesg log is saved to kernel.log.
makedumpfile Completed.
```

- unmount elf dump

```
root@larsson:~>  zgetdump -u /mnt
```

Customer Cases

Network connection is too slow

- Configuration:
  - z/VSE running CICS, connecting to DB2 in zLinux
  - HiperSocket connection from zLinux to z/VSE
  - But also applies to hipersocket connections between zLinux and z/OS
- Problem Description:
  - When CICS transaction were monitored, some transactions take a couple of seconds instead of milliseconds
- Tools used for problem determination:
  - dbginfo.sh
  - s390 debug feature
  - sadc/sar
  - CICS transaction monitor

Network connection is too slow (cont'd)

- s390 debug feature
  - Check for qeth errors:

```
cat /sys/kernel/debug/s390dbf/qeth_qerr
00 01282632346:099575 2 - 00 0000000180b20218  71 6f 75 74 65 72 72 00 | qouterr.
00 01282632346:099575 2 - 00 0000000180b20298  20 46 31 35 3d 31 30 00 |  F15=10.
00 01282632346:099576 2 - 00 0000000180b20318  20 46 31 34 3d 30 30 00 |  F14=00.
00 01282632346:099576 2 - 00 0000000180b20390  20 71 65 72 72 3d 41 46 |  qerr=AF
00 01282632346:099576 2 - 00 0000000180b20408  20 73 65 72 72 3d 32 00 |  serr=2.
```

- dbginfo file
  - Check for buffer count:

```
cat /sys/devices/qeth/0.0.1e00/buffer_count
16
```

- Problem Origin:
  - Too less inbound buffers

Network connection is too slow (cont'd)

- Solution:
  - Increase inbound buffer count (default: 16, max 128)
  - Check actual buffer count with `'lsqeth -p'`
  - Set the inbound buffer count in the appropriate config file:
    - SUSE SLES10:

      in /etc/sysconfig/hardware/hwcfg-qeth-bus-ccw-0.0.F200

      add `QETH_OPTIONS="buffer_count=128"`
    - SUSE SLES11:

      in /etc/udev/rules.d/51-qeth-0.0.f200.rules add ACTION=="add", `SUBSYSTEM=="ccwgroup", KERNEL=="0.0.f200", ATTR{buffer_count}="128"`
    - Red Hat:

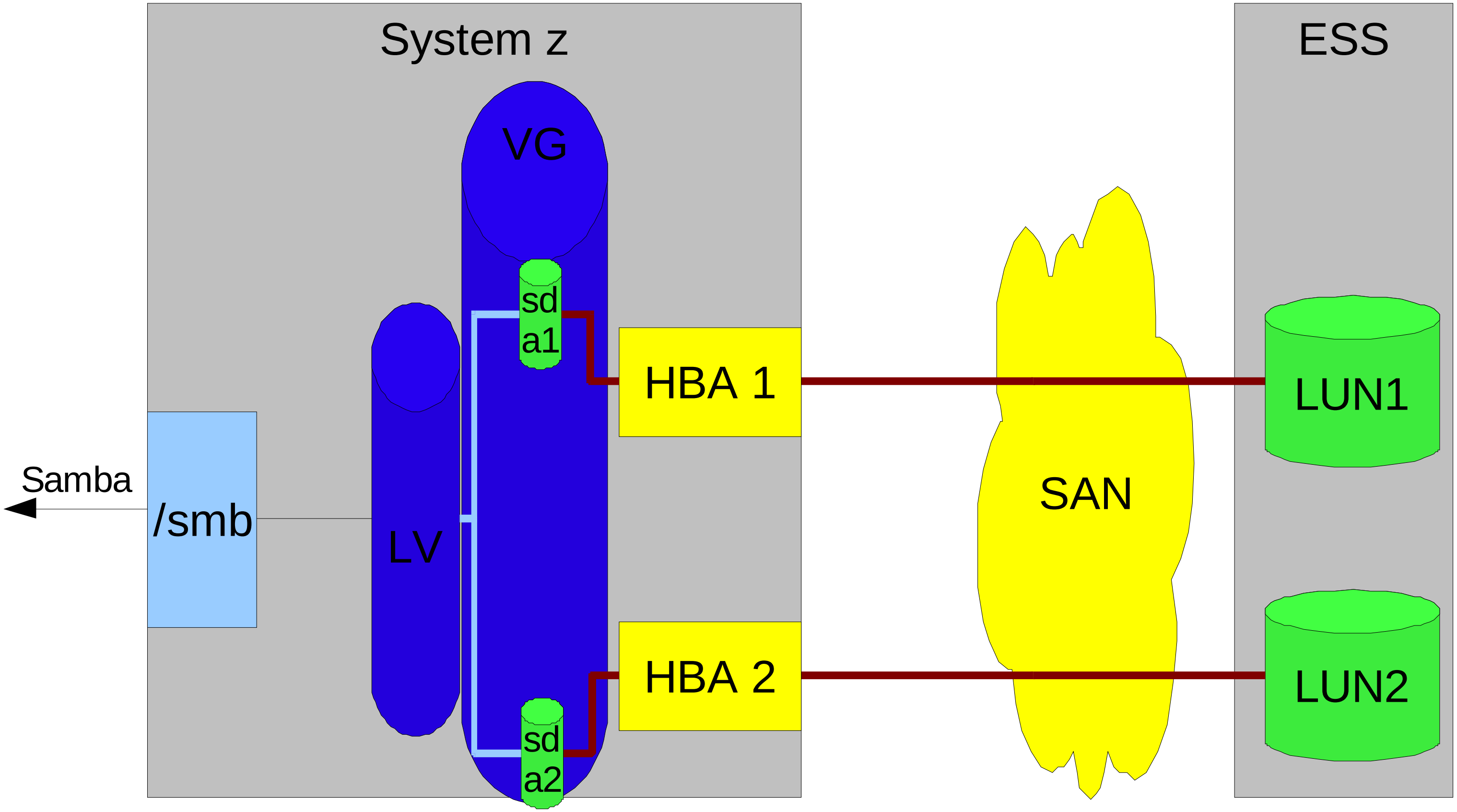
      in /etc/sysconfig/network-scripts/ifcfg-eth0

      add OPTIONS=`"buffer_count=128"`

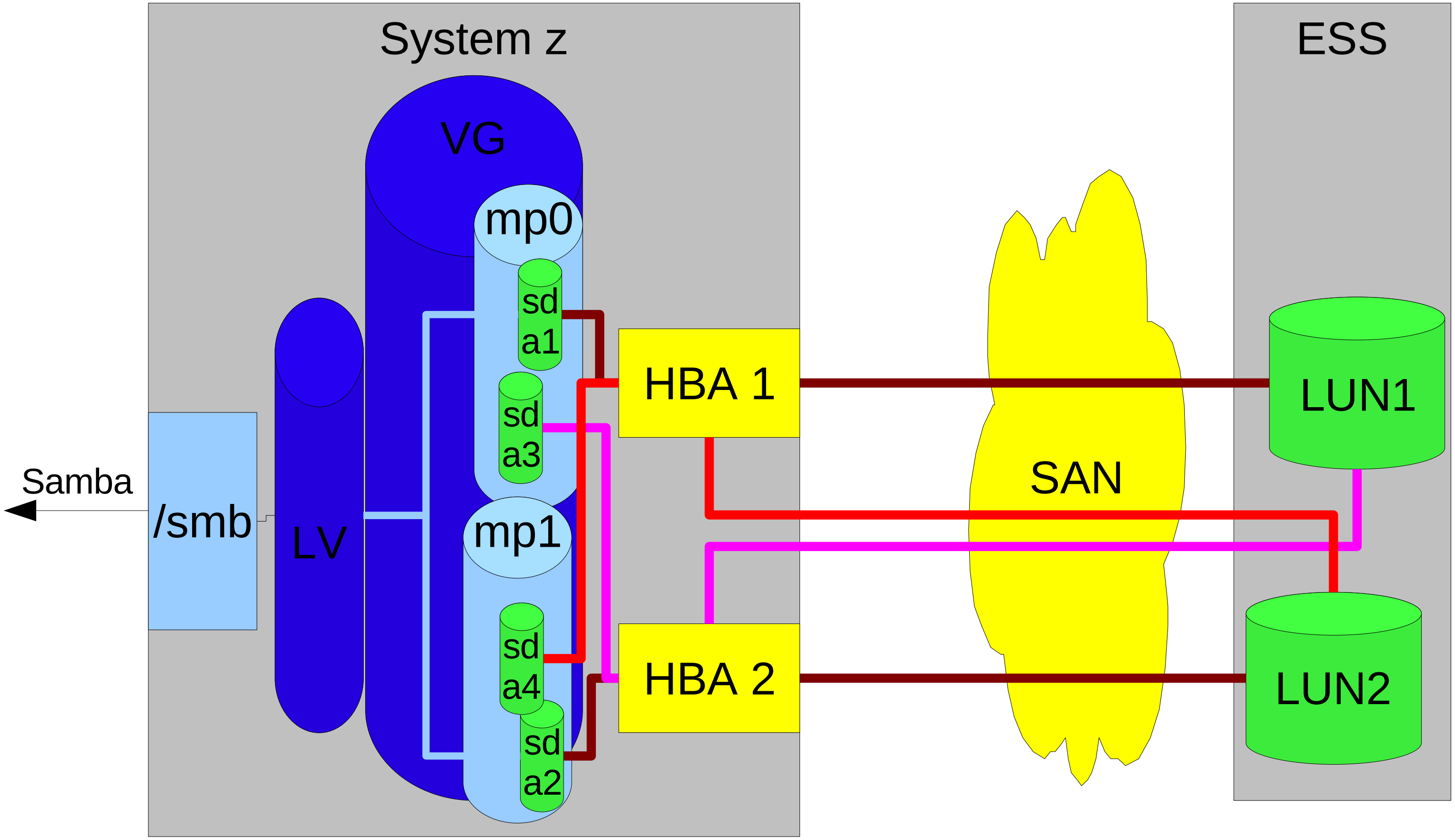
FCP disk: multipath configuration

- Configuration:
  - Customer is running Samba server on Linux with FCP attached disk managed by Linux LVM.
  - This problem also applies to any configuration with FCP attached disk storage
- Problem Description:
  - Accessing some files through samba causes the system to hang while accessing other files works fine
  - Local access to the same file cause a hanging shell as well
    - Indicates: this is not a network problem!
- Tools used for problem determination:
  - dbginfo.sh
- Problem Indicators:
  - Intermittent outages of disk connectivity

FCP disk: multipath configuration (cont'd)



FCP disk: multipath configuration (cont'd)



FCP disk: multipath configuration (cont'd)

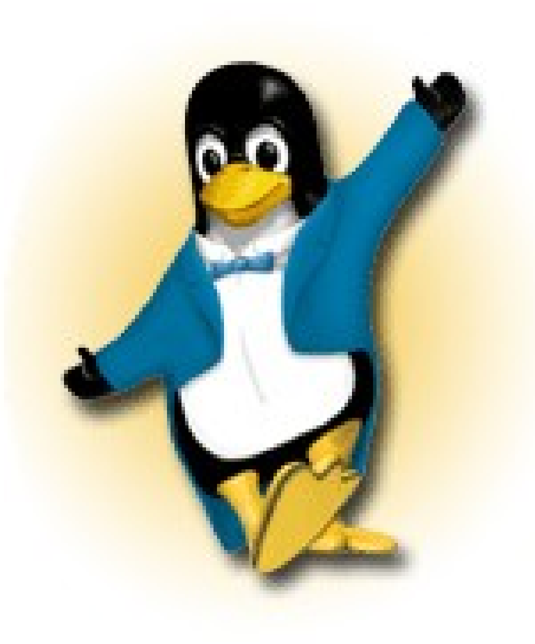
- Solutions:
  - Configure multipathing correctly:
    - Establish independent paths to each volume
    - Group the paths using the device-mapper-multipath package
    - Base LVM configuration on top of mpath devices instead of sd<#>
  - For a more detailed description how to use FCP attached storage appropriately with Linux on System z, see

http://public.dhe.ibm.com/software/dw/linux390/docu/lk33ts04.pdf

References

- Linux on System z project at IBM DeveloperWorks:
  http://www.ibm.com/developerworks/linux/linux390/
- Linux on System z: Tuning Hints & Tips
  http://www.ibm.com/developerworks/linux/linux390/perf
- Optimize disk configuration for performance:
  http://www.ibm.com/developerworks/linux/linux390/perf/tuning_rec_dasd_optimize
- Linux-VM Performance Website:
  http://www.vm.ibm.com/perf/tips/linuxper.html
- IBM Redbooks:
  http://www.redbooks.ibm.com/
- IBM Techdocs:
  http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs

Questions?



Susanne Wintenberger

Certified IT Specialist

Linux on System z

Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-3514
swinten@de.ibm.com