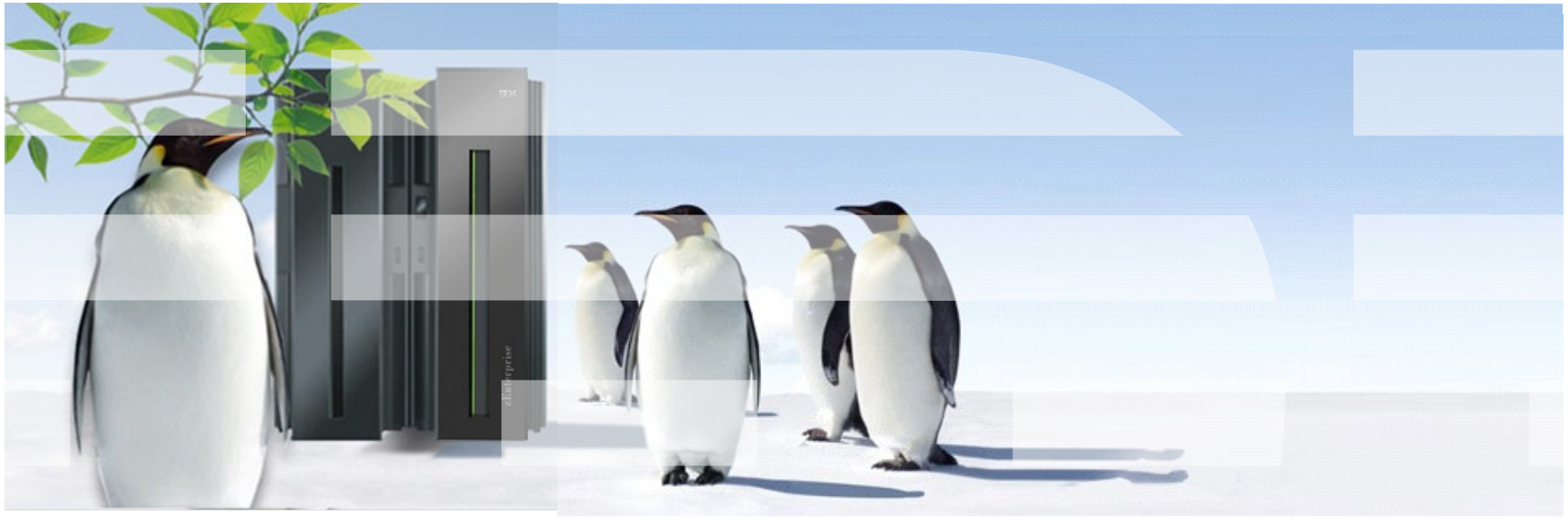
Linux on System z – Problem Determination

Sven Schuetz
Linux on System z Development and Service
sven@de.ibm.com



Agenda

- Introduction
- How to help us to help you
- Systems monitoring
- How to dump a Linux on System z
- Real Customer cases?

Introductory remarks

- **Problem analysis looks straight forward on the charts but it might have taken weeks to get it done.**
  A problem does not necessarily show up on the place of origin
- **The more information is available, the sooner the problem can be solved, because gathering and submitting additional information again and again usually introduces delays.**
- **This presentation can only introduce some tools and how the tools can be used, comprehensive documentation on their capabilities is to be found in the documentation of the corresponding tool.**
- **Do not forget to update your systems**

Describe the problem

- **Get as much information as possible about the circumstances:**
  - What is the problem?
  - When did it happen? (date and time, important to dig into logs)
  - Where did it happen? One or more systems, production or test environment?
  - Is this a first time occurrence?
  - If occurred before: how frequently does it occur?
  - Is there any pattern?
  - Was anything changed recently?
  - Is the problem reproducible?

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

Describe the environment

- **Machine Setup**
  - Machine type (z196, z10, z9, ...)
  - Storage Server (ESS800, DS8000, other vendors models)
  - Storage attachment (FICON, ESCON, FCP, how many channels)
  - Network (OSA (type, mode), Hipersocket) ...
- **Infrastructure setup**
  - Clients
  - Other Computer Systems
  - Network topologies
  - Disk configuration
- **Middleware setup**
  - Databases, web servers, SAP, TSM, (including version information)

Trouble Shooting First-Aid Kit (1/2)

- **Install packages required for debugging**
  - s390-tools/s390-utils
    - dbginfo.sh
  - sysstat
    - sadc/sar
    - iostat
  - procps
    - vmstat, top, ps
  - net-tools
    - netstat
  - dump tools crash / lcrash
    - lcrash (lkcdutils) available with SLES9 and SLES10
    - crash available on SLES11
    - crash in all RHEL distributions

Trouble Shooting First-Aid Kit (2/2)

- **Collect dbginfo.sh output**
  - Proactively in healthy system
  - When problems occur – then compare with healthy system
- **Collect system data**
  - Always archive syslog (/var/log/messages)
  - Start sadc (System Activity Data Collection) service when appropriate (please include disk statistics)
  - Collect z/VM MONWRITE Data if running under z/VM when appropriate
- **When System hangs**
  - Take a 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*
- **Enable extended tracing in /sys/kernel/debug/s390dbf for subsystem**

dbginfo Script (1/2)

- **dbginfo.sh is a script to collect various system related files, for debugging purposes. It generates a tar-archive which can be attached to PMRs / Bugzilla entries**
- **part of the s390-tools package in SUSE and recent Red Hat distributions**
  - dbginfo.sh gets continuously improved by service and development

  Can be downloaded at the developerWorks website directly

  http://www.ibm.com/developerworks/linux/linux390/s390-tools.html
- **It is similar to the RedHat tool sosreport or supportconfig from Novell**

```
root@larsson:~>  dbginfo.sh
Create target directory /tmp/DBGINFO-2011-01-15-22-06-
20-t6345057
Change to target directory /tmp/DBGINFO-2011-01-15-22-
06-20-t6345057
[...]
```

dbginfo Script (2/2)

- **Linux Information:**
  - /proc/[version, cpu, meminfo, slabinfo, modules, partitions, devices ...]
  - System z specific device driver information: /proc/s390dbf (RHEL 4 only) or /sys/kernel/debug/s390dbf
  - Kernel messages /var/log/messages
  - Reads configuration files in directory /etc/ [ccwgroup.conf, modules.conf, fstab]
  - Uses several commands: ps, dmesg
  - Query setup scripts
    - lscss, lsdasd, lsqeth, lszfcp, lstape
  - And much more
- **z/VM information:**
  - 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...]`
  - When the system runs as z/VM guest, ensure that the guest has the appropriate privilege class authorities to issue the commands

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
  - … and much more
  - Reports statistics data over time and creates average values for each item
- **SADC example (for more see man sadc)**
  - **S**ystem **A**ctivity **D**ata **C**ollector (sadc) --> data gatherer
  - /usr/lib64/sa/sadc [options] [interval [count]] **[binary outfile]**
  - /usr/lib64/sa/sadc 10 20 sadc_outfile

SADC/SAR (cont'd)

- /usr/lib64/sa/sadc **-d** 10 sadc_outfile
- -d option: statistics for disk
- Should be started as a service during system start

* SAR example (for more see man sar)
  - **S**ystem **A**ctivity **R**eport (sar) command --> reporting tool
  - sar **-A**
  - -A option: reports all the collected statistics
  - sar -A -f sadc_outfile >sar_outfile

- **Please include the binary sadc data and sar -A output when submitting SADC/SAR information to IBM support**

CPU utilization

Per CPU values:
watch out for
- system time (kernel time)
- iowait time (slow I/O subsystem)
- steal time (time taken by other guests)

File Edit View Terminal Help

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 rates

root@h42lp42

File Edit View Terminal Help

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

read/write operations
- per I/O device
- tps: transactions
- rd/wr_secs: sectors
is your I/O balanced?
Maybe you should stripe your LVs

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

1. **Format and partition dump device**

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

```
root@larsson:~>  fdasd /dev/dasd<x>
```

2. **Prepare dump device in Linux**

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

3. **Stop all CPUs**
4. **Store Status**
5. **IPL dump device**
6. **Copy dump to Linux**

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

DASD dump under z/VM

- **Prepare dump device under Linux, if possible on 64Bit environment:**

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

- **After Linux crash issue these commands on 3270 console:**

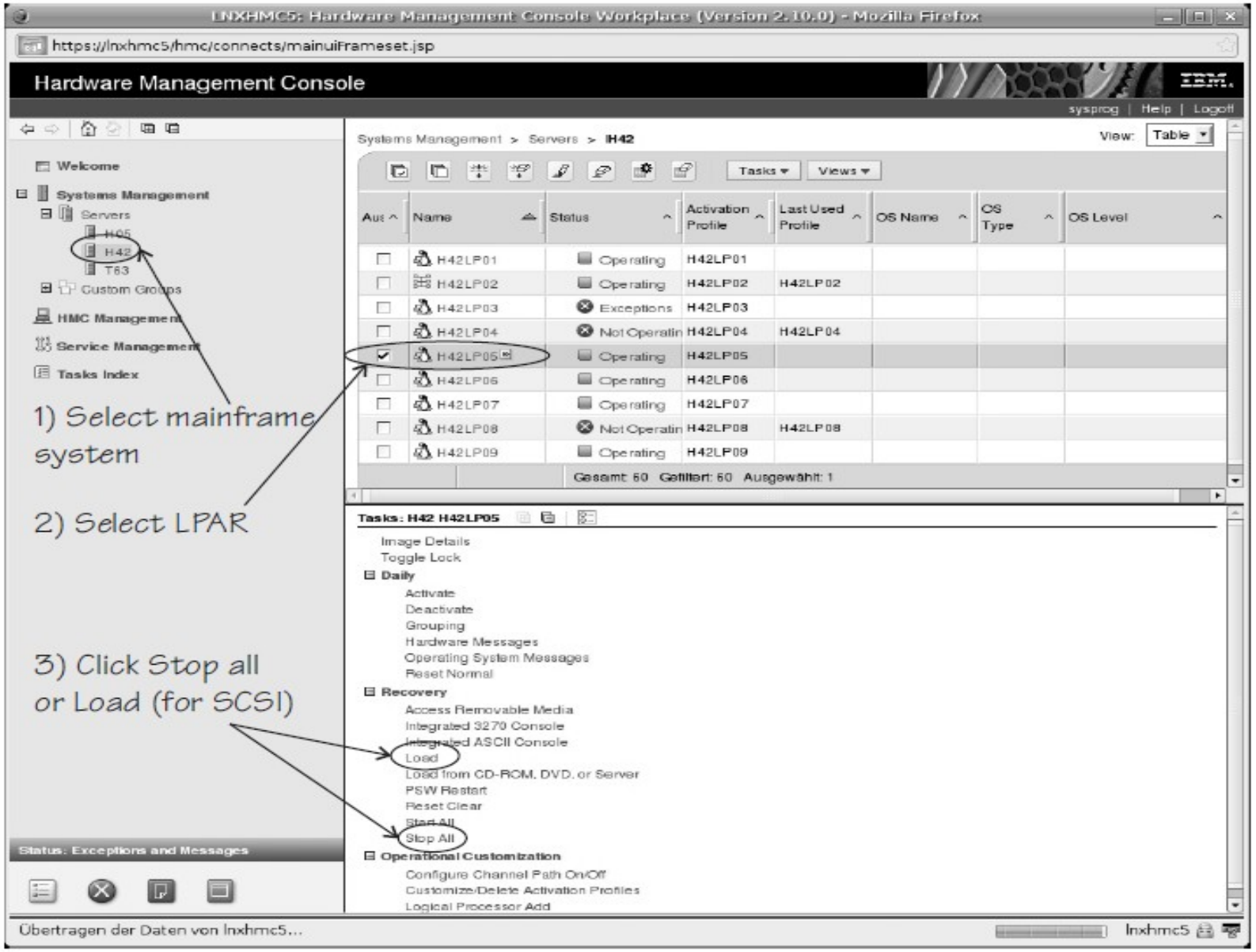
```
#cp cpu all stop
#cp store status
#cp i <dasd_devno>
```

- **Wait until dump is saved on device:**

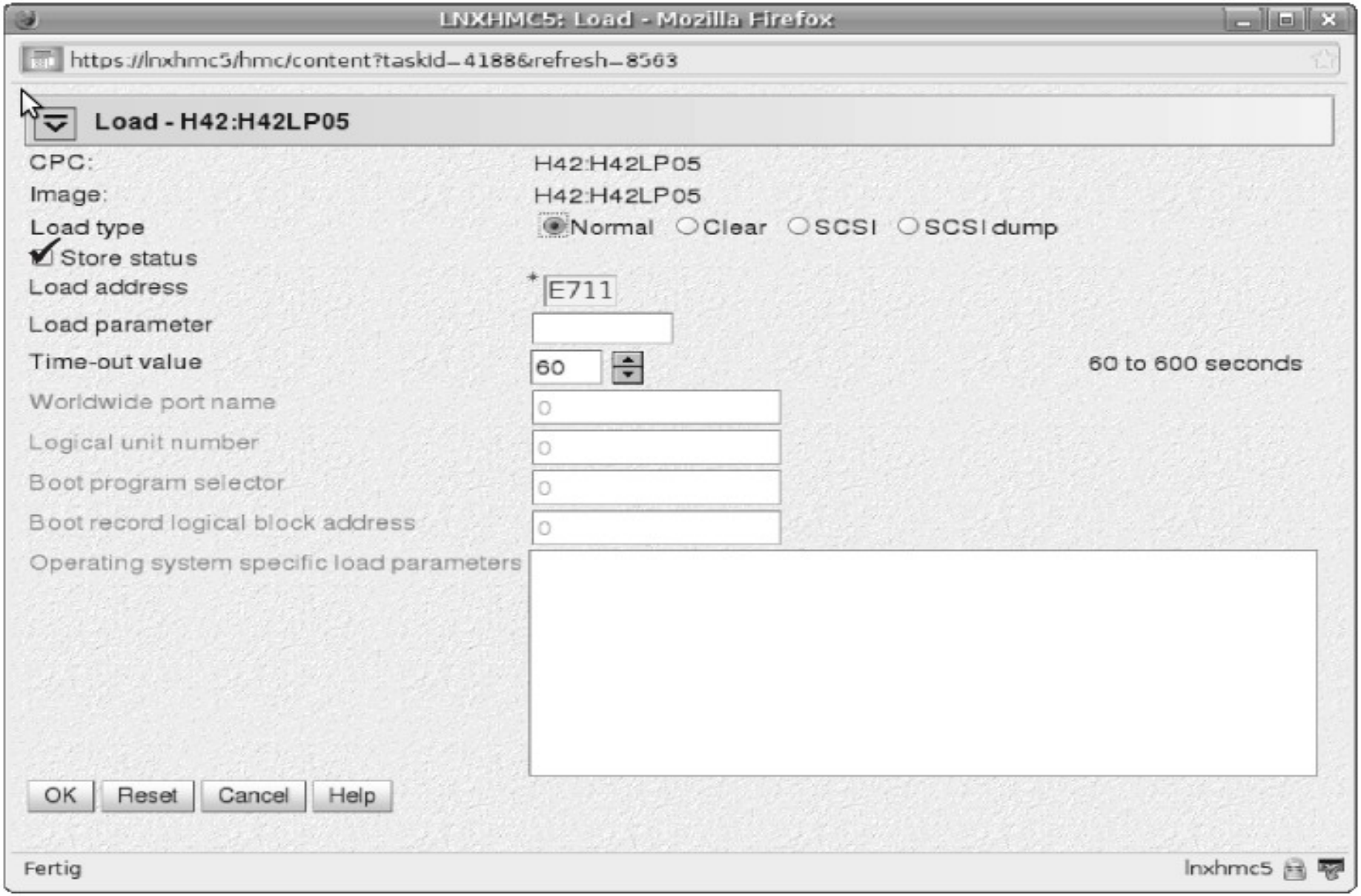
```
00: zIPL v1.6.0 dump tool (64 bit)
00: Dumping 64 bit OS
00: 00000087 / 00000700 MB 0
...
00: Dump successful
```

- **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 (1/2)



DASD dump on LPAR (2/2)



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 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

1. **Create partition with PCBIOS disk-layout (fdisk)**
2. **Format partition with ext2 or ext3 filesystem**
3. **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:*

```
dumptofs=/dev/sda1
target=/dumps
```

4. **Stop all CPUs**
5. **Store Status**
6. **IPL dump device**

Dump tool creates dumps directly in filesystem

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

SCSI dump under z/VM

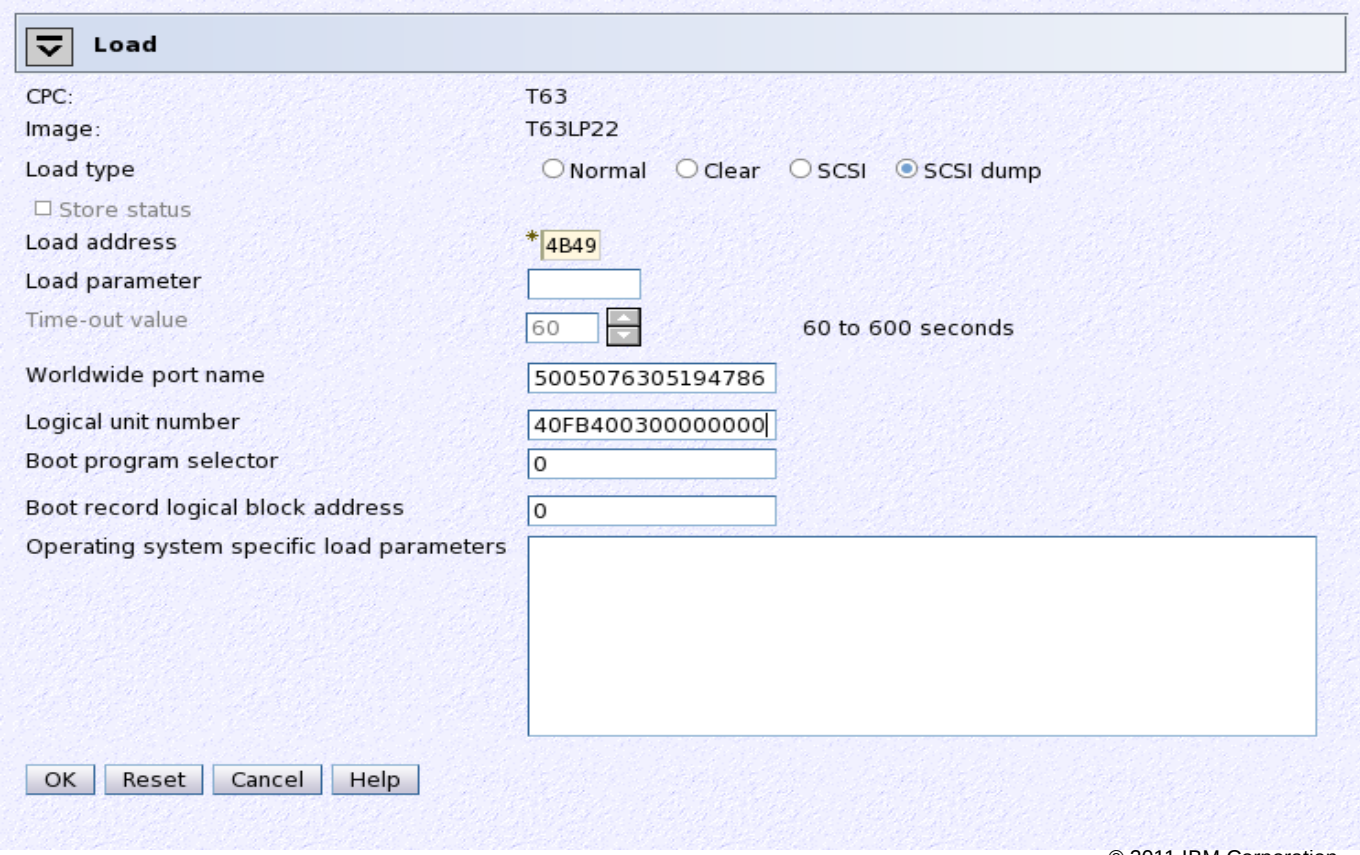
- **SCSI dump from z/VM is supported as of z/VM 5.4**
- **Issue SCSI dump**

```
#cp 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



VMDUMP

- **The only method to dump NSSes or DCSSes under z/VM**
- **Works nondisruptive**
- **Create dump:**

```
#cp vmdump to cmsguest
```

- **Receive dump:**
  - Store the dump from the reader into CMS dump file:

```
#cp dumpload
```

  - Transfer the dump to linux from CMS e.g. FTP
  - NEW: vmur device driver:

```
root@larsson:~>  vmur rec <spoolid> vmdump
```

- **Linux tool to convert vmdump to lkcd format:**

```
root@larsson:~>  vmconvert vmdump linux.dump
```

- **Problem: Dump process relatively slow**

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.
```

How to obtain information about a dump (cont'd)

- **Display information about the dump itself (dump header) and check if the dump is valid, use lcrash with options**
- **'-i' and '-d'.**

```
root@larsson:~>  lcrash -i -d /dev/dasdb1
       Dump Type: s390 standalone dump
          Machine: s390x (ESAME)
           CPU ID: 0xff00012320978000

     Memory Start: 0x0
       Memory End: 0x40000000
      Memory Size: 1073741824

     Time of dump: Thu Oct  8 15:44:49 2009
  Number of pages: 262144
 Kernel page size: 4096
   Version number: 3
     Magic number: 0xa8190173618f23fd
 Dump header size: 4096
       Dump level: 0x4
       Build arch: s390x (ESAME)
 Time of dump end: Thu Oct  8 15:45:01 2009

End Marker found! Dump is valid!
```

Automatic dump on panic (SLES 10/11, RHEL 5/6): dumpconf

- **The dumpconf tool configures a dump device that is used for automatic dump in case of a kernel panic.**
  - The command can be installed as service script under */etc/init.d/dumpconf* or can be called manually.
  - Start service: `# service dumpconf start`
  - It reads the configuration file */etc/sysconfig/dumpconf*.
  - Example configuration for CCW dump device (DASD) and reipl after dump:

```
ON_PANIC=dump_reipl
DUMP_TYPE=ccw
DEVICE=0.0.4711
```

Automatic dump on panic (SLES 10/11, RHEL 5): dumpconf (cont'd)

- Example configuration for FCP dump device (SCSI disk):

```
ON_PANIC=dump
DUMP_TYPE=fcp
DEVICE=0.0.4714
WWPN=0x5005076303004712
LUN=0x4047401300000000
BOOTPROG=0
BR_LBA=0
```

- Example configuration for re-IPL without taking a dump, if a kernel panic occurs:

```
ON_PANIC=reipl
```

- Example of executing a CP command, and rebooting from device 4711 if a kernel panic occurs:

```
ON_PANIC=vmcmd
VMCMD_1="MSG <vmguest> Starting VMDUMP"
VMCMD_2="VMDUMP"
VMCMD_3="IPL 4711"
```

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
```

  - Alternative: lcrash (Compression possible)

```
root@larsson:~>  lcrash -d /dev/dasdxx -s <dir>
```

- **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

- **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
  - Send the dump to the host

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

Dump tool summary

Tool	Stand alone tools			VMDUMP
	DASD	Tape	SCSI	
Environment	VM&LPAR		VM&LPAR	VM
Preparation	zipl -d /dev/<dump_dev>		mkdir /dumps/mydumps zipl -D /dev/sda1 …	---
Creation	Stop CPU & Store status ipl <dump_dev_CUU>			vmdump
Dump medium	ECKD or FBA	Tape cartridges	LINUX file system on a SCSI disk	VM reader
Copy to filesystem	zgetdump /dev/<dump_dev> > dump_file		---	Dumpload ftp … vmconvert …
Viewing	lcrash or crash			

See “Using the dump tools” book on
http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html

Additional Offerings

- **Standard supportline contract might not be enough for every customer**
- **Additional offerings available**
  - Health Check
    - Proactive check of system configuration
    - Recommendations / report to optimize configuration
  - Real Time Disk Mirror solution
    - Enhanced Raid1 implementation to customers with special needs for data availability
  - Premium Service Contract
  - Anything else …
- **Priced features**
- **Working together with GTS / Lab Services**