Linux Bootloaders on System z

Current and Future Implementations

Session 16432

Marcus Kraft
Product Manager
mkraft@suse,com

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

SHARE in Seattle 2015

Copyright (c) 2014 by SHARE Inc. Except where otherwise noted, this work is licensed under http://creativecommons.org/licenses/by-nc-sa/3.0/

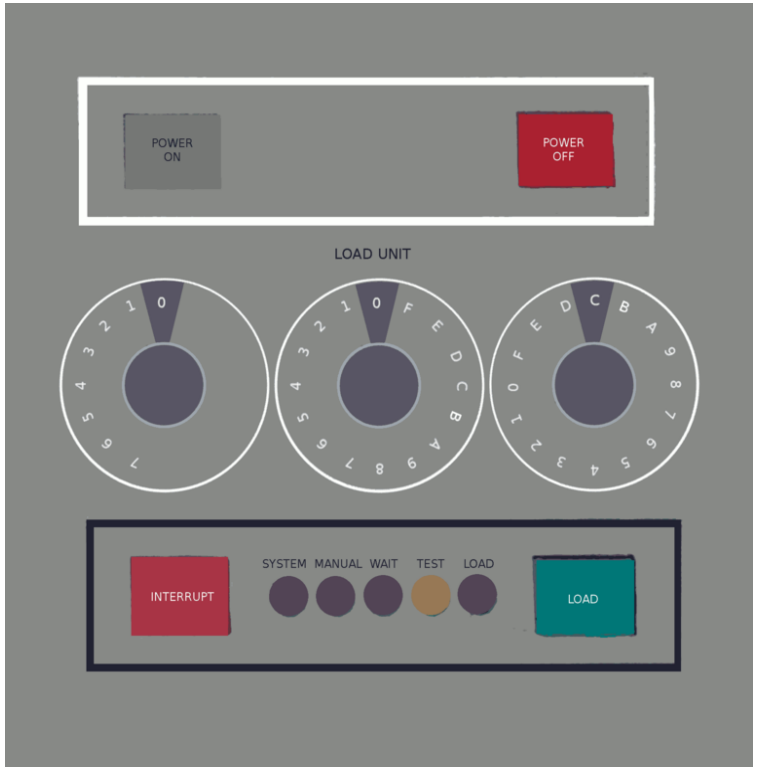
Agenda

- Boot Process on System Z
  - Initial Program Load (IPL)
  - zIPL and limitations
  - Grub 2 function
  - Grub 2 s390 implementation
- Btrfs Intergration
- Demo

IPL Process

Initial Program Load (IPL)

- Formalized with System 360
  - Set the dials, push the button
- Initiated from CP console for z/VM or via HMC
- 1 PSW and 2 CCW loaded from
  - That's enough to load a Command Program



Linux Initial Program Load (zIPL)

- Linux bootloader for System z → zipl
- Configured with /etc/zipl.conf
- Bootloader is written to disk using /sbin/zipl

zIPL Configuration file
/etc/zipl.conf

```
[defaultboot]
defaultmenu = menu


[SLES11_SP3]
    image = /boot/image-3.0.76-0.11-default
    target = /boot/zipl
    ramdisk = /boot/initrd-3.0.76-0.11-default,0x2000000
    parameters = "root=/dev/dasda2 TERM=dumb resume=/dev/dasda1"


:menu
    default = 1
    prompt = 1
    target = /boot/zipl
    timeout = 10
    1 = SLES11_SP3
```

zIPL Operation

- Multi-stage operation
  - Stage 0 – 24 bytes – 2 CCWs to load and TIC to Stage1
  - Stage 1 – 104 byte channel program to load Stage 1b
  - Stage 1b – 1k channel program to load Stage 2
  - Stage 2 – 8k maximum menu and kernel loader
  - Stage 3 – Kernel parameter parser and execution

ZIPL Limitations - Devices

- Limited device capabilities
  - Restricted to specific Channel I/O device types
    - DASD
    - SCSI
    - Tape
  - Boot from the Network? Nope.
  - Device-mapper tends to be finicky

zIPL Limitations – File Systems

- Limited file system support
  - Only ext2, ext3
  - Btrfs and XFS not supported

Grub 2 on System z

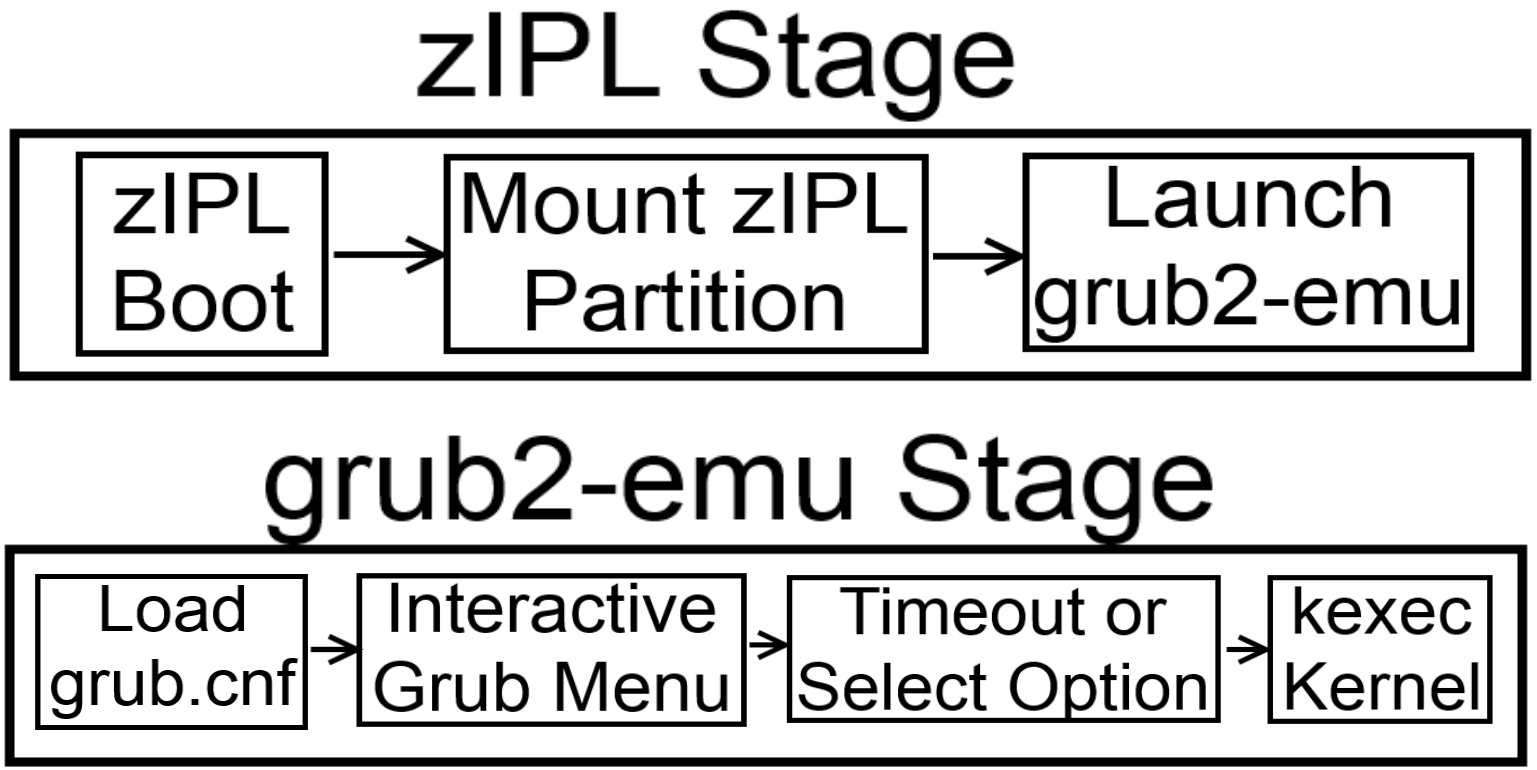
Challenges of Grub 2 on System z

- Large binary size
  - 1.3MB cannot be easily loaded via CCW
- Unique System z drivers
  - DASD, zFCP, QETH all should be cleanly implemented
  - Maintenance a major factor
- Unique display configuration
  - Both 3270 and 3215 need to be supported

Solution: grub2-emu

- User-mode grub2
  - Relatively little modification necessary
- Requires a linux kernel to be booted
  - All supported s390x drivers available
  - udev resolves devices
  - Still uses zIPL, but in a “Stage 4” role
- Full filesystem support
  - BTRFS snapshots
  - Unusual device-mapper configurations
- Two stage boot

Grub 2 Boot Process



How to boot another kernel?

- kexec
  - Kernel-supported reboot
  - Safe shutdown and restart of devices
- Implementation of kexec in Grub 2
  - New feature for grub2-emu
  - Options for kernel and initrd are used and verified
  - Only available to root user

Configuration: grub2

- Special grub2 boot partition
  - Normally mounted to /boot/zipl
- Contains grub2 kernel and initrd
- Contains grub2 config file
  - Allows for dynamic configuration
  - No more need to run zipl when kernel configuration changes

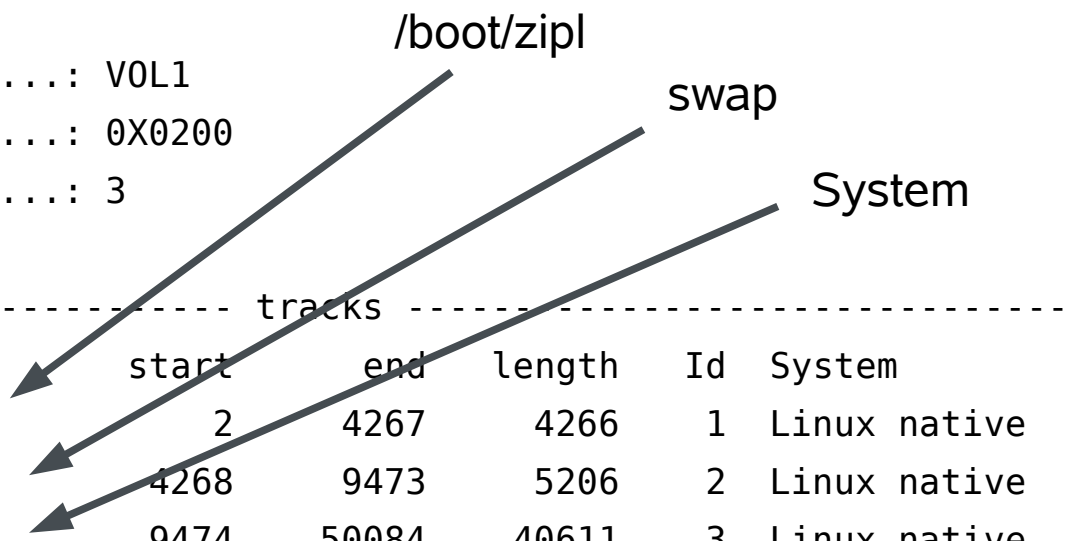
Disk Partitioning

```
# fdasd -p /dev/dasda
...
reading volume label ..: VOL1
reading vtoc ..........: ok
Disk /dev/dasda:
...
  volume label .........: VOL1
  volume serial ........: 0X0200
  max partitions .......: 3

 ------------------------------ tracks ------------------------------
              Device      start      end   length   Id  System
          /dev/dasda1          2     4267     4266    1  Linux native
          /dev/dasda2       4268     9473     5206    2  Linux native
          /dev/dasda3       9474    50084    40611    3  Linux native
exiting...
```

/boot/zipl

swap

System



File Locations

- Multiple binaries are generated from the same source for consistency, then placed in different locations:
  - /boot/.image-3.12.28-4-default.hmac
  - /boot/System.map-3.12.28-4-default
  - /boot/config-3.12.28-4-default
  - /boot/image → /boot/image-3.12.28-4-default
  - /boot/initrd → /boot/initrd-3.12.28-4-default
  - /boot/initrd-3.12.28-4-default-kdump
  - /boot/vmlinux-3.12.28-4-default
  - /lib/modules/3.12.28-4-default/ ...

Btrfs

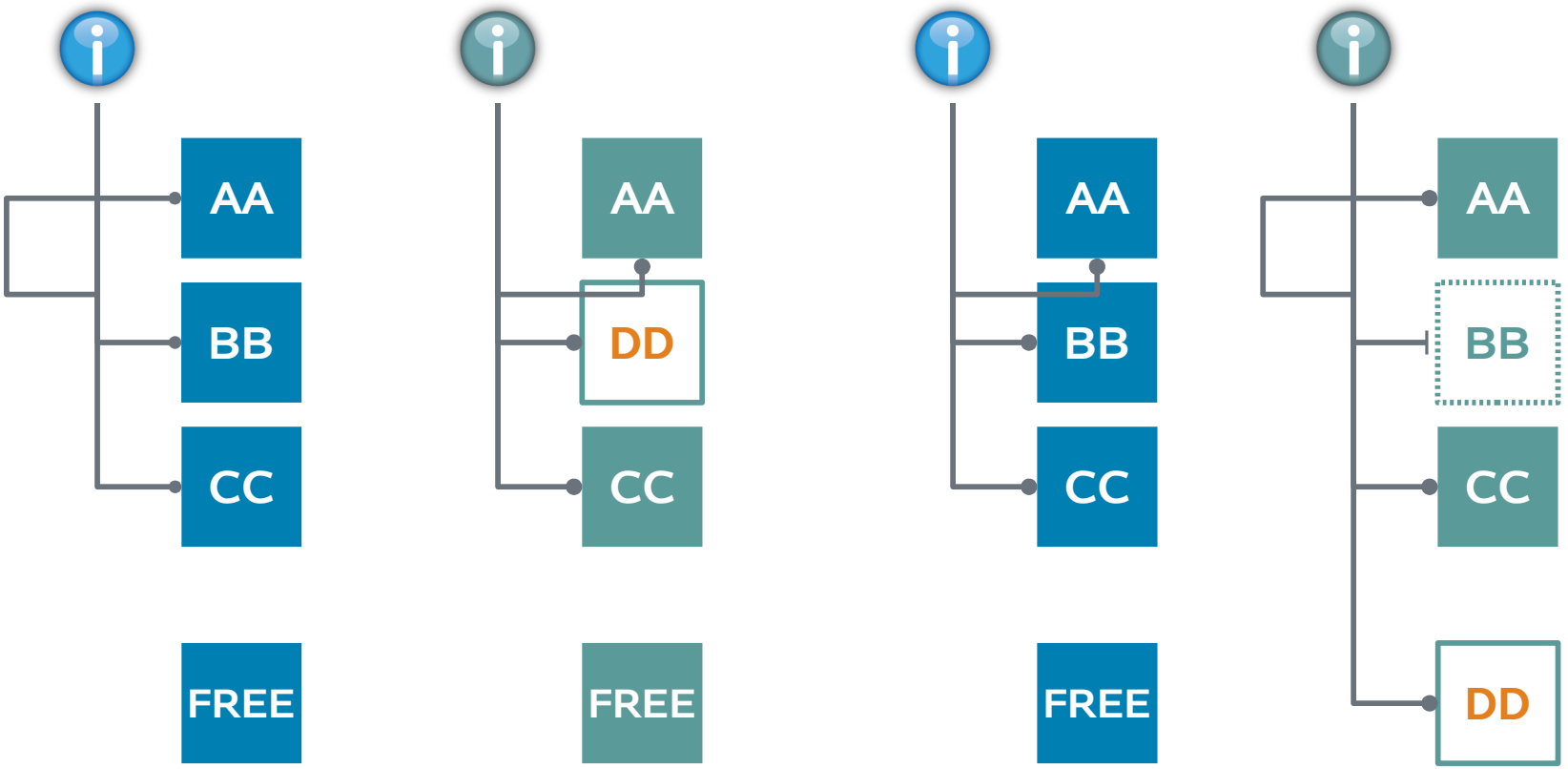
Why btrfs?

Btrfs (better fs) – Features

- Scalability (16 EiB) including effective shrink
- Supports offline in-place migration from ext2, ext3
- Support for Copy on Write
- Powerful Snapshot capabilities
- Other Capabilities:
  - SSD optimization (TRIM support)
  - Data integrity (checksums)

Technology Overview

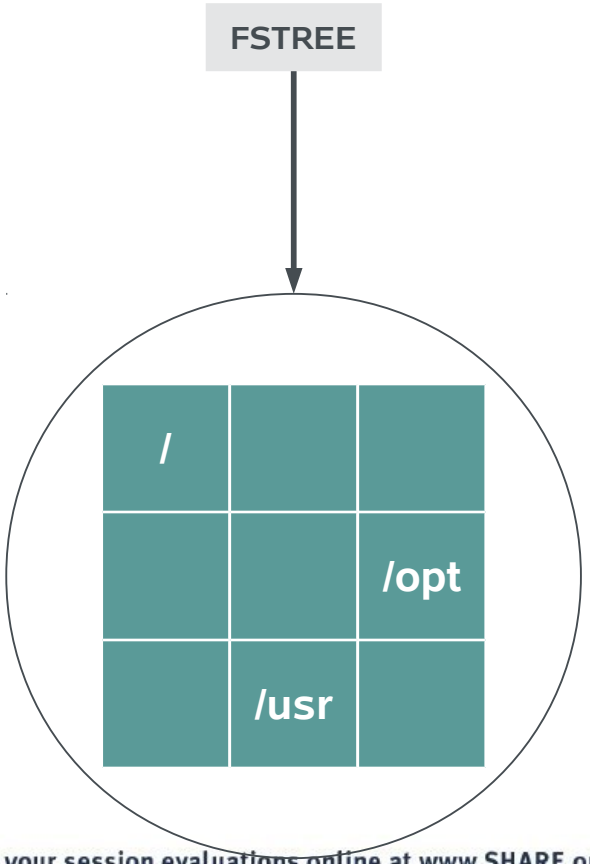
Copy on Write



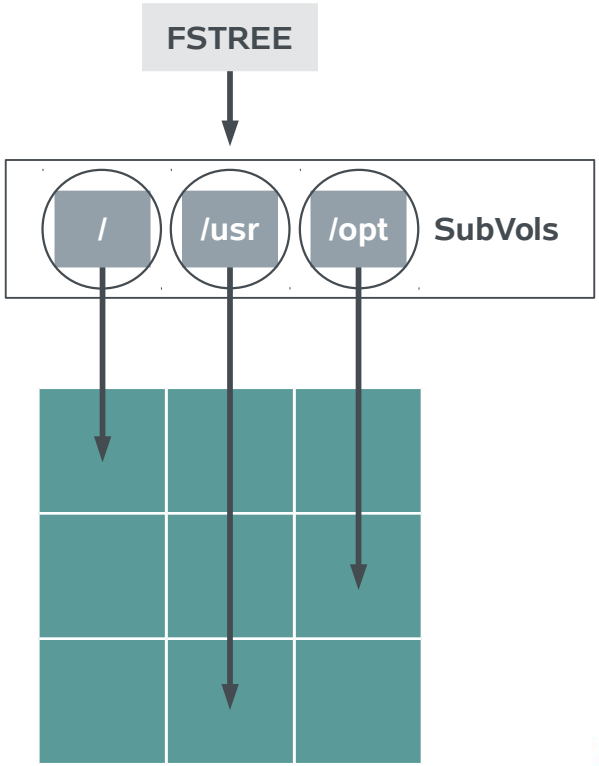
Technology Overview

Subvolume

Normal Filesystem



With Subvolumes



Technology Overview

Subvolume

- A complete filesystem tree
- Usually appears as a sub-directory in the “parent” fs
- Can be mounted separately
- not “just a subdirectory”
- Simliar to
  - two “foreign” filesystems, which are
  - using the same pool of data blocks (and other infrastructure)
- Benefits
  - different parts (subvolumes) of a filesystem can have different attributes, such as quotas or snapshotting rules
  - Copy on Write is possible across volumes
- Basic commandline management
  - “btrfs subvolume …”

Technology Overview

Snapshots

- Copy on Write on a
  - full subvolume tree
  - instead of a single file only
- Every snapshot is again a subvolume of its own
- Snapshots (as subvolumes) can be mounted and accessed as every other subvolume
- Snapshots can be created read-only
- Basic commandline management
  - “btrfs subvolume snapshot …”

Requirements

- Single file rollback (“undochange”)
- Subvolume based rollback
- User interface: cmdline and YaST2 integration
- Snapshots for YaST2 and zypper activities
- Automated snapshots (time based)
- Automated snapshot cleanup (time/number based)
- Allow to work with several kernels

Snapshotting “/” – Challenges

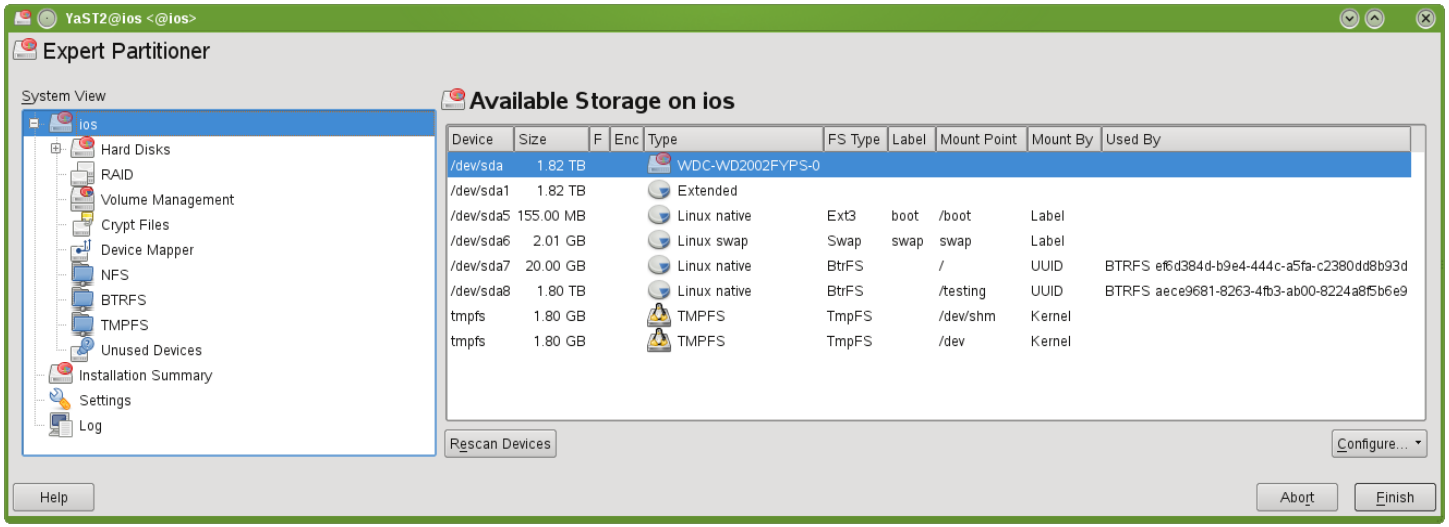
- Multiple Kernels
  - separate /boot
  - zypper integration
- System integrity and Compliance
  - Don't allow to roll back certain log-files etc.
  - Solution: subvolumes instead of directories for

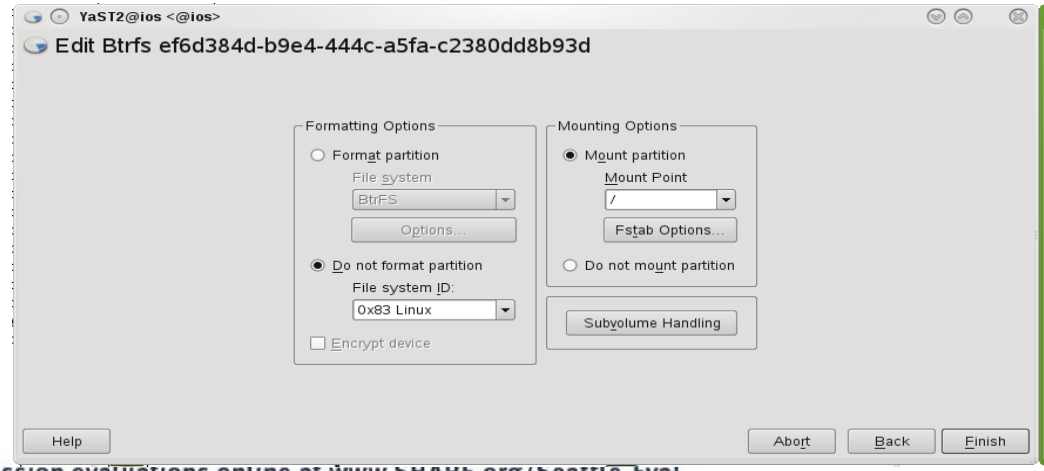
    /tmp
    /opt
    /srv
    /var/spool
    /var/log
    /var/run
    /var/tmp

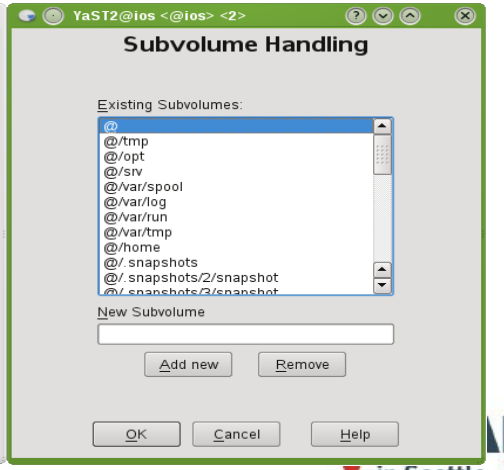
Snapshotting “/”

- We have decided to go the way of “/” in a subvolume
- Disadvantages of this model are mitigated by
  - support from the YaST2 Partitioner to install and configure
  - using “set-default” for the root filesystem to
    - make migration as smooth as possible
    - enable use of “normal” rescue systems

Snapshotting "/" Partitions / Subvolumes







snapper (1)

- cmdline tool to manage snapshots
- Commands
  - snapper list-configs
  - snapper create-config <subvolume>
  - snapper list
  - snapper create
  - snapper modify <number>
  - snapper delete <number>
  - snapper status <number1>..<number2>
  - snapper diff <number1>..<number2> [files]
  - snapper undochange <number1>..<number2> [files]
  - snapper cleanup <cleanup-algorithm>
-

-

snapper (2)

- Configuration files:
  - /etc/snapper/configs/<per subvol>
  - /etc/sysconfig/snapper
- Time based snapshots
- Rules for deleting snapshots automatically

snapper – example

```
2011-08-18 12:33 (0) ~
ios root (0)  # snapper list
Type   | #  | Pre # | Date                                | Cleanup  | Description
-------+----+-------+-------------------------------------+----------+-------------
single | 0  |       |                                     |          | current
single | 1  |       | Wed 17 Aug 2011 04:30:01 PM CEST | timeline | timeline
pre    | 2  |       | Wed 17 Aug 2011 04:31:54 PM CEST | number   | yast lan
post   | 3  | 2     | Wed 17 Aug 2011 04:32:46 PM CEST | number   |
pre    | 4  |       | Wed 17 Aug 2011 04:32:48 PM CEST | number   | yast lan
post   | 5  | 4     | Wed 17 Aug 2011 04:32:59 PM CEST | number   |
pre    | 6  |       | Wed 17 Aug 2011 04:36:10 PM CEST | number   | zypp(zypper)
post   | 7  | 6     | Wed 17 Aug 2011 04:36:11 PM CEST | number   |
pre    | 8  |       | Wed 17 Aug 2011 04:36:16 PM CEST | number   | zypp(zypper)
post   | 9  | 8     | Wed 17 Aug 2011 04:36:19 PM CEST | number   |
pre    | 10 |       | Wed 17 Aug 2011 04:36:26 PM CEST | number   | yast printer
post   | 11 | 10    | Wed 17 Aug 2011 04:37:21 PM CEST | number   |
single | 12 |       | Wed 17 Aug 2011 05:30:01 PM CEST | timeline | timeline
single | 13 |       | Wed 17 Aug 2011 06:30:01 PM CEST | timeline | timeline
single | 14 |       | Wed 17 Aug 2011 07:30:01 PM CEST | timeline | timeline
single | 15 |       | Wed 17 Aug 2011 08:30:01 PM CEST | timeline | timeline
single | 16 |       | Wed 17 Aug 2011 09:30:01 PM CEST | timeline | timeline
single | 17 |       | Wed 17 Aug 2011 10:30:01 PM CEST | timeline | timeline
single | 18 |       | Wed 17 Aug 2011 11:30:01 PM CEST | timeline | timeline
single | 19 |       | Thu 18 Aug 2011 11:30:02 AM CEST | timeline | timeline
single | 20 |       | Thu 18 Aug 2011 12:30:01 PM CEST | timeline | timeline
```

snapper – configuration

```
2011-08-18 13:45 (0) ~
ios root # cat
/etc/snapper/configs/root

# subvolume to snapshot
SUBVOLUME="/"

# filesystem type
FSTYPE="btrfs"

# run daily number cleanup
NUMBER_CLEANUP="yes"
# limit for number cleanup
NUMBER_MIN_AGE="1800"
NUMBER_LIMIT="100"
# create hourly snapshots
TIMELINE_CREATE="yes"
# ...
```

```
# ...
# cleanup hourly snapshots after some
time
TIMELINE_CLEANUP="yes"

# limits for timeline cleanup
TIMELINE_MIN_AGE="1800"
TIMELINE_LIMIT_HOURLY="10"
TIMELINE_LIMIT_DAILY="10"
TIMELINE_LIMIT_MONTHLY="10"
TIMELINE_LIMIT_YEARLY="10"

# cleanup empty pre-post-pairs
EMPTY_PRE_POST_CLEANUP="yes"
# limits for empty pre-post-pair
cleanup
EMPTY_PRE_POST_MIN_AGE="1800"
```

ZYpp Integration

- ZYpp plugin for calling “snapper” on specific actions
- Uses “pairs” of snapshots:
  - one snapshot on ZYpp transaction start
  - one snapshot on ZYpp transaction stop
- Knowledge about pairs is stored in the snapshot metadata

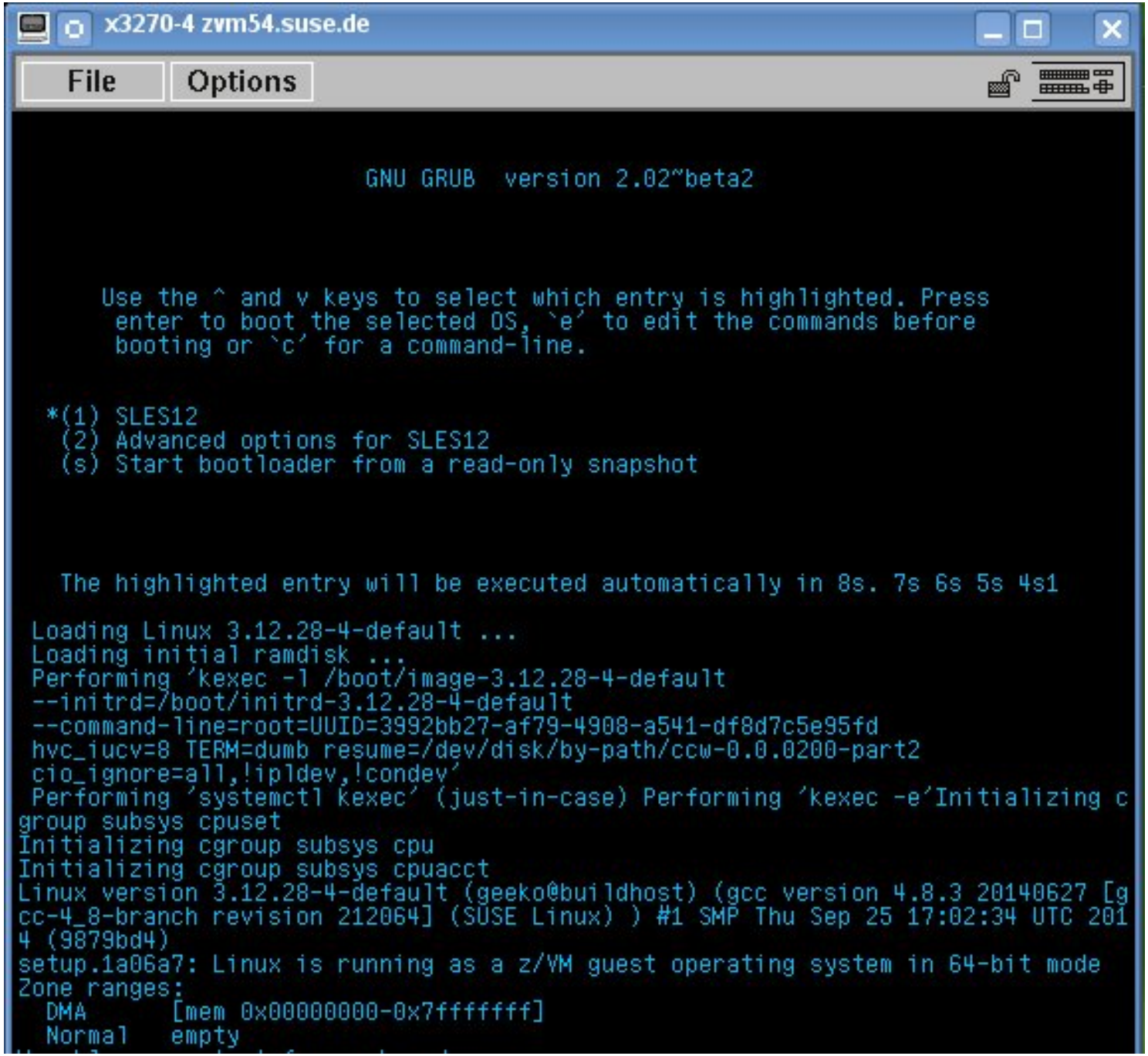
```
2011-08-18 13:34 (0) ~
ios root (0)  # cat /.snapshots/8/info.xml
<?xml version="1.0"?>
<snapshot>
  <type>pre</type>
  <num>8</num>
  <date>2011-08-17 14:36:16</date>
  <description>zypp(zypper)</description>
  <cleanup>number</cleanup>
</snapshot>

2011-08-18 13:34 (0) ~
ios root (0)  # cat /.snapshots/9/info.xml
<?xml version="1.0"?>
<snapshot>
  <type>post</type>
  <num>9</num>
  <date>2011-08-17 14:36:19</date>
  <pre_num>8</pre_num>
  <cleanup>number</cleanup>
</snapshot>
```

YaST2 integration

- Similar to ZYpp
- Snapshot-Pair with every start of a YaST module
- Description covers the name of the module

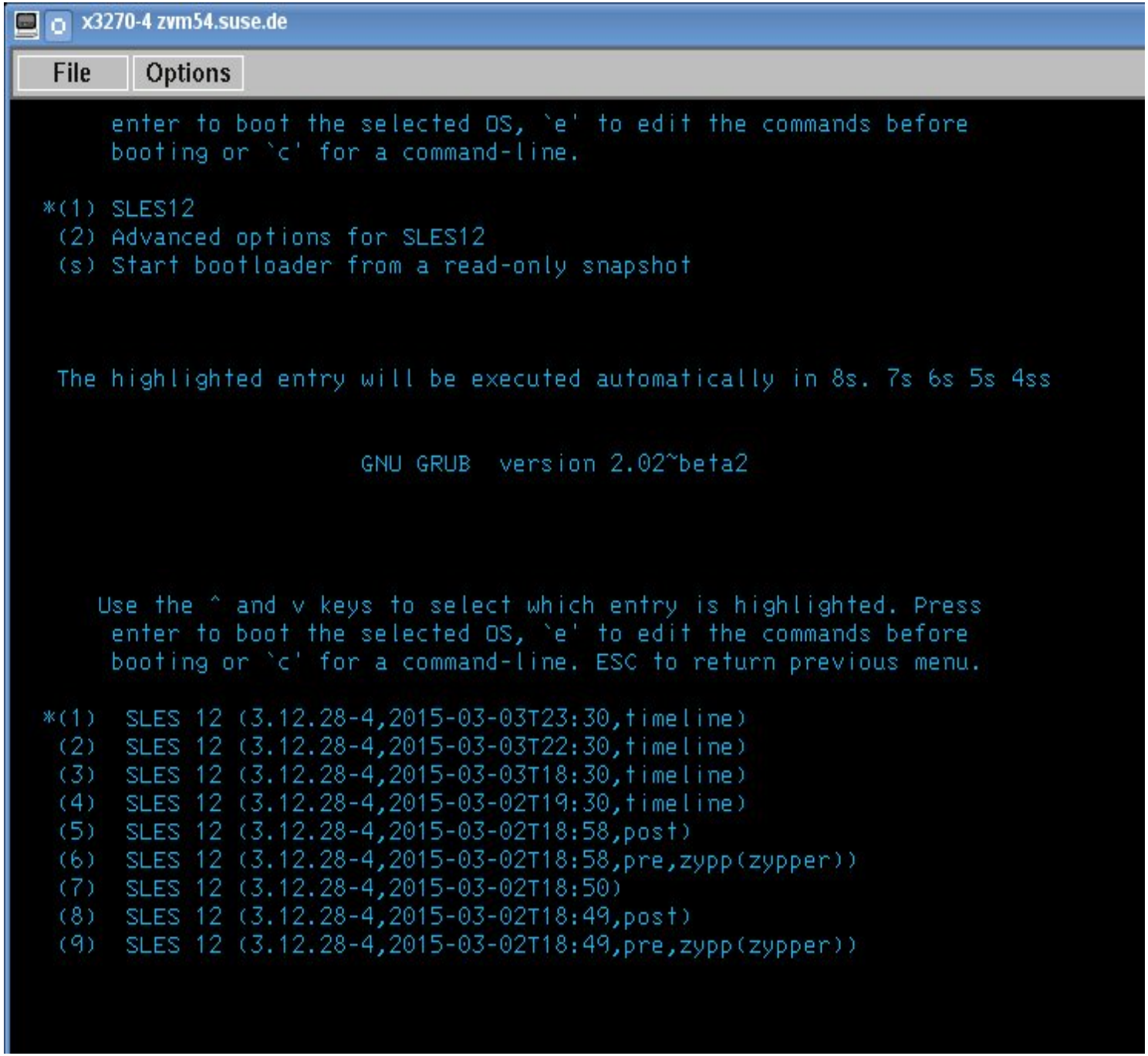
```
2011-08-18 13:32 (0) ~
ios root (0)  # cat /.snapshots/10/info.xml
<?xml version="1.0"?>
<snapshot>
  <type>pre</type>
  <num>10</num>
  <date>2011-08-17 14:36:26</date>
  <description>yast printer</description>
  <cleanup>number</cleanup>
</snapshot>

2011-08-18 13:32 (0) ~
ios root (0)  # cat /.snapshots/11/info.xml
<?xml version="1.0"?>
<snapshot>
  <type>post</type>
  <num>11</num>
  <date>2011-08-17 14:37:21</date>
  <pre_num>10</pre_num>
  <cleanup>number</cleanup>
</snapshot>
```

Demo



SHARE
Educate · Network · Influence

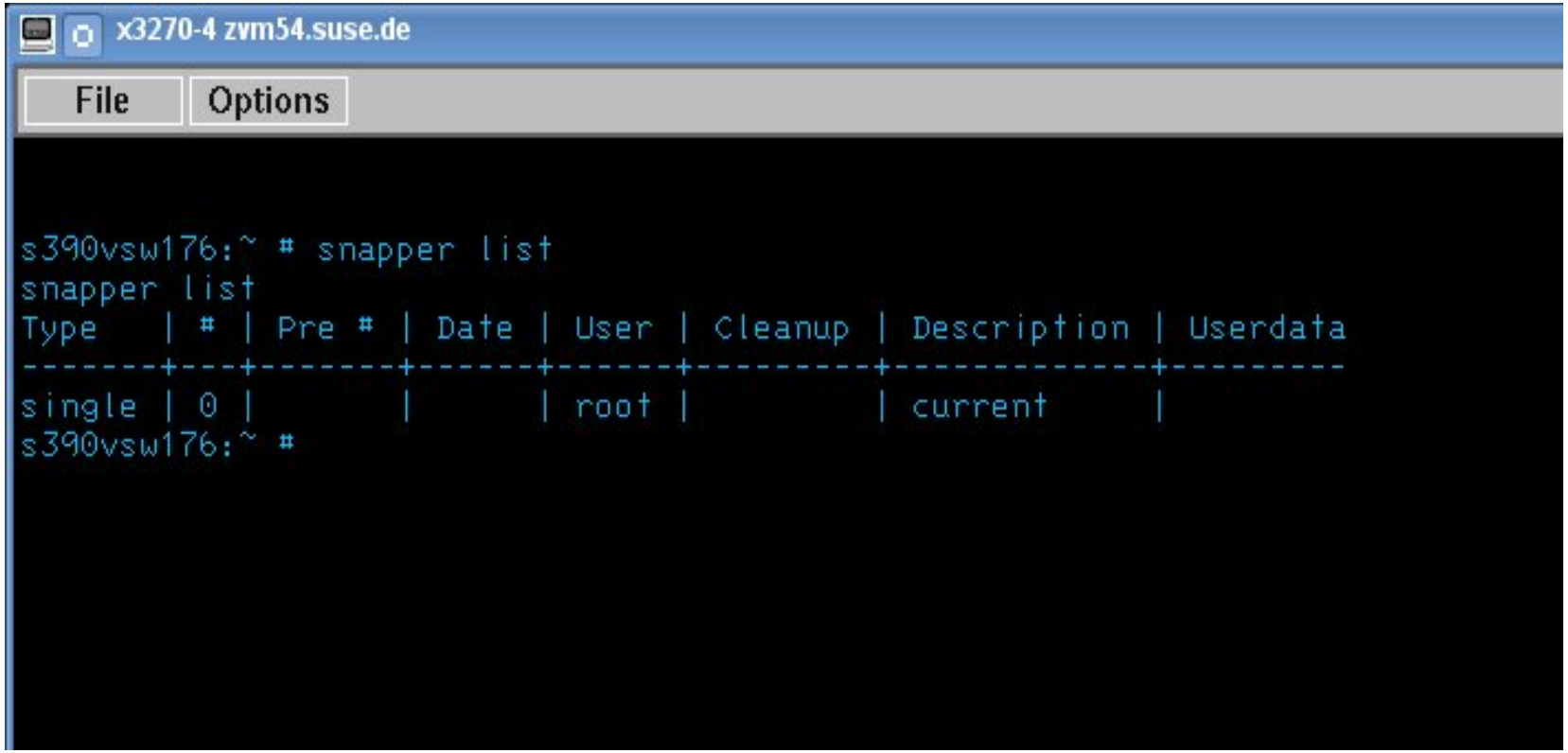
SHARE in Seattle 2015



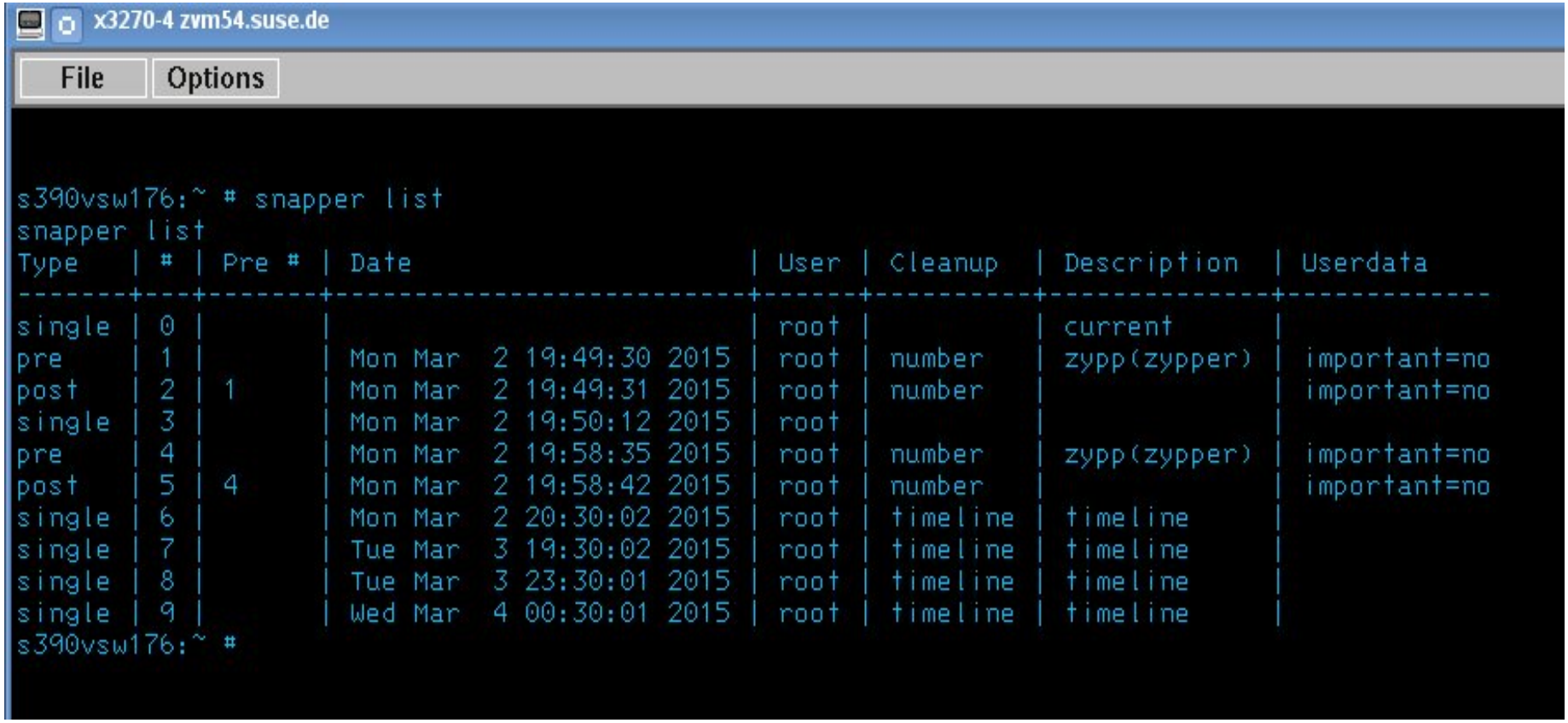
SHARE
Educate · Network · Influence

SHARE in Seattle 2015

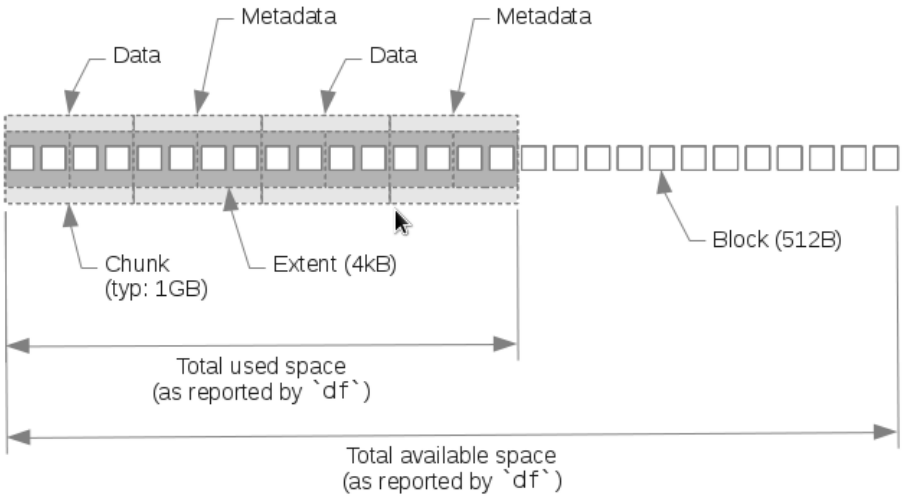
IPL / Boot Snapshot 0



IPL / Boot Snapshot 9



Btrfs Disk Space And Extents

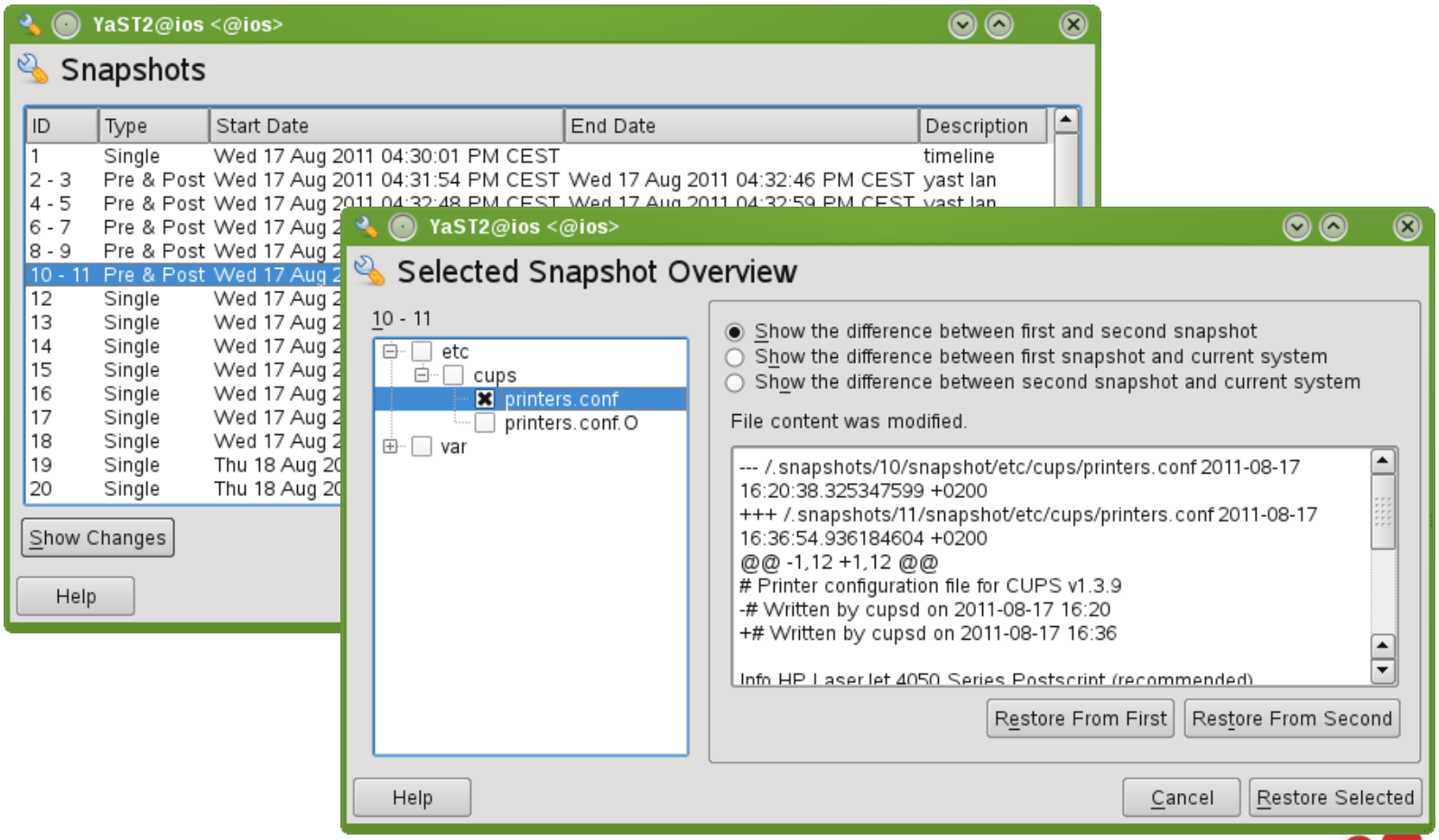


In case of a Btrfs filesystem on a single underlying block device

```
# btrfs filesystem df /
Data: total=14.50GB, used=12.20GB
System, DUP: total=8.00MB, used=12.00KB
System: total=4.00MB, used=0.00
Metadata, DUP: total=1.75GB, used=904.11MB

# df -h /
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda7        20G   14G  4.3G  77% /
#
```

Disk utilization
12,2GB + 2x 0,9GB + = 14 GB

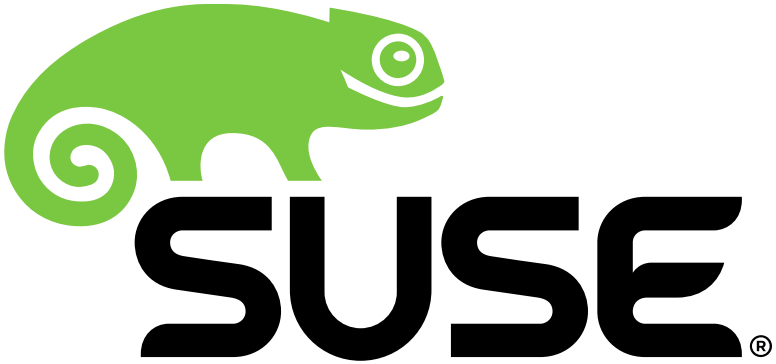
Managing Difference With YaST2



Session 16432

Please Evaluate!

Complete your session evaluations online at www.SHARE.org/Seattle-Eval



Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org

Snapshotting “/” – “/” in a subvolume

- “/” filesystem is in a separate subvolume,
  - mounted as “/” during boot
  - Based on a proposal by the Debian community,
  - this root-subvolume is named “@”
  - Namespace: all Subvolumes starting with “@” are mounted below “@/” automatically
- Benefits
  - Flexible and Future proof
  - subvolume based rollback relatively easy to implement, as all volumes are subvolumes already
- Disadvantage
  - Some learning for people coming from the traditional filesystems
  - “set-default” is necessary to avoid complexity in a first step

napshotting “/” – Traditional Method

- “/” filesystem is mounted as any other Linux filesystem: root volume of the btrfs filesystem = “/” in a user's view
- Benefits
  - No difference for people coming from ext3, xfs, …
  - no “set-default” necessary
- Disadvantages
  - subvolume based rollback introduces additional complexity
  - less flexibility for future improvements,
    - e.g. no “parallel root filesystems”

cgroups

cgroups - Resource Control

Consider a large university server with various users - students, professors, system tasks etc. The resource planning for this server could be along the following lines:

CPUs

Top cpuset (20%)

/ \

CPUSet1 CPUSet2

| |

(Profs) (Students)

60% 20%

Memory

Professors = 50%

Students = 30%

System = 20%

Disk I/O

Professors = 50%

Students = 30%

System = 20%

Network I/O

WWW browsing = 20%

/ \

Prof (15%) Students (5%)

Network File System (60%)

Others (20%)

Source: /usr/src/linux/Documentation/cgroups/cgroups.txt

Improve Operational Efficiency

Systemd: System/Service Manager

- Init Replacement
  - Bring up system and start services
  - Integrate system wide ulimit settings and Cgroups
  - Activation via Socket and d-bus
- Command line “systemctl”
  - Compatibility with SystemV init scripts
  - Provide infrastructure for existing ISV applications
  - LSB compatibility
- SUSE specific usability enhancements
  - Keep insserv, chkconfig and /sbin/service
  - Old style (calling “rc...”) redirected to systemctl
  - LSB compatibility for targets like $network...

Lots Of Mounted Filesystems

- # mount
- sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
- proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
- devtmpfs on /dev type devtmpfs (rw,nosuid,size=244032k,nr_inodes=61008,mode=755)
- securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
- tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
- devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
- tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
- tmpfs on /sys/fs/cgroup type tmpfs (rw,nosuid,nodev,noexec,mode=755)
- cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)
- pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
- cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
- cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpuacct,cpu)
- cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
- cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
- cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
- cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
- cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
- cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
- /dev/dasda3 on / type btrfs (rw,relatime,space_cache)
- systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=28,pgrp=1,timeout=300,minproto=5,maxproto=5,direct)
- debugfs on /sys/kernel/debug type debugfs (rw,relatime)
- mqueue on /dev/mqueue type mqueue (rw,relatime)
- hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
- /dev/dasda3 on /var/tmp type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /var/spool type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /var/opt type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /var/log type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /var/lib/pgsql type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /var/lib/named type btrfs (rw,relatime,space_cache)
- /dev/dasda1 on /boot/zipl type ext2 (rw,relatime)
- /dev/dasda3 on /var/lib/mailman type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /usr/local type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /tmp type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /var/crash type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /srv type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /opt type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /home type btrfs (rw,relatime,space_cache)
- /dev/dasda3 on /boot/grub2/s390x-emu type btrfs (rw,relatime,space_cache)

Lots Of Mounted Filesystems

- # mount
- sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
- proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
- devtmpfs on /dev type devtmpfs (rw,nosuid,size=244032k,nr_inodes=61008,mode=755)
- securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
- tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
- devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
- tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
- tmpfs on /sys/fs/cgroup type tmpfs (rw,nosuid,nodev,noexec,mode=755)
- cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)
- pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
- cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
- cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpuacct,cpu)
- cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
- cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
- cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
- cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
- cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
- cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)

Lots Of Mounted Filesystems

- # mount

  systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=28,pgrp=1,timeout=300,minproto=5,maxproto=5,direct)

  debugfs on /sys/kernel/debug type debugfs (rw,relatime)

  mqueue on /dev/mqueue type mqueue (rw,relatime)

  hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)

Lots Of Mounted Filesystems

```
# mount
/dev/dasda1 on /boot/zipl type ext2 (rw,relatime)
/dev/dasda3 on /boot/grub2/s390x-emu type btrfs (rw,relatime,space_cache)
/dev/dasda3 on / type btrfs (rw,relatime,space_cache)

/dev/dasda3 on /var/tmp type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /var/spool type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /var/opt type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /var/log type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /var/lib/pgsql type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /var/lib/named type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /var/lib/mailman type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /var/crash type btrfs (rw,relatime,space_cache)

/dev/dasda3 on /usr/local type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /tmp type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /srv type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /opt type btrfs (rw,relatime,space_cache)
/dev/dasda3 on /home type btrfs (rw,relatime,space_cache)
```

Unpublished Work of SUSE LLC. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE.
Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.