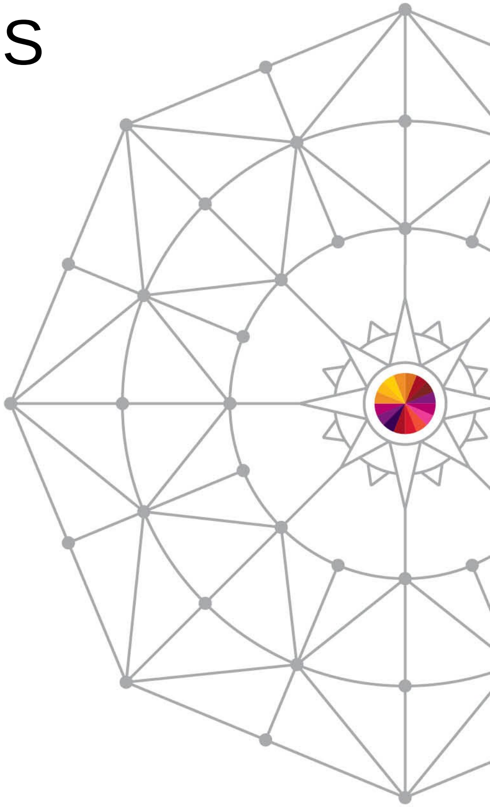
Alternatives to Solaris Containers and ZFS for Linux on System z

Cameron Seader (cs@suse.com)
SUSE

Tuesday, March 11, 2014
Session Number 14540

Agenda

- Quick Overview of Solaris Containers and ZFS
- Linux Containers (LXC)
  - What is LXC?
  - Demo LXC on SLES on System z
- Butterfs (Btrfs)
  - What is Btrfs?
  - Demo Btrfs on SLES on System z



Solaris Containers

- Also known as “Zones”
  - Officially renamed to Oracle Solaris Zones[1]
- Command line tools to manage zones
- Graphical tool “Oracle Enterprise Manager Ops Center” for managing zones
- Dynamically assign resources to a zone or group of zones
- Can run Solaris 8, 9, 10 and some Linux in a zone
  - Using a feature called “branded” zones

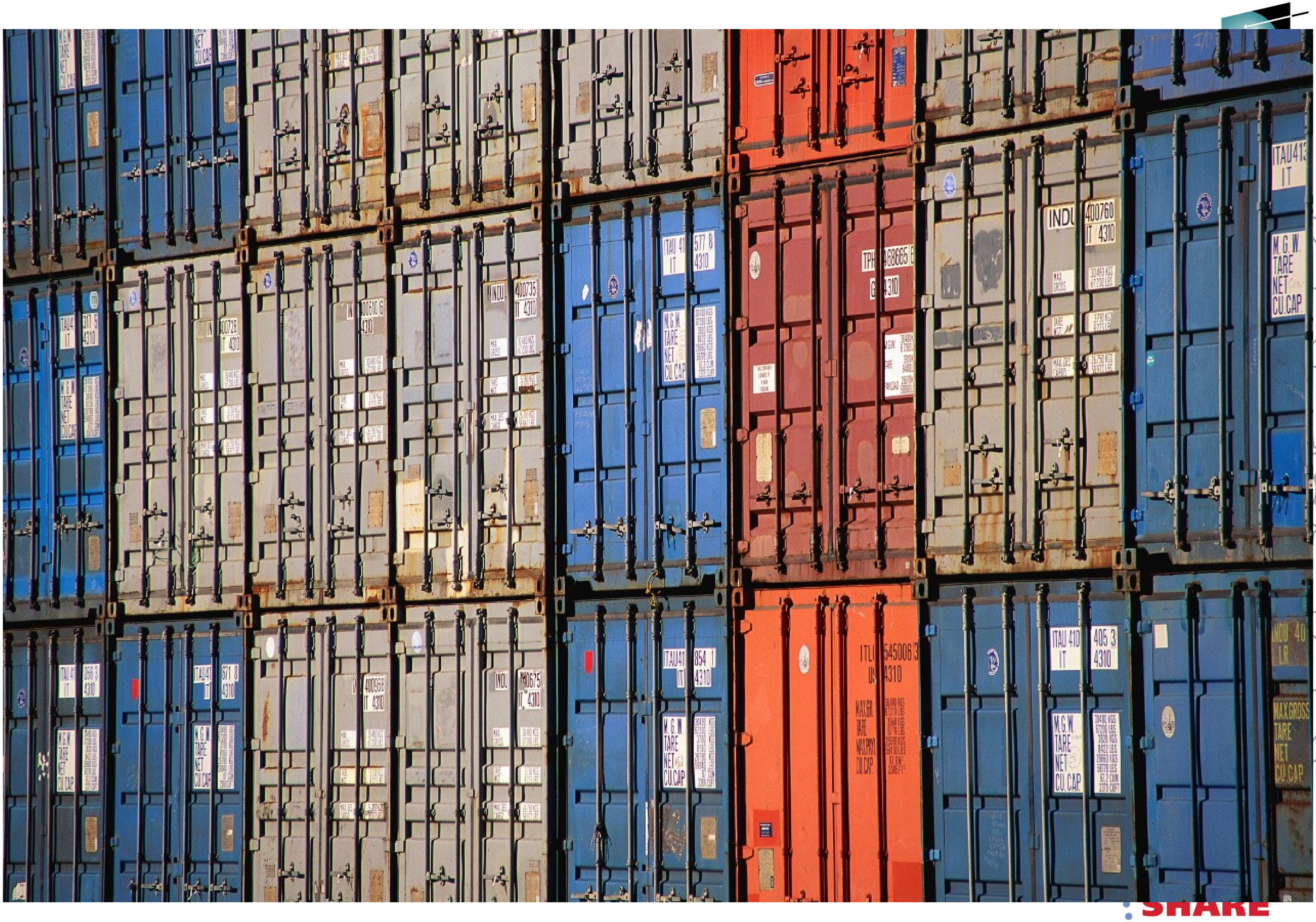
[1] “The Role of Oracle Solaris Zones and Linux Containers in a Virtualization Strategy”,

ZFS

- Combined file system and logical volume manager[1]
- File System
  - Journaling
  - Copy on write
  - Data and metadata verified by checksum
- Integrated Logical Volume Managment
  - Called “Storage Pools”
- Snapshots

[1] “Oracle Solaris ZFS Administration Guide”,

Linux Containers (LXC)



LXC uses a Linux Kernel capability called Control Groups

Control Groups provide a mechanism for aggregating/partitioning sets of tasks, and all their future children, into hierarchical groups with specialized behavior.

- cgroup is another name for Control Groups
- Partition tasks (processes) into a one or many groups of tree hierarchies
- Associate a set of tasks in a group to a set subsystem parameters
- Subsystems provide the parameters that can be assigned
- Tasks are affected by the assigning parameters

Example of the Capabilities of a cgroup

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

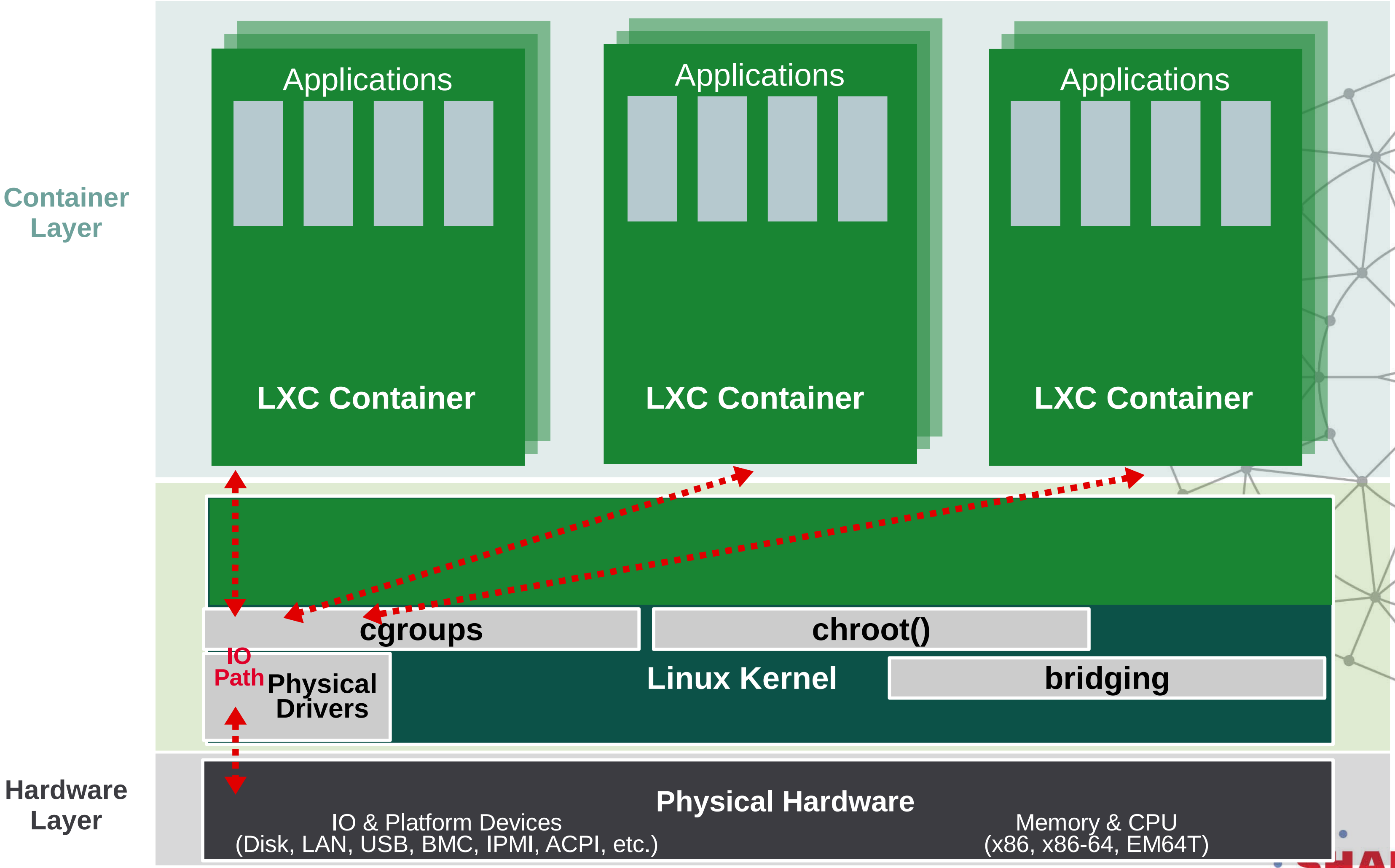
Control Group Subsystems

Two types of subsystems

- Isolation and special controls
  - cpuset, namespace, freezer, device, checkpoint/restart
- Resource control
  - cpu(scheduler), memory, disk i/o, network

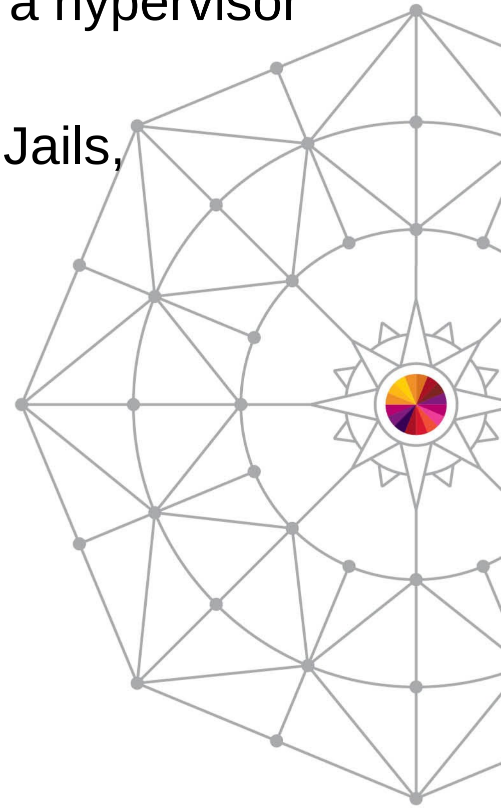
Source: http://jp.linuxfoundation.org/jp_uploads/seminar20081119/CgroupMemcgMaster.pdf

Linux Containers



Linux Containers – Virtualization

- OS Level Virtualization – i.e. virtualization without a hypervisor (also known as “Lightweight virtualization”)
- Similar technologies include: Solaris Zones, BSD Jails, Virtuozzo or OpenVZ
- Advantages of OS Level Virtualization
  - Minor I/O overhead
  - Storage advantages
  - Dynamic changes to parameters without reboot
  - Combining virtualization technologies
- Disadvantages
  - Higher impact of a crash, especially in the kernel area
  - Unable run another OS that cannot use the host's kernel



Linux Containers – Feature Overview

- Supported in SUSE® Linux Enterprise Server 11 SP3:
  - Support for system containers
    - A full SUSE Linux Enterprise Server 11 SP2 installation into a chroot directory structure
  - Bridged networking required
  - Only SUSE Linux Enterprise Server11 SP3 supported in container
  - Easy application containers creation and management
  - Support for AppArmor and LXC integration
- Planned for future SUSE Linux Enterprise Server:
  - Filesystem copy-on-write (btrfs integration)
    - Partial support in SLES11 SP2 LXC update
  - Application containers support
    - Just the application being started within the container

Several Ideas for using LXC on SLES on System z

- Test installation and configuration of an application
- Give developers their “own” system without having to manage separate z/VM guests
- Run multiple applications on a single guest
  - With different IPs per LXC container
  - Limit any combination of CPU, memory and disk resource per LXC container
- Control an application that becomes a resource hog

Demo LXC on SLES on System z

Butterfs (Btrfs)

Why Another Linux filesystem?

- Solve Storage Challenges
  - Scalability
  - Data Integrity
  - Dynamic Resources (expand and shrink)
  - Storage Management
  - Server, Cloud – Desktop, Mobile
- Compete with and exceed the filesystem capabilities of other Operating Systems

What People Say About Btrfs...

Chris Mason (lead developer Btrfs)

- General purpose filesystem that scales to very large storage
- Focused on features that no other Linux filesystems have
- Easy administration and fault tolerant operation

Ted Tso (lead developer Ext4)

- (Btrfs is) “... the way forward”

Others:

- “Next generation Linux filesystem”
- “Btrfs is the Linux answer to ZFS”

A Few Btrfs Concepts

- B-Tree
  - Index data structure
  - Fast search, insert, delete
- Subvolume
  - Filesystem inside the filesystem
  - Independent B-Tree linked to some directory of the root subvolume

- Metadata
  - “normal” metadata: size, Inode, atime, mtime, etc...
  - B-Tree structures
- Raw data
  - Actual content of files

Btrfs Specs

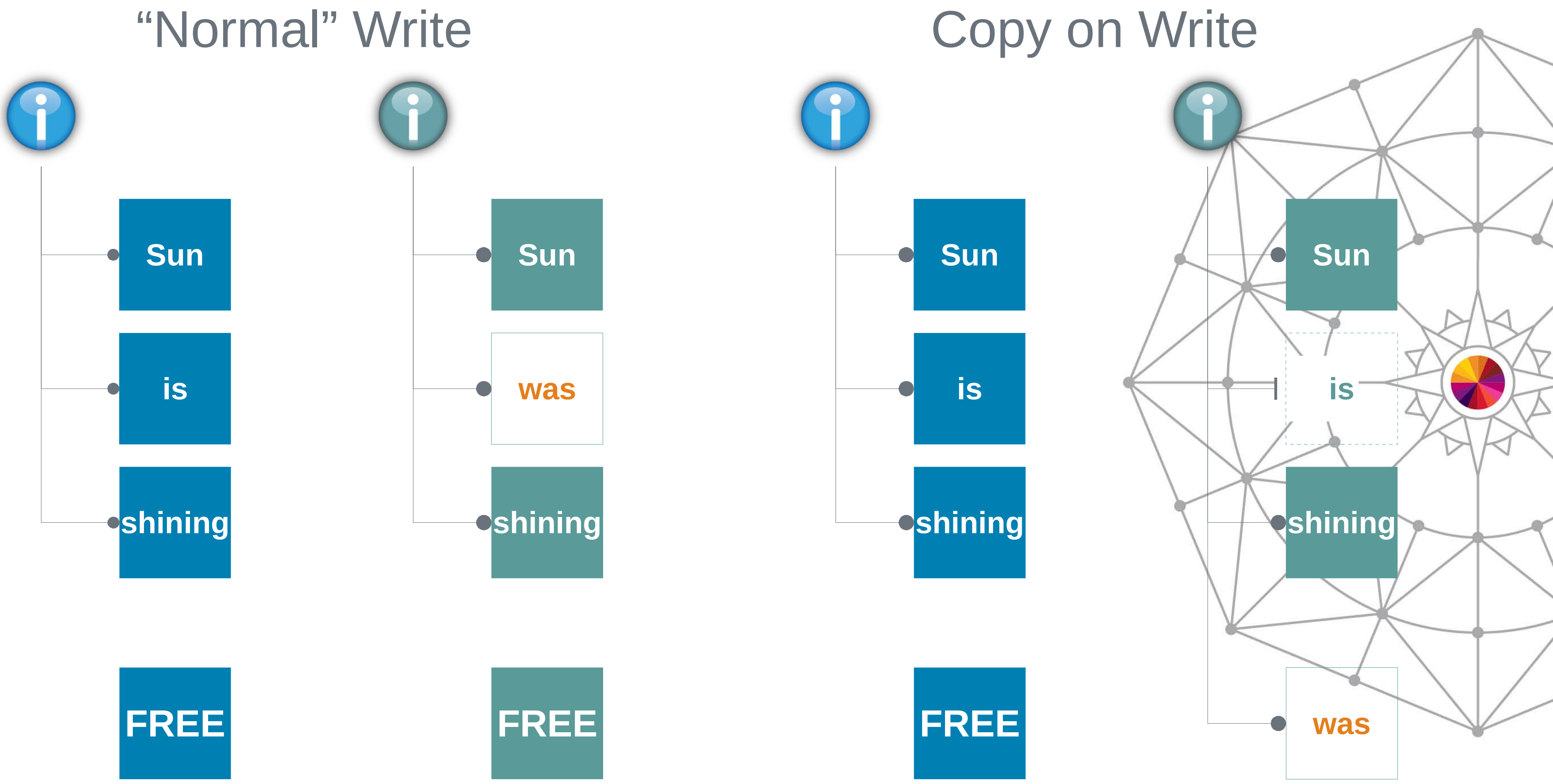
- Max volume size : 16 EB (2^64 byte)
- Max file size : 16 EB
- Max file name size : 255 bytes
- Characters in file name : any, except 0x00
- Directory lookup algorithm : B-Tree
- Filesystem check : on- and off-line
- Compatibility
  - POSIX file owner/permission Hard- and symbolic links,
    Access Control Lists (ACLs) Extended Attributes
    (xattrs),
    Asynchronous and Direct I/O Sparse files

Btrfs Feature Summary

- Extents
  - Use only what's needed
  - Contiguous runs of disk blocks
- Copy-on-write
  - Never overwrite data!
  - Similar to CoW in VMM
- Snapshots
  - Light weight
  - At file system level
  - RO / RW

- Multi-device Management
  - mixed size and speed
  - on-line add and remove devs
- Object level RAID:
  - 0, 1, 10
- Efficient small file storage
- SSD support (optimizations, trim)

Copy on Write explained



Btrfs Feature Summary (cont.)

- Checksums on data and meta data
- On-line:
  - Balancing
  - Grow and shrink
  - Scrub
  - Defragmentation
- Transparent compression (gzip, lzo)
- In-place conversion from Ext[34] to Btrfs
- Send/Receive
  - Similar to ZFS' send/receive function
- Seed devices
  - Overlay a RW file system on top of an RO
- btrfsck
  - Offline FS repair
- Sub-volume Quota support

Btrfs Planned Features

- Object-level RAID 5, 6
- Data de-duplication:
  - On-line de-dup during writes
  - Background de-dup process
- Tiered storage
  - Frequently used data on SDD(s)
  - “Archive” on HDD(s)

Btrfs integration in SLE 11 SP3

Basic integration into

- Installer
  - Btrfs as root file system
  - Recommendation for subvolume layout
- Partitioner
  - Create Btrfs
  - Create subvolumes

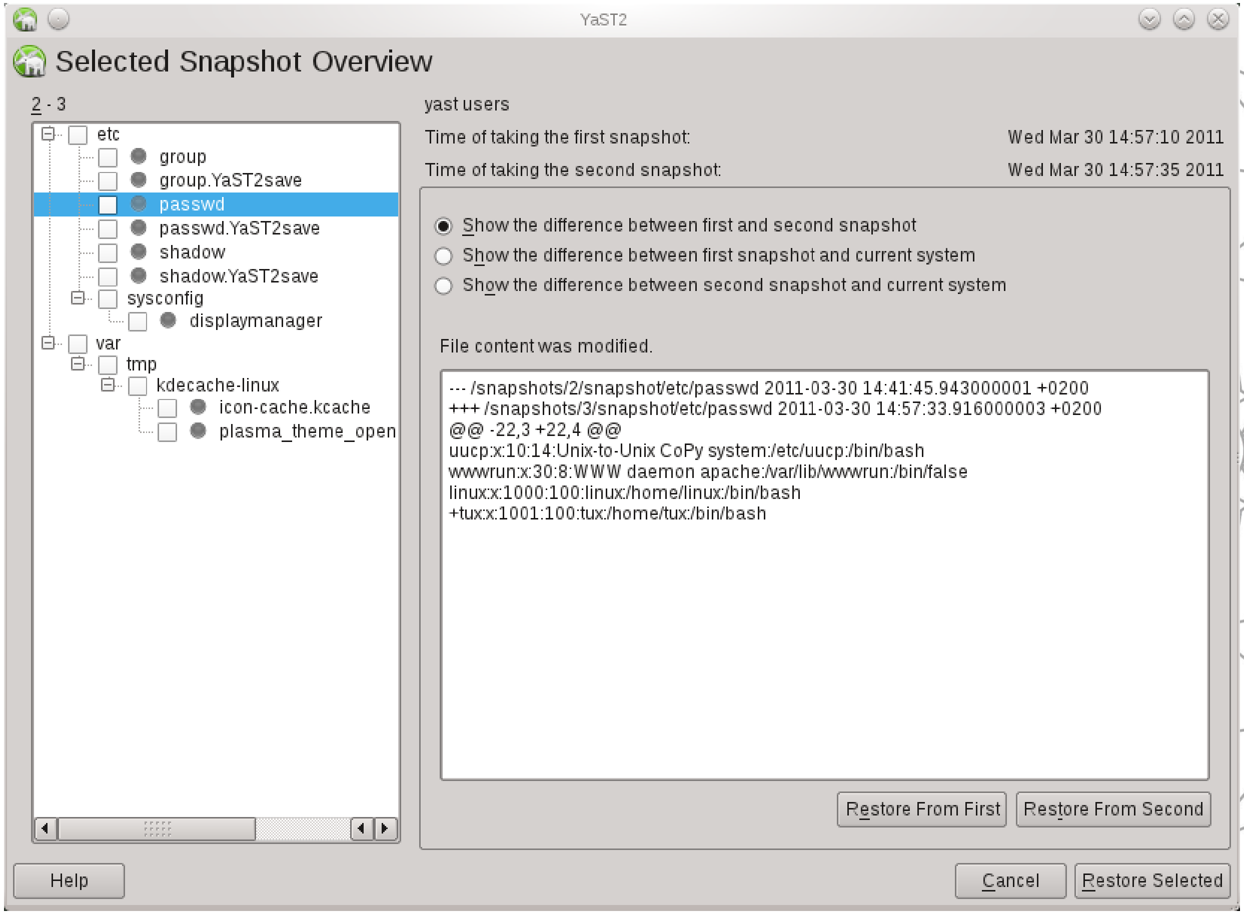
Tools

- Snapper
  - Manage snapshots
  - Automatically create snapshots
  - Display differences between snapshots
  - Faster snapshot comparison
  - Roll-back
  - Snapshot creation as non-root user

Snapshot management with Snapper

Functions

- Automatic snapshots
- Integration with YaST and Zypp
- Rollback
- Integration points



yast users

Time of taking the first snapshot: Wed Mar 30 14:57:10 2011

Time of taking the second snapshot: Wed Mar 30 14:57:35 2011

● Show the difference between first and second snapshot

○ Show the difference between first snapshot and current system

○ Show the difference between second snapshot and current system

File content was modified.

```
--- /snapshots/2/snapshot/etc/passwd 2011-03-30 14:41:45.943000001 +0200
+++ /snapshots/3/snapshot/etc/passwd 2011-03-30 14:57:33.916000003 +0200
@@ -22,3 +22,4 @@
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
linux:x:1000:100:linux:/home/linux:/bin/bash
+tux:x:1001:100:tux:/home/tux:/bin/bash
```

Btrfs integration in SLE
Future Plans

- YaST partitioner support for:
  - Built-in multi-volume handling and RAID
  - Transparent compression
- Transparent compression
- Bootloader support for /boot on btrfs

Several Ideas for using Btrfs on SLES on System z

- Testing a patch on a system
- Rollback after patching a system
  - Rollback of kernel patches with Btrfs not possible due to /boot not being btrfs
- Quickly reset training systems for next class
- Easily fast forward and backward in a demo

Demo Btrfs on SLES on System z

Thank You!!