Using LXC and Btrfs with SUSE Linux Enterprise Server 11 SP2 on System z

Mike Friesenegger
SUSE

Friday, February 8, 2013
Session: 12876
Agenda

• Using Linux Containers (LXC)
  • What is LXC?
  • Demoing LXC on System z
• Why is Btrfs good for Linux on System z
  • Example how Btrfs is useful
Using Linux Containers (LXC)
What Are 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
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:

<table>
<thead>
<tr>
<th>CPUs</th>
<th>Memory</th>
<th>Network I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top cpuset (20%)</td>
<td>Professors = 50%</td>
<td>WWW browsing = 20%</td>
</tr>
<tr>
<td>/</td>
<td>Students = 30%</td>
<td>/</td>
</tr>
<tr>
<td>CPUSet1</td>
<td>System = 20%</td>
<td>\</td>
</tr>
<tr>
<td>\</td>
<td></td>
<td>Prof (15%)</td>
</tr>
<tr>
<td>CPUSet2</td>
<td></td>
<td>Students (5%)</td>
</tr>
<tr>
<td>(Profs)</td>
<td></td>
<td>Network File System (60%)</td>
</tr>
<tr>
<td>60%</td>
<td>Students = 30%</td>
<td>Others (20%)</td>
</tr>
<tr>
<td></td>
<td>System = 20%</td>
<td></td>
</tr>
</tbody>
</table>

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

Linux Containers

Container Layer

Applications
LXC Container
Applications
LXC Container
Applications
LXC Container

Hardware Layer

Physical Drivers
Physical Hardware
IO & Platform Devices (Disk, LAN, USB, BMC, IPMI, ACPI, etc.)
Memory & CPU (x86, x86-64, EM64T)

Linux Kernel

cgroups
chroot()
bridging
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 SP2:
  - 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 Server 11 SP2 supported in container

• Planned for SUSE Linux Enterprise Server 11 SP3 and future:
  - Filesystem copy-on-write (btrfs integration)
    - Partial support in SLES11 SP2 LXC update
  - Application containers support
    - Just the application being started within the container
  - Easy application containers creation and management
  - Research support for AppArmor and LXC
Linux Containers – Use Cases

- Hosting business
  - Give a user / developer (root) access without full (root) access to the “real” system.

- Datacenter use
  - Limit applications which have a tendency to grab all resources on a system:
    - Memory (databases)
    - CPU cycles / scheduling (compute intensive applications)

- Outsourcing business
  - Guarantee a specific amount of resources (SLAs!) to a set of applications for a specific customer without more heavy virtualization technologies
Demoing LXC on System z
Why is Btrfs good for Linux on System z
Data is the customer's gold

Richard Jones, Gartner, formerly Product Manager for SUSE Linux Enterprise Server
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 (xattr)
  - 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

“Normal” Write

Copy on Write

FREE

FREE

FREE

FREE
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
Btrfs Planned Features

- Quota support
  - Aug 2012: 1st implementation available
- 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 SP2

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
  - Roll-back
Btrfs integration in SLE 11
Future plans

- YaST partitioner support for:
  - Built-in multi-volume handling and RAID
  - Transparent compression
- Btrfs support in AutoYaST
- Bootloader support for /boot on btrfs
- Snapshot creation as non-root user (DBus support)
Snapshot management with Snapper

Functions

- Automatic snapshots
- Integration with YaST and Zypp
- Rollback
- Integration points
Example how Btrfs is useful
Session Evaluation
12876 – Using LXC and Btrfs with SLES11 SP2 on System z