

Using Linux Containers as a Virtualization Option

Michal Svec
Product Manager
msvec@suse.com

Mike Friesenegger
Sales Engineer
mfriesenegger@suse.com





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	Disadvantages
<ul style="list-style-type: none">- Minor I/O overhead- Storage advantages- Dynamic changes to parameters without reboot- Combining virtualization technologies	<ul style="list-style-type: none">- Higher impact of a crash, especially in the kernel area- Unable to run another OS that cannot use the host's kernel

Agenda

Control Groups

Introduction to Linux Containers (LXC)

Linux Containers Demo

Questions



Control Groups

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 one or many groups of **tree hierarchies**
- **Associate** a set of tasks in a group to set subsystem parameters
- **Subsystems** provide the parameters that can be assigned
- Tasks are **affected** by the assigned 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](https://www.kernel.org/doc/Documentation/cgroups/cgroups.txt)



Control Group Subsystems

Two types of subsystems:

Isolation and Special Controls

- cpuset, namespace, freezer, device, checkpoint/restart

Resource Control

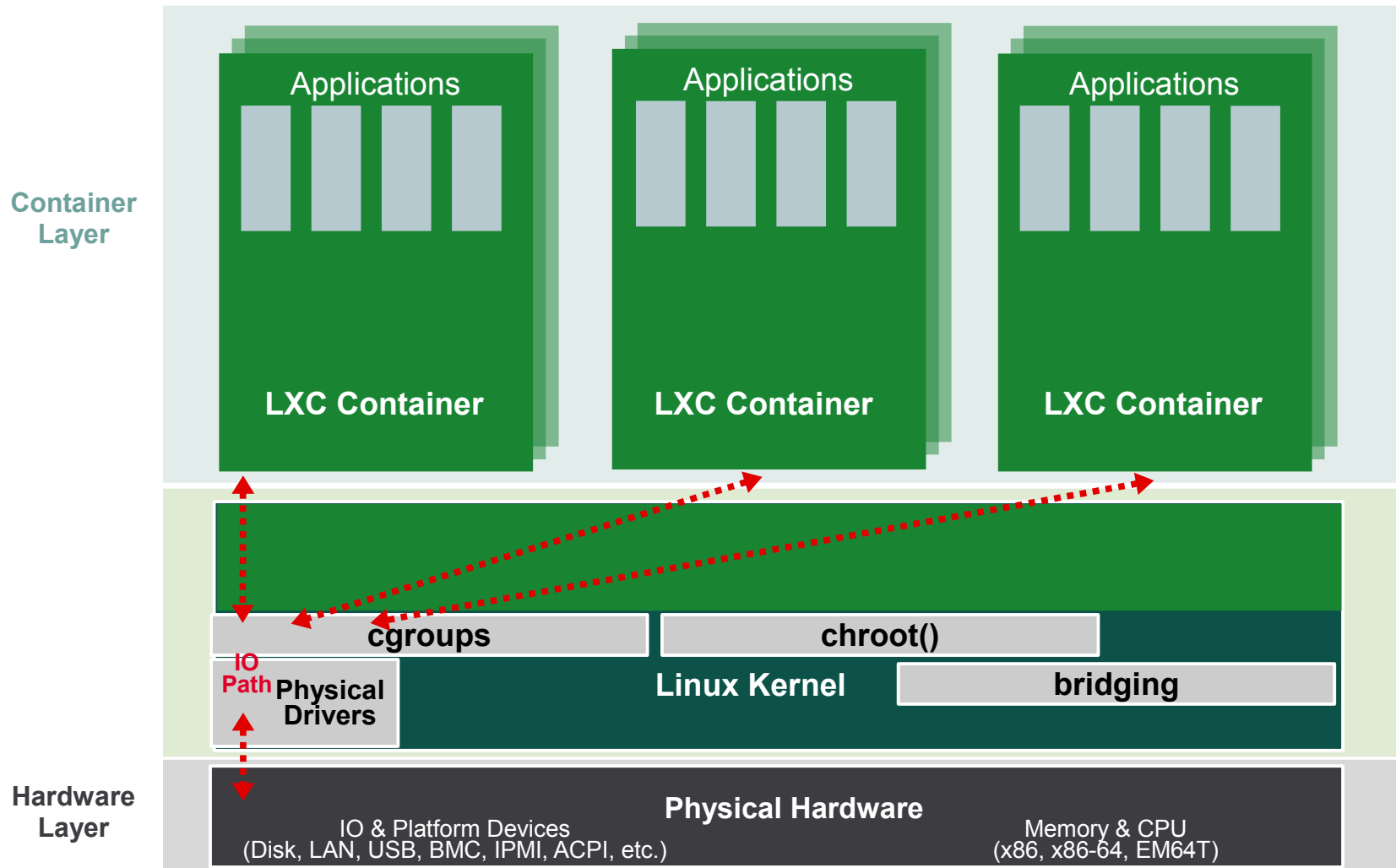
- cpu(scheduler), memory, disk i/o, network

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



Building and Using Linux Containers with LXC

Linux Containers – Architecture



Linux Containers – Security

- User namespaces
 - Prevents evading from containers
- Shared kernel with the host
 - Syscall exploits can be exploited from within the container
 - Solution is in Linux kernel since 3.5 (seccomp2)
- Secure containers with SELinux, AppArmor
 - SELinux policy applies to complete container
 - Support for SELinux with LXC on a case by case basis
 - AppArmor support is ready upstream



Linux Containers – Feature Overview

- Only SUSE Linux Enterprise Server 11 SP3 supported in container
- Support for system containers
- A full SUSE Linux Enterprise Server installation into a chroot directory structure
- Resource control using cgroups
- Bridged networking required



Understanding Solaris Containers

- Also known as a Solaris Zone
- Two components of a Solaris Container
 - Solaris Zones software partitioning technology
 - Solaris Resource Management
- Solaris Zones
 - Virtual mapping from the application to the platform resources
 - Application components are isolated from one another
- Solaris Resource Management
 - Allocate, limit and deny available computing resources
 - Generate extended accounting information for analysis, billing, and capacity planning

Comparing Linux Containers to Solaris Containers

	Solaris Containers	LXC
Included with OS	Yes	Yes
Zone Management		
Command Line Interface	Yes (zonecfg, zoneadm)	Yes (lxc-*)
GUI	Yes (added purchase)	Yes (YaST module)
Fault Isolation		
Application Level	Yes	Yes
Kernel Level	No	No
Privacy/Security	Yes (inside container)	Yes (inside container)
Resource Management		
Project/Task Identifiers	Yes	No
Accounting	Yes	No (evaluating features)
CPU Control	Yes	Yes
Memory Control	Yes	Yes (OOM)
Network Control	Yes	No (in progress)

Linux Containers – Use Cases

- Hosting Business

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

- Enterprise Data Center

- 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



Linux Containers – Outlook

- Cloud use case
 - Bare metal provisioning through the cloud interfaces
- Integration with libvirt: libvirt-lxc
 - Integration with virtualization management tools, incl. cloud
- Application containers
 - More effective storage use
 - Easy creation and management
- Research further container-based technologies
 - systemd, docker.io, ...
- SELinux and AppArmor support for LXC
- File system copy-on-write (btrfs integration)



Linux Containers Demo



Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

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

Join us on:
www.opensuse.org

Unpublished Work of SUSE. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE. 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.

