

LXC

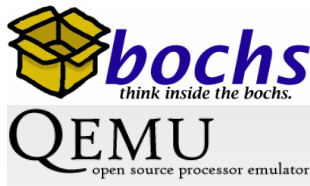
Erkan Yanar

19. März 2011

Virtualisierung zwischen Emulatoren und chroot

Der Begriff Virtualisierung

- Emulation
- chroot



Virtualisierung zwischen Emulatoren und chroot

Der Begriff Virtualisierung

- Emulation
- chroot

SYNOPSIS

```
#include <unistd.h>  
  
int chroot(const char *path);
```

Manual page chroot(2) line 8

Emulation



chroot

Full Virtualization

VirtualBox

Para-Virtualization

Xen

OS-level Virtualization

OpenVZ, Solaris Zones, LXC

Virtualisierungslösungen

Container

- Service/Netzwerk isolieren
- Test/Spielrechner
- Ressourcen dynamisch verteilen
- Native Performance
- Es sind nur Verzeichnisse
- Gängige Commandlinetools (tar, cp ...)
- Hohe Installationsdichte
- kein disjunkter Kernel
- Ideal für RZs mit homogenen Installationen (Linux)
- root in den CT kann wenig
- ...

Überblick

Whats next

Überblick

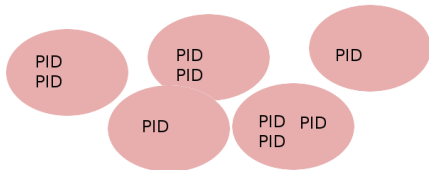
Whats next

- 1** cgroups: Ressourcenmanagment
- 2** LXC: (Teil)Container on top
- 3** OpenVZ vs. LXC

cgroups

Control Groups

- Gruppieren von Prozessen
- gemeinsame Ressourcen
- Childs erben die Gruppe



Control Groups

- Ressourcenverwaltung via VFS
- \geq Kernel 2.6.24
- unabhängig von LXC
- mount: Einbinden von cgroups
- mkdir/rmdir: RessourcenGruppen
- echo: Prozesse zuweisen

Subsysteme

```
/etc/fstab
```

```
cgroup /cgroups cgroup defaults 0 0
```

Zuweisung .. eine falsche

```
# mkdir /cgroups/firefox
```

```
# echo $firefoxpid >/cgroups/firefox/tasks
```

Subsysteme

```
cat /proc/cgroups
```

```
#subsys_name hierarchy num_cgroups enabled  
cpuset 1 5 1  
ns 1 5 1  
cpu 1 5 1  
cpuacct 1 5 1  
memory 1 5 1  
devices 1 5 1  
freezer 1 5 1  
net_cls 1 5 1
```

```
# ls /cgroups
cgroup.procs
cpuacct.stat
cpuacct.usage
cpuacct.usage_percpu
cpu.rt_period_us
cpu.rt_runtime_us
cpuset.cpu_exclusive
cpuset.cpus
cpuset.mem_exclusive
cpuset.mem_hardwall
cpuset.memory_migrate
cpuset.memory_pressure
cpuset.memory_spread_page
cpuset.memory_spread_slab
cpuset.mems
cpu.shares
devices.allow
devices.deny
devices.list
freezer.state
memory.failcnt
memory.force_empty
memory.limit_in_bytes
memory.max_usage_in_bytes
memory.memsw.failcnt
memory.memsw.limit_in_bytes
memory.memsw.max_usage_in_bytes
memory.memsw.usage_in_bytes
memory.soft_limit_in_bytes
memory.stat
memory.swappiness
memory.usage_in_bytes
notify_on_release
tasks
```

cgroups

subsysteme

cpuset. cpus|mems|[cpu|mem]_exclusive
cpu. shares|rt_runtime_us|rt_period_us
cpuacct. stat|usage|usage_percpu
memory. [soft_]limit_in_bytes|use_hierarchy
freezer. echo FROZEN|THAWED > freezer.state
devices. allow|deny
blockio upcomming

```
#cat /cgroups/cpuset.cpus  
0-15  
#echo 0-3 > /cgroups/debian01/cpuset.cpus
```

LXC

LXC

LinuXContainer

LinuXContainer

LXC: better cgroups?

- Spätestens seit 2.6.26 im Kernel (Network-Namespace)
- Mit Hilfe von Namespaces erzeugt LXC Container.
- cgroups dienen der Ressourcenverwaltung.
- LXC übernimmt die Verwaltung der Prozessgruppen
- Modulares Design!

Namespaces

utsname	hostname	[Modular]
Pid	Isolierte PIDs	[Automatisch]
User	Isoliert User	[Automatisch]
Network	Isoliertes Network	[Modular]
Ipc	Isoliertes IPC	[Automatisch]

```
# lxc-checkconfig
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled
```

config

```
lxc.utsname = zeig
lxc.tty = 4
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
lxc.network.hwaddr = 08:00:12:34:56:78
lxc.network.ipv4 = 192.168.1.69
lxc.network.name = eth0
lxc.mount = /lxc/debian/fstab
lxc.rootfs = /lxc/debian/rootfs
lxc.pts = 1024
lxc.cgroup.devices.deny = a
# /dev/null and zero
lxc.cgroup.devices.allow = c 1:3 rwm
lxc.cgroup.devices.allow = c 1:5 rwm
# consoles
lxc.cgroup.devices.allow = c 5:1 rwm
lxc.cgroup.devices.allow = c 5:0 rwm
lxc.cgroup.devices.allow = c 4:0 rwm
```

Network

lxc.network.type

Kein Eintrag Interfaceeinstellungen
 des Hosts

empty loopback

veth Virtual Ethernet
 (bridge)

macvlan MAC-Address based
 Vlan

phys physisches Interface

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
lxc.network.ipv4 =
192.168.1.69
lxc.network.name = eth0
```

Weiters

`lxc.rootfs` chroot

`lxc.mount.entry` Ein Mountpunkt im fstab-Format

`lxc.mount` Pfad zu einem File mit Mountp. im fstab Format

`lxc.tty` Virtuelle Consolen: `lxc-console`

`lxc.pts` Pseudo ttys

```
lxc.tty = 4
```

```
lxc.mount = /lxc/debian/fstab
```

```
lxc.rootfs = /lxc/debian/rootfs
```

Verzeichnisse

`/var/lib/lxc/$CONTAINER` Konfigverzeichnis des Containers

`/var/lib/lxc/$CONTAINER/config` Konfigdatei des Containers

`(lxc.)rootfs` Filesystem des Containers

```
lxc-tool - -name $CONTAINER
```

LXC-Tools

Auszug

lxc-ls Zeigt alle konfigurierten und laufenden Container

lxc-start/stop Starten/Stoppen eines Containers

lxc-ps Wrapper um ps mit Containername

lxc-console Konsolenverbindung zum Container

lxc-execute Starte einen Prozess im ContainerEnvironment

```
# lxc-ls
busy01  debian  first  hiho  lucid
busy01
#
```

LXC-Tools

Auszug

`lxc-ls` Zeigt alle konfigurierten und laufenden Container

`lxc-start/stop` Starten/Stoppen eines Containers

`lxc-ps` Wrapper um `ps` mit Containername

`lxc-console` Konsolenverbindung zum Container

`lxc-execute` Starte einen Prozess im ContainerEnvironment

```
# lxc-start -n busy01
init started: BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu11)

Please press Enter to activate this console.
```

LXC-Tools

Auszug

`lxc-ls` Zeigt alle konfigurierten und laufenden Container

`lxc-start/stop` Starten/Stoppen eines Containers

`lxc-ps` Wrapper um `ps` mit Containername

`lxc-console` Konsolenverbindung zum Container

`lxc-execute` Starte einen Prozess im ContainerEnvironment

```
# lxc-ps --lxc
CONTAINER    PID TTY          TIME CMD
busy01       3971 ?           00:00:00 init
busy01       3975 ?           00:00:00 busybox
busy01       3977 pts/4       00:00:00 getty
busy01       3978 ?           00:00:00 sh
```

LXC-Tools

Auszug

`lxc-ls` Zeigt alle konfigurierten und laufenden Container

`lxc-start/stop` Starten/Stoppen eines Containers

`lxc-ps` Wrapper um `ps` mit Containername

`lxc-console` Konsolenverbindung zum Container

`lxc-execute` Starte einen Prozess im ContainerEnvironment

```
# lxc-console -n busy01  
Type <Ctrl+a q> to exit the console  
busy01 login:
```

LXC-Tools

Auszug

`lxc-ls` Zeigt alle konfigurierten und laufenden Container

`lxc-start/stop` Starten/Stoppen eines Containers

`lxc-ps` Wrapper um `ps` mit Containername

`lxc-console` Konsolenverbindung zum Container

`lxc-execute` Starte einen Prozess im ContainerEnvironment

```
# lxc-execute -n shell /bin/bash
root@shell:/#
```

LXC-Tools

Auszug

`lxc-checkconfig` Zeigt die Kernelconfiguration

`lxc-create` Erstellt einen Container

`lxc-destroy` Löscht das Verzeichnis mit der Config!

LXC-Tools

Auszug

`lxc-checkconfig` Zeigt die Kernelconfiguration

`lxc-create` Erstellt einen Container

`lxc-destroy` Löscht das Verzeichnis mit der Config!

Unnötiges Commando?

```
lxc-create -n name [-f config_file] [-t template]
```

- Kopiere `config_file` nach `/var/lib/lxc/$name/config`
- Führe `template` (evtl) in `rootfs` (`config_file`) aus.
- mehr zu `templates`?

lxc-start und lxc-execute

lxc-start

Startet den Container

lxc-execute

Startet eine Applikation im Container(Enviroment)

ApplikationsContainer

- Modularität Ausnutzen
- libcgroup-Ersatz
- mit lxc.rootfs != / tricky
- ++

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskspace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskspace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskpace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
 - i.e. fork-bombe (numproc)
 - diskspace
 - ioprio
 - vzctl: onboot, userpassword, searchdomain, nameserver ..
 - vzctl (- - save) vs. lxc-cgroup
 - vzcalc
 - Dokumentation
 - Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskspace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskspace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskspace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskspace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskspace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskpace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskspace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

OpenVZ vs. LXC

- Kernel/Distri-Integration
- Livemigration
- venet
- Garantierter Speicher
- i.e. fork-bombe (numproc)
- diskspace
- ioprio
- vzctl: onboot, userpassword, searchdomain, nameserver ..
- vzctl (- - save) vs. lxc-cgroup
- vzcalc
- Dokumentation
- Applikationscontainer

Ende Gelände



erkan yanar

erkan.yanar@linsenraum.de

linsenraum.de/erkules

www.xing.com/profile/Erkan_Yanar