Lightweight virtualization: LXC vs. OpenVZ

Christoph Mitasch
Technology Specialist Thomas-Krenn.AG

Linuxtag Berlin, 11.5.2011

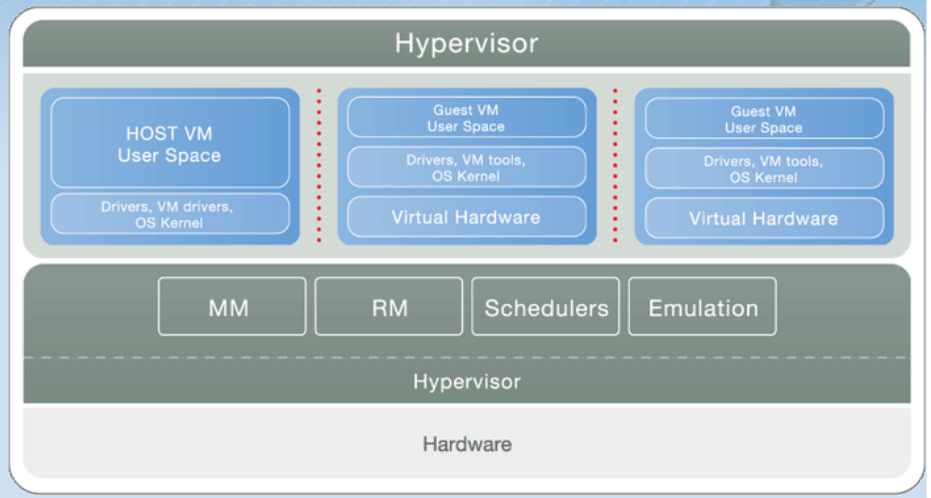
Thomas-Krenn.AG®
Speed is (y)our success

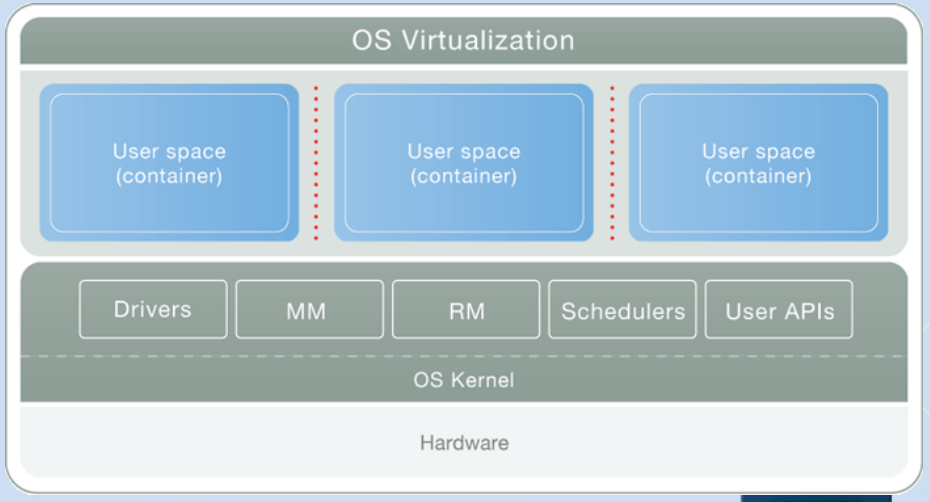
Agenda

1) Virtualisierungstypen
2) OpenVZ
3) LXC
4) OpenVZ vs. LXC
5) Libvirt Support, Pacemaker
6) Migration OpenVZ → LXC

Virtualisierungstypen

- **Hardware Virtualisierung**
  - Full: unmodified Guest OS
  - Para: modified Guest OS
- **Software Virtualisierung**
  - Applikations-Virtualisierung
  - OS-level
    - Linux
      - OpenVZ
      - Linux VServer
      - LXC
    - Solaris: Containers/Zones
    - FreeBSD: Jails





Quelle: http://www.parallels.com/eu/products/pvc46/info/virtualization/

SERVER

Welche Virtualisierung für was?

KVM VPS vs OpenVZ/Virtuozzo vs Xen

	KVM VPS	OpenVZ / Virtuozzo	Xen
Dedicated filesystem of your choice (with direct block level access)	+	–	+
Dedicated RAM with full access and debugging capabilities	+	–	+
Dedicated server like isolation	+	–	+
VNC connection from the very early boot stage	+	–	limited support
PPTP VPN	+	limited support	+
OpenVPN	+	limited support	+
IPSec VPN	+	–	limited support
Firewall Configuration	+	limited support	+
Kernel mode NFS server	+	–	–
Independent kernel	+	–	limited support
Independent kernel modules	+	–	limited support
Full control on sockets and processes	+	–	–
Full guest OS support (Windows, Linux, BSD, OpenSolaris, etc.)	+	–	limited support
Direct dedicated access to PCI / PCIe cards	+	–	limited support
Fine grained swap configuration per VPS	+	–	limited support
Official integration with the Linux kernel	+	–	+

Idee: http://events.linuxfoundation.org/slides/lfcs2010_kolyshkin.pdf OpenVZ
Quelle: http://perfohost.com/kvm.vps.html

Welche Virtualisierung für was?

Fahrrad vs. Auto

Feature	Fahrrad	Auto
Umweltfreundlich	Ja	Nein
Preis	Niedrig	Hoch
Treibstoff benötigt	Nein	Ja
Hält fit	Ja	Nein
Geräusch-Level	Niedrig	Hoch
Führerschein notwendig	Nein	Ja
Parkplatz benötigt	Nein	Ja

Auto vs. Fahrrad

Feature	Auto	Fahrrad
Geschwindigkeit	Hoch	Niedrig
Klimatisierung	Ja	Nein
Mitfahrer	Ja	max. 1
Ladekapazität	Hoch	Niedrig
ABS	Ja	Nein
Gaspedal	Ja	Nein
Elektrische Fensterheber	Ja	Nein

Quelle: http://wiki.openvz.org/Bike_vs_car

OpenVZ – History

- **kommerzielles Produkt „Virtuozzo“ seit 2001**
- **OpenVirtuozzo → OpenVZ**
- **Großteil von Virtuozzo seit 2005 unter GPL als OpenVZ verfügbar**
- **großer Kernel Patch**
  - RHEL 5 Kernel: 2,7 Mio LoC bzw. 85 MB unkomprimiert
    → komplette Mainline Inclusion unwahrscheinlich
  - Teile von OpenVZ werden in Mainline aufgenommen
- **Problem: Patch meistens etliche Minorreleases hinter aktuellem Linux Kernel (2.6.32 vs. 2.6.38)**

OpenVZ – Distro Support

- **Debian**
  - seit Lenny mit dabei `linux-image-openvz-*, vzctl und vzquota`
- **Ubuntu**
  - OpenVZ Kernel war einmalig bei Hardy dabei, seit Lucid nicht mehr
  - LXC von Ubuntu als Ersatz vorgeschlagen
- **OpenVZ Repositories:**
  - stable nur für RHEL 4, RHEL 5

OpenVZ – Userspace Tools

- **vzctl**
  - `vzctl set 102 --hostname vps.thomas-krenn.com --save`
  - `vzctl start 102`
  - `vzctl enter 102`
- **vzlist**

```
# vzlist
      VEID   NPROC   STATUS    IP_ADDR         HOSTNAME
       101   45      running   192.168.1.152   test
       102   11      running   192.168.1.153   vps.thomas-krenn.com
```

- **vzmigrate**
- **generell vz***

OpenVZ – Userspace Tools

- **Beispiel:**

```
# vzctl create 101 --ostemplate debian-6.0
# vzctl set 101 --ipadd 192.168.4.45 --save
# vzctl start 101
# vzctl exec 101 ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:00 init [2]
10068 ?        Rs     0:00 ps ax
21556 ?        S      0:00 /usr/sbin/apache2 -k start
21557 ?        S      0:00 /usr/sbin/apache2 -k start
...
21811 ?        Sl     0:00 /usr/sbin/rsyslogd -c4
21829 ?        Ss     0:01 /usr/sbin/apache2 -k start
21834 ?        Ss     0:00 /usr/sbin/cron
21839 ?        Ss     0:00 /usr/bin/dbus-daemon --system
21861 ?        S      0:00 /usr/sbin/apache2 -k start
22374 ?        Ss     0:00 /usr/sbin/sshd
# vzctl enter 101
bash# logout
# vzctl stop 101
# vzctl destroy 101
```

OpenVZ – Beispiel Konfiguration

- **Beispiel:**

```
ONBOOT="yes"

# UBC parameters (in form of
barrier:limit)
# Primary parameters
AVNUMPROC="40:40"
NUMPROC="300:300"
...
# Secondary parameters
KMEMSIZE="27525120:29360120"
TCPSNDBUF="1277952:2097152"
...
# Auxiliary parameters
LOCKEDPAGES="32:32"
SHMPAGES="8192:8192"
PRIVVMPAGES="245760:267875"
...

# Disk quota parameters (in form
of softlimit:hardlimit)
DISKSPACE="1048576:1153024"
DISKINODES="200000:220000"
```

```
# CPU fair sheduler parameter
CPUUNITS="1000"

VE_ROOT="/vz/root/$VEID"
VE_PRIVATE="/vz/private/$VEID"
OSTEMPLATE="debian-6.0-i386-
minimal"
ORIGIN_SAMPLE="vps.basic"
IP_ADDRESS="192.168.1.153"
HOSTNAME="vps.thomas-krenn.com"
NAMESERVER="192.168.1.1"
```

OpenVZ – Templates

- **Precreated Templates:**
  http://openvz.org/download/template/cache/
  - abgelegt in /var/lib/vz/template/cache/ als .tar.gz
  - kann auch selbst erstellt werden:
    z.B. debootstrap
  - beliebige Distro in .tar.gz einpacken
  - Kleinere Modifikationen notwendig:
    - *sed -i -e '/getty/d' /etc/inittab; update-rc.d -f klogd remove*
    - *…*
  - Abhängigkeit Guest Distro von Host-Kernel
    - z.B.: Debian 6 funktioniert mit RHEL4 Host-Kernel nicht
  - Typische ISO Installation nicht möglich

OpenVZ – Netzwerk

- **venet - Vitual network device**
  - Interface Name innerhalb Container: “venet0”
  - Paket Switching auf IP Basis (Layer 3)
  - keine eigene MAC Adresse
  - kein Broadcast im CT
- **veth – Virtual Ethernet Device**
  - Interface Name innerhalb CT: “ethX”
  - eigene MAC Adresse
  - wird auf Interface von HN gebridged
- **Dedizierte NIC einem Container zuweisen: vzctl set 101 --netdev_add eth1 --save**

Differences between veth and venet

Feature	veth	venet
MAC address	Yes	No
Broadcasts inside VE	Yes	No
Traffic sniffing	Yes	No
Network security	Low [1]	High
Can be used in bridges	Yes	No
Performance	Fast	Fastest

Quelle: wiki.openvz.org

OpenVZ – Ressourcen Management

- **User Beancounters**
  - /proc/user_beancounters

```
       uid  resource           held    maxheld    barrier      limit    failcnt
       508:  kmemsize        1214027    1713049   27525120   29360120          0
             lockedpages           0          0         32         32          0
             privvmpages       59095      69400     245760     267875          0
             shmpages              0          0       8192       8192          0
             dummy                 0          0          0          0          0
             numproc              27         37        300        300          0
             physpages         15461      18973          0 2147483647          0
             vmguarpages           0          0       6144 2147483647          0
          ...
```

- **CPU Scheduler**
- **Disk Quota pro Container**
- **I/O Priority pro Container**

OpenVZ – Checkpoint, Migration

- **Kompletter CT kann in Datei gespeichert werden**
  - Laufende Prozesse, Offene Dateien, Netzwerkverbindungen, Memory, Buffer, …
  - `vzctl chkpnt 101; vzctl restore 101`
- **Dieser Zustand kann wiederhergestellt werden am gleichen Server oder auf anderem -> „Live Migration“**
- **vzmigrate --online <host> <VEID>**

LXC – History

- **LXC sind Userspace Tools für Linux Container basierend auf Mainline Kernel**
- **Linux Container basieren auf:**
  - Kernel Namespaces für Ressourcen Isolation
  - Cgroups für Ressourcen Limitierung
- **seit 2.6.29 sinnvoll verwendbar**
- **als Applikations- und System-Container einsetzbar**

LXC – Distro Support

- **Debian**
  - seit Squeeze mit dabei
    `apt-get install lxc`
  - kein spezieller Kernel notwendig!
- **Ubuntu**
  - seit Lucid mit dabei
- **RHEL**
  - Seit RHEL 6 als Technology Preview dabei
- **SUSE**
  - seit openSUSE 11.2
- **generell jeder Kernel ab 2.6.29 + Userspacetools**

LXC – Userspace Tools

- **lxc-start / lxc-stop**
  - `lxc-start -n vm0 -f /lxc/vm0/config`
- **lxc-create / lxc-destroy**
  - Instanz eines Containers anlegen
    für Start selbst zusätzlich lxc-start notwendig
- **lxc-ls**
  - zeigt erstellte sowie laufende Container an
- **lxc-attach**
  - Kommando direkt in CT ausführen (default: bash)
- **lxc-console**
  - `lxc-console -n vm0 --tty 1`
- **generell lxc-***

LXC – Userspace Tools

- **Beispiel:**

```
# lxc-start -n vm0 -f /lxc/vm0/config -d
# lxc-attach -n vm0
root@vm0 # hostname
vm0
# exit
# lxc-console -n vm0 -t 3

Type <Ctrl+a q> to exit the console

Debian GNU/Linux 6.0 vm0 tty3

vm0 login:
# lxc-ls
vm0
# lxc-freeze -n vm0
# lxc-info -n vm0
'vm0' is FROZEN
# lxc-stop -n vm0
```

LXC – Userspace Tools

- **lxc-checkconfig**
  - überprüft Kernel Namespace und Cgroup Support

```
Found kernel config file /boot/config-2.6.32-5-amd64
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled

--- Control groups ---
Cgroup: enabled
Cgroup namespace: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: missing
Cgroup cpuset: enabled

--- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
File capabilities: enabled
```

LXC – Konfiguration

- **Beispiel: /lxc/vm0.conf**

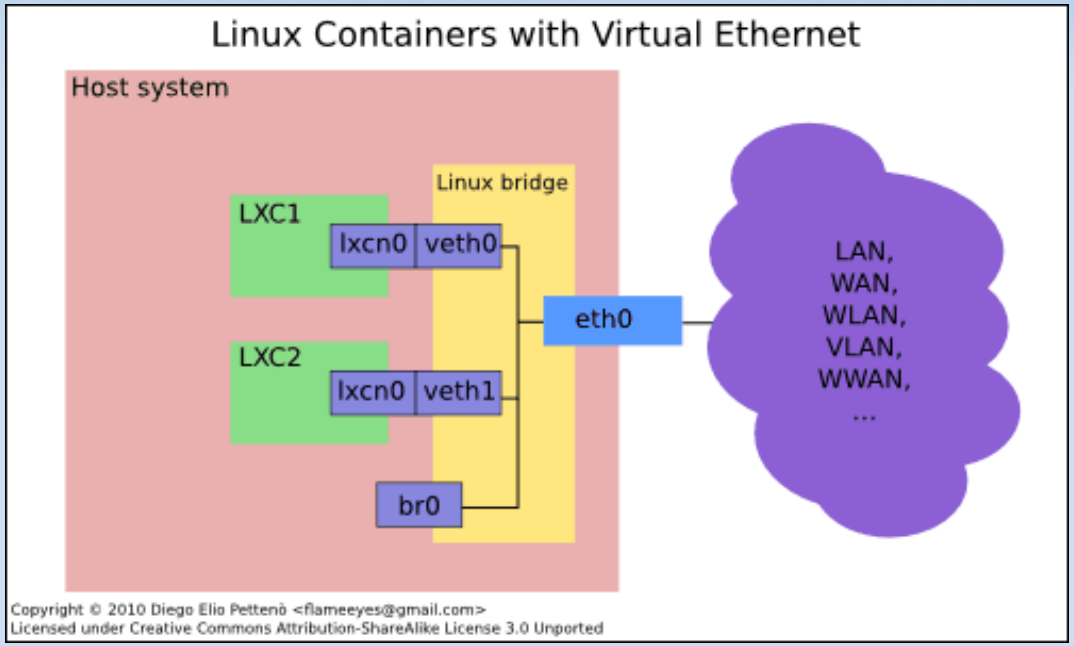
```
lxc.tty = 4
lxc.pts = 1024
lxc.rootfs = /lxc/vm0/
lxc.mount = /lxc/vm0.fstab
lxc.cgroup.devices.deny = a
# /dev/null and zero
lxc.cgroup.devices.allow = c 1:3 rwm
lxc.cgroup.devices.allow = c 1:5 rwm
# consoles
lxc.cgroup.devices.allow = c 5:1 rwm
...

lxc.utsname = lxctest
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0

lxc.cgroup.memory.limit_in_bytes = 512M
```

LXC – Templates

- **keine eigenen precreated Templates**
- **OpenVZ Templates können verwendet werden**
- **können auch mit debootstrap, ... erstellt werden**
- **Template-Skripte**
  - `lxc-debian, lxc-fedora, lxc-ubuntu`
  - Erstellt automatisch Beispiel-Konfiguration
  - Pakete in /var/cache/lxc abgelegt

LXC – Netzwerk

- **Kein Eintrag → Interface-Einstellungen des Hosts**
- **empty**
  → nur Loopback
- **veth**
  → Virtual Ethernet (bridge)
- **vlan**
  → vlan interface
- **macvlan**
  → 3 Modi: private, vepa, bridge
- **phys → dedizierte NIC des Hosts durchgereicht**



LXC – Ressourcen Mgmt. mit Cgroups

- Control groups → cgroups
- als VFS implementiert, seit 2.6.24
  - `mount -t cgroup cgroup /cgroup`
- erlaubt Gruppierung von Prozessen
- Subsysteme (z.B.: blkio, cpuset, memory, …)
- Limitierung, Priorisierung, Accounting
- wird hierachisch weitervererbt
- auch unabhängig von LXC einsetzbar
- in allen aktuellen Distros dabei
- keine Diskspace Limitierung (→ Image File, LVM)

LXC – Ressourcen Mgmt. mit Cgroups

- **Cgroup Limits in LXC Konfiguration setzen, Bsp:**
  - lxc.cgroup.memory.limit_in_bytes = 500M
  - lxc.cgroup.cpuset.cpus = 0
- **Am Host eigene Gruppe unter Containername**

```
root@lxc2:~# cd /cgroup/vm0/
root@lxc2:/cgroup/vm0# ls
...
blkio.reset_stats                    cpu.shares
blkio.sectors                        devices.allow
blkio.throttle.io_service_bytes      devices.deny
blkio.throttle.io_serviced           devices.list
blkio.throttle.write_iops_device     memory.limit_in_bytes
blkio.time                           memory.max_usage_in_bytes
cpuset.mem_exclusive                 notify_on_release
cpuset.mem_hardwall                  tasks
...

root@lxc2:/cgroup/vm0# cat tasks
6549
6756
6810
6813
6814
```

LXC – Namespaces

- **Zur Isolierung von Containern untereinander**

Resource	Status	Article	mainline version
SHARED SUBTREES	Done	lwn	2.6.15
UTSNAME	Done	lwn	2.6.19
PID	Done	lwn	2.6.24
IPC	Done	lwn	2.6.19
USER	Done	lwn	2.6.23
NETWORK	Done	lwn	2.6.26
/PROC	Done	none	2.6.26
RO BIND MOUNT	Done	lwn	2.6.24

Quelle: lxc.sf.net

LXC – Freeze / Unfreeze, Checkpoint

- **Im Moment nur Freeze/Unfreeze möglich**
- **kein kompletter Freeze, werden nur Tasks eingefroren (d.h. ping funktioniert noch)**
- **lxc-freeze / lxc-unfreeze**
- **Checkpointing für Live-Migration in Planung**

LXC – Pittfalls, Recommendations

- **echo b > /proc/sysrq-trigger im Container**
  - mount /proc und /sys readonly im Container
  - drop sys_admin capability
- **Apparmor bzw. SELinux deaktivieren/anpassen**
- **Kernel Logging deaktivieren im Container**
- **lxc.console=/var/log/lxc/vm0.console**
- **evt. Anpassung /lib/init/fstab notwendig**
- **Hwclock Set deaktivieren in Container**
- **root-Zugriff auf Container teilweise problematisch**

OpenVZ vs. LXC

OpenVZ

+ Userspace Tools
+ Live-Migration
+ venet
+ Quota
- Out-of-tree Kernel Patch
- Herstellerabhängigkeit
- Distro-Support
- komplexe Ressourcenlimitierung

LXC

+ Kernel Mainline Support
+ Cgroups
- aufwändige Einarbeitung
- viele Anpassungen für sichere Container notwendig
- „still under development“

```
The lxc is still in
development, so the
command syntax and the API
can change. The version
1.0.0 will be the frozen
version.
```

Libvirt Support, Pacemaker

- **Libvirt unterstützt LXC und OpenVZ**

- **nicht zufriedenstellend bei LXC**
  bei OpenVZ fehlende Funktionen
  → Libvirt Resource Agent „VirtualDomain“ nicht zu empfehlen
- **für OpenVZ eigener „ManageVE“ RA**
- **Anpassung des „ManageVE“ RA für LXC möglich**

Migration OpenVZ → LXC

- **funktioniert problemlos**
- **LXC Konfiguration manuell konvertieren**
- **Aktivieren von TTYs in /etc/inittab für lxc-console**
- **Vorgangsweise:**
  - Konfiguration konvertieren
  - rsync während laufendem Betrieb
  - LXC Container testweise starten und wieder stoppen
  - OpenVZ Container stoppen
  - finaler rsync
  - Starten LXC Container
  - Deaktivieren OpenVZ Container

Zukunft, Resume

- **OpenVZ out-of-tree Patch (85MB) hat keine Zukunft**
- **LXC Mainline Code wird laufend in OpenVZ integriert**
- **Umgekehrt wandern Teile von OpenVZ in Mainline Kernel**
- **→ OpenVZ Patch wird kleiner**
- **Situation ähnlich Xen vs. KVM?**

Danke für die Aufmerksamkeit

cmitasch@thomas-krenn.com
Halle 7.2a, Stand 143