FUJITSU

shaping tomorrow with you

Evalution of Linux Container(LXC) on Embedded Linux

2013.3.8

株式会社富士通コンピュータテクノロジーズ
町田裕樹

目次

FUJITSU

- Linux Container(LXC)の概要
  - Linux Container(LXC)
  - コンテナ型仮想化
  - ユーザランド
- LXCを評価
  - 評価環境
  - 準備
  - アプリケーションコンテナ
  - システムコンテナ

FUJITSU

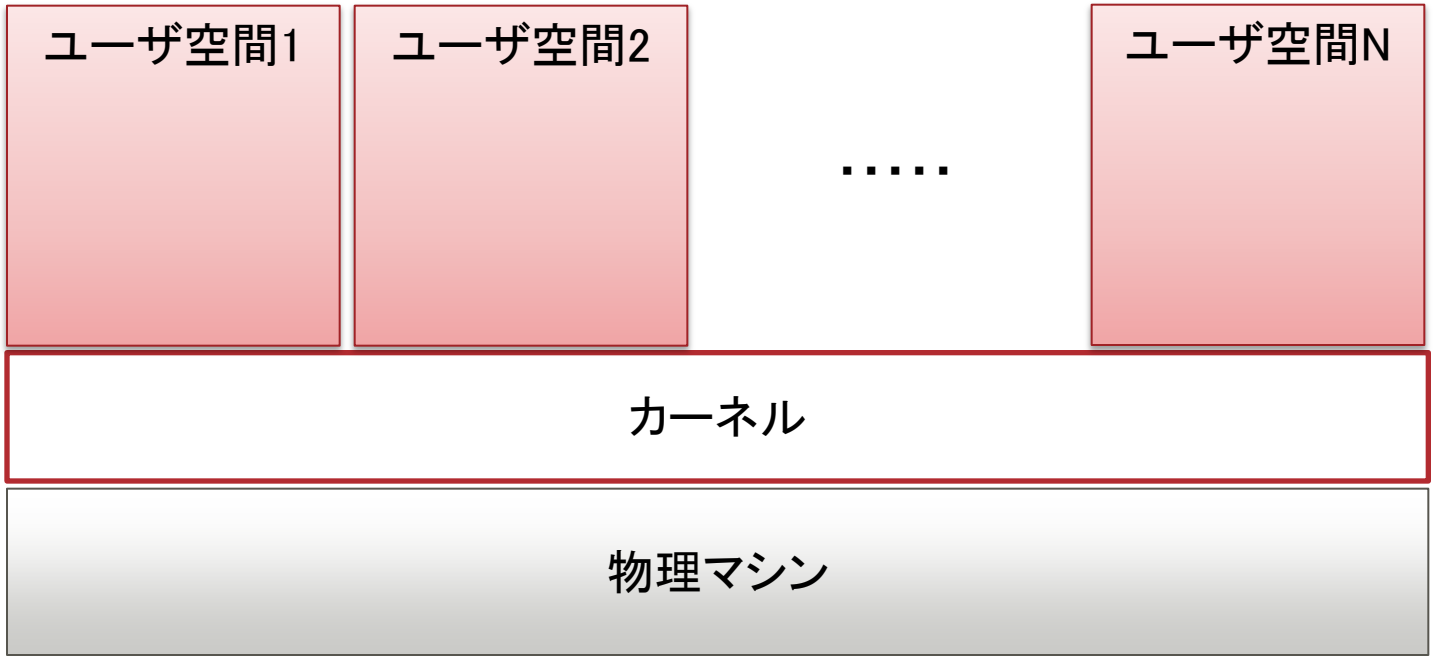
Linux Container(LXC)の概要

- Linux Container(LXC)
- コンテナ型仮想化
- ユーザランド

Linux Container(LXC)

FUJITSU

- Linux Container(LXC)とは
  - 複数のユーザ空間(コンテナ)を持つことができる
  - 軽量でセキュアな仮想環境を実現できる
  - リソース分割、割り当て、制御が可能
    - プロセス: 各ユーザ空間で独立して実行可能
    - デバイス: Cgroupsによりアクセス制限可能



コンテナ型仮想化

FUJITSU

- 関係するカーネルサブシステム
  - Cgroups
    - リソースの制御をする
      (CPU,メモリ,ブロックIO,デバイス)
  - Namespace
    - リソースを独立した空間で利用可能にする
      (PID,ネットワーク,仮想端末 etc…)
- コンテナの種類
  - アプリケーションコンテナ
    - コンテナ上でアプリケーションを実行可能
  - システムコンテナ
    - OSを丸ごとコンテナ上で実行することが可能
    - カーネルをホストOSと共有する
      ※Windowsを動かすことができない

```
-bash-3.2# 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

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

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

カーネルコンフィグレーション

ユーザランド

- LXCを扱うためのツール
  - http://sourceforge.net/projects/lxc/files/latest/download

```
-bash-3.2# ls /usr/bin/lxc-*
lxc-attach      lxc-console   lxc-info      lxc-ps        lxc-stop
lxc-cgroup      lxc-create    lxc-kill      lxc-restart   lxc-unfreeze
lxc-checkconfig lxc-destroy   lxc-ls        lxc-setcap    lxc-unshare
lxc-checkpoint  lxc-execute   lxc-monitor   lxc-setuid    lxc-version
lxc-clone       lxc-freeze    lxc-netstat   lxc-start     lxc-wait
```

liblxcのコマンド群

FUJITSU

LXCを評価

- 評価環境
- 準備
- アプリケーションコンテナ
- システムコンテナ

評価環境

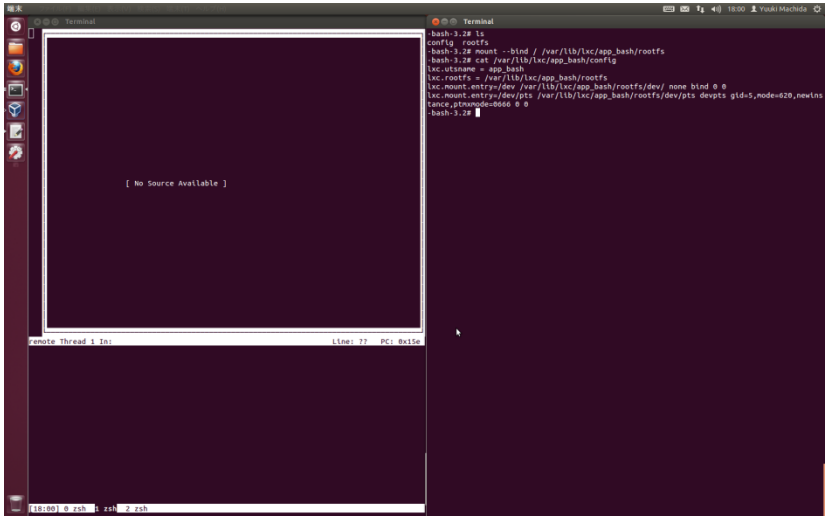
- Embedded Linux(LTSI Kernel 3.4 + Yocto 1.3)
  - カーネル

```
-bash-3.2# uname -a
Linux qemux86 3.4.32 #2 SMP PREEMPT Wed Feb 27 15:28:38 JST 2013 i686 GNU/Linux
```

  - ユーザランド(レシピをFreescale SDKから移植)

```
-bash-3.2# lxc-version
lxc version: 0.8.0-rc1
```

  - QEMU(x86)上で評価



準備(1/2)

FUJITSU

- Cgroupsファイルシステムのマウント

```
-bash-3.2# mount -t cgroup none /sys/fs/cgroup
-bash-3.2# ls /sys/fs/cgroup
cgroup.clone_children                devices.deny
cgroup.event_control                 devices.list
cgroup.procs                         memory.failcnt
cpu.rt_period_us                     memory.force_empty
cpu.rt_runtime_us                    memory.limit_in_bytes
cpu.shares                           memory.max_usage_in_bytes
cpuacct.stat                         memory.memsw.failcnt
cpuacct.usage                        memory.memsw.limit_in_bytes
cpuacct.usage_percpu                 memory.memsw.max_usage_in_bytes
cpuset.cpu_exclusive                 memory.memsw.usage_in_bytes
cpuset.cpus                          memory.move_charge_at_immigrate
cpuset.mem_exclusive                 memory.oom_control
cpuset.mem_hardwall                  memory.soft_limit_in_bytes
cpuset.memory_migrate                memory.stat
cpuset.memory_pressure               memory.swappiness
cpuset.memory_pressure_enabled       memory.usage_in_bytes
cpuset.memory_spread_page            memory.use_hierarchy
cpuset.memory_spread_slab            net_cls.classid
cpuset.mems                          notify_on_release
cpuset.sched_load_balance            release_agent
cpuset.sched_relax_domain_level      tasks
devices.allow
```

準備(2/2)

FUJITSU

- 独立した仮想端末を割り当てる
  - /etc/fstabの/dev/pts行を下記のように変更する。

```
none                /dev/pts       devpts  gid=5,mode=620,newinstance,ptmxmode=0666  0  0
```

  - Multiple /dev/ptsに対応するために下記のコマンドを実行する。

```
rm -f /dev/ptmx
ln -s /dev/pts/ptmx /dev/ptmx
```

アプリケーションコンテナ(1/2)

FUJITSU

- コンテナ上でbashを動かしてみる
  - ホストのrootfsのマウント(評価のため)
    ※ホストとは別のrootfsを利用することもできます

```
-bash-3.2# mount --bind / /var/lib/lxc/app_bash/rootfs
```

  - コンテナの作成

```
-bash-3.2# lxc-create -n app_bash
'app_bash' created
-bash-3.2# lxc-ls
app_bash
```

  - 構成ファイルの編集

```
-bash-3.2# cat /var/lib/lxc/app_bash/config
lxc.utsname = app_bash
lxc.rootfs = /var/lib/lxc/app_bash/rootfs
lxc.mount.entry=/dev /var/lib/lxc/app_bash/rootfs/dev/ none bind 0 0
lxc.mount.entry=/dev/pts /var/lib/lxc/app_bash/rootfs/dev/pts devpts
gid=5,mode=620,newinstance,ptmxmode=0666 0 0
```

アプリケーションコンテナ(2/2)

FUJITSU

- コンテナ上でbashを動かしてみる(続き)
  - コンテナの実行

```
# lxc-execute -n app_bash /bin/bash
```

  - コンテナの終了

```
# lxc-stop -n app_bash
```

※別の端末から実行する

  - コンテナの削除

```
# lxc-destroy -n app_bash
```

システムコンテナ(1/4)

FUJITSU

- コンテナ上でLinuxを動かしてみる
  - コンテナの作成

```
# lxc-create –n sys_elinux
```

  - rootfsの展開と配置
    以下のパスに展開

```
/var/lib/lxc/sys_elinux/rootfs
```

システムコンテナ(2/4)

FUJITSU

- コンテナ上でLinuxを動かしてみる
  - 構成ファイルの編集

```
-bash-3.2# cat /var/lib/lxc/sys_elinux/config
# network configuration
lxc.utsname = sys_elinux
lxc.network.type = empty
lxc.network.flags = up

# file system configuration
lxc.rootfs = /var/lib/lxc/sys_elinux/rootfs
lxc.mount.entry=/dev /var/lib/lxc/sys_elinux/rootfs/dev none bind 0 0
lxc.mount.entry=devpts /var/lib/lxc/sys_elinux/rootfs/dev/pts devpts
gid=5,mode=620,newinstance,ptmxmode=0666 0 0
lxc.mount.entry=proc /proc /var/lib/lxc/sys_elinux/rootfs/proc nodev,noexec,nosuid 0 0
lxc.mount.entry=sysfs /var/lib/lxc/sys_elinux/rootfs/sys sysfs defaults  0 0
```

システムコンテナ(3/4)

FUJITSU

- コンテナ上でLinuxを動かしてみる
  - 構成ファイルの編集(続き)

```
## Devices
#lxc.cgroup.devices.allow          = a
lxc.cgroup.devices.deny           = a
# /dev/null
lxc.cgroup.devices.allow          = c 1:3 rwm
# /dev/zero
lxc.cgroup.devices.allow          = c 1:5 rwm
# /dev/tty[1-4] 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
lxc.cgroup.devices.allow          = c 4:1 rwm
# /dev/{,u}random
lxc.cgroup.devices.allow          = c 1:9 rwm
lxc.cgroup.devices.allow          = c 1:8 rwm
lxc.cgroup.devices.allow          = c 136:* rwm
lxc.cgroup.devices.allow          = c 5:2 rwm
# /dev/rtc
lxc.cgroup.devices.allow          = c 254:0 rwm
```

システムコンテナ(4/4)

FUJITSU

- コンテナ上でLinuxを動かしてみる(続き)
  - コンテナの実行

```
# lxc-start -n sys_elinux
```

  - コンテナの終了

```
# lxc-stop -n sys_elinux
```

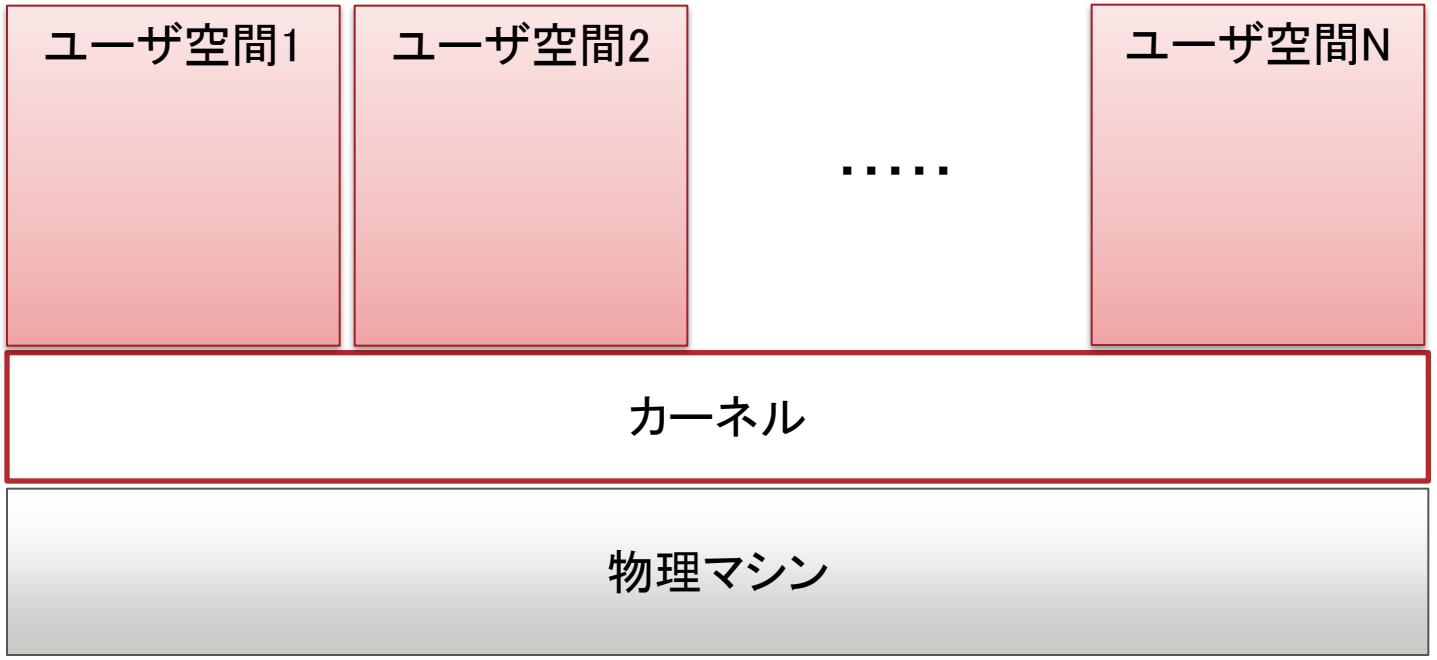
※別の端末から実行する

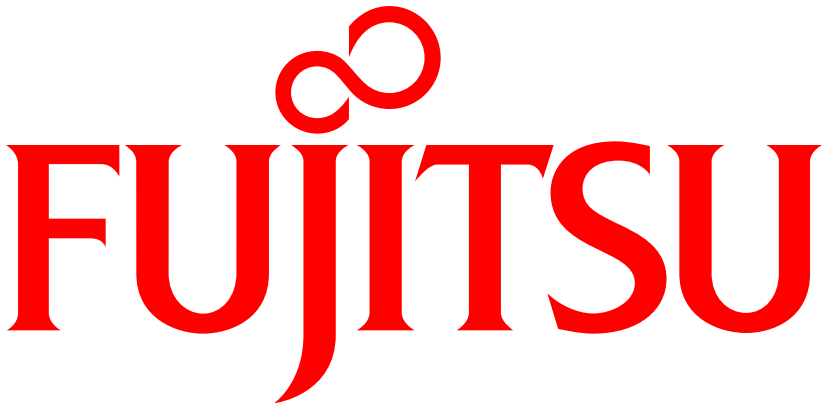
  - コンテナの削除

```
# lxc-destroy -n sys_elinux
```

まとめ

- Linux Container(LXC)とは
  - 複数のユーザ空間(コンテナ)を持つことができる
  - 軽量でセキュアな仮想環境を実現できる
  - リソース分割、割り当て、制御が可能
    - プロセス: 各ユーザ空間で独立して実行可能
    - デバイス: Cgroupsによりアクセス制限可能





shaping tomorrow with you