# Linux コンテナ入門

## コンテナの基礎と最新情報

加藤泰文

第2回 コンテナ型仮想化の情報交換会@東京

# この資料について

この資料は 10/5 に行った『第2回 コンテナ型仮想化の情報交換会@東京』での発表資料に一部加筆を行なっております.

# 自己紹介

- 名前: 加藤泰文
  - http://www.ten-forward.ws/
  - twitter: @ten_forward
  - g+: http://gplus.to/tenforward
  - http://d.hatena.ne.jp/defiant (技術系ネタのブログ)
- 所属:
  - ファーストサーバ株式会社 開発部
- OSS 関係活動
  - Plamo Linux メンテナ，Web ページコンテンツ管理
  - lxc man pages 翻訳
  - Jetspeed 2 ドキュメント翻訳

# 今日の目標

- (初心者の方向け) コンテナの概要と LXC の簡単な使い方を理解いただく
- Linux Kernel のコンテナ関連機能と LXC の更新情報について紹介する
  - 前回勉強会資料からの Update を中心に (kernel 3.8~, LXC 1.0)

- カーネルも LXC も自分が興味のある部分を見ているだけなので，間違いや抜けもあるかもしれません.
- コンテナ関連の最新情報はごく最近の重要なものを紹介します．それ以外は第一回勉強会の資料をご参照ください.
- スライド中には参考文献へのリンクを貼ってあります．後日公開する資料でご確認ください.

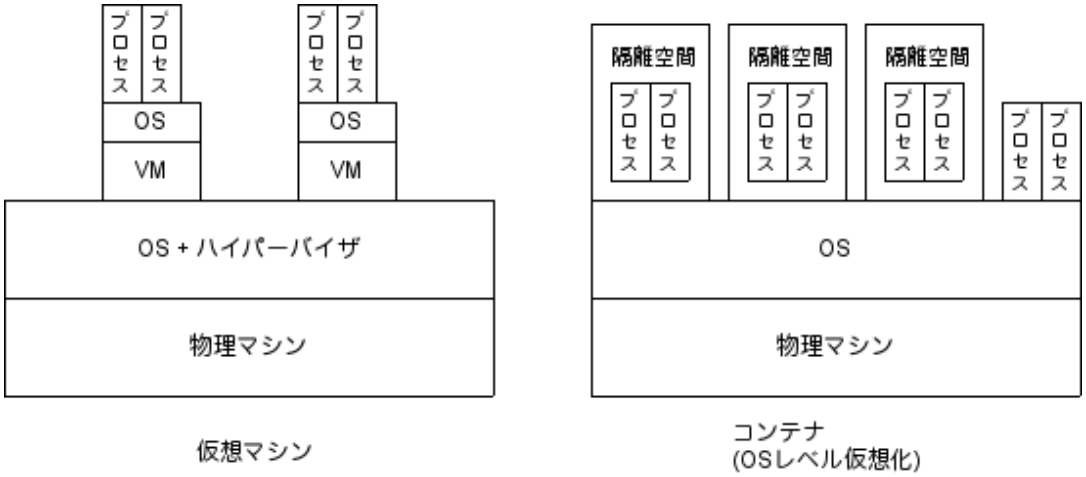
# Agenda

- コンテナの基礎
- Linux におけるコンテナの実装
- Linux コンテナことはじめ
- Linux におけるコンテナの仕組み
- Linux Kernel 最新動向
- LXC 最新動向
- 最後に

# コンテナの基礎

# コンテナとは

- OS レベルの仮想化
- カーネルの機能で (複数の) 独立した空間を作り出し，リソースを分割・分配する
  - プロセスをグループ化して他のグループとリソース空間を隔離
  - グループ化したプロセスに対するリソース制限



# コンテナの特徴

- 高密度化が可能
  - ← 1 つの OS (カーネル) のみが動作している
- オーバーヘッドが小さい
  - ← ハードウェアの仮想化が不要

- 起動が早い
- 必ずしもシステムを動かす必要はない (アプリケーションコンテナ)
  - 例えばコンテナ内では httpd のみが動いている
- 仮想マシンの上でも問題なく動くぜ!w
- 異なる OS のシステム / プログラムは動かせない
- カーネルに関わる操作はできない
  - コンテナ毎にロードするモジュールを変えるなど

# Linux におけるコンテナの実装

# Linux におけるコンテナ実装

カーネルの機能 (+ パッチ) + カーネルの機能を使う userspace ツール

- カーネル + パッチ + userspace ツール
  - OpenVZ / Virtuozzo(商用)
  - Linux VServer
- カーネル + userspace ツール
  - LXC
  - libvirt (lxc ドライバ)
  - systemd(systemd-nspawn)
  - vzctl for upstream kernel

# LXC

- LXC (http://lxc.sourceforge.net/)
  - Linux のコンテナを操作する userspace ツール (コマンド群)
  - 現在の stable は 0.9.0
  - 『Ubuntu』のコンテナツールキットの性格が強い
- libvirt (のLXCコンテナドライバ) (http://libvirt.org/)
  - LXC 側で『オレたちの API で libvirt のコンテナドライバを書き直そう』みたいな話はある
  - 現時点でまだ動きは微妙...

- どちらも同じ "LXC" という名前を使っているが，設定ファイルは別々．カーネルの同じ機能を使った別の実装.
- このセッションでは，"LXC" は前者を指して使います．後者は "libvirt" とします.

# Linux コンテナことはじめ

# Ubuntu でとりあえずコンテナを起動してみる

- 現在の LXC の事実上の開発プラットフォーム
- lxc デベロッパー = Ubuntu デベロッパー
- おそらく Ubuntu が主要な開発のベース．それ以外は後追い．

```
# apt-get install lxc
# lxc-create -n ct01 -t ubuntu
# lxc-start -n ct01 -d
# lxc-console -n ct01
```

- 12.04LTS を使ってる場合 Ubuntu 最新版の kernel のバックポートである linux-current-generic を入れると幸せかも?
  - 入れなくてもちゃんと動きます
  - 3.8 kernel が入るので，色々な面で便利になる可能性があります (3.8 kernel については後述)

```
# apt-get install linux-current-generic
```

# コンテナのプロセスの様子

親環境上でコンテナのプロセスを見ると...

```
# pstree -p
init(1)─┬─acpid(1041)
        :(snip)
        ├─lxc-start(15592)───init(15597)─┬─cron(15877)
        │                                ├─dhclient3(15817)
        │                                ├─getty(15867)
        │                                ├─getty(15871)
        :(snip)
```

コンテナ内で見ると

```
init(1)-+-cron(252)
        |-dhclient3(192)
        |-getty(242)
        |-getty(246)
       :(snip)
```

# アプリケーションコンテナ

コンテナ内で /sbin/init を起動しなくても，目的のプログラムのみ起動可能 (この例は無理矢理感ありますw).

```
# lxc-start -n ct01 -- /usr/sbin/apache2 -DFOREGROUND
```

```
# pstree -p 15535
bash(15535)─┬─lxc-start(19577)
            └─lxc-start(19708)───apache2(19714)─┬─apache2(19740)─┬─{apache2}(19771)
                                                │                ├─{apache2}(19772)
                                                │                ├─{apache2}(19773)
                                                :(snip)
```

- docker は 0.6 より前はアプリケーションコンテナ専用でした (システムの /sbin/init は実行できなかった).

# CentOS 6 で動かしたいんですけど...

- パッケージはなし
- 最新でなく，0.8.0 or 0.7.5 辺りをソースからコンパイルして入れましょう.
  - CentOS 6 の kernel は 2.6.32 で，コンテナ的に見ると太古の昔．廃止された機能とか，未実装の機能とか...
    - ns(namespace) cgroup (3.0 で廃止)
  - なので昔の時点でのバージョンの方がきちんとテストされていて動きます. 一応最新 (0.9.0) でも動くようには実装されていますが...
  - 実装が不十分な net_prio, perf_event cgroup はマウントしない
- 標準にはテンプレート (lxc-centos) が含まれない．github/gist を探せばあります → lxc-centos
- RedHat は libvirt 路線っぽい...
  - RHEL7 はコンテナサポートと言われてますので，libvirt できちんと動くように調整して出荷される?

# Linux におけるコンテナの仕組み

# Linux でコンテナを実現するための機能

- プロセスをグループ化して他のグループと隔離
  - → Namespace (名前空間)
- グループ化したプロセスに対するリソース制限
  - → Cgroups (control groups)
- その他
  - ネットワーク (veth, macvlan)
  - ケーパビリティ
  - Checkpoint/Restore (CRIU)
  - などなど…

# Namespace の種類 (1)

- Mount Namespace: 2.4.19
  - プロセスから見えているマウントの集合，操作を分離する．Namespace 内の mount, umount は他の Namespace には影響しない
  - (参考) マウント名前空間を適用する(IBM developerWorks)
- UTS Namespace: 2.6.19
  - ホスト名など，uname(2) が返す値の集合を分離．setdomainname(2), sethostname(2) で Namespace 内の値のみ変更できる
- PID Namespace: 2.6.24
  - PID 空間の分離．新しい PID Namespaceでは PID 1 から始まる PID が割り当てられる．親から子の PID Namespace は見える (親の空間の PID を持つ) が，子から親は見えない

# Namespace の種類 (2)

- IPC Namespace: 2.6.19
  - SysV IPC オブジェクト，POSIX メッセージキューの隔離
- User Namespace: 2.6.23 ~ 3.8
  - 独立した UID/GID 空間と外部空間のマッピング (例えば，隔離空間では uid/gid 0/0，外部では 1000/1000 とか可能になる)
- Network Namespace: 2.6.26
  - ネットワークリソースの隔離．ネットワークデバイス，アドレス，ルーティングテーブル，ソケット，フィルタリング

# Namespace の操作

- clone(2) で新しいプロセス を生成
- unshare(2) で新しいプロ セスを生成せずに実行コンテキストを制御する
  - unshareの使用例
- setns(2) でプロセスを既存 のNamespaceに関連付ける

# Cgroup (1)

プロセスをグループ化し，グループに対してリソース制限を行う．別にコンテナ専用の仕組みではない．

- cpu
  - CFS(Completely Fair Scheduler) bandwidth control．単位時間内のグループ内のタスクが実行できる合計時間を制限する (3.2 で実装)
    - (参考) Linux 3.2 の CFS bandwidth control
  - 相対配分．グループ間の CPU 時間の割当の割合を指定する．例えば GroupA=100, GroupB=50 とすると A:B = 2:1
- cpuacct
  - グループ内の CPU リソースのレポート (CPU 時間)
- cpuset
  - 割り当てる CPU, メモリノードの割当

# Cgroup (2)

- device
  - デバイスへのアクセス許可，制限の指定
- freezer
  - グループ内のプロセスを全て一時停止する
- memory
  - メモリリソースの制限 (ユーザメモリ，カーネルメモリ)
- blkio (Block IO)
  - I/O weight controller (2.6.33 以降) グループの優先度を指定する
  - I/O throttling (2.6.37 以降) グループ内のプロセスのデバイスに対する bytes/second の合計の指定
    - (参考) Linux 2.6.37 の新機能 "I/O throttling"

# Cgroup (3)

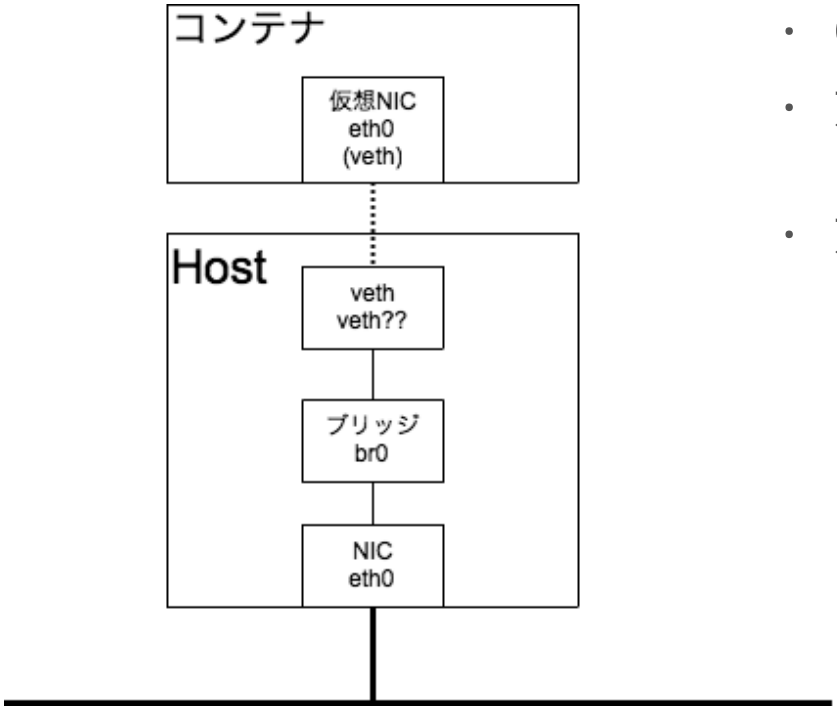
- hugetlb
  - hugetlb に対する制限 (3.6 以降)
  - mm/hugetlb: add new HugeTLB cgroup
- perf_event
  - グループ単位で perf ツールでモニタリング (パフォーマンス解析)
- net_cls
  - パケットに識別子をつけ，トラフィックコントロール (tc) でコントロール可能に
- net_prio
  - グループ間でのネットワークの優先度をインターフェース毎に指定する
  - Linux 3.3 の新機能 Network priority cgroup
  - Linux 3.3 の新機能 Network priority cgroup (2)

# Cgroup (4)

Cgroup はコンテナと関係なく使用可能

```
# mount -t tmpfs cgroup_root /sys/fs/cgroup
# mkdir /sys/fs/cgroup/memory
# mount -t cgroup -o memory cgroup /sys/fs/cgroup/memory (メモリサブシステムのマウント)
# mkdir /sys/fs/cgroup/memory/test01 ("test01" というグループの作成)
# echo $$ > /sys/fs/cgroup/memory/test01/tasks (プロセスをグループに登録)
# cat /sys/fs/cgroup/memory/test01/tasks (グループ内のプロセスの確認)
2824
2837
# echo 30M > /sys/fs/cgroup/memory/test01/memory.limit_in_bytes
(グループに対してメモリ上限 30M という制限を設定)
# cat /sys/fs/cgroup/memory/test01/memory.limit_in_bytes (制限値の確認)
31457280
# cat /sys/fs/cgroup/memory/test01/memory.usage_in_bytes (現在の使用量の確認)
565248
```

# LXC で使うネットワーク機能 ～ veth




- OpenVZ/Virtuozzo 由来の機能
- 対となるインターフェースを生成し，インターフェース間で通信を行う (Layer2 のトンネル)
- 対の片方をホスト側のブリッジに，片方をコンテナに接続 (コンテナ側のインターフェースはコンテナの Network Namespace に属するように設定されるのでホストからは見えない)

# LXC で使うネットワーク機能 ～ macvlan

- 物理インターフェースに別の MAC アドレスが付いた新しいインターフェースを作成．このインターフェースをコンテナに割当
  - 物理インターフェースを promiscuous モードにして該当 MAC アドレスのパケットを受け取る
  - モードの設定が存在: private, vepa, bridge
- 物理インターフェースをそのまま使うのに近いので負荷が低く，パフォーマンスが良い傾向
- (参考) macvlanを使ってみる (驟雨のカーネル探検隊（只今遭難中ｗ)
- lxc の仮想ネットワークのパフォーマンス測定

# Linux Kernel 最新動向

# Linux 3.8

LXC 的な視点からすると，コンテナ実現に必要な主要機能が揃ったバージョン.

# Linux 3.8 の新機能 - User Namespace

- User Namespace
  - LXC のセキュリティは? という FAQ に対する回答
  - ただし，3.8 でとりあえず完成したものの，対応する実装がなされていない filesystem が多く，事実上使えない状態．3.9 で XFS を除いて実装完了
  - /proc/PID/uid_map, /proc/PID/gid_map ファイルにマッピング情報を書く

```
0     100000     10000     (Namespace内のIDの開始 Namespace外のIDの開始 範囲)
```

- Linux 3.8 の User Namespace 機能(1), (2), (3), (4)

# Linux 3.8 の新機能 - setns(2) が実用的に (1)

- 実は User Namespace 以上に重要な新機能
- Namespace を指し示す特殊なファイルが /proc/PID/ns 以下に存在する．3.8 より前はこのファイルが ipc, net, uts のみしか存在しなかった

```
-r-------- 1 root root 0 Mar  1 15:41 ipc
-r-------- 1 root root 0 Mar  1 15:41 net
-r-------- 1 root root 0 Mar  1 15:41 uts
```

- プロセスを既存の Namespace に入れるには setns にこの特殊なファイルのファイルディスクリプタを与える必要がある
  - つまり 3.8 より前は Namespace 外部から Namespace 内でコマンドを実行する事ができないケースがほとんどだった

# Linux 3.8 の新機能 - setns(2) が実用的に (2)

- 3.8 で全ての Namespace に対する特殊なリンクが存在するようになった

```
lrwxrwxrwx 1 root root 0  3月  1日  14:59 ipc -> ipc:[4026532301]
lrwxrwxrwx 1 root root 0  3月  1日  15:06 mnt -> mnt:[4026532299]
lrwxrwxrwx 1 root root 0  3月  1日  15:06 net -> net:[4026532304]
lrwxrwxrwx 1 root root 0  3月  1日  15:06 pid -> pid:[4026532302]
lrwxrwxrwx 1 root root 0  3月  1日  15:06 uts -> uts:[4026532300]
```

- 3.8 でどの Namespace に対しても setns が実行できるようになり，コンテナ外部から内部のコマンドを実行可能になった.
- 同じ Namespace に属している場合，リンクは同じ inode を指す (stat() で簡単に調べられる)
- Linux 3.8 で改良された Namespace 機能と lxc-attach コマンド

# Linux 3.8 の新機能 - Memory Cgroup の Kernel Memory サポート

- TCP Buffer に対する制限は 3.3 で実装 (他の Memory controller の実装とはかなり違う)
  - Linux 3.3 の新機能 Per-cgroup TCP buffer limits, (2), (3)
- スタックとスラブのアカウンティングをサポート
- Linux 3.8 で改良された memory cgroup (2)

# Linux 3.9 の新機能

- User Namespace 機能の様々な部分への実装が進む
  - userns: Allow the unprivileged users to mount mqueue fs
  - userns: Allow the userns root to mount of devpts
  - userns: Allow the userns root to mount ramfs.
  - userns: Allow the userns root to mount tmpfs.
  - userns: Allow unprivileged reboot
  - その他，xfs を除く各種ファイルシステムなど
- blkio の I/O weight controller の完全階層構造サポート

# Linux 3.10 の新機能

- perf_event cgroup の階層構造サポート
- device cgroup のアクセス権変更時の子孫方向への伝搬
- sane_behavior オプション
- memory cgroup の memory pressure レベルの通知サポート

# Linux 3.10 の新機能 memory cgroup の memory pressure レベルの通知

- 従来からある eventfd を使った通知と同様の方法でメモリの Pressure レベルの通知が受けられるようになった.
  - 従来は, メモリのしきい値を設定して値をまたいだ時, OOM Killer が発動した時の通知
- "low", "medium", "critical" の 3 レベル
- Linux 3.10 で memory cgroup に追加された Memory Pressure 通知機能

# Linux 3.11 の新機能

- blkio I/O throttling の完全階層構造サポート
  - ただし，"sane_behavior" オプションを付けたときのみ

# cgroup 再設計 〜 現在の cgroup の問題点

- 今の cgroup はまともじゃない!
  - 標準的なカーネルの API からかなり逸脱している
  - ファイルシステムだからアクセス権さえあれば誰でもカーネルを制御できる
  - サブシステム毎に少しずつ動きが違ったり
  - 色々な所にマウントできたり
  - 不必要に複雑
- (参考) Linux カーネルのすべて: cgroup の再設計 (原文)
- (参考) Changes coming for systemd and control groups

# cgroup 再設計

- でも今の cgroup と互換性を保ったまま全部まともにするのは不可能
- じゃあ，とりあえず出来そうな所をやろう
  - 単一階層構造
    - cgroup は systemd (や他のソフト) が作成して管理するようにして，誰でも cgroup を直接いじることができないようにしよう!
  - sane_behavior オプション
    - サブシステム独特の変な機能を止めて一貫性のある動きを強制する
    - cgroupfs のマウントオプション noprefix と clone_children が許可されなくなる
    - remount できない など
    - (参考) cgroup: introduce sane_behavior mount option

# Kernel のコンテナ関係機能の今後 ~ その他

- XFS の User Namespace 対応 (3.12)
- quota
  - 定期的に湧いて出てくる話題
  - container disk quota (2012年5月)
  - その後どうなったのか追っていません
- Device Namespace
  - DeviceNamespace (Cellrox)
    - Android などのモバイルデバイスで複数の環境を切り替えながら使うのを目的に開発されているようだ
    - Device Namespaces (lxc-devel, containers ML) 辺りから議論が盛り上がっている

# Kernel のコンテナ関係機能の今後 ~ その他

- Syslog Namespace
  - Stepping closer to practical containers: "syslog" namespaces (lwn.net)
  - Add namespace support for syslog
- /proc の memory などの統計値
- checkpoint/restore 関係機能の機能追加が頻繁に行われている

# LXC 最新動向

# LXC の現状

- Ubuntu 12.04LTS
  - 0.7.5 だが，中身はほぼ 0.8.0
  - セキュリティ的にヤバい所は AppArmor で抑えこんでひとまず最低限使える状態に
  - 自分でコンテナ作成，起動して自分で軽く使うには十分
- 最新版 0.9.0
  - API 公開(liblxc)，python3, lua バインディング
  - seccomp サポート
  - コンテナ起動，停止時の各段階でのフックが可能に

# LXC 開発の現状

- 開発体制の変更
  - メインのメンテナが，Daniel Lezcano 氏多忙のため，Serge Hallyn, Stéphane Graber 両氏 (Ubuntu) に変更
  - リポジトリも sourceforge から github へ
- 1.0 に向けて絶賛仕上げ中
  - 現在 1.0 alpha1 (Linux Plumbers に向けてとりあえずまとめた感)
    - alpha1 のタグが打たれた後も，まだまだ新機能が追加されまくっているので，1.0 になる時にどうなるかは読めてません (^_^;)
  - 10月: alpha2，11月: alpha3，12月: rc1，1月: rc2，2月 1.0final
  - Ubuntu の次の LTS がターゲット．(= 1.0 は 5 年間サポートされる)

# lxc-1.0 (1)

目新しい機能というよりは，1.0 としてふさわしくなるように諸々整備をしている感じ.

- API の整備 (stable な API へ) と各コマンドを API ベースで書き直し
- クローン，スナップショット機能の新しい実装
  - announcing lxc-snapshot (S3hh's Blog)
- console の扱いの改良
- テンプレートの改良，バグフィックス
- 非特権コンテナに向けての最初の一歩 (User Namespace)
  - (参考) Creating and using containers – without privilege (S3hh's Blog) ... 試してみたもののうまくいかなかった (;_;)
  - lxc-user-nic (veth ペア作るのにホスト側では特権必要)
  - lxc-usernsexec (定義された uid/gid マッピングを使ってユーザ Namespace でプログラムを起動)
  - shadow も同時に上げる必要がある (newuidmap, newgidmap コマンド)

# lxc-1.0 (2)

- ストレージバックエンド
  - 再設計，整備
  - overlayfs 対応，overlayfs を使った clone
  - zfs 対応 (create, clone, snapshot)
- モニタリングの改良
  - 今まで同時に複数のコンテナのモニタリングとかできなかった
- Android NDK でのビルド
- 日本語 man (!!)

# 最後に

# メーリングリスト / 翻訳

- lxc JP グループ
  - コンテナの話をまったりやっています．たまーにしかメールは来ません．lxc-jp という名前ですが，話題は LXC に限らず何でも OK です．
- lxc man pages 翻訳
  - https://github.com/tenforward/lxc-doc-ja でやってましたが，lxc 本家にマージされたのでどう翻訳をすすめていくか考えてます．協力頂ける方がいらっしゃればご連絡いただければと思います．翻訳作業自体は別の所でやってからある程度まとめて本家に Pull Request した方が良いかと思っています．

# 今後聞いてみたい話

- RHEL7 / libvirt のコンテナ実装
- docker 方面の諸々
- 活用事例
- コンテナ関連実装のもっと深い話
- Apache Mesos
- CRIU

# <Thank You!>

Important contact information goes here.

twitter @ten_forward
www www.ten-forward.ws/
github github.com/tenforward