2011年7月15日
オープンソースカンファレンス 2011　Kansai@Kyoto

Linuxの標準機能「LXC」を用いた仮想化 - 入門編 -

株式会社Joe'sウェブホスティング
山本 政秀

株式会社Joe'sウェブホスティング 概要

- 設立　2002年7月　資本金　1000万円
- 大阪本社　Joe’s ITサロン 梅田, Joe’sビジネスセンター 梅田
- 〒530-0001 大阪府大阪市北区梅田1丁目11番4-923号 大阪駅前第4ビル9階
- 銀座オフィス Joe’s ITサロン 銀座, Joe’sビジネスセンター 銀座
- 〒104-0061 東京都中央区銀座1-3-3　G1 ビル　7階
- 南青山オフィス Joe’sビジネスセンター 青山
- 〒107-0062 東京都港区南青山2-11-13 南青山ビル4階
- 役員　代表取締役 CEO　鈴木禎子、取締役 CTO　山本政秀
- 　　取締役 海外部　鈴木拓人、執行役員　緒方俊輔
- 従業員数　17名
- 事業内容
  - クラウド/ウェブホスティング事業
  - 情報セキュリティ事業
  - 経営支援事業

講師自己紹介

- 山本 政秀（やまもと まさひで）
- 出身地： 兵庫県姫路市
- 生年月日：昭和51年12月9日 34歳
- 2003年5月　入社
- 2007年1月　Joe'sウェブホスティング取締役CTO
- 弊社LXC-VPSの開発を担当

- 写真は米国出張の際cPanelのCEOと撮影

事業内容紹介

- ウェブホスティング事業
  - フルマネージド専用サーバ、root権限付き専用サーバ、VPS、共用サーバ
  - ハウジングサービス、HAクラスタサービス、ドメイン取得
  - コントロールパネル (cPanel/Plesk)
  - Cpanelを日本で最初に提供を開始した会社
    cPanelと言えばJoe's

事業内容紹介

- 情報セキュリティ事業
  - SSL証明書は主要ブランドを網羅、国内最安値で提供
  - ベリサイン、ジオトラスト、グローバルサイン、サイバートラスト、アルファSSL、セコム、コモド
  - 独自の仕入れルート　ベリサインなら39,900円～（他社は倍以上がほとんど）
  - 創業から、証明書の発行・設置業務を行っている。

事業内容紹介

- 経営支援事業
  - Joe'sビジネスセンター－バーチャルオフィス
  - 東京は銀座・青山、大阪は梅田の一等地を登記住所として利用可能
  - 東京駅、大阪駅から徒歩10分以内の距離
  - 会議室貸出し、電話転送、郵便物転送、東京/大阪間テレビ会議

Contact us

- URL
  - joeswebhosting.net（ウェブホスティング/レンタルサーバ）
  - jwh.jp（会社ブログ）、joes-ssl.com（SSL証明書）
  - joes-office.com（バーチャルオフィス）
  - ec-cube.org（EC-CUBE標準サーバ）
  - cpanel-plesk.net（コントロールパネル/VPS）
- Twitter/Facebook
  - @JoesWebHosting、@Joes_office
  - http://www.facebook.com/JoesWebHosting
  - http://www.facebook.com/JoesOffice

本日のアジェンダ

- LXCのご紹介
  - LXCの概要
  - 仮想化のおさらい
  - LXCの詳細
  - 他方式との比較
- LXCコンテナを設定するデモ

LXCの概要

- LinuXContainer
- OSレベルの仮想化
- Namespace+cgroup+ユーザランド
- IBM Daniel Lezcano氏(仏)、
  他数名でプロジェクトを創始（2008年8月頃）
- 弊社も開発に参加している

仮想化とは何か

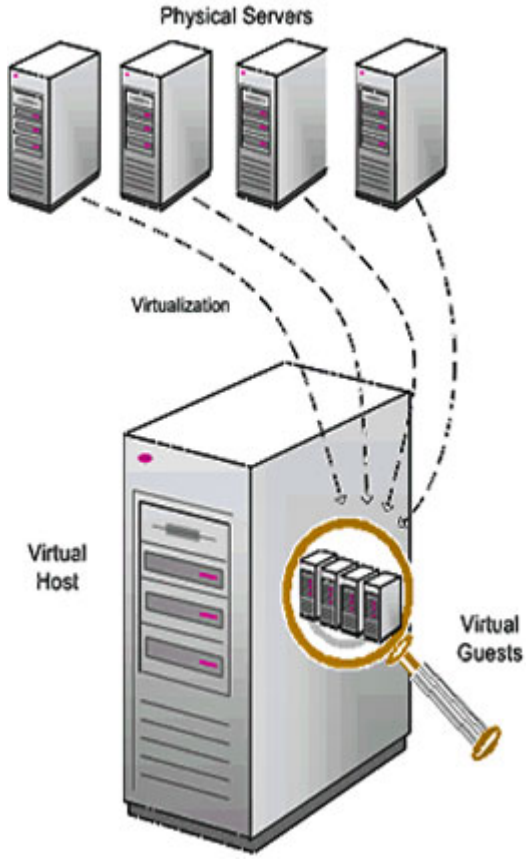
- 仮想化（Virtualization）とはリソース（資源）の抽象化と隠蔽のこと
- 1960年代から既に知られていた
- 仮想化の例
  - VPS
  - 仮想メモリ





仮想化とは何か - 仮想化の例

- VPS



- 利用者からは複数の仮想サーバが存在していると言った事実が見えない　⇒「技術的詳細が隠蔽されている」

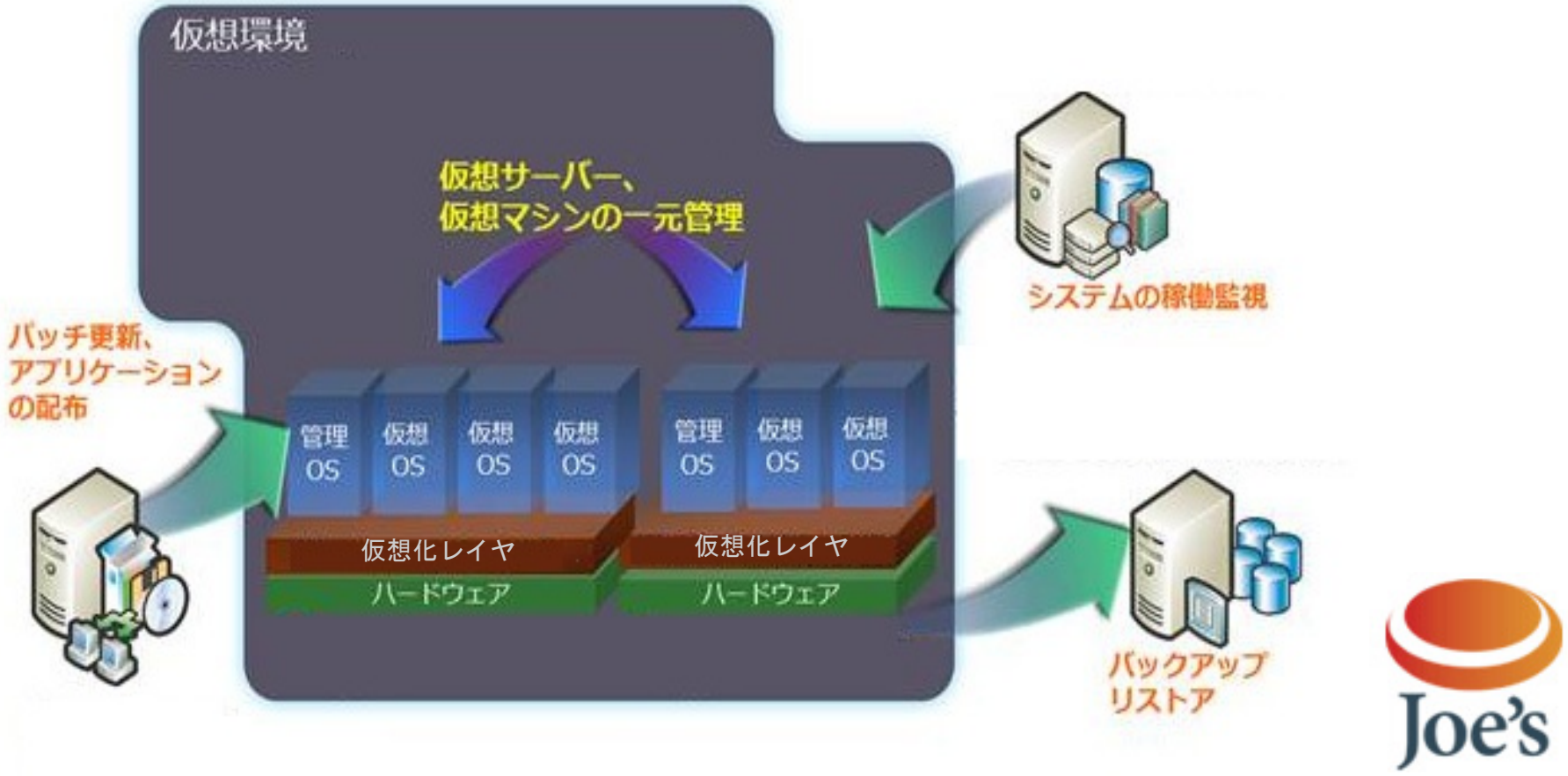
仮想化とは何か - 仮想化の例

- 仮想メモリ
  - メモリの抽象化
  - アプリケーションの開発者は、連続したメモリを単純に利用できる（詳細はOSが行う⇒隠蔽＝知らぬが仏）

```
[root@lxcbase5 ~]# free
             total       used       free     shared    buffers     cached
Mem:      12300272   12175504     124768          0     806780    4571272
-/+ buffers/cache:    6797452    5502820
Swap:      4194168      14400    4179768
```

仮想化とは何か - 仮想化の利点

- 集約による効率の向上とコスト削減



仮想化 - 主な方式

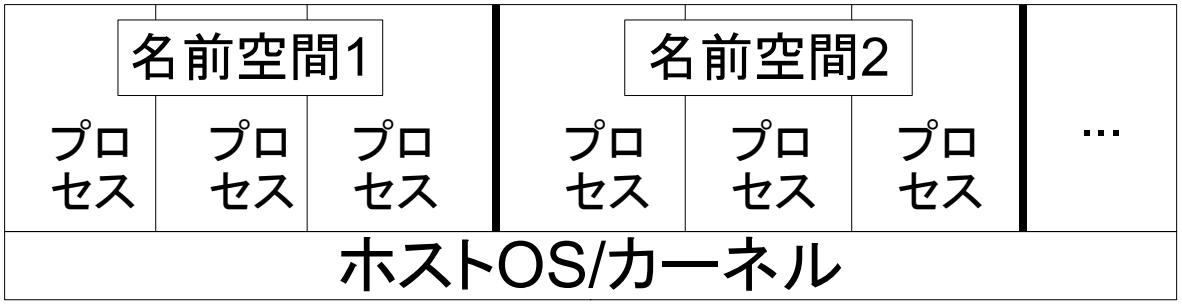
完全仮想	ハードウェアをエミュレート オーバーヘッドが大 **(CPU** 支援で、若干改善**)** 任意の**OS** が利用可	**Bochs, QEMU, KVM**
準仮想	ゲスト**OS** がホスト**OS** の **API** を利用 **(**ハイパーコール**)** オーバーヘッドは中	**Xen, ESX/ESXi, Hyper-V**
OS仮想化(コンテナ)	異なる仮想サーバーを、プロセスの権限で制御 **(OS** 仮想化**)** オーバーヘッドが小 カーネルは固定、**OS** は選べない	**OpenVZ, Virtuozzo, LXC**

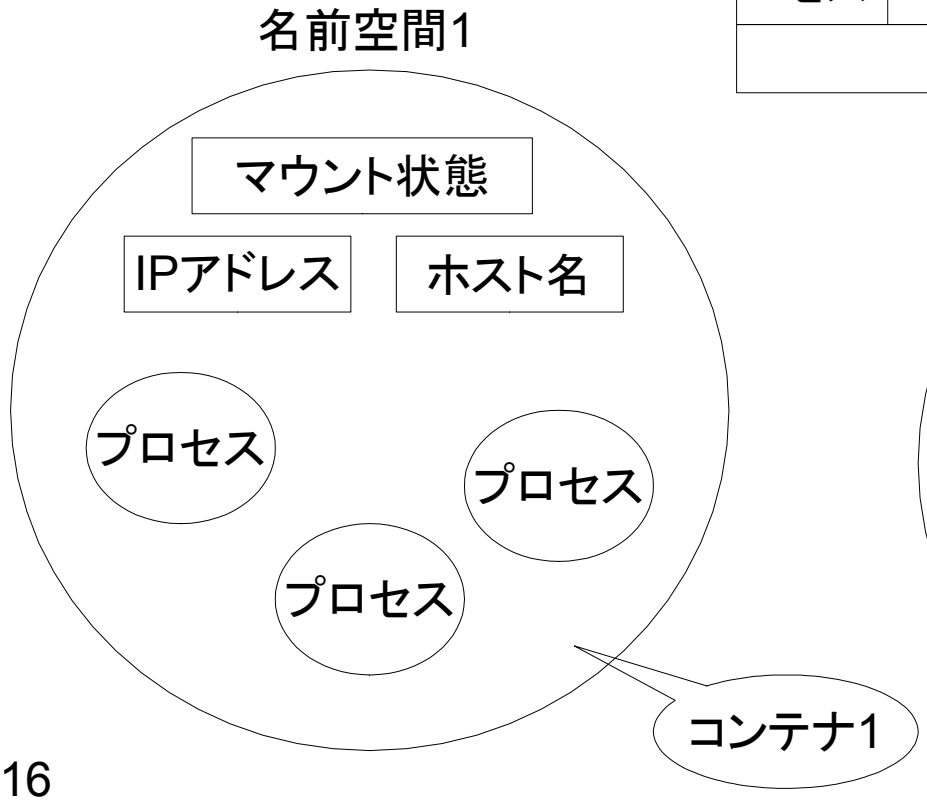
LXCの詳細

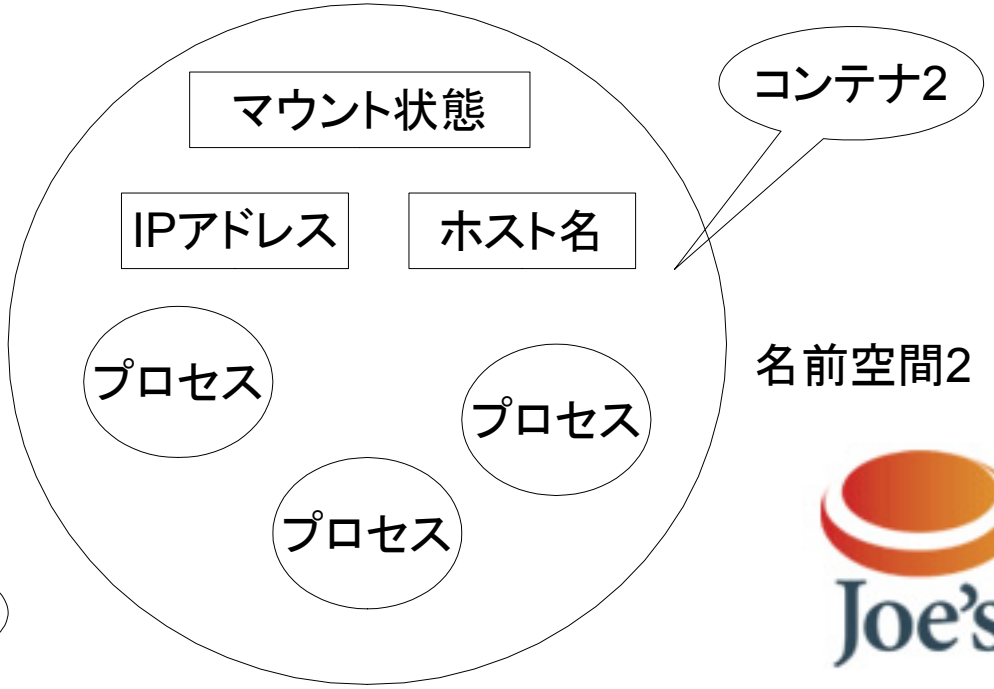
- LXC=Namespace+cgroup+ユーザランドツール
- Namespace（名前空間）
  - 対象となる名前、識別子の集合を他と分離する事を可能とする概念
  - 別の名前空間には干渉できない
- cgroup（ControlGroup）
  - Namespaceと協調して機能し、グループ化された対象に対してリソース制御をする為のもの
  - メモリ、CPU負荷、ブロックI/Oの量を管理/制限

LXCの詳細 - Namespace

- 別の名前空間には干渉できない

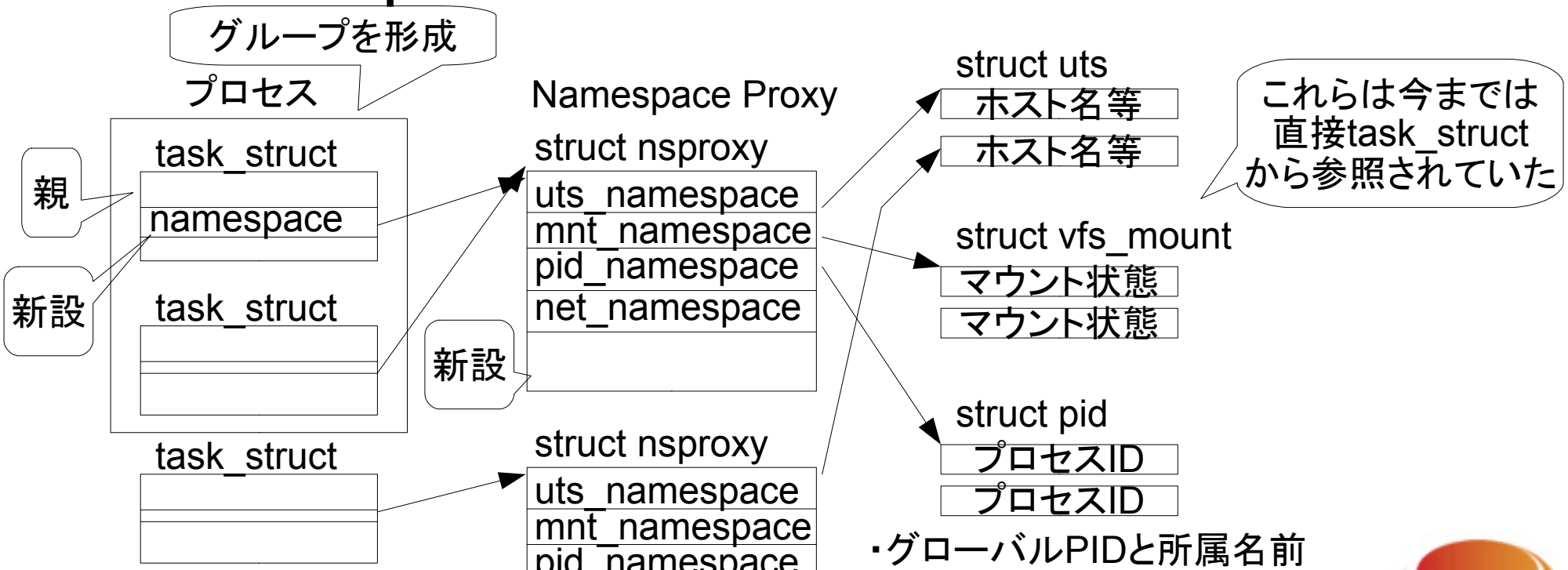






LXCの詳細 – Namespace(続き)

- Namespace（名前空間） カーネル内部では

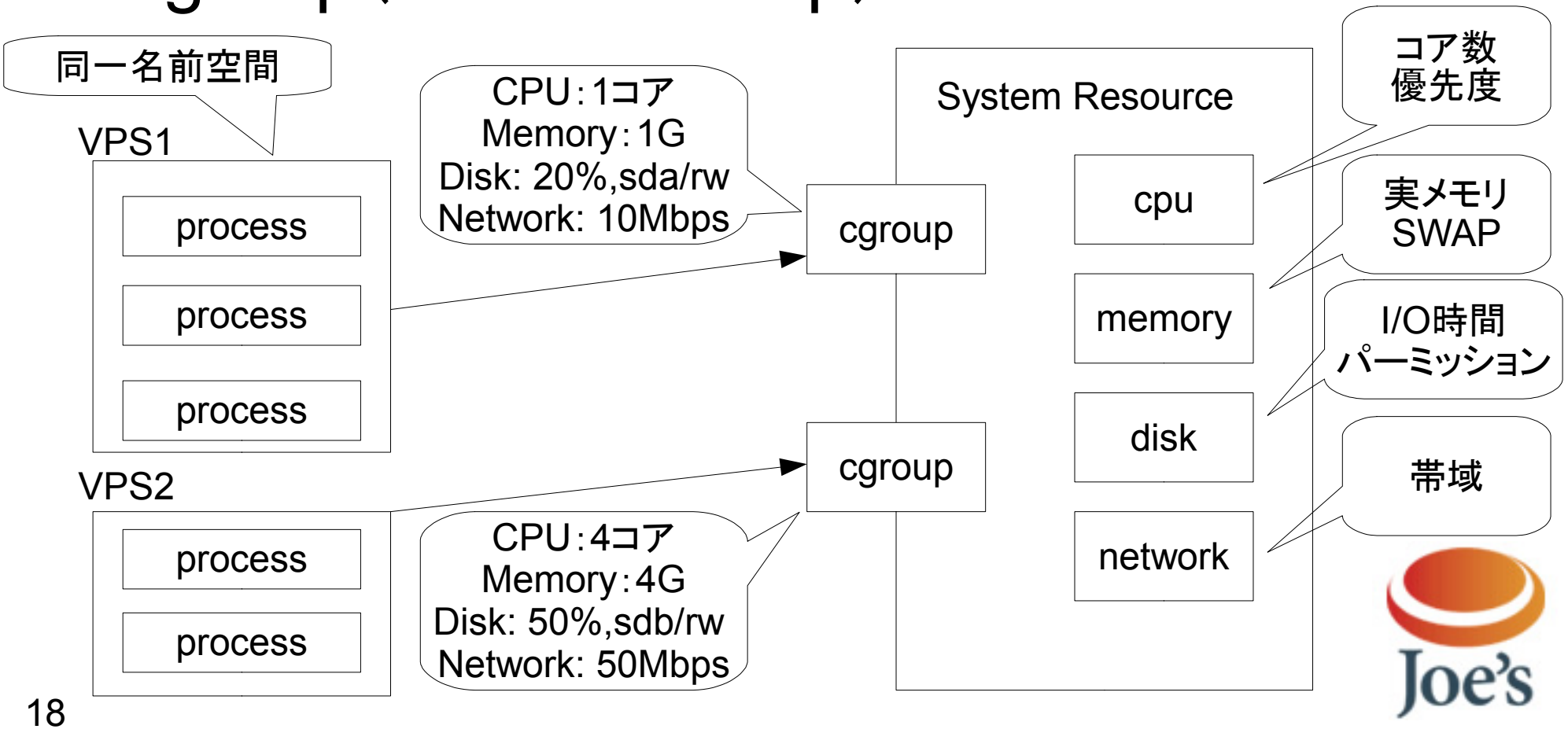


・一部簡略化
・子は親と同じ名前空間
・同じ名前空間であれば
干渉できる

・グローバルPIDと所属名前空間内でしか通用しないPID
・/proc（ps等が参照）は名前空間毎に独自PIDでプロセス情報をエクスポート

LXCの詳細 - cgroup

- cgroup（ControlGroup）



LXCの詳細 - ユーザランド

- ユーザランドツール
  - lxc-create：VEの作成
  - lxc-destroy：VEの削除
  - lxc-start：VEの開始
  - lxc-stop：VEの停止
  - lxc-ps：VE内のプロセスの表示（ps）

    ... etc

    ※ デモで実演

LXCの詳細 - 利点

- Linux Kernelの標準
- 最新ディストリに採用
  - RHEL6
  - Ubuntu OpenVZ⇒LXC
- 高い性能

LXCの詳細 - Linuxの標準

- Kernelは頻繁に再設計される
  - OpenVZは非標準
    - ⇒追いつけない
    - ⇒古いKernelを引きずる
    - ⇒性能改善の恩恵を逃す
  - KVM/LXCは標準
    - ⇒自動追従

Joe's

LXCの詳細 - OpenVZとの比較

- 同じコンテナ型であるOpenVZとの比較
  - *基本は同じ*
  - VirtuozzoはOpenVZがベース
  - LXCの方が**20%以上性能が高い**(弊社調べ)

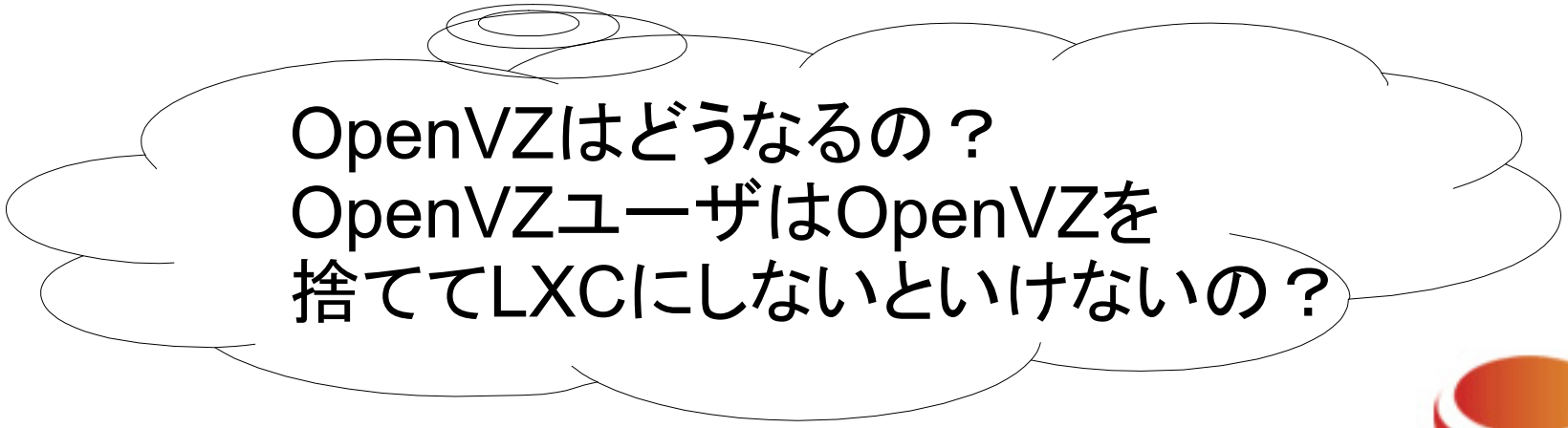
    UnixBench結果： LXC:**5101.8**

    OpenVZ：4028.1 ※

    ※ 7月6日時点の最新のVirtuozzo安定版Kernelで実施

LXCの詳細 - OpenVZとの比較

- LXCとOpenVZの将来性についての考察
  - UbuntuがOpenVZからLXCに切り替え
  - OpenVZ利用者と開発者のとあるやり取り
    - ユーザの質問内容:



※ 出典: http://openvz.livejournal.com/30998.html?thread=96790

LXCの詳細 - OpenVZとの比較

- **OpenVZプロジェクトマネージャ Kir Kolyshkin氏（露）の回答**

- **LXCは大部分IBMのプロジェクトでありOpenVZのサブプロジェクトではない**
- LXCのカーネル側サブシステムであるnamespaceとcgroupの**一部はOpenVZの開発者によっても開発**されている。
- その成果でOpenVZの**既存のコードと入れ替えている**。その際**既存のコードは捨てられている**。
- これによりOpenVZの**パッチセットは小さくなっている**。
- **LXCはまだOpenVZより機能が貧弱**
- これらの活動は**OpenVZをカーネルのメインストリームに取り込んで貰えるようにするための活動の一端**
- 今はまだOpenVZから乗り換える必要は無い

LXCの詳細 - OpenVZとの比較

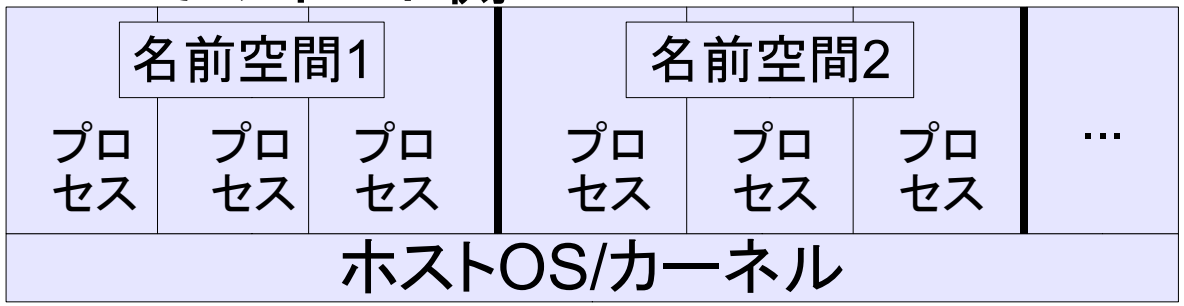
- LXCとOpenVZの将来性についての考察
  - 現段階ではLXCはまだ発展途上
  - OpenVZよりも手軽なので個人で使う分には問題なし
  - サービスとして使う場合現状一部手を入れる必要あり
  - 最終的に、OpenVZのコアとLXCのコアは共通化
  - 総合的に見て**LXC**は将来性あり

LXCの詳細 - OpenVZとの比較

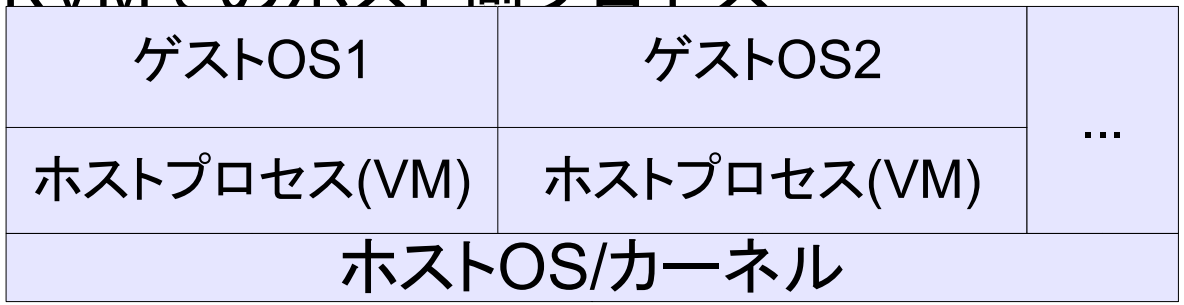
- まとめ
  - OpenVZ:　枯れている、安定している、Kernelの再コンパイルが必要、特殊なハードウェアを必要としない、LXCより20%以上性能が低い
  - LXC:　新しい、軽量、安定している、Linux標準、Kernelの再コンパイルや特殊なハードウェアを必要としない、セキュリティ面で配慮が必要でサービス化には独自の手入れやノウハウが必要

LXCの詳細 - KVMとの比較

- 別方式であるKVMとの比較
  - LXCでのホスト側プロセス



  - KVMでのホスト側プロセス



プロセスの状態から見た比較

LXCの詳細 - KVMとの比較

- 性能比較
  - KVMは**Linux標準**の優れた完全仮想化方式
  - LXCの方が**33%以上性能が高い**(弊社調べ)

    UnixBench結果：　LXC:**5101.8**

    KVM：3411.2　※

※ KVMにおいては、ハードウェアによる支援（Intel-VT、VT-d）の有効化、ゲストカーネルのKVM最適化（カーネルビルドオプションで最適化を指定）、virtioを用いてI/O処理において極力オーバヘッドを低減して計測

同ハードウェア（CPU数,メモリ）、同コマンドでの計測

LXCの詳細 - KVMとの比較

- リソース制御
  - サーバは*通常稼働率が低い⇒遊んでいる*
  - KVMは、動的なリソース制御の自由度が低い
    - *ゲストOS単位*でのリソースの割り当て
    - リソースが余っていても必要とする所で使えない
  - コンテナ型はOSのネイティブプロセス
    - *プロセスグループまたはプロセス単位*での割り当て
    - 余っているリソースを効果的に使える

LXCの詳細 - KVMとの比較

- まとめ
  - KVM： 任意のゲストOS、安定している、強い分離、Linux標準、Kernelの再コンパイルは不要、性能向上のためには対応CPUが必要（Intel-VT等）
  - LXC： VMではなくVEという考え方、より柔軟、より効率的、KVMよりも33%以上性能が高い
  - KVMとLXCではカバー範囲が異なり共存が可能

Joe'sの取り組み

- LXC + cPanel
  - 軽量VPS
  - 「使いやすく多機能なcPanelを安く」
    for EC-CUBEユーザ 否 for All
  - Joe's クラウド (VPSの素)構想

ご清聴ありがとうございました。

■弊社LXC + cPanel利用サービス
http://cpanel-plesk.net/lxc-series/

追加のご質問がございましたら、弊社（協賛企業）
ブースまでお気軽にお立ち寄りください！

付録 - UnixBench

- 比較表

	KVM(Guest)	Virtuozzo(Guest)	LXC(Guest)	LXC(Guest)
Plan	-	PVZ1	LXC1(32bit)	LXC1
Kernel(Host)	2.6.32.41	2.6.18-028stab091.2	2.6.32.41	2.6.32.41
Kernel(Guest)	2.6.39	-	-	-
Arch(Host)	x86_64	x86_64	x86_64	x86_64
Arch(Guest)	x86_64	x86_64	x86(32bit)	x86_64
GLIBC(Guest)	glibc-2.12-1.7.el6_0.5	glibc-2.5-49.el5_5.7	glibc-2.5-49.el5_5.7	glibc-2.5-49.el5_5.7
Dhrystone 2 using register variables	9237.4	5006.5	4379.1	9348.9
Double-Precision Whetstone	3639	3544.8	2854	3786.9
Execl Throughput	2948.7	5156.4	5469	5812.7
File Copy 1024 bufsize 2000 maxblocks	2162	1083.5	4379.5	3120
File Copy 256 bufsize 500 maxblocks	838.2	899	1461.6	1784.8
File Copy 4096 bufsize 8000 maxblocks	3383.4	1984.7	7870.5	9707.2
Pipe Throughput	4939.3	7108.1	5617	6059.5
Pipe-based Context Switching	2207	5439.7	3856.8	4919.2
Process Creation	3462	4627	5177.5	5715.2
Shell Scripts (1 concurrent)	4882.9	8062.4	5848.6	5454.7
Shell Scripts (8 concurrent)	4321.7	8041.2	5817.9	5341.1
System Call Overhead	5041.3	8892.6	5367.9	5632
System Benchmarks Index Score	**3411.2**	**4028.1**	**4694.3**	**5101.8**

付録 - UnixBench

- 実施条件

※ 上記はUnix Bench 5.1.3 において ./Run を引数無しで実行した結果である。
※ VirtuozzoはPararrelsが提供している2011/7/6 時点で利用可能な最新のカーネルを用いた。
※ LXC, Virtuozzo共にCPU、メモリ、I/Oにおいてnolimitにて試行
※ KVMはKVMゲスト最適化を有効とし、Block Deviceはvirtioを用いて極力オーバヘッドの低減に努めた。(ゲストからは/dev/vdaとして見える)
KVM(qemu)起動コマンドは以下の通り。

```
# qemu-system-x86_64 -drive file=/mnt/kvm/kvm1/kvmimage,if=virtio,boot=on -m 2048 -vnc :1
        -net nic,model=virtio,vlan=1,macaddr=00:00:00:00:00:01
        -net tap,vlan=1,ifname=tap1,script=/etc/qemu-ifup -enable-kvm -daemonize -cpu host
-smp 8 -clock hpet
```

※ KVMイメージファイルとLXC-VEの設置先ファイルシステムはどちらもext4とし同じ条件とした。
※ 全てのテストは全く同じハードウェア構成のマシン上で実施した。ハードウェア構成は以下の通り。
Machine: x86_64 (x86_64) 4コア8スレッド
  CPU * 8: Intel(R) Xeon(R) CPU X3440 @ 2.53GHz (5066.9 bogomips)
        Hyper-Threading, x86-64, MMX, Physical Address Ext, SYSENTER/SYSEXIT, SYSCALL/SYSRET, Intel virtualization
メモリ 8GB DDR3
HDD 2T(7200RPM) * 2 RAID1