

ウェブサービスとはてなと 仮想化技術

はてな 田中 慎司

stanaka @ hatena.ne.jp

2009/05/29

アジェンダ

- Webサービスのインフラ
 - 三つの指標
- 仮想化技術
 - Xen
- はてなでの取り組み
 - 仮想化を前提としたハードウェア
 - Xenの運用
 - 仮想化のメリット
- クラウドと仮想化

はてなのサービス群



自己紹介

- (株)はてな 執行役員
- 担当領域
 - システムアーキテクチャ
 - スケーラビリティ
 - サーバ・ネットワーク

Webサービスのインフラ

- 大量のリクエスト
- 個々のリクエストは比較的単純
- クリティカル vs 非クリティカル
 - 処理のほとんどは非クリティカル
- 休みなし

三つの指標

- スケーラビリティ
 - サービスの成長の予想が難しい
- コストパフォーマンス
 - 1リクエストの処理にかけられるコストは低い
- 高可用性
 - 24/365

1. スケーラビリティ

- 大量の同じ役割りを持つサーバ群
 - 構成は、基本的に同一
 - 同じ役割りのサーバが10台～100台
 - トラフィックに合わせて速やかに増減

2. コストパフォーマンス

- 1台のハードで多くのリクエストを処理
 - リソース効率 ↑
- 1台の単価を下げる
 - ハードコスト ↓
- 運用コストを下げる
 - 一人あたりのハード数 ↑

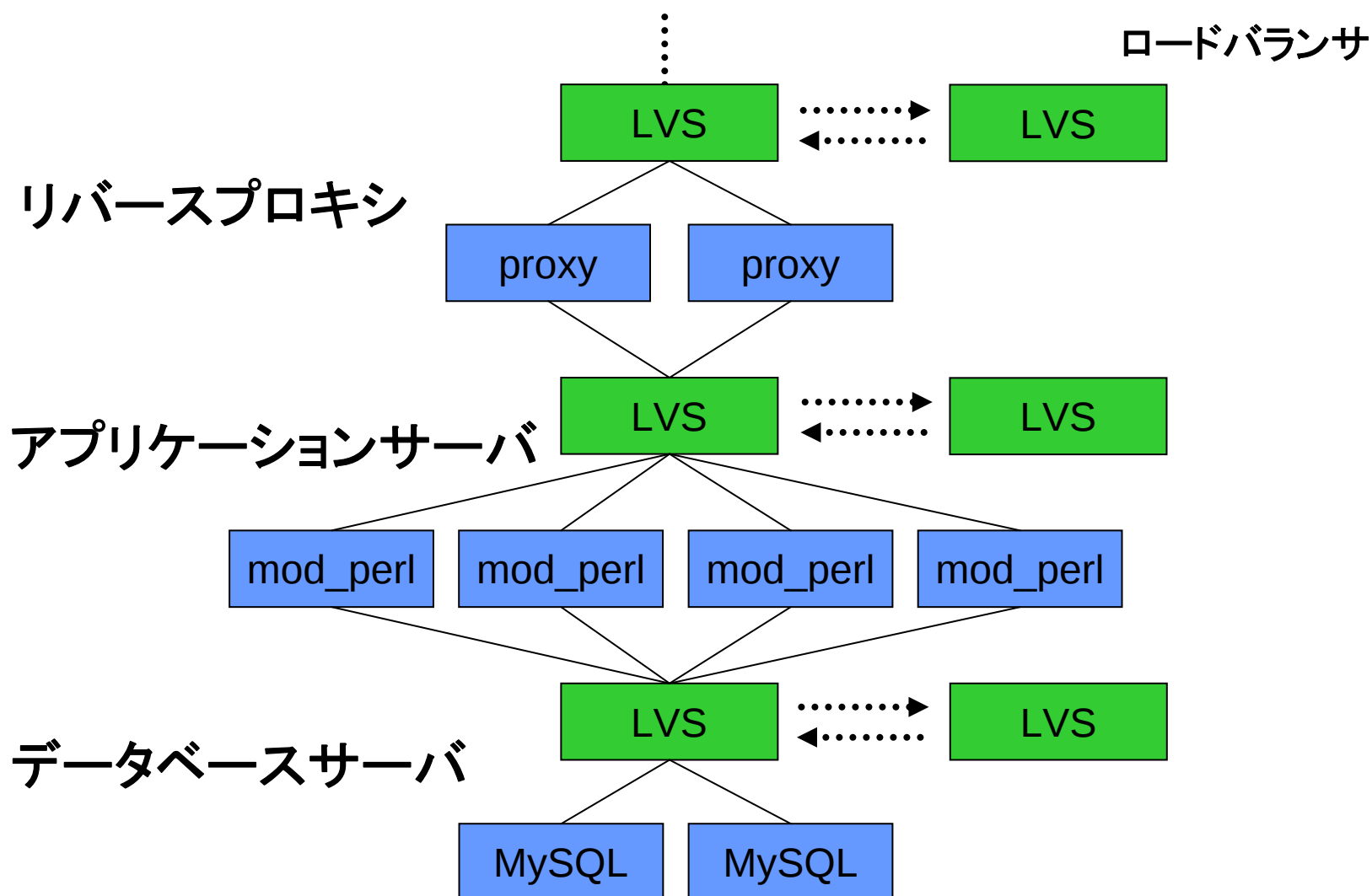
3. 高可用性

- 24/365
- 耐障害性
 - 冗長化
 - フェイルオーバー
- 安定したインフラ
 - 過度なリソース消費の回避
 - 適切なバッファの維持

はてなのシステムの全体像と解説

- 三層構造
 - リバースプロキシ
 - アプリケーションサーバ
 - データベース/ファイルサーバ

バックエンドシステムの三層構造



全体像

- 三層構造がサービス毎に
- サーバー台数の割合
 - Web : DB = 4 : 6 ~ 6 : 4
 - 全体の 6 ~ 7 割が Web or DB
 - それ以外
 - ネットワーク関連、ファイルサーバ、管理用、メール、分散ストレージサーバなど
- 非同期システムが増量中
 - MapReduce計算クラスタなど

仮想化技術への期待

- スケーラビリティ
 - オーバーヘッドの最小化
- コストパフォーマンス
 - リソースの消費効率の向上
 - 運用の柔軟さ
 - 環境の単純化
- 高可用性
 - 環境の隔離

はてなでの仮想化



はてなでのXen

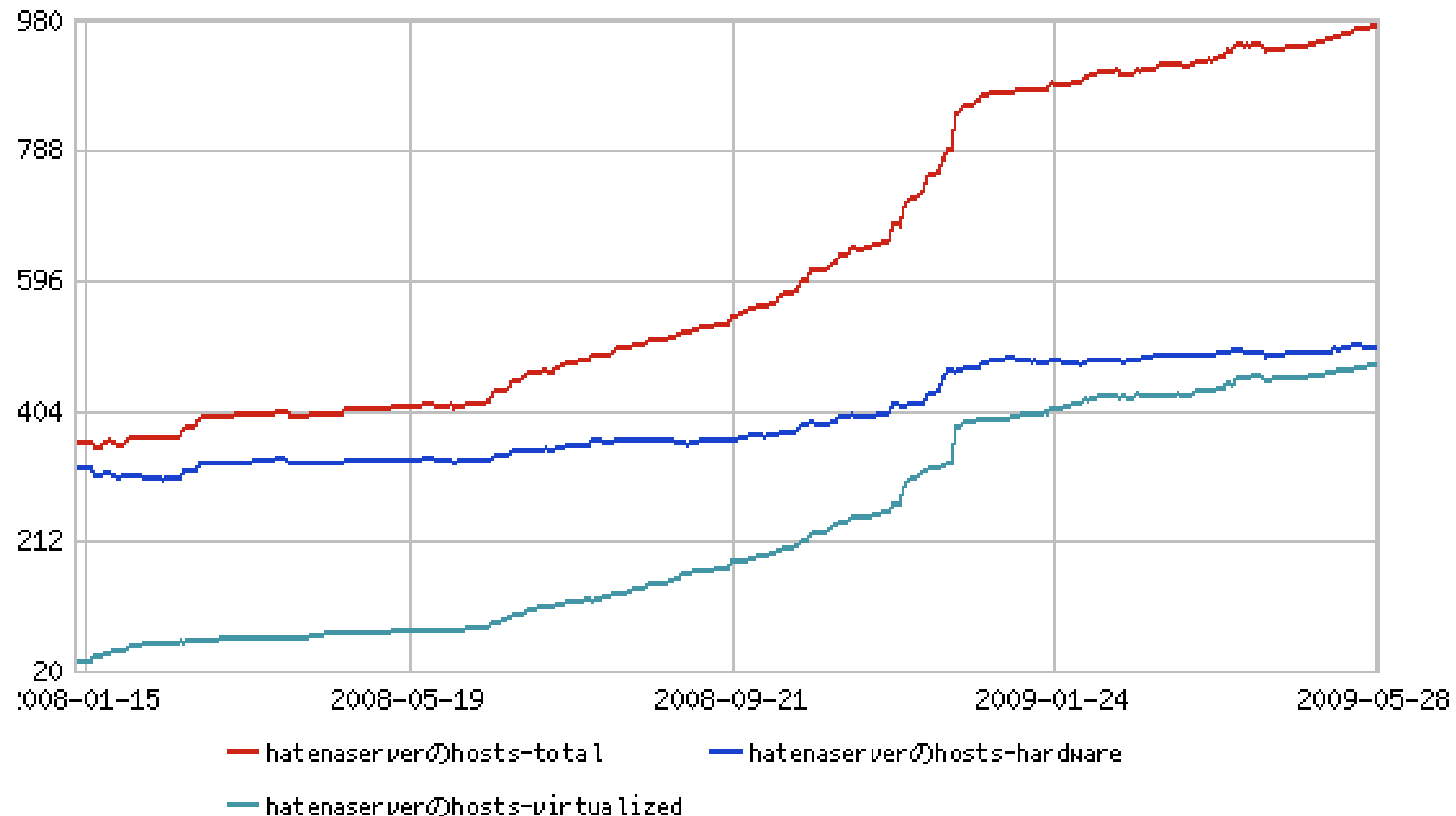
- CentOS 5.2
- Xen 3.0.3
- LVM上にパーティション作成

Xenにおける仮想化のモデル

- 準仮想化(ParaVirtualization)
 - 仮想的なハードウェアを再定義するため、仮想ハードウェアは、操作をするためにはハイパーバイザコールを呼び出す
 - エミュレーションのオーバーヘッドを最小限に抑えられる
 - OSをXen仮想ハードウェア上に移植することが必要
- 完全仮想化(FullVirtualization)
 - 実ハードウェア用のOSがそのまま動作
 - WindowsなどのOSも動作
 - エミュレーションのためのコストが大きくなる
 - デバイスドライバのみ準仮想化とすることも可能

はてなのサーバ台数

■ サーバ 500台 → 仮想化して約1000台



仮想化を前提としたハードウェア

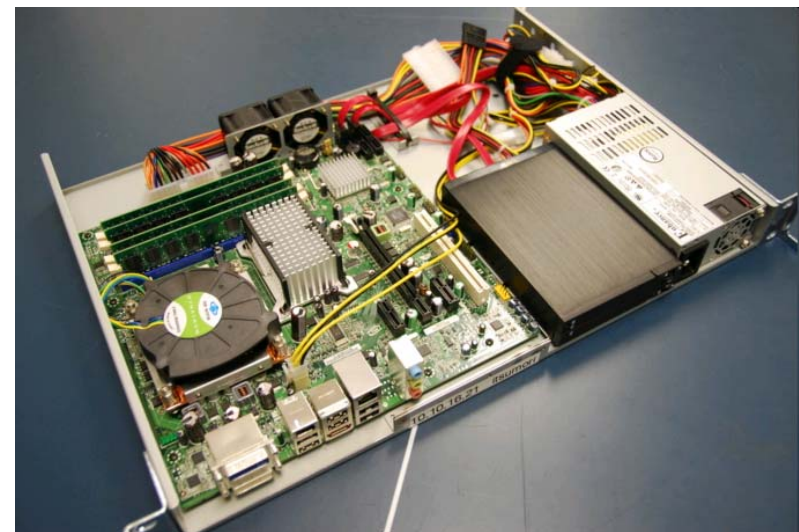
- 安価なハードの有効利用
 - 最小限の管理機能
 - 多コアのCPU
 - 大量のメモリ
 - フレキシブルなIO性能
 - Diskless
 - ハードウェアRAID-10
 - SSD RAID-0
- 管理用のハードコンソールを不要にする
 - IPMI ¥1～2万/サーバ → Intel AMT

仮想化を前提としたハードウェア



仮想化を前提としたハードウェア

- デスクトップ用M/B
 - Intel AMT
- デスクトップ用CPU
- ネットワークポート x 1
- ECCなしメモリ
- RAIDなし or Software RAID



外販も計画しています

仮想化を前提としたハードウェア



参考: Googleのサーバ



仮想化を前提としたハードウェア



仮想化技術のメリット

- IPMIの代替としてのハイパーバイザ
- 環境の抽象化
 - ハード差分の吸収
- リソース消費の制御
 - 過負荷のアラート
 - 負荷の調整
 - monit^{*1}との組み合わせ

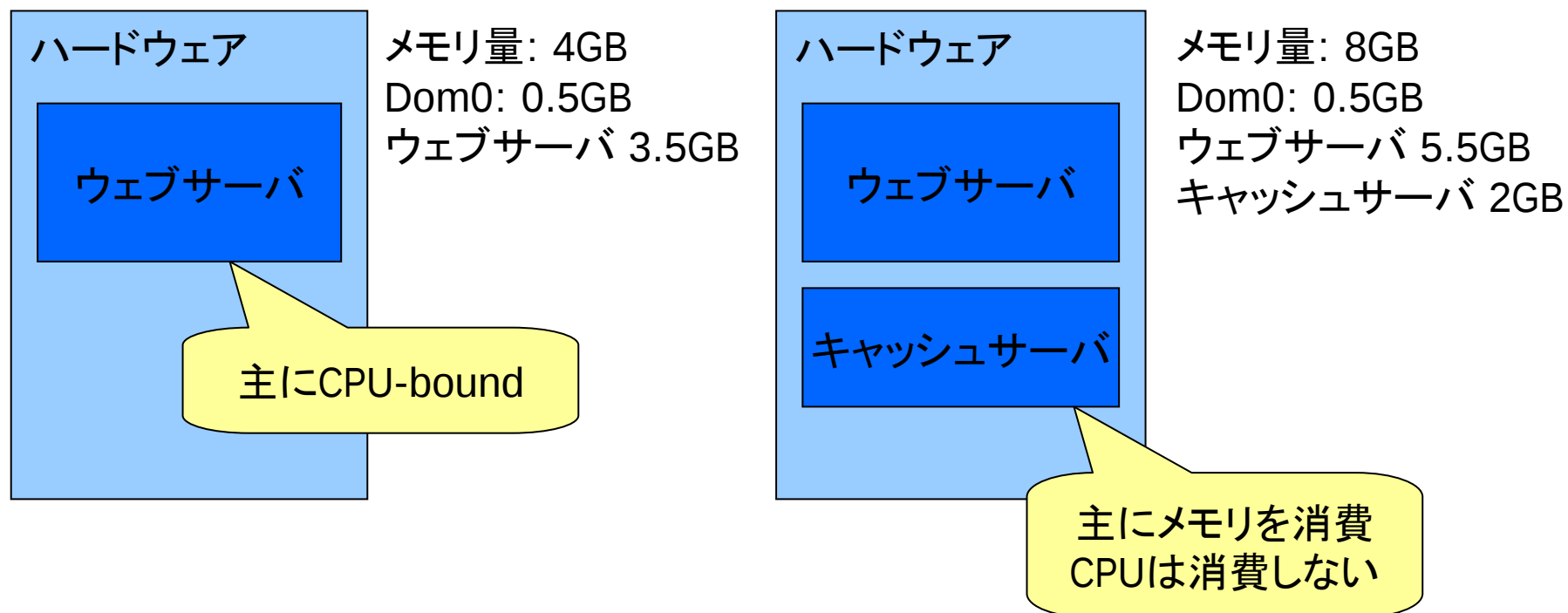
*1: リソース監視ツール <http://mmonit.com/monit/>

仮想化サーバの構築ポリシー

- ハードウェアリソースの利用率の向上
 - 空いているリソースを主に利用するDomUを投入
 - CPUが空いている → ウェブサーバ
 - IOが空いている → DBサーバ
 - メモリが空いている → キャッシュサーバ
- 同居を避ける組み合わせ
 - 同じ傾向、かつ、負荷の高い用途同士
 - 別サーバのウェブサーバ同士など..
- 中央ストレージは使用しない

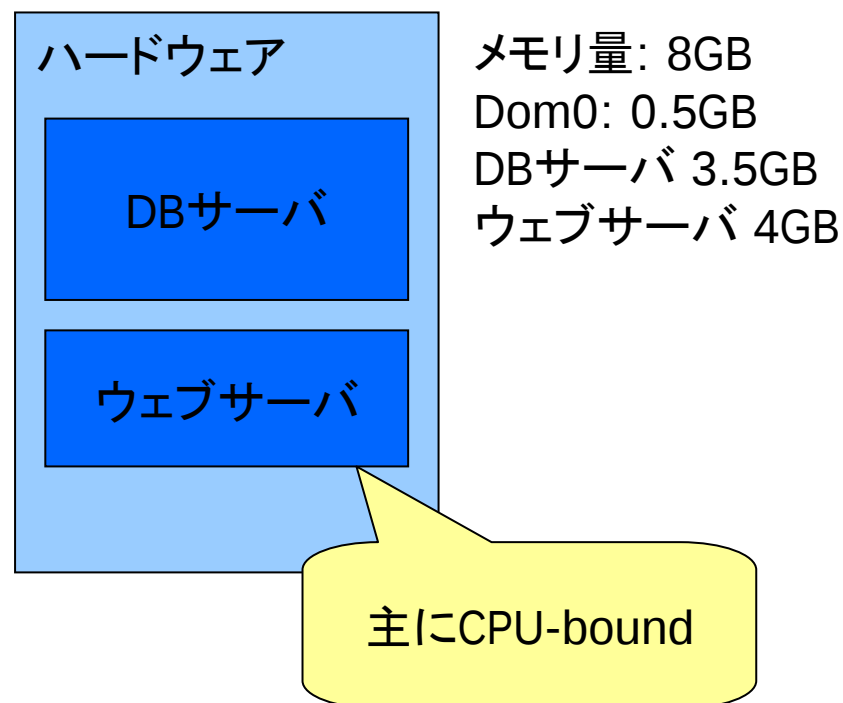
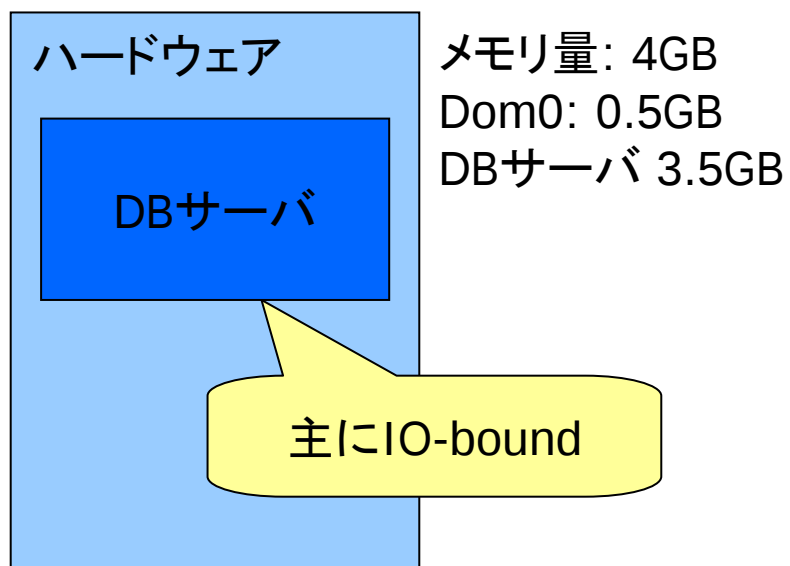
仮想化サーバ

■ ウェブサーバ



仮想化サーバ

■ データベースサーバ



Xenの運用1

- 新規サーバ作成
 - install_xen.sh
 - コマンドー発

```
% ./install_xen.sh
hostname:
ip addr:
mem size: 2048MB
root size: 10GB
hostname must be specified.
./install_xen.sh: -h hostname -i ip_addr -m size_memory[MB]
-s size_root[GB]

% ./install_xen.sh -h test -i 192.168.0.1 -m 1024 -s 20
```

Xenの運用2

- 手動マイグレーション
 - ddでイメージコピー

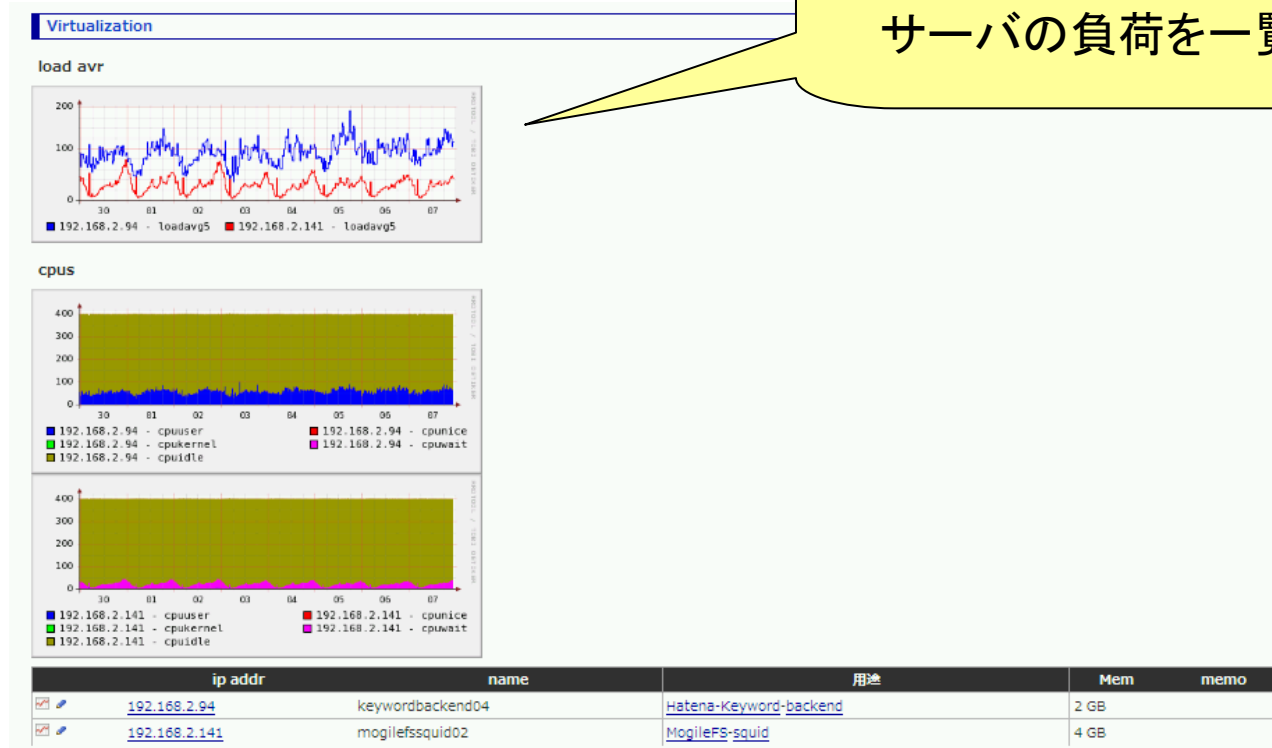
```
dom0a$ time dd if=/dev/mapper/vg00-commondb02root_snapshot | gzip -c  
| nc 192.168.0.2 9210  
20971520+0 records in  
20971520+0 records out  
10737418240 bytes (11 GB) copied, 1096.59 seconds, 9.8 MB/s  
  
real    18m16.651s  
user    9m5.082s  
sys     0m59.292s  
  
dom0b$ nc -l 9210 | gunzip -c | dd of=/dev/vg00/domu2root  
20971520+0 records in  
20971520+0 records out  
10737418240 bytes (11 GB) copied, 1111.61 seconds, 9.7 MB/s
```

Xenの運用3

■ 負荷の監視

■ サーバ管理ツール

サーバの親子関係と、子サーバの負荷を一覧



Xenの運用3 その2

あるラックに含まれるサーバの構成を負荷とともに一覧

Hatena::Servers

サーバー一覧 仮想化ホスト 用途 退役サーバ

仮想化ホスト一覧(beta)

データセンター: サービス: CPU: 状態: ラック: A-2

name (ip addr)		ラック		CPU	Mem (Rate)	HDD	
tsuru 10.10.0.26		代官山 A-2 左	dom0-disk	Quad-Core Opteron 2347 HE	16 GB	465 GB (PERC 5/i RAID-1) 4.8 GB	2CPU
ugodb06 10.10.2.41		Hatena-Ugomemo-db-slave			11 GB	196.9 GB	
ugoproxy03 10.10.2.60		Hatena-Ugomemo-proxy			4 GB	196.9 GB	
gendoh 10.10.0.27		代官山 A-2 右	dom0-disk	Quad-Core Opteron 2347 HE	16 GB	465 GB (PERC 5/i RAID-1) 4.8 GB	2CPU
ugodb02 10.10.2.37		Hatena-Ugomemo-db-slave			15 GB	196.9 GB	
misaka 10.10.0.28		代官山 A-2 右	dom0-disk	Quad-Core Opteron 2347 HE	16 GB	465 GB (PERC 5/i RAID-1) 4.8 GB	2CPU
ugodb03 10.10.2.38		Hatena-Ugomemo-db-slave			15 GB	196.9 GB	

Xenの運用4

- 管理ツールとの強固な連携
 - アプリケーションエンジニアは、あるサービスのサーバがどこに何台あるか把握する必要がない
 - サーバ追加 → 管理ツールに登録して統一管理
 - アプリケーションエンジニアからの操作
 - 監視ツール(Nagios)からの監視
 - 統計データの取得
 - DNSへの登録
 - 全て、管理ツールとシステムの統合済み

Xenの運用4 その2

- Nagios連携

- 管理ツールの登録がそのまま監視対象に

- DNS連携

- 管理ツールに親子関係を登録

子: backend01.host.h

親: hardware01.host.h = parent.backend01.host.h

Xenの運用5

- リソースの自律制御 by monit
 - 仮想OS内で制御
 - プロセスの応答が無くなると強制再起動
 - ハイパーバイザから制御
 - 仮想OSの応答が無くなると強制再起動

仮想化によって得られるもの

- 物理的なリソース制約からの解放
 - リソースの動的な変更
 - VMのマイグレーション・複製

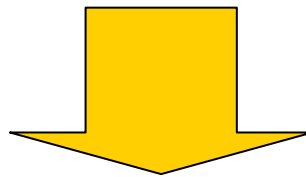
容易なサーバ増設
→ スケーラビリティ

- ソフトレベルの強力なホスト制御
 - 異常動作時の局所化
 - ホストの制御が容易となる

ハードコスト・運用コスト低下
→ コストパフォーマンス・高可用性

クラウド と 仮想化

- ウェブサービスのインフラから見たクラウド
= 仮想化をさらに進めたもの
 - 仮想化レイヤーの抽象度を上げる
- ほぼ自動化された高いスケーラビリティ



システム管理の自律性
ストレージの仮想化
アプリケーションとの積極的連携

まとめ

- Webサービスのインフラ
 - 三つの指標
- 仮想化技術
 - Xen
- はてなでの取り組み
 - 仮想化を前提としたハードウェア
 - Xenの運用
 - 仮想化のメリット
- クラウドと仮想化