
データセンターの仮想化

独立行政法人 産業技術総合研究所
情報技術研究部門
中田秀基



「データセンターの仮想化」

🌐 データセンターを仮想化する？

🌐 データセンターの中で仮想化技術を使う？

Success Story: Animoto.com

- 静止画や動画を音楽と共にアップロードすると「クール」なビデオクリップを自動的に生成するサービス

- ▶ <http://animoto.com>

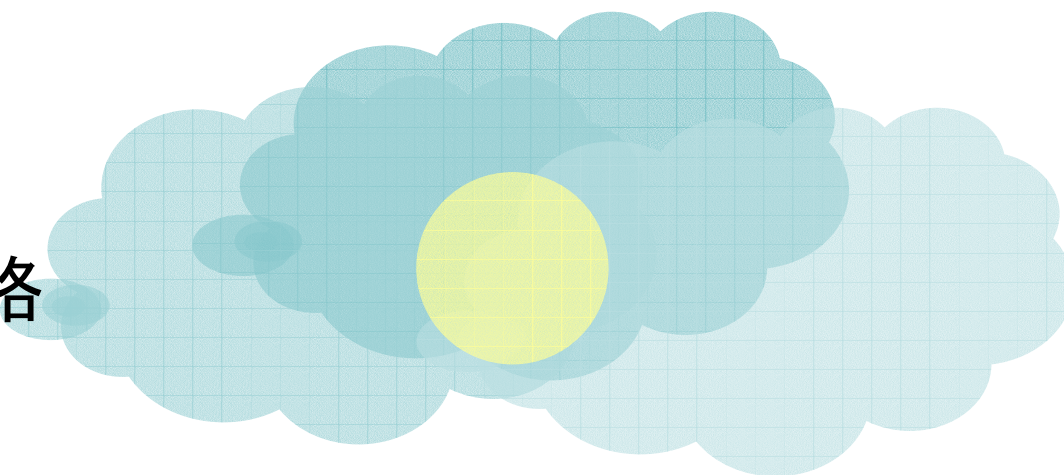
- ▶ Facebookで大ヒット

- ▶ 数十台から数千台へ数日のうちに規模を拡大

- ◎ <http://blog.rightscale.com/2008/04/12/animoto-facebook-scale-up/>

クラウドコンピューティング

- 管理が不要
- 可用性が高い
- スケールによる低価格



Amazonのクラウド

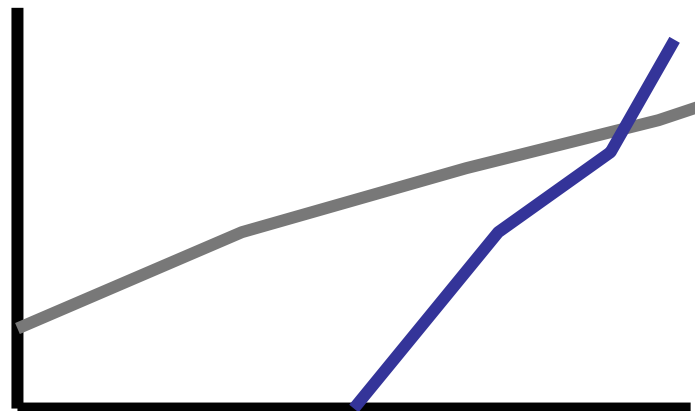


Amazon は裏口でコカインを売っている本屋のようになるだろう。本は、裏でストレージやクラウドコンピューティングを売るための、表向きの顔に過ぎなくなる。

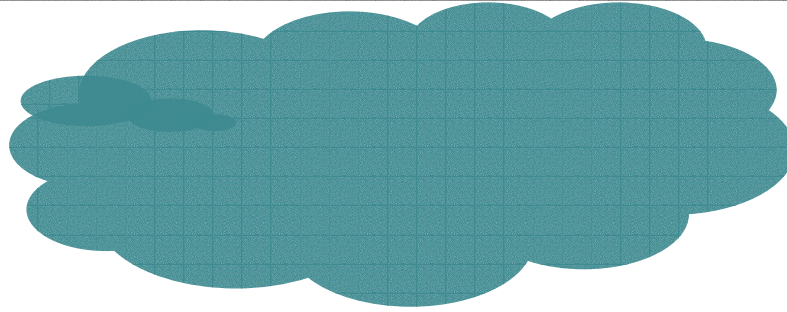
▶ 2008/4/18 – <http://blogs.zdnet.com/BTL/?=8471>



すでに、Amazon のクラウド系サービスのトラフィックはリテールサービス(本など)のトラフィックを上回っている。



2段階の「データセンター仮想化」



発表の概要

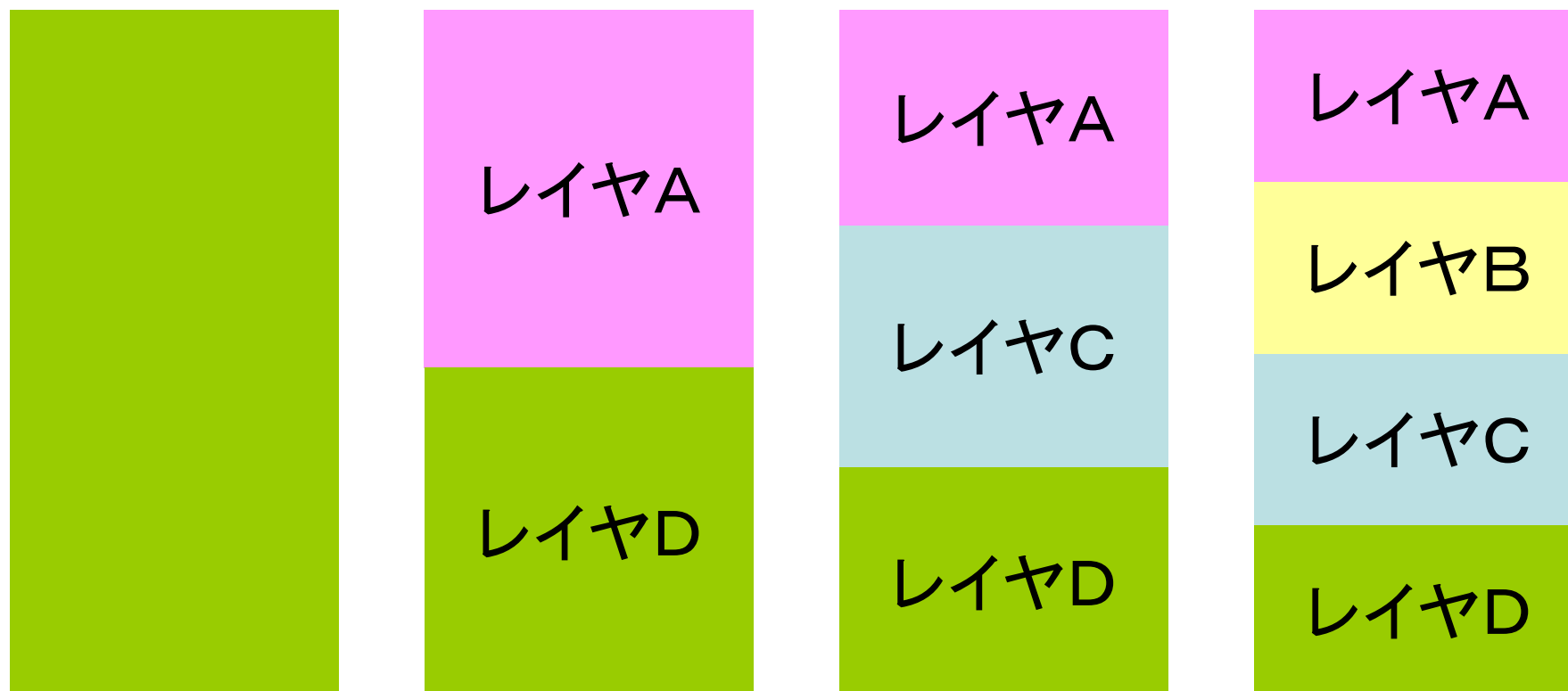
データセンター内仮想化

- ▶ 計算機仮想化とそのメリット
- ▶ さまざまな計算機仮想化技術

データセンター仮想化

- ▶ クラウドコンピューティングの現在

仮想化とは - 抽象化と仮想化



仮想化とは - 抽象化と仮想化



なぜ仮想化？

● コストの削減

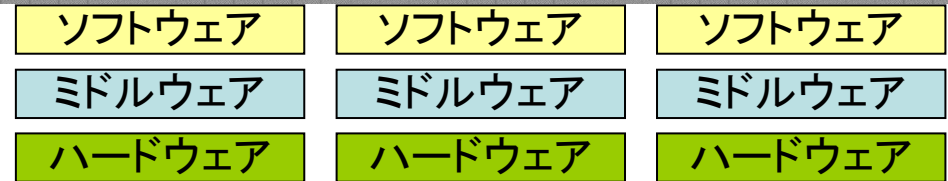
- ▶ ハードウェア保有コスト, リスク
- ▶ 管理コスト

● 仮想化の効果

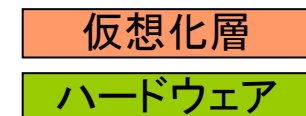
- ▶ サーバ集約
 - ◎ ハードウェア量削減 → 保有コスト低減
 - ◎ 少人数での管理が可能に → 管理コスト低減
- ▶ 実資源と計算機の分離
 - ◎ マイグレーション
 - ◎ より効率的な運用が可能に → 管理コスト低減

なぜ仮想化するのか？

- ハードウェアコスト削減
 - ▶ 集約によるメリット



- 管理コスト削減
 - ▶ ハードウェアの数の削減
 - ◎ 管理する対象が減れば楽になる
 - ▶ ハードウェアからの分離
 - ◎ ハードウェアとシステムのマッピングを自由に変更できる
 - ◎ ハードウェアのメンテナンス



仮想化の背景

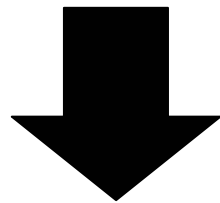
● 計算機の性能とサーバに要請される性能のミスマッチ

▶ 計算機性能はムーアの法則で向上

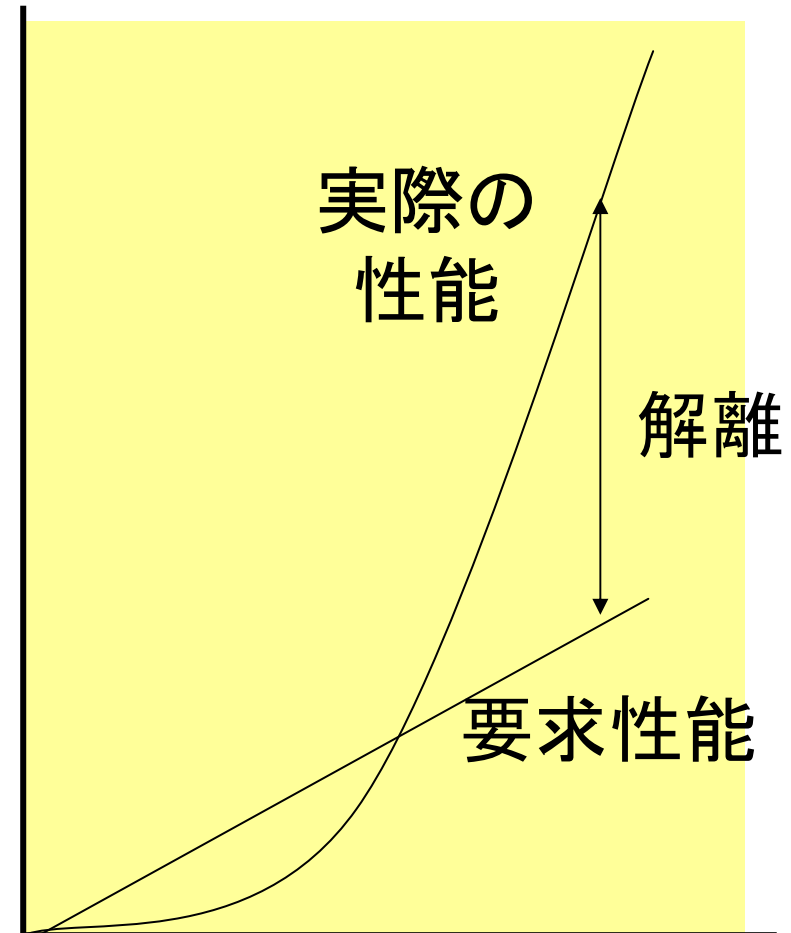
- ④ プロセスの微細化
- ④ マルチコア
- ④ マルチチップ

▶ サーバに要請される性能はそれほど向上していない

- ④ ネットワーク性能が向上しないから？



恒常的に余剰



仮想化の背景(2)

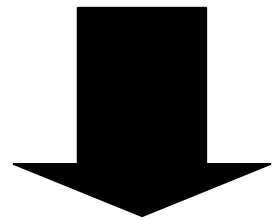
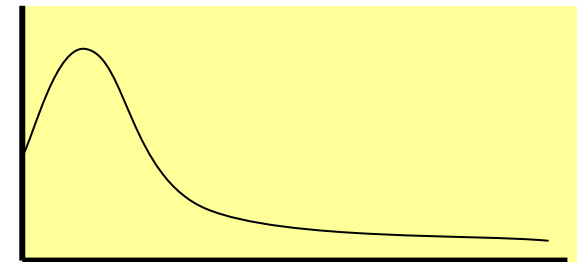
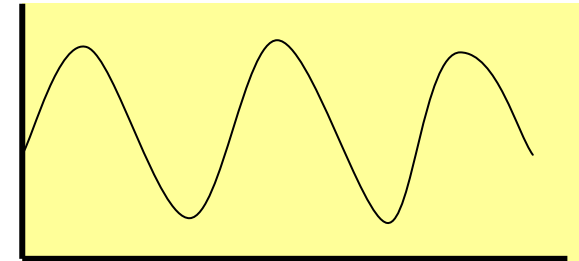
🌐 サーバに要請される性能の時間変動

▶ 24時間, 7日単位の変動

Ⓜ 昼休みに負荷集中, など

▶ サービスインからサービスアウトにかけての長期的変動

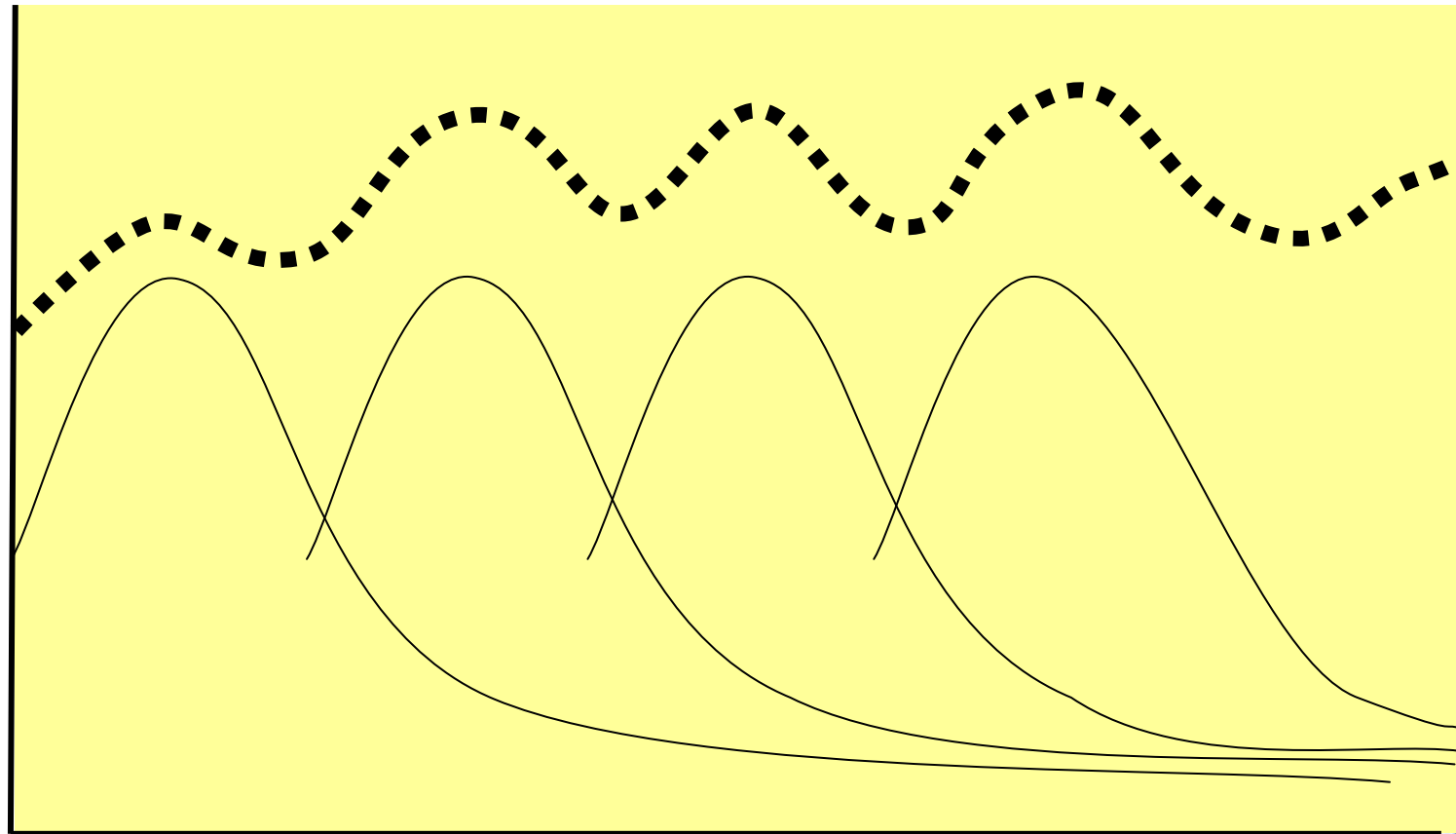
Ⓜ サービスイン直後は高負荷



負荷の不均衡

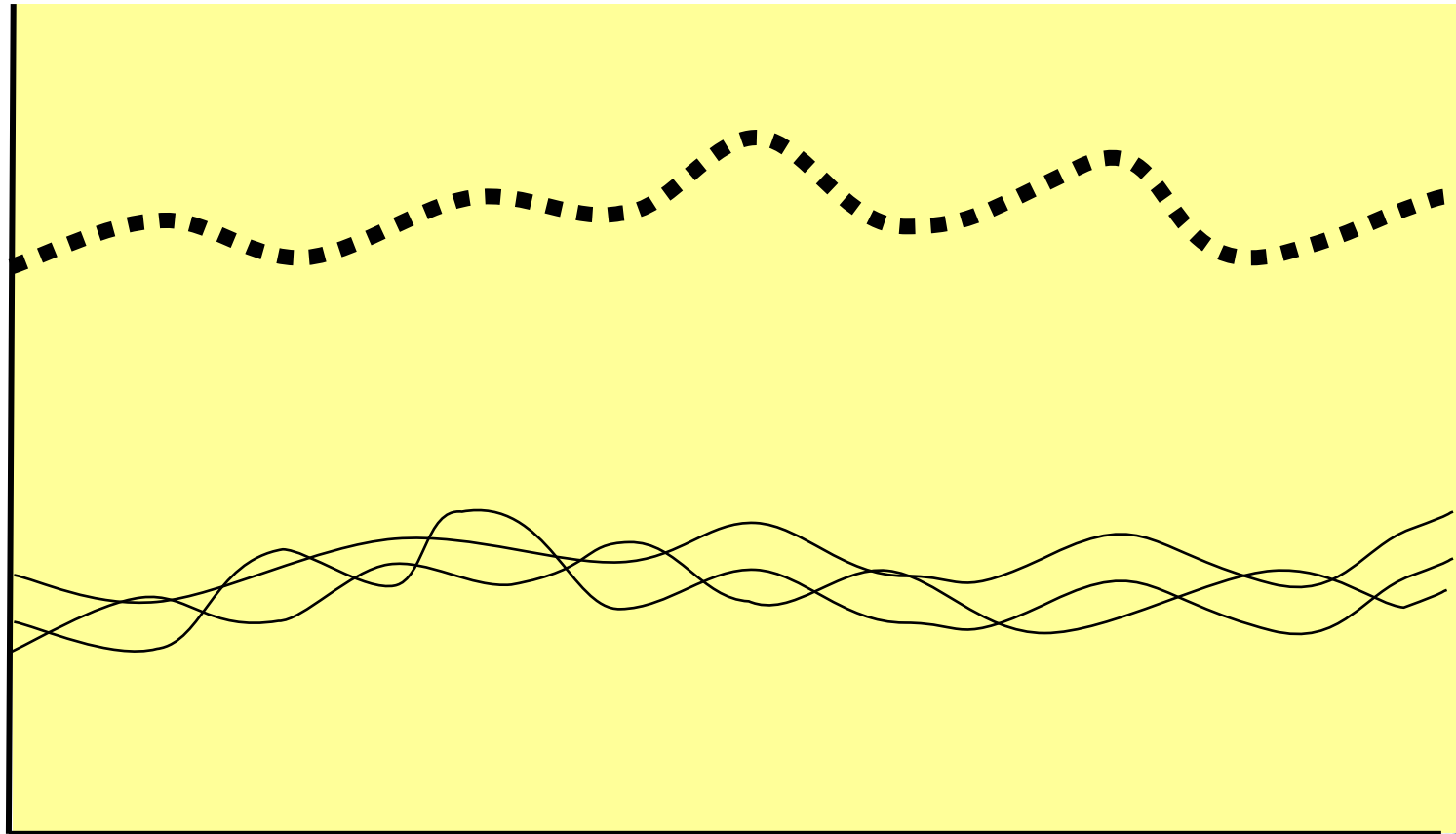
サーバの集約

- 複数のサーバを一つの物理資源で提供
 - ▶ 負荷の平準化
 - ▶ スループットを維持したままHWコストの低減を実現



サーバの集約(2)

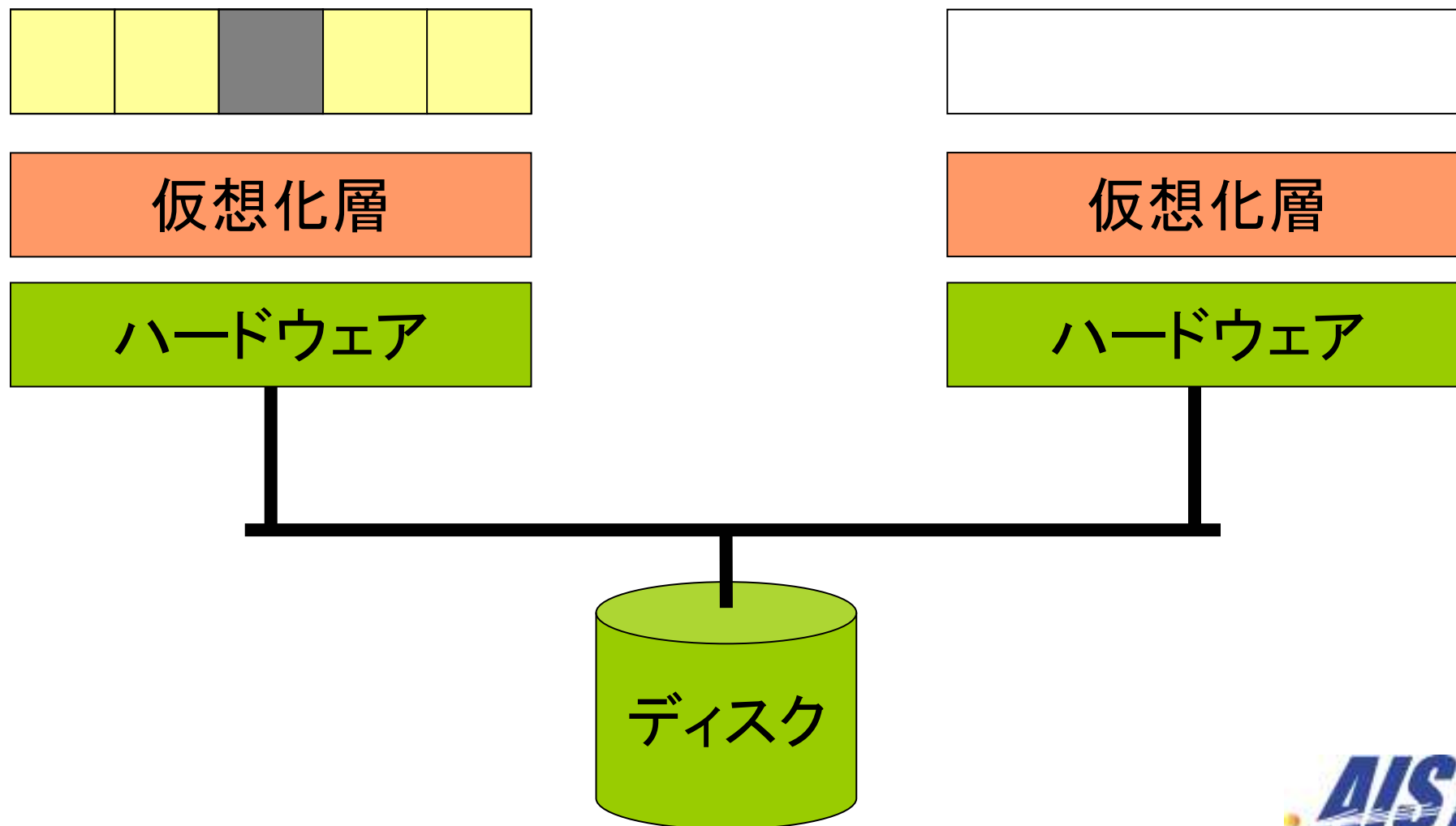
- 負荷が恒常的に低いサーバを一つの物理資源で提供
 - ▶ 管理コストの低減
 - ▶ レガシーNTサーバをP2V変換で仮想化, 集約



ライブマイグレーション

- 稼動中の仮想計算機上のシステムを別の計算機に稼動したまま移動
- ファイルシステムは送信元と送信先で共有していることが前提
 - ▶ NFS, SAN など
- ネットワーク接続も維持できる
 - ▶ ブリッジネットワークで同じサブネット内の移動であれば
 - ▶ スイッチがルーティングし損ねる場合があるが、パケットを出してやれば大丈夫
 - ▶ 別のサブネットであっても、VPNなどを援用することで可能
- 投機的にコピーしておいて、書き換えられたページだけ停止してからコピー
 - ▶ 高速なマイグレーションが可能
- Xen, VMware Infrastructure などでサポート

ライブマイグレーション



ライブマイグレーションの適用例

● ハードウェアメンテナンス

- ▶ メンテナンスのために計画的にハードウェアをシャットダウン

ソフトウェア

ミドルウェア

仮想化層

ハードウェア

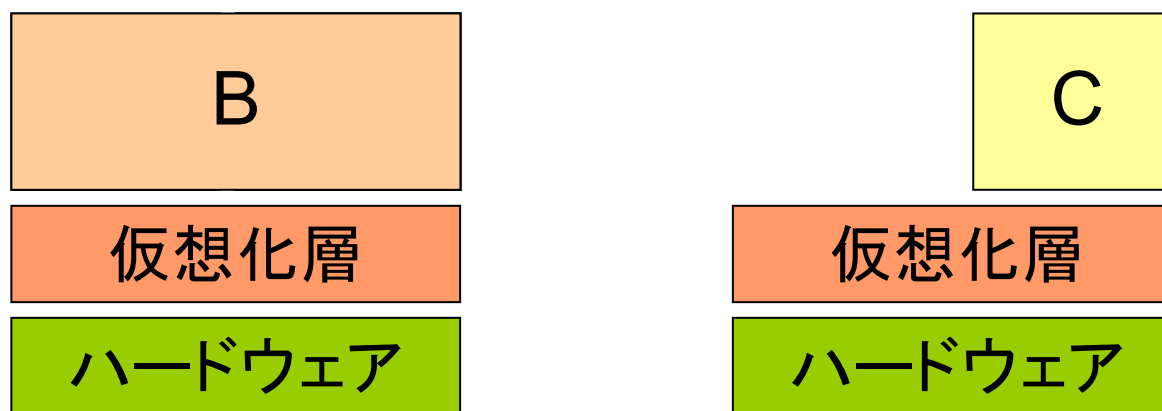
仮想化層

ハードウェア

ライブマイグレーションの適用例

動的負荷分散

- ▶ 負荷の高い仮想サーバはノードを占有
- ▶ 負荷の低い仮想サーバは共有

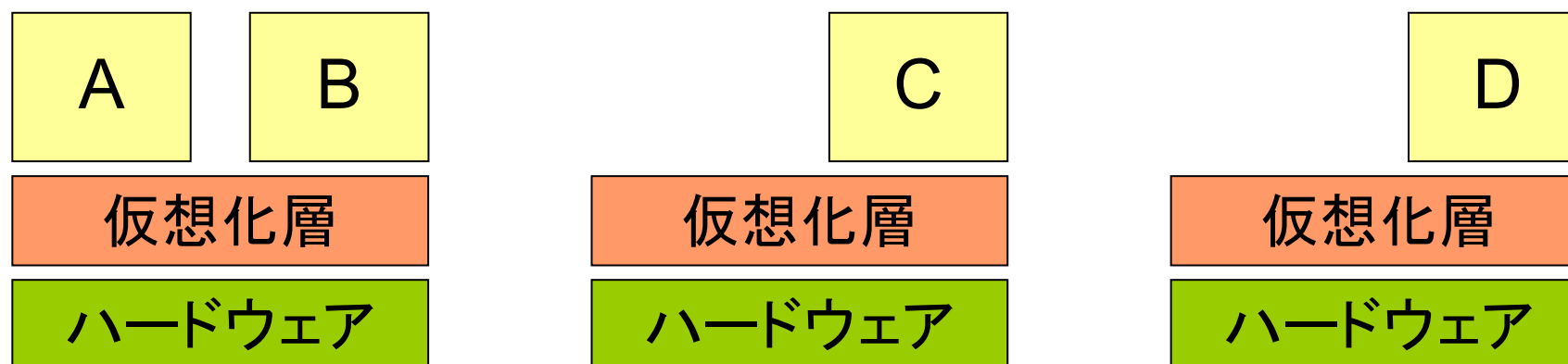


ライブマイグレーションの適用例

● 省電力

- ▶ 低負荷時には一部のノードに仮想サーバを集約
- ▶ 他のノードを停止

● 負荷が上がってきたらハードウェアを再起動してマイグレーション



P2V変換

Physical to Virtual

- ▶ 物理計算機上の稼動イメージを抽出して仮想計算機上で稼動可能に
- ▶ ドライバの入れ替えが必要

- ▶ 一部のOSに対してはOn lineでの抽出が可能
 - Ⓢ 専用プログラムをインストール, 実行
 - Ⓢ プログラムがドライバ情報などを抽出
 - Ⓢ ディスクイメージを吸出し

その他の計算機仮想化の用途

● シンククライアント

● クライアントサンドボックス

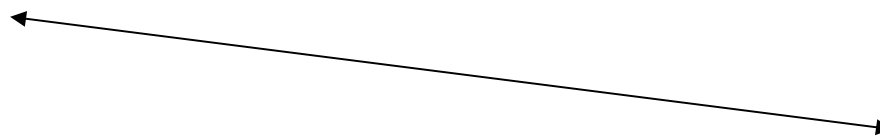
シンククライアントのバックエンドとして

シンククライアント

- ▶ クライアントでは表示するだけ
- ▶ 情報漏えいへの対策として普及
 - 📌 USBメモリをクライアントにさしてもコピーできない

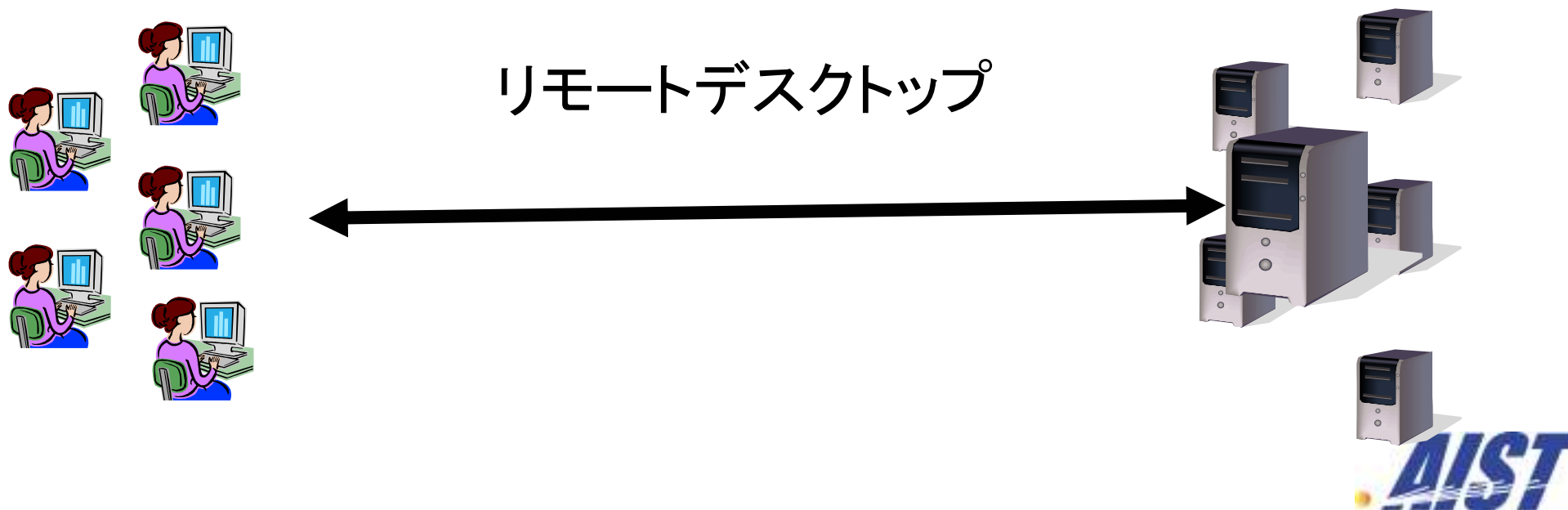


リモートデスクトップ



シンククライアントのバックエンドとして

- バックエンドのサーバクラスタ上でWindowsを実行
 - ▶ 仮想化することにより複数のWindowsを一つのサーバ上で提供可能
 - ▶ ライブマイグレーションで動的に負荷分散も
 - ▶ OSイメージの配備なども実HWを用いるよりは楽



クライアント側サンドボックスとして

- 従業員のPCに直接データを入れるから漏洩する
 - ▶ VMMでサンドボックスを作ってその中でしか作業できないようにすればよい
 - ▶ ゲストOSからは、特定のネットワークアドレスにしかアクセスできないようにVMMで制御
 - ▶ 実行がローカルにおこなわれるのでネットワークが遅くても問題ない
 - ▶ VMware ACE



セキュアVMプロジェクト BitVisor

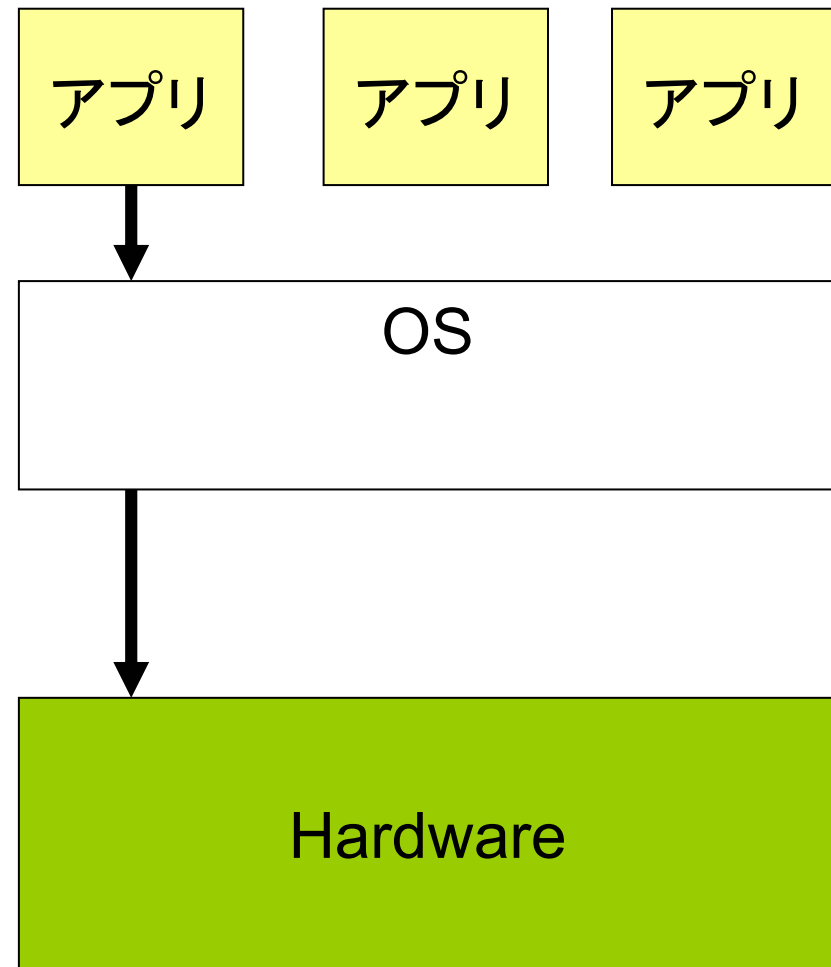
- 文科省のプロジェクト
- 国産の仮想化ソフトを作成し、これにセキュリティ機能を組み合わせる
 - ▶ 仮想化ソフトのレイヤでセキュリティ機能を実現
 - 🌀 VPN, リソース制御など
 - ▶ Ver.0.3 が先日公開
- クライアント側でセキュリティを制御
 - ▶ id 管理と一体化したストレージとネットワーク(VPN)の管理
- 組織
 - ▶ 電通大, 東工大, 慶應, 奈良先端, 豊田高専
 - ▶ 富士通, NEC, 日立, NTT, NTTデータ, ソフトイーサ

仮想計算機の性能

- CPU だけを利用する計算では実計算機とほぼ同じ
 - ▶ CPUをエミュレーションしているわけではない
 - ▶ メモリのマッピング部分でオーバヘッド
- I/Oは遅い
 - ▶ ストレージ, ネットワーク
 - ▶ ドライバ部分で余分なソフトウェアスタックを経由するため
- 実際のアプリケーションへのインパクトはアプリケーション依存
 - ▶ ○ シングルCPU数値演算
 - ▶ ○ Web application, Database
 - ▶ × MPI などによる並列数値演算

計算機仮想化技術の詳細

- アプリケーションのハードウェアに対する操作は最終的にOSの一部のコードによって行われる
- 仮想化を行うには、このルートに何らかの方法でVMMを介在させなければならない



計算機仮想化の分類

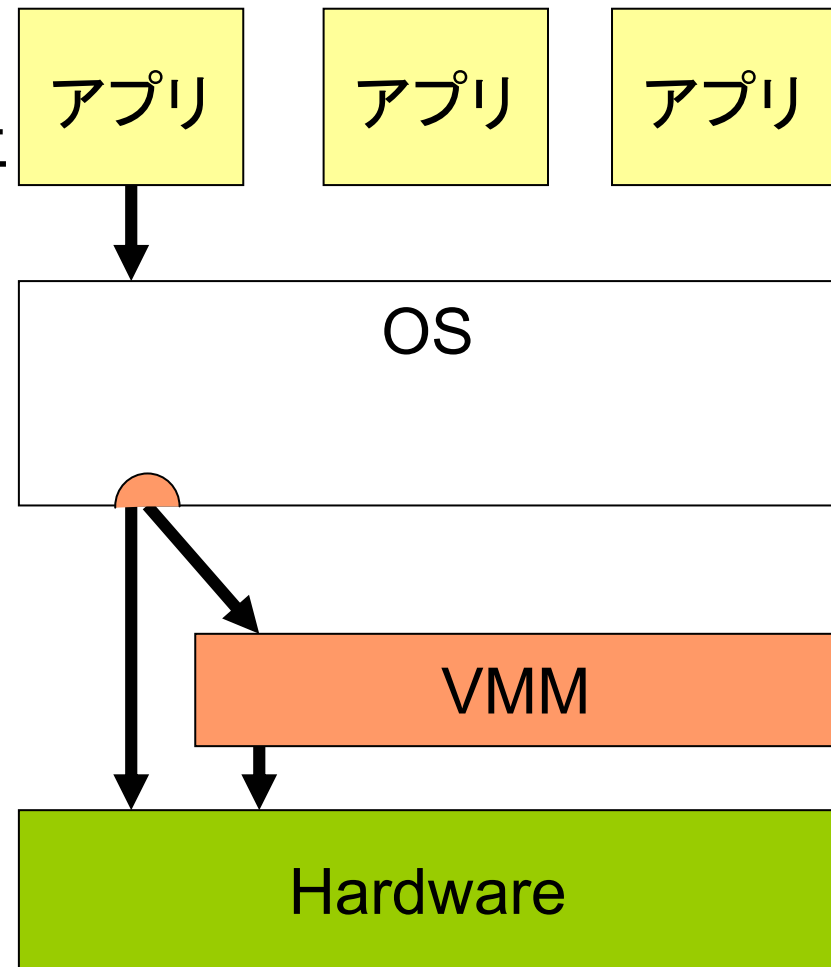
	完全仮想化 BT	完全仮想化 HWサポート	準仮想化
HostOS型	<div> <div>● 計算機仮想化手法</div> <div> <div>▶ 完全仮想化 (Full Virtualization)</div> <div> <div>Ⓢ ハードウェアを含め、計算機全体を完全に仮想化</div> <div>Ⓢ ゲストOSの変更不要 - 何でも動く</div> <div>Ⓢ 2つの方法 <div> <div>Ⓢ コード変換 (Binary Translation)</div> <div>Ⓢ CPUのハードウェアサポートを利用</div> </div> </div> </div> <div>▶ 準仮想化 (Para-virtualization)</div> <div> <div>Ⓢ ゲストOSを変更</div> <div>Ⓢ ハードウェアをエミュレートするわけではない。</div> </div> </div> <div>● OSとの関係</div> <div> <div>▶ OSの上 - ホストOS型</div> <div>▶ OSの下 - Hypervisor型</div> </div> </div>		
Hypervisor OSドライバ			
Hypervisor ドライバ 組み込み			

完全仮想化

- ハードウェアを含めて完全に計算機をエミュレート
 - ▶ 実際のハードウェアとは関係なく、(ドライバが普及している)仮想的なハードウェアを内部のOSに見せる
 - ◎ 例: pcnet32
 - ◎ BIOS や PXE boot のシーケンスもまったく同じ.
 - ▶ ゲストOSは改変する必要なし
 - ◎ どんなOSでも動く
 - ◎ 基本的に, ゲストOSからはゲストであることがわからない

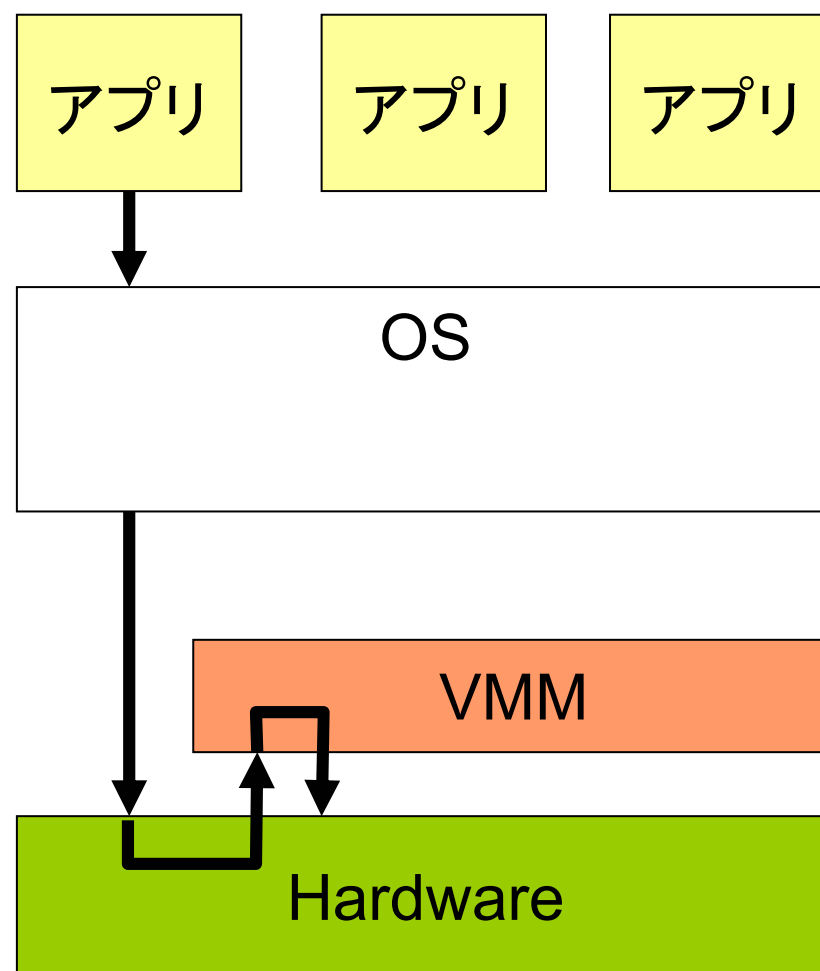
完全仮想化 バイナリ変換法

- ハードウェアにアクセスするコードをトラップ, 動的に改変する
 - ▶ 動的に書き換えるので事前に変更する必要はない
 - ▶ 技術的に非常に高度
 - ▶ 一度書き換えてしまえば, トラップされないので, 意外に実行時のコストは小さい



完全仮想化 CPUのハードウェアサポート

- Intel VT (vanderpool), AMD-V (pacific)
- ▶ CoreDuoやAM2ソケットのAthlon でサポート
- ▶ 相互に互換性無し
- ▶ 新たに仮想計算機用の実行モードを追加
 - ◎ 仮想計算機上の特権命令をトラップして、仮想化システムに引き渡してくれる
- サポートされているCPUがまだ少ないが、仮想化システムの構築は飛躍的に容易に
 - ▶ Xenでもサポート (HVM)
- 必ずしも性能が向上するわけではない。



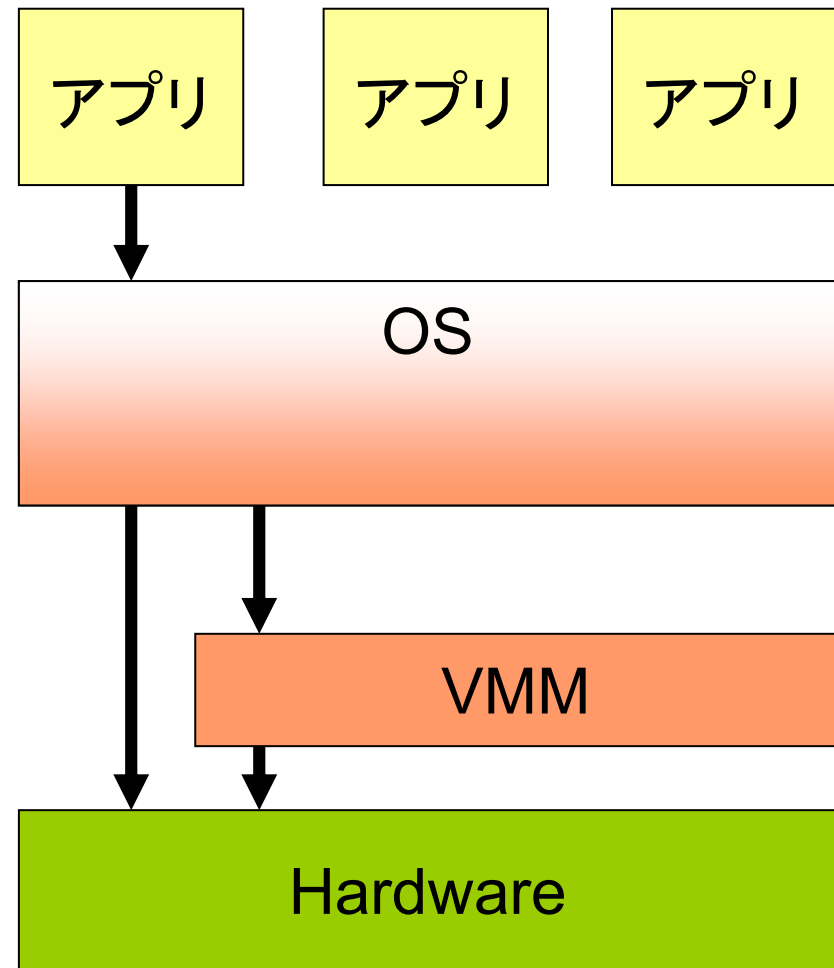
準仮想化

● ゲストOSを一部改変

- ▶ ハードウェアにアクセスする部分をVMMへの呼び出しに変更
- ▶ ハードウェアのエミュレーションコストを削減
- ▶ より高速な実行

● 問題点

- ▶ ゲストOSが限定される
 - ⊗ ソースが入手できるものしか改変できない
 - ⊗ 改変のコストも大きい



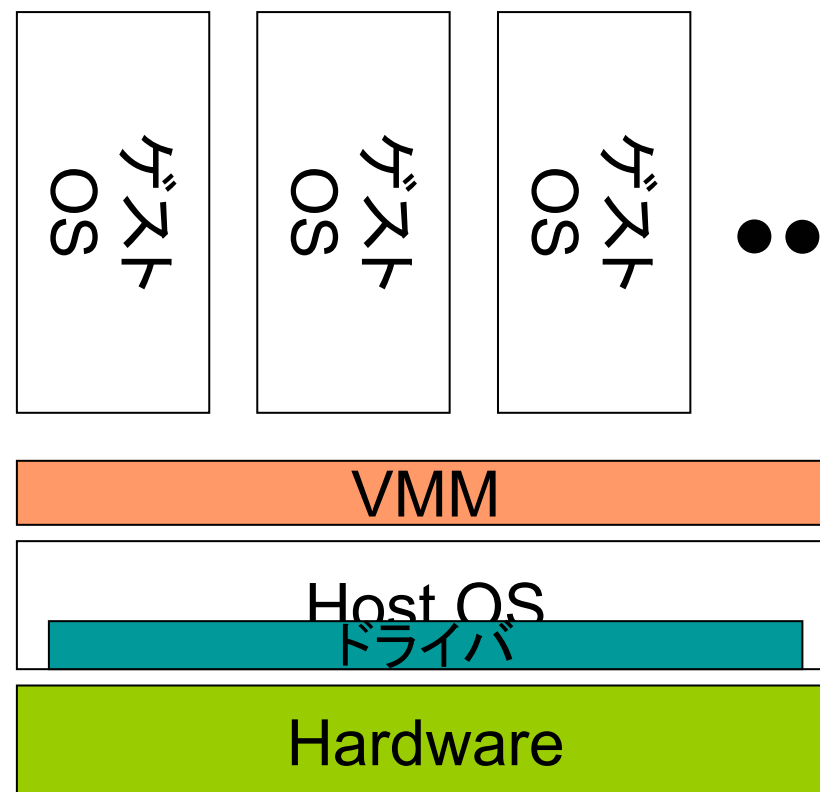
ホスト型

- 通常のOSをホストOSとし、
その上に仮想計算機モニタ
(VMM)を置く

- ▶ VMM上でゲストOSを稼
動

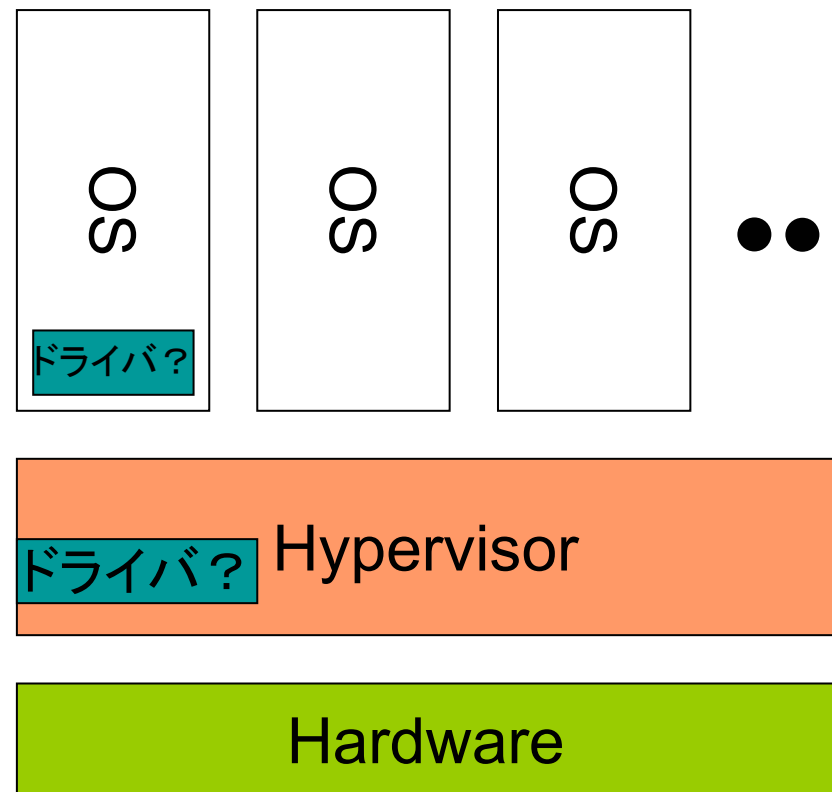
- デバイスドライバはホスト
OSが提供

- ▶ 多様なハードウェアで利
用できる



Hypervisor 型

- ハードウェアの直上にHypervisorと呼ばれるソフトウェア層が稼動。その上でOSが動く
- OS間のスケジューリングをHypervisorで行う
 - ▶ ホスト型よりも柔軟なスケジューリングが可能
- ドライバの位置により2つのタイプ
 - ▶ ゲストOSの一つで？
 - ▶ Hypervisorで？



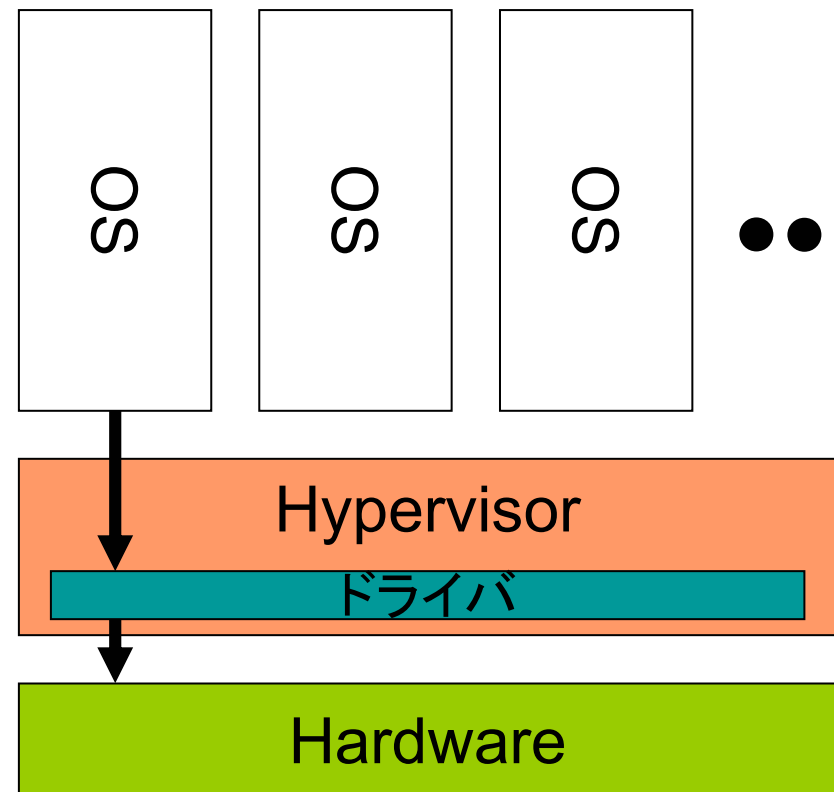
Hypervisor 型

● Hypervisor でドライバを実行

- ▶ ○ 性能的にはもっとも有利
- ▶ X さまざまなハードウェアに対して個別にHypervisorが対応する必要がある
 - Ⓜ 対応ハードウェアが限定される

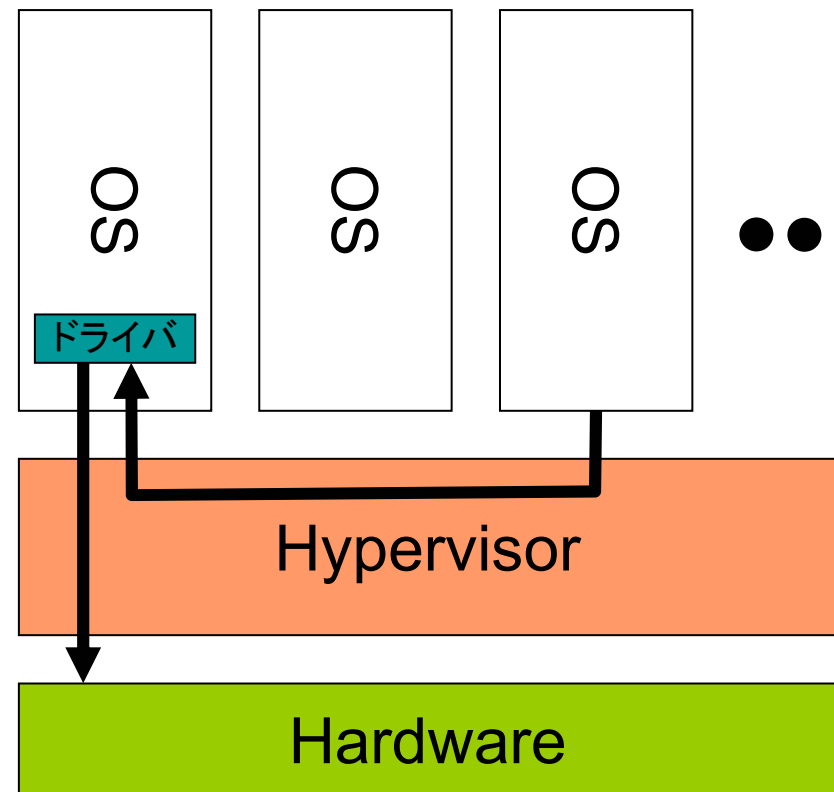
● 例

- ▶ VMware ESX Server
- ▶ 初期のXen(1.X)









Hypervisor 型

- ゲストOSの一つでドライバを実行
 - ▶ ○ OSの持つデバイスドライバをそのまま利用できる
 - ▶ × 性能を出しにくい.
- 例
 - ▶ VMware ESX Server
 - ▶ 初期のXen(1.X)



代表的な仮想計算機

-  VMware
-  Xen
-  Parallels
-  Windows Server Virtualization
-  Vartuozzo / OpenVZ
-  ...

http://en.wikipedia.org/wiki/Comparison_of_virtual_machines

VMWare の製品群

商用の代表的な仮想計算機システム

▶ VMware ESX Server Hypervisor型

➡ ▶ VMware Server HostOS型

▶ VMware Workstation

➡ ▶ VMware Player

▶ VMware ACE

完全仮想化

Xen

- ケンブリッジ大学発

- ▶ 現在はXenSource 社が管理

- オープンソースの仮想計算機システム

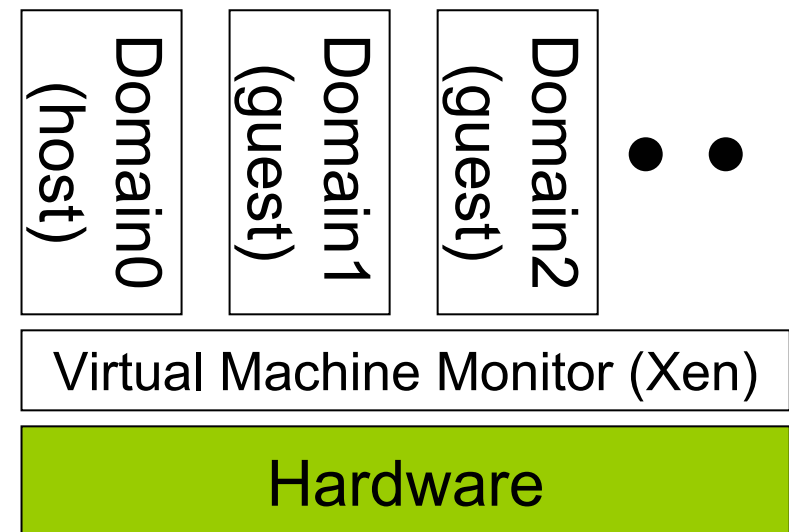
- ▶ HyperVisor型

- ▶ 準仮想化

- Ⓢ ゲストのOSの改変が必要

- ▶ 最近完全仮想化もサポート

- さまざまなディストリビューションにとりこまれつつある



Hyper-V

- Windows Server 2008 (Longhorn) で仮想化はOS標準の組み込み機能に
 - ▶ VMware Server ESX /Infrastructure 対抗
 - ▶ ハイパーバイザ型
 - ▶ 完全仮想化, 準仮想化双方をサポート
 - Ⓢ 準仮想化ならより高速に
 - Ⓢ Xenの準仮想化イメージをサポート. アダプタを介して高速に実行
 - ▶ P2V, V2V変換をサポート
 - Ⓢ 捨てきれないNTサーバ類を収容することで管理コスト削減
 - ▶ 現在RC1- 近いうちに正式版がリリース
 - ▶ ただし...
 - Ⓢ Live Migration が初期のバージョンにはない

計算機仮想化の分類

	完全仮想化 BT	完全仮想化 HWサポート	準仮想化	OS仮想化
HostOS型	VMware WS 他	Parallels		Virtuozzo / OpenVZ
Hypervisor OSドライバ		Windows Server Virtualization Xen HVM	Xen 2.0 以降	
Hypervisor ドライバ 組み込み	VMware ESX Server	VMware ESX Server	Xen 1.0	

OS仮想化

● 計算機仮想化より軽量

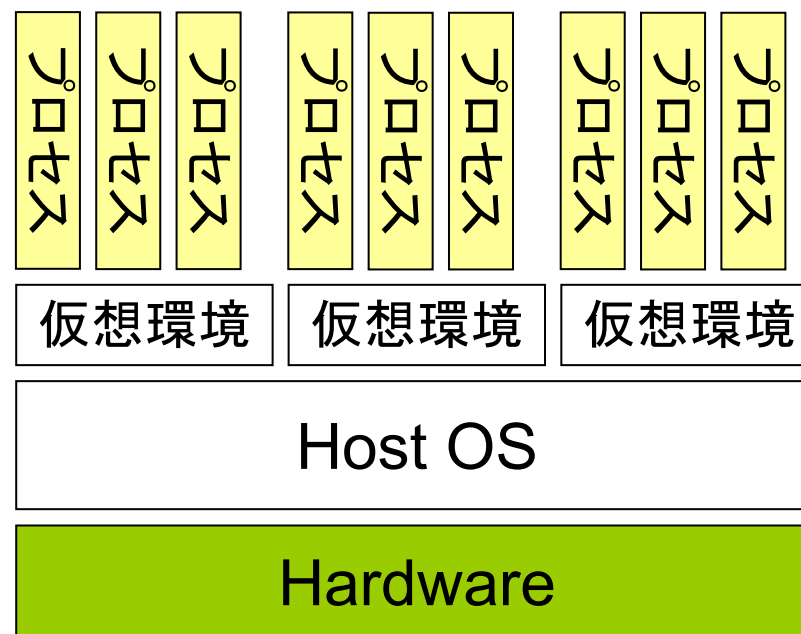
▶ Virtuozzo / OpenVZ

▶ Solaris コンテナ

● ホストOSのカーネルをゲストが共有

▶ ハードウェアの仮想化をしていないため、軽量/高速

▶ アプリケーションのテキストエリアさえ共有



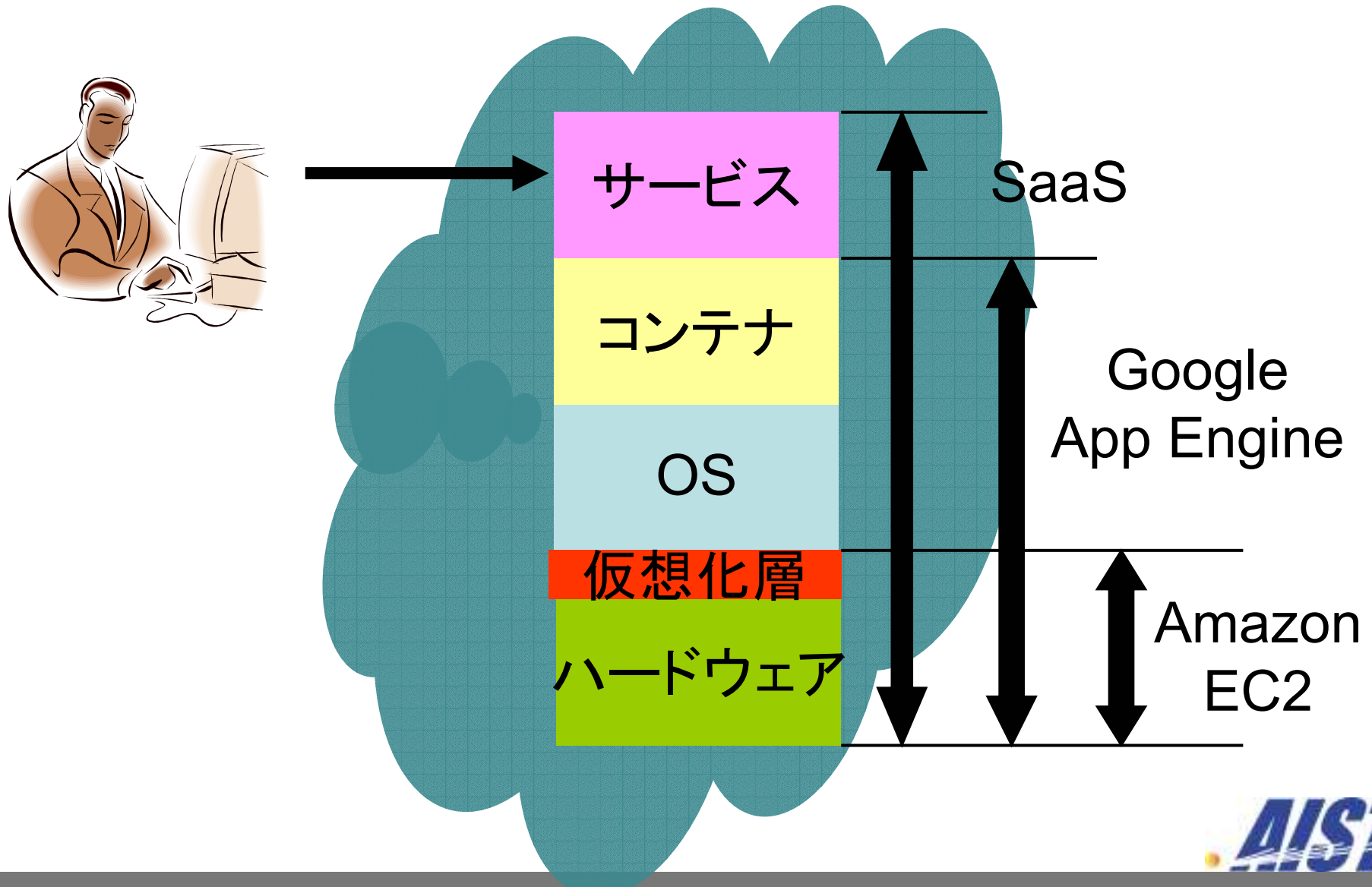
OS仮想化 (2)

- ホストOSとゲストOSが分離されていない
 - ▶ ホストOSとゲストOSは基本的に同じOS
 - ▶ ゲストOS上でのアプリケーションの動作によってホストOSに影響がでるおそれがある
- 軽量であるためホスティング業界では広く用いられている
 - ▶ Virtuozzo – 4GByte メモリ, Apacheだけ動かして70環境までスケール

クラウドコンピューティング



データセンター仮想化のレベル



Amazon のクラウドサービス群

EC2(Elastic Computing Cluoud)

- ▶ Xenによる仮想サーバをクラウド上でホスティング

S3 (Simple Storage Service)

- ▶ 大容量のデータを安価，高信頼で保持

SimpleDB

- ▶ 非常に単純なデータベース機構
- ▶ 属性と値のペア —RDBではない

SQS (Simple Queue Service)

- ▶ 8キロバイトまでのメッセージを高信頼でキューイング

Amazon EC2

- 特定のフォーマットでディスクイメージを作成, S3 にアップロード
- イメージを指定して, 仮想計算機インスタンスをクラウド上に構築
 - ▶ WebサービスAPIで制御
 - ▶ Root権限:ごく普通の計算機として利用可能
- インスタンスをシャットダウンすると, ディスクに書いた内容も失われる
- Core - 2007のOpteron換算で1.0GHz -1.2GHz 程度

	Core数	メモリ	ストレージ	アーキテクチャ	価格
小	1	1.7G	160GB	32bit	\$0.1/hour
大	4	7.5G	850GB	64bit	\$0.4/hour
特大	8	15G	1690GB	64bit	\$0.8/hour
HPC中	5	1.7G	350GB	32bit	\$0.2/hour
HPD特大	20	7G	1690GB	64bit	\$0.8/hour

EC2 - お値段

小を1ヶ月

- ▶ $\$0.1 * 24(\text{時間}) * 30 = 72\text{ドル}$
- ▶ 激安PCを考えるとそれほど安くない？

HPC特大を10ノード * 1日

- ▶ 200Core!
- ▶ $\$0.8 * 24 * 10 = 192\text{ドル}$

Amazon S3

- 高信頼, 大容量ストレージをクラウドでホスティング
- APIはWebService
- お値段
 - ▶ データ保持1GB 15セント/月
 - ▶ 書き込み 10セント /1GB
 - ▶ 読み出し 17セント – 10セント /1GB
- 典型的使い方
 - ▶ Webの画像置き場
 - ▶ バックアップストレージ
 - Ⓜ PCから直接マウントも可能

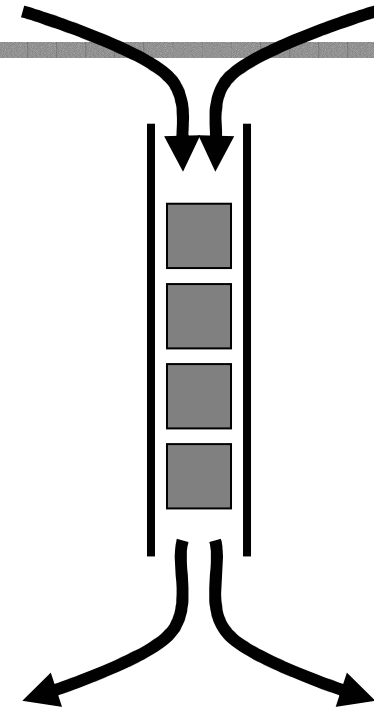
Amazon Simple Queue Service

🌐 高信頼のメッセージキュー

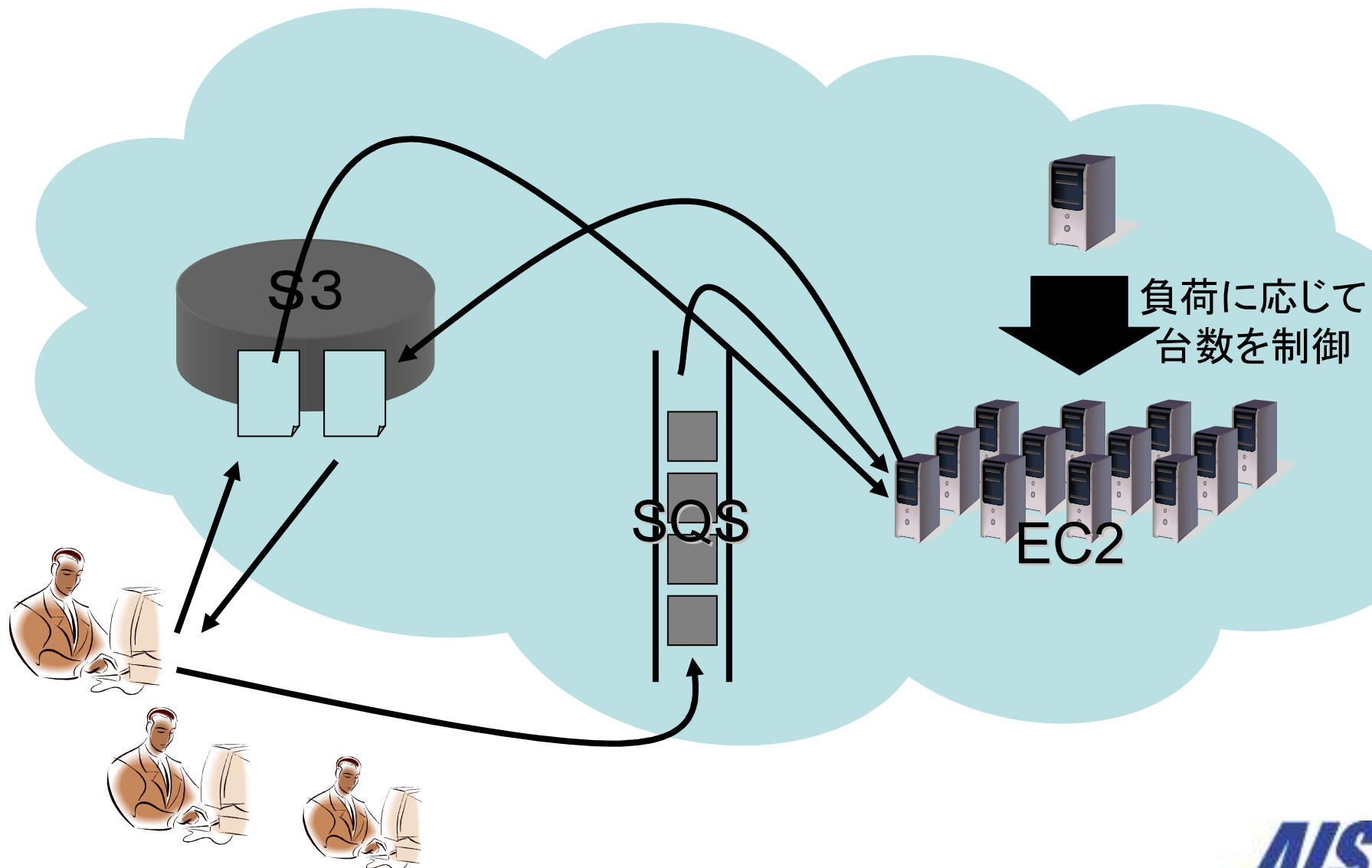
- ▶ First In – First Out
- ▶ メッセージ追加
- ▶ メッセージ取り出し
- ▶ メッセージ削除

- ▶ 取り出し時にタイムアウト付のロックがかかる

🕒 一定時間削除されないとロックが解除され、再度取り出し対象となる。



使用イメージ:ビデオコーデック変換アプリ



Amazon クラウドのエコシステム

- 🌐 Amazon は基本的なWeb Service APIしか提供しない
 - ▶ 機能は十分だがインターフェイスとしては不十分
- 🌐 3rd パーティが, さまざまなサービスを展開
 - ▶ RightScale – EC2上でのシステム構築サポート
 - 🌀 Animoto 社もRightScaleの技術を使用
 - ▶ JungleDisk – S3をバックアップストレージとして使用.

Googleのクラウド

Google Apps

- ▶ Mail, Calender, Docsなどを企業を対象に有料でホスティング

Google App Engine

- ▶ 特定のAPIを用いて記述したサービスをGoogleがホスティング
 - ⊗ Ruby on Rails と等価なWeb Application Frameworkを提供
 - ⊗ スケーラブル・リライアブルなDBアクセス, Googleのアカウントでのユーザ認証
 - ⊗ 負荷に応じて自動的にスケールアウト
 - ⊕ APIが非常に注意深く設計されている
 - ⊗ 現在のところPythonのみ. 次はJavaか?
- ▶ 一定量までは無料, それ以上に対しては従量課金を予定
 - ⊗ 料金体系はAmazonとほぼ同じになりそう

Amazon, Google クラウドのメリット

固定費用 -> 従量課金

- ▶ 固定資産を持つリスクを回避

超大規模化によるスケールメリットで低価格

- ▶ 1000台でも10000台でも管理コストは同じ(?)
- ▶ 管理コストが相対的に安くできる

Amazon, Googleの技術力による高可用性, スケールアウト性

- ▶ Google App Engine のDB
- ▶ Amazon SimpleDB, S3

Amazon, Google クラウドのデメリット

ネットワーク的に遠い

- ▶ データセンターが北米にある(と思われる)ため.
- ▶ レイテンシだけでなくスループットも低い

データを外部に出すことに対する心理的, 法的障壁

まとめ



「コンピュータ」は世界に5つあればいい？

The world needs only five “Computers”

- ▶ 2006/11/10 Sun Microsystems CTO, Greg Papadopoulos in his blog
http://blogs.sun.com/Gregp/entry/the_world_needs_only_five
- ▶ ここでいう大文字の「Computer」は、いわゆる単体の計算機ではなくて、大量の計算機を束ねたもの。
- ▶ 高速ネットワークで接続された、大量の計算機とストレージ群から構成される5000ノード程度のクラスタが分散して配置され、それらがさらに高速ネットワークで接続されている
- ▶ Google, Microsoft.live, Yahoo!, Amazon, eBay, Salesforce.com ...

今後の展望

● 仮想計算機技術は成熟期に

- ▶ もはや「当たり前」の技術
- ▶ ハードウェアサポートは今後も進展
 - ◎ 応用可能範囲の拡大
- ▶ 今後はストレージの仮想化が重要

● データセンターそのものを仮想化しクラウドに置くことが一般的に

- ▶ 省エネルギーの観点からも「超大規模」データセンターが有利
- ▶ 仮想化計算機技術はデータセンター仮想化の基盤技術に

ご清聴ありがとうございました

Q/A

