

---

# グリッドと仮想化

独立行政法人 産業技術総合研究所  
グリッド研究センター  
中田秀基



# 概要

---

## ● なぜ仮想化するのか？

- ▶ 仮想化技術の用途

## ● 仮想化技術の紹介

- ▶ 計算機仮想化の分類

## ● 今後の展望

# 仮想化とは

- 密接に結合していた上下レイヤを分離し、上位レイヤと下位レイヤのマッピングを自由にする
- レイヤの数だけ仮想化が可能な界面が存在する



# グリッド/SOA と仮想化

## ● サービス指向アーキテクチャ

▶ アプリケーションとソフトウェアモジュールを分離

▶ ソフトウェアモジュール  
(サービス)を別のコンテナ  
内で稼働させる

ソフトウェア  
モジュール

ソフトウェア  
モジュール

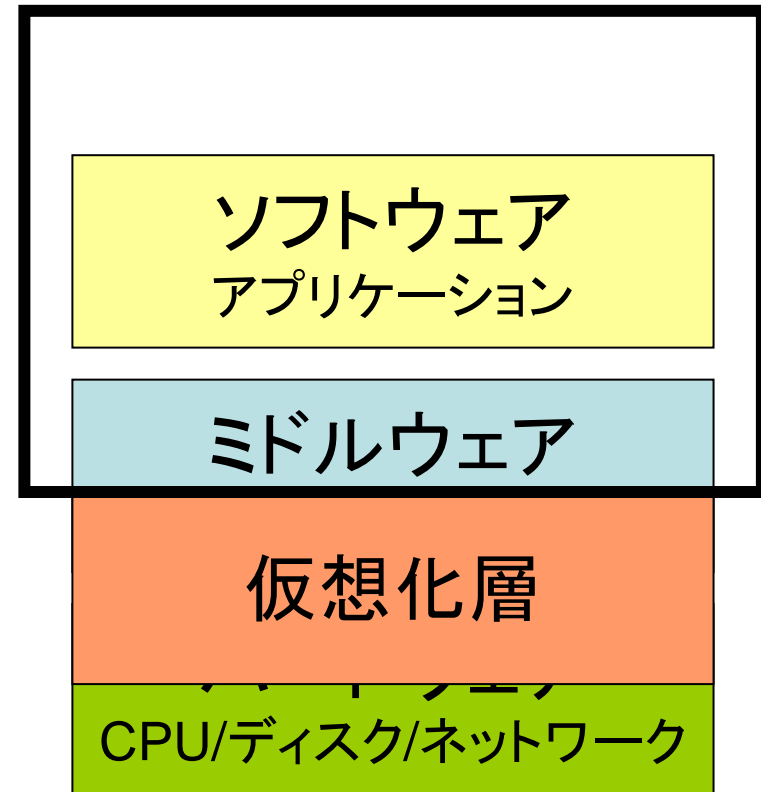
アプリケーション

OS

ハードウェア

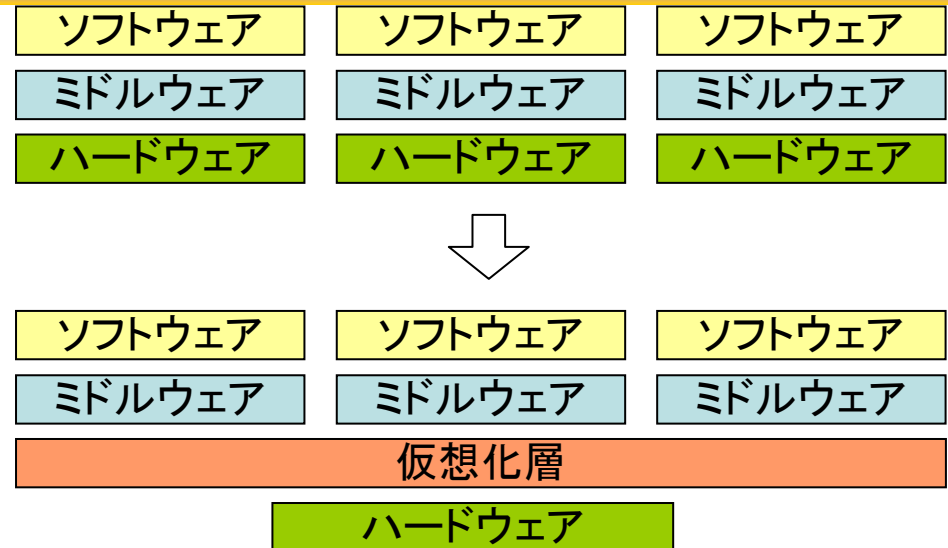
# (計算機の)仮想化とは？

- ハードウェアとミドルウェアの間に仮想化層を導入
- インストールされたアプリケーションとOSの組を直接操作できる対象として切り出す

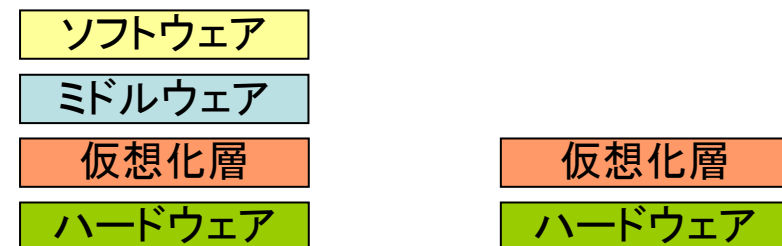


# なぜ仮想化するのか？

- **ハードウェアコスト削減**
  - ▶ 集約によるメリット



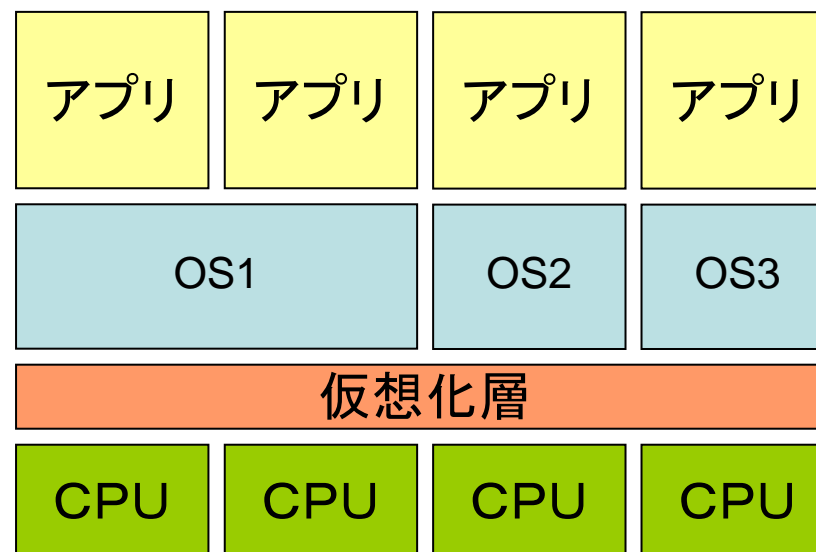
- **管理コスト削減**
  - ▶ **ハードウェアの数の削減**
    - ◎ 管理する対象が減れば楽になる
  - ▶ **ハードウェアからの分離**
    - ◎ ハードウェアとシステムのマッピングを自由に変更できる
    - ◎ ハードウェアのメンテナンス



# 計算機仮想化の歴史

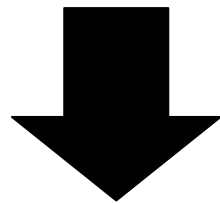
## IBM 370 の時代から

- ▶ メインフレーム, 商用の UNIX (AIX, HP-UX など) では当たり前機能
- ▶ ロジカルパーティションに資源を分割
- ▶ 個々のパーティション内に独立して OS をインストール可能

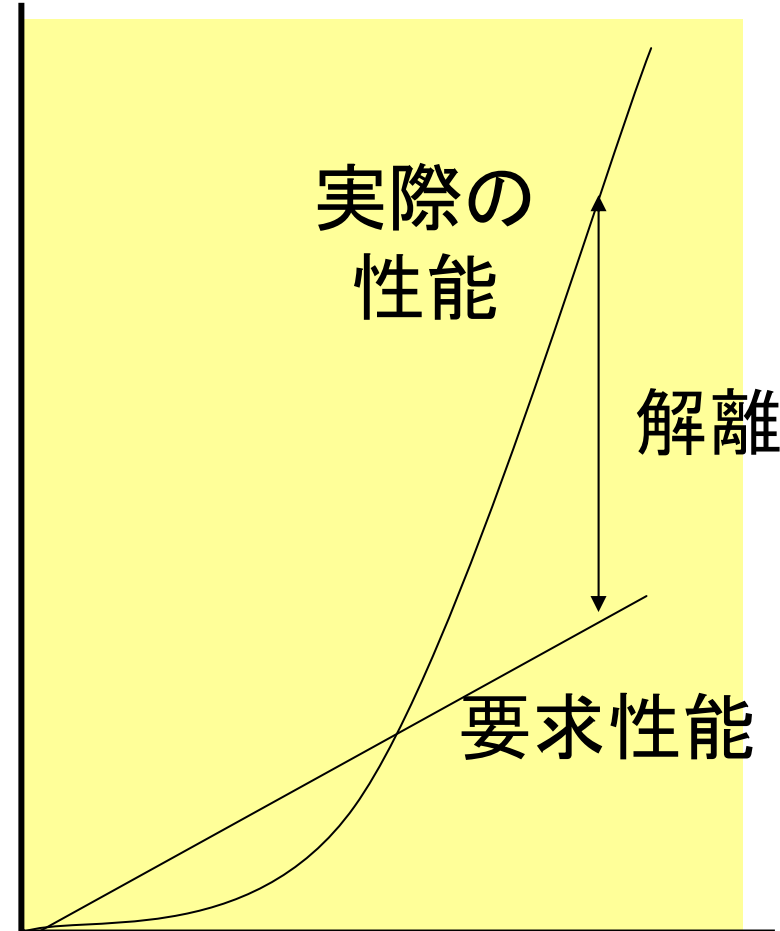


# 仮想化の背景

- 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)

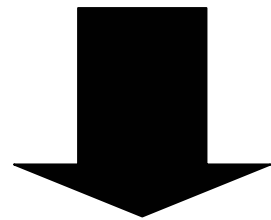
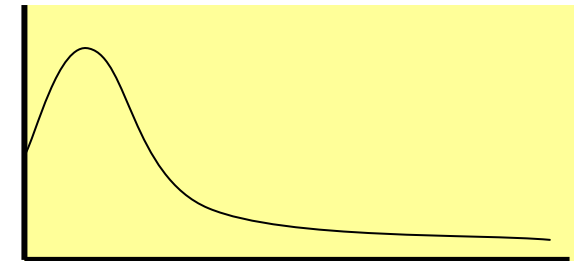
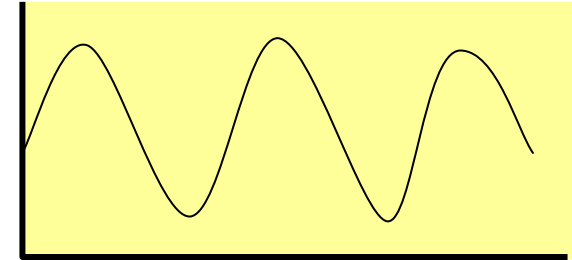
### ● サーバに要請される性能の時間変動

▶ 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

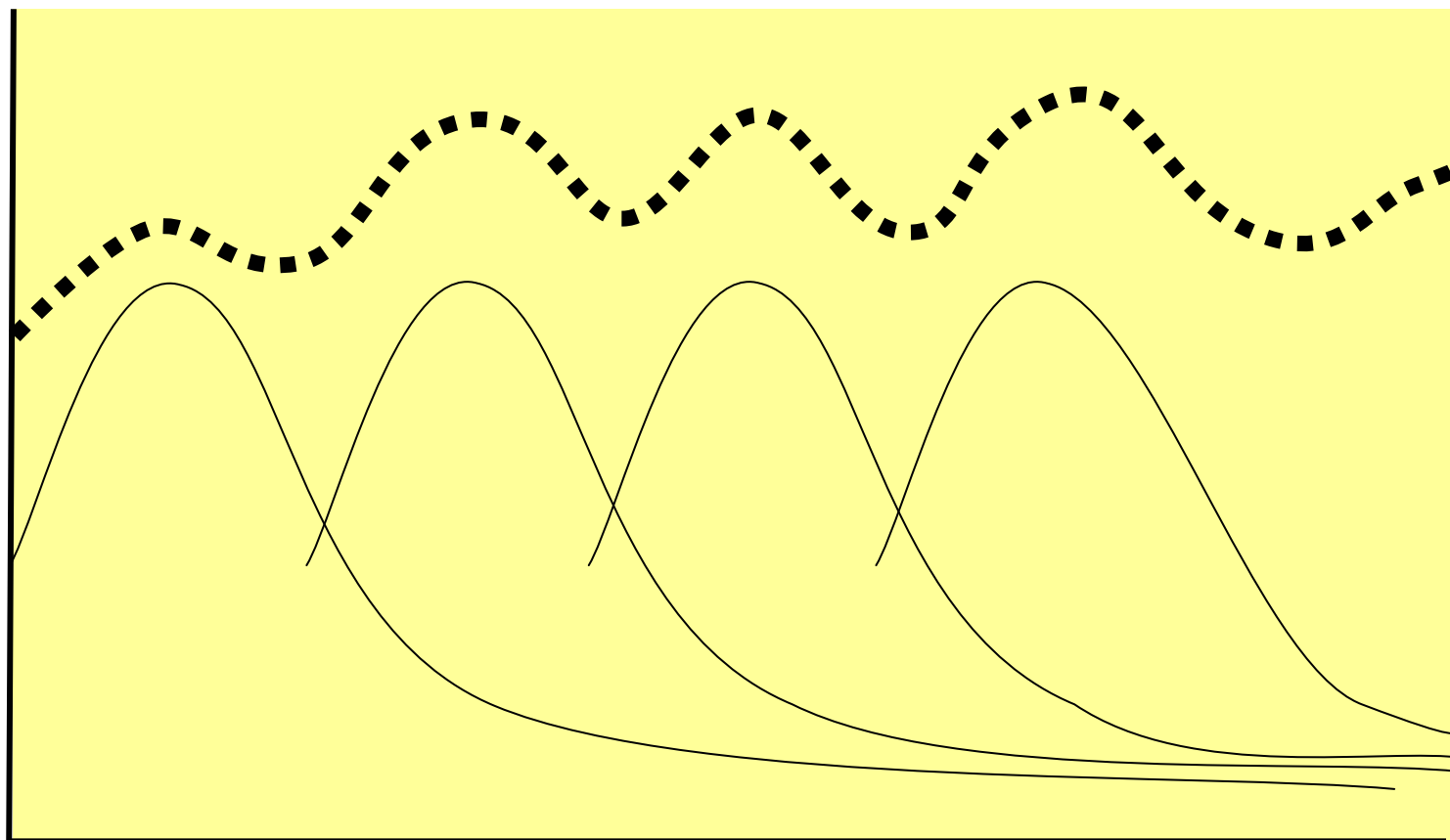


負荷の不均衡

# サーバの集約

## ● 複数のサーバを一つの物理資源で提供

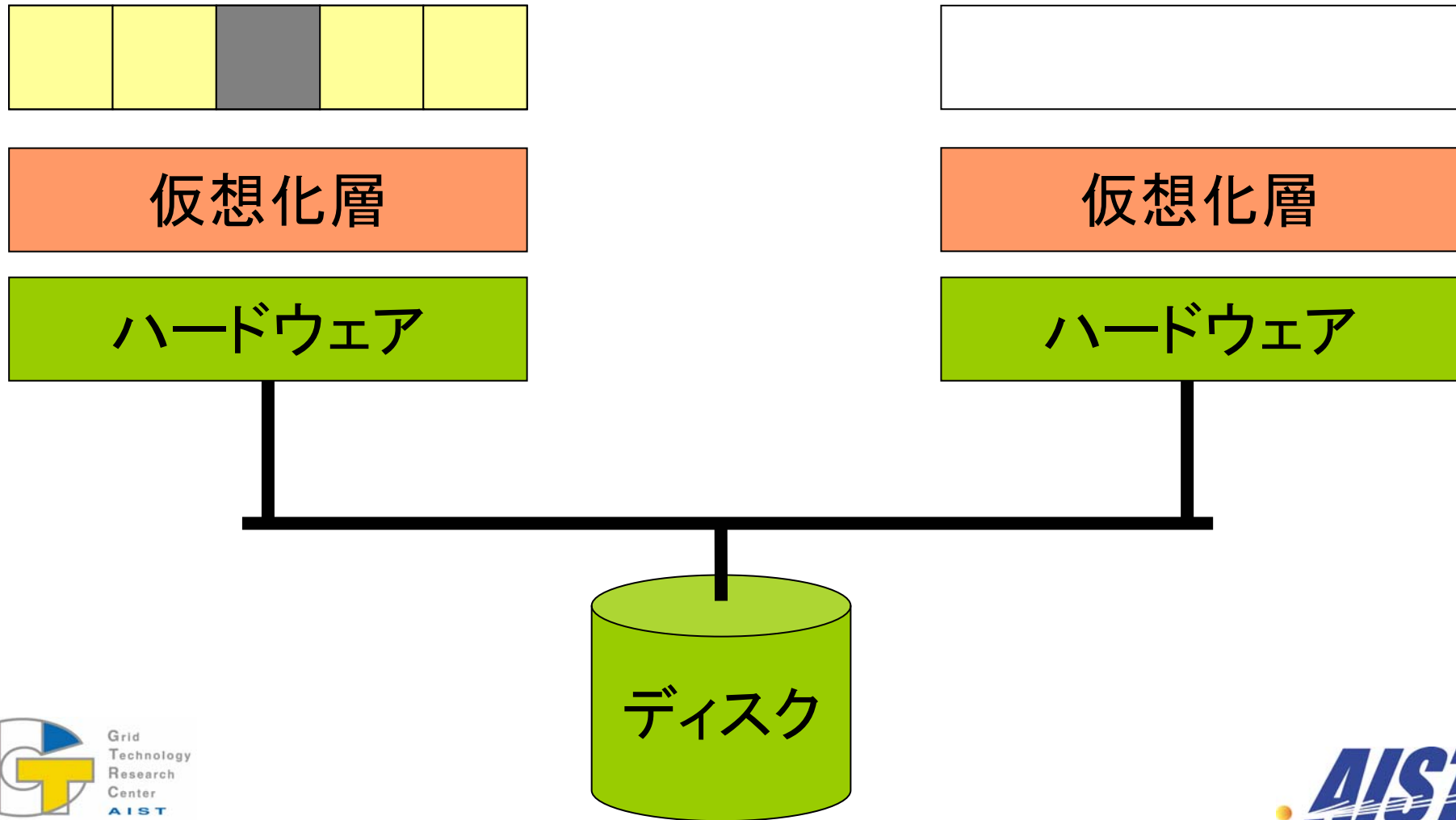
- ▶ 負荷の平準化
- ▶ スループットを維持したままHWコストの低減を実現



# ライブマイグレーション

- 稼働中の仮想計算機上のシステムを別の計算機に稼働したまま移動
- 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

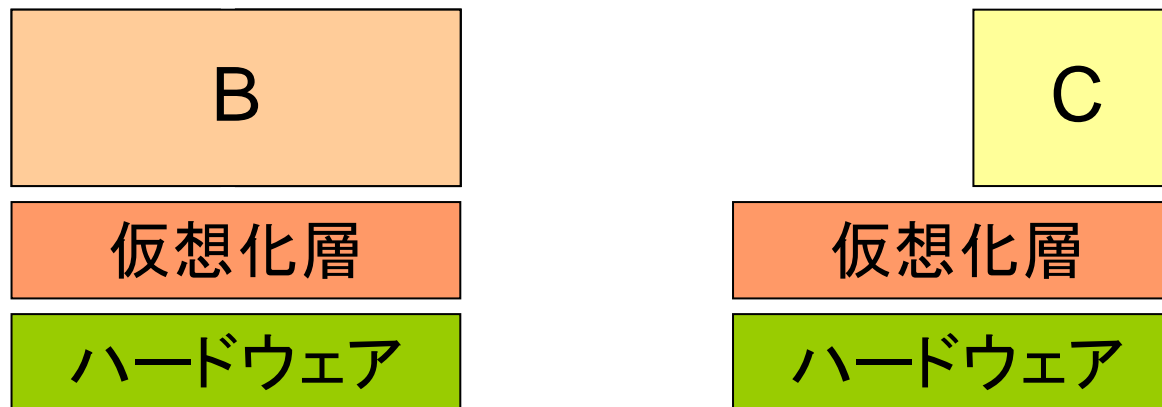
仮想化層

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

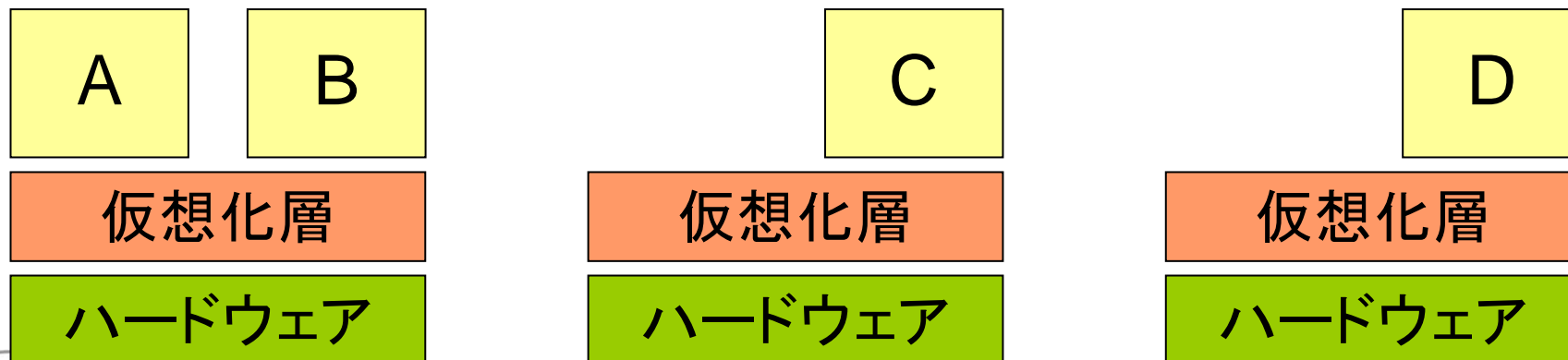


# ライブマイグレーションの適用例

## ● 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



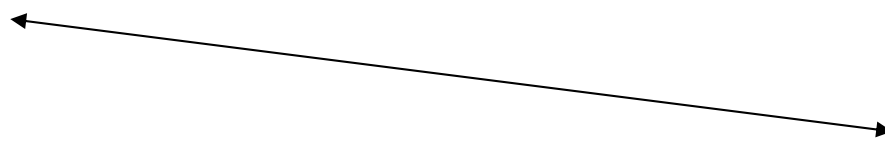
# シンククライアントのバックエンドとして

## ● 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

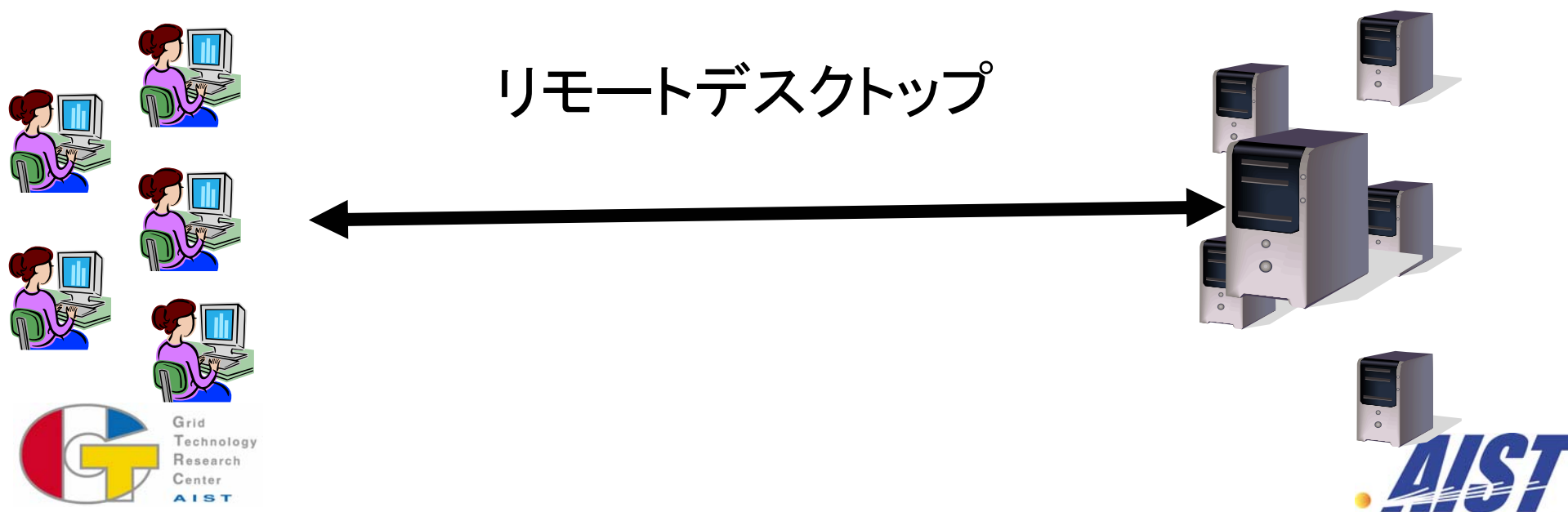


リモートデスクトップ



# 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

---

## ● 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

## ● 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

# 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](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

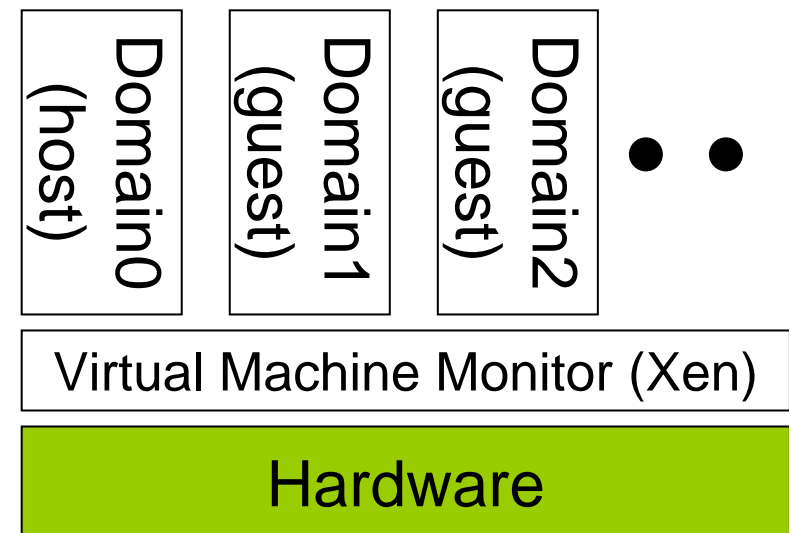
➡ ▶ VMware Player

▶ 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



# Microsoft Windows Server Virtualization

---

## ● Windows Server 2008 (Longhorn) で仮想化はOS標準の組み込み機能に

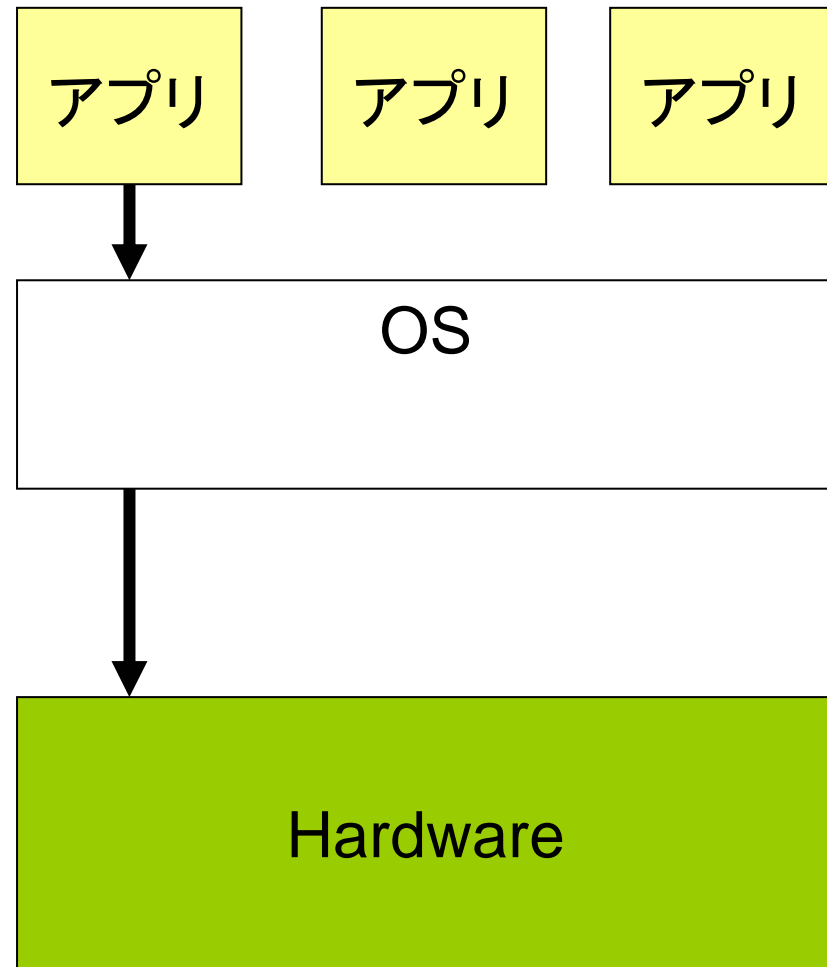
- ▶ VMware Server ESX キラー？
- ▶ Windows Server Virtualization (Viridian)
- ▶ Windows Server 2008 の出荷後180日以内に提供
- ▶ WS2008の出荷は2007年末
- ▶ ただし...
  - Ⓜ Live Migration が初期のバージョンにはない

## ● クライアント側 – Virtual PC 2007

- ▶ 無償提供
- ▶ ゲストとしてWindows98以降をサポート

# 計算機仮想化

- 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



# 計算機仮想化技術の分類

## ● 計算機仮想化手法

### ▶ 完全仮想化 (Full Virtualization)

- ◎ ハードウェアを含め、計算機全体を完全に仮想化
- ◎ ゲストOSの変更不要 - 何でも動く
- ◎ 2つの方法
  - ⊕ コード変換 (Binary Translation)
  - ⊕ CPUのハードウェアサポートを利用

### ▶ 準仮想化 (Para-virtualization)

- ◎ ゲストOSを変更
- ◎ ハードウェアをエミュレートするわけではない。

## ● OSとの関係

### ▶ OSの上 - ホストOS型

### ▶ OSの下 - Hypervisor型

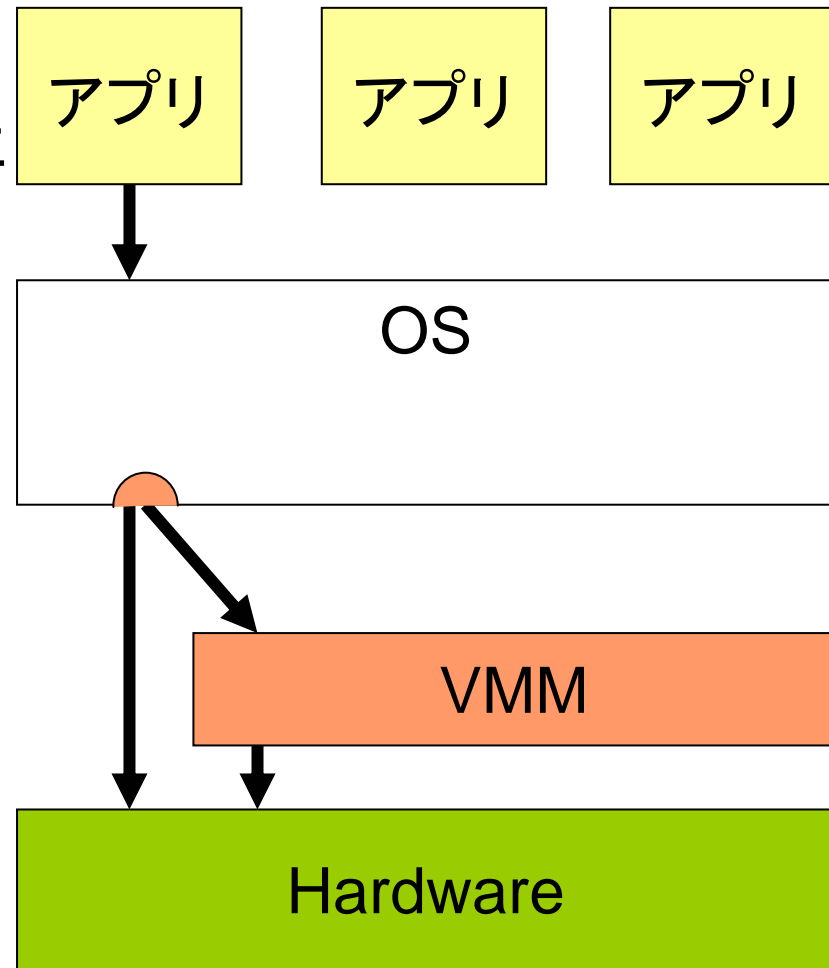
# 完全仮想化

---

- **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)

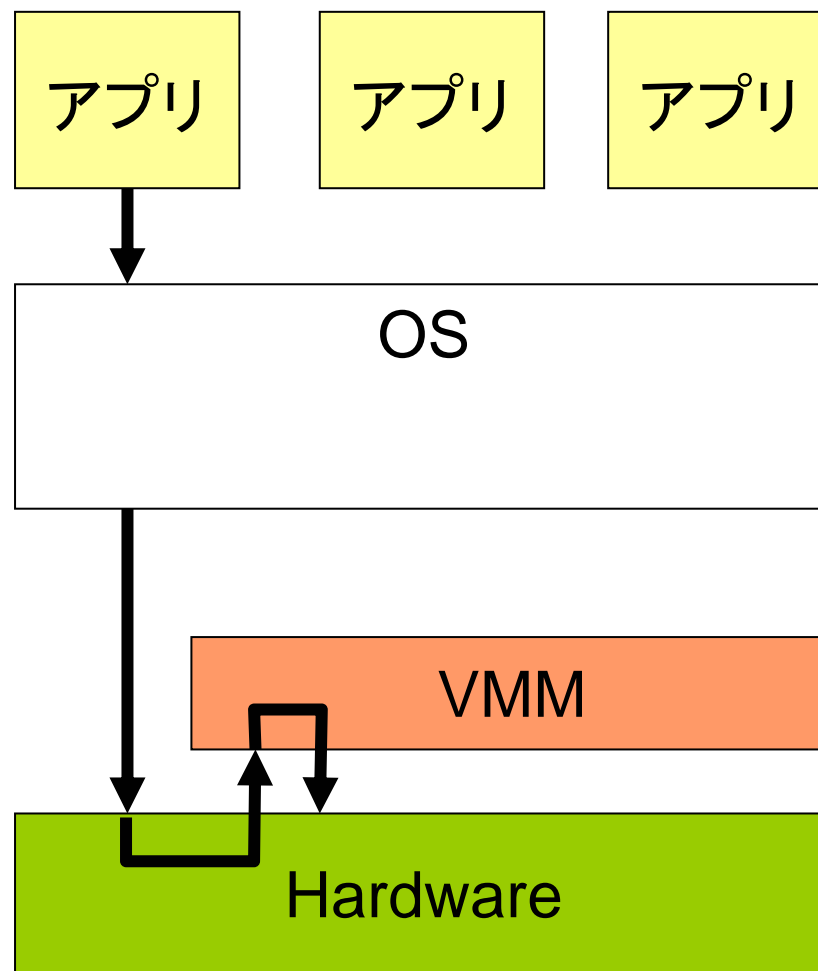
- ▶ CoreDuoやAM2ソケットのAthlonでサポート
- ▶ 相互に互換性無し
- ▶ 新たに仮想計算機用の実行モードを追加

◎ 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。



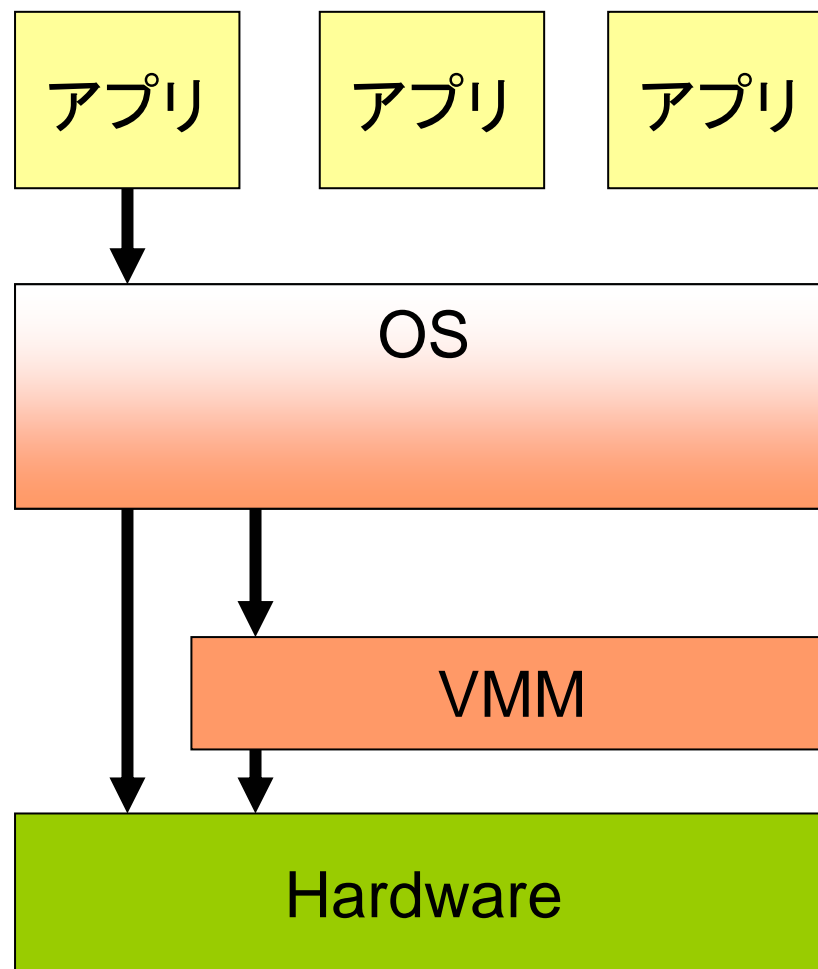
# 準仮想化

## ● 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



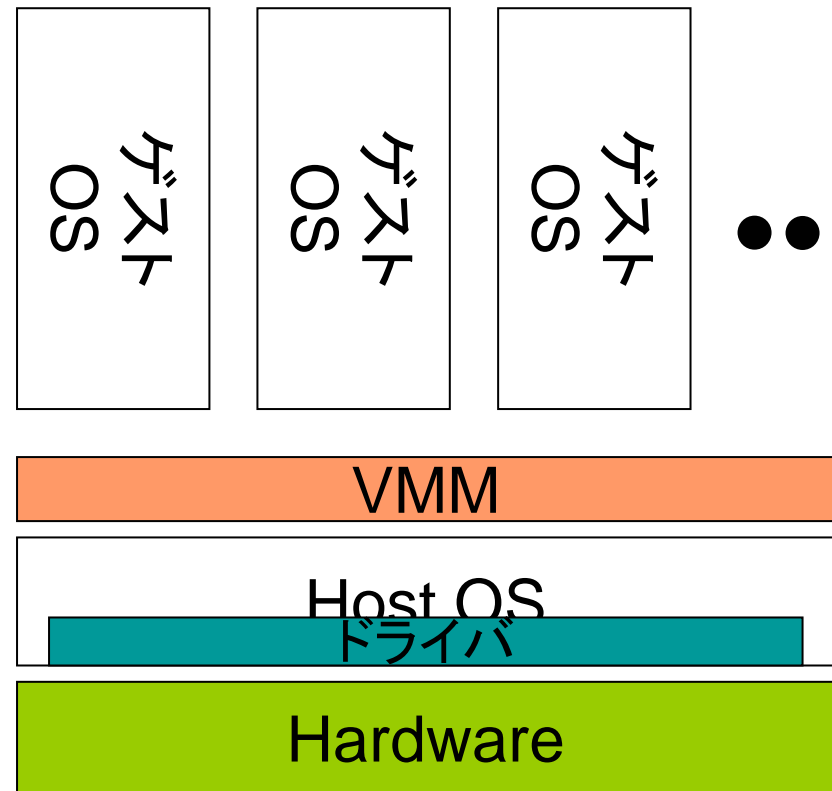
# ホスト型

- 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

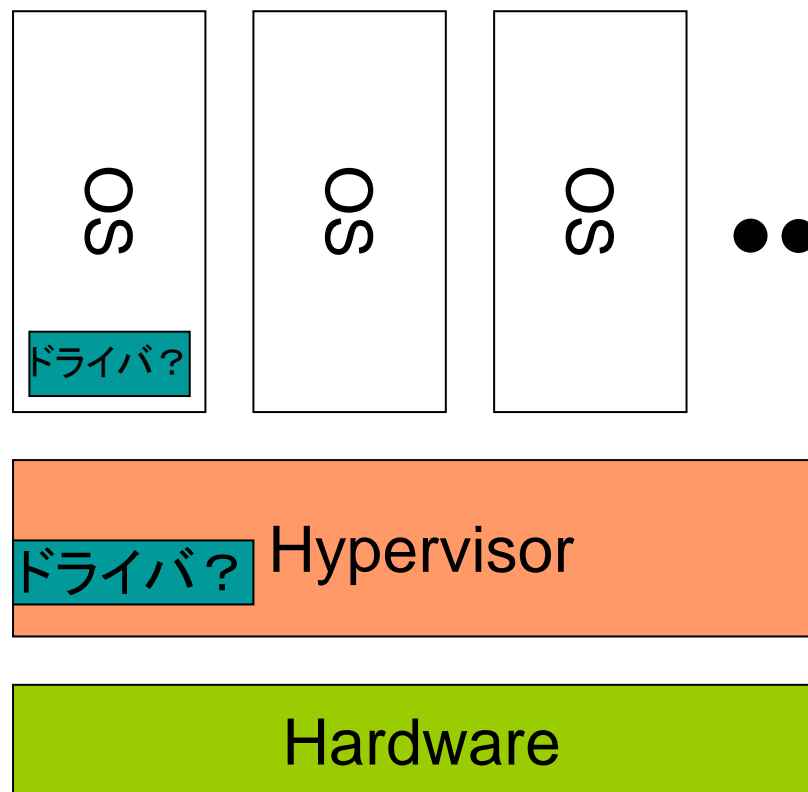
- デバイスドライバはホストOSが提供

- ▶ 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？



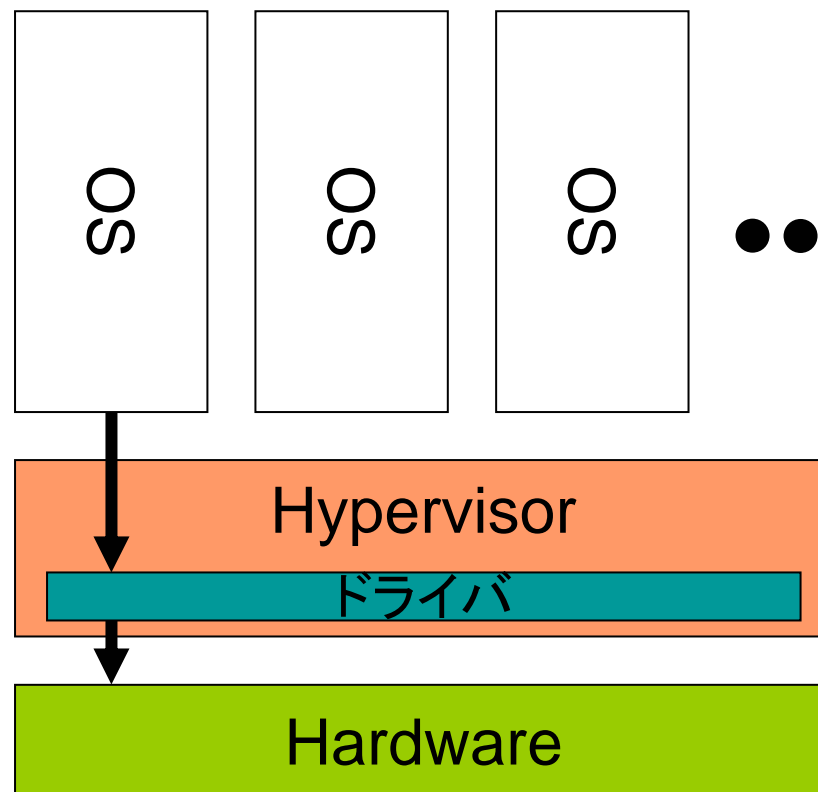
# Hypervisor 型

## ● 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

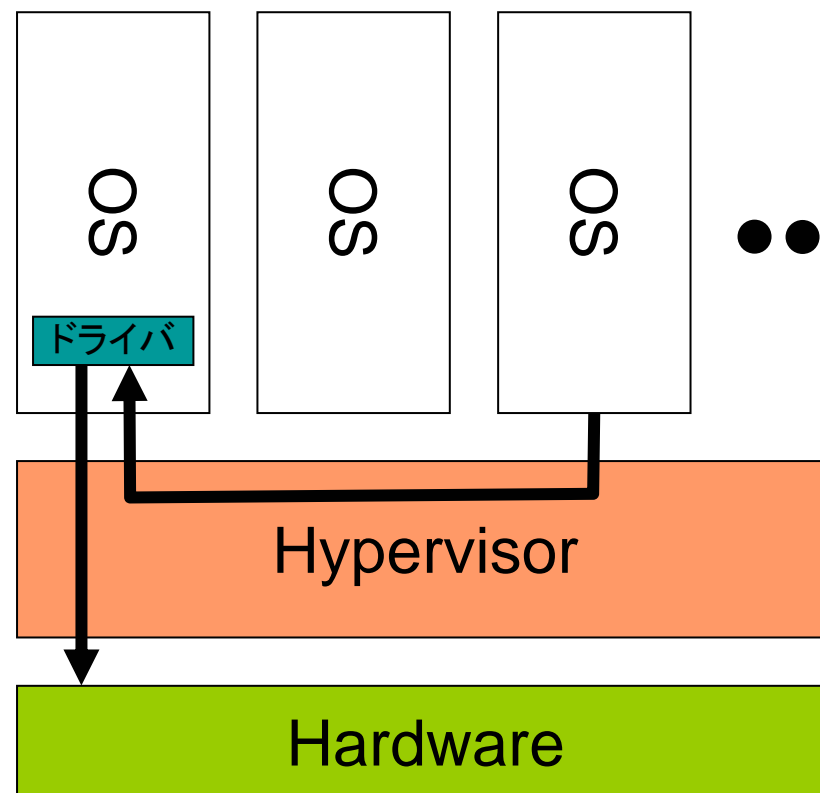
## ● 例

- ▶ 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)



# 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		Parallels VMware WS 他		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	

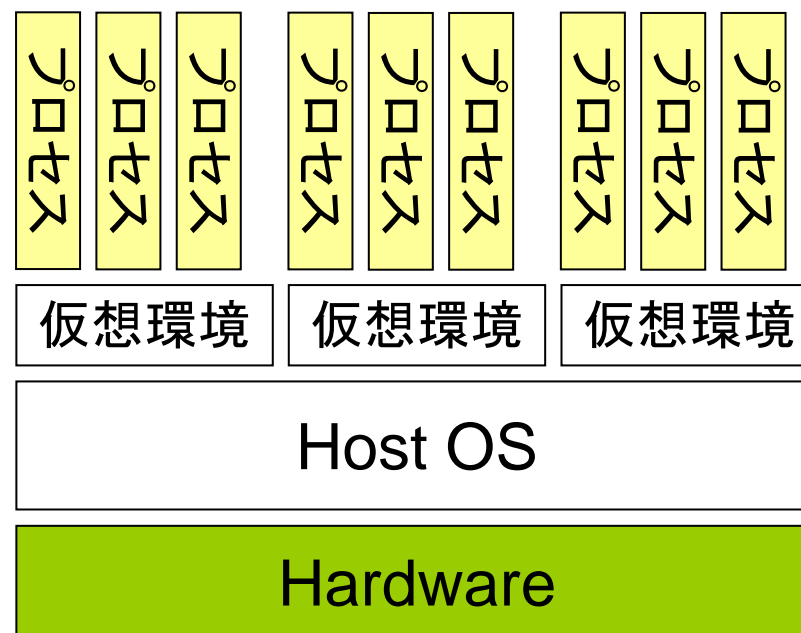
# OS仮想化

## ● 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

---

# 計算機以外の仮想化

# ストレージの仮想化

---

## SAN (Storage Area Network)

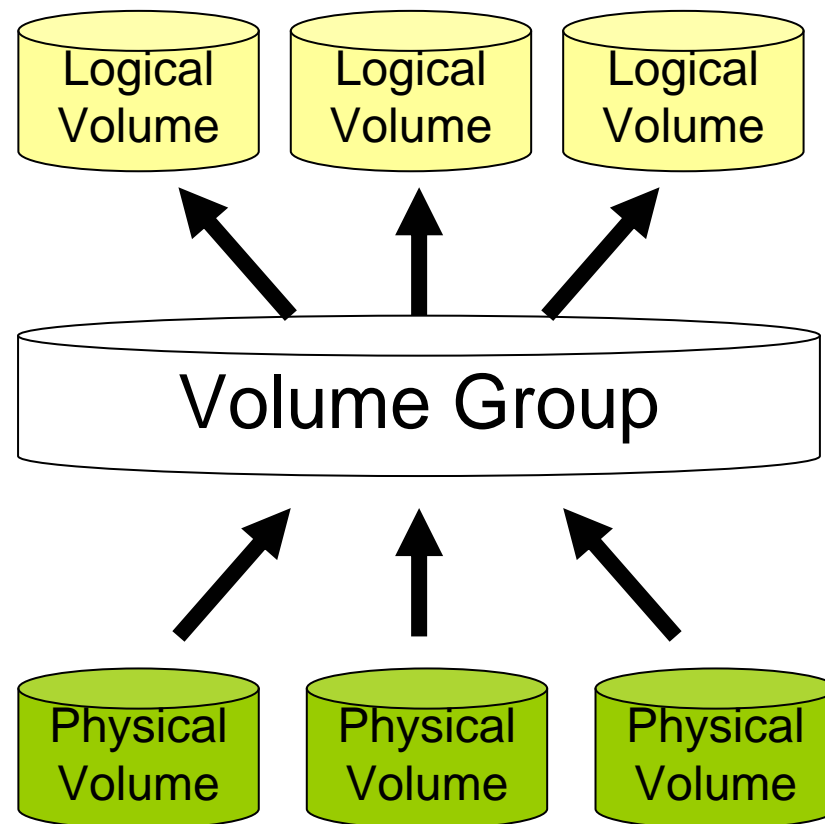
- ▶ ディスクの設置場所の透明化

## LVM (Logical Volume Manager)

- ▶ 物理ディスクの容量からの透明化

# LVM ( Logical Volume Manager )

- 物理ボリューム(ディスク)を一度ボリュームグループに集約, そこから論理ボリュームを切り出す
- 論理ボリュームのサイズはディスクのサイズに非依存
- 物理ディスク間の負荷バランス
- 物理ディスクの追加, 削除が可能



# ネットワークの仮想化

---

## VPN (Virtual Private Network)

- ▶ 通常のネットワーク上にまったく異なるネットワークを構築
- ▶ 暗号化可能・サブネットをまたいで構築可能
- ▶ ex. ソフトイーサなど

## VLAN (Virtual LAN)

- ▶ パケットにタグをつけておき、インターフェイスで選別
- ▶ サブネットをまたぐことはできない。
- ▶ オーバヘッド小

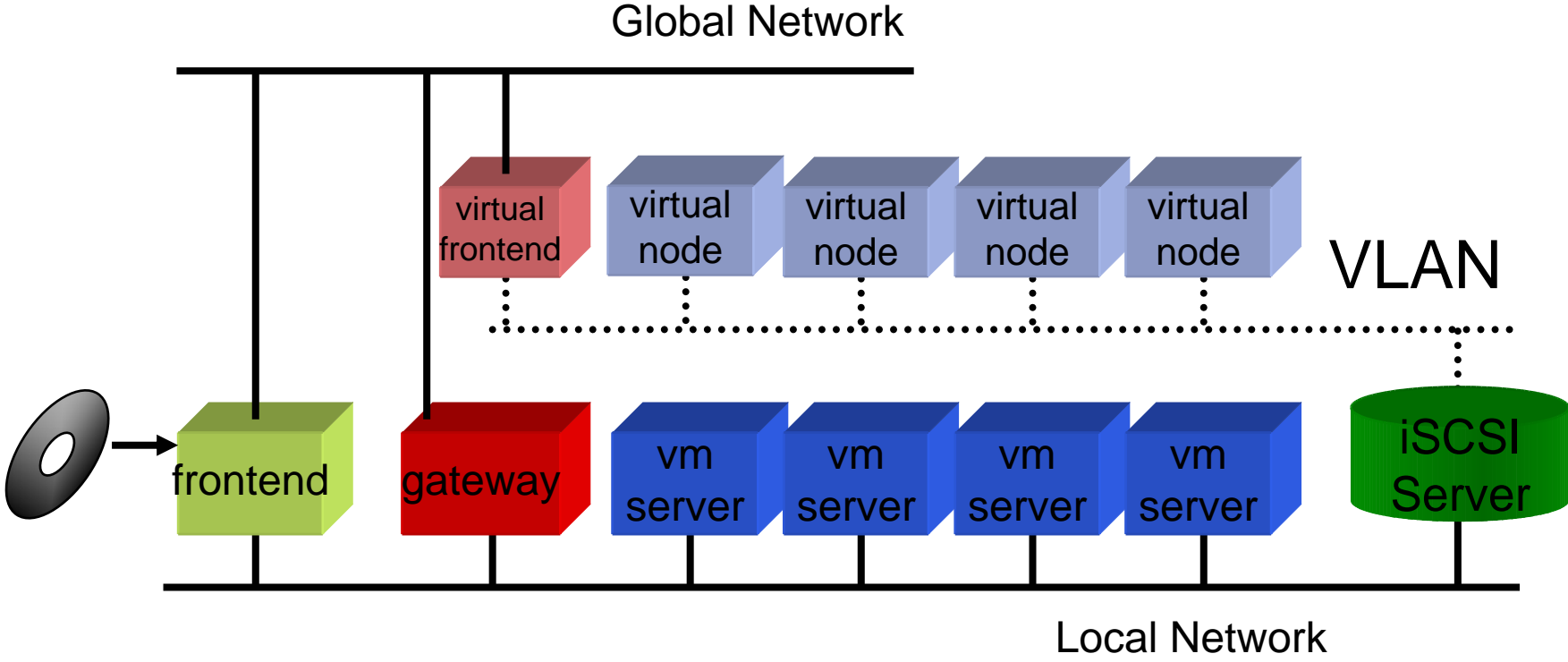
# クラスタの仮想化

---

## ● 産総研の仮想クラスタ管理システム

- ▶ 計算機, ネットワーク, ストレージを仮想化
  - ◎ VMware Server, VLAN, LVM+ iSCSI を利用
- ▶ 構成した計算機群をクラスタ管理システムRocksを用いてインストール
- ▶ 予約ベースでクラスタソフトウェアがインストールされた仮想クラスタを提供

# 仮想クラスタインストール



---

おわりに



# 今後の展望

---

## ● ハードウェアによる仮想化支援

- ▶ デバイス側の対応によりI/Oの速度も遜色なくなる
- ▶ サーバではごく当たり前の機能に
- ▶ クライアントでの普及は？
  - Ⓜ Windows Vista を快適に使うには メモリ2G必要
  - Ⓜ Web 2.0 系のサービスを使うならクライアントはなんでもいい

## ● 準仮想化の標準インターフェイス「paravirt-ops」

- ▶ やっぱり準仮想化は速い
  - Ⓜ Xenが利用
  - Ⓜ VMware も最新版workstationでサポート
- ▶ 標準インターフェイスに対応したゲストであれば、どの仮想化ソフトでも実行できる。

# ソフトウェアライセンスの問題

---

- OS・アプリケーションのコピーが簡単に無数にできてしまう
  - ▶ アクティベーションも無意味
    - ◎ OSが認識するハードウェア自体がコピーされるため
- ホストに対する束縛も難しい
  - ▶ フローティングライセンスが必須
- 従量制と組み合わせるなど新しい枠組みが必要なのかも

# 「仮想化ソフトは消えていく？」

- 日経 Linux 畑 陽一郎 氏
- 仮想化がさまざまなレイヤで定着することにより、逆に仮想化「ソフト」が見えなくなっていく
- 仮想化ソフト単体ではなく、仮想化したアプリケーションの管理ツールが主力商品に？
  - ▶ ex. VMware Infrastructure, Virtual Iron の Virtualization Manager
- 一般的な技術として、サーバ側で利用され、ユーザはそれを意識することはない
  - ▶ コストの低減としてだけ見える

