

# 高性能計算基盤

自習 APDX06:バーチャルマシンの仕組み

<http://arch.naist.jp/Lectures/ARCH/x06/apdx06j.pdf>

Copyright © 2021 奈良先端大 中島康彦

# 1つのハードウェア上で複数のOSを動かす理由

## OS動作環境をサービスとして売る側

- 必要最低限の設備投資で最大限のサービス料を獲得したい
- ユーザの要求は予測不能. 余裕をもった設備投資が必要
- 必要な時に立ち上げ可能な仮想マシンを利用すれば  
過剰設備投資を抑制可能
- 規模が大きくなるほど需要の変動が小
- 設備の利用効率が上がり有利




## サービスを買う側

- OSごとにハードを準備する必要がない
- 使うときだけ支払えばよい
- 一次的な設備増強もお金で解決可能

# バーチャルマシン(仮想計算機)とは

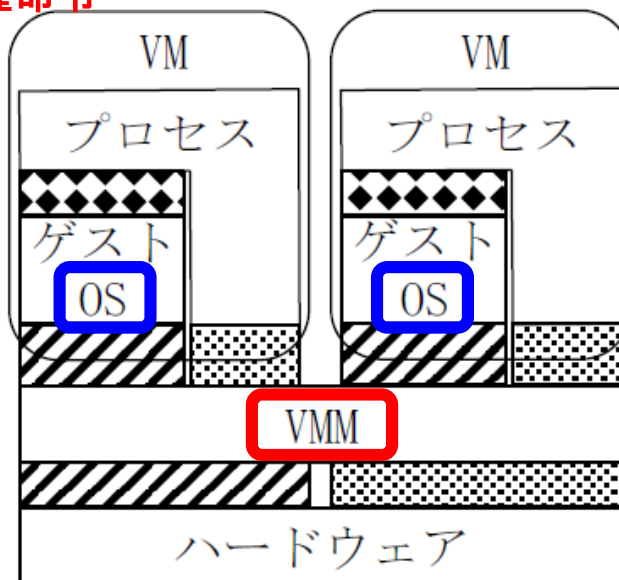
ハードウェアから切り離れた環境でソフトウェアを動かす仕組み

- アプリケーションソフトウェアの場合はOSが介在
- OSの場合はVMM(仮想計算機モニタ)が介在

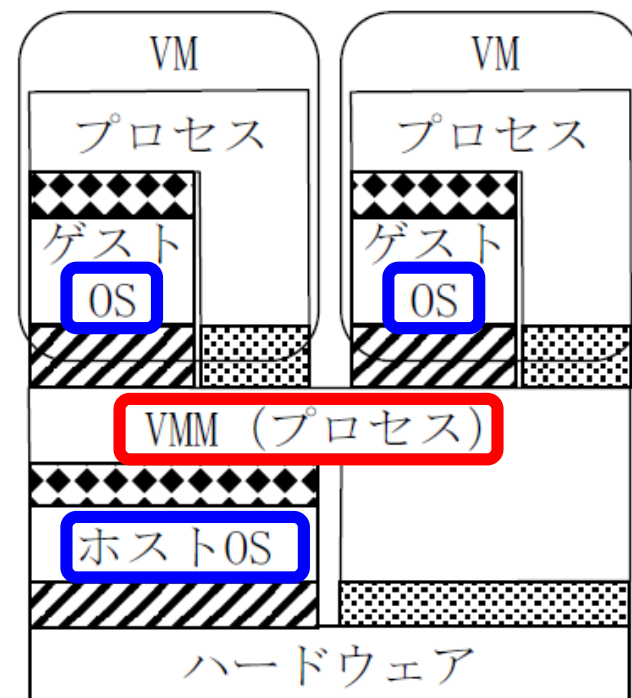
 : ユーザISA 一般命令  
 : システムISA 特権命令  
 : ABI システムコール



(a) OS



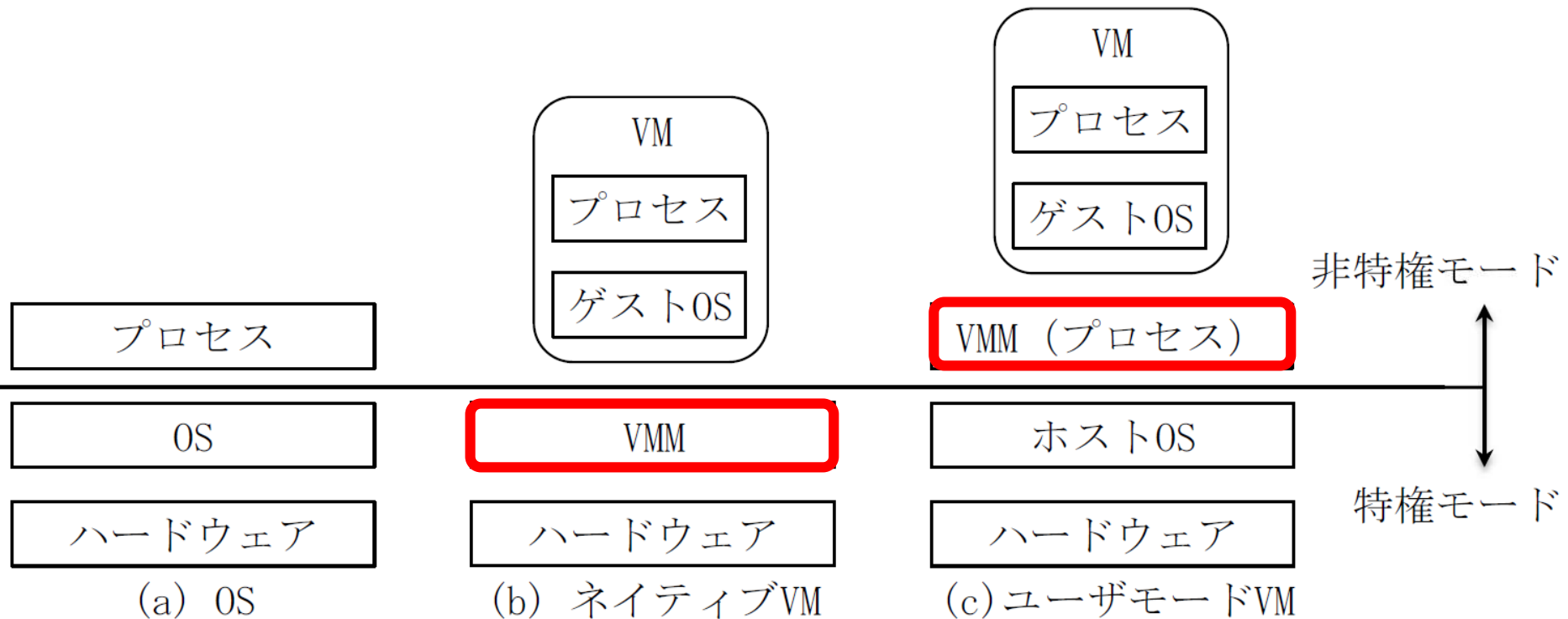
(b) Type I VMM



(c) Type II VMM

# OSの特権命令を通常モードで実行させる

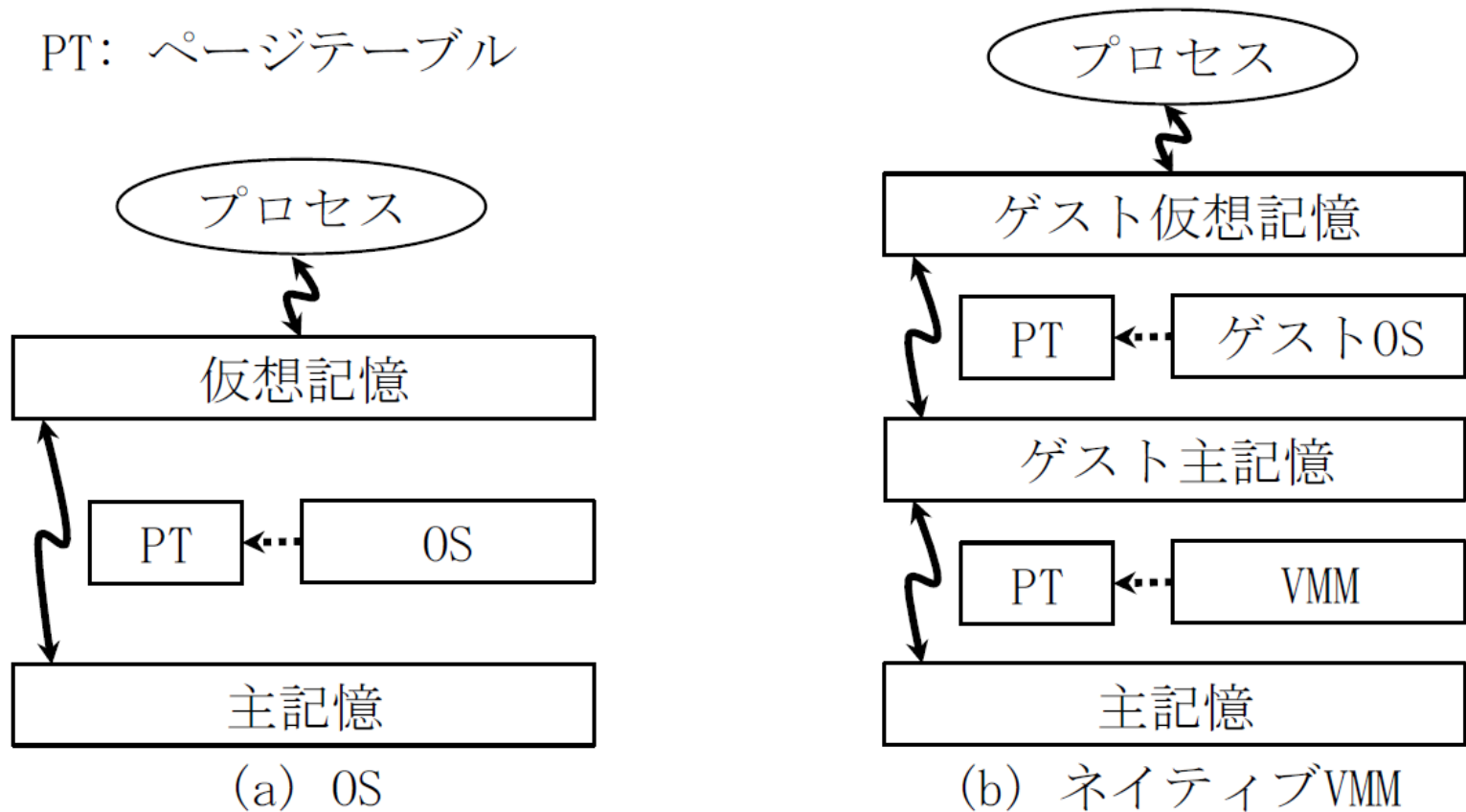
OSは特権命令を発行するが、これをVMMが横取りするために、OSをアプリケーションプログラム(非特権モード)として実行



# メモリ管理

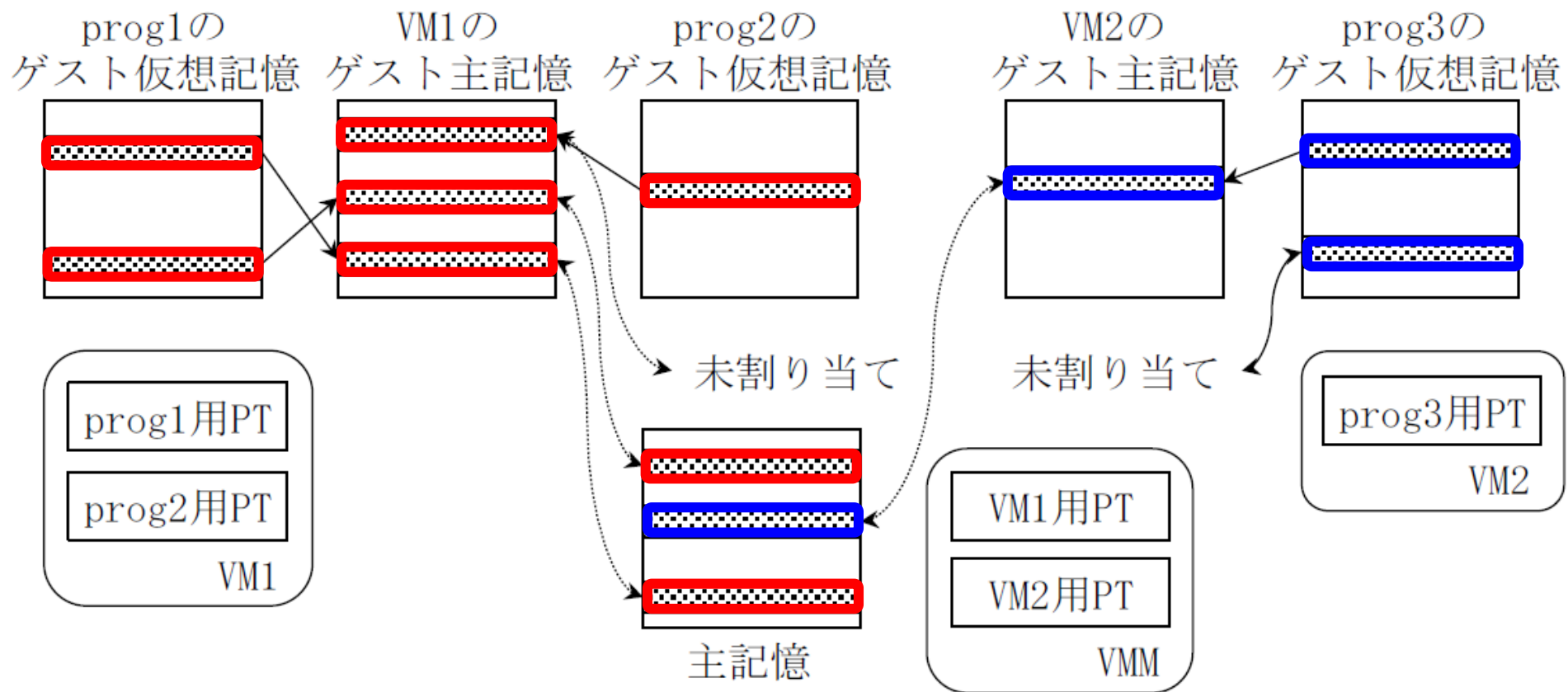
プロセスの先頭アドレスは常に0番地  
OSはプロセスごとにアドレス変換表を用意  
同様に、VMMはOSごとにアドレス変換表を用意

PT: ページテーブル

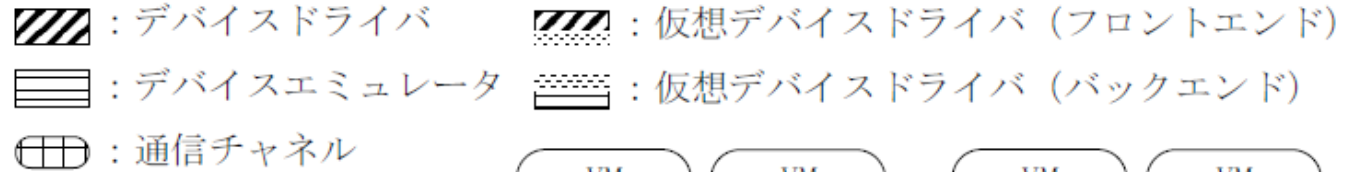


# メモリ管理

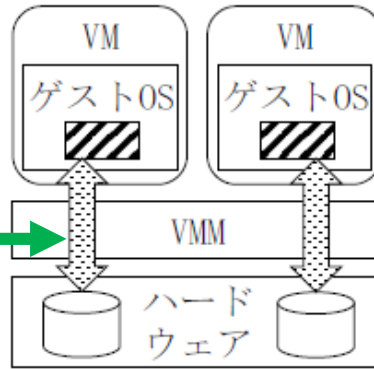
プロセスの空間、OSの空間が、主記憶上では重複しないよう制御する



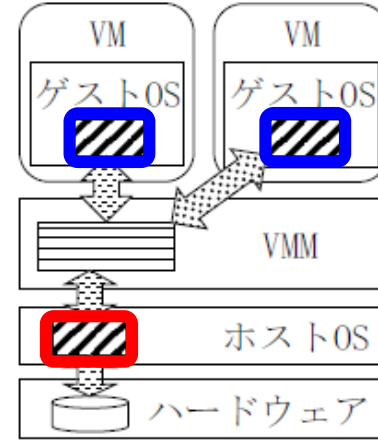
# デバイス管理



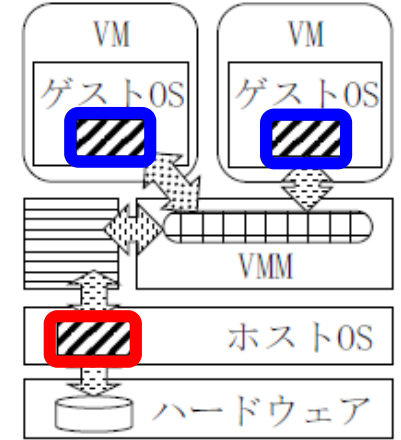
直接制御の場合は  
アドレス変換等  
追加ハードウェアが必要



(a) 直接割当



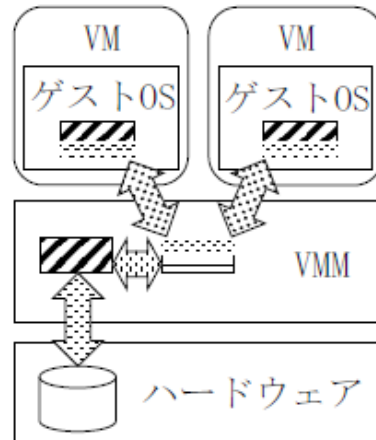
(b-1) デバイスエミュレーション (VMM内)



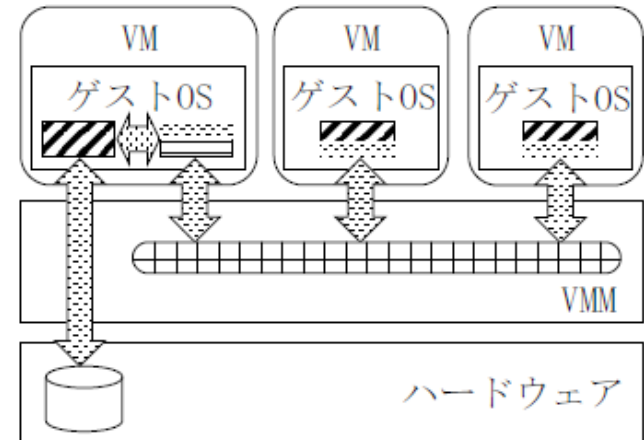
(b-2) デバイスエミュレーション (別プロセス)

ゲストOSのデバイスは旧式を使用

ホストOSのデバイスは最新を使用

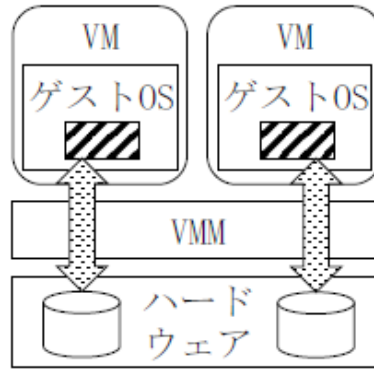
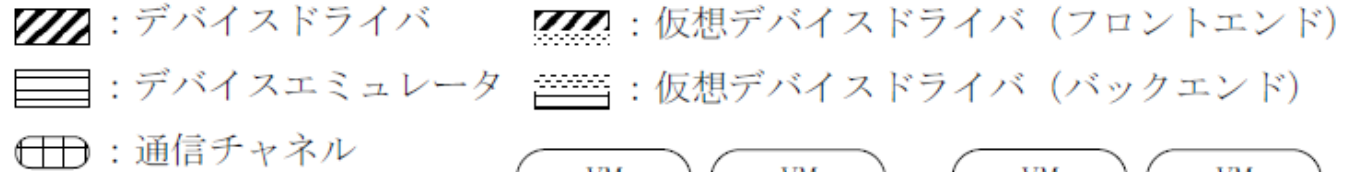


(c-1) 仮想デバイスドライバ (VMM内)

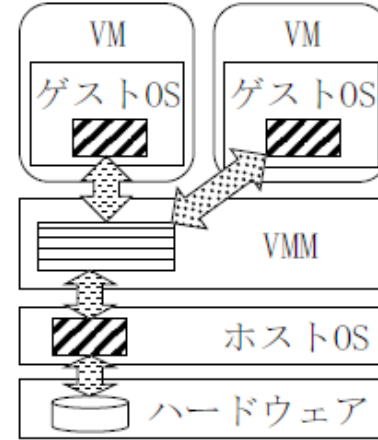


(c-2) 仮想デバイスドライバ (特定VMM)

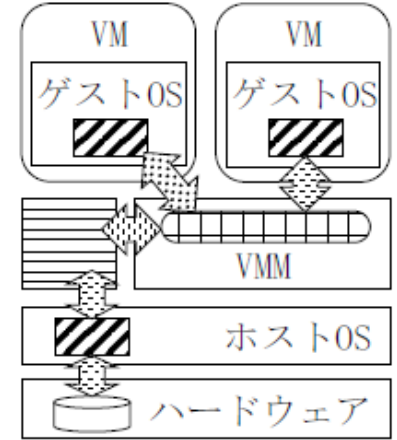
# 演習問題



(a) 直接割当

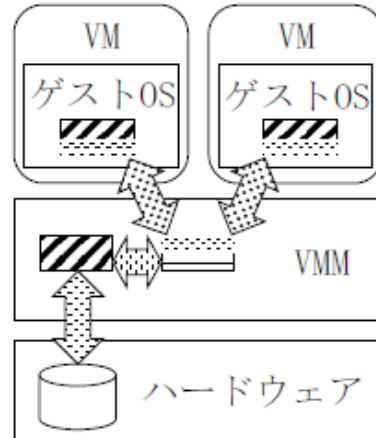


(b-1) デバイスエミュレーション (VMM内)

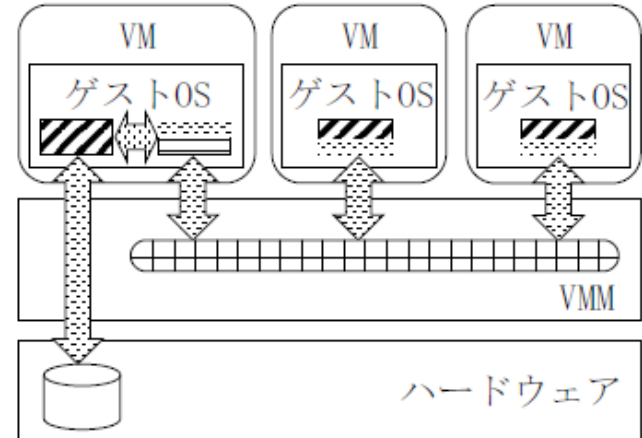


(b-2) デバイスエミュレーション (別プロセス)

●(a) 直接ゲストOS がデバイスをアクセスする場合と、(b-1) ホストOS を介してアクセスする場合の得失は？



(c-1) 仮想デバイスドライバ (VMM内)



(c-2) 仮想デバイスドライバ (特定VMM)



今日はここまで