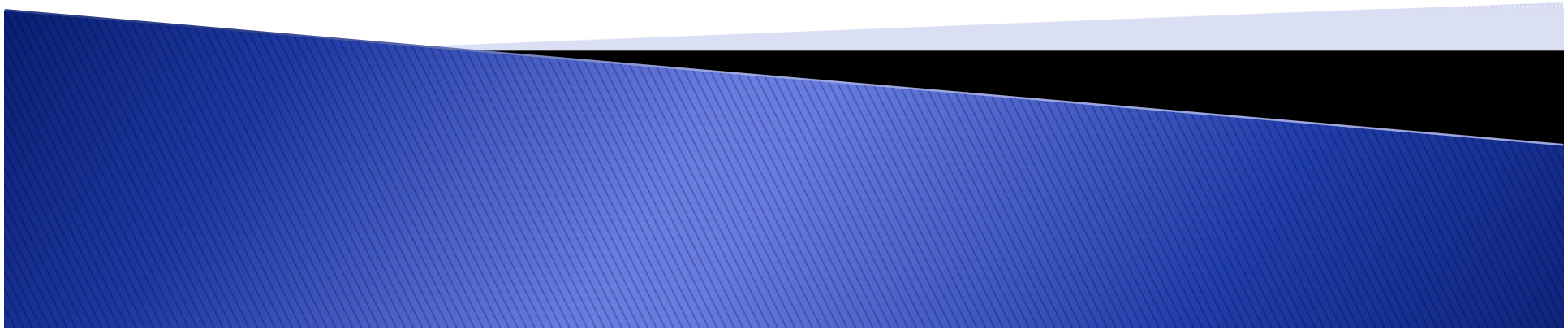
BitVisor 用のストレージスナップショット機能の実装

○奥野航平，大月勇人，瀧本栄二，毛利公一
立命館大学



スナップショット

- VM の状態を保存・復元する機能
- 一瞬で復元が可能

- スナップショットが無い場合
  - ディスクのイメージファイルを書き戻す作業が必要
    →時間が掛かる

- Alkanet において
  - マルウェアに環境を改変される
  - 元に戻す作業が必要
  - もっと高速にしたい！

Snapshot BitVisor

- スナップショット機能を備えた BitVisor
- ストレージに対するスナップショット
  - 再起動時にストレージを初期状態へ復元できる

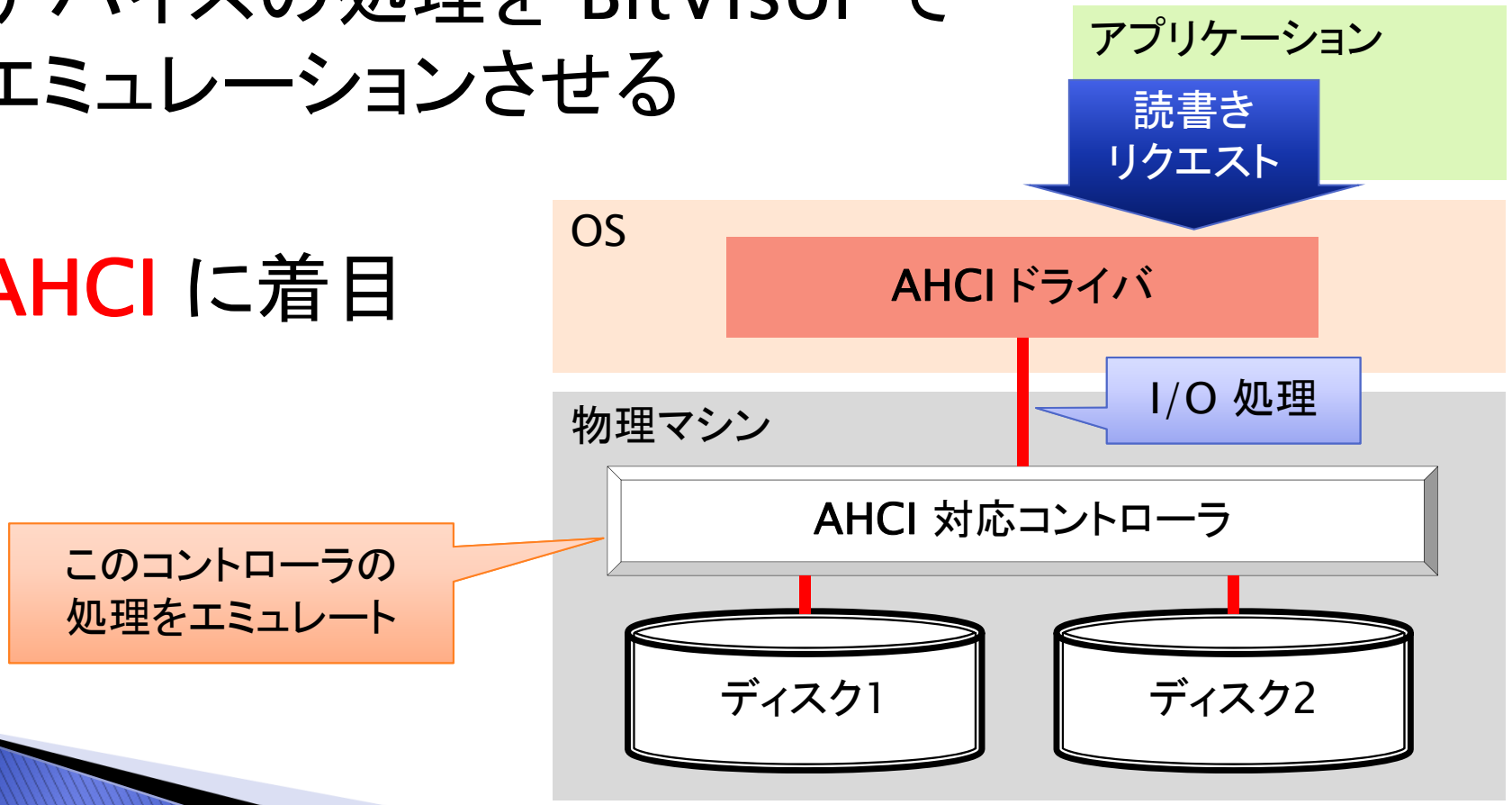
- 利用例
  - Alkanet でマルウェア実行前の環境へ復元する
  - 共有端末や実験環境などで設定を復元する
    - 準パススルーによってハードウェアを物理マシンと同様に扱える

目標

- ストレージのデータを瞬時に復元できる
- 最小限のエミュレーションで実現する
  - VM を使用しているということを意識させない
    - Alkanet での VM 検知対策
  - 仮想化特有のエミュレーションされたデバイスとして見せない
- 簡単な実装で実現する
  - BitVisor の変更量を少なく
  - 軽量な VMM を維持する

ストレージへのアクセスの流れ

- ドライバを経由してストレージへアクセスする
- デバイスの処理を BitVisor で エミュレーションさせる
- AHCI に着目

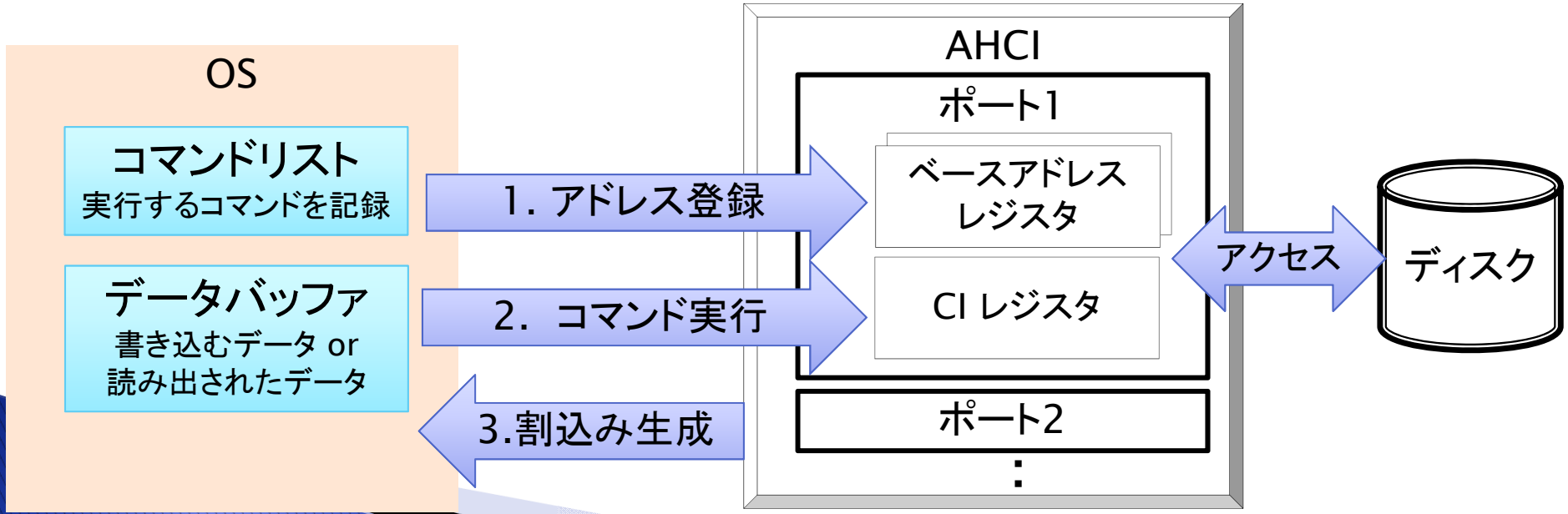


AHCI

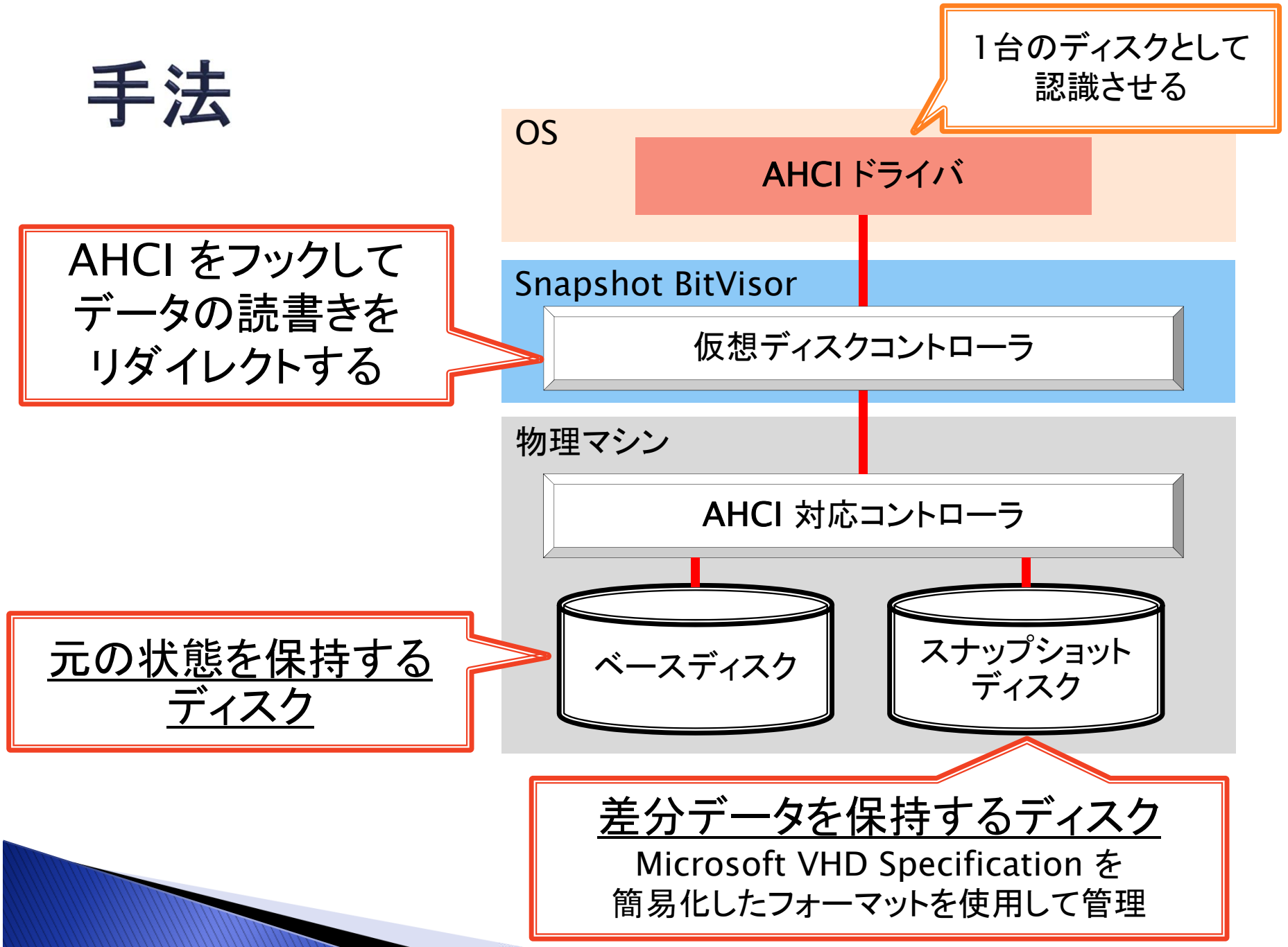
- Advanced Host Controller Interface
  - Serial ATA に特化したコントローラと
    ホストコンピュータとのインタフェース規格
- PC で幅広く使用されているインタフェース
  - SATA ポートがあるマザーボードには
    必ず実装されている
- AHCI に対応することで
  ほとんどの PC で動作可能になる

AHCI を使用したストレージアクセス

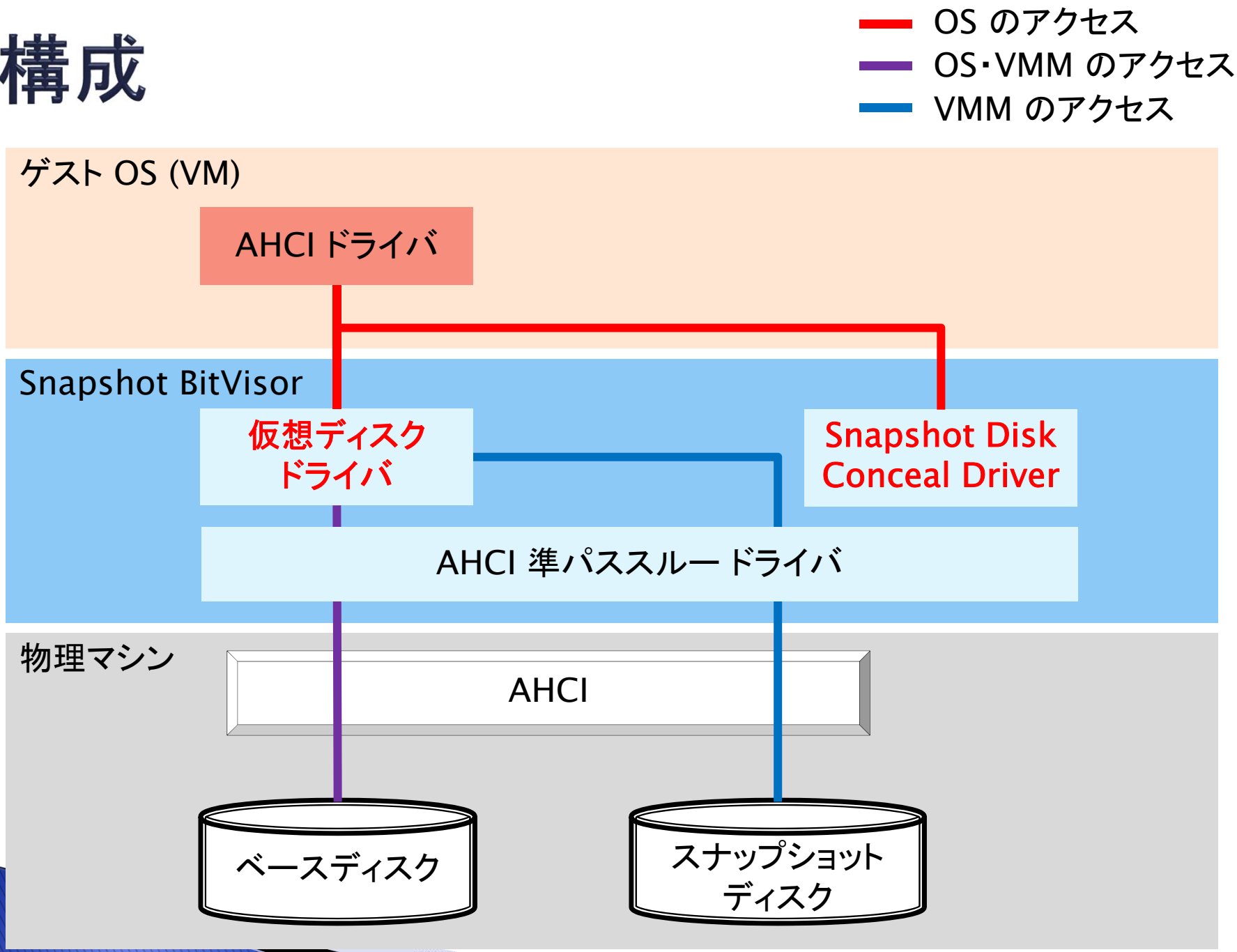
1. DMA 転送に使用するバッファを用意する
   - コマンドリスト と データバッファ の2 種類
   - それぞれベースアドレスレジスタにアドレスをセットする
2. ポートに対してコマンドを発行する
   - キューイング可能で 32 個のコマンドを同時に発行可能
   - コマンドリストのインデックスに対応するビットを CI レジスタにセットする
3. 割込みを受け取る
   - コマンド完了後，コントローラは割込みを発生させる



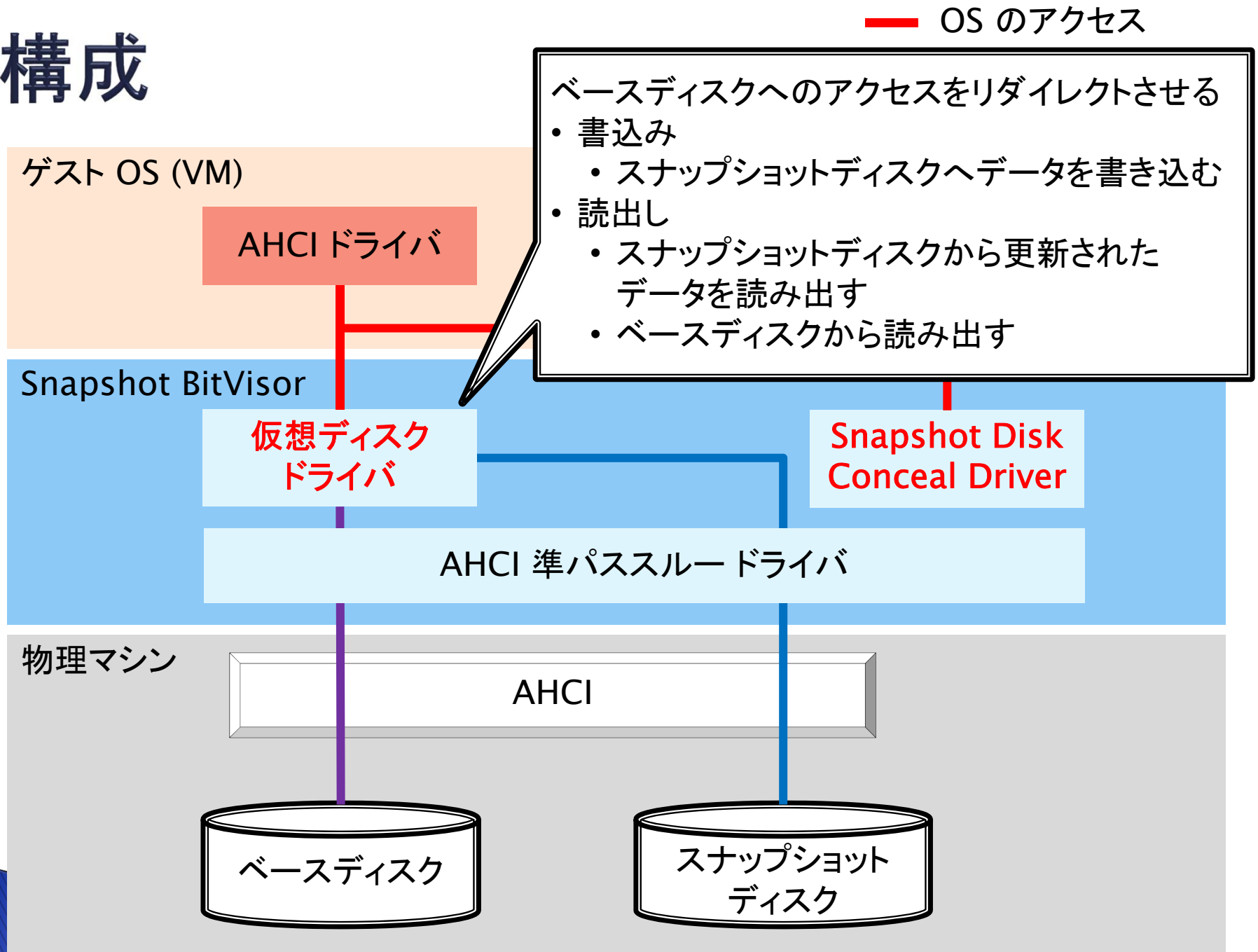
手法


1台のディスクとして
認識させる
OS
AHCI ドライバ
AHCI をフックして
データの読書きを
リダイレクトする
Snapshot BitVisor
仮想ディスクコントローラ
物理マシン
AHCI 対応コントローラ
元の状態を保持する
ディスク
ベースディスク
スナップショット
ディスク
差分データを保持するディスク
Microsoft VHD Specification を
簡易化したフォーマットを使用して管理

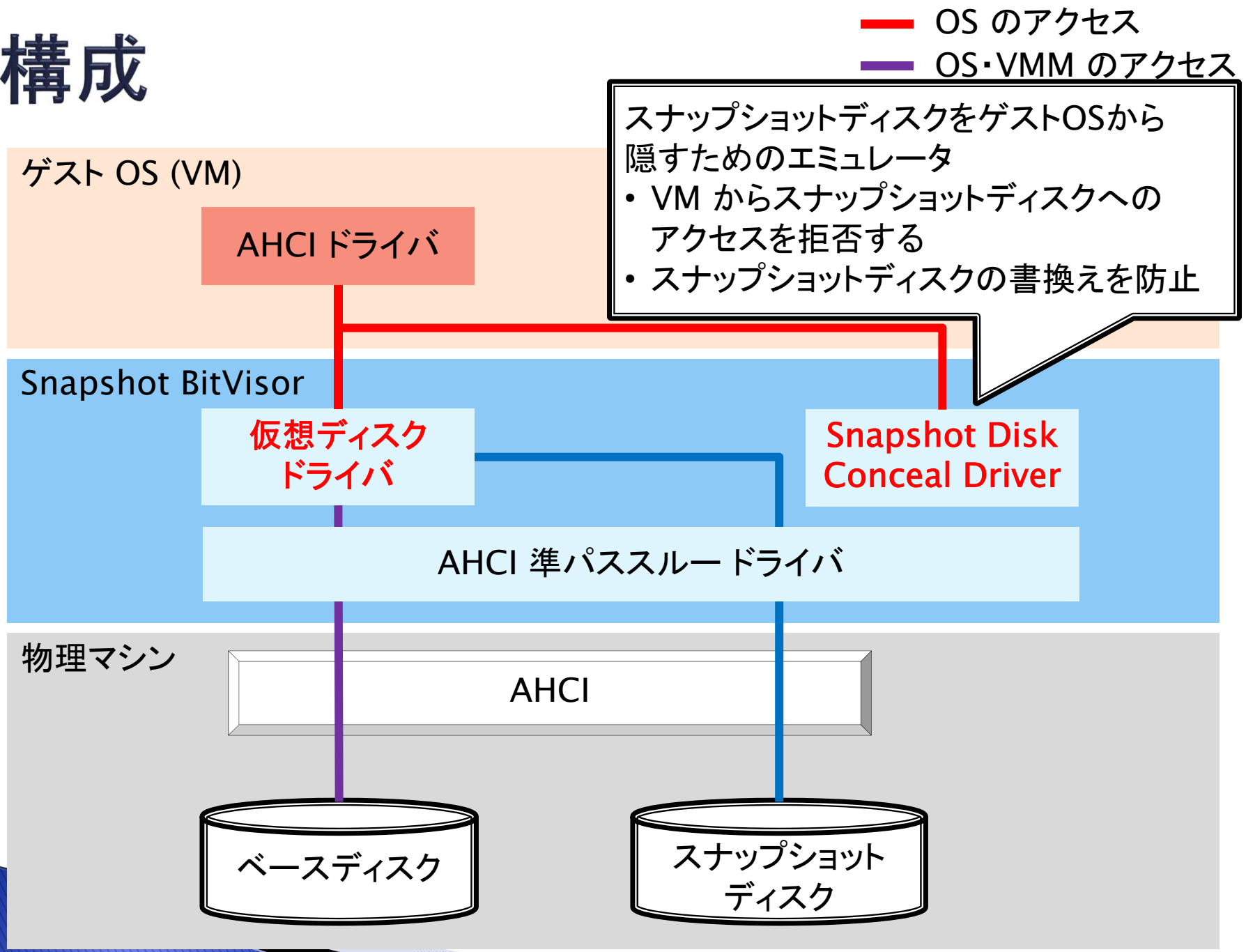
構成


OS のアクセス
OS・VMM のアクセス
VMM のアクセス
ゲスト OS (VM)
AHCI ドライバ
Snapshot BitVisor
仮想ディスク
ドライバ
Snapshot Disk
Conceal Driver
AHCI 準パススルー ドライバ
物理マシン
AHCI
ベースディスク
スナップショット
ディスク

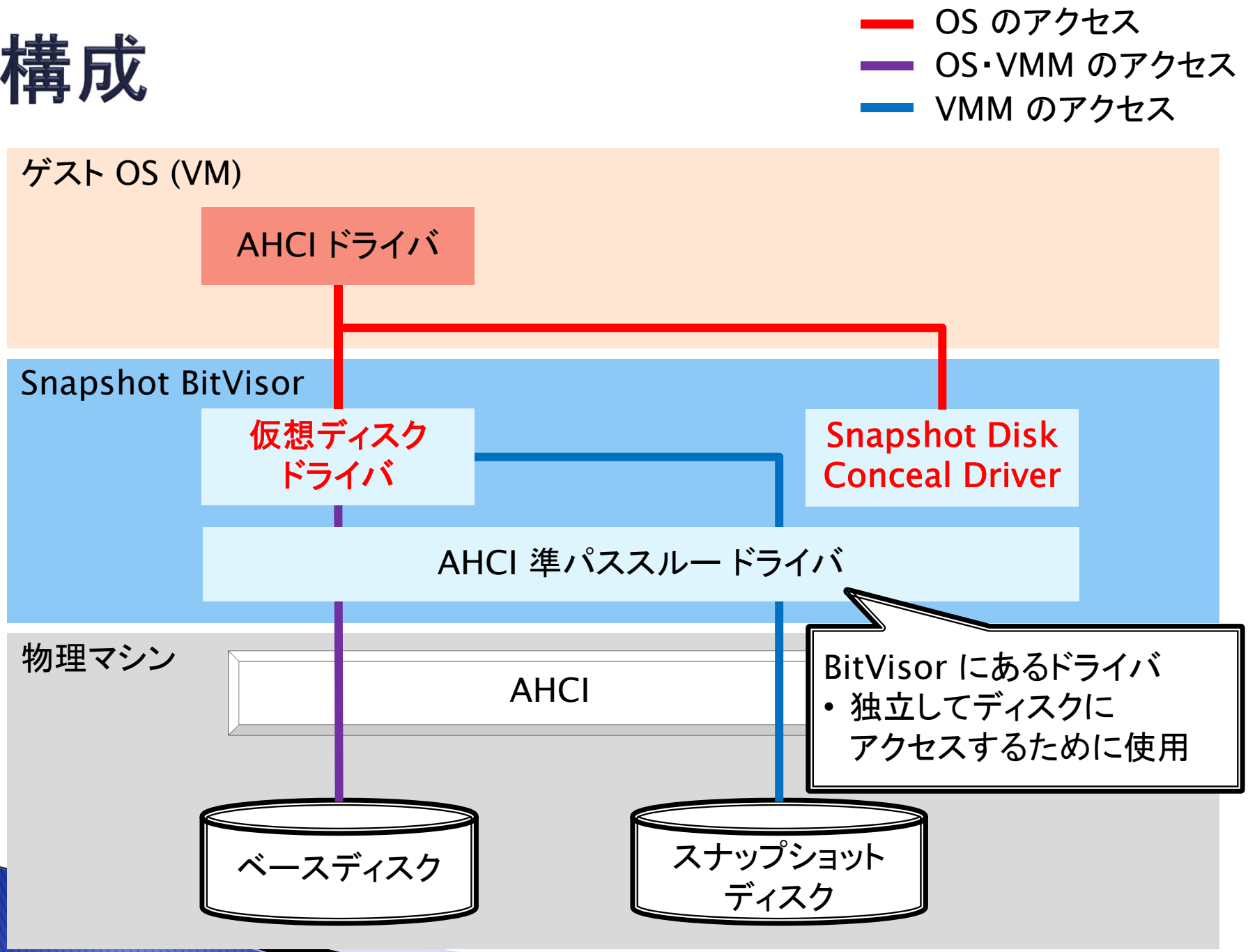
構成


OS のアクセス
ベースディスクへのアクセスをリダイレクトさせる
• 書込み
• スナップショットディスクへデータを書き込む
• 読出し
• スナップショットディスクから更新されたデータを読み出す
• ベースディスクから読み出す
ゲスト OS (VM)
AHCI ドライバ
Snapshot BitVisor
仮想ディスクドライバ
Snapshot Disk Conceal Driver
AHCI 準パススルー ドライバ
物理マシン
AHCI
ベースディスク
スナップショットディスク

構成

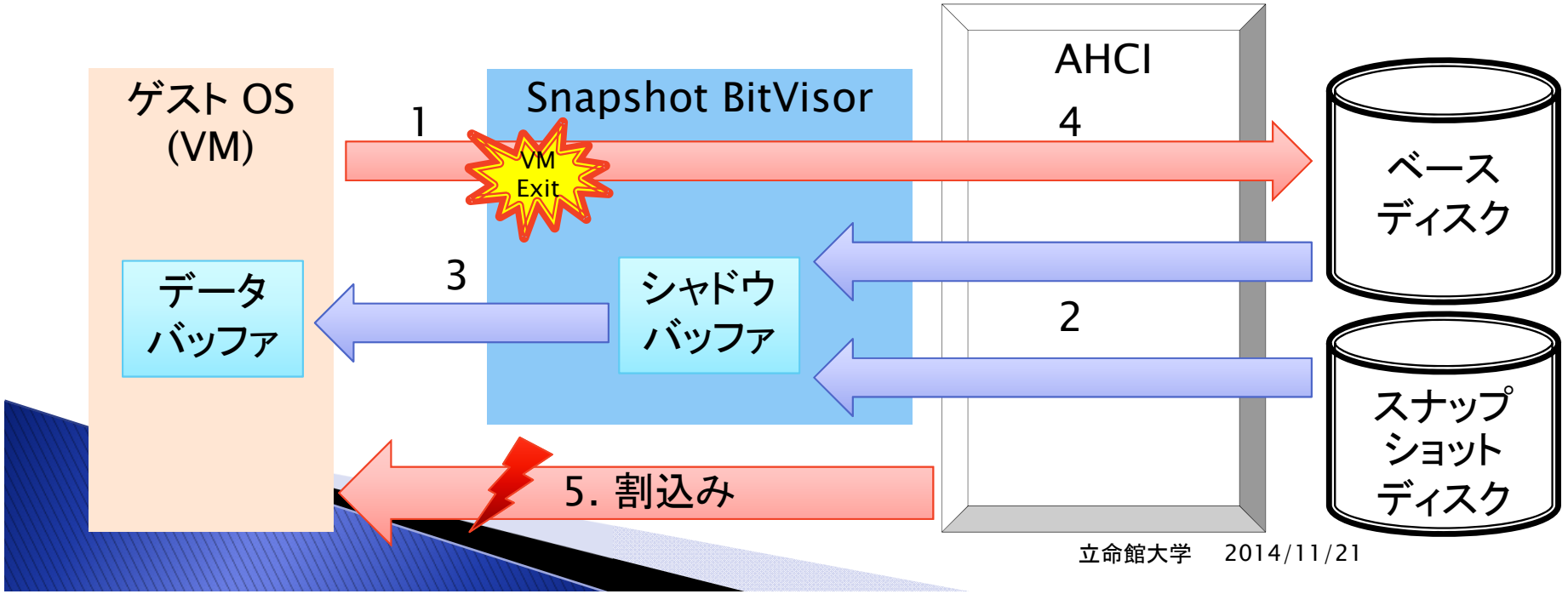

OS のアクセス
OS・VMM のアクセス
スナップショットディスクをゲストOSから隠すためのエミュレータ
• VM からスナップショットディスクへのアクセスを拒否する
• スナップショットディスクの書換えを防止
ゲスト OS (VM)
AHCI ドライバ
Snapshot BitVisor
仮想ディスクドライバ
Snapshot Disk Conceal Driver
AHCI 準パススルー ドライバ
物理マシン
AHCI
ベースディスク
スナップショットディスク

構成


OS のアクセス
OS・VMM のアクセス
VMM のアクセス
ゲスト OS (VM)
AHCI ドライバ
Snapshot BitVisor
仮想ディスク
ドライバ
Snapshot Disk
Conceal Driver
AHCI 準パススルー ドライバ
物理マシン
AHCI
BitVisor にあるドライバ
• 独立してディスクに
アクセスするために使用
ベースディスク
スナップショット
ディスク

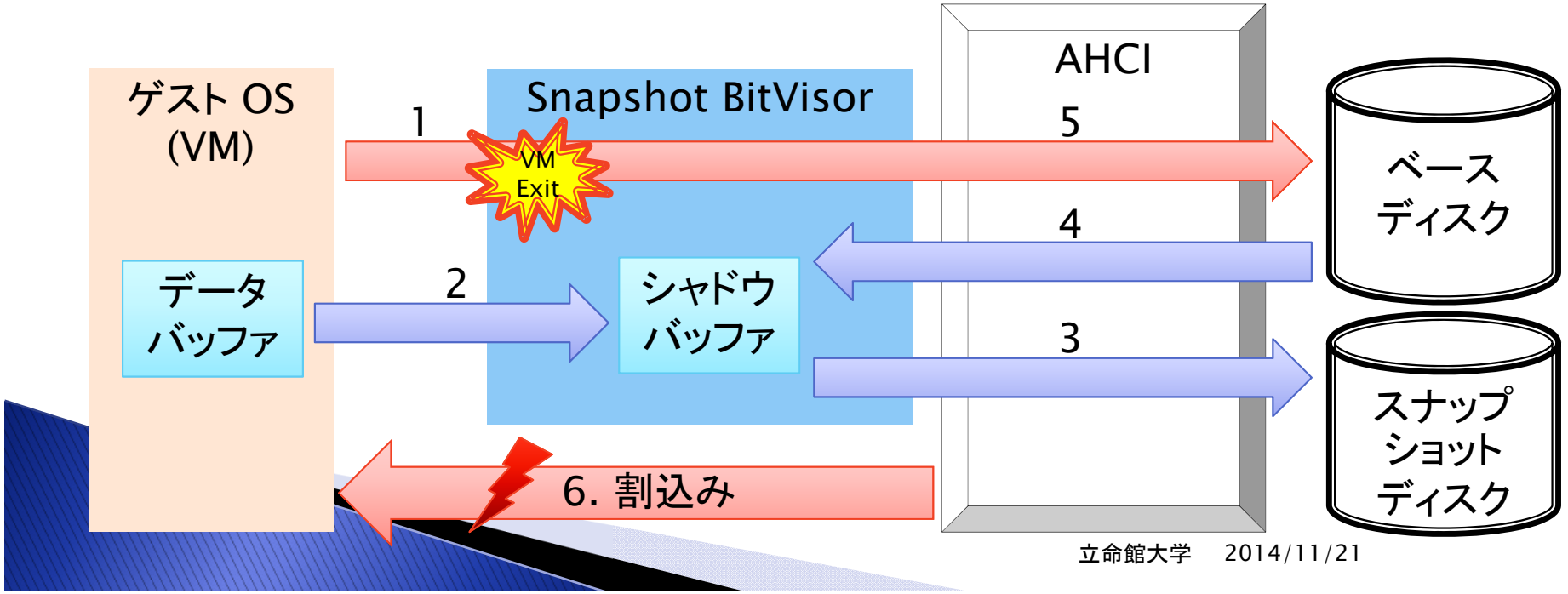
データ読出し手順

1. VM が 読出しコマンドを実行する
   - BitVisor に処理が移る (VM-Exit)
2. データをシャドウバッファに読み出す
   - 差分あり: スナップショットディスクから差分データを読み出す
   - 差分なし: ベースディスクからデータを読み出す
3. シャドウバッファから VM のデータバッファにコピーする
4. 元々の読出しコマンドを実行する
   - 読み出したデータは破棄する
5. 割込みを AHCI に発生させ，VM に割込みを通知する



データ書込み手順

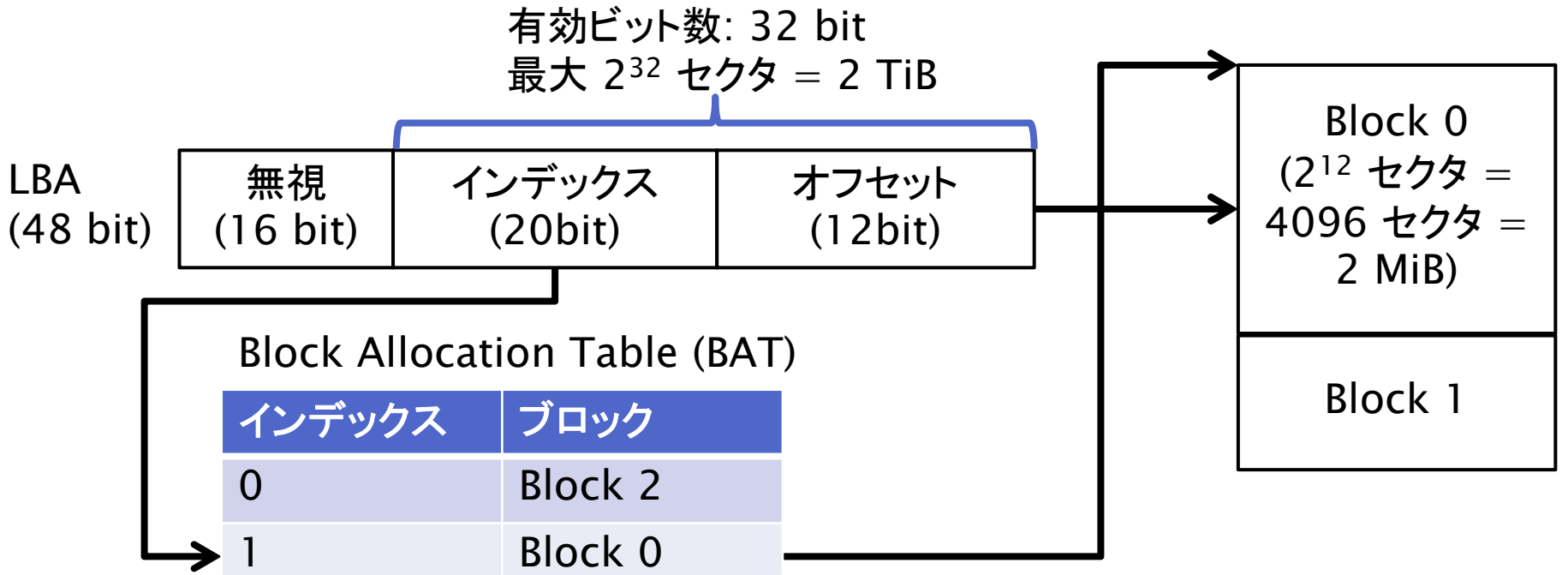
1. VM が 書込みコマンドを実行する
   - BitVisor に処理が移る (VM-Exit)
2. VM のデータバッファからシャドウバッファにコピーする
3. シャドウバッファのデータをスナップショットディスクへ書き込む
4. ベースディスクからシャドウバッファへデータを読み出す
   - 元々の書込みコマンドで書き換えようとしたアドレスのデータを読み出す
5. 元々の書込みコマンドを実行する
   - ただし，シャドウバッファのデータを書き込むようにする
6. 割込みを AHCI に発生させ，VM に割込みを通知する



スナップショットの管理

Microsoft Virtual Hard Disk (VHD) Image Format Specification を簡易化したフォーマットを使用して管理する

- データを 2 MiB のブロックに分割し，ブロックごとに差分データを管理する
- ページングと仕組みはほぼ同じ
  - 1 アドレスに対して 1セクタ = 512 B である点が異なる



スナップショットディスクレイアウト

- ヘッダ [2 MiB]
  - 空いているブロック番号 [4 B]
- BAT [4 MiB]
  - ブロック番号を 20 bit で格納する
    - $2^{20} \times 4\ \mathrm{B} = 4\ \mathrm{MiB}$
  - ブロックの LBA 算出
    - BAT[インデックス] << 12 + 16384
- ブロック0 [2 MiB]
- …

LBA	内容
0	MBR
1–4095	空き
4096 (2 MiB)	ヘッダ
8192 (4 MiB)	BAT
16384 (8 MiB)	Block 0
	…

2 MiB でアライメント

スナップショットを削除するときは，
ヘッダと BAT をゼロクリアする
→ 6 MiB の書込みだけで復元可能

デモンストレーション

- 実装した Snapshot BitVisor の動作をテストする
- 確認項目
  - Windows 7 をベースディスクにインストールし，環境を復元できるか
  - スナップショットディスクを隠せているか

課題点

- Windows 7 の起動で一時的にハングアップする
  - Windows のロゴが表示される直前で約 90 秒ほど停止
    - Windows がディスクの接続状態を確認しているため停止する
    - タイムアウト?により起動が続行する
  - Conceal Driver に不具合がある可能性がある
- 1世代のみしか管理できない
  - ヘッダを変更することで複数世代のスナップショットが管理できるようになる
- 割込み生成のために I/O 待ち時間がある
  - オリジナルのコマンドを実行させて割込みを生成している
    - I/O 待ちができ，スループットが低下している
  - 割込みをエミュレーションによって解決できると考える

おわりに

- Snapshot BitVisor
  - スナップショット機能を実現する BitVisor
  - 2 台のストレージに分け，差分データを保管
  - AHCI を使用してアクセスを捕捉・書換え
  - Windows，Linux で起動することを確認した
  - 約 1100 行の変更で実装できた