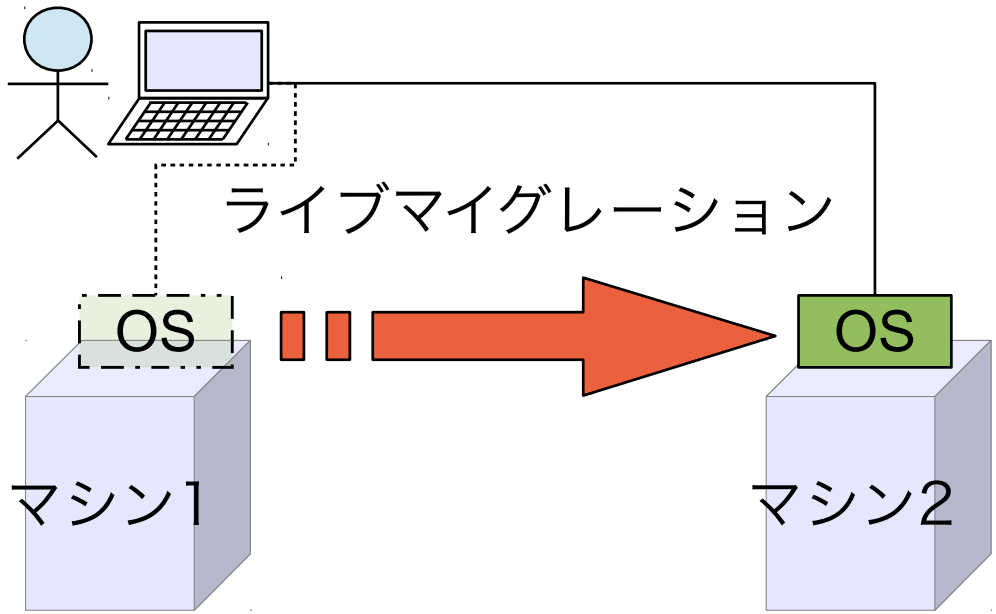
Live Migration on BitVisor

深井 貴明 †, 表 祐志 †, 品川 高廣 ‡

† 筑波大学, ‡ 東京大学

2013 年 12 月 6 日

OSのライブマイグレーション



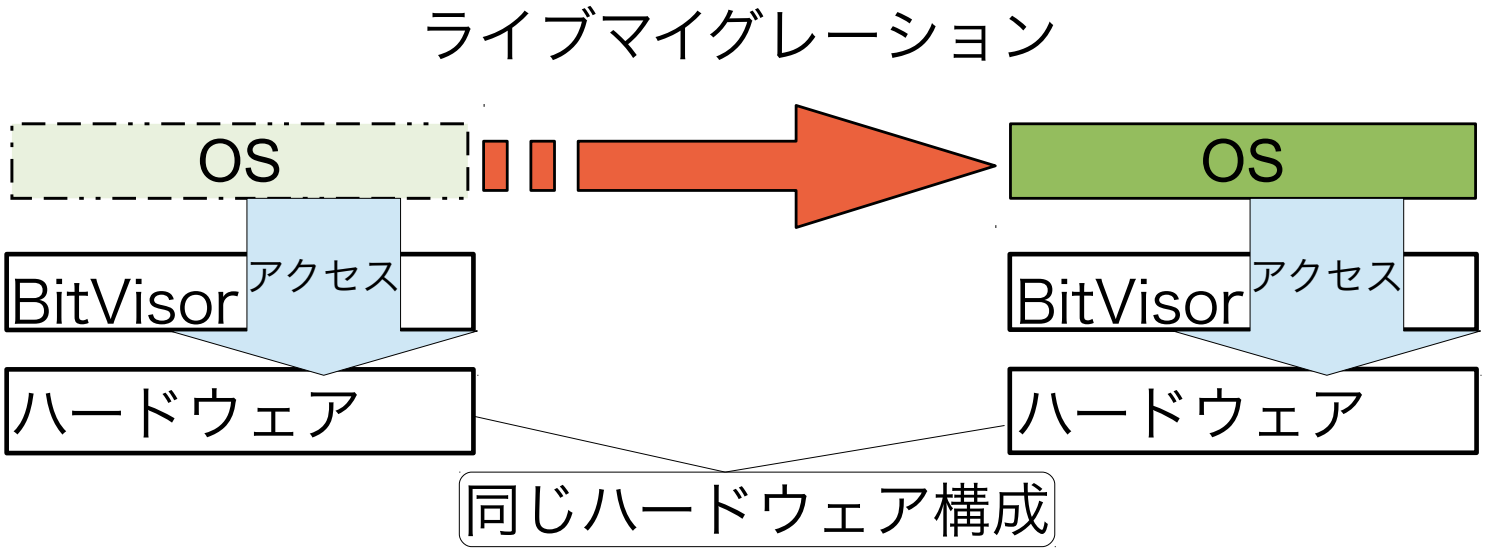
- クラウド上でメンテナンスに用いる重要な機能
  - あるマシン上の OS を無停止で別の物理マシン上へ移動
  - OS に依存しない
  - 主にデバイスを仮想化した VM 環境で実現されている

Live Migration on BitVisor の利点

- メンテナンス性を向上
- 物理マシンの機能を活かせる
  - 仮想化のオーバヘッド削減
  - デバイスの機能をフルに使う
- OS に依存しない

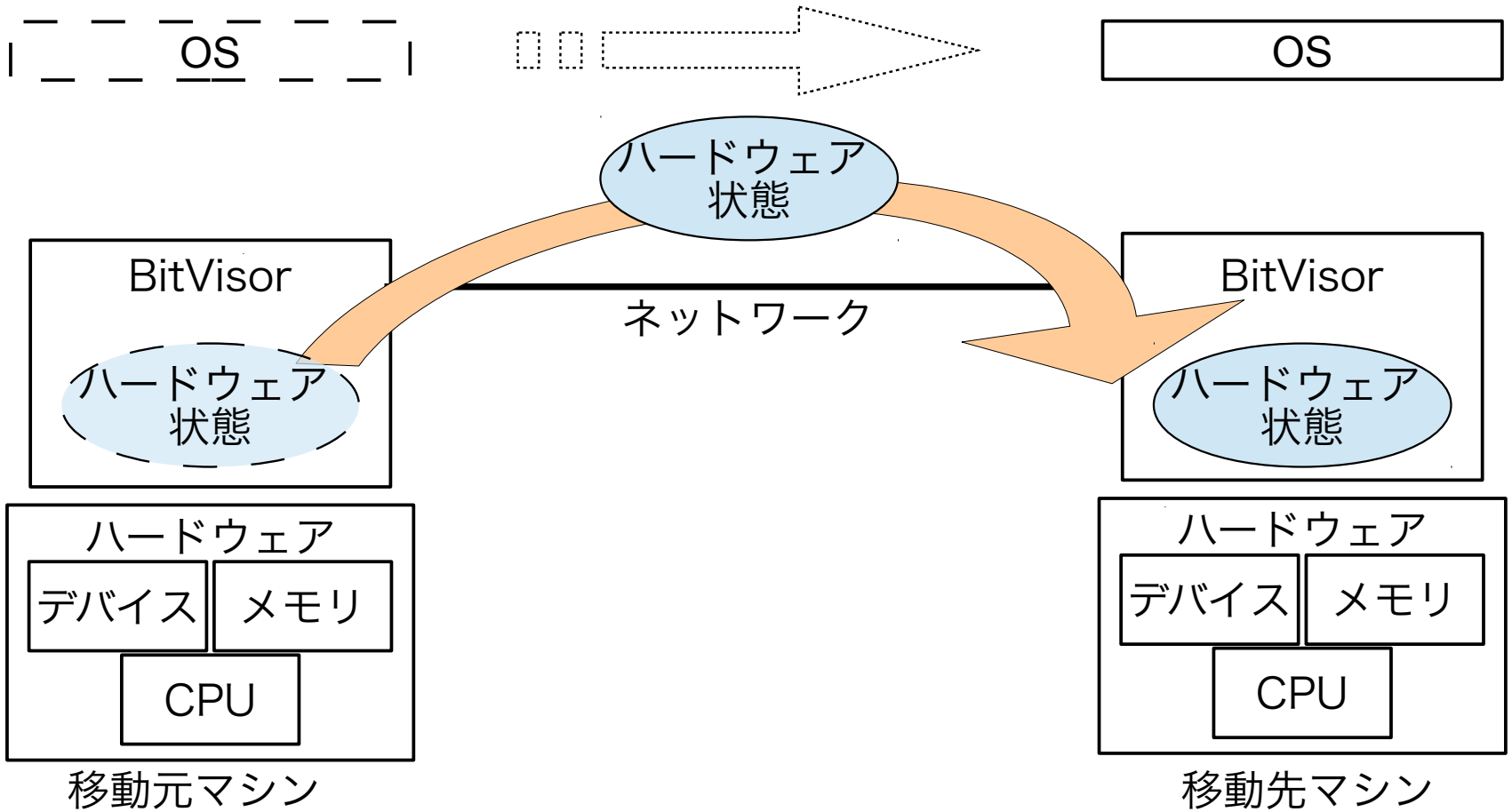
利用場面: データベース処理，HPC，ベアメタルクラウド

Live Migration on BitVisor の全体像

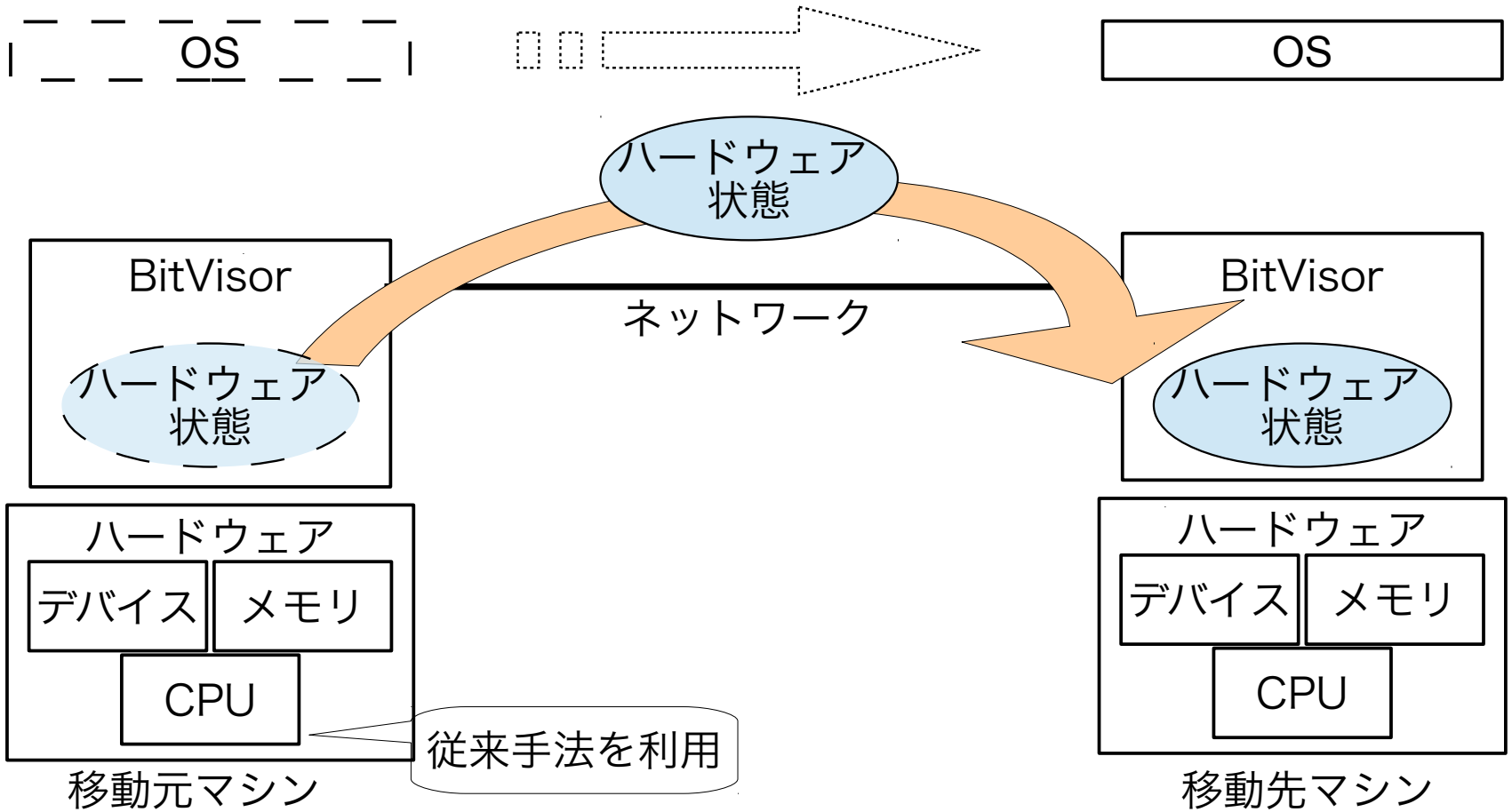


- VMM はデバイスを一切仮想化しない
  → ○　オーバヘッド削減
- VMM のレイヤのみで実現
  → ○　OS に依存しない

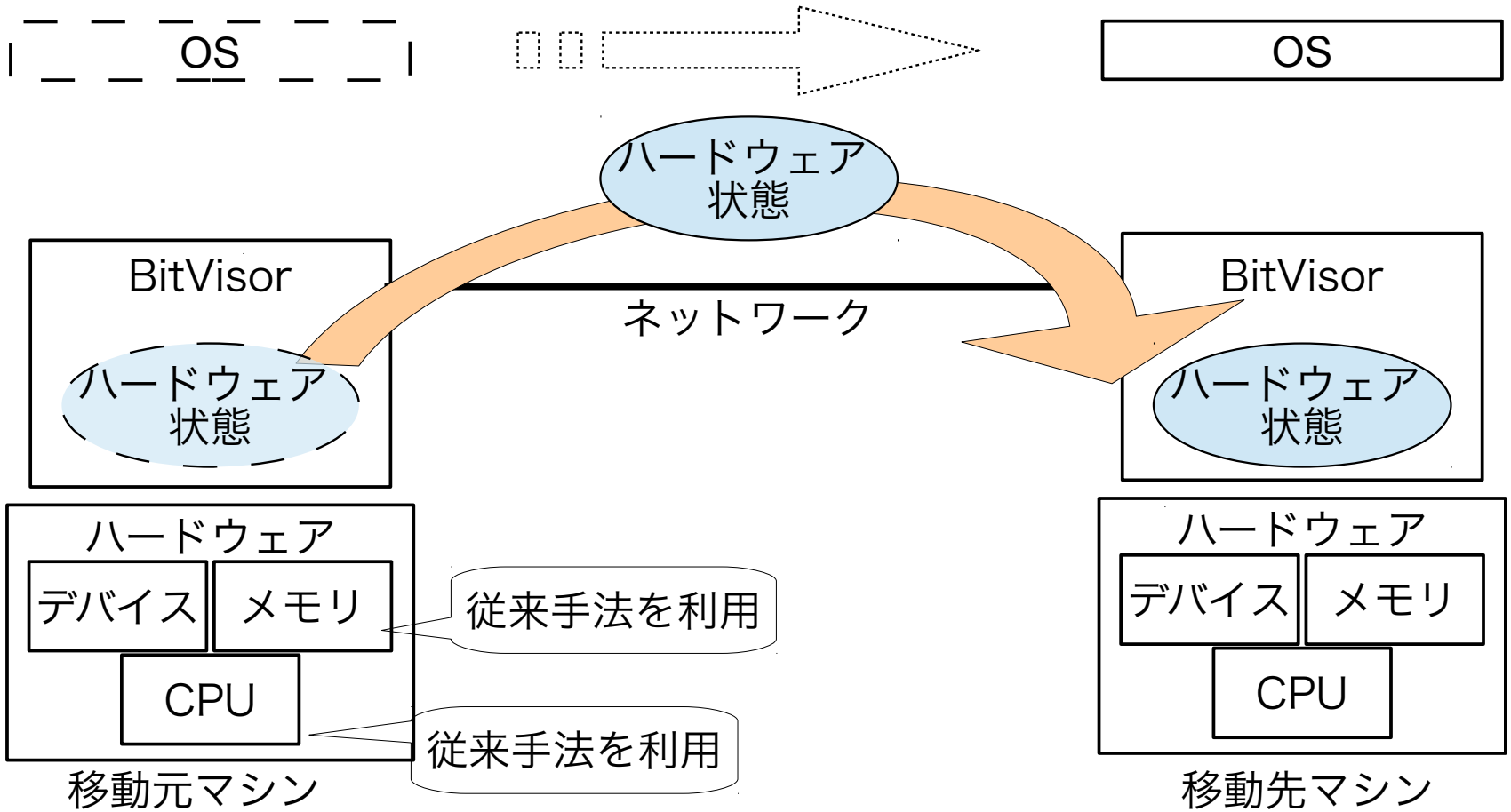
ライブマイグレーションの処理流れ



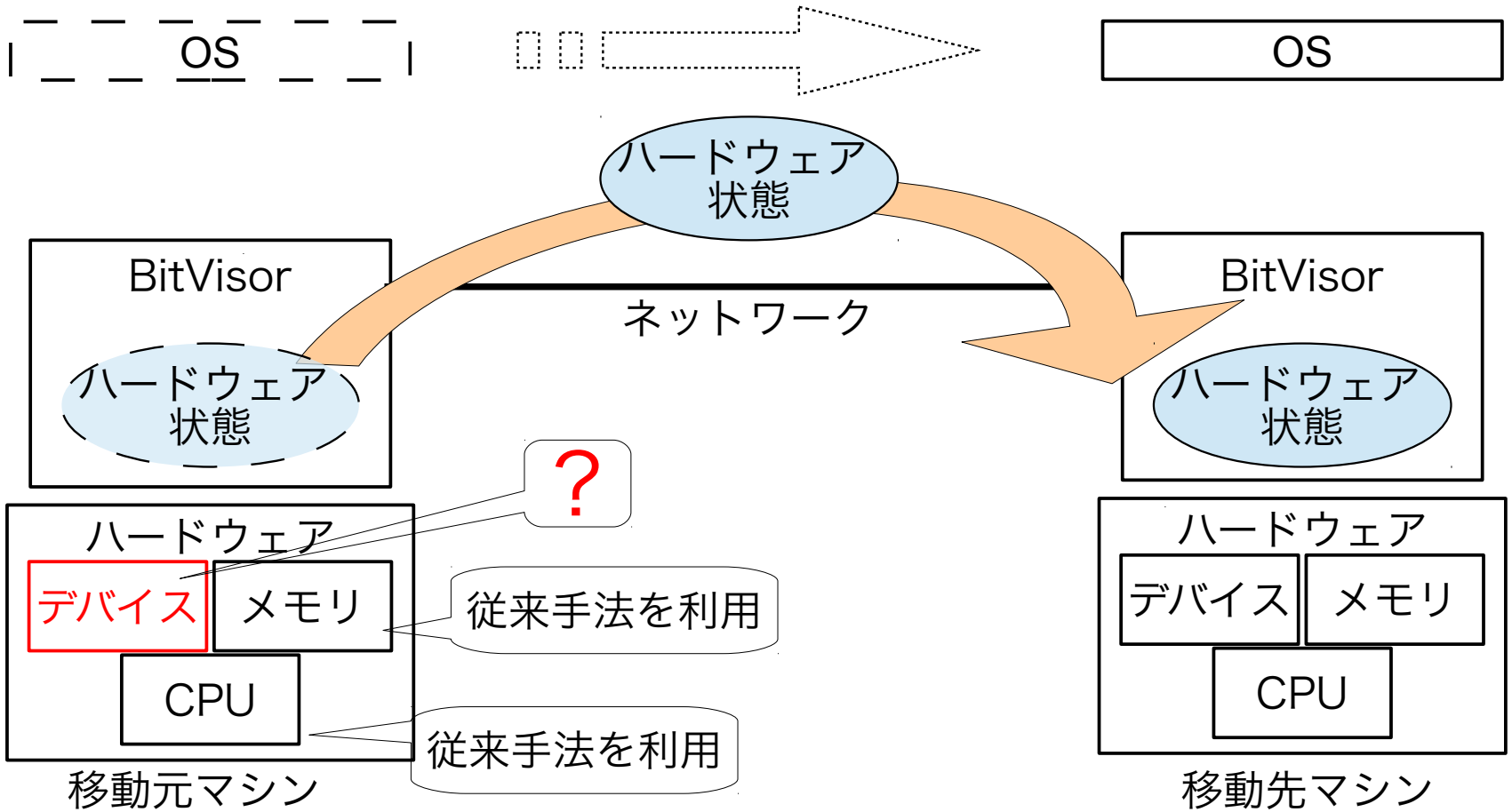
ライブマイグレーションの処理流れ



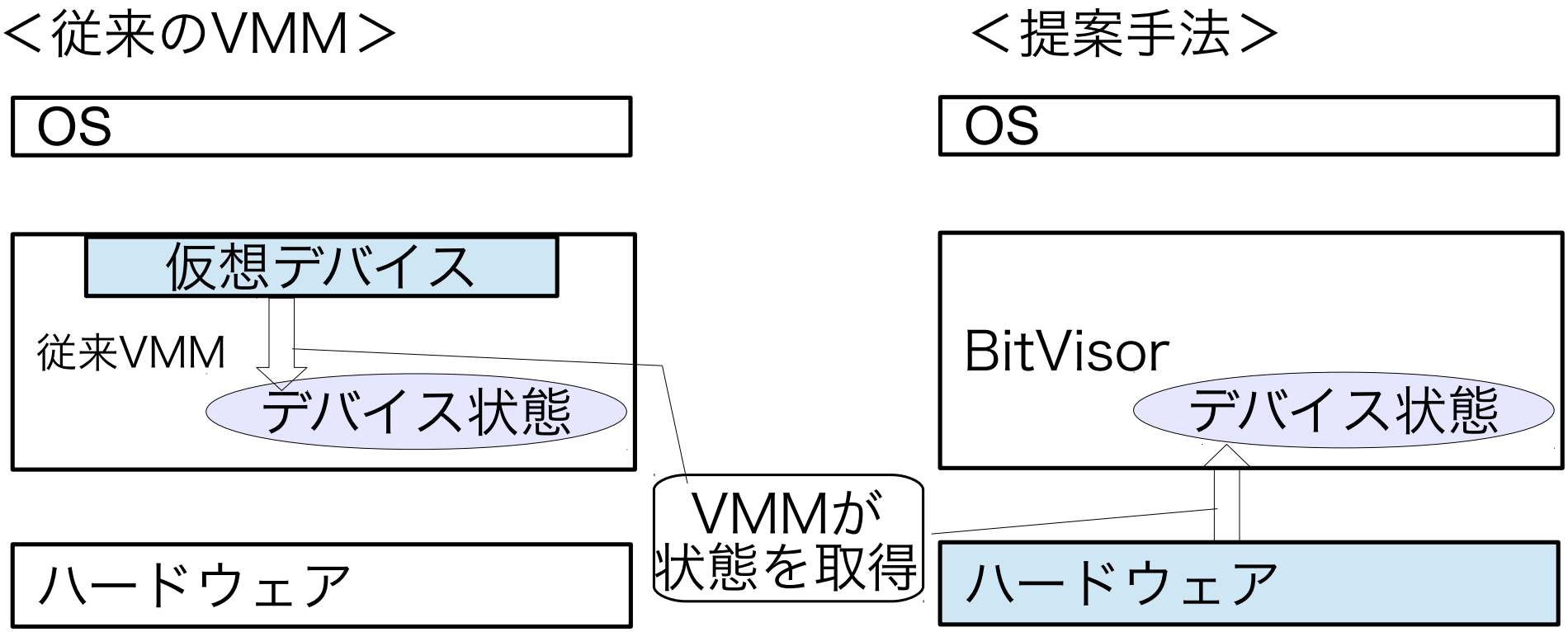
ライブマイグレーションの処理流れ



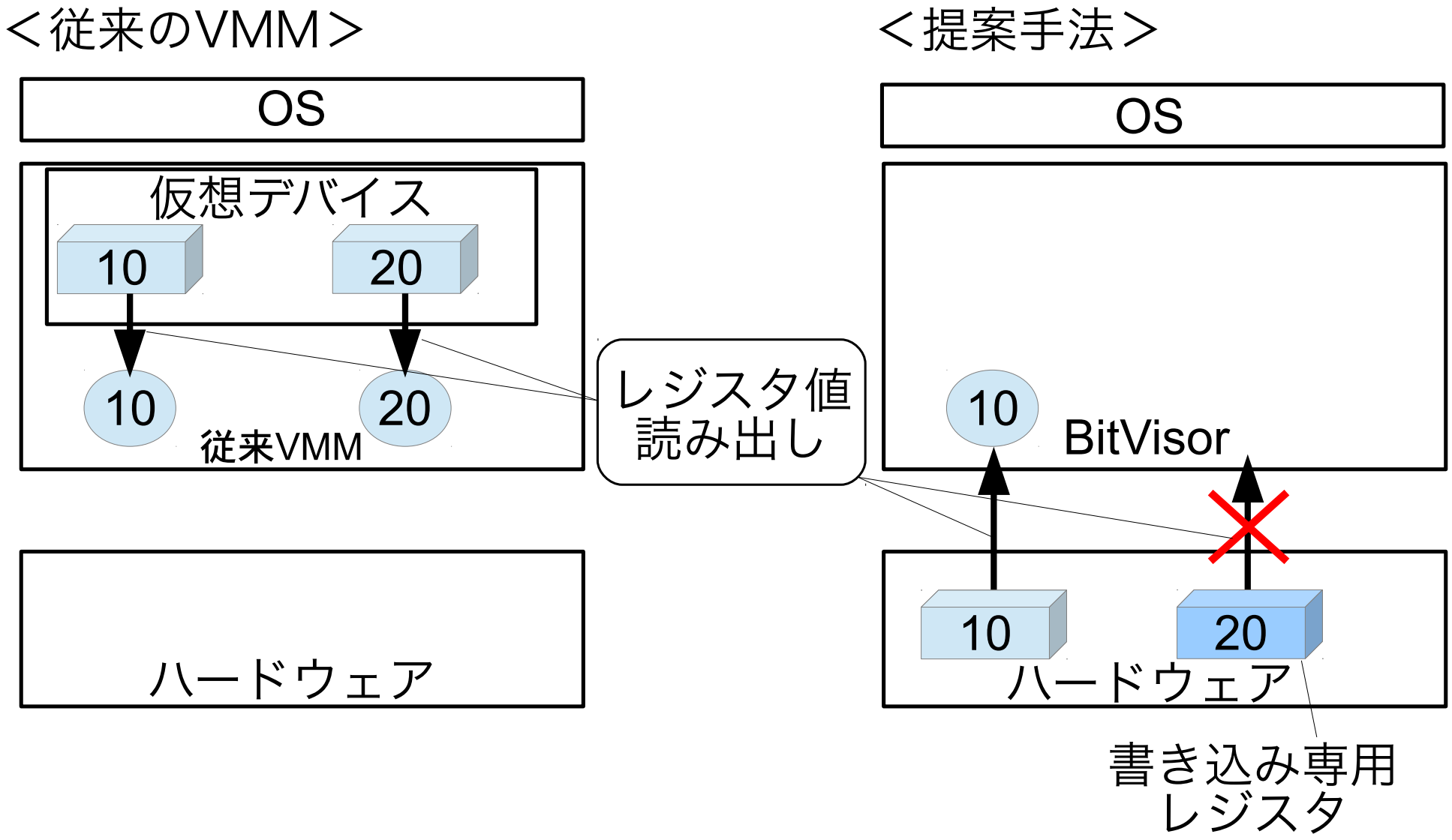
ライブマイグレーションの処理流れ



VMMによるデバイス状態の取得

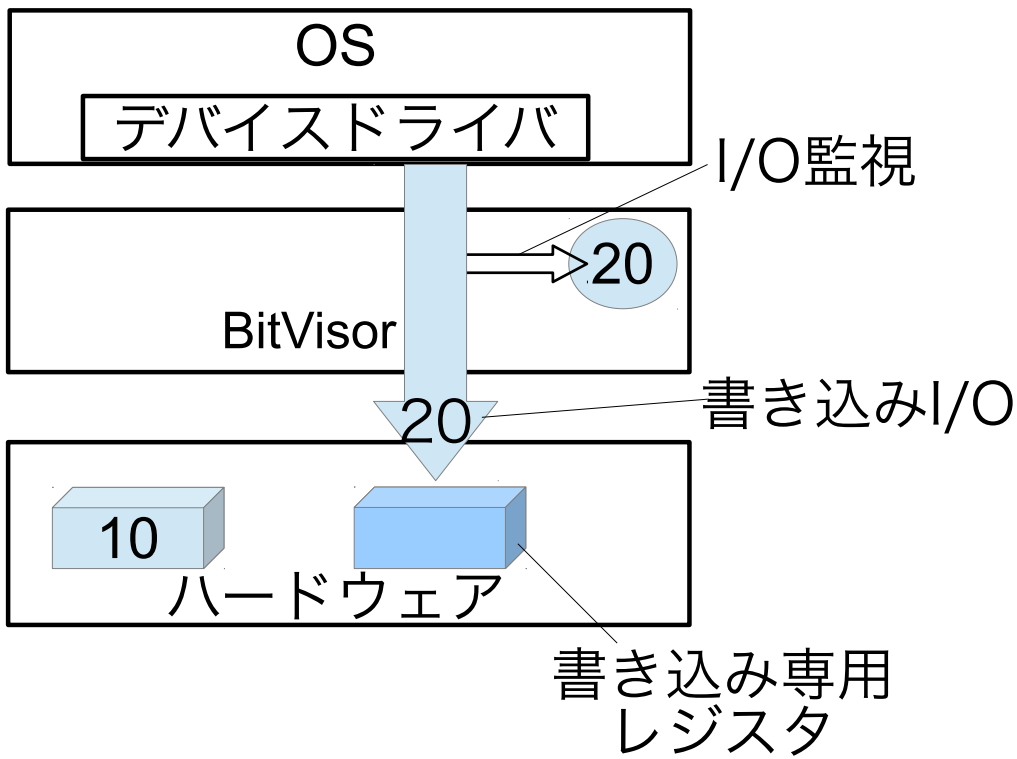


VMMによるデバイスレジスタの値取得



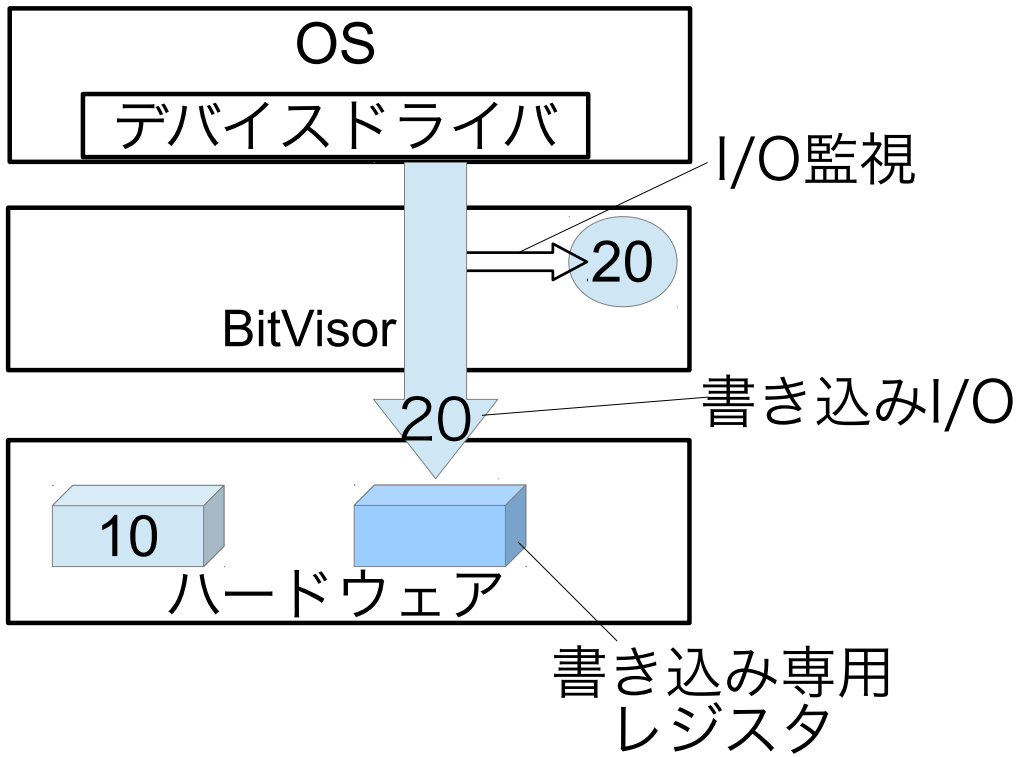
I/O 監視による

書き込み専用レジスタの値の取得



I/O 監視による

書き込み専用レジスタの値の取得



書き込み専用レジスタへの書き込み I/O をどのように判別するか?

I/O の判別方法(1/2)

以下の情報を合わせて判別

1. 宛先ポート
   - どのデバイスへの I/O か
2. I/O タイプ
   - 読み込み or 書き込み
3. 書き込みデータ
   - 一部の I/O は書き込みデータの特定ビットがセットされているか否かで書き込むレジスタが異なる

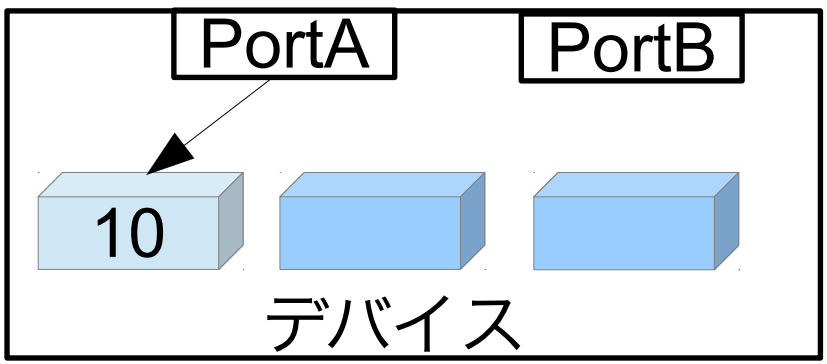
I/O の判別方法(2/2)

4. デバイスの状態
   - 他のレジスタの値によって書き込む先のレジスタが異なる
5. 直近の I/O
   - 一部の I/O は直近の I/O によって書き込むレジスタが異なる.

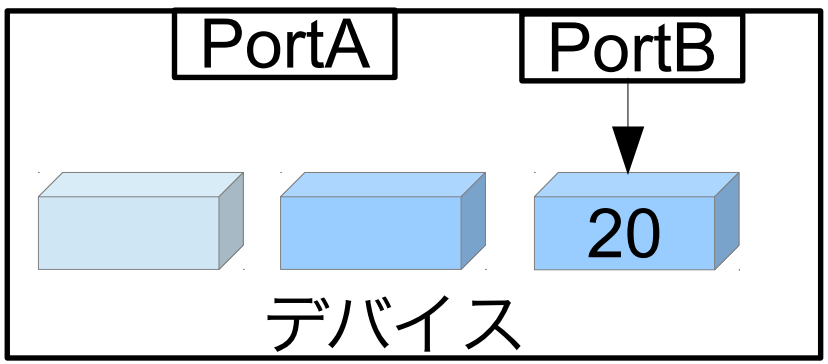
I/O の判別方法(2/2)

4. デバイスの状態
   - 他のレジスタの値によって書き込む先のレジスタが異なる
5. 直近の I/O
   - 一部の I/O は直近の I/O によって書き込むレジスタが異なる.

直近の I/O による書き込み先レジスタの違い



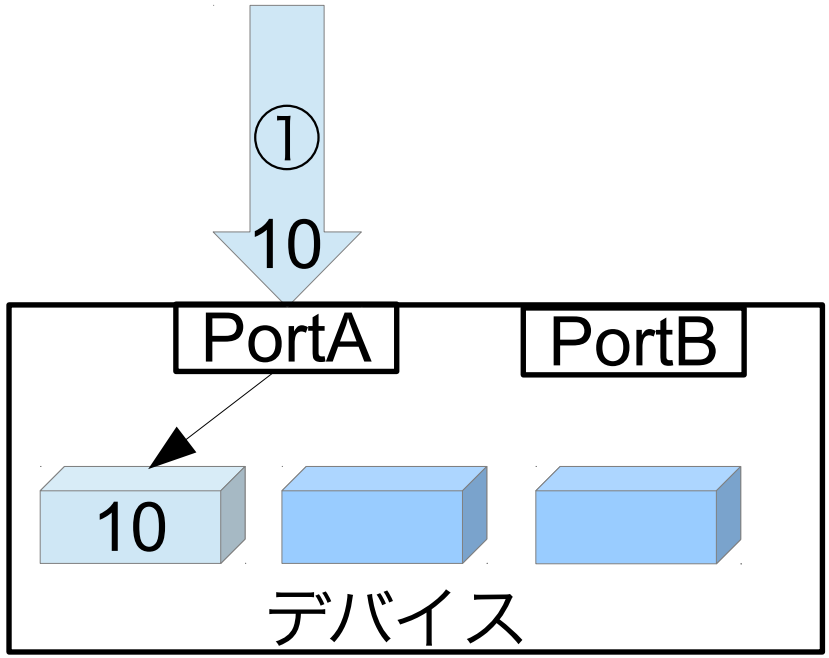
①PortA 10 → ②PortB 20



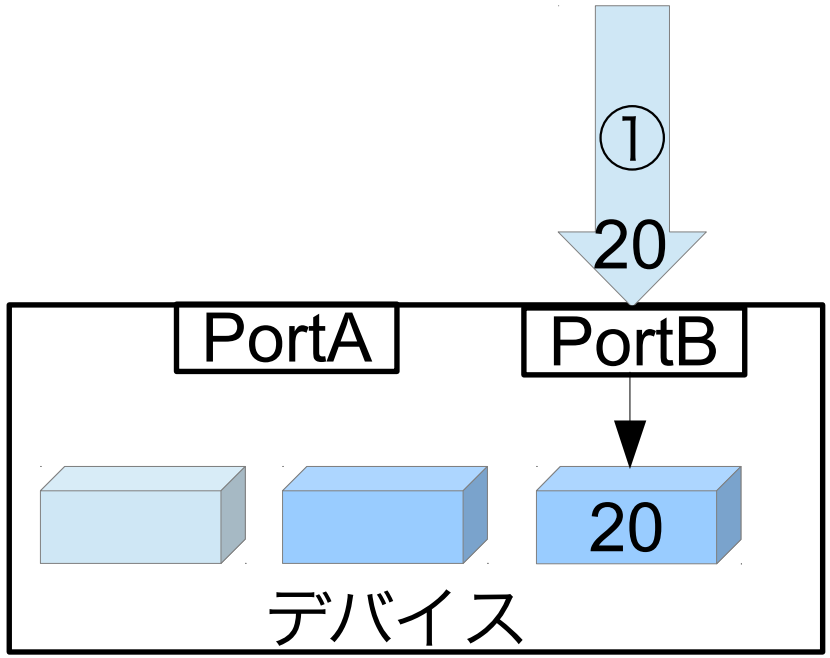
①PortB 20 → ②PortA 10

例: PortA と PortB に異なる順序で 1 度ずつ書き込む

直近の I/O による書き込み先レジスタの違い



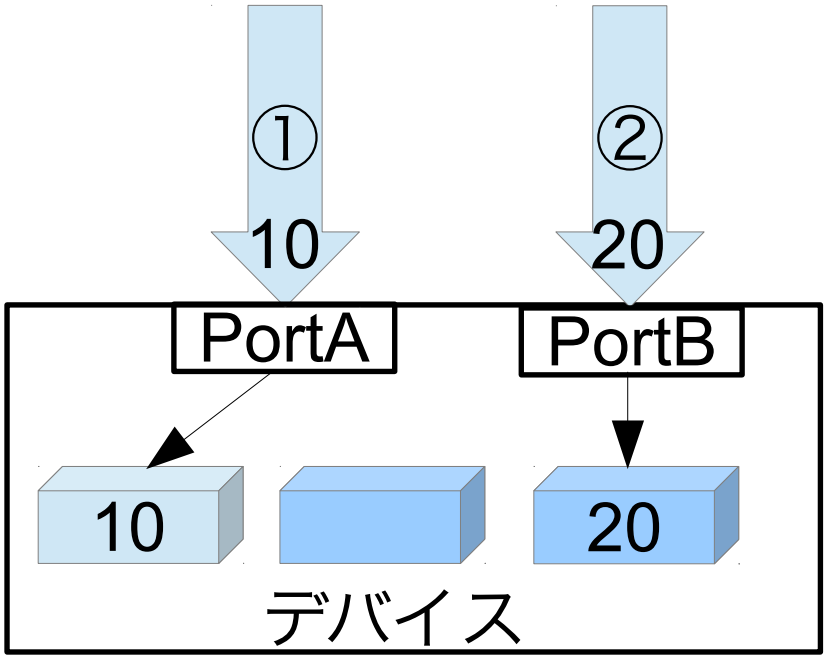
①PortA 10 → ②PortB 20



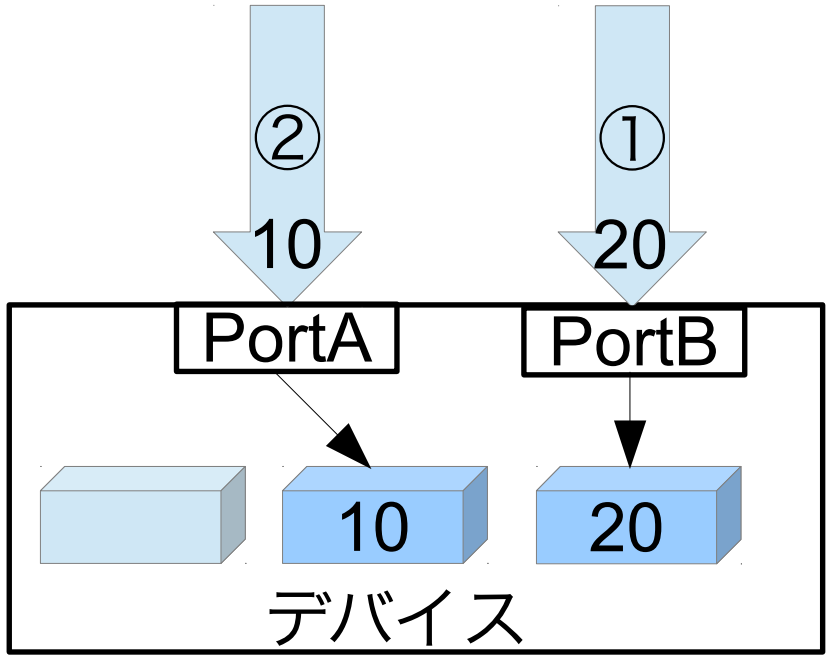
①PortB 20 → ②PortA 10

例: PortA と PortB に異なる順序で 1 度ずつ書き込む

直近の I/O による書き込み先レジスタの違い



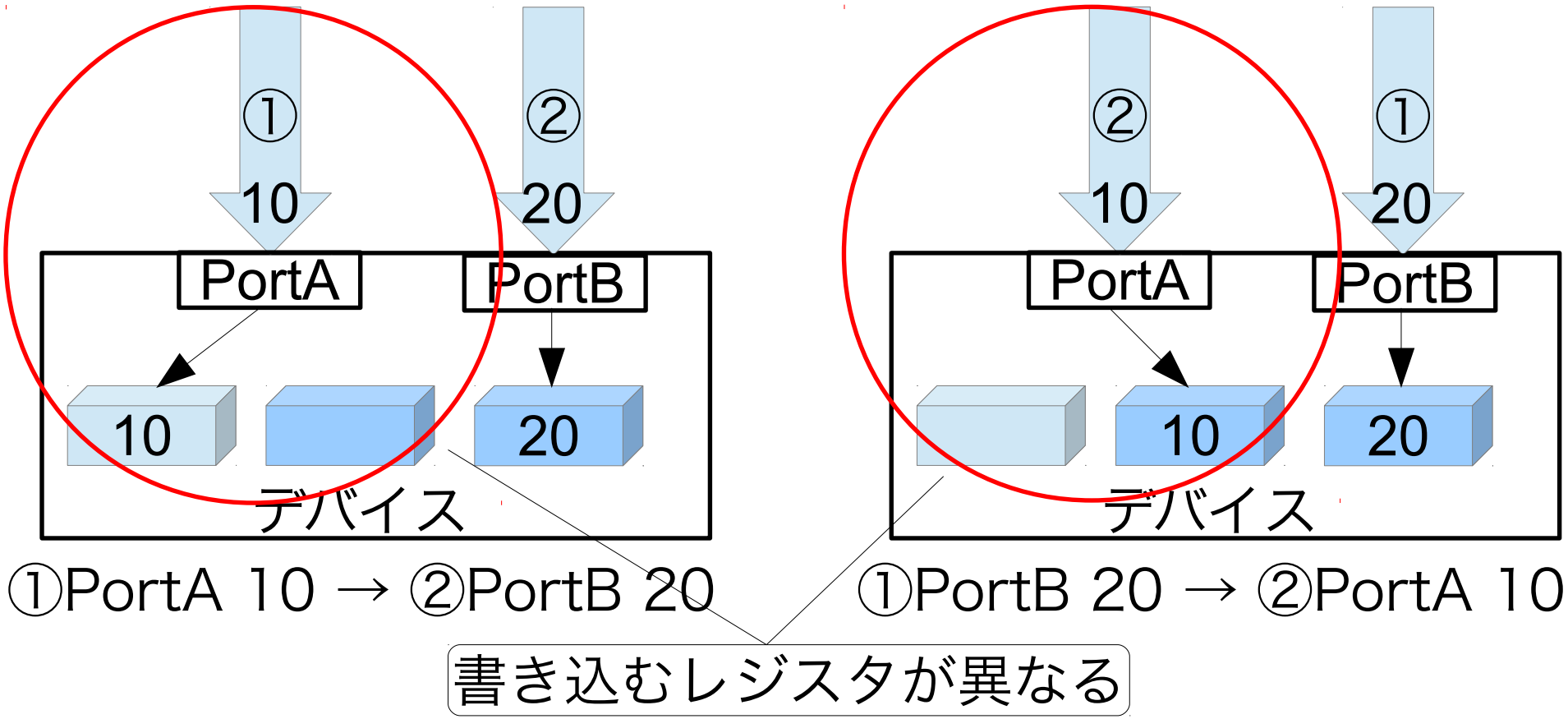
①PortA 10 → ②PortB 20



①PortB 20 → ②PortA 10

例: PortA と PortB に異なる順序で 1 度ずつ書き込む

直近の I/O による書き込み先レジスタの違い



例: PortA と PortB に異なる順序で 1 度ずつ書き込む

状態を取得するためには

- 状態は値ではなく，一連の I/O シーケンスとして記録する.
  - 直近の I/O によって書き込み先レジスタが異なるため.
- デバイス毎に書き込み専用レジスタへの I/O か否かを判別する処理を記述する必要がある.
  - デバイス毎に仕様が異なるため.

以降では，デバイス状態取得の一例として PIC の初期化 I/O についての実装を述べる.

PICのレジスタテーブル

Table 10-3. PIC Registers (LPC I/F—D31:F0)

Port	Aliases	Register Name	Default Value	Type
20h	24h, 28h, 2Ch, 30h, 34h, 38h, 3Ch	Master PIC ICW1 Init. Cmd Word 1	Undefined	WO
		Master PIC OCW2 Op Ctrl Word 2	001XXXXXb	WO
		Master PIC OCW3 Op Ctrl Word 3	X01XXX10b	WO
21h	25h, 29h, 2Dh, 31h, 35h, 39h, 3Dh	Master PIC ICW2 Init. Cmd Word 2	Undefined	WO
		Master PIC ICW3 Init. Cmd Word 3	Undefined	WO
		Master PIC ICW4 Init. Cmd Word 4	01h	WO
		Master PIC OCW1 Op Ctrl Word 1	00h	R/W
A0h	A4h, A8h, ACh, B0h, B4h, B8h, BCh	Slave PIC ICW1 Init. Cmd Word 1	Undefined	WO
		Slave PIC OCW2 Op Ctrl Word 2	001XXXXXb	WO
		Slave PIC OCW3 Op Ctrl Word 3	X01XXX10b	WO
A1h	A5h, A9h, ADh, B1h, B5h, B9h, BDh	Slave PIC ICW2 Init. Cmd Word 2	Undefined	WO
		Slave PIC ICW3 Init. Cmd Word 3	Undefined	WO
		Slave PIC ICW4 Init. Cmd Word 4	01h	WO
		Slave PIC OCW1 Op Ctrl Word 1	00h	R/W
4D0h	–	Master PIC Edge/Level Triggered	00h	R/W
4D1h	–	Slave PIC Edge/Level Triggered	00h	R/W

PICのレジスタテーブル

Table 10-3. PIC Registers (LPC I/F—D31:F0)

Port	Aliases	Register Name	Default Value	Type
20h	24h, 28h, 2Ch, 30h, 34h, 38h, 3Ch	Master PIC ICW1 Init. Cmd Word 1	Undefined	WO
		Master PIC OCW2 Op Ctrl Word 2	001XXXXXb	WO
		Master PIC OCW3 Op Ctrl Word 3	X01XXX10b	WO
21h	25h, 29h, 2Dh, 31h, 35h, 39h, 3Dh	Master PIC ICW2 Init. Cmd Word 2	Undefined	WO
		Master PIC ICW3 Init. Cmd Word 3	Undefined	WO
		Master PIC ICW4 Init. Cmd Word 4	01h	WO
		Master PIC OCW1 Op Ctrl Word 1	00h	R/W
A0h	A4h, A8h, ACh, B0h, B4h, B8h, BCh	Slave PIC ICW1 Init. Cmd Word 1	Undefined	WO
		Slave PIC OCW2 Op Ctrl Word 2	001XXXXXb	WO
		Slave PIC OCW3 Op Ctrl Word 3	X01XXX10b	WO
A1h	A5h, A9h, ADh, B1h, B5h, B9h, BDh	Slave PIC ICW2 Init. Cmd Word 2	Undefined	WO
		Slave PIC ICW3 Init. Cmd Word 3	Undefined	WO
		Slave PIC ICW4 Init. Cmd Word 4	01h	WO
		Slave PIC OCW1 Op Ctrl Word 1	00h	R/W
4D0h	–	Master PIC Edge/Level Triggered	00h	R/W
4D1h	–	Slave PIC Edge/Level Triggered	00h	R/W

0x21 ポートに割り当てられる PIC のレジスタ

Write Only と Read/Write が混在している.

レジスタ	権限
ICW2	Write Only
ICW3	Write Only
ICW4	Write Only
OCW1	Read/Write

- ICWx = Initialization Command Word
  - PIC の初期化時に書き込まれるレジスタ
- OCW1 = Operational Control Word
  - 割り込みマスク

0x21 ポートに割り当てられる PIC のレジスタ

Write Only と Read/Write が混在している.

レジスタ	権限
ICW2	Write Only
ICW3	Write Only
ICW4	Write Only
OCW1	Read/Write

- ICWx = Initialization Command Word ← Write Only
  - PIC の初期化時に書き込まれるレジスタ
- OCW1 = Operational Control Word
  - 割り込みマスク

PICの初期化の仕様

- PIC の初期化は ICW1～ICW4 の 4 つのレジスタへの書き込む
- ICW1 への書き込み I/O はポート 0x20 宛
- ICW1 へ書き込む際は，書き込みデータの 4bit 目をセットする
- ICW2～ICW4 への書き込み I/O はポート 0x21 宛
- ICW2～ICW4 は ICW1 の発行直後に 0x21 へ書き込む

PICの初期化シーケンスのI/O

PICの初期化シーケンスのI/O

OUTB 0x20 0b00010001

PICの初期化シーケンスのI/O

OUTB 0x20 0b00010001

(0x20 への書き込み)
&&(4bit 目が1)
=ICW1

PICの初期化シーケンスのI/O

OUTB 0x20 0b00010001

OUTB 0x21 0b00110000

OUTB 0x21 0b00000100

OUTB 0x21 0b00000001

(0x20 への書き込み)
&&(4bit 目が1)
=ICW1

PICの初期化シーケンスのI/O

OUTB 0x20 0b00010001

OUTB 0x21 0b00110000

OUTB 0x21 0b00000100

OUTB 0x21 0b00000001

(0x20 への書き込み)
&&(4bit 目が1)
=ICW1

続く0x21への3回の書き込み
=ICW2～ICW4

PICの初期化シーケンスのI/O

```
OUTB 0x20 0b00010001
OUTB 0x21 0b00110000
OUTB 0x21 0b00000100
OUTB 0x21 0b00000001
```

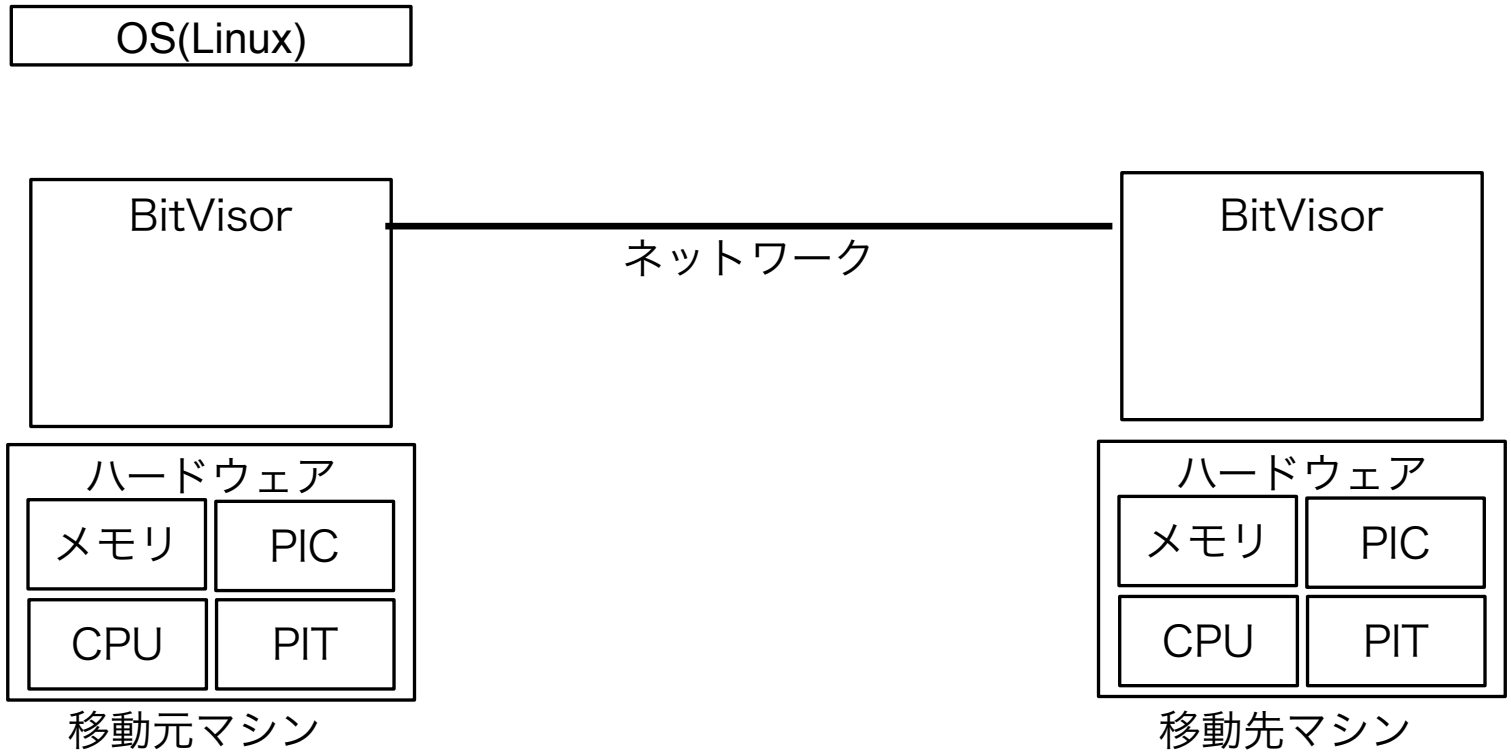
(0x20 への書き込み)
&&(4bit 目が1)
=ICW1

続く0x21への3回の書き込み
=ICW2～ICW4

記録対象I/O

実装状況

機能を制限したLinuxのライブマイグレーションが可能

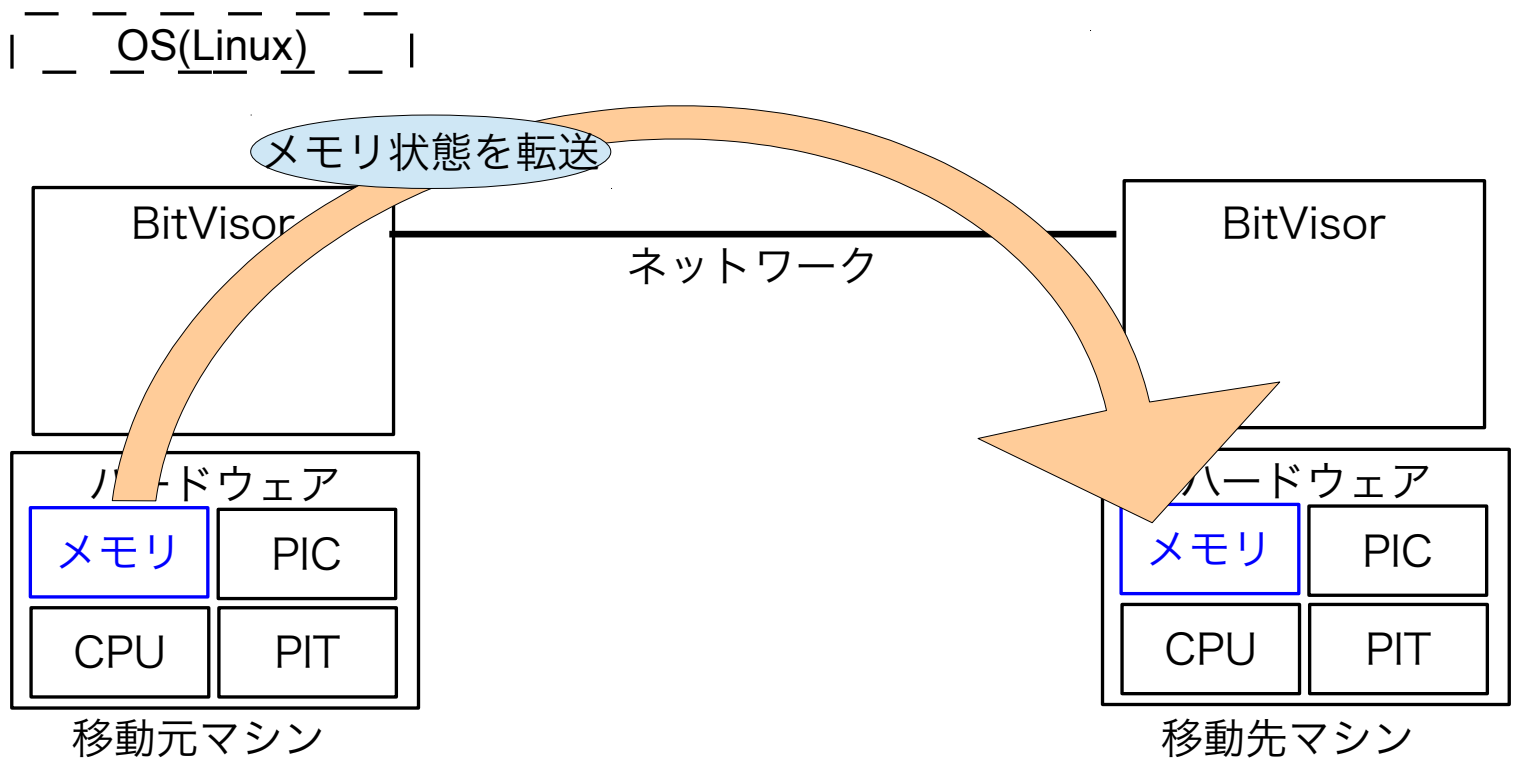


PIC=Programmable Interrupt Controller(割り込みコントローラ)

PIT=Programmable Interval Timer (タイマ)

実装状況

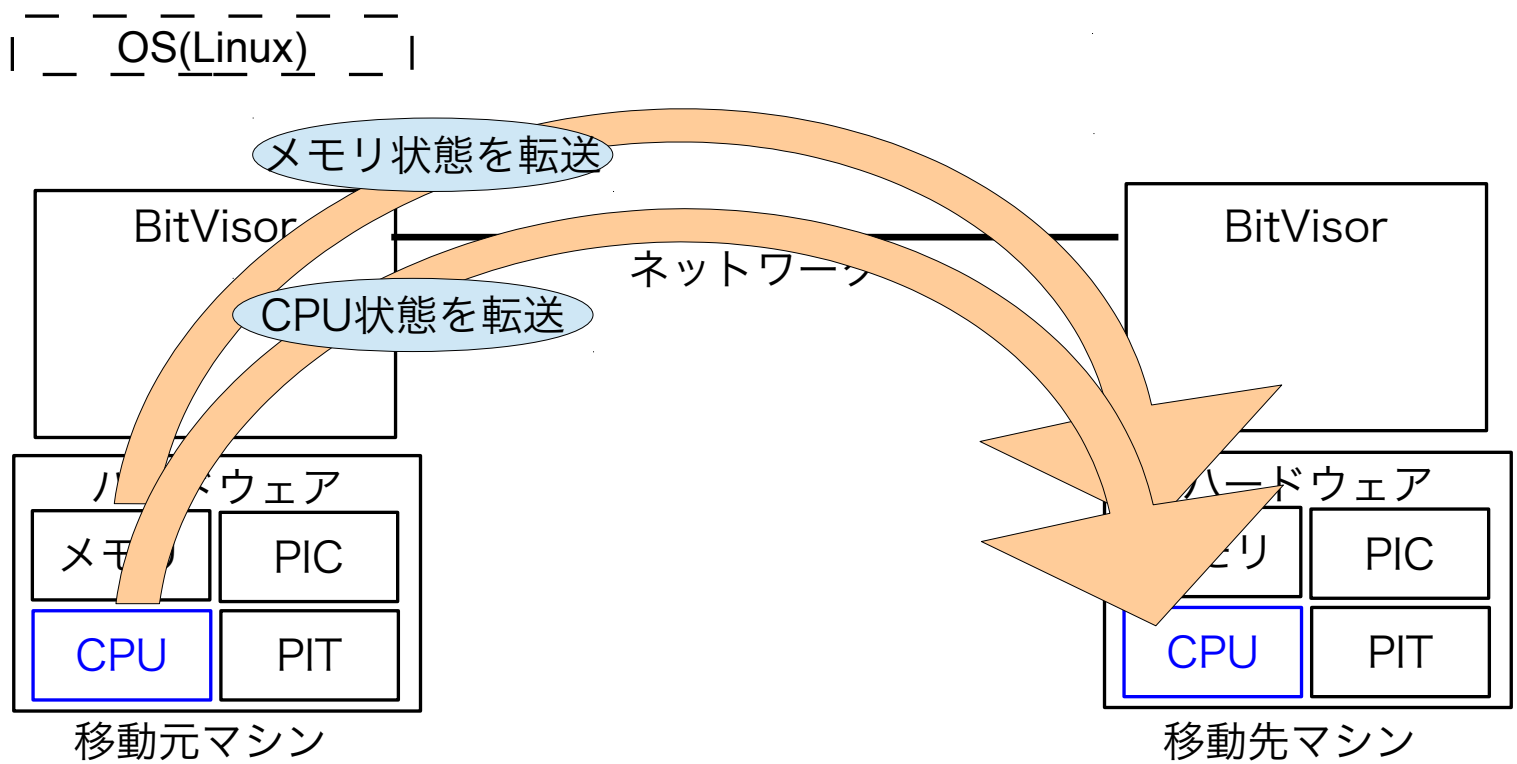
機能を制限したLinuxのライブマイグレーションが可能



メモリのマイグレーションはStop-and-Copy

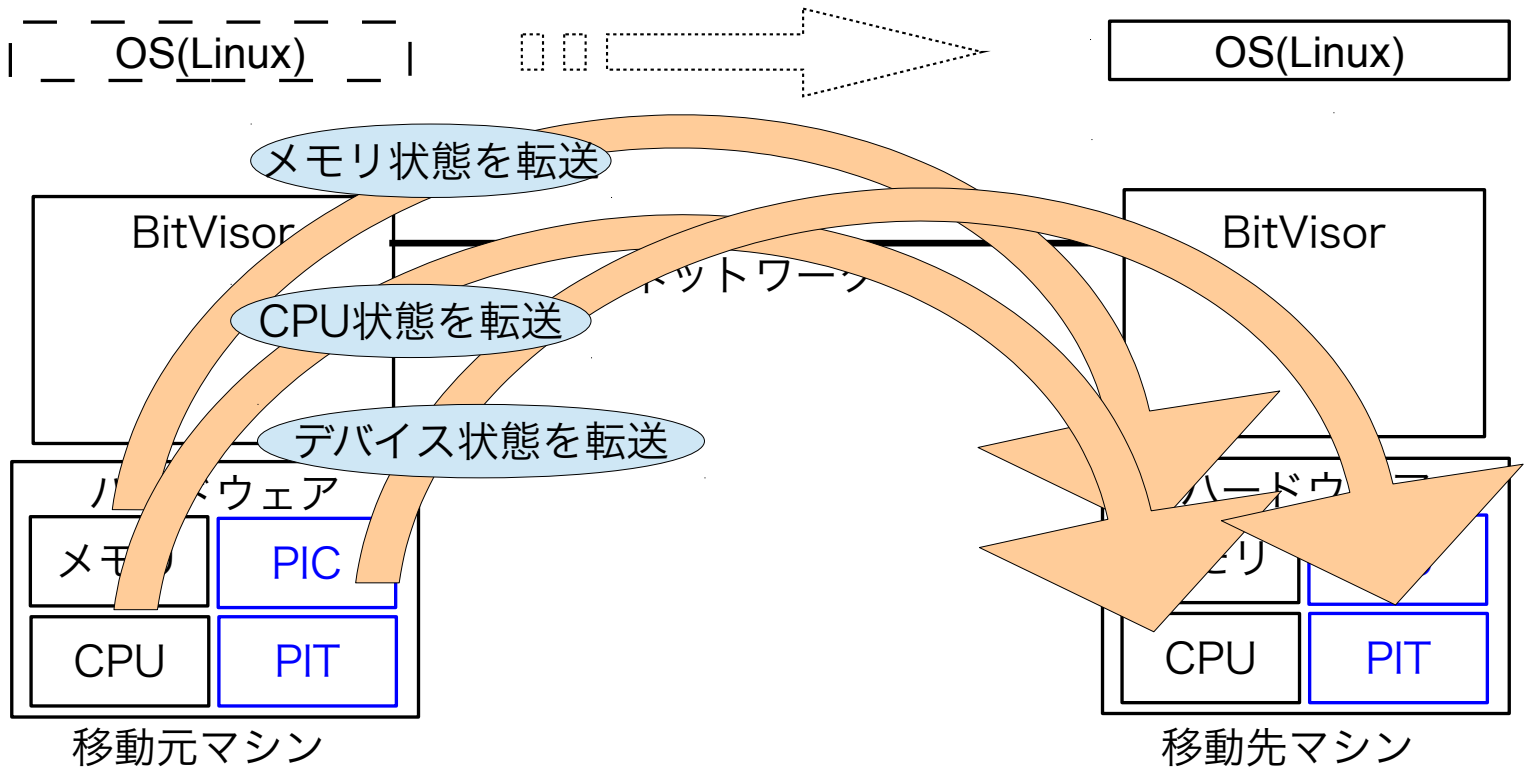
実装状況

機能を制限したLinuxのライブマイグレーションが可能



実装状況

機能を制限したLinuxのライブマイグレーションが可能



おまけ: ライブマイグレーション技術の応用先

- Checkpoint
  - 状態の取得，復元はライブマイグレーションと同じ
- 実行環境のクローン
  - 転送後，止めなければよい
  - 今回のデモも実際はクローン
  - 使い道は...???

まとめ

- 物理マシン間の OS ライブマイグレーション手法を提案
  - デバイスを仮想化しない VMM を利用
  - VMM で OS の I/O を監視し物理デバイスの状態を取得
- 機能を制限した Linux のライブマイグレーションを確認

今後の課題

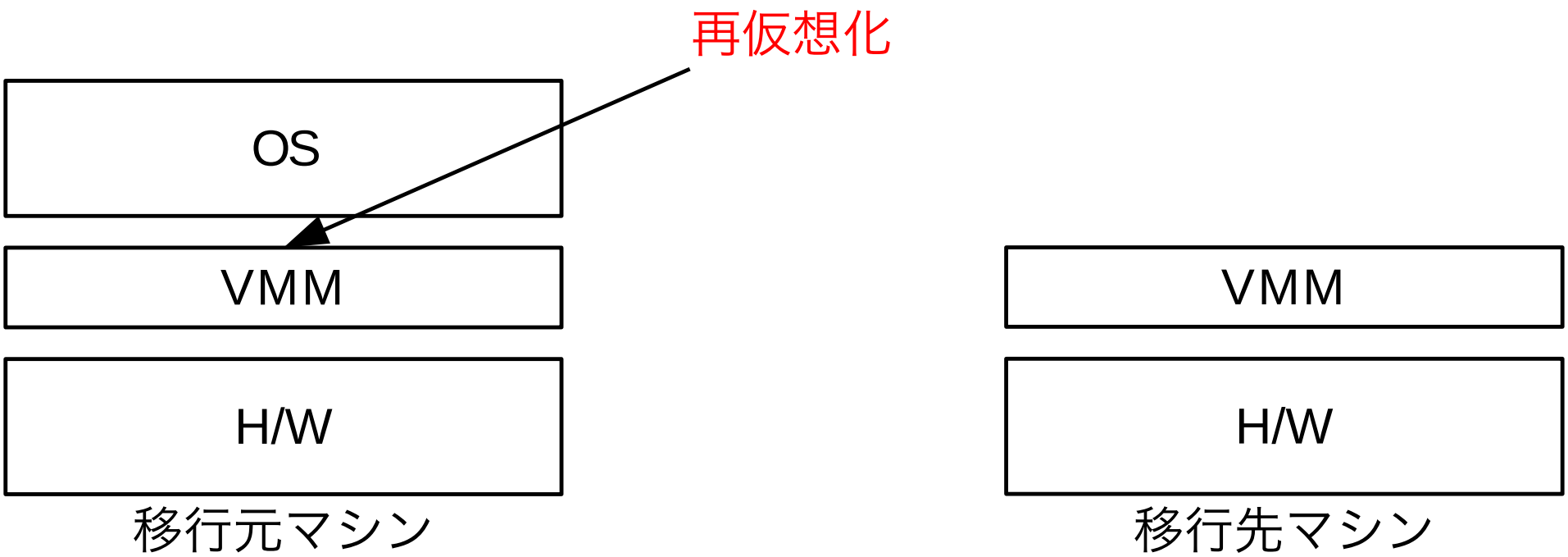
1. PIC,PIT 以外のデバイスへの対応とその評価
   - NIC，ハードディスク，etc. . .
2. マイグレーション時のダウンタイムの削減
   - メモリのバックグラウンドコピーを実装
3. 脱仮想化と再仮想化の利用
   - VMM の動作によるオーバヘッドを削減

今後の課題

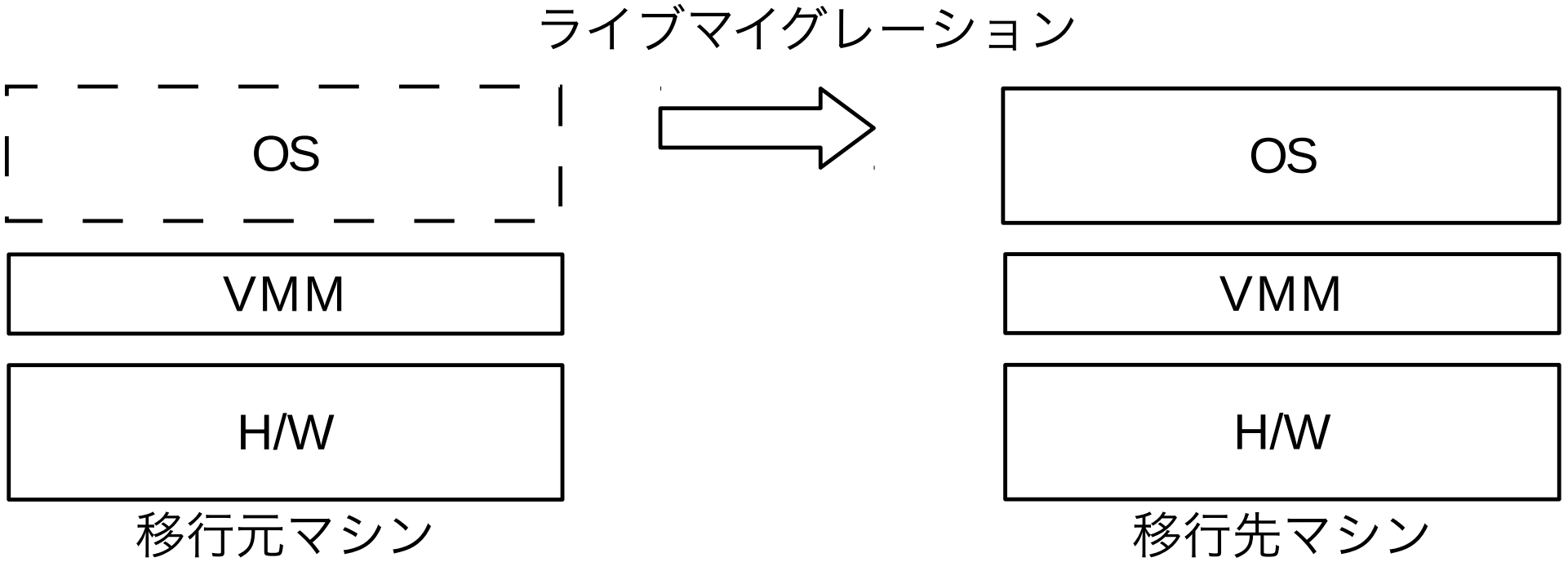
1. PIC,PIT 以外のデバイスへの対応とその評価
   - NIC，ハードディスク，etc. . .
2. マイグレーション時のダウンタイムの削減
   - メモリのバックグラウンドコピーを実装
3. 脱仮想化と再仮想化の利用
   - VMM の動作によるオーバヘッドを削減

脱仮想化と再仮想化の利用

脱仮想化と再仮想化の利用



脱仮想化と再仮想化の利用



脱仮想化と再仮想化の利用