FUJITSU

SCSI support improvement

Jun Kamada <kama@jp.fujitsu.com>
Akio Takebe <takebe_akio@jp.fujitsu.com>
Hitoshi Matsumoto <matsumotohitosh@jp.fujitsu.com>

Fujitsu Limited

Background

SCSI support for guest domain is mandatory feature for server system.

(See http://www.xen.org/files/xensummit_fall07/19_Matsumoto.pdf.)

- ✓ We have developed the pvSCSI driver.
- ✓ Direct I/O (VT-d, IOMMU) is attractive by various reason. (more performance, compatibility to native environment, …)

They can coexist, so …

We concurrently proceed with:

- ✓ The pvSCSI driver improvements (@ Part 1)
- ✓ SAN boot support with VT-d (@ Part 2)

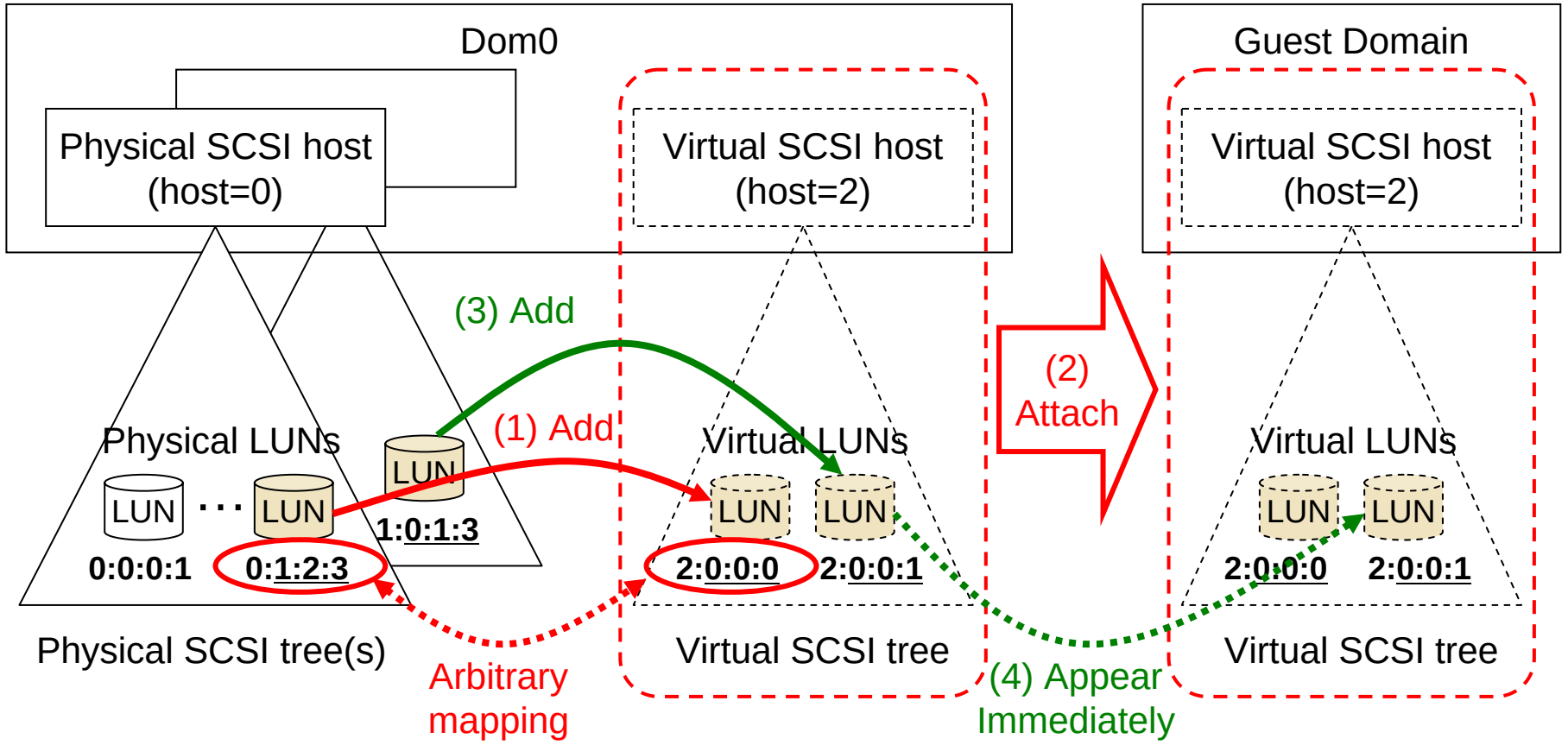
Part 1

The pvSCSI driver improvements
(Request for comments)

Current implementation (Xen 3.3.0)

The pvSCSI driver for Xen 3.3.0 provides:

- ✓ **LUN(Logical Unit Number) pass through**
- ✓ **LUN hot-plug**



Issue of current implementation

Current implementation provides completely virtualized (arbitrarily mapped) SCSI tree to guest domain.

It can provide flexibility, but …

- ✓ Some kind of SCSI commands (REPORT_LUN, EXTENDED_COPY, …) should be emulated on backend.
  (They depend on physical topology of SCSI tree.)
- ✓ A lot of work is needed in order to Implement emulation logic for all the commands, so current implementation supports only mandatory commands.

Does not support full SCSI functionality. :-(

How to solve the issue

1. Implement all emulation logics step by step.
   - Hard work.
   - Cannot support some vendor specific commands, maybe.

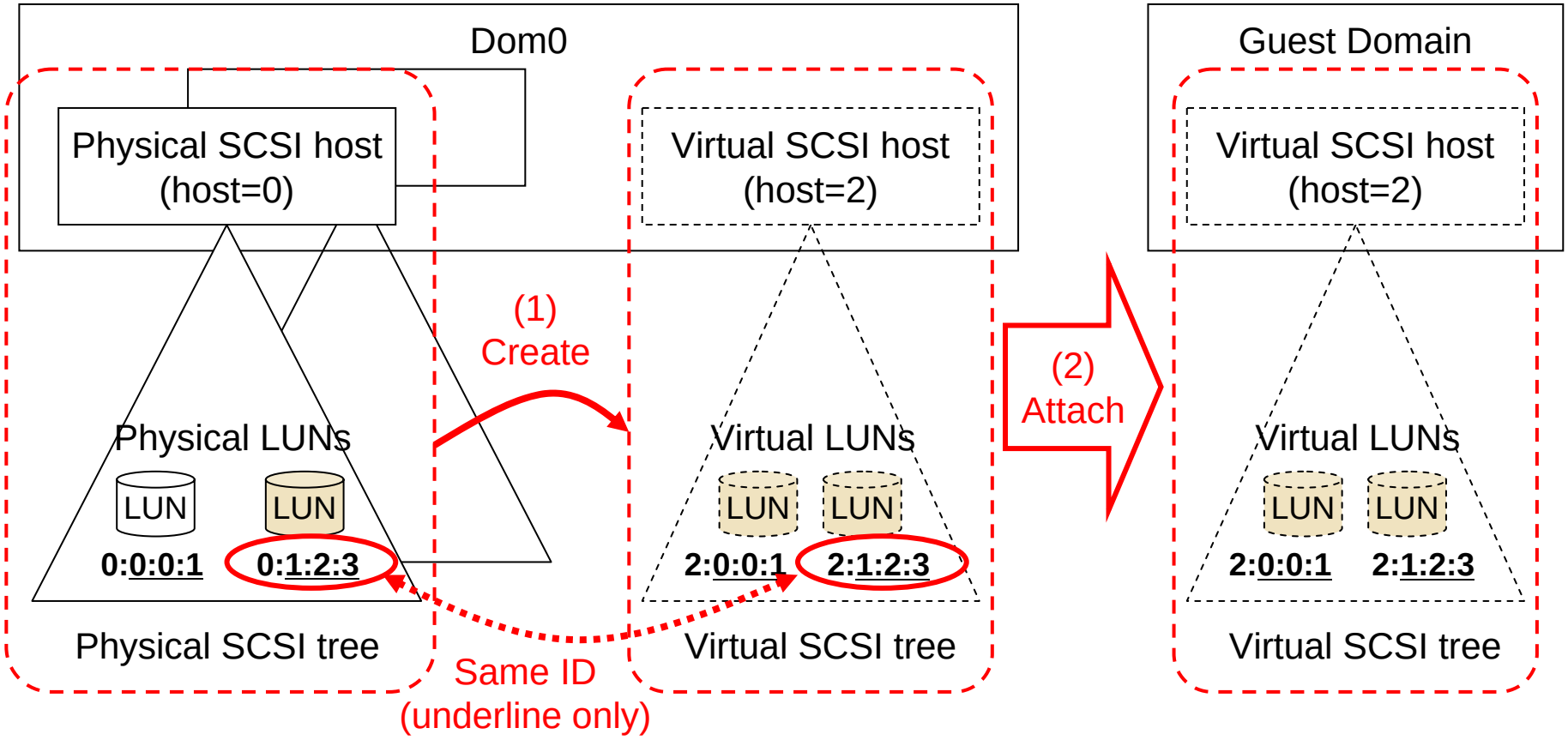
2. "Add" new mode in order to attach whole HBA to guest domain. (It allows bypassing "SCSI command emulation" on backend driver.)
   - Easy to implement. (Details will be shown in following slide.)
   - Can support all vendor specific commands.
   - Can support NPIV(N_Port Id Virtualization).

We propose to take second approach.

Proposed implementation (1/2)

Additional implementation provides:

- ✓ **Host (HBA: Host Bus Adaptor) pass through**



Proposed implementation (2/2)

Modification needed are at most:

- ✓ Backend Driver
  - ➢ LUN/Host mode identification flag for each virtual SCSI tree
  - ➢ Emulation bypassing logic (if the flag shows “host mode”) should be needed. (only one “if” clause (?))
- ✓ Frontend Driver
  - ➢ No need to modify
- ✓ xend
  - ➢ User interface should be modified
  - ➢ LUN scan logic should be added

Conclusion (Part 1)

We will post patches in a few days.

We would appreciate you to send us many comments.

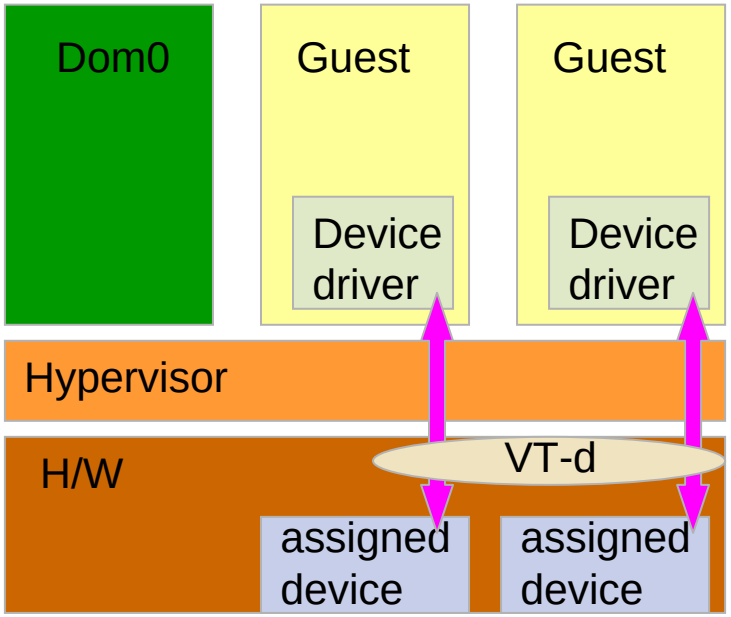
Thanks

Part 2

SAN boot support with VT-d

Introduction
Why do we need SAN boot? (1/2)

VT-d advantages:
- Using the device drivers/applications of native OS
- Isolation
- Performance



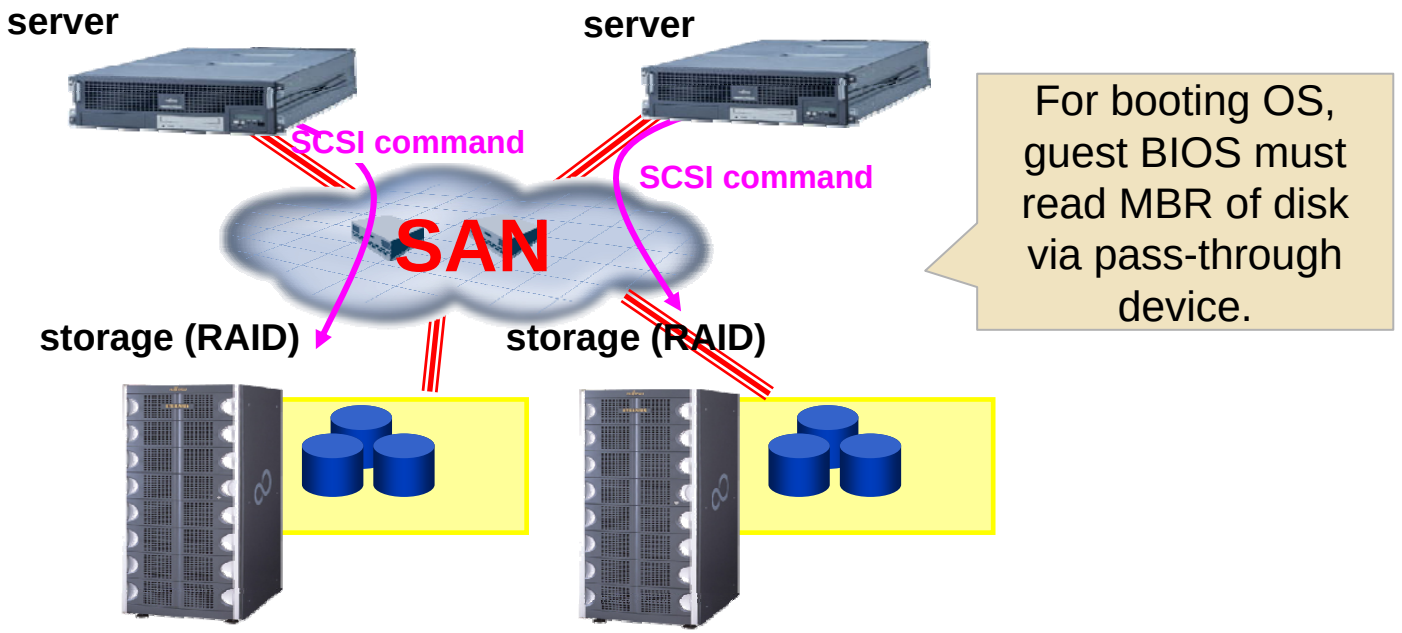
Introduction

Why do we need SAN boot? (2/2)

Current PCI pass-through devices with VT-d are only for “data disk”.
Guest still needs to boot from a emulation disk.

⇨ We want to use pass-through devices for all disk including “boot disk”.

To use the devices as “boot disk”, we need to support some bootable mechanisms which can boot from the devices (e.g. SAN boot) and so on.
If guest can do SAN boot, we don’t need to have emulation disks.



Introduction

What are required for SAN boot?

In the case of SAN boot, BIOS needs to use Expansion ROM of PCI devices for reading MBR. To use Expansion ROM of PCI devices, Guest BIOS needs:

- ✓ Support PCI Firmware Spec and BIOS Boot spec
  - ✓ Support the calling convention of PCI 3.0
  - ✓ Support the calling convention of PCI legacy device
  - ✓ Support the calling convention of BEV style in BIOS Boot spec
  - ✓ Support the calling convention of BCV style in BIOS Boot spec
- ✓ Support BIOS services used by Expansion ROM
  - ✓ interrupt function like “int 0x10”, “int 0x16” …
  - ✓ I/O port accesses

BCV: Boot Connection Vector. It’s typically used by SCSI controller.

BEV: Boot Entry Vector. It’s typically used by Ethernet controller.

Current Issues

- Guest BIOS/qemu-dm support booting from only emulation devices.
- Guest BIOS doesn’t support full BIOS services. So Expansion ROM may use unsupported services.
- There are several kinds of calling convention for Expansion ROMs…(PCI 3.0 spec, legacy spec, PnP Boot BIOS spec, some mix spec and so on)
- BIOS spec is not clear so much…

Enhancement plan

1st Step

- ✓ Improve rombios for supporting BCV style
  - ✓ search Expansion ROM image from PCI devices.
  - ✓ sanity check (checksum, device id…)
  - ✓ map Expansion ROM image to 0xd0000
  - ✓ support BCV style calling convention of Expansion ROM
- ✓ modify qemu-dm for SAN boot

2nd Step

- ✓ Support other boot specs
- ✓ implement unsupported BIOS services into rombios

Design concept (1/3)

As the 1st Step, we Support booting with the calling convention of BCV style in BIOS Boot Spec.

Why do we choice the convention of BCV style?

- Most PCI devices should support the BCV style.
- The calling convention is simple.

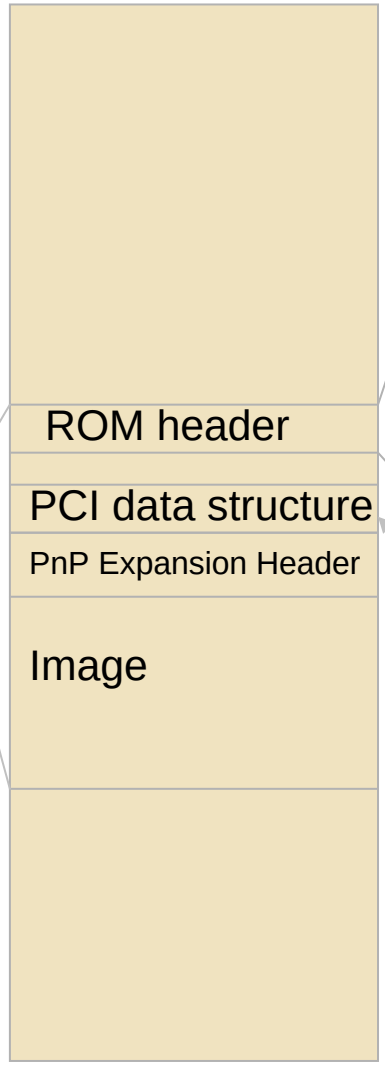
What is BCV style?

BCV style is a way that can read a disk by installing a device specific handler of INT 0x13(disk read). BCV is a pointer that points to code inside the Expansion ROM. By using the code, BIOS can install a special INT 0x13 handler at the initialization.

Design concept (2/3)

How to initialize Expansion ROM

1. hvmloader map the Expansion ROM to 0xd0000



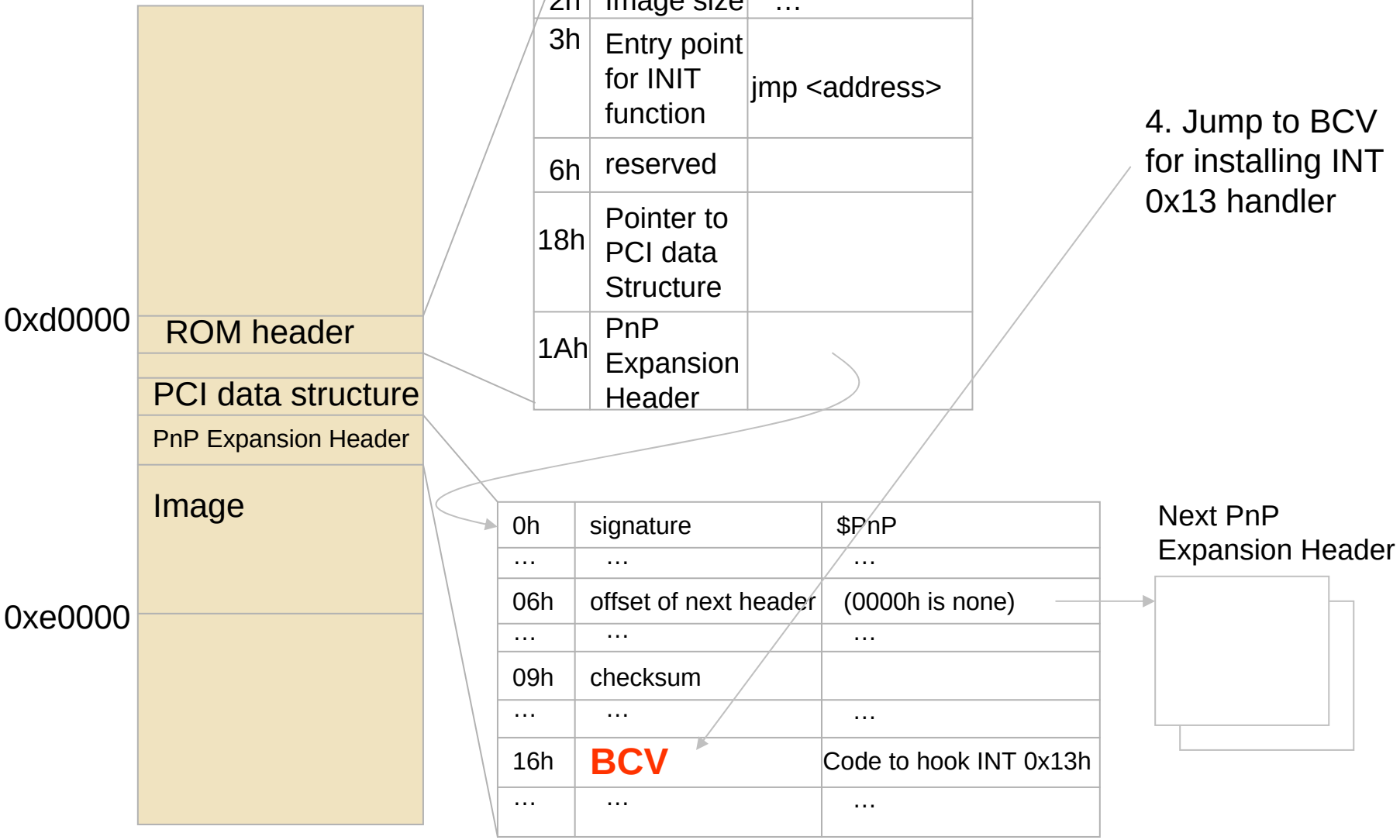
0h	signature	0xaa55
2h	Image size	…
3h	Entry point for INIT function	jmp <address>
6h	reserved	
18h	Pointer to PCI data Structure	
1Ah	PnP Expansion Header	

2. hvmloader,rombios do sanity check

3. rombios jump to Entry point for INIT function after supplying ax register with bus:dev:function number.

Design concept (3/3)

How to initialize Expansion ROM



Sample

Current Status

1st Step

- ✓ Improve rombios for supporting BCV style [Done(prototype)]
  - ✓ We can do SAN boot.
- ✓ modify qemu-dm for SAN boot [WIP]
- ✓ clean up patch [WIP]

2nd Step

- ✓ Support other boot spec
- ✓ implement unsupported BIOS services into rombios

This work was partly funded by Ministry of Economy, Trade and Industry (METI) of Japan as the Secure Platform project of Association of Super-Advanced Electronics Technologies (ASET).

Q&A

Any questions?