redhat.

NVDIMM and PMEM overview

What is it? How does it work? How does it benefit virtualization?

Marc Marí Barceló <markmb@redhat.com>
12/10/2015

WHAT ABOUT ME?

- Red Hat Virtualization Team
- Working on making QEMU faster

You can find me as markmb

AGENDA

Introduction

Project

Namespaces

NFIT

Virtualization

INTRODUCTION

About the name

N

V

D

I

M

M

INTRODUCTION

About the name

Non

Volatile

Double

In-line

Memory

Module

INTRODUCTION

About the name

Non

Volatile

Double

In-line

Memory

Module

DIMM: This is the HW format

http://www.simmtester.com/page/news/images/Samsung_DDR4_DIMM.jpg

INTRODUCTION

About the name

Non
Volatile

Non-volatile: you plug it off and on again, and the Information is still there

Double

In-line

Memory

Module

INTRODUCTION

Why?

VS

IBM 305 RAMAC (1958):

- 5 MB
- 1 ton
- 3200$ / month
- R/W 1 char: 96 µs

Samsung SSD (2015?):

- 1 TB
- 2.5 inch
- 350$
- 500 MB/s

http://www.nealcampbell.com/wp-content/uploads/2012/05/IBM-305-RAMAC-First-Hard-Disk-Drive.png
https://en.wikipedia.org/wiki/IBM_305_RAMAC
http://www.amazon.com/Samsung-2-5-Inch-Internal-MZ-75E1T0B-AM/dp/B00OBRFFAS/

redhat.

INTRODUCTION

Future

THE FUTURE: NVDIMM

INTRODUCTION

Future

THE FUTURE: NVDIMM

?

INTRODUCTION

Virtualization

Why is useful in virtualization?:

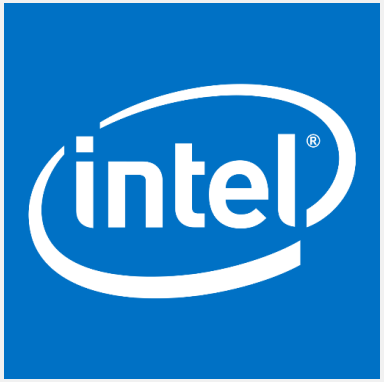
- Pass-through of NVDIMM devices to the guest:
  - Direct access
  - No vm-exits
- Direct access to host files via emulated NVDIMM:
  - Page cache bypass
  - Better performance
- NVDIMM emulation from host file:
  - NVDIMM testing on guests

PROJECT

pmem.io

NVM Library team:

- Intel
- Led by Andy Rudoff
- 25/08/2014 – pmem project created
- Persistent memory programming easier for developers



https://pbs.twimg.com/profile_images/532732477320593408/_rEfoNQ3.png
http://pmem.io/about/
http://pmem.io/2014/08/25/creating-pmem-io.html

redhat.

PROJECT

pmem.io

Userspace libraries:

- libpmemobj: object store in pmem
- libpmemblk: arrays of pmem-resident blocks
- libpmemlog: pmem-resident log
- libpmem: low level pmem support
- libvmem: volatile memory pool
- libvmmalloc: transparent conversion from dynamic memory allocation to pmem allocation

redhat.

http://pmem.io/nvml/

PROJECT

libnvdimm

libnvdimm kernel library + libndctl userspace helper library

- Sent around April this year (merged):
  - https://lkml.org/lkml/2015/4/13/174 - PMEM driver
  - https://lkml.org/lkml/2015/4/18/139 - NFIT-Defined / NVDIMM Subsystem
  - Other patch series

PROJECT

vNVDIMM

Virtual NVDIMM for QEMU

- Not merged yet:
  http://lists.gnu.org/archive/html/qemu-devel/2015-08/msg01774.html
- Only PMEM access mode is implemented

PROJECT

Specs

Drivers and this presentation are based on specs:

- http://pmem.io/documents/
  - NVDIMM Namespace Specification: http://pmem.io/documents/NVDIMM_Namespace_Spec.pdf
  - NVDIMM Drivers Writers Guide: http://pmem.io/documents/NVDIMM_Driver_Writers_Guide.pdf
  - NVDIMM DSM Interface Example: http://pmem.io/documents/NVDIMM_DSM_Interface_Example.pdf
- ACPI 6: http://www.uefi.org/sites/default/files/resources/ACPI_6.0.pdf
- Linux docs: https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/nvdimm/nvdimm.txt
- And code

NAMESPACES

What are they?

Divide the NVDIMMs into logic units of storage.

Like SCSI LUNs

Two types:

- Persistent memory namespaces
- Block mode namespaces

NAMESPACES

Types

Persistent memory namespaces:

- Accessed using loads and stores
- Mapped to physical memory
- Problem: when is the data really persistent? Caches, buffers…
  - New instructions such as CLWB and PCOMMIT
  - Flushing cache but not deleting from cache

NAMESPACES

Types

Block mode namespaces:

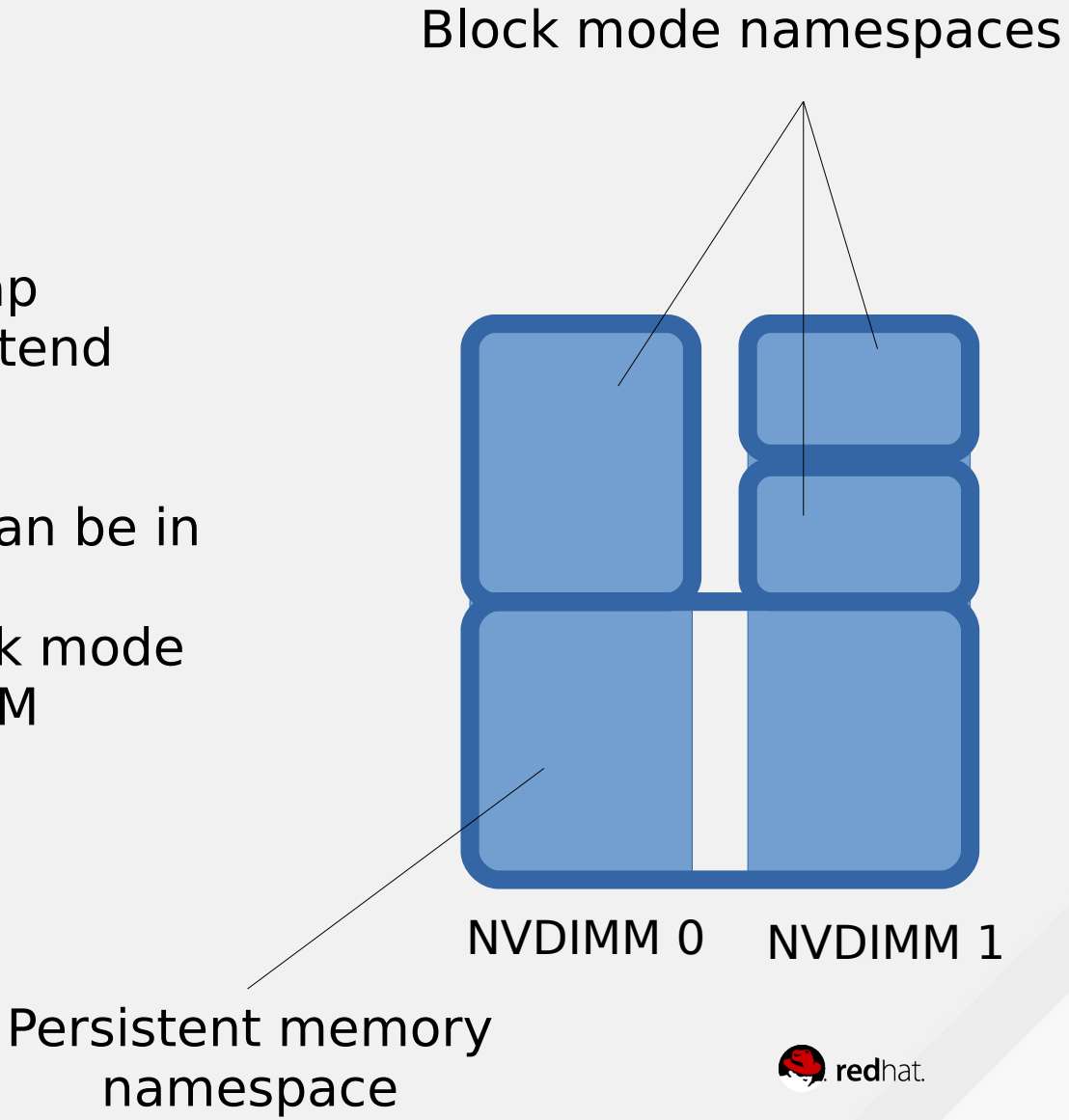
- Accessed using block operations
- Atomicity at block level: in case of power failure when writing, it can be rolled back or rolled forward.
- Indirect access through a Block Window:
  - No mapping the entire memory
  - Reduce address utilization
  - Reduce risk of wrong addressed writes

NAMESPACES

What are they?

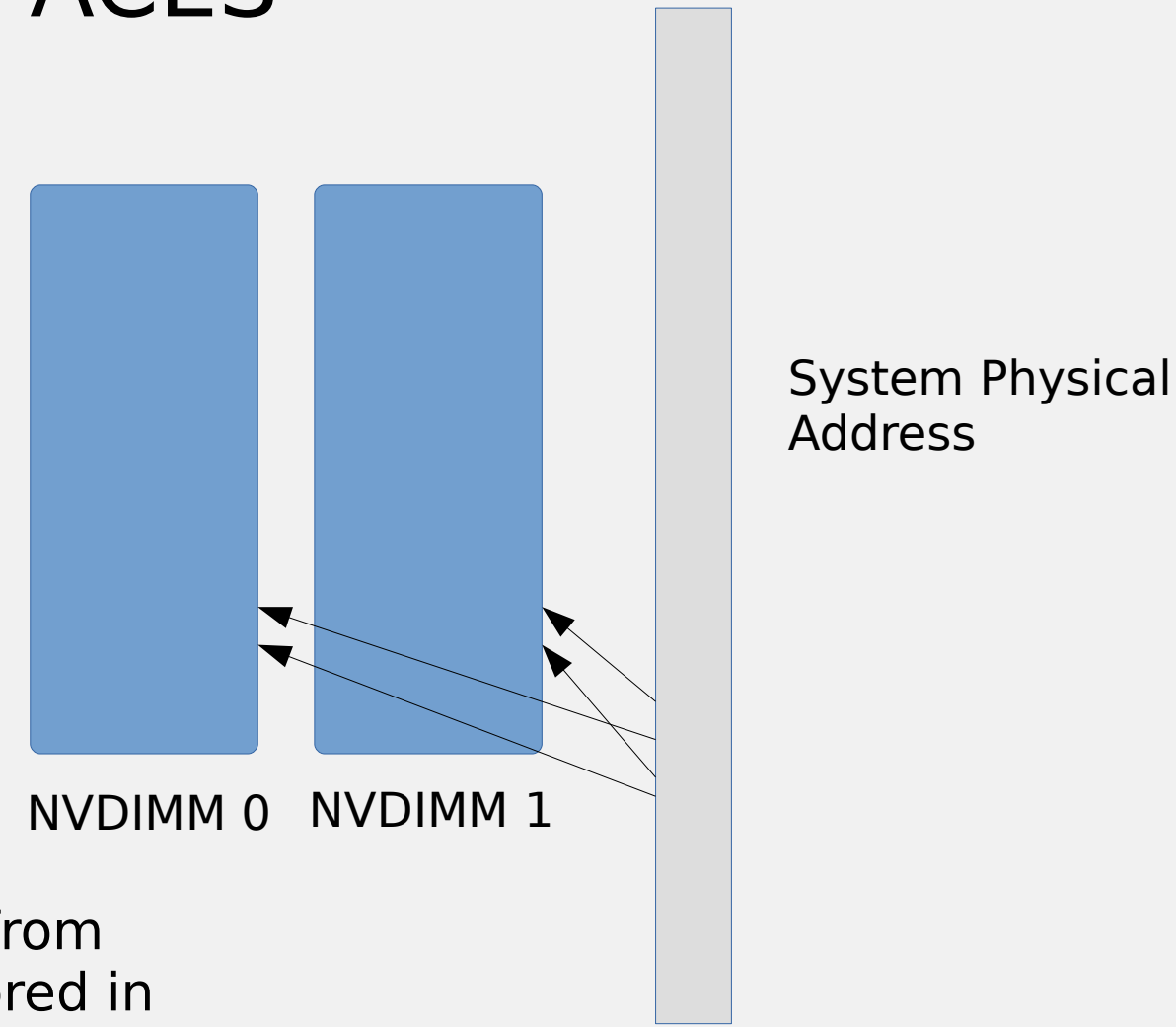
Rules:

- Namespaces cannot overlap
- Pmem namespaces can extend across multiple NVDIMMs (interleaving)
- Block mode namespaces can be in just one NVDIMM
- There can be multiple block mode namespaces in one NVDIMM



NAMESPACES

Interleaving



No visible from guest. Ignored in virtualization

NAMESPACES

Block Window

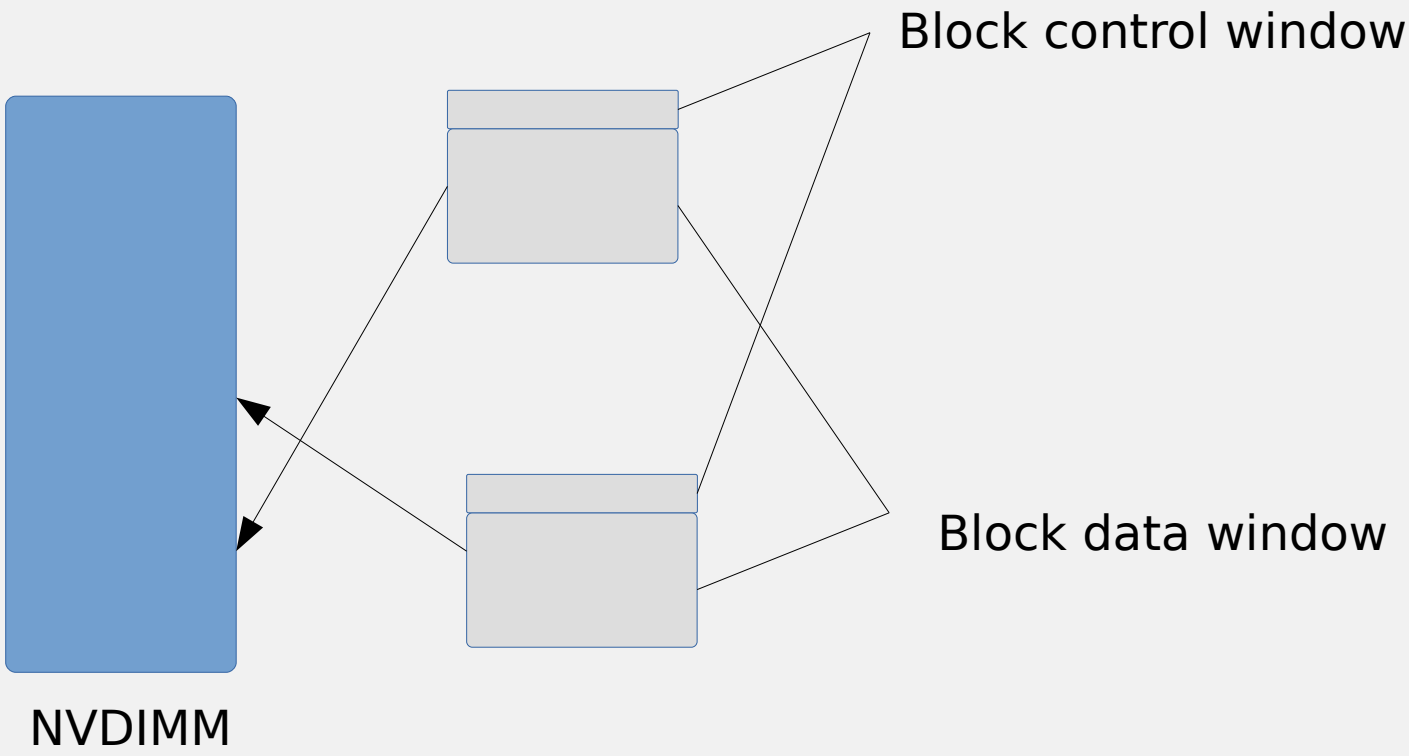
Block windows:

- Block control window:
  - Command/Address register: where to read/write (in device addresses)
  - Status register: status of the operation
- Block data window:
  - The block to be written/the block read

Different block windows can operate in parallel

NAMESPACES

Block Window



NAMESPACES

Block Translation Table

Atomicity at block level in block namespaces.

Block namespaces broken into arenas (up to 512GB):

- Arena info block (backup)
- Arena flog
- Arena map
- Arena data area
- Arena info block

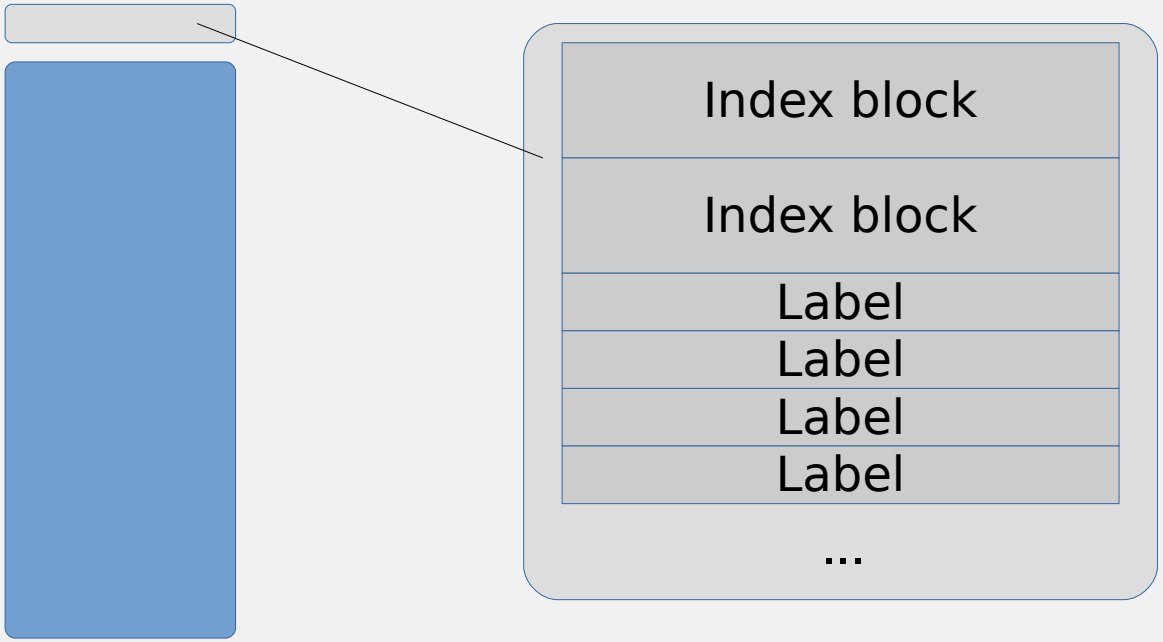
https://github.com/pmem/nvml/blob/master/src/libpmemblk/btt.c

NAMESPACES

Where are they?

Label Storage Area

- Minimum of 128 KB



redhat.

NFIT

What is it?

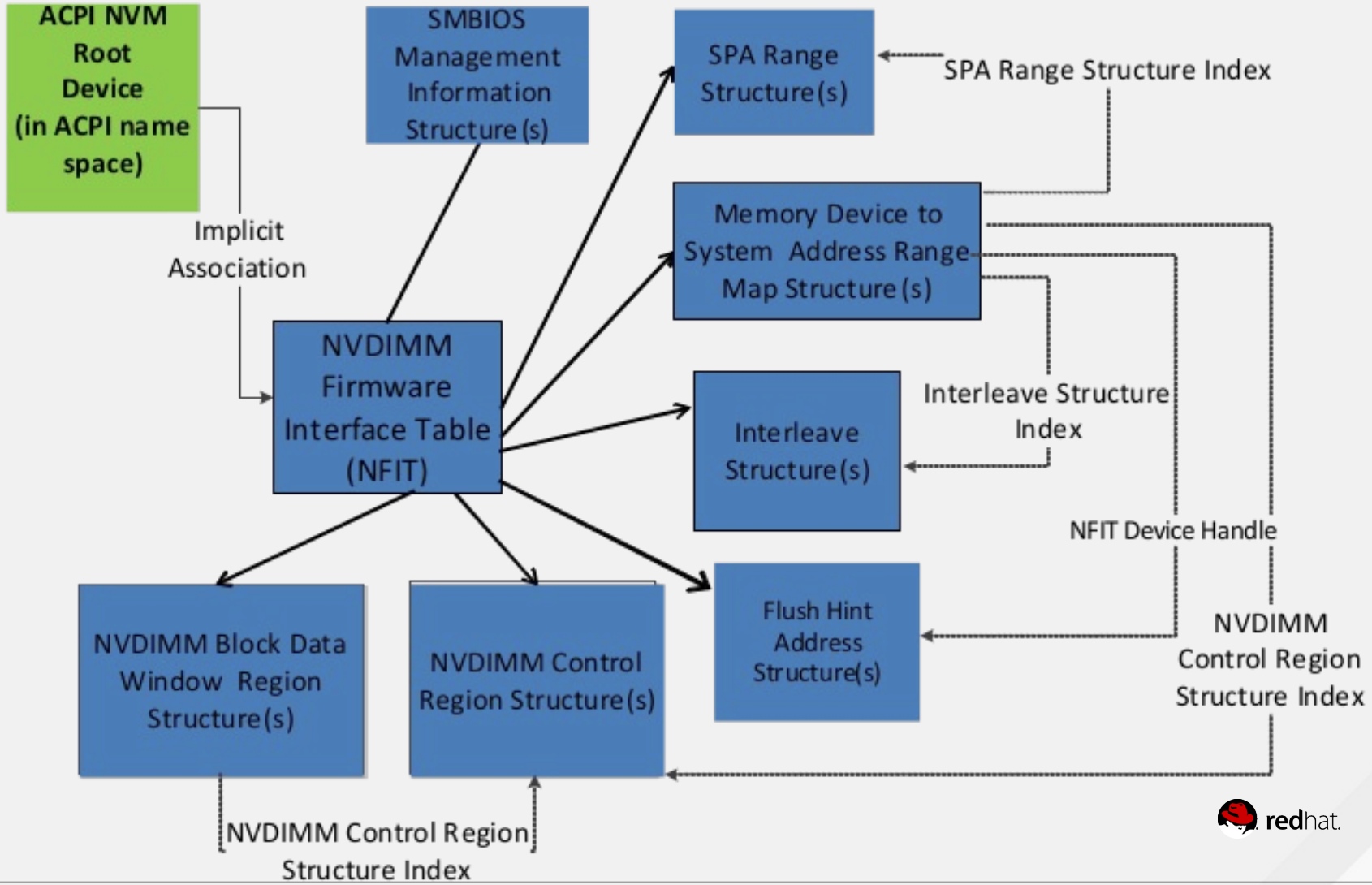
ACPI table

NFIT: NVDIMM Firmware Interface Table

- Standardize access to NVDIMM information and namespace across vendors
- Filled through ACPI _DSM
  - Get SMART information
  - Get and set Namespace Label Data
  - Get Vendor-Specific Command Log
  - Vendor-Specific commands

NFIT

NFIT Table



NFIT

Structures

- SPA Range Structure: defines memory mapped region address, length and properties
- Memory Device to System Address Range Map Structure: memory device properties
- Interleave Structure: defines interleaving properties
- SMBIOS Management Structure: SMBIOS is out of the scope
- NVDIMM Control Region Structure: hardware identifiers, and block control window stuff
- NVDIMM Block Data Window Region Structure: block accesses
- Flush Hint Address Structure: flush pending operations

NFIT

Structures

WARNING:

MemDev Region Size != SPA Range Length

Depends on interleaving!

NFIT

Structures

SPA Range Structure GUID: describe the Address Range Type

- Volatile memory region
- Persistent memory region
- Control region
- Block Data Window Region
- Virtual Disk Region – Volatile
- Virtual Disk Region – Persistent
- And others
- And vendor-defined

VIRTUALIZATION

Linux

Just new block devices:

- /dev/pmem → PMEM namespace
- /dev/nd_blk → Block namespace
- /dev/btt → If it has BTT

- Accessed with usual IO calls
- PMEM also with direct access (load/stores)

pmem.io libraries provide NVDIMM-friendly transactions

VIRTUALIZATION

QEMU

Pending merge:

- PMEM virtualization (no interleaving)

Missing:

- BLK mode: very inefficient, would cause at least two vm-exit
- NUMA support
- Hotplug support

VIRTUALIZATION

QEMU

Creating a NVDIMM device:

```
-device pc-nvdimm,file=/tmp/nvdimm
```

What happens with namespaces?

- `-device pc-nvdimm,file=/dev/pmem,configdata`
  Uses device namespaces
- `-device pc-nvdimm,file=/tmp/nvdimm`
  Creates a readonly namespace in memory

VIRTUALIZATION

SeaBIOS

Steps to boot from an NVDIMM PMEM namespace:

1) Find NFIT table

2) Search for PMEM namespaces. Looking at SPA tables is enough

3) Get the address and length of the area

4) Check Linux magic numbers and flags

5) Copy kernel to low memory

6) Boot normally

Preliminary implementation:

http://www.seabios.org/pipermail/seabios/2015-September/009770.html

redhat.

QUESTIONS?