China Linux Kernel
Developer Conference
Oct 14, 2018

FUJITSU
shaping tomorrow with you

The ideal and reality of NVDIMM RAS

Yasunori Goto / QI Fuli
Linux Development Division
Fujitsu Limited

Agenda

FUJITSU

- Introduction of NVDIMM
- Issues of RAS which we found and solved
  - Distinguish and replace broken NVDIMM (by Yasunori Goto)
  - Monitoring feature of NVDIMM (by Qi Fuli)

FUJITSU

Introduction of NVDIMM

NVDIMM is coming

FUJITSU

- Non Volatile DIMM (NVDIMM)
  - Some vendor released / will release NVDIMM
    - Intel Optane DC persistent memory
  - CPU can read/write NVDIMM directly like RAM
  - Data is not volatile even if system is powered down
  - Though its access latency is a little slower than DRAM, it is expected to be very faster than NVMe
  - May be huge amount (*)
- Use case
  - For example, On memory Database

(*) It depends on type of NVDIMM

Impact of NVDIMM

FUJITSU

- Traditional I/O layer is not suitable for NVDIMM
  - Page cache is not necessary at least
    - It was created for SLOW I/O storage
    - But, it is just redundant for NVDIMM
  - Even system call may be waste of time due to switch kernel mode and user mode
    - Ideal is that application can access to NVDIMM directly without kernel
  - If application calls cpu flush instruction, then data becomes persistent
    - calling sync()/fsync()/msync() becomes redundant
- To achieve the above feature, New access method is expected
  - DAX (Direct Access Mode) is developed in Linux

Filesystem is still useful for traditional software

FUJITSU

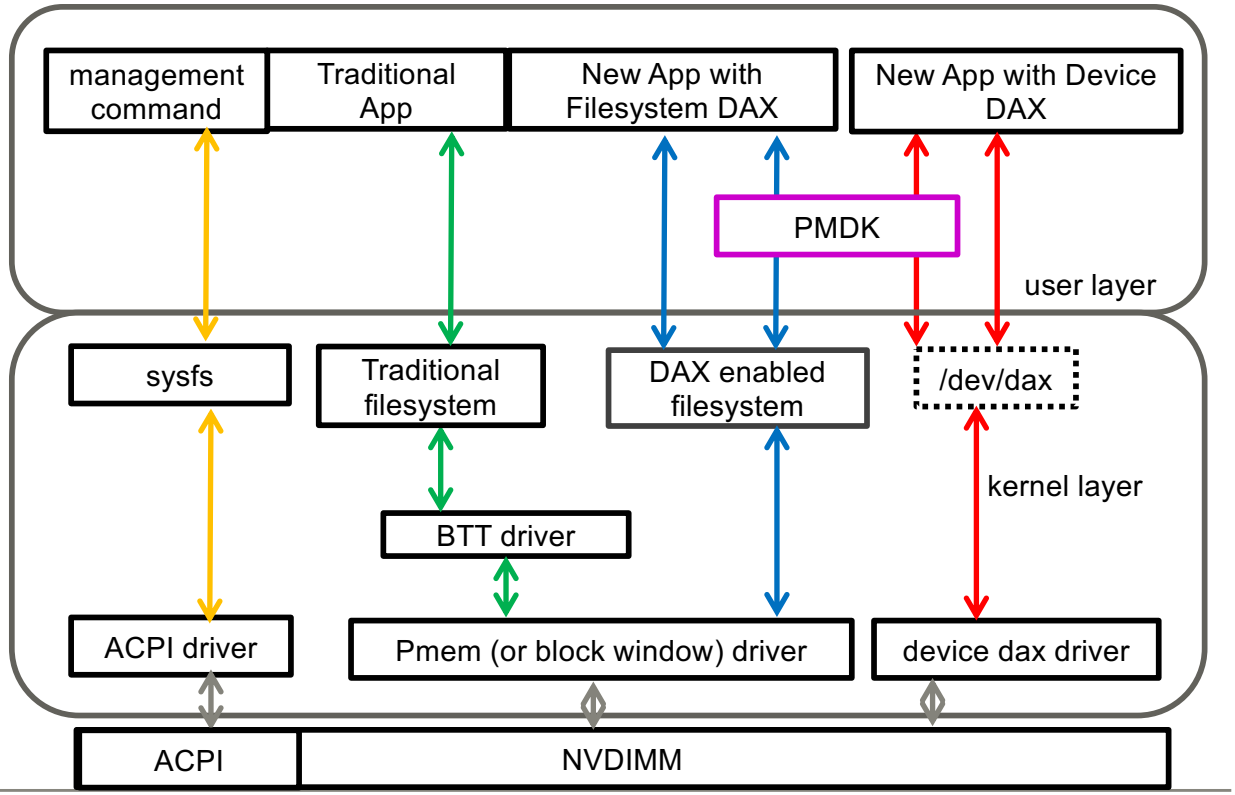
- On the other hand,
  Many software assumes that memory is VOLATILE
  - They allocate memory areas on the RAM and make structures on them
  - When allocate memory area on the NVDIMM like RAM, what will happen?
  - For NVDIMM, many things are necessary. For example
    - Need data structure compatibility on NVDIMM
      - Should not change data structures in NVDIMM
      - If the structures are changed, software update will be cause of disaster
    - Need to detect / correct collapsing data
      - CPU cache is still volatile. If system power down suddenly, then some data may not be stored
      - If the data is broken, software need to detect it and correct it
- Filesystem is still useful for traditional software
  - Filesystem gives many solution for the above considerations
    - Format compatibility of filesystem, data correction, region management, , authority check, etc...

Interfaces of NVDIMM

FUJITSU

- Linux provides some interfaces for application
  - Storage Access

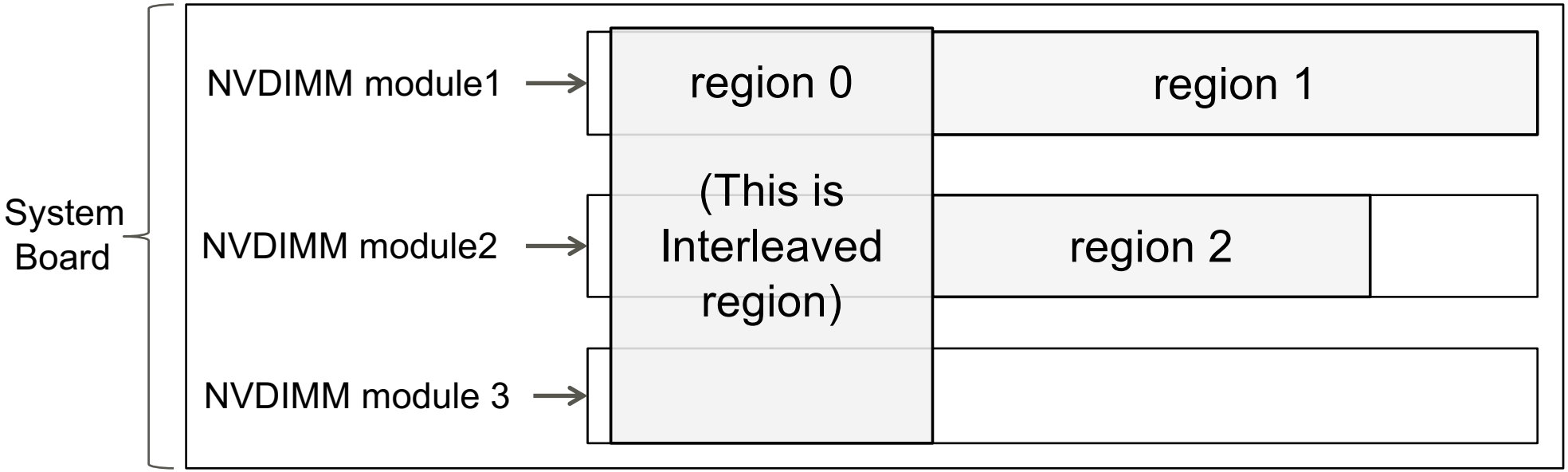
    (*) DAX= Direct Access Mode
    - Application can access NVDIMM via traditional I/O
    - This mode is for old application
  - Filesystem DAX(*) and Device DAX
    - Application can read / write NVDIMM area directly if it calls mmap()
    - Need modification or making new software
  - PMDK is provided
    - libraries and tools for software which uses DAX



Region and Namespace(1/2)

FUJITSU

- To manage huge amount of NVDIMM, “Region” and “namespace” is defined
  - Region
    - Region is created by hardware and firmware feature
    - Each NVDIMM modules are placed on a System Board(Mother Board)
    - You can create regions on NVDIMM modules
      - Interleave is available

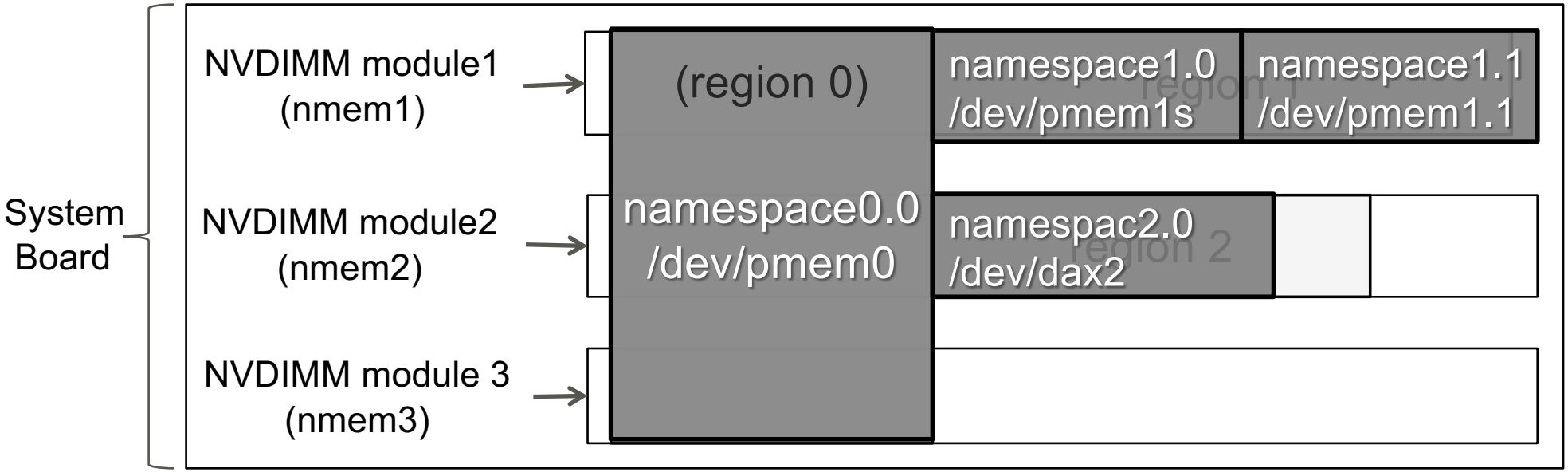


Region and Namespace (2/2)

FUJITSU

Namespace

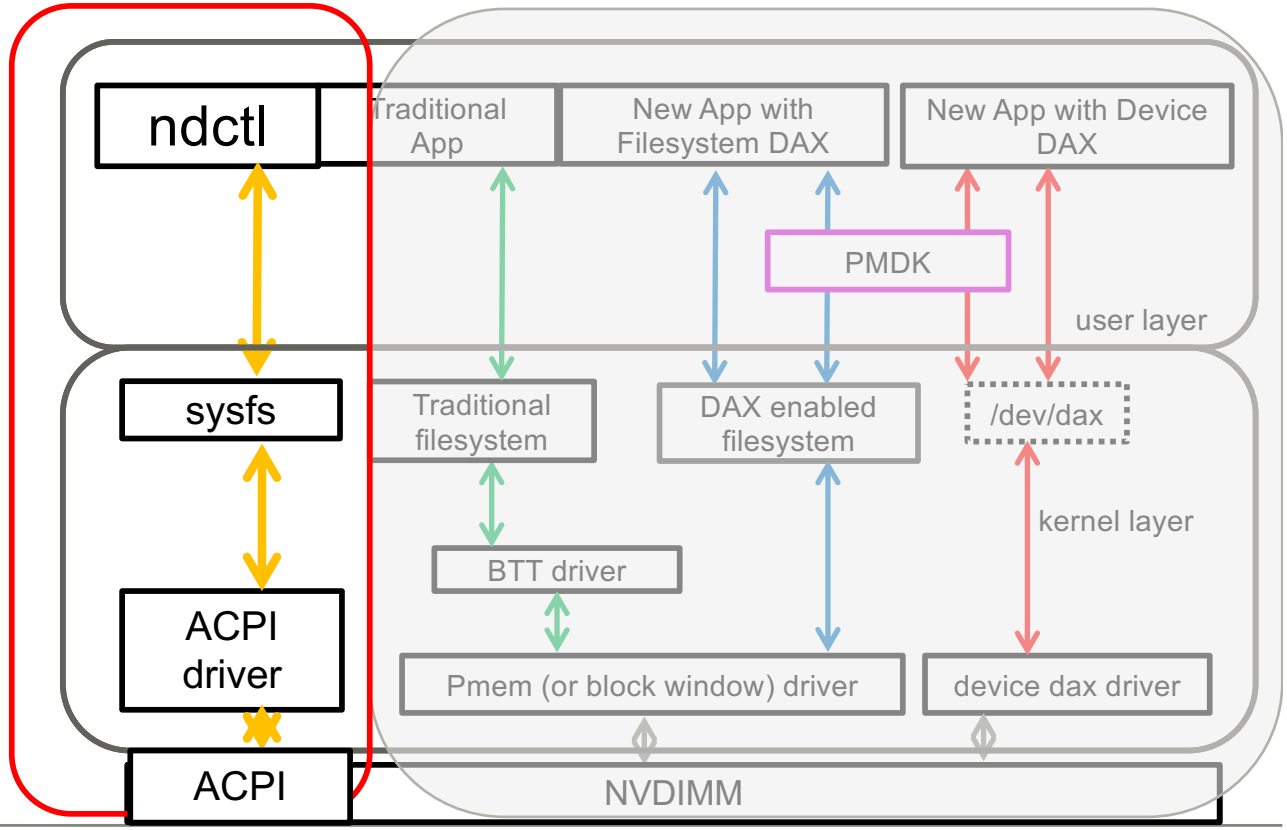
- You can create namespaces on each region by ndctl command
  - Namespace concept is likely LUN of SCSI
  - You can make plural namespaces on a region, if you wish
  - You can specify storage, filesystem-dax, or device dax at this time
- NVDIMM driver names each NVDIMM module “nmemX”
- /dev/pmemX or /dev/daxY is named for each namespace
  - If Storage mode or Filesystem DAX, you can make partition and filesystem on the /dev/pmemXXX



Management command

FUJITSU

- ndctl
  - Many features
    - Create / Manage namespaces
    - Set access mode (storage / Filesystem DAX/ Device DAX)
    - Show status of NVDIMM modules, regions, namespaces
      - Support difference among NVDIMM vendors (NVDIMM-N or 3D-xpoint...)
    - etc..
  - It use ACPI via sysfs and acpi driver for NVDIMM (nfit.ko)
- We add some feature in it for RAS



ACPI spec for NVDIMM

FUJITSU

- ACPI v6.0 or later supports specification for NVDIMM
  - NFIT table is defined for NVDIMM
    - Many sub tables are specified for NVDIMM basic information
      - (Very huge table)
  - In DSDT
    - Some specification of NVDIMM is defined
    - “NVDIMM root device”
      - defined for whole system
      - Its _DSM method has some features like Address Range Scrubbing
      - Notification is used for NFIT update or Memory error detection
    - “NVDIMM device”
      - defined for each NVDIMM modules
      - Its _DSM method is used for manage namespace or Health (SMART) information, etc.
      - If health status is changed, firmware notifies a event to the “NVDIMM device” in DSDT
      - There is a little bit difference among vendors
        (Intel) http://pmem.io/documents/NVDIMM_DSM_Interface-V1.7.pdf
        (HPE) https://github.com/HewlettPackard/hpe-nvm

FUJITSU

Replacement of NVDIMM

- Issues and what I solved

NVDIMM may be broken

FUJITSU

- We need to consider when NVDIMM becomes broken
  - The life of NVDIMM is limited
    - Write endurance of NVDIMM is / will be not infinite
    - NVDIMM module has / may have some feature
      - wear leveling
      - spare blocks instead of broken block
      - If spare block is consumed completely, the NVDIMM modules can not recover
    - NVDIMM-N has a battery in the module
      - It will be also deteriorated
  - Bit error may also occur on NVDIMM
    - “Scrubbing” feature may rescue it from fatal error if possible
      - If it can not rescue, the block becomes broken

We need to have a way to replace broken NVDIMM

Question

- Please imagine a difference between HDD/SSD and NVDIMM
  - How to replace its device?

Answer

FUJITSU

- There is a difference between SSD/HDD and NVDIMM replacement like the followings

HDD/SSD
- A disk unit is removable without opening the cover of box
- Hot-replace is enabled by hardware (hotplug slot, electric switch, etc)
- Device Mirroring is helpful for replacing the device

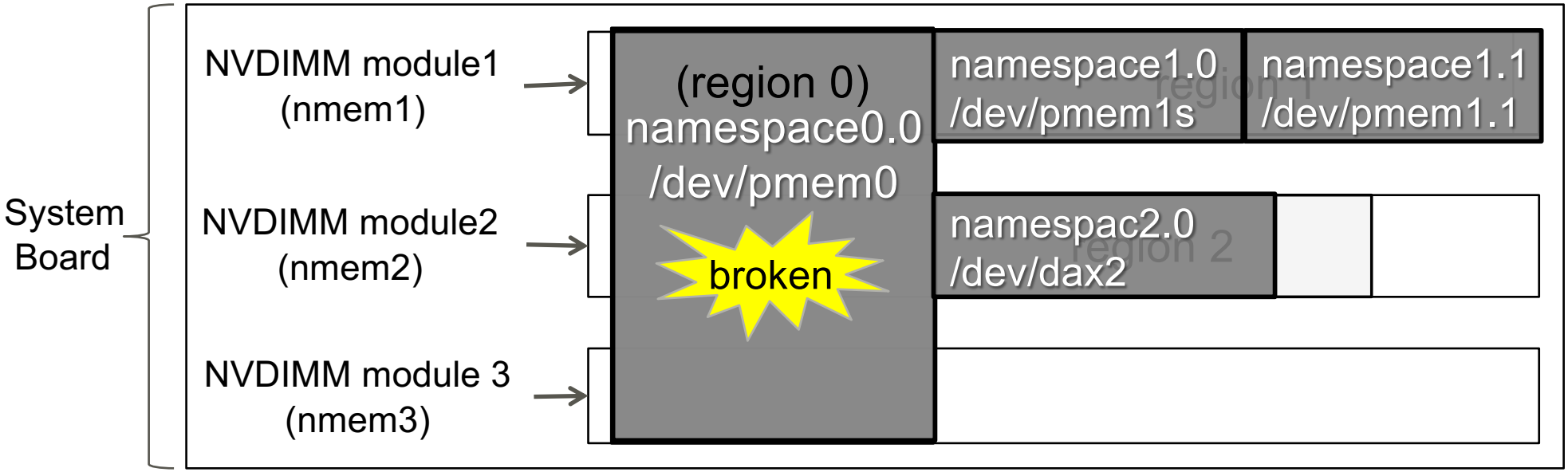
NVDIMM
- You need to open the cover of box to replace NVDIMM
- Hardware does NOT support hot-replace for NVDIMM
- No hardware mirroring for NVDIMM (at least, currently)
  - In addition, you can not use Software RAID with Filesystem DAX or Device DAX

Replacing of NVDIMM is more difficult than SSD/HDD

Replacing broken NVDIMM is complex (1/2)

FUJITSU

- In addition, the specification of NVDIMM (region and namespace) is cause of more difficulty of replacement
  - In this figure, when a block on region 0 is broken eternally, and user need to replace it, what information is necessary?



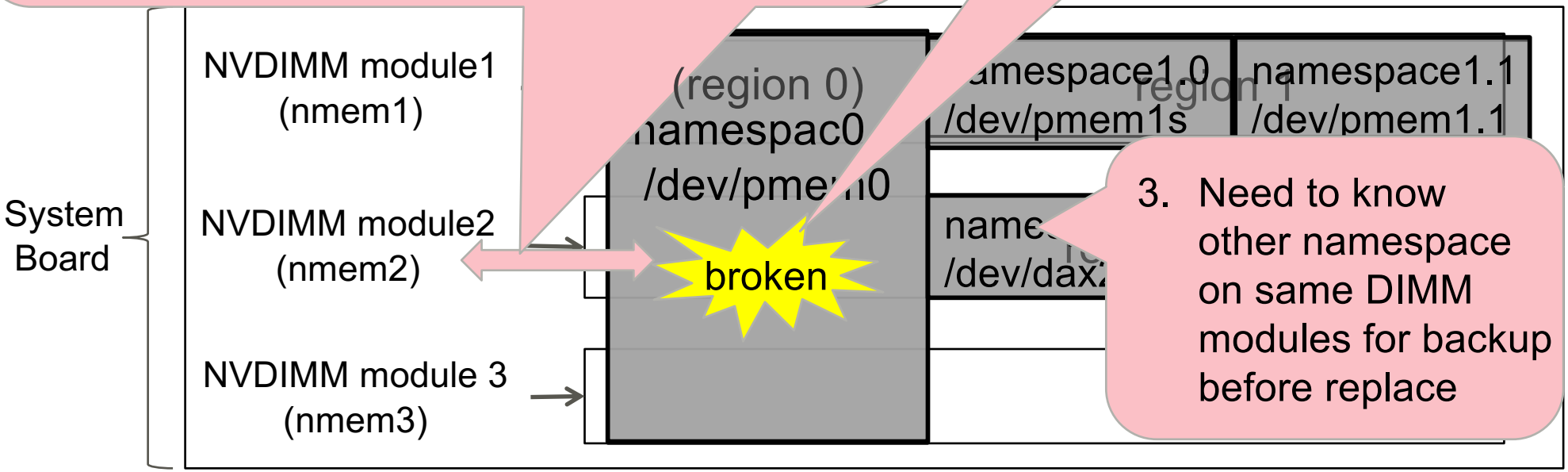
Replacing broken NVDIMM is complex(2/2)

FUJITSU

- To replace broken NVDIMM, the following information is necessary, at least

1. Need a way to distinguish which NVDIMM module is broken (especially, if the region is interleaved, only firmware know it)

2. To replace NVDIMM, Need information of relationship between "Physical" location of NVDIMM module (Ex. Silk print on mother board), and Device name of namespace /dev/pmemX)



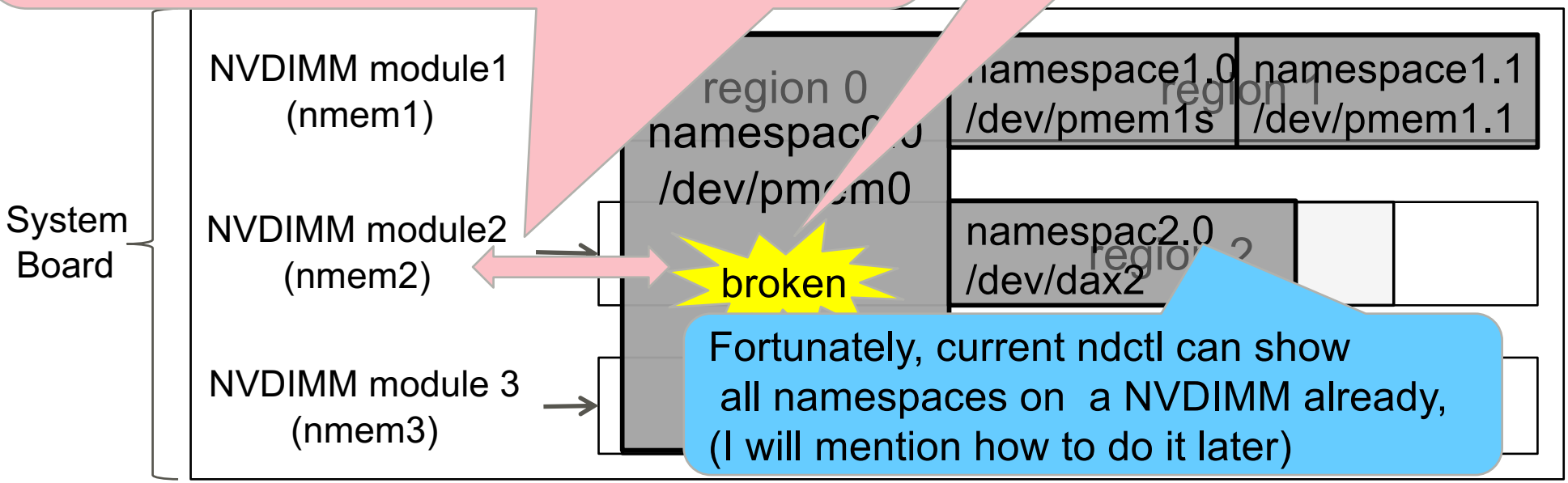
What I developed

FUJITSU

- I made 2 features of ndctl
  - To show important information for replace

1. Show NVDIMM info which has broken block in the region

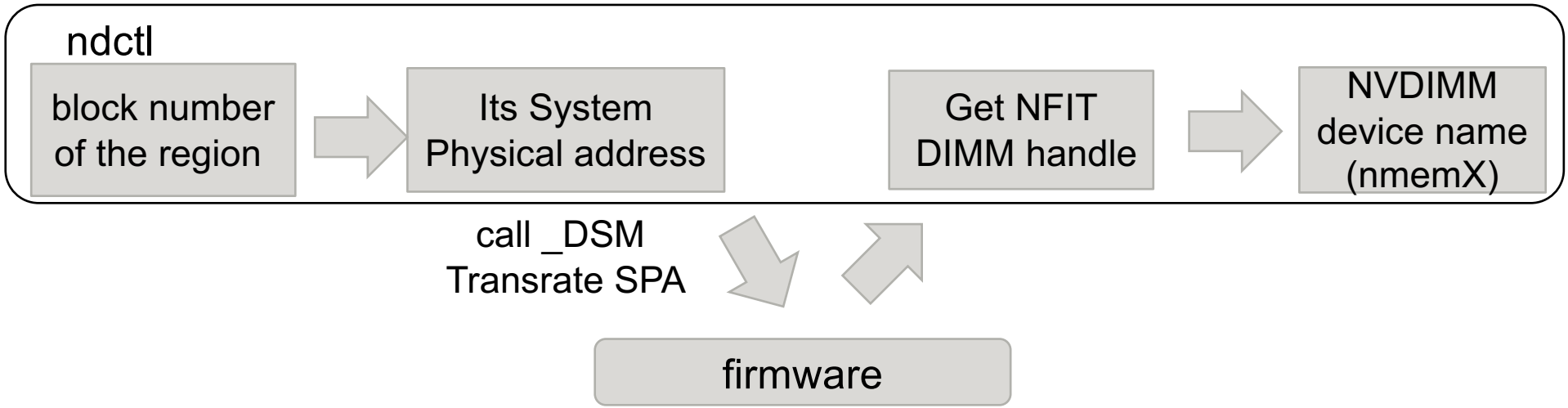
2. Show the id of NVDIMM module to find physical location



To show broken NVDIMM

FUJITSU

- ndctl did not show broken NVDIMM module information
  - It just showed broken “block number” in the region
  - When the region is Interleaved, kernel/driver cannot distinguish NVDIMM
    - Only firmware know the relationship between physical address and NVDIMM module
  - Fortunately, “Translate SPA” is defined in _DSM method of “NVDIMM root Device” at ACPI ver.6.2
    - Firmware translates from System Physical Address to “NFIT DIMM handle” and Physical address in the DIMM(DPA)
- I made the following translation in the ndctl



display bad block information (1/2)

FUJITSU

```
# ndctl list -DRHMu
 {
   "regions":[
     {
       "dev":"region0",
       "size":"250.00 GiB (268.44 GB)",
              :
       "mappings":[
         {
           "dimm":"nmem1",
                  :
         },
         {
           "dimm":"nmem0",
                  :
         }
       ],
        :
```

(cont)

command to show region info with dimm, health and bad block info

(output is JSON format)

region 0 is interleaved region by nmem1 and nmem0

display bad block information (2/2)

FUJITSU

```
            :
         ],
        "badblock_count":1,
        "badblocks":[
          {
            "offset":65536,
            "length":1,
            "dimms":[
              "nmem0"
            ]
          }
        ],
         :
       :
#
```

badblock info in the region

(the previous ndctl showed only this information)

Current ndctl displays which DIMM has badblock in this region

In this example, nmem0 has broken block

To show NVDIMM device location (1/2)

FUJITSU

- What can shows the physical location of NVDIMM modules?
  - /dev/nmemX is named for NVDIMM module by NVDIMM driver
    - It is also used by ndctl
    - However, its id (X) is not related with device location
      - The driver decided it by the order which it found
  - I thought the handle of “NVDIMM Device” in ACPI DSDT may be good
    - Handle is created with the followings
      - node controller id, socket id, memory controller id, etc...
    - However, these id may be dynamic due to hardware/firmware configuration
      - Especially, our some servers can change # of nodes by dividing the box with system configuration
      - So, this may not be good to find physical location too

To show NVDIMM device location (2/2)

FUJITSU

- The remaining option is SMBIOS handle
  - You can confirm physical location of a device with “dmidecode” command
    - In addition, each device has SMBIOS handle, and shown in the command
  - SMBIOS handle is also “NVDIMM Region mapping structure” table in NFIT
  - So, if ndctl command can show it, you can find the location with the handle
  - I made a patch that ndctl show it as phys_id

Example of SMBIOS Handle of ndctl

FUJITSU

```
# ndctl list -Du
 [
  {
    "dev":"nmem1",
    "id":"XXXX-XX-XXXX-XXXXXXXX",
    "handle":"0x120",
    "phys_id":"0x1c"
  },
  {
    "dev":"nmem0",
    "id":"XXXX-XX-XXXX-XXXXXXXX",
    "handle":"0x20",
    "phys_id":"0x10",
    "flag_failed_flush":true,
    "flag_smart_event":true
  }
 ]
```

display DIMM infomation
with human readable format
(phys_id becomes Hexadecimal)

(The nmem0 included broken block
in previous result)

phys_id is SMBIOS handle of
these DIMM

0x10 is broken DIMM's handle
in this example

dmidecode can show the location of DIMM

FUJITSU

 # demidecode

:

Handle 0x0010, DMI type 17, 40 bytes

Memory Device

Array Handle: 0x0004

:

:

Locator: DIMM-Location-example-Slot-A

:

Type Detail: Non-Volatile Registered (Buffered)

SMBIOS handle of DIMM

Locator: shows the place of the DIMM

How to distinguish all namespaces on the DIMM

FUJITSU

- ndctl list command has many filtering feature
  - by region, namespace, bus, and dimm
  - Then list namespace with filtering dimm by ndctl, you can find all the namespaces

```
# ndctl list -N -d 0
 [
   {
     "dev":"namespace0.2",
     "mode":"sector",
     "size":67042312192,
          :
   },
   {
     "dev":"namespace0.0",
     "mode":"sector",
     "size":67042312192,
          :
   }
 ]
```

command to show
all namespaces on the nmem0

This dimm includes 2 namespaces
You need to back up them
before replacing nmem0

FUJITSU

NVDIMM Monitoring Daemon

- ndctl monitor

Who am I?

- QI Fuli
  - Software Engineer at Fujitsu Ltd.
  - PhD Student at the University of Tokyo
  - Working on R&D of PM
  - Email: qi.fuli@fujitsu.com

Why the status of NVDIMM must be monitored?

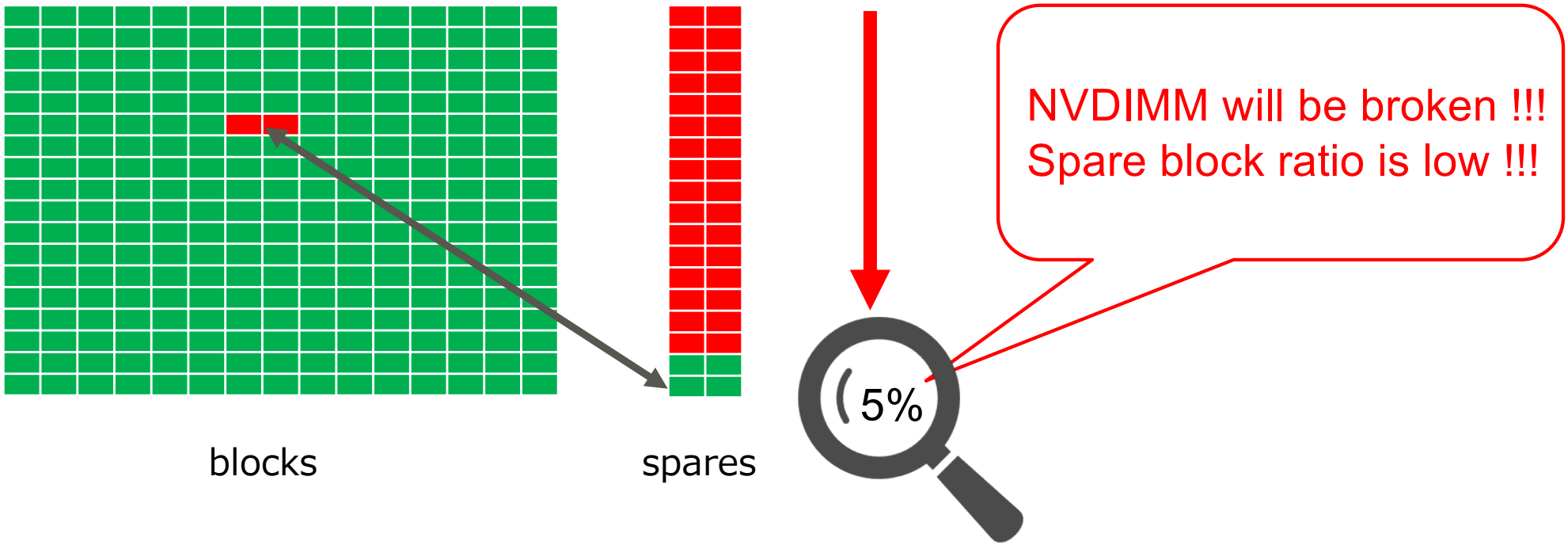
FUJITSU

- NVDIMM has a life span
  - Blocks of NVDIMM will be broken due to endurance problem
  - NVDIMM modules consume spare block
  - When spare decreases to 0, it cannot rescue broken block
  - NVDIMM has no feature, such as mirroring to save its data
- Backup and replacement are needed before no spares
- Users should know the best time to do the backup and replacement

Status of NVDIMM needs to be monitored and be notified before NVIMM becomes broken

Ndctl monitor

FUJITSU



- I created a daemon to monitor NVDIMM
  - monitor health status
  - notify critical status

https://pmem.io/ndctl/ndctl-monitor.html

Features of ndctl monitor (1/3)

FUJITSU

- SMART events come from NVDIMM can be monitored

SMART event	trigger	effect
spares-remaining	spare block value goes below the spare threshold limit	NVDIMM becomes broken; data will be lost; etc.
health-status	normal health status of NVDIMM changes	
unclean-shutdown	last shutdown to be recorded at the next boot	saving data target failed on NVDIMM; etc.
media-temperature	temperature value of nvdimm goes above threshold limit	server is getting too hot; may need remediation; a specific fan fails; etc.,
controller-temperature	controller temperature value goes above threshold limit	

Features of ndctl monitor (2/3)

FUJITSU

- Actions after monitoring
  - Log the output notification
    - Log destination
      - syslog
      - arbitrary file（set in the configuration file or set by command option）
    - Output format
      - JSON (can be analyzed by other log collectors, such as fluentd）
  - Kick other application（to be implemented）
    - When the monitor detects a SMART health event, the application will move the data in real time.

Features of ndctl monitor (3/3)

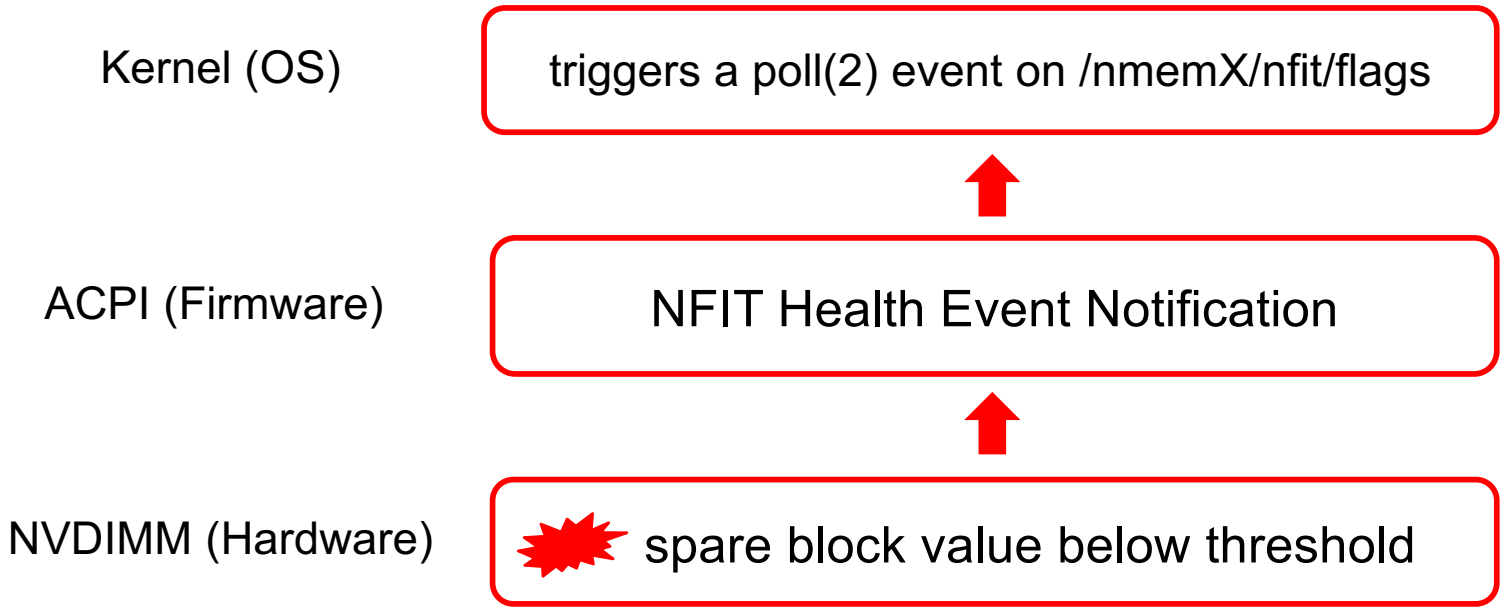
FUJITSU

- Filter of ndctl monitor
  - objects to monitor can be filtered
    - all NVDIMMs are monitored by default
    - objects can be filtered by namespace, region, bus, and dimm
      - backup the data of applications which are running on a certain namespace
        ⇒ filter by namespace
      - allowing to monitor for any media error on the bus
        ⇒ filter by bus
  - SMART health events to monitor can be filtered

ACPI supports for NVDIMM

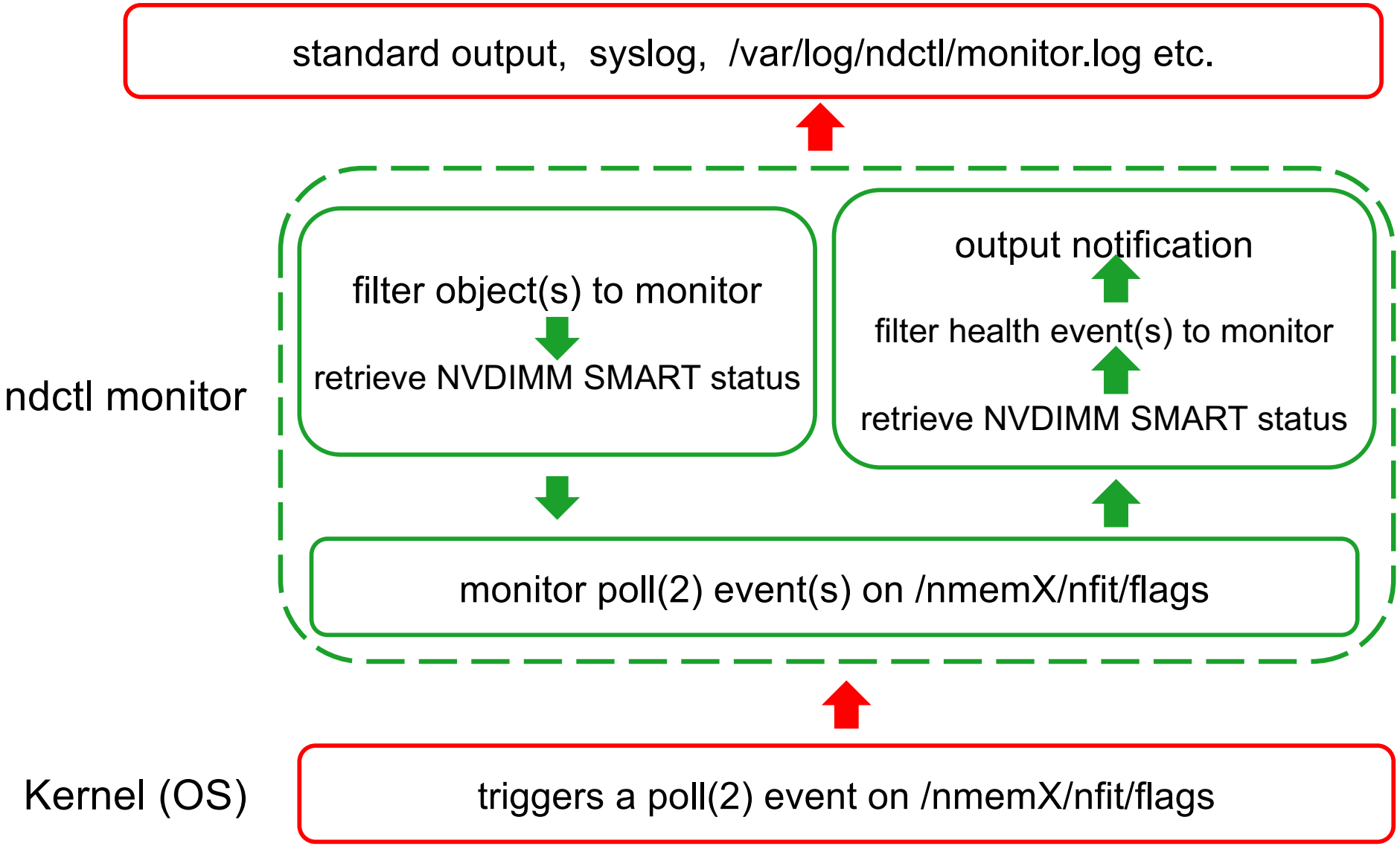
FUJITSU

- NVDIMM has SMART Health Info like SSD/HDD
  - SMART Health Info can be retrieved via a *_DSM* method
  - SMART Health Info includes spare block value and spare threshold limit
- ACPI 6.1 adds “NFIT Health Event Notification”
  - a notification will be sent when the spare block value goes below the spare threshold limit
  - a poll(2) event will be triggered when the notification is received



Architecture of ndctl monitor

FUJITSU



Sample of output notification

FUJITSU


```
"timestamp":"1537323709.432459532",
  "pid":28189,
  "event":{ "dimm-spares-remaining":true },
  "dimm":{
    "dev":"nmem1", "health":{
      "health_state":"non-critical",
      "temperature_celsius":23,
      "controller_temperature_celsius":25,
      "spares_percentage":4,
      "temperature_threshold":40,
      "controller_temperature_threshold":30,
                    …
      "spares_threshold":5,
      "shutdown_state":"clean“
   }
  }
```


SMART Health event

current spare block value

spare threshold limit

How to start ndctl monitor

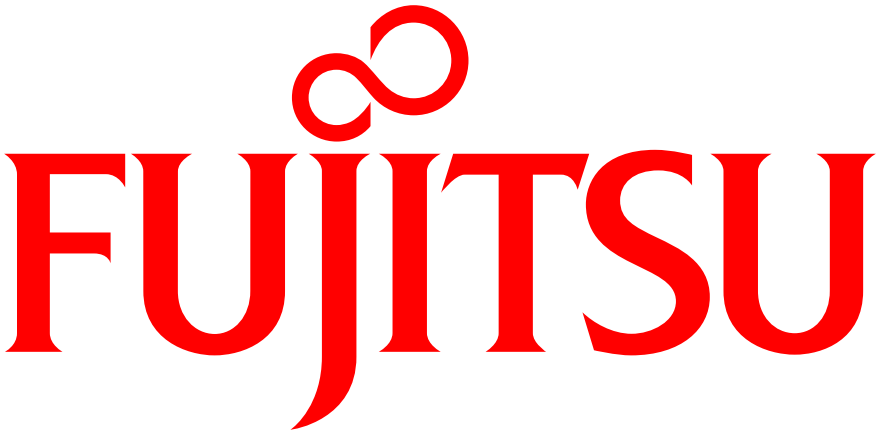
FUJITSU

- Linux distribution support systemd
  - Systemd service
    - ⇒ systemctl start ndctl-monitor.service
- Linux distribution does not support systemd
  - ndctl monitor --daemon
- Users could run monitor as a one shot command
  - ndctl monitor

Summary and Future work

FUJITSU

- Summary
  - Basis of NVDIMM
  - Replacement
    - Replacement of NVDIMM is complex
    - Adding some information in ndctl list for replacing NVDIMM
  - Monitoring
    - Monitoring SMART status of NVDIMM is important
    - Ndctl monitor daemon
- Future work
  - Kicks other applications by daemon
  - plural daemon execution via systemctl



shaping tomorrow with you

Emulation of NVDIMM

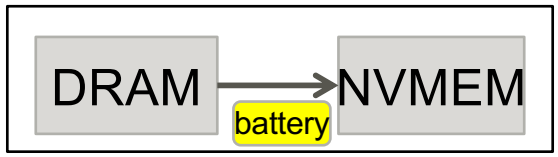
FUJITSU

- You can try NVDIMM interface easily
  - custom e820
    - To try interface for filesystem or DAX
    - it can used with kernel boot option (memmap=XXG!YYG)
      - XX Gbyte RAM is used for NVDIMM interfaces
  - nfit_test.ko
    - Try to test management command (ndctl)
    - It works instead of the firmware for NVDIMM

Types of NVDIMM

FUJITSU

- JEDEC defines some types of DIMM module

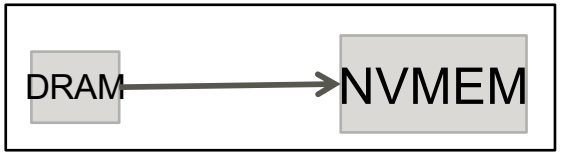


NVDIMM-N

- DIMM module includes DRAM and NVMEM
- Amount of RAM = Amount of NVMEM
- The module includes battely
- Usually, RAM is used
- Backup to NVMEM at Power down

NVDIMM-F

- Only NVMEM is used
- access like storage



NVDIMM-P

- DIMM module includes DRAM and NVMEM
- Amount of RAM is less than NVMEM
- RAM is used as cache