

Project Q&A

Prof. Dongkun Shin (dongkun@skku.edu)

TA – Junho Lee (crow6316@skku.edu)

TA – Somm Kim (sommkim@skku.edu)

Embedded Software Laboratory

Sungkyunkwan University

<http://nyx.skku.ac.kr>

Power-Off Recovery

Normal Power-Off

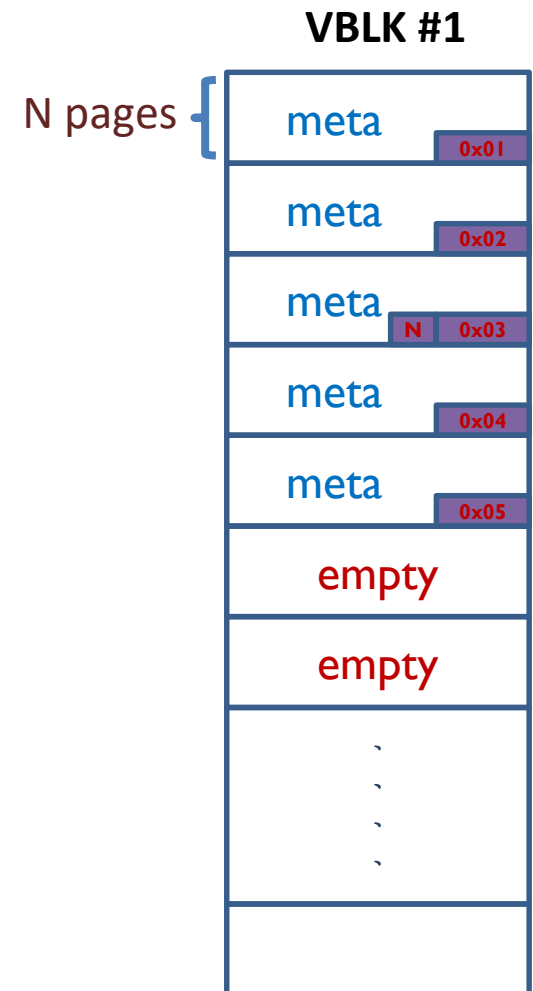
- Store some metadata to NAND flash during NPO (ftl_flush)
 - NOT all of metadata should be stored!
 - Some might be recovered with other metadata
- Recover metadata at boot time (ftl_open)
 - Read metadata from NAND flash
 - Recover other metadata with them
- Do not touch VBLK #0 (special purpose for bad-blk)
- Hint : GreedyFTL in OpenSSD project

Sudden Power-Off

- Your FTL should handle sudden power loss
 - Once sending completion acknowledgment to the host, the write request should be **persistent** after sudden power loss
 - Remind that SPO can be happened anytime, anywhere in your code
 - (That's why it's called 'Sudden' power-off)
 - When `ftl_write`, `nand_program`, `gc` (merge), ...
 - Even when `ftl_flush`, `ftl_open` and `recovering(!)`
- You should consider following things
 - Before SPO happens
 - `NAND_PROGRAM` order (who comes first? data or metadata)
 - What to store to flash memory (which metadata?)
 - After SPO happens (recovery)
 - How to recover metadata (SRAM, DRAM)
 - Are up-to-date data available?

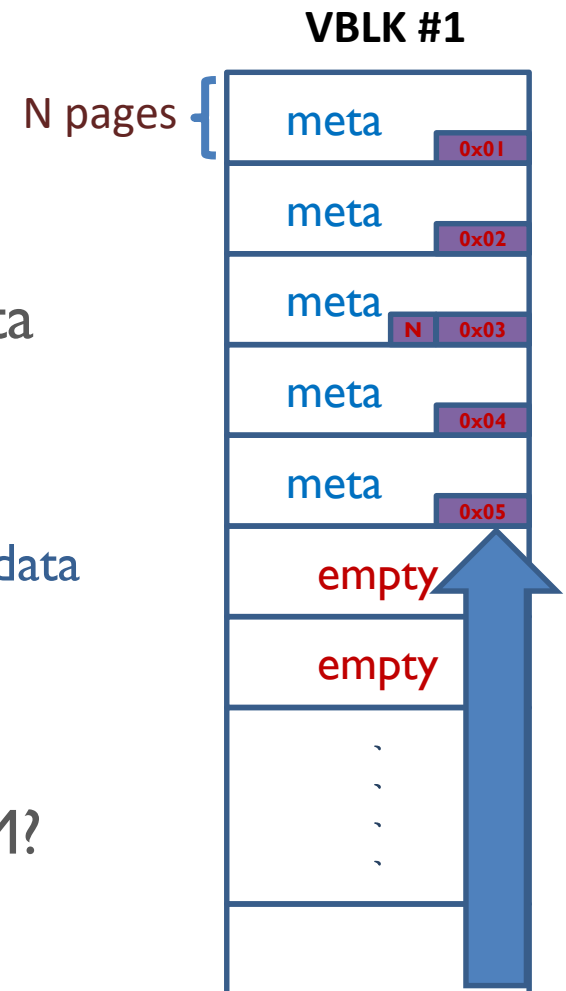
Guideline in SPO

- Consider all cases that sudden power off may happens
- For every `ftl_write`, do following things
 - Data write
 - Metadata write to special VBLK
 - Mark written metadata as ‘up-to-date’
 - Like a special marker



Guideline in SPO (cont'd)

- Figure out whether NPO or SPO
 - Read VBLK #I from tail to head
 - Until finding valid mark
 - If NPO, there's no headache; just read meta
 - If SPO, do recovery
 - Read written metadata at up-to-date zone
 - With a recovered metadata, recover other metadata
- Do not trust `cur_write_page` at SPO
 - What if power loss at `NAND_PROGRAM`?
 - You will need VBLK #2 for metadata
 - **What if power loss during recovery?**



Debugging Tips

- NPO

- Modify `ftl_test.c` to flush and load metadata
- Real tests will be done with real file system and real files

- SPO

- Make `_fault_injection` in your codes
 - Simple `_fault_injection` function code

```
UINT32 _fault_injection (void)
{
    uart_printf("Fault_injection: Power-Off!");
    uart_printf("Shutdown Jasmine board right now");
    while(1);
}
```

Project Grading Policy

Project Grading Policy

- Project I (20%)
 - Extra points (~11/6): 10%
 - No delay
- Project II (50%)
 - Extra points (~11/27): 20%
- Project III (30%)

- Subject to change

Project III Grading Policy

- DFTL improvement (60)
 - Design
 - ex) New idea proposal, ...
 - Implementation
 - ex) Difficulty, Completeness, ...
 - Evaluation
 - ex) Use appropriate and diverse workloads, Comparison with existing techniques, ...
- Additional function (30)
- Report & Presentation (10)
- You can also submit two firmwares.