ICE3028: Embedded Systems Design - Project 2: Page Mapping FTL on NAND simulator

Prof. Jinkyu Jeong (Jinkyu@skku.edu)
TA -- Minwoo Ahn (minwoo.ahn@csl.skku.edu)
TA -- Donghyun Kim (donghyun.kim@csl.skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu

Descriptions

- Develop a page mapping FTL simulator
  - Simulate the operations of page mapping FTL
    - Manage mapping with L2P table
  - Assumption
    - Uniform size of write request
    - Initial state of flash blocks: empty
  - Misc.
    - GC policy: greedy policy
    - GC triggering condition: when # of free blocks becomes one
    - Make Over-Provisioning (OP) ratio adjustable
  - Draw a graph on OP ratio vs. WAF

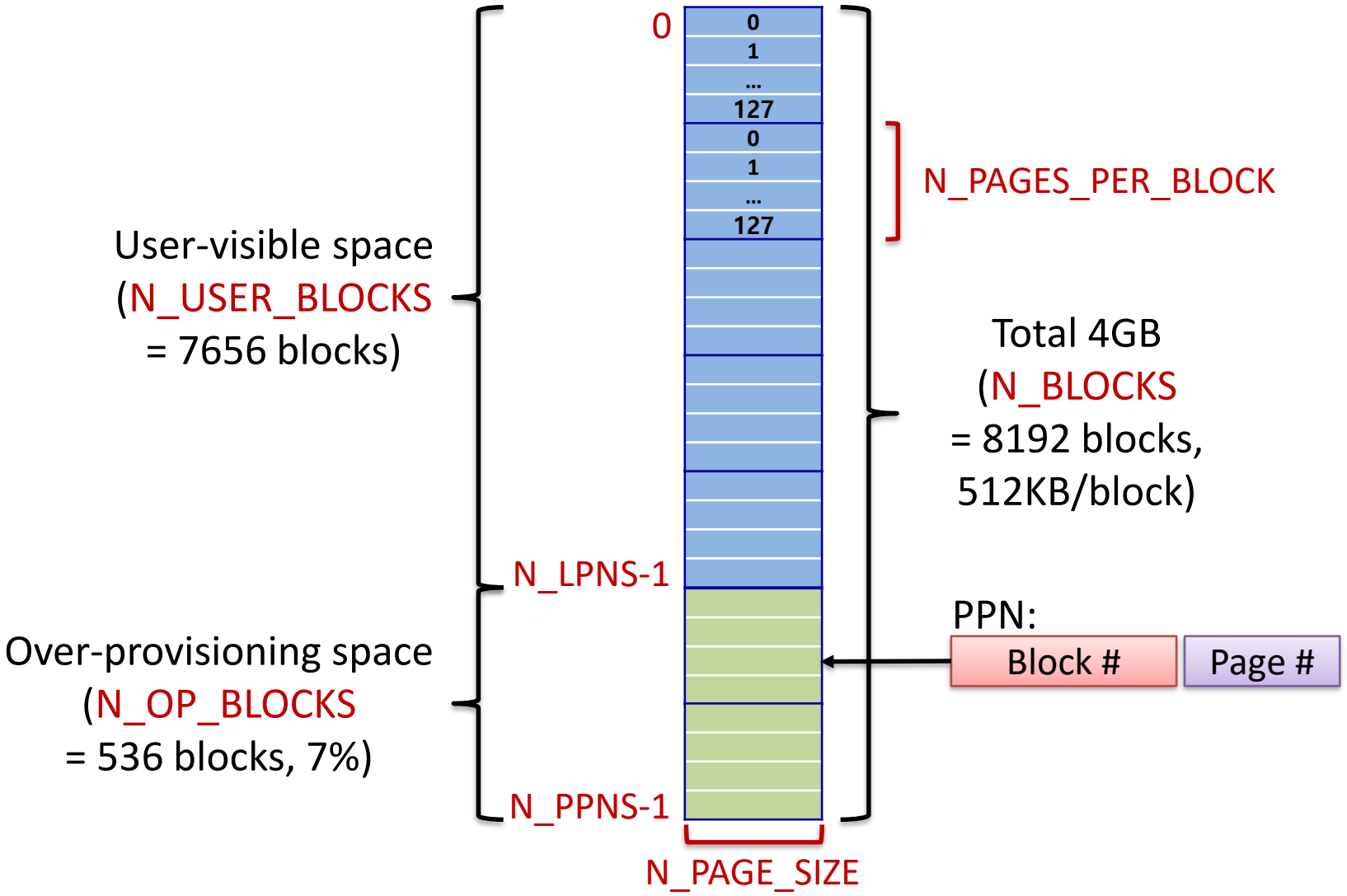
Data Structures

- L2P table
  - Index: logical page number (LPN)
  - Value: physical page number (PPN)
  - Updated on write and GC
- Store the LPN into the spare area of written page
- Per-block information
  - Which page is valid? (optional)
  - # of valid pages (optional)
- How to find the victim block for GC?

Configurations

- Flash memory
  - Total capacity: 4GB
  - Page size: 4KB
  - Pages per block: 128 (block size: 512KB)
- SSD
  - OP ratio: 7% (default), 10%,13%,16%,19%,22%,25%,28%
- Workload
  - Size of each write request: 4
  - Total number of write requests: x10 of the total SSD capacity visible to the user

Configurations (cont'd)



Code Analysis

```
#define   SSD_SHIFT       32
#define   PAGE_SHIFT      12
#define   PAGES_PER_BLOCK_SHIFT   7
#define   OP_RATIO        7
#define   N_RUNS          10

#define   PPN_SHIFT       (SSD_SHIFT - PAGE_SHIFT)
#define   BLOCKS_SHIFT        (PPN_SHIFT - PAGES_PER_BLOCK_SHIFT)
#define   N_PAGE_SIZE         (1 << PAGE_SHIFT)
#define   N_PAGES_PER_BLOCK   (1 << PAGES_PER_BLOCK_SHIFT)
#define   N_PPNS          (1 << PPN_SHIFT)
#define   N_BLOCKS        (1 << BLOCKS_SHIFT)
#define   N_USER_BLOCKS       (N_BLOCKS * 100 / (100 + OP_RATIO))
#define   N_OP_BLOCKS         (N_BLOCKS - N_USER_BLOCKS)
#define   N_LPNS          (N_USER_BLOCKS * N_PAGES_PER_BLOCK)
#define   LPN_COUNTS      (N_LPNS * N_RUNS)
```

Code Analysis (cont'd)

Execution result (Skelaton code)

```
SSD capacity:           4GB
Page size:              4KB
Pages / Block:          128 pages
Block size:             512KB
OP ratio:               7%
Physical Blocks:        8K (8192)
User Blocks:            7656
OP Blocks:              536
PPNs:                   1M (1048576)
LPNs:                   979968
Total runs:             x10
Actual capacity:        4013948928 Bytes
```

show_info()

```
[Run 1] host 979968, valid page copy 0, GC# 0, WAF=1.00
[Run 2] host 1959936, valid page copy 0, GC# 0, WAF=1.00
[Run 3] host 2939904, valid page copy 0, GC# 0, WAF=1.00
[Run 4] host 3919872, valid page copy 0, GC# 0, WAF=1.00
[Run 5] host 4899840, valid page copy 0, GC# 0, WAF=1.00
[Run 6] host 5879808, valid page copy 0, GC# 0, WAF=1.00
[Run 7] host 6859776, valid page copy 0, GC# 0, WAF=1.00
[Run 8] host 7839744, valid page copy 0, GC# 0, WAF=1.00
[Run 9] host 8819712, valid page copy 0, GC# 0, WAF=1.00
[Run 10] host 9799680, valid page copy 0, GC# 0, WAF=1.00
```

sim()

```
Results ------
Host writes:            9799680
GC writes:              0
Number of GCs:          0
Valid pages per GC:     -nan pages
WAF:                    1.00
```

show_stat()

Code Analysis (cont'd)

Execution result, OP ratio=28% (Answer code)

```
NAND: 8192 blocks, 128 pages per block, 1048576 pages
SSD capacity:           4GB
Page size:              4KB
Pages / Block:          128 pages
Block size:             512KB
OP ratio:               28%
Physical Blocks:        8K (8192)
User Blocks:            6400
OP Blocks:              1792
PPNs:                   1M (1048576)
LPNs:                   819200
Total runs:             x10
Actual capacity:        3355443200 Bytes

[Run 1] host 819200, valid page copy 0, GC# 0, WAF=1.00
[Run 2] host 1638400, valid page copy 284001, GC# 6828, WAF=1.17
[Run 3] host 2457600, valid page copy 1011555, GC# 18912, WAF=1.41
[Run 4] host 3276800, valid page copy 1994710, GC# 32993, WAF=1.61
[Run 5] host 4096000, valid page copy 3096437, GC# 48000, WAF=1.76
[Run 6] host 4915200, valid page copy 4247160, GC# 63390, WAF=1.86
[Run 7] host 5734400, valid page copy 5418483, GC# 78941, WAF=1.94
[Run 8] host 6553600, valid page copy 6596464, GC# 94544, WAF=2.01
[Run 9] host 7372800, valid page copy 7777775, GC# 110173, WAF=2.05
[Run 10] host 8192000, valid page copy 8958804, GC# 125800, WAF=2.09

Results ------
Host writes:            8192000
GC writes:              8958804
Number of GCs:          125800
Valid pages per GC:     71.21 pages
WAF:                    2.09
```

pm.c

- ftl_read(long lpn, u32 *read_buffer)
- ftl_write(long lpn, u32 *write_buffer)
- garbage_collection()

- Write above APIs based on NAND simulator written at Project 1

Submission

- Compress your folder as YourStudentID-2.tar.gz
  - With codes and graph
  - Send email to minwoo.ahn@csl.skku.edu
  - Please send mail with uniformized title
    - [ICE3028]YourStudentID-2

- PLEASE DO NOT COPY
  - YOU WILL GET *F* GRADE IF YOU COPIED

- Due date: 11/11(Sun.), 23:59:59 PM
  - -20% per day for delayed submission

Questions

- If you have questions, please email to TAs

- You can also visit Semiconductor Building #400509
  - Please send email before visiting