

# ICE3028: Embedded System Design

## - Project 3: Other Page Mapping FTLs on NAND Simulator

Prof. Jinkyu Jeong ([jinkyu@skku.edu](mailto:jinkyu@skku.edu))

TA -- Minwoo Ahn ([minwoo.ahn@csl.skku.edu](mailto:minwoo.ahn@csl.skku.edu))

TA -- Donghyun Kim ([donghyun.kim@csl.skku.edu](mailto:donghyun.kim@csl.skku.edu))

Computer Systems Laboratory

Sungkyunkwan University

<http://csl.skku.edu>

# Why Other Policies / FTLs?

- We discussed it in Lab 6
  - <http://csl.skku.edu/uploads/ICE3028F18/lab5-multistream.pdf>
- If workload is fully random, greedy is best-effort
  - In real world, there exists ‘locality’
  - Locality-aware GC policy
    - Cost-Benefit policy (1)
  - Host gives hint to FTL
    - Multi-Stream write SSD (2)

# Descriptions

- Develop two different FTLs based on project #2
  - There are not that lots of differences among FTLs
    1. Page-Mapping FTL with cost-benefit policy
    2. Multi-Stream Write SSD
- Skeleton code is in course homepage
  - Can freely add other files for your own use
    - Make sure that executable 'pm\_sim' file comes out with 'make all' command
  - Do not touch pm.h, pm\_sim.h, nand.c and nand.h
    - Your score will be zero
    - If you want to fix it, contact me first

# Descriptions (cont'd)

- Submit your tar.gz file and pdf file to e-mail
  - tar.gz : all the source codes (using 'make tar')
  - pdf : graph file that describes relation among FTL policies, workloads and WAF
    - FTL policy : Greedy FTL, Cost-benefit FTL, Multi-stream SSD
    - Workload : Fully random (as project #2), Hot-cold, Multi Hot-cold
  - e-mail : [minwoo.ahn@csl.skku.edu](mailto:minwoo.ahn@csl.skku.edu)
- Due : 11/22, 23:59:59

# Skeleton Code

- Makefile
  - 3 options in WORKLOAD
    - -DMULTI\_HOT\_COLD (HOT:WARM:COLD = 6:3:1)
    - -DHOT\_COLD (HOT:COLD = 8:2)
    - if there is nothing (fully random like project #2)
  - 3 options in FTL
    - -DMULTI\_STREAM (Multi-Stream SSD mode)
    - -DCOST\_BENEFIT (Cost-benefit GC policy)
    - if there is nothing (Greedy GC policy like project #2)

```
12 WORKLOAD = -DMULTI_HOT_COLD
13           #-DHOT_COLD
14           # if -DHOT_COLD is not set, WORKLOAD will be random as a Project #2
15
16 FTL      = #-DMULTI_STREAM
17           #-DCOST_BENEFIT
18
19           # if both -DMULTI_STREAM and -DCOST_BENEFIT is not set, FTL should be a greedy FTL
```

# Skeleton Code (cont'd)

- pm.c
  - The number of argument in ftl\_write differs among options
    - If MULTI\_STREAM flag is set by Makefile, streamid comes in argument
    - Else, there is no argument named streamid

```
85 #ifdef MULTI_STREAM
86 void ftl_write (int streamid, long lpn, u32 *data);
87 #else
88 void ftl_write (long lpn, u32 *data);
89 #endif
```

- Once you change WORKLOAD and FLAG in Makefile, issue 'make clean' before 'make all'

# Cost-Benefit GC Policy

- Most of codes are similar to Greedy FTL except for selecting victim block in GC
  - Greedy FTL : selects a block with the least number of valid page
  - Cost-Benefit FTL : select a block with the maximum

$$\frac{Benefit}{Cost} = \frac{(1 - u)}{2u} \times age$$

- How to get age?
  - Each physical block has a last modification time
  - Logical time = the total number of writes
    - = # of host\_write + # of gc\_write

# Cost-Benefit GC Policy (cont'd)

- You'll need 'now' function to get logical time
  - Should fix modification time of the current writing block
  - It's already made in skeleton code

```
19 #ifdef COST_BENEFIT
20 static long now(void) {
21     return s.host_write + s.gc_write;
22 }
23 #endif
```



# Multi-Stream Write SSD

- At `ftl_write`, there is one more argument: `streamid`
  - Each write request with different `streamid` has to go to different physical block (MUST NOT mixed `streamid` in single block)
  - GC triggered condition is nearly same as Greedy FTL
    - If there is only one free block left

# Graph

- WAF depends on
  - Workload
    - Fully random (No flag at WORKLOAD field in Makefile)
    - Hot-cold (-DHOT\_COLD at WORKLOAD field in Makefile)
    - Hot-warm-cold (-DMULTI\_HOT\_COLD at WORKLOAD field in Makefile)
  - FTL policy
    - Greedy FTL (No flag at FTL field in Makefile)
    - Cost-Benefit FTL (-DCOST\_BENEFIT at FTL field in Makefile)
    - Multi-Stream SSD (-DMULTI\_STREAM at FTL field in Makefile)
- You should understand what the given result means