A Simple Approach to Find the Address Mapping Scheme of USB Flash Drives

Jaehyuck Cha
(jaehcha@hanyang.ac.kr)

Hanyang University, Korea

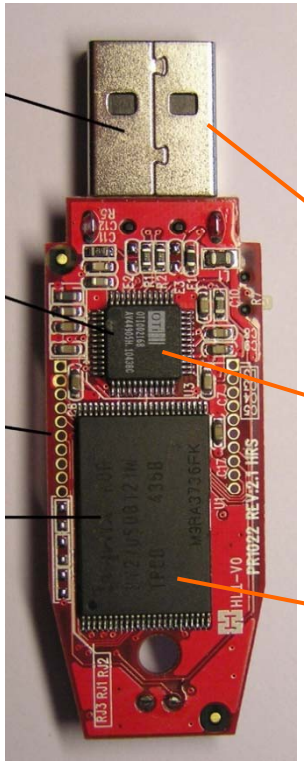
Hanyang University
Database Laboratory

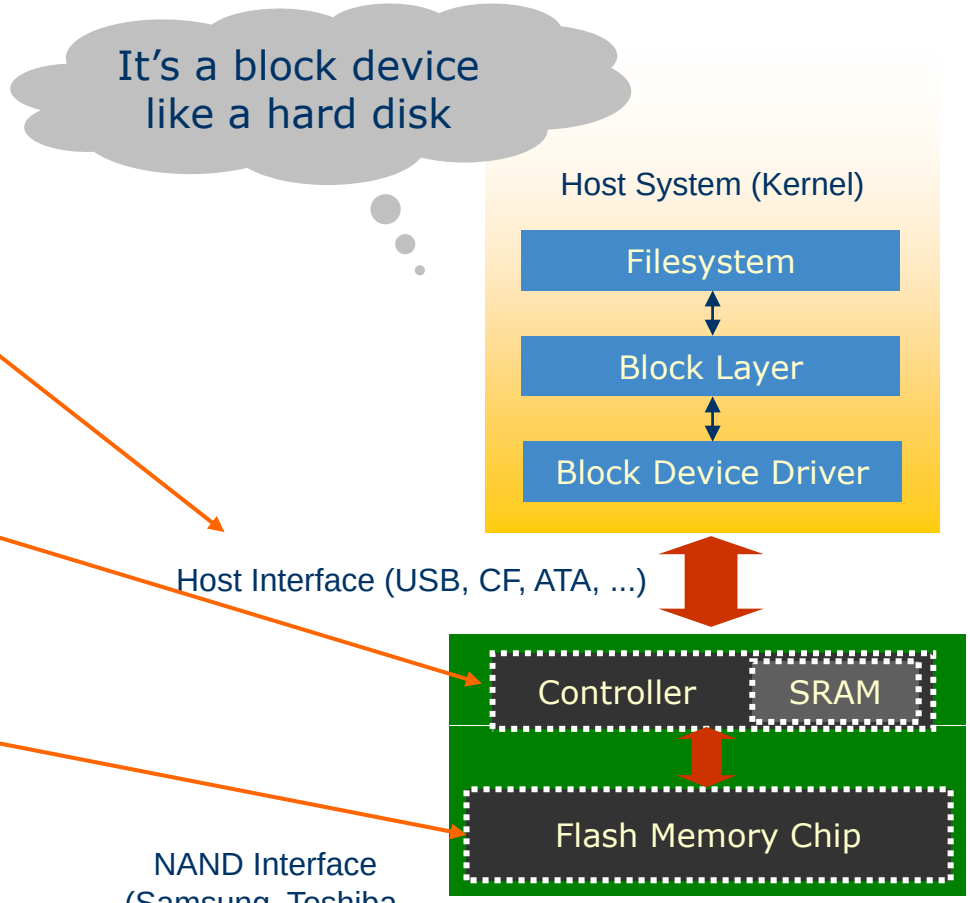
Outline

- USB Flash Drives
- Mapping Algorithms of FTL
- Indentifying Mapping Algorithms
- Experimental Tests on Real Devices
- Conclusion & Further Issues

USB Flash Drives

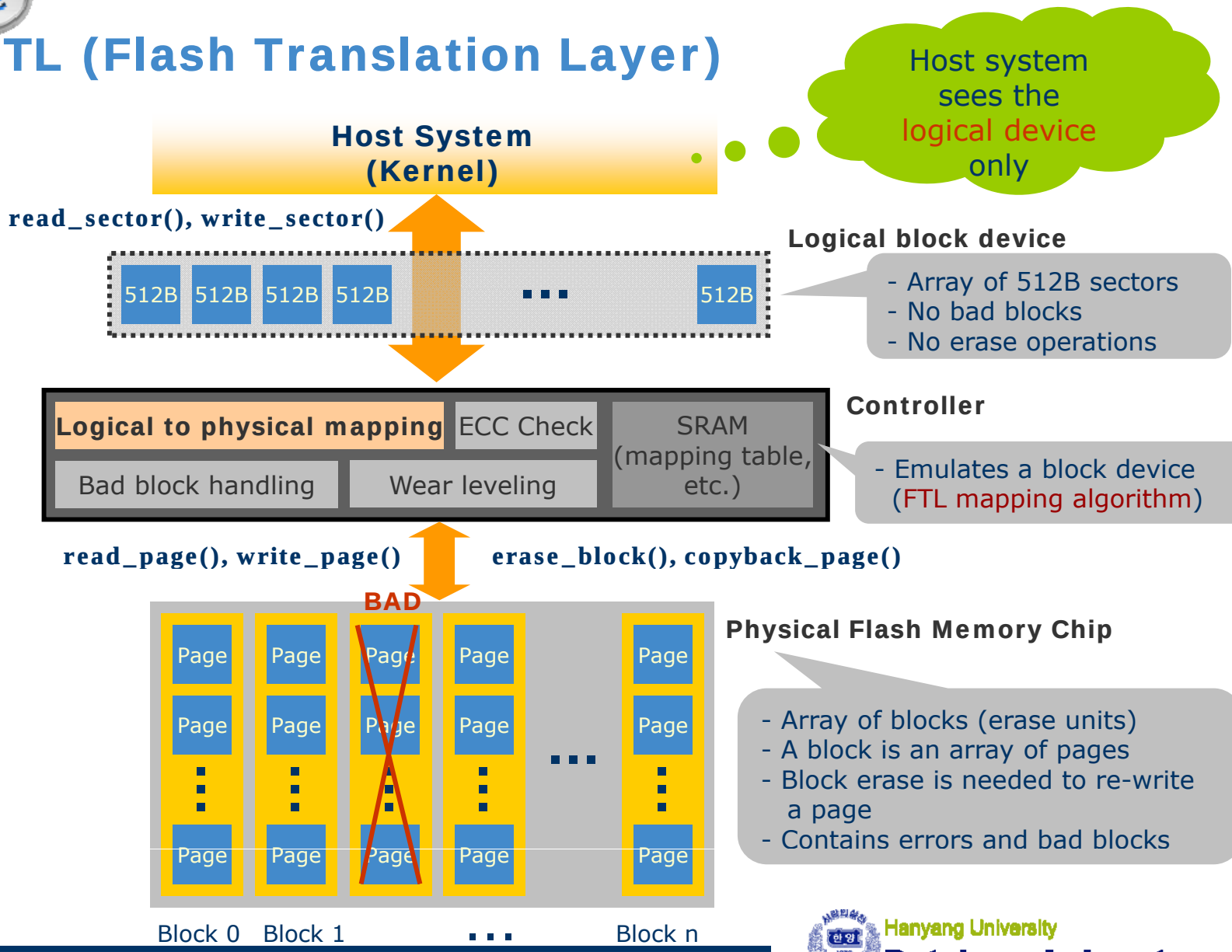
❖ Internals of a USB flash drive





USB Flash Drives (cont'd)

❖ FTL (Flash Translation Layer)



Mapping Schemes of FTL

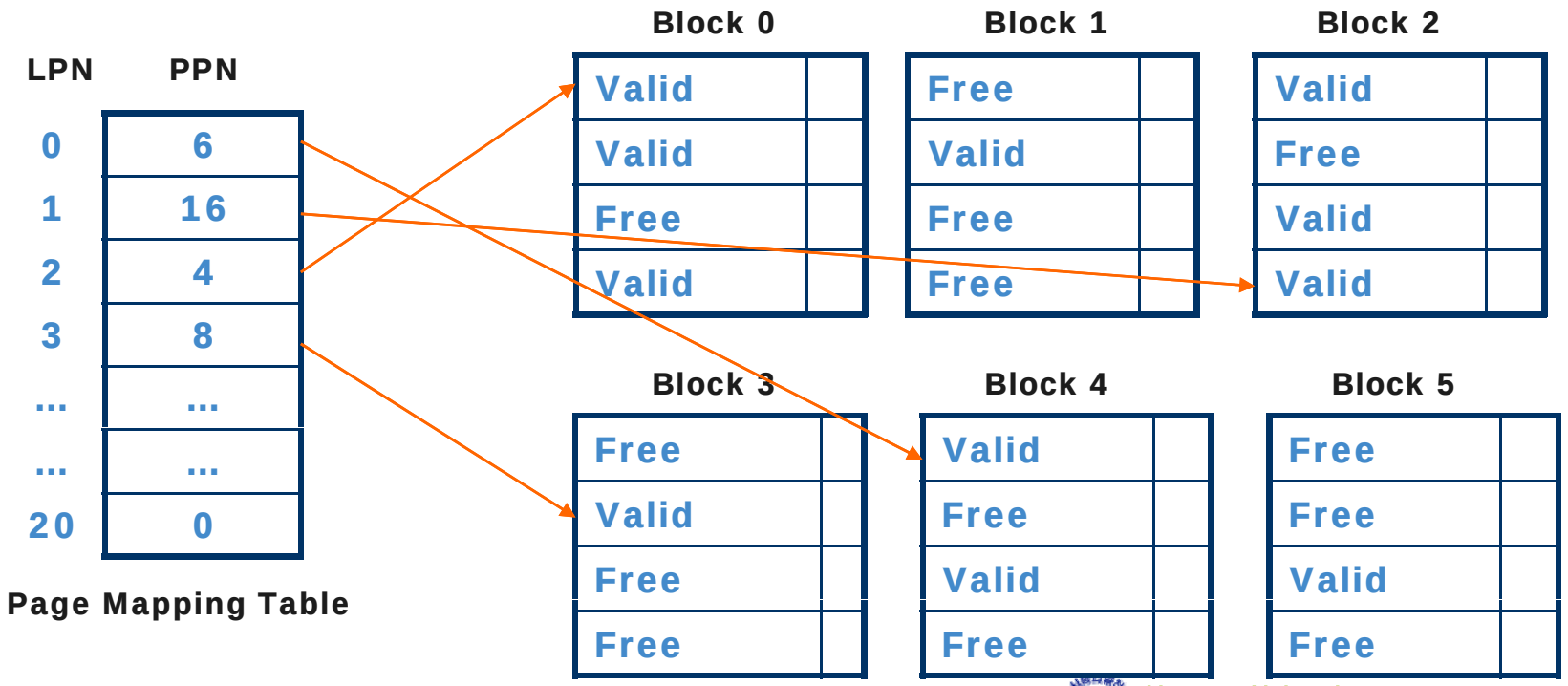
❖ Mapping schemes

- Page-level mapping scheme
- Block-level mapping scheme
- Hybrid mapping scheme

Mapping Schemes (cont'd)

❖ Page-level mapping scheme

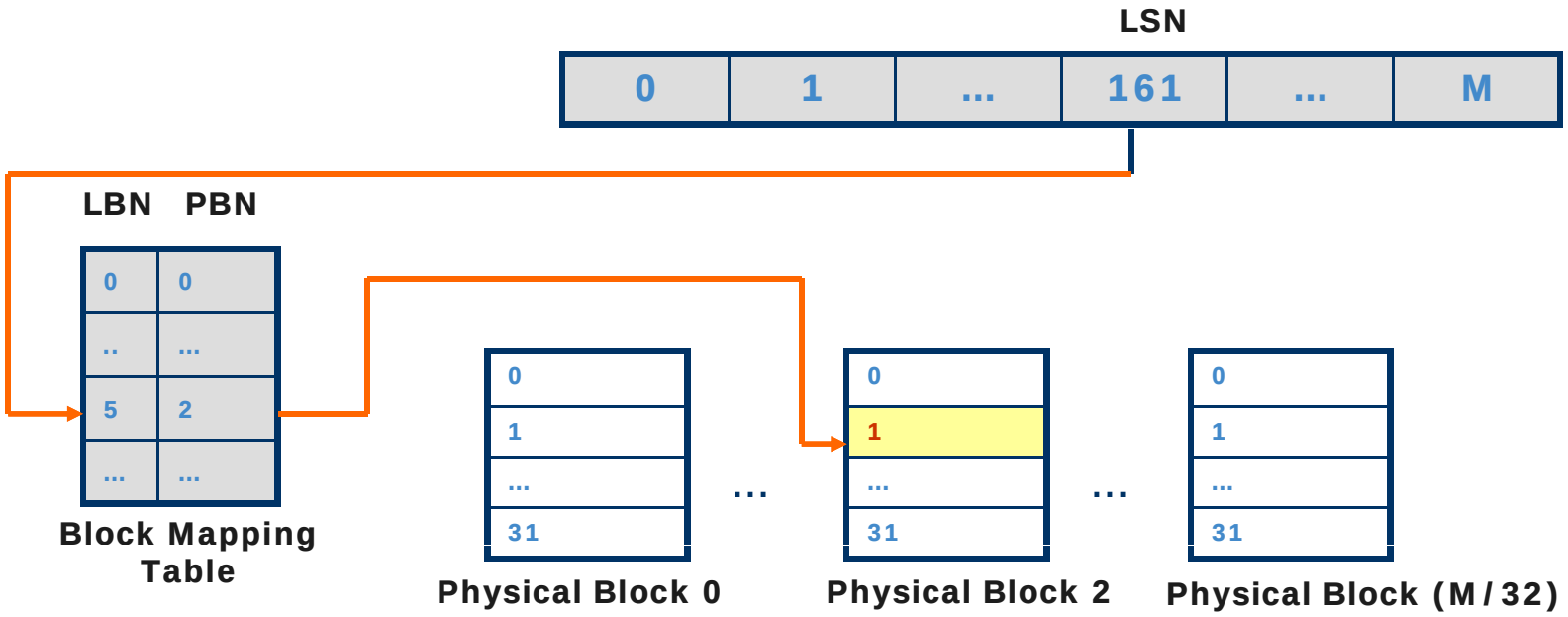
- Both LBN and offset are mapped to different PBN and offset
- Logical pages in the same block can be distributed to different physical blocks



Mapping Schemes (cont'd)

❖ Block-level mapping scheme

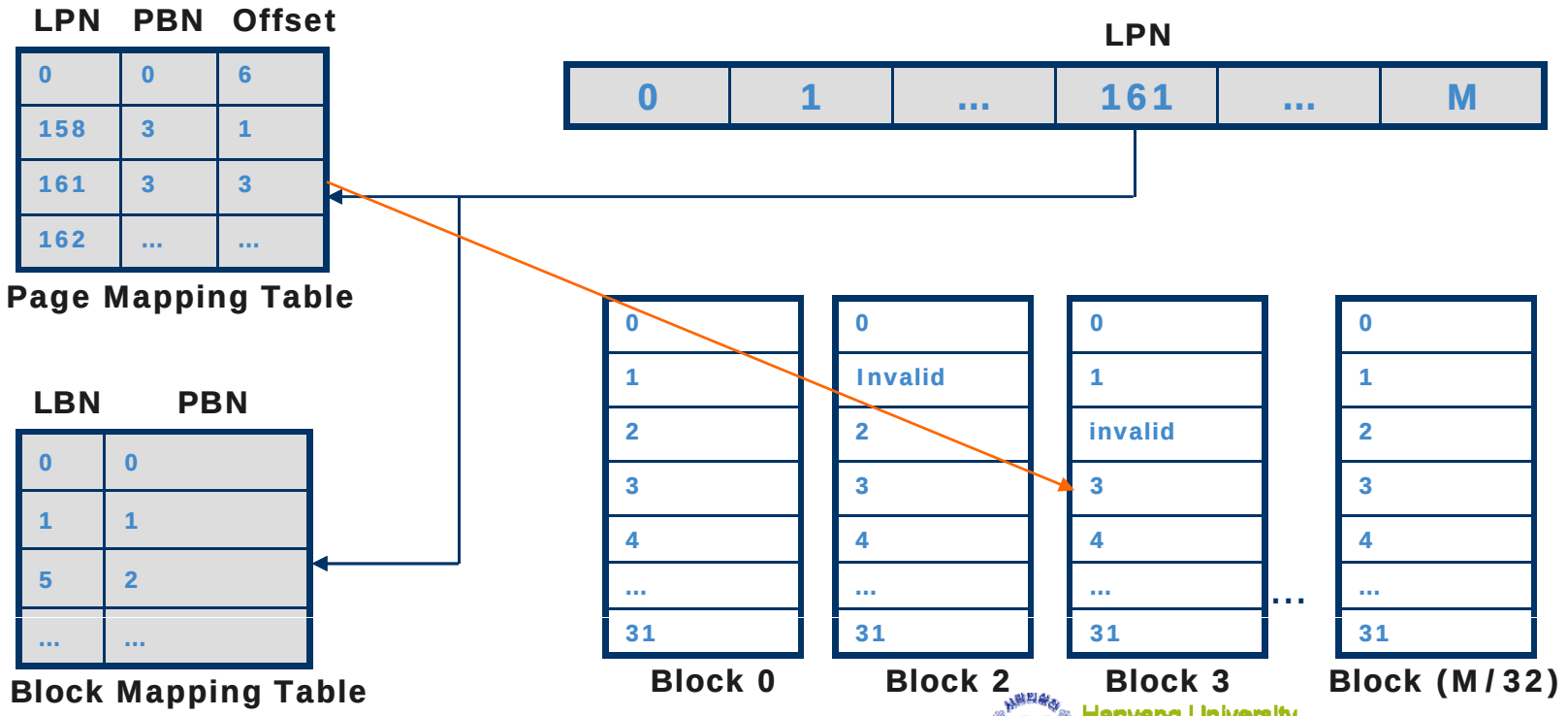
- Only LBN is translated to PBN
- Offset never changed



Mapping Schemes (cont'd)

❖ Hybrid mapping scheme

- Most of blocks called "data block" are allocated by block mapping algorithm
- A few blocks called "log block" are allocated by page mapping algorithm



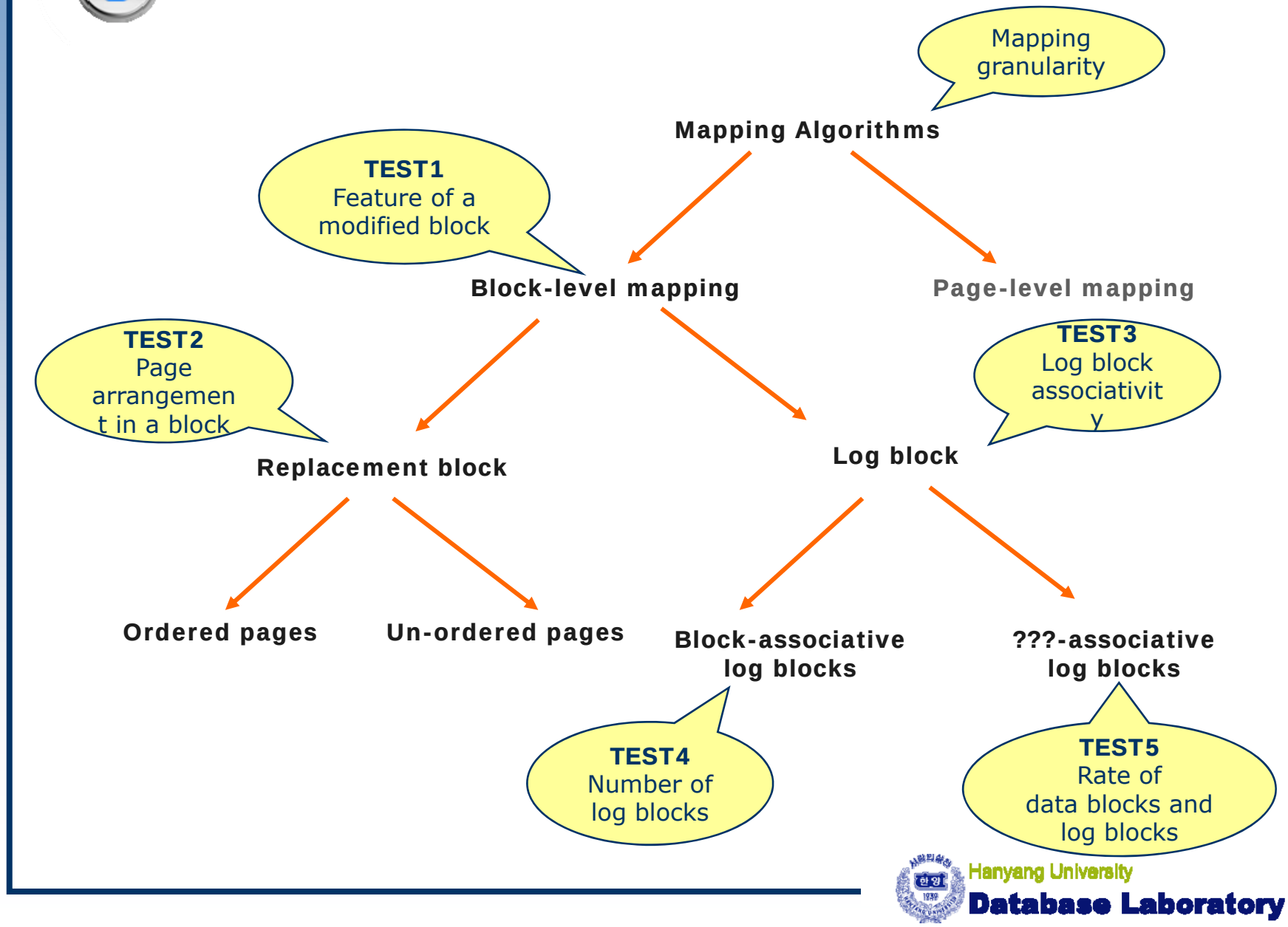
Target Devices

❖ Target devices

- USB flash drives that have large(2k)-page NAND flash memories
  - A block has 64 pages
  - The size of a page is 2KB
- Limitation
  - Pages in a block should be written sequentially

Identification of the Mapping Scheme

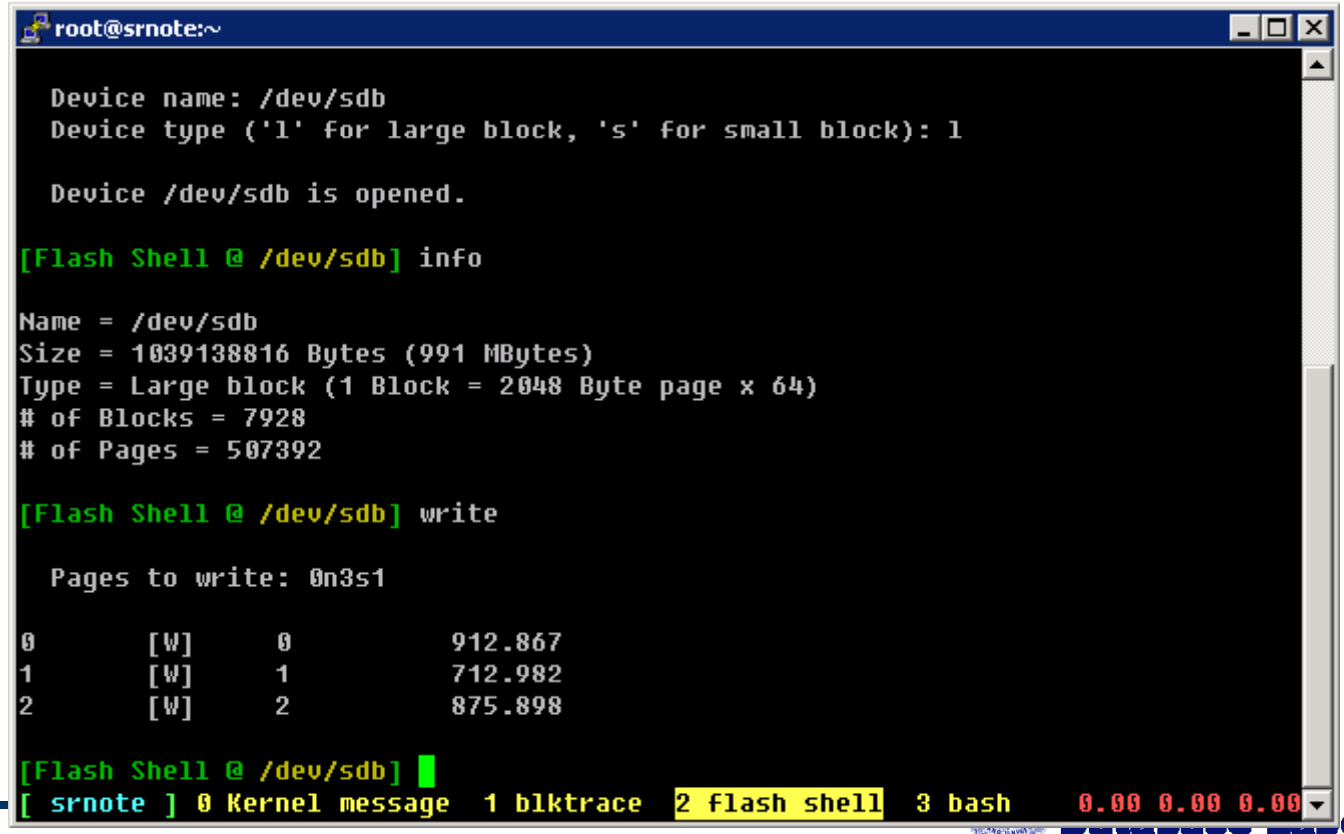
Classification of Major Mapping Schemes



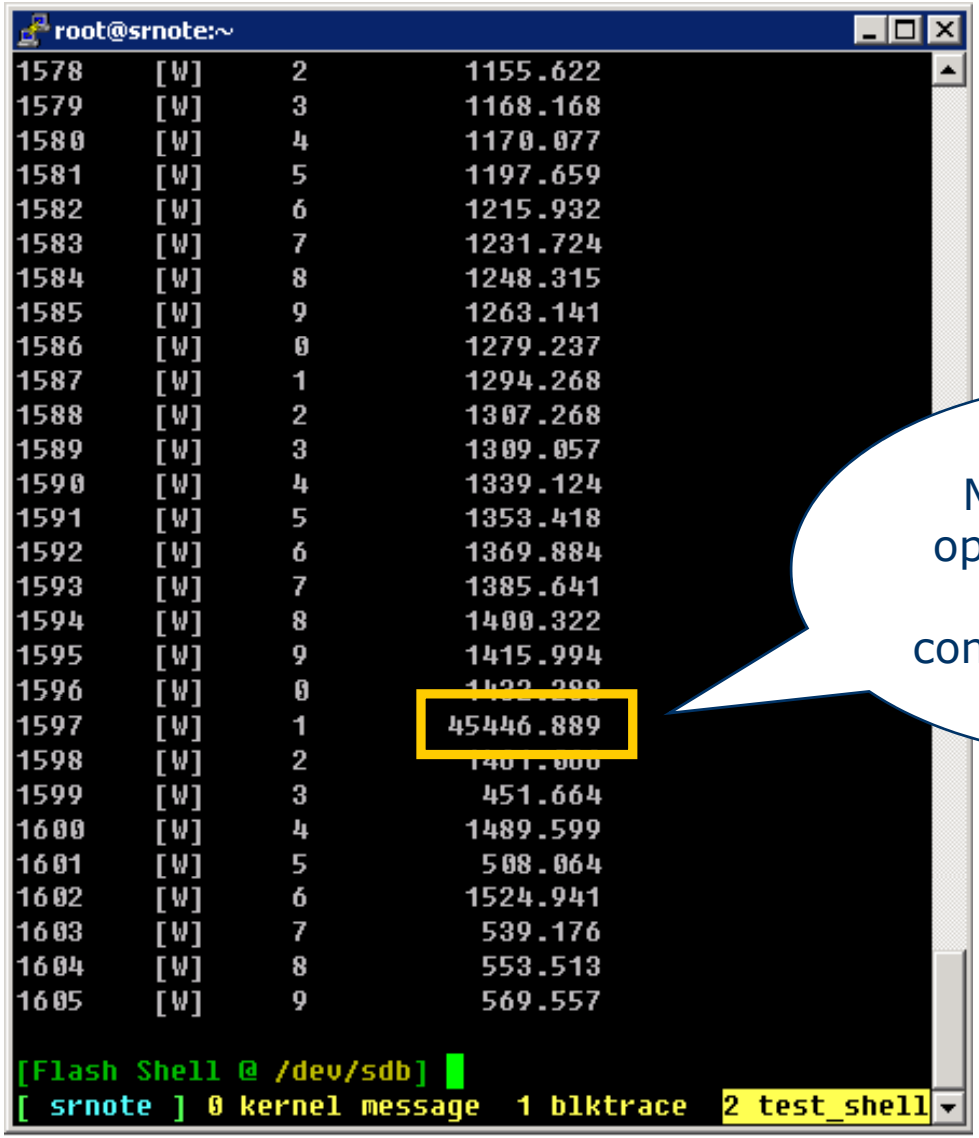
Timing Measurements

Timing measurements

- Open a character device file with O_DIRECT
- Send requests to the device with page-based I/Os
- Measure timing gap between device *issue* and *complete*
  - Block IO trace facility of linux (*blktrace*)
  - Time gap is calculated in the unit of usec



Identification of the Mapping Schemes
Timing Measurements (cont'd)

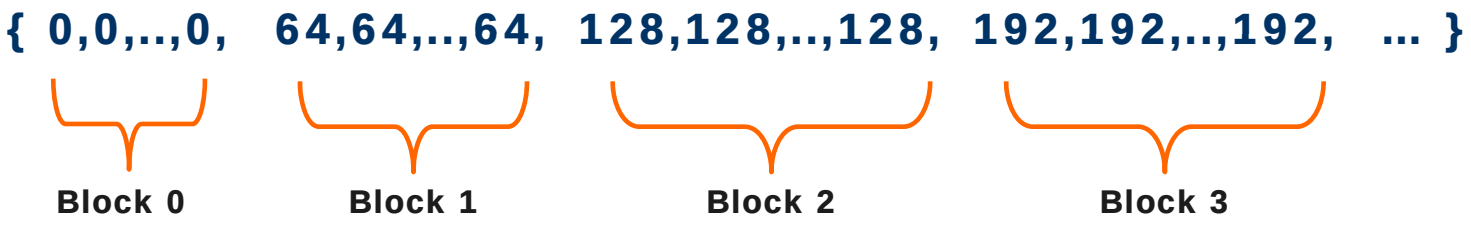


Merge or erase operations can be detected by comparing I/O time

Test Methods 1

❖ TEST1

- Feature of a modified block
- Write pages in the following sequence:



- Check if several consecutive merge operations are generated
  - If so, the block is mapped to a replacement block
  - Otherwise, the block is mapped to a log block

Test Methods 2

TEST2

- Page arrangement in a block
- Write pages in the following sequence:

{ 10,9,8, 74,73,21, 138,137,136, 202,201,200, … }

Block 0 Block 1 Block 2 Block 3

Assumption: # of replacement block per a block is 1

- Check if two consecutive merge operations are generated
  - If so, the replacement block always keeps pages arranged
  - Otherwise, the replacement block manages pages in an arbitrary order

Test Methods 3

❖ TEST3

- Log-block associativity
- Write a same page 128 times:

{ 0, 0, 0, 0, … ,0 }

Assumption: # of log block ≥ 2

- Check the cycle of merge operations
- If the cycle is coincident with the block size, the log block is block-associative
- Otherwise, the log block is not block-associative
  - Fully-associative or set-associative

Experimental Tests on Real Devices

❖ Target devices

- Samsung USB Flash Drive SUB-1G
  - OTI's OTI002168-G controller
  - Samsung's K9K8G08U0A NAND flash memory (SLC)
- SKY digital Swing Solo 1G White
  - Silicon Motion's SM3210F controller
  - Hynix' HY27UG088G5M NAND flash memory (SLC)
- SKY digital Swing Solo 1G Black
  - Silicon Motion's SM3210F controller
  - Samsung's K9G8G08U0M NAND flash memory (MLC)

Experimental Tests on Real Devices (cont'd)

❖ Operational characteristics

- Samsung USB Flash Drive SUB-1G
  - Sequential read: 908.29 usec
  - Sequential write: 944.77 usec
- SKY digital Swing Solo 1G White
  - Sequential read: 915.26 usec
  - Sequential write: 945.97 usec
- SKY digital Swing Solo 1G Black
  - Sequential read: 976.87 usec
  - Sequential write: 1030.47 usec

Experimental Tests on Real Devices (cont'd)

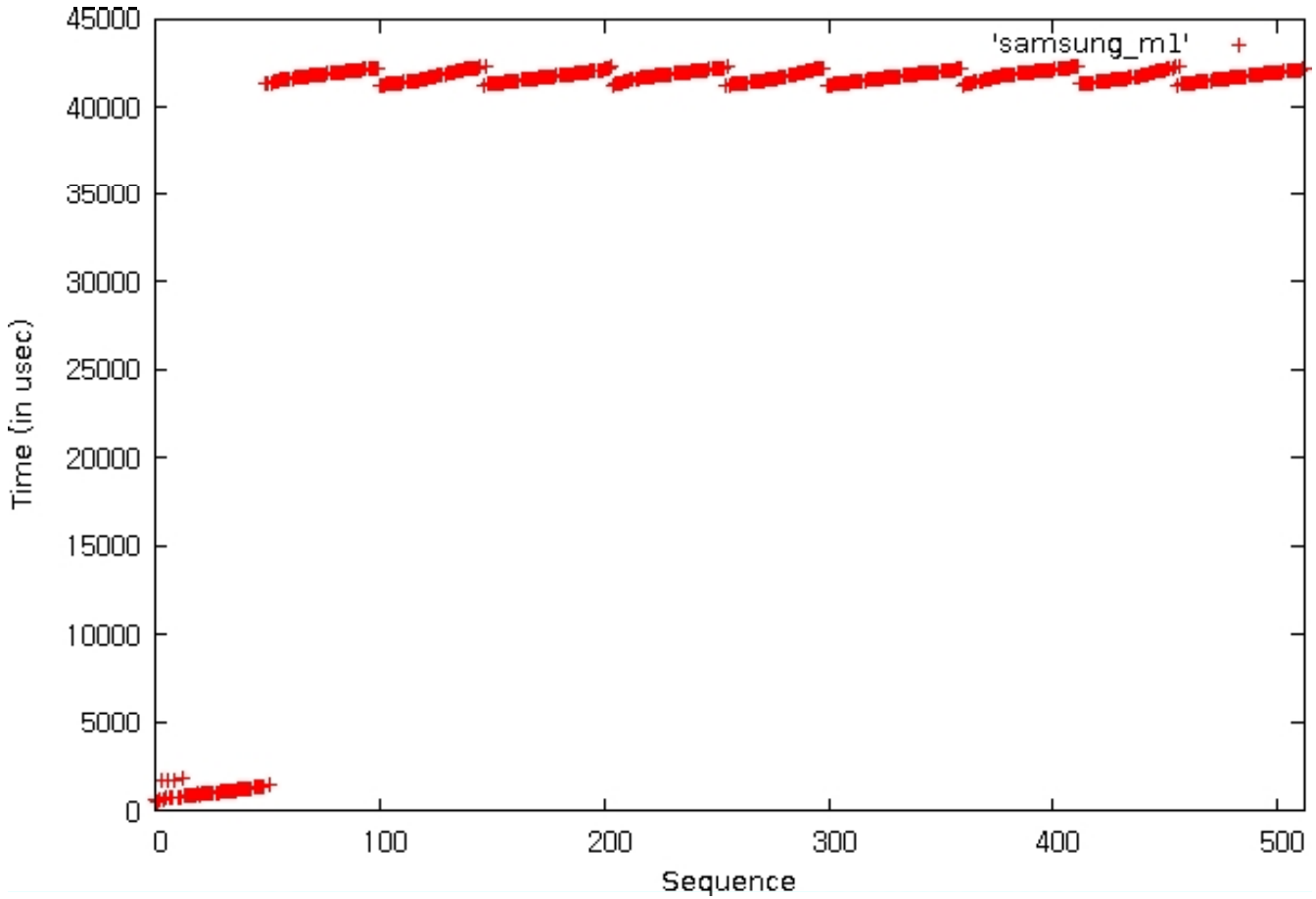
❖ Samsung 1G SLC

- TEST1 – Feature of a replacement block
  - Logical blocks are divided by two regions
    - Region1: Log-scheme replacement block (0~15)
    - Region2: Simple replacement block (16~)

- TEST2 – Page arrangement in a block (Region2)
  - Every write request on Resion2 generates merge
    - Pages in a block are always arranged

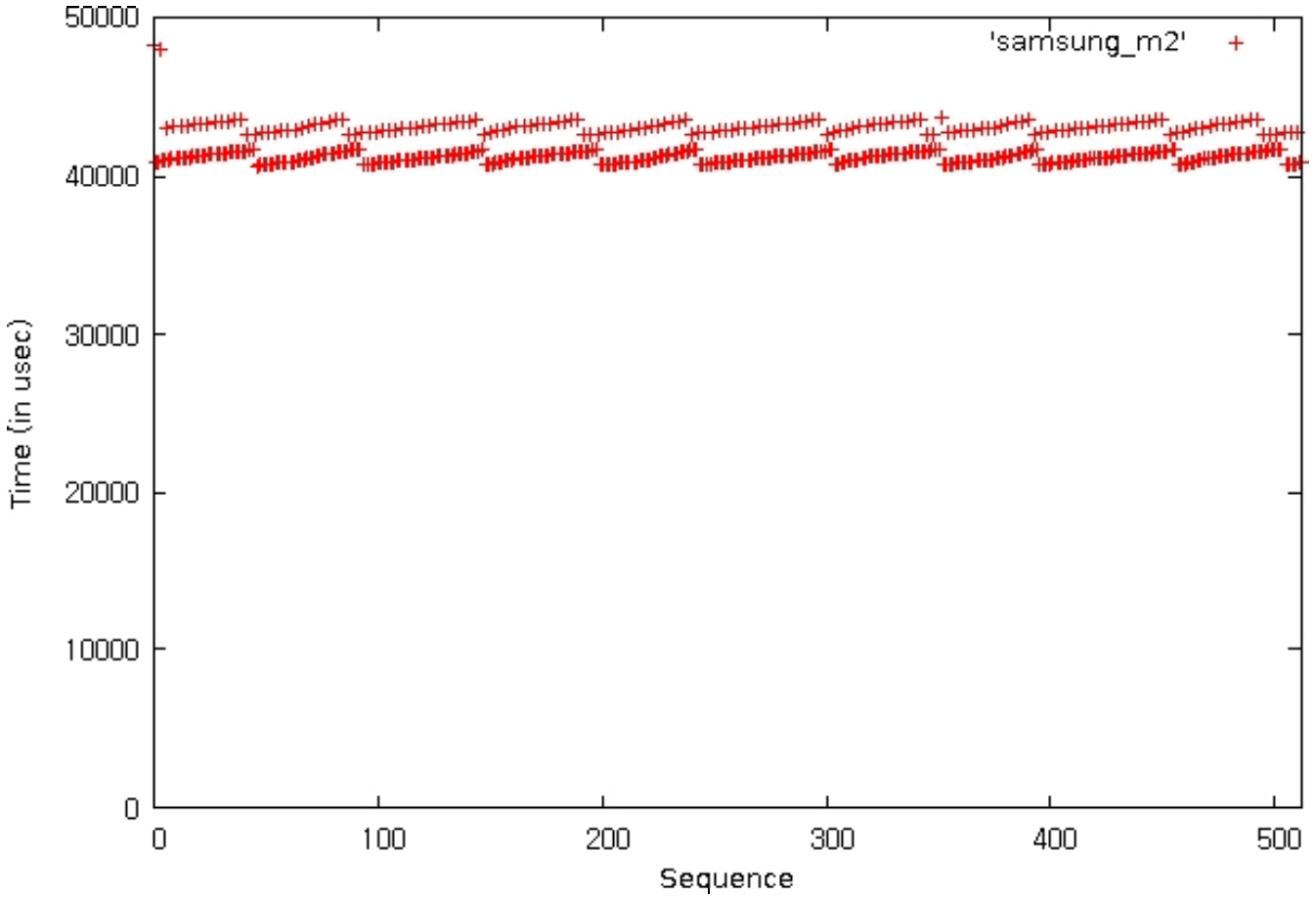
Experimental Tests on Real Devices (cont'd)

❖ TEST1 on Samsung 1G SLC



Experimental Tests on Real Devices (cont'd)

❖ TEST2 on Samsung 1G SLC



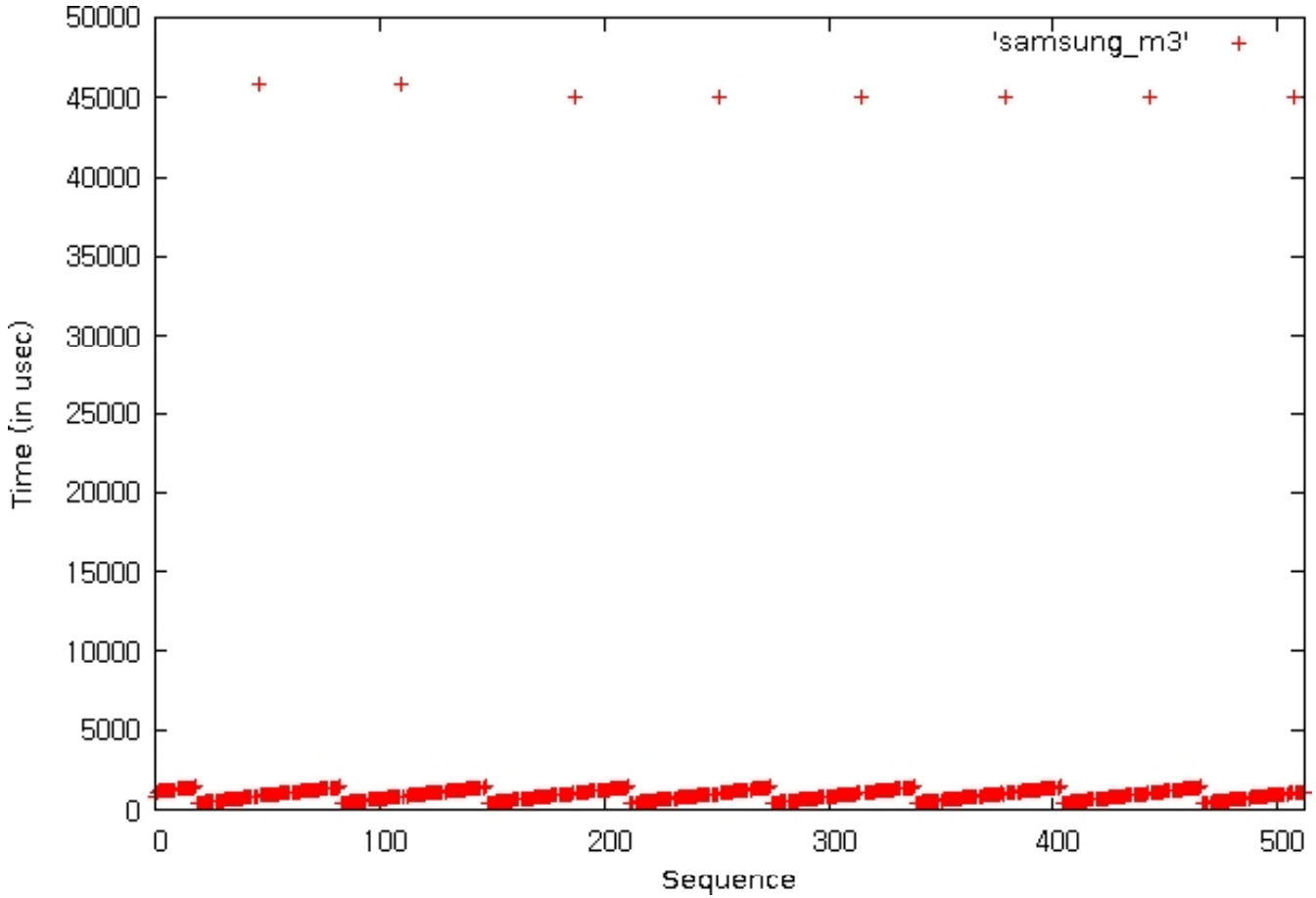
Experimental Tests on Real Devices (cont'd)

❖ Samsung 1G SLC (con'd)

- TEST3 – Log-block associativity (Region1)
  - Merge operations are generated after every 64th write
    - Log blocks are block-associative

Experimental Tests on Real Devices (cont'd)

❖ TEST3 on Samsung 1G SLC



Experimental Tests on Real Devices (cont'd)

Samsung 1G SLC (con'd)

- TEST4 – Number of log blocks (Region1)
  - First 16 blocks(2MB) are mapped to log blocks
  - How many log blocks does the area have?
    - Assume *n* log blocks exists, write following pages 64 times:
      { 0,64,128,…,64*(n-1) }
    - If the assumption is correct, no more than 3 merge operations are generated
  - Result
    - 16 data blocks
    - 16 log blocks corresponding to each of them

	seq 1	seq 2	seq 3		seq 64
Block 0	0	0	0	…..	0
Block 1	64	64	64	…..	64
	…..				
Block *n*	64*(n-1)	64*(n-1)	64*(n-1)	…..	64*(n-1)

Experimental Tests on Real Devices (cont'd)

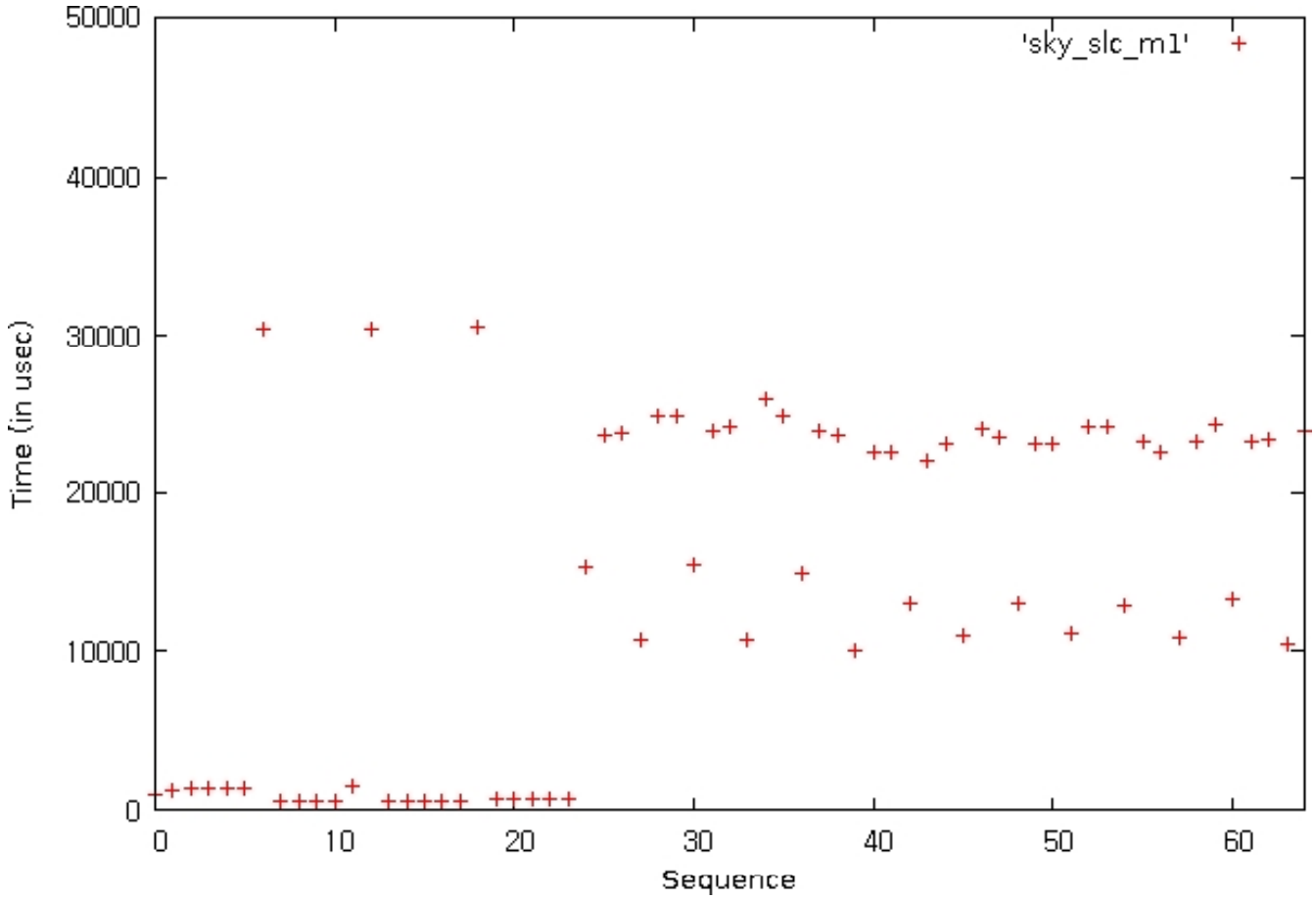
❖ SKY digital 1G SLC

- TEST1 – Feature of a replacement block
  - Logical blocks are divided by two regions
    - Region1: Log-scheme replacement block (0 ~ 7)
    - Region2: Simple replacement block (8 ~)

- TEST2 – Page arrangement in a block (Region2)
  - Every write request on Region2 generates merge
    - Pages in a block are always arranged

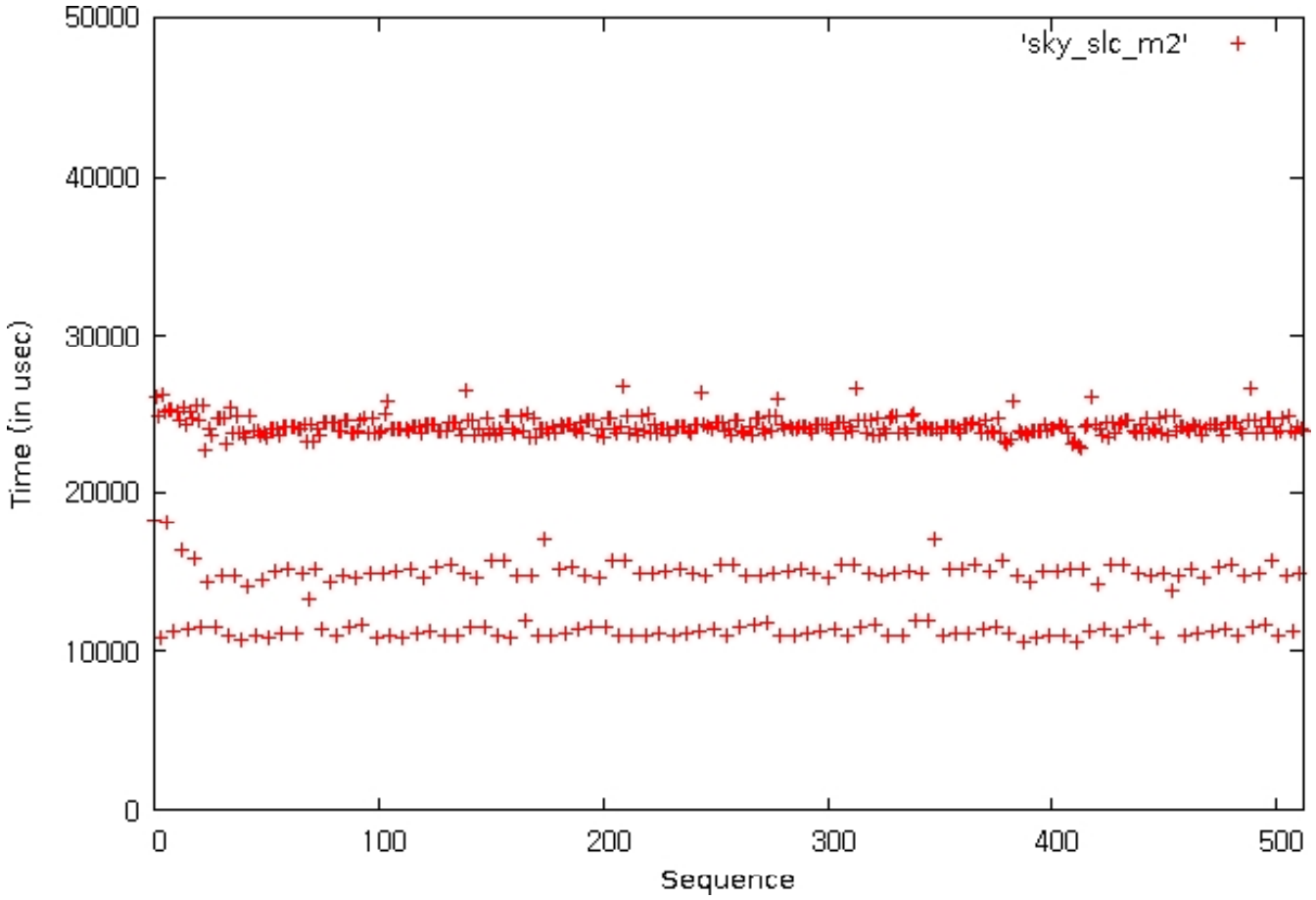
Experimental Tests on Real Devices (cont'd)

❖ TEST1 on SKY digital 1G SLC



Experimental Tests on Real Devices (cont'd)

❖ TEST2 on SKY digital 1G SLC



Experimental Tests on Real Devices (cont'd)

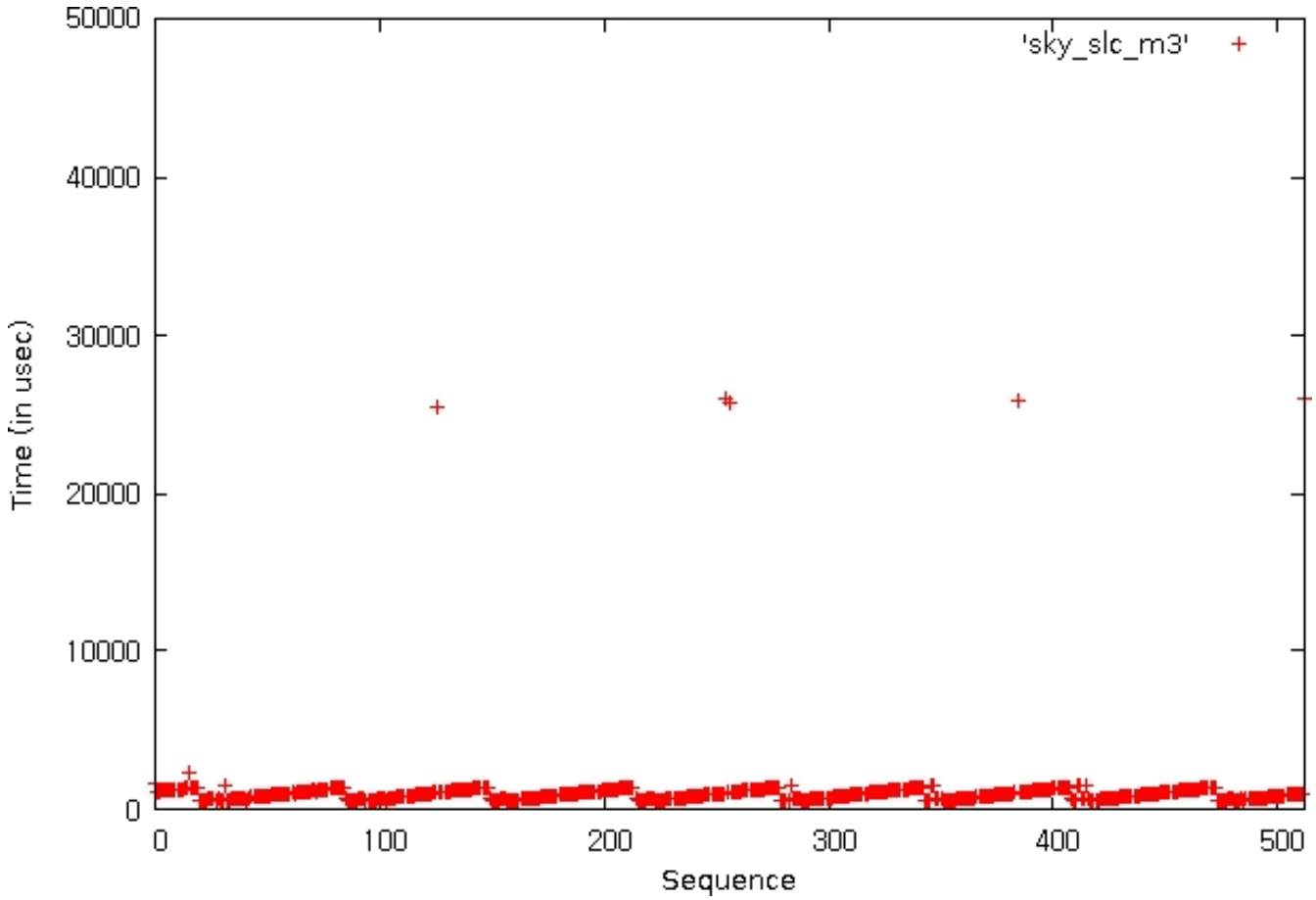
❖ SKY digital 1G SLC (cont'd)

- TEST3 – Log-block associativity (Region1)
  - Merge operations are generated after every 128th write
    - Log blocks are NOT block-associative

- TEST5 – Rate of data blocks and log blocks (Region1)
  - First 8 blocks(1MB) are mapped to log blocks
  - Set-associative log block
    - 4 log blocks for 8 data blocks
      - » 2 log blocks per 4 data blocks
    - Log blocks are gathered to be 2 128KB-chunks
    - Data blocks are gathered to be 4 128KB-chunks
  - 128 writes of { 0,256 } generates maximum 2 merges

Experimental Tests on Real Devices (cont'd)

❖ TEST3 on SKY digital 1G SLC



Experimental Tests on Real Devices (cont'd)

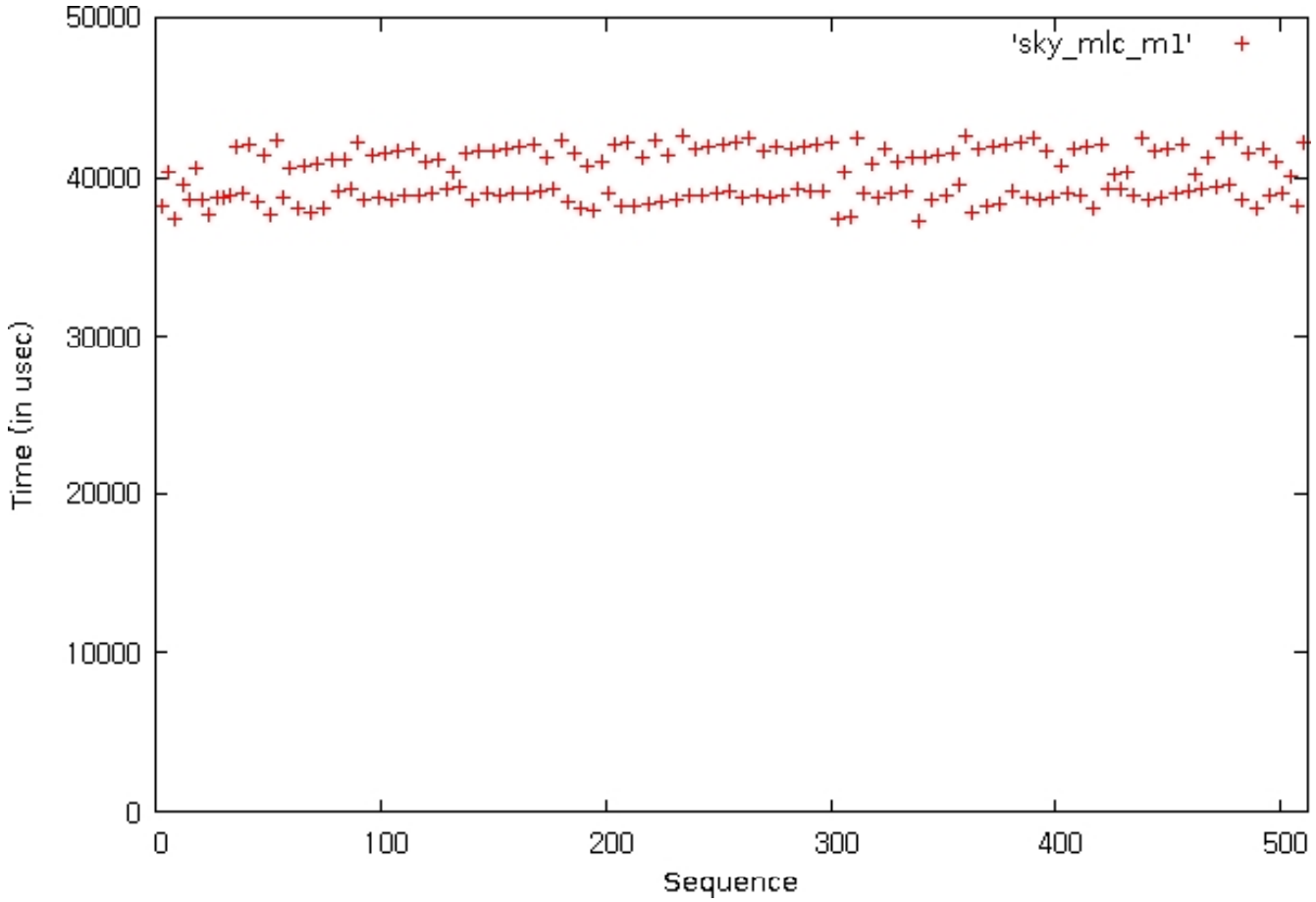
❖ SKY digital 1G MLC

- TEST1 – Feature of a replacement block
  - Every write operation generates merge
    - Simple replacement block

- TEST2 - Page arrangement in a block
  - Every write request on Region2 generates merge
    - Pages in a block are always arranged

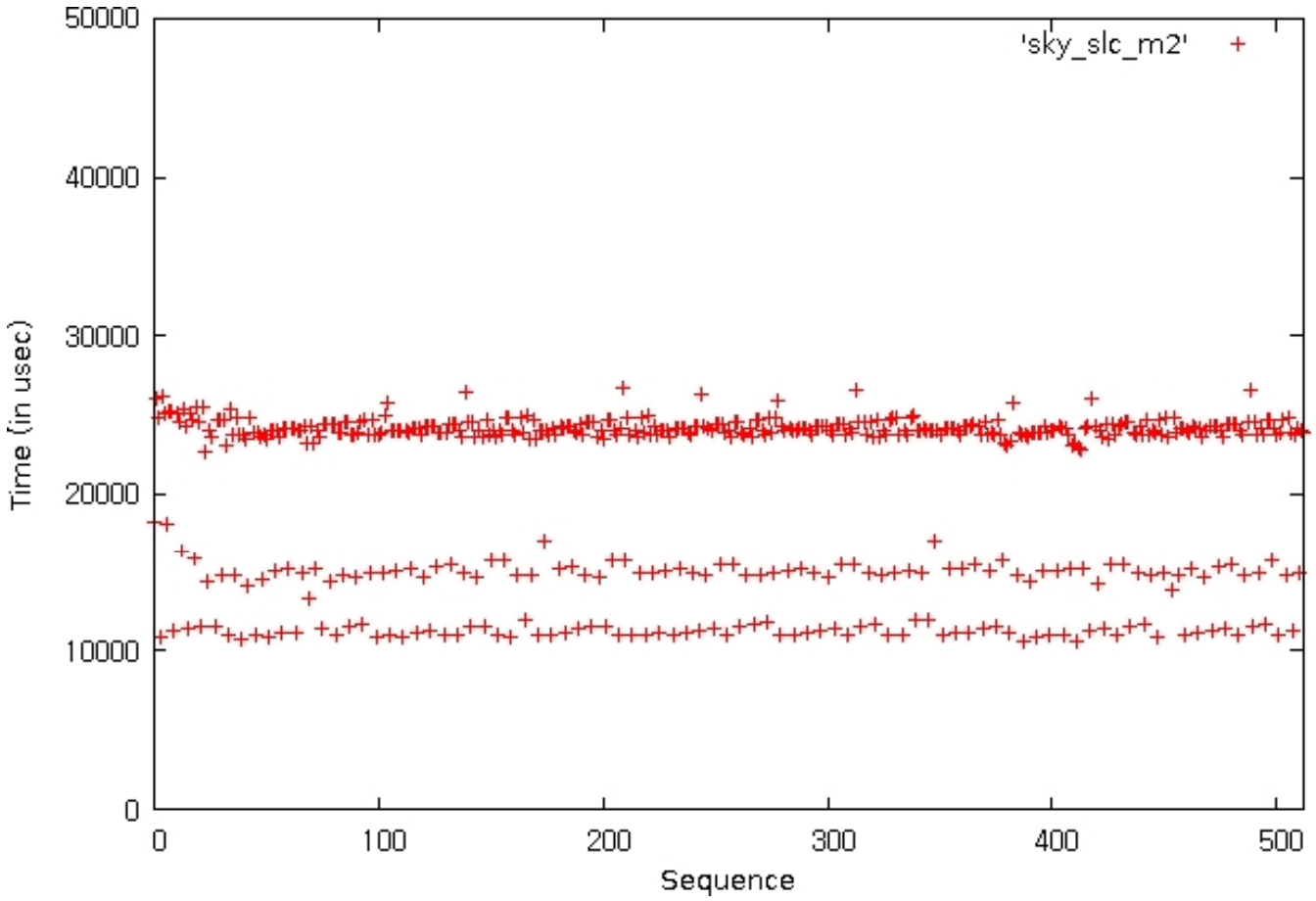
Experimental Tests on Real Devices (cont'd)

❖ TEST1 on SKY digital 1G MLC



Experimental Tests on Real Devices (cont'd)

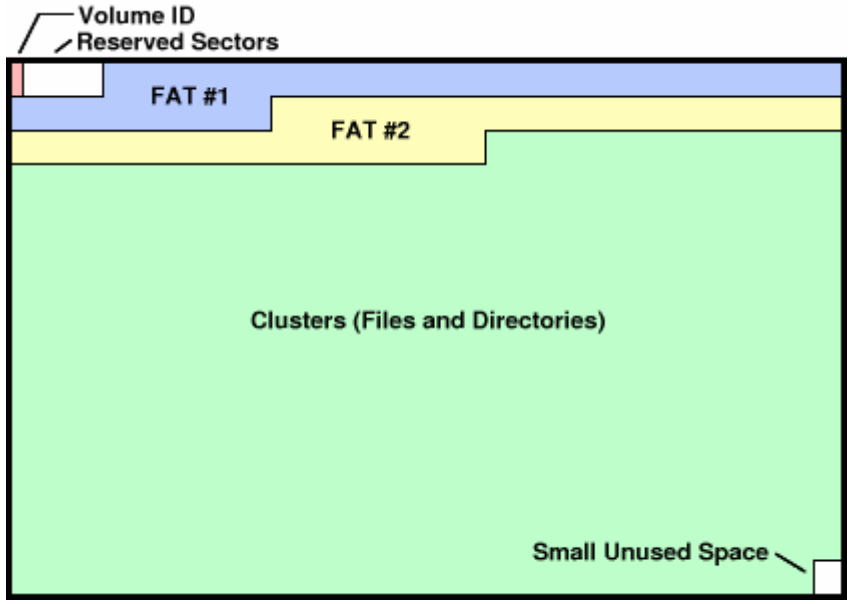
❖ TEST2 on SKY digital 1G MLC



Experimental Tests on Real Devices (cont'd)

❖ Why does two separate regions exist?

- Samsung 1G SLC, SKY digital 1G SLC
- Generally, FAT filesystem is used for USB flash drives
- FAT filesystem
  - FAT area
    - Frequently accessed and modified
  - Data area
    - Sequentially accessed and rarely modified



Log-block mapping

Block-level mapping

Conclusion & Further Issues

- Each USB flash device has its own mapping algorithm
- What about the SSD?
- Is the universal mapping algorithm for all the applications available?

Questions ?

Thank you !