FASTer FTL for Enterprise-Class Flash Memory SSDs

Sang-Phil Lim
Sungkyunkwan University

Sang-Won Lee
Sungkyunkwan University

Bongki Moon
University of Arizona

Table of Contents

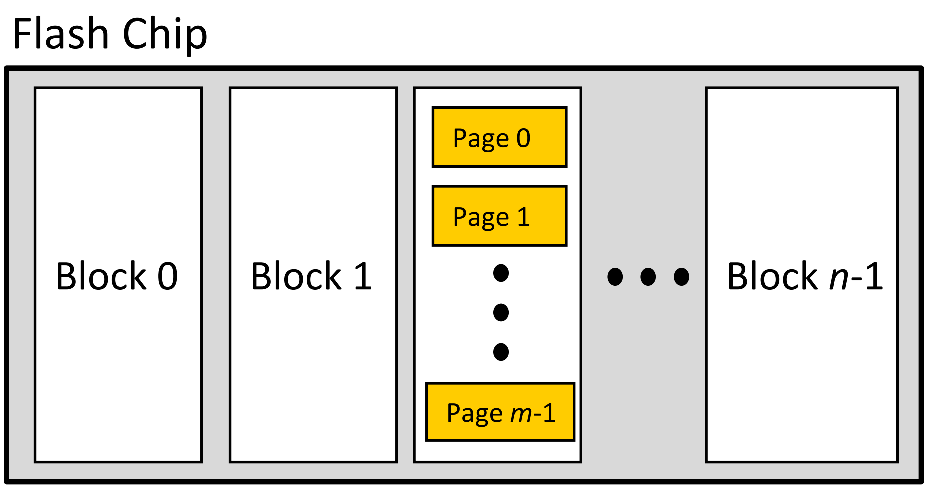
- **Motivation**
- **Background**
  - NAND Flash Memory and Flash Translation Layer (FTL)
- **FAST FTL and OLTP Write Patterns**
- **FASTer FTL for OLTP Applications**
- **Performance Evaluation**
- **Conclusion and Future Work**

Motivation

- **FAST FTL** [Sang-Won Lee et al, ACM TECS '07]
  - Originally designed for random writes
- **FAST has been criticized** [**DFTL**: Aayush Gupta et al, ASPLOS '09, **LAST**: Sungjin Lee et al, SPEED '08]
  - With 3% log space, performance and fluctuation
  - No special mechanism for hot/cold separation
- **A large scale flash SSD**
  - For better performance, it can employ larger log space
- **Revisit FAST with OLTP workloads**
  - Cost competitiveness

Background : NAND Flash Memory

- **NAND Flash memory organization & chip-level performance**



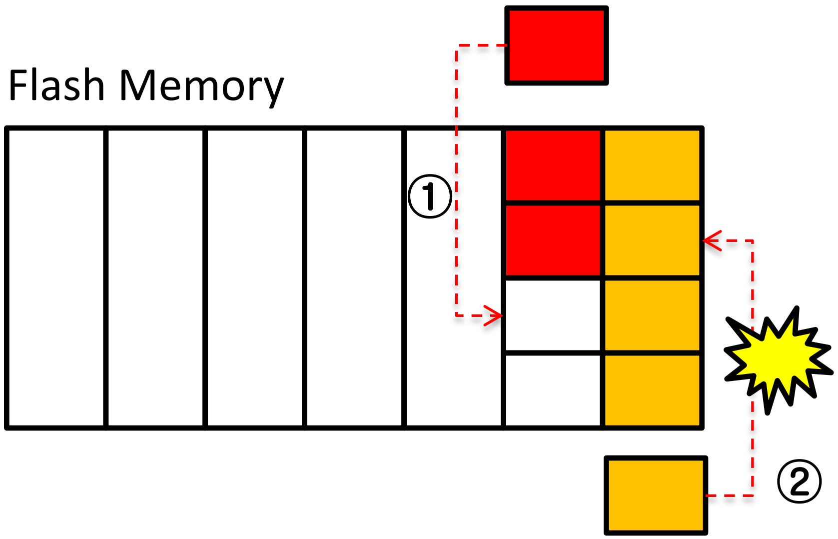
Media	Access latency		
	Read	Write	Erase
NAND Flash [1]	25us (2KB)	200us (2KB)	1,500us (128KB)
HDD [2]	12.7ms (2KB)	13.7ms (2KB)	-

[1] : Samsung Large-block SLC NAND (K9WAG08U1A)
[2] : Seagate 7200rpm Hard Disk (ST380011A)

- **Limitations**
  - 'Erase-before-write' : No in-place update
  - Data can only be written sequentially
  - Block wears out after 100K erases

Background : NAND Flash Memory

- '**<u>Erase-before-write</u>**' **limitation**



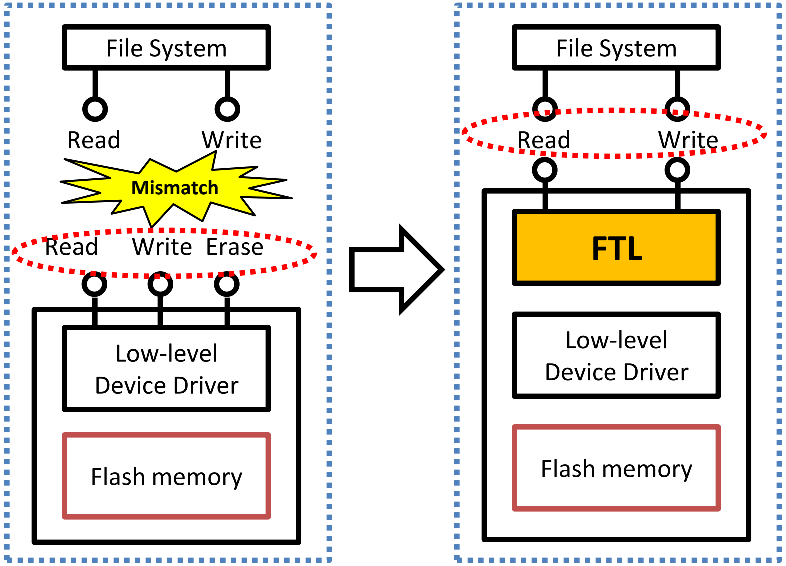
① **New write (2KB) :** 0.2ms

② **Overwrite (2KB)**
- 63 page copy-backs
- 1 new page write
- 1 old block erase

≒ 16ms (> 80x new page write)

Background : Flash Translation Layer(FTL)

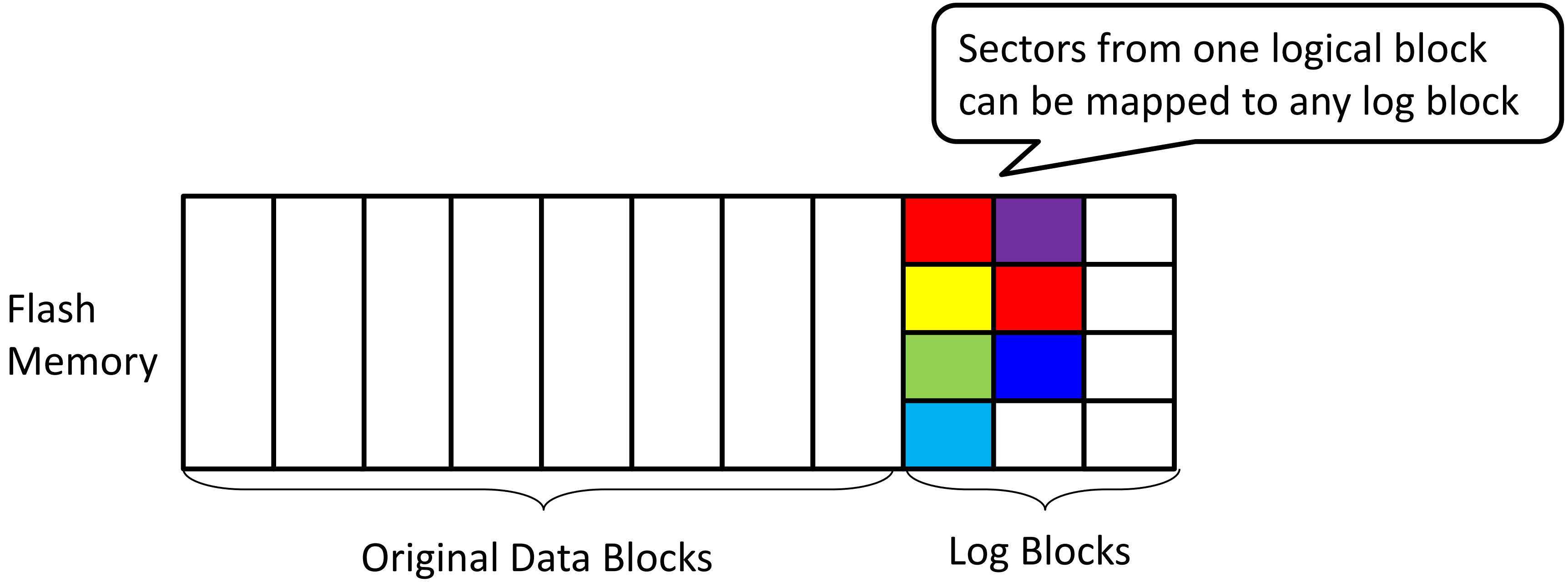
- **A software layer that allows the flash memory to look like a HDD**
  - Address mapping : logical to physical
  - Garbage collection & power-off recovery
  - Wear-leveling & bad block management
  - etc.



- **Popular FTL algorithms**
  - FMAX, BAST, FAST, Super block, LAST
  - DFTL, DAC, etc...

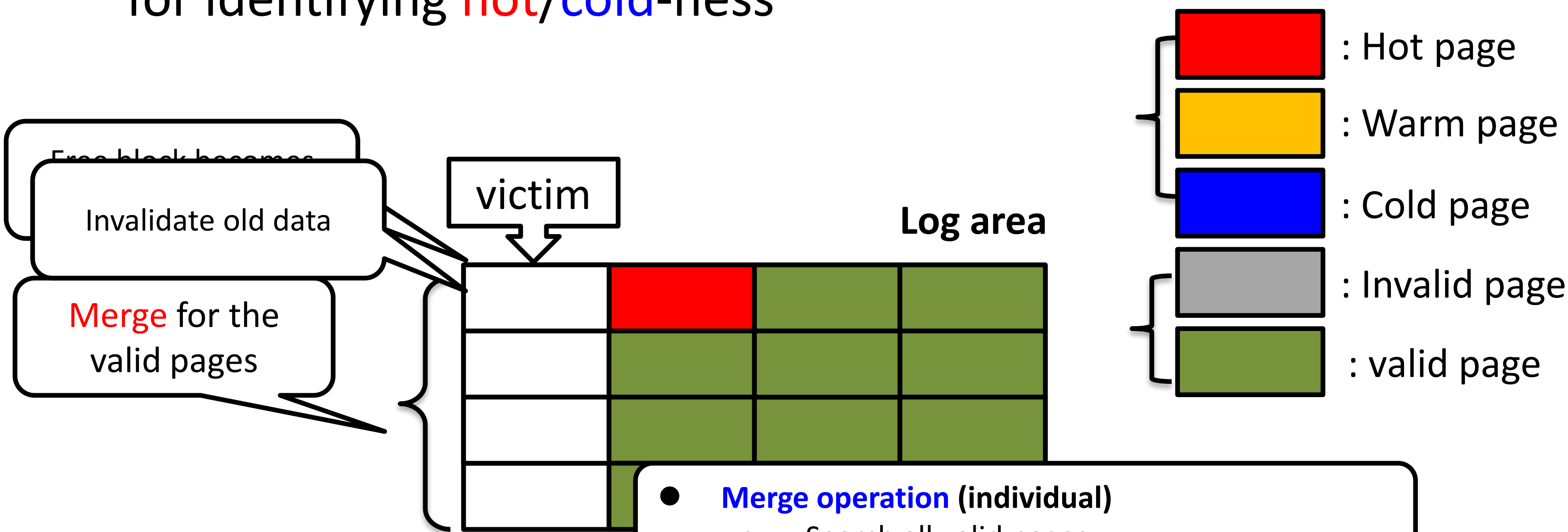
FAST FTL

- A popular log-based FTL [Sang-Won Lee et al, ACM TECS '07]
- FAST FTL is designed for small random writes



FAST FTL vs. Temporal Locality

- FAST FTL can handle temporal locality without any overhead for identifying hot/cold-ness

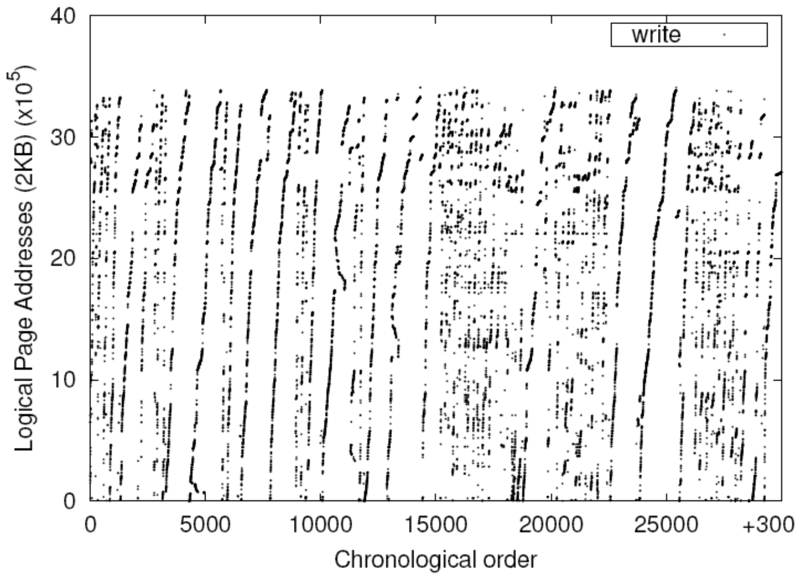


- **Merge operation (individual)**
  - Search all valid pages
  - 64 page copy-backs (to free block)
  - Replace the free block to new data block
  - 1 old data block erase

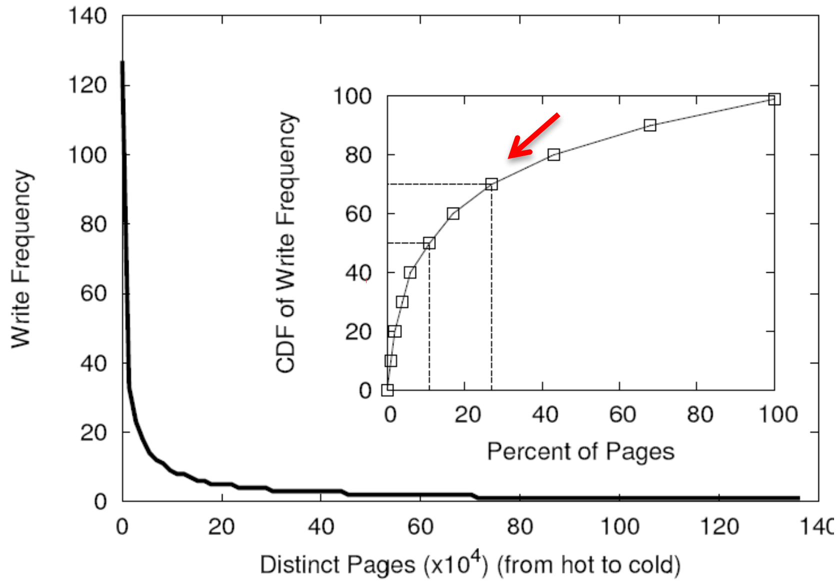
  ≒ 16ms (> 80x new page write)

Write Patterns in OLTP applications

- Randomly scattered over a large address space
- <u>Write skewed</u> : non-uniformly distributed access frequencies



(a) Time-Address Distribution



(b) Write Frequency Distribution

Impact of Write Intervals on FTLs

- We classified the pages using the concept of '*write interval*'
  - **Hot**/**Warm**/**Cold** page
  - Temporal locality in OLTP workload
- OLTP write patterns may match well for FAST FTL
- Write interval vs. log window size ?

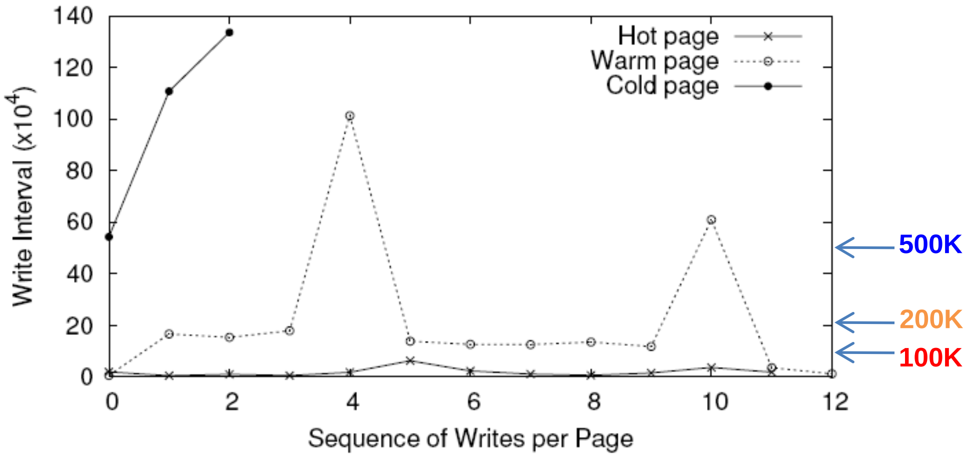


Figure 3. Write intervals: hot, warm and cold pages

Criticism of FAST FTL

- **FAST FTL is criticized in 'DFTL'** [Aayush Gupta et al, ASPLOS `09]
  - They said…"FAST dose not provide any special mechanism to handle temporal locality in random streams."
  - With 3% log space, FAST shows poor performance and high variation

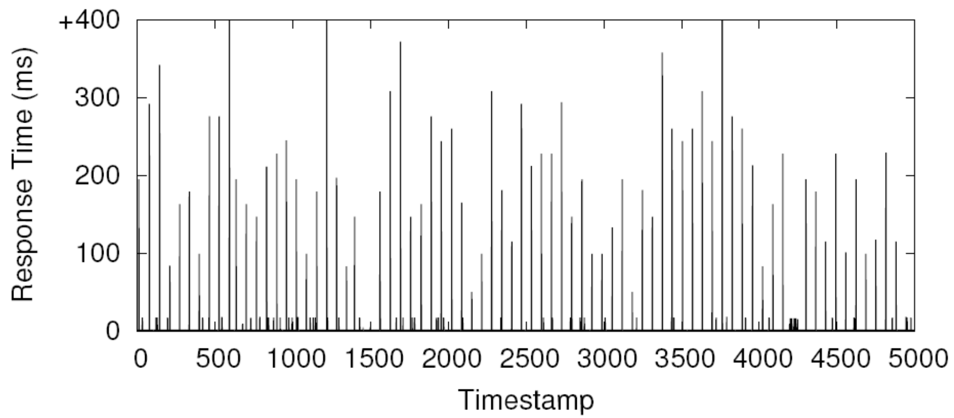
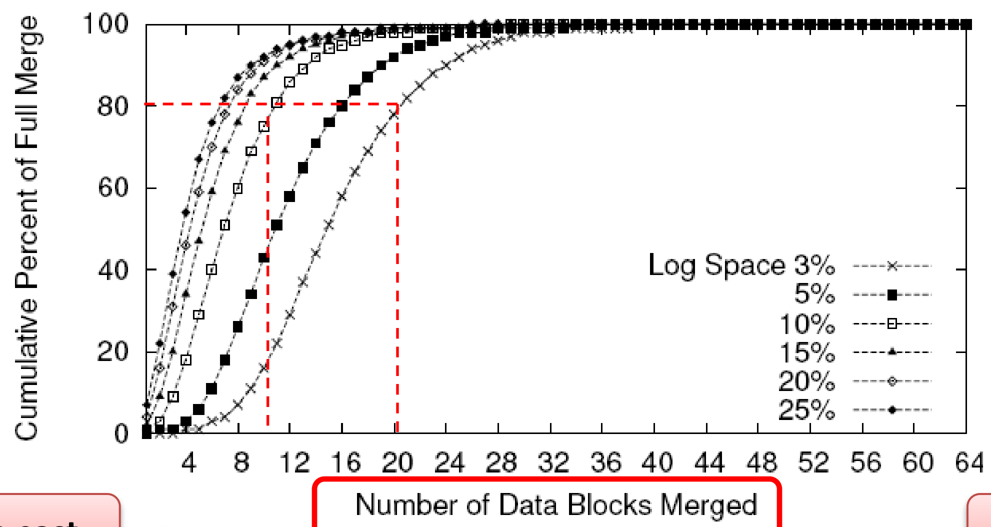


Figure 1. Write response times with FAST (log space: 3%)

Impact of log window size in FAST FTL

- With larger log space, FAST can exploit temporal locality!
- Trade-off : manufacturing cost vs. throughput



Low merge cost ↔ **High merge cost**

Figure 4. Cumulative distribution of a full merge cost

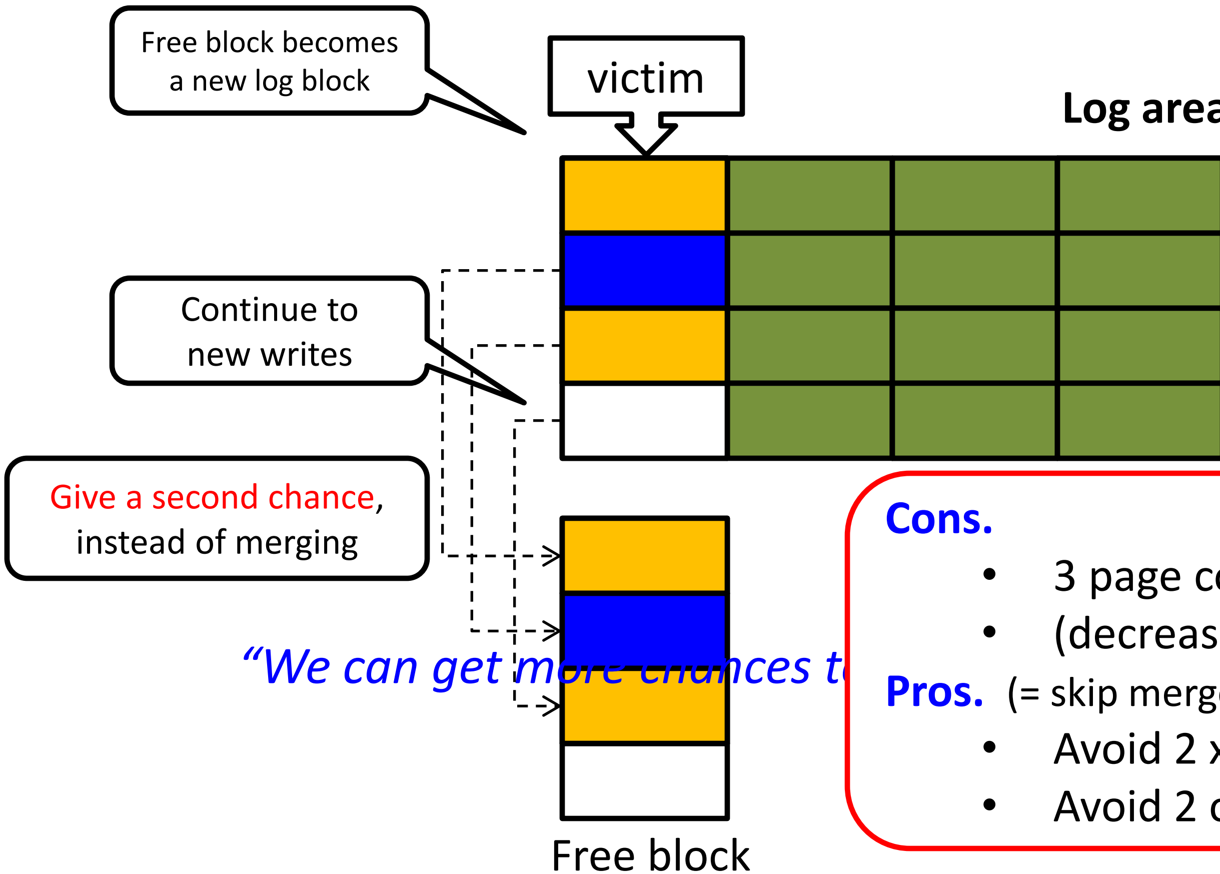
FASTer FTL : Introduction

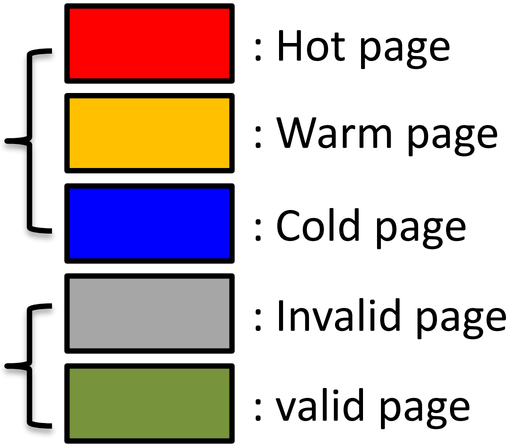
- **FASTer FTL for OLTP workloads**
  - A new FTL scheme which is enhancement of FAST FTL
  - Better performance than FAST
  - Uniform response time

- **Main key ideas**
  - Second chance policy
  - Isolation area

FASTer FTL : Key ideas

1) Second chance policy





“We can get more chances t

Pros. >> Cons.

Cons.

- 3 page copy-backs
- (decrease log block utilization)

Pros. (= skip merges for the **‘warm’** pages)

- Avoid 2 x (64 page copy-backs)
- Avoid 2 old data block erases

FASTer FTL : Key ideas

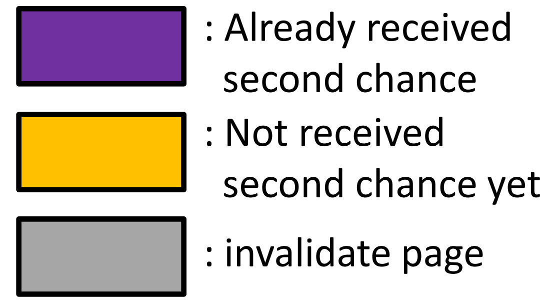
1) Second chance policy

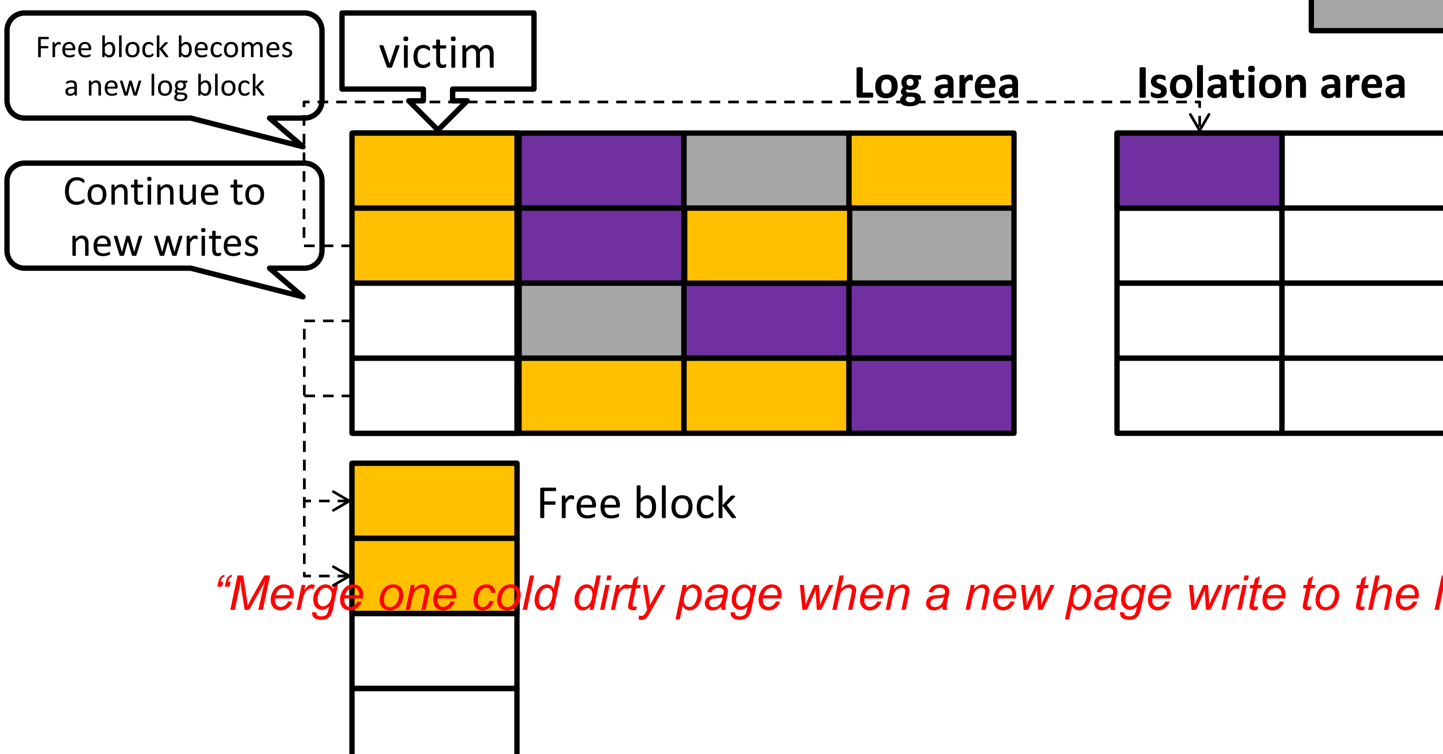
- "Give the second chance for the **'warm'** pages to be invalidated"
- Exploit temporal locality more by doubled log window
- Pros. & Cons. of second chance policy
  - But, Pros. >> Cons.

FASTer FTL : Key ideas

2) Isolation area

- Write buffering for **'cold'** pages
- Mitigate response time fluctuation

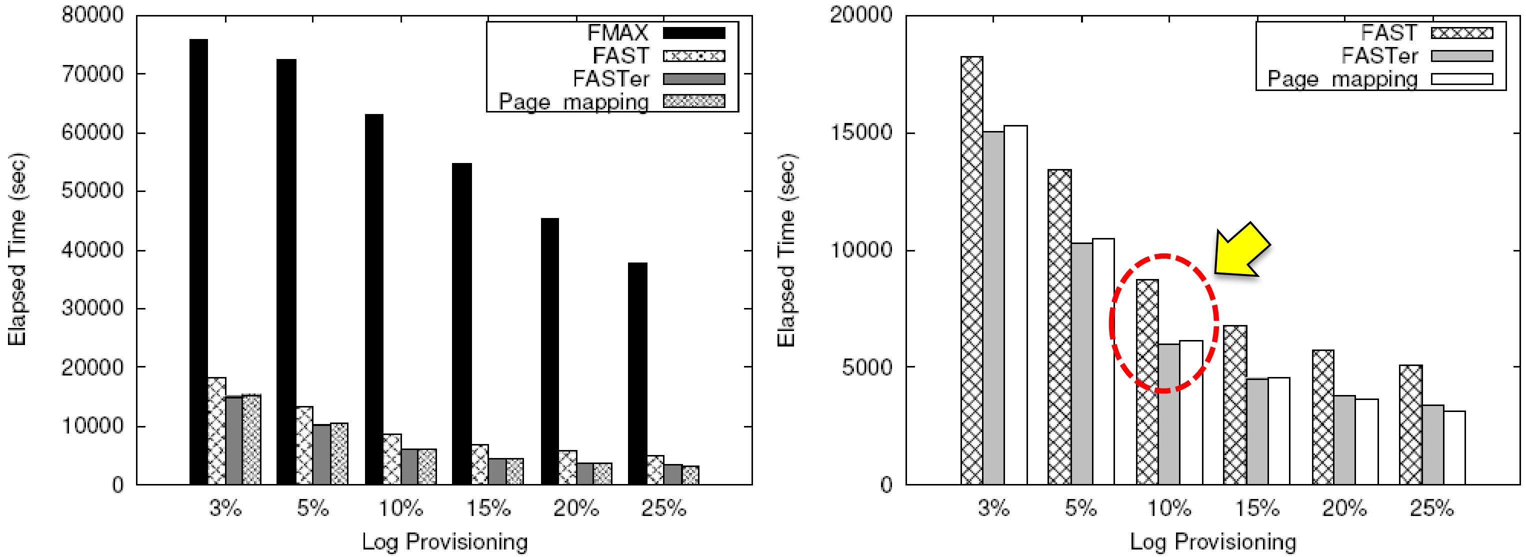




"Merge one cold dirty page when a new page write to the log block"

Performance Evaluation

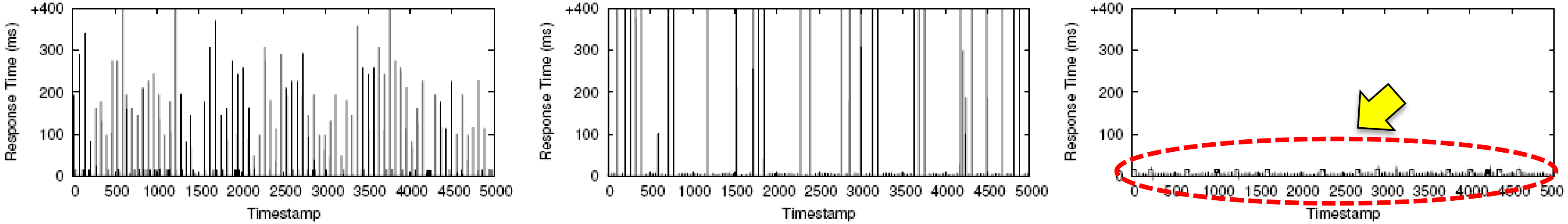
- ***FASTer* shows better throughput than others**
  - Outperformed FAST by more than 30 percent in elapsed time
  - Even similar with page-level mapping FTL [A. Kawaguchi et al, TCON '95]



(a) Elapsed time

Performance Evaluation

- ***FASTer*** **also shows uniform response time**



(a) FAST (b) *FASTer* (without and with an isolation area)

Figure 7. Response time variations with FAST and *FASTer* (log space : 3%)

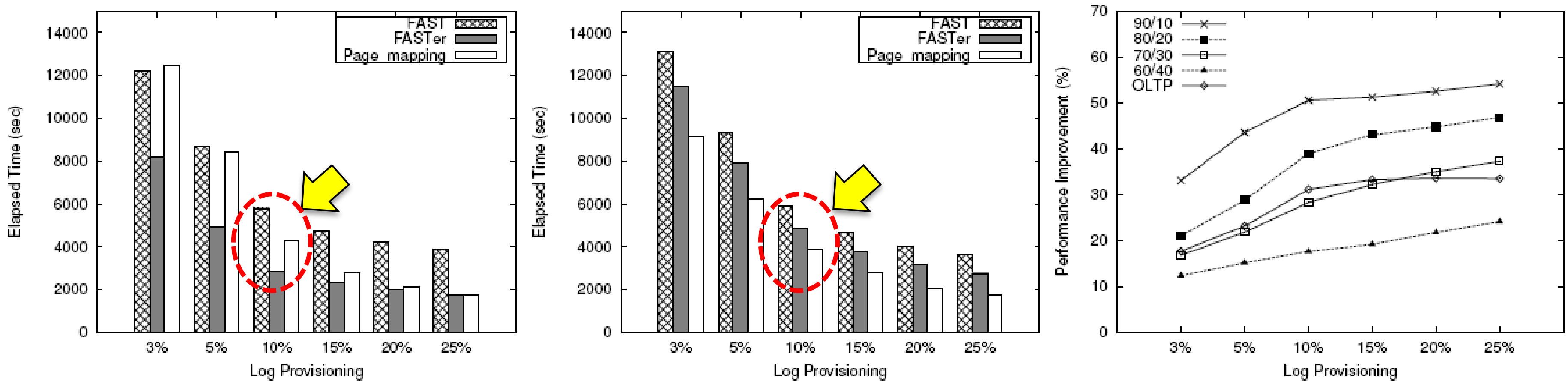
Log Space (%)		3	5	10	15	20	25
Average Response Time (ms)	FAST	3.59	2.64	1.71	1.33	1.12	1.00
	FASTer (without isolation area)	3.11	2.11	1.24	0.92	0.76	0.66
	FASTer (with isolation area)	3.04	2.08	1.20	0.90	0.75	0.66
	Page mapping	3.00	2.05	1.20	0.89	0.72	0.61
Standard Deviation of Response Time (ms)	FAST	27.6	19.9	12.2	9.01	7.20	6.19
	FASTer (without isolation area)	38.9	30.8	21.6	17.3	14.6	12.7
	FASTer (with isolation area)	5.99	5.00	3.66	3.02	2.64	2.40
	Page mapping	5.73	4.74	3.44	2.77	2.32	2.01

Table II
RESPONSE TIME COMPARISON

Performance Evaluation

- **Effect of write 'skewedness' degree**

c.f) (x/y) means x% of writes are directed to y% of pages.



(a) 90/10 (b) 60/40 (c) Improvement ratio between FAST and *FASTer*

Figure 8. Performance comparison of non-OLTP workloads (synthetic workloads generated using a modified IOzone tool [2])

Conclusion and Future Work

- Recent trends in NAND technology have made SSDs more viable in the enterprise storage market
- In this paper, we proposed FASTer FTL as an enhancement of the FAST FTL
- In the future, we will explain FASTer FTL more theoretically and evaluate with various real workloads

Thank you for your attention

Questions & Answers