ADAPT: Efficient Workload-sensitive Flash Management Based on Adaptation, Prediction and Aggregation

Chundong Wang and Weng-Fai Wong

National University of Singapore

Outline

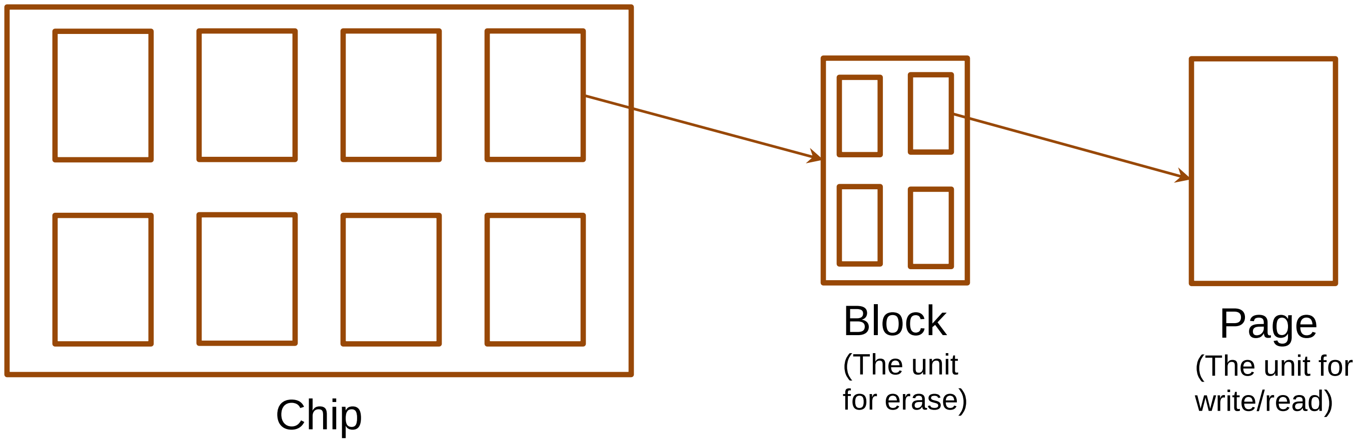
- Introduction
  - NAND Flash Memory
  - FTL and workload
- Background of hybrid mapping
- ADAPT
  - Adaptive partitioning of log space
  - Prediction and Aggregation
- Evaluation
- Conclusion

Outline

- Introduction
  - NAND Flash Memory
  - FTL and workload
- Background of hybrid mapping
- ADAPT
  - Adaptive partitioning of log space
  - Prediction and Aggregation
- Evaluation
- Conclusion

Introduction

- Characteristics of NAND flash memory
  - Three operations: write, read and erase



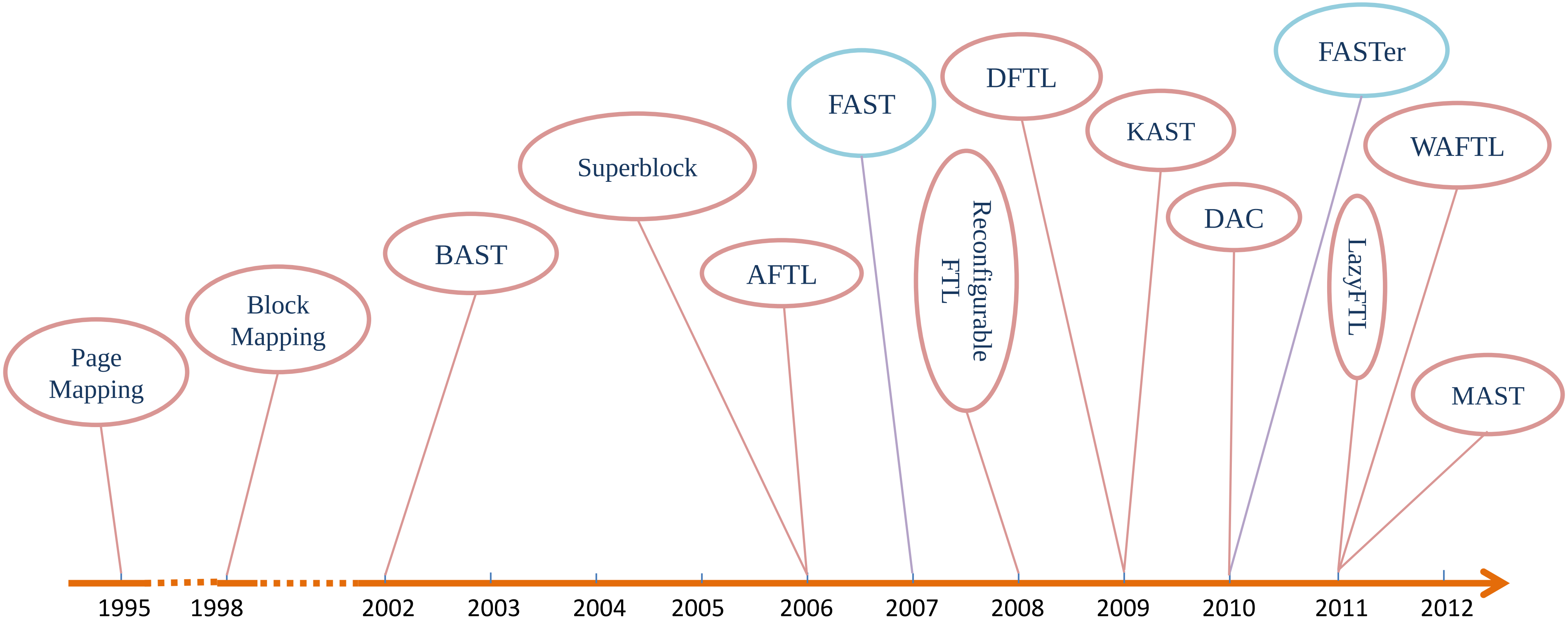
  - Out-of-place updating
    - A page cannot be programmed (written) unless its block is erased first;
    - A block usually comprises multiple pages.

Introduction

- FTL: Flash Translation Layer
  - Embedded software for flash management
  - Functionalities:
    - Address mapping
    - Wear leveling
    - Garbage collection
    - Bad block management

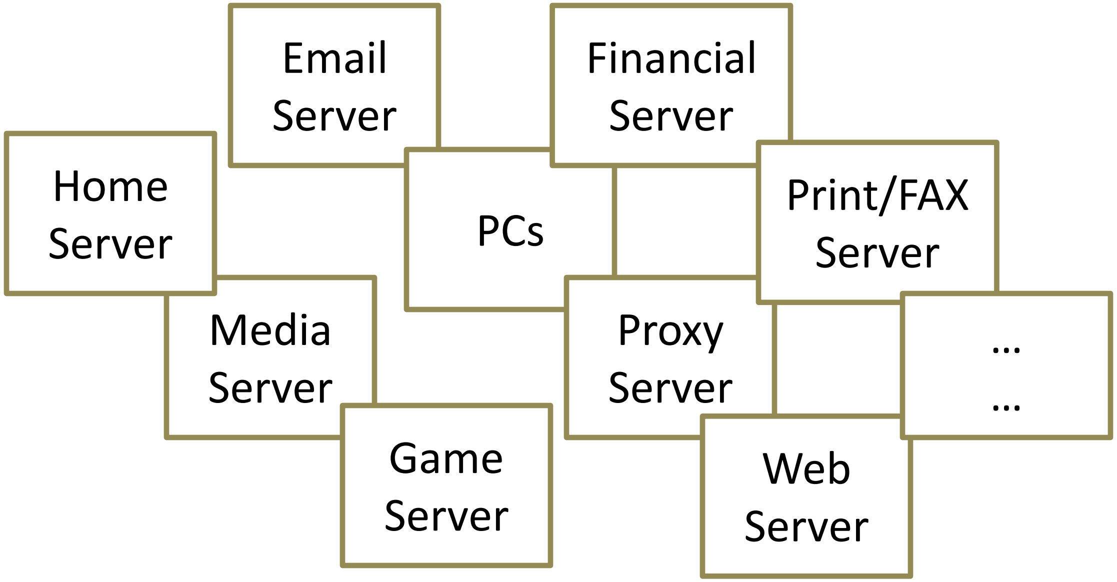
Introduction

- The roadmap of FTL (address mapping)



Introduction

- Various workloads
  - ➢ Distinct access behaviors to secondary storage



Introduction

- Workloads' characteristics: I/O request size.

Workload	Small I/O (%)	Medium I/O (%)	Large I/O (%)
TPC-C_20	99.17	0.83	0.00
SPC1	86.58	10.63	2.79
MSR-hm_0	76.70	13.72	9.58
MSR-mds_0	72.35	19.79	7.86
MSR-prn_0	79.46	8.88	11.66
MSR-prxy_0	87.91	6.82	5.27
MSR-rsrch_0	68.22	25.04	6.74
MSR-stg_0	72.33	18.62	9.05
MSR-ts_0	67.81	25.87	6.32
MSR-web_0	67.50	23.85	8.65

Introduction

- A workload:
  - is mixed by sequential and random requests;
  - but has stable access behaviors.

- The impact on FTL design by workloads:
  - An FTL may be designed for one type of workload, e.g., FASTer for OLTP systems (SNAPI 2010);
  - or be workload-adaptive, e.g, WAFTL (MSST 2011).

NUS National University of Singapore

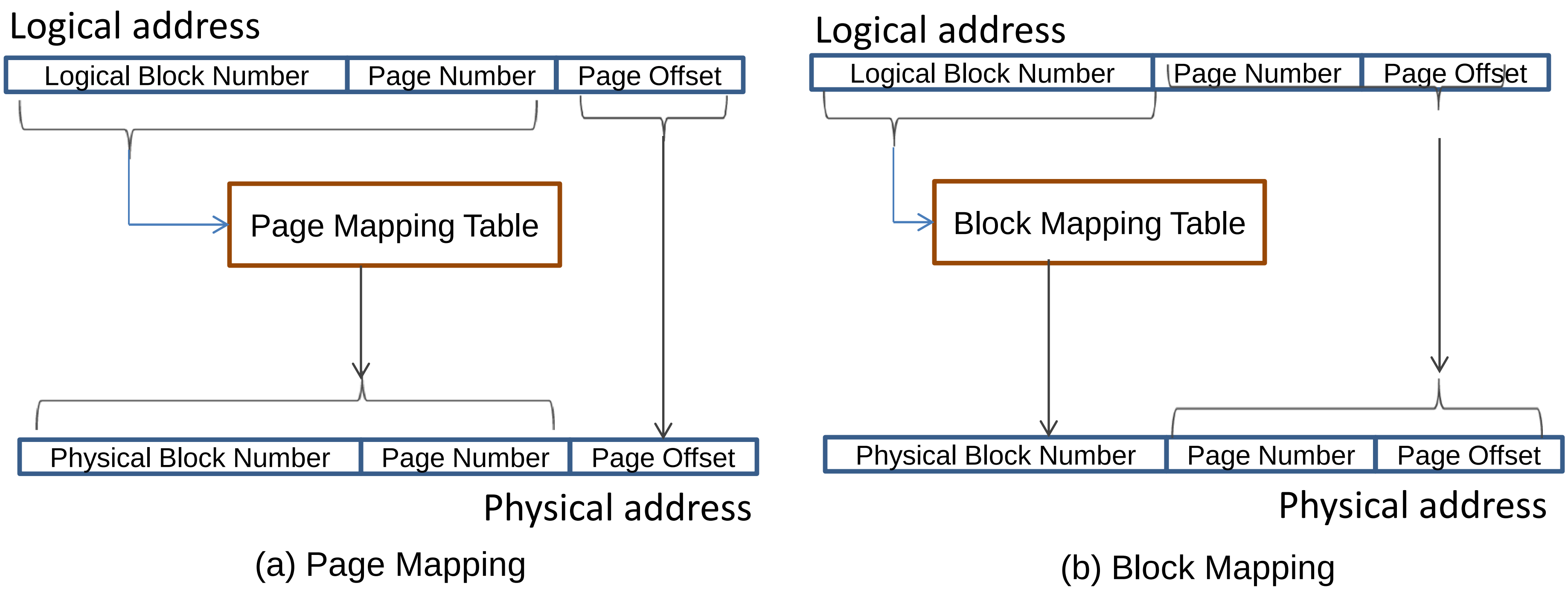
Outline

- Introduction
  - NAND Flash Memory
  - FTL and workload
- Background of hybrid mapping
- ADAPT
  - Adaptive partitioning of log space
  - Prediction and Aggregation
- Evaluation
- Conclusion

NUS National University of Singapore

Hybrid Mapping

- Page-level mapping and block-level mapping;
- Hybrid mapping is a combination of them.



Hybrid Mapping

- Why to be “hybrid”:
  - Block mapping
    - Space economic
    - Inflexible
  - Page mapping
    - Fine granularity
    - Large mapping table;
  - Hybrid mapping takes advantage of them.

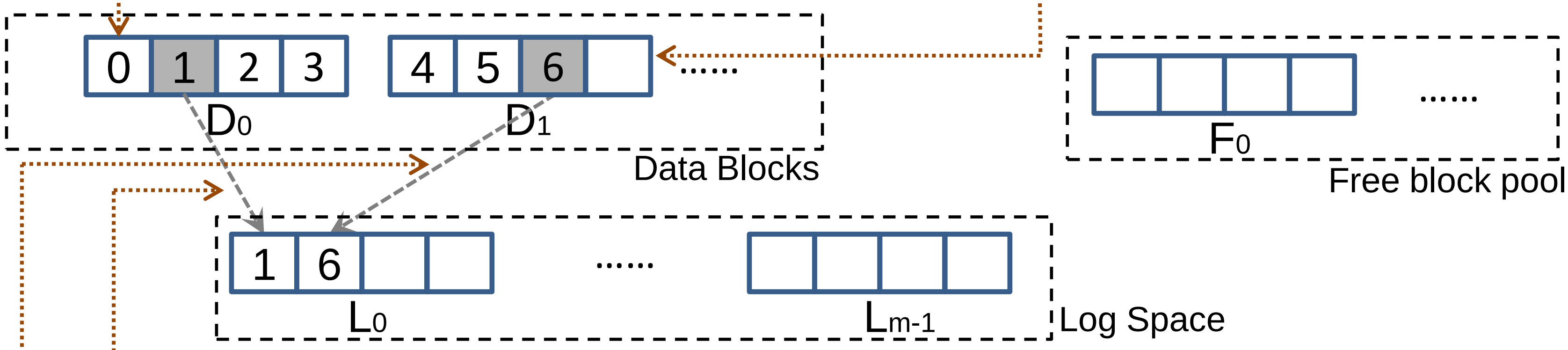
Hybrid Mapping

- How to be “hybrid”:
  - Partitions of physical blocks
    - Data blocks: block mapping;
    - Log space: page mapping;
    - Free block pool: to provide clean blocks.
  - Log space is like a cache to data blocks.

Hybrid Mapping

Block Mapping Table

Logical Block No	Physical Block No
LB_0	D_0
LB_1	D_1
……	……



Log Page Mapping Table

Data Block No	Data Page No	Log Block No	Log Page No
D_0	1	L_0	0
D_1	2	L_0	1
……	……	……	……

Hybrid Mapping

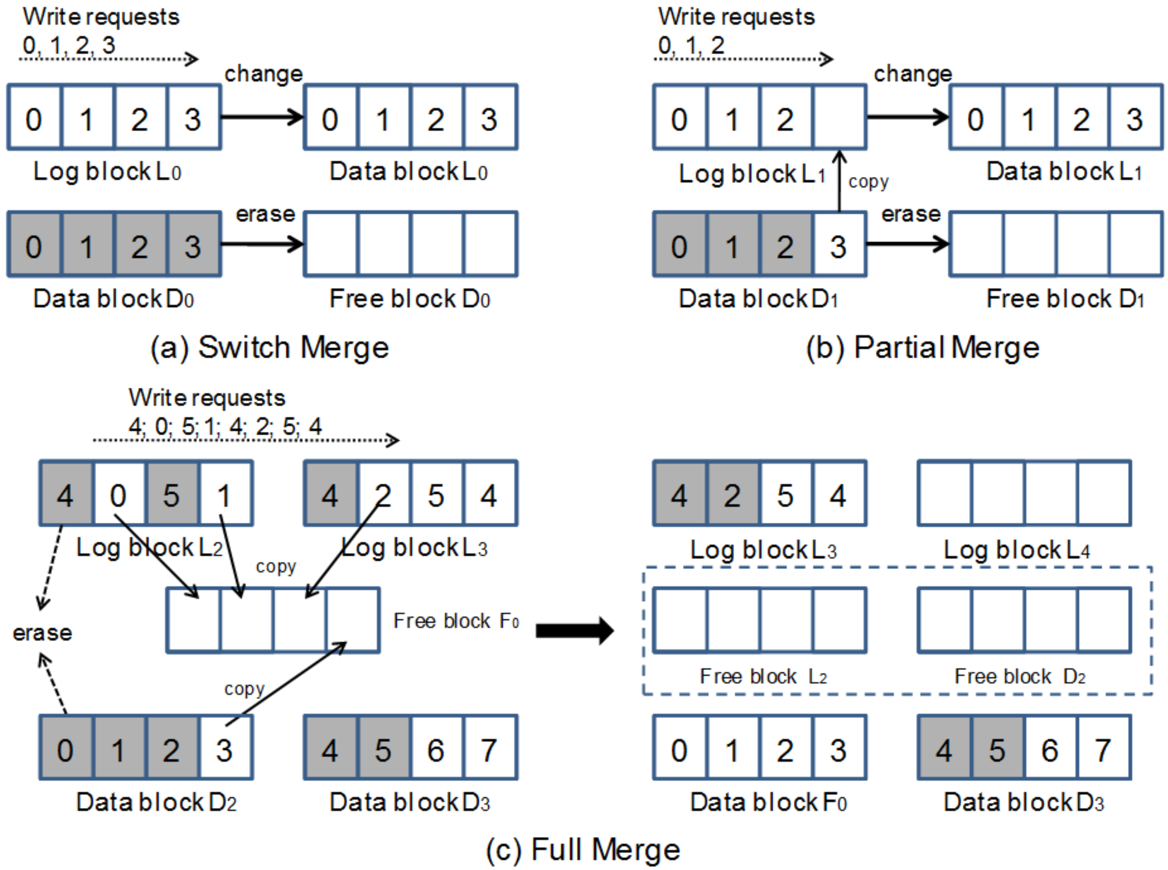
- Log space is further partitioned
  - Sequential area for sequential requests
  - Random area for random requests
- When log pages are used up, merge is called.
- Merge: to make room in log space
  - Switch merge
  - Partial merge
  - Full merge (random area)

More preferred (sequential area)

NUS National University of Singapore

Hybrid Mapping

- Three types of merge



(This figure is adapted from LAST of Lee et.al. in SIGOPS Oper. Syst. Rev.)

Outline

- Introduction
  - NAND Flash Memory
  - FTL and workload
- Background of hybrid mapping
- ADAPT
  - Adaptive partitioning of log space
  - Prediction and Aggregation
- Evaluation
- Conclusion

NUS National University of Singapore

ADAPT

- Overview
  - ADAPT is a hybrid mapping scheme
    - Fully-associative, like FAST and FASTer
    - Log blocks will be managed in a FIFO queue
  - ADAPT's components:
    - Adaptive partitioning of log space
    - Predictive transfers to avoid premature merge
    - Aggregated data movements

ADAPT's Adaptation

- Previously
  - Partitions of two areas were fixed;
  - How to identify a request to be random or sequential was not adaptive.
- ADAPT
  - dynamically adjusts two areas online;
  - identifies requests in an adaptive way.
- Workloads are dynamic.

ADAPT's Adaptation

- Key Idea:

If performance suffers from insufficient random log

blocks, use blocks from sequential area,

and vice versa.

ADAPT's Adaptation

- Two variables to detect performance:
  - $\delta = \frac{count\ of\ switch\ and\ partial\ merge}{count\ of\ sequential\ \log\ block\ allocation} \in [0, 1];$
  - $\varphi = \frac{count\ of\ merged\ pages\ in\ full\ merges}{count\ of\ full\ merge} \leq Block\ size;$
  - δ: sequential area; φ: random area.

ADAPT's Adaptation

- How to use δ and φ
  - In an interval, δ and φ are measured;
  - If $\delta > 0.4$, to enlarge sequential area;
  - Else if $\varphi \geq \frac{Block\ Size}{2}$, to enlarge random area.
- Why?
  - Larger $\delta \rightarrow$ a higher hit rate in sequential area;
  - Larger $\varphi \rightarrow$ full merge to process more valid pages;
  - Enlarging sequential area has a higher priority: switch/partial merge is less expensive.

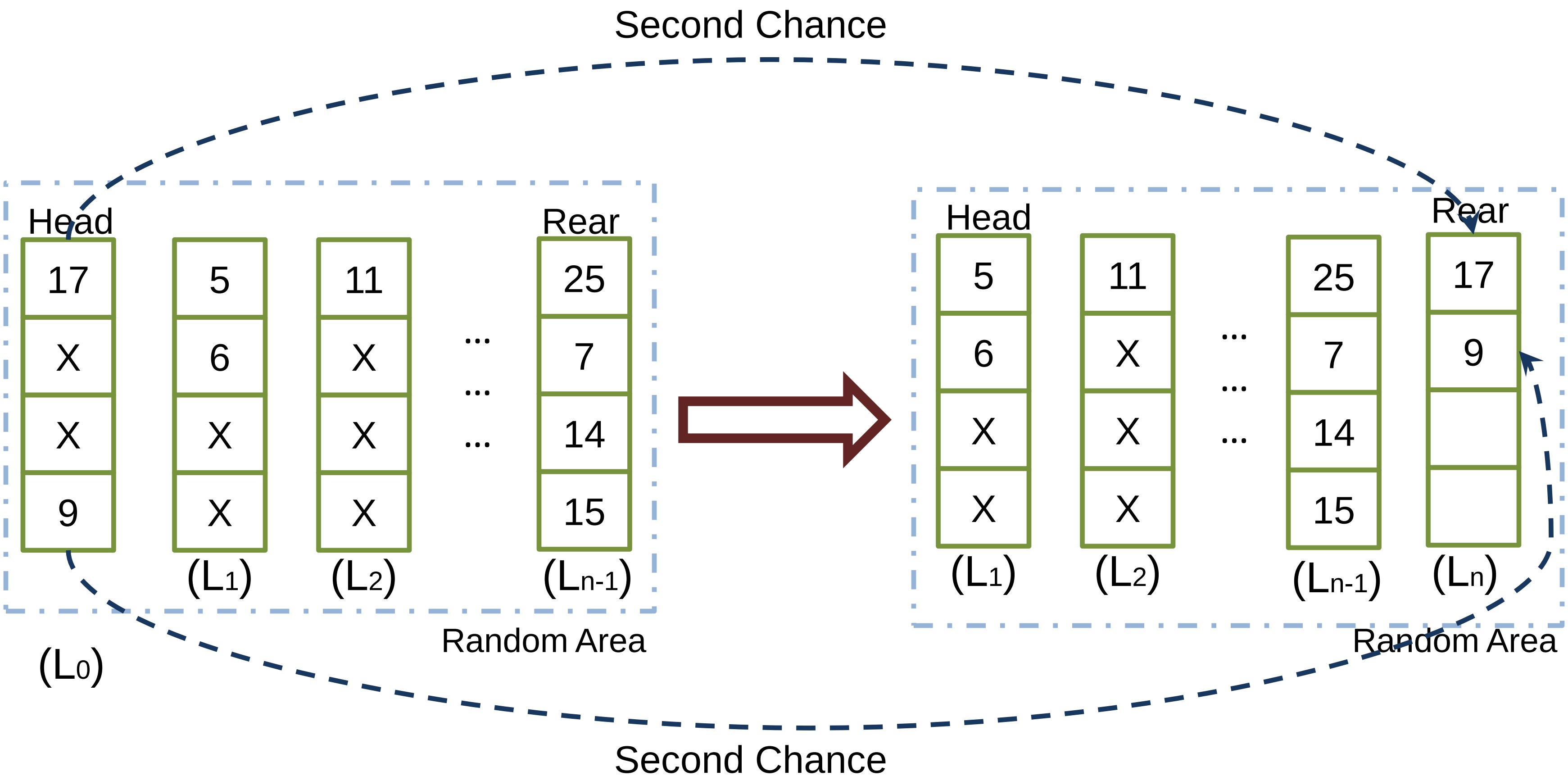
NUS National University of Singapore

ADAPT’s Adaptation

- Also adapts threshold for directing a request to sequential or random area:
  - Observation: over a long period, sequential requests tend to access a similar number of pages;
  - In the recent interval, a very small $\delta \Rightarrow$ sequential area was not very effective;
  - ADAPT adjusts the threshold accordingly.

ADAPT's Predictive Transfer

- FASTer's *second chance scheme*



ADAPT's Predictive Transfer

- FASTer's *second chance scheme*
  - A page of valid data remains in log space;
  - At least one merge is avoided if the page is accessed soon;
  - If not, such movement would be wasteful.

- Why not predict a page's update likelihood?
  - Positive: move it;
  - Negative: merge it.

Merge-or-move decision making

NUS National University of Singapore

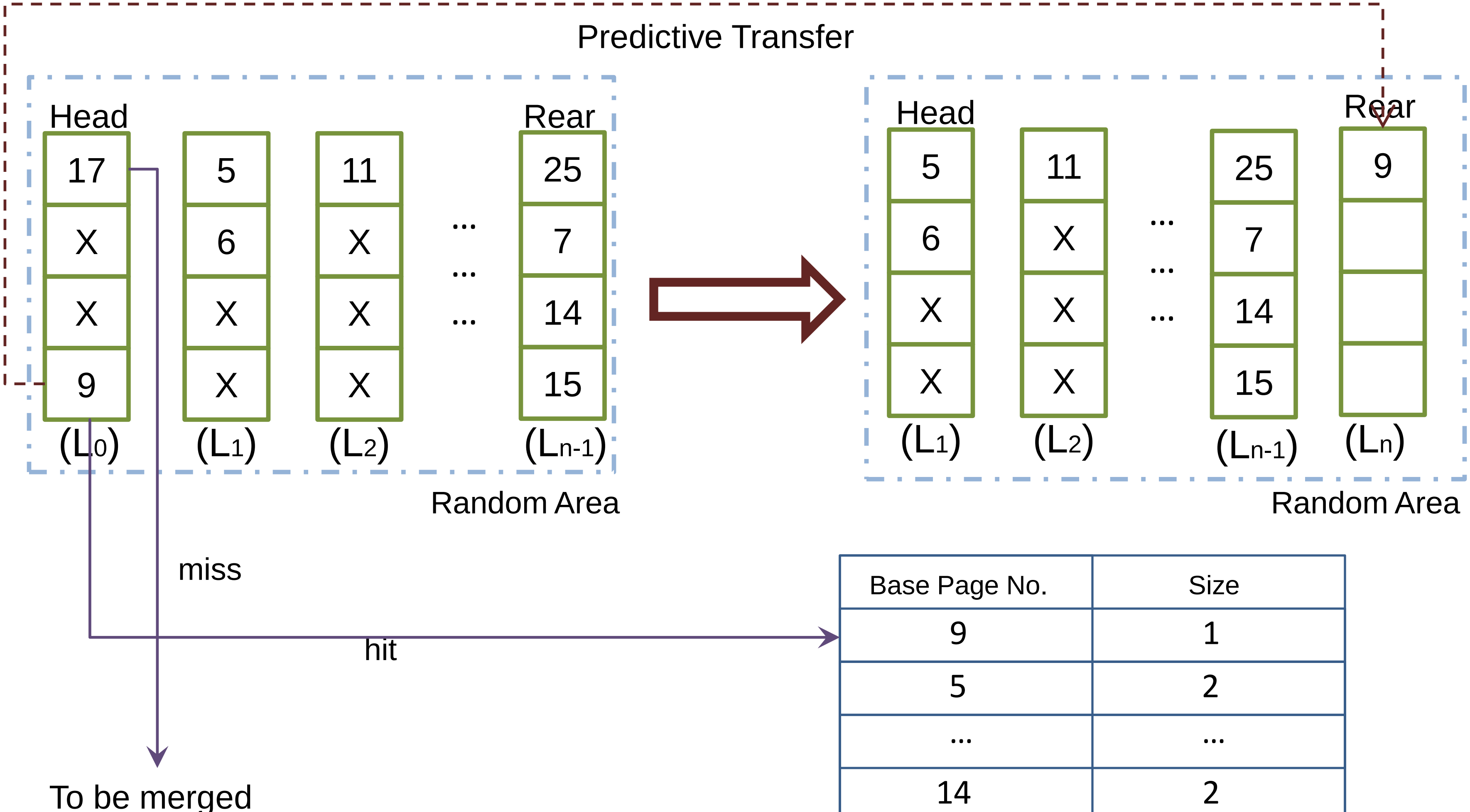
ADAPT's Predictive Transfer

- How to do prediction
  - ➢ Temporal locality:

    A recently-updated page is likely to be written to again.

- HAT: historical access table
  - ➢ Records a history of recent writes to logical pages;
  - ➢ Managed in LRU with fixed space.

ADAPT's Predictive Transfer



Base Page No.	Size
9	1
5	2
...	...
14	2

Historical Access Table (HAT)

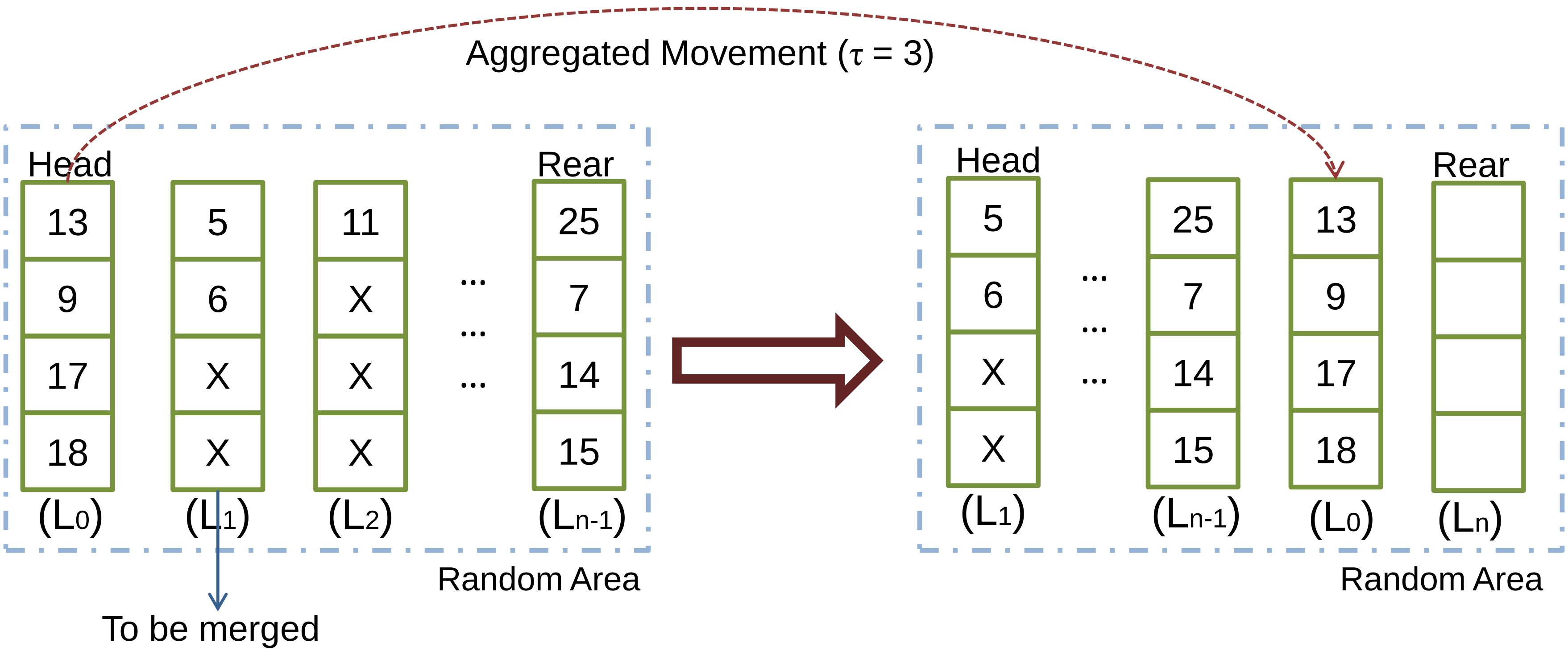
ADAPT’s Aggregated Movement

- Observation:
  - Non-OLTP workloads usually have big and sequential write requests;
  - Many log pages in the victim block to be merged are valid;
  - Inefficient to process one by one.

- ADAPT employs aggregated movement to give a second chance to a whole block.

NUS National University of Singapore

ADAPT's Aggregated Movement



- τ: aggregated movement threshold;
- Upon a merge, L_0 and L_1 are checked;
- If L_0 has more valid pages than τ and L_1 does not, move L_0 and merge L_1.

NUS National University of Singapore

Outline

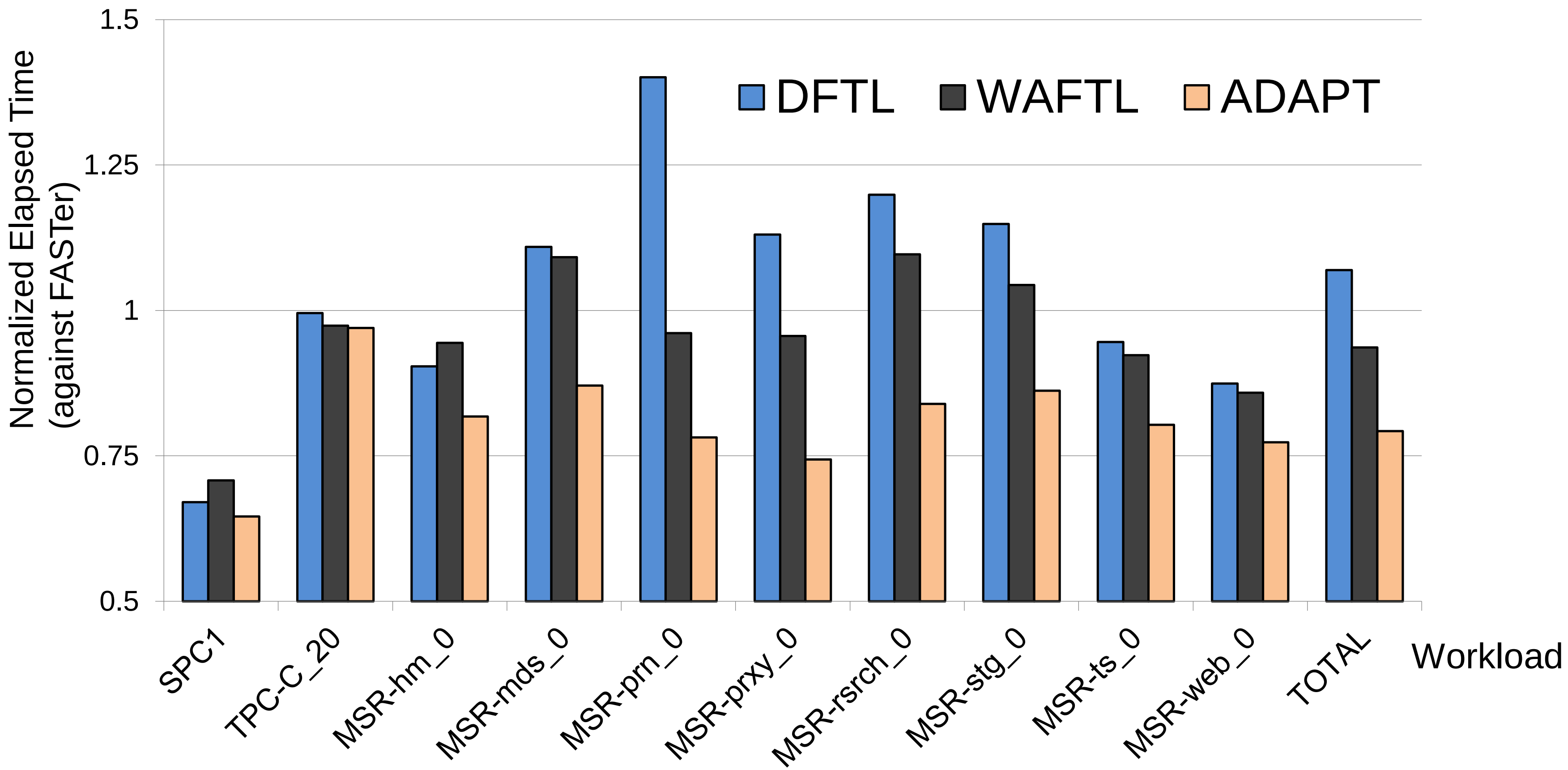
- Introduction
  - NAND Flash Memory
  - FTL and workload
- Background of hybrid mapping
- ADAPT
  - Adaptive partitioning of log space
  - Prediction and Aggregation
- Evaluation
- Conclusion

Evaluation

- Simulation setup
  - FlashSim simulator with GCC-4.6;
  - Ten public workloads;
  - DFTL (ASPLOS 2009), FASTer (SNAPI 2010) and WAFTL (MSST 2011) were implemented for comparisons;
  - The main metric is the elapsed time to finish each workload;
  - The default value of τ is 56; the length of the interval to measure δ and φ is 4000 requests.

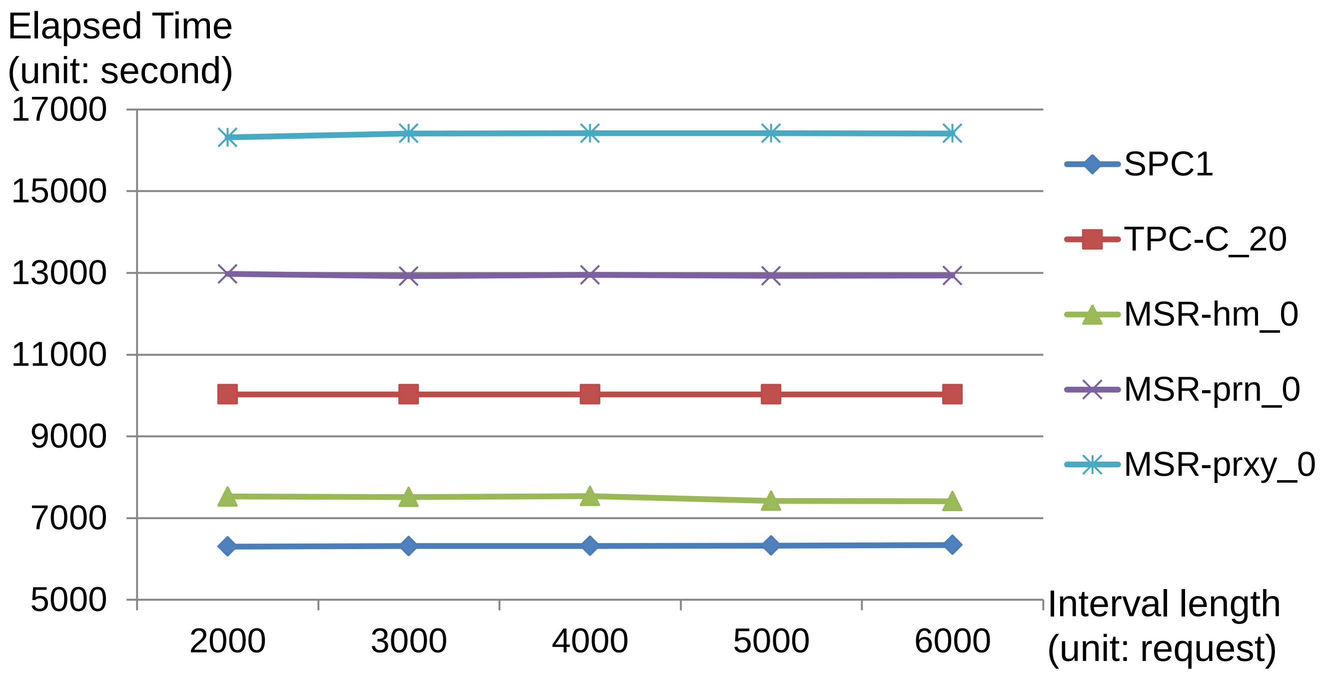
Experimental Results

- Elapsed time (performance)



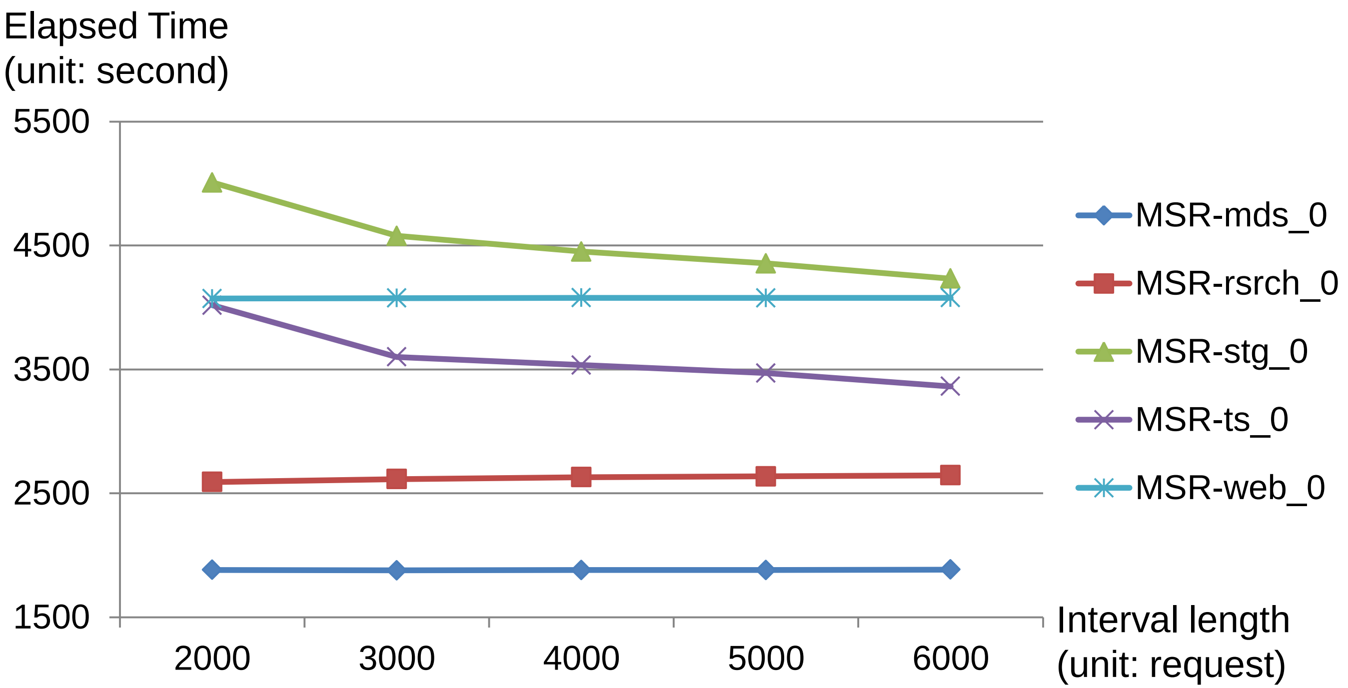
Experimental Results

- Impact of interval length on φ and δ (A)



Experimental Results

- Impact of interval length on φ and δ (B)



Experimental Results

- Prediction hit rates and aggregated moves

Workload	Prediction Hit Rate	Aggregated Movements
TPC-C_20	100.00%	0
SPC1	79.50%	132
MSR-hm_0	95.68%	233561
MSR-mds_0	96.49%	1727
MSR-prn_0	99.93%	124607
MSR-prxy_0	99.72%	8323
MSR-rsrch_0	98.75%	2050
MSR-stg_0	93.24%	1045
MSR-ts_0	95.16%	1165
MSR-web_0	96.99%	5408

Outline

- Introduction
  - NAND Flash Memory
  - FTL and workload
- Background of hybrid mapping
- ADAPT
  - Adaptive partitioning of log space
  - Prediction and Aggregation
- Evaluation
- Conclusion

Conclusion

- ADAPT
  - Fully-associative hybrid mapping scheme
  - Employs
    - adaptive partitioning of log space
    - predictive transfer
    - aggregated movements
- Simulation results show ADAPT can be faster than
  - DFTL by as much as 44.2%
  - WAFTL by as much as 23.5%

Thank you!

Questions?