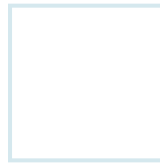
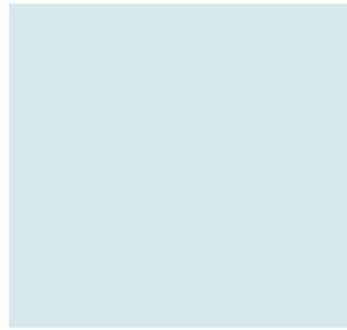


臺灣大學



A NOR Emulation Strategy over NAND Flash Memory

Jian-Hong Lin, Yuan-Hao Chang, Jen-Wei Hsieh, and Tei-Wei Kuo

Embedded Systems and Wireless Networking Laboratory
Dept. of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan

臺灣大學



Agenda

- Research Motivation
- An Efficient Prediction Mechanism
- Performance Evaluation
- Conclusion



Comparison between NAND and NOR

	NOR	SLC NAND (large-block, 2KB-page)
Access Method	Random	Sequential
Access Speed	Read: 23.84 MB/sec Write: 0.07 MB/sec Erase: 0.22 MB/sec	Read: 15.33 MB/sec Write: 4.57 MB/sec Erase: 6.25 MB/sec
Density	Low	High
Price	High (34.55 \$/GB)	Low (6.79 \$/GB)
Application	Code storage •low-end mobile handsets •PC BIOS chips	Data storage •MP3 player (music storage) •Digital Cameras (image storage)



Motivation

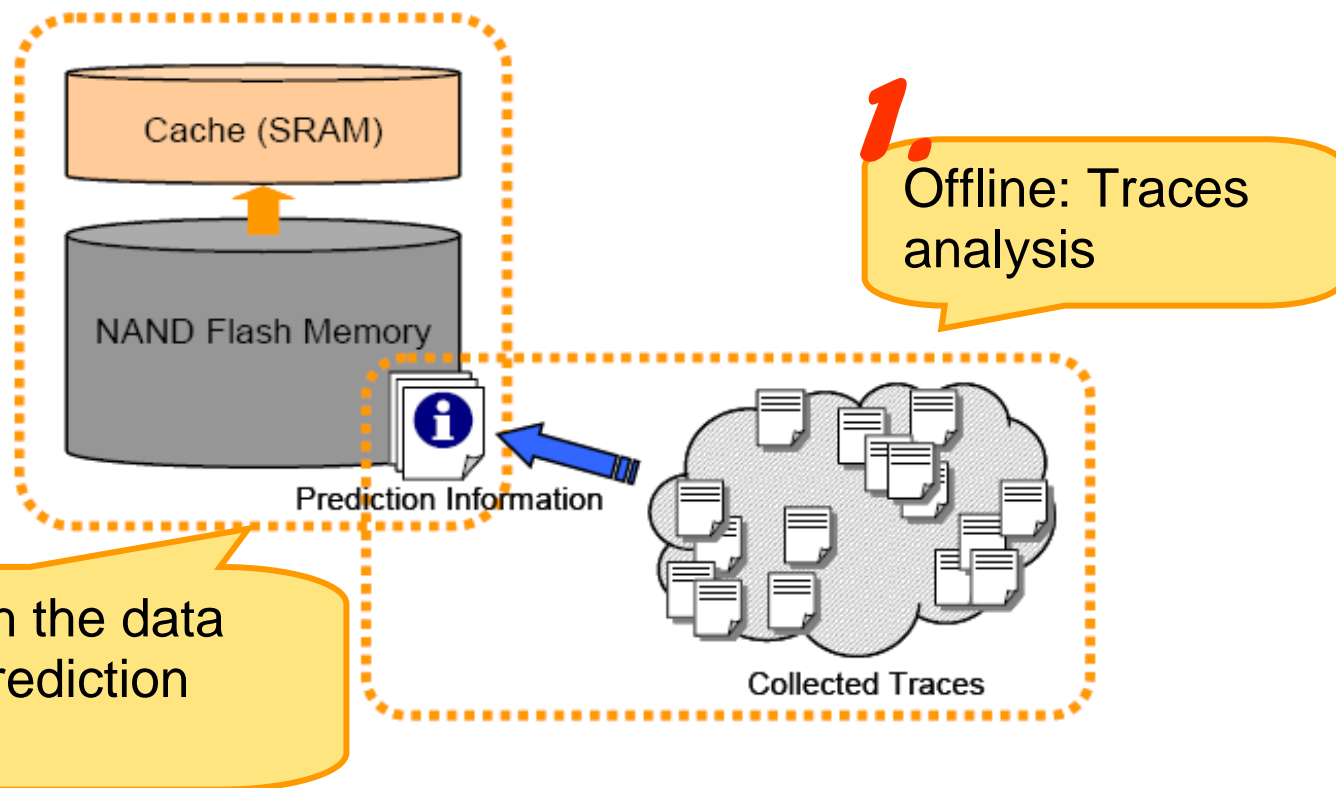
- Replace NOR with NAND for code storage
- Challenge:
 - How to fill up the performance gap between NAND and NOR?
 - Use SRAM for data caching
 - Prediction scheme
 - Implementation design





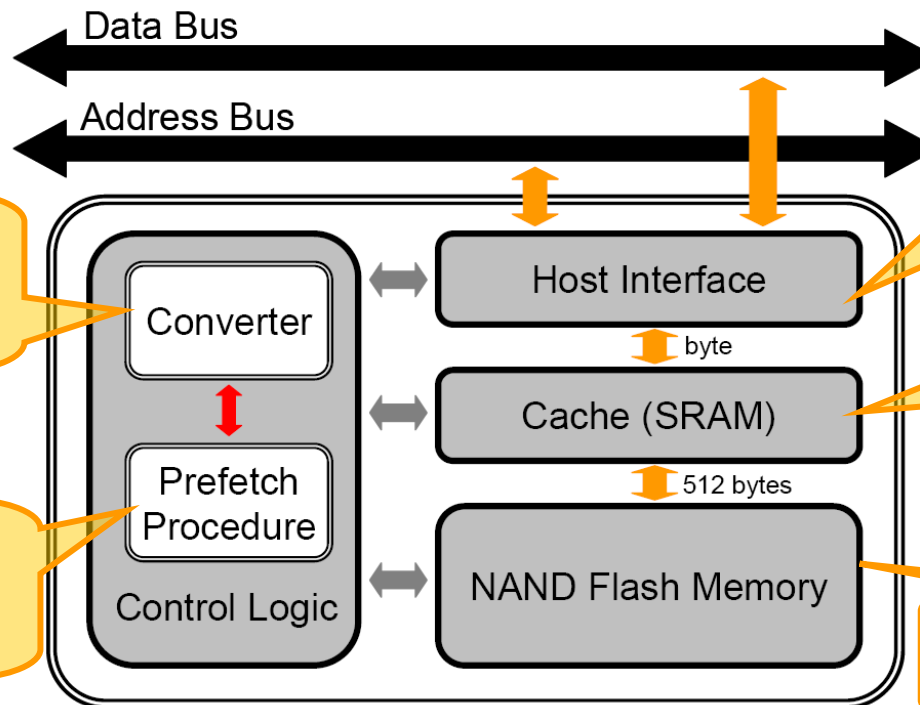
A Prediction-based Prefetching Strategy

- An Example Scenario





The System Architecture



Address translation from byte addressing to LBA addressing

Communication with the host system

Cache for data access

Prefetch data from NAND to SRAM based on prediction info.

For code and data storage

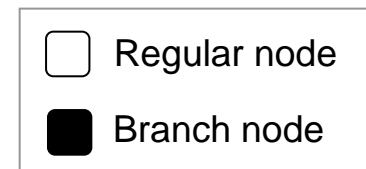
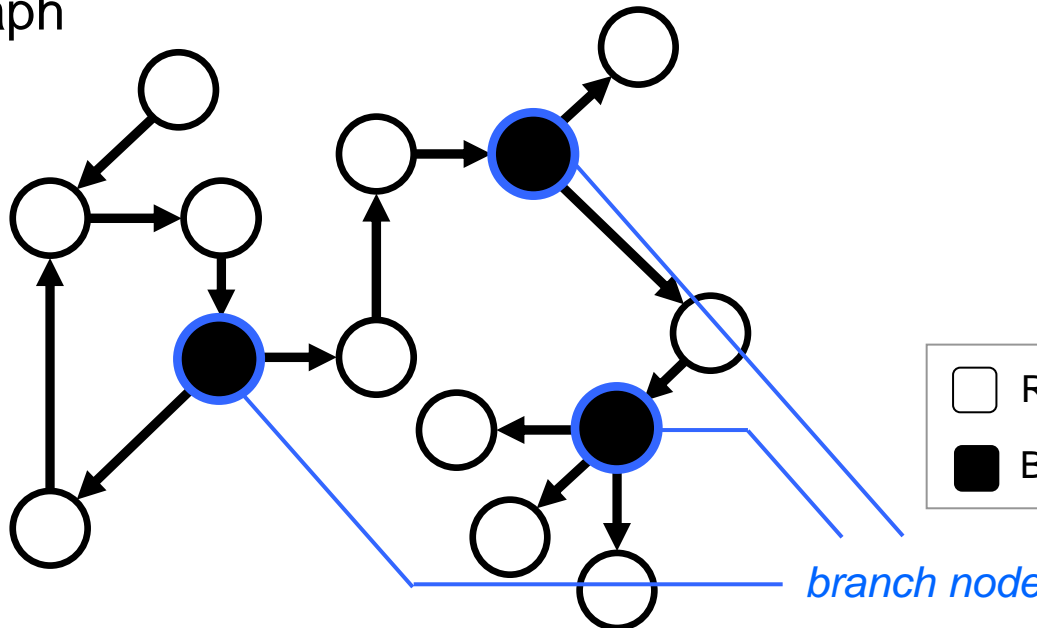
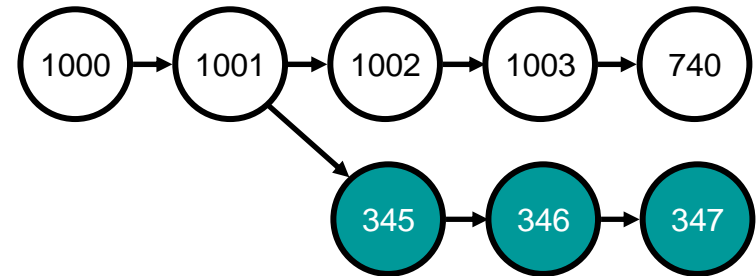


A Prediction Graph

- A prediction graph
 - An Illustration of the access patterns (a sequence of LBA's)
 - One node for each LBA in the graph

e.g.

seq i ... 1000, 1001, 1002, 1003, 740, ...
seq j ... 1000, 1001, 345, 346, 347, ...



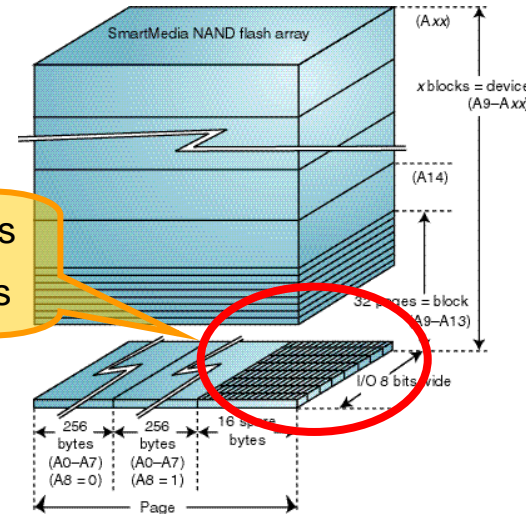


An Implementation

- The Way to Save the Prediction Graph in the Flash Memory

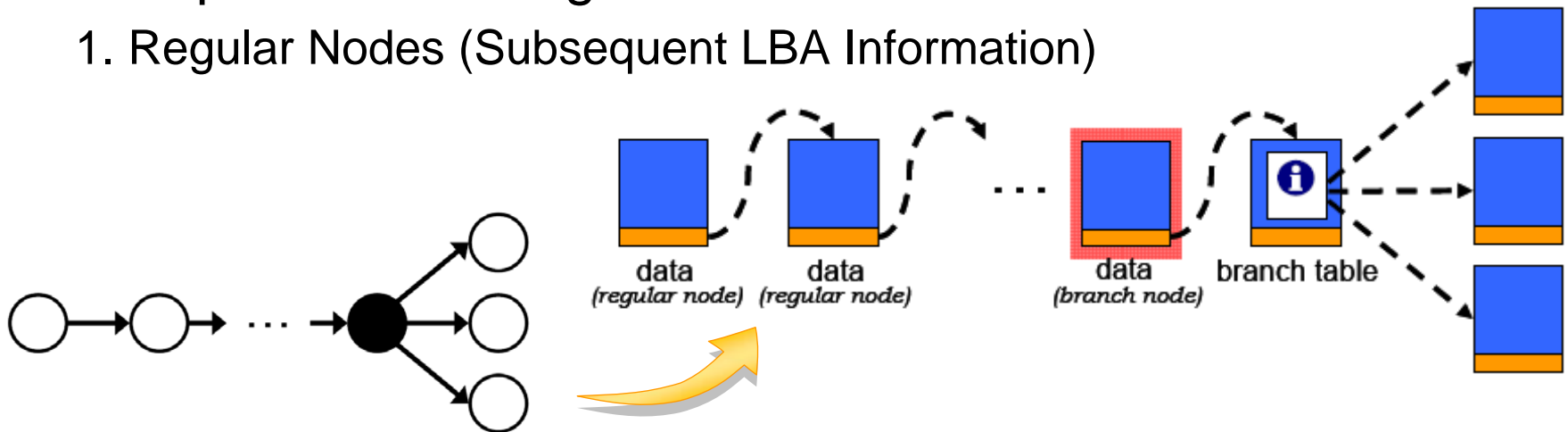


small block: 16 bytes
large block: 64 bytes



- The Spare Area Usage of Nodes

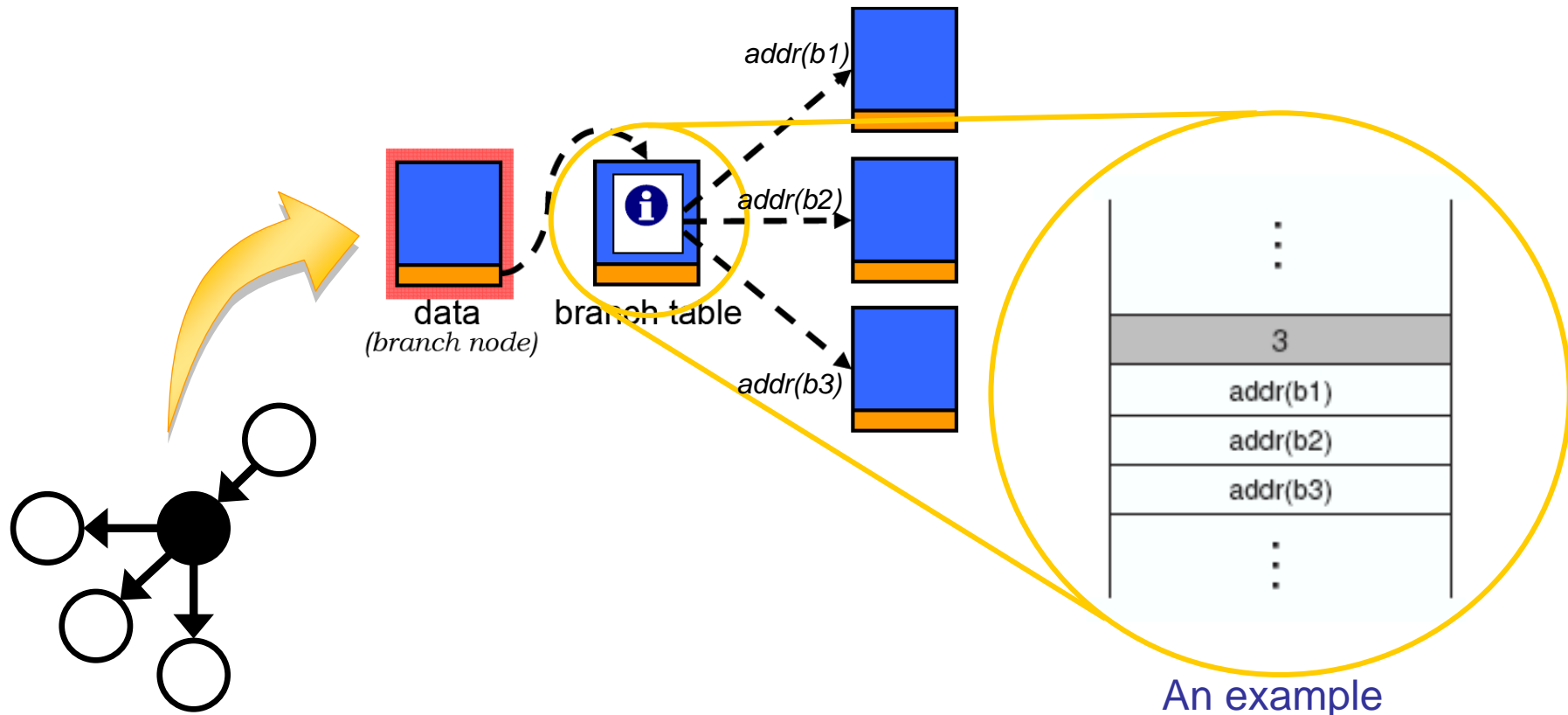
1. Regular Nodes (Subsequent LBA Information)





An Implementation (cont.)

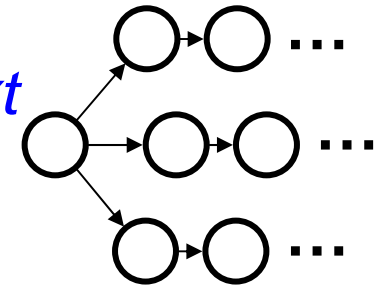
2. Branch Nodes (Assisted by a Branch Table)



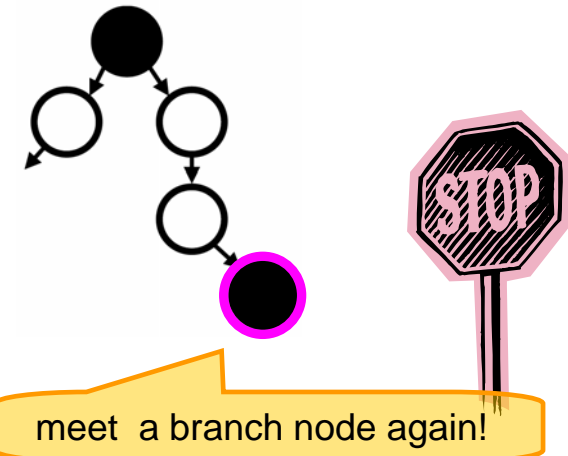


A Prefetch Procedure

- The Objective: The Probability Maximization of Data Accesses over SRAM
- Cyclic Buffer with Two indices: *current* and *next*
- A Greedy Algorithm in the Prefetch Procedure
 - Regular Node → Prefetching of its Subsequent LBA
 - Branch Node → Prefetching of all Possible Following LBA Links in a Round-Robin Way



- Stop Conditions
 1. *next* reaches a branch node again along a link.
 2. *next* and *current* point to the same page.
 3. The caching buffer is full.

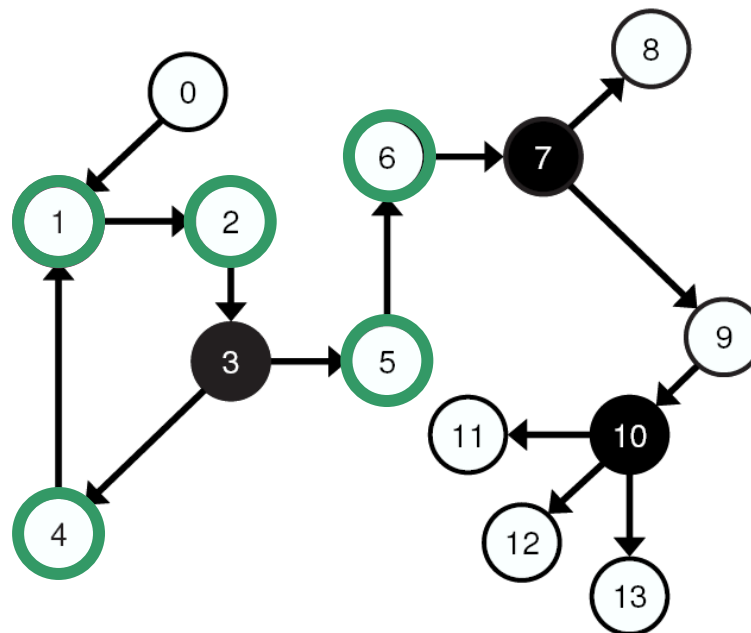




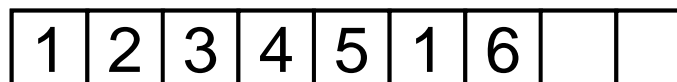
A Prefetch Procedure

An Example

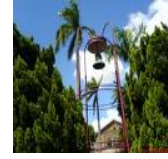
Prediction graph



the cache



current next



Snapshots of the Access Patterns

aoc_10_Trace.txt - ...

檔案(F) 編輯(E) 格式(O) 檢視(V)
說明(H)

R	37263511	8
R	6787071	8
R	37263519	8
R	6788959	8
R	37263527	8
R	6791743	8
R	37263535	8
R	6793727	8
R	37263543	8
R	37263551	8
R	6795807	8
R	37263559	8
R	37263567	8
R	37263575	8
R	6796447	8
R	37263583	8
R	37263591	8
R	37263599	8
R	37263607	8
R	6798303	8
R	37263615	8
R	37263623	8
R	37263631	8
R	37263639	8

RANDOM ACCESS!

AOE II

TTD_10_Trace.txt - ...

檔案(F) 編輯(E) 格式(O) 檢視(V)
說明(H)

R	5839775	128
R	5839903	128
R	5840031	128
R	5840159	128
R	5840287	128
R	5840415	128
R	5840543	128
R	5840671	128
R	5840799	128
R	5840927	128
R	5841055	128
R	5841183	128
R	5841311	128
R	5841439	40
R	5326111	8
R	5326615	7
R	5326599	8
R	5326607	8
R	5326119	8
R	5326231	8
R	5326127	104
R	5326239	128
R	5326367	128
R	5326495	104

TTTD

rd_10_Trace.txt - 記...

檔案(F) 編輯(E) 格式(O) 檢視(V)
說明(H)

R	36798207	128
R	36798335	128
R	36798463	128
R	36798591	128
R	36798719	128
R	36798847	128
R	36798975	128
R	36799103	128
R	36799231	128
R	36799359	128
R	36799487	128
R	36799615	128
R	36799743	128
R	36799871	128
R	36799999	128
R	36800127	128
R	36800255	128
R	36800383	128
R	36800511	128
R	36800639	128
R	36800767	128
R	36800895	128
R	36801023	128
R	36801151	128

BURST READ!

Raiden

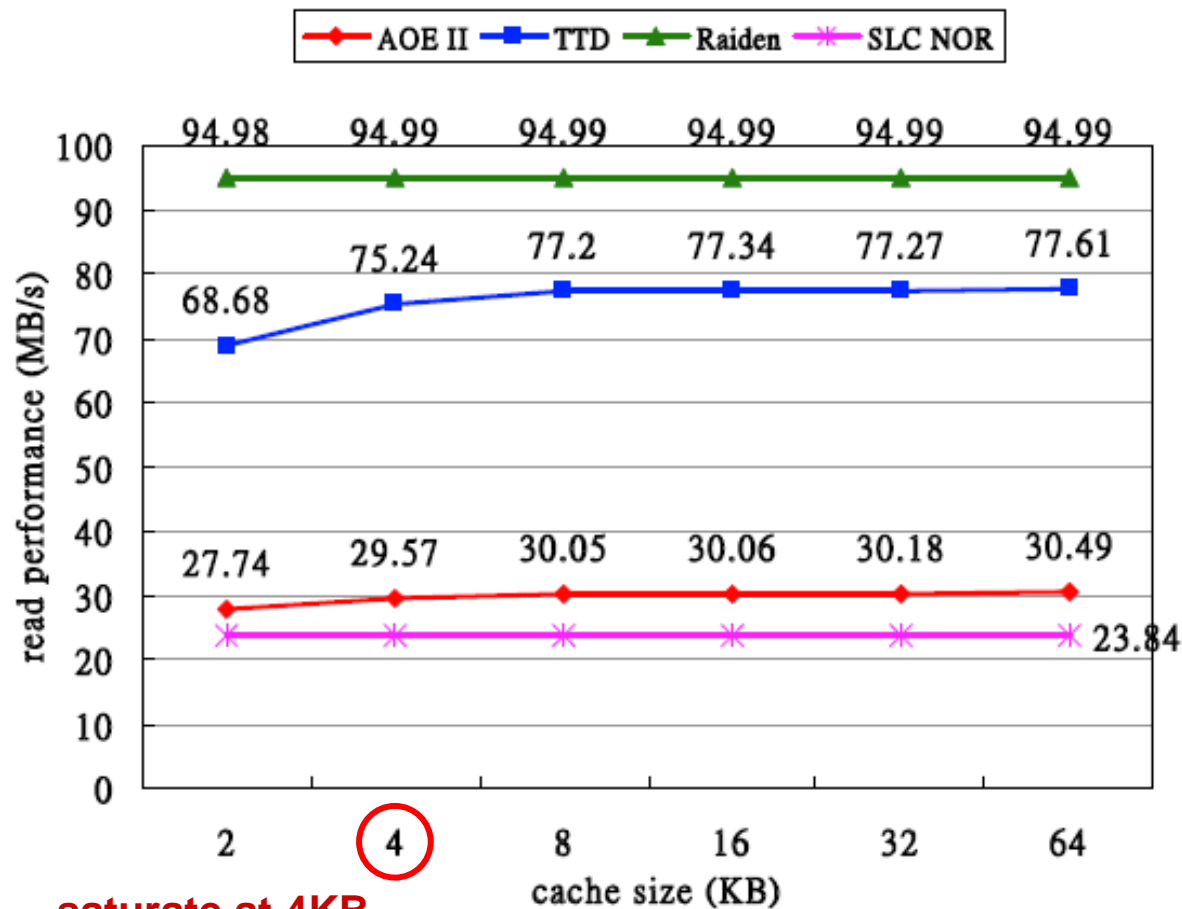


Performance Metrics and Experiment Setup

- Performance metrics
 - Read performance
 - Cache miss rate
 - Main-memory requirement
- Experiment setup
 - Large-block NAND flash memory:
 - Setup time (random access): 25 us/page
 - Serial access time: 50 ns/byte
 - SRAM
 - 10 ns/byte
 - NOR flash memory
 - 40 ns/byte




Read Performance





Conclusion

- An Application-Oriented Approach to Replace NOR with NAND
 - Prefetching of data from NAND based on the trace analysis
 - Limited SRAM requirement
 - Good performance, but the results depending on the predictability of the applications
- Performance Improvement and Overhead Evaluation
 - Read performance better than that of NOR:
 - AOE II: 24%
 - TTD: 216%
 - Raiden: 298%
 - Cache miss rate:
 - Lower than 10% in most cases