

# Flash memory: Models and Algorithms

Guest Lecturer: Deepak Ajwani

# Flash Memory



# Flash Memory



# Flash Memory



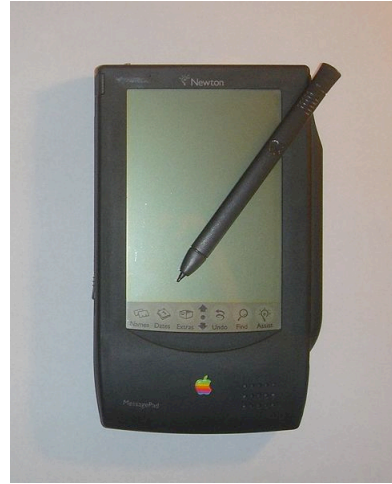
# Flash Memory



# Flash Memory



# Flash Memory



# Flash Memory





# Popularity



- Many recent models of notebook already use SSDs - Sony Vaio UX90, Apple MacBook Air, Lenovo ThinkPad X300, Dell latitude D430, Samsung Q30-SSD
- It is predicted that 50% of all mobile computers would use flash (instead of hard disks) by 2013
- Prices are coming down (as predicted): 120 GB for less than 250 USD

## So, why is it becoming popular?

- Lighter devices
- Smaller sizes (A 1.5 sq.cm. x 1 mm microSD card can have capacity of up to 16 GB)
- More shock resistant
- Can withstand intense pressure, extremes of temperature, and even immersion in water
- Consume less power

## Compared to DRAM

- Much cheaper than RAM
- Non-volatile

## Compared to Hard-disk



Traditional hard disk drive



Solid state hard drive

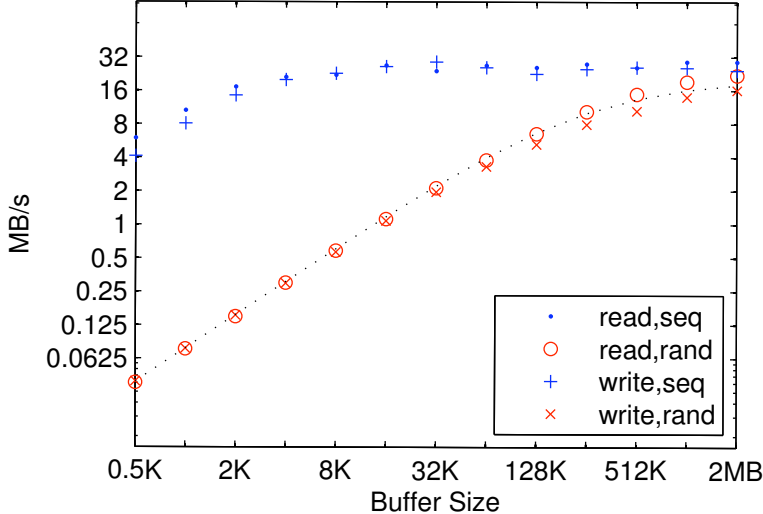
- Finally, getting rid of mechanics!

## DRAM, SSD and HDD

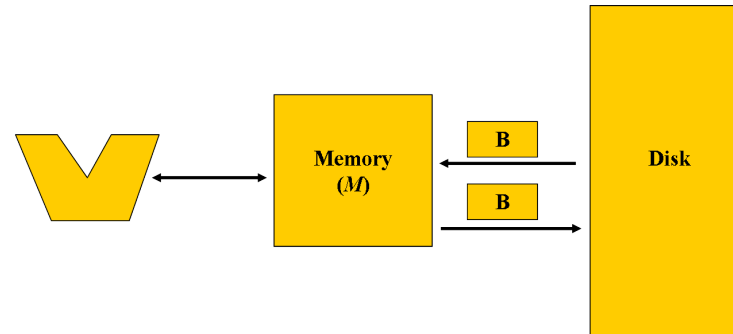
Characteristics	DRAM	SSD	HDD
Volatile	Yes	No	No
Shock Resistant	Yes	Yes	No
Physical Size	Small	Small	Large
Storage Capacity	Small	Large	Largest
Energy Consumption	-	Medium	High
Price	Very high	Medium	Very cheap

# HDD Characteristics

Performance Summary (log scale)



## I/O Model



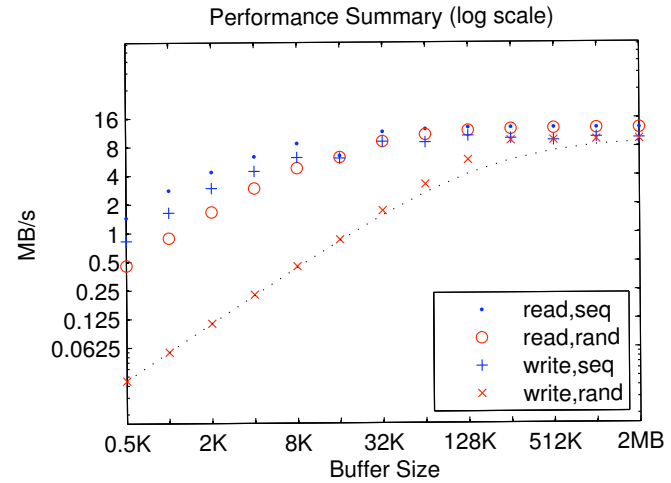
- Two level memory hierarchy
- Faster level has limited capacity  $M$
- In one I/O,  $B$  contiguous elements transfer between the two memories
- Performance measure:  $r + w$

## Flash Memory: Blocks and Pages



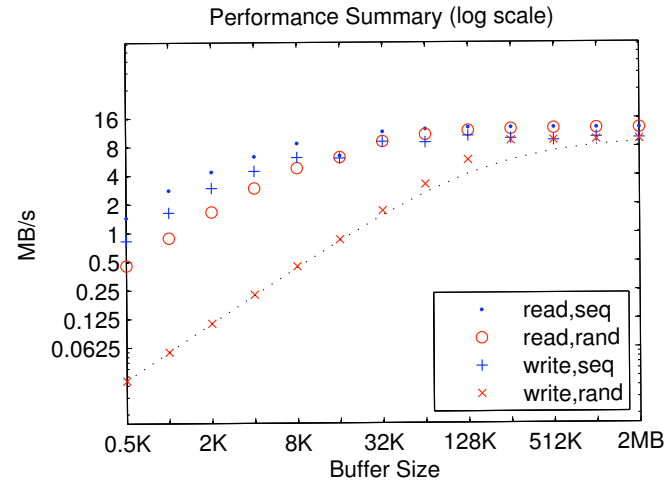
- Memory cells divided into blocks of size 16 KiB to 512 KiB
- Blocks are divided into pages of size 512 Byte to 4 KiB
- Reading:
  - One page at a time
- Writing:
  - One page at a time
  - Resetting a bit to 1 requires erasing the entire block
- Erase:
  - One **block** at a time
  - An expensive operation

## SSD Characteristics



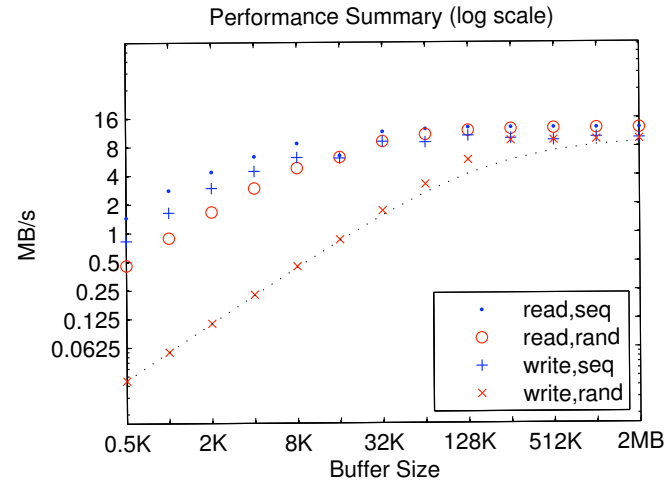
- Writing small data costs more than reading the same amount of data

## SSD Characteristics

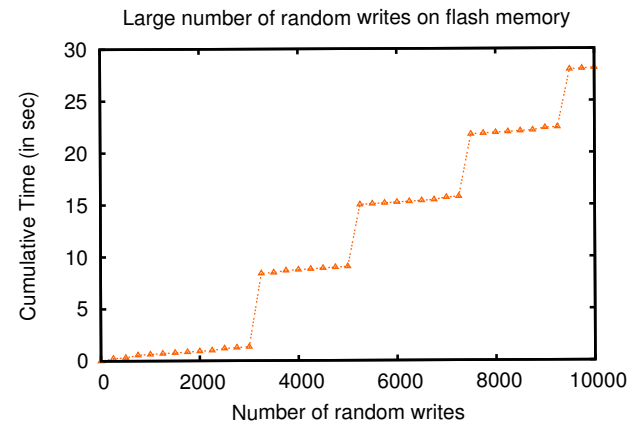


- Sequential writes more expensive than sequential reads

# SSD Characteristics



## Secondary SSD Characteristics - Writes



- Burst of random writes slow down subsequent writes
- **Non-uniform write cost:** Cost of writes depends on past history of the device

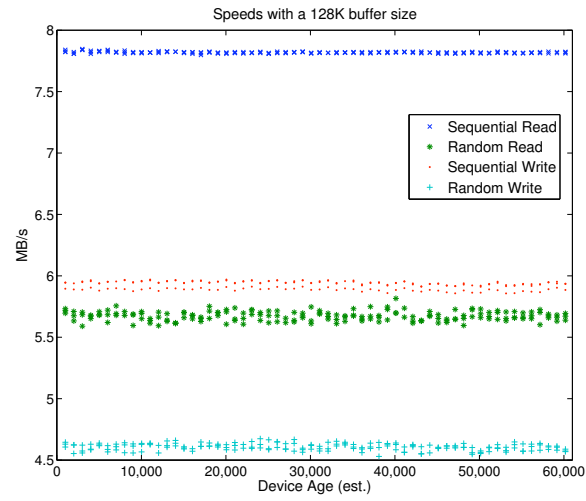
## Other Secondary SSD Characteristics

- **Burst of random writes** can slow down subsequent sequential writes as well, with effects lasting a minute or more
- No such effects on subsequent reads
- **Misaligned random writes** slower than aligned random writes
- No such difference observed for random reads

## Wear Levelling

- **Limited endurance:** Blocks wear out after a finite number (100,000 to 1,000,000) of erase-write cycles
- **Wear levelling:** Dynamically re-mapping the blocks to distribute the write and erase operations evenly across each memory block of the device
- Wear levelling techniques implemented in hardware by built-in micro-controllers
- Controllers forbid direct access to the memory cells
- Error correcting code, Bad block management to extend the life of the device
- One has to write for 30+ years continuously to wear-out a block of a 16 GB flash device

# Secondary SSD Characteristics - Aging



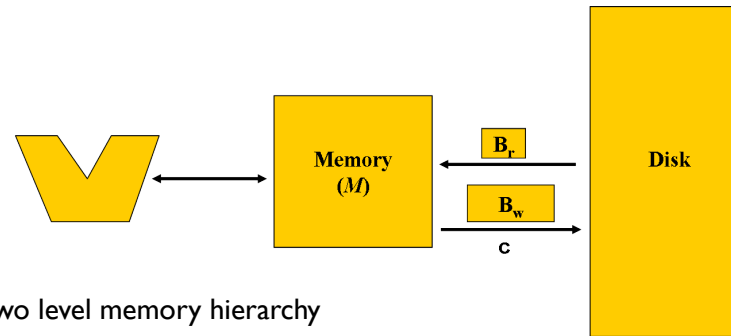
## Summary/Conclusion

- Flash devices fast becoming popular storage medium
- Read/write characteristics of flash devices very different from the hard-disk
- Need different computation model for making best use of the flash devices

## Key SSD Characteristics - Reminder

- Writing small data costs more than reading the same amount of data
- Sequential writes more expensive than sequential reads
- Read Block size  $B_r$  smaller than write Block size  $B_w$

## General Flash Model



- Two level memory hierarchy
- Faster internal memory has limited capacity  $M$
- In one read I/O,  $B_r$  contiguous elements copied from external to internal memory
- In one write I/O,  $B_w$  contiguous elements copied from internal to external memory
- Performance measure:  $r + c w$

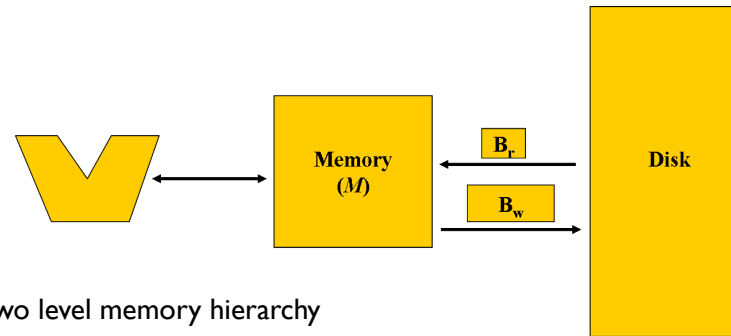
## The write penalty - c

- c is proportional to the ratio between the sequential read throughput and the sequential write throughput
- $c = \frac{B_r}{B_w}$ : I/O model
- c close to 0: Write I/Os are essentially free
- c very high: Design algorithms that only write the final output
- With c varying, one can get trade-offs between read and write I/Os
- Analyzing this trade-off is often quite difficult
- **Focus:** Design and analyze meaningful algorithms

## Sorting in general flash model

- Consider the I/O model ( $B_r = B_w$ )
- If only  $n/B$  write I/Os are allowed, we need  $\Theta(n^2/(M \cdot B))$  I/Os
- If  $k \cdot n/B$  write I/Os are allowed,
  - **Upper bound:**  $O\left(\min\left\{\frac{n^2}{B \cdot M \cdot (M/B)^k} + \frac{k \cdot n}{B}, \frac{n^2}{BM^{k+1}} + k \cdot n\right\}\right)$  read I/Os
  - **Open Problem:** Lower bound for sorting when only  $k \cdot n/B$  write I/Os are allowed

## Unit-Cost Flash Model



- Two level memory hierarchy
- Faster internal memory has limited capacity  $M$
- In one read I/O,  $B_r$  contiguous elements copied from external to internal memory
- In one write I/O,  $B_w$  contiguous elements copied from internal to external memory
- Performance measure:  $B_r r + B_w w$

## Understanding Unit-Cost Flash model

- **Fundamental assumption:** Sequential read throughput is equal to the sequential write throughput, i.e.,  $c = B_w/B_r$
- True for many flash devices. For others, it is a close (less than 1.5) approximation

## Varying granularity of read/write I/Os

- $B_r = B_w$ : I/O Model
- $B_r = 1$ : Full support for random reads, just as it is assumed for DRAM
- In general,  $1 \ll B_r < B_w \ll M \ll n$

## Relation to I/O Model

- An EM algorithm of I/O complexity  $O(f(n))$  I/Os with block size  $B_w$  requires  $O(f(n) \cdot B_w)$  elements transferred in the unit-cost flash model
- An EM algorithm using  $O(f(n))$  read I/Os with block size  $B_w$  requires  $O(f(n) \cdot B_w)$  elements transferred in the unit-cost flash model
- An EM algorithm using  $O(f(n))$  write I/Os with block size  $B_w$  requires  $O(f(n) \cdot B_w)$  elements transferred in the unit-cost flash model
- Lower bound of  $\Omega(g(n, B_r))$  I/Os in the EM model implies a lower bound of  $\Omega(g(n, B_r) \cdot B_r)$  elements

## Unit-Cost Flash Model

- Read-only or scan-only algorithms efficient:
  - Read scan:  $\Theta(n)$
  - Write scan:  $\Theta(n)$
  - Searching:  $O(B_r \log_{B_r} n)$

## Unit-Cost Flash Model: Sorting

- Split the sequence into  $\Theta(M/B_r)$  subsequences
- Recursively sort the subsequences
- Merge the subsequence streams keeping a read block of data from each subsequence in internal memory
- $\Theta(n \cdot \log_{M/B_r} n)$  elements transferred

## Data Structures

- Dynamic B-trees
  - Primary data structure: B-tree with fan-out  $\Theta(B_w)$
  - Each node of the primary structure is a B-tree with fan-out  $\Theta(B_r)$
  - Search + Update:  $O(B_r \log_{B_r} n)$  elements transferred
- Priority Queue
  - $O(\log_{M/B_r} n/B_r)$  elements transferred

## Translation Layer

- Algorithms reads and writes pages of size  $B_r$
- TL obviously groups  $B_w/B_r$  pages into a block of size  $B_w$
- Maintains an LBA mapping
- Keeps track of free pages and blocks
- Keeping map internally - Simulation of EM algorithms on flash model

## Merging vs. Distribution Paradigm

- **Merging framework ideal:** Read small blocks from different streams, output bigger chunk --  $M/B_r$ -way merging possible
- **Distribution framework tricky:** Read a few blocks, collect it in RAM, write to different streams --  $M/B_w$ -way distribution
  - Non-trivial to transform distribution sweeping
- **Sorting:**  $\Theta(n \cdot \log_{M/B_r} n)$

## Summary: Unit-Cost Flash Model

- Scanning:  $\Theta(n)$
- Sorting:  $\Theta(n \cdot \log_{M/B_r} n)$
- B-Trees:  $O(B_r \cdot \log_{B_r} n)$
- Priority Queues:  $O(\log_{M/B_r} n / B_r)$
- Interesting new area with lots of open problems

## References

- E. Gal, S. Toledo: *Algorithms and data structures for flash memories*. ACM Computing Surveys: 37(2), pp. 138-163, 2005
- D. Ajwani, I. Malingier, U. Meyer, S. Toledo: *Characterizing the Performance of Flash Memory Storage Devices and Its Impact on Algorithm Design*. WEA 2008: 208-219
- D. Ajwani, A. Beckmann, R. Jacob, U. Meyer and G. Moruz: *On Computational Models for Flash Memory Devices*. SEA 2009