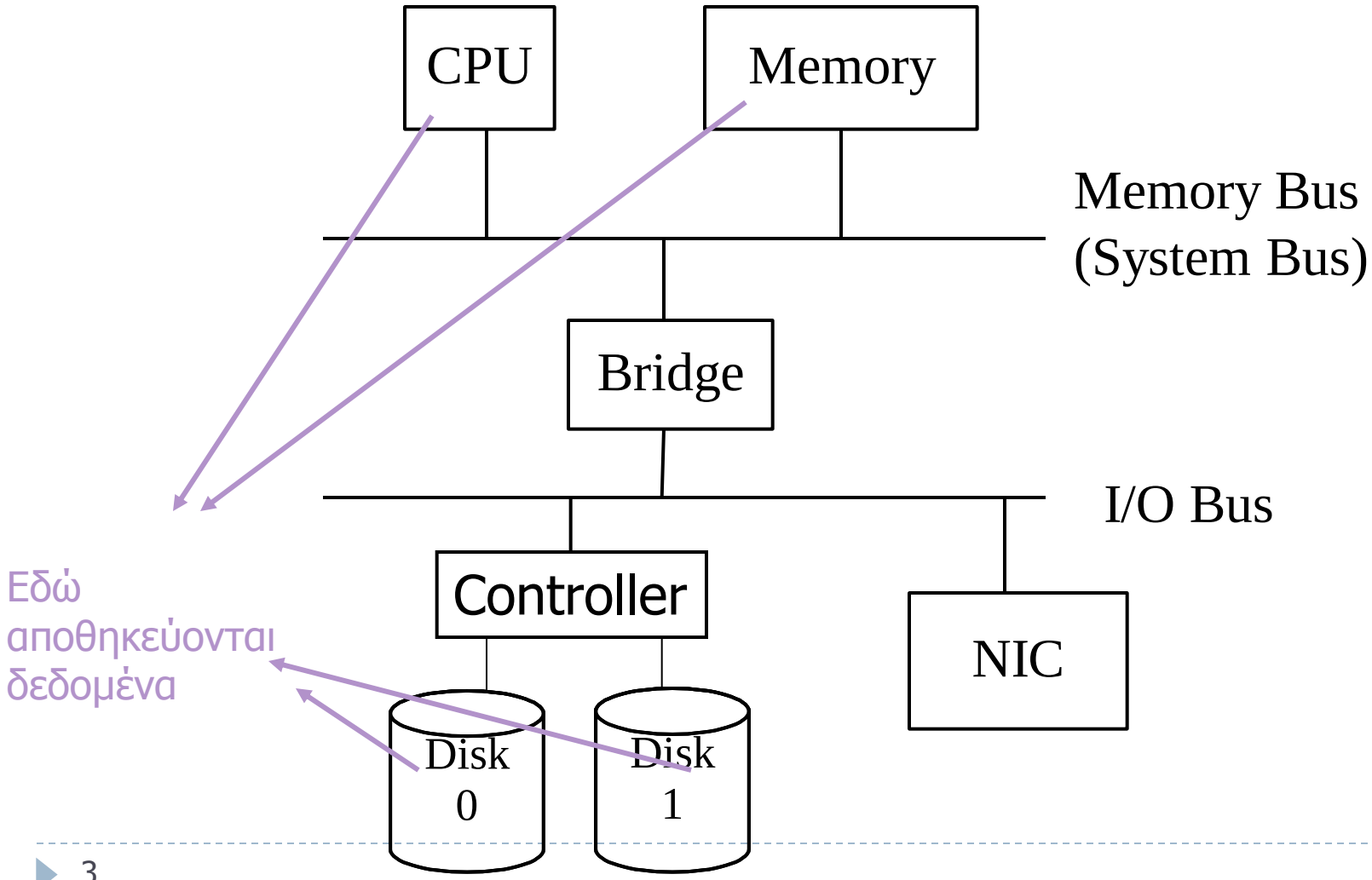
Μέσα Αποθήκευσης

Ιωάννης Κωτίδης

Περιγραφή

- Ιεραρχία Μνήμης
- Ο νόμος του Moore
- Δίσκοι, λειτουργία, παραδείγματα
- Βελτιστοποίηση
- SSDs

Δομή Υπολογιστή



Παράγοντες

- Ταχύτητα μέσου
- Κόστος αποθήκευσης (ενδεικτικές τιμές)
  - Μαγνητικοί δίσκοι: 15 λεπτά/GB
  - Solid State Disks (SSD): 50 λεπτά/GB
  - DDR3: 5 ευρώ/GB
- Αξιοπιστία
  - Απώλεια δεδομένων σε περίπτωση διακοπής παροχής τάσης
  - Απώλεια λόγω σφάλματος υλικού
    - Σημαντικός παράγοντας όταν χτίζουμε data centers

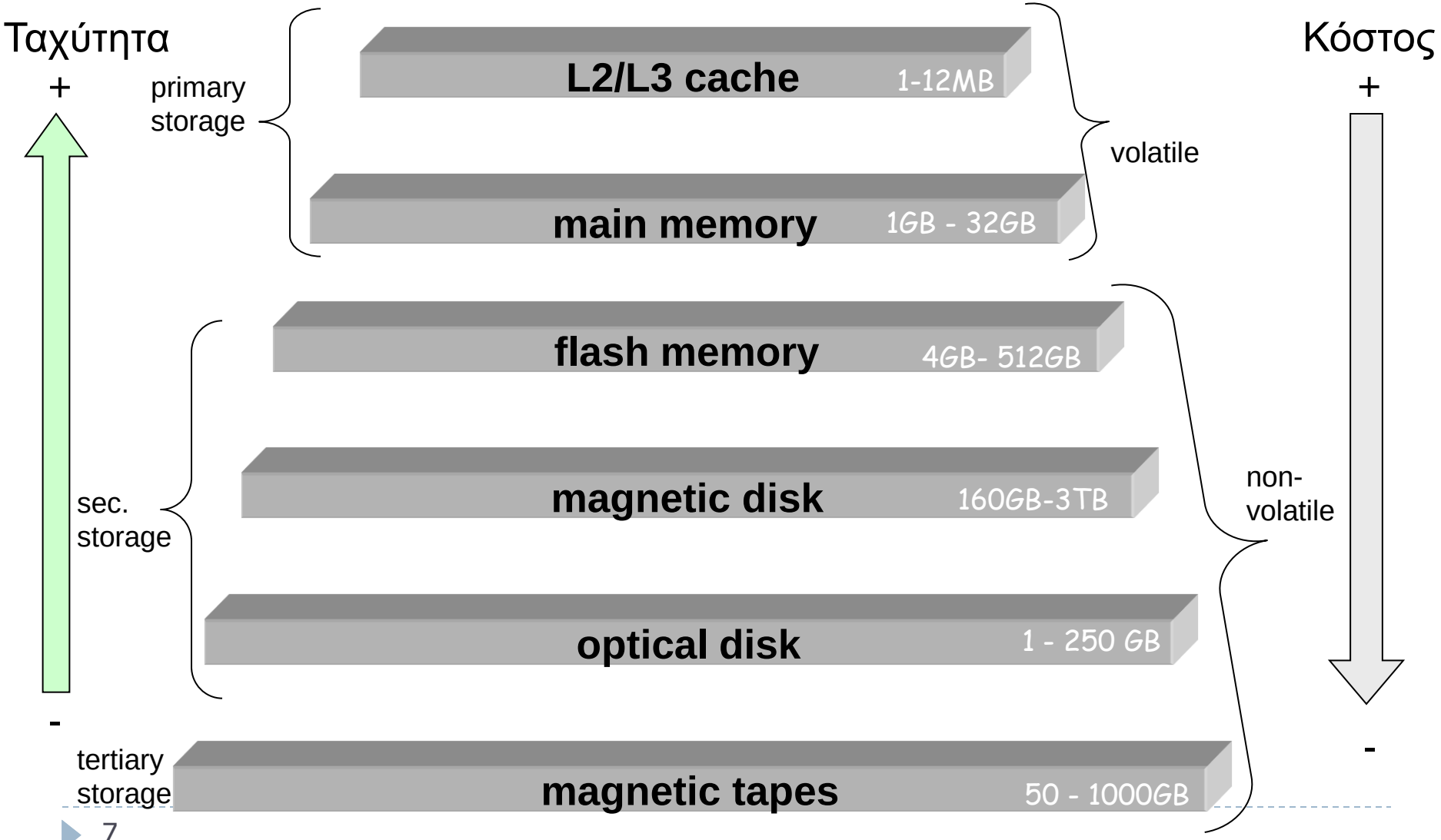
Ταξινόμηση Μνήμης

- Διαφορετικοί τρόποι κατηγοριοποίησης
  - Volatile (πρόσκαιρη) ←→ Non-volatile (μόνιμη)
- *Κύρια (Primary)*: fast, volatile
- *Δευτερεύουσα (secondary*): moderately fast, non-volatile, online
- *Ταινίες (Tertiary):* slow, non-volatile, off-line

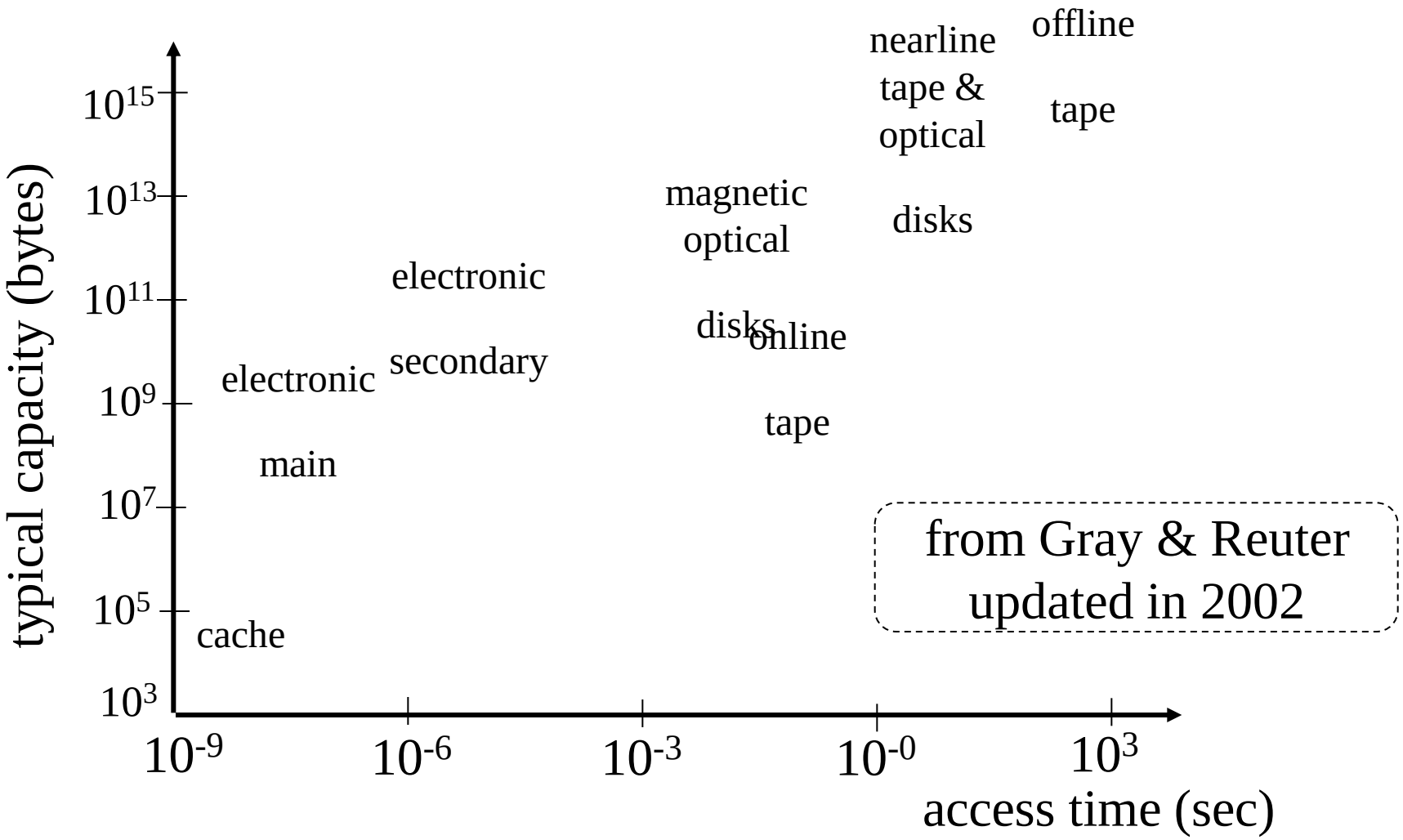
Volatile vs Non-Volatile

- Σημαντικό στοιχείο στη σχεδίαση ενός ΣΔΒΔ
  - *Επηρεάζει δύο από τα χαρακτηριστικά μίας δοσοληψίας*
    - *Atomicity, durability*
- Ακόμα και αν όλη η ΒΔ χωράει στη μνήμη, οι αλλαγές πρέπει να αποθηκευτούν σε κάποιο μόνιμο μέσο
  - Hard disk
  - RAM disks w/ battery
  - Flash memory

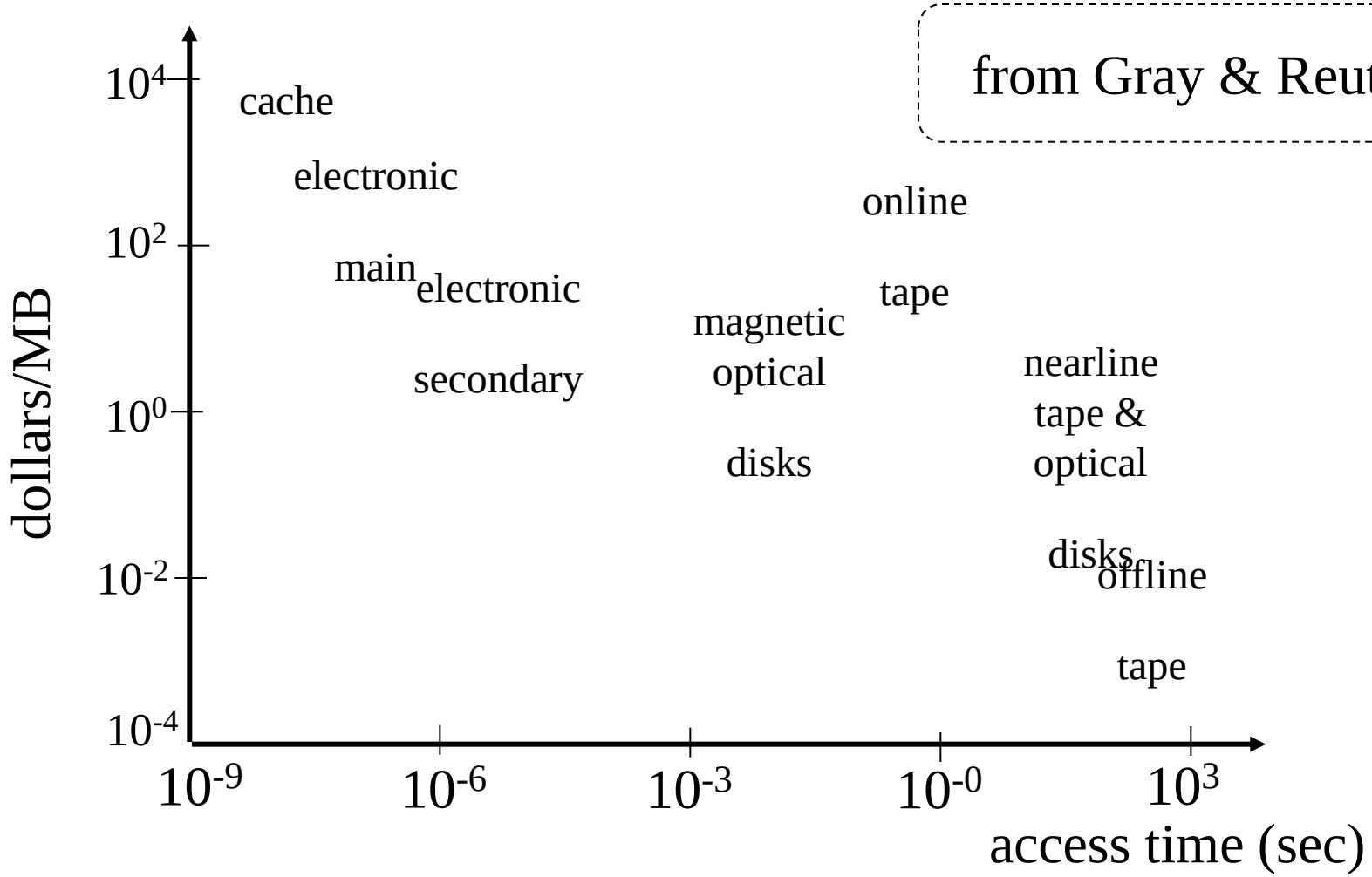
Ιεραρχία Μνήμης



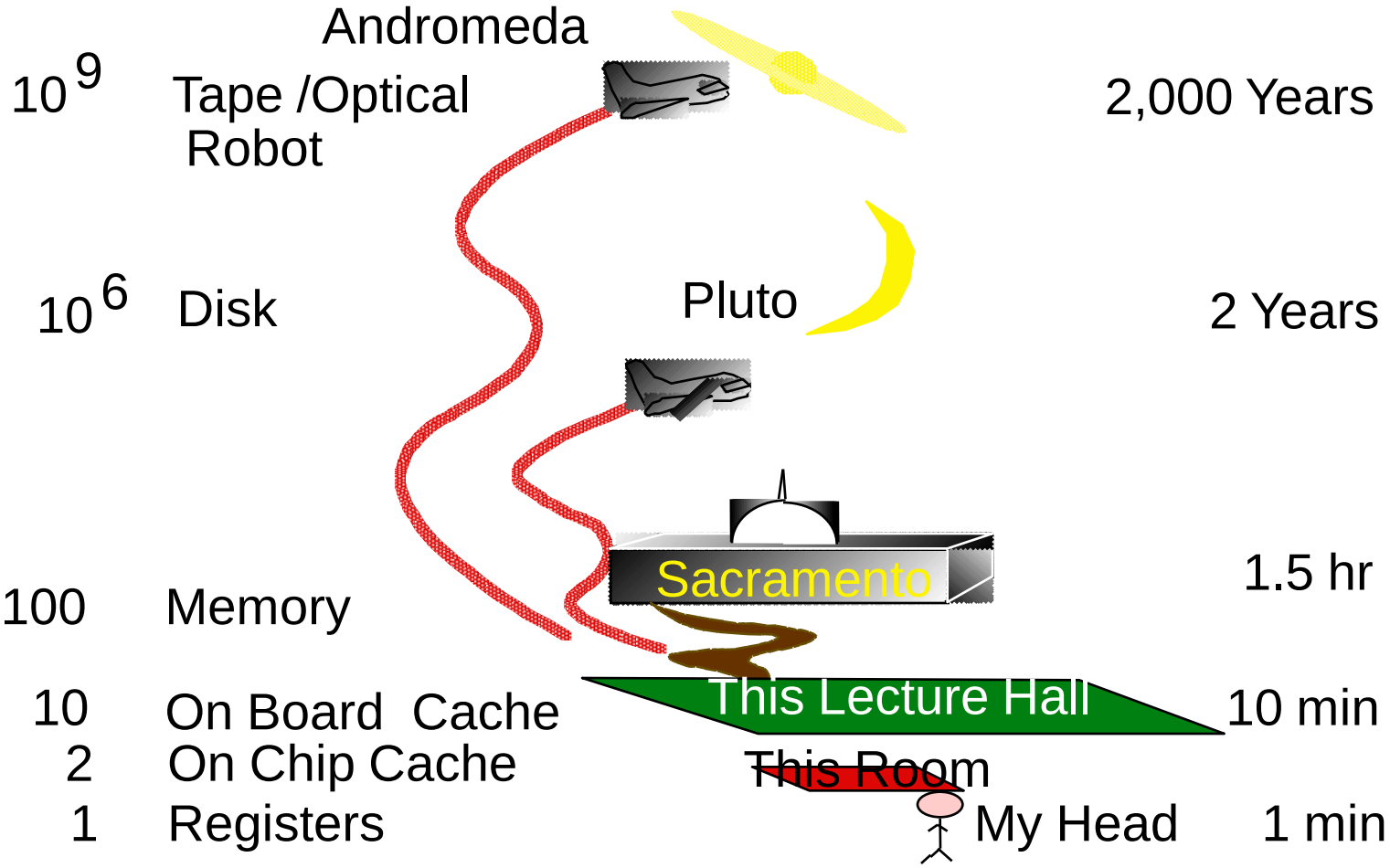
Storage Capacity



Storage Cost

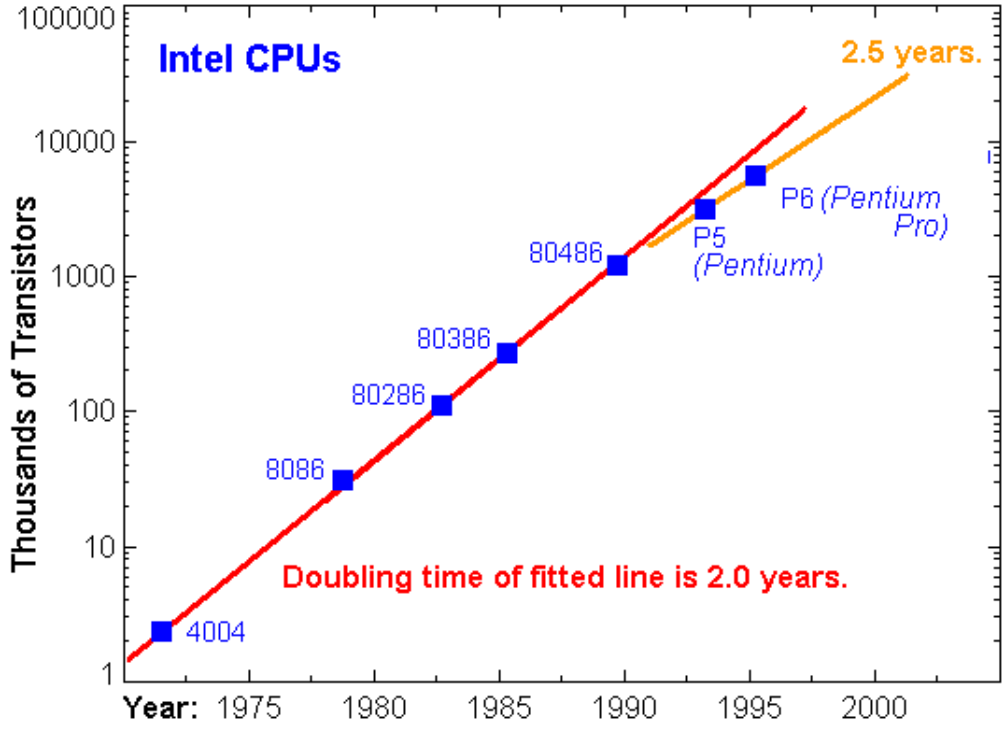


Jim Gray: How Far Away is the Data?



Moore's Law

- Gordon Moore (1965): Ο αριθμός των τρανζίστορ ανά τετραγωνικό εκατοστό διπλασιάζεται κάθε χρόνο



  - x2 τρανζίστορ / 18 μήνες

Αποτυχημένες Προβλέψεις

- Thomas Watson (builder of IBM, in 1943)
  - *"I think there is a world market for maybe five computers."*
- Ken Olsen (founder of Digital Equipment, in 1977)
  - *"There is no reason anyone would want a computer in their home."*

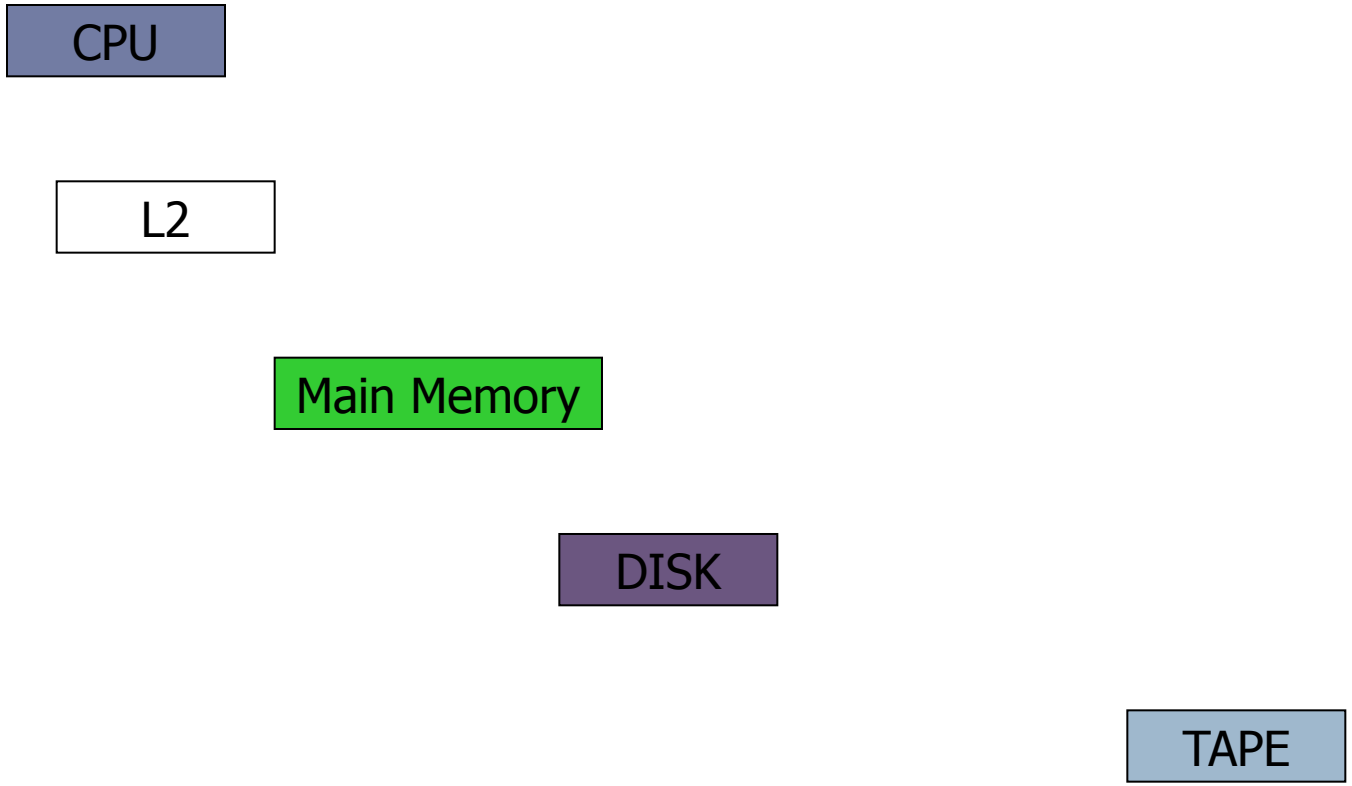
Moore's Law

- Χαρακτηριστικά που ακολουθούν το νόμο του Moore
  - Ταχύτητα επεξεργαστή
    - Νέα τάση: βελτίωση υπολογιστικής ισχύος / Watt
  - Αριθμός bits σε ένα chip
  - Αριθμός bytes σε ένα μαγνητικό μέσο
- Χαρακτηριστικά που **δεν** ακολουθούν το νόμο του Moore
  - Ταχύτητα προσπέλασης στην κύρια μνήμη
  - Ταχύτητα περιστροφής μαγνητικών δίσκων

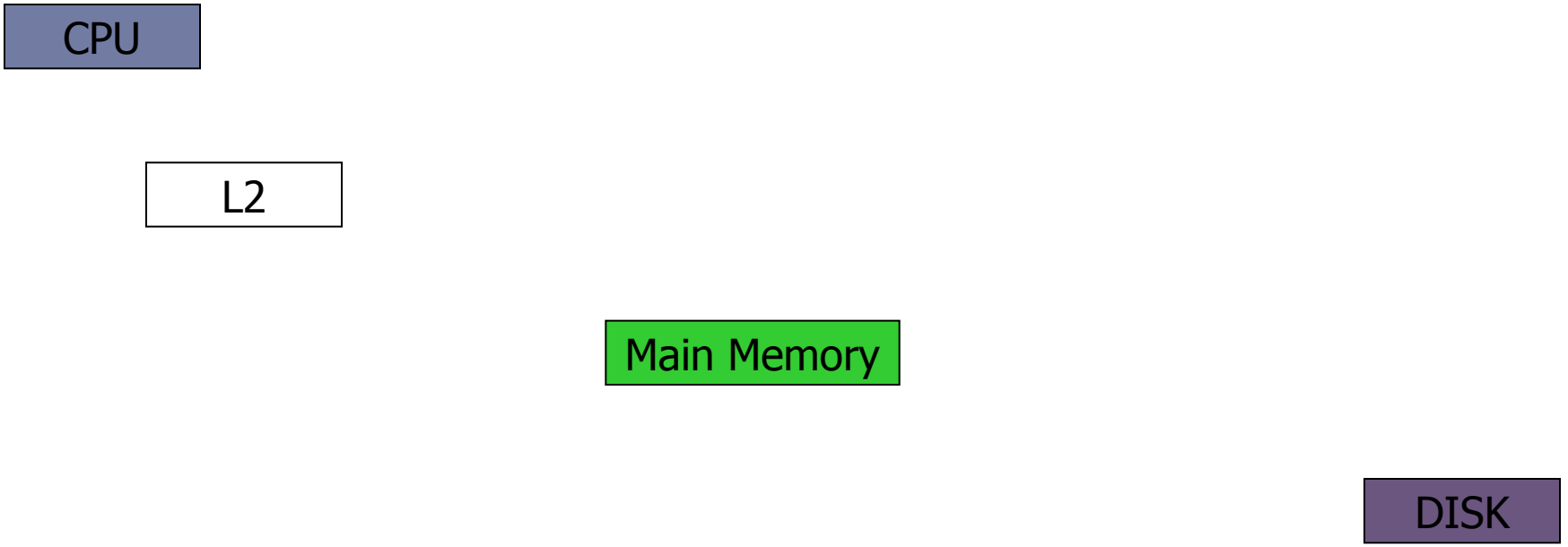
⇒ Καθυστέρηση (Latency) γίνεται σταδιακά μεγαλύτερη

⇒ Ο χρόνος για να μετακινήσουμε δεδομένα/εγγραφές ανάμεσα στα διάφορα επίπεδα της ιεραρχίας αυξάνει σημαντικά σε σχέση με το χρόνο επεξεργασίας στην CPU

Σήμερα

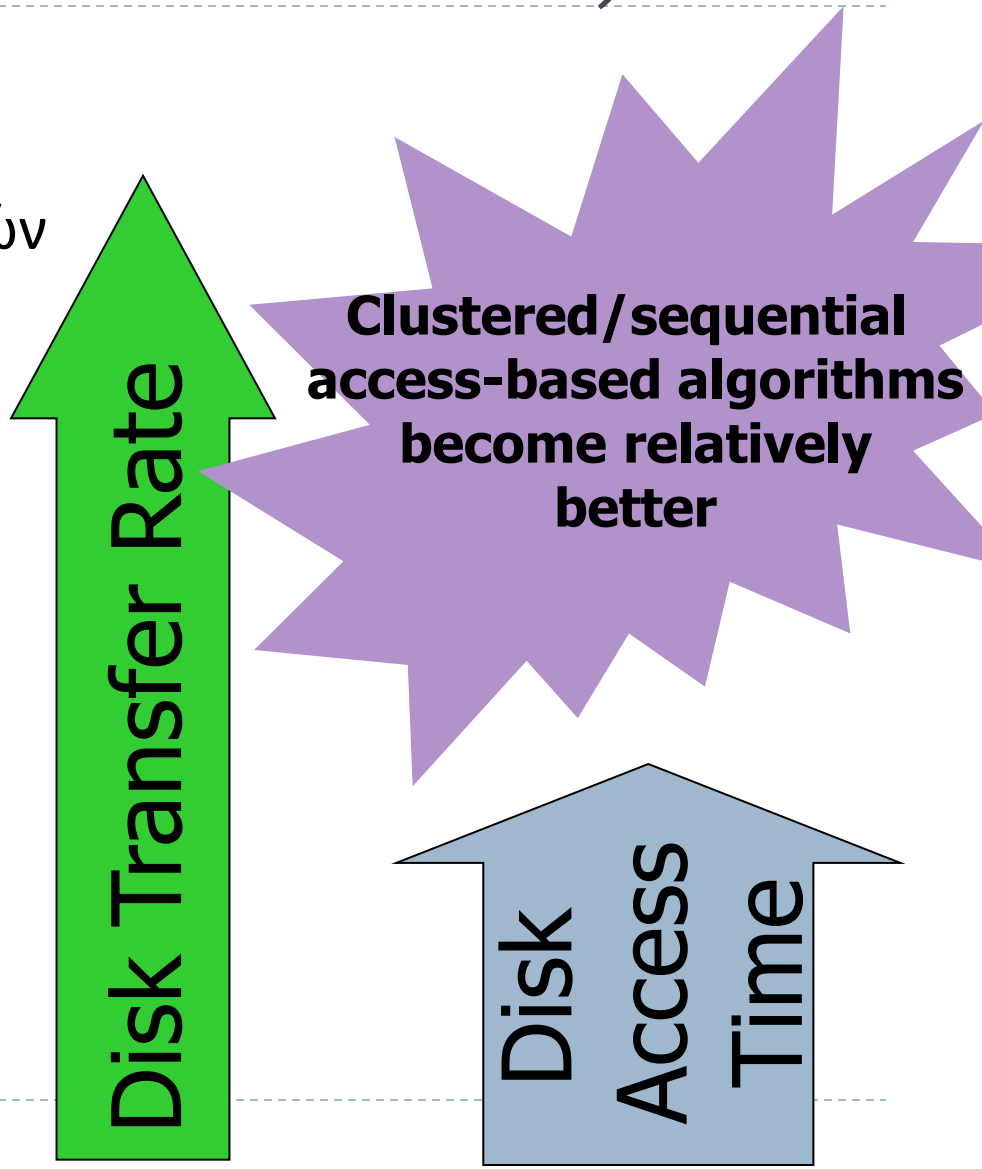


Σε λίγα χρόνια



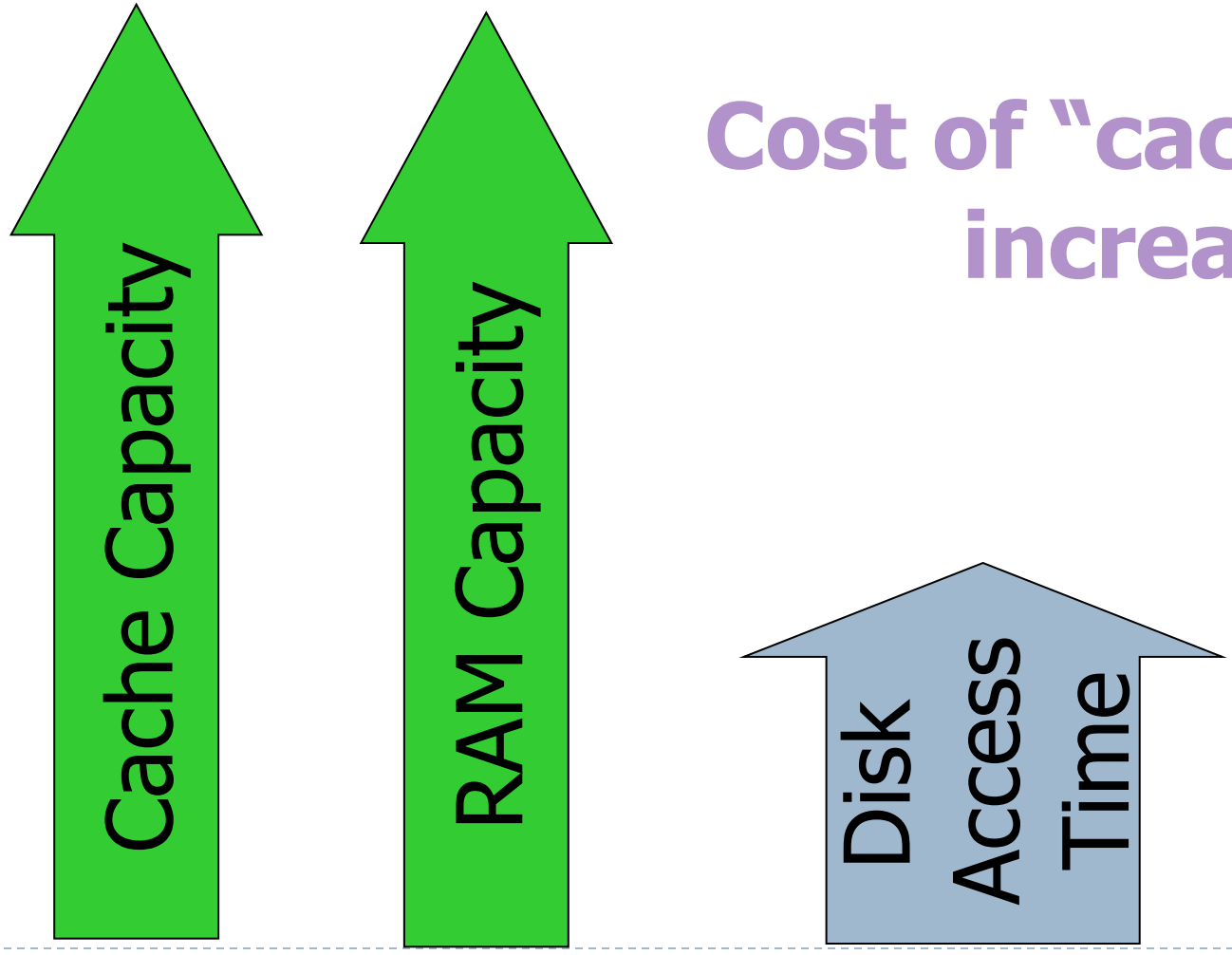
Moore's Law: συνέπειες

- Διαφορετικοί ρυθμοί βελτίωσης χαρακτηριστικών προσπέλασης
  - Ταχύτητα επεξεργαστή
  - Κόστος μνήμης/HDD/SSD
  - Ταχύτητα μνήμης
  - Ταχύτητα περιστροφής HDD
  - Ταχύτητα μεταφοράς
- …οδηγούν σε ανάγκη για αναθεώρηση



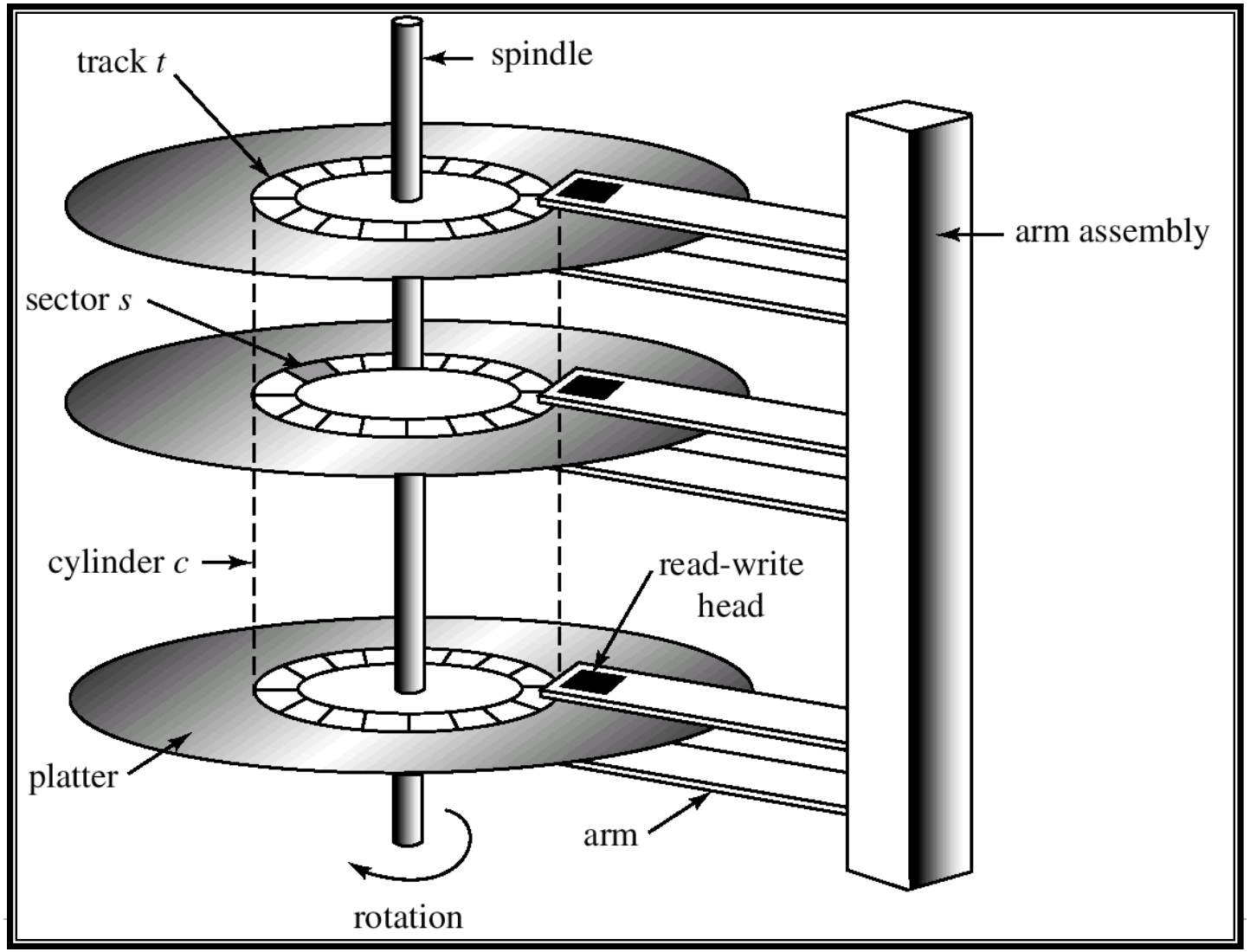
Το ίδιο φαινόμενο και στη RAM

Data Locality Important



Cost of "cache-miss" increases

Δομή Μαγνητικού Δίσκου

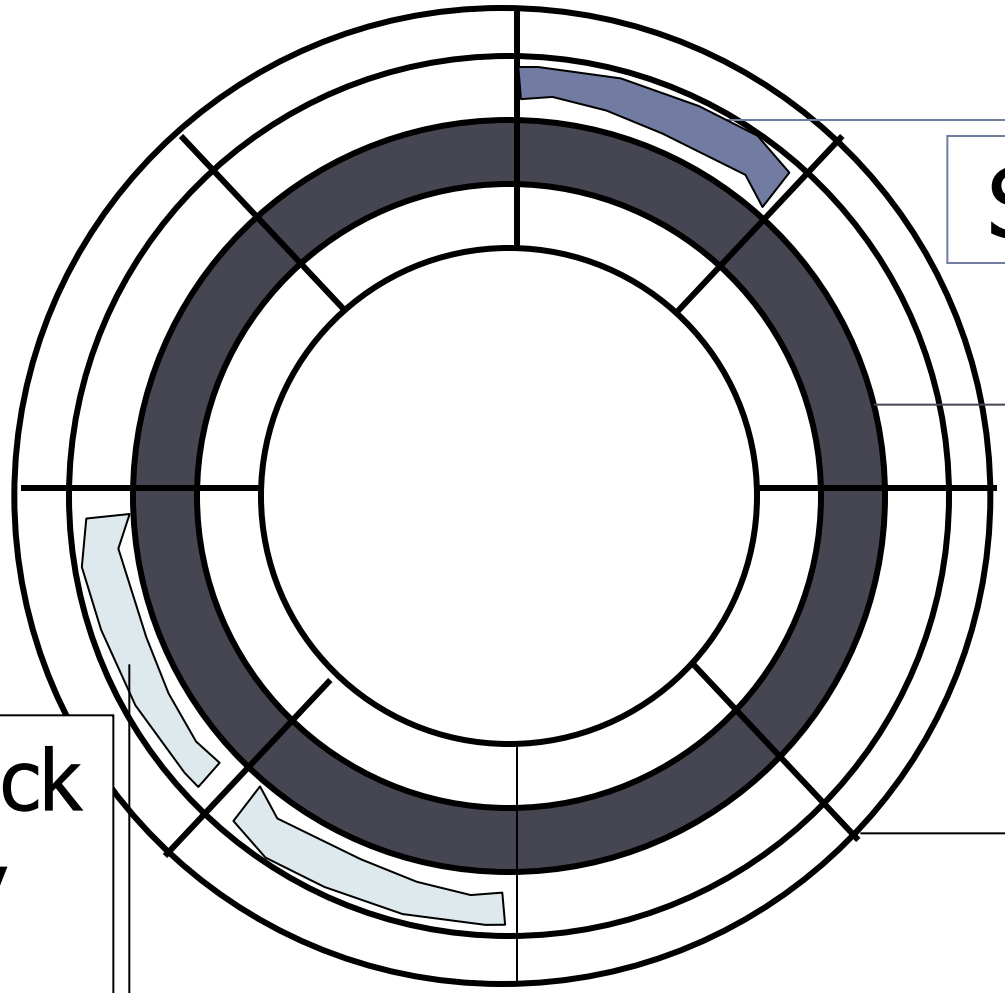


Ορολογία

- Πλακέτα (plate)
- Επιφάνεια (surface)
- Ίχνη/Άτρακτοι (tracks)
- Τομείς (sectors)
- Κεφαλή (head)
- Συστοιχία (cluster/block)

Συνήθως έχουμε διαφορετικό αριθμό sectors ανά track (περισσότερα στα εξωτερικά)

Γενική Εικόνα



Sector

Track

Gap

Logical Block (typically multiple sectors)

Παραδείγματα

Diameter:	1 inch $\rightarrow$ 15 inches
Cylinders:	100 $\rightarrow$ 40000
Surfaces:	1 (CDs) $\rightarrow$
	2 (floppies)
	6-10 (HD)
Sector Size:	512B $\rightarrow$ 54K
Capacity:	360KB (old floppy)
	3TB (newer HDD)

Physical Block Address

- Physical Device (ποιος δίσκος, σε ποιο controller)
  - Cylinder #
  - Surface # (or head#)
  - Sector

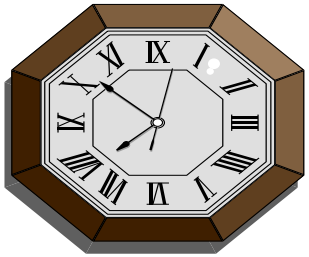
Λογική Δομή Δίσκου

- Οι εφαρμογές βλέπουν το δίσκο σαν ένα μονοδιάστατο πίνακα από λογικές σελίδες (logical blocks).
  - Οι εγγραφές διαβάζουν και γράφουν σελίδες.

- Τα logical blocks του μονοδιάστατου πίνακα αντιστοιχίζονται με φυσικές σελίδες (physical blocks) αποτελούμενες από 1 ή περισσότερα sectors από το σύστημα.

Disk Access Time

I want block X

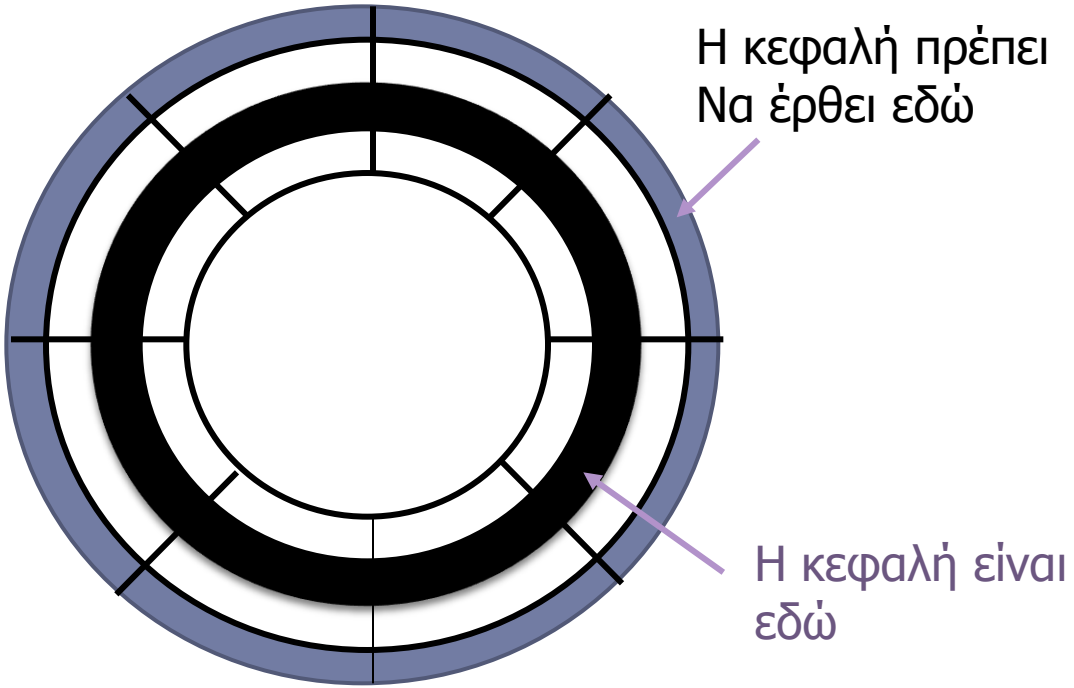


block X in memory

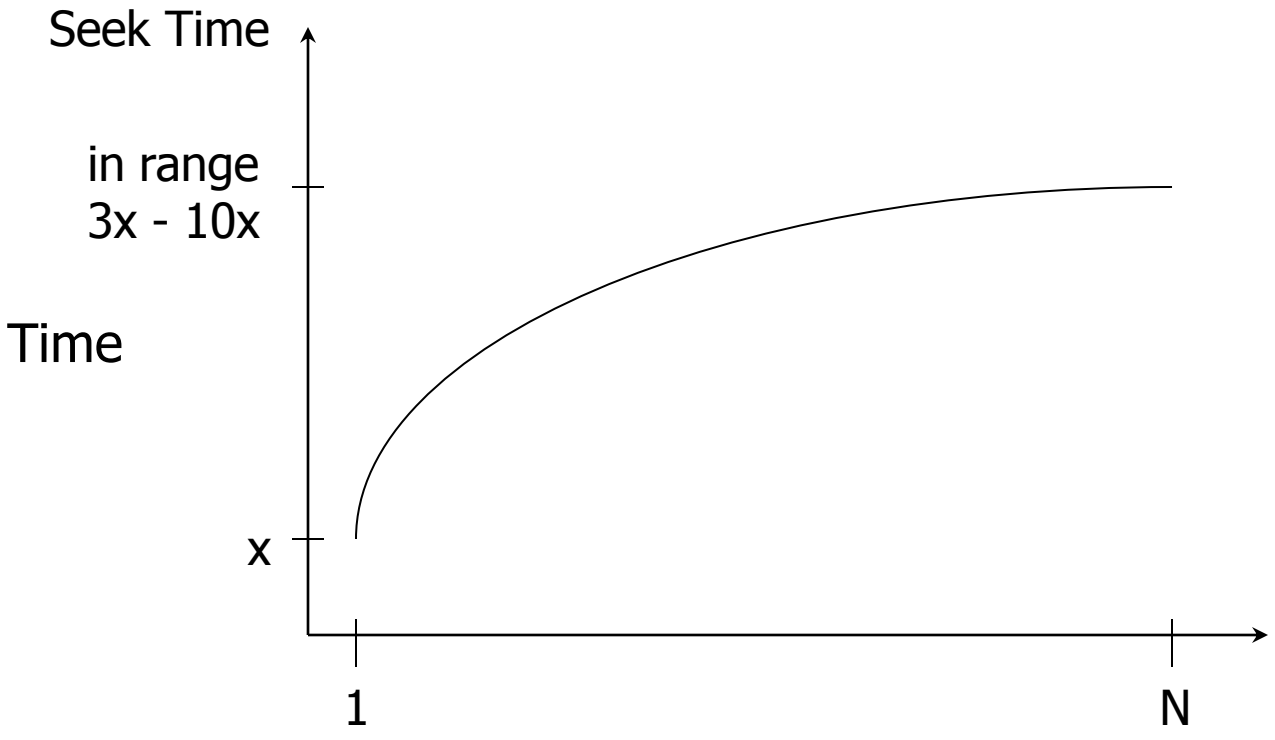
Time = Seek Time (locate track) +
Rotational Delay (locate sector)+
Transfer Time (fetch block) +
Other (disk controller, ...)

Χρόνος Αναζήτησης (Seek Delay/Time)

- Ο χρόνος αναζήτησης ενός δίσκου προσμετράει το χρόνο που απαιτείται για να μετακινηθεί η κεφαλή ανάγνωσης/εγγραφής μεταξύ των ιχνών.



Χρόνος Αναζήτησης σε συνάρτηση με απόσταση μετακίνησης των κεφαλών



Χρόνος εναλλαγής ίχνους/κυλίνδρου (~2msecs)

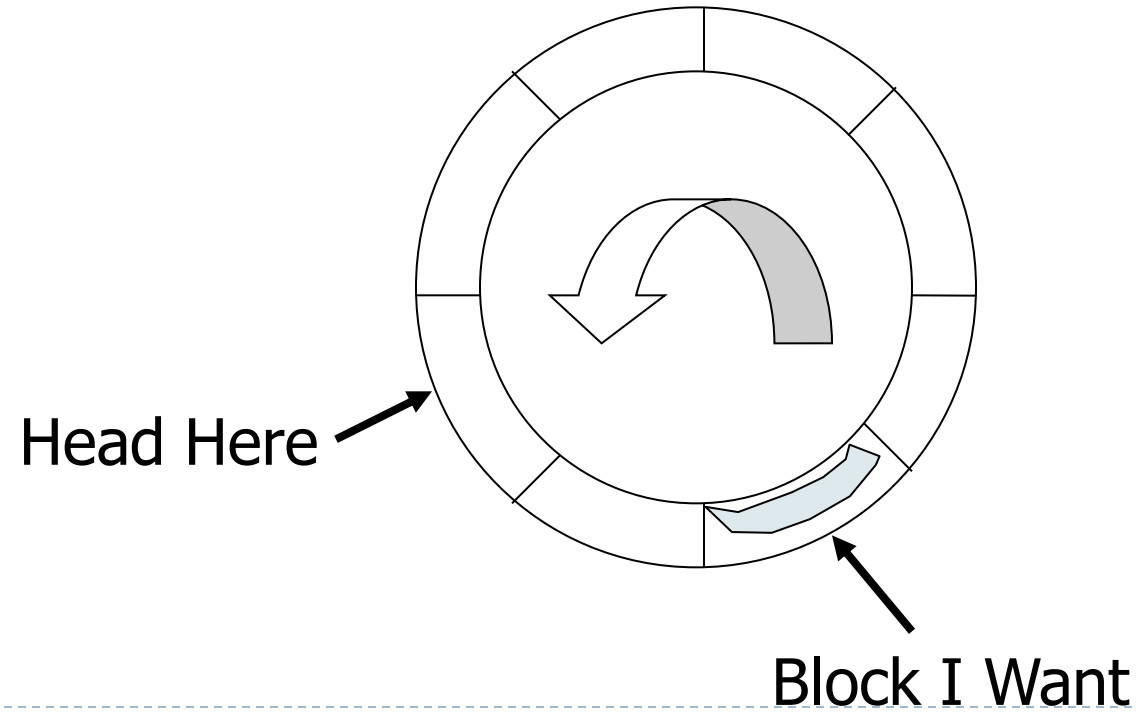
Average Random Seek Time

$$S = \frac{\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \text{SEEKTIME}\ (i \rightarrow j)}{N(N-1)}$$

"Typical" S: 8 ms $\rightarrow$ 40 ms

Λανθάνων χρόνος περιστροφής (Rotational Delay)

- **Λανθάνων χρόνος περιστροφής** είναι ο χρόνος, τον οποίο ο οδηγός πρέπει να περιμένει μέχρι ο σωστός τομέας να φτάσει κάτω από την κεφαλή ανάγνωσης/εγγραφής.



Average Rotational Delay

- Κατά μέσο όσο περιμένω μισή περιστροφή
- R = ο χρόνος για 1/2 περιστροφή

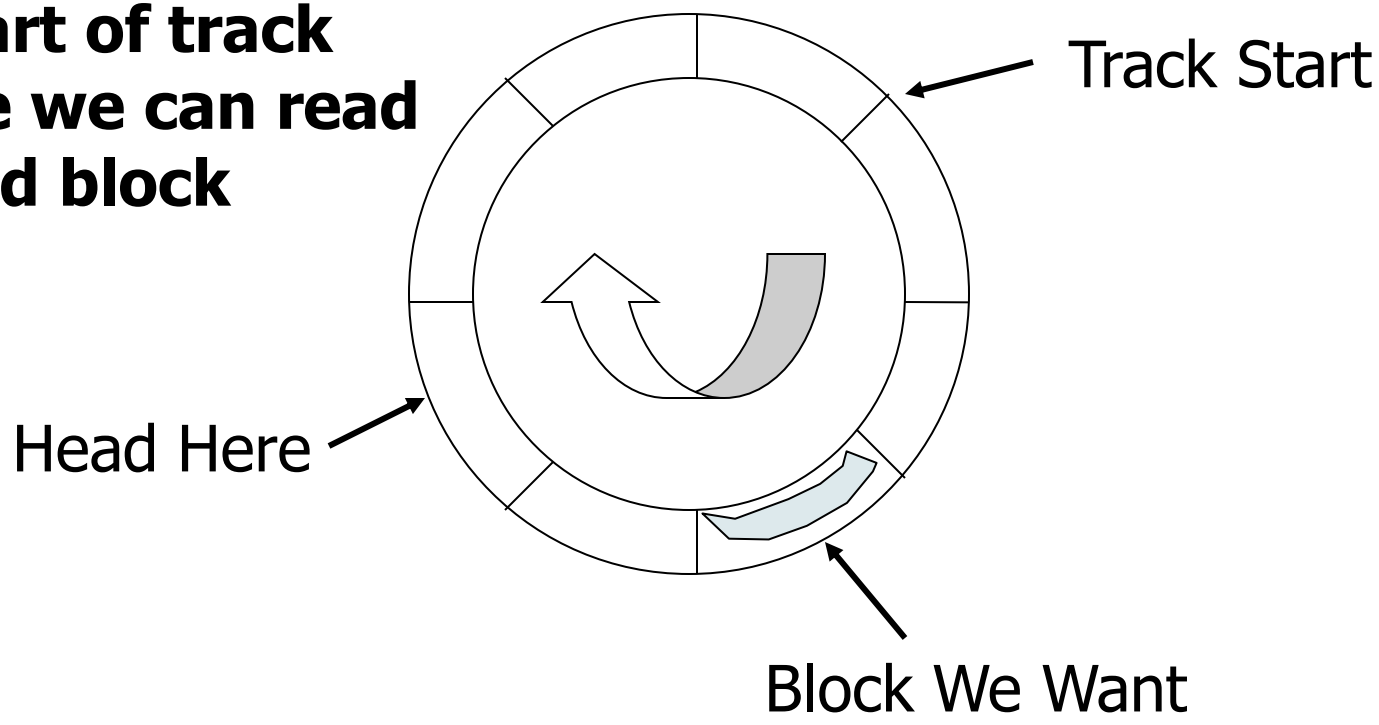
- Έστω δίσκος 7200 RPM

$$\frac{60\ \text{secs}}{R} \qquad \frac{72000\ \text{περιστροφές}}{½\ \text{περιστροφή}}$$

$$R = 30 / 7200 = 4{,}17\text{msecs}$$

Complication

May have to wait for start of track before we can read desired block



Ταχύτητα Μεταφοράς (Transfer Time) (1 block)

- Ο χρόνος που απαιτείται για να «περάσουν» τα sectors του block κάτω από τις κεφαλές
  - Εξαρτάται από την ταχύτητα περιστροφής και το μέγεθος του block (σε sectors)
- 7200 RPM, 128 sectors/track, 2 sectors/block.
- Σε 1 περιστροφή(=60/7200secs) διαβάζω 128 sectors
- Για να διαβάσω 1 block=2 διαδοχικά sectors θέλω

  (2/128) * (60/7200) = 0,13 msecs

- Σύγκριση με seek time ~20msecs, rotational delay ~4msecs

Άλλες καθυστερήσεις

- Χρόνος CPU για στείλει την εντολή για I/O
- Χρόνος επεξεργασίας αιτήματος στον disk controller
- Χρόνος μεταφοράς στη μνήμη μέσω του διαύλου

"Typical" Value: 0

Παράδειγμα

- 1 επιφάνεια
- Rotation speed 7200rpm
- 16384 tracks
- 128 sectors/track
- 4096 bytes/sector
- 4 sectors/block (16KB/block)
- SEEKTIME ($i \rightarrow j$) = [1000 + |j-i|] μs

- Υπολογίστε ελάχιστο, μέγιστο και μέσο χρόνο για την ανάγνωση ενός block.

Υπολογισμός Ελάχιστου Χρόνου

- Η κεφαλή είναι στην αρχή του 1ου sector του block που θέλουμε να διαβάσουμε
  - Υπολογίζω μόνο το transfer time
- 4 sectors είναι τα 4/128 ενός track
- 1 περιστροφή παίρνει 60/7200=8,33ms
- Transfer time = 8,33 * 4 /128 = 0,26ms
- Total time (min) = 0,26ms

Υπολογισμός Μέγιστου Χρόνου

- Υποθέτω μέγιστη μετακίνηση κεφαλής (πχ από το ποιο εσωτερικό στο ποιο εξωτερικό track)
  - Seek time = 1000+(16384-1)μs=17,38ms
- Μετά το seek, η κεφαλή μόλις έχασε την αρχή του block που ψάχνω
  - Full rotational delay =60/7200 = 8,33ms
- Transfer time (όπως πριν) = 0,26ms
- Total time(max)=17,38+8,33+0,26=25,97ms

~100 x min-time!!!

Μέσος χρόνος

- Seek time = AVERAGE(1000+|i-j|)=…=6,46ms
- Rotational delay: μισή περιστροφή = ½*60/7200secs=4,17ms
- Transfer Time 0,26ms
- Σύνολο = 10,89ms

Ελάχιστος Χρόνος	0,25ms
Μέσος Χρόνος	10,81ms
Μέγιστος Χρόνος	25,97ms

- Είδαμε: τυχαία προσπέλαση
- Τι γίνεται αν θέλω να διαβάσω το αμέσως επόμενο block;

Αν γίνει σωστά…

Time to get next block = 0,26ms + Negligible

- skip gap
- once in a while, next cylinder

Παρατήρηση

Rule of Thumb

Random I/O: Ακριβό
Sequential I/O: **Πολύ** Φθηνότερο

- Χρόνος για να διαβάσω 1Block
  - Random I/O: 11 ms. (avg)
  - Sequential I/O: <1ms.

Σειριακή ταχύτητα ανάγνωσης

- Τι γίνεται αν διαβάζω συνεχόμενα sectors στο ίδιο track;
- 128 sectors/track * 4KB/sector σε 8,33ms/track = 512KB/8.33ms = 60MB/sec!!!
  - Μοντέρνοι δίσκοι (PC) 100-150MB/sec
- Random I/O (avg): 11ms/4KB → 0,33MB/sec

Εγγραφή

- Το κόστος εγγραφής είναι παρόμοιο με το κόστος ανάγνωσης
- …εκτός και αν θέλουμε να επαληθεύσουμε, οπότε περιμένουμε μία πλήρη περιστροφή για να διαβάσουμε ότι γράψαμε

Ενημέρωση σελίδας (block)

- Για να αλλάξω το περιεχόμενο μίας σελίδας (block)
  - Διαβάσω τη σελίδα στη μνήμη
  - Κάνω ότι αλλαγές χρειάζονται στο αντίγραφο στη μνήμη
  - Γράφω τη σελίδα πίσω στο δίσκο
- Προσοχή: τα παραπάνω γίνονται ακόμα και θέλω να αλλάζω την τιμή ενός bit!

Παράδειγμα- Megatron 747 Disk

- 3600 RPM
- 1 surface
- 16 MB usable capacity (16 X 2^{20})
- 128=2^7 cylinders
- 1 block=1sector=1KB
- 10% κενό (gaps) ανάμεσα στα sectors
- seek time= 25 ms κατά μέσο όρο
- seek time για να πάω στον επόμενο κύλινδρο= 5 ms.

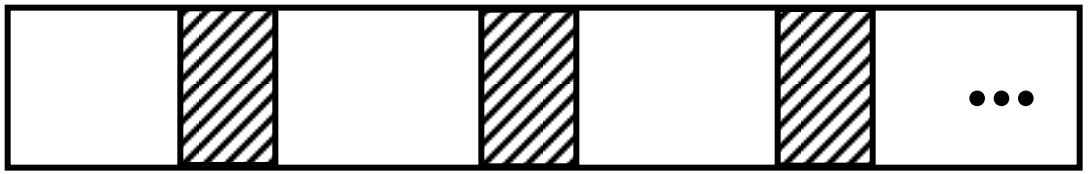
Παρατηρήσεις

- 1 επιφάνεια, ο κύλινδρος ταυτίζεται με το ίχνος (track)
- Χωρητικότητα = 16 MB = $(2^{20})16 = 2^{24}$
- bytes/cyl = $2^{24}/2^{7} = 2^{17}$ = 128 KB
- blocks/cyl = 128 KB / 1 KB = 128

- Δηλαδή κάθε track έχει 128 sectors του 1KB

3600 RPM: 60/3600 = 16,66ms για 1 περιστροφή

One track:



Time over useful data:(16,66)(0,9)=14,99 ms.

Time over gaps: (16,66)(0,1) = 1,66 ms.

Transfer time 1 block = 14,99/128=0,117 ms.

Trans. time 1 block+gap=16,66/128=0,13ms.

T_1 = Time to read one random block (1KB Block)

T1 = seek + rotational delay + transfer time

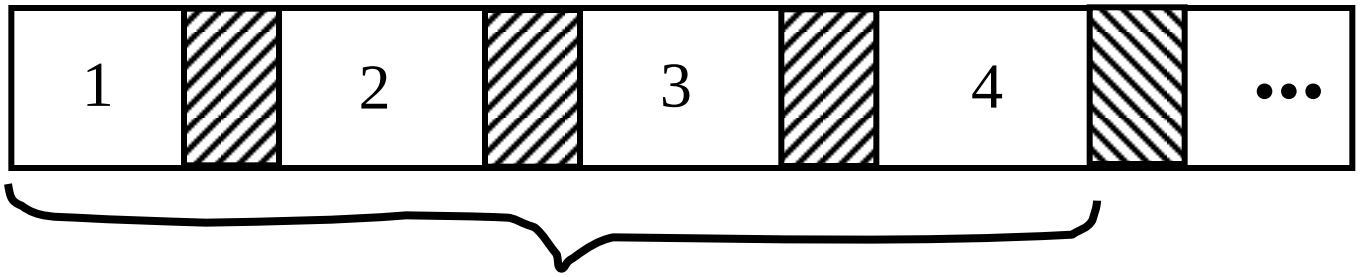
= 25 + (16,66/2) + 0,117 = **33,45 ms**

υποθέτω μισή περιστροφή

Διαχωρισμός block/σελίδας

- Ως τώρα υποθέταμε ότι η «σελίδα» που χρησιμοποιεί το O/S και οι εφαρμογές ταυτίζεται με 1 block στο δίσκο
- Στην πράξη το O/S (ή το ΣΔΒΔ) μπορεί να χρησιμοποιεί το δικό της «μέγεθος σελίδας»
  - Μπορεί να αλλάζει ανάλογα με τη δομή αποθήκευσης (σχέση, ευρετήριο)

Διάβασμα σελίδας 4KB (four sectors) από το δίσκο



One sector+ gap

One sector, no gap

$T_4 = 25 + (16,66/2) + (0,117) \times 1$

$+ (0,130) \times 3 = 33,83$ ms

[Compare to $T_1 = 33,45$ ms]

Τι γίνεται αν 1 σελίδα=1 track=128KB;

T_T = Time to read a full track

(υποθέτω μπορώ να αρχίσω το διάβασμα από οποιοδήποτε sector του track)

$T_T = 25 + (0,130/2) + 16,66 - 0,013 = 41,72$ ms

to get to start of next sector

do not have to read last gap

Ας συγκρίνουμε το κόστος για να διαβάσω 1 σελίδα

- Read 1KB page : 33,45ms
- Read 4KB page : 33,83ms
- Read 128KB page : 41,73ms

- Γιατί να μη μεγαλώσω και άλλο το μέγεθος της σελίδας?

Σωστό Μέγεθος Σελίδας ΣΔΒΔ?

- Μεγάλη σελίδα -> διαμοιράζεται το κόστος ανάγνωσης ανάμεσα σε πολλές εγγραφές

ΑΛΛΑ

- Ίσως διαβάζω και πολλές άλλες εγγραφές που δε χρειάζομαι (και ο χρόνος ανεβαίνει, έστω και λίγο)
  - Επίσης η μεγάλη σελίδα όταν έρθει στη μνήμη καταλαμβάνει περισσότερο χώρο
- Τι γίνεται με την παρακάτω SQL επερώτηση
  - **update Account set Balance=Balance+100 where AcountNumber=12345**

Τι πρέπει να σκεφτώ

- Μικρές – μεγάλες σχέσεις?
- Ευρετήρια
- Read mostly or Update-heavy workload?

- Διαφορετικά μεγέθη σελίδας?
  - Oracle

Optimizations: Prefetching

- Prefetching = λήψη δεδομένων πριν αυτά χρειαστούν
- Locality of access = δεδομένα, στο χρόνο καλούνται από διαδοχικές ή κοντινές περιοχές στο δίσκο.
  - Παράδειγμα: table/relation scan
  - Προϋποθέτει ότι έχουμε μεριμνήσει οι σελίδες της σχέσης να βρίσκονται «κοντά» πχ στο ίδιο track, κύλινδρο
- Μπορούμε να κρύψουμε μέρος του χρόνου ανάγνωσης από το δίσκο χρησιμοποιώντας λίγο περισσότερη μνήμη
  - Double buffering

Double Buffering

- Problem: Have a File
  - Sequence of Blocks B1, B2

- Have a Program
  - Process B1
  - Process B2
  - Process B3

  (σειριακή ανάγνωση)

Single Buffer Solution

(1) Read B1 $\rightarrow$ Buffer

(2) Process Data in Buffer

(3) Read B2 $\rightarrow$ Buffer

(4) Process Data in Buffer ...

Single Buffer Time

Έστω: P = χρόνος επεξεργασίας/σελίδα

R = χρόνος για να διαβάσω 1 σελίδα (υποθέτω τυχαία προσπέλαση)

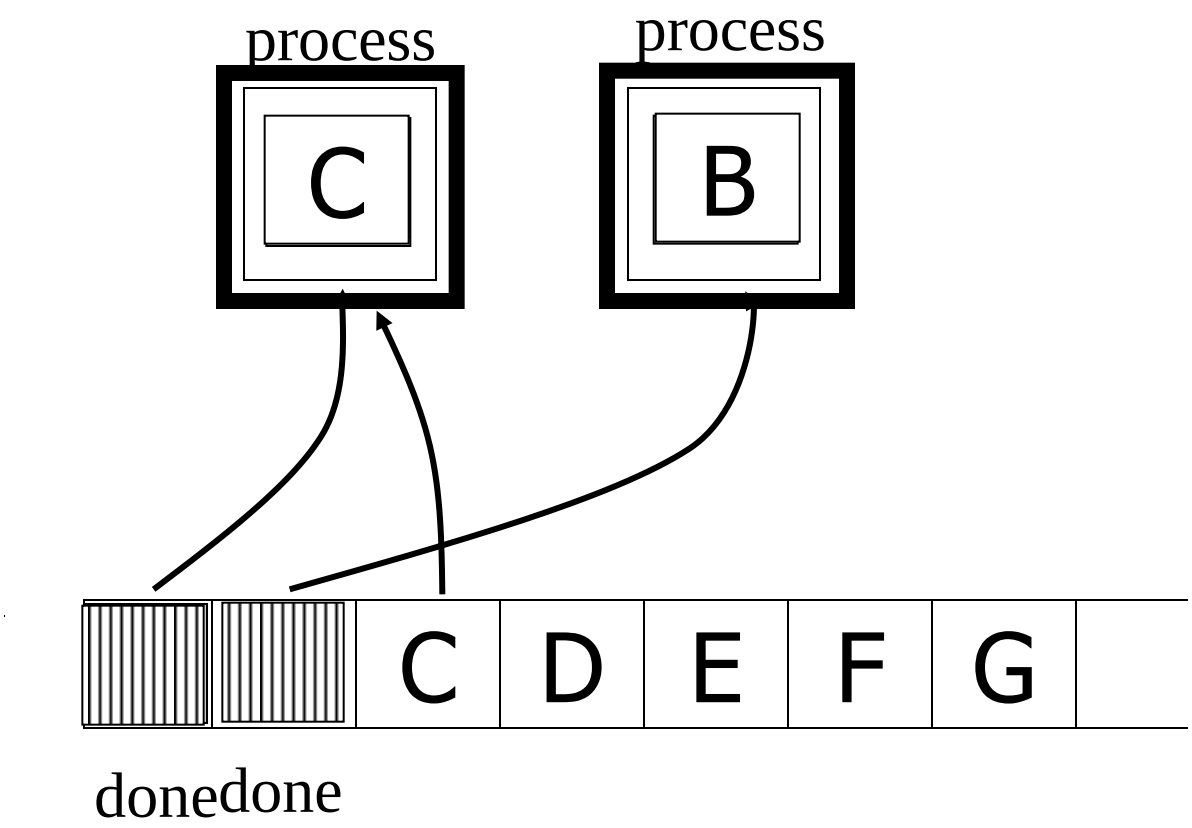
n = # αριθμός σελίδων

Single buffer time = n(P+R)

Double Buffering

Memory:

Disk:



Double Buffering Time

P = χρόνος επεξεργασίας/σελίδα
R = χρόνος για να διαβάσω 1 σελίδα
(υποθέτω τυχαία προσπέλαση)
n = αριθμός σελίδων

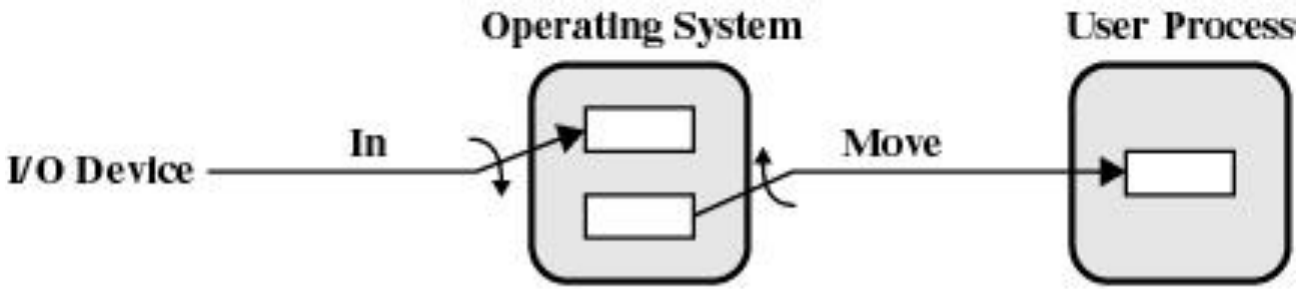
Υποθέτω $P \geq R$

- Double buffering time = R + nP
- Single buffering time = n(R+P)

More on Prefetching (application level)

- Requires efficient asynchronous I/O implementation
- Data dependant scheduling of I/O
- DB process queues up several requests
  - Better chance for disk controller to optimize seeks
  - Better utilization of multiple disks (allow them to work in parallel)
- Big savings in CPU intensive tasks (e.g. *sorting*)

Circular Buffering



(c) Double buffering

(d) Circular buffering

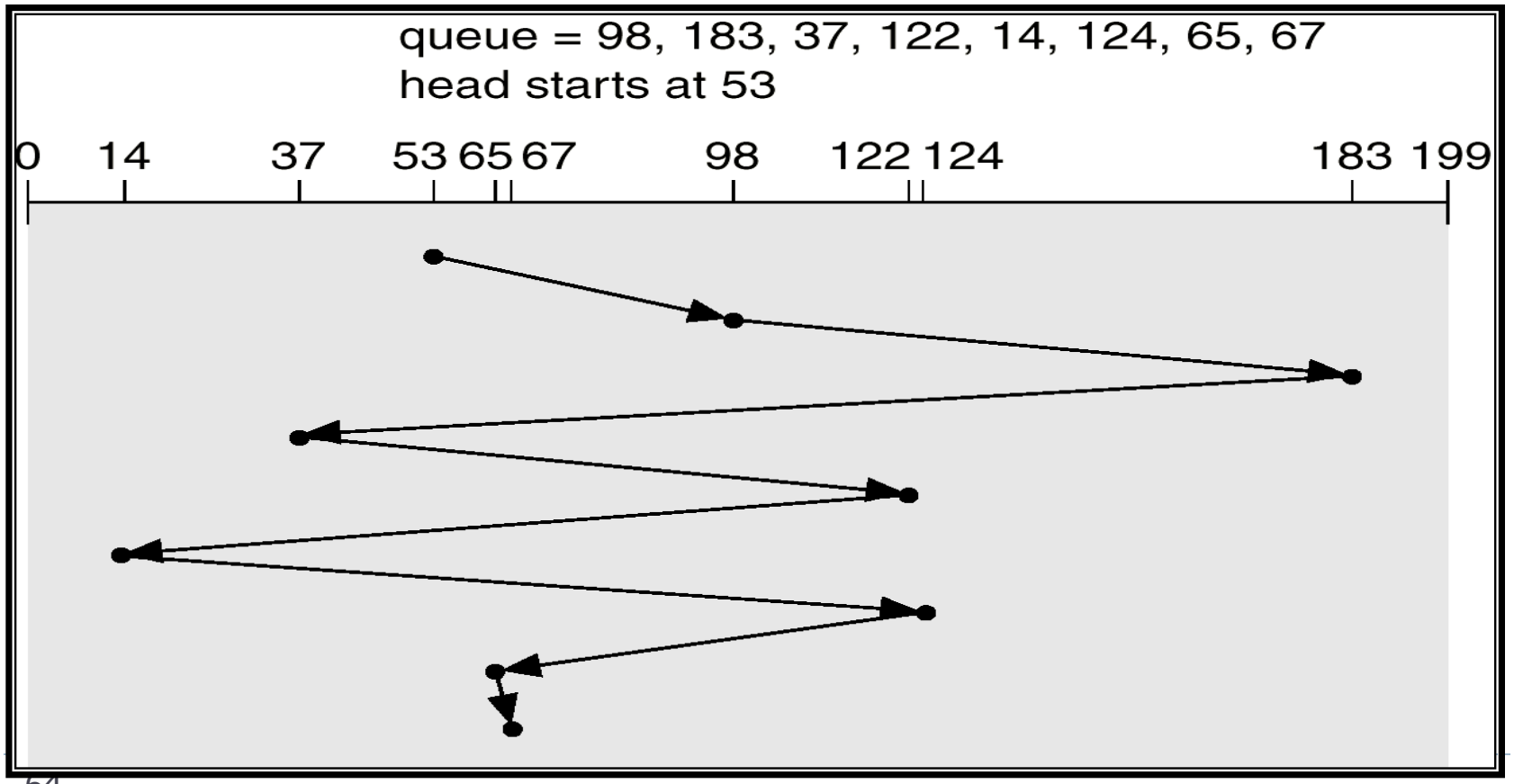
I/O Buffering Schemes (input)

Περισσότερες βελτιστοποιήσεις

- Disk Scheduling Algorithm
  - Elevator algorithm
- Disk Arrays (RAID)
- On Disk Cache

First- Come-First-Served (FCFS)

- Δίκαιη πολιτική
- Αργή

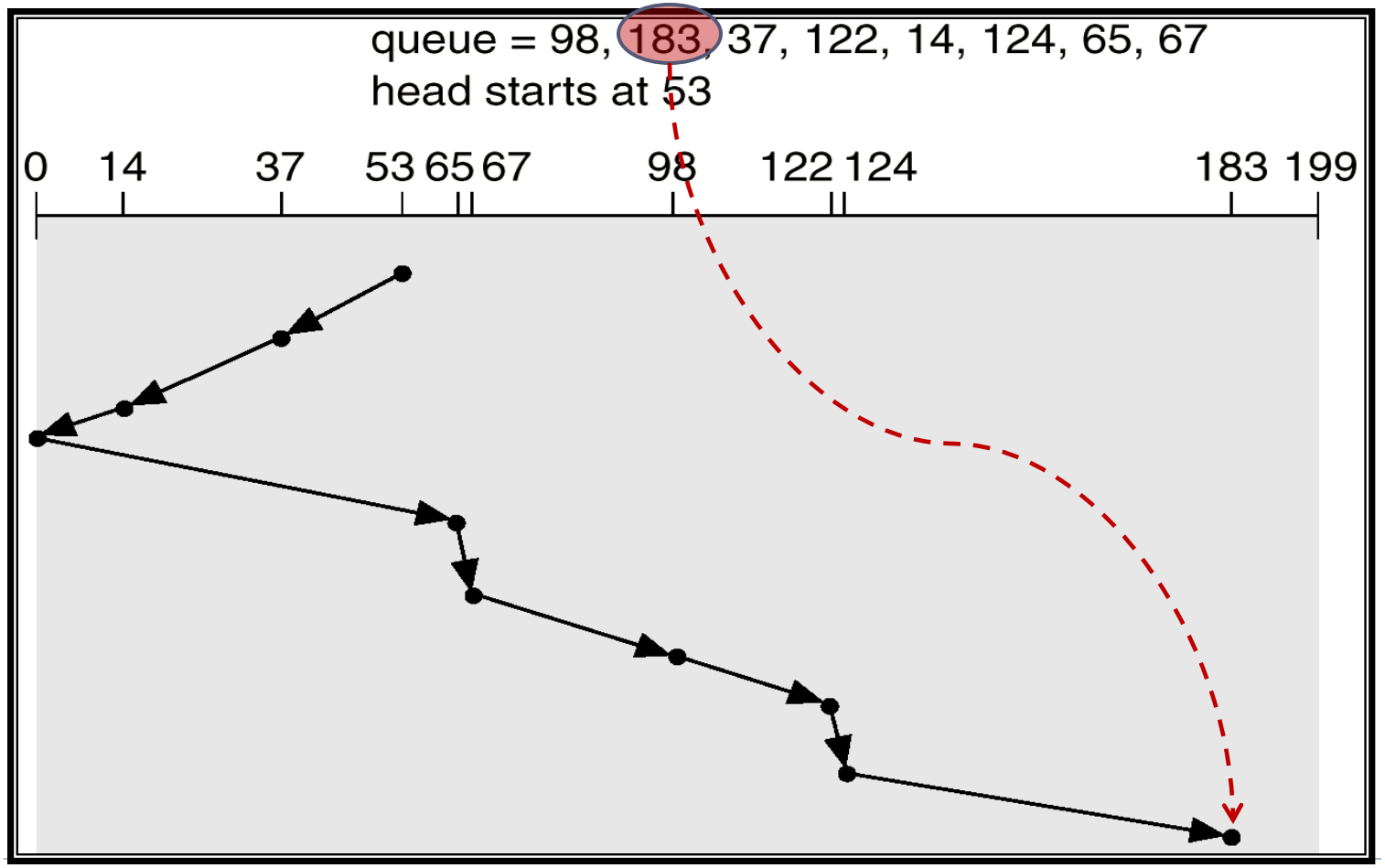


SCAN or Elevator Algorithm

- Η κεφαλή του δίσκου ξεκινάει από τη μία άκρη (πχ έξω) και κινείται προς την άλλη (πχ μέσα) εξυπηρετώντας αιτήσεις. Μόλις φτάσει στο άλλο άκρο αλλάζει κατεύθυνση
  - Συνολικά καλύτερη απόδοση (μειώνει το seek time)
  - Κάποιες αιτήσεις για σελίδες ενδέχεται να περιμένουν αρκετά μέχρι η κεφαλή να αλλάξει κατεύθυνση
  - Νεώτερες αιτήσεις μπορεί να εξυπηρετηθούν νωρίτερα από άλλες που περιμένουν περισσότερη ώρα

SCAN

•Total head movement=208 cylinders



Solid State Disks (SSD)

- Χρήση μνήμης flash, απουσία κινούμενων μερών
  - πολύ γρήγορη τυχαία προσπέλαση (random I/O)
  - **μικρότερη κατανάλωση ρεύματος**
  - αμελητέες απαιτήσεις για ψύξη
  - καθόλου θόρυβος
- Αρκετά ακριβότεροι από HDD

Solid State Disks (SSD)

- Υποστηρίζουν συγκεκριμένο αριθμό εγγραφών-επανεγγραφών (wearing)
  - Μπορώ να γράψω άμεσα σε μία κενή σελίδα (4KB) αλλά αν αυτή δεν είναι κενή η εγγραφή γίνεται σε επίπεδο πολλών σελίδων (πχ 128) κάνοντας τα πρώτα erase ακόμα και αν αλλάζει μία μόνο σελίδα 4K
- Σταδιακή μείωση της απόδοσης, ανάγκη για αναδιάταξη (Garbage Collection, TRIM)

Παράδειγμα Ομάδα 16 σελίδων των 4KB

Ομάδα Χ

FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE

Γράψε σελίδες A,B,C,D

Ομάδα Χ

A	B	C	D
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE

Παράδειγμα: ομάδα 16 σελίδων των 4KB

Γράψε σελίδες E,F,G,H

A	B	C	D
E	F	G	H
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE

Τροποποίησε A',B',C',D'

A	B	C	D
E	F	G	H
A'	B'	C'	D'
FREE	FREE	FREE	FREE

Mark as Stale

Για να γράψω πάλι εδώ πρέπει να κάνω erase όλη την ομάδα

Garbage Collection: Πως ξανακερδίσω τα stale pages?

Ομάδα X

A	B	C	D
E	F	G	H
A’	B’	C’	D’
FREE	FREE	FREE	FREE

Ομάδα Y

FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE

Κάνε erase όλη τη X

FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE

Αντέγραψε στην Y τα χρήσιμα pages της X

E	F	G	H
A’	B’	C’	D’
FREE	FREE	FREE	FREE
FREE	FREE	FREE	FREE