POLITECNICO DI MILANO

Computing Infrastructure





Storage

Files

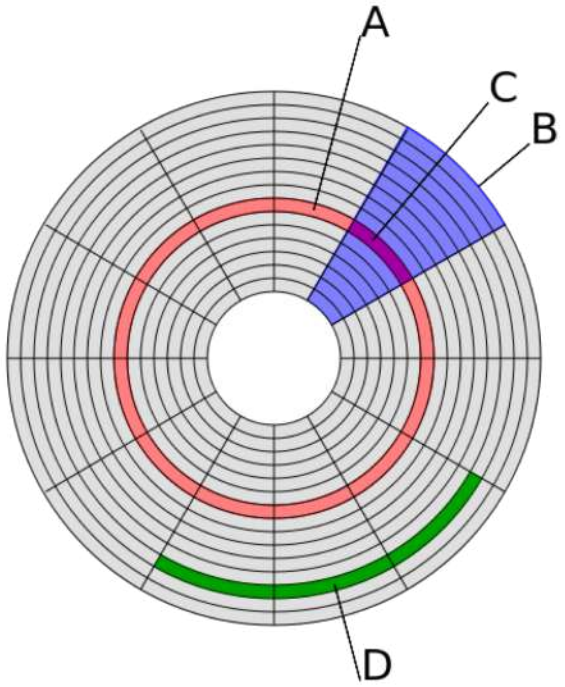
Disks can be seen by a OS as a collection of *data blocks* that can be read or written independently.

Files

To allow the ordering/management among them, each block is characterized by a unique numerical address called LBA *(Logical Block Address)*.

Files

Typically, the OS groups blocks into *clusters* to simplify the access to the disk. Clusters are the minimal unit that a OS can read from or write to a disk.



(A) track (B) geometrical sector
(C) track sector (D) cluster

To reduce the overhead of managing on-disk data structures, the filesystem does not allocate individual disk sectors, but contiguous groups of sectors, i.e., clusters.

Files

- A cluster, or allocation unit, is a group of sectors that make up the smallest unit of disk allocation for a file within a file system. In other words, a file system's cluster size is the smallest amount of space a file can take up on a computer.
- Typical cluster sizes range from 1 sector (512 B) to 128 sectors (64 KB).

Files

Clusters contains:

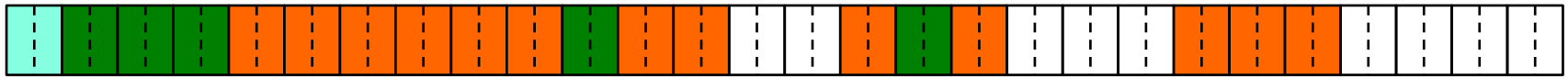
- File data: the actual content of the files
- Meta data: the information required to support the file system.

Meta data contains:

- File names
- Directory structures and symbolic links
- File size and file type
- Creation, modification, last access dates
- Security information (owners, access list, encryption)
- *Links to the LBA where the file content can be located on the disk*

Files

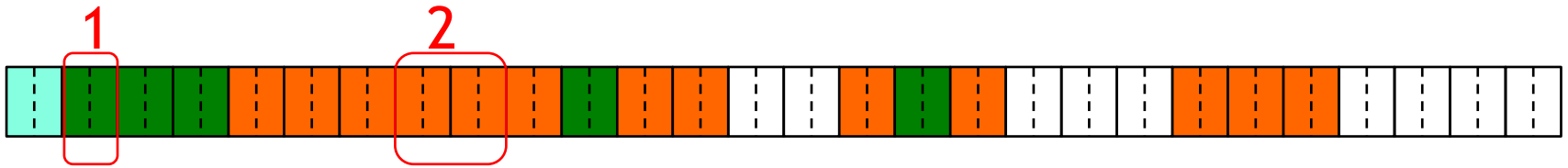
The disk can thus contain several types of clusters:



- Meta data – fixed position (to bootstrap the entire file system)
- Meta data – variable position (to hold the folder structure)
- File data (actual content of the files)
- Unused space (available to contain new files and folders)

Files - reading

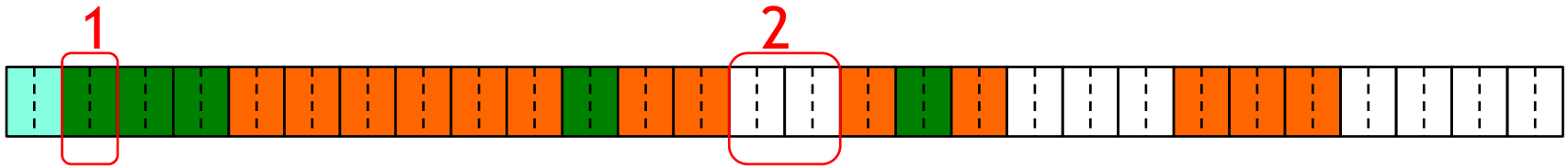
Reading a file requires to:



1. Accessing the meta-data to locate its blocks.
2. Access the blocks to read its content

Files - writing

Writing a file requires to:



1. Accessing the meta-data to locate free space.
2. Write the data in the assigned blocks

Files – writing (2)

Since the file system can only access clusters, the real occupation of space on a disk for a file is always a multiple of the cluster size.

Let us call:

- *s* – the file size
- *c* – the cluster size
- *a* – the actual size on disk

Then, we have:

$$a = ceil(s / c) * c$$

And the quantity ***w = a – s*** is ***wasted disk space*** due to the organization of the file into clusters.

This waste of space is called *internal fragmentation* of files.

Files – writing (2): Example

An example of internal fragmentation:

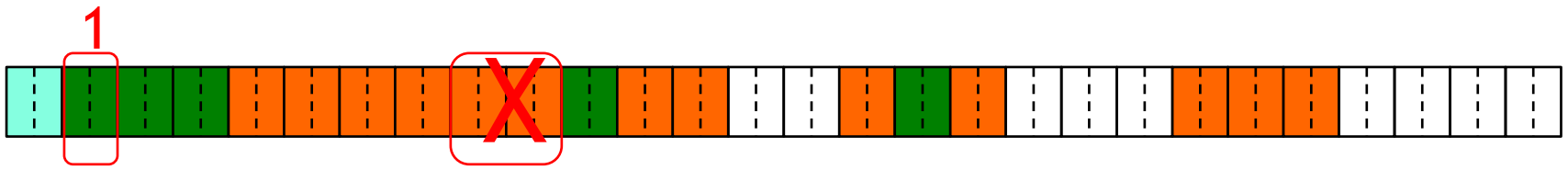
- s - **file size = 27 byte**
- c – **cluster size = 8 byte**
- **actual size on the disk**

$$a = ceil(27 / 8) * 8 = ceil(3.375)*8 = 4 *8 = \mathbf{32\ byte}$$

- ***Wasted disk space = 32 – 27 = 5 byte***

Files - deleting

Deleting a file requires:

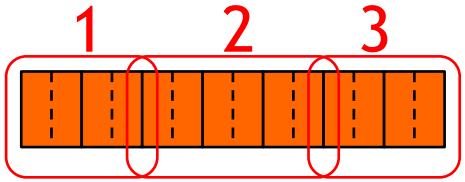


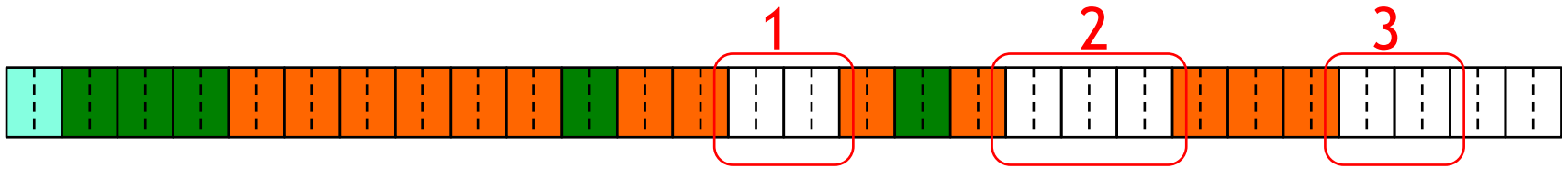
1. Only to update the meta-data to say that the blocks where the file was stored are no longer in use by the O.S.

Deleting a file never actually deletes the data on the disk: when a new file will be written on the same clusters, the old data will be replaced by the new one.

Files – external fragmentation

As the life of the disk evolves, there might not be enough space to store a file contiguously.





In this case, the file is splitt into smaller chunks that are inserted into the free clusters spread over the disk.

The effect of splitting a file into non-contiguous clusters is called *external fragmentation*.

As we will see, this can reduce a lot the performance of an HDD.

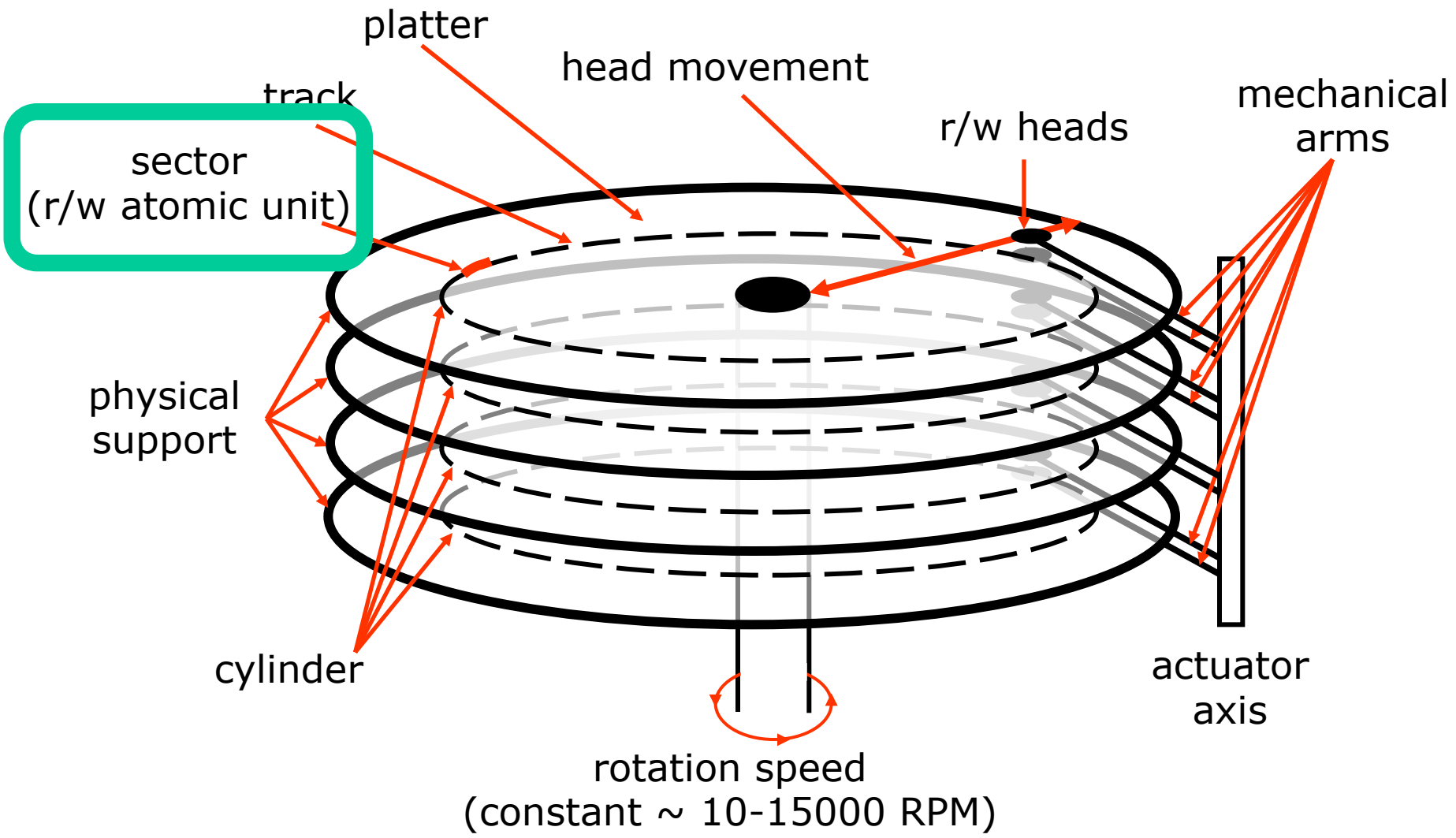
Hard Disk Drives

A hard disk drive (HDD) is a data storage using rotating disks (platters) coated with magnetic material.

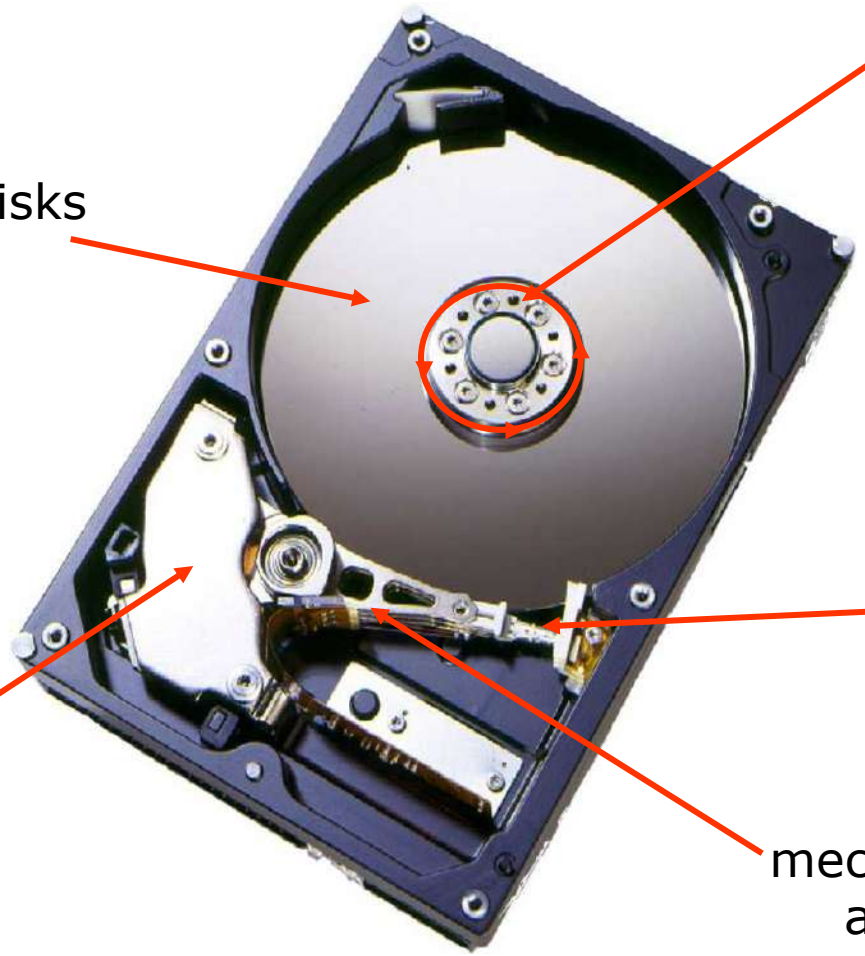
Data is read in a random-access manner, meaning individual blocks of data can be stored or retrieved in any order rather than sequentially.

An HDD consists of one or more rigid ("hard") rotating disks (platters) with magnetic heads arranged on a moving actuator arm to read and write data to the surfaces.

Magnetic disks (with many platters)



Disk unit


spindle
physical disks
r/w heads
actuator
(motor)
mechanical
arms

Disk characteristics (1/2)

diameter: about 9 cm (3,5 ÷ 2.5 in) - two surfaces

rotation speed: 7200 ÷ 15000 RPM round per minute

track density: 16,000 TPI (Track Per Inch)

sectors: 512 Byte (usually), but might be different

heads: can be parked close to the center or to the outer diameter (mobile drives)

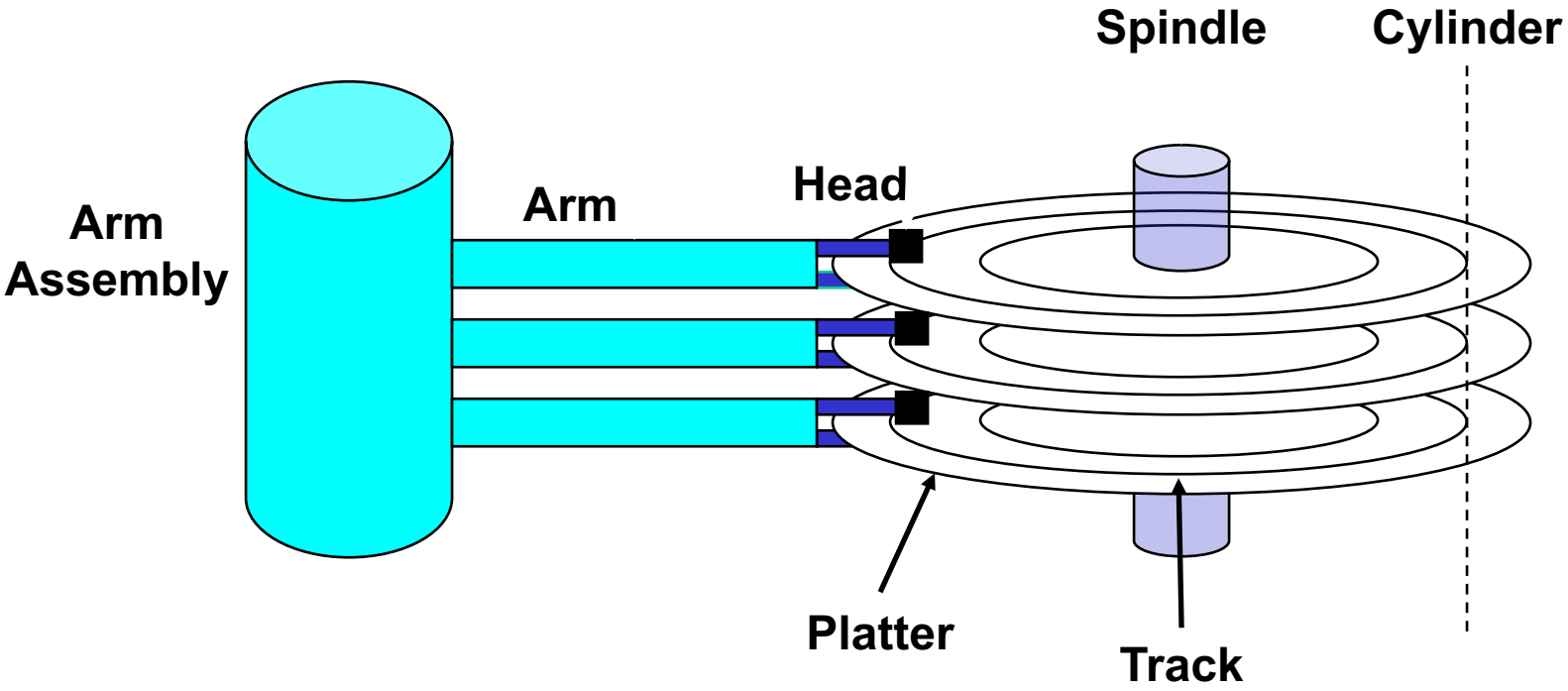
sectors are numbered sequentially, have an header and an error correction code

disk buffer cache: embedded memory in a hard disk drive that has the function of a buffer between the disk and the computer (several MB)

Disk basic characteristics

read/write *heads*

- that float on a film of air (tens of nanometers) above the platters
- one head for each magnetic platter
- *cylinder*: set of tracks with the same radius
- *seek time*: time required to reach the track that contains the data, 3÷14 ms



Some data concerning old HDD Seagate (2004)

Characteristics	Seagate ST373453	Seagate ST3200822	Seagate ST94811A
Disk diameter (inches)	2.50	3.50	3.50
Formatted data capacity (GB)	73.4	200	40.0
Cylinders	31310		
Sectors per drive	143,374,744	390,721,968 (LBA mode)	78,140,160 (LBA mode)
Number of disk surfaces (heads)	8	4	2
Rotation speed (RPM)	15,000	7200	5400
Internal disk cache size (MB)	8	8	8
External interface, bandwidth (MB/sec)	Ultra320 SCSI, 320	Serial ATA, 150	Ultra ATA, 100
Sustained transfer rate (MB/sec)	57-86	32-58	34
Minimum seek (read/write) (ms)	0.2/0.4	1.0/1.2	1.5/2.0
Average seek (read/write) (ms)	3.6/4.0	8.5/9.5	12.0/14.0
Mean time to failure (MTTF) hours	1,200,000@25 ^{o}C	600,000@25 ^{o}C	330,000@25 ^{o}C
Warranty (years)	5	3	-
Nonrecoverable read error per bit read	< 1 per 10^{15}	< 1 per 10^{14}	< 1 per 10^{14}
Price in 2004 ($/GB)	$5	$0.5	$2.5

LBA and CHS

Sectors are usually grouped by the O.S.

Clusters are normally a power-of-two multiple of the sector size.

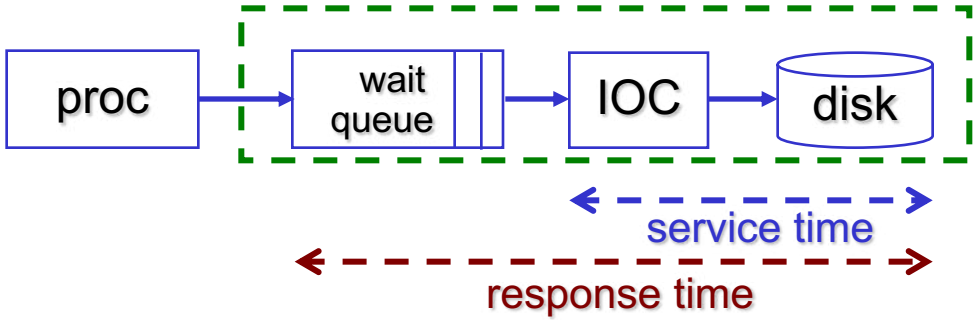
Each sector on a HDD is identified by three numbers (CHS):

- Cylinder
- Head
- Sector

A function in the BIOS of the PC converts the LBA index into the three CHS numbers that identify the sector on the HDD.

HDD disks performance

Disk service time and response time



- service time s_{disk}:

  seek time + rotational latency + data transfer time + controller overhead

  - *seek* time: head movement time (≈ ms), is a function of the number of cylinders traversed
  - *latency* time: time to wait for the sector (≈ ms, ½ round)
  - *transfer* time: is a function of rotation speed, storing density, cylinder position (≈ MB/sec)
  - *controller overhead :* buffer management (data transfer) and interrupt sending time

Disk response time

response time R: time spent in queue waiting for the resource (that is busy executing other requests) and for the execution of the request

- is a function of the:
  - number of requests in the waiting queue (queue lenght)
  - utilization level of the resource
  - mean disk service time
  - variance of disk service time (distribution)
  - arrival rate and inter-arrival times distribution

assuming the same mean values of the metrics, the response time increases as the variability increases

In this course, ***we will not see how to evaluate response time****.*

Exercise 1: mean service time of a I/O operation

read/write of a sector of 512 Byte = 0.5 KB

data transfer rate: 50 MB/sec

rotation speed: 10000 RPM (round per minute)

mean seek time: 6ms

overhead controller: 0.2ms

mean latency: (60s/min)x1000/(2x10000 rpm) = 3.0ms (time for ½ round)

transfer time: (0.5KB)x1000 / (50x1024KB/s) = 0.01ms

mean I/O service time = 6ms + 3ms + 0.01ms + 0.2ms = 9.21ms
seek latency transfer controller

The previous service times considers only the very pessimistic case where sectors are fragmented on the disk in worst possible way.

This is true if the files are very small (each file is contained in one block) or the disk is very (externally) fragmented.

In many circumstances, this is not the case: files are larger than one block, and they are stored in a contiguous way to decrease the needs of seek and rotational latency times.

Suppose we can measure the *data locality* of a disk as the percentage of blocks that do not need seek or rotational latency to be found.

Exercise 2: data locality (see the values of exercise 1)

- locality: the effect is that the seek time and transfer times are zero
  - data locality l=75%: seek+latency affect only 25% of the operations
  - T = (1 – l) * (Ts + Tl) + Tc + Tt

    (6.0 + (0.5 × 60 × 10^3 / 10000)) x 0.25 + (0.5 KB / 50MB × 2^{10}) + 0.2
  - mean time of one I/O op. = (0.25 × (6+3)) + 0.01 + 0.2 = 2.46 ms

    seek+latency **transfer** **controller**

Exercise 2b: data locality with multiple blocks

- Two ways for consider the effect of locality to read/write a file (many blocks):
  1. consider the effect of locality in the mean service time and then multiply this value with the number of I/O operations required to r/w the entire file, i.e., **total = #_of_blocks * mean_time_one_IO**

     E.g., 100K blocks
     Total_time = 100.000 * 2.46 = 246.000 ms

  2. multiply the number of I/O operations affected by locality with the service time computed with no seek and latency times, multiply the remaining number of operations by the complete service time, add the two values i.e., **total = #_of_blocks * I * (transfer + controller) + #_of_blocks * (1- I) * (transfer + controller+seek+latency)**

     E.g., 100K blocks
     Total_time = 75.000 * 0.21 + 25.000 * 9.21 = 246.000 ms

Exercise 3: influence of “not optimal” data allocation

time to transfer a file of 1MB

Case A (locality = 100%):

- 1 initial seek: 6 ms
- 1 latency: 3 ms
- 1 global transfer 1 MB: (1/50) × 1000 = 20 ms
- total time: 29 ms

Case B (locality = 0%): (10 blocks of 1/10 MB “not well” distributed on disk) for each block:

- 1 seek: 6 ms
- 1 latency: 3 ms
- 1 partial transfer: 2 ms
- total time: (6 + 3 + 2) × 10 = 110 ms (+ 279% !)

(controller times not considered)

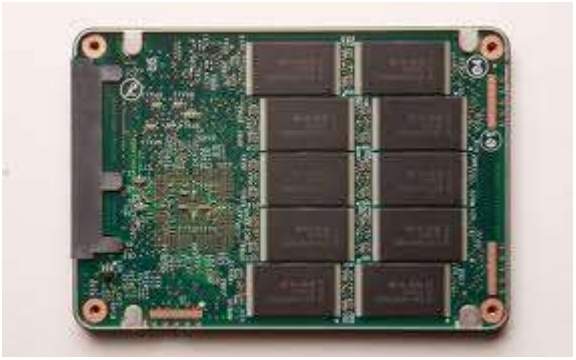
Solid state disks

Solid state disks (SSD):

A solid state drive is a non-volatile storage device.

A controller is included in the device with one or more solid state memory components.

The device should use traditional hard disk drive (HDD) interfaces (protocol and physical) and form factors.

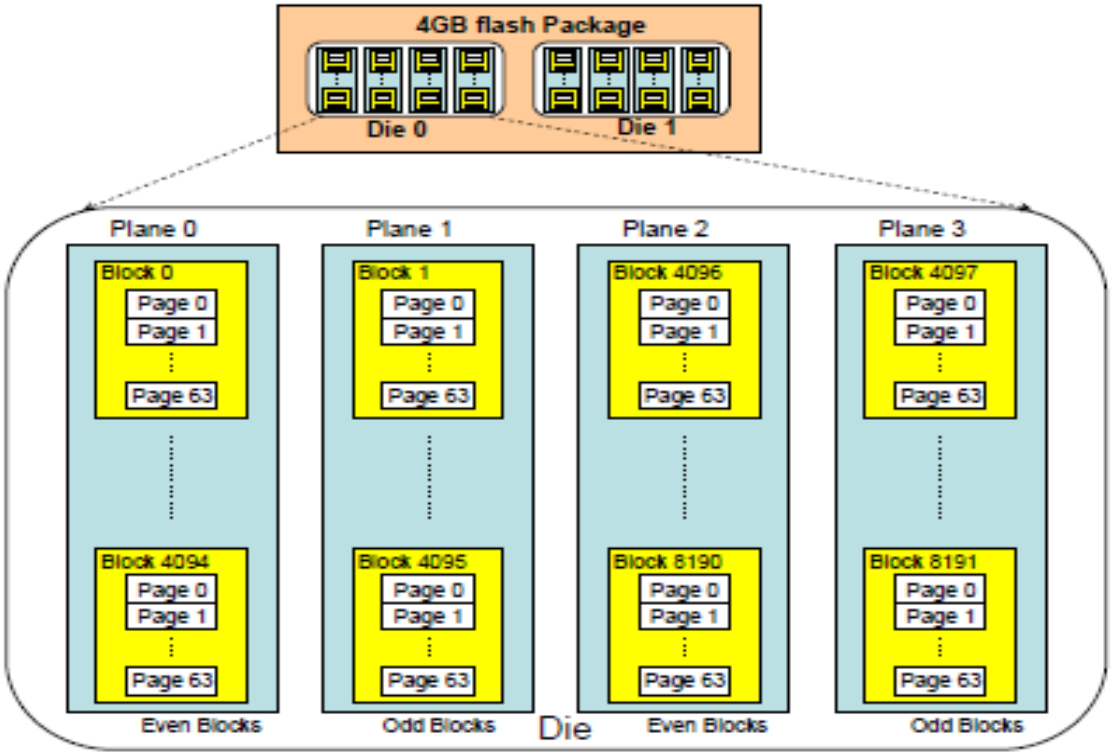


HDD vs SSD

Platters are replaced
by NAND based memories

SSD: Internal organization (1)

- NAND flash is organized into **Pages** and **Blocks**
  - A **page** may contain eight 512-byte logical block addresses (LBAs)
  - A **block** typically consists of 64 pages with a total capacity of 256KB



SSD: Internal organization (1)

- **Pages**: smallest unit that can be read/written. In particular: a sub-unit of an erase block consisting of a number of bytes which can be read from and written to in single operations, through the loading or unloading of a page buffer and the issuance of a program or read command.
- **Blocks** (or Erase Block): smallest unit that can be erased, typically consisting of multiple pages.

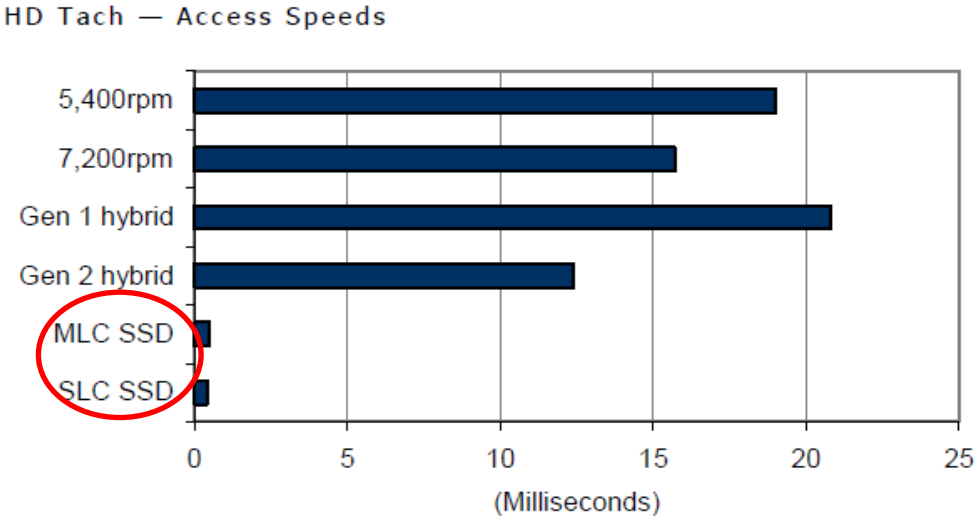
- Pages can be in **three states**:
  - Empty: they do not contain data.
  - Dirty: they contain data, but this data is no longer in use.
  - In use: the page contains data that can be actually read

Page states – writing and erasing

- **Only empty pages can be written**
- **Only dirty pages can be erased, but this must be done at the block level** (all the pages in the block must be dirty or empty)
- **It is meaningful to read only pages in the "in use" state.**
- **If no empty page exists, some dirty page must be erased.**
  - If no block containing just dirty or empty pages exists, then special procedures should be followed to gather empty pages over the disk.
  - To erase the value in flash memory the original voltage must be reset to neutral before a new voltage can be applied, known as write amplification.

Remark: we can write and read a single page of data from a SSD but we have to delete an entire block to release it

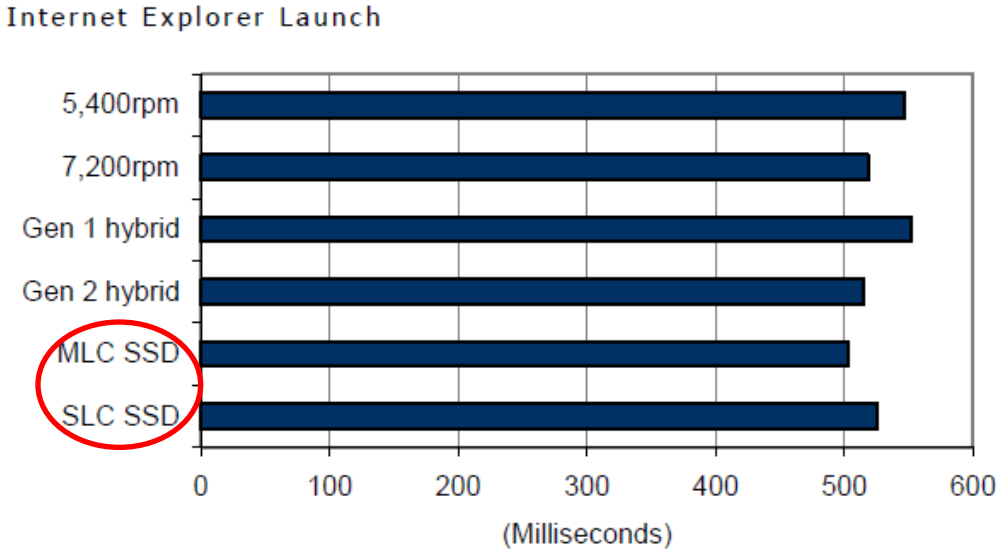
SSD: Performances: from theory...to practice



Access speed
measured at device level in a controlled laboratory environment
HDD vs SSD
~ 15-20x improvement

Same device integrated into a notebook PC with a real application test
~ same performace !!!

Why such behavior ?

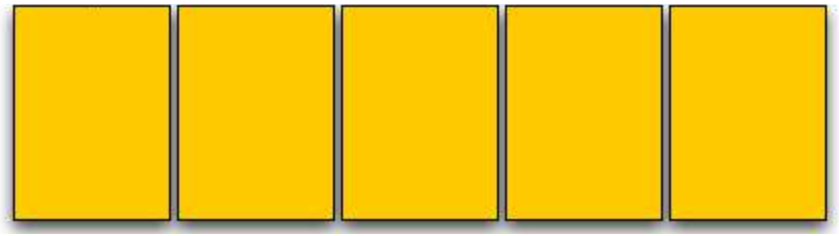


Source: Exposing the Strengths and Weaknesses of HDDs, SSDs, IDC and Hybrids, IDC, 2008

Example:

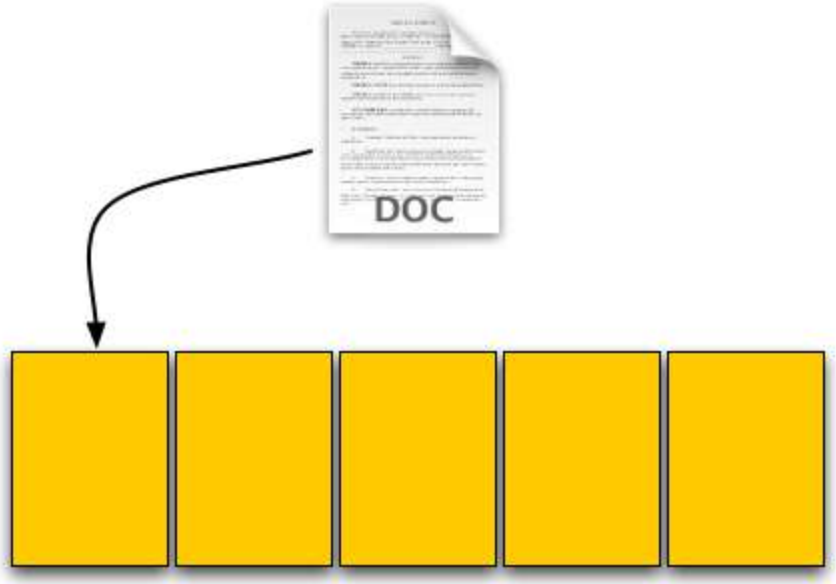
- Hypothetical SSD:

Page Size:	4KB
Block Size:	5 Pages
Drive Size:	1 Block
Read Speed:	2 KB/s
Write Speed:	1 KB/s



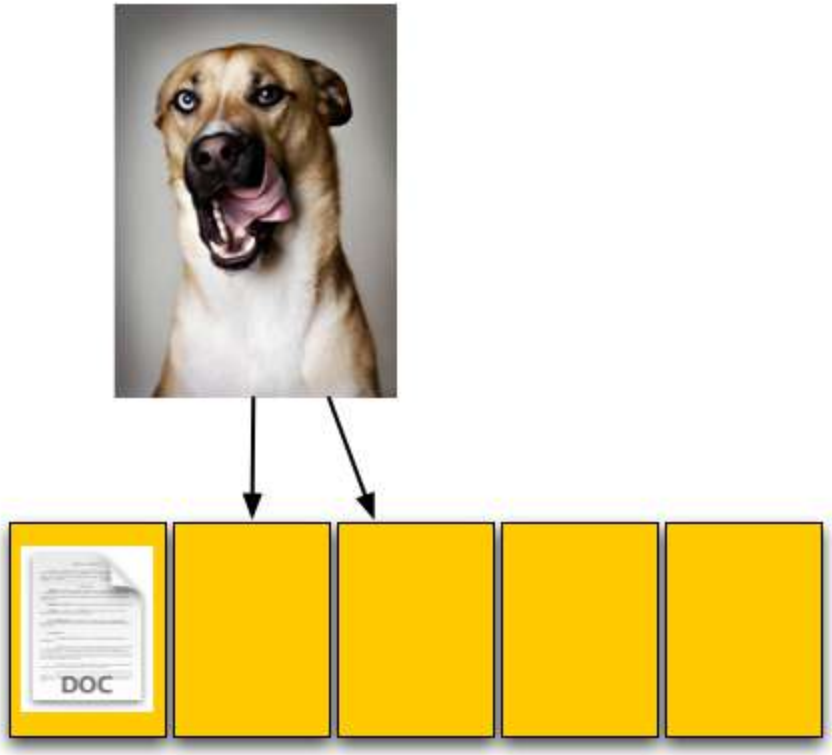
Example:

Lets write a 4Kb text file to the brand new SSD.



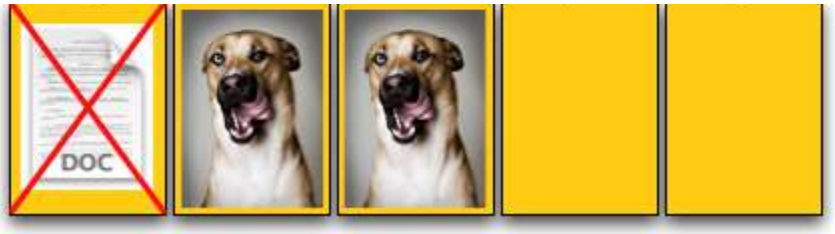
Example:

Now lets write a 8Kb pic file to the almost brand new SSD, thankfully there's space.



Example Cont.

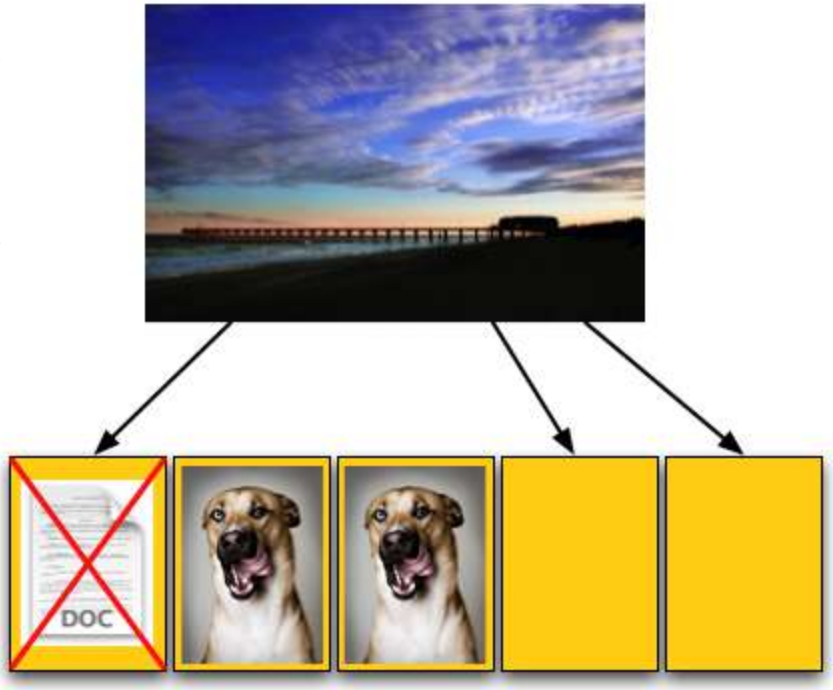
Now let’s delete the txt file in the first page.

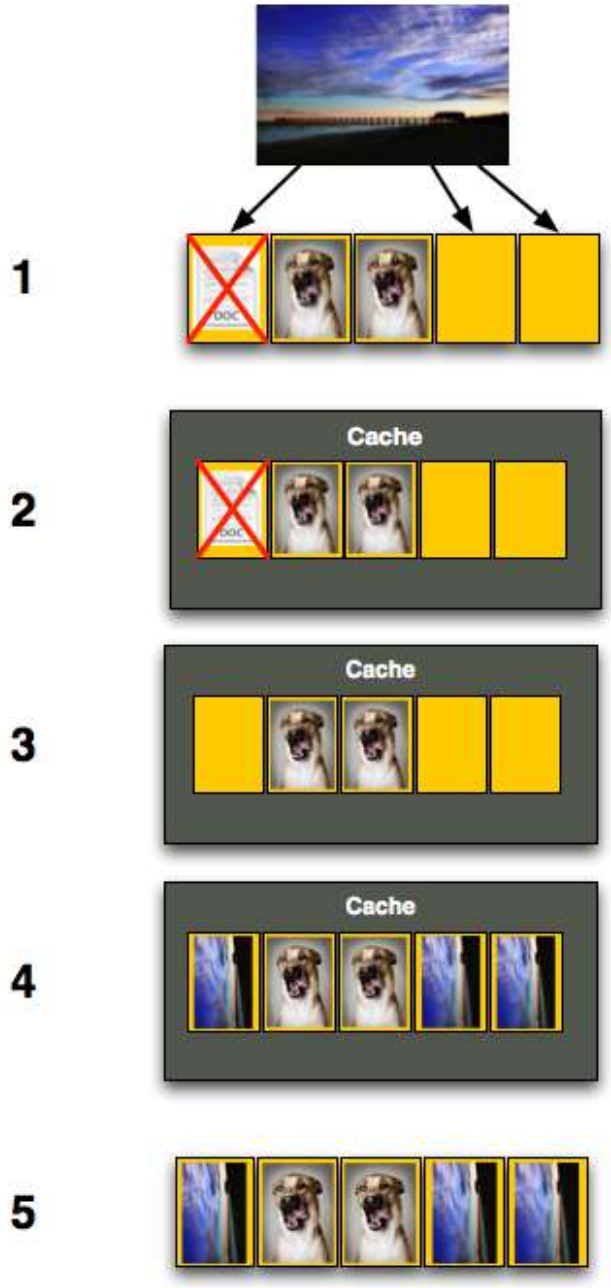


Example Cont.

Finally lets write a 12kb pic to the SSD. How long should it take? We expect to rely on a 1 kb/s write speed (so 12s to the writing) …. but …

The OS is told there are 3 free pages on the SSD when there are only 2 “empty”.



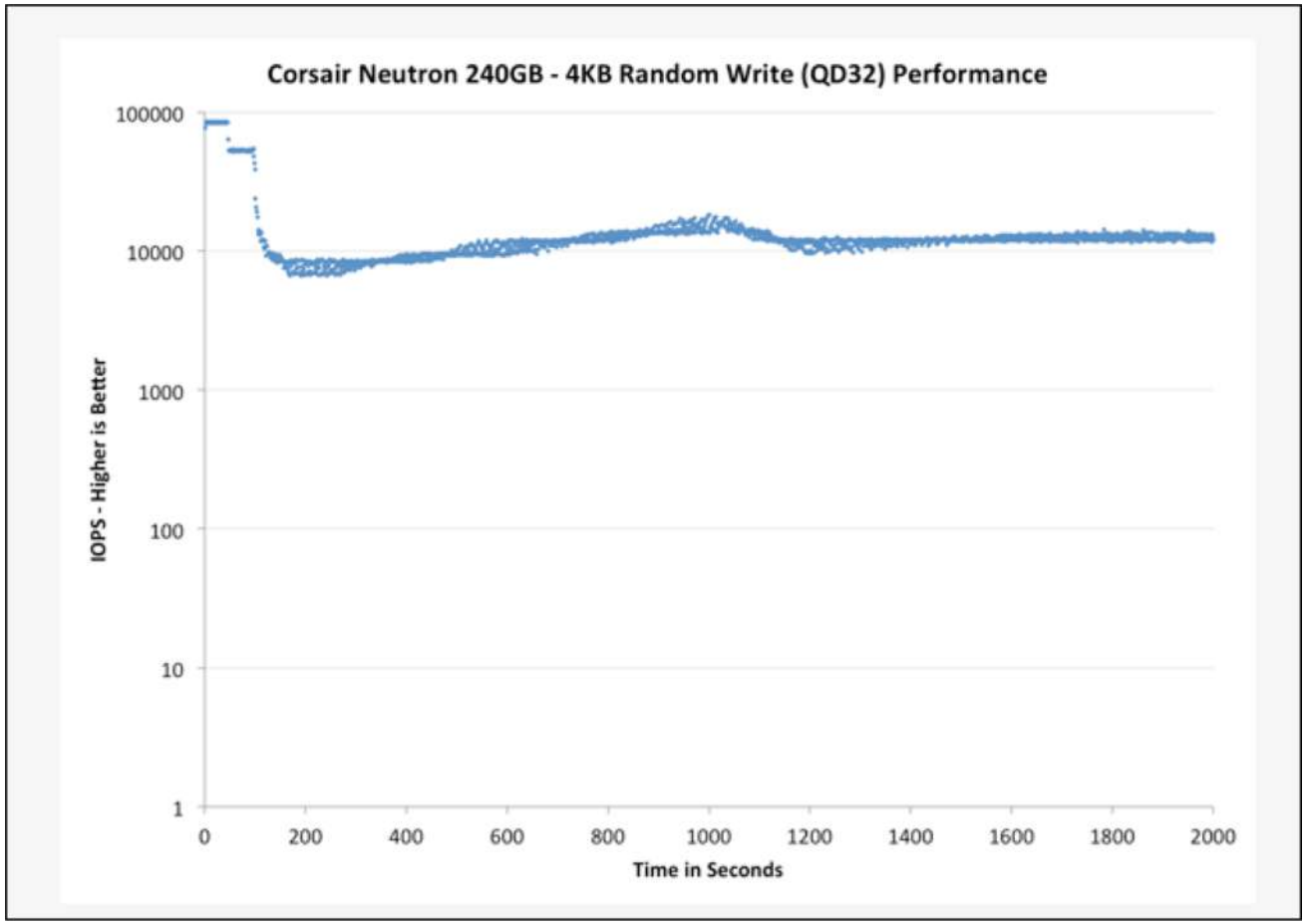


- Step 1: Read block into cache
- Step 2: Delete page from cache
- Step 3: Write new pic into cache
- Step 4: Delete the old block on SSD
- Step 5: Write cache to SSD

- The OS thought it was writing 12 KBs of data when in fact the SSD had to read 12 KBs and then write 20KBs, the entire block.
- The writing should have taken 12 secs but actually took 26 seconds, resulting in a write speed of 0.46KB/s not 1KB/s

Performance degradation

Write amplification degrades a lot the performance of an SSD as time passes:

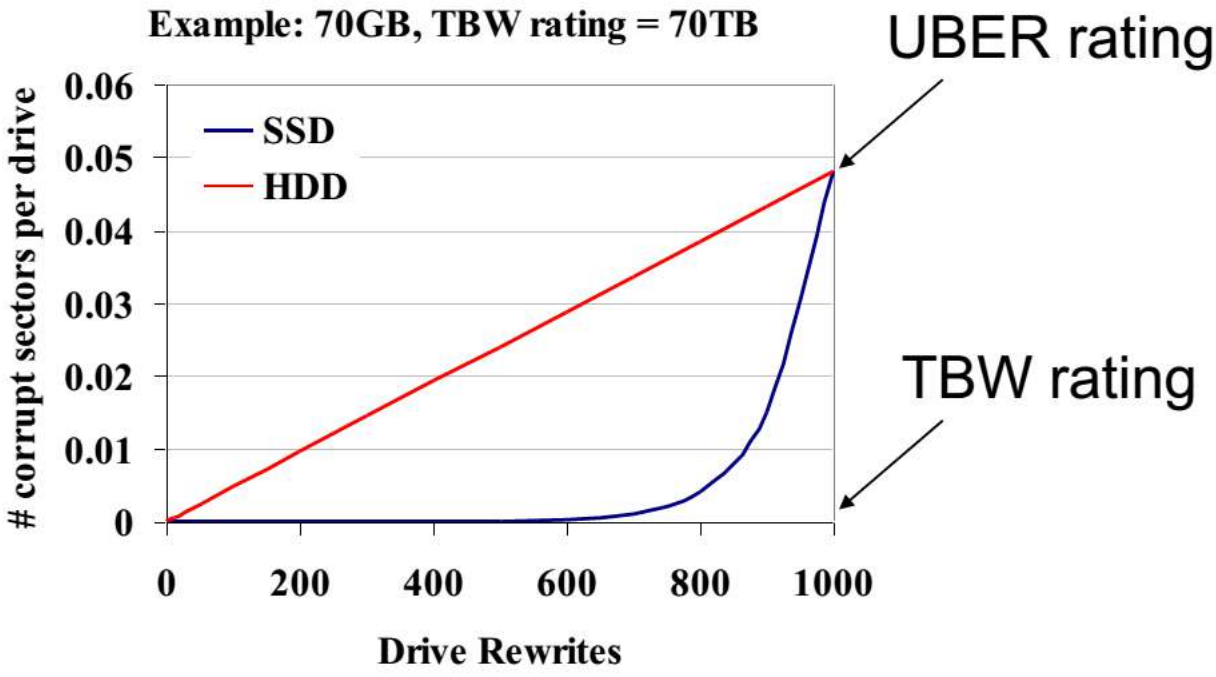


Solid State Disks (SSD) – drawbacks

- They cost more than the conventional HDD
- Flash memory can be written only a limited number of times (wear):
  - have a shorter lifetime
  - error correcting codes
  - over-provisioning (add some spare capacity)
- different read/write speed.
- write performance degrades of one order of magnitude after the first writing
- often the controller become the real bottleneck to the transfer rate
- SSD are not affected by data-locality and must not be defragmented (actually, *defragmentation may damage the disks*).

HDD vs SSD

- **Unrecoverable Bit Error Ratio (UBER):** A metric for the rate of occurrence of data errors, equal to the number of data errors per bits read
- **Endurance rating:** Terabytes Written (TBW is the total amount of data that can be written into an SSD before it is likely to fail). The number of terabytes that may be written to the SSD while still meeting the requirements.



Capacity and TBW

- **Memory cells can accept data recording between 3,000 and 100,000 during its lifetime**
  - Once the limit value is exceeded, the cell "forgets" any new data
- **A typical TBW for a 250 GB SSD is between 60 and 150 Terabytes of data written to the drive.**
  - This means that in order to overcome, for example, a TBW of 70 Terabytes, a user should write 190 GB every day for a year or fill his SSD on a daily basis for two thirds with new files for a whole year
- **It is difficult to comment on the duration of SSDs.**
  - Dell, in an old study (2011), spoke of an estimated duration between three months and ten years explaining, however, that there are so many factors (temperature and workload) that may depend on the life of an SSD that is very difficult to make predictions.

Some data concerning SSD Seagate (Feb.2011)

Specifications	200GB1	100GB1	50GB1
Interface	SATA 3Gb/s	SATA 3Gb/s	SATA 3Gb/s
NAND Flash Type	SLC	SLC	SLC
Storage Type	SSD	SSD	SSD
Guaranteed Logical Blocks	390,721,968	195,371,568	97,696,368
Sustained Data Trans. Rate (Mb/s)	240,000	240,000	240,000
I/O Data Trans. Rate, Max (MB/s)	300	300	300
Sequential Command Rate (MB/s) r/w Peak, r/w Sustained, 128KB	240/220 200/100	240/220 200/60	240/220 200/30
Random Command Rate (IOPS) r/w Peak, r/w Sustained, 4KB	30,000/25,000 25,000/5200	30,000/25,000 25,000/5200	30,000/25,000 25,000/2600
Unrecoverable **R**ead **E**rrors per Bits Read New, Max EOL, Max	1 Logical Block Addressing scheme (LBA) per 10E16 1 LBA per 10E15	1 LBA per 10E16 1 LBA per 10E15	1 LBA per 10E16 1 LBA per 10E15

Hybrid solutions (HDD + SSD)

Some large storage servers use SSD as a cache for several HDD. Some mainboards of the latest generation have the same feature: the combine a small SSD with a large HDD to have a faster disk.

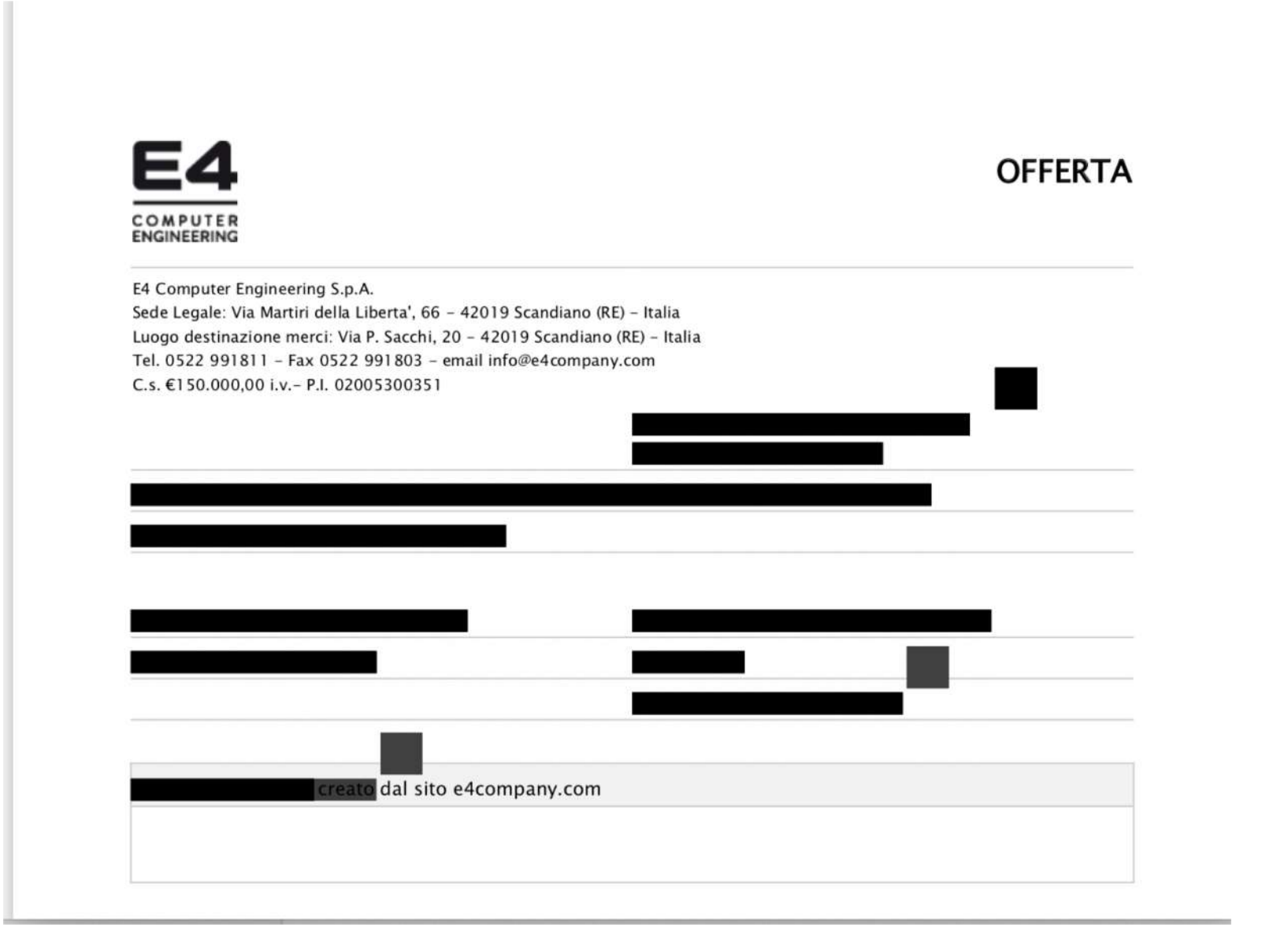
Some HDD manufacturers produce Solid State *Hybrid Disks* (SSHD) that combine a small SDD with a large HDD in a single unit.





A server for a datacenter: an example

An example of Server for a Datacenter



E4
COMPUTER
ENGINEERING

OFFERTA

E4 Computer Engineering S.p.A.
Sede Legale: Via Martiri della Liberta', 66 – 42019 Scandiano (RE) – Italia
Luogo destinazione merci: Via P. Sacchi, 20 – 42019 Scandiano (RE) – Italia
Tel. 0522 991811 – Fax 0522 991803 – email info@e4company.com
C.s. €150.000,00 i.v.– P.I. 02005300351

[redacted] creato dal sito e4company.com

An example of Server for a Datacenter

Descrizione	Prezzo Unit.	Q.tà	Prezzo Tot.
RB120: Server 1U Dual Socket Intel GPU - 2 bays SAS/SATA + 2 internal	€ 9.470,00	1	€ 9.470,00

1 x 1U - 2 x SAS/SATA - Ridondante 2000W
1U Rackmount Black Chassis. 2000W Redundant Power Supplies. 43mm (H) x 437mm (W) x 894mm (D). N. 2 Hot-swap 2.5" SAS/SATA drive bays, n. 2 Internal 2.5" drive bays.

1 x Dual Xeon Scalable - C621 - Server GPU
Proprietary Motherboard. Intel® C621 chipset. Dual Socket P (FCLGA3647). Support up to 165W TDP. N. 12 DIMM Slots supported Memory Types: 2666/2400/2133MHz RDIMM, LRDIMM and 3DS ECC LRDIMM modules. Optimal memory configuration: Six memory channels per CPU.

An example of Server for a Datacenter

E4
COMPUTER
ENGINEERING

OFFERTA

2 x Xeon 8-Core 4110 2,1Ghz 11MB
Intel® Xeon® Silver 4110 Processor. 8Cores. 16Threads. FCLGA3647 Socket.11MB L3. 2,1Ghz Base Frequency. 85W max. TDP. DDR4-2400 Memory type.

6 x DDR4-2666 Reg. ECC 16 GB module
Full brand memory, tested and certified by manufacturer for thorough compatibility with proposed system. The real operating speed depends on the processor's model and on the number of the installed modules. Better performances are achieved through a proper channel configuration.

1 x Intel C621 SATA III 4 ports #

1 x SEAGATE 2TB 2,5" SATA III 7.200RPM
Seagate Enterprise Capacity hard disk drive. Form factor: 2,5". Capacity: 2TB. Interface: 512N SATA 6Gb/s. Buffer: 128MB. Rotational Speed: 7200RPM. Max. Sustained Transfer Rate (MB/s): 136MB/. 2Million-hour MTBF.

1 x Intel S4500 240GB 2,5" SSD SATA III
Intel® SSD DC S4500 Series. Sequential Read (up to): 500MB/s. Sequential Write (up to): 190MB/s. Random Read (100% Span): 69000 IOPS. Random Write (100% Span): 16000 IOPS. Endurance Rating (Lifetime Writes): 0.62 PBW.

1 x Backplane 2 bays SAS/SATA

An example of Server for a Datacenter

E4
COMPUTER
ENGINEERING

OFFERTA

2 x Xeon 8-Core 4110 2,1Ghz 11MB
Intel® Xeon® Silver 4110 Processor. 8Cores. 16Threads. FCLGA3647 Socket.11MB L3. 2,1Ghz Base Frequency. 85W max. TDP. DDR4-2400 Memory type.

6 x DDR4-2666 Reg. ECC 16 GB module
Full brand memory, tested and certified by manufacturer for thorough compatibility with proposed system. The real operating speed depends on the processor's model and on the number of the installed modules. Better performances are achieved through a proper channel configuration.

1 x Intel C621 SATA III 4 ports #

1 x SEAGATE 2TB 2,5" SATA III 7.200RPM
Seagate Enterprise Capacity hard disk drive. Form factor: 2,5''. Capacity: 2TB. Interface: 512N SATA 6Gb/s. Buffer: 128MB. Rotational Speed: 7200RPM. Max. Sustained Transfer Rate (MB/s): 136MB/. 2Million-hour MTBF.

1 x Intel S4500 240GB 2,5" SSD SATA III
Intel® SSD DC S4500 Series. Sequential Read (up to): 500MB/s. Sequential Write (up to): 190MB/s. Random Read (100% Span): 69000 IOPS. Random Write (100% Span): 16000 IOPS. Endurance Rating (Lifetime Writes): 0.62 PBW.

1 x Backplane 2 bays SAS/SATA

Computing Infrastructure





RAID disks

Redundant Arrays of Independent (Inexpensive) Disks

disk arrays: proposed in the 1980's

need to increase the performance, the size and the reliability of storage systems

several *independent* disks that are considered as a single, large, high-performance logical disk

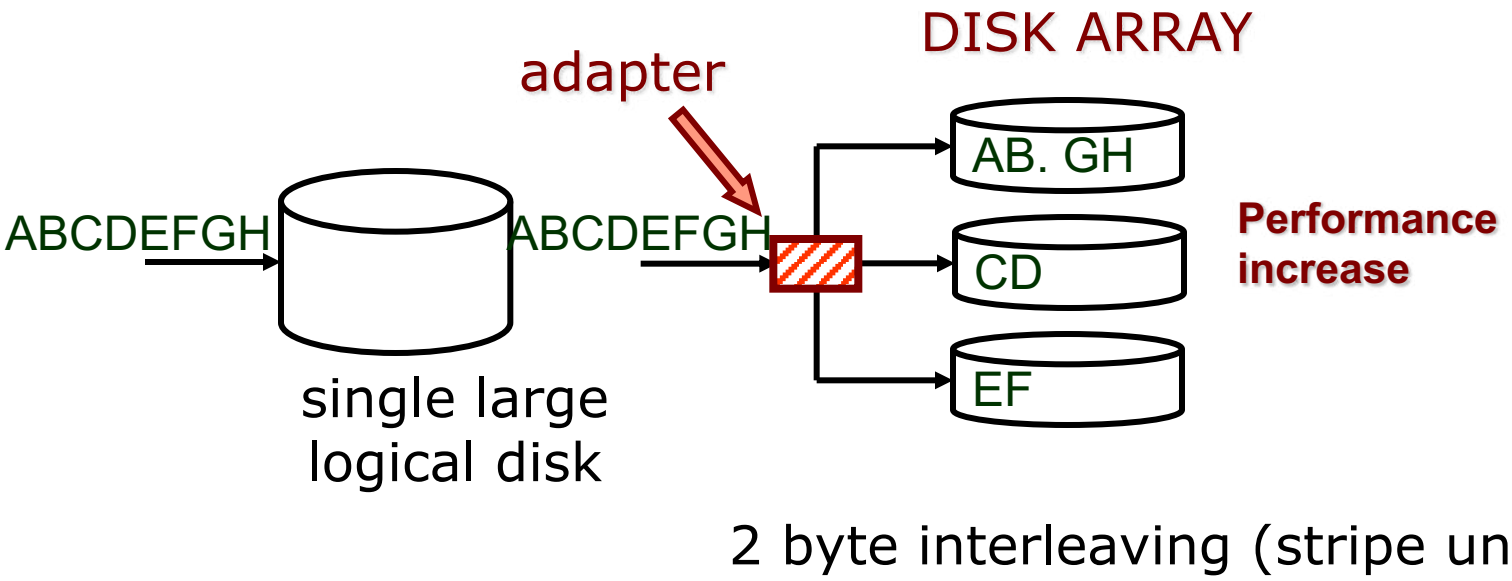
the data are *striped* across several disks accessed in parallel:

- high data transfer rate: large data accesses (heavy I/O op.)
- high I/O rate: small but frequent data accesses (light I/O op.)
- load balancing across the disks

Two orthogonal techniques:

- data striping: to improve performance
- redundancy: to improve reliability

parallelism in a I/O operation (a simple example)



- *I/O virtualization*
  - data are distributed transparently over the disks (no action is required to the users)

data striping

- striping: data are written sequentially (a vector, a file, a table, ...) in units (stripe unit: bit, byte, blocks) on multiple disks according to a cyclic algorithm (*round robin*)
- stripe unit: dimension of the unit of data that are written on a single disk
- stripe width: number of disks considered by the striping algorithm

1. multiple independent I/O requests will be executed in parallel by several disks decreasing the queue length (and time) of the disks
2. single multiple-block I/O requests will be executed by multiple disks in parallel increasing of the transfer rate of a single request

parallelism and reliability

the more physical disks in the array

the larger the size and performance gains

but ... ...

the larger the probability of failure of a disk

⇓

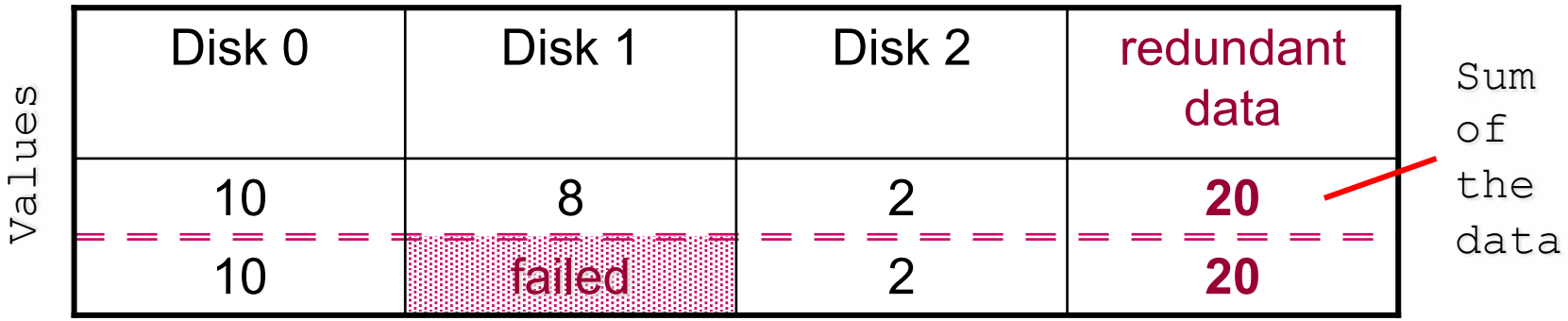
this is the main motivation for the introduction of

redundancy

overall reliability decreases with redundancy

- The probability of a failure (assuming independent failures) in an array of 100 disks is 100 higher the probability of a failure of a single disk
  - if a disk has an Mean Time To Failure (MTTF) of 200,000 hours (~23 years) an array of 100 disks will show a MTTF of 2000 hours (~ 3 months)
- Redundancy: error correcting codes (stored on disks different from the ones with the data) are computed to tolerate loss due to disk failures
- Since write operations must update also the redundant information, their performance is worse than the one of the traditional writes

example (elementary) of a data recontruction



failed = 20 – (10 + 2) = 8

RAID architectures

RAID levels

RAID 0 striping only

RAID 1 mirroring only

- RAID 0+1 (nested levels)
- RAID 1+0 (nested levels)

RAID 2 bit interleaving (not used)

RAID 3 byte interleaving - redundancy (parity disk)

RAID 4 block interleaving - redundancy (parity disk)

RAID 5 block interleaving - redundancy (parity block distributed) – highly utilized

- RAID 5+0 (nested levels)

RAID 6 greater redundancy (2 failed disks are tolerated)

RAID 7 (proprietary solutions)

RAID level 0: *striping, no redundancy*

data are written on a single logical disk and splitted in several blocks distributed across the disks according to a striping algorithm

- used where performance and capacity, rather than reliability, are the primary concerns, minimum two drives required

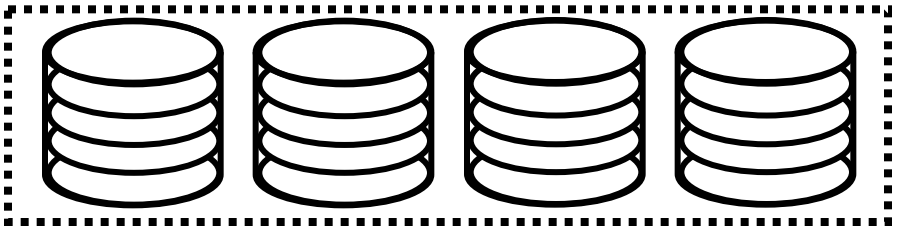
\+ lowest cost because it does not employ redundancy (no error-correcting codes are computed and stored)

\+ best write performance (it does not need to update redundant data and it is parallelized)

– single disk failure will result in data loss

disk array

data



user capacity of 4 physical disks configured as a single logical disk RAID 0

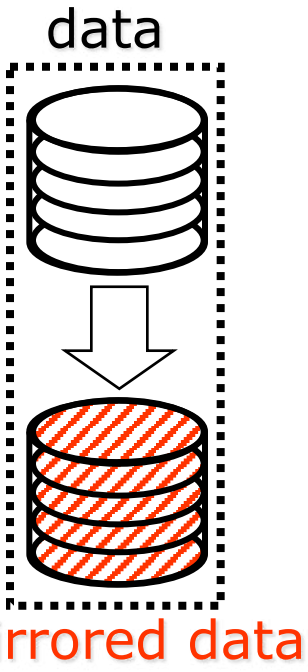
used by the servers with read-only data

	Disk **0**	Disk 1	Disk 2	Disk 3	
Stripe **1**	*Block 1*	*Block 2*	*Block 3*	*Block 4*	
Stripe **2**	*Block 5*	*Block 6*	*Block 7*	*Block 8*	

RAID level 1: *mirroring*

whenever data is written to a disk it is also duplicated (mirrored) to a second disk (there are always two copies of the data), minimum 2 disk drives

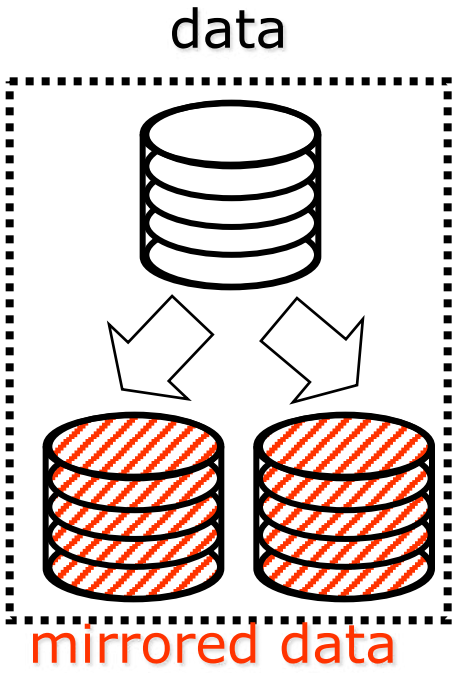
+ high reliability: when a disk fails the second copy is used
+ read of a data: it can be retrieved from the disk with the shorter queueing, seek, and latency delays
+ fast writes (no error correcting code should be computed) – but still slower than standard disks (due to duplication) and much slower than RAID 0
– high costs (50% of the capacity is used)



Disk **0**	Disk 1
Block 1	*Block 1*
Block 2	*Block 2*
Block 3	*Block 3*

RAID level 1: *mirroring (Theoretical)*

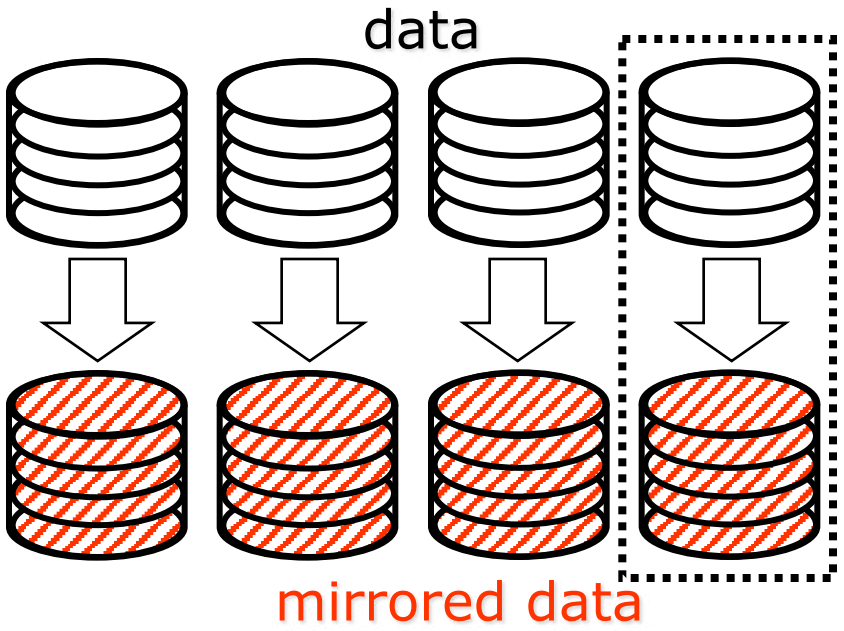
- In principle, a RAID 1 can mirror the content over more than one disk.
  - This give resiliency to errors even if more than one disk breaks
  - It allows with a *voting mechanism* to identify errors not reported by the disk controller.
- In practice this is never used, because the overhead and costs are too high.



RAID level 1: *mirroring – using more disks*

- However if several disks are available (always in an even number), disks could be coupled.
- The total capacity is halved.
- Each disk has a mirror.
- How to organize this? RAID levels can be combined
  - Raid 0 + 1
  - Raid 1 + 0

user capacity of 4 disks (for data)

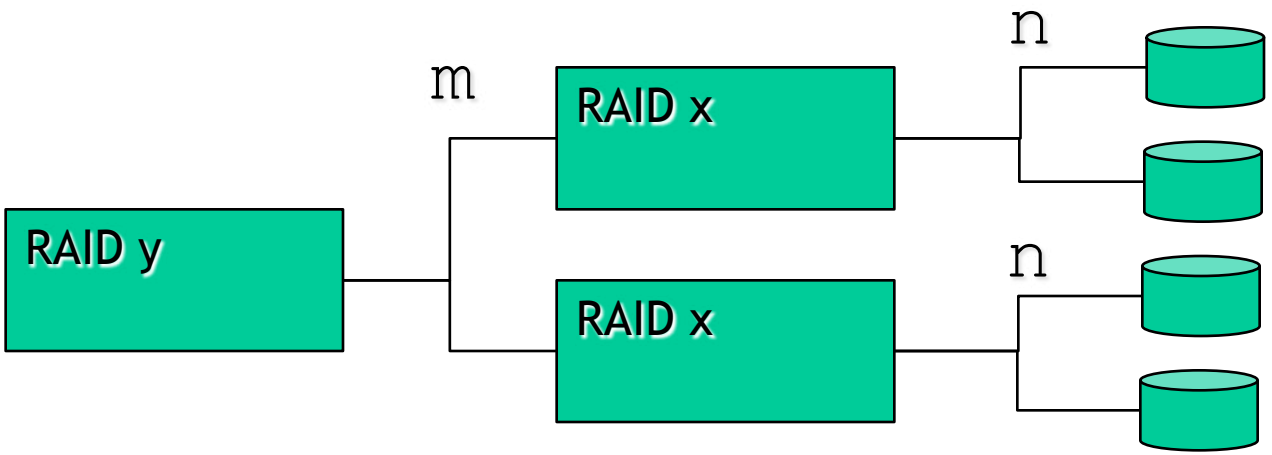


RAID levels (combined)

RAID levels can be combined

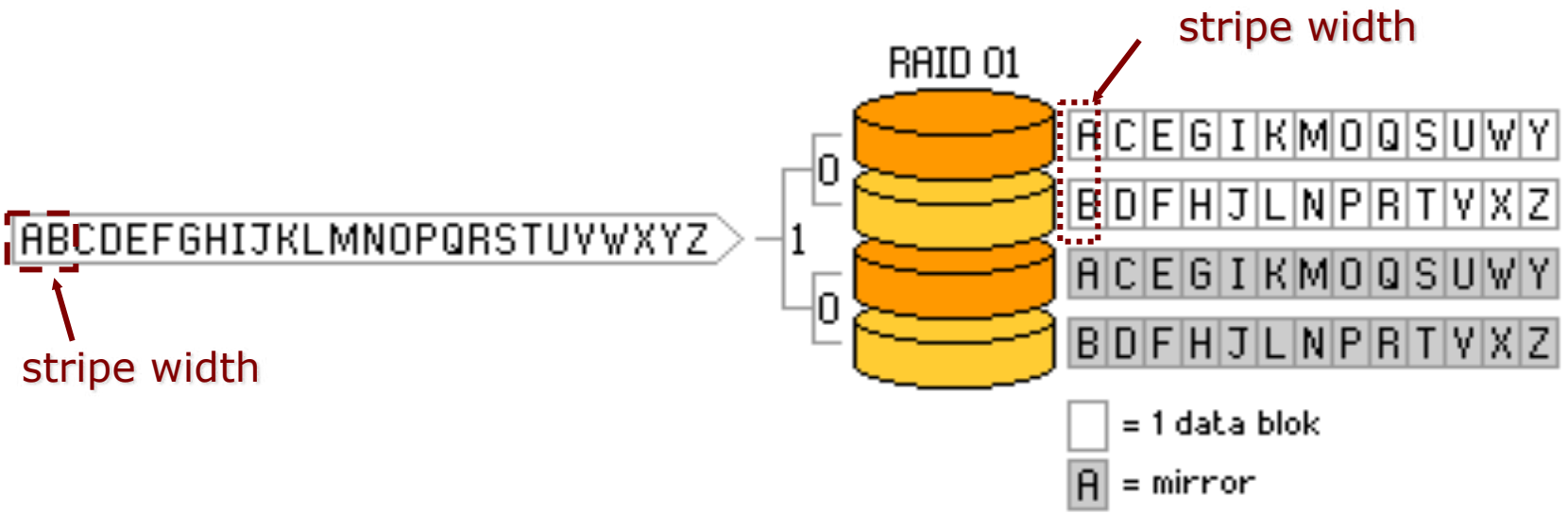
RAID *x* + *y* (or RAID *xy*) =>

- *n* x *m* disks in total
- Consider *m* groups of *n* disks
- Apply RAID *x* to each group of *n* disks
- Apply RAID *y* considering the *m* groups as single disks



RAID level 0 + 1

striping first (RAID 0)
then mirroring (RAID 1)

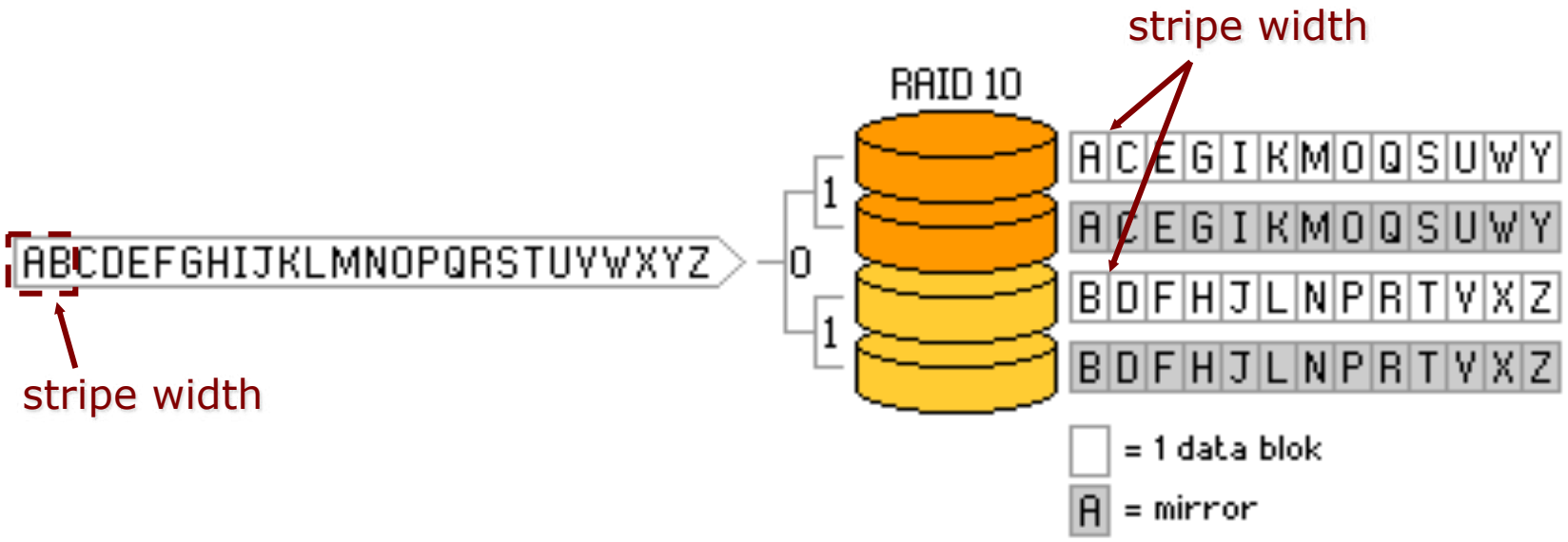


- minimum 4 drives
- after the first failure the model becomes as a RAID 0

RAID level 1 + 0

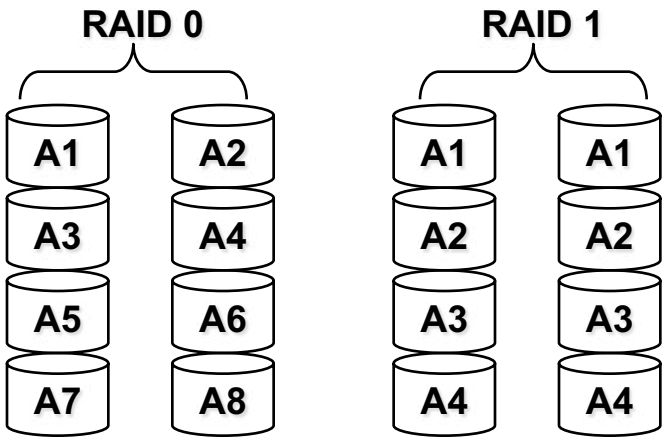
mirroring first (RAID 1)

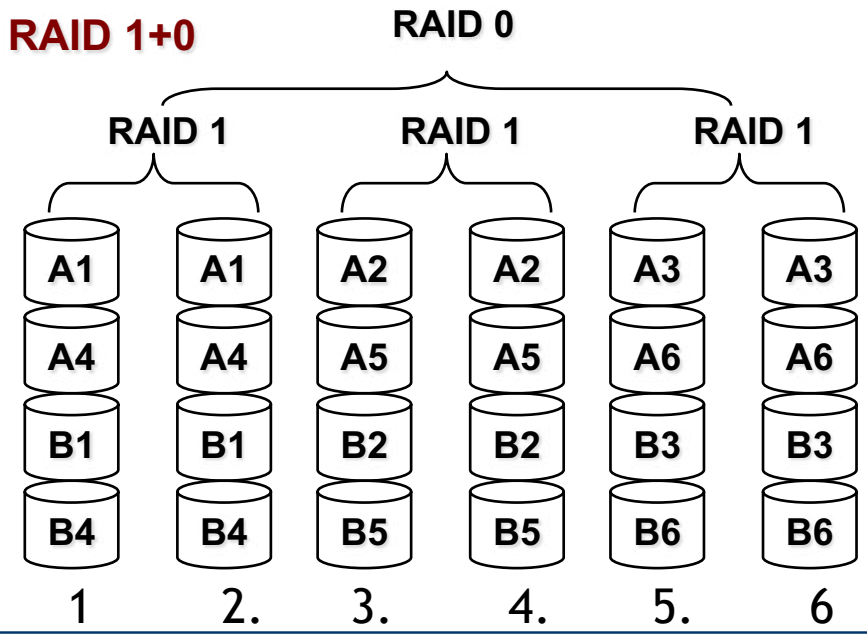
then striping (RAID 0)

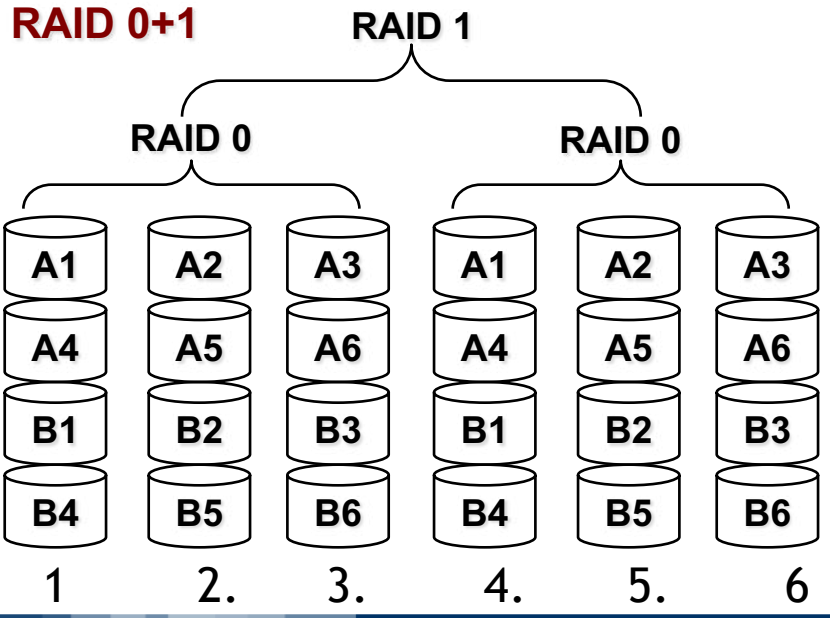


- minimum 4 drives are required
- used in databases with very high workload (fast writes)

RAID 0, 1, 0+1 and 1+0 organizations

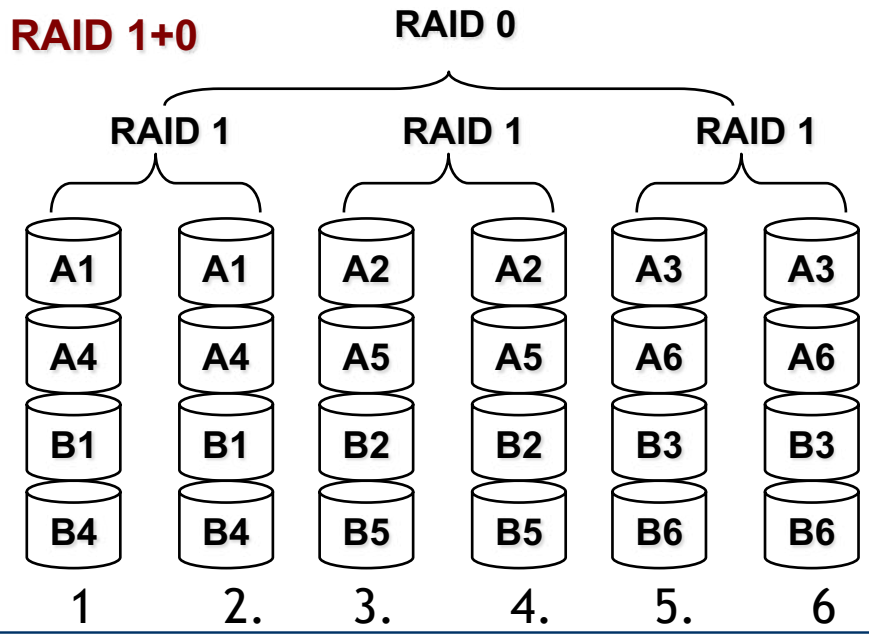


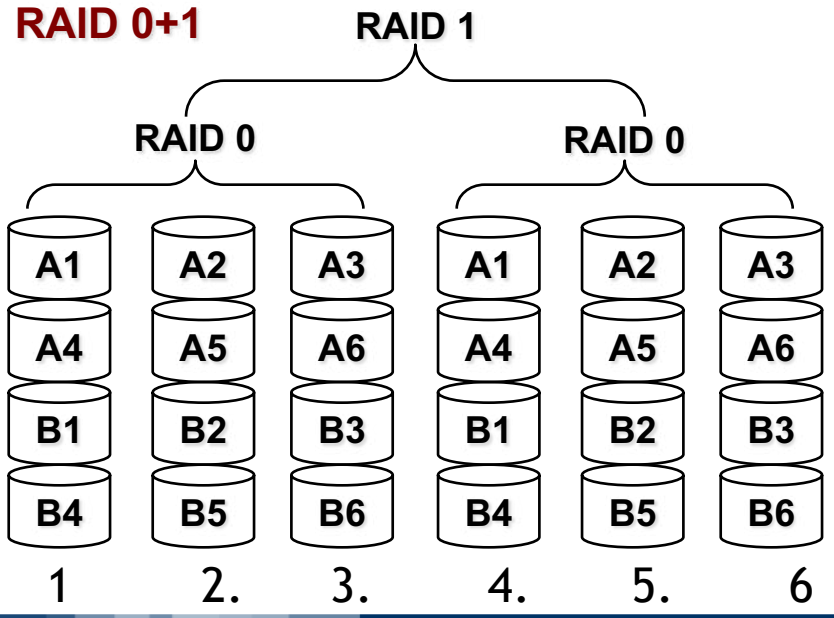




RAID 0, 1, 0+1 and 1+0 organizations

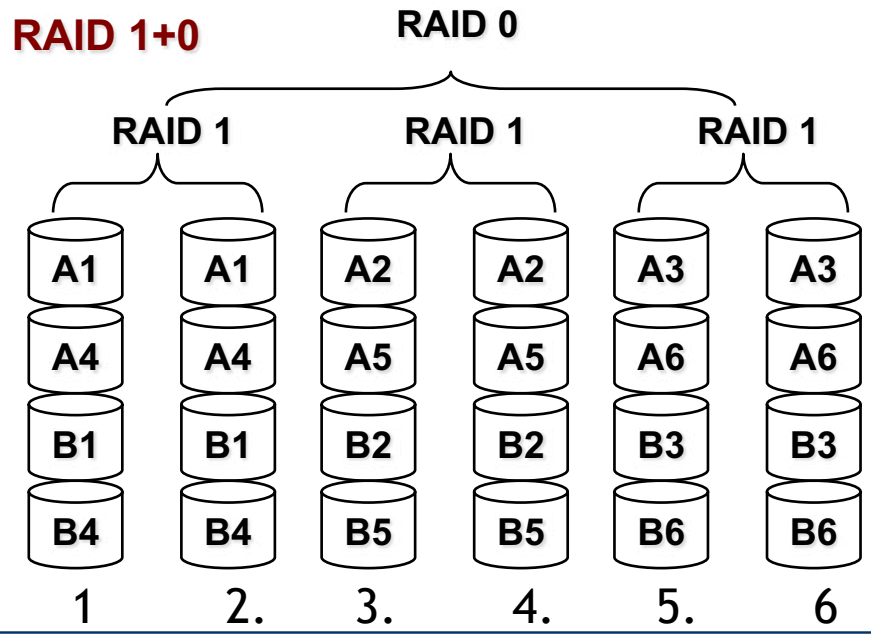
- The blocks are the same but are allocated in a different order
- Performance on both RAID 10 and RAID 01 are the same.
- The storage capacity of RAID 10 and RAID 01 is the same.

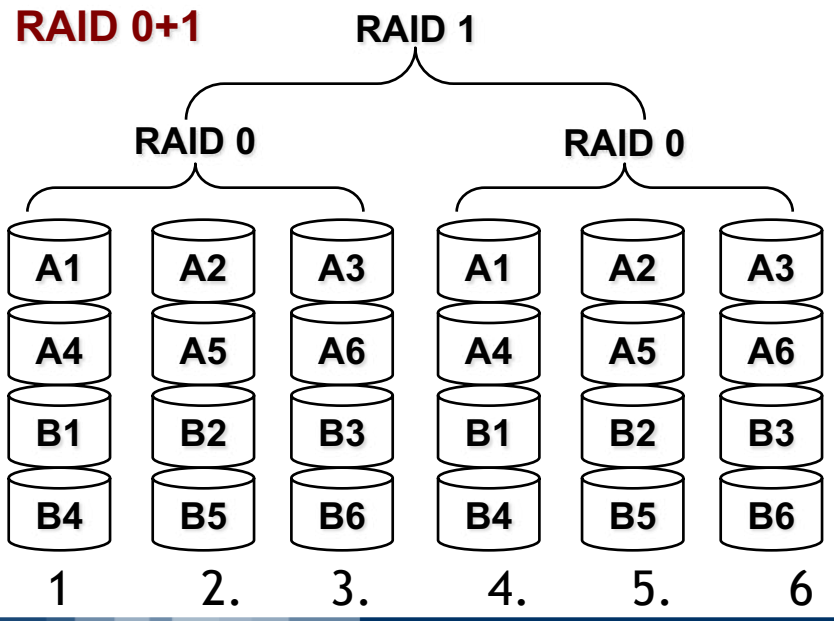




RAID 0, 1, 0+1 and 1+0 organizations

- The main difference is the **fault tolerance level**:
  - ✓On most implementations of RAID controllers, **RAID 0+1 fault tolerance is less**. On RAID 0+1, since we have only two groups of RAID 0, if two drives (one in each group) fails, the entire RAID 0+1 will fail.
  - ✓**RAID 1+0 fault tolerance is larger**. On RAID 1+0, since there are many groups (as the individual group is only two disks), even if three disks fails (one in each group), the RAID 1+0 is still functional. In the above RAID 1+0 example, even if Disk 1, Disk 3, Disk 5 fails, the RAID 1+0 will still be functional.



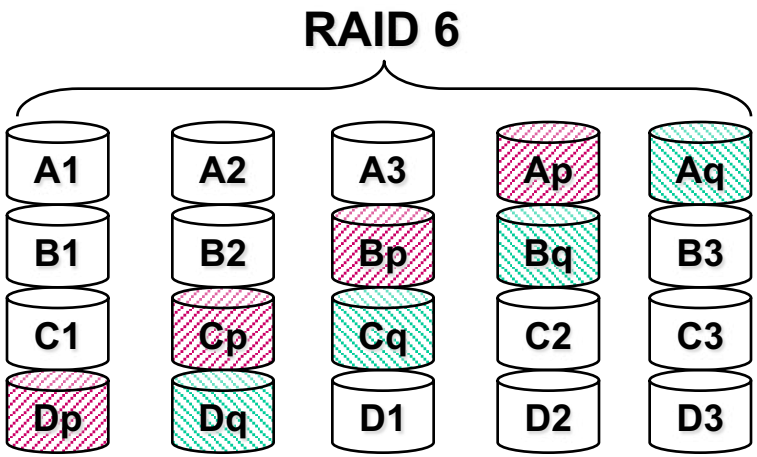


RAID level 6

- The last level RAID is the level 6, able **to withstand the failure of two disks simultaneously, combining two levels of parity, distributed on the various disks**.
- RAID6: more *fault tolerance* with respect RAID5
  - Uses Solomon-Reeds codes with two redundancy schemes: (P+Q) distributed and independent
  - 2 concurrent failures are tolerated
  - high overhead for the computation of parities
  - N + 2 disks required
  - with 4 data disks, each write require 6 disk accesses due to the need to update both the P and Q parity blocks (slow writes)
  - adopted for very critical applications

RAID 6 organization

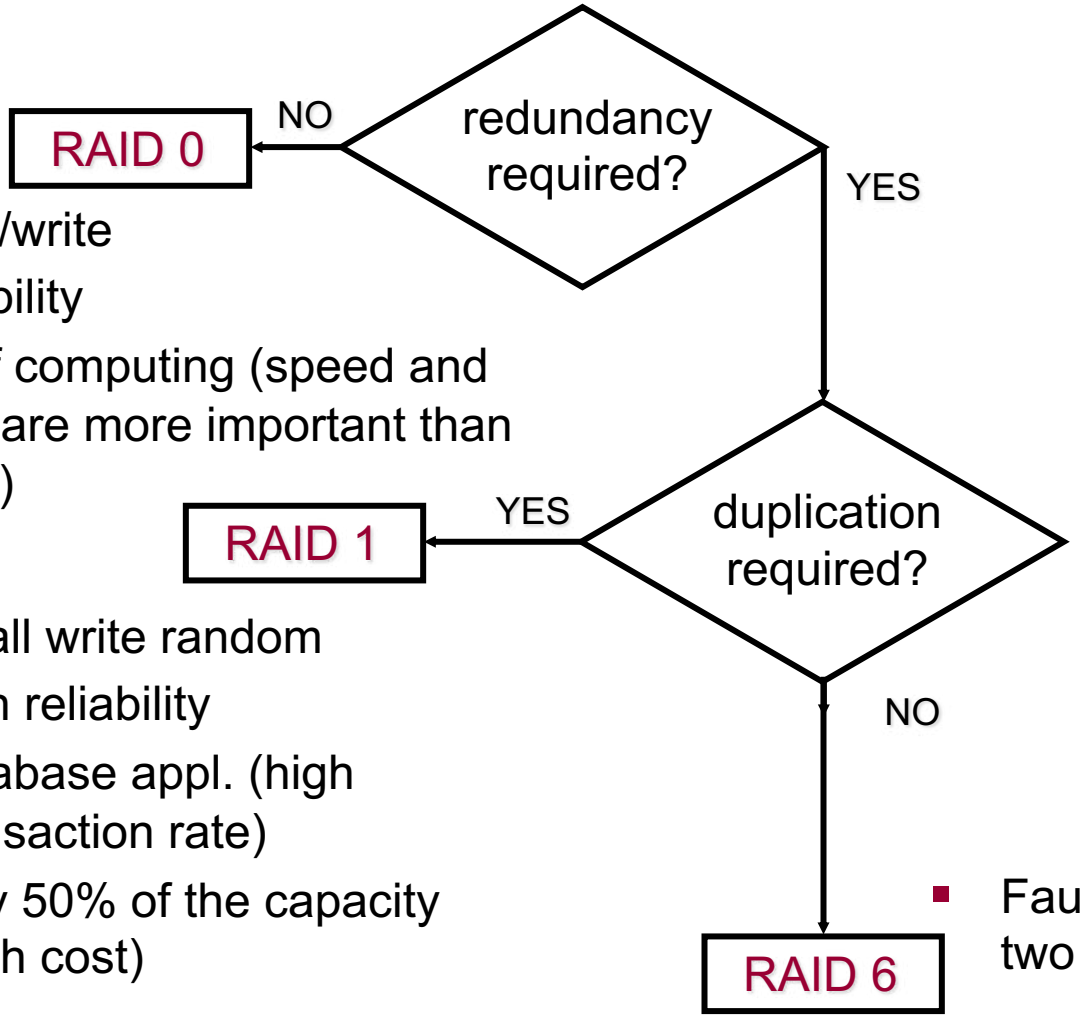
A1, B1,... are data blocks
Ap, Aq,... are parity blocks



not efficient when the number of disks is small !

when the number of disks increases the loss of efficiency decreases but the probability of 2 concurrent failures increases, so RAID 6 becomes mandatory

selection of RAID level (2)



- fast read/write
- low reliability
- high-perf computing (speed and capacity are more important than reliability)

- small write random
- high reliability
- database appl. (high transaction rate)
- only 50% of the capacity (high cost)

- Fault tolerance with two failures

characteristics of RAID levels (3)

RAID level	Capacity	Reliability	R/W performance	Rebuild performance	Suggested applications
0	100%	N/A	Very good	Good	Non critical data
1	50%	Excellent	Very good / good	good	Critical information
3	(n-1)/n	Good	Good / fair	Fair	Single user, large file processing, image processing
5	(n-1)/n	Good	Good/ fair	Poor	Database, transaction based applications
6	(n-2)/n	Excellent	Very good/ poor	Poor	Critical information, w/minimal
1+0	50%	Excellent	Very good/ good	Good	Critical information, w/better performance
3+0/5+0	(n-1)/n	Excellent	Very good/ good	Fair	Critical information w/fair performance

- RAID disks do not protect against multiple failures (except lev. 6)
- thus back-ups!!! back-ups!!!

POLITECNICO DI MILANO



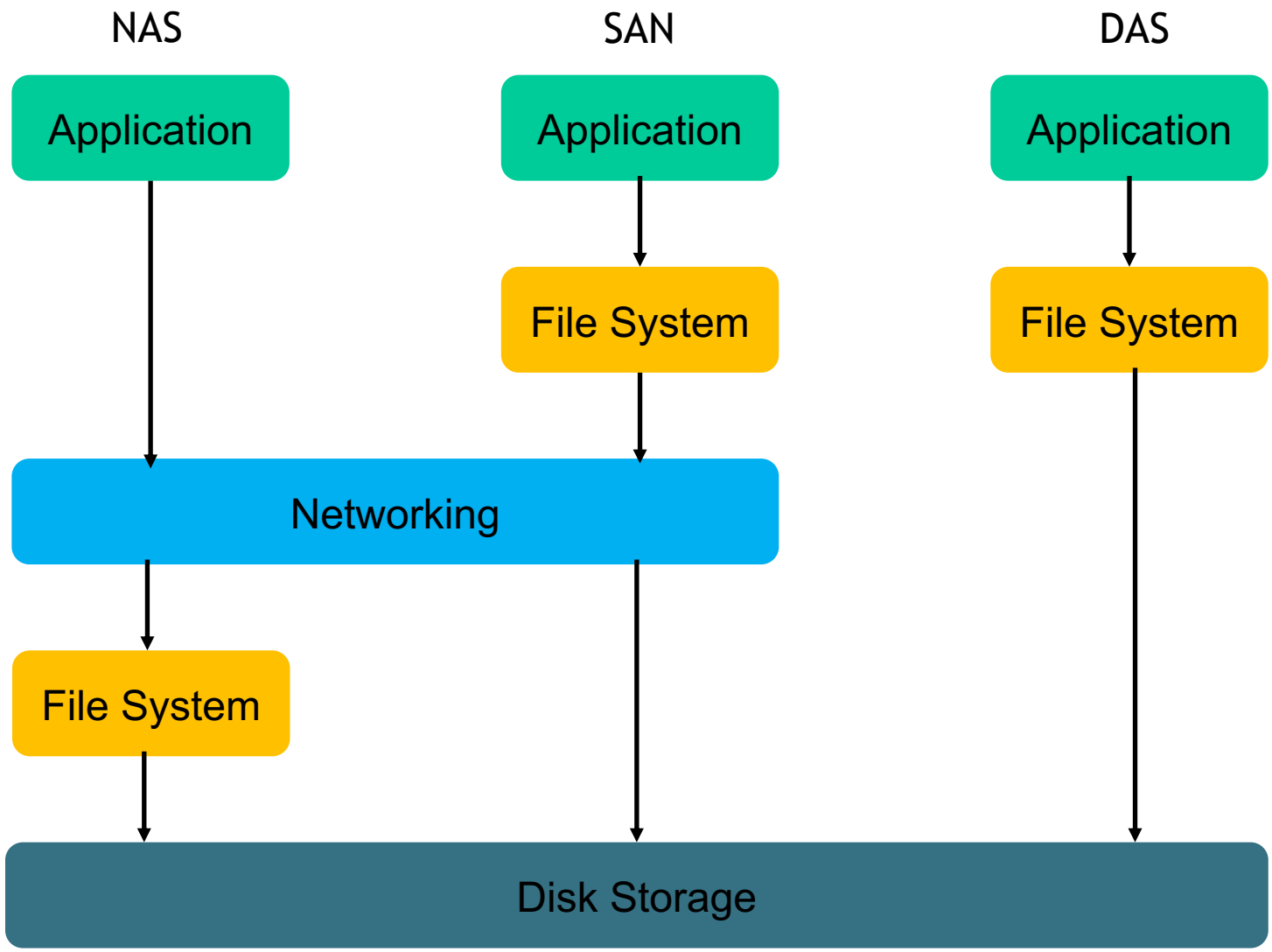


Storage systems: DAS, NAS and SAN

DAS, NAS and SAN

- A **Direct Attached Storage (DAS)** is a storage system directly attached to a server or workstation. They are visible as disks/volumes by the client OS
- A **Network Attached Storage (NAS)** is a computer connected to a network that provides only file-based data storage services (e.g., FTP, Network File System and SAMBA) to other devices on the network and is visible as File Server to the client OS
- **Storage Area Networks (SAN)** are remote storage units that are connected to PC using a specific networking technology (e.g., Fiber Channel) and are visible as disks/volumes by the client OS

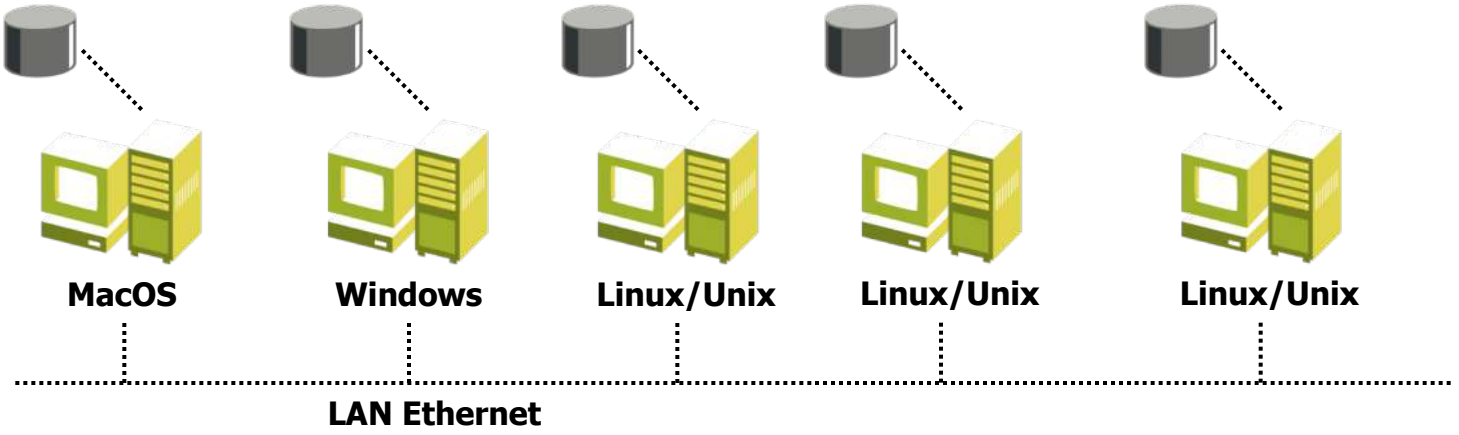
DAS, NAS, SAN: an architectural comparison



DAS
Direct Attached Storage

Direct Attached Storage

- DAS is a storage system directly attached to a server or workstation
- The term is used to differentiate non-networked storage from SAN and NAS (that will be described later)



Direct Attached Storage (DAS): physical model

Main features:

- limited scalability
- complex manageability
- limited performance
- To read files in other machines, the “file sharing” protocol of the OS must be used

Internal and external:

- DAS does not necessary mean “internal drives”.
- All the external disks, connected with a point-to-point protocol to a PC can be considered as DAS.

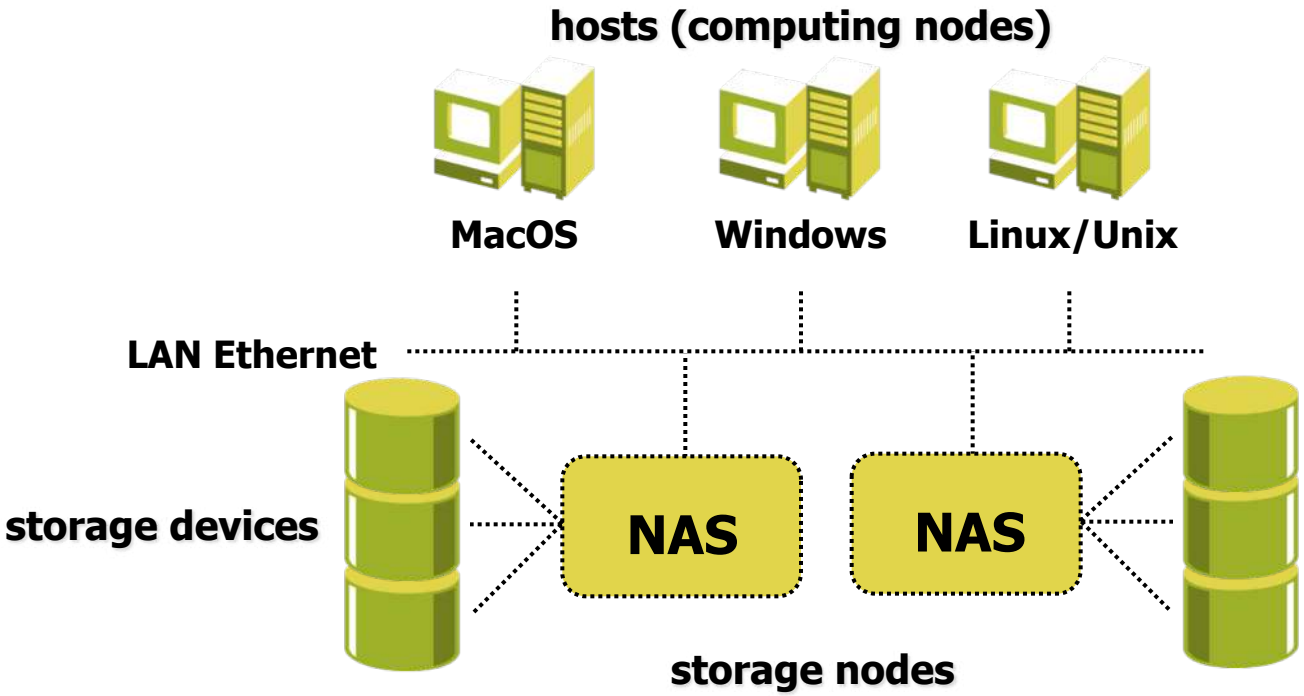
NAS
Network Attached Storage

Network Attached Storage (NAS)

- A NAS unit is a computer connected to a network that provides only file-based data storage services to other devices on the network
- NAS systems contain one or more hard disks, often organized into logical redundant storage containers or RAID
- Provide file-access services to the hosts connected to a TCP/IP network though Networked File Systems/SAMBA

Network Attached Storage (NAS): physical model

- Each NAS element has its own IP address
- Good scalability (incrementing the devices in each NAS element or incrementing the number of NAS elements)



NAS vs DAS

- The key differences between direct-attached storage (DAS) and NAS are
  - DAS is simply an extension of an existing server and is not necessarily networked.
  - NAS is designed as an easy and self-contained solution for sharing files over the network.

- Comparing NAS with local (non-networked) DAS, the performance of NAS depends mainly on the speed of and congestion on the network.

SAN:

Storage Area Network

Storage Area Networks

- **Storage Area Networks**, are remote storage units that are connected to PC using a specific networking technology.

NAS vs SAN I

- NAS provides both storage and a file system.
- This is often contrasted with SAN which provides only block-based storage and leaves file system concerns on the "client" side.

- One way to loosely conceptualize **the difference between a NAS and a SAN** is that
  - **NAS appears to the client OS (operating system) as a file server (**the client can map network drives to shares on that server)
  - **a disk available through a SAN still appears to the client OS as a disk**: it will be visible in the disks and volumes management utilities (along with client's local disks), and available to be formatted with a file system.

NAS vs SAN II

Traditionally:

- NAS is used for low-volume access to a large amount of storage by many users
- SAN is the solution for petabytes (10^{12}) of storage and multiple, simultaneous access to files, such as streaming audio/video.

SAN attached

- SANs have a special network devoted to the accesses to storage devices
- Two distinct networks (one TCP/IP and one dedicated network, e.g., Fiber Channel)
- High scalability (simply increasing the storage devices connected to the SAN network)