DATABASE

CSE3103 : Database
FALL 2020

Nazmus Sakib
Assistant Professor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Mass Storage

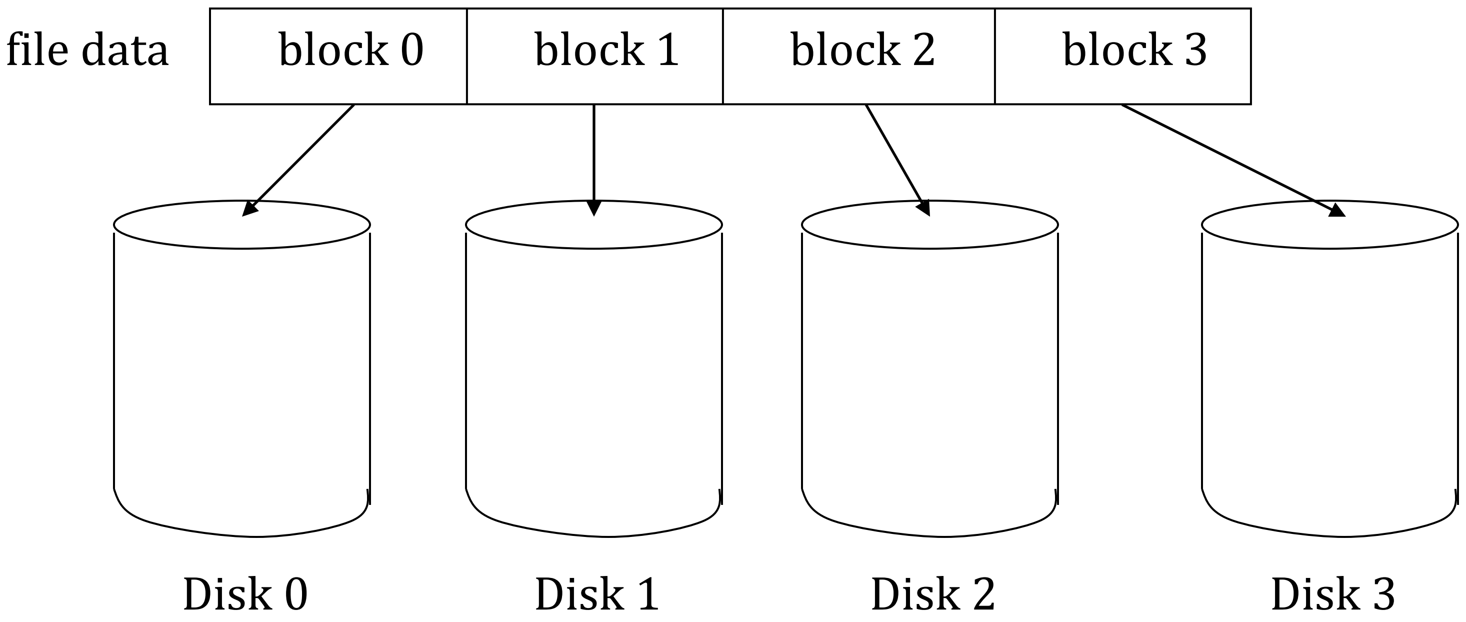
- Many systems today need to store many terabytes of data.
- Don’t want to use single, large disk
  - too expensive
  - failures could be catastrophic
- Would prefer to use many smaller disks.

RAID

- Redundant Array of Inexpensive Disks
- Basic idea is to connect multiple disks together to provide
  - large storage capacity
  - faster access to reading data
  - redundant data
- Many different levels of RAID systems
  - differing levels of redundancy, error checking, capacity, and cost

Striping

- Take file data and map it to different disks
- Allows for reading data in parallel



Parity

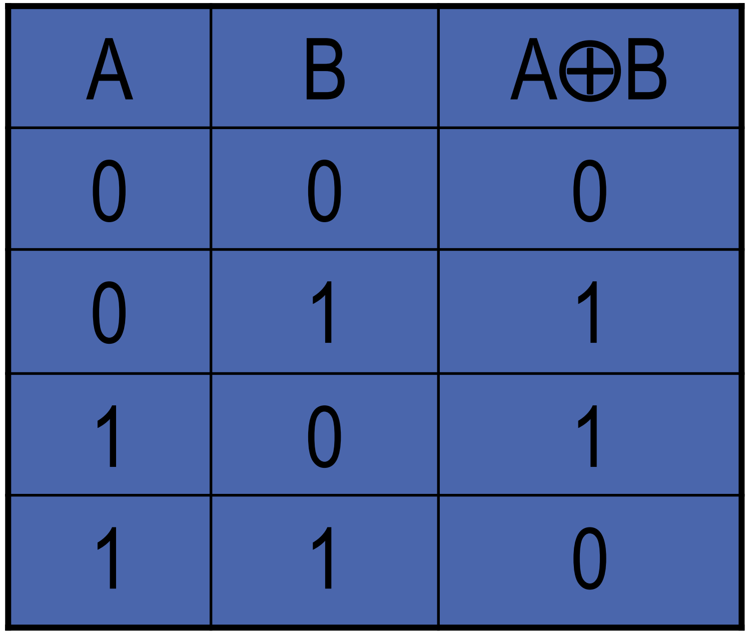
- Way to do error checking and correction
- Add up all the bits that are 1
  - if even number, set parity bit to 0
  - if odd number, set parity bit to 1
- To actually implement this, do an exclusive OR of all the bits being considered
- Consider the following 2 bytes

byte	parity
10110011	1
01101010	0

- If a single bit is bad, it is possible to correct it

How parity works?

- Truth table for XOR (same as parity)



A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Recovering from a disk failure

- Small RAID level 3 array with data disks D0 and D1 and parity disk P can tolerate failure of either D0 or D1

D0	D1	P
0	0	0
0	1	1
1	0	1
1	1	0

D1⊕P=D0	D0⊕P=D1
0	0
0	1
1	0
1	1

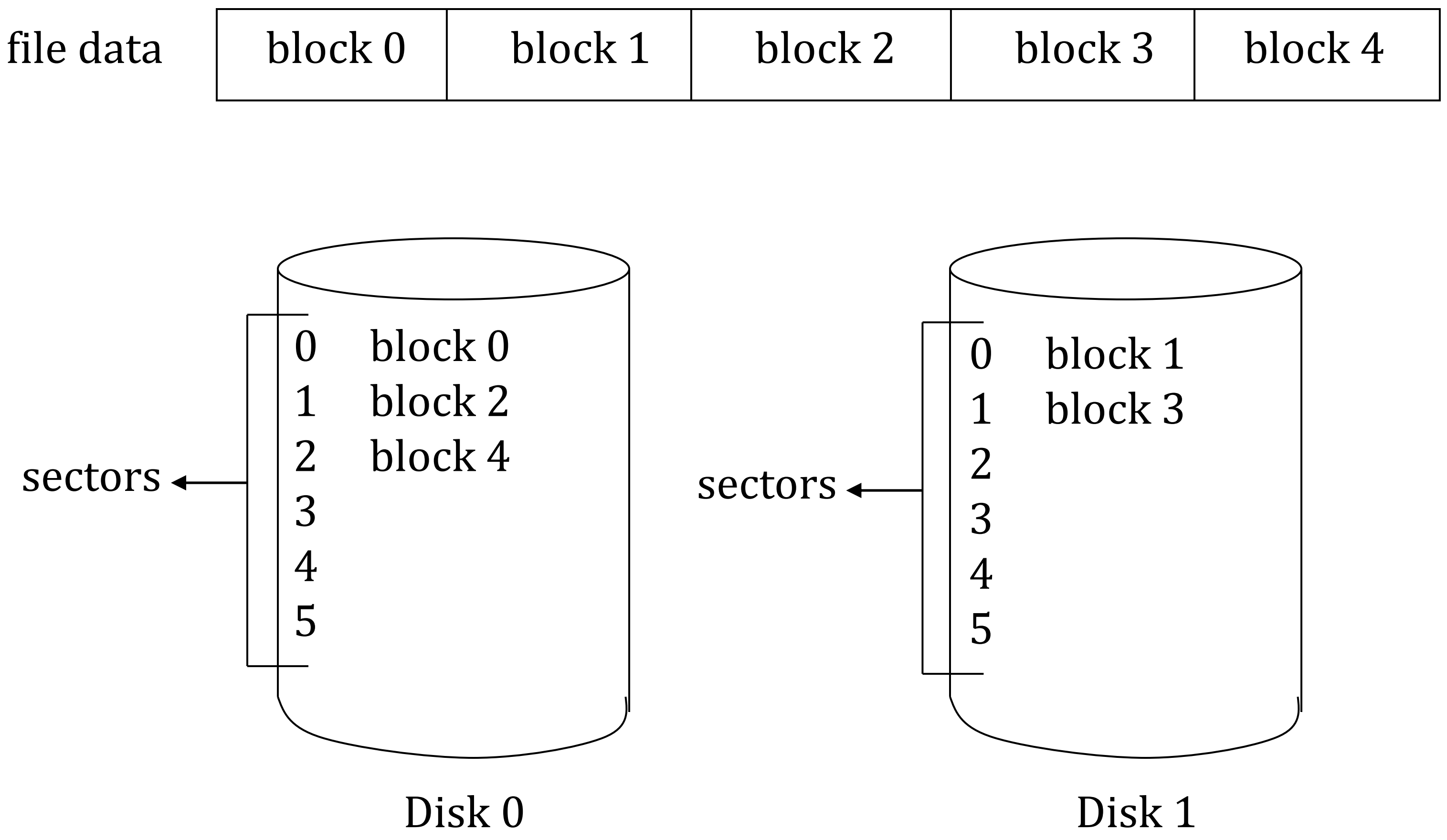
Mirroring

- Keep to copies of data on two separate disks
- Gives good error recovery -if some data is lost, get it from the other source
- Expensive -requires twice as many disks
- Write performance can be slow -have to write data to two different spots
- Read performance is enhanced -can read data from file in parallel

RAID Level-0

- Often called striping
- Break a file into blocks of data
- Stripe the blocks across disks in the system
- Simple to implement
  - disk = file block % number of disks
  - sector = file block / number of disks
- provides no redundancy or error detection
  - important to consider because lots of disks means low Mean Time To Failure (MTTF)

RAID Level-0



file data

block 0	block 1	block 2	block 3	block 4

0	block 0
1	block 2
2	block 4
3	
4	
5	

sectors

Disk 0

0	block 1
1	block 3
2	
3	
4	
5	

sectors

Disk 1

RAID Level-0 analysis

Failure Rate:

- MTBF of RAID0 is roughly proportional to the number of disks in the array.
- Pr(disk fail) = 5%, then

  Pr(at least one fails) = 1 – Pr(nonefails) = 1 – $[1\text{-}0.05]^2$ = 9.75%

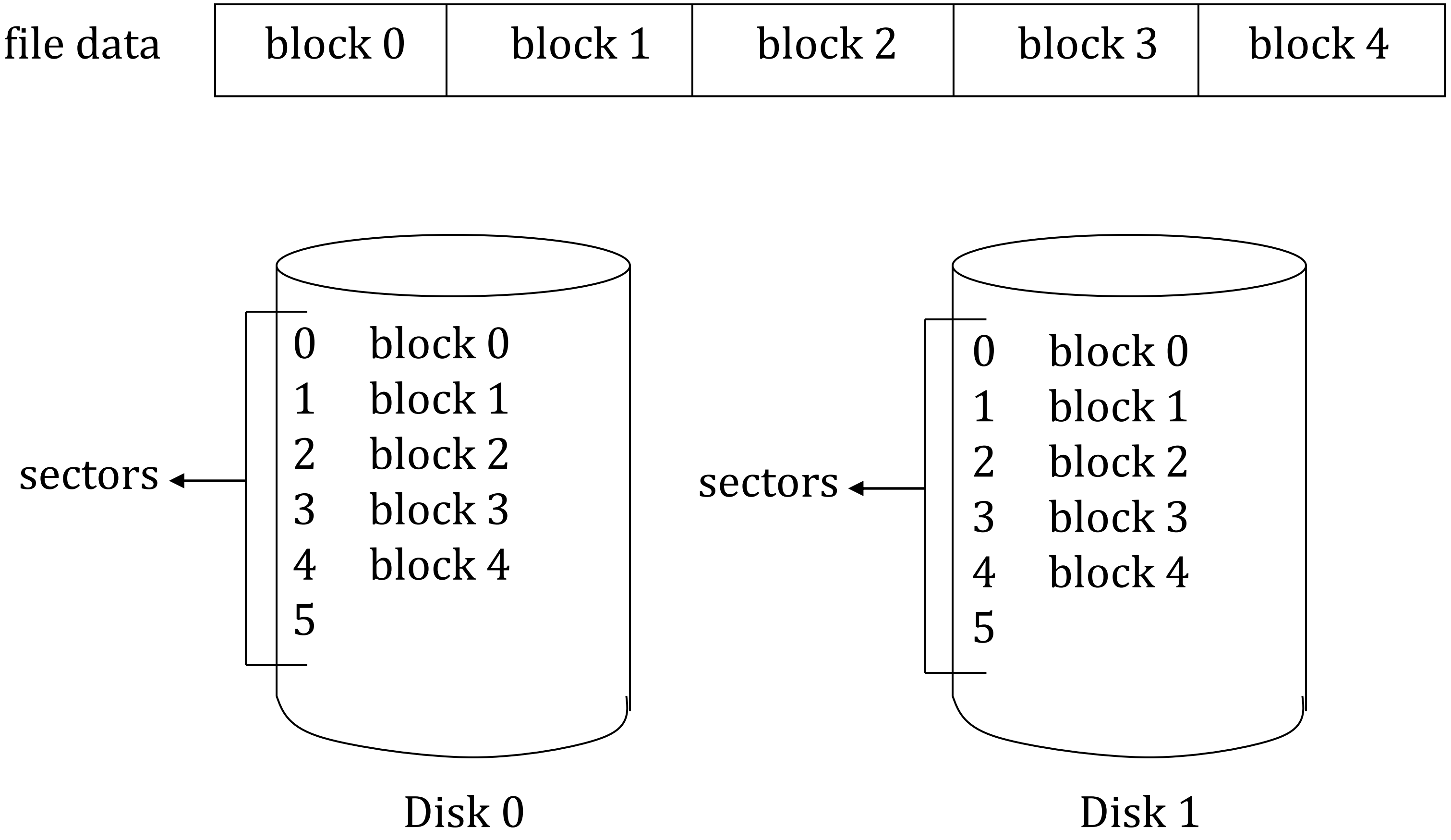
Performance:

- The fragments are written to their respective disks simultaneously on the same sector.
- This allows smaller sections of the entire chunk of data to be read off the drive in parallel, hence good performance.

RAID Level-1

- A complete file is stored on a single disk
- A second disk contains an exact copy of the file
- Provides complete redundancy of data
- Read performance can be improved –file data can be read in parallel
- Write performance suffers –must write the data out twice
- Most expensive RAID implementation –requires twice as much storage space

RAID Level-1



file data	block 0	block 1	block 2	block 3	block 4

sectors

0 block 0
1 block 1
2 block 2
3 block 3
4 block 4
5

Disk 0

sectors

0 block 0
1 block 1
2 block 2
3 block 3
4 block 4
5

Disk 1

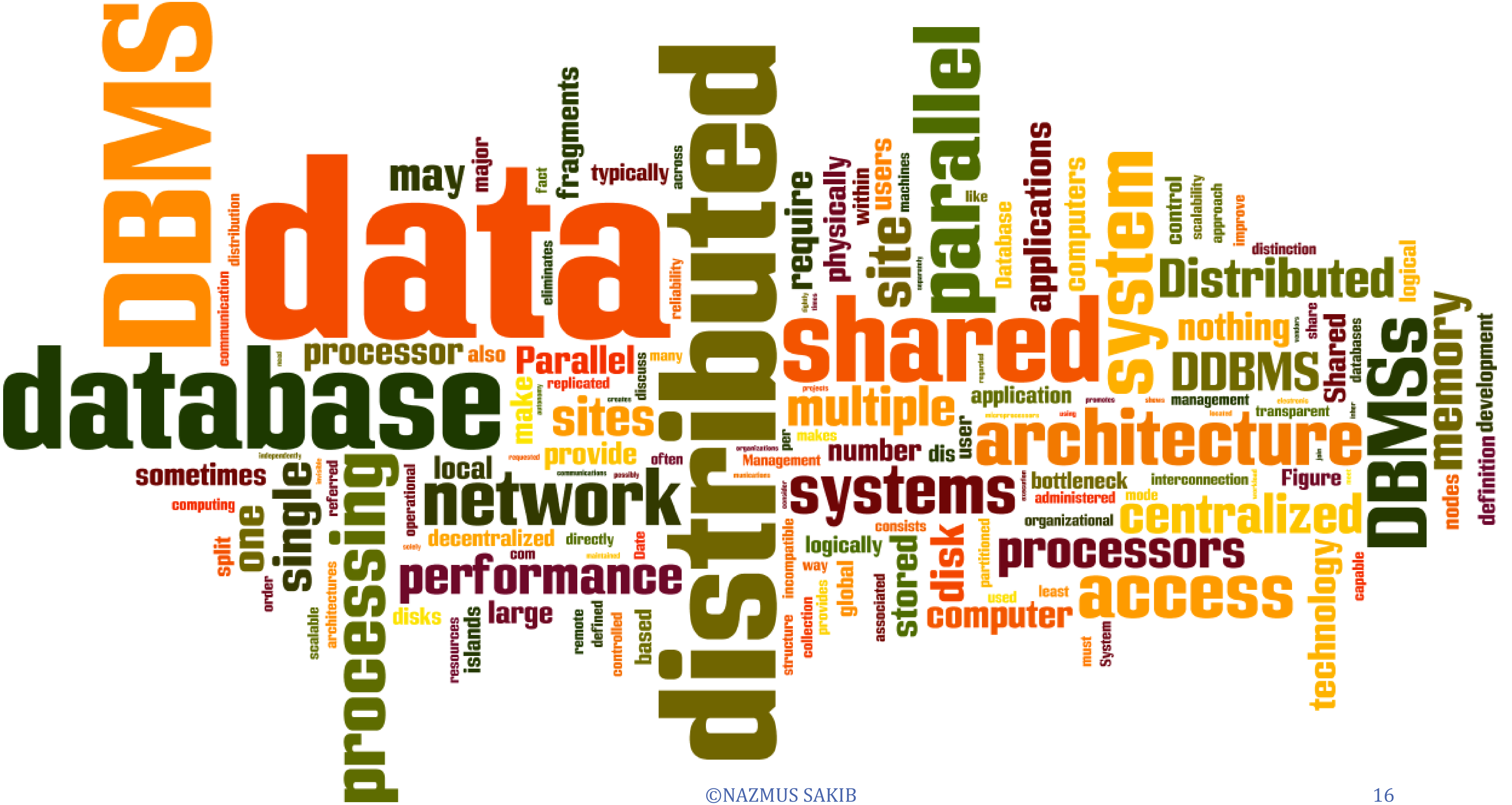
RAID Level-1 analysis

Failure Rate:

- If Pr(disk fail) = 5%, then the probability of both the drives failing in a 2 disk array is P(both fail) = $(0.05)^2$ = 0.25%.

Performance:

- If we use independent disk controllers for each disk, then we can increase the read or write speeds by doing operations in parallel.



distributed
data
database
DBMS
shared
parallel
architecture
system
systems
processing
network
access
performance
processors
centralized
DBMSs
Distributed
DDBMS
multiple
memory
sites
site
disk
computer
technology
applications
computers
single
processor
Parallel
nothing
Shared
may
local
sometimes
number
provide
decentralized
large
stored
require
physically
logically
users
fragments
typically
major
bottleneck
application
management
transparent
interconnection
Figure
organizational
administered
Management
one
make
also
replicated
many
discuss
often
dis
user
makes
per
global
way
consists
partitioned
used
least
mode
must
System
capable
definition
development
nodes
logical
databases
share
distinction
control
scalability
approach
improve
like
Database
within
machines
across
fact
eliminates
reliability
distribution
communication
computing
split
order
referred
operational
scalable
architectures
resources
islands
disks
directly
com
Date
remote
defined
controlled
based
structure
incompatible
collection
provides
associated
independently
invisible
requested
communications
possibly
organizations
promotes
shows
microprocessors
using
located
electronic
projects