ICOM 6005 – Database Management Systems Design

Dr. Manuel Rodríguez-Martínez

Electrical and Computer Engineering Department

Lecture ?? – September October 18, 2001

Readings

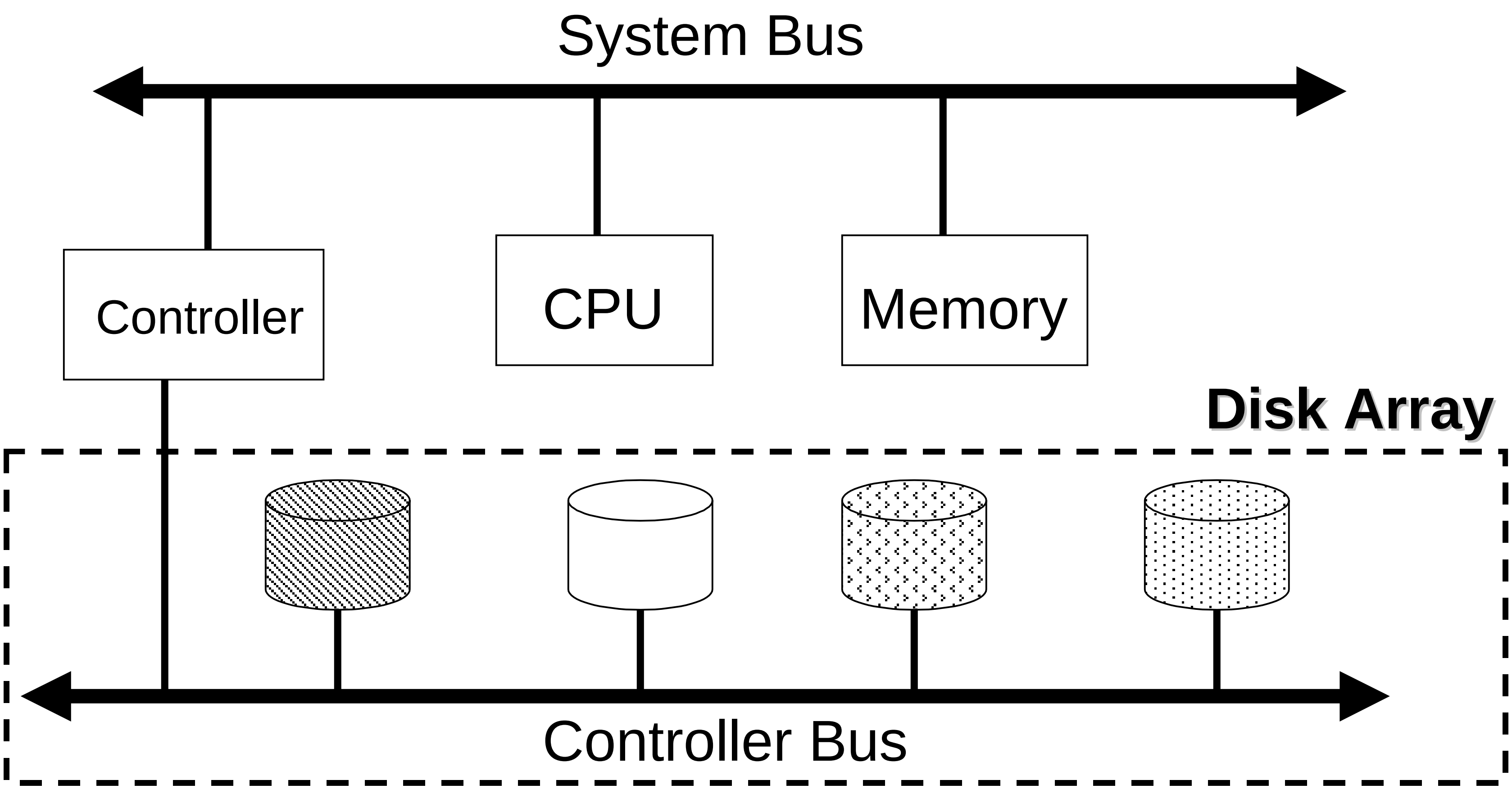
- Read
  - New Book: Chapter 7
    - "Storing Data: Disks and Files"
  - Old Book: Chapter 3
    - "Storing Data: Disks and Files"
  - Paper:
    - "Disk Striping" by Kenneth Salem and Hector Garcia-Molina

Disks as performance bottlenecks …

- Microprocessor speed increase 50% per year.
- Disk performance improvements
  – Access time decreases 10% per year
  – Transfer rate decreases 20% per year
- Disk crash results in data loss.
- Solution: Disk array
  – Have several disk behave as a single large and very fast disk.
    - Parallel I/O
  – Put some redundancy to recover from a failure somewhere in the array

Disk Array

- Several disks are group into a single logical unit.

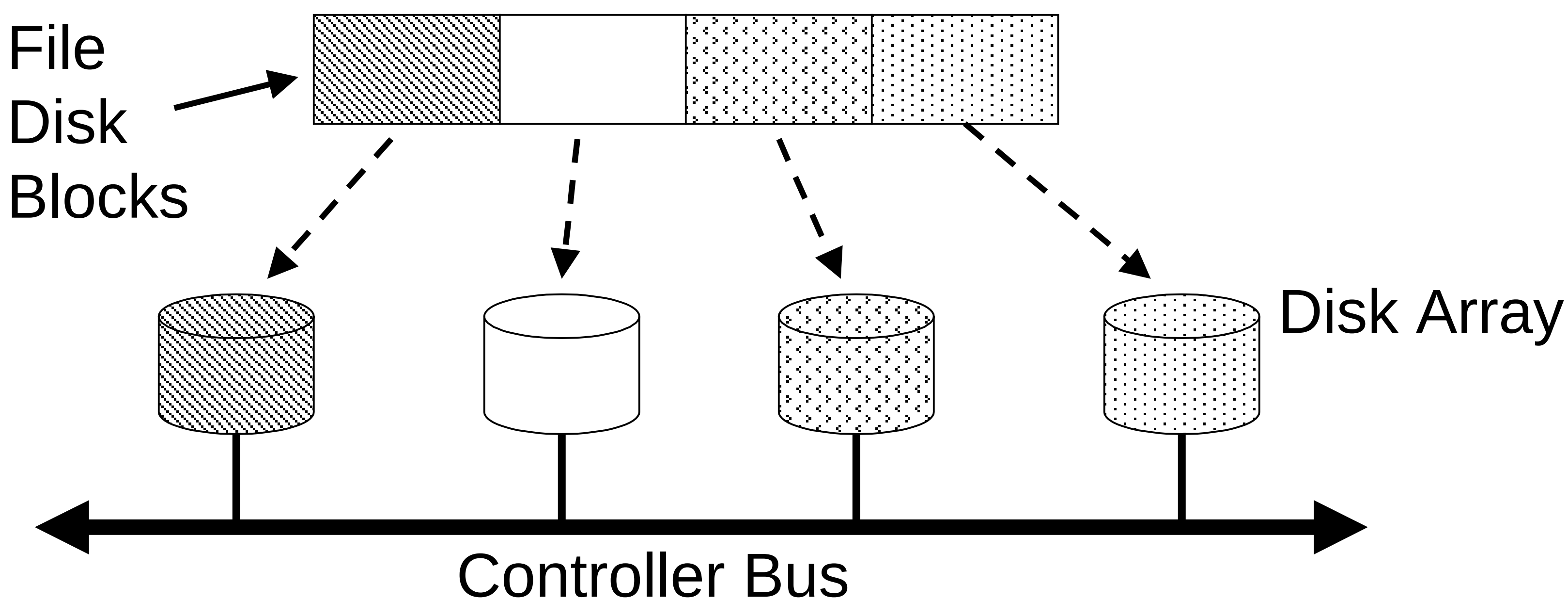


Disk striping

- Disk striping is a mechanism to divide the data in a file into segments that are scattered over the disks of the disk array.
- The minimum size of a segment is 1 bit, in which case each data blocks must be read from several disks to extract the appropriate bits.
  - The drawback of this approach is the overhead of managing data at the level of bits.
- Better approach is to have a striping unit of 1 disk block.
  - Sequential I/O can be run parallel since block can be fetched in parallel from the disks.

Disk Stripping – Block sized

- Disk stripping can be used to partition the data in a file into equal-sized segments of a block size that are distributed over the disk array.



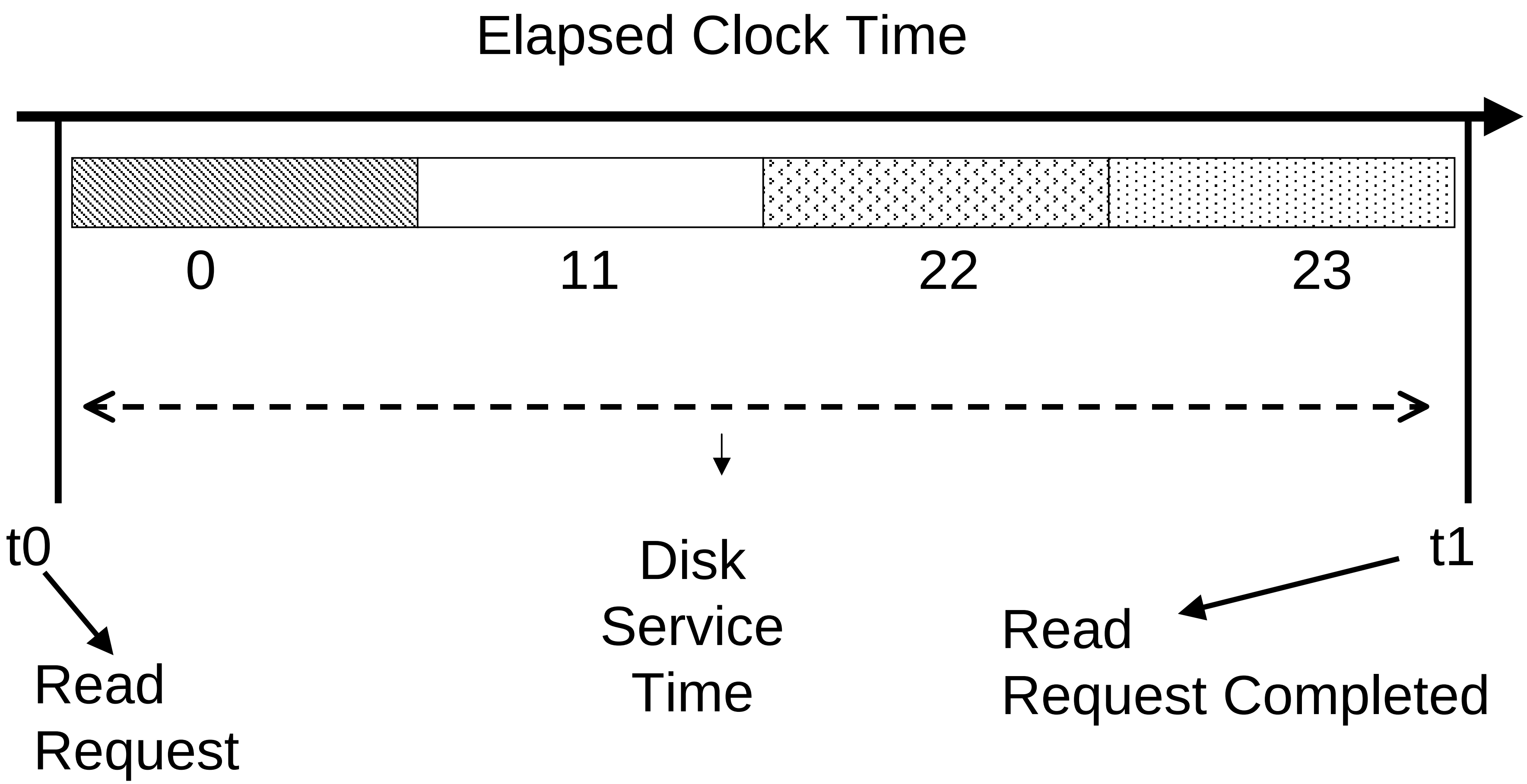
Data Allocation

- Data is partitioned into equal sized segments
  - Stripping unit
- Each segment is stored in a different disk of the arrays
- Typically, round-robin algorithm is used
- If we have n disks, then partition i is stored at disk
  - i mod n
- Example: Array of 5 disks, and file of 1MB with a 4KB stripping unit
  - Disk 0: gets partitions: 0, 5, 10, 15, 20, …
  - Disk 1: gets partitions: 1, 6, 11, 16, 21, …
  - Disk 2: gets partitions: 2, 7, 12, 17, 22, …
  - Etc.

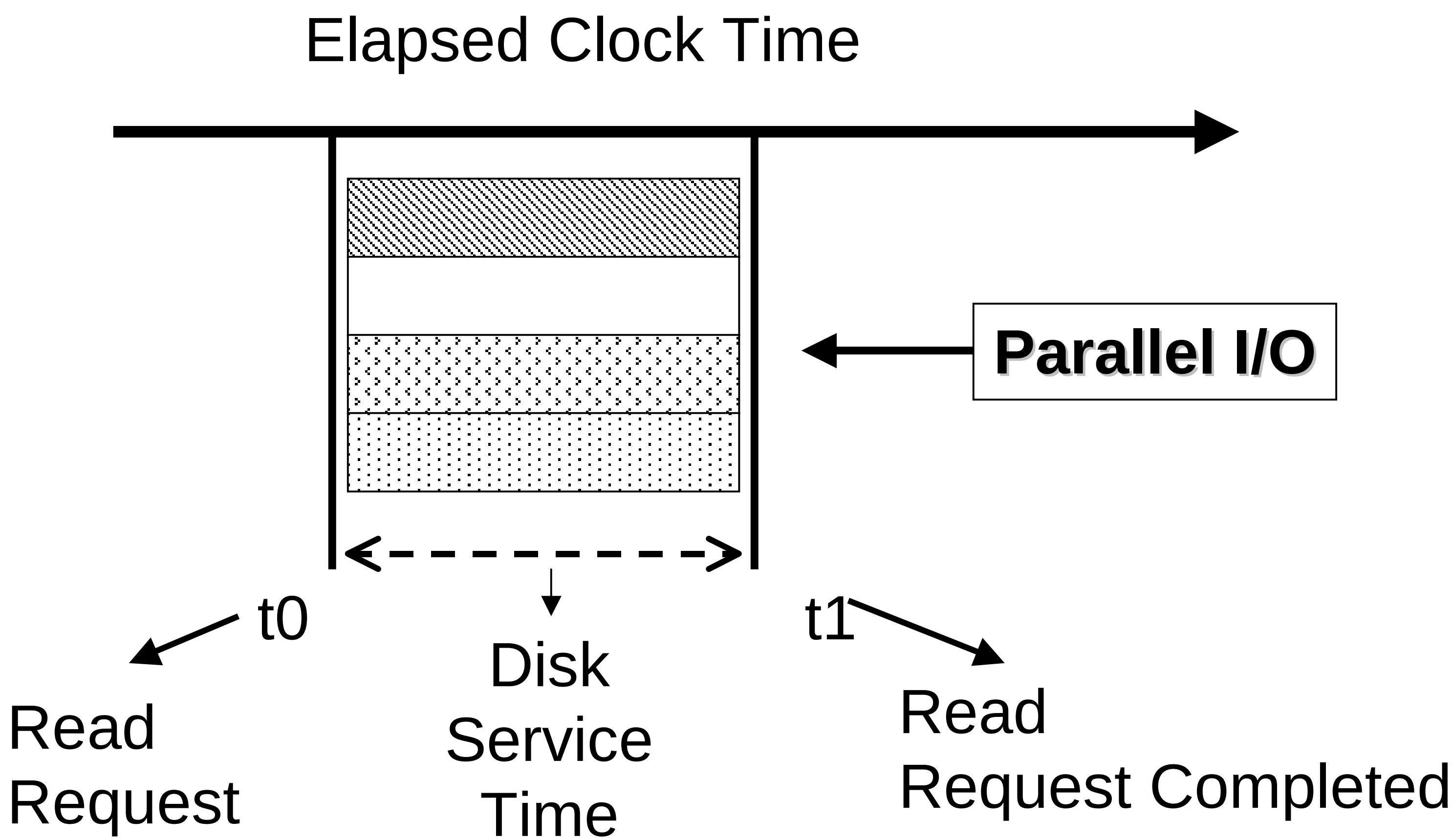
Benefits of Striping

- With striping we can access data blocks in parallel!
  - issue a request to the proper disks to get the blocks
- For example, suppose we have a 5-disk array with 4KB striping and disk blocks. Let F be a 1MB file. If we need to access partitions 0, 11, 22, 23, then we need to ask:
  - Disk 0 for partition 0 at time t0
  - Disk 1 for partition 11 at time t0
  - Disk 2 for partition 22 at time t0
  - Disk 3 for partition 23 at time t0
- All these requests are issued by the DBMS and are serviced concurrently by the disk array!

Single Disk Time Line



Striping Time Line



Time access estimates

- Access time:

  seek time + rotational delay + transfer time
- Disk used independently or in array: IBM Deskstar 14GPX 14.4 GB disk
  - Seek time: 9.1 milliseconds (msecs)
  - Rotational delay: 4.15 msecs
  - Tranfer rate: 13MB/sec
- How does striping compares with a single disk?
- Scenario: 1disk block(4KB) striping-unit, access to blocks 0, 11, 22, and 23. Disk array has 5 disks
  - Editorial Note: Looks like an exam problem!

Single Disk Access time

- Total time = sum of time to read each partition
- Time for partition 0:
  - 9.1 msec + 4.3msec + 4KB/(1MB/1sec)*(1MB/1024 KB)*(1000msec/1sec) =
  - 9.1 msec + 4.3msec + 3.9 msecs = 17.3 msecs
- Time for partition 11:
  - 9.1 msec + 4.3msec + 4KB/(1MB/1sec)*(1MB/1024 KB)*(1000msec/1sec) =
  - 9.1 msec + 4.3msec + 3.9 msecs = 17.3 msecs
- Time for partition 22:
  - 9.1 msec + 4.3msec + 4KB/(1MB/1sec)*(1MB/1024 KB)*(1000msec/1sec) =
  - 9.1 msec + 4.3msec + 3.9 msecs = 17.3 msecs
- Time for partition 23:
  - 9.1 msec + 4.3msec + 4KB/(1MB/1sec)*(1MB/1024 KB)*(1000msec/1sec) =
  - 9.1 msec + 4.3msec + 3.9 msecs = 17.3 msecs
- **Total time**: 4 * 17.3 msec = 69.2 msecs

Stripping Access Time

- Total time: maximum time to complete any read quest.
- Following same calculation as in previous slide:
  – Time for partition 0: 17.3 msec
  – Time for partition 11: 17.3 msec
  – Time for partition 22: 17.3 msec
  – Time for partition 23: 17.3 msec
- Total time:
  – max{17.3msec, 17.3msec 17.3msec 17.3msec} = 17.3 msec
- In this case, stripping gives us a 4-1 better (4 times) performance because of **parallel I/O**.

The problem with striping

- Striping has the advantage of speeding up disk access time.
- But the use of a disk array decrease the reliability of the storage system because more disks mean more possible points of failure.
- Mean-time-to-failure (MTTF)
  - Mean time to have the disk fail and lose its data
- MTTF is inversely proportional to the number of components in used by the system.
  - The more we have the more likely they will fall apart!

MTTF in disk array

- Suppose we have a single disk with a MTTF of 50,000 hrs (5.7 years).
- Then, if we build an array with 50 disks, then the have a MTTF for the array of 50,000/50 = 1000 hrs, or 42 days!, because any disk can fail at any given time with equal probability.
  - Disk failures are more common when disks are new (bad disk from factory) or old (wear due to usage).
- Morale of the story: More does not necessarily means better!

Increasing MTTF with redundancy

- We can increase the MTTF in a disk array by storing some redundant information in the disk array.
  - This information can be used to recover from a disk failure.
- This information should be carefully selected so it can be used to reconstruct original data after a failure.
- What to store as redundant information?
  - full data block?
  - Parity bit for a set of bit locations across the disks