CSCI 4730/6730 OS
(Chap #4 Threads & Concurrency – Part III)

In Kee Kim

Department of Computer Science

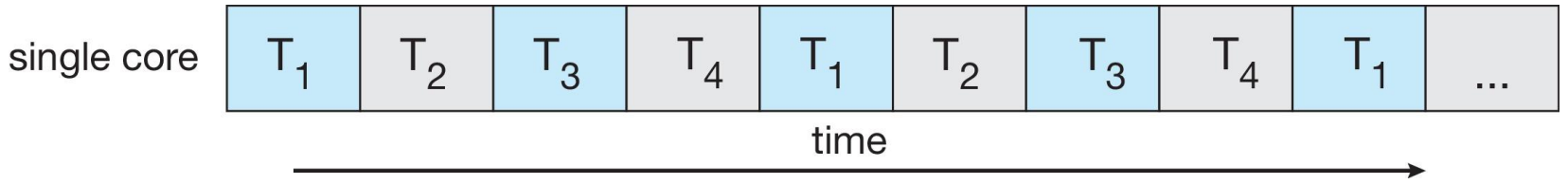
University of Georgia

Multicore Programming

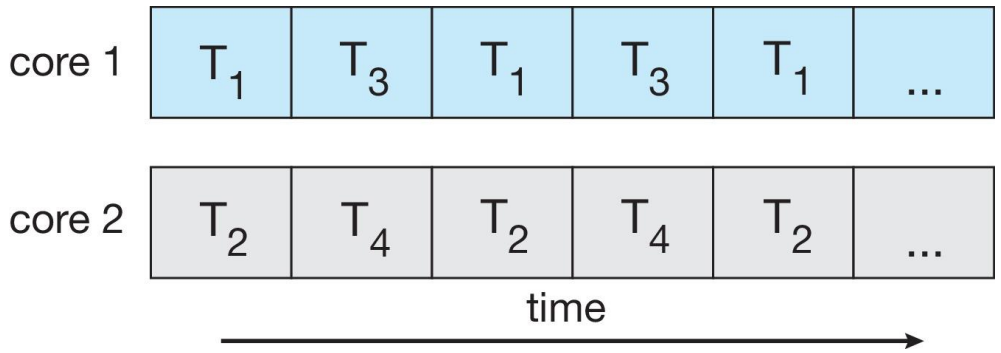
- *Concurrency*
- *Parallelism*
- *Data Parallelism*
- *Task Parallelism*

Multicore Programming

- **Concurrent execution on single-core system:**



- **Parallelism on a multi-core system:**



Each core performs concurrent executions

Data and Task Parallelism

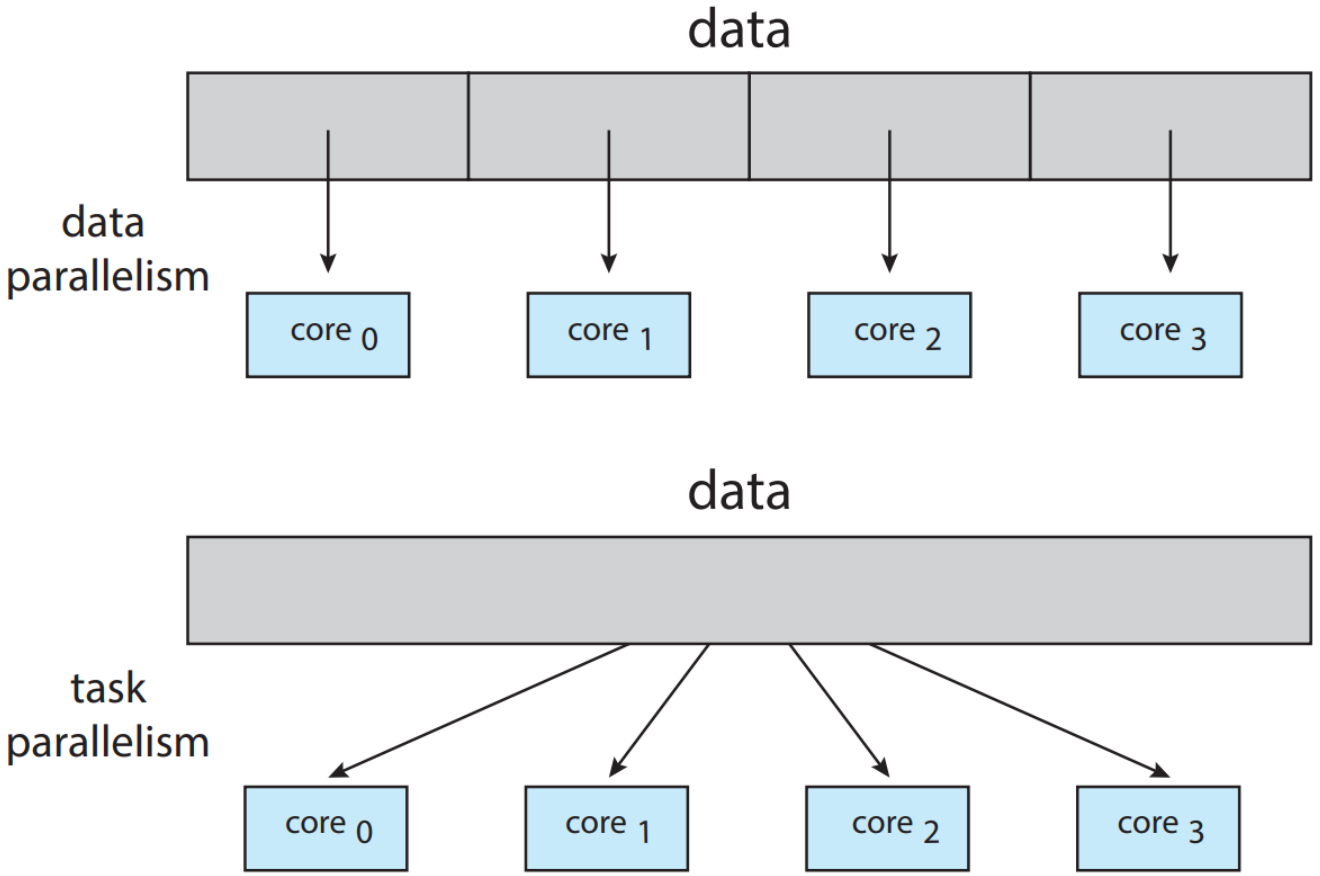


Figure 4.5 Data and task parallelism.

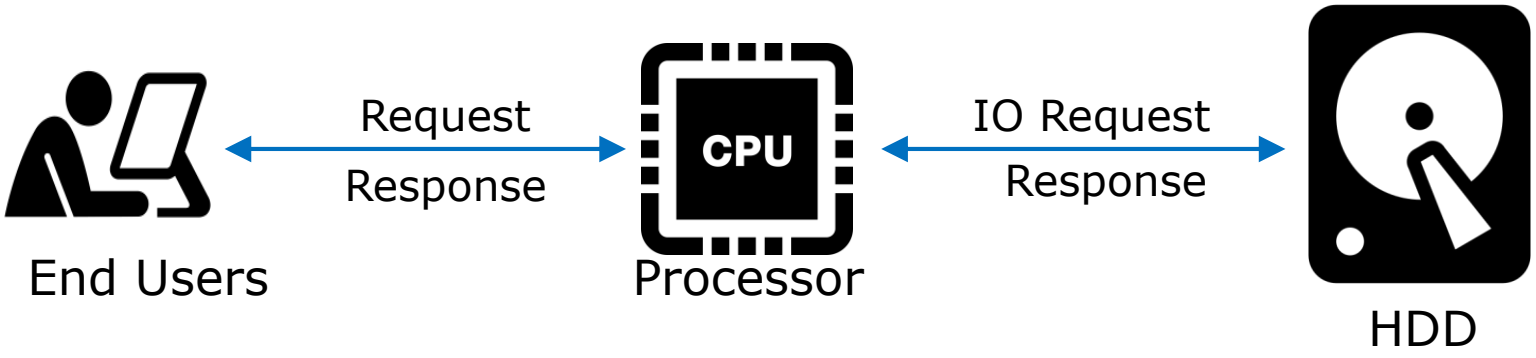
Amdahl's Law

- This is interesting!
- Theoretical performance gain from parallelism (adding more processors) to an application
- Simple question
  - Can I get 4x speed up if I use four threads for my app?
    - Assumption: You have a four-core machine.
    - Compared to single thread architecture?

Amdahl's Law

- Simple question
  - Can I get 4x speed up if I use four threads for my app?
    - Assumption: You have a four-core machine.
    - Compared to single thread architecture?
  - PA #1 -- How much performance improvement?
    - `time ./wc_mul 1 large.txt 0`
    - `time ./wc_mul 10 large.txt 0`

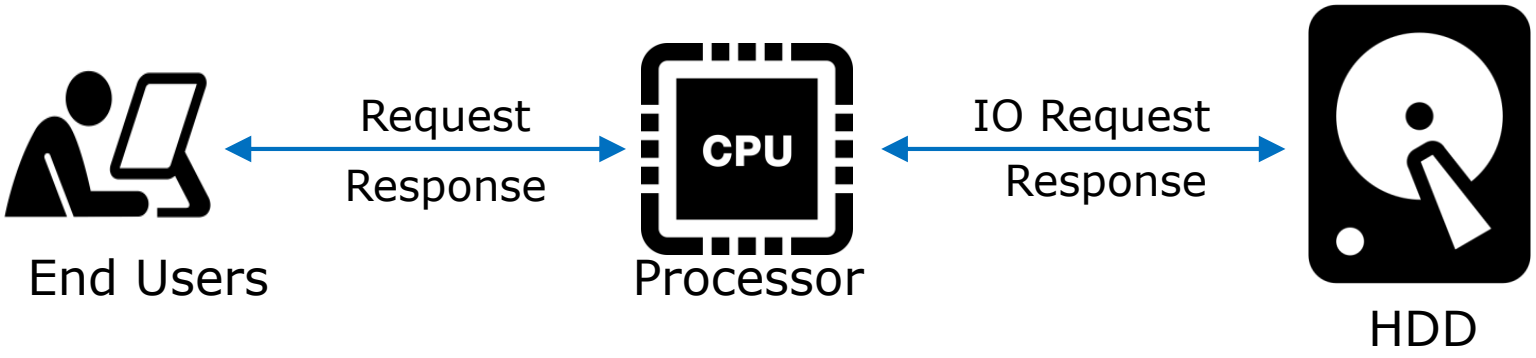
Speedup



Assuming Processor (CPU) takes 0.3 sec, HDD takes 0.7 sec.

What is throughput of this system (Processor + HDD)?

Speedup

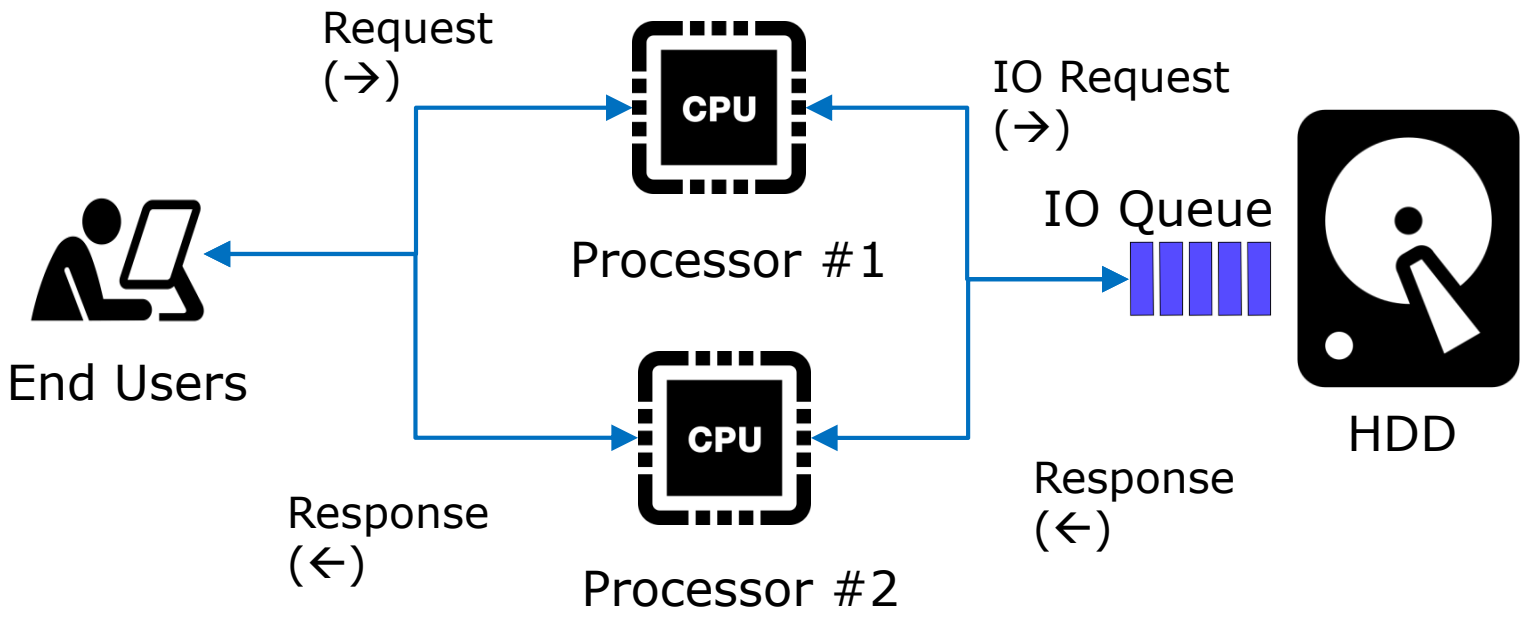


Assuming Processor (CPU) takes 0.3 sec, HDD takes 0.7 sec.

What is throughput of this system (Processor + HDD)?

- e.g., 2 reqs / 2 sec = 1 req/sec

Speedup



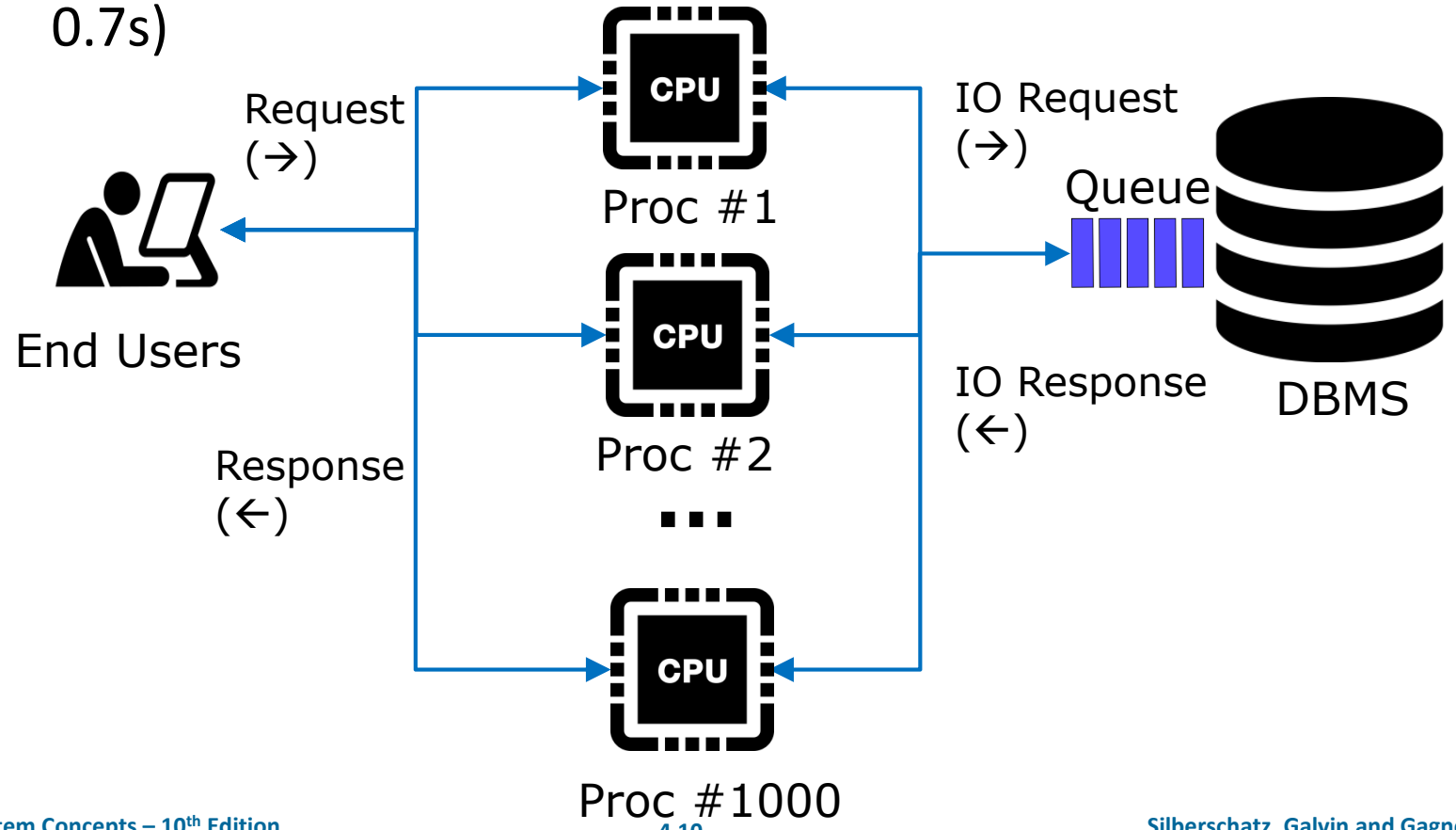
Assume the processor can be replicated; what's throughput now? (Processor: 0.3s, HDD: 0.7s)

→ 2 reqs / (0.3 + 0.7 + 0.7) = 1.18

Speedup

Best speedup?

- What if you have 1000 processors? (Processor: 0.3s, HDD: 0.7s)



Speedup

- **Best speedup?**
  - What if you have 1000 processors? (Proc: 0.3s, HDD:0.7s)
    - **What is throughput with 1K processors?**
    - **1000 reqs/ (0.3 + 1000*0.7) = 1.4279…**
  - What if you have infinite numbers of processors? (Proc: 0.3s, HDD: 0.7s)
    - **What is throughput now?**
    - **In the extreme, 0.3 (processor) is deleted and it becomes 1/0.7 = 1.4285**

Speedup, in general

- Given an application such that
  - **Total Duration** = ***Serial Part Duration*** + ***Parallel Part Duration***
  - Aka, **T = S + P**
- How fast could we make it given **infinite processors** for ***P***? (Aka max speedup?)
  - Answer: $\boldsymbol{Max\ Speedup = \frac{1}{(S)}}$
- E.g., assume that 20% of the job cannot be parallelized, than max speedup w/ infinite procs?
  - **1/(0.2) = 5x faster**

Speedup, in general (cont'd)

- What if we don't have infinite processors?
- Then, assuming we have ***K*** processors by which to attack the ***P (=1-S)***, the formula becomes:

$$Max\ Speedup = \frac{1}{(S) + (\frac{1-S}{K})}$$

- e.g., what if we only have 2 processors?

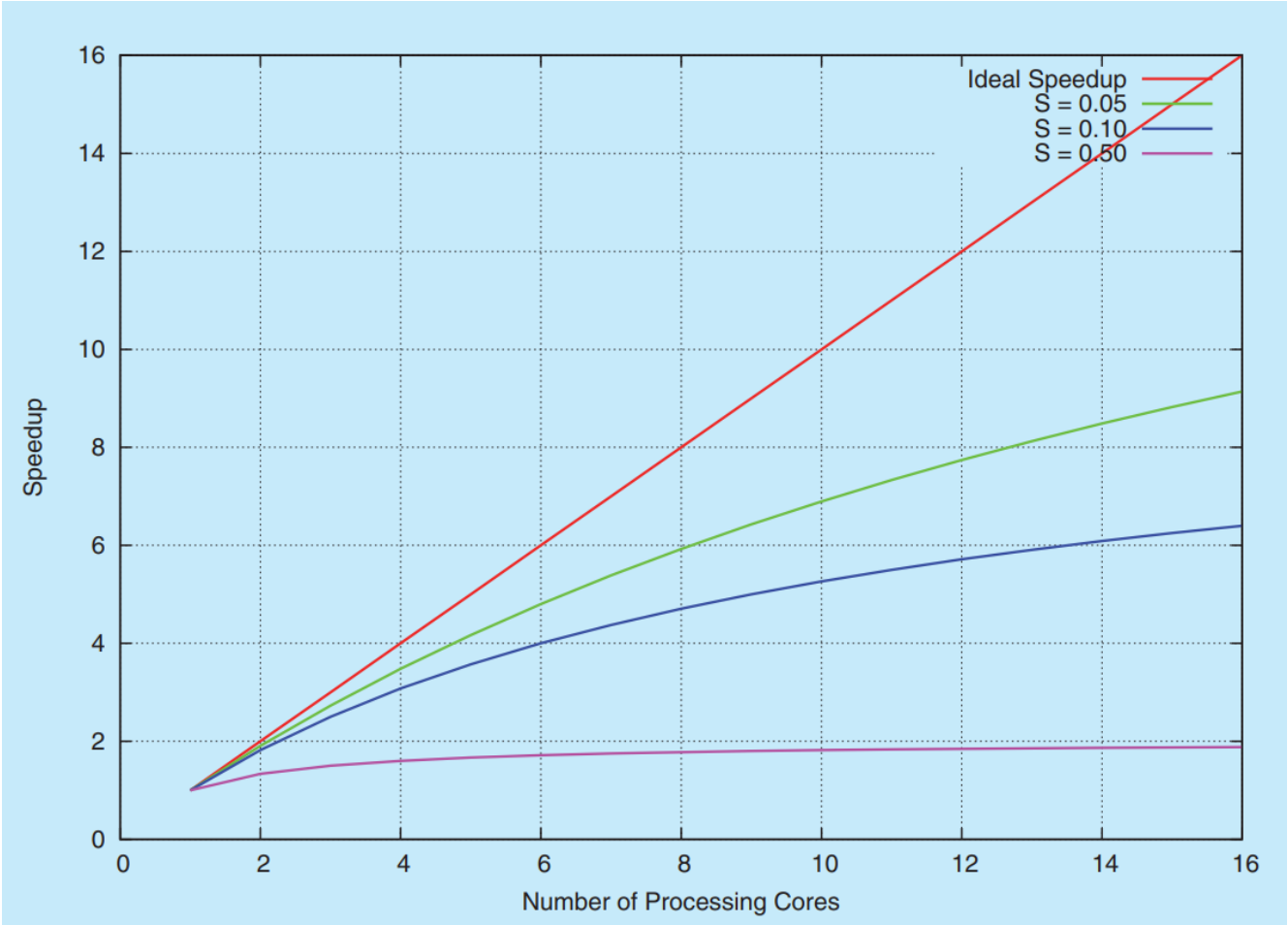
$$Max\ Speedup = \frac{1}{(0.2) + ((1-.2)/2)} = 1.67x$$

This is called Amdahl's law!

Example

- Using Amdahl's Law, calculate the speedup gain of an application that has a 60 percent parallel component for (a) two processing cores and (b) four processing cores.

Amdahl's Law


Ideal Speedup
S = 0.05
S = 0.10
S = 0.50
Speedup
Number of Processing Cores
0
2
4
6
8
10
12
14
16