Chapter 12: Multiprocessor Architectures

Lesson 01:

Performance characteristics of Multiprocessor Architectures and Speedup

Objective

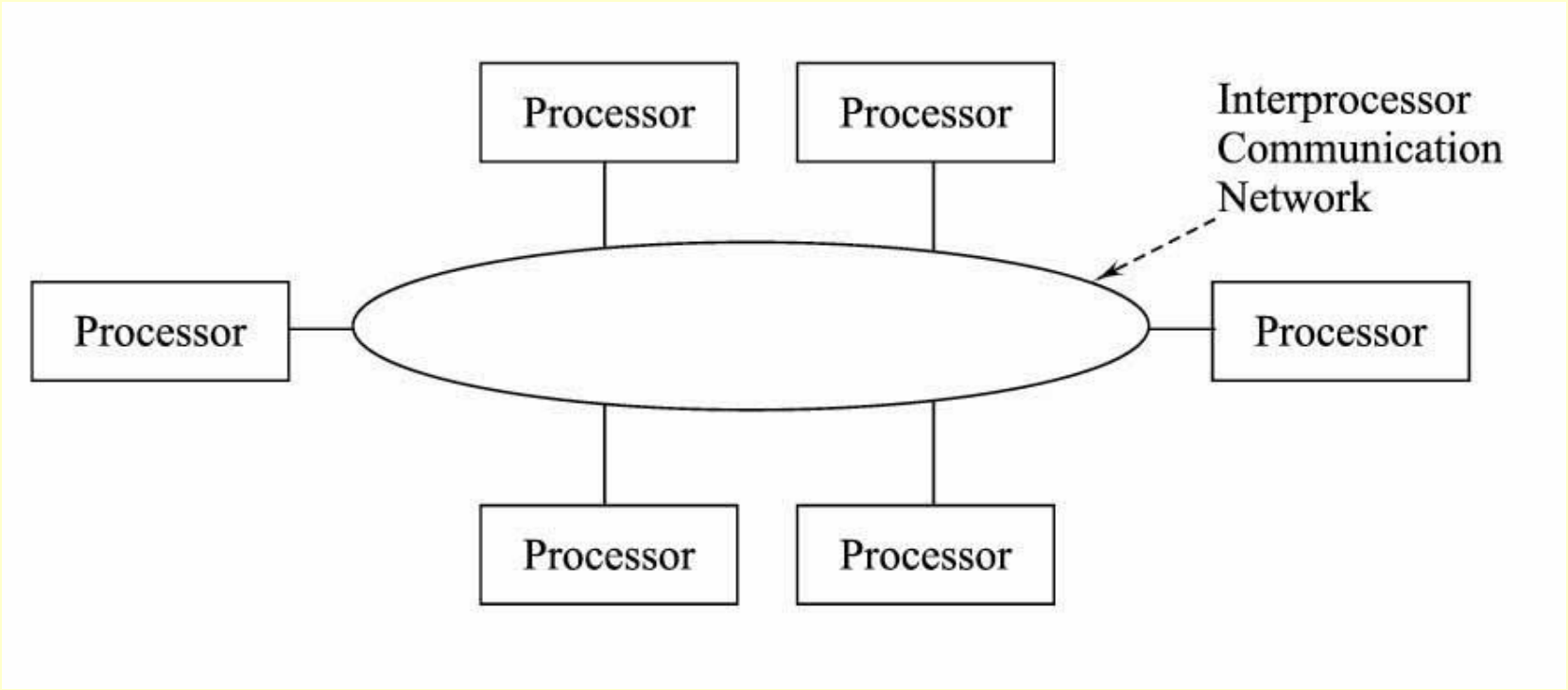
- Be familiar with basic multiprocessor architectures and be able to discuss speedup in multiprocessor systems, including common causes of less-than-linear and super-linear speedup

Basic multiprocessor architectures

Multiprocessors

- A set of processors connected by a communications network

Basic multiprocessor architecture



Early multiprocessors

- Often used processors that had been specifically designed for use in a multiprocessor to improve efficiency

Most current multiprocessors

- Taking advantage of the large sales volumes of uniprocessors to lower prices—The same processors that are found in contemporary uniprocessor systems

Most current multiprocessors

- Number of transistors that can be placed on a chip increases
- Features to support multiprocessor systems integrated into processors intended primarily for the uniprocessor
- More efficient multiprocessor systems built around these uniprocessors

Performance Characteristics of Multiprocessors

Multiprocessors speedup

- Designers of uniprocessor systems, measure performance in terms of *speedup*
- Similarly, multiprocessor architects measure performance in terms of *speedup*

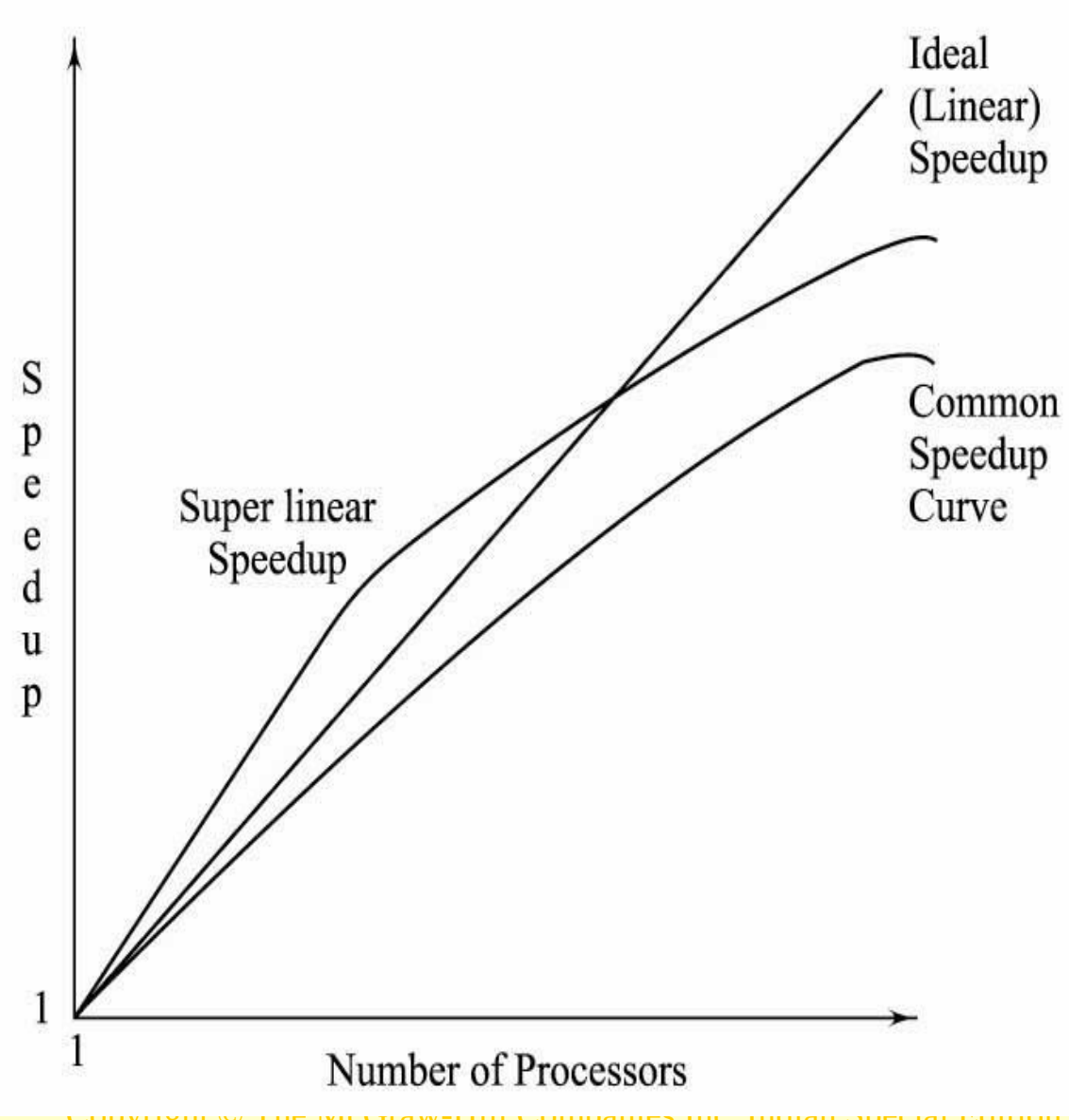
Multiprocessors speedup

- Speedup generally refers to how much faster a program runs on a system with n processors than it does on a system with one processor of the same type

Loop unrolling

- Transforming a loop with N iterations into a loop with N/M iterations
- Each of the iterations in the new loop does the work of M iterations of the old loop

Speedup on increasing the number of processors



Ideal and Practical speedups of Multiprocessors systems

Ideal Speedup in Multiprocessor System

- Linear speedup— the execution time of program on an *n*-processor system would be $1/n^{th}$ of the execution time on a one-processor system

Practical speedup in Multiprocessor System

- When the number of processors is small, the system achieves near-linear speedup
- As the number of processors increases, the speedup curve diverges from the ideal, eventually flattening out or even decreasing

Limitations on speedup of Multiprocessors systems

Limitations

- Interprocessor communication
- Synchronization
- Load Balancing

Limitations of Interprocessor communication

- Whenever one processor generates (computes) a value that is needed by the fraction of the program running on another processor, that value must be communicated to the processors that need it, which takes time
- On a uniprocessor system, the entire program runs on one processor, so there is no time lost to interprocessor communication

Limitations of Synchronization

- It is often necessary to synchronize the processors to ensure that they have all completed some phase of the program before any processor begins working on the next phase of the program

Example of Synchronization in Programs on Multiprocessors systems

Example of Programs that simulate physical phenomena, such as game

- Generally divide time into steps of fixed duration
- Require that all processors have completed their simulation of a given time-step before any processor can proceed to the next

Example of Programs that simulate physical phenomena, such game

- The simulation of the next time-step can be based on the results of the simulation of the current time-step
- This synchronization requires interprocessor communication, introducing overhead that is not found in uniprocessor systems

Effect of the greater amount of time required to communicate between processors

- Lowers the speedup that programs running on the system
- Affects the amount of time required to communicate data between the parts of the program running on each processor

Effect of the greater amount of time required to communicate between processors

- Also affects the amount of time required for synchronization
- Synchronization typically implemented out of a sequence of interprocessor communications

Load balancing on Multiprocessors systems

Load balancing

- In many parallel applications, difficult to divide the program across the processors
- When each processor working the same amount of time not possible, some of the processors complete their tasks early and are then idle waiting for the others to finish

Algorithms that dynamically balance an application's load

- System with low interprocessor communication latencies can take advantage of by moving work from processors that are taking longer to complete their part of the program to other processors so that no processors are ever idle

Algorithms that dynamically balance an application's load

- Systems with longer communication latencies benefit less from these algorithms
- Communication latency determines how long it takes to move a unit of work from one processor to another

Super-linear Speedup of Performance in Multiprocessor System

Superlinear speedups

- Achieving speedup of greater than *n* on *n*-processor systems
- Each of the processors in an n-processor multiprocessor to complete its fraction of the program in less than l/nth of the program's execution time on a uniprocessor

Effect of increased cache size

- Reducing the average memory latency
- When the data required by the portion of the program that runs on each processor can fit in that processor's cache
- Improvement in performance

Better structure

- Some programs perform less work when executed on multiprocessor than they do when executed on a uniprocessor
- Achieve superlinear speedups

Example of Better structure in case of multiprocessor system

- Assume a program that search for the best answer to a problem by examining all of the possibilities
- Sometimes exhibit superlinear speedup because the multiprocessor version examines the possibilities in a different order, one that allows it to rule out more possibilities without examining them

Example of Better structure in case of multiprocessor system

- The multiprocessor version has to examine fewer total possibilities than the uniprocessor program, it can complete the search with greater than linear speedup

Example of No Better structure in case of multiprocessor system

- Rewriting the uniprocessor program to examine possibilities in the same order as the multiprocessor program would improve its performance, bringing the speedup back down to linear or less

Summary

We Learnt

- Extracting of parallelism by multiprocessors
- Speedup is generally near linear for less number of processors and becomes below linear with more systems
- Limitations in getting linear speedup─ Interprocessor communication, synchronization and load balancing

We Learnt

- Super-linear performance when the data can fit into caches alone in the portion of the program running on each processor

End of Lesson 01 on

Performance characteristics of Multiprocessor Architectures and Speedup