# Parallel Processors from Client to Cloud

## 1. Important terms

**Multiprocessor**
A computer system with at least two processors. This computer is in contrast to a uniprocessor, which has one, and is increasingly hard to find today.

**Task-level parallelism** or **Process-level parallelism**
Utilizing multiple processors by running independent programs simultaneously.

**Parallel processing program**
A single program that runs on multiple processors simultaneously.

**Cluster**
A set of computers connected over a local area network that function as a single large multiprocessor.

**Multicore microprocessor**
A microprocessor containing multiple processors ("cores") in a single integrated circuit. Virtually all microprocessors today in desktops and servers are multicore.

**Shared memory multiprocessor (SMP)**
A parallel processor with a single physical address space.

True or false: To benefit from a multiprocessor, an application must be concurrent.

## 2. Amdahl's Law

$$Execution\ time\ after\ improvement = \frac{Execution\ time\ affected\ by\ improvement}{Amount\ of\ improvement} + Execution\ time\ unaffected$$

$$\text{Speed-up} = \frac{\text{Execution time before}}{(\text{Execution time before} - \text{Execution time affected}) + \frac{\text{Execution time affected}}{\text{Amount of improvement}}}$$

$$\text{Speed-up} = \frac{1}{(1 - \text{Fraction time affected}) + \frac{\text{Fraction time affected}}{\text{Amount of improvement}}}$$

**Strong scaling**
Speedup achieved on a multiprocessor without increasing the size of the problem.

**Weak scaling**
Speedup achieved on a multiprocessor while increasing the size of the problem proportionally to the increase in the number of processors.

Strong scaling means measuring speed-up while keeping the problem size fixed.

Weak scaling means that the problem size grows proportionally to the increase in the number of processors.

True or false: Strong scaling is not bound by Amdahl's Law.

## 3. SISD, MIMD, SIMD, SPMD, and Vector

**SISD or Single Instruction stream, Single Data stream.**
A uniprocessor.

**MIMD or Multiple Instruction streams, Multiple Data streams.**
A multiprocessor.

**SIMD or Single Instruction stream, Multiple Data streams.**
The same instruction is applied to many data streams, as in a vector processor.

**SPMD Single Program, Multiple Data streams.**
The conventional MIMD programming model, where a single program runs across all processors.

True or false: As exemplified in the x86, multimedia extensions can be thought of as a vector architecture with short vectors that supports only contiguous vector data transfers.

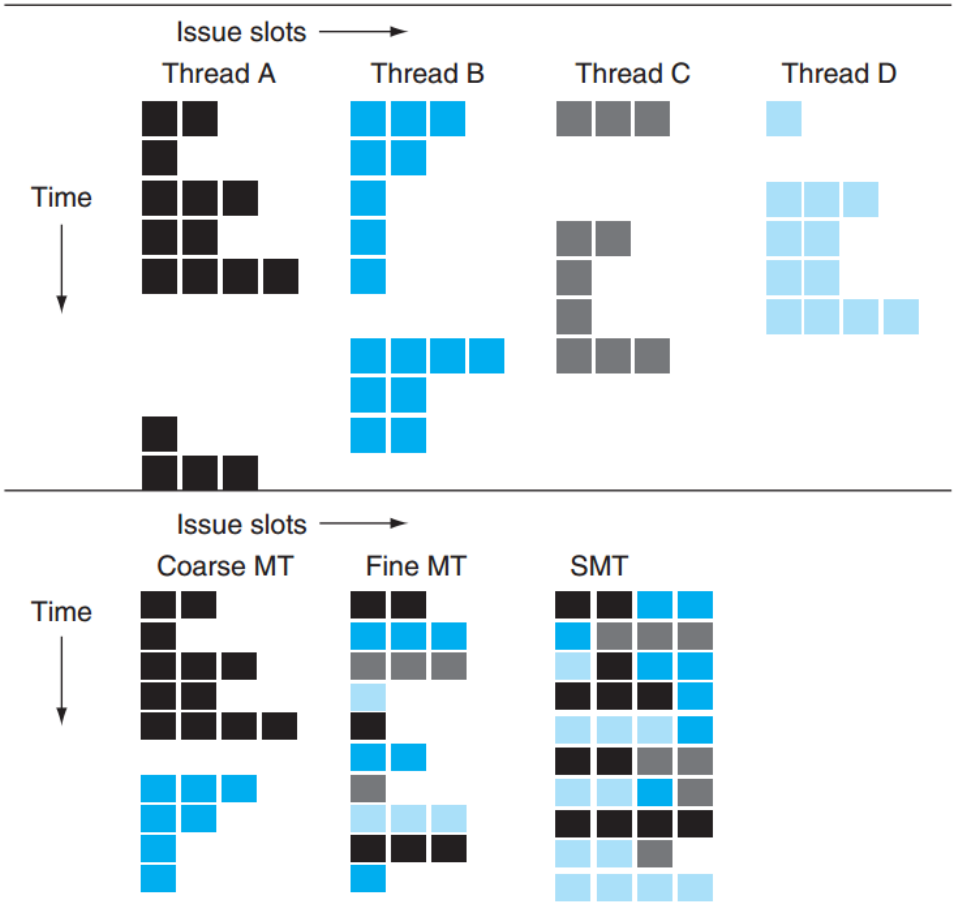
## 4. Multithreading

A related concept to MIMD, especially from the programmer's perspective, is **hardware multithreading**. While MIMD relies on multiple **processes** or **threads** to try to keep multiple processors busy, hardware multithreading allows multiple threads to share the functional units of a single processor in an overlapping fashion to try to utilize the hardware resources efficiently.

**Fine-grained multithreading**
A version of hardware multithreading that implies switching between threads after every instruction. If one thread stalls, others are executed.

**Coarse-grained multithreading**
A version of hardware multithreading that implies switching between threads only after significant events, such as a last-level cache miss.

**Simultaneous multithreading (SMT)**
A version of multithreading that lowers the cost of multithreading by utilizing the resources needed for multiple issue, dynamically scheduled microarchitecture.




True or false: Both multithreading and multicore rely on parallelism to get more efficiency from a chip.

True or false: Simultaneous multithreading (SMT) uses threads to improve resource utilization of a dynamically scheduled, out-of-order processor.

## 5. Shared Memory Multiprocessors

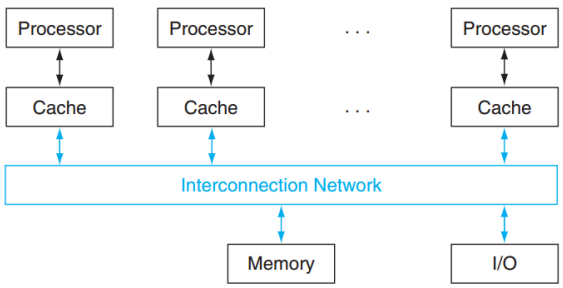



**FIGURE 6.7 Classic organization of a shared memory multiprocessor.**

How to efficiently programming on multiprocessor?

Don’t want to rewrite old programs in single processor.

Share memory among multiple cores.

True or false: Shared memory multiprocessors cannot take advantage of task-level parallelism.