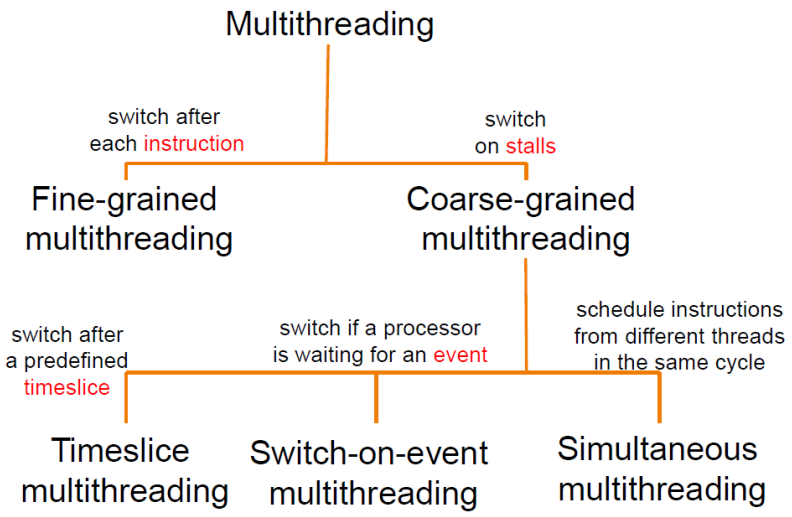
# 1 Chapter 3 - Parallel Computing Platforms

## 1.1 Source of Processor Performance Gain

Parallelism of various forms exist to have some performance gain

- Single Processor
  - Bit Level
    - We don't processs bit by bit, but instead word by word.
  - Instruction Level
    - Pipelining (parallelism across time)
      - Split instruction into different stages
      - allow multiple instructions to occupy different stages at same clock cycle.
      - Number of pipeline stages == Maximum achievable speedup
    - Superscalar (parallelism across space)
      - Duplicate the pipelines
      - Allow multiple instructions to pass through the same stage.
      - Hard to schedule, find instructions that can be run together. (Dymanic - Hardware decision, Static - Compiler decision)
      - Disadvantage: structural hazard
    - But is very limited, at most 2-3 instructions one time. due to data/control dependencies
    - Most CPUs have this. Usually we calculate Instructions per cycle, not Cycle-per-instruction.
  - Thread Level
    - Multithreading was originally a software mechanism
    - allow multiple parts of the same program to execute concurrently
    - Processor can provide hardware support for "thread context" like program counter and register (hyper-threading, etc..)
    - software threads can be executed in parallel




  - Process Level
    - Multiple processes work in parallel
    - independent memorry space, need special mechanism to communicate
    - Operating system provide IPC (Inter-Process communication) mechanism
    - Each process has independent context, can be mapped to multiple processor cores
- Multi Processor
  - Processor Level
    - Shared Memory
    - Distributed Memory

## 1.2 Flynn's Parallel Architecture Taxonomty

- SISD (Single Instruction Single Data)
  - A single instruction stream is executed
  - Each instruction work on single data
  - Most of the uniprocessor
- SIMD (Single Instruction Multiple Data)
  - A single stream of instructins
  - Each instruction work on multiple data
  - supercomputer during 1980s
  - To exploit data parallelism, also known as vector processor
  - Most modern processor has some form of SIMD
- MISD (Multiple Instruction Single Data)
  - Multiple stream of instructins
  - All instruction work on same data at any time
  - No actual implementaion, just here for completeness
- MiMD (Multiple Instruction Multiple Data)
  - Each Processing Unit fetch its own instruction
  - Each Processing operates on its data
  - Most popular model for multiprocessor

Variant - SIMD and MIMD. nVidia GPUs have a set of threads executing the same code (SIMD) and multiple set of threads executing in parallel (MIMD)

## 1.3 Multicore Architecture

### 1.3.1 Hierarchical design

Multiple cores share multiple caches. Cache size increase from the leaves to the root. Found in comon day desktop, GPUs.

### 1.3.2 Pipelined design

Data elements are processed by multiple (different) execution cores in a pipelined way. Bacuase same computation steps have to be applied to a long sequence of data elements. Found in networking application, or GPUs.

### 1.3.3 Network-based design

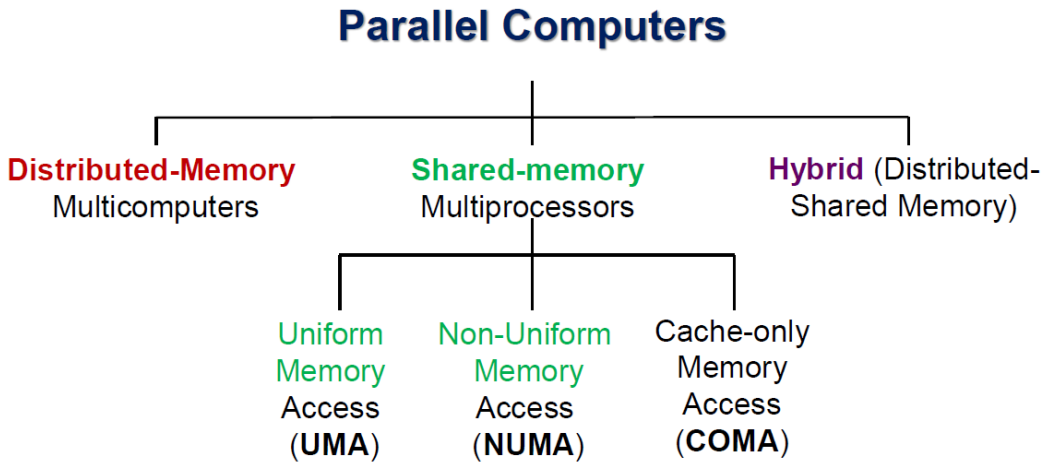
Cores and their local caches and memories are connected via an interconnection network Each core communicate with other cores / memory through this network.

## 1.4 Memory Organization

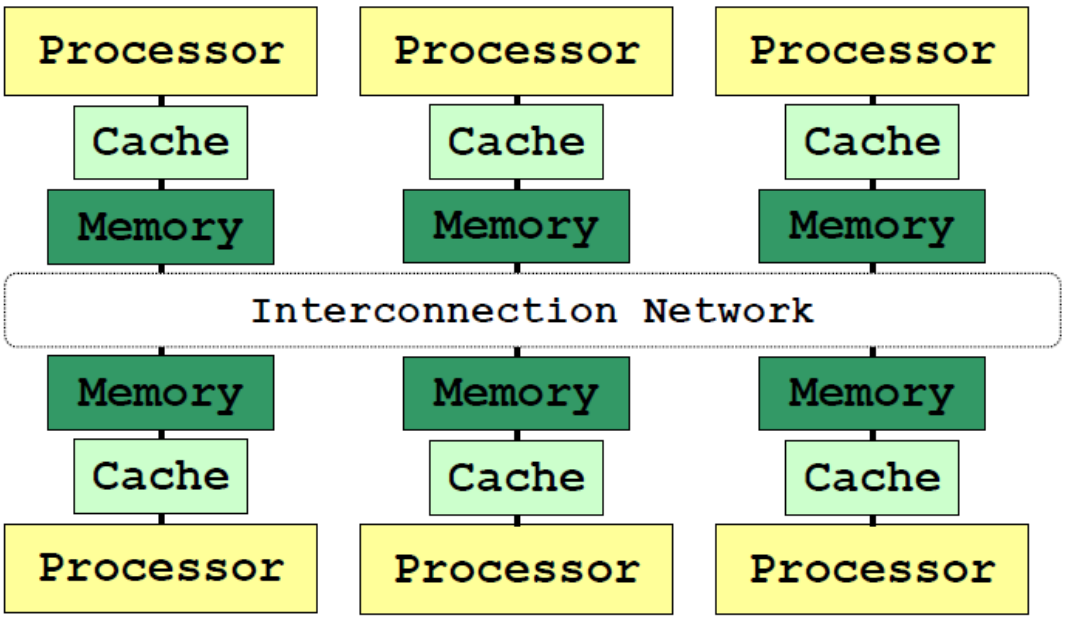
Memory can have consistency problem. If one processor updates value in memory, How will other cores also update their values. This also extends to the cache. Where we have cache coherence problem. If we update value in the cache, how will other processor also update the same value in their caches.

2 Factors differentiate shared memory systems

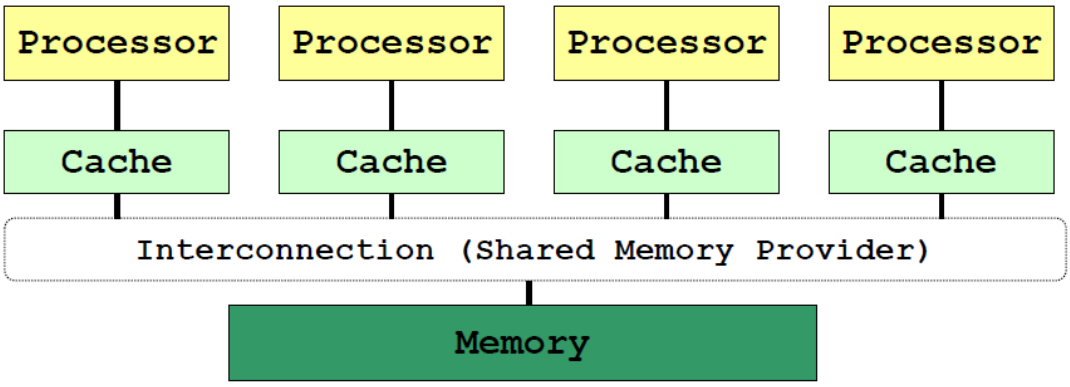
- Processor to Memory Delay (UMA/NUMA)
- presence of a local cache with cache coherence protocol (CC/NCC)




- Distributed Memory Systems
  - Each node is independent, communicate with message passing

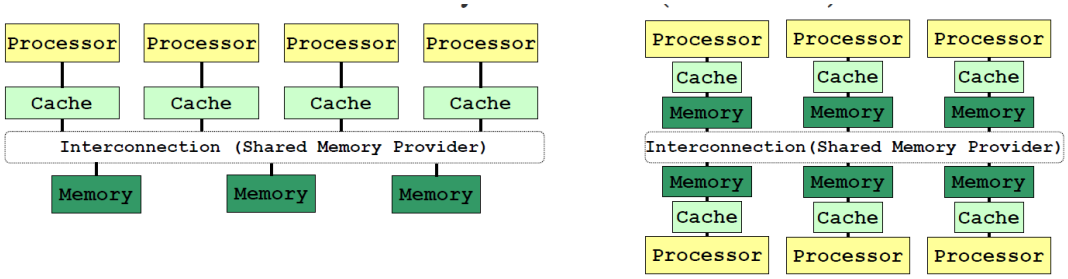



- Shared Memory
  - Uniform Memory Access (Time) (UMA)
    - Latency of accessing main memory is the same for each procesor
    - Good for low under of processors

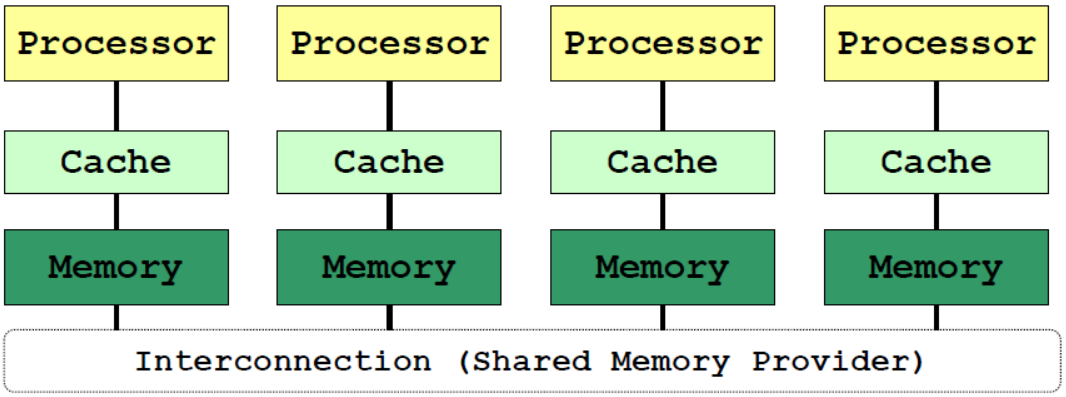



UMA organization for Multiprocessor

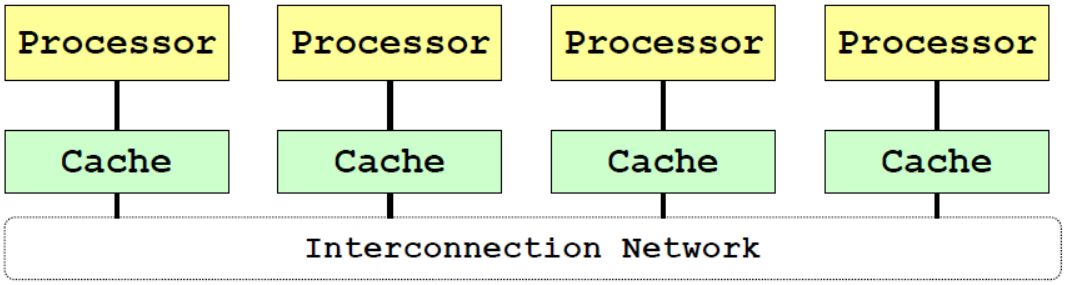
– Non-Uniform Memory Access
  * Physically distributed memory of all processing elemets are combined to form a global shared memory
  * Processor can access local memory faster than remote memory




– Cache Coherent Non Uniform Memory Access (ccNUMA)
  * Each node has cache to reduce contention




– Cache Only Memory Architecture
  * Each memory block works as cache memory
  * Uses cache coherence scheme to migrate datq




– Advantages

        * No need to partition code or data
        * No need to physically move data among processors, there is effficient communication
    - Disadvantages
        * Special synchronization constructs are required
        * Lack of scalability due to contention

- Hybird
    - Servers, use shared among distributed

# 2 Chapter 4