INTRODUCTION TO PARALLEL COMPUTING

Spring 2024

By Wissam H.

WHAT

What is parallelism

- Definition: Parallel computing is a type of computation in which many calculations or processes are carried out simultaneously.
- Importance: Significantly accelerates problem-solving and processing tasks.
- Key Characteristics: Divide and conquer, simultaneous execution, increased computational speed.
- Parallel computing is about hardware and software
  - Hardware : It is a sub-domain of computer architecture. Many kinds of computer organization has been designed to build parallel computers.
  - Software : parallel algorithms are developed to take benefic of parallel hardware. Many programming models exists too, with different balances of performance vs coding complexity.

WHY

Why we need parallelism in computing ?

- To solve complex problem : drug discovery, numerical intensive scientific applications.
- To solve time critical problem : Radar recognition, Weather prediction.
- To model realistically problems that are natively parallel.
- To use efficiently the performance of nowadays processors (CPU did not scale well in frequency, it implemented a multi-cores as an alternate way to improve performance)

HOW

How to build parallel computers

- We build parallel computing by putting many computing component in parallel.
- Parallel computing can be build at different levels, and in different ways
- CPU level parallelism :
  - Pipeline, Superscalar OOO Execution
  - Core multi-Threading (ex: intel Hyperthreading)
  - CPU multi-cores integration
- Server Boards with multiple-CPUs (2-4 CPUs)
- Symmetric-Multiprocessors SMP : 4-64 processors connected on a shared network (a BUS)

HOW TO BUILD PARALLEL COMPUTERS

Heavily parallel computers : hundreds to thousands of CPUs

A physically shared memory system can not handle loads for massive parallel computers, other architectures has been explored :

- MIMD (Multiple Instruction Multiple Data) :
  - Physically distributed logically shared Memory Architecture : Super computer is build using many cluster of smaller parallel computer. Each cluster has a memory at proximity
  - Logically Distributed Memory Architectures : Each processor has its own local memory, and communication is achieved through message passing
- SIMD (Simple Instruction Multiple Data), vectorial super computer:
  - CRAY multiprocessors
  - Nvidia Cuda GPU

HOW TO BUILD PARALLEL COMPUTERS

Exploring new areas of non conventional Von-Neumann Architectures

- Dataflow computers : a philosophical concept of computers where a maximum of parallelism is possible by making any operation start as soon as its operands are ready.
- Optical computers : Optical computing leverages properties of light for data transmission and processing, offering potential advantages over electronics in speed, bandwidth, power consumption, and noise tolerance.
- Biological computers : use biological components, such as DNA, or cells to perform computations. The idea is to leverage the inherent parallelism and massive storage capacity of biological systems for specific computational tasks
- Quantum computers : Quantum computers are a type of computing system that leverages the principles of quantum mechanics to perform computations. Unlike classical computers that use bits to represent information (which can be in a state of 0 or 1), quantum computers use quantum bits or qubits. Qubits can exist in multiple states simultaneously, thanks to a phenomenon called superposition, and can be entangled, enabling complex parallel computations

CHALLENGES

Parallel computing faces many challenges

- program dependencies (resources, data, control).
- Efficient Memory-Network system that scaleup well.
- Algorithm hard to parallelize.
- Good load balancing
- Data conflicts (coherency, consistency, false sharing)
- Synchronization
- Complexity to develop on a non conventional computing model (Quantum Computing)

METRICS

How to measure the performance of computers and parallel computers.

- Processing performance :
  - MIPS/GIPS : millions or billions instruction per second, ex
  - MFLOPS/GFLOPS : , ex : Intel Core i9-14900K is 1,95 GFLOPS
- Memory system performance : Bandwidth GBytes/sec, request latency in nano-sec
- CPU frequency => cycle timing = 1/Frequency
  - ex: 1Ghz CPU => cycle is 1 nano-sec
  - Ex: 2Ghz CPU => cycle is 0.5 nano-sec
- CPU ILP (Instruction Level Parallelism) : how many instructions are executed per cycle
  - ILP is dependent on CPU internal components : width of execution path, Execution units, Memory system performance
- Number of cores per CPU
- Number of CPUs in parallel computer.
- Over All performance estimation : performance = (nb. CPU x nb. Cores x ILP / cycle time) GIPS
  Note : ILP is variable an depends on many factors : CPU physical specification, program being executed, memory system,…
- Correct evaluation use benchmark instead of analytical estimation

PROGRAMMING MODELS FOR PARALLEL COMPUTING

Many programming model, generally related to specific parallel architecture

- Task parallelism
  - Heavy task (process) : UNIX fork
  - Lite task (Thread)
- Parallelism extension (using libraries)
  - Open MP
  - Open ACC
  - Cuda programming
  - Open MPI
- Automated parallelism (with compilers): limited
- New Parallel programming language : contains parallel loop constructs, like HPC (High-Performance Fortran)