Lawrence Livermore National Laboratory

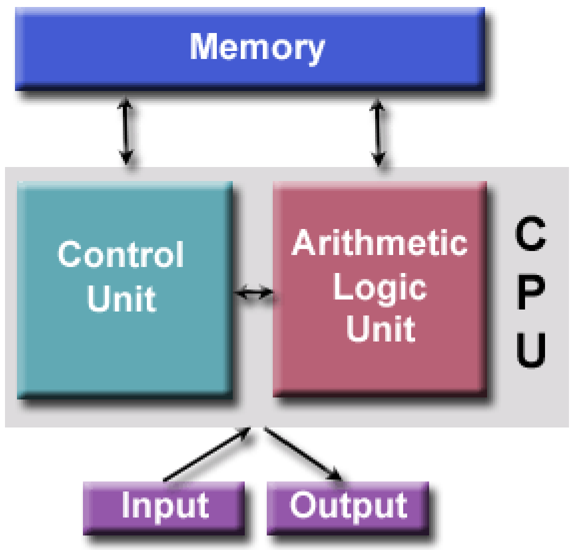
Introduction to Parallel Computing

Author: Blaise Barney - E-mail: blaiseb@llnl.gov
UCRL-MI-133316 - Last Modified: 08/17/2015 17:29:50

Concepts and Terminology

The von Neumann Architecture

- Named after the Hungarian mathematician/genius John von Neumann who first authored the general requirements for an electronic computer in his 1945 papers.
- Also known as "stored-program computer": both program instructions and data are kept in electronic memory. Differs from earlier computers which were programmed through "hard wiring".
- Since then, virtually all computers have followed this basic design:



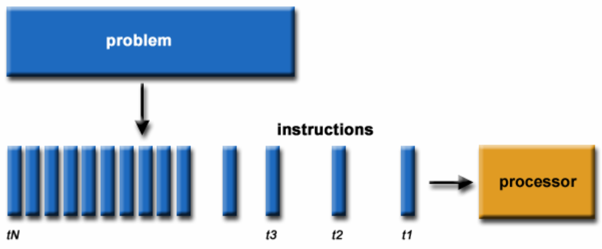
- Comprised of four main components:
  - Memory
  - Control Unit
  - Arithmetic Logic Unit
  - Input/Output
- Read/write, random access memory is used to store both program instructions and data
  - Program instructions are coded data which tell the computer to do something
  - Data is simply information to be used by the program
- Control unit fetches instructions/data from memory, decodes the instructions and then ***sequentially*** coordinates operations to accomplish the programmed task.
- Aritmetic Unit performs basic arithmetic operations
- Input/Output is the interface to the human operator

- More information on his other remarkable accomplishments: http://en.wikipedia.org/wiki/John_von_Neumann
- So what? Who cares?
  - Well, parallel computers still follow this basic design, just multiplied in units. The basic fundamental architecture remains the same.

What is Parallel Computing?

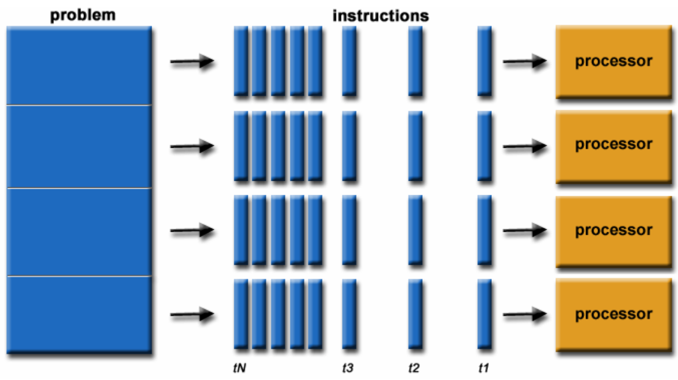
Serial Computing:

- Traditionally, software has been written for ***serial*** computation:
  - A problem is broken into a discrete series of instructions
  - Instructions are executed sequentially one after another
  - Executed on a single processor
  - Only one instruction may execute at any moment in time



Parallel Computing:

- In the simplest sense, ***parallel computing*** is the simultaneous use of multiple compute resources to solve a computational problem:
  - A problem is broken into discrete parts that can be solved concurrently
  - Each part is further broken down to a series of instructions
  - Instructions from each part execute simultaneously on different processors
  - An overall control/coordination mechanism is employed

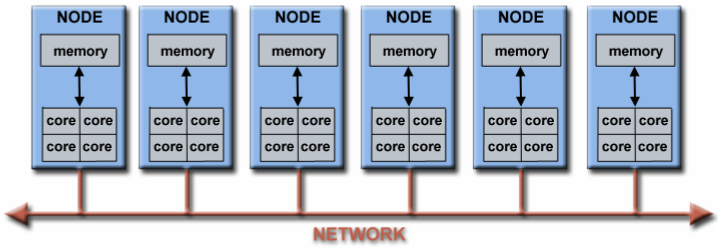


- The computational problem should be able to:
  - Be broken apart into discrete pieces of work that can be solved simultaneously;
  - Execute multiple program instructions at any moment in time;
  - Be solved in less time with multiple compute resources than with a single compute resource.
- The compute resources are typically:
  - A single computer with multiple processors/cores
  - An arbitrary number of such computers connected by a network

Parallel Computers:

- Virtually all stand-alone computers today are parallel from a hardware perspective:
  - Multiple functional units (L1 cache, L2 cache, branch, prefetch, decode, floating-point, graphics processing (GPU), integer, etc.)
  - Multiple execution units/cores
  - Multiple hardware threads

● Networks connect multiple stand-alone computers (nodes) to make larger parallel computer clusters.



● The majority of the world's large parallel computers (supercomputers) are clusters of hardware produced by a handful of (mostly) well known vendors.

Main Vendors' Market Share (%)

HP (35.8) / IBM (30.6) / Cray (12.4) / SGI (4.6) / Bull (3.6) / Dell (1.8) / Fujitsu (1.6)

Source: Top500.org

The IBM Blue Gene/Q installed atArgonne National Laboratory, near Chicago, Illinois.
Source: Wikipedia

Why Use Parallel Computing?

The Real World is Massively Parallel:

- In the natural world, many complex, interrelated events are happening at the same time, yet within a temporal sequence.
- Compared to serial computing, parallel computing is much better suited for modeling, simulating and understanding complex, real world phenomena.

Main Reasons:

● SAVE TIME AND/OR MONEY:

- In theory, throwing more resources at a task will shorten its time to completion, with potential cost savings.
- Parallel computers can be built from cheap, commodity components.

● SOLVE LARGER / MORE COMPLEX PROBLEMS:

- Many problems are so large and/or complex that it is impractical or impossible to solve them on a single computer, especially given limited computer memory.
- Example: "Grand Challenge Problems" (en.wikipedia.org/wiki/Grand_Challenge) requiring PetaFLOPS and PetaBytes of computing resources.
- Example: Web search engines/databases processing millions of transactions every second

● PROVIDE CONCURRENCY:

- A single compute resource can only do one thing at a time. Multiple compute resources can do many things simultaneously.
- Example: Collaborative Networks provide a global venue where people from around the world can meet and conduct work "virtually".

● TAKE ADVANTAGE OF NON-LOCAL RESOURCES:

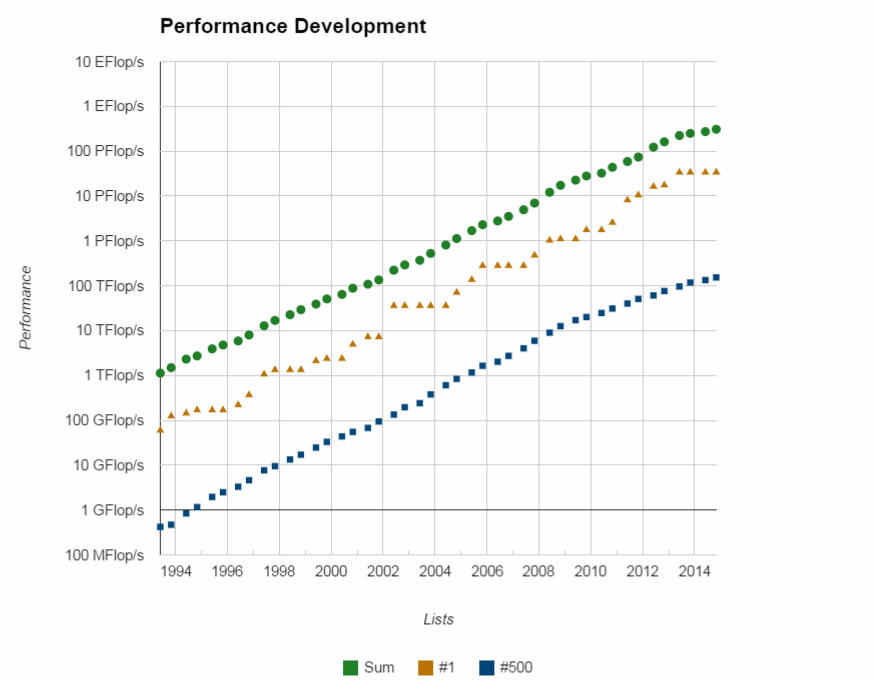
- Using compute resources on a wide area network, or even the Internet when local compute resources are scarce or insufficient.
- Example: SETI@home (setiathome.berkeley.edu) over 1.5 million users in nearly every country in the world. Source: www.boincsynergy.com/stats/ (June, 2015).
- Example: Folding@home (folding.stanford.edu) uses over 160,000 computers globally (June, 2015)

● **MAKE BETTER USE OF UNDERLYING PARALLEL HARDWARE:**

- Modern computers, even laptops, are parallel in architecture with multiple processors/cores.
- Parallel software is specifically intended for parallel hardware with multiple cores, threads, etc.
- In most cases, serial programs run on modern computers "waste" potential computing power.

The Future:

- During the past 20+ years, the trends indicated by ever faster networks, distributed systems, and multi-processor computer architectures (even at the desktop level) clearly show that ***parallelism is the future of computing***.
- In this same time period, there has been a greater than **500,000x** increase in supercomputer performance, with no end currently in sight.
- ***The race is already on for Exascale Computing!***
    -
    - Exaflop = 10^{18} calculations per second



Source: Top500.org

Who is Using Parallel Computing?

Science and Engineering:

- Historically, parallel computing has been considered to be "the high end of computing", and has been used to model difficult problems in many areas of science and engineering:

Atmosphere, Earth, Environment

Physics - applied, nuclear, particle, condensed matter, high pressure, fusion, photonics

Bioscience, Biotechnology, Genetics

Chemistry, Molecular Sciences

Geology, Seismology

Mechanical Engineering - from prosthetics to spacecraft

Electrical Engineering, Circuit Design, Microelectronics

Computer Science, Mathematics

Defense, Weapons

Industrial and Commercial:

- Today, commercial applications provide an equal or greater driving force in the development of faster computers. These applications require the processing of large amounts of data in sophisticated ways. For example:

"Big Data", databases, data mining

Oil exploration

Web search engines, web based business services

Medical imaging and diagnosis

Pharmaceutical design

Financial and economic modeling

Management of national and multi-national corporations

Advanced graphics and virtual reality, particularly in the entertainment industry

Networked video and multi-media technologies

Collaborative work environments