CPSC 4770/6770

Distributed and Cluster Computing

Lecture 1: Introduction to Parallel and Distributed Computing

New Palmetto Account Request

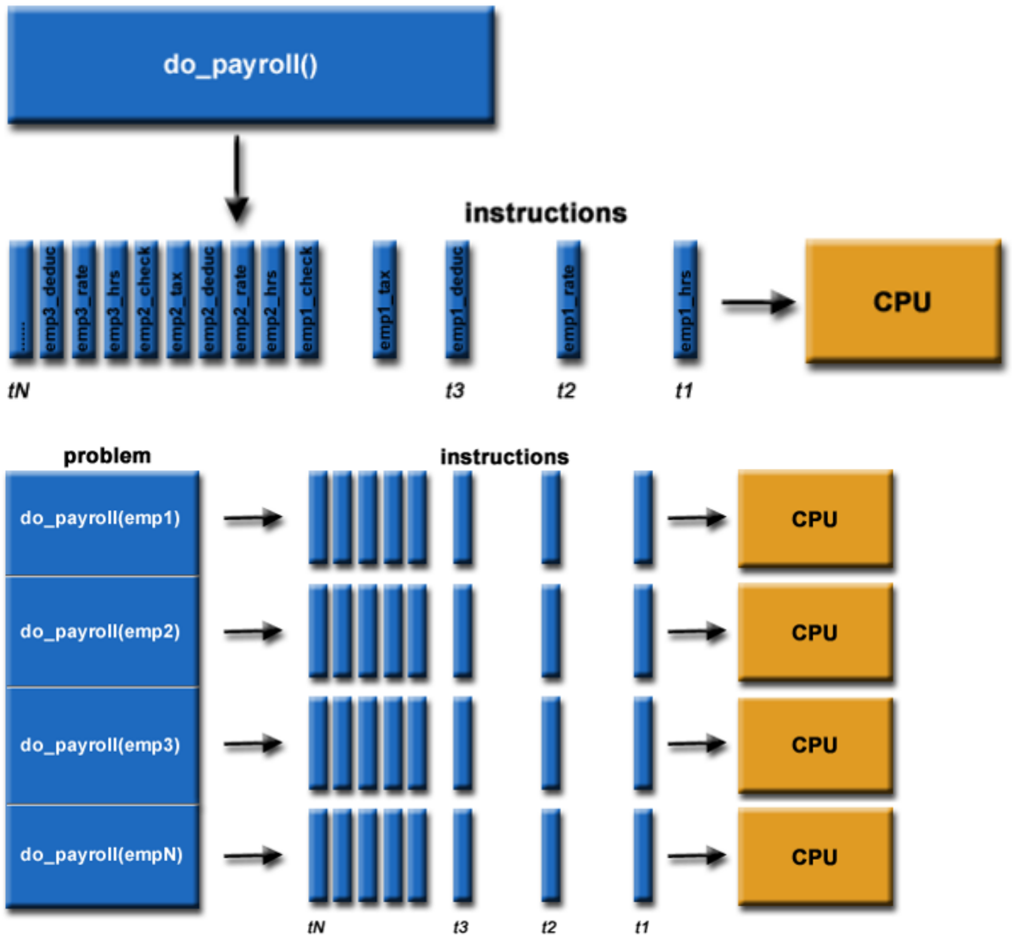
- Please fill out the account request from the link below to request a Palmetto account if you don't have one:

  https://www.palmetto.clemson.edu/palmetto/basic/new/
- Make sure that you correctly select an Account Type (*Educational*), and an appropriate Rank (*Undergraduate* or *Graduate*)
- Your Sponsor's Clemson Email should be that of your course' instructor (jin6@clemson.edu).
- ITHELP will approve the account once they received a correctly filled out form

What is Parallel Computing?

Sequential Computing

Parallel Computing



Source: https://computing.llnl.gov/tutorials/parallel_comp/

- The **problem** should be able to
  - Be broken apart into discrete pieces of work that can be solved simultaneously
  - Be solved in less time with multiple compute resources than with a single compute resource
- The **execution framework** should be able to
  - Execute multiple program instructions concurrently at any moment in time
- The **compute resource** might be
  - A single computer with multiple processors
  - An arbitrary number of computers connected by a network
  - A combination of both
  - A special computational component inside a single compute, separate from the main processors (GPU)

Measuring of Speed

- FLOPS: Floating-Point Operation Per Second
- MFLOPS = 1,000,000 FLOPS
- GFLOPS = 1,000,000,000 FLOPS
- TFLOPS = 1,000,000,000,000 FLOPS
  - Intel's ASCI Red was the first supercomputer in the world to achieve 1TFLOPS in 1993
- PFLOPS = 1,000,000,000,000,000 FLOPS
  - IBM RoadRunner was the first supercomputer to achieve 1 PFLOPS in 2008
  - US’s Summit is the current fastest supercomputer in the world with a peak performance of 200 PFLOPS
- EFLOPS = 1,000,000,000,000,000,000 FLOPS

The Demand of Computation Speed

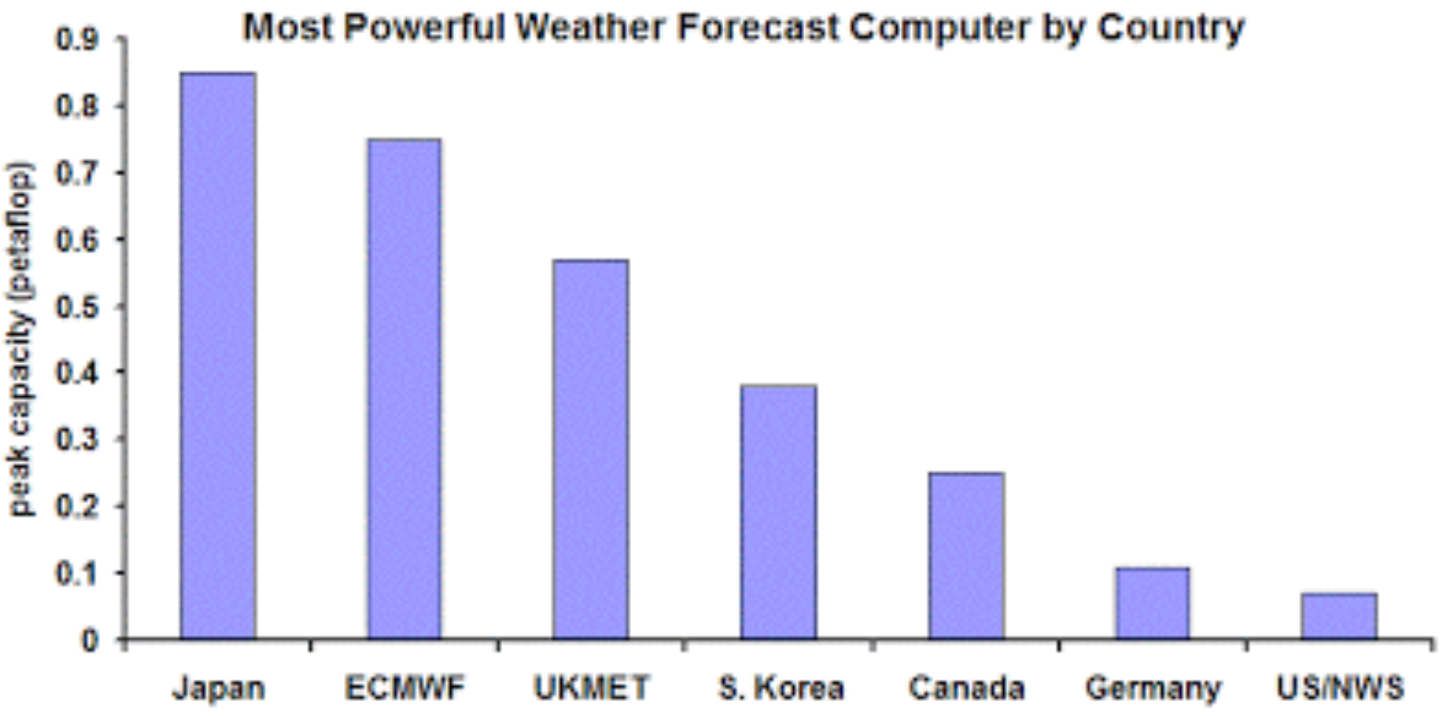
- Suppose whole global atmosphere divided into cells of size 1 mile x 1 mile x 1 mile to a height of 10 miles (10 cells high) – about 5×10^8 cells
- If each calculation requires 200 floating point operations, 10^{11} floating point operations necessary in one time step
- To forecast the weather for one week using 1-minute intervals, we need 60 (min/hr) x 24 (hr/day) x 7 (day) = 10,080 (~ 10^4 time steps)
- The week-long forecast would require $10^{11} \times 10^4 = 10^{15}$ operations
- Maximum operations per clock tick for x86_64 CPU architecture: 4 (flop)
- A single-core CPU with 2.5GHz clock speed will have:

    FLOPS = $2.5 \times 10^9 \times 4 = 10 \times 10^9$ (flops) = 10 (gflops)
- Single computer simulation:

    time = $10^{15}/(10 \times 10^9)/3600/24 = 11.5$ days
- To do this within five minutes:

    FLOPS = $10^{15}/300 = 3.35 \times 10^{12}$ (flops) = 3.4 (tflops)
- PS4: theoretical peak performance of 1.84 (TFLOPS) using 8 cores

Computational Power for Weather Forecast Agencies (2013)



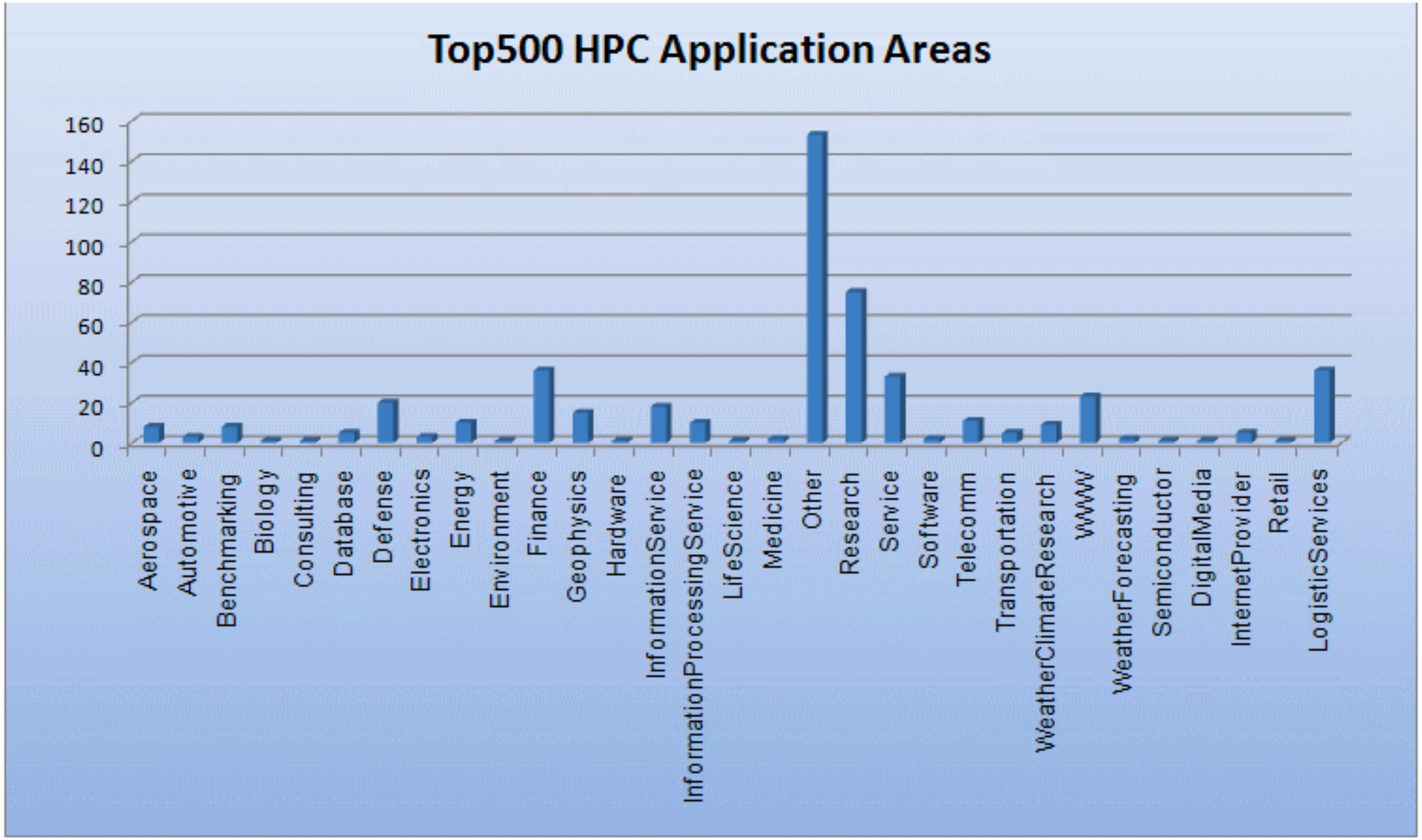
Source: https://blogs.agu.org/wildwildscience/2013/02/17/seriously-behind-the-numerical-weather-prediction-gap/

Catching up

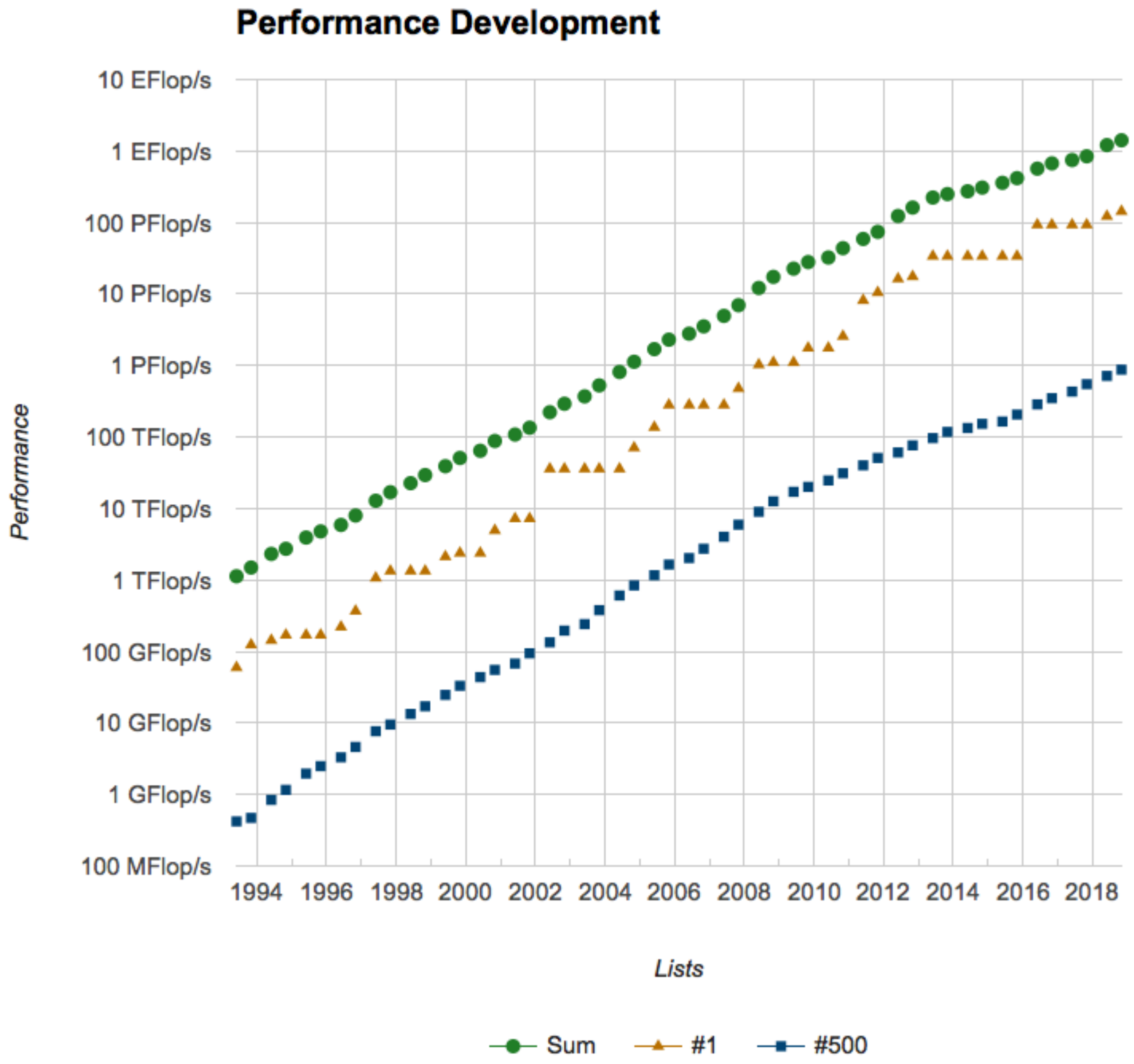
- Two new supercomputers, Luna and Surge
- 2.89 PFLOPS each for a total of 5.78 PFLOPS (previous generation is only 776 TFLOPS)
- Increase water quantity forecast from 4000 locations to 2.7 million locations (700-fold increase in spatial density)
- Can track and forecast 8 storms at any given time
- 44.5 million dollars investment

Source: https://www.noaa.gov/media-release/noaa-completes-weather-and-climate-supercomputer-upgrades

Who is Using Parallel Computing?



Development Over Time



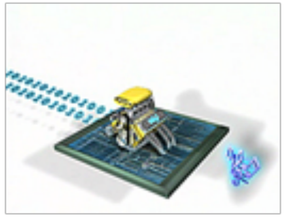
Source: https://www.top500.org/statistics/perfdevel/

Fraud Detection at PayPal

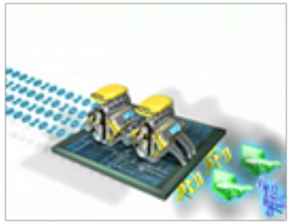
- 10M+ logins, 13M+ transactions, 300 variables per events
- ~4B inserts, ~8B selects
- MPI-like applications, Lustre Parallel File Systems, Hadoop
- Saved over $700M in fraudulent transactions during their first year of deployment

Sources: https://hpcuserforum.com/presentations/tuscon2013/IDCHPDABigDataHPC.pdf
https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/big-data-meets-high-performance-computing-white-paper.pdf

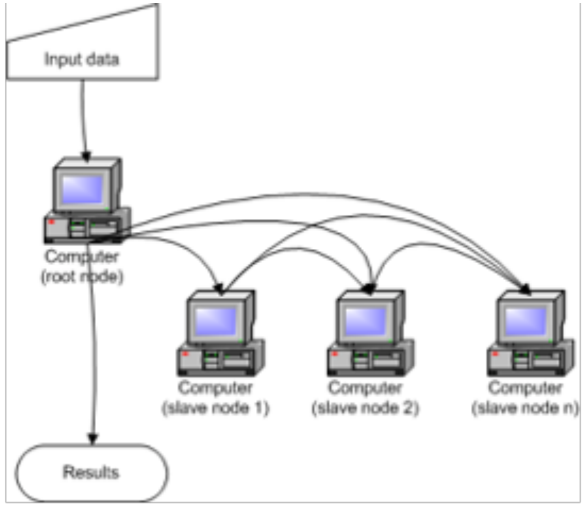
Progress of Parallel and Distributed Computing



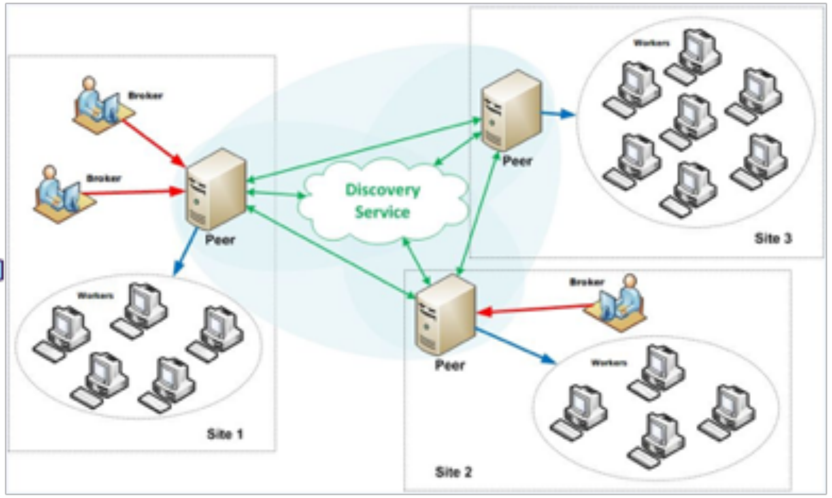
Single computer, single core



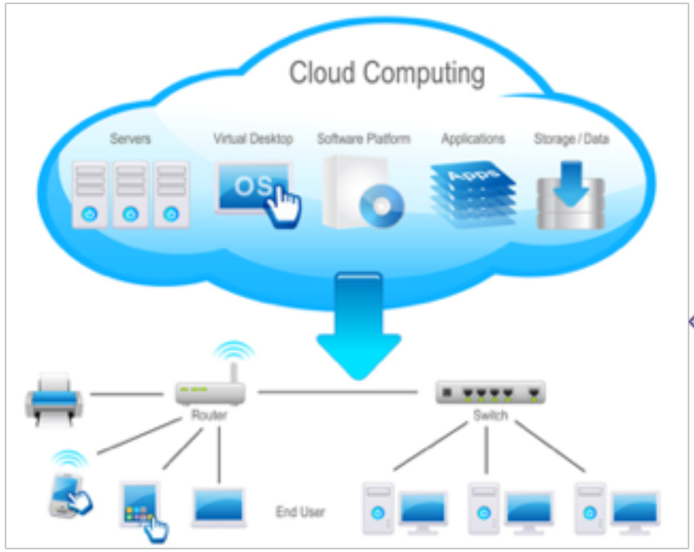
Single site, single computer, multiple cores



Single site, multiple computer, multiple cores



Multiple sites, multiple computer, multiple cores

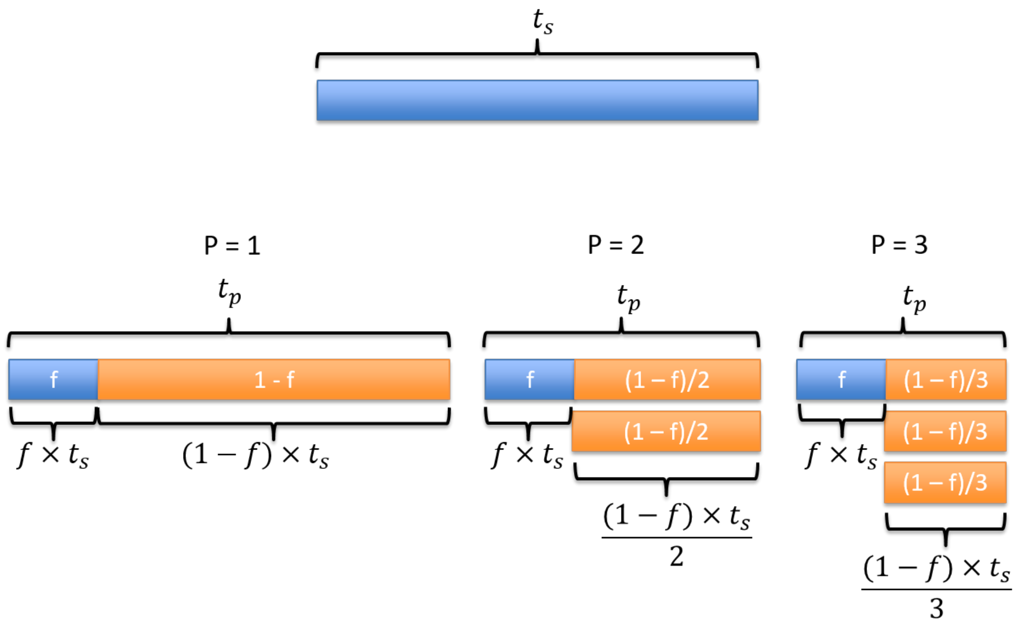


Multiple sites, multiple computer, multiple cores, virtual unified domain

"Distributed Computing Systems are a collection of individual computing devices that **can communicate with each other**." (Attiya and Welch, 2004)

Quantification of Performance Improvement

- **Parallel Speedup**: How much faster the program becomes once *some* computing resources are added
  - $S(p) = \frac{Sequential\ runtime}{Parallel\ runtime} = \frac{t_s}{t_p}$
  - Theoretical Max: Let f be the fraction of the program that is not parallelizable. Assume no communication overhead.



Amdahl's Law:

$$t_p = f t_s + \frac{(1-f)t_s}{p}$$

$$S(p) = \frac{t_s}{f t_s + \frac{(1-f)t_s}{p}} = \frac{p}{pf+1-f} = \frac{p}{(p-1)f+1} \leq p$$

Superlinear speedup: $S(p) > p$ may occur when:
- Poor sequential reference implementation
- Memory caching
- I/O Blocking

Quantification of Performance Improvement

- **Parallel Efficiency**: Ratio of performance improvement per individual unit of computing resource
  - $E = \frac{S(p)}{p} 100\% = \frac{1}{(p-1)f+1} 100\%$
  - E.g., Suppose that 4% of my application is serial. What is my predicted speedup according to Amdahl's Law on 5 processors?
    - f = 0.04, p = 5
    - $E = \frac{1}{(p-1)*f+1} 100\% = \frac{1}{(5-1)*0.04+1} 100\% = 86.206897\%$

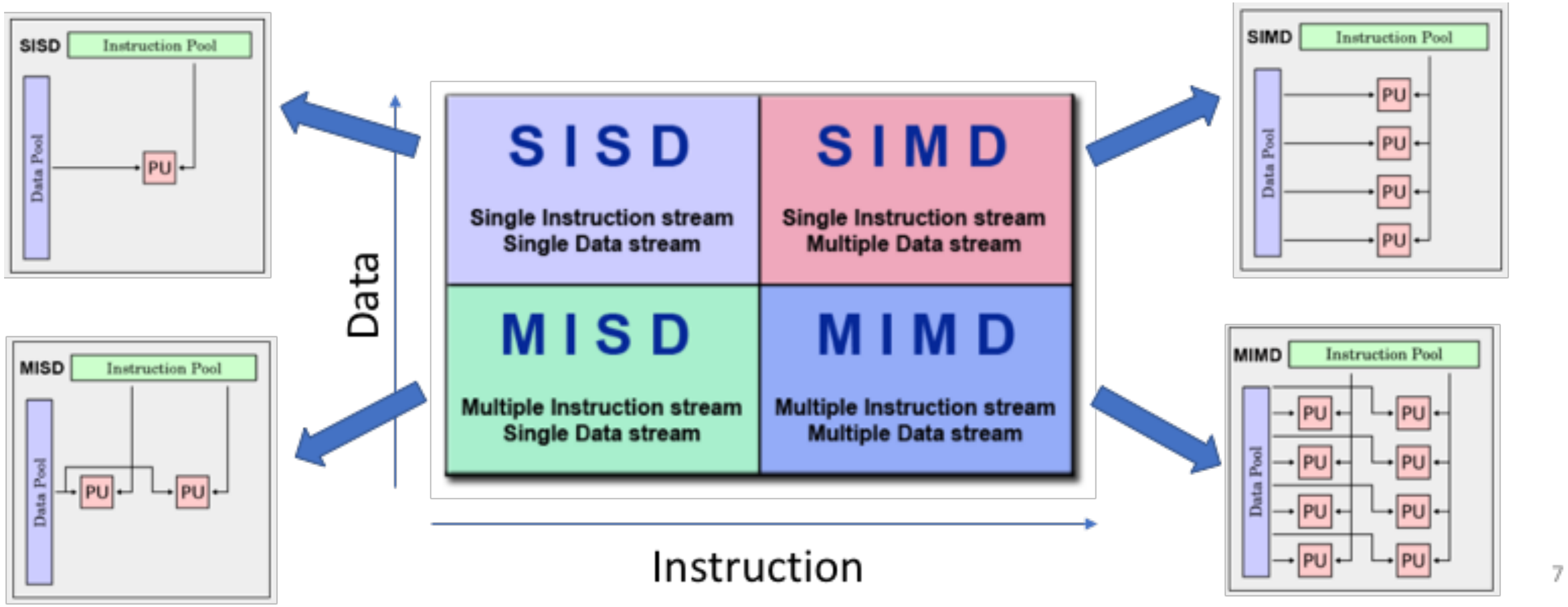
Limiting factors of $S(p)$:
- Non-parallelization code
- Communication overhead

Exercise

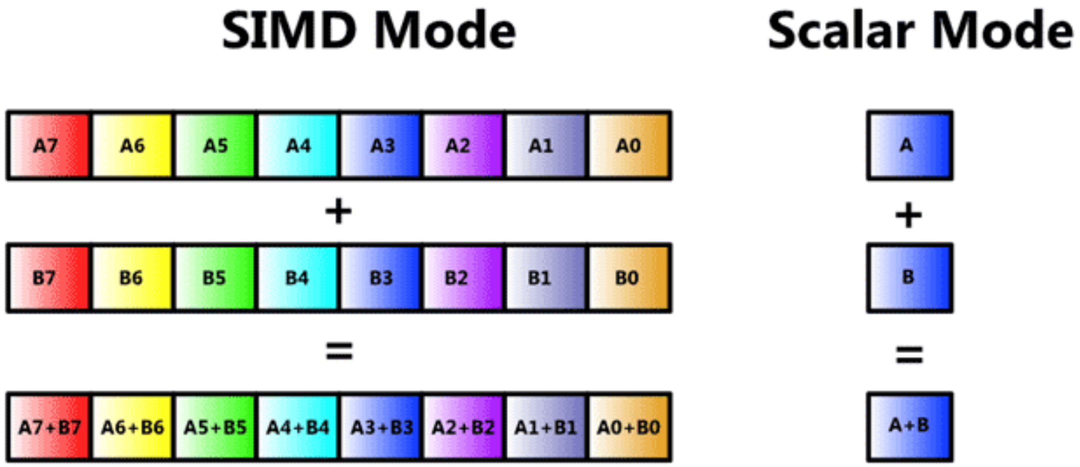
- Suppose that I get a speedup of 8 when I run my application on 10 processors.
  - According to Amdahl's Law, # What portion is serial?
  - What is the speedup on 20 processors? What is the efficiency?
  - What is the best speedup that I could hope for?

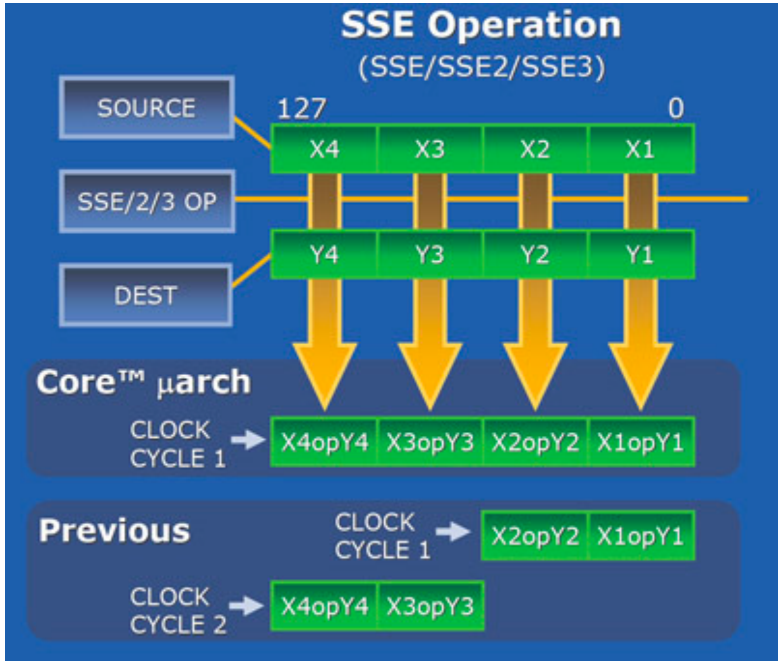
Types of Distributed Computing Systems

- Flynn's Taxonomy
  - Classifies multi-processor computer architectures into four different types according to how instructions and data flow through cores



Streaming SIMD



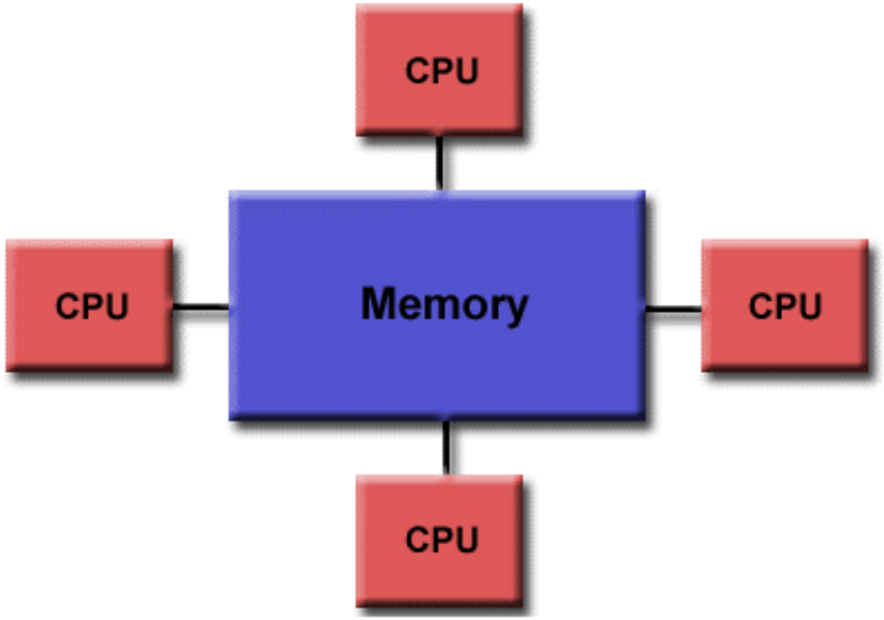


Source: https://software.intel.com/en-us/articles/introduction-to-intel-advanced-vector-extensions

- Intel's SSE
- AMD's 3DNow

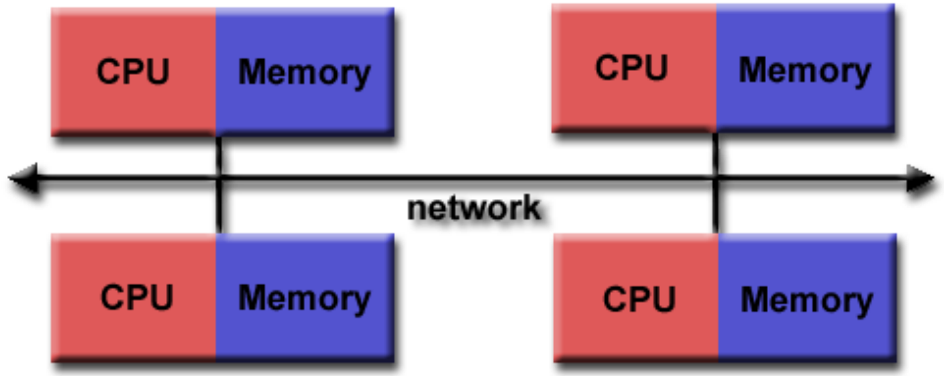
Shared Memory

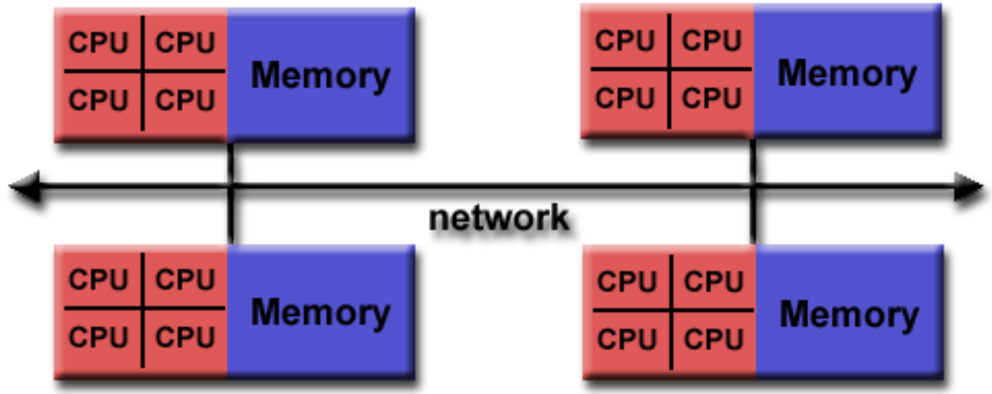
- One processor, multiple threads
- All threads have read/write access to the same memory
- Programming models:
  - Threads (pthread) – programmer manages all parallelism
  - OpenMP: Compiler extensions handle parallelization through in-code markers
  - Vendor libraries (e.g. Intel math kernel libraries)



Message Passing

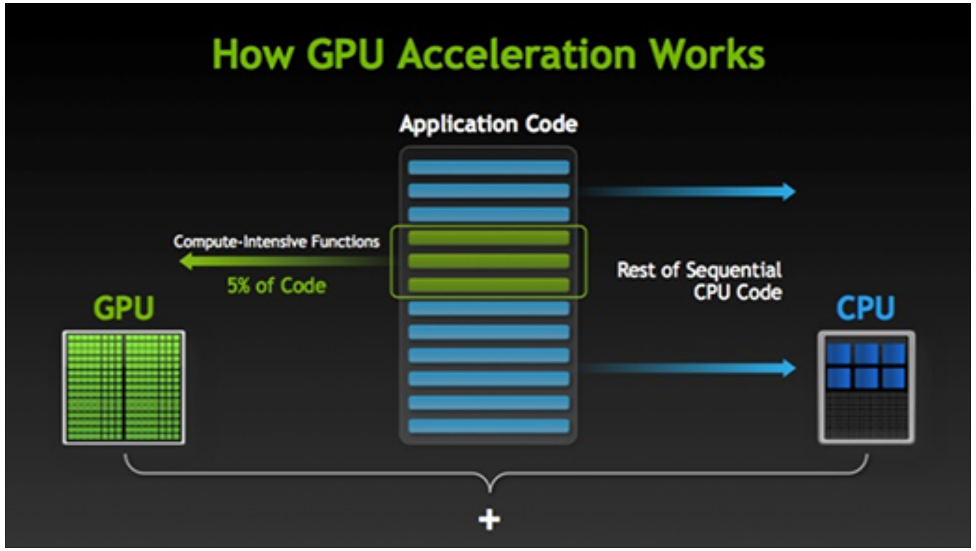
- Multiple processors, local memory
- Data passed via messages through communication network
- Programming models:
  - MPI: standardized message passing library
  - MPI + OpenMP (**hybrid model**)
  - MapReduce programming model





Heterogeneous Computing (Accelerators)

- GPU (Graphic Processing Units)
  - Processor unit on graphic cards designed to support graphic rendering (numerical manipulation)
  - Significant advantage for certain classes of scientific problem
  - CUDA – Library developed by NVIDIA for their GPUs
  - OpenACC – Standard developed by NVIDIA, Cray, and Portal Compiler (PGI)
  - OpenAMP – Extensions to Visual C++ (Microsoft) to direct computation to GPU
  - OpenCL – Set of standards by the group behind OpenGL
- FPGA (field programmable gate array)
  - Dynamically reconfigurable circuit board
  - Expensive, difficult to program
  - Power efficient, low heat



Benchmarking

- LINPACK (Linear Algebra Package): Dense Matrix Solver
- HPCC: High-Performance Computing Challenge
  - HPL (LINPACK to solve linear system of equation)
  - DGEMM (Double Precision General Matric Multiply)
  - STREAM (Memory bandwidth)
  - PTRANS (Parallel Matrix Transpose to measure processors communication)
  - RandomAccess (Random memory updates)
  - FFT (double precision complex discrete fourier transform)
  - Communication bandwidth and latency
- SHOC: Scalable Heterogeneous Computing - Non-traditional systems (GPU)
- TestDFSIO - I/O Performance of MapReduce/Hadoop Distributed File System

Ranking

- TOP500: Rank the supercomputers based on their LINPACK score
- GREEN500: Rank the supercomputers with emphasis on energy usage (LINPACK / power consumption)
- GRAPH500: Rank systems based on benchmarks designed for data-intensive computing