An introduction to High Performance Computing and its Applications

Ashish P. Kuvelkar
Senior Director (HPC- Tech)
C-DAC, Pune

Outline

- Introduction to HPC
- Architecting a HPC system
- Approach to Parallelization
- Parallelization Paradigm
- Applications in area of Science and Engineering

What is a HPC?

High Performance Computing

- Set of Computing technologies for very fast numeric simulation, modeling and data processing
- Employed for specialised applications that require lot of mathematical calculations
- Using computer power to execute a few applications extremely fast

What is HPC?(continued)

Definition 1

- High Performance Computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly.
- A supercomputer is a system that performs at or near the currently highest operational rate for computers.

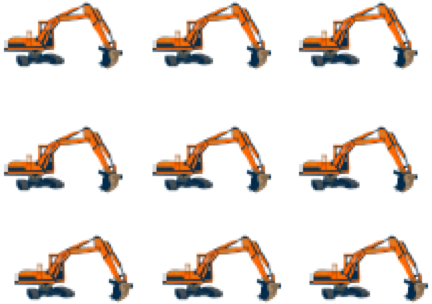
Definition 2 (Wikipedia)

- High Performance Computing (HPC) uses Supercomputers and Computer Clusters to solve advanced computation problems.

Evolution of Supercomputers

- Supercomputer in the 1980s and 90s
  - Custom-built computer systems
  - Very expensive

- Supercomputer after 1990s
  - Build using commodity off-the-shelf" components
  - Uses cluster computing techniques

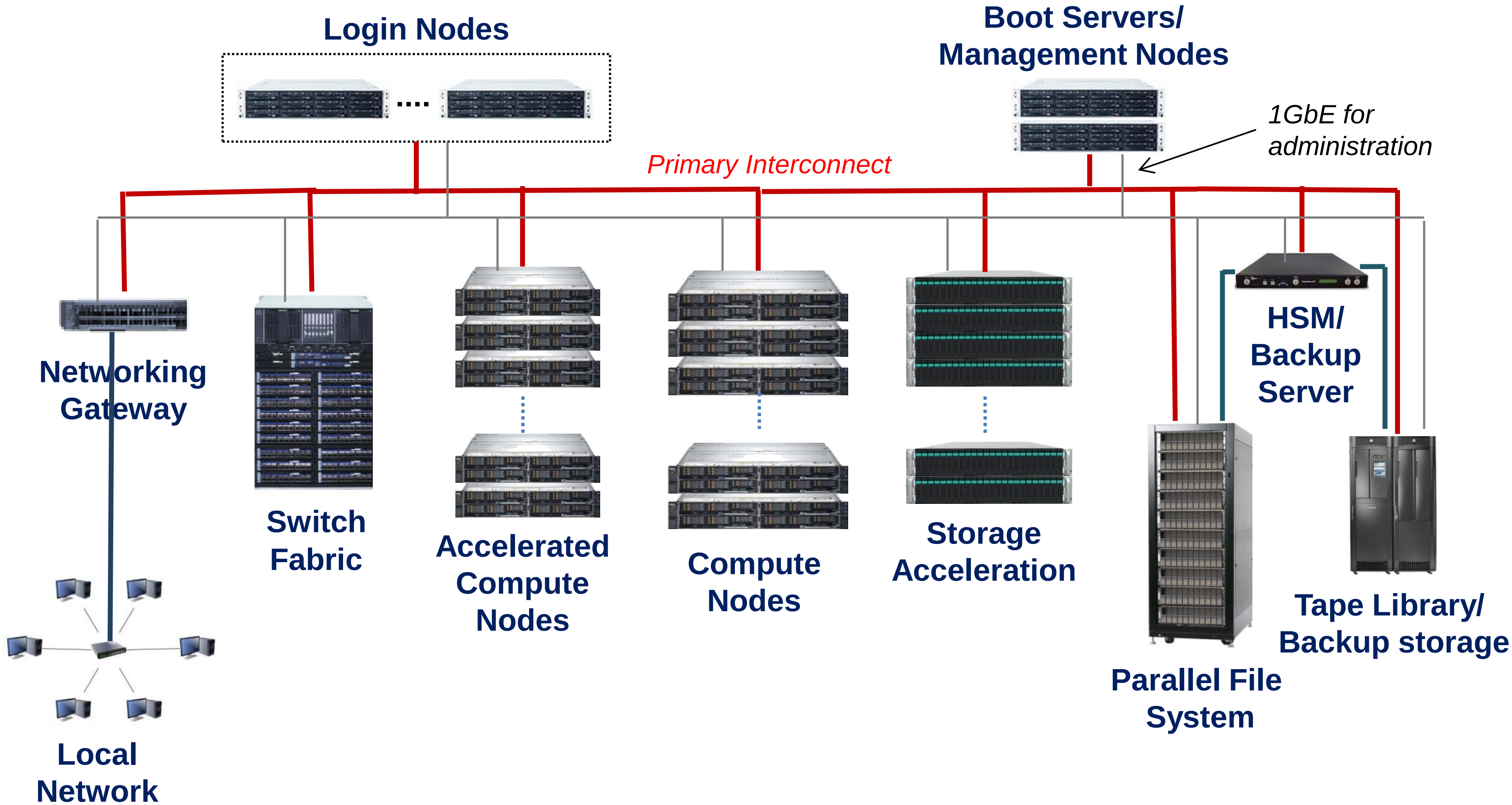


Supercomputers

Cray Supercomputer

PARAM Yuva II

Components of Cluster



HPC Software Stack

HPC Programming Tools	Performance Monitoring	HPCC, IOR, PAPI/IPM, NPB, Netperf
HPC Programming Tools	Development Tools	Alliena DDT/ TAU, Intel Cluster Studio/IBM XC, PGI (PGI SDK), GNU Compiler
HPC Programming Tools	Application Libraries	Ferret/GRADS/PARA view/VISIT, MVAPICH2/ OpenMPI, ACML/ESSL, MPSS/CUDA, BLAS, LAPACK
Middleware Applications and Management	Resource Management/ Job Scheduling	SLURM, Grid Engine, MOAB, Altair PBS Pro, IBM Platform LSF, Torque/ Maui
Middleware Applications and Management	File System	NFS, Local FS (ext3, ext4, XFS), GPFS, Lustre
Middleware Applications and Management	Provisioning	XCAT / ROCKS / C-DAC Developed tools
Middleware Applications and Management	Cluster Monitoring	XCAT / ROCKS / C-DAC Developed tools
Operating Systems	Operating System	Linux (Red Hat, CentOS, SUSE)

Single CPU Systems

- Can run a single stream of code
- Performance can be improvement through
    - Increasing ALU width
    - Increasing clock frequency
    - Making use of pipelining
    - Improved compilers
- But still, there is a limit to each of these techniques
  - Parallel computing, provides relief

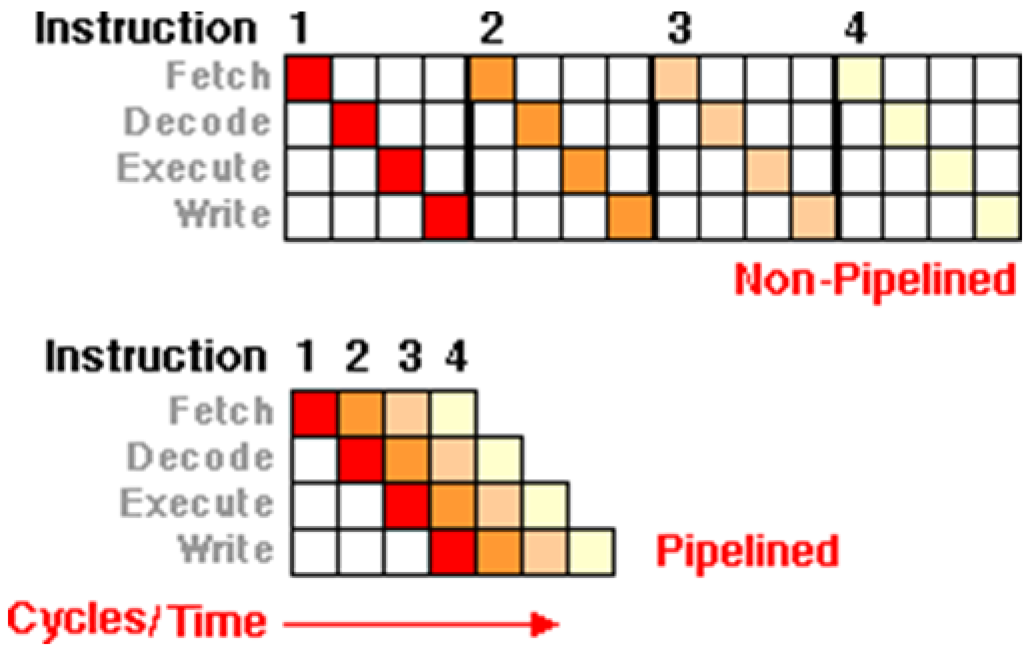
Why use Parallel Computing?

- Overcome limitations of single CPU systems
  - Sequential systems are slow
    - Calculations make take days, weeks, years
    - More CPUs can get job done faster
  - Sequential systems are small
    - Data set may not fit in memory
    - More CPUs can give access to more memory
- So, the advantages are
  - Save time
  - Solve bigger problems

Single Processor Parallelism

- Instruction level Parallelism is achieved through
  - Pipelining
  - Superscaler implementation
  - Multicore architecture
  - Using advanced extensions

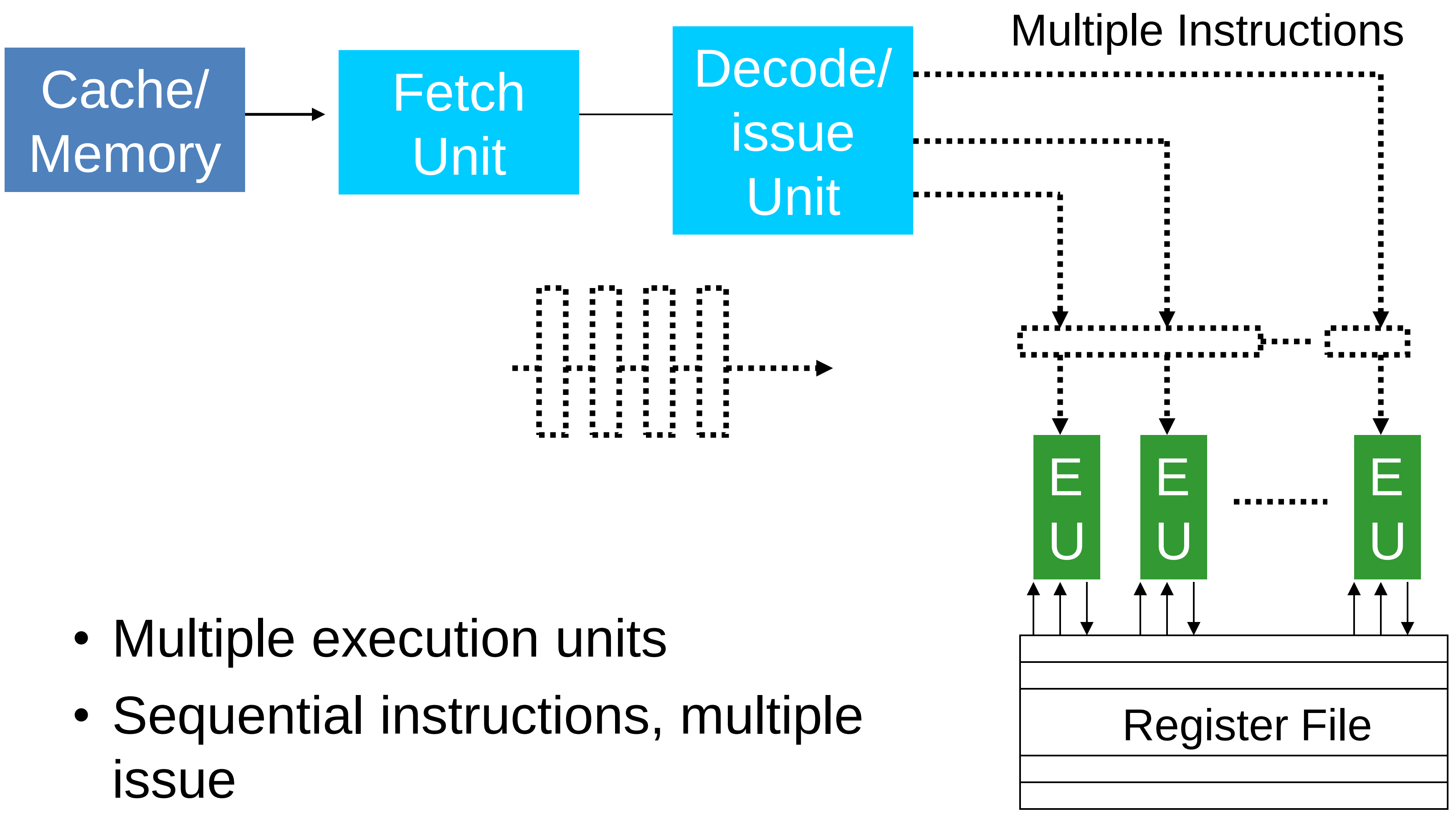
Pipelined Processors



- A new instruction enters every clock
- Instruction parallelism = No. of pipeline stages

Diagram Souce: Quora

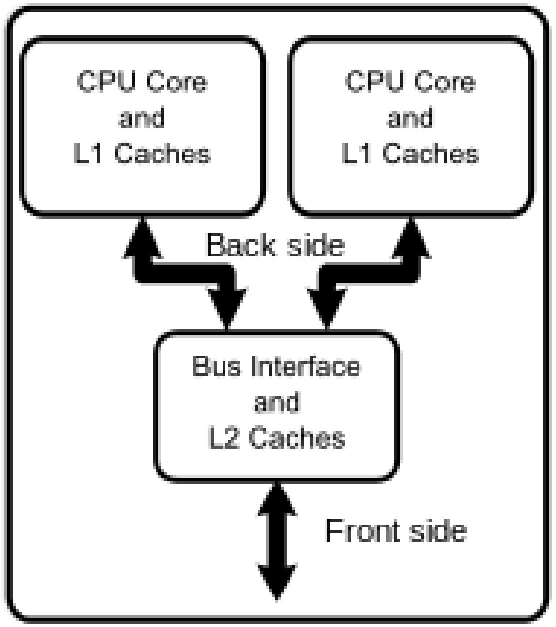
Superscaler



- Multiple execution units
- Sequential instructions, multiple issue

Multicore Processor

- Single computing component with two or more independent processing units
- Each unit is called cores, which read and execute program instructions



Source: Wikipedia.

Advanced Vector eXtensions

- Useful for algorithms that can take advantage of SIMD
- AVX were introduced by Intel and AMD in x86
- Using AVX-512, applications can pack
  - 32 double precision or 64 single precision floating point operations or
  - eight 64-bit and sixteen 32-bit integers
- Accelerates performance for workloads such as
  - Scientific simulations, artificial intelligence (AI)/deep learning, image and audio/video processing

Parallelization Approach

Means of achieving parallelism

- Implicit Parallelism
  - Done by the compiler and runtime system
- Explicit Parallelism
  - Done by the programmer

Implicit Parallelism

- Parallelism is exploited implicitly by the compiler and runtime system
  - Automatically detects potential parallelism in the program
  - Assigns the tasks for parallel execution
  - Controls and synchronizes execution

(+) Frees the programmer from the details of parallel execution

(+) it is a more general and flexible solution

(-) very hard to achieve an efficient solution for many applications

Explicit Parallelism

- It is the programmer who has to
  - Annotate the tasks for parallel execution
  - Assign tasks to processors
  - Control the execution and the synchronization points

(+) Experienced programmers achieve very efficient solutions for specific problems

(-) programmers are responsible for all details

(-) programmers must have deep knowledge of the computer architecture to achieve maximum performance.

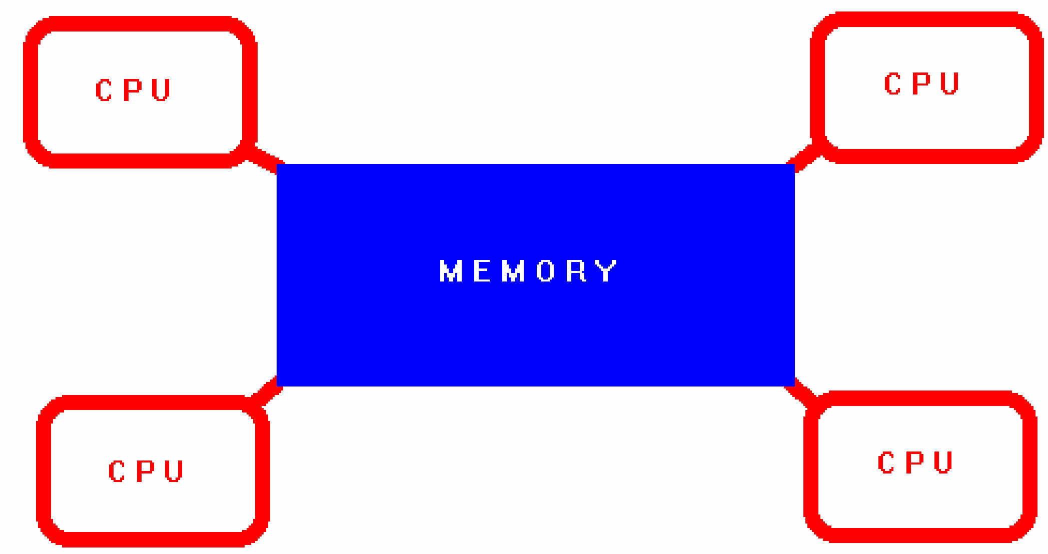
Explicit Parallel Programming Models

Two dominant parallel programming models

- Shared-variable model
- Message-passing model

Shared Memory Model

- Uses the concept of single address space
- Typically SMP architecture is used
  - Scalability is not good



Shared Memory Model

- Multiple threads operate independently but share same memory resources
- Data is not explicitly allocated
- Changes in a memory location effected by one process is visible to all other processes
- Communication is implicit
- Synchronization is explicit

Advantages & Disadvantages of Shared Memory Model

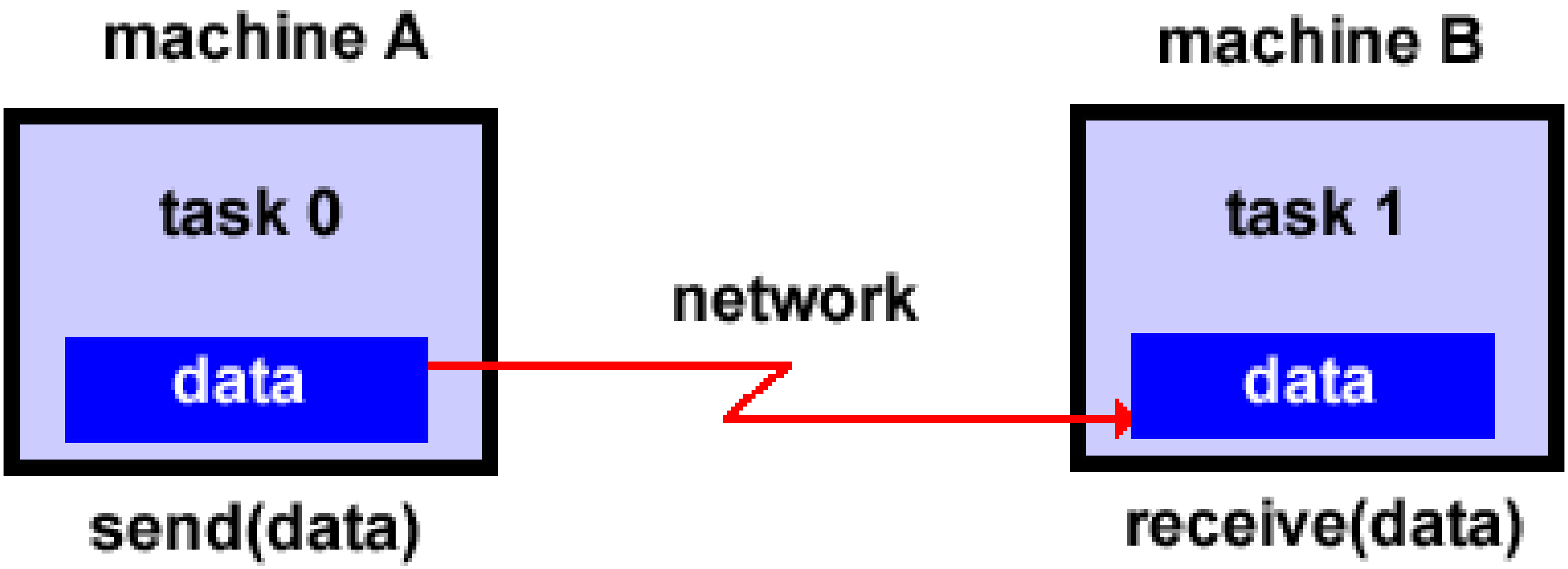
Advantages :

- Data sharing between threads is fast and uniform
- Global address space provides user friendly programming

Disadvantages :

- Lack of scalability between memory and CPUs
- Programmer is responsible for specifying synchronization, e.g. locks
- Expensive

Message Passing Model



Characteristics of Message Passing Model

- Asynchronous parallelism
- Separate address spaces
- Explicit interaction
- Explicit allocation by user

How Message Passing Model Works

- A parallel computation consists of a number of processes
- Each process has purely local variables
- No mechanism for any process to directly access memory of another
- Sharing of data among processes is done by explicitly message passing
- Data transfer requires cooperative operations by each process

Usefulness of Message Passing Model

- Extremely general model
- Essentially, any type of parallel computation can be cast in the message passing form
- Can be implemented on wide variety of platforms, from networks of workstations to even single processor machines
- Generally allows more control over data location and flow within a parallel application than in, for example the shared memory model
- Good scalability

Parallelization Paradigms

Ideal Situation !!!

- Each Processor has a Unique work to do
- Communication among processes is largely unnecessary
- All processes do equal work

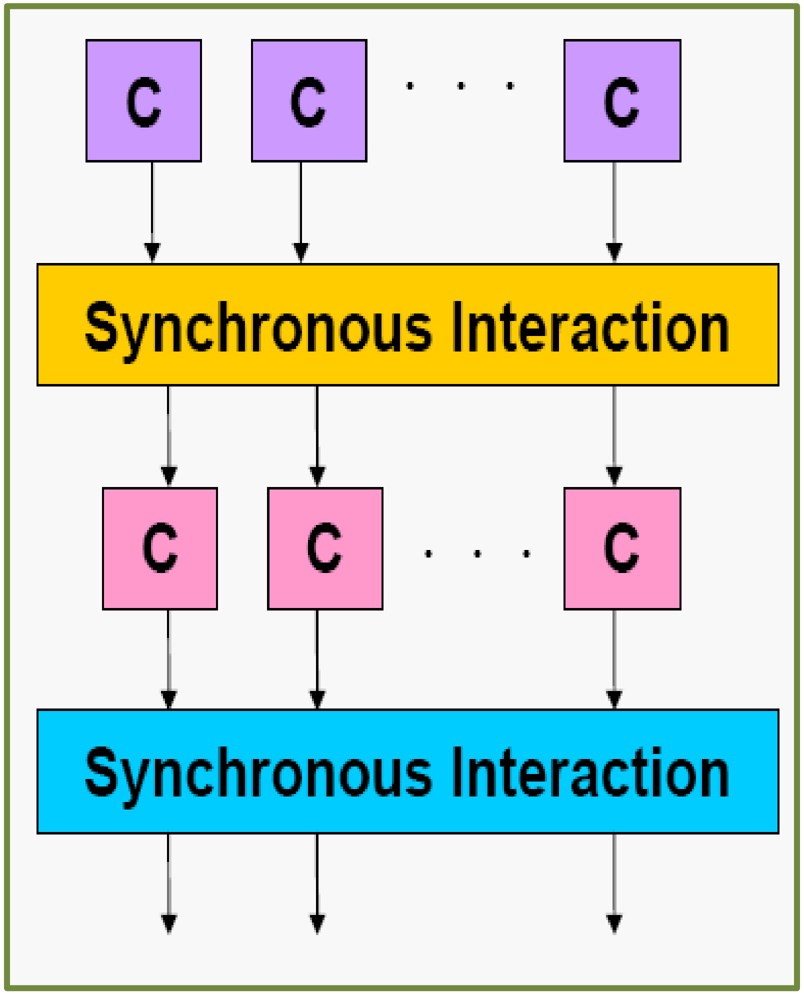
Writing parallel codes

- Distribute the data to memories
- Distribute the code to processors
- Organize and synchronize the workflow
- Optimize the resource requirements by means of efficient algorithms and coding techniques

Parallel Algorithm Paradigms

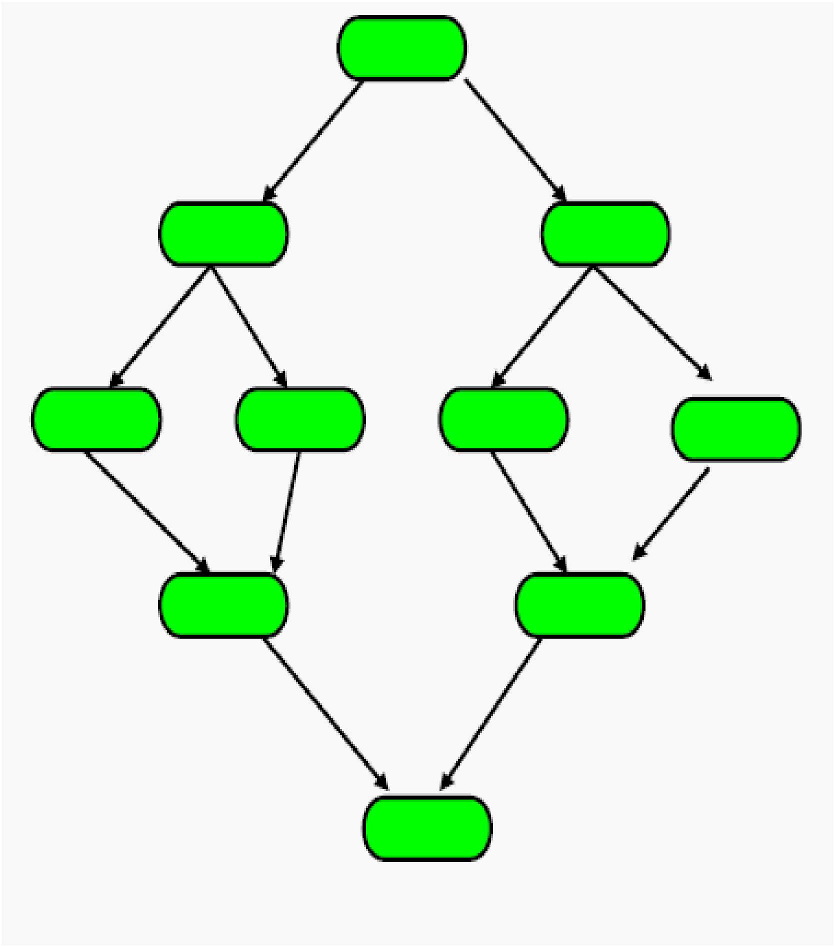
- Phase parallel
- Divide and conquer
- Pipeline
- Process farm
- Domain Decomposition

Phase Parallel Model



- The parallel program consists of a number of super steps, and each has two phases.
- In a computation phase, multiple processes each perform an independent computation.
- In interaction phase, the processes perform one or more synchronous interaction operations, such as a barrier or a blocking communication.

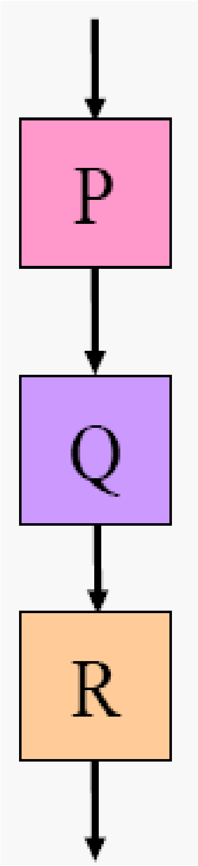
Divide and Conquer model



- A parent process divides its workload into several smaller pieces and assigns them to a number of child processes.
- The child processes then compute their workload in parallel and the results are merged by the parent.
- This paradigm is very natural for computations such as quick sort.

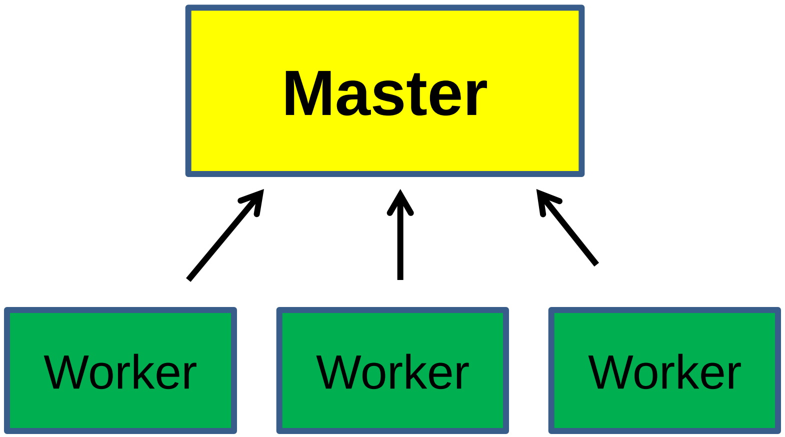
Pipeline Model

Data Stream



- In pipeline paradigm, a number of processes form a virtual pipeline.
- A continuous data stream is fed into the pipeline, and the processes execute at different pipeline stages simultaneously.

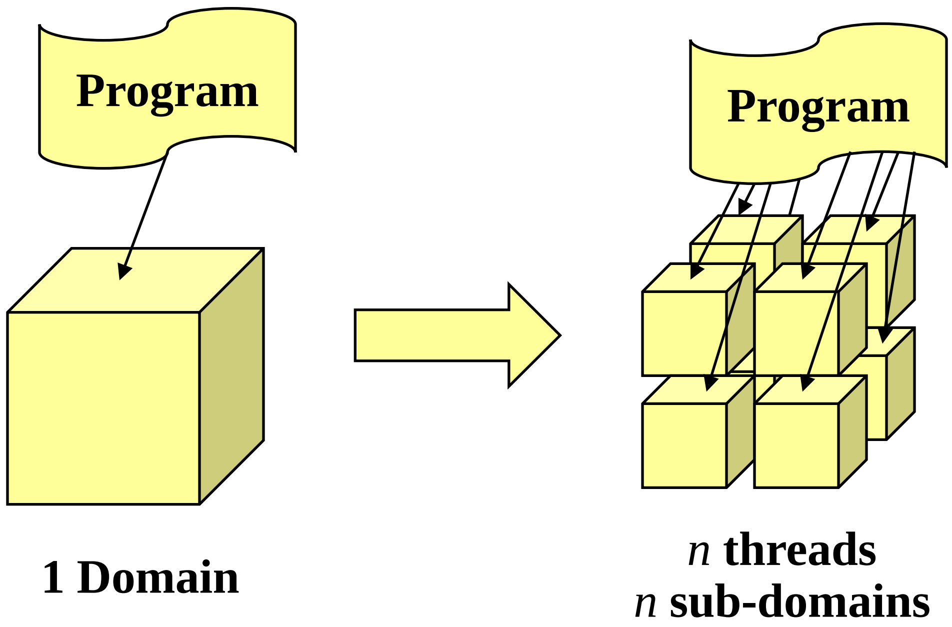
Process Farm Model



- Also known as the master-worker paradigm.
- A master process executes the essentially sequential part of the parallel program
- It spawns a number of worker processes to execute the parallel workload.
- When a worker finishes its workload, it informs the master which assigns a new workload to the slave.

- The coordination is done by the master.

Domain Decomposition



This methods solve a boundary value problem by splitting it into smaller boundary value problems on subdomains and iterating to coordinate the solution between adjacent subdomains.

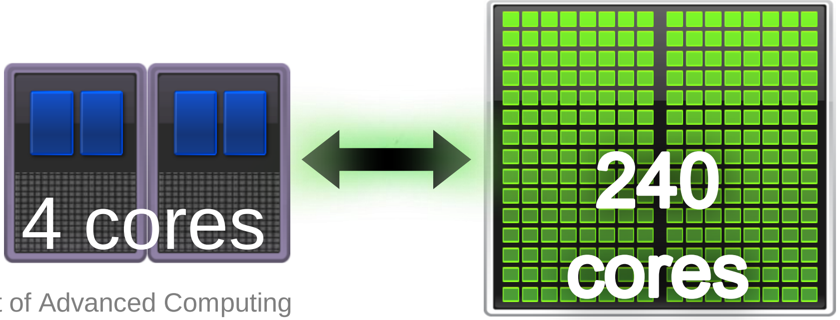
Desirable Attributes for Parallel Algorithms

- Concurrency
  - Ability to perform many actions simultaneously
- Scalability
  - Resilience to increasing processor counts
- Data Locality
  - High ratio of local memory accesses to remote memory accesses (through communication)
- Modularity:
  - Decomposition of complex entities into simpler components

Heterogeneous Computing: GPUs + CPUs

Massive processing power introduces I/O challenge

- Getting data to and from the processing units can take as long as the processing itself
- Requires careful software design and deep understanding of algorithms and architecture of
  - Processors (Cache effects, memory bandwidth)
  - GPU accelerators
  - Interconnects (Ethernet, IB, 10 Gigabit Ethernet),
  - Storage (local disks, NFS, parallel file systems)

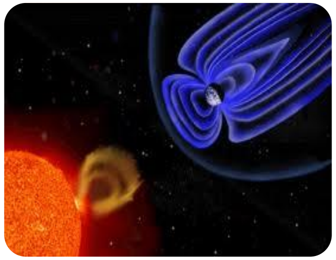


Application Areas of HPC in Science & Engineering

HPC in Science

Space Science

- Applications in Astrophysics and Astronomy



Earth Science

- Applications in understanding Physical Properties of Geological Structures, Water Resource Modelling, Seismic Exploration

Atmospheric Science

- Applications in Climate and Weather Forecasting, Air Quality

HPC in Science

Life Science

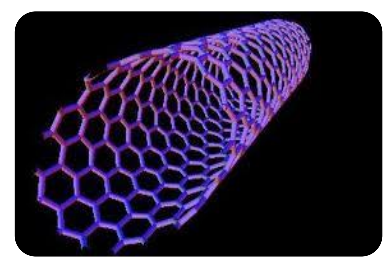
- Applications in Drug Designing, Genome Sequencing, Protein Folding

Nuclear Science

- Applications in Nuclear Power, Nuclear Medicine (cancer etc.), Defence

Nano Science

- Applications in Semiconductor Physics, Microfabrication, Molecular Biology, Exploration of New Materials



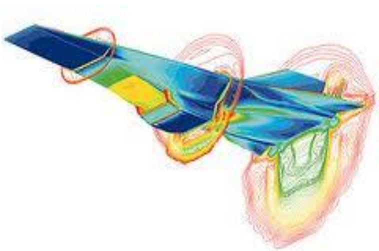
HPC in Engineering

Crash Simulation

- Applications in Automobile and Mechanical Engineering

Aerodynamics Simulation & Aircraft Designing

- Applications in Aeronautics and Mechanical Engineering



Structural Analysis

- Applications in Civil Engineering and Architecture

Multimedia and Animation

DreamWorks Animation SKG produces all its animated movies using HPC graphic technology

Graphical Animation Application in Multimedia and Animation

Thank You

ashishk@cdac.in