An Outlook of High Performance Computing Infrastructures for Scientific Computing

PART 2

Amjad Ali

Centre for Advanced Studies in Pure and Applied Mathematics
(CASPAM)
Bahauddin Zakariya University (BZU),
Multan 60800, Pakistan.

Advanced School on Scientific Software Development (SMR2330 Activity), 20 Feb. to 02 Mar. 2012, ICTP, Italy

OBJECTIVE

- GLIDE over the world of HPC
- Overview of well established & matured technologies

Presentation Outline

- Some Revision from Part 1
- Shared Memory Parallelism
- Distributed Memory Parallelism
- Hybrid Parallelism
- Cluster Computing
- On Parallel Performance
- Conclusion
- Quotation 1 & Quotation 2

A Chinese Proverb

"***Tell*** *me and I'll forget.*

Show *me, and I may remember.*

Involve *me and I'll understand*."

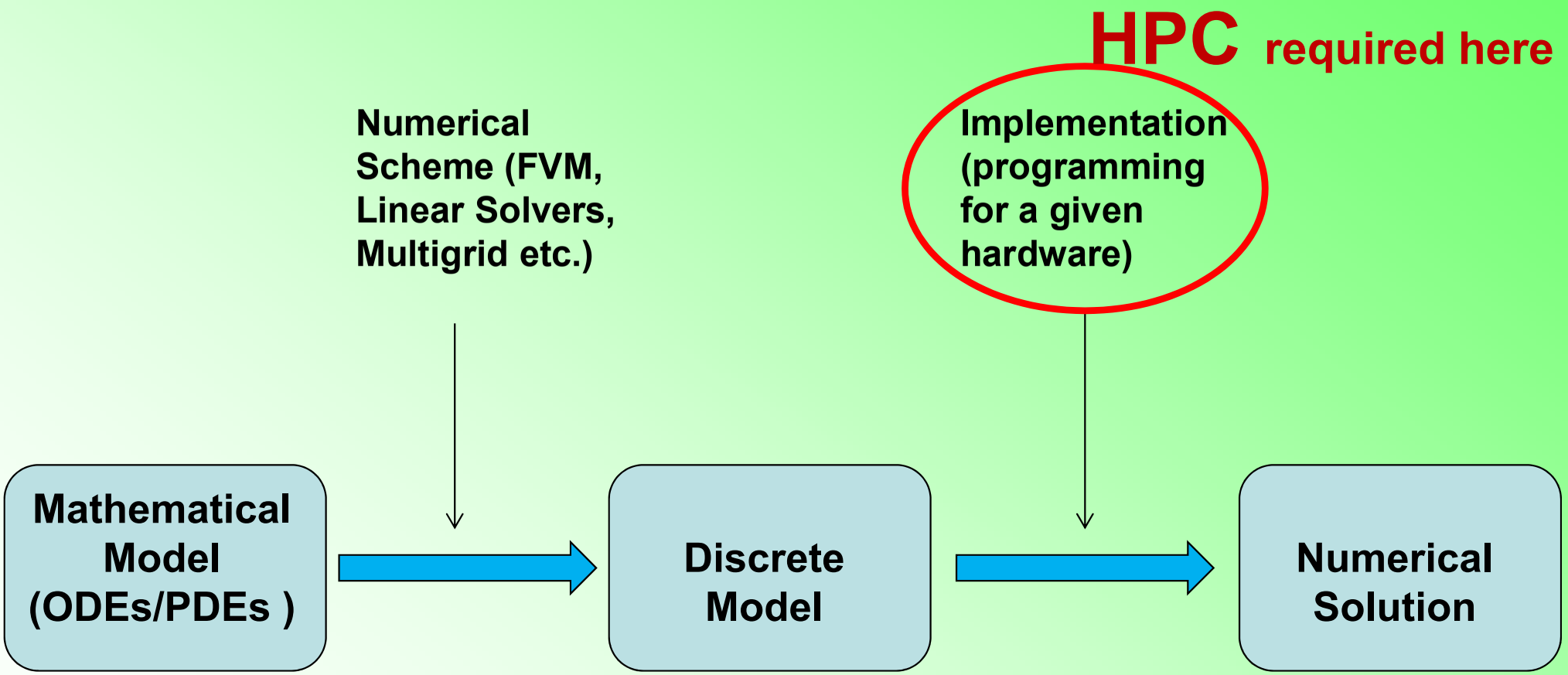
Confucius, died 478 BC

Spirit of Scientific Computing

Obtain some **qualitative predictions** through simulations

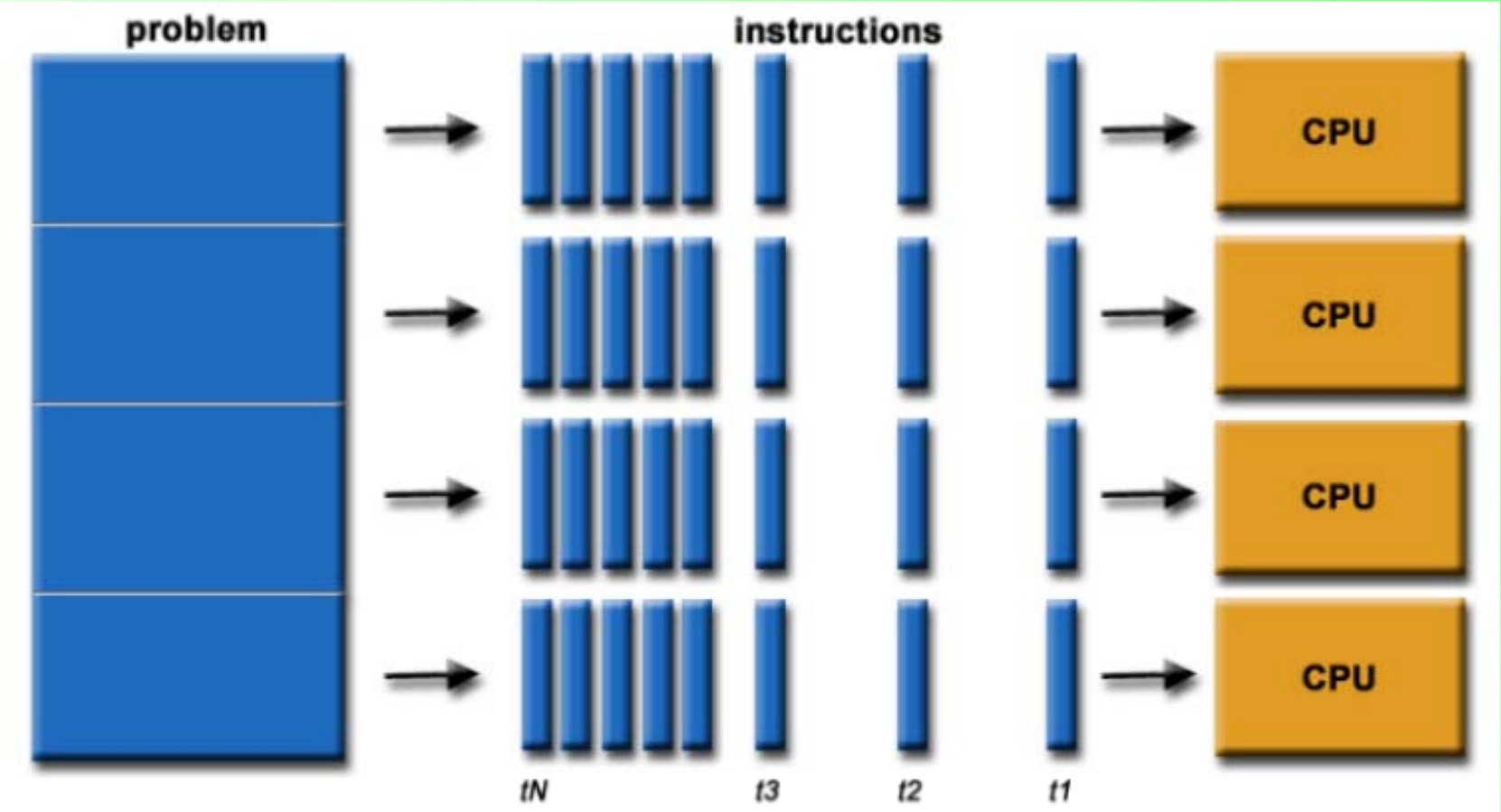
in "**virtual laboratories**" (i.e., the computers)

Scientific Computing needs HPC during implementation



HPC through Parallel Computing

Parallel computing refers to solve a computational problem by working on **different parts of the problem** simultaneously by **multiple processing units**.



Categorization of Parallelism

(from the programmer's perspective)

- **Implicit Parallelism**

  The programmer has not to do “much”. He has to let the parallelization feature of the compiler or a library to be active and the respective coding be favourable for that.

- **Explicit Parallelism**

  It is characterized by the fact that the programmer is responsible for

  (1) **taking care of subdividing the problem** into a number of sub-problems (with respect to data or tasks) for simultaneous execution on a number of processing elements, and

  (2) **managing the coordination** and synchronization among the sub-problems.

Common Forms of Parallelism

- Instruction Level Parallelism
  - (Superscalar, Pipelining, Superpipelining, Vectorization)
- Shared Memory Multithreading
  - (OpenMP, Cilk Plus, p-threads)
- Distributed Multiprocessing
  - (for example, MPI, PVM)
- GPU Parallelism
  - (for example, CUDA, OpenCL)
- Hybrid Parallelism
  - Distributed + Shared (for example, MPI + OpenMP)
  - Shared + GPU (for example, OpenMP + CUDA)
  - Distributed + GPU (for example, MPI + CUDA)

Note:

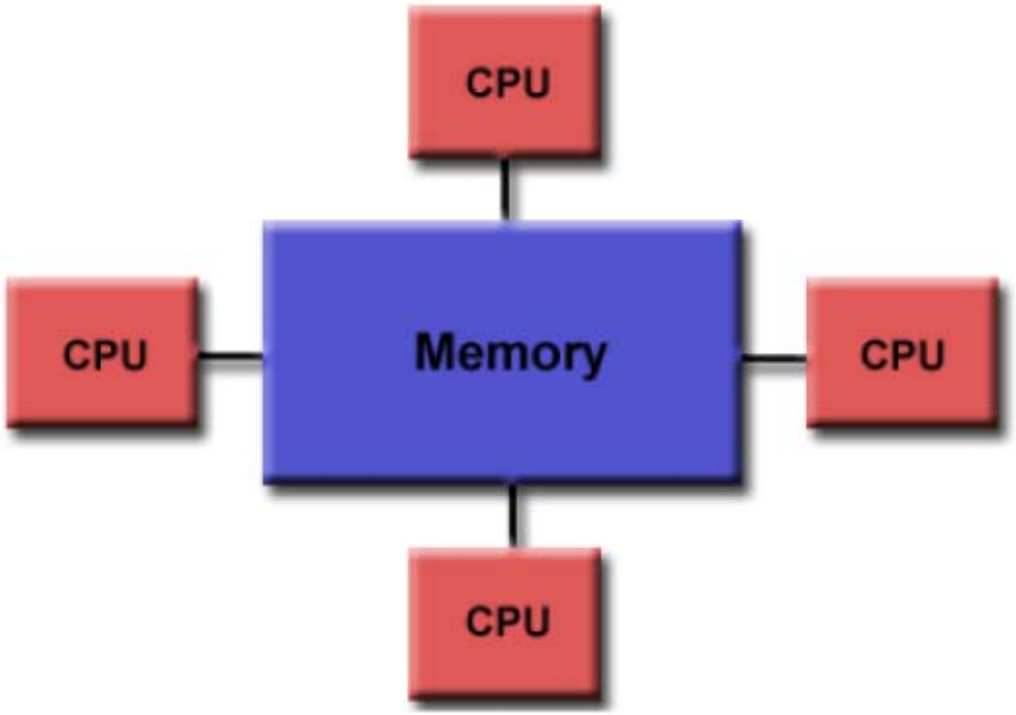
Respective capability both in hardware and software is necessary for any form of parallelism

Shared Memory Parallelism

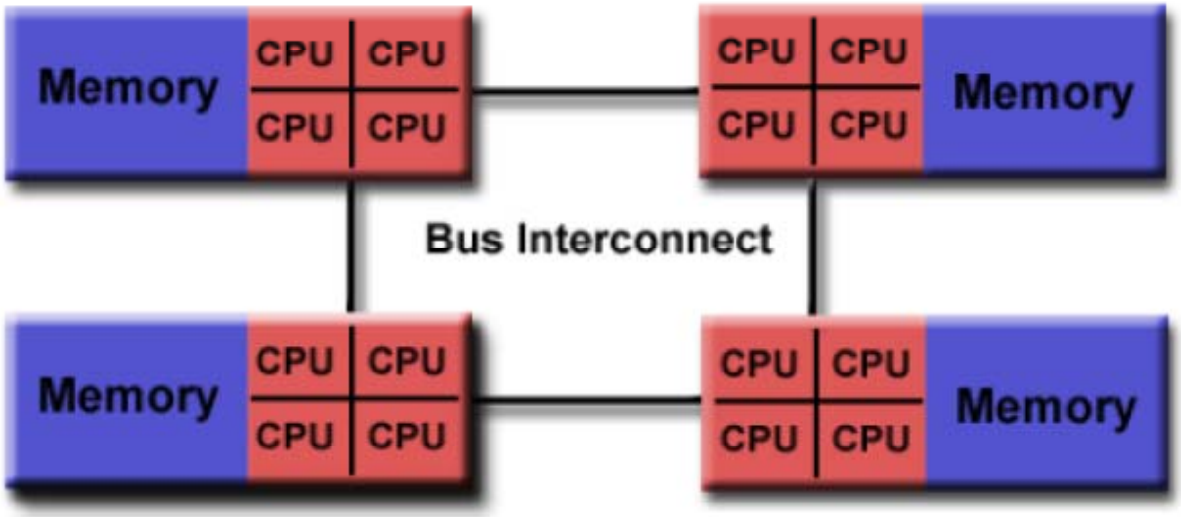
(from a hardware perspective)

- A shared memory parallel architecture is a computer that has a common physical memory, accessible to a number of physical processors.

- The two types of shared memory architectures are:
  - **Uniform Memory Access** (UMA)
  - **Non-Uniform Memory Access** (NUMA) [Figs from Barney-LLNL].

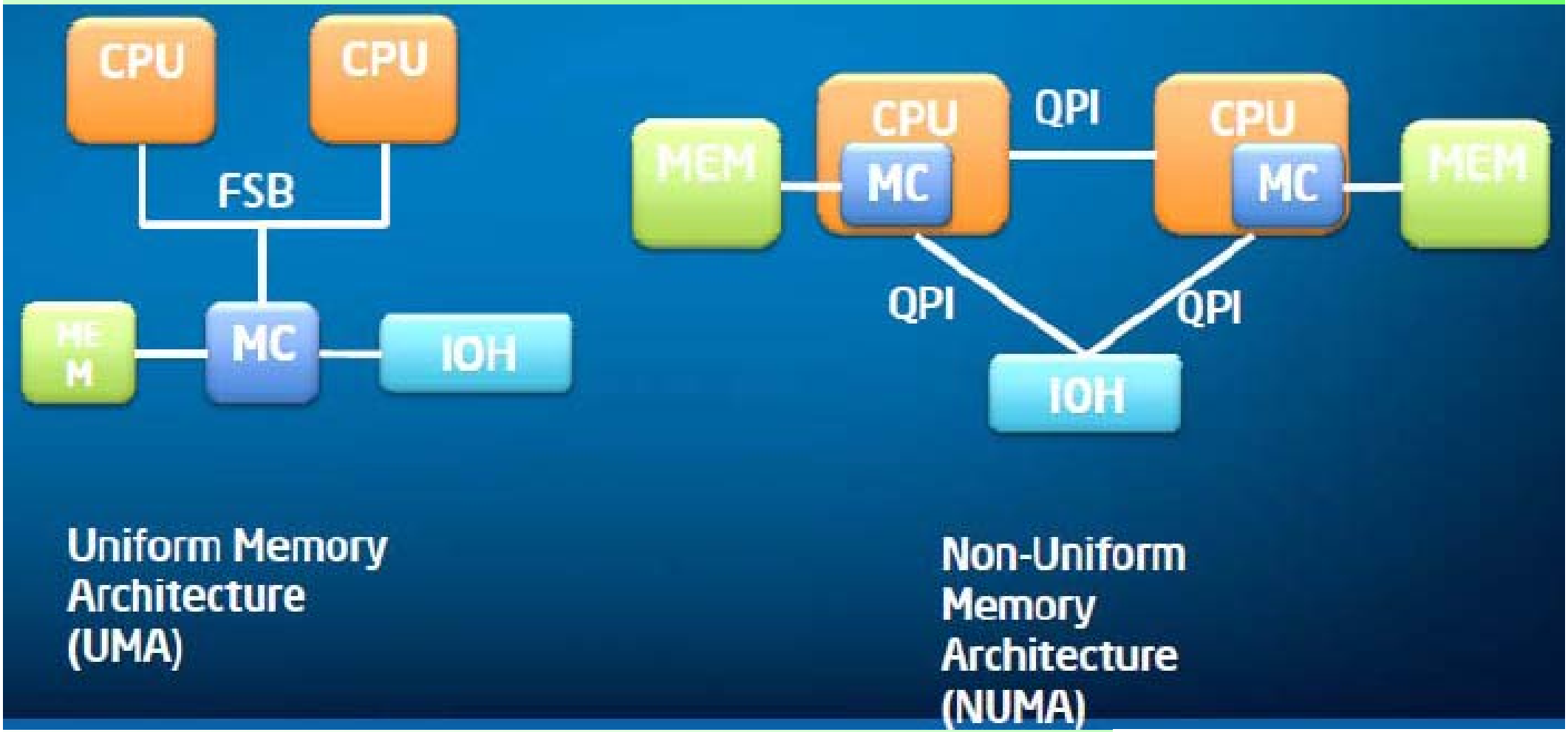


(a) Shared memory architecture UMA



(b) Shared memory architecture NUMA

Example of Xeon 5400 and 5500 based SMP machines



Multicore (or Many-core) CPU based systems are commodity systems

QPI= Quick Path Interconnect

IOH = I/O Hub

MC = Memory controller

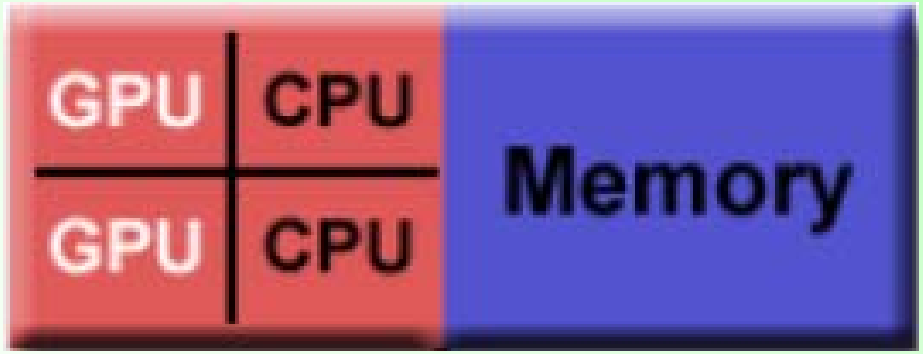
Shared Memory Parallelism

(from a programming perspective)

- The most common form of shared memory programming is the **multithreading**.
- The parallel application might involve multiple execution threads that share a **common** logical address space.
- Standard implementations of threads include **POSIX Threads**, **Intel Threading Building Blocks**, **Cilk Plus** and **OpenMP**.
- Note that, unlike OpenMP, POSIX threads are library based and parallelization with POSIX Threads is explicitly performed by the programmer.
- **Advantages**: simplicity and uniformity because of common global address space.
- **Disadvantages**: lesser scalability, memory contention, exponential growth in cost when core count is increased

General Purpose GPU (GPGPU) Computing

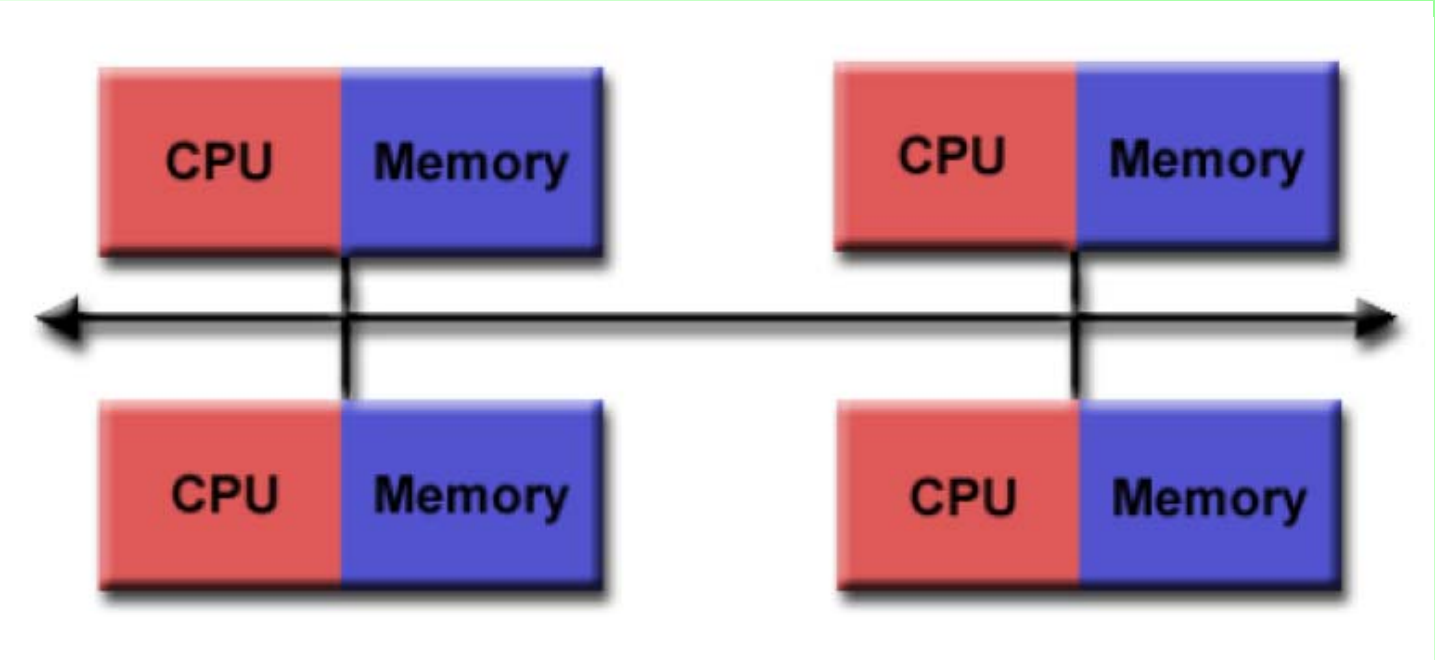
- Recently a more advanced and very fast, but under-developing and tricky, way of computing is realized that make use of graphical processing units (GPUs) for explicit parallel computations.
- With a GPU installed in a computer 10 time faster speed can be achieved, at least in theory, as of the year 2011.
- To make use of GPUs, currently CUDA (for GPUs manufactured by nVidia) and OpenCL are two famous programming models for GPGPU computing.



Distributed Memory Parallelism

(from a hardware perspective)

A distributed memory parallel architecture is a computer that has a number of physical processors each with its own local resources and separate memory space and requiring an interconnection network for mutual communication for accessing memory of other processors. [Figure from Barney-LLNL].



Distributed Memory Parallelism

(from a programming perspective)

- The most common form of distributed memory programming is **multiprocessing**.
- Multiple processes, each allocated with a sub-problem, are mapped to different processors (cores).
- The inter-process communication is performed using the message passing model.
- This is done in a cooperative fashion such that any message SEND call issued by a process must has a matching message RECEIVE call issued by the process that is supposed to receive the message.

- A message passing **implementation** is a library (of subroutines for a variety communication operations) that work in conjunction with the usual C/C++/Fortran compilers.
- **Message Passing Interface (MPI) library** is the most widely used implementation.
- MPI library includes a variety of routines for both **point-to-point** and **collective** communications.
- The communication event involving one sender and one receiver is referred to as point-to-point communication. This is in contrast to the collective communication which could be **one-to-all**, **all-to-one**, or **all-to-all**.
- MPI has emerged as a **de-facto** standard for **portable** and **scalable** parallel programming for distributed memory parallel architectures.

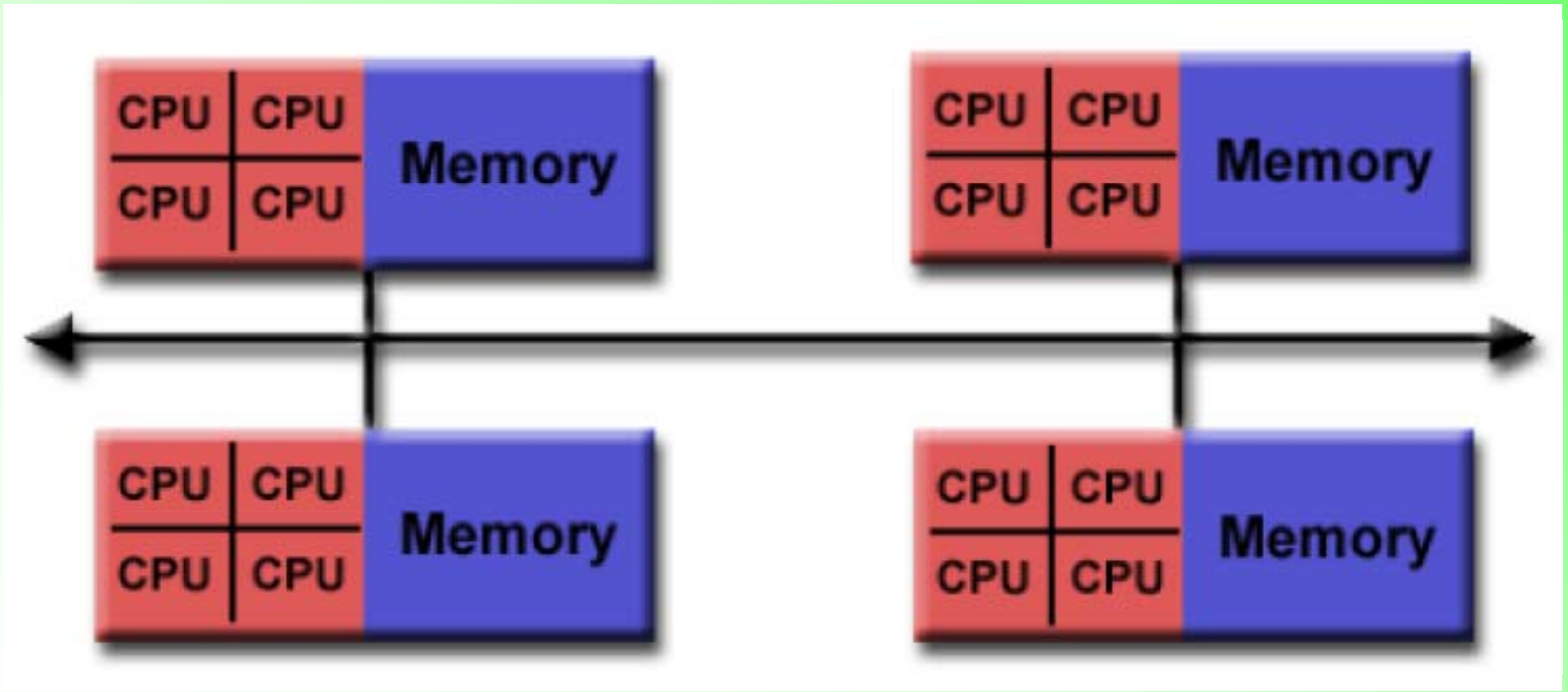
Well-known examples of **MPI implementations** include

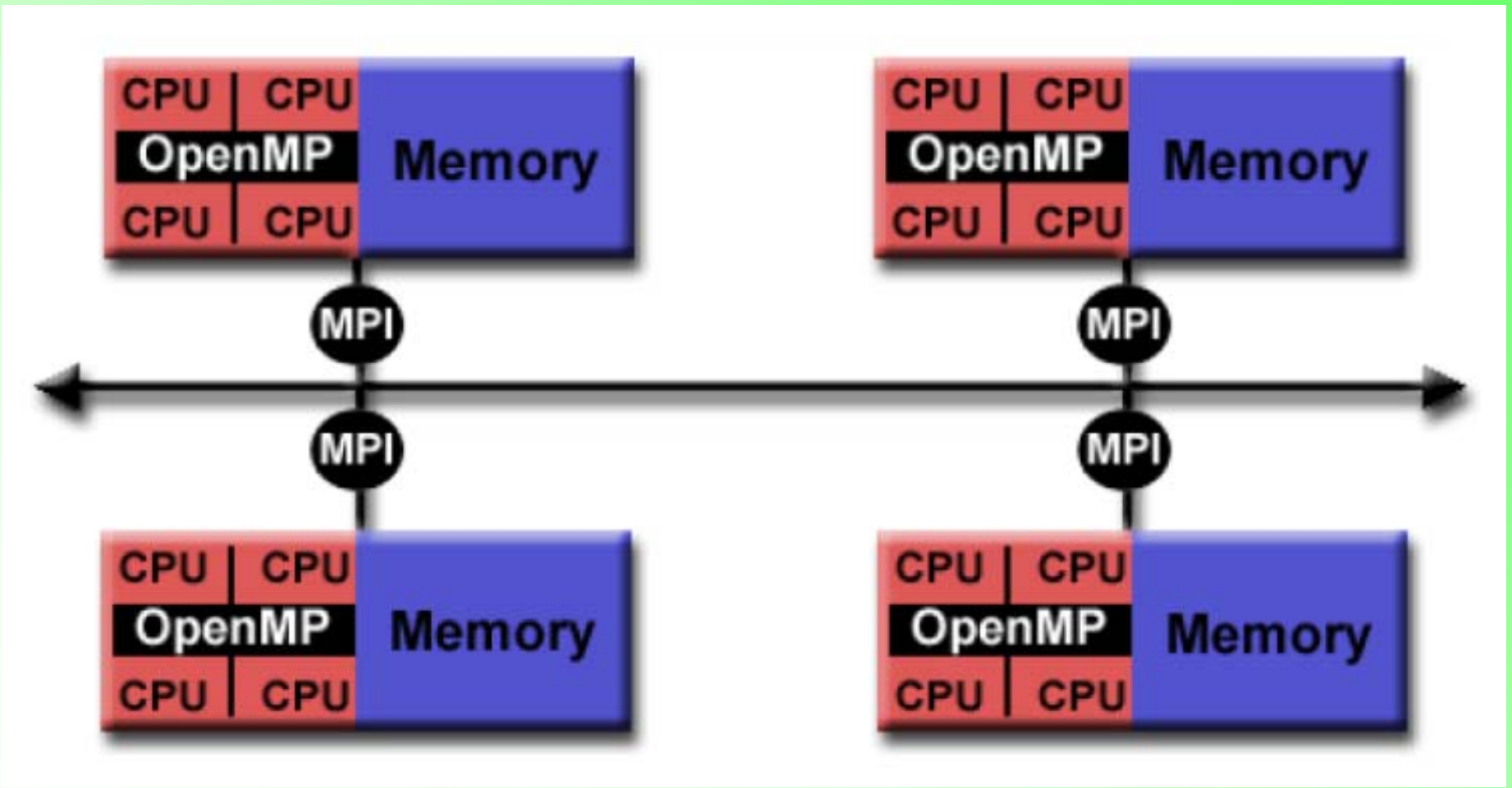
- **OpenMPI** (by Indiana University, open source)
- **MPICH** and **MPICH2** (by Argonne National Lab, open source)
- **MVAPICH** and **MVAPICH2** (by Ohio State University, free)
- **Platform MPI** (by Platform Computing, commercial)
- **Intel MPI** (by Intel, commercial)
- **MSMPI** (by Microsoft, commercial, for MS HPC Server 2008 OS)
- **MPJ Express** (by several including NUST-Pakistan, Java based)

- The advantages and disadvantages associated with distributed memory parallelism are in contrast to those of shared memory parallelism.
- Distributed memory systems are highly scalable and are less costly (per core).
- Even, the mass market commodity off-the-shelf (M²COTS) computer components can be used to build a cost effective system, usually called cluster.
- Programming is challenging. Intensive care in splitting of data structures across the parallel processes and in performing inter-process communication.
- In principle, the distributed memory MPI applications can execute on any shared memory architecture, as well. For example, 24 MPI processes can be executed in parallel on a 24-core machine (1 MPI process per processing core). Even for the case of shared memory space, each of the MPI process consider its memory space isolated from that of any other process.

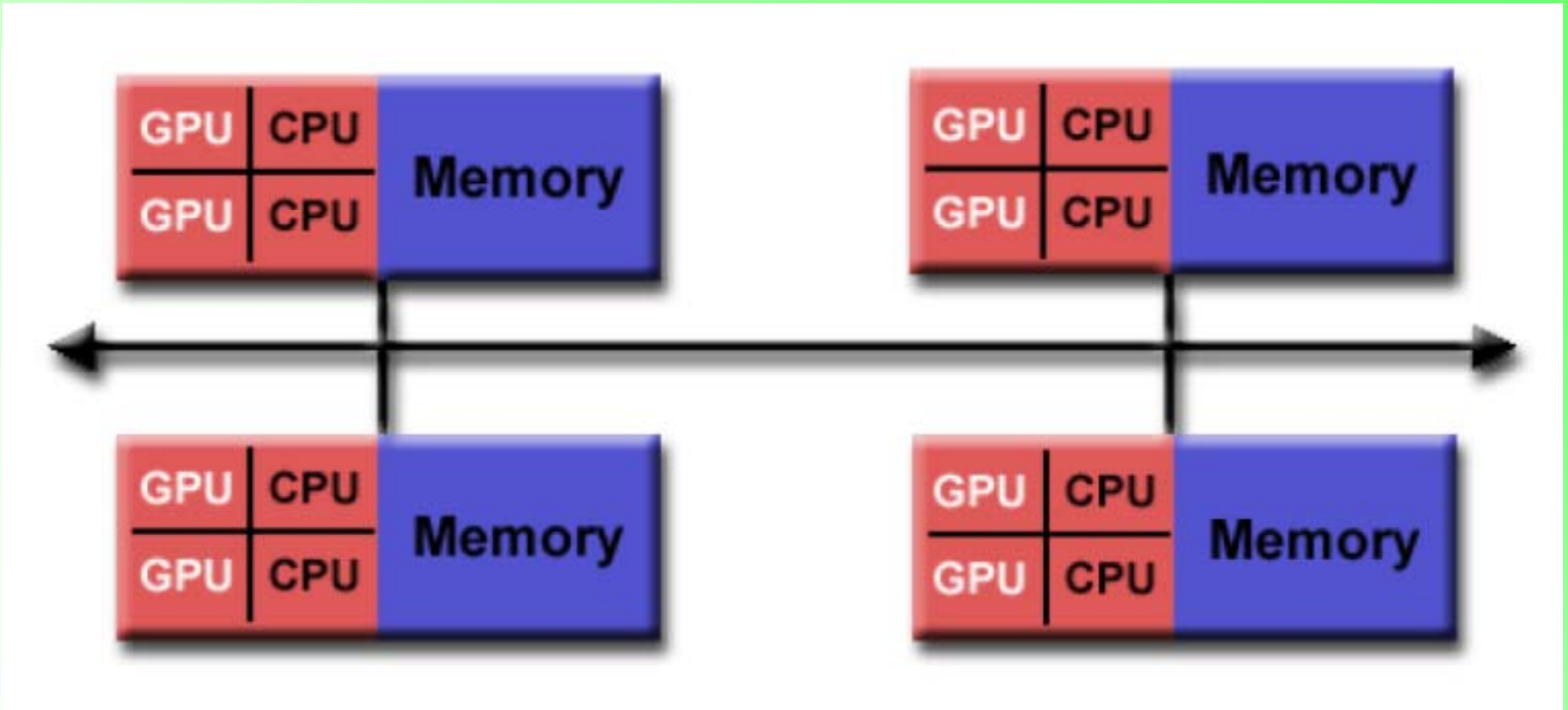
Hybrid Parallelism

(Shared + Distributed Memory System)




CPU
CPU
OpenMP
CPU
CPU
Memory
MPI
MPI
CPU
CPU
OpenMP
CPU
CPU
Memory
MPI
MPI
CPU
CPU
OpenMP
CPU
CPU
Memory
CPU
CPU
OpenMP
CPU
CPU
Memory

MPI + OpenMP



MPI + CUDA

or

MPI + OpenMP + CUDA

Cluster Computing for HPC

- A cluster is composed by interconnection of individual PCs using a interconnection network (simply called interconnect) such that the interconnected machines acts like a single computer.
- The cluster **interconnect** could be as simple as **Fast Ethernet** (of 100Mbps of bandwidth) and **Gigabit Ethernet** (of 1000Mbps of bandwidth). The cluster may also be equipped with a specialized interconnect technology (of high bandwidth and/or low latency), like 10-Gigabit Ethernet, Infiniband, Myrinet or Quadrics.

Basic Cluster Configuration

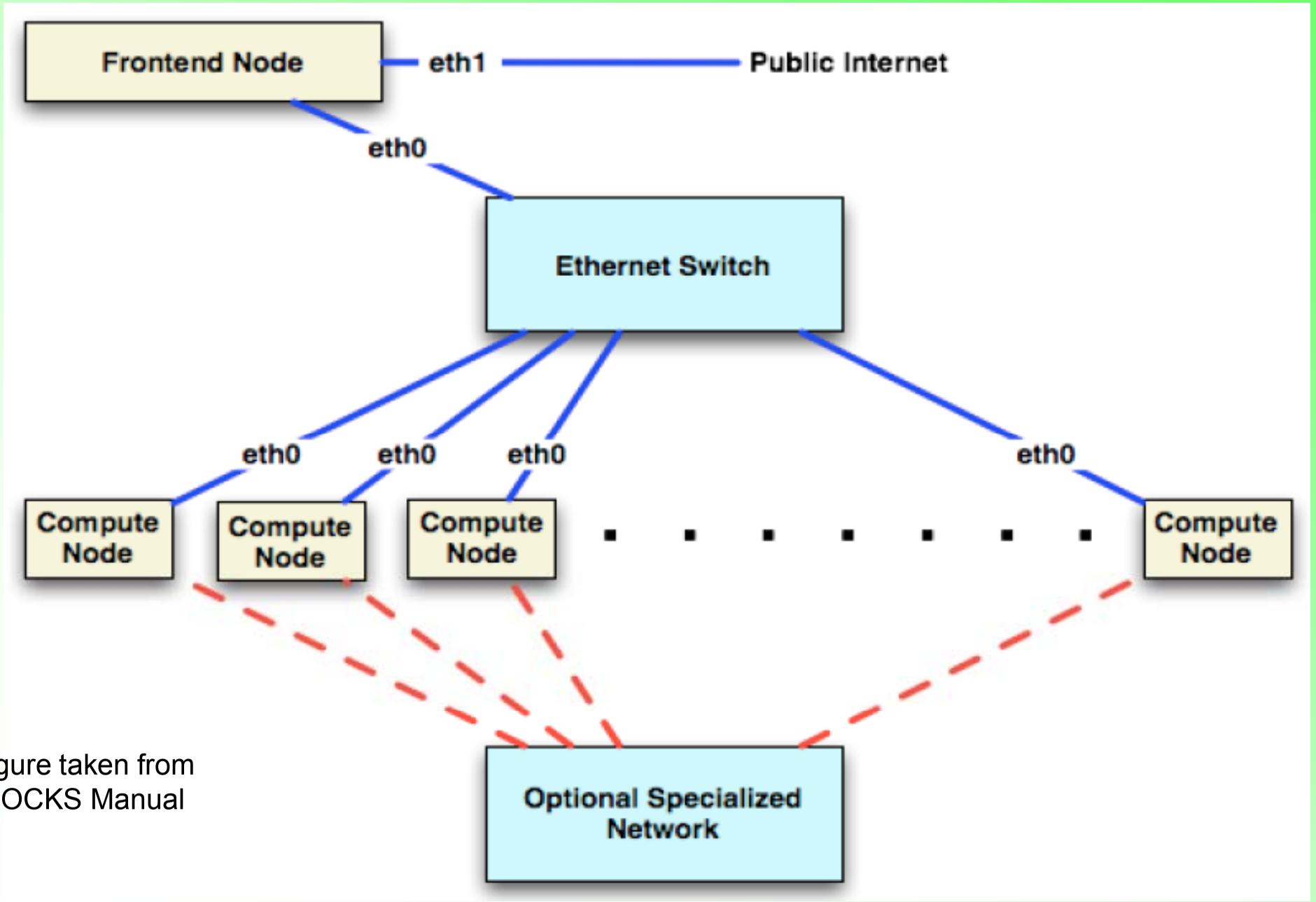
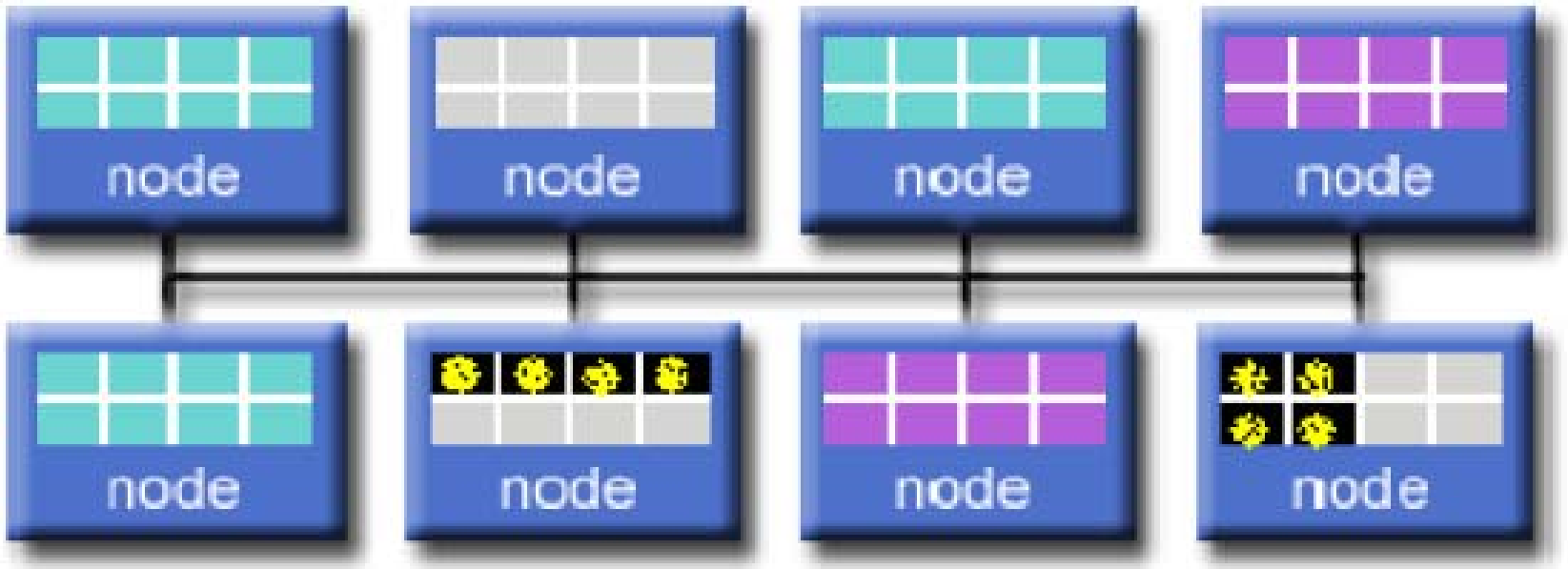


Figure taken from ROCKS Manual

- In a typical scenario, cluster users log-on to the front-end node of the cluster via public Internet and compile their programs and submit their batch jobs for execution.
- The front-end node manages some queues of user requests and schedules the submitted jobs to execute them on the compute node/s according to some pre-specified policy using a job scheduling software, also called **batch scheduler**.
- Examples of **batch schedulers** include,
  - Portable Batch System (PBS)
  - Oracle/Sun Grid Engine (SGE)
  - Platform LSF (Load Sharing Facility)
  - MAUI Cluster Scheduler
  - MOAB Cluster Suite
  - TORQUE Resource Manager
  - Simple Linux Utility for Resource Management (SLURM)

Example of Job Scheduling on a 8 Node Cluster



user1: 2-node, 16-way parallel job

user2: 3-node, 24-way parallel job

user3: 2-node, 8-way parallel job. (4-cores are kept perposefully free on each of the 2 nodes to avoid resource contensions).

- The clusters, as the hybrid memory architectures, offer the most cost effective solution to fulfill the need of high performance parallel computing capabilities for numerical simulations.
- Because of their effectiveness, cost-effectiveness and scalability, 410 (82%) supercomputing machines out of the world's top 500 known supercomputing machines as of November 2011 are clusters (See TOP500.org).
- However, it is worth to mention that clusters compliment rather than compete with the more sophisticated parallel computing architectures, usually called Massively Parallel Processing (MPP) machines, which are only 89 (17.8%) in number out of the total of top 500 machines, as of November 2011.

- As an evidence of this fact note:
  - **410 clusters is has 5804063 CPU cores**
  - **89 MPPs has 3378812 CPU corers**
  - **410 clusters has given total speed 50192818 GFLOPS**
  - **89 MPPs has given total speed 23823974 GFLOPS**
- This means that on the TOP500 list an MPP has **2.68 times** more number of CPU cores and **2.18 times** more speed than those of a cluster, on average.

- The dominant choices of the CPUs are the 64-bit architecture (x86-64) based Xeon (from Intel) and Opteron (from AMD) for the HPC machines. They are round 90% at top500).

- Single ATX box parallel computer like **Limulus** are getting introduced, giving rise to Personal Supercomputing.

- ROCKS and OSCAR are two very famous software for establishing a cluster “automatically”, including installation of Cluster utilization ad managing utilities.

Factors that Effect Performance of Parallel Programs

In general, performance of a parallel program in a distributed/hybrid memory with a given data size depends on many factors including:

- **Processor** (total number, number per node, speed)
- **Memory** (capacity, bandwidth, latency, caching effects)
- **Interconnect** (type, bandwidth, latency)
- **Environment** (OS, compilers, standard, library)
- **Implementation** (Algorithm, Data Structures and Memory Access Pattern
- **Granularity** (i.e., computation to communication ratio)

Considerations for Parallel Efficiency

- Proficient Domain/Task **Decomposition** (for balancing of workload and communication efficiency among the parallel processes). The well known software packages for domain decomposition include **METIS, Chaco, SCOTCH** and **JOSTLE**.

- Exploitation of **Locality of Reference**

- **Efficient Inter-Process Communication**
  1. Communication-Efficient Domain Partitioning
  2. Single Dispatch of Message to the Receiver
  3. Maximization of Local Computations
  4. Overlapping of the Communications with Computation

Conclusion

- Scientific computing demands HPC to obtain solution in reasonable time frames.
- Depending upon the available hardware and software, plan for the parallelization of the code.
- Hybrid parallel solution are becoming quite common.
- Take care of the factors that effect overall parallel performance; adopt strategies for efficiency.

Quotation 1

Life is very short;

Spend it for the purpose it has been given to you, that is,

develop a deep sense of love for the CREATOR ad HIS creature

Quotation 1

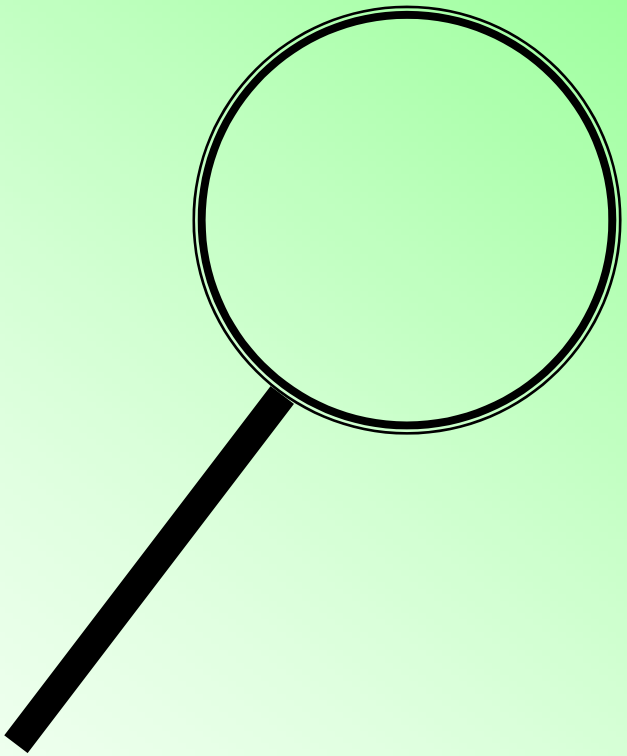
Life is very short;

Spend it for the purpose it has been given to you, that is,

develop a deep sense of love for the CREATOR ad HIS creature

Quotation 2

Don’t Forget Quotation 1



Questions

THANK YOU