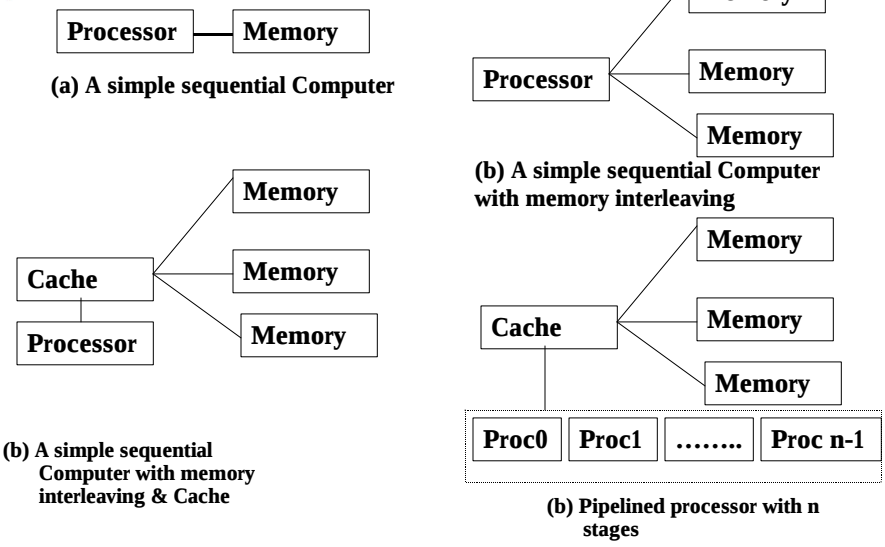
HPC Systems and Models

Dheeraj Bhardwaj
Department of Computer Science & Engineering
Indian Institute of Technology, Delhi –110 016 India
http://www.cse.iitd.ac.in/~dheerajb

Sequential Computers

- **Traditional Sequential computers are based on the model introduced by John-von-Neumann.**
- **Computational Model**
  - **SISD – Single Instruction Stream Single Data Stream**
- **The Speed of an SISD computer is limited by two factors**
  - **The execution rate of instructions**
    - **Overlapping the execution of instruction with the operation of fetching - Pipelining**
  - **Speed at which information is exchanged between memory and CPU**
    - **Memory interleaving**
    - **Cache Memory**

Evaluation of a typical sequential computer

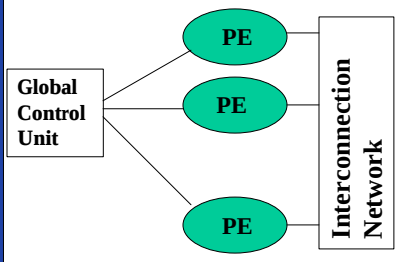


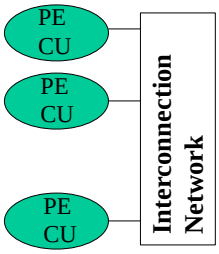
Serial Computer - Limitations

- **Memory interleaving, and to some extent, pipelining is useful only if a small set of operations is performed on large arrays of data**
- **Cache memories do increase processor-memory bandwidth but their speed is still limited by hardware technology.**

A Taxonomy of Parallel Architectures

- **Parallel computers differ along various dimensions**
  - **Control Mechanism**
  - **Address-space Organization**
  - **Interconnection Network**
  - **Granularity of processor**





SIMD: Single Instruction Stream Multiple Data

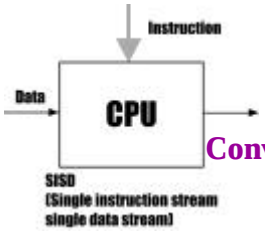
MIMD: Multiple Instruction Stream Multiple Data

SIMD

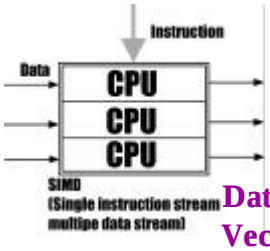
- **Single control unit dispatches instructions to each processing unit**
- **Same instruction is executed synchronously by all processing units**
- **Require less hardware (Single Control Unit)**
- **Naturally suited for data-parallel programs, i.e. programs in which the same set of instructions are executed on a large data set**
- **Very small latency**
- **Communication is just like register transfer**

Classification of Parallel Computers

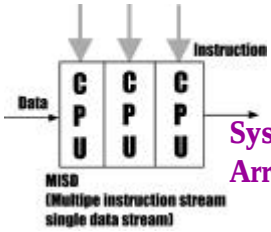
Flynn Classification: Number of Instructions & Data Streams



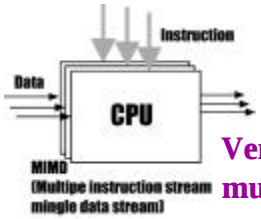
Conventional



Data Parallel, Vector Computing



Systolic Arrays



Very general, multiple approaches

MIMD

- Each processor is capable of executing a different program independent of the other processors
- More hardware
- Individual processors are more complex
- MIMD computer have extra hardware to provide faster synchronization

A drawback of SIMD

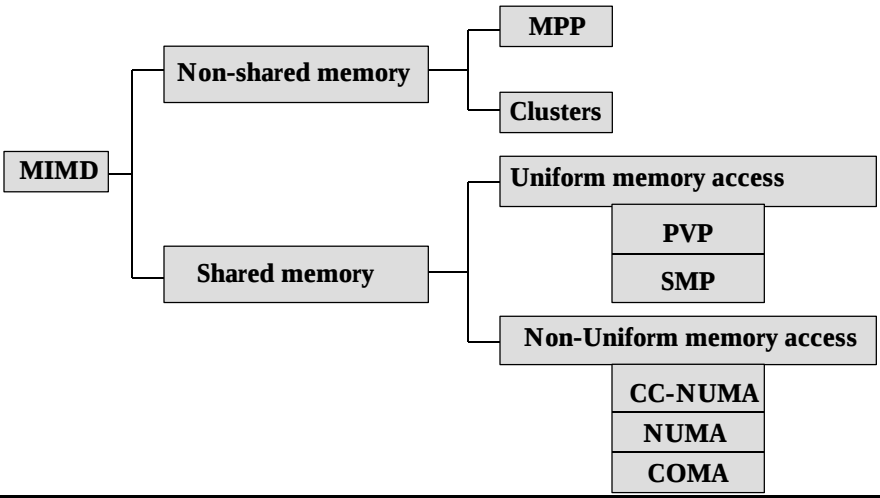
- **Different processors can not execute different instructions in the same clock cycle**
- **In a conditional statement, the code for each condition must be executed sequentially**

```
If (B == 0)
    C = A;
Else
    C = A/B;
```

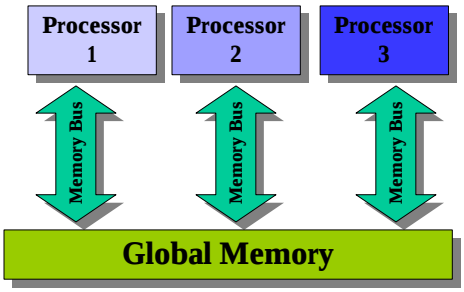
- **Conditional statement are better suited to MIMD computers than to SIMD computers**

MIMD Architecture: Classification

Current focus is on MIMD model, using general purpose processors or multicomputers.



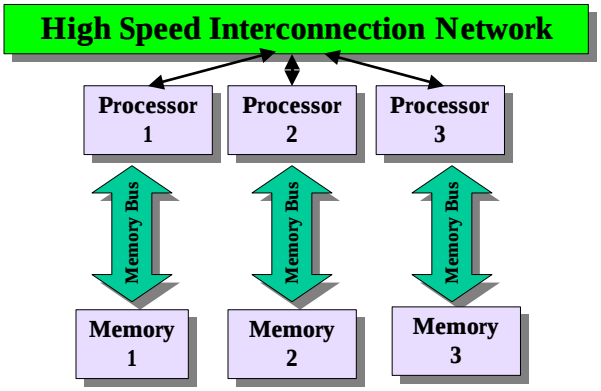
MIMD: Shared Memory Architecture



Source PE writes data to Global Memory & destination retrieves it

- **Easy to build**
- **Limitation : reliability & expandability. A memory component or any processor failure affects the whole system.**
- **Increase of processors leads to memory contention.**
  Ex. : Silicon graphics supercomputers....

MIMD: Distributed Memory Architecture



- **Inter Process Communication using High Speed Network.**
- **Network can be configured to various topologies e.g. Tree, Mesh, Cube..**
- **Unlike Shared MIMD**
  - **easily/ readily expandable**
  - **Highly reliable (any CPU failure does not affect the whole system)**

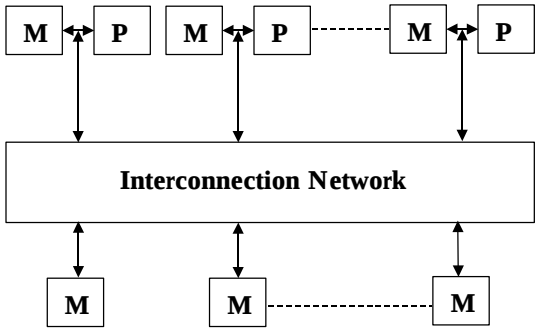
MIMD Features

- MIMD architecture is more general purpose
- MIMD needs clever use of synchronization that comes from message passing to prevent the race condition
- Designing efficient message passing algorithm is hard because the data must be distributed in a way that minimizes communication traffic
- Cost of message passing is very high

Shared Memory (Address-Space) Architecture

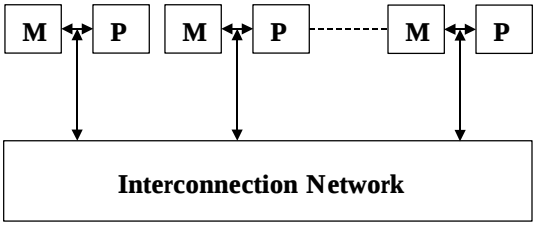
- Non-Uniform memory access (NUMA) shared address space computer with local and global memories
  - Time to access a remote memory bank is longer than the time to access a local word
- Shared address space computers have a local cache at each processor to increase their effective processor-bandwidth.
- The cache can also be used to provide fast access to remotely –located shared data
- Mechanisms developed for handling cache coherence problem

Shared Memory (Address-Space) Architecture



Non-uniform memory access (NUMA) shared-address-space computer with local and global memories

Shared Memory (Address-Space) Architecture

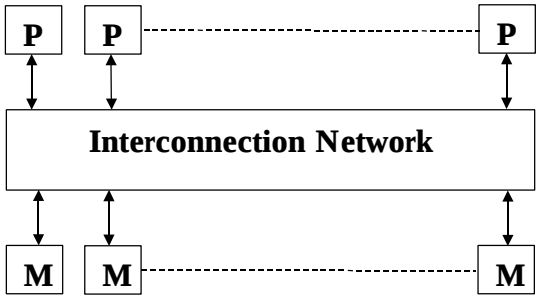


Non-uniform-memory-access (NUMA) shared-address-space computer with local memory only

Shared Memory (Address-Space) Architecture

- **Provides hardware support for read and write access by all processors to a shared address space.**
- **Processors interact by modifying data objects stored in a shared address space.**
- **MIMD shared -address space computers referred as multiprocessors**
- **Uniform memory access (UMA) shared address space computer with local and global memories**
  - **Time taken by processor to access any memory word in the system is identical**

Shared Memory (Address-Space) Architecture



Uniform Memory Access (UMA) shared-address-space computer

Definition

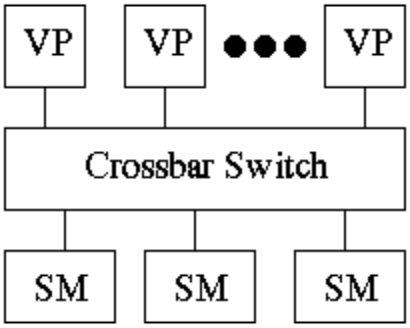
- **Cache – to increase processor-memory bandwidth**
- **Cache Coherence – This problem occurs when a processor modifies a shared variable in its cache. After this modification, different processors have different values of the variable in the other cache are simultaneously invalidated or updated**
- **COMA – Cache only memory access**

Uniform Memory Access (UMA)

UMA – Time taken by a processor to access to any memory word in system is identical

- **Parallel Vector Processors (PVPs)**
- **Symmetric Multiple Processors (SMPs)**

Parallel Vector Processor



VP : Vector Processor

SM : Shared memory

Parallel Vector Processor

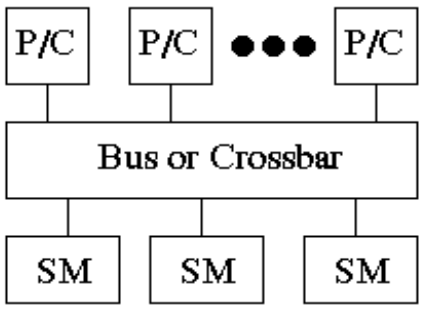
- **Works good only for vector codes**
- **Scalar codes may not perform perform well**
- **Need to completely rethink and re-express algorithms so that vector instructions were performed almost exclusively**
- **Special purpose hardware is necessary**
- **Fastest systems are no longer vector uniprocessors.**

Parallel Vector Processor

- **Small number of powerful custom-designed vector processors used**
- **Each processor is capable of at least 1 Giga flop/s performance**
- **A custom-designed, high bandwidth crossbar switch networks these vector processors.**
- **Most machines do not use caches, rather they use a large number of vector registers and an instruction buffer**

Examples : Cray C-90, Cray T-90, Cray T-3D …

Symmetric Multiprocessors (SMPs)



P/C : Microprocessor and cache

SM : Shared memory

Symmetric Multiprocessors (SMPs) characteristics

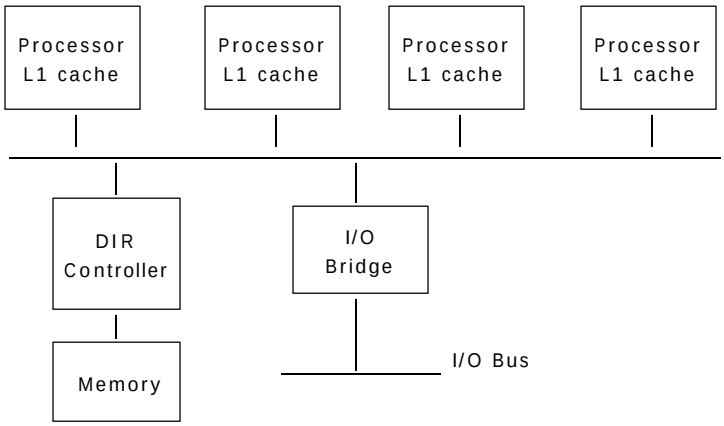
- **Uses commodity microprocessors with on-chip and off-chip caches.**
- **Processors are connected to a shared memory through a high-speed snoopy bus**
- **On Some SMPs, a crossbar switch is used in addition to the bus.**
- **Scalable upto:**
  - **4-8 processors (non-back planed based)**
  - **few tens of processors (back plane based)**

Symmetric Multiprocessors (SMPs)

Symmetric Multiprocessors (SMPs) characteristics

- **All processors see same image of all system resources**
- **Equal priority for all processors (except for master or boot CPU)**
- **Memory coherency maintained by HW**
- **Multiple I/O Buses for greater Input / Output**

Symmetric Multiprocessors (SMPs)



Symmetric Multiprocessors (SMPs)

- **Issues**
- **Bus based architecture :**
  - **Inadequate beyond 8-16 processors**
- **Crossbar based architecture**
  - **multistage approach considering I/Os required in hardware**
- **Clock distribution and HF design issues for backplanes**
- **Limitation is mainly caused by using a centralized shared memory and a bus or cross bar interconnect which are both difficult to scale once built.**

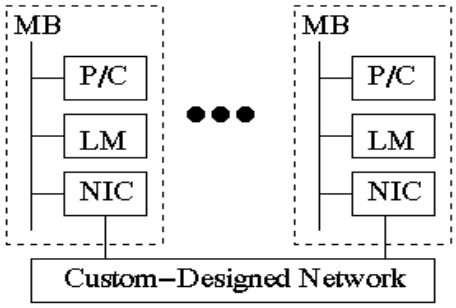
Commercial Symmetric Multiprocessors (SMPs)

- Sun Ultra Enterprise 10000 (high end, expandable upto 64 processors), Sun Fire
- DEC Alpha server 8400
- HP 9000
- SGI Origin
- IBM RS 6000
- IBM P690, P630
- Intel Xeon, Itanium, IA-64(McKinley)

Symmetric Multiprocessors (SMPs)

- Heavily used in commercial applications (data bases, on-line transaction systems)
- System is symmetric (every processor has equal equal access to the shared memory, the I/O devices, and the operating systems.
- Being symmetric, a higher degree of parallelism can be achieved.

Massively Parallel Processors (MPPs)



P/C : Microprocessor and cache; LM : Local memory; NIC : Network interface circuitry; MB : Memory bus

Massively Parallel Processors (MPPs)

- **Commodity microprocessors in processing nodes**
- **Physically distributed memory over processing nodes**
- **High communication bandwidth and low latency as an interconnect. (High-speed, proprietary communication network)**
- **Tightly coupled network interface which is connected to the memory bus of a processing node**

Massively Parallel Processors (MPPs)

- **Provide proprietary communication software to realize the high performance**
- **Processors Interconnected by a high-speed memory bus to a local memory through and a network interface circuitry (NIC)**
- **Scaled up to hundred or even thousands of processors**
- **Each processes has its private address space and Processes interact by passing messages**

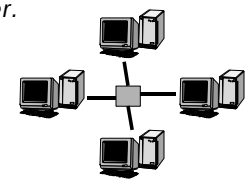
Massively Parallel Processors (MPPs)

- **MPPs support asynchronous MIMD modes**
- **MPPs support single system image at different levels**
- **Microkernel operating system on compute nodes**
- **Provide high-speed I/O system**
- **Example : Cray – T3D, T3E, Intel Paragon, IBM SP2**

Cluster ?

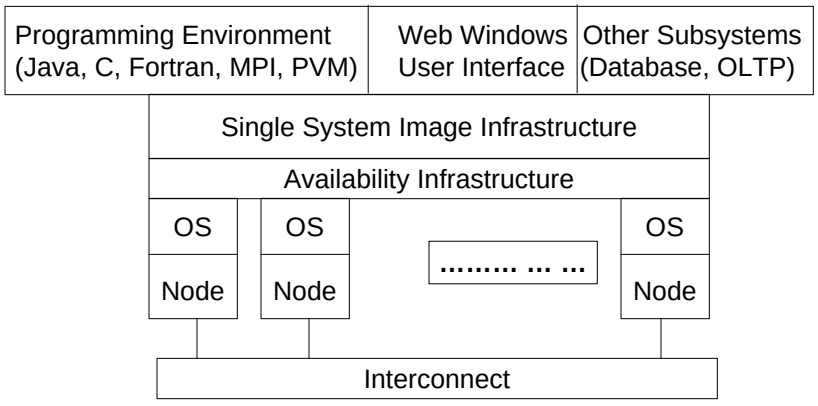
Clus·ter *n.*

1. A group of the same or similar elements gathered or occurring closely together; a bunch: *"She held out her hand, a small tight cluster of fingers"* (Anne Tyler).
2. *Linguistics.* Two or more successive consonants in a word, as *cl* and *st* in the word *cluster.*



A Cluster is a type of parallel or distributed processing system, which consists of a collection of interconnected stand alone/complete computers cooperatively working together as a single, integrated computing resource.

Cluster System Architecture



Clusters ?

- A set of
- Nodes physically connected over commodity / proprietary network
- Gluing Software
  - Other than this definition no Official Standard exists
- Depends on the user requirements
  - Commercial
  - Academic
  - Good way to sell old wine in a new bottle
  - Budget
  - Etc ..
- Designing Clusters is not obvious but Critical issue.

Why Clusters NOW?

- Clusters gained momentum when three technologies converged:
  - Very high performance microprocessors
    - workstation performance = yesterday supercomputers
  - High speed communication
  - Standard tools for parallel / distributed computing & their growing popularity
- Time to market => performance
- Internet services: huge demands for scalable, available, dedicated internet servers
  - big I/O, big computing power

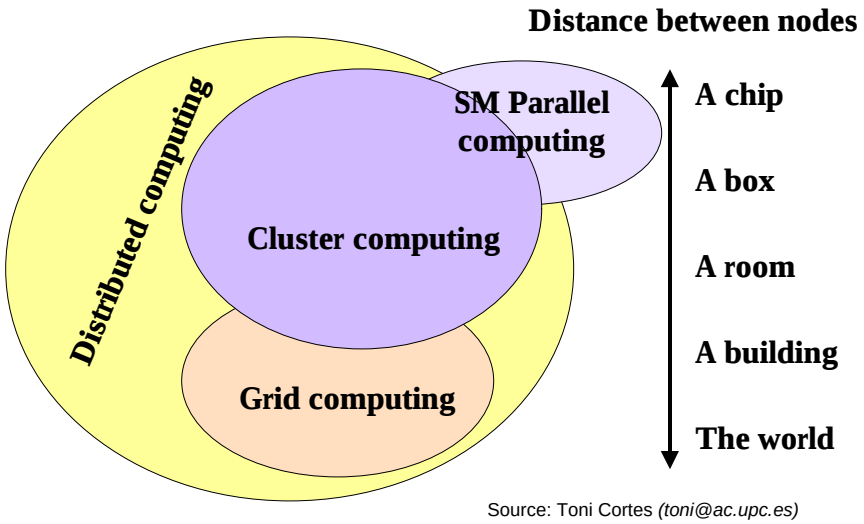
How should we Design them ?

- **Components**
  - **Should they be off-the-shelf and low cost?**
  - **Should they be specially built?**
  - **Is a mixture a possibility?**
- **Structure**
  - **Should each node be in a different box (workstation)?**
  - **Should everything be in a box?**
  - **Should everything be in a chip?**
- **Kind of nodes**
  - **Should it be homogeneous?**
  - **Can it be heterogeneous?**

What Should it offer ?

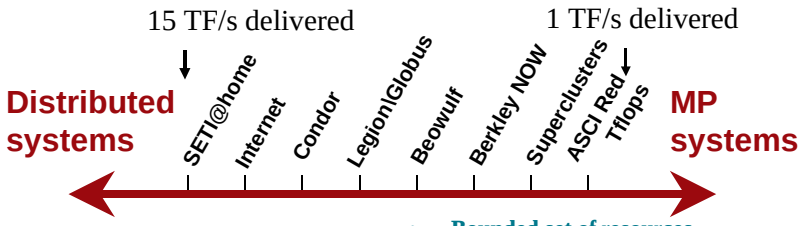
- **Identity**
  - **Should each node maintains its identity (and owner)?**
  - **Should it be a pool of nodes?**
- **Availability**
  - **How far should it go?**
- **Single-system Image**
  - **How far should it go?**

Place for Clusters in HPC world ?



Source: Toni Cortes *(toni@ac.upc.es)*

Where Do Clusters Fit?



- Gather (unused) resources
- System SW manages resources
- System SW adds value
- 10% - 20% overhead is OK
- Resources drive applications
- Time to completion is not critical
- Time-shared
- Commercial: PopularPower, United Devices, Centrata, ProcessTree, Applied Meta, etc.

- Bounded set of resources
- Apps grow to consume all cycles
- Application manages resources
- System SW gets in the way
- 5% overhead is maximum
- Apps drive purchase of equipment
- Real-time constraints
- Space-shared

Src: B. Maccabe, UNM, R.Pennington NCSA

Top 500 Supercomputers

Rank	Computer/Procs	Peak performance	Country/year
1	Earth Simulator (NEC) 5120	40960 GF	Japan / 2002
2	ASCI – Q (HP) AlphaServer SC ES45/1.25 GHz/ 4096	10240 GF	LANL, USA/2002
3	ASCI – Q (HP) AlphaServer SC ES45/1.25 GHz/ 4096	10240 GF	LANL, USA/2002
4	ASCI White (IBM) SP power 3 375 MHz / 8192	12288 GF	LANL, USA/2000
5	MCR Linux Cluster Xeon 2.4 GHz – Qudratics / 2304	11060GF	LANL, USA/2002

- From www.top500.org

What makes the Clusters ?

- The same hardware used for
  - Distributed computing
  - Cluster computing
  - Grid computing
- Software converts hardware in a cluster
  - Tights everything together

Task Distribution

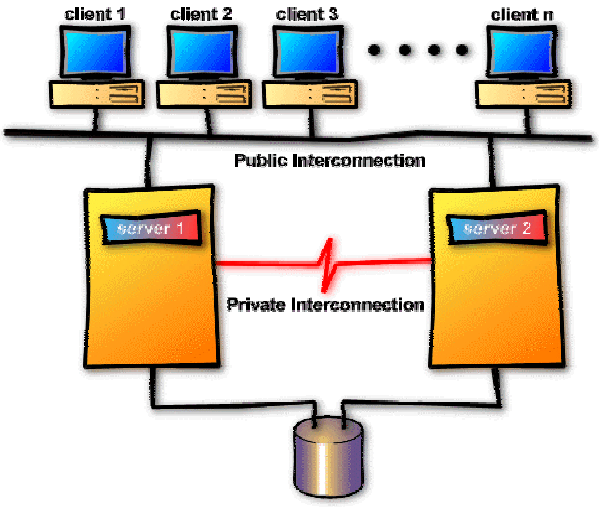
- **The hardware is responsible for**
  - **High-performance**
  - **High-availability**
  - **Scalability (network)**
- **The software is responsible for**
  - **Gluing the hardware**
  - **Single-system image**
  - **Scalability**
  - **High-availability**
  - **High-performance**

Classification of Cluster Computers

Clusters Classification 1

- **Based on Focus (in Market)**
  - **High performance (HP) clusters**
    - **Grand challenging applications**
  - **High availability (HA) clusters**
    - **Mission critical applications**
    - **Web/e-mail**
    - **Search engines**

HA Clusters



Clusters Classification 2

- **Based on Workstation/PC Ownership**
  - **Dedicated clusters**
  - **Non-dedicated clusters**
    - **Adaptive parallel computing**
    - **Can be used for CPU cycle stealing**

Clusters Classification 3

- **Based on Node Architecture**
  - **Clusters of PCs (CoPs)**
  - **Clusters of Workstations (COWs)**
  - **Clusters of SMPs (CLUMPs)**

Clusters Classification 4

- **Based on Node Components Architecture & Configuration:**
  - **Homogeneous clusters**
    - **All nodes have similar configuration**
  - **Heterogeneous clusters**
    - **Nodes based on different processors and running different OS**

Clusters Classification 5

- **Based on Node OS Type..**
  - **Linux Clusters (Beowulf)**
  - **Solaris Clusters (Berkeley NOW)**
  - **NT Clusters (HPVM)**
  - **AIX Clusters (IBM SP2)**
  - **SCO/Compaq Clusters (Unixware)**
  - **……Digital VMS Clusters, HP clusters, ……..**

Clusters Classification 6

- **Based on Levels of Clustering:**
  - **Group clusters (# nodes: 2-99)**
    - **A set of dedicated/non-dedicated computers --- mainly connected by SAN like Myrinet**
  - **Departmental clusters (# nodes: 99-999)**
  - **Organizational clusters (# nodes: many 100s)**
  - **Internet-wide clusters = Global clusters (# nodes: 1000s to many millions)**
    - **Computational Grid**

Clustering Evolution