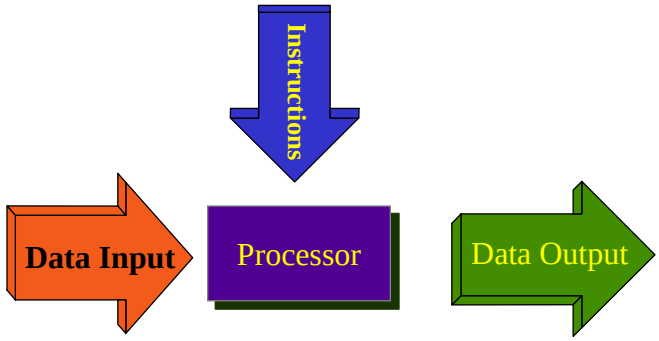
Processing Elements Architecture

Processing Elements

- Simple classification by Flynn (**FLYNN'S TAXONOMY**): (dual concept of Instruction stream and Data stream.)
  - **SISD - conventional**
  - **SIMD - data parallel, vector computing**
  - **MISD - systolic arrays**
  - **MIMD - very general, multiple approaches.**
- Current development is on MIMD model, using general purpose processors.

SISD : A Conventional Computer

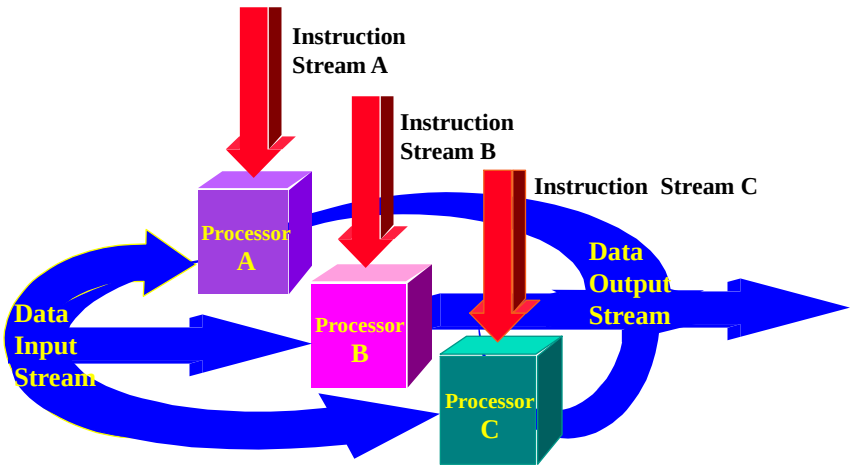


➔Speed is limited by the rate at which computer can transfer information internally.

Ex:PC, Macintosh, Workstations

Source: Raj Kumar Buyya ppts.

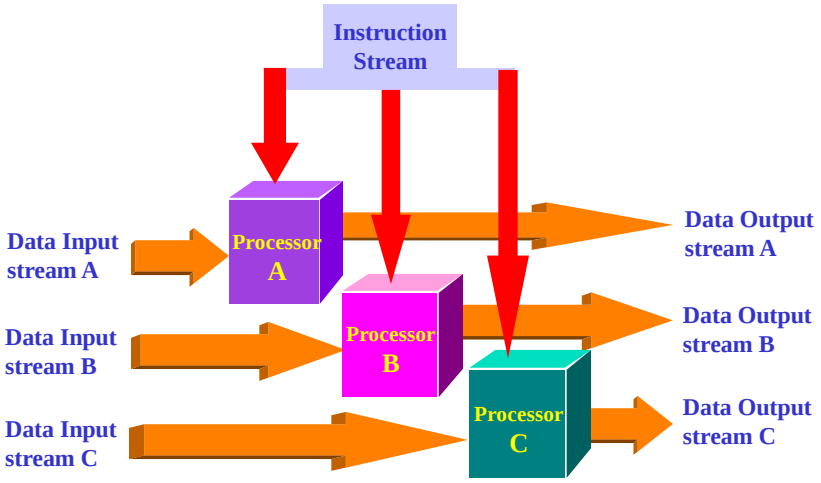
The MISD Architecture



➔ Only an intellectual exercise. Few built, but commercially not available

Source: Raj Kumar Buyya ppts.

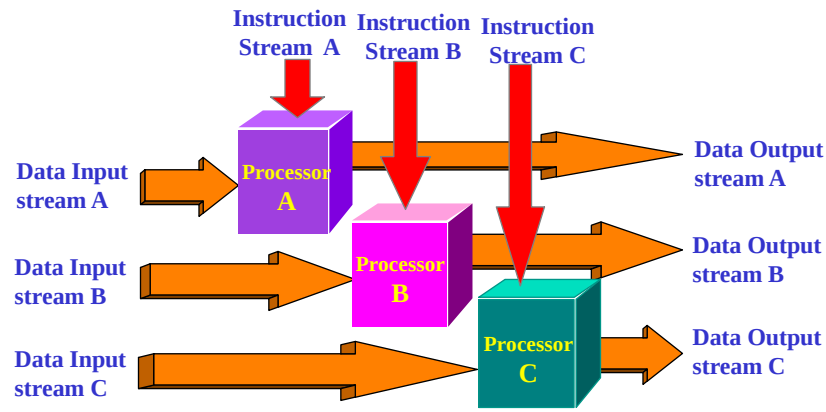
SIMD Architecture



Ex: CRAY machine vector processing, Thinking machine cm*

Source: Raj Kumar Buyya ppts.

MIMD Architecture

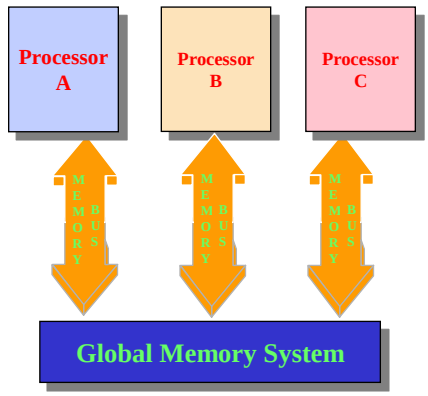


Unlike SISD and MISD a MIMD computer works asynchronously.

- Shared memory (tightly coupled) MIMD
- Distributed memory (loosely coupled) MIMD

Source: Raj Kumar Buyya ppts.

Shared Memory MIMD machine



Source: Raj Kumar Buyya ppts.

Shared Memory MIMD machine

Comm: Source PE writes data to GM & destination retrieves it

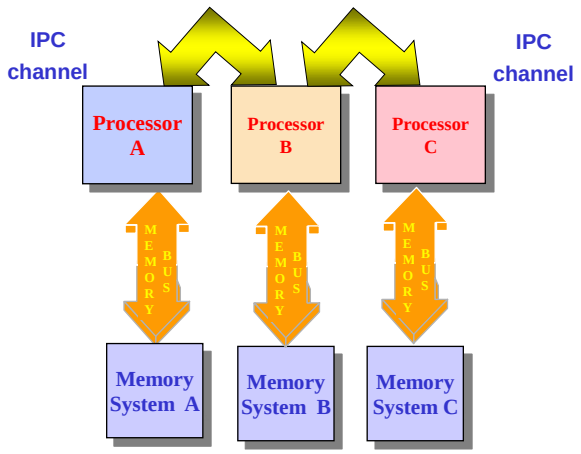
➔ Easy to build, conventional OSes of SISD can easily be ported

➔ 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....

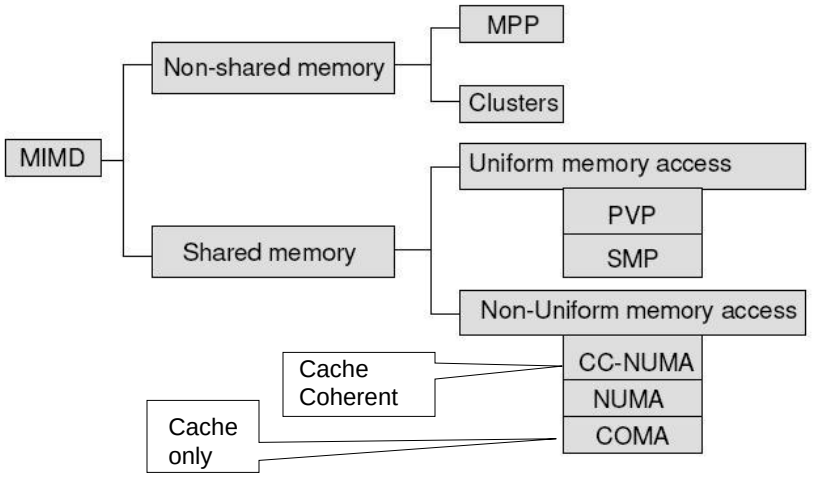
Distributed Memory MIMD



Distributed Memory MIMD

- Communication : IPC on High Speed Network.
- Network can be configured to ... Tree, Mesh, Cube, etc.
- Unlike Shared MIMD
  - ➔**easily/ readily expandable**
  - ➔**Highly reliable (any CPU failure does not affect the whole system)**

Architectural Model of MIMD



Source: Berry Wilkinson's ppts.

Shared memory multiprocessor system

Any memory location can be accessible by any of the processors.

A *single address space* exists, meaning that each memory location is given a unique address within a single range of addresses.

Generally, shared memory programming is more convenient although it does require access to shared data to be controlled by the programmer (using critical sections etc.)

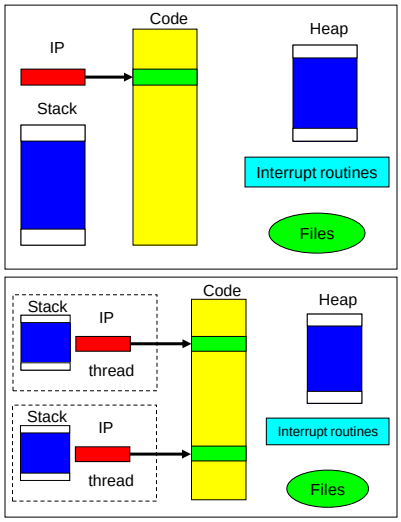
Several Alternatives for Programming Shared Memory Multiprocessors:

- Using heavy weight processes.
- Using threads. Example Pthreads.
- Using a completely new programming language for parallel programming Example Ada or newly designed Cilk++.
- Using library routines with an existing sequential programming language.
- Modifying the syntax of an existing sequential programming language to create a || programming language. Example UPC.
- Using an existing sequential programming language supplemented with compiler directives for specifying parallelism. Example **OpenMP.**
- Using **Threading Building Block (TBB)** from Intel.

Differences between a process and threads

process - completely separate program with its own variables, stack, and memory allocation.

Threads - shares the same memory space and global variables between routines.



"lightweight" process is a kernel or some O.S. thread

Shared Memory Systems and Programming

Topics:

- Regular shared memory systems and programming
- Distributed shared memory on a cluster

Pthreads

IEEE Portable Operating System Interface, POSIX, sec. 1003.1 standard

In Pthread, the main program is a thread itself, a separate thread is created and terminated with the routine itself.

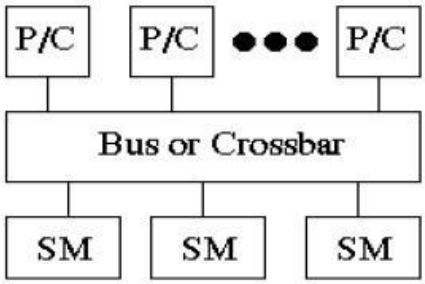
```
pthread_t thread1;   /*handle of special Pthread data type*/
pthread_create(&thread1,NULL,(void *)proc1,(void *)&argv);
pthread_join(thread1, void *status);
```

A thread ID or handle is assigned and obtained from &thread

Symmetric Multiprocessors (SMPs)

Uses commodity microprocessors with on-chip and off-chip caches.

Processors are connected to a shared memory through a high-speed snoopy bus



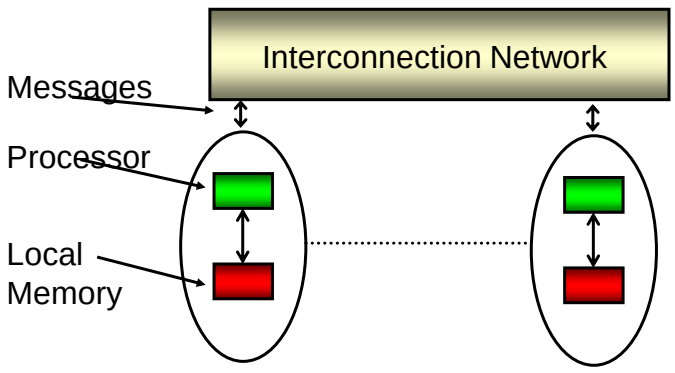
P/C : Microprocessor and cache; SM : Shared memory

- On Some SMPs, a crossbar switch is used in addition to the bus.
- Scalable up to:
  - 48-64 processors
- Equal priority for all processors (except for master or boot CPU)
- Memory coherency maintained by HW
- Multiple I/O Buses for greater Input Output
- All processors see same image of all system resources

- Bus based architecture :
  - Inadequate beyond 8-16 processors
- Crossbar based architecture
  - multistage approach considering I/Os required in hardware
- 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.

Message-Passing Multicomputer

MIMD Complete computers connected through an interconnection network:



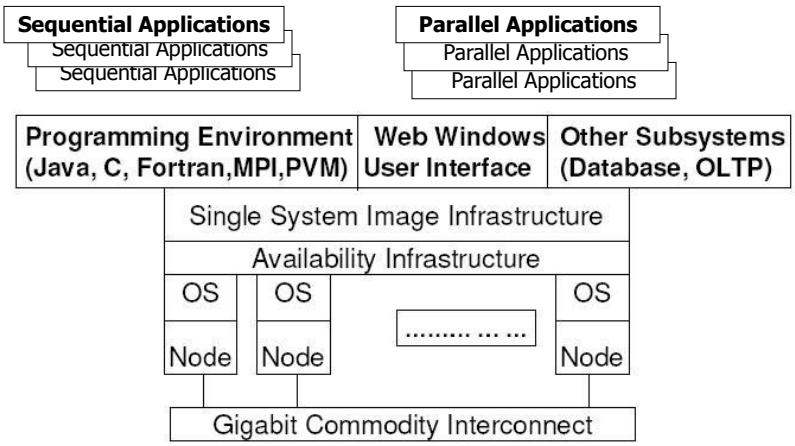
Commercial Parallel Computing Hardware Models

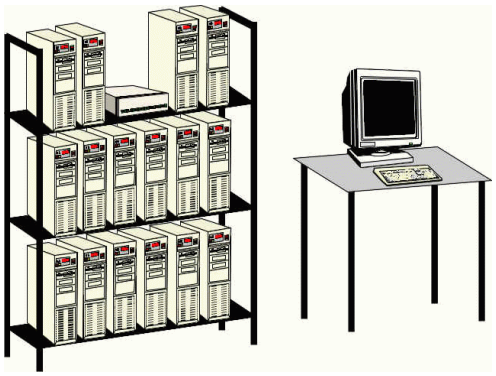
- Single Instruction Multiple Data (SIMD)
- Parallel Vector Processor (PVP)
- Symmetric Multiprocessor (SMP)
- Distributed Shared Memory multiprocessors (DSM)
- Massively Parallel Processor (MPP)
- Cluster of Workstations (COW)

Cluster of Computers – Features

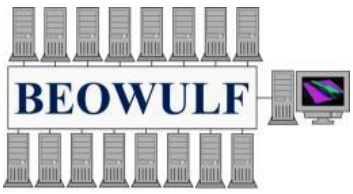
- A **cluster** is a type of parallel or distributed processing system, which consists of a collection of interconnected ***stand-alone computers*** cooperatively working together as a single, integrated computing resources. "***stand-alone***" (whole) computer that can be used on its own (full hardware and OS)
- Collection of nodes physically connected over commodity/ proprietary network
- Cluster computer is a collection of complete independent workstations or Symmetric Multi Processors
- Network is a decisive factors for scalability issues (especially for fine grain applications)
- High volumes driving high performance
- Network using commodity components and proprietary architecture is becoming the trend

Cluster system architecture





Amber and Venus in Grid Computing Lab

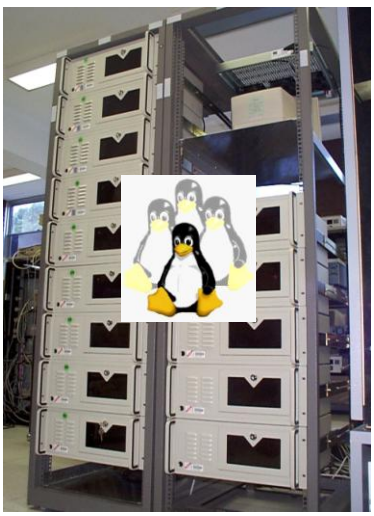


Source: Internet

Common Cluster Modes

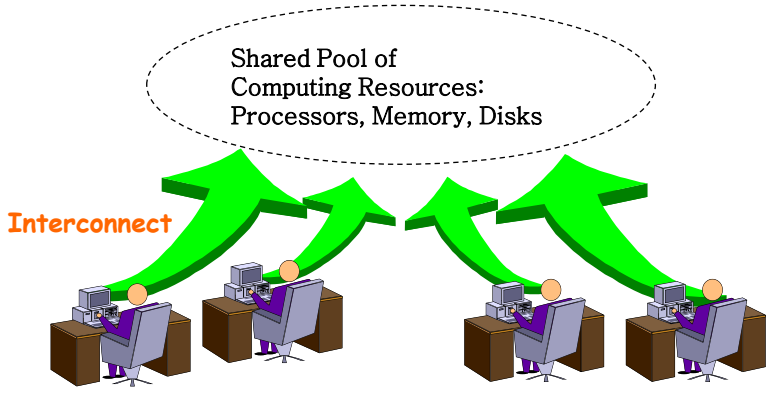
- High Performance (dedicated)
- High Throughput (idle cycle collection)
- High Availability

High Performance Cluster (dedicated mode)



Source: Raj Buyya Slides

High Throughput Cluster (Idle Resource Collection)

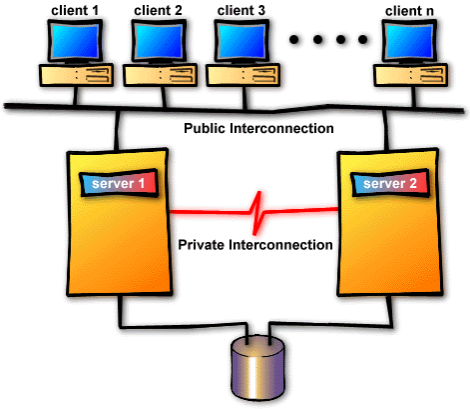


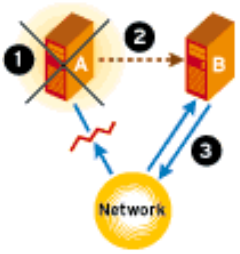
Guarantee at least one workstation to many individuals (when active)

Deliver large % of collective resources to few individuals at any one time

Source: Raj Buyya Slides

High Availability Clusters

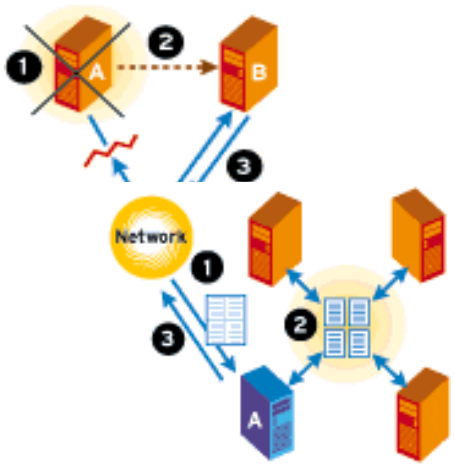




Source: Raj Buyya Slides

HA and HP in the same Cluster

- Best of both Worlds: world is heading towards this configuration)



Source: Raj Buyya Slides

Cluster Components

Prominent Components of Cluster Computers

- Multiple High Performance Computers
  - PCs
  - Workstations
  - SMPs (CLUMPS)
  - Distributed HPC Systems leading to Grid Computing

Source: Raj Buyya Slides

Prominent Components of Cluster Computers

- State of the art Operating Systems
  - Linux (MOSIX, Beowulf, and many more)
  - Microsoft NT (Illinois HPVM, Cornell Velocity)
  - SUN Solaris (Berkeley NOW, C-DAC PARAM)
  - IBM AIX (IBM SP2)
  - HP UX (Illinois - PANDA)
  - Mach (Microkernel based OS) (CMU)
  - Cluster Operating Systems (Solaris MC, SCO Unixware, MOSIX (academic project)
  - OS gluing layers (Berkeley Glunix)

Source: Raj Buyya Slides

Prominent Components of Cluster Computers

- High Performance Networks/Switches
  - Ethernet (10Mbps),
  - Fast Ethernet (100Mbps),
  - Gigabit Ethernet (1Gbps)
  - SCI (Scalable Coherent Interface- MPI- 12µsec latency)
  - ATM (Asynchronous Transfer Mode)
  - Myrinet (1.2Gbps)
  - QsNet (Quadrics Supercomputing World, 5µsec latency for MPI messages)
  - Digital Memory Channel
  - FDDI (Fiber Distributed Data Interface)
  - InfiniBand

Source: Raj Buyya Slides

Prominent Components of Cluster Computers

- Fast Communication Protocols and Services (User Level Communication):
  - Active Messages (Berkeley)
  - Fast Messages (Illinois)
  - U-net (Cornell)
  - XTP (Virginia)
  - Virtual Interface Architecture (VIA)

Source: Raj Buyya Slides

Prominent Components of Cluster Computers

- Parallel Programming Environments and Tools
  - Threads (PCs, SMPs, NOW..)
    - POSIX Threads
    - Java Threads
  - MPI (Message Passing Interface)
    - Linux, NT, on many Supercomputers
  - PVM (Parallel Virtual Machine)
  - Software DSMs (Shmem)
  - Compilers
    - C/C++/Java
    - Parallel programming with C++ (MIT Press book)
  - RAD (rapid application development tools)
    - GUI based tools for PP modeling
  - Debuggers
  - Performance Analysis Tools (PAPI, Performance Application Programming Interface)
  - Visualization Tools

Source: Raj Buyya Slides

Classification of Clusters

- Packaging
  - Compact/Slack
- Control
  - Centralized/Decentralized
- Homogeneity
  - Homogenous/Heterogeneous
- Security
  - Enclosed/Exposed

Classification of Clusters

- Packaging
  - Compact/Slack
- Control
  - Centralized/Decentralized
- Homogeneity
  - Homogenous/Heterogeneous
- Security
  - Enclosed/Exposed

Classify using four orthogonal attributes packaging, control, homogeneity and security

Dedicated v/s Enterprise Clusters

- Dedicated
  - Compact, Centralized, Homogenous, Enclosed
- Enterprise
  - Geographically distributed (slack), Decentralized, heterogeneous, Exposed

Users View of Cluster

The users view the entire cluster as **Single system**, which has multiple processors. The user could say: “Execute my application using five processors.” This is different from a distributed system.

- **Single Entry**
- **Single File Hierarchy**
- **Single Networking**
- **Single Input/Output**
- **Single Point of Control**
- **Single Memory Space**
- **Single Job Management System**
- **Single User Interface**
- **Single Process Space**
- **Single System**
- **Symmetry**
- **Location Transparent**

Job Management

- ❖ Global job management
- ❖ Global system management and configuration
- ❖ Group based scheduling and resource allocation
- ❖ Idle resource detection
- ❖ Co-scheduling of parallel programs
- ❖ Load Sharing Facility (LSF)

High Availability

- ❖ Fault tolerance
- ❖ Check-pointing

A major Issues in Clusters Design

- ➢ Enhanced Performance (performance @low cost)
- ➢ Enhanced System Image (look-and-feel of one system)
- ➢ Size Scalability (physical & application)
- ➢ Fast Communication (networks & protocols)
- ➢ Load Balancing (CPU, Net, Memory, Disk)
- ➢ Security and Encryption (clusters of clusters)
- ➢ Distributed Environment (Social issues)
- ➢ Manageability (admin. And control)
- ➢ Programmability (simple API if required)
- ➢ Applicability (cluster-aware and non-aware app.)

Some Cluster Systems: Comparison

Project	Platform	Communications	OS	Other
Beowulf	PCs	Multiple Ethernet with TCP/IP	Linux and Grendel	MPI/PVM. Sockets and HPF
Berkeley Now	Solaris-based PCs and workstations	Myrinet and Active Messages	Solaris + GLUnix + xFS	AM, PVM, MPI, HPF, Split-C
HPVM	PCs	Myrinet with Fast Messages	NT or Linux connection and global resource manager + LSF	Java-fronted, FM, Sockets, Global Arrays, SHEMEM and MPI
Solaris MC	Solaris-based PCs and workstations	Solaris-supported	Solaris + Globalization layer	C++ and CORBA

Major References

Kai Hwang, Zhiwei Xu, Scalable Parallel Computing (Technology Architecture Programming) McGraw Hill Newyork (1997).

Culler David E, Jaswinder Pal Singh with Anoop Gupta, Parallel Computer Architecture, A Hardware/Software Approach, Morgan Kaufmann Publishers, Inc, (1999), Reprinted in 2004.

Barry Wilkinson And Michael Allen, Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, Upper Saddle River, NJ, 1999.

RajKumar Buyya, High Performance Cluster Computing, Programming and Applications, Prentice Hall, 1999.