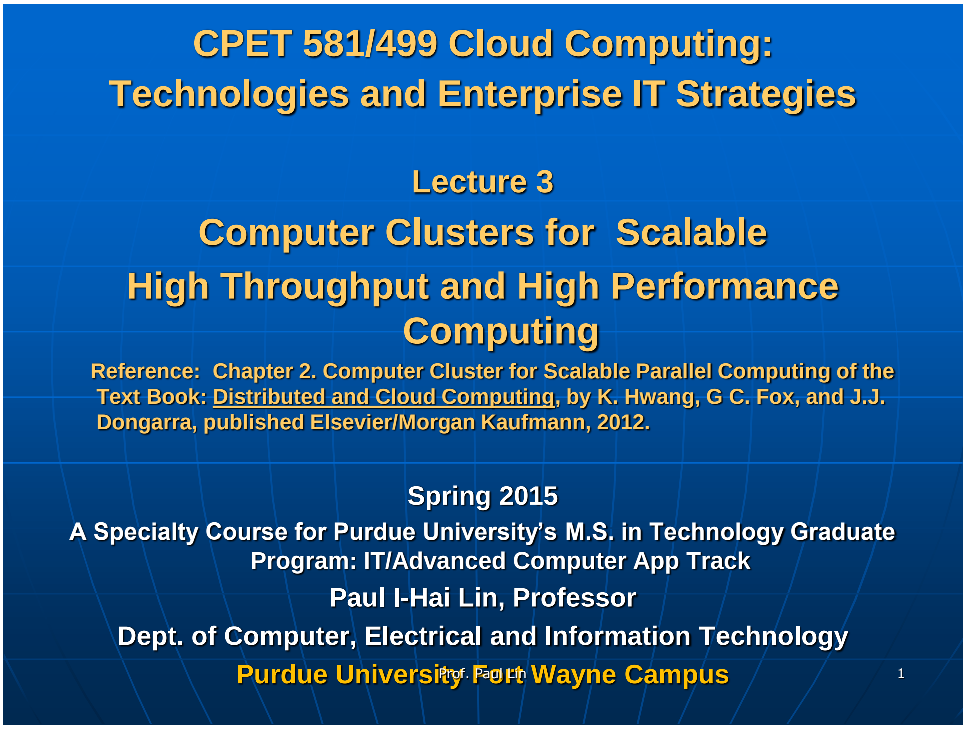


CPET 581/499 Cloud Computing: Technologies and Enterprise IT Strategies

Lecture 3

Computer Clusters for Scalable High Throughput and High Performance Computing

Reference: Chapter 2. Computer Cluster for Scalable Parallel Computing of the Text Book: Distributed and Cloud Computing, by K. Hwang, G C. Fox, and J.J. Dongarra, published Elsevier/Morgan Kaufmann, 2012.

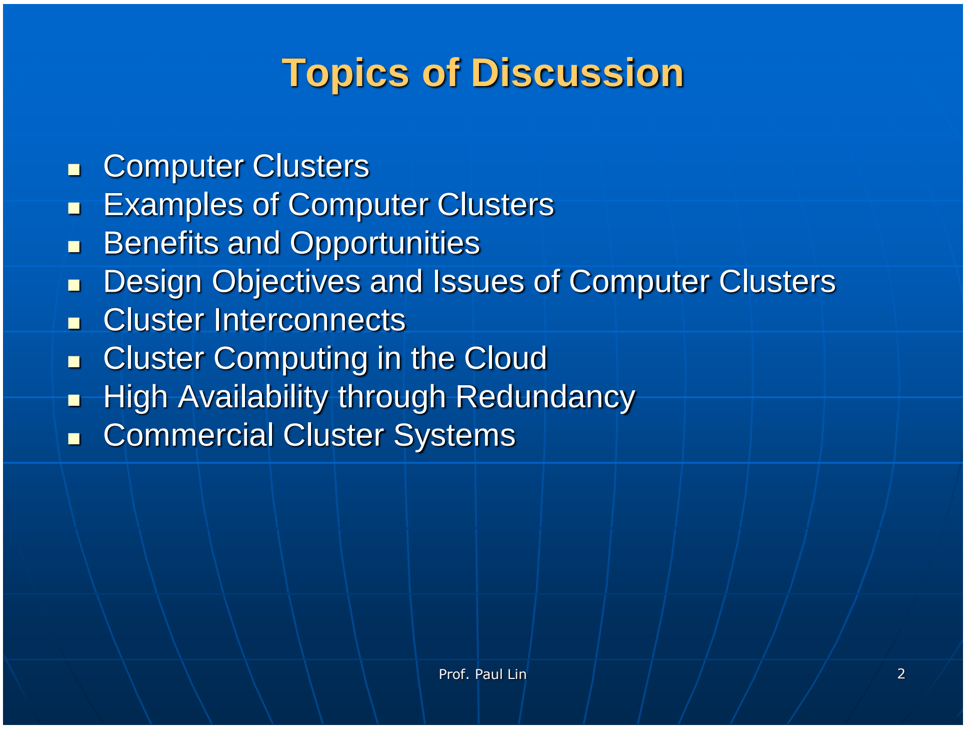
Spring 2015

A Specialty Course for Purdue University's M.S. in Technology Graduate Program: IT/Advanced Computer App Track

Paul I-Hai Lin, Professor

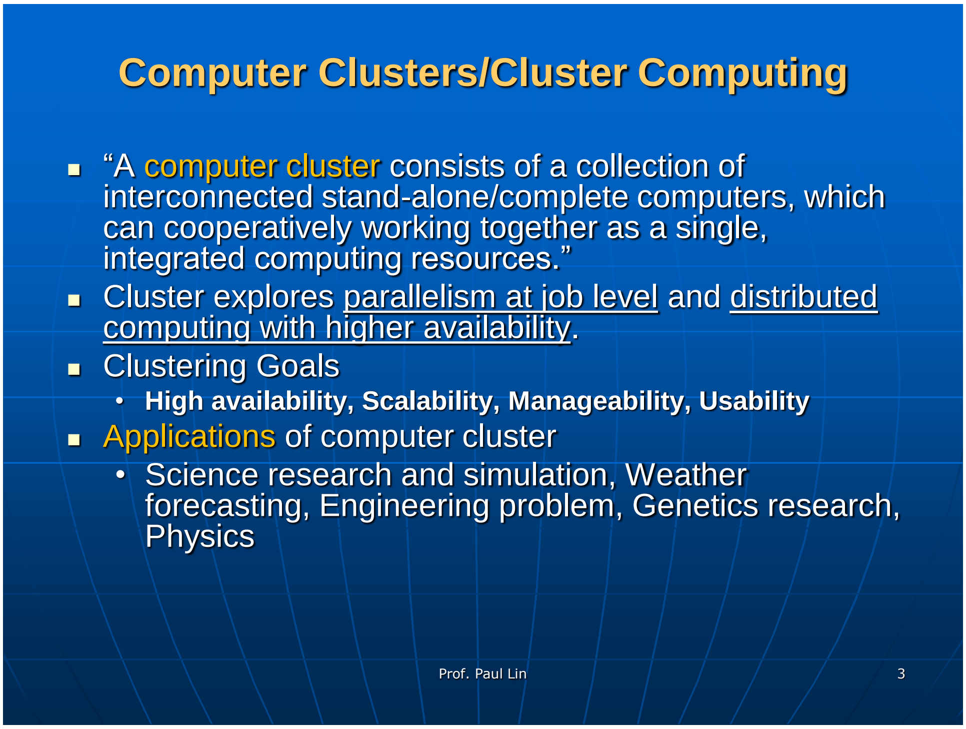
Dept. of Computer, Electrical and Information Technology

Purdue University Fort Wayne Campus



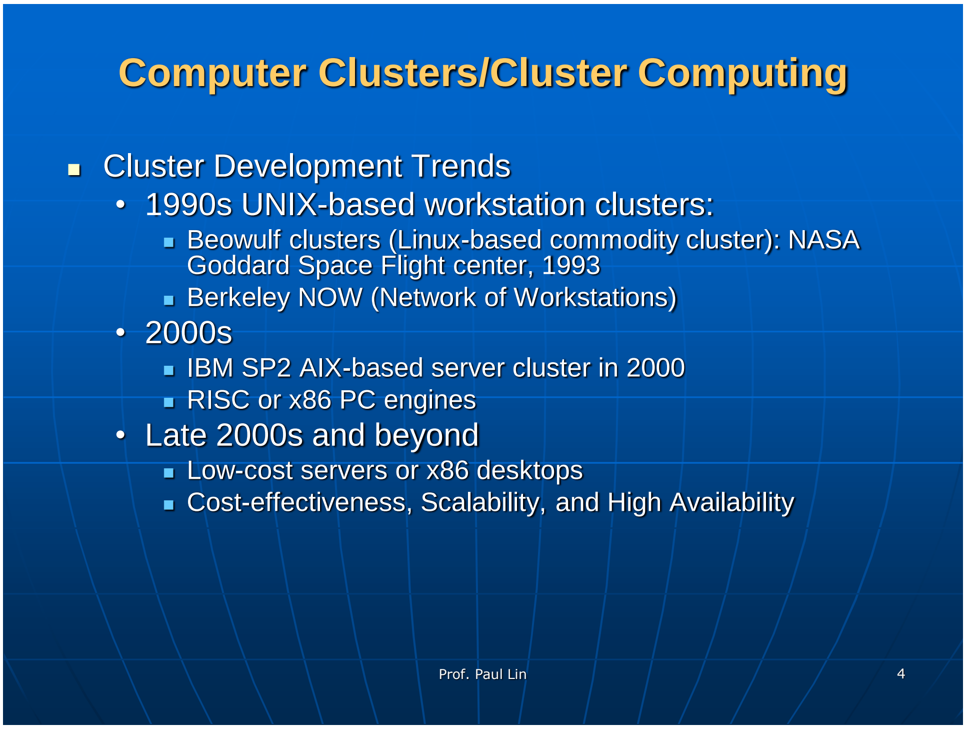
Topics of Discussion

- Computer Clusters
- Examples of Computer Clusters
- Benefits and Opportunities
- Design Objectives and Issues of Computer Clusters
- Cluster Interconnects
- Cluster Computing in the Cloud
- High Availability through Redundancy
- Commercial Cluster Systems



Computer Clusters/Cluster Computing

- "A computer cluster consists of a collection of interconnected stand-alone/complete computers, which can cooperatively working together as a single, integrated computing resources."
- Cluster explores parallelism at job level and distributed computing with higher availability.
- Clustering Goals
  - **High availability, Scalability, Manageability, Usability**
- Applications of computer cluster
  - Science research and simulation, Weather forecasting, Engineering problem, Genetics research, Physics



Computer Clusters/Cluster Computing

- Cluster Development Trends
  - 1990s UNIX-based workstation clusters:
    - Beowulf clusters (Linux-based commodity cluster): NASA Goddard Space Flight center, 1993
    - Berkeley NOW (Network of Workstations)
  - 2000s
    - IBM SP2 AIX-based server cluster in 2000
    - RISC or x86 PC engines
  - Late 2000s and beyond
    - Low-cost servers or x86 desktops
    - Cost-effectiveness, Scalability, and High Availability

Types of Computer Clusters

- High availability/performance clusters
- Load-balancing/leveling clusters
- Web-service clusters
- Storage clusters
  - Parallel file systems
- Database clusters
  - Oracle parallel server

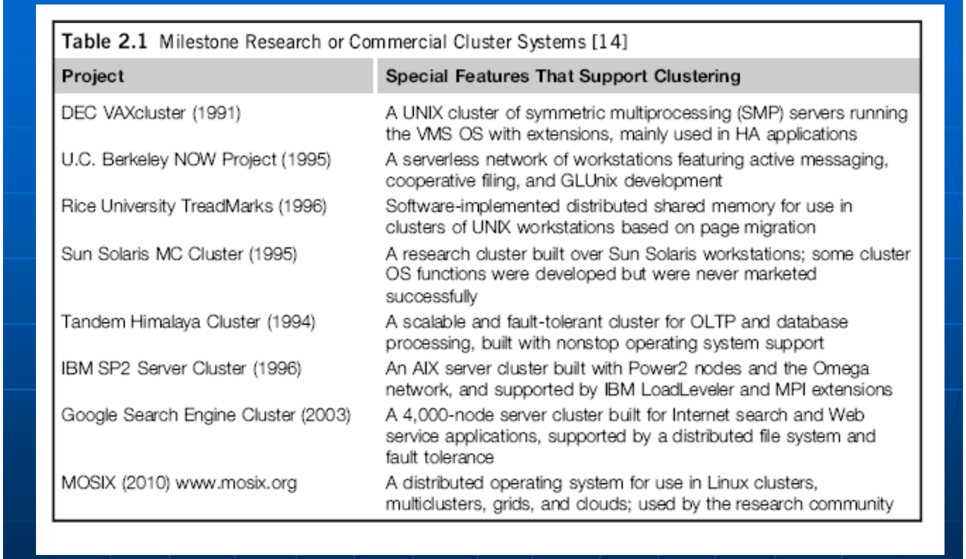
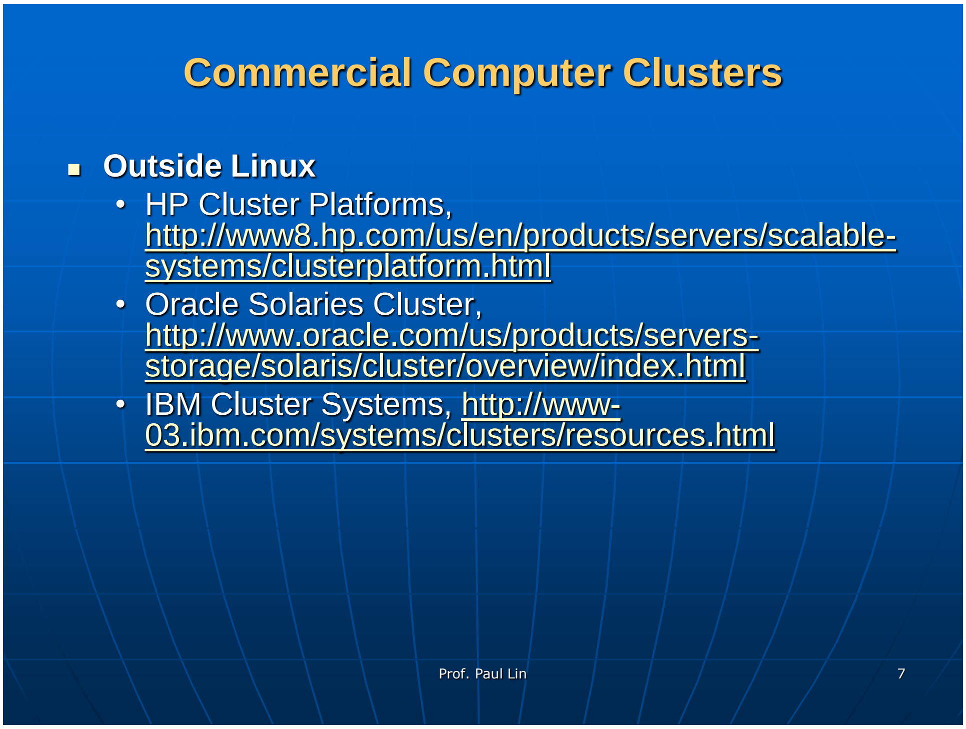


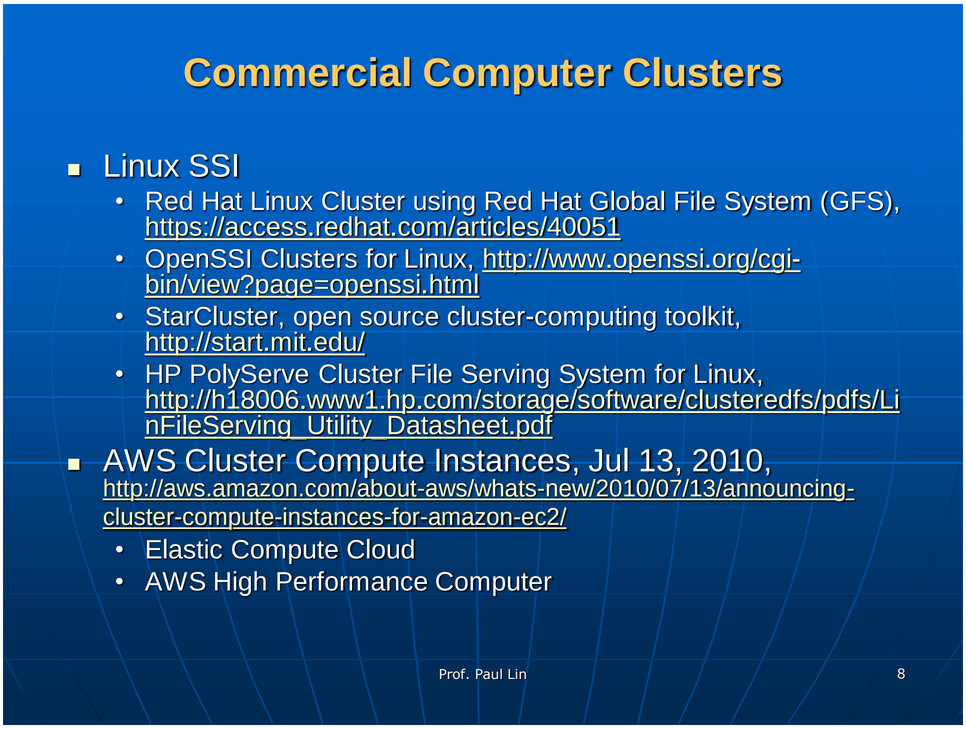
Table 2.1 Milestone Research or Commercial Cluster Systems [14]

Project	Special Features That Support Clustering
DEC VAXcluster (1991)	A UNIX cluster of symmetric multiprocessing (SMP) servers running the VMS OS with extensions, mainly used in HA applications
U.C. Berkeley NOW Project (1995)	A serverless network of workstations featuring active messaging, cooperative filing, and GLUnix development
Rice University TreadMarks (1996)	Software-implemented distributed shared memory for use in clusters of UNIX workstations based on page migration
Sun Solaris MC Cluster (1995)	A research cluster built over Sun Solaris workstations; some cluster OS functions were developed but were never marketed successfully
Tandem Himalaya Cluster (1994)	A scalable and fault-tolerant cluster for OLTP and database processing, built with nonstop operating system support
IBM SP2 Server Cluster (1996)	An AIX server cluster built with Power2 nodes and the Omega network, and supported by IBM LoadLeveler and MPI extensions
Google Search Engine Cluster (2003)	A 4,000-node server cluster built for Internet search and Web service applications, supported by a distributed file system and fault tolerance
MOSIX (2010) www.mosix.org	A distributed operating system for use in Linux clusters, multiclusters, grids, and clouds; used by the research community



Commercial Computer Clusters

- **Outside Linux**
  - HP Cluster Platforms, http://www8.hp.com/us/en/products/servers/scalable-systems/clusterplatform.html
  - Oracle Solaries Cluster, http://www.oracle.com/us/products/servers-storage/solaris/cluster/overview/index.html
  - IBM Cluster Systems, http://www-03.ibm.com/systems/clusters/resources.html

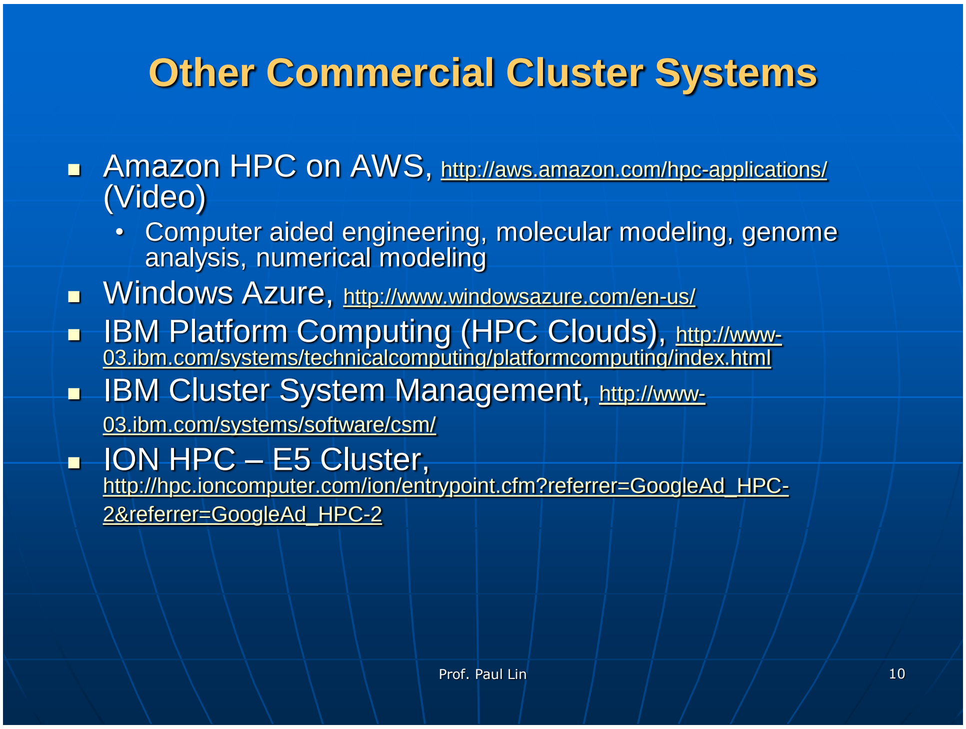


Commercial Computer Clusters

- Linux SSI
  - Red Hat Linux Cluster using Red Hat Global File System (GFS), https://access.redhat.com/articles/40051
  - OpenSSI Clusters for Linux, http://www.openssi.org/cgi-bin/view?page=openssi.html
  - StarCluster, open source cluster-computing toolkit, http://start.mit.edu/
  - HP PolyServe Cluster File Serving System for Linux, http://h18006.www1.hp.com/storage/software/clusteredfs/pdfs/LinFileServing_Utility_Datasheet.pdf
- AWS Cluster Compute Instances, Jul 13, 2010, http://aws.amazon.com/about-aws/whats-new/2010/07/13/announcing-cluster-compute-instances-for-amazon-ec2/
  - Elastic Compute Cloud
  - AWS High Performance Computer

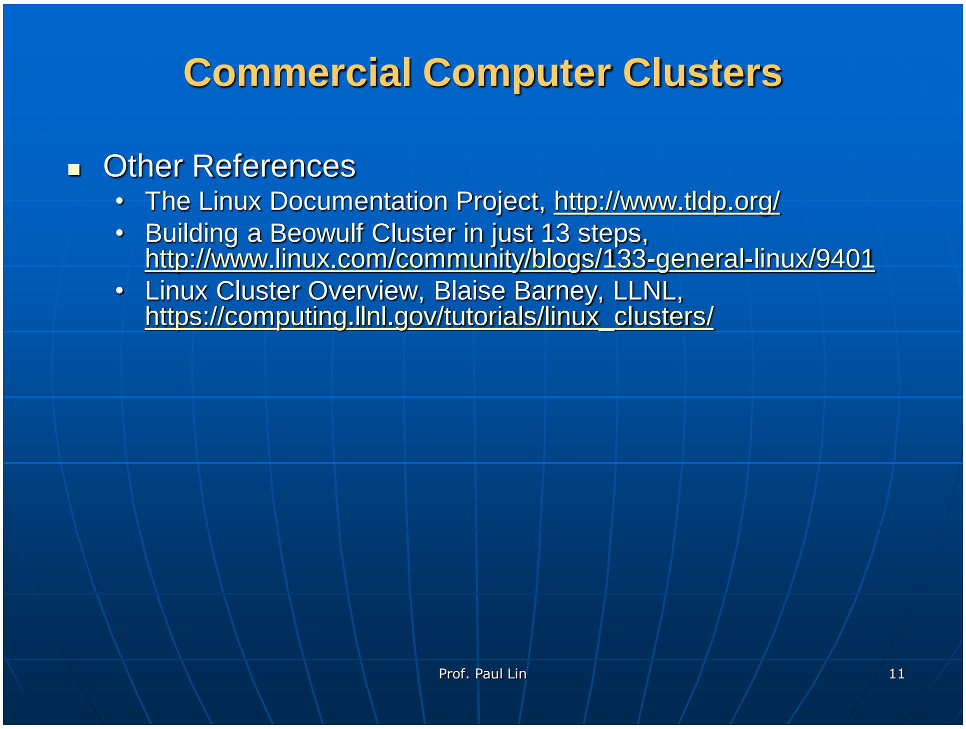
Commercial Single System Image Cluster Systems and Technology

- Open SSI Cluster Project,
- IBM z/VM, SSI and Live Guest Relocation, http://www.redbooks.ibm.com/abstracts/sg248006.html?Open
- Ubuntu SSI Cluster, https://wiki.ubuntu.com/EasyUbuntuClustering/UbuntuKerrighedClusterGuide



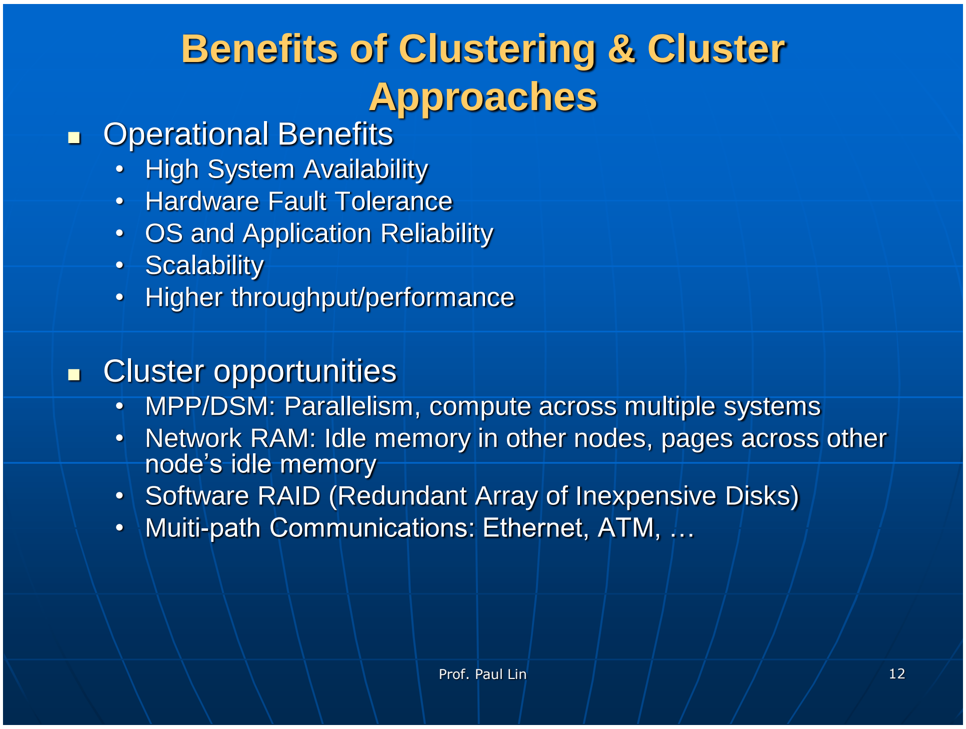
Other Commercial Cluster Systems

- Amazon HPC on AWS, http://aws.amazon.com/hpc-applications/ (Video)
  - Computer aided engineering, molecular modeling, genome analysis, numerical modeling
- Windows Azure, http://www.windowsazure.com/en-us/
- IBM Platform Computing (HPC Clouds), http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/index.html
- IBM Cluster System Management, http://www-03.ibm.com/systems/software/csm/
- ION HPC – E5 Cluster, http://hpc.ioncomputer.com/ion/entrypoint.cfm?referrer=GoogleAd_HPC-2&referrer=GoogleAd_HPC-2



Commercial Computer Clusters

- Other References
  - The Linux Documentation Project, http://www.tldp.org/
  - Building a Beowulf Cluster in just 13 steps, http://www.linux.com/community/blogs/133-general-linux/9401
  - Linux Cluster Overview, Blaise Barney, LLNL, https://computing.llnl.gov/tutorials/linux_clusters/



Benefits of Clustering & Cluster Approaches

- Operational Benefits
  - High System Availability
  - Hardware Fault Tolerance
  - OS and Application Reliability
  - Scalability
  - Higher throughput/performance
- Cluster opportunities
  - MPP/DSM: Parallelism, compute across multiple systems
  - Network RAM: Idle memory in other nodes, pages across other node’s idle memory
  - Software RAID (Redundant Array of Inexpensive Disks)
  - Muiti-path Communications: Ethernet, ATM, …

Cluster Interconnect

- NAS (Network Attached Storage), SAN (Storage Area Network)

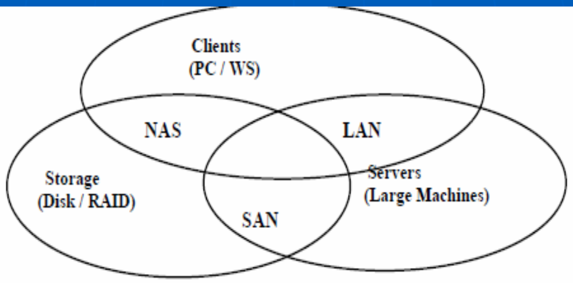


Figure 2.9 Three interconnection networks connecting servers, client hosts and storage devices: the LAN between client hosts and servers. The SAN between servers and disk arrays, and the NAS between clients and storage system.

Design Issues of Computer Clusters

- Communication Networks (latency, bandwidth, $cost)
  - Connection Types:
    - WAN (Wide Area Network),
    - LAN (Local Area Network),
    - SAN (Storage Area Network)
- CPU architecture (performance, $cost)
- Node architecture (local, remote communication, $cost)
- Space considerations (cooling/ventilations, power requirements)

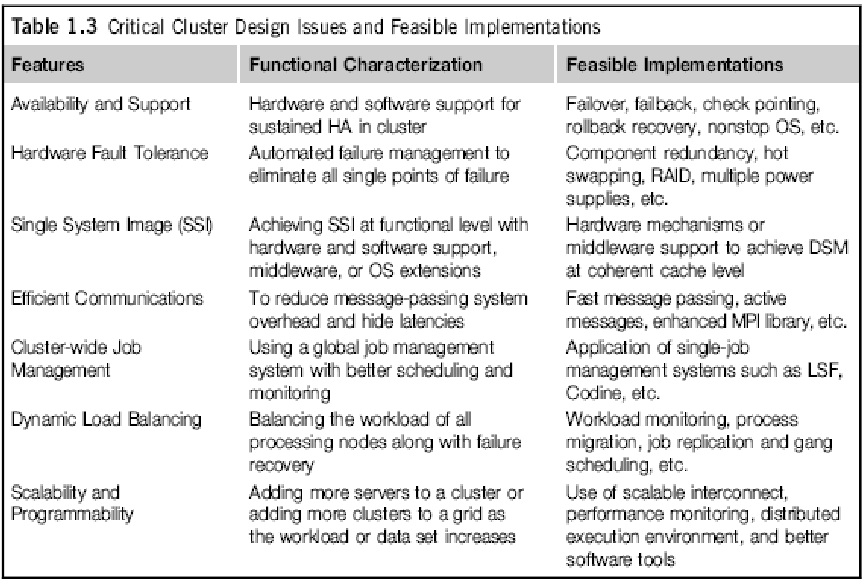


Table 1.3 Critical Cluster Design Issues and Feasible Implementations

Features	Functional Characterization	Feasible Implementations
Availability and Support	Hardware and software support for sustained HA in cluster	Failover, failback, check pointing, rollback recovery, nonstop OS, etc.
Hardware Fault Tolerance	Automated failure management to eliminate all single points of failure	Component redundancy, hot swapping, RAID, multiple power supplies, etc.
Single System Image (SSI)	Achieving SSI at functional level with hardware and software support, middleware, or OS extensions	Hardware mechanisms or middleware support to achieve DSM at coherent cache level
Efficient Communications	To reduce message-passing system overhead and hide latencies	Fast message passing, active messages, enhanced MPI library, etc.
Cluster-wide Job Management	Using a global job management system with better scheduling and monitoring	Application of single-job management systems such as LSF, Codine, etc.
Dynamic Load Balancing	Balancing the workload of all processing nodes along with failure recovery	Workload monitoring, process migration, job replication and gang scheduling, etc.
Scalability and Programmability	Adding more servers to a cluster or adding more clusters to a grid as the workload or data set increases	Use of scalable interconnect, performance monitoring, distributed execution environment, and better software tools

Figure 2.4 A Basic Cluster Architecture

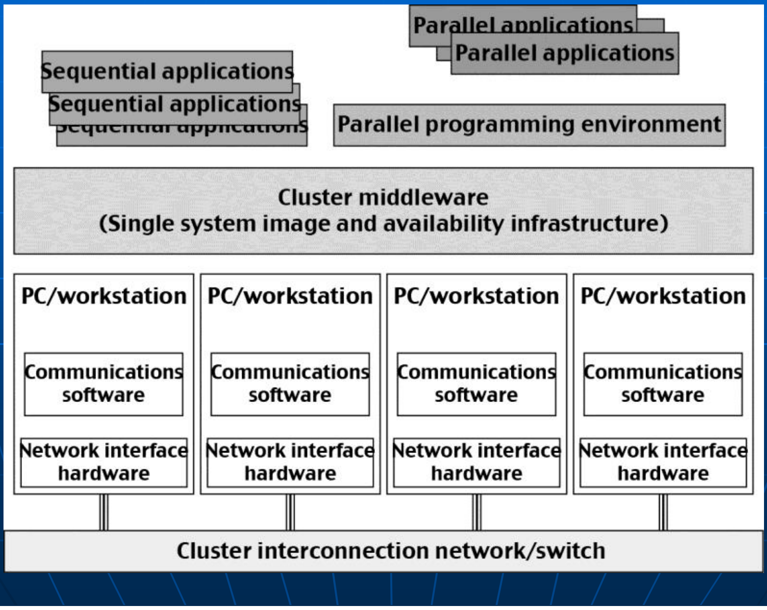
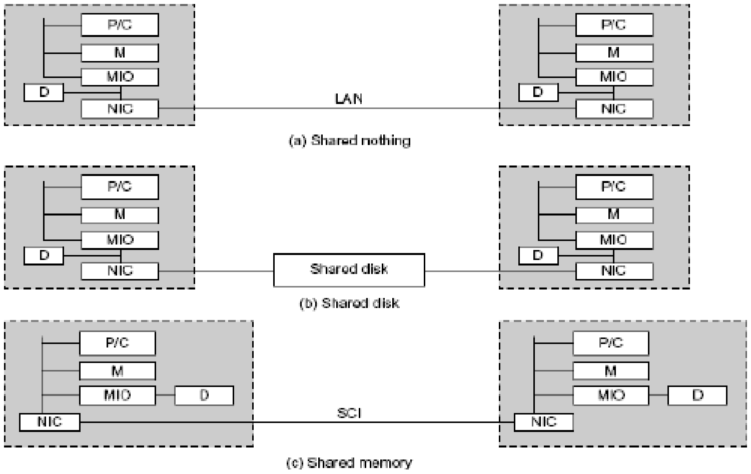


Figure 2.5 Resource Sharing in Cluster of Computers



(a) Shared nothing

(b) Shared disk

(c) Shared memory

FIGURE 2.5

Three ways to connect cluster nodes (P/C: Processor and Cache; M: Memory; D: Disk; NIC: Network Interface Circuitry; MIO: Memory-I/O Bridge.)

(Courtesy of Hwang and Xu [14])

Example 2-1 Modular Packaging of the IBM BlueGene/L System

- The world fastest message-passing MPP built jointly by IBM and LLNL (Lawrence Livermore National Laboratory) teams in 2005, founded by U.S. DoE Accelerated Strategic Computing Initiative (ASCI) Research program
- Scalable MPP (massive parallel processing) system
  - 64 physical racks, interconnected by huge 3D 64 x 32 x 32 torus network
  - A total of 65,536 nodes, two PowerPC 449 FP2 processor per node
- 136 Tflops performance in 2005
- Upgraded to 478 Tflops in 2007
- IBM Blue Gene Project page, http://www.research.ibm.com/bluegene/index.html
- Linux Cluster Overview, Blaise Barney, LLNL, https://computing.llnl.gov/tutorials/linux_clusters/

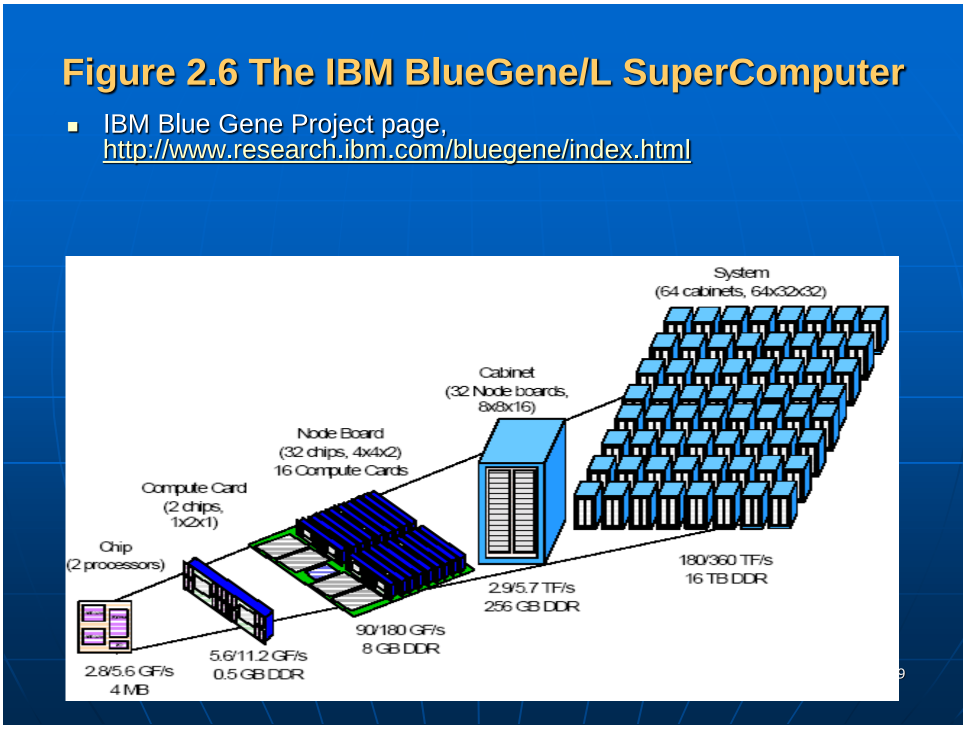
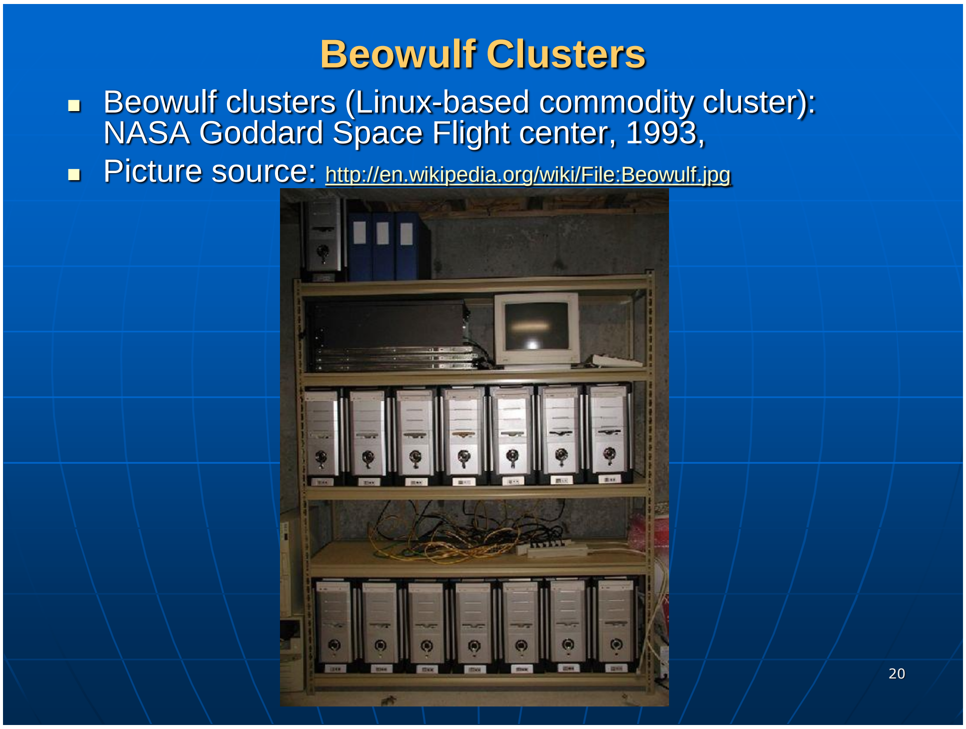


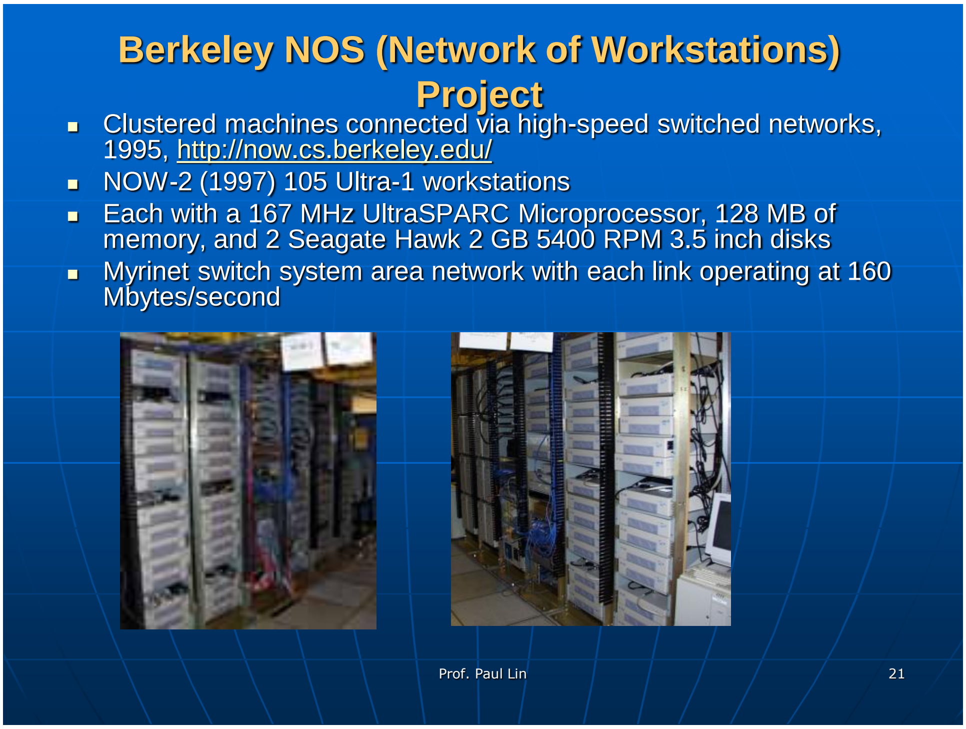
Figure 2.6 The IBM BlueGene/L SuperComputer

- IBM Blue Gene Project page, http://www.research.ibm.com/bluegene/index.html



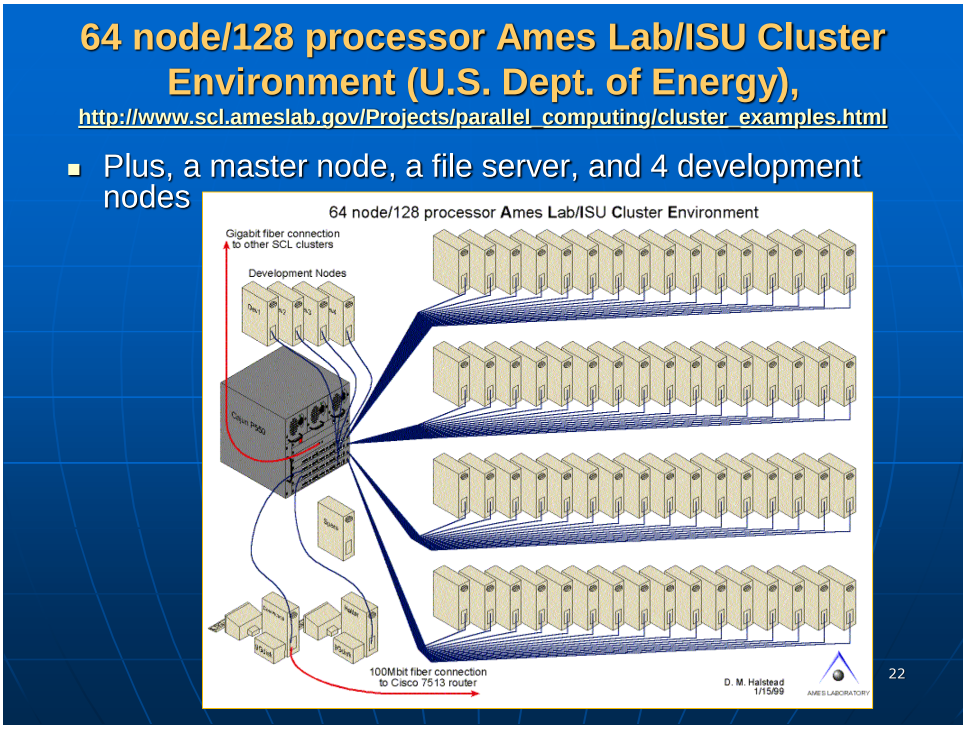
Beowulf Clusters

- Beowulf clusters (Linux-based commodity cluster): NASA Goddard Space Flight center, 1993,
- Picture source: http://en.wikipedia.org/wiki/File:Beowulf.jpg



Berkeley NOS (Network of Workstations) Project

- Clustered machines connected via high-speed switched networks, 1995, http://now.cs.berkeley.edu/
- NOW-2 (1997) 105 Ultra-1 workstations
- Each with a 167 MHz UltraSPARC Microprocessor, 128 MB of memory, and 2 Seagate Hawk 2 GB 5400 RPM 3.5 inch disks
- Myrinet switch system area network with each link operating at 160 Mbytes/second

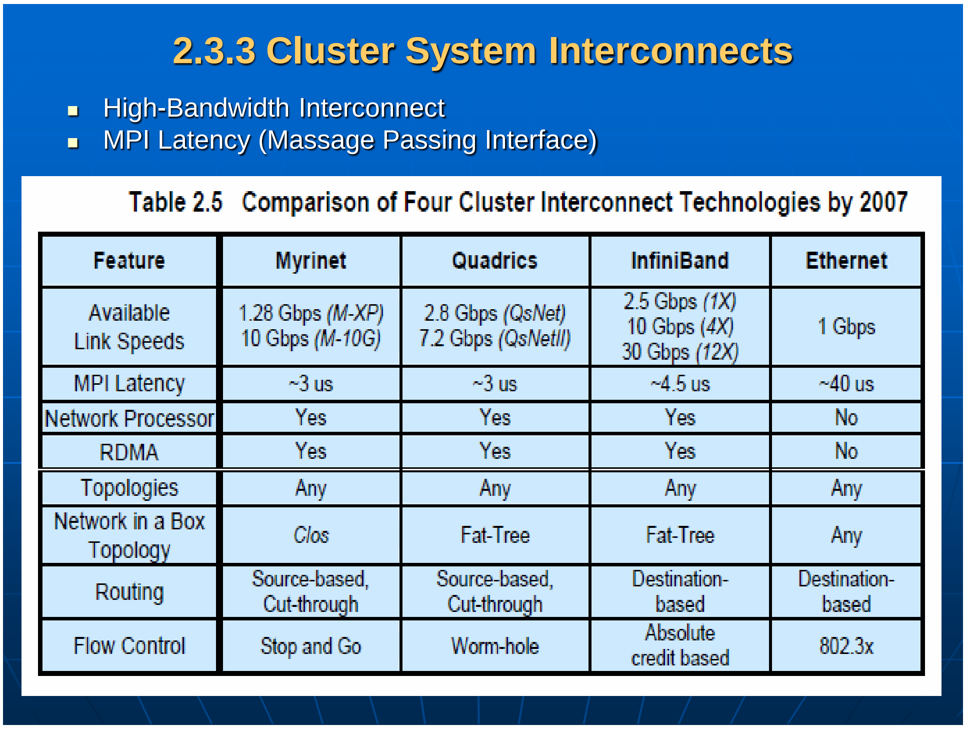


64 node/128 processor Ames Lab/ISU Cluster Environment (U.S. Dept. of Energy), http://www.scl.ameslab.gov/Projects/parallel_computing/cluster_examples.html

- Plus, a master node, a file server, and 4 development nodes

64 node/128 processor Ames Lab/ISU Cluster Environment

Gigabit fiber connection to other SCL clusters
Development Nodes
Dev1
Cajun P550
Spare
100Mbit fiber connection to Cisco 7513 router
D. M. Halstead 1/15/99
AMES LABORATORY

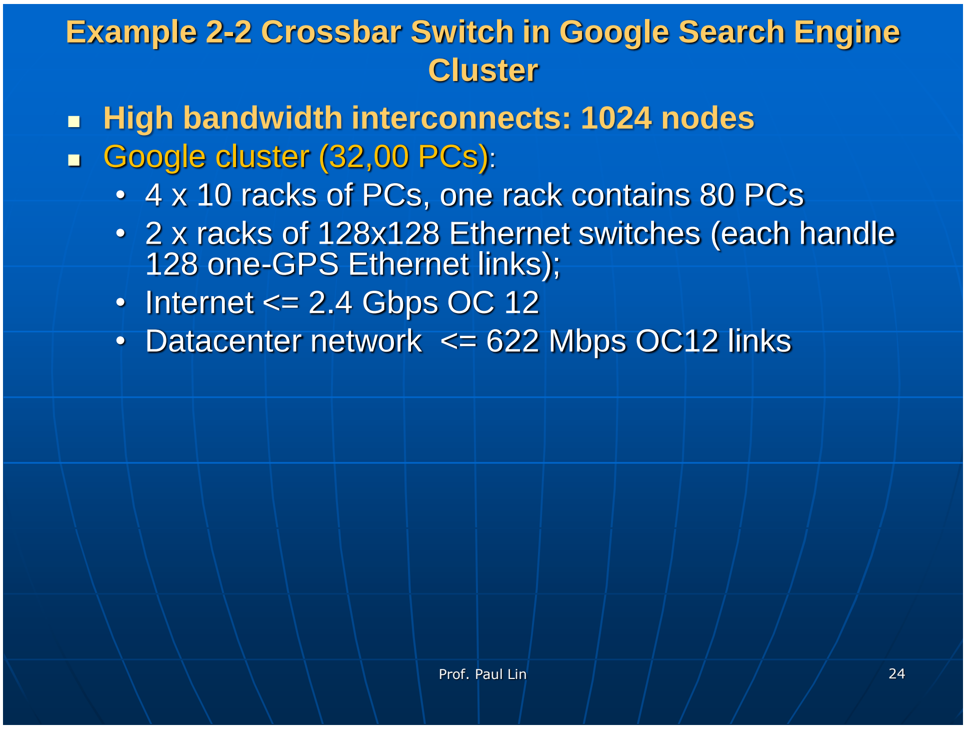


2.3.3 Cluster System Interconnects

- High-Bandwidth Interconnect
- MPI Latency (Massage Passing Interface)

Table 2.5 Comparison of Four Cluster Interconnect Technologies by 2007

Feature	Myrinet	Quadrics	InfiniBand	Ethernet
Available Link Speeds	1.28 Gbps *(M-XP)* 10 Gbps *(M-10G)*	2.8 Gbps *(QsNet)* 7.2 Gbps *(QsNetII)*	2.5 Gbps *(1X)* 10 Gbps *(4X)* 30 Gbps *(12X)*	1 Gbps
MPI Latency	~3 us	~3 us	~4.5 us	~40 us
Network Processor	Yes	Yes	Yes	No
RDMA	Yes	Yes	Yes	No
Topologies	Any	Any	Any	Any
Network in a Box Topology	*Clos*	Fat-Tree	Fat-Tree	Any
Routing	Source-based, Cut-through	Source-based, Cut-through	Destination-based	Destination-based
Flow Control	Stop and Go	Worm-hole	Absolute credit based	802.3x



Example 2-2 Crossbar Switch in Google Search Engine Cluster

- **High bandwidth interconnects: 1024 nodes**
- Google cluster (32,00 PCs):
  - 4 x 10 racks of PCs, one rack contains 80 PCs
  - 2 x racks of 128x128 Ethernet switches (each handle 128 one-GPS Ethernet links);
  - Internet <= 2.4 Gbps OC 12
  - Datacenter network <= 622 Mbps OC12 links

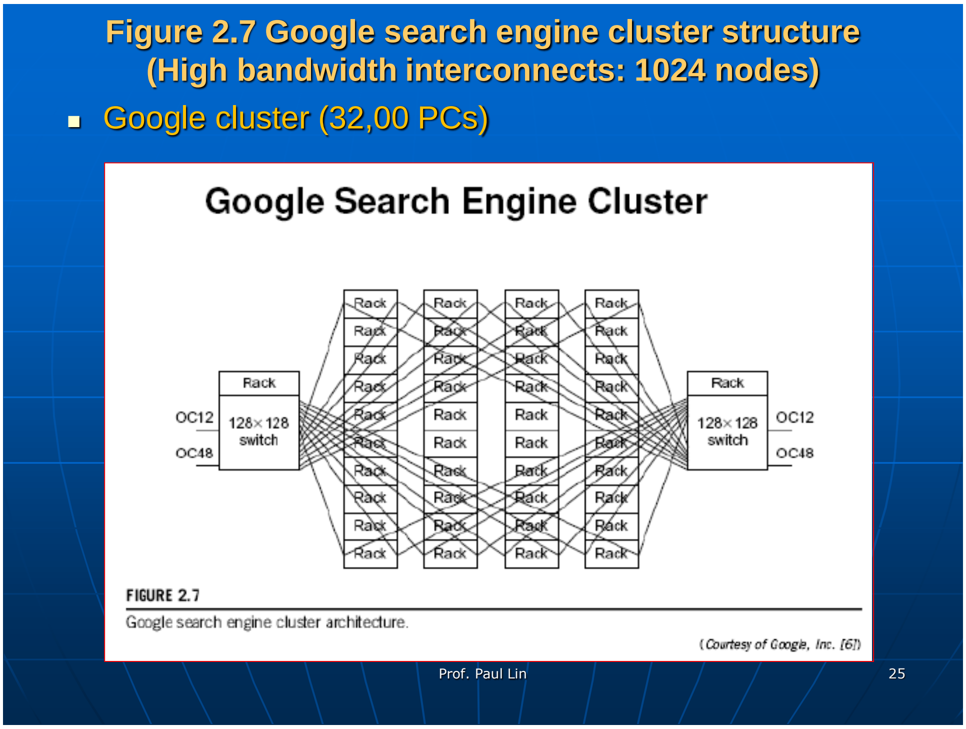


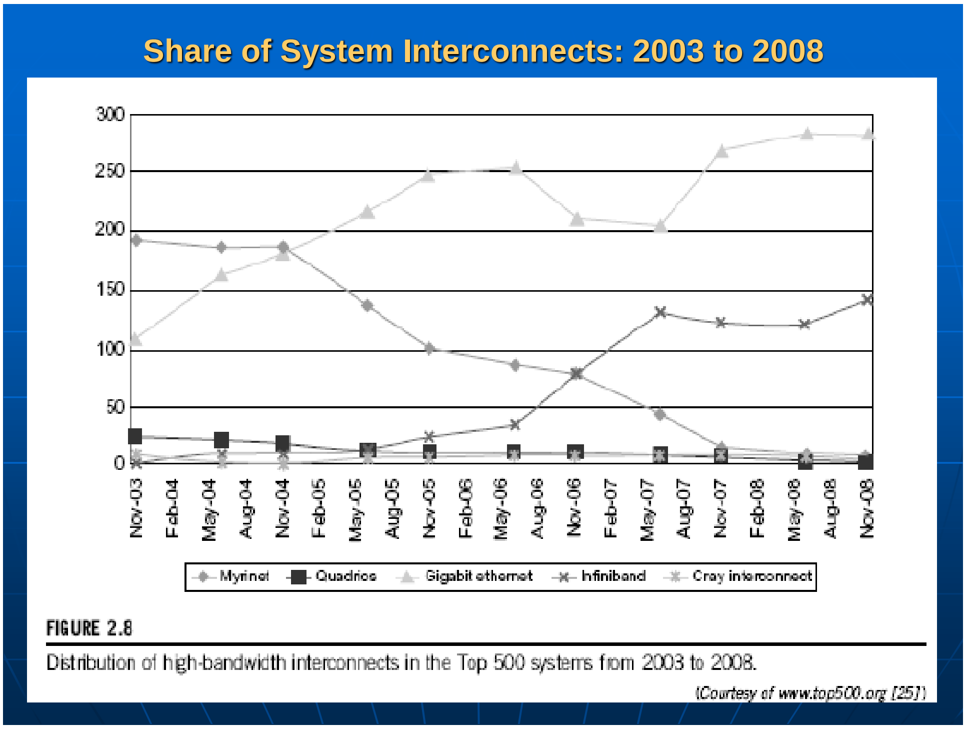
Figure 2.7 Google search engine cluster structure (High bandwidth interconnects: 1024 nodes)

- Google cluster (32,00 PCs)

FIGURE 2.7

Google search engine cluster architecture.

(Courtesy of Google, Inc. [6])



Share of System Interconnects: 2003 to 2008

FIGURE 2.8

Distribution of high-bandwidth interconnects in the Top 500 systems from 2003 to 2008.

(Courtesy of www.top500.org [25])

Example 2.3 The InfiniBand Architecture [8]

- A switch-based point-to-point interconnect architecture
- Support virtual interface architecture (VIA) for distributed messaging
- HCA (Host Channel Adapter), Target Channel Adapter (TCA)

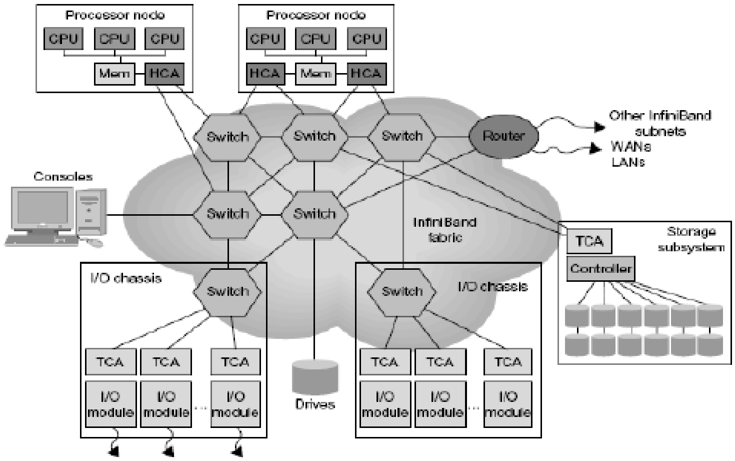


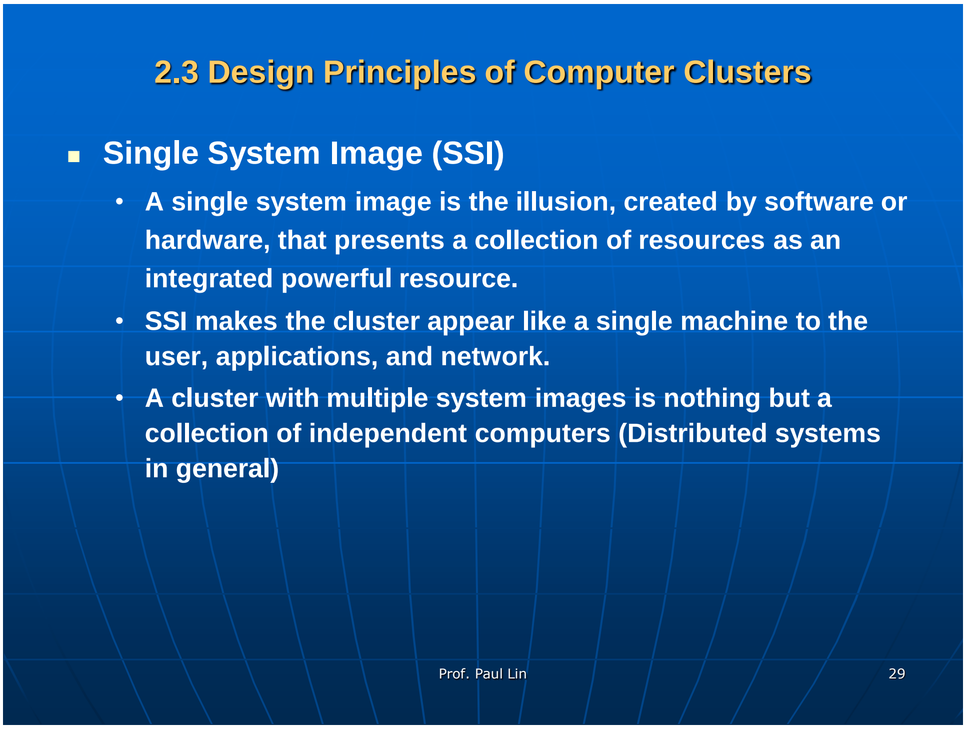
FIGURE 2.9

The InfiniBand system fabric built in a typical high-performance computer cluster.

(Courtesy of Celebioglu, et al. [8])

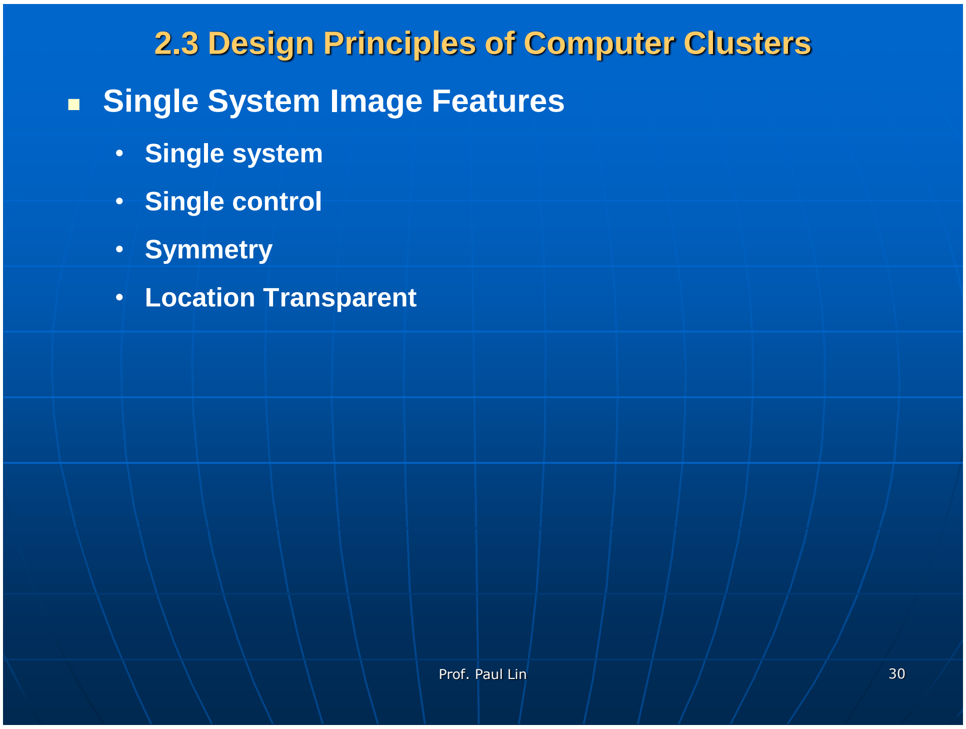
2.2.4 Hardware, Software, and Middleware Support

- **Cluster Middleware**
  - **Resides Between OS and Applications and offers in infrastructure for supporting:**
    - **Single System Image (SSI)**
    - **System Availability (SA)**
  - **SSI makes collection appear as single machine (globalized view of system resources). Telnet cluster.myinstitute.edu**
  - **Checkpointing and process migration**



2.3 Design Principles of Computer Clusters

- Single System Image (SSI)
  - A single system image is the illusion, created by software or hardware, that presents a collection of resources as an integrated powerful resource.
  - SSI makes the cluster appear like a single machine to the user, applications, and network.
  - A cluster with multiple system images is nothing but a collection of independent computers (Distributed systems in general)



2.3 Design Principles of Computer Clusters

- Single System Image Features
  - Single system
  - Single control
  - Symmetry
  - Location Transparent

2.3 Design Principles of Computer Clusters

- Desired Single System Image
  - Single Entry Point
  - Single File Hierarchy:
    - xFS (x file system by Silicon Graphics in 1993), AFS (Andrew file system, Carnegie Mellon University's Andrew Project), Solaris MC Proxy
  - Single Control Point: Management from single GUI
  - Single virtual networking over multiple physical networks
  - Single memory space - Network RAM / DSM (Distributed shared memory)
  - Single Job Management: GlUnix, Codine, LSF, etc.
  - Single User Interface: Like CDE (Common Desktop Environment) in Solaris/NT

2.3 Design Principles of Computer Clusters

- Desired Single System Image
  - Single Entry Point (design issues on how to)
    - Home directory, Authentication, Multiple connections, Host Failure
- Example 2.5: Single Entry Point to access a Cluster from any physical point

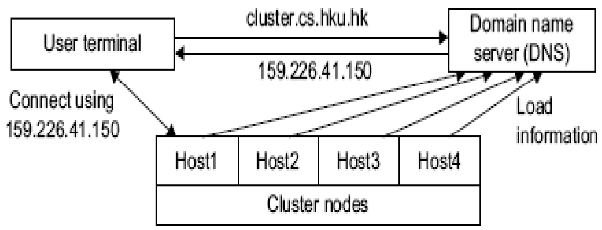
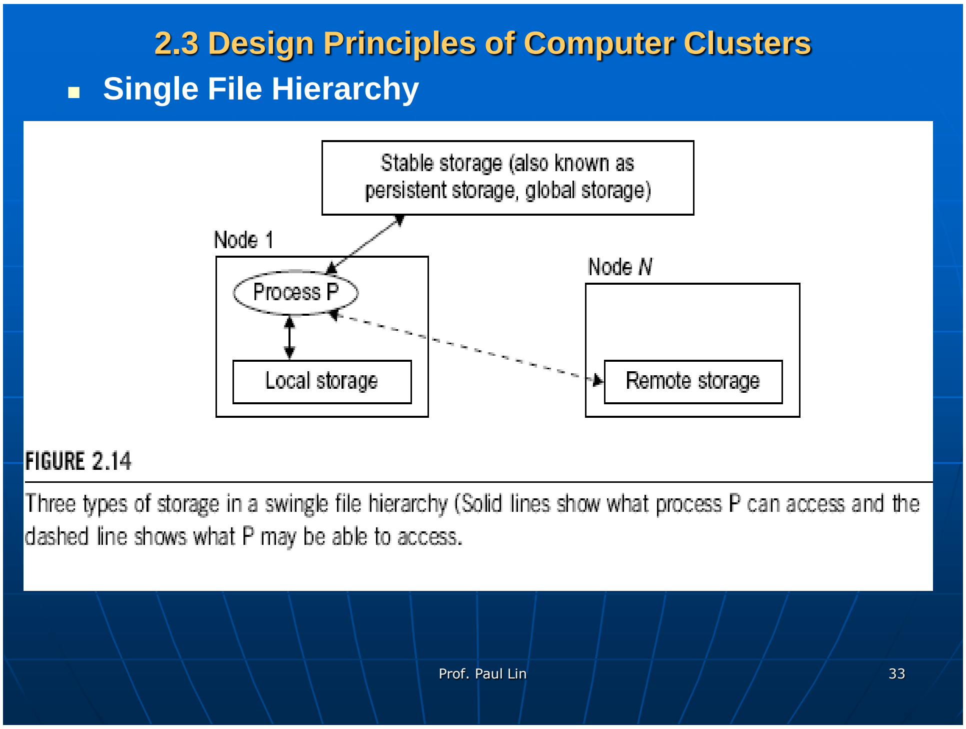


FIGURE 2.13

Realizing a single etry point using a load-balancing domain name system (DNS).

(Courtesy of Hwang and Xu [14])



2.3 Design Principles of Computer Clusters

- Single File Hierarchy

Stable storage (also known as persistent storage, global storage)
Node 1
Process P
Local storage
Node *N*
Remote storage

FIGURE 2.14

Three types of storage in a swingle file hierarchy (Solid lines show what process P can access and the dashed line shows what P may be able to access.

Prof. Paul Lin

33

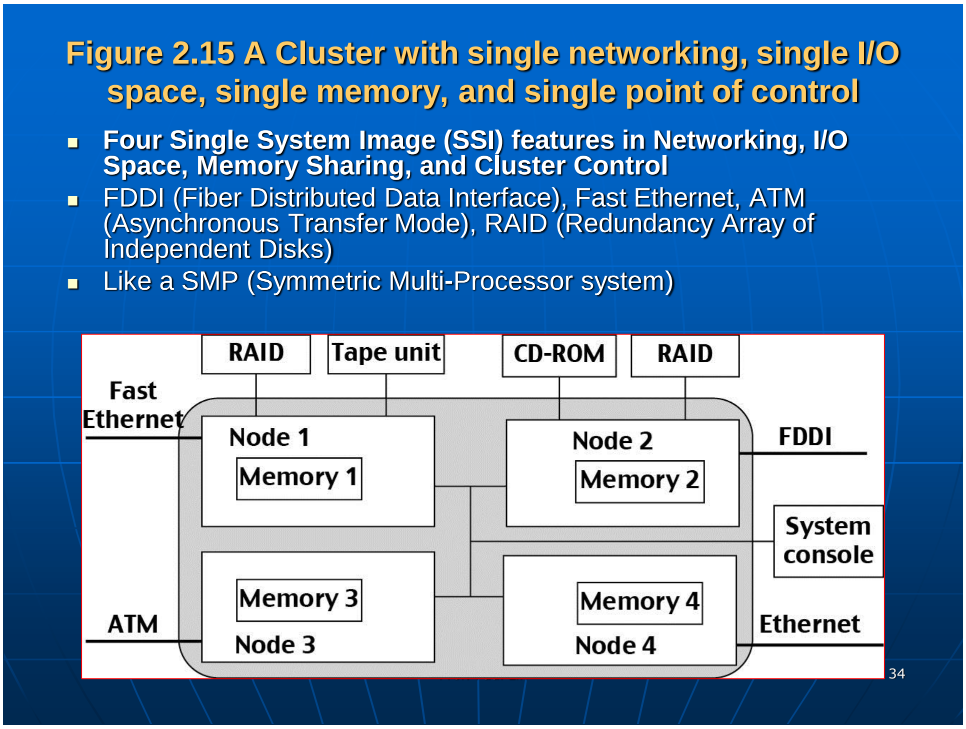
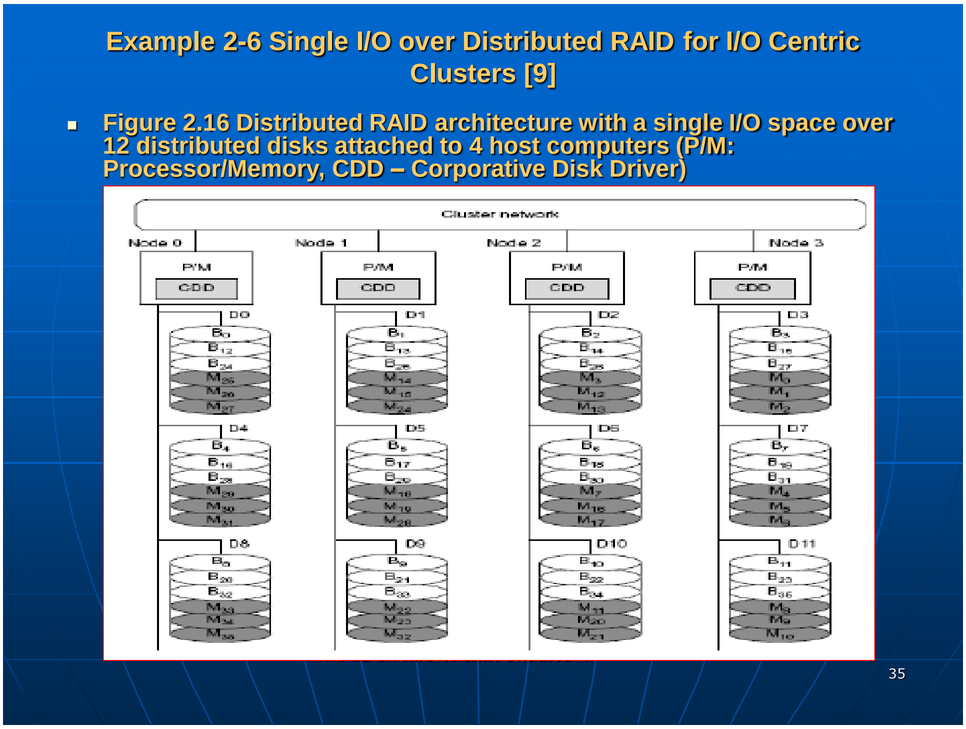


Figure 2.15 A Cluster with single networking, single I/O space, single memory, and single point of control

- **Four Single System Image (SSI) features in Networking, I/O Space, Memory Sharing, and Cluster Control**
- FDDI (Fiber Distributed Data Interface), Fast Ethernet, ATM (Asynchronous Transfer Mode), RAID (Redundancy Array of Independent Disks)
- Like a SMP (Symmetric Multi-Processor system)

34



Example 2-6 Single I/O over Distributed RAID for I/O Centric Clusters [9]

- Figure 2.16 Distributed RAID architecture with a single I/O space over 12 distributed disks attached to 4 host computers (P/M: Processor/Memory, CDD – Corporative Disk Driver)

Cluster network
Node 0
Node 1
Node 2
Node 3
P/M
CDD
P/M
CDD
P/M
CDD
P/M
CDD
D0
B_0 B_{12} B_{24} M_{25} M_{26} M_{27}
D1
B_1 B_{13} B_{25} M_{14} M_{15} M_{24}
D2
B_2 B_{14} B_{26} M_3 M_{12} M_{13}
D3
B_3 B_{15} B_{27} M_0 M_1 M_2
D4
B_4 B_{16} B_{28} M_{29} M_{30} M_{31}
D5
B_5 B_{17} B_{29} M_{18} M_{19} M_{28}
D6
B_6 B_{18} B_{30} M_7 M_{16} M_{17}
D7
B_7 B_{19} B_{31} M_4 M_5 M_6
D8
B_8 B_{20} B_{32} M_{33} M_{34} M_{35}
D9
B_9 B_{21} B_{33} M_{22} M_{23} M_{32}
D10
B_{10} B_{22} B_{34} M_{11} M_{20} M_{21}
D 11
B_{11} B_{23} B_{35} M_8 M_9 M_{10}

35

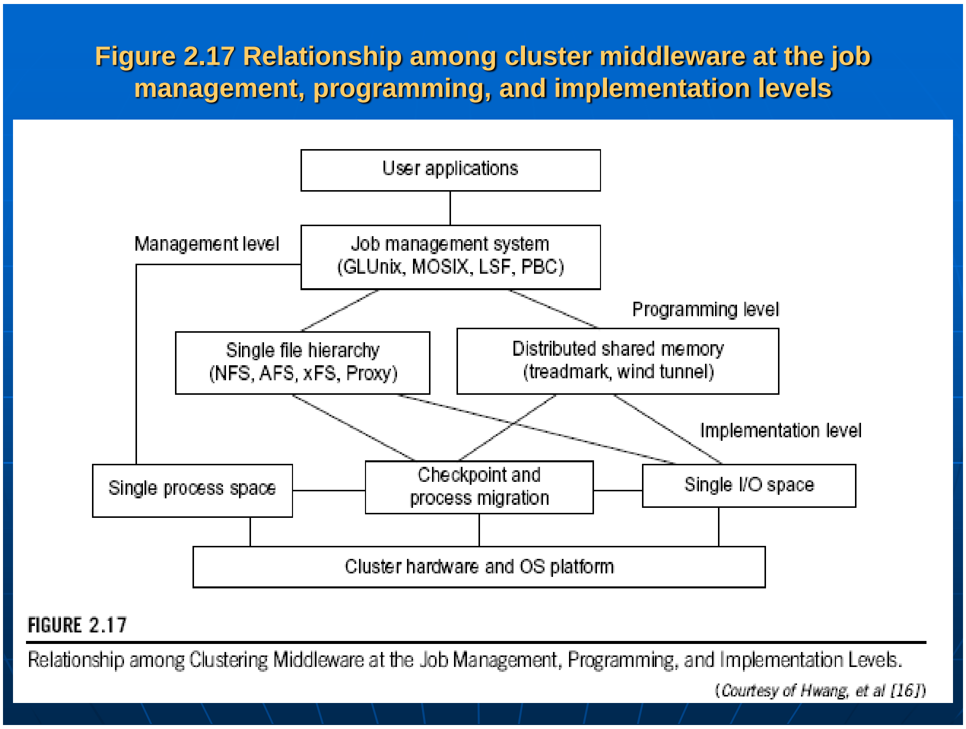


Figure 2.17 Relationship among cluster middleware at the job management, programming, and implementation levels

User applications

Management level

Job management system (GLUnix, MOSIX, LSF, PBC)

Programming level

Single file hierarchy (NFS, AFS, xFS, Proxy)

Distributed shared memory (treadmark, wind tunnel)

Implementation level

Single process space

Checkpoint and process migration

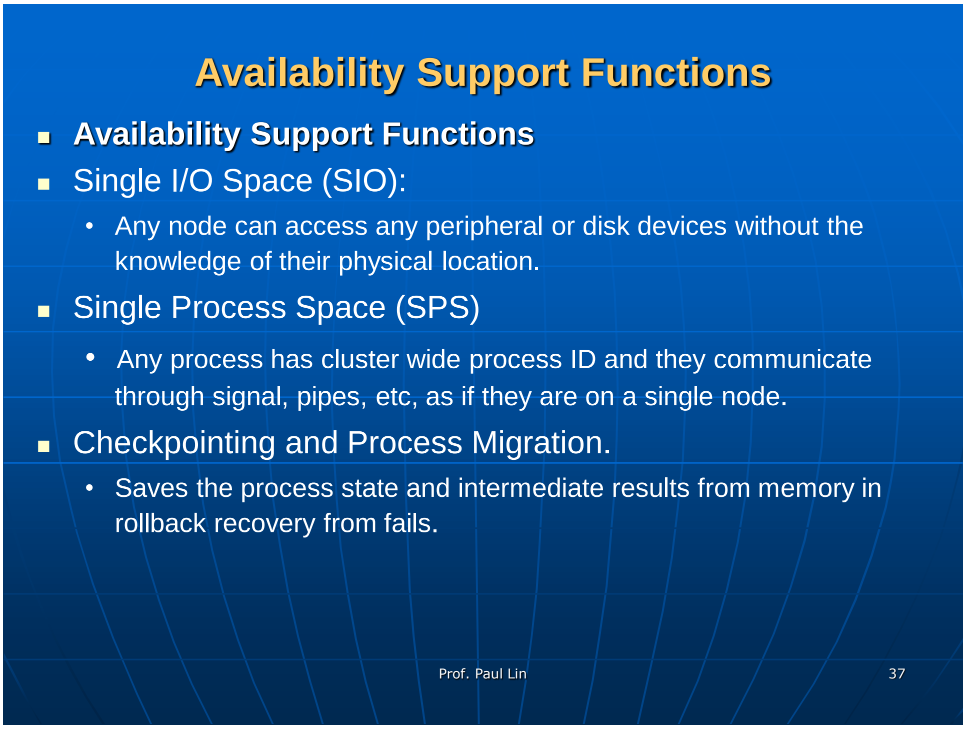
Single I/O space

Cluster hardware and OS platform

FIGURE 2.17

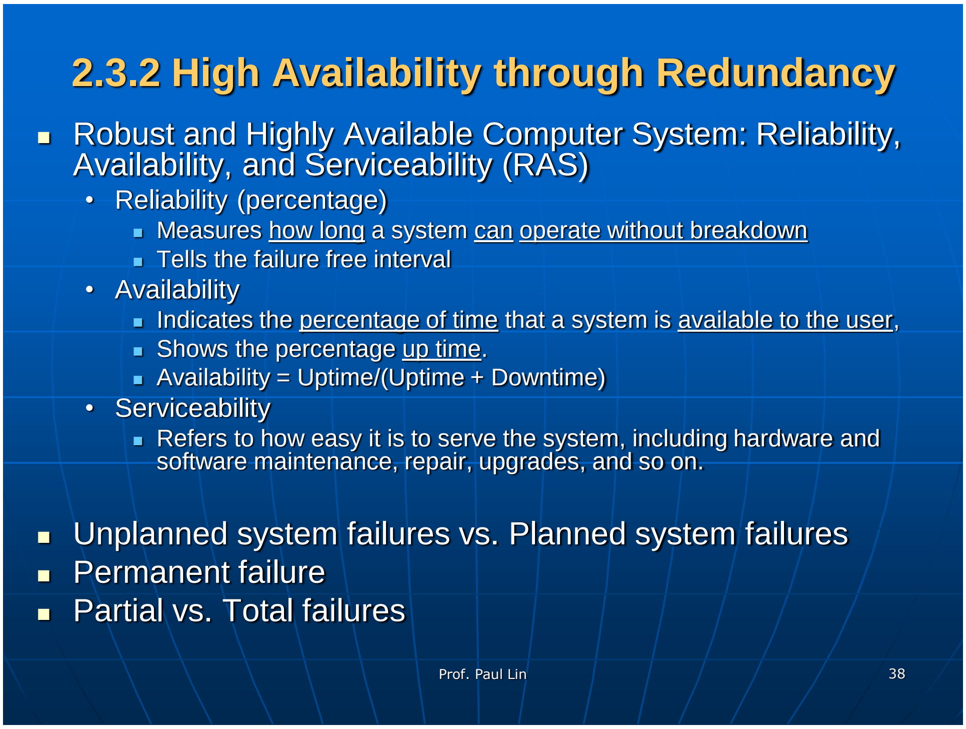
Relationship among Clustering Middleware at the Job Management, Programming, and Implementation Levels.

(*Courtesy of Hwang, et al [16]*)



Availability Support Functions

- **Availability Support Functions**
- Single I/O Space (SIO):
  - Any node can access any peripheral or disk devices without the knowledge of their physical location.
- Single Process Space (SPS)
  - Any process has cluster wide process ID and they communicate through signal, pipes, etc, as if they are on a single node.
- Checkpointing and Process Migration.
  - Saves the process state and intermediate results from memory in rollback recovery from fails.



2.3.2 High Availability through Redundancy

- Robust and Highly Available Computer System: Reliability, Availability, and Serviceability (RAS)
  - Reliability (percentage)
    - Measures how long a system can operate without breakdown
    - Tells the failure free interval
  - Availability
    - Indicates the percentage of time that a system is available to the user,
    - Shows the percentage up time.
    - Availability = Uptime/(Uptime + Downtime)
  - Serviceability
    - Refers to how easy it is to serve the system, including hardware and software maintenance, repair, upgrades, and so on.

- Unplanned system failures vs. Planned system failures
- Permanent failure
- Partial vs. Total failures

System Availability

- From a cluster system design perspective, the Availability can be expressed as shown in Figure 2.19.
- Availability ↑, if MTTF ↑ or MTTR ↓
- Increasing MTTF means increasing the reliability of the system

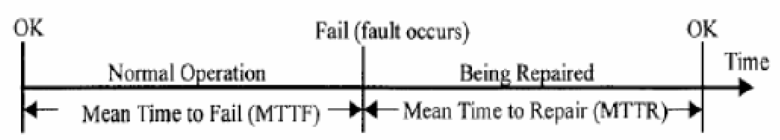


Figure 2.19 The operate-repair cycle of a computer system.

$$Availability = \mathrm{MTTF}/(\mathrm{MTTF} + \mathrm{MTTR})$$

Computer Systems - Availability

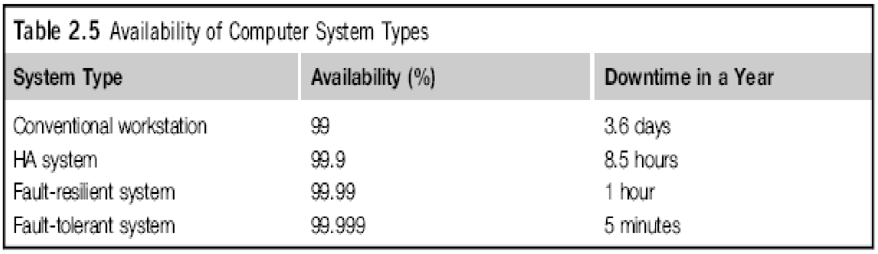
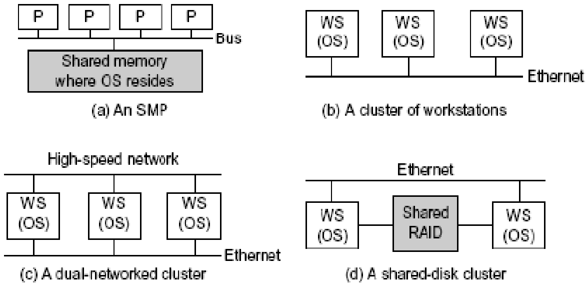


Table 2.5 Availability of Computer System Types

System Type	Availability (%)	Downtime in a Year
Conventional workstation	99	3.6 days
HA system	99.9	8.5 hours
Fault-resilient system	99.99	1 hour
Fault-tolerant system	99.999	5 minutes

Example 2.7 Single Points of Failure in a SMP and in Clusters of Computers

- Configuration (a): Single points of failure: the Shared memory, the OS image, the Memory bus
- Configuration (b): single point of failure is Ethernet
- Configuration (c): Two paths of communication takes care problem (b)
- Configuration (d): the data will not be lost



(a) An SMP

(b) A cluster of workstations

(c) A dual-networked cluster

(d) A shared-disk cluster

FIGURE 2.19

Single points of failure (SPF) in an SMP and in three clusters, where greater redundancy eliminates more SPFs in systems from (a) to (d).

(Courtesy of Hwang and Xu [14])

Parallel Reliability Model

- Analysis of the reliability problem
  - A parallel component/system reliability
  - N identical and independent component/system, with reliability (available time) R
  - The overall system reliability Rs = 1 – (1 – R)^N

Table 2.6 Job Scheduling Issues and Schemes for Cluster Nodes

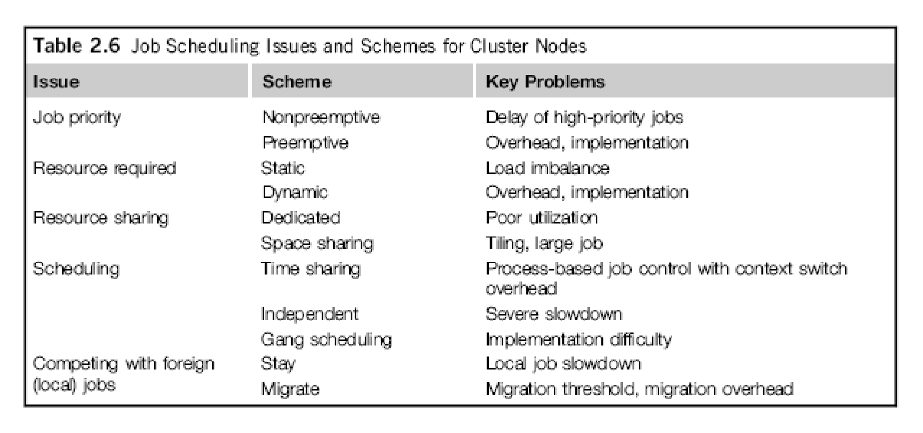
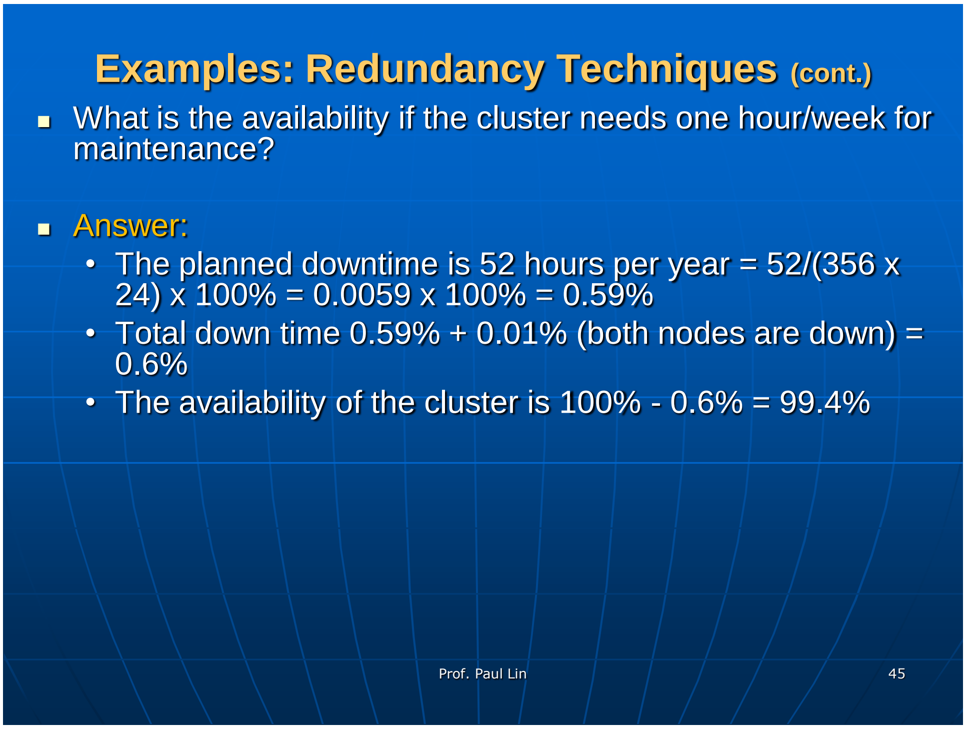


Table 2.6 Job Scheduling Issues and Schemes for Cluster Nodes

Issue	Scheme	Key Problems
Job priority	Nonpreemptive	Delay of high-priority jobs
	Preemptive	Overhead, implementation
Resource required	Static	Load imbalance
	Dynamic	Overhead, implementation
Resource sharing	Dedicated	Poor utilization
	Space sharing	Tiling, large job
Scheduling	Time sharing	Process-based job control with context switch overhead
	Independent	Severe slowdown
	Gang scheduling	Implementation difficulty
Competing with foreign (local) jobs	Stay	Local job slowdown
	Migrate	Migration threshold, migration overhead

Examples: Redundancy Techniques (Parallel Reliability Model)

- Consider the Figure 2-19 (d) A Shared Disk Configuration with 2 nodes
- Assume that
  - (1) **Only "the nodes" can fail**, and the rest of the system interconnect and shared RAID disk is 100% available
  - (2) When a node fail, its workload is **switched over** to the other node in "**Zero time**"
- What is the availability of the cluster if planned downtime is ignored?
  - From Table 2.5, a conventional workstation is available 99% of time.
  - A = MTTF/(MTTF + MTTR)
  - The time both nodes are down is only 0.01%
  - Thus, the availability is 99.99% = 100% – 0.01%
  - This is a Fault-resilient system with only 1 hour of downtime per year



Examples: Redundancy Techniques (cont.)

- What is the availability if the cluster needs one hour/week for maintenance?

- Answer:
  - The planned downtime is 52 hours per year = 52/(356 x 24) x 100% = 0.0059 x 100% = 0.59%
  - Total down time 0.59% + 0.01% (both nodes are down) = 0.6%
  - The availability of the cluster is 100% - 0.6% = 99.4%