Distributed and Cloud Computing

K. Hwang, G. Fox and J. Dongarra

Chapter 2: Computer Clusters for Scalable parallel Computing

Adapted from Kai Hwang
University of Southern California
March 30, 2012

What is a computing cluster?

- A computing cluster consists of a collection of interconnected stand-alone/complete computers, which can cooperatively working together as a single, integrated computing resource. Cluster explores parallelism at job level and distributed computing with higher availability.
- A typical cluster:
  - Merging multiple system images to a SSI (single-system image) at certain functional levels.
  - Low latency communication protocols applied
  - Loosely coupled than an SMP with a SSI

Multicomputer Clusters:

- *Cluster: A network of computers supported by middleware and interacting by message passing*
- *PC Cluster (Most Linux clusters)*
- *Workstation Cluster*
  - *COW – Cluster of Workstations*
  - *NOW – Network of Workstations*
- *Server cluster or Server Farm*
- *Cluster of SMPs or ccNUMA* (cache coherent Non-Uniform Memory Architecture) *systems*
- *Cluster-structured massively parallel processors*
  - *(MPP) – about 85% of the top-500 systems*

Multi-Computer Cluster Components

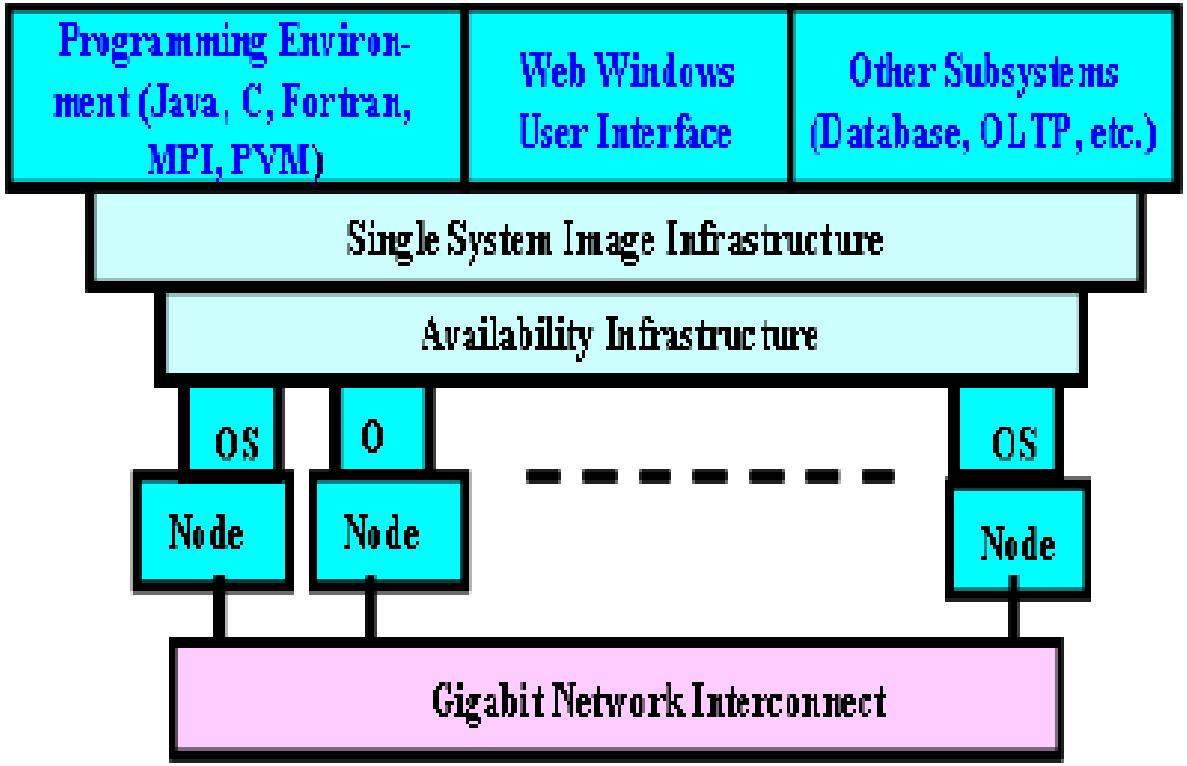


Figure 1.16 The architecture of a working cluster with full hardware, software, and middleware support for availability and single system image.

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

Attributes Used in Cluster Classification

Attributes	Attribute Value	
Packaging	Compact	Slack
Control	Centralized	Decentralized
Homogeneity	Homogeneous	Heterogeneous
Security	Enclosed	Exposed
Example	Dedicated cluster	Enterprise cluster

Cluster Classification

- Scalability : Adding servers to a cluster or adding more clusters to a network as the application need arises.
- Packaging : Compact / Slack
  - Compact – packaged in racks in machine room
  - Slack – PC, workstations geographically distributed
- Control : Centralized / De-centralized.
- Homogenaity: Same vs. different platforms (e.g. CPUs, OSs)
- Programmability : Ability to run a variety of applications.
- Security : Exposure of intra-cluster communication.

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

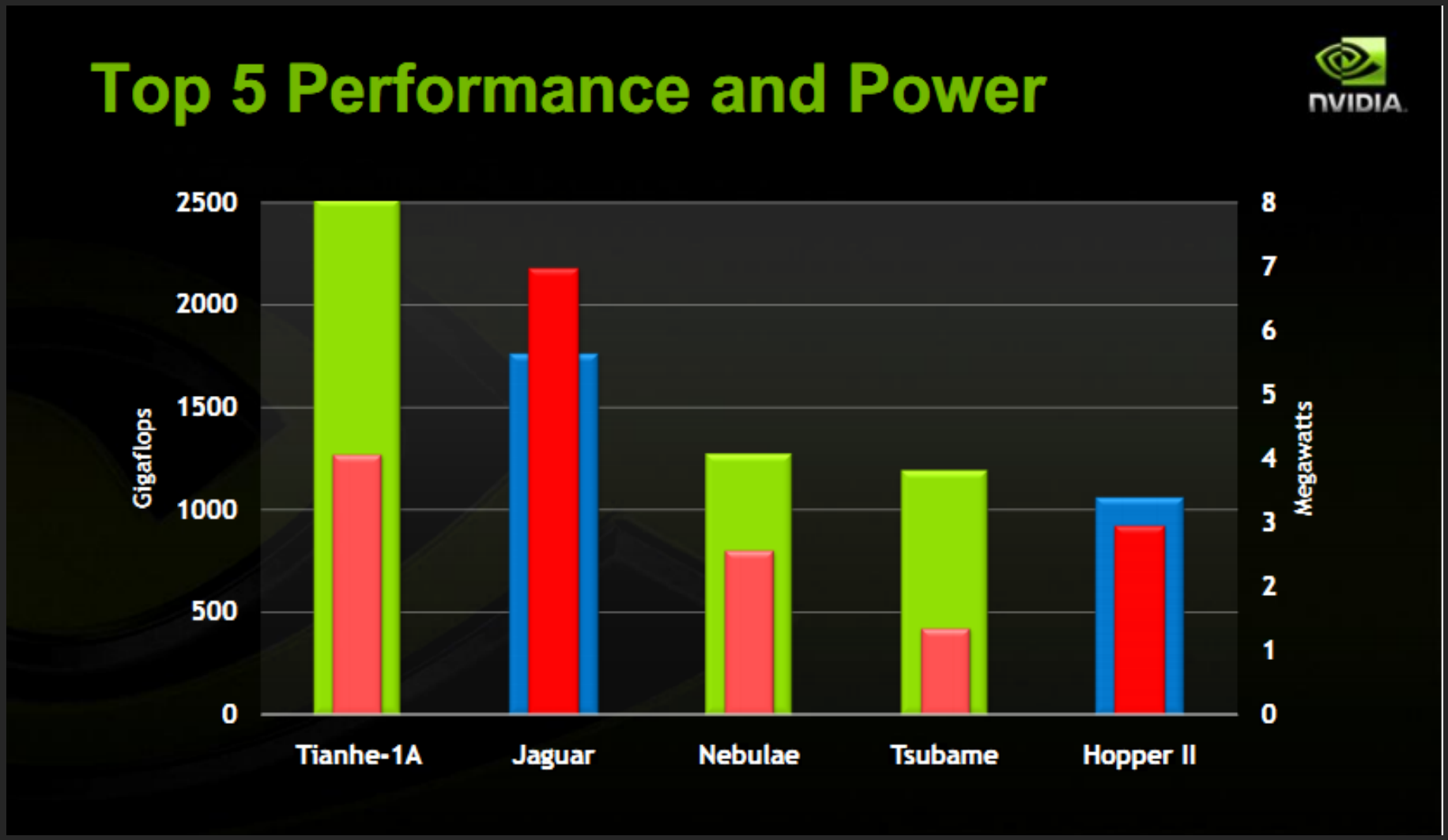
Operational Benefits of Clustering

- High availability (HA) : Cluster offers inherent high system availability due to the redundancy of hardware, operating systems, and applications.
- Hardware Fault Tolerance: Cluster has some degree of redundancy in most system components including both hardware and software modules.
- OS and application reliability : Run multiple copies of the OS and applications, and through this redundancy
- Scalability : Adding servers to a cluster or adding more clusters to a network as the application need arises.
- High Performance : Running cluster enabled programs to yield higher throughput.

Top- 500 Release in June 2010

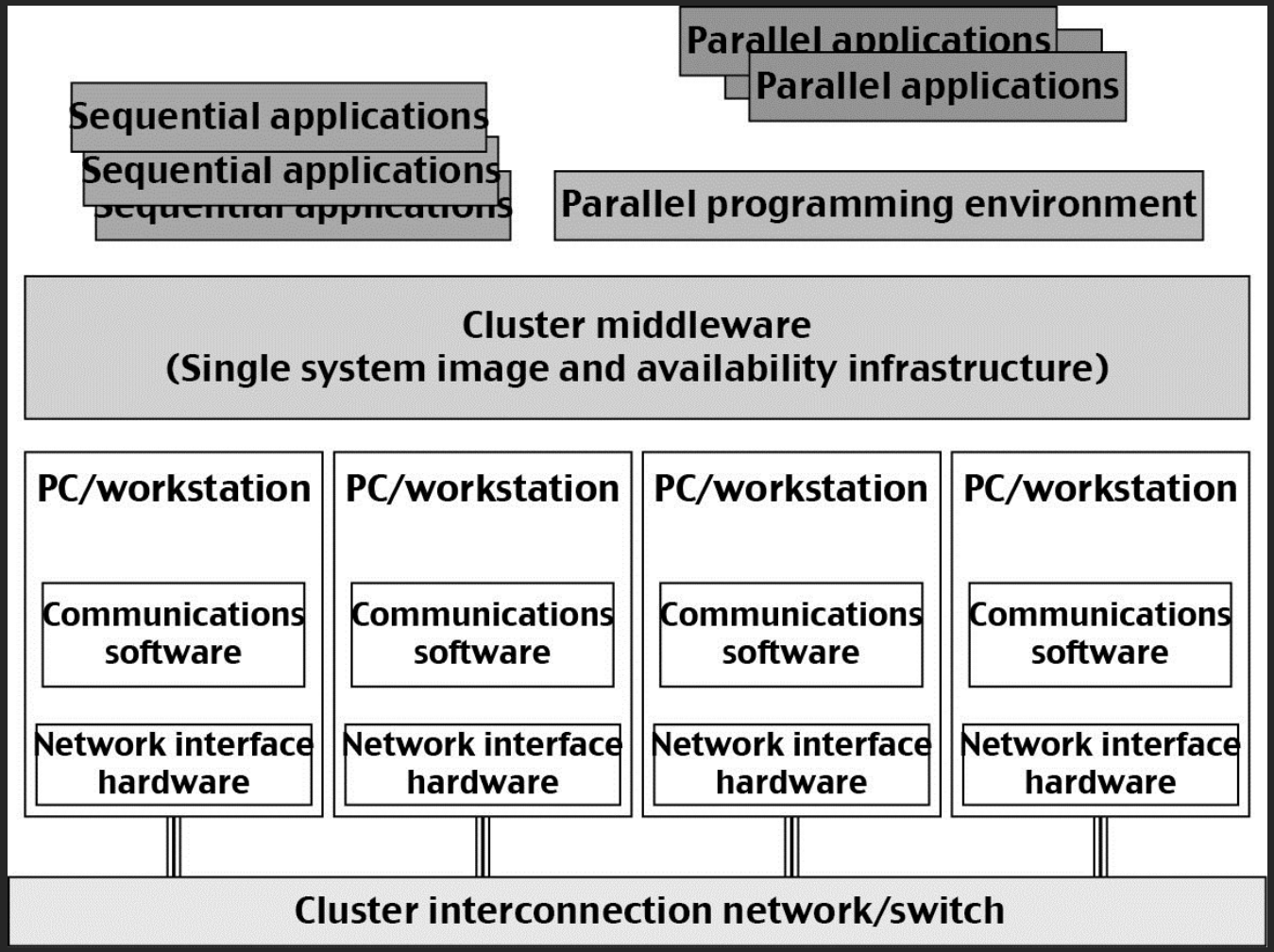
Table 2.3 Top Five Supercomputers Evaluated in Jne 2010

System Rank and Name	Architecture Description (Core size, processor, GHz, OS, and interconnection topology)	Sustained Speed	Power/ system
1. Jaguar at Oak Ridge Nat'l Lab in USA	Cray XT5-HE: An MPP built with 224,162 cores in 2.6 GHz Opteron 6-core processors, interconnected by a 3-D torus	1.759 PFlops	6.95MW
2. Nebulae at China's Nat'l Supercomputer Centre, (NSCS)	Dawning TC3600 Blade System: Built with 120,640 cores in 2.66 GHz Intel EM64T Xeon X5650 and NVidia GPU, running Linux and interconnected by an Infiniband QDR network	1.271 PFlops	2.55MW
3. Roadrunner at DOE/NNSA/LANL in USA	IBM BladeCenter QS22/LS21 cluster of 122,400 cores in 12,960 3.2GHz POWER XCell 8i and 6,480 AMD 1.8GHz Opteron processors, running Linux with Infiniband network	1.042 PFlops	2.35MW
4. Kraken XT5 at NICS, Univ of Tennessee	Cray XT5-HE: An MPP built with 98928 cores of 2.6 GHz Opteron 6-core processors interconnected by a 3-D torus	831.7 TFlops	3.09MW
5. JUGENE at the FZJ Germany	IBM BlueGene/P solution built with 294,912 processors: PowerPC core, 4-way SMP nodes, and 144TB of memory in 72 racks, interconnected by a 3-D torus network	825.5 TFlops	2.27MW



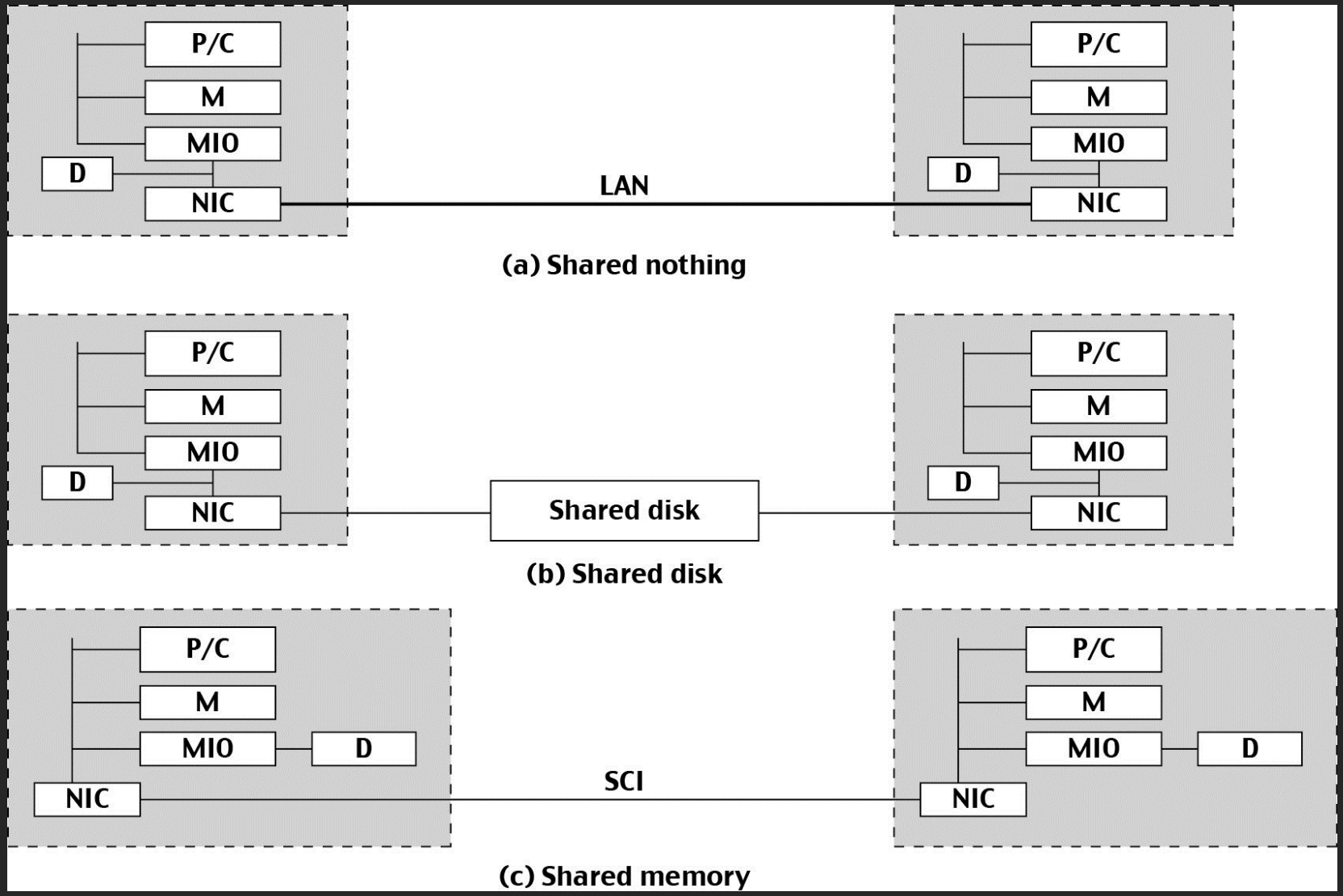
(Courtesy of Bill Dally, 2011)

Basic Cluster Architecture



Computer Cluster built from commodity hardware, software, middleware, and network supporting HA and SSI

Resource Sharing in Cluster of Computers



Three ways to connect cluster nodes.

Compute Node Architectures :

Table 2.4 Sample Compute Node Architectures for Clustered Systems built in 2010

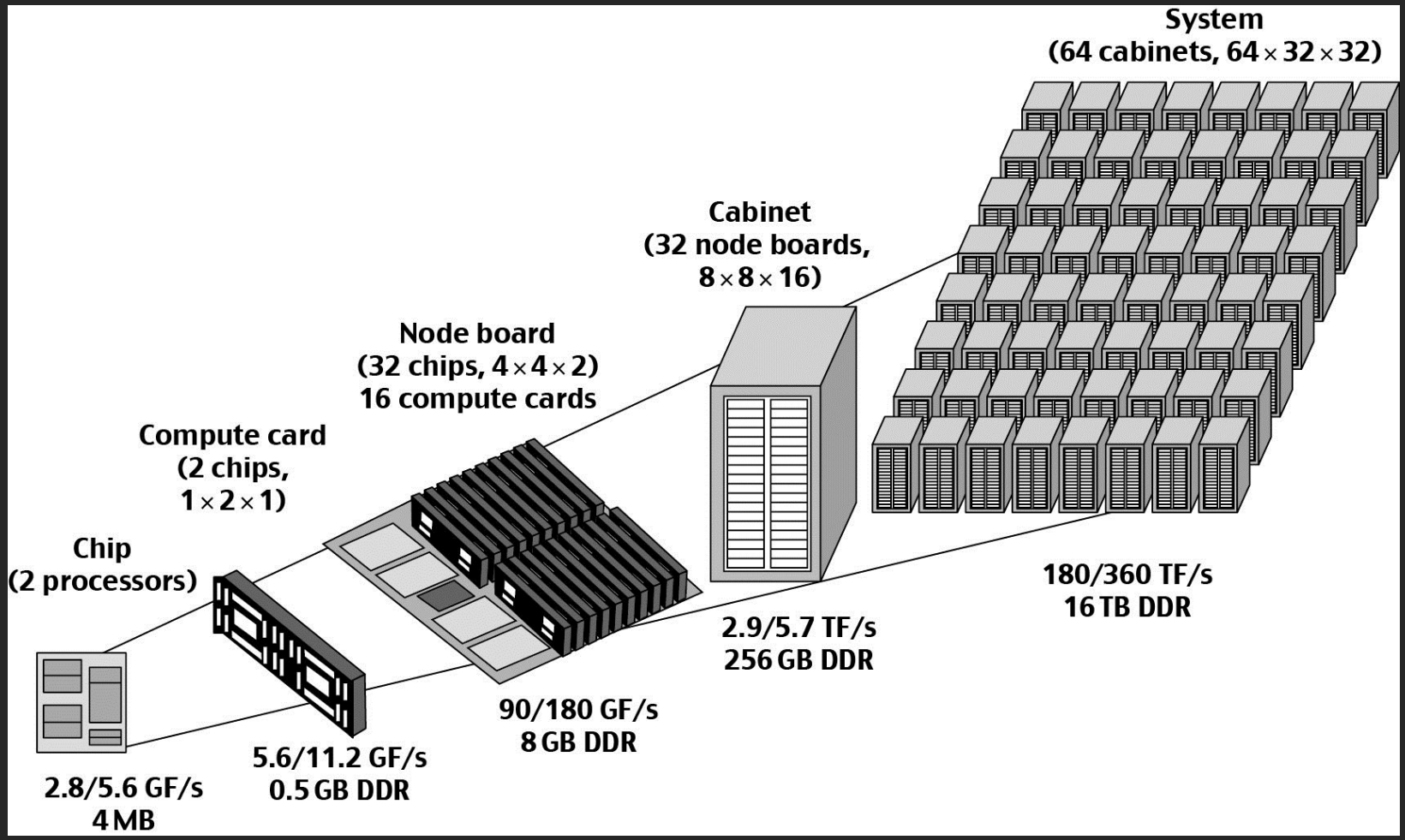
Node Architecture	Major Characteristics	Representative Systems
Homogeneous Node using the same Multicore Processors	One or more multicore processors mounted on the same node with crossbar connected to shared memory or local disks	Cray XT-5 uses two 6-core AMD Opteron Processors in each compute node
Hybrid Nodes using CPU plus GPU or FLP Accelerators	General-purpose CPU for integer operations while GPUs acting as coprocessors to speedup FLP operations	China's Tianhe System using 2 Intel Xeon processors plus 2 AMD GPUs per node

Overview of Blue Gene L

- Blue Gene L is a supercomputer jointly developed by IBM and Lawrence Livermore National Laboratory
- It occupies 17 of the top 100 slots in the rankings at top500.org, including 5 of the top 10
  - 360 TeraFLOPS theoretical peak speed
- Largest configuration:
  - At Lawrence Livermore Nat'l Lab.
  - Runs simulations on US nuclear weapon stockpile
  - 64 physical racks
  - 65,536 compute nodes
  - Torus interconnection network of 64 x 32 x 32

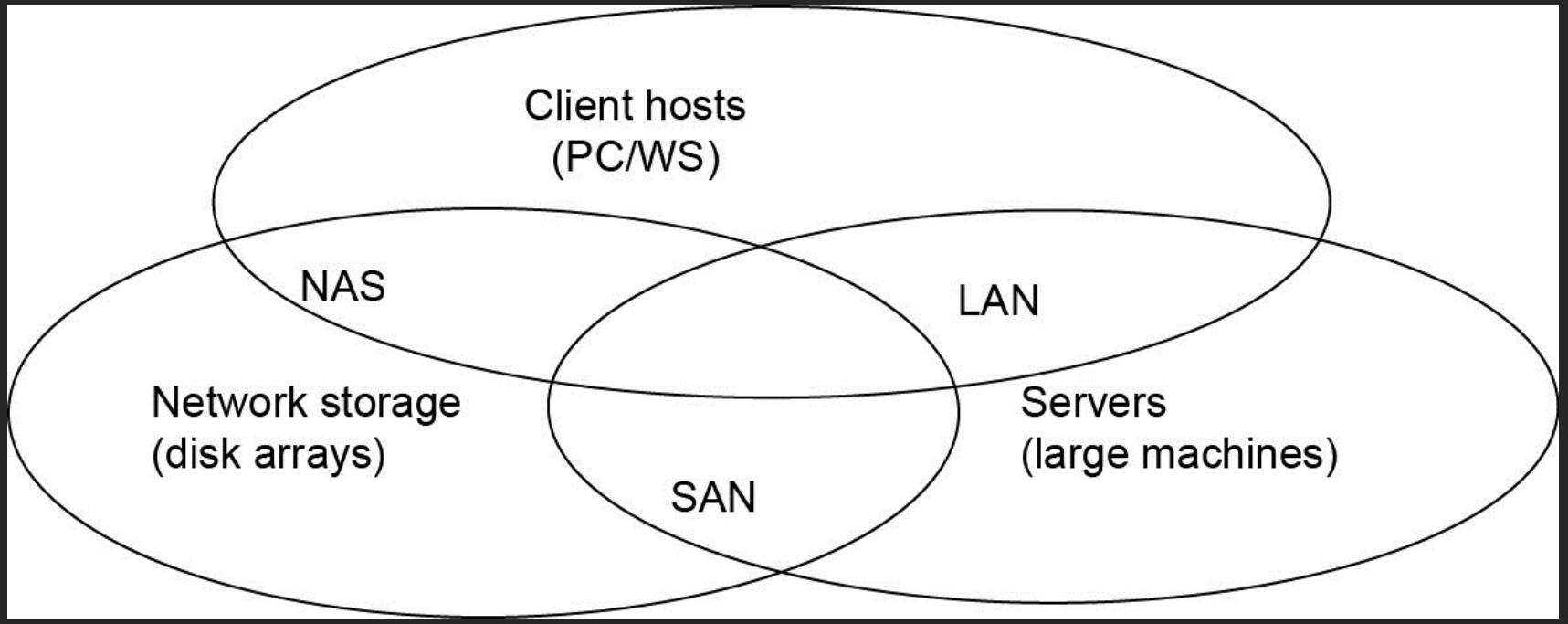
IBM BlueGene/L Supercomputer: The World Fastest

Message-Passing MPP built in 2005



Built jointly by IBM and LLNL teams and funded by US DoE ASCI Research Program

Standard Cluster Interconnects



- SAN (storage area network) - connects servers with disk arrays
- LAN (local area network) – connects clients, hosts, and servers
- NAS (network attached storage) – connects clients with large storage systems

High Bandwidth Interconnects

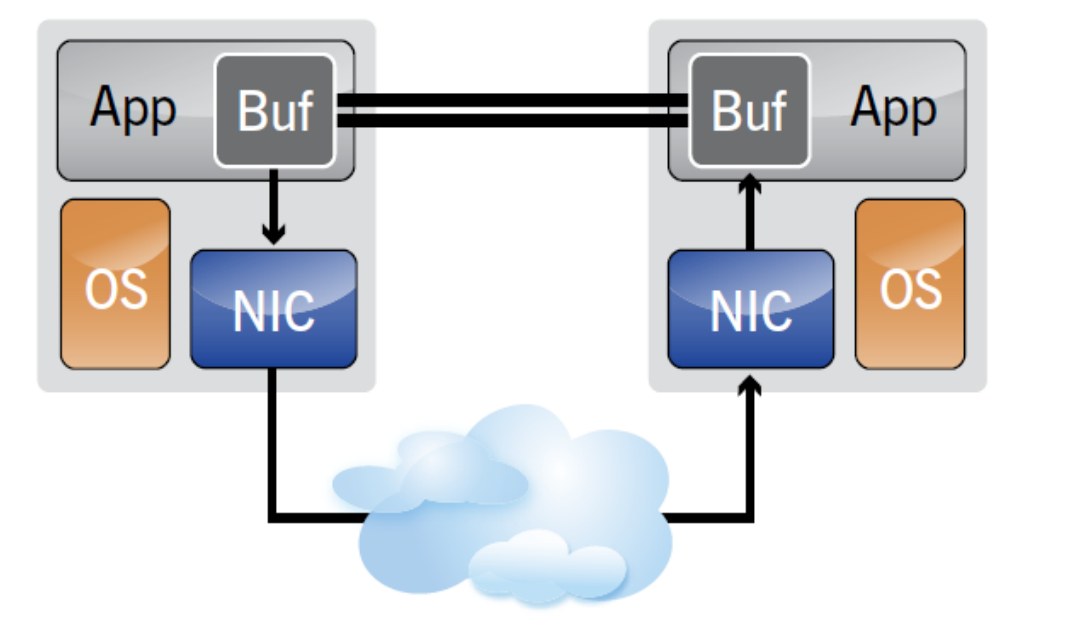
Table 2.5 Comparison of Four Cluster Interconnect Technologies by 2007

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 : InfiniBand (1)

- Provides applications with an easy-to-use messaging service.
- Gives every application direct access to the messaging service without need not rely on the operating system to transfer messages.
- Provides a messaging service by creating a channel connecting an application to any other application or service with which the application needs to communicate.
- Create these channels between virtual address spaces.

Example : InfiniBand (2)

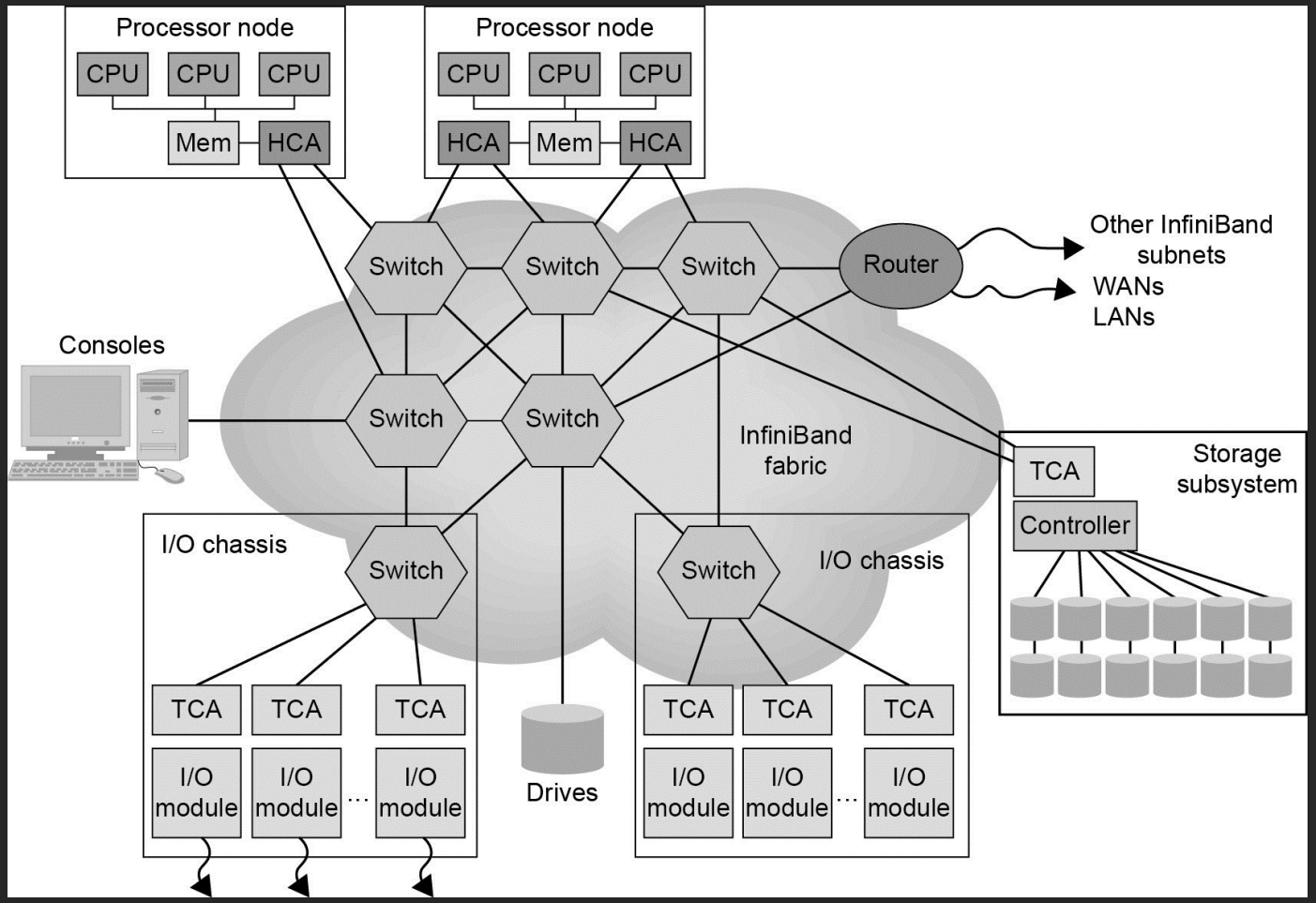


- *InfiniBand creates a channel directly connecting an application in its virtual address space to an application in another virtual address space.*
- *The two applications can be in disjoint physical address spaces – hosted by different servers.*

InfiniBand Architecture

- HCA – Host Channel Adapter. An HCA is the point at which an InfiniBand end node, such as a server or storage device, connects to the InfiniBand network.
- TCA – Target Channel Adapter. This is a specialized version of a channel adapter intended for use in an embedded environment such as a storage appliance.
- Switches – An InfiniBand Architecture switch is conceptually similar to any other standard networking switch, but molded to meet InfiniBand's performance and cost targets.
- Routers – Although not currently in wide deployment, an InfiniBand router is intended to be used to segment a very large network into smaller subnets connected together by an InfiniBand router.

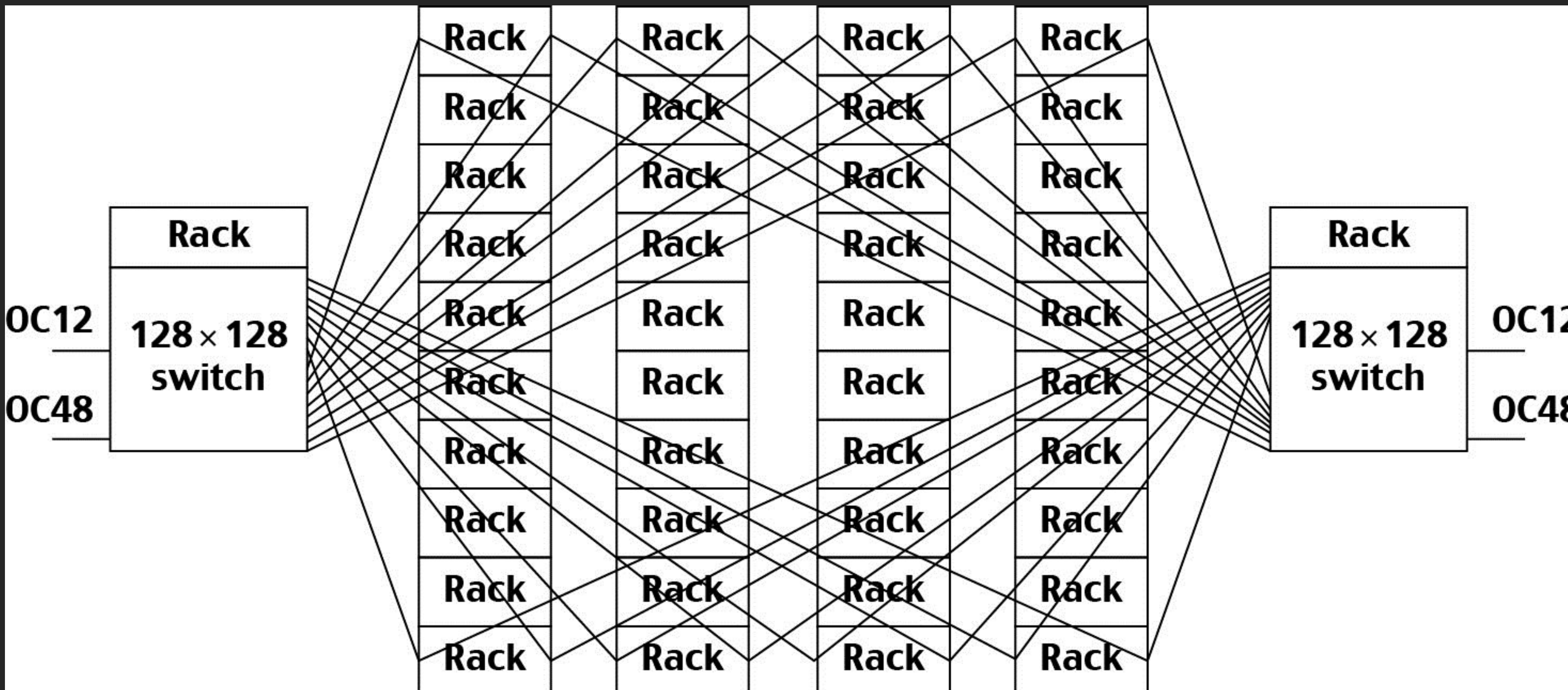
Example: InfiniBand System Fabric

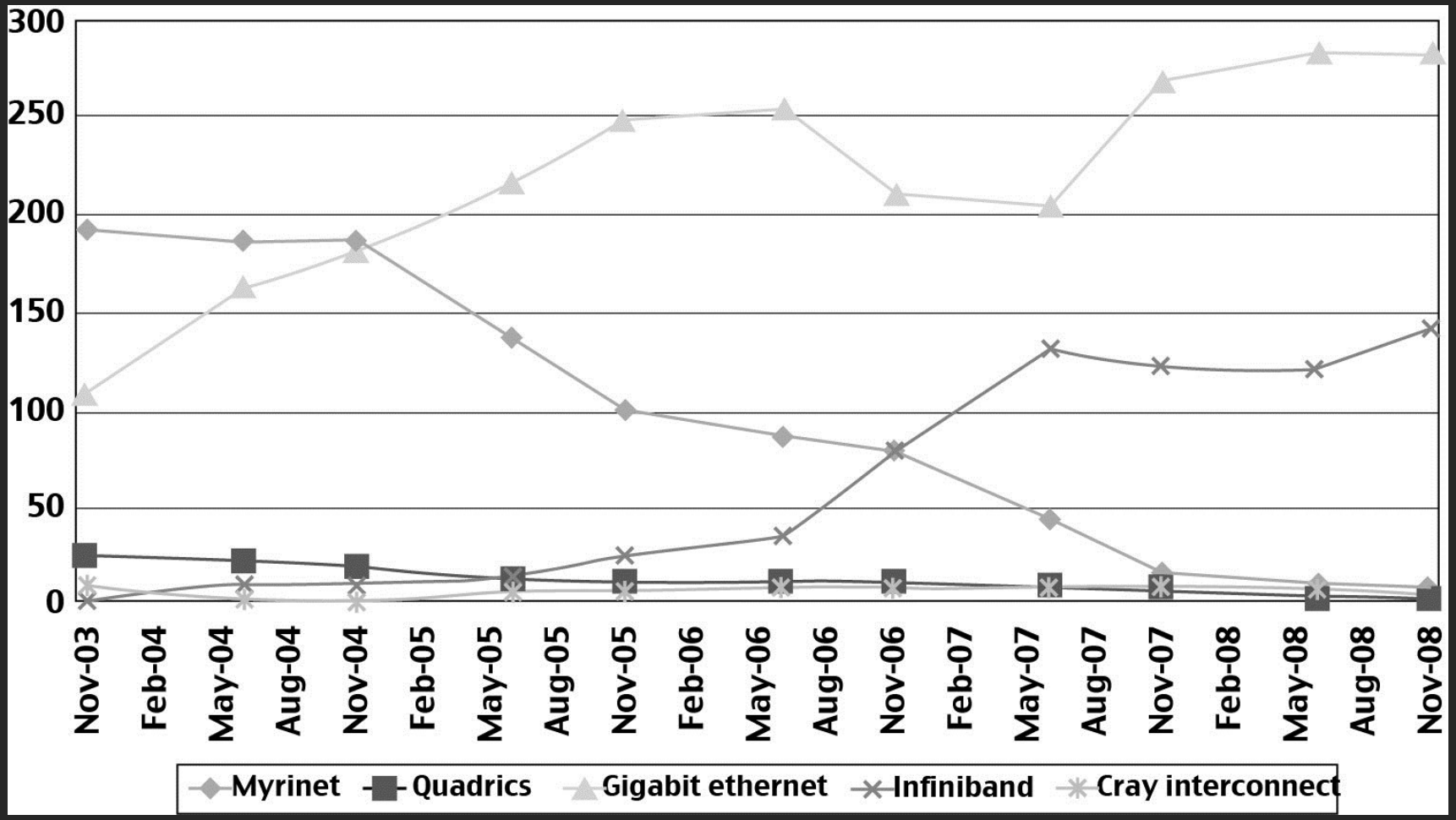


Example: The Big Google Search Engine

- A Supercluster built over high-speed PCs and Gigabit LANs for global web page searching applications provided by Google.
- Physically, the cluster is housed in 40 PC/switch racks with 80 PCs per rack and 3200 PCs in total
- Two racks to house two 128 x 128 Gigabit Ethernet switches, the front hosts, and UPSs, etc.
- All commercially available hardware parts with Google designed software systems for supporting parallel search, URL linking, page ranking, file and database management, etc.

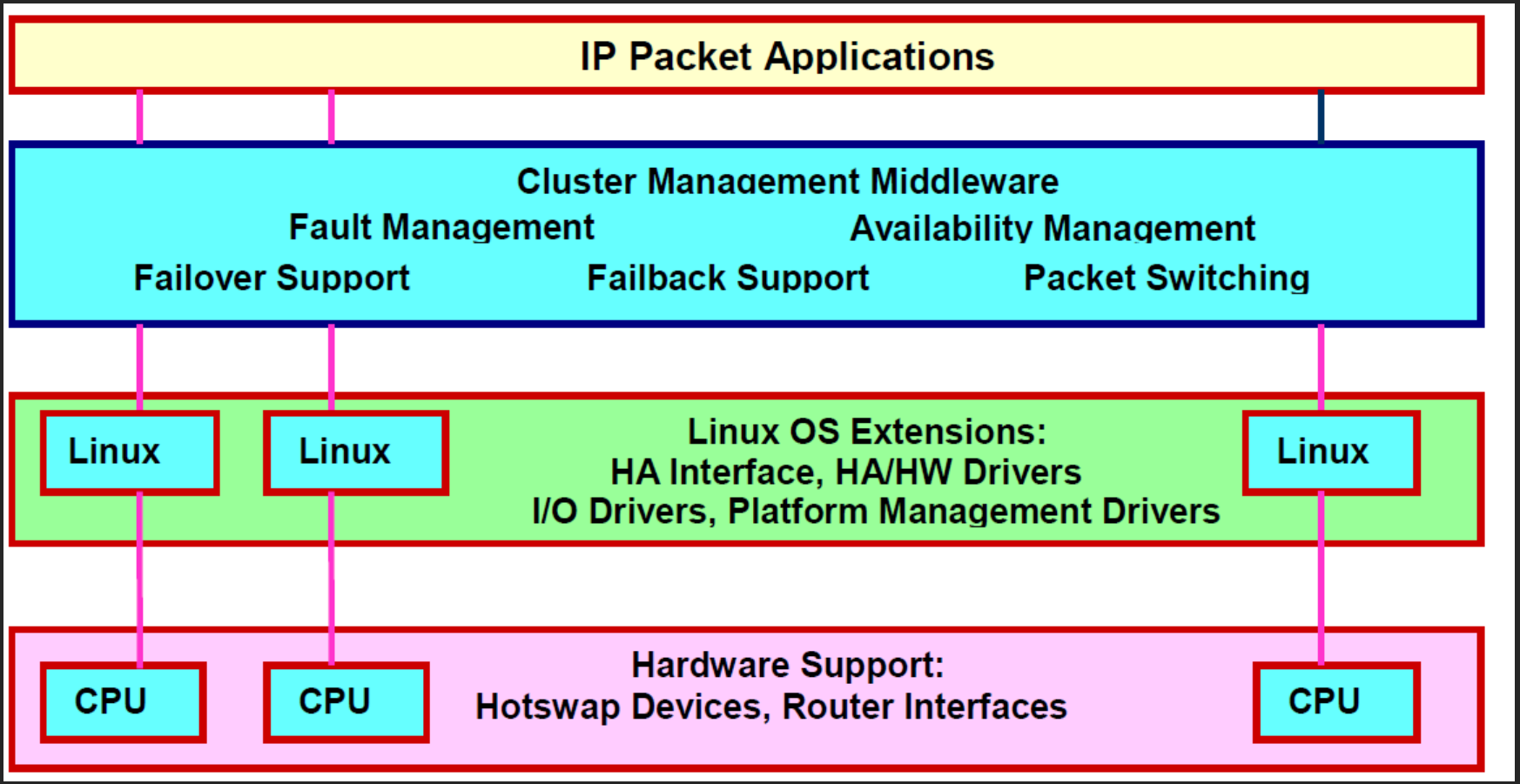
Google Search Engine Cluster





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

Hardware, Software, and Middleware Support



Middleware, Linux extensions, and hardware support for achieving high-availability in a cluster system

Design Principles of Clusters

- Single-system image (SSI)
- High availability (HA)
- Fault tolerance
- Rollback recovery

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)

Single-System-Image Features

- Single System: The entire cluster is viewed by the users as one system, which has multiple processors.
- Single Control: Logically, an end user or system user utilizes services from one place with a single interface.
- Symmetry: A user can use a cluster service from any node. All cluster services and functionalities are symmetric to all nodes and all users, except those protected by access rights.
- Location Transparent: The user is not aware of the whereabouts of the physical device that eventually provides a service.

Basic SSI Services

A. *Single Entry Point*
   - telnet cluster.usc.edu
   - telnet node1.cluster.usc.edu

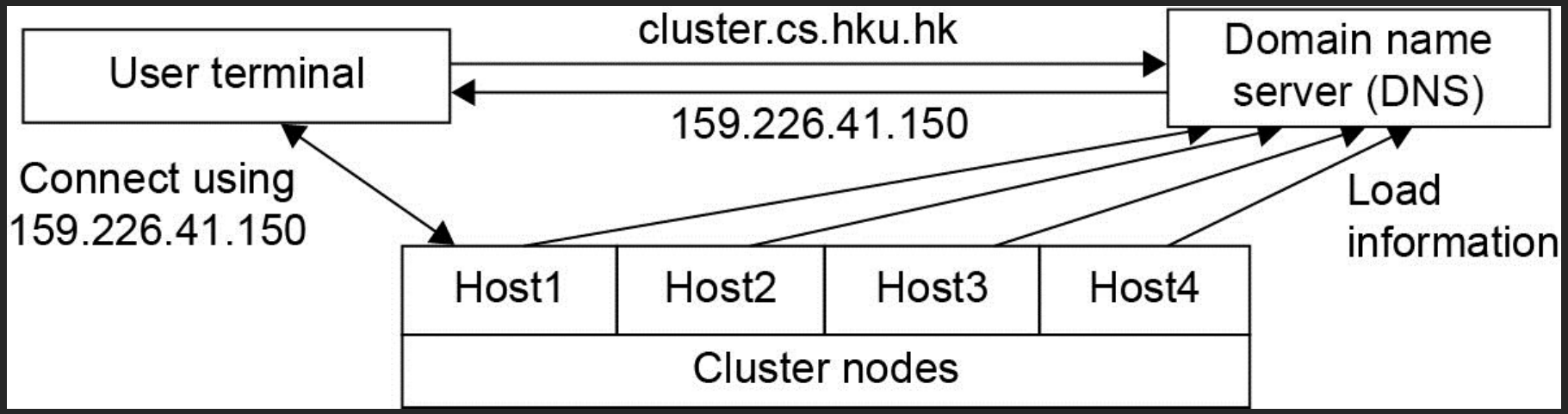
B. *Single File Hierarchy*: xFS, AFS, Solaris MC Proxy

C. *Single I/O, Networking, and Memory Space*

Other

- Single Job Management: GlUnix, Codine, LSF, etc.
- Single User Interface: Like CDE in Solaris/NT
- Single process space

Example: Realizing a single entry point in a cluster of computers



1. Four nodes of a cluster are used as host nodes to receive users' login requests.
2. To log into the cluster a standard Unix command such as "telnet cluster.cs.hku.hk", using the symbolic name of the cluster system is issued.
3. The symbolic name is translated by the DNS, which returns with the IP address 159.226.41.150 of the least-loaded node, which happens to be node Host1.
4. The user then logs in using this IP address.
5. The DNS periodically receives load information from the host nodes to make load-balancing translation decisions.

B. Single File Hierarchy

Single file hierarchy - the illusion of a single, huge file system image that transparently integrates local and global disks and other file devices (e.g., tapes).

Files can reside on 3 types of locations in a cluster:

- *Local storage* - disk on the local node.
- *Remote storage* - disks on remote nodes.
- *Stable storage* -
  - Persistent - data, once written to the stable storage, will stay there at least for a period of time (e.g., a week), even after the cluster shuts down.
  - Fault tolerant - to some degree, by using redundancy and periodical backup to tapes.

Example: Stable Storage

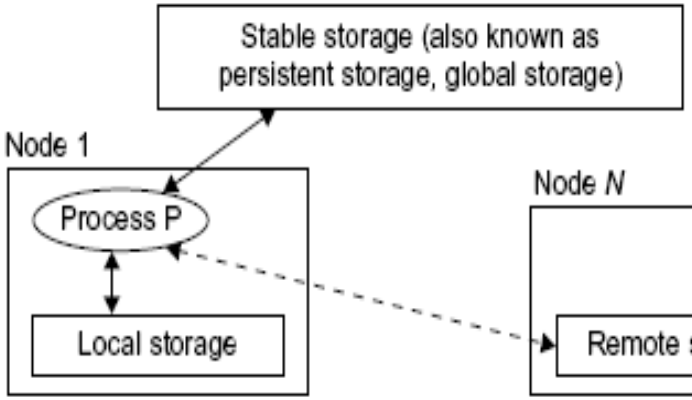


FIGURE 2.14

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.

Could be implemented as one centralized, large RAID disk or distributed using local disks of cluster nodes.

- First approach uses a large disk, which is a single point of failure and a potential performance bottleneck.
- Second approach is more difficult to implement, but potentially more economical, more efficient, and more available.

C. Single I/O, Networking, and Memory Space

To achieve SSI, we need a:

- single control point
- single address space
- single job management system
- single user interface
- single process control

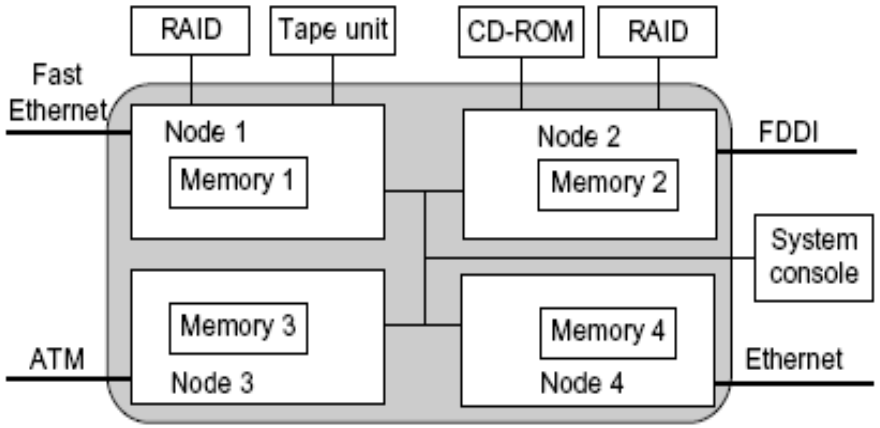


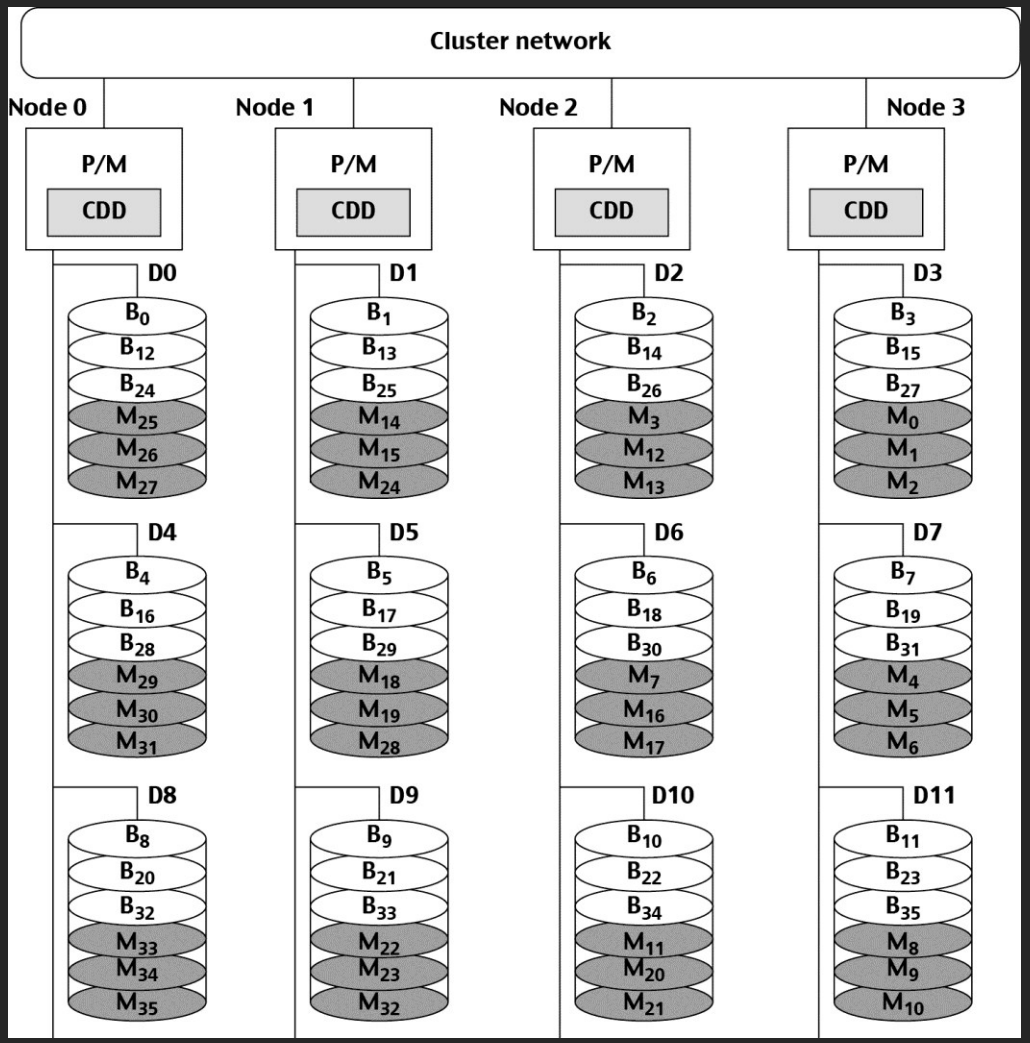
FIGURE 2.15

A cluster with single networking, single I/O space, single memory, and single point of control.

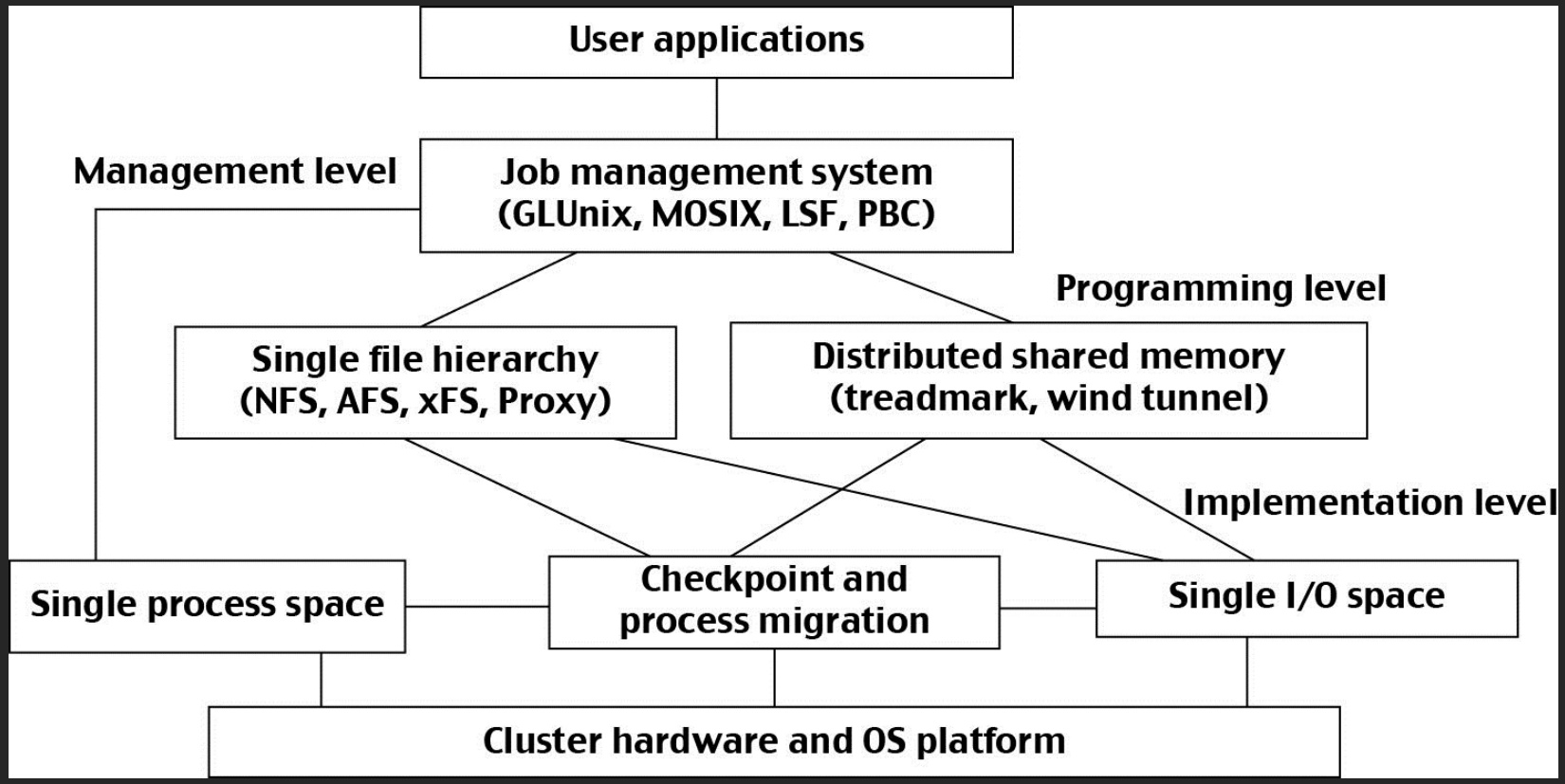
(Courtesy of Hwang and Xu [14])

Example: Distributed RAID - The RAID-x Architecture

- Distributed RAID architecture with a single I/O space over 12 distributed disks attached to 4 host machines (nodes) in a Linux cluster.
- Di stands for disk I
- Bj for disk block j
- Bj' an image (shaded plates) of block Bj.
- P/M refers to processor /memory node
- CDD is a cooperative disk driver.



Middleware Support for SSI Clustering:



Three levels of middleware: job management, programming and implementation

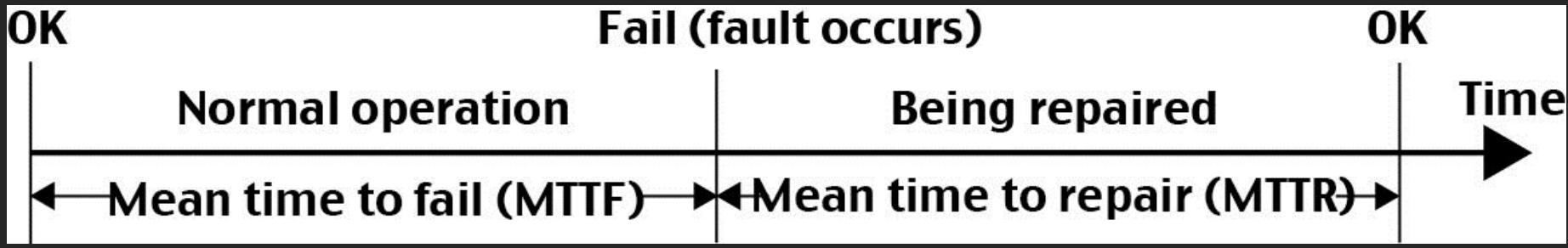
High Availability Through Redundancy

- Three terms often go together: reliability, availability, and serviceability (RAS).
- Availability combines the concepts of reliability and serviceability as defined below:
  - Reliability measures how long a system can operate without a breakdown.
  - Availability indicates the percentage of time that a system is available to the user, that is, the percentage of system uptime.
  - Serviceability refers to how easy it is to service the system, including hardware and software maintenance, repair, upgrade, etc.

Availability and Failure Rate

Recent Find/SVP Survey of Fortune 1000 companies:

- An average computer is down 9 times a year with an average downtime of 4 hours.
- The average loss of revenue per hour downtime is $82,500.



The operate-repair cycle of a computer system.

Availability = MTTF / (MTTF + MTTR)

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

Single Points of Failure in SMP and Clusters

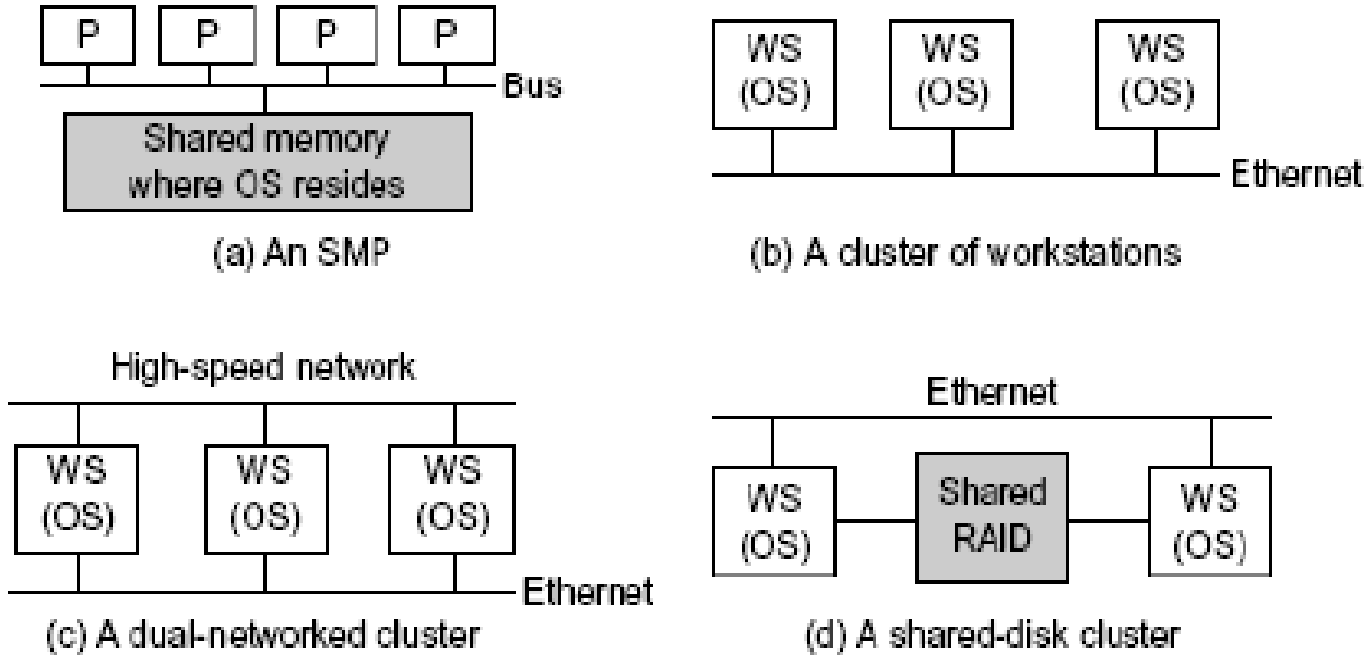


FIGURE 2.19

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])

Fault Tolerant Cluster Configurations

Redundant components configured based on different cost, availability, and performance requirements.

The following three configurations are frequently used:

- Hot Standby: A primary component provides service, while a redundant backup component stands by without doing any work, but is ready (hot) to take over as soon as the primary fails.
- Mutual Takeover: All components are primary in that they all actively perform useful workload. When one fails, its workload is redistributed to other components. (More Economical)
- Fault-Tolerance: Most expensive configuration, as N components deliver performance of only one component, at more than N times the cost. The failure of N–1 components is masked (not visible to the user).

Failover

- Probably the most important feature demanded in current clusters for commercial applications.
- When a component fails, this technique allows the remaining system to take over the services originally provided by the failed component.
- A failover mechanism must provide several functions: failure diagnosis, failure notification, and failure recovery.
- Failure diagnosis - detection of a failure and the location of the failed component that causes the failure.
  - Commonly used technique - heartbeat, where the cluster nodes send out a stream of heartbeat messages to one another.
  - If the system does not receive the stream of heartbeat messages from a node, it can conclude that either the node or the network connection has failed.

Recovery Schemes

- Failure recovery refers to the actions needed to take over the workload of a failed component.
- Two types of recovery techniques:
- Backward recovery - the processes running on a cluster periodically save a consistent state (called a checkpoint) to a stable storage.
  - After a failure, the system is reconfigured to isolate the failed component, restores the previous checkpoint, and resumes normal operation. This is called rollback.
  - Backward recovery is easy to implement and is widely used.
  - Rollback implies wasted execution. If execution time is crucial, a forward recovery scheme should be used.
- Forward recovery - The system uses the failure diagnosis information to reconstruct a valid system state and continues execution.
  - Forward recovery is application-dependent and may need extra hardware.

Checkpointing and Recovery Techniques

- Kernel, Library, and Application Levels : Checkpointing at the operating system kernel level, where the OS transparently checkpoints and restarts processes.
- Checkpoint Overheads: The time consumed and storage required for checkpointing.
- Choosing an Optimal Checkpoint Interval: The time period between two checkpoints is called the checkpoint interval.
- Incremental Checkpoint: Instead of saving the full state at each checkpoint, an incremental checkpoint scheme saves only the portion of the state that is changed from the previous checkpoint.
- User-Directed Checkpointing: User inserts code (e.g., library or system calls) to tell the system when to save, what to save, and what not to save.

Cluster Job Scheduling and Management

A Job Management System (*JMS*) should have three parts:

- A user server lets the user submit jobs to one or more queues, specify resource requirements for each job, delete a job from a queue, inquire about the status of a job or a queue.
- A job scheduler that performs job scheduling and queuing according to job types, resource requirements, resource availability, and scheduling policies.
- A resource manager that allocates and monitors resources, enforces scheduling policies, and collects accounting information.

JMS Administration

- JMS should be able to dynamically reconfigure the cluster with minimal impact on the running jobs.
- The administrator's prologue and epilogue scripts should be able to run before and after each job for security checking, accounting, and cleanup.
- Users should be able to cleanly kill their own jobs.
- The administrator or the JMS should be able to cleanly suspend or kill any job.
  - Clean means that when a job is suspended or killed, all its processes must be included.
  - Otherwise some "orphan" processes are left in the system, wasting cluster resources and may eventually render the system unusable.

Cluster Job Types

Several types of jobs execute on a cluster.

- Serial jobs run on a single node.
- Parallel jobs use multiple nodes.
- Interactive jobs are those that require fast turnaround time, and their input/output is directed to a terminal.
  - These jobs do not need large resources, and the users expect them to execute immediately, not made to wait in a queue.
- Batch jobs normally need more resources, such as large memory space and long CPU time.
  - But they do not need immediate response.
  - They are submitted to a job queue to be scheduled to run when the resource becomes available (e.g., during off hours).

Characteristics of Cluster Workload

- Roughly half of parallel jobs are submitted during regular working hours.
- Almost 80% of parallel jobs run for 3 minutes or less.
- Parallel jobs running over 90 minutes account for 50% of the total time.
- The sequential workload shows that 60% to 70% of workstations are available to execute parallel jobs at any time, even during peak daytime hours.
- On a workstation, 53% of all idle periods are 3 minutes or less, but 95% of idle time is spent in periods of time that are 10 minutes or longer.
- A 2:1 rule applies, which states that a network of 64 workstations, with a proper JMS software, can sustain a 32-node parallel workload in addition to the original sequential workload.
  - In other words, clustering gives a supercomputer half of the cluster size for free!

Multi-Job Scheduling Schemes

- Cluster jobs may be scheduled to run at a specific time (calendar scheduling) or when a particular event happens (event scheduling).
- Jobs are scheduled according to priorities based on submission time, resource nodes, execution time, memory, disk, job type, and user identity.
- With static priority, jobs are assigned priorities according to a predetermined, fixed scheme.
  - A simple scheme is to schedule jobs in a first-come, first-serve fashion.
  - Another scheme is to assign different priorities to users.
- With dynamic priority, the priority of a job may change over time.

Job Scheduling Issues and Schemes for Cluster Nodes

Issue	Scheme	Key Problems
Job priority	**Non-preemptive**	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

Scheduling Modes (1)

Dedicated Mode :

- Only one job runs in the cluster at a time, and at most one process of the job is assigned to a node at a time.
- The single job runs until completion before it releases the cluster to run other jobs.

Space Sharing :

Multiple jobs can run on disjoint partitions (groups) of nodes simultaneously.

- At most one process is assigned to a node at a time.
- Although a partition of nodes is dedicated to a job, the interconnect and the I/O subsystem may be shared by all jobs.

Scheduling Modes (2)

- Time sharing :
- Multiple user processes are assigned to the same node.
- Time-sharing introduces the following parallel scheduling policies:
  1. Independent Scheduling (Independent): Uses the operating system of each cluster node to schedule different processes as in a traditional workstation.
  2. Gang Scheduling: Schedules all processes of a parallel job together. When one process is active, all processes are active.
  3. Competition with Foreign (Local) Jobs: Scheduling becomes more complicated when both cluster jobs and local jobs are running. The local jobs should have priority over cluster jobs.
     - Dealing with Situation: Stay or Migrate job

Migration Scheme Issues

1. Node Availability: Can the job find another available node to migrate to?
   - Berkeley study : Even during peak hours, 60% of workstations in a cluster are available.
2. Migration Overhead: What is the effect of the migration overhead? The migration time can significantly slow down a parallel job.
   - Berkeley study : a slowdown as great as 2.4 times.
   - Slowdown is less if a parallel job is run on a cluster of twice the size.
   - e.g. a 32-node job on a 60-node cluster – migration slowdown no more than 20%, even when migration time of 3 minutes.
3. Recruitment Threshold: the amount of time a workstation stays unused before the cluster considers it an idle node. What should be the recruitment threshold?

Job Management Systems Features

- Most support heterogeneous Linux clusters.
  - All support parallel and batch jobs.
- If enterprise cluster jobs are managed by a JMS, they will impact the owner of a workstation in running the local jobs.
- All packages offer some kind of load-balancing mechanism to efficiently utilize cluster resources. Some packages support checkpointing.
- Most packages cannot support dynamic process migration.
  - They support static migration: a process can be dispatched to execute on a remote node when the process is first created.
  - However, once it starts execution, it stays in that node.
- All packages allow dynamic suspension and resumption of a user job by the user or by the administrator.
  - All packages allow resources (e.g., nodes) to be dynamically added to or deleted.
- Most packages provide both a command-line interface and a graphic user interface.

END