Cluster and Grid Computing

Cheng-Zhong Xu

What’s a Cluster?

- Collection of independent computer systems working together as if a single system.
- Coupled through a scalable, high bandwidth, low latency interconnect.
PC Clusters: Contributions of Beowulf

- An experiment in parallel computing systems
- Established vision of low cost, high end computing
- Demonstrated effectiveness of PC clusters for some (not all) classes of applications
- Provided networking software
- Conveyed findings to broad community (great PR)
- Tutorials and book
- Design standard to rally community!
- Standards beget:
  - books, trained people,
  - Open source SW

Adapted from Gordon Bell, presentation at Salishan 2000

Towards Inexpensive Supercomputing

It is:

Cluster Computing..
The Commodity Supercomputing!
## Scalable Parallel Computers

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>MPP</th>
<th>SMP, CC-NUMA</th>
<th>Cluster</th>
<th>Distributed System</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of Nodes</strong></td>
<td>Ω(100) - Ω(1000)</td>
<td>Ω(10) - Ω(100)</td>
<td>Ω(100) or less</td>
<td>Ω(10) - Ω(1000)</td>
</tr>
<tr>
<td><strong>Node Complexity</strong></td>
<td>Fine or medium grain</td>
<td>Medium or coarse grain</td>
<td>Medium grain</td>
<td>Wide range</td>
</tr>
<tr>
<td><strong>Inter-node Communication</strong></td>
<td>Message passing or shared variables for DSM</td>
<td>Centralized and distributed shared memory</td>
<td>Message Passing</td>
<td>Shared files, RPC, message passing, IPC protocol</td>
</tr>
<tr>
<td><strong>Job Scheduling</strong></td>
<td>Single run queues at best</td>
<td>Single run queue mostly</td>
<td>Multiple queues but coordinated</td>
<td>Independent multiple Queues</td>
</tr>
<tr>
<td><strong>SSE Support</strong></td>
<td>Partially</td>
<td>Always in SMP and some NUMA</td>
<td>Desired</td>
<td>No</td>
</tr>
<tr>
<td><strong>Node OS Copies and Type</strong></td>
<td>N microkernels and 1 monolithic OS at core</td>
<td>One monolithic for SMP and multiple for NUMA</td>
<td>EOS platforms (Homogeneous or microkernel)</td>
<td>EOS platforms (Heterogeneous)</td>
</tr>
<tr>
<td><strong>Address Space</strong></td>
<td>Multiple (Single for DSM)</td>
<td>Single</td>
<td>Multiple or single</td>
<td>Multiple</td>
</tr>
<tr>
<td><strong>Inter-node Security</strong></td>
<td>Unnecessary</td>
<td>Unnecessary</td>
<td>Required if exposed</td>
<td>Required</td>
</tr>
<tr>
<td><strong>Ownership</strong></td>
<td>One organization</td>
<td>One organization</td>
<td>One or More organizations</td>
<td>Many organizations</td>
</tr>
<tr>
<td><strong>Network Protocol</strong></td>
<td>Nonstandard</td>
<td>Nonstandard</td>
<td>Standard &amp; Lightweight</td>
<td>Standard</td>
</tr>
</tbody>
</table>

## Design Space of Competing Computer Architecture

- **Distributed System**
- **Future Cluster**
- **MPP**
- **CC-NUMA**
- **SMP**

**Size Scalability**

**Single System Image**

**Future Cluster**
Clusters of SMPs

- SMPs are the fastest commodity machine, so use them as a building block for a larger machine with a network
- Common names:
  - CLUMP = Cluster of SMPs
  - Hierarchical machines, constellations
- Most modern machines look like this:
  - Millennium, IBM SPs, (not the t3e)...
- What is the right programming model???
  - Treat machine as “flat”, always use message passing, even within SMP (simple, but ignores an important part of memory hierarchy).
  - Shared memory within one SMP, but message passing outside of an SMP.

Cluster of SMP Approach

- A supercomputer is a stretched high-end server
- Parallel system is built by assembling nodes that are modest size, commercial, SMP servers – just put more of them together

![Image from LLNL](image_from_llnl.png)
Motivation for using Clusters

- Surveys show utilisation of CPU cycles of desktop workstations is typically <10%.
- Performance of workstations and PCs is rapidly improving
- As performance grows, percent utilisation will decrease even further!
- Organisations are reluctant to buy large supercomputers, due to the large expense and short useful life span.
Motivation for using Clusters

- The development tools for workstations are more mature than the contrasting proprietary solutions for parallel computers - mainly due to the non-standard nature of many parallel systems.
- Workstation clusters are a cheap and readily available alternative to specialised High Performance Computing (HPC) platforms.
- Use of clusters of workstations as a distributed compute resource is very cost effective - incremental growth of system!!!
**Cycle Stealing**

- Typically, there are three types of owners, who use their workstations mostly for:
  1. Sending and receiving email and preparing documents.
  2. Software development - edit, compile, debug and test cycle.
  3. Running compute-intensive applications.

Cluster computing aims to steal spare cycles from (1) and (2) to provide resources for (3).

However, this requires overcoming the *ownership hurdle* - people are very protective of their workstations.

Usually requires organizational mandate that computers are to be used in this way.

Stealing cycles outside standard work hours (e.g. overnight) is easy, stealing idle cycles during work hours without impacting interactive use (both CPU and memory) is much harder.
Why Clusters now?  
(Beyond Technology and Cost)

- Building block is big enough  
  – complete computers (HW & SW) shipped in millions:  
    killer micro, killer RAM, killer disks,  
    killer OS, killer networks, killer apps.

- Workstations performance doubles every 18 mon.

- Higher link bandwidth (v 100Mbit Ethernet)  
  - Gigabit and 10Gigabit Switches

- Networks are faster  
  – Infiniband switch: 10Gbps+6us (100us in Gigabit net)

- Demise of Mainframes, Supercomputers, & MPPs

Architectural Drivers…(cont)

- Node architecture dominates performance  
  – processor, cache, bus, and memory  
  – design and engineering $ \Rightarrow$ performance

- Greatest demand for performance is on large systems  
  – must track the leading edge of technology without lag

- MPP network technology => mainstream  
  – system area networks

- System on every node is a powerful enabler  
  – very high speed I/O, virtual memory, scheduling, …
...Architectural Drivers

- Clusters can be grown: Incremental scalability (up, down, and across)
  - Individual nodes performance can be improved by adding additional resource (new memory blocks/disks)
  - New nodes can be added or nodes can be removed
  - Clusters of Clusters and Metacomputing
- Complete software tools
  - Threads, PVM, MPI, DSM, C, C++, Java, Parallel C++, Compilers, Debuggers, OS, etc.
- Wide class of applications
  - Sequential and grand challenging parallel applications

Clustering of Computers for Collective Computing: Trends

- High-end Mainframes
- Minicomputers
- Unix WS Clusters
- High-volume PC Clusters

OPPORTUNITIES
&
CHALLENGES

Opportunity of Large-scale Computing on NOW

- Shared Pool of Computing Resources: Processors, Memory, Disks
- Interconnect
- Guarantee at least one workstation to many individuals (when active)
- Deliver large % of collective resources to few individuals at any one time
Windows of Opportunities

- MPP/DSM:
  - Compute across multiple systems: parallel.
- Network RAM:
  - Idle memory in other nodes. Page across other nodes idle memory
- Software RAID:
  - File system supporting parallel I/O and reliability, mass-storage.
- Multi-path Communication:
  - Communicate across multiple networks: Ethernet, ATM, Myrinet

Parallel Processing

- Scalable Parallel Applications require
  - Good floating-point performance
  - Low overhead communication scalable network bandwidth
  - Parallel file system
Network RAM

- Performance gap between processor and disk has widened.
- Thrashing to disk degrades performance significantly.
- Paging across networks can be effective with high performance networks and OS that recognizes idle machines.
- Typically thrashing to network RAM can be 5 to 10 times faster than thrashing to disk.

Software RAID: Redundant Array of Workstation Disks

- I/O Bottleneck:
  - Microprocessor performance is improving more than 50% per year.
  - Disk access improvement is < 10%
  - Application often perform I/O
- RAID cost per byte is high compared to single disks.
- RAIDs are connected to host computers which are often a performance and availability bottleneck.
- RAID in software, writing data across an array of workstation disks provides performance and some degree of redundancy provides availability.
Software RAID, Parallel File Systems, and Parallel I/O

Cluster Computer and its Components
Clustering today

- Clustering gained momentum when 3 technologies converged:
  - 1. Very HP Microprocessors
    - workstation performance = yesterday supercomputers
  - 2. High speed communication
    - Comm. between cluster nodes >= between processors in an SMP.

Cluster Computer Architecture

- Sequential Applications
- Parallel Applications
- Parallel Programming Environments
- Cluster Middleware
  (Single System Image and Availability Infrastructure)
- PC/Workstation
  - Comm. SW
  - Net. Interface HW
- High Speed Network/Switch
Cluster Components...1a
Nodes

- Multiple High Performance Components:
  - PCs
  - Workstations
  - SMPs (CLUMPS)
  - Distributed HPC Systems leading to Metacomputing
- They can be based on different architectures and running difference OS

Cluster Components...1b
Processors

- There are many (CISC/RISC/VLIW/Vector..)
  - Intel: Pentiums, Xeon, Merceed....
  - Sun: SPARC, ULTRASPARC
  - HP PA
  - IBM RS6000/PowerPC
  - SGI MPIS
  - Digital Alphas
- Integrate Memory, processing and networking into a single chip
  - IRAM (CPU & Mem): (http://iram.cs.berkeley.edu)
  - Alpha 21366 (CPU, Memory Controller, NI)
Cluster Components...2
OS

- State of the art OS:
  - Linux (Beowulf)
  - Microsoft NT (Illinois HPVM)
  - SUN Solaris (Berkeley NOW)
  - 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)

Cluster Components...3
High Performance Networks

- Ethernet (10Mbps),
- Fast Ethernet (100Mbps),
- Gigabit Ethernet (1Gbps)
- SCI (Dolphin - MPI- 12micro-sec latency)
- ATM
- Myrinet (1.2Gbps)
- Digital Memory Channel
- FDDI
Cluster Components...4
Network Interfaces

- Network Interface Card
  - Myrinet has NIC
  - User-level access support
  - Alpha 21364 processor integrates processing, memory controller, network interface into a single chip..

Cluster Components...5
Communication Software

- Traditional OS supported facilities (heavy weight due to protocol processing).
  - Sockets (TCP/IP), Pipes, etc.
- Light weight protocols (User Level)
  - Active Messages (Berkeley)
  - Fast Messages (Illinois)
  - U-net (Cornell)
  - XTP (Virginia)
- System systems can be built on top of the above protocols
Cluster Components…6a
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 (globalised view of system resources). Telnet cluster.myinstitute.edu
- SA - Check pointing and process migration..

Cluster Components…6b
Middleware Components

- Hardware
  - DEC Memory Channel, DSM (Alewife, DASH) SMP Techniques
- OS / Gluing Layers
  - Solaris MC, Unixware, Glunix
- Applications and Subsystems
  - System management and electronic forms
  - Runtime systems (software DSM, PFS etc.)
  - Resource management and scheduling (RMS):
    - CODINE, LSF, PBS, NQS, etc.
Cluster Components…7a
Programming environments

- Threads (PCs, SMPs, NOW..)
  - POSIX Threads
  - Java Threads
- MPI
  - Linux, NT, on many Supercomputers
- PVM
- Software DSMs (Shmem)

Cluster Components…7b
Development Tools?

- 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
- Visualization Tools
Cluster Components…8
Applications

- Sequential
- Parallel / Distributed (Cluster-aware app.)
  - Grand Challenging applications
    - Weather Forecasting
    - Quantum Chemistry
    - Molecular Biology Modeling
    - Engineering Analysis (CAD/CAM)
    - ………………
  - PDBs, web servers, data-mining

Key Operational Benefits of Clustering

- System availability (HA). offer inherent high system availability due to the redundancy of hardware, operating systems, and applications.
- Hardware Fault Tolerance. redundancy for most system components (e.g., disk-RAID), including both hardware and software.
- OS and application reliability. run multiple copies of the OS and applications, and through this redundancy
- Scalability. adding servers to the cluster or by adding more clusters to the network as the need arises or CPU to SMP.
- High Performance. (running cluster enabled programs)
Classification
of Cluster Computer

Clusters Classification

- Based on Focus (in Market)
  - High Performance (HP) Clusters
    - Grand Challenging Applications
  - High Availability (HA) Clusters
    - Mission Critical applications
HA Cluster: Server Cluster with "Heartbeat" Connection

Clusters Classification

- Based on Workstation/PC Ownership
  - Dedicated Clusters
  - Non-dedicated clusters
    - Adaptive parallel computing
    - Also called Communal multiprocessing
Clusters Classification..3

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

Building Scalable Systems: Cluster of SMPs (Clumps)

Performance of SMP Systems Vs. Four-Processor Servers in a Cluster
Clusters Classification..4

- 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..5

- Based on node components architecture & configuration (Processor Arch, Node Type: PC/Workstation.. & OS: Linux/NT..):
  - Homogeneous Clusters
    • All nodes will have similar configuration
  - Heterogeneous Clusters
    • Nodes based on different processors and running different OSes.
Clusters Classification..6a
Dimensions of Scalability & Levels of Clustering

Clusters Classification..6b
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)
- (using ATMs Net)
- **Internet-wide Clusters=Global Clusters** (#nodes: 1000s to many millions)
  - Metacomputing
  - Web-based Computing
  - Agent Based Computing
    - Java plays a major role in web and agent based computing
Major issues in cluster design

- Size Scalability (physical & application)
- Enhanced Availability (failure management)
- Single System Image (look-and-feel of one system)
- 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.)

Cluster Middleware
and
Single System Image
A typical Cluster Computing Environment

Application

PVM / MPI / RSH

Hardware / OS

CC should support

- Multi-user, time-sharing environments
- Nodes with different CPU speeds and memory sizes (heterogeneous configuration)
- Many processes, with unpredictable requirements
- Unlike SMP: insufficient “bonds” between nodes
  - Each computer operates independently
  - Inefficient utilization of resources
The missing link is provided by cluster middleware/underware

Application

PVM / MPI / RSH

Hardware/OS

Middleware or Underware

SSI Clusters--SMP services on a CC

“Pool Together” the “Cluster-Wide” resources

- Adaptive resource usage for better performance
- Ease of use - almost like SMP
- Scalable configurations - by decentralized control

Result: HPC/HAC at PC/Workstation prices
What is Cluster Middleware?

- An interface between between use applications and cluster hardware and OS platform.
- Middleware packages support each other at the management, programming, and implementation levels.
- Middleware Layers:
  - SSI Layer
  - Availability Layer: It enables the cluster services of
    - Checkpointing, Automatic Failover, recovery from failure,
    - fault-tolerant operating among all cluster nodes.

Middleware Design Goals

- Complete Transparency (Manageability)
  - Lets the see a single cluster system..
    - Single entry point, ftp, telnet, software loading...
- Scalable Performance
  - Easy growth of cluster
    - no change of API & automatic load distribution.
- Enhanced Availability
  - Automatic Recovery from failures
    - Employ checkpointing & fault tolerant technologies
  - Handle consistency of data when replicated..
**What is Single System Image (SSI)?**

- A single system image is the **illusion**, created by software or hardware, that presents a collection of resources as one, more powerful resource.

- SSI makes the cluster appear like a single machine to the user, to applications, and to the network.

- A cluster without a SSI is not a cluster

---

**Benefits of Single System Image**

- Usage of system resources transparently
- Transparent process migration and load balancing across nodes.
- Improved reliability and higher availability
- Improved system response time and performance
- Simplified system management
- Reduction in the risk of operator errors
- User need not be aware of the underlying system architecture to use these machines effectively
**Desired SSI Services**

- Single Entry Point
  - `telnet cluster.wayne.edu`
  - `telnet node1.cluster.wayne.edu`
- Single File Hierarchy: xFS, AFS, Solaris MC Proxy
- Single Control Point: Management from single GUI
- Single virtual networking
- Single memory space - Network RAM / DSM
- Single Job Management: Glunix, Codine, LSF
- Single User Interface: Like workstation/PC windowing environment (CDE in Solaris/NT), may it can use Web technology

**Availability Support Functions**

- Single I/O Space (SIO):
  - any node can access any peripheral or disk devices without the knowledge of physical location.
- Single Process Space (SPS)
  - Any process on any node create process with cluster wide process wide and they communicate through signal, pipes, etc, as if they are one a single node.
- Checkpointing and Process Migration.
  - Saves the process state and intermediate results in memory to disk to support rollback recovery when node fails. PM for Load balancing...
- Reduction in the risk of operator errors
- User need not be aware of the underlying system architecture to use these machines effectively
It is a computer science notion of levels of abstractions (house is at a higher level of abstraction than walls, ceilings, and floors).

- Application and Subsystem Level
- Operating System Kernel Level
- Hardware Level
Cluster Programming Environments: Example

- Shared Memory Based
  - DSM
  - Threads/OpenMP (enabled for clusters)
  - Java threads (HKU JESSICA, IBM cJVM)
- Message Passing Based
  - PVM (PVM)
  - MPI (MPI)
- Parametric Computations
  - Nimrod/Clustor
- Automatic Parallelising Compilers
- Parallel Libraries & Computational Kernels (NetSolve)

Levels of Parallelism

- Code-Granularity
- Code Item
- Large grain (task level)
- Program

- Medium grain (control level)
  - Function (thread)

- Fine grain (data level)
  - Loop (Compiler)

- Very fine grain (multiple issue)
  - With hardware
Original Food Chain Picture

1984 Computer Food Chain
1994 Computer Food Chain

Computer Food Chain (Now and Future)
What Next ??

Clusters of Clusters and Grid Computing

Computational/Data Grid

- Coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations
  - direct access to computers, sw, data, and other resources, rather than file exchange
  - Such sharing rules defines a set of individuals and/or institutions, which form a virtual organization
  - Examples of VOs: application service providers, storage service providers, cycle providers, etc
- Grid computing is to develop protocols, services, and tools for coordinated resource sharing and problem solving in VOs
  - Security solutions for management of credentials and policies
  - RM protocols and services for secure remote access
  - Information query protocols and services for configuratin
  - Data management, etc
Scalable Computing

Administrative Barriers
- Individual
- Group
- Department
- Campus
- State
- National
- Globe
- Inter Planet
- Universe

Figure due to Rajkumar Buyya, University of Melbourne, Australia, www.gridbus.org