Introduction to Cluster Computing

Prabhaker Mateti
Wright State University
Dayton, Ohio, USA
Overview

- High performance computing
- High throughput computing
- NOW, HPC, and HTC
- Parallel algorithms
- Software technologies
“High Performance” Computing

- CPU clock frequency
- Parallel computers
- Alternate technologies
  - Optical
  - Bio
  - Molecular
“Parallel” Computing

- Traditional supercomputers
  - SIMD, MIMD, pipelines
  - Tightly coupled shared memory
  - Bus level connections
  - Expensive to buy and to maintain
- Cooperating networks of computers
“NOW” Computing

- Workstation
- Network
- Operating System
- Cooperation
- Distributed (Application) Programs
Traditional Supercomputers

- Very high starting cost
  - Expensive hardware
  - Expensive software
- High maintenance
- Expensive to upgrade
Traditional Supercomputers

No one is predicting their demise, but …
*-Computing

- Pervasive Computing
- Ubiquitous Computing
- Grid Computing
- Internet of Things
- Cloud Computing
Computational Grids

“Grids are persistent environments that enable software applications to integrate instruments, displays, computational and information resources that are managed by diverse organizations in widespread locations.”
Computational Grids

- Individual nodes can be supercomputers, or NOW
- High availability
- Accommodate peak usage
- LAN : Internet :: NOW : Grid
“NOW” Computing

- Workstation
- Network
- Operating System
- Cooperation
- Distributed+Parallel Programs
“Workstation Operating System”

- Authenticated users
- Protection of resources
- Multiple processes
- Preemptive scheduling
- Virtual Memory
- Hierarchical file systems
- Network centric
Network

- **Ethernet**
  - 10 Mbps: obsolete
  - 100 Mbps: almost obsolete
  - 1000 Mbps: standard

- **Protocols**
  - TCP/IP
Cooperation

- Workstations are “personal”
- Use by others
  - slows you down
  - Increases privacy risks
  - Decreases security
- …
- Willing to share
- Willing to trust
Distributed Programs

- Spatially distributed programs
  - A part here, a part there, …
  - Parallel
  - Synergy

- Temporally distributed programs
  - Finish the work of your “great grand father”
  - Compute half today, half tomorrow
  - Combine the results at the end

- Migratory programs
  - Have computation, will travel
• Single program, multiple data
• Contrast with SIMD
• Same program runs on multiple nodes
• May or may not be lock-step
• Nodes may be of different speeds
• Barrier synchronization
Conceptual Bases of Distributed+Parallel Programs

- Spatially distributed programs
  - Message passing
- Temporally distributed programs
  - Shared memory
- Migratory programs
  - Serialization of data and programs
(Ordinary) Shared Memory

- Simultaneous read/write access
  - Read : read
  - Read : write
  - Write : write
- Semantics not clean
  - Even when all processes are on the same processor
  - Mutual exclusion
Distributed Shared Memory

- “Simultaneous” read/write access by spatially distributed processors
- Abstraction layer of an implementation built from message passing primitives
- Semantics not so clean
Conceptual Bases for Migratory programs

- Same CPU architecture
  - X86, PowerPC, MIPS, SPARC, ..., JVM
- Same OS + environment
- Be able to “checkpoint”
  - suspend, and
  - then resume computation
  - without loss of progress
Clusters of Workstations

- Inexpensive alternative to traditional supercomputers
- High availability
  - Lower down time
  - Easier access
- Development platform with production runs on traditional supercomputers
Cluster Characteristics

- Commodity off the shelf hardware
- Networked
- Common Home Directories
- Open source software and OS
- Support message passing programming
- Batch scheduling of jobs
- Process migration
Why are Linux Clusters Good?

- Low initial implementation cost
  - Inexpensive PCs
  - Standard components and Networks
  - Free Software: Linux, GNU, MPI, PVM
- Scalability: can grow and shrink
- Familiar technology, easy for user to adopt the approach, use and maintain system.
Example Clusters

- July 1999
- 1000 nodes
- Used for genetic algorithm research by John Koza, Stanford University
- www.genetic-programming.com/
Largest (2007) Cluster System

- IBM BlueGene, 2007
- DOE/NNSA/LLNL
- Memory: 73728 GB
- OS: CNK/SLES 9
- Interconnect: Proprietary
- PowerPC 440
- 106,496 nodes
- 478.2 Tera FLOPS on LINPACK
<table>
<thead>
<tr>
<th>OS</th>
<th>Count</th>
<th>Share</th>
<th>Rmax (GF)</th>
<th>Rpeak (GF)</th>
<th>Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>426</td>
<td>85.20%</td>
<td>4897046</td>
<td>7956758</td>
<td>970790</td>
</tr>
<tr>
<td>Windows</td>
<td>6</td>
<td>1.20%</td>
<td>47495</td>
<td>86797</td>
<td>12112</td>
</tr>
<tr>
<td>Unix</td>
<td>30</td>
<td>6.00%</td>
<td>408378</td>
<td>519178</td>
<td>73532</td>
</tr>
<tr>
<td>BSD</td>
<td>2</td>
<td>0.40%</td>
<td>44783</td>
<td>50176</td>
<td>5696</td>
</tr>
<tr>
<td>Mixed</td>
<td>34</td>
<td>6.80%</td>
<td>1540037</td>
<td>1900361</td>
<td>580693</td>
</tr>
<tr>
<td>MacOS</td>
<td>2</td>
<td>0.40%</td>
<td>28430</td>
<td>44816</td>
<td>5272</td>
</tr>
<tr>
<td>Totals</td>
<td>500</td>
<td>100%</td>
<td>6966169</td>
<td>10558086</td>
<td>1648095</td>
</tr>
</tbody>
</table>

http://www.top500.org/stats/list/30/osfam Nov 2007
Top 500 Computer Systems

- http://top500.org/lists/2015/06/
- http://top500.org/featured/top-systems/
Development of Distributed+Parallel Programs

- New code + algorithms
- Old programs rewritten in new languages that have distributed and parallel primitives
- Parallelize legacy code
New Programming Languages

- With distributed and parallel primitives
- Functional languages
- Logic languages
- Data flow languages
Parallel Programming Languages

- based on the shared-memory model
- based on the distributed-memory model
- parallel object-oriented languages
- parallel functional programming languages
- concurrent logic languages
Condor

- Cooperating workstations: come and go.
- Migratory programs
  - Checkpointing
  - Remote IO
- Resource matching
- http://www.cs.wisc.edu/condor/
Portable Batch System (PBS)

- Prepare a .cmd file
  - naming the program and its arguments
  - properties of the job
  - the needed resources
- Submit .cmd to the PBS Job Server: `qsub` command
- Routing and Scheduling: The Job Server
  - examines .cmd details to route the job to an execution queue.
  - allocates one or more cluster nodes to the job
  - communicates with the Execution Servers (mom's) on the cluster to determine the current state of the nodes.
  - When all of the needed are allocated, passes the .cmd on to the Execution Server on the first node allocated (the "mother superior").
- Execution Server
  - will `login` on the first node as the submitting user and run the .cmd file in the user's home directory.
  - Run an installation defined prologue script.
  - Gathers the job's output to the standard output and standard error
  - It will execute installation defined epilogue script.
  - Delivers stdout and stderr to the user.
TORQUE, an open source PBS

- Tera-scale Open-source Resource and QUEueue manager (TORQUE) enhances OpenPBS
- Fault Tolerance
  - Additional failure conditions checked/handled
  - Node health check script support
- Scheduling Interface
- Scalability
  - Significantly improved server to MOM communication model
  - Ability to handle larger clusters (over 15 TF/2,500 processors)
  - Ability to handle larger jobs (over 2000 processors)
  - Ability to support larger server messages
- Logging
OpenMP for shared memory

- Distributed shared memory API
- User-gives hints as directives to the compiler
- http://www.openmp.org
Message Passing Libraries

- Programmer is responsible for initial data distribution, synchronization, and sending and receiving information
- Parallel Virtual Machine (PVM)
- Message Passing Interface (MPI)
- Bulk Synchronous Parallel model (BSP)
BSP: Bulk Synchronous Parallel model

- Divides computation into supersteps
- In each superstep a processor can work on local data and send messages.
- At the end of the superstep, a barrier synchronization takes place and all processors receive the messages which were sent in the previous superstep
BSP Library

- Small number of subroutines to implement
  - process creation,
  - remote data access, and
  - bulk synchronization.
- Linked to C, Fortran, ... programs
BSP: Bulk Synchronous Parallel model

- http://www.bsp-worldwide.org/
PVM, and MPI

- Message passing primitives
- Can be embedded in many existing programming languages
- Architecturally portable
- Open-sourced implementations
Parallel Virtual Machine (PVM)

- PVM enables a heterogeneous collection of networked computers to be used as a single large parallel computer.
- Older than MPI
- Large scientific/engineering user community
- http://www.csm.ornl.gov/pvm/
Message Passing Interface (MPI)

- http://www-unix.mcs.anl.gov/mpi/
- MPI-2.0  http://www.mpi-forum.org/docs/
- MPICH:  www.mcs.anl.gov/mpi/mpich/ by Argonne National Laboratory and Missisippy State University
- LAM:  http://www.lam-mpi.org/
- http://www.open-mpi.org/
Kernels Etc Mods for Clusters

- Dynamic load balancing
- Transparent process-migration
- Kernel Mods
  - http://openmosix.sourceforge.net/
  - http://kerrighed.org/
  - http://openssi.org/
  - http://ci-linux.sourceforge.net/
  - CLuster Membership Subsystem ("CLMS") and Internode Communication Subsystem
  - http://www.gluster.org/
    - GlusterFS: Clustered File Storage of peta bytes.
    - GlusterHPC: High Performance Compute Clusters
  - http://boinc.berkeley.edu/
    - Open-source software for volunteer computing and grid computing
- Condor clusters
More Information on Clusters

- http://lcic.org/ “a central repository of links and information regarding Linux clustering, in all its forms.”
- www.beowulf.org resources for of clusters built on commodity hardware deploying Linux OS and open source software.
- http://linuxclusters.com/ “Authoritative resource for information on Linux Compute Clusters and Linux High Availability Clusters.”
- http://www.linuxclustersinstitute.org/ “To provide education and advanced technical training for the deployment and use of Linux-based computing clusters to the high-performance computing community worldwide.”
References

- Cluster Hardware Setup
- PVM  http://www.csm.ornl.gov/pvm/
- MPI  http://www.open-mpi.org/
- Condor http://www.cs.wisc.edu/condor/