High-Performance Computing

Roberto Beauclair
High-Performance Computing

- Basic Concepts
- Today
- Future Trends
- Enabling Technology
- Cluster Projects Worldwide
- Linux Cluster
- PVM e/ou MPI
- Remarks
Need of more Computing Power: Grand Challenge Applications

Solving technology problems using computer *modeling*, *simulation* and *analysis*

- Life Science
- Aerospace
- GIS
- Mechanical Design & Analysis (CAD/CAM)
HPC - Today

- In the past decade, the world has experienced one of the most exciting periods in computer development;
- Microprocessors have become smaller, denser, and more powerful;
- The result is that microprocessor based supercomputing is rapidly becoming the technology of preference in attacking some of the most important problems of science and engineering.
Fastest Installed Computers

- **Top500 supercomputers at sites worldwide:**
  http://www.netlib.org/benchmark/top500.html

- **Provides a snapshot of the HPC installed around the world**
  - installation site: customer, location, year
  - system: manufacturer, system type, #processors
  - type of customer: industrial, university, …
  - major application area.

- **Measures the performance for the TPP Linpack Benchmark**
  \[ Ax = b \]
HPC - Future

- PetaFlops = $10^{15}$ Flops (today: $\sqrt{10^{15}}$ Flops)
- 1 P Flops ≈ workstation computing for 1 year
- Building a computer ten times more powerful than all the networked computing capability
- Looking at four interrelated areas:
  - applications and algorithms
  - device technology
  - architecture and systems
  - software technology
HPC - Trends

- **Move toward shared memory**
  - SMPs and Distributed Shared Memory
  - Shared address space with deep memory hierarchy
- **Clustering of shared memory machines**
  - Emergence of PC commodity systems
    - Pentium based (Linux or NT driven)
- **Efficiency of message passing and data parallel programming**
  - Helped by standard efforts such as PVM, MPI and HPF

At UTK cluster of 14 Dual Pentium based: 7.2 G Flops
4 Major Efforts in the US

- **DOE ASCI Program (≈$450M/year)**
  - T FLOPS level computing
    - 1, 3, 10, 30 T Flops systems
- **DOD Modernization (≈$250M/year)**
  - Four Research Centers
- **NSF Supercomputing Centers PACI (≈$65M/year)**
  - NCSA, UCSD, SDSC, …
- **DOE Scientific Simulation Plan (≈$250M/year)**
  - Computer Science and Enabling Technologies
  - Combustion and Climate Modeling
TORC
Tennessee - Oak Ridge Cluster

Combining wide-area Pentium clusters with high speed networking.

- 12 dual - 300MHz P II
- 4 dual & 2 single - 200MHz P Pro & 300 MHz Alpha
- Mixed Linux/NT
- 10 dual - 266MHz P II

Myrinet & 100 Mb/s Ethernet

http://www.netlib.org/utk/torc/
ATLAS Project (PhiPAC)
Automatically Tuned Linear Algebra Software

- Automatic generation of computational kernel for RISC architectures via parameters study.
- Code generator takes about 1-2 hours to run.
  - Done once for a new architecture
  - Written in ANSI C
  - Self adapting code
- Currently provided:
  - matrix-matrix & matrix-vector multiply
- Code is generated for “Super-Scalar”, cached based architecture.
Avalon

- DEC Alpha/Linux Cluster with 70 processors
- 10 G Flops for $152k (67 G Flops/Million $)
- 60 million particle molecular dynamics simulation of shock-induced plasticity. This simulation sustained approximately 10 G Flops over a 44 hour period, and saved 68 G bytes of raw data.
- Also performed a gravitational treecode N-body simulation of galaxy formation using 9.75 million particles, which sustained an average of 6.78 G Flops over a 26 hour period.
HPC - Application Directions

- Past
  - mostly monolithic
  - mostly one programming language
  - computation intensive
  - batch
  - hours/days/weeks

- Present & Future
  - multi-modular
  - multi-language
  - multi-developers
  - computation intensive
  - data intensive
  - real time
  - minutes/hours/days
  - visualization in real time
  - interactive steering
HPC - Platform Directions

- **Past**
  - vector processors
  - SIMD MPPs

- **Present**
  - distributed memory MPs
  - shared memory MPs
HPC - Platform Directions

- Future
  - Distributed Computers
  - Heterogeneous Platforms
  - Issues
    - Latencies
      - internode
      - intranode
    - Bandwidth
      - different for different links
      - different based on traffic
HPC - Metacomputing Objectives

- Flexibility and extensibility
- Site autonomy
- Scalable architecture
- Single global namespace
- Easy-to-use, seamless environment

- High performance through parallelism
- Security
- Management / exploitation of heterogeneity
- Multi-Language interoperability
- Fault-tolerance
Metacomputing Projects

System Users
Scientists and engineers using computation to accomplish Lab missions.

Intelligent Interface
A knowledge-based environment that offers users guidance on complex computing tasks.

Cluster Operating System
The software which coordinates the interplay of computers, networks and storage.

Supercomputing
Heterogeneous collection of high-performance computer hardware and software resources.

Networking
The hardware and software that permits communication among distributed users and computer resources.

Mass Storage
A collection of devices and software that allow temporary and long-term archival storage of information.

Middleware
Software tools that enable interaction among users, applications and system resources.
HPC on Linux Cluster

No Low Cost Supercomputing
HPC on Linux Cluster

- Performance of PC components has almost reached of those used in supercomputers …
  - Microprocessors
  - Networks
  - Operating Systems
  - Programming Environment
  - Applications

- Rate of performance improvement of commodity components is too high.
The Need for Alternative Supercomputing Resources

- Cannot afford to buy “Big Iron” machines?
- Cluster are best-alternative!
  - Supercomputing-class commodity components are available.
  - They “fit” very well with today’s future funding model.
  - Can leverage upon future technological advances
    VLSI, CPUs, Networks, Disk, Memory, Cache, OS, Programming Tools, Applications, ...
Linux Cluster: Best choice!

- High Performance Computing
  - Parallel computers/supercomputer-class workstation cluster
  - Dependable parallel computers
- High Availability Computing
  - Mission-critical systems
  - Fault-tolerant computing
By the way, what is a Cluster?

- 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 resource.

- A typical cluster:
  - Network: Faster! Closer connection than a typical LAN
  - Low latency communication protocols
  - Looser connection than SMP
So, What’s so different about Clusters?

- Commodity Parts?
- Communication Packages?
- Incremental Scalability?
- Independent Failure?
- Intelligent Network Interface?
- Complete System on Every Node
  → virtual memory, scheduler, files, ...
- Nodes can be used Individually or Combined...
Computer Food Chain :)
The Beowulf Project was started at Center of Excellence in Space Data and Information Sciences (CESDIS) in the summer of 1994 with the assembly of a 16 node cluster developed for the Earth and Space Sciences project (ESS) at the Goddard Space Flight Center (GSFC).

The project quickly spread to other NASA sites, other R&D labs and to universities around the world. The project’s scope and the number of Beowulf installations have grown over the years; in fact, they appear to continue to grow at increasing rate.

Beowulf-class: Is a pile-of-PCs that bring unprecedented price-performance scalable computing, by assembling a low-cost, commodity-based, parallel computer.
Cluster Configuration 1
Dedicate Cluster
Cluster Configuration 2
Enterprise Cluster

Shared Pool of Computing Resources:
Processors, Memory, Disks

Interconnect

Guarantee at least one workstation to many individuals

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
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

PC/Workstation
Comm. SW
Net. Interface HW

PC/Workstation
Comm. SW
Net. Interface HW

PC/Workstation
Comm. SW
Net. Interface HW

High Speed Network/Switch
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 \( \approx \) farms)
- Distributed Environment (Social issues)
- Manageability (admin. And control)
- Programmability (simple API if required)
- Applicability (cluster-aware and non-aware app.)
Linux-Based Tools for

High Availability Computing

High Performance Computing
Hardware

- Linux OS is running/driving...
  - PCs (Intel x86 processors)
  - Workstations (Digital Alphas)
  - SMPs (CLUMPS)
  - Clusters of Clusters

- Linux supports networking with
  - Ethernet (10 Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps)
  - SCI (Dolphin - MPI - 12 µs latency)
  - ATM
  - Myrinet (1.2 Gbps)
  - Digital Memory Channel
  - FDDI
Communication Software

- Traditional OS supported facilities (heavy weight due to protocol processing)
  - Sockets (TCP/IP), Pipes, etc.

- Light weight protocols (user Level)
  - Active Messages (AM) (Berkeley)
  - Fast Messages (Illinois)
  - U-net (Cornell)
  - XTP (Virginia)
  - Virtual Interface Architecture (industry standard)
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.impa.br

- SA - Check pointing and process migration
Cluster Middleware

- OS / Gluing Layers
  - Solaris MC, UnixWare, MOSIX
  - Beowulf “Distributed PID”

- Runtime Systems
  - Runtime systems (SW-DSW, PFS, etc.)
  - Resource management and scheduling
    - NQS
    - PBS
    - Codine
Programming Environment

- Threads (PCs, SMPs, NOW)
  - POSIX Threads e Java Threads
- MPI
  - http://www-unix.mcs.anl.gov/mpi/mpich
- PVM
  - http://www.epm.ornl.gov/pvm
- SW-DSM
  - Shared Memory
Development Tools

- Compilers: C, C++, Fortran, Java
- Debuggers
- Performance Analysis Tools
- Visualization Tools
- Network Load Balancing
  - http://proxy.iinchina.net/~wensong/ippfs
  - High Performance *by serving through light loaded machine*
  - High Availability *detecting failed nodes and isolating them from the cluster*
  - Transparent/Single System View
Applications

- Sequential (benefit from the cluster)
- Parallel / Distributed (Cluster-aware app.)
  - Grand Challenging applications
    - Weather Forecasting
    - Quantum Chemistry
    - Molecular Biology Modeling
    - Engineering Analysis (CAD/CAM)
    - Ocean Modeling
    - ...
  - Parallel Databases and Data-Mining
  - Web Servers
Typical Cluster Computer Environment

Application

PVM / MPI / RSH

Hardware/OS
Cluster Computer Should Support:

- Multi-user and Time-sharing Environment
- 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
Multi-computer OS for UNIX

- An OS layer that provides the applications with the illusion of working on a single system
- Remote operations are performed like local operations
- Transparent to the application → user interface unchanged
- http://www.mosix.cs.huji.ac.il/

Application

PVM / MPI / RSH

MOSIX

Hardware/OS
MOSIX is Main tool

Preemptive process migration that can migrate → any process, anywhere, anytime

- Supervised by distributed algorithms that respond on-line to global resource availability - transparently
- Load-balancing - migrate process from over-loaded to under-loaded nodes
- Memory ushering - migrate processes from a node that has exhausted its memory, to prevent paging/swapping
MOSIX for Linux at HUJI

- A scalable cluster configuration:
  - 50 Pentium-II 300 MHz
  - 38 Pentium-Pro 200 MHz (some are SMPs)
  - 16 Pentium-II 400 MHz (some are SMPs)
- Over 12 GB cluster-wide RAM
- Connected by the Myrinet 2.56 Gbps LAN
  Runs Red-Hat 6.0, based on Kernel 2.2.7
- Upgrade: HW with Intel, SW with Linux
- Download MOSIX:
  - http://www.mosix.cs.huji.ac.il/
Nimrod - A tool for parametric modeling on clusters

Nimrod: A Computational Workbench

- High Level Abstraction for Computational Modellers
- Little or no programming
- Ease of use
- Use of Distributed Computational Resource
- Heterogeneous platforms

Job processing with Nimrod
PARMON: A Cluster Monitoring Tool

PARMON Client on JVM

parmon

High-Speed Switch

PARMON Server on each node

parmond
Resource Utilization at a Glance
Top500 Supercomputing (www.top500.org) Sites declared Avalon (http://cnls.lanl.gov/avalon/), Beowulf cluster, the 113th most powerful computer in the world.

→ 70 processor DEC Alpha cluster (cost: $152K)
→ Completely commodity and Free Software
→ price/performance is $15/M FLOPS,
→ performance similar to 1993’s 1024-node CM-5
Adoption of the Approach
PC Clusters as Supercomputers

- NASA Goddard Space Flight Center - Beowulf, Hrothgar and theHive
- Los Alamos National Laboratory - Loki and Avalon
- Oak Ridge National Lab - Stone
- Drexel University - DragonWulf, Hrothgar
- CACR at Caltech - Hyglac and Naegling
- High Energy Physics lab in Germany - Hermes
- Ames Lab, SCL - Alice
- Duke University - Brahma
- Sandia National Lab, Livermore - Alice
- Clemson University - Grendel
- FEE CTE Prague - Magi
- NCAR - Tux

and many others.
PC Clusters as Supercomputers

- What are they doing with them?
  - Physics - high energy simulations
  - Cosmology - N-body system simulations
  - Seismic Data Processing
  - Photorealistic Rendering - RayTracing
  - Computer Science - Pure research interest
  - Continuous Speech Recognition Research
  - Generally any research environment, where CPU power and data storage are more important than standard support from SW & HW vendor.
I’m Not Going to Pay a Lot for This Supercomputer!

- The limited number of sale opportunities force vendors to try satisfying all customers, with the usual result that no one is really happy. There is simply no way to provide highly specialized software (such as parallelizing compiler) and simultaneously keep costs out of stratosphere.

- Commodity computer components are increasingly faster, cheaper and smaller. It is now possible to take these off-the-shelf parts and assemble machines which run neck-and-neck with the “big boys” of supercomputing, and is some instances, surpass them.
Loki

- Los Alamos National Laboratory

  *I’m Not Going to Pay a Lot for This Supercomputer!*

- 16 nodes
  - Pentium Pro
  - 2Gb RAM
  - 50Gb Disk

- Server
  - Dual Pentium Pro
  - 512 Mb, 25Gb Disk

- Linux Beowulf-class (RedHat)
Hyglac

- High Performance Computing Group at the Jet Propulsion Laboratory in Pasadena, California.
- It was originally assembled at Caltech’s Center for Advanced Computing Research.
- 16 node Beowulf-class computer. Each node with:
  - Pentium Pro 200, 128 MB RAM, 2.5 GB Disk
- LINUX (Red Hat), MPICH, LAM MPI and PVM
- Compilers installed:
  - Gnu C, C++ and FORTRAN, NAG F90, F77 and F90
Loki

Hyglac
<table>
<thead>
<tr>
<th>Machine</th>
<th>Procs</th>
<th>GFlops</th>
<th>MFlops/proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCI Red</td>
<td>1408</td>
<td>96.53</td>
<td>68.5</td>
</tr>
<tr>
<td>TMC CM-5</td>
<td>512</td>
<td>14.06</td>
<td>27.5</td>
</tr>
<tr>
<td>Intel Paragon</td>
<td>512</td>
<td>13.70</td>
<td>26.8</td>
</tr>
<tr>
<td>IBM SP-2</td>
<td>128</td>
<td>9.52</td>
<td>74.4</td>
</tr>
<tr>
<td>Cray T3D</td>
<td>256</td>
<td>7.94</td>
<td>31.0</td>
</tr>
<tr>
<td>Origin 2000</td>
<td>24</td>
<td>5.02</td>
<td>20.9</td>
</tr>
<tr>
<td>Loki + Hyglac</td>
<td>32</td>
<td>2.19</td>
<td>68.4</td>
</tr>
<tr>
<td>Loki</td>
<td>16</td>
<td>1.28</td>
<td>80.0</td>
</tr>
</tbody>
</table>
Naegling

- Caltech’s Beowulf Machine
- 114 Pentium Pro Nodes
  - each with 128 MB RAM, 3.1 GB Disk
- The front-end machine has 128 MB extra RAM and 8 GB extra storage space.
- Linux (RedHat), MPICH, PVM and BSP; Gnu C, C++ and FORTRAN (g77)
- Photorealistic Rendering by a Monte Carlo Method
theHive
Highly-parallel Integrated Virtual Environment

- The purpose of theHIVE is to demonstrate the usefulness of low cost high performance computers on NASA’s earth and space science applications and to study the ability of theHIVE to fulfill the need for a high performance computing resource integrated seamlessly into the working environment of multiple personal workstations.

- theHIVE is a Beowulf-class parallel computer, (e.i. a cluster of PCs running LINUX, connected by its own private LAN.) theHIVE is a cluster of 64 compute nodes, BEEs and 2 types of host or front-end nodes, the QUEEN and one or more DRONEs. The QUEEN administers theHIVE and the DRONEs allow access to theHIVE by users and data gathering facilities, such as remote sensing satellite antennas.
theHive
Highly-parallel Integrated Virtual Environment

64 Dual Pentium Pro
(128 processors)
8 Gb of RAM
1.2 Tb of Disk

66 Dual Pentium Pro
(133 processors)
22 Gb of RAM
900 Gb of Disk

10 Quad Pentium III
(40 processors)
5 Gb of RAM
45 Gb of Disk

16 Dual Pentium III
(32 processors)
8 Gb of RAM
72 Gb of Disk
eniac 2000

- University of Pennsylvania
- Funded by Intel Corporation
  - RedHat 5.0 & Windows NT 4.0
  - Fat-Tree topology (CLUMP)
  - 8 Compute-Nodes with
    - 4 processors (Pentium Pro 200)
    - 1Gb RAM, 9Gb Disk
  - 27 Development-Nodes with
    - Pentium II 300, 128 Mb, 4Gb Disk
Dominic: Cluster Cookbook

- < £ 15.000
- Ingredients:
  - 17 Dual Pentium II, 128 Mb RAM, 6.4Gb Disk
  - 17 Fast Ethernet Network Cards (100 Mbps)
  - 1 Hub
  - 2 Cabinets
  - Linux
  - PVM & MPI
MAGI

- Linux Cluster Supercomputer for Speech Recognition Group at CTU Prague
  - 8 Pentium Pro
  - 64 Mb RAM
  - 8.6Gb IDE
  - 100 Mbps Network
  - RedHat 4.0
  - PVM
What is PVM?

- Parallel Virtual Machine (PVM) is currently the most popular communication protocol for implementing distributed and parallel applications.
- Provides a framework to develop parallel programs on a network of heterogeneous machines, handling message routing, data conversion and task scheduling. An application is written as a collection of co-operating tasks.
- PVM can use streams or datagram protocols (TCP or UDP).
Compute \( \pi \) Using PVM

```c
#define NPROC 4

main(int argc, char **argv)
{
    register double lsum, width;
    double sum;
    register int intervals, i;
    int mytid, iproc, msgtag = 4;
    int tids[NPROC];  /* array of
        task ids */

    /* enroll in pvm */
    mytid = pvm_mytid();

    /* Join a group and, if I am the first
    instance,
        iproc=0, spawn more copies of myself
    */
    iproc = pvm_joingroup("pi");

    if (iproc == 0) {
        tids[0] = pvm_mytid();
        pvm_spawn("pvm_pi", &argv[1], 0,
            NULL, NPROC-1, &tids[1]);
    }

    /* make sure all processes are here */
    pvm_barrier("pi", NPROC);

    /* get the number of intervals */
    intervals = atoi(argv[1]);
    width = 1.0 / intervals;
    lsum = 0.0;
    for (i = iproc; i<intervals; i+=NPROC){
        register double x = (i + 0.5) *
            width;
        lsum += 4.0 / (1.0 + x * x);
    }

    /* sum across the local results & scale
        by width */
    sum = lsum * width;
    pvm_reduce(PvmSum, &sum, 1, PVM_DOUBLE,
        msgtag, "pi", 0);

    /* have only the console PE print the
        result */
    if (iproc == 0) {
        printf("Estimation of \pi is
                %14.12lf\n", sum);
    }

    /* Check program finished, leave group,
        exit pvm */
    pvm_barrier("pi", NPROC);
    pvm_exit();
    exit(0);
}
```
What is MPI?

- At the Supercomputing ’92 conference, a committee, later known as the MPI forum, was formed to develop a message-passing standard.

- The advantages of using MPI are:
  - It is a standard and should make porting among machines easy.
  - Vendors are responsible for developing efficient implementations of the MPI library.
  - Multiple implementations are freely available.
Compute $\pi$ Using MPI

```c
main(int argc, char **argv) {
    register double width;
double sum, lsum;
register int intervals, i;
int nproc, iproc;

    /* Initialize MPI */
    if (MPI_Init(&argc, &argv) != MPI_SUCCESS) exit(1);

    MPI_Comm_size( MPI_COMM_WORLD, &nproc);
    MPI_Comm_rank( MPI_COMM_WORLD, &iproc);

    MPI_Comm_size( MPI_COMM_WORLD, &nproc);
    MPI_Comm_rank( MPI_COMM_WORLD, &iproc);

    /* get the number of intervals */
    intervals = atoi(argv[1]);
    width = 1.0 / intervals;

    /* do the local computations */
    lsum = 0;
    for(i=iproc; i<intervals; i+=nproc){
        register double x = (i + 0.5) * width;
        lsum += 4.0 / (1.0 + x * x);
    }

    /* sum across the local results & scale by width*/
    lsum *= width;
    MPI_Reduce(&lsum, &sum, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);

    if (iproc == 0) {
        printf("Estimation of pi is %14.12lf\n", sum);
    }
}
```

Conclusions Remarks

Clusters are promising..

- Solve parallel processing paradox
- Offer incremental growth and matches with funding pattern
- New trends in hardware and software technologies are likely to make clusters more promising and fill SSI gap.
- Linux (PC) based Cluster Supercomputers can be seen everywhere!
Announcement: formation of

IEEE Task Force on Cluster Computing
(TFCC)

http://www.dcs.port.ac.uk/~mab/tfcc/
References and Links

- **Jack Dongarra**
  - http://www.netlib.org/utk/people/JackDongarra

- **NetSolve**
  - http://www.cs.utk.edu/netsolve/

- **Top500**
  - http://www.netlib.org/benchmark/top500.html

- **TORC**
  - http://www.netlib.org/utk/torc/
References and Links

- **Avalon**
  - [http://bifrost.lanl.gov/MD/MD.html](http://bifrost.lanl.gov/MD/MD.html)

- **Rajkumar Buyya**

- **IEEE Task Force on Cluster Computing**
  - [http://www.dcs.port.ac.uk/~mab/tfcc/](http://www.dcs.port.ac.uk/~mab/tfcc/)
E-Mail

tron@impa.br
Linux Cluster
Beowulf-class at IMPA

Draft Proposal
Linux Cluster at Visgraf

- 8 Processor Nodes
  - Pentium II 400 (Dual ?)
  - 128 Mb RAM (more ?)
  - 4.5Gb Disk (more ?)

- File Server
  - 30Gb Disk (more ?)
  - CD-ROM (RW ?, DVD ?)
  - 5Gb Tape DAT Backup

- Interactive Workstation
  - Graphic Accelerator
  - Color Printer (?)
Linux Cluster at IMPA

File Server

IMPA Network

File Server

NFS

Message-Passing

Interactive Workstation at Visgraf

NFS

Switch

Message-Passing

Interactive Workstation at Fluid

IMPA Network