

Cluster Computing for Computational Science

2005 Biennial Conference of the ACMS
Huntington University

Charles Peck
Earlham College

Overview

- Introduction
- Hardware Taxonomy
- Software Infrastructure
- Computational Science
- The Future
- The Bootable Cluster CD
- An Example of Computational Science:
Protein Folding
- Resources

Introduction

- The Cluster Computing Group at Earlham College
- Pedagogically motivated undergraduate student/faculty research
- What's a cluster?
 - Becker and Sterling
 - Beowulf - COTS hardware and open source software, NOW and POP

Hardware Taxonomy

- Clusters are one type of HPC resource, also consider large SMP and vector based "supercomputers"
- Why a cluster?
 - Cost effective
 - Broadly applicable
 - Redundancy can improve overall reliability
 - Very favorable memory to CPU ratio, *e.g.* BLAST
- Pre-engineered vs DIY clusters
- Parallelism is the key, present from die to SMP to cluster

Nodes

- Uni-processor vs SMP
- Name brand vs off-brand vs home-built
- Disk-full vs disk-less using PXE and related technologies
- Performance and capacity of the components - CPU, RAM, bus, NIC, disk

Interconnect

- Switches vs hubs
- Managed vs un-managed switches
- Fast ethernet and GigE
- Latency and bandwidth
- Non-COTS Interconnects
 - 10 GigE (IEEE 802.3ea)
 - Infiniband
 - Myrinet
 - Quadrics
 - Dolphin

Other Hardware Considerations

- Power
- Cooling
- Monitoring - temperature, humidity, power

Software Infrastructure

- Why use open source software?
- Linux vs other open source operating systems
- General distributions vs scientific distributions
- System management tools - C3, SystemImager, Ganglia
- Programming Tools
 - Message passing libraries - MPI, PVM
 - Other parallel paradigms, *e.g.* openMOSIX, shared memory, threads.
- Benchmarking

Computational Science

- What types of problems are clusters well suited for?
 - Embarrassingly parallel
 - Memory intensive
 - Decomposition by processing or by data
 - Simulations generally
- Domain specific software packages
- General purpose packages
- Caveats
 - Speedup and Efficiency
 - Computation to communication ratio

The Future

- Computational power continues to grow (Moore) but the level of abstraction available to scientists has not changed appreciably
- Harnessing grid based resources (individual machines, clusters, labs, *etc.*) for large scale distributed computing projects, *e.g.* SETI@home and Folding@Home
- Grand Challenge problems generally

The Bootable Cluster CD

- A "wet lab" for computational science education
- Facilitates instruction about parallel and distributed computing, not meant for permanent installations
- Support for both x86 and PowerPC based nodes
- A wide variety of software built-in and ready to use
- Reduces friction so that the class focus is on algorithms and data structures
- Curriculum modules under development, *e.g.* GROMACS, traveling salesperson, *etc.*

An Example of Computational Science: Protein Folding - 1

- Simulating the self assembly of proteins from chains of amino acids using Newton's laws of motion and ensemble molecular dynamics
- GROMACS, one popular open source molecular dynamics package
- The science
 - Molecular dynamics simulates Newton's laws of motion for systems with hundreds to millions of mass points (atoms)
 - Given a PDB file describing a sequence of amino acids and initial conditions (T, P, *etc.*) and length of simulation:

- * Generate all potential trajectories and measure their energy, bond angles, *etc.*
- * Choose one of the trajectories as the most likely (minimum energy) and use that as the next stable state
- * Stop when a stable conformation is detected or set number of steps is reached
- * Advance the time by one step and loop to generate all potential trajectories

An Example of Computational Science: Protein Folding - 2

- Each step represents 0.002 picosecond of simulation time, 500 steps = 1 ps, 1000 ps = 1 nanosecond, 1000 ns = 1 microsecond
- Fast folding proteins can take 10 microseconds to fold, others much longer
- Current large-scale clusters can sustain about 1 nanosecond of simulation time per 24 hours of real time
- 1000 days of real time for each microsecond of simulation
- 10,000 days to completely fold a fast protein

Resources

Beowulf mailing list, archives, How-Tos, *etc.*
- www.beowulf.org

IEEE Task Force on Cluster Computing -
www.ieeetfcc.org

ClusterWorld - www.clusterworld.com

Computing in Science and Engineering, IEEE
CS and the AIP - www.computer.org/cise

Linux Journal - www.linuxjournal.com

The Krell Institute - www.krellinst.org

The Bootable Cluster CD

- bccd.cs.uni.edu
- TWiki - bccd.cs.uni.edu/cgi-bin/twiki/view/Main/WebHome