

A Bioinformatics Introduction to Cluster Computing **By Andrew D. Boyd, MD and Abhijit Bose, PhD**

- Goals:
1. To introduce biologists and bioinformaticians to cluster computing
 2. To decrease the steep learning curve to use of a cluster
 3. To teach about how to solve problems with parallel computing
 4. To allow hands-on participation in using a Linux cluster

Andrew D. Boyd, MD

Dr. Boyd is a postdoctoral research fellow in Biomedical Information at the University of Michigan. He is under the mentorship of Dr. Brian Athey, who is an Associate Professor in Biomedical Informatics and the Director of the Michigan Center for Biological Information. His research is on bioinformatics applications on cluster computing. Dr. Boyd has presented a similar workshop to the University of Michigan Bioinformatics Students in March 2004. The feedback from the students was positive; they were able to after the course understand the basic Principles of Cluster Computing to apply to future research topics.

Abhijit Bose, PhD

Dr. Bose is the associate director of MGrid (Michigan Grid Research and Infrastructure Development) and a scientist at The University of Michigan. He is also a member of the DARPA-funded Virtual Soldier project team at Michigan. He has extensive experience in cluster computing and parallel algorithms. He has organized and taught eight NPACI (National Partnership for Advanced Computing Infrastructure) Parallel Computing workshops at Michigan since 2000. He has also taught a number of courses at the Annual Summer Insitute organized by the San Diego Supercomputer Center. These popular workshops have always been oversubscribed and additional sessions are often organized.

Tutorial Level: Introductory

Intended audience: Biologists, Bioinformaticians, and Algorithmicians who have some familiarity with UNIX/LINUX command line interface but have had little or no experience on a cluster computing system. The tutorial provides a comprehensive review of Linux clusters, available technologies for building such clusters, programming models with specific examples drawn from Bioinformatics applications. The participants will have a unique opportunity to use a Linux cluster during the hands-on session of the tutorial. It is recommended that participants bring their laptops enabled with Wireless/Ethernet cards and ssh client (putty for Windows) to take full advantage of the hands-on session.

Abstract:

Commodity processor-based clusters have rapidly become the compute systems of choice for computational modeling and simulation in many scientific fields. Many biological computations such as genomic and protein analysis are increasingly being performed on large-scale clusters based on Linux. There are many ways to configure a cluster, including 32 and 64-bit CPUs, interconnects such as Myrinet, Ethernet, Infiniband, and storage systems such as commodity RAIDs and SANs. This tutorial focuses on clusters

and their application to solving large-scale bioinformatics problems. The participants will learn the basics of Linux clusters, programming environments, batch and interactive computations, and a series of examples from Bioinformatics including how to use BLAST, Smith-Waterman, INTERPROSCAN etc. The basics of parallel computing on clusters such as Message Passing Interface (MPI) and threads will be introduced. The participants will learn how to interpret common debugging, system and application-related error messages on clusters. A unique feature of the tutorial is a hands-on session on a high-performance Linux cluster at The University of Michigan that the participants will be able to use during the course of the tutorial.

Outline

1. Introduction to clusters: Overview of common computing platforms: shared-memory vs distributed memory systems and clusters.
2. Generic architecture of a cluster: Processor families, common subsystems and networking with examples from currently available cluster architectures.
3. Programming environments
 - a. Programming models: Message Passing Interface (MPI), Threads, OpenMP, simple parallelisms using compiler directives.
 - b. Compilations: Tuning parameters for clusters and specific processor families.
 - c. Debugging: Common debugging tools such as PGDBG, ddd and TotalView, debugging with checkpoints across multiple processes.
 - d. Profiling: Common profiling tools such as PGPROF, gnu prof etc.
4. Scheduling and Job Submissions
 - a. Overview of scheduling on clusters, queues, time limits
 - b. Checkpointing and restarting
 - c. Different types of schedulers: Sun Grid Engine (SGE), Condor, PBS
 - d. Usage instructions and commands for above schedulers:
 - i. Preparing a submission script
 - ii. Parameters for each type of scheduler
 - iii. Submission and query instructions
5. Bioinformatics Examples
 - a. Genome versus Genome sequence comparison searches
 - i. BLAST
 - ii. Smith Waterman
 - iii. BLAT
 - iv. SSAHA
 - b. Protein domain identification using INTERPROSCAN
 - c. Molecular Dynamics simulation
6. A Bioinformatics Example problem worked out
 - a. Scope of the problem
 - i. Re-annotate oligo sequences with new unigene identifiers
 - b. One method attempted
 - i. mpiBLAST
 - c. Results of the scalability
 - d. Second method attempted

- i. Simple perl scripts
 - e. Results of the scalability
- 7. Example Cluster (Hands-on Session)
 - a. Overview of the hardware and software
 - b. Write perl scripts to run a number of example bioinformatics problems
 - c. Apply the lessons learnt in compiling, debugging and scheduling
- 8. Benchmarking
 - a. Open-source tools for benchmarking clusters
 - b. Timing utilities
 - c. Comparison among different processor families
 - d. Comparison among different interconnections
- 9. Troubleshooting
 - a. Common systems-related error messages in Linux clusters
 - b. Common programming errors (MPI, Threads, I/O)
 - c. Techniques to troubleshoot application errors such as memory, file access etc.
- 10. Information to bring to a System Administer when problems arise
 - a. Common trace files and commands for looking up error messages in Linux clusters