Applying Grid Technologies to Bioinformatics

Michael Karo[1],
Christopher Dwan[1], John Freeman[1],
Jon Weissman[2], Miron Livny[3], Ernest Retzel[1]

Abstract

The science of bioinformatics provides researchers with the tools necessary to unravel the mysteries of life and evolution, discover cures for disease, and control the evolution of living organisms. To assist researchers in managing the growing data processing and management demands associated with bioinformatics, we have created a production system that draws upon Grid based technologies to control several aspects of the process. We briefly discuss system architecture, results, and future directions of the project.

1. Introduction

There has been an explosion in the amount of data that characterizes the genetic makeup of many organisms within the last decade. Biologists collect, characterize, classify, and assemble this data to understand how an organism functions. The data processing applications involved are resource intensive, and require integration of data from many sources. Without a proper resource management infrastructure in place, the problem quickly becomes intractable.

Two important factors to consider when designing a solution for this environment are scalability and extensibility. As the amount of interest and funding in genetic research continues to grow, the ability to keep pace with the exponential growth in data and demand for processing becomes an ever increasing challenge. In addition, new algorithms for analyzing genetic data more efficiently and effectively are emerging everyday. A system designed to process data using a static set of applications would quickly become obsolete.

The Center for Computational Genomics and Bioinformatics (CCGB) at the University of Minnesota has created a production system to assist bioinformatics researchers in their studies. Many novel aspects of this system are based on Grid technologies [1], and are intended to address numerous concerns in addition to those already mentioned. We present an architecture that has been tested to meet the current needs of our organization and its numerous collaborators.

2. System Model

By masking the complexities of the workload management components, users are able to access resources and applications that would otherwise require in–depth knowledge of the underlying system. In addition, accurate characterization of application resource requirements contributes significantly to the effectiveness of their management and scheduling. The `tkbatch` infrastructure integrates this information and provides an interface for users to specify their processing needs to the underlying workload management system.

Separating the core infrastructure components from those that characterize the data processing applications greatly reduces the maintenance requirements of the core while providing the extensibility necessary for adapting and configuring applications. The components that characterize application sets are known as *toolboxes*. Through the `tkbatch` interface, a user organizes a set of *toolbox* components into *tool trees* that describe the specific applications, parameters, and dependencies to the underlying process management components. The interface restricts the user from creating incompatible relationships between applications and data types, thereby limiting the amount of human error that may be introduced into the process. The interface also allows users to define a set of *input files* that are to be processed. A *tool tree* coupled with an *input file set* forms a *watchlist* which contains all information necessary to construct job scripts and ancillary submission files.

A data management infrastructure known as *bioDATA* is responsible for providing a web based interface to processing results and statistical information. It consists of a set of tools that maintain a well defined directory structure known as the *bioDATA tree*. It works in conjunction with the data processing components to provide a current view of processing results with a minimal amount of human intervention. Data is first organized by project or grant, allowing access to sensitive or proprietary data to be controlled at the highest level. The data is then organized by submission set, which typically consists of data collected using similar methods. This allows researchers to apply the same parameters and processing methods against all data within a particular submission. Finally, the results of data processing are organized in a way that mirrors the *tool tree* used for processing and analysis.

In addition to housing raw data and processing results, metadata is also maintained within the *bioDATA tree* which may include such things as tissue type, sequence library, vector and primer, contact information and so forth. The metadata is stored in XML formatted files which reside within the data directories. The *bioDATA crawler* parses these files to produce HTML formatted

---

1 Center for Computational Genomics and Bioinformatics, University of Minnesota, Minneapolis, MN, USA, [mek | cdwan | jlf | ernest] @ahc.umn.edu
2 Department of Computer Science, University of Minnesota, Minneapolis, MN, USA, jon@cs.umn.edu
3 Department of Computer Science, University of Wisconsin, Madison, WI, USA, miron@cs.wisc.edu

summary tables and hypertext links. Summary statistics including read quality and average sequence length are also incorporated into the HTML files. Several CGI scripts exist to provide additional dynamic views of the information in the *bioDATA tree*.

Resource and process management for all nodes in the computing cluster is controlled by the *Condor High Throughput Computing System* [6]. The Condor system employes a scheme known as *Matchmaking* [2,3] to assign resource consumers (jobs) to resource providers (computers) in an opportunistic fashion. Condor provides a metascheduling component known as the *Directed Acyclic Graph Manager* (*DAGman*) that manages the dependencies for a predefined set of jobs.

3. Going Global

Condor has allowed us to address scalability issues through a facility known as *flocking* [4,5]. Flocking allows independent Condor clusters to share resources by migrating jobs between them. The CCGB cluster is currently configured to migrate jobs with two external clusters. The first comprises ten dual processor Linux systems located in the Computer Science Department at the University of Minnesota. The second is the Condor cluster at the University of Wisconsin, Madison. Our experiences configuring, managing, and adapting applications to this extended environment have been generally positive. One strategy we have utilized to reduce network bandwith demands has been to mirror certain datasets remotely. This has improved performance of homology searches significantly.

4. Results

Within six months of its initial deployment, over one million independent jobs were submitted through `tkbatch` and processed by our local Condor cluster. As more nodes are added to the cluster, both locally and through the use of flocking, throughput of the overall system will continue to increase. For example, one of the most demanding applications utilized by bioinformatics researchers is called BLAST [7]. The BLAST application produces a report that indicates the similarity of a query sequence to others stored within a set of *sequence databases*. Over 400,000 of these reports have been generated by our local researcher staff.

Many of the nodes within our cluster also function as workstations for our users. The impact on these users has been minimal, primarily due to the use of job preemption in scheduling policies for workstation class systems. The flexibility that Condor provides to enforce these types of scheduling policies will contribute to the expansion of our local cluster.

Over 1.1 million web pages are currently being served based on data in the *bioDATA tree*. The *bioDATA* component eliminates the need for human intervention when maintaining views of dynamic data, allowing researchers and developers to focus on constructing new and useful views. Feedback from collaborators has been very positive, now that they are able to examine their data in new ways.

The most important result has been the dramatic increase in throughput, allowing researchers to manage more and larger projects. One such project, sponsored by the Soybean Growers Association, consists of over 150,000 genetic sequence components. It is estimated that this project has grown to over twice the size that would have been possible to manage prior to the introduction of our tools. Eight full scale sequencing projects are currently managed by `tkbatch` and *bioDATA* while an additional eight legacy projects are planned for future integration. Turnaround time for data analysis has been dramatically reduced, allowing researches the ability to conduct broader studies against their data and to obtain answers to questions that could not have been previously posed.

5. Conclusions and Future Work

Integrating existing Grid based tools with software components that adapt them to configurable application sets produces an environment that exhibits excellent usability and resource utilization characteristics. Furthermore, we have demonstrated that Grid technologies can be employed to provide solutions for real world problems that require high levels of both scalability and extensibility in production environments. By integrating mechanisms for process and data management, we have developed a robust system for managing large scale bioinformatics projects.

Harnessing and adapting the tools and methodologies of Grid based computing has greatly accelerated our efforts to roll out a model for collaborative research computing. As these technologies continue to be developed, we will adapt them to meet our needs and enhance our environment.

References

1. I. Foster and C. Kesselman, *The Grid*. Morgan Kaufmann Publishers, Inc. © 1999
2. Rajesh Raman, Miron Livny, and Marvin Solomon, "Resource Management through Multilateral Matchmaking", *Proceedings of the Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9)*, Pittsburgh, Pennsylvania, August 2000, pp 290−291.
3. Rajesh Raman, Miron Livny, and Marvin Solomon, "Matchmaking: Distributed Resource Management for High Throughput Computing", *Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing*, July 28−31, 1998, Chicago, IL.
4. D. H. J Epema, Miron Livny, R. van Dantzig, X. Evers, and Jim Pruyne, "A Worldwide Flock of Condors : Load Sharing among Workstation Clusters" *Journal on Future Generations of Computer Systems* Volume 12, 1996
5. Jim Basney and Miron Livny, "Deploying a High Throughput Computing Cluster", in *High Performance Cluster Computing*. Rajkumar Buyya, Editor, Vol. 1, Chapter 5, Prentice Hall PTR, May 1999.
6. The Condor Project, http://www.cs.wisc.edu/condor
7. Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman, "Basic Local Alignment Search Tool", in *Journal of Molecular Biology*, Vol. 255, pages 403−410, 1990