Supporting Bioinformatics Applications using High Throughput Computing

Alex Younts
Purdue University
ay@purdue.edu

The field of bioinformatics is quickly growing to encompass many computation techniques to gather, search, analyze, and utilize biological information. The programs used in the field have advanced from simple linear searching to advanced statics-based text searching algorithms in the goal of reducing the time it taking to go through large amounts of genetic data collected [3].

Progress has been made between biologists and the computing world to make the process point and click simple. An example is the Biocompute group from Notre Dame. Through a grant from the National Science Foundation, the group has constructed and begun running a web-base portal allowing researchers to quickly search genetic databases using BLAST, SSAHA [4], and SHRIMP [1]. Performing a search on a small database is simple for almost any size query, but the problem comes from the need to search larger and larger databases for answers. The Biocompute team is using Condor to distribute some of the work done by researchers, but a barrier to expanding their work is the requirement to move large databases between sites running Condor.

The inherent problem with running data-intensive jobs between Condor sites is the complexity in maintaining high-bandwidth data transfers given the differences in network implementations and available storage. The potential for overwhelming a site's networking or local storage is a high enough road block to prevent researchers' data-intensive Condor jobs from ever running at remote sites. The solution to the problem is to be intelligent about what data flows over remote network links and to design resilient enough storage to support demand, while still working within real-world budgets.

To support data-intensive projects like Biocompute jobs from Notre Dame at Purdue University, two resources have to be provided. The first resource is Condor but this relationship between the two institutions has been established for several years. The second resource is scalable storage. Condor jobs at Purdue generally land on compute resources with local disk capable of providing 10-50GB of temporary, local storage. Bringing the data to local machines frees up rare networked storage to handle other tasks. Local storage may not be fast storage but the aggregate throughput is what makes running a large number of jobs possible.

Pre-staging all possible data sets on execution machines is not possible but staging large data sets on storage as individual sites is possible. Staging data sets to sites does have the disadvantage of still needing to use some networked resource to copy the required data set to local storage before a job starts, however simply copying a file is easier than supporting the reading and seeking operations that bioinformatics applications such as BLAST or SSAHA perform on a shared storage resource.

To create a scalable solution to distribute the files, there are two technologies that can be employed. The first is the Hadoop Distributed File System (HDFS) which is a highly scalable, distributed file system tuned to support write-once, read-many, large files. While many different file systems could fill this need, HDFS is unique in that it can be used to scavenge storage and present it as a unified namespace. The second technology that will allow for easy data set management and distribution is the Chirp server from Notre Dame University. Chirp now has a native HDFS interface and allows users to discover and use storage in an easy way [2]. These two technologies and the ability to host the data sets in a scalable way by adding and removing backing storage allow data intensive jobs through Condor to be a viable solution by providing a way to distribute large data sets.

A work flow to start a job at a remote site is now easier to envision. Instead of having to find local resources or support a WAN-based file system, users can use a web portal to upload any custom data sets which can be distributed to remote sites and then launch their applications with a few clicks. Their compute jobs can stage the required data sets to machine-local storage and return the answers back to researchers in much shorter times by utilizing the grid.

Supporting bioinformatics applications using high throughput computing resources by using Condor and a new scalable file system for data set distribution will open up the thousands of Condor slots to more potential use and allow for more science to be accomplished.

Resources

[1] http://biocompute.cse.nd.edu/
[2] http://www.nd.edu/~ccl/software/chirp/
[3] Achuthsankar S Nair Computational Biology & Bioinformatics - A gentle Overview, Communications of Computer Society of India, January 2007
[4] http://www.sanger.ac.uk/resources/software/ssaha/