An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics

Author: Ronald Taylor, Pacific Northwest National Laboratory, Richland, WA (ronald.taylor@pnl.gov)

This talk will present an overview of Apache Hadoop (http://hadoop.apache.org/) and associated open source software projects. Current Hadoop usage within the bioinformatics community will be summarized, and the advantages of Hadoop will be discussed as they pertain to subareas of bioinformatics analysis.

Hadoop is a software framework that can be installed on a commodity Linux cluster to permit large scale distributed data analysis. Hadoop provides the robust Hadoop Distributed File System (HDFS) as well as a Java-based API that allows parallel processing across the nodes of the cluster. Programs employ a Map/Reduce execution engine which functions as a fault-tolerant distributed computing system over large data sets - a method popularized by use at Google. There are separate Map and Reduce steps, each step done in parallel, each operating on sets of key-value pairs. Processing can be parallelized over thousands of nodes working on terabyte or larger sized data sets. The Hadoop framework automatically schedules map tasks close to the data on which they will work, with "close" meaning the same node or, at least, the same rack. Node failures are also handled automatically.

In addition to Hadoop itself, which is a top-level Apache project, there are subprojects build on top of Hadoop, such as Hive (http://hadoop.apache.org/hive/), a data warehouse framework used for ad hoc querying (with an SQL type query language) and used for more complex analysis; and Pig (http://hadoop.apache.org/pig/), a high-level data-flow language and execution framework whose compiler produces sequences of Map/Reduce programs for execution within Hadoop. Also, I will discuss HBase (http://hadoop.apache.org/hbase/), another Hadoop subproject inspired by Google's BigTable. Each table is stored as a multidimensional sparse map, with rows and columns, each cell having a time stamp. HBase adds a distributed, fault-tolerant scalable database onto the Hadoop distributed file system, permitting random access to the stored data. Other "NoSQL" scalable databases (e.g., Hypertable, Cassandra) will be briefly introduced as HBase alternatives. Additional topics covered will include (1) the Apache Mahout project (http://lucene.apache.org/mahout), which is parallelizing many machine learning algorithms in Hadoop, (2) Cascading (http://www.cascading.org/), an API for defining and executing fault tolerant data processing workflows on a Hadoop cluster, and (3) use of the Amazon Elastic Compute Cloud to run Hadoop.

Recent applications of Hadoop in bioinformatics will also be summarized. Processing of high throughput sequencing data (for example, mapping extremely large numbers of short reads onto a reference genome) is an area where Hadoop-based software is making an impact. Thus, these examples will highlight, among others, the Cloudburst software (M. Schatz, 2009) and other Hadoop-based software from University of Maryland researchers for analysis of next-generation DNA sequencing data (http://www.cbcb.umd.edu/software/). Also, applications such as CloudBLAST (Matsunaga et al. 2008) will be presented. Dean and Ghemawat made the point in their recent article (Jan 2010, Comm. of the ACM) that Hadoop is well-suited to fill a need for analysis of high throughput data coming from heterogeneous systems. In conclusion, I will briefly describe a new project at Pacific Northwest National Laboratory wherein we are proposing Hadoop and HBase-based development of a scientific data management system that can scale into the petabyte range, that will accurately and reliably store data acquired from various instruments, and that will store the output of analysis software and relevant metadata, all in one central distributed file system. My concluding point, extracted from that project, follows Dean & Ghemawat. That is, for much bioinformatics work not only is the scalability permitted by Hadoop and HBase important, but also of consequence is the ease of integrating and analyzing various disparate data sources into one data warehouse under Hadoop, in relatively few HBase tables.