Survey on Various Applications of Hadoop in Bioinformatics

Minerva Laishram
Master of Technology
Computer Science & Engineering
Visveswaraya Technological University

ABSTRACT

Apache Hadoop is an emerging technology that is widely used in the data intensive applications like Big Data Analysis. This technology is currently used in the searching applications of Google, Yahoo, and Amazon. But recently it is found that Hadoop is widely being used in the data analysis of genome sequences of Biosciences. Bioinformatics is the new field of science where the computer technology is used in studying the molecular biology. For example, Hadoop technology can be used in finding a particular genome sequence from a huge datasets of Genes in Genetics.

General Terms

Bigdata, MapReduce, HDFS, Bioinformatics

Keywords

Interest Locality, Data Groupings, CloudBLAST, Cloudburst

1. INTRODUCTION

Bioinformatics is an interdisciplinary field of science where the computer technologies are used in studying the molecular biology [5]. For example, Hadoop technology can be used in finding a particular genome sequence from a huge datasets of Genes in Genetics. Actually the concept of Bigdata is applied to Bioinformatics also. The various molecular data of DNA and RNA in the form of Bigdata can be analyzed using Hadoop technology. Various genomic sequences that are produced from million of species of our planet are stored as a huge datasets. Apache Hadoop is an open source technology that is very easily available in the Internet as an open source. Otherwise, the Bigdata of DNA and RNA sequences of the large number of species of the world can be processed on the latest technology available as Amazon Elastic Compute Cloud. MapReduce concepts are already used by Google, Yahoo etc.

Interest Locality is very important concept in the research fields of data analysis as the researchers or the domain scientists are only interested in a particular subset from the huge dataset, and scientists are expected to access one subset more frequently than others. [1] Big datasets may be available as text files, such as FASTA and FASTQ etc. In processing these types of text files, it takes hours to process. The scientists are trying to develop new algorithms in order to reduce these processing times making it more efficient, at the same time more reliable as compared to the existing algorithms.

The concept of parallel data processing applications that are already used by scientists were Message Passing Interface (MPI) and Hadoop on cloud computing platform to speed up the computational time takes on processing and analysis. Hadoop is an open source implementation developed as an Apache project. The man responsible for Hadoop discovery is Doug Cutting. Hadoop is provided with two very important concepts of data processing: Hadoop Distributed File System (HDFS) and MapReduce Programming model. HDFS is a distributed file system that stores each file in the form of sequence of blocks, each block having size of 64MB; all blocks in a file except the last block are of same size. HDFS is designed to process data in parallel form. MapReduce programming model is used as model to process data by calling map and reduce functions. MapReduce splits large inputs into small pieces and each pieces are distributed among the multiple nodes and then in each node, it is processed by calling the reduce function.

In this paper, Section 1 gives a brief introduction on the concept of bioinformatics and Apache Hadoop. Section 2 discuses about the background of Apache Hadoop and its model. Section 3 gives various applications of the Hadoop in bioinformatics. Finally section 4 gives conclusions.

2. BACKGROUND

2.1 Apache Hadoop

Apache Hadoop is an open source tool developed mainly for data intensive applications. This tool requires to be running in a cluster environment. Apache Hadoop can be installed in different ways: pseudo-distributed mode, standalone mode and multi-node cluster mode [7]. In pseudo-distributed mode, the Hadoop daemons are running as separate java processes. Apache Hadoop runs on UNIX operating system only.

In multi-node Hadoop cluster, each cluster is composed of two parts: Hadoop Distributed File System (HDFS) and Hadoop MapReduce. The Bigdata to be analyzed are stored in the file system known as Hadoop Distributed File System. In other words, HDFS provides storage system for the Bigdata. HDFS is designed to store terabytes or petabytes of data on clusters. It is also designed to become highly fault-tolerance, high throughput, and high capacity distributed file system. The important features of HDFS are its write once-read-many and streaming access models that make HDFS efficient in distributing and processing data, reliably storing large amounts of data, and robustly incorporating heterogeneous hardware and operating system environments[2]. It divides each file into small fixed-size blocks (e.g. 64MB) and stores multiple (default three) copies of each block on cluster nodes.

The different node types that are available in every Hadoop clusters are: Namenode (HDFS master), Jobtracker (MapReduce master), Datanodes (HDFS slave) and Tasktracker (MapReduce slave). In HDFS's master/slave architecture, one node acts as a master node and other nodes as slaves. The master node is known as Namenode which manages the file system namespaces and regulates client accesses to the data. There are a number of worker nodes, called Datanodes, which store actual data in units of blocks. The Namenode maintains a mapping table which maps data blocks to Datanodes in order to process write and read requests from HDFS clients. HDFS allows a secondary Namenode to periodically save a copy of the metadata stored on the Namenode in case of Namenode failure. Jobtracker keep tracks of all the tasks that are running in each Datanodes. Each Datanodes sends a heartbeat signal to Namenode showing that each Datanodes are functioning well. Figure 1 illustrates the HDFS architecture.

Hadoop MapReduce is the programming model that runs on HDFS. MapReduce programming model consists of two functions: map and reduce function. When a MapReduce job is submitted to the cluster, it is divided into map tasks and reduce tasks, where each map task will process one block (e.g., 64 MB) of input data. A Hadoop cluster uses slave (worker) nodes to execute map and reduce tasks. A reduce task uses remote procedure calls to read the intermediate data generated by the map tasks of the job. Each reduce task is responsible for a region (partition) of intermediate data with certain keys. Thus it has to retrieve its partition of data from all worker nodes that have executed the map tasks. This process is called shuffle, which involves many-to-many communications among worker nodes. The reduce task then reads in the intermediate data and invokes the reduce function to produce the final output data (i.e., output key/value pairs) for its reduce partition.

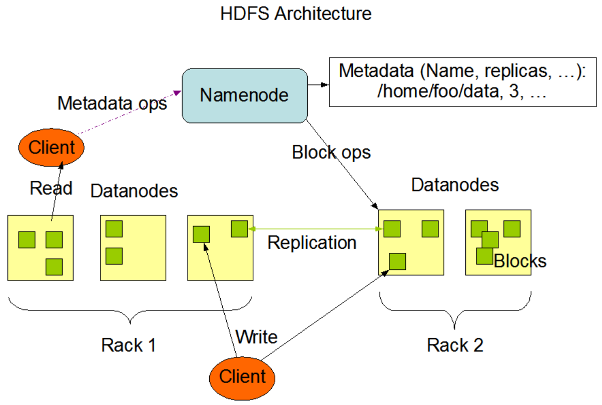

Fig. 1. HDFS architecture

As can be seen by the above discussions about Apache Hadoop, it gives a very good reason to use Hadoop as a tool in developing various applications in bioinformatics. The distributed file system allows us to store huge dataset. On other hand, its programming model will provide an efficient way to design an algorithm according to its map and reduce model.

3. APPLICATIONS

3.1 DRAW

Big data parallel computing frameworks and large-scale distributed file systems are being continuously developed by academic and industrial pioneers to facilitate the high performance runs of data-intensive applications, such as bio-informatics, astronomy, and high-energy physics. Generally, many scientific and engineering applications have interest locality: 1) domain scientists are only interested in a subset of the whole data set, and 2) scientists are likely to access one subset more frequently than others. In matter of mammal's genome data pools, the chimpanzee is usually compared with human. These co-related data have high possibility to be processed as a group by specific domain applications. Here in this algorithm, the "data grouping" is done to represent the possibility of two or more data (e.g., blocks in Hadoop) to be accessed as a group.

The overall data distribution may be balanced by using Hadoop's default data placement strategy, but there is no guarantee that the data accessed as a group is evenly distributed. Thus, in order to process the distributed data in parallel, a MapReduce job is split into many map tasks. Also, to exploit the predictability of data access patterns and improve the performance of distributed file systems, dynamic data grouping is effective. After analysis, the possibility for random data distribution is to evenly distribute the data from the same group. The observation shows that this possibility is affected by three factors: 1) the number of replica for each data block in each rack (NR); 2) the maximum number of simultaneous map tasks on each node (NS); and 3) the data grouping access patterns. Therefore, a new Data-gRouping-AWare data placement scheme (DRAW) algorithm is developed that takes into account the data grouping effects to significantly improve the performance for data-intensive applications with interest locality. DRAW is designed and implemented as a Hadoop-version prototype.

DRAW meaning "Data-gRouping-AWare" is the new data placement scheme for data intensive applications with interest locality. Interest locality is the process of finding out the required particular dataset from the huge big datasets. DRAW is the new algorithm developed using the Hadoop technology for the applications in bioinformatics. DRAW is designed at rack-level, to optimize the grouping distribution inside a rack [5]. In this algorithm, there are three parts: a data access history graph (HDAG), a data grouping matrix (DGM) and an optimal data placement algorithm (ODPA). HDAG is used to exploit the system log files learning the data grouping information. The Namenode in each Hadoop cluster rack maintains system log files which records every system operation, including the files that have been accessed. DGM is used to quantify the grouping weights among the data and generate the optimized data groupings. OPDA is used to form the optimal data placement.

3.2 CloudBLAST

CloudBLAST is the combination of MapReduce and virtualization on distributed resources for bioinformatics application [3]. This is a new approach to parallelization, deployment and management of bioinformatics applications that integrates several emerging technologies for distributed computing. This approach uses MapReduce paradigm to parallelize tools and manage their execution, machine virtualization to encapsulate their execution environments and commonly used data sets into flexibly deployable virtual machines, and network virtualization to connect resources behind firewalls/NATs while preserving the necessary performance and the communication environment.

CloudBLAST approach uses the following techniques: machine virtualization, network virtualization and Hadoop. These three techniques can be combined to deploy important bioinformatics applications based on BLAST on computer clusters on distinct administrative domains connected by a wide-area network (WAN) [4]. The experimental validation is done by deploying a Xen-based virtual cluster across two sites, at the University of Chicago (UC) and the University of Florida (UF), using virtual workspaces for authenticating users and deploying VMs, and ViNe for connecting the nodes behind NATs, over a 200Mbps WAN link. In this approach, Apache Hadoop, an open source implementation is used for MapReduce paradigm to parallelize the execution of NCBI BLAST2, a sequential implementation of BLAST made publicly available by the National Center for Biotechnology Information.

Experimentally, the overheads of the above-mentioned technologies (virtual workspaces and ViNe) are insignificant, demonstrating the feasibility and benefits of executing BLAST on virtualized resources and across sites. CloudBLAST and mpiBLAST showed similar performance with a small advantage to CloudBLAST in a scenario where the target database fits in memory. The CloudBLAST approach is evaluated by comparing its performance with both purely sequential implementations and a leading MPI-based solution (also proposed and deployed for the first time by this work on a non-emulated WAN-based virtual network crossing distinct administrative domains). The evaluation is also done on the overhead introduced by virtualization and Hadoop by comparing the performance of the BLAST application using physical and virtual machines, on LAN and WAN clusters, and with Hadoop and MPI.

3.3 Cloudburst

Cloudburst is an open source tool available as a model for parallelizing algorithms with MapReduce [4]. An enormous amount of sequence data is generated using this next-generation DNA sequencing machines.This tool is a read mapping algorithm optimized for mapping next-generation sequence data to the human genome and other reference genomes, for use in a variety of biological analyses including SNP discovery, genotyping and personal genomics [5]. This tool uses MapReduce-based read mapping algorithm modeled after RMAP, but runs in parallel on multiple machines with Hadoop. Optimization of this algorithm is done for mapping many short reads from next-generation sequencing machines to a reference genome allowing for a user specified number of mismatches or differences.

RMAP is an algorithmic technique called seed-and-extend to accelerate the mapping process. In this algorithm, first find sub-strings called seeds that exactly match in both the reads and the reference sequences, and then extend the shared seeds into longer, inexact alignments using a more sensitive algorithm that allows for mismatches or gaps. These techniques also uses a variety of methods for finding and extending the seeds, and have different features and performance. However, each of these techniques is used for execution on a single computing node, and as such requires a long running time or limits the sensitivity of the alignments they find.

Cloudburst is a read mapping algorithm that indexes the non-overlapping k-mers in the reads as seeds whose size s=m/(k+1) is computed from the minimum length of the reads (m) and the maximum number of differences or mismatches (k). Thus, CloudBurst is considered to be as a new highly sensitive parallel seed and extend read-mapping algorithm optimized for mapping single end next generation sequence data to reference genomes. This tool provides all alignments for each read with up to a user-specified number of differences including both mismatches and indels. The alignments is filtered to find the single best non-ambiguous alignment for each read, and output is identical to RMAPM (RMAP using mismatch scores). As a result, CloudBurst can replace RMAP in a data analysis pipeline without changing the results, but provides much greater performance by using the open-source implementation of the distributed programming framework MapReduce called Hadoop [7].

4. CONCLUSIONS

Apache Hadoop is an open source technology that are easily available in the net. This technology can be used in developing various applications in various fields. This technology is mainly used in bigdata analysis. But this technology is found in bioinformatics applications as it can be shown in the previous sections.The MapReduce paradigm of Hadoop is found to be used widely for develpoing various algorithms of bioinformatics. Some of these algorithms such as DRAW,CloudBLAST,Cloudburst etc., are already disussed in the previous sections.

In futute, the application can be developed using the Hadoop technology with more efficient algorithm in analysing the genomic datasets of Bioinformatics. Analysis of these algorithms can be done by comparing such different bioinformatics-MapReduce-based algorithms. The most recent algorithm developed for bioinformatics applications is DRAW (Data-gRouping-Aware) with the new concept of Interest Locality. In IT industry, the most wanted algorithm is the one which is most efficient, less time consuming and most reliable. It would be very challenging to prove that the performance of Data-gRouping-Aware data placement algorithm to be more efficient comparing with the already existing algorithms like cloudBLAST, cloudburst etc.

5. REFERENCES

[1] “Exploiting High Performance on Bioinformatics Applications in a Cloud System”, Taylor N. Job and Jin H. Park, Proceedings of the World Congress on Engineering and Computer Science 2014 Vol II WCECS 2014, 22-24 October, 2014, San Francisco, USA.

[2] “HOG: Distributed Hadoop MapReduce on the Grid” Chen He, Derek Weitzel, David Swanson, Ying LuComputer Science and Engineering University of Nebraska– Lincoln,Email: fche, dweitzel, dswanson, ylug@cse.unl.edu.

[3] “CloudBLAST: Combining MapReduce and Virtualization on Distributed Resources for Bioinformatics Applications”, Andréa Matsunaga, Maurício Tsugawa and José Fortes Advanced Computing and Information Systems Laboratory Department of Electrical and Computer Engineering, University of Florida PO Box 116200, Gainesville, FL, 32611-6200, USA{ammatsun, tsugawa, fortes}@ufl.edu. Fourth IEEE International Conference on eScience.

[4] “CloudBurst: highly sensitive read mapping with MapReduce”, Michael C. Schatz∗ Center for Bioinformatics and Computational Biology, University of Maryland, College Park MD 20742, USA.

[5] “Using Bioinformatics Applications on the Cloud”, Hyungro Lee School of Informatics and Computing, Indiana University 815 E 10th St.Bloomington, IN 47408 lee212@indiana.edu.

[6] “DRAW: A New Data-gRouping-AWare Data Placement Scheme for Data Intensive Applications With Interest Locality” Jun Wang, Qiangju Xiao, Jiangling Yin, and Pengju Shang. Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32826 USA, IEEE TRANSACTIONS ON MAGNETICS, VOL. 49, NO. 6, JUNE 2013.

[7] https://Hadoop.apache.org/