IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 14, Issue 6 (Sep. - Oct. 2013), PP 89-93
www.iosrjournals.org

Paralyzing Bioinformatics Applications Using Conducive Hadoop Cluster

Bincy P Andrews[1], Binu A[2]
[1]*(Rajagiri School of Engineering and Technology*, Kochi)
[2] *(Rajagiri School of Engineering and Technology*, Kochi*)*

Abstract : *Bioinformatics may be defined as the application of computer science to molecular biology in the form of statistics and analytics. The bioinformatics applications deal with bulk amount of data. Researchers are now facing problems with the analysis of such ultra large-scale data sets, a problem that will only increase at an alarming rate in coming years. More over big challenge is involved in processing, storing and analyzing these peta bytes of data without causing much delay. Most of the bioinformatics algorithms are sequential thus making situation rather worse. This implies that data manipulations by means of uniprocessor systems are impractical. However most of the biological problems have parallel nature. Hence a practical and effective approach involves the usage of parallel clusters of workstations. Hadoop can be used to tackle this class of problems with good performance and scalability. This technology could be the basis of a computational parallel platform for several problems in the context of bioinformatics applications. Normally, Hadoop is deployed over high performance computing systems which are expensive involving complex deployment scenarios that only big enterprises are able to make it possible. So for smaller research organizations where cost is an important factor cannot choose systems with high computational capabilities for cluster set up. Rocks cluster is a viable solution in such scenarios. Rocks Cluster Distribution originally called NPACI Rocks is a Linux distribution intended for high-performance computing clusters. This paper implements a cost-effective cluster for paralyzing bioinformatics applications by deploying Hadoop over rock cluster and Emphasizes on the usage of commodity clusters for paralyzing bioinformatics applications by providing necessary justifications. Results show that paralyzing bioinformatics application saves much time compared to stand alone mode of execution effectively under optimal cost considerations.*

Keywords :*Hadoop, Clusters, Rocks Cluster, commodity clusters, cluster environment, big data, bioinformatics*

I. INTRODUCTION

Bioinformatics may be defined as the application of computer science to molecular biology in the form of statistics and analytics. Some of the application areas of bioinformatics include:

- molecular medicine
- personalized medicine
- preventative medicine
- gene therapy
- drug development
- waste cleanup
- climate change studies
- alternative energy sources
- biotechnology
- antibiotic resistance
- forensic analysis of microbes
- bio-weapon creation
- crop improvement
- insect resistance
- vetinary sciences

Most of these applications deal with bulk amount of data. Bioinformatics researchers are now facing problems with the analysis of such ultra large-scale data sets, a problem that will only increase at an alarming rate in coming years. More over big challenge is involved in processing, storing and analyzing data these peta bytes of data without causing much delay. Also most of the bioinformatics algorithms are sequential thus making situation rather worse. This implies that data manipulations by means of uniprocessor systems are impractical. However most of the biological problems have parallel nature. Hence a practical and effective approach involves the usage of parallel clusters of workstations. The various advantages of using parallel clusters are:

- Save time

- Save money
- Solve larger &complex problems quickly
- Provide concurrency
- Use of non local resources

Hadoop can be used to tackle such class of problems with good performance and scalability. This technology could be the basis of a computational parallel platform for several problems in the context of bioinformatics applications. The Hadoop platform was designed to solve problems where there is lot of data that doesn't fit nicely into tables because of its complex and unstructured nature. Hadoop project and associated software provide a foundation for scaling to peta byte scale data warehouses using clusters, providing fault-tolerant parallelized analysis on such data using a programming style named MapReduce. Thus Hadoop will definitely be a promising solution for bioinformatics applications. Normally, Hadoop is deployed over high performance computing systems which are expensive that only big enterprises are able to make it possible. Moreover such deployment scenarios are complex making it infeasible for smaller organizations to handle. . In normal scenarios a cluster set up is highly influenced by following factors like:

- Cost: Normally cost is very high for systems with much computational power that are used to set up a cluster.
- Power: for systems with much computational capabilities power requirements is also high.
- Temperature: usually air conditioned atmosphere is required for maintaining a cluster.

All these above mentioned factors never get along. That is if we are choosing systems with high computational capabilities cost will get elevated and also temperature requirement will have to be taken into consideration. So for smaller research organizations where cost is an important factor cannot choose systems with high computational capabilities for cluster set up. This is where rocks cluster comes in. If we are using rocks it is not necessary to meet each of these factors. For setting up of rocks high performance systems are not required there by cost factor reduces considerably. Also temperature is not a big constrain. Rocks Cluster Distribution originally called NPACI Rocks is a Linux distribution intended for high-performance computing clusters. Rocks was initially based on the Red Hat Linux distribution, however modern versions of Rocks are now based on CentOS that simplifies mass installation onto many computers. Rocks include many tools such as MPI which are not part of CentOS but are integral components that make a group of computers into a cluster. Most important feature or rocks is that it doesn't need high performance computing nodes for deployment. More over rocks contain rolls that can be used for bioinformatics applications and its main advantage is that all these tools get automatically installed during installation of operating system and no further configuration of tools are necessary. Rocks cluster has got following features:

- Easy deployment of cluster
- Addition & deletion of new host can be easily accomplished
- Rocks automatically edit most of the files that is used to identify its hosts.
- Password-less SSH
- No additional login to compute nodes can be performed due password less SSH.
- Rich set of rolls that automatically get configured during rock cluster installation.

This paper implements a cost-effective cluster for paralyzing bioinformatics applications by deploying Hadoop over rock cluster and Emphasizes on the usage of commodity clusters for paralyzing bioinformatics applications by providing necessary justifications. Old, low cost - low performance, single core machines were selected as the compute nodes for the cluster. Cost factor affected both master node & for Cisco switch. Job submissions were done using some of the existing systems in the campus network which acts as user nodes there by reducing additional cost or bandwidth. Low cost compute nodes had only low power requirement and low power backup. After cluster set up testing of bioinformatics tool fasta in both stand alone & parallel environment were done. Results show that paralyzing bioinformatics application saves much time compared to stand alone mode of execution. Remaining chapters are organized as follows: chapter II provides as overview of related works and chapter III provides detailed system design and implementation followed by its performance evaluation in chapter IV. Finally paper concludes with chapter V.

II. BACKGROUND

Parallelism can be achieved by either of these methods:

- Develop or adapt existing software to distribute and manage parallel jobs, or
- Modify existing applications to make use of libraries that facilitate distributed programming such as MPI, OpenMP, RPC and RMI.

Effort required for first approach is recurrent since new versions of the original sequential code may render the parallelized application obsolete. So created versions may lag and lack in features of the latest sequential tool version. Also it is very difficult to incorporate failure handling mechanisms in such systems. First approach is cost effective compared to other: [9] [10] [11] are a small sample of Grid-based solutions for bioinformatics applications that automate the process of transferring or sharing large files, selecting appropriate application binaries for a variety of environments or existing services, managing the creation and submission of a large number of jobs to be executed in parallel and recovering from possible failures. Also cloud-blast paralyses bioinformatics application using MapReduce framework. Most of These works deals with execution of a single file at a time. Processing of large no of files at same time is not being addressed, which is often necessary if amount of data to be dealt with is more. More over most of these works concentrate on paralyzing bioinformatics applications using blast. BLAST (Basic Local Alignment and Search Tool) is the most widely used sequence alignment tool. ClustalW, HMMER, FASTA, Glimmer etc. are similar tools. FASTA and BLAST have the same goal: to identify statistically significant sequence similarity that can be used to infer homology. The FASTA programs offer several advantages over BLAST:

- Rigorous algorithms unavailable in BLAST (Table I). Smith-Waterman (ssearch36), global: global (ggsearch36), and global: local (glsearch36) programs are available, and these programs can be used with psiblast PSSM profiles.
- Better translated alignments. fastx36, fasty36, tfastx36, and tfastx36 allow frame-shifts in alignments; frame-shifts are treated like gap-penalties, alignments tend to be longer in error-prone reads.
- Better statistics. BLAST calculates very accurate statistics for protein: protein alignments, but its model-based strategy is less robust for translated-DNA: protein and DNA: DNA scores.
- FASTA uses an empirical estimation strategy, and now provides both search-based, and high-scoring shuffle-based statistics (-z 21).
- More flexible library sequence formats. The FASTA programs can read FASTA, NCBI/ formatdb, and several other sequence formats, and can directly query MySQL and Postgres databases. The programs offer several strategies for specifying subsets of databases.
- A very efficient threaded implementation. The FASTA programs are fully threaded; both similarity scores and alignments can be calculated in parallel on multi-core hardware. On multi-core machines, FASTA can be faster than BLAST while producing better alignments with more accurate statistical estimates.

Considering such advantages this paper adopts FASTA for paralyzing bioinformatics applications. More over in most of the paralyzing approaches using Hadoop incurs much cost & also involves complex mechanisms for cluster setup. In order to reduce both cost & complexity rock cluster is used for cluster set up. Rock is an operating system which helps in easy deployment of cluster.

III. SYSTEM DESIGN AND IMPLEMENTATION

The system components and modules are shown in Fig 4.1.

3.1 Public network

Public network consists of user nodes used for job submission. Systems present in the campus network were chosen as user nodes there by reducing further costs.

3.2 Private network

Private network consists of low cost low performance single core systems. They are very cheap. They are connected to front end node by means of Ethernet switch. Once job is submitted by master node or frontend node to each of these compute nodes they process it and generates results.

3.3 Frontend node

It is a high performance multiprocessor system that accepts job request from user nodes and assign it to compute nodes for processing. It also consists of several tools for cluster monitoring (ganglia), bioinformatics applications, resource management etc.

3.4 Ethernet switch

It is a 300 series 28 port Cisco Ethernet switch. It restricts traffic in the private network formed by the frontend node and compute nodes. The identification of compute nodes in the cluster is accomplished by means of DHCP requests.

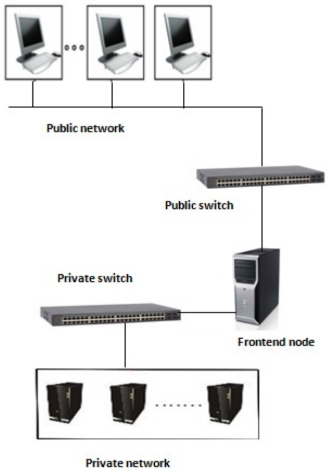

Figure 3.1: system components and modules

3.5 Hadoop Distributed File system

The Hadoop Distributed File System (HDFS) is a subproject of the Apache Hadoop project. It is a distributed, highly fault-tolerant file system designed to run on low-cost commodity hardware. HDFS provides high-throughput access to application data and is suitable for applications with large data sets. Both input & output data are stored in Hadoop distributed file system.

3.6 Bioinformatics module

This is the most important module. In order to paralyze fasta MapReduce program should have following:

- fasta input files handler
- Mapper
- Main program

Since fasta doesn’t require a Reducer to handle the intermediate map output, Reducer is not necessary. Input to MapReduce program is a set of fasta files. Initially all input files will be uploaded to HDFS. By default fasta programs like fasta36, ssearch36 etc are installed automatically during rocks installation. Since it is present on both frontend node as well as compute nodes it should not be uploaded to HDFS. Input file handler collects all uploaded input files from HDFS to create key-value pair which will be passed to map function. The Mapper mainly creates a java process to call the fasta program. The Map key is the filename, and the Map value includes the full HDFS path for each uploaded input files. Each map task downloads the assigned input file from HDFS, and passes this input to run the fasta program. Finally output file will be created and will be uploaded to HDFS. The entire process is depicted in figure 3.2. Here there are n files say F1, F2…FN. Each of which in turn contains n fasta formatted sequences say $s=\{s_1,s_2\ldots s_n\}$. Each of these n files containing n sequences will be assigned to n workers. Now workers follow MapReduce paradigm, process each input files and writes corresponding results back to HDFS. Output files are represented by O1, O2...ON.

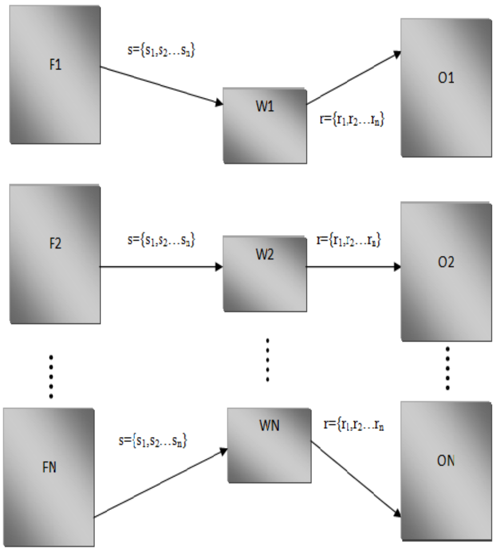
F1
$s=\{s_1,s_2\ldots s_n\}$
W1
$r=\{r_1,r_2\ldots r_n\}$
O1
F2
$s=\{s_1,s_2\ldots s_n\}$
W2
$r=\{r_1,r_2\ldots r_n\}$
O2
FN
$s=\{s_1,s_2\ldots s_n\}$
WN
$r=\{r_1,r_2\ldots r_n$
ON

Figure 3.2: processing of MapReduce task

IV. PERFORMANCE EVALUATION

The evaluation of system performance was done by system benchmarking and comparing communication overheads.

4.1 Experimental Setup

Experimental setup consists of master node connected to twelve compute nodes by means of a Cisco SG 300 switch. Rocks cluster version 6.1. Emerald Boa was installed on the master node with bioinformatics Roll. The MAC address of all the compute nodes were detected using a command in Rocks called "insert-ethers". The compute nodes are assigned with private IP address automatically by rocks depending on frontend nodes private ip range. The MAC address – IP address mapping is also done by the rocks cluster operating system and corresponding details will be stored in its internal database. Now all identified compute nodes are installed with Rocks kernel via PXE boot. Upon completion of cluster setup the Hadoop file system is deployed. Test was done to compare time requirement for running bioinformatics tool in parallel as well as in standalone environment. Input set constitutes a folder consisting of fasta formatted files of 68kb each. Time taken for execution in parallel environment was calculated by varying number of input files. Finally results were tabulated. The performance Analysis tests were conducted using UnixBench and NetPipe tools.

4.2 Experiment Results

TABLE 4.1 shows time taken for executing fasta in sequential as well as in parallel mode. We can see that time taken for execution in parallel environment is much less compared to sequential execution of fasta. Number of input files was varied for each execution in parallel environment.

No of files	Time taken in sequential mode (seconds)	Time taken in cluster environment (seconds)
2	1minute 14 seconds	26 seconds
4	2minutes 28 seconds	42 seconds
8	4minutes 56 seconds	59 seconds
16	9minutes 12seconds	1minute 40 seconds
32	18minutes 24seconds	3minutes 20 seconds

Table 4.1: Time taken

Thus we can conclude that paralyzing bioinformatics application is inevitable and proposed system is a viable solution. Also proposed system requires only minimal cluster setup cost, minimal energy consumption irrespective of temperature requirement. Thus proposed system meets its objective efficiently.

V. CONCLUSION

This paper, proposes implementation of a Hadoop cluster for paralyzing bioinformatics application and its performance analysis based on cost effectiveness. The implementation is done using low cost commodity machines which can minimize the cost to a large extent. Most of the existing approaches involve high performance systems for cluster setup. But when the cost aspect is considered, the high performance machine involves a large cost and usage of such systems for cluster set up is not feasible in a small university or research organization. Considering the performance verses cost effectiveness, proposed commodity cluster model for paralyzing bioinformatics application is an adaptable approach for small research organization or university.

Acknowledgements

We are greatly indebted to the college management and the faculty members for providing necessary facilities and hardware along with timely guidance and suggestions for implementing this work.

REFERENCES

[1] Rocks Cluster Installation Memo, Hiroyuki Mishima, DDS, Ph.D., Research Fellow, and Jun Ni, Ph.D. Associate Professor Medical Imaging HPC & Informatics Lab Department of Radiology, College of Medicine, University of IowIowaCity, IA 52242, USA

[2] Rocks cluster emerald boa 6.1 User Guide –ebook

[3] Hadoopin Practice –ebook

[4] http://www.hadoop.apache.org

[5] http://www.rocksclusters.org/

[6] http://www.ibm.com/developerworks/library/l-hadoop-1/

[7] http://ankitasblogger.blogspot.in/2011/01/hadoop-cluster-setup.html

[8] http://icanhadoop.blogspot.in/2012/09/configuring-hadoop-is-very-if-you-just.html

[9] Arun Krishnan, "GridBLAST: a Globus-based high-throughput implementation of BLAST in a Grid computing framework," Concurrency and Computation, v.17, Issue 13, pp. 1607-1623, 2005, doi:10.1002/cpe.v17:13.

[10] H. Stockinger, M. Pagni, L. Cerutti, L. Falquet, "Grid Approach to Proc of the 2nd IEEE Intl. Conf. on e-Science and Grid Computing, 2006, doi:10.1109/E-SCIENCE.2006.70.

[11] J. Andrade, M. Andersen, L. Berglund, and J. Odeberg, "Applications of Grid Computing in Genetics and Proteomics," LNCS, 2007, doi:10.1007/978-3-540-75755-9.