Enhancing Hadoop MapReduce Performance for Scientific Data using NoSQL Database

Hamoud Alshammari, Hassan Bajwa, Jeongkyu Lee
Department of Computer Science and Engineering
University of Bridgeport, Bridgeport, CT

Abstract

Scientific data sets usually have similar jobs that are frequently applied to the data by different users. In addition, many of these data sets are unstructured, complex, and required fast and simple processing. In order to increase the performance of the existing Hadoop and MapReduce algorithm, it is necessary to develop an algorithm based on the type of data sets and requirements of the jobs.

In this poster, we represent a Hadoop MapReduce environment that uses genomic and biological data as an example of unstructured and complex data.

Introduction

Scientific data and applications usually require large complex amounts of data processing and computational capabilities [1]. Also, these data has different format that the scientists work on. However, the scientists usually manipulate some specific kind of jobs on these data. One of these data is the genome data set, which consists of 24 files each file is about the data of one chromosome [2].

One of the popular jobs that usually scientists process on these data is finding a given sequence on either a specific chromosome or on all of them [3]. This process usually takes a long time doing that buy the traditional ways. So, processing this kind of data using Hadoop and its ecosystem gives better results.

NoSQL database gives more efficiency than using the native Hadoop because of the ability of indexing search process, which reduce the time of searching on a database. NoSQL database is considered as a structured database since it has indexing system to reduce the access time processing.

Current Hadoop MapReduce

Shown in (Figure1) Client A and Client B are searching for similar sequence in BigData. Once Client A finds the sequence, Clint B will also go through the whole BigData again to find the same results. Since each job is independent, clients do not share results between each other neither get benefits from the metadata of each other [4].

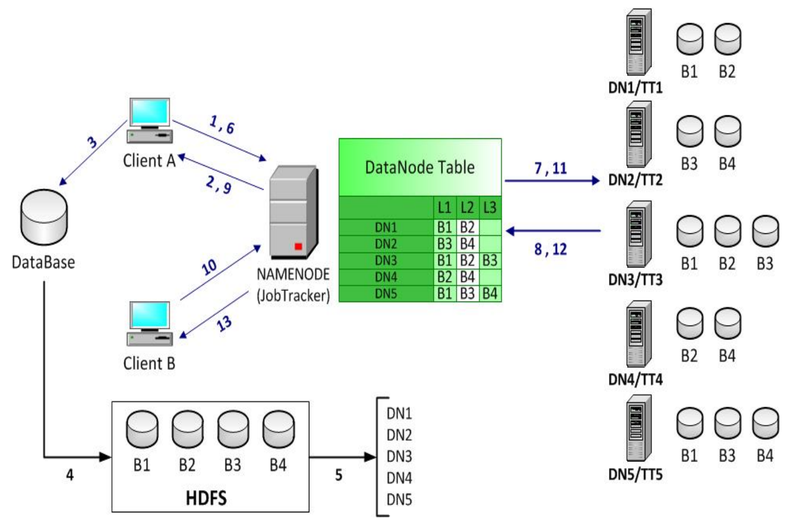

Figure1: Current Hadoop MapReduce Architecture

Problem Definition

Native Hadoop MapReduce algorithm (Figure1) represents a good solution environment for many jobs that are applied generally on a big data. However, native Hadoop performs the jobs on the same way every time, so any job has almost a same performance each time this job gets processed.

In current Hadoop, any client is looking for a super sequence with sub-sequence already has been searched will have to go through the BigData again. Thus the cost to perform the same job will stay the same each time.

Enhanced Hadoop MapReduce

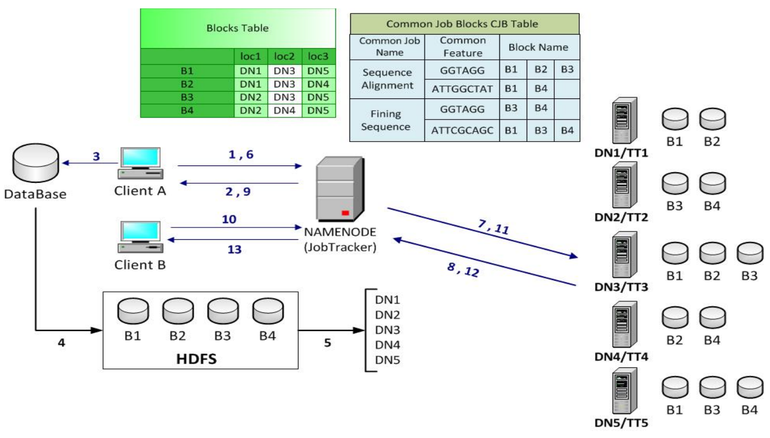

Figure2: Enhanced Hadoop MapReduce Architecture

The Enhanced Hadoop MapReduce Architecture shown in (Figure2) shares the main concept of the current Hadoop but try to increase the performance of Hadoop MapReduce by applying a Common Job Block table (CJBT) [4].

In our work we determined the specific files that carry the data. However, in some examples both ideas end up with the same results because of the size of data and the number of source files.

HBase As NoSQL Database

Common Feature	Column Family: Finding_Sequence		
	CF: A	CF: B	…
GGGGCGGGG	Chromosome1	Chromosome2	
GAATGACTC	Chromosome7	Chromosome11	
TTTAGCC	Chromosome3	Chromosome7	

Table1: DNAIndex Table; Example of HBase table components for a specific Job of specific data set.

Table1 shows that it stores a sample of a metadata of the contents of the source files that join between the common feature (DNA sequence) and the file names inside the source data files. Each job has its own table on the database. So, we might have as many column families – databases – as number of different data sets, and we might have as many tables as number of common jobs inside a single database.

Result and Conclusion

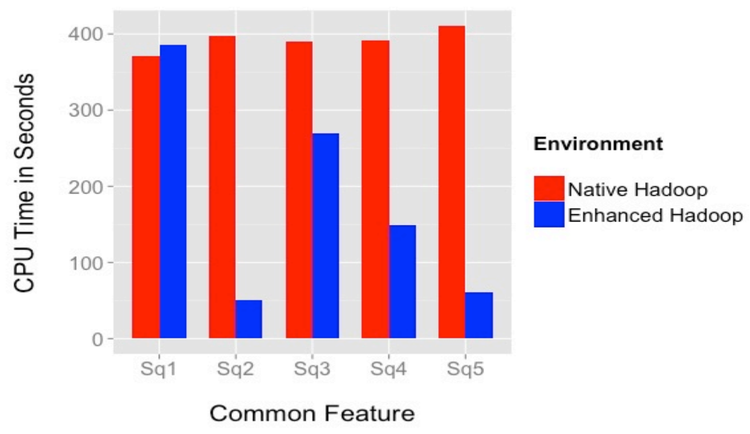

Figure3: Overall performance of selected queries in native Hadoop and enhance Hadoop.

Some compressions have been shown in above, and Sq1 to Sq5 are sequences in DNA (e.g. Sq3 is CATTTCTGCTAAGA). We can see the differences between the two Hadoop versions, and that show the developed performance in enhanced Hadoop comparing with the native one.

References

[1] T. White, *Hadoop: The definitive guide*: " O'Reilly Media, Inc.", 2012.
[2] S. Leo, F. Santoni, and G. Zanetti, "Biodoop: Bioinformatics on Hadoop," in *Parallel Processing Workshops, 2009. ICPPW '09. International Conference on*, 2009, pp. 415-422.
[3] H. Mathkour and M. Ahmad, "A comprehensive survey on genome sequence analysis," in *Bioinformatics and Biomedical Technology (ICBBT), 2010 International Conference on*, 2010, pp. 14-18.
[4] H. Alshammari, H. Bajwa, and L. Jeongkyu, "Hadoop based enhanced cloud architecture for bioinformatics algorithms," in *Systems, Applications and Technology Conference (LISAT),* 2014 IEEE Long Island, 2014, pp. 1-5.