IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

To implement CBPGA for Simplified Large Data on Hadoop Map Reduce

Shalini Jain[1], Prof. Saurabh Kapoor[2]

Research Scholar, Computer Science & Engineering Dept, Gyan Ganga Institute of Technology & Sciences, Jabalpur, (M.P.), India[1]

Asst. Prof., Computer Science & Engineering Department, Gyan Ganga Institute of Technology & Sciences, Jabalpur, (M.P.), India[2]

Abstract: This paper introduces implementation CBGPA (Cluster Based Parallel Genetic Algo) [1] for simplified large data on Hadoop Map Reduce. Hadoop is a framework used for processing large amount of data in a parallel and distributed manner .Its provides the reliability in storing the data and efficient processing system. The two main gears of Hadoop are the HDFS (Hadoop Distributed File System) and Map Reducing (for processing). Map Reduce is a programming model which enables parallel processing in a distributed environment. Classify similar objects under the same group called cluster. Metaheuristic techniques, such as Genetic Algorithms (GAs) [2], It is one important data mining methods constitute the best alternative to find near-optimal solutions for such problems within a reasonable execution time and limited resources. To improve efficiency better approach is used called Map Reduce for Parallelization Genetic Algo(MRPGA)[3][4] by using the features of Hadoop. An analysis of proposed Algo CBPGA to evaluate performance gains with respect to the current algo MapRedue Word Count [5]. Our proposal aim is to evaluate both the parallel algo are compared based on speedup the no of processing node on different size of text files and find the solution within a reasonable time. Parallel implementation of the CBPGA algorithm makes the algorithm faster and scalable in order to find the optimal solutions while working with large data cluster in a parallel manner.

Index Terms: Big Data, word count, hadoop, mapreduce, cluster ,Parallel Genetic Algorithm.

1. INTRODUCTION

Big data is a term used to address data sets of large sizes. Such data sets are beyond the possibility to manage and process within tolerable elapsed time. For such a scenario parallelization is a better approach .Hadoop Map reduce[6] is a parallel programming technique build on the frameworks of Google app engine map reduce. It is used for processing large data in a distributed environment. It is highly scalable and can be build using commodity hardware. Hadoop map reduce splits the input data into particular sized chunks and processes these chunks simultaneously over the cluster. It thus reduces the time complexity for solving the problem by distributing the processing among the cluster nodes fig 1.

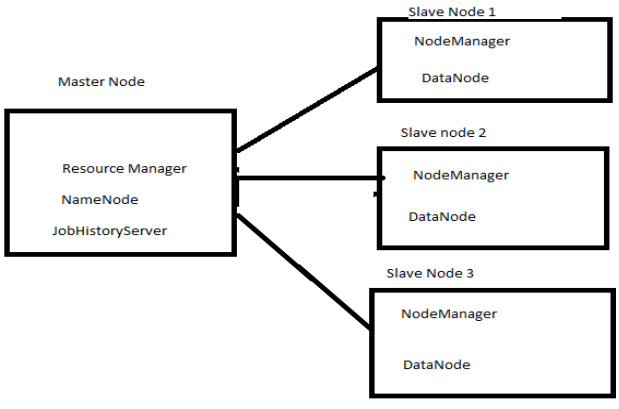

Figure 1 Data Cluster on Hadoop

2. HADOOP MAPREDUCE WORDCOUNT

Hadoop mapreduce is a programming model which uses the map reduce paradigm for processing. It is inspired by the Google app engine mapreduce. It allows for huge scalability by using commodity hardware. Mapreduce uses HDFS (hadoop distributed file system) which is another component of hadoop framework for storing and retrieval of data. The processing time is reduced by splitting the data set into blocks depending upon the block size. The block size is usually 64mb or 128mb. This split data is then processed parallel over the cluster nodes. Mapreduce thus provides a distributed approach to solve complex and lengthy problems.

2.1 MAPREDUCE PARADIGM

Mapreduce programming paradigm involves distributed processing of large data cluster[7] over the cluster. Under this paradigm the input data is spitted according to the block size. The data split is performed by the input format. These splits are assigned a specific key by the record reader and thus a key, value pair is generated. Key, value pairs are then subjected to a two phase processing.

This two phase processing comprises of a map phase and a reduce phase. The architecture of a basic map reduce paradigm is depicted in figure 2.

The map phase is composed of a mapper or a map(). Map phase is executed in the mapper of each node.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

The reduce phase is composed of a reducer or a reduce().After receiving the mapped results reducer performs the summary operations to generate final result.

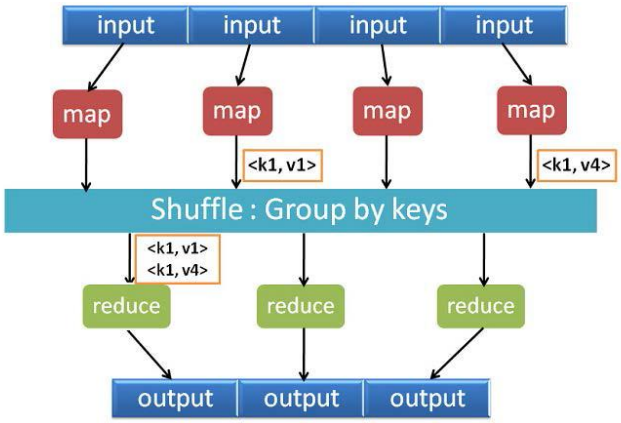

Figure2 MapReduce Data Flow

MAP PHASE:
- The mapper receives the key-value pairs generated by the record reader.
- The mapper performs the distributed algorithm to process the key-value pairs and generates the mapping results in form of intermediate key-value pairs .
- The intermediate key-value pairs are then passed on to the reducer.

REDUCE PHASE:
- The mapped results of the mapper are shuffled.
- The shuffled results are then passed on to the appropriate reducer for further processing.
- Combined output of all the reducers serves as the final result.

2.2Mapreduce word count
Map Reduce WordCount [8] reads text files and counts the occurrence of each word. The input is text files and the output is also text files, each line of which contains a word and the count of how often it occurred, separated by a tab. way of running the WordCount program, used here, is compiling the .java file and creating its jar file and then executing the program. The program includes Mapper and Reducer interfaces to provide the map and reduce tasks.

3. EXISTING GENETIC ALGORITHMS

Genetic Algorithm is a nature inspired heuristic approach used for solving search based and optimization problems. It belongs to a class of evolutionary algorithms [9]. In GAs we evolve a population of candidate solutions towards an optimal solution. GA simulates nature based techniques of crossover, mutation, selection and inheritance to get to an optimal solution. Under GA we implement the law of survival of the fittest to optimize the candidate solutions The technique of GA progresses in the following manner:

1. Initial population of candidate solutions is created
2. Each individual from the population is assigned a fitness value using appropriate fitness function
3. Parents are selected by evaluating the fitness
4. Offspring are created using reproduction operators i.e. crossover, mutation and selection on parents
5. New population is created by selecting offspring based on fitness evaluation
6. Steps 3,4,5 are repeated until a termination condition is met

Generally genetic algorithm will find good solutions in reasonable amount of time, but increases in time to find solutions if they are applied to harder and bigger problems. To overcome this problem we will go for parallel implementation of genetic algorithm.

4. PARALLEL GENETIC ALGORITHMS

In the following sections we discuss some strategies commonly used for parallelizing GA [10]. Then, we propose a customized approach to implement Clustering based parallel GA on hadoop map reduce.

4.1 Parallel implementations
Parallel implementation of GA is realized using two commonly used models as:
- Coarse-grained parallel GA
- Fine-grained parallel GA

The PGA consists of multiple computing nodes, those depends on type of PGA used. There are 4 major types of PGA's, they are master-slave, coarse-grained, fine-grained & hierarchical hybrids.

1 Master-Slave GA In Master-Slave GA, one computing node will be the master and the others are slaves. The master node is responsible to hold the population and performs most of the genetic algorithm operations. The master will assign one or more computing intensive tasks to slaves by sending one or more chromosomes to them and it would then wait for the slaves to return their results.
2 Coarse-Grained GA In Coarse-Grained, the population is divided into computing nodes which have a sub-population and executes genetic algorithm on its own. The nodes will exchange chromosomes with each other ensuring that good solutions can be spread to other nodes. This exchange can be called as migration where a node sends its best chromosome to other nodes. The other nodes which are having the worst chromosomes will be replaced by the received ones.
3. Fine-Grained GA In Fine-Grained, each computing node only has a single chromosome and is arranged in a spatial structure. Here each node communicates only with other neighboring nodes and the population is the collection of all the chromosomes in each node. To execute a genetic operation, a computing node must interact with its neighbors. The good traits of a superior individual can be spread to the entire population due to the overlapping of neighborhoods.
4 Hierarchical hybrids This is the final PGA type which is structured in two levels. It operates as a coarse-grained in higher level and as a fine-grained in lower level. Among the four types of PGA's, the fine-grained genetic

algorithm has the highest level of parallelism and also has a large communication overhead because of high frequency of interactions between neighboring nodes.

4.2 Proposed Parallel Genetic Algorithm The coarse-grained genetic algorithm [11] has been chosen for proposed algorithm. In this PGA implementation, all the computing nodes randomly create their own subpopulation and each of them will execute genetic algorithm on its own. One of the computing nodes will be assigned special task to gather results from all the other nodes and then choose the best result as the output of parallel genetic algorithm. This node is called as the collector node. In proposed algorithm, each chromosome is encoded as a series of node id's that are in the path from source to destination. The first gene in the chromosome is always the source and the last gene in the chromosome is always the destination. Since different paths may have different number of intermediate nodes, the chromosomes will be of variable length. However, the maximum length of a chromosome cannot exceed the total number of nodes in the network. Any repeated nodes in the chromosome signify that the path represented by the chromosome contains a loop and in network routing, any loop should be eliminated.

5. PROPOSED WORK

Implementation of CBPGA USING HADOOP MAPREDUCE

In this sub section we propose the format of GA we used for clustering based problems. Along with this we discuss our customized approach to exploit Coarse-grained parallel GA model. This approach successfully implements GA based clustering on hadoop map reduce. Crux of this approach lies in performing a two phased clustering in mapper and then, in the reducer. To begin, the input data set is split according to the block size by the input format. Each split is given to a mapper to perform the First phase clustering. The first phase mapping results of each mapper are passed on to a single reducer to perform the Second phase mapper. We thus, are using multiple mappers and a single reducer to implement our clustering based parallel GA.

PGA implementation on Hadoop MapReduce

The main techniques used to parallelize the proposed GA using MapRedue programming model are (Geronimo et al., 2012):

- Each iteration of the GA is treated as distinct MapReduce job
- Multiple Map functions are invoked from multiple distributed nodes attached to the Hadoop cluster to parallelize the chromosome fitness evaluation
- A single Reduce function is invoked to collect the output of all Map functions and run all the genetic functions such as crossover, mutation, survival selection and parents selection which are required to generate a new generation of population

The proposed implemented PGA on MapReduce model has the following modules (Geronimo et al., 2012):

- A Parallel Genetic Algorithm
- A Master node
- A number of Mapper nodes and a Reducer nodes
- Input Format and Output Format: splits the data for inputs to the multiple Map functions and stores output of the Reduce function to Hadoop distributed file system

The proposed algorithm was evaluated with respect to the execution time and branch coverage (Geronimo et al., 2012). The execution time is calculated using system clock and the total time. The total time comprises of the following complements:

- InitTime is the total time needed for the Parallel Genetic Algorithm to initialize a Map function with the required data (such as SUT instrumented bytecode, JUnit, test cases)[12]. This information is required to run the fitness evaluation in every iteration
- EvalTime is the total time taken to evaluate the fitness of chromosomes

6. CONCLUSION AND FUTURE WORKS

In this paper one of the basic programs of Hadoop, that is, MapReduce WordCount is executed in a large cluster. The changes in the size of input files and the number of reduce tasks affecting the execution time of the program is studied.

The execution results showed that the time needed by a WordCount program for execution increases as the size of the input files is increased. It is observed that although increasing the size causes an increase in the execution time of the program, but large number of small files takes longer to execute as compared to a single larger file. It is also observed that the increase in time is not proportional and it decreases as the files are increased in size. Also, an increase in the number of reducers causes an increase in the time taken for the completion of WordCount program.

So, we compared it with the parallel genetic algorithm (PGA) evolving for Hadoop Map Reduce. The progress shows that, by using the parallel genetic algorithm the performance of GA operators are effective. Parallel GAs is well suited for the large size of data sets. The reason behind the parallel GAs are efficiently and reliability for solving a problem in a polynomial time in a parallel manner.

This paper aims at comparing the execution time of WordCount under varying conditions. The execution time may vary depend up on the different size of text files and no. of nodes. the effective structured system lead to the retrieval of data in minimum time. On the whole, the configuration of the Hadoop is very important when there is a need to improve the performance.

REFERENCES

[1] Big Data Clustering Using Genetic Algorithm CBGPA On Hadoop Mapreduce Nivranshu Hans, Sana Mahajan, SN Omkar INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 4, ISSUE 04, APRIL 2015 ISSN 2277-8616.

[2] A Review on Genetic Algorithm Practice in Hadoop MapReduce Mrs. C. Sunitha , Ms. I. IJSTE - International Journal of Science Technology & Engineering | Volume 2 | Issue 5 | November 2015 ISSN (online): 2349-784X.

[3] MRPGA: An Extension of MapReduce for Parallelizing Genetic Algorithms - Chao Jin, Christian Vecchiola and Rajkumar Buyya.

[4] C. Jin, C. Vecchiola, and R. Buyya, "Mrpga: an extension of mapreduce for parallelizing genetic algorithms", in eScience, 2008. eScience '08. IEEE Fourth International Conference on, IEEE, 2008, pp. 214–221.

[5] Parallelization of Genetic Algorithms using MapReduce Suman Saha European Journal of Applied Social Sciences Research (EJASSR) Vol-2, Issue 1 www.ejassr.org Jan-Mar 2014.

[5] International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 5, May 2015 Copyright to IJARCCE DOI 10.17148/IJARCCE.2015.4542 184 Analysis of Research Data using MapReduce Word Count Algorithm Manisha Sahane1, Sanjay Sirsat2, Razaullah Khan3.

[6] International Journal of Advanced Research in Computer Science and Software Engineering Research Paper Analysis of Bidgata using Apache Hadoop and Map Reduce Mrigank Mridul, Akashdeep Khajuria, Snehasish Dutta, Kumar N.

[7] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51, no. 1 (2008): 107-113.

[8] International Journal of Scientific Development and Research (IJSDR) www.ijsdr.org 130 HADOOP MAPREDUCE - WORDCOUNT IMPLEMENTATION 1P. Deepika, 2Prof. G. R. Ananatha Raman ISSN: 2455-2631 © March 2016 IJSDR | Volume 1, Issue 3 IJSDR1603025

[9] Scaling Genetic Algorithms using MapReduce - Abhishek Verma, XavierLlor'a, David E. Goldberg, Roy H. Campbell.

[10] Scaling Populations of a Genetic Algorithm for Job Shop Scheduling Problems using MapReduce - Di-Wei Huang, Jimmy Lin.

[11] A Framework for Genetic Algorithms Based on Hadoop Filomena Ferrucci_, M-Tahar Kechadi†, Pasquale Salza_, Federica Sarro‡ arXiv:1312.0086v2 [cs.NE] 15 Dec 2013.

[12] L. Di Geronimo, F. Ferrucci, A. Murolo, and F. Sarro,"A parallel genetic algorithm based on hadoop mapreducefor the automatic generation of junit test suites",in Software Testing, Verification and Validation (ICST),2012 IEEE Fifth International Conference on, IEEE,2012, pp. 785–793.