To Comparison & Implementation of GA with CBPGA for Simplified Large Dataset on Hadoop Map Reduce

Shalini Jain[1], Prof. Saurabh Kapoor[2], Prof. Ashok Verma[3]

[1] *Research Scholar, Computer Science & Engineering Department*

[2] *Asst. Prof., Computer Science & Engineering Department*

[3] *Assoc. Prof. & Head of Computer Science & Engineering Department*

Gyan Ganga Institute of Technology & Sciences, Jabalpur, (M.P.), India

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. To improve efficiency better approach is used called Map Reduce for Parallelization Genetic Algorithm (MRPGA)[3][4] by using the features of Hadoop over cluster. An analysis of proposed Algorithm CBPGA to evaluate performance gains with respect to the current algorithm MapRedue Word Count [14] and GA [5]. Our proposal aim is to evaluate both the time 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.

Keywords: Big Data, word count, Hadoop, Map Reduce, cluster , Parallel Genetic Algorithm.

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.

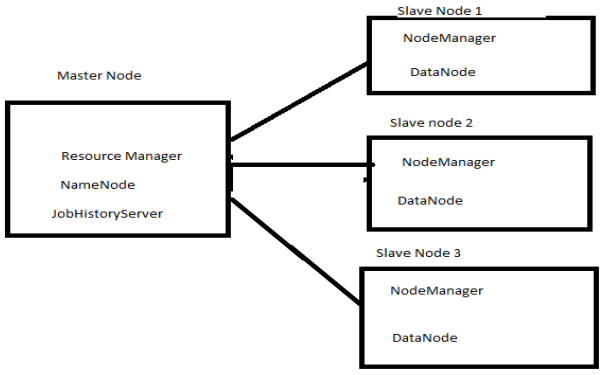


Figure 1 Data Cluster on Hadoop

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.

I. SYSTEM MODEL

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.

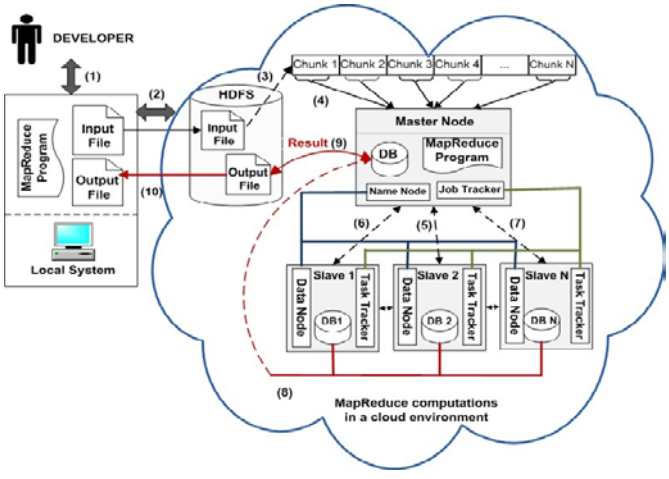


Fig 2.1:OVERVIEW ARCHITECTURE OF CBPGA

II. PREVIOUS WORK

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.

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.

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.

III. PROPOSED METHODOLOGY

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
- InputFormat and OutputFormat: 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.

IV. SIMULATION/EXPERIMENTAL RESULTS

1. Software and Hardware Requirement

The minimum requirements are as follows:

Hardware Requirement:

- 4GB RAM
- 20 GB Share of the hard disk
- Intel Core i3 2100 CPU
- 10/100/1000 Ethernet LAN Connectivity

Software Requirement:

- Linux operarting system 64bit
- Hadoop
- Virtual machine

2. Implementation

Steps:

2.1 Configuring VMware Workstation7.1.4 on Santo OS

2.2 Create and Configure New Virtual Machine

2.3 Configuring Hadoop

2.4 Compiling the Program

2.5 Creating a jar

2.6. Running the Program

3. Results

Our scenario involves implementations of the PGA algorithm in different cases. We have tested the performance of the PGA implementations on the Hadoop cluster 2.7.1, Java version 1.6 with the focus on speedup and performance.

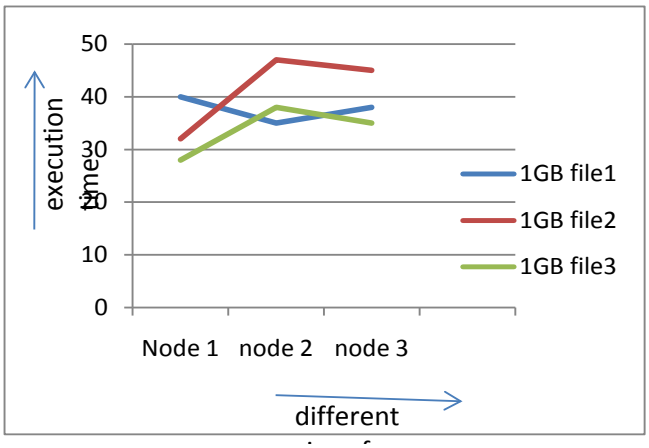
Case 1: Execution Time with Fixed Sizes of text files on different nodes

In this case, we keep input files are fixed and increase the number of nodes execute at different iteration.

Table1: Same size files on different node.

Text Size	1 Node (sec)	2 Nodes (sec)	3 Nodes (sec)
1GB	40	32	28
1GB	35	47	38
1GB	38	45	35

Graph 1: Same size files on different node.



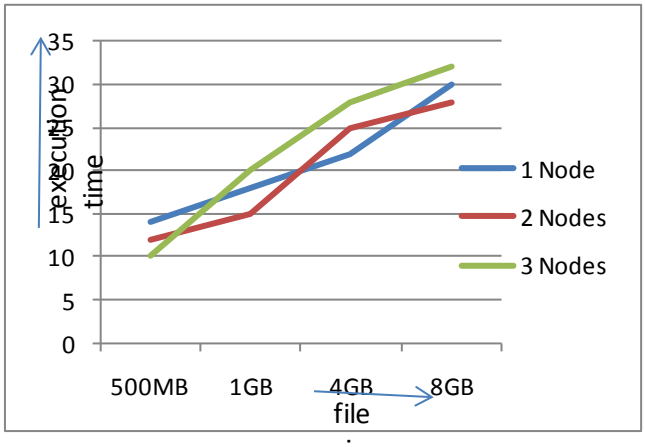
Case 2: Execution Time with different Sizes of text files on different nodes

In this case, we increase the file size and observe the performance of an algo on different nodes.

Table2: Different files size on Different Node

Text Size	1 Node (sec)	2 Nodes (sec)	3 Nodes (sec)
500MB	14	12	10
1GB	18	15	20
4GB	22	25	28
8GB	30	28	32

Graph 2: Different files size on Different Node



Hadoop implementations the performance has improved when we increased the number of nodes. Thus, adding more resources while keeping the node size fixed and also with increasing the file size decreases the execution time. However, with increasing file size and keeping number of nodes, iterations and dimensions constant gradually increases the execution time. We also observed that the performance of the first implementation is faster than the second implementation. For larger node dimensions, if we

would increase the hardware and resources, both implementations can scale well.

Compare Table

In this implementation, we take different file size as input at different size of nodes because it is compare it at above graph which show that it takes less time to complete its execution by using PGA. In addition, all the nodes of the Hadoop cluster work on entire files compute on different node independently. The node will take the complete files and calculate the execution time by using PGA.

We compare our implementation PGA with the existing work GA. Here we compare both algo for different file size on different size in nodes with in minimal execution time over a cluster.

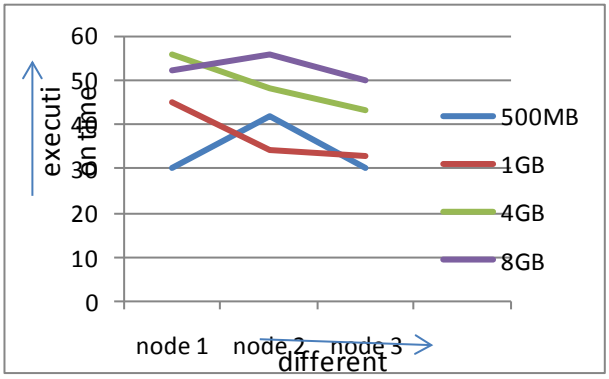
1. Execution Time with different files Sizes on different nodes in a cluster by using GA

Table 3: different size of file on different nodes by using GA

File size	Node 1 (sec)	Node 2 (sec)	Node 3 (sec)	GA(sec)
500MB	30	42	30	34
1GB	45	34	33	37.4
4GB	56	48	43	49
8GB	52	56	50	52.7

The table or graph shows that by using different file size of text file on different nodes in a cluster the time is also increases and much different means its depend on file size if we take large size file its also take more time to calculate. The average calculation of a cluster for 500MB files in 34 sec and so on...For a different node in a cluster executes a file independently.

Graph 3: different size of file on different nodes

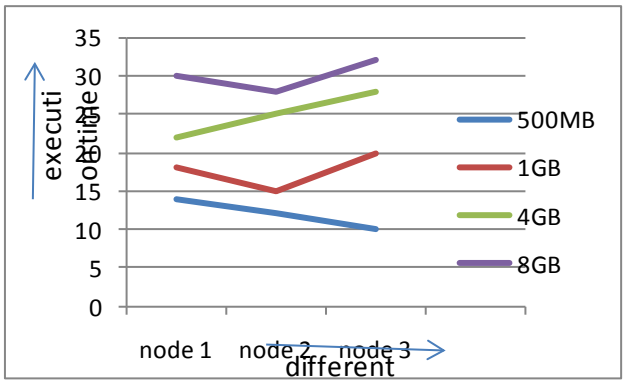


2. Execution Time with different files Sizes on different nodes in a cluster by using PGA

Table 4: different size of file on different nodes

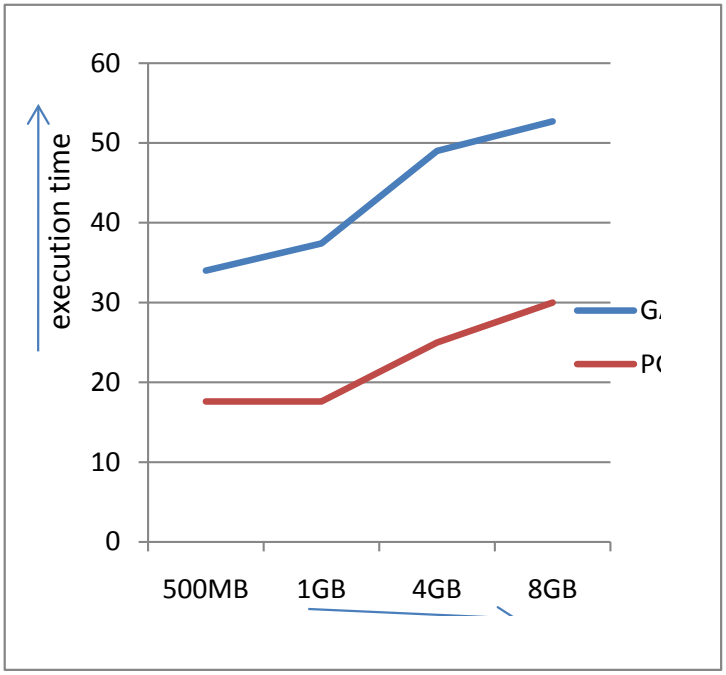
File size	Node 1(sec)	Node 2(sec)	Node 3(sec)	PGA(sec)
500MB	14	12	10	17.6
1GB	18	15	20	17.6
4GB	22	25	28	25
8GB	30	28	32	30

Graph 4: different size of file on different nodes



Here is our comparison graph between GA and PGA where the different file size is execute on different nodes over a single cluster comparison a time for both algo.

Table 4.1: Comparing Graph between GA & PGA of different size of file on different nodes

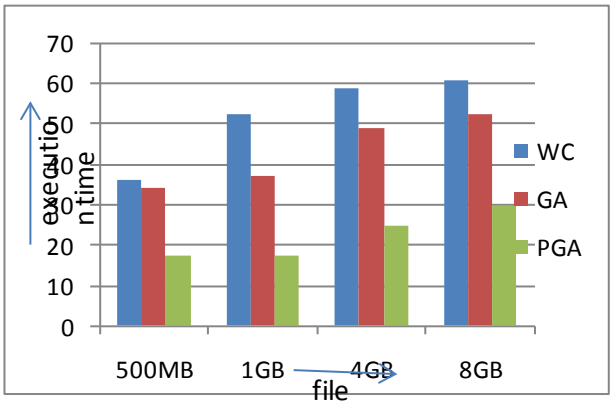


Our implementation is best because of it compares with it existing algo its execution time is less with others and is uses the features of Hadoop in a distributed and parallel manner

Table 5: Comparison on different algo

Size of File	Time Taken by using WC in sec	Time taken by using GA (approx) in sec	Time taken by using PGA in sec
500MB	36.3	34	17.6
1GB	52.4	37.4	17.6
4GB	59	49	25
8GB	61	52.7	30

. Graph 5: Comparison on different algo



V. CONCLUSION

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.

VI. FUTURE SCOPES

In Future, we Hope to Improve Upon the Accuracy and Enhance the Speed Gains. Further, semantics can be added to increase the speed of computing nodes. After this semantic similarity measures can be applied to cluster.

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.

[6]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.

[7] 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.

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

[9] 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

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

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

[12] 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.

[13] 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.

[14] MapReduce WordCount: Execution and Effects of Altering Parameters ,Dr. A.J Singh, Vibha Sarjolta Professor, Department of Computer Science, Himachal Pradesh University, Shimla, India Research Scholar, Department of Computer Science, Himachal Pradesh University Shimla, India International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 10, October 2015.