A Review on Genetic Algorithm Practice in Hadoop MapReduce

Mrs. C. Sunitha
Professor and Head
Department of BCA & MSc SS
Sri Krishna Arts and Science College, Kuniamuthur, Coimbatore

Ms. I. Jeevitha
Assistant Professor
Department of BCA & MSc SS
Sri Krishna Arts and Science College, Kuniamuthur, Coimbatore

Abstract

In recent days generating data transfer become faster than ever. Need to maintain huge datasets, systems are increasingly interconnected. The Big data is the collection of large data sets, like billions of billion data stored and accessed at one place that cannot be processed by using traditional computing techniques. The Big data at whole will survey with several tools, techniques and framework. The ubiquitous key to the big data access is Hadoop. Hadoop is a framework used for processing large amount of data in parallel. Hadoop provide the reliability in storing the data and efficient processing system. Two main gears of Hadoop are the HDFS (Hadoop Distributed File System) and Map Reducing (for processing). Hadoop cluster is a vital element, where it folds all the datasets. It is constructed by nodes i.e. server, most are slave nodes, few are master nodes which are interconnected. Map reducing is a processing model; accomplishing the task by using Map and Reduce method. Genetic algorithm (GA) is a dominant metaheuristic search technique, which is used to solve many real world applications. The GAs find the optimal solution with reasonable time, that can be executed in parallel. Though implementing GAs in Hadoop is not easy, finding the solution which could survey for the fittest is superior.

Keywords: Genetic Algorithm, Hadoop, Map Reduce, Parallel GAs

I. INTRODUCTION

Apache Hadoop is software framework that processes the Bigdata such as data in range of petabytes. The framework was developed by Doug Cutting, the creator of Apache Lucene as a part of Web Search Engine Apache Nutch [1]. Apache Hadoop, the parallel computing system solves the problem in large Datasets. Hadoop uses same machine for storage and processing as it significantly reduces the need of data transfer across the network. Hadoop has nowadays become a major force. The buzzwords these days are Bigdata and Hadoop. Hadoop system has three major components, HDFS (Hadoop Distributed File System), MapReduce and Yarn. HDFS is used to store the data across systems in the form of blocks. MapReduce is used to fetch the required data, Yarn acts as an interface between other application like Hbase, Spark, etc. and HDFS. Hadoop need a support of five strong pillars which are follows:

A. PIG:

Pig is a high level language that works on semi-structured data like log files. It uses the language called Pig Latin. Queries written in a Pig Latin is compiled to Maps and Reduces, and then executed on a Hadoop cluster. In MapReduce, it is very tough to create a joins between tables. But using Pig it is easily accomplished. It provides the best way for joining data sets, filtering, sorting and grouping data. Pig provides one major functionality, user defined functions (UDFs), and user can create his own function to process the data. MapReduce model composed of three phases, processing of input records, forming groups of related records and producing the group as outputs. The Mapper handles the first two phases, and Reducer handles the third phase. Pig Latin exposes explicit primitives that perform action from each phase. Pig works on local mode (`works on local file system) and Hadoop or MapReduce mode extracts Pig Latin into MapReduce jobs and executes them on the cluster. The figure state the process of the Pig , where the pig queries are written to fetch the data stored in HDFS, then Pig scripts are internally converted into MapReduce jobs and finally the MapReduce jobs query in turn to HDFS file system and return the result for the query.

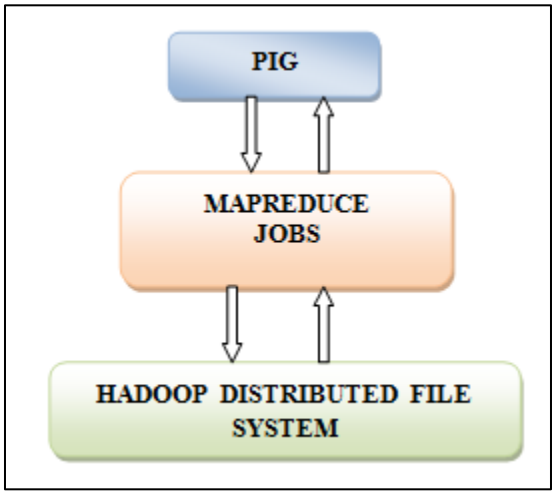

Fig. 1: PIG PROCESS

B. SQOOP:

SQOOP is a open source tool used to manage the data interaction between traditional RDBMS and Hadoop environment. Sqoop works on three processes, first request the metadata information about the table to the relational DB, second create Java Classes from the received file, and finally jar is made out of the complied files.

C. HIVE:

Hive act as data warehouse of Hadoop. It's easy to write and implement queries in Hive. Hive is not an traditional RDBMS, it can be used for batch processing. It has default metastore that contains location of tables. Hive queries are called HQL (Hive Query Language).

D. HBase:

It is a column oriented database present in the top of HDFS. It provides the comfort zone for sparse data sets. It doesn't care about the type of data provide. It follows three different modes: standalone, pseudo-distributed and full distributed.

E. ZOOKEEPER:

It provides coordination service for distributed application. It solves the deadlocks, race conditions, partial failures and networking problems in the distributed application. Zookeeper has a leader, observer and follower. Only a leader can process and commit to write operation request via follower. Voting process carry out between the followers if the leader goes down. Observer just listens to the voting results but never take part in voting process.

MapReduce is a context for processing a large Datasets using centered key value pairs. MapReduce which split the function into map and reduce. The Map function always run first, the output of the map function act as input to the reduce function. These two tasks are carried out in parallel across the clusters [11]. The MapReduce automatically handles failures, hiding the complexity of fault tolerance from the programmer. If the node crashes, MapReduce reruns the tasks on a different machine. If a node is available but is performing poorly, a term that we call a straggler, MapReduce runs a speculative copy of its task (also called a "backup task") on another machine to finish the working faster. Without this mechanism of speculative execution, a job would be a slow as the misbehaving task. Stragglers can rise for many reasons, including the faulty hardware and misconfiguration [1].

The problem with non-polynomial complexity search for an optimum solution, it involves huge resource and execution time. The best alternative to find solution for nearer—optimal is Genetic Algorithms (GAs) within a reasonable time and limited resources. Genetic Algorithms consider as a leading metaheuristic techniques is mostly used solve the real world application. Apache Hadoop is a common service to solve the problem in parallel distributed system. It is easy to adapt the Genetic Algorithm which is a parallel processing in optimal. The GAs algorithm simulates the biological process of reproduction. It begins with the initial population of individual [2]. Genetic Algorithm (GAs) is efficient search methods based on the principles of natural selection and genetics [3].

II. MAPREDUCE

Google projected the MapReduce (Dean and Ghemawat, 2008) that enable the users to develop a large scale distributed application. MapReduce is a programming model [6] and implemented for processing large data sets. Two distinct function which is similar to divide and conquer method, which is Map and Reduce. Map function handle the parallelization, Reduce function collects and merge the results. A large number of data analytical jobs can be expressed as a set of MapReduce jobs. MapReduce jobs are automatically parallelized and executed on a cluster of commodity machines. A MapReduce job consists of two phases, map phase and reduce phase. The map phase processes input key/value pairs, evaluate and produce new list. The reduce phase produce final key/value pair per group according to the intermediate key value. The MapReduce perform the following transformation to their input [1]:

$$map\ (k1,v1) \rightarrow list(k2,v2)$$

$$reduce\ (k2,list(v2)) \rightarrow list(k3,v3)$$

The map invocation are scattered across multiple machines by automatically partitioning the input data into a set of M splits. The Reduce works is based on the result of Map invocation and with intermediate key value it produce the result [9].

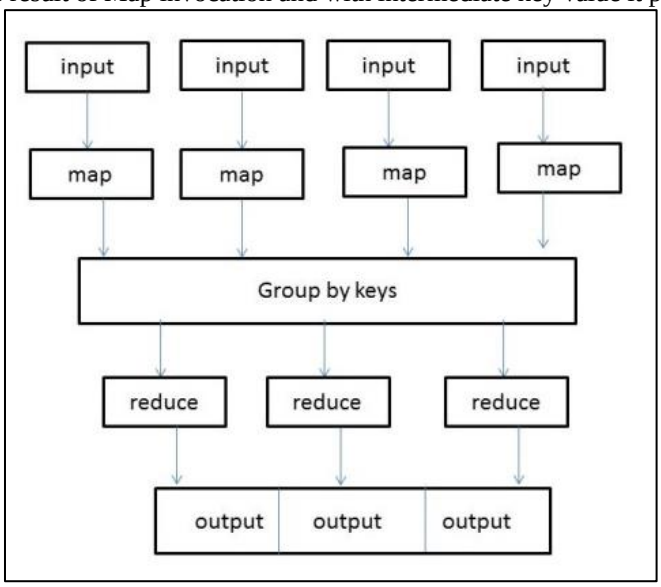

Fig. 2: MapReduce Data flow Overview

A. HADOOP MAP REDUCE:

Hadoop MapReduce is a framework for processing large data sets in parallel across Hadoop cluster. It uses two steps Map and Reduce process [10]. An initiated job has a map and reduces phase. The map phase counts the words in each document, then reduce phase aggregates the per document data into word counts spanning the entire collection. The reduce phase use the results from map tasks as input to a set of parallel reduce tasks and it consolidates the data into final result. The Map-Reduce uses the key value pairs as input and produce a set of output key value pairs. The key value pairs in the input set (KV1), can be mapped on the cluster and produce the (KV2) as output. The reduce phase finally performs the set of operation and make new output set (KV3).

The output of Map function contains multiple key values. Multiple keys values act as an input for the Reduce function. The mapper extracts the support call identifier (pass to reducer as a key) and the support call description (pass to the reducer as the value). Each map tasks receives a subset of the initial centroids and is responsible for assigning each input data point, to the nearest centroid (cluster). Every time the mapper generates a key / value pair, where the key is the cluster identifier and the value corresponds to the coordinates of the point [12]. The algorithm uses a combiner to reduce the amount of the data to be transferred from the mapper to the reducer. The Hadoop system follows different task before approaching the map/reduce function.

1) Job Client:

A job client prepares a job for execution after submitting the map/reduce job. The job client validates the job configuration, and generates input split. Then job client submit the validated job to the Job Tracker.

2) Job Tracker:

The Job Tracker is responsible for scheduling jobs, and dividing the job for map/reduce tasks. It creates map task for each split and assign each map task to the Task Tracker.

3) Task Tracker:
Task Tracker manages the tasks of one worker node and report status to the Job Tracker. The tasks are spawned by the task tracker and run the jobs map or reduce function.
4) Map Task:
Depending on the key value the data will be mapped by the Map task. The map task notifies the completion of the Task Tracker. Fetch input data locally based on the keys values and makes it available for the Reduce Task.
5) Reduce Task:
It aggregates the result from the Map Task and creates a final result set which is smaller than the input set. The Reduce task can begin as soon as the completion of the Map Task. It is not necessary that all the Map Task should complete before any reduce task can begin.

III. GENETIC ALGORITHM

Genetic algorithm is adaptive heuristics search algorithm based on the evolutionary ideas of natural selection and genetics. It also mimics the process of natural selection. This heuristic sometimes identified as metaheuristic, is used to generate the useful solutions to optimization and research problems [13]. In order to implement feature subset selection method is Hadoop platform using genetic algorithm. GAs begins with a set of k randomly generated states called "population". Each state is called "individual" and is represented as a string over a finite alphabet. This string is called "chromosome" and each symbol "gene". Genetic algorithm are the way of solving problem by mirroring processes nature uses ie Selection , Crossover , Mutation and Accepting to develop solution to the problem. GAs uses the random search technique to solve the optimization problems. Every iteration of the algorithm generates a new population and consists of the following steps: [4]
1) Selection:
Each state is evaluated by the fitness function (b). Each value influences the random choice among the successors for the next step. Once chosen the k successors are grouped into couples (c);
2) Crossover:
The algorithm chooses for each individual a division point, which is called crossover point. At this point the sexual reproduction (d) in which two children are created begins: the first takes the first part of the first parent and the second of the second parent; the other takes the second part of the first parent and first part of the second parent, the logic is used to identify the individual in the set.

The crossover parents form a new offspring, or if no crossover is performed offspring copy the parents. To make individuals meet their constraint, newly generated individuals need to be modified. Using the crossover probability Pc, empty pool set Pl for each individual a population is generate a real number $q \in [0,1]$, For each individuals in the set generate a new individual thus the crossover operation accomplished.
3) Mutation:
When the offspring's are generated, each gene is subjected to a random mutation with a small independent probability (e). It selects the individual from the offspring of crossover according to the mutation probability Pm.
Genetic algorithm is good at taking large, potentially huge search spaces and navigating them, looking for optimal combination of things, the solution one might not find anywhere. Genetic algorithm begins with the set of solution called population, solution from one population are taken to form a new population where new population is better than exist one. Solution is selected according to the fitness from the existing, is repeated until the some condition is satisfied [7].

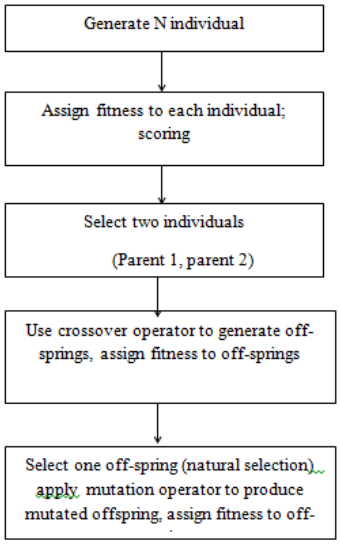

Fig. 3: Genetic Algorithm Flow

A. A Parallel Genetic Algorithm Based On Hadoop Map Reduce:

In the following section we first give some background reporting on the strategies proposed in the works to parallelize genetics algorithm and recollecting the main aspects of MapReduce and Hadoop MapReduce. Several GAs parallelization techniques exists based on the grain of parallelization to complete. Basically three types of parallelization techniques exists,

1) Fitness evaluation level (also consider as global parallelization model).
2) Population level (also coarse grained parallelization or island model).
3) Individual level (fine-grained parallelization or grid model) [4].

In global parallelization model, master node manages the population and distributes the individual among the salve nodes; compute only the fitness value of the individuals [14]. The model is followed to MapReduce by assigning some mappers the job of evaluating the accuracy for those individuals that are executed in parallel. An initial population drawn randomly and combination of random attribute is taken for the fitness calculation.

The individual with best accuracy are allotted for the selection, and then selection process choose the individuals that doing as parents during crossover. This is achieved by tournament selection process where numerous tournaments between small numbers of individuals are selected at random. Crossover is the procedure of obtaining higher than one parent solution and making child from them. A single crossover point is accomplished; data beyond the point will be swapped between the two parent solutions and children will be the solution.

Mutation defines the probability to mutate in which subclass exchange the attributes with each other [5]. In population level model, the population is subdivided in several subpopulation of relatively large size which are located in several nodes (or islands). Thus genetic algorithm is executed on each subpopulation and such subpopulation exchange information by allowing some individuals to migrate from one node to another.

Finally grid model, each node is placed on the grid and all the GA operation performed in parallel, simultaneously evaluate the fitness and applying the operation to the small neighboring. The fine-grained model is the extension of the coarse grained model with slight modification. Each PGAs involves the crossover and mutation for calculating the fitness level and there for producing the best accuracy.

IV. COMPARATIVE STUDY OF GENETIC ALGORITHM AND PARALLEL GENETIC ALGORITHM

The genetic algorithm is applied for solving many optimization problems. Genetic algorithms are based on perfect population model. Parallel genetic algorithms are an extension of genetic algorithms that are designed to work for parallel computers. Parallel Genetic algorithms simulate the evolution finite population more realistically [9]. Genetic algorithm is a group of computational models based on natural selection principles. This algorithm transforms the problem in a particular domain into a model using chromosome – data structure and finds the solution using selection and mutation operator. The parallel genetic algorithm tends to increase the processing speed, capacity to process large amount of data in polynomial times [8]. The genetic algorithm acts a traditional approach for solving the problem, using the GAs itself we can trace out the difficulties in MapReduce and decrease the time which is used to produce the result. Genetic algorithms have the ability to avoid being trapped in local optimal solution using traditional methods, which search from a single point. It uses probabilistic selection rule and uses a fitness score that is obtained from objective function. The GA is used to evolve the optimal solution based on the techniques of selection, mutation, crossover and inheritance in polynomial time.

V. CONCLUSION

In this paper we review the Genetic algorithm and 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. The execution time may vary depend up on the data sets but 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.

REFERENCE

[1] Improving Job Scheduling in Hadoop MapReduce Himangi G. Patel, Richard Sonaliya Computer Engineering, Silver Oak College of Engineering and Technology, Ahmedabad, Gujarat, India. June 2015 | IJIRT | Volume 2 Issue 1 | ISSN: 2349-6002.
[2] Hadoop: The Definitive Guide, T. White, Third. O'Reiily, 2012.
[3] Scaling Simple and Compact Genetic Algorithms using MapReduce Abhishek Verma, Xavier Llor_a, David E. Goldberg, Roy H. Campbell IlliGAL Report No. 2009001 October, 2009.
[4] A Framework For Genetic Algorithm Based On Hadoop , arXiv :1312.0086v2[cs.NE] 15 dec 2013, Filomena Ferrucci, M-Tahar Kechadi, Pasquale Salsa, Federica Sarro
[5] A New Approach For Feature Subset Selection Based On Hadoop ,Ramya P V1, Shashikala B2 June 2015 | IJIRT | Volume 2 Issue 1 | ISSN: 2349-6002
[6] "MapReduce: simplified data processing on large clusters." Dean, Jeffrey, and Sanjay Ghemawat. Communications of the ACM 51, no. 1 (2008): 107-113.
[7] Parallel genetic algorithms , population genetics and combinatorial optimization H Mühlenbein - Parallelism, Learning, Evolution, 1991 - Springer

[8] Parallelization Of Genetic Algorithm Using Hadoop Ms. Rahate Kanchan Sharadchandra. Student M.E.(Computer Science & Engineering) Walchand Institute of Technology, Solapur y (IJERT) Vol. 1 Issue 9, November- 2012 ISSN: 2278-0181
[9] MapReduce: simplified data processing on large clusters," in Proceedings of the 6th Symposium on Operating System Design and Implementation, 2004, pp.137-150J. Dean, S. Ghemawat.
[10] Apache Hadoop Map Reduce, http://hadoop.apache.org/mapreduce/
[11] Big Data Clustering Using Genetic Algorithm On Hadoop Mapreduce Nivranshu Hans, Sana Mahajan, SN Omkar International Journal Of Scientific & Technology Research Volume 4, Issue 04, April 2015 ISSN 2277-8616
[12] http://techalpine.com/what-are-the-hadoop-mapreduce-concepts/
[13] "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley, 1989. D. E. Goldberg,
[14] "Parallel Genetic Algorithms: Theory and Applications," Frontiers in Artificial Intellingence, 1993, vol.14 J. Stender,