Parallelization Of Genetic Algorithm Using Hadoop

Ms. Rahate Kanchan Sharadchandra.
Student
M.E.(Computer Science & Engineering)
Walchand Institute of Technology, Solapur

Prof. L.M.R.J. Lobo
Professor & Head,
Department Information Technology,
Walchand Institute of Technology, Solapur

Abstract

Cloud computing changes the way we think about technology. According to the National Institute of Standards and Technology (NIST) cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources e.g. networks, servers, storage, applications, and service be rapidly provisioned and released with minimal management effort or service provider interaction[1]. Our basic project idea is to parallelize Genetic Algorithm (GA) so that they can use cloud computing framework. We consider GA's which are bound to be parallel. Here we will use Hadoop technologies like MapReduce framework, Hive, H-base, HDFS (Hadoop Distributed File System), Pig, Chukwa, Avro, ZooKeeper etc.
Hadoop often serves as a sink for many sources of data because Hadoop allows you to store data cost effectively and process that data in arbitrary ways at a later time. Hadoop doesn't maintain indexes or relationships; you don't need to decide how you want to analyze your data in advance.
By implementing this project we are able to process large amount of data. Our project helps to increase the processing speed and capability to process huge amount of data in polynomial times.

1. Introduction

GAs tends to find good and novel solutions to hard problems in a reasonable amount of time. GAs are effective at solving NP-Hard (non-deterministic polynomial-time hard) problems. An example of an NP-hard problem is the Travelling Salesman Problem [2]. GAs is effective at solving NP-Hard problems because they need only traverse a small percentage of the problem space to get a "good" solution. GA's are naturally parallel. In this project we are try to parallelize GA to improve the processing speed. We are going to use Hadoop MapReduce framework approach. The motivation for implementing GAs in Hadoop is that MapReduce fares well in terms of scalability, fault tolerance, and ease-of-use.

Hadoop is an open-source implementation of MapReduce. MapReduce allow computation over terabytes of data to become routine. Hadoop is written in Java. In this project we are trying to improve the processing speed as well as process huge data.

The rest of the document is organized as follows. Section 2 provides a background of the related work fields covering a brief introduction about each. Section 3 describes the Literature Review. Section 4 discusses the methodology and lastly conclusion.

2. Background

Before delving into the implementation details of how to integrate MapReduce and genetic algorithms, discussion of the component technologies is in order, starting with cloud computing.

The following subsections include a brief overview of various topics.

2.1 Cloud Computing

Cloud computing is a latest new computing paradigm where applications, data and IT services are provided over the Internet. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. It define in three models Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) [3]. Cloud computing also

describes applications that are extended to be accessible through the Internet. These cloud applications use large data center and powerful servers that host Web applications and Web services. Anyone with a suitable Internet connection and a standard browser can access a cloud application. Figure 1. Shows overall idea about cloud computing.

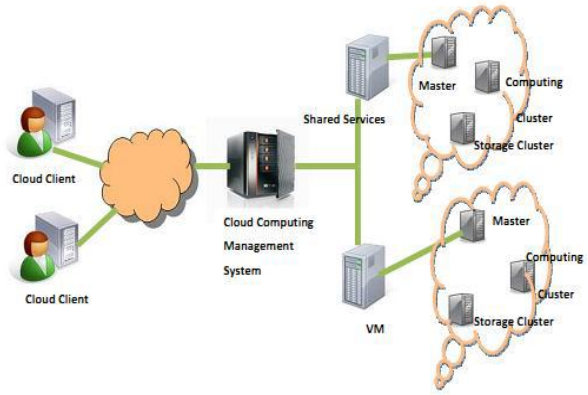

Figure 1. Cloud Computing
Source : Tasks Scheduling optimization for the Cloud Computing Systems [11]

2.2 Genetic Algorithm

Genetic algorithm is a family of computational models based on principles of evolution and natural selection .These algorithms convert the problem in a specific domain into a model by using a chromosome-like data structure and evolve the chromosomes using selection, recombination, and mutation operators. The range of the applications that can make use of genetic algorithm is quite broad. There are two types of GS's named as Sequential GA and Parallel GA. Much of the research in genetic algorithms has been in sequential GAs. Genetic algorithms encode candidate solutions as data structures called chromosomes, or individuals. Each individual in a population is assigned a numeric fitness value by the fitness function. Once all the fitnesses are assigned, selection begins. A popular selection algorithm is roulette wheel (RW) selection. In RW selection, first the total fitness is calculated by summing the fitnesses of every individual in the population. Then each individual's fitness is divided by the total fitness to find its relative fitness. This relative fitness corresponds to the probability this individual will be selected and allowed into the mating pool. There are several classes of parallel genetic algorithms (PGAs) these are global, coarse-grained, fine-grained, and hybrid. Global PGAs use a single population and simply parallelize the evaluation of the fitness, and then sequentially produce the next generation. Coarse-grained parallelization involves evolving many separate subpopulations, called demes, in parallel. Coarse-grained PGAs often also implement migration, allowing an individual to move from one deme to another. Migration depends on the topology of the demes, how they are connected, and often a probability of migration use to be specified. These smaller subpopulations exhibit less selective pressure than do Global PGAs. Fine-grained parallelization involves assigning one individual per processor core. This is usually undertaken with special hardware. Hybrid parallelization is some combination of the above techniques.

2.3 Hadoop

Hadoop is an Apache project; all components are available via the Apache open source license. Hadoop is written in Java. It doesn't maintain indexes or relationships; you don't need to decide how you want to analyze your data in advance. It breaks data into manageable chunks, replicates them, and distributes multiple copies across all the nodes in a cluster so you can process your data quickly and reliably later. Hadoop is also use to conduct analysis of data. Hadoop introduce many components like MapReduce, Hive, H-base, HDFS (Hadoop Distributed File System), Pig, Chukwa, Avro, ZooKeeper etc. information is given in Table 1.

Table 1. Hadoop Components [9]

Component	Developer	Description
MapReduce	Yahoo !	Distributed computation framework
HDFS	Yahoo !	Distributed file system
H-Base	Powerset (Microsoft)	Column-oriented table service
Pig	Yahoo !	Dataflow language and parallel execution framework
Hive	Facebook	Data warehouse infrastructure
ZooKeeper	Yahoo !	Distributed coordination service
Chukwa	Yahoo !	System for collecting management data
Avro	Yahoo ! & Cloudera	Data serialization system

Yahoo! has developed and contributed to 80% of the core of Hadoop (HDFS and MapReduce). H-

Base was originally developed at Powerset, now a department at Microsoft. Hive was originated and developed at Facebook. Pig , ZooKeeper and Chukwa were originated and developed at Yahoo!. Avro was originated at Yahoo! and is being co-developed with Cloudera.

2.4 MapReduce Framework

Google took the automatic parallelization from functional programming map and reduce, to develop a framework called MapReduce. This framework enables highly fault-tolerant massively scalable computation to occur on a network of commodity hardware. MapReduce has proven successful at allowing computation over terabytes of data to become routine. Hadoop, an open-source implementation of MapReduce, has been successfully used by Facebook, Yahoo, Netflix, and other players in the cloud-computing space.

MapReduce programs break problems into Map and Reduce phases. The Map Phase handles all of the parallel computation, and the Reduce phase handles all of the sequential computation. The programming model of MapReduce takes a set of input key/value pairs, and produces a set of output key/value pairs. The userspecified map and reduce functions are of the following type:

map(k1, v1) → list(k2, v2)

reduce(k2, list(v2)) → list(v2)

The output key/value pairs of the map phase are sorted by their key and each reducer gets a particular key and a list of all the values belonging to that key. Hadoop automatically does this sorting and routing of information between the many machines in the cloud Infrastructure, often using a distributed file system such as HDFS to perform the communication and synchronization. The input keys are usually extracted from massive data files and are split among many mapper nodes. If any one mapper node fails, a "master node" will automatically distribute the map task to a different mapper node. This, along with redundant storage, provides the fault-tolerance that is vital to any successful cloud-computing framework.

Hadoop is written in Java, Hadoop provides two options. The Hadoop Streaming interface allows any STDIN/STDOUT executable program to be used as a mapper or reducer, effectively supporting all programming languages with one stroke. Hadoop provides the Distributed Cache as a simple way of accomplishing this. The Distributed Cache will copy a file to every node in the job, which can then be read by the configure method of a mapper or reducer [4].

3. Literature Review

Being a recent technology people are opting to select the fields of Cloud Computing as their related area for research work. Some related work indicated below has motivated us to our proposed work. A strong relevance is seen to be established at national and international level. However applying a combination of Hadoop and Genetic Algorithms would result into a definatly more optimized approach.

Abhishek Verma [5] presented step-by-step transformations for three illustrative cases selecto-recombinative genetic algorithms and estimation of distribution algorithms and reviewed some best practices during the process in his thesis. Transformations have shown that Hadoop's MapReduce model can help scale easily and transparently evolutionary computation algorithms. Results have shown that Hadoop is an excellent choice when we have to deal with large problems, as long as resources are available, being able to maintain iteration times relatively constant despite the problem size.

Mocanu, Andreica, Tapus, N. [6] where focused on minimizing the global execution time of processor, for that purpose they use Hadoop Map-Reduce functionality by implementing a scheduler based on a genetic algorithm .

MRPGA: An Extension of MapReduce for Parallelizing Genetic Algorithms [7]. This paper appears in eScience, 2008. eScience '08. IEEE Fourth International Conference 2008. Authors Chao Jin presents an extension to the MapReduce model featuring a hierarchical reduction phase. This model is called MRPGA (MapReduce for parallel GAs), which can automatically parallelize GAs. They describe the design and implementation of the extended MapReduce model on a .NET-based enterprise grid system in detail.

In October of 2009, the first paper integrating GAs and Hadoop was published by the Illinois Genetic Algorithms Laboratory (IlliGAL) [8]. This new development permits a comparison of the approaches taken at IlliGAL. The IlliGAL work did not create a custom Chromosome type, instead relying on an array of long long ints as a chromosome. The inability to represent the pair of the DNA (ByteArray) and the fitness as a single value means that the IlliGAL team had to use the key as the chromosome and the value as the fitness. This does not make use of the Hadoop data model very well as the keys determine the partitioning of the output to reducers.

In this paper they described architecture of Hadoop Distributed File System **(**HDFS) and report on experience using HDFS to manage 25 petabytes of enterprise data at Yahoo [9].

Mr. Chidambaran Kollengode [12] Director Cloud Computing, Yahoo! India R&D, talks about the Cloud computing in Yahoo with emphasis on Hadoop Grid. The Grid Computing group at Yahoo! Bangalore focuses on Grid frameworks that

scale to thousands of machines and handle peta-bytes of data. The group is especially involved in the development of the Open Source Hadoop platform and its deployment within Yahoo!

Raghava, N.S.[13] expresses that Cloud computing is one of the highly researched areas today, with an objective of taking advantage of various computational resources. In his paper he has used cloud computing environment with the aim to speed up the matching process of biometric traits. They have used iris recognition, a biometric technique, as it is one of the strongest methods of authentication. Also Iris recognition is stable over time. They have used Hadoop, an open source cloud computing environment, to develop this model. Hadoop implements Map/Reduce framework in Java. Map/Reduce makes easy to process large amount of data on cloud. The results show that there is an effective speedup and efficiency gain of Iris template matching on Hadoop process over sequential process.

India's No.1 Hadoop / Cloud Computing Training Big data incorporate uses cloud computing for training using online facilities for cloud computing and Hadoop.

4. Methodology

This proposal combines a hybrid parallel genetic algorithm with a single MapReduce cycle per generation. This hybrid parallel genetic algorithm exploits global parallelization in the map phase and coarse-grained parallelization in the reduce phase. Each mapper evaluates the fitness of chromosomes in parallel as per the global PGA approach. The mappers emit an output key/value pair. The individuals are then separated into subpopulations. Reducer in Hadoop processes the values of a unique key. Additionally, migration of individuals between demes is as elegant as emitting a different deme_id in the map phase. The reducers are responsible for producing the next generation of a deme via selection, crossover, and mutation. For each successive generation, the output of the reducers is fed back as input to the mappers until stopping conditions are met.

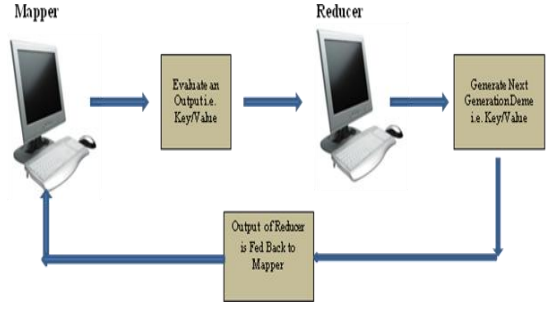

Figure 2. Mapper and Reducer

This implementation allows an enormous level of control over the parameters of the GA. Using an XML file placed into the Distributed Cache, a number of demes are specified with a unique deme_id. Each deme has a textual label, population size, chromosome length, crossover probability, mutation probability, and a list of neighbor demes with a migration probability for each. Migration can occur over an arbitrary topology of demes. In addition, the deme_id is passed to the fitness function along with every individual chromosome, so a fitness function could evaluate demes differently. The deme_ids population sizes and chromosome lengths from the XML file are also used to seed the initial generation with random individuals. It is believed that the flexibility and generality of this approach will encourage use of the reference implementation as a starting point for future GA work in Hadoop.

5. Conclusion

In this model we use parallelize Genetic Algorithm (GA) so that they can use cloud computing framework. With the help of Genetic Algorithm, we try to process large amount of data. This model helps to increase the processing speed and capability to process huge amount of data in polynomial times. Application of this model is data driven. Here we use Apache Hadoop which helps to develops open-source software for reliable, scalable, distributed computing [10].

6. References

[1] Kanchan A. Khedikar and Prof. Mrs. S. S. Apte, "Latest Technology In Networking: Cloud Architecture", in *International Conference* ICETT 2010.

[2] P.Larrañaga, C.M.H.Kuijpers, R.H.Murga, I.Inza, andS.Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators", in *The Artificial Intelligence Review*, vol. 13, no. 2, p. 129, Apr 1999.

[3]http://searchcloudcomputing.techtarget.com/definition /cloud-computin

[4] "Genetic Algorithms in the Cloud" from *MENTION*.

[5] Abhishek Varma, "Scaling Simple, Compact And Extended Compact Genetic Algorithms Using MapReduce", A thesis submitted to University of Illinois at Urbana-Champaign, in 2010, Urbana, Illinois.

[6] Mocanu, Andreica, Tapus, N., "Cloud Computing—Task scheduling based on genetic algorithms", *Systems Conference (SysCon)*, 2012 IEEE International on 19-22 March 2012.

[7] Chao Jin, "MRPGA: An Extension of MapReduce for Parallelizing Genetic Algorithms", eScience, 2008. eScience '08. *IEEE Fourth International Conference* on date 7-12 Dec. 2008.

[8] A. Verma, X. Llorà, D. E. Goldberg, and R. H. Campbell, "Scaling genetic algorithms using MapReduce," in 2009 Ninth *International Conference* on Intelligent Systems Design and Applications, Pisa, Italy, 2009, pp. 13–18.
[9] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler, "The Hadoop Distributed File System", ©2010 IEEE.
[10] Apache Hadoop. http://hadoop.apache.org/
[11] Sandeep Tayal, "Tasks Scheduling optimization for the Cloud Computing Systems" in (IJAEST) *International Journal* Of Advanced Engineering Sciences And Technologies,Vol No. 5, Issue No. 2, 111 – 115.
[12] Chidambaran Kollengode, "Cloud Computing and Hadoop" 2010
[13] Raghava, N.S. "Iris recognition on Hadoop: A biometrics system implementation on cloud computing" , Cloud Computing and Intelligence Systems (CCIS), 2011 *IEEE International Conference* on 15-17 Sept. 2011, Page**(s):** 482 - 485