A CLOUD COMPUTING MODEL BASED ON HADOOP WITH AN OPTIMIZATION OF ITS TASK SCHEDULING ALGORITHMS

Yulu Hao, Meina Song, Jing Han and Junde Song
School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China

Keywords: Cloud computing, The computing model, Hadoop, Task scheduling algorithms.

Abstract: Cloud Computing is a great revolution in the IT field, which means that, computing can be used as a commodity, like coal, gas or electricity. As an efficient computing model of Cloud Computing, within HDFS and the Map/Reduce system, Hadoop is an-open source code architecture for very large data process application. In this paper, we will do specific research on the Cloud Computing based on Hadoop and through the implement of it, optimize the task scheduling algorithms of this architecture. Through this paper, a better algorithm and more practical ways based on Hadoop to build up a Cloud Computing model are achieved.

1 INTRODUCTION

As a new method of sharing infrastructures, Cloud Computing is not only the development and extension of Distributed Computing, Parallel Computing and Grid computing, but also the commercial implementation of these concepts in computer science. The basic principle is to make users access to computers and the storage system according to their needs by switching resources to required applications, using a non-local or remote server clusters to provide services for them (including computing, storage, software and hardware and other services). Truly on-demand computing, Cloud Computing is significantly improving the efficiency in the use of hardware and software resources.

Among the key technologies of Cloud Computing, the computing model is to examine how to make the programming of certain types of applications more efficient. Nowadays, there are lots of such models and Hadoop is a good one of them. It is open source distributed system architecture, increasingly becoming a very practical and efficient platform for programming and developing.

In spite of being convenient and efficient, Hadoop is a relatively young project, which can be and should be improved. We will have a detailed study and research of it and try to find a better algorithm for its job scheduling after an implementation of Cloud Computing system based on Hadoop personally.

2 THE HADOOP ARCHITECTURE FOR CLOUD COMPUTING

Hadoop is an open source project of the Apache Software Foundation and it provides many kinds of reliable, scalable software in the distributed computing environment, with the ability to help us easily set up our own large-scale cluster systems on general hardware. In spite of within many subprojects, Hadoop mainly consists of the HDFS (Hadoop Distributed File System) and Map/Reduce Engine, which are the two most important subprojects of the Hadoop architecture. (Hadoop)

2.1 HDFS

Designed for general hardware, HDFS is a distributed file system which has a lot in common with existing common distributed file systems. Consists of a primary node called Name Node and a child node called Data Node, HDFS is a typical Master/Slave architecture, which can shield the complex underlying structure through the primary node and provide convenient file directory mapping

Hao Y., Song M., Han J. and Song J..
A CLOUD COMPUTING MODEL BASED ON HADOOP WITH AN OPTIMIZATION OF ITS TASK SCHEDULING ALGORITHMS.
DOI: 10.5220/0003586805240528
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (SSE-2011), pages 524-528
ISBN: 978-989-8425-53-9

to customers. The HDFS architecture is shown in Figure 1.

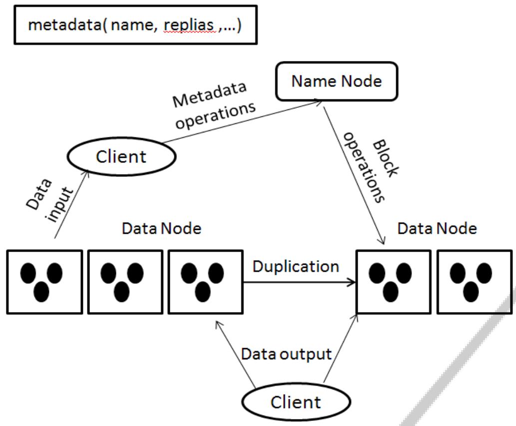

Figure 1: The HDFS architecture.

Figure 1 clearly shows us the interaction between the Name Node, the Data Node and the client. For example, while writing a file, at the very beginning, the client sends the request of writing a file to the Name Node, and then according to the size of the file and the configuration of the file blocks, the Name Node returns the information of the Data Nodes it controls to the client, at last, using the information of address in the Data Nodes it got just now, the client depart the file into blocks and put them into each Data Node one by one in order.

2.2 The Computing System Map/Reduce

The Hadoop architecture uses the Cloud Computing model of Map/Reduce proposed by Google, which is not only a technology of distributed computing, but also a simplified distributed programming model (Dean and Ghemawat, 2008). A typical process of Hadoop Map/Reduce generally includes the following steps.

2.2.1 Input

User applications often need to provide functions of Map and Reduce, and specify the input and output of the location (path), and other operating parameters.

2.2.2 Map

Regarding input operations of the client as key/value pairs (<key, value>), Map/Reduce is to call the user-defined Map function to deal with each <key, value>, then, a number of new intermediate <key, value>'s will be generated, which may have a different type from the previous ones. The main code of this process is as figure 2.

```
Public void map(int nMaps, int val)
{
    Int nMaps = key.get();
    Int m = val.get();
    Float ran[] = new float[m];
    For(int j = 0; j<MyMapNo; j++){
        //calculating each Map's
        //first value in <key, value>,
        //and put them into ix, iy, iz
    }
    For(int i = 1; i<nMaps; i++){
        //calculating each Map's
        //incremental of each two
        // adjacent values
    }
    k = (ix + iy + iz)%M4;
    ran[0] = (float)((k + 1.0))/M1);
    emit(0, ran[0]);
    for(int i = 1; i<m; i++){
        //calculating all
        //the values of the Map's
        //put them into ix, iy, iz
        k = (ix + iy + iz - 3)%M4;
        ran[i] =(float)((k + 1.0))/M1);
        emit(0, ran[i]);
    }
}
```

Figure 2: The main code of Map.

2.2.3 Shuffle & Sort

This stage is to ensure that the input of Reduce is just the sorted output of Map.

2.2.4 Reduce

The architecture is to traverse the intermediate data and for each unique key value, user-defined Reduce function will be operated and a new <key, value> is to be output. The main code of this process is as figure 3.

```
Public void reduce(int key, Iterator values)
{
    ......
    SequencesFile.Writer writer
    = SequenceFile.createWriter(fileSys,
        conf, outFile, IntWritable.class,
        FloatWritable.class,
        CompressionType.NONE);
    while(values.hasNext()){
        float result = values.next().get();
        writer.append(new IntWritable(0),
        new FloatWritable (result));
    }
    Writer.close();
}
```

Figure 3: The main code of Reduce.

2.2.5 Output

Now the result of the Reduce output is written to the output directory of the file.

Since Map/Reduce and HDFS are ordinarily running on the same set of nodes, the bandwidth of the whole cluster network can be used efficiently.

2.3 Summery and the Advantages of Hadoop

As the research of Hadoop shows above, we can see clearly the advantages of it.

Scalable: Not only the scalability of storage but also that of computing is one of the fundamental designs. As a result, the scalability of Hadoop is so simple that any of modification of the existing architecture is not needed.

Economic: it can run on any ordinary PC, with no special hardware requirements.

Reliable: The reliability of distributed processing is ensured by the backup and recovery mechanism of distributed file system and the task monitoring of Map/Reduce.

Efficient: That the implementation of efficient data exchange of distributed file system and the processing model of Map/Reduce within Local Data, are the basis of preparation for the efficient processing of vast amounts of information.

3 THE STEPS TO IMPLEMENT CLOUD COMPUTING BASED ON HADOOP

Just for outside customers, Private clouds only provide an interface to be called. Internal implementation relies on dynamic start and distribution of computer nodes. Designed to be scalable to the size of cloud dynamic clusters, can be effectively utilized computing resources and the advantages of virtual machine technology. In the private cloud, for each node requires a sound management control mechanisms, in particular cloud stretching and node startup, shutdown, data dump and resource allocation.

3.1 Hardware Configuration

According to the research of Hadoop above, the cluster of Hadoop has two most significant parts, the Master and the Slave. The main job of Master is configuring Name Node and Job Tracker, and that of Slave is configuring Data Node and Task Tracker. Now, take my experiment for example. There are three personal computers to build up the cluster, among which, one is deployed with Name Node and Job Tracker, and the other two are deployed with Data Node and Task Tracker. The operating system is Red Hat Linux with jdk-1.6.0-14 and Hadoop 0.20, and IP's with their hosts' names are added to the file of the path '/etc/hosts' of each node.

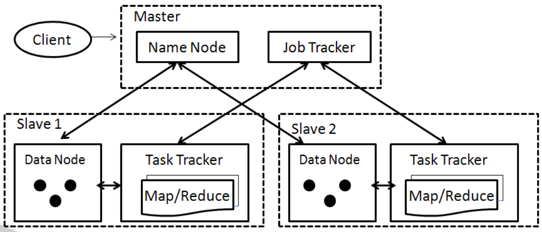

Figure 4: Hadoop-based Cloud Computing model.

If the computer is used as a Name Node, all IP's and their hosts' names in the whole cluster should be added into the file 'hosts'. Now in my experiment:

192．168．1．103　　lin-1
192．168．1．104　　lin-2
192．168．1．105　　lin-3

Otherwise, the computer is used as a Data Node. Then only needed are its own IP and the IP of the Name Node.

192．168．1．103　　lin-1
192．168．1．104　　lin-2

3.2 The Establishment of SSH Trusted Certificate

Since the master starts up Hadoop of all Slaves through SSH, we need to establish a one-way or bidirectional certificate to ensure that the password is unnecessary when instructions are executed. A pair of keys of Name Node is generated by using ssh-keygen and then its public key are copied to the file 'authorized' located at 'home/ssh' of each computer, as the result of which, the access between nodes without a password is implement.

Here is the specific process in my experiment:

Start up 'cygwin' and execute the instruction '$ ssh-host-config'. When the system pops up the question 'Should privilege separation be used?', 'no' should be entered, while 'yes' should be entered to the question 'Do you want to install sshd as a service?'. Finally, when faced with the tips 'Enterthe value of CYGWIN for the daemon: [ntsec]', choose 'ntsec'. After being told that the sshd service is installed successfully at local, enter the instruction '$net start sshd' to start up SSH. Then make the instruction '$ net start sshd' to generate key/value pairs and in this way they can automatically saved into the directory of '.ssh'. After that these two instructions to add public key RSA into the public key authority file ‘authorized_keys’.

$ cd.ssh

$ cat id_rsa.pub >> authorized_keys

At last, execute ‘$ ssh localhost’ and then the connection based on SSH can be complemented without passwords.

3.3 Setting Up the Hadoop Platform

In the installation directory of Hadoop, there stores various configuration files in the folder ‘conf’ and command files in the folder ‘bin’.

(1). Get the configuration file ‘adoop-eiIv’ modified as you want. In this file, environment variables should be set up, among which, ‘JAVA-HOME’ is necessary to be set but HADOOP-HOME is not, with a default value of the parent directory of the file ‘bin’. In my experiment, I got my variables set as follows.

export ADOOP_HOME=/home/lin/Hadooplnstall/hadoop

export JAVA_HOME=/usr/java/jdk 1.6.0

(2). Get core-site.xml modified, which configures IP and the port of Name Node.

(3). Get hdfs—site.xml modified, which configures the number of copies.

(4). Get mapret-site.xml modified, which configures IP and the port of Job Tracker.

(5) Get Name Node formatted by execute Hadoop name node – format, the format instruction in the folder ‘bin’.

(6) Get conf/hadoop-site.xml edit as follows. It is a main configuration file of the Hadoop architecture. (Baun and Kunze, 2009)

```
<? xml version = "1.0"?>
<?xml-stylesheet type="text/xsl"
    href = "configuration.xsl"?>
<!-- Put site-specific property
    overrides in this file. -->
<configuration>
<property>
<name>fs.default.name</name>
</property>
<property>
<name>mapred.job.tracker</name>
<value>localhost:9001</value>
</property>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

Figure 5: The conf/hadoop-site.xml.

3.4 Design and implement Applications based on Hadoop

(1). Get the codes of Map ready and output intermediate results by handling the input of the Key-value.

(2). Get the codes of Reduce ready and output the final result by statute the intermediate results.

(3). Get Input Format and Out Format defined. The main job of Input Format is to test and verify whether the input type is in accordance with the configuration, while the main job of Out Format is to test and verify whether the output directory already exists and whether the output type is in accordance with the configuration. If both of them are satisfied, the results summarized by Reduce will be output.

4 THE IMPROVEMENT OF HADOOP SCHEDULING ALGORITHM

The Map/Reduce scheduling is asked directly from Job Tracker by Task Tracker actively, whose principle is similar to ordinary, non-preemptive scheduling operating system, that task once assigned, it cannot in off. According to research, there are several typical scheduling algorithms as follows:

FIFO: The default one, First In First Out.

RR: Round-Robin.

HPF: Height Priority First.

WRR: Weighted Round Robin.

However, according to experiments before this paper, none of the algorithms above is perfect. Considering the model of Task Tracker actively requesting for tasks and the feature of non-preemptive in Hadoop task scheduling system, to make tasks avoid the long wait before scheduled, while scheduling priority of each task and can be adjusted according to actual situation, we will propose PBWRR (Priority Based Weighted Round Robin) proper for Hadoop according to the actual situation of the network needs of Hadoop scheduling (Zhang et al., 2008; Hofmann and Woods).

The basic idea of this algorithm PBWRR is like this: Put all jobs to run into a queue, then in the unweighted case, the tasks are turned to Task Tracker in turn to get performed, and in the weighted case, the weight of the larger job can be performed multiple tasks in a rotating.

The specific steps of this algorithm are:

1. In the situation of the resource of Task Tracker available, task tracker takes the initiative to submit assignments job tracker request.

2. When receives the request of task assignment from Task Tracker, Job Tracker takes one of the tasks of the current job to the Task Tracker which sent a request just now to get it execute. At the same time, the value of ‘thisRoundTask’ should be minus 1. After that, if the result is less than 1, then pointer will be moved to the next job, or the pointer will keep unmoved and waiting until the next request from Task Tracker arrives.

3. When the pointer is at the end of the queue, all information of the whole queue will be updated. If there is some job finished or just arriving at the moment, all the values of ‘thisRoundTask’ of each job will be reset and recalculated and the pointer will be moved to the beginning of the queue again.

Using PBWRR to run the Cloud Computing system based on Hadoop again, with the compare with those older algorithms, it is proved that PBWRR is more suitable for Cloud Computing, since in Cloud Computing environment, it is the users’ fee rather than the sequence of them sending a request that the system provide users with different levels of service through. PBWRR extends from the characteristics of transparent of Hadoop to developers and customers, and has a high capacity of clearly distinguishing the level of customer service, while ensuring a degree of fairness to ensure that the situation that most of the resource is kept by a minority of high-priority customers can not happen. Hence, it is a good complement and improvement of the existing scheduling algorithms of Hadoop.

5 CONCLUSIONS

This paper studies the key technologies of cloud Computing and the principle of Hadoop with the method to implement Cloud Computing based on it. The main problem is how to optimize and improve the scheduling algorithm of this architecture. Of course, after all of these, we are more familiar with Cloud Computing and have the ability to skillfully build up such a computing platform in practice. The file configuration and algorithm optimization are originally achieved and brought up by hard working during writing this paper. However, in spite of finding out an improved version of algorithm ‘PBWRR’, there may be lots of risks and deficiencies since it has not been though extensive, rigorous testing. In the future, how to build up private cloud for commercial systems should be on focus (Pearson, 2009), with improved ‘PBWRR’ and to make customers more and more safe with their private information. It is believed that with the rapid development of Cloud Computing, there will be more and more efficient and practical algorithms for scheduling, computing and so on. In addition, there will even be more and more architectures sufficient and safe enough for Cloud Computing. I am looking forward to it.

ACKNOWLEDGEMENTS

This work is supported by the National Key project of Scientific and Technical Supporting Programs of China （Grant Nos.2008BAH24B04, 2008BAH21B03; the National Natural Science Foundation of China (Grant No.61072060) ; the program of the Co-Construction with Beijing Municipal Commission of Education of China.

REFERENCES

Hadoop; http://hadoop.apache.org/

Dean J; Ghemawat S; “MapReduce: Simplified Data Processing on Large Clusters”, 2008

Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph Katz, Gunho Lee. Above the Clouds: A View of Cloud Computing [EB/OL]. http:// www.eecs.berkely.edu/Pubs/TechRpts/2009/EECS-2009-28.HTML, 2009(2)J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892.

Jiyi Wu; Lingdi Ping; Xiaoping Ge; Ya Wang; Jianqing Fu; “Cloud Storage as the Infrastructure of Cloud Computing” Intelligent Computing and Cognitive Informatics (ICICCI), 2010

Shufen Zhang; Shuai Zhang; Xuebin Chen; Shangzhuo Wu; “Analysis and Research of Cloud Computing System Instance” Future Networks, 2010. ICFN '10.

Baun, C.; Kunze, M.; “Building a private cloud with Eucalyptus” E-Science Workshops, 2009 5th IEEE International Conference on

Hofmann, Paul; Woods, Dan; “Cloud Computing: The Limits of Public Clouds for Business Applications” Internet Computin

Mikkilineni, R.; Sarathy, V.; “Cloud Computing and the Lessons from the Past”Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09. 18th IEEE International Workshops on

Kang Chen; Weimin zheng; “Three Carriages of Cloud Computing: Google, Amazon and IBM” http:// soft.cow.corn.cn/it/htm2008/20080512-423960.shtml

Stephen Baker; “Google and the Wisdom of Clouds” 2007

S. Pearson; “Taking Account of Privacy when Designing CloudComputing Servcices[C]” The 2009ICSE Workshop on Softuare Engineering Challenges of Cloud Computing, UK, 2009.