Green Master based on MapReduce Cluster

Ming-Zhi Wu, Yu-Chang Lin, Wei-Tsong Lee, Yu-Sun Lin, Fong-Hao Liu

Dept of Electrical Engineering Tamkang University, Taiwan, ROC
Dept of Electrical Engineering Tamkang University, Taiwan, ROC
Dept of Electrical Engineering Tamkang University, Taiwan, ROC
Chung-Shan Institute of Science and Technology, Taiwan, R.O.C
National Defense University, Taipei City, Taiwan, R.O.C

dean64188@hotmail.com, bearlaker34@gmail.com
wtlee@mail.tku.edu.tw, amani.amani@mail.tbcnet.net,
lfh123@gmail.com

Abstract—MapReduce is a kind of distributed computing system, and also many people use it nowadays. In this paper, the Green Master based on MapReduce is proposed to solve the problem between load balance and power saving. There are three mechanism proposed by this paper to improve the MapReduce system efficiency. First, a brand new architecture called Green Master is designed in the system. Second, Benchmark Score is added to each services in the cluster. In the last, an algorithm about how to distinguish the high score service and the low score service, and how to use them effectively.

Keywords—MapReduce, Benchmark, Cloud Network

1 Introduction

The algorithm in this paper will be used to improve the system efficiency based on MapReduce[1] of Hadoop. Hadoop is a kind of open source software that develop from Google MapReduce, and it can will create a cluster that connects each services. The cluster is used to make more computing resources called computing pool, and it can be expanded more and more. In the end, we can decide what we want to get or how to execute the program through coding the Map Function and Reduce Function.

As usual, in order to make the maximum computing resources, the services must keep the high-speed state, but it also has a lot of unnecessary waste. For example, service performance usually are not the same to each other, some of them are very high, but some of them are very low. if we allocate the same amount of work to all service, it must cause a part of service will complete the work early, but it still have to wait other service that performance is poor, and the waiting time means resources wastes. We will talk about how to make the service off if the performance is too low that seriously affects the system performance.

2 Related works

2.1 Master of MapReduce

Master of MapReduce Master Node is the most important node on MapReduce which cannot be replaced by other nodes. It includes map function, reduce function and mapreduce runtime system. Master node manages receiving command from user and assigning tasks to task trackers, and it stores status of task trackers in database. The status is verified in three different types: Idel, In-processing and completed. The memory address and size of processing data in HDFS(GFS in Google, HDFS in Hadoop) are notified to Master node, and assign map function and task tracker to complete the task.

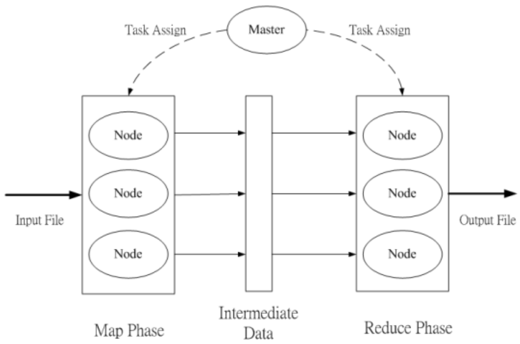

Fig. 1. MapReduce Architecture

2.2 Benchmark

Benchmark[2], generally speaking, is a value about something 's performance or ability and make comparison. However, a performance comparison of virtualization technology for the moment is not very common, VM Benchmark is a new type of test methods. It is discussed virtual environment build through virtualization and virtual machine management VM resources (hard discs, memory) . We have adopted Virtual Machine system build, and we introduce the mechanism of the Benchmark to distinguish the VMs' performance.

3 Implementation

In this section, the algorithm of Green Master will be explained how to implement. It includes Green Master System, Input File Index, Server Information, Queue, Record, Load Balance Optimization, Power Saving Algorithm, and Decision Algorithm. And we will discuss the detail at the following.

3.1 Green Master System

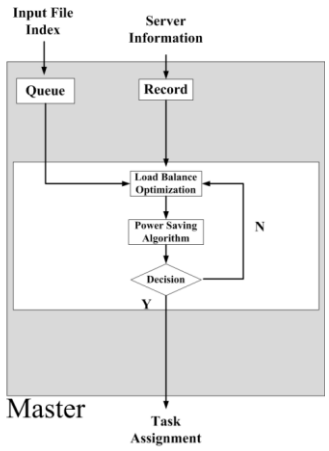

Fig. 2. Green Master Architecture

The Green Master is a brand new architecture transformed from Hadoop's Master, and it can apply to each nodes that install the Hadoop. The brand new architecture called Green MapReduce System(GMS), and it can help users manage the node in the cluster to save the system consumption and service computing overhead. The Green Master does not change the Map Function and Reduce Function, it just changes the task allocation master according to server loading and server's Benchmark Score to achieve the goal about the energy saving.

It is not accepted that the system performance reduces caused by someone virtual machine low efficiency, especially in the Cloud Computing Network environment. It is not accepted that the system performance reduces caused by someone virtual machine low efficiency, especially in the Cloud Computing Network environment. In order to solve the above problems, Green Master is designed to delete the poor services and allocate the job distribution. Green Master is divides into eight blocks, and it includes Input File Index, Queue, Server Information, Record, Load Balance Optimization, Power Saving Algorithm and Decision Algorithm. Green Master has a strong adaptability to many systems, for an instance, when we need a great amount of computing resources to calculate tasks, we can use Green Master to avoid energy wastes. For another instance, when the system equipment has a strong non-conformance, and the system can use the Benchmark Score in the Green Master to arrange the tasks allocation according to the services capability.

3.2 Server Information

The Server Information in the Green Master is to estimate the services' capability called Benchmark Score, and it will keep running and send the results to Green Master. In addition, whenever a new server join or quit the cluster, Benchmark Score will change. The range of the Benchmark Score is from zero to one hundred, and it is according to CPU computing performance, Memory read/write and Disk I/O rate to estimate the Benchmark Score. In other hand, the highest CPU response time, Memory read/write and Disk I/O is defined as 100 Benchmark Score. The definition of poorer virtual machines' Benchmark Score are based on the highest one.

$$BenchmarkScore = \frac{X_n}{Top_{VM}} \times 100\% \qquad (3.1)$$

where the Top is the highest value of the virtual machine, and the x is the value of the virtual machine like CPU response time to be measured. Because of the CPU response time, Memory read/write and Disk I/O rate have to be considered in the formula, so we turn formulas evolution as follows:

$$Benchmark\ Score = \sum^{i} \frac{X_n}{Top_{VM}} \times W_i,\ W_i \in \{Measured\ Event\},\ i \in \{1, 2, ..., m\} \qquad (3.2)$$

where the W_i is the event of Benchmark Score. In our case, the i of W_i is three, there are CPU response time, Memory read/write and Disk I/O rate respectively.

3.3 Recorder

Recorder is used for recording server information. Recorder refresh when it receives newer server information. A new recording table is established for information record when there is new node joins in to the cluster. Servers update and refresh server information in recorder during the working time.

3.4 Load Balance Optimization

Load Balance[3] Optimization will allocate the work loading according to the information collecting from the above-mentioned blocks. The Benchmark Score is more higher, and the work loading is more; the Benchmark is lower, and the work loading is less. The job is allocated to VMs through Load Balance Optimization, and the formula is following:

$$Task\ Dsitribution\ Ratio = \frac{Local_{VM}}{\sum_{i=1}^{n} Score_i} \times 100\% \qquad (3.3)$$

where Total Score is the sum of the VMs' Benchmark Score, and the Local Score is the VM's Benchmark what you want to estimate. In our experiment, we use six VM in the experiment environment and calculate the work loading ratio as following:

3.5 Power Saving Algorithm

In this paper, Power Saving Algorithm (PSA)[4][5] will check the utilization of the server. In the first state, we allocate the work loading to VM according to the Benchmark Score, then the second state, we will determine the utilization of the VM. In Figure 4, we can find that the huge difference of the work loading between Benchmark Score 100 and Benchmark Score 5, but they use almost same energy. This paper presents PSA to discuss how to get the balance between efficiency and energy management.

$$T_n = \sum_{i=1}^{n} a_i \times \frac{B_i}{\sum_{j=1}^{n} B_j} + \varepsilon \qquad (3.4)$$

$$E_n = T_n \times N_{VM} \times P_{VM},\ N_{VM} \in \{1, 2, ..., n\} \qquad (3.5)$$

$$V_n = \frac{E_{n+1} - E_n}{T_{n+1} - T_n},\ \text{n} \geq 2 \qquad (3.6)$$

where T_n is system computing time, and a_i is the system time which one virtual machine completed alone, and the B_i is the Benchmark Score of one virtual machine, and the ε is the error time. E_n is the energy(J) of virtual machine. P is the power(W) of virtual machine. V_n is the ratio of energy consumption.

3.6 Decision Algorithm

Decision Algorithm[6] in GMS is to judge the result which is from PSA reasonable or not. The formula is as following:

$$\alpha \leq \gamma \qquad (3.7)$$

where α is the system consumption through PSA, and γ is without PSA. If γ is greater than α, then the system will back to Load Balance Optimization.

4 Simulation Result

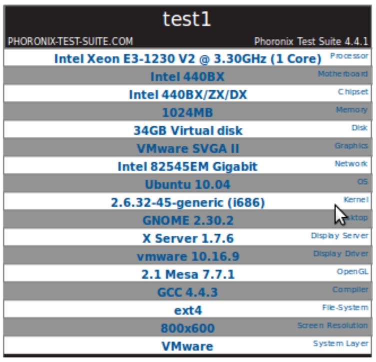

Fig. 3. Experiment Environment

Figure 4 shows the highest performance virtual machine in the experiment environment of this paper.

4.1 The Relationship between system computing time and system consumption

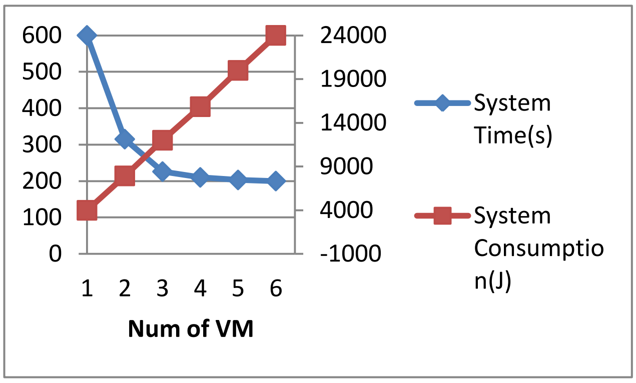

Fig. 4. System Time and Consumption

In Figure 5, we can find that the cross point between the system time and system consumption is between two VMs and three VMs. In fact, the number of VM of the best performance in our experiment is three VMs.

4.2 Comparison between Original and Green Master

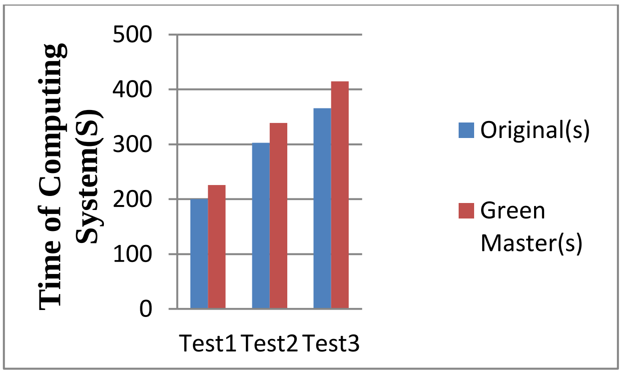

Fig. 5. System Computing Time

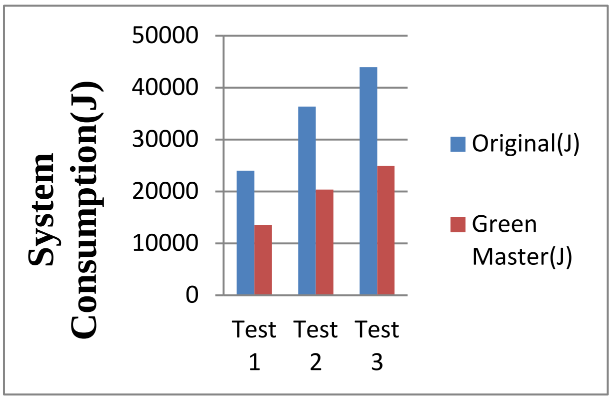

Fig. 6. Power Consumption Saving

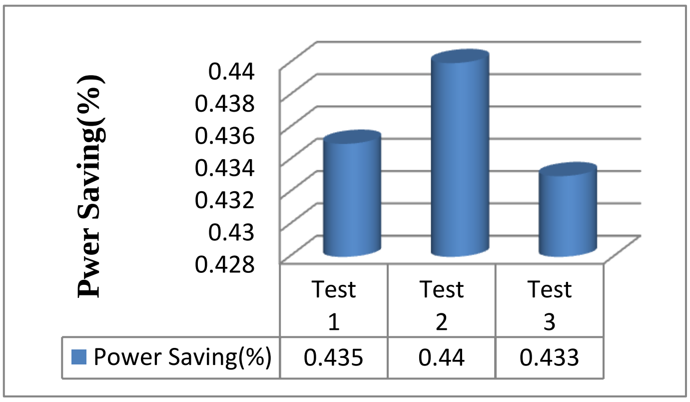

Fig. 7. Ratio of Power Saving

In Figures 6 and 7, we take several different sizes of test file in our experiment environment, we can clearly find the original system time is less than Green Master, but system consumption is almost twice larger than Green Master.

5 Conclusion

The idea of Green Master optimizes system power consumption by lower the performance slightly. In this paper, we provide a appropriate trade-off between power saving and performance loses, and improves energy conservation of the system.

References

[1] *Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simpli_ed Data Processing on Large Clusters," OSDO 2004.*

[2] Xia Xie, Qingcha Chen, Wenzhi Cao, Pingpeng Yuan, Hai Jin, "Benchmark Object for Virtual Machines," 2010 Second International Workshop on Education Technology and Computer Science.

[3] *Lars Kolb, Andreas Thor, Erhard Rahm, "Load Balancing for MapReduce-based Entity Resolution," 2012 IEEE 28th International Conference on Data Engineering.*

[4] *Tao Zhu1,2, Chengchun Shu1, Haiyan Yu1, "Green Scheduling: A Scheduling Policy for Improving the Energy Efficiency of Fair Scheduler," 2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies.*

[5] *Sandhya S V, Sanjay H A*, Netravathi S J, Sowmyashree M V, Yogeshwari R N, "Fault-Tolerant Master-Workers framework for MapReduce Applications," 2009 International Conference on Advances in Recent Technologies in Communication and Computing.*

[6] *Congchong Liu and Shujia Zhou, "Local and Global Optimization of MapReduce Program Model," 2011 IEEE World Congress on Services.*

[7] *NovQinlu He,Zhanhuai Li'Xiao Zhang, "Study on Cloud Storage System based on Distributed Storage Systems", Computational and Information Sciences(ICCIS), 17-19 Dec 2010, pp. 1332 - 1335*

[8] Kevin D.Bowers,Ari Juels, and Alina Oprea., "HAIL: A HighAvailability and Integrity Layer for Cloud Storage", Computer and communications security (*CCS*), Nov 2009, pp. 187-198

[9] Mingyue Luo,Gang Liu, "Distributed log information processing with Map-Reduce: A case study from raw data to final models", Information Theory and Information Security(*ICITIS*), 17-19 Dec. 2010, pp.1143-1146