Journal of Korea Multimedia Society Vol. 15, No. 12, December 2012(pp. 1499-1505)

The Creation and Placement of VMs and Tasks in Virtualized Hadoop Cluster Environments

Tae-Won Kim†, Hae-jin Chung††, Joon-Mo Kim†††

ABSTRACT

Recently, the distributed processing system for big data has been actively investigated owing to the development of high speed network and storage technologies. In addition, virtual system that can provide efficient use of system resources through the consolidation of servers has been increasingly recognized. But, when we configure distributed processing system for big data in virtual machine environments, many problems occur. In this paper, we did an experiment on the optimization of I/O bandwidth according to the creation and placement of VMs and tasks with composing Hadoop cluster in virtual environments and evaluated the results of an experiment. These results conducted by this paper will be used in the study on the development of Hadoop Scheduler supporting I/O bandwidth balancing in virtual environments.

Key words: Hadoop, Distributed Database, Virtual Machine, Big Data

1. INTRODUCTION

Recently, the distributed processing system for big data has been actively investigated owing to the development of high speed network and storage technologies. Google's MapReduce framework and Apache Hadoop MapReduce framework[1] are representative of the distributed processing system, of these, Hadoop MapReduce framework is widely used as open source. In addition, virtual system that can provide efficient use of system resources through the consolidation of servers has been increasingly recognized. Amazon's EC2(Elastic Compute Cloud) virtualized data center is an example of combination of these virtual technologies with distributed processing system for big data. But, since MapReduce framework is designed based on the homogeneous computing environments, in heterogeneous computing environments such as virtual environments many issues will occur[2]. Optimizing the performance of MapRuduce by distributing limited computing resources in a virtual machine environments to each virtualized node is one of the issues.

In this paper, we did experiments on the optimization of I/O bandwidth according to the creation and placement of VMs and tasks with composing Hadoop cluster in virtual environments and evaluated the results of experiments. This paper is organized as follows. We will examine MapReduce framework and sharing computing resources for MapReduce performance in virtual en-

※ Corresponding Author : Joon-Mo Kim, Address : (448-701) 509, 2nd EngineerBuilding, 152, Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do, 448-701, Korea, TEL : +82-31-8005-3665, FAX : +82-31-8021-7221, E-mail : q888@dku.edu
Receipt date : Sep. 23, 2012, Revision date : Nov. 8, 2012
Approval date : Nov. 19, 2012
† Department of Computer Science and Engineering, Dankook University
(E-mail: sega12645@gmail.com)
†† Department of Computer Science and Engineering, Dankook University
(E-mail: zztzizi@gmail.com)
††† Department of Computer Science and Engineering, Dankook University
※ This work was supported by the Industrial Strategic technology development program(10039143, Multi-Gbps Parallel I/O Virtualization and DB Performance Optimization for the Data-centric Computing on Many-core System) funded by the Ministry of Knowledge Economy, Korea.

vironments in chapter 2, in chapter 3, we will evaluate and analyze the results through the experiment of the performance and then conclude and find the future research directions in chapter 4.

2. BACKGROUND

2.1 MapReduce

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key[3].

Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I and passes them to the Reduce function.

The reduce function, also written by the user, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values. Typically just zero or one output value is produced per Reduce invocation. The intermediate values are supplied to the user's reduce function via an iterator. This allows us to handle lists of values that are too large to fit in memory.

Hadoop framework used in the experiments of this paper is the typical framework using MapReduce programing. The performance of Hadoop framework is represented by measuring the performance of MapReduce programing.

2.2 Sharing computing resources and determining the number of virtual nodes in virtual environments

In general, the number of data nodes composed of a Hadoop cluster increases as a performance of parallel data processing increases[4]. However, virtual nodes share the resources of limited a physical node when they consist of the Hadoop cluster in virtual environments. Therefore, increasing the number of virtual nodes cannot guarantee the increasing performance of general Hadoop cluster in virtual environments.

The resources of physical node that affected by performance of virtual nodes are CPU, Memory and I/O devices. Of these, CPU and Memory resources through virtualization will ensure that each virtual node has uniform and isolated performance. But, I/O devices are assigned the irregular I/O performance by the degree which each virtual nodes share a assigned physical I/O resource[5]. When virtual nodes cannot be equally distributed I/O bandwidth, nodes with slow I/O performance will occur. We called them Straggler[6]. The Fig. 1 is an example of stragglers in heterogeneous computing environments

In the present paper, when the parallel distributed processing system for big data is composed in virtual machine environments, we will analyze performance issues regarding determining the number of virtual nodes and distribution of I/O

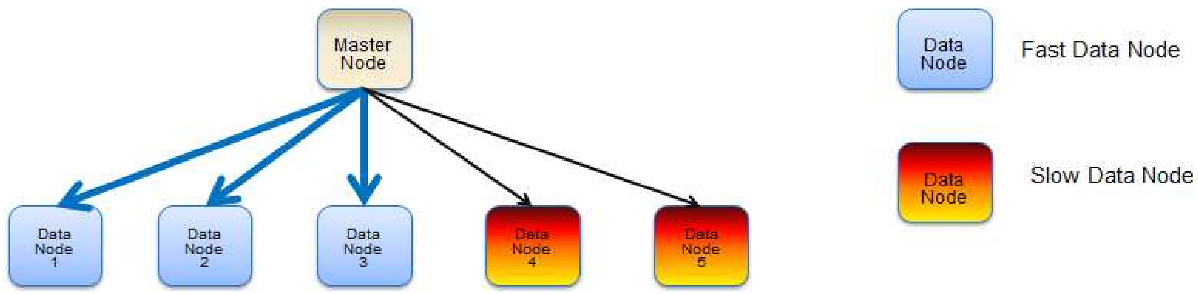

Fig. 1. An example of stragglers in heterogeneous computing environments.

Table 1. Experimental conditions of virtual Hadoop cluster

Physical node	The number of node	1		
	CPU	intel 2.93GHZ 6 core * 2		
	Memory	4GB * 11 = 44GB		
	Disk	600GB * 4		
	OS	Linux Fedora 14		
Virtual node	Total number of nodes	3~12	The number of name node	1
			The number of data node	2~11
	OS	Linux 2.6.18-238		
Hadoop		Hadoop 1.0.0		

bandwidth through experiments. Also we will assume that each virtual node is equally assigned the CPU and Memory resources and has enough network bandwidth. Therefore we will consider assigned Disk I/O bandwidth in the study.

3. EXPERIMENT AND ANALYSIS

3.1 Experimental Conditions

In oder to evaluate the performance, 3 to 12 virtual nodes produced on a single physical node were used in an experiment. The virtual solution called Xen[7] was used and Hadoop was used as big data distributed processing MapReduce framework. The cluster environments of virtual Hadoop is shown in the following Table 1.

The physical node used in the experiment had four hard disk. Each physical node's hard disk performance was about 100MB/s and we didn't use the hard disk raid to deploy each node I/O bandwidth. One of the hard disk was assigned to the domain 0 of Xen and the remaining three hard disks were divided into four partitions and then were assigned to the nodes in the Hadoop cluster. In oder to assign equal I/O bandwidth, the number of disk were respectively separated by 1, 2, 3, 4 and a partition was separated and then assigned by a, b, c, d as shown in Table 2.

The reason disk partition was allocated to virtual nodes as shown in Table 2 is as follows.

First, it is for I/O bandwidth balancing. When the experiment with three data nodes is assumed to be done and data nodes 1, 2, 3 are deployed to hard disk 2 in oder, I/O bottle neck will occur because of the centralization of hard disk 2. As a result, the overall performance of Hadoop cluster will low. This can be shown in 3.3 Experiment 2.

Second, because domain 0 of Xen should not receive affect on performance associated with a particular node. Therefore, as illustrated in Table 2, hard disk was allocated only domain 0. Finally, because NameNode does not have an effect on performance during the operation of MapReduce.

3.2 Experiment 1 : According to the Number of Virtual Nodes

In order to evaluate the performance of the number of virtual data nodes on a physical node, we set the number of virtual nodes up 2 to 11 and then measured the performance of Hadoop cluster by

Table 2. Node distribution according to the hard disk partition

HDD 1				HDD 2			
a				a	b	c	d
Domain 0				Name Node	Data Node1	Data Node4	Data Node7
HDD 3				HDD 4			
a	b	c	d	a	b	c	d
Data Node2	Data Node5	Data Node8	Data Node10	Data Node3	Data Node6	Data Node9	Data Node11

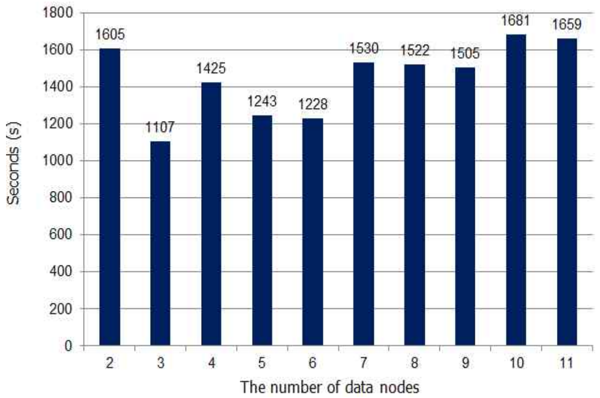

Fig. 2. 16GB Terasort results with virtual Hadoop cluster.

using Terasort with 16GB data generated by Teragen. Table 3 and Fig. 2 are the results of the experiments.

The results can be obtained as follows through experiment 1.

(1) When we ran the test using the 3 data node (in the case connecting each virtual data node to each hard disk), MapReduce performance was the best. We became aware of the fact that I/O performance was critically applied to performance of Hadoop cluster

(2) Because straggler was raised in experiment used 4 data nodes, the experimental resulting value was smaller than that of 5, 6 data nodes. We could know that I/O bandwidth balancing was considerably important in the virtual Hadoop cluster environments through experimental results.

(3) In the experiment used 7 or more data nodes we were confirmed that MapReduce performance was more poor than others because large amounts of I/O operations each hard disk were assigned to each hard disk.

Table 3. 16GB Terasort results with virtual Hadoop cluster

The number of data nodes (Unit: Number)	Composition of Hadoop cluster	Total working time (Unit: Seconds)
11	Name Node Data Node 1~11	1659
10	Name Node Data Node 1~10	1681
9	Name Node Data Node 1~9	1522
8	Name Node Data Node 1~8	1530
7	Name Node Data Node 1~7	1505
6	Name Node Data Node 1~6	1228
5	Name Node Data Node 1~5	1243
4	Name Node Data Node 1~4	1425
3	Name Node Data Node 1~3	1107
2	Name Node Data Node 1~2	1605

3.3 Experiment 2 : Different I/O Bandwidth Balancing

We had the second experiment that set up different I/O bandwidth balancing. We composed 2 Hadoop clusters(cluster A and B) having 5 data nodes. Five data nodes were equally assigned to cluster A's hard disk. On the other hand, Hadoop cluster B was intensively assigned 5 data nodes to hard disk 1, 2. Fig. 3 is a Hadoop cluster that

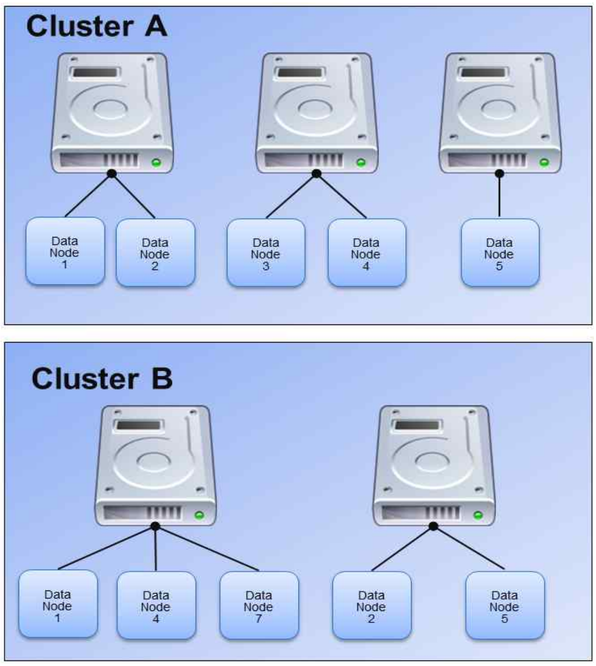

Fig. 3. Hadoop cluster configuration of experiment 2.

Table 4. MapReduce performance test according to the difference of I/O bandwidth balancing

Hadoop cluster	The number of nodes	Composition of Hadoop cluster	Total working time (Unit: Seconds)
A	5	Name Node Data Node 1,2,3,4,5	1243
B	5	Name Node Data Node 1,4,7,2,5	1985

is built in Experiment 2. And Table 4 and Fig. 4 show the results about experiment 2. From the results of experiment 2, we could find that Hadoop cluster B was roughly 60% slow than Hadoop cluster A.

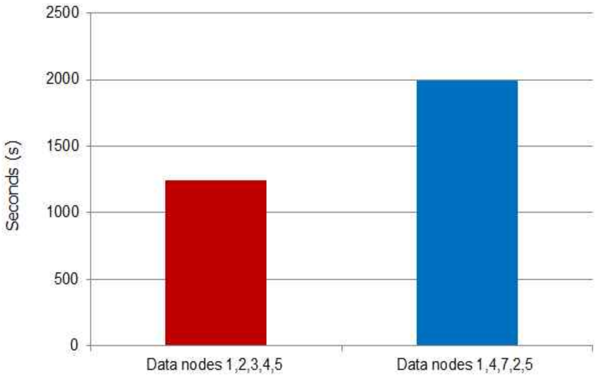

Fig. 4. MapReduce performance test according to the difference of I/O bandwidth balancing.

3.4 Experiment 3 : Virtual nodes placement considering heterogeneous environments

For the third experiment heterogeneous environments of Hadoop virtual machine cluster builds as cluster C in Fig, 5 below. And to compare the performance, cluster D evenly distributing the additional virtual nodes applied to the Hadoop cluster policy in an existing physical machine environments and cluster E applied to the imbalance virtual node distribution appropriate in virtual machine environments are built.

To compare each performance of cluster, the same 10GB of data were generated through

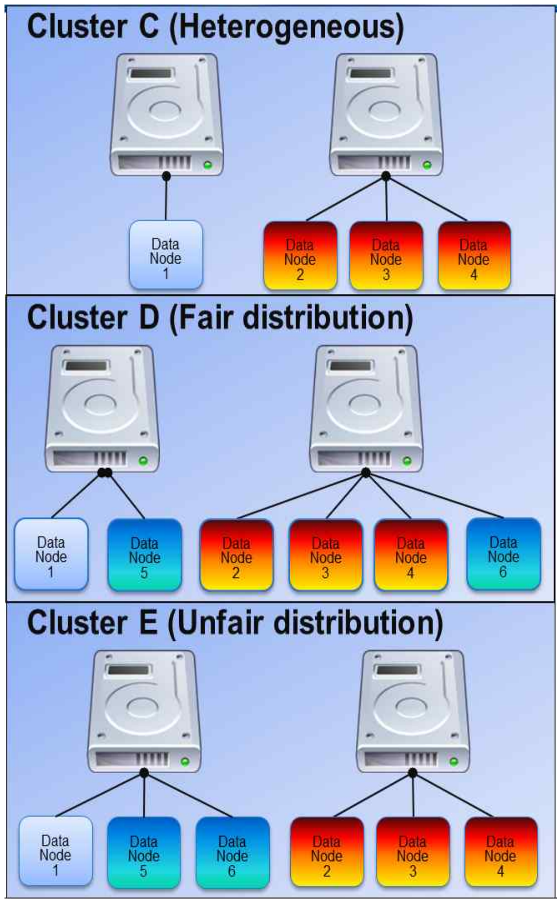

Fig. 5. Hadoop cluster configuration of experiment 3.

Hadoop's Teragen. After each cluster performance was analyzed and compared through Terasort. The results of the experiment can be seen through Fig. 6 below. To look at the experimental results, the result of the cluster E 783 seconds is better than that of the cluster D 849 seconds. Approximately

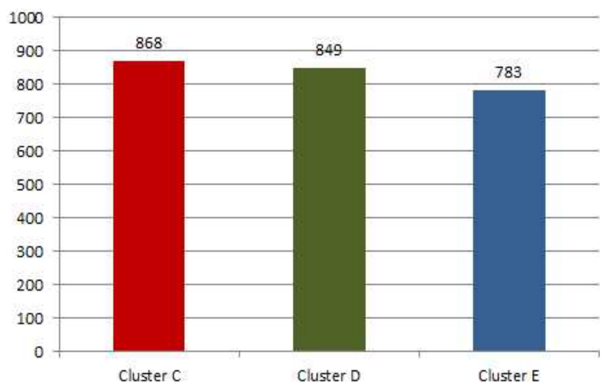

Fig. 6. The results of experiment 3. (Unit: Seconds).

10% performance has been improved compared to the existing. This means that when the virtual nodes are added in a Hadoop cluster of virtual machine environments, the unfair distribution of nodes by considering the virtual machine environments improves the performance of the cluster.

3.5 Experiment 4: Task placement considering heterogeneous environments

In the fourth experiment, we built Hadoop cluster of the same virtual machine environments, and then compared the performance of each task depending on the placement of the cluster. Hadoop cluster constructed as Fig. 7 below, Cluster F and G are of the same cluster environments. Cluster F evenly distributed Reduce task taking into account the existing Hadoop Scheduler policy (Fair Scheduling). Cluster G assigned two Reduce tasks to the better performance data node 1 and one Reduce task to the bad performance data node 2, 3, 4 by considering virtual machine environments. Experiment 4, the same as experiment 3, also compared the performance of the cluster through Terasort after generating 10GB of data.

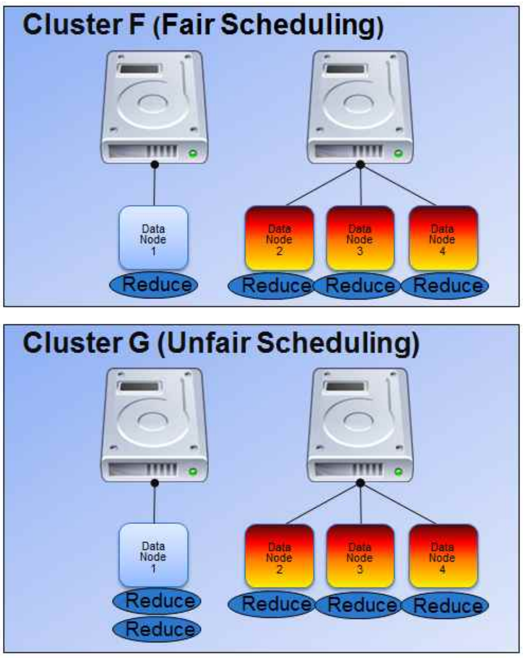

Fig. 7. Hadoop cluster configuration of experiment 4.

The results of the experiment are shown in Fig. 8 below. Experimental results tell that the performance of the cluster G is about 11% better than that of the cluster F. Through this experiment we can see that Fair Scheduling of Hadoop is unsuitable in virtual machine environments. As well as this experiment showed that Unfair Scheduling should be done by considering heterogeneous environments in Hadoop cluster environments in a virtual machine environments.

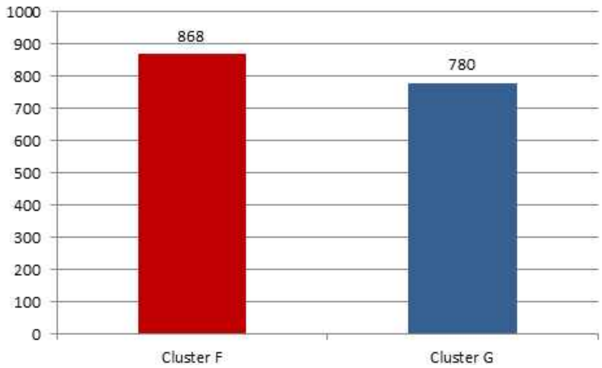

Fig. 8. The results of experiment 4. (Unit: Seconds).

4. CONCLUSION AND FUTURE RESEARCH

In the present paper, experiments on performance evaluation of the creation and placement of VMs and tasks by considering I/O bandwidth balancing in composing Hadoop cluster in virtual machine environments has been done and from the experimental results the following conclusions can be drawn:

First, the performance of disk I/O was considered to be a critical factor in composing Hadoop cluster in virtual machine environments ; in oder to increase the number of virtual data nodes by using the way to effectively improve the performance of MapReduce, hard disk number of physical node have to grow.

Second, I/O bandwidth balancing of each virtual data nodes in virtual machine environments had an effect on the performance of MapReduce up to 60%

; it is very important to compose virtual Hadoop cluster.

Third, when Hadoop cluster is composed in virtual machine environments, the creation and deployment of additional nodes should be made by considering the heterogeneous environments. Only then the performance of parallel programming gets better.

Fourth, the performance of unfair scheduling under heterogeneous environments is superior to that of existing fair scheduling in virtual machine environments. That is, it shows that instead of existing scheduler algorithm the virtual Hadoop scheduler algorithm is required.

In the future, the results of an experiment conducted by this present paper are supposed to be used as a reference of development of Hadoop scheduler supporting I/O bandwidth balancing in virtual machine environments and then as a way to improve performance of Hadoop MapReduce in virtual machine in virtual machine environments the algorithm for supporting I/O bandwidth balancing will be suggested.

REFERENCE

[1] D. Borthaku, *The Hadoop Distributed File System: Architecture and Design*, Hadoop Project Website, 2007.

[2] Chao Tian, "A Dynamic MapReduce Scheduler for Heterogeneous Workloads," *International Conference on Grid and Cooperative Computing,* pp. 218-224, 2009.

[3] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM,* Vol. 51, No. 1, pp. 107-113, 2008.

[4] S.H. Kim, T.H. Keum, and C.H. Jeon, "Implementation and Performance Analysis of Testbed Clusters for Cloud Computing," *The Korean Institute of Information,* Vol. 36, No. 1, pp. 545-548, 2009.

[5] Mate Zaharia, "Improving MapReduce Performance in Heterogeneous Environments," *Proc. OSDI'08,* Vol. 13, No. 4, pp. 277-298, 2008.

[6] S. Ghemawat and S.T. Leung, "The Google File System," *Proc. SOSP'03,* Vol. 37, No. 5, pp. 29- 43, 2003.

[7] Paul Barham, "Xen and the Art of Virtualization," *Proc. SOSP'03,* Vol. 35, No. 5, pp. 164-177, 2003.

[8] Hwang, "Web-based Distributed Visualization System for Large Scale Geographic Data," *Journal of Korea Multimedia Society,* Vol. 14, No. 6, pp. 835-848, 2011.

Taewon Kim

received the B.S., M.S degrees in Computer Engineering, Dankook University, Korea in 2007, and 2010 respectively. He is currently an Ph.D. Candidate in Dankook University. his research interests include distributed database, web search, bigdata, Hadoop and Mapreduce progrmaing.

Haejin Chung

received the B.S., M.S degrees in Computer Engineering, Dankook University, Korea in 2009, and 2011 respectively. She is currently an Ph.D. Candidate in Dankook University. Her research interests include database, bigdata, Hadoop and Mapreduce.

Joonmo Kim

is an Associate Professor in the Department of Computer Engineering, Dankook University. He received his Ph.D. in Computer Science from University of Minnesota in 2001 and worked as a researcher at KISA for two years, prior to joining Dankook University in 2004. He received his BE in Computer Engineering from Seoul National University in 1989. His research interests include issues of computability.