Implementation of High-Efficient R/W System on Green Cloud Computing

Tin-Yu Wu*, Tsung-Han Lin

Department of Electrical Engineering, Tamkang University, Taipei, Taiwan

Abstract— **With the popularization of cloud computing, more and more network users choose to store data and operate various programs on the Internet. Under such a circumstance, instead of the functions of a system, what users care most now is if they can access to the Internet smoothly and quickly. Thus, based on the present system, we propose a novel fast R/W mechanism, which manages data efficiently by Cloud Distributed Computing Server (CDCS), chooses the optimal transmission path and enhances the reliability and efficiency of data access, to conform to the requirement of green cloud computing.***

***Index Terms*— Green Cloud Computing, CDCS**

Ⅰ. INTRODUCTION

With the fast development and expansion of broadband and wireless network in recent years, the ubiquitous network has been getting mature and the network access methods also have greatly changed. In addition to saving or editing data at the local host, to store data in the remote hosts becomes the current trend that allows users to access or edit the same document anytime and anywhere and the network utilization can be more flexible and diversified. The concept of cloud computing therefore takes shape: the system automatically divides the complicated computing process into numerous subprograms, which are further analyzed by the cloud constructed by several hosts and the final result will be returned to users. Even when the User Equipment (UE) is not efficient enough, the network resources can help the computing process and access the remote database.

For the time being, the commonly-seen applications of cloud computing services include e-mail, online shopping, data sharing, online games, and so on. Thanks to the acceleration of NET surfing, users can use browsers, intelligent mobile phones and other mobile devices to check the data stored in the cloud anytime and anywhere. This paper attempts to enhance the present cloud computing system by implementing a high-efficient R/W system.

* Correspondence author: tyw@mail.tku.edu.tw (T.Y.Wu)

Since the performance of hosts or servers is different and the quality of network connections also differs, we propose a selection mechanism to implement on the data nodes to achieve synchronized access. Through Cloud Distributed Computing Server (CDCS), we can not only manage and distribute data effectively, but also find the optimal path in terms of different UE. Moreover, distributed multi-node simultaneous reading method is adopted to accelerate the reading and reduce the response time, so that users can experience the optimal network services. Even if the processing capacity of the UE is limited, the network utility still can be improved because of such a good selection mechanism, which conforms to the requirement for green cloud computing.

In addition, the great losses of lives, properties, and important information caused by several recent catastrophes are still impressive. The issues of data backup and disaster recovery therefore grab the attention of the whole world. Thus, in addition to fast access and computing, data backup and network security are the major concerns in cloud computing system. In order to enhance the efficiency and reliability of cloud computing system, we modify the access mechanism and backup scheme of Hadoop system: the auto remote backup mechanism increases the data reliability by data backup at remote nodes, offline backup, multiple error detection, and intelligent recovery.

Section II introduces related works and background and Section III explains our proposed Cloud Distributed Computing System (CDCS). The simulation results and performance analysis are given in Section 4 and the conclusion is given in Section 5.

Ⅱ. RELATED WORKS AND BACKGROUND

This section first gives the introduction for cloud computing and analyzes its dis/advantages. Next, the concept and architecture of Hadoop, including HDFS and MapReduce, will be further explained.

A. Cloud Computing

Instead of a brand new technique, cloud computing is in fact a concept: to handle the data on the Internet. Cloud computing system is actually a virtual system constructed by innumerable servers on the Internet, and to enter cloud computing system is to allow this virtual system to process your data. Basic techniques of cloud computing are mainly derived from Distributed Computing and Grid Computing.

Huge computing resources means more diversified new services for users of other hosts to compute or access data, or obtain online services by the computing resources released by enterprises. In this way, cloud computing comes into existence. Within several seconds, network users can dissemble the complex computing process into innumerable subprograms by the great virtual system constructed by numerous servers on the Internet, and the final result will be returned to users after the computing and analysis. Like a super computer, this virtual network is able to deal with a huge amount of information no matter the computing resources of the local UE is adequate or not. As shown in Figure 1, the final result seems to fall from the Internet, the "Cloud."

- Distributed Computing
  In distributed computing, a large computational task is divided into many segments, and each of which is solved by one computer in the network.
- Grid Computing
  Grid computing is also a form of distributed computing. Its main feature is, by distributed computing, to integrate the resources of different platforms, architectures and computers into virtualized resources, and deal with data in the remote servers. Grid computing aims to solve one huge task that needs complicated processing, while cloud computing tends to be popular applications. In other words, cloud computing can be regarded as the popularized grid computing.

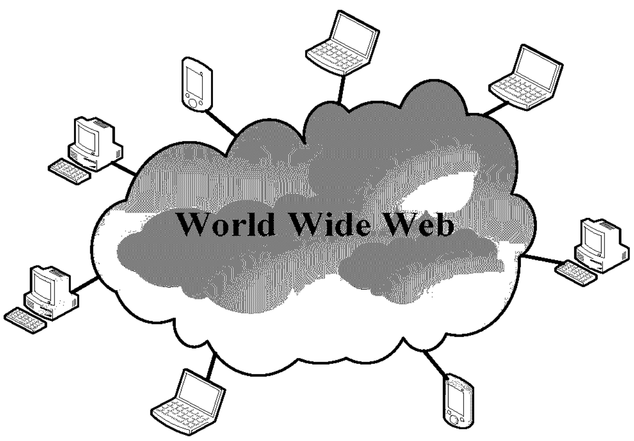

Figure 1. Diagram of Cloud Computing

(A) Advantages of Cloud Computing

- ✓ The computing capacity is provided as a kind of service that can be obtained by enterprises or individuals on the Internet. Users put the data in the cloud and access them whenever they have a computer and an Internet connection.
- ✓ The ubiquitous network resources have eliminated traditional work settings so that the companies and individuals can access data anywhere. The web-based trend is to store and compute data in the servers and to transmit the result to the clients after the computing. Cloud computing services substantially help the communications between the companies and clients.
- ✓ Cloud computing helps to reduce the cost. By renting the server space, the companies can continue to update the software, save the expenditure, and increase the storage whenever needed.
- ✓ Cloud computing directly improves the efficiency of the companies: not only the transmission time of data is reduced, but also the speed to develop new products is accelerated.

(B) Disadvantages of Cloud Computing

- ✓ Users' privacy is easily violated because users' habits, hobbits, and personal information all can be recorded by service providers.
- ✓ Even the speed for users to connect to the cloud is fast, the cloud computing applications are still slower than accessing data on your desktop PC. This might be the obstacle to popularize cloud computing.
- ✓ Once the cloud computing service providers, like Google App Engine, get offline, or the submarine cables in the Pacific Ocean are broken, users will encounter great inconveniences.

(C) Types of Cloud Computing Services

The Cloud Computing services are mainly divided into three categories:

- SaaS (Software as a Service)
  The service providers rent the software to the companies or charge the advertisement fees to run the system.
- PaaS (Platform as a Service)
  Through the execution environment and programming provided by the service providers, clients can develop their own applications. GAE of Google and Azure

of Microsoft, for example, both belong to PaaS.

- IaaS (Infrastructure as a Service)
  In general, traditional ASP can only provide services like SaaS or PaaS. To sustain the infrastructure mostly needs the support of other hardware companies.

B. Apache Hadoop

Being developed by Apache Software Foundation, Hadoop is an open-source implementation, including a distributed file system. Hadoop the platform is used to handle and store a huge amount of data, and can be implemented in a computer cluster built by general PCs. By parallelizing data processing across many nodes, Hadoop can obtain the responses rapidly. Whenever an error occurs to a node, the system immediately acquires the backup data and deploys the computing resources. Roughly speaking, Hadoop includes two major parts: Hadoop Distributed File System (HDFS) and MapReduce.

(A) HDFS (Hadoop Distributed File System)

HDFS is a huge distributed file system that provides single namespace to ensure the consistency of data. The access mode of HDFS is write-once-read-many, which means that once the file is established and written, it cannot be modified. Every file is spilt into many blocks, and each of which resides on a different Datanode. The clients obtain the positions of the blocks from the Namenode and request the Datanodes for data. Once an error occurs to the data, the replication is adopted to reply.

In Figure 2, HDFS architecture shows that the communications between the Namenode, Datanodes, and client are based on the TCP/IP protocol. Before writing, instead of sending the command to the Namenode directly, the client first stores the user data temporarily in the local file. When the size of the user data stored in the temporary files reaches 64M, the client notifies the Namenode. The Namenode responds to the RPC requests issued by the client, inserts the file name into the data layers, and finds a block in a Datanode to store the data. Simultaneously, the Namenode notifies the client the information of the Datanodes and the relative blocks, so that the client can write the blocks stored in the local file to the assigned Datanodes.

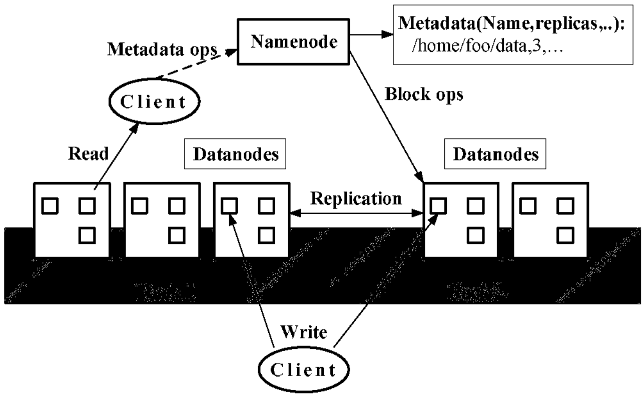

Figure 2. HDFS Architecture

Based on the backup mechanism, the data is first written to a local file, next to the node of a remote rack, and another node of the same remote rack. The rest blocks can be deployed randomly and users usually choose the nearest node to read data.

(B) MapReduce

As an important technique of Google, MapReduce is a programming model for processing large amounts of data by parallel programming. The framework of MapReduce that coordinates the resources allocation and handles the input, output and execution of the software allows the developers without parallel computing experiences to put in use. The name of MapReduce is derived from two main functions in the model: Map and Reduce. In the "Map" step, the data is handled in parallel, which means the task is split into several small problems to be solved. In the "Reduce" step, the results are combined together to obtain the final output. In other words, the computing process is completed by separating, Map, and combining, Reduce.

Ⅲ. CLOUD DISTRIBUTED COMPUTING SYSTEM

Our proposed scheme is based on the storage method of Hadoop, the model of cloud computing system. The file system of Hadoop, Hadoop Distributed File System, is a virtual file system. It seems that the document storage is similar to physical file systems: users set up catalogs and replicate the documents from physical file systems. However, the documents are in fact distributed on different hosts by the specific Hadoop storage mechanism, which still needs to be modified and improved. While Hadoop distributes and stores data by random selection, our proposed scheme calculates the weight with the consideration for the Hard Disk (HD) space, the bandwidth, the Host load and the storage area. Based on the calculated weight, we choose the appropriate storage area and achieve the remote backup.

Our proposed storage algorithm includes three mechanisms: selection mechanism of data storage, synchronized data storage mechanism, and auto remote backup mechanism. Selection mechanism of data storage helps to find out the optimal transmission node in the network as the storage space. However, because the bandwidth of different hosts differs, asynchronous data might emerge while accessing to the same data amount. Therefore, synchronized data storage mechanism is proposed to solve this problem. Also, we add a remote backup mechanism with the consideration to the possibilities of invalidity of hosts and damages of hard disks.

In terms of reading data, we chiefly adopt multi-node simultaneous reading method to accelerate the speed. The remote hosts automatically look for the substitute hosts to backup the local and remote files. Thus, there will be more hosts for us to choose and download while reading data.

A. Selection Mechanism of Data Storage

The parameters that selection mechanism of data storage takes into consideration include hard disk space of the hosts, the bandwidth, the load of the hosts and the regional factor. The selection architecture illustrated in Figure 3 shows that before storing data in the network, users search for the hosts in the neighborhood by CDCS, and divides the data into A, B and C segments. The target of the search is the hosts with enough hard disk space, like Host1 to Host6 in Figure 3. According to the parameters, the algorithm figures out the weight of the hosts as the basis of storage selection. The analysis of CDCS shows that the weight of Host1, Host3 and Host5 is similar and thus, the phenomenon of asynchronous data can be reduced. Therefore, data segments are individually stored in the selected hosts. By considering the linking situations of the hosts, CDCS maintains the high-speed convenience of LAN and keeps the system load-balancing.

The parameters used to calculate the weight is listed in the following (1), including the HD space, the bandwidth, the link speed, the storage area, the Host load and its reliability. We assign different priority weight to different parameters, and the total of the weight serves as the basis of selection. The related functions are given in (2) to (7). It reveals that our experimental architecture can simulate the optimal weight for different scenarios and apply it to the cloud network.

Weight = { HD space, Bandwidth, Link speed, Area, Host load, Reliability }*100% (1)

HD space = *Consult Host for the information* (2)

Bandwidth = *Host's Total Data Traffic / Time (Kbps)* (3)

Link speed = *Average Data Traffic / Time between UE and Host (Kbps)* (4)

Area = *Check Hop Count through Routing Table* (5)

Host load = *Compare the Host's {CPU, Memory. HD}* (6)

Reliability = *Compare the Packet Loss Rate and Effective Link Rate (%)* (7)

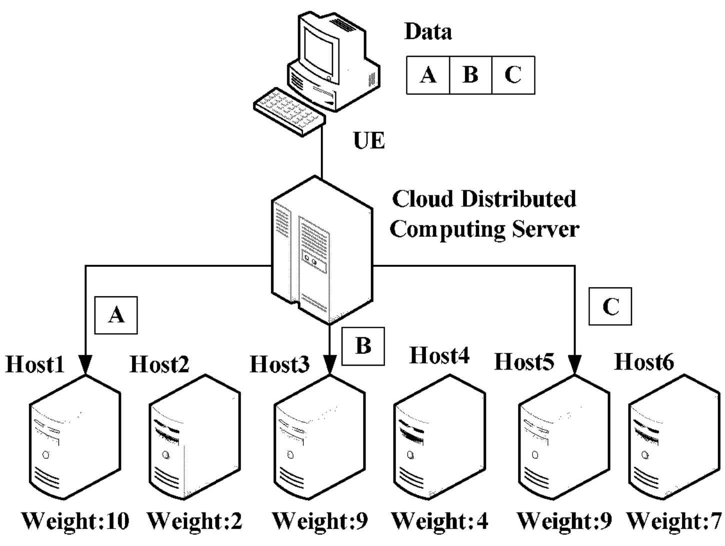

Figure 3. Selection mechanism of data storage

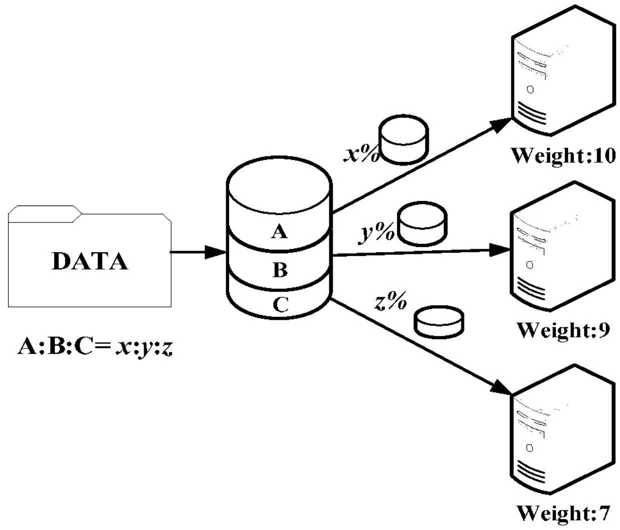

Figure 4. Synchronized data storage mechanism

B. Synchronized Data Storage Mechanism

Generally speaking, when the weight is used as the basis of selection, not every time can the system select the hosts with similar weight, which means that different speed of the hosts causes the data storage asynchronous that leads to the difficulty in sequencing all nodes, the uncertainty of the finish of data storage, and the situation that the nodes wait to be selected mutually. To solve the above-mentioned problems, we propose to add the synchronized data storage mechanism, as shown in Figure 4. By using CDCS to select the hosts with higher weight, the mechanism divides the data to different extent according to the weight of the hosts. The host with higher weight stores more percentage of data, which is determined based on the weight, like x%, y% and z% illustrated in Figure 4. In this way, appropriate percentage division helps to achieve synchronized data processing and enhance the efficiency of network access.

C. Auto Remote Backup Mechanism

In recent years, the great losses of lives, properties, and important data caused by several catastrophes are still impressive to the whole world and the issues of data backup and disaster recovery therefore grab the attention of the public. Thus, an auto remote backup mechanism is proposed in this paper. Besides the backup at the remote access, several characteristics of this mechanism further include offline backup, multiple error detection and intelligent recovery. After transmitting data to the selected host, the UE leaves the system because the local and remote backup is processed by the selected host, which features the characteristic of offline backup. CDCS ordinarily records the storage areas of the data segments and checks the legality of the data periodically. However, when any abnormal data is detected, CDCS first judges whether the loss is local or remote, and notifies the rest hosts that own this data to search for new hosts to backup. This mechanism thus depicts the features of multiple error detection and intelligent recovery.

As shown in Figure 5, Host1 is one of the hosts selected by CDCS through the selection mechanism of data storage. Because of the auto backup mechanism, the selected Host1 receives the completed data and automatically searches for another host with similar weight in the local area for auto backup. To prevent some situations from happening, like the total break down of the network of certain area, we set up a remote backup mechanism. After finding Host2 with similar weight in the local area, Host1 simultaneously seeks the remote Host3 for backup. In our definition, the remote host must be at least 30 kms away from the local host. By simulating the experimental model, we can acquire the optimal experience value.

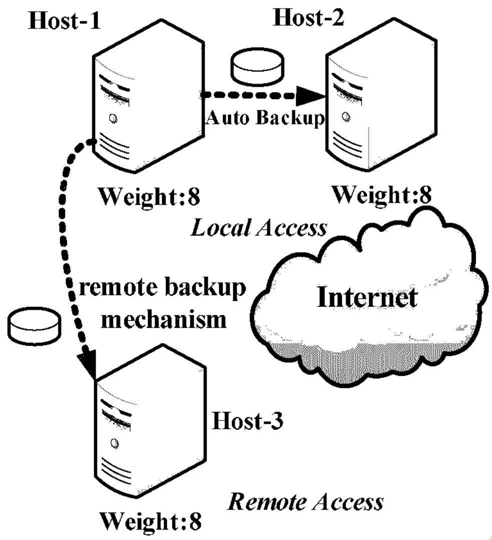

Figure 5. Auto remote backup mechanism

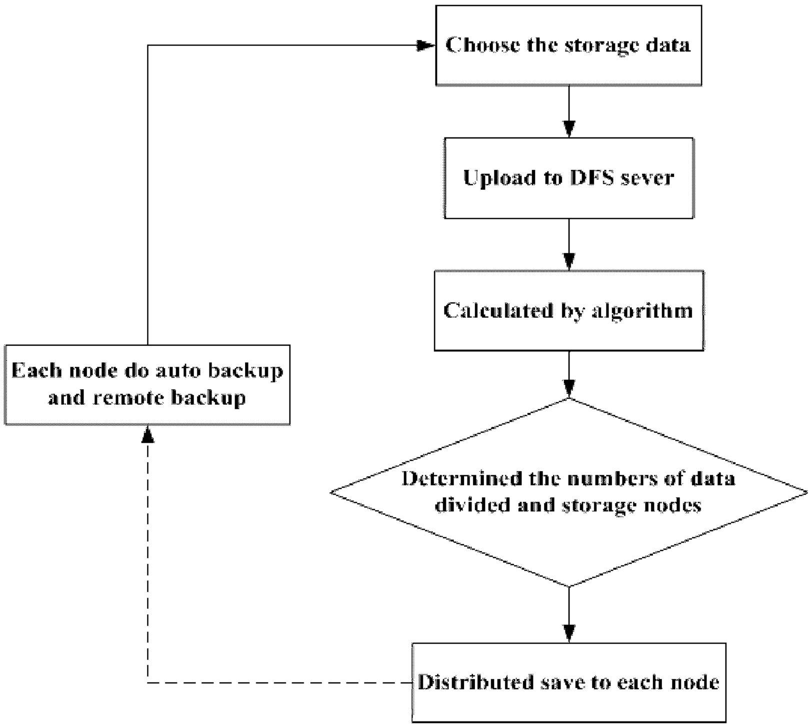

Figure 6. Flowchart of data storage algorithm

Based on the weight calculated by CDCS in our algorithm, CDCS divides the data proportionally and selects the hosts with similar but higher weight as the storage space. Because of the mechanisms of auto backup and remote backup, the selected hosts further search for other hosts with similar weight for backup to increase the reliability of data and reduce the load of CDCS. In such a manner, there will be more choices of the source hosts and the efficiency of data transmission can be increased. The detailed flowchart is given in Figure 6.

D. Distributed Multi-node Simultaneous Reading

Being faster than single reading, multi-node simultaneous reading will improve the reading performance. As shown in Figure 7, to backup through the selected host can increase the choices of hosts while reading data. All hosts in Figure 7 are the access points that can download data simultaneously to enhance the access speed.

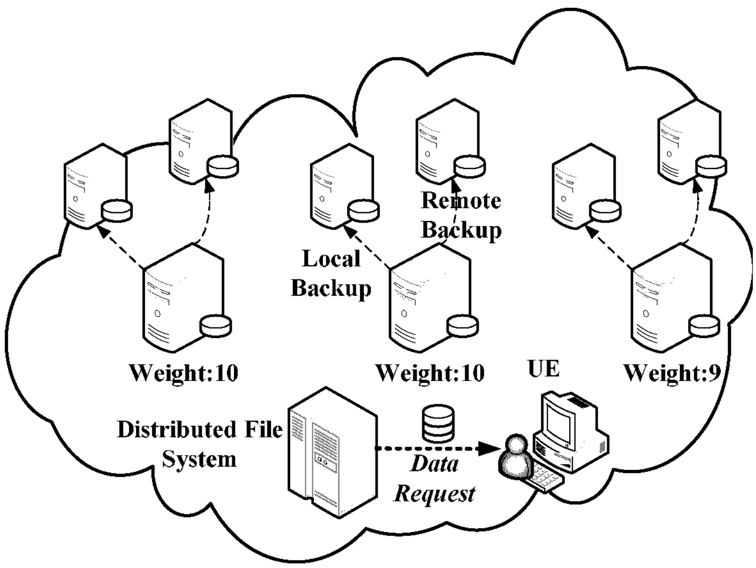

Figure 7 Diagram of distributed multi-node simultaneous reading

Table 1. Simulation Parameters

Parameter	Value
Scenario	Wired topology
Protocol	TCP
Application	FTP
Number of nodes	25
Data Size	384/ 768/ 1152 /1536/ 1920/ 2304 (Mbytes)
Bandwidth	2~8 (Mbps)
Transmission Delay	10~20 (ms)
Routing Protocol	Bellman-Ford

Ⅳ. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

NS-2 is adopted to simulate the performance of data upload/download. We establish the local and remote nodes and set the parameters of all nodes randomly based on the variable scope. Related simulation parameters are listed in the following Table 1. With regard to the data upload, we compare HDFS and the proposed CDCS in different data sizes and calculate the needed time to transmit data as shown in Figure 8. As for HDFS, the default block size is 128 Mbytes and the mechanism searches for the local and remote nodes for transmission and data backup. However, our proposed CDCS takes the bandwidth and handling capacity of every node into consideration, and adjusts the upload data size of every node proportionally. The total upload time of Th-CDCS is therefore better than HDFS. To accelerate the transmission speed of large files, we modify the original method by adding the threshold value and the agent mechanism. Through the local network, the node that backup at the remote access searches for new agents to accelerate users' total upload time, and the remote backup will be processed by the agent. The threshold value in our simulation is set to 1000Mbytes. With the increase of the threshold value for one unit, a local node with better performance is added to support the computing. Thus, Figure 8 shows that Th-CDCS performs better in transmitting large files and users' waiting time can be reduced. For example, to transmit a file of 2304Mbytes, when HDFS spends 1536 seconds and CDCS 1152 second, Th-CDCS spends only 682 seconds, approximately 45% of HDFS

.

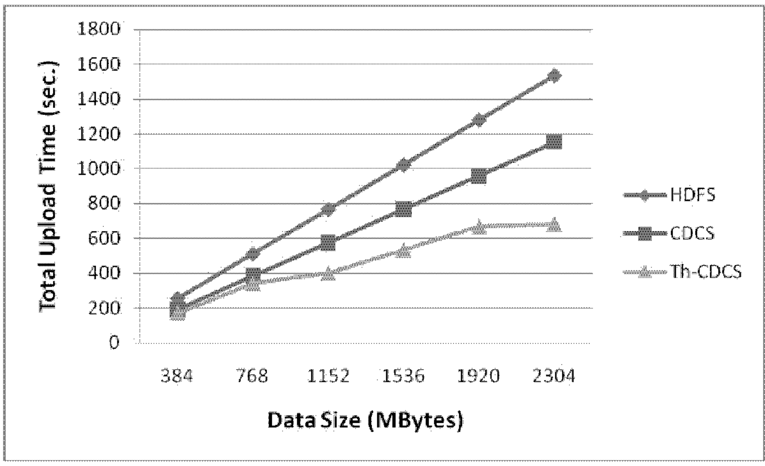

Figure 8. Total upload time to data size

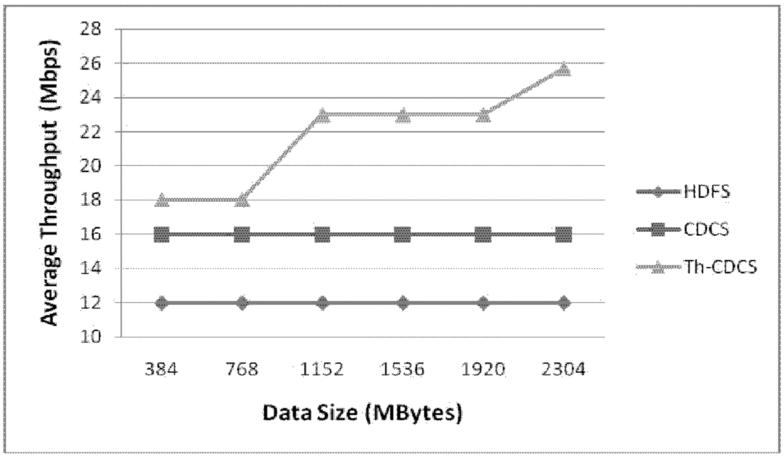

Figure 9. Average throughput to data size

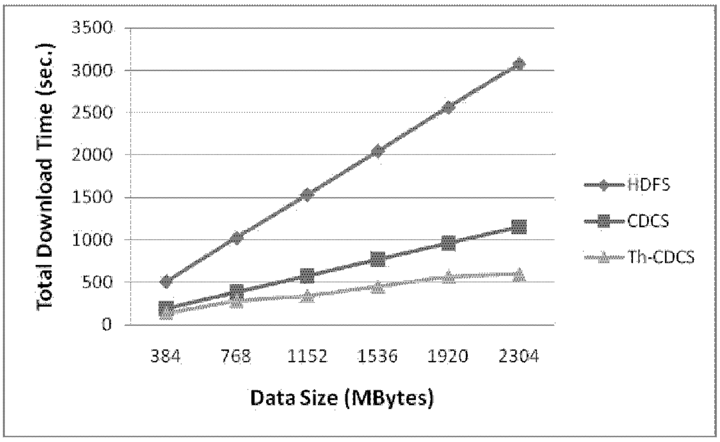

Figure 10. Total Download Time to Data Size

Figure 9 shows the graph of the average throughput to data size. CDCS performs better in the average throughput because it distributes the data storage according to the perofrmance of the nodes. Nevertheless, because of the agent mechanism, Th-CDCS performs even better.

As for the data download, the clients of HDFS usually access to the nearest backup while CDCS adopts the multi-node simultaneous reading mechanism. Therefore, when the data size is the same, the download time of CDCS is better than HDFS. But, owing to the agent nodes, Th-CDCS provides more datanodes for users to read simultaneously and the performance is even better.

V. CONCLUSION

This paper proposes a distributed cloud computing system that considers various network parameters to determine the host of data storage by CDCS. By taking synchronized data storage into account, this system uses CDCS to select the hosts with higher weight to store more percentage of data, which is determined by the proportion of weight. Thus, synchronized processing can be achieved to enhance the efficiency of network access. The characteristics of this mechanism include offline backup, multiple error detection and intelligent recovery. Also, we choose the remote hosts that are at least 30kms away from the local host with similar weight for auto backup. To enhance the reading performance, we use distributed multi-node simultaneous reading method to download data at one time to improve the access speed.

ACKNOWLEDGEMANT

This study was supported by the National Science Council, Taiwan, under grant no. NSC 99-2221-E-032-040.

REFERENCE

[1] N. Golpayegani and M. Halem, "*Cloud Computing for Satellite Data Processing on High End Compute Clusters*" CLOUD '09 Proceedings of the 2009 IEEE International Conference on Cloud Computing, pp.88-92, September 21-25, 2009

[2] Xuhui Liu, Jizhong Han, Yunqin Zhong, Chengde Han and Xubin He, "*Implementing WebGIS on Hadoop: A Case Study of Improving Small File I/O Performance on HDFS*" CLUSTER '09. IEEE

International Conference on Cluster Computing and Workshops, pp.1-8, August 31-September 4, 2009
[3] Tauhida Parveen, Scott Tilley, Nigel Daley and Pedro Morales, "*Towards a distributed execution framework for JUnit test cases*" 2009 IEEE International Conference on Software Maintenance, pp. 425-428, September 20-26, 2009
[4] G. Mackey, S. Sehrish and Jun Wang, "*Improving metadata management for small files in HDFS*" CLUSTER '09. IEEE International Conference on Cluster Computing and Workshops, 2009, pp. 1-4, August 31 2009-September 4, 2009
[5] Gui Bingxiang and Feng Hongcai, "*An New Data Parallelism Approach with High Performace Clouds*" 2009 Pacific-Asia Conference on Circuits, Communications and Systems, pp. 21-30, May 16-17, 2009
[6] J. kanayake, S. Pallickara and G. Fox, "*MapReduce for Data Intensive Scientific Analyses*" eScience '08. IEEE Fourth International Conference on eScience 2008, pp. 277-284, December 7-12, 2008
[7] Stephen H. Hall, Garrett W. Hall, James and A. McCall, "*High-Speed Digital System Design: A Handbook of Interconnect Theory and Design Practices*" A Wiley-Interscience Publication, JOHN WILEY & SONS, INC., 2000
[8] G. N. Srinivasa Prasanna, Amrita Lakshmi, Sumanth. S, Vijaya Simha, Jyotsna Bapat and George Koomullil, "*Data Communication over the Smart Grid*" IEEE International Symposium on Power Line Communications and Its Applications (ISPLC 2009), pp. 273-279, March 29-April 1, 2009
[9] Tin-Yu Wu, Kai-Hua Kuo, Hua-Pu Cheng, Jen-Wen Ding and Wei-Tsong Lee, "*Increasing the Lifetime of Ad Hoc Networks Using Hierarchical Cluster-based Power Management*" KSII Transactions on Internet and Information Systems, Vol. 5, Issue 1, pp. 5-23, January 2011.
[10] I-Ju Liao, Tin-Yu Wu, Jen-Wen Ding, and Wei-Tsong Lee, "A Novel 3D Streaming Protocol Supported Multi Mode Display" Journal of Networking and Computer Applications (JNCA), Vol. 34, Issue 5, pp. 1509-1517, September 2011.

BIOGRAPHIES

Tin-Yu Wu currently works as an Assistant Professor in the Department of Electrical Engineering, Tamkang University, Taipei, Taiwan. He received his M.S., and Ph.D. degrees in the Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan in 2000 and 2007 respectively. His research interests focus on the next-generation Internet protocol, mobile computing and wireless networks.

Tsung-Han Lin currently is a Master Student in the Department of Electrical Engineering, Tamkang University, Taipei, Taiwan.. His research interests focus on the cloud computing and Internet of Thing.