Enhancing the Quality of Data Grid Service with Node Selection Mechanism

S.C. Wang[1], H.J. Lyu[2], K.Q. Yan[3*], M.L. Chiang[4], S.S. Wang[5]
[1]scwang@cyut.edu.tw
[2]s9730614@cyut.edu.tw
[*3]kqyan@cyut.edu.tw
[4]mlchiang@cyut.edu.tw
[5]sswang@cyut.edu.tw
Chaoyang University of Technology
[*]Corresponding author

***Abstract*—Grid is an environment which combines all computing and resources. Grid can convert it into a readily available, standard, cost-effective computing power by utilizing available computing resources fully. Nowadays, there are many agencies and enterprises that need data devices with accessing large amounts of data, but these devices need to spend a large cost. In the data grid environment, there are many available resources of storage nodes including heterogeneous or homogeneous. It can reduce the time cost for managing data files and access data efficiently by using these resources. In this study, a mechanism of storage node selection is proposed to reduce the cost of storage node access.**

***Keywords*—Data grid, Distributed Data processing, Service Oriented, Clustering**

1. INTRODUCTION

Distributed system is usually utilized as a solution to overcome the lack of resource. As technology advances, resources of distributed system were not only homogeneous, and they are heterogeneous [3] In addition, network bandwidth and quality drastically improve in a speed even faster than the enhancement of computer performance, systems that execute distributed applications over network, such as grid computing [7], Peer-to-Peer (P2P) [11] etc. Grid computing utilizes the network to integrate resources available on idle resources scattered in every region to facilitate distributed application. P2P connects nodes on a network to share resources, communicate, and exchanges messages, and each node plays the role of both a client and a server at same time [11]. Newly distributed system is compared with tradition distributed systems, it provides better large-scale resource sharing, improves resource applications and the broad-field Internet access environment [15]. The limit of space on the conventional distributed system can be thus broken to gain the cross-platform feature and fully exploit the large resource of idle computers [14,19].

In recent years, network bandwidth and quality has drastically improved over even computer performance. The various communication and computing tasks in fields such as telecommunications, multimedia, information technology, and construction simulations, have been integrated and applied in a distributed computing environments. There are several options for establishing distributed computing such as cluster computing and grid computing [9].

Cluster computing combines several personal computers or workstations to conduct distributed applications through local network in the fixed area. The main disadvantage is that cluster systems are limited to a fixed area, making the application inflexible in variation.

Grid computing utilizes the network and combines idle resources from different geographic locations for distributed applications. Grid computing is distinguished from conventional distributed computing or cluster computing by its focus on large-scale resource sharing, innovative applications, in a widely-connected network, typically the Internet. The limit of space on conventional distributed systems can thus be broken to gain cross-platform features and fully exploit the large resources in idle computers [6,16].

Newly distributed system aggregates the rich storage resources from every part of the world to form a larger storage capability and assist in the application of large amounts of storage requests in scientific arena. However, resources are distributed in different geographical locations. Stability and performance of each resource is different. In other words, newly distributed system is dynamic and resources are composed of heterogeneous resources. Thus, an important problem is resources selection. Therefore, this study a mechanism of resource (storage node) selection is proposed, which can select an effective storage node to lower the odds of selecting ineffective storage nodes.

The second section of the study focuses on literature review, and the discussion of the proposed Hierarchical Clustering Mechanism (HCM) is detailed in the third section. The fourth section covers the design of Storage Node Decided Mechanism (SNDM). The fifth section centers on an example. Finally, the conclusion and the future work are presented in the sixth section.

2. RELATED WORK

Application of grid computing can be roughly divided into two types: computation grid and data grid [2]. Computational grid mainly solves problems with demand for CPU computation over long period of time. Many servers, workstations, supercomputers, and personal computers distributed in various locations are effectively integrated to construct a virtual supercomputer. Usually, software or current settings of machines intended to join in the grid computing do not have to be adjusted. They only have to be installed with software packages, such as Globus Toolkit [4,18], so as to be equipped with the ability to participate in and share resource of the grid system [5]. In principle, data grid is similar to computation grid, yet the study and application of the former mainly focus on the storage of and access to massive amount of data for scientific computation.

In grid computing environment, however, resources are contributed by the computers located anywhere in the network. In this environment, the designing of a suitable storage node selection algorithm that dynamically adjust in accordance with the variation of resources requires the consideration of numerous and complicated factors. The storage node selection algorithm is more complex, the more loading on the system would be; thus, system performance can be seriously affected. Therefore, a proper storage node selection algorithm becomes extremely important in the grid environment. This section will introduce relevant studies on grid computing, follows by detailed description.

The purpose of grid computing is to divide large-scaled data computing, storage, or analysis tasks into smaller sections to be assigned to idle computers available in the network [13]. It creates performance comparable to supercomputer with existing hardware and enables mutual data sharing [20]. Framework of grid computing is to integrate distributed resources in the network. Computers that provide resources only have to release idle computing and storing capabilities. That is, the resources in a grid environment are composed of idle resources provided by volunteers on the network [1]. Thus, new resources can be accessed any time, even when some original nodes fail to provide resources [6].

Since grid computing facilitates the development of many scientific researches, many projects associated with grid computing are having been proposed. Many grid computing projects are well known, such as SETI@HOME, Folding@HOME and LCG (LHC Computing Grid).

SETI@HOME is hosted by the University of Berkeley in the California was started in May 1995 [8,21,22]. The main purpose is to analyse incoming data from the Arecibo radio telescope in Puerto Rico [21]. Data recorded on high density tapes at the Arecibo Observatory amounted up to about 35 Gigabyte tapes per day, which were then mailed to University of Berkeley due to lack of sufficient bandwidth [8]. Those data were then divided into smaller sections of approximately 0.35 MB and sent from the SETI@home server over the Internet to people around the world [22].

Folding@HOME [17] is a grid project involved with study on proteins. Proteins are biology’s workhouses; it is a nano-machine inside human body [17]. Before acting out their biological functions, proteins assemble themselves, a process called “fold”. Although protein folding is critical and fundamental for virtually every creature, this folding process remains to be a mystery for people. Moreover, proteins' failure to fold correctly would lead to many well known diseases, such as Alzheimer's disease, Mad Cow (BSE), CJD, ALS, Huntington's, Parkinson's disease, and many Cancers and cancer-related syndromes [17]. Thus, the project for the simulation of protein folding, misfolding, convergence, and incidence of related diseases was proposed. The purpose was to enhance understanding about the etiology of Alzheimer's disease, Mad Cow (BSE), CJD, and Parkinson's disease.

LCG (LHC Computing Grid) is mainly promoted by European Organization for Nuclear Research (CERN) [20]. Its purpose is to integrate computation resources of worldwide major physics research centers and satisfy the need for massive computation, programming, data management, and system maintain for the experiments on LHC (Large Hadron Collider) [20]. Every year, LHC produces data with a size of about 10 petabytes (10 million Gigabytes). That is more than 1000x the amount of information in printed form every year around the world, and nearly 10% of all information that humans produce on the planet each year, including digital images, photos and what have you. In short, that is a lot of information [20]. Those data can help scientists find smaller particles. In order to utilize those data, scientists have to use large computer to analyse and filter useful information [20].

Grid computing can provide abundant resources to assist many studies and it can be employed in more and more projects that have so far yielded numerous findings.

However, in the data grid environment, many distributed storage resources (storage nodes) are integrated in different sites. If a user asks an I/O request to store the related data, then the data grid environment needs to search the appropriate storage node from a huge storage nodes set to select a suitable storage node to store the related data. The cost of search time is the bottleneck of data grid. In order to reduce the system load and avoid the lost of I/O request, a storage node selection mechanism according to the characteristics of storage node is proposed. When a user asks an I/O request, the storage node selection mechanism will accord the characteristics of request to search a suitable storage node.

In other words, the storage nodes can be arranged effectively by our proposed method to maintain the consistency of storage nodes and to access data effective.

3. HIERARCHICAL CLUSTERING MECHANISM

In the data grid, there are a lot of storage nodes with different CPU capacity, storage capability, and communication bandwidth. These storage node resources can be divided into homogeneous and heterogeneous resources, and in the mixed state lead to the manager cannot easier to control and manage. In order to help managers can manage and monitor efficiently of the resources, and the resources can be enable to use for different areas of data storage requirements, the mechanism of storage node grouping is necessary to be explored.

In a data grid, the storage resources of different geographically are integrated. If users propose the I/O request of the storage, they need to search an appropriate storage node from the large collection of storage nodes. The cost of search is very large, and the performance of search is also poor. The summarization of shortcoming as following:

(1) The appropriate storage node cannot to find to store file.
(2) The resource of storage node will be wasted.
(3) The store processing time is too long.
(4) The security of file saving cannot to be provided.
(5) The stability of the storage node cannot guarantee.

However, if these storage nodes are clustered at the beginning, then the above disadvantages can be avoided. When users ask the requirements of data storage or data acquisition, the characteristics of the requirements are analysed first. According to the characteristics of the requirements, the suitable group of storage nodes can be matched to fit the requirements, and the search sample of storage nodes can be lessened. The advantages of a suitable clustering mechanism are shown in fallowing:

(1) The cost of search time can be saved.
(2) These storage resources can easy to manage.
(3) Depend on the property of data grid application to provide a more flexible ways to select the storage nodes.
(4) Screen out storage nodes of inappropriate, to increase the utilization of storage resources.

In this study, a clustering mechanism is based on the SOS (Service Oriented Storage) architecture [10] to drive the optimization of the data access. When the users submit the I/O requests, the I/O requests will be stored in the *I/O request Queue* and get into the SOS by the submitted sequence to determine the suitable storage node. The concept of the proposed mechanism is to cluster the storage nodes first that can improve the efficiency of the storage node elected and the storage node managed.

In this study, a **Hierarchical Clustering Mechanism** (**HCM**) is proposed to cluster the storage nodes in the data grid and increase the efficiency of data grid application. At first, the storage node agent is used to gather the information (memory, bandwidth, CPU speed etc.) of all storage nodes in data grid environment and the information is stored in the node database. Then all storage nodes information will be provided for HCM to cluster the storage nodes by their characteristics.

In the data grid, various storage nodes in homogeneous or heterogeneous environment are integrated and provided different capacity. However, the mainly researches and applications in the data grid are the large number of scientific computing in data storage and access. Therefore, the evaluation factors of storage node that will affect the storage validity need to be considered when the I/O request submits the data access requirement.

However, the aim of clustering mechanism is used to reduce the search amount of storage nodes. Nevertheless, if the number of storage nodes in a group is large, then the search cost cannot to be reduced efficiency. Otherwise, if the number of storage nodes in a group is too small, then the group maybe cannot to provide the service of user's I/O request. Therefore, the HCM contains two mechanisms, the **Clustering Mechanism** and **Group Adjustment Mechanism**, are introduced as follows.

3.1 Clustering Mechanism

The **Clustering Mechanism** of HCM is based on the characteristics of user's I/O requests to cluster the storage nodes. The progressions of the **Clustering Mechanism** of HCM are shown in Figure 1, and the implementation steps of HCM are described as following.

Step 1. According to the historical information of the specific application to analysis the requirements of I/O request.
Step 2. Determine the order of evaluation factors for clustering.
Step 3. Store the result of analysis, the condition of clustering, and the order of evaluated factors for clustering to the node.
Step 4. To cluster the storage nodes by the ordering of evaluated factors
Step 5. Repeat Step 4, until all storage nodes are clustered into groups.
Step 6. The clustering results are stored into node database.

Clustering Mechanism of HCM
Analysis the historical information of I/O request Begin 1. According to the historical information of the specific application to analysis the requirements of I/O request. 2. Determine the order of evaluation factors for clustering. 3. Store the result of analysis, the condition of clustering, and the order of evaluated factors for clustering to the node database. End
Clustering the storage nodes Begin 1. i= 1 /* The first level clustering */ 2. C={n_1,n_2,n_3,…n_k} /* C is a cluster and C is the set of all storage nodes in the initial state */ 3. ***Clustering***(The i-the priority evaluation factor, C) 4. do (to pick a cluster C_k from C) 5. for i = 2, n /* n is the number of priority evaluation factors */ 6. ***Clustering***(The i-the priority evaluation factor, C_k) 7. until (C = Φ) End
Procedure *Clustering*(The priority evaluation factor, Cluster)
1. According to the priority evaluation factor to cluster the storage nodes 2. Until clustering is finish, return the set of clusters C=(C_1,C_2, C_3, …C_m). /* C is a cluster and C is the set of clusters after doing *Clustering* procedure */

Figure 1 The Clustering Mechanism of HCM

3.2 Group Adjustment Mechanism

The objective of **Clustering Mechanism** of HCM is to reduce the search number of the storage nodes. However, if the number of storage nodes in a group is large, then the search cost cannot to be reduced efficiency. Otherwise, if the number of storage nodes in a group is too small, then the group maybe cannot to provide the service of user's I/O request.

Therefore, the **Group Adjustment Mechanism** is also provided by HCM. If the number of storage nodes in a group is larger than the maximal number of allowable storage nodes (MaxSN), means the clustering result is not very well, and then the group needs to be divided into two groups. Then, the attribute value of evaluation factor is changed and the group information in the node database is updated.

Otherwise, if the number of storage nodes in a group is smaller than the minimal number of allowable storage nodes (Mins), this group is merged with other group that the attribute value of evaluation factor is simular. If the number of storage nodes in a merged group is larger than the MaxSN, then the merged group is divided into two groups, until the number of nodes in all groups is between the MinSN and MaxSN, then the **Group Adjustment Mechanism** is finished. The progression steps of **Group Adjustment Mechanism** are shown as following:

Step1: If the number of storage nodes in a group is greater than MaxSN, then the group is divided into two groups.

Step2: If the number of storage nodes in a group is lesser than MinSN, then the group is merged with other groups that the attribute value is simular.

Through the **Group Adjustment Mechanism**, the cost of node search can be reduced, and the lost of node search can be avoided.

The **Clustering Mechanism** and the **Group Adjustment Mechanism** of HCM are shown in Figure 2.

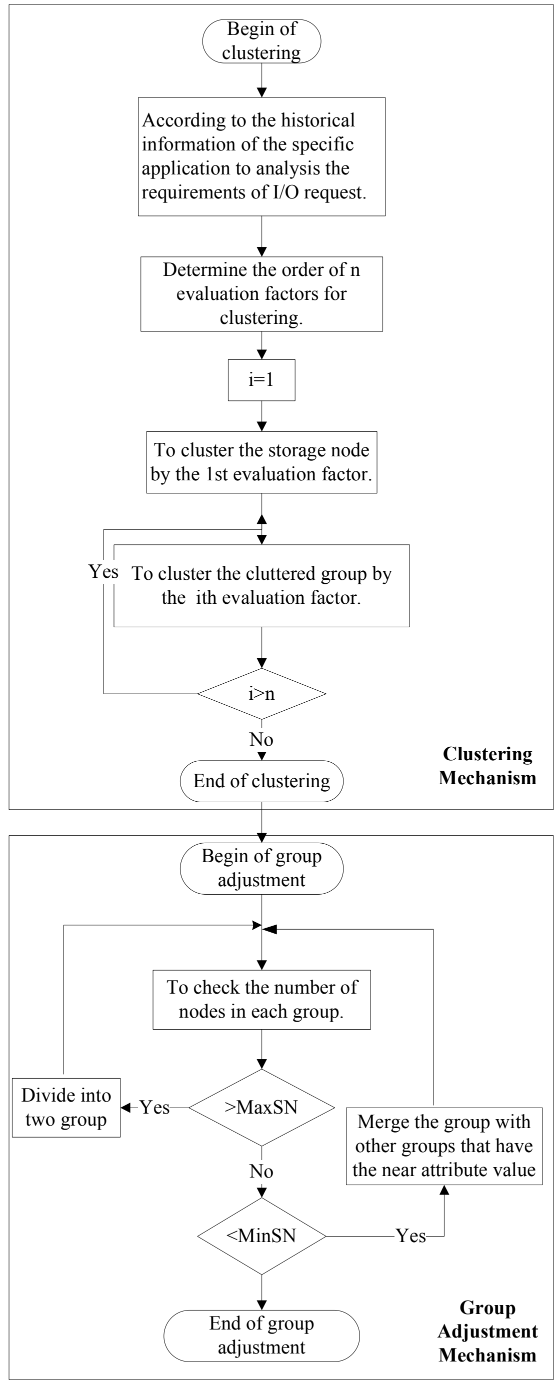

Figure 2. The Flow of HCM

4. STORAGE NODE DECIDED MECHANISM

In the data grid environment, there are many homogeneous and heterogeneous storage nodes can provide the node space, the processing power, and other resources so on [13]. In order to drive the optimized storage node access, the HCM is based on the SOS (Service Oriented Storage) framework [10]. According to the storage node characteristics, the storage nodes can be clustered to the service node group by HCM. However, the service node group is selected by using the screening factor in the SOS system, then, the storage node is chosen form the service node group to provide the service.

In this study, the appropriate service node group selected effectively is considered when a user submits an I/O request to the SOS. A good mechanism for decide the service node group and the storage node can reduce the cost of data store and enlarge the utilization of storage node space. Thus, it can effectively maintain the load balancing and enhance the performance of the system. The proposed method has the main advantage of locating proper resources according to the properties of I/O request. It can reduce the decrease of system performance resulting from miss-selection of ineffective nodes, and maintain the load balancing of the system.

In this study, a **Storage Node Decided Mechanism** (**SNDM**) is proposed. The SNDM contains two mechanisms, the **Service Node Group Decided Mechanism** and the **Storage Node Decided Mechanism.** When a user submits an I/O request to the SOS, the **Service Node Group Decided Mechanism** is used to select a suitable service group, and the **Storage Node Decided Mechanism** is used to select the appropriate storage node.

The aim of the **Service Node Group Decided Mechanism** is to filter out a set of appropriate storage groups and choose one from these groups. The aim of the **Storage Node Decided Mechanism** is to choose an appropriate storage node from this selected group.

Moreover, in the SOS, the storage node agent is responsible for the message of "data storage request" transmitting to the selected storage node, and the correspondence "ACK" message returning from the storage node. The progressions of the storage node selection are shown in Figure 3.

When the user's I/O request is submitted to the SNDM, the **Service Node Group Decided Mechanism** queries the information of storage nodes form the node database to determine the suitable service node group. Then, the **Storage Node Decided Mechanism** is used to select an appropriate storage node from the suitable service node group and the I/O request is serviced. When carries on the storage node choice, storage node agent synchronized records the related information of storage node into the node database.

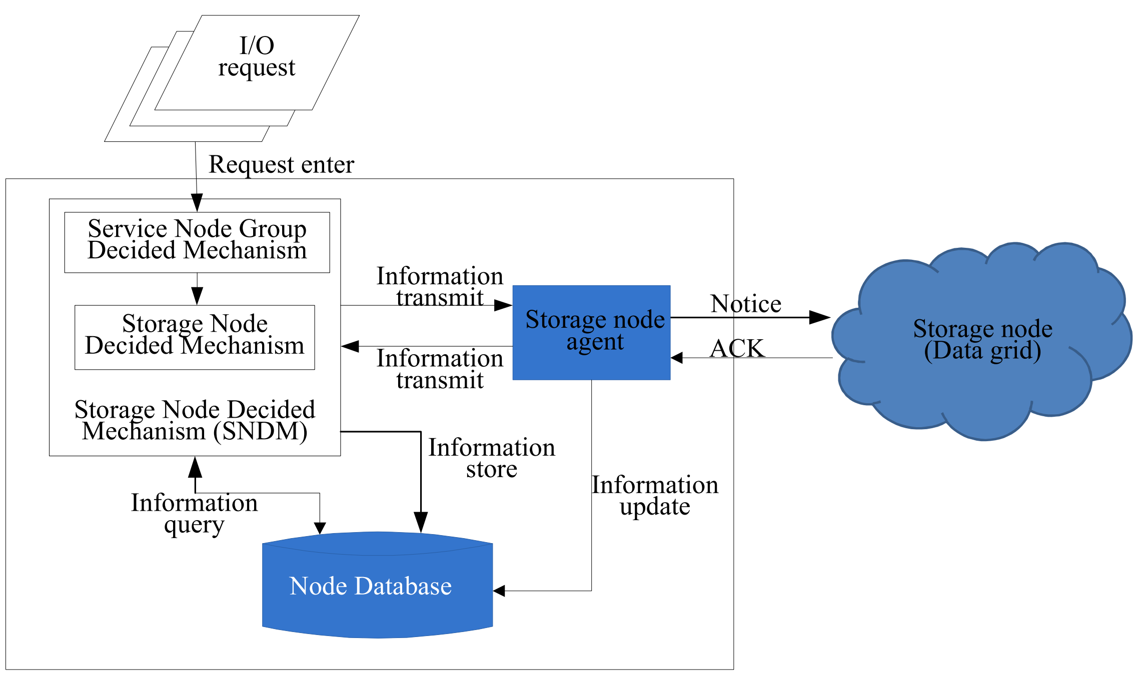

Figure 3. The progressions of the storage node selection

4.1. The Storage Node Agent

In this study proposes a policy to handle possible changes in storage node status by employing storage node agent to gather storage node information; from the information gathered by the storage node agent, the storage node agent can determine whether a storage node is ineffective. While the storage node is ineffective, a new storage node will be found for replacement, so as to maintain the efficiency of system.

The storage node agent collects the related information of all the storage nodes in the node database and compares the collected information with historic ones, in order to confirm (ACK) if the storage node is still effective. If the storage node remains effective, the distribution of the I/O request will not be re-adjusted. If the storage node is confirmed ineffective, a storage node will be re-selected from the selected service node group.

In this study, the storage node agent mainly aims to manage the storage nodes that provide storage resources in this data grid environment. When the storage node agent is launched, the storage node agent will collect related information of each storage node participating in this data grid, such as CPU utilization, remaining CPU capability, remaining memory, etc. After all these information are collected, they will be provided to the dispatcher and assist it in maintaining the efficiency of the system.

4.2 Service Node Group Decided Mechanism

When a user sends an I/O request for a data storage required, the **Service Node Group Decided Mechanism** is used to select a service node group. Each I/O request includes the related information suchlike the size of I/O request, the type of I/O request, and so on. In this study, the **Service Node Group Decided Mechanism** extracts the necessary information of I/O request, and then uses the First Fit scheduling policy [12] to screen the candidate of service node groups.

In the **Service Node Group Decided Mechanism**, the screening factors play an important role. According to the type of requested file, the screening factor is chosen. For example, for a multimedia file request, the storage space and the bandwidth are used as the screening factors. However, the information of storage space and bandwidth required will be extracted out by the **Service Node Group Decided Mechanism** to select the candidate of service node groups.

In the **Service Node Group Decided Mechanism**, two kinds of cases need to be considered.

1. When an I/O request is submitted to the SOS, then the **Service Node Group Decided Mechanism** needs to be run to choose the storage node group,
2. When the operation of data store is failure to operate, then the service node group needs to be re-selected by **Service Node Group Decided Mechanism**.

The **Service Node Group Decided Mechanism** is divided into six steps as following, and the progressions of **Service Node Group Decided Mechanism** is shown in Figure 4.

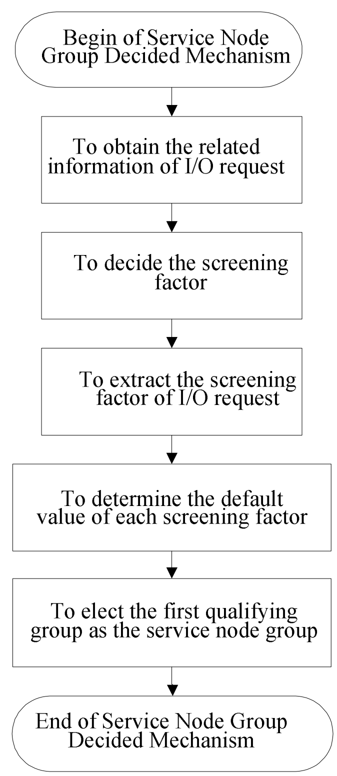

Figure 4. The progressions of Service Node Group Decided Mechanism

Step1. To obtain the related information of I/O request.

Step2. According to the HCM grouping factor to decide the screening factor.

Step3. To extract the screening factor of I/O request.
Step4. To determine the default value of each screening factor.
Step5. To elect the first qualifying group as the service node group.
Step6. End execution of the **Service Node Group Decided Mechanism** and enter the **Storage Node Decided Mechanism to choose** an appropriate storage node from this selected group.

4.3 Storage Node Decided Mechanism

After the progressions of **Service Node Group Decided Mechanism**, a service node group is elected. The capacity of storage nodes within this service node group are quite. However, the storage node can be selected by using the evaluation factors to provide services.

The **Storage Node Decided Mechanism** is divided into two steps as following, and the progressions of **Storage Node Decided Mechanism** is shown in Figure 5.

Step1. By using the evaluation factors to select the storage node.
Step2. The selected storage node information with the I/O request information is send to the storage node agent.

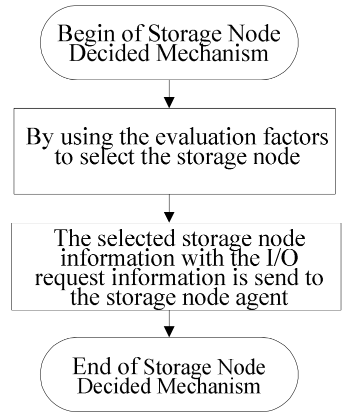

Figure 5. The progressions of Storage Node Decided Mechanism

5. EXAMPLE

In this section, an example is given to discuss the proposed HCM and SNDM. The parameters of example are listed in Table 1. However, the information of all storage nodes in the data grid is gathered in node database by storage node agent. The information of all storage nodes in node database includes storage space, bandwidth, CPU capacity, etc.

TABLE 1. THE PARAMETERS OF EXAMPLE

Parameter	Definition
Environment	Data grid
The number of storage nodes	40
Application	The data store of digital library
File Type	Web file, Multimedia, Data backup, Picture, and others

The historical information of the I/O requests for a specific digital library is shown in Table 2. According to the historical information shown in Table 2, the smallest storage capacity required of the specific digital library is 50MB. The largest storage capacity required of the specific digital library is 15G. Therefore, the evaluation factor of storage node’s space cannot less than 50MB. Due to the storage requirement of the I/O requests for a specific digital library shown in Table 3, the storage space required of each I/O request is different. Therefore, the frequency of each I/O request is analysed to get the quantity of storage space required, then HCM can use this information to cluster the storage nodes.

TABLE 2. THE HISTORICAL INFORMATION OF THE I/O REQUESTS FOR A SPECIFIC DIGITAL LIBRARY

File Type		Storage capacity	Amount
Data backup		10~15G	5
Multimedia	.avi	2~4G	10
	.rmvb	250MB~650MB	15
	.mp3 and .wma	5~10MB	20
	flash	3MB~5MB	25
	.mpeg and .wmv	50MB~1G	30
Web file		30G	35
Picture		2G	40

TABLE 3. THE STORAGE REQUIREMENT OF THE I/O REQUESTS FOR A SPECIFIC DIGITAL LIBRARY

Storage capacity	Amount	Ratio
3MB~10MB	45	25%
50MB~1G	45	25%
2 G ~15G	55	30.56%
30G	35	19.44%
Total	180	100%

The information of all storage nodes in this example is shown in the Table 4. The information includes storage capability, bandwidth and CPU capacity.

According to the historical information of the I/O requests for a specific digital library, the first priority of evaluation factor can be gotten that is the space of storage node, and the second priority of evaluation factor is the bandwidth. Due to the analysis result of I/O request, the range of the storage space can be set. Then, the range of the storage space is used to cluster the storage nodes first. The storage nodes of each group after first clustering are shown in Table 5.

Subsequently, base on the historical information shown in Table 2, the bandwidth of storage node can be divided into three segments, less then or equal 5Mbps, between 5Mbps and 10Mbps, and greater then 10Mbps. However, the bandwidth is used to cluster the group after first clustering, and the groups of second level clustering are shown in Table 6.

According to the historical information of different the I/O requests for a specific digital library, assumes the screening factors of groups are the space of storage node and bandwidth, and shown in Table 7. Depend on the Table 7; if the I/O request is multimedia file, then the storage node space and the bandwidth are used to cluster the storage nodes. If the I/O request is website file, data backup file or image file, then the storage node space is used to cluster the storage nodes. If we cannot judge the file type of the I/O request, then the storage node space and the bandwidth are used to cluster the storage nodes.

When the storage node space of a I/O request is used to select the storage nodes, then the selected space of storage node needs greater than the range of the storage space of a group that is one of the evaluation factors. In addition, the first eligible group will provide the service. In terms of storage node space and bandwidth as the screening factors, then the range of the storage space of a group is filtered out first, then the first eligible group that the bandwidth is coincidence is selected.

TABLE 4. THE DATA OF STORAGE NODES

Storage Node ID	Storage capability (GB)	Bandwidth (Mbps)	CPU capability (MHz)
1	10	1	200
2	30	8	450
3	6	2	266
4	11	10	300
5	17	12	266
6	20	8	400
7	18	1	200
8	30	2	300
9	42	10	400
10	40	12	266
11	16	10	400
12	36	12	450
13	46	8	200
14	27	1	300
15	43	2	500
16	9	10	266
17	17	12	500
18	21	1	200
19	26	8	300
20	46	2	500
21	32	10	266
22	17	12	400
23	19	10	300
24	13	12	200
25	23	1	300
26	15	8	400
27	9	2	200
28	21	10	450
29	3	12	400
30	35	1	266
31	21	8	450
32	19	2	200
33	7	1	400
34	25	2	266
35	5	10	450
36	7	12	300
37	60	10	200
38	89	12	450
39	4	8	266
40	73	1	450

TABLE 5. THE STORAGE NODES OF EACH GROUP AFTER 1ST CLUSTERING

Storage capability	Storage node	1st level group
50M~10G	1,3,16,27,29,33,35,36,39	1
11G~20G	4,5,6,7,11,17,22,23,24,26,32	2
21G~30G	2,8,14,18,19,25,28,31,34	3
>30G	9,10,12,13,15,20,21,35,37,38,40	4

In this example, there are four users A, B, C and D. Assume user A, B, C and D sends an I/O request sequentially, and then four I/O request information can be gathered. The I/O request information of each user is shown in Table 8. The information includes the size of I/O request and the type of I/O request. The size of I/O request for user A is 15G and the type of I/O request is the multimedia. The size of I/O request for user B is 20G and the type of I/O request is the multimedia. The size of I/O request for user C is 15M and the type of I/O request is the image. The size of I/O request for user D is 400M and the type of I/O request is the data file.

TABLE 6. THE GROUPS OF 2ND LEVEL CLUSTERING

Size of storage	Bandwidth(Mbps)	Storage nodes	1st level group	2nd level group
50M~10G	< 5	1,3,27,33	1	1.1
	5 ~ 10	16,35,39		1.2
	>10	29,36		1.3
11G~20G	< 5	7,32	2	2.1
	5 ~ 10	4,6,11,23,26		2.2
	>10	5,17,22,24		2.3
21G~30G	< 5	8,14,18,25,34	3	3.1
	5 ~ 10	2,19,28,31		3.2
	>10			3.3
>30G	< 5	15,20,30,40	4	4.1
	5 ~ 10	9,13,21,37		4.2
	>10	10,12,38		4.3

TABLE 7. THE PRIORITY OF EVALUATION FACTOR

File Type	Evaluation Factor	
	Node space	Bandwidth
Web file	V	X
Data Backup	V	X
Picture	V	X
Multimedia	V	V
Others	V	V

TABLE 8. THE USER'S I/O REQUEST

User	Size of I/O request	Type of I/O request
User A	15G	Multimedia
User B	20G	Multimedia
User C	15M	Picture
User D	400M	Data

When user A submits an I/O request to the data grid, according to the Table 6, the groups 3.2, 4.1, 4.2, and 4.3 are selected as the candidate of service groups by the Service Node Group Decided Mechanism. But, by using the first fit scheduler, the group 3.2 is selected to provide the service of I/O request [12]. Then, the storage nodes 2, 19, 28, and 31 in group 3.2 are the candidate of service storage nodes. Finally, the storage node 2 is selected to provide the service by the first fit scheduler [12].

When the type of I/O request submitted by user B is multimedia. The screening factors are the storage node space and bandwidth. And the storage node space needs greater than 20G and the bandwidth is largest, hence the candidate of service groups is 3.2, 4.1, 4.2, and 4.3. According to the first fit [12], the group 3.2 is used to service. And the storage nodes 19, 28, and 31 of group 3.2 are the candidate of service storage nodes. Base on the first fit [12], the storage node 19 is selected to service.

The type of I/O request submitted by user C is image. Therefore, the screening factor is the storage node space only. The storage node space of group 1.1 is greater than 15M, and the storage node 1 is selected to provide service by first fit.

The type of I/O request submitted by user D submitted is data file. Hence, the screening factor is the storage node space only. The storage node space of group 2.1 is greater than 400M, and base on the first fit [12], the storage node 7 is selected to service.

6. CONCLUSIONS

In recent years, network bandwidth and quality has been drastically improved, even much faster than the enhancement of computer performance. The various communication and computing tasks in the fields such as telecommunication, multimedia, information technology, and construction simulation, can be integrated and applied in a distributed computing environment in nowadays. However, as the demands of many researches for computing resources gradually grow, grid computing integrated with a distributed computing environment and the Internet has gained more attention. Grid computing is to utilize the idle storage resources (storage nodes) on the network to facilitate the execution of complicated tasks that require a lot of resources. Namely, the composition of grid resources is dynamic and varies with time. Thus, when selecting storage nodes for storing an I/O request, the dynamic of the storage nodes in the grid must be considered, and to exploit the effectiveness of the resources, they have to be properly selected according to the properties of the I/O request. This study proposed an efficient storage node selection mechanism to maintain the storing performance and reduce the cost of storage node access.

In this study, the Hierarchical Clustering Mechanism (HCM) and the Storage Node Decided Mechanism (SNDM) are proposed for the data grid environment. Since the data grid environment is more complexity than tradition distributed system. Therefore, if the proposed mechanisms can reduce the cost of storage node access in the data grid environment, the proposed mechanism also can support to solve the cost of storage node access in other distributed system.

ACKNOWLEDGMENT

This work was supported in part by the Taiwan National Science Council, under Grants NSC95-2221–E-324-019-MY3, NSC96-2221-E-324-021 and NSC-97-2221-E-007-MY3.

REFERENCES

[1] J. H. Abawajy, “Fault-tolerant scheduling policy for grid computing systems,” in Proc. of the *18th International Parallel and Distributed Processing Symposium (IPDPS‘04)*, 1004, pp. 238.

[2] M. Baker, R. Buyya and D. Laforenza, “Grids and grid technologies for wide-area distributed computing,” *Software: Practice and Experience*, Vol. 32, No. 15, pp. 1437-1466, 2002.

[3] C.Y. Chang, “Using the concept of grid computing on improving the performance of service-oriented systems,” *Master Thesi*s, Department and Information Management System, Chung Yuan Christian University, 2003.

[4] I. Foster and C. Kesselman, “Globus: a metacomputing infrastructure toolkit,” T*he International Journal of Supercomputer Applications and High Performance Computing,* Vol. 11, No. 2, pp. 115-128, 1997.

[5] I. Foster, and C. Kesselman, *The grid: blueprint for a new computing infrastructure*, Morgan Kauffmann, 1st edition, 1999.

[6] I. Foster, C. Kesselman, and S. Tuecke, “The anatomy of the grid: enabling scalable virtual organizations,” *The International Journal of High Performance Computing Applications*, Vol. 15, No. 3, pp. 200-222, 2001.

[7] N. Fujimoto, and K. Hagihara, “A comparison among grid scheduling algorithms for independent coarse-grained tasks,” in Proc. of *Applications and the Internet Workshops*, pp. 629-635, 2004.

[8] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, and M. Leboisky, “SETI@home: massively distributed computing for SETI,” *Computing in Science & Engineering*, Vol. 3, No. 1, pp. 78-83, 2001.

[9] R. Oldfield, and D. Kotzl, “Armada: a parallel I/O framework for computational grids,” *Future Generation Computing Systems*, Vol. 18, No. 4, pp. 501-523, 2002.

[10] K. Ranganathan and I. Foster , “Identifying dynamic replication strategies for a high-performance data grid”, in Proc. of the *International Workshop on Grid Computing*, Denver, November 2001.

[11] M. Ripeanu, “Peer-to-Peer architecture case study: Gnutalla network,” in Proc. of the *Peer-to-Peer Computing*, pp. 99-100, 2001.

[12] A. Silberschatz, P. Galvin, G. Gagne, *Operating System Concepts*, John Wiley & Sons, Inc., Sixth Edition, 2001.

[13] J. Tourino, M.J. Martin, J. Tarrio, and M. Arenaz, “A grid portal for an undergraduate parallel programming course,” *IEEE Transactions on Education*, Vol. 48, No. 3, pp. 391-399, 2005.

[14] K.W. Tse, W. K. Lam, and P. K. Lun, “Reservation aware operating system for grid economy,” *ACM SIGOPS Operating Systems Review*, Vol. 37, No. 3, pp. 36-42, 2003.

[15] K. Yang, X. Guo, A. Galis, B. Yang, and D. Liu, “Towards efficient resource on-demand in grid computing,” *ACM SIGOPS Operating Systems Review*, Vol. 37, No. 2, pp. 37-43, 2003.

[16] CERN, http://public.web.cern.ch/Public/Welcome.html, 2009.

[17] Folding@Home, http://folding.stanford.edu/, 2009.

[18] Globus Toolkit, http://www.globus.org/toolkit/, 2009.

[19] GridCafe, http://gridcafe.web.cern.ch/gridcafe/, 2009.

[20] LCG - LHC Computing Grid Project, http://lcg.web.cern.ch/lcg/, 2009.

[21] SETI@home,http://setiathome.berkeley.edu/, 2009.

[22] SETI@home-From Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Seti %40home, 2009.