Future Generation Computer Systems 26 (2010) 1127–1140

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Implementation of a medical image file accessing system in co-allocation data grids[☆]

Chao-Tung Yang[a,*], Chiu-Hsiung Chen[b], Ming-Feng Yang[a]

[a] *High-Performance Computing Laboratory, Department of Computer Science, Tunghai University, Taichung, 40704, Taiwan*
[b] *Tungs' Taichung MetroHarbor Hospital, No. 699, Sec. 1, Jhongci Rd., Wuci Township, Taichung County 435, Taiwan*

ARTICLE INFO

Article history:
Received 9 November 2009
Received in revised form
26 May 2010
Accepted 28 May 2010
Available online 8 June 2010

Keywords:
Medical images
Data grid
Grid computing
Co-allocation
File transferring

ABSTRACT

There are two challenges of using the PACS (Picture Archiving and Communications System). First, PACS are limited to certain bandwidths and locations. Second, the high cost of maintaining Web PACS and the difficult management of Web PACS servers. Besides, the quality of transporting images and the bandwidth of accessing large files from different locations are difficult to guarantee. For instance, radiologists make use of PACS information system for achieving high-speed accessing medical images. Physicians, on the other hand, utilize web browsers to indirectly access the PACS information system via non-high-speed network. The insufficient bandwidth may cause bottleneck under a host of querying and accessing. As hospitals exchange large files such as medical images with each other via WANs, the bandwidth cannot support the huge amount of file transportation. In this paper, we propose a PACS based on data grids, and utilize MIFAS (Medical Image File Accessing System) to perform querying and retrieving medical images from the co-allocation data grid. MIFAS is also suitable for data grid environments with a server node and several client nodes. MIFAS can take advantage of the co-allocation modules to reduce the medical image transfer time. Also, we provide experiments to show the performance of MIFAS. Furthermore, in order to enhance the security, stability and reliability in the PACS, we also provide the user-friendly management interface.

1. Introduction

Nowadays, 2D, 3D, and 4D medical imaging devices are increasingly needed by hospitals. With the progress of medical photograph, the resolution of medical images is raising. Therefore, the scale of medical image files range from MB to GB. The size of high-resolution medical images, such as 64/128-slice CT scans, 3.0T MRI, and PET, often exceed one hundred MB or more. However, the speed of progress on many high-quality imaging devices and it related infrastructure are not match. Current Picture Archiving and Communication Systems (PACS) [1–4] are unable to provide efficient query response services. It is difficult to sustain huge queries and file retrievals under limited bandwidth. Therefore, the quality of communication in the Web PACS network would be restricted by bandwidth and conventional access strategies about exchanging and downloading a large amount of images. In order to enhance the quality of medical treatment, the medical imaging needs to associate with efficiency file transfer strategy to achieve high-speed accessing.

In this paper, we present a new strategy for processing medical image queries, which is based on the co-allocation [5–10] strategy for data grid environments. A data grid is a system composed of multiple servers that work together to manage information and related operations – such as computations – in a distributed environment. Our proposed system is called the Medical Image File Accessing System (MIFAS) for co-allocation data grids. To solve these problems, we propose the PACS based on the co-allocation data grid environment. MIFAS helps us to transfer huge medical images into the co-allocation data grid environment. We utilize the Globus Toolkit 4.0.7 [11–13] to establish the data grid environment for deploying co-allocation strategy and processing medical images. MIFAS helps users to quickly retrieve medical images from Medical Data Grid. The Cyber Agent Service and the Grid Service GUI desk application are implemented to assist in query and retrieve medical images. Also, MIFAS provides resume broken transfer to deal with the unstable circumstance of network. It not only enhances the overall quality of medical care system but also supports multiple replicas of medical images for failover recovery.

This paper presents a strategy to improve the security, stability and reliability of the PACS. Our strategies focus on integrating the service of processing medical images and stimulating PACS architecture into grid environments. The remainder of this paper is

---

[☆] This work is supported in part by the National Science Council, Taiwan R.O.C., under grant nos. NSC 97-2622-E-029-003-CC2 and NSC 98-2622-E-029-001-CC2.

[*] Corresponding author. Tel.: +886 4 23590415; fax: +886 4 23591567.
E-mail addresses: ctyang@thu.edu.tw (C.-T. Yang), t2884@ms.sltung.com.tw (C.-H. Chen), orsonyang@gmail.com (M.-F. Yang).

0167-739X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2010.05.013

organized as follows. Background review and studies are presented in Section 2. The Cyber Agent Transformer is introduced in Section 3. Experimental results are presented in Section 4. Section 5 concludes this article.

2. Background

2.1. Data grids

Data grids enable the sharing, selection, and connection of a wide variety of geographically distributed computational and storage resources for solving large-scale data-intensive scientific applications (e.g., high energy physics, bioinformatics applications, and astrophysical virtual observatory) [14–16]. The term "Data grid" traditionally represents the network system with distributed storage resources, from archival systems to caches and databases, which are linked using a logical name space to create global, persistent identifiers, and provide uniform access mechanisms [17,1,18–20,10].

Distributed scientific and engineering applications could access huge amounts of data between storage systems; these file often generated by many geographically distributed applications and users for analysis and visualization. Data grids consist of scattered computing and storage resources located in different countries/regions yet accessible to users. Data grid also provides file replication, which means datasets could be replicated within grid environments for reliability and performance. With replication, clients could discover existing data replicas and create or register new replicas.

Replica selection is important to data-intensive applications, it can provide location transparency. When a user requests a data set, the system determines an appropriate way to deliver the replica to the user. Another issue concerning replica selection is the prediction of the transfer time. Therefore, it involves the inspection of many characteristics and is a complex piece of work.

In situations where replicas are to be selected based on access time, Grid information services can provide information about network performance and perhaps the ability to reserve network bandwidth, while the metadata repository can provide information about the size of the file. Based on this, the selector can rank all of the existing replicas to determine which one will yield the fastest data access time. Alternatively, the selector can consult the same information sources to determine whether there is a storage system that would result in better performance if a replica was created on it.

2.2. Co-allocation model

The proposed architecture [6] consists of three main components: an information service, a broker/co-allocator, and local storage systems. Fig. 1 shows the co-allocation of data grid, an extension of the basic template for resource management [21] provided by the Globus Toolkit. Applications specify the characteristics of desired data, and pass attribute descriptions to a broker. The broker searches for available resources, gets replica locations from the Information Service [22] and Replica Management Service [7] to retrieve lists of physical file locations.

We implemented the following eight co-allocation [23] schemes including Brute-Force (Brute), History-based (History), Conservative Load Balancing (Conservative), Aggressive Load Balancing (Aggressive) [5], Dynamic Co-allocation with Duplicate Assignments (DCDA), Recursively Adjusting Mechanism (RAM) [23], Dynamic Adjustment Strategy (DAS) [24], and Anticipative Recursively Adjusting Mechanism (ARAM) [25]. In [5], the author proposes the co-allocation architecture for co-allocating grid data transfers across multiple connections by exploiting the partial copy feature of GridFTP. It also provides Brute-Force, History-Base, and Dynamic Load Balancing for allocating data block.

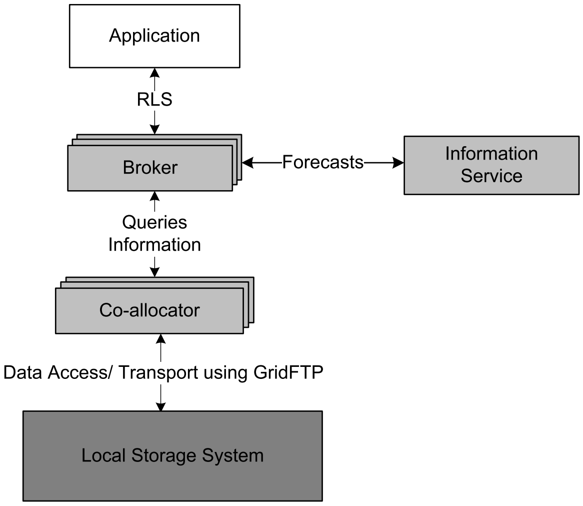


Fig. 1. Data grid co-allocation architecture.

- Brute-Force Co-Allocation: Brute-Force Co-Allocation works by dividing files equally among "n" available flows (locations). Thus, if the data to be fetched is size, "S" and there are "n" locations to fetch it from, then this technique assigns to each flow a data block of size, "S/n". For example, if there are three sources, the target file will be divided into three blocks equally. And each source provides one block for the client. With this technique, although all the available servers are utilized, bandwidth differences among the various client–server links are not exploited.
- History-based Co-Allocation: The History-based Co-Allocation scheme keeps block sizes per flow proportional to transfer rates predicted by the previous results of file transfer results. In history-based allocation scheme, the block size per flow is commensurate to its predicted transfer rate, decided based on a previous history of GridFTP transfers. Thus, the file-range distribution is based on the predicted merit of the flow. If these predictions are not accurate enough, renegotiations of flow sizes might be necessary as slower links can get assigned larger portions of data, which could be weight heavily on the eventual bandwidth achieved. With the history-based approach, client divides the file into "n" disjoint blocks, corresponding to "n" servers. Each server "i", $1 \leq i \leq n$, has a predicted transfer rate of "B_i" to the client. In theory then, the aggregate bandwidth "A" achievable by the client for the entire download is $A = \sum_{i=1}^{i=n} Bi$. For each server "i", $1 \leq i \leq n$, and for the data to be fetched is size of, "S", the block size per flow is $S_i = \frac{Bi}{A} \times S$.
- Conservative Load Balancing: One of their dynamic co-allocation is Conservative Load Balancing. The Conservative Load Balancing dynamic co-allocation strategy divides requested datasets into "k" disjoint blocks of equal size. Available servers are assigned single blocks to deliver in parallel. When a server finishes delivering a block, another is requested, and so on, till the entire file is downloaded. The loadings on the co-allocated flows are automatically adjusted because the faster servers will deliver more quickly providing larger portions of the file.
- Aggressive Load Balancing: Another dynamic co-allocation strategy, presented in [5], is the Aggressive Load Balancing. The Aggressive Load Balancing dynamic co-allocation strategy presented in [5] adds functions that change block size deliveries by: (1) progressively increasing the amounts of data requested from faster servers, and (2) reducing the amounts of data requested from slower servers or ceasing to request data from them altogether.
- Neither prediction nor heuristics approaches, the DCDA scheme dynamically co-allocates duplicate assignments and copes

nicely with changes in server speed performance. When a requested block is received from a server, one of the unassigned blocks is assigned to that server. The co-allocator repeats this process until all blocks have been assigned. DCDA behaves well even when server links are broken or idled. The DCDA scheme is flawed: it consumes network bandwidth by repeatedly transferring the same blocks. This wastes resources and can easily cause bandwidth traffic jams in the links between servers and clients.

The ARAM [25] scheme adjusts the workloads on selected replica servers and handles unpredictable variations in network performance. Our algorithm uses the finish rates of the previously assigned transfers to anticipate the bandwidth status for the next selection, adjust workloads, and reduce file transfer times in grid environments. Our approach is useful in grid environments with the unstable network. It not only reduces the idle time that waiting for slowest server, but also decreases the file transfer completion times.

2.3. Medical images

Medical image processing technique refers to procedures that use special equipment to capture images of various body organs for clinical purposes (medical procedures intended to aid in diagnosis or examine diseases) or medical science researches (including normal anatomy and function).

This technique refers to one aspect of biological imaging, incorporating radiology, radiological sciences, endoscopy, thermography, medical photography, and microscopy for human pathological investigations.

In clinical applications, medical imaging is also known as radiology or "clinical imaging". Diagnostic radiography indicates the technical aspects of medical imaging and focus on the acquisition of medical images. Radiologic technologists and physicians are responsible for acquiring high-quality medical images of diagnostic and performing radiological interpretations.

In the fields of Medicine, Medical Engineering, Medical Physics, and Bioinformatics, Medical Imaging is usually defined as the technology for image creation, retrieval and storage. Researches on medical image applications and interpretations are classified as radiology or other relevant medical sub-disciplines, areas of medical science and neuroscience, cardiology, psychology, etc. Many techniques developed for medical imaging also have scientific and industrial applications.

Medical imaging can be seen as the solution of mathematical inversion problems, all related analyses are inferred from observed signals. In the case of ultrasonic devices, the probe produces ultrasonic pressure waves and echoes inside the tissue to show the body's internal structure. A projection radiography probe produces X-ray radiation which is absorbed at different rates by various tissue types such as bone, muscle, and fat.

Modern medical imaging technology includes:

- X-ray: Radiographs, more commonly known as X-rays, are often used to determine the type and extent of a fracture as well as to detect pathological changes in the lungs.
- Computed Tomography (CT): Digital geometry processing is used to generate a 3D image of the inside of an object from a large series of 2D X-ray images taken around a single axis of rotation.
- Ultrasound: Medical ultrasonography uses high frequency sound waves of between 2.0 and 10.0 MHz to produce 2D images, traditionally on a TV monitor.
- Magnetic Resonance Imaging (MRI): MR imaging uses a powerful magnetic field, radio waves and a computer to produce detailed pictures of organs, soft tissues, bone and virtually all other internal body structures.
- Gamma camera: Gamma rays (denoted as γ) are the form of electromagnetic radiation or the light emission of frequencies produced by sub-atomic particle interactions, such as electron–positron annihilation or radioactive decay.
- Positron Emission Tomography (PET): It is a nuclear medicine imaging technique which produces a 3D image or maps of functional processes in the body. The system detects pairs of gamma rays emitted indirectly by a positron-emitting radionuclide (tracer), which is introduced into the body on a biologically active molecule. Images of tracer concentration in 3D space within the body are then reconstructed by computer analysis.
- Others: Fluoroscopy, Angiography, Microscopy, Photo Acoustic Imaging, Thermography, Endoscopy, and etc.

ImageJ [26] is a public domain Java-based image processing software developed by the National Institutes of Health that runs on Windows, Mac OS, Mac OS X, Linux, Sharp PDA, and other platforms. It can display, edit, analyze, process, save and print 8-bit, 16-bit, and, 32-bit images in TIFF, GIF, JPEG, BMP, DICOM, FITS and raw image formats. It supports stacks and series of images that share a single window and provides multithreading for time-consuming operations. The accessing of image files could be performed in parallel with other operations. ImageJ is a free open source software that supports custom upgrades, edits and plug-ins. It has a built-in editor and Java compiler, and provides users with any IDE to directly process images.

3. System design and implementation

3.1. System architecture

MIFAS was deployed in the co-allocation data grid with Globus Toolkit 4.0.7. We aggregated desktop PCs and servers to establish a data grid. The descriptive medical image information (metadata) about logical data items is stored in the MIFAS Catalog Service. The four-layer architecture of the data grid is shown in Fig. 2. The yellow parts are development and implementation by Health-Box. Health-Box (H-Box) is a set of integrated scripts developed by the High Performance Lab at Tunghai University. It provides a quick way to form grid environments and integrated grid framework sets. H-Box integrates the grid middleware, Globus Toolkit, to connect the nodes, then installs necessary software, including MPICH [27], Ganglia [11], NWS [28], SYSSTAT [5], SRB [29], JDK [30], Apache Ant [31], XML-Parser [32], and xinetd [33]—all Open Source Software. We provide a grid manager to download, distribute, and modify it. Thus, we can use H-Box to quickly form prototyping of medical imaging storage grid architectures.

3.2. System flow

3.2.1. System workflow

Our design for the co-allocation grid is shown in Fig. 3. As Web-based Enquiries PACS, every client node access point uses the Cyber Agent to enter the co-allocation data grid, and manage queries and image retrievals. Overall, the benefit of our method is to speed up query accessing and image retrieving. It also provides the security for queries and image retrievals in the data grid environment.

3.2.2. Simple cyber agent transformer workflow

Fig. 4 shows the simple Cyber Agent Transformer workflow and transfer steps. Basically, physicians search and retrieve medical images via the MIFAS co-allocation data grid. The steps of the MIFAS co-allocation data grid workflow are described below. As users want to access the data grid, they must first set up a User Certificate, Private Key, Certificate Authority (CA) file, and a Proxy File for retrieving the data grid authentication. After the access is

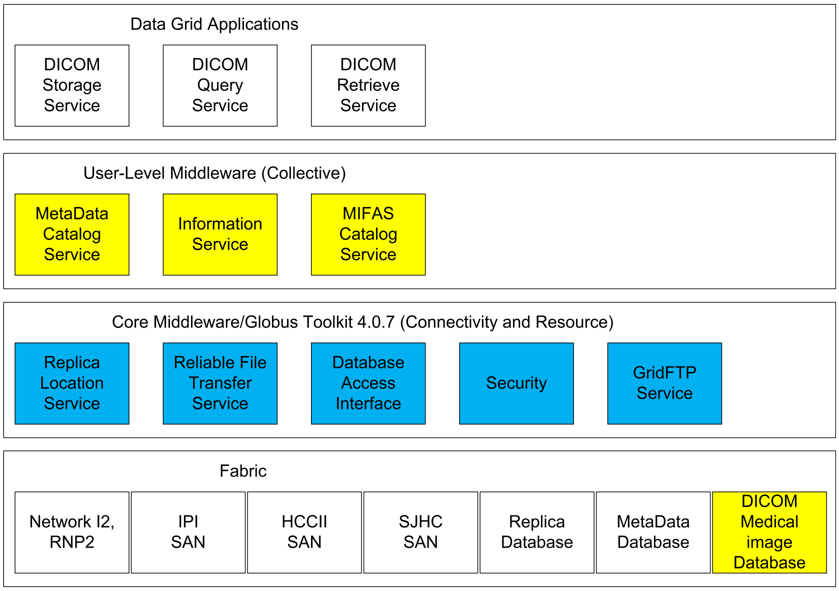


Fig. 2. Overview architecture of MIFAS in co-allocation data grid. Yellow: developed at HPCLab of THU; blue: globus toolkit.

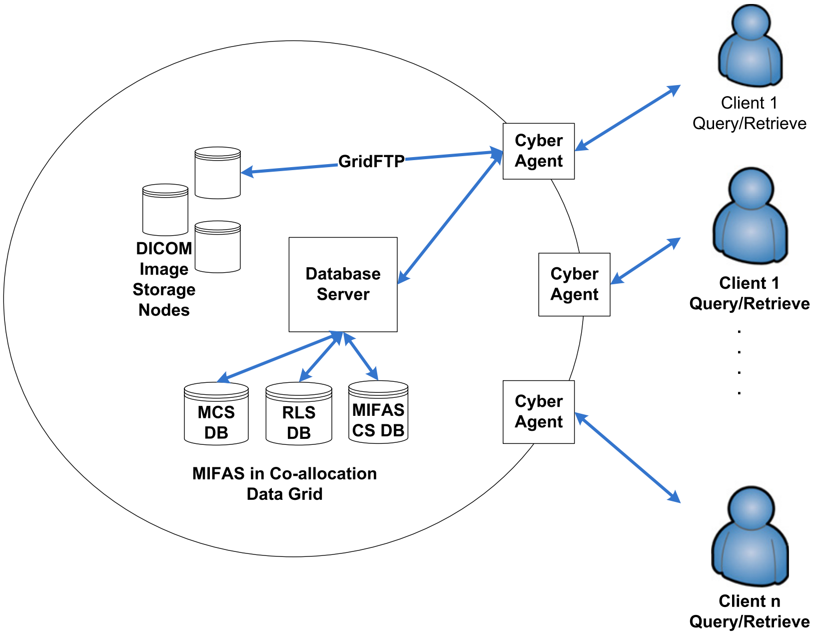


Fig. 3. Workflow overview of medical image in co-allocation data grid.

granted, users can retrieve medical image information from the MIFAS Catalog Service database listed in Cyber Agent Transformer.

When a user requests for medical images or DICOM Image Storage nodes from Cyber Agent, the request is processed. If any storage node is available, it is obtained from the remote medical image storage via the GridFTP protocol, which could speed up whole transfers. This retrieval method is secured by the Globus certificate authorization service. The cyber agent uses the MIFAS co-allocation method to download images in parallel. As the transfer is finished, the user views the medical images by ImageJ. In general, this is a convenient method for downloading medical images from a data grid.

3.2.3. The cyber agent transformer

In the previous work [34], we gave experimental results for the Cyber Agent Transformer, a powerful new toolkit for the replica management and data transfer in data grid environments. It not only accelerates data transfer rates, but also manages replicas over various sites. The friendly interface enables users to easily monitor replica sources, and add files as replicas for automatic cataloging by our Replica Location Service. Moreover, we provide a function for administrators to delete and modify replicas. The Cyber Agent Transformer can be invoked with either the logical file name of a data file or a list of replica source host names. When users search for files by the logical file name, the Cyber Agent Transformer searches Replica Location Services to find all corresponding replicas, and notifies each source to start parallel transfers. The file is then gathered from replica sources and finally combined into a single file.

3.2.4. Cyber agent transaction flow

Fig. 5 shows the Cyber Agent Transaction flow. In order to obtain accesses to the grid, users must first set up the User Certificate, the Private Key, the CA file, the Proxy File, and IP address. They

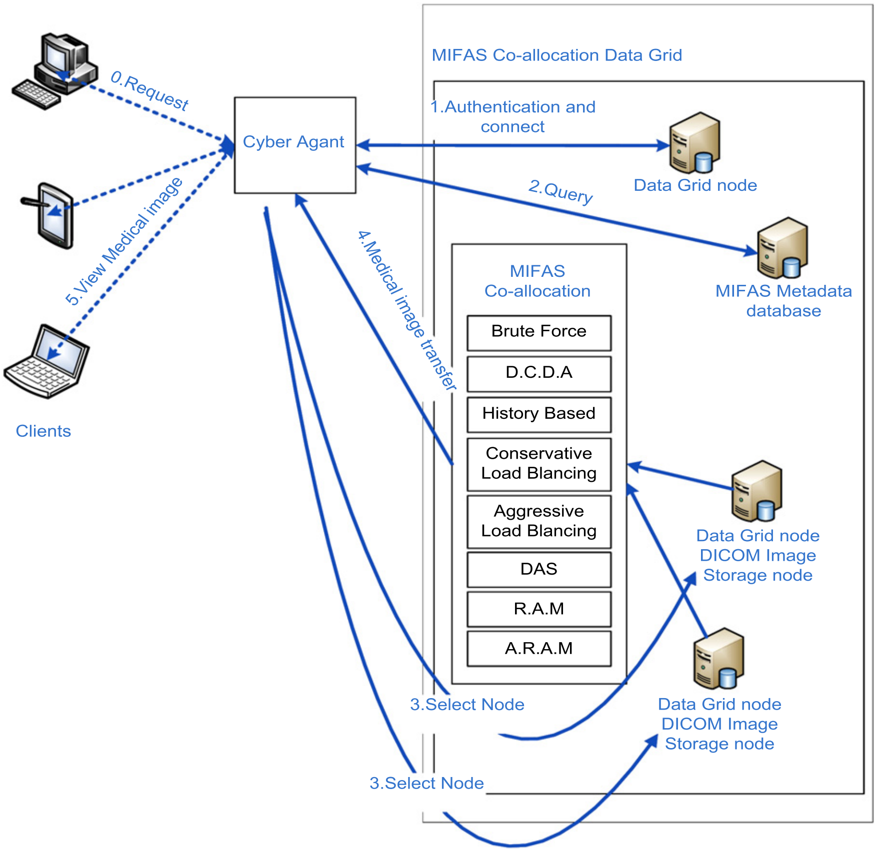


Fig. 4. Simple workflow for cyber agent transformer.

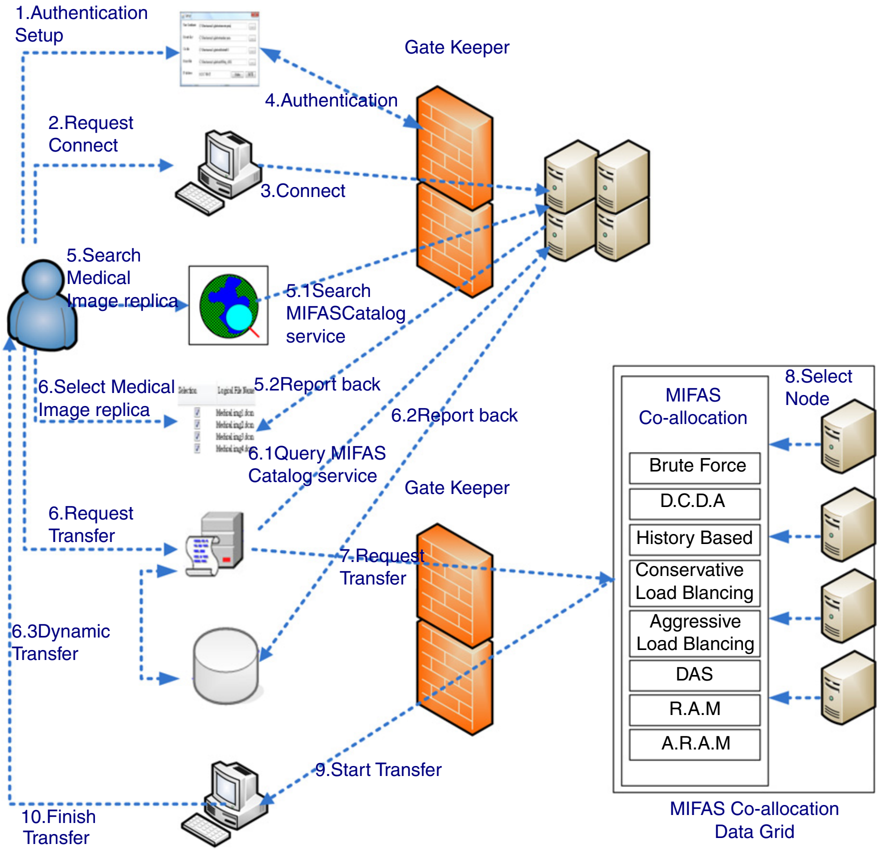
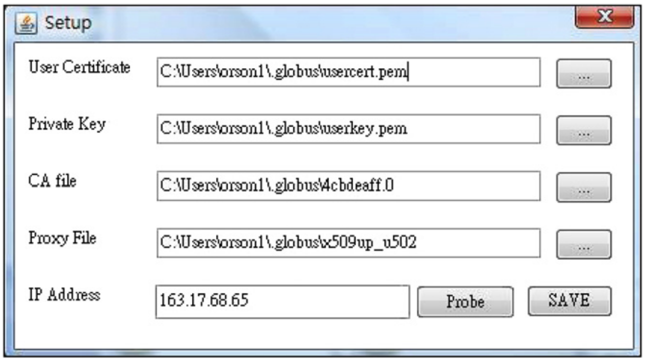


Fig. 5. Cyber agent transaction flow.



User Certificate: C:\Users\orson1\.globus\usercert.pem

Private Key: C:\Users\orson1\.globus\userkey.pem

CA file: C:\Users\orson1\.globus\4cbdeaff.0

Proxy File: C:\Users\orson1\.globus\x509up_u502

IP Address: 163.17.68.65

Fig. 6. Cyber agent transformer setup.

may then connect to any data grid site via the GridFTP Browser. The system automatically authenticates the site certification as the connection is made. The security mechanism of our grid environment is described below.

Steps 5 and 6 show how users search the Medical Image Replica Service for the MIFAS catalog service, and how requests are reported. The system ranks all replica servers according to our replica selection model [7,35,8,6,36,37], and users can then choose the better servers for parallel downloading. Parallel downloading [38,6,36,34] is a technique used to fetch and download files from multiple sources including Web servers, file servers, P2P nodes, etc.

The Data Transfer Service is invoked in step 6. Information on replicas chosen by the user is picked up by the GridFTP Job Controller. MIFAS Co-allocation is then used to transfer the desired files.

3.3. GUI and system operations

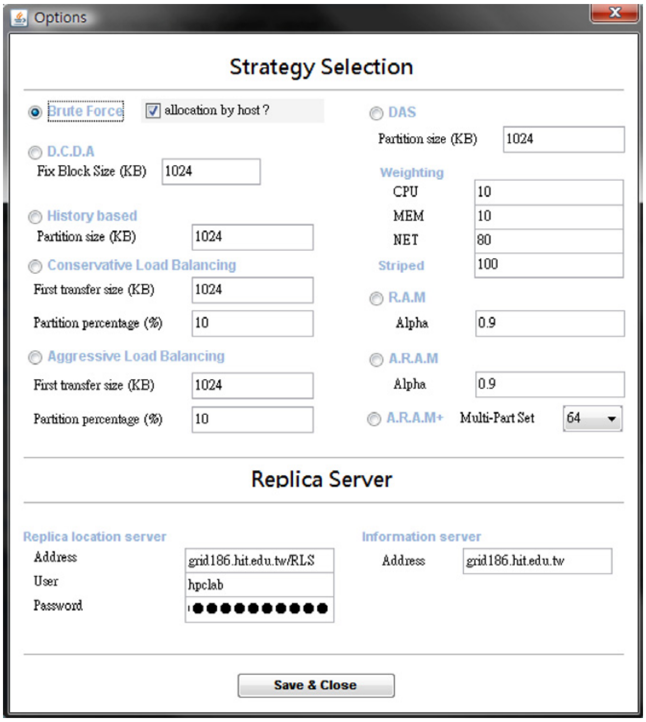
We developed and implemented a user-friendly GUI for Cyber Agent Transformer to help users to download and manage files in data grid environments. It was implemented in the Java CoG library, and can be run on any operating system with JVM. The entire set up and operation process is shown below.

- Authentication Setup: To conduct the image file retrieval, the user must setup the Cyber Agent Transformer environment—as shown in Fig. 6. Users are required to provide certain security mechanisms, such as the user certificate, the private key, the certificate authority file, and the proxy file (conforming to X.509) to provide a safe transfer environment.
- Strategy Selection: The user can set the relative parameters for a different algorithm before file downloading. There are eight algorithms for downloading medical image files. Fig. 7 shows the GridFTP client tool used for file transferring.
- User Tools: We designed tools to assist users in downloading medical image files, and setting up some environment configurations. Fig. 8 shows the user tools.
- Selection: Search Replica is used to find medical image files for downloading. Fig. 9 shows selecting a Replica download from a candidate node.
- Messages: Users can see statuses, transfer times, and transfer rates for all download in the message box. Fig. 10 shows the job download and completion dialogs.

4. Experimental environments and results

4.1. Medical image environments

Our experiments were conducted and evaluated on the TIGER grid, which consists of more than one hundred processors distributed over ten clusters located at seven educational institutions: Tungs' Taichung MetroHarbor Hospital (TUNG), Tunghai University (THU), National Changhua University of Education (NCUE), National Taichung University (NTCU), Hsiuping Institute of Technology (HIT), National Da–Li Senior High School (DL), Lizen High School (LZSH) and Long Fong Elementary School (LFPS). The TIGER grid network environment logical diagram is shown in Fig. 11, Fig. 12 presents the Ganglia monitor page of our test-bed. Fig. 13 shows all grid test-bed machine statuses on one monitor page.



Strategy Selection

● Brute Force ☑ allocation by host ?

○ DAS
Partition size (KB): 1024

○ D.C.D.A
Fix Block Size (KB): 1024

Weighting
CPU: 10
MEM: 10
NET: 80

○ History based
Partition size (KB): 1024

Striped: 100

○ Conservative Load Balancing
First transfer size (KB): 1024
Partition percentage (%): 10

○ R.A.M
Alpha: 0.9

○ Aggressive Load Balancing
First transfer size (KB): 1024
Partition percentage (%): 10

○ A.R.A.M
Alpha: 0.9

○ A.R.A.M+ Multi-Part Set: 64

Replica Server

Replica location server
Address: grid186.hit.edu.tw/RLS
User: hpclab
Password: ●●●●●●●●●●

Information server
Address: grid186.hit.edu.tw

Fig. 7. Download strategy selection.

Table 1
The end-to-end transmission rates from THU to others using NWS (Mbps).

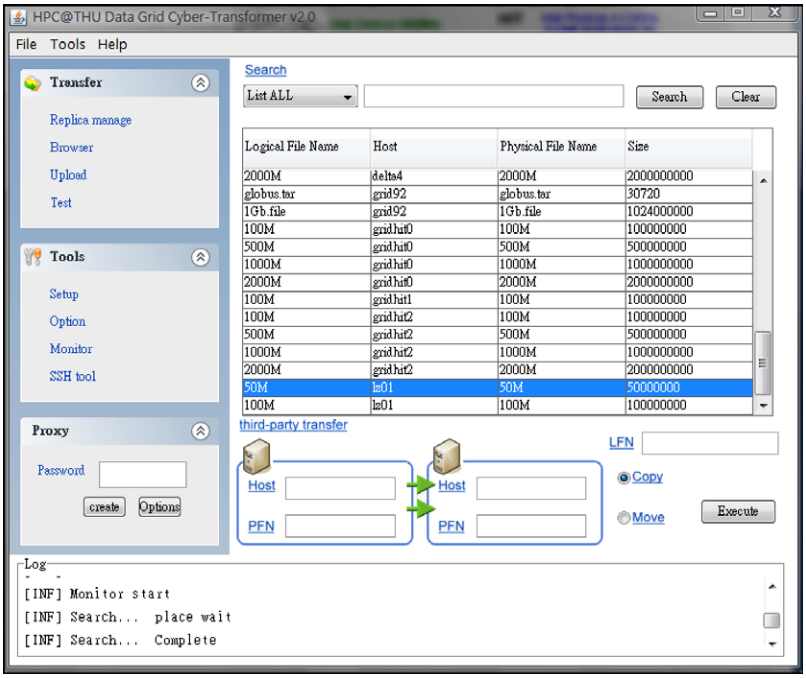
Case	Bandwidth average	High	Low
THU → HIT	37.815	70.349	20.952
THU → LZ	48.139	73.466	31.678
THU → DL	16.673	17.920	12.182
THU → NTCU	23.432	39.824	13.176
THU → TUNG	3.683	3.774	3.529

They are interconnected by the 1 Gbps Taiwan Academic Network (TANET). The TIGER grid platform consists of approximately 60 computing nodes with some 224 CPUs running at various speeds and a total storage capacity of approximately 5TB. All these institutions are in Taiwan, and each is at least 10 km from Tunghai University (THU). All machines have Globus 4.0.7 or higher installed. End-to-end transmission rates from THU to each educational unit are listed in Table 1.

We used the Cyber Agent Transformer to perform wide-area data transfer experiments. We deployed our co-allocation client on our test-bed at Tunghai University (THU) in Taichung City, Taiwan, and fetched files from selected replica servers at: National Da–Li Senior High School (DL), Lizen High School (LZ), Tungs' Taichung MetroHarbor Hospital (TUNG), and Hsiuping Institute of Technology School (HIT). These institutions are all in Taichung, Taiwan, 10–30 km from THU.

4.2. Cross-hospital PACS architecture

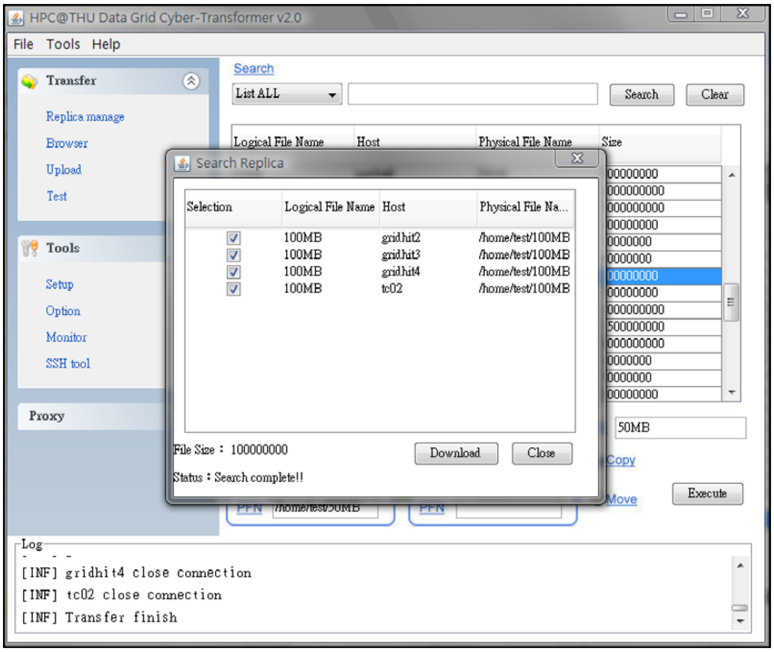
Using the TIGER Grid system, we tried to simulate a PACS serving two or more hospitals, and performed several experiments on issues of concern.



Logical File Name	Host	Physical File Name	Size
2000M	delta4	2000M	2000000000
globus.tar	grid92	globus.tar	30720
1Gb.file	grid92	1Gb.file	1024000000
100M	gridhit0	100M	100000000
500M	gridhit0	500M	500000000
1000M	gridhit0	1000M	1000000000
2000M	gridhit0	2000M	2000000000
100M	gridhit1	100M	100000000
100M	gridhit2	100M	100000000
500M	gridhit2	500M	500000000
1000M	gridhit2	1000M	1000000000
2000M	gridhit2	2000M	2000000000
50M	lz01	50M	50000000
100M	lz01	100M	100000000

```
[INF] Monitor start
[INF] Search... place wait
[INF] Search... Complete
```

Fig. 8. User tools.



Selection	Logical File Name	Host	Physical File Na...
☑	100MB	gridhit2	/home/test/100MB
☑	100MB	gridhit3	/home/test/100MB
☑	100MB	gridhit4	/home/test/100MB
☑	100MB	tc02	/home/test/100MB

File Size : 100000000

Status : Search complete!!

50MB

/home/test/50MB

```
[INF] gridhit4 close connection
[INF] tc02 close connection
[INF] Transfer finish
```

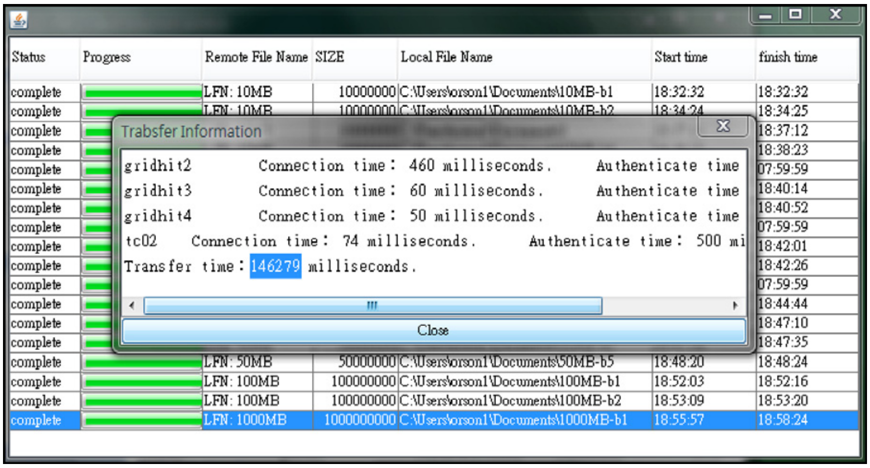
Fig. 9. Selecting replica download from candidate node.

4.3. Experiment 1: compare 8 co-allocation schemes for downloading different data sizes

In this experiment, we downloaded the files whose size is from 10 MB to 1 GB with the eight MIFAS Co-allocation algorithms. The results show the best algorithm varied according to the file size downloaded. Fig. 13 shows the download data for the eight transfer modes. Fig. 14 shows no differences among the eight MIFAS Co-allocation algorithms on downloading a 10 MB file. In this case, all methods used to query or retrieve files would be completed around the same time. Fig. 14 also shows that the ARAM algorithm was the best method for 50 MB–1 GB files. This experiment showed the best transmission method, which were used to perform other experiments.

4.4. Experiment 2: compare query/retrieval times from local grid node using ARAM and web PACS

In this experiment, we simulated a Web PACS in the local grid node and used the best transfer method, ARAM, for comparison tests. Physicians may need to search and retrieve the files listed in Table 2 for diagnosis or to compare medical cases. These files are usually X-ray images, CT scans, or series' of CT scans. In order to compare the difference in data retrieval performance between the Web PACS and the Cyber Agent Transformer, we customized test-bed A, as shown in Fig. 15. The Cyber Agent Transformer retrieved images via parallel-download from Medical Data Grid B (Data Flow B, Fig. 15), whereas the Web PACS retrieved from the Web PACS (Data Flow A in Fig. 15). The times for the MIFAS Co-allocation



Status	Progress	Remote File Name	SIZE	Local File Name	Start time	finish time
complete		LFN: 10MB	10000000	C:\Users\orson1\Documents\10MB-b1	18:32:32	18:32:32
complete		LFN: 10MB	10000000	C:\Users\orson1\Documents\10MB-b2	18:34:24	18:34:25
complete						18:37:12
complete						18:38:23
complete						07:59:59
complete						18:40:14
complete						18:40:52
complete						07:59:59
complete						18:42:01
complete						18:42:26
complete						07:59:59
complete						18:44:44
complete						18:47:10
complete						18:47:35
complete		LFN: 50MB	50000000	C:\Users\orson1\Documents\50MB-b5	18:48:20	18:48:24
complete		LFN: 100MB	100000000	C:\Users\orson1\Documents\100MB-b1	18:52:03	18:52:16
complete		LFN: 100MB	100000000	C:\Users\orson1\Documents\100MB-b2	18:53:09	18:53:20
complete		LFN: 1000MB	1000000000	C:\Users\orson1\Documents\1000MB-b1	18:55:57	18:58:24

Trabsfer Information

```
gridhit2        Connection time： 460 milliseconds.     Authenticate time
gridhit3        Connection time： 60 milliseconds.      Authenticate time
gridhit4        Connection time： 50 milliseconds.      Authenticate time
tc02    Connection time： 74 milliseconds.     Authenticate time： 500 mi
Transfer time：146279 milliseconds.
```

Fig. 10. Job download and completion dialogs.

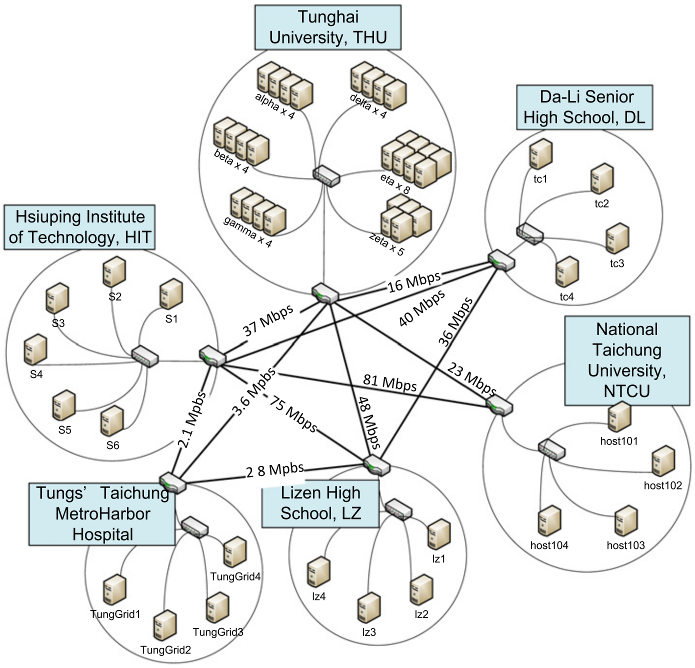


Fig. 11. TIGER grid topology.

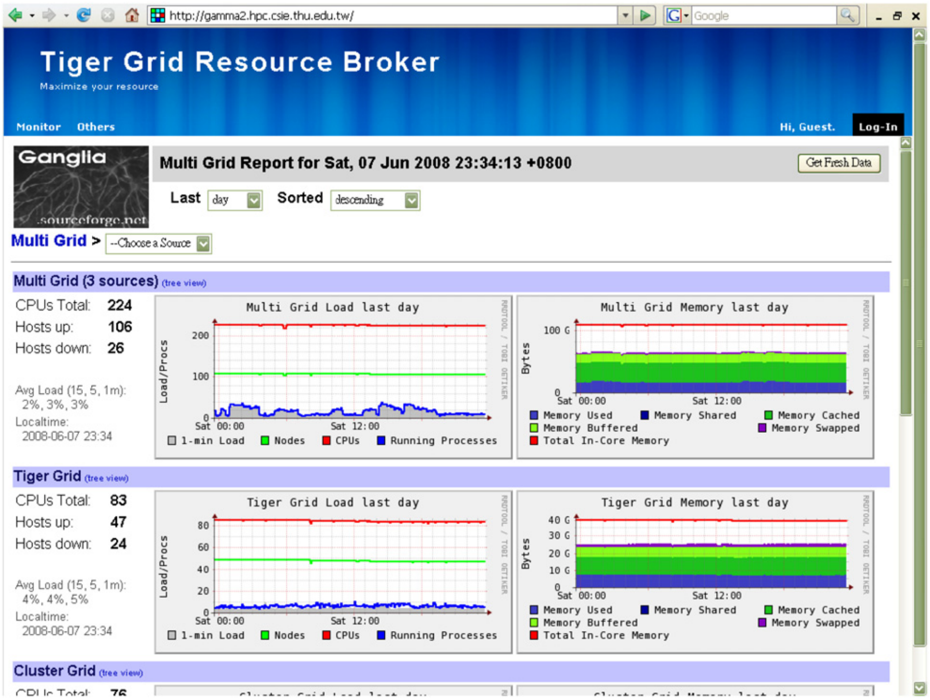
ARAM and Web PACS are shown in Figs. 16 and 17. The results show that the performance in the end-to-end query and the retrieval of the first medical image by ARAM was better for all sizes than those by Web PACS. And the average transfer time was better than that of the Web PACS. We then tested retrieving ImageJ from the Medical Data Grid, as shown in Fig. 18.

4.5. Experiment 3: compare query/retrieve times from remote grid node using ARAM and web PACS

Experiment 2 showed that ARAM obtained images more efficiently than the Web PACS while it was based on a local grid node. We transferred the experiment to a wide-area network for extending the MIFAS performance to a broader zone. An experiment was performed to prove that MIFAS is also better than Web PACS in wide-area networks. In order to confirm the transfer performance of ARAM, the comparison tests on the images listed in Table 2 was performed. Test-bed B was customized as shown in Fig. 19. Cyber Agent Transformer retrieved images via parallel-download from Medical Data Grid C (Data Flow B, Fig. 19), whereas the Web PACS retrieved images from Web PACS (Data Flow A, Fig. 19). The MIFAS Co-allocation ARAM and Web PACS times for remote grid node downloads are shown in Fig. 20. The results was presented in Fig. 21. The experimental data is the contents of Table 2. Case 1 has 42 pictures of CT. Case 2 has 180 pictures of CT. Case 3 has 5 pictures of X-ray. The expected test got the first medical image. Relatively, all download pictures should be tested fairly by the average amount of time. Therefore, the Fig. 21 shows the transfer average times for each medical image. The results show ARAM was better than Web PACS.

Table 2
Query and retrieve for X-ray, CT.

Image query and retrieve	Image data (MB)
CT 42 512 × 512	∼22
X-ray Chest 5 2320 × 2828	∼65
A series of CT case 180 512 × 512	∼79



http://gamma2.hpc.csie.thu.edu.tw/

Tiger Grid Resource Broker

Maximize your resource

Monitor Others Hi, Guest. Log-In

Multi Grid Report for Sat, 07 Jun 2008 23:34:13 +0800

Last: day Sorted: descending

Multi Grid > --Choose a Source

Multi Grid (3 sources) (tree view)

CPUs Total: 224
Hosts up: 106
Hosts down: 26

Avg Load (15, 5, 1m): 2%, 3%, 3%
Localtime: 2008-06-07 23:34

Tiger Grid (tree view)

CPUs Total: 83
Hosts up: 47
Hosts down: 24

Avg Load (15, 5, 1m): 4%, 4%, 5%
Localtime: 2008-06-07 23:34

Cluster Grid (tree view)

CPUs Total: 76

Fig. 12. Ganglia page of TIGER.

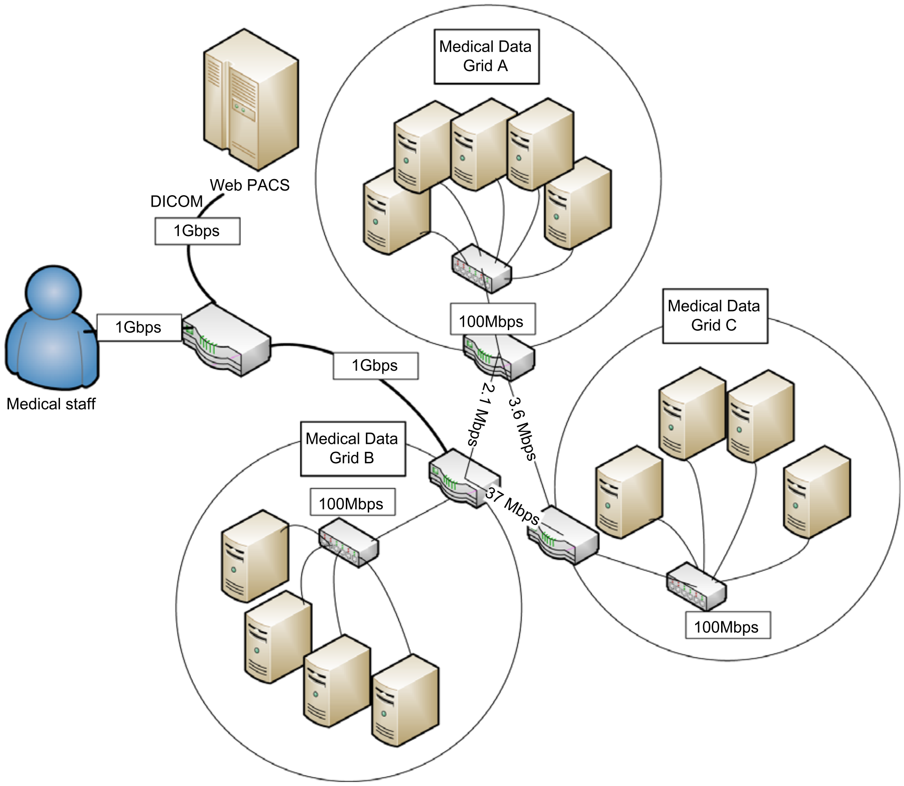


Fig. 13. Cross-hospital PACS architecture.

4.6. Experiment 4: conventional medical image exchange vs. co-allocation download

All the medical images that exchanged between hospitals are produces from high-level imaging systems such as 64-slice CT and 3.0T MRI. There are various PACSs simulated to exchange the 64-slice CT and 3.0T MRI images listed in Table 3. In order to compare the difference in data retrieval performance between PACSs, testbed C was customized to transfer images from the THU PACS to the HIT PACS, as shown in Fig. 22.

The Cyber Agent Transformer exchanged images in the conventional way (The data flow A in the Fig. 22), as compared to parallel downloading from the Medical Data Grid (The data flow B in the Fig. 22). MIFAS Co-allocation and DICOM transfer results are shown

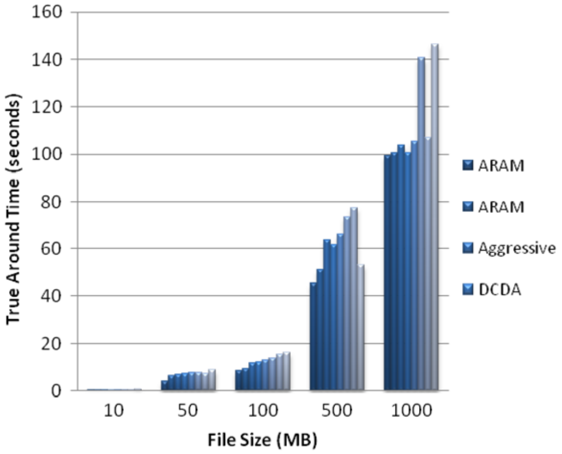


Fig. 14. Completion time for various file size and co-allocation algorithms.

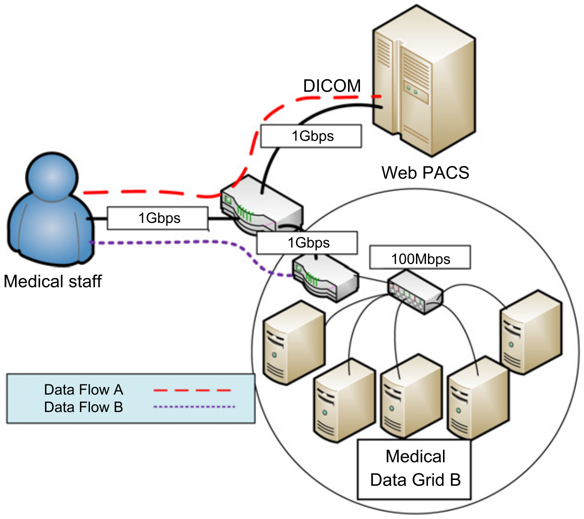


Fig. 15. Our test-bed A.

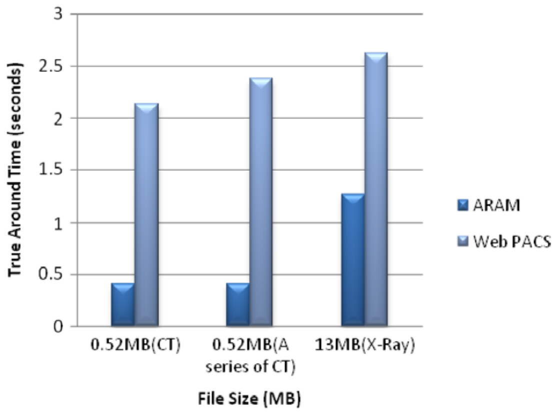


Fig. 16. Compare query/retrieval times for the first image from local grid node using ARAM and web PACS.

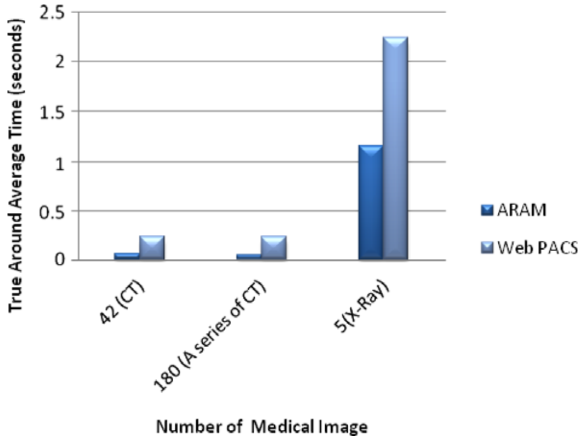


Fig. 17. Compare average ARAM transfer times from local grid node and web PACS.

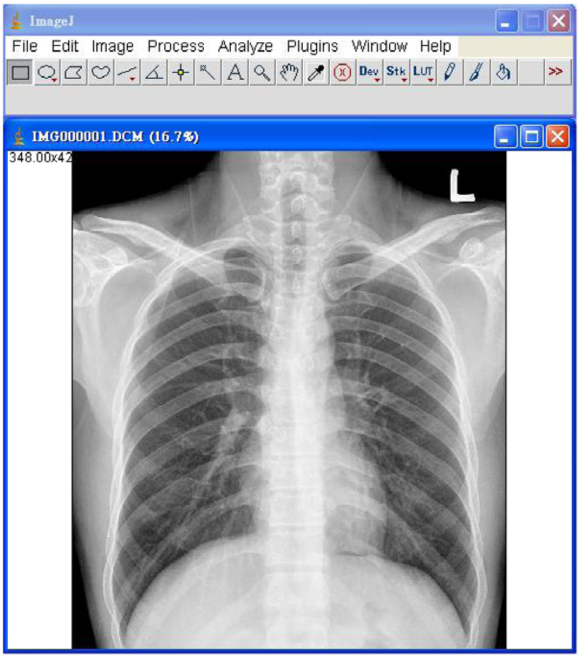


Fig. 18. Medical image display.

Table 3
Estimated sizes for 64-slice CT and 3.0T MRI exchange images.

Image exchange	Image data (MB)
3.0 T (3.0T) MRI 1400. (Video content is about 128 × 128 or 256 × 256 or 512 × 512 or 560 × 560 or 576 × 576 or 640 × 640 images)	∼499.8
64-slice CT Scanner 2000 512 × 512	∼1058

in Figs. 23 and 24. The Fig. 23 shows the transfer times for case 1. We can see the ARAM and PACS transfer times. ARAM transfer time for the 3.0T medical images is near 45.297 s. PACS transfer time for the 3.0T medical images is near 140 s. ARAM transfer data is near 88.306069 Mpbs. PACS transfer data rate is near 28.57142857 Mpbs. Therefore, ARAM is better than PACS. The Case 2 transfer time is shown in Fig. 24. ARAM transfer time for the 64-slice CT medical images is near 102.453 s. PACS transfer time for the 64-slice CT medical images is near 310 s. ARAM transfer data rate is near 78.084585 Mpbs. PACS transfer data rate is near 25.80645161 Mpbs. ARAM is better than PACS. The graphs show that the Co-allocation downloading performed better than the DICOM protocol. Downloading from multiple image replicas improves overall performance.

4.7. Experiment 5: compare 8 co-allocation schemes on stable and broken network links

The system reliability of the hospital must be enhanced since broken network links could cause file transfer failures in complex heterogeneous grid environments. Physicians use the Web

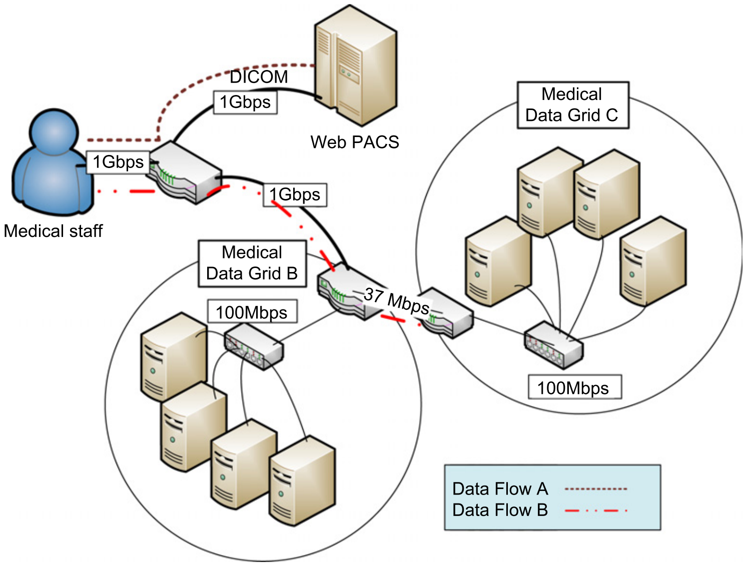


Fig. 19. Our test-bed B.

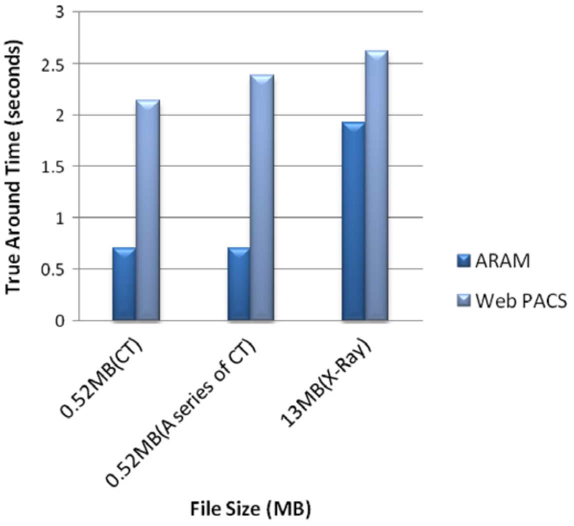


Fig. 20. Compare query/retrieval times for the first image from remote grid node using ARAM and web PACS.

PACS for medical image queries, retrievals, and exchanges. Unfortunately, network interruptions will delay or halt the Web PACS and medical image exchanges. Network fault tolerance design is important in improving the usability and reliability of the full grid system. Among all co-allocation methods, only ARAM was able to overcome network faults.

As shown in Figs. 25–27, we built an environment with replicas at four grid sites. Each site was disconnected in a period of time during file transfer. The ARAM scheme was designed to continue file transfers from the previous point, and avoid having faster sites waiting for slower sites. Therefore, the overall file transfer performance keeps stable.

4.8. Experiment 6: image machine recovery

In the near, Tungs' Taichung MetroHarbor Hospital experienced a Web PACS breakdown that affected physicians in diagnosing patients. It took us 6 months to reconstruct the Web PACS without affecting ongoing general operations and providing image data to physicians. (We used only the off-peak times of Monday to Friday from 22:00 pm to 8:00 am, Saturday afternoon, and Sunday.

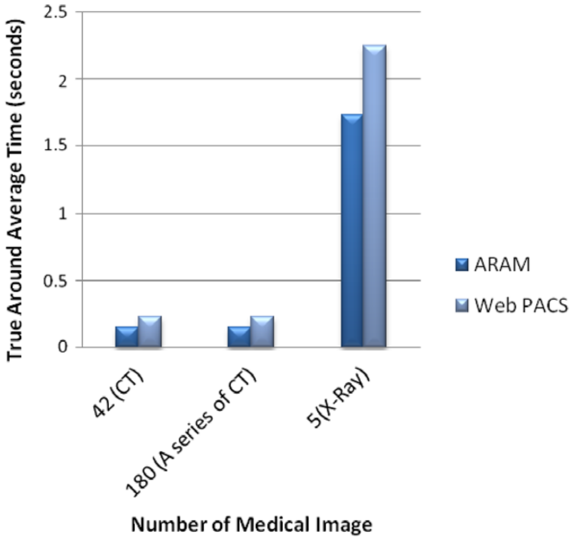


Fig. 21. Compare the average transfer times from a remote grid node for ARAM and web PACS.

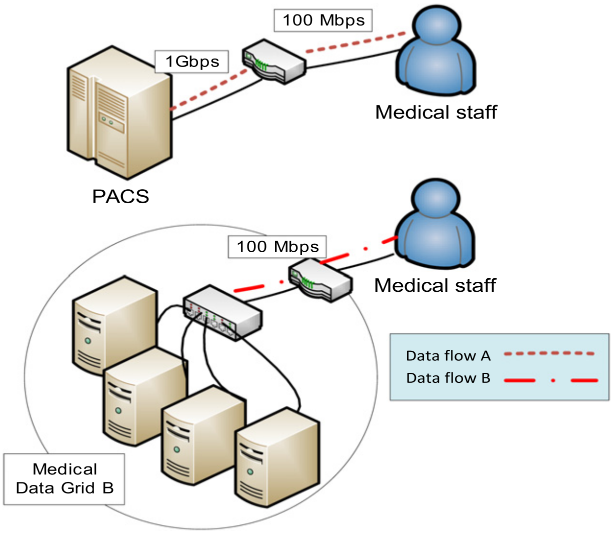


Fig. 22. Our test-bed C.

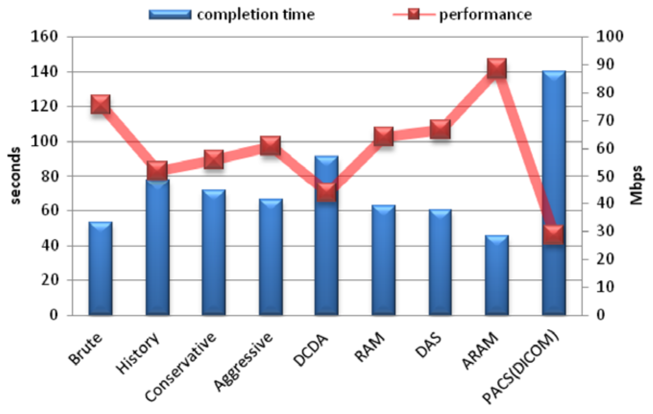


Fig. 23. 3.0 T performances on (3.0T) MRI.

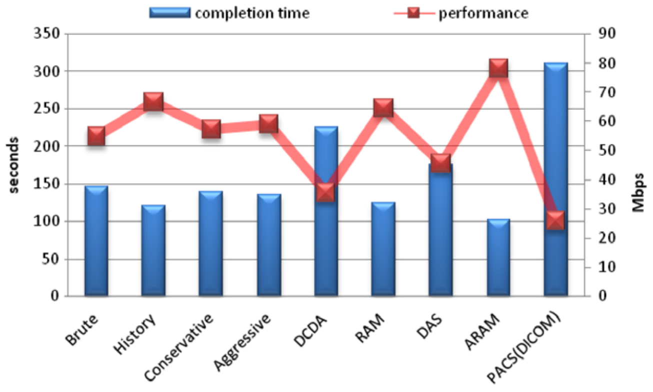


Fig. 24. Performance of 64-slice CT.

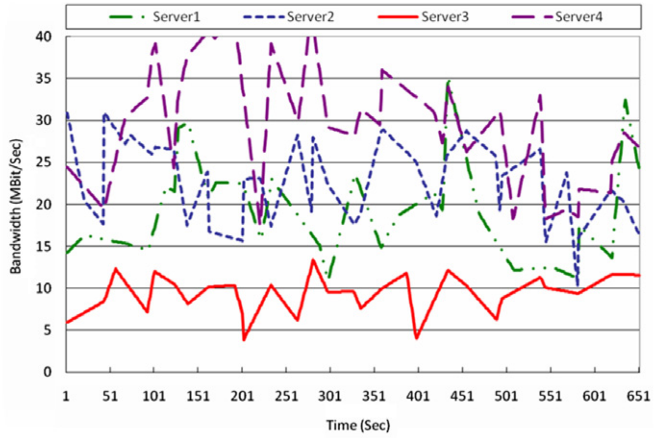


Fig. 25. Stable network link.

If we had been able to focus completely on reconstruction, it would have taken approximately 30 days.) In addition to the expense for machine replacements (over 1 million to date, and eventually exceed US$ 125,000–156,250 dollars if the Web PACS is incomplete), we mobilized many hospital staff members and medical imaging system specialists, and spent excessive time on the reconstruction.

A simulation on this incident was performed with grid environment to reduce time, expense and manpower, as well as to minimize the effect on physicians' diagnoses. We utilized the H-Box to construct medical image storage grid architecture and enhance grid stability and security. We analyzed the following aspects from a managerial perspective.

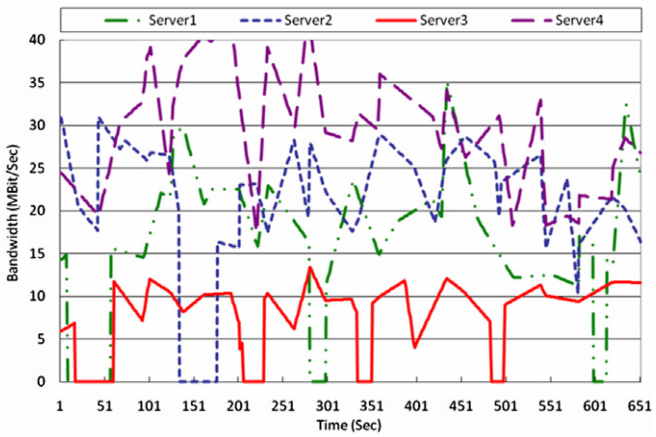


Fig. 26. Broken network link.

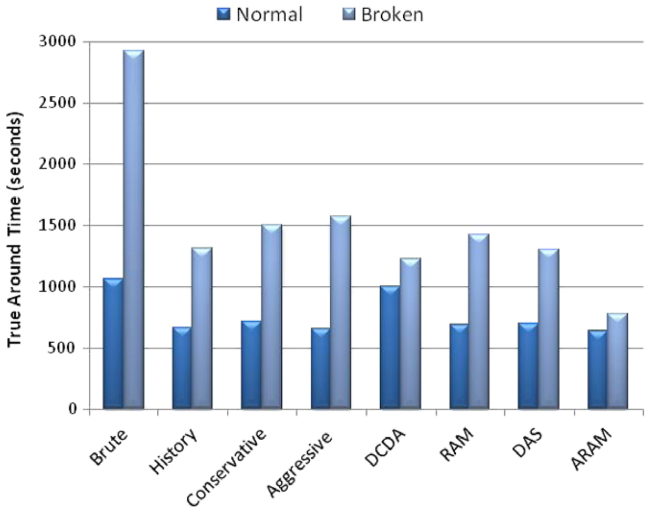


Fig. 27. Compare various schemes on various network statuses.

- Management: Managers are able to construct a medical image storage environment using grid techniques and methods without asking medical imaging specialists and other personnel for help. Consequently, we can use these resources to train the staff ourselves.
- Hardware and software: We spent less on the new grid node than on the Web PACS. Desktop PC with large-capacity storage (PC and 6 TB storage*2), only about 150,000 dollars, which is a lot cheaper than the Web PACS machine. And, the software and techniques we used are all open source and freeware, whereas all the Web PACS software must be purchased.
- Time: We reduced the time required to construct a medical image storage environment with the new grid node, and the time for medical image recovery. Efficiently recovering environments are usually important for hospitals, and same as reducing waiting times and medical staff costs.

In this paper, we report the construction of a grid node within one hour using H-Box, while on the other hand, it took us 5–6 h to construct a new Web PACS and install all the software. It took 6 months working only during off-peak hours to recover these medical images, and 20 days for full-time recovery (about 10 TB capacities). With a grid node completely focused on recovering medical images, MIFAS takes about near 8 days. Considering these options, we organized the data and presented them in a simple

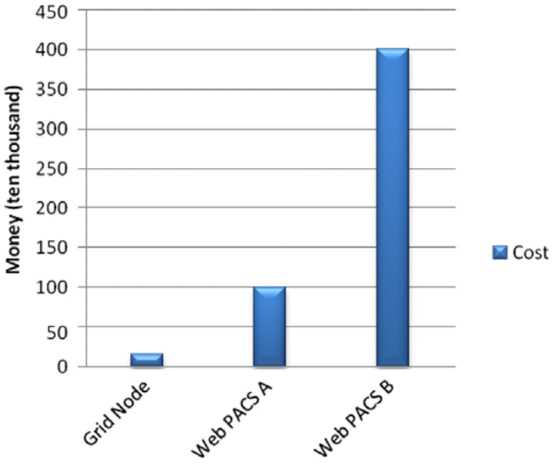


Fig. 28. Compare machine replacement cost.

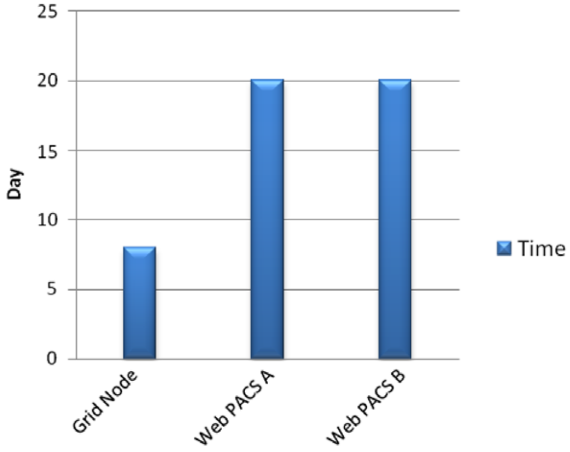


Fig. 29. Compare storage environment construction and medical image recovery times.

way. Fig. 28 shows both are better than the Web PACS-related software and hardware. And Fig. 29 shows both constructed and recovered data faster than the Web PACS.

5. Conclusions

This paper proposed the MIFAS solution to reduce medical images' transfer time, and integrated medical image process technique into co-allocation data grid environments. We also report on implementing a Cyber Agent Service that enables users to use co-allocation data grid. Currently, MIFAS offers four advantages.

First, compared with PACS, MIFAS can reduce co-allocation transfer time from experimental results. Second, MIFAS provided a fast, secure, stable, reliable system for obtaining medical images. Three, co-allocation architecture enables parallel downloading from a co-allocation data grid. It can also speed up downloads and overcome network faults. Four, we provided easy management, reduced expense, and increased the stability of medical image system.

We reported on successfully moving medical images on the MIFAS Co-allocation data grid. We proposed a means of integrating a medical image file accessing system with a co-allocation data grid to improve medical image query, retrieval, exchange, and download speeds. Our experiments showed ARAM to be the best among the eight co-allocation schemes. We found that parallel downloading via File Transfer Protocols yields better performance than single-point downloading. ARAM also overcomes the problem of broken network links. It completes transfer jobs by continuing from the previous point.

In our managerial experiments we used H-Box to build a medical image storage grid node on a desktop PC and tested image recovery with it. We found that the recovery rate and building new grid nodes are both better than making a new Web PACS. Anyone can use H-Box to establish an open source medical image storage environment, and use grid architecture to increase medical image storage stability without incurring the high cost of a Web PACS.

Furthermore, we enhanced the security with a data grid authentication environment: the User Certificate, the Private Key, the Certificate Authority (CA) File, and the Proxy File. In conclusion, Medical Image File Accessing in a co-allocation data grid provides users with a reliable and secure environment for processing queries and medical image retrievals efficiently.

References

[1] V. Breton, R. Medina, J. Montagnat, Datagrid, prototype of a biomedical grid, Methods of Information in Medicine 42 (2) (2003) 143–147.

[2] G.V. Koutelakis, D.K. Lymperopoulos, Member, IEEE, a grid PACS architecture: providing data-centric applications through a grid infrastructure, in: Proceedings of the 29th Annual International Conference of the IEEE EMBS Cite International, Lyon, France, ISBN: 978-1-4244-0787-3, August 23–26, 2007.

[3] N.E. King, B. Liu, Z. Zhou, J. Documet, H.K. Huang, in: Osman M. Ratib, Steven C. Horii (Eds.), The Data Storage Grid: The Next Generation of Fault-Tolerant Storage for Backup and Disaster Recovery of Clinical Images Medical Imaging 2005: PACS and Imaging Informatics, in: Proceedings of SPIE, vol. 5748, 2005, pp. 208–217.

[4] Z. Zhou, S.S. Chao, J. Lee, B. Liu, J. Documet, H.K. Huang, A data grid for imaging-based clinical trials, in: Steven C. Horii, Katherine P. Andriole (Eds.), Medical Imaging 2007: PACS and Imaging Informatics, in: Proc. of SPIE, vol. 6516, 2007, p. 65160U.

[5] SYSSTAT utilities home page. http://perso.wanadoo.fr/sebastien.godard/.

[6] Chao-Tung Yang, Chun-Pin Fu, Ching-Hsien Hsu, File replication maintenance and consistency management services in data grids, Journal of Supercomputing (ISSN: 1573-0484) (2010).

[7] S. Vazhkudai, Enabling the co-allocation of grid data transfers, in: Proceedings of Fourth International Workshop on Grid Computing, 17 November 2003, pp. 44–51.

[8] L. Yang, J. Schopf, I. Foster, Improving parallel data transfer times using predicted variances in shared networks, in: Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid, CCGrid '05, 9–12 May 2005, pp. 734–742.

[9] Chao-Tung Yang, Yao-Chun Chi, Ming-Feng Yang, Ching-Hsieh Hsu, An anticipative recursively-adjusting mechanism for parallel file transfer in data grids, Concurrency and Computation: Practice and Experience (ISSN: 1532-0626) (2010).

[10] C.T. Yang, I.H. Yang, K.C. Li, S.Y. Wang, Improvements on dynamic adjustment mechanism in co-allocation data grid environments, The Journal of Supercomputing, Springer 40 (3) (2007) 269–280.

[11] Ganglia. http://ganglia.sourceforge.net.

[12] Open Grid Forum. http://www.ogf.org/.

[13] The Globus Alliance. http://www.globus.org/.

[14] B. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, S. Tuecke, Data management and transfer in high-performance computational grid environments, Parallel Computing 28 (5) (2002) 749–771.

[15] B. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, S. Tuecke, Secure, efficient data transport and replica management for high-performance data-intensive computing, in: Proc. of the Eighteenth IEEE Symposium on Mass Storage Systems and Technologies, 2001, pp. 13–28.

[16] B. Allcock, S. Tuecke, I. Foster, A. Chervenak, C. Kesselman, Protocols and services for distributed data-intensive science, in: ACAT2000 Proceedings, 2000, pp. 161–163.

[17] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunszt, M. Ripeanu, B. Schwarz, H. Stockinger, K. Stockinger, B. Tierney, Giggle: a framework for constructing scalable replica location services, in: Proc. of SC 2002, Baltimore, MD, 2002.

[18] W. Hoschek, J. Jaen-Martinez, A. Samar, H. Stockinger, K. Stockinger, Data management in an international data grid project, in: Proceedings of the First IEEE/ACM International Workshop on Grid Computing-Grid 2000, Bangalore, India, ISBN: 3-540-41403-7, December 2000, pp. 77–90.

[19] S. Vazhkudai, J. Schopf, I. Foster, Predicting the performance of wide area data transfers, in: Proceedings of the 16th International Parallel and Distributed Processing Symposium, IPDPS 2002, April 2002, pp. 34–43.

[20] C.M. Wang, C.C. Hsu, H.M. Chen, J.J. Wu, Efficient multi-source data transfer in data grids, in: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, CCGRID'06, 16–19 May 2006, pp. 421–424.

[21] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, The data grid: towards an architecture for the distributed management and analysis of large scientific datasets, Journal of Network and Computer Applications 23 (3) (2001) 187–200.

[22] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunszt, M. Ripeanu, B. Schwarz, H. Stockinger, K. Stockinger, B. Tierney, Giggle: a framework for constructing scalable replica location services, in: Proc. SC, 2002, pp. 1–17.

[23] Chao-Tung Yang, Ming-Feng Yang, Wen-Chung Chiang, Enhancement of anticipative recursively adjusting mechanism for redundant parallel file transfer in data grids, Journal of Network and Computer Applications (ISSN: 1084-8045) 32 (4) (2009) 834–845.

[24] Chao-Tung Yang, Shih-Yu Wang, William C. Chu, A dynamic adjustment strategy for parallel file transfer in co-allocation data grids, in: Journal of Supercomputing, Springer Netherlands, Article, June 2009. doi:10.1007/s11227-009-0307-4.

[25] Chao-Tung Yang, Yao-Chun Chi, Ming-Feng Yang, Ching-Hsien Hsu, An anticipative recursively-adjusting mechanism for redundant parallel file transfer in data grids, in: Proceedings of the Thirteenth IEEE Asia-Pacific Computer Systems Architecture Conference, ACSAC 2008, Hsinchu, Taiwan, Aug. 4–6, 2008.

[26] ImageJ. http://rsb.info.nih.gov/ij/.

[27] MPICH. http://www-unix.mcs.anl.gov/mpi/mpich1/.

[28] Network Weather Service. http://nws.cs.ucsb.edu/ewiki/.

[29] C. Baru, R. Moore, A. Rajasekar, M. Wan, The SDSC Storage Resource Broker, in: Proceedings of CASCON'98, 1998.

[30] JDK. http://java.sun.com/javase/.

[31] Apache Ant, http://ant.apache.org/.

[32] XML-Parser. http://www.w3schools.com/Xml/.

[33] xinetd. http://www.xinetd.org/.

[34] Chao-Tung Yang, I-Hsien Yang, Shih-Yu Wang, Ching-Hsien Hsu, Kuan-Ching Li, A recursively-adjusting co-allocation scheme with a cyber-transformer in data grids, Future Generation Computer Systems (ISSN: 0167-739X) 25 (7) (2009) 695–703.

[35] S. Vazhkudai, S. Tuecke, I. Foster, Replica selection in the globus data grid, in: Proceedings of the 1st International Symposium on Cluster Computing and the Grid, CCGRID 2001, May 2001, pp. 106–113.

[36] Chao-Tung Yang, Ming-Feng Yang, Wen-Chung Chiang, Implementation of a cyber transformer for parallel download in co-allocation data grid environments, in: Proceedings of the 7th International Conference on Grid and Cooperative Computing (GCC2008) and Second EchoGRID Conference, 24–26 October 2008, Shenzhen, Guangdong, China, pp. 242–253.

[37] X. Zhang, J. Freschl, J. Schopf, A performance study of monitoring and information services for distributed systems, in: Proceedings of 12th IEEE International Symposium on High Performance Distributed Computing, HPDC-12 '03, August 2003, pp. 270–282.

[38] C.T. Yang, C.H. Chen, K.C. Li, C.H. Hsu, Performance Analysis of Applying Replica Selection Technology for Data Grid Environments, PaCT 2005, in: Lecture Notes in Computer Science, vol. 3603, Springer-Verlag, 2005, pp. 278–287.

Chao-Tung Yang is a Professor of Computer Science at Tunghai University in Taiwan. He received a B.Sc. Degree in Computer Science from Tunghai University, Taichung, Taiwan, in 1990, and the M.Sc. Degree in Computer Science from National Chiao Tung University, Hsinchu, Taiwan, in 1992. He received the Ph.D. Degree in Computer Science from National Chiao Tung University in July 1996. He won the 1996 Acer Dragon Award for an outstanding Ph.D. dissertation. He has worked as an Associate Researcher for ground operations in the ROCSAT Ground System Section (RGS) of the National Space Program Office (NSPO) in Hsinchu Science-based Industrial Park since 1996. In August 2001, he joined the Faculty of the Department of Computer Science at Tunghai University. He got the Excellent Research Award from Tunghai University in 2007. In 2007 and 2008, he got the Golden Penguin Award by Industrial Development Bureau, Ministry of Economic Affairs, Taiwan. His research has been sponsored by Taiwan agencies of National Science Council (NSC) and Ministry of Education. He is serving in a number of journal editorial boards, including *International Journal of Communication Systems* (IJCS), *International Journal of Next-Generation Computing* (IJNGC), *Journal of Convergence Information Technology* (JCIT), *International Journal of Hybrid Information Technology* (IJHIT), *International Journal of Grid and High Performance Computing* (IJGHPC), and "Grid Computing, Applications and Technology" Special Issue of *Journal of Supercomputing*, "*Grid and Cloud Computing*" Special Issue of *International Journal of Ad Hoc and Ubiquitous Computing* (IJAHUC). His present research interests are in cloud, grid and cluster computing, parallel and multicore computing, and Web-based applications. He is a member of the IEEE Computer Society and ACM.

Chiu-Hsiung Chen received a B.S. Degree in the Department of Information Management from Overseas Chinese Institute of Technology in 2003. He received M.S. Degree in the Department of Computer Science from Tunghai University in July 2009. He is a software engineer at Tungs' Taichung MetroHarbor Hospital. His research interests include grid computing, cluster computing and PACS.

Ming-Feng Yang received a B.S. Degree in Hsiuping Institute of Technology in 2004. He received M.S. Degree in the Department of Computer Science from Tunghai University in July 2009. He is a software engineer at Hsiuping Institute of Technology. His research interests include grid computing, cluster computing and internet-based applications.