Distributed PACS using Network Shared File System

Tomoyuki HIROYASU,[†1] Yoshiyuki MINAMITANI,[†2]
Masato YOSHIMI[†3] and Mitsunori MIKI [†3]

In this paper, a distributed PACS (Picture Archiving and Communication Systems) using a network shared file system is proposed. Several distributed hospitals and sites use PACS to manage medical images. For these distributed sites to work together, it is necessary to construct integrated PACS for these distributed files. In the proposed system, the DICOM (Digital Imaging and Communications in Medicine) metadata and the DICOM image data are registered separately on the server. In the proposed system, several operations can be performed using only the metadata. For example, images can be searched using only the information stored in the metadata, and therefore the client can find the target image without downloading the whole DICOM data. DICOM image size is usually huge. Therefore, if a file can be found or transformed without downloading the whole data, the file operation can be performed very quickly. To implement the proposed system, the open source Gfarm network shared file system, which treats metadata as XML data, was used. This paper describes construction of the proposed system, and discusses experimental comparison of Gfarm and NFS with regard to transfer rates.

1. Introduction

Several types of medical information technology have recently been introduced into hospitals and medical research centers. These systems store and utilize many types and huge amounts of medical information, typically medical image data from MRI, NIRS, CT, etc. The management system used for medical images is the picture archiving and communication system (PACS). PACS stores, browses and manages medical images sent from medical devices such as MRI, NIRS, CT, etc. PACS was designed to use medical images stored at each hospital or facility. However, there has been a recent increase in collaborations between hospitals. When several hospitals collaborate in this way, these medical images must be integrated. One of the easiest ways to integrate distributed medical images or distributed PACS is to integrate all of the medical images onto a central server. However, the amount of medical image information is expected to increase markedly with the future development of medical equipment and systems. In addition to the file size of medical images, the number of such images is increasing on a daily basis. Based on current trends, it is estimated that over one billion diagnostic imaging procedures will be performed in the USA in 2014, which will generate about 100 petabytes of data[1)2)]. Therefore, there are limitations to having a central imaging server for distributed PACS integration. To solve this problem, we focused on distributed PACS that treats and utilizes distributed medical images as one data.

Medical image data standards, such as DICOM (Digital Imaging and communications in Medicine), are very important. DICOM consists of metadata and actual images. The metadata includes various types of medical information, such as inspection items, patient information, etc.

Here, we propose a novel distributed PACS in which only the DICOM metadata are stored on a central server. Actual images are saved on the distributed site. For example, to find a specific image, the user accesses the information stored in the metadata. Thus, it is inefficient to access the entire medical image when user retrieves only the DICOM metadata. Using this method, a high-speed extraction result can be obtained with retrieval of only the metadata.

To implement the proposed system, Gfarm was utilized as a network shared file system[3)]. Gfarm stores the metadata of the file as XML data. Therefore, the proposed system can be constructed easily with Gfarm. In the experimental system, the DICOM metadata are stored in Gfarm XML metadata. DICOM is distributed and stored on the storage node. Moreover, DICOM can be retrieved using the XML metadata. A flexible security policy is necessary when DICOM data are shared. DICOM includes information that can specify an individual, such as the patient's name, address, etc. This access to individual information is also controlled by managing the metadata. In the experimental system, individual information is deleted when the medical images are shared.

†1 Faculty of Life and Medical Sciences, Doshisha University
†2 Graduate School of Engineering, Doshisha University
†3 Department of Science and Engineering, Doshisha University

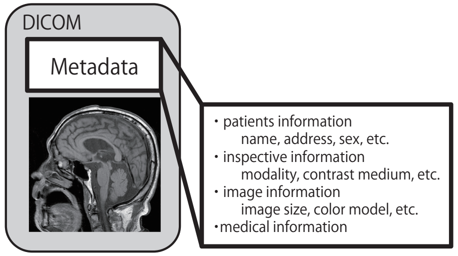

Fig. 1 Outline of DICOM

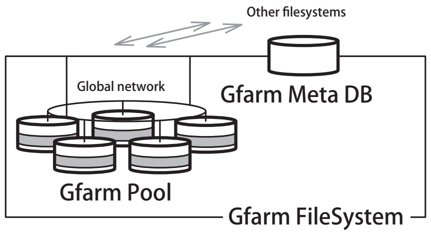

Fig. 2 Gfarm architecture[5)]

2. Background

2.1 DICOM

DICOM is a typical medical image standard and a network communications protocol for use with medical image diagnosis equipment, such as MRI and CT defined by the ACR (American College of Radiology) and NEMA (National Electrical Manufacturers Association). DICOM is also an international standard (ISO TC-215). DICOM is a container format that can include various types of data, such as image data, and it consists of actual data and information metadata[4)]. Fig.1 shows an outline of DICOM.

Various types of information, such as image data size and color model, as well as patient information such as name, address, date of birth and sex are stored in the metadata. The DICOM standard defines more than 3000 terms. When medical images are sorted or found, the information in the metadata is used. However, because information than can specify individual patients is included in the metadata, this raises privacy issues in information sharing.

2.2 PACS

PACS is now considered the typical standard in the radiology community and is widely implemented even on very large scales[6)7)]. PACS have been introduced at a number of hospitals and are used to manage medical images with electronic patient records[8)9)]. However, it has recently become necessary to construct PACS capable of enabling cooperation with outside hospitals. One solution for integrating distributed PACS is to use cloud computing systems, which promise lower costs, high scalability, and high availability[10)11)].

2.3 Distributed file system and Gfarm

2.3.1 Distributed file system

GFS (Google File System) is a distributed file system developed by Google[12)]. On GFS, a large amount of data of PC cluster can be processed. MapReduce is a widely used process to treat such large amounts of data on GFS.

HDFS (Hadoop Distributed File System) is an open source file system which refers GFS[13)]. GFS and HDFS are designed to treat data of the order of several million MB, and these systems are not suitable for accessing large amounts of small data. At the same time, methods for these file systems are not versatile because files are operated through the API that is different from standard file systems. On the other hand, Gfarm supports the POSIX standard API and enables decentralized parallel processing. Gfarm shows equivalent read performance and about 30% higher write performance compared to HDFS[14)].

2.3.2 Gfarm

Gfarm is a global distributed file system used to share data and to support distributed data-intensive computing[3)]. Gfarm federates local file systems of compute nodes connected on a network, and can be used at various scales, such as LAN, PC clusters, and large-area clusters. Gfarm is provided under an open source license and may be used as a substitude for Network File System (NFS). Gfarm consists of three types of node: metadata servers, which manage the preservation of location information of each file; I/O servers where the main body of data is stored; and clients that access the files. The metadata servers manage file system data of a virtual directory tree and the locations of the actual files, etc., as metadata. First, the client inquires about the position of the I/O server, which stores a file of the metadata server when it access the file which is stored in Gfarm. The client then accesses the I/O server where the file is stored. Fig. 2 shows the Gfarm architecture.

The Gfarm metadata server not only to manages preservation of the location information of files but also treats metadata as XML data. This mechanism, called XML extended attributes, specifies the related XML file of each saved file, and enables retrieval of the file by XPath. As retrieval becomes possible without directly accessing the file, Gfarm enables high-speed information retrieval in file sharing of the large area.

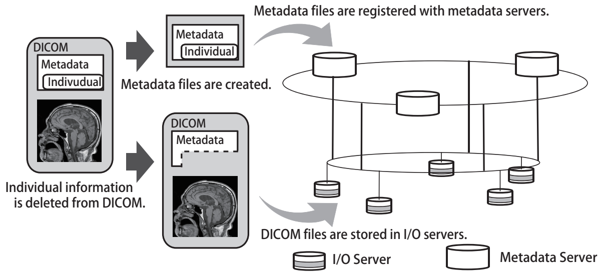

Fig. 3 Outline of proposed system

3. Proposed Distributed PACS

In this chapter, the proposed distributed PACS, which treats metadata and real data of DICOM separately, is described.

3.1 Outline of proposed system

Here, we assume that several PACS utilized at multiple sights are integrated. As described in Chapter 1, the proposed system involves integration of several PACS with no central PACS. To share large-sized DICOM file over the large-area environment, it is necessary to construct a system to integrate the PACS used in each location. In integrated PACS, it is inefficient to access the whole medical image data when retrieving the DICOM metadata. Thus, it is necessary to separate metadata information from DICOM data.

In the proposed system, DICOM data are stored at each site and DICOM metadata information are stored separately on the metadata server. When metadata are stored, if it is necessary to separate and manage individual information for privacy protection. The access authority to individual information is set according to the authority of the client. It is necessary to construct a system where only clients that have access authority to individual information can retrieve the whole DICOM data.

To achieve this goal, it is necessary to prepare a distributed file system that can be used for metadata, and the metadata information should be managed. The open source network shared file system Gfarm is used for the experimental system discussed here as discussed in the next chapter. Fig. 3 shows the outline of the proposed system.

As shown in Fig. 3, XML are made from the metadata of DICOM and XML metadata is specified for the XML extended attribute of file system. XML is stored on the metadata server. Therefore, the XML metadata alone can be used without the need to access the large DICOM file image information. High-speed retrieval can be achieved because the file system itself retrieves the file. The proposed system stores the DICOM files and allows sharing over a large area. On the other hand, patient information is separated, and the associations with stored DICOM files are preserved by the I/O servers. Account information of the client is set to the DICOM file and access to individual information is controlled appropriately. The following three DICOM file operations are performed:

- Saving DICOM files
  When users store the DICOM file through the client system, the client of the proposed system stores DICOM metadata as XML on the metadata server. DICOM data containing no individual information are stored on the I/O Server. If necessary, individual information is separated from DICOM data.
- Reading DICOM files
  When the user accesses the DICOM file, the proposed system reads individual information from the metadata server according to the authority of the client. The proposed system combines the DICOM file obtained from the I/O server with individual information on the client side.
- Retrieval DICOM metadata
  The retrieval is performed with the metadata server for the DICOM metadata stored in file system. Moreover, it is possible to use only the DICOM metadata by downloading the XML used for retrieval.

3.2 Merits of the proposed system

Here, the following four advantages of the proposed system are described.

- Promotion of telemedical care
  There is a great deal of interest in telemedicine services with diagnosis based on medical images. The proposed system can facilitate such telemedicine services. The patient may ask a doctor to make a diagnosis using medical images taken at another hospital. In this case, the patient's individual information is not necessary for diagnostic imaging. In the proposed system, DICOM data can be stored without individual information. Thus, the pro-

posed system may contribute to the promotion of telemedicine care because it can be used to acquire only the data necessary for diagnostic imaging.

- Developing Medical treatment Cloud
  Patients can receive different examinations using the same medical images obtained at several hospitals if each hospital participates in this system. When medical images are used for the examination, it is currently necessary to obtain permission from the hospital where they were originally taken. The proposed system will allow the development of a medical treatment cloud system, which will allow patients to easily obtain a second opinion.
- Academic use for medical images
  Statistical data of medical images is very important for academic use. Patients' individual information is not necessary for academic use of medical images.

4. Implementation of the Proposed System

The proposed system was implemented using Gfarm and an experimental system was developed. Gfarm stores XML data as extended XML attributes of the saved file. These mechanisms are suitable for the proposed distributed file system. Gfarm is also distributed under an open source license and is therefore easy to develop. This chapter describes the implemented system.

4.1 Outline of the implemented system

The actual production system will be located in several remote locations. However, the implemented experimental system was constructed on a virtual server using the Hyper-V function of Windows HPC Server 2008. We constructed an experimental system consisting of one metadata server, two I/O servers, and one client on a virtual node. The next chapter discusses a number of experiments performed using this system. To avoid the influence of communication time and the overhead of each file system is focused, the processing overhead of each file system, the assessment experiment was performed on this virtual system. Tables 1 and 2 show the specifications of the server and of each virtual node, respectively. Fig. 4 shows the flow of the implemented system.

As shown in the figure, the client divides the metadata and actual images, and these data are stored on Gfarm. When images are searched or derived, the client can obtain the metadata and actual data from Gfarm integrated as DICOM data. The following sections describe Gfarm and how to use the commnads.

Table 1 Specifications of the server

CPU	Quad-Core AMD Opteron 2.3GHz × 2
Memory	DDR2 667 MHz 8GB
OS	WindowsServer HPC Edition 2007

Table 2 Specifications of each virtual node

Number of processors	1
Memory	512MB
OS	CentOS 5.5
Kernel	2.6.18-194.el5
gfarm	2.4.0
gfarm2fs	1.2.1

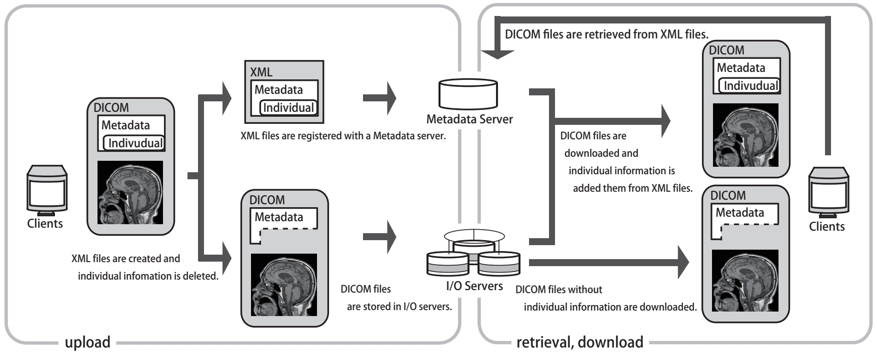

Fig. 4 Flow of implemented system

4.2 Commands for the experimental system

We prepared the commands for downloading and uploading of DICOM data. As Gfarm can search a file from XML data, the existing Gfarm search command is used for file retrieval. These commands were prepared on the client system and implemented in Python.

- Uploading DICOM file
  We prepared a "dgf_up" command to upload DICOM files to the experimental system. When the client uploads DICOM ifles to the experimental system using the "dgf_up" command, the experimental system produces XML from the metadata of the specified DICOM files. Individual information is then deleted from the DICOM files. The experimental system stores DICOM image data separately from individual information on the I/O server. The experimental system adds the XML extended attributes using the Gfarm command "gfxattr" for DICOM files stored on the I/O server. The XML

extended attributes are managed on the metadata server.

- Downloading DICOM file
  We prepared the "dgf_down" command to download DICOM files from the experimental system. When the client acquires DICOM files including individual information using the "dgf_down" command, the client downloads the DICOM files separately from the individual information from the I/O server and XML from the metadata server. The client acquires DICOM files including individual information by reading the individual information from the XML and adding it to the DICOM files on the client side. Moreover, clients that do not have access authority to the individual information can only acquire the DICOM files without the individual information.
- Retrieving DICOM file
  The client retrieves DICOM files with XPath using the Gfarm command "gffindxmlattr". Clients that do not have access authority to the individual information can also retrieve DICOM files in the same way using the metadata without the individual information.

5. Experimental Assessment

As described in this chapter, Gfarm was confirmed to show sufficient performance as a file system for medical images and to be useful for separation and retrieval of metadata in the implemented system.

Gfarm was compared with NFS[15)], which is a well-known small-scale file sharing system. NFS is not a file system which is designed for large-area environments. However, NFS was compared from the viewpoint of the ease of file access and high-speed performance. In the experimental system, three virtual machines were constructed as an experimental environment for Gfarm. On the other hand, two virtual machines were constructed as an experimental environment for NFS. In NFS, one virtual machine stored the DICOM files minus the individual information and the other stored XML files with the individual information.

Table 3 shows the sizes of DICOM and XML files used in the experiment. We measured and evaluated the time required to write to the system and delete individual data for thousands of DICOM files and XML files. We measured the time required for processing five times using the "time" command.

Table 3 DICOM and XML file sizes

File	Size	Number	Sum size
DICOM	135KB	1000	132MB
XML	16KB	1000	16MB

Table 4 Results for saving DICOM and XML files

Process	Gfarm	NFS
real	5m22.507s	5m48.150s
user	0m20.957s	0m05.509s
system	0m56.123s	1m53.721s

Table 5 Results for saving XML files

Process	Gfarm	NFS
real	2m29.210s	1m23.912s
user	0m14.642s	0m00.098s
system	0m26.630s	0m13.922s

Table 6 Results for saving DICOM files

Process	Gfarm	NFS
real	2m47.496s	4m03.428s
user	0m00.156s	0m00.149s
system	0m02.200s	1m49.631s

5.1 DICOM and XML file storage

In this section, we compare the processing speeds when DICOM and XML files are stored at the same time on Gfarm and NFS. NFS stored DICOM files on one of two servers, with XML files stored on the other server. Gfarm stored DICOM files on two I/O servers with the XML extended attributes stored on one metadata server. Table IV shows the measurement results for saving 1000 DICOM files and 1000 XML files. In this table, "real" is the total time, "user" is the user CPU time, and "system" is the system CPU time.

5.2 XML file storage

In this section, the processing speeds of storing only XML files are compared between Gfarm and NFS. NFS stored XML files on one server. Gfarm added the metadata to the DICOM data already stored on the I/O server. Table 5 shows the measurement results for saving 1000 XML files.

5.3 DICOM file storage

In this section, the processing speeds of storing only DICOM files between Gfarm and NFS were compared. Table 6 shows the measurement results for saving 1000 DICOM files.

5.4 XML creation

In this section, the access speed with creating XML from DICOM files was measured. Table 7 shows the results for creating XML from 1000 DICOM files.

5.5 Deleting individual information of dicom

Table 8 shows the access speeds when individual information was deleted from 1000 DICOM files.

Table 7 Results when creating XML from DICOM files

Process	Time
real	8m06.603s
user	7m34.788s
system	0m31.611s

Table 8 Results when deleting individual info from DICOM files

Process	Time
real	12m08.176s
user	11m29.293s
system	00m32.916s

6. Discussion

As shown in Table 4, the access time for writing is shorter for Gfarm than for NFS. This result suggested that Gfarm has smaller write overhead than NFS. It has also been reported that Gfarm has the same access time as HDFS. Therefore, the proposed system has high file access capability because it uses Gfarm.

As shown in Table 5, the Gfarm processing time was longer than that of NFS with regard to writing the XML files on each server. For Gfarm, there is a process for adding XML extended attributes. This process may take some time.

On the other hand, as shown in Table 6, the processing time of Gfarm is shorter than that of NFS. Moreover, it was confirmed that the system CPU time of Gfarm was smaller than that of NFS. These results showed that Gfarm has a mechanism to reduce the overhead.

Tables 7 and 8 show the operation times for creating and deleting DICOM data. The proposed system took more than 8 minutes to create 1000 XML files. It also took more than 12 minutes to delete patient information from XML data of 1000 DICOM files. These overhead times are too long and must be reduced in future. The proposed system was implemented using Python. The module involved in data operations will be rewritten in C to improve the speed.

7. Conclusions and Future Work

In this paper, a distributed PACS was proposed. In the proposed system, the metadata and the actual images of DICOM data are treated differently. For system implementation, the Gfarm network shared file system was used. Through comparative experiments between NFS and Gfarm, we confirmed that the proposed system inherited the characteristics of Gfarm. Thus, the proposed system may have high scalability of access speed and data size. We also confirmed the effectiveness and improved ease of retrieval access control as the proposed system only can access the DICOM metadata.

In future work, we will examine the scalability of large-size files stored in Gfarm. We will also improve the system for use in medical facilities. To achieve this goal, we will examine security and access control mechanisms.

References

1) Teng, C.-C., Mitchell, J., Walker, C., Swan, A., Davila, C., Howard, D. and Needham, T.: A Medical Image Archive Solution in the Cloud, *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on* (2010).
2) Frost &Sullivan: Prepare for disaster & tackle terabytes when evaluating medical image archiving (2008). http://www.frost.com.
3) Tatebe, O., Hiraga, K. and Soda, N.: Gfarm Grid File System, *New Generation Computing*, Vol.28, pp.257–275 (2010).
4) Herman Oosterwojk: *DICOM Basics Third Edition edition*, OTech Inc (2005).
5) Datafarm, G.: . http://datafarm.apgrid.org.
6) D. Bandon and C. Lovis and A. Geissbuhler and J.-P. Vallee: Enterprise-wide PACS: beyond radiology, an architecture to manage all medical images, *Academic Radiology 12 (2005)*, pp.1000–1009.
7) Huang HK: Enterprize PACS and image distribution, *Comput Med Imaging Graphics 2003*, Vol.27, pp.241–253.
8) Munch, H., Engelmann, U., Schroeter, A. and Meinzer, H.: The integration of medical images with the electronic patient record and their web-based distribution, *Acad Radiol 2004*.
9) Ratib, O., Ligier, Y., Bandon, D. and Valentino, D.: Update on digital image management and PACS, *Abdom Imaging 2000*, Vol.25, pp.333–340.
10) Zheng, R., Jin, H., Zhang, Q. and Chu, P.: Heterogeneous Medical Data Share and Integration on Grid, *Proceedings of 2008 International Conference on Biomedical Engineering and Informatics (BMEI' 08), IEEE Computer Society Press* (2008).
11) Erberich, S., Silverstein, J., Chervenak, A., Schuler, R., Nelson, M. and Kesselman, C.: Globus MEDICUS — federation of DICOM medical imaging devices into healthcare grids, *Stud Health Technol Inform*, Vol.126, pp.269–278.
12) GHEMAWAT S.: The Google File System, *Proceedings of 19th ACM Symposium on Operating Systems Principles (SOSP-19), 2003*, (online), DOI:10.1145/1165389.945450 (2003).
13) The Apache Software Foundation: Apache Hadoop. http://hadoop.apache.org/.
14) Mikami, S., Ohta, K. and Tatebe, O.: Data Intensive distributed computing using MapReduce on Gfarm file system, *Information Processing Society of Japan*, Vol. 2010, No.4 (2010).
15) Callaghan, B., Pawlowski and Staubach: NFS Version 3 Protocol Specification, *RFC 1813* (1995).