SmartWADO: an Extensible WADO Middleware for Regional Medical Image Sharing

Lijun Liu¹ · Li Liu² · Xiaodong Fu² · Qingsong Huang² · Yin Zhang³ · Qiaoyi Luo² · Xin Xiong¹

Published online: 27 March 2015 © Society for Imaging Informatics in Medicine 2015

Abstract Medical image sharing is an important problem in modern radiology, with wide applications in Internet and mobile devices. Some important features need to be added and optimized to medical image sharing. In this paper, we present an extensible Web Access to DICOM Persistent Objects (WADO) middleware based on image cache and real-time Web monitor technology for regional medical image sharing. We first develop the extension method of WADO standard and workflow of extended WADO service. Then, we design a medical image cache method to improve the performance of medical image on-demand transmission. Using the real-time monitor can discover the performance bottlenecks and optimized critical points. The experimental results show that the middleware effectively delivers medical images and reports to Web clients over the Internet, regardless of the platform used for access. It can be deployed in one hospital to provide WADO service to medical workers and also can be applied to regional picture archiving and communication systems (PACS) to transmit medical images and reports to Internet users in a way that is transparent to end-user applications.

Qingsong Huang kmustailab@hotmail.com

> Lijun Liu cloneiq@126.com

- ¹ Department of Biomedical Engineering, Kunming University of Science and Technology, Kunming 650500, China
- ² Department of Computer Science and Technology, Kunming University of Science and Technology, Room 403, Building Xinzi, No. 727, South Jingming Rd., Chenggong District, Kunming City, Yunnan Province, China
- ³ Information Centre, Kunming Municipal Hospital of Traditional Chinese Medicine, Kunming 650051, China

Keywords DICOM · WADO · Image cache · Web monitor · Regional medical image sharing

Introduction

Medical imaging technology has become the primary method of diagnosis [1]. Due to the large size and increasing information of medical images, storing and sharing the information are becoming increasingly important [1-3]. Cloud-based picture archiving and communication systems (PACS) are efficient for archiving medical images, allowing Web clients to access images and reports from anywhere, over wireless networks, regardless of the platform used for access [4]. With popularity of the Internet and mobile devices, remotely accessing and presenting medical images through the Web are important for modern radiology [5]. Digital Imaging and Communications in Medicine (DICOM) is widely used for reviewing images and performing primary diagnosis within radiology and other imaging departments, by means of PACS [6]. Web Access to DICOM Persistent Objects (WADO), which is another DICOM Internet extension, specifies Web-based service for accessing and presenting DICOM persistent objects such as images and reports from PACS [7]. Despite the fact that WADO has some disadvantages, it is widely applied to integration and sharing of medical images and is also included in other medical information exchange standards as a retrieve service. The IHE crossenterprise document sharing for imaging (XDS-I) and cross-community access for imaging (XCA-I) integration profiles include components common to the WADO standard [8], which are considered to be the most likely future standard for medical image exchange.

WADO service has become an important component of modern PACS to ensure that medical images will be easily accessible within Web-based applications through open standard. Many Web PACS or middleware based on WADO have been presented and developed to solve medical image sharing and accessing over the Web [5, 9, 11-13], which have provided alternative approaches. Paper [5] compares medical image access and presentation system (MIAPS) with eRAD PACS, MERGE eFilm, AGFA IMPAX, and DIPACS from "Retrieval methods," "Tile-based transmission," "Image prefetching," "System architecture," "Platform independent," "Mobile terminal support," and "Security and privacy protection." From Figure 20 in this paper, we can see that MIAPS provides two unique functions: multiple DICOM image retrieval methods and tile-based transmission of single-slice DICOM images [5]. But several important features need to be added and optimized to enhance the capabilities of WADO service: (1) absence of a mechanism to support multiple PACS simultaneously, (2) absence of performance monitor and analysis method to the key points of WADO service, (3) inability to provide a real-time dynamic performance monitor screen, and (4) a need for the image cache performance to be further optimized [5, 9].

To overcome these shortages, this paper presents an extensible WADO service framework (SmartWADO) based on image cache and real-time Web monitor technology, which is specially designed as a cost-efficient service middleware allowing for accessing, sharing, and visualizing DICOM persistent objects from any-where and anytime over the Internet. The proposed system includes three highlights: (1) an extension method base on a routing table to support multiple PACS simultaneously, (2) a Web-based real-time monitor method to discover the performance bottlenecks and optimized critical points, and (3) a medical image cache method base on key-value to improve the performance of medical image on-demand transmission.

The rest of this paper is organized as follows: "Related Works" provides a brief overview of PACS and WADO service. Some of their shortcomings are pointed out. "System Description and Computational Methods" details the architecture, workflow, relative computational methods, and implementation. First, we provide an overview of system architecture. Second, the extension method of the WADO standard and the workflow of extended WADO service are depicted. Third, the on-demand medical image cache structure and method are described. Fourth, a memorybased real-time monitor method and Web-based interface are shown. Finally, we report the implementation of SmartWADO. "System Evaluation" demonstrates the evaluation of functionality and performance. Salient points are given in "Discussion." Finally, we conclude our work and present the future work in "Conclusion."

Related Works

In modern medical systems, huge amount of text, words, images, and videos are produced and stored in PACS, and the medical domain strongly feels the need to share information and to make it easily accessible to other physicians [10]. DICOM WADO and WADO via Web services are two important stepping stones to ensure that medical images will be easily accessible within these Web-based applications through open standard [8]. In this context, there are many research works on the WADO standard and WADO via Web services, which are mainly polarized into three approaches:

- (1) WADO service is integrated into Web-based PACS to provide service for a specific PACS. Several Webbased PACS architectures and systems based on WADO are presented and implemented to request and transmit images and reports from archives to Internet users [11–13]. The authors of [14] present a fast flash-based Web PACS viewer for displaying large DICOM images, and WADO service is exploited with the goal of retrieving small image tiles to compose an image on the client side. Paper [15] explores the current state of the art in bringing radiological images to modern mobile phones and draws a conclusion that mobile Web browsers can be used to read DICOM images from PACS archive with WADO, and mobile Web-based radiological applications can satisfy basic and advanced requirements of radiologists in everyday medical image analysis.
- (2) Extending the WADO standard to provide more useful functions. There have been published efforts to propose extensions to WADO such as the Web access to DICOM archives (WADA) service [16]. The WADA service introduces a query and reporting extension to the WADO standard. It can access all levels of DICOM hierarchy and process requests for medical reports related to series and studies.
- (3) Integrating and sharing medical images using the WADO standard through the Internet. In [17], the authors described how XDS-I can be used in conjunction with WADO and DICOM transfer syntax for JPEG 2000 Interactive Protocol (JPIP) to enable streaming of medical images directly from EHR-connected imaging sources to image processing workstations. MIAPS is a Web-based system developed in [5] for medical images sharing and teleradiology through the Internet. A RESTful image gateway for multiple medical image repositories is developed in [9] to provide transparent bridging between PACS-DICOM and Web universes.

These studies gave a big boost to the development of the WADO standard, which offers alternative solutions for

integrating, sharing, accessing, and visualizing images and medical reports from anywhere and anytime over the Internet. However, it is desirable to provide access to more than a single PACS [9]; WADO service only queries or retrieves DICOM persistent objects from one PACS simultaneously because of missing remote application entity (AE) information of remote PACS in a WADO request. The current implementation of WADO service is difficult to position the performance bottlenecks and optimize key points immediately, because they do not provide performance monitor and analysis method to the key points of a WADO response. The real-time performance of current systems cannot be found directly because the Webbased real-time monitor and visualization display are not supported. The cache performance needs to be further optimized because image caching structures and algorithms are relatively simple [5, 9].

System Description and Computational Methods

System Architecture

The SmartWADO is implemented as a service middleware and the architecture is depicted in Fig. 1. The Web client, usually a Web browser, connects to the SmartWADO via

Fig. 1 Architecture of SmartWADO

wired or wireless (Wi-Fi, 3G, 4G) networks with HTTP/ HTTPs protocol over the Internet to access images and medical reports. The SmartWADO connects to PACS servers with DICOM protocol or WADO to query or retrieve DICOM persistent objects, which will be parsed and converted to a specific format according to the WADO request. The whole procedure can be monitored and displayed in the Web page, especially as regards the performance of key points.

As shown in Fig. 1, the SmartWADO consists of WADO service, cache service, and monitor service. As a core component of SmartWADO, the WADO service provides a Web interface to access medical images and reports which may come from more than one PACS. The cache service stores recently requested images to accelerate image transmission. The monitor service records the response time of every key point, and the results will be displayed via a Web page with dynamic curve.

Extension for the WADO Standard

We present a simple method to extend the WADO standard. As already mentioned, the WADO service only retrieves DICOM persistent objects from one PACS simultaneously because of missing AE information of remote PACS in the WADO request. However, it is desirable to provide access to



more than a single PACS [9], especially in the environment of regional medical image sharing. The extension method is that we add a remote PACS routing table on the server side.

In SmartWADO, we establish a mapping between StudyUID and remoteAE. SmartWADO gets the value of StudyUID from the WADO request and retrieves DICOM persistent objects from the corresponding remoteAE. The details of StudyUID and remoteAE are stored in the routing table on the server side, where r_{excelsy} (Algorithm 1997) r_{excelsy}

Figure 2 shows the workflow of extension. First, SmartWADO gets the value of StudyUID from the WADO request sent by the Web client. Second, SmartWADO gets the details of a corresponding remoteAE from the routing table and retrieves DICOM persistent objects from this remoteAE. If no remoteAE is found, SmartWADO retrieves DICOM persistent objects from default PACS.

Extended WADO Service

wadoPort, and aeStatus.

The Web client sends query parameters to the WADO service through the HTTP GET method. The WADO service first retrieves DICOM persistent objects from the remote PACS according to remoteAE. Then, the object is converted to an image or report (a recently requested image is obtained from cache). Finally, the WADO service responds to the Web client by seeding one or more objects in a proper Multipurpose Internet Mail Extension (MIME) type. Figure 3 depicts the

Fig. 2 Workflow of extension for WADO

flowchart of the WADO service. The detailed procedure of the extended WADO service is as follows:

- Step1 The Web client sends a WADO request to SmartWADO.
- Step 2 The WADO service determines the content type. If the content type is not an image type (image/jpeg, image/png), then go to step 4.
- Step 3 The cache service retrieves the requested image from cache. If the image is null, then go to step 4; otherwise, go to step 9.
- Step 4 The WADO service determines the retrieve type of the request. If retrieve type is WADO, then go to step 8.
- Step 5 The WADO service downloads the DICOM persistent object from the remote PACS via C-MOVE or C-GET.
- Step 6 The WADO service parses and coverts the DICOM file to a specific format according to the content type. If the content type is not an image (image/jpeg, image/png) then go to step 9.
- Step 7 The cache service puts the image to cache; then, go to step 9.
- Step 8 The WADO service sends the WADO request to the remote PACS and gets a response. If the content type is an image type (image/jpeg, image/png), then go to step 7.
- Step 9 The WADO service returns the image or report to the Web client.

Image Cache Service

Cache service is an important component to accelerate image transmission, which stores recently requested images to the cache server. For the next request, the cached image is directly delivered to the Web client





Fig. 3 Flowchart of the extended WADO service

without retrieving from the remote PACS and decoding from the DICOM persistent object.

Figure 4 shows the cache structure. It consists of a cache index and a cache file; the former is used to store the cache index to quickly locate the cache file, and the latter is used to store cache files. Cache indexes are stored in memory and cache files are stored on disk. The cache index consists of a key and a value, which is implemented as a hash table and stored in memory. Each index in the cache is denoted by a unique key, where $Key = \left\{ \frac{studyUID \cup seriesUID \cup objectUID \cup rows \cup columns \cup region \cup windowWidth }{windowCenter \cup imageQuality \cup frameNumber \cup contentType} \right\}$.

The parameters in the key (*studyUID*, *seriesUID*, *objectUID*, *rows*, *columns*, *region*, *windowWidth*, *windowCenter*,



Fig. 4 Cache structure in cache service

imageQuality, contentType, and *frameNumber*) are defined by the WADO request. Each index in the cache corresponds to a unique value used to store the information of the cache file, where Value={filePath, imgSize, accCount, lastAccTime}.

Value update follows the LRFU (Least Recently/ Frequently Used) policy: When the storage space of cache indexes or cache files exceeds a predetermined threshold, the least recently used or least frequently used cache file is firstly replaced by a new image file. The LRFU policy uses a value, called CRF, to quantify the likelihood that a block will be referenced in the near future. Each reference to a block in the past contributes to this value, and a reference's contribution is determined by a *weighing function* F(x), where x is the time span from the reference in the past to the current time. The theoretical model for the algorithm is as the follows [18]:

$$C_{t_{\text{base}}}(b) = \sum_{i=1}^{k} F(t_{\text{base}} - t_{b_i})$$
(1)

where the CRF value of a block *b* at time t_{base} is denoted by $C_{t_{\text{base}}}(b)$, F(x) is a weighing function, and $\{t_{b_1}, t_{b_2}, t_{b_3}, \dots, t_{b_k}\}$ are the reference times of block *b* and $t_{b_1} < t_{b_2} \dots < t_{b_k} < t_{\text{base}}$.

In fact, $F(x) = (1/p)^{\lambda x}$ is used as an adequate weighting function, where x is the difference between the current time and the time of a reference in the past,

 $p \ge 2$, and λ ranges from 0 to 1. The theoretical model is simplified as follows:

$$C_{t_{b_k}}(b) = F(0) + F(\delta)C_{t_{b_{k-1}}}(b)$$
(2)

where $\delta = t_{b_k} - t_{b_{k-1}}$.

As for implementation, we realize four methods to manage cache service. The method called searchIndex searches index from the hash table and quickly locates the cache file. The getFromCache method gets the image file from the cache server according to the index. The writeToCache puts recently requested images to the cache server, including the index and image file. The least recently/frequently used file is replaced with the new requested image file using the updateCache method when the storage space of cache indexes or cache files exceeds the predetermined threshold.

Real-Time Monitor Service

We propose a memory-based real-time monitor method and implement a Web-based interface for displaying the performance of SmartWADO. The former is used to monitor response time of key points, and the latter displays monitor data via a Web page with dynamic curve. We can timely detect performance bottlenecks and optimized critical points of the WADO service. Meanwhile, monitor data will be written to a log file.

There are four performance key points in the response of the WADO service: content type validation, DICOM file downloading, DICOM object parsing and conversion, and cache access. The response time and file size (cache size) of these key points are monitored and recorded in a memory object named monitorObject, where

 $monitorObject = \left\{ reqCount, \ \widetilde{fileSize}, \ validateTime, \ loadTime, \ parseTime \\ download-dicom-file-from-PACS, \\ \widetilde{cacheCount, \ cacheSize} \\ \end{array}, \\ \widetilde{cacheCount, \ cacheSize} \\ \end{array} \right\}$

The numerical values of the attributes in monitorObject (*reqCount*, *validateTime*, *loadTime*, *parseTime*, *fileSize*, *cacheCount*, *cacheTime*, *cacheSize*) are automatically calculated by summing.

How to display these values is another problem. The Web interface displays the statistical value in each interval. We regularly get monitor data from the memory object and calculate the value of each monitored attribute.

Let *n* is the number of requests for time period *t*, *m* is the cache count in *t*, p_{time_1} presents the *i*th request response time of the *j*th critical point, f_{size_i} is the file size of the *i*th request, c_{time_k} is the cache response time of the *k*th request, and c_{size_k} is the cache image size of the *k*th request. The calculations are as follows:

Average size of file and cache transfer:

transSize =
$$\left\{ \left(\sum_{i=1}^{n-m} f_{\text{size}_i} \right) / (n-m), \left(\sum_{i=1}^{m} c_{\text{size}_j} \right) / m \right\}$$
(3)

Average rate of file and cache transfer:

transRate =
$$\left\{ \left(\sum_{i=1}^{n-m} f_{\text{size}_i} \right) / t, \left(\sum_{j=1}^{m} c_{\text{size}_j} \right) / t \right\}$$
 (4)

Average response time of each point:

$$\left\{ \left(\sum_{i=1,j=1}^{n-m} p_{\text{time}_{ij}} \right) / (n-m), \left(\sum_{i=1,j=3}^{n-m} p_{\text{time}_{ij}} \right) / (n-m), \left(\sum_{j=1}^{m} c_{\text{time}_{j}} \right) / m \right\}$$
(5)

Cache hit rate:

$$cacheRate = \frac{m}{n}$$
(6)

Average overall response time:

overallTime =
$$\left(\sum_{i=1}^{n-m}\sum_{j=1}^{3}p_{\text{time}_{ij}} + \sum_{j=1}^{m}c_{\text{time}_j}\right)/n$$
 (7)

As for implementation, we use the open-source package HighChart [19] to dynamic display monitor data via the Web interface. HighChart is a charting library written in pure JavaScript, offering an easy way of adding interactive charts to the Web application. First, a program defined in the Web page gets monitor data from memory in each interval. Second, monitor data is calculated using the above methods (Eqs. (3–7)). Third, frontend dynamic updates the HighChart curve in the Web page using these data. In addition, server memory usage, CPU usage, and memory usage of the Web container are also monitored, and the high-usage metrics will be alarmed.

Implementation of SmartWADO

We implemented the SmartWADO mainly with Java technologies for portability and interoperability reasons. DICOM communication is provided through the dcm4che2 toolkit, which is a robust implementation of DICOM standard based on Java [20]. We use opensource PACS, namely dcm4chee, which is an implementation for image manager archive based on dcm4che2 library, to provide DICOM archives for storage and retrieval of DICOM images [20]. The SmartWADO is prepackaged and deployed within the Tomcat application server.

The function of SmartWADO consists of configuration, cache management, system monitor, and service engine. The Web interface of core configuration is shown in Fig. 5. After SmartWADO start-up, the detailed configuration of the remote PACS registered to the routing table via the Web interface is given in Fig. 5a. DICOM listener service must be started when the retrieve type is C-MOVE; AE and port are configured in Fig. 5b. Cache management provides some functions relative to cache service. Some parameters which are used to guide cache service are maintained

(b)

in Fig. 5c. All of the configuration information is performed through an XML file named server-config.xml.

After these configurations, SmartWADO is started up and provides WADO service to Web clients. Service engines are some programs that run in the background, such as WADO service, cache service, monitor service, DICOM conversion service, and DICOM listener service. The monitor service will monitor all WADO requests and display monitor data via a Web page with dynamic curve. Figure 6 illustrates six different dynamic monitor curves.

From left to right, up to down, Fig. 6a shows the usage of CPU, memory, and Tomcat memory. The high-usage metrics will be alarmed. The average transfer size and rate in each interval are depicted in Fig. 6b, c; the value of the former is calculated by Eq. (3), and that of the latter is obtained by Eq. (4). The average response time of key points is displayed in Fig. 6d, and

🚴 SmartW	ADO-An Exten	ed 👌 🗙 🔍 🚴 Oviyam 2	2.0	×							~
+ → C	222.1	. 97.201.191 :8088	/SmartWA[00/							Ξ
رِهُي s	SmartW	ADO An I	Extensible W	ADO mi	ddleware	based on	image cache f	or regional	Medical I	mage sha	iring
Home	Configu	ration Cache Ma	anagement	Moni	toring St	ation H	lelp				
PACS Ser	rvers DIC	OM Listener	nguages								
+ Add	@ Edit	🗑 Delete 🔒 V	/erify 🛹 l	Default							
show 1	l0 🔻 entrie	s					:	Search:			
AeStatus	Remote ID	Description	Remote AE	Host Na	ne	Port	RetrieveType	WADO Context	WADO Port	Update	
	dcm4chee1	Open PACS System	DCM4CHEE	localhost		11112	C-MOVE				
	dcm4chee2	Open PACS system	DCM4CHEE	222, 197	201.130	11112	C-GET				
	dcm4chee3	Open PACS system	DCM4CHEE	222, 197	201.130	11112	WADO	wado	8080		
	pacaone	KM-chinese Medicine Hosptital	PacsOne	222, 56, 37, 21		18088	WADO	wado	8080		
	R-PACS	R-PACS	R-PACS	222, 197	201.128	10088	C-MOVE				
	Copyrig	ht © 2014 SmartW/	ADO. Kunmi	ng Univ	ersity of S	Science ar	nd Technology.	cloneiq@'	126.com		
PACS Servers DICOM Listener Languages					Cache	Configer	Cache Status	Cache Files	6 Cact	ie Logs	
DICOM	4 Local Liste	ner(C-Move)			Cach	e Configer					
Remote AE SmartWADO					Cach	Cache Type Only Image Cache					
Port 1085					Cach	Cache Root Dir /CacheData (*)					
ı⊐ Update ► Start 📣 Verify 🔹 Stop					Cache Replace Least Recently/Frequently Used(LRFU) Replace Circle 0 (0:auto,*)					~	

Fig. 5 Web interface of core configuration. a Configuration of the remote PACS routing table. b Configuration of the DICOM listener. c Configuration of the cache parameter

🖻 Update

(c)



Fig. 6 a-f Web interface of the real-time monitor

the value is calculated by Eq. (5). Figure 6e displays cache hit rate in each interval, and the value is

calculated by Eq. (6). The overall response time is shown in Fig. 6f, and the value is calculated by Eq. (7).



a Smart phone. b iPad. c Web browser

Fig. 7 SmartWADO accessed via devices with different resolutions.

(a)

(c)



System Evaluation

Functionality Evaluation

To demonstrate functionality and performance, SmartWADO is installed in the Hospital of Traditional Chinese Medicine and Intelligent System Laboratory in our university. There are two servers, five PCs, one iPad, and one smart phone in our network. One server is used to deploy SmartWADO to provide WADO service over the Internet. The other is used as PACS to store and retrieve DICOM files. The open-source PACS (dcm4chee), which contains 20 GB of DICOM files (more than 9.6 million images), is deployed on this server. The five PCs are used as workstation or Web browser to retrieve images or reports. All servers and PCs are connected by a gigabit switch. The iPad and smart phone are used to access images or reports over wireless connections, such as Wi-Fi, 3G, 4G, or WiMax. Figure 7 shows the scenario wherein SmartWADO is accessed via different devices.

Performance Evaluation

Performance Estimation of Key Points

We evaluate the performance of the four key points as mentioned above to discover the performance bottlenecks; the fetching times of the key points for seven different concurrent retrieval requests were measured. We read all log files in the selected folder, analyze the data, and display results. Figure 8 depicts the average response time and trends of the key points.

From Fig. 8, the average response time to download DICOM files is far higher than that of the other three key points. With the increase of requests, the average response time to download DICOM files witnesses a trend of increase by leaps and bounds and that of the other three key points reveals a general trend of slight rise, especially for cache response time.

Image Cache Hit Rate Analysis

We analyze image cache hit rate with seven different concurrent retrieval requests in a steady state. The data from log files is analyzed and displayed in Fig. 9.

As can be seen from Fig. 9, the image cache hit rate reveals a general trend of slight rise, from 57 to 64 %, with the increase of concurrent requests. There are some slight fluctuations in 15 concurrent requests. The reason is that some of the requests are generated by a program in a random way. This procedure cannot simulate practical scenarios, and it needs to be further optimized.

Overall Performance Evaluation

To evaluate the overall performance of SmartWADO with different concurrent retrieval requests, real-time response time (no cache), cache response time, and overall response time for the seven different concurrent retrieval requests are evaluated. Real-time response time consists of content type validation time, DICOM file download time, and DICOM object parsing and conversion time. Figure 10 shows the average response time and trends of the four indicators.

As indicated in Fig. 10, the real-time response time is far higher than the cache response time. The former reveals a trend of dramatic increase while the latter witnesses a trend of slight rise. The reason is that real-time response needs to retrieve DICOM persistent objects from the remote PACS and



Fig. 9 Image cache hit rate





convert it to a specific format according to Web clients, yet cache response directly delivers the image to the Web browser from the cache. Although, the overall response time is also on the rise, but it is much lower than the real-time response time. The overall response time is no larger than 4 s due to contributions of cache service.

Discussion

Many teleradiology systems have been implemented and deployed worldwide in recent years. Paper [5] compares the properties of MIAPS with those of other systems and the details are depicted in Figure 20 in this paper. As can be seen from the paper, compared with the above systems, our paper provides three advantages. The first advantage stems from the extension to the WADO standard. We present a simple method to extend the WADO standard for regional PACS. The method adds a remote PACS routing table on the server side. The second is associated with image cache service, which proposes an on-demand medical image cache and transmission method based on key-value. The last is that we propose a memory-based real-time monitor method and implement a Web-based interface for dynamic displaying of the value of performance.

The first and third features are not found in the above systems. Medical image cache has been implemented in paper [5] and paper [9]; the former presents a cache structure based on key-value and replacement algorithm follows the FIFO scheme, and the latter uses Least Recently Used (LRU) as the replacement mechanism. The above method improves the performance of image transmission. However, those structures and replacement policies are relatively simple. The cache structure proposed in our paper is also based on key-value, but only cache indexes are stored in the main memory; cache files are stored on disk. This structure not only can quickly determine whether the request has been cached but also can quickly locate cache files. Our replacement mechanism follows the LRFU policy which inherits the benefits of the two policies (Least Recently/Frequently Used) and allows a flexible tradeoff between recency and frequency of references in basing the replacement decision [18]. From Fig. 8, we can draw the following conclusions: (1) Average response time directly related to concurrent request number. With the increase of the concurrent request number, the average response time also increases, though margins of the rises varied. The reason is that more requests exert more pressure on SmartWADO and remote PACS. (2) To download DICOM files is the key to performance. As can be seen from Fig. 9, the image cache hit rate reveals a general trend of slight rise. Figure 10 shows that the proposed cache structure and method contribute a lot to the overall performance of SmartWADO.

Currently, SmartWADO provides function and workflow to access and share medical images and reports, which can be easily integrated with other existing systems, such as PACS, EMR (electronic medical record) system, HER (electronic health record) system, or PHR (personal health record). In this version, we only implement the WADO-URI communication model and cannot provide a query mechanism. The query mechanism is a novel function that modern DICOM Web service should support. In addition, cache service only cache medical images; the other formats, such as DICOM files, medical reports, or videos, cannot be cached.

Conclusion

In this paper, we present SmartWADO which is an extensible WADO middleware based on image cache and real-time Web monitor technology for regional medical image sharing. The extension for the WADO standard, real-time Web monitor, and image cache are the major novel features. SmartWADO is designed as an abstraction layer over deployed PACS, which can retrieve DICOM persistent objects from multiple PACS simultaneously. The experimental results show that the middleware effectively delivers medical images and reports to Web clients over the Internet, regardless of the platform used for access. SmartWADO can be deployed in a hospital, which retrieves DICOM persistent objects from a single PACS, to provide WADO service to medical workers. It also can be applied to work across various institutions, which retrieves DICOM persistent objects from multiple PACS simultaneously, to transmit medical images and reports to Internet users in a way that is transparent to end-user applications.

In future work, we will further improve the system as follows: The first is that we will reconstruct the system and implement the rest of the communication modes (WADO-WS, WADO-RS, QIDO-RS, and STOW-RS) according to the WADO standard (2014 edition) [21]. Then, we plan to further optimize cache service; all persistent object types will be taken into account and a pre-fetching method may be implemented.

Acknowledgments We would like to thank the editor and all anonymous reviewers for their insightful comments. This work is supported by the National Natural Science Foundation of China under Grant Nos. 81360230, 61462051, 71161015, and 61462056; Social Development of Science and Technology Project of Yunnan Province under Grant No. 2010CA016; Applied Fundamental Research Project of Yunnan Province under Grant No.2014FA028; and Science and Technology Innovation Fund Project of China under Grant No. 11C26215305904.

Conflict of Interest There is no declaration of conflicts of interest.

References

- Godinho TM, Silva LAB, Viana-Ferreira C, Costa C and Oliveira JL: Enhanced regional network for medical imaging repositories, Information Systems and Technologies (CISTI), 2013 8th Iberian Conference on. IEEE, 2013, 1–6
- Hiroyasu T, Minamitani Y, Miki M, Yokouchi H and Yoshimi M: Distributed PACS using distributed file system with hierarchical meta data servers, Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE. IEEE, 2012, 5891–5894
- Yang CT, Chen LT, Chou WL, Wang KC: Implementation of a medical image file accessing system on cloud computing, Computational Science and Engineering (CSE), 2010 I.E. 13th International Conference on. IEEE, 2010, 321–326
- Ferraro de Souza R, Becker Westphall C, dos Santos DR Westphall C M: Challenges of Operationalizing PACS on Cloud Over Wireless Networks, ICWMC 2013, The Ninth International Conference on Wireless and Mobile Communications. 2013, 265– 270
- Shen HL, Ma DF, Zhao YW, Sun HL, Ye RW, Huang L, Lang B, Sun Y: MIAPS: a web-based system for remotely accessing and presenting medical images. Comput Methods Prog Biomed 113(1):266–283, 2014

- Koutelakis GV, Anastassopoulos GK, Lymberopoulos DK: Application of multiprotocol medical imaging communications and an extended DICOM WADO service in a teleradiology architecture. International Journal of Telemedicine and Applications. doi: 10.1155/2012/271758, 2012: 4
- Web access to DICOM persistent objects, DICOM part 18, http:// medical.nema.org/dicom/2011/11_18pu.pdf. Accessed 07-02-2011.
- Lipton P, Nagy P, Sevinc G: Leveraging Internet technologies with DICOM WADO. J Digit Imaging 25(5):646–652, 2012
- Valente F, Viana-Ferreira C, Costa C, Oliveira JL: A RESTful image gateway for multiple medical image repositories. IEEE Trans Inf Technol Biomed 16(3):356–364, 2012
- Farruggia A, Magro R, Vitabile S: A text based indexing system for mammographic image retrieval and classification. Futur Gener Comput Syst 37:243–251, 2014
- Koutelakis G, Triantafyllou G, Mandellos G, Koukias M, Lymper Opoulos D: A web PACS architecture based on WADO service of DICOM standard, World Scientific and Engineering Academy and Society (WSEAS), 284–288,2005
- Koutelakis GV, Lymperopoulos DK: PACS through web compatible with DICOM standard and WADO service: advantages and implementation, 2006. EMBS'06. 28th Annual International Conference of the IEEE. IEEE, 2006, 2601–2605
- Bui AAT, Morioka C, Dionisio JDN, Johnson DB, Sinha U, Ardekani S, Taira RK, Aberle DR, El-Saden S, Kangarloo H: OpenSourcePACS: an extensible infrastructure for medical image management. IEEE Trans Inf Technol Biomed 11(1):94–109, 2007
- Arguiñarena EJC, Macchi JE, Escobar PP, del Fresno M, Massa JM, Santiago MA: Dcm-ar: a fast flash-based Web-PACS viewer for displaying large DICOM image, Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE. IEEE, 2010, 3463–3466
- Drnasin I, Grgic M: The use of mobile phones in radiology, Proceedings of ELMAR 17–21, 2010
- Koutelakis GV, Lymberopoulos DK: WADA service: an extension of DICOM WADO service. IEEE Trans Inf Technol Biomed 13(1): 121–130, 2009
- Noumeir R, Pambrun JF: Images within the electronic health record, Image Processing (ICIP), 2009 16th IEEE International Conference on. IEEE, 2009, 1761–1764
- Lee D, Choi J, Kim JH, Noh SH, Min SL, Cho Y, Kim CS: LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies. IEEE Trans Comput 50(12):1352– 1361, 2001
- 19. HighChart. http://www.highcharts.com. Accessed 2014/6/29
- DCM4CHE and DCM4CHEE. http://www.dcm4che.org. Accessed 2014/6/29
- Web Access to DICOM persistent Objects, DICOM PS3.18 2014a -Web Services. http://medical.nema.org/medical/dicom/2014a/ output/pdf/part18.pdf, Accessed 2014/6/29