Expert Systems with Applications 36 (2009) 10708–10726

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Service-oriented grid computing system for digital rights management (GC-DRM)

Min-Jen Tsai *, Yuan-Fu Luo

Institute of Information Management, National Chiao Tung University, 1001 Ta-Hsueh Road, 300 Hsin-Chu, Taiwan, ROC

ARTICLE INFO

Keywords:
Condor
Distributed computing
Globus Toolkit
Grid computing
Peer-to-peer
Web Services

ABSTRACT

Service-oriented computing and applications have recently gained significant attention since they provide new service infrastructure and development of service-oriented technology. Under such trend and ubiquitous computing requirement, grid computing is becoming popular in scientific and enterprise computing due to its flexible deployment and implementation. In this paper, we proposed a service-oriented digital rights management (DRM) platform based on grid computing (called GC-DRM) which is in the compliance of Grid Portal standards by using porlet. The platform integrates Globus Toolkit 4 and Condor 6.9.2 and uses web 2.0 to construct the web-based user interface for providing job submission, control, management, monitor for DRM services. GC-DRM can provide different categories of services which include watermark embedding and extraction, image scrambling, visible watermark embedding, image tamper-detection and recovery. In addition, GC-DRM has been applied to analyze the robustness of digital watermark by filter bank selection and the performance can be improved in the aspect of speedup, stability and processing time compared with NaradaBrokering based Computing Power Services (NB-CPS) and Web Services based Computing Power Service (WS-CPS). Therefore, GC-DRM can be concluded as a superior service-oriented computing which provides the user friendly environment with efficient DRM service performance based on grid computing architecture.

1. Introduction

Enterprises are generally under the pressure to improve their computing powers based on their enterprise computing needs and the technology development. One solution is to pursue the new computer hardware and replace the old ones but the software integration will become another bottleneck. If enterprise information systems need to be expanded or updated with a full replacement of equipment, such system does not have the scalability from software engineering point of view. In order to enable the system to be easily extended, and reduce the total expenses within the organization, SOA (Tsai et al., 2006; Tsai & Wang, 2008) (Service-Oriented Architecture) concept is becoming important. To achieve the goal of SOA, the system should be designed to meet the loosely coupled environment with open standard compatibility requirements. Under such circumstance, the other solution is to continue to use the existing systems with the help of the Internet and apply the SOA with distributed computing technology which will discover and collect the idle computer resources together and save businesses the hardware cost. Therefore, the new trend of enterprise computing has lately been switched into P2P computing or grid computing (Baker, 2005; Ernemann et al., 2002; Foster & Iamnitchi, 2003; Hamscher et al., 2000; Hastings et al., 2003; Scheres et al., 2005; Talia & Trunfio, 2003) from the traditional client-server framework. Although both distributed computing model suit the job allocation for the implementation of the work, but small-scale business organizations, in the use of these distributed computing model, still facing safety, motivation, flexibility, compatibility and process management issues. Therefore, Tsai et al. (2006) proposed a Web Services based Power Computing Services (WS-CPS) architecture and deployed WS-CPS in the trusty network, to meet these issues.

Because of the advantages of digital media and rapid development of digital signal processing, a wide variety of multimedia contents have been digitalized and easily distributed or duplicated without any reduction in quality through both authorized and unauthorized distribution channels like Internet. Therefore, intellectual property which leads to the Digital Rights Management (DRM) (Macq, Dittmann, & Delp, 2004; Mandal, 2005; Merabti & Llewellyn-Jones, 2006) concept to ensure the digital rights becomes a critical issue we concerned in an era of knowledge-based economy. From previous studies which are using different distribute computing strategies to promote the applications of DRM, Tsai et al. (2006) has analyzed the robustness of digital image watermark by using Web Services (Huhns & Singh, 2005). To further improve the system resilience, fault tolerance, efficiency of job scheduling and the instability in congested network environment, a NaradaBrokering based Computing Power Services (NB-CPS) (Tsai & Hung, 2009) has been applied to utilize the P2P grid to integrate

* Corresponding author. Tel.: +886 3 571 2121x57406; fax: +886 3 572 3792.
E-mail address: mjtsai@cc.nctu.edu.tw (M.-J. Tsai).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.02.066

the computational grids, P2P networks under the hybrid environment.

In this study, we intend to apply service-oriented grid computing architecture to provide the DRM service (GC-DRM) by using the digital image watermarking technology and intend to achieve speedy computing environment with the following objectives:

(1) *Flexible resource sharing*: Computational resources can be flexibly shared through the Internet with load balance approach.
(2) *Open standards*: Open standards can help the integration of different grid applications and the development programs from various virtual organizations with compatibility, scalability.
(3) *Friendly interface and easy of use*: Grid computing systems are generally complicated for users. GC-DRM intends to provide an easy-to-use interface operation.

This paper will be organized as following. Section 2 will briefly explain the related works of the grid computing development and DRM essentials. In Section 3, the GC-DRM design and implementation will be explained. The performance comparison and discussion is analyzed in Section 4 and conclusion will be given in Section 5.

2. Related works

2.1. Grid computing

Grid computing (Baker, 2005; Ernemann et al., 2002; Foster & Iamnitchi, 2003; Hamscher et al., 2000; Hastings et al., 2003; Scheres et al., 2005; Talia & Trunfio, 2003) is a phrase in distributed computing that means multiple independent computing clusters acting like a "grid" because they are composed of resource nodes not located within a single administrative domain. Under such deployment, grid computing generally can offer online computation or storage such as the metered commercial service, utility computing or computing on demand. In addition, grid computing can create the "virtual supercomputer" by using spare computing resources within an organization through the network of geographically dispersed computers. Therefore, grid computing can be applied to a large number of computing for advanced scientific, mathematical, academic problems or sophisticated engineering applications. Besides, the computing resources from different geographical or organizational management provided by varied professional groups, these communities will be cooperated as a virtual organization (VO) (Foster, Kesselman, & Tuecke, 2001).

2.1.1. Globus Toolkit (GT)

The Open Grid Services Architecture (OGSA) aims to define a new common and standard architecture for grid-based applications. OGSA defines what Grid Services are, what they should be capable of, what types of technologies they should be based on, but does not give a technical and detailed specification (which would be needed to implement a Grid Service). Grid Services are specified by the Open Grid Services Infrastructure (OGSI) (Foster, 2005) which is a formal and technical specification of the concepts described in OGSA.

The development of grid computing is generally necessary to choose a useful tool as the middleware. Globus Toolkit (Allen, 2005; Grid Application Toolkit, 2005; The Globus Alliance, 2007) is adopted in this study since it is the only free open source programming with widely support and its aim is to deliver the three design strategies mentioned in Section 1. The Globus Toolkit is an implementation of OGSI. In addition, the Globus Consortium is a non-profit organization formed by global computing leaders who support the Globus Toolkit, the de facto standard for open source grid computing infrastructure. The Globus Consortium also leverages its broad base of participants to further accelerate the evolution of Grid in the enterprise applications.

The evolution of Globus Toolkit (GT) and web application is shown in Fig. 1. From the early version of GT, the GT3:OGSI is not compatible to the general Web Service like WSDL (Foster, 2005). Since GT4 starts to use the state management standard prospective, GT4 is compatible with Web Service Resource Framework (WSRF) (Foster, 2005) which can support the traditional Web Service, WSN (WS-Notification) to refactor the OGSI functions.

The design of Globus Toolkit follows the Grid architecture (Allen, 2005) which is often described in terms of "layers" as shown in Fig. 2. Each layer provides a specific function and they are Fabric layer, Resource layer, Middleware layer and Applications layer respectively. The higher layers are generally user-centric, whereas lower layers are more focused on computers and networks: hardware-centric. Their brief description is as follows.

The lowest layer is the fabric, which connects Grid resources. It implements the local, resource-specific operations that occur on specific resource and access control defined by the interface. Above the fabric layer lies the resource layer: actual Grid resources, such as computers, storage systems, electronic data catalogues, that are connected to the network. Resource layer contains a number of protocols and services, such as HTTP-based Grid Resource Allocation Management (GRAM) used for allocation of computational resource and monitoring, high-performance data access and transport (Grid FTP) for high-speed transfers.

The next layer is the middleware layer which provides the tools that enable the various elements (servers, storage, networks, etc.) to participate in the Grid. The highest layer of the structure is the application layer, which includes applications in science, engineering, business, finance and more, as well as portals and development toolkits to support the applications.

The topmost layer is the applications layer comprises the user applications by using each layers of interfaces provided within a VO environment. Applications are constructed in terms of, and by calling upon, services defined at lower layer. In most common Grid architectures, the application layer also provides the so-called serviceware, the sort of general management functions such as measuring the amount a particular user employs the Grid, billing for this use (assuming a commercial model), and generally keeping accounts of who is providing resources and who is using them – an important activity when sharing the resources of a variety of institutions amongst large numbers of different users.

Within the middleware layer, there is a layer of resource and connectivity protocols, and a higher layer of collective services. Resource and connectivity protocols handle all "Grid-specific" network transactions between different computers and Grid resources. The Grid's network is the Internet. Computers contributing to the Grid must recognize Grid-relevant messages. This is done with communication protocols, which allow the resources to communicate with each other, enabling exchange of data, and authentication protocols, which provide secure mechanisms for verifying the identity of both users and resources.

2.1.2. Condor

A job scheduler is an enterprise software application which can convert a collection of distributed workstations and cluster into a high-throughput computing facility. Since Condor Project (2007) is the only completely free software, this study has adopted it to perform the job scheduler function and it will be briefly explained as follows.

Condor uses cycle scavenging technology which is the key concept using idle computer power with appropriate resources,

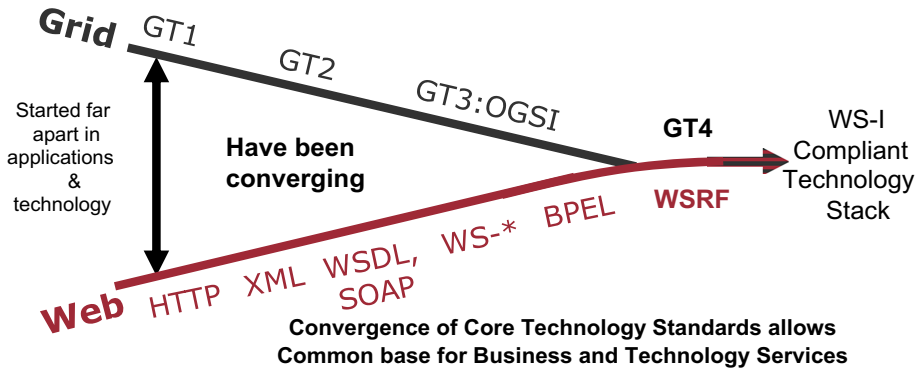

Fig. 1. The evolution of Globus Toolkit.

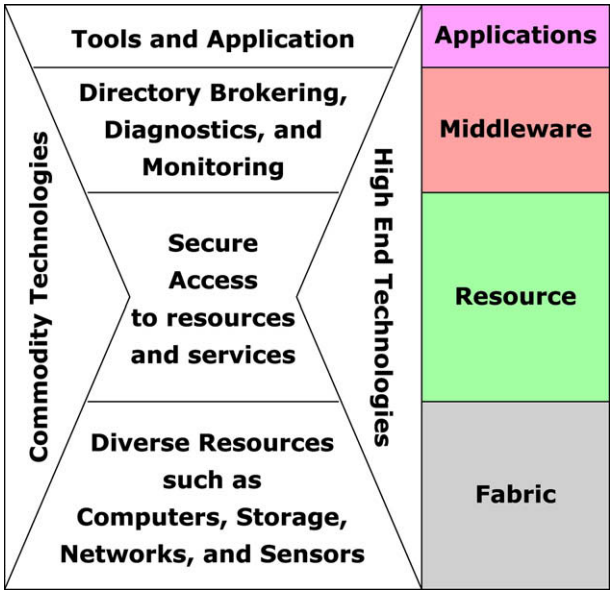

Fig. 2. The layered Grid architecture and hourglass model.

Condor was originally designed for applications within a LAN, and a single administrative domain, and then was developed for multiple management domain. Condor Pool architecture is shown in Fig. 3 which is a collection of machines running Condor called a pool. It is composed by the central manager, the submit machine and the execution machine, each has a number of core daemons, and the introduction is as follows.

Central Manager: It is the resource broker for a pool, only one central manager per pool. Condor_collector daemon collects all the relevant information about the state of computer in a Condor pool. Condor_negotiator is responsible for all the matchmaking and dynamic pair, used by the set of the time interval. The Matchmaking cycle periodically performs this work.

Submit Machine: It is the service which responds to all the submitted jobs into the pool. Condor_schedd in the Submit Machine is responsible for the maintenance of the job queue and all inquiries by the implementation of this daemon. Condor_shadow is in the charge of working out relevant resource management with the right to perform the remote system call when job need to access associated files.

Execution Machine: It is the service which executes the submitted jobs. Condor_startd represents a given resource to the Condor pool. It advertises central attributes about that resource that are used to match it with pending resource for matchmaking. Condor_starter actually spawns the remote Condor job and sets up the execution environment, communication with Condor_shadow in submit machine.

To make Condor and Globus Toolkit efficiently work together, Condor team launched the Condor-G (Condor-G, 2007; Frey, 2002) for the grid systems. Condor-G provides the grid computing community with a powerful, full-featured task broker. Used as a front-end to a computational grid, Condor-G can manage thousands of jobs destined to run at distributed sites. It provides job monitoring, logging, notification, policy enforcement, fault tolerance, credential management, and it can handle complex job-interdependencies.

The Condor-G system leverages the recent advances in two distinct areas: (1) security and resource access in multi-domain environments as supported within the Globus Toolkit, and (2) the management of computation and resources within a single administrative domain, embodied within the Condor system. Condor-G combines the inter-domain resource management protocols of the Globus Toolkit and the intra-domain resource and job management methods of Condor to allow the user to harness multi-domain resources as if they all belong to one single domain.

2.2. Web 2.0

The term "Web 2.0" (Gibson, 2007; Treese, 2006; Zajicek, 2007) has become the major mechanism of the web front-end system which uses the new techniques for Internet applications. In addition, W3C has also brought forward the Accessible Rich Internet Applications (ARIA) specification, hoping to add the additional semantic data into HTML and XHTML for better user interface and high dynamic interaction.

Based on the basic HTML, the widely used technologies to implement the Web 2.0 are JavaScript and Cascading Style Sheets (CSS). The use of JavaScript can make Web sites more dynamic with rich user interface. The web pages designed by CSS can enable richly styled elements with additional color design appearance.

The commonly used technical mechanism in Web 2.0 is AJAX (Asynchronous JavaScript and XML) and the most famous AJAX application is Google Map (Google Web Toolkit, 2006). Users can mark destinations and browse the map easily on the web page as a general application. It is not like the traditional Web based interface mode which requires the necessary and particular request to download the target map for appearance. For traditional Web 1.0 mode, the pages make the necessary re-loading at the same time while the request is issued. However, the mechanism of Web 2.0 is an asynchronous manner where there is no need to wait for the background invocation response from the web server. The advantage of such an approach is that pages will not be waiting

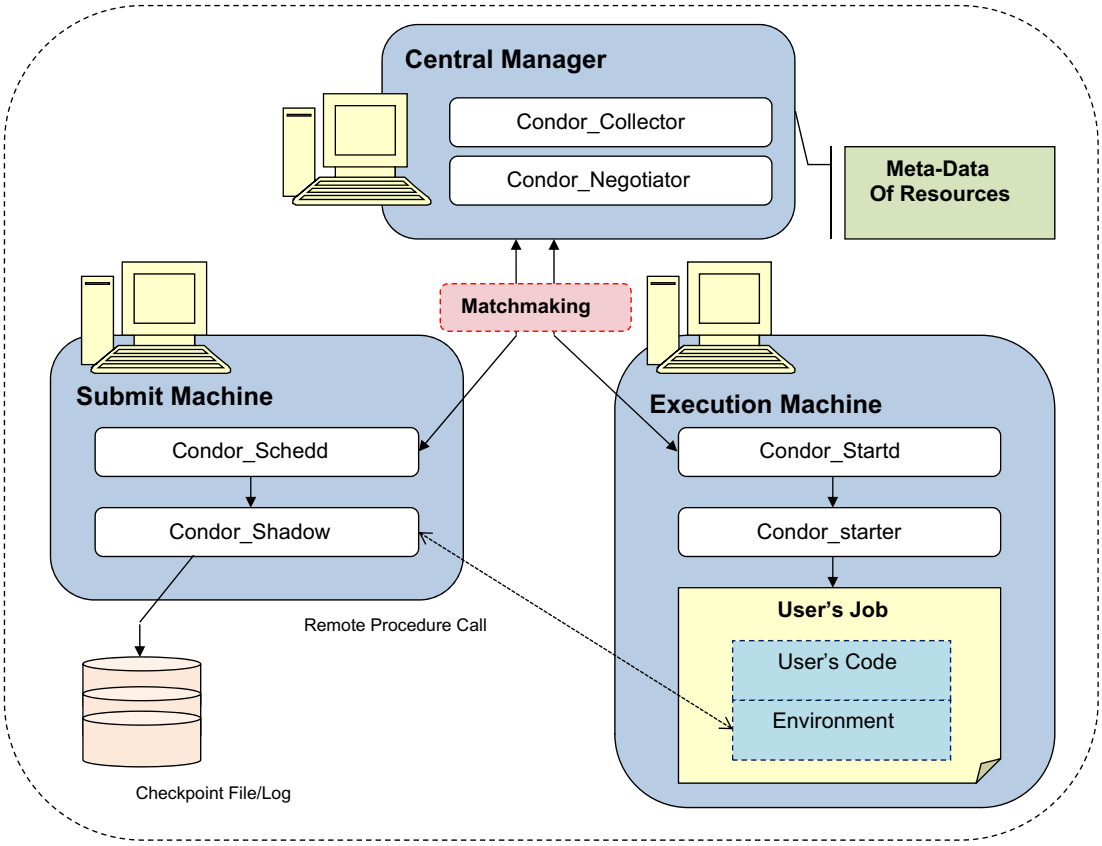

Fig. 3. Condor pool architecture.

for the response to the blank screen, allowing users feel shorter waiting time with high interactivity.

2.3. Digital rights management

Grid computing can be used for a large number of computation intensive applications like 3D medical image processing (Scheres et al., 2005). In addition, the use of the Globus Toolkit as a Data Grid connects each of the major biomedical databases (Hastings et al., 2003) which can be linked up as a big back-end database, providing information on a large number of inquiries and the back-up with large savings of hardware costs. In this study, we will leverage the grid computing capability to the digital imaging applications for the digital rights management (DRM) (Macq et al., 2004; Mandal, 2005) system and propose a GC-DRM system.

In full flourish of the Internet advancement, wide variety of digital contents have been transmitted, disseminated or replicated easily through the Internet. For digital content owners and related industries, unauthorized copies have caused a great loss for their business. Therefore, DRM concept and related technology have been developed in order to solve this problem. According to the Association of American Publishers (AAP), DRM is defined as a technology and processes used to establish and protect digital content of the intellectual property rights in various behaviors. Regardless of its use in the process of reproduction whether acts, DRM can still continue to track and manage digital content usage.

2.3.1. DRM system model

DRM is a system to protect high-value digital assets and control the distribution and usage of those digital assets (Merabti & Llewellyn-Jones, 2006). Even different DRM vendors have different DRM implementations, names and ways to specify the content

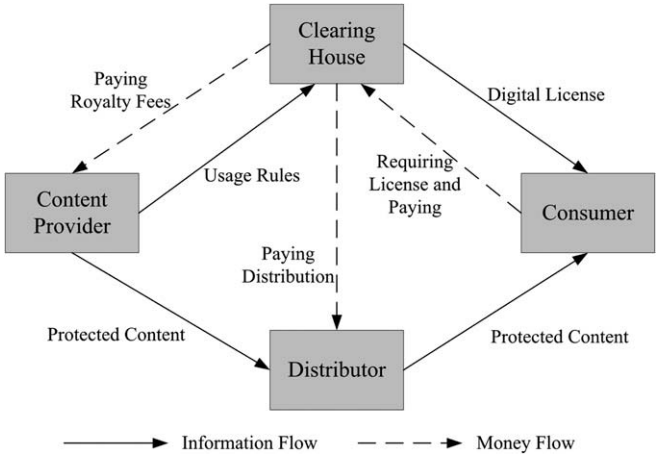

Fig. 4. A common DRM system model (Liu et al., 2003).

usage rules, the basic DRM process is the same which usually involves four parties: the content provider, the distributor, the clearinghouse and the consumer. Fig. 4 is a widely used DRM system architecture (Liu et al., 2003). The system architecture maps the participants to the entire DRM system and the various roles are briefly explained as following:

The content provider such as a music record label or a movie studio holds the digital rights of the content and wants to protect these rights.

The distributor provides distribution channels, such as an online shop or a web retailer. The distributor receives the digital content

from the content provider and creates a web catalogue presenting the content and rights metadata for the content promotion.

The consumer uses the system to consume the digital content by retrieving downloadable or streaming content through the distribution channel and then paying for the digital license. The player/viewer application used by the consumer takes charge of initiating license request to the clearinghouse and enforcing the content usage rights.

The clearinghouse handles the financial transaction for issuing the digital license to the consumer and pays royalty fees to the content provider and distribution fees to the distributor accordingly. The clearinghouse is also responsible for logging license consumptions for every consumer.

A typical DRM model used by current DRM implementations work as follows.

Firstly, the content provider encodes the digital content into the format supported by the DRM system. The digital content is then encrypted and packaged for the preparation of distribution. The content provider may use watermarking technology to embed digital codes into the digital content that can identify the ownership of the content and the usage rules.

Next, the protected content is transferred to the appropriate content distribution server, e.g. web server or steaming server for on-line distribution. The digital license containing content decryption keys and usage rules is sent to the clearinghouse. The usage rules specify how the content should be used, such as copy permit, pay-per-view, etc. At the other end of the process, the consumer downloads the digital content from the web server or requests streaming content from the streaming server. To be able to consume the protected content, the user has to request a valid license from the clearinghouse. After receiving the license request, the clearinghouse verifies the user's identity for example by having the user present a valid digital certificate, charges his account based on the content usage rules, and generates transaction reports to the content provider. Finally, the license is delivered to the consumer's device after the consumer has paid through the e-commerce system, and the protected content can be decrypted and used according to the usage rights in the license.

In this model, consumers can pass along received digital content to other people through super-distribution, which lets vendors market their digital content to a vast amount of potential customers without direct involvement. Although digital content can be freely distributed, to utilize the content, the recipient has to contact the clearinghouse and provide whatever information or payment required for the license.

2.3.2. Digital image watermarking

Since conventional cryptographic systems permit only valid principals (key holders) access to encrypted data, there is no way to track their reproductions or retransmissions once such digital data are decrypted. Consequently, digital watermarking (Macq et al., 2004; Mandal, 2005) has been extensively studied (Cox, Kilian, Leighton, & Shamoon, 1997; Tsai, 2004; Tsai & Lin, 2008; Tsai & Shen, 2008; Wang & Lin, 2004) and regarded as a potentially effective means for protecting copyright of digital media in recent years. Digital watermarking embeds secret information in the digital content to identify the owner and it describes methods and technologies that allow hiding information, for example a number of sequence or recognizable pattern, in digital media, such as images, video and audio. GC-DRM currently only considers the digital image content applications to simplify the system complexity. However, the DRM techniques required for other media address the similar issues in this study and the research results could be extended to other digital content easily.

GC-DRM uses digital image watermarking for the protection of ownership which is a watermark-based protection approach. Regardless of exploiting the digital watermarking techniques, Fig. 5a and b describe the generic structure for watermark

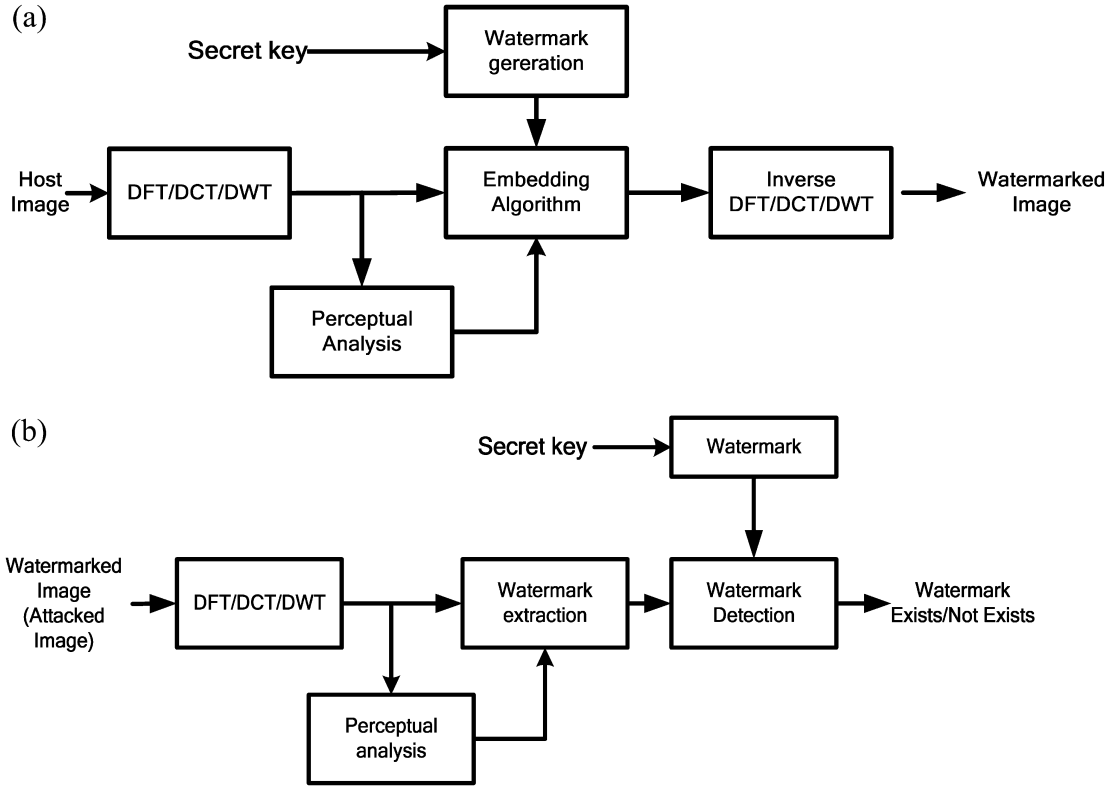

Fig. 5. The flow chart of (a) watermark embedding and (b) watermark extraction.

embedding and extraction processes. First, a host image (original image) directly embeds watermark in spatial domain or is transformed into frequency domain through the well-known spread spectrum approach, i.e. DFT (Discrete Fourier Transform), DCT (Discrete Cosine Transform) or DWT (Discrete Wavelet Transform) (Cox et al., 1997). However, the algorithms using transform domain offer more robust than directly embedding watermark into spatial domain (Cox et al., 1997). Then, coefficients are passed through a perceptual analysis block that determines how strong of the watermark in embedding algorithm so that the resulting watermarked image is imperceptible. The secret key uses to generate watermark and watermark embedding location more. The watermark is embedded using a specific well-designed algorithm based on mathematical or statistical model. If the coefficients in frequency domain, the inverse spread spectrum approach is then adopted to obtain a watermarked image. The watermark extraction applies the similar operations in embedding processes. It employs the inverse operations or uses the mathematical or statistical characteristic to extract the embedded watermark. Watermark detection decides whether an image has been watermarked and the watermark exists or not by calculating the normalized correlation (NC) (Wang & Lin, 2004) between the embedded watermark and the extracted one.

2.3.3. Benchmarking of image watermarking for DRM system

Since watermarking plays an important role for DRM system, it is necessary to build a platform for benchmarking of image watermarking. Therefore, the OpenWatermark (Macq et al., 2004; OpenWatermark, 2008) proposes a modern architecture for cooperative programming exchange that developers can work on their preferred programming language and operating systems.

OpenWatermark is a distributed application system whose initial purpose is to allow the execution and the comparison (i.e., benchmarking) of programs uploaded by the user. The user first logs into a Web site using her/his preferred Web browser, fills a form where she/he is asked to explain some characteristics (such as the programming language used and the syntax of its command-line arguments) of her/his program and to upload it. Those characteristics, programs as well as the input data sets, are stored into an SQL database. The OpenWatermark system (see Fig. 6) determines on which machine the execution should be scheduled and the context of those executions, that is to say, mainly which data sets and parameters should be used and what kind of output should be expected. It then connects to the machines concerned and requests them to download the executable from the database as well as the associated data sets, run it using the previously specified command-line options, and upload back to the database the results of this execution. Finally, the results could be consulted by the user on the same Web site as soon as they become available. The user interface is entirely constituted of Web pages written in Java Server Page (JSP), and communicates with the Java application responsible for the execution of the tested program running on each of benchmarking hosts using Remote Method Interface (RMI) and with the SQL server using Java Database Connectivity (JDBC).

This architecture is, therefore, independent of the operating system used and, hence, easily portable on any platform supporting Java, RMI, and JDBC. This study will use grid computing to build the web 2.0 based DRM system (GC-DRM) and its performance will be compared with the OpenWatermark system.

3. GC-DRM design and implementation

This section will introduce GC-DRM design and its architecture in details. The service-oriented mechanism and applications will be also explained.

3.1. GC-DRM participants

From Section 2.3.1, DRM system generally has four different participants: content provider, distributor, consumer and clearing house. Accordingly, all participants in GC-DRM can be categorized as GC-DRM platform administrator, content provider, and consumer. Since the functions of the clearing house can be applied by using the existing system in order to meet the requirements of electronic payment (Liu et al., 2003; Macq et al., 2004; Merabti

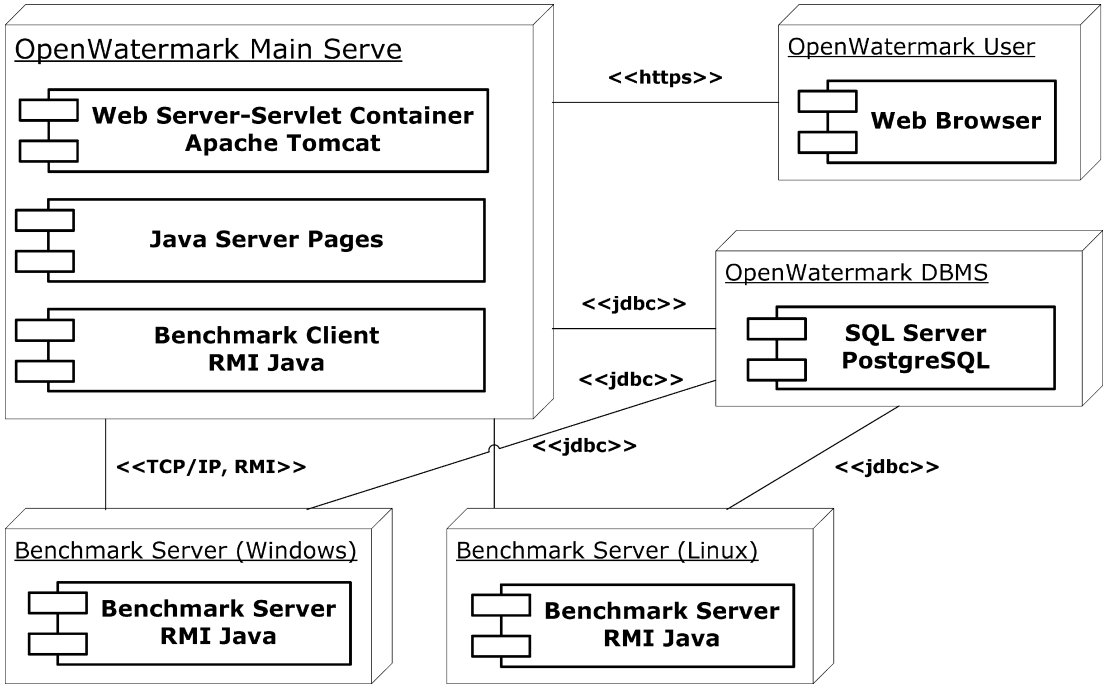

Fig. 6. OpenWatermark architecture.

& Llewellyn-Jones, 2006), the GC-DRM platform administrator can provide the clearing house functions in practice. In brief, each role of the participant is explained as following and their associated relationship is shown in Fig. 13.

GC-DRM platform administrator: the role is responsible for managing the entire platform functionality. In GC-DRM, it merges with the digital content distributor for the sake of a single entrance, but the dual function can be partitioned if it is necessary. The GC-DRM platform administrator will ensure the architecture secure so the whole platform is easy with fairness for consumer use.

Content provider: the digital content creators will use GC-DRM to promote the sale of their digital products with the protection of their intellectual property.

Content consumer: the content consumer will purchase the digital contents through the GC-DRM architecture with legitimate right of ownership. In the mean time, the rights of the digital contents are protected and not infringed.

In GC-DRM, six different operation steps among the participants are depicted in Fig. 7. The steps will be explained as follows.

Step 1: Content consumers send the purchasing request to the digital content provider.
Step 2: Digital content provider receives the request and forward the request to GC-DRM platform administrator to handle the transaction.
Step 3: GC-DRM platform administrator will verify the consumer's identification information.
Step 4: Digital content consumer responds GC-DRM platform verification request. If there is no registration information, the consumer will be required to create the user account. The transaction action will be recorded by the GC-DRM platform.
Step 5: If the content purchase transaction is successful, the watermarked copies of digital content will be transmitted to the digital content provider. The digital content provider will register the purchase information of the consumer for the follow-up check.
Step 6: While the purchase transaction is confirmed, the watermarked digital content will be distributed from the digital content provider to the digital content consumer. If the transaction is not through, further purchasing information will be delivered to the consumer.

Since GC-DRM is a service-oriented grid computing architecture, the platform can be treated as a set of on-demand services on the website or grid computing system on the portal. Therefore, the users only see a single entrance for GC-DRM system and enjoy the distributed computing advantage for DRM services without knowing the detailed GC-DRM operations.

3.2. GC-DRM system architecture

From a system point of view, the GC-DRM system can be constructed by the layered architecture as shown in Fig. 8, namely the Presentation layer at the web server, the Middleware layer at grid middleware container and the Batch system layer at the scheduling unit. Given such systematic scenario, GC-DRM three layers can be installed independently in different computers. Therefore, high SOA is achieved for GC-DRM. The detailed description about each layer is as following:

Presentation Layer: GC-DRM services are mainly selected by users at this layer which is web 2.0 based approach using Apache Tomcat 5.5.23 web Server. The web server provides URL connections via Http and the desktop application mode connects Middleware Layer is also allowed. The presentation layer can be further constructed by UI and CGI layers.

UI Layer presents the system interface for user operation with CGI Layer link. The HTML page can send a request to several CGI programs at same time in order to update the web page dynamically, in the mean time, the purpose of software component reuse is achieved.

CGI layer is mainly responsible for the request of UI layer and uses Servlet, Porlet, JSP server-side API for composition with JavaBean to encapsulate related information about each application service and business logic operation. Servlets (Java Community Process, 2008) are the Java platform technology of choice for extending and enhancing Web servers. Servlets provide a component-based, platform-independent method for building Web-based applications, without the performance limitations of CGI programs. And unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), servlets are server- and platform-independent. Portlets (Java Community Process, 2008) are web components specifically designed to be aggregated in the context of a composite page. Usually, many portlets are invoked to in the single request of a portal page. Each

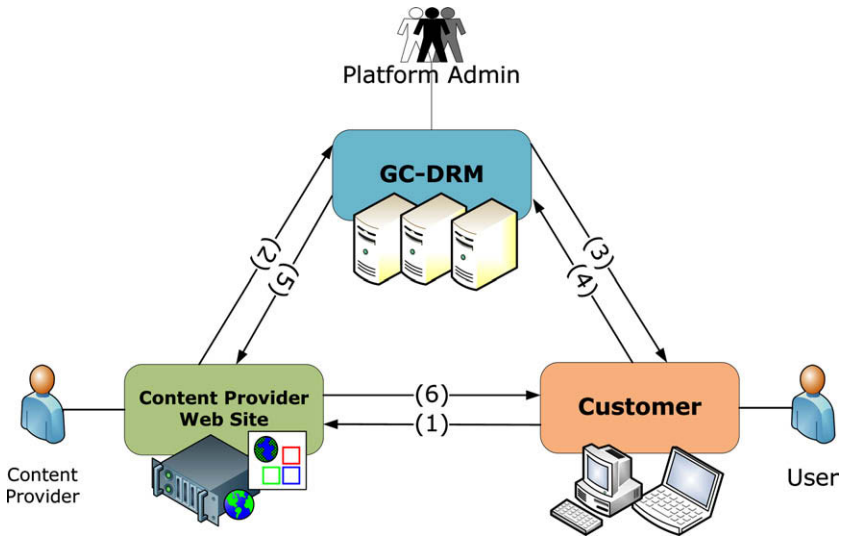

Fig. 7. GC-DRM participant relationship and operation steps.

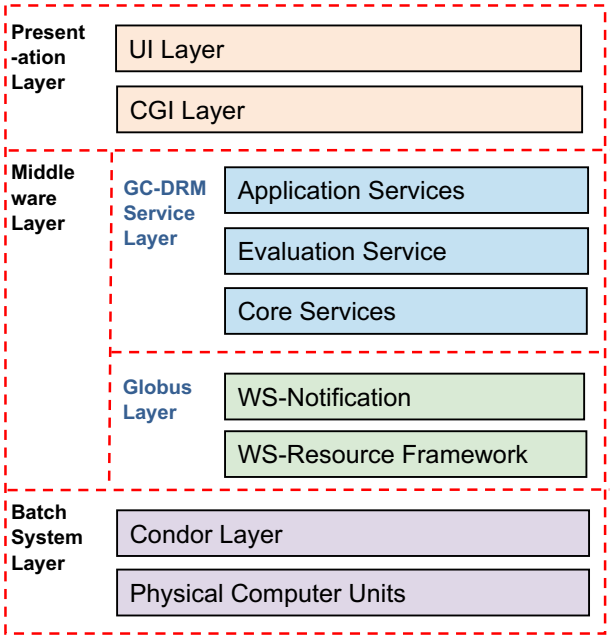

Fig. 8. Three layered architecture for GC-DRM system.

portlet produces a fragment of markup that is combined with the markup of other portlets, all within the portal page markup.

While a certain service is selected by the user, the CGI layer will forward the request to GC-DRM service client via JavaBeans and encapsulate its process at Presentation layer. After this procedure, GC-DRM service client invokes the Middleware layer for service via client stubs and service stubs. A functional view mapped layered architecture is shown in Fig. 9 and the Presentational layer is mapped to the first layer of Fig. 9.

Middleware Layer: This layer mainly uses the Globus Toolkit (ws-core_4.0.4) (Condor-G, 2007; Java Community Process, 2008) to connect with the Presentation Layer and Batch System Layer

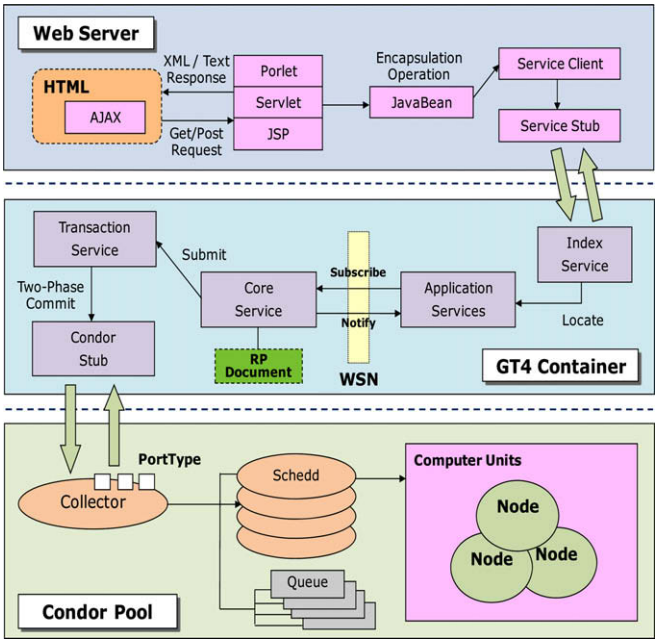

Fig. 9. GC-DRM functional view by mapping layered architecture Fig. 8.

and all GC-DRM application services are deployed at this layer. It comprises GC-DRM service layer and Globus Toolkit layer.

GC-DRM service layer concentrates on the core of the services provided by GC-DRM, where the service usage is defined by WSDL and the application can be seen as a series of core service invocation with varied parameters. While the job of core service is invoked at service layer, it will use Globus layer to store the current job state and notify Presentation layer when job state is changed.

Globus Layer comprises WSRF and WSN. While the core service is invoked through a series of operations, the processed data will be sent to the Batch system layer. In the mean time, the Globus Layer will create WS-Resource with each job by WSRF which is the pairing of a Web service that exposes on more stateful resources. The state of WS-Resources is maintained as a group of properties called resource properties. For WS-Resource in GC-DRM, resource properties will store the job state and associated information about the job. While the job status is changed, the service client of WSN will be initiated to modify the WS-Resource data. The functional behavior of Middleware layer is shown in the second layer of Fig. 9.

Batch System Layer: This layer is responsible to the job dispatching and is currently using Condor 6.9.2 software package (Condor-G, 2007). Since its upper layer is the Middleware Layer, it can allow multiple job dispatching instances co-existing and the Middleware Layer can handle the consolidated management with data processing. This layer comprises Condor Layer and physical computer units.

Condor Layer is constructed by a flexible number of computers called the Condor Pool (Fig. 3) for GC-DRM platform. Due to the policy based approach, it is also possible to partition the Condor Pool into many different pools. The benefit of such arrangement can expedite the system management since the computing units within the same pool are generally with the similar or same specification and the functional features.

The physical computer units refer to the actual processing computing platforms in the Batch System Layer. According to the various demands of each job, the Condor system will safeguard the computer units in GC-DRM. In order to provide multi-functionality, the Condor system offer many configuration options available for users to choose their own preference.

3.3. Core functional invocation of GC-DRM

Core functional invocation of GC-DRM is the foundation of high-level service as a sequence of invocation of application service components. While the application service is called, the core functional invocation will begin action until the job is dispatched to the physical computer units and the functional view is shown in Fig. 10.

The Middleware layer sends jobs to the Condor Layer via Web service interface. Several core functional invocation of GC-DRM services are detailed in Sections 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.3.5.

3.3.1. Job status notification service

Job Status Notification Service (JSNS) is the communication link between the Presentation Layer and Batch System Layer which will report the current job status for GC-DRM users. It uses the publish-subscribe (also called event-driven, notification-based, or observer/observable pattern) approach for loosely coupled systems. WSRF and WSN API of Java ws-core modules in Globus Toolkit provides the capability of JSNS to create a job status buffer in this service as shown in Fig. 11. If job status is changed, WSN initiates the update to GC-DRM service client of Presentation Layer. Without such mechanism, the whole system performance will be slowed down significantly since Presentation Layer needs to poll

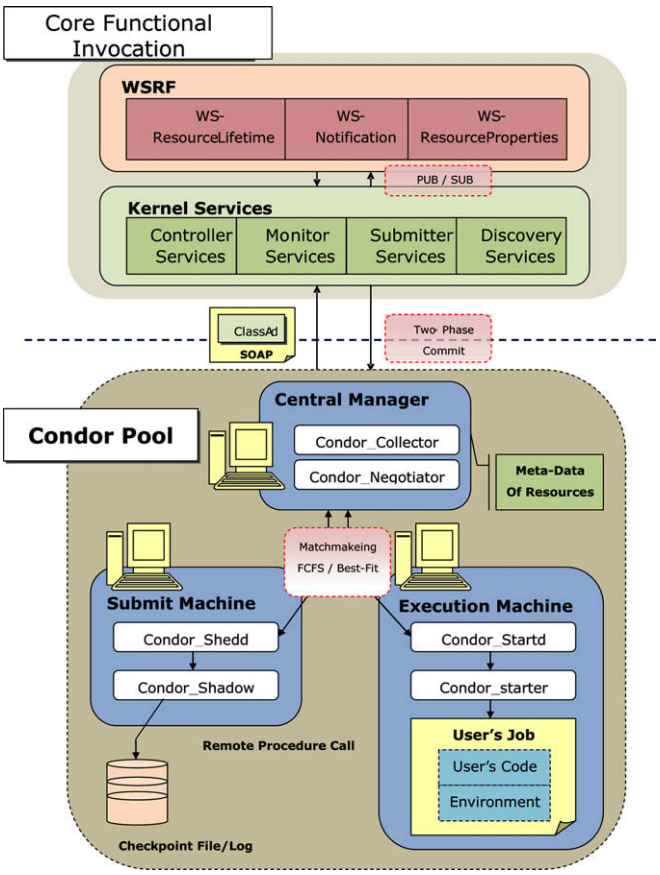

Fig. 10. Core functional invocation from middleware to the Condor pool.

Middleware Layer frequently and Middleware Layer does the similar action of polling to Batch System Layer.

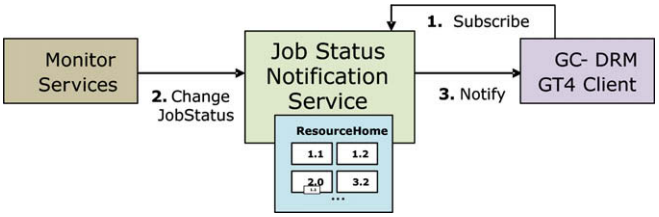

Fig. 12. Applying PUB/SUB flow of JSNS.

At GC-DRM, each job maps to a WS-Resource which contains three ResourceProperty: JobID, scheddURI, JobStatus. Since multiple resources can concurrently exist, the Factory/Instance design pattern (Sotomayor & Childers, 2006) is used for multiple WS-Resource management. The functional invocation flow of WS-Resource creation is shown in Fig. 18. After the establishment of the WS-Resource, JSNS will use WSN API to list the WS-Resource on the topicList and then GC-DRM service client will make the subscription request. For addressing the service, WS-Addressing specifies a construct called EndPoint Reference (EPR) that allows us to address a web service endpoint and its resource. Using EPR address through NotificationConsumerManager class setting, JSNS will establish the subscription action. If Monitor service revises ResourceProperties in the WS-Resource, for example, job status changed from idle to running, the NotifyCallback class for GC-DRM service client will get the job state change message and the flow is shown in Fig. 12. Condor system calls the group of jobs a cluster and each job within the cluster is called a process. Condor job ID contains the cluster number, a period symbol and process number, for example, 1.1. Single job is also with a cluster but with a single process (i.e., process 0). Applying the WS-addressing standards including Uniform Resource Identifier (URI) of grid service and the EPR of WS-Resource, WS-Resource can be easily managed for GC-DRM.

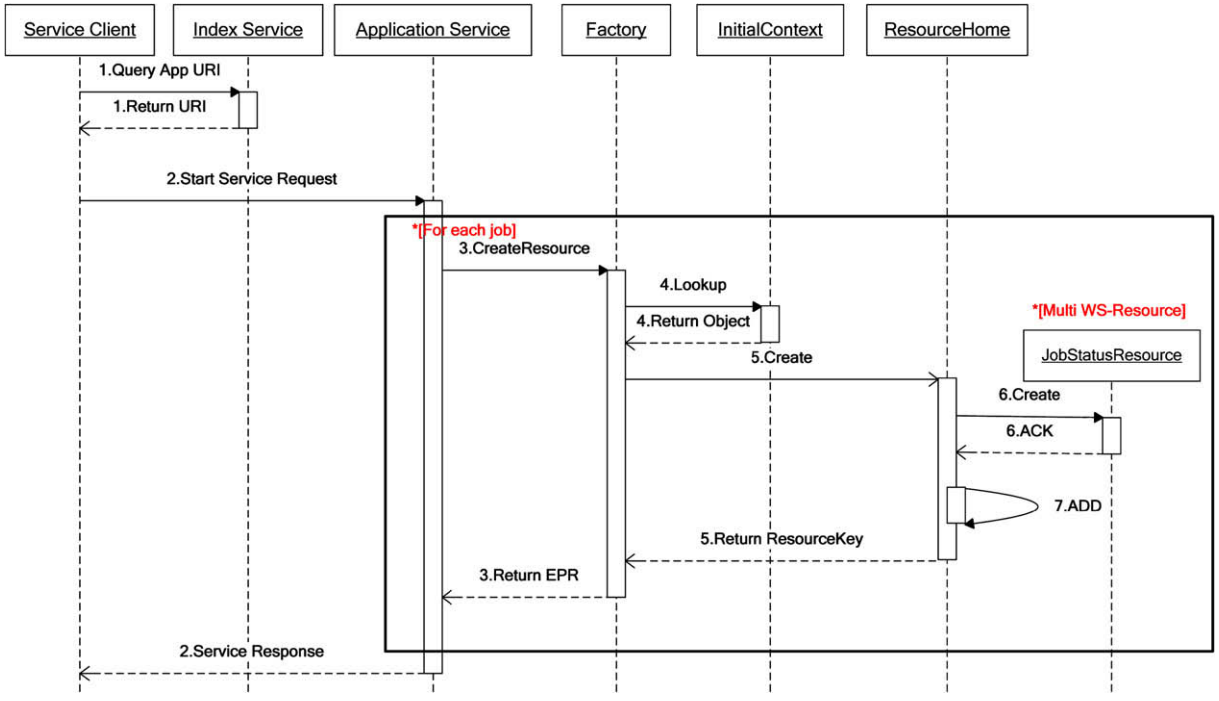

Fig. 11. The functional invocation flow of WS-Resource for JSNS.

3.3.2. Discovery service

Discovery service is primarily responsible for the communication link between the Condor Collector in the Batch System Layer. Since Condor Pool allows many Condor Submitters and the Condor submitter URIs will be registered in the Condor Collector while condor submitter is activated, the Discovery service inquires the Condor Collector and will store the first one of Submitter URI to avoid repeated inquiries with additional overhead. An example is shown in Fig. 13 while two Condor Submitters are in the Condor pool. The Discovery service will inquire the Condor Collector and store Submitter2 URI for future answer and response actions.

3.3.3. Job Controller Service

Job Controller Service (JCS) can handle the jobs sent to the Condor Pool for job removal (Remove action), suspended (Hold action), starting job after suspended (Release action), re-dispatch job (Reschedule action) . . .etc.

Job Controller Service must first get the job location information from the Submitter URI of Discovery Service. A request with JobID to Condor Submitter is issued for the job control. In Condor Pool, each Submitter would have a job queue. While Condor Submitter receives the request of JCS, it will mark the associated Job for designated action. An ACK signal will be sent back to JCS when the request is completed. In Fig. 14, a job removal request is issued from JCS and a submitter URI is found at the Discovery Service. In Condor Pool, the job will be removed from the pool of submitter and the ACK message is sent back to Job Controller service eventually.

3.3.4. Job Monitor Service

Job Monitor Service (JMS) need to inspect the job status within a certain time interval for Batch System layer. The operation begins to make use of Discovery service for Condor submitter URI, then get current job status in Condor Pool. By comparing with the JobStatus ResourceProperty in WS-Resource, the Monitor service will adjust the WS-Resource as shown in step 2 of Fig. 12 if job status is changed. Therefore, WS-Resource likes a variable collection which stores the continuous job states for each job in Condor pool. By comparing the job status from JMS, the ResourceHome class instance of WS-Resource will be updated for current job status. Since the use of WSN API in the GC-DRM service client will subscribe the job status, the GC-DRM Service client at Presentation layer will be notified when the status of subscribed job is changed. A functional invocation flow of Monitor Service is shown in Fig. 15.

3.3.5. Job Transaction Service

Job Transaction Service (JTS) is mainly responsible for jobs generated by application services (explained in Section 3.4) transmitted to the Condor Pool. In this services, it use a mechanism known

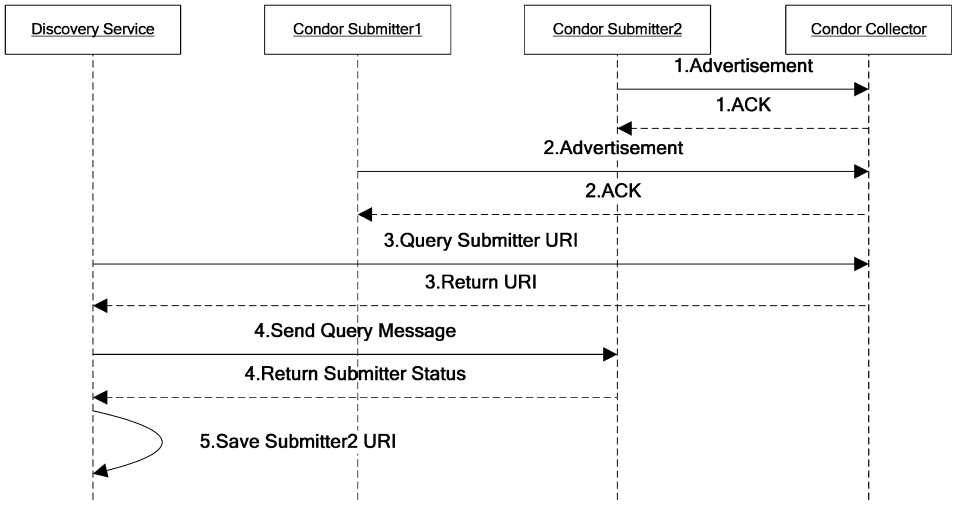

Fig. 13. The functional invocation flow of Discovery Service.

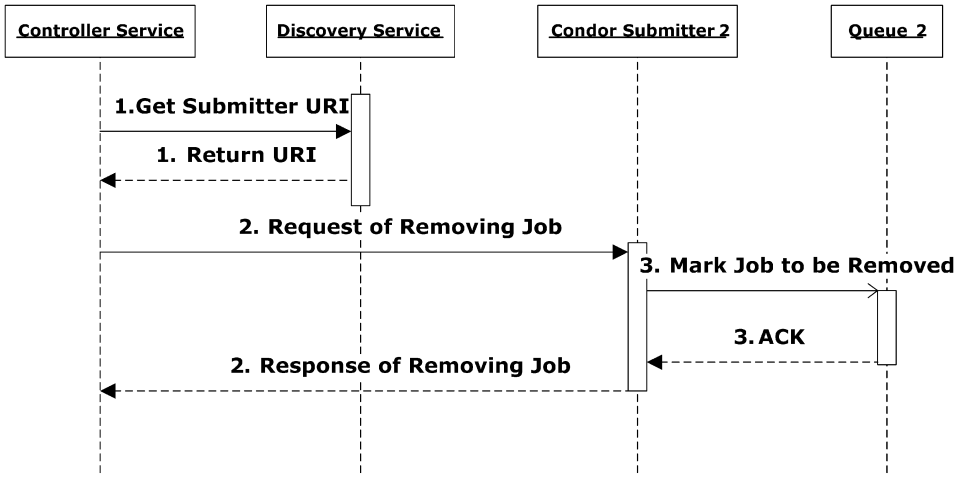

Fig. 14. An example of job removal functional flow by Job Controller Service with the Condor Pool.

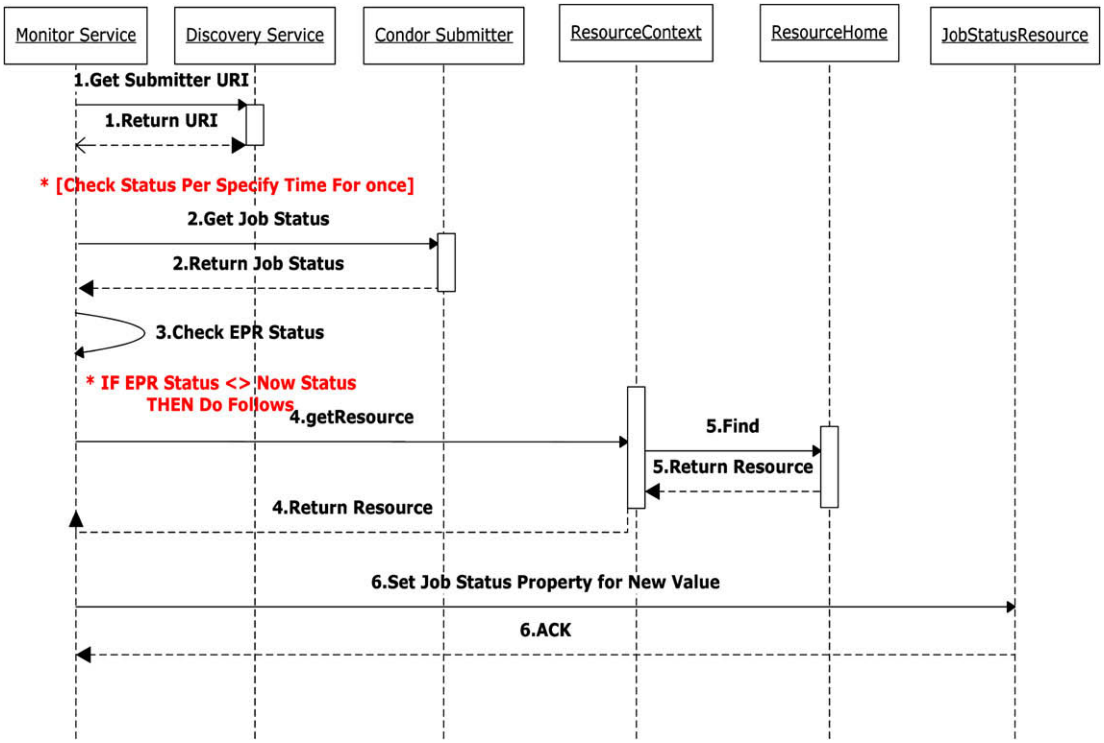

Fig. 15. The functional invocation flow of Monitor Service.

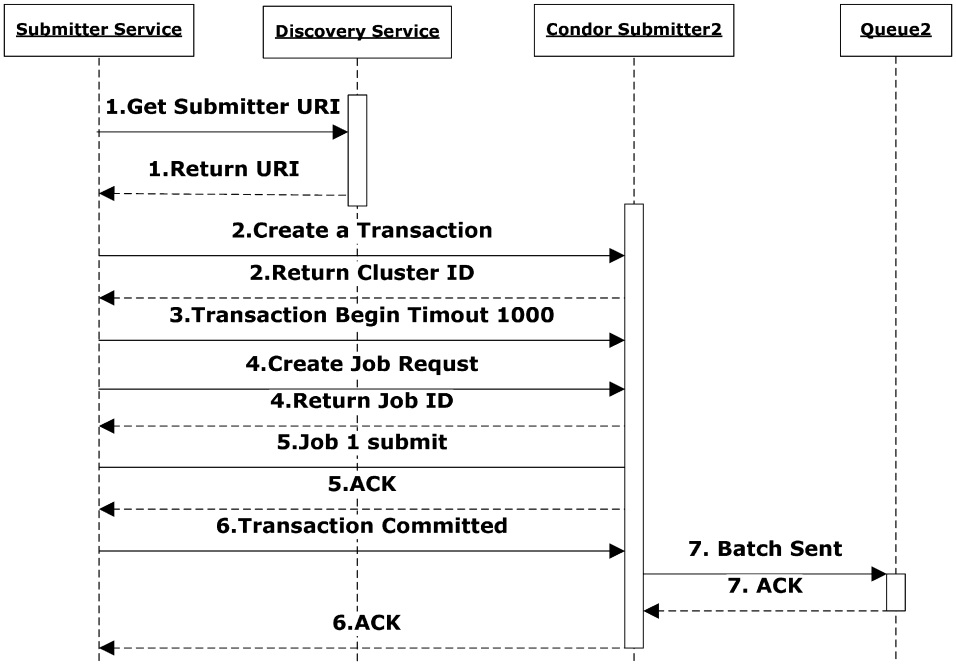

Fig. 16. The two-phase commit mechanism of JTS operation.

as two-phase commit (Samaras, 1995) to ensure job delivery in order to avoid the exception. If exception happens, JTS will rollback with pervious situation to continue the job execution. The two-phase commit mechanism of JTS operation is detailed in Fig. 16. In the beginning, when JTS is ready to send jobs, the Discovery service will first get the Condor Submitter URI and then send the transaction creation request to Condor Submitter. In the mean time, a timeout interval for this transaction will be granted.

In Condor system, the number of ClusterID can be regarded as job group number and ProcID can be regarded as job numbers in one job groups. After transaction is created, it will send the job request for job recreation and reply back with the ProcID. While the job related files are ready after the preparation, JTS will issue the Commit command to first send out those job files to Condor submitter. After Submitter Service issues the Commit command, Condor submitter then will send the tasks to Queue for actual execution. Such mechanism is intended to avoid the case of

exception when it submitted to the Condor Submitter for reliability purpose.

3.4. Application services on GC-DRM

Since GC-DRM is a service-oriented grid computing system, the application services can be flexibly composed or aggregated by using the existing service modules. Add-on modules can be easily put together with the entire system to achieve more customized services. Currently, there are five different application services for GC-DRM including robust watermark embedding and extraction, visible watermarking, image tamper-detection and recovery, filter bank selection and benchmarking. The application service class diagram is depicted in Fig. 17 which shows their interconnectivity.

The application services implement several algorithms which are the core services for GC-DRM and the detailed information will be following.

3.4.1. Watermark embedding and extraction service

Watermarking techniques can be divided into invisible and visible ones according to the perceptivity of watermark data in watermarked contents.

Invisible watermark schemes can be broadly classified into three types: robust watermarks (WM), fragile (or semi-fragile) watermarks and captioning watermarks (Lu & Liao, 2001). For copyright protection and ownership verification, robust watermarks are adopted because they should be nearly resistant to any image processing operations as desired. For content authentication and integrity verification, fragile (or semi-fragile) watermarks are used because they are fragile to certain alterations and modifications of the authenticated multimedia. Semi-fragile watermarks are more practical than fragile watermarks, since they are robust to some mild modifications such as JPEG compression and channel AWGN (additive white Gaussian noise) caused by exchange and storage but fragile to malicious attacks like image cropping. Captioning watermarks are mainly used for conveying side information, so the algorithms are required to convey more information than robust watermarks.

On the other hand, visible watermarking schemes protect copyrights in a more active method. They not only prevent pirates but also recognize copyright of multimedia data. Digital contents embedded with visible watermarks will overlay recognizable but unobtrusive copyrights patterns identifying its ownership. Therefore, a useful visible watermarking technique should remain details of contents and ensure embedded patterns difficult or even impossible to be removed, and no one could use watermarked data illegally.

Currently, GC-DRM is providing WTQ (Wang & Lin, 2004), WTGM (Tsai & Shen, 2008), visible watermarking (Tsai & Lin,

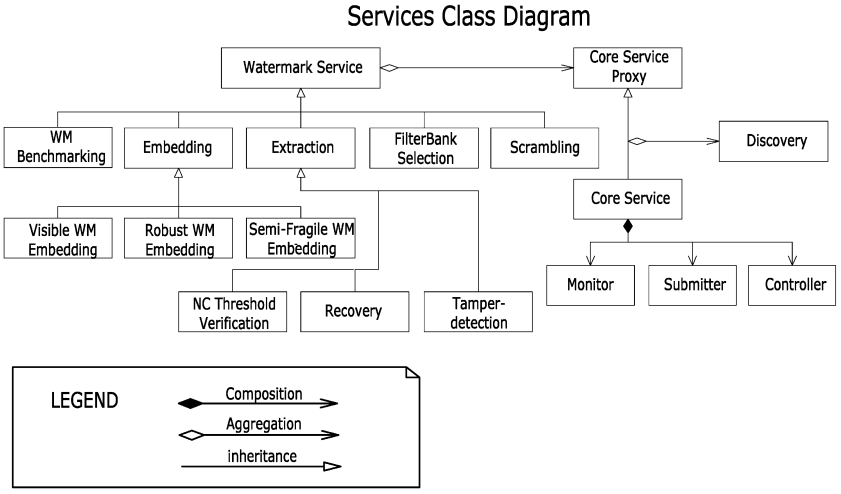

Fig. 17. Application service class diagram.

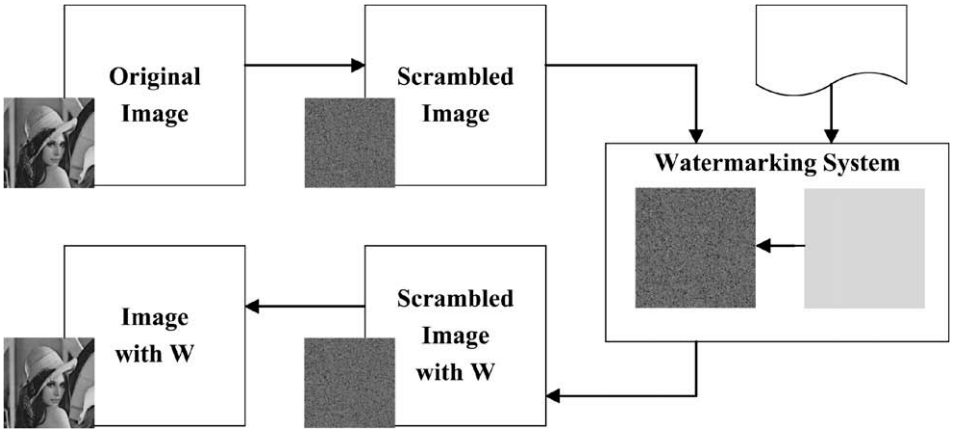

Fig. 18. An illustration of scrambling service in GC-DRM.

2008) and semi-fragile watermarking (Tsai & Chien, 2008) algorithms for watermark embedded and extraction services. Such service could be easily extended according to user needs and use scenario.

3.4.2. Image scrambling service

Compared to original image, a scrambled image is more robust under the detection of attackers which is one of the application services in GC-DRM. Scrambling is like a kind of encryption. When we scramble an image with a secret seed, the scrambled image is just like another image. If someone doesn't know the secret seed, it is difficult for the receiver to reconstruct the scrambled image and extract the watermark correctly.

Image scrambling can be regarded as a "pre-processing" with watermarking system for security enhancement as shown in Fig. 26. This process flow is scrambling with original image and then embeds the watermark into scrambled image. The watermark can be also scrambled as well. After image scrambling, the behavior of the image is like a noise signal and hard for the detector to notice the watermark existence. There are several scrambling option provided in GC-DRM including random ordering, toral automorphism (Tsai, 2000) and Fibonacci transformation (Zou et al., 2004).

3.4.3. Image tamper-detection and recovery service

In GC-DRM, an image tamper-detection and recovery scheme based on discrete wavelet transform (DWT) (Zou et al., 2004) is provided. By using the property of DWT multi-resolution structure, the service will generate the semi-fragile watermark from low frequency bands and embed the recovery information into the high-frequency bands based on the Human Visual System approach. The image tamper-detection system is able to locate precisely any malicious alteration made to the image and restore the altered or destroyed regions based on the recovery mechanism. Therefore, the requirements of ownership protection and tampering detection of digital right management (DRM) are met and the legal usage of digital content is available. An example of tamper-detection and recovery service is shown in Fig. 19 where the tampered area is detection in Fig. 19c and the recovered image is obtained at Fig. 19d.

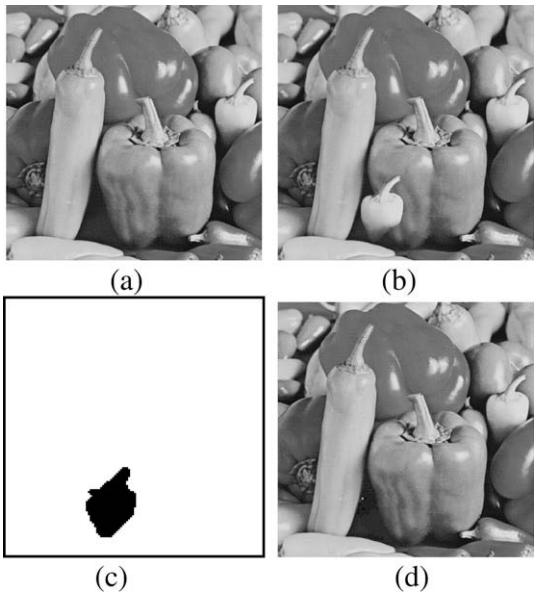

Fig. 19. (a) Original image. (b) Tampered image. (c) Tamper-detection and (d) Recovered image from GC-DRM.

3.4.4. Watermark benchmarking service

In Petitcolas et al. (2001), the research made use of StirMark (2005) to examine the DRM benchmarking through various signal processing, geometrical and non-geometrical attacks. Therefore, GC-DRM also provides the StirMark for watermark benchmarking service. After benchmarking service, normalized correlation (NC) is used to examine the watermark existence and the peak-signal-noise-ratio (PSNR) is compared with the original image for image quality verification. The formula of NC is as following:

$$\rho(W,W') = \frac{\sum_{m=1}^{N_w} w_m w'_m}{N_w} \tag{1}$$

where inserted watermark W and extracted watermark W' are PN sequences of length N_W. The coefficient value is within -1 and 1. The existence decision is "yes" if $\rho\ (W,W') \geqslant \rho_T$ and "no" if $\rho\ (W,W') < \rho_T$. The threshold ρ_T is chosen based on the probability of false positive error P_{fP} which is computed by Kundur and Hatzinakos (1998):

$$P_{fP} = \sum_{n=\lceil N_w \times (\rho_T+1)/2 \rceil}^{N_w} \binom{N_w}{n} P_E^{N_w-n} \cdot (1-P_E)^n \tag{2}$$

Given the reasonable assumption, $P_E = 0.5$ AND $N_W = 512$ as the watermark length, P_{FP} will be as low as 8.45×10^{-9} while $\rho_T = 0.25$. That means the appropriate ρ_T will be selected to meet the requirement given a false positive probability.

The definition of PSNR is

$$\mathrm{PSNR(dB)} = 10 \cdot \log_{10}(255^2/\mathrm{MSE}) \tag{3}$$

where MSE is the mean square error of the watermarked image and the original image per pixel. Even PSNR values do not truly represent the subjective visual quality for watermark images, the values do have the positive correlation with the image fidelity.

In addition, GC-DRM also provides the similar function like OpenWartermark which allows users to upload watermark algorithms for execution and comparison. The performance comparison will be discussed in Section 4.

3.4.5. Filter bank selection service

Since GC-DRM is a grid computing based system, its distributed computing capability needs to be examined in this study. In order to test its performance, a digital watermarking algorithm with wavelet filter bank selection is performed (Tsai, 2004; Tsai et al., 2006). Discrete Wavelet Transform (DWT) based watermark algorithm makes use of filters to filtrate and construct the signals of a digital image. Because filter is a key component of DWT-based digital watermark algorithm, large volume of computing power is required to search the best filter among lots of filter groups. The algorithm consists of decomposition, embedding, reconstruction, and detection procedures. Through the comparison of original watermark and embedded watermark from the received image, a similarity function based on correlation statistics is calculated and the authority of the digital image can be verified. Therefore, the filter bank selection experiment is an appropriate case implementation and its performance will be shown in Section 4.

4. GC-DRM experiments and discussion

The proposed GC-DRM system has been implemented and intensively tested by using commonly available image database. The GC-DRM portal is shown in Fig. 20 where the user can login the home page for the service.

The entire GC-DRM services are simulated as the real e-business transaction format since this study assumes the content providers

Fig. 20. GC-DRM Home page with user login menu.

will cooperate with the GC-DRM platform to sale the digital content and the consumers can purchase the copyright protected digital content online. The transaction flow is shown in Fig. 21 with seven steps. The detailed procedures are as follows.

(1) *Login*: the user can view the GC-DRM home page and register to login to start the GC-DRM services.
(2) *Service Selection*: after successful login, a servie list menu will provide the GC-DRM service options for user. The description about the service will be explained in detail during the selection. An example of visible watermarking usage is shown in Fig. 22 where a visible school logo has been embedded into the Lena image.
(3) *Image Illustration*: while the service is selected, GC-DRM will illustrate the current sales of digital content and the associated information such as the image information, price, author name, . . .etc. An example is shown in Fig. 23.
(4) *Content Purchase*: after the user orders the digital content, this sale will be registered in user's shopping cart for the transaction service.
(5) *Watermark functions*: while online payment action is completed, an GC-DRM algorithm list will be available for user's selection.
(6) *System Dispatch*: after a GC-DRM algorithm is selected in step (5), the selected service will be dispatched for system execution. Associated files with the user transaction will be sent to the back-end for processing. An example of dispatch execution is shown in Fig. 24.
(7) *Real-time Monitor*: GC-DRM can conduct real-time monitoring for the transaction process. A real-time monitoring page is shown in Fig. 25. While the service is completed, the digital content can be downloaded for user and the detailed info about the digital content will be displayed for user reference. An example of watermarked digital image is shown in Fig. 26 where the image scrambling service is applied and the PSNR value of watermarked image is displayed. The watermarked image is ready to be downloaded by choosing the "download" selection at the fourth row, last column of the table.

Through above seven steps, the user can easily purchase the digital image content through GC-DRM. The application services discussed at Section 3.4 on GC-DRM have been fully implemented. Further comparisons will be conducted in the following sections.

4.1. Distributed computing coordination experiment

In order to test the performance of the GC-DRM system for the distributed computing capability, the filter bank selection service is performed and compared. The task is highly dependent on the computing power and filter selection jobs can be operated in parallel, thus, this experiment is suitable for the verification of the system coordination.

The wavelet filter-evaluation algorithm test total 76,177 filters which are grouped into several tasks with 100 filters in each task. Each task is to extract the watermark from a watermarked raw image and JPEG2000 image by different filters and to calculate the correlation coefficient of two images. Without distributed computing coordination, the total computation time consumes 21 h and 12 min by a HP Pentium IV 3.2G MHz desktop PC with 512M DDRII RAM. If the task is performed by a lower capable computer like Pentium III 737 MHz PC with 512M RAM, the total computation time will take 47 h and 19 min.

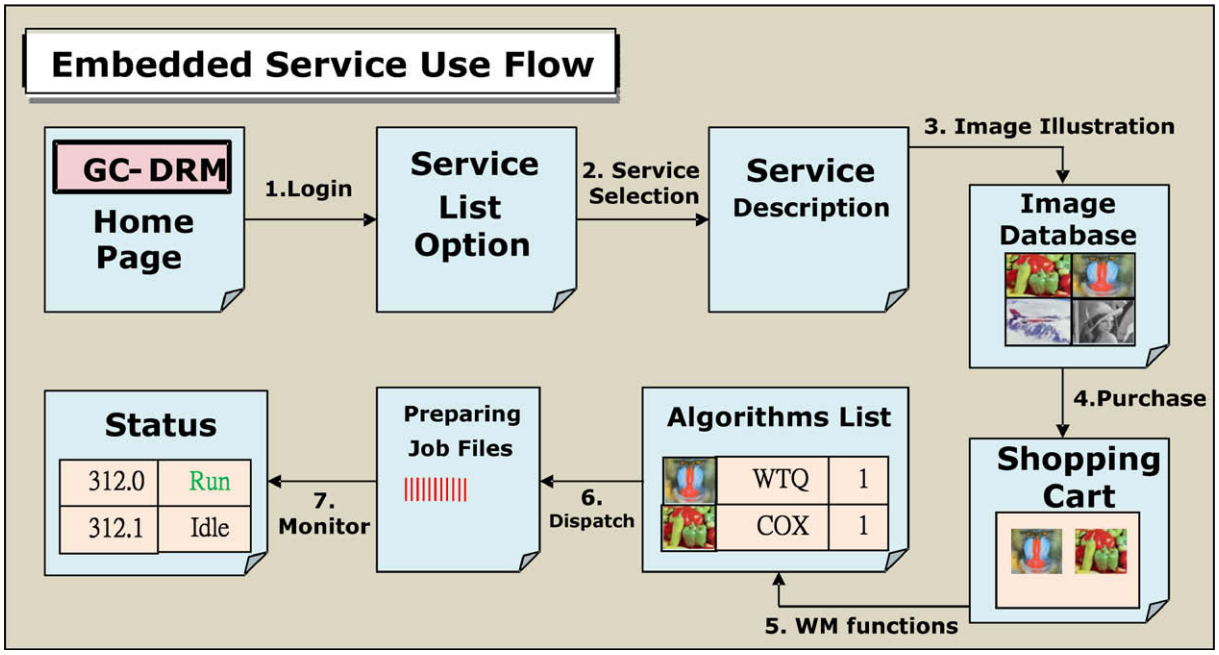

Fig. 21. GC-DRM embedded watermark service use flow.

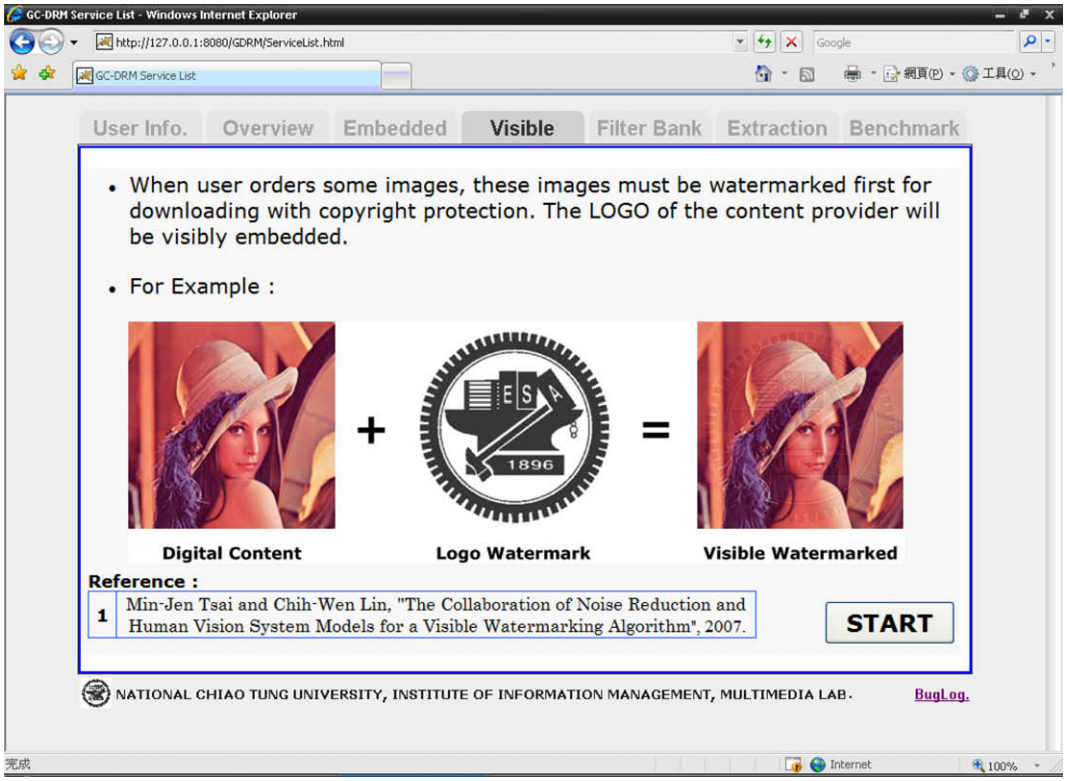

Fig. 22. Visible watermarking service usage page.

Under a Computing Power Service (CPS) (Tsai et al., 2006) environment with 5 same model of Pentium IV 3.2G/512M RAM desktop computer, the total computation time is 4 h 59 min. CPS is a Web Services based P2P architecture for distributed computing in a trusty network. CPS employs Web Services protocols with the flexibility in enterprise computing and integrated with BPEL in workflow control.

In addition, a new computing design – NaradaBrokering based Computing Power Services (NB-CPS) (Tsai & Hung, 2009) has also been tested for filter bank selection experiment. NB-CPS utilize the P2P grid to integrate the computational grids, distributed objects and P2P networks under the hybrid environment. However, the total computation time is 4 h 22 min which is very close to the total computation time of 4 h 20 min by using GC-DRM.

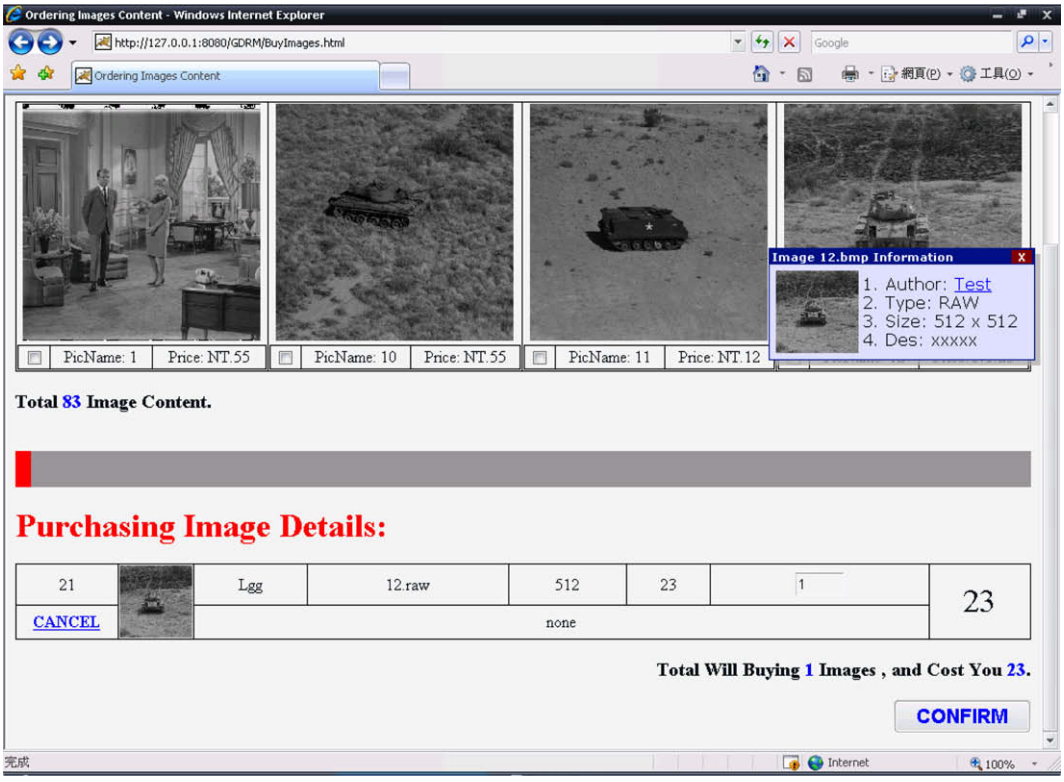

Fig. 23. Digital content illustration.

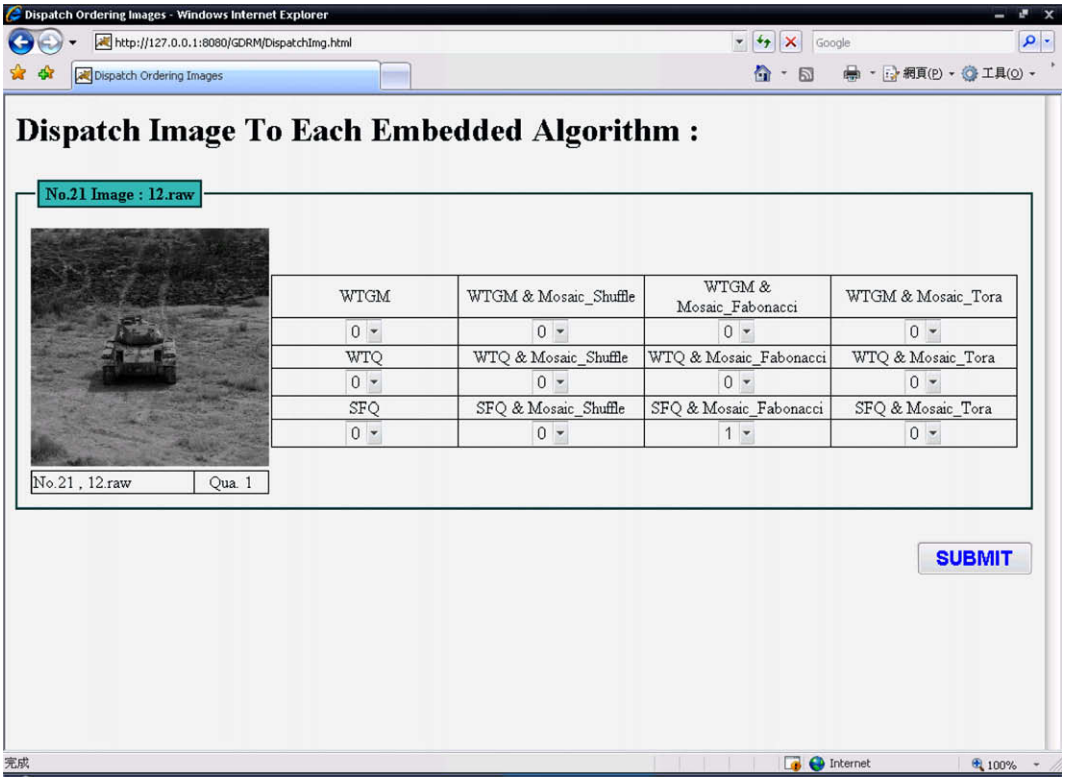

Fig. 24. An example of dispatch execution.

In Table 1, the total computation time for each technique: NB-CPS, WS-CPS and GC-DRM is listed for comparison. We can find the total time spent is pretty comparable among them but GC-DRM consumes the least amount of time. Therefore, the total processing time is reduced the most under GC-DRM approach which proves GC-DRM has the best of collaboration capability.

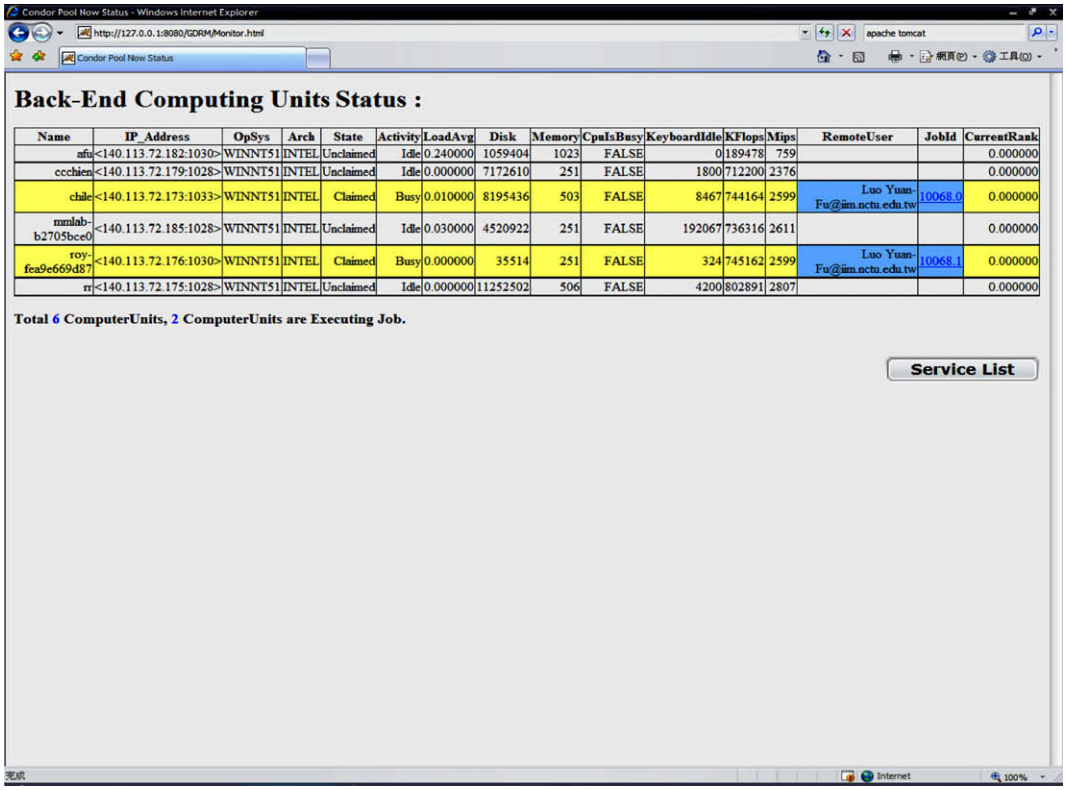

Fig. 25. A real time monitoring page of GC-DRM.

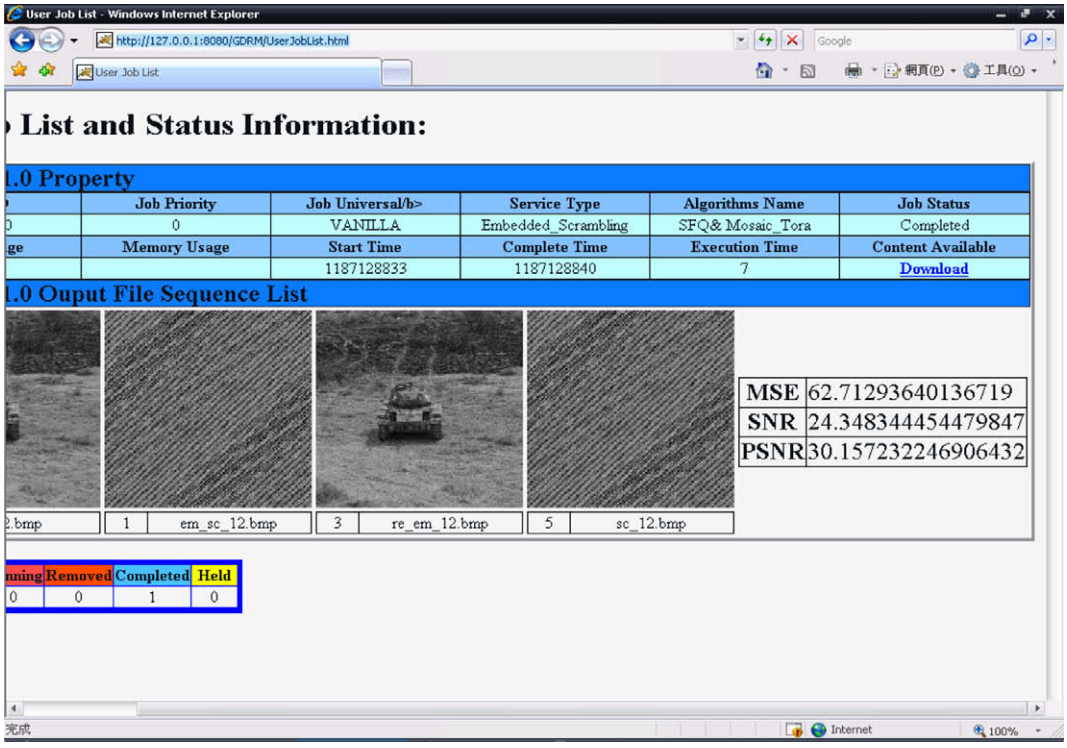

Fig. 26. The info about the watermarked digital content.

To further compare the characteristics of WS-CPS,NB-CPS and GC-DRM, WS-CPS is bases on the architecture of Web Services which lacks fault tolerance mechanism in system architecture as described in Tsai and Hung (2009). NB-CPS is using The NaradaBrokering Project (2006) which is an event brokering system designed to run on a large network of cooperating broker nodes and supports heterogeneous client configurations. However, NaradaBrokering does not like The Globus Alliance (2007) which

Table 1
Total computation time of filter bank selection experiment by NB-CPS, WS-CPS and GC-DRM.

	NB-CPS	WS-CPS	GC-DRM
Computation time	04:22:00	04:59:00	04:20:40

provides much more API supports with many open source codes available. In addition, the service of GC-DRM can be seen as a complete watermarking chain and the digital content will pass through this chain more than once, with various parameters. Therefore, GC-DRM has make the service modular so that its pieces can be easily exchanged, developed by different researchers. Such modularized approach will increase the system management for interoperability and the collaboration. Therefore, GC-DRM is more functional-rich and manageable than NB-CPS or WS-CPS since they all need extra works to add new service in NaradaBrokering or Web Services environment.

4.2. Comparison with OpenWatermark system

Since OpenWatermark is a thread-based benchmarking framework which is to test watermarking algorithms with the same purpose as GC-DRM, it is necessary to compare the performance of both systems. OpenWatermark is not initially designed for distributed computing and need use two benchmark servers for distribute computing including Linux and Windows operating systems, for the purpose of running the algorithms at different platform. However, it does not include the scheduling capability for OpenWatermark and it is not suitable for computation intensive tasks. On the other hand, GC-DRM is based on grid computing and it can use Condor to efficiently connect multiple computing units or groups for data processing under heterogeneous platform. Therefore, GC-DRM can achieve the high throughput computing requirement.

For user interface comparison, OpenWatermark is using traditional web-based interface design which needs to re-load webpage frequently, or refresh the waiting time continuously. An example of OpenWatermark webpage is shown in Fig. 27. Such approach will make users feel longer webpage navigation interval with less efficient system performance. On the contrary, GC-DRM interface is based on Web 2.0 technology and it allow asynchronous request and doesn't need web server response to refresh the web pages. Therefore, it can real-time control the information with the job status for the user with better system performance. The detailed comparison of GC-DRM and OpenWatermark (Macq et al., 2004; OpenWatermark, 2008) is tabulated in Table 2. From Table 2, it is very clear that GC-DRM provides more features with advanced techniques than OpenWatermark.

Table 2
Feature comparison of GC-DRM with OpenWatermark system.

Feature	System	
	GC-DRM	OpenWatermark
Web-end technology	JSP/Servlet/Porlet	JSP
Web technology	Web 2.0	Web 1.0
Interface friendly level	High	High
Distributed technology	Grid computing	Java RMI
Processing large jobs	Yes	No
System response speed	Fast	Slower
Processing dependency job	Yes	No
Real-time job status	Yes	No
Back-end scalability	High flexibility	Low flexibility

4.3. Future works

For future study, GC-DRM will try to apply GSI (Grid Security Infrastructure) (Petiticolas et al., 2001) to develop a trusty platform with higher robustness and security for the distributed system. Another issue is the implementation of Mobile Grid (Condor Project, 2007; Foster et al., 2001) since current trend of grid computing has extended to the mobile devices. Although there are still many limitations for the mobile devices, such as limited memory, limited

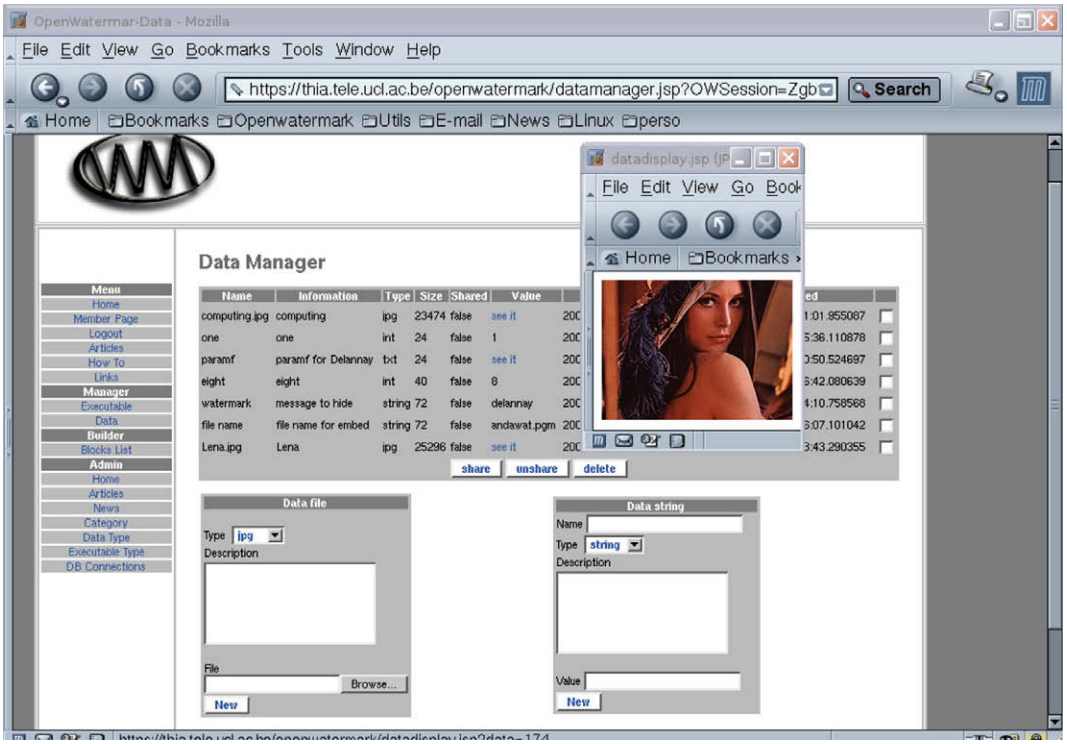

Fig. 27. An example of OpenWatermark webpage.

battery life, small screens, slow GPRS or 3G network speed, DRM applications for mobile device are more important since many digital contents are offered for mobile devices. GC-DRM can broaden its wired network services into wireless network environment or provide a link to the Internet within grid system.

5. Conclusion

In this paper, we proposed a service-oriented digital right management platform based on grid computing (GC-DRM). The platform integrates Globus Toolkit 4 and Condor 6.9.2 and uses web 2.0 to construct the web-based user interface for providing job submission, control, management, monitor and DRM services. GC-DRM provides a user friendly environment with efficient DRM service and benchmarking services. This study has completed three goals: the first one, grid computing environment is constructed and the DRM services are fully implemented with open standards. The second one, GC-DRM is based on the use of improved scheduling for Condor job load sharing and coordination to reduce the total processing time with the best performance by evaluating the filter bank selection compared with NB-CPS and WS-CPS. The third one, we had built a grid computing of the foundation platform in GC-DRM which provides a single interface for users to operations without installing additional software. In summary, the GC-DRM design facilitates the service orientated architecture and the service modularized approach systematically increase the system management for interoperability and the collaboration with high efficiency.

Acknowledgement

This work was supported by the National Science Council in Taiwan, Republic of China, under Grant NSC96-2416-H009-015 and NSC97-2416-H009-034.

References

Allen, G. et al. (2005). The grid application toolkit: Toward generic and easy application programming interfaces for the grid. *Proceedings of the IEEE, 93*(3), 534–550.

Baker, M. et al. (2005). Emerging grid standards. *IEEE Computer, 38*, 43–50.

Condor-G, (2007). <http://www.cs.wisc.edu/condor/condorg/>.

Condor Project, (2007). <http://www.cs.wisc.edu/condor>.

Cox, I. J., Kilian, J., Leighton, F. T., & Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing, 6*(12), 1673–1687.

Ernemann, C. et al., (2002). On advantages of grid computing for parallel job scheduling. In *Proceedings of the 2nd IEEE/ACM international symposium on cluster computing and the grid* (pp. 39–42).

Foster, I. et al. (2005). Modeling and managing state in distributed systems: The role of OGSI and WSRF. *Proceedings of the IEEE, 93*(3), 604–612.

Foster, I., & Iamnitchi, A. (2003). On death, taxes and the convergence of peer-to-peer and grid computing. In *Proceedings of 2nd international workshop on peer-to-peer systems* (pp. 118–128).

Foster, I., Kesselman, C., & Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications, 15*(3), 200–222.

Frey, J. et al. (2002). Condor-G: A computation management agent for multi-institutional grids. *Cluster Computing, 5*, 237–246.

Gibson, B. (2007). Enabling an accessible web 2.0. In *ACM proceedings of the 2007 international cross-disciplinary conference on web accessibility* (Vol. 225, pp. 1–6).

Google Web Toolkit, (2006). <http://code.google.com/webtoolkit/>.

Grid Application Toolkit, (2005). <http://www.gridlab.org/>.

Hamscher, V. et al., (2000). Evaluation of job-scheduling strategies for grid computing. In *Grid computing – GRID 2000: First IEEE/ACM international workshop* (Vol. 1971, pp. 191–202).

Hastings, S. et al., (2003). Image processing for the grid: A toolkit for building grid-enabled image processing applications. In *Proceedings of the 3rd IEEE/ACM international symposium on cluster computing and the grid* (pp. 36–43).

Huhns, M. N., & Singh, M. P. (2005). Service-oriented computing: Key concepts and principles. *IEEE Internet Computing, 9*(1), 75–81.

Java Community Process, (2008). <http://jcp.org/en/home/index/>.

Kundur, D., & Hatzinakos, D., (1998). Digital watermarking, using multiresolution wavelet decomposition. In *Proceedings of the IEEE ICASSP* (Vol. 5, pp. 2869–2972).

Liu, Q. et al. (2003). Digital rights management for content distribution. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers* (Vol. 21, pp. 49–58).

Lu, C. S., & Liao, H. Y. M. (2001). Multipurpose watermarking for image authentication and protection. *IEEE Transactions on Image Processing, 10*(10), 1579–1592.

Macq, B., Dittmann, J., & Delp, E. J. (2004). Benchmarking of image watermarking algorithms for digital rights management. *Proceedings of the IEEE, 92*(6), 971–984.

Mandal, P. et al., (2005). Watermark based digital rights management. In *Proceedings of the international conference on information technology: Coding and computing* (Vol. 1, pp. 74–78).

Merabti, M., & Llewellyn-Jones, D. (2006). Digital rights management in ubiquitous computing. *IEEE Multimedia, 13*(2), 32–42.

OpenWatermark, (2008). <http://www.openwatermark.org/>.

Petiticolas, F. A. P. et al. (2001). A public automated web-based evaluation service for watermarking schemes: StirMark benchmark. In *Proceedings of the SPIE: Security and watermarking of multimedia contents III* (Vol. 4314, pp. 575–584).

Samaras, G. et al. (1995). Two-phase commit optimizations in a commercial distributed environment. *Distributed and Parallel Databases, 4*(4), 325–360.

Scheres, S. H. W. et al., (2005). Grid computing in 3D-EM image processing using Xmipp. In *Proceedings of the 18th IEEE symposium on computer-based medical systems* (pp. 561–563).

Sotomayor, B., & Childers, L. (2006). *Globus Toolkit 4: Programming Java services*. Morgan Kaufman.

StirMark, (2005). <http://www.petitcolas.net/fabien/software/StirMarkBenchmark_4_0_129.zip>.

Talia, D., & Trunfio, P. (2003). Toward a synergy between P2P and grids. *IEEE Internet Computing, 7*, 95–96.

The Globus Alliance, (2007). <http://www.globus.org/>.

The NaradaBrokering Project, (2006). Indiana University, <http://www.naradabrokering.org/>.

Treese, W. (2006). Putting it together: Web 2.0: Is it really different? *netWorker, 10*(2), 15–17.

Tsai, M. J. et al. (2000). Joint wavelet and spatial transformation for digital watermarking. *IEEE Transactions on Consumer Electronics, 46*(1), 241–245.

Tsai, M. J. (2004). Filter bank selection for the ownership verification of wavelet based digital image watermarking. *ICIP, 5*, 3415–3418.

Tsai, M. J. et al., (2006). In *A collaborated computing system by web services based P2P architecture*. *Springer's lecture notes on computer science* (Vol. 3865, pp. 194–204).

Tsai, M. J. & Hung, Y. K. (2009). Distributed computing power service coordination based on peer-to-peer grids architecture. Part 2. *Expert Systems with Applications*, *36*(2), 3101–3118.

Tsai, M. J., & Chien, C. C. (2008). Authentication and recovery for the wavelet-based semi-fragile watermarking. *Optical Engineering, 47*(6), 067005.

Tsai, M. J., & Lin, C. W. (2008). Wavelet based multipurpose color image watermarking by using dual watermarks with human vision system models. *IEICE Transactions on Fundamentals, E91-A*(6), 1426–1437.

Tsai, M. J., & Shen, C. H. (2008). Differential energy based watermarking algorithm using wavelet tree group modulation (WTGM) and human visual system. *IEICE Transactions on Fundamentals, E91-A*(8).

Tsai, M. J., & Wang, C. S. (2008). A computing coordination based fuzzy group decision-making (CC-FGDM) for web service oriented architecture. *Expert Systems with Applications, 34*(4), 2921–2936.

Wang, S. H., & Lin, Y. P. (2004). Wavelet tree quantization for copyright protection watermarking. *IEEE Transactions on Image Processing, 13*(2), 154–165.

Zajicek, M. (2007). Web 2.0: Hype or happiness? In *Proceedings of the 2007 international cross-disciplinary conference on web accessibility* (Vol. 225, pp. 35–39).

Zou, J. C. et al. (2004). The generalized Fibonacci transformations and application to image scrambling. In *Proceedings of the IEEE ICASSP* (Vol. 3, pp. 385–388).