MediaCloud: A New Paradigm of Multimedia Computing

Wen Hui[1, 2], Chuang Lin[2] and Yang Yang[1]
[1] School of Computer and Communication Engineering, University of Science and Technology Beijing
Beijing 100083, China
[2] Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China
[e-mail: hwen97@126.com, yyang@ustb.edu.cn, chlin@tsinghua.edu.cn]
*Corresponding author: Wen Hui

Received November 30, 2011; revised February 13, 2012; revised March 22, 2012; accepted April 17, 2012; published April 25, 2012

Abstract

Multimedia computing has attracted considerable attention with the rapid growth in the development and application of multimedia technology. Current studies have attempted to support the increasing resource consumption and computational overhead caused by multimedia computing. In this paper, we propose *MediaCloud*, a new multimedia computing paradigm that integrates the concept of cloud computing in handling multimedia applications and services effectively and efficiently. *MediaCloud* faces the following key challenges: heterogeneity, scalability, and multimedia Quality of Service (QoS) provisioning. To address the challenges above, first, a layered architecture of *MediaCloud*, which can provide scalable multimedia services, is presented. Then, *MediaCloud* technologies by which users can access multimedia services from different terminals anytime and anywhere with QoS provisioning are introduced. Finally, *MediaCloud* implementation and applications are presented, and media retrieval and delivery are adopted as case studies to demonstrate the feasibility of the proposed *MediaCloud* design.

Keywords: Cloud computing, multimedia computing, virtualization, service overlay, task scheduling

This research was supported by the Project 60932003 supported by the National Natural Science Foundation of China (NSFC), the Project 2010CB328105 supported by the National Basic Research Program of China (973 Program), and the Program for New Century Excellent Talents in University. We express our thanks to Dr. Wenwu Zhu and Dr. Hao Yin who checked our manuscript.

http://dx.doi.org/10.3837/tiis.2012.04.012

1. Introduction

With the explosive growth of multimedia applications over the Internet, considerable research interests have recently been directed towards multimedia computing. Multimedia computing is a technology that can generate, edit, process, and search for media content, such as images, videos, audios, and graphics, among others [1]. Multimedia computing technology has the potential to enable a large number of applications, ranging from multimedia e-mail and video players to sophisticated real-time conferencing and virtual/augmented reality. The abundance of multimedia applications and services leads to more demands, such as real-time computing and distributed processing.

Cloud computing is an emerging technology that aims to allow users to easily obtain a wide range of web-based services that previously required substantial hardware/software investments and professional skills [2]. Cloud computing technology generally incorporates a combination of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). A "Cloud" resembles a data center with huge amounts of computing power and storage, whereby a set of virtualized resources can be dynamically allocated for users on demand and consequently charged based on usage. A Cloud provides a simple and pervasive way for service providers to develop and deploy their own services and for end users to access various applications without worrying about the implementation details. However, aside from its strong computing power, a Cloud needs to provide Quality of Service (QoS) to support rich multimedia applications and services. Providing effective and efficient multimedia computing in a Cloud faces the following challenges:

(1) **Heterogeneity**. In this work, heterogeneity primarily includes device heterogeneity and network heterogeneity, among others. With the advances in third-generation (3G) and fourth-generation (4G) wireless communications, the Internet provides ubiquitous access at a very large scale, and thus, users may desire to access various multimedia services on any device anytime and anywhere. Different devices (e.g., PC, mobile phones, and PDA) and networks (e.g., wired and wireless networks) have different characteristics for processing multimedia services. Thus, adaptation capability is indispensable for accommodating various terminals and access networks.

(2) **Scalability**. Cloud-based multimedia services typically have to manage concurrently a large number of users and, consequently, deal with bursts in resource demands. Scalability is required when dealing with a large number of users. This condition implies that the resources consumed to achieve a given performance objective and the traffic coming into these services have to increase in a graceful manner with the growing number of users. Moreover, the costs of administration and maintenance should be as small as possible.

(3) **Multimedia QoS provisioning**. Multimedia applications and services differ from traditional applications in many ways, involving large amounts of data and user access, which have very specific and stringent QoS requirements. Multimedia QoS represents a set of quantitative and qualitative characteristics that are necessary to achieve the required functionality of a multimedia application. For example, latency requirement is a key issue that needs to be resolved for smooth real-time conferencing. Besides latency (delay), common QoS parameters also include delay variation (jitter), bandwidth, and so on. Current Clouds are not designed for multimedia applications with stringent QoS requirements. Thus, the QoS of such services cannot be easily achieved. Providing multimedia applications in a Cloud must support multimedia applications with QoS provisioning.

To address the aforementioned challenges, the current study proposes a new cloud-based multimedia computing paradigm, called "*MediaCloud*". More specifically, a layered architecture with good QoS control, which allows upper multimedia services to obtain physical resources efficiently and in a scalable manner, is presented. Then, *MediaCloud* technologies are presented. Using these technologies, users can access multimedia services from different terminals anytime and anywhere with QoS provisioning.

MediaCloud can solve problems in multimedia computing by taking advantage of the idea of cloud computing. Using *MediaCloud*, users can subscribe to their preferred multimedia services as needed and access these services more conveniently and efficiently. For example, when a mobile client is unable to complete media rendering tasks because of limited computing capacity and battery life, he/she can send these requests to *MediaCloud*. Then, all or most of the rendering tasks are conducted in *MediaCloud*, resulting in better performance from the perspective of the client. *MediaCloud*-based video adaptation is another example. When a mobile client requests video programs, *MediaCloud* can transform these requested videos into new versions that meet the customized parameters of the client, such as screen size and bandwidth. Moreover, *MediaCloud* also offers many advantages for service providers. Using *MediaCloud*, service providers can deploy their services through a uniform interface regardless of the implementation details.

The remainder of this paper is organized as follows. Section 2 introduces the related work on cloud-based multimedia computing. Section 3 presents the *MediaCloud* architecture. Section 4 discusses the key technologies of *MediaCloud*. Section 5 presents *MediaCloud* implementation and applications, and uses a media retrieval scenario and a media delivery scenario as case studies. Finally, concluding remarks are given in Section 6.

2. Related Work

A rapid growth on research on multimedia computing has occurred in recent years. Server-based multimedia computing performs all computations through a set of servers [3][4][5], which is restricted by deployment costs and the pressure of the backbone network. Content Delivery Network (CDN) pushes multimedia content to the edge, thereby effectively reducing communication overhead [6][7][8]. However, CDN technology still faces challenges in scalability and QoS provisioning, and CDN edge servers do not have computational capabilities. Inspired by grid computing [9], Peer-to-Peer (P2P) multimedia computing partitions multimedia computing works or workloads between peers [10][11][12], significantly improving scalability. However, QoS problem remains.

To the best of our knowledge, only few reported works on cloud-based multimedia computing are presently available. Ferretti et al. presented a cross-layer architecture that offers mobility support to wireless devices that execute multimedia applications [13]. Rings et al. proposed the integration of grid and cloud computing strategies and standards into the Next Generation Network (NGN) to support multimedia services [14]. These studies provide multimedia services in general-purpose cloud environments without QoS provisioning. When the number of users continues to scale up, the real-time performance of these studies may still be difficult to be guaranteed. Zhu et al. proposed a multimedia cloud computing concept that provides QoS and allocates cloud resources for multimedia services [15]. *MediaCloud*, as proposed in the current paper, addresses the heterogeneity, scalability, and multimedia QoS provisioning issues caused by the provision of multimedia computing in the Cloud.

3. MediaCloud Architecture

The basic idea in the design of *MediaCloud* is to process complex services with efficient resource allocation, scalability, and QoS provisioning. **Fig. 1** shows a layered view of *MediaCloud*, which is logically divided into three layers, namely, the *Media Service Layer* (*MSL*), the *Media Overlay Layer* (*MOL*), and the *Resource Management Layer* (*RML*). Thousands of physical resources that provide the "horse power" for large-scale multimedia services are widely distributed in different areas. The management of such resources is necessary for providing an appropriate runtime environment for multimedia services, and these resources need to be exploited at best. Formulating solutions on the *RML* utilizes multiple physical resources as virtual resources, allocating dedicated shares of physical resources among multiple *Virtual Resource Clusters* (*VRCs*) and consequently, ensuring effective control over heterogeneous resources. Moreover, *MediaCloud* relies on overlay technology for organizing *VRCs* to fulfill the QoS requirements of different multimedia services and achieve resource optimization. The *MOL* supports the construction of service overlay networks. In delivering multimedia services via the *MOL*, the *MSL* is needed to provide an interactive interface in which users can customize their own services.

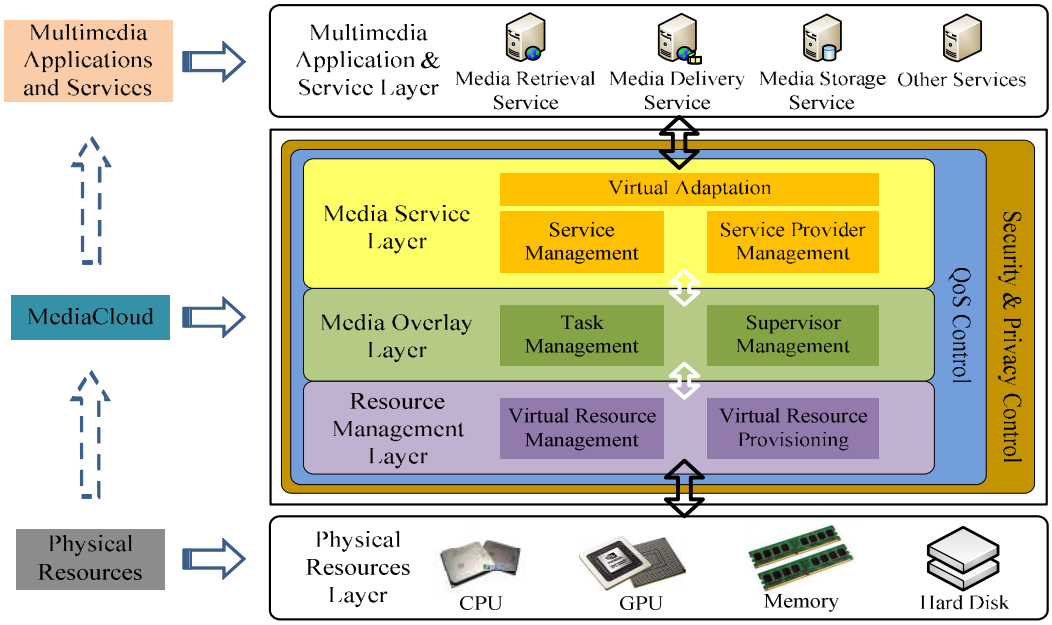

Fig. 1. *MediaCloud* architecture

a) ***Media Service Layer***: The *MSL*, which provides services that assist service providers in delivering multimedia services to end users, is the highest-level layer. These services primarily include the customization of multimedia services, service adaptation, accounting, and billing. The *MSL* represents the platform on which the multimedia applications are deployed in *MediaCloud*. The *MSL* provides components and tools that facilitate the development and deployment of multimedia applications, converting each requested multimedia service into a service task. In the *MSL*, the *Service Management* module is primarily used to negotiate QoS with users and create service tasks. The *Service Provider Management* module is responsible for configuring the client environment and charging for

the resources based on usage. The *Virtual Adaptation* module is used for adapting multimedia services for different devices.

b) ***Media Overlay Layer***: The *MOL* is the middle layer. The *MOL* constructs the *Media Service Overlay Networks* (*MSONs*) based on existing network infrastructure such that different service tasks can be assigned to different *MSONs* to fulfill the QoS requirements of different multimedia services. In the *MOL*, each service task from the *MSL* is executed by dividing each task into several subtasks to enhance the capability of *MOL* in processing multiple tasks in parallel. The *Task Management* module is used to form *MSONs* as well as to manage and schedule service tasks. The *Supervisor Management* module performs service task maintenance.

c) ***Resource Management Layer***: The *RML*, which ensures that all physical resources, such as CPU, GPU, and storage, can support the *MSONs* in the *MOL* efficiently, is the lowest-level layer. The *RML* virtualizes physical resources and constructs *VRCs*. The *Virtual Resource Management* module maps the resources needed by the *VRCs* into real physical resources, depending on the specific needs of upper service tasks. The *Virtual Resource Provisioning* module provides on-demand scheduling and physical resources to the *VRCs*. This capability facilitates the implementation of resource provisioning with different granularities in each *MSON*.

As can be seen in **Fig. 1**, *QoS Control* is implemented across the three layers. The *MSL* negotiates QoS requests with users. The QoS requirements are mapped onto QoS parameters through the further processing of multimedia services in the lower layers. The *MOL* supplies QoS support to the *MSL* by providing service overlays with QoS provisioning.

In addition, unlike desktop computing, where all the data are stored in local hard disks, the media data stored in *MediaCloud* are distributed anywhere, and thus, *Security and Privacy Control* is very important. The *RML* protects the data against network attacks. In the process of service tasks scheduling, the *MOL* ensures the security of data transmission. The *MSL* is responsible for identifying unauthorized access.

The proposed *MediaCloud* architecture has the following properties. First, the use of virtualization technologies allows the efficient use of resources. Such technologies enable multimedia applications to dynamically acquire the resources they need. Moreover, through the use of virtual resources provided by *MediaCloud*, service providers can reduce their administration and maintenance costs. Second, the overlay networks provide an effective way of supporting multimedia applications and services with QoS provisioning. Third, the service adaptation mechanism supports the adaptive service provisioning of different devices and networks.

4. MediaCloud Technologies

This section presents the key technologies of *MediaCloud* in each layer.

4.1 MediaCloud Virtualization

The maximization of limited physical resources to provide the resources needed by upper layer tasks efficiently is the core problem of the *RML*. The *RML* is designed to facilitate the aggregation of resources in an environment where services in different domains need different hardware and software configurations and are subject to different machine and network administration policies. Virtualization allows the efficient multiplexing of resources of a shared and distributed infrastructure. *Virtual Machines* (*VMs*) offer good opportunities for load balancing and fault tolerance that build upon growing support for checkpoints and live

migration of running VMs, which are interconnected through *VRCs*. Each *VRC* comprises VMs of the same level of configuration in terms of CPU, GPU, or storage. Specifically, the *RML* involves two key technologies, namely, virtual resource management and virtual resource provisioning.

Virtual resource management primarily refers to virtual resource mapping. In other words, virtual resource management technology constructs *VRCs* depending on the requirements of different service tasks while mapping the resources needed by the resulting *VRCs* into the underlying physical resources. For example, for multimedia applications related to graphics, the VMs build each *VRC* with the same level of GPU capacity. For normal multimedia processing, the VMs make *VRCs* with the same level of CPU capacity. For storage types of multimedia applications, the VMs construct *VRCs* with the same level of storage capacity. As a result, the *RML* can facilitate QoS provisioning for different types of multimedia services in *MediaCloud*. Current related studies primarily focus on resource selection using objective functions that correspond to economic benefits, such as dependence on long-term average revenue for CPU and bandwidth [16][17].

Virtual resource provisioning primarily involves reasonable scheduling and providing physical resources to *VRCs*. The related methods can be categorized into two types, namely, static [18] and dynamic provisioning [19]. The former cannot change the allocation strategy during the lifetime of the virtual clusters, whereas the latter can dynamically adapt to resource usage.

Fig. 2 shows the proposed solution for managing virtual resources and providing virtual resources in *MediaCloud*. As can be seen in **Fig. 2**, the *Virtual Resource Management* module dynamically constructs the *VRCs*. The *Virtual Resource Management* module is composed of several components, including *Information Collector*, *Management Policy Engine*, *Migration Actuator*, and *Monitoring Agents* that run inside the VMs. The *Monitoring Agents* gather the utilization information of virtual resources in each VM, thereafter submitting the information to the *Information Collector*. The *Information Collector* stores the information into the *Information Table* for the definition of related management policies, such as policies for the construction of *VRCs* and for the migration of VMs as needed. Then, the *Management Policy Engine* obtains the resource information from the *Information Table*, designs the management policies, and records these policies into the *Management Policy Table*. Subsequently, the *Management Policy Engine* triggers the *Migration Actuator* to implement the policies. More specifically, the *Migration Actuator* copes with the dynamic construction of *VRCs* and live migration of the VMs by scheduling the related *Hypervisors*. In an emergency, the *Information Collector* directly sends a trap signal to the *Management Policy Engine*.

The *Virtual Resource Provisioning* module is responsible for virtual resource provisioning. The *Virtual Resource Provisioning* module consists of the *Resource Monitor*, *Resource Abstractor*, *Provisioning Policy Engine*, and *Resource Allocator*. The *Resource Monitor* tracks the utilization of all the physical resources in real time, thereafter submitting the information to the *Resource Abstractor*. The *Resource Abstractor* analyzes the available resources and stores the analyzed results into the *Virtual Resource Table*. Based on the *Virtual Resource Table*, the *Provisioning Policy Engine* communicates with the *Virtual Resource Management* module to determine the resource requirement of each VM, promoting appropriate policies of virtual resource provisioning. The provisioning policies are saved into the *Provisioning Policy Table*. Ultimately, the *Provisioning Policy Engine* triggers the implementation of the policies through the *Resource Allocator*, which then dynamically allocates and coordinates the resources for each VM.

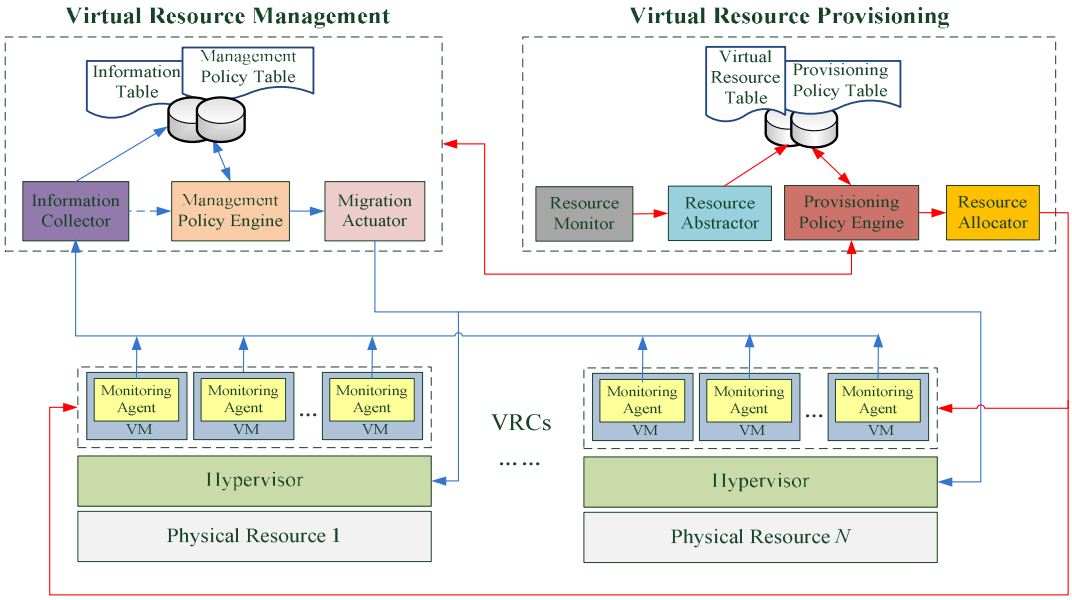

Fig. 2. Virtual resource management and virtual resource provisioning

4.2 MediaCloud Overlay

The *MediaCloud* service overlay is a key technology in the *MOL*. The *MOL* is a critical layer for satisfying the QoS requirements of each multimedia service. The *MOL* is responsible for constructing service overlay networks and processing the multimedia service tasks from the *MSL* in the service overlays.

The service overlay network can provide an effective way for addressing the QoS provisioning problem. A number of related studies on service overlay network are currently available. For example, in Service Overlay Networks (SONs) [20], the bandwidth is provisioned with certain QoS guarantees from individual network domains to build a logical end-to-end service delivery. Although SONs rely on underlying networks to provide QoS services, OverQoS [21] presents a Controlled Loss Virtual Link (CLVL) abstraction to provide Internet QoS using overlay networks and to perform bundle loss control on each virtual link. The QoS Overlay Network (QSON) [22] addresses the distributed QoS routing problem in backbone overlay networks. In the *MediaCloud* environment, *MSONs* are constructed based on the *VRCs*. The different service tasks from the *MSL* are assigned to different *VRCs* based on their QoS requirements. Moreover, service tasks are processed efficiently by partitioning them into several subtasks, such that these subtasks can be processed in parallel on the *VRC*.

Fig. 3 illustrates the formation of *MSONs* and the scheduling of service tasks on the *MSON*. The *MSON* often spans different network domains and performs service-specific data forwarding, resource management and provisioning, and QoS control functions. The underlying network domain with certain bandwidth, traffic, and other QoS guarantees provides the logical link between two *VRCs*. The QoS guarantees are specified in a bilateral SLA between the *MSON* and the network domain. The underlying network domains aggregate traffic based on the *MSON* where they belong. Thus, these domains consequently perform traffic and QoS control based on the corresponding SLAs. In the *MOL*, a special routing protocol, called the "*MediaCloud* Protocol", is used for *MSONs*. Identifying QoS-satisfied overlay paths that form *MSONs* for upper layer QoS-sensitive multimedia service tasks and

balancing the traffic load on overlay links are the primary functions of the *MediaCloud* Protocol. When a service task arrives, the *Task Management* module first uses the *Task Engine* to analyze the QoS requirement and then designs a service-specific overlay topology. The *Task Management* module then allocates appropriate resources for this overlay by communicating with the *Virtual Resource Management* module in the *RML*.

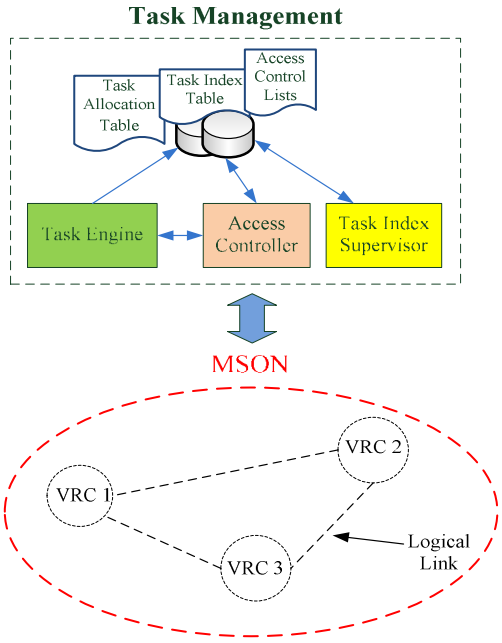

Fig. 3. Task management

While forming the *MSON*, the *Task Management* module allocates the service tasks from the *MSL* to the *VRCs* and divides each service task into several subtasks that are processed in parallel. This task allocation information is stored in the *Task Allocation Table*. For the duration of the entire process, the *Access Controller* is responsible for the security issues, which may be implemented by the distributed lock mechanism and some *Access Control Lists* (*ACLs*). In addition, the *Task Index Supervisor* builds the subtask indices depending on the subtask allocation information to locate and schedule the subtasks rapidly. This subtask index information is stored in the *Task Index Table*.

4.3 Media Service Adaptation

Media service adaptation is a key technology in the *MSL*. The *MSL* is primarily responsible for service customization and management, and it handles *MediaCloud* interaction with users. Through the *MSL*, multimedia services or applications can be supported for processing in the lower layers according to specific user preferences.

The multimedia services that *MediaCloud* provides are not only for a specific type of client, but for a family of potential clients, such as PCs, PDAs, and mobile phones. The service variability among different clients must be analyzed and modeled to make the services generic and serviceable to different users in a given domain. Moreover, effective methods for dynamically adapting services for different types of terminals are necessary [23][24].

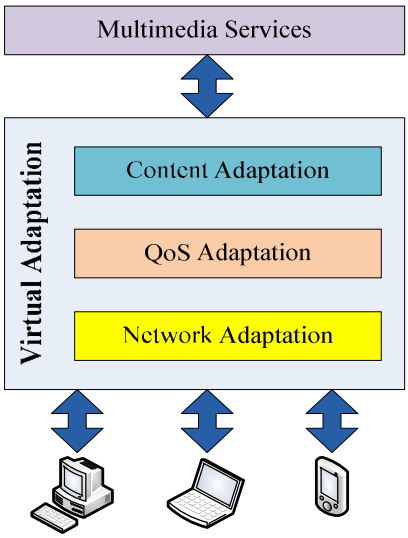

Fig. 4. Service adaptation

In *MediaCloud*, the *Virtual Adaptation* module can be divided into three submodules, namely, the *Content Adaptation*, *QoS Adaptation*, and *Network Adaptation* submodules, as shown in **Fig. 4**. The *Content Adaptation* submodule resolves content transformations based on different device capabilities. Moreover, the *Content Adaptation* submodule is usually related to mobile devices that require special handling because of their limited computational power, small screen size, and limited battery life. The *QoS Adaptation* submodule is capable of defining different QoS parameters for the media content data, as well as providing the related resource scheduling policies and QoS provisioning mechanisms. Based on the resource scheduling policies and the QoS provisioning mechanisms, the *Network Adaptation* submodule analyzes the available access modes and the related transmission requests, and then it selects the optimal transmission and network load modes for different content data.

5. MediaCloud Implementation and Applications

MediaCloud supports a variety of multimedia applications and services, such as media retrieval, storage and sharing, authoring and mashup, adaption and delivery, and media rendering [15]. This section presents the implementation of a prototype system based on *MediaCloud*. Moreover, media retrieval and media delivery applications are used to illustrate how the *MediaCloud* system outperforms the traditional architecture.

5.1 Implementation

The *MediaCloud* prototype was deployed on the ChinaCache network infrastructure[1], which comprises of 500 servers that are located within each district of China. **Fig. 5** shows the primary location of the servers for the *MediaCloud* system. In the *MediaCloud* system, the *Virtual Resource Provisioning* module monitors the network and the changes in server status, and it computes the resources consumed by taking advantage of the network coordinates approach [25]. Depending on the resource change, the *Virtual Resource Management* module adaptively organizes the resources and constructs *VRCs* using a hierarchical clustering

[1] ChinaCache is the leading provider of Internet content and application delivery services in China.

algorithm [26]. The *Task Management* module schedules service tasks to different *VRCs* and allocates resources to the tasks based on the fuzzy clustering model. The fuzzy clustering model uses the location of *VRCs* in the network coordinate space (4-dimensional vector) and the computing capacity of each *VRC* (1-dimensional vector) as properties. In this work, the computing capacity refers to the time consumed to process unit data. The objective function under the fuzzy condition is as follows:

$$J(U)=\sum_{h=1}^{H}\sum_{j=1}^{n}(u_{hj}{}^{2}\sum_{i=1}^{5}[w_i(b_{ij}-v_{ih})]^2) \tag{1}$$

where u_{hj} denotes the probability that the jth client node belongs to the hth *VRC*, w_i is the attribute weight vector, b_{ij} is the normalized feature vector of the client nodes, v_{ih} is the attribute vector of *VRCs*, H is the number of *VRCs*, and n is the number of client nodes. Then, the optimal fuzzy recognition matrix is calculated based on the Lagrange multiplier approach. Finally, the *Virtual Adaptation* module handles the media data, such as transcoding, to make the content adaptive to different devices.

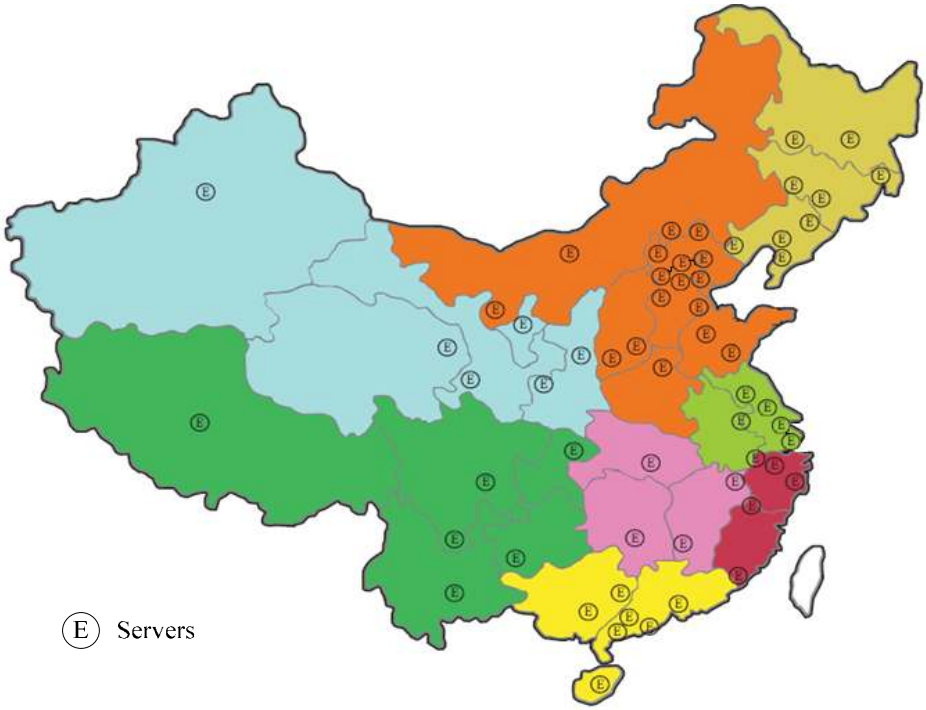

Fig. 5. Primary location of servers

Fig. 6 illustrates how the *MediaCloud* prototype system supports multimedia services.

Step 1: When a user requests a multimedia service, the *Service Provider Management* module in the *MSL* first verifies user authority. The *Service Management* module in the *MSL* then negotiates the QoS with the user and creates a multimedia service task. Moreover, the *Service Provider Management* module is further responsible for configuring the client environment and for providing security for the user.

Step 2: Based on the QoS requirements of the multimedia service task, such as the maximum delay and the minimum bandwidth, the *VRCs* are formed based on available resources. The *Task Management* module in the *MOL* schedules the service task to an appropriate *VRC* while allocating the related resources that support its execution. Then, the

service task is divided into several subtasks that are processed in parallel as required. For the duration of the entire process, the *Supervisor Management* module monitors all the subtasks and resources in real time. On the one hand, the *Supervisor Management* module handles emergencies, ensuring the reliability and availability of the execution environment. On the other hand, the *Supervisor Management* module also evaluates resource usage such that the *Service Provider Management* module can charge the user based on the evaluation results.

Step 3: The *Virtual Resource Provisioning* module in the *RML* monitors the available resources, and the *Virtual Resource Management* module communicates with the *Task Management* module to support *VRC* construction.

Step 4: After all the subtasks are completed, the *Task Management* module returns the result to the user through the *Service Management* module, and the *Service Provider Management* module charges the user for the resources based on usage.

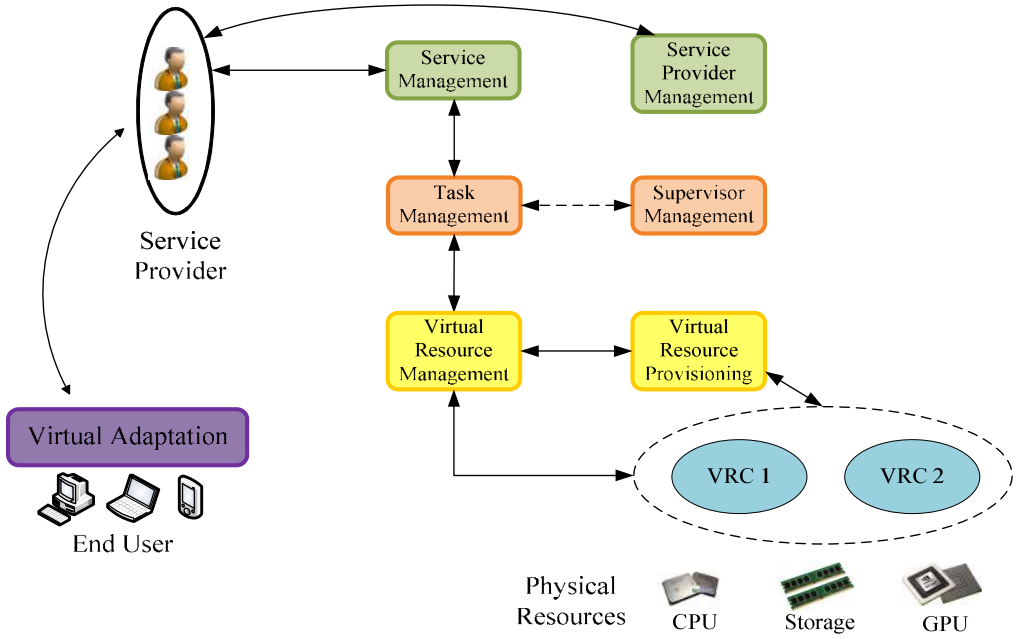

Fig. 6. Workflow of the *MediaCloud* system

Then, a media retrieval application scenario [27] and a media delivery application scenario are used to evaluate the performance of the *MediaCloud* system. The system configuration of the servers is in the following: Dell 2850 (Xeon 1.6G*2/4G/2T) and Red Hat Enterprise Linux AS 4.

5.2 Media Retrieval Application

A practical application for identifying illegal videos over the Internet is discussed in the media retrieval application scenario. Users designate to detect some media sources in the Internet. Then, the *MediaCloud* identifies pirated videos in the designated media sources by comparing the visual features of such videos with the copyrighted video feature database. Finally, the *MediaCloud* system returns the detection results to the users.

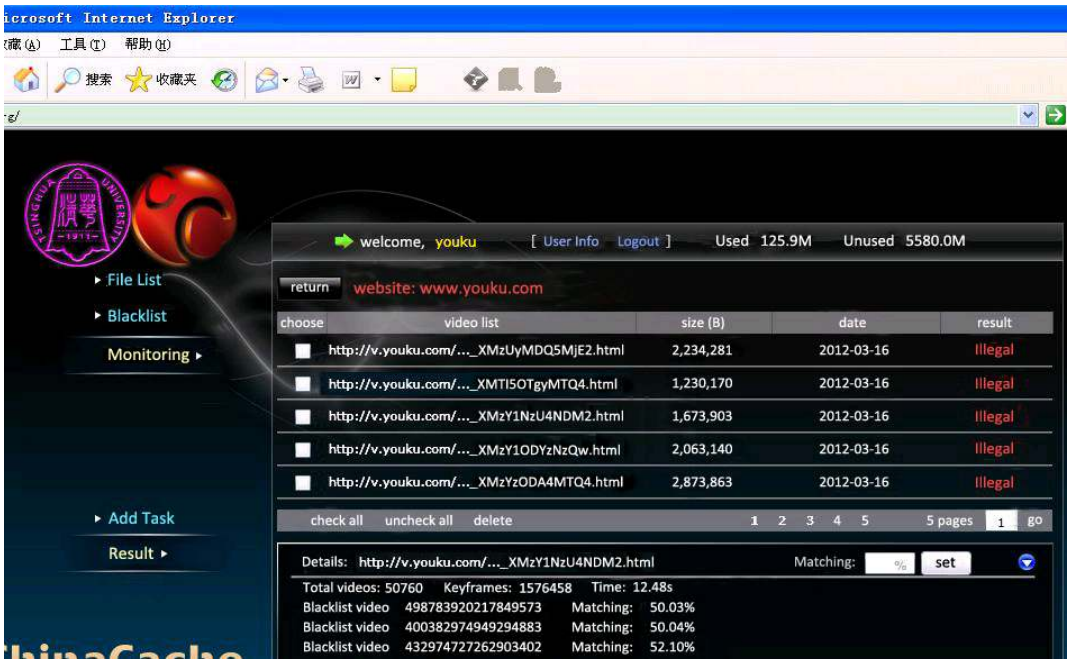

Fig. 7. PC client

Fig. 8. Mobile client

A desktop with Dell Dimension 3100 (P4 2.8G/3G/300G) running Windows XP Professional and a Nexus One running Android 2.2 were used as clients, as shown in **Fig. 7** and **Fig. 8**, respectively. Two sets of experiments were performed to evaluate how the *MediaCloud* system performs in a real environment. In the first set of experiments, the efficiency of the *MediaCloud* system in large scale for a varying number of media sources and different lengths of query videos was evaluated. In the second set of experiments, the scalability of the *MediaCloud* system was tested. A number of common video sharing websites, such as Youku.com[2] and Tudou.com[3], were detected in the experiments to test the performance of the *MediaCloud* system in a real environment. The copyrighted video feature database comprises 250,000 videos crawled from websites, such as Youku.com and Tudou.com. These videos are about 9.5 T and 83,333 hours.

5.2.1 Efficiency Evaluation

The first set of experiments evaluated the efficiency of the *MediaCloud* system [28]. First, detection time[4] with respect to the number of media sources was evaluated. In this experiment, the number of media sources under surveillance varied from 10 to 50. **Fig. 9** shows the relation between the average detection time of 50,000 queries and the number of media sources. The detection time of the *MediaCloud* system was compared with two other existing video retrieval systems, namely the hierarchical framework-based system [29] and the Locality Sensitive Hashing (LSH)-based system [30]. Let HF-S and LSH-S represent these two systems, respectively, and MC-P and MC-M represent the PC and mobile clients, respectively. As can be seen in **Fig. 9**, the proposed *MediaCloud* system has higher efficiency than the other systems for both PC and mobile clients even though the number of media sources affects the detection time to a certain degree.

Then, the detection time was evaluated with respect to video length. In this experiment, users requested the detection of 10 arbitrary media sources. **Fig. 10** shows the relation between the average detection time of 50,000 queries and the video length. The *MediaCloud* system was compared with two other systems in [29] and [30] for both PC and mobile clients. As can be seen in **Fig. 10**, the *MediaCloud* system is more efficient than the two other systems regardless of the content of the Internet videos and their sources.

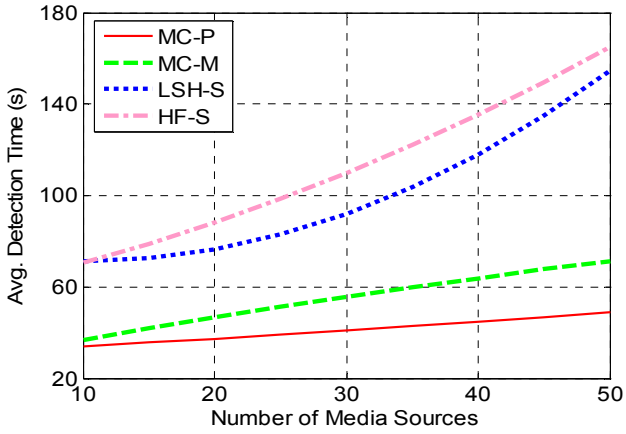

Fig. 9. Comparison of detection time with increasing number of media sources

[2] http://www.youku.com/
[3] http://www.tudou.com/
[4] Detection time refers to the time from user sending a retrieval request to when he receives the result.

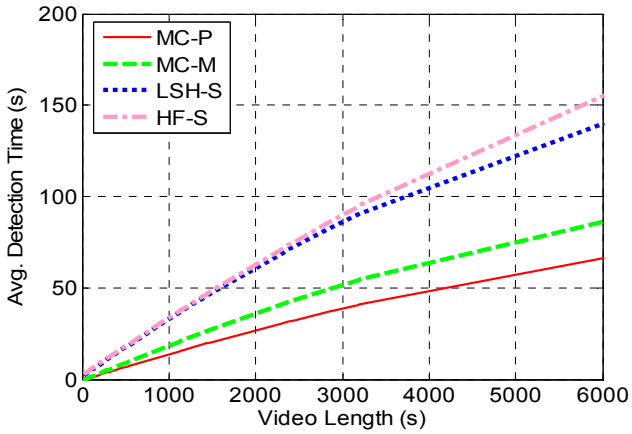

Fig. 10. Comparison of detection time with changing video length

5.2.2 Scalability Evaluation

The detection time was further examined based on a variety of videos from 10 arbitrary media sources to evaluate the scalability of the *MediaCloud* system. **Fig. 11** shows the detection time with respect to the number of user queries. As can be seen in **Fig. 11**, the *MediaCloud* system has a relatively stable detection time with increasing number of user queries, demonstrating better scalability than two other systems in [29] and [30]. Furthermore, the fluctuation of the detection time is observed. This fluctuation can be attributed to a number of factors, such as video length.

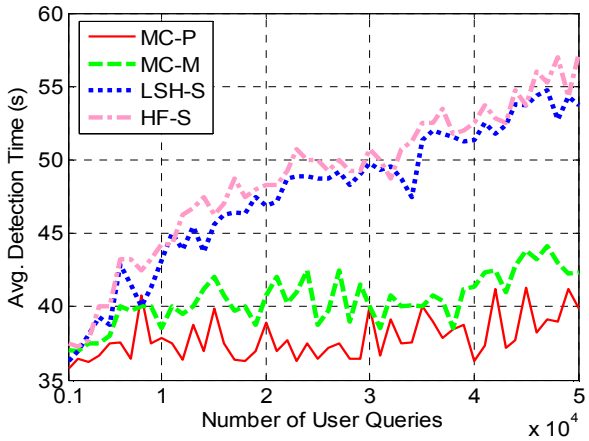

Fig. 11. Comparison of detection time with increasing user queries

5.3 Media Delivery Application

In the media delivery application scenario, users request to watch video programs. The *MediaCloud* system transforms requested videos into new versions before transmitting them to the users to meet the customized requirements of the clients.

A desktop with Dell Dimension 3100 (P4 2.8G/3G/300G) running Windows XP Professional was used as the client, and two sets of experiments were performed. In the first set of experiments, the efficiency for different video sizes was evaluated. In the second set of experiments, the scalability of the system was tested. The 10,000 test videos came from

common video sharing websites, such as Youku.com and Tudou.com. The parameters of the original videos were 640×480, 30 fps, and 1.5 Mbps. The parameters of the new version were 576×432, 25 fps, and 1.2 Mbps.

5.3.1 Efficiency Evaluation

The first set of experiments evaluated the efficiency of the *MediaCloud* system. The service delay[5] with respect to the video size was evaluated. **Fig. 12** shows the relation between the average service delay of 100 queries and the video size. The service delay of the *MediaCloud* system was compared with two other CDN-based video delivery systems, namely the network distance-based system [31] and the load-based system [32]. Let ND-S and LD-S represent these two systems, respectively, and MC-S represent the *MediaCloud* system. As can be seen in **Fig. 12**, the *MediaCloud* system has higher efficiency than the traditional CDN-based video delivery systems regardless of the video size.

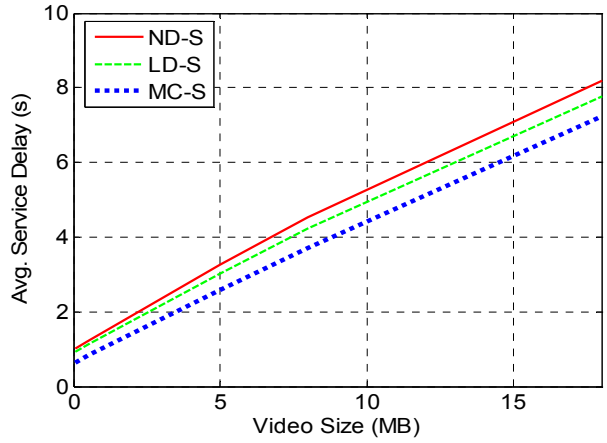

Fig. 12. Comparison of service delay with changing video size

5.3.2 Scalability Evaluation

The service delay was examined when users requested 10 MB videos to evaluate scalability. **Fig. 13** shows the service delay with respect to the number of user requests. The *MediaCloud* system was compared with the two other systems in [31] and [32]. As indicated in **Fig. 13**, the *MediaCloud* system has better scalability with increasing number of user queries even though it does not have any obvious advantage when the number of user queries is not very high.

6. Conclusion

The current study proposed the *MediaCloud* concept and framework for multimedia computing. *MediaCloud* addresses the following three key problems for the new cloud-based multimedia computing paradigm: heterogeneity, scalability, and multimedia QoS provisioning. A layered architecture for *MediaCloud* that comprises the *Resource Management Layer*, the *Media Overlay Layer*, and the *Media Service Layer* was presented. Moreover, the key technologies by which the *MediaCloud* can provide multimedia applications and services effectively and efficiently with QoS provisioning was presented. A media retrieval application and a media delivery application were used as case studies to demonstrate how *MediaCloud* can support multimedia services efficiently.

MediaCloud still faces problems that are open for future research. These problems include

[5] Service delay refers to the time from user sending a request to when he receives the video.

determining how the Quality of Experience (QoE) of multimedia services is addressed in addition to the original QoS in *MediaCloud*, outlining how the potential security threats can be dealt with, and identifying the appropriate charging model for multimedia services, among others.

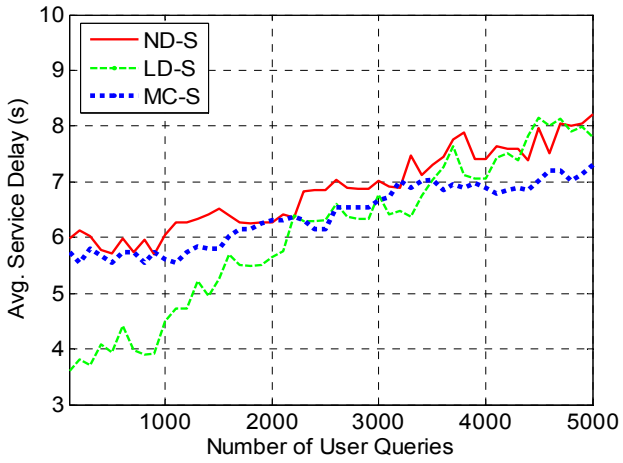

Fig. 13. Comparison of service delay with increasing user queries

References

[1] S. Reisman, *Multimedia Computing: Preparing for the 21st Century*, IDEA Group Publishing, Harrisburg, PA, 1994. Article (CrossRef Link).

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol.53, no.4, pp.50-58, Apr.2010. Article (CrossRef Link).

[3] G. Jun, "Home media center and media clients for multi-room audio and video applications," in *Proc. of 2nd IEEE Consumer Communications and Networking Conference*, pp.257-260, Jan.2005. Article (CrossRef Link).

[4] J. Nieh and S. Yang, "Measuring the multimedia performance of server-based computing," in *Proc. of 10th Int. Workshop on Network and Operating System Support for Digital Audio and Video*, pp.55-64, Jun.2000. Article (CrossRef Link).

[5] K. Lee, D. Kim, J. Kim, D. Sul and S. Ahn, "Requirements and referential software architecture for home server based inter-home multimedia collaboration services," *IEEE Transactions on Consumer Electronics*, vol.50, no.1, pp.145-150, Feb.2004. Article (CrossRef Link).

[6] N. Carlsson and D. Eager, "Server selection in large-scale video-on-demand systems," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol.6, no.1, pp.1-26, Feb.2010. Article (CrossRef Link).

[7] J. Mol, A. Bakker, J. Pouwelse, D. Epema and H. Sips, "The design and deployment of a bittorrent live video streaming solution," in *Proc. of 11th IEEE Int. Symposium on Multimedia*, pp.342-349, Dec.2009. Article (CrossRef Link).

[8] G. Fortino, C. Mastroianni and W. Russo, "Collaborative media streaming services based on CDNs," *Content Delivery Networks*, vol.9, no.3, pp.297-316, Jul.2008. Article (CrossRef Link).

[9] F. Berman, G. Fox and A. Hey, "*Grid Computing: Making the Global Infrastructure a Reality*,"John Wiley & Sons Inc, 2003. Article (CrossRef Link).

[10] C. S. Lin, "Improving the availability of scalable on-demand streams by dynamic buffering on p2p networks," *KSII Transactions on Internet and Information Systems*, vol.4, no.4, pp.491-508, Aug.2010. Article (CrossRef Link).

[11] J. Yu, H. Lee, Y. Im, M. Kim and D. Park, "Real-time classification of Internet application traffic

using a hierarchical multi-class SVM," *KSII Transactions on Internet and Information Systems*, vol.4, no.5, pp.859-876, Oct.2010. Article (CrossRef Link).
[12] L. Zhao, J. G. Luo, M. Zhang, W. J. Fu, J. Luo, Y. F. Zhang and S. Q. Yang, "Gridmedia: a practical peer-to-peer based live video streaming system," in *Proc. of 7th IEEE Workshop on Multimedia Signal Processing*, pp.1-4, Oct.005. Article (CrossRef Link).
[13] S. Ferretti, V. Ghini, F. Panzieri and E. Turrini, "Seamless support of multimedia distributed applications through a cloud," in *Proc. of 3rd IEEE Int. Conf. on Cloud Computing*, pp.548-549, Jul.2010. Article (CrossRef Link).
[14] T. Rings, G. Caryer, J. Gallop, J. Grabowski, T. Kovacikova, S. Schulz and I. Stokes-Rees, "Grid and cloud computing: opportunities for integration with the next generation network," *Journal of Grid Computing*, vol.7, no.3, pp.375-393, Aug.2009. Article (CrossRef Link).
[15] W. Zhu, C. Luo, J. Wang and S. Li, "Multimedia cloud computing," *IEEE Signal Processing Magazine*, vol.28, no.3, pp.59-69, May.2011. Article (CrossRef Link).
[16] M. Yu, Y. Yi, J. Rexford and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *ACM SIGCOMM Computer Communication Review*, vol.38, no.2, pp.17-29, Apr.2008. Article (CrossRef Link).
[17] A. Haider, R. Potter and A. Nakao, "Challenges in resource allocation in network virtualization," in *Proc. of 20th ITC Specialist Seminar*, pp.22-30, May.2009. Article (CrossRef Link).
[18] J. Lu and J. Turner, "Efficient mapping of virtual networks onto a shared substrate," *Technical Report WUCSE-2006-35*, St. Louis: Department of Computer Science and Engineering, Washington University, 2006. Article (CrossRef Link).
[19] C. Courcoubetis and R. Weber, "Economic issues in shared infrastructures," in *Proc. of 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, pp.89-96, Aug.2009. Article (CrossRef Link).
[20] Z. Duan, Z. Zhang and Y. Hou, "Service overlay networks: SLAs, QoS, and bandwidth provisioning," *IEEE/ACM Transactions on Networking*, vol.11, no.6, pp.870-883, Dec.2003. Article (CrossRef Link).
[21] L. Subramanian, I. Stoica, H. Balakrishnan and R. Katz, "OverQoS: an overlay based architecture for enhancing Internet QoS," in *Proc. of 1st USENIX Symposium on Networked Systems Design and Implementation*, pp.6-19, Mar.2004. Article (CrossRef Link).
[22] L. Lao, S. Gokhale and J. Cui, "Distributed QoS routing for backbone overlay networks," in *Proc. of Int. Conf. on Networking*, pp.1014-1025, May.2006. Article (CrossRef Link).
[23] H. R. Motahari Nezhad, B. Benatallah, A. Martens, F. Curbera and F. Casati, "Semi-automated adaptation of service interactions," in *Proc. of 16th Int. Conf. on World Wide Web*, pp.993-1002, May.2007. Article (CrossRef Link).
[24] S. Chuang and A. Chan, "Dynamic QoS adaptation for mobile middleware," *IEEE Transactions on Software Engineering*, vol.34, no.6, pp.738-752, Dec.2008. Article (CrossRef Link).
[25] N. Ball and P. Pietzuch, "Distributed content delivery using load-aware network coordinates," in *Proc. of 4th ACM Int. Conf. on Emerging Networking Experiments and Technologies*, pp.77, Dec.2008. Article (CrossRef Link).
[26] W. Hui, C. Lin and Y. Yang, "Resource-aware server selection in content delivery networks," *Journal of Beijing University of Posts and Telecommunications*, vol.35, no.3, Jun.2012. Article (CrossRef Link).
[27] H. Yin, W. Hui, H. Li, C. Lin and W. Zhu, "A novel large-scale digital forensics service platform for Internet videos," *IEEE Transactions on Multimedia*, vol.14, no.1, pp.178-186, Feb.2012. Article (CrossRef Link).
[28] MPEG Video Sub-Group, "Call for proposals on image and video signature tools," *Technical Report ISO/IEC MPEG W9216*, 2007. Article (CrossRef Link).
[29] X. Wu, A. Hauptmann and C. Ngo, "Practical elimination of near-duplicates from web video search," in *Proc. of 15th ACM Int. Conf. on Multimedia*, pp.218-227, Sep.2007. Article (CrossRef Link).
[30] W. Dong, Z. Wang, M. Charikar and K. Li, "Efficiently matching sets of features with random histograms," in *Proc. of 16th ACM Int. Conf. on Multimedia*, pp.179-188, Oct.2008. Article

(CrossRef Link).
[31] Y. Chen, J. Leblet and G. Simon, "On reducing the inter-AS traffic of box-powered CDN," in *Proc. of 18th Int. Conf. on Computer Communications and Networks*, pp.1-6, Aug.2009. Article (CrossRef Link).
[32] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang and S. Lu, "BCube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol.39, no.4, pp.63-74, Oct.2009. Article (CrossRef Link).

Wen Hui received the B.S. degree from Beijing University of Technology, China, in 2004, and the M.E. degree from Beijing University of Posts and Telecommunications, China, in 2008. She is currently a joint Ph.D. student in the School of Computer and Communication Engineering, University of Science and Technology Beijing, China, and the Department of Computer Science and Technology, Tsinghua University, China. Her research interests are multimedia computing, communications and applications.

Chuang Lin is a professor of the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He received the Ph.D. degree in Computer Science from Tsinghua University in 1994. His current research interests include computer networks, performance evaluation, network security analysis, and Petri net theory and its applications. He has published more than 300 papers in research journals and IEEE conference proceedings in these areas and has published three books. Prof. Lin is a member of ACM Council, a senior member of the IEEE and the Chinese Delegate in TC6 of IFIP. He serves as the Technical Program Vice Chair, the 10th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004); the General Chair, ACM SIGCOMM Asia workshop 2005; the Associate Editor, IEEE Transactions on Vehicular Technology; the Area Editor, Journal of Computer Networks; and the Area Editor, Journal of Parallel and Distributed Computing.

Yang Yang is a professor of the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China. His current research interests include multimedia communication, grid computing, wireless communication, image processing, and pattern recognition. He has published more than 150 papers in research journals and IEEE conference proceedings in these areas and has published two books.