Journal of Software Engineering and Applications, 2016, 9, 155-174
Published Online May 2016 in SciRes. http://www.scirp.org/journal/jsea
http://dx.doi.org/10.4236/jsea.2016.95013

An Adaptive Video-on-Demand Framework for Multimedia Cross-Platform Cloud Services

Chao-Hsien Hsieh[1], Chih-Horng Ke[2*], Chiang Lee[1]

[1]Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan
[2]Department of Information Management, Chang Jung Christian University, Taiwan
Email: chaohsienhsieh@gmail.com, *kech@mail.cjcu.edu.tw, leec@mail.ncku.edu.tw

Received 3 April 2016; accepted 8 May 2016; published 11 May 2016

Abstract

Although Video-On-Demand (VOD) has been in existence for years, its cross-platform applicability in cloud service environments is still in increasing need. In this paper, an Adaptive Video-On-Demand (AVOD) framework that is suitable for private cloud environments is proposed. Private cloud has the key advantage of satisfying the real need of both consumers and providers. Hence, demands such as reasonable benefits for provider and high quality for consumers are essential design considerations in this framework. The difficulty is that these two factors are always high in one end and low in the other, and hard to find a delicate balance. Cloud service could be an opportunity for the multimedia providers to obtain higher benefits and cost less for the consumers but with an even better quality in service. An adaptive framework for such a cloud service environment is proposed to resolve this problem. Some interesting phenomena are observed from the experimental results including CPU utilization, data reading and writing speed, memory usage, port configuration execution time, and bandwidth.

Keywords

Cloud Service, HTML5, Streaming, Video-on-Demand

1. Introduction

VIDEO distribution has been discussed for many years, especially in the applications of multimedia services. The appearance of various classes of handheld mobile devices induces the necessity of video stream that runs

[*]Corresponding author.

How to cite this paper: Hsieh, C.-H., Ke, C.-H. and Lee, C. (2016) An Adaptive Video-on-Demand Framework for Multimedia Cross-Platform Cloud Services. *Journal of Software Engineering and Applications*, **9**, 155-174.
http://dx.doi.org/10.4236/jsea.2016.95013

across different platforms. Users require not only stable and high quality videos but also online video watching without the limitations of the playing platform.

Currently, online video watching through the popular YouTube is an important trend. "YouTube Statistics" mentions that YouTube receives 4320 hours of video data every hour and supplies an extremely large volume of multimedia resources for user viewing. However, the following shortcomings are observed. First, the management mechanism of YouTube web server is ambiguous because it only helps content providers to offer their video products freely. But the content providers cannot set the price of video watching by themselves and cannot obtain any benefit from YouTube. The profit of advertisement belongs to Google Inc., not the video providers. Second, even though YouTube videos can be watched through various portable devices and operating platforms, for instance, iOS, Android OS, etc., APP or Adobe Flash Player still needs to be installed in advance. Precious system resources are always occupied by the player software. Third, YouTube does not provide personalized service, such as different priorities, authorities, or video qualities for different identities. Lastly, clients must endure advertisements embedded in the video.

YouTube is a public cloud application. Its video files are owned and operated by third-party service providers. Public cloud is advantageous due to the fact that companies can lease the appropriate amount of cloud computing services and adjust this amount flexibly according to their operational needs. However, because the entire business information (including possible astute materials) is placed on a public cloud, security and privacy threats are accordingly inevitable.

If an enterprise can afford to build a private cloud, then it can build its own cloud computing platform. Companies that can afford to build a private cloud possess the advantage of having complete management authority and, therefore, ensure high security and privacy protection, as mentioned by Subashini *et al.* [1]. If not treated well, security issues will affect the growth of business applications on cloud computing platforms.

Therefore, private cloud services not only meet the demand of higher corporate security but also deal with the service charge and the personal information protection problems. Also, it provides more suitable and better services for different platform users, different user roles, and different browsers. In the near future, regional private cloud services could draw unprecedented attention for its wide applicability and accessibility. For one example, a campus teaching information service cloud can be built to offer various services to teachers and students. Based on the paying-by-using principle, teaching materials in the cloud can be used repeatedly and supplied online to in-campus and off-campus learners. For another example, a cloud of health information consulting service can be established to provide health care education and training services for people in a community. At the same time, patients can be informed of proper health education concepts so that a better understanding of prevention and treatment through video materials can be realized. Last but not least, a government administration service cloud may be another interesting example. Each government department has its own unique feature. A private cloud environment can be built in order to integrate services. As a service to the people, internal documents and propaganda films can be made available not only for internal use of each department (for education and training) but also for information exchange with other departments. Additionally, some valuable information can be accessed through charged services to meet various needs of the people in such a private cloud environment.

The purpose of this work is to propose an adaptive Video-On-Demand (VOD) framework as an integrated application of multimedia services in a cloud environment. This system provides private cloud services without embedded software in the web page. Users can efficiently watch video streams on various devices and platforms. The following are the contributions of this work:

- Implementation of VOD services in a private cloud: to design and implement VOD services in a private cloud framework.
- Easy-to-use interface setting: to provide a simple and easy-to-operate interface for setting various system resources, such as the number of port available and content provider profit distribution ratio.
- Adaptive VOD services: to provide genuine adaptive Video-On-Demand (VOD) services, which offer customized deals, such as different rates and video viewing hours, for users of different status.
- Content sharing setting: content providers possess the right to switch on/off content sharing, no matter whether the video resources are placed virtually or in a physical machine.
- Cross-platform audio and video playback: through integrated HTML5 cross-platform, videos can be played back synchronously/asynchronously on various OS platforms.
- Multi-user performance analysis: to monitor and analyze the execution performance of multiple web pages

opened by users in playback environment of video streams using HTML5.

This paper is organized as follows. The environment and related technologies are presented in Section 2. The proposed methodology, which describes the system architecture, consumer controlling procedure, content provider controlling procedure, administrator controlling procedure, and queuing port allocation algorithm (QPAA) are described in Section 3. The case study is presented in Section 4. Then some numerical results and analyses, followed by real measurements, are shown in Section 5. Section 6 provides an overview of related works. Eventually, conclusions and future works are given in Section 7.

2. Environment and Related Technologies

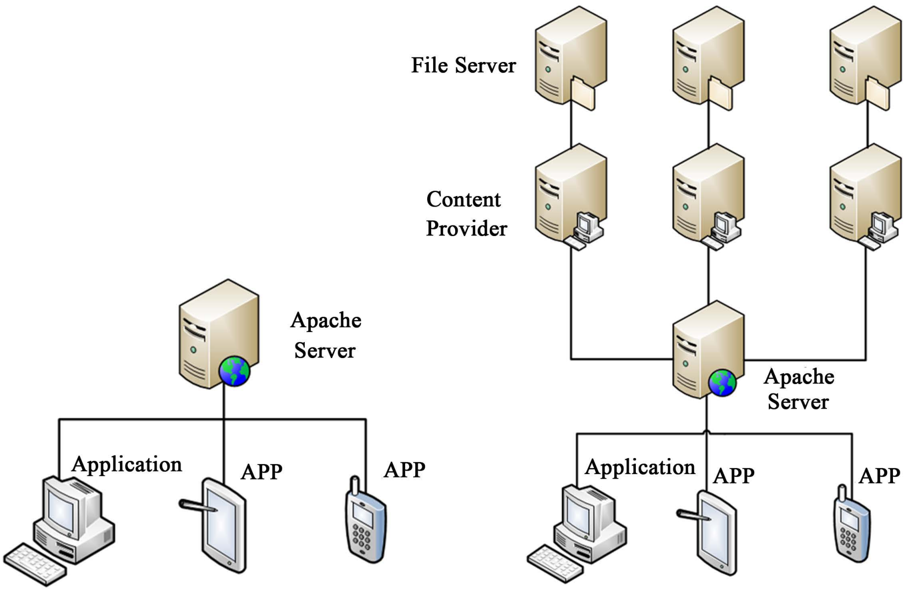
The first subsection in this category deals with the classification of a variety of video service frameworks and describes their advantages and disadvantages. At the same time, the adapted framework proposed in this study is designed. In the second subsection, the technologies used would be described.

2.1. Classifications of Framework

Recently, some practical methods for the payment mechanism of video program through web browser are represented by the following two categories. The first category mentions two cases for video stream without cloud service.

As shown in **Figure 1(a)**, the case of server-side scripting is used in website design, which includes embedding scripts into the source code of HTML. The apache server refers to the web server and is the IIS server that can install Windows. The process, starting from the client request to the server website, is managed by a script running on the server side until the client receives the server response. Structurally, it is similar to the Internet Server Architecture based on TCP Servers mentioned by Banerjee [2]. One of its disadvantages is that most of the computing resources in the server send a page back to the client for display via the server's web browser. When several clients require a server at the same time, this activity might cause the server to overload. Another drawback is that all data and software must be stored in the web server. If the web server hard disk is damaged, it will cause a one-time loss of data. In addition, the previously saved files will not be easily retrieved.

In **Figure 1(b)**, the apache server functions as an agent. It is connected to the PCs of the content providers, then establishes a link to the file servers through these PCs, as mentioned by Maly *et al.* [3] in their file sharing system diagram. As shown in the diagram, video files supplied by the content providers are stored in the file servers. In this mechanism, consumers pay for watching their favorite video files. Also, content providers can



(a)

(b)

Figure 1. Video stream without cloud service.

share their benefit with platform providing operators. Following are some flaws uncovered by this case. First, as seen in **Figure 1(a)**, when many people are online, the apache server experiences bottleneck. Second, the link between web page and video file can be easily traced to a physical machine, making this mechanism less secure.

The second category is video stream with public cloud service, as shown in **Figure 2**. This architecture diagram is similar to the equipment architecture diagram of Hu *et al.* [4]. It is also a current popular architecture for multimedia file streaming on the Internet. For instance, the server used by YouTube, etc. is based on apache and is installed with Hadoop framework. Many cloud computing systems are developed through this structure, where user device retrieves data by connecting App or application to the server. Its flaws are described as follows. First, because it plays video on a native application, which is an embedded software (e.g. flash player) in the web browser, it is more complex and much slower than web applications in accessing context information [5]. Second, the need for developing different App for different OS/devices increases developmental costs and time. Third, Watanabe *et al.* [6] questioned the security of embedded software, e.g. flash player.

Based on the merits of the three types of video stream with/without cloud service and the requirement in special environment application, the system illustrated in **Figure 3** was designed. This system possesses the following strengths: 1) Due to cloud characteristic, the node entity linked by Hadoop cannot be tracked externally;

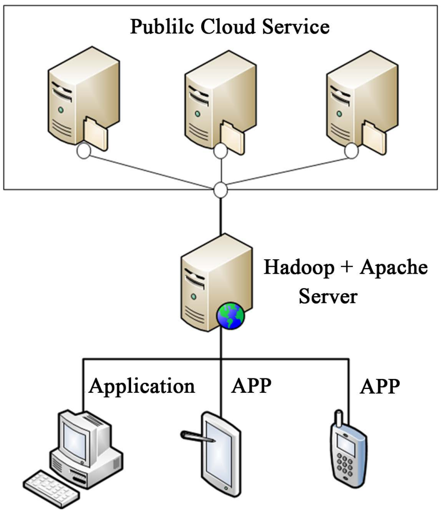


Figure 2. Video stream with public cloud service.

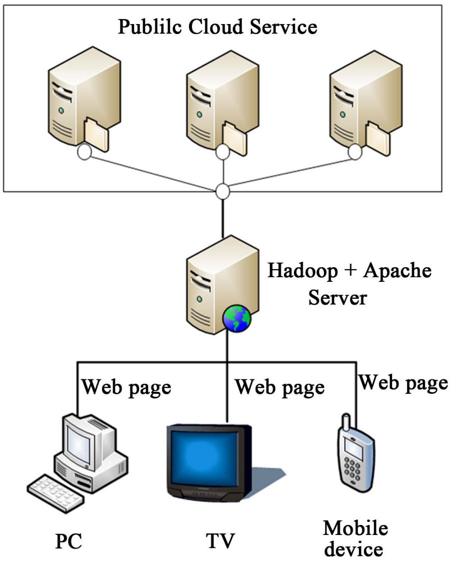


Figure 3. Video stream with private cloud service.

2) The use of HTML5 cross-platform features does not require development of different application or APP for different terminal devices using different operating systems. It only needs to be linked to the web page, opened once, and will be available for use in a variety of platforms; 3) It does not require additional software, such as adobe flash player, to view embedded video content; hence, this system gives a better performance [5].

2.2. Technologies of Framework

Some of the related technologies of the proposed frame-work are as follows:

Hadoop

Because Apache Hadoop is an open-source software for reliable, scalable, and distributed computing, many cloud service systems adopt this software as their cloud computing environment. Hadoop Distributed File System (HDFS) is also a distributed file system that can supply high throughput access to application data. Therefore, HDFS architecture is very suitable for video streaming service.

***HTML*5**

Pascual *et al.* [5] discussed the differences between web application and native application. In order to be applicable to any platform in the web application, HTML5 is adopted, developed, and implemented only once. Also, HTML5 shows a more efficient performance in playing video than running apps in Adobe Flash on mobile devices. In this paper, clients can access and play videos directly through HTML5 in any OS platform. Furthermore, it is easily compatible to different web technologies, e.g., JavaScript, PHP, and so on. Due to the aforementioned description, this technology exhibits both optimal operational performance and compatibility of programming languages.

Virtual Machine

The virtual machine is helpful to clients that do not have ample computer knowledge to build their own server and set their particular IP address. The content provider can easily port their resource content into a virtual machine. In actuality, this system can be compatible with both virtual and physical machines because of the absence of specific limitation in the cloud computing architecture. Currently, the popular virtual machine is VMWare software. The advantages of VMW are software is that it is open source and works very well with Hadoop.

3. Methodology

The system architecture of the adaptive VOD framework meets the standard three parts, *i.e.*, software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). The topmost part of **Figure 4** shows users' access of Internet service by means of a variety of devices via intermediate network. The first subsection will first discuss in detail each component function of SaaS, PaaS, and IaaS. Then, controlling procedure will be presented from the point of view of consumers, content providers, and administrators. Lastly, port allocation algorithm will be also described.

3.1. System Architecture

As a private cloud service, the Adaptive Video-On-Demand (AVOD) framework based on cloud architecture has three sections, as shown in **Figure 4**. Consumers can connect to the AVOD framework through various devices or platforms, namely A, B, and C in **Figure 4**. Each likely represents a PC, tablet PC, or smart phone. Video delivery and certification authority between client and server must go through the Internet via a web browser. First, IaaS contains a cluster data center and a physical machine. That means content providers can own their server, either a virtual or a physical machine, in IaaS. Resources, e.g., video files, can be placed and managed on virtual or physical machines. Content providers can control their machine as they wish.

Second, PaaS offers many services. The related services with their descriptions are as follows.

- Virtual machine (VM) monitor service. This service manages the current quantity and status of the VM. VM can be monitored by the administrator of AVOD. For example, when the administrator wants to know the VM port already registered in this architecture (if the VM is currently connected or already offline, or the last time the VM was connected, and the length of the VM connection time), he can look up the information in this service.
- Accounting service. This service provides movie-related billing information. For example, it calculates relevant information about personal viewing costs and expenses paid by consumers.

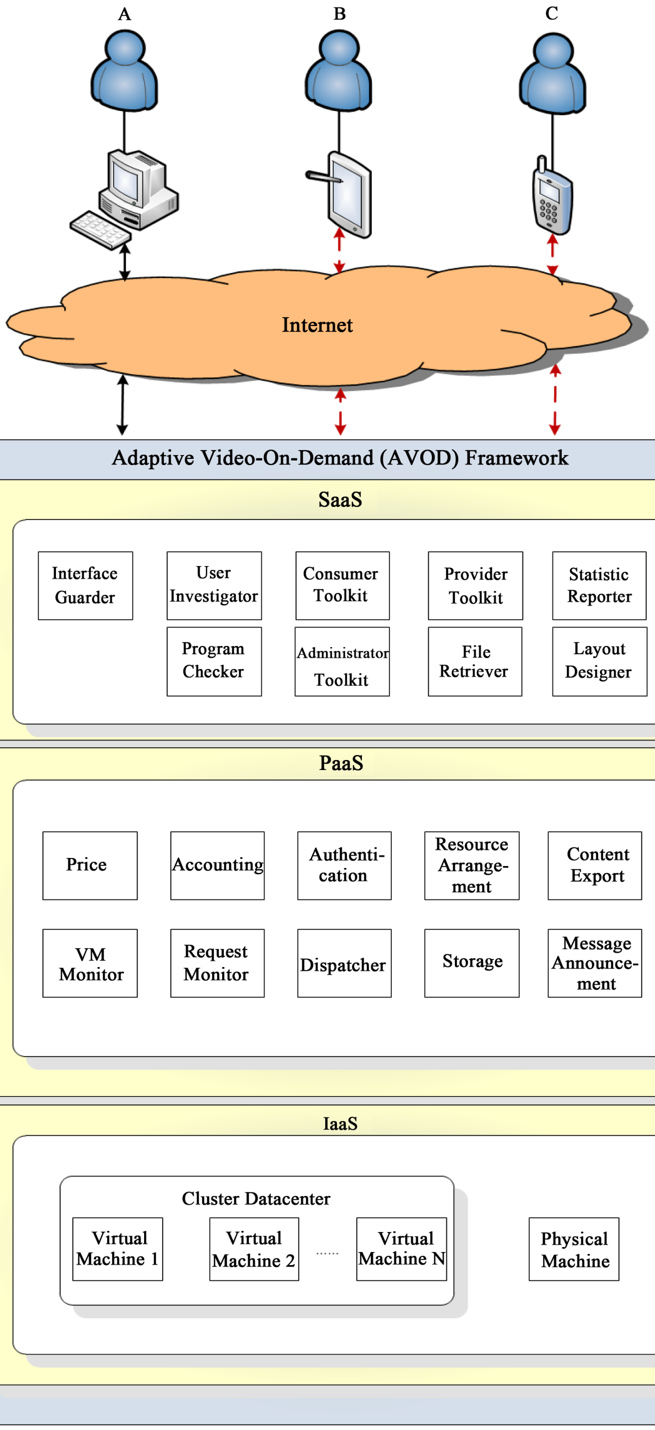


Figure 4. System architecture.

- Request monitor service. This service provides video viewing record inquiries, e.g., check movie title watched by the consumer, viewing time and history, as well as the amount spent.
- Authentication service. This service verifies consumers, administrators, and content providers, and determines different services available to different identities. This authentication service is mainly based on ID and password. Consumers can be further divided into three different account levels, namely, VIP, member, and guest. Account payment method for different levels of service varies, and details will be discussed later.

- Dispatcher service. This service assigns port number to logged-in consumer. Due to limited port numbers in different private cloud service, the dispatcher service is necessary to optimize the allocation of port numbers. Details of dynamic port management algorithm will be mentioned in Section 3.5.
- Resource arrangement service. This service provides integrated system resources and is responsible for managing the distribution of consumer demand and system resources, such as CPU, bandwidth, and memory. The difference with dispatcher service is that this service is mainly responsible for system resource allocation outside the port.
- Price service. This service displays the cost of service and the allocation of funds, including payable service fee of consumers and profits gained by content providers. As consumers are categorized as VIP, member, and guest, the following methods can be considered in designing usage rights and cost computation. For example, guests are not required to pay any user charge; therefore, when the network environment provided by the system becomes saturated, guest account that first logged in would be kicked-out, and so on, until there is no more guest account to be removed. Only then can the member account that first logged in be rejected. VIPs have absolute priority usage rights. Taking a university professor as the content provider as an example, if the total percentage of profit is 100%, after deducting platform service charges, the following proportional allocations can be assigned: personal usage (70%), department (20%), and university (10%).
- Storage service. This service provides storage of in-formation that consists of two parts: 1) video files and 2) history data. Information such as the behavior of consumers and content providers, the time that video was uploaded by the content provider, the scope of service and time limitation, accounting, price, and video files that can be read by consumers, etc. can be found here.
- Content export service. This service deals with data output. For example, to boost the film's click-through rate, managers can download a movie and make a preview clip for users to download and watch.
- Message announcement. This service deals with system reply messages. Consumers, administrators, and content providers are likely to encounter various abnormal conditions. This service displays text messages of abnormal conditions.

Lastly, SaaS provides nine service tools as described below.

- Interface guarder. This is a common login interface tool that enables cross-platform interface. User identity is classified as general consumer, content provider, and administrator. It provides a channel for easy access to other services. Shared interface guarder not only reduces the development of different operating system login screen but also distinguishes functions that can be used by different identities.
- Program checker. This tool is used to provide a list of currently available programs. In addition, it helps individual users find programs that they want to watch, e.g., top-ranking movie program according to popularity.
- User investigator. Here, users fill in their degree of satisfaction. At the same time, this tool carries out a general survey about topics such as favorite movie genre or title of particular video programs.
- Consumer toolkit. Provided here are some common consumer tools, such as personal record sheet and consumption logbook. The purpose is to enable account users to check their video file viewing records, consumption status, etc.
- Provider toolkit. Provided here are some commonly used content provider tools, such as records of video file usage and personal income status. The purpose is to provide content providers a management mechanism where they can check video files provided by individuals, record of viewed video files, and income status. Content providers can change the policy with the private cloud owner.
- Statistic reporter. The purpose of this tool is to calculate the usage status of consumers and content providers and to generate reports. These reports include daily and weekly reports provided to managers in order to regulate the number of consumers, amount of movies watched, most watched film, and the quantity of content providers and title of movies that they provided, etc.
- Layout designer. This tool enables administrators to set up an orderly layout, documentation, and preview of movies on the website, and provides template feature for movie template of users that allows them to freely select their favorite websites while watching a movie.
- Administrator toolkit. Provided here are some popular tools frequently used by the administrator. The purpose is to provide the administrator with a management mechanism. For example, 1) to obtain the current quantity and status of the VM monitor; 2) to obtain the profit status of content providers and viewing frequency of related videos; 3) to obtain consumer spending status and videos watch; 4) to publish video web page display layout.

- File retriever. This tool offers assistance to those who want to access the files. It retrieves files from the database or video file folder through the assistance of content export service. Files removed can be a single file or batch files. The administrator can read and download files according to varying restrictions; content providers can only read and download files that they uploaded; and consumers can read or download files according to their level of service.

3.2. Consumer Controlling Procedure

There are three types of users in the current system, namely, the administrator, content providers, and consumers. This subsection mainly explains the relevant procedures when consumers use this system. **Figure 5** is the consumer log in and system usage flow chart. As to the administrator and content provider flow chart, they will be described in the next two subsections.

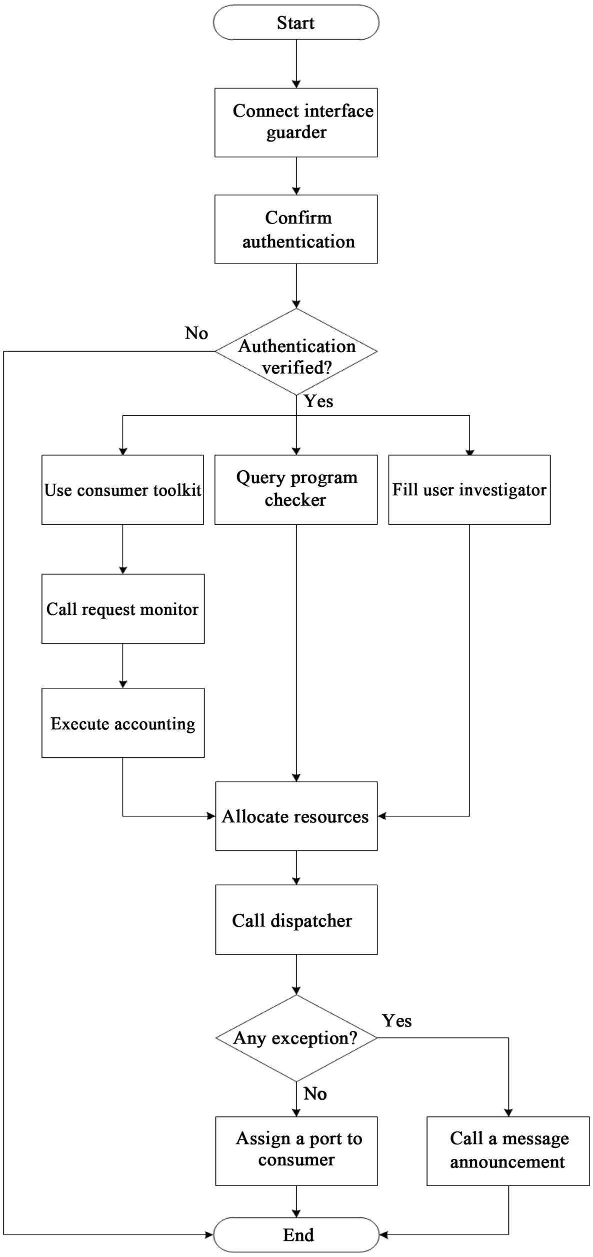


Figure 5. Flow chart of consumer.

- After the consumer is linked to the SaaS common interface guarder, authentication service will be activated and authentication status will be confirmed. The system will record the log in time of the user and web address for future analysis and understanding of user behavior.
- After confirming the authentication status of consumer's identity, the system will provide three services, namely consumer toolkit, program checker, and user investigator.
- − First, the consumer toolkit calls for a request monitor service, then a calculation function of accounting is implemented. Afterwards, the current state of consumer spending is sent back, and then resources will be ready to use.
- − Second, the program checker provides consumers, as per their inquiries, with the most currently available video programs, the most currently popular video programs, or a particular category of video programs. Afterwards, they can use the resources.
- − Lastly, the user can fill in a questionnaire with prize incentives in user investigator. The benefits are twofold: 1) to understand the needs and habits of consumers and 2) to improve customer service satisfaction. These feedback mechanisms are designed in order to make the system reflect consumer needs.
- When a consumer successfully reaches the step of "allocate resources", CPU processing, bandwidth, and memory are allocated. Then he enters into the dispatcher in order to be assigned a port number. The dispatcher will be further described in Section 4.5. At this point, if any error occurs, a message announcement will be sent back with the corresponding cause of the error; if not, then a port number will be assigned to the consumer.

3.3. Content Provider Controlling Procedure

Figure 6 discusses the flow chart of content provider.

- Connect interface guarder and authentication follow the same procedure as described in the previous subsection until authentication status confirmation.
- After authentication status confirmation is completed, it goes directly to provider toolkit, and will then be asked whether to upload files or disable them.
- − If ready to upload file

  i. Start and enter into personal VM monitor operation mode.

  ii. Then open the directory folder of the content provider and select the video files to be uploaded.
- − If not ready to upload file and will simultaneously disable file

  i. Start and enter into personal VM monitor operation mode.

  ii. Then disable video files so that it will no longer be available for use.
- − If not ready to upload file, at the same time not ready to disable file

  i. Enter directly into price service.

Finally, regardless of any of the above situations, the consumer can enter into price service to check personal profit and video files viewed.

3.4. Administrator Controlling Procedure

Figure 7 presents the administrator flow chart.

- The state verification of the interface guarder and authentication are the same as in the previous section.
- After confirmation of the authentication status, the system will provide four tools to be used, namely, statistic reporter, layout designer, administrator toolkit, and file retriever.
- First, the statistic reporter will automatically make a report of all documents. For example, it presents the state of system usage in the form of daily or weekly reports. For tracking and analysis of the administrator, the state of spending and profits of consumers and content providers and the state of the uploaded videos of content providers are included in the reports.
- Second, the layout designer helps administrators to rearrange the order of videos on the web interface so that consumers can easily access popular video files in order to increase the click-through rate. Also, a self-modifying web page layout can be provided to administrators in order to highlight upcoming events and make flexible changes to the page layout.
- Third, the administrator toolkit is used to manage the VM monitor in order to examine whether the VM usage state is normal, and also monitor the content provider's income and consumer spending.

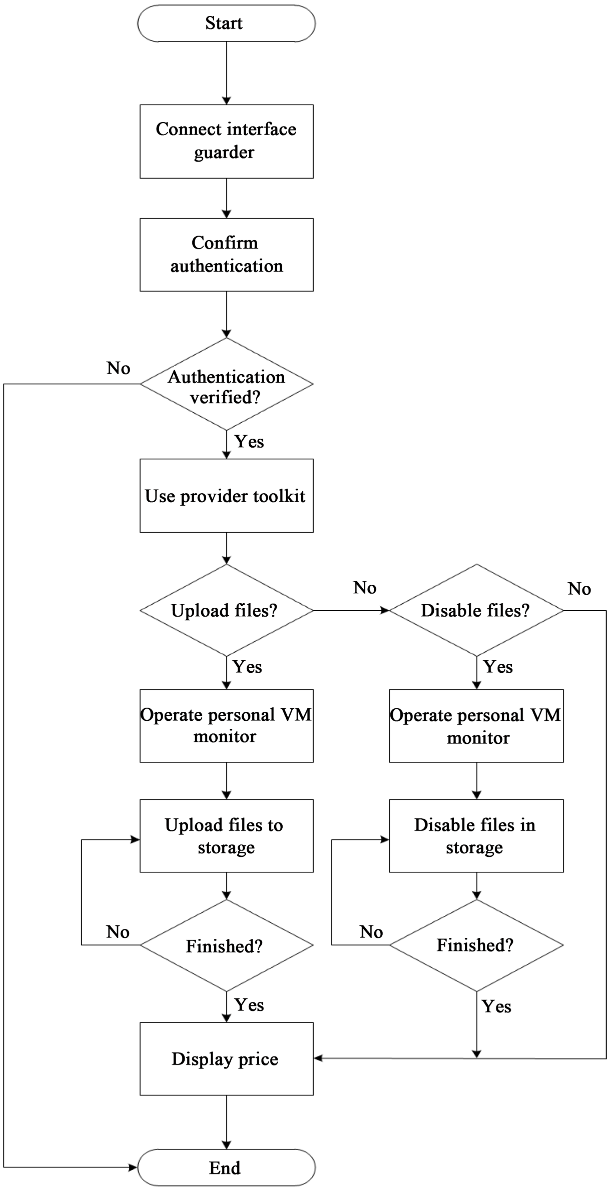


Figure 6. Flow chart of content provider.

- Lastly, the file retriever helps administrators to obtain and process files in batches. It sends the content export service to output the needed file. Then, administrators can easily download the file and upload an updated file. “e.g.” means “for example”.

3.5. Queuing Port Allocation Algorithm (QPAA)

In this framework, the port pool is set by the administrator when the environment parameters of the system are firstly initialized. Then, no one is allowed to receive service when the number of port pool is full. Here, the system accepts three identifications, VIP, member, and guest. VIP receives top priority and costs more than a member. Guests are not charged and have the lowest priority. In addition, port allocation is important for this system. The flow chart of port management is shown in **Figure 8**. After the web page is connected, the registered client is logged in either as VIP, member, or guest. Because of the local storage of HTML5, the client only registers once whether the web browser is opened or closed. The record is always valid until operating system is reset. The following section mentions 2 cases, namely, a new web page and an existing web page.

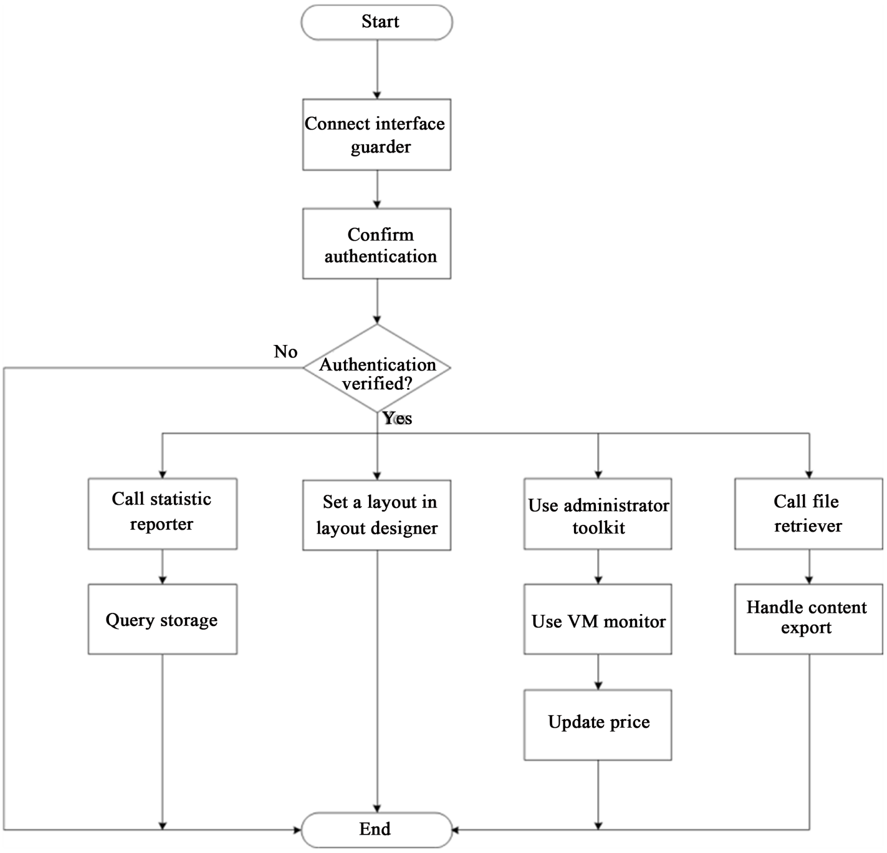


Figure 7. Flow chart of administrator.

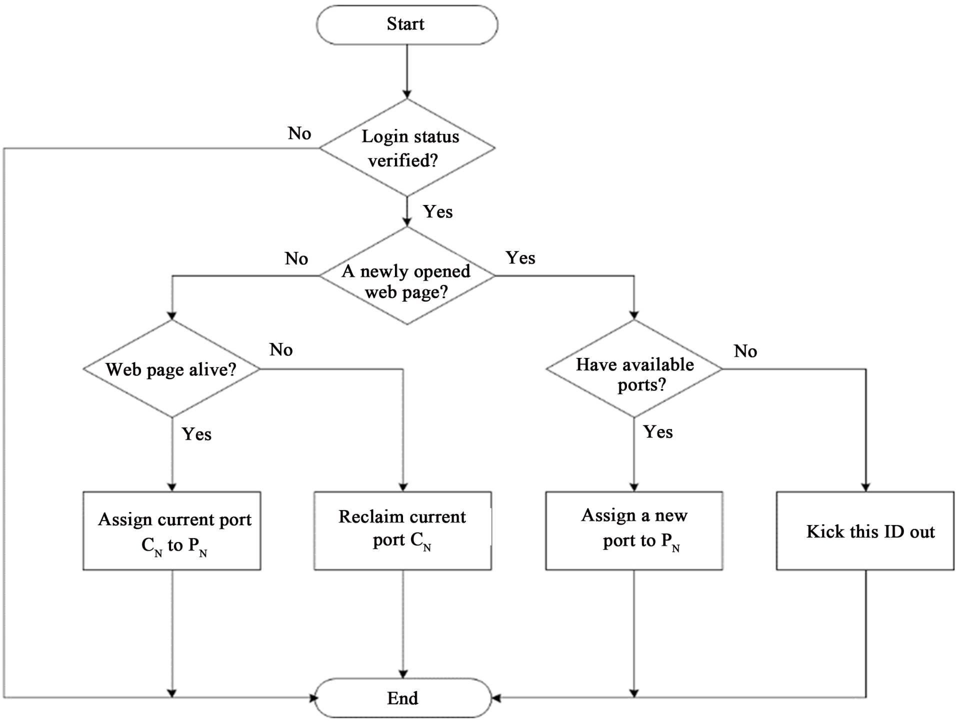
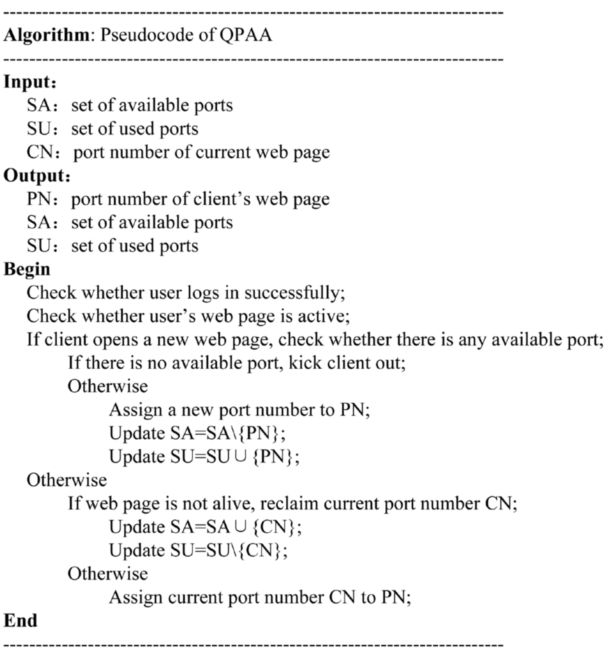


Figure 8. Flow chart of port management.

- If there is already an existing web page, check if the current web page is alive. "Alive" means it is an open website and that users continue to operate on it, such as moving the mouse or the keyboard to enter data.
  - If it is alive, then assign current port CN to PN.
  - If it is not alive, then reclaim current port CN.
- If it is a newly opened web page, check if the current system still has usable ports.
  - If at least one port is usable, then assign a port to PN.
  - If there is no usable port, then do not assign a port. Also, do not allow a new comer to obtain service. Afterwards, remove the applicant ID.

The QPAA algorithm is described as follows. There are three input parameters: set of available ports, set of used ports, and port number of current web page. After QPAA algorithm processing, three output parameters will be generated: set of available ports, set of used ports, and port number of client's web page.



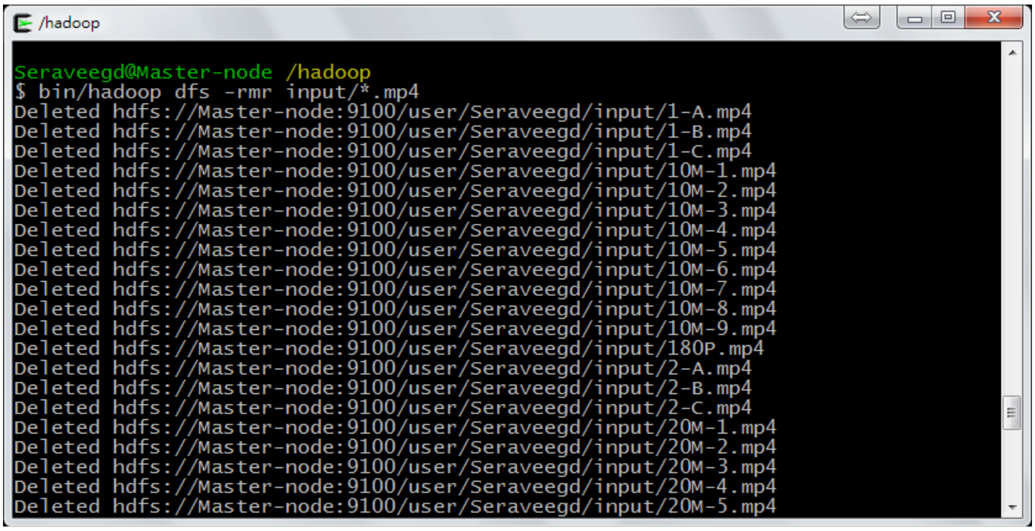
```
------------------------------------------------------------------------------
Algorithm: Pseudocode of QPAA
------------------------------------------------------------------------------
Input:
    SA： set of available ports
    SU： set of used ports
    CN： port number of current web page
Output:
    PN： port number of client's web page
    SA： set of available ports
    SU： set of used ports
Begin
    Check whether user logs in successfully;
    Check whether user's web page is active;
    If client opens a new web page, check whether there is any available port;
        If there is no available port, kick client out;
        Otherwise
            Assign a new port number to PN;
            Update SA=SA\{PN};
            Update SU=SU ∪ {PN};
    Otherwise
        If web page is not alive, reclaim current port number CN;
            Update SA=SA ∪ {CN};
            Update SU=SU\{CN};
        Otherwise
            Assign current port number CN to PN;
End
------------------------------------------------------------------------------
```

4. Case Study

This case study investigates the following applied situation. After some content providers uploaded video files to the Hadoop Distributed File System (HDFS), users can introduce videos to other people for their non-synchronized viewing and entrance into this system. Also, content providers can regulate whether their machine would be either open or close. For this system, VM can be dynamically increased or decreased through the opening or closing of VM by content providers. Then, a list of streaming programs will be updated automatically in the web page. On the client side, a new web page is opened when he picks an interesting video program. On the contrary, the web page ID is deleted directly by the system when the client closes the video stream web page. Then, the system reclaims the extra unproductive resources and then reuses them, especially port numbers and memory buffers.

In order to have a great system portability, this work is done in the Windows environment. As a server administrator, Cygwin software helps input shell command that increases and deletes Hadoop HDFS for video files when content providers do not have their own PC and file storage. Administrators can support this kind of situation. The detailed operation in shell command is shown in **Figure 9**.

Here, related information about curricula taught by professors is used as an example. The professor offers personalized video service and sets the profit distribution ratio. He can manage various resources on individual VM nodes, such as uploading, enable or remove personal recorded videos, and view current profit status. It is assumed that the professor gathers precious lecture materials for students in his class, students in school and out of school. Then, this system can provide an environment with a payment mechanism for non-students in class. Also, this mechanism can automatically allocate profit to the school, department, and professor. This work has a



```
Seraveegd@Master-node /hadoop
$ bin/hadoop dfs -rmr input/*.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/1-A.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/1-B.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/1-C.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/10M-1.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/10M-2.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/10M-3.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/10M-4.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/10M-5.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/10M-6.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/10M-7.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/10M-8.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/10M-9.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/180P.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/2-A.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/2-B.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/2-C.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/20M-1.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/20M-2.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/20M-3.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/20M-4.mp4
Deleted hdfs://Master-node:9100/user/Seraveegd/input/20M-5.mp4
```

Figure 9. Managing video files through administrator.

management interface as shown in **Figure 10**. There are two setting port parameters: one is the starting port number, which refers to the port number at the start of open access; another is the ending port number, which indicates the port number at the end. For example, in **Figure 10**, there are a total of 35 open ports, with 8081 as the starting port number and 8116 as the ending port number. The administrator can set a basic port number for VIP, member, and guest. The fee for VIPs and members are different. The basic port number mentioned here is the main port number reserved for each type of log-in user. This prevents cases of a user not being able to log in after another type of user finishes using the port number. The final set of parameters sets a bonus distribution for the three roles: professor, department, and school. The present case study is mainly directed for university environment, but it can be revised for application in other environment. This work is easy and available for modification and use in other fields and for more than three roles. Based on the above parameters, the administrator can easily update them through web GUI.

Every client, whether he has VIP, member, or guest status, can still access the main web page (common interface guarder) in **Figure 11**. Explanation will be given below according to the status with which the client is logged in. Taking the student Mr. Chang as an example, if his status is VIP, then his ID will be vip1. After he successfully logs in, vip1 will be the identity of this client in **Figure 12**. At this point, it is regarded that authentication is completed. Mr. Chang can choose from the following different actions: 1) enter the query list, choose the video program produced by Professor Lee, either in 1-A, 1-B, 1-C, 2-A, 2-B, 2-C, 3-A, 3-B, or 3-C in **Figure 11**; 2) enter the feedback page and conduct a survey; 3) propose request monitor to check out individual consumption status in order to know individual consumer behavior. Moreover, operating procedures for members and guests are similar to VIPs. The difference lies in priority and user bandwidth limitation.

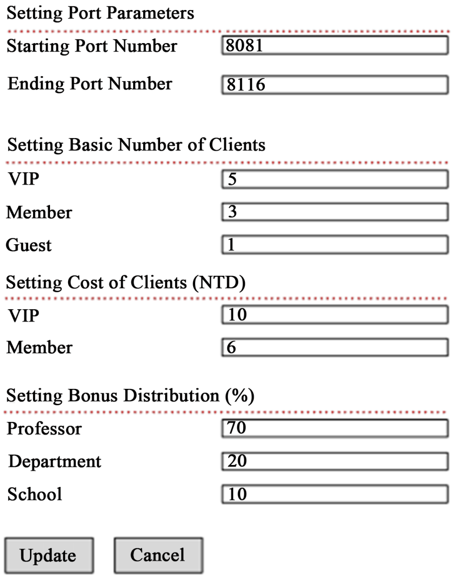
Finally, as the administrators, content providers, and consumers use this system, it has been discovered that in the process of video streaming, people are mostly concerned about CPU usage, read speed, write speed, memory usage, execution time of port allocation, and bandwidth issues. Under this framework, what is the tolerant maximum allowable range? These relevant data give system operators an assessment of the necessary system environment and cost computation. Section 5 will discuss the results of this experiment.

5. Measurements

This section represents the experimental setup and results.

5.1. Experimental Setup

The environment is a fixed IP address, a server that uses Hadoop v1.2.0 with Java JDK v7u21. This server connects three different content providers individually. Each content provider is set to three distinct video files, which must be compatible with HTML5 video format, like mp4, ogg, H.264, and webm. Mp4 files are used in this work and have the same 320 × 180 resolution while streaming over 20 machines at the same time.



Setting Port Parameters

Starting Port Number	8081
Ending Port Number	8116

Setting Basic Number of Clients

VIP	5
Member	3
Guest	1

Setting Cost of Clients (NTD)

VIP	10
Member	6

Setting Bonus Distribution (%)

Professor	70
Department	20
School	10

Figure 10. Interface of management.

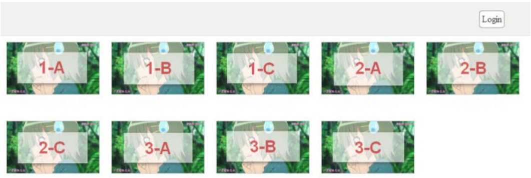


Figure 11. Main web page.

Figure 12. Video streaming in client.

The hardware configuration of the server is shown in **Table 1**. The server contains cloud service using Apache Hadoop. Also, **Table 2** describes the hardware configuration of the client as experimental machines.

5.2. Experimental Results

In order to understand the relationships among different numbers of clients with different numbers of video streaming, there are six items of concern: CPU usage, read speed, write speed, memory usage, execution time of port allocation, and bandwidth. The list of system parameters in **Table 3** comes from real clients with video through HTML5 web page. For instance, 1 video and 1 client of **Table 3** has six parameters: first, CPU usage is 0.51%; read speed is 4.51 k per second and write speed is 635 bytes per second; memory usage is 10 m; execution time of port allocation is 20 ms; and lastly, bandwidth consumed is 94.6 k. In another example, 3 videos and 10 clients of **Table 3**, the parameters are as follows: CPU usage is 2.25%; read speed is 28.6 k per second; write speed is 19.5 k per second; memory usage is 30m; execution time of port allocation is 24 ms; and bandwidth consumed is 2.37 m.

Table 1. Hardware configuration ofthe server.

Name	Description
CPU	AMD Phenom™ 9650 Quad-Core Processor 2.30 GHz
Memory	DDR2 800 HMz 4.00 GB
OS	Window 7 Pro
Network Card	Gigabit Ethernet excessive

Table 2. Hardware configuration of the client.

Name	Description
CPU	2.7 GHz quad-core Intel Core i5 processor
Memory	8 GB (two 4 GB) of 1600 MHz DDR3 memory
OS	OS X Mountain Lion
Network Card	Gigabit Ethernet

Table 3. List of system parameters.

Name	1 client	5 clients	10 clients	15 clients	20 clients
1 video	0.51% 4.51 k 635B 10 m 20 ms 94.6 k	0.54% 4.56 k 1.49 k 10 m 20 ms 364 k	1.21% 8.67 k 3.54 k 10 m 21 ms 784 k	1.39% 16.7 k 7.04 k 10 m 21 ms 1.16 m	2.88% 33 k 20 k 30 m 19 ms 1.04 m
2 videos	0.52% 5.23 k 750B 10 m 20 ms 178 k	0.89% 7.27 k 2.86 k 10 m 21 ms 887 k	0.95% 19.5 k 16.4 k 10 m 25 ms 1.33 m	2.12% 30.6 k 20.9 k 70 m 24 ms 2.06 m	2.34% 41.4 k 24.2 k 50 m 48 ms 2.6 m
3 videos	0.92% 5.63 k 1.01 k 10 m 20 ms 235 k	2.11% 11.1 k 7.51 k 10 m 22 ms 1.03 m	2.25% 28.6 k 19.5 k 30 m 24 ms 2.37 m	3.45% 55.4 k 38.1 k 80 m 61 ms 4.15 m	4.45% 76.7 k 44.6 k 50 m 604 ms 3.72 m
4 videos	0.95% 4.61 k 1.4 k 10 m 20 ms 333 k	3.58% 19.3 k 9.16 k 30 m 25 ms 1.63 m	3.78% 52.8 k 33.4 k 40 m 34 ms 2.88 m	4.2% 67.8 k 64.4 k 80 m 603 ms 4.40 m	5.01% 96.7 k 74.5 k 10 m 1.81 s 4.34 m
5 videos	0.98% 5.63 k 1.83 k 10 m 21 ms **369 k**	3.81% 40.8 k 44.1 k 50 m 23 ms **2.03 m**	3.97% 83.3 k 65.4 k 100 m 60 ms **4.6 m**	5.12% 86 k 72 k 30 m 1.74s **4.01 m**	5.58% 129 k 104 k 40 m 4.13 s **3.92 m**

To summarize the six system parameters for video stream based on **Table 3**, the detailed analysis is represented in **Figures 13-19**.

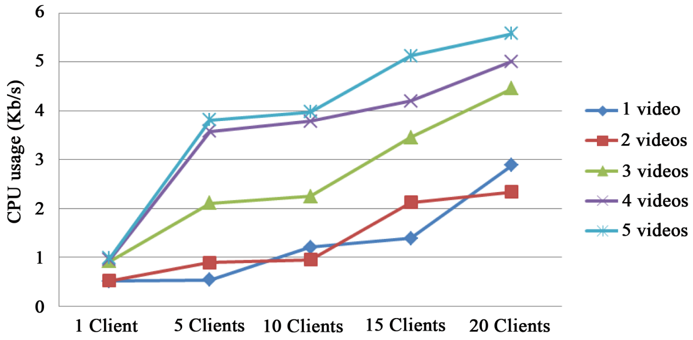


Figure 13. CPU usage of video stream.

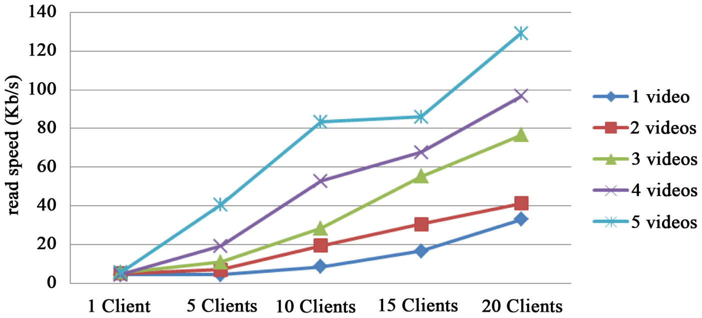


Figure 14. Read speed of video stream.

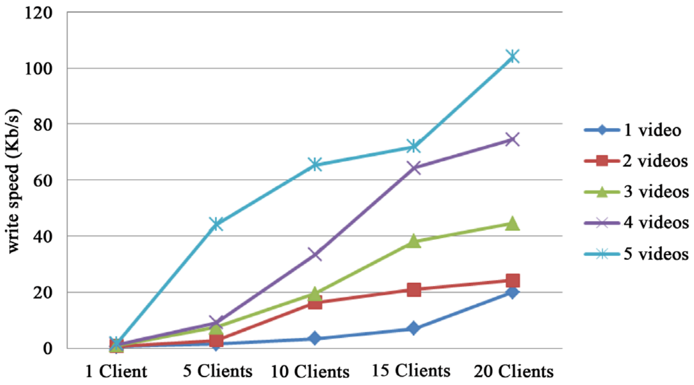


Figure 15. Write speed of video stream.

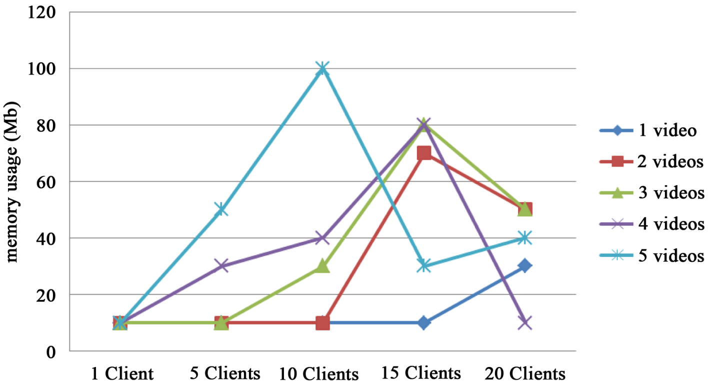


Figure 16. Memory usage of video stream.

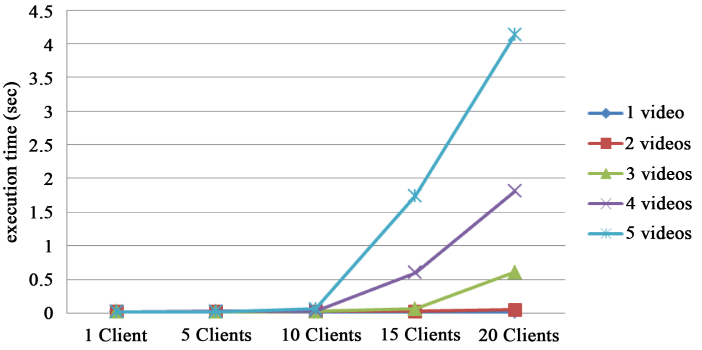


Figure 17. Time of port allocation of video stream.

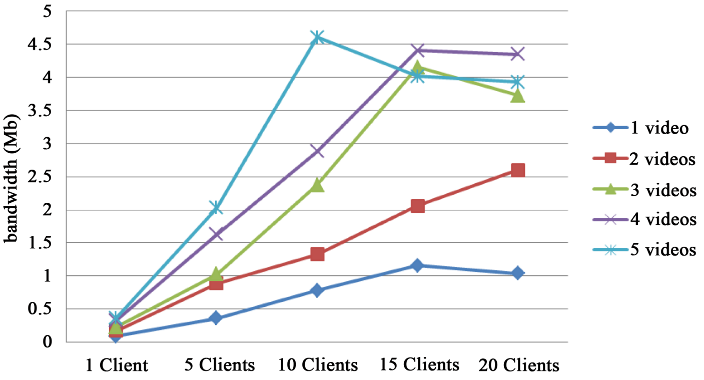


Figure 18. Bandwidth of video stream.

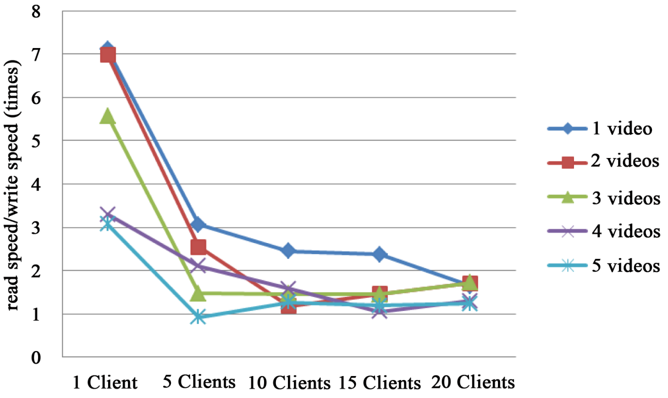


Figure 19. The comparison of read speed and write speed.

In **Figure 13**, the vertical axis represents CPU usage and the horizontal axis represents the number of clients. As the number of clients increases, CPU usage also displays a clear increase. As illustrated, when there are 20 clients, CPU usage during video stream is at most less than but near to 6%. Also, the gap between the highest and the lowest value does not exceed 4%. Taking 5 clients as an example, the highest CPU usage is 3.81 for 5 video files, the lowest is 0.53 for 1 video file. Furthermore, when client number reaches 20, the highest CPU usage is 5.58 for 5 video files, with lowest at 2.34 for 2 video files. This means that when the framework uses HTML5 to play videos, scalable well for many clients can be used because its maximum CPU usage is not high.

Figure 14 shows the situation of reading video file from server to client. The vertical axis represents read speed and the horizontal axis represents the number of clients. The read speed escalated as the number of clients increases. Then, as the number of clients and video files increase, a rapid proportional growth is observed and the reading of video files takes more Kb/s.

Analogous to **Figure 14**, the write speed in **Figure 15** has the similar trend as read speed does. Obviously, write speed is always slower than read speed at the same environment. In this figure, write speed focuses on updating verification code in the server for each client.

As depicted in **Figure 16**, the vertical axis represents memory usage while the horizontal axis represents the number of clients. Memory usage in 10 clients with 5 video files reaches 100 Mb. Then some web pages are suspended because the upper bound of the bandwidth is approached by more requests from new clients. As a result, the bandwidth goes back to a lower number. This is the reason why memory usage goes down after arriving at the maximum 100 Mb.

Time of port allocation is represented in **Figure 17**. The vertical axis represents execution time of port allocation while the horizontal axis represents the number of clients. Some abnormal time were detected. When there are 4 or 5 videos, abnormalities were observed when the number of clients is greater than 10; when there are 3 videos, abnormalities are detected when the number of clients is greater than 15. In the **Figure 17**, when there are less than 10 clients, regardless of the number of videos (1 - 5), execution time is generally stable. However, in a state where there are more than 10 clients, after five videos, the execution time starts to rise. Therefore, 5 videos with 10 clients is the steady state upper limit.

As can be seen from **Figure 17**, when there are more than 3 video files, the execution time starts to change significantly as the number of clients exceeds a certain value. Taking 3 video files as an example, when the number of clients reaches 15 and 20, execution time is 0.061 and 0.604, respectively, with a nearly tenfold difference; but when there are 4 video files, and number of clients is 10 and 15, execution time is 0.034 and 0.603, respectively, with a nearly eighteen fold difference.

Bandwidth is important to this work. In **Figure 18**, the vertical axis represents the bandwidth and the horizontal axis represents the number of clients. For 5 videos, having more than 10 clients will reach the bandwidth's upper limit; also, when there are 3 or 4 videos, having more than 15 clients will also reach the bandwidth's upper limit; the bandwidth boundary will not be reached with 1 or 2 videos. When bandwidth boundary is reached, the succeeding bandwidth drops because some web pages stop streaming video files.

Video files that stopped streaming will cause these video streaming web pages to time out, for instance, 20 clients with 3 video files, 20 clients with 4 video files, 20 clients with 5 video files, and 15 clients with 5 video files. When bandwidth is almost depleted while playing multi-videos, obviously the size of bandwidth shrinks because of automatic disconnection of the web page. The range of bandwidth is between 3.9 Mb and 4.3 Mb. This is called an "Escape Band". Often, when network bandwidth exceeds 3.9 Mb, web pages time out as they enter this bandwidth. Bandwidths between 3.9 Mb and 4.3 Mb depend upon stable condition of network at that time.

The bandwidth out of the Escape Band can assist in tracking the combination of number of clients and video files. Through monitoring software, bandwidth reduction can be observed.

Figure 19 presents the comparison of read speed and write speed. When the number of clients is greater than 5, the read speed/write speed ratio rapidly decreases. As long as the number of clients increases to 20 clients or more, the read speed/write speed ratio becomes nearly the same regardless of the number of video streaming. So, when running several clients, read speed and write speed will be slowed down. Both read speed and write speed are closely related.

6. Related Works

Currently, the related works can be set in the following categories: Web TV and TV cloud service. In this section, these categories are described as follows.

Web TV

When sharing HTML5 applications for PC, portable devices, digital TV, etc., Fukai *et al.* [7] provided their study of GUI for TV based on web browser technology. This is a relatively new technology, with limited available references. Comparing this paper to that of Fukai, Fukai focuses on personalized DTV, presented by way of web applications [as shown in the architecture of **Figure 1(a)**], with no service mechanism under cloud architecture nor role of content provider and video upload related functions. On the contrary, our work integrated the HTML5 features, its cloud architecture provides more services and is integrated in malleable environment (as shown in the architecture of **Figure 3**). Meanwhile, the functionality required for content providers is supplied so that they do not need to deal with the hardware and software environment. They simply need to focus on providing good video files.

Cloud service for TV

The Ota, Kubota, and Gotoch [8] mention famous video sharing sites such as YouTube, Netflix, and so on. Ota *et al.* built media cloud services on PaaS and provided various end devices to obtain service [8]. Because of the popularity of end devices for consumers, especially smartphone and tablet PC, the demand on video content is also relatively important. Therefore, in the case of PaaS, media cloud services provide media processing, especially in terms of video contents.

VOD service is also based on video content. As a dynamic requirement, video content is moved from available resources to user location under "cloud-based" VOD service [9]. Furthermore, Goncalves *et al.* [10] proposed VOD use-case to define its policies as part of Service Level Agreements (SLA). These policies for elastic resource provision show the video content to interested consumers through cloud services, especially dynamic resource management. Wang *et al.* [11] demonstrate adaptive mobile video streaming framework that provides efficient video streaming services in cloud environments but do not mention how to handle dynamic management for multiple service providers and VOD. Even though some researchers proposed a new framework under the situation of video traffic over mobile networks in the cloud, they did not discuss VOD on asynchronous transmission cross-platform without any extra software embedded in web browser. Our paper not only handles dynamic management for multiple service providers and VOD, but also VOD on asynchronous transmission cross-platform without any extra software embedded in web browser. A cloud computing framework for simulation software has a simple mechanism to manage the virtual machines through cloud [12]. The cloud computing framework in this study not only manages the virtual machine but also provides related cloud services.

7. Conclusion and Future Studies

This study provides a general framework for media cloud service running on cross-platform with multiple clients especially in video streaming. Then, under this framework and using the algorithms applied in this thesis, port allocation handling is easily accomplished. In addition, there are six items in system analysis, namely CPU usage, read speed, write speed, memory usage, execution time of port allocation, and bandwidth. This study points out the video stream that is efficient and powerful for many clients with many video files. All OS platforms can be executed in this framework.

In future studies, the following two points are mentioned. First, social video sharing with friends through online cloud is an extension of this paper. This paper talks about asynchronous video streaming for private viewing. Further study can cover a social network that is a sharing location for friends who are online. Friends can share their favorite video files with each other. Also, they can watch the same video synchronously. Second, the resolution analysis is important to TV program. So far, no research or mathematical model can precisely indicate the effect of different resolutions on different devices that run TV programs under HTML5 streaming and the number of devices that can efficiently play video at the same time. This is what this study plans to do subsequently.

Acknowledgements

This study was supported by NSC under the Grant NSC 100-2221-E-006-250-MY3.

References

[1] Subashini, S. and Kavitha, V. (2011) A Survey on Security Issues in Service Delivery Models of Cloud Computing. *Journal of Net-Work and Computer Applications*, **34**, 1-11. http://dx.doi.org/10.1016/j.jnca.2010.07.006

[2] Banerjee, K.S. (2002) TCP Servers: A TCP/IP Offloading Architecture for Internet Servers Using Memory-Mapped Communication. Master Thesis, Department of Computer Science, Rutgers University, New Brunswick.

[3] Maly, R.J., Zurich, E.T.H., Mischke, J., Kurtansky, P. and Stiller, B. (2003) Comparison of Centralized (Client-Server) and Decentralized (Peer-to-Peer) Networking. Semester Thesis, Department of Informatics, Zurich University, Zurich.

[4] Hu, N.Z., Lee, C.Y., Hou, M.C. and Chen, Y.L. (2013) A Cloud System for Mobile Medical Services of Traditional Chinese Medicine. *Journal of Medical Systems*, **37**, 9978-9991. http://dx.doi.org/10.1007/s10916-013-9978-8

[5] Espada, J.P., Crespo, R.G., Martinez, O.S., G-Bustelo, B.C.P. and Lovelle, J.M.C. (2012) Extensible Architecture for Context-Aware Mobile Web Applications. *Expert Systems with Applications*, **39**, 9686-9694. http://dx.doi.org/10.1016/j.eswa.2012.02.151

[6] Watanabe, T., Cheng, Z., Kansen, M. and Hisada, M. (2010) A New Security Testing Method for Detecting Flash Vulnerabilities by Generating Test Patterns. *The* 13*th International Conference on Network-Based Information Systems*,

Takayama, 14-16 September 2010, 469-474. http://dx.doi.org/10.1109/nbis.2010.28

[7] Fukai, Y., Hirano, Y., Itoh, Y., Saga, S. and Kubo, R. (2012) Web Browser Based GUI for TV. *The* 1*st IEEE Global Conference on Consumer Electronics*, Tokyo, 2-5 October 2012, 579-580. http://dx.doi.org/10.1109/gcce.2012.6379919

[8] Ota, K., Kubota, H. and Gotoh, T. (2011) Media Cloud Service with Optimized Video Processing and Platform. *Fujitsu Scientific and Technical Journal*, **47**, 395-400.

[9] Callegati, F., Cerroni, W. and Campi, A. (2012) Application Scenarios for Cognitive Transport Service in Next-Generation Net-Works. *IEEE Communications Magazine*, **50**, 62-69. http://dx.doi.org/10.1109/MCOM.2012.6163583

[10] Goncalves, P., Roy, S., Begin, T. and Loiseau, P. (2012) Dynamic Resource Management in Clouds: A Probabilistic Approach. *IEICE Transactions on Communications*, **E95-B**, 2522-2529.

[11] Wang, X., Chen, M., Kwon, T.T., Yang, L.T. and Leung, V.C.M. (2013) AMES-Cloud: A Framework of Adaptive Mobile Video Streaming and Efficient Social Video Sharing in the Clouds. *IEEE Transactions on Multimedia*, **15**, 811-820. http://dx.doi.org/10.1109/TMM.2013.2239630

[12] Angeli, D. and Masala, E. (2012) A Cost-Effective Cloud Computing Framework for Accelerating Multimedia Communication Simulations. *Journal of Parallel and Distributed Computing*, **72**, 1373-1385. http://dx.doi.org/10.1016/j.jpdc.2012.06.005