An Iaas Based Interactive Video Platform for E-Learning and Remote Services

Hsiao-Tien Pao[1] Hsin-Chia Fu[2] Yi-Jui Lee[3]

[1] Department of Management Science, National Chiao Tung University
Hsinchu 300, Taiwan, ROC
htpao123@gmail.com

[2] College of Engineering, Huaqiao University
QuanZhou 362021, Fujian, China
hcfu99@gmail.com

[3] Department of Computer and Engineering Science, National Chiao Tung University
Hsinchu 300, Taiwan, ROC
yjlee@gmail.com

Received 17 May 2013; Revised 21 July 2013; Accepted 25 July 2013

Abstract. Interactive Video Platform intends to help people communicate with each other through Internet multimedia. Over the platform, users around the world may be linked together to interactively help each other solving their problems. In general, when a research based web prototype was opened to general public, it may often be paralyzed by too many users logged in. Thus, this paper proposes a generic implementation method: first converting the existing web hardware architecture to an IaaS cloud, then porting the application software to the IaaS cloud. Cloud computing architecture has better scalability, higher availability and lower cost-effectiveness. Therefore, when a cloud computing based interactive video platform has been completed, the system usually can provide flexible and reliable services to general public. In addition, since the proposed IaaS cloud approach is not limited or associated to any particular application or architecture, thus the proposed implementation method can be widely applied to convert most of research prototypes into real world application systems.

Keywords: Cloud computing, Infrastructure as a service, IaaS, Web site architecture

1 Introduction

Recently, multimedia and Internet technologies have remarkable progress, such as excellent picture quality and faster transmission speed as well as wider bandwidth. Thus, face to face meeting, classroom teaching, and onsite services can all be replaced by real-time video interaction through Internet, without participants getting together at one place. Conforming to this trend, we proposed to integrate video streaming and Internet multimedia communication technologies, to develop a system called "Interactive Video Platform for E-learning and Remote services [1],"so that users at the platform can proceed various interactive activities. By sharing video streaming, images, and text documents on focused objects or areas, participants at distance away may all get-together at a platform with face to face like interaction to help each other.

When an interactive video platform (IVP) enters public practical stage, it can immediately shorten the distance between people at remote area and people in metropolitan area, so that the life, education and business level of remote areas can be effectively improved. Like many Internet and/or WWW research, when a research prototype was opened to general public, it is often difficult to afford the heavy load brought by massive users. This paper proposes a methodology to lift the load capacity of a web platform, by finding a universal common solution model, so that most of web platform will be able to provide the majority of Internet user ready-to-use, high-performance services.

According to the NIST Definition of Cloud computing [2], cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Thus, Infrastructure as a Service (IaaS) provides the consumer to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

This research has two objectives: (1) using existing computer hardware to construct an IaaS cloud [3, 4], then converting and/or porting the core software of the IVP prototype [1] to the IaaS cloud, such that the IaaS based IVP (IIVP) can provide 24 hours a day and 7 days a week service, also can effectively sustain the heavy workloads brought by the massive general public users; (2) propose a universal generic method and procedures for research people to convert their prototype into a real world applications. The proposed methods, procedures and implementation will focus on the following three themes:

1. **Scalability**: When system load rises, the maximum service capacity of the platform can be improved by means of horizontal scaling.
2. **Availability**: When high-availability design applies, reliability of each component, single point failure, and system maintenance time can be greatly reduced.
3. **Cost-effectiveness**: When reducing construction and maintenance manpower expenditure, the service performance of a system with limited resources can be greatly enhanced.

In the followings, Section 2 briefly introduces the basic concepts of the prototype *Interactive Video Platform (IVP) for E-learning and Remote services*, and some existing IaaS cloud [3, 4] based WWW systems. Section 3 describes how to convert the IVP prototype hardware into an IaaS cloud, and how to port IVP software to the created virtual machines in an IaaS. Section 4 presents the experiments and the performance evaluation of the IIVP prototype. From the experimental results, improvements and corrections on the proposed IIVP system can be verified and further improvements can be conducted from that on. Finally, Section 5 presents summary and a few words of future works.

2 Research Review of The Related Research

This section first presents the basic concepts of some popular WWW systems, and a brief description of advantages and disadvantages of the existing Interactive Video Platform. Then, concepts of how to use IaaS architecture to build a better Interactive video platform is introduced.

2.1 Large WWW Architecture Evolution Instances

The architecture evolution of LiveJournal

LiveJournal [5] was founded in 1999 by Brad Fitzpatrick. Earliest version of LiveJournal was built on one server, and soon after the number of users exceeds the upper limits of a single server, such as the CPU utilization overload becomes a bottleneck of the Web server. Then, two more servers were added to the system, thus a total of two Web servers and one database server were installed at the LiveJournal Web platform. As the user continues to increase, the problem of insufficient single database I/O capability began to appear. Then, the single database server structure was modified into multiple database server architecture. By using master-slave configuration, a temporary and quick solution was achieved to conquer the I/O capacity problem often encountered at the single database server environment.

In short, based on the scalable design, additional computing load for the web and database servers due to increasingly new users, can be handled by installing more powerful servers.

The architecture design of Flickr

Flickr [6] is a web platform for photo storage, and online community social sharing. Fig. 2.1 depicts the architecture diagram of the Flickr site [7].

As shown in Fig. 2.1, a pair of load balance severs for the purpose of high-availability design is located at the forefront of the Flickr, and subsequently followed by Squid catches, as reverse proxy servers, and PHP App servers for the storage and execution of Flickr WWW codes. The Squid catches are responsible for static pages and image cache in order to reduce the burden of the actual pictures storage at the NAS devices and file systems. PHP App Servers are responsible for managing dynamically generated Web pages.

2.2 Interactive Video Platform

As shown in Fig. 2.2, the interactive video platform (IVP) uses the Client-Server architecture to achieve its structure functionality needs. A user at client-side may set up available video devices first, and then uses a browser to connect the IVP for audio and video communicate with other users at the platform. The server clusters at the platform process the video contents from sending users and dispatch the video contents to designated users. The platform contains three types of different function server clusters: Web Server (WBS), Media Server (MDS), and Message Server (MGS).

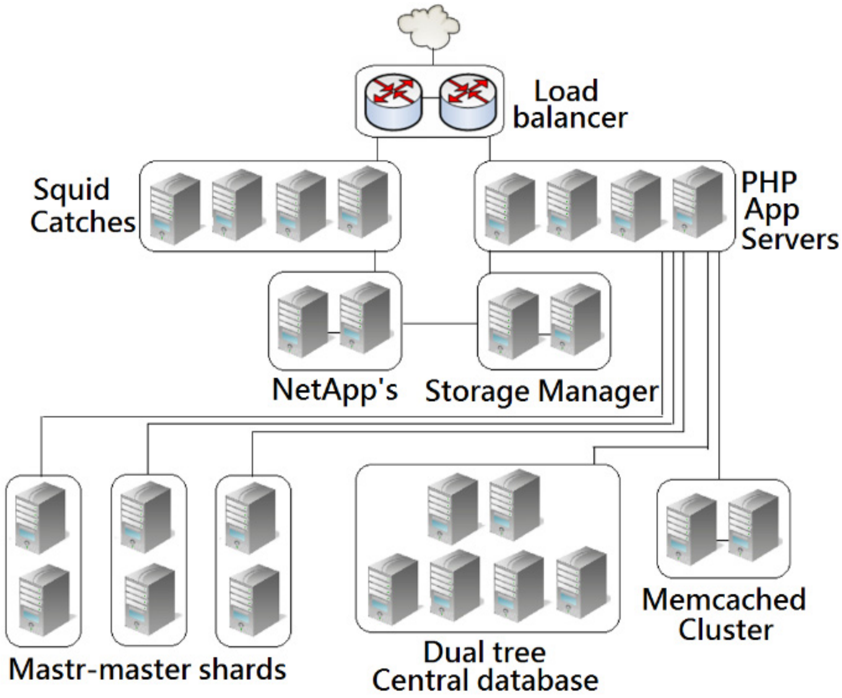

Fig. 2.1. Architecture of Flickr.

When there are more and more users at an IVP, its workload will eventually transcend the service capability of a single server. Thus, the server will not be able to deal with the instant needs of users, and cause the users to wait for a long time or unable to get connected to the platform. Eventually, the IVP's service quality is greatly reduced. Therefore, how to establish a platform for scalability to meet the ever-increasing load, becomes a serious issue for the high performance architecture design of a platform.

The major components of IVP are summarized as follows.

Web Server (WBS) Design: Web server provides client side users a browser to connect other users for various types of server functions available at an IVP. A user, whether or not to be an expert, must be a registered user to use a browser to connect the IVP. In addition to general personal information, e.g., user name, email address, nickname, and phone numbers, registration at IVP also includes personal interests, skills and expertise level. According to the skills and the expertise level of a login user, WBS connects associated experts for the online users to conduct various interactive activities, such as e-learning, remote services, or social discussion.

Media Server (MDS) Design: Media server provides users of various video associated functions available at IVP, such as publishing, co-editing, storing, and reusing of video clips. Adopting Client-Server architecture for the design of IVP is mainly for the convenience of server-side, to centralize editing and managing of audio and video contents and database, and to reduce workload and hardware requirements at the client-side.

Message server (MGS) Design: Among various interactive processes at the IVP, prompt and clear message communication between users is the major activities of a MGS. If there is only video sharing or interaction among participants, there may be semantic ambiguity or emphasis vague situation between users. Therefore, additional interactive text message communication is added and provided by MGS. The benefits of interactive text message communication are at least three fold: be able to achieve careful expressions of semantics, be able to repeatedly viewing and reading messages, and be able to provide multinational language translation. A MGS provides interactive communication service by linking associated users and experts in a ***discussion chamber***, and manipulation and management of each discussion chamber like mechanism. The aforementioned streaming video transfer is also controlled by the associated mechanism of a discussion chamber.

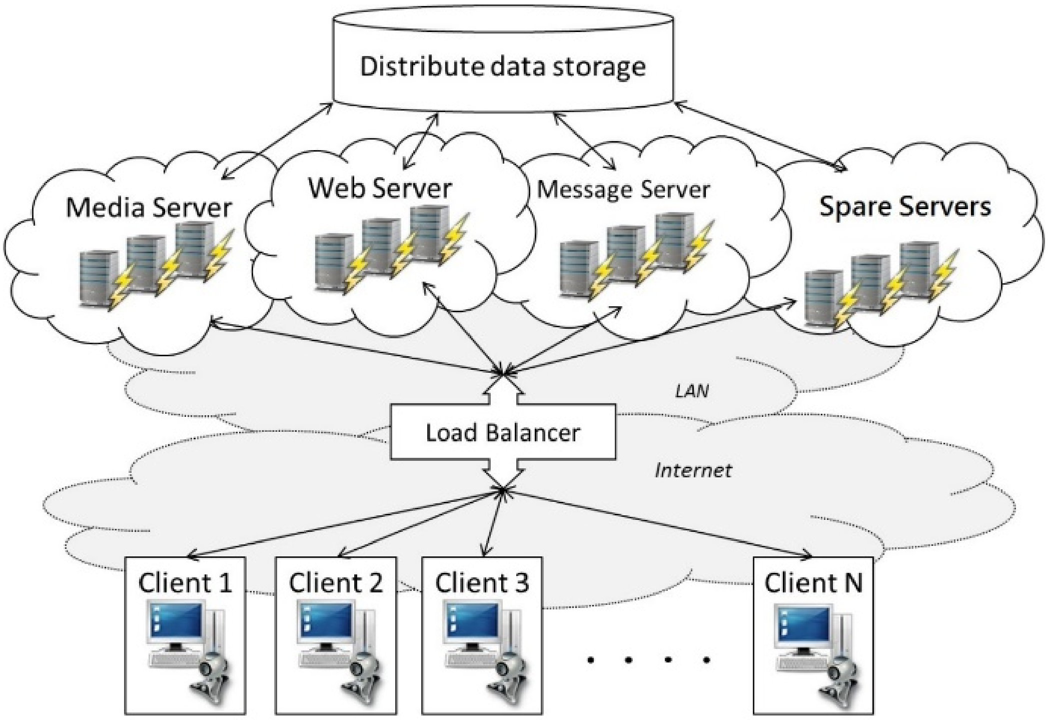

Fig. 2.2. The architecture diagram of the proposed interactive video platform (IVP).

Architecture Evolution Remarks

As the previous description on the structure evolution of LiveJournal and the web architecture design concepts of Flickr, it is clear to see that in order to achieve the expected design goal of IIVP, designers need to address the problems at the level of web architecture, and also to consider issues at the scope of infrastructure:

1. **Insufficient of available hardware machines:** In order to ensure an IVP running at normal to hardship situation, the new architecture needs to be operated and tested with extendable scalability and high availability. In this situation, more than enough hardware computing machines are needed to maintain a flexible running capability.
2. **Difficult to predict future demand**: From the LiveJournal development and evolution history, the expansion of a web architecture is an ongoing process, since the actual load or bottleneck of a web site may be varied at different time, and/or different user conditions. Cal Henderson, Flickr chief architect, wrote in his book [8]: "*The bottleneck is difficult to be found by solely using the benchmark or testing programs, and in fact it is usually found in actual operational data collection and is detected by through and deep analysis*."
3. **Increase of the maintenance costs**: Since the high availability design for various applications, and the scalable design for concurrent multiple servers, will drastically increase the required maintenance manpower and expenses. Or, the system may not be in normal working conditions or performing at the original design capacity, due to lack of adequate maintenance support.

Thus, an effective improvement method for a web site needs to consider both the architecture of the site itself and also its related infrastructure.

2.3 Infrastructure as a Service

According to the conclusions of the previous section, it is clear to see that in order to improve the performance of IVP, there are three main issues to face during the architecture design of the infrastructure construction. They are: (1) Insufficient of available hardware machines, (2) Difficult to predict future demand, (3) Increase of the maintenance costs. The design goal of a cloud computing infrastructure as a service (IaaS), is for on demand services by unlimited virtual machines, storage space, and efficient monitoring and management systems.

Based on the on-demand service, developers may always acquire just enough computing equipment, and then through just in-time self-management interface for more or less computing resources when needed.

IaaS for WWW applications

Porting an existing WWW application to an IaaS cloud has the following benefits: (1) Problems of lack enough physical machines can be solved by renting more virtual machines. (2) The difficult of predicting future demand can be alleviated by dynamically adjusting the number or the type of virtual machines. (3) Since an IaaS usually provides some management and maintenance services to low level physical machines, web designers may focus on scalable new development work. In other words, an IaaS based system designer only needs to focus on the improvement of scalability, availability, and cost-effectiveness of the web site itself.

However, most of commercial available IaaS are not suitable to host an application driven web site. For example, Amazon Web Services (AWS) is an IaaS provided by Amazon, which includes Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3), Amazon Cloud Watch, and Amazon Simple DB, etc. Through WWW interface, a user can lease demand services, such as EC2 for flexible virtual machine lease, and S3 for user file online storage services. According to news report, S3 has been continuously crashed for 8 hours in 2008, and AWS was crashed three days in 2011. Thus, all the migrated applications on the S3 and/or AWS were all failed and may be completely destroyed. In fact, S3 and AWS experts usually do not have enough understanding on the installed application software, and vice versa the application software developers. Thus, when the IaaS system were recovered from crash, it is unknown and uncontrollable when an IaaS based application system can be completely restored back to work.

In view of this, we suggest that any web application developer needs to have better understanding of cloud computing architecture and infrastructure software, so as to build a cloud computing IaaS from most available and suitable computer hardware, e.g., desktop PCs and open source cloud computing software. In other words, based on the IaaS model, one can first convert available computing hardware into a small private IaaS cloud, and then design and implement a new IVP architecture, based on virtual machines available from the completed IaaS Cloud. At this point, the number of servers needed by the IVP is not limited to the available physical machines. Thus, the scalability problem mentioned earlier can be alleviated. Also, as long as the needed virtual machines can be migrated and created among various physical machines, then there is no need to estimate the accurate number of servers and to massively change the entire platform architecture, when additional physical severs are installed for the implementation of various new applications. By using the information collected from a cloud monitoring system and virtual machines, the aforementioned difficult to predict future infrastructure demand becomes less critical issues. On the third problem, usually the damaged physical machine needs to be replaced or maintained by users. However, since the servers are running on virtual machines, damaged servers may be recovered or regenerated by installing the corresponding software image files. Also, the clouds built-in monitoring system can effectively prompt the virtual machine running status, thus manpower cost for server maintenance can be greatly saved.

Therefore, by going through a complete integration process, i.e., from basic components to a full functional IVP, student designers in this program will not only learn general cloud computing parallel software coding, but also achieve much better concept understanding and hands-on experiences on building a cloud computing infrastructure.

As the previous analysis, we will conduct our approach in two aspects: cloud computing infrastructure and interactive video platform architecture. On the cloud infrastructure side, the existing or available hardware will be converted to an IaaS cloud, so that any current or future site architecture improvement or change will not be limited by the total number of available physical machines, at the same time the web management and maintenance burden can be greatly reduced. On the web site structure improvement aspect, we will reference from the experience of today's large-scale web sites in this area to achieve a scalability, availability, and cost-effectiveness interactive video service platform.

3 IaaS Based IVP Design

This section presents a small sized IaaS architecture, and the design issues of an IaaS based interactive video platform (IIVP). The small sized IaaS is built based on basic cloud computing hardware and software components, then the architecture of data storage and communication networks for IaaS is presented. The presented design issues of IIVP mainly include how to use an IaaS to simplify the hardware architecture and to improve the performance of the original IVP. More detail design descriptions of IaaS and IIVP architectures are referred to [9].

3.1 IaaS Architecture

This section introduces the IaaS architecture which includes the design of IaaS storage (cf. Section 3.1.1) and IaaS network (cf. Section 3.1.2). Basically, an IaaS cloud [3, 4] needs physical hardware like PCs or servers, and operational software like the hypervisor to operate virtual machines. However, setting network structure and event management among virtual machines all needs human labor work. Thus, it usually requires management software, e.g., Virtual Infrastructure Manager (VIM), to deal with the generation and maintenance of virtual machines, the virtual URL setting, the installation and the management of the image files for virtual machines. Therefore, the user's working level are moved from physical nodes up to logical units, the cloud.

Fig. 3.1 depicts an IaaS cloud architecture, which contains 5 physical nodes, e.g., nebula-master, and nebula-slave 0~3.

Table 3.1 shows the hardware configuration of the five nodes in the IaaS cloud. The storage device in each node contains two 1TB hard disk with *RAID 1* setup to allow the node with better fault tolerant capability.

On the software configuration, Red Hat CentOS 6.2 x86_64 [10], Kernel-based Virtual Machine (KVM) [11], and OpenNebula [12] are selected for the IIVP. Red Hat selects Kernel-based Virtual Machine (KVM) as their main virtualization solution, therefore KVM is selected as the hypervisor for the proposed IIVP. OpenNebula provides very large autonomy to designers, such that an IaaS designer has great flexibility to plan and to design their own cloud to cooperate with a VIM.

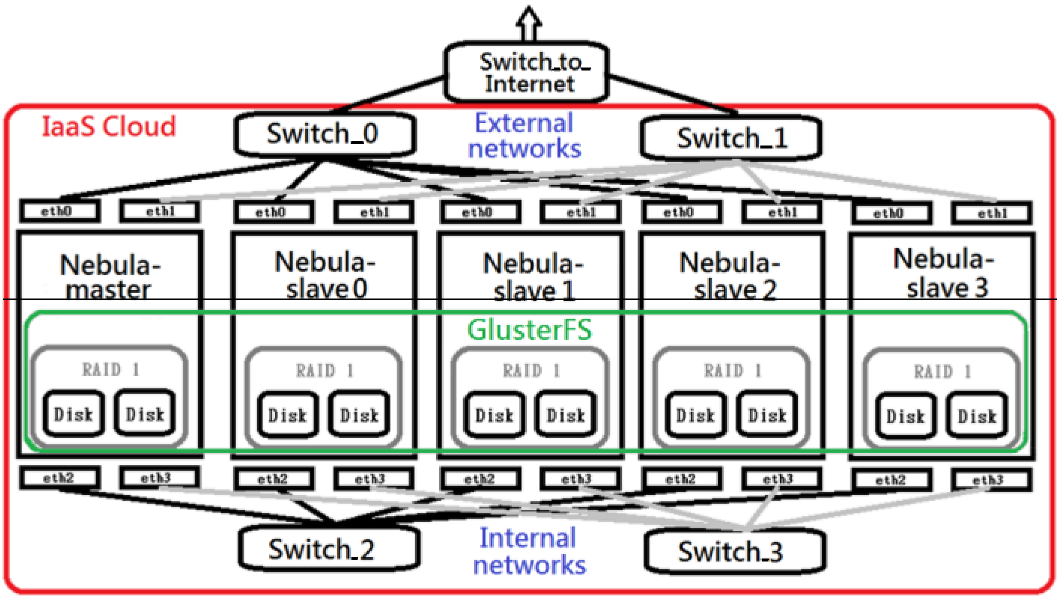

Fig. 3.1. IaaS cloud architecture.

Table 3.1. The hardware configuration of the IIVP.

Node name	Nebula-master	Nebula-slave0, Nebula-slave2	Nebula-slave1, Nebula-slave3
Processor	Intel I5-750	AMD FX-8120	AMD P II-945
Mother board	Asus P7P55D-E	Asus M5A97	Asus M4A88T-M
Memory board	DDR3-1333 16G	DDR3-1600 16G	DDR3-1600 16G
Hard disk device	WD1002FAEX * 2	WD1002FAEX * 2	WD1002FAEX * 2

As the responsibilities of each node, Nebula-master is the host of the entire cloud. Nebula-master receives a command issued by a user, then passes the command through *ssh* to a targeted Nebula-slave. The hypervisor in the targeted Nebula-slave performs virtual machine management operations according to the received command.

The following two subsections focus on how to design and to build storage and network systems in an IaaS based cloud.

3.1.1 IaaS Storage Design

The storage system is responsible for the data storage of virtual machine files and software disk images. On the types of setting of a storage system, it can be roughly divided into: non-shared storage as well as shared storage. Taking into account the advantages of using live migration that brings the platform more service capacity and maintenance convenience, shared storage mode and GlusterFS [13] file system were selected to implement the storage system. On the implementation, four nodes, i.e., Nebula-slave0 ~ Nebula-slave3, contribute part of their own disk space to form a shared storage area. All files saved in the shared storage area are first duplicated by GlusterFS and then are hashed into two nodes, to avoid the data accessing problems due to single node failure.

3.1.2 IaaS Network Design

According to various operational functions of the IaaS, the interconnection network may demand different types of planning and designing. Therefore, the design of network system for IIVP will be divided into *internal* and *external networks* as shown in Fig. 3.1. Connecting by *Switch_0*, *Switch_1* and *Switch_to_Internet*, external network provides communication channels among virtual machines, physical nodes, and Internet. By using *Switch_2* and *Switch_3*, internal network provides communication among virtual machines and IaaS cloud inner management, and data transfer among shared storage.

Since IVP and some similar research projects often have infrastructure problems to implement real world applications, this section reports how to design and implement an OpenNebula based IaaS cloud to tackle inadequate infrastructure problems. Having finished IaaS cloud based infrastructure design and implementation, the follow-up web architecture development, deployment and operation may use virtual machines to replace physical servers, to avoid the limitations of the available physical components, to achieve online migration functions through shared storage and high availability network design, to allow more resource allocation flexibility, and to enhance physical machines with better resistance to failure.

3.2 IIVP architecture

This section describes how to design the architecture of an IaaS based interactive video platform. First, a general description of the IIVP is presented, then followed by the description of major components in IIVP, including their principle functions and ways to improve their performance.

The proposed architecture of an IaaS based IVP (IIVP) is shown in Fig. 3.2. As shown in Fig. 3.2, there are total of six categories of server clusters in IIVP, which are *load balancer*, *shared storage cluster*, *web cluster*, *message cluster*, *media cluster*, and *database cluster*.

When a user logins at an IIVP, the load balancer will first assign the user to a Web server, then the Web server will perform the followings according to users' needs: (1) access data to and from shared storage, (2) connect to a media server for the exchange of audio and video with other on-line users, (3) transfer user hand-drawn sketch and/or text messages to and from all the associated users in a discussion chamber (cf. Section 2.2). The following subsections describe the organization and functions of each clusters.

3.2.1 Load balancer

Briefly speaking, when a web architecture went through scalability modification, one of visible difference is originally a single server's responsibilities will be diverted to multiple servers, i.e., a server cluster. How to properly and evenly distribute login users to web servers will be the major responsibility of a load balancer. A load balancer may be implemented by either hardware or software. However, considering easy to update and/or cost performance, it is generally prefer to software load balancer. The HAProxy and Linux Virtual Server (LVS) are two most popular software load balancer projects. In addition to *Virtual Server via NAT* mode as HAProxy, LVS also has two unique operation modes, the *Virtual Server via Direct Routing* and *Virtual Server via IP Tunneling*. Thus, IIVP selects LVS as the load balancing solution. To confrontation the impact of load balancer failures and to avoid single load balancer failure, usually two LVS servers are installed to provide mutual support service mode.

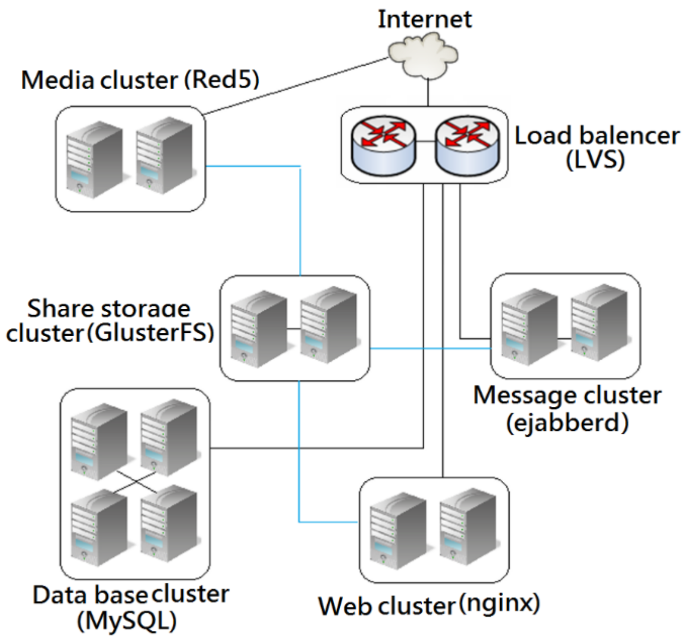

Fig. 3.2. The proposed architecture of IaaS based IVP (IIVP).

3.2.2 Shared storage cluster

Shared storage service allows text data, video clips and pictures can be stored and/or retrieved by any web server in an IIVP. Also, any media server in an IIVP may store or retrieve video clips in shared storage, instead of accessing video clips from the media server where the video clips were recorded. Shared storage uses GlusterFS to implement its distributed file system as the IaaS infrastructure, so that users may save troubles to learn a variety of similar software. As the matter of scalability, by increasing the number of servers in shared storage may improve their service capacity; and duplicate copies of all the stored data may allow the shared storage continues to run when partial service nodes failed.

3.2.3 Web cluster

The improvement of web cluster in cloud infrastructure is to change a single physical server to multiple virtual servers, so as to provide IIVP with more power to perform information management and user guidance. The improvement not only enhances the processing speed, also the reliability and availability of the web services. Automatic generation of virtual servers makes the design for extendable availability in response to physical server failure becomes possible. When a load balancer performs health check, failed virtual or physical servers will be detected and removed, then new virtual servers will be automatically generated. Users at the lapsed servers will be re-assigned to other web server and the session will be temporarily disabled, thus users will lose part of the status quo, but will not aware of any other abnormal status.

3.2.4 Message cluster

Since the software written for original message server did not provide all the functions required by IIVP, so we decided to rewrite the message server (MSG) codes for the IIVP. The functions performed by a MSG were quite similar to an Internet chat system. After further analyzing several similar chat systems, we found the Extensible Messaging and Presence Protocol (XMPP), which was approved by the Internet Engineering Task Force, contains a large number of open source codes for the interaction between server and client users, thus we decided to use these codes to implement the software for message cluster. Then, among several XMPP based servers, *ejabberd* [14] is a widely used XMPP cluster. And, when more services are needed, *ejabberd* may have more computing power by assigning extra virtual or physical server nodes to effectively solve the scalability problems.

3.2.5 Media cluster

The functions of a media server include: (1) On-line video stream transfer between users, and (2) off-line video playback. As video files are all stored in the shared storage cluster, so any video clip playback can be provided by a media server. However, on-line video stream transfer is not the case, video stream is first temporary stored in the media server, where the video stream was generated. Thus, if a user wants to receive a particular online streaming, then one needs to connect to a specific media server, which may led to the problems of uneven load distribution over the media servers in IIVP. Free media server software, e.g., Red5 Media Server typically does not support cluster features, thus media server uneven load issues need to be solved by IIVP designers.

Many cluster software packages commonly use ***simple hash*** to achieve load balancing among servers. Simple hash codes for media interactive activities in IIVP are temporary stored in a message server. According to the activity name and time code, each IIVP media interaction will perform SHA-1 hashing to generate a hash code. According to the characteristics of the hash code, users at client side can figure out which media server to connect, and to ensure that all users of the same activity are connected to the right media server, thus to reach a balance over the media servers at the macroscopic point of views.

3.2.6 Data base cluster

In general, a database is responsible for the storage and management of information in a computer system. How to strengthen data storage accessing speed, and how to fight against the single point (server) failure will be the focus point for IIVP database cluster designers.

General speaking, there are three approaches to increase database access capabilities, namely: (1) cache, (2) duplicate copy, and (3) segmentation. These three issues can provide the demand for the expansion of a database. In fact, the implementation of these approaches, all require a lot of manpower to perform substantial modification at the web interface codes. Therefore, IIVP adopts MySQL Cluster approaches - using the built-in NDB engine to perform data segmentation in its own cluster, so as to achieve a transparent solution for scalability issues in the database cluster. In terms of usability, based on mirror configuration in NDB, partial nodes failure do not affect access to the entire database. This can be easily achieved by correctly setting load balancer pointing to MySQL Cluster, without substantial modification of the code.

This section presents the architecture improvement of the proposed IIVP. Aiming to the improvement of availability and scalability for each system components in IVP, so that the IIVP has the ability to resist partial components failure, and the IIVP can be extended at any time according to actual demand for services.

4 System test and evaluation

This section presents experimental results and the performance evaluation of the IaaS based Interactive video platform (IIVP) prototype.

4.1 IaaS Infrastructure test on network systems

This section presents the ***iperf*** [15] test on the infrastructure of IaaS network subsystem, in order to verify whether or not the internal and/or external networks of IIVP do perform their data transfer functions as original planning. There are two test items: (1) test whether or not both networks are in normal operation mode, and (2) test the network switches in simulated failure operating modes, by shutting off one of the two network switches. The tested hardware and software specifications are shown in Tables 4.1 and 4.2.

As shown in Table 4.3, when half of the network switches fail, the network system in IaaS infrastructure can provides partial networking functions, and the combined network cards may output 1.983 times of single network card transfer bandwidth. The verification results show that the network switch design in IaaS infrastructure do play its own role.

4.2 IaaS infrastructure test on storage system

For the IaaS infrastructure test on storage system, the built-in Linux ***dd*** command will be used to test the write-in performance of the shared storage. The test command measures the average write-in rates while a total of 5G byte data are used. There are three different test conditions: (1) under normal circumstances, (2) shut off two slave nodes, resulting in the loss of a complete copy from the original two copy of data in the shared storage, and (3) shut off the internal network switch, such that network transfer speed is about half of the full transfer rates. According to the specified write-in rates, it is observed that (1) whether or not the survival of GlusterFS normally operates under the condition of single copy of data, and (2) whether or not the GlusterFS performs under half network bandwidth.

Table 4.1. Hardware configuration of the network systems in IaaS based infrastructure.

Server or client / Hardware spec	Server side	Client side
Node name	nebula-slave0	nebula-slave2
Server CPU	AMD FX-8120	AMD FX-8120
Server memory	Kingston DDR3 1600 16G	Kingston DDR3 1600 16G
Mother board	ASUS M5A97	ASUS M5A97
External network switches	Realtek 8111E D-link DGE-528T	Realtek 8111E D-link DGE-528T
Internal network switches	Intel EXPI9301CT PRO/1000CT	Intel EXPI9301CT PRO/1000CT

Table 4.2. Software configuration of the network systems in IaaS infrastructure.

Software catego-ries	Software name and version
Operating systems	CentOS 6.2 X86_64
Test software	*Iperf* 2.0.5 (iperf-2.0.5-3.el6.x86_64.rpm)

Table 4.3. Testing results from the network systems in IaaS based infrastructure.

Network categories / Connection status	External net-work	Internal net-work
Normal connection	944Mbps	1.87Gbps
One switch shut off	944Mbps	943Mbps

Table 4.4. Testing results from the shared storage system in IaaS based infrastructure.

Network condition / Test conditions	Normal internal net-work bandwidth	Half internal network bandwidth
All nodes normal	80.2Mbyte/sec	49.4Mbyte/sec
Loss of a copy	82.5Mbyte/sec	45.4Mbyte/sec

As shown in Fig. 4.4, all nodes, which are in normal condition or loss of a complete image copy, have same test results. In fact, loss of a complete single copy does not affect the work of the shared storage. Therefore, GlusterFS indeed, as claimed has the ability to continue services when partial nodes failed.

As shown in Tables 4.3 and 4.4, half of the internal network failure impacts equivalent to the loss of half bandwidth, i.e., the shared storage write-in rate decreases approximately 41.7%. On the third test condition, *shut off the internal network switch* is equivalent to half of the internal network failure, or loss of half bandwidth. The associated testing results are listed in the right column of Table 4.4. As can be seen from the network of IaaS infrastructure, the use link aggregation increase the network bandwidth and ensure major key influence to the performance of the shared storage.

4.3 Tests for scalability, availability, and cost-effectiveness.

As described in Section 1, the major themes of the proposed IIVP are the *scalability*, *availability*, and *cost effectiveness*. The following subsections 4.3.1, 4.3.2, and 4.3.3, are presented to depict the achievements of the proposed IIVP prototype on these three themes.

4.3.1 Tests for scalability

Scalability tests are performed to measure the advantages and disadvantages between IIVP and IVP, in terms of response time, video stream smoothness, HardDisk I/O rates, network loads, and user counts. As shown in Section 2.2, the IVP contains an Media server (MDS), a Web server (WSS), and an Message server (MSS), each of which are implemented by a PC based hardware and software [1]. According to the benchmark and performance evaluation of IVP server cluster [1, Section 5], the overall performance of IVP was limited by video stream through put, or the HardDisk I/O rate of the MDS. In other words, the IVP can provide 175 on-line users with 700kbps smooth video stream. The MDS server consumes about 130Mbps network capacity, and 10% of CPU computing loads.

The IIVP was first implemented by creating 3 VMs from the prototype of the IaaS. Each of VM has the same hardware and software features as the IVP servers, and then implemented with Media, Web, and Message application server software. Then, extra Media and Web servers were added by using more created VMs. According to Table 3.1, IIVP has ten 16G hard drives, which may provide a total of 1260 Mbps disk I/O rates, and has 4 AMD *FX-8120* and/or *AMD P II-945* grade CPUs. It is clear to see that, these 4 slave servers in IaaS should have much more computing power than needed computing capacity for the following various benchmark tests. Table 4.5 shows IIVP system testing performance and evaluation results, which includes video stream smoothness, response time, Disk I/O rates, network loads and total number of on-line users.

4.3.2 Tests for availability

The availability test intends to observe how a normal operating server responses to a suddenly shutdown (forced failure), and to check whether or not the responses in accordance with the expected performance. As shown in Table 4.6, the shutdown nodes for usability tests include: (1) load balancer, (2) web server, (3) message server, (4) media server, and (5) database server, etc.

According to the test results in Table 4.6, the failed components in IIVP response in consistent with prediction, and the improvement in terms of availability are more than expectation.

Table 4.5. System testing performance and evaluation results of the proposed IIVP prototype.

	#users	**#MDS**	**#WBS**	**#MSS**	**Response time**	**Video smooth**	**Disk I/O rates**	**Network loads**
Case 1	175	1	1	1	>250ms	OK	122.5Mbps	130Mbps
Case 2	1050	6	1	1	>2sec.	OK	735Mbps	780Mbps
Case 3	1050	6	2	1	<1sec.	OK	735Mbps	780Mbps
Case 4	1750	10	2	1	<1sec.	OK	1.22Gbps	1.3Gbps
Case 5	2625	15	2	1	>2sec.	OK	1.83Gbps	1.95Gbps
Case 6	2625	15	3	1	>1.5sec.	OK	1.83Gbps	1.95Gbps
Case 7	2625	15	5	1	<1sec.	OK	1.83Gbps	1.95Gbps

Table 4.6. Results of IIVP tests for availability.

Category of failed nodes	Category of predicted error	Test Results
Load balancer	No failure should be shown	OK
Web server	When switch pages prompted re-login, then everything backs to normal	OK
Message server	Prompt disconnection, then return to normal after re-login	OK
Media server	The video screen shows interruption, and there is possibility of replay failure	OK
Database server	Some probabilistic errors, then return to normal operation	OK

4.3.3 Remarks on cost effectiveness

As shown in Table 4.5, the overall IaaS system performance is still limited by the real hardware components. However, under the hardware performance limits, an IaaS can be flexibly adjusted by creating various number of different functional virtual machines (VMs). Therefore, the performance and efficiency of the application system can be greatly enhanced. As stated in Section 4.3.1, when there are more on-line users, more virtual machines and application servers can be created and implemented to host more application loads. Conversely, when less users are on-line, merging application servers and shut down unused VMs can save utility energy consumption. Furthermore, as described in Section 4.1 and Table 4.1, slave processors in IaaS can be flexibly added to implement more application servers for more computing power and storage capacity. In addition, as described in Section 3.2.2, shared storage cluster is implemented to duplicate copies of all the stored data to allow the shared storage, and/or backup servers continue to run when partial service nodes failed.

5 Conclusions and Future works

This paper presents IaaS cloud design concepts and virtual machine technologies, so that most of laboratory existing or available hardware can be converted into an IAAS private cloud [16, 17]. In an IaaS, more virtual machines can be created than the total number of physical machines, at any time the computing resources of virtual machines can be dynamically re-allocated, and the infrastructure administrative burden for various software and hardware can be effectively reduced.

In this study, an IVP prototype has been successfully put into practical stage, providing the general public an interactive service platform. It is especially convenient for people in remote areas to receive assistance from people at metropolitan areas, by narrowing the cyber distance among them, so as to effectively reduce culture, education, and daily life gaps between urban and rural areas.

The proposed methodology can be applied to all the similar research programs or projects. Learning from the experiences of this study, researcher may have significantly lesser difficulties and problems while transferring research results into practical applications. Thereby, research results are no longer subject to limited resources, to be ended at laboratory experiments and/or demonstration. In addition, the general public can all enjoy the benefit brought from these research results.

The current IIVP only provides services in the form of private cloud [16, 17]. Later, if the IaaS private cloud could combine multiple public clouds by establishing efficient and effective management and migration methodologies, then the combined clouds could generate huge computing power to provide great service performance for people around the world.

References

[1] H.-C. Fu, Y.-Y. Xu, H.-T. Pao, J. Wang, "Interactive Video Platform for E-learning and Remote Services," *International Journal of Computer Science*, Vol. 10, No. 1, pp. 154-161, 2013.

[2] P. Mell, T. Grance, "The NIST Definition of Cloud Computing," http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[3] M. D. De Assunção, A. Di Costanzo, R. Buyya, "Evaluating the Cost-Benefit of Using Cloud Computing to Extend the Capacity of Clusters," in *Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing*, pp. 141-150, 2009.

[4] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, D. Epema, "A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing," In *Proceedings of Cloud Computing 2009*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 34, pp. 115-131, 2010.

[5] LiveJournal, http://www.livejournal.com/, 2012.

[6] Flickr, http://www.flickr.com/, 2012.

[7] C. Henderson, "Scalable Web Architectures: Common Patterns and Approaches," Web 2.0 Expo, 15th–18th April 2007.

[8] C. Henderson, "Building Scalable Web Sites: Building, Scaling, and Optimizing the Next Generation of Web Applications," O'Reilly, U.S.A., 2006.

[9] Y.-J. Lee, *Architecture Design And Implementation of a Video Based Remote Service Platform in Cloud Environment*, Master thesis, Department of Computer and Engineering Science, National Chiao-Tung University, July 2012.

[10] CentOS: The Community ENTerprise Operating System, http://isoredirect.centos.org/centos/6/isos/ x86_64/.

[11] Kernel-based Virtual Machine, http://www.linux-kvm.org/page/Main_Page.

[12] About the OpenNebula.org Project, http://open-nebula.org/about:about

[13] Getting Started and QuickStart Guides, http://glu-ster.org/community/documentation//index.php/Main_Page.

[14] Ejabberd, http://wiki.xmpp.org/web/Ejabberd

[15] iperf, http://en.wikipedia.org/wiki/Iperf

[16] K. Salah, M. Al-Saba, M. Akhdhor, O. Shaaban, M.I. Buhari, "Performance Evaluation of Popular Cloud Iaas Providers," in *Proceedings of International Conference for Internet Technology and Secured Transactions (ICITST), 2011,* pp. 345 - 349, 2011.

[17] S. Bhardwaj, L. Jain, S. Jain, "*Cloud Computing: A Study of Infrastructure as a Service (IAAS)*," *International Journal of Engineering and Information Technology*, Vol. 2, No. 1, pp. 60-63, 2010.