

Videoconference Capacity Leasing on Hybrid Clouds

Javier Cerviño, Fernando Escribano, Pedro Rodríguez, Irena Trajkovska, Joaquín Salvachúa
Dpto. Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, 28040, Madrid
{jcervino, fec, prodriguez, irenatr, jsalvachua}@dit.upm.es

Abstract—This paper proposes a new methodology focused on implementing cost effective architectures on Cloud Computing systems. With this methodology the paper presents some disadvantages of systems that are based on single Cloud architectures and gives some advices for taking into account in the development of hybrid systems. The work also includes a validation of these ideas implemented in a complete videoconference service developed with our research group. This service allows a great number of users per conference, multiple simultaneous conferences, different client software (requiring transcoding of audio and video flows) and provides a service like automatic recording. Furthermore it offers different kinds of connectivity including SIP clients and a client based on Web 2.0. The ideas proposed in this article are intended to be a useful resource for any researcher or developer who wants to implement cost effective systems on several Clouds.

Index Terms—videoconference; cloud computing; hybrid clouds.

I. INTRODUCTION

Today, hybrid clouds are too often seen as a tool to reuse previous systems in addition to the ones provided by a public cloud to give the final user access to services through both public and private networks of an enterprise as seen in [1]. There are some works on this kind of systems such as OpenNebula [2], Zimory¹ or VMWare vCloud². However, as we know, there is a wide variety of public cloud providers each with its own characteristics, strengths and weaknesses and that should be taken into account when designing and deploying applications as well as the currently provisioned resources in an existent private cloud. According to this idea, in this paper we give another reason to deploy services in an hybrid cloud: to enhance the usage of resources efficiently. We think that there are many systems that can benefit from deploying on hybrid clouds instead of single provider clouds. We provide guidelines for developers so they can design their applications and services accordingly.

We want to validate this concept by presenting a system that follows this principle. This work is part of the Global Project³ and it is based on a service which offers to the users the possibility of scheduling videoconference sessions through a web portal (named GlobalPlaza⁴). The users could attend to each of these sessions through the web and even take part

of the meetings that would be created with this application. With this tool several users should be able to join to a single videoconference session and control it. This control is based on the interaction modes that are setup at different moments (presentations, conferences, questions, etc.). The system presented takes into account the different resources needed in order to establish a videoconference and reacts accordingly. In this project a new service that offers videoconference which can be accessed through different technologies has been created. These technologies can be summarized as SIP [3], web browser and Isabel application access, the latest being an application developed in our research group.

A decent number of applications and services that offer to their users the possibility of establishing videoconference sessions of multiple users through a Web portal are available today. Some of these systems allow the recording of their sessions. We can find examples of these portals such as WebEx⁵, FlashMeeting⁶ or Marte [4]. On the other hand, during this period some Internet portals have arisen to allow the users to upload their own recorded videos of lessons, conferences, etc. In these other portals the videos can also be accompanied with slides or presentations that have been used in the session. Examples of these services are Youtube⁷, RTVDoc⁸ or iTunesU⁹.

The great majority of these systems use Cloud Computing infrastructures in order to offer their services, either in the recording or in the storage of the generated videos. However, none of them offer a wide solution for creating, storing and distributing videos using Cloud technologies. At least at the moment of writing this paper we did not find an example of this kind of system.

We propose this work as a starting point of Cloud videoconference systems which are aware of the use of the available resources and the costs that are generated by them. Section III introduces the main motivation of this research, explaining how hybrid architectures can work better in some scenarios. Section IV gives us an example of the concepts presented in the previous section, this example is an implementation of a videoconference system that we have developed in our research group. Section V explains the validation of the system

¹<http://www.zimory.com/>

²<http://www.vmware.com/products/vcloud/>

³<http://www.global-project.eu/>

⁴<http://globalplaza.org>

⁵<http://www.webex.com>

⁶<http://flashmeeting.open.ac.uk/home.html>

⁷<http://www.youtube.com>

⁸<http://www.ucm.es/info/tvdoc/>

⁹<http://www.apple.com/education/itunes-u/>

in terms of cost and resource usage. Finally, some conclusions are presented in section VI.

II. RELATED WORK

We have designed, developed and made some tests of a new architecture for a session based videoconference system. This system presents a web application in which users can schedule, delete and modify videoconference sessions in which they are going to participate with many other users. It is intended to be used in many different scenarios such as classrooms, congress, tutorials, meetings, etc.

Furthermore, the system is focused on the optimal usage of the available resources. To do so we studied the existent hybrid infrastructures that were used for any other purposes rather than videoconference. And we also based our work on similar videoconference systems that do not use any type of Cloud infrastructures.

In this section we are going to present the related work in which we have based on.

A. Hybrid infrastructures

There are many researchs done in this topics, and when we started our work we could check some Cloud technologies that allowed hybrid architectures. From our point of view the most important were OpenNebula, Eucalyptus.

OpenNebula is an IaaS (Infrastructure as a Service) application. The main objective of OpenNebula is to allow the creation of a private Cloud in a data center. But it also has an API which can directly manage virtual machines on Amazon EC2 and another standard API named OCCI, so it is very interesting for hybrid architectures that use both third party resources and its own ones. In our case the most important feature of OpenNebula is that it can schedule the creation/deletion of instances of virtual machines, but this feature does not work in the case of Amazon EC2 instances.

Eucalyptus is also an IaaS application, with features that are similar to the ones from OpenNebula. This service lacks of a scheduler that can start and stop instances of the virtual machines, so we think that is less interesting for our purposes.

There are other important systems that offer similar features to the user. There is a very interesting table comparing different tools in [5].

On one hand they both are very interesting for future purposes even though they achieve the objective of hybrid clouds. On the other hand they do not meet all our needs of scheduling, so we finally decided to create a service that worked with Amazon EC2 directly and also with our current virtual machine solution, that was VMWare.

Obviously there is previous work about trying to improve resource utilization and allocating resources in Clouds, such as [6]. But all proposals consider only one Cloud provider, because all most of them are based on hypervisor element.

Furthermore, there are arising some researchs like the one in [7] that proposes a measurement tool for comparing different Cloud providers in order to select the best-performing provider for a given application of a Cloud customer. This tool can

TABLE I
COMPARISON OF VIDEOCONFERENCING SYSTEMS

Feature	FlashMeet.	GoToMeet.	WebEX	AdobeCon.	Skype
Web	yes	yes	yes	yes	no
Scheduling	no	yes	yes	yes	no
Recording	yes	yes	yes	yes	no
Streaming	yes	no ^a	no ^a	yes	no

^aNot in the same product.

be used to perform a comprehensive measurement study over four major cloud providers, namely, Amazon AWS, Microsoft Azure, Google AppEngine, and Rackspace CloudServers. And it says that the results of such study is ephemeral due to the periodic changes that every Cloud provider introduces in their software and hardware, and because the user demands vary over time.

B. Videoconference systems

Nowadays there are some videoconference systems that allow users to schedule web videoconference sessions or to participate through their web browsers. We can see FlashMeeting, Adobe Connect, WebEx, GoToMeeting, Skype, Marte 3.0 [4] and so on. Table I shows the features that are presented on the each of these systems.

Examples like WebEx that are running on Cisco private Clouds are very similar to the Conference Manager. But here in this work we propose the use of Hybrid Clouds. In other words, we want to use different infrastructures from different Cloud providers in order to enhance the global performance of our videoconference application.

III. MOTIVATION

A videoconference system that allows a great number of users per conference, multiple simultaneous conferences, different client software (requiring transcoding of audio and video flows) and provides a service like automatic recording, like the one we are trying to build requires a lot of computing resources. Figure 1 shows the videoconference scenario we are trying to build. This scenario includes several videoconference clients. Some are connected through a MCU and others participate via Flash or SIP. In both cases transcoding of the data flows is necessary. The scenario also includes a RTMP server for the flash clients and a SIP server for the SIP clients. In order to allow cost effective scaling of our videoconference system, the use of cloud computing resources appears as a natural approach, since they provide the illusion of infinite computing resources available on demand and the ability to pay for use of computing resources on a short-term basis as needed [8].

However the use of cloud computing resources from a single provider comes with several disadvantages as shown in [8], [9]. Critical problems that can benefit from an hybrid cloud architecture are listed below in no particular order.

- *Geographical location and legal issues.* It may be useful to start some services in a specific location for performance or legal reasons. The use of different providers

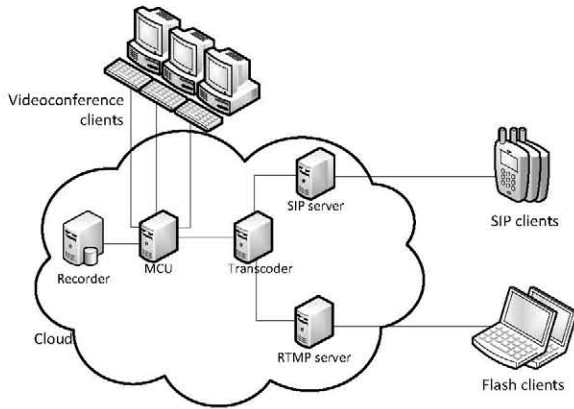


Fig. 1. Videoconference Scenario

will give us access to more locations or will allow us to start some services in our private cloud that may be more suitable for sensible data.

- *Cost and lock-in.* Different providers may offer different services at a different price. Furthermore this price may change over time. By using several providers we can use this to our advantage. In addition the use of a single provider may result in a lock-in problem.
- *Availability.* Cloud Computing service by a single company is a single point of failure. By using different providers we can achieve better availability.
- *Wasting of existing resources.* In some environments a lot of resources are already available to use. By moving all services to the cloud we are wasting these resources. The use of hybrid private/public clouds can avoid this problem.

In light of the problems listed above, the use of resources from different providers as well as private resources can help us to provide a service with better performance, lower cost and avoiding or mitigating most of the problems of cloud computing. This will be proved for our videoconference service in section V of this paper.

To be able to effectively make use of hybrid clouds we need two things. First we need to make use of a virtual infrastructure manager [2] to provide a uniform and homogeneous view of virtualized resources, regardless of the underlying virtualization platform. Second, we need to split our service in three parts:

- *CPU intensive modules.* Parts of the application that consume most of the CPU cycles needed to provide a service. In our case we have identified the transcoding and recording modules of our videoconference system as the CPU intensive modules.
- *Bandwidth intensive modules.* Modules that consume most of the bandwidth. In our videoconference system, the MCUs and RTMP servers are bandwidth intensive components.
- *Storage intensive modules.* Disk servers and databases fall into this category. In our case the recorded conferences

are stored in a NFS server.

This division gives us the opportunity of placing the modules that need a specific kind of resource where it better serves our needs and objectives. We have named this partition as *Cloud computing Resource Oriented Partition* or CROP. An example of this partition widely is widely used in video streaming systems, where we have the video encoder (CPU intensive) sending the encoded video to a streaming server (bandwidth intensive) and storing the video on a disk server (storage intensive). The videoconference system described in the next section fulfills these requirements and results in a scalable and cost-effective system.

IV. IMPLEMENTATION

The work we are going to explain throughout this section is based on a system that allows to schedule several videoconference sessions. These sessions are going to be created by the users and they are going to use the available resources in each moment. First there is an enumeration of the objectives of this work and then there is the description of the system.

A. Objectives

This implementation has two main objectives:

- To develop a videoconference system which can schedule, record and stream videoconference events. This system has an API that can be used by external applications in order to offer the services to their users. The access to the service could be done through different ways: SIP [10], Web browser (with Flash Player) and Isabel [11] and [12].
- A new resource reservation system which takes into account the different scenarios, technologies and resources that are going to be used. This is the objective that gives us the hybrid cloud perspective.

B. Conference Manager

Figure 2 shows a general architecture of the Conference Manager, which is the name of the videoconference system. Where only those components that are important to better understand the function of the hybrid cloud scheduler are present. This section describes all the components of the Conference Manager, that is divided into three parts:

1) *API:* This is the interface between the scheduler and third party services. This API is based on the REST methodology [13], so it represents conference events and sessions as resources that are accessible through HTTP requests. All these requests have to be authenticated via a mechanism that has already been implemented in [14].

There are many requests that are forwarded to the scheduler in order to create, modify or delete some videoconference events or sessions. Each event consists of one or more sessions that are going to be recorded or not depending on what the service is requesting. So the way that an event with several sessions has to be created is by firstly requesting the creation of such event and then the creation of all of its sessions, one by one.

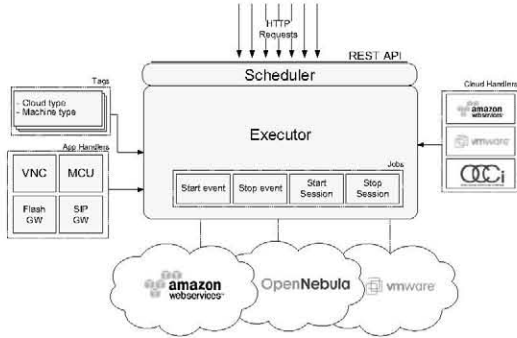


Fig. 2. Conference Manager Architecture

2) *Scheduler*: The next one is the responsible for scheduling all the videoconference events so they are executed by the next component. Each event consists of several jobs that have to be performed in a given order to correctly start or stop the event and its sessions.

The scheduler (which is based on the Quartz scheduling framework [15]) has to check the correctness of the system in the way that it can not use more than a maximum limit of resources, even if they are on the cloud. When there is a request for creating a new event or session it checks if there are enough available resources throughout its duration.

It is also responsible for deciding at which cloud provider the machines are going to be executed. This decision is based on three parameters that depict the starting of each of these resources: The current price of the resources that are required in each cloud provider in order to execute the machine, the geographical region where the session is going to take place and the type of machine that is going to be started by the executor. All the information about the cloud provider that is going to host the machine is stored at the machine tags.

There are many differences between start and stop jobs and also between event and session jobs. For each event there are two jobs associated with it, the first is the job that prepares all the necessary resources for such event. The latter does the opposite, it stops all the machines and releases all the resources that are used. It is the same with the session jobs, but the difference between events and sessions is the type of resources they are going to hold and release. In the context of this paper event jobs are more interesting since most of the resources they are going to manage are machines that can be hosted on different clouds.

3) *Executor*: This component executes the jobs scheduled by the previous component. This is done at the time scheduled with the dispatch of the required event and the information stored within it. For example, a new start event job could be scheduled in order to run the required resources for that given event. Once an event start is dispatched this component starts the machines and then the videoconference application with a specific configuration. This application is usually Isabel since this is the core videoconference service of this system.

As we have already mentioned above there are two types of event jobs, the start and the stop of the event. When the executor initializes a start event job it follows the next steps:

- 1) Retrieval of the event configuration from the database. It is useful in order to choose and get the machine tags which are going to be started.
- 2) Initialization of the machines on the given clouds. In order to do this the executor uses the corresponding cloud handlers, which are tools that can communicate with the API of its cloud provider. In this step all machines are going to be started and it does not end until all machines are running.
- 3) Execution of Isabel (or any other application such as VNC server). Once all the machines are running it starts Isabel on each machine with different parameters. Depending on this configuration Isabel can operate as Master, Flash gateway or SIP gateway depending on the functions that are required. The next section explains in detail each of these functions.

V. VALIDATION OF THE HYBRID CLOUD

In this section we are going to validate the system explained in the previous one in the terms introduced in section III. Subsection V-A gives details about Isabel resources and their costs while in subsection V-B we analyze the costs of these resources on different cloud typologies.

A. Isabel resource usage

1) *CPU cost*: The Isabel videoconference system uses a windowing system in order to represent accurately the process of the conference, showing all videos, applications and white-board on the screen. In a traditional Isabel conference there can be different clients connected through the network. This network can be organized as a tree or mesh overlay depending on the requirements of each scenario. Each Isabel node of this network can perform different functions: the main function would be the interactive mode in which users can take part of the session. Other function is the MCU that only forwards multimedia streams between Isabel terminals, SIP transcoder transforms the traffic between Isabel and SIP clients using the SIP server. Finally the Web Isabel transcoder does the same thing offering the users access from a web browser, in this particular case there is a RTMP server that forwards multimedia data using RTMP protocol to Adobe Flash clients.

Depending on the type of Isabel that is running on a machine, it could use more or less CPU. The use of CPU increases with the number of users for each type of Isabel node. The reason for this CPU usage on the SIP and web transcoders is that the main task of the application is to generate a single video showing the conference session, so it has to render all the videos and encode them with the corresponding codec.

2) *Bandwidth cost*: Something similar occurs with the bandwidth because depending on the way in which the users are connected to the conference, it would need more or less bandwidth. The bandwidth increases with the number of Web

users. It always reaches a limit in a way that we will explain later in this section. Again, in this case the MCU uses more bandwidth than in an Isabel session, most of all when its topology follows a star network. In these networks the MCU is going to forward all the traffic between nodes. This topology is not the most recommended but it is preferred to be used by the users because it is easier to be configured and it requires less machines. The SIP transcoder also consumes a lot of bandwidth as it is the entry gate to the videoconference for any SIP phone, so usually all SIP phones are connected to it directly. Finally, the RTMP server is the third of all by bandwidth consumption as a consequence of the number of users that are directly connected to it. It is also the responsible for sending the video to all web users.

For the calculation of the web bandwidth that a user consumes in a videoconference session we have taken into account that there is always a top limit in the number of video and audio streams. This is the maximum number of user videos that are shown in the screen at each moment. In our case, and based on a great number of recorded videoconference sessions in the past, we propose the equation 1 that references the real number of users who are sending video and audio streams at each moment. It can be summarized by saying that there is always a maximum number of videos that are shown in the screen and that the number of videos sent to the system is going to increase up to this maximum.

$$\alpha_{web} \cdot N_{web-users} = \begin{cases} N_{web-users} & \text{if } N_{web-users} < N_{max} \\ N_{max} & \text{if } N_{web-users} \geq N_{max} \end{cases} \quad (1)$$

Multiplying the number of videos that will be sent at a given time to the system by the bandwidth of each of these videos (it will be the same for all of them) we get the overall bandwidth of the system.

$$BW_{web-in} = \alpha_{web} \cdot N_{web-users} \cdot BW_{web-user} \quad (2)$$

$$BW_{web-out} = BW_{web-isabel} \cdot N_{web-users} \quad (3)$$

Both the CPU usage and the bandwidth are critical when we have to choose the right topology in order to make an optimal use of the resources which are available in the network and in the system. In subsection V-B we can see the definition of some of the variables that are part of the previous equations.

Figure 1 shows the architecture of a typical session in which users can be connected through Isabel as well as through the web transcoder. In order to simplify the equations SIP traffic will be left out of this scope. This architecture also has a system that records the video generated from the session.

3) *Costs of an Isabel session in the University:* The requirement for the above described resources as well as others that are not very critical, entails a usage in the University that is proportional to the number of machines used in each session. Most systems usually deal with this kind of problems and nowadays there are Cloud Computing based solutions that solve them.

These solutions are focused on the variation of the demand for some services (whether it is in large periods of time such as months or in short periods such as hours). The workload of these services and the resources needed usually varies along the time. The consequence is that during the workload peaks the system has less available resources than needed and that during periods of lower workload the number of available resources will increase.

In Spain all universities are directly connected with a high bandwidth network named RedIRIS, so in these scenarios the bandwidth consumption is not usually a problem. In the case of CPU usage, higher investments in machines are required depending on the number of the offered simultaneous sessions during the periods of low workload and during the peaks, so any investment saving on this resources will always be welcome. This is the reason why the case of CPU usage is a different problem and we have to use different approach to deal with it.

B. Calculation of the Cloud Computing architectures' cost

In the cost calculation we intended to abstract details of the IaaS (Infrastructure as a Service) provider. To do this, we have defined a set of constants that refer to the cost of each resource. Nowadays the best known example is Amazon EC2¹⁰ in which there are CPU and bandwidth usage costs. In the case of CPU the costs can vary according to the type of machine to be used and its amount of memory. In the case of bandwidth usage, it refers to the outgoing traffic as well as the incoming traffic to the Cloud Infrastructure. Specifically we have defined the next cost constants:

- C_{mcu}, C_{tr} : They represent the cost of a machine running an Isabel application that acts as MCU and transcoder. This constants are measured in \$/hour.
- $C_{T_{in}}, C_{T_{out}}, C_{T_{internal}}$: These are the costs of the incoming, outgoing and internal traffic data respectively, measured in \$/bit consumed per hour.
- $BW_{isabel-user}$: This is the bandwidth used by an Isabel user in each direction.
- $BW_{web-isabel}$: This is the bandwidth used between the RTMP transcoder and the RTMP server.
- $BW_{web-user}$: This is the bandwidth used by a RTMP client in each direction.
- $N_{isabel-users}$: This is the number of Isabel clients that are connected to the session.
- $N_{web-users}$: The number of RTMP clients (or web users) that are connected to the session.

Next we present two different architectures for the same system. In the first one, all the components that are part of the videoconference are executed in the Cloud Computing system where we are measuring the cost. In the latter we have divided the components separating those which make an intensive usage of CPU and those which are associated with the bandwidth consumption.

¹⁰<http://aws.amazon.com/ec2/>

1) *Costs of a single Cloud architecture:* The first architecture contains all of the components in an external Cloud provider, such as Amazon EC2. All the costs can be calculated knowing the provider rates, but we will first show the formulas that relate the resource usage (CPU and bandwidth) with the cost that is associated with each of them. Having the formulas facilitate the reader to calculate the cost with different Cloud vendors. The CPU cost is the result of summing the machines that are needed for an Isabel session. Although according to the figure 1 a session could be composed of more machines than those included in the formula, some of them are not used only by a single session. They are shared among different sessions, so the value for any scenario in which there are several simultaneous sessions is negligible. In that way we have three machines which have different requirements with respect to CPU and memory.

$$C_{cpu} = (C_{mcu} + C_{tr}) \cdot t$$

The result represents a single second duration cost of the videoconference.

The second formula is the one which works out the cost of the traffic generated by the internal, outgoing and incoming Isabel transmissions that are sent through the Cloud infrastructure. In the case of Isabel we could assume that the consumed bandwidth will always be the same for each connection between Isabel terminals, so the formula will be the following:

$$C_{BW_{isabel}} = (N_{isabel-users} \cdot (C_{T_{in}} + C_{T_{out}}) + 2 \cdot C_{T_{internal}}) \cdot BW_{isabel-user} \cdot t \cdot 3600$$

The case of the bandwidth consumed by the web users depends on the number of users who are connected to the session. Here we can see the variables defined in (2) and (3).

$$C_{BW_{rtmp}} = (C_{T_{out}} \cdot BW_{web-out} + C_{T_{in}} \cdot BW_{web-in}) \cdot t \cdot 3600$$

$$C_{BW_{recording}} = C_{T_{out}} \cdot t \cdot BW_{web-isabel} \cdot 3600$$

2) *Costs of the hybrid architecture:* The second architecture is oriented to reduce the cost separating those components that use more CPU from those that consume more bandwidth. Mainly we have to take into account the increasing number of users. The CPU cost is the result of summing all the machines that are required in a session. Although there are more machines in the global system, the rest of them are going to be started in the internal datacenter of the University, so they are part of the cost of Cloud resources. Consequently we have three machines that have different requirements of CPU and memory. We can see this on figure 3.

$$C_{cpu} = 2 \cdot C_{tr} \cdot t$$

The result is the cost of all the CPUs per second in a videoconference session.

The cost of Isabel traffic in this case is the main consequence of all the internal traffic in the Cloud infrastructure,

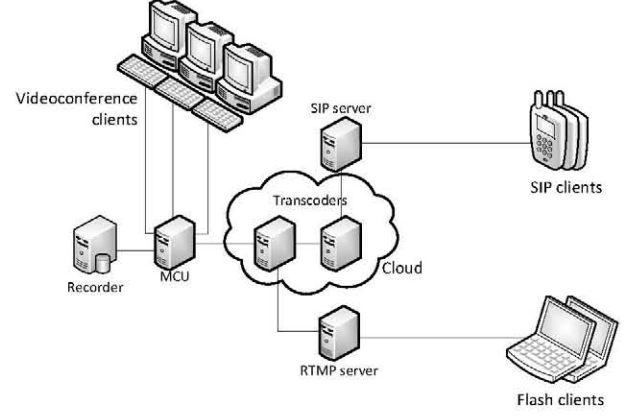


Fig. 3. Hybrid Architecture

plus an outgoing connection that lets one transcoder and the MCU be connected and forward the traffic towards other Isabel clients.

$$C_{BW_{isabel}} = (C_{T_{in}} + C_{T_{out}} + 2 \cdot C_{T_{internal}}) \cdot BW_{isabel-user} \cdot t \cdot 3600$$

The cost of the traffic originated by other web clients is left out because in this case there is only one web connection.

$$C_{BW_{rtmp}} = (C_{T_{out}} \cdot BW_{web-isabel} + C_{T_{in}} \cdot BW_{web-in}) \cdot t \cdot 3600$$

The cost of the traffic that is generated in order to record the session only has one direction as shown in the following formula:

$$C_{BW_{recording}} = C_{T_{out}} \cdot BW_{web-out} \cdot t \cdot 3600$$

Figure 4 depicts the differences between both architectures and how the cost remains constant in spite of the increasing number of web users for the hybrid case, which comes as a result of applying the methodology presented in section III. This behavior is due to the deployment of part of the resources in the data center of our University, in which we have some old machines that can do a light work but that are connected via a high bandwidth connection to the Internet. It shows how the resources used in a session can be reallocated in different clouds or data centers in order to develop a cost effective architecture. For the calculus we have taken the price of the Amazon EC2 resources applicable at the time of writing this paper.

C. Example scenario

In order to validate the formulas described previously and confirm the stability of the hybrid architecture as well as to quantify the cost requirements for a videoconference session over such architecture, we have made a simple real scenario by simulating one version of the architecture represented on figure 3. We have established three scenarios, all of them including six computers to participate in a videoconference using the Isabel environment and two virtual machines placed in the

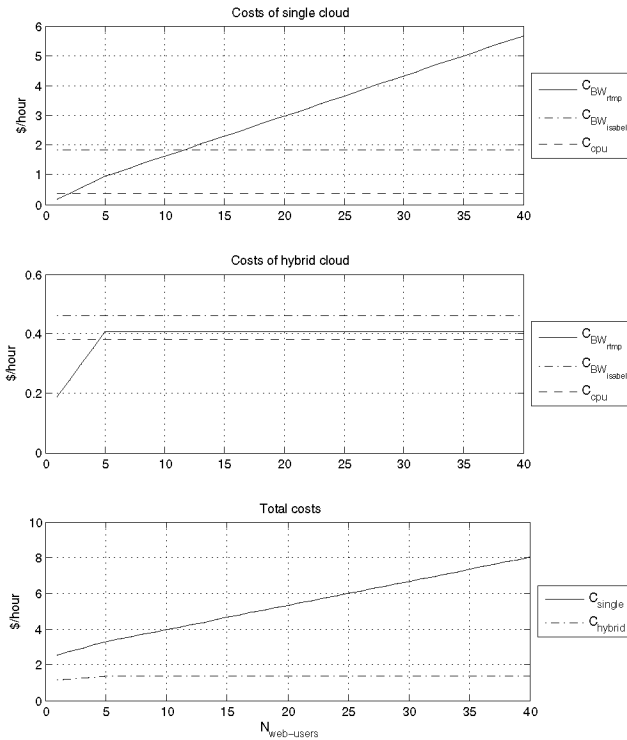


Fig. 4. Cost Comparison

Amazon EC2 cloud environment. Two medium EC2 instances have been chosen to host the machines' images where Isabel's components MCU and gateway was set up and preconfigured. The tests were divided in three scenarios differentiating by the type of traffic being sent during the videoconference.

Scenario 1 Consisted of five Isabel physical machines used as clients and one that connects to the session using an Isabel client via the web portal GlobalPlaza instead of connecting directly to the server machines. All these clients connect to the Isabel server machines in the EC2 cloud by setting up the DNS of the Amazon instances. This was the basic scenario due to the Isabel "chat mode" (enabled participation in the videoconference without any interaction) set up in the client machines that uses less bandwidth in the IN and OUT traffic. The tests were set up to one hour duration.

Results: It was verified an accurate functioning of the Isabel clients as well as a 100% stability of the Isabel servers in the cloud throughout the entire hour. We could have noticed very low range or almost no delays in the video and audio, that didn't affect the overall communication. Also no interruptions or jitter was registered in none of the clients. The costs we approximately calculated were from a range of 0.5\$ per hour. From here we have confirmed a correct execution of the entire session for this scenario. Moreover it was proven that by establishing a hybrid architecture of servers hosted in the cloud and clients (peers) outside, applied over videoconference real example scenario contributes to the stability of the session and achieves a reasonable cost model.

Scenario 2 Consisted of five Isabel physical machines used as clients and one that connects to the session through the web portal GlobalPlaza. All these clients connect to the Isabel server machines in the EC2 cloud. For this test, one of the clients was set up in "VNC mode" i.e., using a VNC desktop to be able to interact during the videoconference by showing slides or videos. The rest of the clients have stayed in the "chat mode". This test lasted for two hours. The first hour the VNC was set up to repeatedly show a video and the second hour it was showing slide presentation. In both cases the CPU usage by this client was supposed to rapidly increase together with the bandwidth traffic.

Results: The results were the same as in the first experiment showing no significant delays in any of the clients, neither in the one using the VNC mode. The Isabel servers stayed on during the two hours. As for the delays of the videos and the audios, again no distortion whatsoever was spotted to affect the communication. The usage of the VNC has incremented the costs, especially the VNC video which even in this case have stayed in reasonable limits [0.8\$-1.3\$ per hour] with respect to the time and the load.

Scenario 3 Consisted of five Isabel physical machines used as clients and one that connects to the session through the web portal GlobalPlaza. All these clients connect to the Isabel server machines in the EC2 cloud. Here we have replaced the "VNC mode" in one of the clients with "VGA mode" i.e., show videos in VGA resolution. The rest of the clients have stayed in the "chat mode". This test lasted for two hours. The first hour the VGA client was set up to repeatedly show a video and the second hour it was showing slide presentation. Also in this experiment was expected that in both cases the CPU usage by VGA client will rapidly increase together with the bandwidth traffic.

Results: The results didn't differ much from the ones obtained in scenario 2 related to delay and jitter in the video and the audio. EC2 showed again to be stable in hosting the Isabel servers during the entire time. The usage of VGA video has incremented the OUT traffic but not much more than the VNC mode. Costs in this case stayed lower than in the scenario 2 [0.7\$-1\$ per hour]. VGA video has logically measured higher cost than VGA slides presentation.

Apart from these simulation scenarios, at the moment of writing this paper we have already used a model of the hybrid architecture in several real case scenarios, such as videoconferencing events hosting users from different countries. There were enabled two main types of participation: the first one permitted joining the session in order to talk and make presentations, the second one was just watching the conference through web video streaming with the portal GlobalPlaza.

Figures 5(a) and 5(b) show several connections made around the world in order to assist to the Rails Conference 2010 and the Isabel Workshop 2010. These were real life scenarios that used the described hybrid architecture with Amazon EC2 cloud. These sessions were configured to use 1MB of traffic for each Isabel connection at most, and 500KB for each Web connection. This examples has confirmed the architecture to



(a) Rails Conference 2010



(b) Isabel Workshop 2010

Fig. 5. Videoconference real scenarios

be a cost-effective and stable solution used in an interactive videoconference streaming session as an example use case.

VI. CONCLUSIONS

In this paper we have shown how a videoconference system can be greatly improved with usage of hybrid cloud resources. We have gone through this by establishing the advantages of partitioning the system into CPU, bandwidth and store intensive parts. Finally we have validated our claims with three experimental scenarios and a videoconference scenario where the use of a hybrid cloud architecture results in much lower cost than using only the initial establishment of the architecture with local physical resources. We have measured a cost interval of [1.5\$-2\$] for one hour necessary to host an Isabel videoconference session with five machines participating set up in: chat mode, VNC with video, VNC with presentation, VGA with video and VGA with presentation respectively.

We think that other systems can benefit from this idea by using a virtual infrastructure manager and a scheduler that starts the resources required for a service in the most appropriate provider based on location, price and resource type.

We intend to further develop this architecture, make it as general as possible and abstract away the details of the system

being managed by the scheduler and the executor. Furthermore we are interested to make a relation between the resources used in one videoconference and the final cost of the established session. With this we would be able to previously calculate the approximate cost of the videoconference and thus organize the nodes and the resources in the most suitable manner.

REFERENCES

- [1] Use Case Discussion Group, "Cloud Computing Use Cases v2.0," p. 45, 2009.
- [2] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Computing*, vol. 13, no. 5, pp. 14–22, 2009.
- [3] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," RFC 2543 (Proposed Standard), Internet Engineering Task Force, Mar. 1999, obsoleted by RFCs 3261, 3262, 3263, 3264, 3265. [Online]. Available: <http://www.ietf.org/rfc/rfc2543.txt>
- [4] J. Cerviño, P. Rodríguez, J. Salvachúa, F. Escribano, and G. Huecas, "Marte 3.0: Una videoconferencia 2.0," 2008, pp. 209–216.
- [5] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," *IEEE Internet Computing*, vol. 13, no. 5, pp. 14–22, Sep. 2009. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5233608>
- [6] Y. Zhang, G. Huang, X. Liu, and H. Mei, "Integrating resource consumption and allocation for infrastructure resources on-demand," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 75–82.
- [7] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: comparing public cloud providers," in *Proceedings of the 10th annual conference on Internet measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010, pp. 1–14. [Online]. Available: <http://doi.acm.org/10.1145/1879141.1879143>
- [8] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "Above the clouds: A berkeley view of cloud computing," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28*, 2009.
- [9] N. Leavitt, "Is cloud computing really ready for prime time?" *Growth*, vol. 27, p. 5, 2009.
- [10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), Internet Engineering Task Force, Jun. 2002, updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954. [Online]. Available: <http://www.ietf.org/rfc/rfc3261.txt>
- [11] J. Quemada, G. Huecas, T. de Miguel, J. Salvachúa, B. Fernandez, B. Simon, K. Maillat, and E. Lai-Cong, "Educanext: a framework for sharing live educational resources with isabel," in *WWW (Alternate Track Papers & Posters)*, 2004, pp. 11–18.
- [12] J. Quemada, T. de Miguel, S. Pavon, G. Huecas, T. Robles, J. Salvachua, D. Ortiz, V. Sirvent, F. Escribano, and J. Sedano, "Isabel: An Application for real time Collaboration with a flexible Floor Control," in *2005 International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Ieee, 2005, pp. 1–9. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1651238>
- [13] R. Fielding, "Representational state transfer (rest)," *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine, 2000.
- [14] P. Rodríguez, D. Gallego, J. Cerviño, F. Escribano, J. Quemada, Salvachúa, and J. A., "VaaS : Videoconference as a Service," in *CollaborateCom*, Washington, 2009.
- [15] C. Cavaness, *Quartz Job Scheduling Framework: Building Open Source Enterprise Applications*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2006.