Abstract - The current generation of commercially available Web conferencing systems is limited by the number of simultaneous users that can join a Web conference. This number is often limited to a few thousand users at the most. Recently, mconf, a distributed open source-based web conferencing system developed by Brazilian NREN team has been deployed primarily for the educational use. In this paper, we discuss the design of distributed Web conferencing system like mconf that can support a very large number of simultaneous users. Various deployment issues with server load balancing, bandwidth management and streams management are discussed.

Keywords: Web conferencing, Webinars, Network traffic

1 Introduction

Web conferencing is an interactive online seminar held by a presenter and attended by an audience. Web conferences usually follow an agenda and provide auditory and visual access to the information being shared. Typically, the voices of those attending the Web conferencing are muted during the presentation so that the presenter's message comes across clearly; however, the presenter can take polls or interact with the audience, let people ask questions, and more. Alternatively, the presenter may invite live questions at the end of the Web conference. Similarly, a webcast is a presentation without the interaction.

Web conferences are increasingly being used to deliver marketing material, training and other announcements to captive, geographically dispersed audiences. A number of commercial companies exist that offer commercial Web conferencing capabilities. However, most of these systems only scale to a few thousand users at best. The capacity of GotoWeb conferencing, run by Citrix Corporation, is 1000 users [1]. The capacity of Webex, owned by Cisco Systems is 3000 users (a maximum of 500 participants only can see live, shared video of presenters or panelists, in video-enabled events) [2]. Adobe Connect Server [8] can support up to 1500 users in a meeting, but it is suggested that a Server cluster be used with one server for every 500 concurrent users. Most other commercially available systems do not scale to large number of users.

A typical university may have several thousand students. Similarly, many corporations have tens of thousands of employees. Corporations often need to make policy announcements. To address such a scenario, the companies are forced to use overhead broadcasting systems or invite people to conference rooms or cafeterias. These methods are not desirable since confidential information is often heard by everyone present in the building or there is no interactivity that presenters may like to have. Also, with the cost of education rising, it will be useful to deliver interactive lectures to large number of students across multiple campuses.

Mconf, available from mconf.org, [12] is an open-source multi-conference system built on top of the BigBlueButton open source conferencing system [13]. It is composed of a web portal that provides access to web conferences, shared documents, spaces etc., and a web conference system powered by BigBlueButton which provides the base functionality of the conferencing system. The contribution of mconf team is to add a layer of functionality to load-balance the streaming servers and other load-monitoring communications.

The distributed architecture used in mconf makes it easy to deploy distributed conferencing system using autonomous collection of servers. However, there are still many challenges to deal with if the users are distributed over large geographical areas or in countries with different network bandwidths. We discuss some of these issues later.

In this paper, we discuss a distributed architecture for scalable Web conferencing systems similar to that deployed in mconf. It will allow Web conferencing systems to scale to tens of thousands of users. The rest of paper is organized as follows. In Section 2, we describe the features of a typical Web conferencing system. In Section 3, we discuss the scaling dimensions of web conferencing systems and some benchmarking of single
server systems. In Section 4, we present a distributed scalable architecture for deployment of a scalable Web conferencing and conferencing system. In Section 5 we describe bandwidth considerations for scaling Web conferencing systems. In Section 6 we discuss intra-operability and operational issues. Finally, we summarize the paper and discuss future work.

2 Features of a Typical Web Conferencing System

Modern web conferencing systems are expected to provide a wide range of features. Web conferences are just like a conference room-based seminars; however, participants view the presentation through their Web browsers and listen to the audio either through their telephone or via computer speakers. A key feature of a Web conferencing is its interactive elements: the ability to give, receive and discuss information in real time. Most modern Web conferencing systems present a broad range of features. Some features are easily scalable while others are not. Therefore, any architecture that is presented must be able to address most, if not every desirable feature.

All Web conferencing systems provide a similar set of base media broadcast features enhanced by proprietary reporting and event management features. Some of the currently available features are:

- Support for large audiences (100+)
- Registration pages
- Screen and applications sharing
- Document and File sharing
- Chat for asking questions
- Whiteboard
- Collecting attendee feedback
- Polling
- Q&A
- Recordings
- Practice mode
- Voice over the Internet
- Video
- Streaming media
- Reporting (attendance, chat transcript)
- Event management (reminders, planning, invitation)
- Security for User Systems

Therefore, scaling requires the ability to scale the delivery of all the above features.

3 Scaling Dimensions of Web Conferencing

The scaling a web conferencing has two distinct dimensions:

i) Host large number of conferences at the same time:

This goal is easily achieved by many commercial systems and is only limited by the number of streaming servers in the network. Each server can only support limited users. So, once that limit is known then scaling is achieved by simply replicating the servers. However, using a single cluster located at one location is not sufficient if the users are distributed over a large area. One of the key challenges of web conferencing systems is to minimize jitter and delays for all receivers to achieve a satisfactory media stream. Adding a suitable load monitoring layer can help minimize the cost.

ii) Increase the limit of users in a single web conference: This is the ability to host a conference and deliver content to a very large number of simultaneous users. This is a much harder problem. It implies enabling a single meeting to be held using more than one streaming server.

Video streaming is a highly demanding application. In a Bluebutton benchmark the following results have been obtained [11] using a server Xeon 3450 quad core 2.66 GHz with hyperthreading:

- 20 voice users and 20 webcams: 30% CPU
- 40 voice users and 20 webcams: 45% CPU
- 60 voice users and 20 webcams: 70% CPU
- 80 voice users and 20 webcams: 90% CPU

It is clear that scaling a web conference for users that come from a large geographical area can only be done by using distributed server systems.

4 Distributed System Architecture for Scalable Web Conferencing

As mentioned earlier, mconf uses a distributed deployment of servers with a load balancer. In mconf, the web conference servers in the network are shared: every institution that is a member of the network might use any server in it. The server that will be used is decided in the moment a web conference is created and this decision is based in several factors. The server selected might be, for example, the one that is nearest to the user creating the conference, or the server that has more resources available (e.g. CPU and memory) in the moment.

To create a distributed and large-scale conferencing system, one can use a similar architecture. While this architecture works, there are some specific challenges in making a distributed, autonomous systems work. These issues are discussed later.

To scale a single instance of Web conferencing system, as in the case of mconf, we need to add three core components:
i) Replicated web servers with the redundant load balancers: Web servers are used for interaction with the end users in activities such as Web conferencing reservations, finding Web conferencing to join and all other administrative functions like password verification etc. We call these functions Meeting Management (MM) functions.

ii) Replicated media streaming servers: A media server can only serve a limited number of streams. Therefore, replication of media streaming servers is very important for scaling and availability. The key aspect of replicating media servers is to create a chain of media streaming on demand from one server to another.

iii) A protocol to distribute users to various streaming servers and do the dynamic real time load balancing: Since each streaming server has limitations as to the number of streams it can serve, it is important for the system to keep track of the load on each server and update the user clients as to which streaming servers a user client should connect. The users often join and leave a Web conferencing at random times. This means that any load balancing of server and users should be done in real time, requiring the meeting manager managing the Web conferencing to be aware of all users that drop out of a Web conferencing, along with the load on each server so that optimum load balancing can be achieved.

The load balancing process interacts with the client to collect the data that it considers in media server allocation. Two basic steps are followed:

1. Provide a list of possible (and available) media servers nearest to the client's physical location to the user client.

2. Get the measurement of end-to-end delay, and the link bandwidth between client and each media server in this list.

It is not necessary to rely on one particular load balancing method. For example, mconf load balancer implements several algorithms, but only one in use [12]. Some of these algorithms are:

- Select the server with less CPU load;
- Select the server with fewer users;
- Select the server that is geographically nearest to the client.

Since all servers need to implement a similar criterion, it is better to implement a single method. For example, mconf uses the following algorithm [12]: Order the servers by proximity to the client. If there is more than one server in a ~300km radius, select the one that has less CPU load in the moment. It does not consider servers that are not up or that are not responding properly.

After considering the data supplied by the client, a media server is assigned to the client for joining a live stream. Figure 1 shows the architecture and its various components.
Management System (MMS) servers via a server load balancer (SLB). The SLB is used to distribute incoming HTTP requests amongst a pool of MMS Servers. The SLB can be configured to allocate requests in a round robin fashion or based on MMS server load.

The SLB monitors the MMS servers, and removes them from the pool in case of failure. The SLB itself has a backup in case of failure. The use of the SLB thus promotes availability, but it also promotes scalability, by making it easy to add new servers without impacting any system components beyond the SLB. In addition, all MMS servers are stateless, meaning that any server can take over from any other server in case of a failure.

The Media Servers implement the communication for individual meetings: screen sharing, VoIP, video, file and chat data. The MMS Server makes the decision as to what Media server is to be used for a given meeting. MMS Servers load-balance Media servers, and remove Media servers from their available pool should Media servers become unavailable. This provides both scalability in the number of meetings that can be hosted, and failover capability should a Media server cease to operate correctly.

Finally, conference data (such as account information, schedule information, meeting status and billing data) is stored on a database server. To provide failover capability, a backup database server mirrors the database server in real time.

4.1 Architectural Tradeoffs in Distributed Media Servers

Any architecture embodies tradeoffs. Distributing servers at many locations may not be possible because of cost considerations or lack of availability of high speed bandwidth. In Web conferencing systems, it is important to keep the cost down. Therefore, if it is possible to use servers that can be provisioned on demand then the cost can be reduced without affecting scalability. However, centralizing all servers at one physical location may lead to degrading of the media experience for some users. Therefore, a server allocation must consider the delay between the clients and the servers in considerations when assigning users to a particular media streaming server.

4.2 Integrating Telephony Voice With Media Servers

While most users join a web conference using a computer system with attached microphone and speakers, some users may like to join a phone-only connection. Sometimes, even users with PCs and desktops may like to have their audio delivered via a phone connection.

The integration of VOIP conferencing services and the Media server is done using RTMP. VOIP servers such as open-source Asterisk or Free Switch are integrated with media streaming servers. Their phone switch provides a RTMP channel to the media servers. The quality of voice using a phone will depend on the general call quality between the VoIP switch and the phone.

4.3 Dealing with Difference in Bandwidths in Different Areas

One of the challenge in delivering a web conference to a large number of users is to deal with different bandwidth available to users. For instance, the users in the US have access to 25 Mps connections at home using fiber or cable. However, users in developing countries only have access to a 100 to 500 kbps links in their homes. This means that a video stream that is easily received and played by the users in the US cannot be received by users in other countries due to lack of bandwidth.

This problem can be addressed to some extent by the use of scalable video encoding. Scalable Video Coding (SVC) is the name for the extension of the H.264/MPEG-4 AVC video compression standard. SVC standardizes the encoding of a high-quality video bitstream that also contains one or more subset bitstreams. A subset video bitstream is derived by dropping packets from the larger video to reduce the bandwidth required for the subset bitstream. The subset bitstream can represent a lower spatial resolution (smaller screen), lower temporal resolution (lower frame rate), or lower quality video signal.

This allows users in the low bandwidth areas to display only one layer (poor quality) and the users in high bandwidth area to display all the layers (high quality video).

5 Media Server Bandwidth

A media streaming server is the most important part of the Web conferencing system. The scalability of a media server can be computed by estimating the bandwidth required for Web conferencing.

We describe a method for estimating bandwidth for an application such as video conferencing or Web conferencing as described in [6]. A simple formula for calculating bandwidth is used for both the server bandwidth (which is required to determine the approximate number of server instances that the system needs) and the client bandwidth (which should be calculated as part of overall bandwidth strategy to insure...
good quality of service to each client).

Clients connect directly to a Web conferencing Media Server which handles on demand and real time video, audio, and data delivery. As the number of clients increases, the resulting high bandwidth requirements may potentially increase the latency during the Web conferencing. As discussed earlier, Web conferencing Media Servers need to be provisioned in the proper number and at the proper locations to ensure that the bandwidth requirements are satisfied.

In the case of Web conferencing, the key parameter is the maximum number of clients attending at each major location. Since clients usually need to register before the Web conferencing, the organizer can easily estimate attendance. The following analysis examines the bandwidth requirements at a single location relying on experimental results from [9] and [10].

Bandwidth requirements depend on the type of usage necessary for the Web conferencing such screen sharing, audio, webcam video and streaming video. Experimental results in [9] indicate that typical webcam video bandwidth per client is 239 Kbps and 501 Kbps depending on whether the client is using a DSL or LAN connection respectively. It is assumed that the Web conferencing Media Server is aware of the type of connection the client uses and adjusts the webcam video parameters for each client accordingly. Specifically, the DSL bandwidth results from webcam video of 640x480 at 15 frames per second (fps) and compression quality of 85; the LAN bandwidth results from webcam video of 640x480 at 20 fps and compression quality of 90.

Similarly, experimental results in [10] indicate higher typical webcam video bandwidth per client of about 900 Kbps for LAN connections due to higher-quality compression despite a lower resolution of 640x360. Results in [10] also show that webcam video bandwidth is highly dependent on whether the subject is moving. The bandwidth of 900 Kbps was observed with a slowly moving subject while a stationary subject resulted in a significantly smaller bandwidth. ([10] reports a reduction of bandwidth from 308 Kbps to 58 Kbps using a lower video resolution of 320x180).

Depending on quality and frame rate, the bandwidth needed for screen sharing has been observed to be between 138 and 164 Kbps for a LAN connection and slightly less for a DSL connection [9]. VoIP audio bandwidth was observed at 44 Kbps for a LAN connection and 22 Kbps for a DSL connection.

Bandwidth for streaming video plus audio was observed to be approximately 1200 Kbps (with small variation depending on the file format).

The total bandwidth requirements per client are approximately 700 Kbps for a LAN and 400 Kbps for a DSL connection, without the use of streaming video. Given the experimental results in [10], a more realistic estimate of the needed bandwidth for a LAN connection would be 1000 Kbps. Assuming that a Web conferencing Media Server has a 1 Gbps network connection available, the total number of clients that could be supported by a single server would be approximately 1000 clients, all using LAN connections, to 2000 clients, all using DSL connections. If streaming video (at 1.2 Mbps) is also used by the Web conferencing, then these bandwidths are reduced to approximately 450 clients, all using LAN connections, and 600 clients, all using DSL connections. For a mixture of LAN and DSL clients, the required bandwidth would be between these two endpoints.

Even if the network connection bandwidth available to a Web conferencing Media Server at a single location Web conferencing Media Server is 1 Gbps, conservative estimates suggest that a single server may be able to support up to 500 clients. When network connection bandwidth allows more clients to be connected, then two or more servers would be utilized to support all the connected clients.

6 Intra-operability and Operation Issues

Many conferencing systems are available today but there is no interoperability between them. In order to accomplish interoperability among different vendors, common standards must be followed. The two most prominent standards are H.323 (ITU-T H.323, 2009) and SIP (Session Initiation Protocol), and both are used worldwide to allow interoperability among different systems.

The basic idea of these protocols is to have some common messages to initiate the session and exchange capabilities. Using the information exchanged, the different parties in the conference can agree on the same group of audio and video codecs, as well as the data protocol, allowing the interoperability. In other words, if a distributed system on geographical large scale is to be deployed all systems must use the same software for signaling and load balancing.

This leads to another problem common to all distributed systems. Since systems interoperate and connect with one another, the software change in one
server affects all other. If the network consists of 1000 nodes, changing software on all 1000 nodes at the same time requires special software tools and logistics.

7 Conclusions

In this paper we presented various aspects of a Web conferencing and conferencing system. We also proposed a system architecture that can be used to deploy a highly scalable Web conferencing delivery architecture. Web conferencing delivered to large, geographically dispersed audiences, can benefit by relying on a distributed architecture of media servers provisioned on demand and conveniently located so as to reduce cost and maintain scalability. Our analysis shows that individual media servers can support between 450 and 2000 clients depending on video requirements of the Web conferencing and the network bandwidth available at each client. Therefore, a small number of media servers provisioned at the proper locations can readily support large audiences.

The system described can be built using a Flash Media Server (Adobe Flash Server, Red5, Wowja Flash Media Server), and adding client software and load balancing modules. A simple prototype can be developed using mostly open source components and integrating open source VoIP telephone systems such as Asterisk or FreeSwitch.

8 References