INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 3 - V7I3-2087)
Available online at: https://www.ijariit.com

A study on Web conferencing system and the deployment

Sudarshan M.
sudarshan.is17@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

Harith L. K.
harithlk.is17@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

K. Vadhi Raja
kvadhiraja.is17@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

Pranava B.
pranavab.is17@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

Dr. G. S. Mamatha
mamathags@rvce.edu.in
RV College of Engineering, Bengaluru, Karnataka

ABSTRACT

In the current scenario due to the spread of COVID-19, there has been an increase in the use of web conferencing systems to communicate among people. Web conferencing systems have enabled organizations, universities, and individuals to communicate over the internet from their homes during the current pandemic. Web conferencing system not only allows people to communicate over the internet, but it also has other features which enable users spread across different regions to collaborate like whiteboard, polls, chat, and others. As the number of users is increasing for such systems, scaling and load balancing becomes vital to handle all the users. This paper presents an exploration of the web conferencing systems and explores a case study of an open-source web conferencing system and scaling of the respective system to meet the demands from the system.

Keywords— *BigBlueButton, Greenlight, Scalelite, AWS, CloudFormation, Automation, Load-Balancing, Cloud*

1. INTRODUCTION

The Internet has changed perspective on meetings, online classes, webinars, and other interactions which happen online. These systems provide both modes synchronous and asynchronous communication and enable virtual connections that strongly resemble in-person interactions [1]. Web conferencing systems include features such as screen sharing, whiteboarding, polling, public chat, private chat, recording etc. which allow a seamless real-time interaction between the users of the system [2].

With the recent outbreak of COVID-19, regulations to restrict physical meeting, mobility and public life were imposed. Due to this, organizations and educational institutions adapted to virtual interactions [3]. It has enabled people from different places to connect, help teachers conduct classes online and employees to work from home.

BigBlueButton is an open-source web conferencing platform which provides us with basic features of a Web Conference tool [4]. It also provides other services like whiteboarding, polling and recording [4]. BigBlueButton provides high quality audio and video communication through WebRTC which enables high quality media communications directly between browser and devices [4]. As the number of users are growing, a server cannot handle all the users.

Given the current situation, many users are shifting to online meetings, classes, webinars etc. As the number of users are more, a server cannot handle all the users. Having a web conferencing system running as a standalone server does not power the system to accommodate increasing number of users in a meeting.

In this paper, we focus on the open-source web conferencing system and how scaling of the server load is done by using an open-source load balancer called Scalelite to allocate the increasing user demand.

2. ARCHITECTURE OF THE WEB CONFERENCE APPLICATION

The web conferencing system has several components as shown in the figure above. HTML5 client is a simple single-page responsive web application that is built using React.js to provide a good user experience. This is where the WebRTC JavaScript APIs are accessed to provide end-to-end audio and video transfer in the application [3].

HTML5 server is built upon Meteor.js, which is a framework for Mobile, Web, and Desktop applications. This framework is used to enable the connection between the HTML5 client and server [3]. This is also connected to a MongoDB database which contains all the information related to the meetings running on the server, the user clients connected to the meeting [3].

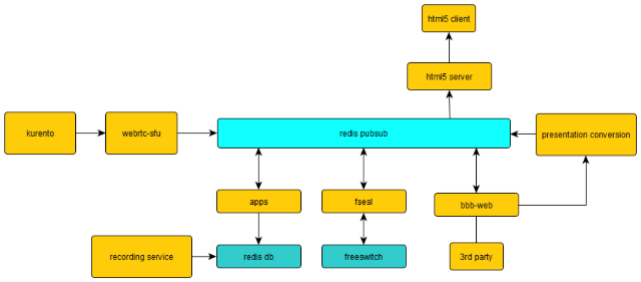

Fig. 1 Architecture of the web conferencing system

The server also verifies if the users have permission to access all the collection data from Redis.

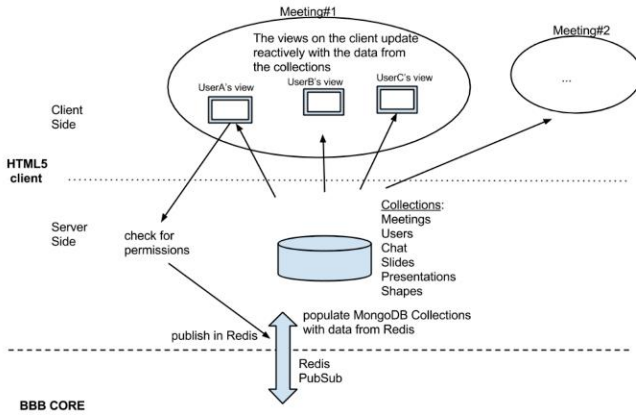

Fig 2. HTML5 client communication with other BigBlueButton components

2.1 Components of BigBlueButton

The main aspect of customising BigBlueButton is to analyse and understand how various components of BigBlueButton interact with each other, which makes it easier to develop the required features in the application. The various components of BigBlueButton are explained.

2.2 BigBlueButton HTML5 Client

The client-side responsive single page, which is built upon WebRTC for peer connection, all the media functionality (audio/video) and ReactJS in order to render the HTML elements used in the application. The HTML5 client interacts with the BigBlueButton server by establishing a connection over the port 443. These connections are made through web-sockets and nginx is configured to handle these connections [3].

2.3 BigBlueButton HTML5 Server

The HTML5 server runs as the backend server for nginx to create the required connections and configure the application to be run. This HTML5 server is built by JavaScript and developed using MeteorJS framework which is used to establish and connect the client with the required server. The HTML5 server is also connected to a database MongoDB, which is used to store the client communication information and thus synchronize with the BigBlueButton server [3].

The database used here (MongoDB) stores the details related to the meetings run on the server, and the clients connected to that meeting and thus records are maintained by the application and can be retrieved in the form of collections.

2.4 BBB Web

BigBlueButton web component is developed based on Java which is coded in Scala. This component handles the meeting state, it contains a copy of the same. This component also handles all the API implementations of BigBlueButton. The API components of BigBlueButton are configured to handle all the third-party integrations, the API exposes an endpoint to be connected which can be then configured to control the BigBlueButton server.

2.5 RedisPubSub

The whole application needs a particular channel to communicate, this channel is provided by RedisPubSub for the applications to connect over different BigBlueButton servers [5].

2.6 Redis DB

Redis is generally used as a cache storage, which stores all the temporary information or real-time information about an event, in this case, the meeting. Redis DB is configured to store the recordings initiated in the BigBlueButton application; the events which occur during the recording process are also stored in RedisDB in raw formats like a PDF, an Adobe Flash video (.FLV), or a waveform file (.wav) which are then further processed [3].

2.7 Apps Akka

BigBlueButton apps consist of various other components which provide the collective functionality of real-time meeting and other add-on features of the application like the list of meeting attendees, a public chat option, a collaborative scratch board for rough works and textual explanations. The logic behind the meeting business is stored in MeetingActor. It also stores the messages sent during the meeting by the users and processes them [3].

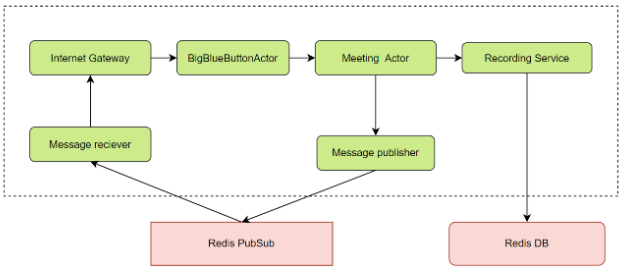

Fig 3. The Structure of BigBlueButton Apps Akka

2.8 FSESL Akka

This component is extracted and configured in a way such that it integrates with its own application. By doing this, the application would be able to allow the users who are using audio conference systems which are not based on Freeswitch to integrate with the application and communication becomes easy. The interaction between the FreeSwitch Event Socket Layer (FSESL) and the BigBlueButton apps happens via the messages which are fetched and read from the Redis PubSub [3].

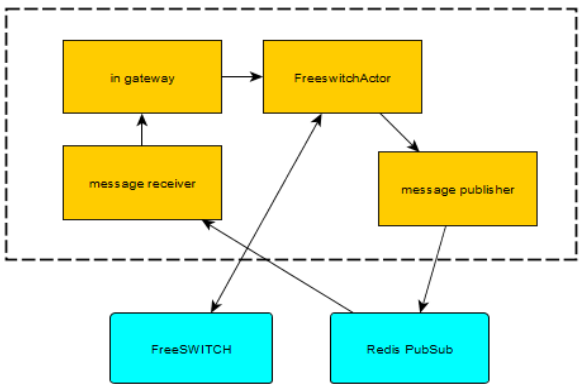

Fig 4. Components of FSESL

2.9 FreeSwitch

FreeSwitch is an integrated tool which provides the capabilities like voice input/output and conferencing in BigBlueButton. FreeSwitch allows the users who join a meeting to join the voice input and output through various media devices such as headsets, inbuilt-speaker and microphones and other external peripheral devices. FreeSwitch is also compatible with Voice over IP and thus enables the users to join a meeting via a mobile device and dial in to join the voice [3].

2.10 Kurento and WebRTC-SFU

Kurento is a framework which implements a media server which can be integrated with the SFU model and also an MCU model. This media server acts as the handler for the webcam streaming, fetching the data from users' system for screen sharing purposes. This media server also provides the user with media controller handling and also media stream management [3].

2.11 WebRTC FOR PEER-TO-PEER COMMUNICATION

WebRTC enables real time communication between browsers [6]. WebRTC provides necessary APIs that call the browser's JavaScript functions and enable effective video conferencing. It makes use of codecs for encoding and decoding the video and audio messages. WebRTC enables real time streaming of video, audio and other data. It takes care of any latency as well as loss during data transfer. WebRTC makes use of UDP protocol in order to facilitate the same.

A major challenge with peer-to-peer communication is establishing a network socket connection with another computer's web browser for bidirectional transmission of data [7]. Most computers sit behind a firewall and a NAT device and it is another issue that needs to be handled during peer-to-peer communication.

WebRTC makes use of the PeerConnection mechanism for data transfer. It uses the Interactive Connectivity Establishment protocol together with the Session Traversal Utilities to allow UDP packets to traverse NAT boxes and firewalls [6]. WebRTC has one more component called MediaStream which helps to handle the captured data locally as well as remotely [8].

BigBlueButton makes use of a TURN server which helps in creating a public IP address towards which the data packets can be directed at. BigBlueButton also uses Kurento, a media server, which helps strengthen the WebRTC ecosystem.

3. FEATURES OF THE WEB CONFERENCE APPLICATION

Contemporary web conferencing systems have many more features than just video and audio sharing with users spread across various regions. A lot of collaborative features exist in current web conferencing systems which give us the feel of having a real conversation.

BigBlueButton is an open-source web conferencing system which offers many collaborative features as well which helps the users of the system communicate the information in a more effective and efficient manner. The features this system offers are:

(a) It allows the users to turn on their microphone to transfer voice to other users which allow them to interact with one other.
(b) Webcams can be enabled which transmits video signals from webcam to other users, which allow them to see the users who have their webcams on.
(c) Public chat to chat with other users in the conference using text messages.
(d) Private chat which allows users to communicate with selective members of the conference which is totally hidden from the other users in the same conference.
(e) Polling feature allows the users in the conference to multiple choice questions in the conference where other users can select and results can be obtained in percentage. This feature can be used to make decisions based on the users' opinion or can be used to evaluate users for different questions posted.
(f) Whiteboard allows users to write on the screen, draw shapes or add text on a blank screen which helps to collaborate with others which helps in communicating the concepts better.
(g) Breakout rooms allow the moderator of the meeting to separate out the users into separate meetings for private discussions among different groups of members.
(h) Shared notes feature allows all the users in the meeting to share a single document which can be edited by others in the meeting.
(i) The recording feature lets the users record the meetings and the recordings can be played from within the same environment which makes the recordings easily accessible and requires no plugins to be installed to play the videos.
(j) The meeting owner also gets additional features such as muting other users in the meeting and giving permissions to present or permissions which the meeting owner has. The owner can also set the joining policy for the meeting to set the permission to enter the meeting such as ask to join, always accept or always decline. It also lets the meeting owner restrict individual features like sharing webcam, microphone and other things by locking them for users.
(k) Attendance reports can be generated to make a note of a list of the users who attended the meeting by the meeting owner. The system generates a text file with the list of users sorted by name.
(l) The system also lets users set status like raise hands, happy, sad, confused, and other reactions to a message.

4. SCALING THE WEB CONFERENCING SYSTEM

A web conferencing system should be able to handle many users as in case of a college or university conducting multiple meetings simultaneously. It is not possible to handle all the load by a single server running and it will degrade the performance of the server for all the users using the system [9]. Scaling is an important aspect in web conferencing systems which will enable us to handle large numbers of users. There are two ways of scaling in general, vertical, and horizontal. Vertical scaling refers to increasing the capacity of the existing resources to handle a larger load on the system [9]. This is not usually a good practice as this will usually require bringing down the existing server or system and replacing it with a better server or system [9]. Horizontal scaling refers to adding servers or systems of lighter capacity as and when required. This will enhance the performance of a web conferencing system as the existing servers or systems keep running while new servers or systems are added. In this section, we will look at a load balancer specifically designed for BigBlueButton, which will horizontally scale the requests which are made to BigBlueButton servers.

Scalelite load balancer is used to scale and manage the pool of BigBlueButton servers. It works in a way such that the pool of

BigBlueButton servers appear as a single very scalable server. Frontend manager applications like Greenlight is used which is a simple UI for the BigBlueButton server. Greenlight sends API requests in BigBlueButton to the Scalelite load balancing server which distributes the API requests to the least loaded server in the pool.

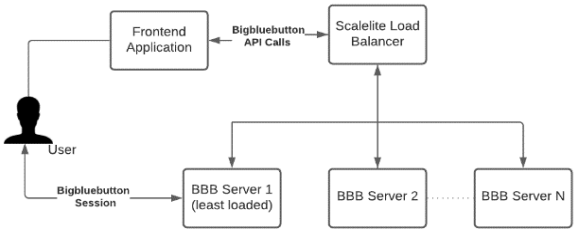

Fig 3: Load Balancing design

A single BigBlueButton server supports about 200 concurrent users in the meet. Using Scalelite, this number can be increased significantly. E.g., if a Scalelite pool contains 10 BigBlueButton servers then it will support about 2000 concurrent users, which is also equivalent to 20 concurrent meets of 50 users each. By adding more servers to the Scalelite pool, it can be scaled higher. When there is an incoming API request to the Scalelite, then it will poll each BigBlueButton server in the pool to determine which servers are reachable online, the load that each of the servers currently have and checks if the servers are ready to receive API requests and places the incoming API requests on the least loaded BigBlueButton server in the pool. Scalelite has different components like:

- Pool of BigBlueButton Servers
- Load Balancer Server
- NFS Shared Volume
- PostgreSQL Database
- Redis Cache

Multiple BigBlueButton servers for the Scalelite pool can be setup by following the instructions in official BigBlueButton documentation. Scalelite server will have a Spool Directory which is mounted on each of the BigBlueButton servers in the pool. The recording files from the meetings are stored at the mount point on the BigBlueButton Server. Then these published recording files are transferred from the mount point on the BigBlueButton Server to the Central Storage System used by Scalelite [10]. PostgreSQL database can be setup to handle the meeting related data such as login information and meeting schedules. Redis will hold the Instance ID of application, its Public URL, Shared Secret, Internal Meeting IDs, and the relevant metrics. It can act as the datastore from which the Scalelite gets to know where to forward the users when there is a meet or video conference [11].

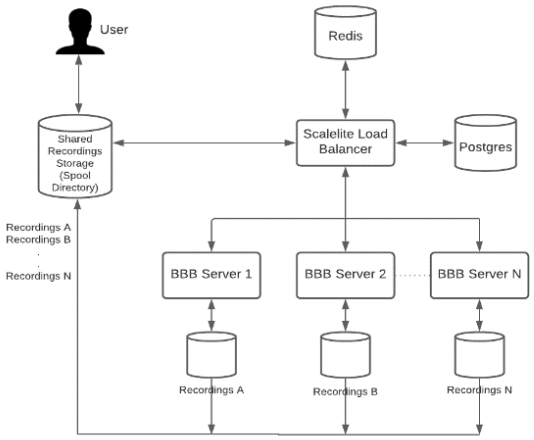

Fig 4: Storage of Recordings architecture

5. DEPLOYMENT OF THE APPLICATION ON CLOUD

A scalable BigBlueButton has three most obvious components which must be deployed in order for the system to work from end to end, the front end, the load balancer and the BigBlueButton servers. Along with these components, there are requirements for setting up database and cache for the system to function properly.

As mentioned in the previous section, scaling a web conferencing system has two different dimensions:

(a) To host large number of simultaneous meetings: This can be achieved by setting up large number of servers simultaneously and load balance the requests onto them. This will enable the system to handle large number of simultaneous meetings of a decent capacity [12].

(b) To host a meeting to accommodate a large number of users: This must be done by using a server of a larger capacity to serve the web conferencing system to a large number of users. The problem with this particular system is that increasing the capacity of individual servers has to be done by replacing it with a server of larger capacity [12].

Scaling in both dimensions proves to be very important in case of a web conferencing system as the meeting parameters are different for different purposes. One of the key benefits of cloud computing is scalability. Scaling of resources is possible using cloud computing as we can scale up or scale down the web conferencing servers as and when required [13].

Automatic Scaling scales the application based on the workload on it, by provisioning the resources on demand. This allows the system to handle peak loads by scaling up the systems when the load increases and scaling down the resources when the application does not need them. This dynamic scaling of resources helps the web conferencing system to handle the load on the system [14].

The scalable web conferencing system BigBlueButton requires several resources to be deployed in the cloud to be functioning properly. One way of doing this would be to manually deploy the server instances and the container instances in the cloud using respective resources and manually configure all the settings to connect the entire application as mentioned in the documentation for BigBlueButton and Scalelite. We could also leverage Infrastructure as Code capabilities of a cloud to deploy our resources. In Infrastructure as Code, the entire set of scripts, resources, their connection, dependencies, and other things can be provided in the form of code (templates) which can be used to deploy and configure the entire web conferencing system [15].

6. BENEFITS OF CLOUD DEPLOYMENT

There are many advantages of deploying the web conferencing system on the cloud, which are:

6.1 Scalability

The important feature required for the web conferencing system as the system needs to handle the load which falls on the system. For this purpose, we need the systems to scale based on the load which falls on them and it is possible to leverage scalability using cloud.

6.2 Disaster Recovery

Cloud providers offer functionality to backup vital data of the users and restore them in case of a disaster [13]. This offers the

web conferencing system to backup essential data such as recordings in case a failure occurs.

6.3 Cost Efficiency

Cloud offers features like multi-tenancy using virtualization which will provide a low cost on the resources being used [13]. The cloud uses the pay as you go model which will charge for only the resources being utilised. Since the web conferencing system does not have a constant load, using the cloud will only charge for the respective resources used by the system which is dependent on the load on the system (number of users and meetings).

6.4 Availability

The resources deployed on the cloud will have high availability at any time [13]. This feature makes the web conferencing system deployed on the cloud available at any time of the day which will give a better user experience.

6.5 Unlimited Storage Capacity

The cloud providers offer almost closer to unlimited storage for the users, which can be used to store the recordings done using the web conferencing system [16].

7. DISADVANTAGES OF CLOUD DEPLOYMENT

Even though the resources which are deployed on the cloud offer a lot of benefits, there are certain limitations in deploying the web conferencing system onto the cloud. The primary limitation using the cloud is Data Security. The data related to the web conferencing system such as recording and other user information will be stored in cloud storage. Unauthorized access by other users may lead to data loss or access to confidential data stored on the system [16]. Proper security rules must be enforced using the cloud provider's existing tools.

8. CONCLUSION

The development of the architecture should be feasible and should mainly consider the performance and scalability metrics. The increasing requirement of robust features in a Web conference tool increases the complexity of the application and the developers must find a way to meet the demands without drastically affecting the performance. Cloud deployment provides higher accessibility, security and scalability according to the requirement costs. While there are certain limitations to increased feature additions, the Trade-off factoring should be done to optimize performance and the level of complexity.

9. REFERENCES

[1]. Schroeder, Barbara (2004) "Microsoft Live Meeting 2007: Web Conferencing System for Virtual Classrooms," Innovate: Journal of Online Education: Vol. 1: Iss. 1, Article 6.

[2]. Nedeva, Veselina & Dineva, Snejana & Atanasov, Svetoslav. (2014). Effective E-learning Course With Web Conferencing. 10.13140/2.1.2605.6648.

[3]. Janine Hacker , Jan vom Brocke , Joshua Handali , Markus Otto & Johannes Schneider (2020) "Virtually in this together – how web-conferencing systems enabled a new virtual togetherness during the COVID-19 crisis", European Journal of Information Systems, 29:5, 563-584, DOI: 10.1080/0960085X.2020.1814680

[4]. "Open-Source Web Conferencing", [online] Available: https://docs.bigbluebutton.org/.

[5]. R. Baldoni, M. Contenti, S. T. Piergiovanni and A. Virgillito, "Modeling publish/subscribe communication systems: towards a formal approach," *Proceedings of the Eighth International Workshop on Object-Oriented Real-Time Dependable Systems, 2003. (WORDS 2003).*, 2003, pp. 304-311, doi: 10.1109/WORDS.2003.1218097.

[6]. G. Suciu, S. Stefanescu, C. Beceanu and M. Ceaparu, "WebRTC role in real-time communication and video conferencing," *2020 Global Internet of Things Summit (GIoTS)*, 2020, pp. 1-6, doi: 10.1109/GIOTS49054.2020.9119656.

[7]. K. I. Zinnah Apu, N. Mahmud, F. Hasan and S. H. Sagar, "P2P video conferencing system based on WebRTC," *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2017, pp. 557-561, doi: 10.1109/ECACE.2017.7912968.

[8]. L. L. Fernández, M. P. Díaz, R. B. Mejías, F. J. López and J. A. Santos, "Kurento: a media server technology for convergent WWW/mobile real-time multimedia communications supporting WebRTC," *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2013, pp. 1-6, doi: 10.1109/WoWMoM.2013.6583507.

[9]. A. Baćo, M. Mihajlović, M. Savić and M. Ljubojević, "System for Management of Scalelite Based Videoconferencing Systems," 2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH), 2021, pp. 1-6, doi: 10.1109/INFOTEH51037.2021.9400690.

[10]. "Scalelite", *Blindside Networks*, [online] Available: https://github.com/blindsidenetworks/scalelite.

[11]. "How to build a scalable BigBlueButton Video Conference Solution on AWS", [online] Available: https://aws.amazon.com/blogs/opensource/how-to-build-a-scalable-bigbluebutton-video-conference-solution-on-aws/

[12]. Kumar, Brijesh; Katsinis, Constantine, "Design and Deployment of Highly-Scalable Distributed Web Conferencing Systems" Proceedings on the International Conference on Internet Computing (ICOMP); Athens : 1-6. Athens: The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2014.

[13]. Dar, Ab. (2016). "Survey On Scalability In Cloud Environment". International Journal of Advanced Research in Artificial Intelligence. 5. 5.

[14]. D. Bellenger, J. Bertram, A. Budina, A. Koschel, B. Pfander, C. Serowy, ¨ I. Astrova, S. G. Grivas, and M. Schaaf, "Scaling in cloud environments," in Proc. of the 15th WSEAS International Conference on Computers, 2011.

[15]. M. Artac, T. Borovssak, E. Di Nitto, M. Guerriero and D. A. Tamburri, "DevOps: Introducing Infrastructure-as-Code," *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, 2017, pp. 497-498, doi: 10.1109/ICSE-C.2017.162.

[16]. Karim, F., & Goodwin, R. (2013). "Using cloud computing in e-learning systems". *International Journal of Advanced Research in Computer Science & Technology, 1*(1), 65–69.