Abstract—With the advent of Facebook and Twitter, people are demanding high-performance high-quality online multimedia systems where a large number of users can collaborate with each other. Incorporating these features usually makes the system more bandwidth consuming, less spontaneous; latency is often high, and often, only low quality content can be used. In this paper we present the ACE architecture which is implemented in a virtual collaborative environment and overcomes the mentioned drawbacks.

Keywords- ACE, A-VIEW, e-learning; collaborative object; collaborative architecture; architecture for collaborative environment

I. INTRODUCTION

In online networks such as Facebook and Twitter, social collaboration [1] is reaching new heights. People get to know and learn from each other. In formal education, collaboration is starting to play a major role. Chen et al. [2] have presented a collaborative education model and architecture to support collaborative e-learning. Their model emphasizes that education has been shifting from “teacher” centered to “student” centered, and is now shifting to “environment” centered, where teachers, students and parents can interact, collaborate and cooperate with each other.

In a typical distance education interactive live class, the teacher is the controller of the system, and the students are the recipients. Figure 1 shows how the A-VIEW system [3,4] uses multiple displays at the student node of a live class. The four quadrant approach consists of the teacher video, the document, the whiteboard, and the interaction window.

The A-VIEW system (Amrita Virtual Interactive E-Learning World) has been implemented and is being used live at several campuses at Amrita. This system represents a typical live e-learning environment and has the added advantage of configurable multiple displays. In this kind of an educational system, several services such as teacher video, document, whiteboard, chat, desktop sharing, internet browser, animation, interaction window are present. These services can be implemented as a part of a service oriented architecture (SOA) [5], which provides flexibility and allows developers to work in parallel.

Finally, these collaborative services need to be integrated into a common framework with a natural and intuitive user interface. When several services and large numbers of users are integrated in collaborative environments, one major issue is that there needs to be a strategy for managing the bandwidth load on the server and the client nodes. A large number of live services and users increases load on the servers, thereby degrading system performance. Furthermore, as the quality and the quantity of the multimedia materials increases substantially, and the users are allowed to connect to the live session after it has been in progress for some time, more issues arise. These mainly pertain to the amount of bandwidth required during the live sessions, and the initialization of each collaborating user with the multimedia material [6].

In this paper, we present a system architecture, ACE (Architecture for Collaborative Environments), which
addresses some of these issues. We present an approach of multiple clusters of servers; which simultaneously work together to have a steady live online system with large numbers of users. In addition, other server clusters provide an initialization cache for the new users that are joining. We show how this approach maintains a steady state and how the addition of new large number of users to this system does not perturb the steady state. In addition, we show that there are several types of collaborative multimedia objects, and describe how these collaborative objects differ in terms of their states and control strategies used during the live collaborative sessions.

II. ARCHITECTURE FOR COLLABORATIVE ENVIRONMENTS (ACE)

To understand the dynamics of a live collaboration environment, let us consider an e-learning system with client server architecture. Let us assume that this system is designed for 100 users and each user consumes roughly 100 kbps. This would imply that the server needs to have a bandwidth of around 10 Mbps. Further, let us assume that the multimedia content that is being actively shared by all the users is loaded locally at each client node from the server and has a size of 10 MB. Assume that this content of 10 MB has been loaded by all the users before the live session starts. Suppose fifty users are initially connected to the system. Since each user requires 100 Kbps for live collaboration, the net bandwidth usage will be 5 Mbps after all initialization processes have been completed. Now, let us assume that another fifty users are trying to connect to the system after the live session is in progress. Now, these users require bandwidth to download the initial content of 10 MB, and also they are connected to the live system. Hence, the available bandwidth of 5 Mbps is not enough to support the new users. This perturbs the steady-state system performance of the existing system and does not offer adequate performance for the new 50 users.

Note that the above example assumes that the entire bandwidth content of 10 MB is loaded locally for each client node. In this way, the control strategy for collaborative communication is more effective. For example, consider a case where all the nodes have been preloaded with the same large document. When one user moves to page 20 of a large document, the control strategy only needs to send a message with page number 20 to the other nodes. In another case, when one user moves in virtual reality, only the incremental information about the movement in that environment needs to be communicated to the other nodes. This reduces the amount of information that needs to be transmitted in real-time, thereby reducing the load on the system, and also allows for much higher quality of content that can be maintained at the local nodes. For example, in such a system some users can have very high quality animation graphics of huge sizes; and some users can have medium and/or low size graphics; depending on their ability to cache the multimedia content. Thus the system provides for a dynamic range of multimedia content depending upon the performance characteristics of the individual collaborating user.

So, this is one of the primary characteristics of ACE. Each user is assumed to have the multimedia content loaded at the client node before the user is fully connected to the active system. In addition, as shown in Figure 2, ACE primarily consists of two server clusters, namely active and passive servers, which communicate with each other. Active servers continuously update the passive servers with the collaborative object information that is changing. A collaborative object is a standard object-oriented wrapper with a set of messages and actions. The system has a distributed architecture with local intelligence and uses message passing technology for communicating with other objects in the virtual collaborative environment.

ACE is implemented based on a distributed client-server architecture where users collaborate using control messages. The architecture, as shown in Figure 2, consists of an active server cluster and a passive server cluster. The active cluster consists of live media servers and content servers, while the passive cluster consists of database servers and content servers. As the content in the active servers gets updated by the live users, it is incrementally transmitted to the passive servers. Thus, the passive servers maintain an updated copy of the multimedia content.

The active cluster serves the live users and the passive cluster initializes the newly logged in users. ACE also determines the most suitable server for each user, so that the collaboration between the users remains smooth and steady throughout the session. For this, ACE considers two factors, server load and server’s proximity with the client node, to determine which server should be allocated for the user.

In any architecture, time taken to download the multimedia content is another critical factor which affects the performance of the system. The transfer time from the server to the client node depends upon the size of the content, the number of users downloading simultaneously, and the bandwidth of the server and the client nodes.

Several experiments were conducted for measuring transfer time and the load on the system and the client nodes. The results of one experiment are shown in Figure 3.
For users connecting to the passive server, it is found that as the size of the multimedia content is increased, the time taken to load the content for one user increases linearly, assuming that the other parameters remain constant. In general, the approach is to load the multimedia content from the passive server to the users who are logging in.

In ACE the active server continuously updates the passive server with all the events related to collaborative objects, such as document sharing, whiteboard, and chat. The active cluster records all the collaborative object events in the content server. These recorded events are used to incrementally update the database and content server of the passive cluster. When a new user logs in, the local node will be updated by the passive cluster for faster access.

After successful initialization, the user gets switched from the passive server to the active server. During this transition, new events may have occurred at the active server. To prevent any loss of state information, the user’s information has a timestamp associated with it. When the user is switched to active server, the user’s timestamp is compared with the recorded timestamps in the active server, and for any new events that have occurred during the transition, the user content will be updated and synchronized to the latest information. One limitation here is that there might be small delays to synchronize the users with the live sessions, due to the updates that need to be done when transitioning from the passive to active clusters.

III. CLASSIFICATION OF COLLABORATIVE OBJECTS
There can be different types of collaborative objects. Collaborative objects are classified as synchronous and asynchronous.

a. Synchronous
Synchronous collaborative objects are used to achieve real-time interaction among users, and are widely used in an e-learning world. Synchronous objects are further classified into static, dynamic and active based on their behaviour.

i. Static objects:
The contents of a static object cannot be modified. The participants can share the static object data with others but cannot modify it. For example while sharing a document (see Figure 4a), the presenter can navigate through the various pages of the document and it gets reflected at the viewer’s end. But the presenter is not allowed to change the content of the document during sharing. The 3D engine (see Figure 4b) is another example where users can collaboratively rotate and view engine parts but cannot modify them. Virtual Chemistry lab (see Figure 4c) is yet another example where students and teachers can collaboratively do their experiments in a virtual world [8].
b. Dynamic objects:

Unlike static objects, dynamic objects can be modified during collaboration. Document Annotation (see Figure 5a) is an example where each user can dynamically modify the document object. White board (see Figure 5b) is another example where users can make any kind of drawings on a canvas.

c. Active objects

A dynamic object becomes an active object if it can make decisions and modify the behaviour of the system. The Chess application (see Figure 6a), where the application imposes time restrictions on the users to make a move, serves as an example for an active object. Another example includes the virtual reality toolkit (see Figure 6b), a gaming application [7,8]. In this case, the system decides whether the users should play or walk through the environment, depending on the time. Other examples could include games that change the weather conditions randomly.

b. Asynchronous

Asynchronous objects are those in which interaction among users are not in real-time. The Playback discussion forum (see Fig. 7a and 7b) is an example where users can post their doubts in audio/video formats. Anyone can post their comments on the original question, like a discussion forum.

IV. Modes of collaboration

In a collaborative system different users are assigned different privileges based on the roles they play during the collaborative session. The different roles include “teacher,” “student,” and “researcher.” Based on these roles, different modes of collaboration may be present. For a typical e-
learning system, these modes of collaboration can be classified as follows:

a. **Teacher-Students**

In this mode teacher leads and the student follows. The teacher has maximum access to all the resources, such as collaboration tools and learning materials, but students have limited access to these. For example, during a live session, the teacher can upload and control a document and students can only view whatever teacher is presenting. In short, the students won’t have rights to control the document.

b. **Many Teachers and Students**

In this mode, more than one teacher can interact with the students. As in the previous mode, teachers have maximum access to all resources and students have limited access to the resources. There will be a moderator, either a teacher or a student, and they can pass control to any of the attendees.

c. **Student-Students**

In this mode, a student can collaborate with other students. However, they will have limited access to the application resources. There will be one moderator for the session. In Johnson & Johnson’s words “the vast majority of the research comparing student–student interaction patterns indicates that students learn more effectively when they work cooperatively” [9].

d. **Teachers/Researchers:**

In this mode a teacher or researcher can collaborate with other teachers or researchers. They will have maximum access to all the resources. There will be one moderator, a teacher or a researcher, for the session.

V. **CONCLUSION**

ACE is an architecture with a distributed infrastructure which is best suited for environments with a large number of collaborating users and a rich multimedia content. This approach shows how multiple clusters of servers are used to achieve optimal performance. The active servers simultaneously work with the passive servers to have a steady live online system with a large numbers of users. The passive server clusters provide an initialization cache for the new users that are joining. This approach maintains a steady state and the addition of new large number of users does not disturb it. This architecture has been tested and found highly effective for a virtual collaborative e-learning environment, which can share several types of collaborative objects, and which can be in any of the four modes of collaboration.

**ACKNOWLEDGEMENTS**

The research and development for A-VIEW has been funded by the National Mission on Education (MHRD), TIFAC, ISRO, and Amrita University. We are thankful to IIT Bombay for their partnership and feedback. In particular we would like to thank Mr. N.K. Sinha (MHRD) and Dr. Kannan Moudgalya (IIT Bombay).

We are extremely grateful to the members of the Amrita E-Learning Research Lab whose hard work and dedication has made this work possible.

**REFERENCES**


[8] Carl Dreher1, Torsten Reiners1, 2, Naomi Dreher1, and Heinz Dreher1, Member, IEEE, “3D Virtual Worlds as Collaborative Communities Enriching Human Endeavours: Innovative Applications in e-Learning”, 3rd IEEE International Conference on Digital Ecosystems and Technologies, 2009