# Issues for the Performance Monitoring on Networked Virtual Environments: the case of EVE

C. Bouras, E. Giannaka, Th. Tsiatsos
Research Academic Computer Technology Institute
Computer Engineering and Informatics Dept., Univ. of Patras
Patra, Greece
bouras@cti.gr, giannaka@cti.gr, tsiatsos@cti.gr

***Abstract*—The wide expansion and increasing demand for Networked Virtual Environments resulted in efforts for the optimization of these environments with the enhancement of advanced features, which offered extended functionality and created an improved sense of realism. However, the added functionality often creates resource overhead both in the system as well as in the network host, which results in performance reduction. This paper presents such a networked virtual environment and based on the system's architecture tries to trace down with the time and recourse consuming parameters and proposes models on which a networked virtual environment, like the one presented, could achieve optimization of its performance.**

***Keywords-networked virtual environment, performance monitoring, performance optimization, consistency***

## I. Introduction

The inherent need for communication, collaboration, resulted in the evolvement of the primitive services originally offered (e.g. e-mail) to advanced functionalities, which offer a high sense of realism to the user, forming a reality, the so-called virtual reality. Many platforms which adopted the virtual reality technology were implemented and networked virtual environments started to be enhanced with additional functionality, as text and audio chat, streaming media support, application sharing, etc. However, these enhanced features are demanding both in terms of system as well as in terms of network resources. Therefore, the big challenge is perceived to be the optimization of the network performance and the reliability of the networked environments.

In this paper we describe such a networked virtual environment, called EVE [1], which is intended to be used for the support of multi-user virtual environments and the provision of advanced services that can efficiently support distance learning and educational purposes. In addition, we focus on the aspects of the system that should be taken into consideration in order the virtual environment to be efficient and reliable, as well as the parameters that should be monitored and simulated in order to identify the time consuming and demanding components of this platform. By identifying these parameters, it becomes more feasible to define and determine ways or architectural models, under which the platform could present an optimized networked performance. The performance monitoring and evaluation of such a networked virtual environment can be performed using tools called, networked simulators.

The remainder of this paper is organized as follows. In section II we present the architecture of EVE in terms of how the platform is implemented. Section III is dedicated on how performance monitoring of EVE could be realized, on which aspects of the platform we should focus, as well as on the parameters that we should monitor. Section IV proposes a model that could be used for the extension of our networked virtual environment and describes how this architecture could contribute to the optimization of the system's performance. Finally, section V summarizes and concludes the paper.

## II. EVE Architecture

EVE's architecture is based on a client-multi server platform model. The current form of EVE constitutes a simple structure, which allows and supports the basic functionality that the platform is intended to offer [1], [2], [4].

Regarding the Server Side, the servers on which the platform relies, is the message server and two application servers. The Message server is responsible for the manipulation of the virtual worlds that constitute the training area of the platform and supports the illusion to the participants that they share a common space by updating the view of the world every time that a shared object is modified. Three servers, each of which is used for a specific sequence of operations, constitute this message server: the connection server, the init server and the VRML server. The Application servers are responsible for providing specific functionality to the participants of the virtual world. In the current form of EVE there are two application servers available, a chat server, which is responsible for the text chat support and allows group chat and one-to-one communication between two users, and an audio server that is responsible for the audio communication between the users of the platform.

Regarding the Client Side, as depicted in Fig. 1, in order the users' client to communicate with EVE's servers and have access to the provided functionalities they need a web browser for the communication with the web server of the system, a VRML browser, a VRML client to allow the navigation of the user's avatar in the virtual training space, a main client that is responsible for the primary connection of the user to the Message Server, a chat client which is responsible for the text chat communication between the users of the same virtual space and an audio client that allows the audio communication between participants.

For the transmission of the packets and the achievement of the communication of the connected clients with the host servers there are three types of communication supported in EVE, each of which is found to be optimum for a certain kind of messages [11]. We categorized the messages exchanged in the EVE communication platform in the following categories: a) the messages related with the initialization of the virtual world and the initial connection of a client to a server as well as the messages exchanged between the servers of the platform, b) the position messages that are related with the avatars' position in the virtual environment and finally c) the important messages, which correspond to messages that are vital for the consistency of the networked virtual environment. For simplicity reasons, we consider as important messages all messages except for the position messages. The connection types supported by EVE are the following: A) TCP Communication that is characterized by reliability in the transmission of information packets. For the EVE platform, this type of communication is selected for the server-to-server communication as well as for the initial connection of a client to the message server, where the current status of the world is transmitted. B) Simple UDP Communication that is characterized by the high speed in the transmission of the information packets. However, it cannot assure the reliable and correct delivery of the data packets. Therefore, this type of communication is selected for the transmission of messages that their possible failure in delivery does not imply a severe impact on the consistency of the virtual world. Such messages are the position messages. C) UDP with one-way acknowledgement, which is characterized by reliability, safety and a timely delivery. Therefore, a UDP connection with one-way acknowledgement was implemented for the support of the previously described important messages. The loss of these messages could have catastrophic results for the communication and interaction in the framework of the networked virtual environment. The architecture of EVE is presented in Fig.1.

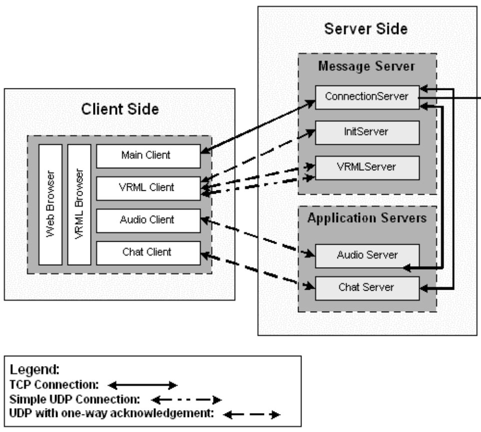


Figure 1. EVE's architecture

## III. PERFORMANCE MONITORING ON EVE

Up to this stage of development of the EVE platform, we mainly focused on the implementation and integration of advanced services. However, even though the functionality is successfully enhanced in each virtual, what matters most is the efficiency and the reliability that the platform presents under the network pressure and the system limitations. For this reason, we need to monitor the networking performance of the platform, to identify the boundaries in which the platform operates efficiently and trace down the factors that they can generate a performance downgrade in order to be able to deal with them.

### *A. Constraints to performance*

The performance of EVE is affected by two main factors: the characteristics of the network that hosts each client-server connection and the architectural model of the system. In particular, and regarding the network host, the main constraints to the platform's performance [8] is the bandwidth, which can cause bottlenecks, collisions along with loss of information and the delays that cause problems in the distribution and synchronization of the interactions between the participants and the platform.

As far as it concerns the constraints that affect the platform's performance from the system's point of view, the load is focused on the following parameters [10]: a) the inefficient management of the platform's resources which causes overload and processing delays, b) The inability of the system to react and handle efficiently the case of a possible failure and c) the overload, which arises when the number of the participants exceeds the platform's potentialities.

On the one hand, the network characteristics are unpredictable and thus, it is difficult to promise that all the client-server connections will be handled equally. On the other hand, the architectural model on which EVE is based, could be extended to provide high quality of service, reliability and high network performance in general.

### *B. Aspects of the EVE architecture*

What is essential and vivid for a networked virtual environment is the maintenance of consistency on what the participating clients view and what has really happened in the virtual worlds. This consistency depends on a number of factors and how efficiently these factors are managed by the system's architectural model [12].

**Resource Utilization:** This factor corresponds to the way that the networked virtual environment is designed to distribute and dispense its available resources. There are cases, where the system favors certain procedures over others, having as a result the downgrade of some information or operations that are more valuable and vital to the end user.

**Consistency of the virtual environments:** This aspect is related to the manipulation, the update and the timely delivery of the shared events that are modified in the Virtual Environment in order to maintain the consistency for all connected clients.

**Failure Management:** This factor corresponds to the mechanisms that the platform uses to handle failures that might come up.

**Scalability:** This factor is related to the number of the clients that can be hosted efficiently by the EVE platform, in a way that preservers all the primitives of a networked virtual environment.

### *C. Parameters to be measured*

The parameters that should be monitored and estimated, in order to have accurate results about the networked performance of the EVE platform, arise from the constraints and the factors that were described in the previous subsections. More precisely, we can answer how efficiently and reliably are EVE's functions over the network by evaluating the parameters and variables that are vital for the system [9]. These parameters are the following: a) the number of the requests that are launched by the host servers that could be either the message server or one of the application servers, b) the number of the messages that are disallowed due to system failures or network congestions, c) the number of the replies that return to the host servers from the connected clients, d) the time it takes to the message server or to the application servers to build a message before transmitting it to the corresponding clients, e) the time it takes a request to return as a reply from the client, f) the time it takes to a host server to process the requests that are received by the connected clients, g) the time it takes to the host servers to transfer a message on the network, h) the time that a message spends on the network, i) the time it takes to the message server and the application servers to exchange messages, j) the average bandwidth that each message and packet type requires, k) the memory usage of the host servers during the processing of the different types of messages and packets and l) the CPU usage of the host servers for the processing of the connected clients' requests.

Table I presents the parameters that will be measured and the aspects of the system that each of these parameters affects.

## IV. OPTIMIZING PERFORMANCE

As described, in its current form EVE is based on a simple architectural model that can support efficiently only a limited number of parallel virtual worlds and simultaneous users. However, during the primer implementation of the EVE platform, we took into account the possible need for extension due to increased demand and participation. The new architectural model uses the same rational as the original model, which means that both the processing load as well as the communication load should be distributed regarding with the connections and requests [4].

### *A. Extended Architectural Model*

The new model, which will be used for the extension of the EVE platform, is still based on two types of servers, the message server and the application servers. The model on which the extended architecture for the optimization of the platform is based, can be seen in Fig. 2.

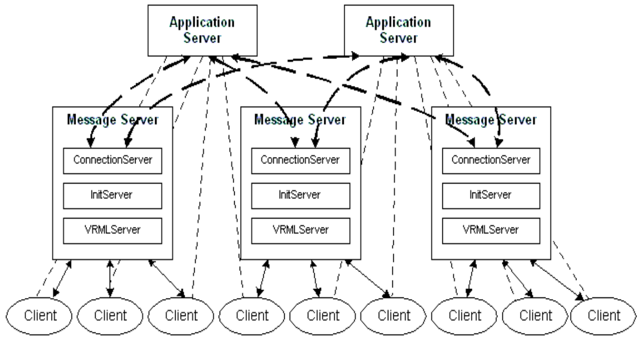


Figure 2. Expanded architectural model

The number of the message servers can be increased regarding the client connections and the processing load. Each message server shelters a certain number of virtual worlds. This number will result from the performance monitoring that we will conduct on EVE.

However, the main concept and innovation of this architectural model is the fact that each message server is a back up server for a number of the rest message servers. This implies that if a failure happens, the clients supported and hosted by the "damaged" message server can be distributed to the other servers of the system. In addition, each message server can be a client to another message server, when the processing load of the first one exceeds its hosting capabilities.

Regarding the application servers, their communication with the message servers and the clients is similar to the simple model, in which EVE is currently based, with one basic difference. Every time, that a clients sends a request to one of the application servers, the application server communicates with the ConnectionServer of all available message servers, in order to certify the identity of the client which has posted the request. After this verification has been completed, the application server communicates directly with the interested clients.

Finally, regarding the network communication, this is still based on the three types of communication, presented above: the TCP communication, the simple UDP communication and the UDP communication with one-way acknowledgement.

### *B. Optimizing the aspects of the system*

In this subsection we will describe why we proposed the specific architectural model for the expansion of our networked virtual environment and in particular, how we expect to optimize the aspects of the system, presented in a previous section. Firstly, this approach improves the resource utilization and management since the processing load has been divided in multiple host servers. Thus, each server is concentrated in supporting a number of clients that it can handle efficiently. Regarding the sharing of events, this is realized faster, since each message server that is responsible for the sharing, updates and distributes the shared events of a certain number of virtual environments, which it hosts. In addition, the failures are handled more effectively since, at any time, the tasks of the "damaged" message server can be carried out by another back up server. Finally, this architectural model makes the platform more scalable by diving the processing load and allowing the connection of additional servers to the system, in a transparent way for the end user. At this point it should be mentioned that this approach increases the messages exchanged in the networked virtual environment, since some additional messages for the communication and synchronization of the available servers are needed. However, in most cases the overhead created from these messages does not have a serious impact in the overall system's performance.

TABLE I. PARAMETERS TO BE MEASURED

| Parameters \ Aspects of the system | Resource Utilization | Consistency | Failure Management | Scalability |
|---|---|---|---|---|
| **Number of requests from the host servers** | √ | | | √ |
| **Number of disallowed messages** | | | √ | |
| **Number of replies from the clients** | √ | | | √ |
| **Time for building a message** | | √ | | |
| **Time for a request to be delivered to the server from the client** | | √ | | |
| **Processing time on the server** | | √ | | |
| **Transmission Time** | | √ | | |
| **Time on the network** | | √ | | |
| **Time for messages between the servers** | | √ | | √ |
| **Average bandwidth for each type of message** | √ | √ | | |
| **Memory** | √ | | | |
| **CPU** | √ | | | |

## V. CONCLUSIONS AND FUTURE WORK

This paper examined the performance monitoring of networked virtual environments, and more precisely of EVE, a small-scale virtual environment designed for educational purposes. The effectiveness of EVE could be qualified be monitoring and simulating the systems' performance under multiple testing scenarios. However, the question that comes up is what could actually be measured in order to have accurate results about the platform's efficiency and reliability? Therefore, we tried to make an approach on the precise parameters that we could measure and monitor, in order to identify the boundaries within which the platform could effectively provide the anticipated functionality. Finally, based on the fact, that the current form of EVE can support a small number of clients, due to its limited host resources, we followed a primer approach on how the architectural model could be expanded in order for the platform to be characterized of scalability and optimized performance.

Consequently, our next step is to conduct the testing scenarios and experiments on EVE platform with the use of ns-2 network simulator that best meets our input needs. After extracting the necessary information about the factors and parameters that downgrade the system's performance we intend to expand it, using as basis the optimized architectural model described. Our target is the implementation and completion of an advanced networked educational platform, which will efficiently meet the theoretical concepts of EVE.

## REFERENCES


[1] EVE (Educational Virtual Environments) prototype, http://ouranos.ceid.upatras.gr/vr/.

[2] Bouras C., Giannaka E., Tsiatsos Th., "Virtual Collaboration Spaces: The EVE Community". The 2003 Symposium on Applications & the Internet- SAINT-2003, Orlando, Florida, January 27-31 2003, pp. 48-55

[3] Bouras C., Tsiatsos Th., "Building Educational Virtual Environments". IEEE International Conference on Advanced Learning Technologies (ICALT-2002), Kazan, Russia, September 9-12 2002, pp.547-548 (short paper)

[4] Bouras C., Tsiatsos Th., "Architectures and Protocols for Educational Virtual Environments". C. Bouras, T. Tsiatsos, IEEE International Conference on Multimedia and Expo (ICME2001),Tokyo, Japan, August 22-25 2001

[5] Bouras C., Psaltoulis D., Psaroudis C., Tsiatsos Th., "A Platform For Sharing Educational Virtual Environments". C. Bouras, D. Psaltoulis, Ch. Psaroudis, T. Tsiatsos, 9th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2001) Split, Dubrovnik (Croatia) Ancona, Bari (Italy), 9-12 October 2001, Vol. II, pp. 659-666

[6] Bouras C., Psaltoulis D., Psaroudis C., Tsiatsos Th., "An Educational Community Using Collaborative Virtual Environments". C. Bouras, D. Psaltoulis, C. Psaroudis, T. Tsiatsos, 1st International Conference on Web-based Learning, Hong Kong, China, August 17 – 19 2002, pp. 180-191

[7] Zyda M., "Networking Large-Scale Virtual Environments". Proceedings of Computer Animation '96, 3-4 June 1996, Geneva, Switzerland, IEEE Computer Society Press, pp. 1-4

[8] Faisstauner C., Schmalstieg D., Purgathofer W., "Scheduling For Very Large Virtual Environments and Networked Games Using Visibility and Priorities". In Proceedings of the 4th International Workshop on Distributed Interactive Simulation and Real-Time Applications (DIS-RT 2000), pages 31-38, San Francisco, CA, August 2000.

[9] Deligiannidis L., Couch A., Jacob R., "Performance Characterization of Distributed Virtual-Reality Applications: A Case Study". Proc. of the 3rd International Conference on Internet Computing (IC2002), CSREA Press and IEEE Internet Computing Magazine, June 2002

[10] De Oliveira C., Shirmohammadi S., Georganas N., "Collaborative Virtual Environments Standards: A Performance Evaluation". IEEE DiS-RT'99, Greenbelt, MD, October 1999

[11] Bouras Ch., Psaltoulis D., Psaroudis Ch., Tsiatsos Th., "Multi-User Layer in the EVE Distributed Virtual Reality Platform". The 5th International Workshop on Multimedia Network Systems and Applications (MNSA'2003) (to appear)

[12] Singhal S., Zyda M., "Networked Virtual Environments: Design and Implementation", ISBN 0-201-32557, ACM Press, 1999