On the characterization of avatars in Distributed Virtual Worlds

Pedro Morillo[φ], Marcos Fernández[φ] and Juan Manuel Orduña[♣]

φ Robotics Institute, University of Valencia (Spain)
♣ Department of Computer Science, University of Valencia (Spain)
{Pedro.Morillo@uv.es}

Abstract
Current Distributed Virtual Environment (DVE) systems run simulations based on a server-network architecture, where the population of avatars should be properly assigned to the servers in the DVE. This goal, called partitioning problem, is a hot research problem in the field of networked 3D real time graphics. Some approaches have been proposed for resolving this problem, all of them based on a very basic model which describes the behaviour of an avatar in a virtual world. This model estimates the workload that a given avatar adds to its server as an independent and static value. In order to design scalable and accurate partitioning schemes we propose a new characterization of the workload generated by the avatars in the system. In this model, the workload is computed as a function of two factors: avatar movement rate and factor of presence. Evaluation results show that this proposed model correlates with real systems properly, providing an accurate abstraction of avatars for load balancing purposes.

Keywords:
Virtual Reality, Distributed Simulations, DVE, Avatars' Workload, Multi-Server Architectures.

1. Introduction

Distributed Virtual Environments (DVE) are systems where many users connected from different computers share the same 3D virtual scene [16][4]. Nowadays they are used in a wide variety of applications [16], such as collaborative design [15], civil and military [18] distributed training simulations, distributed and interactive e-learning [13] and multiplayer games [9, 1].

In these systems, each client is represented by an entity (usually a humanoid) called avatar. The avatar state is controlled by the user's input. Since these systems support visual interactions between multiple users in the shared 3D virtual environment, every change in the simulation has to be propagated to the rest of participants. Thus, the scalability of the system becomes a key issue as the number of participants in the DVE grows [4]. A good scalability must lead the DVE system to offer a consistent view of the scene without compromising the interactive performance of the clients.

As it is described in [16,17], the main methodology for enhancing scalability in a DVE system is a multiple-server architecture. In these architectures, also called server-network or hierarchical, simulation control is organized by a set of interconnected servers. All the multiplatform clients of the DVE are assigned to one of these servers. When an avatar changes the state or position the associated client sends one updating message to a server, and then the server propagates this message to other servers and clients.

In order to avoid a message outburst when the number of clients increases, areas of influence (AOI) are specified for each one of the avatars. In this way, messages are propagated from one avatar to those avatars fall into its AOI. Depending on the destination server, two kinds of messages are defined in a DVE scheme (see figure 1): intra-server messages when both the sender and receiver are allocated in the same server, and inter-server messages when they are not. In this framework a client only establishes a connection to a server of the DVE system. Through this connection it sends changes in status of the avatar and receives only the information of the avatars which are included into its AOI. Therefore, in order to maintain a consistent state in a large DVE system, avatars must complete very little workload, storage and exchange of information.

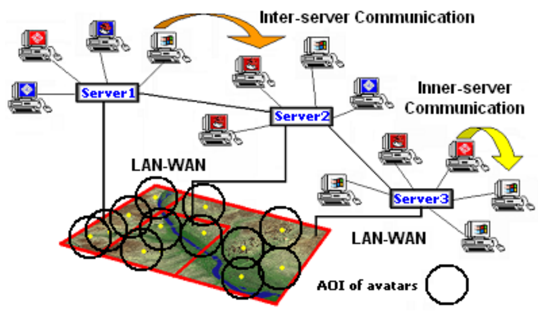

Figure 1: *An example of a basic DVE system where 3 servers are managing 13 avatars.*

Since avatars produce the workload of the system, the server-network approach distributes the overall workload among the servers in order to improve the scalability of the DVE. When an actual and efficient DVE system is designed the partitioning problem must be solved. This problem consists in balancing the overall workload among the servers [10] in the system while minimizing the amount of inter-server messages. The most important approaches to this problem [3,12,14] are based on a very simplified model that describes the behaviour of the avatars which are simulated in the DVE. In these approaches, the workload generated by avatars to the server where they are assigned is modelled as a fixed value. During the simulation time, the avatars maintain this permanent value independently of any inherent factor.

Although we can model every kind of DVE system by adapting the parameters of the presented framework simulation, we have observed and monitorized some relevant behaviours of avatars. In our simulations where *n* avatars must be allocated in *S* servers, it has been very common to obtain different performance results from the same partition of the world. So in consecutives simulations, when a set of clients are assigned to the same server different latencies and CPU utilizations can be measured for the same configuration of the DVE system. These results do not agree with behaviour of avatars referred in the literature. During the simulation time, this behaviour models the workload generated by an avatar as a static value. Therefore, in order to represent simulations accurately, a new characterization of the avatars within a DVE system is needed.

In this paper, we present a new model for describing the workload of avatars that are running in a DVE system. Experimental results show that this workload is represented by a variable function. For each avatar in each simulation step, this non-static value is related to its rate movement and the amount of clients which are surrounding it. These empirical results provide a more accurate model of avatars that can help in turn to develop more accurate partitioning strategies. We plan to use these results to improve DVE throughput (defined as the number of users supported by the system [16]) efficiency and scalability of our DVE system. This system is a collaborative driving simulator [5].

The rest of the paper is organized as follow: Section 2 describes how the partitioning problem has been traditionally handled in DVE systems. Section 3 details the proposed avatar characterization that allows to study the behaviour of DVE systems experimentally. Next, Section 4 presents some correlation and evaluation results. Finally, Section 5 presents some concluding remarks and future work to be done.

2. Background

Architectures based on networked servers are becoming a de-facto standard for DVE systems [16,17]. In these architectures, the control of the simulation relies on several interconnected servers. Multi-platform client computers must be attached to one of these servers. When a client modifies an avatar, it also sends an updating message to its server, which in turn must propagate this message to other servers and clients. Servers must render different 3D models, perform positional updates of avatars and transfer control information among different clients. Thus, each new avatar represents an increase in both the computational requirements of the application and also in the amount of network traffic. When the number of connected clients increases, the number of updating messages must be limited in order to avoid a message outburst. In this sense, concepts such as areas of influence [11,18], locales [2] or auras [8] have been proposed for limiting the number of neighbouring avatars that a given avatar must communicate with.

In order to design a scalable DVE system, Lui and Chan [11] detail a mechanism which describes the behaviour of a multi-server system. Through a quality function, this mechanism estimates the efficiency of a DVE considering two factors. The first one measures how the computing workload of the system generated by the avatars of the DVE is distributed. Avatars should be proportionally assigned to DVE servers according to the computing resources of each server. The second factor measures the number of inter-server messages generated from the avatars updates.

Taking into account these two concepts the partitioning problem has been addressed from three points of view. First, Pekkola and Robinson [14] define a local strategy where an overloaded server can migrate avatars to neighbouring servers in order to decrease its workload. Two servers are neighbours if it is possible to find avatars allocated to them in such a way that the intersection of their AOI is not empty. Second, Rynson [3] extends this local concept of balancing. He proposes an algorithm which allows to migrate avatars from overloaded servers to the most underloaded of the neighbouring servers. Finally, since it is very difficult to obtain optimal features if all the servers of the DVE are not involved in the load balancing, [12] and [11] describe global schemes of allocation. Based on modern heuristic or custom programming, respectively, these approaches perform an algorithm which obtains the target server for every avatar of the system.

Although all these approaches try to maximize throughput and minimize storage and communication requirements for maintaining a consistent DVE, they are based on a simplified avatar model. All of them characterize the workload generated by an avatar, when it is allocated in a server, as a fixed value. In their simulations the workload generated by each avatar does not change and it is independent of any parameter of the DVE system.

3. A new avatar model

In consecutives simulations, when a set of clients are attached to the same server different latencies and CPU utilizations can be obtained for the same configuration of the DVE system. These results did not agree with the behaviour of avatars referred in the literature where a static value models the workload of an avatar during the simulation. Therefore, in order to represent accurately simulations, a new characterization of the avatars a DVE system is presented.

Additionally, the spatial distribution of avatars in the virtual world determines the workload that each avatar adds to the server where the avatar is assigned. In fact, it is noticeable that the impact of assigning some avatars to a server in the system grows, in terms of latencies and CPU utilization, as their Euclidean distance in the virtual world decreases. As we have described in Section 1, it is explained because when an avatar moves around the scene it sends an update message to the server where it is assigned. As every server knows the position of each member of the simulation the server has to propagate these updating data to all the avatars contained in the AOI of the avatar, which is the sender of the message. Since nowadays multicast is not fully available on the Internet and most of DVE systems are running on this network (usually as web-based applications [1,3,9,14,15,16]) when a server receives an update message it must send unicast messages to the avatars included in the AOI of the avatar that has moved.

In order to take into account this fact we have defined a new parameter called *presence factor* (fp). Fp of an avatar A is the number of avatars in whose AOI this avatar A appears.

We propose the characterization of avatars in generic DVE systems by simulation. The evaluation methodology used is based on the main standards for modelling collaborative virtual environments, that is FIPA [6] and HLA [7]. We have developed a simulation tool that models the behaviour of a generic DVE system with a network-server architecture. Specifically, we have implemented a set of multi-threaded servers. Each thread in a server uses blocking sockets for communicating with a client. Each client simulates the behaviour of a single avatar, and it is also implemented as a multi-threaded application. One of the threads of the client manages the communication with the server it is assigned to, and another thread manages user information (current position, network latency, etc.).

Our simulator model is composed of a set of *S* interconnected servers and *n* avatars. Following the approach specified in FIPA and HLA standards, one of the servers acts as the main server (called Agent Name Service [6], Federation Manager [7] or Loading Collector [3]) and manages the whole system. The main server also maintains a partitioning table where each new avatar is associated to a server of the DVE system. In this way, once the network address and the port number (where the main server listens) are known, any avatar can join the simulation through this main server. This avatar will be assigned to one of the servers and the new relation (avatar to server) will be added to partitioning table. At this point, the new avatar must connect with the assigned server in order to start the simulation.

In each simulation, all avatars sharing the same AOI must communicate between them for notifying both their position in the 3D virtual world and any change in the state of the elements in that AOI. The message structure used for notifying avatar movements is the Avatar Data Unit (ADU) specified by DIS [7].

A simulation consists of each avatar performing 100 movements, at a rate of one movement every 2 seconds. Each time an avatar performs a movement, it notifies that movement to the server it is attached to by sending a message with a timestamp. That server must then notify that movement to all the avatars in the same AOI of the sender avatar. When that notification arrives to the avatars, they return an ACK message to the server, which in turn propagates that ACK messages to the sender avatar. When an ACK message arrives, the sender avatar computes the round-trip delay for communicating with each neighbour avatar. We have denoted this round-trip delay (measured in real-time) as the system response. When a simulation ends, each avatar has computed the average system response for the avatars in its AOI. At this point, all avatars send the average system responses to their respective servers, and then the servers compute the average system response for each server. Finally, the main server computes the average system response (ASR) for that simulation.

An actual and scalable DVE system must keep ASR as low as possible as the number of avatars in the system increases. However if the *rate of movement* of these avatars becomes faster, then it is more difficult to support low levels of ASR than when they are moving slowly. There is the same problem when avatars avoid even distributions within the scene. The proposed model takes into account both problems and therefore it defines avatars workload as a function of both parameters.

Also, we have measured CPU utilization in all the servers of the system. Both *ASR* and also *CPU utilization* show the impact of each avatar in the server where it is assigned.

Finally, following different distributions of avatars in the virtual world proposed in the literature such as uniform, skewed and clustered distributions [3,11,12,14], several DVE configurations have been simulated.

4. Correlation and Simulation results

In this section we present the correlation results for the avatar model proposed in the previous section. The hardware platform for both clients and servers in the DVE system are PC's with processor Pentium IV at 1.7 GHz, with 256 Mbytes of RAM and with NVidia MX-400 graphic cards. Each server has been implemented in a single PC, while up to 50 clients have been allocated in each PC. We have used a 10 Mbps Ethernet as the interconnection network. This simulation tool has been executed on a Windows 2000 Professional operating system. We have tested almost 90 different DVE systems and we have simulated more than 2000 experiments on a great number of different configurations, ranging from small virtual worlds (composed of 3 servers and 180 avatars) to large virtual worlds (composed of 900 avatars and 6 servers). Since we have obtained very similar results in all of them, due to space limitations we only show the results of performed correlations for large worlds.

For the purpose of avatars characterization, each client only simulates a given rate of movements through the virtual world. During the simulation it is assumed that no changes are produced in any static element of the AOI. Also, the size of the surface that models the AOI of the avatar does not change.

We have observed changes in both ASR and CPU utilization when avatars that are allocated in a server change the movements per second *(movement rate)* in the 3D virtual scene. This change leads a variation in the number of messages to be processed by the server and thus it increases CPU utilization for managing this task. Figure 2-left shows the performance results for a DVE when several configurations are simulated. In this case the DVE system is composed of 180 avatars in 3 servers. This figure shows on the X-axis the values of movement rate accomplished by all the avatars in the simulation. The Y-axis shows ASR (Y-left) and CPU utilization (Y-right) values for the simulations performed with these partitions. Each point in the plot represents the average value of the ASR and CPU utilization obtained after 30 simulations of the same DVE system. The standard deviation for any of the points shown in the plot was not higher than 30 ms. for the ASR and 4% for the CPU utilization in any case.

This figure shows that the workload generated by a given number of avatars which are running in a DVE system correlates with their movement rate.

Figure 2-right shows the behaviour of a DVE system when presence factor of avatars is modified. This figure shows why as a collection of avatars comes closer in the virtual space the workload generated by these avatars increases. This grouping movement causes an increment of the *"fp"* of these avatars, and therefore an increasing in the amount of necessary messages for managing them. These new messages mean more CPU utilization for accomplishing this task.

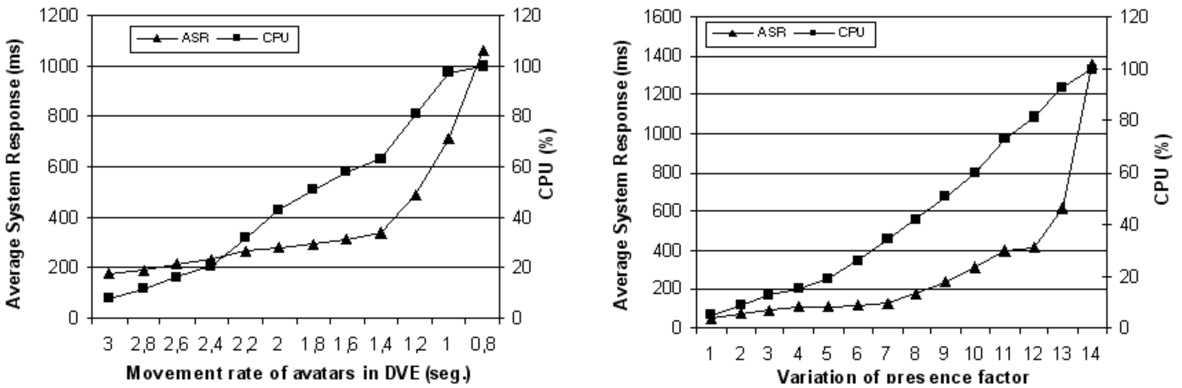

Figure 2: *Correlation of movement rate (left) and presence factor (right) of avatars with the ASR and CPU utilization in a DVE. Performance results had been obtained from a DVE composed of a 180 avatars simulated on 3 equal servers.*

5. Conclusions and future work

The task of efficiently assigning avatars to servers in DVE systems has become a hot research problem. Literature concerning this subject proposes a very basic model which describes the behaviour of avatars in a DVE.

In this paper we have proposed a new model of avatar characterization. This characterization is based in the study of the effects of the rate movement and density of neighbouring avatars (factor of presence). In order to evaluate these effects, average system response and CPU utilization have been elected as performance evaluation parameters. The results show a good correlation of both rate movement and factor of presence with system performance. Therefore, we can conclude that in order to design efficient DVE systems actually, the partitioning mechanisms should take into account the non-static contribution of avatars to the servers they are assigned. Immediately before the partitioning algorithm is executed, the system should compute the new workload of all avatars. The new value should be obtained from the current movement rate and factor of presence.

As future work to be done, we are currently working on the implementation of a new partitioning scheme based on a global load balancing method. This new scheme considers the ideas proposed in this paper had been derived from traditional algorithms used in distributed systems. In the medium term we want to apply this engine of DVE system in the development of a driving simulator. In this system, future drivers will share a 3D virtual town where hundreds of drivers will be connected through a low bandwidth network such as the Internet.

References

1. M. Abrash. “Quake’s game engine: The big picture”. *Dr. Dobb’s. Journal*. Spring 1997.
2. D.B.Anderson, J.W.Barrus, J.H.Howard, “Building multi-user interactive multimedia environments at MERL”. *IEEE Multimedia*, 2(4):77-82, Winter (1995).
3. N.Beatrice, S.Antonio, L.Rynson and L.Frederick, "A Multi-Server Architecture for Distributed Virtual Walkthrough" *Proceedings of ACM VRST*, pp. 163-170, Nov. (2002).
4. E. Churchill, D.Snowdon, Alan J. Munro. *"Collaborative Virtual Environments".* Springer Verlag. (2001).
5. M.Fernández, I.Coma, G.Martín and S.Bayarri, “An Architecture for Optimal Management of the Traffic Simulation Complexity in a Driving Simulator”, *Lecture Notes in Control and Information Sciences, Springer-Verlag,* Vol. 243, (1999).
6. FIPA Agent Management Specification. *Foundation for Intelligent Physical Agents, 2000.* Available at http://www.fipa.org/specs/fipa2003.
7. F.Kuhl, R.Weatherly, J.Dahmann, *“Creating Computer Simulation Systems: An Introduction to the High Level Architecture”*, Prentice-Hall PTR, Upper Saddle River, NJ, (1999).
8. C.Greenhlagh “Awareness-based communication management in MASSIVE systems” *Distributed Systems Engineering*, vol 5, (1998).
9. M. Lewis and J. Jacboson. “Game Engines in Scientific Research”. *Communications of the ACM*. Vol 45. Nº1. January (2002).
10. J.Lui, M.Chan, K.Oldfield “Dynamic Partitioning for a Distributed Virtual Environment” *Department of Computer Science. The Chinese University of Hong Kong*, (1998).
11. J.Lui, M.Chan, “An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems”, *IEEE Trans. Parallel and Distributed Systems*, Vol. 13, March (2002).
12. P.Morillo, M.Fernández, J.Orduña “A Comparison Study of Modern Heuristics for Solving the Partitioning Problem in DVE”. *Proc. of 2003 Int. Conference on Computational Science and its applications (ICCSA’ 2003).* Montreal, Canada. (2003).
13. T. Nitta, K. Fujita, S. Cono. “An Application Of Distributed Virtual Environment To Foreign Language”. *Proceedings of FIE2000. IEEE Education Society*. Kansas City, Missouri. October (2000).
14. S.Pekkola, M.Robinson, M.Saarinen, J.Korhonen, S.Hujala, “Collaborative Virtual Environments in the Year of the Dragon,” *Proceedings of the third international conference on Collaborative Virtual Environments (ACM),* pp.11-18, (2000).
15. J.M. Salles, R. Galli, A. Almeida, C. Belo, J. Rebordão. “mWorld: A Multiuser 3D Virtual Environment”. *IEEE Computer Graphics March-April 1997* (Vol. 17, No. 2).
16. S. Singhal, and M. Zyda, *“Networked Virtual Environments Design and Implementation”*. Addison Wesley, July (1999).
17. D.Snowdon, C.Greenhalgh, S.Benford, et al,"A Review of Distributed Architectures for Networked Virtual Reality", *IEEE VRAIS, Virtual Press*, Vol. 2, No. 1,155-175 (1996).
18. M.Zyda, J.Pratt, R.David, et al. *"NPSNET and the Naval Postgraduate School Graphics and Video Laboratory*," Presence, Vol. 2, No. 3., pp. 244-258.