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
In a Distributed Virtual Environment system several users connected from different computers can share the same virtual world. Current 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 generated by an avatar to the server where is allocated 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 our model this behaviour has been successfully correlated with the movement rate of the avatars and a new parameter proposed, called factor of presence, which measures the density of avatar in a virtual scene.

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

1. Introduction

Distributed Virtual Environments (DVE) are systems where many users connected from different computers share the same 3D virtual scene in the Internet [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 of the simulation is represented by an entity, usually 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, as the number of participants in the DVE grows [4] the scalability of the system becomes in a key issue in order 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 is a multiple-server architecture. In these architectures, also called server-network or hierarchical, simulation control is organized by interconnected servers. Multiplatform clients are allocated in one of these servers, so that for each avatar update a client sends one update message to a server, and the server propagates the message to other servers and clients. In order to avoid a message outburst when the number of clients increases, areas of influence (AOI) are defined for each one of the avatars. In this way, messages are only propagated from one avatar to the avatars that fall into its AOI. Depending on which is the destination server, two kinds of messages are defined in a DVE scheme (see figure 1): Fast intra-server communications when both the sender and receiver are allocated in the same server, and inter-server otherwise. For each local update, clients only send one message to a server and receive messages from a server in order to update all avatars' neighbours. Therefore, they must complete very little workload, storage or messaging in order to maintain a consistent state among many avatars in a big DVE.

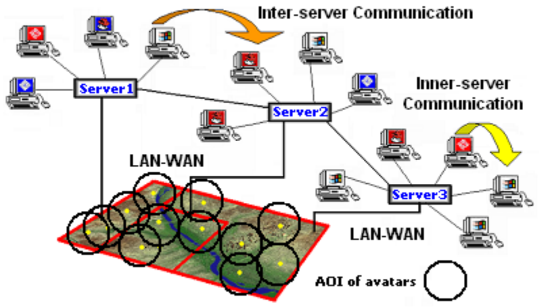

Figure 1: *An example of a basic DVE system developed on a multiple-server architecture composed by 3 servers which are simulating 13 avatars.*

Since the workload of the system is produced by the avatars, the server-network approach divides the overall workload into each server in order to improve the scalability of the DVE. In order to design actually and efficient DVE systems the partitioning problem appears [10] when the workload of each server wants to be balanced and the amount of inter-server messages minimized. 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 allocated is modelled as a fixed value. During the simulation time, they maintain this permanent value independently to any inherent factor.

In this paper, we present a new model for describing the workload of the avatars that are running in a DVE. Experimental results show that this workload is represented by a variable function. Furthermore, for every 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, a collaborative driving simulator [5].

The rest of the paper is organized as follow: Section 2 describes how has been traditionally handled the partitioning problem in DVE systems. Section 3 details the proposed avatar characterization that allows to experimentally study the behaviour of DVE systems. 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, that 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 increasing 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 like 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. Depending on their origin and destination avatars, messages in a DVE system can be intra-server or inter-server messages.

In order to simulate and 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 estimate the efficiency of a DVE considering two factors. The first one checks how is distributed computing workload of the system generated by the avatars of the DVE. Avatars should be proportionally allocated in 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. In first place, 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 topologically nearby if it is possible to find avatars allocated to them in such a way that the intersection of their AOI is not empty. In second place, 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 is very difficult to obtain optimal features if all the servers of the DVE are not involved in the load balancing, [12] and [11] describes global schemes of allocation. Based on modern heuristic or custom programming, respectively, they perform an algorithm which obtains the target server for every avatar of the system.

Although all these approaches maximize throughput, storage and communication requirements for maintaining a consistent DVE, they are based on a simplified model of behaviour of avatars. 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 to any parameter of the DVE.

3. Characterization Setup

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, 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. Concretely, 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 for assigning a given server to each new avatar. In this way, once the network address and the port number where the main server is listening, avatars can join the simulation through this main server, that assigns each new avatar to one of the servers in the system. 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 also 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 per second. Each time an avatar performs a movement, he notifies that movement to the server he 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 these avatars, they return an ACK message to the server, that 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 these average system responses to their respective servers, and the servers then computes the average system response for each server. Finally, the main server computes the average system response (ASR) for that simulation. A actually scalable DVE system must keep this measure as low as possible as the number of avatars in the system increases. When the rate of movement of this avatars becomes faster it is more difficult to support low levels or ASR than when they are moving slowly. The same problem appears when avatars avoid even distribution within the scene. These behaviours will be parameterised and measure for characterizing the behaviour of DVE systems.

In addition to measure the ASR we have studied the *CPU load* as measure of utilization of each server in each simulation. Both parameters show the impact of an avatar when it is allocated by a server.

Finally, in order to evaluate the performance of each partitioning method, usually 3 different distributions of avatars in the virtual world are proposed in the literature: uniform, skewed and clustered distributions of avatars [3,11,12,14]. Following this distributions avatars are distributed in the scene.

4. Correlation and Simulation results

In this section we present the correlation and simulation results of the avatars' behaviour in a DVE system. These results have been obtained for the DVE model described 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 and simulate 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 and due to space limitations we only show the results of performed correlations for small worlds.

For the purpose of characterization of avatars, each client only simulates a given rate of movements through the virtual world. During the simulation it assumes 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.

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 attached to the same server different ASR and CPU loads can be obtained for the same configuration of the DVE system. These results did not agree with 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, new characterization of the avatars a DVE system are needed.

From the obtained results we have observed changes in ASR and CPU load when the avatars that are allocated in a server change the movements per second (rate movement) in the 3D virtual scene. This change lead a variation of the number of messages which has to be processed by the server and thus a modification in the CPU load necessary 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 X-axis the values of the movement rate accomplished by all the avatars of the simulation. The Y-axis shows ASR (left side) and cpu load (right side) values for the simulations performed with these partitions. Each point in the plot represents the average value of the ASR and cpu load 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 load in any case.

This figure clearly shows that the workload generated by a definite number or avatars which are running in a DVE system correlates with the movement rate of them.

Additionally, distribution of avatars in the virtual world determine the workload that each avatar adds to the server where that avatar is assigned. In fact when some avatars are allocated in a server if it is noticeable that its impact in the system, in terms of ASR and cpu load, grows as their Euclidean distance in the virtual world decreases. In this kind of systems, such as we have described in Section 1, 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 this 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 in 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 which 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. Figure 2-right shows the behaviour of a DVE system when presence factor of the avatars is properly modified. This figure shows why as a collection of avatars comes closer in the virtual space the workload generated by these avatars to the servers where they are attached increases. This grouping movement causes an increment of the “fp” of these avatars and therefore an incoming of the number of necessary messages for managing them. These new messages represents more cpu load for accomplishing this task.

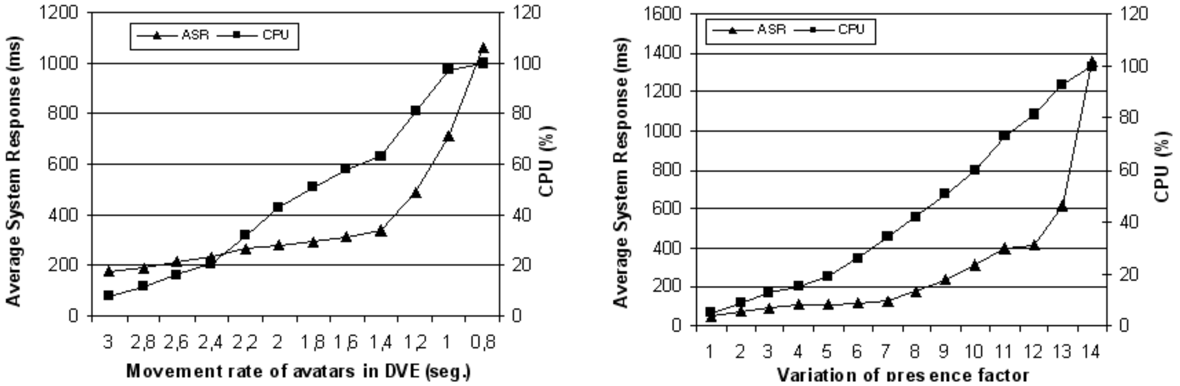

Figure 2:*Correlation of movement rate (left) and presence factor (right) of avatars with the ASR and CPU load in a DVE. Performance results had been obtained from a DVE composed of a 180 avatars simulated on 3 equal servers.*

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 studied the impact of the avatars, in terms of generated workload to the servers, while the simulation is running. Empirical results show that this parameter must not be modelled by a static value, as it has been traditionally done.

We have also proposed a new characterization of the behaviour of avatars. 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 the cpu load have been elected as performance evaluation parameters. The results show a good correlation of both the rate movement and the factor of presence with the system performance.

Therefore, we can conclude that in order to design actually and efficient DVE systems, the partitioning mechanisms should take into account the non-static contribution of avatars to the servers where they are attached. During the simulation and immediately before the partitioning task has to be executed the system should recalculate the new workload of all the 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 in the implementation of a new partitioning scheme based on a global load balancing method. This new scheme considers the ideas proposed in this paper and it 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.Martın 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.