

Improving the Performance of Distributed Virtual Environment Systems

Pedro Morillo, Juan M. Orduña, Marcos Fernández, and José Duato, *Member, IEEE*

Abstract—The last years have witnessed a dramatic growth in the number as well as in the variety of distributed virtual environment systems. These systems allow multiple users, working on different client computers that are interconnected through different networks, to interact in a shared virtual world. One of the key issues in the design of scalable and cost-effective DVE systems is the partitioning problem. This problem consists of efficiently assigning the existing clients to the servers in the system and some techniques have been already proposed for solving it. This paper experimentally analyzes the correlation of the quality function proposed in the literature for solving the partitioning problem with the performance of DVE systems. Since the results show an absence of correlation, we also propose the experimental characterization of DVE systems. The results show that the reason for that absence of correlation is the nonlinear behavior of DVE systems with regard to the number of clients in the system. DVE systems reach saturation when any of the servers reaches 100 percent of CPU utilization. The system performance greatly decreases if this limit is exceeded in any server. Also, as a direct application of these results, we present a partitioning method that is targeted to keep all the servers in the system below a certain threshold value of CPU utilization, regardless of the amount of network traffic. Evaluation results show that the proposed partitioning method can improve DVE system performance, regardless of both the movement pattern of clients and the initial distribution of clients in the virtual world.

Index Terms—Distributed applications, distributed/network graphics.

1 INTRODUCTION

PROFESSIONAL high performance graphic cards currently offer a very good frame-rate for rendering complex 3D scenes in real time. Fast Internet connections have also become available worldwide at a relatively low cost. These two factors have made possible the current growth of Distributed Virtual Environment (DVE) Systems. These systems allow multiple users, working on different client computers that are interconnected through different networks (and even through the Internet) to interact in a shared virtual world. This is achieved by rendering images of the environment as the user would perceive them if he was located in the virtual environment. Each user is represented in the shared virtual environment by an entity called an *avatar*, whose state is controlled by the user through the client computer. Since DVE systems support visual interactions between multiple avatars, each client computer needs to be reported about any positional change of those avatars in the neighborhood of the avatar controlled by that client computer. DVE systems are currently used in many different applications [39], such as civil and military distributed training [31], collaborative design [37], and e-learning [5]. Commercial multiplayer game environments are also considered DVE systems [2], [23], [42] since these systems

address the same issues and use the same simulation techniques that DVE systems do [38].

DVE systems can be based on different platforms. Different factors make DVE systems inherently heterogeneous:

- Each client computer controlling an avatar may be completely different. A very different performance can be found when comparing not only different processors, but also other computer resources such as memory size and speed, etc. Each client computer can have a different combination of these and other resources.
- Different networks can be found in a single DVE system. Since each network can have any existing network technology, different network technologies can be simultaneously found in a single DVE system.
- Depending on the application, avatars can show different communication rates. For example, the communication rate of avatars in a collaborative 3D environment may greatly differ from the communication rate of avatars in a 3D virtual military battle.

As a result, the establishment of a general methodology for the design of an efficient DVE system is a complex task. Nevertheless, some aspects of DVE systems have been studied:

- *Data Model*: This concept describes some conceivable ways of distributing persistent or semipersistent data in a DVE [28]. Data can be managed using a replicated, shared, or distributed strategy.
- *Communication Model*: The way that avatars communicate with each other has an effect on the system performance. Different communication schemes, such as broadcast, peer-to-peer [21], or unicast

• P. Morillo, J.M. Orduña, and M. Fernández are with the Departamento de Informática, Universidad de Valencia, Av. Vicent Adreés Estellés, s/n, 46100-Burjassot (Valencia), Spain.

E-mail: {Pedro.Morillo, Juan.Orduña, Marcos.Fernandez}@uv.es.

• J. Duato is with DISCA, Facultad de Informática, Universidad Politécnica de Valencia, Camino de Vera, s/n, 46022 Valencia, Spain.

E-mail: jduato@gap.upv.es.

Manuscript received 25 Nov. 2003; revised 5 May 2004; accepted 15 Sept. 2004; published online 20 May 2005.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0218-1103.

determine different network traffic patterns for exchanging information between avatars.

- *View Consistency:* This problem has already been defined in other computer science fields such as database management [6]. View consistency consists of ensuring that all avatars sharing a virtual space have the same vision on all the objects in that virtual space.
- *Network Traffic Reduction:* Keeping a low number of messages allows DVE systems to efficiently scale with the number of avatars in the system. Traditionally, techniques like dead-reckoning offered some level of independence to avatars [14], [18]. Also, broadcast or multicast solutions allow us to decrease the number of messages required for keeping a consistent state of the system [11], [22], [42].

One of the key issues in the design of a scalable and efficient DVE system is the *partitioning problem* [13]. It consists of efficiently assigning the workload (avatars) among the servers in the system. The partitioning problem determines the overall performance of the DVE system, since it has an effect not only on the workload that each server in the system is assigned to, but also on the interserver communications (and, therefore, on the network traffic). Some methods for solving the partitioning problem have already been proposed [13], [26], [27], [39]. However, there are still some features in the proposed methods that can be improved. For example, instead of using the ad hoc search method proposed in [27], several heuristic search methods can be used for solving this problem [32], [33]. Nevertheless, the main task that should still be done is to correlate the quality function used in the search method with DVE system performance. Otherwise, the proposed method could provide inefficient solutions for the partitioning problem.

This paper experimentally studies if the quality function used in the literature for solving the partitioning problem actually measures in an effective way the different performance levels that different partitions should produce in the same DVE system. Since the results show a total absence of correlation, this paper also studies the experimental characterization of DVE systems. This characterization study measures the impact of different parameters of the quality function on the performance of DVE systems [34]. The results show that the behavior of DVE systems is nonlinear with the number of avatars in the system, regardless of the value of the quality function. Also, the results show that the performance of DVE systems greatly decreases when any of the servers reaches 100 percent of CPU utilization. Therefore, as a direct application of these results, we propose a partitioning method that is targeted to keep all the servers in the system below a certain threshold of CPU utilization, regardless of the amount of network traffic. Evaluation results show that the proposed partitioning method can improve DVE system performance, regardless of both the movement pattern of avatars and also the initial distribution of avatars in the virtual world.

The rest of the paper is organized as follows: Section 2 describes the partitioning problem and the existing proposals for solving it. Section 3 details the proposed characterization setup, which allows us to experimentally study the behavior of DVE systems. Next, Section 4 presents the correlation and characterization results. Section 5 describes

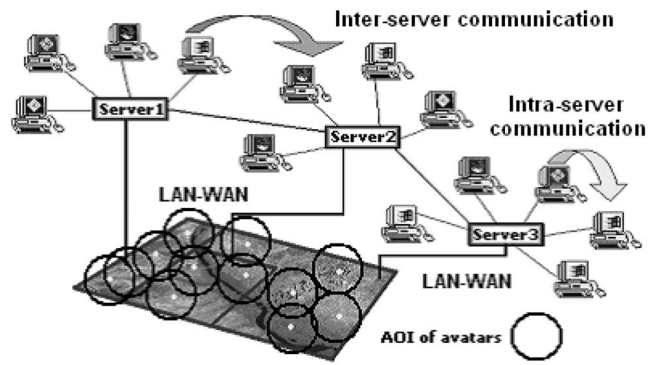


Fig. 1. An example of multiserver architecture for a DVE system.

a new partitioning method that takes into account the characterization results shown in the previous section. Finally, Section 6 presents some concluding remarks and the future work to be done.

2 THE PARTITIONING PROBLEM IN DVE SYSTEMS

Architectures based on networked servers are becoming a de-facto standard for DVE systems [17], [24], [28], [39], [40]. In these architectures, the control of the simulation relies on several interconnected servers. Client computers are attached to one of the servers in the system. When a client computer modifies the state (usually the position) of 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 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, different approaches have been proposed in order to limit the number of neighboring avatars that a given avatar must communicate with. Concepts like areas of influence (AOI) [39], [42], locales [3], or auras [15] define a neighborhood area for avatars in such a way that a given avatar must send notification of its movements (by sending an updating message) only to those avatars located in its neighborhood. Thus, the AOI of a given avatar determines the amount of network traffic generated by that avatar. Other approaches use three tiered architectures [1], [22], data filtering [41], or distributed cache management [8] in order to minimize the impact of network traffic on the performance of the DVE system.

Depending on their origin and destination avatars (client computers), messages in a DVE system can be intraserver or interserver messages. In order to design a scalable DVE system, the number of interserver messages should be minimized. Since intraserver messages only require a single server, the more intraserver messages there are, the fewer resources the application will require to exchange the same number of messages. Fig. 1 shows an example of a DVE system with a multiserver architecture, composed of three servers. It also shows an example of both intraserver and interserver communication.

Lui and Chan have shown the key role of finding a good assignment of avatars (clients) to servers in order to ensure both a good frame rate and a minimum network traffic in DVE systems [27]. They propose a quality function for evaluating each assignment of clients to servers (partition), denoted as C_p . This quality function takes into account two parameters. One of them measures how balanced the computing workload is in that partition, which is denoted as C_p^W . In order to minimize this parameter, the computing workload should be proportionally distributed among all the servers in the DVE system according to the computing resources of each server. The other parameter of the quality function measures the overall interserver communication requirements, which is denoted as C_p^L . In order to minimize this parameter, avatars sharing the same AOI should be assigned to the same server. Quality function C_p is defined as

$$C_p = W_1 \cdot C_p^W + W_2 \cdot C_p^L, \quad (1)$$

where $W_1 + W_2 = 1$. W_1 and W_2 are two coefficients that weight the relative importance of the computational and communication workload, respectively. These coefficients should be tuned according to the specific features of each DVE system. Thus, if the servers in the DVE system are connected through low performance networks (i.e., the Internet), then the quotient W_1/W_2 must be close to zero. On the contrary, if the servers are interconnected using high performance networks, the quotient W_1/W_2 must be close to one. Using the quality function C_p (and assuming $W_1 = W_2 = 0.5$), Lui et al. propose a partitioning algorithm that must be periodically executed [25], [26], in order to adapt the partition to the current state of the DVE system (avatars can join or leave the DVE system at any time and they can also move everywhere within the simulated virtual world). Lui and Chan have further suggested a testing platform for the performance evaluation of DVE systems, as well as a parallelization of the partitioning algorithm [27].

Other approaches for solving the partitioning problem have been also proposed. One of them [29] groups avatars following regular distributions. In order to ensure good performance, this approach generates a number of regular distributions equal to the number of servers in the DVE system. However, this proposal does not obtain good performance when avatars are located following a nonuniform distribution.

A different approach rejects dynamic concepts associated with avatars like AOI, aura, or locale [40]. This proposal divides the 3D virtual scene into a regular grid. A multicast group is created for each grid cell in such a way that avatars sharing a cell also share multicast packets and are assigned to the same server. Although this approach provides a fast way of solving the partitioning problem, the performance of the static partitioning is quite low when avatars are located following a clustered distribution. In this case, the servers controlling the areas where clusters are located are overloaded, increasing the overall cost of the quality function. A similar idea has been also suggested using unicast communications [7].

The partitioning method proposed by Lui and Chan currently provides the best results for DVE systems. However, the correlation of the quality function C_p

proposed by Lui and Chan [27] with DVE system performance has not been studied yet and the parameters W_1 and W_2 should still be tuned. In this sense, the characterization of DVE systems is essential to design partitioning strategies that actually improve the performance of DVE systems.

3 CHARACTERIZATION SETUP

We propose the characterization of generic DVE systems by simulation. The evaluation methodology used is based on the main standards for modeling collaborative virtual environments, such as FIPA [12], DIS [10], and HLA [20]. We have developed a simulation tool (written in C++) which models the behavior of a generic DVE system composed of several interconnected servers. Concretely, we have implemented a set of multithreaded servers (each server on a different, dedicated computer). Each thread in a server uses blocking sockets for communicating with a client computer. Each client controls the behavior of a single avatar and it is also implemented as a multithreaded application. One of the threads manages the communication with the server assigned to that client, while another thread manages user information (current position, network latency, etc.).

Our simulator models a DVE system composed of S interconnected servers and n avatars. Following the approach specified by FIPA and HLA standards, one of the servers acts as the main server (called *Agent Name Service* in [12] or *Federation Manager* in [20]) and it manages the whole system. Avatars can join the simulation through this main server, which uses a partitioning file for assigning each new avatar to one of the servers in the system. In each simulation, all avatars sharing the same AOI must communicate among them (by sending updating messages) in order to notify their position in the 3D virtual world. The message structure used for notification of avatar movements is the *Avatar Data Unit (ADU)* specified by DIS [10].

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, the client computer controlling that avatar sends notification of that movement to its server by sending a message with a timestamp. The server must then send notification of the movement to all the client computers controlling the avatars in the same AOI of the sender avatar. When that notification arrives at the destination client computers, they return an ACK message to the server, which in turn propagates these ACK messages to the sender client computer.

When an ACK message arrives, the sender client computer can measure the round-trip delay for each message sent. We have denoted such round-trip delay (measured in real-time) the *system response*. Since destination client computers send the ACK message back to the client computer controlling the sender avatar, the same clock is used for both the initial and final timestamp. In this way, clock skewing is avoided when computing system latency. When a simulation ends, each avatar (the client computer controlling each avatar) has computed the average system response for communicating with the avatars in its AOI. At this point, all avatars send their

average system responses to their respective servers, which in turn compute their own average system response. Finally, starting from the average value provided by each server, the main server computes the average system response for the entire system and for that simulation, denoted as the *average system response (ASR)*. An efficient DVE system must keep ASR value as low as possible, regardless of the number of avatars in the system. On the other hand, system throughput consists of the maximum number of avatars that the system can support without greatly increasing the ASR. Since the ASR (average system latency) shows how fast users perceive the system, we have considered the ASR as the main performance measure in order to characterize the behavior of DVE systems.

4 SIMULATION AND CORRELATION RESULTS

In this section, we present the results provided by the simulation model described in the previous section. First, we present the characterization of the workload generated by avatars in a DVE system. Next, we present the correlation of the quality function C_p with the DVE system performance. Finally, we present the characterization of DVE systems [34].

We have performed approximately 2,000 experiments on a wide variety of DVE systems with a multiserver architecture, ranging from small virtual worlds (three servers and 180 avatars) to large virtual worlds (900 avatars and six servers). Since we have obtained very similar results in all of them, for the sake of brevity, we present only the results for a small DVE configuration.

Due to characterization purposes, in each simulation, each client has only performed a fixed number of movements at a fixed rate. We have also assumed that no changes are produced in any static element of the virtual world and the AOI of avatars has the same size during the whole simulation. Although these assumptions are not realistic, they only have an effect on the workload produced by avatars, but they do not have any effect on the behavior of DVE servers.

The hardware platform used for simulating both clients and servers in the DVE system have been PCs 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 the same PC. We have used a 10 Mbps Ethernet as the interconnection network. The simulation tool has been executed on a Windows 2000 Professional operating system.

4.1 Characterization of Avatars

The first step in the characterization of DVE systems is to estimate the workload generated by avatars. It can be seen that, when avatars move faster, then the CPU utilization in DVE servers increases. The same behavior can be observed when avatars are located following uneven distributions within the scene. Therefore, we have proposed a model that takes into account both factors in order to estimate the workload generated by avatars [35].

Concretely, we define the *presence factor* $P_f(i)$ as the number of avatars in whose AOI avatar i appears. This presence factor and the movement rate of avatar i will

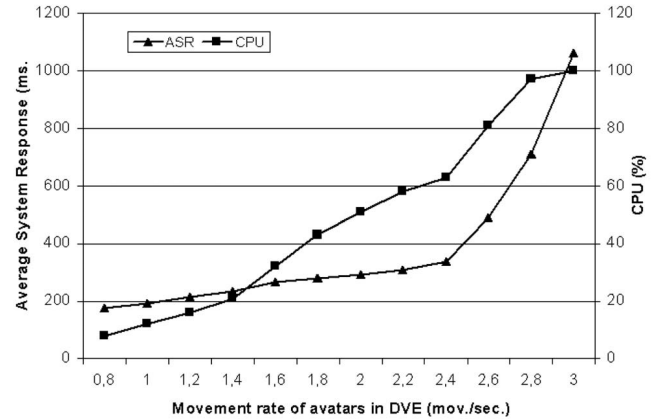


Fig. 2. Variation of CPU utilization and ASR with the movement rate of avatars.

determine the network traffic and the workload generated by avatar i [35]. Effectively, when the avatars that are allocated in a given server change the number of movements per second (movement rate) in the 3D virtual scene, then there is a variation in the number of messages to be processed by that server. Therefore, when a given avatar moves faster, it will require more CPU utilization in order to process its messages. We have measured the percentage increase of CPU utilization due to changes in the movement rate of avatars. A significant example is shown in Fig. 2. The results shown in this figure have been obtained from a DVE system composed of 180 avatars and three 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 several simulations performed with the same partition. 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 percent for the CPU utilization in any case.

Fig. 2 shows that the workload generated by avatars (in terms of both ASR and CPU utilization) is related to their movement rate. Therefore, the movement rate of avatars can provide a way of estimating the workload generated by avatars.

Additionally, Fig. 3 shows the behavior of a DVE system when the presence factor of avatars is modified. This figure shows on the X-axis the values for the presence factor of all avatars in a DVE system. The Y-axis shows the ASR and the average CPU utilization values for different simulations of the same partition when different values for the presence factor are used.

Fig. 3 helps to explain why, when avatars come closer in the virtual space, then the workload generated by these avatars increases. The grouping causes an increase of the P_f of the avatars and, therefore, an increase in the amount of updating messages sent by each one of them when they move. These new messages mean more CPU utilization for accomplishing the task. Based on these results, we have proposed a method for estimating the workload that avatars generate to a given DVE system [35].

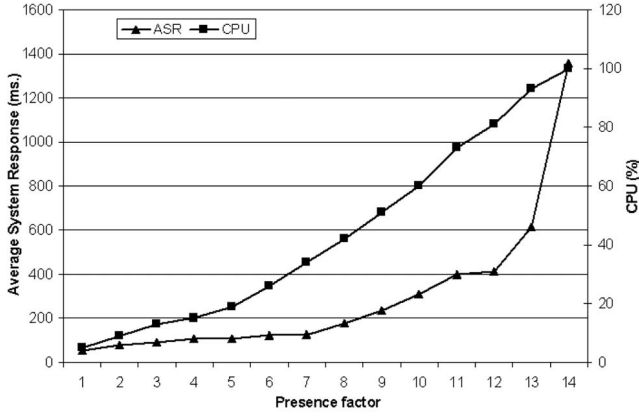


Fig. 3. Variation of CPU utilization and ASR with the presence factor of avatars.

In order to evaluate the performance of a given partitioning method, usually three different avatar distributions in the virtual world have been suggested in the literature: uniform, skewed, and clustered [27]. The reason for using different distributions is that they generate a different workload. Fig. 4 shows an example of these avatar distributions in a 2D virtual world. In this figure, the virtual world is a square and avatars are represented as black dots.

However, the workload that each avatar represents for the server to which it is assigned can be made independent of the distribution of avatars in the virtual world. Since different avatar distributions in the virtual world only differ in the workload generated by avatars, only one of the distributions is necessary for evaluation purposes. The effect of testing the partitioning method with different avatar distributions can be obtained by changing the presence factor and the movement rate of avatars. Therefore, for the purpose of characterizing DVE systems, we have considered only a uniform distribution of avatars, with the same AOI and the same movement rate for all avatars in the system.

4.2 DVE Systems Characterization

Taking into account the characterization of both the workload generated by avatars shown above, in this section, we analyze the correlation of quality function C_p and the DVE systems characterization.

Fig. 5 shows the performance results for a small DVE system composed of three servers and 180 avatars when different partitions (showing different values of the quality function C_p) are simulated. This figure shows on the X-axis

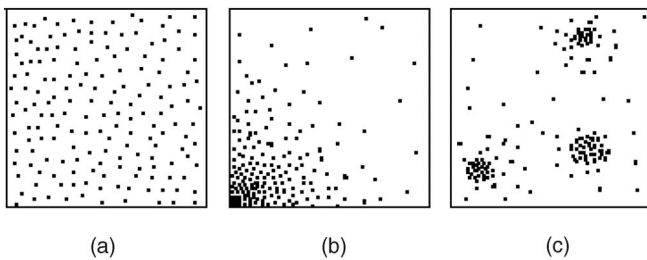


Fig. 4. Distributions of avatars in a 2D virtual world: (a) uniform, (b) skewed, and (c) clustered.

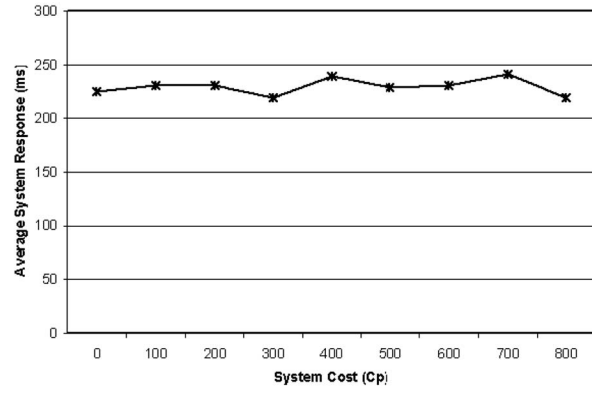


Fig. 5. Correlation of quality function C_p with average system response.

the values of the quality function C_p obtained for different partitions (assignments of avatars to the servers in the system). The Y-axis shows ASR values for the simulations performed with these partitions. Each point in the plot represents the average value of ASR obtained after 30 simulations of the same DVE system under the same workload. The standard deviation for any of the points shown in the plot was not higher than 25 ms in any case.

Fig. 5 clearly shows that C_p does not correlate with DVE system performance. The ASR values hardly change as C_p values are greatly increased. These results do not show a linear correspondence between the values of quality function C_p and the ASR values. Therefore, a characterization study is needed in order to capture the behavior of DVE systems.

The first goal of our characterization study was to find out the reason for the absence of correlation between C_p and the system performance. In order to achieve this goal, we studied the behavior of DVE systems as the number of avatars in the system increases. Fig. 6 shows the ASR values obtained for different DVE configurations, composed of three, four, and five servers, as the number of avatars in the system increases. In this figure, the X-axis shows the total number of avatars in the DVE system. The value of C_p associated with the simulated partitions was zero in all cases (perfect workload balancing and absence of inter-server messages, that is, C_p^L and C_p^W equal to zero). ASR values seem to be invariant with regard to the number of avatars in the system (ASR plot has the shape of a flat line)

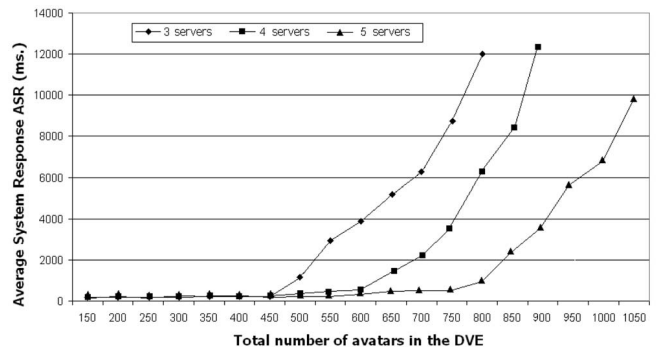


Fig. 6. Variation of average system responses with the number of avatars in the system.

TABLE 1
Simulation Results for a DVE Configuration of Five Servers and a "Perfect" Partition ($C_p = C_p^W = C_p^L = 0$)

Utilization (%)						
Number of avatars	CPU1	CPU2	CPU3	CPU4	CPU5	ASR
600 (120,120,120,120,120)	51	54	52	53	51	205.7
700 (140,140,140,140,140)	71	69	76	78	74	274.0
800 (160,160,160,160,160)	100	98	95	100	100	1001.4
900 (180,180,180,180,180)	100	100	100	100	100	3068.3
1000 (200,200,200,200,200)	100	100	100	100	100	4172.2
System Responses (ms)						
Number of avatars	SR-S1	SR-S2	SR-S3	SR-S4	SR-S5	ASR
600 (120,120,120,120,120)	206.16	199.5	204.9	227.1	190.6	205.7
700 (140,140,140,140,140)	284.2	260.8	265.9	287.4	271.6	274.0
800 (160,160,160,160,160)	998.5	965.5	897.5	1158.5	987.1	1001.4
900 (180,180,180,180,180)	3158.2	3152.2	2857.5	3215.5	2958.2	3068.3
1000 (200,200,200,200,200)	4095.1	4199.2	4201.2	4012.0	4193.2	4172.2

until a saturation point is reached. From that saturation point, ASR values greatly increase as new avatars are added to the system. These results clearly show that the behavior of a DVE system is nonlinear in relation to the number of avatars in the system. However, C_p function is defined (1) as a linear function of both workload balancing and interserver communications. Therefore, this nonlinear behavior can explain the absence of correlation shown in Fig. 5.

In order to determine the reason for the nonlinear behavior of DVE systems, Table 1 shows the CPU utilization and the average system responses corresponding to simulation results (DVE configuration of five servers) shown in Fig. 6. The upper part of this table shows the percentage of CPU utilization for each server in the system, as well as the ASR value (in milliseconds) obtained for each simulation. The left column shows the total number of avatars in the system for each simulation and how these avatars are distributed among the three servers. The lower part of this table shows the average system responses (in milliseconds) obtained for each server (SR-Sx) corresponding to the same simulations. Also, the global ASR value is shown again on the right column. This table shows that the DVE system reaches the saturation point when *some* of the servers in the system reach a 100 percent CPU utilization (for 800 avatars, the CPU utilization in servers 1, 4, and 5 reaches 100 percent, while, for this simulation, the ASR increases greatly). All the SR-Sx values increase as the respective CPU utilization does. Also, there are no significant differences between the three SR-Sx values, while the CPU utilization remains under 100 percent. However, when the CPU utilization surpasses 99 percent in some servers, then the system response for that server increases greatly, also increasing the global ASR accordingly. Therefore, we can conclude that the nonlinear behavior of DVE systems shown in Fig. 6 is due to the 100 percent limit in any CPU utilization.

Since the quality function C_p does not take into account the CPU utilization in order to measure the quality of a partition, it cannot take into account the nonlinear behavior of DVE systems as the number of avatars increases either. For example, a given partition can be composed of too many avatars for a given server (thus providing a poor system performance) and, at the same time, it can be a well-balanced partition, resulting in a very low C_p value.

These results indicate that, although a partition can provide perfect workload balancing and minimum interserver communication requirements (thus providing an optimum C_p value as the partitions in Fig. 6 do), ASR will be very poor if the number of avatars assigned to some servers requires a CPU utilization ranging from 95 to 100 percent or more. Also, given a DVE system and given a certain number of avatars in the system, the performance of that DVE system will remain practically invariant with regard to C_p if none of the servers reaches a CPU utilization close to 100 percent. This is the case for the simulations whose results are shown in Fig. 5, where only 90 percent of the CPU utilization was reached in the worst case.

In order to prove that the correlation results shown in Fig. 5 can be explained by the nonlinear behavior of DVE systems shown in Fig. 6, Fig. 7 shows the CPU utilization measured in each server for those simulations. Fig. 7 shows on the X-axis the values of the quality function C_p , as Fig. 5 does. However, in this figure, for each C_p value the CPU utilization of each server is shown as a bar diagram. The numbers on the top of each bar diagram indicate the number of avatars assigned to each server. The Y-axis show the percentage of CPU utilization.

Fig. 7 reveals that none of the servers in the DVE system reaches 100 percent of CPU utilization, even for the worst partition considered. Also, this figure shows that all the

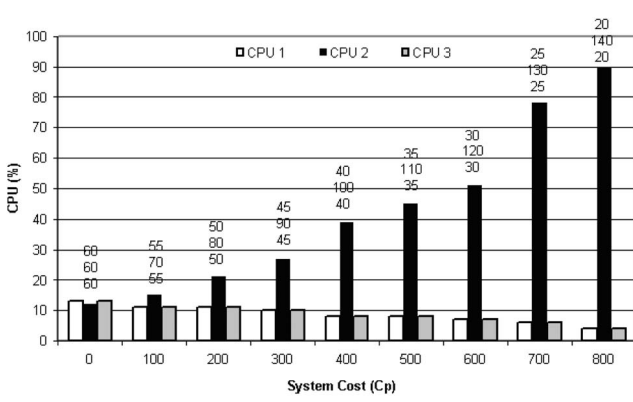


Fig. 7. CPU utilization for the experiments shown in Fig. 5.

new avatars are assigned to the same server in order to increase the value of the quality function C_p in different partitions. In this way, workload is unbalanced and, therefore, C_p is increased. When C_p reaches a value of 800 (the worst case considered in Fig. 5) server 2 only reaches 90 percent of CPU utilization. That is, this DVE system is working on the “flat line” zone of Fig. 6 for all the partitions considered. The resulting workload for server 2 generated by any of the partitions does not lead to system saturation. Therefore, all the partitions considered result in a similar ASR value.

The next step when making a methodical characterization study of DVE systems was to study the behavior of DVE systems as the two terms of the sum in (1) vary. Given a DVE system composed of three servers, Fig. 6 and Table 1 illustrate the behavior of the DVE system with the best partitions as possible (the values of C_p^W and C_p^L are both equal to zero).

Next, we have optimized the C_p^W value (workload balancing) and we have simulated (under the same workload) both a partition with an optimum (minimized) C_p^L value (interserver communication requirements) and also a partition with the worst C_p^L value as possible. Additionally, for the same DVE system, we have changed the partitions in order to obtain the worst C_p^W value possible, performing the same simulations with the partitions showing both the best and the worst values of the term C_p^L .

Table 2 shows the simulation results for the same DVE configuration whose results are shown in Table 1. However, this configuration has been simulated now under partitions with the worst value of C_p^L possible (the maximum number of interserver messages). When comparing Table 1 and Table 2 in a line-by-line analysis, we can see that an increase in the number of interserver messages produces a fixed (but not significant) increase in ASR values if the system is not saturated (the difference between the ASR values in both tables in the first two lines remains constant, about 48 ms.). The effect of the number of interserver messages in the ASR values becomes more important as servers enters deep saturation. Also, we can see that, for the same number of avatars (rows corresponding to 600 and 700 avatars), Table 2 shows slightly higher CPU utilization values. However, these differences are not significant. Therefore, if the workload is well balanced, then the number of interserver messages does not have a significant effect on the system throughput (the number of avatars that a given DVE system can manage without enter saturation) nor on the system latency (ASR).

Additionally, Tables 3 and 4 illustrate the CPU utilization and the ASR values for the same DVE system (composed of five servers) when it is simulated under

TABLE 2
Performance Results for a DVE Configuration of Five Servers and a Partition with Minimized C_p^W Term and Maximized C_p^L Term

Utilization (%)						
Number of avatars	CPU1	CPU2	CPU3	CPU4	CPU5	ASR
600 (120,120,120,120,120)	62	59	61	59	60	250.2
700 (140,140,140,140,140)	70	76	84	81	88	322.4
800 (160,160,160,160,160)	99	98	100	100	100	1121.7
900 (180,180,180,180,180)	100	100	100	100	100	3375.2
1000 (200,200,200,200,200)	100	100	100	100	100	4824.4
System Responses (ms)						
Number of avatars	SR-S1	SR-S2	SR-S3	SR-S4	SR-S5	ASR
600 (120,120,120,120,120)	259.5	249.7	265.4	249.8	226.7	250.2
700 (140,140,140,140,140)	341.9	298.8	292.5	336.1	342.6	322.4
800 (160,160,160,160,160)	1187.5	1085.8	1274.4	998.9	1062.1	1121.7
900 (180,180,180,180,180)	3467.4	3254.0	3143.3	3537.1	3474.0	3375.2
1000 (200,200,200,200,200)	4458.6	4658.2	4987.6	5198.3	4819.3	4824.4

TABLE 3
Average System Responses and CPU Utilization Values for a Partition where the Term C_p^W Is Maximized
and the Term C_p^L is Minimized

Utilization (%)						
Number of avatars	CPU1	CPU2	CPU3	CPU4	CPU5	ASR
600 (120,120,120,120,120)	55	53	50	56	51	237.2
625 (120,120,145,120,120)	59	50	74	58	53	246.0
650 (120,120,170,120,120)	52	51	100	54	54	605.9
675 (120,120,195,120,120)	53	57	100	56	53	2874.6
700 (120,120,220,120,120)	56	56	100	53	59	4824.4
System Responses (ms)						
Number of avatars	SR-S1	SR-S2	SR-S3	SR-S4	SR-S5	ASR
600 (120,120,120,120,120)	234.4	243.8	231.3	253.9	222.8	237.2
625 (120,120,145,120,120)	229.1	241.3	305.1	227.5	227.1	246.0
650 (120,120,170,120,120)	225.7	230.0	2101.8	232.0	239.9	605.9
675 (120,120,195,120,120)	237.0	241.3	13435.1	231.8	227.9	2874.6
700 (120,120,220,120,120)	4458.6	4658.2	4987.6	5198.3	4819.3	4824.4

TABLE 4
Performance Results for a Partition where Both C_p^L and C_p^W Terms Are Maximized

Utilization (%)						
Number of avatars	CPU1	CPU2	CPU3	CPU4	CPU5	ASR
600 (120,120,120,120,120)	57	58	51	57	50	244.4
625 (120,120,145,120,120)	60	56	77	54	57	339.6
650 (120,120,170,120,120)	52	54	100	57	58	1623.4
675 (120,120,195,120,120)	56	55	100	59	56	5087.0
700 (120,120,220,120,120)	58	61	100	52	55	9825.9
System Responses (ms)						
Number of avatars	SR-S1	SR-S2	SR-S3	SR-S4	SR-S5	ASR
600 (120,120,120,120,120)	241.4	239.6	241.0	261.5	238.7	244.4
625 (120,120,145,120,120)	301.6	309.5	399.6	320.4	367.1	339.6
650 (120,120,170,120,120)	1012.5	1249.4	3129.5	1824.6	900.9	1623.4
675 (120,120,195,120,120)	3996.4	4363.3	8313.4	4040.6	4721.1	5087.0
700 (120,120,220,120,120)	1020.7	11212.3	21123.4	8556.4	7216.5	9825.9

partitions where the term C_p^W is maximized (the workload is unevenly distributed). Table 3 shows the performance results for a partition that minimizes the term C_p^L (the number of interserver messages), while the partition whose results are shown in Table 4 maximizes this term. The most important result that Tables 3 and 4 show is that the saturation point of a DVE system is reached when *any* of the servers reaches the 100 percent limit of CPU utilization.

Effectively, in both tables, only server 3 reaches 100 percent of CPU utilization for 650 avatars and, for that simulation, ASR goes off in both tables. This effect cannot be clearly observed in Tables 1 and 2 because, in those partitions, the workload is perfectly balanced and, therefore, several servers reach 100 percent of CPU utilization in those tables simultaneously.

A line-by-line comparison of Tables 3 and 4 shows that the CPU utilization values are very similar in both tables. This result indicates that the term C_p^L does not have any effect on the system throughput when the term C_p^W is worsened (the system reaches saturation with 650 avatars in both tables). However, the system responses obtained for each server greatly differ between these two tables. When a given server reaches saturation (650 avatars, server S3), if the number of interserver messages is minimized (Table 3), then the avatars assigned to other servers do not use this server for communication purposes and, therefore, the saturation of server S3 does not affect the average latencies obtained for the rest of the servers. On the contrary, if the term C_p^L is maximized (Table 4), then the saturated server is used by avatars assigned to other servers for sending updating messages. Therefore, the system responses for the rest of the servers are also greatly increased. This behavior explains the difference between these two tables in the ASR value obtained for 650 avatars. This value is 605 for 650 avatars in Table 3, while it increases up to 1,623 in Table 4 for the same number of avatars. That is, if the workload is not properly balanced, then the saturation of a given server is propagated faster to other servers, greatly increasing the ASR.

A comparison of Tables 3 and 4 with Tables 1 and 2 shows that balancing the workload (minimizing the term C_p^W of quality function C_p) results in the improvement of the DVE system throughput (the number of avatars supported by the system while still providing a reasonable ASR). Effectively, the same DVE system composed of five servers reaches saturation with 800 avatars in Tables 1 and 2, while it reaches saturation with 650 avatars in Tables 3 and 4. However, there are no significant differences between the ASR values in the first two lines of Table 1 and the corresponding values in Table 3 nor between the corresponding values of Table 2 and Table 4 on the first two lines. These results indicate that the term C_p^W has an effect on the system throughput, but it does not have a significant effect on the ASR. On the contrary, the term C_p^L mainly affects the ASR values (system latency). However, this effect is significant only when the system reaches saturation. If the system is working below the saturation point, then the impact of C_p^L is not significant. Therefore, this characterization study shows that the partitioning method should be targeted to balance the workload among the servers in the system in such a way that none of them reaches 100 percent of CPU utilization.

5 A NEW PARTITIONING METHOD

5.1 Method Description

The results in the previous section suggest that it is necessary to develop a partitioning method that is targeted to keep all the servers in the system below a certain threshold of CPU utilization, regardless of the amount of network traffic.

Recently, a partitioning method (based on a load balancing technique) that takes into account the nonlinear behavior of DVE servers has been proposed for solving the partitioning problem [4]. In this proposal, all objects within each region of the virtual world are managed by a given

server and an adaptive load balancing scheme is provided. Nevertheless, this scheme provides only local load balancing among the servers managing adjacent regions. The reason for such locality is the reduction of the number of interserver messages (in this proposal, when the AOI of a given avatar crosses the boundary of two servers, both servers are responsible for transferring object models to that avatar). However, the local scope of the load balancing technique limits the performance of the partitioning method, since the workload supported by a given server can only be distributed among some of the servers (those managing adjacent regions), instead of using all servers. The performance evaluation of this proposal considers that all avatars in the virtual world move circularly, starting and ending at the same location. Under this simplifying assumption of a uniform movement pattern of avatars, local load balancing provides a good performance. However, it is very unlikely that all avatars in a DVE system move following such a uniform pattern. DVE systems usually have certain “hot-points” which avatars tend to head for [16]. In other cases, like in 3D networked games, these hot-points are game resources (energy, weapons, etc.) that dynamically appear and disappear and only those avatars located within a given radius of the hot-points tend to approach these points [30]. In these situations, all the neighboring regions around the hot points tend to get crowded with avatars and, therefore, all the servers managing these regions tend to get saturated, while other servers support a low load. A global load balancing technique would use all the servers in the system and it could provide better performance than a local load balancing technique for such cases.

We propose a partitioning method that (unlike the partitioning method proposed in [4]) performs an object-oriented management of avatars. Servers manage the workload generated by objects (avatars), rather than regions of the virtual world. This approach allows a global scope in the load balancing technique, using any server in the DVE system for load balancing purposes when a given server reaches saturation. Concretely, we propose the *Avatar Load Balancing (ALB)* technique [36]. This proposal is based on a server-initiated load balancing technique developed for distributed operating systems [9]. ALB starts with an initial assignment of avatars to servers. In order to obtain this initial assignment, a GRASP-based algorithm [32] or any other algorithm can be used. As avatars move within the virtual world freely, both the CPU utilization and also the number of avatars that each server supports are constantly monitored. From these measurements, the workload that each avatar adds to its server is statistically computed, taking into account the two factors shown in [35]: the movement rate of that avatar and the *presence factor* (P_f) of that avatar. In this way, the estimated workload that each avatar represents to its server is periodically estimated. When a given server S_x reaches 99 percent of CPU utilization, then the ALB algorithm is started. The threshold value (99 percent) should be tuned according to the system latency in order to avoid having any server reach 100 percent of CPU utilization during the time interval required to compute the ALB algorithm and to reassign avatars. The purpose of the ALB algorithm is to select a number of avatars currently assigned to S_x and to assign them to the server with the lowest CPU utilization (denoted

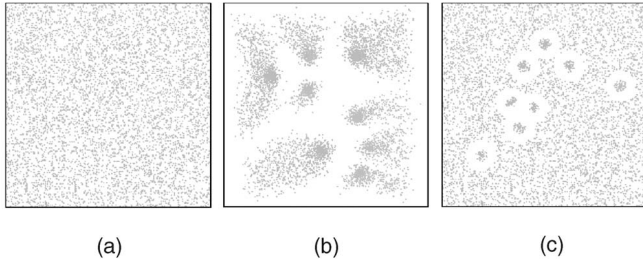


Fig. 8. Final distribution of avatars for (a) CCP, (b) HP-Near, and (c) HP-All movement patterns applied to an initial uniform distribution of avatars.

as S_y), in such a way that the resulting estimated CPU utilization of S_x is reduced to 90 percent. The threshold value should be low enough to ensure that saturation is avoided in S_x , while, at the same time, it should be as high as possible in order to avoid the *cascading effect* (if all the servers in the system are close to saturation, S_y may have a high CPU utilization. Assigning more avatars to S_y may lead this server to reach saturation, which in turn leads to starting the load balancing algorithm again).

The first step of the ALB algorithm consists of sorting the avatars currently assigned to S_x by their presence factor P_f . Then, the first avatars in this ranking are assigned to S_y until the estimated workload of the reassigned avatars represents at least 10 percent of CPU utilization in server S_x . The reason for using the criterion of the presence factor is that usually the avatars with the highest presence factor are also the closest avatars in the virtual world. Therefore, most of the messages generated by these reassigned avatars will be intraserver messages. In this way, the proposed technique not only avoids the saturation of S_x (the main goal of the load balancing technique), but it also tries to reduce the amount of interserver messages (this feature helps to decrease system latency, as shown in the previous section).

5.2 Evaluation Results

In order to evaluate the performance of the ALB technique, we have evaluated the performance of different DVE configurations by simulation, as described in Section 3. For comparison purposes, we have simulated the proposed technique and the *Adaptive Region Partitioning Technique* (ARPS) [4], the only existing partitioning method that takes into account the nonlinear behavior of DVE servers with the number of avatars assigned to them. We have simulated DVE systems with three different movement patterns of avatars: Changing Circular Pattern (CCP) [4], Hot-Points-ALL (HP-ALL), and also Hot-Point-Near (HP-Near). CCP considers that all avatars in the virtual world move circularly, starting and ending at the same location. HP-ALL considers that there are certain “hot-points” where all avatars tend to head for sooner or later, as shown in [16]. Finally, HP-Near also considers these hot-points; however, only those avatars located within a given radius of the hot-points tend to approach these points [30]. An iteration in a given movement pattern consists of all the avatars in the system performing a movement. A simulation consists of 100 iterations. Additionally, different initial distributions of avatars in the virtual world (uniform, skewed, and clustered) have been considered, as in other studies [4], [27]. As an example, Fig. 8 shows the final distributions of avatars in a 2D virtual world when the different movement patterns are applied and the initial distribution is a uniform distribution of avatars.

TABLE 5
Results for the ARPS Technique with an Initial Uniform Distribution of Avatars and HP-ALL Pattern

	ARPS (Iteration number)				
	1	10	11	12	13
S0	9.4	98.5	59.6	86.1	97.6
S1	9.4	10.6	10.7	11.2	15.3
S2	8.6	36.2	47.9	59.6	86.1
S3	9.8	64.7	88.4	96.7	100
S4	8.7	21.1	82.8	95.9	75.6
S5	11.6	48.9	67.4	83.3	81.9
S6	8.9	19.1	18.1	31.5	90.3
S7	9.3	57.7	75.9	99.2	100
S8	8.6	68.4	82.2	99.1	100
$\sum \delta_w$	25	907	897	1069	997

Although we have simulated all the movement patterns mentioned above in this section, we only present the results of the experiments performed with HP-Near and HP-ALL patterns. The reasons are both space limitations and also that the CCP movement pattern hardly starts the load balancing algorithm if the initial workload does not lead to system saturation since all avatars follow a uniform circular path. The results for both a clustered and a skewed initial distribution are not presented either since they only differ with the results presented here in the initial level of workload that the DVE system supports. In this section, we present the results for HP-ALL and HP-Near movement pattern of avatars and for a uniform initial distribution of avatars.

We have tested a large number of different DVE configurations, ranging from SMALL virtual worlds (composed of three servers and 180 avatars) to LARGE virtual worlds (composed of 6,000 avatars and nine servers). Due to space limitations, we only present the results for a LARGE DVE configuration (the worst case for the partitioning method).

Tables 5 and 6 show the performance evaluation results obtained for ARPS and ALB load balancing techniques when the initial distribution of avatars is the uniform distribution and avatars move following the HP-ALL pattern. Both tables show the percentage of CPU utilization in each server after different iterations of the simulation. Each row shows the CPU utilization for a different server, and each column shows the CPU utilization for all the servers and for a given partitioning method after a given iteration. Additionally, the last row shows the resulting sum of standard deviations $\sum \delta_w$ for the average estimated workload w that DVE system servers support. This value measures how balanced the workload is after each iteration (the lower this value is, the better the workload balancing that is achieved). In order to make a fair comparison, in both strategies we have obtained the initial assignment of avatars by means of the same initial algorithm (the one proposed in [4]). Therefore, the first column in Tables 5 and 6 show the same CPU utilization.

TABLE 6
Results for the ALB Technique with an Initial Uniform
Distribution of Avatars and HP-ALL Pattern

	ALB (Iteration number)					
	1	11	12	13	14	15
S0	9.4	89.5	89.4	89.3	98.6	100
S1	9.4	42.1	52.7	88.5	97.2	100
S2	8.6	45.1	56.4	74.4	99.5	99.9
S3	9.8	75.0	93.8	89.4	97.4	100
S4	8.7	31.3	63.9	84.2	93.7	100
S5	11.6	55.6	69.5	78.5	98.3	99.6
S6	8.9	33.3	53.4	60.3	99.2	94.6
S7	9.3	74.2	92.8	89.5	96.6	98.4
S8	8.6	84.6	89.5	89.4	99.1	100
$\sum \delta_w$	25	475	35	128	210	296

Since the HP-ALL pattern is applied to all avatars, all of them tend to head for a given location. As a consequence, the presence factor of avatars greatly increases and so does the total amount of workload. Therefore, the CPU utilization increases in all servers from the left to the right columns of the tables.

Close examination of Tables 5 and 6 reveals that the DVE system quickly reaches saturation with the HP-ALL movement pattern since only 14 out of 100 iterations could be simulated (Table 6, ALB technique) without reaching the saturation point. From that point, the overall workload generated by avatars is too big for this DVE configuration (except for servers *S4* and *S6*, all servers show a CPU utilization of 99 percent or more).

On the other hand, Tables 5 and 6 show that the ALB method provides better balanced partitions than the ones provided by ARPS technique. As a result, the ARPS technique (Table 5) only keeps the system below saturation until iteration 12. Indeed, servers *S7*, *S8*, and *S3* reach 100 percent of CPU utilization in the 13th iteration, while in this iteration, server *S1* is below 16 percent of CPU utilization. In iteration 12, servers *S7* and *S8* reach 99 percent of CPU utilization, while server *S1* is below 12 percent. The reason for this behavior is the local scope of the load balancing technique. Under this movement pattern, all avatars tend to continuously concentrate on some points or regions of the virtual world, and the servers managing these regions cannot share that workload with the servers managing remote regions (with few avatars). Thus, this partitioning method cannot avoid having servers *S3*, *S7*, and *S8* reach saturation in iteration 13. On the contrary, the ALB technique (Table 5) is able to share the workload among all the servers and, thus, after iteration 11, the value of $\sum \delta_w$ is around half of the value provided by the ARPS method for the same iteration. As a result, the ALB technique keeps the system below the saturation point in iterations 12, 13, and 14, although, in iteration 14, all servers are around 94 percent of CPU utilization. The system only reaches saturation (iteration 15) when the overall workload reaches the maximum workload that the system can support (as avatars concentrate, the number of avatars in the AOI of these avatars increases and, therefore, the workload that each avatar adds also increases).

TABLE 7
Results for the ARPS Technique with an Initial Uniform
Distribution of Avatars and HP-Near Pattern

	ARPS (Iteration number)					
	1	11	12	13	14	15
S0	9.0	66.2	93.9	54.2	74.3	65.1
S1	9.1	9.0	8.8	8.5	8.5	12.7
S2	8.8	26.9	33.3	44.4	55.8	79.8
S3	9.8	50.6	64.4	88.7	97.2	59.6
S4	9.0	15.4	16.0	76.3	94.6	100.0
S5	11.3	35.5	42.4	60.0	74.2	100.0
S6	9.5	14.0	18.0	15.8	26.9	28.5
S7	9.2	37.5	53.0	69.5	91.0	100.0
S8	8.6	54.7	65.1	78.1	98.7	100.0
$\sum \delta_w$	22	643	897	865	1030	1615

Tables 7 and 8 show the results for an initial uniform distribution of avatars and an HP-near movement pattern of avatars. Since, in this pattern, only some of the avatars tend to concentrate around the hot-points, the workload generated by avatars in this simulations is lower than the workload generated for the HP-ALL case and it does not reach the total workload that the DVE system can support. Table 7 shows that the ARPS technique can only keep the system below the saturation point until the 15th iteration. Again, servers managing the crowded regions reach 100 percent of CPU utilization, while servers managing distant regions are around 12 percent of CPU utilization. On the contrary, Table 7 shows that the ALB technique balances the workload among all the servers, showing a significantly lower value of $\sum \delta_w$ for the first 15 iterations. As a result, this technique is able to keep all servers below 100 percent of CPU utilization during the whole simulation (100 iterations).

TABLE 8
Results for the ALB Technique with an Initial Uniform
Distribution of Avatars and HP-Near Pattern

	ALB (Iteration number)							
	1	12	13	14	15	16	25	100
S0	9.0	66.2	93.9	89.4	98.4	97.7	98.4	91.6
S1	9.1	9.0	8.8	38.9	89.5	96.5	87.5	92.6
S2	8.8	26.9	33.3	45.7	87.6	95.4	97.5	89.9
S3	9.8	50.6	64.4	81.1	74.1	89.6	89.6	98.6
S4	9.0	15.4	16.0	53.6	65.2	88.5	90.0	97.5
S5	11.3	35.5	42.4	52.9	89.6	94.6	90.4	90.2
S6	9.5	14.0	18.0	22.5	95.2	97.6	97.5	97.2
S7	9.2	37.5	53.0	65.9	96.6	96.9	94.1	94.3
S8	8.6	54.7	65.1	79.6	97.1	98.1	77.5	92.9
$\sum \delta_w$	22	643	897	214	74	89	61	40

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented the experimental correlation of the quality function proposed in the literature for solving the partitioning problem (C_p) with the performance of DVE systems. Since results show an absence of correlation, we have also proposed a characterization study of DVE systems. DVE systems show a nonlinear behavior with the number of avatars in the system. The average system response (round-trip delay of messages notifying movements of avatars) remains practically invariant with the number of avatars in the system until the DVE system reaches a saturation point. This saturation point is given by the limit of 100 percent CPU utilization in any of the servers. Since quality function C_p does not take into account CPU utilization in order to measure the quality of a partition, it cannot take into account the nonlinear behavior of DVE systems with the number of avatars either.

We have also studied the effects of the two terms of C_p (workload balancing and the amount of interserver messages) on the performance of DVE systems. The results show that workload balancing mainly has an effect on system throughput, while the amount of interserver messages mainly has an effect on system latency. However, if none of the servers reaches saturation, then the amount of interserver messages does not have any significant effect on system latency. Therefore, in order to design an efficient and scalable DVE system, the partitioning method should be targeted to balance the workload among the servers in the system in such a way that none of them reaches 100 percent of CPU utilization.

Taking into account these results, we have proposed a partitioning method based on a load balancing technique. Unlike the only existing technique that takes into account the nonlinear behavior of DVE servers with the number of avatars they support, in this approach, servers manage the workload generated by avatars, not regions of the virtual world. The global scope of the proposed technique allows us to balance the workload generated by all of the avatars as much as possible. We have evaluated the proposed technique by simulation. The results show that the proposed strategy can improve the DVE system performance, particularly for nonuniform movement patterns of avatars. Moreover, the proposed strategy allows to keep the DVE system below the saturation level if the overall workload generated by all of the avatars is lower than the total workload that the DVE system can support, regardless of both the movement pattern and the initial distribution of avatars in the virtual world.

ACKNOWLEDGMENTS

This paper is supported by the Spanish MCYT under Grant TIC2003-08154-C06-04.

REFERENCES

- [1] H. Abrams, K. Watsen, and M. Zyda, "Three Tiered Interest Management for Large-Scale Virtual Environments," *Proc. ACM Symp. Virtual Reality Software and Technology '98*, Nov. 1998.
- [2] M. Abrash, "Quake's Game Engine: The Big Picture," *Dr. Dobbs's J.*, Spring 1997.
- [3] D.B. Anderson, J.W. Barrus, and J.H. Howard, "Building Multi-User Interactive Multimedia Environments at MERL," *IEEE Multimedia*, vol. 2, no. 4, pp. 77-82, Winter 1995.
- [4] N. Beatrice, S. Antonio, L. Rynson, and L. Frederick, "A Multi-Server Architecture for Distributed Virtual Walkthrough," *Proc. ACM Symp. Virtual Reality Software and Technology '02*, pp. 163-170, Nov. 2002.
- [5] C. Bouras, D. Fotakis, and A. Philopoulos, "A Distributed Virtual Learning Centre in Cyberspace," *Proc. Fourth Int'l Conf. Virtual Systems and Multimedia, VSMM '98*, Nov. 1998.
- [6] P.A. Bernstein, V. Hadzilacos, and N. Goodman, *Concurrency, Control and Recovery in Database Systems*. Addison-Wesley, 1997.
- [7] T. Carneiro and J. Arabe, "Load Balancing for Distributed Virtual Reality Systems," *Proc. Int'l Symp. Computer Graphics, Image Processing, and Vision (SIBGRAPI '98)*, pp. 158-165, Jan. 1998.
- [8] M. Capps, "The QUICK Framework for Task-Specific Asset Prioritization in Distributed Virtual Environments," *Proc. IEEE Virtual Reality Conf.*, Mar. 2000.
- [9] R. Chow and T. Jhonson, *Distributed Operating Systems and Algorithms*. Addison Wesley-Longman, 1997.
- [10] DIS, 1278.1 IEEE Standard for Distributed Interactive Simulation-Application Protocols (ANSI), DMSO, DoD High Level Architecture, 1997.
- [11] J. Falby, M. Zyda, D. Pratt, and R. Mackey, "NPSNET: Hierarchical Data Structures for Real-Time Three-Dimensional Visual Simulation," *Computers and Graphics*, vol. 17, no. 1, pp. 65-69, 1993.
- [12] FIPA Agent Management Specification, Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs/fipa00023/>, 2000.
- [13] T.A. Funkhouser, "Network Topologies for Scalable Multi-User Virtual Environments," *Proc. IEEE Virtual Reality Ann. Int'l Symp.*, Apr. 1996.
- [14] E. Frecon and M. Stenius, "DIVE: A Scalable Network Architecture for Distributed Virtual Environments," *Distributed Systems Eng.*, vol. 5, 1998.
- [15] F.C. Greenhalgh, "Awareness-Based Communication Management in MASSIVE Systems," *Distributed Systems Eng.*, vol. 5, 1998.
- [16] C. Greenhalgh, "Analysing Movement and World Transitions in Virtual Reality Tele-Conferencing," *Proc. Fifth European Conf. Computer Supported Cooperative Work (ECSCW '97)*, Sept. 1997.
- [17] C. Greenhalgh, A. Bullock, E. Frécon, D. Llyod, and A. Steed, "Making Networked Virtual Environments Work," *Presence*, vol. 10, no. 2, Apr. 2001.
- [18] E.P. Harvey, R.L. Schaffer, and S.M. McGarry, "High Performance Fixed-Wing Aircraft Simulation Using SIMNET Protocols," *Proc. 23rd Ann. Summer Computer Simulation Conf.*, pp. 965-970, July 1991.
- [19] J.C. Hu, I. Pyarali, and D.C. Schmidt, "Measuring the Impact of Event Dispatching and Concurrency Models on Web Server Performance over High-Speed Networks," *Proc. Second IEEE Global Internet Conf.*, Nov. 1997.
- [20] F. Kuhl, R. Weatherly, and J. Dahmann, *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Upper Saddle River, N.J.: Prentice-Hall PTR, 1999.
- [21] Y. Kawahara, H. Morikawa, and T. Aoyama, "A Peer-to-Peer Message Exchange Scheme for Large Scale Networked Virtual Environments," *Proc. Eighth IEEE Int'l Conf. Comm. Systems (ICCS 2002)*, Nov. 2002.
- [22] D. Lee, M. Lim, and S. Han, "ATLAS—A Scalable Network Framework for Distributed Virtual Environments," *Proc. ACM Collaborative Virtual Environments Conf. (CVE 2002)*, pp. 47-54, Sept.-Oct. 2002.
- [23] M. Lewis and J. Jacobson, "Game Engines in Scientific Research," *Comm. ACM*, vol. 45, no. 1, Jan. 2002.
- [24] J.C.S. Lui, M.F. Chan, and K.Y. Oldfield, "Dynamic Partitioning for a Distributed Virtual Environment," Dept. of Computer Science, Chinese Univ. of Hong Kong, 1998.
- [25] J.C.S. Lui and W.K. Lam, "General Methodology in Analysing the Performance of Parallel/Distributed Simulation under General Computational Graphs," *Proc. Third Int'l Conf. Numerical Solution of Markov Chain*, Sept. 1999.
- [26] J.C.S. Lui, M.F. Chan, K.Y. So, and T.S. Tam, "Deriving an Optimal Synchronization Interval for Supporting View Consistency in a Distributed Virtual Environment," *Proc. Fifth Int'l Workshop Multimedia Information Systems (MIS '99)*, pp. 99-106, Oct. 1999.
- [27] J.C.S. Lui and M.F. Chan, "An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 3, Mar. 2002.
- [28] M.R. Macedonia, "A Taxonomy for Networked Virtual Environments," *IEEE Multimedia*, vol. 4, no. 1, pp. 48-56, Jan.-Mar. 1997.

- [29] M.R. Macedonia, M.J. Zyda, D.R. Pratt, D.P. Brutzman, and P.T. Barham, "Exploiting Reality with Multicast Groups, *IEEE Computer Graphics and Applications*, pp. 38-45, Sept. 1995.
- [30] M. Matijasevic, K.P. Valavanis, D. Gracanin, and I. Lovrek, "Application of a Multi-User Distributed Virtual Environment Framework to Mobile Robot Teleoperation over the Internet," *Machine Intelligence & Robotic Control*, vol. 1, no. 1, pp. 11-26, 1999.
- [31] D.C. Miller and J.A. Thorpe, "SIMNET: The Advent of Simulator Networking," *Proc. IEEE*, vol. 83, no. 8, pp. 1114-1123, Aug. 1995.
- [32] P. Morillo and M. Fernández, "A GRASP-Based Algorithm for Solving DVE Partitioning Problem," *Proc. 2003 Int'l Parallel and Distributed Processing Symp. (IPDPS 2003)*, Apr. 2003.
- [33] P. Morillo, M. Fernández, and J.M. Orduña, "A Comparison Study of Modern Heuristics for Solving the Partitioning Problem in Distributed Virtual Environment Systems," *Proc. Int'l Conf. Computational Science and its Applications (ICCSA 2003)*, pp. 458-467, May 2003.
- [34] P. Morillo, J.M. Orduña, M. Fernández, and J. Duato, "On the Characterization of Distributed Virtual Environment Systems," *Proc. European Conf. Parallel Processing (Euro-Par 2003)*, pp. 1190-1198, Aug. 2003.
- [35] P. Morillo, M. Fernández, and J.M. Orduña, "On the Characterization of Avatars in Distributed Virtual Worlds," *Proc. EUROGRAPHICS 2003*, pp. 215-220, Sept. 2003.
- [36] P. Morillo, J.M. Orduña, M. Fernández, and J. Duato, "An Adaptive Load Balancing Technique for Distributed Virtual Environment Systems," *Proc. 15th IASTED Int'l Conf. Parallel and Distributed Computing Systems*, Nov. 2003.
- [37] J.M. Salles Dias et al., "mWorld: A Multiuser 3D Virtual Environment," *IEEE Computer Graphics and Applications*, vol. 17, no. 2, Mar.-Apr. 1997.
- [38] J. Smed, T. Kaukoranta, and H. Hakonen, "A Review on Networking and Multiplayer Computer Games," Technical Report 454, Turku Centre for Computer Science, Apr. 2002.
- [39] S. Singhal and M. Zyda, *Networked Virtual Environments. Design and Implementation*. Addison-Wesley, July 1999.
- [40] P.T. Tam, "Communication Cost Optimization and Analysis in Distributed Virtual Environment," Technical Report RM1026-TR98-0412, MPhil second term paper, Dept. of Computer Science & Eng., The Chinese Univ. of Hong Kong, 1998.
- [41] H. Trefftz, I. Marsic, and M. Zyda, "Handling Heterogeneity in Networked Virtual Environments," *Presence*, vol. 12, no. 1, pp. 38-52, Feb. 2003.
- [42] L. Zou, M. Ammar, and C. Diot, "An Evaluation of Grouping Techniques for State Dissemination in Networked Multi-User Games," *Proc. Ninth Int'l Symp. Modelling, Analysis, and Simulation of Computer and Telecomm. Systems (MASCOTS '01)*, Aug. 2001.



and cluster computing. For more details on his work, go to: <http://informatica.uv.es/~pmorillo>.



research is currently supported by the Spanish MCYT and it addresses cluster computing and distributed virtual environment systems.



Currently, he is involved in the development of a immersive visualization facility call VISIONARC (<http://robotica.uv.es/grupos/artec/english>).



José Duato received the MS and PhD degrees in electrical engineering from the Technical University of Valencia, Spain, in 1981 and 1985, respectively. Currently, Dr. Duato is a professor in the Department of Computer Engineering (DISCA) at the same university. He was also an adjunct professor in the Department of Computer and Information Science, The Ohio State University. His current research interests include interconnection networks, multiprocessor architectures, networks of workstations, and switch fabrics for IP routers. Professor Duato has published more than 280 refereed papers. He proposed a powerful theory of deadlock-free adaptive routing for wormhole networks. Dr. Duato served as a member of the editorial boards of the *IEEE Transactions on Parallel and Distributed Systems* and *IEEE Transactions on Computers*. He has served as cochair or a member of the program committee for more than 40 conferences, including the most prestigious conferences in his area (HPCA, ISCA, IPPS/SPDP, ICPP, ICDCS, EuroPar, and HiPC). He is a member of the IEEE and the IEEE Computer Society.

Pedro Morillo is an assistant professor in the Department of Informatics at the University of Valencia, Spain. In this department, he belongs to the computer graphics group (ARTEC), where he heads two projects on collaborative design and distributed virtual environments. Last July, he delivered his PhD dissertation that will be defended the last week of December 2004. His research addresses distributed virtual environments systems, load balancing, metaheuristics, and cluster computing.

Juan M. Orduña received the MS degree in computer engineering from the Technical University of Valencia, Spain, in 1990. He worked at Telefónica de España and Manpel Electrónica, S.A., and at the Technical University of Valencia as a computer engineer. He received the PhD degree in computer engineering from the University of Valencia in 1998. Currently, he is a lecturer professor in the Department of Informatics at the University of Valencia, Spain. His research is currently supported by the Spanish MCYT and it addresses cluster computing and distributed virtual environment systems.

Marcos Fernández received the Msc degree in physics and the PhD degree in computer science from the University of Valencia. His PhD degree has been oriented to the generation of traffic for driving simulators. He is a lecturer professor in the Department of Informatics at the University of Valencia, Spain. He is also the coordinator of the ARTEC Graphics Group at the Robotics Institute. Currently, his research addresses highly immersive virtual environments.

José Duato received the MS and PhD degrees in electrical engineering from the Technical University of Valencia, Spain, in 1981 and 1985, respectively. Currently, Dr. Duato is a professor in the Department of Computer Engineering (DISCA) at the same university. He was also an adjunct professor in the Department of Computer and Information Science, The Ohio State University. His current research interests include interconnection networks, multiprocessor architectures, networks of workstations, and switch fabrics for IP routers.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.