Exploiting Virtual Objects' Attributes and Avatar's Behavior in DVEs Partitioning

Christos Bouras[1, 2], Eri Giannaka[1, 2], Thrasyvoulos Tsiatsos[1, 3]
[1]Research Academic Computer Technology Institute, N.Kazantzaki Str., Patras University GR-26500 Rion, Patras, Greece,
[2] Computer Engineering and Informatics Dept., Univ. of Patras, Greece
[3]Aristotle University of Thessaloniki, Department of Informatics, PO BOX 888, GR-54124, Thessaloniki, Greece
bouras@cti.gr, giannaka@cti.gr, tsiatsos@csd.auth.gr

Abstract

Partitioning constitutes one of the most critical challenges a distributed virtual environment needs to handle and is related to the efficient assignment of the existing entities of the virtual world to the available resources of the system. The efficient partitioning arises from the need of high consistency maintenance among participating users' view, as virtual reality applications are characterized by the high degree of realism they aim to achieve. To the direction of partitioning algorithms, a lot of work has been done, which has been more intense due to the evolution of network technologies and the familiarization of users with virtual reality systems. This paper presents a partitioning approach, which is based on the degree of interaction of each virtual object and the behavior of the participating users, in terms of moving trends, which is usually directed by the actions they can perform on the surrounding objects. In particular, the partitioning takes place when the virtual world contains only objects, in the sense that no avatars are yet connected and objects' attributes are handled as indicators for avatars' future behavior in the system. This prediction of avatars' behavior based on objects' attributes defines the initial partitioning of the virtual space.

1. Introduction

The last decade Networked Virtual Environments emerged as efficient means and solutions for supporting collaboration and communication among scattered users. This emergence was importantly driven by the advances of hardware and software, the evolution of network infrastructures and the familiarization of users with the three-dimensional representation of information. The wide adaptation of the Internet and its establishment to a vital resource for the realization and completion of a wide variety of processes in combination with the high degree of realism achieved by virtual reality and the various applications developed played an important role for the wide exploitation of virtual environments from various types of end users. In particular, virtual environments, which could support either a large number of users or a number of demanding applications, were developed for meeting different types of needs, varying from entertainment and communication to collaboration at work, learning and training. For supporting large-scale applications, either to the number of users or resources needed, Distributed Virtual Environments (DVEs) emerged as a de facto solution. DVEs need to address more serious problems than traditional virtual reality systems. These problems are related, among others, to the control of network traffic, latency, and reliability. These characteristics, in turn, introduce a series of problems that need to be solved for achieving sustainability and for optimizing the performance of the distributed system. One of the key issues in the design of a scalable DVE system is the partitioning problem.

The partitioning problem is related to the efficient assignment of the workload to the servers of the system. To this direction many algorithms and techniques have been proposed, each of which aims at eliminating and overcoming certain aspects of this problem. The approach presented in [4] proposes a heuristic search based on Ant Colony Systems (ACS). This heuristic search method uses positive feedback to improve the use of good search paths, while using negative feedback to escape from local minima. Another approach [3] is to logically partition virtual environments by associating spatial, temporal, and functionally related entity classes with network multicast groups. This is accomplished by exploiting the actual characteristics of the real-world large-scale environments that are to be simulated, and by focusing an entity's processing and network resources to its area of interest via an Area of Interest (AOI) Manager. Another different approach rejects dynamic concepts associated to avatars like AOI, aura or locale [5]. This proposal divides the 3 Dimensional (3D) virtual scene in 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. Finally, Lui and Chan [1] have described the importance of finding a good assignment of the participating clients to the available servers for managing the workload and the communication cost and for achieving a better networking performance. This partitioning algorithm currently achieves very good best results for DVEs [2].

This paper presents an approach for handling the partitioning problem, which is based on objects' attributes and avatars' behavior. The approach aims at balancing the workload among the servers as well as at minimizing the communication among them. In particular, the approach proposes a technique for predicting avatars' movement within the virtual environment by taking into account the impact of objects' existence, in terms of the degree of interaction that the objects of the virtual environment carry. This technique, is then used for the partitioning of the virtual environment while it also encounters the impact (in terms of workload and communication cost) that the presence of objects and the interactions with users' avatars introduces.

The paper is structured as follows: Section 2 presents the partitioning approach proposed as well as the main parameters upon which the approach is based. Section 3 presents the partitioning algorithm and Section 4 describes the experiments conducted for proving the efficiency of the algorithm. Section 5 discusses the results of the experiments while Section 6 concludes the paper and presents the planned next steps.

2. Main Parameters of the Partitioning Approach

In their vast majority, virtual environments are comprised by two types of entities, avatars and objects. The objects contained in a virtual environment usually tend to support certain types of functionality mainly related to the context and scope of the virtual world. Thus, both the type and actions that can be performed on an object may vary among different virtual environments. However, their existence is not arbitrary, as virtual environments tend to simulate complete scenarios or processes. The approach presented in this paper is mainly driven by the assumption that the attributes of the participating entities affect users' behavior during a session. The algorithm for predicting the entities' behavior within the virtual environment encounters the following metrics for the existing entities:

- Area Of Interest (AoI) of the participating avatars, which is the region of the virtual world that, if an activity happens in this region, the avatar needs to know so as to update its own state and to make his/her view consistent
- Degree of Interaction: In a virtual environment, users have the ability to communicate with each other as well as with the objects of the world for achieving the goals that each of the environment aims to support. As mentioned above, the interactions that take place within a virtual environment are significantly affected by the number of actions allowed and supported by the system. In a world where only avatars exist, the interactions are mainly driven by the social behavior established between the participating users, which usually follow the basic principles of real world interactions. However, in a virtual environment, which, consists of objects as well, the behavior can be driven by the purpose of these objects and their role for achieving the goals that each environment aims to support. In particular, objects that allow a number of actions to be performed on them have a higher possibility of constituting a point of interaction with the users of the session. The number of actions that can be performed on an object may vary and can concern the modification of its location, size, shape, color, texture, etc. We define as object's degree of interaction (DoI) the number of actions that can be performed on each object. The prediction technique proposed encounters this degree of interaction for being able to predict the avatars' distribution within the virtual environment after a certain period of time.

3. Partitioning Approach

The objective of the partitioning in a DVE is the efficient assignment of the virtual world and its entities to the available servers of the system, so as to maintain a good overall performance. The efficient assignment is related both to balancing the workload among the servers as well as to minimizing the communication among them. Initially, the virtual world is divided in equal-sized cells. The number of cells is related to the average diameter D of avatars' Area of Interest (AoI) and all cells lie in one server. Each cell can comprise a number of objects. The workload for handling each cell depends both on the number of objects of this cell as well as on the DoI of these objects. In particular, the number of objects increases the workload that each cell introduces to the server, while DoI "forecasts" the workload that each object will introduce to the server when users will interact with it. Furthermore, DoI could be considered as indicator for avatars' behavior (in terms of motion and interaction) within the virtual environment. In particular, DoI value of an object could be used as prediction parameters in terms that when a number of objects is within the AoI of an avatar then the possibility of the avatar in selecting the object that it will visit and interact with is strongly related to the DoI of these objects.

Apart from a balanced workload distribution of the virtual world to the servers of the system, another important factor for the efficient partitioning is the minimization of the communication among participating servers. For handling this factor, the partitioning algorithm takes into account some additional observations. In particular, in the vast majority of virtual environments there are certain points/areas ("entry points"), where the user is transferred when entering the world or when transported from one place to another. We refer to these entry points as "hot spots", in the sense that in these areas there is often a high concentration of avatars. These areas either empty or not, add workload to the servers and should be therefore cautiously handled. When avatars enter the virtual environment or are transported from one point to another (hot spots) and given their AoI diameter D , will most likely move towards neighboring cells of the hot spots. Thus, based on this observation and in combination to the effect that DoI plays in avatars' behavior (motion and interaction), the algorithm, when distributing cells to the existing servers tries to keep the hot spots and its neighboring cells to the same server so as to minimize the communication cost among the servers that avatars' movement and interaction would introduce, if these cells relied in different machines. At this point it should be mentioned that the rational of the algorithm is to mark the "predicted" crowded places within the virtual environment and assign them to the available servers so that workload will be balanced. For handling cases where the number of hot spots within the virtual environment is less than the number of available servers, we introduce the term of the "virtual" hot spot. The word "virtual" indicates that the selected cell is not an actual entry point to the virtual world, but it introduces some heavy workload to the server. The incorporation of virtual hot spots allows us to distinguish the crowded places of the virtual world and contribute to a more balanced distribution of workload.

Based on the main concepts of the partitioning approach, described above, the partitioning approach in algorithmic terms is presented.

```
1. Divide the virtual world into $N$ disjoint cells
   of area $D^2$, where $D$ is the average diameter
   of the Area of Interest ( $AoI$ ) of the avatars
/* Create a graph for the virtual environment*/
2. For each cell create a node and calculate {
       o the number of active, static, multi-
         user objects $Nob_i$ that reside in this
         cell
       o for each object $ob_i$ calculate the
         workload it introduces to the server
         based on its $DoI$ as follows:
         $ob_j\_cw \times doi\_ob_i \div 5$ /* where
         $doi$ is the normalized value (in scale
         of 1 to 10) of $DoI$
       o Add the workload of each object to
         the total workload of each
         cell: $C_i\_cw = C_i\_cw + ob_j\_cw$ }
3. Find cells which constitute entry points and
   mark them as hotspots:
   if ( $C_i.kind = entry$ ) { $C_i.hot\_spot = true$ }
4. for ( $C_i.hot\_spot = true$ ) {
       o find its neighbors $C_j$ and create an
         edge: addEdge( $C_i$ , $C_j$ )
       o mark neighbors:
         $C_j.neighbor = true$ }
5. Calculate the total number of hot_spots
6. If ( $hot\_spot\_num < Server\_num$ ) {
/* Create additional "virtual" hot spot areas from cells
that are not neighbors of hot_spot cells based on the
maximum workload so that all servers will be assigned
with at least one hot spot */
       o For
         ( $Server\_num - hot\_spot\_num$ ){
             ▪ For ( $C_i.neighbor != true$ ) {
                   Select max $C_i\_cw$
                   Mark $C_i$ as hot spot:
                   $C_i.hot\_spot = true$ }}
       o Find its neighbors $C_j$ and create an
         edge: addEdge( $C_i$ , $C_j$ )
```

```
/*Begin with the assignment of cells to the available servers using round robin */
7.  for ($i = 0; i < Server\_num; i++;$) {
        o  assign one hot spot $C\_hs$ to each server }
8.  for ($i = 0; i < Server\_num; i++;$) {
        o  for ($C\_hs_i$) {
        o  while ($C\_hs_i$ has neighbors) { assign the neighbor of $C\_hs_i$ for which max $Cj\_cw$ } } }
/* When all servers are assigned with at least one hot spot and all of the hot spot's neighbors, we assign the remaining cells to the servers, by examining whether this cell is neighbor to one cell of the partitions that came up from the previous steps and by calculating the workload of each server and selecting the one with the minimum workload each time */
9.  for (remaining cells) {
        o  select cell with max $C_i\_cw$
        o  find neighbors of $C_i$
        o  while ($C_i$ has neighbors $C_j$) {
               ▪  find server number of $C_j$
               ▪  calculate server workload: $s\_cw_j$
               ▪  select $\min(s\_cw_i)$ }
               ▪  set $server[Server\_num]$ as $candidateServer$ }
        o  else for ($i = 0; i < Server\_num; i++;$) {
        o  calculate server workload: $s\_cw$
        o  select $\min(s\_cw)$
        o  set $server[Server\_num]$ as $candidateServer$ }
10. for (remaining cells) {
        o  select cell with max $C_i\_cw$
    assign cell to $candidateServer$ }
```

4. Experiments

This section presents the experiments conducted for proving the algorithms' efficiency under different types of objects' and entry points' distributions within a virtual environment. In particular, we examine three different cases (scenarios) for the same virtual world, where the entry points (hot spots) and the virtual hotspots are differently placed within the environment. For each of these cases we present the following: a) the initial state of the virtual environment, which consists only of objects, b) the final distribution of cells and entities (objects and avatars) to servers after the application of the proposed algorithm and c) the results of the total cost (workload and communication), when avatars enter the virtual environment. For the presentation of the results in Figure 1(c), Figure 2(c) and Figure 3(c) we define as "Initial Step", the step where cells and objects are assigned to the available servers, as "Final Step" the situation where avatars have also joined the virtual world, as "T_Wload", which stands for Total Workload, the workload distribution among the servers of the system, as "Id_Wload", which stands for "Ideal Workload", the situation where workload is perfectly balanced among the servers, and as "Com_Cost" the overall communication cost among the different servers (partitions) of the system. It should be noted that after the Initial Step of all scenarios presented the communication cost is 0, as there are no avatars and therefore no type of communication in the system.

4.1 Scenario 1

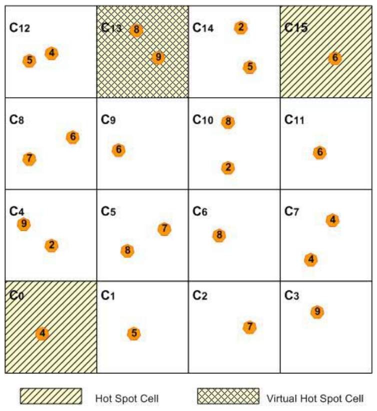
We consider a small-scale virtual environment with dimensions 4x4. The objects of the virtual environment are marked with the cycles, while the value of these cycles represents the DoI value for each object. The stars represent the participating avatars in the virtual world. We consider that the average workload for an object and the avatars is 10, the average diameter D, which defines the AoI of an avatars is 1 and the number of servers available equals to 3. The results of the algorithm for this scenario are presented in Figure 1.

4.2 Scenario 2

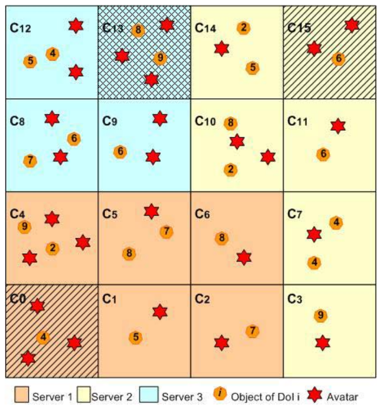
The second application of the algorithm (scenario 2) presents the case where the available hot spots and virtual hot spots share common neighbors. In this case, we calculate the total number of neighboring cells for each hot spot and we give priority (when assigning the neighbors) to the hot spot with the smaller number of neighbors (so as to ensure that workload will be balanced and all hot spots are assigned with neighbors). The initial and final state of this scenario along with the workload and communication cost results are presented in Figure 2.

4.3 Scenario 3

The last scenario presented in this paper constitutes the most complicated case for the application of the algorithm, since it combines the conflict resolution of both neighboring hot spots and common neighbors. Once again, the rational of scenario 2 is applied and the results of the algorithm are presented in Figure 3.



(a)

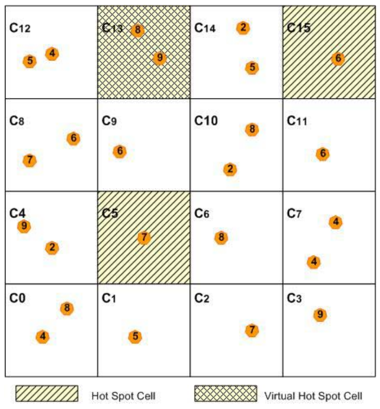


(b)

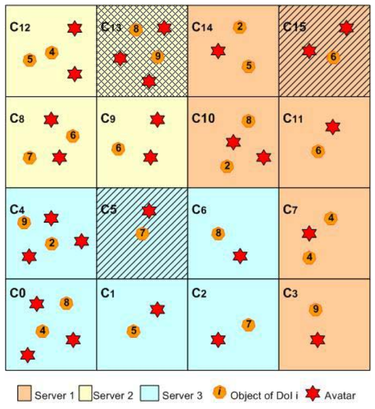
	T_Wload	Id_Wload	Com_Cost
Initial Step	**282**	**94**	
Server 1	100		
Server 2	92		
Server 3	90		
Final Step	**552**	**184**	**87**
Server 1	200		
Server 2	172		
Server 3	180		

(c)

Figure 1: Initial State (a), Final State (b) and Experimental Results (c) for Scenario 1



(a)

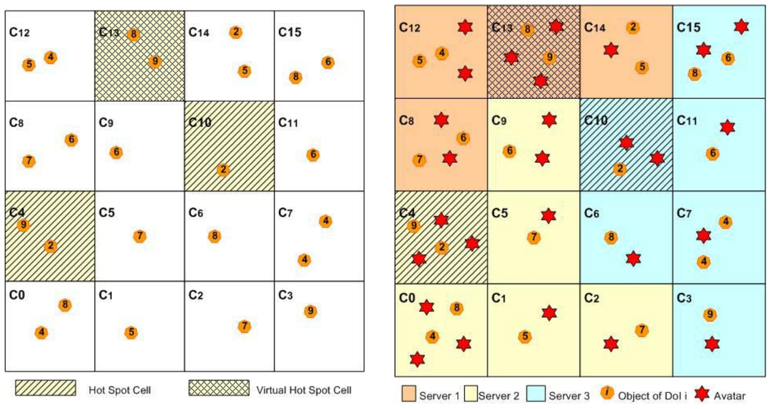


(b)

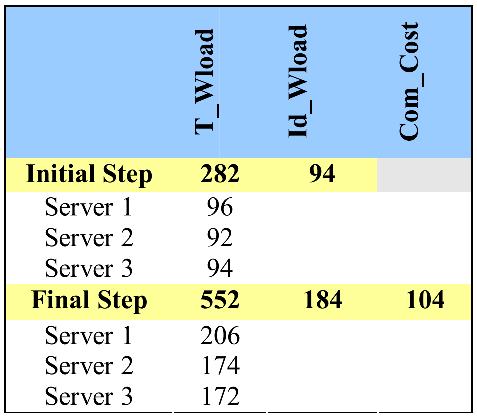
	T_Wload	Id_Wload	Com_Cost
Initial Step	**282**	**94**	**-**
Server 1	100		
Server 2	92		
Server 3	90		
Final Step	**552**	**184**	**86**
Server 1	200		
Server 2	172		
Server 3	180		

(c)

Figure 2: Initial State (a), Final State (b) and Experimental Results (c) for Scenario 2



(a) (b)



	T_Wload	Id_Wload	Com_Cost
Initial Step	**282**	**94**	
Server 1	96		
Server 2	92		
Server 3	94		
Final Step	**552**	**184**	**104**
Server 1	206		
Server 2	174		
Server 3	172		

(c)

Figure 3: Initial State (a), Final State (b) and Experimental Results (c) for Scenario 3

6. Discussion of the Results

As it can be extracted by the results of the algorithm in all three scenarios, this approach achieves a relatively balanced workload among existing servers in all cases examined, both in the initial and the final state. In the initial state of the virtual world the workload of each server is based on the objects it contains. At the final stage of the experiments each cell's workload adds the workload that the participating avatars introduce. Furthermore, avatars presence and the interaction between them and with the objects, introduces communication cost among the servers of the system in the cases that avatars of one server (partition) interact with avatars or objects of another server. In all experiments conducted, the communication cost lies in reasonable values in regard to the size of the virtual environment and the number of entities it consists.

For proving the algorithm's effectiveness we applied to the virtual environment the partitioning approach of Lui & Chan [1], which, as mentioned in [2] achieves good results. The approach of Lui & Chan defines a quality function C_P which is calculated as follows: $C_P = 0.5 \times C_W + 0.5 \times C_L$, where $C_W = \sum_{i=1}^{i=n} \left| C_{Wi} - IdealC_w \right|$ and represents the workload sum of the variations of each partition workload from a perfectly balanced workload distribution, while C_L represents the communication cost among the partitions (servers) of the system. The results of the application of this algorithm to the virtual environment as well as the comparison of the results are presented in Table 1 and Figure 4.

In the table of the results, C_W represents the total workload of the system after the application of the algorithm, calculated as mentioned above, C_L represents the communication cost among all servers of the system and C_P represents the total system cost, calculated by the equation provided above. From the results extracted we note that the even though Lui & Chan approach achieves a perfectly balanced workload, the communication cost of the system is very high, which in turns results to the increment of the total system cost.

	C_W	C_L	C_P
Scenario 1	32	87	59,5
Scenario 2	32	86	59
Scenario 3	44	104	74
Lui & Chan	0	205	100,5

Table 1: Results of the Total System Cost

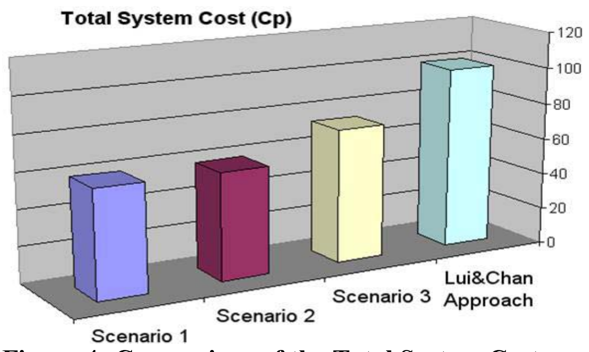


Figure 4: Comparison of the Total System Cost

On the other hand, the results of the approach presented in this paper indicate that the partitioning combines a relatively balanced workload (with low deviation from the ideal) with a significantly lower communication cost in comparison to that of approach [1]. Thus the proposed partitioning technique achieves in all cases examined lower total system cost.

7. Conclusions and Future Work

This paper presented an approach for the partitioning of a DVE. The partitioning takes place when users are not yet connected to the system and is based on objects' attributes. In particular, the presented algorithm takes into account the "importance" of the objects and based on this makes a prediction on avatars' behavior when they will enter the virtual space. It should be mentioned that the main objective of this algorithm is to create partitions in a way that any rebalancing needed, can take place in larger time intervals, so as to minimize often changes on the servers' state. In addition, even though the presented algorithm cannot be applied (in its current version) to persistent virtual environments, however, with minor modifications, could be applied for the further partitioning of overloaded areas (or cells) within a persistent environment. From the experiments conducted, which present the results both of the initial assignment and the results when avatars join the virtual environment, as well as from the comparison of the results by those achieved by the approach presented in [1], it can be extracted that the algorithm achieves a relatively balanced workload distribution among the servers of the system as well as reasonable communication cost.

Some of our planned next steps the application of the algorithm in large-scale virtual environments as well as the upgrade of the algorithm for use in persistent virtual environments.

10. References

[1] John C.S. Lui, M.F. Chan: An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems, IEEE Transactions on Parallel and Distributed Systems, Volume 13, No. 1, (Jan 2002).

[2] P. Morillo, J.M. Orduna, J. Duato: On the Characterization of Distributed Virtual Environment Systems, Proceedings of European Conference on Parallel Processing Euro-Par'2003, Klagenfurt, Austria, (August 2003).

[3] Michael R. Macedonia, Michael J. Zyda, David R. Pratt, Donald P. Brutzman, Paul T. Bar-ham: Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments, VRAIS, p. 2, Virtual Reality Annual International Symposium (VRAIS'95), (1995).

[4] P. Morillo, M. Fernandez, J.M. Orduna: An Evolutive Approach to the Partitioning Problem in Distributed Virtual Environment Systems, XIV Jornadas de Paralelismo, p.p. 299-304, Madrid, Spain, (September 2003).

[5] P.T.Tam: Communication Cost Optimization and Analysis in Distributed Virtual Environment, M. Phil second term paper, Technical report RM1026-TR98-0412, Department of Computer Science and Engineering. The Chinese University of Hong Kong, (1998).