Journal of Network and Computer Applications 34 (2011) 89–101

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A dynamic management scheme for DVEs

Christos Bouras [a,b,*], Eri Giannaka [a,c], Thrasyvoulos Tsiatsos [d]

[a] Computer Engineering and Informatics Dept., Univ. of Patras, GR-26500 Rion, Patras, Greece
[b] Research Academic Computer Technology Institute, N. Kazantzaki Str., Patras University, GR-26500 Rion, Patras, Greece
[c] Athens Information Technology, 0.8 km Markopoulou Ave., 19002 Paiania, Attika, Greece
[d] Department of Informatics, Aristotle University of Thessaloniki, PO Box 114, GR-54124, Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 9 February 2010
Received in revised form
7 August 2010
Accepted 25 August 2010

Keywords:
Distributed virtual environments
Dynamic resource management
Load balancing
Networked servers' architecture
Communication cost
VR techniques and systems

ABSTRACT

Advances in computer technology and networking infrastructures in combination with advanced applications and services, have expanded the adoption of distributed virtual environments and promoted their use in a wide range of areas, such as learning and training, collaborative work, military applications and multiplayer online games. The characteristics and requirements of such DVEs differ significantly given the diverse objectives, scope and context that each virtual world aims at supporting. However, one common characteristic of DVEs is their dynamic state with users entering and leaving the system randomly, resulting in changes of the requirements for the DVE system. These changes require effective load distribution and management of the communication cost so that consistency is always maintained. This paper presents a dynamic management approach for DVEs driven by the diversity of different applications' characteristics and requirements. This approach exploits the dynamic nature of these systems for selecting and assigning, on an on-demand basis, the resources necessary for the efficient operation of the system. The experiments conducted to validate the behavior of the approach illustrate that it can significantly minimize communication cost among the system servers together with effective workload distribution.

1. Introduction

Distributed virtual environments (DVEs) simulate real or imaginary worlds by incorporating rich media and graphics. DVEs became more popular during the last decade, which is likely attributable to the wide expansion of high-speed internet access providing the basic support medium for these systems, as well as to the significant advances of both hardware and software. A large number of platforms and applications were designed and developed for supporting large-scale DVEs, which were gradually adopted in a wide range of both academic and industrial environments. However, the large number of users these systems aim to support in combination with the need for rich graphics and a high level of realism raise a constant trade-off between system performance and fault tolerance. The decision on the techniques and approaches used to deal with this trade-off is usually related to the objectives, scope and context that each virtual world aims at supporting, along with its special characteristics. In particular, the requirements may vary significantly among virtual worlds with diverse simulated scenarios. For example, in the case of an educational DVE, the consistency of the world would not be significantly affected if a number of position messages (i.e., messages sent each time a user changes his/her position) were lost. However, if position messages were lost while in a virtual battlefield, where soldiers move and run, then the sense of realism, users' awareness and performance would be significantly impacted. One common characteristic of DVEs is their dynamically changing state with users entering, navigating, interacting and leaving the system randomly (at will), resulting in continuously changing utilization of resources for the DVE system. These changes, in turn, call for effective load distribution and management of the inter-server communication cost so that consistency is always maintained and extended scalability is supported.

Research has focused on algorithms and techniques for load distribution as well as resource and communication management to improve the performance of these highly demanding systems. Recent research indicates that one of the main issues of networked servers DVEs is scalability. Morillo et al. (2005) presented that DVE systems reach saturation when any of the available servers reach 100% of CPU utilization which dramatically decreases overall system performance, while severely damaging awareness. On this basis, algorithms and techniques for performance optimization and scalability should focus on

* Corresponding author at: Research Academic Computer Technology Institute, N.Kazantzaki Str. Patras University, GR-26500 Rion, Patras, Greece.
Tel.: +30 2610 960375, +30 2610 996951; fax: +30 2610 969016.
E-mail address: bouras@cti.gr (C. Bouras).

1084-8045/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2010.08.013

making the system more resistant to continuous changing states over time. Based on that, it could be stated that for a system with a certain number of servers, the goal is to identify the optimal assignment of resources to serve as many users as possible, while minimizing the communication cost at the same time. This needs to be achieved with guaranteed efficiency based on each application's special requirements. To address this issue, this paper presents an approach for dynamic resource management and load distribution of networked servers DVEs, with the aim to extend their scalability and improve overall performance. More specifically, it proposes a method for optimizing the management of these environments by using the servers of the system on an on-demand basis, to limit the number of reassignments needed and to minimize unnecessary communication cost among the servers in order to reduce the effect of network latency. This dynamic exploitation of system's servers and resources constitutes the main novel contribution of this work. It is therefore advancing the state-of-the-art that is mainly addressing exploitation of all available system servers at any given time, without consideration of the actual and dynamically changing requirements of the virtual world. The behavior of the proposed dynamic management approach is evaluated through a series of experiments for different settings of the virtual world. The results of the experiments clearly show that the major contribution of the dynamic management scheme is the significant reduction of the inter-server communication cost. Given the fact that DVEs depend strongly on the underlying network characteristics, the reduction of the messages exchanged among the system's servers is of increased value for the viability, scalability and performance of the system. Furthermore, the dynamic management approach achieved balanced workload among the system's servers even in highly demanding cases, without reaching the saturation point of 100% of CPU utilization throughout the duration of the experiments.

The rest of the paper is structured as follows: Section 2 outlines some of the related work in the area of algorithms and techniques for load distribution and balancing in DVE systems, while Section 3 presents the dynamic management approach in terms of its main concepts, principles and the parameters measured. Section 4 presents the experiments conducted for evaluating the behavior and efficiency of the approach under different settings of the DVE system. Finally, Section 5 provides conclusions of the paper.

2. Related work

For handling DVEs and facing the scalability issue, existing approaches fall usually into one of the following architectures: (a) networked servers architectures, (b) peer-to-peer architectures and (c) server cluster architectures. Out of the three approaches, the server cluster can provide better latency guarantees, but remains the most expensive and it can also become a single point of failure (Chertov and Fahmy, 2006). Use of peer-to-peer approaches has increased interest in recent years. Even though peer-to-peer architectures seem effective on handling the scalability issue (Morillo et al., 2007), both latency and awareness issues still remain unresolved (Chertov and Fahmy, 2006). Server and network architectures developed to reduce the effect of network latency for DVEs approaches could be classified as (a) latency-driven distribution (LDD) and (b) resource-driven distribution (RDD) (Ta et al., 2006). Specifically, LDD approaches focus on the distribution of a DVE over the networking architecture, while the focus of the RDD approaches is on load distribution. The dynamic management approach presented in this paper refers to distributed networked servers architectures and is similar to the RDD concept. The below paragraphs summarize representative work and approaches relevant in this area.

The load distribution problem, which is related to the effective assignment of world entities, such as clients, cells and regions, to the servers of the DVE system has drawn increased research interest and a number of algorithms and methods have been proposed. In particular, in the LOT technique (Lui and Chan, 2002) authors have showed the key role of finding a good assignment of clients to servers and have proposed a three-step partitioning method for load balancing and communication refinement. Other approaches explored the use of micro-cells (Duong and Zhou, 2003; De Vleeschauwer et al., 2005) for load distribution in large-scale DVEs. Duong and Zhou (2003) used micro-cells to reassign servers in a cluster if the load on the server they are currently residing on becomes too large, while De Vleeschauwer et al. (2005) proposed the dynamic assignment of microcells to a set of servers to redistribute the load. However, the frequent remapping required with these methods could lead to high overhead for servers. Therefore, a locality aware load balancing method was proposed which reduces cross-server communication (Chen et al., 2005). However, even though it considers awareness of spatial locality in a virtual space, the method also leads to frequent region migrations. A new mechanism for sharing roles and separating service regions (SRSS) was proposed (Jang and Yoo, 2008), which reduced unnecessary partitions of short duration. Furthermore, an adaptive strategy that takes into account the non-linear behavior of DVE servers has been presented (Beatrice et al., 2002) with local scope for the load balancing technique. However, this strategy provided good performance only for uniform movement patterns of users. Following this work, Morillo et al. (2003) proposed an adaptive load balancing technique of global scope, which avoided DVE saturation as long as possible, regardless of both the movement patterns of users and the initial distribution of users in the virtual world. Other approaches adopt the use of thresholds for defining the number of clients a server is willing to serve (Chertov and Fahmy, 2006). In particular, each server uses a client threshold value to determine the number of clients it is willing to serve. If the client threshold is exceeded, the server attempts to migrate part of the load to a nearby server. Also, mirrored architectures have been proposed, which replicate DVE zones at multiple servers (Ta et al., 2006).

Regarding commercial DVE systems, the World of Warcraft (WoW) (World of Warcraft Architecture) adopts a distributed architectural model, where the world servers (or realms) constitute complete, self-contained copies of the game world. The realms are distributed among different parts of the world, but a world is a collection of servers, not a single server. When a user logs through his/her account, an authentication server verifies the data and transfers the user to the realm on which he/she last played. As far as it concerns Second Life (Kumar et al., 2008), users run a client program that connects to a central server, which employs four main clusters of machines, namely a central database, a logging database, an inventory database and a search database. The system uses visibility computation to determine the relevant subset of data to send to each client. Servers store all objects and perform the key actions to evolve the world while maintaining reasonable consistency. Second Life divides the world into 256×256 m^2 tiles, each of which is statically bound to a particular server that executes on a single CPU core. To minimize communication, Second Life partitions objects in the virtual world among the servers so that at any given time, only one server maintains an object's state.

Research in the direction of load distribution and resource management for DVEs has been very active with numerous approaches presented and adopted. The majority of these