Server Placement for Enhancing the Interactivity of Large-Scale Distributed Virtual Environments

Duong N. B. Ta* Suiping Zhou
School of Computer Engineering
Nanyang Technological University, Singapore 639798
{pa0236892b, asspzhou}@ntu.edu.sg

Abstract

Distributed Virtual Environments (DVEs) allow many simultaneous human users to interact with each other in shared, 3D virtual worlds. The interactivity of a DVE is of crucial importance, as its success greatly depends on users' perceptions when interacting within the virtual world. In this paper, different from traditional application-centric approaches like dead reckoning, we propose a network-centric approach to enhance the interactivity of DVEs by directly reducing the network latencies in client-server communications. We consider a key problem with this approach, termed the server placement problem. Generally, this problem concerns how to place servers in the network to reduce client-server communication latencies. We then suggest several degree-based server placement approaches. Extensive simulation studies using realistic models have shown that appropriate server placement is very effective in enhancing the interactivity of large-scale DVEs.

Keywords: *Server placement, interactivity, distributed virtual environments.*

1. Introduction

In recent years, advances in networking technologies, computer graphic and CPU power have popularized the deployments of Distributed Virtual Environments (DVEs). Large-scale DVEs refer to a class of applications that enable thousands of geographically distributed users[1] to simultaneously interact with each other in a shared, computer-generated 3D virtual world, where each client is represented by an *avatar* [22]. A client controls the behavior of his/her avatar by various *inputs*, and the *updates* of an avatar's state need to be sent to other clients in the same *zone* of the virtual world to support the interactions among clients. Prominent applications of DVEs include online games, military simulations, collaborative designs, virtual shopping mall, etc. The latest study by the market research company IDC (http://www.idc.com) on the online gaming market of Asia/Pacific (excluding Japan) revealed that the subscription revenue is about US$1.09 billion in 2004, an increase at about 30% compared to 2003. Also according to this study, this market will be more than doubled in 2009.

Enhancing the interactivity of DVEs is of crucial importance, since the users may not feel that they are *interacting* with other elements in the virtual world if the responses from the DVE are much slower than what the users experience in real-life. However, in general, seeking a good balance between enhancing interactivity and other important issues in DVEs such as maintaining consistency, managing resource bottlenecks and securing the virtual world system under various technical problems such as large Internet latency, high resource demands and high potential of security threats is a very challenging task. In this paper, we propose a *network-centric* approach to enhancing the interactivity of large-scale DVEs. Our methodology is different from traditional *application-centric* approaches, e.g., dead reckoning, which aims to "hide" the effect of network latencies by predicting the state of remote users at the application level. Since the activities of human users in DVEs are highly unpredictable, such approach may result in visual perception errors, or in more serious cases, inconsistent DVE states [26].

Our network-centric approach is based on a geographically distributed server architecture (GDSA)[8, 15], in which multiple geographically distributed servers are connected to each other. Each client is connected to one of these servers, and clients interact with each other through these servers. This server-based architecture essentially provides good ways to implement consistency control, resource management and security mechanisms. In addition, in order to deal with large-scale DVEs with hundreds, or even thousands of clients interacting simultaneously, usually the vir-

*Contact author.
[1] In this paper, the terms "user", "participant" and "client" all refer to DVE users and are used interchangeably.

tual world is spatially partitioned into many distinct *zones*, with each zone managed by only one server, as in [3]. A client only interacts with other clients in the same zone,[2] and may move to other zones. As a server only needs to handle one or more zones instead of the entire virtual world, the system is more scalable. In this paper, we refer to such a partitioning approach as the *zone-based approach*.

In our network-centric approach, we consider a key networking problem in enhancing the interactivity of DVEs: the *server placement problem*. This problem aims to reduce the client-server communication delays by finding good network locations to place the servers, and it is generally NP-hard. To address this problem, we have suggested several server placement mechanisms. Extensive simulation studies have shown that appropriate degree-based server placement is very effective in enhancing the interactivity of large-scale DVEs. To our knowledge, our work is the first to consider degree-based server placement to enhance the interactivity of DVEs.

The rest of the paper is organized as follows. Section 2 introduces the server placement problem in DVEs, and several server placement approaches are discussed in Section 3. Next, our evaluation methodology is presented in Section 4. Simulation study is described in Section 5, and Section 6 concludes the paper.

2. Server placement problem

A key question in the GDSA is where to place the servers to reduce the communication delay between the clients and the servers. This problem has been studied extensively in the context of Web replica placement [10, 18, 21]. However, to our knowledge there's no existing work that assesses the suitability of Web replica placement approaches to the server placement problem in DVEs. In this paper, we look at the state-of-the-art results in the Web replica placement problem. We then discuss the difference between the server placement problem in DVEs and the Web replica placement problem. Finally, we suggest some possible approaches adapted from Web replica placement strategies that we believe appropriate to the server placement problem in DVEs.

2.1 Web replica placement problem

The explosive growth of the World Wide Web in the late 1990's had demanded efficient content delivery architectures to offer better service to web users at lower costs. A Content Distribution Network (CDN) is a system of many (hundreds or thousands) well-connected servers deployed over the Internet to transparently deliver web contents to end users by allocating the replicas of contents at the edges of the Internet, i.e., close to the users. Hence, in large CDNs like Akamai [1], the Web replica placement is crucial to the system performance.

Informally stated, the Web replica placement problem concerns how to choose K web replicas among N potential sites (i.e., network locations) in the network, where $K < N$, to minimize a given objective function, usually the clients's access latencies, or the total bandwidth consumption. After the replicas are in place, each web client connects to its closest replica to retrieve the contents needed. Here, it is assumed that each client needs to access only one replica to get all of its interested contents. The Web replica placement is usually formulated as a minimum K-median problem, which is a well-known NP-hard problem. In [21], the authors proposed a greedy placement algorithm that is shown to be able to perform very well compared to a "super-optimal" algorithm[3] that uses Lagrangian relaxation with subgradient optimization. The authors showed in [21] that the median performance in terms of client-server communication delays of the greedy algorithm is within a factor of 1.1-1.5 of the optimal.

2.2 Server placement problem in DVEs

Although the Web replica placement problem has been well-studied in the context of CDNs with promising results, the appropriateness of the proposed approaches to the server placement problem in DVEs has not yet been studied. The main difference between the Web replica placement problem and the server placement problem in DVEs lies in the "connection" of clients to servers. In the first problem, it's relatively straightforward to let web clients connect to their closest replicas, hence the greedy placement approach is feasible. However, in DVEs, the large virtual world is partitioned into multiple distinct zones, and each zone is hosted by a separate server. A client cannot simply connect to its closest server since that server may *not* manage its zone. Hence, the greedy placement approach is not appropriate for the server placement problem in DVEs, as it assumes that clients always connect to their closest servers.

So, for the server placement problem in DVEs, we need an approach that can place the servers in appropriate network locations to reduce clients' communication latencies without knowing *in advance* which server the clients will connect to. In [10, 18], the authors have shown that by placing Web replicas at key network locations, i.e., at the network nodes (routers or Autonomous Systems[4] (ASes)) with high node degrees, the clients' communication latencies can

[2] For simplicity, we say that a client is in a zone if its avatar is currently residing in that zone.

[3] The "super-optimal" solution may be better than the optimal, but may not be feasible.

[4] An Autonomous System (AS) in the Internet is a collection of IP networks and routers under the control of one (or sometimes more) entity that presents a common routing policy to the Internet (Wikipedia).

be significantly reduced, regardless of the clients's locations in the network. We feel that such approaches are applicable to the server placement problem in DVEs. However, the mechanisms in [10, 18] were only evaluated under the assumption that clients would eventually connect to their closest servers, although this assumption was not used in deciding server placements. Hence, in this paper, we adapt the placement approaches in [10, 18] to address the server placement problem in DVEs, and evaluate the appropriateness of these approaches in DVEs. To our knowledge, our work is the first to consider server placements to enhance the interactivity of DVEs. In the following section, we suggest several possible server placement approaches.

3. Server placement approaches for DVEs

3.1. Random placement

The random placement serves as a basic approach for comparison in the sense that any good server placement approach should perform better than the random placement. In this approach, servers are randomly placed in the network.

Remark 3.1. *The computational cost of the random placement approach is $O(Mm)$, where M is the total number of routers and m is the number of servers.*

3.2. Degree-based placement

The degree-based server placement approaches aim to place the servers at some key locations in the network. These key locations are usually the network nodes (routers or ASes) with high node degrees. The degree-based approaches have been shown to perform comparably to the greedy server placement algorithm in [21] in the context of the Web replica placement problem [18]. However, whether they are appropriate for DVEs needs to be further investigated.

3.2.1 Max-router

In the Max-router placement approach, we sort all the routers in the network in descending order of their node degrees using Quicksort, and place servers at the routers with highest node degrees. Intuitively, a node with larger node degree is likely to be closer to other nodes, hence it is a good choice for server placement. For example in Fig. 1, router *B* has the largest node degree (which is 5). If a server is placed there, then the network distances (in number of hops[5]) from client c_1 and c_2 to the server are 2 and 1, respectively. However, if the server is placed at router A, which has a node degree of 1, then the above client-server distances become 3 and 2, respectively. Thus, placing servers at nodes with large node degree may effectively reduce the client-server communication delay, which implies good interactivity.

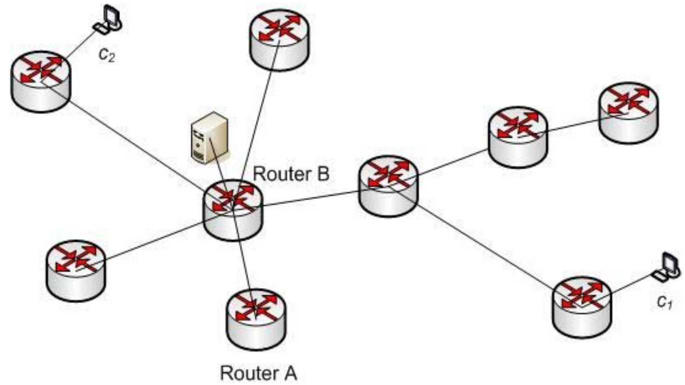


Figure 1. Max-router placement approach

Remark 3.2. *The computational cost of the Max-router placement approach is $O(M \log M) + O(m)$, or $O(M \log M)$, where M is the total number of routers in the network topology and m is the number of servers, $m \ll M$.*

3.2.2 Max-AS/Max-router

The Max-AS/Max-router placement approach is similar to the Max-router approach in the sense that they both aim to place servers at nodes with large degree in the network topology. The main difference between the two approaches is that Max-AS/Max-router uses a hierarchical placement approach which is based on the structural nature of real network topologies.

First, Max-AS/Max-router approach selects the ASes with largest AS-level node degrees. In each AS, we place one server at the router with the largest node degree. By spreading servers over multiple ASes, this approach may further reduce network latencies for geographically distributed clients compared to the Max-router approach.

To implement the Max-AS/Max-router approach, we sort all the routers according to the node degrees of their corresponding ASes (each router belongs to an AS). If there are two routers with the same AS-level node degree, we consider their router-level node degrees. We then iterate over the sorted list of all routers to select one best router in each AS to place the server.

Remark 3.3. *The computational cost of the Max-AS/Max-router placement approach is $O(M \log M) + O(M)$, or $O(M \log M)$, where M is the total number of routers.*

[5] Indeed, network hop counts are good indications of round-trip network delays [17].

3.3. Other degree-based placement approaches

As comparative references, we also consider some alternative server placement approaches, namely Min-router, Max-AS/Min-router and Min-AS/Max-router. The Min-router approach may serve as an "upper bound" on how bad a server placement approach can be. The Max-AS/Min-router and Min-AS/Max-router may help us to identify which one (AS or router) is the major factor in the hierarchical server placement approach. Another reason for studying these approaches is that in practice, sometimes we may have to adopt these methods for server placement since typically the routers with high node degrees may be very busy due to the high volumes of network traffic, thus it may be impossible to add any more service to them [18]. Moreover, some economic and administrative constraints may make it harder to place servers at the selected network locations.

3.3.1 Min-router

In this approach, in contrast to the Max-router placement, the routers with minimum node degrees are selected to place the servers.

3.3.2 Max-AS/Min-router

This approach is similar to Max-AS/Max-router, but in each AS, the router with minimum node degree is selected to place the server for that AS.

3.3.3 Min-AS/Max-router

This approach is similar to Max-AS/Max-router, but we select the ASes with minimum node degrees, and in each selected AS we select the router with maximum node degree to place the server.

4. Evaluation methodology

In this section, we present our methodology in evaluating the effectiveness of proposed server placement approaches. In the evaluation, we strive to be as realistic as possible, by using a wide range of network topologies, including both real and synthetic ones. We model the network latencies based on existing real Internet measurements and studies [17, 24]. In addition, for the workloads, based on recent studies of real online game systems, e.g., [9], we simulate various client distributions (clustered, uniform, etc.) both in the physical world (the network) and the virtual world.

Table 1. Network topologies

Topology	Nodes	Links
Flat, Waxman (F-W)	3000	6000
Flat, Barabasi-Albert (F-BA)	3000	5997
Flat, real (F-R)	3024	5192
Hierarchical, Barabasi-Albert/Waxman (H-BA/W)	3000	6197
Hierarchical, real/real (H-R/R)	3300	13442

4.1. Network models

In our simulations, we used both synthetic topologies generated by the popular topology generator BRITE [2] and real Internet topologies. Table 1 lists the topologies used in this paper. We use BRITE to generate both flat and hierarchical topologies based on the well-known Waxman and Barabasi-Albert model (F-W, F-BA, H-BA/W). The Waxman model [25] considers all pairs of nodes, and then decides whether to add a link between any two nodes with a probability that depends on the distance between these two nodes and the longest distance between any two nodes in the network. The Barabasi-Albert model [7] generates network topologies that exhibit power-laws as observed in the seminal paper by Faloutsos et al. [11] in 1999.

To complement the synthetic topologies generated by BRITE, we use a real, flat Internet topology collected from NLANR [4]. For diversity, we also collect a real AS-level topology (generated by processing the Border Gateway Protocol (BGP) routing tables) with 110 nodes from [5], and use the DFN (German Research Network) topology [12] with 30 nodes as the router-level topology to construct a realistic hierarchical topology (H-R/R). While these topologies are not complete, they at least partially reflect the "true" topology of the Internet, which may have great impacts on the simulation results.

For simplicity, we assume that the round-trip network delay between any two nodes in the network topology is proportional to the number of link-hops between them. This assumption is similar to the one used in most of the previous work [18, 21]. In fact, a recent Internet measurement [17] also showed that round-trip delay is well-correlated with network hop counts. Moreover, to obtain more realistic simulations results, we use both shortest-path routing and AS-level hierarchical routing [24] where possible to calculate the network delays. The AS-level hierarchical routing is a more realistic routing strategy for our simulations, since to some extent it may reflect the "true" routing practice in the current Internet.

4.2. Workload models

4.2.1 Resource usage

Resource usage can be measured by CPU usage, network bandwidth usage, etc. Since the network bandwidth often represents the major operating cost in current server-based online games [14], in this paper, we assume that the server CPU is not a bottleneck, and measure the resource consumption by the network bandwidth usage.

It is well-known that the bandwidth requirement in client-server based DVEs increases quadratically with the total number of clients that are interacting with each other [20]. Thus, we can estimate in advance the bandwidth requirement of each client in a zone based on the number of clients in that zone, as in [20].

4.2.2 Client distributions

Based on existing studies of real online game systems, e.g., [9], we simulated different client distributions (clustered, uniform, etc.) both in the physical and the virtual world. Table 2 shows the combination of different virtual world (VW) and physical world (PW) distributions.

Table 2. Distribution types

Type	0	1	2	3
Clusters in PW	No	Yes	No	Yes
Clusters in VW	No	No	Yes	Yes

Different DVE configurations are used for performance evaluation. A specific DVE configuration is determined by the number of servers, the number of zones, the number of clients, and the total resource capacity of the system. We use the notation *number of servers-number of zones-number of clients-capacity* to denote a DVE configuration. For example, the notation 20s-400z-5000c-2500cp means that the DVE has 20 servers, 400 zones, 5000 clients and 2500Mpbs server bandwidth in total.

4.2.3 Physical world-virtual world correlation

In general, we should note that clients gathering in the same zone of a DVE may not necessarily close to each other in terms of their physical locations. On the other hand, it is natural to observe that clients that are close to each other in their physical locations (e.g., from the same country or the same geographic region) tend to gather in a specific zone of the virtual world due to their common cultural preferences. These phenomena may have great impacts on the performance of the proposed placement approaches.

To model the correlation between clients' locations in the physical world and those in the virtual world, we use a correlation parameter δ, where $0 \leq \delta \leq 1$ [16]. The higher the value of δ is, the stronger the tendency for clients from the close geographic locations to gather in specific zones of the virtual world.

4.3 Performance measure

For interactive applications like DVEs, the client-server communication delay is the most important *Quality of Service* (QoS) parameter that the system provides to clients [13]. In this paper, we say that a client is *with QoS* or *without QoS* if the communication delay between the client and its server is smaller or larger than a given bound, respectively. For different types of DVEs, there are different delay bound requirements. For example, First Person Shooter (FPS) games typically require a delay bound of $250ms$ [13], while car-racing games have much more stringent latency requirement, at about $100ms$ [19]. It is noted that in DVEs, the communication delay is the sum of the network delay and the processing delay at the server. However, in this paper, we assume that the server CPU is not a bottleneck, thus the client-server communication delay is determined by the client-server network delay.

The main performance measure used in the analysis is the percentage of clients with QoS in the system, denoted as $pQoS$. Results presented here are obtained by averaging the results of 50 simulation runs.

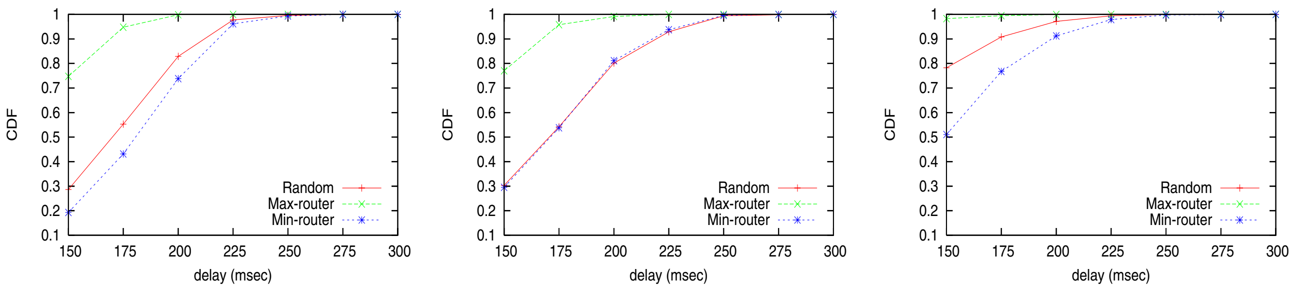
4.4 Default settings

Unless otherwise stated, the following assumptions and default values are used in the simulations. The clients are uniformly distributed in the physical world as well as in the virtual world. To estimate bandwidth requirement [20], the input sending frequency of each client (frame rate) is set to 25 messages per second, and the size of each input or update is 100 bytes, which are close to real settings [6]. The maximum round-trip delay between any two nodes in the network topology is set to $300ms$. The interactivity requirement, i.e., the DVE delay bound, is set to $150ms$. The default DVE configuration is 20s-400z-5000c-2500cp.

We have conducted many sets of experiments, and found similar results. Due to space limitation, in this paper, only the most typical results are shown.

5. Results

Fig. 2 and Fig. 3 show the cumulative distribution function (CDF) of client-server round-trip communication delays under several server placement approaches and network topologies. For flat topologies (Fig. 2), we observe that the Max-router placement performs very well: in terms of $pQoS$, the performance improvements are 157% (F-W



(a) F-W topology (b) F-BA topology (c) F-R topology

Figure 2. Impacts of server placement - CDF with flat topologies

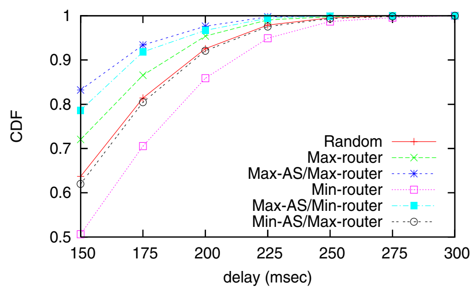


Figure 3. Impacts of server placement - CDF with the hierarchical topology H-BA/W

topology), 168% (F-BA topology) and 26% (F-R topology) compared to the Random placement, and 157%, 275% and 92% compared to the Min-router placement. Moreover, from these figures, it is clear that Max-router not only has a high ratio of clients with QoS, but also provide better interactivity for clients that are without QoS guarantees.

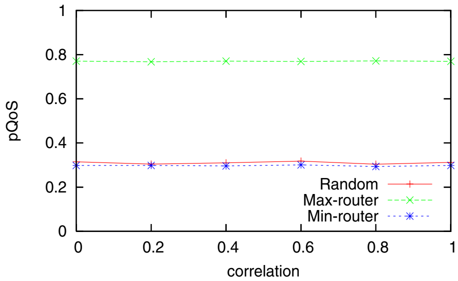
It should be noted that the performance improvement in F-R topology is not as high as those in F-W and F-BA topologies. However, this is not because the Max-router placement performs poorly in F-R topology. As observed in Fig. 2(c), the Random placement also performs quite well in F-R topology. One possible explanation for this result is that the F-R topology, which is a real-world AS-level topology collected from [4], has good connectivity, i.e. any node can reach any other node with a small number of hop counts. Indeed, the average round-trip delay between any two nodes in F-R is much smaller than those in F-W and F-BA, about 30% and 57%, respectively.

For hierarchical topologies (a typical result is shown in Fig. 3), we obtain similar results to the case of flat topologies, except that the Max-AS/Max-router placement performs the best: its performance improvements in terms of $pQoS$ over the Random placement are 38% (H-BA/W) and 48% (H-R/R). This can be explained by the fact that Max-AS/Max-router spreads the servers across multiple ASes, while Max-router may place several servers in the same AS. In addition, it is observed that Max-AS/Min-router and Min-AS/Max-router both perform worse than Max-AS/Max-router, while Min-AS/Max-router is the worst among the three. This observation suggests that the AS-level node degree seems to be the major factor that affects the performance. However, we should also note that only selecting the ASes with largest AS-level degrees to place the servers is not enough: the router-level node degree within an AS is also an important factor.

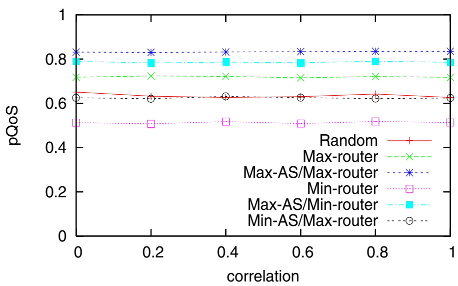
It is noted that our results here are similar to the observations in [18], although [18] addresses the Web replica placement problem, which is very different from the server placement problem in DVEs. Recall that for a typical DVE, which is partitioned into multiple distinct zones, we do not know in advance which server a client will connect to. Thus, we have to place the servers without knowing clients' locations, then randomly assign zones to servers and let each client connect to the server hosting its zone. In contrast, in the context of Web replica placement problem, a client just simply connects to its closest server. Hence, to some extent, the Web replica placement problem may be "easier" to solve than the DVE server placement problem. Nonetheless, it's clear that the degree-based placement approach is universal, i.e., it can be applied into both DVE server placement and Web replica placement with good results.

5.1 Impacts of correlations

Fig. 4 shows the impacts of physical world-virtual world correlation values. Recall that the higher the correlation value is, the stronger the tendency for clients from the close geographic locations to gather in specific zones of the virtual world. However, it seems that this phenomenon does

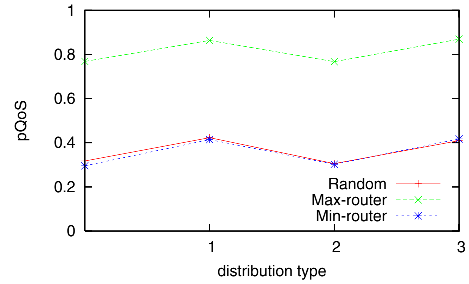


(a) F-BA topology

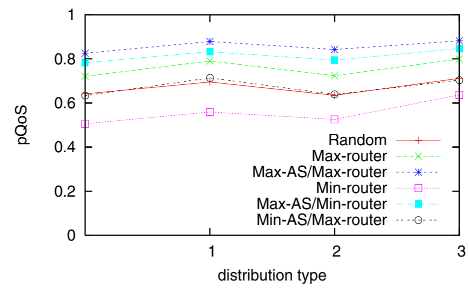


(b) H-BA/W topology

Figure 4. Impacts of virtual world-physical world correlations on server placement approaches



(a) F-BA topology



(b) H-BA/W topology

Figure 5. Impacts of client distributions on server placement approaches - $pQoS$

not have any impact on $pQoS$ of all the server placement approaches.

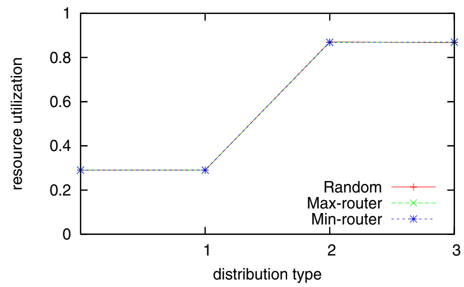
5.2 Impacts of client distributions

In this experiment, we evaluate the impacts of different client distributions in both the physical world and the virtual world on the performance of server placement approaches. Fig. 5 and Fig. 6 show the simulation results. It is observed that client distributions do have some impacts on $pQoS$ and resource utilization. More specifically, the performance in terms of $pQoS$ of all server placement approaches seems to be better when clients are clustered in the physical world (distribution type 1 and 3, Fig. 5). On the other hand, the clustered distribution of clients in the virtual world mainly has effects on the resource utilization (see Fig. 6). We observe that the resource utilization becomes very high when clients are clustered in the virtual world (distribution type 2 and 3). This can be understood, recall that the bandwidth requirement at the server increases quadratically with the number of clients in a zone.

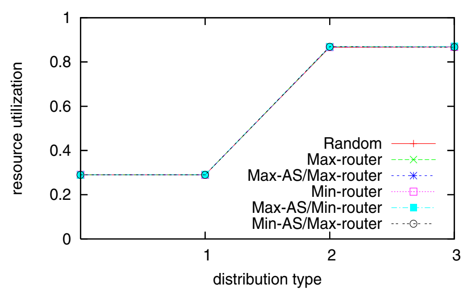
6. Conclusion and future work

In this paper, we have proposed to take a network-centric approach, which is different from the traditional application-centric approach, to enhance the interactivity of large-scale DVEs. In particular, we have considered the server placement problem to directly reduce the communication delays between clients and servers. Several degree-based server placement approaches have been proposed and evaluated via extensive simulations. As far as we know, this is the first work to consider server placement approaches in DVEs. Our main findings are as follows:

- The degree-based server placement approaches such as Max-router and Max-AS/Max-router, which do not require to know the clients' locations in advance, are able to perform very well in terms of interactivity, i.e, $pQoS$, compared to the Random server placement approach. This result indicates that the degree-based server placement methodology is suitable for both the Web replica placement problem [18], and the server placement problem in DVEs.
- Common phenomena in large-scale DVEs like physical world-virtual world correlation, or client clustering do



(a) F-BA topology



(b) H-BA/W topology

Figure 6. Impacts of client distributions on server placement approaches - resource utilization

not seem to have bad effects on the interactivity provided by degree-based server placement approaches. In contrast, the interactivity seems to be better when clients are clustered in the network. This may indicate that degree-based server placement can tolerate a high degree of unexpected workload patterns, which are very likely in dynamic applications like DVEs.

In the future work, we will investigate the possiblity of combining server placement and client assignment mechanisms [23] to further enhance interactivity of large-scale DVEs.

References

[1] Akamai. *http://www.akamai.com*.
[2] BRITE. *http://www.cs.bu.edu/brite*.
[3] Evequest. *http://www.everquest.com*.
[4] NLANR. *http://moat.nlanr.net/Routing/rawdata*.
[5] SSF. *http://www.ssfnet.org/Exchange/gallery/*.
[6] A. Abdelkhalek, A. Bilas, and A. Moshovos. Behavior and Performance of Interactive Multi-player Game Servers. *Special Issue of Cluster Computing: the Journal of Networks, Software Tools and Applications*, 2002.
[7] A. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, pages 509–512, 1999.
[8] D. Bauer, S. Rooney, and P. Scotton. Network infrastructure for massively distributed games. In *Proc. of NetGames*, 2002.
[9] W. chi Feng and W. chang Feng. On the geographic distribution of online game servers and players. In *Proc. of NetGames*, 2003.
[10] E. Cronin, S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt. Constrained mirror placement on the internet. *IEEE Journal on Selected Areas in Communications*, 2002.
[11] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In *ACM SIGCOMM*, 1999.
[12] O. Heckmann, M. Piringer, J. Schmitt, and R. Steinmetz. Generating Realistic ISP-Level Network Topologies. *IEEE Communication Letters*, 2003.
[13] T. Henderson and S. Bhatti. Networked games: a QoS-sensitive application for QoS-insensitive users? In *Proc. of the ACM SIGCOMM*, 2003.
[14] B. P. J. Mulligan. *Developing Online Games: An Insider's Guide*. New Riders Games, 2003.
[15] M. Mauve, S. Fischer, and J. Widmer. A generic proxy system for networked computer games. In *Proc. of NetGames*, 2002.
[16] C. D. Nguyen, F. Safaei, and P. Boustead. Comparison of distributed server architectures in providing immersive audio communication to massively multi-player online games. In *Proc. of ATNAC*, 2004.
[17] K. Obraczka and F. Silva. Network Latency Metrics for Server Proximity. In *In Proc. of the IEEE Globecom 2000*, 2000.
[18] R. G. P. Radoslavov and D. Estrin. Topology-informed internet replica placement. In *Proc. of Web Caching and Content Distribution Workshop*, 2001.
[19] L. Pantel and L. Wolf. On the Impact of Delay on Real-Time Multiplayer Games. In *Proc. of NOSSDAV*, pages 23–29, 2002.
[20] J. Pellegrino and C. Dovrolis. Bandwidth requirement and state consistency in three multiplayer game architectures. In *Proc. of the ACM NetGames*, 2003.
[21] L. Qiu, V. Padmanabhan, and G. Voelker. On the placement of web server replicas. In *Proc. of IEEE INFOCOM*, 2001.
[22] S. Singhal and M. Zyda. *Networked virtual environments: design and implementation*. Addison-Wesley, Reading, MA, 1999.
[23] D. B. N. Ta and S. Zhou. Efficient Client-To-Server Assignments in Distributed Virtual Environments. In *Proc. of IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2006.
[24] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin. The Impact of Routing Policy on Internet Paths. In *Proc. of IEEE INFOCOM*, 2001.
[25] B. M. Waxman. Routing of Multipoint Connections. *IEEE Journal on Selected Areas in Communications*, 9:1617–1622, 1988.
[26] S. Zhou, W. Cai, B. S. Lee, and S. J. Turner. Time-space consistency in large-scale distributed virtual environments. *ACM Transactions on Modeling and Computer Simulation*, 14(1):31–47, 2004.