

# Performance Analysis of Cloud-based CVE Communication Architecture in Comparison with the Traditional Client Server, P2P and Hybrid Models

<sup>1,2</sup>Abdulsalam Ya'u Gital, <sup>1</sup>Abdul Samad Ismail, <sup>3,5</sup>Haruna Chiroma, <sup>4</sup>Adamu I. Abubakar, <sup>2</sup>Bala Ma'aji Abdulhamid, <sup>6</sup>Jaafar Zubairu Maitama <sup>4</sup>Akram Zeki

<sup>1</sup>Department of Computer Science, Universiti Teknologi Malaysia.

<sup>2</sup>Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi Nigeria

<sup>3</sup>Artificial Intelligence Department, University Malaya, Kuala Lumpur, Malaysia.

<sup>4</sup>Department of Information Systems, International Islamic University, Malaysia.

<sup>5</sup>Federal College of Education (Technical), Gombe, Nigeria

<sup>6</sup>Faculty of Computer Science and Information Technology, Bayero University Kano

[lagital@acm.org](mailto:lagital@acm.org), [abdsamad@utm.my](mailto:abdsamad@utm.my), [chiroma@acm.org](mailto:chiroma@acm.org), [100adamu@gmail.com](mailto:100adamu@gmail.com), [akramzeki@yahoo.com](mailto:akramzeki@yahoo.com)

**Abstract—** Gital *et al.* (2014) proposed a cloud based communication architecture for improving efficiency of collaborative virtual environment (CVE) systems in terms of Scalability and Consistency requirements. This paper evaluates the performance of the proposed CVE architecture. The metrics use for the evaluation is response time. We compare the cloud-based architecture to the traditional client server and peer-2-peer (P2P) architecture. The comparison was implemented in the CVE systems. The comparative simulation analysis of the results suggested that the CVE architecture based on cloud computing can significantly improve the performance of the CVE systems.

**Keywords-** Collaborative Virtual Environment, Cloud Computing, Client Server, Peer-to-Peer, Hybrid

## I. INTRODUCTION

In the current types of CVE systems, participants and or collaborators log in and out of the system at any time, and system virtually simulate the activities of users within the virtual world. Thousands of users interact in the shared virtual world. The viable communication architectures used for CVE systems are the traditional client-server, P2P, client-multi-server and/or Hybrid. Detailed description of these architectures can be found in [1]. Client-server architecture is designed based on a central server. All nodes are connected to the single central server, which manages the communication between different nodes and stores data. This type of network architecture enables the server to make contact with all the nodes at the same time. Therefore, when two users want to interact together, all the communications have to pass through the server, increasing latency during interactions. When the number of users increases, a bottleneck can occur on the server as a result of numerous communication requests, resulting in slow communications [2]. Peer-to-peer (P2P) architecture enables high-speed communications between pairs of users because all events are transmitted directly from one user or participant to

another one. Therefore, P2P enables few users to efficiently communicate and have a closely coupled interaction. However, when the number of users is many, the amount of information to be transmitted on the network will saturate the network thereby causing huge delay in transmission. As a result, it is difficult to contact all the nodes at the same time to transmit collaborative environment changes [2]. The P2P architecture used applications such as MR Toolkit [3], SIMNET [4], and NPSNET [5].

Hybrid architecture overcomes the limitations of client server and P2P as it uses both peer-to-peer connections and several servers. This speeds up communication among multiple users and guarantees consistent collaboration. SPLINE [6] uses client-multi-server with several servers sharing information (messages, events, etc.) with peer-to-peer connections between these servers. At the beginning of a session, the session manager connects users to one of these servers, then users only communicate with their assigned server. This solution avoids the bottleneck on the server when the number of users increases and it makes it possible to easily connect nodes with slower connections. Indeed, each server can perform additional processing such as compression or communication with a specific protocol. However, the use of too many servers may increase the system latency and the load of the servers [2]. The client multi-server architecture operated like the hybrid, the only difference is that client multi-server does not necessarily connect the servers peer to peer. The servers are either centralized or distributed in different network. All the servers in either the centralized or distributed multi-server architecture contain the same application. Users connected to each server may be relocated to another server due to server overload. These disadvantages of the traditional architectures necessitate the design of a new CVE architecture based on cloud computing by Gital *et al.* The architecture based on cloud computing provides a good management of both computing and network resources that can efficiently support the scalability and consistency requirements in CVE system.

In view of this, this paper evaluates the performance of the cloud based architecture to determine the efficiency of the architecture in comparison with the client server, peer-to-peer, and hybrid models. The rest of the paper is organized as follows. Section II reviews state of the CVE systems implement based on client server, peer-to-peer and hybrid models. Section III presents the materials and methods for the performance evaluation. The results and discussions are presented in Section IV. Section V concludes the work.

## II. RELATED WORK

### A. Literature Review

This section presented a background review on some CVE systems. The review presents the state of the art collaborative virtual environment systems design based on the traditional communication architectures. Distributed Interactive Virtual Environment (DIVE) was introduced by [7]. DIVE is one of the most acknowledged Virtual Collaborative System, which is a tool kit for building distributed VR application in a heterogeneous network environment [8]. DIVE allows many users and applications to interact in real-time through virtual environment. It is one of the early systems that continue to be developed and improved over the years. DIVE uses multicast protocols for the communication simulating a large shared memory for a process group through the network [7, 9]. DIVE focuses on peer-to-peer multicast communication instead of Client-Server architecture because of interaction time that may introduce lags [7, 10].

Massively Multimedia Online Game (MMOG) is the most practically successful and widely deployed type of CVE or real-time distributed simulation [11]. A MMOG allows players to act together concurrently in the virtual world over the Internet. With the advancements in computer graphics, artificial intelligence, and the availability of high speed networks, all games played over the Internet are growing rapidly [12-14]. These real-time distributed virtual environments are characterized by a large number of concurrent users involved in the same virtual world [15, 16]. The scalability of the system depends on the available bandwidth of servers and clients, types and frequencies of actions, and as well as player density in a given region. Synchronous communication and proper coordination among the parties are important and can be defined through end-to-end delay, therefore, end to end delay should not be more than 200ms in some cases [17]. In MMOG, due to highly reactive actions among the players, the requirement of frequent updates with a reasonable end-to-end delay imposes a firm time constraint. Simulation Network (SIMNET) [18] was designed for a local area network with small numbers of players (less than 50). SIMNET's reliance on broadcasting PDU's over bridged network is still the most common mode of communication for DIS. SIMNET is

a family of large-scale combat-training oriented simulations. The aim of SIMNET systems is to support a large number of participants and only successful demonstrations of around 300 simultaneous players have been reported [19]. Updates are sent over the network using multicast. The update transmission system is limited to information relevant to the training application. However, participants may join the running simulation at any time, everyone has to broadcast full status information in regular intervals to inform the new participant since SIMNET uses P2P model in it design. [18].

A Software toolkit for network based virtual world was proposed by (BrickNet) [20]. BrickNet enables graphical objects to be maintained, managed, and used efficiently, and permits objects to be shared by multiple virtual worlds or clients. A client can connect to a server to request objects of its interest. These objects are deposited by other clients connected to the same server or another server on the network. Depending on the availability and access rights of objects, the server satisfies client requests. The communication part of BrickNet has been implemented using UDP. BrickNet consists of a network of servers that allow clients to connect. Clients cannot change their server, but they can share information across servers. In particular, they can lease out objects to other clients. Clients communicate by messages routed by the servers [19, 21], [22]. Gaming over Content-Oriented Publish/Subscribe System (G-COPSS) was proposed by Chen, et al. [23]. G-COPSS is a modification of content oriented publish/subscribe (COPSS)[24] designed for supporting needs of a multi-user game environment due to increasing action of a particular region in a game. G-COPSS is designed as a decentralized content-centric communication framework to support MMORPG.

The fundamental capability of disseminating information based on content – without the need of knowing who to send it to or who to query for information – makes the content centric communication fabric very suitable for gaming applications. G-COPSS uses the push-based multicast to guarantee on time update delivery. Adopting the content-centric solution defeat many of the limitations of a server-based or a P2P-based solution in terms of scalability, responsiveness etc. G-COPSS try to provide an efficient, distributed communication infrastructure for MMORPG. G-COPSS is a decentralized gaming platform that optimizes gaming environment. It uses a multi-layer hierarchical map functionality to help scene rendering and update dissemination, and provides extra attributes to improve the experience of players moving between regions. The use of multicast protocol and peer to peer depreciates the quality of G-COPSS because it can only scale with limited number of game types and when the number of users exponentially increases, users at all regions will need to filter large volumes of data. This in turn creates delay and affects consistency of the system. The Model and Implementation of a Hybrid P2P Framework for Massive Virtual

Environments (Audrey) [25] is a hybrid P2P architecture. Audrey model specifies a managed server within a P2P framework, placing its design into the class of hybrid P2P systems. This server is in charge of account registration, login, logout, bootstrapping, long-term persistence, and many other tasks. The server maintains the virtual environment activities. It is very much a server with ultimate VE responsibility. The connected peers of Audrey are arranged into a Voronoi-based overlay network [26, 27]. Open Cobalt [28] is a platform designed based on peer to peer technology. This platform does not require any centralized server to function. It is designed for constructing, accessing, and sharing multi-user virtual worlds, virtual exhibit spaces, and game-based learning and training environments both on local area networks and across the Internet. Open Cobalt uses peer to peer collaborative protocol in order to reduce reliance on server infrastructure used for the support of large number of users interacting within a virtual world, providing 3D virtual world hyperlink function in order to form a large distributed network of connected users in the collaboration spaces. Other are (MASSIVE I, II and III) [29-31], Blue Banana [32] is a modification of SOLIPSIS [33] etc. All CVE systems currently uses either client server, peer-to-peer or combination of both as hybrid. Therefore achieving the scalability and consistency requirement of the current state are the major challenges of the CVE system. The proposed cloud based CVE architecture is describe in the following section.

### B. Overview of Cloud based Model

The cloud based model follow the concept of mass data processing in cloud computing in its design, this is because cloud computing technology provide a cutting age technique for provision of adequate network and computing resources, and storage capabilities to handle all types of applications with mass data processing requirements, and that are delay sensitive and require reliable data transmission. The cloud-based CVE is designed to improve the state of scalability and consistency in CVE systems. The framework for the design of the cloud based CVE system is presented in[34]. The framework of the cloud-based CVE model is layered

framework, it consist of: Infrastructures Layer, Platform Layer and an Application Layer. Details of the cloud-based architecture can be found in[35].

## III. MATERIALS AND METHODS

The simulation was performed to compare cloud based architecture with the traditional architectures used for implementation of CVE systems as discussed earlier. NS2 simulator is used for the simulation of the experimental setup on a machine with the following configurations: Intel (R) Core (TM) i5-2410m processor, 2.30GHz speed, 4.00GB RAM with Ubuntu operating system. The network topology used in this simulation is flex bell topology shown in Fig. 1. The choice of the network topology comes as a result of literature scrutinized in similar area of research. TCP is used as the transport protocol. The topology consists of TCP senders, TCP receivers and a pair of routers. The link between the sender's nodes and routers is termed sender's link and it is connected to different router because the users are formed from different subnet and each subnet is connected to a router A, while the link between the receivers and router B is called the receiver link. The sender and receiver links represent a local area network (LAN). The link between routers R and router A represent the bottleneck link linking users to the cloud. The links between the sender's nodes and the Cloud link are full wired duplex link. The bandwidth of the sender's links is set to 20Mbps with 10ms delay. The bandwidth of the receiver links that represent the cloud is also set to 45Mbps with 10ms delay. The bottleneck link is set to 10Mbps with 20ms delay to represent a connection to cloud infrastructures. The number of sender's node which is equivalent to the number of concurrent collaborators, is set in six different scenarios as follows: 200 with 2 receivers' node, 400 with 4 receivers' node, 600 with 6 receivers' node, 800 with 8 receivers' node, and 1000 with 10 receivers' node respectively. This setting represents a virtual environment with ten partitions each handling 100 users, 100 users is the expected threshold for each server (Receiver node) in the system. Details of the parameters used are shown in Table I.

TABLE I. PARAMETERS FOR THE SIMULATION

Link	Bandwidth	Delay	Queue Limit	Window Size	Packet Size	Traffic Type	Link Type
Link to the Cloud.	10Mbps	20ms	100	8000kb	552B/200B	Telnet/CRB	Full Duplex
Senders Link	20Mbps	10ms	-	-	-	Telnet/CRB	Full Duplex
Link to cloud Infrastructures	450Mbps	10ms	-	-	-	Telnet/CBR	Full Duplex

## IV. RESULTS AND DISCUSSION

The evaluation in this Section tries to determine the suitability of the communication architecture for the design of CVE system. The validation is conducted by comparing

with the traditional architectures in different simulations varying the number of users. The increasing number of user is the major problem of the traditional architectures. The metrics use for the evaluation are throughput and delay.

#### A. Throughput analysis

The throughputs of the cloud based architectural model is evaluated to determine its effectiveness compared to the traditional architectures used for CVE systems. The analysis was conducted using the parameters described in Section 4. The simulations were conducted in five different scenarios each with a different number of users (200, 400, 600, 800, and 1000). The simulation results are shown in Table I, where the average throughputs of each of the architectures in all the scenarios are presented. From the average throughput values in Table I, the cloud-based model gain an average throughput of 1227.74 kbps in the first scenario with 200 users, and P2P, hybrid and client-server gain 932.97, 846.77 and 658.64 kbps respectively. Even though the cloud-based architecture produces more oscillated throughput pattern, it still perform better than the traditional architectures in this scenario.

TABLE II. AVERAGE THROUGHPUT

Number of Users	Cloud-Based	P2P	Hybrid	Client-server with Single Server
200	1227.74	932.97	846.77	658.64
400	1138.53	943.44	940.53	561.38
600	1140.72	968.88	955.06	564.13
800	1144.81	981.16	992.73	472.11
1000	1052.66	702.31	833.64	250.43

In the second, third, fourth and the fifth scenarios, the cloud-base model still gain better throughput with 113.53kbps in the second scenario, 1140.72kbps in the third scenario, 1144.81kbps in the fourth scenario and 1052.66 kbps in the last scenario with 1000 user. It can be seen that the throughput reduces with a factor when the number of user increases from 800-1000. The throughput performance of the cloud-based architecture shows a good improvement compared to the traditional P2P, client-server and hybrid architectures. The result shows that large number of users can be accommodated with any CVE system design with the architecture as compared with other models.

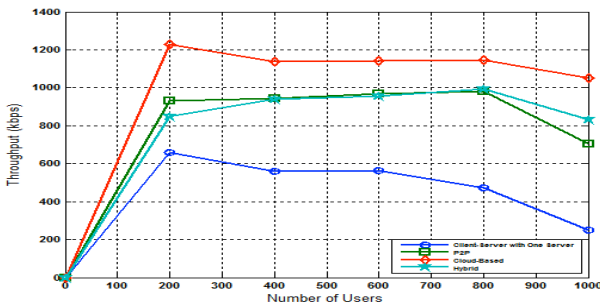


Figure 1. Average Throughput of client-server, P2P, hybrid and cloud-based models

The performance of hybrid model compared to P2P and client-server is better at the highest load in the last scenario. When compare the throughput of the models, it has been clearly proven that the cloud-based model perform better than all the traditional models, next to the cloud-based model is the hybrid model followed by P2P and lastly client server. This result is illustrated in Figure 1.

#### B. Data analysis

In this section, the delay behavior of the cloud-based model is analyzed compared to traditional client-server model, P2P, and hybrid for validation. Five scenarios are run, in each case, the delay incurred by each of the model is recorded and evaluated. Table II shows the average delay of the models. In the first scenario with 100 users, the average delay of the cloud-based model is 56.224ms. Comparing the results reveals the performance of the cloud-based model to be good for collaborative activities, this is because the average delay in the other models in the first scenario is relatively higher than that of cloud-based model with P2P having average delay of 58.612ms, hybrid having 72.11ms and client-server with average delay 181.56ms. The first scenario shows that all the models can maintain good consistency state when the number of users is not large. The little difference is an indication that even with few users, the cloud-based is more efficient.

TABLE III. AVERAGE DELAY OF THE ARCHITECTURE

Number Of Users	Hybrid	Client-server with Single Server	P2P	Cloud-Based
200	238.05	621.37	238.76	201.33
400	244.9	1013.2	243.02	202.56
600	247.86	1035.4	251.45	204.21
800	255.41	1072.31	257.33	206.58
1000	263.75	1093.45	261.42	210.04

In the second scenario, client-server delay increases rapidly, an indication that shows the bottleneck at the server. Hybrid delay also increases almost at the same rate does client-server. P2P shows moderate increase in delay. Cloud-based model at this stage does not show significant increase in the delay. In the fourth to sixth scenarios client-server perform worse than all the models with the highest average delay of 4109.6ms in the final scenario with 1000 users. This is followed by hybrid model with 1655.604ms average delay. P2P delay behavior shows that P2P maintain a certain level of good performance with 162.80ms average delay.

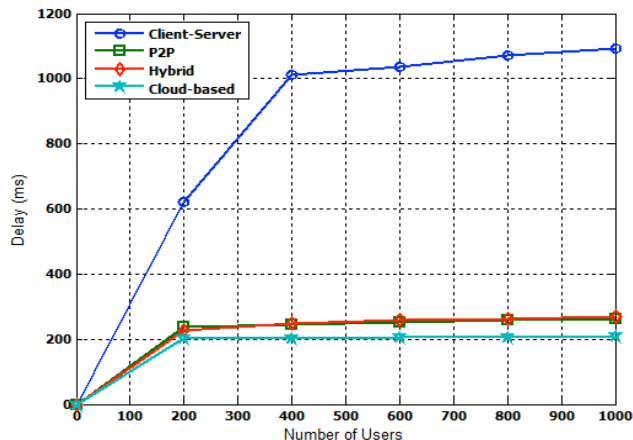


Figure 2. Average Delay of client-server, P2P, hybrid and cloud-based models

The overall results shows that comparing the delay of all the models, the cloud-model out perform all the models with minimum delay of 120.126ms as average delay with the highest member of users considered during the simulation. These results are illustrated in Figure 2. The figure shows that the performance of the cloud-based model is better than that of client-server, hybrid and P2P. P2P also shows average good performance in terms of delay compare to other models. The result of this simulation proves the capabilities of the proposed cloud-based model. Delay wise, cloud-based model shows a promising performance and can improve the scalability and consistency state of the CVE systems greatly.

## V. CONCLUSIONS

This paper analyzed the performance of cloud based architectural model for designing CVE systems. The results obtain from the simulation analysis are compared to that of the traditional client server, Peer-Peer, and Hybrid architecture. The results show that the cloud-based model performance can efficiently improve the performance of the CVE systems even with recent exponential growth of concurrent and simultaneous users. This evaluation proves the capability of the proposed architectural model for satisfying the scalability and the consistency requirements of CVE systems. The comparisons between the traditional models validate the cloud-based model performance. This concludes that the model is promising and can improve the current state of CVE system. In our future work, we intend to design a CVE system using the cloud based architecture to test the architectural suitability of the CVE systems. The limitation of the study is the lack of inclusion of other parameters for the analysis. However, future study will add more parameters for the analytic study.

## ACKNOWLEDGMENT

This research is funded by the Kulliyyah of Information and Communication Technology (KICT), International Islamic

University Malaysia (IIUM). The authors would like to thank the Kulliyyah for their support and cooperation.

## REFERENCES

- [1] D. Delaney, T. Ward, and S. McLoone, "On consistency and network latency in distributed interactive applications: A survey—Part I," *Presence: Teleoperators and Virtual Environments*, vol. 15, pp. 218-234, 2006.
- [2] C. Fleury, T. Duval, V. Gouranton, and B. Arnaldi, "Architectures and mechanisms to efficiently maintain consistency in collaborative virtual environments," in *Proc. of the 3rd IEEE VR 2010 Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS 2010)*, 2010, pp. 87-94.
- [3] C. Shaw and M. Green, "The MR toolkit peers package and experiment," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, 1993, pp. 463-469.
- [4] J. Calvin, A. Dickens, B. Gaines, P. Metzger, D. Miller, and D. Owen, "The SIMNET virtual world architecture," in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, 1993, pp. 450-455.
- [5] M. R. Macedonia, "A network software architecture for large scale virtual environments," Monterey, California. Naval Postgraduate School, 1995.
- [6] R. Waters, D. Anderson, J. Barrus, D. Brogan, M. Casey, S. McKeown, et al., "Diamond park and spline: A social virtual reality system with 3D animation, spoken interaction, and runtime modifiability," 1996.
- [7] E. Frécon and M. Stenius, "DIVE: A scaleable network architecture for distributed virtual environments," *Distributed Systems Engineering*, vol. 5, p. 91, 1998.
- [8] D. Ta, T. Nguyen, S. Zhou, X. Tang, W. Cai, and R. Ayani, "A framework for performance evaluation of large-scale interactive distributed virtual environments," in *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, 2010, pp. 2744-2751.
- [9] C. Carlsson and O. Hagsand, "DIVE—A platform for multi-user virtual environments," *Computers & graphics*, vol. 17, pp. 663-669, 1993.
- [10] W. Piekarski and R. Smith, "Robust gloves for 3D interaction in mobile outdoor AR environments," in *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, 2006, pp. 251-252.
- [11] J. C. de Oliveira, D. T. Ahmed, and S. Shirmohammadi, "Performance enhancement in mmogs using entity types," in *Distributed Simulation and Real-Time Applications, 2007. DS-RT 2007. 11th IEEE International Symposium*, 2007, pp. 25-30.
- [12] T.-Y. Hsiao and S.-M. Yuan, "Practical middleware for massively multiplayer online games," *Internet Computing, IEEE*, vol. 9, pp. 47-54, 2005.
- [13] A. Denault, J. Kienzie, C. Dionne, and C. Verbrugge, "Object-oriented network middleware for massively multiplayer online games," Technical Report SOCS-TR-2008.5, McGill University, Montreal, Canada 2008.
- [14] A. Denault and J. Kienzie, "Journey: A massively multiplayer online game middleware," *Software, IEEE*, vol. 28, pp. 38-44, 2011.
- [15] J. Smed, T. Kaukoranta, and H. Hakonen, "Aspects of networking in multiplayer computer games," *Electronic Library, The*, vol. 20, pp. 87-97, 2002.
- [16] J. Smed, T. Kaukoranta, and H. Hakonen, *A review on networking and multiplayer computer games*: Citeseer, 2002.
- [17] M. Claypool and K. Claypool, "Latency and player actions in online games," *Communications of the ACM*, vol. 49, pp. 40-45, 2006.
- [18] D. C. Miller and J. A. Thorpe, "SIMNET: The advent of simulator networking," *Proceedings of the IEEE*, vol. 83, pp. 1114-1123, 1995.

- [19] D. Schmalstieg and M. Gervautz, "On System Architectures for Virtual Environments," in *Proceedings of the 11th Spring Conference on Computer Graphics, Bratislava, Slovakia*, 1995.
- [20] G. Singh, L. Serra, W. Png, and H. Ng, "BrickNet: A software toolkit for network-based virtual worlds," *Presence-Teleoperators and Virtual Environments*, vol. 3, pp. 19-34, 1994.
- [21] C. Fleury, T. Duval, V. Gouranton, and B. Arnaldi, "Architectures and Mechanisms to Maintain efficiently Consistency in Collaborative Virtual Environments," in *SEARIS 2010 (IEEE VR 2010 Workshop on Software Engineering and Architectures for Realtime Interactive Systems)*, 2010.
- [22] R. V. Chander, "An Improved Object Interaction Framework for Dynamic Collaborative Virtual Environments," *IJCSNS*, vol. 10, pp. 91-95, 2010.
- [23] J. Chen, M. Arumaithurai, X. Fu, and K. Ramakrishnan, "G-COPSS: a content centric communication infrastructure for gaming applications," in *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, 2012, pp. 355-365.
- [24] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. Ramakrishnan, "Copss: An efficient content oriented publish/subscribe system," in *Architectures for Networking and Communications Systems (ANCS), 2011 Seventh ACM/IEEE Symposium on*, 2011 pp. 99-110.
- [25] J. D. Mathias and D. Watson, "Audrey: The Model and Implementation of A Hybrid P2P Framework for Massive Virtual Environments," 2012.
- [26] S.-Y. Hu and G.-M. Liao, "Scalable peer-to-peer networked virtual environment," in *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, 2004, pp. 129-133.
- [27] J.-F. Chen, W.-C. Lin, T.-H. Chen, and S.-Y. Hu, "A forwarding model for voronoi-based overlay network," in *Parallel and Distributed Systems, 2007 International Conference on*, 2007, pp. 1-7.
- [28] J. Lombardi and M. Lombardi, "Opening the metaverse," in *Online worlds: Convergence of the real and the virtual*, ed: Springer, 2010, pp. 111-122.
- [29] S. Benford and L. Fahlén, "A spatial model of interaction in large virtual environments," in *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13-17 September 1993, Milan, Italy ECSCW'93*, 1993, pp. 109-124.
- [30] C. Greenhalgh and S. Benford, "MASSIVE: a collaborative virtual environment for teleconferencing," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 2, pp. 239-261, 1995.
- [31] C. Greenhalgh, J. Purbrick, and D. Snowdon, "Inside MASSIVE-3: flexible support for data consistency and world structuring," in *Proceedings of the third international conference on Collaborative virtual environments*, 2000, pp. 119-127.
- [32] S. Legtchenko, S. Monnet, and G. Thomas, "Blue Banana: resilience to avatar mobility in distributed MMOGs," in *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, 2010, pp. 171-180.
- [33] D. Frey, J. Royan, R. Piegay, A.-M. Kermarrec, E. Anceaume, F. Le Fessant, "Solipsis: A decentralized architecture for virtual environments," in *1st International Workshop on Massively Multiuser Virtual Environments*, 2008.
- [34] A. Y. Gital and A. S. Ismail, "A Framework for the Design of Cloud Based Collaborative Virtual Environment Architecture," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2014.
- [35] A. Y. Gital and A. S. Ismail, "An Alternative Design Of Collaborative Virtual Environment Architecture Based On Cloud Computing," *Journal of Theoretical & Applied Information Technology*, vol. 61, 2014.