

Analyzing Virtual World Region Fidelity on Scalability and Simulation Performance

Sean C. Mondesire¹, Jonathan Stevens², Rebecca Leis², Douglas B. Maxwell¹

¹US Army Research Laboratory, Orlando, FL, USA

²Institute for Simulation and Training, University of Central Florida, Orlando, FL, USA

Email: sean@cs.ucf.edu, jonathan.stevens@knights.ucf.edu, rleis@ist.ucf.edu, douglas.maxwell3.civ@mail.mil

Received 30 June 2015; accepted 24 July 2015; published 27 July 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Virtual world simulation offers tremendous potential opportunities to improve, and optimize, individual and collective echelon training for the military. The US Army Research Laboratory (ARL) Military Open Simulator Enterprise Strategy (MOSES) project's charter is to investigate simulation-based training technology for use in military specific training domains. Of particular interest are attributes of virtual worlds such as geographically distributed large-scale trainee support. We have initiated a series of experiments to determine appropriate benchmarks for simulator performance and to determine appropriate independent variables in order to create a robust predictive model that will enable virtual world training scenario designers to calculate, a priori, the number of trainees that may synchronously operate in a virtual world. The present paper's purposes were to determine the effect that virtual world region fidelity had on server performance and determine whether this independent variable would be appropriate for inclusion into our predictive model. We found that region fidelity had a statistically significant effect on the simulator's processor memory usage but had no significant effect on the simulator's vertical scalability, CPU usage nor network performance. In this paper, we discourse on the purpose of this research, our experimental methodology and results, and discuss the significance of our findings.

Keywords

Region Fidelity, Vertical Scaling, Virtual World Simulation, Performance Analysis, Distributed Simulation

1. Introduction

The use of virtual simulation for training has been proven to increase performance and reduce costs [1] [2]. Thus,

How to cite this paper: Mondesire, S.C., Stevens, J., Leis, R. and Maxwell, D.B. (2015) Analyzing Virtual World Region Fidelity on Scalability and Simulation Performance. *Open Journal of Modelling and Simulation*, 3, 126-135.
<http://dx.doi.org/10.4236/ojmsi.2015.33014>

the US Army heavily leverages virtual *simulation-based training (SBT)* [1] as this class of simulation has empirically demonstrated the successful transfer of *knowledge, skills and abilities (KSAs)* from the simulated environment to the live, or real, environment [2]-[6]. A relatively new variant of virtual simulation is virtual world simulation, which offers tremendous opportunities to improve and optimize individual and collective echelon training for the military. Virtual world simulation is defined as “persistent immersive simulated environments in which a participant uses an avatar (a digital representation of oneself) to interact with digital agents, artifacts, and contexts” [7]. Our research effort is an enduring mission: we seek to optimize the virtual world simulation experience to provide a more effective, and efficient, construct in which to execute military training.

While virtual world simulation is a relatively new and emerging domain [1], this technology offers many potential benefits to improve military training. One of the key advantages of employing virtual worlds for training are the cost savings that may be achieved through the use of distributed training [8] [9]. Virtual worlds enable distributed training through their unique architecture, composed of three major components: a virtualized server farm, thin-client viewers used by exercise participants to interact with the synthetic environment and a reliable network connection between the nodes. Another positive attribute of virtual world simulation is that it offers a persistent, modifiable environment that can accommodate large amounts of trainees who may join the training event asynchronously yet subsequently participate in the training event underway, with the entire collective echelon. A third advantage of virtual worlds for training, unlike traditional virtual SBT, is the potential to support large-scale exercises. Virtual worlds have the potential to be scalable to accommodate more and more simultaneous users, larger simulated operational areas, and more complex scenarios.

The entertainment-driven, three-dimensional platform *Second Life*® [10], is a popular example of a virtual world. From this initial virtual world, the *OpenSimulator (OpenSim)* [11] virtual world was created and offers a Second Life-inspired, open-source platform that allows users to maintain developmental control. OpenSim is a three-dimensional, persistent virtual world simulator that serves as the main component of the *US Army Research Laboratory’s (ARL) Military OpenSimulator Enterprise Strategy (MOSES)* project. Under project MOSES, ARL aims to determine the effectiveness of virtual worlds for military training. Currently, ARL is primarily focused on comparing MOSES’ efficacy to more established, baseline training conditions for military personnel, such as live training and virtual SBT. With the military’s desire for effective, large-scaled virtual trainers, the MOSES project is continuously making strides to increase OpenSim’s scalability to simultaneously train hundreds of personnel in a single virtual environment.

To optimize military training simulation, the MOSES project has initiated investigation into the utility and suitability of a virtual world as a standard platform for military training. As previously discussed, due to limitations of current technologies, virtual SBT is often applied in individual or small group settings. To conduct collective training at higher military echelons, we must determine virtual world hardware and software constraints, tradeoffs, and improvements to provide the best system architecture in order to provide effective training for large groups. Further, proven methodology for measuring and implementing scalability efforts have yet to be established; thus, the MOSES team has incrementally investigated system architecture constraints; particularly, by investigating the effect hardware vertical scaling has on the performance of the virtual world platform.

Scalability efforts have included experimentation involving various independent variables that have been hypothesized to affect virtual world simulator performance. Specifically, the MOSES team has previously assessed simulator performance by analyzing the effect that the number of CPU cores [12], legacy versus prototype hardware configurations [13], network bandwidth [14], and processing memory usage [15] have on virtual world simulator performance. With the information collected, the MOSES research project plans to create a lightweight predictive model that will enable virtual world training scenario designers to calculate, a priori, the number of trainees that may synchronously operate in the virtual world. The model will provide end-users with the minimum resource requirements to support a target number of trainees. It will also allow developers to justify hardware upgrades when necessary and for the optimization of older systems when upgrades are not available.

In this paper, we extend our research thrust by examining another critical independent variable of our predictive model: the aesthetic richness and realism of the virtual environment, referred to as *region fidelity*. Specifically, we focused on determining the effect, if any, that a virtual world’s region fidelity had on the virtual world’s server performance. We decomposed region fidelity into three levels: low, medium and high (to be discussed) in order to ascertain the effect on the virtual world simulator’s performance. The results of this study will be used by the MOSES team in our ongoing effort to develop the desired predictive model. On a wider

context, this examination seeks to inspire other virtual worlds to investigate the impact of region fidelity on their simulator's scalability and performance. Additionally, this extended research contributes to effective methodologies that can examine and expand user-scalability in similar virtual world systems.

2. Background

The US Military conducts virtual training research to both increase the quality of training and continually reduce costs [8]. The Army has published set training standards fueling the adoption of current technologies as one effort to support this research. The *Army Learning Model (ALM)* is an initiative of the *Army Training and Doctrine Command (TRADOC)*, which calls for increased use of technology-based methods leading to learner-centric, outcome-based, training [9]. The ALM changes the content and delivery of training such that military personnel train both in institutional schoolhouses and home station with operational units [9]. Furthermore, the ALM explicitly calls for the adoption of emerging technologies used for individual and collective training, such as the use of virtual worlds. Yet, existing simulation and training content continues to focus on individual training, while both military and large business organizations require alternative domains in which to conduct collective training. However, current virtual world technologies are useful for team (2 - 4 users), squad (8 - 12 users), and platoon units (26 - 55 users), as they still do not provide an adequate training venue for large level military echelons such as battalion (500 - 1000 users), brigade (3000 - 5000 users), and division (10,000 - 25,000 users) formations. As a result, the MOSES project is investigating ways to conduct larger-scale training in virtual worlds.

Recent MOSES research has focused on increasing OpenSim supported echelon levels by examining vertical scalability techniques to increase the number of concurrent users supported. In these efforts, vertical scaling was examined by measuring OpenSim user-scalability and performance as the simulator was allocated various quantities of hardware resources and capabilities.

Because OpenSim is a CPU-intensive virtual world, the team initially investigated CPU-based vertical scaling on the simulator. First, research was performed that compared simulator performance between a legacy 2010 and modern 2015 server [13]. In the work, experiments compared how the different CPU technologies performed with an increased number of concurrent users in the simulation. Results found that there were no significant differences for the number of concurrent users the system was able to support between the tested CPU types. The main conclusion from this work was that the simulator's scalability did not solely improve with the introduction of new CPU technology and research of other methods of vertical scalability was necessary. Based on the results from the first experiment, the team further investigated CPU vertical scaling, this time by analyzing the role of the amount of CPU cores to simulator scalability [12]. In the experiments, 1, 2, 3, 4, 8, 16, and 32 processor cores were evaluated for their impact on how many concurrent users could be supported by the simulator. The results showed that an increase in the amount of cores also did not significantly increase user scalability. However, the work concluded that OpenSim will not meet higher echelon training levels solely with CPU-based vertical scaling due to the non-linear increased scalability of the simulator and its reliance on other types of resources.

The next MOSES experiment investigated network bandwidth limitations on the simulator's user scalability [15]. Specifically, simulator performance was evaluated when its bandwidth was restricted in *megabits per second (Mbps)*, testing 3 G (1.4 Mbps), 4 G (6 Mbps), residential broadband (25 Mbps), and commercial T1 (55 Mbps) network speeds. Results showed that the slowest speed (3 G) supported significantly less users than the other tested speeds and that there was no significant performance difference between 4G, broadband, and T1. Additionally, these results provided virtual world developers and administrators with a few noteworthy contributions and observations. First, the work demonstrated the feasibility of conducting training on slower, less expensive bandwidths; although the slowest speed was not able to support large number of trainees, the simulator was operational for lower echelons. Second, the work showed that diminished returns of concurrent users existed when the simulator operated on faster, more expensive networks.

The team continued to investigate OpenSim's hardware vertical scaling by next examining the simulator's memory usage. Experiment results showed that there were statistically significant differences in OpenSim's process memory once 22 concurrent users were present in the simulation. This threshold is meaningful to military simulation administrators because 22 concurrent users fall between the squad and a platoon echelon. Finding the optimal hardware configuration and processing memory could assist OpenSim developers in determining

cost worthiness to server upgrades. Particularly, for squad level units or lower, server upgrades may be unnecessarily expensive, whereas for platoon level units or higher, server upgrades would be beneficial [1].

To date, MOSES research has been concerned with how the system's hardware supports the number of simultaneous trainees. This presented work investigated a software-based, previously uninvestigated simulation variable, region fidelity, and its effects on user scalability in OpenSim. We turned our attention to the fidelity of the virtual environment because it contributes to the user's overall training effectiveness. Focus was also turned to region fidelity because the virtual world's realism and richness were hypothesized to affect simulator performance due to its resource demands. In this work, we determined the impact of region fidelity on the simulator's performance by analyzing OpenSim's CPU, memory, network, and user scalability responses to supporting different levels of region fidelity.

3. Methodology

Region fidelity measures the richness, quality, and realism of the virtual environment's content. Because MOSES has a goal of supporting highly realistic and effective military training environments, increasing the region fidelity inside of OpenSim has become a priority to the project and its end-users.

Prior to this investigation, it was unclear how much of an impact region fidelity had on the simulator's performance. What was clear, however, as determined through informal testing, was that highly detailed virtual objects require increased amounts of computational resources from the simulator. With these added resource demands, the need for additional hardware, CPU processing, and network bandwidth were always assumed and estimated. The present experiment measured the impact of region fidelity on a virtual world simulator by examining OpenSim under three levels of region fidelity. The experiment measured the simulator's user scalability, CPU performance and memory demands, and network data transfer rates as a result of changes to the region's fidelity.

To perform the analysis, three test cases were executed that collected simulator performance metrics on independent OpenSim simulation runs using low, medium, and high levels of region fidelity. The remainder of this section defines the three region fidelity levels, the process of adding load to the simulation, the experiment's system settings and hardware specifications, the complete step-by-step process to test the three levels of fidelity, and the process of capturing statistical output.

3.1. Region Fidelity and Simulation Load

Three different levels of region fidelity were simulated and evaluated using OpenSim. Each level of region fidelity was defined based upon prior training events whereby MOSES served as a military-based training simulation. The classifications have been used to measure the level of detail of the virtual environment's objects and realism of the training scenario. Inside OpenSim, each environment object is made up of *primitive objects* (*prims*) that define the complexity of the object, similar to polygons in other three-dimensional systems. The measure of prims is an integer value of how many primitive shapes that compose an object. From the MOSES-based classification, low fidelity was defined as a region with less than 5000 prims, medium fidelity contained 5000 to 10,000 prims, and the high fidelity region contained more than 10,000 prims. In this experiment, prior to user logins, the tested prim count was: 1521 prims for the low fidelity region, 7894 prims for the medium fidelity region, and 16,481 prims for the high fidelity region.

In addition to evaluating the simulator's performance with the three region fidelity levels, the simulator's performance was also evaluated against emulated user-based load through the use of automated bots. The bots served as proxies for human users and were carefully created to mimic the same loads and memory footprints as humans logged into the MOSES using a client. The amount of bots present in the simulation served as both an independent and dependent variable. Bot (or avatar) count served as an independent variable due to our experimental methodology which loaded these entities at set time intervals and because each connected user was hypothesized to contribute to both network traffic and resource demands associated with the region fidelity. Avatar count also served as a dependent variable since the number of bots operating in the simulator fluctuates as a function of simulator utilization, which is a critical response variable or performance metric worthy of analysis.

3.2. System Settings Hardware Specifications

In the traditional OpenSim architecture, the simulator is hosted on a separate computer from that of the viewer

software user's computer. The experiment used virtualization to host the simulator on a 2015 Intel Server with physical specifications of 120 Intel Xeon E7-4890v2 CPU cores and 1.5 TB of RAM, connected to a 1 gigabit shared commercial network. OpenSim 8.2 was executed on a virtual machine (VM) hosted on the Intel server that was allocated the traditional MOSES configuration of 8 CPU cores and 32 GB of RAM.

Each bot was instantiated on one of three separate 2010 AMD Bulldozer servers. Each AMD Bulldozer contained 48 AMD Opteron 6100 CPU cores, 500 GB of RAM, a native install of Ubuntu Desktop 14.04 64-bit, and was hosted on a separate network from the Intel server.

3.3. Experiment Process

The process of collecting data for each of the three tested region fidelity levels is as follows:

- 1) The VM hosting the simulator was started.
- 2) An instance of the simulator was started and allowed to initialize.
- 3) The simulator was loaded with either the low and medium, or high fidelity region content.
- 4) The VM was rebooted and the simulator restarted and initialized.
- 5) Statistics collection was enabled, outputting data to a log file at a one-second interval.
- 6) 90 bots were sequentially launched, with a 30-second delay between each launch.
- 7) All of the bots were disconnected from the simulator.
- 8) The OpenSim instance was terminated.
- 9) The VM was shutdown.

3.4. Statistics Collection

Several statistics were gathered to determine the impact of region fidelity on the simulator's scalability, CPU performance, memory demands, and network bandwidth.

Simulator scalability was measured by the maximum amount of concurrent users the simulator could maintain for each of the region fidelities. With the goal of improving OpenSim's scalability to support perhaps thousands of concurrent users, measuring the amount of users the simulator can sustain with the varying levels of region fidelity is crucial to this and ongoing virtual world research.

Process CPU utilization percentage (ProcCPU) was the percentage of CPU time dedicated to the OpenSim instance over the total amount of CPU time. Process memory usage percentage (ProcMem) was the percentage of the amount of RAM used by OpenSim over the total amount of RAM available to the host VM. Both ProcCPU and ProcMem provided a quantifiable measure of the resource demands OpenSim had at any given time of the simulation.

It was suspected that the level of detail for each virtual object also had an effect on the amount of data that is transferred from the simulator. The amount of data that is transferred from the simulator is dependent on the byte size of each virtual object present in the region. To determine how region fidelity affected the simulator's network performance, we collected the average total number of UDP packets transmitted to and from the simulator in one second, referred to as *total packets*.

4. Results

The purpose of this experiment was to determine the effect, if any, that a virtual world's region fidelity had on the virtual world's server performance. This is a significant research question as defense organizations, and to a lesser extent, civilian institutions, migrate more of their training from the live environment to a virtual domain. Thus, the ability to predict the required resources needed to support effective virtual world training events is becoming more critical in order to properly support these more common activities. This study contributes to that overarching goal by examining the effect that a critical independent variable, region fidelity, has on server performance.

The independent variable examined was virtual world region fidelity. Region fidelity was composed of three levels: low, medium and high; previously described. The dependent variables examined included: the simulator's scalability, CPU performance, memory demands and network data transfer rates. The simulator's scalability was defined as both the maximum number of avatars that the server could successfully load and manage as well as the average number of avatars operating in the virtual world. Both of these differentiations were ana-

lyzed separately. CPU performance was measured using the process CPU utilization percentage as the performance metric. Memory demand consisted of the server's process memory usage percentage. Finally, network data transfer rates was measured using the Average Total Packets performance metric, which offers a solid measurement to analyze region fidelity's effect on network performance.

4.1. Simulator Scalability

The maximum number of avatars supported in each region is depicted in Table 1. From Table 1, it is evident that region fidelity did not have a significant effect on the maximum number of avatars supported in the virtual world. While this metric is clearly important to this study, it should be noted that it was an *instantaneous* measurement of performance and thus not necessarily representative of the true effect that region fidelity had on avatar count.

As a result of the above, we conducted additional analysis into region fidelity's effect on the number of avatars supported in the virtual world (Figure 1). We analyzed 24 successive, random intervals of time from the entire data collection period and calculated the average number of avatars operating in the virtual world during each interval. Since avatars were automatically logged out (removed) from the virtual world in response to server overutilization, we believe this is a better method of determining fidelity's effect on scalability as this variable was analyzed continuously. ANOVA found no significant main effect of region fidelity on the average count of avatars in the virtual world $F(2, 71) = 0.03, p = 0.97$. ANOVA was conducted at $\alpha = 0.05$.

4.2. CPU Performance

CPU performance was analyzed through the use of the variable, process CPU utilization percentage; defined as the percentage of CPU time dedicated to the OpenSim instance over the total amount of CPU time. Similar to scalability, we analyzed 24 successive, random intervals of time from the entire data collection period and subsequently calculated the average CPU utilization percentage during each interval. ANOVA found no significant main effect of region fidelity on process CPU utilization $F(2, 71) = 1.48, p = 0.23$. ANOVA was conducted at $\alpha = 0.05$. Processor CPU performance is depicted in Figure 2.

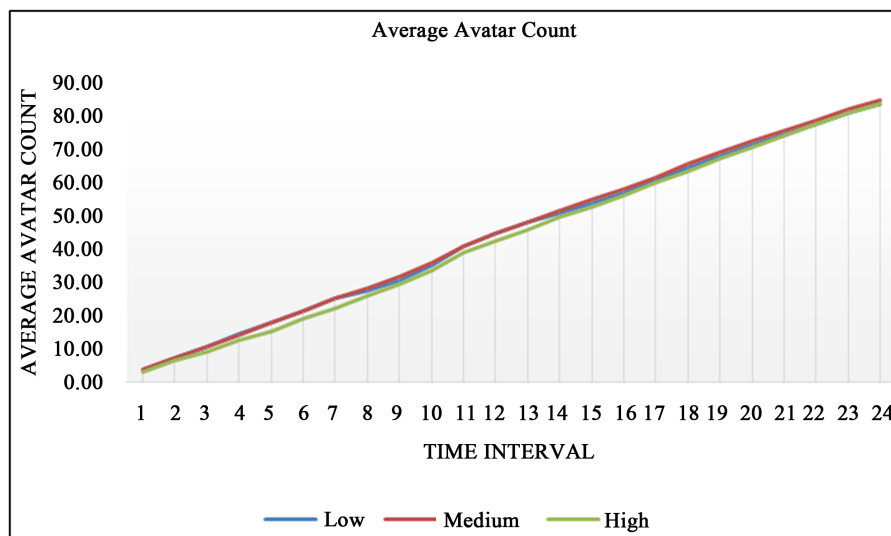


Figure 1. Average avatar count.

Table 1. Maximum number of avatars by fidelity level.

Region	Maximum Avatar Count		
	Low	Medium	High
Maximum	84	85	84

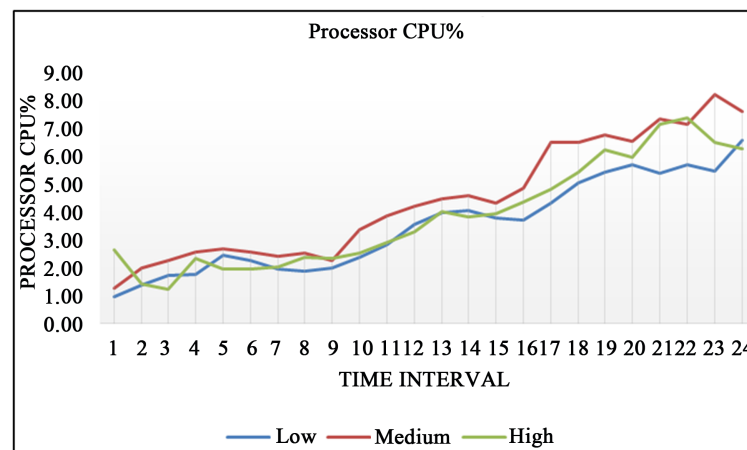


Figure 2. Processor CPU percent.

4.3. Memory Demands

Process memory usage percentage was the amount of RAM used by the server divided by the total amount of RAM available to the host virtual machine. We analyzed this dependent variable over 24 successive, random intervals of time from the entire data collection period and subsequently calculated the average process memory usage percentage. ANOVA found a significant main effect of region fidelity on process memory usage percentage $F(2, 71) = 48.62, p = 0.00$. ANOVA was conducted at $\alpha = 0.05$. Post-hoc pairwise comparisons using the Tukey HSD test indicated that the mean process memory usage percentage of the three levels was significantly different from each other, $p < 0.001$. Process memory usage percentage was found to significantly increase with each corresponding increase in region fidelity. Relationally, the process memory usage percentage of Low Fidelity < Medium Fidelity < High Fidelity. Processor memory usage is depicted in Figure 3.

4.4. Network Data Transfer Rates

Network data transfer rates were statistically analyzed using the Average Total Packets performance metric as the dependent variable. Average Total Packets is the average number of packets sent and received, both to and from the simulator. This variable provides a practical measurement of the network data transfer rate associated with the server's region fidelity. Similar to before, we analyzed this dependent variable over 24 successive, random intervals of time from the entire data collection period and subsequently calculated the average value.

Average Total Packets is depicted in Figure 4. ANOVA found no significant main effect of region fidelity on Average Total Packets $F(2, 71) = 1.79, p = 0.17$. ANOVA was conducted at $\alpha = 0.05$. Our data analysis revealed the region's fidelity did not significantly influence network transfer rates, according to this performance metric.

5. Discussion

The US Army has recognized the benefits of *simulation-based training (SBT)* and is committed to improving the efficiency of transfer of training, reducing cost and improving the overall user experience associated with SBT. Virtual worlds for training represent a novel approach to achieving these objectives. Furthermore, to date, minimal research has been conducted that examines both the efficacy of virtual worlds for military training as well as the effects that various critical independent variables have on virtual world architecture performance. The MOSES virtual world project team has initiated investigation into the utility and suitability of virtual worlds for military training. We have initiated a collection of experiments to determine appropriate benchmarks for simulation performance and to determine appropriate independent variables in order to create a predictive model. The present paper focuses on region fidelity as a potential independent variable for the predictive model.

The purpose of this research effort was to determine the effect that virtual world region fidelity had on server performance. The results of this study will be used by the MOSES team in our ongoing effort to develop a predictive model that will enable virtual world training scenario designers to calculate, a priori, the number of trainees that may synchronously operate in the virtual world. This is critical information, particularly for the mili-

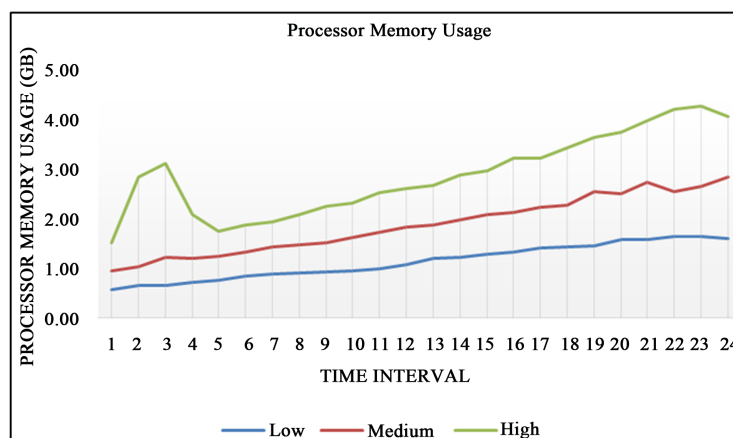


Figure 3. Processor memory usage.

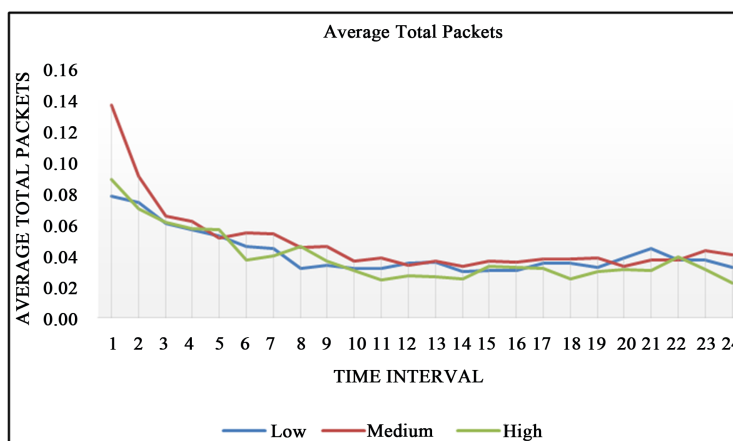


Figure 4. Average total packets.

tary training community, as virtual world training becomes ever more ubiquitous, due in large part to both the lower cost of virtual training as well as recent advancements in simulation training technology that support realistic training conditions. Simultaneously, the virtual world community is provided with a methodology to determine the impact of content richness and realism on their own simulators.

We isolated and controlled the independent variable, region fidelity, so as to measure its effect on critical performance measures (dependent variables) of the virtual world server architecture. Those dependent variables included the simulator's scalability, CPU performance, memory demands and network data transfer rates. These dependent variables were selected in large part due to their criticality in optimizing server performance, which in turn optimizes the training experience for end-users in the form of minimized latency, reduced errata during log-in procedures, as well as the proper calculation of physics, ballistics and motion effects which provide an enhanced training experience.

Our results were noteworthy and a number of observations were noted. First, the simulator's CPU usage was not significantly affected by an increase in the total amount of prims present in the region. This observation is attributed to the tested environments primarily containing non-physical objects. By populating the regions with basic objects, the simulator required minimal CPU time used to process physics calculations (including collisions and gravitational forces per object) and execute CPU-intensive scripted instructions. Furthermore, these results indicated that the CPU-footprint for each prim was so minimal that there was no significant distinction between the three region fidelity levels.

Second, there was a statistically significant difference in the amount of memory utilized by the simulator based upon the region fidelity employed. As hypothesized, the simulator retrieved and cached data corresponding to each accessed primitive object in the regions, requiring additional memory usage with the increased re-

gion fidelity's prim count. However, the practical significance of this finding is minimized as the amount of memory used never exceeded 5% of the server's total available memory, for all three region fidelity levels.

Next, we analyzed the impact of region fidelity on the simulator's network performance. We found no significant difference in the average total packets sent and received by the simulator. From this result, it was deduced that the rate at which data was transmitted to and from the simulator was not affected by the fidelity of the region. The lack of significant difference in transmission is attributed to the simulator's use of cache to store each object and packet throttling. Because each object in a region is served to its clients on-demand (based on the user's in-world location), the simulator accesses the data for the objects around the user from the asset server and stores the result in a cache. Initially this object retrieval requires network communication between the asset server and the OpenSim instance but as the simulator fills its cache, there are more cache hits and less requests to the asset server. This use of a cache means that after the simulation accepted the first few users, the simulator's network traffic to the asset server stabilized as fewer objects were required to be retrieved. The remaining data transmitted by the simulator is exchanged between OpenSim and each of its connected users. OpenSim throttles the number of packets transmitted based on its simulation frame, commonly referred to as a *game-loop* iteration in other systems. Therefore, the simulation state is constantly transmitted from the simulator to its users at a controlled rate, regardless of the amount of data to transfer. It is hypothesized that this packet throttling was the primary reason why we discovered no significant difference between the three region fidelity levels.

Finally, simulator scalability, measured as both the maximum number of avatars and average number of avatars operating in the virtual world, was not influenced by the fidelity level of the region chosen. Because there were no significant differences in CPU usage and the total amount of packets transmitted and because the total memory demand was very minimal for the three levels of fidelity, the simulator never approached the point where processing load would affect user scalability. Therefore, we can conclude that region fidelity did not have a direct impact on user scalability in the simulator.

Following the scientific method, based upon our results, we conclude that region fidelity had a minimal effect on virtual world simulator performance. We attribute the majority of this finding to the low-intensive primitive objects used to construct the different virtual world fidelity levels. Our experimental design was externally valid, as the region fidelity levels settings were based upon our team's experience in designing, developing, resourcing and executing virtual world training events for a military audience.

Future work will extend our analysis of various simulator attributes on the performance of a virtual world. For our next work we will extend the presented network analysis by capturing additional network statistics and analyzing more network-based test cases. This analysis will allow us to measure how much data is transmitted to and from the simulator and measure and identify cases of when the simulator would be under network duress. Thereafter, we will examine the impact of scripted and physical objects on the virtual world. These types of objects have CPU, memory, and network costs currently unknown to the OpenSim and MOSES community. We aim to determine the resource demands associated with these object categories and educate the community for improved virtual world design, implementation, and execution.

References

- [1] Lele, A. (2013) Virtual Reality and its Military Utility. *Journal of Ambient Intelligence and Humanized Computing*, **4**, 17-26. <http://dx.doi.org/10.1007/s12652-011-0052-4>
- [2] Harrington, M. (2011) Empirical Evidence of Priming, Transfer, Reinforcement, and Learning in the Real and Virtual Trillium Trails. *IEEE Transactions on Learning Technologies*, **4**, 175-186. <http://dx.doi.org/10.1109/TLT.2010.20>
- [3] Blow, C. (2012) Flight School in the Virtual Environment. United States Army Command and General Staff College, Fort Leavenworth, Kansas.
- [4] Seymour, N., Gallagher, A., Roman, S., O'Brien, M., Bansal, V., Anderson, D. and Satava, R. (2002) Virtual Reality Training Improves Operating Room Performance. *Annals of Surgery*, **236**, 458-464. <http://dx.doi.org/10.1097/0000658-200210000-00008>
- [5] Hays, R., Jacobs, J., Price, C. and Salas, E. (1992) Flight Simulator Training Effectiveness: A Meta-Analysis. *Military Psychology*, **4**, 63-74. http://dx.doi.org/10.1207/s15327876mp0402_1
- [6] Salas, E., Rosen, M., Held, J. and Weissmuller, J. (2009) Performance Measurement in Simulation-Based Training: A Review and Best Practices. *Simulation & Gaming*, **40**, 328-376. <http://dx.doi.org/10.1177/1046878108326734>
- [7] Dawley, L. and Dede, C. (2013) Situated Learning in Virtual Worlds and Immersive Simulations. In: Spector, J.M.,

- Merrill, M.D., Elen, J. and Bishop, M.J., Eds., *Handbook of Research on Educational Communications and Technology*, New York, Springer, 723-734.
<http://www.amazon.com/Handbook-Research-Educational-Communications-Technology/dp/1461431840>
- [8] Dutta, D. (1999) Simulation in Military Training: Recent Developments. *Defense Science Journal*, **49**, 275-285.
<http://dx.doi.org/10.14429/dsj.49.3840>
 - [9] Stafford, R.J. and Thornhill, W.M. (2012) The Army Learning Model: Changing the Way Sustainers Train. *Army Sustainment*, **44**, 28-31.
 - [10] Linden Research, Inc. (2015) Second Life Official Site. <http://secondlife.com/>
 - [11] Overte Foundation (2015) Main Page. <http://opensimulator.org/>
 - [12] Mondesire, S.C., Stevens, J. and Maxwell, D.B. (In Press) Vertical Scalability Benchmarking in Three-Dimensional Virtual Worlds Simulation. *Proceedings of the 2015 Summer Simulation Multi-Conference*, Chicago, 26-29 July 2015.
 - [13] Mondesire, S.C., Stevens, J. and Maxwell, D.B. (In Press) An Analysis of Increases Vertical Scaling in Three-Dimensional Virtual World Simulation. *Proceedings of the 8th International Conference on Simulation Tools and Techniques*, Athens, 24-26 August 2004.
 - [14] Mondesire, S.C., Stevens, J. and Maxwell, D.B. (Submitted) Network Bandwidth's Effect on Virtual World Simulator Performance Optimization. *Proceedings of the 2015 Interservice/Industry Training, Simulation and Education Conference (IITSEC)*, Orlando, 30 November-4 December 2015.
 - [15] Stevens, J., Mondesire, S.C., and Maxwell, D.B. (Submitted) Human Entities' Effect on Server Performance in Distributed Virtual World Training. *Proceedings of the 2015 Fall Simulation Interoperability Workshop*, Orlando, 30 August-4 September 2015.