

Textures in Second Life: Measurement and Analysis

Huiguang Liang and Mehul Motani
Department of Electrical and Computer Engineering
National University of Singapore

Wei Tsang Ooi
Department of Computer Science
National University of Singapore

Abstract

We collected packet traces from Second Life client sessions and analyzed the packet contents. We observed that textures constitute a majority of the network traffic. We further characterized the textures from three selected regions in Second Life in terms of their size and spatial distributions. We found that textures in these regions exhibit a different size distribution from files on a file system or documents on the Web. We also verified the intuition that texture objects are spatially non-uniformly distributed. Surprisingly, we found that the selected Second Life regions can contain up to hundreds of megabytes of textures, and there exist locations in these regions that encompass a large portion of these textures within their area-of-interest. Our work motivates the need to manage textures carefully and efficiently in the design of networked virtual environments such as Second Life, and hints at the amount of storage and bandwidth required at a peer if peer-to-peer techniques are applied for texture caching. Our traces are useful for simulation studies and can lead to a model to generate realistic workload for networked virtual environments.

1. Introduction

Metaverses are an increasingly popular class of networked applications that provide a platform for user-created virtual worlds. In a metaverse, a user can purchase or rent a plot of land, construct buildings and objects, and define rules of behavior within the land. Further, avatars can interact, communicate, and trade with each other. Developing a scalable metaverse poses many research challenges and is in many ways more challenging than developing an MMORPG.

A crucial difference between MMORPGs, such as World of Warcraft, and metaverses, is that, the content in a metaverse is user-created, and therefore is dynamic and is constantly expanding in size. Unlike objects in a

static virtual world, which can be distributed to users on a storage media, most objects in a metaverse needs to be downloaded on demand, as and when needed. In an MMORPG, most traffic is consumed by communications between players and state updates, whereas in a metaverse, a significant portion of the network traffic is consumed by transmission of objects.

To design and evaluate a scalable architecture that supports a metaverse application, it is important to understand the network traffic and objects access patterns. While there is previous work in understanding network traffic at the packet-level [5, 4], there are no existing similar studies at the application level. Such application-level studies would be useful to understand, for instance, how many objects are in the metaverse, how large are the objects, how are the object sizes distributed, etc. This would guide system design and help in determining proper design parameters, such how much storage should be allocated to cache the objects in a peer-to-peer metaverse.

As a first step towards understanding objects access patterns in metaverses, we conducted a study on Second Life, the most popular metaverse at the time of writing. In this paper, we present our approach to collect object traces in Second Life. We found that, textures constitute a significant amount of network traffic in Second Life, and therefore focus on its measurement and analysis. We observe that there are hundreds of megabytes of textures in a region, and their spatial distributions are skewed. As a result, certain locations within a region could contain a significant portion of the region's textures in their area-of-interest.

The rest of this paper consists of five sections. Section 2 introduces the background and related work, including Second Life and measurement research on it. Section 3 describes our methods to collect the texture traces. Section 4 presents the results of preliminary measurements that establish textures as the main data type transferred in Second Life. Section 5 presents an analysis of textures collected. Finally, Section 6 concludes the paper.

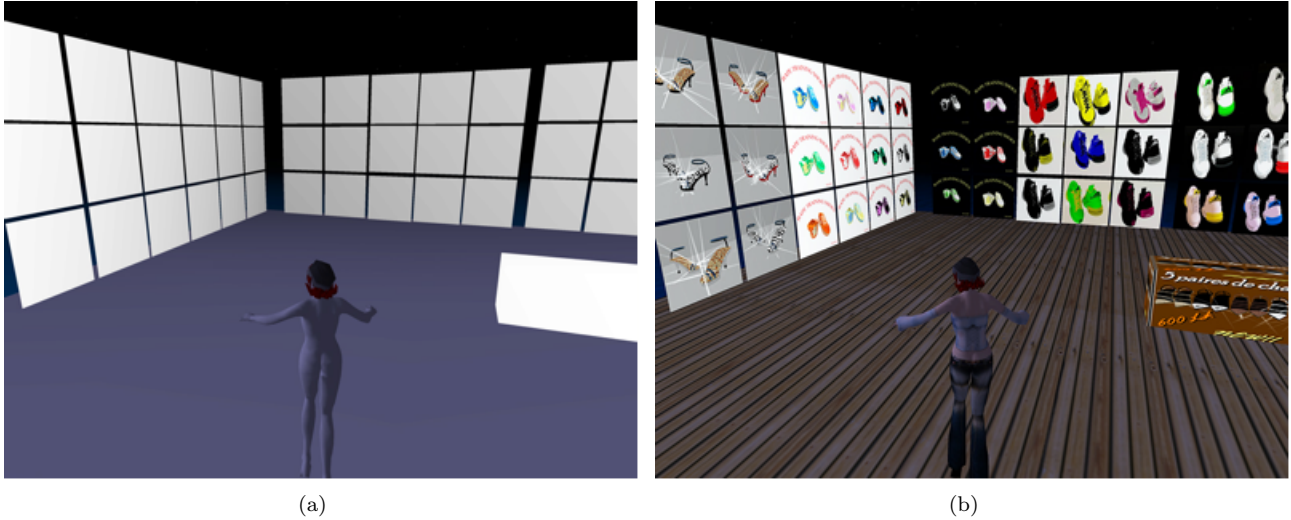


Figure 1. Textures in Second Life: (a) A scene with textures removed. (b) A scene with textures.

2. Background and Related Work

In this section, we describe Second Life, a metaverse environment in which we conducted our measurement experiments. We also discuss previous work in the literature related to measurement and analysis of Second Life traces.

2.1. Second Life

Second Life is currently the most popular metaverse application available, with an average of 38,000 simultaneous users spending 28 million man-hours in January of 2008 alone¹. The virtual world in Second Life is divided into independent *regions*. Each region is a 256m by 256m of land.

A unique aspect of Second Life is that users can create anything they want within the virtual world. Second Life provides tools for users to easily build objects, such as houses, furniture, and clothes. The behavior of the objects can be scripted using Linden Scripting Language. Users can upload images into the virtual world and apply them as textures on the objects. Textures are used not only to give the right appearance that matches an object material, but also used as advertisement, notice boards, paintings on walls within Second Life (See Figure 1). Textures are encoded progressively using JPEG 2000 format in Second Life.

We choose Second Life to conduct our measurement, as the Second Life client is open source, providing opportunities to reverse engineer the protocols on which Second Life runs. One of the effort to reverse engineer

Second Life is the libsecondlife project, which provides an open source C# library that parses various packets exchanged between a Second Life client and server.

In this study, we are particularly interested in the following packet types.

- **ObjectUpdate** and **ObjectUpdateCompressed**: sent from the server to the client to inform the client of the existence of objects. Each packet can encapsulate a list of objects. Each object description includes a list of textures, each identified by a 128-bit ID, the identify of the parent object, and position of the object relative to its parent.
- **RequestImage**: the client can request for a particular texture, identified by an ID, after receiving descriptions of objects through the **ObjectUpdate** and **ObjectUpdateCompressed** packets.
- **ImageData** and **ImagePacket**: sent from the server to the client. **ImageData** contains the size of the texture as well as the base resolution representation of the texture. **ImagePacket** contains additional levels of representation of the texture, allowing rendering of the texture at increasing resolution.
- **ImprovedTerseObjectUpdate**: sent periodically or when needed in order to inform the client about updates to objects (such as changes of texture by owner) while it lies within the client's view. This packet also contains a Boolean indicating if the object is an avatar.

¹http://secondlife.com/whatis/economy_stats.php

2.2. Related Work

Kinicki et al. [5] and Fernandes et al. [4] have conducted measurements and analysis of Second Life packet-level network traffic. They focus on understanding the traffic behavior of a client when the avatar performs different actions at different regions, under different network conditions. They analyze the throughput, packet sizes, and packet inter-arrival times of the traffic generated by Second Life.

We [7] and La et al. [6] have collected and analyzed movements of avatar within Second Life. La et al. focuses on metrics related to mobile communications, while we analyzes metrics relevant to systems issues in a networked virtual environments.

None of the previous work specifically analyzes the content of Second Life. This paper is the first step towards understanding an important type of content, textures, in Second Life.

Our work is inspired by similar studies on file size distributions [8, 9, 3] and Web object size distributions [1, 2]. These influential studies allow system designers to select proper parameters for file systems and caching proxies, as well as explain self-similar network traffic patterns. Our aim is similar in spirit. We hope that by understanding textures in a networked virtual environment, the research community can design and build a better system.

3. Methodology

3.1. Second Life Proxy

To collect information about packets transferred in Second Life, we wrote a proxy server using libsecondlife. We then log into a chosen region in Second Life using the official Linden Lab client, configured to connect to the proxy instead of the default Second Life server. The proxy will then connect to the Second Life server and relay packets between the client and the server. We clear the local cache of the client before logging in to ensure that all objects in the region are requested from the server. We log all packets at the proxy.

3.2. Regions

To run our experiments, we chose two popular regions, Freebies and Isis, as well as a medium popular region, Ross.

- Isis has a strong adult theme. Avatars in Isis can participate in paraphiliac activities and buy adult-oriented items.

- Freebies is a region that gives away free objects, such as clothes, accessories, and other inventory items.
- Ross is a region with medium popularity that is provided by Linden Lab to distribute information and serves as meeting place for avatars (so called *infohub*).

Isis and Freebies are consistently among the most popular regions in Second Life at the time of our experiments (21 - 27 March 2008).

4. Preliminary Studies

We first describe a preliminary experiment we conducted to find the amount of textures transferred in Second Life. We log into each of the chosen region and move our avatar systematically, step-by-step, within the region. At each step, the avatar stops and waits for all textures to be downloaded and rendered before moving to the next step. We spent 10 minutes in each region and count how much bytes of textures are transmitted. This counting can be done by filtering out all packets of types `ImageData` and `ImagePacket` in Second Life. We disable transmissions of audio in this experiment.

The data collected are summarized in Table 1. We found that textures constitute 61% to 88% of all packets in terms of size. This experiment verifies that textures are the dominant type of data transmitted in Second Life.

Next, we now describe how we collect information about textures from Second Life.

5. Textures: Measurement and Analysis

To collect information about textures in a region of Second Life, we repeat the experiments above, but instead of 10 minutes, we move the avatar systematically until the whole region is covered and all textures within the region have been downloaded. We log, at the proxy, all packets of types `ObjectUpdate` and `ObjectUpdateCompressed`, which gives the position of the textures in a region, and packets of type `ImageData` packets, which gives the size of each texture.

We focus only on textures applied to objects in this study. Avatars in Second Life are textured as well. But as avatars are mobile, the position of the textures changes, and therefore it is not meaningful to analyze their spatial distribution or predict if the textures will be downloaded by another avatar without adding a

Region	Textures (KB)	Non-Textures (KB)	Ratio
Freebies	42,706	14,739	0.743
Isis	15,994	9,967	0.616
Ross	52,951	7,099	0.882

Table 1. Summary of Packets Collected

Region	Number of Textures	Total Size (Bytes)
Freebies	1,782	153,348,148
Isis	3,527	351,639,821
Ross	2,860	235,947,020

Table 2. Summary of Texture Information Collected

time component in our analysis. We therefore excluded avatar textures in this study.

Some of the textures collected in our traces do not have a position, and are excluded from the analysis as well. This anomaly occurs because Second Life reports positions of textures relative to the objects they are attached to. To resolve the position of a texture object, the position of its parent object is needed. But, the Second Life server sometime do not report the object positions, possibly because the object is an avatar that has moved out of range and therefore are the state updates are culled by the server. Note that each texture has a position in a 3D space, but for the purpose of this study, we focus only on the x- and y- coordinates.

Table 2 summarizes the textures traces collected.

5.1. Distribution of Texture Sizes

We found that texture sizes range from 140 Bytes to 1536 KB. We plot the CDF of the texture sizes for the three regions in Figure 2. Textures from Isis are larger in general than those from Ross and Freebies, but they show similar size distributions. In particular, all regions exhibit a bump in the distributions at 10 KB, 50 KB, 100 KB, and 200 KB, indicating that these are popular sizes for textures within these regions. The median texture size over all three regions is 95 KB.

The median texture size is also much larger than the median reported in previous studies on file systems and web servers, which gives a median close to 1 KB [9]. This is not surprising, since textures are images that are much larger in size than text and html documents, which are the dominant file types in previous studies. The dominance of textures of certain sizes, however, are not expected and are not seen in size distributions of files and web documents.

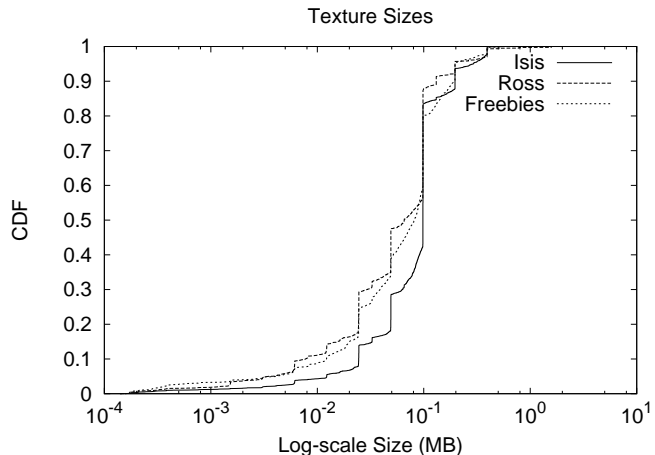


Figure 2. Distribution of Texture Size

5.2. Spatial Distribution of Texture

Next, we look at how these textures are distributed within a region. We divide a region into 256x256 cells and count how many textures are there within a region based on reported texture position.

Figure 3 shows the distribution of textures within the three regions. Each pixel in the figure corresponds to a cell, and the density of the pixel corresponds to the number of textures found in each cell. Brighter pixel means more textures. The brightness is drawn at log scale due to the skewness of the distribution.

We can see from the Figure 3 that the textures are not evenly distributed and they tend to concentrate along the walls of rooms and corridors. A cell contains up to 19 individual textures. Such cells are observed in Ross and Freebies, giving the total size of 1448 KB and 1065 KB respectively. The cell with the largest amount of textures, in terms of total bytes, has 3913 KB of textures (in Freebies).

5.3. AoI Volume

We now investigate how much texture would need to be downloaded by an avatar, if the avatar is positioned in a given cell. We assume that all textures within a

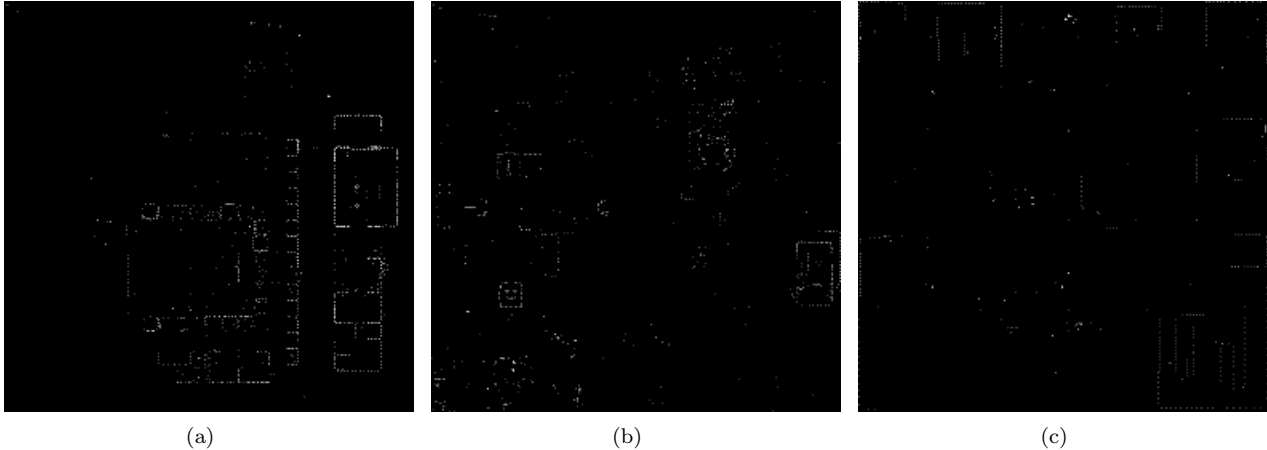


Figure 3. Spatial Distribution of Textures for (a) Isis, (b) Ross, and (c) Freebies

given area-of-interest (AoI) of the avatar is to be downloaded (we observe that Second Life does not consider occlusions when deciding what objects to transmit). We set the AoI to a radius of 64m, the default value in Second Life. We call the total size of textures within a cell’s AoI as the *AoI volume*.

Figure 4 shows the spatial distribution of AoI volume within the three regions. Again, each pixel corresponds to a cell, and the density of the pixel corresponds to the AoI volume.

Due to the uneven spatial distribution of textures within a region, we can see a high concentration of AoI volumes in certain areas of the regions. The CDF of the cells’ AoI volume is shown in Figure 5. The median AoI volume for Freebies, Isis, and Ross are 17.6 MB, 45.4 MB and 40.1 MB respectively, while the maximum for the regions are 46.8 MB, 198.0 MB, and 75.8 MB. We found that textures are especially abundance in Isis and are more concentrated.

6. Conclusion

In this paper, we provide the first characterization of textures in Second Life, a popular metaverse application. Our findings are as follows.

First, we show that, in Second Life, transmission of textures constitutes a majority of network traffic. In a region with medium popularity (not many other avatars), we observe that textures consume up to 88% of all network traffic. We expect that similar observations can be made in other metaverse applications where objects are modeled as constructive solid geometry and require fewer bytes to represent. This observation motivates the need to understand and characterize the texture workload in a networked virtual

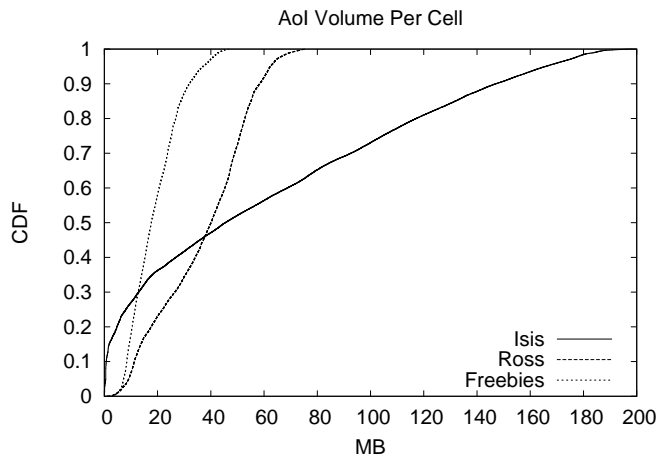


Figure 5. Distribution of AoI Volume

environment such as Second Life.

Second, we show that, in Second Life, regions can contain up to hundreds of Megabytes of textures. Further, an avatar can potentially encompass large amounts of textures within its AoI, again up to hundreds of Megabytes. Our previous studies of avatars in Second Life [7] shows that we have about 2,700 avatars visiting a region (Isis) in a day. Even if each avatar downloads 10% of the textures, 94.5 GB of textures needs to be served from one region alone. This motivates the needs of better texture caching algorithms, either locally or at a proxy. For a peer-to-peer NVE, our observation shows that there are potential benefits for a peer to cache textures on behalf of other peers. Our measurements also hint at the reasonable values we should set for the size of such a cache.

Third, we show that textures are not uniformly

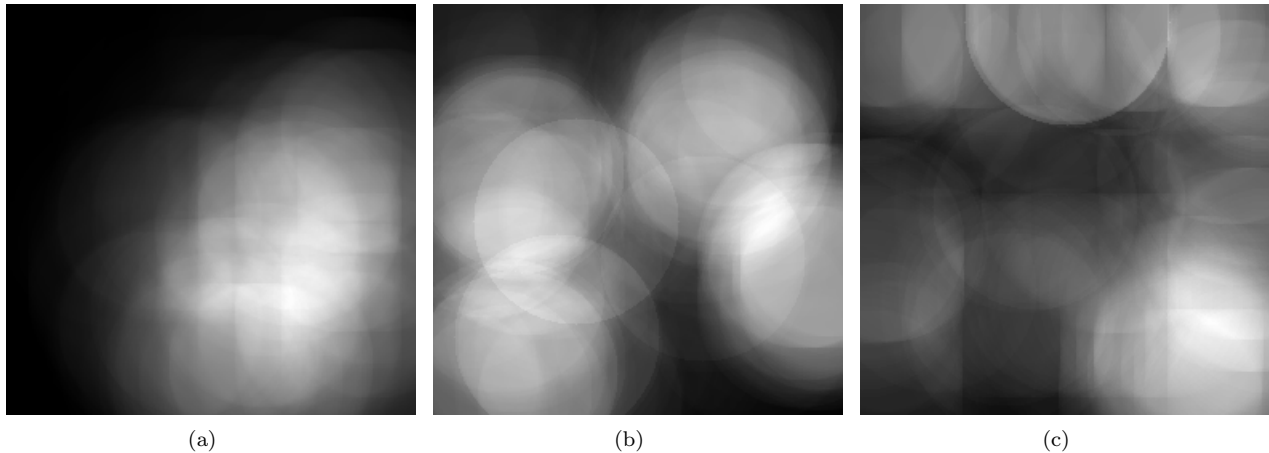


Figure 4. Spatial Distribution of Aol Volume for (a) Isis, (b) Ross, and (c) Freebies

distributed within a region. For an NVE that uniformly distributes a virtual world into cells, and assigns a server (or super-peer) to serve each cell, there can potentially be serious imbalance in the server load due to the different amount of textures stored in each server/super-peer.

Our traces are also useful in other ways. They can be used in trace-based simulations to evaluate new algorithms or new protocols for NVEs, allowing evaluations in more realistic and practical settings compared to synthetic inputs. The trace can also be used to derive a mathematical model to generate realistic workload for NVE research. We plan to make our traces public to the research community.

6.1. Future Work

We plan to continue the same experiments on more regions. We are in the process of combining our avatar mobility traces with our texture traces, so that we can model which textures are downloaded by which avatars at what time, therefore allowing us to generate the working set of textures referenced. This further allows us to characterize the access workload on the server.

We excluded the texture traces of avatars in this work. We would like to further characterize these avatar textures and understand the impact of such moving textures in the design of NVEs.

6.2. Acknowledgment

We would like to thank Ming Feng Neo and Ian Tay of National University of Singapore, as well as Vikram Srinivasan of Bell Labs India for useful discussions throughout the project.

References

- [1] P. Barford, A. Bestavros, A. Bradley, and M. Crovella. Changes in web client access patterns: Characteristics and caching implications. *World Wide Web*, 2(1-2):15–28, June 1999.
- [2] M. E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.
- [3] J. R. Douceur and W. J. Bolosky. A large-scale study of file-system contents. In *Proceedings of the 1999 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 59–70, Atlanta, Georgia, United States, 1999.
- [4] S. Fernandes, C. Kamienski, D. Sadok, J. Moreira, and R. Antonello. Traffic analysis beyond this world: the case of Second Life. In *Proceedings of the 17th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Urbana-Champaign, IL, June 2007.
- [5] J. Kinicki and M. Claypool. Traffic analysis of avatars in Second Life. In *Proceedings of the 18th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Braunschweig, Germany, May 2008.
- [6] C.-A. La and M. Pietro. Characterizing user mobility in Second Life. Technical Report RR-08-212, Institut Eurecom.
- [7] H. Liang, M. Neo, I. Tay, M. Motani, and W. T. Ooi. Avatar mobility in networked virtual environment: Measurements, analysis, and implications. Technical Report arXiv:0807.2328.
- [8] S. J. Mullender and A. S. Tanenbaum. Immediate files. *Software: Practice and Experience*, 14(4):365–268, 1984.
- [9] A. S. Tanenbaum, J. N. Herder, and H. Bos. File size distribution on UNIX systems: then and now. *SIGOPS Operating Systems Review*, 40(1):100–104, 2006.