Is There Life in Second Life?

Matteo Varvello[†⋆], Fabio Picconi[†], Christophe Diot[†], Ernst Biersack[⋆]
[†] Thomson, Paris, France
[⋆] Institut Eurecom, Sophia-Antopolis, France
{*matteo.varvello,fabio.picconi,christophe.diot*}*@thomson.net, ernst.biersack@eurecom.fr*

ABSTRACT

Social virtual worlds such as Second Life are digital representations of the real world where human-controlled avatars evolve and interact through social activities. Understanding the characteristics of existing virtual worlds can be extremely valuable to optimize their design. In this work we perform the first extensive analysis of Second Life. We have crawled around 13000 Regions over one month, and gathered information about objects, avatars, and server state. The analysis of our traces shows several surprising results. We find that 30% of the Regions are never visited during a six day period, whereas only few Regions have large peak populations. Moreover, the vast majority of Regions are static, i.e., objects are seldom created or destroyed. Interestingly, avatars interact similarly to humans in real life, gathering in small groups, visiting the same places and meeting the same avatars again, showing a highly predictable behavior. Based on these observations, we discuss several techniques to enhance Second Life or other similar social virtual worlds.

1. INTRODUCTION

In the past few years, we have observed a quick growth of social virtual worlds. These are Networked Virtual Environments (NVEs) where people can meet, play, trade and even contribute to the development of the NVE. Second Life[1] (SL), launched in 2003 by Linden Lab, has become the most popular social virtual world, reaching 14 million registered users in June 2008.

SL consists of a virtual land, divided into fixed-size *Regions*, where users interact via their digital representation called *avatar*. The main innovative feature of SL is user-generated content: avatars participate in the development of the virtual environment by creating *objects* such as cars, walls, trees, and buildings. In addition, SL has created a full-blown economy, attracting multiple companies which have invested millions of dollars in order to build their own virtual products and advertisement.

Despite the widespread interest that SL has generated on both users and companies, very little is known about its characteristics. Its design is based on a Client/Server architecture: each Region is managed by a server called "Simulator", and users run "thin" stateless clients which simply perform the three-dimensional rendering of the virtual world. However, the Client/Server protocol is not public, and reverse engineering efforts are underway[2]. Moreover, SL provides very little information about the characteristics of the virtual world, usually citing only the total number of Regions, registered accounts, and online avatars.

Given the lack of information about SL, recent work has focused on studying some of its aspects. Some authors have analyzed the network traffic generated by the SL client [5] [11], while others have characterized avatar interactions in a few selected Regions [7]. However, no studies have looked at other important characteristics of SL, such as the object distribution and creation rate, or the impact of the number of concurrent avatars on server load and user experience. Most importantly, no SL study has collected information at a global scale, i.e., over a large portion of the entire SL world.

In this work, we present the first comprehensive study of SL. Our main motivation is to understand both its global and local-scale characteristics in terms of content (i.e., user-generated objects) and active avatar population. By shedding some light on SL, our study can be useful to understand how to improve its design. In addition, it allows us to study the social behavior of human beings in a virtual context.

In order to conduct this study, we designed and deployed a *crawler* application. Our crawler is a modified SL client that connects to SL servers, and exploits standard avatar capabilities to collect information about the virtual world. We monitored the public part of SL, i.e., around 13000 Regions, during April 2008. We find that the number of objects per Region was roughly constant during one month (see Section 5.2). The active population at any point of time was between

[1] http://secondlife.com/

[2] http://www.libsecondlife.org/

CoNEXT'08, December 9-12, 2008, Madrid, Spain.

30000 and 50000 avatars, i.e., about 0.3% of the registered avatars. Quite surprisingly, about 30% of the Regions were continuously empty during six days (see Section 5.3).

We also notice that avatars tend to organize in small groups of 2 to 10 avatars. This result suggests that the human "attention budget" theory [8] [10] may also hold in social virtual worlds. Large group of avatars are very rare, and are mostly driven by the presence of events such as concerts and shows (see Section 5.4). Moreover, we observe that avatars, like their human counterparts, tend to visit the same virtual places across different sessions.

Finally, we selected 5 very popular Regions and monitored them in more detail (see Section 5.5). We find that these Regions have a strong social activity: half of the avatars form groups of a few "Good Friends" who meet frequently at the same locations. Around 5% of their population stays connected almost all the time, and is most likely identifiable as *bots* (i.e., automated avatars). Surprisingly, despite the possibility to walk, fly or teleport, avatars do not move around 90% of the time, suggesting that users spend their time socializing rather than moving around the Region. These Regions also reveal SL's scalability issues: servers become often overloaded even for small populations, and are forced to significantly slow down the Region's "virtual time" to artificially reduce avatar activity.

Based on our observations, we identify several mechanisms to improve SL's design (see Section 6). Given the highly static object distribution and predictable nature of avatar behavior, caching and prefetching techniques could be particularly useful. In addition, we believe that hybrid Peer-to-Peer and Client/Server architectures could be an interesting alternative to SL's current design, exploiting direct communications between clients, while still preserving a server for critical tasks (e.g., security and persistency).

To the best of our knowledge, this is the first attempt to perform a comprehensive study of SL, and to understand its object and avatar characteristics. Furthermore, the traces we collected, which we make publicly available, provide valuable information for the design of future NVEs.

2. RELATED WORK

Networked Virtual Environments (NVEs) have recently attracted the attention of the research community. Svoboda et al. [12] have analyzed the traffic generated by World of Warcraft (WoW), a popular multiplayer online game, using the traces from the METAWIN[3] project. Their work focuses on the modeling of WoW traffic patterns and proposes a NS-2 traffic generator to reproduce it. They also show the modest bandwidth requirements of the WoW client, which typically generates only 6.9 *kbps* of downlink traffic.

Fernandes et al. [11] have analyzed the traffic generated by the Second Life (SL) client. They select two Regions which are representative of popular and unpopular places and collect traffic traces for 5 days. They study bandwidth consumption, packet size, and packet inter-arrival time on both the uplink and downlink. Their results show that the average downlink traffic at the client is about 40 times larger than other common NVEs. This reveals SL's higher network cost compared to previous NVEs. In addition, they show that the traffic is highly dependent on avatar behavior.

Kinicki and Claypool [5] have extended the analysis performed in [11], focusing on four Regions selected according to object and avatar density. The main contributions of the paper are twofold. First, they reproduce the results obtained in [11]. Second, they show that SL Regions with high avatar and object density require a bandwidth 10 times larger than "empty" Regions.

In a different approach, La and Michiardi [7] collect data directly from SL servers. They deployed a simple crawler application that monitors avatar behaviors for short periods of time. They show that the distribution of avatar contact-times is similar to that observed in real-world experiments. Although their methodology is similar to ours, their work focuses on comparing avatar mobility to that of humans, rather than performing a comprehensive study of SL.

[3]http://www.ftw.at/ftw/research/projects

3. SECOND LIFE

In this Section we present a general description of SL.

3.1 Virtual World Description

The virtual world of SL is composed of *Regions*, which are independent lands of 256x256 meters. Each Region has a maximum of four adjacent Regions, and can be either *public* or *private*. The former are owned by Linden Lab, while the latter are purchased by individuals or companies. Owners of private Regions have total control on their virtual land. They can, for instance, limit access to a selected set of avatars. Both types of Regions run on Linden Lab servers called "Simulators".

The appearance of a Region is defined by the objects it contains. Each Region has a specific policy on object creation and destruction. For instance, "Sandbox" Regions are used by avatars to test new objects, which are automatically destroyed shortly after their creation.

SL also provides a *map* which contains a small visual representation of all Regions, both public and private. The map, browsable after an avatar logs in, shows how many avatars are connected to each Region by displaying points located at the avatar coordinates. Avatar identities are not shown on the map.

3.2 Client/Server Architecture

SL is based on a Client/Server architecture. Since our crawler interacts with SL servers, we now shortly describe the server-side of SL. A complete description of the SL architecture can be found in [11].

Login Server - It is the entry point in SL, and handles username and password verifications. The Login Server is also responsible of granting or denying access to the Regions (e.g., access may be denied during failures or maintenance operations). It maintains the following statistics: number of connected users and number of logins in the last 24 hours.

Simulator - It is the server responsible for a given Region. Its main role is to maintain the state of its Region, and compute the information about objects and land features that needs to be transmitted to the clients. It also manages chat among avatars located within the Region. A Simulator handles a maximum of 100 avatars[4]. For clarity, we will refer to Simulators simply as *servers*.

We do not know the total number of servers in SL, or whether they employ load balancing techniques. Moreover, SL does not mention the usage of protective measures against Denial of Services attacks and crawling operations.

3.3 Avatar Capabilities

Each user is associated to an avatar by registering at the SL website[1]. This registration requires filling out an on-line form with private information and a valid e-mail address. We now give a short description of the avatar capabilities which are used by our crawler to monitor SL.

A user entering SL must perform a login procedure. After authentication, its avatar joins the virtual world. The Region where the avatar appears is either specified in the login request or derived from the avatar coordinates at its last connection. An avatar can walk, run and fly within a Region. It can also directly move to adjacent Regions provided they are public. It is also possible to perform a *teleport* operation to rapidly cover large distances. The target destination of the teleport can be within either the same Region, or any other Region selected from the map.

Avatars have a view of the virtual world, (i.e., land, avatars and objects), within a limited *visibility area*. This area corresponds roughly to a sphere with radius equal to 35 meters. When an avatar teleports to a Region, the server informs it about the locations and identifiers of all objects on the Region. In the following, we refer to this event as "initialization phase". Finally, an avatar can request several Region statistics to a server. A complete description of these statistics can be found at *wiki.secondlife.com*.

Automated avatars are called *bots*. Region owners often use bots to show a non-zero population in their Regions or simply to welcome visitors. In order to prevent the usage of bots, SL disconnects avatars when they have not moved during the last 15 minutes. However, simple scripts allow bots to perform repetitive actions, such as short walks, which is enough to circumvent SL's bot detection mechanism. Thus, we can assume that avatars which barely move and remain connected for a long time are most likely bots.

4. METHODOLOGY

In this Section we describe our crawler, its functionalities and limitations. Then, we explain how the trace collection was performed.

[4]http://en.wikipedia.org/wiki/Real_estate_(Second_Life)

4.1 Crawler Description

The main idea behind our crawler is to exploit standard avatar capabilities to obtain information about the virtual world. Our crawler is basically a modified SL client which we developed using libsecondlife[2], a set of $C\#$ libraries that allow third party applications to interact with SL. This modified client must be associated to an avatar registered on the SL website in order to be able to log in to the virtual world. While our client logs in to SL to collect information, it does not interact with other avatars.

Our crawler is composed of multiple *subcrawlers*, each specialized in a different monitoring task (see Figure 1). The reasons for this are twofold. First, with this approach, different types of information can be collected using different crawling techniques. Second, splitting the crawling into different tasks allows us to control the temporal resolution of each type of information that we collect. For instance, tracing the movement of avatars requires sampling their position very frequently (e.g., every 30 seconds), whereas determining the total number of objects in the system can be done much less often (e.g., once per day). We use multiple instances of each subcrawler in order to speed up the crawl, each instance being associated with a unique avatar identity[5].

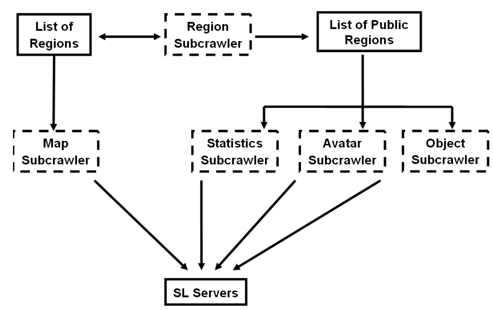


Figure 1: Architecture of the SL crawler

We now describe each subcrawler, including its goals, crawling technique and relationships with other subcrawlers.

The *Region Subcrawler* monitors SL to maintain an up-to-date list of Regions. This information is dynamically updated as new Regions are continuously created. Region discovery is performed via a random walk among adjacent Regions (we use a list of Regions obtained at *stats.slbuzz.com* as bootstrap). The Region Subcrawler teleports to each Region in the list to retrieve the set of adjacent Regions. As new Regions are discovered they are added to the list. In addition, Region accessibility is verified to determine whether a Region is public or private. The list of public Regions is then used by the Statistics, Avatar and Object Subcrawler, while the complete list (public and private Regions) is used by the Map Subcrawler.

The *Object Subcrawler* tracks the evolution of objects in all public Regions. It teleports to a public Region and

[5]For our experiments, we created about 300 different avatars.

accomplishes the initialization phase, during which it is informed by the server of the coordinates and identifiers of all objects on the Region. Then, it dumps this information and teleports to a new Region.

The *Statistics Subcrawler* collects the statistics maintained by public Region servers. It teleports to a public Region, queries its server, dumps the results, and then moves to another public Region. We collect the following server statistics: number of connected avatars, time dilation (i.e., a measure of the load on the server) and total number of packets going in and out from the server.

The *Map Subcrawler* monitors the location of avatars as shown on the SL map. For each Region, the subcrawler locates it on the map and collects the coordinates of all the avatars currently connected to it. This task only requires logging in to SL. Unfortunately, the Map Subcrawler cannot identify the avatar identities as they are not shown on the map. This limitation led us to develop the Avatar Subcrawler.

The *Avatar Subcrawler* obtains the identity and position of the avatars connected to public Regions. First, it uses the map to determine the position of avatars within a Region. Then, it teleports to each of these coordinates to obtain the identities of nearby avatars and to determine their coordinates with greater accuracy. Given the limited visibility area of an avatar, the Avatar Subcrawler may need to teleport several times in order to crawl the entire Region.

Note that the Avatar, Statistics and Map Subcrawler collect redundant information. We exploit this redundancy to check the correctness of our traces.

4.2 Problems and Limitations

During the deployment of our crawler we encountered multiple problems, which we had to solve in order to be able to collect data in a scalable and accurate way.

4.2.1 Crawling Performance

We refer as *crawling performance* to the number of Regions a subcrawler monitors in a given time. As mentioned in Section 4.1, we run multiple instances of each subcrawler in parallel in order to increase the crawling performance. However, we observed that increasing the number of instances does not always improve performance. Figure 2 shows that the performance of the Statistics Subcrawler degrades beyond 60 concurrent instances. This behavior suggests that SL employs a rate-limiting policy against IP addresses which generate a large amount of traffic.

4.2.2 Experimental Hazards

SL does not officially mention banning policies against avatars with unusual behaviors. However, many of the avatars associated to our subcrawlers were banned. The banning procedure consists in an exclusion from SL due to "account verification". While we could not identify the exact behavior that causes banning, we found that a heavy usage of the teleport increases the chances of being banned.

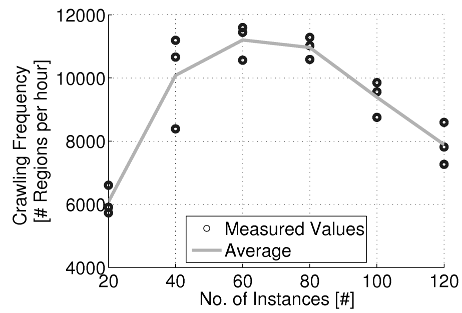


Figure 2: Crawling Performance [Statistic Subcrawler ; 18 hrs experience]

If a banned avatar attempts to login several times, its IP address will be blacklisted. To avoid this, our subcrawler detects when it has been banned and automatically replaces the associated avatar with a new one.

Finally, we also observed a high degree of instability in SL. During a one-month period, the service was down multiple times due to maintenance, server updates or crashes[6]. While our short traces were mostly unaffected, this had an impact on our long-term ones (see gap in Figure 6).

4.2.3 Teleport (In)Efficiency

The Avatar Subcrawler relies on the teleport to collect avatar identities. However, we observed that a teleport can fail when the coordinates of its destination lie inside an object, e.g., a building, thus limiting the completeness of the traces. In order to quantify this crawling error, we compared the number of avatars observed by the Avatar Subcrawler with those shown on the SL map during a 12-hour period. The resulting comparison (not shown for space reasons) indicates that the Avatar Subcrawler correctly identifies all avatars in only 50% of the Regions.

In addition, our original Avatar Subcrawler suffered from heavy banning due to a high number of teleports. We solved this by assigning a unique IP address and subcrawler instance to a given Region, and modifying this assignment every hour. This avoids banning, but requires a number of IP addresses and avatars that is linear with the number of Regions, limiting crawling scalability. For these reasons, we ran the Avatar Subcrawler on a few very popular Regions, whose object composition does not impact the teleport.

4.3 Data Collection

We gather data from several sources. First, we collect traces using our crawler. Second, SL provides official information through their website and Login Server. When possible, we compare our crawler results with these two sources to check their consistency. Finally, we use visual inspection to confirm some of our observations and interpretations. In the remainder of this Section we describe the configuration that we used for each subcrawler.

[6]http://blog.secondlife.com/category/announcements-news/

As mentioned in Section 4.1, each subcrawler collects data at different time resolutions. In addition, some subcrawlers can traverse a set of Regions much faster than others, according to the technique they use to collect information. We call *crawling frequency*, the frequency at which a subcrawler completely monitors a set of target Regions. Finally, note that some subcrawlers require more resources than others (e.g., IP addresses, avatars), and are therefore executed for shorter periods of time.

We monitored the evolution of Regions and objects during 28 days with a crawling frequency of 24 hours. We used respectively 3 instances of the Region Subcrawler and 5 instances of the Object Subcrawler. Traces were collected between March 29 and April 25, 2008, except for April 4 and 5 during which the SL service was down[7].

We ran 60 concurrent instances of the Statistic Subcrawler, as this yields the highest crawling performance (see Figure 2). With this configuration, it can crawl about 11000 Regions in one hour. On March 29, the Region Subcrawler identified 12765 public Regions, so we set the crawling frequency to 90 minutes to be able to monitor all public Regions with a safe time margin. Traces were collected for 6 days between March 29 and April 4, 2008.

We monitored the SL map with 40 instances of the Map Subcrawler and a crawling frequency of 15 minutes. Traces were collected between April 18 and April 21, 2008. A list of 17526 Regions was used, i.e., those identified by the Region Subcrawler on April 18.

Finally, we performed a fine-grain monitoring of a few Regions. We combined in a single crawler the functionalities of the Statistics, Map and Avatar Subcrawler (we refer to it as Stat/Map/Av Subcrawler). Given the limitations of the Avatar Subcrawler (see Section 4.2.3), we selected 5 highly popular Regions where the Avatar Subcrawler achieves a 100% accuracy. We used 5 Planetlab machines and a crawling frequency of 30 seconds. Traces were collected during 3 days between May 01 and May 04, 2008.

Table 1 summarizes our crawler's configuration and trace length.

Subcrawler	Instances	IPs	Regions	Frequency	Days
Region	3	1	-	1/24 hrs	28
Object	5	1	-	1/24 hrs	28
Statistics	60	1	12765	1/90 min	6
Map	40	1	17526	1/15 min	3
Stat/Map/Av	5	5	5	1/30 sec	3

Table 1: Second Life Crawling Summary

5. MULTI-LEVEL ANALYSIS OF SECOND LIFE

In this Section we analyze the traces collected by our crawler. First, we measure some global characteristics, such as the number of unique Regions, online users, and aggregate server traffic. We then focus on objects and avatars, analyzing their spatial and temporal distribution across all Regions. Finally, we select five popular Regions and perform a fine-grain analysis of their activity.

[7]http://blog.secondlife.com

5.1 A Global View

We now analyze some system-wide characteristics of SL, and compare our crawler measurements with official SL figures.

5.1.1 Regions

Table 2 summarizes the total number of Regions discovered by our crawler, as well as the official number reported by the SL website.

We observe that our crawler discovered a larger number of Regions compared to official figures. These additional Regions are not reachable and were discovered as adjacent of active ones. Therefore, they are probably a fraction of the virtual world reserved for future customers, and thus do not count in the official statistics.

	March 29	April 18	April 25
Public Regions (RS)	12765	13220	13261
Total Regions (RS)	17280	17526	17573
Total Regions (SL)	13693	N/A	14150

Table 2: Number of Regions in SL (RS=Region Subcrawler, SL=Second Life website)

The 6-day trace collected by the Statistics Subcrawler shows that many Regions experienced periods of unavailability. There are two possible causes for this: (i) the Region server was down, (ii) the maximum allowed number of avatars had been reached.

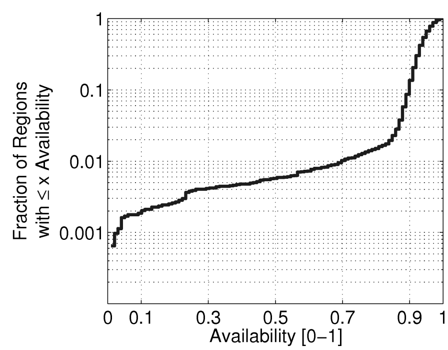


Figure 3: CDF of Region availability [Statistics Subcrawler]

We compute the server *availability* as the probability that a server accepts a connection from the Statistics Subcrawler, i.e., the number of times it accepts a connection divided by the total number of connection attempts during 6 days. We recall that the Statistics Subcrawler makes at least one connection attempt to each Region every 90 minutes. Figure 3 shows the Cumulative Distribution Function (CDF) of Region availability. We observe that 90% of the Regions have an availability of 0.9 or more, but only 1% show a high availability of 0.99 or higher. This is probably due to short maintenance operations or failures. The bottom 1% of the Regions had an availability of 0.7 or lower. An analysis of our traces shows that these highly unstable Regions are not among the most popular (see Section 5.3), suggesting that their periods of unavailability are due to server faults rather than the maximum allowed population (100 avatars) being reached.

5.1.2 Users

Figure 4 shows the evolution over time of the number of on-line users, as measured by the Map Subcrawler and reported by the Login Server[8]. Both curves show the same daily cycle. However, the Login Server reports 10000-20000 more users than the Map Subcrawler. Moreover, during a major SL outage on Friday at 14:00, the Login Server reported a drop of 10000 avatars, while our Map Subcrawler observed a decrease of 20000 avatars. This suggests that the values provided by the Login Server may be inaccurate, and are probably averaged over several minutes.

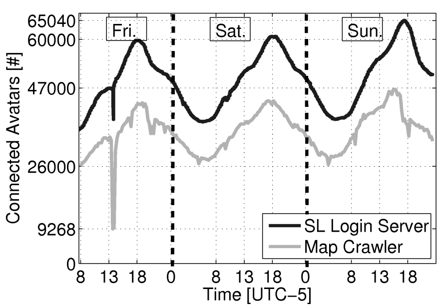


Figure 4: Active population over time [Map Subcrawler; SL Login Server; Coordinated Universal Time - 5]

5.1.3 Server Traffic

The Statistics Subcrawler collects information about the rate of outgoing server packets as reported by SL servers. Although not shown for space reasons, the curve of the aggregate traffic generated by all servers shows a daily cycle ranging from 1.7 to 3.2 million packets per second. Moreover, the traffic's daily cycle closely follows that of Figure 4, which is not surprising for a Client/Server architecture. In fact, the correlation coefficient between the number of avatars in a Region and the traffic volume is 0.8.

Knowing that the average packet size in SL is 500 bytes [11], the aggregate traffic generated by all servers at peak time can be estimated at around 13 $Gbps$. The peak number of users is 47000 measured on Sunday at 18:00, which yields an average bandwidth consumption of 280 $kbps$ per client. This confirms the results reported in [11], and shows the high network cost of the SL service. Even worse, this cost occurs when only 47000 out of the 13 millions registered users is active.

[8]This data is obtained by monitoring the Login Server at http://secondlife.com/app/login/

5.2 Object Distribution and Dynamics

We now analyze the 28-day trace collected by the Object Subcrawler with a crawling frequency of 24 hours.

5.2.1 Object Distribution

We identified about 7 million unique user-generated objects across all public Regions. Figure 5 shows the CCDF of the object distribution. Since the distribution did not significantly change over 4 weeks, we only plot the data for the initial (March 29) and final distribution (April 25). Notice that 30% of the Regions are almost empty, containing less than 100 objects. Around 65% of the Regions contain a relatively low object count, between 100 to 1000, while only 5% of the Regions show 1000 objects or more. The richest Region contains nearly 13000 objects. We note that the Object Subcrawler does not retrieve the size of an object as this would take too much time and make the crawl very slow. Therefore, it is possible that some Regions with a low object count actually have a more complex environment (e.g., bigger objects) compared to other Regions with a larger object count.

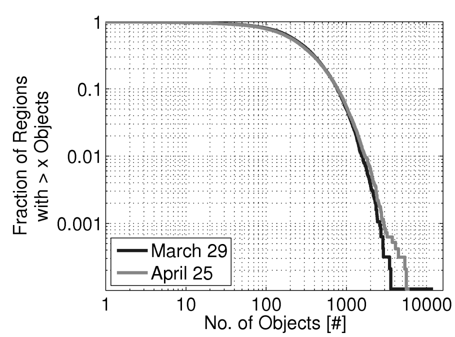


Figure 5: CCDF of the Object distribution across Regions [Object Subcrawler]

5.2.2 Object Dynamics

We now analyze the evolution of the number of objects over time. For each Region, we compute the difference between the number of objects it contains at day i, and its initial object count observed at day 1, i.e., the first day of the trace. Figure 6 shows some significant percentiles of the distribution of these differences measured for all Regions (the gap between days 6 and 9 is due to a SL outage). We observe that 50% of the Regions (between the 25th and 75th percentiles) are almost completely static, showing a small amplitude of variation between -50 and +50 objects after 28 days. The 10th and 90th percentiles remain between -250 and +200 objects, showing modest object variation rates in most Regions. The median value is nearly zero, and the percentiles are almost symmetrical, indicating a similar object creation and destruction rate. In fact, our traces show that the total number of objects in SL remains approximately constant over time. Notice the presence of two drops between days 20-25 and 25-27. During these days, the SL website reported that their servers were being updated. Thus, we believe that these drops correspond to objects being lost and then recovered during server updates. Finally, although not shown in the Figure, we observed minimum and maximum variations close to -4000 and +4000 objects respectively. This implies that the Regions at the bottom and top 10% of the distribution show a highly unstable behavior, with a large number of objects being continuously created and destroyed. These are mostly Regions where users test their objects (e.g., Sandbox Regions), and which typically erase objects soon after they are created.

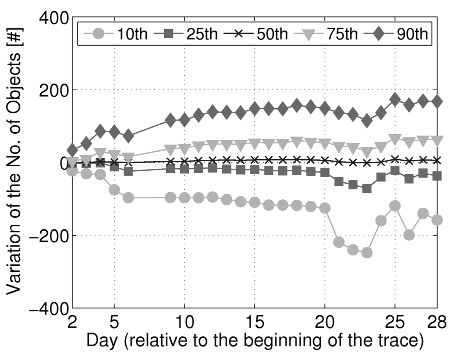


Figure 6: Distribution of the variation of the number of objects per Region [Object Subcrawler]

5.3 Region Popularity

We now analyze the popularity of Regions in terms of the number of avatars that visit them. We use the 6-day trace collected by the Statistic Subcrawler.

Figure 7 shows the CCDF of the number of avatars per Region, measured every 90 minutes for all Regions. Therefore, it represents the most likely distribution of avatars across Regions. Notice that 45% of the Regions are empty, while only 2% have more than 20 avatars. Despite the limit of 100 avatars set by SL, we see a clear cut-off at about 90 avatars. In fact, only in a few cases does the population approach the limit of 100 avatars per Region.

As the number of avatars in a given Region is highly dynamic, we study for each Region the distribution of the population with time. For each Region we calculate the *Population CDF*, i.e., the Cumulative Distribution Function of the number of avatars observed during the 6-day period. Since we cannot plot the CDFs for the 12765 monitored Regions, we will take a few significative percentiles and plot their distribution among all Regions.

Figure 8 shows the distribution among Regions of the 25th, 50th, 75th, and 100th percentiles of the Population CDFs.

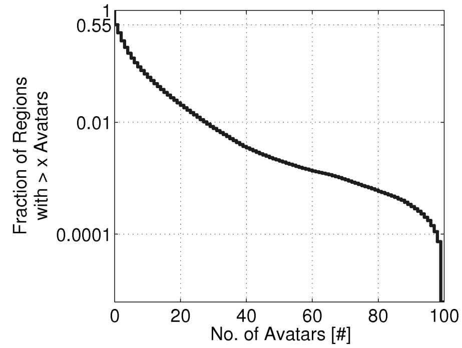


Figure 7: CCDF of Avatar distribution across Regions [Statistics Subcrawler]

Note that the 100th percentile is simply the maximum population observed for a given Region. Similarly, the 75th percentile may be interpreted as a peak population, the 50th percentile as a median or typical population, and the 25th percentile as a residual population. Accordingly, Figure 8 shows that 30% of the Regions are empty all the time, while around 45% of the Regions have always less than 5 avatars. The 75th percentiles curve overlaps with the 100th percentile between 0 and 4 avatars. As a consequence, Regions whose population is small most of the time have a small population all the time with no exception. Focusing now on larger populations, we observe that around 5% of Regions have at least 30 avatars as a maximum population, but that this number drops to 18 avatars for a peak population (75th percentile) and to 12 avatars for a typical population (50th percentile). Hence, although a non-negligible number of Regions are occasionally densely populated, they usually contain few avatars. Most of the Regions have a small residual population (25th percentile), e.g., it is at most 3 for 80% of them. This indicates that most Regions are occasionally almost empty. Only a small portion of Regions (around 3%) have a significant residual population (more than 10 avatars), showing that Regions with continuous activity are rare. These results indicate that SL Regions differ in many ways, and may necessitate different resource provisioning according to their popularity profile.

5.4 Virtual Groups

In this Section we focus on the spatial distribution of avatars inside a Region. We are interested in determining to what degree avatars concentrate in groups where each avatar is within visibility range from each other (35 meters, as defined by SL). The rationale is that avatars located within such *virtual groups* are highly likely to interact with each other. The results we present in this Section are obtained from the analysis of the 3-day trace collected by the Map Subcrawler with a crawling frequency of 15 minutes.

We estimate the number of virtual groups in a Region by using the *k-means* clustering algorithm [1] to partition avatars in circles of radius $r \leq 35$ meters. We proceed as

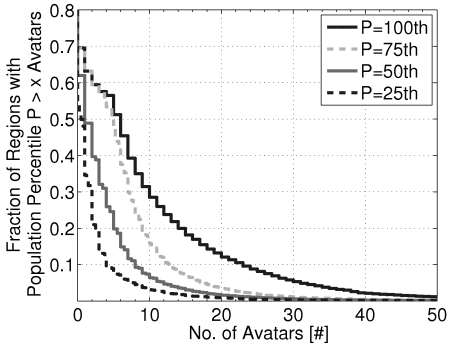


Figure 8: CCDF of 25th, 50th, 75th, and 100th percentiles of the Population CDFs [Statistics Subcrawler]

follows. Let n be the number of avatars in a Region. The algorithm takes the avatar coordinates $a_i = (x_i, y_i)$ with $1 \leq i \leq n$, and a number of target partitions k. It then clusters the avatars into k circular areas with center coordinates $c_j = (x_j, y_j)$ and radius r_j, where $1 \leq j \leq k$. We run the algorithm iteratively for increasing values of k until all circles have a radius $r_j \leq 35$ meters. The final value of k gives the number of virtual groups in the Region, and c_j the groups' center coordinates.

We use the k-means clustering algorithm since it aims at minimizing the distance of avatars from the center of the virtual group. Note that this algorithm does not track groups which move across the Region. However, we observed that about 90% of the time avatars do not move (see Section 5.5), so this limitation only has a minor impact on our clustering scheme.

5.4.1 Virtual Group Sizes

Figure 9 shows the CDF of virtual group sizes across all Regions. Notice that 50% of the groups are composed of a single avatar. These are either bots or isolated users exploring a Region. Surprisingly, 45% of the virtual groups are made of only 2-10 avatars. This could be explained by the *budget of attention* theory [8], which suggests that human beings can only focus their attention to a maximum of 5-9 objects at the same time. Finally, we observe a negligible number of virtual groups larger than 20 avatars. We verified by visual inspection that these large avatar groups are driven by the presence of an event, such as concerts and shows. Note that large avatar crowds which extend beyond 35 meters may be split by our algorithm into smaller groups. This is consistent with our goal of identifying groups of avatars which are interacting (an avatar cannot interact with someone outside its visibility area).

5.4.2 Points of Interest (POIs)

Regions usually contain Points-of-Interest (POIs), i.e., spots which attract several avatars. In order to detect the presence of POIs, we look for virtual groups which are stable with respect to time and location. Therefore, we use a metric which

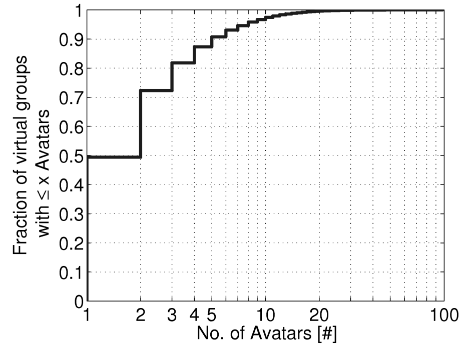


Figure 9: CDF of virtual group sizes [Map Subcrawler]

we call the group's *spot lifetime*, which is defined as follows. For every new group, we record its initial center coordinates. Since the group may move or dissolve, we compute its spot lifetime as the time elapsed from its creation until we observe no virtual groups centered within 35 meters (the visibility area) from its initial center coordinates. Moreover, if the center of a group moves more than 35 meters from its original position, we consider that a new group is observed at the new center coordinates.

Figure 11 shows the CDF of spot lifetimes for all groups and for different ranges of S, the average group size. We notice that groups with large sizes tend to exist longer. This suggests the presence of POIs near the center of groups with high spot lifetimes. We also observe that 50% of large virtual groups ($S > 10$) have a rather short spot lifetime. We interpret this as event-driven groups, i.e., located near short-lived POIs. Conversely, the remaining 50% have a very long spot lifetime. Intuitively, the area around popular POIs is unlikely to become empty, especially in popular Regions, resulting in very long spot lifetimes.

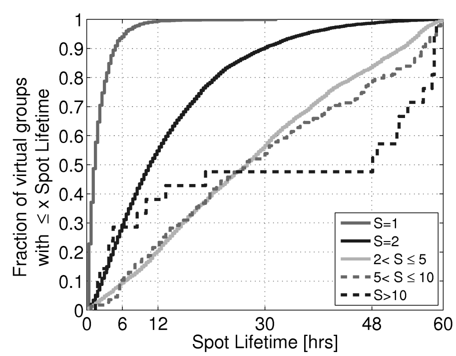
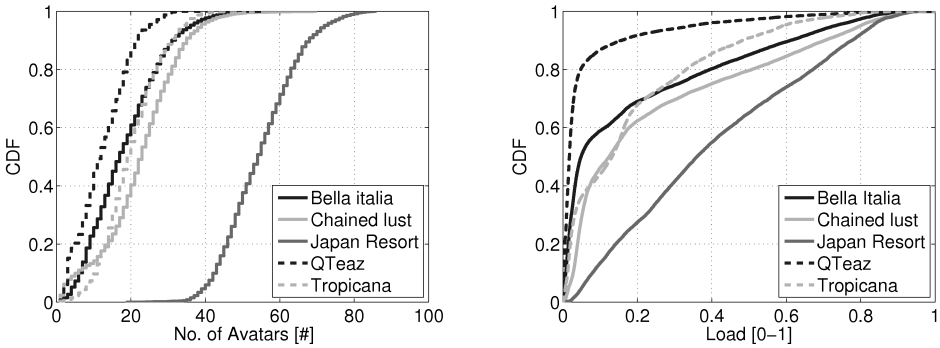


Figure 11: CDF of spot lifetimes [Map Subcrawler]

Figure 11 also provides some interesting insight on isolated avatars, corresponding to the curve of single-avatar groups ($S = 1$). Around 40% of these avatars have a spot lifetime of a few minutes, and can be identified with avatars exploring a Region. Thus, the locations traversed by these avatars are unlikely to be a POI. However, 10% have a lifetime between



(a) Population (b) Server load

Figure 10: Population and Server Load CDF [Stat/Map/Av Subcrawler]

5 and 32 hours, i.e., they stay at the same spot for a very long time. It is unlikely that this behavior is coming from human beings, so we identify these avatars as bots. Given that 50% of the virtual groups are composed by a single avatar (see Figure 9), we conclude that at least 5% of the entire population in SL is likely identifiable as bots.

5.5 Analysis of the most Popular Regions

We select 5 Regions among the 20 most popular ones whose object composition allows the teleport operation to succeed at all times (see Section 4.2.3). The results we present in this Section are obtained from the analysis of the 3-day trace collected by the Stat/Map/Av Subcrawler with a crawling frequency of 30 seconds. We first give a brief description of these Regions.

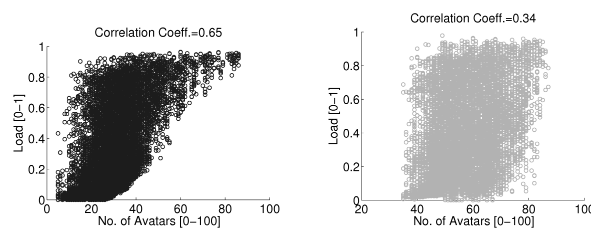
Bella Italia is a meeting place made of a central square with some trees and benches. *Chained Lust* is a shop with adult content. The land contains a single building with lots of objects. Avatars entering the Region appear directly inside the building. *Japan Resort* is an island with few trees and thatched huts. *QTeaz* is a Region dedicated to leisure. It consists of few small buildings with games and other activities. *Tropicana* is a virtual resort. There is a beach, a small lake and several vacation facilities. There are advertisements about music events where avatars can dance and meet.

5.5.1 Region Popularity and Load

Figure 10(a) shows the Population CDF (see Section 5.3) for the five Regions. QTeaz and Japan Resort are respectively the least and most popular Regions, while the other three Regions have a comparable popularity. Notice that all Regions are almost never empty. In addition, Japan Resort has never less than 20 avatars. The active population per Region rarely exceeds 40 concurrent avatars, except for Japan Resort, whose peak population is 84 avatars.

In SL, servers "slow down" virtual time as a way to cope with high loads. This is called time dilation (td), and is defined as follows: $td = 1$ means that the server is running at full speed, whereas $td = 0.5$ means that it is running at half-speed. We consider $(1 - td)$ as a measure of load on a server, e.g., $td = 0$ means maximum load and $td = 1$ means minimum load. Figure 10(b) plots the CDF of the load per Region. As expected, the more a Region is popular, the more its server is loaded. We can see that half of the time, Japan Resort has a load larger than 0.35. Interestingly, we observe that both Japan Resort and Chained Lust exhibit very high load values (e.g., larger than 0.8) despite the significant difference in their Population CDFs.

We now analyze the impact of avatar population on the server load. Figure 12 shows a scatterplot of the number of avatars and the server load. We only plot the data for Chained Lust and Japan Resort since they are the most representative.



(a) Chained Lust (b) Japan Resort

Figure 12: Scatterplot of the No. of Avatars and Server Load [Stat/Map/Av Subcrawler]

Figure 12(a) shows a weak positive correlation of 0.6, and the presence of three different trends. When the population in the Region is lower than 20 avatars, the server is generally lightly loaded. In the range between 20 and 50 avatars, the server load and number of avatars seem completely uncorrelated. Finally, when the population grows over 50 avatars the server is always very loaded, as we never observe load values smaller than 0.6. The trends we highlighted in Figure 12(a) are representative of all Regions, with the exception of Japan

Resort (see Figure 12(b)). This Region shows a lower correlation coefficient of 0.34. In addition, we observe a general trend similar to Chained Lust, but starting at around twice the population size. This result suggests that Japan Resort is assigned more server resources than other Regions.

5.5.2 Avatar

Avatars join and leave a Region multiple times. We use the term *session* to denote the time an avatar spends in a Region. Note that a session does not span multiple Regions, as we cannot detect whether avatars leaving a Region are moving to other Regions or leaving SL. Moreover, we will assume that each user is associated with a unique avatar. Figure 13 shows the CDF of user session times for each Region. Despite the different Regions popularity, users spend roughly the same time in each Region. 50% of the users stay connected less than 10 minutes per session while 15% stay connected 100 minutes or more. Finally, 5% stay connected more than 10 hours. An analysis of the movement of these 5% shows that 98% of the time they do not change their position at all, suggesting that they are bots.

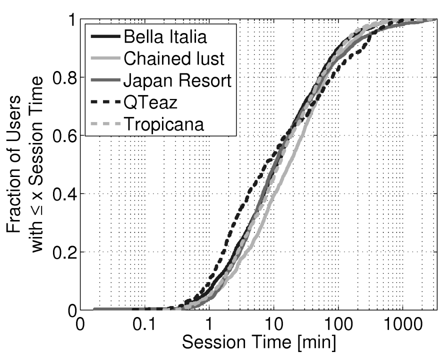


Figure 13: CDF of User Session Times [Stat/Map/Av Subcrawler]

Figure 14 illustrates an analysis of avatar movement patterns. We distinguish between standing, walking, running, flying and teleporting according to the avatar's speed. Surprisingly, avatars stand on the same point more than 80% of the time. The remaining time they are mostly teleporting or walking (flying and running only account for negligible values). This highly static behavior is probably due to two factors. First, in popular Regions avatars spend most of their time chatting with nearby avatars. Second, these Regions experience high server load (see Figure 12), which introduces lag and thus makes avatar movement more difficult.

5.5.3 Virtual Groups

We now analyze the interaction between an avatar and virtual groups. Figure 15 shows the CDF of avatar visits to the same virtual group during 3 days. About 50% of the avatars come back at least once in 3 days to a previously visited group, while 30% revisit the same group at least once per day. We conclude that there exists a high level of predictability in avatar behavior. This derives from the social

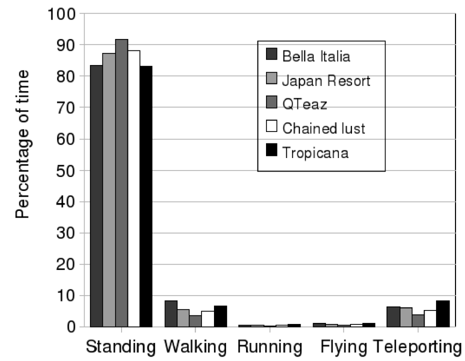


Figure 14: Avatar movement patterns [Stat/Map/Av Subcrawler]

component of SL: avatars are attracted to places they like, or where they can meet avatars they already know or who share similar interests.

We now analyze how much time avatars spend close to each other. We say that avatars *meet* whenever they are within their visibility area. Intuitively, the number of times that two given avatars meet gives an idea of the degree of relationship between them.

In order to simplify our analysis, we define three types of avatar relationships. We say that two avatars are *Acquaintances* when they meet only once. They are *Friends* when they meet more than once and in less than 50% of the sessions. Finally, two avatars are *Good Friends* if they meet in 50% or more of their sessions.

Figure 16 shows respectively the median, average and maximum number of Acquaintances, Friends and Good Friends per avatar. As expected, the Figure shows that avatars tend to have a large number of Acquaintances, and much fewer Friends and Good Friends. In particular, more popular Regions seem to produce more Acquaintances. This is probably due to the presence of a larger number of visitors, i.e., avatars connecting only once. Note that avatars with long session times, such as bots, may encounter a high number of these visitors, which explains the extremely high maximum values of Acquaintances (e.g., 800 for Japan Resort).

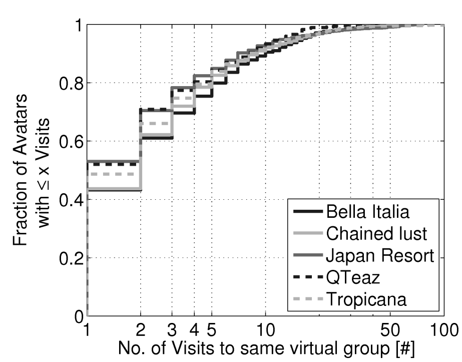


Figure 15: CDF of avatar visits to the same virtual group [Stat/Map/Av Subcrawler]

We also observe that 50% of the avatars have some Good Friends, while the same fraction of avatars has zero Friends, which may seem counterintuitive. In fact, there are two reasons behind this. First, the social component of SL favors strong interaction between avatars. Second, many avatars visit a Region only a few times during our trace period. This low number of sessions favors Good Friends over Friends. For instance, two avatars who connect only twice need to meet only once to be considered as Good Friends. The high maximum number of Good Friends may also be explained by the presence of bots. The median number of Good Friends (i.e., 1 to 4) may be interpreted as the typical number of avatars which are strongly socially connected (e.g., "first life" friends).

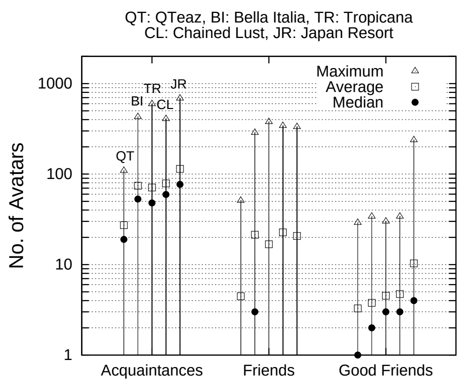


Figure 16: Median, Average and Maximum number of Acquaintances, Friends and Good Friends per avatar [Stat/Map/Av Subcrawler]

6. GUIDELINES FOR A NEXT GENERATION SECOND LIFE

As discussed in Section 5.1.3, SL generates large amounts of traffic. Next generation virtual worlds are expected to be more complex, and thus may require even higher bandwidths. As the system scales in number of users and bandwidth, the classic Client/Server (C/S) architecture will clearly show its limitations.

In this Section we review our observations and propose some solutions in order to enhance SL's design. Note that our suggestions are motivated by the observed characteristics of the virtual world, not by the specific implementation of the SL protocol. Therefore, they may be useful for other Networked Virtual Environments (NVEs) that show object and avatar characteristics similar to SL.

6.1 Enhancements to SL

We now describe two techniques, caching and prefetching, that can be useful to improve SL's design.

6.1.1 Caching

We have observed that the object composition of SL Regions does not change much over time (see Figure 6). In addition, avatars tend to visit multiple times the same virtual places in different sessions (see Figure 15). This means that it is highly likely that a server sends duplicate objects to a client across different sessions.

A client-side caching system could be used to reduce server traffic. Whenever an avatar visits a new virtual place, the client caches the objects received from the server. If cache space is limited, longer lifetime objects may be given higher cache priority. When the avatar visits the same place again, the server only needs to transmit data corresponding to new or modified objects. This simple scheme could reduce both server traffic and object discovery latency at the client.

6.1.2 Prefetching

We can easily identify in each Region several Points-of-Interest (POIs) which avatars are most likely to visit. These POIs consist of locations associated to virtual groups with very long spot lifetimes (see Figure 11). This implies that objects located close to these POIs are very likely to be transmitted from the server to the clients.

Based on this observation, the server could use the information it collects about POIs to predict avatar behavior. When an avatar enters a Region, the server first transmits the data about its immediate surroundings. Whenever free server bandwidth is available, the server also transmits data about nearby POIs. In this way, avatars moving toward a POI may have already downloaded the objects they need before reaching it, decreasing rendering latency.

6.2 Hybrid P2P and Client/Server Architecture

In a pure Peer-to-Peer (P2P) social virtual world, each user contributes its hardware resources to sustain the system. The main advantage of this solution is scalability and low cost, as most hardware resources are effectively contributed for free. The main difficulty is that efficient and robust P2P services are hard to design, as many problems not found in C/S systems, such as churn-tolerance, overlay construction, and security need to be addressed. Nevertheless, many of these difficulties can be lessened or completely avoided by adopting a hybrid P2P and C/S architecture, while still retaining many advantages of pure P2P architectures.

In this Section we argue that given the observations presented in this study, P2P could be particularly useful in systems implementing social virtual worlds. We will limit our discussion to a hybrid P2P and C/S architecture, in which the main goal is to offload the server via a direct communication among clients. Thus, we will assume that the server remains a vital component of the system, acting as a trusted authority (e.g., to guarantee safe economic transactions), and ensuring the persistency of Regions and objects.

6.2.1 Distributed Caching

The social aspect of SL encourages users to spend their time together. We found that about 50% of the avatars have at least one *Good Friend* (see Section 5.5.3). This means that there is a high chance that every time an avatar connects to SL it will interact with a set of avatars it has repeatedly encountered before. This suggests that the caching system mentioned in Section 6.1.2 can be enhanced by taking into account the presence of highly synchronized avatars.

The main idea is to build a distributed cache using the information provided by the social network, i.e., that some avatars meet frequently in the virtual world. Therefore, avatars could first attempt to download data from their friends using the P2P network, and only resort to contacting the server when no friends are available. Moreover, having each friend store a different piece of content would reduce the size of the client's cache size.

6.2.2 P2P Avatar State Management

In a virtual world each avatar is characterized by a state, such as its position, appearance and actions. When avatars interact, the information about their state has to be propagated to nearby avatars. Thus, the server has two main functions: 1) transmit data about the land and objects to avatars, 2) relay avatar state among nearby avatars. Clearly, the server traffic generated by the latter can be reduced or eliminated if nearby avatars are allowed to communicate with each other in a P2P fashion.

In the past few years, several P2P schemes have been proposed [4] [6] in which clients connect to other clients whose avatars are close in the virtual coordinate space. However, these solutions perform poorly when large virtual groups are present, or when avatars move quickly [2]. Our analysis shows that avatars stand at the same place 90% of the time (see Figure 14). In addition, they organize in small groups of 2-10 avatars (see Figure 9). Thus, our study provides evidence that existing P2P solutions can be effective for the management of avatar interactions in SL.

7. CONCLUSIONS AND FUTURE WORK

Second Life (SL) has received a lot of press coverage and even some major companies and governments have set up a presence on it. The one figure that is usually cited as an indication of the raving success of SL is its more than 10 million registered avatars.

We have carried out a detailed evaluation of a large portion of SL, and made some interesting observations. Almost 30% of the Regions do not attract any visitors, and only few Regions are quite popular. Moreover, the number of concurrent participants barely reaches 50,000. In comparison, World of Warcraft, a popular multiplayer on-line game, reaches peaks of one million concurrent players. So one is tempted to paraphrase the famous American comedian W. C. Fields saying "I went to Second Life and it was closed". We also find that avatars exhibit a behavior that very much resembles that of humans: they get together in popular places, where they frequently meet their friends.

From a systems perspective, we observe that SL shows very poor scalability. The number of avatars per Region is limited to 100. Even worse, servers experience peaks of high load with as little as 20 avatars, and become clearly overloaded beyond 50 avatars. This decreases user experience and limits social interaction in the virtual world.

Based on our observations, we suggest several mechanisms to improve the next generation of social virtual worlds. We discuss the applicability of a caching system to reduce the server traffic, as well as a prefetching algorithm to improve the object discovery latency experienced by the client. In addition, the static behavior of avatars suggests that a hybrid Peer-to-Peer and Client/Server architecture could be highly efficient in reducing server load and increasing user experience.

We are currently working on the deployment of a SL client which implements a Delaunay Network [3] [9] among avatars located in the same Region. At the same time, we are evaluating the effectiveness of a distributed caching system based on information derived from the social network.

Acknowledgements

The authors would like to thank the University of Mannheim for making available computing and bandwidth resources for this study. We also would like to thank Augustin Chaintreau for his insightful comments and suggestions.

8. REFERENCES

[1] K. Alsabti, S. Ranka, and V. Singh. An Efficient K-Means Clustering Algorithm. 1997.

[2] H. Backhaus and S. Krause. Voronoi-Based Adaptive Scalable Transfer Revisited: Gain and Loss of a Voronoi-Based Peer-to-Peer Approach for MMOG. In *Proc. of Netgames*, Mellbourne, Australia, September 2007.

[3] A. Bowyer. Computing Dirichlet Tessellations. *Computer Journal*, 24(2):162–166, 1981.

[4] S.-Y. Hu, J.-F. Chen, and T.-H. Chen. VON: A Scalable Peer-to-Peer Network for Virtual Environments. *Network, IEEE*, 20(4):22–31, 2006.

[5] K. James and C. Mark. Traffic Analysis of Avatars in Second Life. In *Proc. of Nossdav*, Braunschweig, Germany, May 2008.

[6] J. Keller and G. Simon. SOLIPSIS: A Massively Multi-Participant Virtual World. In *Proc. of PDPTA*, Las Vegas, Nevada, USA, 2003.

[7] C.-A. La and P. Michiardi. Characterizing User Mobility in Second Life. In *Proc. of WOSN*, Seattle, USA, August 2008.

[8] G. A. Miller. The Magical Number Seven, Plus or Minus two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 63:81–97, 1956.

[9] M. Ohnishi, R. Nishide, and S. Ueshima. Incremental Construction of Delaunay Overlay Network for Virtual Collaborative Space. In *Proc of. C5*, Cambridge, MA, USA, January 2005.

[10] J. Pang, F. Uyeda, and J. R. Lorch. Scaling Peer-to-Peer Games in Low-Bandwidth Environments. In *Proc. of IPTPS*, Bellevue, Washington, USA, February 2007.

[11] F. Stenio, K. Carlos, S. Djamel, M. Josilene, and A. Rafael. Traffic Analysis Beyond This World: the Case of Second Life. In *Proc. of Nossdav*, Urbana-Champaign, IL, USA, June 2007.

[12] P. Svoboda, W. Karner, and M. Rupp. Traffic Analysis and Modeling for World of Warcraft. In *Proc. of ICC*, Urbana-Champaign, IL, USA, June 2007.