CS457 Assignment 1

Daniel Burstyn (20206120)

January 30, 2009

Contents

2 Services and Outcomes

2.1 Services

2.1.1 Process update packet locally

The first service, is one that faces the user directly. This service is processing of user's update packets, and making the necessary changes to the local server. This conveys the user's new location to all users that share that local server. Ideally, this when a user sends an update packet, this is the only service that needs to be used, but sometimes the system needs to go one step further.

2.1.2 Update location to remote servers

Occasionally when a user sends an update packet, there are other users in his vision domain that need to be notified, but are not on the local server. In this case, the user's local server needs to send a syn packet to various other remote servers. This is the second service of the system.

2.2 Outcomes

2.2.1 Success

Since the failure of servers is outside the scope of this assignment, success is the only possible outcome of our services.

3 Performance Metrics

3.1 Scalability

The first set of performance metrics are those that measure scalability. Scalability is something that itself is very hard to measure, so we must pick our metrics carefully.

3.1.1 Utilization

Utilization is defined as the proportion of time that a server is busy. The lower the utilization, the more time the server is idle. We want to minimize the time that the server is idle (and thus maximize utilization) because it is not cost effective to purchase servers that sit idle, and for the system to be scalable, we need to minimize cost. This can be considered in both a individual and global sense in terms of when a specific server is busy, and all servers aggregated.

3.1.2 Latency

Latency refers to the time between the user sends his update packet, and when his local server begins processing the request. With many users, the queue on servers can become quite large, causing latency to increase. This is a individual metric because it is specific to each user's local server.

3.1.3 Number of syn packets per update

This is the average number of syn packets that a server has to send when it receives an update packet. This is a global metric because it is averaged across the whole system. The sending of syn packets to other remote servers is the most expensive task to perform for a server. As population grows, so does the density within the VE, and thus the number of syn packets that need to be sent when the average user moves also increases. In order for the system to be scalable, we want to minimize the number of syn packets that need to be sent.

3.2 Consistency

The other aspect of the system that is important to us is user state consistency. There are a number of metrics that help us determine how consistent the system is on average.

3.2.1 Duration of inconsistent data

This is the amount of time that the system is in an inconsistent state. Since this deals with the entire system's state, it is a global metric. Using the same notation as the textbook, this time is $\approx \max(D_{l1}, D_{l2}, \ldots, D_{ln})$ Where l is the local server out of n servers. Creating and sending of syn packets is very simple is fast enough to be negligible in this situation. This is an important metric that we absolutely want to minimize because it affects how realistic the VE interactions are. Delay between users seeing eachothers actions causes laggy interactions that are bad for the user's experience.

3.2.2 Number of remote servers causing inconsistency

This is the number of remote servers that require syn packets upon a user movement so that the system can maintain a consistent state. Again, since this is measured across the whole system, it is thus a global metric. This is an important metric because it is the largest bottleneck in maintaining consistency across the system.

4 System Parameters and Workload Parameters

4.1 System Parameters

4.1.1 Number of servers

It is fairly obvious that the number of servers available to the system is an important parameter and has an impact on a number of performance metrics.

4.1.2 Server capacity

Server capacity is the rate at which a server can process user requests. This affects how many users a server can handle, and thus how soon new servers must be purchased.

4.1.3 Network latency

This includes both the time it takes for a user to send an update packet to their local server (d_{au}), and the time it takes to transmit a syn packet from one server to another (D_{ab}). This is important as it affects how long the system is in an inconsistent state.

4.1.4 Syn service requirement

This is the time it takes for a remote server to service a syn packet and update it's state.

4.1.5 Update service requirement

This is the time it takes for a user's local server to process an update request, including creating and sending off necessary syn packets.

4.2 Workload Parameters

4.2.1 Number of users

The number of users is perhaps the most important parameter because it specifies the scale of the system.

4.2.2 Service requirement of users

This refers to the average service requirement of a user in terms of how often they send update packets, and how often those update packets require syn packets to be sent to remote servers. For example if users tended to all stay close together, a higher proportion of update packets would require syn packets as well.