A Geographic Redirection Service for On-line Games

Chris Chambers Wu-chi Feng Wu-chang Feng
OGI@OHSU
{chambers,wuchi,wuchang}@cse.ogi.edu

Debanjan Saha
IBM Research
dsaha@us.ibm.com

ABSTRACT

For many on-line games, user experience is impacted significantly by network latency. As on-line games and on-line game servers proliferate, the ability to discover and connect to nearby servers is essential for maintaining user satisfaction. In this paper, we present a redirection service for on-line games based on the geographic location of players relative to servers. As our results show, the service better meets client demand, saving each client and the Internet as a whole, thousands of miles of networking inefficiency.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed Applications*
; K.8.0 [**Personal Computing**]: General—*Games*

General Terms

Performance, Management, Measurement

1. INTRODUCTION

Video games are an increasingly popular pastime in the United States, with the 2003 IDSA report estimating that 60% of all Americans play video games [1]. Many of these games are networked and run in some variation of the client-server model. To support the game, a large number of servers are distributed widely throughout the world in the interest of scalability and decreasing latency for players. These on-line video games fall into several categories. Massively multi-player on-line games (MMOGs) such as Everquest or Dark Age of Camelot place clients in a persistent world where they interact with hundreds of other players via their avatar in a fantasy realm. Real-time strategy games pit a few players (2-6) against each other as commanders of armies attempting to wipe out the opposing forces. First person shooters cast a set of players (6-30) as individual soldiers armed with projectile weapons, who attempt to wipe out the opposing soldiers in various locales. While some games, such as MMOGs, rely on centralized servers controlled by the owning corporation (Everquest, Dark Age of Camelot), others genres, such as First Person Shooters, frequently utilize a widely distributed server model. For the most popular FPS at this time, Counter-Strike (a variant of Half-Life), there are more than 30,000 registered servers running at any given moment.

For FPS games, latency is a strong determinant in user satisfaction [2, 3, 4, 5]. Because of this, it is imperative that clients can easily find and connect to servers that are close to them. In order to address this problem, we have designed and implemented a geographic redirection service that quickly and efficiently connects players to nearby servers. Our focus is initially on the game of Counter-Strike due to its popularity, the large number of deployed servers, and the fact that we have access to an extremely popular server for the game.

It is important to note that these Counter-Strike servers are not under control of the parent corporation of the game but are rather governed by individuals, and subject to their whims and schedule. This style of game server architecture is referred to in this paper as public-server architecture.

Players on these public-server games must find a server meeting their game needs in a changing server lansdscape every time they want to play. Latency is of high concern to players, but there are other concerns such as the server's custom rules, map, and existing crowd. The standard method of finding a suitable server to play on is, firstly, to download a list of all of the currently registered servers from the central registry. Secondly, automated process then contacts the servers and retrieves detailed information on the current game being played. The player can then sort this list locally by any criteria and connect to their chosen server. Unfortunately, such mechanisms do not scale, do not work effectively, and consume a significant amount of bandwidth.

To demonstrate the efficacy of our approach, we modified the Counter-Strike game server for an extremely popu-

MM'03 November 2–8, 2003, Berkeley, California, USA.

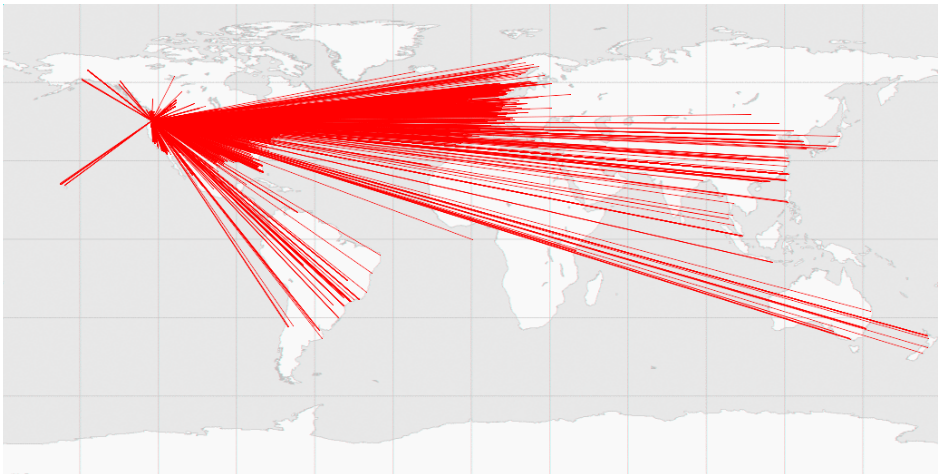

Figure 1: Overflow connections on cs.mshmro.com 6/17/03-6/19/03

lar gaming destination, cs.mshmro.com, so that it transparently redirected players based on their geographic locations. For reasons that are not completely clear, some servers are enormously popular while others languish. Some explanations include network positioning, word of mouth reputation, server rules, or something biased in the black-box algorithm used by Half-Life's "Quick Start" button. cs.mshmro.com is forced to turn away over 2000 people per day simply due to being full. Figure 1 shows a map of the world with a line drawn between cs.mshmro.com and every client who tried to connect but found the server full during a 48 hour period (there are 5400 lines). Ideally, full game servers would be able to redirect potential clients to other servers, perhaps to servers even better suited to the client needs. Our service provides a scalable, centralized, redirection architecture by which servers with high load can, instead of dropping clients when full, redirect the clients to a server that meets client needs, as well as improving the overall connectivity of the network. The management of where these clients go is completely handled by a redirection master, a service locatable anywhere on the Internet. We envision the application of this service to any public-server game, such as Quake, Half-Life, or Neverwinter Nights, as well as any network service that runs on a large number of geographically distributed servers (i.e. a geographic "any-cast").

2. METHODOLOGY

In order to match clients with servers, geographic positioning information is used to locate a server that is on the same continent as the client. As seen in Figure 1, clients often select servers very far from home, even when there are similar servers nearby. Our geopositioning information is obtained from a commercial tool [6], which has a success rate in obtaining GPS data from client IP addresses of over 60% for Counter-Strike traffic to our server [7]. Redirection is not performed on addresses that cannot be mapped, but the mapping tool itself is being updated continuously, presumably increasing its success rate and thereby allowing for broader participation in redirection over time.

In addition to geographic data, we need to know which servers are currently running, which ones are usable, and where they are located. Many current public-server games have a master server which tracks all of the game servers, for licensing purposes as well as to aid players in finding a server, and a variety of tools have been developed to access this master server. We use QStat [8], which can also contact a public server and retrieve its characteristics. We define a server as "good" when it is up, not full, and has game rules that match the game rules we define as important. For this study, we only considered one game rule to be critical, which was the rule allowing play without a password. Surprisingly, around 40% of all randomly selected Counter-Strike servers are protected with a private password. Our goal is to redirect players who cannot play on a server to a likely candidate server in their own region of the world. In order to define geographic regions, we partition the world into a small number of distinct rectangular areas based on eye-balled land mass and server density. The results of this study used 7 such regions, roughly corresponding to the 6 inhabited continents with an extra region for the western coast of the United States, where the test server is located. Regional granularity is important in order to avoid redirecting clients within a region to servers that are significantly further away than their initial choice. For our first pass, however, we strive mainly to decrease the transoceanic traffic. We anticipate defining regions more fine-grained than continents will very probably yield much better results in terms of saved overall bandwidth and saved latency from player to server. The redirection master periodically retrieves a listing of all registered servers and performs geographic lookups on each of them, categorizing each server into a particular region of the world, and storing this information in a database. This is done daily, and serves to capture a rough view of the available registered servers for the day. This process can take up to an hour.

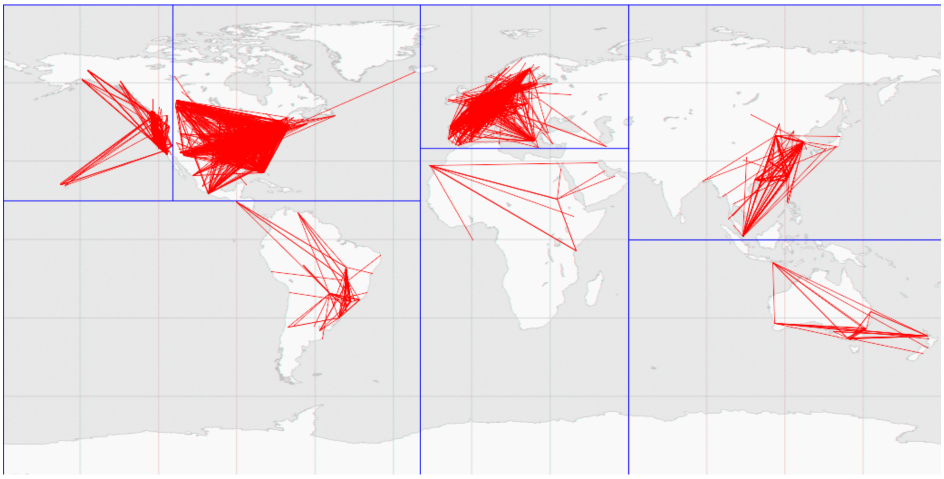

Figure 2: Redirected connections on cs.mshmro.com 6/16/03-6/17/03

In addition to being time consuming, polling the servers once a day gives us little assurance that these servers are still up several hours later, and tells us nothing about how full the server is at any given moment. To address these issues, we frequently (every 5 minutes) select a subset of servers from our large list to be redirect servers, 5 per region, and verify them to be good. These are the servers to which clients are redirected to when our server is full. The frequent verification of this list gives us some measure of confidence that the servers are still good, although it does not establish any guarantees. Reducing the list of servers to just a few selected redirect servers updated every few minutes has several benefits. First, it reduces the amount of polling and processing required by the redirect master. Second, it allows the service to fill up the redirect servers, giving players on those servers close to the same game play that they would have experienced on the original server (they even play with the same people they would have played with if there were room). Finally, it allows for scalability of the service; by measuring the number of redirection requests in the last few time periods, provisioning of adequate redirect servers for the future can be performed. The redirect master does the work of collating and continually verifying the list of redirect servers per region, which means in order for a game server to participate in client redirection, all that is required is the installation of a small server plug-in which contacts the redirect master and receives a redirect server given the client IP address.

3. EVALUATION

The study used two Linux servers, one running Half-Life's Counter-Strike mod on cs.mshmro.com and the redirect plug-in, and the another running the redirect master. The game server's redirection plug-in was written in Small [9], and the management of the redirect master was implemented as a set of perl modules storing data in a MySQL database (the same database with the geographic IP lookup). All data is taken from a few days of redirection traffic from June 17th to June 19th, 2003. Figure 2 graphically shows the locations to which refused players were routed over the 3 day period, as well as explicitly delineating the 7 regions. One metric used to determine the effectiveness of this redirection service is simply the distance saved. Another, more accurate measure of network efficiency is kilobit-miles. As shown in [10], a typical Counter-Strike player utilizes 56kbps of network traffic. If a player was playing over a two mile link, and was rerouted to a server one mile closer, the network would save 56 kilobits per second of traffic over a one mile link, or 56 kilobit-miles. Since converting between kilobit-miles and miles is a constant conversion for Counter-Strike traffic, we simply measure miles saved. It is important to note that these are extra miles of links which gameplay traffic no longer has to traverse, and furthermore these savings have a dimension in time for as long as the player plays. While miles saved and network latency saved are only roughly related, it is expected that with the continued build-out of Internet infrastructure (in terms of exchange points and last-mile links), the gap will continue to shrink. By logging each client's location, c, and the location of the server, s, to which they were redirected, and with the knowledge of OGI's location at o, the number of miles saved can be calculated by computing the great circle distance between c and o, and subtracting the distance between c and s. During the study, the redirector found good servers for 7069 clients, saving over 15.5 million miles, or an average of 2203 miles per redirected client. To better capture the relationship between geographic miles saved and network latency, we used a tool called King, which computes latency between two arbitrary IP addresses on the Internet by using recursive DNS lookups [11].

Using the latency data collected by King, the latency between each redirected player and our server and the latency between the player and the server they were redirected to can be computed and compared. Table 1 shows the latency and distance reductions of redirected players for the server, as well as the available sample sizes. The network latency

Continent	Distance Sample Size	Latency Sample Size	Latency (ms)	Distance (mi)
Europe	1914	787	46.97	4456.18
N. America (West)	1689	1215	-10.01	-1.42
N. America (East)	2614	1901	13.38	1017.07
Asia	574	223	75.23	4981.10
Australia	77	69	-15.77	5889.95
S. America	180	107	153.14	5421.81
ALL	7069	4318	19.39	2203.22

Table 1: Average latency and distance reduction for redirected players

data is a subset of the geographic data, as we were only able to accurately determine latency between 61% of the redirected clients and servers using King. The latency savings per player are modest to small for players in nearby regions and relatively large for players in faraway continents. While the bulk of clients are affected favorably by redirection, a few are very adversely affected. We attribute these outliers to clients who are geographically distant but extremely well connected to other continents, errors in our measurements, and to errors in the construction of the regions. For an example of the latter, look at the clients in Greenland redirected to servers in the midwestern United States. They would probably be better served by a server in Greenland, or at least in Europe. Errors in our measurements can occur in the geographic lookup, or in the King measurements. The majority of clients come from Europe and North America. As the table shows, the clients in regions furthest away from the server (Europe, Asia) benefit most from redirection, whereas clients in the same region as the server receive little (if any) latency savings. Indeed, while European clients save on average 47 ms of latency and 4456 miles, clients in North America West frequently get redirected to servers (slightly) further away than our own. The correlation between miles and milliseconds saved is definitely inexact. Typically, as thousands of miles are saved, so are tens of milliseconds. The notable exception to this correlation is Australia, wherein clients saved 5900 miles of geographic distance but ended up with higher latency on average. This may be attributed to the small sample size of Australian clients (69) and the good network connectivity between parts of Australia and the western coast of the United States.

4. CONCLUSION

We have presented a redirection service for game servers that benefits clients, servers, and the Internet as a whole. Clients can play on servers that are just as good as, if not better than, the server they initially contact. Servers can provide better service to their clients than simply refusing connections, at minimal overhead to themselves. In addition, the Internet as a whole saves in thousands of bandwidth-miles per redirect, at no cost. The service is scalable, by manipulating the size of the regions, and easily deployable.

As part of future work, we plan on studying the impact of region granularity on the performance of the system. In our first pass, the regions defined by the redirect master were determined arbitrarily. It is our hypothesis that by increasing the granularity of these regions, our system would save correspondingly more bandwidth-miles and client latency. To address this, we plan on implementing a dynamic scheme to calculate the regions based on server or player density, or potentially on the aggregate redirection rate of the global redirection service. We would also like to perform data collection on a set of servers in future work. We rely on the King tool heavily in verifying the benefits of redirection, but our results using King with regards to likelihood of a valid lookup, as well as consistency of latency are somewhat contradictory with the results shown in their paper. We would like to perform a more rigorous analysis of the usefulness of King with respect to the gaming population and possibly use King measurements as a basis for redirection.

5. REFERENCES

[1] IDSA: Interactive Digital Software Association, "IDSA Digital Press Room," `http://www.idsa.com/pressroom.html`, 2003.

[2] G. Armitage, "Sensitivity of Quake3 Players To Network Latency," in *Internet Measurement Workshop (Poster Session)*, November 2001.

[3] T. Henderson and S. Bhatti, "Modelling User Behavior in Networked Games," in *ACM Multimedia*, 2001, pp. 212–220.

[4] T. Henderson, "Latency and User Behaviour on a Multiplayer Game Server," in *Networked Group Communication*, 2001, pp. 1–13.

[5] T. Henderson, "Observations on Game Server Discovery Mechanisms," in *NetGames*, April 2002.

[6] Geobytes, Inc., "Geobytes Home Page," `http://www.geobytes.com/`, 2003.

[7] W. Feng and W. Feng, "On the Geographic Distribution of On-line Game Servers and Players," in *NetGames 2003*, May 2003.

[8] QStat Developers, "QStat - Real-time Game Server Status," `http://www.qstat.org/`.

[9] AMX Mod Developers, "AMX Mod Server Plugin," `http://amxmod.net/`.

[10] W. Feng, F. Chang, W. Feng, and J. Walpole, "Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server," in *Proc. of the Internet Measurement Workshop*, November 2002.

[11] K. Gummadi, S. Saroiu, and S. Gribble, "King: Estimating latency between arbitrary internet end hosts," in *Internet Measurement Workshop*, November 2002.