

## **CS290F HW Tech topic #01**

### **Global Hosting Systems**

This patent was filed with the US Patent office in May of 1999, it is titled "Global Hosting System" and carries the patent number 6,108,703. The inventors are F. Thomson Leighton and Daniel M. Lewin, both from Universities in Massachusetts.

The patent describes a system for distributed hosting that enables content providers to only serve the base HTML content and letting servers placed around the world serve data objects contained in the site. By using a hierarchical DNS system they enable each request to be directed to the server closest to the client making the request, they also include a hash based system for load balancing and hinder stale content.

The main improvements this system brings over the obvious solution is being less expensive, better utilizing network resources, unlimited cost effective scalability, allowing the content provider to measure the number of hits to his website and supporting dynamic content.

The system described is very clever and seems to fix all the major complaints with the current systems. CDNs have become an integral part of today's internet, and are only becoming more important as more and more multimedia content is delivered via the internet.

This is the first patent I have read so thoroughly, without knowing a lot about how patents are supposed to be structured, I would say that this patent includes a lot of details and takes into consideration a lot of edge cases. It was easy to read and seemed to follow a well structured formula. I was impressed by the fact that they included mobile devices and other internet connected devices in their definition of a client.

The only weakness I can identify in the patent is that they do not spend much time discussing the benefits of a CDN versus local caching.

### **An Analysis of Internet Content Delivery Systems**

This paper was presented in the 5th Symposium on OS Design and Implementation in 2002, hosted by the USENIX Association. USENIX is an old and well regarded association.

First the authors present their motivation and what they hope to achieve in the paper, their goal is to measure all TCP traffic going in and out of the University of Washington over a 9 day period. They then hope to identify different types of CDN traffic and point out characteristics of this traffic. The traffic types they analyze are Akamai (CDN), Kazaa, Gnutella and regular WWW traffic, as well as other TCP traffic.

They present a very thorough set of data that includes a lot of characteristics of P2P data vs WWW data, as well as Akamai vs WWW. The main points in the paper is that P2P data is very data intensive, the bandwidth of a Kazaa peer is 90 times that of a WWW client, they also point out that Kazaa TCP sessions last a lot longer than WWW connections. Lastly they look at the effects of caching on Akamai and Kazaa traffic, in both cases they show that caching could have a positive effect. Especially with Kazaa given the big file sizes.

The results they come up with seem rather obvious, at least today, this might have been considered a big contribution at its time, but today all of this seems obvious. Another weakness in the paper is the fact that they repeat themselves, the summary chapter is

very similar to the conclusion chapter. Also the paper is filled with a lot of numbers and graphs without doing that much analyzes. The biggest strength of the paper is that looks at a lot of aspects of the traffic. Their results might be beneficial to big organizations, especially the results from their caching tests.

### **Anycast-Aware Transport for Content Delivery Networks**

This paper was presented during the 2009 WWW Conference in Madrid, the WWW conference is a long running conference with low acceptance rate.

The paper starts out with the authors presenting the motivation for their solution, they say that using anycast for CDN networks have two major drawbacks. The first is the fact that load balancing is not possible, the second is that a route can be changed mid transfer and cause termination of the TCP session and requiring the user to restart the transfer.

The authors then present their solution, which they say is very simple, their solution is to modify all the clients to resume transfer in the case of a changed route by sending a new request for the remaining data. They also suggest making some changes to the server implementation to better facilitate the new connection model. The solution is simple and well explained, the authors claim that requiring the users to download a program or extension to enable this type of transfer is not a big deal since some CDNs already require it and some applications can include it in an update (iTunes is the example used).

The authors then present a performance evaluation of the system, they show that the performance loss is not significant, and the benefits of anycast (mainly using the network to route to the closest server instead of basing it on the DNS IP address) make it a viable solution. One interesting thing to note is that the distance from a user to his/her DNS was considered a non-issue in the original CDN patent from 1999, but apparently this has become a bigger issue since then.

The authors analyze edge cases and look at security implications as well as different performance factors, all in all this looks like a thoroughly thought out solution to the problems with anycast.