
Global Hosting System

This patent was filed May 1999 with regards to proposing a new method for efficiently distributing content to end users thus essentially allowing web content providers to scale well to larger amount of traffic. Prior to this patent, scaling was handled in two fashions, the first being the addition of servers, referred to as mirrors, which could be in the same geographic location or in multiple geographic locations. These mirrors contain identical copies of the origin server thus making them difficult to manage, as the patent argues. This sort of scaling allows the web server to handle larger volumes of traffic however, does not generally improve the end user experience as the web server may still be far away. The second method mentioned was scaling through web caching proxies in which ISPs would attempt to reduce their traffic by saving local copies of content. The patent argues that "...such caching can produce devastating results to the Content Provider...". However this method does indeed benefit the end users' experience as they would have lower latency access to cached information. Their argument is on the basis that the content provider cannot obtain accurate hit counts, nor are they in control of whether content is up to date however the authors seemed to neglect that even HTTP/1.0 had a "no-cache" header thus granting the content provider full control over this information.

The goal of the patent is different than that of a published paper. As I see it, a published paper is a way for someone to share information with the world, whereas a patent is a way for someone to essentially say, "I have this system and you can't have it; unless you pay me". Thus the goal of this patent is to claim stake to the described methodology for more efficient content distribution. As such I have a hard time evaluating the contribution of the patent. The idea is ingenious but if I cannot freely implement it and utilize it myself then it's all for nothing. However in the interest of my grade, let's proceed on the assumption that the authors wanted to share this information with the world thus removing my biased opinion on software patents.

Before diving into the specifics this patent was one of the hardest pieces of writing I've ever read. It was incredibly redundant, unorganized and had numerous spelling and grammatical mistakes many of which appeared in the same sections. I took notice to the corrections page however this didn't come close to acknowledging a third of the mistakes. As a reader I find these mistakes to be quite offending. Regarding redundancy, certain parts of the text were repeated more than three times leading me to believe this document could be half as long with redundancy removed. It's no surprise that patent lawyers have a difficult time determining what a patent actually says when information is repeated five times in different forms.

The basics of this patent is to explain a system that places content providers' content as near to the end users as possible. This system is accomplished by CDNs through the placement of servers within numerous unique ISPs. These servers are sometimes referred to as a point of presence (POP). The content provider's static files, or objects, are given a serial number by which to refer to them and this serial number is included as part of a virtual hostname used by the CDN to load balance access to the object. When an end user performs a DNS lookup on the hostname, the CDN's first tier DNS server responds by pointing to second tier DNS server(s) in the requester's geographic location. The second tier DNS server then responds with server addresses ordered by those with the least load. The resulting server will serve the object out of its cache if available and up to date, otherwise the server will proxy the request for the end user meanwhile saving the object in its cache for later use.

The patent goes into incredible detail with regard to this process and addresses a few other subtleties of which the following I took note as positive considerations. First, they specified DNS TTL intervals to optimize DNS caching thus reducing the number of overall DNS queries. Second, the mention of consistent hashing is a great foresight to minimize the amount of work needed when adding or removing servers from the CDN. Third, another foresight is the workaround suggestion with respect to large network DNS servers which aren't necessarily in the end user's region. Finally, they mention midstream redirection as another means for load balancing however the patent doesn't propose a means to do midstream redirection.

One thing this patent neglects to consider is the advent of HTTP/1.1 which originally came about in 1997. The relevant aspects of HTTP/1.1 are persistent connections and additional cache-control headers. Persistent connections allow a single TCP connection to make multiple HTTP requests thus eliminating the 3-way handshake for additional HTTP requests. Their system of using different hostnames for each object does not seem to allow for persistent connections, and additionally has the added overhead that a DNS lookup must be made for each object. Though the benefits of having the content closer to the end users very likely outweighs the consequences, the consequences still should have been addressed.

Finally I find it interesting that the patent explains a great deal of the client/server background information but assumes that the reader knows how DNS works. Though decreasing the length of this document would be my priority, the document should have included a brief explanation of how DNS operates.

An Analysis of Internet Content Delivery Systems

This paper seems to have been originally published in OSDI-02 and additionally published in ACM SIGOPS Operating Systems Review, Winter 2002. OSDI alone is a great systems conference and the addition of the journal submission seems to give this paper further credibility.

The authors of the paper sought to measure the impact of varying content delivery systems as compared to those systems from the three years prior. The content of the paper is a measurement study conducted at UW over a period of 9 days looking at web traffic, Akamai CDN traffic, and p2p traffic specifically within the Gnutella and Kazaa networks. The paper presents a detailed report of said traffic from the viewpoint of clients, servers and objects. The key points the authors desire to get across are (1) that the quantity of p2p traffic has vastly surpassed that of traditional web traffic, (2) that the median size of objects (files) transferred are larger, (3) that p2p requires significantly more outgoing bandwidth and traditional web traffic, and (4) that p2p systems do not scale.

The paper starts by overviewing the three systems, web, CDN and p2p and follows with their methodology. Their methodology for the most part is pretty strong however I personally have a few concerns. First, given that Gnutella made up such a small portion of the traffic, I question why it was targeted for analysis. Second, the authors acknowledge that unclassified TCP traffic made up over 43% of their network trace. This percentage of unclassified traffic seems incredibly significant. I would have liked to see a classification for the remainder of the traffic or at very least an acknowledgement that this unclassified percent was consistent with previous data.

The remainder of the paper details the data they collected from many different angles however they neglect to consider one angle which I think hurts their analysis. In section 5.3 the paper reads, “The Kazaa curve shows 600 external Kazaa peers … supply 26% of the Kazaa bytes to internal peers; this result, however, is somewhat unexpected.” While the authors expected the Kazaa server load to be more distributed, they don't consider internal peering within the UW network. If the authors were to consider the internal portion of the Kazaa network they may have found that the majority of content was retrieved from within the network, thus only reaching outside the network for new or rare objects. It seems reasonable that internal peering has the fortunate consequence of serving as a forward cache for the network thus giving reason to why they found no benefit to having an internal cache and offering an explanation to why there was not an even distribution of load to external Kazaa servers.

The paper ends by discussing caching in such situations. They suggest a a local web proxy cache would achieve about the same hit rate (after warm up) as Akamai's CDN which is great, however that only would save a small amount of incoming bandwidth compared to the p2p traffic. They additionally looked at p2p caching and found that having a reverse proxy cache within their ISPs would grant the University with considerable savings however given the legality of content in p2p, networks it may not be feasible.

Furthermore I question if this analysis is representative of many ISPs. UW is a college campus which stereotypically has poor students in their late teens / early twenties who according to my speculation are more likely to participate in p2p activities than the normal Internet user. This paper mentions its target audience is large organizations, and service providers but I think the target audience really lies within campus networks.

Despite my criticism for the paper this sort of analysis needs to be done from time to time at a large scale. I don't think every three years a paper of this sort should be published in a major conference, however perhaps this was okay as it was the first paper to compare this sort of traffic to a previous study (was it?).

On presentation I feel this paper did a decent job with the material. The included tables and figures were not as readable as they could be, however they consistently depicted what their commentary described. The paper was organized logically and was fairly easy to read given that it was mostly explaining numbers from their dataset.

Anycast-Aware Transport of Content Delivery Networks

This paper was published in WWW 2009 in the Performance, Scalability and Availability track. From what I know, WWW is a well known conference with a fairly low acceptance rate. Based on the recent work mentioned in this paper, they sought to address issues with the load balancing of content distribution servers when serving large files. They mentioned prior work regarding malicious users DoSing content distribution servers by initiating connections for larger files, which under conventional means, may take upwards of ten minutes to timeout. Additionally the paper considers the DNS flux of current CDN servers inferior to their

proposed use of anycast.

Anycast is the concept that multiple hosts can share the same IP address and end users will reach the closest host to them because network routers continuously optimize their routes. Anycast has the disadvantage that the end users are not guaranteed to reach the same host throughout their communication and as such this paper addresses concerns with content distribution and connection interruption.

The paper's proposed solution is simple, just do anycast, however they realize by doing so a few issues need to be addressed. The first issue is with clients recognizing connection disruptions and re-requesting content. They propose lessening the retry count on TCP connections and then resuming the download via another request with an offset header. The proposed method was shown to have little overhead even in a lossy environment.

The second issue was with dormant connections on the server side and when to free those connections. Again the solution was to lessen the retry count thus making timeouts occur in ~600ms. Their experimentation showed that this timeout was high enough to not cause premature timeouts, but low enough to avoid potential DoS attacks.

The third issue was regarding any ill effects of having multiple streams simultaneously switch to a new server. The primary concern was with oscillation due to TCP slow start, thus they setup a test framework to mimic the situation and were unable to reproduce TCP oscillation therefore claimed the issue was not a problem. While I have never personally seen TCP oscillation, I am aware of it only because someone has observed it before thus I find it difficult to believe that they couldn't reproduce TCP oscillation. I question if their use of Linux TC for creating a bottleneck somehow internally resolved the possible oscillation that a real router may have, or if the implicit synchronization of using only a single client and server machine, even with 1000 concurrent connections, is satisfactory for reproducing oscillation.

The fourth issue dealt with how splitting a download into multiple chunks would effect the overall download throughput. Through experimentation they show that at a loss rate of 10% the connections are only reduced by 6% in throughput thus assuring that throughput is not effected. While I agree these numbers are good, their argument that 10% is a high loss rate is questionable, especially given that many users are utilizing wireless networks which may have loss rates exceeding 10% (I'm guessing).

The final issue dealt with the security implications of such a system. Because of the lessened TCP timeout the authors assured their proposed system would be more resilient to attack, which makes sense.

While I initially felt the idea behind this paper was incredibly trivial, the authors consideration of potential drawbacks to the system reinforced my opinion that this system would be quite simple to implement and would have little drawback. As previously mentioned a little testing should be done with respect to wireless networks, but other than that I think it overall is a solid contribution.

Finally, this paper, unlike the patent, was incredibly well written. Not only was the organization superb, they even took the effort to address which sections of the paper solutions to problems were mentioned in. For instance on page 305 they reference experiments that they discuss later in the paper. On top of readability the figures are easy to understand and the figure commentary in the text is also well done thus making this paper a pleasure to read.