Akamai Paper Review

By Taeju Park and Andrew Quinn

Paper Reference

Nygren, E., Sitaraman, R.K., and Sun, J., "The Akamai Network: A Platform for High-Performance Internet Applications," *Proc. of ACM SIGOPS '10*, 44(3):2-19, Jul. 2010.

Summary

This paper starts from discovering the major challenges of the Internet to deliver large amount of content from origin servers to end-users. After that, they introduce the overall architecture and key techniques of Akamai's edge platform to overcome the challenges based on their design principle. The most powerful architectural approach they leveraged is a highly distributed edge platform. Since Akamai's edge servers are deployed at thousands of locations around the world, each end-user can be assigned to an edge server located in close proximity. The key techniques which are used by the Akamai platform are application-specific implementations, tiered distributions and the overlay network. Through these techniques, Akamai's edge platform achieves their design goal. They shows the effectiveness of their platform by introducing several business cases.

Review

Strengths

This paper ultimately had some awesome nuggets of research which were unfortunately hidden amongst marketing fluff. The authors seemed most interested in selling their work, focusing on things like providing reliability, and suggesting that “To guide our design choices, we begin with the assumption that a significant number of failures is expected to be occurring at all times in the network”. They seem to think this is a novel contribution of their work, or at least surprising in a distributed system; both of these are incorrect. Instead, the main contributions of their paper are the realization that the AS routing of the internet is extremely unreliable (this is basically what is described in Section 3), and using a large set of small data centers scattered throughout the internet in order to overcome the AS level routing challenges. Interestingly, they don’t really motivate the need to use many small data centers, even while this decision is what guides all of the major contributions that Akamai makes in this paper.

Akamai breaks client connections into two pieces. In the first piece, clients connect to edge servers. This will be an extremely fast connection because the client and edge server are co-located in the same AS. The edge server then coordinates traffic to the origin server, which may be across the wide area internet. Akamai speeds up this traffic by leveraging the highly distributed nature of their data centers. Specifically, they take advantage of two things:

1. They can use the newest and best advances in networking technologies to make connects fast over the wide internet. Essentially, they’re fixing the problems of client adoption in these advancements by taking over the adoption themselves

2. They have knowledge of the state of the internet at large which lets them make better AS level routing decisions than BGP.

In section 3, one of the major limitations of the internet that the authors note is that many protocol advancements require client adoption. Unfortunately, client adoption is often quite slow; for example, it was cited that internet explorer 6 was eight years old, and also one of the most used web browsers at the time of publication. Akamai reduces the effects of poor client adoption by leveraging new technologies in communications between the edge host and the origin server. Since this portion of the transaction accounts for most of the data transmission, Akamai reduces the effects of poor client adoption and is still able to take advantage of the networking community's research.

Akamai also uses their high distributed data centers to inform them of the state of the internet at large. They leverage this information for a few purposes. First, they essentially do their own AS level routing. This allows them to perform better than BGP, as they can always prefer the fastest route (thereby removing AS level business logic in routing decisions). They can use the same logic for adding redundancy when sending packets, as they can send the same information via different AS level paths. Both of these optimizations are only possible because Akamai has a data center in nearly every AS, allowing them both the ability to choose different routing paths, and the information to know which paths are the fastest.

Implications and Extensions

One interesting implication of their work is that they are making inter AS routing decisions. Essentially, they don't follow the decisions that BGP makes. In some cases this is unlikely to be a problem; they often have RTT information and statistics that ASs don't have when making routing choices. However, there are times when an AS decides NOT to choose a

route because of business logic (i.e. a tier 2 AS may not want to route traffic to a tier 1 AS because of cost). Since Akamai chooses routes without regard to these business decisions, they are disregarding the autonomy of the AS. While this is beneficial for the user of internet services, it seems surprising that there are many ASs which are willing to go along with this; I'm surprised Akamai isn't constantly getting sued.

Another interesting implication of their work is from cached content in edge servers. To improve the response times of requests made by users, they use the concept of cache. Highly-accessed content is stored in an edge-server and the cached data is transmitted to the end-user. It seems like memory hierarchical system. Thus, in order to keep consistency between cached content and the content in origin servers, an edge server has to check content consistency by using either event-driven or time-driven pings. In this paper, they didn't delve into the details of content consistency. Thus, we wonder how Akamai is able to deal with these content consistency problems, as caching is unlikely to benefit the ever increasing dynamic websites of today.

Many of the performance improvements from the Akamai platform comes from the large number of users and cached content. However, if there is only few customer, all the requested content will be served by the origin server because no users have previously requested the content. In this case, the performance of the Akamai platform might significantly drop. To solve this problem, we think that prefetch techniques, which are extensively studied in computer architecture community, can be applied to the Akamai system.

Discussion

Nearly all of the ways that Akamai has improved performance come from this geo-distributed set of data centers, which is interesting when comparing their approach to the

approach taken by other CDNs. For example, google, which has to server youtube, only has a few dozen data-centers in the world (https://www.google.com/about/datacenters/). It is highly likely that google is able to see roughly the same application performance as Akamai, especially since youtube is cited in this paper as one of the reasons for needing CDNs. But, the authors of the paper have done nothing to argue why using many data centers is superior to using a few data centers. The system that Akamai has build relies upon having many data centers, but the authors do not compare their approach against an approach with less data centers. Instead, the authors only provide the intuition that without machines in nearly all ASs, traffic has to cross through many networks. Based on the fact that google only has a few dozen data centers, I find it very disheartening that they don't validate this design decision more.

Additionally, the authors provide essentially no evaluation. They instead list off a few clients of theirs which have seen some improvements. I find this very unfortunate. Not only do they refrain from a low level analysis of which part of their system accounts for the speedups, they also do nothing to show how many data centers they need in order to use their approach. For example, it may be possible to approximate some of their behavior with only using data centers sitting on tier 1 ASs. Without a detailed evaluation, I have no idea which of their ideas works well, and which ones do not.