HW-4

Mayunk Kulkarni PID:A53285335

Experiment results and Conclusions

The results obtained show that it is better to assign the location of the hashes randomly than repeatedly ping the blockstores for each hash depending on the file size. As the file size increases, it is observed that the performance deteriorates as we have 4-5 data stores with which we need to ping with. Along with this, it is found that it is faster to complete the download operation with the original server selector function as no RPC calls are made for server selection. The RPC calls accumulate to form a large delay. These RPC calls are basically the calls made by the download operation to the metastore process asking for the block location. Though they are on the same system, the whole system of transporting a request and getting an answer takes up time. During upload operation there is a massive delay for uploading. It is considered faster to upload to random servers for large files if calculating for every single hash in the file. Ultimately it is found that if the file sizes are small, the time taken to download reduces significantly. The time taken for the original method is around 3.67 seconds whereas the new method is 1.536 seconds. This is more than a two-fold improvement in download time.

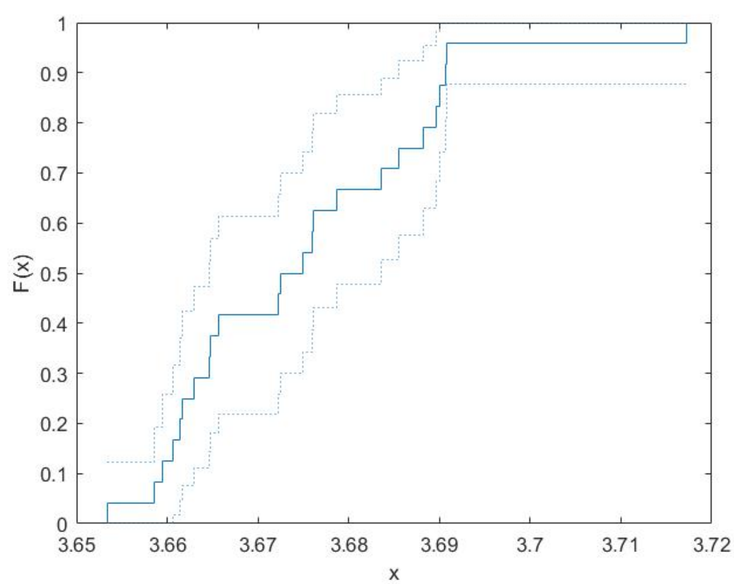

Figure 1: CDF for 4k files for method 1

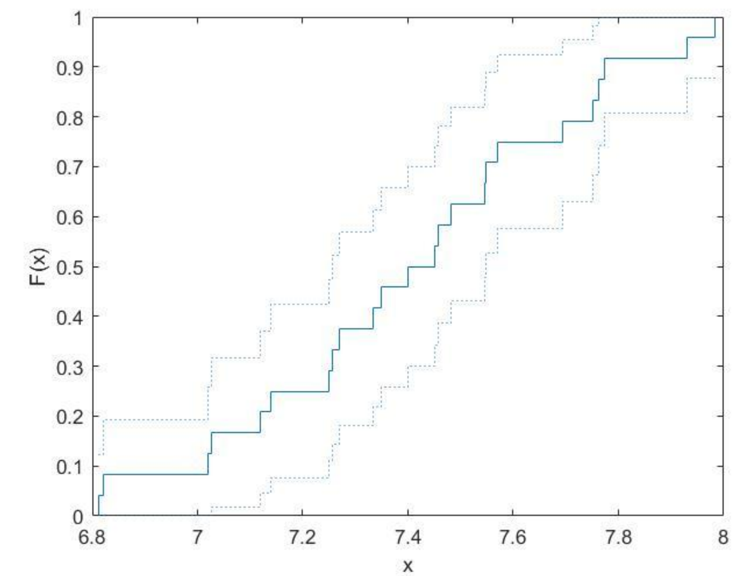

Figure 2: CDF for 60k files for method 1

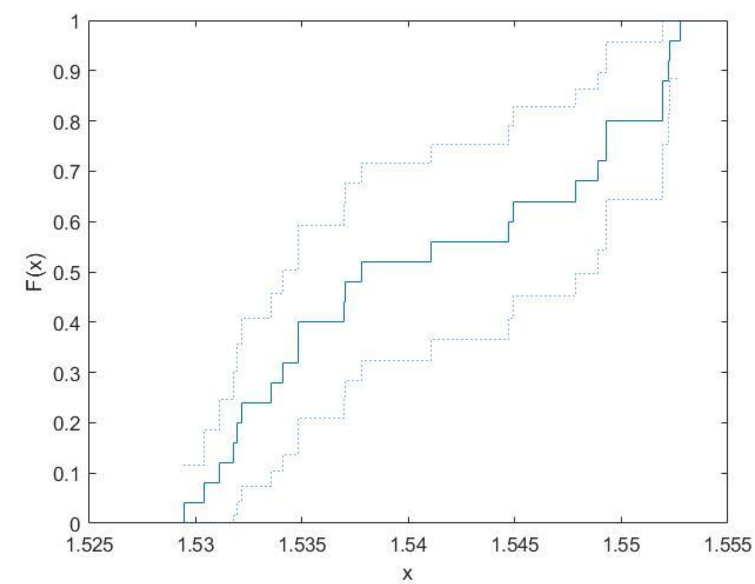

Figure 3: CDF for 4k files for method 2

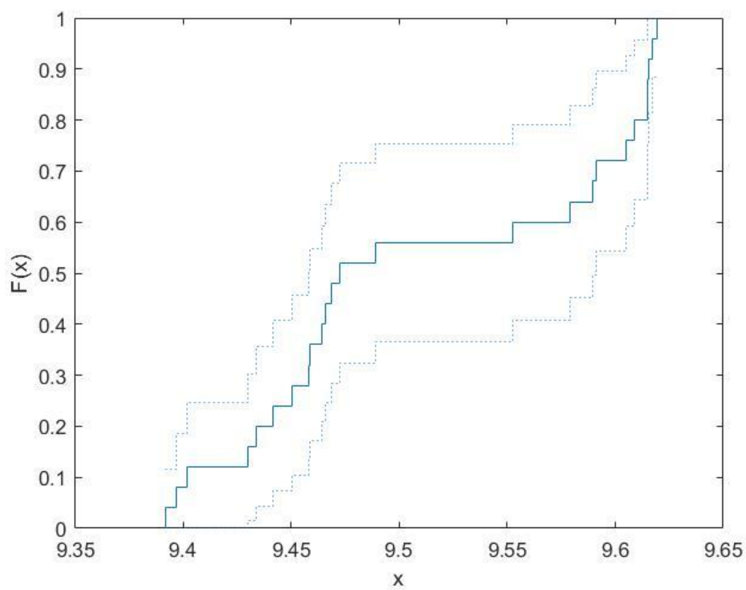

Figure 4: CDF for 60k files for method 2

Paper Summary

- What are some of the challenges to delivering content over the Internet?

  The paper talks about Akamai's engineers going through several challenges they faced in providing a strong content delivery network for its clients on the environment provided by Internet. This involved various challenges. For instance, the peering points in the network act as bottlenecks due to less investment as they do not provide any revenue thus reducing incentive in improving them. Inefficient protocols run by the internet such as BGP may slow down the connection as many routers do not have updated route tables. Another reason for delivering content is the chance that networks between two points may get outages which would lead to further problems in delivery of content. Added with this, protocols such as TCP are not built to handle the large amount of data that is being transferred over the internet. It simply wasn't built to handle the amount of data being currently transferred.

- Why isn't a simple web server model sufficient to efficiently deliver content? (In other words, what need or gap does Akamai address?)

  A simple web server model is not sufficient to efficiently deliver content as the current situation involves delivering content to a multitude of users located in various geographical locations. Due to this, having a centralized server delivering content to various other end points reduces the performance of the system due to handling multiple requests at the same time. Akamai provides a content delivery network (CDN) to address these challenges. These CDNs improved website performance by caching the site in the edge of the network hence reducing the load faced by the server. This also reduces the latency as the content is provided at the edge of the network closer to the clients and having multiple such nodes providing content reduces the bottleneck of a single server providing content to multiple end points.

- Briefly describe how Akamai enables distribution of live and streaming content?

  Akamai handles live and streaming content in two major ways. One of the ways of ensuring increased bandwidth and reduced latency in streaming is introducing a more distributed, tiered (multi-level) network that provides high demand content closer to the edges of the network and the infrequent demand content on the higher tiers based on its demand. This kind of distribution reduces the load on the origin server. For companies providing high quality content to its users, this reduces the bottleneck for content and reduces the load on the main server. Along with this, for live streaming content the CDNs use a different method that ensures that content reaches the users without fail. To ensure this multiple end points receive the content so that in case an end point crashes, there is always another end point to provide the content. As live stream service usually comes from a single source, this source sends the content to a reflector point which replicates the data and sends it to the edge servers. This feature allows rapidly replicating streams and providing it to many users. To reduce replicated content flooding the network, the CDN follows a publish-subscribe mode where the content is published by the main server and the edge servers subscribe to a particular service that reduces the number of streams that are being replicated.

- What information does the Akamai CDN use to map requests to resources? How (briefly) does it implement that mapping?

When the user types a URL into his/her browser, the domain name of the URLis translated by the mapping system into the IP address of an edge server to serve the content. The mapping system uses historical data and current data about the route tables, server conditions to provide an edge server that can best provide the services the user needs. The content is cached in these edge servers but also sometimes for dynamic content the edge server requests that data from the main server using efficient transport protocols.