The Measured Performance of Content Distribution Networks

Kirk L. Johnson, John F. Carr, Mark S. Day, M. Frans Kaashoek

SightPath, Inc.

135 Beaver Street, Waltham, MA 02452, USA

Abstract

We have measured the performance of two commercial content distribution networks (CDNs), one operated by Akamai and one operated by Digital Island. Although there are differences in the implementation of these services, both CDNs redirect requests by using DNS. In this paper, we describe our simple measurement technique for a DNS-based CDN, our data for the two commercial services, and our interpretation of that data. Our main conclusion is that CDNs provide a valuable service, but that neither Akamai nor Digital Island can consistently pick the best server of those available. Contrary to some widely-disseminated marketing messages, we argue that CDNs succeed not so much by choosing an "optimal" server as by avoiding notably bad servers.

1. Introduction

A *content distribution network* (CDN) is an architecture of Web-based network elements, arranged for efficient delivery of Web content. For our purposes, "Web content" includes any of the various forms of data items that can be distributed via the Web. To date, CDNs have been used primarily for the distribution of heavily-requested graphic files (such as GIF files on the home pages of popular servers). In the absence of a content distribution network or proxy caches, all requests for web content go to the *origin server* named in the host part of the URL identifying the content. Although such an origin server may in fact be implemented as a large "farm" of server machines, those machines are usually in a single small geographic area. All of those machines may be far from a given client (in terms of network latency).

In contrast, a content distribution network has multiple replicas of each content item being hosted. A request from a browser for a single content item is *routed* to a "good" replica, where "good" usually means that the item is served to the client quickly compared to the time it would take fetch it from the origin server. Static information about geographic locations and network connectivity is typically not sufficient to do a good job of choosing a replica. Instead, a CDN must incorporate dynamic information about network conditions and load on the replicas, routing requests so as to balance the load. A CDN is effectively a collection of widely-dispersed caches, with two crucial differences. First, the caches are potentially populated by a means other than the requests from clients; and second, the caches are coordinated by a mechanism that routes client requests to a "good" cache.

We have measured the performance of two commercial content distribution networks (CDNs), one operated by Akamai [1,3] and one operated by Digital Island [2]. Neither company has published details of its technology, and indeed both companies treat those details as important trade secrets. There are visible differences in the implementation of these services (for example, the so-called "Akamaizing" of URLs is different from whatever Digital Island does). Despite the secrecy and visible differences, both CDNs appear to redirect requests by using DNS. In this paper, we describe our simple measurement technique for a DNS-based CDN, our data for the two commercial services, and our interpretation of that data.

Our measurements show interesting performance properties of the services. However, our approach has limits: it is focused on latency; it does not test the load-balancing capability of a CDN; and it does not allow a head-to-head comparison of the two CDNs on identical data.

Our main conclusion is that CDNs provide a valuable service, but that neither Akamai nor Digital Island can consistently pick the best server of those available. Contrary to some widely-disseminated marketing messages, we argue that CDNs succeed not so much by choosing an "optimal" server as by mostly avoiding notably bad servers.

2. Methodology

We measure these CDNs from the outside, because we have no privileged access to the configuration or operation of the services. This approach has two advantages: we are thus measuring the end-to-end system performance as it would be experienced by end users; and almost anyone can use our technique.

We gather data in two steps. In the first step, we repeatedly query well-known and geographically-distributed DNS name servers to determine the set of servers that are in use by the CDN. In the second step, we bypass DNS entirely and send requests to the set of servers determined by the first step. By splitting the process in this way, we are able to compare the choice made by the CDN with the alternatives that were available to it, so as to understand whether the CDN is doing well at the task of server selection.

In our experiments, we fetch a few-kilobyte GIF file from each server. On such fetches, latency is typically more interesting than bandwidth.

2.1 Determining the servers

We started with sites that were known to be distributed via the CDN of interest, and mechanically extracted fully-qualified domain names (FQDNs) of the form known to be used by the CDNs of interest. We then repeatedly resolved those FQDNs at a collection of 45 name servers in North America, iterating 100 times with 5 minutes between each iteration, and recording the servers returned.

2.2 Measuring the servers

Armed with a collection of IP addresses for servers, we then measured the performance of each such server on a request, and compared the performance of the full set to the performance of the server chosen by the CDN. In each experiment, a particular GIF file was repeatedly fetched via HTTP from each of the identified servers. The same file was also repeatedly fetched from the server identified by resolving the FQDN that was used in the URL that named the GIF file. For each fetch, the target IP address, size of returned object (in bytes), and fetch latency were recorded. The measured latencies include TCP connection setup time, but do not include time spent performing name resolution for the measurements of the CDN's selected server.

For our test of Akamai's service, the file was 4672 bytes and was fetched from each server 25 times, with requests to servers interleaved (i.e., fetch the first time from each server, then fetch the second time from each server, etc.). Our tests ran on machines in three different geographic locations, which we characterize as A, B, and C. On each test machine, a shell script was used to repeatedly run a single experiment (collecting the output to a timestamped file), sleep for 90 minutes, and repeat. Over 80 experiments were run on each machine in this manner. The tests were carried out on several consecutive days and at different times, with similar results – we present data from one day.

For our test of Digital Island's service, the file was 3776 bytes. Other conditions were similar to those described for Akamai. Our tests ran on machines in the same three locations as our Akamai measurements. However, since the Digital Island tests took place on different dates, fetching different data, and using different machines, we use the names X, Y, and Z for the locations to emphasize that the tests cannot be directly compared.

Our location name	Geographic location	Operating System	Narrowest bandwidth to Internet
A, X	Waltham, Massachusetts, USA	RedHat Linux 5.1	T1 (1.544 Mb/s)
B, Y	Cambridge, Massachusetts, USA	SunOS 5.5.1	10 Mb/sec Ethernet
C, Z	Boulder, Colorado, USA	BSDI 3.1	T1

Table 1. Locations used for measurements

Table 1 shows some relevant information about the physical locations used for our measurements. We include information about operating system of each test machine (since TCP implementations can vary considerably in their performance), and identify the narrowest bandwidth to the Internet from that test machine.

3. Results

For each experiment, we show cumulative distributions plotted versus latency. Since we want to understand the frequency and magnitude of bad choices, a cumulative distribution is a good way to present this kind of measurement -- especially compared to presenting best-case or average latency.

On a cumulative distribution plot, an ideal result would coincide with the y-axis, representing 100% of results with zero latency. Although we can't reasonably expect zero latency, we can hope for consistent performance, which would correspond to a vertical line. Accordingly, the flatter (less consistent) and further to the right (slower) a curve is plotted, the worse it is.

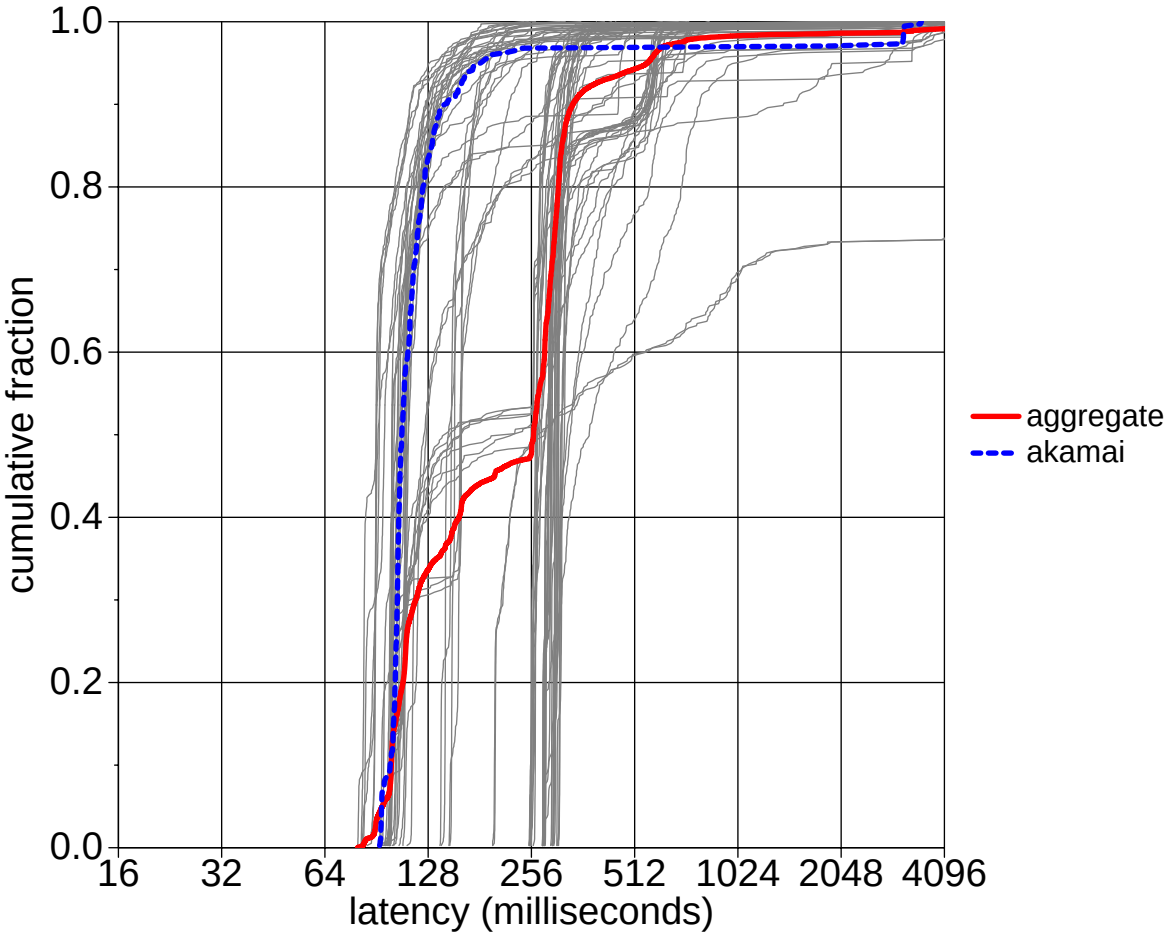

Figure 1. Akamai latency at location A

In each plot, we show each individual server's cumulative distribution as a gray (light) line. Simply by looking at the gray lines, we can see that there is significant variation among the candidates, and that some servers would be very poor choices. In Figure 1, for example, the upper-leftmost gray curve shows that the best server responds in 128ms or less almost 95% of the time. The lower-rightmost gray curve shows that the worst server has responded within 2 seconds about 70% of the time, and thus that the latency is worse than 2 seconds for 30% of the requests to that server.

We note that the vertical "bands" of servers are consistent with geography – since location A is on the East coast of North America, the left-hand group of servers are probably also at East-coast locations. There are a few servers in the Midwest, then a second group of servers on the right for West-coast locations.

The blue (dashed) line represents measurements taken by resolving the CDN's FQDN (thus invoking the CDN server selection mechanism). The red (solid) line represents the cumulative distribution function if all of the individual server measurements are aggregated. This aggregated plot represents a trivial "randomized" server selection mechanism, choosing a server from a uniform distribution of the available servers. Such a random server selection incorporates no knowledge of the network.

Figure 1 shows the latency of the Akamai service, as measured at location A. From this plot, we can observe that Akamai is not always picking the fastest server available from this location. That is, there are servers with cumulative distribution functions to the left of Akamai's cumulative distribution, meaning that they have lower latency.

We can also observe that Akamai is usually adding value compared to simply choosing a server at random from a uniform distribution of the available servers. However, it is not always adding value: because the CDN is not always picking the fastest server, one could occasionally do better by a random choice (thus the area on the lower right where the red (solid) line is to the left of the blue (dashed) line). The figure also shows that sometimes the CDN picks a notably slow server, and a random choice would have been better (thus the area at the top where the red (solid) line is above the blue (dashed) line.

Finally, we can observe that the highest latencies plotted are large – multiple seconds, rather than milliseconds. Accordingly, some clients are experiencing long delays despite the use of a CDN. Other measurements indicated that these long latencies are due to packet losses. Since the losses were usually after the first packet had been acknowledged, we believe that most of these long latencies were caused by network congestion.

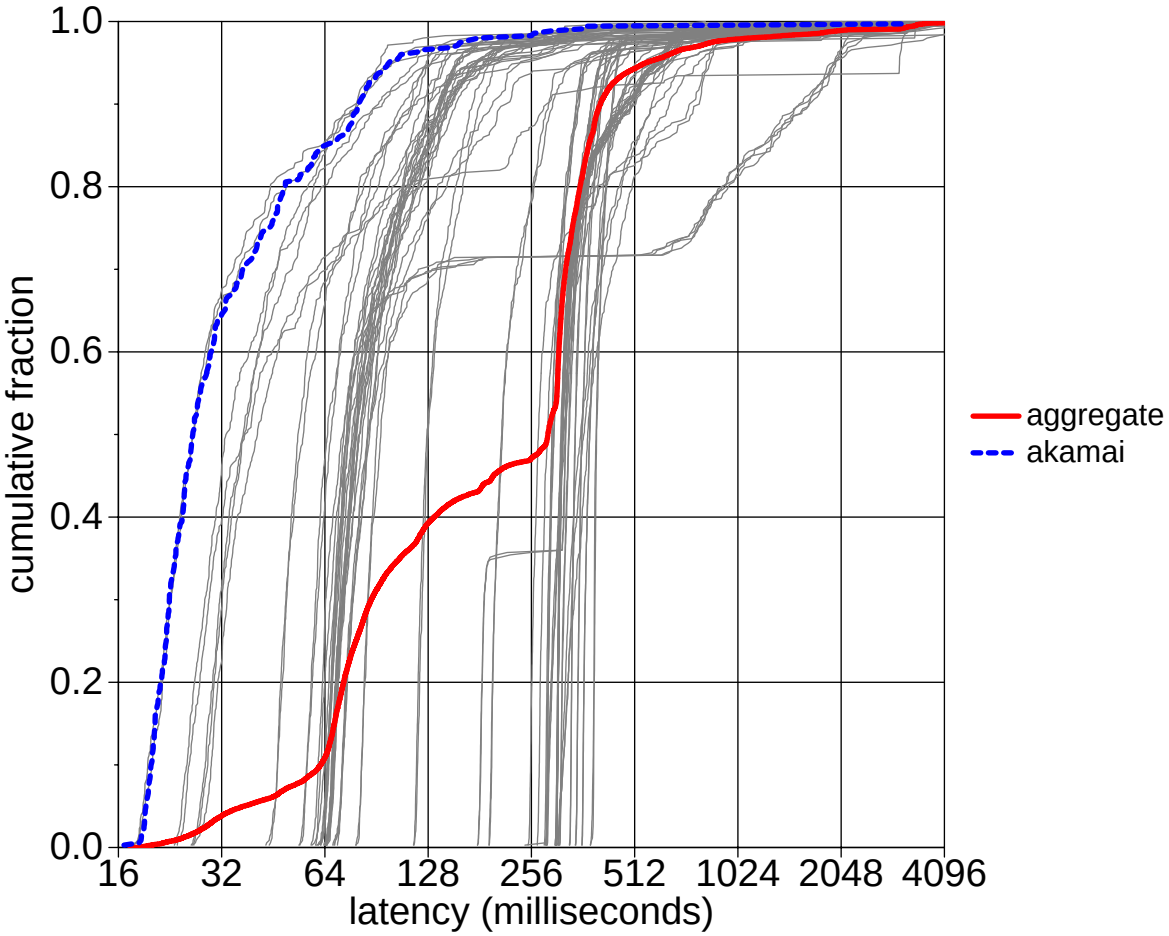

Figure 2. Akamai latency at location B

Figures 2 and 3 are two additional plots from the other network locations we used, which show many of the same features but which also give a sense of the variability due to geography and other factors. In Figure 2, we see the performance of Akamai as measured from location B, and Akamai is doing better compared to random selection: that is, there is a much larger gap between the performance of the CDN and the performance of a random choice.

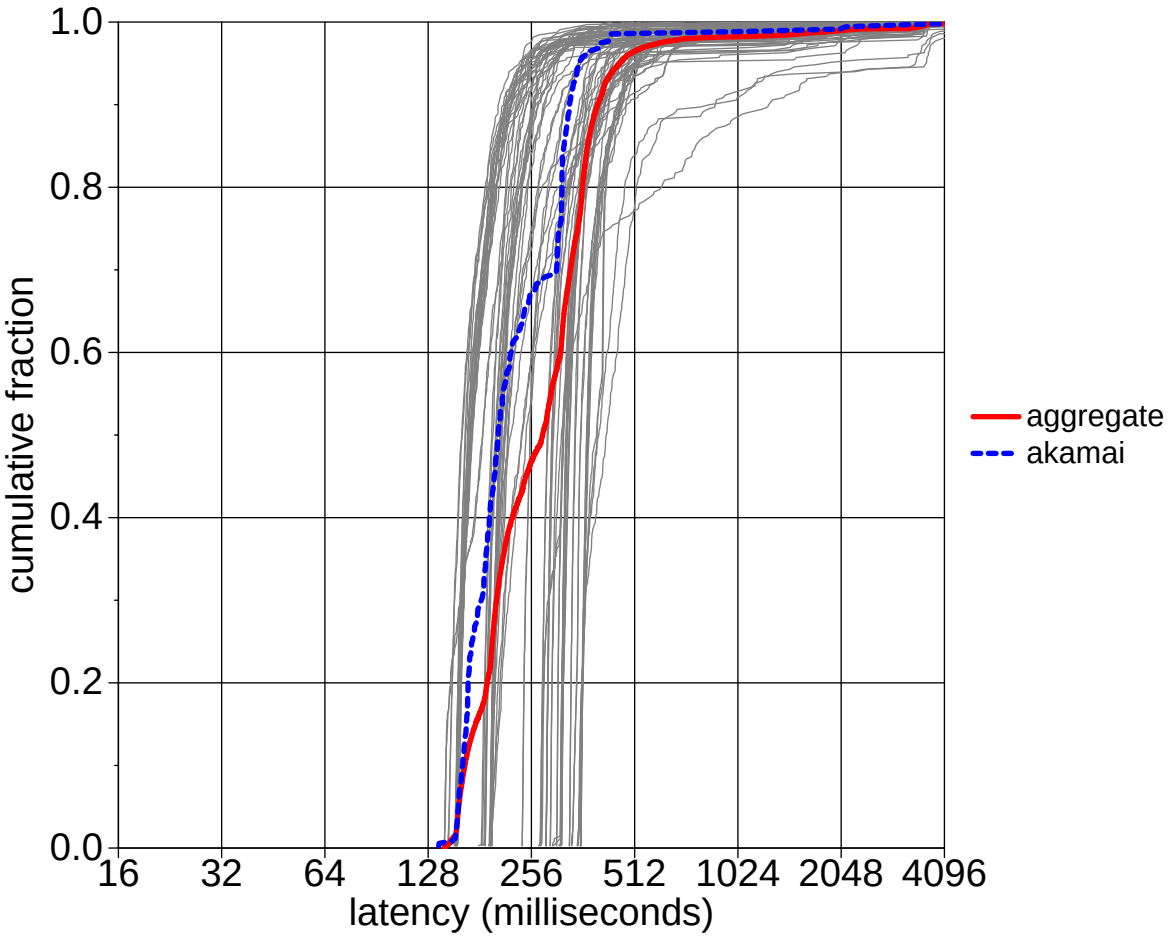

Figure 3. Akamai latency at location C

In contrast, Figure 3 shows Akamai's performance as measured from location C. The gap between the CDN and a random selection is much narrower, but the differences among all servers are also narrower.

For some reason, the CDN is not as effective at delivering content to this location. We speculate that the nearest node of the CDN is still quite far away from the client at location C.

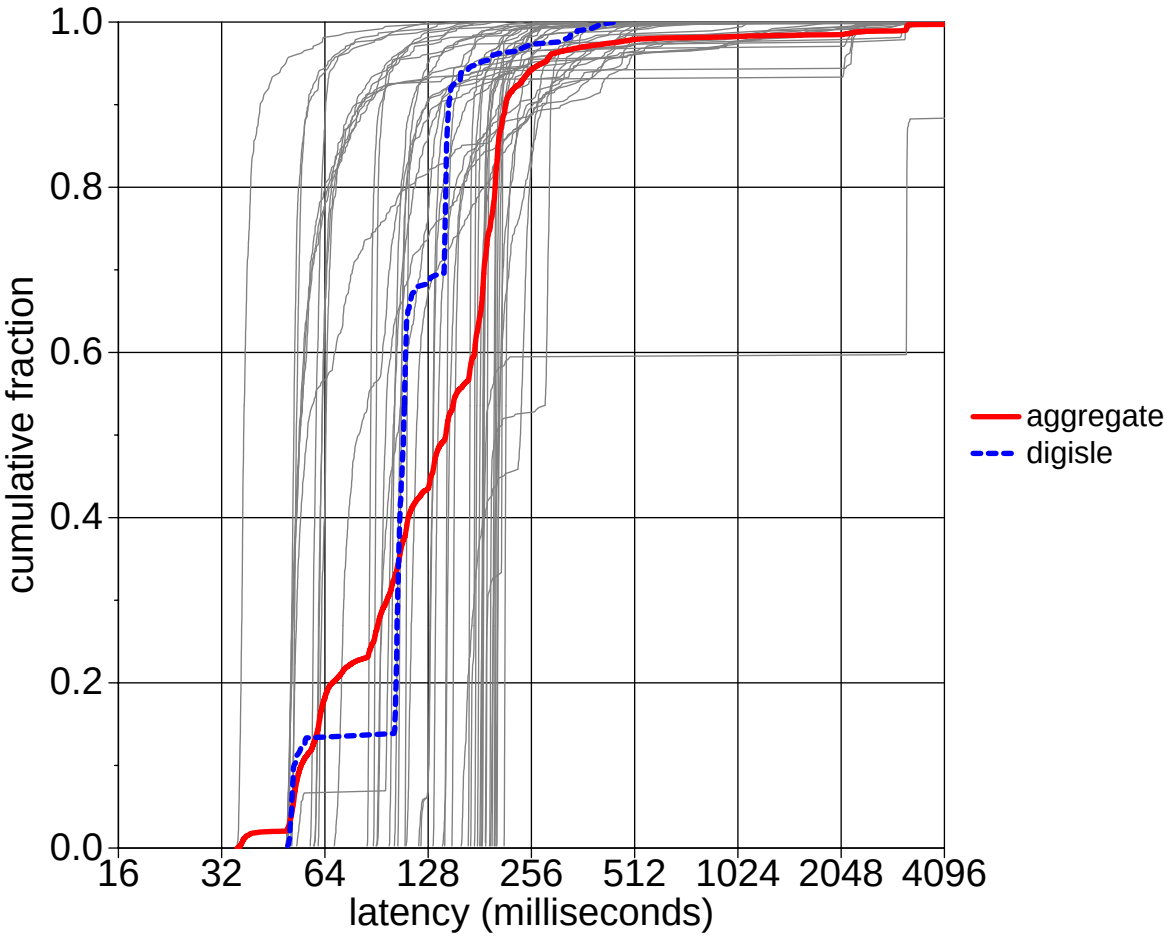

Figure 4. Digital Island latency at location X

Figure 4 shows our first plot of Digital Island data, for location X. This plot is similar in many ways to the plot of Akamai data for location A, which is what we would expect. Recall that A and X are the same geographic location but that the experiments are not otherwise comparable – in particular, we cannot compare Figure 4 to Figure 1 to determine which service is performing better.

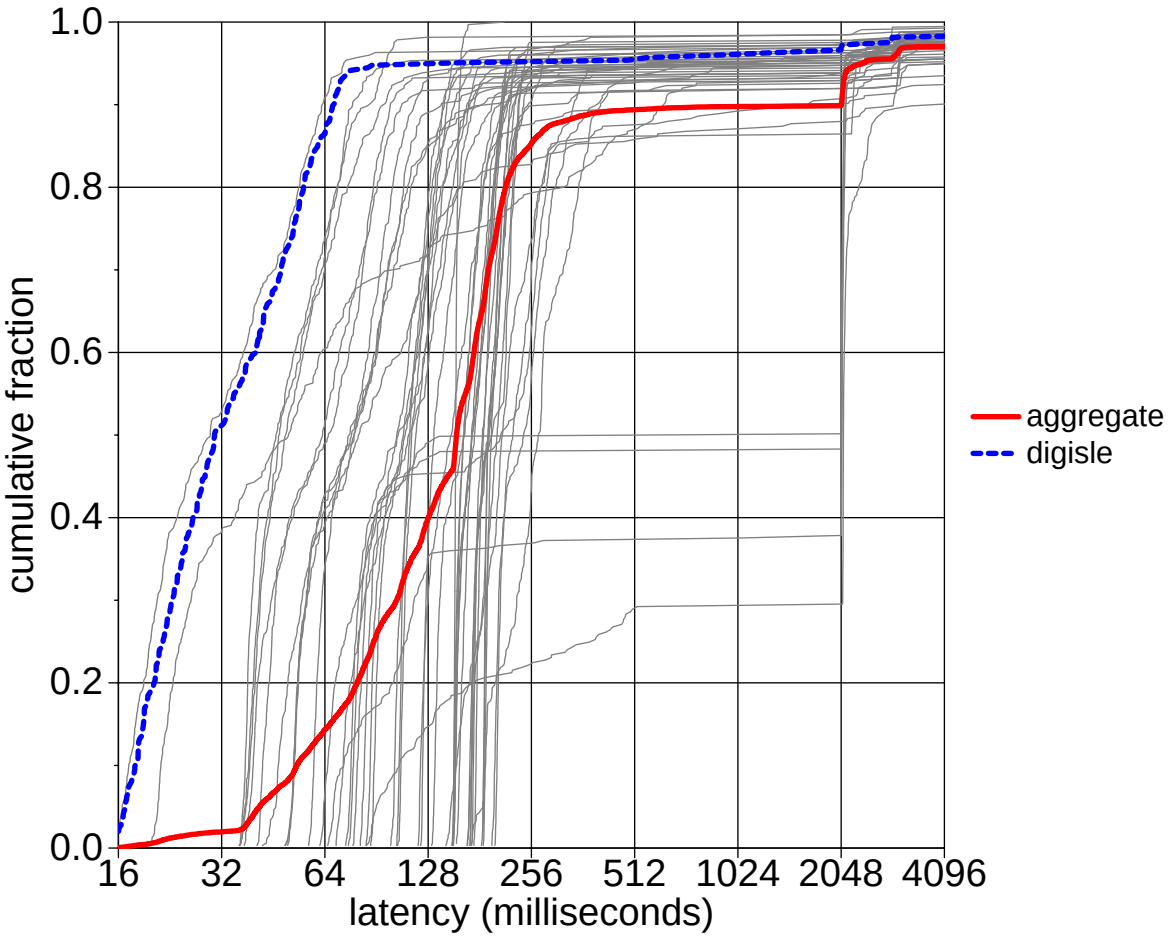

Figure 5. Digital Island latency at location Y

Figure 5 shows the latency measured for Digital Island at location Y. As with Akamai at location B, there is a considerable spread between the latency as delivered by Digital Island and the latency to be expected from a random choice.

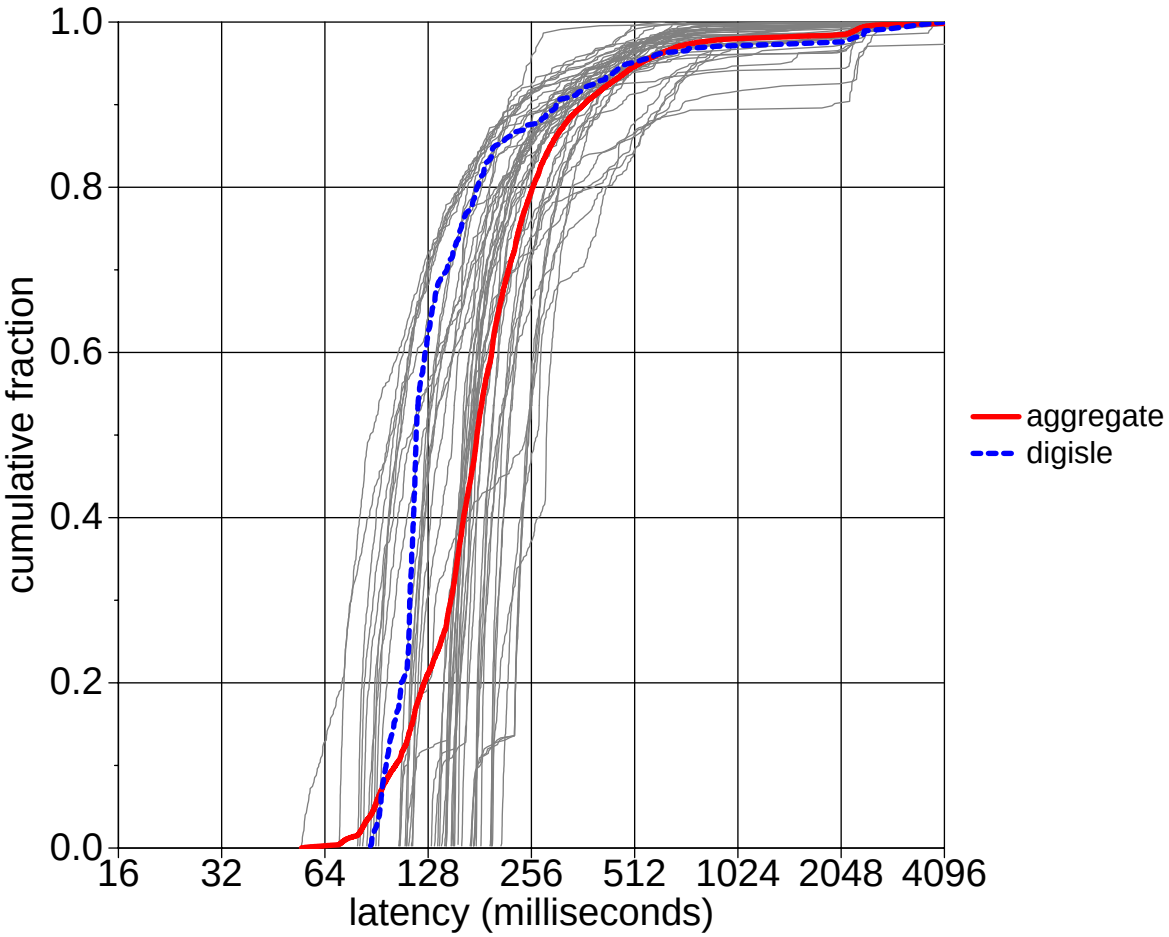

Figure 6. Cumulative distribution of latency, location Z

Finally, Figure 6 shows the latency measured for Digital Island at location Z. As with Akamai at location C, the value of the CDN has dropped considerably – there is a much smaller distance between the performance of Digital Island and the performance possible by choosing randomly. However, it's also clear that the differences among the servers are relatively less conspicuous.

4. Discussion

The data show quite clearly that neither commercial CDN service chooses the optimal server consistently. However, the claimed advantages of a CDN are real and measurable. The performance improvement can be quantified not only by comparison to the performance using the origin server alone, but also by comparison to other choices of server.

Choosing between a server with a latency of 50 ms and one with a latency of 100 ms is not the important problem for a CDN. In the complexity of the real Internet, the true challenge for a CDN is to pick a reasonably good server while avoiding any unreasonably bad servers. Although both measured services mostly do well at this task, both do occasionally pick bad choices. Those bad choices have measured latencies that are almost certainly worse than the latency achievable by going directly to the origin server. Although CDNs are intended to improve performance for end users, in a small fraction of cases these CDNs apparently degrade performance.

5. References

[1] http://www.akamai.com

[2] http://www.digitalisland.com

[3] David Karger, Alex Sherman, Andy Berkheimer, Bill Rogstad, Rizwan Dhanidina, Ken Iwamoto, Brian Kim, Luke Matkins, and Yoav Yerushalmi. Web Caching with Consistent Hashing. Proceedings of the 8th World Wide Web Conference (WWW8). See www.www8.org/w8-papers/2a-webserver/caching/paper2.html.

6. Acknowledgement

The authors gratefully acknowledge Robert Morris's input.

7. Vitae

Kirk Johnson (tuna@sightpath.com) is a Consulting Engineer at SightPath, Inc. His interests include parallel and distributed systems, computer architecture, digital speech and signal processing, data compression, and cryptography. He received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT) in 1995.

John Carr (jfc@sightpath.com) is a Member of Technical Staff at SightPath. He studied planetary science at MIT and went into computer programming. He worked for MIT Athena/Information Systems, and later spent several years as a contractor doing systems and network programming. Among the companies he has worked for are IBM, DEC, Polaroid, and Rational.

Mark Day (mday@sightpath.com) is Senior Scientist at SightPath. He was previously at Lotus, where his work on notification and synchronous groupware contributed to Lotus Sametime and to the IETF working group on Instant Messaging and Presence Protocol. His interests include distributed systems, programming languages, graphic design, and coordination theory. He received his Ph.D. from MIT in 1995.

Frans Kaashoek (kaashoek@sightpath.com) is Chief Scientist at SightPath and an Associate Professor of Electrical Engineering and Computer Science at MIT.