DBit: A Methodology for Comparing Content Distribution Networks

Zahaib Akhtar*, Alefiya Hussain†, Ethan Katz-Bassett*, Ramesh Govindan*
*University of Southern California, †USC/Information Sciences Institute

ABSTRACT

Content distribution networks (CDNs) are increasingly used to deliver high-performance web services. As the volume of content grows, these networks constantly evolve to improve performance. Their performance is difficult to characterize because it depends on factors including the distribution of clients, the geographical scope of the CDN, the specific way in which the content delivery is engineered, and so on. In this paper, we ask the following question: given two CDNs, how do we determine whether one CDN's user-perceived performance is significantly better than another? In a departure from prior work, we propose a methodology called DBit that identifies distributional differences in user-perceived performance, and we apply it to several CDNs that serve photos. Using 14.5 million photo fetches on PlanetLab from three popular CDNs over a period of 8 months, we find that DBit can identify significant performance differences not just between CDNs, but also across time and location. To assess the robustness of our findings, we complement the PlanetLab dataset with data from home networks in 1470 ASes worldwide

1. INTRODUCTION

More than a decade ago, Akamai was the first to build out a large content distribution network (CDN) that hosted content on behalf of other providers. With the recent increase in image and video content, and the advent of cloud-based services, many other companies, such as Google, Microsoft, Facebook, Apple and Netflix all have either built their own CDNs or are reported to be building one [1, 2, 3, 7, 15]. If it is not already true, it is quite likely that in the near future, a very large proportion of Internet traffic will be served by CDNs. For this reason, it is important to understand the performance of these CDNs.

Assessing the performance of a CDN is hard, as it depends upon many factors, such as the placement of the front-end servers in the topology and the quality of their connectivity to clients, the cache sizes and the cache efficacy, the compute capacity of the front-ends, the network connectivity between the front-ends and the back-end, the efficiency of the storage system (e.g., for photos and videos) or the compute system (e.g., for search) at the back-end, and so forth. In addition, while CDNs can get a fairly complete picture of their own infrastructure by instrumenting the entire "stack" [15], assessing the performance of a CDN can be much harder for a third-party. Since requests to a CDN for content terminate at the front-ends, it can be hard to directly measure and understand aspects of a CDN's operation beyond these front-end servers.

As a first step towards understanding CDN performance, in this paper we ask a foundational question: *can we develop a methodology that can discern whether or not two CDNs offer statistically significant performance?* Given two CDNs **A** and **B**, our proposed methodology, called DBit, can answer the question "Is **A** significantly better than **B**?" or the converse "Is **B** significantly better than **A**?". DBit can assess performance for a continuous-valued performance metric such as latency or throughput. Suppose we are interested in the question: does CDN **A** have a significantly better latency than **B**? Conceptually, DBit has three distinct stages which can be collectively used to answer this question. In the first stage, we use multiple distributed vantage points to measure samples of the latency from CDNs **A** and **B**. These samples approximate the distribution of latency seen for each CDN at each vantage point. In the second stage, DBit determines, at each vantage point whether the latency distribution of **A** is statistically better than that of **B**. At the end of this stage, DBit assesses the proportion of vantage points at which **A**'s performance is statistically better than that of **B**. In the third stage, DBit uses a statistical test to determine if this proportion is significantly different than by chance.

To evaluate DBit, we collected 14.5 million photo fetches from three popular CDNs from PlanetLab nodes over a period of 8 months. In addition, to assess potential vantage point bias, we also collected photo latency measurements from the RIPE Atlas measurement platform whose vantage points are located in home networks on more than 1500 ASes. We find that DBit can uncover many significant differences that are consistent with intuition such as the differences between hot and cold fetches, and time-of-day differences. In addition, it is able to find interesting and previously unknown regional variations in performance between CDNs, and differences between overall performance and tail performance.

Moreover, when DBit deems one CDN's latency distribution to be better than another, the magnitude of these differences are often several tens to hundreds of milliseconds. Such differences in magnitude can significantly impact revenue [18].

Such a capability can be used in two ways. First, while companies like Keynote [17] provide other companies with basic web performance measurements from distributed vantage points, a third-party provider like Keynote could adopt this methodology to provide a rigorous basis for interpreting measurement results and assessing the comparative performance of major CDNs. Such an assessment can be one factor in a customer deciding to use a CDN service, or for a CDN to assess the impact of topology or policy changes. Second, a CDN can use this methodology to compare itself against its competitors (for whom it will not have direct access to performance metrics). This can focus the CDN's engineering efforts, and may reveal opportunities for performance differentiation. For example, the CDN may find that it does not perform as well as its competitors in a key geographic area, and may choose to build out more there.

2. HOW DBIT WORKS

We now describe DBit with reference to two CDNs **A** and **B**. Conceptually, DBit has three distinct stages.

Stage 1: DBit obtains active measurements to CDNs from a set of vantage points $\mathcal{V}$. Each active measurement from a node v in $\mathcal{V}$ to **A** or **B** produces one sample of the performance metric of interest (e.g., latency or throughput). Suppose we model this performance metric as a random variable $X_{A,v}$ and $X_{B,v}$ for the two CDNs at each node v. In general, the distributions of X are not known a priori and can be different for different CDNs, because the corresponding performance metric may depend upon many factors, such as client and Photo CDN load, competing network traffic, *etc.*

Stage 2: DBit looks for statistically significant *distributional differences* in the empirical distributions of the random variables X. That is, rather than comparing the CDNs at v using first or second-order statistics, we ask: is the *distribution* of $X_{A,v}$ statistically worse than $X_{B,v}$, denoted by $X_{A,v} > X_{b,v}$? This question can be answered using a specific instance of hypothesis testing: a non-parametric hypothesis test, the two-sample one-sided Kolmogorov-Smirnov test [23] (or K-S test), can confirm or refute the hypothesis that $X_{A,v} > X_{B,v}$, at a specified *significance level* (or, as it is sometimes known, a *p-value*). Intuitively, the significance level indicates a degree of confidence in the verdict. Smaller significance levels are better. For example, a significance level of 0.05 (which we use in this paper) means that the null hypothesis (that $X_{A,v}$ is not statistically worse than $X_{B,v}$) can be rejected with 95% confidence. The output of this stage is a single bit $b_{A,B,v}$ for each node v which is 1 (respectively 0) if the null hypothesis could (respectively could not) be rejected at the given level of significance. Thus, if $b_{A,B,v}$ is 1, then **A**'s performance is statistically worse than **B**'s at the specified p-value.

In practice, particularly for noisy empirical distributions, it is possible for the one-sided K-S test to declare $X_{A,v} > X_{B,v}$ *and* $X_{B,v} > X_{A,v}$. DBit conservatively generates two bits $b_{A,B,v}$ and $c_{A,B,v}$, where the former is defined as above, and $c_{A,B,v}$ is 1 only if the hypothesis that $X_{B,v} > X_{A,v}$ can be *rejected*. The intent is to declare that $X_{A,v} > X_{B,v}$ only when, at node v, **A** is conclusively distributionally worse than **B**.

Stage 3: DBit determines whether the per-node v distributional differences are statistically different across the entire set of vantage points $\mathcal{V}$. The key intuition we use is: if *a significant number* of b_v bits are one *and* a significant number of c_v bits are one, then we say that A performs better than B (assuming that less is better). To do this, we use the binomial test [5], a hypothesis test that determines if a collection of binary outcomes is unlikely to have occurred by chance. The binomial test derives the confidence (again expressed as a significance level or p-value) that the overall outcome (the proportion of 1s across the total population of clients) is significantly different than just by chance (i.e., different than if the probability at each node of a 1 or a 0 were 50%). When the binomial test is true for a low-enough P-value (in our paper, 0.01) we say that (in our example), **A** is better than **B**.

3. APPLYING DBIT TO CDN COMPARISONS

In the rest of the paper, we demonstrate how to apply DBit to compare different CDNs, explore how well it works, and what its conclusions reveal about the state of CDNs today. In general, a CDN may host different services whose notions of performance may be different (e.g., a photo service may depend on latency of photo downloads, while a shared docs service's performance metric may be how quickly writes propagate to participants). Moreover, these performance metrics can be affected by software designs (caching decisions, transport protocol optimizations). So, rather than apply DBit to some generic notion of a CDN, we applied to specific services.

In particular, we focus on latency of access to photos on Facebook, Google+ and Flickr (henceforth collectively called Photo CDNs) (For photos, Facebook uses the Akamai CDN infrastructure in addition to its own CDN, so we are also able to assess Facebook's use of Akamai). Our choice of services is not intended to be exhaustive, and we leave to future work the exploration of other CDNs (e.g., Netflix's CDN for video).

In the following paragraphs, we describe the measurement methodology.

3.1 Methodology

Vantage Points. To measure Photo CDNs, we use two sets of vantage points. The first is PlanetLab, where we use 189 vantage points from distinct sites. As we discuss below, PlanetLab provides much more visibility into components of performance, but may be biased because its nodes are located at universities. To test for bias in our results, we also use a set of vantage points selected from the RIPE Atlas measurement infrastructure, whose nodes are housed at homes of volunteers. This set of 1470 vantage points, each in a distinct AS, is qualitatively different from PlanetLab and representative of home users. Unfortunately, the RIPE Atlas provides a more constrained measurement interface (as discussed below), with limited visibility into different components of performance.

Metrics. We measure the latency of photo downloads for the three Photo CDNs from these vantage points. For photos we measure three forms of latency: *cold-fetch latency*, which is the time taken to download a photo for the first time, *hot-fetch latency* which is the time taken to download a recently accessed photo, and *scaling latency*, which is the time taken to download a *scaled* version of a recently accessed photo.

The reference stream. To measure these forms of latency, we uploaded a total of 600,000 photos, or 200,000 for each of the three providers (Facebook, Google+ and Flickr). We uploaded these photos to newly created accounts and set privacy settings to ensure that these photos could not be accessed by other users of these Photo CDNs. We assigned each Planetlab node 150 unique photos from each Photo CDN to ensure that, when a client fetches one of these photos for the first time, it is truly a "cold" access (i.e., no other client could have accessed it). All photos were JPEG images of the same resolution (960x640) and the same JPEG quality factor (97). We chose fixed photo sizes to ensure that size differences do not significantly affect our latency measurements.

We followed substantially the same methodology for the RIPE vantage points, but with one important difference: all accesses from RIPE vantage points were to photos that were of size 1x1 pixels. Since most RIPE nodes are in home networks, we used smaller photos to minimally disturb home users' perceived network performance. Moreover, because of limitations in the way measurement campaigns can be mounted on RIPE, we were only able to measure hot-fetch latencies from the RIPE vantage points.

We describe below the precise methodology for generating the reference stream to assess hot and cold fetch latency.

URL Generation. After uploading the photos, we acquire the direct CDN URL for each photo by calling the corresponding provider-supported APIs (Facebook's Graph API, Flickr's Data API, Google+ /Picasa Data API). Since Google and Flickr serve photos entirely from their own CDNs, they achieve latency optimizations using DNS redirection: when a client makes a DNS request, the service returns the IP address of a nearby cache server. Thus, the photo URL for Google and Flickr is independent of client location. In contrast, Facebook serves users both from its own CDN and Akamai, so a user may either be directed to Akamai node or to Facebook's own cache server. To ensure that each vantage point's Facebook photo URLs are user-representative for that vantage point, we make API calls using Facebook's Graphs API frome each of the node. We confirm the user-representativeness of URLs by also loging in to Facebook from each of our sites and comparing the API returned URLs with those in user's actual photo albums.

For our scaling latency measurements, vantage points request scaled images with consistent scaling parameters, so that we can meaningfully compare latency measurements across clients. Google+ and Facebook permit dynamic resizing and cropping, by encoding the desired scale in the photo URL. Flickr, by contrast, scales images to a fixed set of sizes when a client uploads them, and it does not permit dynamic scaling.

3.2 Photo CDN Architectures

Before understanding our results and the rest of our methodology, we describe some details of Photo CDN architecture gleaned from published studies and from our own measurements. The general architecture of the three Photo CDNs we study is as follows: Photo CDNs direct client photo fetches to *front-end servers*, and a cache miss results in an access to a *photo back-end*. In what follows, every reference to the Akamai CDN refers to *Facebook's use of Akamai*.

Google+: Recent work has shown that Google has expanded its Web serving infrastructures and directs search requests to satellite front-ends, which relay the requests to back-end data centers [7]. Through measurements spanning the Google address space , we have found that these satellite front-ends also serve requests for Google+ photos. Therefore, Google+ has front-ends at 1400 distinct sites across the world [7].

When a user logs into Google+, the homepage displays feeds from different sources which may include images. When a user clicks on a photo or a shared album feed, Google uses DNS redirection to direct the user to a nearby front-end server [7]. Photo manipulation operations such as scaling or cropping are encoded into the URLs for photos.

Facebook: Facebook uses its own set of cache front-end servers [15] and also relies on Akamai [10] for front-end servers to serve photos around the globe. In our paper, we treat the two sets of cache servers differently, since, as we show, they have different latency properties. Beyond nine known Facebook front-end servers (the edge caches in [15]), we have discovered through active probing 14 additional cache servers belonging to Facebook. Of these additional servers, 1 is in the US and the remaining are in Europe and Asia.[1]

When a user logs in, their timeline may include images served from both Akamai and Facebook's front-end servers. Facebook's published photo back-end design co-locates an "origin" cache with the photo storage subsystem [6]. Facebook also encodes photo scaling operations in the photo URL, with origin caches performing the scaling [15].

Flickr: An analysis of DNS names from our measurements indicates Flickr directs clients to three photo back-ends. These

[1] Seattle, Amsterdam, Paris, Frankfurt, Hong Kong, Kuala Lumpur, London, Lulea (Sweden), Madrid, Milan, Tokyo, Sao Paulo, Singapore, Vienna

locations host three of the five known Yahoo! data centers [9]. Flickr also uses DNS redirection, but does not support dynamic scaling of photos.

3.3 Latency Measurements

From each PlanetLab vantage point, we periodically fetch the URLs for photos from each Photo CDN using cURL. cURL provides separate latency values for the DNS resolution, TCP connection setup, time between HTTP GET and first byte and time to download the entire photo.

To measure the hot-fetch latency, each vantage point takes an exclusive subset of URLs for each Photo CDN, and repeatedly fetches each of those URLs every 30 minutes. Every fetch after the first is deemed a hot-fetch. As described earlier, Photo CDNs cache recent fetches. Our choice of measurement interval increases the likelihood that a hot-fetch was served from the cache, without increasing load on the service providers. As we show later, the latency distributions of hot-fetch and cold-fetch are significantly different, validating our methodology.

To measure cold-fetch latency, clients use the remaining URLs[2] (those not used for hot-fetches) and sequentially retrieve one of them an hour. With a set of 150 photos per node the inter-access interval between cold-fetches is 6 days, likely long enough for the content to have been flushed from any caches. From preliminary experiments on Facebook, we found that eviction time at various caches was between 52 and 60 hours, less than half our cold-fetch interval.

To measure scaling latency, clients fetch a cropped version of each hot-fetched image, immediately after fetching the non-cropped version. Scaling can increase latency because it imposes processing on the photo fetch path, and can either be performed at the back-end or at the cache server, and each choice impacts latency differently.

All our experiments ran over a period of eight months during which we retrieved a total of 14.5 million photo fetches across all PlanetLab vantage points. Our latency measurements for photos from RIPE Atlas ran for a period of 12 days.

4. EVALUATION

In this section, we evaluate DBit. DBit is a methodology for comparing CDNs, but can also be thought of as a binary metric for CDN comparison. It is difficult to evaluate metrics per se, especially when there is no prior metric to compare against. Moreover, since DBit relies on the KS-test and the Binomial test, accepted statistical tests for testing distributional differences and testing for biased coins, respectively, we do not focus on whether its ability to discriminate is sound but rather on, is DBit useful, using the vantage points and performance metrics we deploy? Specifically, we resort to a qualitative evaluation that explores two questions:

- Does the metric correctly identify differences known to be (likely) true and hide differences known not to exist? In other words, does the metric conform to intuition?
- Does the metric identify differences between CDNs for which there exists at least one plausible explanation for the difference based on what we know about how these CDNs work (Section 3)?

Especially when addressing the second question, we come up with interesting (and previously unknown) performance differences between CDNs as they are today. While we offer plausible explanations for many of these, future work can conduct rigorous experiments to ascertain the causes for these differences.

In what follows, we report results on applying this metric to various aspects of Photo CDN performance, beyond hot and cold fetch performance differences. We examine time of day and regional differences in latency performance, outliers in the measurement infrastructure, and tail performance.

With respect to the CDN differences we do observe, we emphasize two points. First, CDNs are continuously evolving entities, so any differences we observe today may not hold in the future. Second, as discussed previously, DBit itself cannot *explain* the differences we see, and additional measurements are needed for that. Doing these additional measurements is left to future work, since, for each observed difference, we will likely need to design a separate measurement campaign to explain that observed difference. More important, any plausible explanations we offer for the differences should be treated as hypotheses that need to be verified by actual experiments.

Methodology Validation Results								
	Akamai		Facebook		Flickr		Google+	
	H >C	C >H	H >C	C >H	H >C	C >H	H >C	C >H
Aggregate	0	0.9864	0.034	0.7007	0	0.9116	0	0.9796
P-value	1.1e-44	1.2e-40	3.0e-33	8.1e-08	1.1e-44	1.1e-29	1.1e-44	5.9e-39
Significant	✓		✓		✓		✓	

Table 4—KS-test and Two tailed P-values from binomial test at 99% significance level to validate Hot fetch Distribution always lies to the left of Cold fetch distribution for all Photo CDNs. Aggregate values represent the summed of fraction KS-Test Test Results.

Hot vs. Cold Fetches. We first compare the hot and cold latency distribution for each Photo CDN for all PlanetLab vantage points. Intuitively, if our methodology for issuing hot and cold fetches correctly captures caching behavior, the hot fetch distribution for a client should lie to the left (meaning faster performance) of the client's cold fetch distribution for the same Photo CDN. Table 4 shows the results of the comparison. We frame each comparison as a hypothesis. For example, the Akamai column shows two hypotheses: H >C and C >H, where H and C denote hot and cold fetches respectively. The Aggregate row shows the total fraction of nodes which reject the given hypothesis, so under the hypothesis H >C, all nodes reject the hypothesis leading to a value of ~ 0. Conversely, the high aggregate value under

[2]Hot-fetches are distinguished from cold-fetches only by the time at which the fetch is made, not by any syntatic differences in the URL.

Global Hot Fetch Comparison: Photo													
		Facebook Vs Akamai		**Facebook Vs Google+**		**Facebook Vs Flickr**		**Flickr Vs Akamai**		**Flickr Vs Google+**		**Akamai Vs Google+**	
		FB >Ak	Ak >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Global	**Aggregate**	0.4136	0.2407	0.537	0.1667	0.1296	0.6481	0.8086	0	0.9444	0	0.5556	0.1605
	P-value	3.4e-02	2.5e-11	3.9e-01	1.8e-18	5.1e-23	2.0e-04	7.9e-16	3.4e-49	6.1e-35	3.4e-49	1.8e-01	3.5e-19
	Significant?	✗		✗		✓		✓		✓		✗	
N. America Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
N. America	**Aggregate**	0.2537	0.3731	0.3731	0.2388	0.0149	0.7761	0.9104	0	0.9701	0	0.5522	0.209
	P-value	6.7e-05	5.0e-02	5.0e-02	2.2e-05	9.2e-19	6.5e-06	1.5e-12	1.4e-20	3.1e-17	1.4e-20	4.6e-01	1.8e-06
	Significant?	✗		✗		✓		✓		✓		✗	
Europe Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Europe	**Aggregate**	0.4937	0.1646	0.6709	0.1139	0.2278	0.5949	0.7595	0	0.962	0	0.5696	0.1139
	P-value	1.0e+00	1.1e-09	3.2e-03	7.8e-13	1.3e-06	1.2e-01	4.2e-06	3.3e-24	2.7e-19	3.3e-24	2.6e-01	7.8e-13
	Significant?	✗		✓		✗		✓		✓		✗	

Table 1—Hot fetch KS-Test for the Planetlab client using a significance level of 0.01 for the Binomial Test. The results from the complete set, North American and European clients are shown.

Global Cold Fetch Comparison: Photo													
		Facebook Vs Akamai		**Facebook Vs Google+**		**Facebook Vs Flickr**		**Flickr Vs Akamai**		**Flickr Vs Google+**		**Akamai Vs Google+**	
		FB >Ak	Ak >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Global	**Aggregate**	0.1296	0.6667	0.4938	0.3519	0.0185	0.9074	0.8519	0.0123	0.9691	0.0062	0.8642	0.0432
	P-value	5.1e-23	2.7e-05	9.4e-01	2.0e-04	2.4e-43	2.1e-28	1.2e-20	4.5e-45	3.1e-40	5.6e-47	3.3e-22	1.8e-37
	Significant?	✓		✗		✓		✓		✓		✓	
N. America Cold Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
N. America	**Aggregate**	0	0.8507	0.3134	0.5373	0	0.9403	0.806	0.0149	0.9403	0	0.8657	0.0448
	P-value	1.4e-20	4.0e-09	3.1e-03	6.3e-01	1.4e-20	1.1e-14	4.5e-07	9.2e-19	1.1e-14	1.4e-20	6.8e-10	6.8e-16
	Significant?	✓		✗		✓		✓		✓		✓	
Europe Cold Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Europe	**Aggregate**	0.1772	0.5696	0.6076	0.2278	0.0253	0.8734	0.8734	0.0127	1	0.0127	0.9241	0.0253
	P-value	5.2e-09	2.6e-01	7.1e-02	1.3e-06	1.1e-20	5.5e-12	5.5e-12	2.7e-22	3.3e-24	2.7e-22	1.0e-15	1.1e-20
	Significant?	✗		✗		✓		✓		✓		✓	

Table 2—Cold fetch KS-Test for the Planetlab client using a significance level of 0.01 for the Binomial Test. The results from the complete set, North American and European clients are shown.

the hypothesis C >H signifies that an overwhelming majority of the nodes accepted the hypothesis that the distribution of cold fetches is indeed worse than hot fetches. We then test the resulting aggregate values using a binomial test to determine if chance could have explained these aggregates. The resulting P-values, shown in the next row, are negligibly small, ruling out the possibility of chance with high probability. Thus, we can conclude that, for Akamai, hot fetches are distributionally better than cold fetches.

By similarly examining other columns, we see that this distributional difference persists across other Photo CDNs. Thus, DBit correctly identifies significant statistical differences between hot and cold fetches. Moreover, this experiment demonstrates that DBit can be used not just to compare CDNs, but to compare different aspects of performance of a given CDN.

Hot-Fetch Photo Latency. Table 1 shows the all-pairs KS-test and the accompanying binomial test results for hot-fetch latency, calculated over all Planetlab clients. The results reveal that, for hot content, Flickr's performance is statistically worse than all other Photo CDNs i.e. Facebook, Google+ and Akamai, as evident from the higher aggregate values in the respective columns and the low P-values from the binomial tests. On the other hand, for hot-fetches, Facebook is statistically indistinguishable from Akamai and Google+, and Akamai is statistically indistinguishable from Google+. In summary, for hot-fetches, we can establish a partial ordering across Photo CDNs: $\{$Ak,G+,FB$\} \succ \{$Fl$\}$ from the full set of Planetlab Clients. One plausible explanation for this difference is the significantly smaller number of cache server sites for Flickr, compared to the number for the other CDNs, leading to longer network latencies for some clients.

Regional Differences. With DBit, we are also able to observe regional differences, by conditioning the distributions on vantage points in different continents. In Europe, Facebook's hot fetch performance is inferior compared to Google+, whereas as with global result, both are statistically indistinbuishable from Akamai. The test between Facebook and Flickr is indistinguishable, giving us the following partial ordering in Europe: Photo CDNs: $\{$Ak,G+$\} \succ \{$FB,Fl$\}$. However, the partial ordering in North America is consistent with that of the global set. Thus, Facebook compares favorably with Google+ in North America, but not in Europe. There is at least one plausible explanation for this: Facebook is US-based, with large data centers on both coasts, near the largest centers of population, and its careful engineering of the photo stack [15] may compensate for Facebook's relatively fewer front-ends, but this compensation does not seem

Global Hot Fetch Comparison: RIPE Atlas like Measurements													
		Facebook Vs Akamai		**Facebook Vs Google+**		**Facebook Vs Flickr**		**Flickr Vs Akamai**		**Flickr Vs Google+**		**Akamai Vs Google+**	
		FB >Ak	Ak >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Global	**Aggregate**	0.6358	0.2469	0.7099	0.2407	0.3333	0.6173	0.9259	0.037	0.9506	0.0432	0.4877	0.3827
	P-value	6.8e-04	7.7e-11	9.2e-08	2.5e-11	2.7e-05	3.5e-03	1.7e-31	8.1e-39	3.6e-36	1.8e-37	8.1e-01	3.5e-03
	Significant?	✓		✓		✓		✓		✓		✗	
N. America Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
N. America	**Aggregate**	0.4925	0.3134	0.6119	0.2985	0.2239	0.7463	0.8955	0.0448	0.9552	0.0448	0.5224	0.2687
	P-value	1.0e+00	3.1e-03	8.6e-02	1.3e-03	6.5e-06	6.7e-05	1.3e-11	6.8e-16	6.8e-16	6.8e-16	8.1e-01	1.9e-04
	Significant?	✓		✓		✗		✓		✓		✗	
Europe Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Europe	**Aggregate**	0.7468	0.1899	0.7975	0.1772	0.3924	0.5443	0.962	0.0127	0.9873	0	0.4557	0.4557
	P-value	1.3e-05	2.3e-08	9.4e-08	5.2e-09	7.1e-02	5.0e-01	2.7e-19	2.7e-22	2.7e-22	3.3e-24	5.0e-01	5.0e-01
	Significant?	✓		✓		✗		✓		✓		✗	

Table 3—Hot Fetch Results from Planetlab using 320 Samples from each Photo CDN. The photos used here were the same as those used for RIPE and latency metric is the sum of TCP and GET time (time to first byte). The green cells above show the results which are consistent with the RIPE Results in 8

to be sufficient in Europe where it has only one datacenter in Sweden and seven edge caches compared to 10 in the US. (§3.2)

Finally, we should note that our current set of Planetlab vantage points does not include any in Africa, only five in South-America, and about 15 in Asia. Thus, our "global" comparisons might change with a larger measurement infrastructure and representation from these regions.

Vantage point bias. Are PlanetLab vantage point biased? We evaluate our vantage point selection by comparing our hot-fetch results for PlanetLab with those from RIPE. Recall that, on the RIPE Atlas, we are only able to obtain hot-fetch, because of platform limitations (§ 3). Moreover, on the RIPE Atlas platform, we are only able to obtain TCP and GET times. So, in comparing these two sets, we re-run DBit on PlanetLab, but only use the TCP and GET (time between HTTP GET and first Byte) times, so we can meaningfully compare those two sets. Table 8 shows that the RIPE results are almost entirely consistent with the PlanetLab results shown in Table 3, both globally and in North America and Europe. Of the 18 different comparisons, the two of them differ *only* in 1 of the comparisons, the one between Akamai and Google+. We are currently investigating the cause of this single difference, but we conclude from this that there is substantial agreement between the two vantage point sets, indicating that PlanetLab is not biased (or, at least, if it is, it is biased similarly to the much larger set of RIPE Atlas probes, which are less focused in educational networks).

Cold-Fetch Latency. Table 2 compares cold-fetch latency across Photo CDNs. In this case, the results are qualitatively different from hot-fetches: Facebook is distributionally better than both Flickr and Akamai, and distributionally indistinguishable from Google+. Flickr is distributionally worse than both Akamai and Google+, and Google+ is distributionally better than Akamai. This results in the following partial ordering: $\{\text{FB,G+}\} \succ \text{Ak} \succ \text{Fl}$.

Recall that a cold-fetch request travels to a photo back-end. However, given the way these Photo CDNs are architected, we have no visibility into the back-end dynamics. We do know that Facebook has invested significant effort in engineering the back-end [6, 15]. These engineering decisions represent a plausible explanation for Facebook's superior performance, and likely Google+ employs similar techniques given its comparable performance. Because Flickr performs qualitatively worse, we believe it does not incorporate some or all of these optimizations. Finally, Akamai cold-fetches have to be routed to Facebook back-ends, and we hypothesize that there might be some non-optimality in that path (which Facebook cannot control or engineer).

Inter-Photo CDN Photo Scaling Comparison							
		Facebook Vs Akamai		**Facebook Vs Google+**		**Akamai Vs Google+**	
		FB >Ak	Ak >FB	FB >G+	G+ >FB	Ak >G+	G+ >Ak
Global	**Aggregate**	0.247	0.7229	0.8614	0.1205	0.9759	0.0181
	P-value*	4.5e-11	8.4e-09	2.3e-22	7.8e-25	6.7e-43	1.6e-44
	Significant?	✓		✓		✓	

Table 5—Inter-Photo CDN KS test Results for Image Scaling

Regional Differences. Facebook's cold-Fetch comparison in Europe matches the results for the hot fetches. Observe that Facebook's comparison with Akamai in Europe is indistinguishable as opposed to a win for Facebook in the Global set. Similarly the comparison with Google+ shows that Facebook performs worse than Google+ with respect to cold fetches in Europe. This is an interesting insight, and suggests that Facebook's recent latency optimizations in North America [15] (likely motivated by revenue concerns) have paid off.

Do the differences matter? DBit cannot itself indicate the magnitude of the difference, so we resort to a different methodology to understand this. For each of our vantage points where the comparison favors the "winning" CDN, we compute the median latency values and plot the resultant CDF. Figure 2 shows the CDF for Google+ in comparison to other Photo CDNs. Observe that for hot fetches the worst case latency difference between Google+ and Flickr can be as large as ∼3 seconds, and for hot fetches this can be over 5 seconds. Even for the other services, there is a substantial difference. For more than half of the vantage points at which Google+ is better than Facebook, the median latency of Facebook is almost 0.5 seconds higher than that of Google+. Since revenue

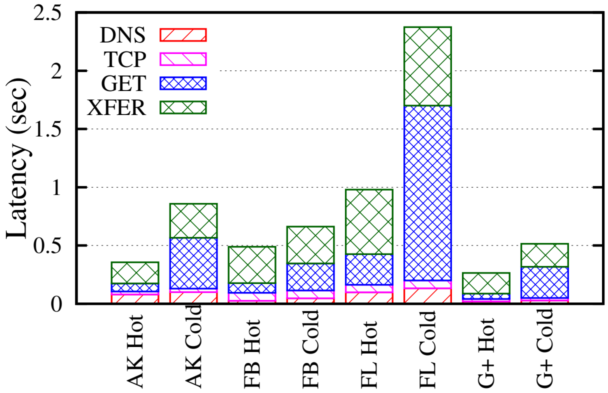

Figure 1—Average Latency of each stage in the GET request for both hot and cold fetches for all Photo CDNs

Time of Day Variation Detected by DBit								
	Akamai		Facebook		Flickr		Google+	
	P2 >P5	P5 >P2	P2 >P5	P5 >P2	P2 >P5	P5 >P2	P2 >P5	P5 >P2
Aggregate	0.1296	0.9444	0.3889	0.8519	0.2407	0.9259	0.5926	0.8704
P-val	2.3e-08	2.9e-12	1.3e-01	1.4e-07	1.8e-04	3.8e-11	2.2e-01	2.3e-08
Significant	✓		✗		✓		✗	

Table 7—Time of Day Comparison for Photo Hot Fetches for all Photo CDNs

is believed to be impacted by latency [18], these differences can matter from a business perspective.

What components of latency dominate? For both cold and hot fetches, accesses to photos incur several components: DNS resolution, TCP connection setup, the GET time (the time between issuing the HTTP request and getting the first byte) and the transfer time. Figure 1 shows the relative proportion of the contributions of each of these components for each of CDNs; the stacks represent averages for these components. We see that TCP connection setup times are almost negligible, and DNS resolution times across all the CDNs are comparable. Different CDNs distinguished themselves by optimizing GET times and transfer times, which are also the two most dominant components for all the CDNs. As expected, in every case, the GET time is significantly higher for cold fetches than for hot fetches; this added latency reflects the time to access the back-end storage. In most cases, the transfer time is comparable for cold and hot fetches. This suggests that the bottleneck for almost every CDN may be between the cache server and the vantage point.

Tail latency. Because latency can impact revenue, content providers are interested in engineering for tail latency [12]. DBit can also be used to study whether the tail latency distributions of different CDNs differ significantly, by appropriately truncating the original distributions. Table 6 shows that for hot fetches, while Google+, Facebook, and Akamai are indistinguishable when we consider the complete global distribution, Google+ is better than the other two for the tail latency. For cold fetches, Google+ and Facebook were indistinguishable when considering the complete global distribution, but Google+ has better tail latency.

Time of Day Variability. Internet traffic and server load is known to follow diurnal patterns. DBit can be used to understand two aspects of diurnal variability. First, the above global cold and hot fetch differences are computed across measurements obtained at different times of day. Are time-of-day differences a possible explanation for these differences? To understand this question, we divided the 24 hour day into 4 hour periods (labeled **P1** to **P6**), with **P1** from 2:30am-6:29am, **P2** from 6:30am-10:29am and so on (these periods are determined by local time at the corresponding client's timezone). We then repeated the DBit comparisons for all CDNs for both cold and hot fetch latencies across each time period. We found (results omitted for brevity) that, in every period, the results were identical with the cold and hot fetch results described above. This suggests that time-of-day differences are unlikely to be an explanation for the performance differences discussed earlier.

Second, we use DBit to understand whether a given CDN performs better at some periods than at other periods. For each Photo CDN, we compare latency distributions pairwise between periods for each type of fetch using a subset of our clients which lie in the same time zone. Our methodology detects that for all Photo CDNs there is a clear performance difference between **P2**, **P5** and **P6**. We show the result for hot photo fetches in Table 7 for the comparison between **P2** and **P5**. **P6** is omitted for brevity. **P2**, the early morning hours between 6:30am-10:29am, sees better performance than evening (6:30pm-10:29p.m) and night hours (10:30pm-2:29am) of **P5** and **P6** respectively. Akamai and Flickr experience the greatest performance difference. The worse performance during evening hours (6:30pm-10:29pm) may either be because photo sharing is fundamentally a social activity and/or because Akamai's other clients are accessed predominantly at these times. Thus, DBit is able to uncover diurnal performance differences that conform to intuition.

Photo Scaling Latency. In order to compare dynamic scaling of photo content, we fetch a scaled version of a photo just after completing the regular fetch. In Table 5, which does not include Flickr (§3.1), Google+ is distributionally better than Facebook. Facebook scaling requests travel to the photo back-end [15], but Google+ and Facebook are distributionally indistinguishable for cold-fetches (whose requests also travel to the back-end). So this statistical difference suggests Google+ may be scaling photos in its cache servers. In North America, however, Google+ is distributionally indistinguishable from Facebook (table omitted for brevity) in scaling performance, suggesting that cache server to back-end paths dominate performance in other parts of the globe.

Outliers: Using DBit, we can also detect outliers: "good" clients whose latency distributions (hot and cold fetch combined) for a given Photo CDN are significantly better than all other clients, and "bad" clients whose latency distributions are significantly worse than all other clients. We find that most of the good clients are in North America or Europe. Two of our clients, located in Boston and Washington, are best across all Photo CDN. This may be because the sites

		Facebook Vs Akamai		Facebook Vs Google+		Facebook Vs Flickr		Flickr Vs Akamai		Flickr Vs Google+		Akamai Vs Google+	
Global Hot Fetch Comparison: Photos 90th %ile													
		FB >Ak	Ak >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Global	**Aggregate**	0.3457	0.4506	0.7037	0.216	0.1173	0.7963	0.7778	0.0617	0.963	0.0123	0.784	0.1543
	P-value	1.1e-04	2.4e-01	2.3e-07	1.9e-13	1.0e-24	1.3e-14	6.6e-13	9.5e-34	8.1e-39	4.5e-45	1.9e-13	6.6e-20
	Significant?	✗		✓		✓		✓		✓		✓	
Global Cold Fetch Comparison: Photos 90th %ile													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Global	**Aggregate**	0.0556	0.9506	0.6975	0.2716	0.0185	0.9815	0.7099	0.0494	0.9877	0.0062	0.9691	0.0185
	P-value	6.1e-35	3.6e-36	5.4e-07	5.3e-09	2.4e-43	2.4e-43	9.2e-08	3.6e-36	4.5e-45	5.6e-47	3.1e-40	2.4e-43
	Significant?	✓		✓		✓		✓		✓		✓	

Table 6—Tail latency for both hot and cold photo fetches from the global set of Planetlab Clients

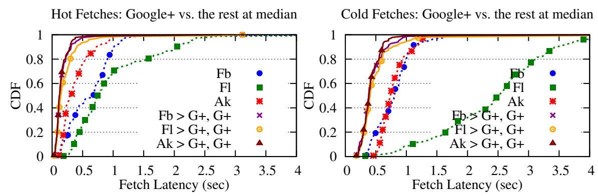

Figure 2—Do the differences matter? Distribution of medians for Google+ in comparison to other Photo CDNs

have better Internet connectivity or the PlanetLab nodes at these sites are close to the corresponding Photo CDN's cache server. We find other good performing clients in Canada and USA (4), Germany (3), France (2), Ireland and UK one each. The worst client in our set for Photo CDNs is located in Cyprus, and we find multiple bad clients in Spain (3), Portugal (3), Greece (2), Brazil (2), Ecuador (1) and Jordan (1).

5. RELATED WORK

One piece of research develops a model for indirectly measuring the front-end to back-end latency for popular search engines [8]. By contrast, DBit proposes a methodology to rigorously compare a variety of aspects of CDN performance. Other notable work is centered on understanding peer-to-peer content delivery systems [20] and ISPs [19]. By contrast, our work focuses on a different setting (photo and video sharing) and on one metric, latency, but explores distributional differences. Finally, Tariq et al. [22] propose a What-If scenario evaluator to understand how changes to network infrastructure will translate to user perceived performance. By contrast, DBit can help a CDN understand what aspects of its network infrastructure it needs to improve in order to match a competitor's performance.

Our work is inspired by recent work on optimizing photo fetch performance [15]. An earlier paper [6] describes the design of the photo back-end storage server. In contrast, our work describes a black-box methodology for comparing Facebook and other Photo CDNs.

Other work has explored or measured latency optimizations in a variety of contexts, identifying a repeatedly-used design pattern of back-end data centers and front-end satellites/caches. This pattern has been used by Google to improve Web latency [7], by Yahoo! for access to its services [9], by Akamai for generic content distribution [10], and Youtube to optimize video bandwidth and latency [13]. We are not aware of work that has attempted to devise a methodology to not only compare Photo CDNs but also to generalize it to other latency metrics.

6. CONCLUSION

In this paper, we propose a novel methodology to compare different aspects of CDN performance. DBit is able to bring out interesting and plausible cold and hot fetch performance differences, time-of-day differences and regional differences in performance. While DBit can assess statistically significant differences, it turns out that the magnitude of the performance difference is several tens or hundreds of milliseconds, which can impact user satisfaction. It is also able to assess whether providers differ significantly in the way they handle the tail of the latency distribution. DBit is but a first step towards a comprehensive methodology for CDN evaluation and much work remains: exploring other CDNs (like Netflix's video CDN), developing DBit capable methodologies for assessing video download efficacy and understanding chunking strategies, and understand systematic ways in which one can explore the causes of the differences indicated by DBit.

7. REFERENCES

[1] Apple building its own CDN. http://blog.streamingmedia.com/2014/02/apple-building-cdn-software-video-delivery.html, 2012.

[2] Business Insider: NetFlix building its own CDN. http://goo.gl/PXi8yJ, 2012.

[3] Microsoft Azure Content Delivery Network. http://blog.streamingmedia.com/2014/02/apple-building-cdn-software-video-delivery.html, 2012.

[4] New York Times: Facebook Buys Instagram for $1 Billion. http://dealbook.nytimes.com/2012/04/09/facebook-buys-instagram-for-1-billion/, 2012.

[5] GraphPad Software - Binomial Test. http://www.graphpad.com/guides/prism/6/statistics/index.htm?stat_binomial.htm, 2013.

[6] D. Beaver, S. Kumar, H. C. Li, J. Sobel, P. Vajgel, and F. Inc. Finding a needle in haystack: Facebooks photo storage. In *In Proc. of OSDI*, 2010.

[7] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan. Mapping the Expansion of Google's Serving Infrastructure. In

		Facebook Vs Akamai		Facebook Vs Google+		Facebook Vs Flickr		Flickr Vs Akamai		Flickr Vs Google+		Akamai Vs Google+	
Global Hot Fetch Comparison: RIPE Atlas													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Global	**Aggregate**	0.6814	0.2115	0.7416	0.1593	0.2851	0.5582	0.8568	0.0535	0.8768	0.0469	0.5422	0.2851
	P-value	1.7e-23	3.2e-59	3.1e-41	2.4e-84	1.2e-32	1.6e-03	2.5e-93	0.0e+00	0.0e+00	0.0e+00	2.3e-02	1.2e-32
	Significant?	✓		✓		✓		✓		✓		✓	
Africa Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Africa	**Aggregate**	0.5758	0.303	0.7576	0.2121	0.3333	0.5758	0.6061	0.1212	0.7273	0.0606	0.6667	0.2424
	P-value	4.9e-01	3.5e-02	4.6e-03	1.3e-03	8.0e-02	4.9e-01	3.0e-01	1.1e-05	1.4e-02	1.3e-07	8.0e-02	4.6e-03
	Significant?	✓		✓		✗		✗		✓		✗	
Asia Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Asia	**Aggregate**	0.7308	0.1282	0.7949	0.0641	0.3205	0.4103	0.7949	0.1282	0.8077	0.0769	0.3333	0.3718
	P-value	5.6e-05	9.7e-12	1.5e-07	1.5e-16	2.0e-03	1.4e-01	1.5e-07	9.7e-12	3.8e-08	1.9e-15	4.3e-03	3.1e-02
	Significant?	✓		✓		✗		✓		✓		✗	
Europe Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Europe	**Aggregate**	0.7365	0.1597	0.7844	0.1257	0.2874	0.5549	0.9202	0.016	0.9301	0.0259	0.5689	0.2834
	P-value	5.7e-27	6.5e-57	4.7e-39	3.8e-70	6.5e-22	1.6e-02	8.2e-92	0.0e+00	3.0e-97	0.0e+00	2.4e-03	1.0e-22
	Significant?	✓		✓		✓		✓		✓		✓	
North America Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
N. America	**Aggregate**	0.3778	0.4667	0.3778	0.4444	0.1333	0.7333	0.8	0.0444	0.8222	0.0667	0.5111	0.2556
	P-value	2.6e-02	6.0e-01	2.6e-02	3.4e-01	5.2e-13	1.1e-05	8.1e-09	4.3e-21	4.4e-10	1.1e-18	9.2e-01	3.8e-06
	Significant?	✗		✗		✓		✓		✓		✗	
Oceania Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
Oceania	**Aggregate**	0.7917	0.25	0.9583	0.0417	0.7917	0.1667	0.3333	0.5417	0.2917	0.3333	0.625	0.3333
	P-value	6.6e-03	2.3e-02	3.0e-06	3.0e-06	6.6e-03	1.5e-03	1.5e-01	8.4e-01	6.4e-02	1.5e-01	3.1e-01	1.5e-01
	Significant?	✗		✓		✓		✗		✗		✗	
S. America Hot Fetch Comparison													
		FB >Ak	AK >FB	FB >G+	G+ >FB	FB >Fl	Fl >FB	Fl >Ak	Ak >Fl	Fl >G+	G+ >Fl	Ak >G+	G+ >Ak
S. America	**Aggregate**	0.5238	0.4762	0.8095	0.1429	0.0952	0.8571	0.8095	0.0476	1	0	0.5238	0.1429
	P-value	8.3e-01	1.0e+00	7.2e-03	1.5e-03	2.2e-04	1.5e-03	7.2e-03	2.1e-05	9.5e-07	9.5e-07	8.3e-01	1.5e-03
	Significant?	✗		✓		✓		✓		✓		✗	

Table 8—KS-Test Results for Photo Hot Fetch, Global and region wise for RIPE Atlas probes. Red boxes show comparisons which differ from the global result. Green cells show distinguishable comparisons

Proceedings of the ACM Internet Measurement Conference (IMC '13), October 2013.

[8] Y. Chen, S. Jain, V. K. Adhikari, and Z.-L. Zhang. Characterizing roles of front-end servers in end-to-end performance of dynamic content distribution. In *Internet Measurement Conference*, pages 559–568. ACM, 2011.

[9] Y. Chen, S. Jain, V. K. Adhikari, Z.-L. Zhang, and K. Xu. A first look at inter-data center traffic characteristics via yahoo! datasets. In *INFOCOM*, pages 1620–1628. IEEE, 2011.

[10] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl. Globally distributed content delivery. In *In IEEE Internet Computing*, 2002.

[11] Facebook. A Focus on Efficiency. https://fbcdn-dragon-a.akamaihd.net/hphotos-ak-prn1/851575_520797877991079_393255490_n.pdf, 2013.

[12] T. Flach, N. Dukkipati, A. Terzis, B. Raghavan, N. Cardwell, Y. Cheng, A. Jain, S. Hao, E. Katz-Bassett, and R. Govindan. Reducing web latency: The virtue of gentle aggression. In *Proceedings of the ACM SIGCOMM 2013*, SIGCOMM '13, pages 159–170. ACM, 2013.

[13] P. Gill, M. F. Arlitt, Z. Li, and A. Mahanti. Youtube traffic characterization: a view from the edge. In *Internet Measurement Conference*, pages 15–28. ACM, 2007.

[14] C. Huang, A. Wang, J. Li, and K. W. Ross. Measuring and evaluating large-scale cdns. Technical Report MSR-TR-2008-106, Microsoft, 2008.

[15] Q. Huang, K. Birman, R. van Renesse, W. Lloyd, S. Kumar, and H. C. Li. An analysis of facebook photo caching. In *SOSP*, pages 167–181. ACM, 2013.

[16] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In *Proceedings of the ACM SIGCOMM 2014*, SIGCOMM '14, pages 187–198, Chicago, USA, 2014.

[17] Keynote. KeyNote. http://www.keynote.com/solutions/testing/load-testing, 2012.

[18] G. Linden. Make data useful. http://sites.google.com/site/glinden/Home/StanfordDataMining.2006-11-28.ppt, 2006.

[19] R. Mahajan, M. Zhang, L. Poole, and V. S. Pai. Uncovering performance differences among backbone isps with netdiff . In *NSDI*, pages 205–218. USENIX Association, 2008.

[20] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An analysis of internet content delivery systems. In *5th Symposium on Operating*

Systems Design and Implementation (OSDI) 2002, Boston, MA, USA, December 2002.

[21] S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft. Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 457–466, New York, NY, USA, 2011. ACM.

[22] M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering what-if deployment and configuration questions with wise. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, SIGCOMM '08, pages 99–110, New York, NY, USA, 2008. ACM.

[23] E. W. Weisstein. Kolmogorov-Smirnov Test. http://mathworld.wolfram.com/Kolmogorov-SmirnovTest.html, 2013.