

محاسبه یک شبکه تحویل محتوای تجاری

چکیده

امروزه شبکه های تحویل محتوا (CDNs)، بخش مهمی از زیر ساخت WEB پیشرفته شده اند. در این مقاله، عملکرد ارائه دهنده محتوای Akamai بررسی می شود. به این صورت که عملکرد پلتفرم موجود Akamai سنجیده شده و از دیدگاه طراحان CDN و مشتریان احتمالی، سوالی کلیدی و ساختاری مطرح می شود به این صورت که آیا پلتفرم های منطبق با Akamai که سعی در گسترش سرور ها در مکان های مختلف اینترنت دارند، از مزایای عملکرد اصلی روی مرکز داده ای یکپارچه تر که توسط سایر CDNs های قدرتمند نظیر Limelight پشتیبانی می شوند برخوردار هستند یا خیر؟ اعتقاد ما بر این اساس است که روشی که در این مقاله شرح داده ایم، برای محققین عرصه CDN مفید خواهد بود.

دسته بندی ها و تعاریف

C.2.1 شبکه های ارتباطی کامپیوتری

معماری شبکه و طراحی

واژگان عمومی

محاسبه، عملکرد، آزمایش

واژگان کلیدی: شبکه های تحویل محتوا، Akamai، عملکرد CDN

1. مقدمه

پس از دوره ای زمانی که از ادغام صنعت گذشت، علاقه به شبکه های تحویل محتوا احیا شده است. ده ها شرکت CDN جدید ظهور کرده و تبدیل به بخش مهمی از زیر ساخت WEB شده اند: تنها Akamai ادعا می کند که 20 درصد از ترافیک کل WEB را ارائه می دهد. [3]

دو نکته اصلی فروش یک سرویس CDN موارد زیر هستند:

1- بر اساس ظرفیت تقاضا، سرویس در اختیار تولید کنندگان محتوا قرار دهند. 2- از نقطه نظر کاربر، عملکرد دسترسی به محتوا را بهبود بخشند، زیرا محتوا را از مکان های مختلف تحویل می دهند. این مقاله بر روی نظریه دوم تمرکز دارد و عملکرد پلتفرم Akamai که در حال حاضر استفاده می شود را در نظر می گیرد و این سوال را پاسخ می دهد که چرا توسعه زیاد یک پلتفرم CDN بایستی وجود داشته باشد تا مزایای در دسترس بودن را برای کاربران ایجاد نماید.

مطالعه ما در مورد توزیع پلتفرم با بحث بر روی 2 روش اصلی در طراحی CDN که طی سالها با هم ادغام شده اند به هدف رسیده است. یک رویکرد که به طور نمونه توسط Akamai و Digital Island ارائه شده (در حال حاضر متعلق به سطح 3 ارتباطات)، ایجاد نقاط با ظرفیت نسبتا محدود را در نظر می گیرد که در صورت امکان، برای بسیاری از مکان ها در دسترس باشد. به عنوان مثال، پلتفرم Akamai در سال 2007 بیش از 3000 مکان در بیش از 750 شهر در بیش از 70 کشور را ارزیابی کرد که هر مکان کمتر از 10 سرور در اختیار داشت و رد پای آنها بیشتر از پیش رشد داشته است (<http://www.akamai.com/hdwp> را مشاهده کنید). رویکرد دیگر، مراکز داده بزرگ حاوی هزاران سرور ولی در مکان های کمتر را در نظر می گیرد. نمونه هایی از ارائه دهندگان با این رویکرد، Limelight و AT&T هستند. در حال حاضر Limelight 20 مرکز داده را در وب سایت خود لیست کرده است. [13]

عملاً ممکن است دلایل پیچیده ای وجود داشته باشد که در انتخاب این طراحی نقش دارند. از یک طرف Akamai تلاش می کند تا به منظور کاهش ترافیک بالا، سرورهای Cache خود را در برخی ISP ها به صورت رایگان قرار دهد، بنابراین هزینه اجرای پلتفرم خود را کاهش می دهد. از سوی دیگر، کنترل پلتفرم های متصل شده توسط Limelight و AT&T می تواند بهتر باشد و اغلب به جای مراکز داده ای که همراه CDN اجاره داده می شود، در مراکز داده ای که مالک آن هستند توسعه پیدا کنند. در عین حال، تعداد زیادی مکان ها اغلب به عنوان ترجمه مستقیم ذکر شده اند تا در دسترس بودن کلاینت و عملکرد تحویل محتوا بهبود یابد. بنابراین بدون اینکه در ارتباط با شایستگی این دو رویکرد قضاوتی داشته باشیم، بر روی رویکرد تجاری آنها تمرکز کرده و به این سوال پاسخ می دهیم: چه تعداد مکان از بعد اینکه عملکرد کلاینت مفید باشد کافی است؟ یا به صورت متفاوت مطرح می کنیم: چه زمانی با افزایش تعداد مکانها، برخورد کاهش پیدا می کند و بهبود دسترسی کلاینت حاصل می گردد. در حقیقت با توجه به مفاهیم عملکرد تقویت پلتفرم، مطالعه ما سیمای مهمی از دو رویکرد را نشان می دهد، و در نتیجه به بحث در مورد نقاط قوت آنها پرداخته خواهد شد.

توجه کنید که تعداد مکان ها نسبت مستقیم به کل ظرفیت CDN و همچنین توانایی CDN برای ارائه ظرفیت بر اساس تقاضای صورت گرفته از سمت ارائه دهندگان محتوا دارد. با فراهم کردن اتصالات کافی شبکه، منبع تغذیه و سرورها در یک مرکز داده مشخص، می توان ظرفیت CDN را در تعدادی مرکز داده نسبتاً کوچک متراکم نمود. به طور مثال جمله برگرفته شده از وب سایت Limelight را در نظر بگیرید: " هر Network Delivery Center مربوط به Limelight چایگاه هزاران سرور است " [13] که می تواند اشاره به این باشد که Limelight حداقل 20 هزار سرور در کل 20 مرکز داده اش باید در اختیار داشته باشد که حداقل 3 مرتبه بدتر از Akamai است، با وجود اینکه 2 مرتبه از نظر مکانی کوچک تر از Akamai است. از نظر ظرفیت شبکه، Limelight در ماه اوت سال 2009، دارای پهنای باند 2.5 ترابایت بود [12]، با این وجود ما نمی توانیم مقداری مشابه برای Akamai پیدا کنیم، این مقدار بیشتر از مجموع حد نصاب ترافیکی است که آن را تحویل داده است. [2]

فرضیه ای که مطرح می شود این است که شبکه های تحویل محتوا مدت ها پیش این تحقیقات را انجام داده اند. این امکان وجود دارد، ما هرگز نمی دانیم. به هر حال، تحقیقات اختصاصی و بررسی عمومی در اختیار عموم قرار نداشته و غالباً در راستای منافع صورت می گیرند. این مقاله تلاش دارد با بررسی عملکرد Akamai به این پرسش پاسخ دهد. به این دلیل Akamai را انتخاب کردیم که هم از لحاظ سهم بازار و هم از لحاظ وسعت، ارائه دهنده برتر CDN است. رویکرد عمومی ما این است که بررسی کنیم هنگامی که از چند مرکز داده عملیات انجام می گیرد، عملکرد تحویل محتوای پرسرعت Akamai به چه دلیلی می تواند آزاد دهنده باشد. توجه کنید که مفهوم عملکرد تقویت مرکز داده به وسیله CDN مشابه، اهمیت به سزایی دارد، زیرا احتمال این را برطرف می کند که نسخه های نامربوط در CDN های مختلف بتوانند بر روی نتایج تاثیر بگذارند. خلاصه ای از نتایج مقدماتی آشکار شده اند [24]. این مقاله، بررسی کامل را ارائه می دهد.

فعالیت ما، دیدگاه ها را در ارتباط با موارد زیر که مربوط شبکه های تحویل محتوا سهیم می کند:

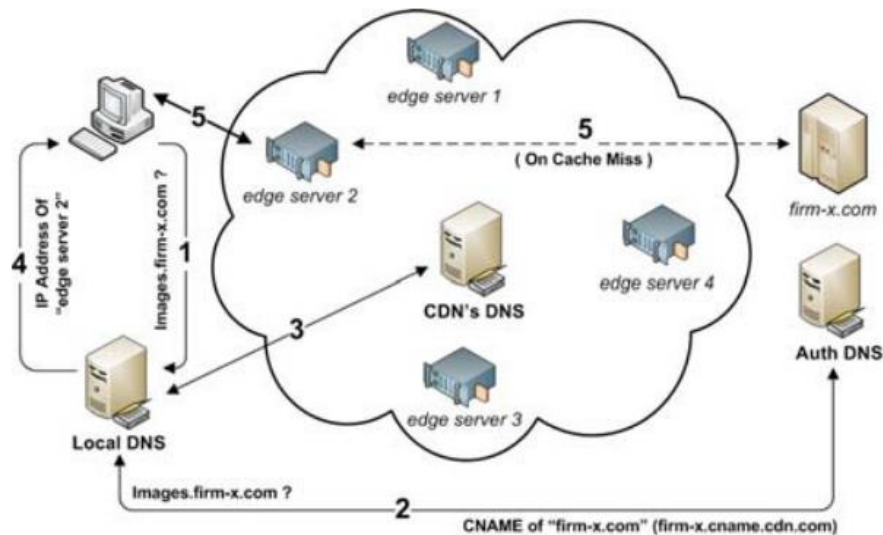
- بهبود عملکرد CDN: CDN ها ظرفیت را بر اساس تقاضا ارائه می دهند، و از این رو برای اشتراک ارائه دهندگان محتوا از اضافه بار پیشگیری می کنند. ولی آیا CDN ها تجربه کاربر در خلال بارگیری معمولی را بهبود می بخشند؟ کریشنامورتی و همکارانش عملکرد CDN های مختلف را مقایسه کرده اند [11] ولی از نظر ما مطالعه ای که انجام داده ایم اولین مطالعه ای است که ارزیابی مستقیم و مستقلی از بهبود عملکرد دانلود های Akamai ارائه می دهد. علاوه بر این، ما هر دو مورد بهبود عملکرد پیشنهاد شده توسط Akamai به ارائه دهندگان محتوا و همچنین کیفیت سرور Akamai انتخاب شده در هنگامی که یک cache را برای دانلود انتخاب می کند را در نظر می گیریم.
- میزان تولید پلتفرم. سوالی را در نظر می گیریم، تا چه اندازه مقادیر زیادی از نقاط حضور عملکرد CDN را بهبود می بخشد. در نظر گرفتن تنها یک جنبه از این موضوعات (عملکرد)، به ما کمک می کند و نیاز نیست در مورد بهتر بودن رویکردهای بیشتر توزیع شده در مقابل روش های تلفیقی برای طراحی CDN از دیدگاه مشتری بحثی را صورت بدهیم.

علاوه بر مشاهدات عملکرد فوق، امیدواریم که روشی که برای بدست آوردن آنها به آن می پردازیم برای پژوهش عملیاتی دیگران مفید باشد.

2. فعالیت مرتبط

امروزه تعدادی از CDN ها سرویس های پر سرعت ارائه می دهند [19,1,13,8,17]. مطالعه در این مورد می تواند مفید باشد زیرا در مورد اولویت های سرمایه گذاری زیر ساخت CDN ها و همچنین برای کمک به مشتریان در انتخاب یک ارائه دهنده CDN، تصمیم گیری خواهد شد.

مقالات گوناگونی عملکرد CDN ها را مورد مطالعه قرار داده اند. کریشنامورتی و همکارانش عملکرد چندین CDNs را در زمان مطالعه مقایسه کردند [11]. این مقاله اولین نشانه از این است که ممکن است آثار وجود CDN مستقیماً به عنوان عملکرد آن تفسیر نگردد. مطالعه ما یک مطالعه مستقیم در ارتباط با عملکرد مفاهیم تقویت مرکز داده یا همان CDN است.



تصویر 1: شبکه تحویل محتوا

Canali و همکارانش [5] عملکردی که توسط کاربر از تحویل CDN درک می شود را با در نظر گرفتن موارد مرتبط با فواصل مکانی و با نظارت بر عملکرد تحویل CDN به مدت 2 سال و از سه مکان مورد مطالعه قرار دادند. Su و

Kusmanovic تحویل محتوای Akamai را از دیدگاه امنیتی تجزیه و تحلیل کردند [22]. ما تحویل منظم ترافیک وب توسط Akamai را در نظر گرفته که به روشی متفاوت از جریان، مهندسی شده است. Biliris و همکارانش عملکرد مفاهیم بالا بردن سرعت از طریق ارائه محتواهای مشابه از طریق CDN های متفاوت را برای کاربران مختلف بررسی نموده اند [4]. این بررسی نشان داده است که هیچ نوع CDN به تنهایی پوشش کافی برای تمامی کاربران اینترنت را ارائه نمی دهد. در کنار نتایج حال حاضر ما که Akamai با تقویت مراکز داده اش عملکرد نزولی اش را تحمل نمی کند، این نتیجه حاکی این است که CDN ها ممکن است قادر باشند تا دسترسی مکانی شان را سهولت بخشند.

Su و همکارانش [21]، امکان استفاده از انتخاب سرور Akamai برای یافتن مسیرهای شلوغ باکیفیت بر روی اینترنت را بررسی کرده اند و در این پروسه جوانب مختلفی از عملکردهای Akamai را مورد بررسی قرار داده اند. Poese و همکارانش [16] مشاهده کرده اند که انتخاب سرور CDN می تواند از طریق کمک از ISP مربوط به کلاینت بهبود یابد. این مطالعات نیازمند کشف سرور های Akamai دارد، تجربه ما در این مورد، برای تحقیقات آتی مفید خواهد بود.

3. سابقه: شبکه های تحویل محتوا

یک CDN، زیر بنایی است برای تحویل موثر محتوای مبتنی بر وب جهت ارائه به کاربران اینترنت. یک اپراتور CDN (نظیر Akamai) تعداد زیادی از edge server ها را از طریق اینترنت گسترش می دهد. ارائه دهنده محتوا hostname های مشخص را به DNS مربوط به CDN تحویل می دهد به صورتیکه وقتی یک کاربر بر روی یک URL تحویل داده شده کلیک می کند، محتوا به جای اینکه از سایت ارائه دهنده محتوا دانلود شود، از طریق نزدیک ترین edge server دانلود می شود.

مکانیزم پایه در شکل 1 توضیح داده شده است. جهت انجام عملیات تحویل host name تحت عنوان images.firm-x.com ، سرور DNS مربوط به firm-x تقاضاهای DNS را با برگرداندن CNAME

(canonical name) برای hostname های تحویل داده شده به CDN بازمی گرداند، شبیه پاسخ به کاربر، به طور مثال `image.firm-x.com.CDNname.net` (مرحله 2 در شکل). اکنون کاربر با یک query که در سیستم DNS برای دامنه `CDN-name.net` وارد می شود بایستی canonical name را حل کند که با CDN عمل می کند (مرحله 3). عمل آخر یک edge server را انتخاب می کند (به طور نمونه نزدیک به کلاینت، تا زمانیکه بار سرور و شرایط شبکه اجازه دهد) و به query با آدرس IP سرور انتخاب شده پاسخ می دهد، دانلود محتوا از سرور انتخاب شده تصویب می گردد (مرحله 4). Edge sever به صورت یک cache عمل می کند- محتوا را به صورت نمونه از محل ذخیره محلی تهیه می کند (اگر محتوا را از درخواست های قبلی داشته باشد) یا محتوا را از سایت مبدا تهیه می کند و آن را به کلاینت ارسال می کند و برای نیاز آتی آن را ذخیره می کند.



تصویر 2: یک نقشه Google نمونه از نقاط مقیاس DipZoom

در واقع، Akamai یک سیستم DNS در مرحله ای را به خدمت می گیرد (به دلیل پیچیدگی در شکل نشان داده نشده است): سرورهای سطح بالای مرکزی پاسخ های نوع NS را به query ها بر می گرداند، آنها را به سرورهای DNS سطح پایین نزدیک شان راهنمایی می کنند و در آخر عملیات Hostname به IP را انجام می دهند. ساختار پلتفرم حقیقی، آدرس دهی های پیچیده دیگری را انجام می دهد. مثلاً، Sherman و همکاران سیستم مدیریت پیکربندی خبره Akamai را توصیف نموده اند [20].

4. روش شناسی

این بخش روش های تجربی اصلی که جهت مطالعه مورد استفاده قرار می دهیم را توصیف می کند.

4.1 کشف edge server

بیشتر تحقیقات CDN نیاز به کشف edge server های CDN دارند [11,5,21]. تکنیک پایه برای کشف edge server به خوبی استقرار یافته و به سادگی مستلزم شناسایی و حل یک hostname تحویل داده شده است. مثلاً دستور nslookup روی images.amazon.com یا روی canonical name به صورت a1794.1.akamai.net معمولاً حداقل دو آدرس IP از edge server ها بر می گرداند.

در هر صورت چالش هنگامی روی می دهد که شخصی بخواهد بر روی یک مجموعه بزرگ از edge server ها کار کند، این مورد نیاز دارد به اینکه hostname از مکان های جغرافیایی متفاوت استخراج شود. برای جلوگیری از پیچیدگی به دسترسی و ارتباط به host های دیگر، از DipZoom استفاده کرده ایم که یک پلتفرم محاسبه اینترنت نقطه به نقطه است [7,26]. DipZoom تعداد زیادی نقاط محاسباتی در اطراف جهان دارد و اجازه می دهد که تجربیات جهانی به صورت برنامه های جاوای محلی اجرا شوند بدون اینکه نیاز باشد تا به صورت آشکار با نقاط محاسباتی جداگانه تعامل داشته باشند. هنگامی که نقاط محاسباتی DipZoom در دسترس، از نظر زمانی متفاوت باشند، معمولاً بیش از MP 400 فعال در دسترس، غالباً در گره های PlanetLab وجود دارند و همچنین روی بعضی از هاست های residential و academic. به عنوان نمونه ای از پوشش جغرافیایی، شکل شماره 2 یک تصویر گرفته شده از Google map را به صورتی نشان می دهد که نقاط DipZoom فعال از هم جدا شده اند [7].

طی بررسی که انجام دادیم، canonical name های ارسال شده به Akamai را گردآوری نمودیم به این صورت که آنها را دانلود کرده و منابع صفحات وب را از 95 مشتری Akamai که بر روی وب سایت Akamai لیست شده بودند تست نمودیم. سپس متناوباً (دوباره در هفته) عملیات صورت گرفت توسط DNS را از این نام ها به تعداد زیادی از مکان های شبکه در یک بازه زمانی 13 هفته ای را انجام دادیم. نتیجتاً زیر 12 هزار Akamai edge server

حاصل شد که 10231 سرور در انتهای فرآیند اکتشاف قابل ردیابی بودند. با کلاسه بندی این edge server ها بر اساس شهر و سیستم های مستقل، به طور سنتی تخمین زدیم که حداقل 308 مکان کشف شده اند. عدد حقیقی بزرگتر است به صورتیکه edge sever های کشف شده 864 از 24 پیشوند را به صورت مجزا نشان می دهد. به طور واضح بخش کشف شده از پلتفرم Akamai فقط مجموعه ای از سرورهای Akamai و مکان ها را نشان می دهند به هر حال همانطور که در بخش 6.1 توضیح خواهیم داد، این مورد در راستای نتایج ما نیست.

4.2 لغو انتخاب CDN Edge Server

برای سنجش عملکرد CDN، ارزیابی استاندارد عملکرد از طریق دانلود صفحه موثر است به صورتیکه از طریق ابزار Curl گزارش شده باشد [6]. جهت اندازه گیری عملکرد دانلود از یک edge server مشخص به جای انتخاب سرور Akamai، بایستی مستقیماً با استفاده از IP آدرس خام آن به جای DNS host name از URL به edge server مطلوب متصل شویم. بررسی های ما نشان داده که جهت پردازش درخواست توسط یک Akamai's edge server کافی است که خیلی ساده HTTP host header را که با یک درخواست با استفاده از DNS host name مناسب ارائه شده بوده است را مضمول نماییم.

مثلاً دستور زیر با ارائه host header مورد انتظار از طریق مقدار دستوری "H-" با موفقیت عملیات دانلود را از Akamai edge server انجام می دهد (با آدرس آی پی : 206.132.122.75).

```
curl -H Host:ak.buy.com \  
"http://206.132.122.75/db_assets  
/large_images/093/207502093.jpg"
```

4.3 کنترل Edge Server Caching

بعضی از آزمایش های ما مستلزم این هستند که درخواست های HTTP به جای Edge Server Cache از سرور مبدا انجام شوند. به طور معمول، درخواست یک cache برای داشتن موردی از سرور اصلی می تواند با استفاده از

سربرگ Cache Control مربوط به HTTP انجام پذیرد. به هر حال، همانطوریکه ما به اختصار نشان خواهیم داد، Edge Server های Akamai برای سربرگ Cache-Control در درخواست های کلاینت ارزشی قائل نمی شوند [23]. جهت انجام ماهرانه بدست آوردن محتوا از edge server اصلی، موارد زیر را استخراج می کنیم. از یک طرف، cache های پیشرفته رشته های URL شامل رشته جستجو کلی را به عنوان کلید Cache استفاده می کنیم (بخش اختیاری یک URL بعد از “?”). به خصوص یک درخواست برای foo.jpg?randomstring برای سرور اصلی ارسال خواهد شد، زیرا cache به احتمال زیاد یک شیء را با این URL ذخیره نموده است. از طرف دیگر، سرورهای مبدا رشته های جستجوی غیر منتظره را در URL های متعبر دیگر نادیده می گیرند. بنابراین درخواست فوق تصویر foo.jpg معتبر را از سرور مبدا بازخواهد گرداند.

جهت تأیید این تکنیک یک سری از دانه‌ها از “planetlab1.iii.u-tokyo.ac.jp” مربوط به یک شیء آمازون به صورت زیر:

”http://g-ec2.images-amazon.com/images/G/01/nav2/gamma/n2CoreLibs/n2CoreLibs-utilities-12475.js”

را با استفاده از edge server به آدرس 60.254.185.89 را انجام دادیم. با انتخاب کلاینت و edge server که به هم نزدیک هستند ولی هر دوی آنها از سرور مبدا فاصله دارند (هر دوی کلاینت و سرور مبدا در ژاپن هستند در حالیکه مبدا شاید در امریکا باشد) ما امیدواریم بتوانیم دانه‌ها از cache را از مبدا توسط تفاوت عملکرد تشخیص دهیم.

	Target & Parameter	Throughput
1	/foo.js?rand1	43 KB/s
2	/foo.js?rand1	7750 KB/s
3	/foo.js?rand1	5300 KB/s
4	/foo.js?rand2	61 KB/s
5	/foo.js?rand2	8000 KB/s
6	/foo.js?rand2	4850 KB/s
7	no-cache, /foo.js?rand2	5070 KB/s
8	no-cache, /foo.js?rand2	6780 KB/s
9	no-cache, /foo.js?rand3	56 KB/s

جدول 1: در آغاز در مقابل تکرار عملکرد از یک شیء ، یک رشته جستجوی تصادفی افزوده شده.

جدول 1 توان عملیاتی سری های دانلود را نمایش می دهد (برای راحتی آدرس URL شیء بلند با "foo.jpg" جایگزین شده است). دانلود 1 ، یک دانلود non-cache است و عملکرد ضعیف را نشان می دهد. دانلود های 2 و 3، دانلود هایی با رشته تصادفی مشابه هستند، آنها عملکرد قابل ملاحظه بالاتری را نشان می دهند که نشان دهنده این واقعیت است که این درخواست ها از سمت cache اجرا می شوند که این آدرس را به عنوان نتیجه ای از اولین دانلود ذخیره می نماید. دانلود های 4 الی 6 این رفتار را با یک حلقه تصادفی متفاوت تأیید می کنند. با این حال دانلود های 7 و 8 با استفاده از URL های دیده شده از قبل با یک سربرگ no-cache ، عملکرد قوی مشابهی را تولید می کنند که نشان می دهد آنها با وجود اینکه به صورت no-cache تعریف شده اند، از سمت cache پردازش شده اند. با این حال، دانلود شماره 9، که به صورت no-cache تعریف شده است و با یک رشته تصادفی جدید، عملکردی قابل مقایسه با دانلود 1 و 4 نمایش می دهد و نشان دهنده دانلودهای no-cache است. این نشان می دهد که دستور no-cache از سربرگ HTTP-cache control، در درخواست های کلاینت توسط Akamai بر روی caching تاثیر نمی گذارد.

به طور خلاصه، می توان به صراحت رفتار caching مربوط به edge server های Akamai را با اضافه کردن رشته های جستجو به URL های تصویر کنترل نمود. اولین چیزی که از یک رشته پدیدار می شود باعث می شود تا دانلود از سرور اصلی شروع شود در حالیکه دانلود های بعدی از رشته مشابه، از Edge server مشابه از edge server cache انجام خواهد شد. یک سری از بازیابی ها از محتوای مشابه با رشته های تصادفی باعث می شود تا edge server در هر لحظه حاوی کپی اصلی شود.

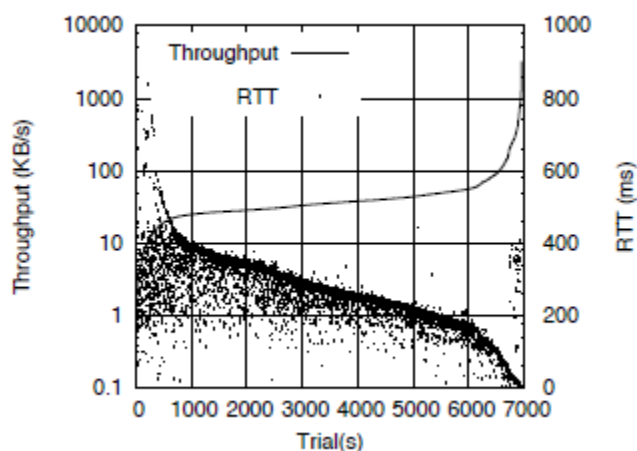
4.4 ارزیابی Caching Bias سمت کلاینت

همانطور که ذیلا مشاهده می کنید بیشترین تجربیات ما شامل مقایسه عملکرد دانلود از مکان های مختلف Akamai است. مساله اساسی این است که bias موجود معرفی شده توسط caching به صورت بالقوه توسط IPS هایی که از طریق نقاط محاسباتی ما به اینترنت متصل می شوند مورد استفاده قرار گیرد. در واقع دانلود cache شده ، بدون در

نظر گرفتن مکان Akamai که ما آشکارا درخواست HTTP را هدایت می کنیم، تمایل دارد که از Cache انجام شده باشد.

برای تخمین میزان استفاده Transparent cache توسط ISP هایی که توسط نمایندگان DipZoom استفاده می شوند، به این نکته توجه می کنیم که چگونه عملکرد دانلود HTTP از یک Edge Server مربوط به Akamai بستگی به تاخیر رفت و برگشت ping مربوط به خودش دارد. اگر Transparent cache به طور گسترده ای استفاده شود، ما هیچ گونه وابستگی نخواهیم دید.

برای مطالعه این ارتباطات، مجموعه ای از چهار دانلود Curl را ، با پنج Ping و یک trace root از هر نقطه اندازه گیری به edge server های مختلف، انجام می دهیم. برای عملکرد دانلود HTTP، میانگین توان های عملیاتی دانلود گزارش شده از حداقل آخرین سه مورد از curl ها (اولین curl تضمین می دهد که شیء در edge server cache است)، برای ping RTT، را پیدا می کنیم، همچنین میانگین RTT ها از پنج عدد ping. ما یک شیء قابل cache به صورت outsource شده Amazon 437,688B برای دانلود های curl استفاده می کنیم.



شکل 3: ارتباط بین توان مصرفی دانلود و RTT

تعداد 385 نقطه محاسباتی و 20 عدد edge server توزیع شده گسترده شده را در این آزمایش استفاده کرده ایم. برای اطمینان از اینکه edge server های مقصد به صورت گسترده ای در اطراف دنیا توزیع شده اند، تمامی edge

server های پوشش داده شده را در 20 دسته بر اساس مسافت تخمین زده شده بین سرورها و مراکز انتخاب شده از دسته ها به عنوان اهداف مان، دسته بندی می کنیم (بخش 6 را ملاحظه کنید).

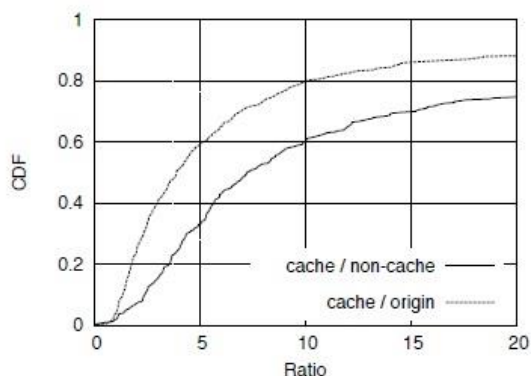
شکل 3 مقادیر ping RTT و توان مصرفی دانلود curl را در هر آزمایش که به ترتیب افزایش توان مرتب شده است را نقطه گذاری می کند. (مجموع تعداد آزمایشات کمتر از $20 * 385$ است زیرا تمامی اندازه گیری ها موفق نبودند). این تصویر، وابستگی آشکار عملکرد دانلود بر روی مسافت ping را نمایش می دهد. از لحاظ کمی، دو مقیاس نشان دهنده ضریب همبستگی Spearman -0.8074، بیانگر یک وابستگی قوی هستند و بنابراین استفاده قابل توجهی از ISP-caching سمت کلاینت وجود ندارد.

در ادامه یک روش مستقیم جهت کشف یک transparent cache را توضیح داده ایم. ما سرور وب خودمان را با یک شیء قابل cache گسترش می دهیم و دوبار آن را از هر نقطه محاسباتی درخواست می کنیم. سپس گزارش دسترسی به سرور را امتحان می کنیم. اگر دانلودها از یک MP داده شده از طریق ISP cache فیلتر شده باشند، گزارش شامل فقط یک درخواست از این MP است (دیگری توسط ISP cache خاتمه یافته است)، در غیر این صورت گزارش شامل هر دو درخواست خواهد بود. در اینجا 527 نقطه محاسباتی را تست کرده ایم (هر تعداد که می توانستیم جمع آوری کنیم) و فقط 3 مورد را با یک درخواست تکی یافتیم. بنابراین اکثریت MP ها (99.43٪) Transparent cache بین خود و وب سرورها ندارند. وقتی این آزمایش چند ماه بعد از پایان مطالعه ما صورت گرفت، دو تجربه ای که در اینجا توضیح داده شده اند، سطح بالایی از اطمینان را به ما دادند که caching سمت کلاینت از یافته های ما منحرف نمی شوند.

4.5 محاسبه عملکرد Edge Server

در بخش 5 و 6.1 ما از عملکرد دانلود به عنوان معیار خوبی از یک edge server استفاده می کنیم و آن را به عنوان توان عملیاتی موثر بیان می کنیم که اندازه شیء است که توسط زمان دانلود تقسیم بندی شده است. با توجه به اینکه عملکرد دانلودها از اشیاء با اندازه های متفاوت غالباً تحت تاثیر حالات مختلف شبکه هستند (پهنای باند برای اشیاء

بزرگ و RTT برای اشیاء کوچک)، یافته‌های اصلی مان را بر اساس اشیاء در محدوده‌ای وسیع بررسی می‌کنیم (از 10K تا 400K). علاوه بر این، بخش 6.3 زمان Round-trip از کلاینت را به عنوان معیار خوب از یک سرور استفاده می‌کند.



شکل 4: مزایای عملکرد دریافت Akamai

5. عملکرد AKAMAI CDN

این بخش به این نکته می‌پردازد که چگونه پلتفرم فعلی Akamai عملیاتش را برای سرعت بخشیدن به دریافت محتوا به خوبی انجام می‌دهد. ابتدا، مطالعه می‌کنیم که چگونه بر عملکرد دانلود کاربران نهایی تاثیر می‌گذارد و سپس کیفیت انتخاب سرور آن را بررسی می‌کنیم.

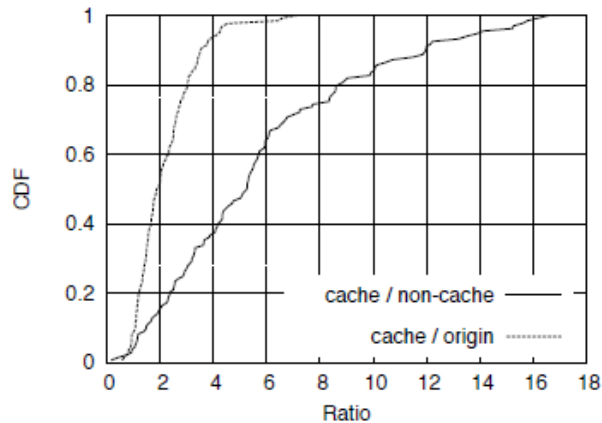
5.1 آیا یک CDN عملکرد را بالا می‌برد؟

به طور کلی یک شبکه تحویل محتوا می‌تواند عملکرد وب را به 2 روش بهبود بخشد: با محافظت از جلوگیری از اضافه بار برای سایت‌های وب و با تحویل محتوا به کاربران از مکان‌های نزدیک. در این مطالعه، تمرکز ما بر روی جنبه دوم است. برای پاسخ دقیق به این سوال، موردی که به آن نیاز خواهیم داشت مقایسه عملکرد دانلود میان یک CDN است با دانلود مستقیم از شیء مشابه از سرور اصلی. متأسفانه متوجه شدیم که سرورهای اصلی میزبان محتوای تحویل

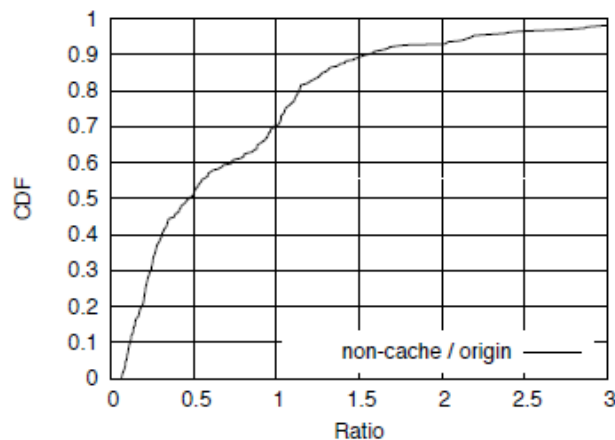
داده شده Akamai معمولاً از کاربران اینترنتی مخفی هستند: تلاش های ما برای درخواست این محتوا از سایت های اصلی که میزبان صفحات حاوی محتوا هستند موفق نبود.

بنابراین، ما متوسل می شویم به تخمین از عملکرد دانلود از سرورهای اصلی. اولاً هنگامی که از Caching در سرورهای edge ممانعت می کنیم، آن را با زمان دانلود از محتوای تحویل داده شده Akamai تخمین می زنیم. این کار عملاً عملکرد از دست دادن cache در Akamai را اندازه گیری می کند و از این رو جریمه از دست دادن را به مقادیر صحیح اضافه می کند. ولی با فرض اینکه زیرساخت Akamai به نحو احسن بهینه شده است، این مورد می تواند عملکرد را از دانلود مستقیم نشان دهد. ثانیاً عملکرد سرور اصلی را با دانلود عملکرد از یک شیء متفاوت، با سایز مشابه تخمین می زنیم که برای تحویل CDN از وب سایت مشابه برون سپاری نشده است. حتی اگر شیء برون سپاری نشده از سرور مشابه سرویس شده باشد تا موردی که توسط Akamai از دست رفته باشد، عملکرد این دانلود عملکردی را نشان می دهد که سایت وب می تواند بدون دستیابی به یک CDN خارجی آن را به دست آورد.

برای این تجربیات 377 نقطه محاسباتی را استفاده نموده ایم. برای محتوای تحویل داده شده CDN، از یک شیء آمازون برون سپاری شده با سایز 50K استفاده کرده ایم و روش موجود در بخش 4.3 را نیز برای کنترل caching آن مورد استفاده قرار داده ایم. برای دانلود از سایت اصلی، یک صفحه استاتیک برون سپاری نشده از 55 کیلوبایت را در وب سایت آمازون پیدا کردیم. (ما آن را استاتیک در نظر می گیریم با این فرض که داده های مختلف از نقاط بالاتر منتج به محتوای مشابهی شده اند). بدیهی است که نتایج در این قسمت اکیدا فقط برای آمازون قابل قبول است، چون مشتری دیگر ممکن است سایت اصلی با عملکردی متفاوت داشته باشد. به هر حال یک نقطه داده مشخص با یک ارائه دهنده محتوای بزرگ و احتمالاً با طراحی خوب را پیشنهاد می دهند.



شکل 5: مزایای عملکرد کلاینت محلی از دریافت Akamai



شکل 6: مقایسه دانلود بدون cache از طریق Akamai و دانلود از سرور اصلی.

تصویر 4 توابع توزیعی انباشته را از ضرایب عملیاتی cache to origin و cache to noncache را در تمامی محاسبات جمع آوری شده ترسیم می کند. در هر دو مورد، مقادیر بیش از 1 این معنی را می دهد که CDN عملکرد را روی دریافت مستقیم بهبود بخشیده است (تحت برآورد تخمین مستقیم عملکرد تحویل) و مقدار زیر 1 معنی مخالف را می دهد. تصویر نشان می دهد که یک CDN وعده یک بهبود عملکرد قابل توجه، بر اساس هر دو تخمین ها از عملکرد تحویل مستقیم را می دهد. تحویل CDN از هر دو مورد تحویل های nocache و اصلی به ترتیب 98٪ و 96٪ از موارد بهتر عمل می کند. علاوه بر این در 67٪ و 41٪ از موارد، دریافت CDN حداقل 5 مرتبه سریع تر از به ترتیب دریافت no-cache و اصلی است. یک دلیل ممکن برای چنین بهبود چشمگیری می تواند وجود داشته باشد و آن اینکه بیشتر نقاط اندازه گیری ما به خوبی به اینترنت متصل شده اند و تنگنای مسافت را تجربه نخواهند کرد.

بنابراین برای مشاهده مزایای کلاینت های موجود، بر اساس شکل 5، نتایج مشابه را مجدداً ایجاد می کنیم ولی تنها برای نقاط محاسباتی با بیشترین پهنای باند دانلود زیر 1.5Mbps برای اتصال حال حاضر. برای این نقاط محاسباتی، مزایای عملکرد به طور قابل ملاحظه ای از دست می رود. اکنون تحویل CDN در 2.3٪ از موارد، حداقل 5 مرتبه سریع تر از تحویل اصلی است. در هر صورت تحویل CDN کماکان عملکرد دانلود کلاینت های موجود را بهبود می بخشد، تحویل CDN در 95.5٪ از موارد از no-cache و در 91٪ از موارد از تحویل اصلی پیشی می گیرد.

سرانجام، فضای بین دو منحنی در هر دو شکل نشان می دهد که cache جهانی Akamai هزینه بردار است، به این معنی که سربار سرورهای edge برای یک درخواست، هنگامی که مورد درخواست شده از سرور اصلی واکنشی شده باشد، اضافه می گردند. کیفیت این هزینه را در شکل 6 نشان می دهیم به صورتیکه CDF مربوط به nocache to origin نسبت توان عملیاتی را ترسیم می کند. و نشان می دهد که در 70٪ از موارد، دانلودهای اصلی از دانلود های no-cache سریعتر هستند و در 50٪ از موارد حداقل 2 مرتبه سریع تر هستند. این نشان می دهد که هزینه Akamai می تواند قابل توجه باشد.

محدودیت بالقوه آزمایش در این قسمت در این است که عملکرد یک مورد وب تکی در نظر گرفته می شود و دانلود های صفحه کلی را در نظر نمی گیرد. معمولاً صفحات شامل چندین مورد جاسازی شده هستند که ممکن است تاثیر مستقیم و دانلودهای شتاب یافته شده CDN را به صورت متفاوت داشته باشند. در واقع بسته به نصب سرویس های CDN، مرورگر می تواند به طور مستقیم شیء HTML را مستقیماً از سایت اصلی و اشیاء جاسازی شده را از سرور edge مربوط به CDN دانلود کند. بنابراین، دریافت CDN می تواند مستلزم DNS resolution اضافی باشد (برای تبدیل نام های میزبان مربوط به URL های جاسازی شده) و یک برقراری اتصال TCP اضافه (برای ایجاد اتصال برای هر روی سرورهای edge و اصلی). به هر حال این مورد نباید به طور موثری نتیجه ما را تحت تاثیر قرار دهد، زیرا این تاثیرات، بر روی اشیاء جاسازی شده روی یک صفحه و معمولاً روی صفحات مختلف در یک جلسه از بین رفته اند (اگر کاربر به این صفحات در کوتاه مدت دسترسی پیدا کند به صورتی که DNS که cache شده است پاسخ دهد و اتصالات مداوم بین دسترسی های صفحه به صورت صحیح باقی بمانند). Akamai به کلاینت ها اجازه می دهد تا

cache DNS به مدت 20 ثانیه پاسخ دهد (گرچه بسیاری آنها را به مدت بیشتری استفاده می کنند [15]). ما همچنین حدود 1000 edge server مربوط به Akamai را برای پشتیبانی اتصال پایدار بررسی کرده ایم و یافته ایم که آنها برای مدت زمان طولانی غیر عادی 500 ثانیه یک درخواست را حفظ می کند (در مقایسه با پیش فرض Apache از 15 ثانیه). این مورد فرصت فراوان برای استفاده مجدد از اتصال را فراهم می آورد.

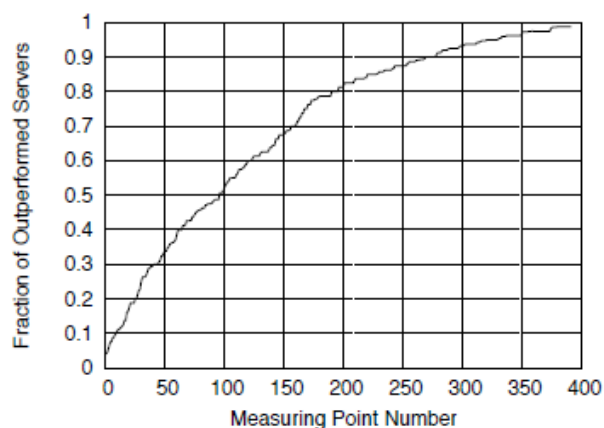
5.2 به چه دلیل انتخاب یک سرور Akamai مناسب است؟

هنگامی که یک کلاینت شروع به دانلود یک محتوای cache شده به Akamai می کند، بهترین edge server برای این کلاینت و محتوای خاص به صورت پویا انتخاب شده اند. این بخش به بررسی این نکته می پردازد که چگونه الگوریتم انتخاب Akamai از دیدگاه عملکرد کلاینت مناسب است.

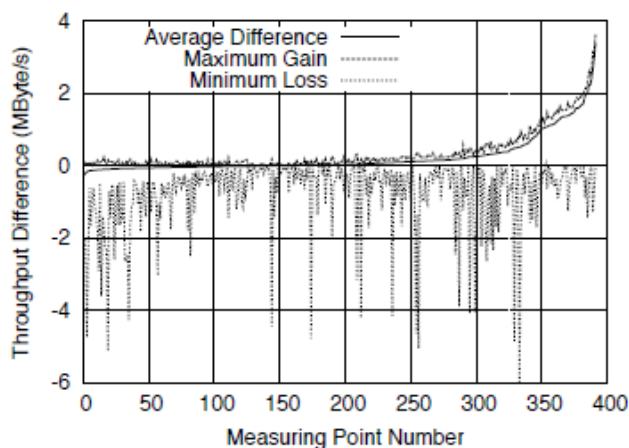
برای پاسخ به این سوال، ما یک مشتری را انتخاب کردیم، Amazon، و توان عملیاتی دانلود یک شیء دریافت شده Akamai را از 50 کیلو بایت از نقاط محاسباتی، اندازه گیری کردیم. از بیش از 10231 سرور Edge که پیدا کردیم، 430 سرور را مختص Amazon کشف کردیم. در هر 391 نقطه محاسباتی، 80 سرور edge به صورت تصادفی از این 430 سرور انتخاب شده اند و شیء از سرور Edge انتخاب شده Akamai دانلود شده و از هر 80 سرور Edge جایگزین، با استفاده از روش بخش 4.2 برای دانلود از یک edge ویژه. به دلیل اینکه تمامی آزمایش از یک نقطه محاسباتی داده شده، زمان قابل ملاحظه ای را مصرف می کند، ما همیشه یک زوج محاسباتی را یکی بعد از دیگری، یکی را برای سرور انتخابی Akamai و یکی برای سرور edge جایگزین داده شده انجام می دهیم. توجه داشته باشید که سرور انتخابی Akamai ممکن است از یک جفت به بعد تغییر نماید. نهایتاً تکنیک های بخش 4.3 را برای تضمین اینکه سرور edge شیء را از cache خودش در هر موردی دریافت می کند، استفاده می کنیم.

در تصویر 7، برای هر نقطه محاسباتی، درصد زمانی که سرور انتخاب شده Akamai بهتر از سرور جایگزین در افزایش ترتیب این استاندارد، عملیات را انجام داده است را مشاهده می کنیم. به عنوان مثال، در نقطه محاسباتی شماره 200، می بینیم که تقریباً 80٪ از 80 سرور edge تصادفی، توسط سرور انتخاب شده Akamai عملکرد بهتری دارند.

انتخاب سرور برای 199 نقطه محاسباتی کمتر شمارش شده موفقیت کمتری داشته و برای 191 نقطه محاسباتی که بالاتر شمرده شده اند موفقیت بیشتری داشته است. به طور کلی این آزمایش مطالعه محدود اولیه را تأیید می کند [9] که CDN ها به ندرت بهترین سرور edge را انتخاب می کنند ولی با موفقیت از بدترین موارد جلوگیری می کنند. در واقع در حدود 75٪ از MP ها ، سرور انتخاب شده Akamai، نیمی از جایگزین ها را بهتر انتخاب می کند.



شکل 7: تعدادی از سرور ها توسط سرور انتخاب کننده Akamai انجام شده اند



شکل 8: تفاوت توان عملیاتی دانلود بین Akamai انتخاب شده و یک سرور Edge جایگزین

شکل 7 نشان می دهد که چگونه به دفعات سرور های انتخاب شده Akamai بهتر یا بدتر از جایگزین ها عمل می کنند ولی نشان نمی دهد که چه اندازه. برای تهیه این اطلاعات در شکل 8 میانگین تفاوت را به صورت حداکثر بهره و حداکثر از دست دادن توان عملیاتی سرور انتخاب شده Akamai در برابر سرورهای جایگزین ترسیم نموده ایم.

میانگین تفاوت بر روی 80 سرور جایگزین دریافت شده است، مقادیر منفی نشان می دهند که سرورهای جایگزین به طور میانگین بهتر عمل می کنند و مقادیر مثبت مربوط به مزیت سرور انتخاب شده Akamai است. حداکثر بهره و حداکثر ضرر، تفاوت های توان عملیاتی بر روی سرور انتخاب شده Akamai هستند، یعنی آهسته ترین و سریع ترین سرور جایگزین. اگر مثبت باشد، حداکثر بهره، مزیت سرور انتخاب شده Akamai را در برابر بدترین سرور جایگزین نشان می دهد. به طور مشابه، اگر منفی باشد، حداکثر ضرر، نشان دهنده حداکثر جریمه در برابر بهترین سرور جایگزین است.

منحنی تفاوت میانگین نشان می دهد که 147 نقطه محاسباتی وجود دارند که سرور انتخاب شده Akamai ، با حداکثر 255 کیلوبایت بر ثانیه برای نقطه محاسباتی 1، به طور میانگین کارکرد پایین تر دریافت می کند. به هر حال سرورهای انتخاب شده Akamai دارای عملکرد بالاتر در میانگین 244 نقطه محاسباتی بالای 3.38 مگابایت بر ثانیه مزیت به طور میانگین هستند. جالب است که گر چه سرور های انتخاب شده Akamai به طور میانگین دارای عملکرد بالایی هستند، منحنی حداکثر اتلاف نشان می دهد که در موارد معدودی بهترین سرورهای جایگزین به طور مشخصی بالای 6.35 مگابایت بر ثانیه در سرورهای انتخاب شده Akamai بهتر عمل می کنند.

به طور خلاصه، معمولاً Akamai تصمیم انتخاب سرور خوبی را اتخاذ می کند و فضای اساسی برای پیشرفت وجود دارد.

6. عملکرد Akamai CDN تلفیقی

Akamai تلاش می کند تا با قرار دادن سرور edge در تعداد زیادی از مکان های شبکه، محتوا را به کاربران تحویل دهد. سوالی که مطرح می شود این است: اگر تعداد مراکز داده به صورت قابل توجهی کاهش یابد هزینه عملکرد چه اندازه خواهد بود ؟

ابتدا روش مان را برای مرکز داده تلفیق شده بیان می کنیم، سپس مطالعه ای را با استفاده از محاسبات دریافت شده از نقاط محاسباتی DipZoom ارائه می دهیم و سپس با یک مطالعه زنده درگیر با کاربران حقیقی اینترنت نتیجه را

به پایان می‌رسانیم. هر دوی مطالعات نشان می‌دهند که یک تلفیق مناسب بدون یک تاثیر قابل توجه در عملکرد، امکان پذیر است.

6.1 تلفیق مرکز داده

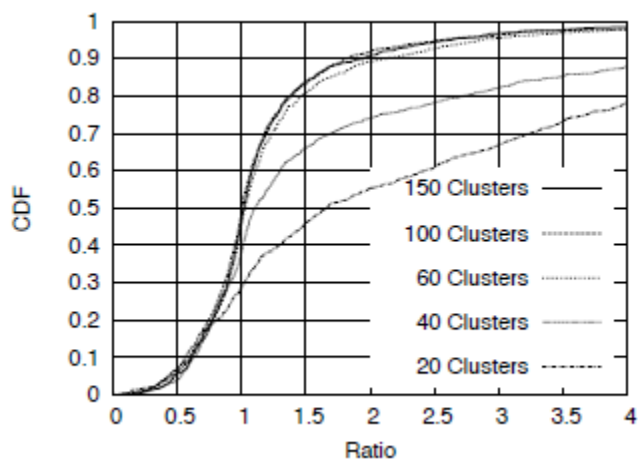
روش ما برای مطالعه یک CDN تلفیقی فرضی، به صورتی است که در ادامه بیان می‌کنیم. ابتدا سرورهای edge را که می‌دانیم در مراکز داده تلفیقی فرضی نزدیک به هم هستند را گروه بندی می‌کنیم و این سرورها را به عنوان big cluster در نظر می‌گیریم. سپس هر big cluster را در محل سرور مرکزی در آن Cluster که نماینده Cluster نامیده می‌شود قرار می‌دهیم. برای این منظور، برای یک کلاینت داده شده، سرور انتخاب شده توسط Akamai یا S_{akam} با نماینده دسته که به S_{akam} تعلق دارد جایگزین می‌گردد. به عبارت دیگر، فرض می‌کنیم که تمامی کلاینت‌ها که به هر سرور موجود در یک دسته بزرگ واقع در یک پلتفرم موجود ارسال شده بودند، از طریق مرکز دسته به هنگام تلفیق سرویس دهی خواهند شد. سپس مفاهیم عملکرد این جابه جایی را با مقایسه عملکرد دانه‌ها از هر دو سرور در نظر می‌گیریم.

قصد ما این است که مجدداً تاکید کنیم، مطالعه ما مفاهیم تلفیق CDN را بر روی نزدیکی کلاینت‌ها به مراکز داده در نظر می‌گیرد: مجموع ظرفیت CDN، هم در پهنای باند شبکه و هم در ظرفیت سرور نسبت عکس دارد با تعداد مراکز داده به اندازه ای که مکان‌ها می‌توانند توسط پردازش‌های بالاتر و ظرفیت اتصال در هر مرکز داده جبران کنند. به دلیل این که مولفه‌های ما بار اضافی روی سرورهای Akamai تحمیل می‌کنند، عملکرد دانه‌های ما مجاورت میان سرور‌ها و کلاینت‌ها را تحت حالت بار نرمال منعکس می‌کنند. در حقیقت هر تفاوت بار سرور در تفاوت‌های زوج سرور منحصر بفرود بایستی در برابر تلفیق پلتفرم فعالیت کند، زیرا Akamai هنگامی که از مراکز دسته، صرفنظر از بار آنها استفاده می‌کنیم از اضافه بار سرورها در انتخاب سرورشان جلوگیری می‌کند.

جهت دسته بندی edge server ها، با تخمین مسافت‌های شبکه جفتی بین تمامی سرورها با استفاده از یک روش سه وجهی پویای پیشنهاد شده شروع می‌کنیم [25] و سپس گروه بندی سرورهای نزدیک به تعدادی از دسته‌های

از پیش تعریف شده را با استفاده از اضافه نمودن یک الگوریتم دسته بندی سلسله مراتبی جهت رسیدن به ماتریس فاصله انجام می دهیم. (ما از این دسته بندی سلسله مراتبی با روش پیوند کامل استفاده می کنیم که با رویه cut-the-tree پیگیری می شود و هر دو به وسیله نرم افزار R [18] برای این محاسبات ارائه شده اند). می توانیم به همان اندازه از تکنیک های دیگر نظیر Network-Aware-Clustering استفاده کنیم [10]. به هر حال هدف نشان دادن این است که می توان تعداد زیادی از سرورها را در مکانهای کمتر تلفیق کنیم و اینکه تا زمانیکه به عملکرد قابل مقایسه پلتفرم تلفیقی پی نبرده ایم، به چه دلیلی انتخاب سرورها برای تلفیق بی اهمیت است. به عبارت دیگر دسته بندی ناقص، یافته های ما را محافظه کارانه می سازد: دسته بندی بهتر می تواند منجر شود به پلتفرم تلفیق شده با ساختار بهتر که حتی مراکز داده کمتری دارد.

به همین ترتیب، نتیجه یک کشف ناقص از پلتفرم Akamai این است که مکان های بالقوه کمتری برای مراکز داده تلفیقی داریم (در واقع، روش های ما از بین edge server های کشف شده فقط به عنوان مکان های تلفیقی انتخاب خود را انجام می دهند). این باعث می شود تا یافته های ما که امکان تلفیق را واضح می سازند، محافظه کارانه باشند. به عبارت دیگر، کشف ناقص، یافته های ما را در مورد تلفیق مکانهایی که کشف کردیم، از بین نمی برد.



شکل 9: عملکرد یک پلتفرم تلفیقی Akamai با تعداد متفاوتی از مراکز داده

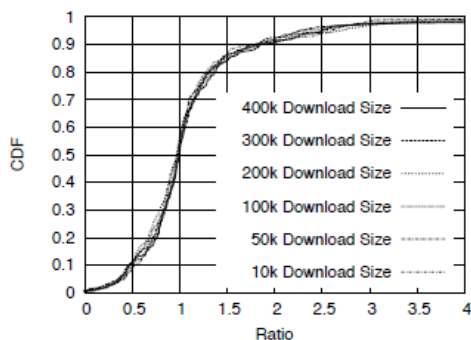
6.2 آزمایش DipZoom

جهت قضاوت عملکرد پلتفرم Akamai تلفیقی فرضی، عملکرد دانلود از پلتفرم تلفیق شده را با پلتفرم موجود با استفاده از نقاط محاسباتی DipZoom مقایسه می کنیم. 412 نقطه محاسباتی در این آزمایش داریم. از هر نقطه محاسباتی شیء مربوط به یک مسافت داده شده را از سرور انتخابی Akamai و از مرکز دسته اش در پلتفرم تلفیقی دانلود نمودیم. مرکز به وسیله یک سرور در دسترس که به طور تصادفی از پنج سرور نزدیک به مرکز دسته انتخاب شده نمایش داده شده است.

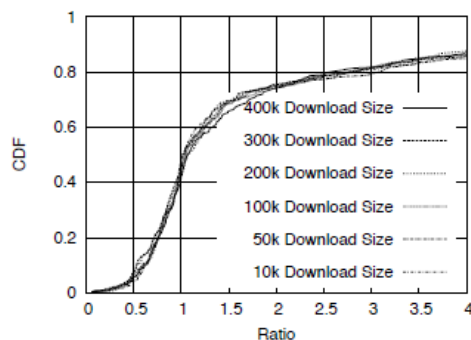
جهت جلوگیری از تغییرات شرایط شبکه، هر جفت از دانلودها را به صورت متوالی انجام می دهیم. علاوه بر این، هر شیء را از دو سرور درخواست می کنیم تا مطمئن شویم سرور، شیء را از cache خود دریافت می کند، بنابراین مانع امکان خرابی نتایج به وسیله cache-miss می شود. با قرار دادن شیء در cache سرور، سه دانلود از هر سرور انجام می دهیم و میانگین را به عنوان عملکرد دانلود در نظر می گیریم. عملکرد پلتفرم های Akamai تلفیقی و پلتفرم موجود را به وسیله ضریب، میان توان عملیاتی دانلود در دو پلتفرم مقایسه می کنیم. ضریب عملکرد بیشتر از 1 به این معنی است که بازده پیکربندی موجود Akamai از نظر دانلود، بهتر از پیکربندی تلفیقی و برعکس است.

سرورهای 10 و 231 را در 100، 150، 60، 40 و 20 مرکز داده گروه بندی کرده و بنابراین اختلاف درجه تلفیق را در نظر می گیریم. برای بررسی اینکه امکان وابستگی نتایج مان به اندازه اشیاء دانلود شده مشخص شود، میزان دانلودها را به صورت دقیق با یافتن یک شیء بزرگ (بیش از 400K) کنترل کردیم و یک سربرگ Range مناسب در درخواست HTTP مشخص نمودیم. تائید می کنیم که سرورهای Akamai درخواست های Range را مورد توجه قرار داده و از آن تبعیت می کنند.

شکل 9 CDF ضرایب توان عملیاتی دانلود پیکربندی های پلتفرم جاری تا پلتفرم تلفیق شده را نمایش می دهد. این شکل، نقاط داده فراهم شده به وسیله دانلود یک شیء آمازون برون سپاری شده از سایز 50K، 100K، 150 K و 10K را از 412 نقطه محاسباتی از اطراف دنیا منعکس می نماید. منحنی ها 5522 ضریب را منعکس می کنند (برخی دانلودها موفق نبودند).



(a) 60 Data Centers



(b) 40 Data Centers

شکل 10: عملکرد یک پلتفرم Akamai تلفیقی با اندازه شیء مقصد متفاوت.

همانطور که در شکل 9 نشان داده شده است، سرورهای edge تلفیقی در 100، 150 و 60 مرکز داده موجب کاهش قابل توجه عملکرد نمی شوند، پلتفرم های موجود و تلفیقی مزیت عملکرد روی همدیگر را با احتمال مساوی نمایش می دهند. فقط وقتی ما به 40 و 20 مرکز داده می رسیم، پلتفرم اصلی شروع به عملکرد بهتر در پیکربندی تلفیقی به میزان 60٪ و 70٪ زمان برای به ترتیب 40 و 20 مرکز داده می کند. علاوه بر این همانطور که در شکل 10 نمایش داده شده، این نتایج شدیداً مستقل از اندازه شیء مقصد هستند.

آزمایش فوق مخلوطی از نقاط اندازه گیری مان را بازمی گرداند که به سمت میزبان های متصل به هم، منحرف شده اند. به این دلیل که CDN ها معمولاً با سایت های مصرف محور با حجم بالا مورد استفاده قرار گرفته اند، که جایی است که کاربران ممکن است اتصالات کم حجمی داشته باشند، عملکرد پیکربندی تلفیقی را به صورت جداگانه برای نقاط اندازه گیری با پهنای باند دانلود متفاوت در نظر می گیریم. مخصوصاً نقاط اندازه گیری را با حداکثر توان عملیاتی دانلود که در جریان آزمایش شکل 9 مشاهده می کنیم، گروه بندی می نماییم.

نتایجی که در شکل 11 نشان داده شده اند، نشان می دهند که پیکربندی Akamai با تعداد زیادی از مراکز داده به کاربران متصل است. برای نقاط محاسباتی با بیش از 6 مگا بیت بر ثانیه پهنای باند، پیکربندی موجود، هنگامی که به 60 مرکز داده می رسند، پیکربندی تلفیقی را بهتر انجام می دهند. به هر حال، با توجه به این مورد، پهنای باند دریافتی از نقاط محاسباتی، مزیت کمتری از پیکربندی موجود داشته، و به دلیل اینکه CDN ها معمولاً توسط سایت

های مشتری محور با حجم بالا استفاده شده اند، سوال مهم این است که چگونه تلفیق ممکن است بر کاربران معمولی حال حاضر تاثیر بگذارد.

بنابراین، در تصویر 12 عملکرد پیکربندی موجود را با پیکربندی تلفیقی برای تمامی نقاط محاسباتی با پهنای باند زیر 1.5 مگا بیت بر ثانیه که برای کاربران DSL پهنای باند دانلود معمولی است، مقایسه می نمایم. تصویر 12 نشان می دهد در صورتیکه سرورها تا 40 مرکز داده تلفیق شوند، این کلاینت ها شاهد عملکرد قابل توجه متفاوتی نمی شوند. به طور کلی نتیجه ای که گرفتیم این است که می توان بخش کشف شده پلتفرم Akamai مان را در 60 مرکز داده و برای کاربران معمولی تا 40 مرکز داده بدون از دست دادن قابل توجه عملکرد تلفیق کنیم. حتی با تخمین محافظه کارانه از 308 مرکز داده اصلی، این مورد نشان دهنده تلفیقی واضح است. به هر حال، همانطوریکه بیان کردیم، تعداد حقیقی مکان های Akamai نشان داده شده در این مطالعه بالاتر است که حتی باعث می شود یافته های ما بیشتر قابل توجه باشند.

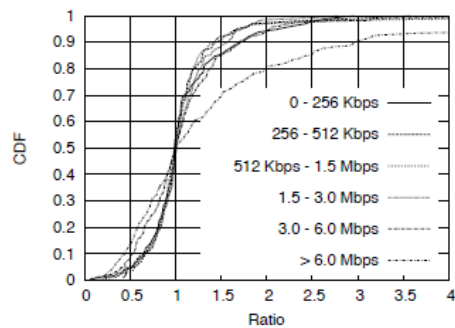
6.3 یک مطالعه زنده

مطالعه بخش قبلی، دانلود های واقعی را اندازه گیری نمود ولی آنها را از نقاط محاسباتی DipZoom انجام داد. در این بخش، محاسبات را به صورت زنده از تعداد زیادی نقاط برتر و از کاربران اینترنت واقعی انجام می دهیم ولی تاخیر بین کاربران و edge server ها را اندازه گیری می کنیم. صفحه وبی را با نرم افزار AJAX می سازیم که وقتی بارگذاری شد تاخیر سروری را که Akamai برای این مرورگر انتخاب می کند را به صورت لیستی از سایر سرورهای edge مربوط به Akamai اندازه گیری کند و نتایج را به ما گزارش دهد. ما از یک شرکت تجاری متوسط درخواست نمودیم تا صفحه مان را در یک اندازه کوچک جا دهد و همچنین آن را در صفحات وب مان جای گذاری نمودیم. زمانیکه کلاینت ها به این سایت های وب دسترسی پیدا نمودند، نتایج محاسبات مان را جمع آوری می کنیم.

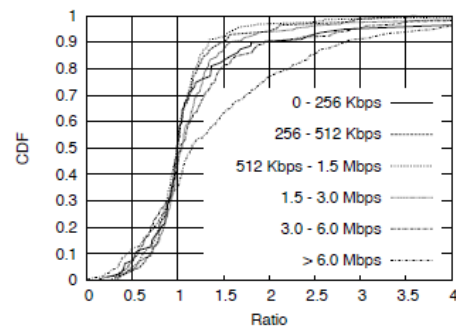
ما یک CNAME انتخاب نمودیم (a1694.g.akamai.net، توسط pcword.com استفاده شده است) که متوجه شدیم توسط Akamai به تعداد زیادی (979) از سرورهای edge داده شده است. این سرورها 343 شبکه 24 ساعته مختلف نشان می دهند، با استفاده از city+AS ابتکاری، برآورد کردیم که 168 مکان را نشان می دهند. جهت مقایسه پیکربندی فعلی با پیکربندی تلفیق شده، تعداد 979 سرور را به 10، 50 و 100 دسته با استفاده از روش دسته بندی، تقسیم نمودیم. تعداد 100 سرور را از مراکز 100 دسته ای انتخاب کرده و با استفاده از نرم افزار AJAX تاخیر را از کلاینت ها به این سرور ها را محاسبه نمودیم. جهت کاهش تعداد محاسبات، نمایندگان دسته های بزرگتر را از میان این 100 سرور انتخاب می کنیم (در دسته بندی ما، یک دسته بزرگتر به عنوان پیوندی از دسته های کوچکتر ساخته شده است، بنابراین همیشه می توانیم در این روش سرور مناسب را پیدا کنیم). این مورد نتایج ما را خراب نمی کند زیرا اختیار این را داریم که مراکز داده تلفیقی مان را ایجاد نمائیم.

نمی توانیم راهی را برای انتقال یک سربرگ Header سفارشی پیدا کنیم تا درخواست ارسال Javascript بدهد زیرا برای این کار لازم است تا یک شیء حقیقی را از یک سرور edge انتخابی Akamai بدست آوریم (بخش 4.2، بنابراین یک URL ساختگی که در تمامی موارد "Bad Request" را بازمی گرداند ارائه می دهیم. با tcpdump این پاسخ را به وسیله کلاینتی که درگیر دو RTT است تائید می کنیم که به ما اجازه دهد تا تاخیر بین کلاینت و سرور edge درگیر شده را اندازه بگیرد.

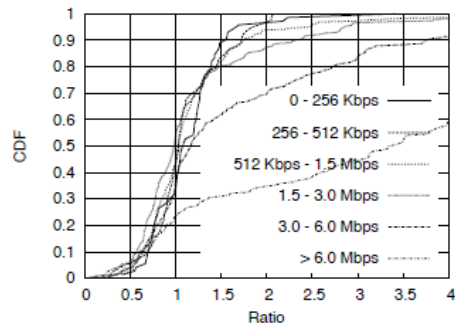
24079 محاسبه از 2926 آدرس IP کلاینت تا زمانی که این توشته را تهیه می کردیم جمع آوری کردیم. مطابق با پایگاه داده GeoIP، این نقاط برتر، یک محیط گسترده را پوشش می دهند که 47 ایالت از امریکا و 43 کشور دیگر را نشان می دهند و چنانچه تجزیه و تحلیل های آماری ما نشان خواهند داد، برای رسیدن به یک نتیجه معنادار مناسب هستند.



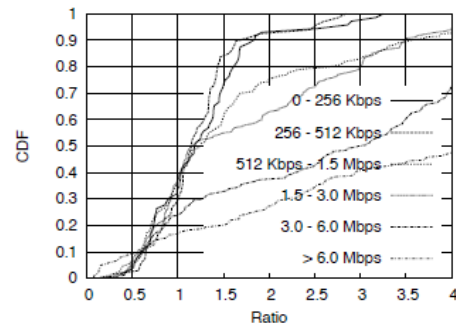
(a) 100 Data Centers



(b) 60 Data Centers



(c) 40 Data Centers



(d) 20 Data Centers

شکل 11: عملکرد یک پلتفرم Akamai تلفیق شده با سرعت دانلود متفاوت

تفاوت بین تاخیرات اندازه گیری شده از هر کلاینت تا نزدیک ترین مرکز داده تلفیق شده و سرور انتخابی Akamai را در نظر می گیریم. مقدار منفی نشان می دهد تا زمانیکه یک مقدار مثبت مزیت پلتفرم Akamai موجود را منعکس نکند، پلتفرم تلفیقی در مشاهده متناظر بهتر از پلتفرم Akamai موجود عمل می کند. این فرض در نظر گرفته می شود که پیکربندی تلفیقی، انتخاب مرکز داده را به صورت کامل انجام می دهد و بنابراین بایستی به عنوان یک مزیتی بالا در ارتباط با تلفیق در نظر گرفته شود. به هر حال متوجه شدیم که مکان هایی که کمی دورتر هستند، انتخاب سرور را آسان تر می کنند.

جهت جلوگیری از استفاده محاسبات وابسته در تجزیه و تحلیل، میانگین تمامی محاسبات از کلاینت مشابه را در نظر می گیریم، به صورتیکه هر کلاینت فقط یک نقطه داده در تجزیه و تحلیل ما قرار دهد. علاوه بر این، جهت برطرف کردن محاسبات وابسته ممکن از شبکه مشابه در مکان مشابه، فقط یک کلاینت که به صورت تصادفی انتخاب شده است را از تمام کلاینت ها با شهر مشابه و سیستم مستقل بر اساس GeoIP استفاده می کنیم. این کار تعداد نقاط داده را برای تجزیه تحلیل از 2926 به 2029 کاهش داد.

جهت ارزیابی اهمیت نتایج، فضاهای اطمینان برای میانگین های گزارش شده ایجاد نمودیم. چون توزیع مشاهدات ناشناخته است، از روش خود راه انداز غیر پارامتری جهت تخمین میانگین جمعیت و میانه و جهت ساختن فضای اطمینان برای میانگین استفاده می کنیم. به طور خاص روش فاصله افزایش دهنده تصحیح خطا [16] (BCa) با 10000 مجموعه نمونه گیری و توابع ارائه شده Matlab که هسته روش را انجام می دهد را استفاده نمودیم.

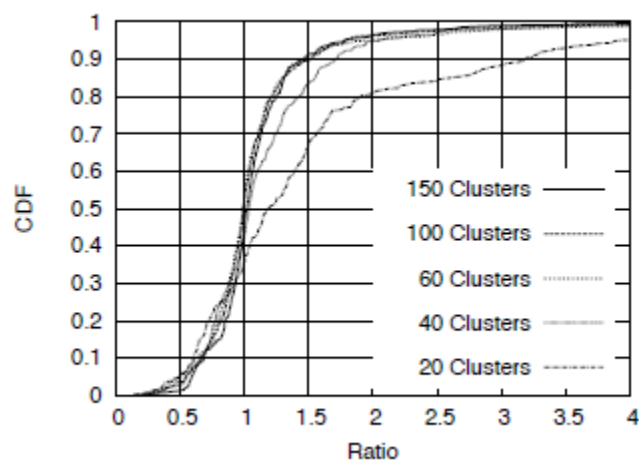
جدول 2 نتایج این محاسبات را با فضاهای اطمینان برای میانه ها که برای احتمال اعتماد 95٪ ساخته شده اند را به صورت خلاصه در آورده است. نتایج نشان می دهند که با هر دوی 100 و 50 مراکز داده تلفیقی کماکان می توان یک مرکز داده نزدیک تر به سرور انتخاب شده Akamai برای اکثر کلاینت ها پیدا نمود و متوسط فاصله به نزدیکترین مرکز داده تلفیقی از طریق تمامی کلاینت ها نیز کمتر است. در واقع کل فضای اطمینان که برای تفاوت فاصله میانه منفی است نشان می دهد که نتیجه فوق از نظر آماری واضح است و حتی حد بالاتر از فضای اطمینان نشان دهنده تفاوت RTT از بیش از 40 میلی ثانیه برای 50 مرکز داده و 70 میلی ثانیه برای 100 مرکز داده است. فقط با 10 مرکز داده، نتایج متفاوت می بینیم: وقتی کل فضای اطمینان برای میانگین تفاوت فاصله هنوز منفی است (نشان می دهد فاصله از کلاینت ها به نزدیک ترین مرکز داده تلفیقی هنوز از نظر میانگین کوچکتر است از سرور انتخابی Akamai در پلتفرم جاری) تفاوت میانگین بیانگر مزیت پلتفرم جاری است و نشان می دهد که پلتفرم Akamai جاری، برای اکثر کلاینت ها بهتر عمل می کند. همچنین متوجه شدیم که تفاوت میانگین بین پلتفرم های Akamai و تلفیقی در تمامی پیکربندی ها کمتر است. این استنباط می شود که میانگین تفاوت در برابر Akamai به وسیله انتخاب های سرور ضعیف گاه به گاه منحرف شده اند، که یک مورد یافته شده از بخش 5.2 را تأیید می کند.

به طور کلی نتایج ما نشان می دهند که حتی با تلفیقی که به صورت قابل توجه صورت گرفته، می توان بیشتر از ترکیب در مورد عملکردهایی که از تلفیق متاثر شده اند، انتخاب سرور را انجام داد.

Configuration	Best in 10	Best in 50	Best in 100
Sample Median	16.00	-5.00	-18.40
Sample Mean	-15.66	-54.43	-81.53
Bootstrap Median	15.954417	-5.013368	-18.260679
Bootstrap Mean	-15.695077	-54.313225	-81.697586
95% conf. interval of the mean	[-33.816798, -5.129345]	[-72.209521, -43.798741]	[-99.081914, -70.692385]

جدول 2: تفاوت مسافت RTT (به میلی ثانیه) از کلاینت ها تا نزدیک ترین مرکز داده در یک پلتفرم تلفیقی فرض

شده و تا سرور انتخاب شده Akamai در پلتفرم حال حاضر (مطالعه زنده)



شکل 12: عملکرد یک پلتفرم Akamai تلفیقی برای اتصالات سرعت جاری

7. نتیجه

این مقاله مطالعه گسترده ای از عملکرد CDN مربوط به Akamai را ارائه می دهد. با استفاده از پلتفرم محاسباتی DipZoom توانستیم تعداد زیادی از سرور های Edge مربوط به Akamai را برای مطالعه مان کشف کنیم و صدها نقطه قوت برای محاسبات عملکردمان به کارگیریم. به نظر ما، این مقاله اولین مطالعه ای است که برای ارائه یک ارزیابی مستقیم و مستقل از بهبود عملکرد دانلود های پر سرعت Akamai صورت گرفته است. سپس کیفیت انتخاب سرور Akamai را در مقیاسی وسیع مطالعه نمودیم. از دیدگاه عملکرد، این مقاله نوعی ارزیابی از امکان تلفیق پلتفرم Akamai به چندین مرکز داده بزرگ است. ما دریافتیم که تلفیق به صورت آشکار و قابل توجه، بدون اینکه عملکرد پلتفرم به طور قابل محسوسی پایین بیاید امکان پذیر است.

REFERENCES

- [1] Akamai Technologies. <http://www.akamai.com/html/technology/index.html>.
- [2] Akamai Technologies. Facts and Figures. http://www.akamai.com/html/about/facts_figures.html.
- [3] Akamai Technologies. Perspectives. <http://www.akamai.com/html/perspectives/index.html>.
- [4] A. Biliris, C. Cranor, F. Douglis, M. Rabinovich, S. Sibal, O. Spatscheck, and W. Sturm. CDN brokering. In 6th Int. Web Caching Workshop, 2001.
- [5] C. Canali, V. Cardellini, M. Colajanni, and R. Lancellotti. Evaluating user-perceived benefits of content distribution networks. In Int'l Symp. on Perf. Eval. of Comp. and Telecomm. Sys. (SPECTS), 2004.
- [6] cURL - tool for transferring files with url syntax. <http://curl.haxx.se/>.
- [7] Dipzoom - deep internet performance zoom. <http://dipzoom.case.edu>.
- [8] Intelligent content distribution service. AT&T Inc. http://www.business.att.com/service_fam/overview.jsp?repopid=ProductSub-Category&repopitem=eb_intelligent_content_distribution&serv_port=eb_hosting_storage_and_it&serv_fam=eb_intelligent_content_distribution&segment=ent_biz.
- [9] K. Johnson, J. Carr, M. Day, and M. Kaashoek. The measured performance of content distribution networks. In 5th Web Caching Workshop, 2000.
- [10] B. Krishnamurthy and J. Wang. On Network-Aware Clustering of Web Clients. In The ACM SIGCOMM, Aug. 2000.
- [11] B. Krishnamurthy, C. Wills, and Y. Zhang. On the use and performance of content distribution networks. In The First ACM SIGCOMM Internet Measurement Workshop, pages 169–182, 2001.
- [12] Limelight Networks brings dynamic streaming to the web. <http://blog.llnw.com/2009/08/limelight-networks-brings-dynamic-streaming-to-the-web/>.
- [13] Limelight Networks. <http://www.limelightnetworks.com/platform/cdn/>.
- [14] MaxMind. <http://www.maxmind.com>.
- [15] J. Pang, A. Akella, A. Shaikh, B. Krishnamurthy, and S. Seshan. On the responsiveness of dns-based network control. In ACM IMC, pages 21–26, 2004.
- [16] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann. Improving content delivery using provider-aided distance information. In ACM IMC, 2010.
- [17] Rapid Edge Content Delivery Network. <http://www.rapidedgecdn.com/>. [18] The R project for statistical computing. <http://www.r-project.org>.
- [19] Savvis Inc. <http://www.savvis.net/corp/Products+Services/Digital+Content+Services/Content+Delivery+Services/>.
- [20] A. Sherman, P. A. Lisiecki, A. Berkheimer, and J. Wein. Acms: the akamai configuration management system. In NSDI, pages 245–258, 2005.
- [21] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante. Drafting behind akamai (travelocity-based detouring). In SIGCOMM, pages 435–446, 2006.
- [22] A.-J. Su and A. Kuzmanovic. Thinning akamai. In The 8th ACM IMC, pages 29–42, Oct. 2008.
- [23] S. Triukose, Z. Al-Qudah, and M. Rabinovich. Content delivery networks: Protection or threat? In ESORICS, pages 371–389, 2009.
- [24] S. Triukose, Z. Wen, and M. Rabinovich. Content delivery networks: How big is big enough? (Poster paper). In ACM SIGMETRICS, 2009.
- [25] Z. Wen and M. Rabinovich. Network distance estimation with dynamic landmark triangles. In ACM SIGMETRICS, pages 433–434, 2008.
- [26] Z. Wen, S. Triukose, and M. Rabinovich. Facilitating focused internet measurements. In ACM SIGMETRICS, pages 49–60, 2007.