Universal Online Sketch for Tracking Heavy Hitters and Estimating Moments of Data Streams

2020 INFOCOM Toronto (Q1)

3 citations

Shigang Chen: University of Florida (Computer Networks, Big Data, Internet Security)

اندازه گیری ترافیک در مسایل مختلف حوزه مانیتورینگ شبکه مثل تشخیص حملات، تشخیص آنتروپی می‌تواند موثر باشد. برای تشخیص آنومالی (یکی از راهکارهای تشخیص حملات) بسته‌هایی که به صورت استریمی وارد می‌شوند را بر اساس هدرشان به جریان‌های مختلفی تقسیم میکنند و حال اطلاعات آماری آن جریانها را به دست می‌اورند [1] و از این اطلاعات آماری برای تشخیص بی نظمی استفاده می‌شود. با توجه به منابع محدود دستگاه‌های میانی، الگوریتم‌های زیرخطی [2] مختلفی به نام اسکچ‌های عمومی مثل: count min و count sketch ارایه شدند. اما مشکلشان این بود که هر کدام برای یکی از مسایل اندازه‌گیری طراحی شده بودند و فراهم کردن اسکچ‌های مختلف برای مسیله‌های دیگر فضای زیادی می‌طلبید. لذا Universal sketch هایی بایستی طراحی می‌شد که نه تنها جریان‌های بزرگ heavy hitter را تشخیص می‌دادند بلکه اطلاعات آماری کلی نیز در مورد توزیع ترافیک یا همان moment ها نیز ارایه می‌دادند. یکی از روش‌های موفق در این زمینه univmon می‌باشد که دراصل چندین جدول هشینگ دارد که هر بسته معمولاً چندبار ذخیره [3] می‌شد و لذا حجم زیادی را در بر می‌گرفتند و سرعت مان را نیز کند می‌کردند. لذا قصد داریم روشی به نام LUV ارایه دهیم که از اسکچی بر پایه نمونه برداری تدریجی [4] (به جای نمونه برداری سلسه مراتبی [5]) می‌باشد که از ActiveCM+ در عوض CS استفاده می‌کند که سربار عملیات هر بسته را می‌کاهد و دارای نرخ گذر بالا (مانند اسکچ‌های تکی) می‌باشد. در خلاصه باید گفت، اسکچ‌های قبلی به منظور اندازه گیری جریان می‌بودند و اسکچ‌های جهانی نیز مشکلات سربار دسترسی حافظه دارند.

مانیتورینگ شبکه با مفاهیم زیر مرتبط می‌باشد:

۱-Flow moment: اطلاعاتی که از تمامی جریان‌های استریم در یک لحظه می‌توان به دست آورد.در واقع وضعیت کل ترافیک را نشان می‌دهد. می توان در بازه‌های زمانی lg را حساب کرد و به اصطلاح یک سری زمانی از اون معیار به دست آورد و سپس با مقایسه آن با یک مقدارآستانه ای، رویدادهای آنومالی را تشخیص دهیم. $L_g = \sum_{1 \le f \le \mathcal{F}} g(n_f)$

این که تابع g چی باشد، شناسه‌های مختلفی از ترافیک را نشان می‌دهد که می‌تواند در عملیات مانیتورینگ مفید واقع شود. (به مقاله مراجعه شود). برای مثال برای شناسایی حملات منع خدمت می‌توان هاست‌هایی که به سمت آنها جریانات بیش ازحدی ازطرف منابع مختلف می‌باشد را شناسایی کنیم. برای اینکار می‌توان از یک اسکچ برای ذخیره سازی هاست های موردنظر و $g(x)=x^0$ قرار داد.

۲-Heavy hitter: جریانی است که سایز آن تاثیر زیادی بر روی یک flow moment می‌گذارد یا به عبارت ساده حجم زیادی از ترافیک را تشکیل می‌دهند: $H_g = \{f \mid g(n_f) \ge \alpha L_g\}.$

---

[1] Flow-level statistics
[2] Sub-linear
[3] record
[4] Progressive sampling
[5] Hierarchical sampling

ضبط هردوی heavyhitter ها و moment ها به شناسایی آنومالی‌ها و مانیتور شبکه کمک خواهد کرد. به عنوان مثال برای تخمین آنتروپی و کنترل ترافیک ارسالی به یک IP مشخص، می‌توان جریانات را بر اساس آدرس مبدا تقسیم کرد و آنتروپی جریان را برابر مقدار زیر تعریف کرد. حالا هر تغییری شگرفی در این مقدار به معنای تحت حمله قرار گرفتن یک آدرس می‌باشد.

$$E = -\sum_{1 \le f \le \mathcal{F}} \frac{n_f}{n} \log \frac{n_f}{n}$$

معیارهایی که برای ما مهم است:

- Heavy hitter estimation
- Moment Estimation
- Memory Overhead: اسکچ‌ها چون روی سوییچ‌ها و روترها می‌باشند، لذا فضای حافظه کمی‌دارند و باید بین دقت اسکچ و میزان فضای موردنیاز، سبک-سنگین[6] کرد. باید اون کران پایین دقت محاسبات را بتوان تضمین کرد.
- Packet-Processing cost: محاسبات هشینگ و دسترسی به حافظه که برای هر بسته صورت می‌گیرد. هشینگ نیز به دلیل نمونه برداری بسته یا برای دستیابی به مکان تصادفی در حافظه می‌باشد. ازآنجا که برای هربسته نمونه برداری تنها یک بار انجام می‌شود، بیشتر هزینه به خاطر محاسبات هشینگ متعدد برای نوشتن/خواندن حافظه می‌باشد.

راهکارهای ارایه شده قبلی:

- اسکچ‌های اندازه‌گیری سایز هر جریان: Count Min از یک آرایه دوبعدی شمارنده‌ها شامل d سطر که معمولاً برابر ۴ می‌باشد. هر بسته را dبار به یکی از این شمارنده‌ها هش می‌کند و مقدارش را افزایش می‌دهد. Conservative Update همه هش‌ها را می‌خواند و کوچکترین آنهارا آپدیت می‌کند. اگر سایز یک فلو را کویری بزنیم، این دو اسکچ کوچکترین مقدار را از بین d تا هش انتخاب می‌کنند و تخمین می‌زنند. Count Sketch به جای افزایش شمارنده‌های هش شده، مقدار 1-/1+ هش شده به شمارنده‌های هش شده اضافه می‌کند. که تابع هش 1-/1+ به صورت شبه تصادفی، یک جریان را به 1+ یا 1- نگاشت می‌کند. و وقتی کویری بزنیم مقدار میانگین یا میانه اون d تا شمارنده را برمی‌گرداند.VACو.. نیز از جمله راهکارهای دیگر می‌باشند.
  برای تمامی اسکچ‌های گفته شده، دقت اندازه‌گیری به دو عامل بستگی دارد:
  1. اگر تعداد بسته‌های جریان‌ها افزایش یابد، شمارنده‌های اشتراکی جریان‌ها (مثلاً به دلیل تصادم هش) نرخ خطای بالا خواهند داشت .
  2. حافظه اختصاص داده‌شده به اسکچ هر چه بیشتر باشد، شمارنده‌ها افزایش می‌یابد و نرخ خطا به دلیل اشتراک کمتر، کاهش خواهد یافت
- اسکچ‌های جهانی: توانایی اجرای چندین عملیات اندازه‌گیری مختلف را دارند. دیگر ماژول‌های جدا برای اندازه‌گیری سایز جریان، شناسایی heavy hitter ها و moment ها لازم نیست.OpenSketch از یک سری ماژول‌هایی برای استفاده مشترک بین چندین task، استفاده می‌کند. اما Univmon از راهکار دیگری استفاده می‌کند. از چندین CS (رسماً ۱۵ تا) استفاده می‌کند. با احتمال $1/2^i$ جریان‌ها را نمونه برداری و پکت‌هایشان را در CS مربوطه ضبط می‌کند.

حال به معرفی نوآوری های استفاده شده در LUV می‌پردازیم:

نمونه‌برداری تدریجی : به جای نمونه برداری سلسله مراتبی که در univmon استفاده می‌شود و هر بسته را در تعداد متغیری اسکچ(میانگین ۴ تا) ضبط می‌کند، هر بسته را دقیقاً در یک اسکچ ذخیره می‌کندو لذا شمارنده‌ها نویز کمتری خواهند داشت و دقت بالاتر خواهد رفت. روش ما و univmon هر دو از چندین اسکچ برای اندازه گیری استفاده می‌کنند.

[6] tradeoff

ActiveCM+ Sketch: حافظه کمتر، هش‌های کمتر و بارحافظه کمتر نسبت به univmon استفاده می‌کند. LUS برای هر بسته حداکثر ۵ بار هش و ۲بار دسترسی به حافظه نیاز خواهد داشت. دلیل دقت بهتر در حافظه یکسان نسبت به CS استفاده از شمارنده‌های بیشتر می‌باشد، چون که خروجی شمارنده‌ها نیز تعداد بیت کمتری خواهند داشت لذا می توان تعداد شمارنده‌ها را افزایش داد. داده ساختار فشرده ActiveCM Sketch، داده‌ها را به صورت احتمالی ضبط می‌کند و نسخه بهبود یافته آن +ActiveCM، امکان شناسایی heavy-hitter ها را نیز خواهد داشت. ActiveCM از یک داده ساختار فشرده و دو روش زیر استفاده می‌کند:

- Counter sharing
- Counter compression

ActiveCM از عملیات زیر پشتیبانی می‌کند:

- ضبط بسته رسیده شده: زمانی که بسته جدید رسید، تمامی شمارنده‌های مجازی متناظر با جریان را یک واحد افزایش می‌دهد (البته با احتمال $\frac{1}{2^{Pc_f[i]\cdot a}}$:) که البته می‌توان آن را به صورتی تغییر داد که تعداد بایت‌های جریان را نیز بشمارد. همانطور که می‌دانیم d تا هش به ازای هر بسته بایستی اجرا کرد.اما به جای اینکار، یک هش ۶۴ بیتی یکجا از جریان حساب می کنیم و اونو به d قسمت تقسیم می‌کنیم.
- کویری و برگداندن سایز یک جریان: کوچکترین مقدار از بین d شمارنده جریان را برمیگرداند.

+ActiveCM : سایز هر جریان را تخمین می‌زند مثل CM و CS اما با حافظه‌ای کمتر. بعدن ارتقاش میدیم با استفاده از یک مین هیپ ، k تا جریان و شمارنده متناظرشان را شناسایی کند ← heavy hitter. همانند یک فیلتر اولیه از هیپ کمینه استفاده می‌کند.

کران‌دار بودن: در این بخش ثابت می‌کند که مقدار کویری بازگردانده شده، کران دار می‌باشد:

$$Pr\{\hat{n_f} \geq n_f + (n - n_f)\frac{2d}{m}\} \leq \left(\frac{1}{2}\right)^d$$

اجرا و ارزیابی: در یک حجم محدودی از حافظه، +ActiveCM را با اسکچهای دیگر شناسایی heavy hitter ها از نظر کارای مقایسه می‌کنیم. سپس LUV را با univmon برای تخمین درلحظه moment ها مقایسه می‌کند. نرخ گذر بسته‌ها که متاثر لز سرعت عملیات هش(درج بسته‌ها) و دسترسی به حافظه(درج و کویری) هست، را نیز بررسی می‌کنیم. داده‌های آزمایش نیز از دیتاست cadia می باشند.

- شناسایی Heavy hitter ها با یک حافظه محدود :
  - نرخ گذر: در مقایسه با اسکچهای دیگر نرخ گذر بالاتری خواهد داشت.
  - دقت تخمین
    - Average Relative Error(ARE)
  - میزان تاثیر اندازه جدول مجازی یا همان d را بر روی دقت +ActiveCM نیز بررسی می‌کنیم. و مقدار بهینه آن را به دست می‌اوریم
- مقایسه با Universal Sketch های دیگر: Univmon از ۱۴ لایه سلسله مراتبی استفاده می‌کند. درحالی که LUS از ۱۰ تا subsketch که ActiveCM+ شامل ۴ ستون می‌باشند، استفاده می‌کند. در موارد زیر برتر می‌باشد:
  - نرخ‌گذر بسته‌ها
  - دقت تخمین سایز جریان. چون بسته‌ها را تنها در یک subsketch نمونه‌برداری شده، وارد می‌کند، نویز و خطای کمتری خواهد داشت.

- همچنین تاثیر تعداد subsketch را نیز بررسی کردیم. اگر در univmon لایه‌ها را افزایش دهیم، دقت افزایش می‌یابد اما در LUS با افزایش تعداد subsketch ها چون آخرین subsketch یک جدول هش می‌باشد، که تمامی جریان‌های نمونه برداری شده را در خود جای داده است، این امر تاثیری نخواهد داشت.