Gerhard van Andel *

*Colorado University, CSCI/ECEN5273

ABSTRACT

It's been more than 20 years of web crawling, back from the days when a book contained all the websites in the world to now where companies like Google have crawled millions of pages a day. [1] In the past few years though a change has been growing in how people search the web. In this change Google started working on mobile-first indexing, the crawling of the web using a smartphone Googlebot. Forcing companies that if they wanted to stay competitive in the web market they needed to adapt to these new found habits of the 21st century internet traveler. The rules are not so much enforced but created superficially through Google and other companies page ranking systems. The great and powerful placing of what links will be first in the modern age. Being at the top of a search can be the difference in getting business or being cast out to the depths of the second page. I used my term project to study the web and load times across regions to try and simulate how user experience of a web site can change depending on where you are.

General Terms

Search Engines, Information Retrieval, PageRank, User Experience

1. INTRODUCTION

With this search engines still typically look at the desktop version of a page's content to evaluate its relevance to the user, but in the last four years the growth of mobile devices have increased dramatically with around 90% of all internet users having mobile access. So why should any of this matter, while most sites continue to grow in online presence the need for both a speedy connection for the client along with the server. But waiting for customers to complain on twitter for your web site to be slow is not what you want for your business. Customer experience metrics are important to alert staff of production issues across web sites just like any other service level indicator (SLI), an SLI is a carefully defined quantitative measure of some aspect of the level of service that provides an idea on the health of the service. With this a SLI can be as simple as a web page load time. If a website does not load within three seconds the chance that the user gives up increases by 32% and goes to a different page. So monitoring traffic and response time from servers can be an important data point for web experiences across all industry sectors.

2. RELATED WORK

How do you get the top search in real estate searches well you need to know about search engine optimization (SEO). SEO is often about making small modifications to parts of your website. When viewed individually, these changes might seem like incremental improvements, but when combined with other optimizations, these changes could have a noticeable impact on a site's user experience and performance in the world of organic search results. So how do sites get added to search engines?

The first step is the crawling, crawling pages that are known because the search engine has already crawled them. Other pages are discovered when the search engine follows a link from a known page to a new page. This can be done from a single web site or the owner of the website can publish a sitemap normally linked from a robots.txt file that lives at the root of the site. Once a page is discovered, figuring out what a page is about is the cream of the search engine industry. Each company holds a different way to classify and index each web page on the internet, this can be improved by using page headers and content to convey content. Once the pages have been crawled and indexed into the search engine now you need a user to search for something. [1]

Example would be cat food, but maybe the user wants a recommendation to buy cat food and needs to provide the location of the search to improve results as you don't need the results of cat food stores in Iceland when you live in Boulder. All of this just to find your furry friend some food, but now when you click on the web page the search engine has provided you it fails to load or worse it loads very slowly. So now the search engine will filter the fastest loading web pages to the top of the results. With this it is important to provide a good user experience with the loading of a web page. Companies around the world currently provide uptime and load time responses.

3. PROBLEM CHARACTERIZATION

Speed is a crucial aspect in providing a pleasant user experience to visitors of your site. It ensures that a visitor's limited attention span and time are spent on actual content and not wasted waiting for images and scripts to load. Studies have shown that users will not tolerate more than a 4 second load time. [2] If your site fails to offer a quick response, your users will leave and that can be a loss in advertising revenue or if you run an e-store through the web site loss of business. gMonitor a simple way to monitor, and crawl web pages to track the user experience.

4. MILESTONES

4.1 Checkpoint 1

Have a working prototype pulling http/https based metrics. I will use python requests to pull data from a website then using beautiful soup to parse the web site, pull all the links from the site and then follow the links to determine site speed on leaf sites. Using redis to lock on the domain so workers can follow robots.txt and then store this data into google cloud storage. To test this ecosystem I used docker-compose this allowed me to make changes locally. This increased my productivity a lot as I no longer had to push the changes to the docker registry. I am still working on how to do the correlation across requests. To compare load times of the web sites.

The redis database works well but I can fill it very fast as the crawler grows exponentially. To reduce the number of tasks in the database by pushing completed jobs back into cloud storage and on request load it out of cloud storage. I did not think that link information would take up as much space as I thought but I crawled 70 web pages and that linked to more than 1200 other pages so I may have to think how to store more information elsewhere. Have two or more regions working to poll web traffic from more than one location. Then figure a way to collectively pool the data sources. I would like to keep the data locality, something that I don't need to share between regions.

4.2 Checkpoint 2

While working through the small problems of database locks I found that Redis has an expiry, so when you add an entry into the database the entry will expire the data field as the time has past. I was able to reduce the domain database from growing very large using this as the domains are generally grouped as you will have links from one page linking to pages within the same domain. This also allowed me to then get an updated version of the robots.txt after the domain would expire from the database. I also used the expiry of the domain data to ensure the release of the domain lock. If I did a deploy in the middle of a search I could have a deadlock on a message, so putting an expiry on that would ensure domain locks would be released at some point. The domain locks and rabbitmq found a separate issue in release of the message I was looking for a way to put the message at the back of the queue but from what I saw if a domain lock was being used the message would be rejected and then placed back on the message queue only to get picked back up right away.

Using a priority queue would allow for the lowering of the priority of a message along with track the number of times I have seen that message would be techniques I could use to optimize around this. Offloading completed requests from the redis database into cloud storage was also a big help as a small amount of jobs are pulled after they complete. I did look at using Google's big-table service but found that redesigning the platform for the use of big-table was more than I wanted in the last two weeks of the assignment. Along with this I found I could do a better job loading content from a website so that I could also index the loading of other content being style sheets, images, and javascript.

5. METHODOLOGY

gMonitor will take a request for an endpoint that will send a url to search. The design of gMonitor was based on a micro-service design with message queues between workers. This would allow each system to scale independently. The service will then create a task for the url and using a producer consumer model the rest interface would place a message on the spider queue, spider workers will pull the web sites for this task and store it into google cloud storage. Then the spider worker will create a task for a scanner worker to use beautiful soup and load the data linked to the original web page. The scanner work will filter out for included links that the web page contains. Then create another task for a spider to follow those links to see how the user experience will be from following links on the web page. Then once this is complete it adds a message on the last queue the cleaner. The cleaner worker helps reduce the footprint of the cache and pushes items that have been scanned and updated into deep storage on google cloud. This helps keep unused data off the cache and allowing data that is being accessed more frequently to be on the cache. This design can be visually represented in figure 1. If a user requests the data or the results of the scan the controller can pull this data from storage with an increase of time to the call.

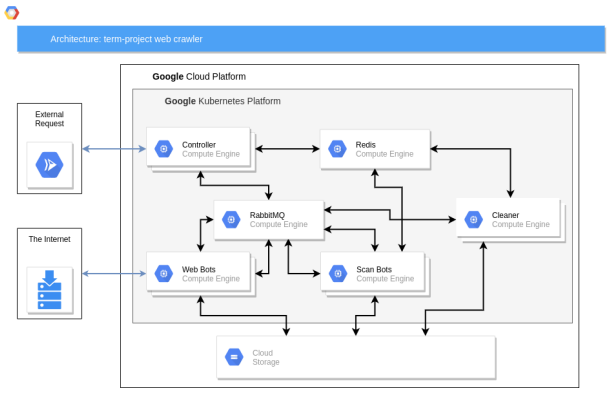

Figure 1: Architectural Diagram

Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling the containerized applications along with providing a simple interface to Google infrastructure like cloud storage. With this I was able to use the same template in three regions, United-States central, Europe central, and East Asia. In these three regions I then tried to keep my requests similar but with the nature of the distributed system I did not control the queues and what jobs were being worked.

6. INSIGHTS GLEANED

The first test I ran was in the US central region. For the data set I used the popular web site news.ycombinator. This web site links to many different types of web sites and uses a voting system to bring favorite / popular content to the top of the front page. In doing so I used this as my base line as to what to expect from the other regions. This is shown in figure 2

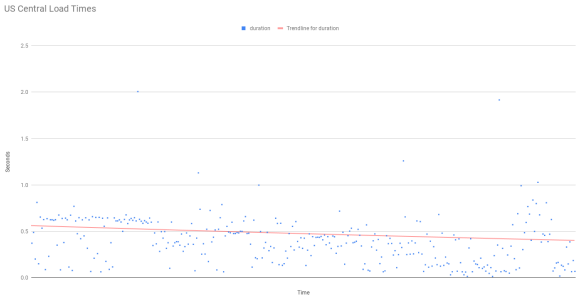

Figure 2: Central United States Load Times Graph

In the Europe region I was surprised to see the load times were not so different. I am making an assumption that news.ycombinator uses some sort of content delivery network (CDN). This allowed the same data to be easily provided in the European region along with the web site probably has more active users as English is a common language. This is shown in figure 3

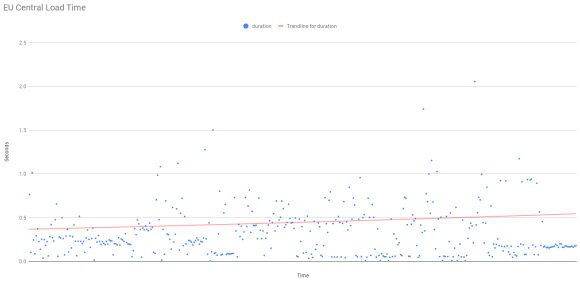

Figure 3: Central Europe Load Times Graph

In the East Asia region I was surprised to see the load times were not so different to cause an alarm. I am making an assumption that news.ycombinator uses some sort of content delivery network (CDN). With this the request duration's were above average compared to the United-States and Europe. This is shown in figure 4

7. FUTURE WORK

I would want to also start collecting on the location of the remote server along with using troubleshooting tools such as traceroute to better explain the result I obtained.

Having more time I think I would have tried to implement the current and future industry standards in monitoring speed for web pages. These are PageSpeed from Google and YSlow from Yahoo. Along with some of the new advancements from increased usage of HTTP/2. When you

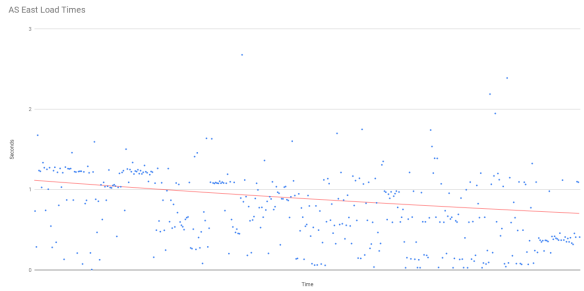

Figure 4: East Asia Load Times Graph

profile a web page with PageSpeed, it evaluates the page's conformance to a number of different rules. These rules are general front-end best practices you can apply at any stage of web development. However there are differences in how YSlow and PageSpeed perform the calculations. Each algorithm analyzes a page using a set of rules that they believe are most relevant to page speed and performance. Most of the rules overlap or are very similar to each other, but in general your scores should be comparable.

8. CONCLUSION

Although the results were different they were not as different as I was expecting. As stated in the future work I would want to track how the tcp connections for the http get requests are being made more, and to do this I would need to use troubleshooting tools to ensure the request is being sent to the correct location. It has been fun to work on the project as I did enjoy making a small service using micro service architecture and using a third party message queue.

9. REFERENCES

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. 1998.

[2] B. Jones, B. Rogan, C. Stahl, D. Creager, H. V. Madhyastha, I. Grigorik, J. E. Tuttle, L. Chen, M. Efimov, P. Papageorge, and S. Burnett. Network error logging: Client-side measurement of end-to-end web service reliability. In *17th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2020*, 2020.