HIGH-PERFORMANCE APPLICATION

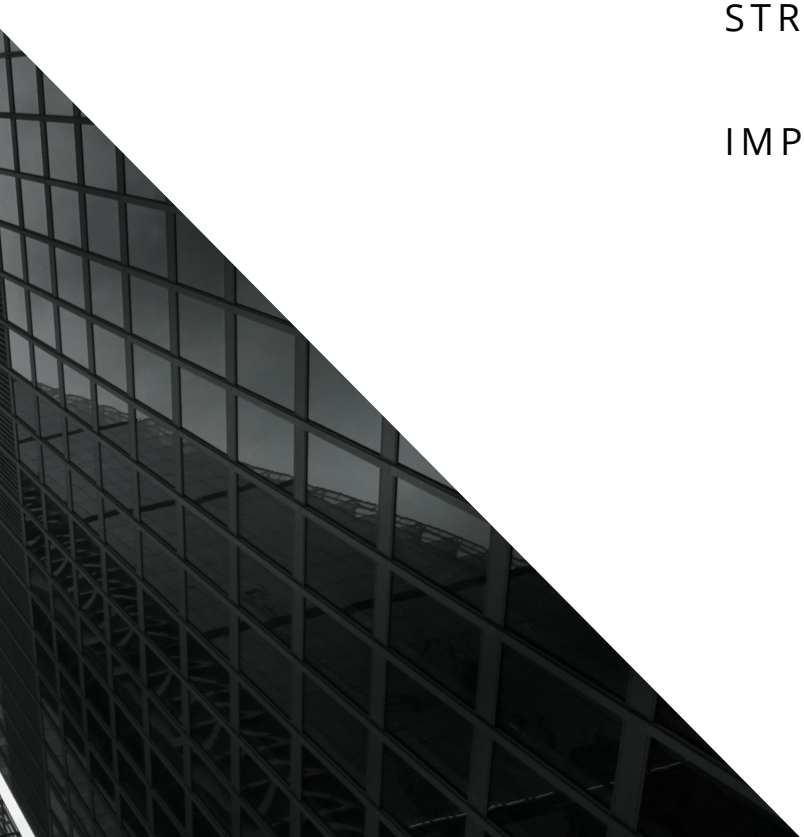
PERFORMANCE IS THE KEY TO THE SUCCESS OF AN APPLICATION

CACHING AN APPLICATION

2022

Table of

CONTENTS



INTRODUCTION

Any application needs four key things, that is visualisation, data, security, and optimisation. The main objective of this case study is to optimise the application, which is data enriching and has a data source in terabytes. To provide the user with a very smooth application experience is our main motive.

APPLICATION KEYS

User Interface or experience is the key for any application to be successful. To give the user the best experience of a data-enriched application, refresh and loading time is the main key. For an app having data in terabytes, it may take around a minute or more to load cause of the big data. It will result in a bad user experience with a loss of interest in it, irrespective of the data or quality of the application. If the app is lagging then nothing will work.

"If your application is lagging then boot up else the future will be far"

-Utkarsh Shukla

CHALLENGES

"Remember this: Be kind to your mind."
– @reallygreatsite

High Traffic

High traffic means a large number of simultaneous requests from users expecting fast load times.

Auto Scaling

Auto Scaling means automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost.

Low Latency

Low latency is the ability of a web application to provide responses with minimal delay.

SOLUTION

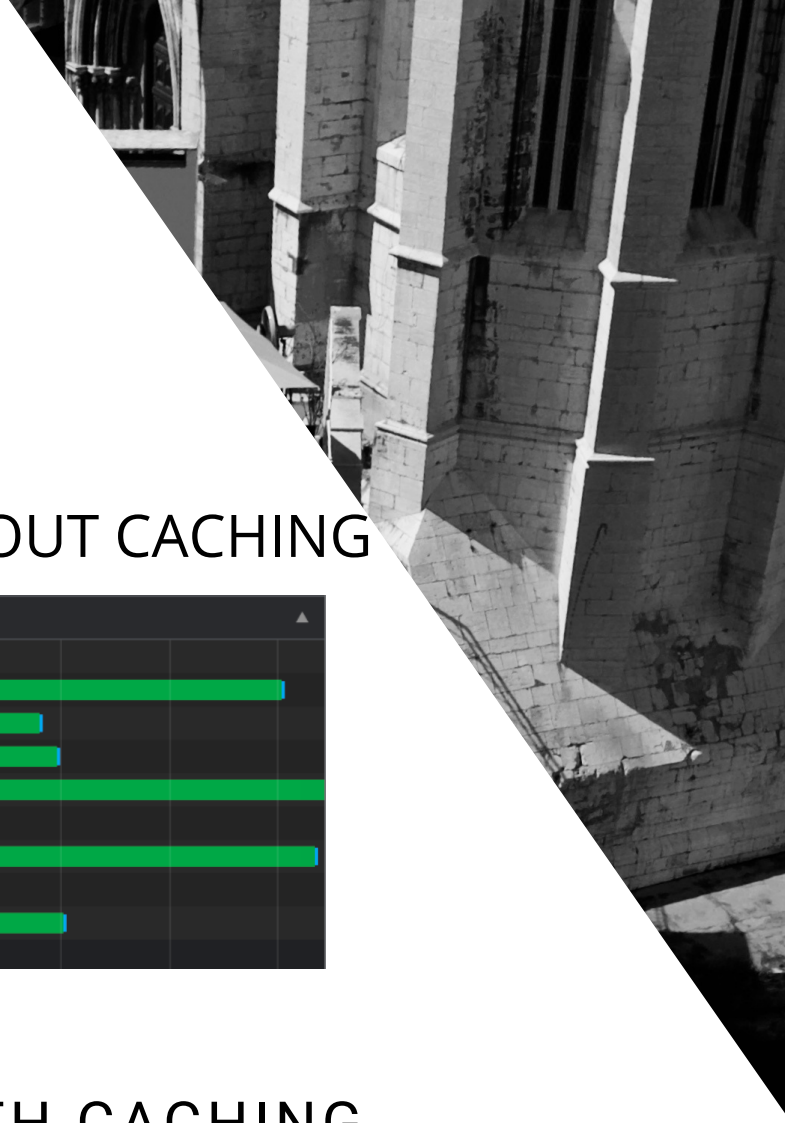
- The first solution that came up was to do caching.
- The next question was to do server side or client side caching.
- Since in big data applications data gets updated every week and caching on the client-side will be very hectic, it is better to go on with the server cache.
- Now the next question is, which caching technique should we use and either to go with a server or serverless caching.
- The best caching technique that we have found out after discussing multiple solutions was Redis using ElastiCache. A serverless caching technique.
- Redis was preferred as it can change the data in place without having to re-upload the entire data value.
- Within serverless applications caching typically means faster API response times, faster page load speeds, reduced latency, faster query speeds, etc.

BENEFITS

- Amazon ElastiCache allows you to seamlessly set up, run, and scale popular open-source compatible in-memory data stores in the cloud.
- Boost application performance, reducing latency to microseconds.
- Scale with just a few clicks to meet the needs of your most demanding, internet-scale applications.
- Reduce costs and eliminate the operational overhead of self-managed caching.
- Build with your choice of Redis or Memcached, two popular open-source caching technologies.
- Cache your data to reduce pressure on your backend database, enabling higher application scalability and reducing operational burden.
- Use ElastiCache to store non-durable datasets in memory and support real-time applications with microsecond latency.

STRATEGY

- We are caching the responses of the APIs in our Redis database.
- The Key of the cache data will always be having the URL in it, with the query params in case of GET calls and the request body in the case of POST calls.
- This is done so that every different request can be cached.
- If the same parameters are passed at a different sequence then it is treated as the same key, because it returns the same response.
- Now for the purge and ttl thing, we are currently going to use a one-week duration or as per the data updating policy.
- We can create a glue job that is attached to every glue job that alters the table used in the application.
- This glue job will flush the database whenever a table is altered only the cache of that particular table will be flushed.
- This is done so that no data mismatch can occur because of the already cached data.
- We have also created a script that will run every time a code is pushed to the main(dev, uat, and prod) branch.
- This script will clear the Redis database so that the new API responses can be cached, and no outdated response is shown.

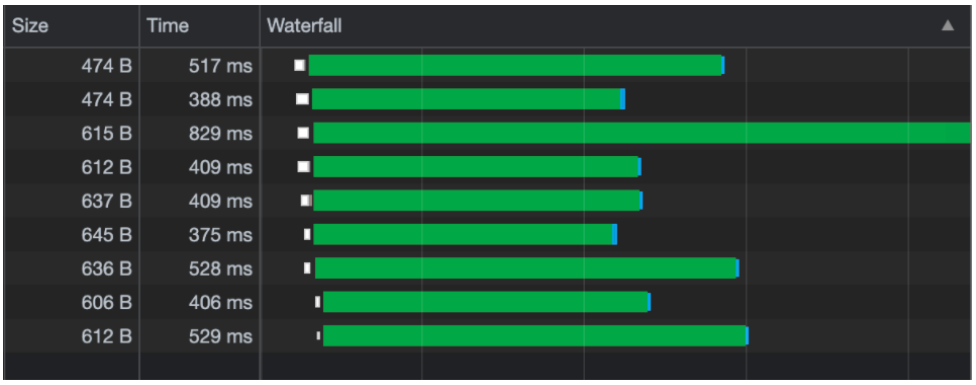


IMPACT

RESPONSE TIME WITHOUT CACHING

Size	Time	Waterfall
474 B	1.52 s	
473 B	4.39 s	
615 B	2.13 s	
613 B	2.31 s	
636 B	4.82 s	
649 B	1.57 s	
636 B	4.69 s	
607 B	1.58 s	
612 B	2.33 s	

RESPONSE TIME WITH CACHING



Size	Time	Waterfall
474 B	517 ms	
474 B	388 ms	
615 B	829 ms	
612 B	409 ms	
637 B	409 ms	
645 B	375 ms	
636 B	528 ms	
606 B	406 ms	
612 B	529 ms	

- As we can see from the above results that there is a great change in the Response time.
- The percentage change in the response time is more than 90 percent.

CONCLUSION

- To improve the performance of any application an optimised solution is essential.
- For a data driven application having a refresh rate of more then a day, caching is very important.
- We can cache the data for all the API'S at a very low cost.
- Not even we have to worry about the infrastructure and load balancing.
- In this way, we can say that AWS ElastiCache is a very optimised solution to implement caching.

by Utkarsh Shukla
(Digital Consultant Statusneo)

Date 12 Feb 2022