CS 443 Cloud Computing & (Mobile) Applications

Design Report

Spring 2020

Team Number: 5

Burak Kırımlı - 21400690

Ali Can Zeybek - 21401162

Mert Saraç - 21401480

1. System Design

We decided to choose a microservice architecture to enhance scalability. The architecture consists of 10 microservices. Those are “redirection-service”, “authentication-service”, “link-retrieval-service”, “link-analytics-service”, “link-store-service”, “user-analytics-retrieval-service”, “user-link-analytics-retrieval-service”, “user-past-link-analytics-retrieval-service”, “link-deletion-service”, “user-past-link-analytics-arrangement-service”. All incoming requests go through the Zuul API Gateway then sent to corresponding microservice. All microservices registered to Eureka Naming Server and all requests directed via Ribbon loadbalancer. All requests also assigned an id by Sleuth and went through an MQ to Zipkin for tracing. All requests except direct GET requests to shortened URLs and registration requests require authentication which is provided by “authentication-service”. “Authentication-service” first validates the user and then sends back a token which will be validated by API Gateway afterwards. The GET requests to shortened URLs are handled by “redirection-service” who asks to “link-retrieval-service” for specific URLs then redirects users and informs “link-analytics-service” in order to update that links statistics. Shortened URL storages handled by “link-store-service” who update database according to users request. Users statistics are retrieved via “user-analytics-retrieval-service” who call “user-link-analytics-retrieval-service” for link analytics. “User-link-analytics-retrieval-service” finds and returns all the non-expired links created by a specific user. For the users expired link statistics “user-past-link-analytics-retrieval-service” is called. “User-past-link-analytics-retrieval-service” finds and returns the users expired link statistics. For the deletion of expired links “link-deletion-service” works in specific time periods and both deletes the expired links from the database and informs “user-past-link-analytics-arrangement-service” who is responsible for updating users statistics for links that are expired. Also hystrix will be used for fault tolerance on every service.

2. Details of the System

As mentioned, the system will use microservice architecture. While using microservice architecture our aim is to create a 12-factor app. External APIs and libraries to be used are; spring boot, spring cloud, RabbitMQ, feign, ribbon, eureka, hystrix, sleuth and zipkin. For system wide analytics time passed in each service for each request,cpu time of service process for each request,memory usage of services for each request, and total storage usage of overall system will be monitored. For user statistics number of redirections for each link will be monitored until the expiration of those link, users all time total link clinks and number of links created by any user will be monitored.

3. SLAs(Service Level Agreements)

3.1. SLOs(Service Level Objectives)

- 99.5% of the redirection requests to original URLs when a short links of that URLs are accessed are going to be handled successfully.
- Expiration of 99.2% of short links are being handled by the system successfully (according to user-provided rules).
- The system is going to provide “link analytics” for the users and “system-wide analytics” for the admins, these analytics are going to be updated automatically within 2 seconds and these analytics are going to be accurate 99.0%.
- The system is going to be available 99.3% of the time.
- Redirections are going to happen with at most 5 seconds latency per request.
- Short links are going to be very hard to predict (i.e. the possibility for predicting a link is 1/56800235584)

3.2. Remediation Policies

- The short links that the system produced are going to be very hard to predict to protect our users and their links from possible hacker attacks 6 characters are going to be used for security (26 uppercase letters + 26 lowercase letters + 10 numbers = 62 different option for each character, for 6 characters there are 56800235584 combinations).
- Expiration of short links are being handled by the system to protect links from being expired by others.

- Analytics (“link analytics” for the users and “system-wide analytics” for the admins) are going to be updated in regular time intervals to provide the users and admins accurate information.
- The system is going to be available 99.3% of the time to let the users benefit from the system without any outage.

3.3. Penalties/Incentives Related to Objectives

- Main incentive is providing high percentage of availability for the users.
- The most important penalty is to lose users’ trust.
- If short links are easy to predict, the system becomes easy to hack.
- If expiration of the links cannot have handled by the system automatically, there may be some failures in the system due to the storage.

3.4. Exclusions and Caveats

- Due to system maintenance, the system will be unavailable during the maintenance.
- Due to system upgrade, the system will be unavailable during the upgrade.
- Due to the server outage, the system will be unavailable until the server provider fix the problem.

4. Tech. Stack Decisions

4.1. Front-End Design Decision: Flutter, Dart, HTML, CSS, JavaScript

4.1.1. Flutter, Dart (programming language that is used in Flutter)

- First reason for choosing Flutter framework and Dart language for implementing the project’s front-end of mobile application is Flutter framework is around 90% compatible with Android and IOS. With the help of this compatibility, it isn’t necessary to write different codes for Android and IOS. [1]
- Second reason for choosing Flutter framework is Flutter has hot reload feature. With the help of that feature, developer can view the changes that he/she made in the code on emulators. [1]

4.1.2. HTML, CSS, JavaScript

- The reason for choosing HTML, CSS and JavaScript for implementing the project’s front-end of webpage is HTML is a simple language that everyone can easily learn and build a basic webpage on browsers. With the help of CSS, the webpage that is built with using HTML can become more decorative and user friendly. Lastly, with the help of JavaScript, developers can make this webpage dynamic such as adding events to buttons. [2]

4.2. Development and Cloud Decisions: Spring Boot, Spring Cloud

4.2.1. Spring Boot

- The reason for choosing Spring Boot in development stage of the project is Spring Boot reduces huge amount of time for development stage and it increases productivity. Spring Boot also provides a lot of plugins to develop and test Spring Boot Applications and again it has a lot of plugins to work with databases. It is easy to develop with Java. [3]

4.2.2. Spring Cloud

- The reason for choosing Spring Cloud in implementing the cloud stage of the project is Spring Cloud provides developers to log correlation and tracing features. Also Spring Cloud has “Spring Cloud Gateway” which is programmable router based on Project Reactor. The most important feature of Spring Cloud is it provides client side load balancer that makes load balancing easy. Spring Cloud provides an implementation for circuit breaker pattern. With the help of this circuit breaker pattern, microservice can continue operating when a related service fails. [4]

4.3. Database Decision: Cassandra

4.3.1. Cassandra

- First reason for choosing Cassandra for implementing the project’s database is Cassandra’s high availability and fault tolerance. The other reason is that Cassandra is open-source which means it is free and everyone can access it.

With the help of being open-source, topics and questions about Cassandra are very common in online community. [5]

4.4. Project Management Tools Decisions: Skype, Google Drive

4.4.1. Skype

- The reason for choosing Skype for communicating during the implementation of the project is ease of use. Skype is the most common tool for online video chat and voice calls.

4.4.2. Google Drive

- The reason for choosing Google Drive for sharing the project's milestones (front-end codes, reports etc.) is ease of use.

5. Trade-off Analysis

5.1. Performance vs. Scalability

- If we want to prevent to let the single user handle with slow system, we need to improve the performance of our project. A slow system under heavy load is a scalability problem that every developer can face with. Between these two problems, as a developer team, we need to choose the most important one and need to prevent the possibility that it occurs. However, these two problems related to each other, it is very hard to improve both at the same time. Since we are aiming to serve huge amount of users, choosing scalability is more logical for our project.

5.2. Latency vs. Throughput

- Latency means the required time to produce a result for a user. Throughput is the number of results that are produced per unit of time. If we want to choose between these two, we are going to choose maximizing throughput with acceptable latency. Since we are going to serve a huge amount of users, instead of decreasing latency and throughput at the same time, we need to increase throughput with acceptable latency.

6. Assumptions

- Short URLs are expected to be accessed 100000 per day.
- On average, 50000 URLs are transformed to short URLs per day.
- 7500 new user registrations are expected per day.

7. Estimations

7.1. Capacity Estimations

SLAs are one of the most important factors when we estimate the capacity of the system since they are quantifiable. Other important players for capacity estimations are our assumptions and our trade-off analysis. As we consider all of these, we reach a consensus on these:

10 Virtual Machine, 12 CPU Cores, 32 GB Memory and 120 GB Raw Storage Capacity, 100 Mbps Bandwidth

In addition, we also consider its financial side when we do these capacity estimations.

7.2. Cost Estimations

In the light of the capacity estimations, cost estimations should be also done as it is mentioned before. For cost estimation, AWS Total Cost of Ownership is very helpful for us. When we give our capacity estimations to the program, $154,982 for servers, $52,795 for storage, $68,716 for network, $1,620 for IT-labor and total of them is $278,112 for on-prem. These are 3 years cost breakdown.

8. Testing

Acceptance Testing: To make sure we build the right system and it meets with the users' expectations.

Scalability Testing: Since we choose scalability over performance, we need to do scalability testing to system requires the capability to provide scale up and scale down facilities when they are needed.
Availability Testing: During users’ mission-critical activities, system must be available and users should not be faced with undesired results. That is why availability of the system should be tested.
Disaster Recovery Testing: Our system must has high ratio of availability like it was mentioned at Availability Testing and if some problems occur such as network outage, breakdown due to extreme load and system failures, how fast the failure indicted and if there is any loss of data should be observed.

References

[1] “What is Flutter? Benefits and limitations”. https://blog.codemagic.io/what-is-flutter-benefits-and-limitations/. [Accessed: March 24, 2020]

[2] “What are HTML, HTML5, CSS, and JavaScript? What are their advantages?”. https://www.quora.com/What-are-HTML-HTML5-CSS-and-JavaScript-What-are-their-advantages . [Accessed: March 24, 2020]

[3] “Spring Boot Tutorial”. https://www.journaldev.com/7969/spring-boot-tutorial . [Accessed: March 25, 2020]

[4] “Spring Cloud and its advantages”. https://www.javacodemonk.com/spring-cloud-and-its-advantages-3ac60b2c . [Accessed: March 25, 2020]

[5] “What are advantages and disadvantages of Cassandra database?”. https://www.quora.com/What-are-advantages-and-disadvantages-of-Cassandra-database-You-know-I-need-these-parameters-Distribution-Replication-Object-Oriented-XML-and-unique-properties-of-it . [Accessed: March 26, 2020]

[6] “Getting Started With Cloud Testing,” https://www.softwaretestinghelp.com/getting-started-with-cloud-testing/. [Accessed: 02-Apr-2020].

[7] “How the Cost of Cloud Computing is Calculated,” *Expedient*, 01-May-2015. [Online]. Available: https://www.expedient.com/knowledgebase/blog/2015-05-01-how-the-cost-of-cloud-computing-is-calculated/. [Accessed: 02-Apr-2020].

[8] “8 Factors to Consider in Cloud Computing Capacity Planning” https://www.sungardas.com/globalassets/_multimedia/document-file/sungardas-8-factors-to-consider-in-cloud-computing-capacity-planning-white-paper.pdf [Accessed: 02-Apr-2020].