Data Center Project Final Write-Up

Title : Document Indexing and search in an E-learning platform

Project participants : Pradyoth Srinivasan, Vandana Sridhar (5253 - 001)

Detailed Project Overview and Goals:

This system helps users search and access resources such as e-books and other documents and additionally helps upload and store resource materials. Given a user query, this application helps in finding the appropriate and relevant resources.

For this project, we ended up utilizing five datacenter components and learnt their interactions and interconnections. We utilized REST APIs, message queues, key-value stores, Cloud storage services and virtual machines to construct our application. Additionally we incorporated the Elasticsearch service to a major portion of our application to facilitate our search and indexing platform.

Detail Description of the components:

1. **REST API interface:**

    a. **Purpose** - To create our user interface and accept user requests through API endpoints. We have 3 endpoints namely index, search and getURL. The index endpoint allows users to send in documents through our interface and this sends it to our workers for it to index documents through elasticsearch. Our search endpoint takes in search queries from the user to lookup relevant documents. Finally our getURL endpoint sends the URL of the relevant document back to the user for them to open.

    b. **Why select this component - Advantages and Disadvantages**
        i. **Advantages:** It provides an easy interface for exchanging information between the client and server. The REST platform provides a clear demarcation between the client and the server programs. Throughout the course of this project, REST has been easy to adapt to and write code in.
        ii. **Disadvantages:** REST took more time to process data intensive requests. REST encounters latency issues since the latency time depends on the number of hops it takes the request to travel from the client to server and vice versa. REST requests slow down when the server is deployed in a different region.

c. **Interactions with other components:** The REST client interacts with the REST server by sending requests to it. The client either sends documents to be indexed or a search phrase to obtain the relevant documents through PUT and GET requests respectively. The server consists of endpoints to facilitate these client requests and the server acknowledges these by providing a response to the client and forwards the task of processing the request data to the worker via the rabbitMQ messages queue. The server additionally uploads the documents to the storage services on the google cloud.

2. **Messaging queues: RabbitMQ**

a. **Purpose :** We utilized the RabbitMQ messaging queue to place tasks sent from the server onto the queue for the available workers to consume. Similarly we used a queue to send the response URL back from the worker to the server's endpoint and the client takes the result through this endpoint accordingly. Finally we used the RabbitMQ exchanges to set up a logging channel between services in order to understand their interactions. Error/ Debug messages were logged as well.

b. Why select this component - Advantages and Disadvantages:

i. **Advantages:** Message queues provide communication between our different services and makes it easier for coordination between these distributed services. They also facilitate better performance and enable us to scale our application to handle heavy traffic.

ii. **Disadvantages:** When a rabbitmq channel starts to consume requests all of the functions that are supposed to be invoked after it are rendered useless. It is also harder to customize and change the configuration files.

c. **Interactions with other components:** the RabbitMQ messaging queue is set up in the server and it interacts with the worker when tasks are placed on the queue. On the worker side, the receiver queue is established to consume the requests. Similarly another queue on the worker sends a response to the client.

3. **Key-value store - Redis**

a. **Purpose:** We used Redis in our project to store the document's hash ID along with its name. The lookup service retrieves the name of the document from redis through the hash ID.

b. Why select this component - Advantages and Disadvantages:

i. **Advantages:** It stores the data in key-value pairs as large as 512 MB. Hence data can be easily retrieved through keys. It is an in-memory database so data retrieval is much faster than any relational database. It caches large amounts of data easily.

ii. **Disadvantages:** One cannot store proper relational data in Redis since it doesn't define proper schemas.

c. **Interactions with other components:** the Redis database interacts with the worker. The worker queries it with the hash ID and gets back the document name.

4. **Cloud storage services - Google Cloud Storage:**

    a. **Purpose:** We used google cloud storage to store the documents uploaded by the user along with their contents. Additionally the documents in google cloud storage is accessed by the lookup service and the public URL of the document is sent back to the user
    b. **Why select this component - Advantages and Disadvantages:**
        i. **Advantages:** It is a robust service included as a part of the google cloud platform. It provides infrastructure that helps us store data in a secure and user friendly environment. It is easy to manipulate and access. It provides massive storage capacity.
        ii. **Disadvantages:** Files are automatically referenced as private and it takes significant configuration changes to make files accessible to remote users. Having slow internet connection and latency issues can hinder users from viewing and accessing files on google cloud storage.
    c. **Interactions with other components:** Google cloud storage is accessed through server to upload documents sent by the user. The lookup service in the worker accesses the required file in the cloud storage and sends back the public URL to the user.

5. **Virtual Machines - Google compute engine:**

    a. **Purpose:** All our services are hosted on virtual machines to create to a distributed environment for interactions.
    b. **Why select this component - Advantages and Disadvantages:**
        i. **Advantages:** Virtual machine provide easy creation, easy maintenance. They are always available and they provide convenient recovery. Promotes easy rollback in changes.
        ii. **Disadvantages:** They are less efficient than real machines since they don't access computer hardware resources directly. Usability is significantly slowed down because of this. Un-familiar users might find it hard to configure their networking and hardware features.
    c. **Interactions with other components:** Services are deployed in the virtual machines and VMs are SSH'ed into for the client,server and worker to interact with each other.

Explains the capabilities and limits of your final solution:

Capabilities:

- Our system can give users the ability to upload multiple documents concurrently.
- It allows users to upload text files and also give users the ability to search for relevant resources using a free flowing query
- This system utilises Elasticsearch to efficiently index and search documents.
- The documents that are retrieved are also ranked based on the relevancy of the document with respect to the query
- The user is given the power to open up the contents of the document by clicking on the public URL link we provide
- Since we have multiple workers, we can handle significant work loads upto 1000 concurrent files.

Limitations:

- Since we didn't deploy our system to Kubernetes, our systems can't autoscale depending on the workload.
- All the virtual machines have to be running for the application to work.
- The application doesn't handle all file formats.