Requirements of a Distributed Search System's Design

Let's identify the requirements of a distributed search system and outline the resources we need.

We'll cover the following

Requirements

Let’s understand the functional and non-functional requirements of a distributed search system.

Functional requirements

The following is a functional requirement of a distributed search system:

- **Search**: Users should get relevant content based on their search queries.

The functional requirement of a distributed search system

Non-functional requirements

Here are the non-functional requirements of a distributed search system:

- **Availability**: The system should be highly available to the users.
- **Scalability**: The system should have the ability to scale with the increasing amount of data. In other words, it should be able to index a large amount of data.
- **Fast search on big data**: The user should get the results quickly, no matter how much content they are searching.
- **Reduced cost**: The overall cost of building a search system should be less.

The non-functional requirement of a distributed search system

Resource estimation

Let's estimate the total number of servers, storage, and bandwidth that is required by the distributed search system. We'll calculate these numbers using an example of a YouTube search.

Number of servers estimation

To estimate the number of servers, we need to know how many daily active users per day are using the search feature on YouTube and how many requests per second our single server can handle. We assume the following numbers:

- The number of daily active users who use the search feature is three million.
- The number of requests a single server can handle is 1,000.

The number of servers required is calculated using this formula:

$$\frac{Number\ of\ active\ users}{queries\ handled\ per\ server} = 3K\ servers$$

If three million users are searching concurrently, three million search requests are being generated at one time. A single server handles 1,000 requests at a time. Dividing three million by 1,000 gives us 3,000 servers.

The number of servers required for the YouTube search service

Storage estimation

Each video's metadata is stored in a separate JSON document. Each document is uniquely identified by the video ID. This metadata contains the title of the video, its description, the channel name, and a transcript. We assume the following numbers for estimating the storage required to index one video:

- The size of a single JSON document is 200 KB.
- The number of unique terms or keys extracted from a single JSON document is 1,000.
- The amount of storage space required to add one term into the index table is 100 Bytes.

The following formula is used to compute the storage required to index one video:

$$Total_{storage/video} = Storage_{/doc} + (Terms_{/doc} \times Storage_{/term})$$

Total Storage Required to Index One Video on YouTube

Storage per JSON doc (KB)	No. of terms per doc	Storage per term (Bytes)	Total storage per video (KB)
200	1000	100	300

In the table above, we calculate the storage required to index one video. We have already seen that the total storage required per video is 300 KB. Assuming that, on average, the number of videos uploaded per day on YouTube is 6,000, let's calculate the total storage required to index the videos uploaded per day. The following formula is used to compute the storage required to index the videos uploaded to YouTube in one day:

$$Total_{storage/day} = No.\ of\ videos_{/day} \times Total_{storage/video}$$

Total Storage Required to Index Videos per Day on YouTube

No. of videos per day	Total storage per video (KB)	Total storage per day(GB)
6000	300	1.8

The total storage required to index 6,000 videos uploaded per day on YouTube is 1.8 GB. This storage requirement is just an estimation for YouTube. The storage need will increase if we provide a distributed search system as a service to multiple tenants.

Summarizing the storage requirement of a distributed search system for videos uploaded to YouTube per day

Bandwidth estimation

The data is transferred between the user and the server on each search request. We estimate the bandwidth required for the incoming traffic on the server and the outgoing traffic from the server. Here is the formula to calculate the required bandwidth:

$$Total_{bandwidth} = Total_{requests\_second} \times Total_{query\_size}$$

Incoming traffic

To estimate the incoming traffic bandwidth, we assume the following numbers:

- The number of search requests per day is 150 million.
- The search query size is 100 Bytes.

We can use the formula given above to calculate the bandwidth required for the incoming traffic.

Bandwidth Required for Incoming Search Queries per Second

No. of requests per second	Query size (Bytes)	Bandwidth (Mb/s)
1736.11	100	*f* 1.39

Outgoing traffic

Outgoing traffic is the response that the server returns to the user on the search request. We assume that the number of suggested videos against a search query is 80, and one suggestion is of the size 50 Bytes. Suggestions consist of an ordered list of the video IDs.

To estimate the outgoing traffic bandwidth, we assume the following numbers:

- The number of search requests per day is 150 million.
- The response size is 4,000 Bytes.

We can use the same formula to calculate the bandwidth required for the outgoing traffic.

Bandwidth Required for Outgoing Traffic per Second

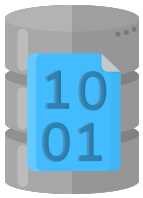
No. of requests per second	Query size (Bytes)	Bandwidth (Mb/s)
1736.11	4000	*f* 55.56

Summarizing the bandwidth requirements of a video search

> **Note:** The bandwidth requirements are relatively modest because we are assuming text results. Many search services can return small thumbnails and other media to enhance the search page. The bandwidth needs per page are intentionally low so that the service can provide near real-time results to the client.

Building blocks we will use

We need a distributed storage in our design. Therefore, we can use the blob store, a previously discussed building block, to store the data to be indexed and the index itself. We'll use a generic term, that is, "distributed storage" instead of the specific term "blob store."



Distributed storage: Blob store

To conclude, we explained what the search system's requirements are. We made resource estimations. And lastly, we mentioned the building block that we'll use in our design of a distributed search system.