

Requirements of a Blob Store's Design

Identify the requirements of and make estimations for the blob store.

We'll cover the following

- Requirements
 - Functional requirements
 - Non-functional requirements
- Resource estimation
 - Number of servers estimation
 - Storage estimation
 - Bandwidth estimation
- Building blocks we will use

Requirements

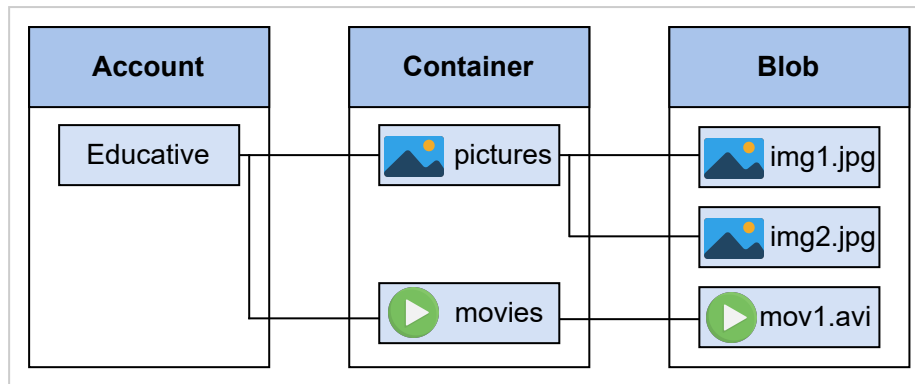
Let's understand the functional and non-functional requirements below:

Functional requirements

Here are the functional requirements of the design of a blob store:

- **Create a container:** The users should be able to create containers in order to group blobs. For example, if an application wants to store user-specific data, it should be able to store blobs for different user accounts in different containers. Additionally, a user may want to group video blobs and separate them from a group of image blobs. A single blob store user can create many containers, and each container can have

many blobs, as shown in the following illustration. For the sake of simplicity, we assume that we can't create a container inside a container.



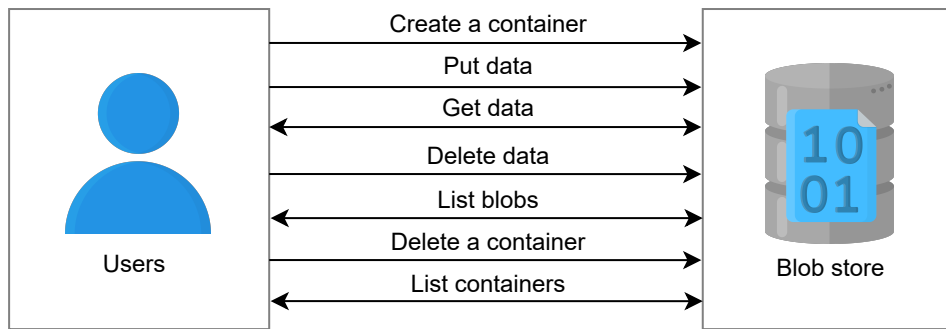
Multiple containers associated with a single storage account, and multiple blobs inside a single container

- **Put data:** The blob store should allow users to upload blobs to the created containers.
- **Get data:** The system should generate a URL for the uploaded blob, so that the user can access that blob later through this URL.
- **Delete data:** The users should be able to delete a blob. If the user wants to keep the data for a specified period of time (retention time), our system should support this functionality.
- **List blobs:** The user should be able to get a list of blobs inside a specific container.
- **Delete a container:** The users should be able to delete a container and all the blobs inside it.
- **List containers:** The system should allow the users to list all the containers under a specific account.

?

Tt

☾

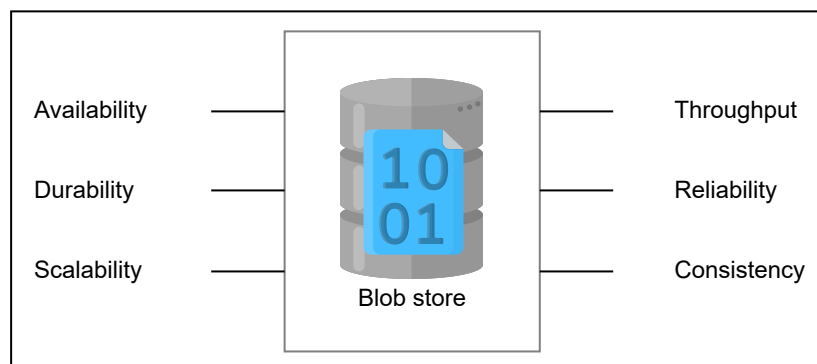


Functional requirements of a blob store

Non-functional requirements

Here are the non-functional requirements of a blob store system:

- **Availability:** Our system should be highly available.
- **Durability:** The data, once uploaded, shouldn't be lost unless users explicitly delete that data.
- **Scalability:** The system should be capable of handling billions of blobs.
- **Throughput:** For transferring gigabytes of data, we should ensure a high data throughput.
- **Reliability:** Since failures are a norm in distributed systems, our design should detect and recover from failures promptly.
- **Consistency:** The system should be strongly consistent. Different users should see the same view of a blob.



The non-functional requirements of a blob store

Resource estimation

Let's estimate the total number of servers, storage, and bandwidth required by a blob storage system. Because blobs can have all sorts of data, mentioning all of those types of data in our estimation may not be practical.

> Therefore, we'll use YouTube as an example, which stores videos and thumbnails on the blob store. Furthermore, we'll make the following assumptions to complete our estimations.

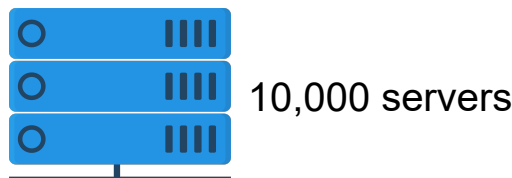
Assumptions:

- The number of daily active users who upload or watch videos is five million.
- The number of requests per second that a single blob store server can handle is 500.
- The average size of a video is 50 MB.
- The average size of a thumbnail is 20 KB.
- The number of videos uploaded per day is 250,000.
- The number of read requests by a single user per day is 20.

Number of servers estimation

From our assumptions, we use the number of daily active users (DAUs) and queries a blob store server can handle per second. The number of servers that we require is calculated using the formula given below:

$$\frac{\text{Number of active users}}{\text{Queries handled per server}} = 10K \text{ servers}$$



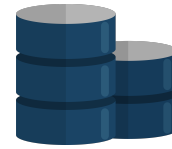
Number of servers required by a blob store system dedicated to storing YouTube data

Storage estimation

Considering the assumptions written above, we use the formula given below to compute the total storage required by YouTube in one day:

$$Total_{storage/day} = No. of videos_{/day} \times (Storage_{/video} + Storage_{/thumbnail})$$

Putting the numbers from above into the formula gives us $12.51 TB_{/day}$, which is the approximate storage required by YouTube per day for keeping a single copy of the uploaded video in a single resolution.



= 12.51 TB/day

Total Storage Required to Store Videos and Thumbnails Uploaded Per Day on YouTube

No. of videos per day	Storage per video (MB)	Storage per thumbnail (KB)	Total storage per day (TB)
250000	50	20	<i>f</i> 12.51

Bandwidth estimation

Let's estimate the bandwidth required for uploading data to and retrieving data from the blob store.

Incoming traffic: To estimate the bandwidth required for incoming traffic, we consider the total data uploaded per day, which indirectly means the total storage needed per day that we calculated above. The amount of data transferred to the servers per second can be computed using the following formula:

?

Tt

☾

$$Total_{bandwidth} = \frac{Total_{storage_day}}{24 \times 60 \times 60}$$

>

Bandwidth Required for Uploading Videos on YouTube

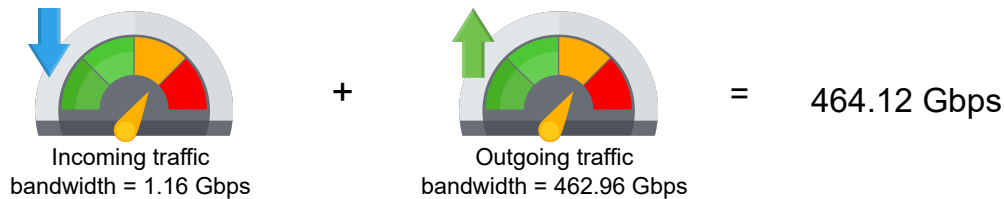
Total storage per day (TB)	Seconds in a day	Bandwidth (Gb/s)
12.51	86400	<i>f</i> 1.16

Outgoing traffic: Since the blob store is a read-intensive store, most of the bandwidth is required for outgoing traffic. Considering the aforementioned assumptions, we calculate the bandwidth required for outgoing traffic using the following formula:

$$Total_{bandwidth} = \frac{No. \text{ of active users}_{/day} \times No. \text{ of requests}_{/user/day} \times Total_{data_size}}{Seconds \text{ in a day}}$$

Bandwidth Required for Downloading Videos on YouTube

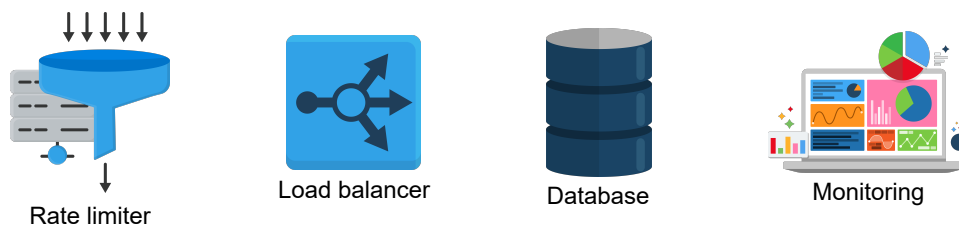
No. of active users per day	No. of requests per user	Data size (MB)	Bandwidth required (Gb/s)
5000000	20	50	<i>f</i> 462.96



Summarizing the bandwidth requirements of a blob store system for YouTube videos only

Building blocks we will use

We use the following building blocks in the design of our blob store system:



Building blocks for the design of a task scheduler

- **Rate Limiter:** A rate limiter is required to control the users' interaction with the system.
- **Load balancer:** A load balancer is needed to distribute the request load onto different servers.
- **Database:** A database is used to store metadata information for the blobs.
- **Monitoring:** Monitoring is needed to inspect storage devices and the space available on them in order to add storage on time if needed.

In this lesson, we discussed the requirements and estimations of the blob store system. We'll design the blob store system in the next lesson, all while following the delineated requirements.

← Back

Next →

☒ Mark as Completed

