ID2221 Data Intensive Computing - Review Questions 1 - Group 19

by Artem Sliusarenko and Eugen Lucchiari Hartz

1.) Explain how a file region can be left in an undefined state in GFS?

In Google File System (GFS) operations that modify the file system, such as creating or editing a file, are known as namespace mutations. These operations are designed to be atomic which means that they either complete successfully or not at all, ensuring that changes are made in a consistent and reliable manner.
When a mutation occurs, the state of the affected file region can be categorised as consistent, defined or undefined. A consistent state means that the data in that region is identical across all replicas, which allows the client to read from any of them without issue. A defined state is a more specific case of consistency, where not only the data is consistent across all replicas, but it also fully reflects the intended changes of the mutation.

However, a region can enter an undefined state if something goes wrong during the mutation process. For example, if two mutations happen at the same time, the data may become a mix of both changes, which leads to inconsistencies. Similarly, if a mutation fails the data in different replicas might end up being different which causes the region to be inconsistent and therefore in an undefined state.

2.) Briefly explain the read operation in GFS? What's the role of the primary node?

When a client wants to read a file or a specific part of it in GFS, the client first identifies which chunks of the file are needed. The client then contacts the GFS master to obtain the locations of the chunk servers that store these chunks. The master responds with the information about the chunk replicas. The client typically caches this information to avoid redundant queries to the master. The client then directly reads from one of the chunk servers. If the selected chunk server fails, the client can switch to another server on the list, which ensures continued access to the data and prevents failure. The chunk server then sends the requested data to the client after each request.

The primary node in GFS has several key responsibilities. It acts as the lease holder for specific chunks, with the GFS master that assigns these leases. The primary node is responsible for determining the order in which operations, such as mutations, are executed, which ensures that all replicas remain consistent. Additionally, it manages mutation propagation, which makes sure that any changes to the chunk are replicated to all secondary replicas. The primary node also does error handling by communicating the status of mutations back to the client so that the mutation error can get handled by the client.

3.) Using one example show that in the CAP theorem, if we have Consistency and Partition Tolerance, we cannot provide Availability at the same time.

The CAP theorem states that a distributed system can simultaneously provide at most two of the following three guarantees: consistency, availability, and partition tolerance. Consistency means that any read operation will always return the most recent write, ensuring the data is always up to date. Partition tolerance refers to the system's ability to continue functioning even if communication is lost between parts of the system which means the system can still operate despite some links failing and certain parts being unable to communicate with each other.

As an example you could have a distributed online shopping platform where product inventory is stored across multiple servers in different regions. If a customer purchases the last item of a product, the system needs to update the inventory across all servers to reflect that the item is out of stock. Now, suppose there is a network partition so that some servers cannot communicate with others due to a network issue. To maintain consistency the system might choose to delay confirming the purchase until all servers can be updated with the out-of-stock status. However, during this delay, the system becomes unavailable for processing new purchases of that item until the partition is resolved. This ensures that the inventory is consistent across all servers and that the system can tolerate the partition, but it sacrifices availability because the system cannot serve new requests during the partition.

4.) Explain how the consistent hashing works.

Consistent hashing involves mapping both data and nodes to the hash space. The entire hash space is considered to be a circle (hash ring). Hash function is used to assign a position (a hash value) to both nodes (hashing nodes identifier - usually IP address) and keys (usually a file name) on the circle. After the key is hashed it is stored on the first node that appears on the ring in the clockwise direction from the key's hash value.This helps to delegate the responsibility for a certain range of keys to a certain node. For example if we have 3 nodes: A, B, C. Using hash function they are placed at positions 10, 30 and 70. Now we have a key "file.txt" and its hash is 40. Since node 70 is the first node clockwise from 40 this file will be stored at position 70 (node C).

5.) Explain the finding process of tablets in BigTable.

BigTable, developed by Google, is a distributed storage system designed to manage huge amounts of data efficiently. The key concept in BigTable is dividing data into smaller units called tablets and providing an efficient method to locate these tablets.

In BigTable the data is stored in Tablet(s). Tablets are managed by Tablet servers. Root Tablet contains the location (basically a tablet server address) of all Tablets, in the metadata

table, that contains the data we need. Chubby is responsible for keeping track of live Tablet servers and it stores the location of the root Tablet.
To discover the data Chubby accesses the root Tablet, retrieves the location of all Tablets that store the data from the metadata table and provides that information to the client. Clients can cache the locations for future use.