Scott Viet Phong Nguyen, 350313

Anh Duc Nguyen, 379744

Task 1 – CAP Theorem (20+20 = 40%)

a) Shortly explain the CAP theorem by example of the Domain Name System (DNS).

The CAP theorem says that it is impossible to ensure all three factors (Consistency, availability and partition tolerance) at the same time in a distributed system.

- Consistency: In distributed systems all replicas of the manipulated data has to be up to date.
- Availability: Acceptable response (no error or timeout)
- Partition tolerance: the failure tolerance of server networks, even if messages get lost or partitions of the network don't work

It's only possible for maximum of 2 factors at the same time in a distributed system.

The Domain Name System can be categorized in the two factors Consistency and Availability, but not in the partition tolerance. DNS has to have a very high availability for a large amount of requests that needs to be processed and a high failure tolerance if some individual DNS-server goes down. But the consistency of the data is not guaranteed everytime. It is possible that changes of an DNS-entry can take days to have an effect on the DNS-hierarchy and then it can be seen by the clients.

---

b) Shortly describe two dimensions of data consistency from both a data-centric and a client-centric perspective.

Data consistency is the correctness of saved data in databases.

Data-centric: Describes the guarantees of consistency from the intern point of view, the storage provider perspective.

Two dimensions:

Ordering describes

- The execution order of requests on different replicas

Staleness describes

- The difference in time between the commit of an update (if dirty reads are possible this changes to the start of an update) and reaching the last replica

Client-centric: Describes the data consistency from the view of the client/application.

Two dimensions:

Ordering describes

- The ordering of requests visible to clients

Staleness describes

- The difference in time between the commit of an update (of dirty reads are possible this changes to the start of an update) and the last point in time when the previous value was still returned

---

Task 2 – Dynamo (10+10+20+10+10 = 60%)

a) Which queries does Dynamo support and for which type of data storage is Dynamo optimized?

DynamoDB supports GET/PUT operations using a user-defined primary key (Can either be a single-attribute partition key or a composite partition-sort key). Additionaly you can query on non-primary key attributes using Global Secondary Indexes and Local Secondary Indexes.

Smaller data elements or file pointers are best saved in DynamoDB, because Amazon Dynamo DB stores structured data, indexed by primary key, and allows low latency read and write access to items ranging from 1 byte up to 400KB.

---

b) Pessimistic replication (as implemented in Dynamo, for example) is used to offer high availability and low latency. True or false?

False, because pessimistic replication only performs well in local-area networking environments and its faces a trade-off regarding availability (while increasing the number of replicas improves read availability, it decreases write availability).

---

c) Dynamo uses vector clocks to determine the total order of write operations.

Given the vector clocks in the table below with conflicting versions on servers A, B, and C. Please state whether or not the conflict can be reconciled automatically (yes/no) and how the vector clock must look like after a conflict resolution.

Vector clocks before conflict resolution	Can be reconciled automatically (yes/no)	Vector clocks after conflict resolution
D1 ([A,1]) D2 ([B,1]) D3 ([C,1])	yes	([A1],[B1],[C1])
D1 ([A,1] [B,2]) D2 ([A,2] [B,2])	yes	([A,2][B,2])
D1 ([A,1] [B,4] [C,13]) D2 ([A,2] [B,3] [C,15]) D3 ([A,2] [B,4] [C,15]) D4 [A,1] [B,3] [C,14])	yes	([A,2][B,4][C,15])
D1 ([A,1] [B,2] [C,1]) D2 ([A,2] [B,2])	yes	([A,2][B,2][C1])

---

d) Shortly explain the trade-off between consistency, read latency, and write latency in Dynamo and how a Dynamo-based application could be tuned either towards fast reads or towards fast writes using the (N,R,W) configuration.

If W+R > N, then the write set and the read set always overlap and one can guarantee strong consistency. In the primary-backup RDBMS scenario, which implements synchronous replication, N=2, W=2, and R=1. No matter from which replica the client reads, it will always get a consistent answer. In asynchronous replication with reading from the backup enabled, N=2, W=1, and R=1. In this case R+W=N, and consistency cannot be guaranteed.

In R=1 and N=W we optimize for the read case, and in W=1 and R=N we optimize for a very fast write.

---

e) What is the minimum cluster size, i.e., number of servers, of a Dynamo configuration (N=9, R=1, W=9) and why? For this minimum cluster size: how many data records are stored on each node after 1 million data records have been inserted into the Dynamo cluster by a client program?

With the configuration (N=9, R=1, W=9) the minimum cluster size is 9. To maintain consistency among its replicas, Dynamo uses a consistency protocol similar to those used in quorum systems. This protocol has two key configurable values: R and W. R is the minimum number of nodes that must participate in a successful read operation. W is the minimum number of nodes that must participate in a successful write operation. Setting R and W such that R + W > N yields a quorum-like system.

For this minimum cluster size there are about 111112 data records in each node, inserted by a client program.