Data Intensive Computing - Review Questions 1

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

   In GFS, failed mutations (writes or record appends) at any of the replicas lead to chunks that are undefined and inconsistent. The main sources of inconsistency are concurrency, machine failures or network partitions. Due to this inconsistent state of the file region, different clients may see different data at different times.

2. Briefly explain how HopsFS uses User Defined Partitioning (UDP) and Distributed Aware Transaction (DAT) to improve the system performance (e.g., CPU usage, network traffic and latency)?

   With the aim of improving the performance of the file system operations, HopFS uses User Defined Partitioning to distribute data across different database nodes: the namespace is partitioned such that the metadata for all immediate descendants of a directory reside on the same database shard for efficient directory listing. This allows Partition Pruned Index Scans, meaning that scan operations are localized to a single database shard. Moreover, HopFS uses Distributed Aware Transaction to choose which Transaction Coordinator handles which file systems operations: the transaction is started on the database shard that stores all/most of the metadata required for the current file system operation.

3. Show with an example that in the CAP theorem, we can have only two properties out of Consistency, Availability, and Partition Tolerance at the same time.

   Let's make a *Reductio ad absurdum*: let's assume that we have a system that assures property of Consistency, Availability and Network Partition. This means that if I read from any node, I should be able to receive the most recent value (Consistency) with no waiting time (Availability) and errors (Network partition). However, assume that I'm performing a Write to a node X1 and a subsequent read to a node X2 for the same resource. Since the network is partitioned, I won't receive immediately the most recent value at node X2, being the network partitioned. That means that the aforementioned system is not consistent.

4. How does BigTable provide strong consistency?

   BigTable provides strong consistency for mainly two reasons:

   (a) Chubby service consists of five active replicas, but only one is elected as Master and manages the changes. To manage the failures, BigTable uses Paxos algorithm. **TODO rivedi** Only one tablet server is responsible for a given piece of data.

   (b) The underlying distributed file system, GFS, is in charge of the replicas, assuring the consistency.

5. Write a simple query in Cypher for each of the following requests:

   - Match a Person called "John Doe"

     MATCH (n:Person {name: 'John Doe'})
     RETURN n;

   - Find FRIENDS_OF "John Doe"

     MATCH (:Person {name:'John Doe'})-[:FRIEND_OF*]->(p:Person)
     RETURN p.name;

   - Count "John Doe" 's direct acquaintances

     MATCH (:Person {name:'John Doe'})-[:FRIEND_OF*1]->(p:Person)
     WITH count (p) AS directAcquaintancesCount
     RETURN directAcquaintancesCount;