Assignment 4

1.

Type	R	A	B
Records	1 000 000	1 000 000	10 000 000
Record length		400B	600B
Fetched B	600B	300B	300B
Volume MB	600MB	400MB	6000MB
NettoVolume MB		300MB	3000MB

Without filter:

n = ceil(VA/M) = ceil(300/20) = 15

VJnl = VA + nVB + VR

VJnl = 400 * (15 * 6000) + 600 = 91000 MB = 91GB

With filter:

A = Volume of A as MB

A' = Netto Volume of A as MB

Read A = 400MB, Write A' = 300MB

Read B = 6000MB, Write part B' (1/3) = 1000MB

Read A' and B' = 1300MB, Write A * B = 2000MB

VR = 600MB

Total = 400 + 300 + 6000 + 1000 + 1300 + 2000 + 600 = 11600MB

2.

a)

In the compendium by K. Bratsbergengen, i cannot find any elaboration on the vertical fragmentation topic. I will therefore only elaborate about the three location selection mechanisms mentioned for horizontial fragmentation:

- Value ranges: A node is given a value range spesific to the node, which it is responsible for.
- Round Robin: Records in all nodes are placed in a circular manner.
- Hashing: Records are places in an address based on a hash algorithm. The primary key is used for

input in the hash algorithm.

b)

Hashing is a good method due to its properties. Hashing will in most cases make lookup go fast and distribute the data evenly. It also easily allows for parallel execution. Hashing will also not require any indexing.

3.

Consistent hashing

Dynamo has a consept of Node which data can be assigned to, and the nodes is organised in a ring. Data is assigned to each nod by hashing the key of the data item and using the hash as a position on the ring. Every node is responsible for all the data items on the ring between itself and its predecessor. Dynamo also creates virtual nodes to handle the possibility of difference in node performance or to handle the possibility of loss of nodes. Each ring has a lot of virtual nodes, and each virtual node is assigned/controlled by a physical node. When a node goes down, the virtual nodes can be redistribued of the available node, and if more nodes is added, virtual nodes can be offloaded by redistribution again.

Since data needs to be redundant. A node coordinates its own data and replicates it to the next N-1 nodes in the ring.

Vector clocks

Vector clocks is used for versioning objects in Dynamo. The vector clock is a counter which is paired with each node in the database.

Sloppy quorum

To implement sloppy quorum, read and writes are done to the first N healthy nodes. The healty nodes must not be next in the ring. Sometimes the node that should be the master node of the replica is down, and it will be saved to another node. If so, it the replica will contain information about where it should reside so it can be moved at a later time.

Merkle trees

A Merkel tree is a special kind of tree where each node is a hash of the values to individual keys. Dynamo maintains a Merkel tree on each node for each of the nodes virtual nodes. The synchronizatin of two nodes require the nodes to start by exhangeing the root of the Merkel tree for the range of keys that they have in common.

Gossip-based membership protocol

To ensure that each node has each node has a persisted membership change history, the gossip-based membership protocol is implemented. The protocol makes each node contact a randomly selected peer every second.

4.

a)

How does RamClouds ensure Â«durabilityÂ» of data?

RamCloud is stored in DRAM which is fast, but volatile, which is a big case when we think about powerfailures and other outages. RamCloud counters this by using what the calles "buffered logging". This technology uses a copy of the data on multiple servers. To ensure presistancy, the data is written to one server, and then sent to other servers and temorarily stored in DRAM. When the other servers acknolegedes and confirms that the data is stored, the write operation is confirmed to the user.

b)

How does Ousterhout argue that RAMCloudâ€™s potential to support ACID transactions is better than for traditional disk-based distributed databases?

Ousterhout argues that RamCloud has better ACID potential than traditional disk-based storage because of the lower latency of DRAM. ACID's properties is mostly built for handling the possible problems of concurrent transactions. As DRAM is much faster than traditional storage, the risk of conflict is reduced.

5.

How does Facebook TAO solve the problem that the social graph spans the whole world, and that the data should be close to the user?

Facebook Tao solves the distributed storage problem of its social graph by splitting the data into hundres of thousands of "shards". A shard is a set of data that is contained in a logical database. A database server (MySQL) is responsible for one or more shards. Facebook runs TAO as a single-master per shard and rely on MySQL replication to propegate updates from a regional datacenter to all other regions, known as slave regions. A slave cannot update the persistant storage of a shard by itself, it will forward the request to the master for the spesific shard which will update and replicate it out to the slaves.

Typically a master for user data is choosen based on geographical data, so that the master is usually close to the origin of the request.

6.

Google Spanner uses timestamps for everything. It acutally stores the data on the format , which one can say is a deep and important integration.

First, the time implementation in Spanner, known as TrueTime (TT), is a clock that is fetched from GPS and Atomic clocks. The TT.now() function returns the current time, but in a slightly unconventional way. The current time is a interval between the earliest possible time and the latest possible time the current time can be, covering the possible inaccurecy of the time in to account.

The .now is complemented with .after and .before which returns True if the time given is guaranteed after or before, respectivly.

The user of the intervals will make spanner slow down to make sure that there is no local time conflict with the global time system and therefore provide a reliable timestamp for each transaction.