Big Data Analytics
Exam 2020-08-20
Grading Info for Questions 1–5 and 10–12

Olaf Hartig

Question 1 (1p)

Solution: We have to write application code that first issues a 'get(key)' query with the given user ID as the key. From the corresponding value that is retrieved as a result of this query, the application has to extract the array of IDs of the users liked by the given user. Thereafter, the application has to iterate over this array and, for each user ID in this array, another 'get(key)' query has to be issued. From the values retrieved by these queries, it is possible to extract the names of the users liked by the given user.

Remarks:

- There is no need to do a 'get(key)' *"for all user IDs"* or *"for every user ID"* or *"for all keys."*
- Just doing a 'get(key)' with the given user ID is not enough because it would allow us only to retrieve the IDs of the users liked by the given user, but not their names.

Question 2 (1p)

No example solution here because almost all of you got this one right.

Remark: Key-value stores are also schema-free. That is, the values of different key-values do not all have to have the same structure.

Question 3 (1p)

Also here, no example solution needed because almost all of you got this one right as well.

Remarks:

- A few of you wrote something about *"several nodes"* or *"nodes in a network."* That is not correct as an answer because it is related to scaling horizontally (scale out) rather than scaling vertically (scale up).
- Just saying what can be done to scale up (e.g., adding more RAM, bigger disks, etc) is not sufficient as an answer to the question.

Question 4 (1p)

Solution: The claim is *wrong*. In a master-slave system, changes to a database object are addressed to the node that is the master for the database object in question. This master node then requests all the corresponding slave nodes to change their copy of the database object accordingly.

Question 5 (1p)

Solution: Three nodes have to confirm the writes in this case. To achieve strong consistency for writes ("write consistency"), we need to have $2 \cdot W > N$. In this case, N=4. Therefore, the smallest value for W that satisfies the inequality $2 \cdot W > 4$ is W=3.

Question 10(a) (1p)

Example Solution: We may extend the existing key-value database by adding another array into each value such that this array contains the names of all the users liked by the current user. For example, the first key-value pair in the given database would be extended as follows:

`"alice_in_se"` → `"Alice, 1987, [bob95, charlie], [Bob Charlie]"`

(the other key-value pairs will have to be extended accordingly).

Given the so-extended database, it is now possible to implement our data retrieval request by issuing only a single 'get(key)' query, where the key is the given user ID. From the corresponding value that is retrieved as a result of this query we now can directly extract the names of the users liked by the given user.

Remarks:

- Many of you have tried something like I did in the related example solution for the previous exam. However, that solution does not help for the problem given in the current exam.
- In particular, some of you proposed to add additional key-value pairs where the keys are concatenations of a user ID and some prefix such as "LikedBy:", and the corresponding value is a copy of the array of the user IDs of the users liked by the user mentioned in the key (for instance, "LikedBy:selaya" -> "[charlie]"). However, this extension does not really help to make the data retrieval request more efficient because the number of 'get(key)' queries that would have to be issued to the database would still be the same as with the original database.

Question 10(b) (1p)

Solution: The advantage of extending the database as described above is that it allows us to implement the given data retrieval request more efficiently (by requiring few queries that need to be issued to the key-value store). The disadvantages are that, due to the added redundancy, the extended database requires more storage space, and updating the database without introducing inconsistencies becomes more difficult.

Remark: I was expecting you to explicitly mention both of the disadvantages: i) more storage space needed and ii) potential for introducing inconsistencies when updates are not done carefully. If your answer mentions only one of these two disadvantages, you cannot get the full point.

Question 11 (1p)

Remark: Most of your answer where correct or, at least, went into the right direction. So, the only thing I want to emphasize again is that read scalability is ***not*** just about being able to process a regularly high number of read operations, but to be able to deal with situations in which such a number increases. Hence, answers that are not explicit about the possibility that the typical number of read operations may increase do not get the full point.

Question 12 (1p)

Solution: The claim is *wrong* because only a limited number of keys need to be remapped. That is, when a node is removed, the only keys that are remapped are the ones whose hash value is in the range assigned to the removed node. Similarly, when a node is added, the only keys that are remapped are the ones whose hash value is in the range assigned to the newly added node.