CS/CE/TE 6378: Project III

Instructor: Ravi Prakash

Assigned on: April 11, 2018
Due date and time: April 26, 2018, 11:59 pm

This is a project to be completed in groups of two and you are required to demonstrate its operation to the instructor and/or the TA to get credit for the project. All group members have to be present and able to answer questions during the demonstration.

Only one group member should submit the assignment files. Make sure to include names of both group members in the README file.

1 Requirements

1. Source code must be in the C /C++ /Java programming language.
2. The program must run on UTD lab machines (`dc01, dc02, ..., dc45`).

2 File system with replicated chunks

This is an extension of the previous project. You can use your code from Project 2. The extensions are as follows:

1. Instead of three file servers, now you have five file servers named $S_1, S_2, \ldots, S_5$.
2. For each chunk, three replicas are maintained at servers $S_i,\ S_j$ and S_k, where $1 \leq i,\ j,\ k \leq 5$ and $i \neq j \neq k$.
3. When a new chunk is to be created, the M-server selects three chunk servers at random and asks each one of them to create a copy of the chunk.
4. For each chunk, the M-server maintains information about the chunk servers that host replicas of the chunk.
5. Any append to a chunk is performed to all live replicas of the chunk. Use two-phase commit protocol, with the appending client as the coordinator and the relevant chunk servers as cohorts, to perform writes. Assume there is no server failure during an append operation.
6. If multiple appends to the same chunk are submitted concurrently by different clients, the M-server determines the order in which the appends are performed on all the chunks. Once a client completes an append to a chunk it informs the M-server. Only then does the M-server permit another client to perform an append to the same chunk.
7. A read from a chunk can be performed on any one of the current replicas of the chunk.
8. A recovering chunk server may have missed appends to its chunks while it was down. In such situations, the recovering chunk server, with the help of the M-server, determines which of its chunks are out of date, and synchronizes those chunks with their latest version by copying the missing appends from one of the current replicas.
9. Only after all chunks of a recovering chunk server are up-to-date does that chunk server resume participating in append and read operations.

3 Submission Information

The submission should be through eLearning in the form of an archive consisting of:

1. File(s) containing the source code.
2. The README file, which describes how to run your program.

DO NOT submit unnecessary files.