CSCE–438 / Spring 2019 Homework 4

Homework 4: A Scalable and Highly Available SNS Service

130 pts

1 Overview

The objective of this assignment is to improve the scalability and availability of the SNS service developed in Homework 3.2. For this assignment the following must be considered:

1. The Tiny SNS Service functionalities and requirements that were provided in Homework #2 and #3.2 are still required for this assignment.
2. In Homework #3.2, there was one Available Server only. Now, however, every processes is an Available server.
3. When the client wants to connect to the SNS service, it contacts the Routing process. The Routing process returns to the client the IP address and port number for the Master Available server for this client.
4. Each client has a corresponding Master Available server and a Slave Available server.
5. A single Available server can serve as Master Available server for some clients and as Slave Available server for other clients.
6. If the Master Available server for a client crashes, the client should contact the Slave Available server for it.
7. When designing the Router server, consider that the number of clients connected to the each Available server needs to be balanced.
8. The data should be consistent between the Master and Slave Available server for each client. That is, when a client sends an update to its Master Available server, the data should be replicated to the Slave Available server as well.
9. You can not assume that the servers are time synchronized, i.e., we will modify the clocks of the servers.

10. You should not assume the communication (i.e., physical layer or the connection between servers) is reliable. We may disable the network interface from at most 1 server at a time. Your system should be able to recover.
11. You should have a robust system on heavy workloads. We will stress test your system by sending large amount of commands and postings (i.e., FOLLOW/UNFOLLOW/LIST/TIMELINE and post message)

2 What to Hand In

2.1 Design

Start with the your code for HW3.2. Before you start hacking away, write a design document. The result should be a system level design document, which you hand in along with the source code. Do not get carried away with it, but make sure it convinces the reader that you know how to attack the problem. List and describe the components of the system.

2.2 Source code

Hand in the source code, comprising of a makefile, source code files and startup scripts for starting your system on each of the 4 servers your system will run on.

The code should be easy to read (read: well-commented!). The instructors reserve the right to deduct points for code that they consider undecipherable.

2.3 Grading criteria

The 130 pts for this assignment are given as follows: 5pts for complete design document, 5pts for compilation, 120 pts for test cases (the test cases have different weights).