CSE M552
Project 4

Please note: this assignment is required for CSE M552 students, and optional for CSE 452 students. CSE 452 students who do this assignment will receive the maximum of their grade on this assignment, and their grade on the final. (In other words, by doing this assignment, the final becomes optional for CSE 452 students.) As with the earlier project assignments, this part of the project can be done in teams of 2-3.

This assignment is open-ended: add some significant functionality to your peer-to-peer Twitter application. Students should submit a one paragraph description of their proposed project by email to me by the project 2 due date.

Grading will be based on the correctness and efficiency of the design and implementation, as well as the degree of difficulty of the proposed work. Try to scope the topic to roughly the effort of the other three assignments.

Options include:

Add support for disconnected operation, e.g., with peer-to-peer merging in the absence of connectivity to the server cloud.

Add support for low bandwidth operation, for P2P twitter at home.

Add support for scalability, so that a very large number of users can post messages concurrently. For example, this may require multiple logical object stores for scalability, and therefore commit will need to work even if objects in the transaction are stored across multiple volumes.

Add support for nodes to enter and leave the system permanently; this would be needed for a practical peer-to-peer system. Note that the Paxos implementation in assignment 3 assumed that the group membership is fixed.

Add support in your system for distributed assertions – constraints that can be verified to be true at any instant in time, but which apply to state across the distributed system.

Add support for byzantine fault tolerance to your storage system, that is, to gracefully handle the case where an attacker, posing as a peer, tries to disrupt the system.

Other project of your choosing.