

Project Title: Gossip Protocol

Team Member: Jiahui Zhang

Team #: 3

Overview: A replication system with gossip protocol (anti-entropy approach) will be implemented on top of Client/Server code base.

Goals:

1. Communication among nodes where seed nodes can push new updates that eventually spread out to all other nodes
2. Resolve function in pull nodes that can be use by anti-entropy where discrepancy can be resolved
3. New nodes should be able to join the gossip and sync up with current data
4. **(Stretch)** Implement persistent data storage solution (i.e., SQLite) and distributed transactions

Design:

1. No failure is assumed but when dead nodes come back alive, they should be able to enter the gossip loop again like a new node
2. No workload is assumed but nodes should be able to handle both client orders and gossip bombing
3. No primary is enforced. Therefore, all Servers can receive and process orders from clients. Single record is marked with timestamp and id as primary key.

Evaluation (Current Focus on Correctness):

1. Are all nodes infected? Does anti-entropy make sure all infected nodes have all data eventually?
2. Could new node enter the loop of gossip and receive data? Could new node send gossip to other nodes?
3. Could pull nodes updates record correctly when receive multiple push messages at the same time (i.e. resolving multiple conflicting records)?

Timeline:

1. Nov.16: nodes should communicate to other nodes and receive clients' records
2. Nov.23: nodes should be able to send gossips
3. Nov.30: new nodes should be able to join the loop and anti-entropy should be able to resolve all discrepancies automatically
4. Dec+: SQLite