CompSci 512: Distributed System

Lab 3 (10/23/15)

Lecturer: Jeff Chase, TA: Max Demoulin Group Members: Junghoon Kang, Chaoren Liu

1 Transaction server application description

The TestHarness object first runs initializeStores to populate the KVStore actors. After initializing the KVStore actors, the LoadMaster actor fires out transactions to each TransactionService actors. Each TransactionService actor will randomly generate transactions. One transaction contains two key-value pairs and we name it Akey and Bkey. We can think of Akey and Bkey as two bank accounts. The transaction transfers some money from account Akey to account Bkey. In KVClient, we implement begin, voteOnAKey, Abort and submit. In begin, we acquire locks and get the value. To avoid the deadlock, we first grab the value with smaller key, and then grab the value of the larger key. In begin, we grab all the keys we need in current transaction.

To handle the key access conflict, we create a queue for each key on each KVStore actor. If one transaction asks for a key that is free, we just release the key and enqueue the client. If some transaction asks a key that is held by other transactions, we just enqueue the asker actor. When a key is release by one transaction after it commits, we check the queue of the key. If there are anyone is waiting for the key, we just send the value of the key to the first one in the queue.

To commit, we implement two phase commit. That is after finishing the calculations of one transaction, the client asks the KVStore actors which get involved in the transaction. The KVStore actor check the status and vote commit or Abort. If the client receive one Abort or the client cannot receive response in a time period, the client Abort the transaction. If all participant KVStore actors vote commit, the client writes to KVStores and release the locks. This implementation adopts the two-phase locking. In case of the Abort, we purge the cache on the client and the KVStore actors release the locks.

To test the robustness of the App when one client loses connection to KVStore actors for some time period, KVStore actors randomly ignore the commit request messages from some clients. After timeout (5 seconds), we just Abort the transaction and release the locks of the transactions. So when you run our program, you can notice some transactions fail and some transactions succeed.