**Midterm: Sharding**

*In this Midterm assignment, you will start with a standalone MongoDB database, turn it into a sharded cluster with two shards, and shard one of the collections. You will create a development environment on your local machine: no replica sets, and only one config server. In production you would always use three config servers and replica sets as part of a sharded cluster.*

***Remember to insert all commands directly into the document and MongoDB screenshots with the same command along with the results. Failure to comply with this deliverable requirement will result in NO credit being given!***

**Part A: Establish a sharding database server.**

Step 1: Create 2 subdirectories, one for each shard: ***\mongodb\shard1, shard2***

Step 2: Start a regular ***mongod*** process adding the option ***--shardsvr*** along with ***--dbpath shard1*** and unique ***--port #***

Step 3: Create a new directory called **configdb** in **\data**

Step 4: Start the mongod ***config server*** instance.

Step 5: Start the ***mongos*** instance.

Step 6: Start a ***mongo*** shell.

Step 7: Add the first shard ("localhost**: port#**").

Step 8: Verify shard status and then use the ***stock*** database.

Step 9a: Copy paste into the ***mongos>*** prompt (inside the mongo shell):

**o = {**

 **ticker : 'abcd',**

 **time : new Date(2012,2,3),**

 **price : 110,**

 **shares : 200,**

 **details : {**

 **asks : [ 110.07, 110.12, 110.30 ],**

 **bids : [ 109.90, 109.88, 109.70, 109.5 ],**

 **system : 'abc',**

 **lag : 0**

 **}}**

Step 9b: Copy paste into the ***mongos>*** prompt

**db.trades.insert(o)**

Step 10c: Copy/paste 6c and hit enter. *It will take ~5 minutes to insert lots of documents.*

**var j = 100;**

 **for( var i = 0; i < 1000000; i++ ) {**

 **if( i % 10000 == 0 ) {**

 **print(db.getLastError() + ' ' + i);**

 **}**

 **if( ++j >= 500 ) j = 100;**

 **o.ticker = 'z' + j;**

 **o.time = new Date(2012, 2, 3, 9, i%60, (i/60)%60, (i/3600)%1000);**

 **db.trades.insert(o);**

 **}**

Verify that the ***stock.trades*** data is visible via mongos. Note at this point the stock database isn't **"sharding enabled".**  But the data is visible via mongos.

Step 10: When it is done check the status of the ***trades*** collection (via ***count( )*** ).

Step 11: Browse the document.
**Part B: Enable sharding on database and collection.**

Step 12: Check the current index status and chunk status.

Step 13: Create a compound index and compound shard key, **ticker** & **time**, for the ***trades*** collection.

Step 14: Enable sharding and then check shard status.

Step 15: After sharding the collection, look at chunk status.

**Part C: Add a new shard**

Step 16: Start a regular ***mongod*** process as the 2nd  shard on a new unique *--* *port #*. Use ***--shardsvr*** along with --***dbpath shard2***.

Step 17: Add the 2nd shard to the cluster and check the shard status.

Step 18: Now wait for the balancer to move data among the two shards more evenly.

Step 19: Check shard status. ***Highlight the number of chunks across two shards.***