MongoDB System Investigation

1 Data Model

a What is the data model of your system? key-value, document column family?

MongoDB is a document database.

b Give an example of data stored in your system and the function to insert data. If you have installed the system, screenshots of inserting data are acceptable. Note, screenshots of your team using the system are acceptable for this and any question, screenshots of documentation are not acceptable for this question or any other question.

I use MongoDB Atlas

```
Matthew:~ ugp$ mongo "mongodb+srv://cluster0-hvt3i.gcp.mongodb.net/test"  --username buiman
MongoDB shell version v4.2.0
Enter password:
connecting to: mongodb://cluster0-shard-00-00-hvt3i.gcp.mongodb.net:27017,cluster0-shard-00-01-hvt3i.gcp.mongodb.ne
t:27017,cluster0-shard-00-02-hvt3i.gcp.mongodb.net:27017/test?authSource=admin&compressors=disabled&gssapiServiceNa
me=mongodb&replicaSet=Cluster0-shard-0&ssl=true
2020-04-27T18:51:21.598-0700 I  NETWORK  [js] Starting new replica set monitor for Cluster0-shard-0/cluster0-shard-
00-00-hvt3i.gcp.mongodb.net:27017,cluster0-shard-00-01-hvt3i.gcp.mongodb.net:27017,cluster0-shard-00-02-hvt3i.gcp.m
ongodb.net:27017
2020-04-27T18:51:21.599-0700 I  CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to cluster0-shard-00-01-hvt3i.
gcp.mongodb.net:27017
2020-04-27T18:51:21.599-0700 I  CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to cluster0-shard-00-00-hvt3i.
gcp.mongodb.net:27017
2020-04-27T18:51:21.599-0700 I  CONNPOOL [ReplicaSetMonitor-TaskExecutor] Connecting to cluster0-shard-00-02-hvt3i.
gcp.mongodb.net:27017
2020-04-27T18:51:22.011-0700 I  NETWORK  [ReplicaSetMonitor-TaskExecutor] Confirmed replica set for Cluster0-shard-
0 is Cluster0-shard-0/cluster0-shard-00-00-hvt3i.gcp.mongodb.net:27017,cluster0-shard-00-01-hvt3i.gcp.mongodb.net:2
7017,cluster0-shard-00-02-hvt3i.gcp.mongodb.net:27017
Implicit session: session { "id" : UUID("52af78ba-a5be-430b-b92c-748157ae746e") }
MongoDB server version: 4.2.6
```

2 User Interface`

a How do users interact with your system?

Using Mongo Shell, Application, and MongoDB Compass.

b Give two examples of interactions with your system. These interactions should be of different types – e.g. range queries or graph navigation queries. Screenshots of entering queries are acceptable.

```
@(shell):1:39
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.students.insertOne( { name: "Matthew Bui" } );
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5ea78d5ae9c01dfd807bcdd1")
}
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.students.find({ "name": "Matthew Bui" }).pretty()
{ "_id" : ObjectId("5ea78d5ae9c01dfd807bcdd1"), "name" : "Matthew Bui" }
MongoDB Enterprise Cluster0-shard-0:PRIMARY>
```

3 Indexes

a Does your system support indices? If so, what types of indices are supported?

Yes, it does. Index types: Single Field, Compound Index, Multikey Index, Geospatial Index, Text Indexes, Hash Indexes

b Give one or two examples of the creation of indices in your system. If your system supports primary and secondary indices, give an example of one of each. Again, screenshots of creating / using indices are acceptable.

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.students.createIndex( { name: 1 })
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1588040104, 2),
                "signature" : {
                        "hash" : BinData(0,"A6cqC1SxZACe/uYg6L2+u/skZYs="),
                        "keyId" : NumberLong("6818910453901557763")
                }
        },
        "operationTime" : Timestamp(1588040104, 2)
}
```

```
MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.students.createIndex({name:1,address:1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1588040886, 2),
                "signature" : {
                        "hash" : BinData(0,"t66FsCsWdCzv6LFnx8mxMjEuUOw="),
                        "keyId" : NumberLong("6818910453901557763")
                }
        },
        "operationTime" : Timestamp(1588040886, 2)
}
```

4 Consistency

a What consistency options and guarantees does your system support?

We can use “majority”, “local” for read concern and “majority” and specify numbers of “w” for write concern. The use combination of read and write concerns brings us different effect on consistency.

Read Concern	Write Concern	Read own writes	Monotonic reads	Monotonic writes	Writes follow reads
"majority"	"majority"	✅	✅	✅	✅
"majority"	{ w: 1 }		✅		✅
"local"	{ w: 1 }				
"local"	"majority"			✅	

b Does the system support any type of transactions?

In MongoDB, a write operation is atomic on a single documents, even for embedded documents. It has support an operation for multi-document transaction, “db.collection.updateMany()”, but it just helps your query, not your whole transaction because the operation is only atomic for each document.

5 Scalability and Replication

a Does your system support replication?

Yes, Mongo DB supports replication using replica set.

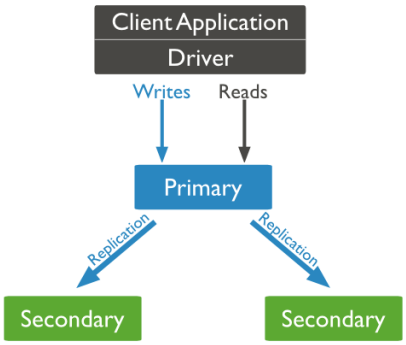
b Does your system support sharding?

Yes, Mongo DB supports sharding.

c Give examples of how one would set up replication and sharding in your system.

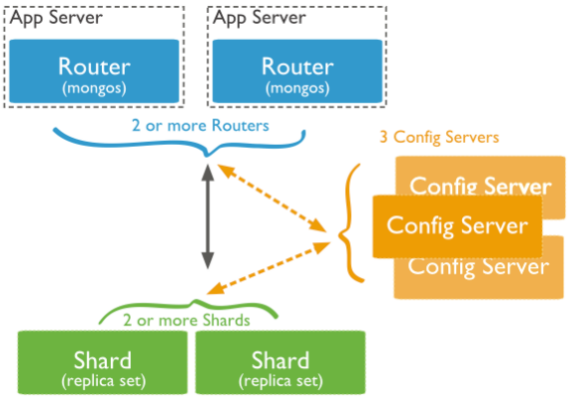
How to setup replica set[2]:

- A replicate set has at least three node: 1 primary and 2 secondary.
- 2 secondary replicates from the primary.
- When the failure occurs, 1 secondary would be picked to become a primary.
- After the recovery, the failed node becomes the secondary node.



How to setup sharding[3]:

- Each shard is a replica set.
- Config server provide the information about shard (metadata).
- Query routers are the middle points between server and application. They asks config server for metadata and go directly to shards(Like GFS).



REFERENCES:

[1] https://docs.mongodb.com/
[2] https://www.tutorialspoint.com/mongodb/mongodb_replication.htm
[3] https://www.tutorialspoint.com/mongodb/mongodb_sharding.htm