**MongoDB Shard Configuration**

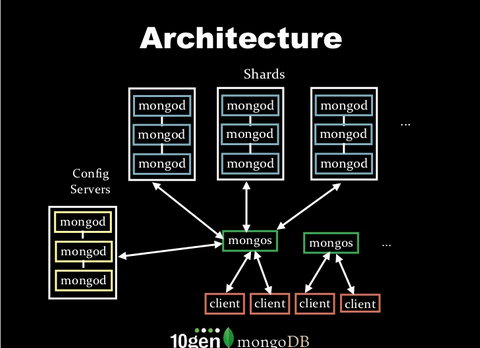
**References:** <http://docs.mongodb.org/master/MongoDB-Manual-master.pdf>

[**http://www.10gen.com/presentations/mongosv-2011/how-and-when-to-scale-mongodb-with-sharding**](http://www.10gen.com/presentations/mongosv-2011/how-and-when-to-scale-mongodb-with-sharding)

This document is assumes knowledge of MongoDB replication as covered in MongoDB Replica Set Configuration.

Sharding is the process of adding additional servers for scaling read and write performance. In a production envirionment , replica sets are sharded. Additional components such as a backup server and replication across data centers which are still relevant are are independent and coexist with replica sets used for sharding.

A sharded production MongoDB database environment looks like:



A MongoDB shard consists of:

1. 3 config servers in the yellow boxes on the bottom left You need 3 to keep a reliable cluster state. 3 provides a majority vote to determine correct data. If this data is corrupted or unstable the cluster is unusable and data may become unrecoverable. If there is 1 config server and the config server crashes the cluster is unusable(need to test for data loss and downtime).
2. 9 or more mongod instances to hold cluster data in the blue boxes. In a normal production environment each of the mongod instances is a replica set.
3. One or more mongos instances in the green boxes
4. Clients which if implemented as REST servers are the orange boxes.

**Goal create a simulated production cluster and deliver control scripts which are not provided by Mongodb. Deliver monitoring control scripts also. Create a backup and restore process.**

**First step: create a replica set architecture:**

Test; copy data/db to a separate server and get this up and running. Can you zip copy and read data? State of data/db must be more recent than the latest entry in oplog and then you an recover in 2 stages, 1) copy data directory over then replicate using oplog This is the same process as adding a new node, copy the /data/db directory over and let the oplog bring the new server up to state.

**Important: create a versioning system so you know where the replicas are and whet they are for.**

**Important: there aren’t any minimum 3 member replica set architectures. Miniumum are 4 member replica sets setup over different data centers using arbiters; preferred architecture is 2 replicas in site A, 2 secondary only in Site B and a n arbiter in site A.**

The difference with this configuration is there are 2 sites, one for production stability and for preventing a single point of failure. Site A and Site B. Site B is in a different data center. We can add a 1h backup in a separate data center also.

Create a 3 member replica set on separate computers? Yes, use mongod.conf to give every server the same port 27017. Set up /etc/hosts to see mongodb0, mongodb1, mongodb2.

**Can configure authentication using auth() and keyFile? This uses ssh. Not tested so 3 servers only pass encrypted traffic.**

**Test comman line replica set replSet=rs0/mongodb0, mongodb1,mongodb2**

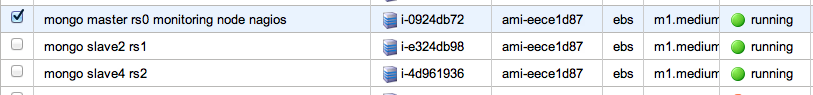
**Amazon AWS Mongo sharded replica set setup:**

This is a representation of a fully replicated Mongo Cluster setup. The cluster is first setup manually then automated in later documents using a combination of bash shell scripts, Puppet for configuration and Apache whirr for instance launching.

Start 3 replica sets on 3 different servers then add 6 more VM servers to reach 9 servers for a complete production ready replica set. Configure the mongos and 3 configuration servers.

Administrative tasks before Sharding:

Start 3 medium CentOS AWS instances, install MongoDB on them:



Start 3 AWS instancers first before scaling to 9. It is a lot easier to open up 3 ssh windows and modify files manually than 9 ssh windows. (Add Puppet recipes as solution) Add the following entries into /etc/hosts. Note: because the IP addresses change on each powerup you have to reset this file. This keeps the HostNames/IP address mappings in one spot.

You do not need to restart the networking components after this

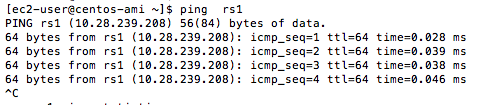
Contents of /etc/hosts on all 3 machines, rs0, rs1, rs2

10.29.25.214 rs00

10.28.239.208 rs01

10.28.211.83 rs02

Test you can ping to rs0, rs1 and rs2.



Test the replica set by adding a million rows, time the result and make sure you have the same amount of data in the 3 directories across 3 servers.

var startTime=new Date();

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

db.replicatest.save({“numentry”:i,“date”:new Date()});

}

var endTime=new Date();

print “elapsedTime:”+((endTime-startTime)/1000);

Run the following script on each server/VM where you want 3 replica set processes:

#!/bin/sh

replicadir1="slave1/db"

replicadir2="slave2/db"

replicadir3="slave3/db"

if [ ! -d $replicadir1 ];then

mkdir -p "$replicadir1"

fi

if [ ! -d $replicadir2 ];then

mkdir -p "$replicadir2"

fi

if [ ! -d $replicadir3 ];then

mkdir -p "$replicadir3"

fi

mongod --port 27017 --dbpath "$replicadir1" --replSet rs0 --fork --logpath slave1/log

mongod --port 27018 --dbpath "$replicadir2" --replSet rs0 --fork --logpath slave2/log

mongod --port 27019 --dbpath "$replicadir3" --replSet rs0 --fork --logpath slave3/log

mongo localhost:27017 --quiet replicaset.js

**Run the following script on each server/VM for the full deployment of 9 servers:**

**IS there splitting of server in MONGODB? No, you add a new server when one of the servers is full because it is out of disk space. Can you add an extra volume or drive to a running machine and how do you add a cold server to a cluster. Problem is the slow down in performance.**

Create a js init file in init.d to store the config and to init mongod on sudo service mongod restart. (not in tutorial)

init.js:

rs.initiate();

rs.config();

rs.add(mongodb1); rs.add(mongodb2);rs.add(mongodb3);

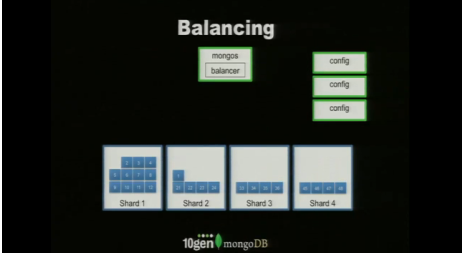
Configure a mongodb node as a secondary only node, this node can never be elected as a primary:

Configure a mongodb node as a backup node. This node is a delayed version by 1h. (Add this after the cluster is up)

Once the replica sets are running and verified to be correct(insert and verify data is there), convert this to sharded form:

**Second Step: convert replica set cluster to sharded cluster.**

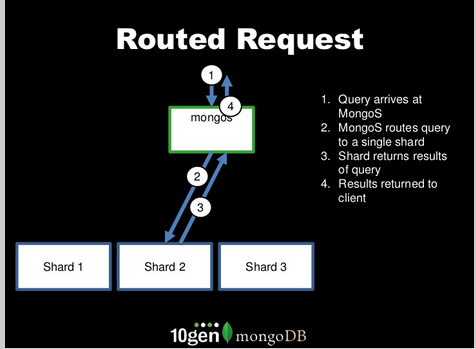
One of the runtime conditions in a shareded environment is the movement of chunks from 1 shard to another called balancing. These should take place one block at a time. This helps keep one server from overflowing and can result in better read performance.



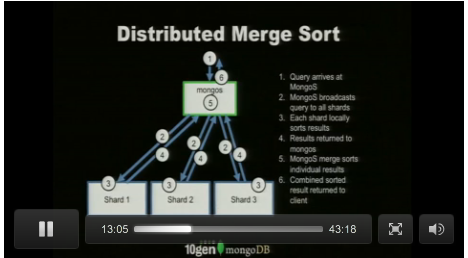
When a shard becomes to big and can no longer distribute chunks it should split:

There are 3 different types of requests in a sharded configuration. This is different than doing a scan over a result set.

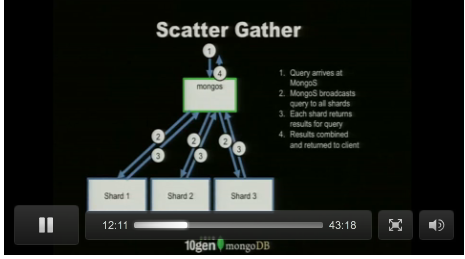
Routed request: a request sent to MongoS is directed to a particular shard. This is used if you know the shard key and can specify the server.



Distributed merge sort



**Scatter Gather:**

****

**Sharding Administration: Enable/Disable sharding. How to enter/remove data from a running cluster. Delete the replicas also after removal. Per collection. Like migration process**

**Journaling different from oplog. How???????**

**What is difference between using LVM snapshot and not using this? Snapshot 100k rows, upload and then download and test for data integrity. What does journaling do? When does the system use the journal? Looks like the replica/sharding uses oplog.**

**Multiple backup scenarios:**

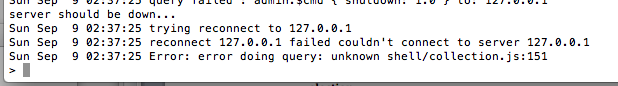
1. **can create backup onto existing system if there is room.**
2. **Cannot fit onto existing system, create separate replica and allow data to replicate or take existing replica off line backup then let it catch up. Using replica means turn off balancer.**
3. **Snapshot or binary dump are both alternatives.** 
   1. **Snapshot when the file system supports snapshots. LVVM example:**

**Q) After power on reset, what happened to the primary? After restart I only see secondary? Because we changed the /etc/hosts file and mongo cant find the other replicas so it defaults to secondary. Lets add the new servers as secondaries and wait for primary election.**

**Macintosh HD:Users:dc:Desktop:Screen Shot 2012-09-08 at 6.56.06 PM.png**

**To change host names, you have to do this manually. Is this process done using the config servers in normal operation?**

1. **db.shutdownServer(), verify you don’t see SECONDARY>**

****

1. **restart the mongod serversusing mongod –port 27017 –fork –dbapth srv/db –logpath srv/log**
2. **start a mongo shell and you should see a > prompt, start the rs configuration until you see the PRIMARY prompt. The rs initialization is different b/c there is already a config.**

**Journal file disappears over time. Where is this in the logs?**