MONGO SHARDING + GRIDFS

By Josh Fermin

WHAT IS SHARDING?

Storing data across multiple nodes/machines.

A single machine may run out of storage or have bottlenecked reads and writes

Sharding is **horizontally scalable** - Add machines as demand increases.

DATABASE ISSUES

High throughput and large amounts data put a lot of stress on a single server.

Set sizes larger than RAM stress I/O capacity

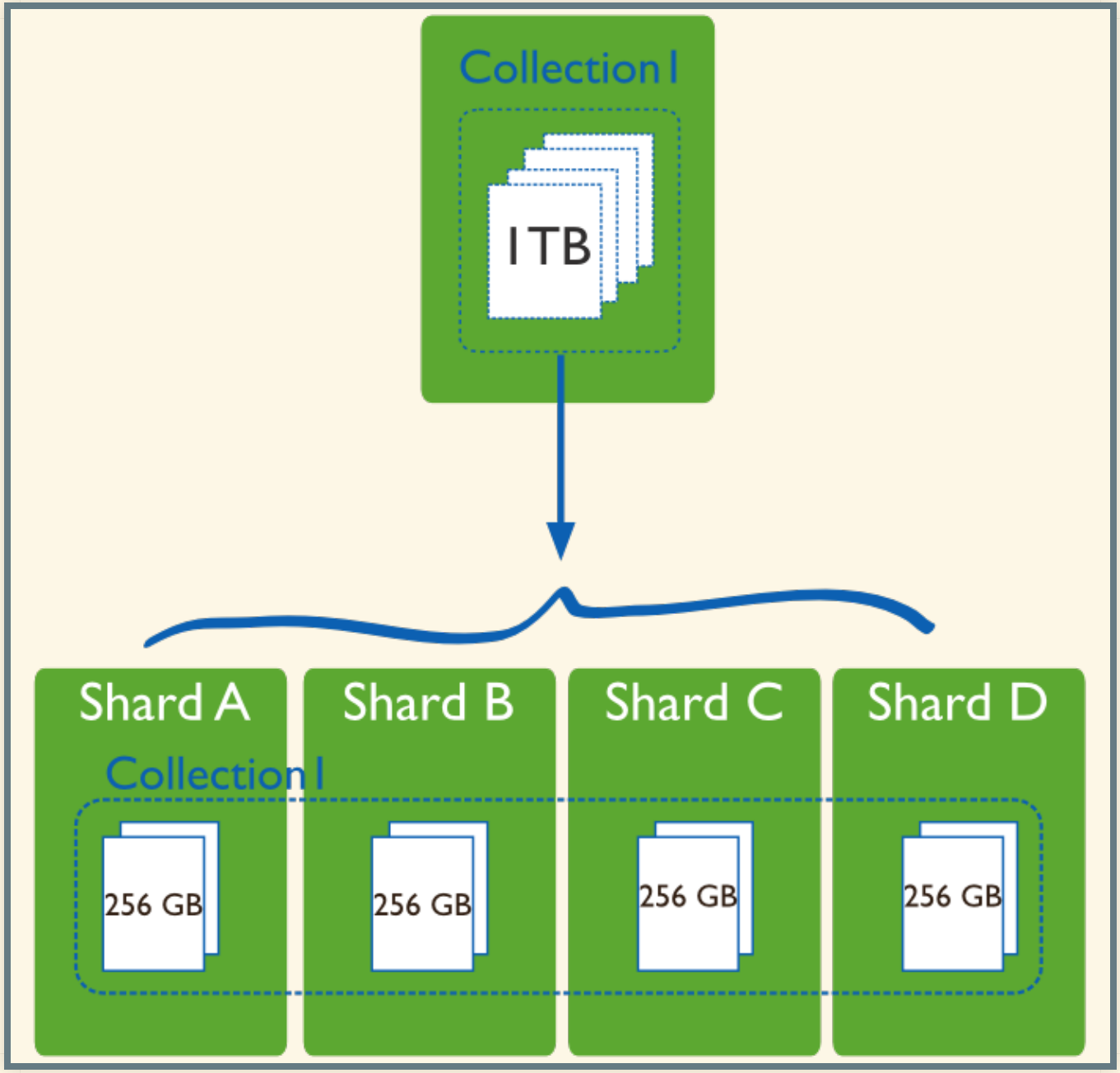
High query rates can exhuast CPU capacity of a single server.

TWO SOLUTIONS

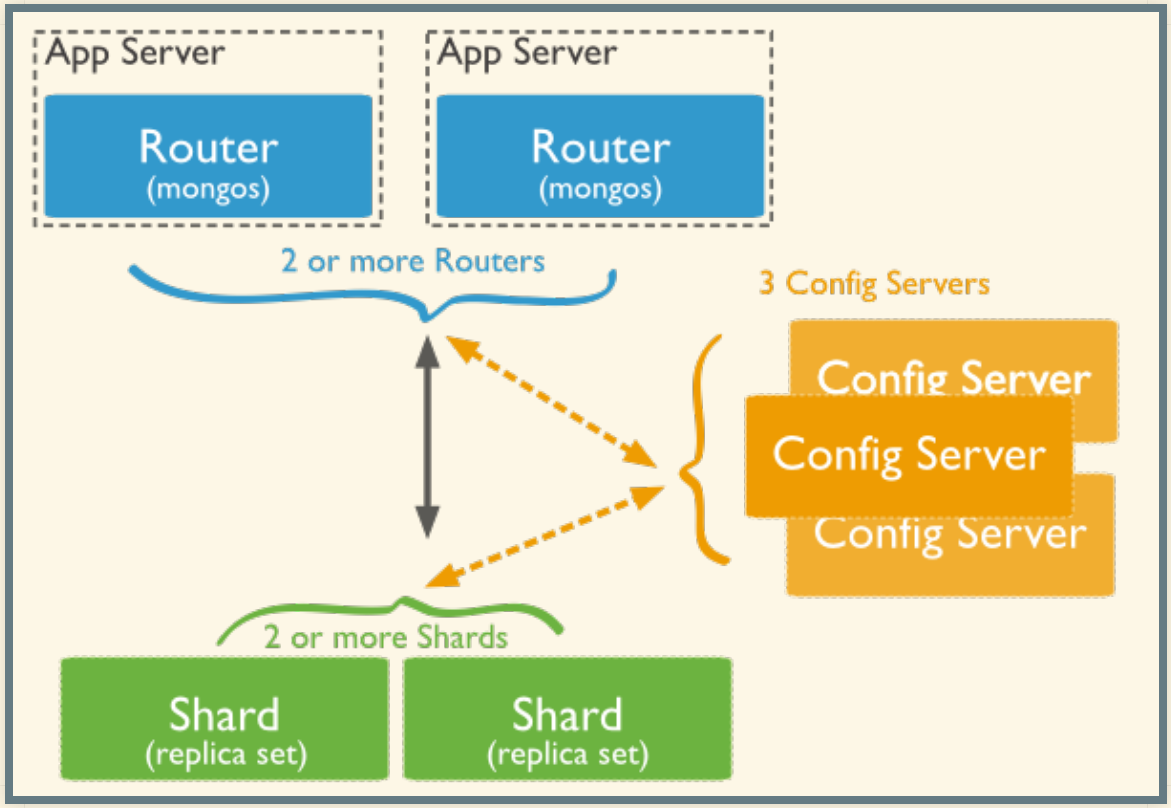
- Vertical Scaling:
  - Add more CPU and storage resources to a single server
  - Limitations: Expensive.

- Sharding (Horizontal Scaling):
  - Distribute data set over multiple servers or shards
  - Each shard is an independent db and all together they make up a single logical db.

SHARDED COLLECTION


Collection I
1 TB
Shard A
Collection I
256 GB
Shard B
256 GB
Shard C
256 GB
Shard D
256 GB

SHARDING IN MONGODB



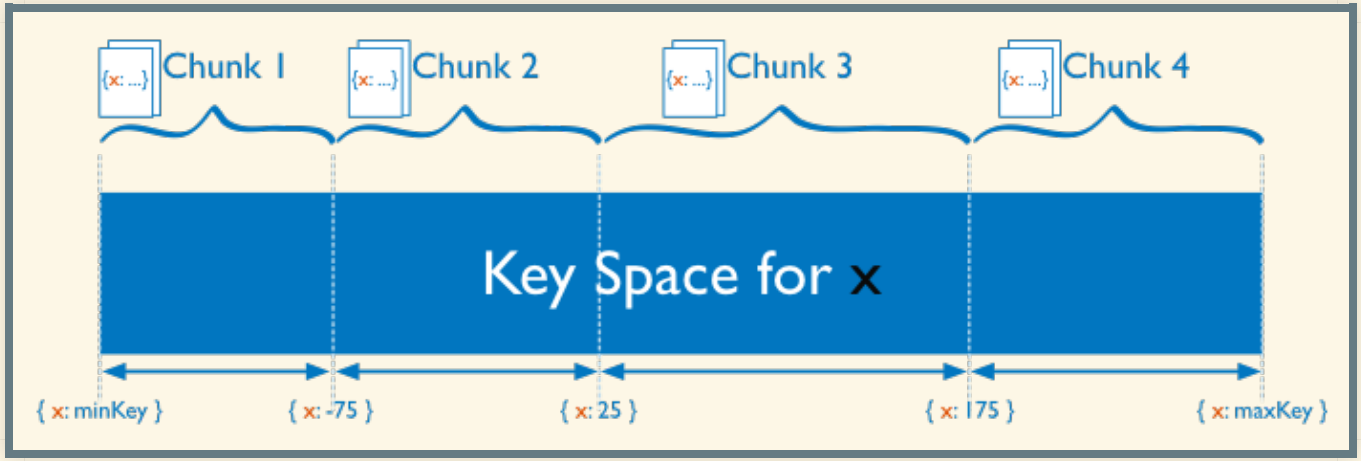
DATA PARTITIONING

Sharding happens on a collection level.

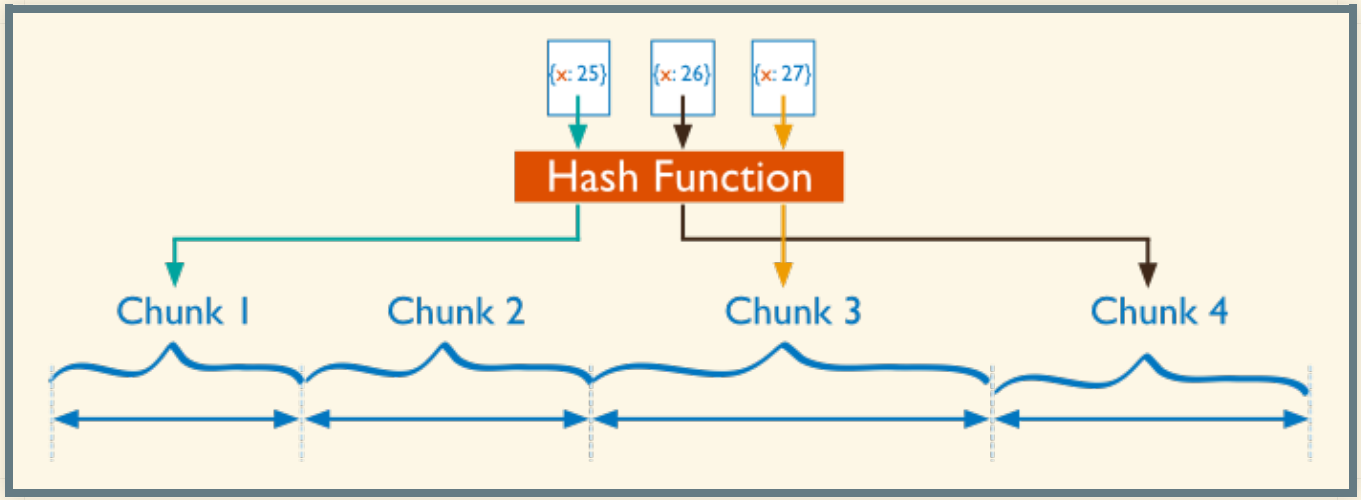
Sharding divides a collection's data by a shard key

Two types of partitioning: **range based** and **hash based**

RANGED BASED PARITIONING



HASH BASED PARITIONING



MONGO SHARDING TUTORIAL

Taken from Mongo Docs.

FIRE UP CONFIG SERVER

Download Mongo here

```
mkdir /data/configdb
```

```
mongod --configsvr --dbpath /data/configdb
```

- data/configdb is where server metadata will be stored
- In production, run configsvr on three different hosts

SETUP MONGOS - ROUTING SERVICE

```
mongos --configdb localhost:27019 --port 27017
```

- Lightweight, do not require data directories
- Usually have two of these in production

CREATE DBS FOR SHARDS

```
mkdir /data/shard1
```

```
mongod --dbpath /data/shard1 --port 27018 --shardsvr
```

- This starts a standalone mongod instance

CREATE ANOTHER SHARD

```
mkdir /data/shard2
```

```
mongod --dbpath /data/shard2 --port 27020 --shardsvr
```

Note different port

- Data will be partitioned to these two shards, based on shard key

FIRE UP MONGOS SHELL

```
mongo --port 27017
```

```
mongos> sh.addShard("localhost:27018")
{ "shardAdded": "shard0001", "ok" : 1 }
```

```
mongos> sh.addShard("localhost:27020")
{ "shardAdded": "shard0002", "ok" : 1 }
```

SHARDING ON DB/COLLECTION LEVEL

First create a database and a collection:

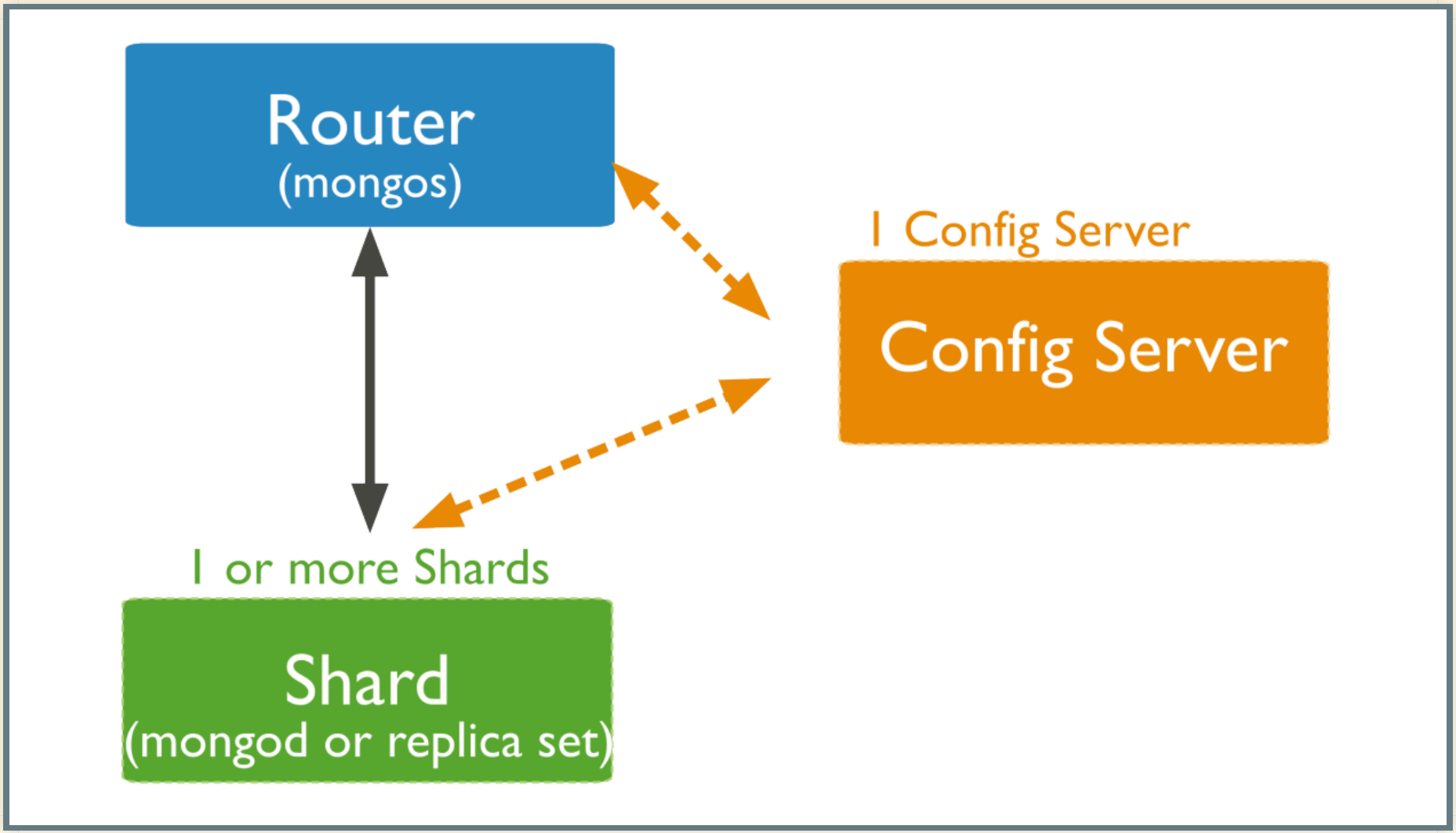
```
mongos> use testDB
mongos> db.createCollection(testCollection)
```

Then enable sharding on each:

```
mongos> sh.enableSharding("testDB")
mongos> sh.shardCollection("testDB.testCollection", { "md5":"hashed" })
```

Second param is choosing the shard key as md5 with hash partitioning. For range based do: "md5":1

WHAT WE JUST DID



SO WHAT IS THIS GRIDFS BUSINESS?

GridFS is an extension on top of mongo

Allows you to store/retrieve files that exceed BSON doc limit of 16mb

HOW IS THAT POSSIBLE?

Instead of one doc per file, GridFS breaks it into 255k chunks.

Each chunk becomes a separate document.

When you query for a file, the driver or client will reassemble it as needed.

GRIDFS COLLECTIONS

GridFS uses **two collections** to store files.

- Chunks: store binary chunks
- Files: store file metadata

IMPLEMENTATION

Two ways:

- Drivers: Basically a high level API
- mongofiles: mongo shell command line tool

EXAMPLE - C# DRIVER

```
using MongoDB.Driver;
using MongoDB.Driver.GridFS;
```

CONNECT TO MONGOS AND GET INFO

```
MongoClient client = new MongoClient("mongodb://localhost:27017");
MongoServer server = client.GetServer();
MongoDatabase database = server.GetDatabase("testDB");
MongoCollection testCollection = database.GetCollection("testCollection"
```

GRIDFS DRIVER

```
MongoGridFS gfs = new MongoGridFS(database);
gfs.Upload(@"C:\test.txt");
```

After uploading, the chunks and files collections will be created.

CONCLUSION

- If everything was set up correctly you can now read/write any files you want (movies, music, etc) into mongo using GridFS
- You can shard on the chunks collection if you'd like giving you horizontal scaling.
- Learned how to use MongoDB Sharding and GridFS API

RESOURCES

- Deploy a Sharded Cluster
- Mongo Sharding Guide