Scaling MongoDB

Sharding into, and beyond the Multi-Terabyte Range

MongoNYC 2013

Kenny Gorman
Founder, ObjectRocket

@objectrocket @kennygorman

Philosophy

Design

@scale

Philosophy

scalability is the ability of a system, network, or process to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth.

- André B. Bondi, 'Characteristics of scalability and their impact on performance', *Proceedings of the 2nd international workshop on Software and performance*

Philosophy

- Benefits of scaling horizontally
  - Adding compute in linear relation to storage
  - Use many smaller systems
  - Cost benefits
  - Start small, grow over time
  - Incremental approach

Philosophy

- Do I need to scale?

  yes...

  and...

  no...

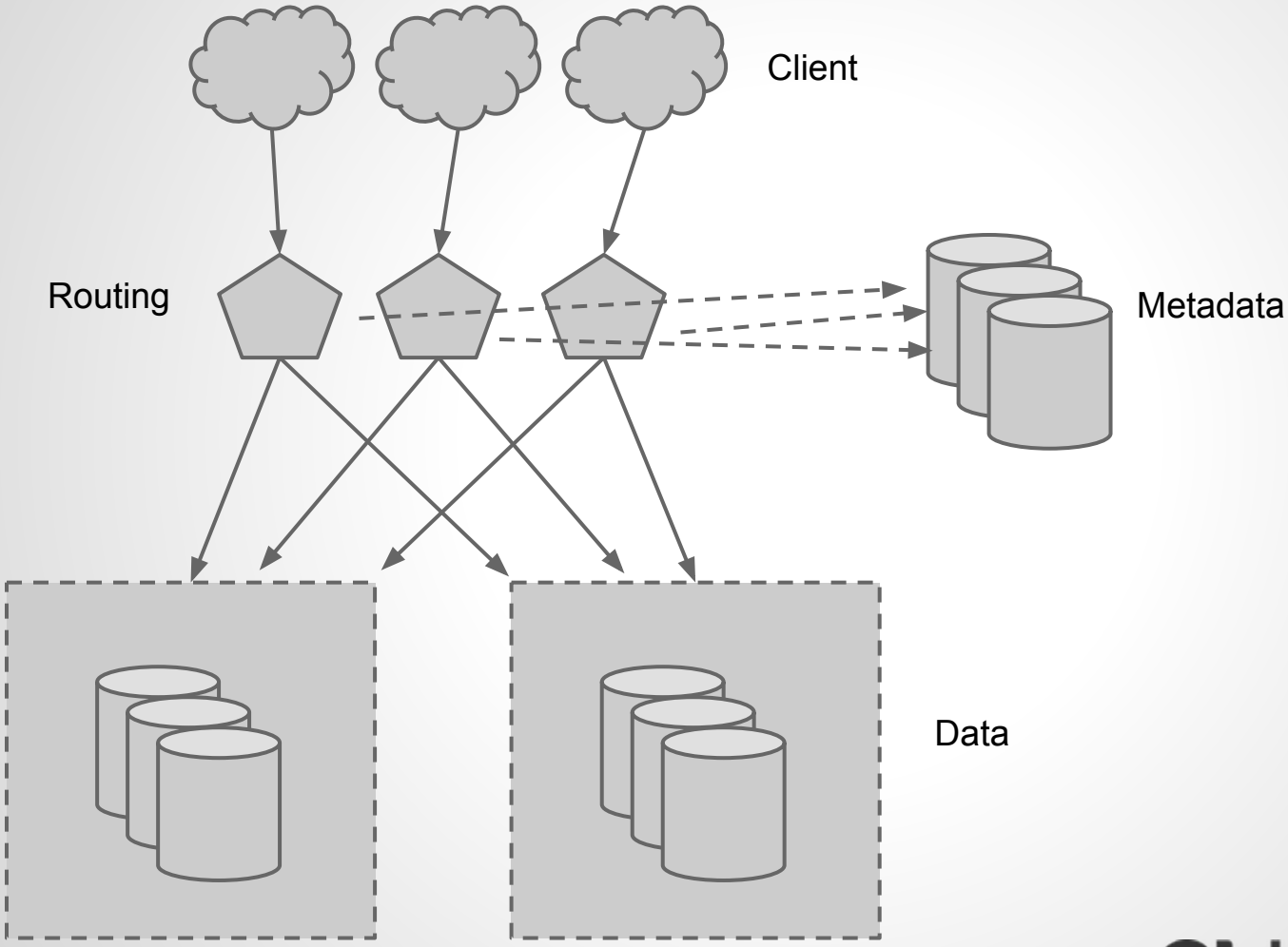
Philosophy

- Scale vertically works only for a while
  - Ratio's get whack
  - You are going to wish you did something else for a living
- Split workloads
  - Figure out what needs to be colocated with what.
- Then scale horizontally
  - Shard!

Design

- Scaling vertically vs horizontally
- MongoDB horizontal scalability; sharding
  - Sharding architecture
  - Sharding keys and collections
- Achieving your scaling goals
  - Tuning for writes
  - Tuning for reads
  - Lock scopes

Design - Architecture

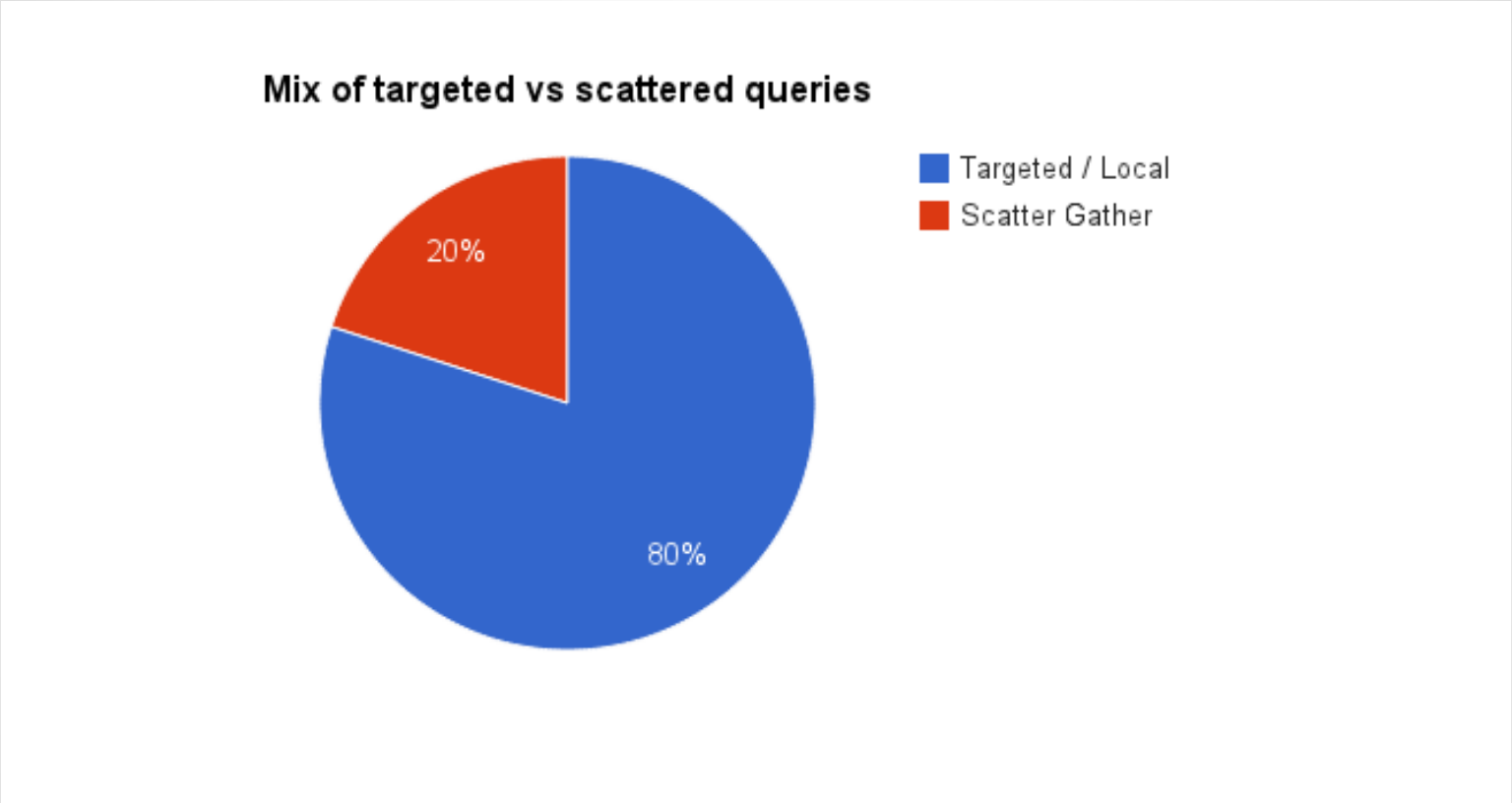


Design - Shard keys and patterns

- Keys
  - Range based keys
    - Know your access patterns
  - Hash based keys
    - More generic/easy option
  - Use profiler and explain() to identify queries and what patterns should be used.
  - Local and Scatter Gather
    - May not get every query to be local

PLAN AHEAD

Design - Shard keys and patterns

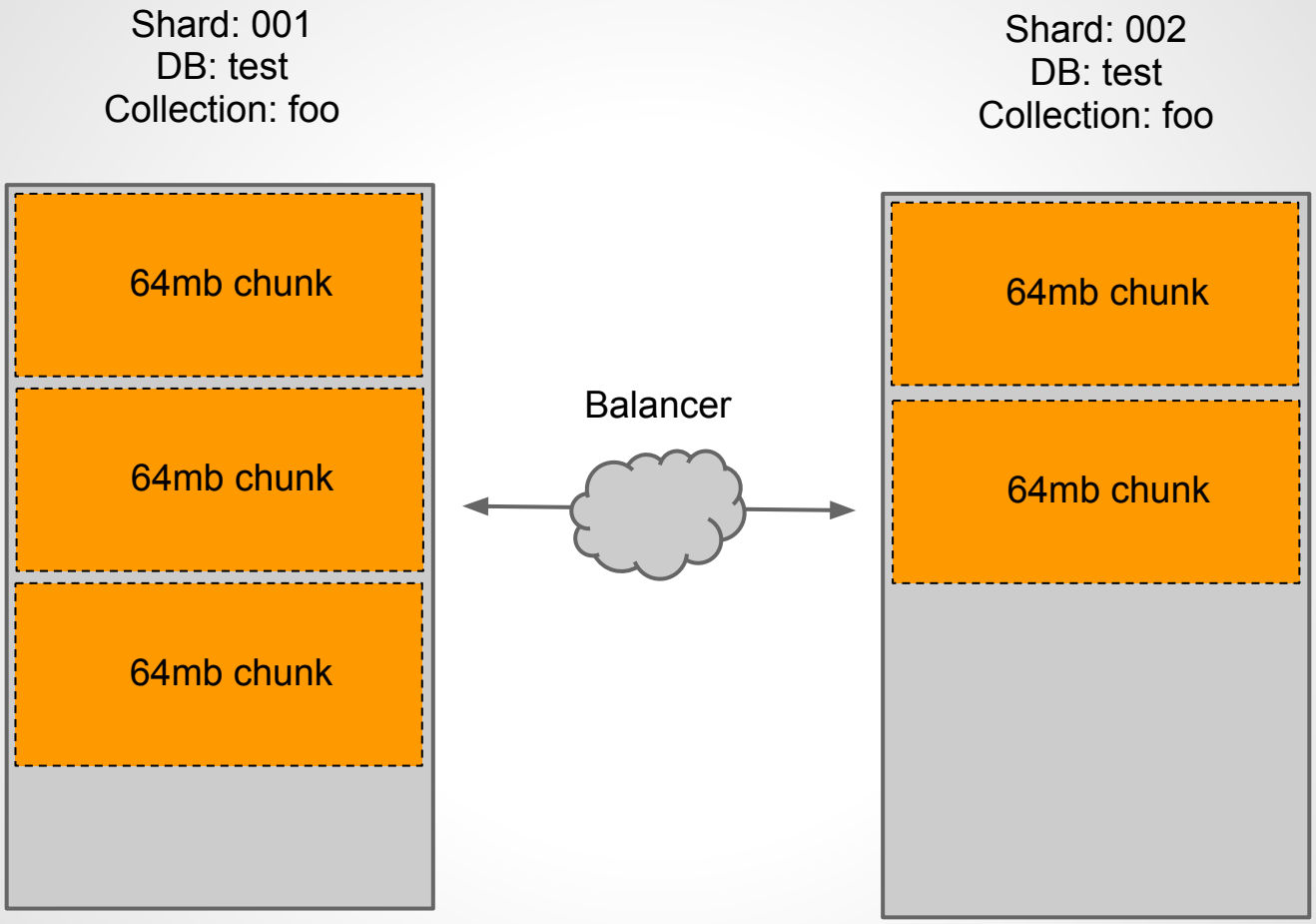


Design - Sharding Collections

- Shard Collection
- Based on a shard key
  - Range based
  - Hash based
- Chunks
- Chunk location
  - Balancer
  - Manual

```
{
 "_id" : "mydb.users",
      "lastmod" : ISODate("1970-01-16T20:33:39.757Z"),
      "dropped" : false,
      "key" : {
            "_id" : "hashed"
      },
      "unique" : false,
      "lastmodEpoch" : ObjectId("51a8d7a261c75a12f1c7f833")
}
```

Design - Chunks



Design - Chunks

```
{
 "_id" : "mydb.users-_id_-2315986245884394206",
     "lastmod" : { "t" : 89, "i" : 0 },
     "lastmodEpoch" : ObjectId("51a8d7a261c75a12f1c7f833"),
     "ns" : "mydb.users",
     "min" : { "_id" : NumberLong("-2315986245884394206") },
     "max" : { "_id" : NumberLong("-2237069208547820477") },
     "shard" : "d3b07384d113edec49eaa6238ad5ff00"}
{
     "_id" : "mydb.users-_id_-2395340237016371204",
     "lastmod" : { "t" : 88, "i" : 0 },
     "lastmodEpoch" : ObjectId("51a8d7a261c75a12f1c7f833"),
     "ns" : "mydb.users",
     "min" : { "_id" : NumberLong("-2395340237016371204")},
     "max" : { "_id" : NumberLong("-2315986245884394206")},
     "shard" : "15e894ac57eddb32713e7eae90d13e41"
}
```

Design - take aways

- Getting the proper shard key is critical. Once defined it's a pain in the a^$ to change.
- Creating a shard key that achieves your goals can sometimes be tricky. Take time to test this portion in dev/sandbox environments.
- Use profiler and explain() to figure out if you are using proper keys
- Understanding the Balancer's effect on your workload is critical. You probably need more I/O capacity than you think.

@scale - balancing

- Balancing is hard
  - Visibility
  - Balancer process

@scale - balancing

```
var balance_check = function(n) {
 if ( n ) {
          var output = db.chunks.aggregate([
               { $group : { _id : { "_id":"$ns", "shard":"$shard" },
                    chunks : { $sum : 1 } }},
               { $match : { "_id._id" : n } },
               { $sort : { "chunks" : 1 } }
               ]);
     } else {
          var output = db.chunks.aggregate([
               { $group : { _id : { "_id":"$ns", "shard":"$shard" },
                    chunks : { $sum : 1 } }},
               { $sort : { "chunks" : 1 } }
               ]);
     }
     printjson(output);
};
```

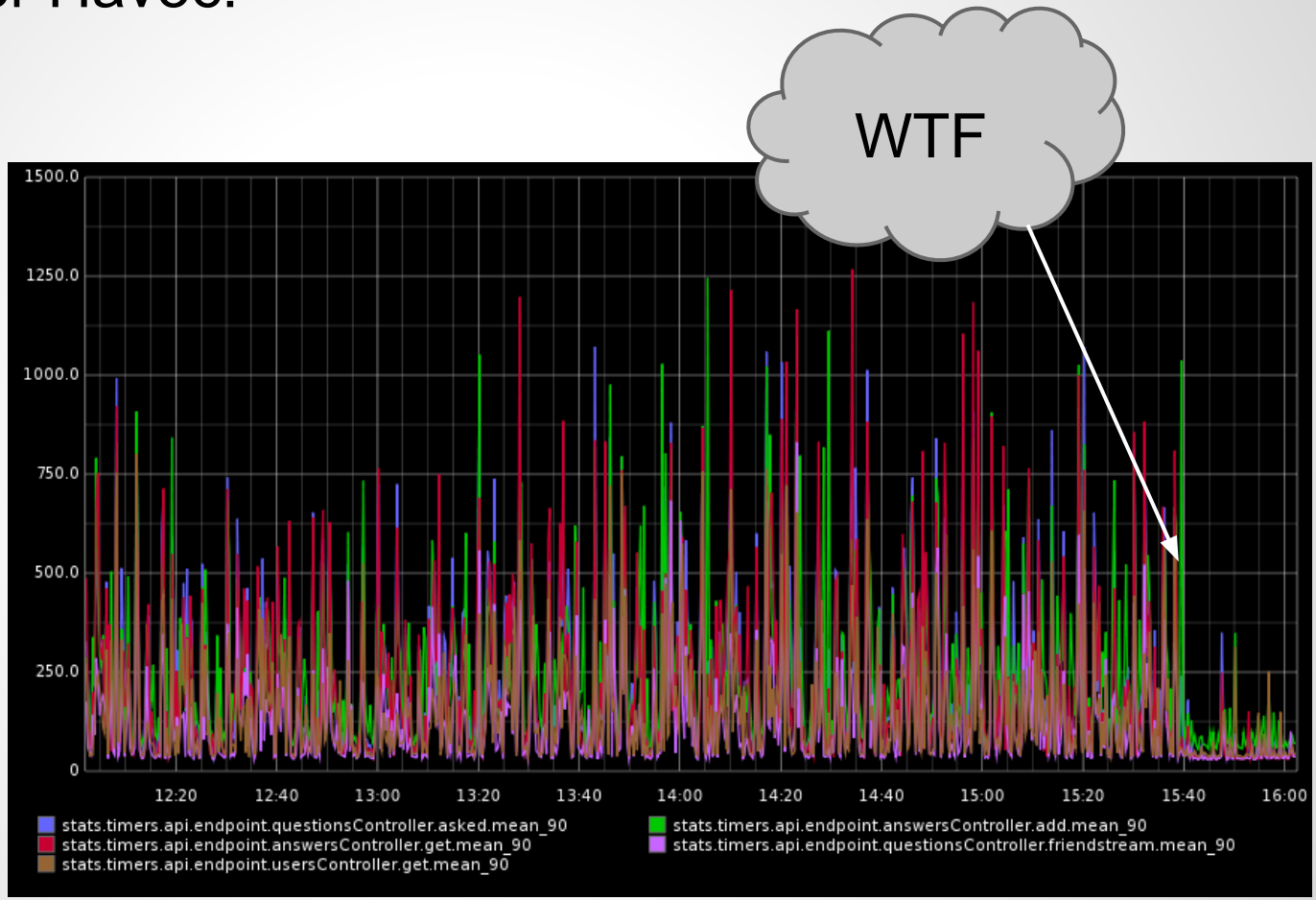
https://gist.github.com/kgorman/5775530

@scale - balancing

```
mongos> balance_check("mydb.users")
{ "result" : [
          { "_id" : {
                    "_id" : "mydb.users",
                    "shard" : "884e49a58a63060782d767feed8e6c88"
               },
               "chunks" : 1   #<------ !!!!!!! OH NO
          },
          { "_id" : {
                    "_id" : "mydb.users",
                    "shard" : "15e894ac57eddb32713e7eae90d13e41"
               },
               "chunks" : 77
          },
          { "_id" : {
                    "_id" : "mydb.users",
                    "shard" : "1134604ead16f77309235aa3d327bb59"
               },
               "chunks" : 77
          },
          { "_id" : {
                    "_id" : "mydb.users",
                    "shard" : "d3b07384d113edec49eaa6238ad5ff00"
               },
               "chunks" : 78
```

@scale - balancing

- Balancer Havoc!



ObjectRocket

@scale - balancing

- Balancer 'Fixes'
  - Pre-splitting
  - Windows
  - Micro-windows
  - Custom scripts
  - Don't use it

@scale - Monitoring

- Monitor everything. But some key items:
  - Shard size
  - Balancer on/off
  - Response time
  - Balance of OPS across shards
  - Failed migration of chunks
  - Locks
  - I/O

- Get histograms!
  - Graphite

@scale - Capacity

- Don't fail to plan
- Disk space/size is critical
  - maxSize()
  - extending disk space
  - adding cpu or memory capacity
  - Slave 'tricks'
    - Shell game
- Compute resources
- You need disk I/O no matter what anyone says.
  - Size for balancer workloads too

http://blog.foursquare.com/2010/10/05/so-that-was-a-bummer/

1. *We're making changes to our operational procedures to prevent **overloading**, and to ensure that future occurrences have safeguards so foursquare stays up.*

@scale

- Things to watch for
  - Out of disk space or other resources
    - Don't wait!
  - Balancer havoc
  - No more I/O left
  - Shard Asymmetry
  - Scatter gather's

- Things to ensure you do
  - Use maxSize, leave yourself a bit of wiggle room
  - Leave profiler on!
  - Explain and profile your queries

Contact

@kennygorman
@objectrocket
kgorman@objectrocket.com
https://www.objectrocket.com

WE ARE HIRING!
https://www.objectrocket.com/careers/