Advanced Database Design

Gregory S. DeLozier, Ph.D.

gdelozie@kent.edu

Mongo Replication

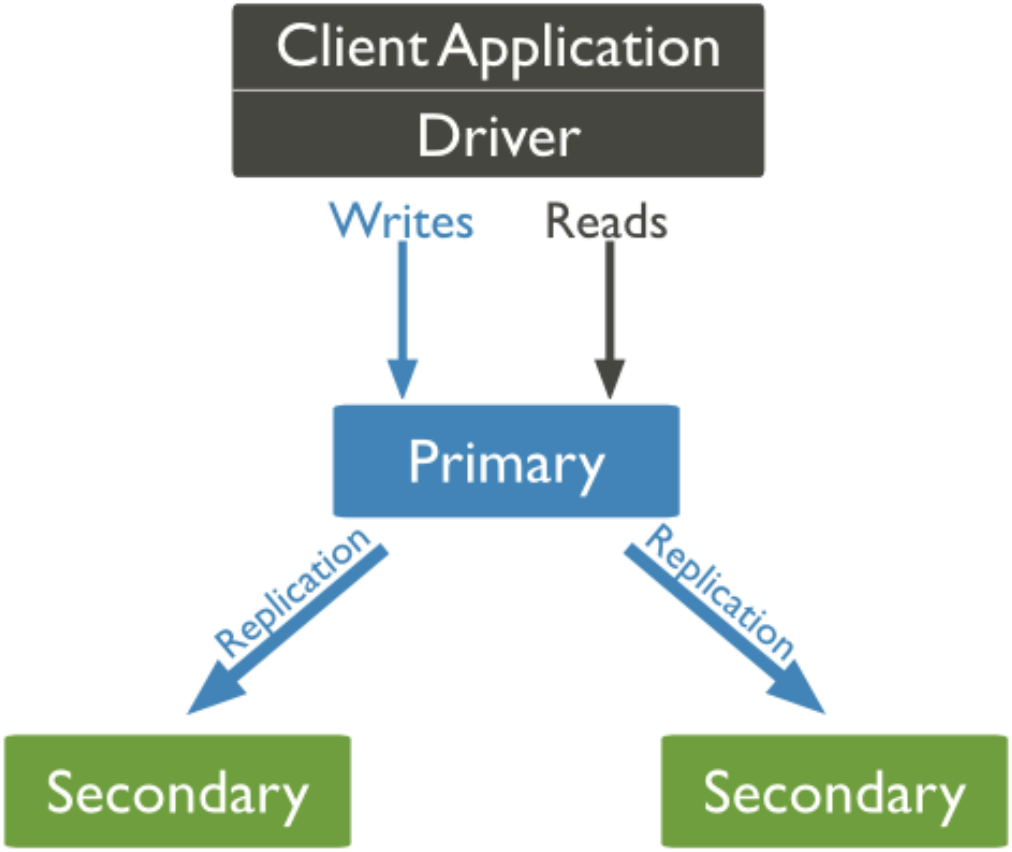
- Why replicate?
  - Safer – reduces chances of loss
  - Increase read capacity – read from replicas

- What are replicas?
  - Mongod instances that have the same data
  - At different ports or URLs

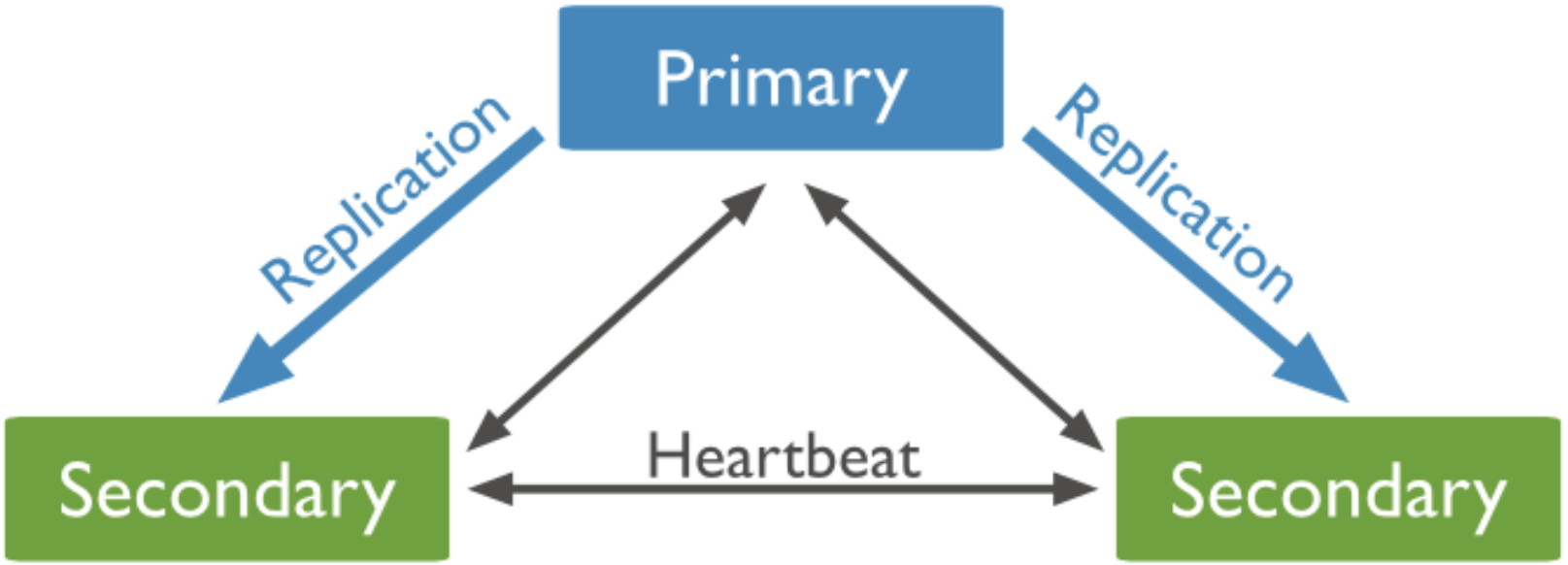
Replica Sets

- Primary
  - The controlling member
  - There can only be one
  - Accepts all write operations
- Secondary
  - Controlled by primary – duplicates changes
  - Can be many of these
  - Can directly accept read requests

Replica Set Concept



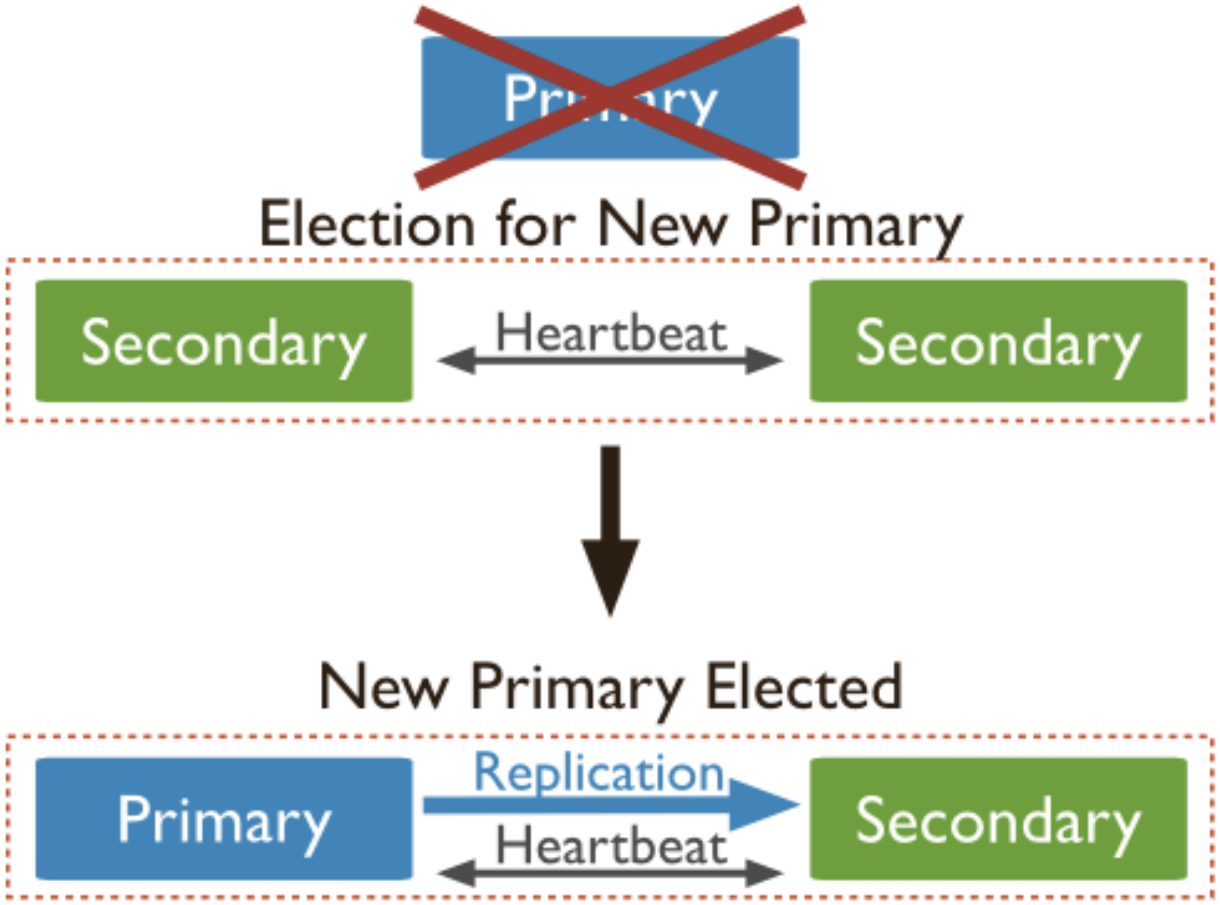
Heartbeat Signal



Replica Communication

- If heartbeat not received from primary
  - Missing heartbeat is failure mode
  - Secondaries vote on who becomes primary
  - Eventually one secondary gets a majority
  - That secondary becomes the new primary
- New primary communicates by:
  - Heartbeat
  - Responds to requests to the replica set
- If old primary recovers, it becomes secondary

Replication Election



Set Up A Replica Cluster

- Set up a 3-way replicaset
- Based on:
  - http://api.mongodb.org/python/current/examples/high_availability.html

Set up some Mongo instances

- mkdir -p ~/data/db0 ~/data/db1 ~/data/db2
- $ mongod --port 27017 --dbpath ~/data/db0 --replSet foo &
- $ mongod --port 27018 --dbpath ~/data/db1 --replSet foo &
- $ mongod --port 27019 --dbpath ~/data/db2 --replSet foo &

Initiate the Replication Set

- Using PyMongo
- **>>> from pymongo import** MongoClient
- **>>>** c = MongoClient('localhost', 27017)
- **>>>** config = {'_id': 'foo', 'members': [
  {'_id': 0, 'host': 'localhost:27017'},
  {'_id': 1, 'host': 'localhost:27018'},
  {'_id': 2, 'host': 'localhost:27019'}
  ]}
- **>>>** c.admin.command("replSetInitiate", config)

Connect to the Replica Set

- Any of these will work:

```
>>> MongoClient('localhost', replicaset='foo')
MongoClient('localhost', 27017)
>>> MongoClient('localhost:27018', replicaset='foo')
MongoClient('localhost', 27018)
>>> MongoClient('localhost', 27019, replicaset='foo')
MongoClient('localhost', 27019)
>>> MongoClient('mongodb://localhost:27017,localhost:27018/?replicaSet=foo')
MongoClient(['localhost:27017', 'localhost:27018'])
```

- Construction is non-blocking, takes time:

```
>>> from time import sleep
>>> c = MongoClient(replicaset='foo'); print c; sleep(0.1); print c
MongoClient('localhost', 27017)
MongoClient([u'localhost:27019', u'localhost:27017', u'localhost:27018'])
```

Demonstrate Failover

```
>>> db = MongoClient("localhost", replicaSet='foo').test
>>> db.test.insert_one({"x": 1}).inserted_id
ObjectId('...')
>>> db.test.find_one()
{u'x': 1, u'_id': ObjectId('...')}
```

```
>>> db.client.address
('localhost', 27017)
```

Kill service on 27017

```
>>> db.test.find_one()
Traceback (most recent call last):
pymongo.errors.AutoReconnect: ...
```

Eventually new primary is elected

```
>>> db.test.find_one()
{u'x': 1, u'_id': ObjectId('...')}
>>> db.client.address
('localhost', 27018)
```

Secondary Reads

```
>>> client = MongoClient(
...     'localhost:27017',
...     replicaSet='foo',
...     readPreference='secondaryPreferred')
>>> client.read_preference
SecondaryPreferred(tag_sets=None)
```

```
>>> from pymongo import ReadPreference
>>> client.read_preference
SecondaryPreferred(tag_sets=None)
>>> db = client.get_database('test', read_preference=ReadPreference.SECONDARY)
>>> db.read_preference
Secondary(tag_sets=None)
>>> coll = db.get_collection('test', read_preference=ReadPreference.PRIMARY)
>>> coll.read_preference
Primary()
```

About Vagrant (time permitting)

- Vagrantup.com
- -- demo time –

Homework

- Get a replica set working
- Make the Todo app use it!
- Kill a server
- Watch it recover
- Restart the server
- Show that it still works

MongoDB

- Resources
  - http://www.mongodb.org
  - http://www.mongodb.org/about/introduction/
  - http://docs.mongodb.org/manual/
  - https://mongolab.com/