

# MongoDB 复制集（副本集）

李焕贞

河北师范大学软件学院



# 本章大纲

- 复制集的介绍
- 复制集的使用
- $O(\log)$

# ■ 复制集

复制：将一个数据库实例中的所有数据库**改变复制**到另一个独立的数据库实例中。

主从复制（旧版本）：一旦主库出现故障，需要**手动**把主库切换到最可靠的从库，而其他从库就得到从新的主库去同步

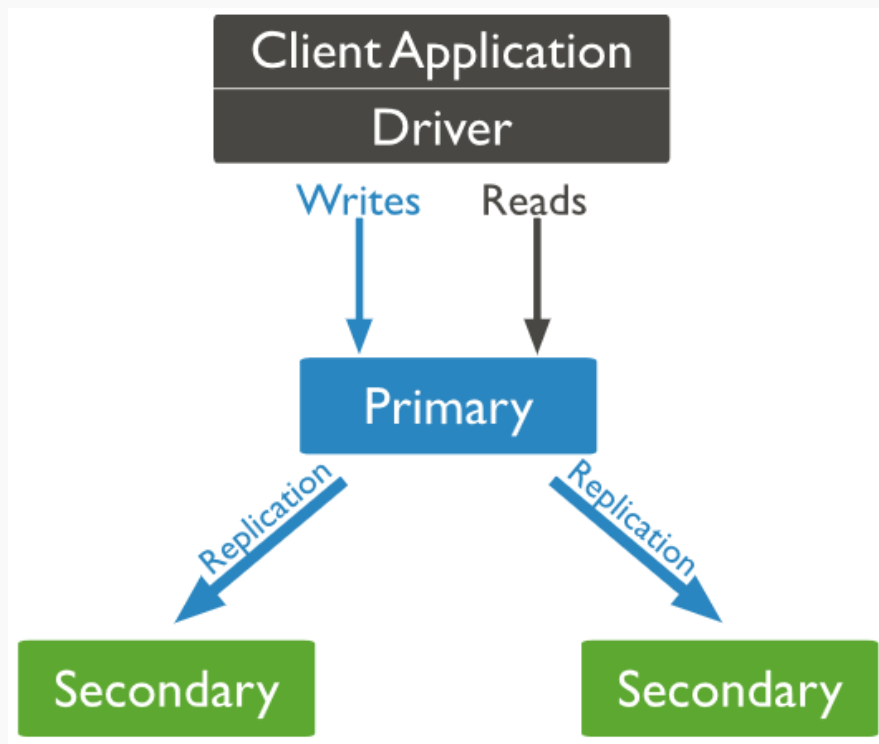
**副本集（replSet复制集）**：主库出现故障时，能自动主从切换，从而故障得以恢复，其它从库从新的主库同步数据，整个过程不需要手工干预。

副本集就是能从主从自动切换的复制

# ■ 数据的冗余性和可用性

- 复制集提供了冗余，并提高了数据的可用性
- 可读能力
- 备份
- 灾难恢复
- 报告

## 复制集-定义



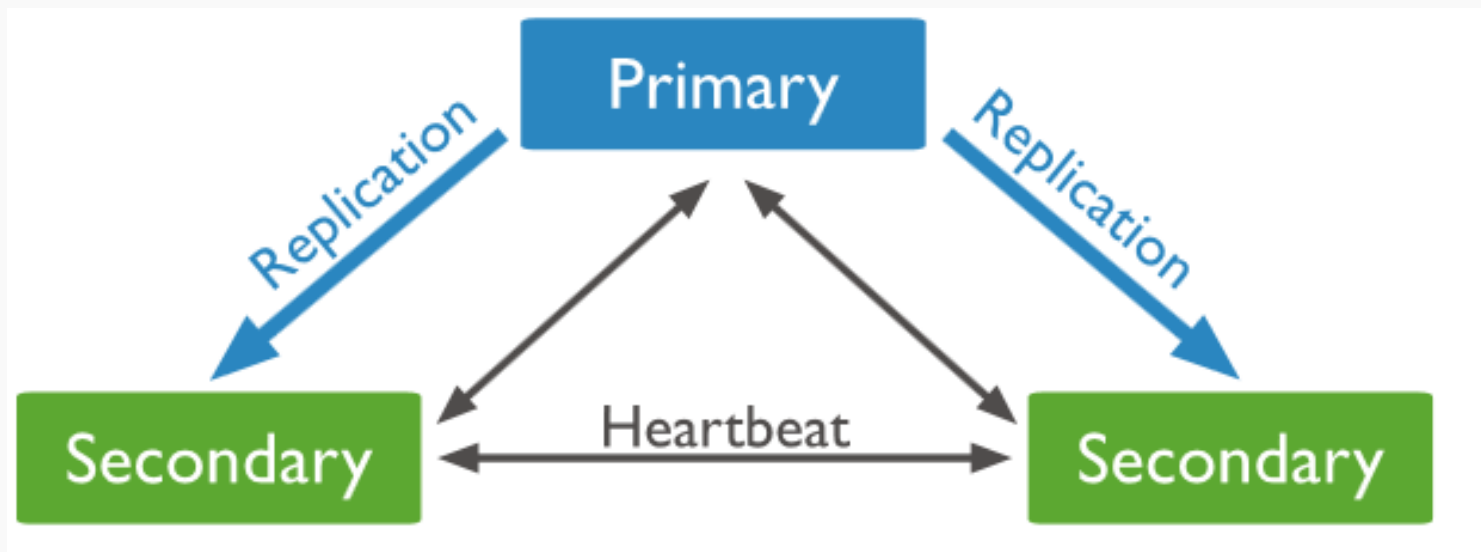
# 复制集-原理

## MongoDB复制原理

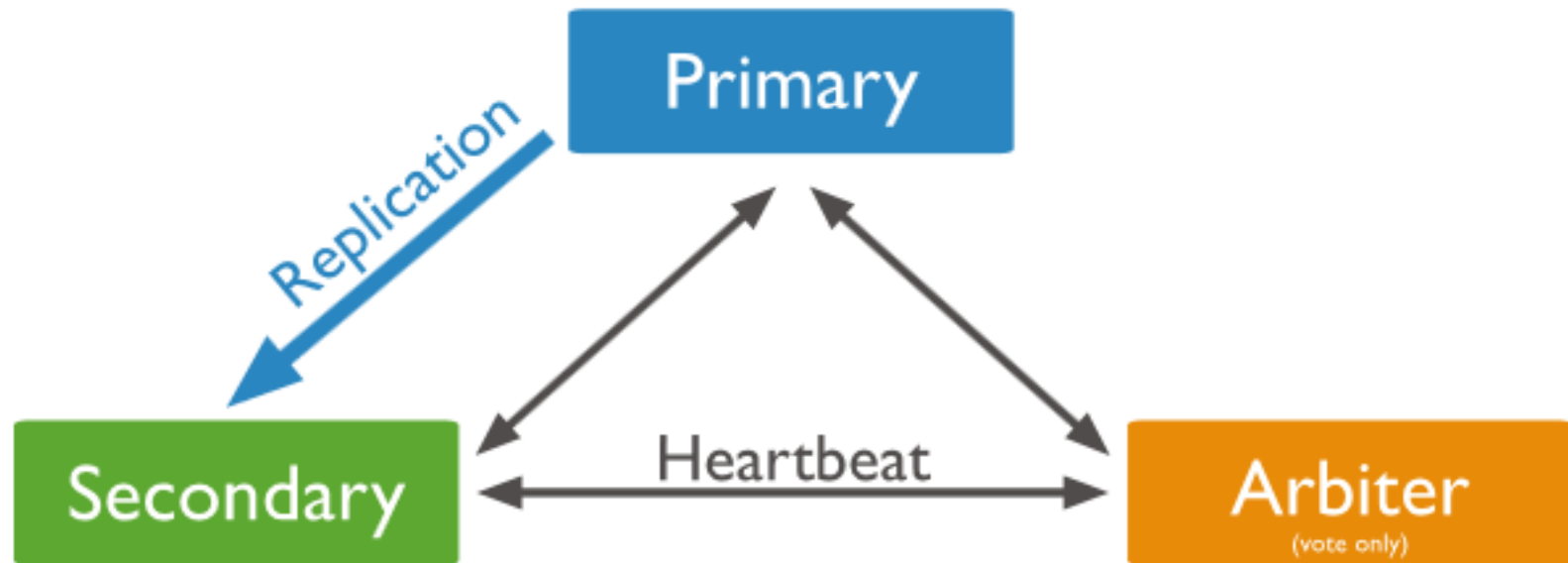
MongoDB的复制至少需要两个节点。其中一个为主节点，负责处理客户端请求，其余的都是从节点，负责复制主节点上的数据。

MongoDB各个节点常见的搭配方式为：一主一从、一主多从。主节点记录在其上的所有操作oplog，从节点定期轮询主节点获取这些操作，然后对自己的数据副本执行这些操作，从而保证从节点的数据与主节点一致。

## ■ 复制集-架构



## 复制集-架构

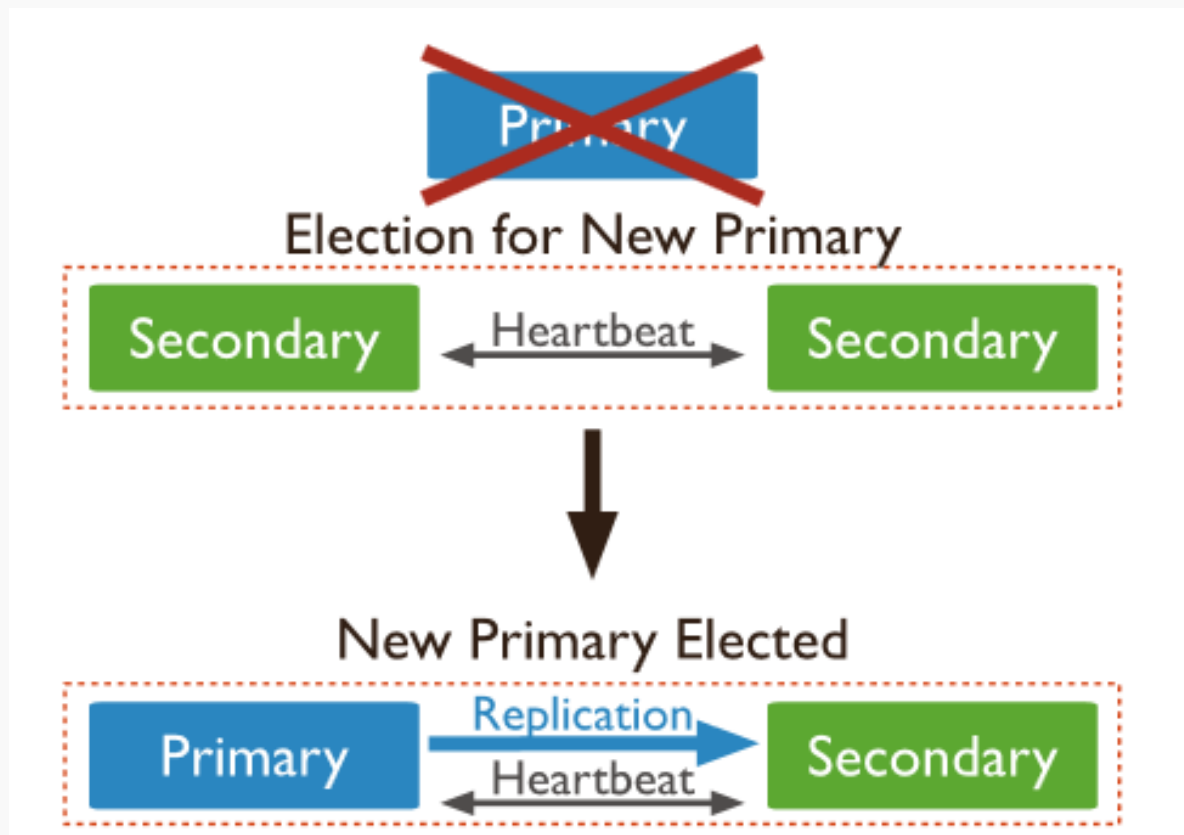




# 复制集-特性

- 异步复制
- 自动故障切换
- 读操作
- 附加功能

## 自动故障切换





# 本章大纲




- 复制集的介绍
- 复制集的使用
- $O(p \log p)$

# 复制集的搭建

1. 复制集的搭建
2. 添加仲裁节点
3. 添加意向候选节点
4. 设置某个从节点不能成为主节点
5. 配置隐藏从节点
6. 配置延迟节点
7. 配置无选举权节点
8. 将从节点变为仲裁者

# 复制集的搭建

准备：创建相关目录

台电脑 > 新加卷 (D:) > Program Files > MongoDB > Server > 4.0 > data > repl		
名称	修改日期	类型
 p1	2019/3/18 11:27	文件夹
 s1	2019/3/18 11:27	文件夹
 s2	2019/3/18 11:27	文件夹

# ■ 复制集的搭建

## 1、启动多个mongodb实例

```
mongod --dbpath=p1\ --smallfiles --port 10000 --replSet "magic"
```

```
mongod --dbpath=s1\ --smallfiles --port 10001 --replSet "magic"
```

```
mongod --dbpath=s2\ --smallfiles --port 10002 --replSet "magic"
```

## 2、连接任意一台mongodb实例（主节点）

```
mongo --port 10000
```

# 复制集的搭建

3、（主节点）初始化复制集

```
rs.initiate()
```

4、（主节点）查看复制集的状态

```
rs.status()
```

5、（主节点）另外两个从节点加入复制集

```
rs.add("127.0.0.1:10001")
```

```
rs.add("127.0.0.1:10002")
```

# 添加仲裁节点

1、启动仲裁节点所需的mongodb实例

```
mongod --dbpath=arbiter\ --smallfiles --port 10003 --replSet "magic"
```

2、连接上主节点

```
mongo --port 10000
```

3、添加仲裁节点到副本集中

```
rs.addArb("127.0.0.1:10003")
```

4、查看复制集的状态

```
rs.status()
```



# 添加意向候选节点

1、获取当前副本集配置信息

```
var cfg=rs.conf()
```

2、设置相应的权值

```
cfg.members[1].priority=10
```

3、重新设置复制集

```
rs.reconfig(cfg)
```

4、查看复制集的状态

```
rs.status()
```

# ■ 设置某个从节点不能成为主节点

1、获取当前复制集配置信息

```
var cfg=rs.conf()
```

2、将不希望成为主节点的从节点的priority设置为0

```
cfg.members[1].priority=0
```

3、重新设置复制集

```
rs.reconfig(cfg)
```

4、查看复制集的状态

```
rs.status()
```

# 配置隐藏从节点

1、获取当前复制集配置信息

```
var cfg=rs.conf()
```

2、将隐藏节点的priority设置为0

```
cfg.members[1].priority=0
```

3、将隐藏节点的hidden设置为true

```
cfg.members[1].hidden=true
```

4、重新设置复制集

```
rs.reconfig(cfg)
```

5、查看复制集的状态

```
rs.status()
```

## 配置延迟节点

1、获取当前复制集配置信息

```
var cfg=rs.conf()
```

2、将隐藏节点的priority设置为0

```
cfg.members[1].priority=0
```

3、将隐藏节点的hidden设置为true

```
cfg.members[1].hidden=true
```

# 配置延迟节点

## 4、设置延迟节点的延迟时间

`cfg.members[1].slaveDelay=60*60*24`，单位是秒

## 5、重新设置复制集

`rs.reconfig(cfg)`

## 6、查看复制集的状态

`rs.status()`

注意：需要`rs.slaveOk()`才允许从延迟节点读取数据

# 配置无选举权节点

## 1、获取当前复制集配置信息

```
var cfg=rs.conf()
```

## 2、将希望设置无选举权节点votes为0

```
cfg.members[0].votes=0
```

## 3、重新设置复制集

```
rs.reconfig(cfg)
```

## 4、查看复制集的状态

```
rs.status()
```

# ■ 将从节点变为仲裁者

1、确认该节点与所有客户端都断开连接mongostat

2、关闭该从节点

3、（主节点）将从节点从REPL的配置信息中删除

```
rs.remove("127.0.0.1:10002")
```

4、（主节点）确认复制集已经没有该节点

```
rs.conf()
```

5、将从节点的数据目录删除或者重命名

# ■ 将从节点变为仲裁者

6、创建新的目录供仲裁节点使用

7、启动仲裁节点所需的mongodb实例

```
mongod --dbpath=arbiter\ --smallfiles --port 10005 --replSet "magic"
```

8、连接上主节点

```
mongo --port 10000
```

9、添加仲裁节点到副本集中

```
rs.addArb("127.0.0.1:10005")
```

10、查看复制集的状态

```
rs.status()
```





# 本章大纲

- 复制集的介绍
- 复制集的使用
- $O_p \log$

# 什么是 $O(\log)$

一种特殊的封闭集合

保留了有限的、所有的操作

异步复制

幂等操作

# ■ 什么是Opllog

有限集合:

```
db.runCommand({create:"cap",capped:true,size:1024,  
max:1000})
```

最多只能有1000条文档

# Oplog的大小

## For Unix and Windows systems

The default oplog size depends on the storage engine:

Storage Engine	Default Oplog Size	Lower Bound	Upper Bound
<a href="#">In-Memory Storage Engine</a>	5% of physical memory	50 MB	50 GB
<a href="#">WiredTiger Storage Engine</a>	5% of free disk space	990 MB	50 GB
<a href="#">MMAPv1 Storage Engine</a>	5% of free disk space	990 MB	50 GB

# Oplog的大小

## For 64-bit macOS systems

The default oplog size is 192 MB of either physical memory or free disk space depending on the storage engine:

Storage Engine	Default Oplog Size
In-Memory Storage Engine	192 MB of physical memory
WiredTiger Storage Engine	192 MB of free disk space
MMAPv1 Storage Engine	192 MB of free disk space

# 改变Oplog的大小

## 1、连接数据库

```
mongo --host <hostname>:<port>
```

## 2、查看oplog大小

```
use local
```

```
db.oplog.rs.stats().maxSize
```

## 3、修改oplog大小

```
db.adminCommand({replSetResizeOplog: 1, size: 16000})
```

# 复制集的管理

- 灾备切换
- 数据回滚
- 常用命令

# ■ 灾备切换

fail over策略

新的副本集，默认情况下，使用版本1进行副本集复制，3.2版本之前采用版本0的协议，此外引入新的配置选项electiontimeoutmillis





# 灾备切换

环境：1主2从1仲裁

- 1、主库宕机 （use admin / db.shutdownServer()）
- 2、从库宕机
- 3、主库宕机后，1从也宕机
- 4、仲裁下线后

# 数据回滚

回滚操作是对原主节点成员在故障转移后重回副本集的写操作。

回滚在以下情况是不会发生的，如果写操作在宕机主节点前复制到副本集的另一个从节点中了，并且该成员仍然可用，并能够访问到大多数的副本集。

```
<database>.<collection>.<timestamp>.bson
```

```
bsondump
```

# 常用命令

`rs.add()` 添加复制集节点

`rs.addArb()` 添加副本集仲裁节点

`rs.conf()` 显示当前复制集配置文件

`rs.freeze()` 暂时冻结当前节点不参与竞选主节点

`rs.initiate()` 初始化复制集

`rs.printReplicationInfo()` 打印复制集信息

`rs.printSlaveReplicationInfo()`，打印复制集从节点信息

## 常用命令

`rs.status()` 打印复制集状态

`rs.remove()` 删除某个节点

`rs.stepDown()` 将当前复制集节点下线

`rs.slaveOk()` 默认从节点不允许读取数据，此命令允许当前session通过从节点读取数据


## 分析复制集状态

```
use admin
```

```
db.runCommand({replSetGetStatus:1})
```

等同于

```
rs.status()
```

- 
- A replica set can have up to [50 members](#) but only 7 voting members.
  - 在副本集的环境中，要是所有的Secondary都宕机了，只剩下Primary。最后Primary会变成Secondary，不能提供服务。
  - 一个复制集中可设置50个成员，但只有7个投票成员（包括primary），其余为非投票成员（Non-Voting Members）。非投票成员是复制集中数据的备份副本，不参与投票，但可以被投票或成为主节点。