mongoing 中文社区 | IT大咖说 知识共享平台

中國東方航空 CHINA EASTERN SKYTEAM

MongoDB在东航的应用实践

东航信息部西北分中心
彭长华

信 息 部
IT SOLUTION

About Me


```
{
    "_id" : ObjectId("5b0e640e7007c9db7eb6c834"),
    "姓名" : "彭长华",
    "公司" : "东方航空",
    "部门" : "信息部西北分中心研发分部",
    "职位" : "技术总监",
    "软件研发经验" : "11年",
    "mongoDB使用经验" : "4年"
}
```

目录

中國東方航空 CHINA EASTERN

1 东航应用场景

中國東方航空 CHINA EASTERN SKYTEAM

信 息 部
IT SOLUTION

1.1 东航的应用场景

中國東方航空 CHINA EASTERN

东方航空于2014年初就开始使用MongoDB，目前有很多生产系统在使用。

航班搜索引擎

采用复制集架构，支持日均**3000W+**的航班查询，以及**2亿条**的库存数据及运价缓存， **99%以上**查询效率低于**200ms**。

旅客实时服务平台

采用复制集架构，支持日均上千万的数据处理量，和**3亿次**的服务查询量，涉及到高并发读写。

Customer系统

基于旅客大数据分析从不同角度计算各个旅客的价值，采用分片架构支持**20亿**条数据存储量。

信 息 部 IT SOLUTION

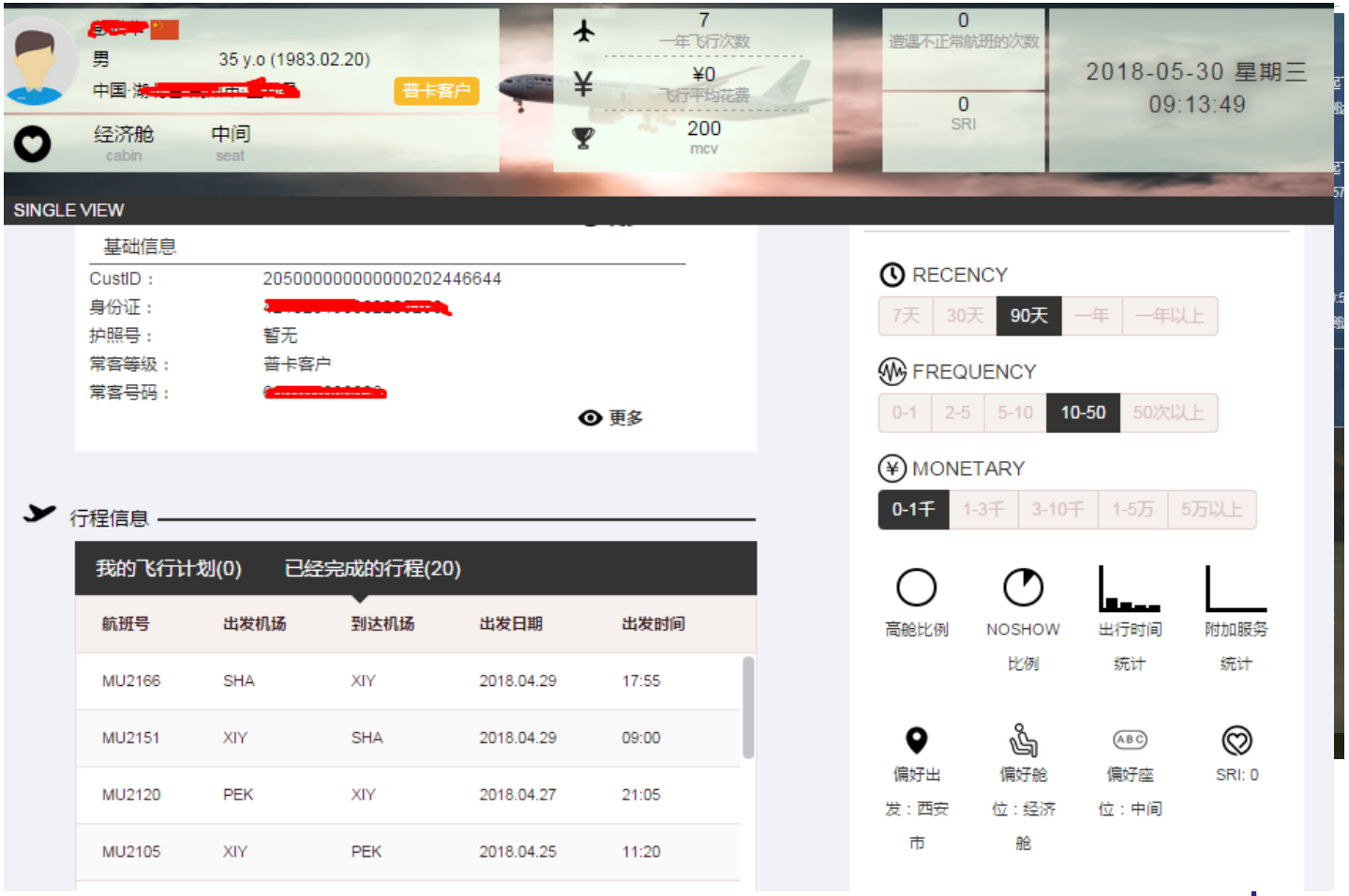
1.2 东航的应用场景

中國東方航空 CHINA EASTERN

1 航班搜索引擎

2 旅客实时服务平台

3 Customer系统



信 息 部 IT SOLUTION

2 部署架构选择

中國東方航空 CHINA EASTERN

信 息 部
IT SOLUTION

2.1.部署架构一（复制集）

项目类型

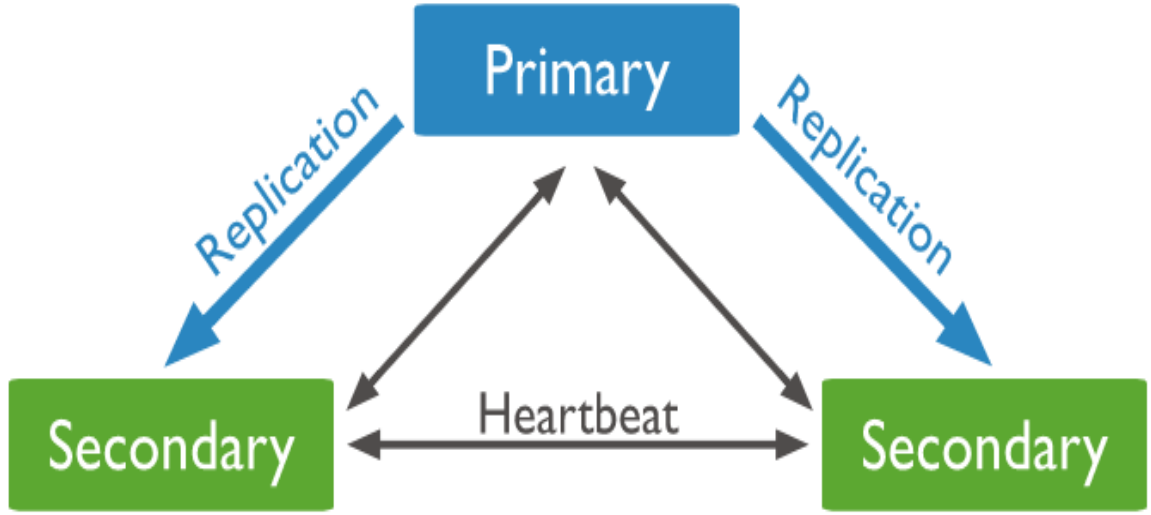
航班搜索引擎、旅客实时服务平台

数据量

支持千万级的数据处理量，和上亿次的服务查询量，99%以上查询效率低于200ms

硬件配置

3台 64 核 512G SSD硬盘



服务器的增加能否提高集群的读写处理能力？

2.2.部署架构二（分片）

项目类型

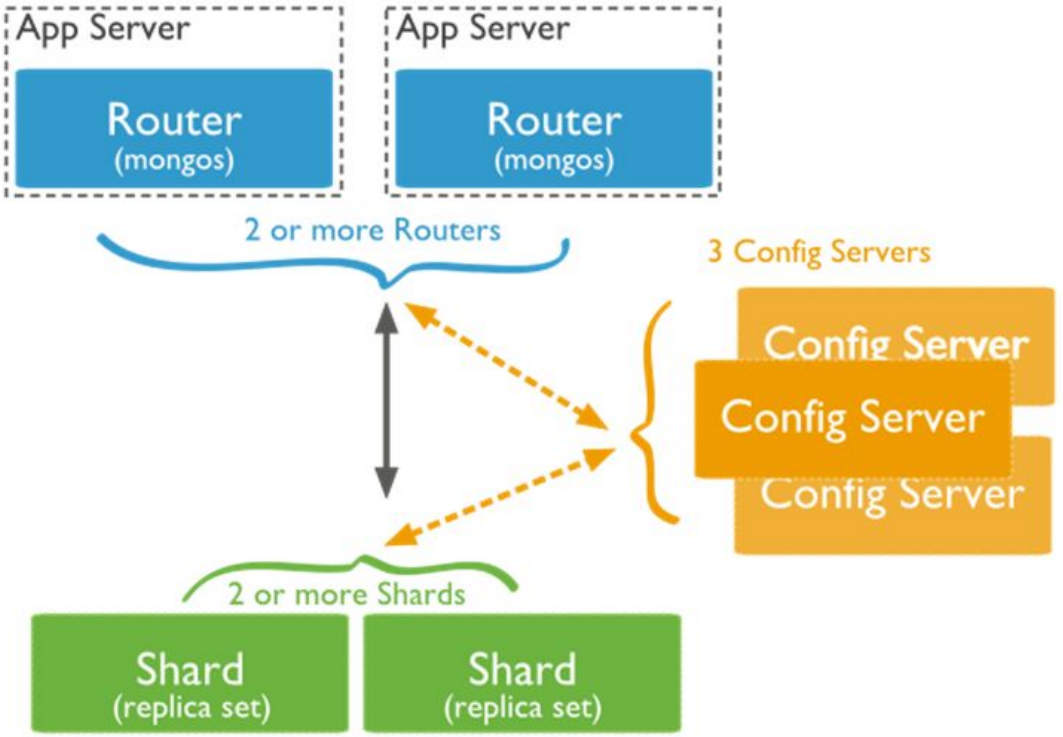
Customer系统

数据量

用户数8000W+ ，数据总量有20亿+，同时数据是一个不断增长的过程。

硬件配置

6 台 8核 16G 500G，二个分片。



3 设计模式

中國東方航空 CHINA EASTERN

信 息 部
IT SOLUTION

3.1.设计原则

关注点	遵循原则	限制条件
• 应用需要什么数据。 • 应用如何读取数据，采用哪些条件进行查询，以及查询频率。 • 应用如何写数据，写入频率。	• 为性能优化，而不是空间。 • 为最重要场景优化，不求面面俱到。	• 单个文档最大不能超过 16M。 • ACID事务：3.X版本原子更新仅限于单文档 • 数组不宜无限制频繁增长。

3.2. 1对1

场景 旅客某天出行的订座、出票、值机、安检和登机等信息

优点 执行一次查询就可以获得旅客的相关信息

缺点 无法像操作独立文档那样来操作订座等信息。必须首先操作（比如查询）旅客文档后，才有可能继续操作订座或值机信息。

3.3 1对N

场景 查询航班下多个舱位的价格

优点 执行一次查询就可以获得航班下所有舱位价格

缺点 无法像操作独立文档那样来操作舱位信息,必须首先操作（比如查询）航班文档后，才有可能继续操作舱位信息

内嵌所有舱位价格

```
"flightInfo" : {
    "carrier" : "MU",
    "flightNo" : "9201",
    "oriEng" : "SHA",
    "desEng" : "XIY",
    "flightDt" : ISODate("2018-07-23T
},
"fareInfo" : [
    {
        "clas_tp" : "C",
        "price" : "2220",
    },
    {
        "clas_tp" : "J",
        "price" : "4310",
    },
    {
        "clas_tp" : "F",
        "price" : "5390",
    },
    {
        "clas_tp" : "Y",
        "fd_price" : "1390",
        "price" : "1390",
        "promocode" : ""
    },
    {
        "clas_tp" : "Z",
        "price" : "450",
    }
```

3.4. 1对NN

场景

查询飞机上所有零部件信息

```
{
    "actype" : "A320",
    "tailNO" : "B2306",
    "productDate" : ISODate("2017-03-02T00:00:00.000+08
    "manufacturer" : "Airbus Corp",
    "parts" : [
        ObjectId("5ad170ec71f8ebc28c657dc4"),
        ObjectId("5ad170ff71f8ebc28c657dc5"),
        ObjectId("5ad170ff71f8ebc28c6
    ]
}
```

飞机信息

```
{
    "_id" : ObjectId("5ad170ec71f8ebc28c657dc4"),
    "partno" : "123-aff-456",
    "name" : "#4 cabin",
    "qty" : 1,
    "cost" : 0.94,
    "price" : 113.99
}
{
    "_id" : ObjectId("5ad170ff71f8ebc28c657dc5"),
    "partno" : "123-aff-457",
    "name" : "#4 cockpit",
    "qty" : 1,
    "cost" : 1,
    "price" : 113.99
}
```

零部件信息

```
var p = db.getCollection('aircraft').findOne({"tailNO":"B2306"});
r= db.parts.find({_id:{$in: p.parts}}) ;
print(r)
```

3.5. 1对NNN

场景 统计每台服务器产生的大量日志。

```
{
    "_id" : ObjectId("5ad1790c71f8ebc28c657dc7"),
    "name" : "xb.ceair.com",
    "ipaddr" : "127.18.80.66"
}
{
    "_id" : ObjectId("5ad6d99c68a40ed768c610ac"),
    "name" : "xa.ceair.com",
    "ipaddr" : "127.18.80.67"
}
```

服务器信息

```
"_id" : ObjectId("5ad1799c71f8ebc28c657dc8"),
"hostId" : ObjectId("5ad1790c71f8ebc28c657dc7")
"createTime" : ISODate("2018-03-28T17:42:41.382
"level" : "ERROR",
"message" : "not enough disc space"

"_id" : ObjectId("5ad17a4a71f8ebc28c657dc9"),
"hostId" : ObjectId("5ad179
"createTime" : ISODate("2018-
"level" : "ERROR",
"message" : "out of memeory"
```

日志信息

```
var  h = db.getCollection('hosts').findOne({ "name" : "xb.ceair.com"})
var logs = db.log.find({"hostId" : h._id})
print(logs)
```

3.6.设计模式比较

总结

- 集合设计由业务驱动，非数据驱动
- 优先考虑内嵌，忘掉关系型数据库范式设计
- 适当的冗余设计

关注点	引用设计	内嵌设计
更新效率	最高	最低
查询效率	最低	最高
适合子文档大小	子文档大	子文档小
子文档增幅	大	小
事务性	无跨表事务(3.X版本)	支持
适用场合	一对非常多， 多对多	一对一，一对很少且不需要单独访问内嵌内容

4 性能监控

中國東方航空 CHINA EASTERN SKYTEAM

信 息 部
IT SOLUTION

4.1. mongostat

./mongostat --host 127.0.0.1 –port 27117

```
insert query update delete getmore command dirty used flushes vsize  res qrw arw net_in net_out conn set repl                time
    *0    *0     *0     *0       0     2|0  0.0% 0.8%       0 1.57G 142M 0|0 1|0   158b    59.8k   16 rs1  PRI May 11 10:01:45.672
    *0    *0     *0     *0       0     4|0  0.0% 0.8%       0 1.57G 142M 0|0 1|0   686b    60.8k   16 rs1  PRI May 11 10:01:46.671
    *0    *0     *0     *0       1     3|0  0.0% 0.8%       0 1.57G 143M 0|0 1|0  2.42k    61.5k   16 rs1  PRI May 11 10:01:47.674
    *0    *0     *0     *0       0     4|0  0.0% 0.8%       0 1.57G 143M 0|0 1|0   687b    60.9k   16 rs1  PRI May 11 10:01:48.671
    *0    *0     *0     *0       0     2|0  0.0% 0.8%       0 1.57G 143M 0|0 1|0   158b    59.7k   16 rs1  PRI May 11 10:01:49.671
    *0    *0     *0     *0       0     3|0  0.0% 0.8%       0 1.57G 143M 0|0 1|0   685b    60.7k   16 rs1  PRI May 11 10:01:50.672
    *0    *0     *0     *0       0     2|0  0.0% 0.8%       0 1.57G 143M 0|0 1|0   158b    59.8k   16 rs1  PRI May 11 10:01:51.671
    *0    *0     *0     *0       3     7|0  0.0% 0.8%       0 1.57G 143M 0|0 1|0  5.23k    64.8k   16 rs1  PRI May 11 10:01:52.672
    *0    *0     *0     *0       0     1|0  0.0% 0.8%       0 1.57G 143M 0|0 1|0   157b    59.7k   16 rs1  PRI May 11 10:01:53.672
    *0    *0     *0     *0       0     4|0  0.0% 0.8%       0 1.57G 143M 0|0 1|0   686b    60.8k   16 rs1  PRI May 11 10:01:54.672
insert query update delete getmore command dirty used flushes vsize  res qrw arw net_in net_out conn set repl                time
```

每秒执行getmore次数

缓存中无效数据所占的百分比

正在使用的缓存百分比

指checkpoint的触发次数在一个轮询间隔期间

虚拟内存使用量，单位是MB

物理内存使用量，单位是MB

集群名

当前已经建立的连接数。

发送端网络速率，单位是bytes

接收端网络速率，单位是bytes

已激活active的读/写次数

队列中waiting的读写次数

4.2. mongotop

./mongotop --host 127.0.0.1 --port 27117

```
                    ns    total    read    write    2018-05-11T10:47:33+08:00
             test.test2     39ms     0ms     39ms
          local.oplog.rs     38ms    38ms      0ms
    admin.$cmd.aggregate      0ms     0ms      0ms
       admin.system.keys      0ms     0ms      0ms
      admin.system.roles      0ms     0ms      0ms
    admin.system.version      0ms     0ms      0ms
config.system.sessions      0ms     0ms      0ms
   config.transactions      0ms     0ms      0ms
              local.me      0ms     0ms      0ms
local.replset.election      0ms     0ms      0ms

                    ns    total    read    write    2018-05-11T10:47:34+08:00
```

数据库名及集合名

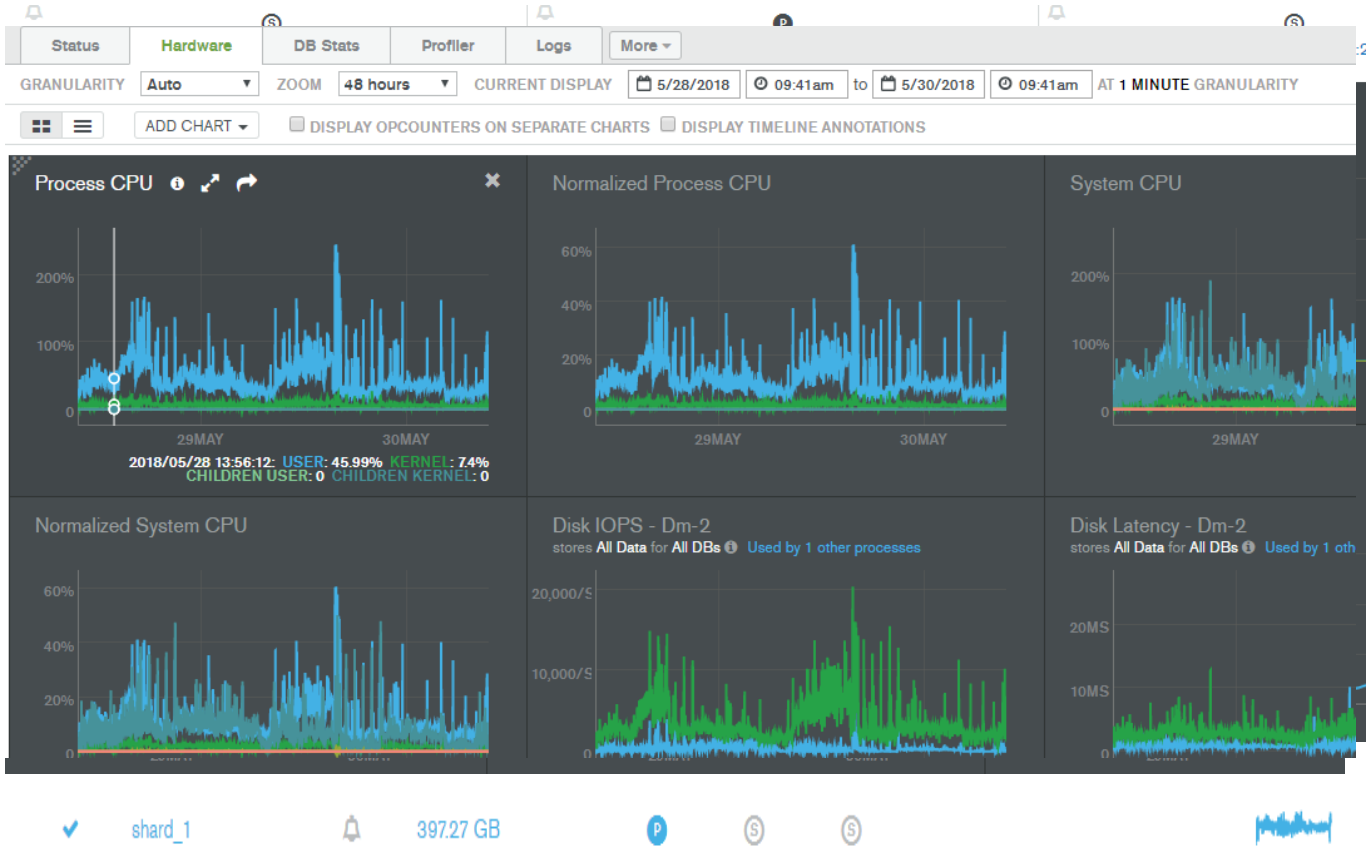
花费在集合上的总时间

执行写操作花费的总时间

执行读操作花费的总时间

4.3 Ops Manager

- 监控集群运行状态，如连接数、每秒查询次数等mongostat相关参数
- 监控分片使用状态
- 监控服务器CPU、磁盘空间、IO状态



4.4 日志分析

1）启动或禁用分析器

开启 ：db.setProfilingLevel(1,100)

禁用 ：db.setProfilingLevel(0)

当前分析器状态：db.getProfilingStatus()

2) 慢查询

db.system.profile.find()

获取长于200ms查询：

db.system.profile.find({ “millis” :{$gt:200}}).sort({ “millis” :-1})

3) 日志分析工具

mtools 是一组非常好用的 MongoDB 日志分析工具

https://github.com/rueckstiess/mtools

4.5 执行计划

解析方式：queryPlanner, executionStats和allPlansExecution

```
db.inventory.find(
{"flightInfo.oriEng" : "XIY",
"flightInfo.desEng" : "SHA",
"flightInfo.flightDt" : ISODate("2018-07-
20T00:00:00.000+08:00")
}).explain("executionStats")
```

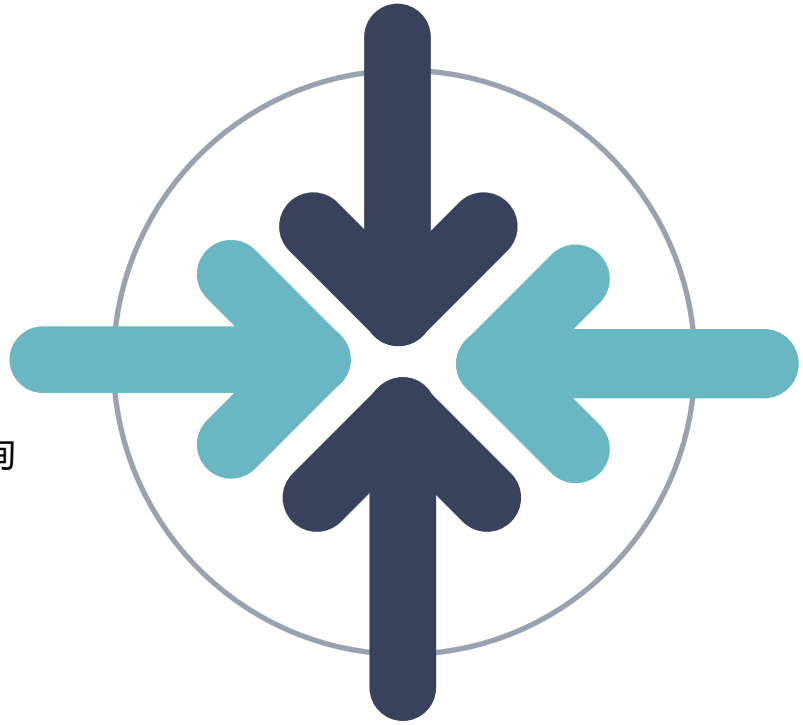
5 性能优化

中國東方航空 CHINA EASTERN SKYTEAM

信 息 部
IT SOLUTION

5.1. 索引优化

1. **提前规划好索引与后台执行**
   如果在对有数据的集合建索引必须后台执行。
2. **创建合适的索引**
3. **组合索引字段顺序**
   组合索引中的字段的顺序：精确匹配—排序—范围查询
4. **关注应用读/写(read/write) 比率**
5. **确定所有的索引都在RAM中**



5.2 Kill掉慢请求操作

- 客户端发起的耗时请求遍历集合、建索引、mapreduce、aggregation等，主动断开连接后，后端的请求仍然执行。

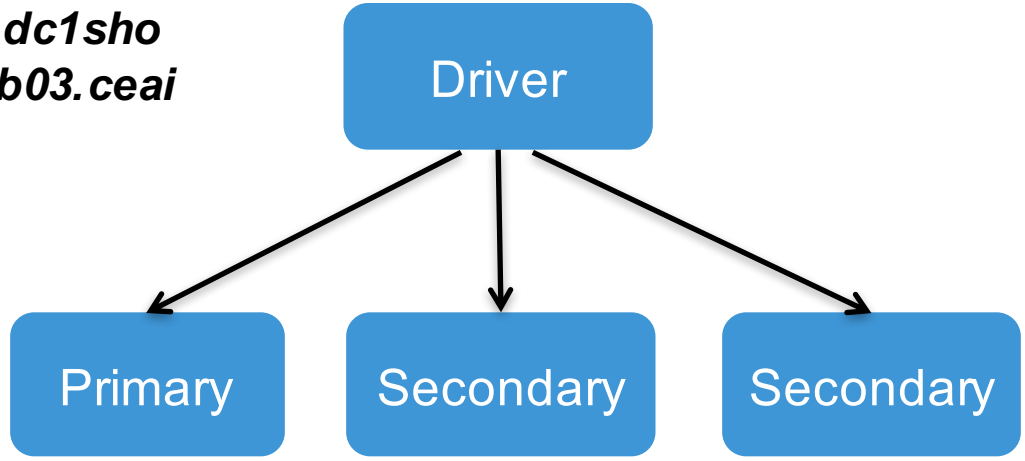
- killOp通常需要与currentOp组合起来使用；
- 先根据currentOp查询到请求的opid，然后根据opid发送killOp的请求。

突然正在运行的航班搜索服务接口超时严重？

5.3.正确连接复制集

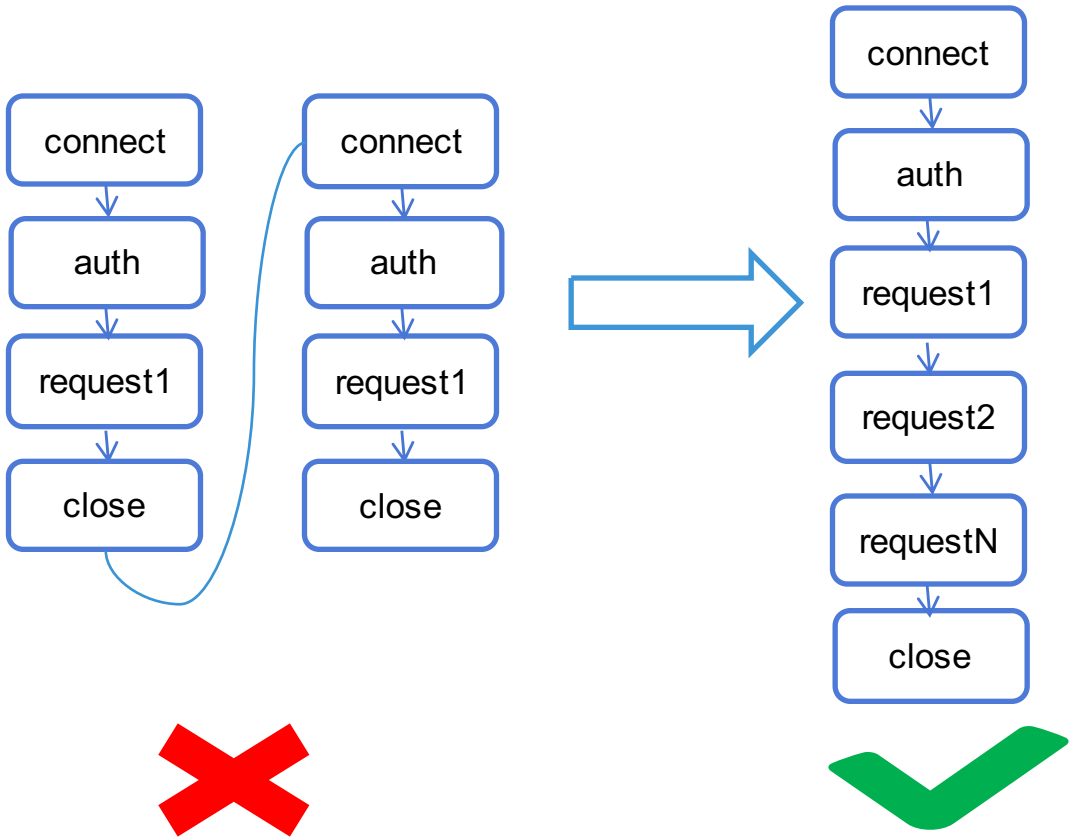
驱动指定多个节点连接复制集，Primary有故障时可自动重新连接。

```
<mongo:mongo id="routeCacheMongo" replica-set="dc1shopmdb02.ceair.com:27117,dc1shopmdb01.ceair.com:27117,dc1shopmdb03.ceair.com:27117">
```



5.4.使用长连接

避免使用短连接，多次创建连接关闭连接等操作。

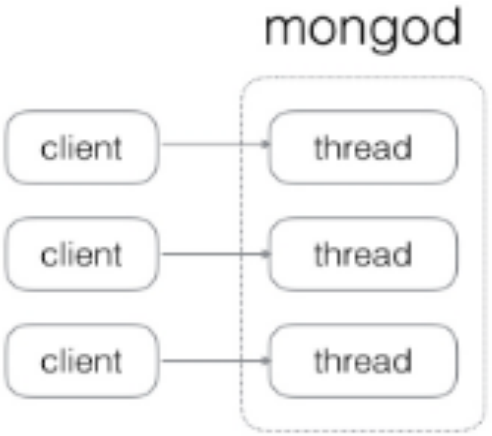


5.5 配置合理的连接数

1) 合理设置 thread per connection服务模型。

- 每个线程需要1M的栈空间
- 大量连接时，线程切换开销大

2) Driver通过Connection String URI的maxPoolSize参数来配置连接池大小。



5.6.减少返回不必要的内容

1).使用投射 （ projection ）来减少返回的内容

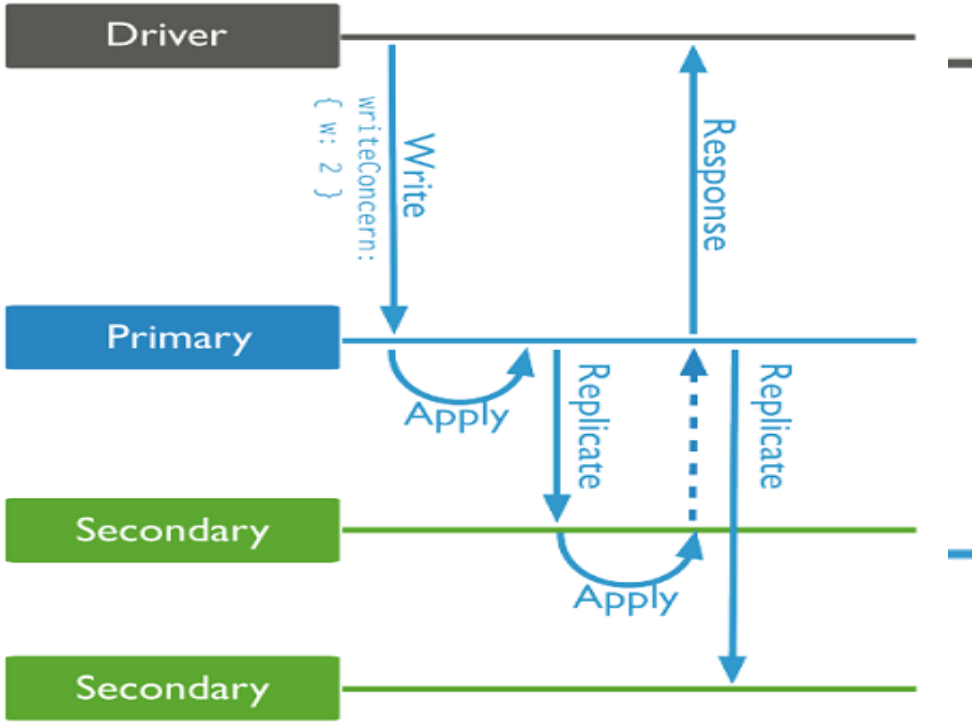
```
db.getCollection('inv').find({ “"flightInfo.oriEng":"XIY","flightInfo.desEng":"SHA"” }
,{"flightInfo":1,"_id":0})
```

2).使用skip或limit，尽可能的减少文档输出。

```
db.getCollection('inv').find({}).skip(10).limit(2)
```

5.7 正确使用写关注

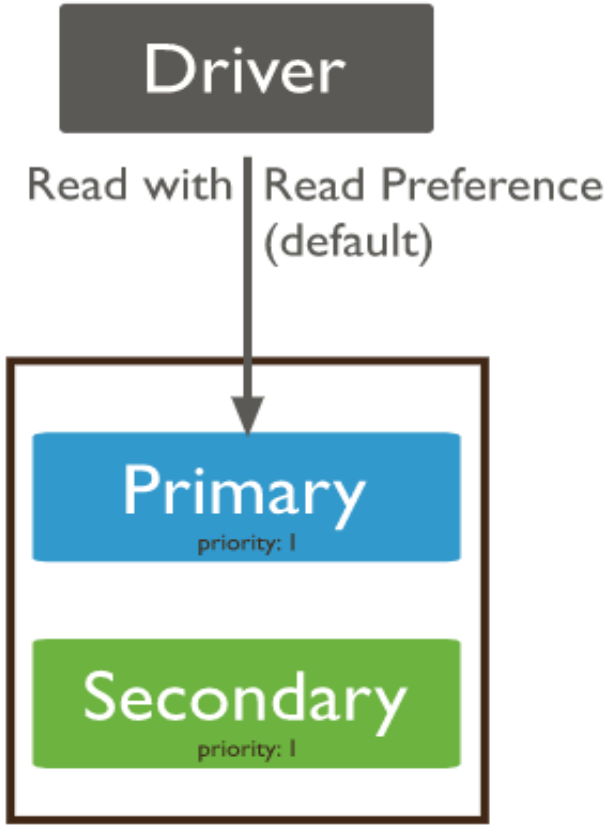
关注点	含义
{w:0}	写入不需要Server返回响应。
{w:1}	3.x driver默认行为。要求确认操作已经传播到指定的单个mongod实例或副本集主实例，后台默认每100ms保证日志刷盘，每60s数据刷盘。
{w:1，j:true}	写入到主实例，并且日志刷盘后向客户端确认。
{w:”majority”}	写入到已经传递到绝大多数投票节点以及主节点后进行应答。



在进行写入操作时如何权衡性能与一致性？

5.8 正确使用读选项设置

关注点	含义
primary	主节点，默认模式，读操作只在主节点，如果主节点不可用，报错或者抛出异常。
primaryPreferred	首选主节点，大多情况下读操作在主节点，如果主节点不可用，如故障转移，读操作在从节点。
secondary	从节点，读操作只在从节点， 如果从节点不可用，报错或者抛出异常。
secondaryPreferred	首选从节点，大多情况下读操作在从节点，特殊情况（如单主节点架构）读操作在主节点。
nearest	最邻近节点，读操作在最邻近的成员，可能是主节点或者从节点。



中國東方航空 CHINA EASTERN 信 息 部 IT SOLUTION

关爱 创新 协作 承诺

谢谢大家！