MongoDB
开发实践

唐峰

锦木信息

锦木信息

MongoDB partner，　提供MongoDB订阅、咨询、技术支持服务。

Shanghai Jinmu Information Technology Co., Ltd.

Jinmu is a MongoDB reseller and services partner based on Shanghai. They provide NoSQL database, big data software, and consulting services solutions. Jinmu supports the full project lifecycle from initial design to production support. Industry focuses include financial services, transportation and retail. Their mission is to provide top-tier data solutions to customers.

主要客户

中國東方航空 CHINA EASTERN

Volkswagen

COSCO SHIPPING

太平洋保险 CPIC

中国太平 CHINA TAIPING

GLOBALFOUNDRIES

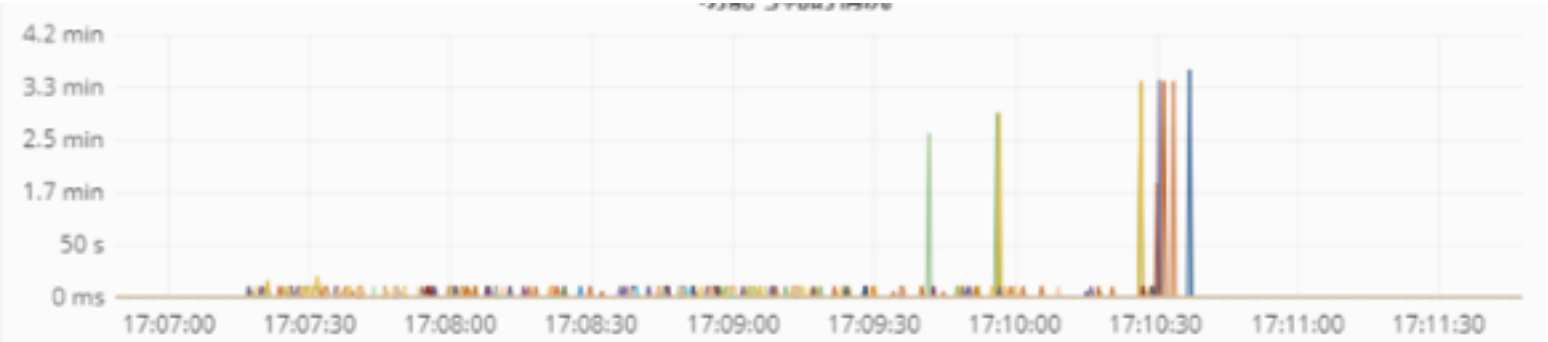
CHINA FOREIGN EXCHANGE TRADE SYSTEM 中国外汇交易中心 全国银行间同业拆借中心 NATIONAL INTERBANK FUNDING CENTER

国泰君安证券 GUOTAI JUNAN SECURITIES

议程

1. 连接设置
2. 性能相关
3. 模型设计
4. 架构相关

连接设置



问题： 1个mongos 不可用，应用整体延迟增高，性能明显下降

配置： **connection string:** mongodb://mongos1:port,mongos2:port,mongos3:port

mongodb version： 3.4.13

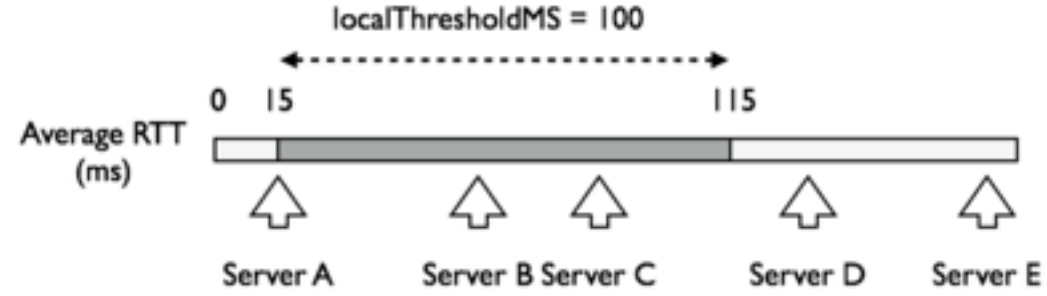
driver : C 1.5.0

连接设置

原理:

serverSelectionTimeoutMS 30s

localThresholdMS 15ms



Servers A, B and C are in the latency window

C Driver's

单线程模式

```
mongoc_client_t *client = mongoc_client_new (
    "mongodb://hostA,hostB/?replicaSet=my_rs");
```

连接池模式

```
mongoc_client_pool_t *pool = mongoc_client_pool_new (
    "mongodb://hostA,hostB/?replicaSet=my_rs");

mongoc_client_t *client = mongoc_client_pool_pop (pool);
```

议程

1. 连接设置
2. **性能相关**
3. 模型设计
4. 架构相关

索引

- 索引的选择性

索引

- 索引的选择性
- 索引前缀

{ a: 1, b: 1, c: 1 }

filter
- { a: 1}
- { a: 1, b: 1}
- { a: 1, b: 1, c: 1 }

sort
- sort.{ a: 1}
- sort.{ a: 1, b: 1}
- sort.{ a: 1, b: 1, c: 1 }

({ a: 5 }).sort({ b: 1, c: 1 })

({ a: { $gt: 2 } }).sort({ c: 1 })

索引

- 索引的选择性
- 索引前缀
- 复合索引：等值 排序 范围

db.restaurants.find({ “avgcost”: { $lte: ‘300’}, cuisine: ‘Sushi’ }).sort({ stars: 1 })

db.restaurants.createIndex({ “cuisine”: 1, “stars”: 1, “avgcost”: 1 })

等值 排序 范围

索引

- 索引的选择性
- 索引前缀
- 复合索引：等值 排序 范围
- 索引创建

1. foreground：快 锁库
2. Background：慢 mongoshell不会退出
3. 从节点启动到standalone模式，手工创建索引，再加入复制集

索引

- 索引的选择性
- 索引前缀
- 复合索引：等值 排序 范围
- 索引创建
- $indexStats

```
db.collection.aggregate( [
  { $indexStats : { } },
  { $project : { key: 1, "accesses.ops": 1 } }
] ).pretty
```

CRUD

- update

```
db.collection.update(<filter>, <update>, <options>)
```

```
db.foo.find()
db.foo.update({a : 5, b : 5}, { $set : {c : 10}}, {upsert : true})
db.foo.find()
{ "_id" : ObjectId("5a014bab2e7289371f61d474"), "a" : 5, "b" : 5, "c" : 10 }
```

multi : 默认false

CRUD

- update
- Bulk Write

```
db.collection.bulkWrite(
    [ <operation 1>, <operation 2>, ... ],
    {
        writeConcern : <document>,
        ordered : <boolean>
    } )
```

- order顺序插入数据，中间出现错误则退出
- Unorder并行插入，遇到错误不会退出

CRUD

- update
- Bulk Write
- 写关注

```
{ w: <value>, j: <boolean>, wtimeout: <number> }
```

w : 数据写入节点后向客户端确认

- w: 1 默认
- w: 0
- "majority”

J： 确认写操作写入journal

wtimeout: 写入超时时间，仅w大于1时有效

CRUD

- update
- Bulk Write
- 写关注
- count

```
db.collection.count()
db.collection.find( { a: 5, b: 5 } ).count()
```

```
db.collection.aggregate( [
        { $match: <query condition> },
        { $count: "myCount" }
] )
```

CRUD

- update
- Bulk Write
- 写关注
- count
- $inc

```
$ ./mongo
> c = db.uniques_by_hour;
> c.find();
> cur_hour = new Date("Mar 05 2009 10:00:00")
> c.ensureIndex( { hour : 1, site : 1 } );
> c.update( { hour : cur_hour, site : "abc" },
            { $inc : { uniques:1, pageviews: 1} },
            { upsert : true } )
> c.find();
{"_id" : "49aff5c62f47a38ee77aa5cf" ,
 "hour" : "Thu Mar 05 2009 10:00:00 GMT-0500 (EST)" ,
 "site" : "abc" , "uniques" : 1 ,
 "pageviews" : 1}
> c.update( { hour : cur_hour, site : "abc" },
            { $inc : { uniques:1, pageviews: 1} },
            { upsert : true } )
> c.find();
{"_id" : "49aff5c62f47a38ee77aa5cf" ,
 "hour" : "Thu Mar 05 2009 10:00:00 GMT-0500 (EST)" ,
 "site" : "abc" , "uniques" : 2 , "pageviews" : 2}
> c.update( { hour : cur_hour, site : "abc" },
            { $inc : { uniques:0, pageviews: 1} },
            { upsert : true } )
> c.find();
{"_id" : "49aff5c62f47a38ee77aa5cf" ,
 "hour" : "Thu Mar 05 2009 10:00:00 GMT-0500 (EST)" ,
 "site" : "abc" , "uniques" : 2 , "pageviews" : 3}
```

http://blog.mongodb.org/post/171353301/using-mongodb-for-real-time-analytics

议程

1. 连接设置
2. 性能相关
3. **模型设计**
4. 架构相关

模型设计

内嵌

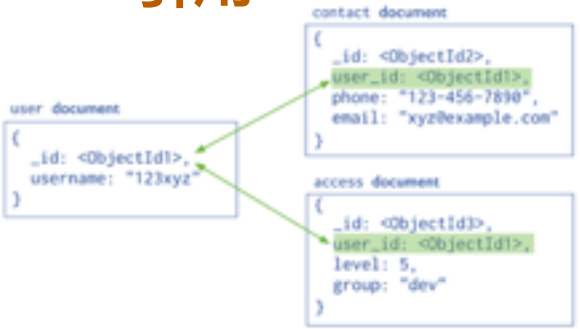
```
{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
              phone: "123-456-7890",
              email: "xyz@example.com"
            },
  access: {
             level: 5,
             group: "dev"
           }
}
```

Embedded sub-document

Embedded sub-document

- 富JSON、包含关系、一对多
- 更快的查询性能
- 关联数据原子更新

引用



- 内嵌大量数据重复且查询无法受益
- 多对多
- 大的层级结构

模型设计

```
#users
{
        "_id" : "Messi",
        "fullname" : "LeoMessi",
        "country" : "Argentina",
        "followers" : ["Neymar", "C Ronaldo"],
        "following" : ["Neymar", "C Ronaldo"]
}
```

- **关注、粉丝信息内嵌在用户集合**
  - ✓ 结构清晰
  - ✓ 查询快速
- **但是……**
  - 用户集合增长
  - 粉丝数量剧增（>16M）
  - 数组操作代价高

模型设计

```
#followers
{
        "_id" : "ObjectId(…)",
        "from" : "Neymar",
        "to" : "Messi"
}
{
        "_id" : "ObjectId(…)",
        "from" : "C Ronaldo",
        "to" : "Messi "
}
{
        "_id" : "ObjectId(…)",
        "from" : "Messi",
        "to" : "C Ronaldo"
}
{
        "_id" : "ObjectId(…)",
        "from" : " Messi",
        "to" : " Neymar"
}
```

梅西关注了谁：

```
db.followers.find({ from : "Messi"},{_id:0, to:1})
```

```
#索引
{
        "v" :  1,
        "key" : { "from" : 1, "to" : 1 },
        "name" : "from_1_to_1 "
}
```

梅西有多少粉丝：

```
db.followers.find({ to :"Messi"}).count()
```

```
#索引
{
        "v" :  1,
        "key" : { "to" : 1, "from" : 1 },
        "name" : "to_1_from_1 "
}
```

模型设计

分片

```
#索引
{
        “v” :  1,
        “key" : { “from” : 1, “to” : 1 },
        ”name" : ”from_1_to_1 "
}


{
        “v” :  1,
        “key" : { “to” : 1, “from” : 1 },
        ”name" : ”to_1_from_1 "
}
```

使用from分片

✓ 某人关注了谁

✓ targeted to sh

```
#followers
{
        “_id” : ”ObjectId(…)",
        ”from" : "Neymar",
        ”to" : "Messi"
}
```

- 某人被谁关注了
- scatter-gather to all shards

```
#following
{
        “_id” : ”ObjectId(…)",
        ”from" : "Neymar",
        ”to" : "Messi"
}
```

议程

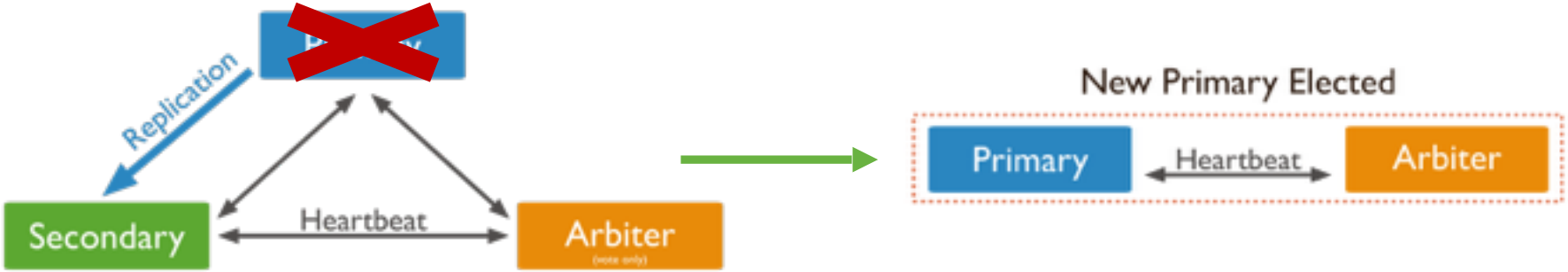
1. 连接设置
2. 性能相关
3. 模型设计
4. **架构相关**

系统配置

版本	数据一致性	网络安全	硬件配置	系统配置	架构
OS	Journal配置	认证与权限控制	内存大小和WT Cache	内核和文件系统选择	复制集
MongoDB	脏读配置	通讯加密	压缩和加密对硬件的要求	启用NTP	分片集
Driver	写关注	存储加密	SSD和readahead	禁用atime	压缩算法选择
		网络监听	禁用NUMA	ulimit配置	
		防火墙	SWAP空间	禁用Huge Page	
		审计	RAID配置	禁用SELinux	
		MongoDB独立账户	远程文件系统	TLS/SSL库	
		禁用服务器端脚本	分散存储	内存过量（VMWare）	
		连接池大小		TCP KeepAlive	

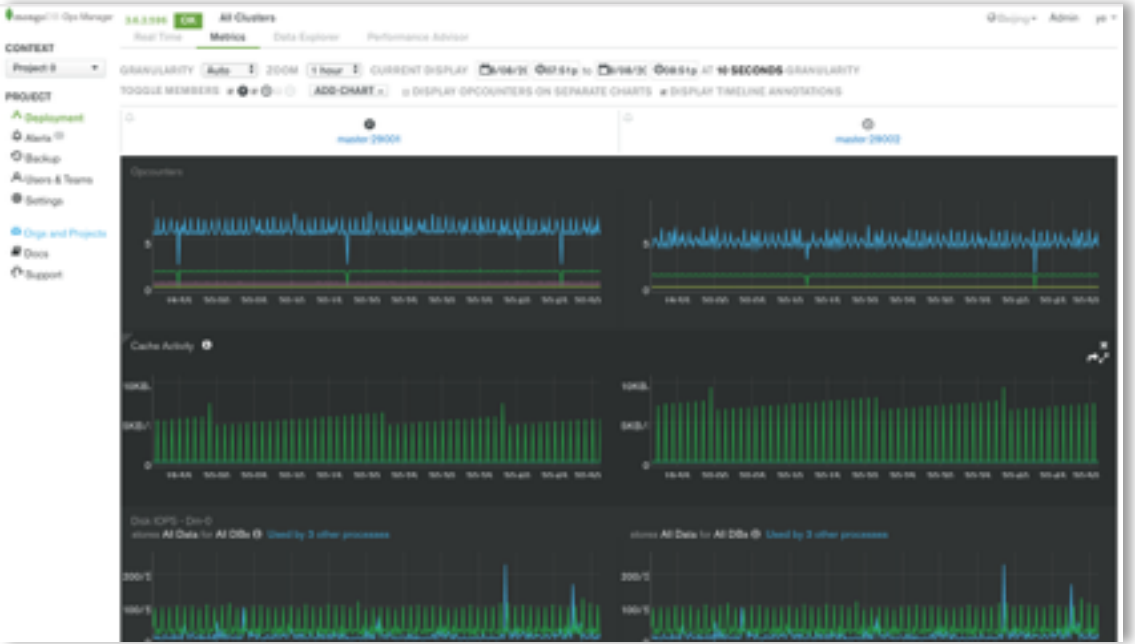
https://docs.mongodb.com/manual/administration/production-notes
https://docs.mongodb.com/manual/tutorial/
https://docs.mongodb.com/manual/faq/diagnostics

复制集



- write concern : majority
- 分片：chunk迁移受阻
- 分布式锁

Ops Manager



减少95%的运维操作

- 一个单击可以完成部署、扩展、升级及更多的管理员任务
- 图形化监控面板，支持100多关键指标
- 快速任意点备份及恢复，并支持分片集群

招人

更多问题?