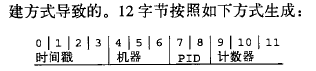
mongodb的自动生成的唯一id是根据：



因此可以通过这个id后的当时的时间戳;

> db.tt.find()

{ "\_id" : ObjectId("58037b045a808d2420e8fa53"), "a" : 1 }

> ObjectId("58037b045a808d2420e8fa53").getTimestamp() //得到建立时的时间戳

ISODate("2016-10-16T13:05:08Z")

[如何让mongoDB不排序](http://bbs.csdn.net/topics/380197294" \t "http://bbs.csdn.net/forums/_blank" \o "如何让mongoDB不排序)？

Mongodb再插入时如果没有指定id值的话会自动生成，而排序默认可能按照这个排序返回结果，而此值是根据时间戳机器码进程值和随机数决定的，因此对于同一台机器而言（不同的机器插入的数据也应该一样），只是时间的不同而排序又是二进制排序（一个字节一个字节的排序），因此返回的结果还是按照插入时的顺序。

负载均衡：

通过./mongod --dbpath master --port 10000 --master启动一个mongodb实例。  
2、通过./mongod --dbpath slave --port 10001 --slave启动另一个mongodb实例，与上一个实例在同一台服务器上。  
3、在slave的shell中添加了一个source是localhost:10000。  
  
经过以上步骤，slave会自动同步master上的数据。

Mongdo数据库快照;

1、数据库快照是数据库（称为“源数据库”）的只读静态视图。在创建时，每个数据库快照在事务上都与源数据库一致。在创建数据库快照时，源数据库通常会有打开的事务。在快照可以使用之前，打开的事务会回滚以使数据库快照在事务上取得一致。

2、内存快照，某一时刻内存数据状态

BSON只支持UTF8编码。

Mongodb查询特殊字符要转义

> db.zhengze.find({"title":/吗\(\)你/i}) //括号要转义

{ "\_id" : ObjectId("5809bee9219b0cb57c5ead2a"), "title" : "你好吗()你好吗" }

Mongodb的数据大小限制：

由于在MongoDB中，1.7版本之前，BSON对象的大小只有4MB的限制，1.7-1.8版本，大小限制是16MB

Mongodb复制一个数据库中的数据到另一个数据库：

语法：db.copyDatabase(fromdb, todb, fromhost, username, password)

> db.copyDatabase("MoliFresh","MoliFresh","192.168.100.249","")

{ "ok" : 1 }

Mongodump命令帮助：

D:\mongo\bin>mongodump /version

mongodump version: r3.2.9

git version: 4a4e7d30773b28cf66f75e45bc289a5d3ca49ddd

Go version: go1.6.2

os: windows

arch: amd64

compiler: gc

OpenSSL version: OpenSSL 1.0.1p 9 Jul 2015

删除单个文档的一个属性：

> delete d.\_id

> d

{

"\_t" : "StarNode",

"Name" : null,

"Desc" : null,

"ImageUrl" : null,

"PageUrl" : null,

"TimeCreated" : ISODate("0001-01-01T00:00:00Z"),

"TimeModified" : ISODate("2016-10-13T09:11:56.634Z"),

"Status" : 0,

"Deleted" : 0,

"Locked" : 1,

"OriginalId" : null,

"NodeType" : 0,

"Relations" : [ ],

"Sex" : null,

"Country" : null,

"Birthday" : null,

"ForeignName" : null,

"OtherNames" : null,

"Astrology" : null,

"Occupation" : null

}

mongo有快照，在生成静态只读视图时是不阻塞写操作的

mongodb复制集的故障恢复功能是什么原理

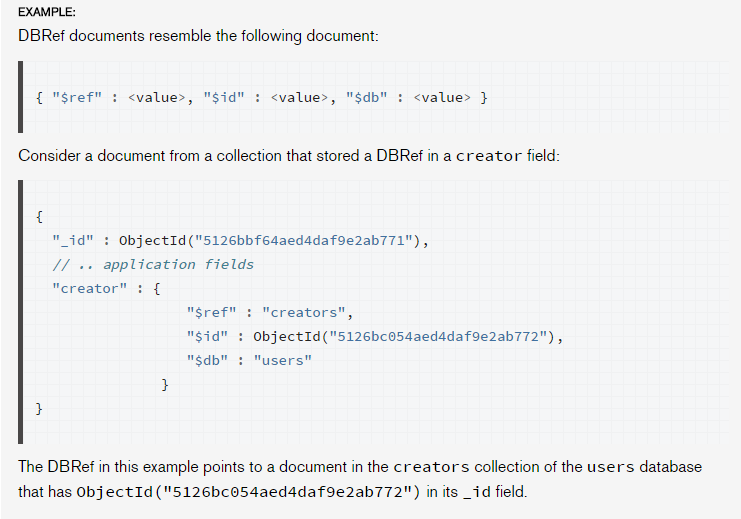
MongoDB的主从节点上都有一个Oplog的集合(collection)。Oplog是一个固定大小的集合，当这个集合写满数据时，新的数据会覆盖掉最老的数据。在MongoDB的主节点上增删改数据后，这个操作会写到主节点的Oplog里。这些写到Oplog里的操作都是幂等的操作，也就是说不管你执行这个操作一次还是多次，得到的结构都是一样的。  
  
假设我们有三个节点的复制集A、B和C。A是主节点，B是从节点，C为仲裁节点。B从A上同步数据。在A宕机后，B成为新的主节点。A再次启动后变成从节点，并尝试从B上同步数据。  
  
在同步数据时，A会检查自己的Oplog里最新的操作的时间戳，在B上找到对应的时间戳的操作然后同步该时间戳后面的操作。但如果A宕机后，B上进行了很多的操作，B的Oplog里老的操作被覆盖掉了。当A再次启动后，A的Oplog里最新的时间戳比B上最老的时间戳还老，说明A已经无法从B上同步数据了，这个时候就会报“too stale to catch up”的错误。这个时候只能从主节点上重新同步所有的数据。

可以查看oplog时间，使用db.getReplicationInfo()，如果超过时间则需要重新复制了。

或者：> db.printReplicationInfo()

{ "errmsg" : "neither master/slave nor replica set replication detected" }

文档关联：



模糊查询是不使用索引

第一，关键字模糊查询肯定慢，无论 code 上是不是有索引，都是全表扫描，不走索引。

相对来讲，主要关联性不强对速度要求高的就可以使用mongodb

Mongodb 如何做到在只保留最新的50条记录，删除掉旧的

MongoDB有两种集合能实现比较类似的功能。  
  
一种是TTL(Time To Live)集合，mongod会自动删除过期的文档。  
http://docs.mongodb.org/manual/tutorial/expire-data/

Data expiration is useful for some classes of information, including machine generated event data, logs, and session information that only need to persist for a limited period of time.



另一种是capped集合，该集合是一个固定大小的集合，当该集合存满数据时，mongod会自动通过覆盖最旧的文档来存放新的文档。  
<http://docs.mongodb.org/manual/core/capped-collections/>

MMS可视化监控mongodb：

<http://blog.csdn.net/zhaowenzhong/article/details/50593798>

Mongodb导入csv数据：

csv文件：user\_csv.dat

category,name,loc  
 "咖啡","咖啡","[70,80]"  
 "茶","茶吧","[70,80]"  
 "tea","hangzhou tea bar","[71,81]"  
 "coffee","hangzhou coffee bar","[71,81]"

导入数据：

D:\mongodb\bin>mongoimport.exe -d test -c mapinfo --type csv --headerline --file

D:\mongodb\data\user\_csv.dat

2016-10-22T10:19:02.304+0800 connected to: localhost

2016-10-22T10:19:02.509+0800 imported 4 documents

> db.mapinfo.find() //注意中文用utf-8

{ "\_id" : ObjectId("580acda2781bcc476820486d"), "category" : "茶", "name" : "茶

吧", "loc" : "[70,80]" }

{ "\_id" : ObjectId("580acda2781bcc476820486e"), "category" : "tea", "name" : "ha

ngzhou tea bar", "loc" : "[71,81]" }

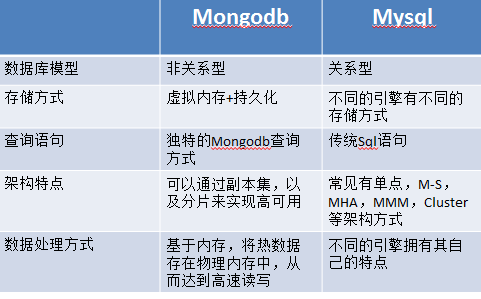
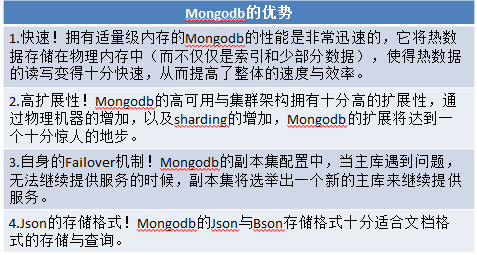
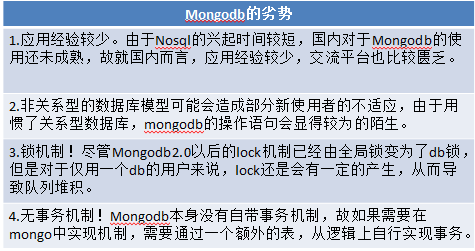
{ "\_id" : ObjectId("580acda2781bcc476820486f"), "category" : "coffee", "name" :

"hangzhou coffee bar", "loc" : "[71,81]" }

{ "\_id" : ObjectId("580acda2781bcc4768204870"), "category" : "咖啡", "name" : "

咖啡", "loc" : "[70,80]" }

Mongodb和mysql(关系型数据库)对比

  
  
  
  
  
  
再来分析下应用场景，  
a.如果需要将mongodb作为后端db来代替mysql使用，即这里mysql与mongodb 属于平行级别，那么，这样的使用可能有以下几种情况的考量：  
(1)mongodb所负责部分以文档形式存储，能够有较好的代码亲和性，json格式的直接写入方便。(如日志之类)  
(2)从data models设计阶段就将原子性考虑于其中，无需事务之类的辅助。开发用如nodejs之类的语言来进行开发，对开发比较方便。  
(3)mongodb本身的failover机制，无需使用如MHA之类的方式实现。  
  
这种情况也是不少的，我手上的游戏就是有nodejs+mongodb的，用户总量也是千万级别的，流水什么还不错。  
  
b.将mongodb作为类似redis ，memcache来做缓存db，为mysql提供服务，或是后端日志收集分析。  
考虑到mongodb属于nosql型数据库，sql语句与数据结构不如mysql那么亲和 ，也会有很多时候将mongodb做为辅助mysql而使用的类redis memcache 之类的缓存db来使用。  
亦或是仅作日志收集分析。

> db.getCollection("test").find()[0]

{

"\_id" : ObjectId("5802258f5a808d2420e8f90e"),

"url" : "www.baidu.com",

"pageviews" : 15

}

查询索引：

> db.cc.ensureIndex({"confData":1})

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"ok" : 1

}

> db.cc.getIndexes()

[

{

"v" : 1,

"key" : {

"\_id" : 1

},

"name" : "\_id\_",

"ns" : "test.cc"

},

{

"v" : 1,

"key" : {

"confData" : 1

},

"name" : "confData\_1",

"ns" : "test.cc"

}

]

mongodb可以导出csv文件

Mongodb的分片

Csdn博客bbs=======1、6-10页已完