sudo firewall-cmd --add-port=27017/tcp --permanent  
sudo firewall-cmd --reload

sudo firewall-cmd --permanent --add-rich-rule "rule family="ipv4" \  
source address="10.10.20.0/24" port protocol="tcp" port="27017" accept"

mongod --port 3333 --dbpath "F:\Database\MongoDB\mongo3\data" --logpath "F:\Database\MongoDB\mongo3\log\mongDB3.log" --replSet test –logappend

分片技术可以解决单表数据量太大以至于单台机器都无法支持的情况，是一种水平扩展技术。其中也会有一些技术难点，比如一致性哈希等。

这里记录一下自己使用MongoDB实现shard功能。机器MacOS，MongoDB版本：3.4.9。

在MongoDB的shard集群中共有三种角色：

（1）config，用于存储配置信息，在3.4版本的MongoDB中，要求config节点必须是一个大于等于3个节点的复制集，这样容错率更高；

（2）routing，用户配置整个shard集群，实现路由功能，是程序与MongoDB的shard集群交互的入口；

（3）shard，存放分片数据的节点。

这里的拓扑为：

由c0，c1，c2组成的复制集作为config节点集，c0：<127.0.0.1：40000>，c1：<127.0.0.1：40001>，c2：<127.0.0.1：40002>

由s0，s1组成的shard，s0：<127.0.0.1：30001>，s1：<127.0.0.1：30002>

由m0组成的routing，m0：<127.0.0.1：50000>

根目录为：/data/shard-project/，在该目录下新建log，config和data目录，分别存放日志，配置和数据。在data目录下分别建立c0，c1，c2，s0，s1目录，m0不需要data目录。

下面分别建立各个节点的配置文件：

c0.conf:

|  |
| --- |
| dbpath=/data/shard-project/data/c0  logpath=/data/shard-project/log/c0.log  logappend=true  port=40000  fork=true  configsvr=true  replSet=cs |

c1.conf:

|  |
| --- |
| dbpath=/data/shard-project/data/c1  logpath=/data/shard-project/log/c1.log  logappend=true  port=40001  fork=true  configsvr=true  replSet=cs |

c2.conf:

|  |
| --- |
| dbpath=/data/shard-project/data/c2  logpath=/data/shard-project/log/c2.log  logappend=true  port=40002  fork=true  configsvr=true  replSet=cs |

s0.conf:

|  |
| --- |
| dbpath=/data/shard-project/data/s0  logpath=/data/shard-project/log/s0.log  logappend=true  port=30001  fork=true  shardsvr=true |

s1.conf:

|  |
| --- |
| dbpath=/data/shard-project/data/s1  logpath=/data/shard-project/log/s1.log  logappend=true  port=30002  fork=true  shardsvr=true |

m0.conf:

|  |
| --- |
| logpath=/data/shard-project/log/m0.log  logappend=true  port=50000  fork=true  configdb=cs/localhost:40000,localhost:40001,localhost:40002 |

然后启动：

s0，s1:

|  |
| --- |
| sudo mongod -f /data/shard-project/config/s0.conf  sudo mongod -f /data/shard-project/config/s1.conf |

c0，c1，c2：

|  |
| --- |
| sudo mongod -f /data/shard-project/config/c0.conf  sudo mongod -f /data/shard-project/config/c1.conf  sudo mongod -f /data/shard-project/config/c2.conf |

接着配置复制集：

登陆c0：

mongo --port 40000

配置复制集，并且指定自己为primary：

config={\_id:"cs",members:[{\_id:0,host:"localhost:40000",priority:1}]}

rs.initiate(config)

 最后提示符会改变，添加c1和c2：

rs.add("localhost:40001")

rs.add("localhost:40002")

最后检查复制集：

|  |
| --- |
| cs:PRIMARY> rs.status()  {  "set" : "cs",  "date" : ISODate("2017-10-03T21:10:19.697Z"),  "myState" : 1,  "term" : NumberLong(1),  "configsvr" : true,  "heartbeatIntervalMillis" : NumberLong(2000),  "optimes" : {  "lastCommittedOpTime" : {  "ts" : Timestamp(1507065009, 1),  "t" : NumberLong(1)  },  "readConcernMajorityOpTime" : {  "ts" : Timestamp(1507065009, 1),  "t" : NumberLong(1)  },  "appliedOpTime" : {  "ts" : Timestamp(1507065009, 1),  "t" : NumberLong(1)  },  "durableOpTime" : {  "ts" : Timestamp(1507065009, 1),  "t" : NumberLong(1)  }  },  "members" : [  {  "\_id" : 0,  "name" : "localhost:40000",  "health" : 1,  "state" : 1,  "stateStr" : "PRIMARY",  "uptime" : 251,  "optime" : {  "ts" : Timestamp(1507065009, 1),  "t" : NumberLong(1)  },  "optimeDate" : ISODate("2017-10-03T21:10:09Z"),  "electionTime" : Timestamp(1507064883, 2),  "electionDate" : ISODate("2017-10-03T21:08:03Z"),  "configVersion" : 3,  "self" : true  },  {  "\_id" : 1,  "name" : "localhost:40001",  "health" : 1,  "state" : 2,  "stateStr" : "SECONDARY",  "uptime" : 29,  "optime" : {  "ts" : Timestamp(1507065009, 1),  "t" : NumberLong(1)  },  "optimeDurable" : {  "ts" : Timestamp(1507065009, 1),  "t" : NumberLong(1)  },  "optimeDate" : ISODate("2017-10-03T21:10:09Z"),  "optimeDurableDate" : ISODate("2017-10-03T21:10:09Z"),  "lastHeartbeat" : ISODate("2017-10-03T21:10:19.261Z"),  "lastHeartbeatRecv" : ISODate("2017-10-03T21:10:18.259Z"),  "pingMs" : NumberLong(0),  "syncingTo" : "localhost:40000",  "configVersion" : 3  },  {  "\_id" : 2,  "name" : "localhost:40002",  "health" : 1,  "state" : 2,  "stateStr" : "SECONDARY",  "uptime" : 26,  "optime" : {  "ts" : Timestamp(1507065009, 1),  "t" : NumberLong(1)  },  "optimeDurable" : {  "ts" : Timestamp(1507065009, 1),  "t" : NumberLong(1)  },  "optimeDate" : ISODate("2017-10-03T21:10:09Z"),  "optimeDurableDate" : ISODate("2017-10-03T21:10:09Z"),  "lastHeartbeat" : ISODate("2017-10-03T21:10:19.261Z"),  "lastHeartbeatRecv" : ISODate("2017-10-03T21:10:18.727Z"),  "pingMs" : NumberLong(0),  "syncingTo" : "localhost:40001",  "configVersion" : 3  }  ],  "ok" : 1  } |

可以看到已经配置好了3个复制集。

启动m0：

sudo mongos -f /data/shard-project/config/m0.conf

登陆m0：

mongo --port 50000

使用admin库配置，添加分片，打开test数据库的分片选项，为test下的persons表添加片键：

|  |
| --- |
| use admin  db.runCommand({addshard:"localhost:30001"})  db.runCommand({addshard:"localhost:30002"})  db.runCommand({enablesharding:"test"})  db.runCommand({shardcollection:"test.persons",key:{\_id:1}}) |

下面开始测试：

向test.persons数据库添加500000条数据，查看分片情况：

|  |
| --- |
| use test  for(var i = 0; i < 500000; i++) db.persons.insert({age:i, name:"ly"}) |

由于数据量比较大，需要一段时间。

我实际上后面又添加了50000条数据，其实不需要，但是如果要计算后面的每一个shard条数之和，可能会引起混淆，所以说明一下。

完成之后，为了显示效果，这里需要把chunk的size调小，调成1M：

|  |
| --- |
| use config  db.settings.save( { \_id:"chunksize", value: 1 } ) |

然后，Mongo会在后台把chunk拆分，均摊到每一个shard上。如果不改chunksize，可能最后只会分到一个shard上面。

检测每一个shard的情况：

|  |
| --- |
| use test  db.persons.stats() |

显示：

|  |
| --- |
| mongos> db.persons.stats()  {  "sharded" : true,  "capped" : false,  "ns" : "test.persons",  "count" : 550000,  "size" : 26400000,  "storageSize" : 10850304,  "totalIndexSize" : 7352320,  "indexSizes" : {  "\_id\_" : 7352320  },  "avgObjSize" : 48,  "nindexes" : 1,  "nchunks" : 50,  "shards" : {  "shard0000" : {  "ns" : "test.persons",  "size" : 17761440,  "count" : 370030,  "avgObjSize" : 48,  "storageSize" : 9039872,  "capped" : false,  "wiredTiger" : {  "metadata" : {  "formatVersion" : 1  },  "creationString" : "access\_pattern\_hint=none,allocation\_size=4KB,app\_metadata=(formatVersion=1),block\_allocation=best,block\_compressor=snappy,cache\_resident=false,checksum=on,colgroups=,collator=,columns=,dictionary=0,encryption=(keyid=,name=),exclusive=false,extractor=,format=btree,huffman\_key=,huffman\_value=,ignore\_in\_memory\_cache\_size=false,immutable=false,internal\_item\_max=0,internal\_key\_max=0,internal\_key\_truncate=true,internal\_page\_max=4KB,key\_format=q,key\_gap=10,leaf\_item\_max=0,leaf\_key\_max=0,leaf\_page\_max=32KB,leaf\_value\_max=64MB,log=(enabled=true),lsm=(auto\_throttle=true,bloom=true,bloom\_bit\_count=16,bloom\_config=,bloom\_hash\_count=8,bloom\_oldest=false,chunk\_count\_limit=0,chunk\_max=5GB,chunk\_size=10MB,merge\_max=15,merge\_min=0),memory\_page\_max=10m,os\_cache\_dirty\_max=0,os\_cache\_max=0,prefix\_compression=false,prefix\_compression\_min=4,source=,split\_deepen\_min\_child=0,split\_deepen\_per\_child=0,split\_pct=90,type=file,value\_format=u",  "type" : "file",  "uri" : "statistics:table:collection-5-1934451794160605558",  "LSM" : {  "bloom filter false positives" : 0,  "bloom filter hits" : 0,  "bloom filter misses" : 0,  "bloom filter pages evicted from cache" : 0,  "bloom filter pages read into cache" : 0,  "bloom filters in the LSM tree" : 0,  "chunks in the LSM tree" : 0,  "highest merge generation in the LSM tree" : 0,  "queries that could have benefited from a Bloom filter that did not exist" : 0,  "sleep for LSM checkpoint throttle" : 0,  "sleep for LSM merge throttle" : 0,  "total size of bloom filters" : 0  },  "block-manager" : {  "allocations requiring file extension" : 1123,  "blocks allocated" : 1938,  "blocks freed" : 1151,  "checkpoint size" : 6922240,  "file allocation unit size" : 4096,  "file bytes available for reuse" : 2084864,  "file magic number" : 120897,  "file major version number" : 1,  "file size in bytes" : 9039872,  "minor version number" : 0  },  "btree" : {  "btree checkpoint generation" : 21,  "column-store fixed-size leaf pages" : 0,  "column-store internal pages" : 0,  "column-store variable-size RLE encoded values" : 0,  "column-store variable-size deleted values" : 0,  "column-store variable-size leaf pages" : 0,  "fixed-record size" : 0,  "maximum internal page key size" : 368,  "maximum internal page size" : 4096,  "maximum leaf page key size" : 2867,  "maximum leaf page size" : 32768,  "maximum leaf page value size" : 67108864,  "maximum tree depth" : 3,  "number of key/value pairs" : 0,  "overflow pages" : 0,  "pages rewritten by compaction" : 0,  "row-store internal pages" : 0,  "row-store leaf pages" : 0  },  "cache" : {  "bytes currently in the cache" : 54288061,  "bytes read into cache" : 0,  "bytes written from cache" : 53373061,  "checkpoint blocked page eviction" : 0,  "data source pages selected for eviction unable to be evicted" : 0,  "hazard pointer blocked page eviction" : 0,  "in-memory page passed criteria to be split" : 24,  "in-memory page splits" : 12,  "internal pages evicted" : 0,  "internal pages split during eviction" : 0,  "leaf pages split during eviction" : 6,  "modified pages evicted" : 16,  "overflow pages read into cache" : 0,  "overflow values cached in memory" : 0,  "page split during eviction deepened the tree" : 0,  "page written requiring lookaside records" : 0,  "pages read into cache" : 0,  "pages read into cache requiring lookaside entries" : 0,  "pages requested from the cache" : 2411481,  "pages written from cache" : 1909,  "pages written requiring in-memory restoration" : 0,  "tracked dirty bytes in the cache" : 9845189,  "unmodified pages evicted" : 0  },  "cache\_walk" : {  "Average difference between current eviction generation when the page was last considered" : 0,  "Average on-disk page image size seen" : 0,  "Clean pages currently in cache" : 0,  "Current eviction generation" : 0,  "Dirty pages currently in cache" : 0,  "Entries in the root page" : 0,  "Internal pages currently in cache" : 0,  "Leaf pages currently in cache" : 0,  "Maximum difference between current eviction generation when the page was last considered" : 0,  "Maximum page size seen" : 0,  "Minimum on-disk page image size seen" : 0,  "On-disk page image sizes smaller than a single allocation unit" : 0,  "Pages created in memory and never written" : 0,  "Pages currently queued for eviction" : 0,  "Pages that could not be queued for eviction" : 0,  "Refs skipped during cache traversal" : 0,  "Size of the root page" : 0,  "Total number of pages currently in cache" : 0  },  "compression" : {  "compressed pages read" : 0,  "compressed pages written" : 1869,  "page written failed to compress" : 0,  "page written was too small to compress" : 40,  "raw compression call failed, additional data available" : 0,  "raw compression call failed, no additional data available" : 0,  "raw compression call succeeded" : 0  },  "cursor" : {  "bulk-loaded cursor-insert calls" : 0,  "create calls" : 19,  "cursor-insert key and value bytes inserted" : 43675843,  "cursor-remove key bytes removed" : 1803531,  "cursor-update value bytes updated" : 0,  "insert calls" : 841504,  "next calls" : 321138,  "prev calls" : 2,  "remove calls" : 471474,  "reset calls" : 2411447,  "restarted searches" : 0,  "search calls" : 1248799,  "search near calls" : 321145,  "truncate calls" : 0,  "update calls" : 0  },  "reconciliation" : {  "dictionary matches" : 0,  "fast-path pages deleted" : 0,  "internal page key bytes discarded using suffix compression" : 4958,  "internal page multi-block writes" : 11,  "internal-page overflow keys" : 0,  "leaf page key bytes discarded using prefix compression" : 0,  "leaf page multi-block writes" : 34,  "leaf-page overflow keys" : 0,  "maximum blocks required for a page" : 2,  "overflow values written" : 0,  "page checksum matches" : 630,  "page reconciliation calls" : 72,  "page reconciliation calls for eviction" : 16,  "pages deleted" : 12  },  "session" : {  "object compaction" : 0,  "open cursor count" : 7  },  "transaction" : {  "update conflicts" : 0  }  },  "nindexes" : 1,  "totalIndexSize" : 5877760,  "indexSizes" : {  "\_id\_" : 5877760  },  "ok" : 1  },  "shard0001" : {  "ns" : "test.persons",  "size" : 8638560,  "count" : 179970,  "avgObjSize" : 48,  "storageSize" : 1810432,  "capped" : false,  "wiredTiger" : {  "metadata" : {  "formatVersion" : 1  },  "creationString" : "access\_pattern\_hint=none,allocation\_size=4KB,app\_metadata=(formatVersion=1),block\_allocation=best,block\_compressor=snappy,cache\_resident=false,checksum=on,colgroups=,collator=,columns=,dictionary=0,encryption=(keyid=,name=),exclusive=false,extractor=,format=btree,huffman\_key=,huffman\_value=,ignore\_in\_memory\_cache\_size=false,immutable=false,internal\_item\_max=0,internal\_key\_max=0,internal\_key\_truncate=true,internal\_page\_max=4KB,key\_format=q,key\_gap=10,leaf\_item\_max=0,leaf\_key\_max=0,leaf\_page\_max=32KB,leaf\_value\_max=64MB,log=(enabled=true),lsm=(auto\_throttle=true,bloom=true,bloom\_bit\_count=16,bloom\_config=,bloom\_hash\_count=8,bloom\_oldest=false,chunk\_count\_limit=0,chunk\_max=5GB,chunk\_size=10MB,merge\_max=15,merge\_min=0),memory\_page\_max=10m,os\_cache\_dirty\_max=0,os\_cache\_max=0,prefix\_compression=false,prefix\_compression\_min=4,source=,split\_deepen\_min\_child=0,split\_deepen\_per\_child=0,split\_pct=90,type=file,value\_format=u",  "type" : "file",  "uri" : "statistics:table:collection-5--2521018454738123524",  "LSM" : {  "bloom filter false positives" : 0,  "bloom filter hits" : 0,  "bloom filter misses" : 0,  "bloom filter pages evicted from cache" : 0,  "bloom filter pages read into cache" : 0,  "bloom filters in the LSM tree" : 0,  "chunks in the LSM tree" : 0,  "highest merge generation in the LSM tree" : 0,  "queries that could have benefited from a Bloom filter that did not exist" : 0,  "sleep for LSM checkpoint throttle" : 0,  "sleep for LSM merge throttle" : 0,  "total size of bloom filters" : 0  },  "block-manager" : {  "allocations requiring file extension" : 223,  "blocks allocated" : 223,  "blocks freed" : 4,  "checkpoint size" : 1753088,  "file allocation unit size" : 4096,  "file bytes available for reuse" : 40960,  "file magic number" : 120897,  "file major version number" : 1,  "file size in bytes" : 1810432,  "minor version number" : 0  },  "btree" : {  "btree checkpoint generation" : 7,  "column-store fixed-size leaf pages" : 0,  "column-store internal pages" : 0,  "column-store variable-size RLE encoded values" : 0,  "column-store variable-size deleted values" : 0,  "column-store variable-size leaf pages" : 0,  "fixed-record size" : 0,  "maximum internal page key size" : 368,  "maximum internal page size" : 4096,  "maximum leaf page key size" : 2867,  "maximum leaf page size" : 32768,  "maximum leaf page value size" : 67108864,  "maximum tree depth" : 3,  "number of key/value pairs" : 0,  "overflow pages" : 0,  "pages rewritten by compaction" : 0,  "row-store internal pages" : 0,  "row-store leaf pages" : 0  },  "cache" : {  "bytes currently in the cache" : 24532401,  "bytes read into cache" : 0,  "bytes written from cache" : 6199839,  "checkpoint blocked page eviction" : 0,  "data source pages selected for eviction unable to be evicted" : 0,  "hazard pointer blocked page eviction" : 0,  "in-memory page passed criteria to be split" : 4,  "in-memory page splits" : 2,  "internal pages evicted" : 0,  "internal pages split during eviction" : 0,  "leaf pages split during eviction" : 0,  "modified pages evicted" : 0,  "overflow pages read into cache" : 0,  "overflow values cached in memory" : 0,  "page split during eviction deepened the tree" : 0,  "page written requiring lookaside records" : 0,  "pages read into cache" : 0,  "pages read into cache requiring lookaside entries" : 0,  "pages requested from the cache" : 180043,  "pages written from cache" : 220,  "pages written requiring in-memory restoration" : 0,  "tracked dirty bytes in the cache" : 15472806,  "unmodified pages evicted" : 0  },  "cache\_walk" : {  "Average difference between current eviction generation when the page was last considered" : 0,  "Average on-disk page image size seen" : 0,  "Clean pages currently in cache" : 0,  "Current eviction generation" : 0,  "Dirty pages currently in cache" : 0,  "Entries in the root page" : 0,  "Internal pages currently in cache" : 0,  "Leaf pages currently in cache" : 0,  "Maximum difference between current eviction generation when the page was last considered" : 0,  "Maximum page size seen" : 0,  "Minimum on-disk page image size seen" : 0,  "On-disk page image sizes smaller than a single allocation unit" : 0,  "Pages created in memory and never written" : 0,  "Pages currently queued for eviction" : 0,  "Pages that could not be queued for eviction" : 0,  "Refs skipped during cache traversal" : 0,  "Size of the root page" : 0,  "Total number of pages currently in cache" : 0  },  "compression" : {  "compressed pages read" : 0,  "compressed pages written" : 218,  "page written failed to compress" : 0,  "page written was too small to compress" : 2,  "raw compression call failed, additional data available" : 0,  "raw compression call failed, no additional data available" : 0,  "raw compression call succeeded" : 0  },  "cursor" : {  "bulk-loaded cursor-insert calls" : 0,  "create calls" : 3,  "cursor-insert key and value bytes inserted" : 9275921,  "cursor-remove key bytes removed" : 0,  "cursor-update value bytes updated" : 0,  "insert calls" : 179967,  "next calls" : 0,  "prev calls" : 1,  "remove calls" : 0,  "reset calls" : 179967,  "restarted searches" : 71,  "search calls" : 1,  "search near calls" : 0,  "truncate calls" : 0,  "update calls" : 0  },  "reconciliation" : {  "dictionary matches" : 0,  "fast-path pages deleted" : 0,  "internal page key bytes discarded using suffix compression" : 506,  "internal page multi-block writes" : 0,  "internal-page overflow keys" : 0,  "leaf page key bytes discarded using prefix compression" : 0,  "leaf page multi-block writes" : 3,  "leaf-page overflow keys" : 0,  "maximum blocks required for a page" : 124,  "overflow values written" : 0,  "page checksum matches" : 37,  "page reconciliation calls" : 5,  "page reconciliation calls for eviction" : 0,  "pages deleted" : 0  },  "session" : {  "object compaction" : 0,  "open cursor count" : 3  },  "transaction" : {  "update conflicts" : 0  }  },  "nindexes" : 1,  "totalIndexSize" : 1474560,  "indexSizes" : {  "\_id\_" : 1474560  },  "ok" : 1  }  },  "ok" : 1  } |

可以看到每一个shard上数据的大小；

也可以通过

printShardingStatus()

来查看shard信息：

|  |
| --- |
| mongos> printShardingStatus()  --- Sharding Status ---  sharding version: {  "\_id" : 1,  "minCompatibleVersion" : 5,  "currentVersion" : 6,  "clusterId" : ObjectId("59d3fc36c6a41efbca5eb228")  }  shards:  { "\_id" : "shard0000", "host" : "localhost:30001", "state" : 1 }  { "\_id" : "shard0001", "host" : "localhost:30002", "state" : 1 }  active mongoses:  "3.4.9" : 1  autosplit:  Currently enabled: yes  balancer:  Currently enabled: yes  Currently running: no  Balancer lock taken at Tue Oct 03 2017 17:08:06 GMT-0400 (EDT) by ConfigServer:Balancer  Failed balancer rounds in last 5 attempts: 0  Migration Results for the last 24 hours:  23 : Success  databases:  { "\_id" : "test", "primary" : "shard0000", "partitioned" : true }  test.persons  shard key: { "\_id" : 1 }  unique: false  balancing: true  chunks:  shard0000 25  shard0001 25  too many chunks to print, use verbose if you want to force print |

至此，shard配置成功。