mongodb分片

一般用得比较少，需要较多的服务器，还有三种的角色

一般把mongodb的副本集应用得好就足够用了，可搭建多套mongodb复本集

mongodb分片技术

mongodb副本集可以解决数据备份、读性能的问题，但由于mongodb副本集是每份数据都一模一样的，无法解决数据量过大问题

mongodb分片技术能够把数据分成两份存储，假如shijiange.myuser里面有1亿条数据，分片能够实现5千万左右存储在data1，5千万左右存储在data2

data1、data2需要使用副本集的形式，预防数据丢失

mongodb分片集群三种角色介绍

router角色 #mongodb的路由，提供入口，使得分片集群对外透明。router不存储数据

configsvr角色 #mongodb的配置角色，存储元数据信息。分片集群后端有多份存储，读取数据该去哪个存储上读取，依赖于配置角色。配置角色建议使用副本集

shardsvr角色 #mongodb的存储角色，存储真正的数据，建议使用副本集

依赖关系

当用户通过router角色插入数据时，需要从configsvr知道这份数据插入到哪个节点，然后执行插入动作插入数据到sharedsvr

当用户通过router角色获取数据时，需要从configsvr知道这份数据是存储在哪个节点，然后再去sharedsvr获取数据

mongodb分片集群的搭建说明

使用同一份mongodb二进制文件

修改对应的配置就能实现分片集群的搭建

mongodb分片集群实战环境搭建说明

configsvr #使用28017，28018，28019三个端口来搭建

router #使用27017，27018，27019三个端口来搭建

shardsvr #使用29017，29018，29019，29020四个端口来搭建，两个端口一个集群，生产环境肯定是要三个端口

mongodb配置角色的搭建，配置文件路径/data/mongodb/28017/mongodb.conf

systemLog:

destination: file

logAppend: true

path: /data/mongodb/28017/mongodb.log

storage:

dbPath: /data/mongodb/28017/

journal:

enabled: true

processManagement:

fork: true

net:

port: 28017

bindIp: 127.0.0.1

replication:

replSetName: shijiangeconf

sharding:

clusterRole: configsvr

mongodb配置服务集群的启动跟单例的启动方式一致，都是使用mongod

分片集群的配置角色副本集搭建

config = { \_id:"shijiangeconf",

configsvr: true,

members:[

{\_id:0,host:"127.0.0.1:28017"},

{\_id:1,host:"127.0.0.1:28018"},

{\_id:2,host:"127.0.0.1:28019"}

]

}

rs.initiate(config)

验证是否搭建成功

/usr/local/mongodb/bin/mongo 127.0.0.1:28017

rs.status()