

博客专区 > Crazy_Coder的博客 > 博客详情

原 荐

Mongodb高可用集群搭建

Crazy_Coder 发表于 19小时前 阅读 634 收藏 67 点赞 2 评论 3

收藏

HOT

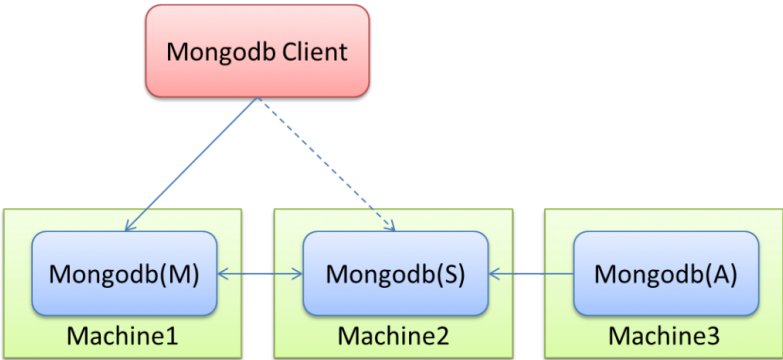
摘要: 最近公司项目要求Mongodb集群且高可用, 在查询了一系列文章, 再结合项目实际情况, 所以采用了这一套高可用集群方案

方案选项

Mongodb的三种集群方式的搭建：Replica Set / Sharding / Master-Slaver。

1. Replica Set (副本集)

其实简单来说就是集群当中包含了多份数据, 保证主节点挂掉了, 备节点能继续提供数据服务, 提供的前提就是数据需要和主节点一致。



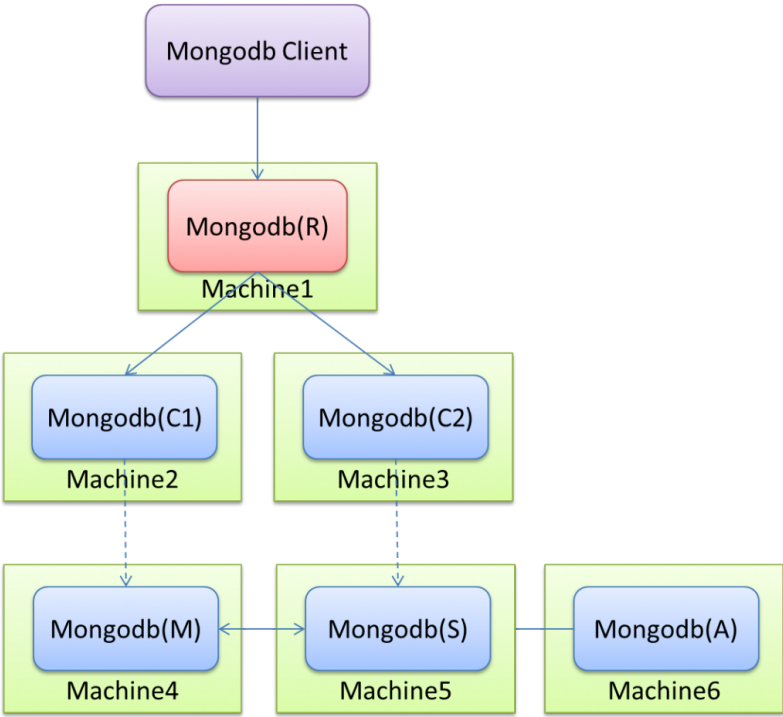
Mongodb(M)表示主节点, Mongodb(S)表示备节点, Mongodb(A)表示仲裁节点。主备节点存储数据, 仲裁节点不存储数据。客户端同时连接主节点与备节点, 不连接仲裁节点。

默认设置下, 主节点提供所有增删查改服务, 备节点不提供任何服务。但是可以通过设置使备节点提供查询服务, 这样就可以减少主节点的压力, 当客户端进行数据查询时, 请求自动转到备节点上。这个设置叫做Read Preference Modes, 同时java客户端提供了简单的配置方式, 可以不必直接对数据库进行操作。

仲裁节点是一种特殊的节点, 它本身并不存储数据, 主要的作用是决定哪一个备节点在主节点挂掉之后提升为主节点, 所以客户端不需要连接此节点。这里虽然只有一个备节点, 但是仍然需要一个仲裁节点来提升备节点级别。我开始也不相信必须要有仲裁节点, 但是自己也试过没仲裁节点的话, 主节点挂了备节点还是备节点, 所以咱们还是需要它的。

2. Sharding

和Replica Set类似, 都需要一个仲裁节点, 但是Sharding还需要配置节点和路由节点。就三种集群搭建方式来说, 这种是最复杂的。部署图如下:



- 方案选项
- 基于Replica Set
- 客户端连接
- 附录

3. Master-Slaver

这个是最简答的集群搭建，不过准确说也不能算是集群，只能说是主备。并且官方已经不推荐这种方式。

4. 综述

- Master-Slaver 官方已经不推荐这种方式了。不采用！
- Sharding 过于复杂，考虑项目情况，也不采用！
- Replica Set 集群、高可用，部署也简单，采用这种情况，下面介绍这种情况的部署！

基于 Replica Set 的方式进行部署

1. 准备

由于是在本机虚拟机部署，所以只采用一台机器（ 10.211.55.10 ），开启4个端口（ master ： 27017 ， slaver ： 27018、 27019 ， arbiter ： 27020 ）

2. 配置文件(mongodb配置文件具体属性解释请参考附录1)

- master

```
# master.conf
dbpath=/opt/mongodb/data/master
logpath=/opt/mongodb/logs/master.log
pidfilepath=/opt/mongodb/logs/master.pid
directoryperdb=true
logappend=true
replSet=ynzw
bind_ip=10.211.55.10
port=27017
oplogSize=10000
fork=true
noprealloc=true
```
- slaver1

```
# slaver1.conf
dbpath=/opt/mongodb/data/slaver1
logpath=/opt/mongodb/logs/slaver1.log
pidfilepath=/opt/mongodb/logs/slaver1.pid
directoryperdb=true
logappend=true
```

```

replSet=ynzw
bind_ip=10.211.55.10
port=27018
oplogSize=10000
fork=true
noprealloc=true

```

- slaver2

```

# slaver2.conf
dbpath=/opt/mongodb/data/slaver2
logpath=/opt/mongodb/logs/slaver2.log
pidfilepath=/opt/mongodb/logs/slaver2.pid
directoryperdb=true
logappend=true
replSet=ynzw
bind_ip=10.211.55.10
port=27019
oplogSize=10000
fork=true
noprealloc=true

```

- arbiter

```

# arbiter.conf
dbpath=/opt/mongodb/data/arbiter
logpath=/opt/mongodb/logs/arbiter.log
pidfilepath=/opt/mongodb/logs/arbiter.pid
directoryperdb=true
logappend=true
replSet=ynzw
bind_ip=10.211.55.10
port=27020
oplogSize=10000
fork=true
noprealloc=true

```

3. 启动4个节点

```

/opt/mongodb/bin/mongod -f /opt/mongodb/conf/master.conf
/opt/mongodb/bin/mongod -f /opt/mongodb/conf/slaver1.conf
/opt/mongodb/bin/mongod -f /opt/mongodb/conf/slaver2.conf
/opt/mongodb/bin/mongod -f /opt/mongodb/conf/arbiter.conf

```

4. 配置主，备，仲裁节点

- 客户端连接master、slaver中任意一个节点mongodb

```

/opt/mongodb/bin/mongo 10.211.55.10:27017

```

- 开始配置

```

> use admin
> cfg = {
  _id: "ynzw",
  members: [
    {
      _id: 0,
      host: '10.211.55.10:27017',
      priority: 3
    },
    {
      _id: 1,
      host: '10.211.55.10:27018',
      priority: 2
    },
    {
      _id: 2,
      host: '10.211.55.10:27019',
      priority: 1
    },
    {
      _id: 3,
      host: '10.211.55.10:27020',
      arbiterOnly: true
    }
  ]
}

```

方案选项

基于Replica Set

客户端连接

附录

```

    }
  ]
};
> rs.initiate(cfg)

```

cfg是可以任意的名字，当然最好不要是mongodb的关键字，conf, config都可以。最外层的_id表示replica set的名字，members里包含的是所有节点的地址以及优先级。优先级最高的即成为主节点，即这里的10.211.55.10:27017。特别注意的是，对于仲裁节点，需要有个特别的配置——arbiterOnly:true。这个千万不能少了，不然主备模式就不能生效。

• 检验

配置的生效时间根据不同的机器配置会有长有短，配置不错的话基本上十几秒内就能生效，有的配置需要一两分钟。如果生效了，执行rs.status()命令会看到如下信息：

```

ynzw:SECONDARY> rs.status()
{
  "set" : "ynzw",
  "date" : ISODate("2017-05-26T06:47:32.069Z"),
  "myState" : 1,
  "term" : NumberLong(8),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "members" : [
    {
      "_id" : 0,
      "name" : "10.211.55.10:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 24,
      "optime" : {
        "ts" : Timestamp(1495781239, 2),
        "t" : NumberLong(8)
      },
      "optimeDate" : ISODate("2017-05-26T06:47:19Z"),
      "electionTime" : Timestamp(1495781239, 1),
      "electionDate" : ISODate("2017-05-26T06:47:19Z"),
      "configVersion" : 1,
      "self" : true
    },
    {
      "_id" : 1,
      "name" : "10.211.55.10:27018",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 18,
      "optime" : {
        "ts" : Timestamp(1495781239, 2),
        "t" : NumberLong(8)
      },
      "optimeDate" : ISODate("2017-05-26T06:47:19Z"),
      "lastHeartbeat" : ISODate("2017-05-26T06:47:31.424Z"),
      "lastHeartbeatRecv" : ISODate("2017-05-26T06:47:31.247Z"),
      "pingMs" : NumberLong(0),
      "syncingTo" : "10.211.55.10:27017",
      "configVersion" : 1
    },
    {
      "_id" : 2,
      "name" : "10.211.55.10:27019",
      "health" : 1,
      "state" : 2,
      "stateStr" : "SECONDARY",
      "uptime" : 18,
      "optime" : {
        "ts" : Timestamp(1495781239, 2),
        "t" : NumberLong(8)
      },
      "optimeDate" : ISODate("2017-05-26T06:47:19Z"),
      "lastHeartbeat" : ISODate("2017-05-26T06:47:31.424Z"),
      "lastHeartbeatRecv" : ISODate("2017-05-26T06:47:31.734Z"),
      "pingMs" : NumberLong(0),
      "syncingTo" : "10.211.55.10:27018",
      "configVersion" : 1
    },
    {
      "_id" : 3,
      "name" : "10.211.55.10:27020",
      "health" : 1,
      "state" : 7,
      "stateStr" : "ARBITER",

```

方案选项
基于Replica Set
客户端连接
附录

```
        "uptime" : 18,
        "lastHeartbeat" : ISODate("2017-05-26T06:47:31.424Z"),
        "lastHeartbeatRecv" : ISODate("2017-05-26T06:47:30.437Z"),
        "pingMs" : NumberLong(0),
        "configVersion" : 1
      },
    ],
    "ok" : 1
  }
  ynzw:PRIMARY>
```

如果配置正在生效，其中会包含如下信息：“stateStr”：“RECOVERING”

客户端连接

- 1. 使用 spring-data-mongodb + mongo-java-driver 操作mongodb
- 2. 10.211.55.10:27017、10.211.55.10:27018、10.211.55.10:27019 是主从节点，由于仲裁节点不存储数据，所以不需要连接
- 3. 代码

```
List<ServerAddress> servers = new ArrayList<>(addresses.size());
for (String address : addresses) {
    String host = address.split(":")[0];
    Integer port = Integer.valueOf(address.split(":")[1]);

    servers.add(new ServerAddress(host, port));
}

MongoClient client = new MongoClient(servers);
MongoDatabase db = client.getDatabase(mongoName);
```

附录

1. mongodb配置文件具体属性解释

- dbpath: 数据存放目录
- logpath: 日志存放路径
- pidfilepath: 进程文件，方便停止mongodb
- directoryperdb: 为每一个数据库按照数据库名建立文件夹存放
- logappend: 以追加的方式记录日志
- replSet: replica set的名字
- bind_ip: mongodb所绑定的ip地址
- port: mongodb进程所使用的端口号，默认为27017
- oplogSize: mongodb操作日志文件的最大大小。单位为Mb，默认为硬盘剩余空间的5%
- fork: 以后台方式运行进程
- noprealloc: 不预先分配存储

© 著作权归作者所有

分类：技术类 字数：1560

打赏

点赞

收藏

分享

- 方案选项
- 基于Replica Set
- 客户端连接
- 附录