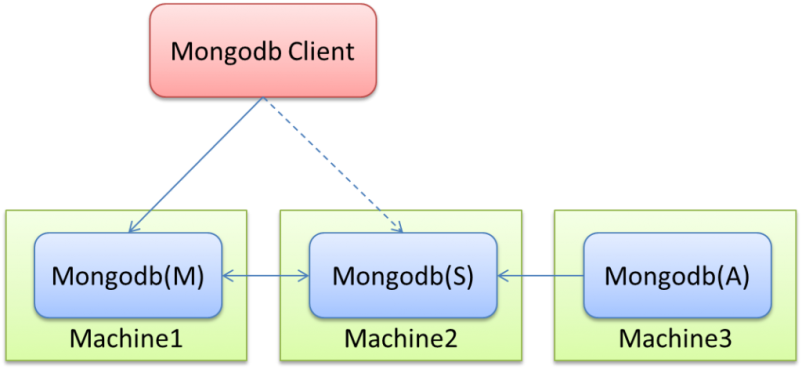
 Mongodb是时下流行的NoSql数据库，它的存储方式是文档式存储，并不是Key-Value形式。关于Mongodb的特点，这里就不多介绍了，大家可以去看看官方说明：[http://docs.mongodb.org/manual/](http://docs.mongodb.org/manual/" \t "_blank)

       今天主要来说说Mongodb的三种集群方式的搭建：Replica Set / Sharding / Master-Slaver。这里只说明最简单的集群搭建方式（生产环境），如果有多个节点可以此类推或者查看官方文档。OS是Ubuntu\_x64系统，客户端用的是Java客户端。Mongodb版本是mongodb-linux-x86\_64-2.2.2.tgz

**Replica Set**

       中文翻译叫做副本集，不过我并不喜欢把英文翻译成中文，总是感觉怪怪的。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。如下图：



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](http://docs.mongodb.org/manual/applications/replication/#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。

介绍完了集群方案，那么现在就开始搭建了。

1.建立数据文件夹

一般情况下不会把数据目录建立在mongodb的解压目录下，不过这里方便起见，就建在mongodb解压目录下吧。

mkdir –p /mongodb/data/master

mkdir –p /mongodb/data/slaver

mkdir –p /mongodb/data/arbiter

#三个目录分别对应主，备，仲裁节点

#/mongodb指mongodb的解压目录，我的是/mongodb-linux-x86\_64-ubuntu1204-3.2.4，下文的也是如此！！！！！

2.建立配置文件

由于配置比较多，所以我们将配置写到文件里。

1. #master.conf
2. dbpath=/mongodb/data/master
3. logpath=/mongodb/log/master.log
4. pidfilepath=/mongodb/master.pid
5. directoryperdb=true
6. replSet= learningspark
7. port=27017
8. fork=true
9. noprealloc=true
10. #slaver.conf
11. dbpath=/mongodb/data/slaver
12. logpath=/mongodb/log/slaver.log
13. pidfilepath=/mongodb/slaver.pid
14. directoryperdb=true
15. replSet= learningspark
16. port=27017
17. fork=true
18. noprealloc=true
19. #arbiter.conf
20. dbpath=/mongodb/data/arbiter
21. logpath=/mongodb/log/arbiter.log
22. pidfilepath=/mongodb/arbiter.pid
23. directoryperdb=true
24. replSet=learningspark
25. port=27017
26. fork=true
27. noprealloc=true

参数解释：

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

3.启动mongodb

进入每个mongodb节点的bin目录下

1. ./monood -f master.conf
2. ./mongod -f slaver.conf
3. ./mongod -f arbiter.conf

注意配置文件的路径一定要保证正确，可以是相对路径也可以是绝对路径。

4.配置主，备，仲裁节点

可以通过客户端连接mongodb，也可以直接在三个节点中选择一个连接mongodb。（本机上的ip地址必须是内网ip！！！！）

1. ./mongo 10.10.148.130:27017   #ip和port是某个节点的地址
2. >use admin
3. >cfg={ \_id:"learningspark", members:[ {\_id:0,host:'10.10.148.130:27017',priority:2}, {\_id:1,host:'10.10.148.131:27017',priority:1},
4. {\_id:2,host:'10.10.148.132:27017',arbiterOnly:true}] };
5. >rs.initiate(cfg)             #使配置生效

       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10.10.148.130:27017。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。

      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：

1. {
2. "set" : "learningspark",
3. "date" : ISODate("2013-01-05T02:44:43Z"),
4. "myState" : 1,
5. "members" : [
6. {
7. "\_id" : 0,
8. "name" : "10.10.148.130:27017",
9. "health" : 1,
10. "state" : 1,
11. "stateStr" : "PRIMARY",
12. "uptime" : 200,
13. "optime" : Timestamp(1357285565000, 1),
14. "optimeDate" : ISODate("2013-01-04T07:46:05Z"),
15. "self" : true
16. },
17. {
18. "\_id" : 1,
19. "name" : "10.10.148.131:27017",
20. "health" : 1,
21. "state" : 2,
22. "stateStr" : "SECONDARY",
23. "uptime" : 200,
24. "optime" : Timestamp(1357285565000, 1),
25. "optimeDate" : ISODate("2013-01-04T07:46:05Z"),
26. "lastHeartbeat" : ISODate("2013-01-05T02:44:42Z"),
27. "pingMs" : 0
28. },
29. {
30. "\_id" : 2,
31. "name" : "10.10.148.132:27017",
32. "health" : 1,
33. "state" : 7,
34. "stateStr" : "ARBITER",
35. "uptime" : 200,
36. "lastHeartbeat" : ISODate("2013-01-05T02:44:42Z"),
37. "pingMs" : 0
38. }
39. ],
40. "ok" : 1
41. }

如果配置正在生效，其中会包含如下信息：

1. "stateStr" : "RECOVERING"

同时可以查看对应节点的日志，发现正在等待别的节点生效或者正在分配数据文件。

       现在基本上已经完成了集群的所有搭建工作。至于测试工作，可以留给大家自己试试。一个是往主节点插入数据，能从备节点查到之前插入的数据（查询备节点可能会遇到某个问题，可以自己去网上查查看）。二是停掉主节点，备节点能变成主节点提供服务。三是恢复主节点，备节点也能恢复其备的角色，而不是继续充当主的角色。二和三都可以通过rs.status()命令实时查看集群的变化。