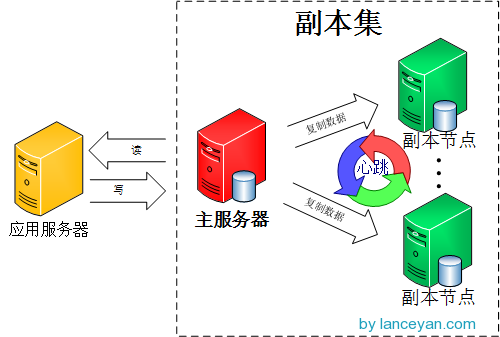
**一、副本集概念**

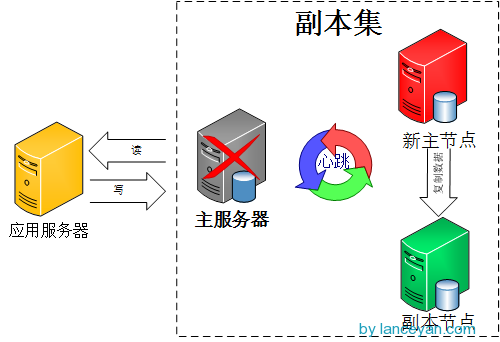
什么是副本？可能第一印象想到的是游戏副本，游戏副本就是为了让每个玩家都有一个独立的游戏环境，这种环境的复制就是副本的一种体现。MongoDB也提供了对副本的支持，副本集中有多个副本保证数据库的容错性，即使一个副本挂掉了还是会存在很多副本；并且支持副本间的自动选举，切换，解决了上一篇博文讲到的主从复制的故障恢复得人工进行的问题。其实主从复制就是一个单副本，缺少扩展性，容错性。

**二、副本集原理及图解**

应用服务器，也就是客户端，连接到MongoDB的整个副本集，副本集中有一台主服务器负责整个副本集的读写，副本节点定期同步主节点的数据和oplog，以保证数据的一致性，一旦主节点宕机或挂掉，副本节点会通过心跳机制检测到，并根据事先副本集创建时设置的节点优先级进行主节点的重新选举，从而保证高可用。如此，客户端完全不必关心副本集的健康状况，MongoDB集群也可以长期保持高可用。

副本集提供了MongoDB集群故障的自动回复机制，扩展性高，容错性强，是MongoDB官方强烈推荐使用的集群解决方案。





三**、副本集使用说明**

以4个节点为例，ip,端口，日志以及数据存放路径如下

**节点1**  
地址：127.0.0.1:1001 # 在项目中，是实际的机器IP和端口，

如192.168.1.101:27017

安装路径：D:\mongodb\_cluster\MongoDB1  
 日志存放路径：D:\mongodb\_cluster\MongoDB1\log\mongodb.log  
 数据存放路径：D:\mongodb\_cluster\MongoDB1\data

**节点2**  
地址：127.0.0.1:1002

安装路径：D:\mongodb\_cluster\MongoDB2  
 日志存放路径：D:\mongodb\_cluster\MongoDB2\log\mongodb.log  
 数据存放路径：D:\mongodb\_cluster\MongoDB2\data

**节点3**  
地址：127.0.0.1:1003

安装路径：D:\mongodb\_cluster\MongoDB3  
 日志存放路径：D:\mongodb\_cluster\MongoDB3\log\mongodb.log  
 数据存放路径：D:\mongodb\_cluster\MongoDB3\data

**节点4**  
地址：127.0.0.1:1004

安装路径：D:\mongodb\_cluster\MongoDB4  
 日志存放路径：D:\mongodb\_cluster\MongoDB4\log\mongodb.log  
 数据存放路径：D:\mongodb\_cluster\MongoDB4\data

**分别启动节点命令如下**

启动节点1：  
 mongod --dbpath D:\mongodb\_cluster\MongoDB1\data --logpath D:\mongodb\_cluster\MongoDB1\log\mongodb.log --logappend --port 1001 --replSet guanghuan

启动节点2：  
 mongod --dbpath D:\mongodb\_cluster\MongoDB2\data --logpath D:\mongodb\_cluster\MongoDB2\log\mongodb.log --logappend --port 1002 --replSet guanghuan

启动节点3：  
 mongod --dbpath D:\mongodb\_cluster\MongoDB3\data --logpath D:\mongodb\_cluster\MongoDB3\log\mongodb.log --logappend --port 1003 --replSet guanghuan

启动节点4：  
 mongod --dbpath D:\mongodb\_cluster\MongoDB4\data --logpath D:\mongodb\_cluster\MongoDB4\log\mongodb.log --logappend --port 1004 --replSet guanghuan

**启动节点命令解释**

dbpath： 数据存放路径

logpath：日志存放路径

logappend ： 日志拼接声明

replSet：声明是副本集，后面格式为副本集名称/与之位于同一个副本集的节点地址（副本集要求至少要有两个节点，一个主，一个从）

--fork：后台启动，这里为了效果没有添加此参数

**初始化节点（只能初始化一次，执行初始化之前，必须启动所有节点）**

mongo 127.0.0.1:1001

use admin

**rs.initiate(**

**{"\_id" : "guanghuan",**

**"members" : [**

**{"\_id" : 1, "host" : "127.0.0.1:1001",** "**priority":3},**

**{"\_id" : 2, "host" : "127.0.0.1:1002", "priority":2},**

**{"\_id" : 3, "host" : "127.0.0.1:1003", "priority":1},**

**{"\_id" : 4, "host" : "127.0.0.1:1004", "arbiterOnly" : true}**

**]**

**});**

**初始化节点命令解释**

\_id：副本集名称，与启动节点中的名称保持一致

host：副本集成员地址

priority：优先级，即选举成为新master的优先级，越大优先级越高

**arbiterOnly：仲裁节点**

**验证 : rs.status() # 查看副本集的状态**

登录主节点，即优先级最高的master节点，输入是否主节点的命令：rs.isMaster()

相关指令：

rs.remove(hostportstr) ： 删除节点hostportstr:127.0.0.1:1003

rs.add(hostportstr) :增加节点

登录所有副本集成员，设置：rs.slaveOk()，不然副本集不允许读取数据

修改节点优先级：

**cfg = rs.conf()**

**cfg.members[0].priority=10**

**rs.reconfig(cfg)**

将该主节点关闭，登录优先级第二的从节点，输入是否主节点的命令： 发现在主节点宕机后，优先级较高的节点被选举为了master节点，继续负责读写的工作，从而持续地提供服务，保持集群高可用。

**四、仲裁者节点及其作用**

当集群中的节点数量为偶数个时，投票选举机制会根据数据最后操作，更新时间戳，优先级等来判定谁将成为master节点，如果出现以上条件都符合的情况，且票数相等，此投票环节需要等待若干分钟，这对于客户端来说是无法接受的。

仲裁节点的出现打破了这个僵局，仲裁节点并不需要太多系统资源，也并不持有数据本身，而是参与投票并有效协调从节点争master的撕逼。

集群中部署多仲裁节点时，需在初始化副本集时，添加属性arbiterOnly:true。

备注：

1. 整个副本集中的数据库，用户名和密码要统一
2. 增、删、改 操作只能在master机器上进行
3. 读取 操作 根据操作程序的配置，可以从不同的机器上读取

|  |
| --- |
| *# conn = MongoClient('mongodb://127.0.0.1:1001,127.0.0.1:1002/')* conn = MongoClient([**'127.0.0.1:1001,127.0.0.1:1002'**])  **from** pymongo **import** ReadPreference  test = conn.get\_database(**'test'**, read\_preference=ReadPreference.SECONDARY\_PREFERRED) |

'PRIMARY', ： 在主节点上读取  
'PRIMARY\_PREFERRED', ： 主节点优先  
'SECONDARY', ： 只在 从节点 上读取  
'SECONDARY\_PREFERRED', ：从节点 优先  
'NEAREST', ： 自动选择最空闲的节点读取