**day17-拔高课程**

# 今日大纲

1. Redis的持久化
2. Redis的主从
3. Redis的集群
4. mysql 优化
5. tomcat优化

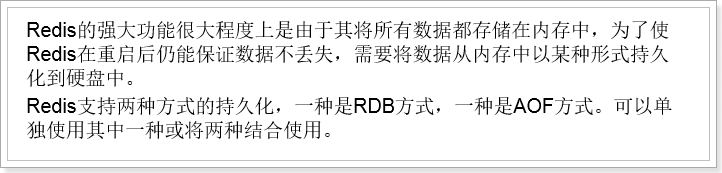
# 每日反馈

# 课堂笔记

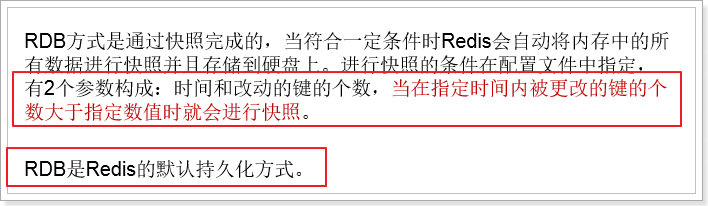
# Redis的持久化

## 什么是持久化

持久化，就是将数据保存到磁盘，机器宕机或者重启数据不丢失，如果存储到内存中的数据，会丢失。

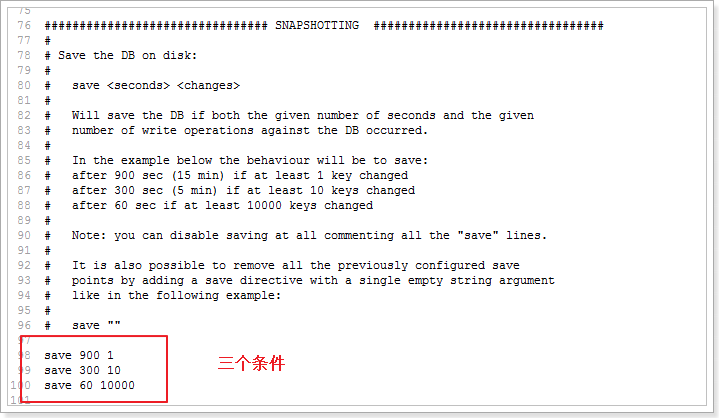


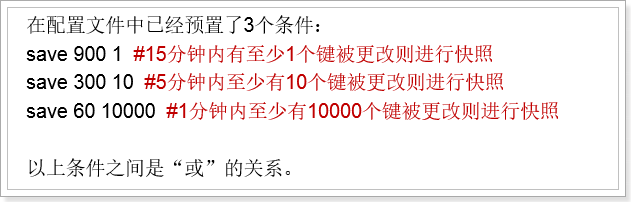
## RDB



### 配置

#### 持久化条件配置

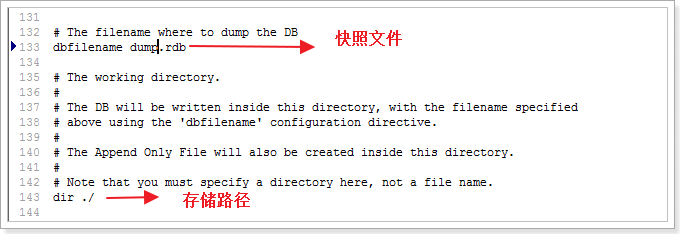




#### 指定rdb快照文件

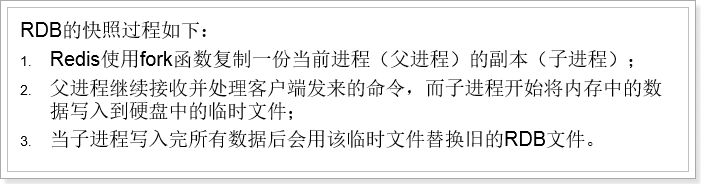
当持久化的任意条件触发，就会在配置文件指定位置生成一个 后缀名为.rdb的文件，里面保存所有数据，因此也称为快照文件。





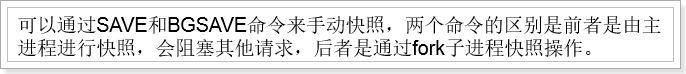
### 创建RDB快照

#### 快照的过程



#### 手动快照

如果没有触发自动快照，需要对redis执行手动快照操作，SAVE和BGSAVE都是执行手动快照，但是两者有区别：



在Redis的高级版本中，一旦服务重启，会自动做RDB

#### 快照需要注意



#### RDB文件的压缩



压缩和不压缩的优缺点：

压缩：

优点：减少磁盘存储空间

缺点：消耗CPU资源

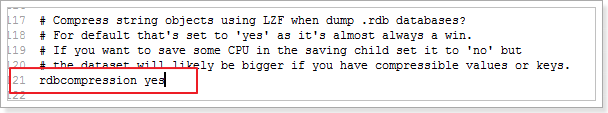
不压缩：

优点：不消耗CPU资源

缺点：占用磁盘空间多

如何选择？ 看需求、看服务器资源情况。

Redis默认是开启压缩的，配置：



### RDB快照恢复

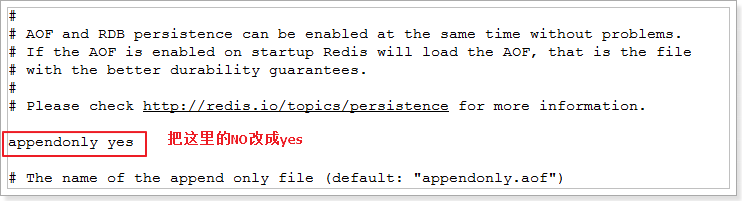


**思考：**如何关闭RDB持久化？

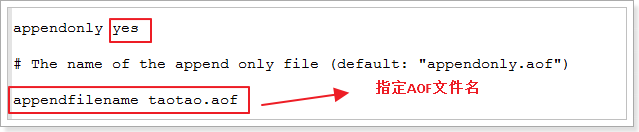
## AOF

### 开启AOF

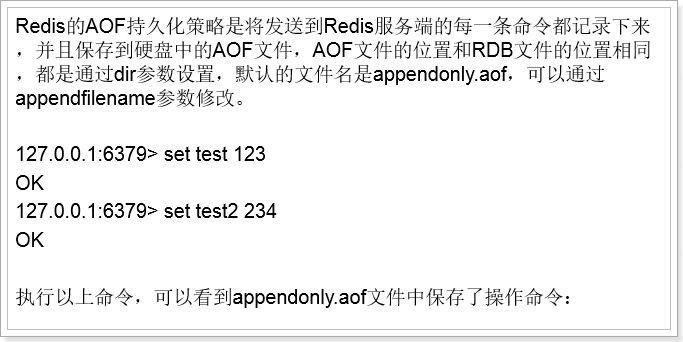
AOF与RDB是可以共存的，不过默认只开启了RDB，我们需要修改参数来开启ROF持久化：



指定AOF文件的名字：

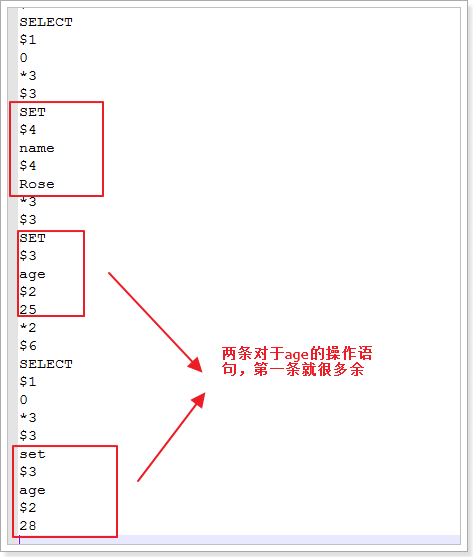


### AOF的工作方式



### 测试AOF

当客户端向服务器发送Redis命令时，Redis会将所执行的命令记录到aof文件中，如下：



当redis服务器重启后，会将执行该aof文件，达到数据恢复的目的。

### AOF文件的重写

在刚才的案例中，我们进行了两次对age的操作，都记录在了AOF文件中。其实对于一个变量的多次操作，只需要记录最后一次即可！

AOF机制会自动对AOF文件进行优化，当文件过大时，就会把文件中多余（对同一个key的重复操作）命令移除，从而保证文件体积。

自动重写的规则，可以在配置文件中设置：

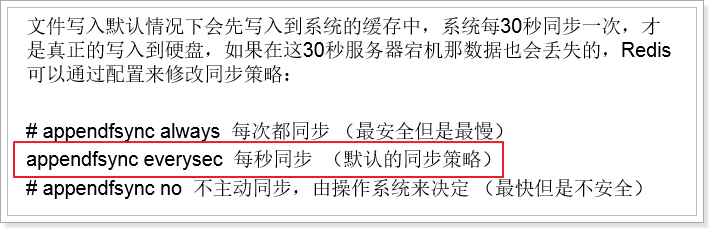


我们也可以手动进行优化：



目的：去除数据的中间执行过程，保留最终数据命令即可。

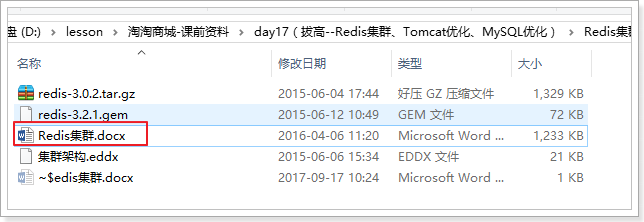
### 文件同步策略



注意：即使每秒做文件同步也可能导致数据丢失。

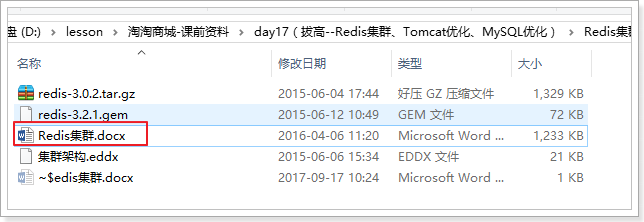
# Redis主从复制

参考课前资料中的《Redis集群.docx》



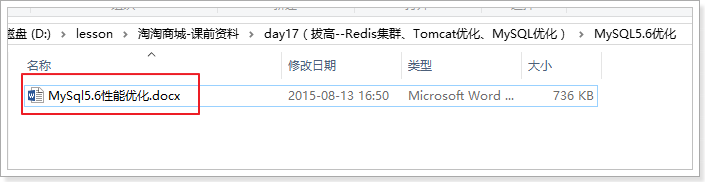
# Redis的集群

参考课前资料中的《Redis集群.docx》



# MySQL优化

参考课前资料：《MySQL5.6性能优化.docx》



# tomcat7性能优化

参考课前资料：《Tomcat7性能优化.docx》

