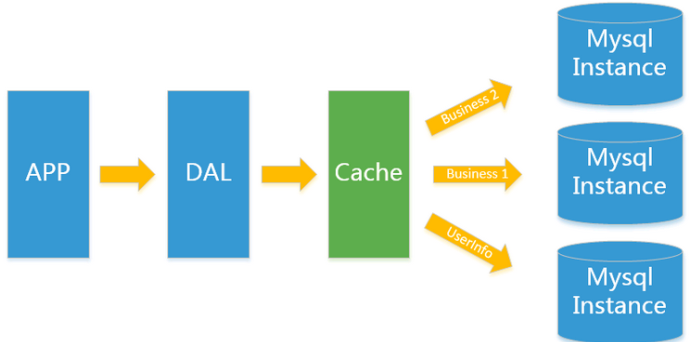
1、redis简介（sina）



**是什么**

1）Redis:REmote DIctionary Server(远程字典服务器)

2）是完全开源免费的，用C语言编写的，遵守BSD开源协议，

是一个高性能的(key/value)分布式内存数据库，基于内存运行,

并支持持久化的NoSQL数据库，是当前最热门的NoSql数据库之一,也被人们 称为数据结构服务器。

3）Redis 与其他 key - value 缓存产品有以下三个特点：

3.1）Redis支持数据的持久化，可以将内存中的数据保持在磁盘中，重 启的时候可以再次加载进行使用

3.2）Redis不仅仅支持简单的key-value类型的数据，同时还提供list， set，zset，hash等数据结构的存储

3.3）Redis支持数据的备份，即master-slave模式的数据备份

**为什么**

1）内存存储和持久化：redis支持异步将内存中的数据写到硬盘上，同 时不影响继续服务

2）取最新N个数据的操作，如：可以将最新的10条评论的ID放在 List集合

3）模拟类似于HttpSession这种需要设定过期时间的功能

4）发布、订阅消息系统

5）定时器、计数器

2、redis安装

文档

3、redis数据类型

数据存储方式 key value

kv模式

五大数据类型：string list set zset hash

4、redis基本操作

键：keys \*

string: 单键单值 key:string value:string

set/get/del/append/strlen 设置 获取 删除 追加 长度

incr/decr/incrby/decrby 要是数值才能加减

getrange/setrange 指定范围内

setex(键秒值)/setnx(set if not exist)

mset/mget/msetnx

expire key n 对已经存在的key设置过期时间，新设的会替代旧的

ttl key 查看过期时间 -1永不过期 -2已经过期

hash: 单键单对象 key :string value:object

key:{name:zhaoliying,age:10}

hset/hget/hmset/hmget/hgetall/hdel

hset key field value 设置 key 属性 值

hget key field 获取key 属性 拿到属性值

hlen

hlen key 获取属性的个数

hexists

hexists key key里面的某个field

hkeys/hvals

hkeys key 获取key里面所有的属性

hvals key 获取key里面所有的属性值

hincrby/hincrbyfloat

hincrby key field 整数

hincrbyfloat key field 小数

hsetnx

hsetnx key field 值 如果这个属性不存在 设置

列表：单键单列表 key:string value:list

lpush/rpush/lrange

lpush 从左边插入 键 值

rpush 从右边插入

lrange lrange list 0 -1 从左边开始查看范围元素

lpop/rpop

lpop 从左边每次出一个，删除

rpop 从右边每次出一个

lindex

按照索引获取值

llen

计算list 长度

lrem

lrem key 删除 n个value

ltrim

ltrim key 开始index 结束index 截取指定范围后再赋值给key

rpoplpush

rpoplpush 源列表 目的列表

从源列表右边取一个，然后从左边加入目的列表

lset

lset key index value 为列表某一个元素赋值

linsert

linsert key before/after 值1 值2

在列表的值1前面 或者后面 插入一个值2（第一个值得前面或者后面）

集合：单键单集合 key:string value :set

sadd/smembers/slsmember

scard

获取集合中的个数

srem

srem key 值 删除集合中的元素

srandmember

srandmember key 某个整数 随机出整数个值,不删除

spop key

随机出栈 一次一个 删除

smove

smove key1 key2 在key1里面的某个值

将key1里面的某个值赋值给key2

数学类：

sdiff:差集

sdiff key1 key 2 取key1里面和key2不同的值

sinter:交集

sinter key1 key2 取key1里面和key2里面相同的值

sunion:并集

sunion key1 key2 取key1里面和key2里面所有不重 复的元素

有序集合： 单键单集合key:string value:集合

在set 基础上加了一个score值（分值） 权重

之前set 是k1 v1 v2 v3

现在zset是 k1 score1 v1 score2 v2

zadd/zrange/zrange….withscores

zadd key score1 v1 score2 v2

zrange key 0 -1 / zrange key 0 -1 withscores 查看值/查看 分和值

zrangebyscore/zrangebyscore….withscores

zrangebyscore key 开始score 结束score 查看范围分的值

注意：（ 代表不包含 zrangebyscore key (60 (90

limit 2 2 从下标为2的开始拿2个

zrem

zrem key 值 删除指定一组值

zcard/zcount key score(区间)/zrank key values值

zcard key 获取组的个数

zcount key score1 score2 获取分值在score1和score2之 间的个数

zrank key 值 获取对应值的下标

zscore key 值 获取对应值的分数

如果想要清空所有的数据库数据：flushall

如果只想要清空单个数据库数据：flushdb (先进入要清空的数据库select )

如果想要知道key的类型 type key 可以查看数据的类型

5、redis事务（redis部分支持事务 ）

案例？

是什么？

本质上就是一堆命令的集合，一个事务中所有的命令都会被序列化，串行化执行，不会被其它命令插入或者加塞，在队列中一次性、顺序性、排他性的执行一系列命令

怎么做？

五个命令：

Multi 开启事务

Exec 执行事务

Discard 放弃事务

Watch 监控一个或者多个键

Unwatch 放弃监控

五个应用场合：

1. 正常执行
2. 放弃事务
3. 一只老鼠坏一锅粥 执行前出错（命令出错）
4. 谁犯法谁坐牢 执行后出错（值出错）
5. Watch监控

乐观锁/悲观锁/CAS(check and set)

案例

先监控 再事务 有人加塞 一切从新开始

先监控 再事务 知道有人加塞 取消监控 从头开始

小结：

当事务执行后所有的监控都将取消掉~ watch类似于给我们的键加上乐观锁，当在执行事务提交的时候，值被改变，那么这次的事务将会失败，整个事务都会失败

6、redis持久化策略：（官方文档）

什么是redis的持久化？

Rdb:（redis database）：

概念

fork

.rdb文件

配置文件

模拟生成rdb文件

备份（主机和备机）

Save bgsave flushall

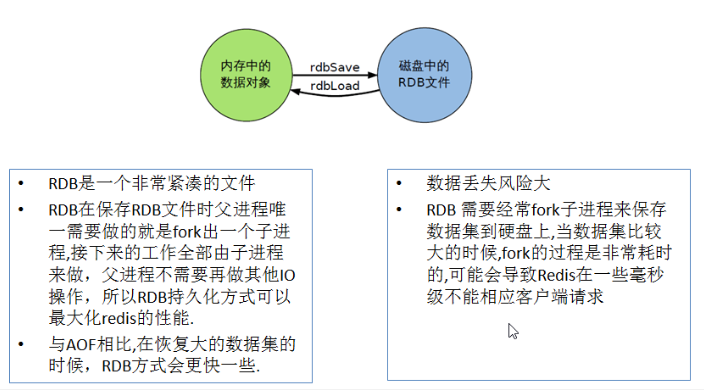
备份恢复

优点：备份速度快

缺点：数据精度不保证 fork

怎么不配置rdb

小结:



Aof:（append only file）：

为什么还有aof

概念

.aof文件

配置

Aof的启动

修复坏掉的aof文件 --fix

恢复

Aof和dump.rdb先找谁

Rewrite：

是什么？

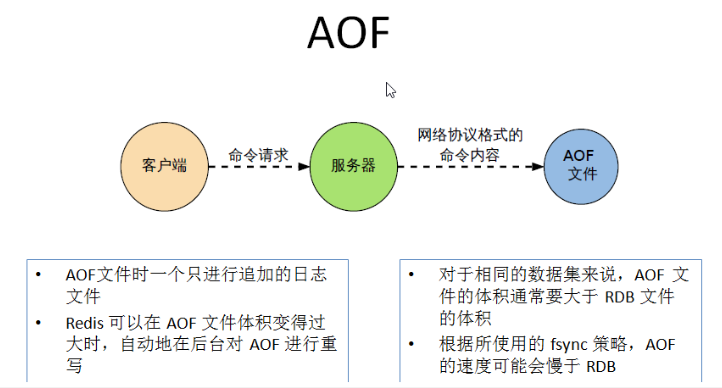
重写原理fork？

触发条件 ：上次重写1倍并且64M以上？

优点：灵活配置、数据完整性高

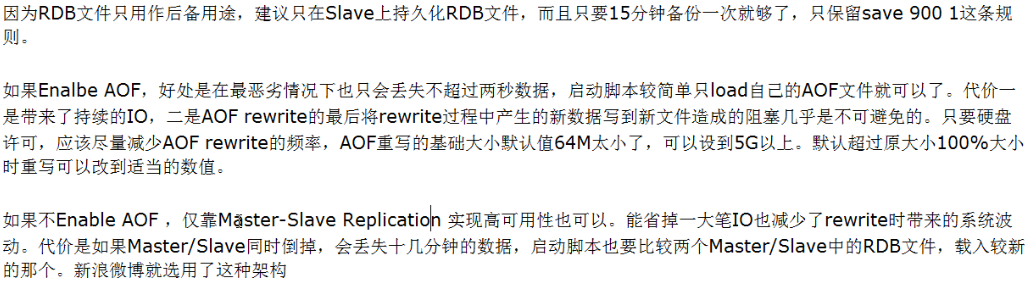
缺点：文件会biger,运行效率比rdb低

小结：



思考：如果我使用了aof可不可以不使用rdb

最终持久化策略处理：



7、主从复制

一主二从：

1）主机：进入配置文件，把bind选项修改为自己电脑的ip

2）从机：进入配置文件，把bind选项修改为从机的ip

bind 192.168.28.12

slaveof 主机ip 6379

3）当配置完成后一定记得重启主从两个redis服务

4）主机连接：redis-cli -h 主机ip

5）从机连接：redis-cli –h 从机ip

6）在主的上面去设置值，在从的上面可以获取到值，代表主从设置完成

注意：主机挂掉，从机死等 从机挂掉后，翻身还是咸鱼

代代相传：不能隔代传递，中间有一代死，后面全死

翻身做主：slaveof no one

哨兵模式：其实是翻身做主的自动化

1、新建sentinel.conf文件

2、在配置文件当中配置监控：监控主机

3、sentinel monitor 自定义哨兵名字 主机ip 主机端口 投票基数

4、sudo redis-sentinel sentinel.conf启动哨兵

8、用Pythonredis驱动包和redis数据库交互

使用redis包里面的StrictRedis模块，创建连接对象

使用连接对象可以直接使用方法增删改查

增删改查的方法和终端命令方法相同。

9、消息的订阅和发布

1）开三个终端，都去连接上redis

2）在其中两个终端上去订阅频道：subscribe 频道名(可以是多个)

3）在第三个终端上去发布消息 publish 频道名 消息信息