Redis基础

Redis是一个开源的，先进的key-value存储。

它通常被称为数据结构服务器，因为键可以包含字符串，哈希，链表，集合和有序集合。

Redis 简介

Redis 是完全开源免费的，遵守BSD协议，是一个高性能的key-value数据库。

Redis 与其他 key - value 缓存产品有以下三个特点：

Redis支持数据的持久化，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用。

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。

Redis支持数据的备份，即master-slave模式的数据备份。（主--从）

Redis 优势

性能极高 – Redis能读的速度是110000次/s,写的速度是81000次/s 。

丰富的数据类型 – Redis支持二进制案例的 Strings, Lists, Hashes, Sets 及 Ordered Sets 数据类型操作

原子 – Redis的所有操作都是原子性的，同时Redis还支持对几个操作全并后的原子性执行。

丰富的特性 – Redis还支持 publish/subscribe, 通知, key 过期等等特性。

Redis与其他key-value存储有什么不同？

Redis有着更为复杂的数据结构并且提供对他们的原子性操作，这是一个不同于其他数据库的进化路径。

Redis的数据类型都是基于基本数据结构的同时对程序员透明，无需进行额外的抽象。

Redis运行在内存中但是可以持久化到磁盘，所以在对不同数据集进行高速读写时需要权衡内存，

因为数据量不能大于硬件内存。在内存数据库方面的另一个优点是，相比在磁盘上相同的复杂的数据结构，在内存中操作起来非常简单，这样Redis可以做很多内部复杂性很强的事情。同时，在磁盘格式方面他们是紧凑的以追加的方式产生的，因为他们并不需要进行随机访问。

Redis 安装

Window 下安装

一.安装redis服务

Redis 支持 32 位和 64 位。这个需要根据你系统平台的实际情况选择，

这里我们下载 32位 Redis-Windows-32bit.zip压缩包到 C 盘，解压后，将文件夹重新命名为redis。

打开一个 cmd 窗口 使用cd命令切换目录到 C:\redis 运行 redis-server.exe redis.windows.conf 。

后面的那个 redis.windows.conf 可以省略，如果省略，会启用默认的。

这时候另再开启一个cmd窗口，原来的不要关闭，不然就无法访问服务端了。

切换到redis目录下运行 redis-cli.exe -h 127.0.0.1 -p 6379 。

按照上面的运行是可以的,但是这样就会造成两个命令行窗口都要开启,特别是服务器端,一旦关闭了那么客户端就无法使用了,那么这样可以把redis服务安装到windows服务里面

Redis是可以安装成windows服务的，开机自启动，命令如下:

redis-server --service-install redis.windows.conf

Redis并没有启动，启动命令如下：

redis-server --service-start

停止命令：

redis-server --service-stop

卸载命令：

redis-server --service-uninstall

二.安装php redis扩展

1.phpinfo检测是否开启

2.php扩展目录 extension\_dir

php\_redis.dll放置到php扩展目录下

3.php配置文件修改

extension=php\_redis.dll

4.重启服务apache服务

Ubuntu 下安装

在 Ubuntu 系统安装 Redi 可以使用以下命令:

$sudo apt-get update

$sudo apt-get install redis-server

启动 Redis

$ redis-server

查看 redis 是否启动？

$ redis-cli

以上命令将打开以下终端：

redis 127.0.0.1:6379>

127.0.0.1 是本机 IP ，6379 是 redis 服务端口。现在我们输入 PING 命令。

redis 127.0.0.1:6379> ping

PONG

以上说明我们已经成功安装了redis。

Redis 数据类型

Redis支持五种数据类型：

string（字符串）

list（列表）

set（集合）

zset(有序集合)

hash（哈希）

String（字符串）

string是redis最基本的类型，你可以理解成与Memcached一模一样的类型，一个key对应一个value。

string类型是二进制安全的。意思是redis的string可以包含任何数据。比如jpg图片或者序列化的对象 。

string类型是Redis最基本的数据类型，一个键最大能存储512MB。

实例

redis 127.0.0.1:6379> SET name "runoob"

OK

redis 127.0.0.1:6379> GET name

"runoob"

在以上实例中我们使用了 Redis 的 SET 和 GET 命令。键为 name，对应的值为 runoob。

注意：一个键最大能存储512MB。

--------------------------------------------

set 命令：设置一个键和值，键存在则只覆盖，返回ok

> set 键 值

例如： >set name zhangsan

get 命令：获取一个键的值，返回值

> get 键 例如：>get name

setnx命令：设置一个不存在的键和值（防止覆盖），

> setnx 键 值 若键已存在则返回0表示失败

setex命令：设置一个指定有效期的键和值（单位秒）

> setex 键 [有效时间] 值

例如: >setex color 10 red

不写有效时间则表示永久有效，等价于set

setrange命令：替换子字符串 (替换长度由子子串长度决定)

> setrange 键 位置 子字串

> setrange name 4 aa 将name键对应值的第4个位置开始替换

mset命令：批量设置键和值,成功则返回ok

> mset 键1 值1 键2 值2 键3 值3 ....

msetnx命令：批量设置不存在的键和值,成功则返回ok

> msetnx 键1 值1 键2 值2 键3 值3 ....

getset命令：获取原值，并设置新值

getrange命令：获取指定范围的值

>getrange 键 0 4 //获取指定0到4位置上的值

mget命令： 批量获取值

>mget 键1 键2 键3....

incr命令： 指定键的值做加加操作，返回加后的结果。

> 键 例如： >incr kid

incrby命令： 设置某个键加上指定值

> incrby 键 m //其中m可以是正整数或负整数

decr命令： 指定键的值做减减操作，返回减后的结果。

> decr 键 例如： >decr kid

decrby命令： 设置某个键减上指定值

> decrby 键 m //其中m可以是正整数或负整数

append命令：给指定key的字符串追加value，返回新字符串值的长度

>append 键 追加字串

strlen求长度

>strlen 键名 //返回对应的值。

List（列表）

Redis 列表是简单的字符串列表，按照插入顺序排序。

你可以添加一个元素到列表的头部（左边）或者尾部（右边）。

实例

redis 127.0.0.1:6379> lpush runoob redis

(integer) 1

redis 127.0.0.1:6379> lpush runoob mongodb

(integer) 2

redis 127.0.0.1:6379> lpush runoob rabitmq

(integer) 3

redis 127.0.0.1:6379> lrange runoob 0 10

1) "rabitmq"

2) "mongodb"

3) "redis"

redis 127.0.0.1:6379>

列表最多可存储 232 - 1 元素 (4294967295, 每个列表可存储40多亿)。

--------------------------------------------

list即可以作为“栈”也可以作为"队列"。

操作：

>lpush list1 "world" //在list1头部压入一个字串

>lpush list1 "hello" // 在list1头部压入一个字串

>lrange list1 0 -1 //获取list1中内容

0:表示开头 -1表示结尾。

>rpush list2 "world" //在list2尾部压入一个字串

>rpush list2 "hello" // 在list2尾部压入一个字串

>lrange list2 0 -1 //获取list2中内容

0:表示开头 -1表示结尾。

>linsert list2 before hello there

在key对应list的特定位置前或后添加字符串

>lset list2 1 "four"

修改指定索引位置上的值

>lrem list2 2 "hello" //删除前两个hello值

>lrem list2 -2 "hello" //删除后两个hello值

>lrem list2 0 "hello" //删除所有hello值

>ltrim mylist8 1 3 //删除此范围外的值

>lpop list2 //从list2的头部删除元素，并返回删除元素

>rpop list2 //从list2的尾部删除元素，并返回删除元素

>rpoplpush list1 list2 //将list1的尾部一个元素移出到list2头部。并返回

>lindex list2 1 //返回list2中索引位置上的元素

>llen list2 //返回list2上长度

Set（无序集合）

Redis的Set是string类型的无序集合。

集合是通过哈希表实现的，所以添加，删除，查找的复杂度都是O(1)。

sadd 命令

添加一个string元素到,key对应的set集合中，

成功返回1,如果元素已经在集合中返回0,key对应的set不存在返回错误。

sadd key member

实例

redis 127.0.0.1:6379> sadd runoob redis

(integer) 1

redis 127.0.0.1:6379> sadd runoob mongodb

(integer) 1

redis 127.0.0.1:6379> sadd runoob rabitmq

(integer) 1

redis 127.0.0.1:6379> sadd runoob rabitmq

(integer) 0

redis 127.0.0.1:6379> smembers runoob

1) "rabitmq"

2) "mongodb"

3) "redis"

注意：以上实例中 rabitmq 添加了两次，但根据集合内元素的唯一性，第二次插入的元素将被忽略。

集合中最大的成员数为 232 - 1(4294967295, 每个集合可存储40多亿个成员)。

-------------------------------

>sadd myset "hello" //向myset中添加一个元素

成功返回1，失败(重复)返回0

>smembers myset //获取myset中的所有元素(结果是无序的)

>srem myset "one" //从myset中删除一个one

成功返回1，失败(不存在)返回0

>spop myset //随机返回并删除myset中的一个元素

>srandmember myset //随机获取myset中的一个元素，但是不删除

> smove myset1 myset2 zhangsan:将myset1中zhangsan移动到myset2中

> scard myset1 返回myset1的个数

> sismember myset zhangsan:判断张三是否在myset中

>sdiff myset1 myset2 //返回两个集合的差集

以myset1为标准，获取myset2中不存在的。

>sdiffstore dstset myset1 myset2 ...// 返回所有集合的差集，并保存到dstset中

>sinter myset1 myset2 myset3... // 返回N个集合中的交集

>sinterstore dstset myset1 myset2 ... // 返回N个集合的交集并存储到dstset中

> sunion myset1 myset2 ...//返回所有集合的并集

> sunionstore dstset myset1 myset2// 返回所有集合的并集，并存储到dstset中

zset(sorted set：有序集合)

Redis zset 和 set 一样也是string类型元素的集合,且不允许重复的成员。

不同的是每个元素都会关联一个double类型的分数。redis正是通过分数来为集合中的成员进行从小到大的排序。

zset的成员是唯一的,但分数(score)却可以重复。

zadd 命令

添加元素到集合，元素在集合中存在则更新对应score

zadd key score member

实例

redis 127.0.0.1:6379> zadd runoob 0 redis

(integer) 1

redis 127.0.0.1:6379> zadd runoob 0 mongodb

(integer) 1

redis 127.0.0.1:6379> zadd runoob 0 rabitmq

(integer) 1

redis 127.0.0.1:6379> zadd runoob 0 rabitmq

(integer) 0

redis 127.0.0.1:6379> ZRANGEBYSCORE runoob 0 1000

1) "redis"

2) "mongodb"

3) "rabitmq"

----------------------------------------

> zadd zset 1 one 向zset中添加one，排序为1排序

> zrem zset one:删除zset中one

> zincrby zset 2 one:如果one存在，则顺序增加2，如果one不存在，那么就是2

> zrank zset one:返回one在 zset中排名(从小到大的排序)

> zrevrank zset one:返回one在zset中排名(从大到小的排序)

> zrange zset 0 -1 withscores:根据score排序（根据score从小到大排序）

> zrevrange zset 0 -1 withscores:根据score排序（根据score从大到小排序）

> zrangebyscore zset 2 3 withscores:返回集合中score在给定区间的元素（包含2和5）

> zcount zset 2 3:返回集合中给定区间的数量

> zcard zset:返回集合中元素的个数

> zscore zset one:返回one元素的score

> zremrangebyrank zset 3 3:删除集合中排名在给定区间的元素

> zremrangebyscore zset 1 2:将zset中从小到大排序结果的score在1-2之间的删除

Hash（哈希）

Redis hash 是一个键值对集合。

Redis hash是一个string类型的field和value的映射表，hash特别适合用于存储对象。

实例

127.0.0.1:6379> HMSET user:1 username runoob password runoob points 200

OK

127.0.0.1:6379> HGETALL user:1

1) "username"

2) "runoob"

3) "password"

4) "runoob"

5) "points"

6) "200"

以上实例中 hash 数据类型存储了包含用户脚本信息的用户对象。

实例中我们使用了 Redis HMSET, HGETALL 命令，user:1 为键值。

每个 hash 可以存储 232 -1 键值对（40多亿）。

------------------------------------------

hset命令：设置一个哈希表的键和值

>hset hash名 键 值

如：>hset user:001 name zhangsan

hget命令： 获取执行哈希名中的键对应值

hsetnx命令：设置一个哈希表中不存在的键和值

>hsetnx hash名 键 值 //成功返回1，失败返回0

如：>hsetnx user:001 name zhangsan

hmset命令:hmset user:001 username zhangsan age 20 sex 1 批量设置

hmget user:001 username age sex:批量获取值

>hexists user:001 name //是否存在， 若存在返回1

>hlen user:001 //获取某哈希user001名中键的数量

>hdel user:001 name //删除哈希user:001 中name键

>hkeys user:002 //返回哈希名为user:002中的所有键。

>hvals user:002 //返回哈希名为user:002中的所有值。

>hgetall user:002 //返回哈希名为user:002中的所有键和值。

Redis高级实用特性

1. 安全性：为Redis添加密码

警告：因为redis 速度相当快，所以在一台比较好的服务器下，

一个外部的用户可以在一秒钟进行150K 次的密码尝试，

这意味着你需要指定非常非常强大的密码来防止暴力破解。

1.进入配置文件：

vi /usr/local/redis/etc/redis.conf

设置：requirepass redis的密码

2. 重启服务：

# ./redis-cli shutdown 执行关闭

# ./redis-server /usr/local/redis/etc/redis.conf 启动

3. 登录（两种）

# ./redis-cli 客户端命令链接服务器

>auth 密码值 //授权后方可使用

# ./redis-cli -a 密码 //连接时指定密码来进行授权







