**目 录**

[第1章 Memcache](#_Toc17174_WPSOffice_Level1) [3](#_Toc17174_WPSOffice_Level1)

[1.1 简介](#_Toc6966_WPSOffice_Level2) [3](#_Toc6966_WPSOffice_Level2)

[1.1.1 数据缓存](#_Toc6966_WPSOffice_Level3) [3](#_Toc6966_WPSOffice_Level3)

[1.1.2 Memcache](#_Toc22689_WPSOffice_Level3) [3](#_Toc22689_WPSOffice_Level3)

[1.2 安装](#_Toc22689_WPSOffice_Level2) [3](#_Toc22689_WPSOffice_Level2)

[1.2.1 Windows系统](#_Toc1914_WPSOffice_Level3) [3](#_Toc1914_WPSOffice_Level3)

[1.2.2 Linux系统](#_Toc7199_WPSOffice_Level3) [4](#_Toc7199_WPSOffice_Level3)

[1.2.3 连接](#_Toc24611_WPSOffice_Level3) [5](#_Toc24611_WPSOffice_Level3)

[1.3 操作](#_Toc15025_WPSOffice_Level2) [5](#_Toc15025_WPSOffice_Level2)

[1.3.1 stats查看状态](#_Toc17394_WPSOffice_Level3) [5](#_Toc17394_WPSOffice_Level3)

[1.3.2 set设置](#_Toc10211_WPSOffice_Level3) [6](#_Toc10211_WPSOffice_Level3)

[1.3.3 get获取](#_Toc12579_WPSOffice_Level3) [6](#_Toc12579_WPSOffice_Level3)

[1.3.4 delete删除](#_Toc3227_WPSOffice_Level3) [6](#_Toc3227_WPSOffice_Level3)

[1.4 Php中操作](#_Toc24188_WPSOffice_Level2) [6](#_Toc24188_WPSOffice_Level2)

[第2章 Redis](#_Toc6966_WPSOffice_Level1) [7](#_Toc6966_WPSOffice_Level1)

[2.1 简介](#_Toc2649_WPSOffice_Level2) [7](#_Toc2649_WPSOffice_Level2)

[2.1.1 特点](#_Toc4412_WPSOffice_Level3) [7](#_Toc4412_WPSOffice_Level3)

[2.1.2 优势](#_Toc2093_WPSOffice_Level3) [7](#_Toc2093_WPSOffice_Level3)

[2.1.3 与其他k-v存储的不同](#_Toc30481_WPSOffice_Level3) [8](#_Toc30481_WPSOffice_Level3)

[2.2 安装](#_Toc8568_WPSOffice_Level2) [8](#_Toc8568_WPSOffice_Level2)

[2.2.1 Windows安装](#_Toc18604_WPSOffice_Level3) [8](#_Toc18604_WPSOffice_Level3)

[2.2.2 Linux安装](#_Toc27824_WPSOffice_Level3) [9](#_Toc27824_WPSOffice_Level3)

[2.2.3 RedisDesktopManager安装](#_Toc6935_WPSOffice_Level3) [10](#_Toc6935_WPSOffice_Level3)

[2.3 Php操作Redis](#_Toc10915_WPSOffice_Level2) [11](#_Toc10915_WPSOffice_Level2)

[2.3.1 链接Redis数据库](#_Toc20924_WPSOffice_Level3) [11](#_Toc20924_WPSOffice_Level3)

[2.3.2 操作](#_Toc29876_WPSOffice_Level3) [11](#_Toc29876_WPSOffice_Level3)

[2.3.3 Redis高级使用特性](#_Toc20934_WPSOffice_Level3) [13](#_Toc20934_WPSOffice_Level3)

[2.3.4 Memcache与Redis的区别](#_Toc30509_WPSOffice_Level3) [13](#_Toc30509_WPSOffice_Level3)

# 

# 第1章 Memcache

## 1.1 简介

### 1.1.1 数据缓存

由于经常要重复从数据库中取出相同的数据，这样的重复工作增加了数据库的负担，而数据缓存机制是解决这类问题的好办法。Php中数据的缓存目前我们要掌握三种(其实是两种，另一种基本框架都有自带的文件缓存，查手册即可。)第一种文件缓存,第二种Memcache，第三种Redrs。

### 1.1.2 Memcache

Memcache是一款高性能的，分布式的内存对象缓存系统。其原理就是通过在内存里面维护一个统一的巨大hash表(哈斯表)来存储而各种格式的数据，包括图像、视频、文件以及数据库检索的结果等。由于Memcache存储在内存中且使用非阻塞的网络I/O，对内部对象实现引用计数，同时使用自己的页块分配器和哈希表，因此虚拟内存不会产生碎片从而保证输出所以他的I/O数据很快(I/O速度：cpu>内存>硬盘)。

* 【特点】

Memcache的缓存是一种分布式的，可以让不同主机上的多个用户同时访问。同时对数据的I/O速度很快。

* 【分布式】分布式是一种计算机的算法,这里我们只探究分布式网络存储技术。

分布式网络存储技术是将数据分散的存储于多台独立的机器设备上。分布式网络存储系统采用可扩展性的系统结构，利用多台存储服务器分担存储负荷，利用位置服务器定位存储信息，不但解决了传统中式存储系统中单存储服务器的瓶颈问题，还提高了系统的可靠性、可用性和扩展性。

* 【缓存结构】Memcache的缓存结构：

1. 在内存中缓存数据
2. 数据形态以key->value结构
3. 安全度非常差(关机，开机缓存将清空)

* 【缺点】优缺点是相互的

1. 由于存储在内存中，因此速度很快，但同时对内存的要求也提高
2. 也由于存储在内存中，因此安全性比较差

## 1.2 安装

### 1.2.1 Windows系统

* 【安装服务】不管安装什么有要区分自己的环境是32还是64

1. 在网上下载memcache-1.2.1-win32.zip举例)解压放某个盘下，比如在c:\memcached
2. 以管理员身份进入cmd命令进，cd到解压的目录下操作
3. C:\memcached\memcached.exe -d install //安装

memcached.exe -d uninstall //卸载

memcached.exe -d start //启动

memcached.exe -d stop //停止

1. 在启动后连接

telnet 127.0.0.1 11211 //连接memcache 11211默认端口号

1. 使用quit退出

* 【安装扩展】在windows下安装php扩展

1. 首先打开phpinfo() 查看自己PHP的版本信息和操作系统是64还是32，是nts还是ts
2. 下载与之对应的php\_memcached.dll放入php文件下的ext目录下
3. 修改php.ini文件 添加extension=php\_memcache.dll
4. 重启apache
5. 查看phpinfo() 是否有memcache

* 【命令参数】启动Memcache常用参数

-p <num> 设置端口号(默认不设置为: 11211)

-U <num> UDP监听端口(默认: 11211, 0 时关闭)

-l <ip\_addr> 绑定地址(默认:所有都允许,无论内外网或者本机更换IP，

有安全隐患，若设置为127.0.0.1就只能本机访问)

-d 独立进程运行

... -d start 启动memcached服务

... -d restart 重起memcached服务

... -d stop|shutdown 关闭正在运行的memcached服务

... -d install 安装memcached服务

... -d uninstall 卸载memcached服务

-u <username> 绑定使用指定用于运行进程<username>

-m <num> 允许最大内存用量，单位M (默认: 64 MB)

-P <file> 将PID写入文件<file>，可以使得后边进行快速进程终止, 需要与-d 一起使用

-M 内存耗尽时返回错误，而不是删除项

-c 最大同时连接数，默认是1024

-f 块大小增长因子，默认是1.25

-n 最小分配空间，key+value+flags默认是48

-h 显示帮助

* 【注意】在Windows下安装需要注意以下细节

1.管理员身份进入win10 右键win标识->Windows PowerShell(管理员)(A)->输入：start cmd

2.如果输入telnet命令的时候 出现telnet不是内部命令,修改操作如下:

windows系统->控制面板->功能->启动和关闭Windows功能->勾选 telnet客户端

### 1.2.2 Linux系统

Liunx下安装操作：先1安装再2安装 顺序可以改变。

* 【安装扩展】先安装memcache扩展模块pecl-memcache-php7.zip(php7版本)

1. 解压：unzip pecl-memcache-php7.zip
2. 进入目录：cd pecl-memcache-php7
3. 准备编译环境(生成configgure)执行：/uer/local/php/bin/phpize
4. 检测配置：./configure --with-php-config=/usr/local/php/bin/php-config
5. 编译和安装：make && make install (安装成功后就会生成一个memcache.so文件/usr/local/php/lib/php/extensions/no-debug-non-zts-20151012/)
6. 修改php配置文件php.ini ：vim /usr/local/php/etc/php.ini 在配置文件中添加

extension\_dir='/usr/local/php/lib/php/extensions/no-debug-non-zts-20151012/';

extension="memcache.so";

1. 重启apache服务即可(查看phpinfo()。

* 【安装服务】安装缓存服务，rpm包安装memcached-1.4.4-3.el6.i686.rpm在镜像文件中

1. 挂着ISO镜像文件：mount /dev/sr0/media
2. 安装memcached：yum -y install memcached
3. 添加memcache用户：useradd memcache
4. 启动memcache服务：memcached -d -m 128 -l 127.0.0.1 -p 11211 -u memcache
5. 查看端口是否开启：netstat -tunp
6. 设置开机自启动：chkconfig memcached on

### 1.2.3 连接

Memcached的连接：telnet 127.0.0.1 11211

## 1.3 操作

### 1.3.1 stats查看状态

Stats这里显示状态信息

* 【参数】

STAT pid 1552 服务进程的进程ID

STAT uptime 3792 服务从启动到当前所经过的时间，单位是秒。

STAT time 1262517674 服务器所在主机当前系统的时间，单位是秒。

STAT version 1.2.6 组件的版本。这里是我当前使用的1.2.6。

STAT pointer\_size 32 服务器所在主机操作系统的指针大小，一般为32或64.

STAT curr\_items 1 表示存放当前的所有缓存对象的数量。不包括已经从缓存中删除的对象。

STAT total\_items 2 表示从启动到当前，系统存储过的所有对象数量，包括已经删除的对象。

STAT bytes 593 表示系统存储缓存对象所使用的存储空间，单位为字节。

STAT curr\_connections 2 表示当前系统打开的连接数。

STAT total\_connections 28 表示从memcached服务启动到当前时间，系统打开过的连接的总数。

### 1.3.2 set设置

set()添加或修改一条数据

* 【格式】 <命令> <键> <标记> <有效期> <数据长度>

<键> -key 是发送过来指令的key内容,最大不能超过250个字符

<标记> -flags，是调用set指令保存数据时候的flags标记，是一个16位无符号的整数

<有效期>是数据在服务器上的有效期限，如果是0，则数据永远有效，单位是秒

<数据长度>单位是字节，最大不能超过1024kb

* 【实例】

set 键名 标记 有效时间 数据的长度

set love 10 0 3

* 【注意】发送完数据以后，客户端一般等待服务器端的返回。

STORED:数据保存成功

NOT\_STORED 数据保存失败，是因为服务器端这个数据key已经存在

### 1.3.3 get获取

* 【格式】

get 键名

### 1.3.4 delete删除

delete 删除缓存

* 【格式】

delete 键名

* 【补充】

flush\_all 清空所有缓存

## 1.4 Php中操作

Php操作Memcache的使用的前提是该Php是否开启Memcache功能，配置文件是否修改

* 【参数】Php Memcache类中的参数

connect(ip,port) 连接一台memcache服务器

addServer(ip,potr) 添加一台memcache服务器

getExtendedStats() 获取连接池中所有的服务器信息

getServerStatus(host,port) 获取指定服务器的连接状态

add(key,value,tag,expires) 添加一条数据

replace(key,value,tag,expires) 修改一条数据

set(key,value,tag,expires) 添加或修改一天数据

get(key) 获取一条数据

delete(key) 删除指定的一条数据

flush() 删除全部

close() 关闭连接

getStats() 获取memcache服务器信息 返回数组

getVersion() 获取memcache服务器版本号

* 【实例】

//实例化Memcache

$memcache = new Memcache();

//连接Memcache

$memcache->connect('127.0.0.1','11211');

$memcache->addServer('127.0.0.1','11211'); //添加服务器

//保存数据

$memcache->set('name','xiaohigh',MEMCACHE\_COMPRESSED,60);

//获取数据

$memcache->get('name');

//清除所有数据

$memcache->flush();

# 第2章 Redis

## 2.1 简介

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，链表，集合，有序集合和哈希。

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。

### 2.1.1 特点

Redis 与其他key-value缓存产品有以下三点的特性：

1. 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。
2. 不仅仅支持简单的key-value类型的数据，同时提供string、list、set、zset、hash等数据结构的存储。
3. 支持数据的备份，即master-slave模式的数据备份。

### 2.1.2 优势

1.性能级高 - Redis能读的速度是110000次/s，写的速度是81000次/s。

2.数据类型丰富

3.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。

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

### 2.1.3 与其他k-v存储的不同

1.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。

2.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。

3.Redis运行在内存中但是可以持久化到磁盘，所有在对不同数据库集进行高速读写时需要权衡内存。

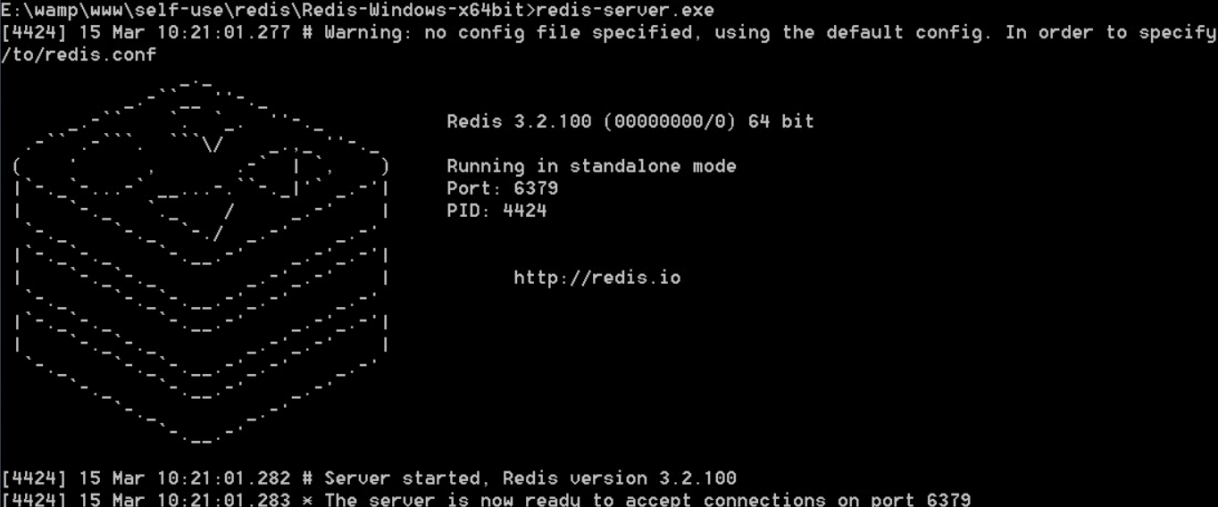
4.因为数据量不能大于硬件内存，在内存数据库方面的另一个优点是，相比在磁盘上相同的复杂的数据结构，在内存中操作起来非常简单，这样Redis可以做很多内部复杂性很强的事情。

## 2.2 安装

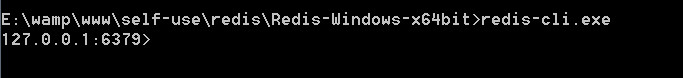
### 2.2.1 Windows安装

* 【安装服务】Redis支持32和64根据自己系统平台的实际情况选择。(拿32来举例)

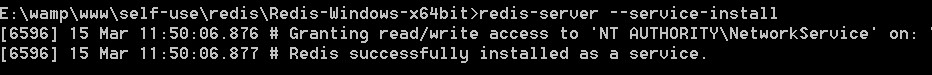
1. 解压Redis-Windows-32bit.zip压缩包到C盘，解压后将文件夹重命名为redis。
2. 打开cmd窗口使用cd命令切换到redis目录下 redis-server.exe (出现以下界面表示成功)



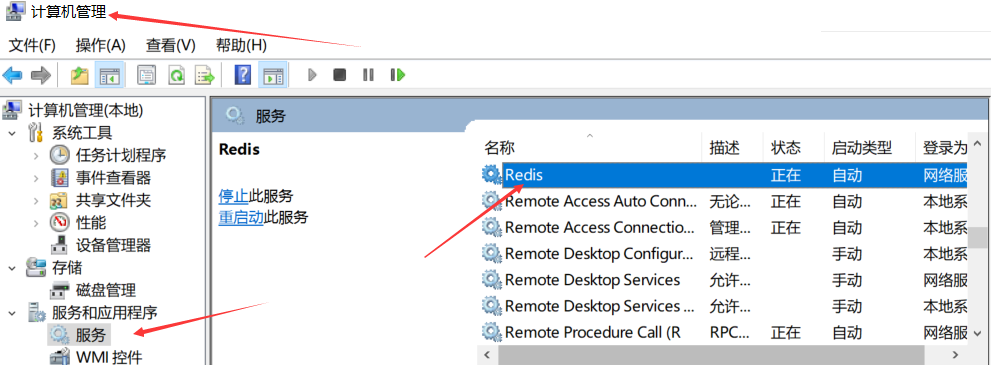
1. 再开启一个cmd窗口，原窗口不要关闭，不然就无法访问服务器了,再次切换到目录下，执行输出redis-cli.exe -h 127.0.0.1 -p 6379



1. 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，一旦关闭那么客户端就无法使用了，那么可以把redis服务按照到windows服务里面,同样cms命令进入redis目录，输入redis-server --service-install



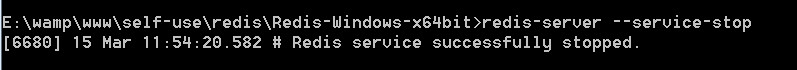
按照后右键计算机->管理->服务和应用程序->查看redis服务表名按照成功。



1. 开启redis服务 redis-server --service-start

redis-start

1. 关闭redis服务 redis-server --service-stop



1. 卸载命令 redis-server --service-uninstall

redis-uninstall

* 【安装扩展】跟memcache安装类似

1. phpinfo检测是否开启。(查看自己PHP的版本vc几和TS还是NTS的环境)
2. 根据自己的版本找到相应的php\_redis.dll文件，后放置php扩展目录
3. 修改php配置文件修改 extension=php\_redis.dll
4. 重启服务apache服务

### 2.2.2 Linux安装

* 【安装服务】

1. 将redis-3.2.1.tar.gz上传(http://redis.io/download可下载)
2. 解压：tar zxvf redis-3.2.1.tar.gz
3. 进入目录cd redis-3.2.1
4. 编译：make make完后redis-3.2.1目录下会出现编译后redis服务程序redis-server，还有用于测试的客户端程序redis-cli，两个程序位于安装目录src目录下。
5. 启动redis服务前准备，修改配置文件：vim redis.conf：

第128行 daemonize no 由no改为yue

1. 再启动redis服务进程后，进入src目录启动redis服务：

cd src

./redis-server ../redis.conf

1. 开启客户端服务:（cd src） ./redis-cli

* 【简化开启redis服务操作】以上开启服务到开启客户端很是麻烦，我们可以简化操作

1. 新建redis目录 mkdir /usr/local/apache2/htdocs/redis
2. 进入目录中复制需要的文件

cd redis-3.2.1

cp redis.conf src/redis-server src/redis-cli /usr/local/apache2/htdocs/redis

1. 进入该目录，然后启动在一个目录下操作 cd /usr/local/apache2/htdocs/redis

* 【安装扩展】

1. 下载扩展包地址：<http://github.com/phpredis/phpredis/tree/php7>
2. 将phpredis-php7.zip 文件上传。
3. 解压文件后进入目录：unzip phpredis-php7.zip cd phpredis-php7（进入该目录）
4. 由于目录中没有configure配置环境，所以需要编译下
5. /usr/local/php/bin/phpize (来准备一个编译环境)
6. 配置：
7. ./configure --with-php-config=/usr/local/php/bin/php-config
8. 编译和安装：make && make install
9. 安装成功后就生成一个redis.so文件
10. 修改php配置文件
11. vim /usr/local/etc/php.ini
12. 722行 extension\_dir='/usr/local/php/lib/php/extensions/no-debug-non-zts-20151012/';(如果配置文件中有就不必在加)。
13. 724行(大概) extension=”redis.so”;
14. 重启apache服务即可(查看phpinfo())

### 2.2.3 RedisDesktopManager安装

RedisDesktopManager是图形软件可以时事看见你对redis操作的数据。安装很简单下一步操作即可,但window vista至window 10下使用php远程代码来连接CentOS下的Redis Server是如果出错，则需要修改配置文件,修改配置文件中62行的bind 127.0.0.1 为自动获取IP:0.0.0.0，Server主机，就能通过Server获取的IP和port:6379。



## 2.3 Php操作Redis

Php搭配Redis非关系型数据库

### 2.3.1 链接Redis数据库

<?php

$redis = new Redis(); //创建Redis对象

$redis->connect(‘127.0.0.1’,6379); //链接Redis服务器

$redis->select(“1”); //选择数据库 Redis一共有16个数据库从0-15

?>

### 2.3.2 操作

* 【数据类型】

Redis支持五种数据类型：

string（字符串），list（列表），set（集合）及zset(sorted set：有序集合)，hash（哈希）

* 【String】字符串是redis最基本的类型，一个键最大能存储512MB。你可以理解成与Memcache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，可包含任何数据如：jpg图片或者列化的对象。
* 【方法】这里我之总结Php下的操作方法。

1.写入：$redis->set(‘name’,’pass’);

2.获取：$redis->get(‘name’);

3.检测Key是否存在：$redis->exists(‘name’);

4.写入并添加时效：$redis->setex(‘name’,10,’Hyt’); //参数1：键，参数2：有效时，参数3：值

5.删除：$redis->del(‘name’);

6.先获取后设置：$redis->getset(‘age’,18);

7.递增：$redis->incr(‘age’); //执行一次增加1

$redis->incrBy(‘age’,5);//执行一次增加参数2。

8.递减：$redis->decr(‘age’); //执行一次减少1

$redis->decrBy(‘age’,5);//执行一次减少5

* 【list】Redis列表是简单的字符串列表，安装插入顺序排列。你可以添加一个元素到列表头部(左边)或者尾部(右边)。
* 【方法】这里我之总结Php下的操作方法(方法基本上都是l开头)。

1.左侧添加，返回链表的长度：$redis->lpush(‘list’,’hello’);

2.右侧添加，返回链表的长度：$redis->rpush(‘list’,’world’);

3.在指定位置添加：$redis->linsert(‘list’,’before’,’abc’,bca);

4.返回链表的长度：$redis->llen(‘list’);

5.返回某个区间的值：$redis->lrange(‘list’,0,3); //-1表示最后

6.返回指定索引上的值：$redis->lindex(‘list’,2);

7.修改指定索引上的值：$redis->lset(‘list’,0,100);

8.删除：$redis->lpop(‘list’); //左侧删除一个元素 $redis->rpop(‘list’);//右侧

9.删除健名的元素：$redis->lrem(‘list’,’name’);

10.移动元素：$redis->rpoplpush(‘list’,’list-1’);

* 【set】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0(1)。
* 【方法】这里我之总结Php下的操作方法(方法基本上都是s开头)。

1.添加元素：$redis->sadd(‘set’,’Hyt’);

2.获取集合中的所有元素：$redis->smembers(‘set’);

3.删除某个元素：$redis->srem(‘set’,’hyt’);

4.将某个值从该集合移动到另一个集合：$redis->smove(‘set’,’set-1’,’hyt’);

5.返回集合元素个数：$redis->scard(‘set’);

6.随机取出一个元素：$redis->srandmemeber(‘set’);

7.检测值是否在该set中：$redis->sismember(‘set’,’hyt’);

8.交集：$redis->sinter(‘set’,’set-1’);

9.并集：$redis->sunion(‘set’,’set-1’);

10.补集：$redis->sdiff(‘set’,’set-1’);

* 【zset】Redis zset好set一样也是string类型元素的集合，且不允许重复成员。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（score）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。

* 【方法】

1.添加元素：$redis->zadd(‘zset’,1,’hyt’);//参数1：健名，参数2：权重，参数3：值

2.增加某个值的权重：$redis->zincrby(‘zset’,3,’hyt’);//在原有的基础上增加权重

3.获取整个集合的结果：$redis->zrange(‘zset’,0,-1);//正序 zrevrange()//倒序

4.获取某个值的权重：$redis->zscore(‘zset’,’hyt’);

5.获取集合中的元素个数：$redis->zcard(‘zset’);

6.返回指定元素在集合中的位置：$redis->zrank(‘zset’,’hyt’);

7.根据元素值的权重范围进行删除：$redis->zremrangebyscore(‘zset’,0,4);

8.根据元素的位置范围进行删除：$redis->zremarangebyrank(‘zset’,0,4);

9.将两个集合的交集存入到一个集合中：$redis->zinterstore(‘zset’,[‘zset-1’,’zset-2’]);

将两个集合的并集存入到一个集合中：$redis->zunionstore(‘zset’,[‘zset-1’,’zset-2’]);

* 【hash】Redis hash是一个键值对集合。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。
* 【方法】

1. 设置值：$redis->hset(‘hash’,’name’,’hyt’);
2. 获取某个值：$redis->hget(‘hash’,’name’);
3. 获取当前哈希下的所有内容：$redis->hgetall(‘hash’);
4. 检测哈希中健名是否存在：$redis->hexists(‘hash’,’name’);
5. 删除哈希中的指定健名：$redis->hdel(‘hash’,’name’);
6. 返回哈希中的元素个数：$redis->hlen(‘hash’);
7. 批量获取：$redis->hmget(‘hash’,[‘name’,’age’,’sex’]);
8. 批量设置：$redis->hmset(‘hash’,[‘name’=>’hyt’,’age’=>’18’,’sex’=>’男’]);
9. 对哈希下的某个值进行递增：$redis->hincrby(‘hash’,’age’,2);//递减是负数

* 【通用】方法

1. 获取健名：$redis->keys(‘u\*’); //\*通配符匹配多个符号 //?匹配单个符号
2. 设置Key生命周期：$redis->expire(‘name’,100); |$redis->setTimeout(‘a’,100)
3. 返回剩余时间：$redis->ttl(‘age’);
4. 删除Key有效时间：$redis->persist(‘age’);
5. 设置AUTH权限：(设置)config set requirepass 密码

(清空)coonfig set requirepas ‘’

1. 验证密码：$redis->auth(密码);
2. 移动一个KEY-value到另一个库:$redis->move(‘age’,3);
3. 重命名一个Key：$redis->rename(‘age’,’class’);
4. 返回key所指向的value的数据类型(命令行模式下)：type 健名
5. 清空数据库:清空当前数据库->$redis->flushDB();

清空所有的数据库->$redis->flushAll();

### 2.3.3 Redis高级使用特性

* 【添加密码】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

requirepass redis的密码

1. 重启服务:

#./redis-cli shurdown 执行关闭  
#./redis-server /usr/local/redis/etc/redis.conf 启动

1. 登陆(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

### 2.3.4 Memcache与Redis的区别

Memcached是一个自由开源的，高性能，分布式内存对象缓存系统。

Redis是一个开源的使用ANSI C语言编写、遵守BSD协议、支持网络、可基于内存可持久化的日志型、key-value数据库，并提供多种语言的API。

Redis通常被称为数据结构服务器，因为值可以是字符串、列表、集合、有序集合、哈希等类型。

其区别在于：

1. 运行:

Memcached是多线程，非阻塞IO复用的网络模型。

Redis使用单线程的IO复用模型。

1. 持久化:

Memcached不可以持久化。

Redis支持持久化保存。

1. 数据类型:

Memcached数据类型单一。

Redis数据类型:字符串、链表、集合、有序集合、哈希

1. 事务

Memached不支持事务。

Redis支持事务。