Django使用redis缓存服务器

redis相信大家都很熟悉了，和memcached一样是一个高性能的key-value数据库，至于什么是缓存服务器，度娘都有很明白的介绍了，我在这里就不一一介绍了。
那我们一般什么情况下才会使用缓存服务器呢？可不是什么情况都需要的哦，一般来说是在需要频繁对一个字段读取的时候才会需要将这个字段放入到缓存服务器上，而且由于key-value数据库一般只是放很简单的数据，所以在选择保存的对象的时候要注意选择好。
下面我就来介绍如何在Django中配置使用redis数据库，首先是先安装redis了，在Ubuntu中执行下面这句命令：
#安装Redis服务器端

```
sudo apt-get install redis-server
```

然后为了能在Django中使用redis，还需要安装redis for Django的插件：

```
pip install django-redis
```

这是一个开源的项目，github地址是https://github.com/niwibe/django-redis，感谢作者。
那么现在就是在Django的settings中配置了。

```
CACHES = {
    'default': {
        'BACKEND': 'redis_cache.cache.RedisCache',
        'LOCATION': '127.0.0.1:6379',
        "OPTIONS": {
            "CLIENT_CLASS": "redis_cache.client.DefaultClient",
        },
    },
}
REDIS_TIMEOUT=7*24*60*60
CUBES_REDIS_TIMEOUT=60*60
NEVER_REDIS_TIMEOUT=365*24*60*60
```

其实只是需要CACHES中的那几条就可以了，后面这三句可以不需要的，只是我后面的例子里需要用到，我就在这里配置了。
好了，现在连接和配置都已经完成了，那么在项目中该如何使用呢？接下来看下面这段例子吧。

```
from django.conf import settings
from django.core.cache import cache
#read cache user id
def read_from_cache(self, user_name):
    key = 'user_id_of_'+user_name
```

```
    value = cache.get(key)
    if value == None:
        data = None
    else:
        data = json.loads(value)
    return data
#write cache user id
def write_to_cache(self, user_name):
    key = 'user_id_of_'+user_name
    cache.set(key, json.dumps(user_name), settings.NEVER_REDIS_TIMEOUT)
```

通过上面的这两个方法就可以实现对redis的读取操作了，只需要将需要的字段当参数传入到方法中就好了。

那么之前提到的memcached呢？其实也是一样的配置：

```
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache'
        'LOCATION': '127.0.0.1:11211',
    }
}
```

当然用法也是和我上面的例子是一样的了。其实对于redis这样的缓存服务器来说，配置都是很简单的，而具体的使用也不难，官网上面也有很多简单明了的例子可以供我们参考，只有一点需要注意的，那就是对于要将什么样的信息保存到redis才是我们真正需要关心的。

Django中redis的使用方法(包括安装、配置、启动)

一、安装redis：

1.下载：

wget http://download.redis.io/releases/redis-3.2.8.tar.gz

2.解压

tar -zxvf redis-3.2.8.tar.gz

3.复制，放到/usr/local目录下

sudo mv ./redis-3.2.8 /usr/local/redis

4.进入到redis目录下

cd /usr/local/redis/

5.生成

sudo make

6.测试，时间会比较长

sudo make test

7.安装，将redis的命令安装到usr/local/bin/目录中

sudo make install

8.安装成功后，进入usr/local/bin/目录中查看：

cd /usr/local/bin

```
-rwxr-xr-x  1 root root 2683024 Feb 15 16:37 redis-benchmark
-rwxr-xr-x  1 root root   28848 Feb 15 16:37 redis-check-aof
-rwxr-xr-x  1 root root 5548376 Feb 15 16:37 redis-check-rdb
-rwxr-xr-x  1 root root 2859496 Feb 15 16:37 redis-cli
lrwxrwxrwx  1 root root      12 Feb 15 16:37 redis-sentinel -> redis-server
-rwxr-xr-x  1 root root 5548376 Feb 15 16:37 redis-server
```

9.将配置文件复制放到/etc/redis目录下：

sudo cp /usr/local/redis/redis.conf /etc/redis/redis.conf

可能刚开始etc下没有redis目录，需要自己手动创建一个redis文件夹。

以上在第6步的时候可能会遇到这样的问题：

You need tcl 8.5 or newer in order to run the Redis testmake: *** [test] Error 1

解决办法：安装tcl

```
1  wget http://downloads.sourceforge.net/tcl/tcl8.6.1-src.tar.gz
2  sudo tar xzvf tcl8.6.1-src.tar.gz -C /usr/local/
3  cd /usr/local/tcl8.6.1/unix/
4  sudo ./configure
5  sudo make
6  sudo make install
```

二、 配置redis：

配置信息在/etc/redis/redis.conf下，打开sudo vi /etc/redis/redis.conf。

核心配置选项：

绑定ip：bind 127.0.0.1

端口号：port 6379

是否以守护进程运行：daemonize yes　　必须改为yes

数据库文件：dbfilename dump.db

数据库文件存储路径：dir /var/lib/redis　　可改可不改，改的话提前创建好文件夹

日志文件：logfile /var/log/redis/redis-server.log　　必须要改，提前创建好文件夹

数据库，默认有16个：database 16

主从复制：slaveof

配置小结：主要更改两块：1，守护进程；2.日志文件路径

三、启动redis：

1.根据配置文件启动redis服务器

sudo redis-server /etc/redis/redis.conf

2.启动redis客户端：

redis-cli

3.输入ping命令测试：

127.0.0.1:6379> pingPONG

4.关闭redis服务器：

ps aux|grep redis # 查看redis进程号kill -9 pid redis进程号 # 关闭redis服务器

5.切换数据库：默认有16个，通过0-15来标识，默认是第一个数据库0号数据库。

select n

四、redis的数据操作：

1.redis的存储格式为key-value格式。key是字符串类型，value的类型有5种：string、hash、list、set、zset。

2.redis中关于键的常见的操作：

2.1查看左右的键：keys *

2.2查看某个键是否存在，存在返回1不存在返回0：exists key1

2.3查看键的值对应的数据类型：type key1

2.4删除键值对：del key1 key2

2.5设置键的过期时间，如果没有指定默认一直存在：expire key seconds

2.6查看键的有效时间：ttl key1

2.7清空数据库：flushall

3.string类型的基本操作：

3.1保存

set key value　　保存单个键值对 mset key1 value1 key2 value2　　保存多个键值对 setex key seconds value　　设置键值和过期时间 append key value追加值

3.2获取

get key　　获取单个 mget key1 key2　　获取多个

3.3删除

del key

4 hash类型的基本操作：

4.1保存

```
hset key field value #设置单个属性

hmset key field1 value1 field2 value2 ... #设置多个属性
```

4.2获取

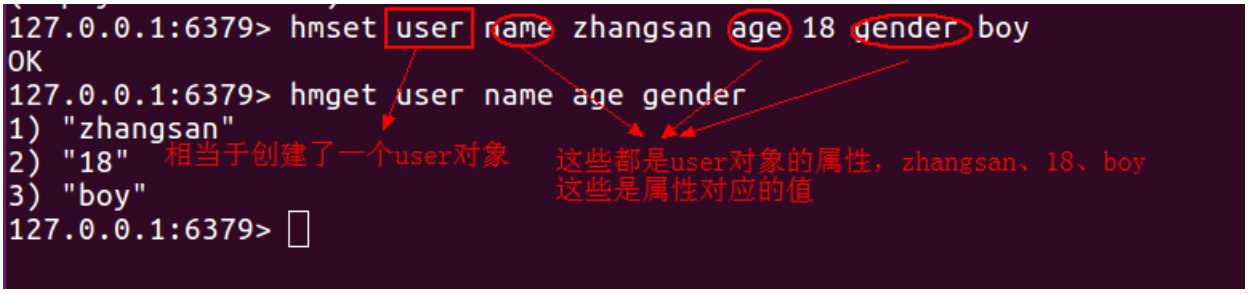
```
hkeys key # 获取指定键的所有属性
hget key field # 获取单个属性的值
hmget key field1 field2 ... # 获取多个属性的值
hvals key # 获取所有属性的值
```

4.3删除

```
del key # 删除整个hash的键和值
hdel key field1 field2 ... # 删除属性和属性对应的值
```

4.4关于hash类型的个人理解：

可以将hash类型理解为用来存储对象：



5 list类型的基本操作：

5.1保存

```
lpush key value1 value2 ... #从左侧依次插入数据
rpush key value1 value2 ... #从右侧依次插入数据
linsert key before或after 现有元素 新元素 #从指定元素的前或后插入新元素
```

5.2获取

lrange key start stop

start、stop为元素的下标索引，从左侧开始，第一个元素为0，-1标识最后一个元素。获取所有的元素：lrange key 0 -1

5.3删除指定元素

lrem key count value

将列表中前count次出现的值为value的元素移除。

count > 0: 从头到尾移除

count < 0: 从尾到头移除

count = 0: 移除所有

6.set类型的基本操作

特点：无序集合、元素唯一性不重复、没有修改操作

6.1增加元素

sadd key member1 member2 ...

6.2获取元素

smembers key # 返回所有元素

6.3删除指定元素

srem key member1 member2 ...

7.zset类型的数据操作

特点：有序集合、元素唯一性不重复、没有修改操作、每个元素都会关联一个double类型的权重，根据权重从小到大排列

7.1增加

zadd key score1 member1 score2 member2 ...

7.2获取

```
zrange key start stop    # 根据索引获取

zrangebyscore key min max    # 获取权重在min和max之间的数据
zscore key member    # 返回成员member的score值
```

7.3删除

```
zrem key member1 member2 ... # 删除指定元素
zremrangebyscore key min max #删除权重在指定范围的元素
```

五、redis与python进行交互：

1.在虚拟环境中安装redis包：

```
pip install redis
```

2.调用模块：

```
from redis import StrictRedis
```

3.创建对象：

```
sr = StrictRedis(host='localhost', port=6379, db=0)    # 默认就是这样的值，
```

4.用sr这个对象操作redis，将redis中的命令当作函数让sr调用就ok。

六、Django框架中session存储到redis中的配置

默认情况下session是存储在数据库中的，但是当用session保存用户的状态时，用户频繁的访问服务器，会增大数据库的压力，也会降低用户访问的速度。为了解决这个问题将session存储到redis中。

第一种配置方法：（不使用Django中session默认的存储方式，直接将session存储的地方配置到redis中）

```
# 1.在虚拟环境中安装包
pip install django-redis-sessions==0.5.6

# 2.在Django项目的settings文件中增加下面的配置
SESSION_ENGINE = 'redis_sessions.session'
SESSION_REDIS_HOST = 'localhost'
SESSION_REDIS_PORT = 6379
SESSION_REDIS_DB = 2
SESSION_REDIS_PASSWORD = ''
SESSION_REDIS_PREFIX = 'session'
```

第二种配置方法：（先将Django中的缓存设置为redis，然后将session的存储地方设置为Django的缓存中）

```
#1.先在虚拟环境中安装包
pip install django_redis

#2. 设置redis作为django的缓存设置
CACHES = {
 "default": {
  "BACKEND": "django_redis.cache.RedisCache",
  # 把这里缓存你的redis服务器ip和port
  "LOCATION": "redis://172.16.179.142:6379/12",
```

```
  "OPTIONS": {
   "CLIENT_CLASS": "django_redis.client.DefaultClient",
  }
 }
}

# 3.设置redis存储django的session信息
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
SESSION_CACHE_ALIAS = "default"
```

以上这篇Django中redis的使用方法(包括安装、配置、启动)就是小编分享给大家的全部内容了，希望能给大家一个参考，也希望大家多多支持脚本之家。