Redis是用单线程来处理多个客户端的访问的,目前,几乎所有的主流语言都有redis客户端(<http://redis.io/clients>)

# Java客户端

Java有很多个redis客户端,这里讲一下Jedis

Jedis是java的第三方开发包,在java中获取第三方开发包的方法主要有两种

一种是下载目标版本的jar包加入到项目中

一种是使用集成构建工具,如maven,gradle将目标版本的包加入到项目中

通常使用第二种

以maven为例,在项目中加入以下的依赖即可:

<dependency>

<groupId>redis.clients</groupId>

<artifactId>fedis</artifactId>

<version>2.8.2</version>

</dependency>

Jedis本身不带序列化工具,例如xml/json/protobuf/thrift等.

Jedis客户端连接redis服务器的方式有两种:

一种是直连方式:

Jedis jedis=new Jedis(“domain”,port);

Jedis.set(“key”,”value”);

String value =jedis.get(“key”);

每次都会建立TCP连接，使用完再断开。

一种是连接池的方式,使用JedisPool类作为Jedis的连接池，同时用Apache的通用对象池工具common-pool作为资源的管理工具：

//common-pool连接池配置，这里使用默认配置

GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();

//初始化Jedis连接池

JedisPool jedis = new JedisPool(poolConfig,”domain”,port);

Jedis jedis = null;

try{

//从连接池获取jedis对象

jedis= jedisPool.getResource();

//执行操作

jedis.get(“key”);

}catch(Exception e)

{

logger.error(e.getMessage(),e);

}finally{

If (jedis!=null){

//如果使用的是JedisPool，close操作不是关闭连接而是归还连接池

jedis.close()

}

}

Jedis的pipeline和lua脚本具体见教程。

# Python客户端

Redis-py

Import redis

client= redis.StrictRedis(host=’127.0.0.1’,port=6379)

key = “hello”

setResult = client.set(key,”python-redis”)

value = client.get(key)

#pipeline

Import redis

def mdelete(keys)

client = redis.StrictRedis(host=’127.0.0.1’,port=6379)

#生成pipeline，禁用事务

pipeline = client.pipeline(transaction=False)

#将命令封装到pipeline中，此时命令并没有被执行

for key in keys

print pipeline.delete(key)

#执行pipeline

return pipeline.execute()

# 客户端管理

《Redis开发与运维》第4.4节