# 第六章 Python爬虫分布式

本节所讲内容：

6.1 linux安装scrapy

6.2 使用scrapy-redis进行分布式编程

上述是我们编写的单机版的爬虫，我们可以实现获取糗事百科所有的图片链接，但是有以下问题

1. 它效率太低
2. 它提升效率有瓶颈
3. 单ip请求频率太高容易，容易服务器封掉
4. 使用代理ip还有ua池也有单机资源局限问题

所以针对这样的需求，我们采用scrapy redis分布式，发动多台服务器同时爬取一个项目。

部署scrapy-redis环境

分布式需求

Windows 1台

Linux 1台 (centos7.6) -- 虚拟机

两台可以相互ping通

两台都安装Python 3.7.2

保证环境相同。

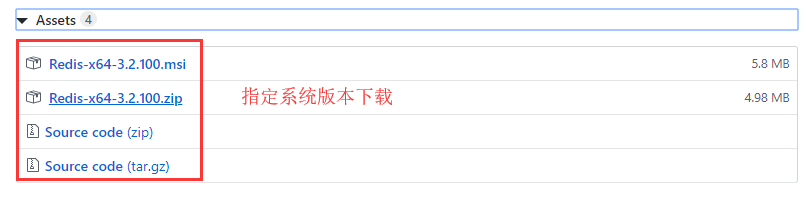
## 6.1 linux安装scrapy

### 6.1.1 Redis数据库

1、Window 下安装

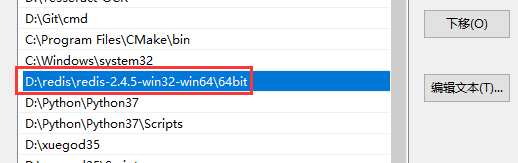
下载地址：<https://github.com/MSOpenTech/redis/releases>。

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 Redis-x64-xxx.zip压缩包到 C 盘，解压后，将文件夹重新命名为 redis。

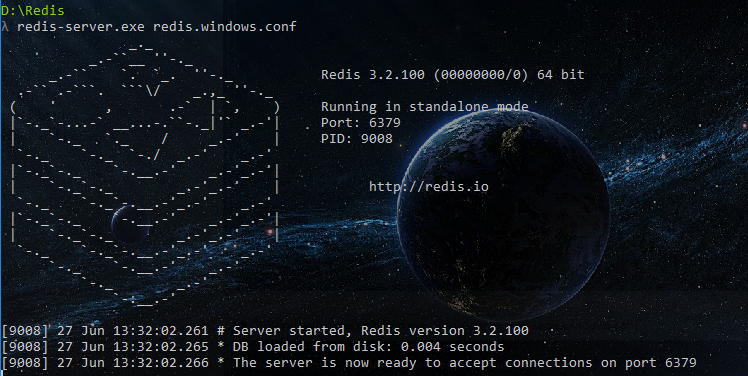


打开一个 cmd 窗口 使用cd命令切换目录到 C:\redis（你自己的redis安装目录） 运行 redis-server.exe redis.windows.conf 。

如果想方便的话，可以把 redis 的路径加到系统的环境变量里，这样就省得再输路径了，后面的那个 redis.windows.conf 可以省略，如果省略，会启用默认的。输入之后，会显示如下界面：



这个起来了，是上次没有关闭导致的



注意：在安装时候出现的错误

1、Redis Creating Server TCP listening socket \*:6379: unable to bind socket

redis-cli.exe

127.0.0.1:6379>shutdown

not connected>exit

然后重新运行redis-server.exe redis.windows.conf，启动成功！

3、Centos版本

Linux下redis安装以及使用已经讲过并且安装过了，我们快速过一下就好。

[root@xuegod63 yum.repos.d]# yum install epel-release

[root@xuegod63 yum.repos.d]# yum install redis -y

[root@xuegod63 ~]# systemctl start redis #启动redis

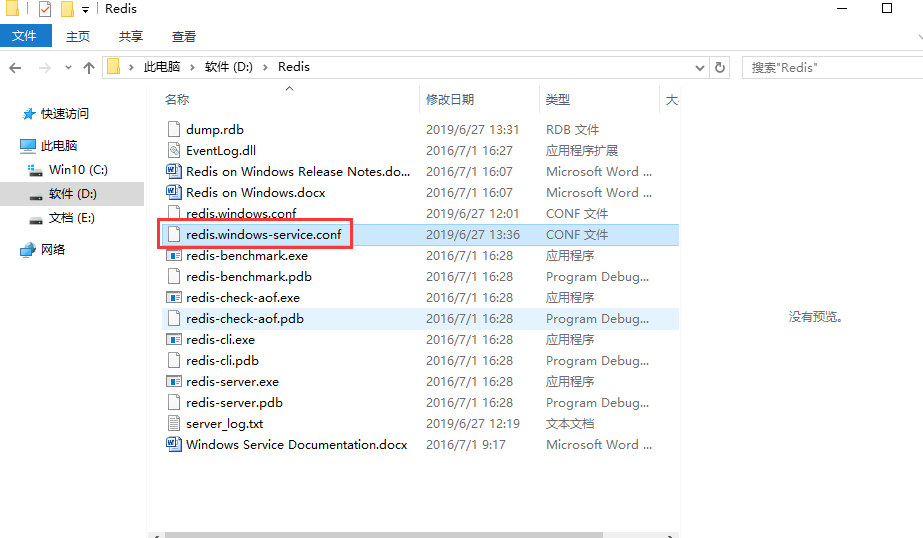
[root@xuegod63 ~]# netstat -antup | grep 6379

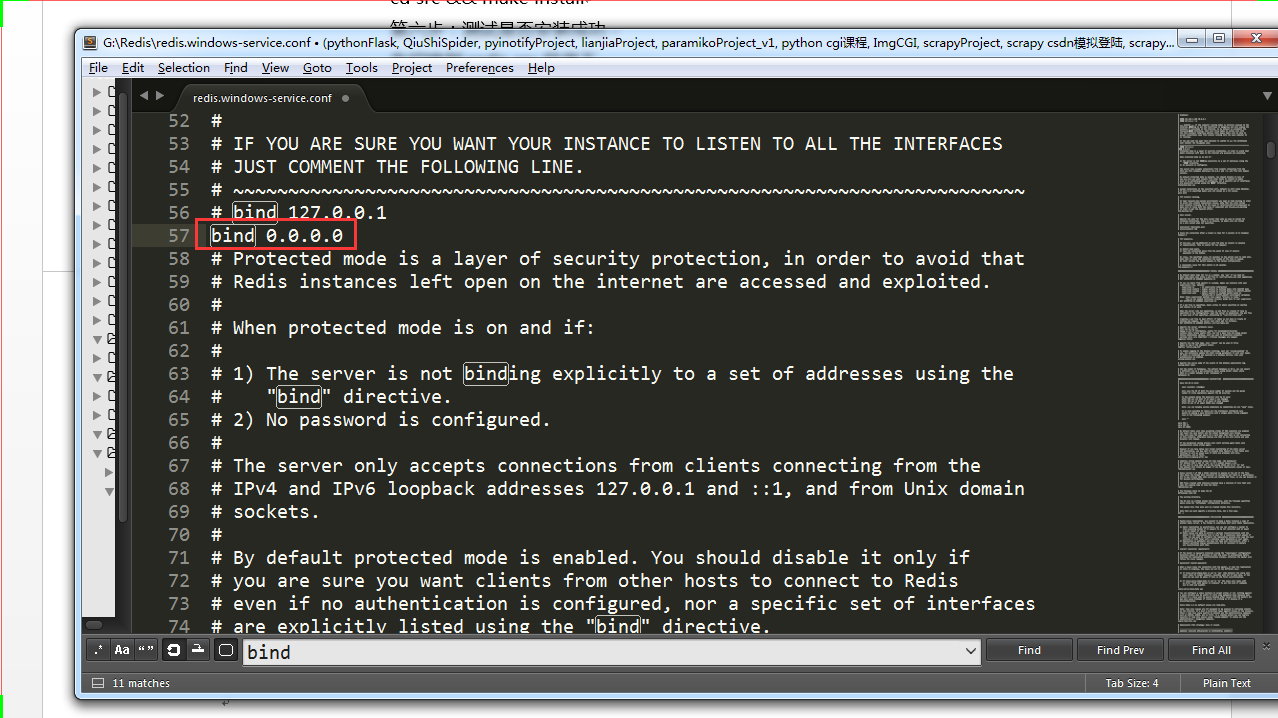
### 6.1.2 Scrapy\_redis安装

安装命令：pip insall scrapy\_redis

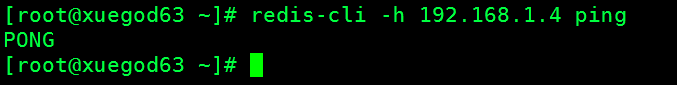
对准备做分布式的服务端的redis数据库进行配置

Windwos为服务端





然后用命令证明ping通没有问题:redis-cli –h 192.168.1.10 ping



## 6.2 使用scrapy-redis进行分布式编程

分布式环境

Window server :192.168.1.4

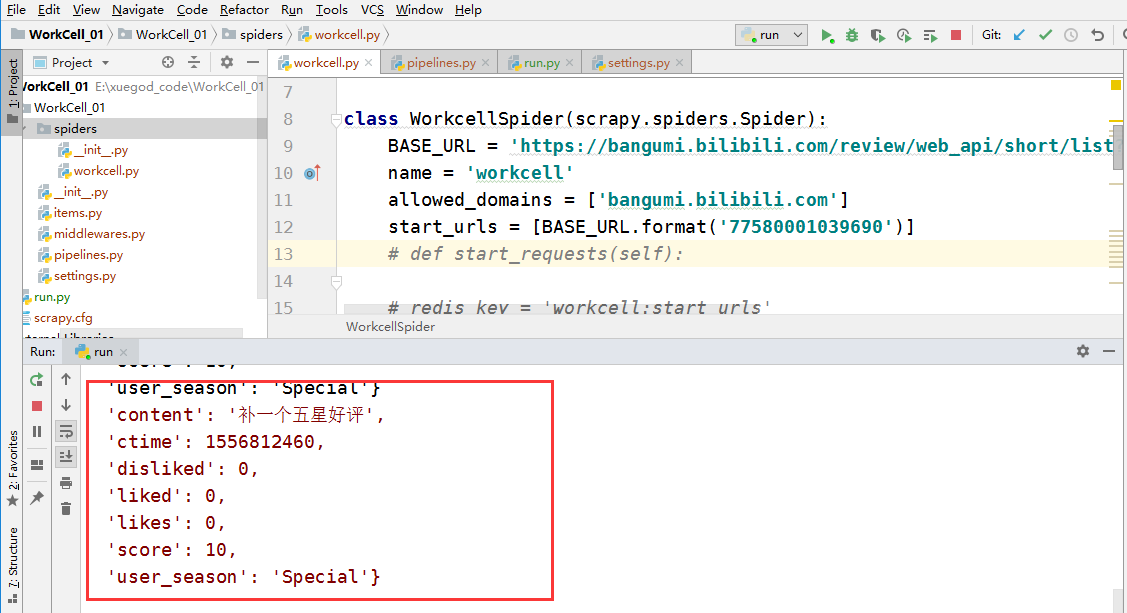
Centos client2 :192.168.1.89

Pycharm

为了让大家有更好的课程节奏，我们采用上面使用的哔哩哔哩视频评论，workcell\_01

1. 修改爬虫的代码

既然我们使用的是redis快速爬取，首先确保自己的代码是可运行的。



### 6.2.1 Settings 配置修改

REDIS\_URL = 'redis://192.168.1.10:6379' #爬虫采用的redis数据库的地址

SCHEDULER = 'scrapy\_redis.scheduler.Scheduler' #采用scrapy\_redis的调度器

DUPEFILTER\_CLASS = 'scrapy\_redis.dupefilter.RFPDupeFilter' #采用scrapy\_redis的去重过滤器

ITEM\_PIPELINES = {

'scrapy\_redis.pipelines.RedisPipeline': 300,

}#采用scrapy\_redis的管道

#允许暂停，redis请求记录不丢失在redis中保持scrapy-redis用到的各个队列，从而允许暂停和暂停后恢复，也就是不清理redis queues

#True 保留

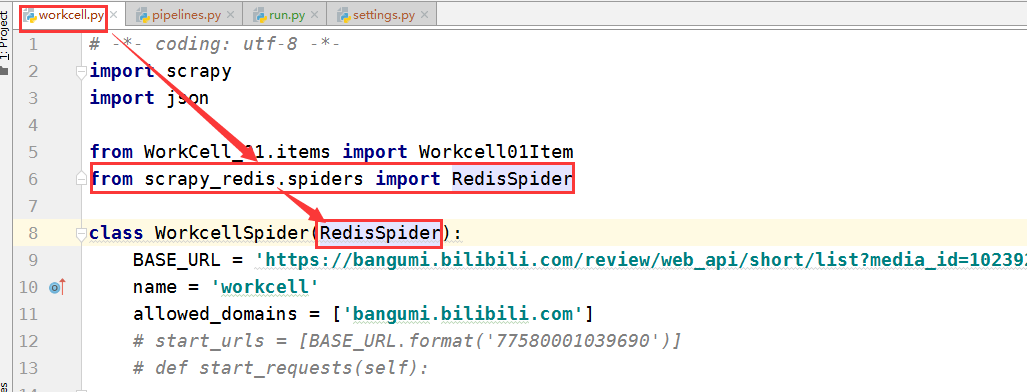
#False 清除

SCHEDULER\_PERSIST = True



### 6.2.2 Spider 修改

1. 先修改爬虫继承的类



1. 修改注释start\_urls



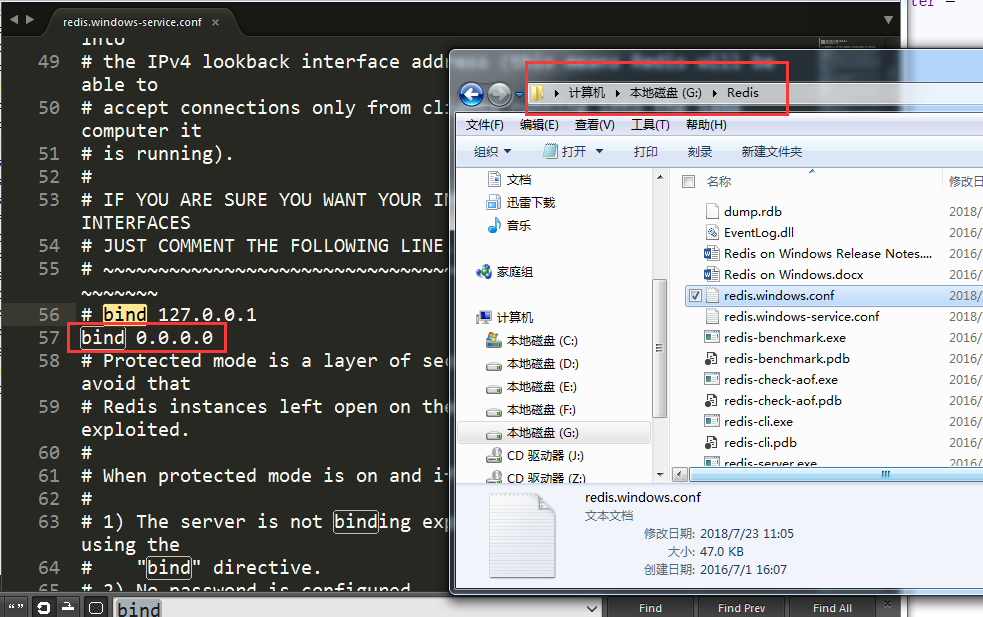
完整代码如下

import scrapy  
import json  
  
from WorkCell\_01.items import Workcell01Item  
from scrapy\_redis.spiders import RedisSpider  
class WorkcellSpider(RedisSpider):  
 BASE\_URL = 'https://bangumi.bilibili.com/review/web\_api/short/list?media\_id=102392&folded=0&page\_size=20&sort=0&cursor={}'  
 name = 'workcell'  
 allowed\_domains = ['bangumi.bilibili.com']  
 *# start\_urls = [BASE\_URL.format('77680001039690')]  
 # def start\_requests(self):  
  
 # redis\_key = 'workcell:start\_urls'* def parse(self, response):  
 print(response.url)  
 *#把json字符串变成python可操作对象* resdata = json.loads(response.body\_as\_unicode())  
 *# print(resdata)* if resdata['code'] == 0:  
 if len(resdata['result']['list']) > 0:  
 data = resdata['result']['list']  
 cursor = data[-1]['cursor']  
 for one in data:  
 item = Workcell01Item()  
 item['author']=one['author']['uname']  
 item['content']=one['content']  
 item['ctime']=one['ctime']  
 item['disliked']=one['disliked']  
 item['liked']=one['liked']  
 item['likes']=one['likes']  
 item['score']=one['user\_rating']['score']  
 item['user\_season']=one['user\_season']['last\_ep\_index'] if 'user\_season' in one else''  
 print(item)  
 yield item  
 yield scrapy.Request(self.BASE\_URL.format(cursor),callback=self.parse)

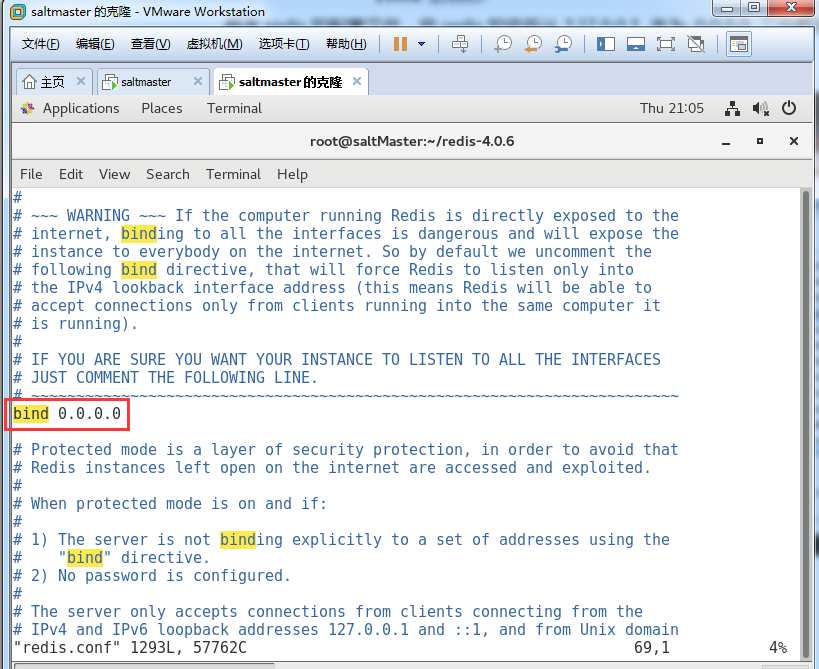
### 6.2.3 修改redis的配置文件

将redis的监听从127.0.0.1 改为 0.0.0.0其他节点都要改

Windows：

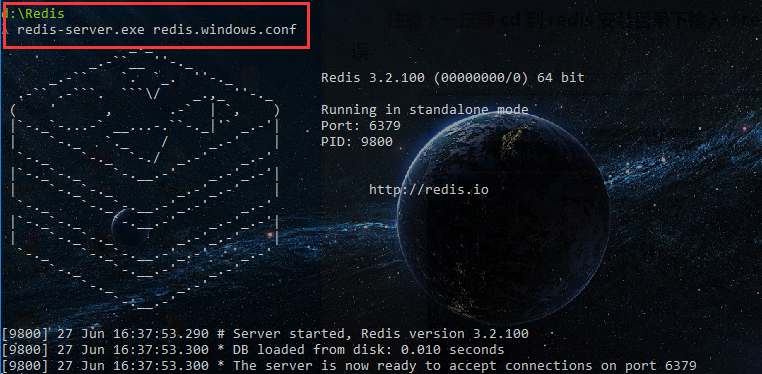


Centos



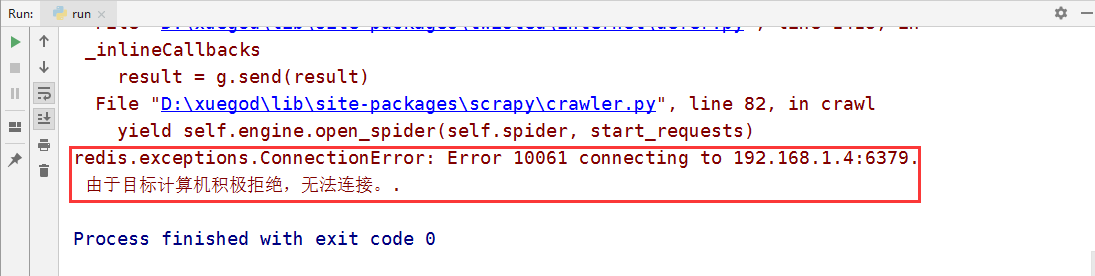
启动windows的redis环境

注意：一定要cd到redis安装目录下输入：redis-server.exe redis.windows.conf 否则在运行项目的时候会出现错误



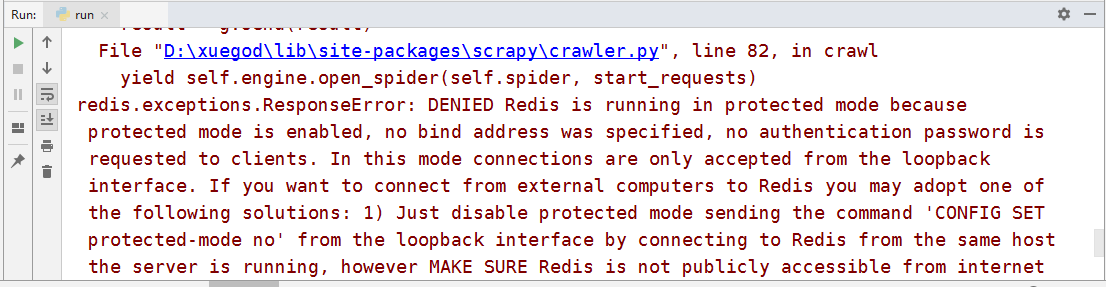
错误提示：

1. 在开启项目前没有开启redis-server 就会出现如下错误

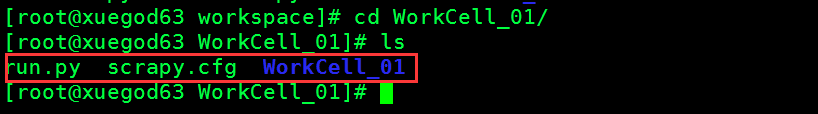


1. 开启了redis-server但是没有cd到redis安装目录。

解决方案：一定要cd到redis安装目录下输入：redis-server.exe redis.windows.conf

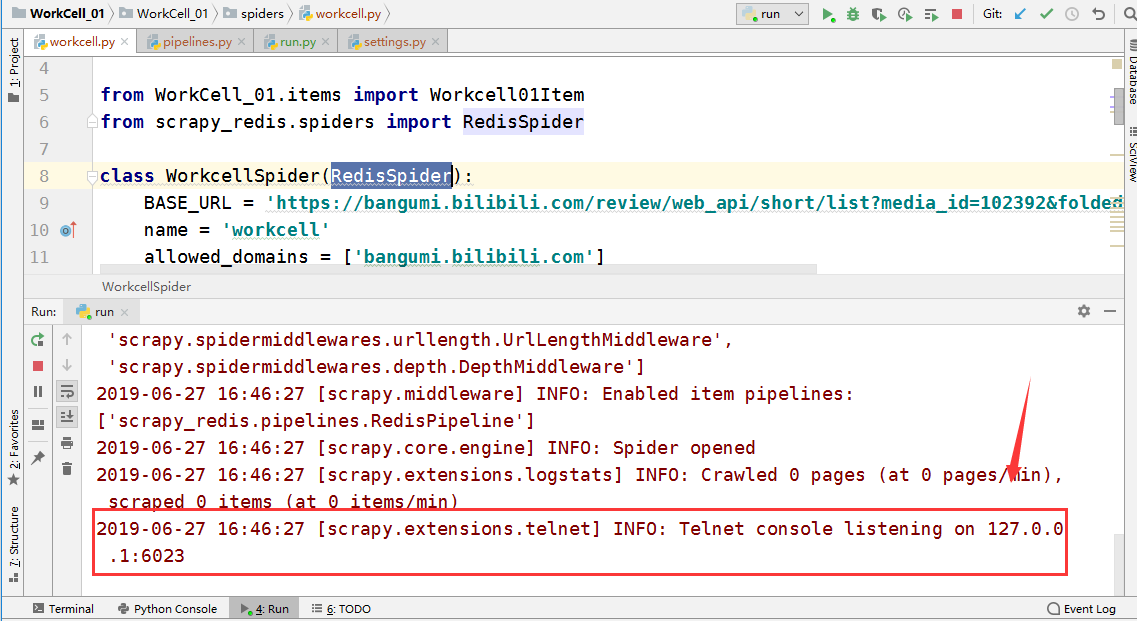


将刚才修改好的爬虫的代码分别上传到两台客户端上。



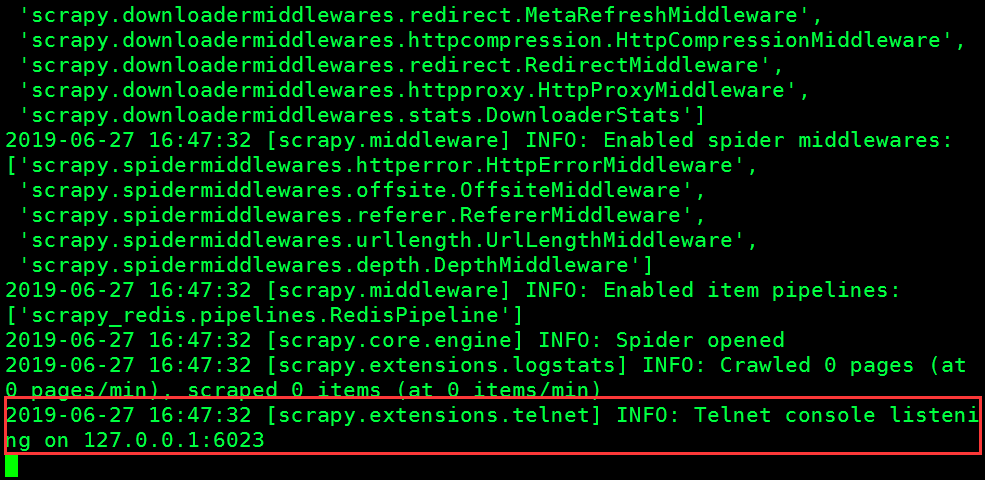
### 6.2.4 启动分布式爬虫

1. 启动服务端的爬虫



1. 启动客户端的爬虫

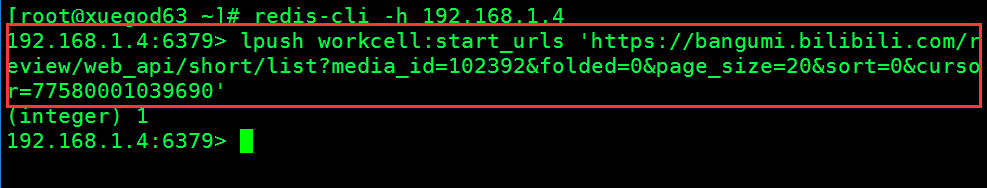
Python run.py



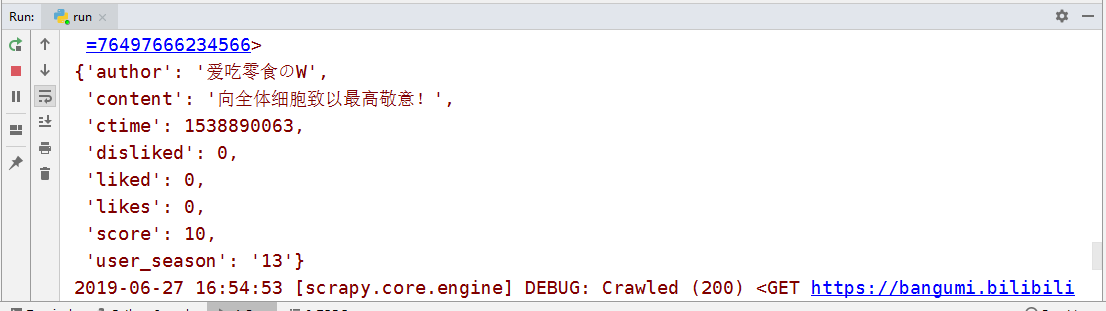
这个时候redis数据库里面没有起始地址，我们在任意端，通过redis数据库添加启动url(start\_urls)。

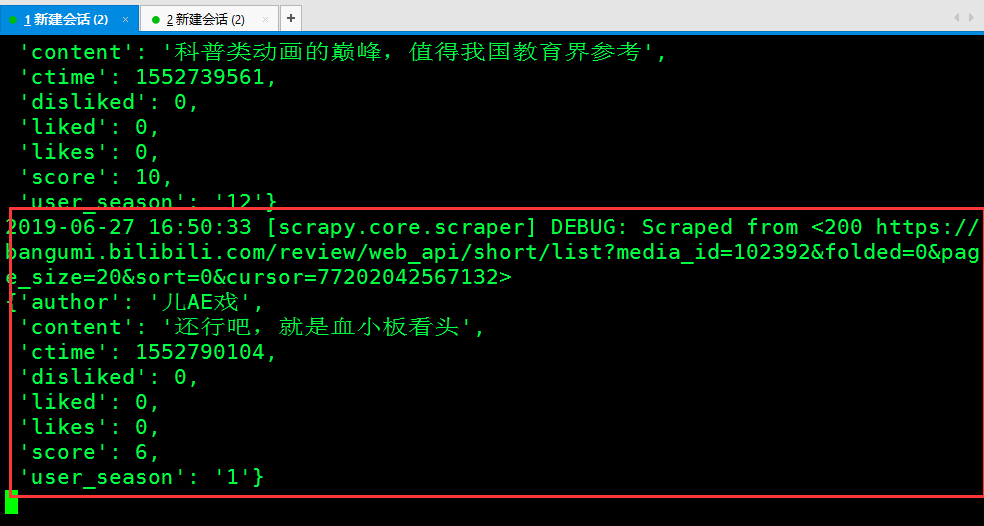
[root@xuegod63 ~]# redis-cli -h 192.168.1.4

192.168.1.4:6379> lpush workcell:start\_urls 'https://bangumi.bilibili.com/review/web\_api/short/list?media\_id=102392&folded=0&page\_size=20&sort=0&cursor=77680001039690'

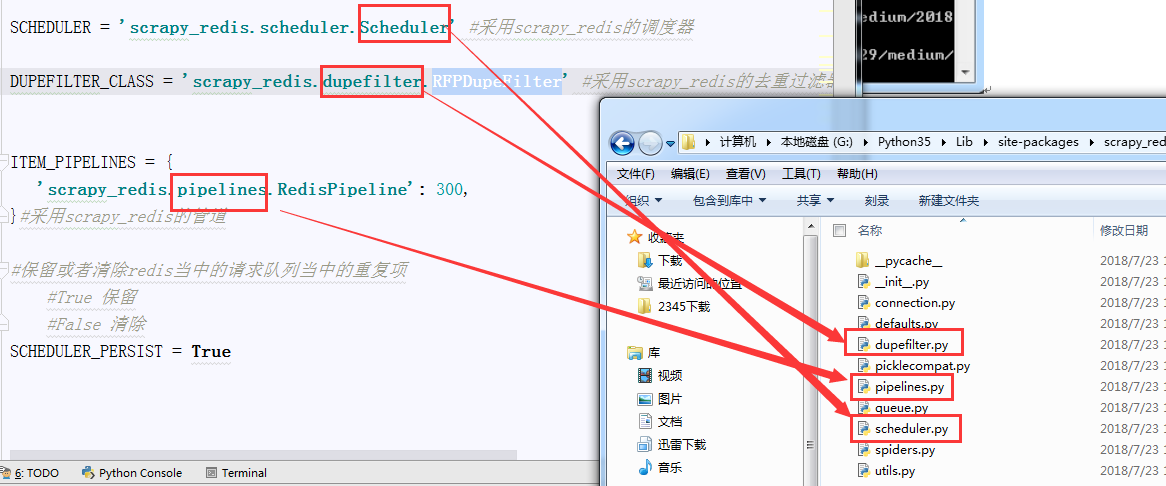


这样就开始爬取了





Scrapy-Redis对应源码



分布式爬虫的目的是将一台服务器的爬取任务，分配给多台服务器同时爬取。

这样编写：

1. 提高爬虫的效率
2. 降低爬虫被拒绝

总结：

6.1 linux安装scrapy

6.2 使用scrapy-redis进行分布式编程