****

分布式系统与云计算

**数据科学与大数据技术**专业

**大数据1802** 班

指导老师\_余腊生\_

姓名 **刘爱芳**

学号 **8208181116**

电话 **17877780889**

**2021** 年**12**月**23**日

**分布式网络爬虫**

**一、实验目的**

本次实验目标为设计一个分布式网络爬虫实现一下功能：

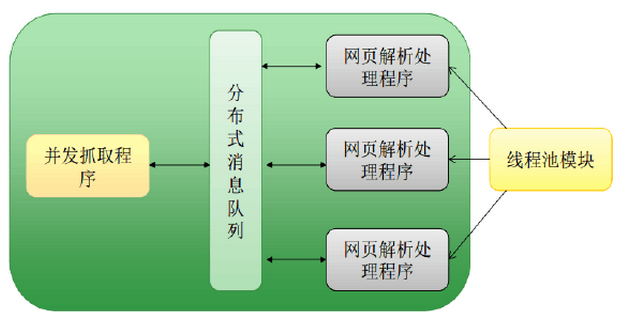
（1） 从一个给定的网址中分析其所包含的URL并爬取对应的网页，直到爬取完全部不重复的网页为止。

（2） 支持分布式爬取，同时记录输出每一个网页的大小。

（3） 采用多线程结构设计，实现高性能的网络爬虫。

**二、实验内容**

本次实验主要的内容就是利用IO复用抓取网页，并多线程的分析每个抓取到的网页所包含的URL信息，通过消息队列将抓取网页的部分和分析网页部分进行通信，最终记录下160000网页中所包含的所有URL，实现分布式网络爬虫。



实验环境为Windows，我们通过celery架构实现分布式的结构，用redis存储celery的broker和爬取的数据。通过python编写爬虫代码，从一个网页爬取该一个网站内的全部URL以及连接的外链和网页的大小，以json格式返回到redis中存储。

**1、下载并安装redis、celery和相关包**

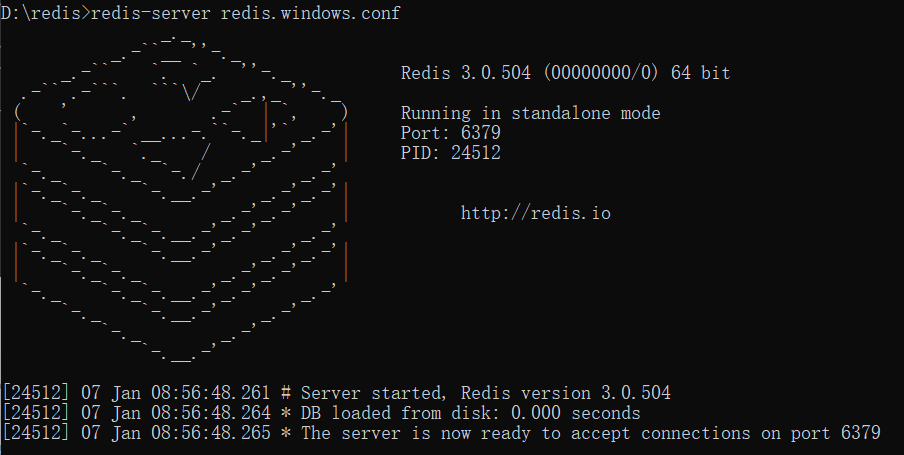
celery会把所有任务都通过消息队列发送给各个分布式节点进行执行，所以可以很好的保证url不会被重复抓取；在检测到worker挂掉的情况下，它会尝试向其他的worker重新发送这个任务信息，这样任务可以继续，不会影响到其他节点；celery自带任务路由，我们可以根据实际情况在不同的节点上运行不同的抓取任务。

由于celery只是任务队列，而不是真正意义上的消息队列，自身不具有存储数据的功能，所以broker和backend需要通过第三方工具来存储信息，我们这里选择Redis作为broker兼backend。

**（1）下载安装redis**

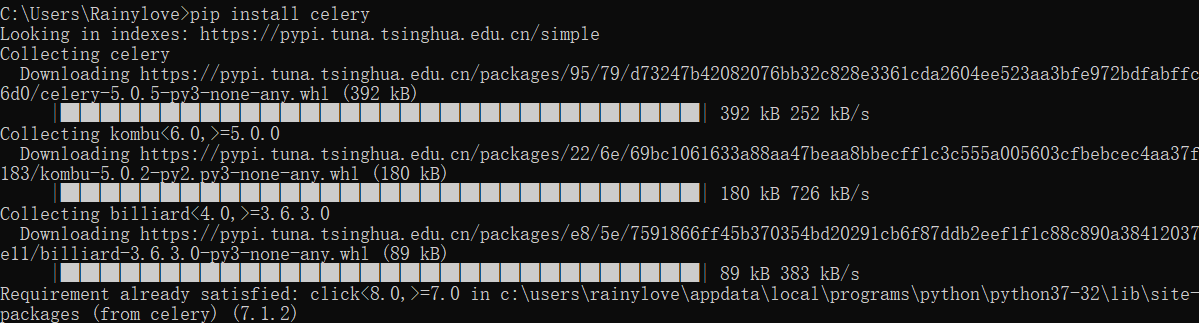
首先pip install redis。

下载Redis-x64-3.0.503.msi；cmd中进入redis目录，使用redis-cli.exe命令安装redis。使用redis-server redis.windows.conf指令打开redis服务，可以看到在6379端口。

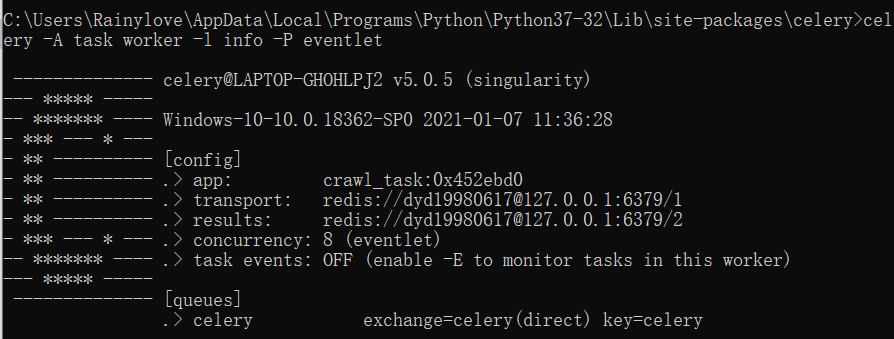


**（2）下载安装celery**

pip install celery。

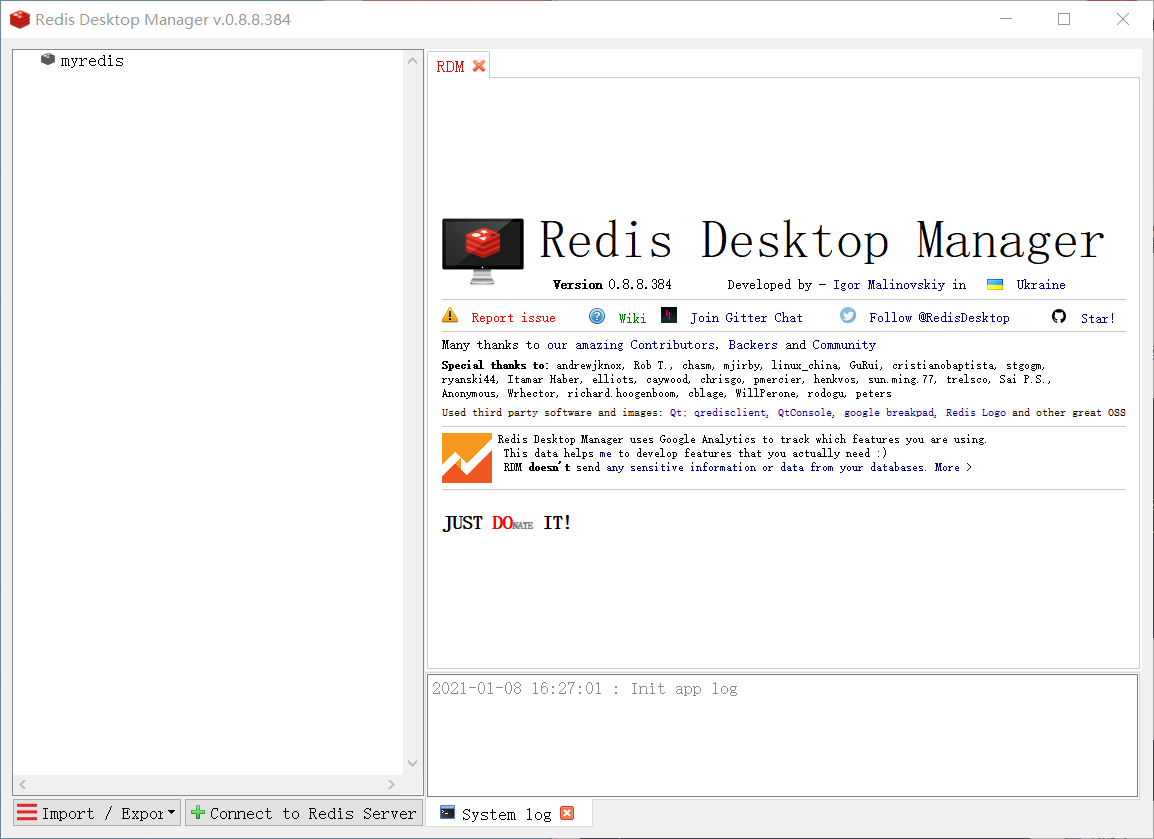


可新建一个task.py任务文件，使用celery -A task worker -l info -P eventlet检验celery是否可以运行。成功则会显示如下图：



在win10环境下运行celery时还需要安装eventlet包，以用于任务（task）执行单元；安装bs4，是python用于爬虫的一个重要库。

**（3）安装RedisDesktopManager**



**2、编写相关代码**

**（1）编写task代码**

代码主要包括getInternalLinks()、getExternalLinks ()、getAllExternalLinks()三个获取网站内链接的函数，分别抓取网站内链接、外链接，以及汇总两种链接。每个函数内主要对出现的链接进行正则化，规范格式，并统计出现的网站链接、适当解析网站内容等。

通过多线程模拟分布式系统操作，再将结果汇总。

**（2）编写workers**

我们需要做Celery实例的初始化工作，include是在初始化celery app的时候需要引入的内容，主要是注册为网络调用的函数所在的文件，基本配置项如：

CELERY\_TASK\_SERIALIZER：任务序列化方式

CELERY\_RESULT\_SERIALIZER：任务执行结果序列化方式

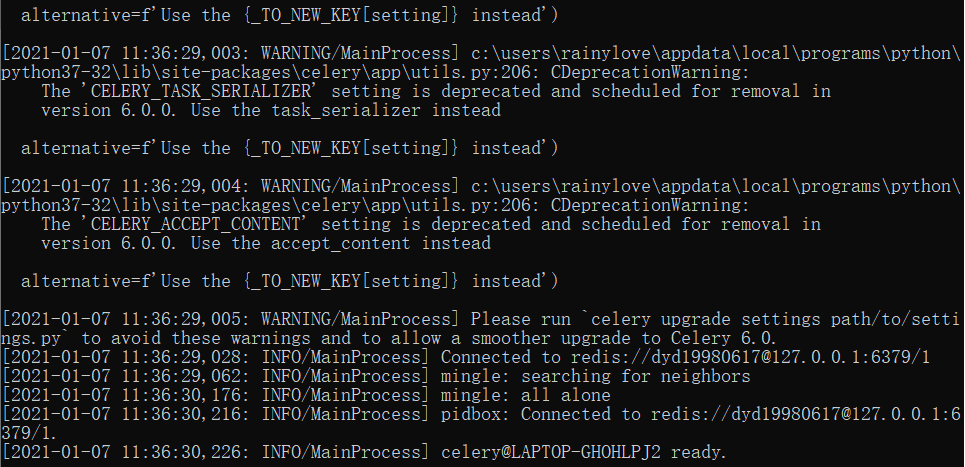
CELERY\_ACCEPT\_CONTENT：指定任务接受的内容序列化类型(序列化)，一个列表；

在这里我们都以json格式存储。

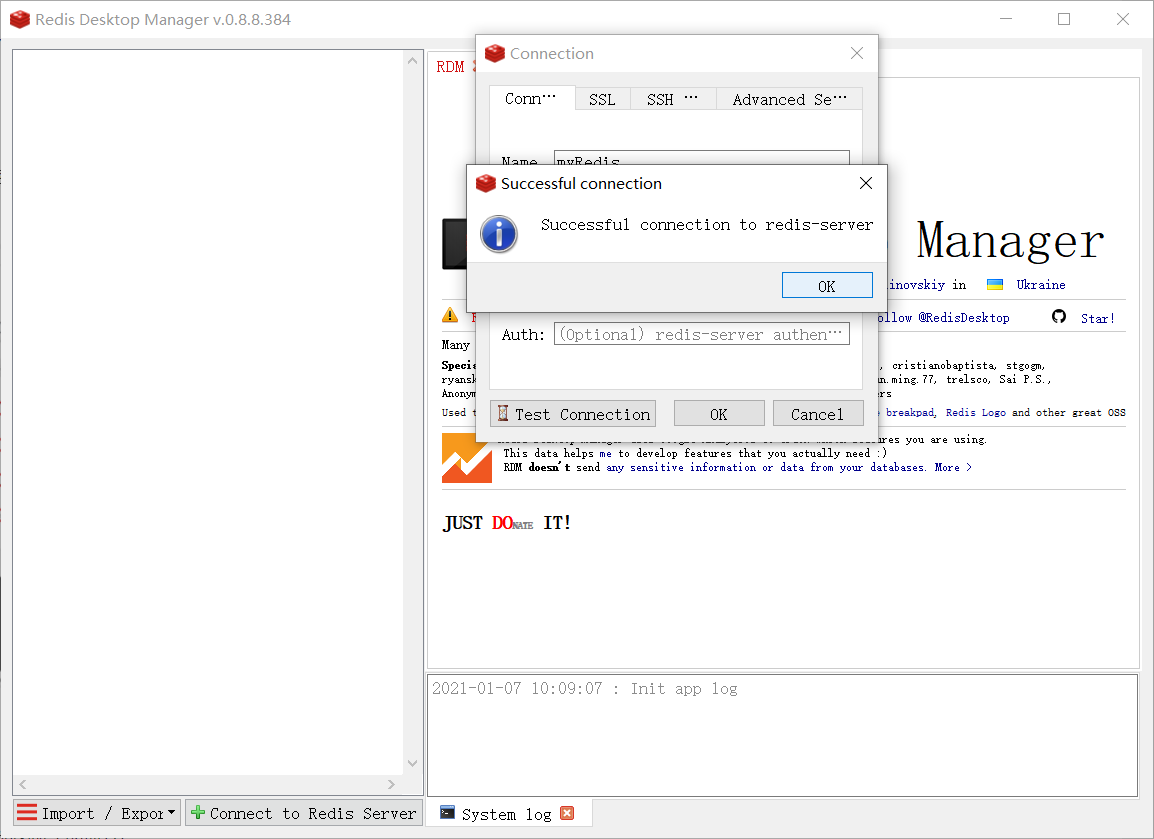
**3、实现爬虫**

**（1）运行相关程序**

参照步骤1开启redis和celery。



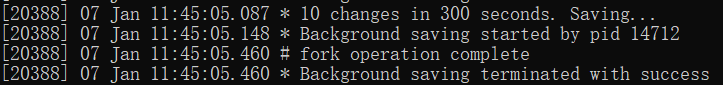
连接Redis Desktop Manager。自主命名Name，Host为127.0.0.1，端口已设定好，为6379；测试成功以后就可以连接。



**（2）运行代码**

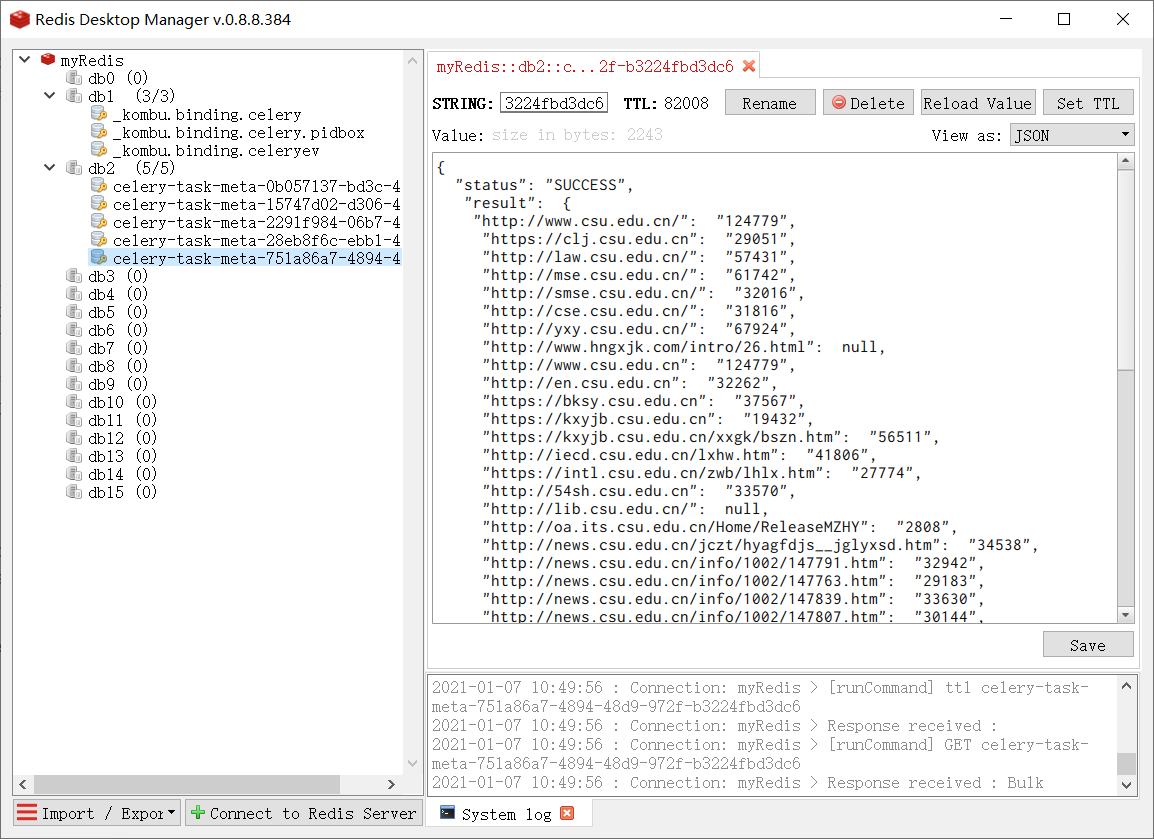
运行crawling.py，该程序通过app.send\_task()来开始执行任务。

运行代码后celery应显示如下字段，表示开始爬取信息：

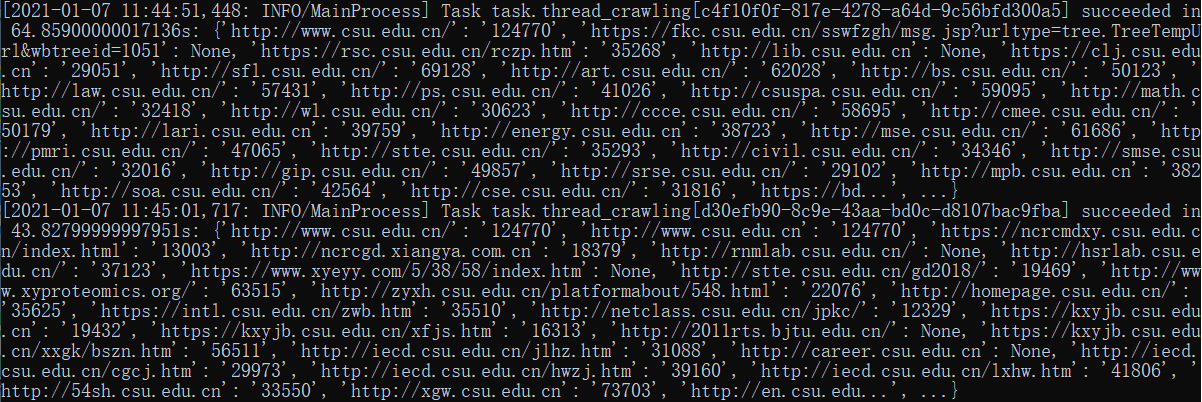


**（3）查看结果**

采集完成后，我们通过Redis Desktop Manager查看结果，如下所示：



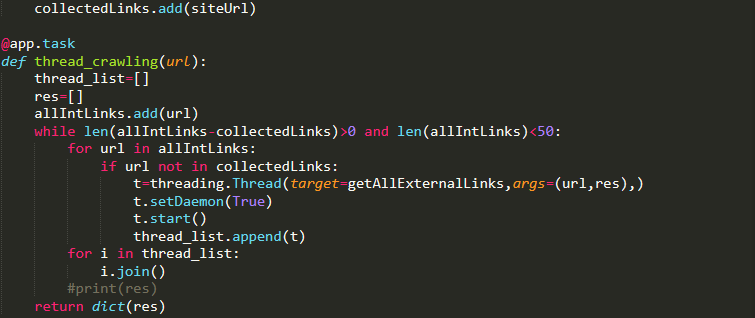
Celery中也会显示爬取到的信息，但是是未经处理的，看起来比较混乱，结果如下：



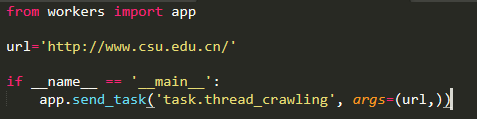
**三、主要代码**

Task：





Crawling：



Workers：

