中南大学

《分布式系统与云计算》

课程实验报告



学生姓名 ： 项岩

学 号 ： 8208180127

实验名称 ： 网络爬虫

指导教师 ： 余腊生

学 院 ： 计算机学院

专业班级 ： 大数据1801

# 网络爬虫

## 一、实验背景

随着国际互联网的迅速发展，网上的信息越来越多，全球网页数量超过20亿，每天新增加730万网页。要在如此浩瀚的信息海洋里寻找信息，就像“大海捞针”一样困难。在实际生活中我们经常会使用像百度、Google这些搜索引擎检索各种信息，搜索引擎正是为了解决这个问题而出现的技术，而网络爬虫正是搜索引擎所需要的关键部分。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。

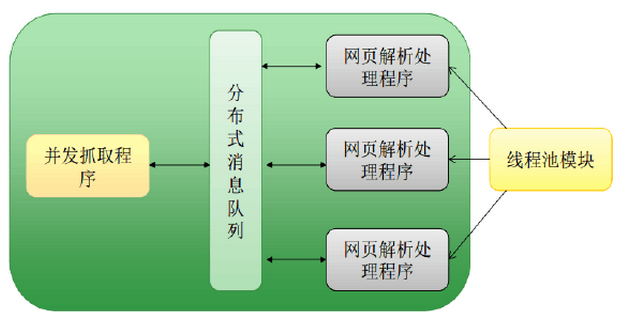
## 二、实验目标

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。



## 三、实验环境

1.独立计算机

2.windows 10操作系统

3.python工具包

4.pycharm

5.celery

6.redis3.2

7.redis Desktop Manager

## 四、实验内容

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，用python编写的爬虫可以从一个网页爬取该一个网站内的URL以及连接的外链还有它们的网页大小，以json格式返回到redis中存储。

### 1. celery架构的搭建（项岩）

#### celery介绍

按celery官网的介绍来说：Celery 是一个简单、灵活且可靠的，处理大量消息的分布式系统，并且提供维护这样一个系统的必需工具。它是一个专注于实时处理的任务队列，同时也支持任务调度。

#### celery的核心知识点

broker：

翻译过来叫做中间人。它是一个消息传输的中间件，可以理解为一个邮箱。每当应用程序调用celery的异步任务的时候，会向broker传递消息，而后celery的worker将会取到消息，执行相应程序。这其实就是消费者和生产者之间的桥梁。

backend:

通常程序发送的消息，发完就完了，可能都不知道对方时候接受了。为此，celery实现了一个backend，用于存储这些消息以及celery执行的一些消息和结果。

worker:

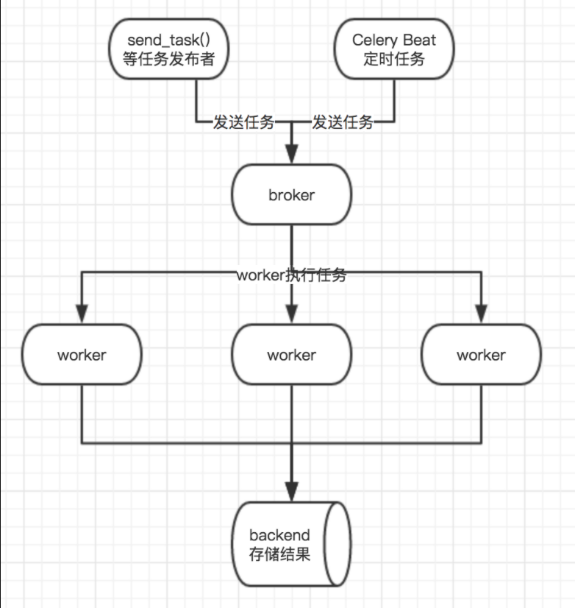
Celery类的实例，作用就是执行各种任务。注意在celery3.1.25后windows是不支持celery worker的！

producer:

发送任务，将其传递给broker

beat: celery实现的定时任务。可以将其理解为一个producer，因为它也是通过网络调用定时将任务发送给worker执行。注意在windows上celery是不支持定时任务的！

#### celery的架构示意图



#### celery架构实现

先安装celery

在项目根目录下，这里我新建一个项目叫做celerystudy，然后切换到该项目目录下，新建文件tasks.py，我们先通过app=Celery()来实例化一个celery对象，在这个过程中，我们指定了它的broker，是redis的db 2,也指定了它的backend,是redis的db3.

定义了一个add函数，重点是@app.task，它的作用在我看来就是将add()注册为一个类似服务的东西，本来只能通过本地调用的函数被它装饰后，就可以通过网络来调用。这个tasks.py中的app就是一个worker。它可以有很多任务，比如这里的任务函数add。

将worker启动起来后，我们就可以通过网络来调用add函数了。我们在后面的分布式爬虫构建中也是采用这种方式分发和消费url的。在命令行先切换到项目根目录，然后打开python交互端。  
 这个时候我们可以在worker的界面看到接收的任务和计算的结果。

这里是异步调用，如果我们需要返回的结果，那么要等rs的ready状态true才行。这里add看不出效果，不过试想一下，如果我们是调用的比较占时间的io任务，那么异步任务就比较有价值了.

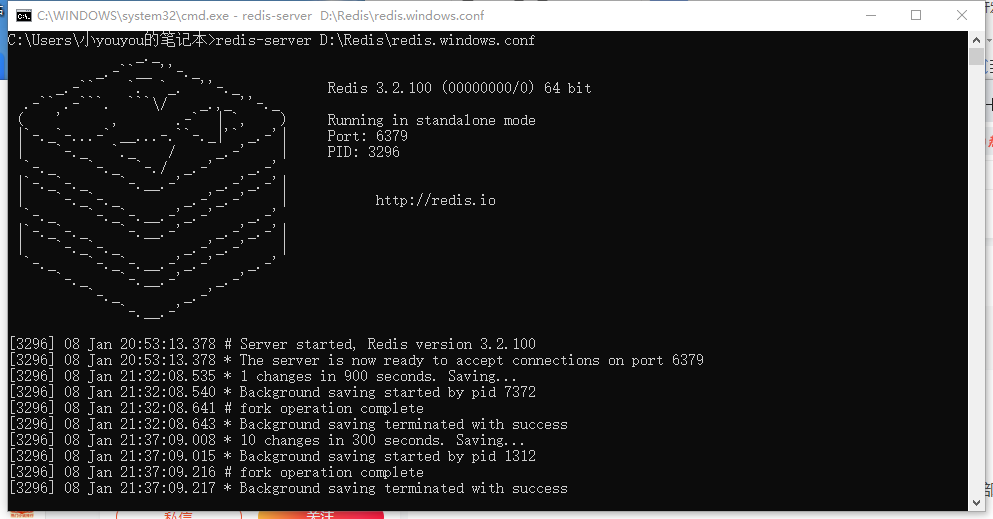
### 2.Redis在windows下的安装（李勤政）

#### 1.下载



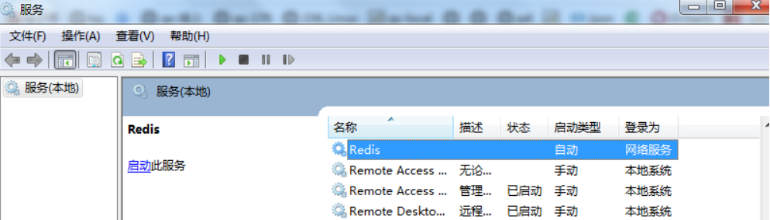
#### 2.解压缩后在文件夹中创建两个文件夹dbcache、logs

#### 3.双击redis-server.exe，启动redis



#### 4.接下来，要将redis注册为系统服务，放在“服务”项中

打开命令行，进入redis目录，执行 redis-server.exe --service-install redis.windows.conf --loglevel verbose，然后查看“服务”。另外，配置一下环境变量，将redis路径添加到path中。



#### 5.redis.windows.conf 文件的配置

这里只配置5项基本参数：

daemonize no   配置redis是否为后台运行

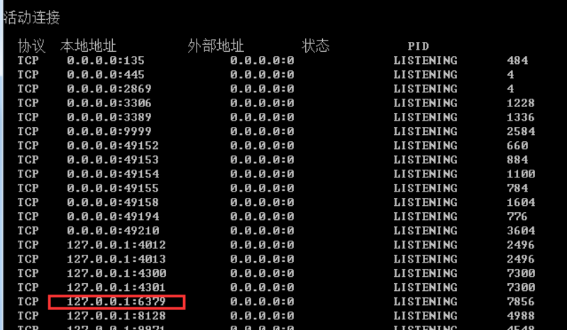
pidfile /var/run/redis\_6379.pid   设置进程保存路径

logfile "./logs/logfile.log"    日志保存目录

databases 16     支持的数据库个数

dir ./dbcache      保存数据文件的目录

#### 6，在服务项中，启动redis



#### 7，通过命令行连接redis，输入 redis-cli

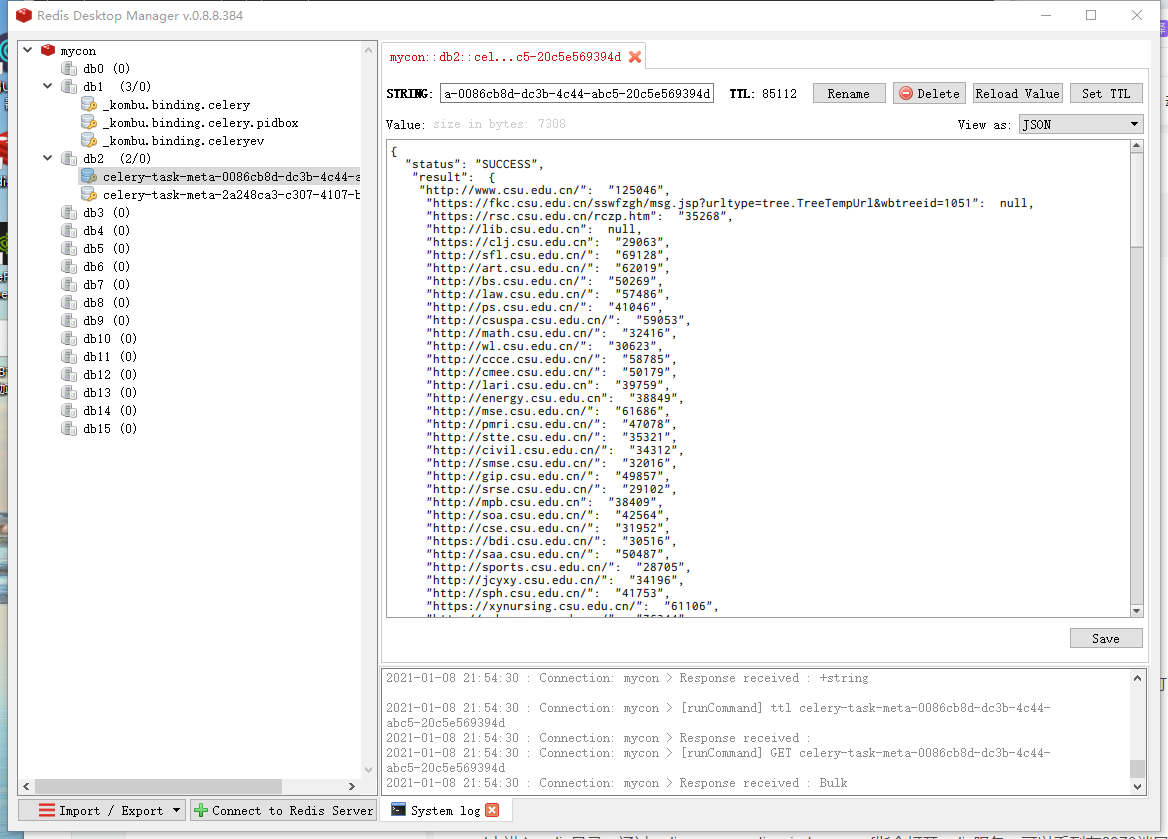
### 3.爬虫的构建

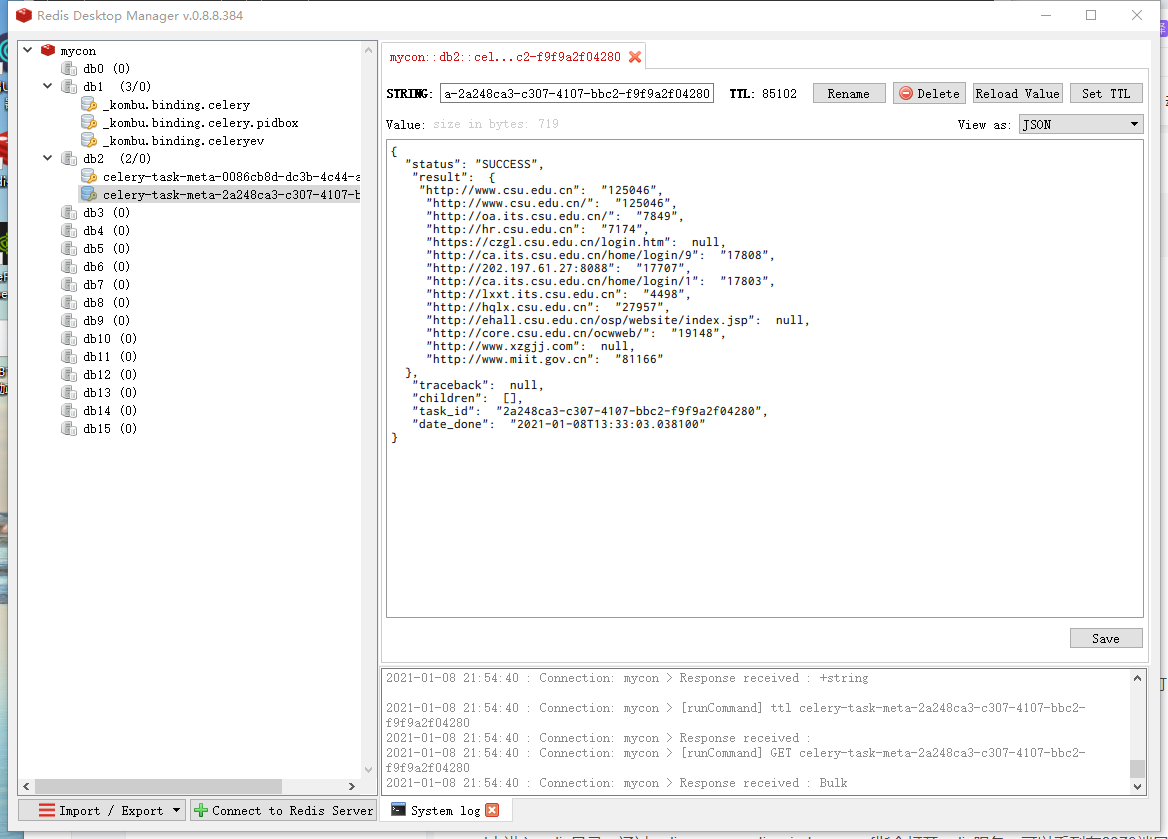
我门选择使用[celery](http://docs.celeryproject.org/en/latest/index.html)作为分布式任务调度工具，是分布式爬虫中任务和资源调度的核心模块。它会把所有任务都通过消息队列发送给各个分布式节点进行执行，所以可以很好的保证url不会被重复抓取；它在检测到worker挂掉的情况下，会尝试向其他的worker重新发送这个任务信息，这样第二个问题也可以得到解决；celery自带任务路由，我们可以根据实际情况在不同的节点上运行不同的抓取任务。

## 五、代码清单

由于代码清单过长，以附件的形式放在文件夹中。

## 六、实验结果展示





## 七、参考文献

《Redis在windows下的安装、启动、配置》——[scorpio\_meng](https://blog.csdn.net/scorpio_meng)

《如何构建一个分布式爬虫》——[resolvewang](https://www.jianshu.com/u/2285873d6fc4)

《分布式爬虫》——[近西](https://my.csdn.net/yuandong_d)

《实现简单分布式爬虫》——[近西](https://my.csdn.net/yuandong_d)