温良Miner
温一壶月光下酒

使用scrapy-redis搭建分布式爬虫环境

scrapy-redis简介

scrapy-redis是scrapy框架基于redis数据库的组件，用于scrapy项目的分布式开发和部署。

有如下特征：

□ 分布式爬取

您可以启动多个spider工程，相互之间共享单个redis的requests队列。最适合广泛的多个域名网站的内容爬取。

□ 分布式数据处理

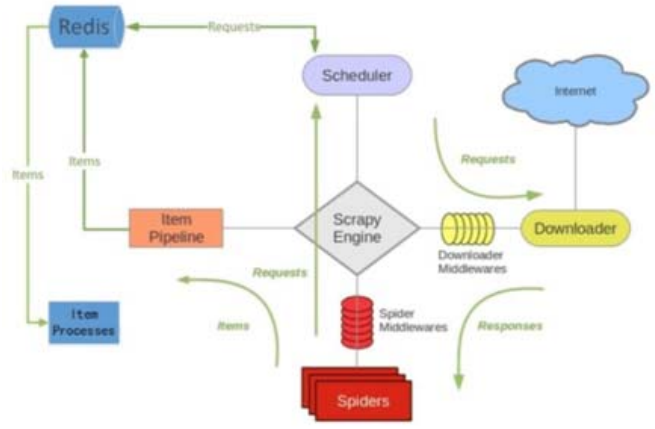
爬取到的scrapy的item数据可以推入到redis队列中，这意味着你可以根据需求启动尽可能多的处理程序来共享item的队列，进行item数据持久化处理

□ Scrapy即插即用组件

Scheduler调度器 + Duplication复制 过滤器，Item Pipeline，基本spider

scrapy-redis架构

scrapy-redis整体运行流程如下：



1. 首先Slaver端从Master端拿任务（Request、url）进行数据抓取，Slaver抓取数据的同时，产生新任务Request便提交给 Master 处理；

2. Master端只有一个Redis数据库，负责将未处理的Request去重和任务分配，将处理后的Request加入待列，并且存储爬取的数据。

Scrapy-Redis默认使用的就是这种策略，我们实现起来很简单，因为任务调度等工作Scrapy-Redis都已经好了，我们只需要继承RedisSpider、指定redis_key就行了。

缺点是，Scrapy-Redis调度的任务是Request对象，里面信息量比较大（不仅包含url，还有callback函数、headers等信息），

可能导致的结果就是会降低爬虫速度、而且会占用Redis大量的存储空间，所以如果要保证效率，那么就需件水平。

scrapy-redis安装

通过pip安装即可：pip install scrapy-redis

昵称：温良Miner
园龄：11个月
粉丝：7
关注：0
+加关注

< 2019年3月 >

日	一	二	三	四	五	六
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

搜索
找找看
谷歌搜索

常用链接
我的随笔
我的评论
我的参与
最新评论
我的标签

最新随笔
1. scrapy抓取豆瓣电影相关数据
2. scrapy结合selenium抓取武汉市环保局空气质量日报
3. scrapy抓取国家社科基金项目数据库
4. pyspark报错Exception: Java gatew ay process exited before sending its p ort number解决方法

: Yo
l. Tr
:kag

urllib(2)
redis(2)
requests(2)
更多

一般需要python、redis、scrapy这三个安装包

官方文档：https://scrapy-redis.readthedocs.io/en/stable/

源码位置：https://github.com/rmax/scrapy-redis

参考博客：https://www.cnblogs.com/kylinlin/p/5198233.html

scrapy-redis常用配置

一般在配置文件中添加如下几个常用配置选项：

1(必须). 使用了scrapy_redis的去重组件，在redis数据库里做去重

```
DUPEFILTER_CLASS = "scrapy_redis.dupefilter.RFPDupeFilter"
```

2(必须). 使用了scrapy_redis的调度器，在redis里分配请求

```
SCHEDULER = "scrapy_redis.scheduler.Scheduler"
```

3(可选). 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

```
SCHEDULER_PERSIST = True
```

4(必须). 通过配置RedisPipeline将item写入key为 spider.name : items 的redis的list中，供后面的分布式处理 item 这个已经由 scrapy-redis 实现，不需要我们写代码，直接使用即可

```
ITEM_PIPELINES = {
    'scrapy_redis.pipelines.RedisPipeline': 100 ,
}
```

5(必须). 指定redis数据库的连接参数

```
REDIS_HOST = '127.0.0.1'
REDIS_PORT = 6379
```

scrapy-redis键名介绍

scrapy-redis中都是用key-value形式存储数据，其中有几个常见的key-value形式：

1、 “项目名:items” -->list 类型，保存爬虫获取到的数据item 内容是 json 字符串

2、 “项目名:dupefilter” -->set类型，用于爬虫访问的URL去重 内容是 40个字符的 url 的hash字符串

3、 “项目名: start_urls” -->List 类型，用于获取spider启动时爬取的第一个url

4、 “项目名:requests” -->zset类型，用于scheduler调度处理 requests 内容是 request 对象的序列化 字符串

scrapy-redis简单实例

在原来非分布式爬虫的基础上，使用scrapy-redis简单搭建一个分布式爬虫，过程只需要修改一下spider的和配置文件即可，很简单。

原非分布式爬虫项目，参见：https://www.cnblogs.com/pythoner6833/p/9018782.html

首先修改配置文件，在settings.py文件中添加代码：

```
DUPEFILTER_CLASS = "scrapy_redis.dupefilter.RFPDupeFilter"
SCHEDULER = "scrapy_redis.scheduler.Scheduler"
SCHEDULER_PERSIST = True
ITEM_PIPELINES = {
    'scrapy_redis.pipelines.RedisPipeline': 100,
}
REDIS_HOST = '127.0.0.1'
REDIS_PORT = 6379
```

然后需要修改的文件，是spider文件，原文件代码为：

随笔分类

并发编程(6)
后端开发(4)
机器学习(1)
网络爬虫(22)

随笔档案

2019年1月 (4)
2018年12月 (1)
2018年9月 (2)
2018年7月 (5)
2018年6月 (3)
2018年5月 (33)
2018年4月 (2)

积分与排名

积分 - 17455
排名 - 31057

最新评论

1. Re:scrapy下载中间件结合selenium抓取全国空气质量检测数据
你这个代码只能爬到第一个城市第一个月第一天的数据呀
--ChristyL

2. Re:scrapy项目部署
写得挺好的，你的小点蓝色背景用的什么？
--JakeLong

阅读排行榜

1. 创建指定python版本的虚拟环境(2480)
2. 使用scrapy-redis搭建分布式爬虫环境(2099)
3. 经典算法之K近邻（回归部分）(1621)
4. Scrapy中的反反爬、logging设置、Request参数及POST请求(1316)
5. scrapy的一个简单小项目(1223)

评论排行榜

1. scrapy项目部署(1)
2. scrapy下载中间件结合selenium抓取全国空气质量检测数据(1)

```python
import scrapy
# from scrapy_redis.spiders import RedisSpider
# 导入待爬取字段名
from tencent.items import TencentItem, DetailsItem

class TencentWantedSpider(scrapy.Spider):
    name = 'tencent_wanted'
    # allowed_domains = ['hr.tencent.com']

    start_urls = ['https://hr.tencent.com/position.php']
    # redis_key = 'tencent:start_urls'
    base_url = 'https://hr.tencent.com/'

    def parse(self, response):
```

修改为：

```python
# -*- coding: utf-8 -*-
import scrapy
from scrapy_redis.spiders import RedisSpider
# 导入待爬取字段名
from tencent.items import TencentItem, DetailsItem

class TencentWantedSpider(RedisSpider):
    name = 'tencent_wanted'
    # allowed_domains = ['hr.tencent.com']

    # start_urls = ['https://hr.tencent.com/position.php']
    redis_key = 'tencent:start_urls'

    base_url = 'https://hr.tencent.com/'

    def parse(self, response):
```

只修改了两个地方，一个是继承类：由scrapy.Spider修改为RedisSpider

然后start_url已经不需要了，修改为：redis_key = "xxxxx"，其中，这个键的值暂时是自己取的名字，

一般用项目名：start_urls来代替初始爬取的url。由于分布式scrapy-redis中每个请求都是从redis中取出来的，因此，在redis数据库中，设置一个redis_key的值，作为初始的url，scrapy就会自动在redis中取出redis_key的值，作为初始url，实现自动爬取。

因此：来到redis中，添加代码：

```
127.0.0.1:6379> lpush tencent2:start_urls https://hr.tencent.com/position.php?
(integer) 1
```

即：在redis中设置一个键值对，键为tencent2:start_urls , 值为：初始化url。即可将传入的url作为初始爬url。

如此一来，分布式已经搭建完毕。

分类：网络爬虫

标签：scrapy-redis，分布式

温良Miner
关注 - 0
粉丝 - 7
+加关注

0 0

« 上一篇：python与redis交互及redis基本使用
» 下一篇：scrapy自动抓取蛋壳公寓最新房源信息并存入sql数据库

★本文目录