Scrapy框架的研究与应用

SY1906514 汪凌风 SY1906508 明　昊

SY1906513 汪丽萍 SY1906431 郑泽西

SY1806220 闫奕涛 SY1906509 邵志钧

2020/03/17

目　　录

[Scrapy框架的研究与应用 1](#_Toc35378177)

[一、引言 3](#_Toc35378178)

[1.1 项目背景 3](#_Toc35378179)

[1.2 术语解释 4](#_Toc35378180)

[二、项目概述 5](#_Toc35378181)

[2.1项目目标 5](#_Toc35378182)

[2.2产品目标与应交付成果 5](#_Toc35378183)

[2.3项目约定 5](#_Toc35378184)

[2.4项目验收方式与依据 6](#_Toc35378185)

[2.5项目进行日期 6](#_Toc35378186)

[2.6项目遵循标准 6](#_Toc35378187)

[三、项目方案介绍 7](#_Toc35378188)

[3.1项目目标 7](#_Toc35378189)

[3.2功能说明 7](#_Toc35378190)

[3.3 项目开发环境 8](#_Toc35378191)

[3.4 项目技术路线 8](#_Toc35378192)

[四、项目组织团队 9](#_Toc35378193)

[4.1成员组成 9](#_Toc35378194)

[4.2组织结构 9](#_Toc35378195)

[4.3人员分工 9](#_Toc35378196)

[4.4协作与沟通 11](#_Toc35378197)

[4.5人员贡献率计算方案 12](#_Toc35378198)

[五、实施计划 13](#_Toc35378199)

[5.1方法与工具 13](#_Toc35378200)

[5.2总体进度安排 13](#_Toc35378201)

# 一、引言

## 项目背景

　　随着大数据时代的来临，网络爬虫在互联网中的地位将越来越重要。互联网中的数据是海量的，如何自动高效地获取互联网中我们感兴趣的信息并为我们所用是一个重要的问题，而爬虫技术就是为了解决这些问题而生的。

　　Scrapy 基于python，是一种快速的高级 web crawling 和 web scraping 框架，用于对网站进行爬网并从其页面提取结构化数据。它可以用于广泛的用途，从数据挖掘到监控和自动化测试。

根据官方文档，Scrapy具有以下优势：

* 内置支持 selecting and extracting 使用扩展的CSS选择器和XPath表达式从HTML/XML源中获取数据，并使用正则表达式提取助手方法。
* interactive shell console （ipython-aware）用于尝试使用css和xpath表达式来获取数据，在编写或调试spider时非常有用。
* 内置支持 generating feed exports 以多种格式（json、csv、xml）存储在多个后端（ftp、s3、本地文件系统）。
* 强大的编码支持和自动检测，用于处理外部、非标准和中断的编码声明。
* 允许使用 signals 以及定义良好的API（中间件， extensions 和 pipelines ）。
* 广泛的内置扩展和用于处理的中间产品。
* A Telnet console 用于挂接到运行在Scrapy进程中的Python控制台，以便内省和调试爬虫程序。

## 术语解释

|  |  |
| --- | --- |
| **Engine** | 引擎，处理整个系统的数据流处理、触发事务，是整个框架的核心。 |
| **Item** | 项目，它定义了爬取结果的数据结构，爬取的数据会被赋值成该Item对象。 |
| **Selectors** | Scrapy有自己的数据提取机制。它们被称为选择器，因为它们“选择”HTML文档的某些部分 XPath 或 CSS 表达。 |
| **Scheduler** | 调度器，接受引擎发过来的请求并将其加入队列中，在引擎再次请求的时候将请求提供给引擎。 |
| **Downloader** | 下载器，下载网页内容，并将网页内容返回给“蜘蛛”。 |
| **Spider** | “蜘蛛”，其内定义了爬取的逻辑和网页的解析规则，它主要负责解析响应并生成提取结果和新的请求。spider是定义一个特定站点（或一组站点）如何被抓取的类，包括如何执行抓取（即跟踪链接）以及如何从页面中提取结构化数据（即抓取项）。换言之，spider是为特定站点（或者在某些情况下，一组站点）定义爬行和解析页面的自定义行为的地方。 |
| **Item Pipeline** | 项目管道，负责处理由蜘蛛从网页中抽取的项目，它的主要任务是清洗、验证和存储数据。 |
| **Downloader Middlewares** | 下载器中间件，位于引擎和下载器之间的钩子框架，主要处理引擎与下载器之间的请求及响应，自定义扩展下载功能的组件。 |
| **Spider Middlewares** | 蜘蛛中间件，自定义扩展Engine和Spider中间通信的功能组件。在该机制中，可以插入自定义功能来处理发送到的响应。 |

表1 术语解释目录

　　参考：Scrapy中文文档：https://www.osgeo.cn/scrapy/

# 二、项目概述

## 2.1 项目目标

从scrapy框架出发，根据对源代码与开源文档的分析，逆向编写软件需求规格说明书。并且根据软件需求规格说明书，对所选框架进行改进。对软件进行测试，完成测试需求规格说明书和软件测试报告。完成规定的八项实验并提交综合实验总结报告。

## 2.2 产品目标与应交付成果

项目的产品目标与应交付成果如表2所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 交付时间 | 要求 | 备注 |
| No.1 | 项目计划书 | 2020/3/13 | 可行、明确 | 暂无 |
| No.2 | 项目进度表 | 2020/3/13 | 可行、详细 | 暂无 |
| No.3 | 项目工作日志 | 待定 | 暂无 | 暂无 |
| No.4 | 软件需求规格说明书 | 待定 | 明确、详细 | 暂无 |
| No.5 | 需求检查单 | 待定 | 详细 | 暂无 |
| No.6 | 测试需求规格说明书 | 待定 | 完整、详细 | 暂无 |
| No.7 | 软件测试分析报告 | 待定 | 详尽 | 暂无 |
| No.8 | 软件测试评审报告 | 待定 | 充分 | 暂无 |
| No.9 | 项目工作量统计报告 | 待定 | 详细、充分 | 暂无 |
| No.10 | 项目方法总结报告 | 待定 | 完整、详细 | 暂无 |
| No.11 | 项目个人总结 | 待定 | 充分、详实 | 暂无 |

表2 应交付成果列表

## 2.3 项目约定

1. 在无不可抗力影响的情况下，小组成员都应按照项目进度表按时完成分配的任务。
2. 项目分工均充分考虑了各位小组成员的个人能力，在本人同意的前提下予以分配。
3. 小组每周至少召开两次讨论会，用以汇报本周的完成情况以及分布下周的工作。情况特殊一周会召开多次会议。
4. 小组成员在遇到自己无法解决的问题时会及时在交流群抛出以加快解决。
5. 小组成功的工作成果原则上应先提交至个人分支的备用库内，经小组组长审核通过之后方可合并至主分支。

## 2.4 项目验收方式与依据

* 验收方式

项目验收方式采取课堂评审与交付验收

* 验收依据

依旧8个实验的实验要求组件互评

## 2.5 项目进行日期

原则上本项目由2019年春季学期第二周开始，于2019年春季学期第十七周截至。视突发情况与不可抗力灵活调整。

## 2.6 项目遵循标准

本项目遵从以下标准：

GB/T 13702-1992 计算机软件分类与代码

GB/T 19003-2008 软件工程

GB/T 9386-2008 计算机测试文档编制

GB/T 9385-2008 计算机软件需求规格说明

GB/T 5532-2008 计算机软件测试规范

GB/T 8567-2006 计算机软件文档编制规范

# 三、项目方案介绍

## 3.1 项目目标

从Scrapy框架出发，根据对源代码与开源文档的分析，逆向编写软件需求规格说明书。并且根据软件需求规格说明书，对所选框架进行改进。改进内容包括框架的易用性和性能等。对改进后的软件进行测试，完成测试需求规格说明书和软件测试报告。完成规定的八项实验并提交综合实验总结报告。

## 3.2 功能说明

　　首先从框架本身，Scrapy是一款能够实现网页爬虫的框架，其本身有良好的并发性能，能胜任数据量较大的爬虫任务，在不存在反爬的情况下可以轻松达到上千的QPS，同时可以实现多层级、较为复杂的爬虫逻辑；对于所有爬虫任务使用统一的爬虫框架，每个任务创建相应的spider，规范代码结构，并且提供开发维护效率；自带了shell调试工具，可以方便的进行网页内容捕获的测试；Scrapy自带了较好的状态日志系统，可以方便的针对爬虫任务的爬取量、实时速率、失败请求进行分析，统计等。其优势也在项目概述中有所介绍。

当然Scrapy这个框架也并非完美，其本身是有一些缺陷的，如：与用户交互的方式依然是古老的命令行交互，没有自带的可视化用户操作界面，其性能上也有可以改进的地方，如在爬取大量信息时，容易出现内存泄漏的问题。本项目的主要工作在于对Scrapy进行功能扩展和性能改进，针对刚才所述的这些缺点，目前计划为Scrapy增加如下新的功能点：

* 为Scrapy定制可视化web用户界面。
* 在web界面中输入用户需要筛选的标签、目标网站等信息，为用户输出爬虫的内容，时长等，并对输出进行排版。
* 为用户显示爬取信息的词云。
* 将在web界面提供调试与编译python代码的功能。
* 针对一些特定网站给出性能优化方案和性能指标对比。

## 项目开发环境

* 1. 操作系统：
     1. Windows7及其以上，安装了python3内核
     2. Ubuntu 16.04及以上，安装了python3内核
  2. 硬件环境：
     1. CPU为当前主流即可，没有硬性要求
     2. 内存4G及其以上
  3. 软件开发环境：
     1. Vscode
     2. Pycharm
  4. 编程语言：

前端使用原生html，后端使用python

* 1. 版本管理工具：

Github

## 项目技术路线

　　为scrapy进行可视化，实现可视化使用是一个B/S架构，服务器和浏览器为同一机器。

　　服务器使用python语言的flask框架进行搭建。前端使用flask框架提供的前端支持与原生html语言，同时使用前端ui包Element.ui进行搭建。这是一个前后端不分离的项目。

　　使用场景为用户在IDE内创建一个实体对象，调用run方法。即可在浏览器内查看localhost的特定端口看到ui调试界面。

　　后端使用flask框架提供用于各个场景的接口。前端编写网页并且调用这些接口完成项目预期实现的功能。与实现一个线上网站的方法比较类似。

# 四、项目组织团队

## 4.1 成员组成

此次项目由我们项目组进行实施完成，我们项目组代号为B组，由六名成员组成。具体人员信息如下表所示：

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | Github账号 |
| 汪凌风 | sy1906514 | Doge-Lin |
| 明昊 | sy1906508 | MHloulis |
| 汪丽萍 | sy1906513 | wangliping102 |
| 郑泽西 | sy1906431 | Zexizheng |
| 闫奕涛 | sy1806220 | mryeator@163.com |
| 邵志钧 | sy1906509 | Francis-Shao |

表3 成员信息

## 4.2 组织结构

　　此次项目的组织形式暂定由组员和组长组成。组长负责沟通协调，安排会议时间与方式等事务。组员应当按照组长的安排要求参加会议，领取任务并按时完成分配的任务。

在经过第一次会议讨论后，我们确定了汪凌风同学作为这次项目组的组长，其余同学为项目组的组员。

## 4.3 人员分工

在经过会议讨论之后，我们确定了人员的具体分工如下表所示。

|  |  |
| --- | --- |
| 姓名 | 工作描述 |
| 汪凌风 | 1. 组员之间的沟通协调，安排各种事宜。 2. 框架扩展阶段对性能进行优化，如：改进Scrapy的去重机制；针对一些特定网站给出性能优化方案和性能指标对比等。 3. Engine，Scheduler，Downloader部分核心代码阅读与分析。 4. 根据任务的具体阶段与成员能力认取任务并完成 |
| 明昊 | 1. 框架扩展阶段输⼊筛选的标签、⽹⻚、显示爬⾍的时⻓和内容部分的实现与调试。 2. Downloader middlewares，PipelineManger部分代码阅读与分析。 3. 根据任务的具体阶段与成员能力认取任务并完成 |
| 汪丽萍 | 1. 框架扩展阶段输⼊筛选的标签、⽹⻚、显示爬⾍的时⻓和内容部分的实现与调试。 2. Spider middlewares，ExtensionManger部分代码阅读与分析。 3. 根据任务的具体阶段与成员能力认取任务并完成 |
| 郑泽西 | 1. 框架扩展阶段在线可视化代码编辑与编译部分的实现与调试。 2. 请求对象、响应对象和数据解析类部分代码阅读与分析。 3. 根据任务的具体阶段与成员能力认取任务并完成 |
| 闫奕涛 | 1. 框架扩展阶段爬取指定⽹⻚显示词云部分的实现与调试。 2. 爬虫类和爬虫管理器部分代码阅读与分析。 3. 根据任务的具体阶段与成员能力认取任务并完成 |
| 邵志钧 | 1. 会议记录。 2. 框架扩展阶段对性能进行优化，如：改进Scrapy的去重机制；针对一些特定网站给出性能优化方案和性能指标对比等。 3. Spiders，Pipeline部分核心代码阅读与分析。 4. 根据任务的具体阶段与成员能力认取任务并完成 |

表4 人员分工

其中，第一周人员分工如下表所示。

|  |  |
| --- | --- |
| 姓名 | 工作描述 |
| 汪凌风 | 组员之间的沟通协调，参与讨论，背景部分文档撰写 |
| 明昊 | 参与讨论，项目概述部分文档撰写 |
| 汪丽萍 | 参与讨论，实际计划部分文档撰写 |
| 郑泽西 | 参与讨论，项目团队组织部分文档撰写 |
| 闫奕涛 | 参与讨论，文档整理与格式调整 |
| 邵志钧 | 会议记录，参与讨论，PPT制作与展示 |

表5 第一周人员分工

第二周人员分工如下表所示。

|  |  |
| --- | --- |
| 姓名 | 工作描述 |
| 汪凌风 | 组员之间的沟通协调，参与讨论，新增项目方案部分文档撰写 |
| 明昊 | 参与讨论，新增项目方案部分文档撰写。 |
| 汪丽萍 | 参与讨论，实际计划部分文档修改与调整。 |
| 郑泽西 | 参与讨论，项目团队组织部分文档修改与调整 |
| 闫奕涛 | 参与讨论，文档整理与格式调整 |
| 邵志钧 | 会议记录，参与讨论，文档一些模块的增加。 |

表6 第二周人员分工

## 4.4 协作与沟通

经过讨论，我们确定了使用腾讯会议和微信群的形式进行沟通。小组成员每周通过腾讯会议的方式进行1至2次的线上会议，对该周具体阶段的任务进行深入分析，完成任务的厘清，成员的具体交流，一些团队决定的做出等任务，通过微信群的方式随时完成团队内成员的交流，问题的讨论，任务的交接等具体事务。项目组内通过文档、Git等工具协作完成相应的具体任务。后续可能通过teambition等软件完善协作与沟通的过程。每次沟通和协作的重要决定与进展都会记录具体内容。在框架调整阶段，通过github实现代码的托管，工作的协调配合与软件版本的变更。

|  |  |  |
| --- | --- | --- |
| **工具** | **频率** | **目的** |
| 腾讯会议 | 每周1至2次 | 对具体阶段的任务进行深入分析，任务厘清，团队交流 |
| 微信群 | 随时 | 成员的交流，问题的讨论，任务的交接等 |
| Github | 随时 | 代码的托管，工作的协调配合与软件版本的变更 |
| 文档等 | 随时 | 项目的协作，任务的交接 |
| teambition | 后续可能会使用 | 完善协作与沟通的过程 |

表7 协作与沟通方式

## 4.5 人员贡献率计算方案

在经过讨论之后，我们确定了采用互评的方式来进行评估人员贡献率的方案。其具体准则如下，在项目结束后，每个小组成员以百分制的方式对别的小组成员进行评分。每个成员的最终贡献率等于自己得到的评分除以小组成员的总分。

　为成员i以百分制的方式对成员j做出的评分。

# 五、实施计划

## 5.1 方法与工具

　　通过MS project对项目进行计划安排和追踪，每阶段针对计划完成情况动态调整计划。

## 5.2 总体进度安排

软件进度的总体安排按照老师要求以周为单位进行，每周针对特定任务，小组开会确定任务分工和计划安排，小组成员按分工内容和工期来完成本周任务。

完成项目最后期限：本学期校历第十七周。

项目分工原则：本项目的分工依据为组员的个人特长、情况及阶段任务量。在确定项目目标的过程中，通过多次组内会议确定了每位成员的个人优势。组长综合每位成员的个人特点以及个人意愿对任务进行了整体上的划分，其中任务分工见第四章节。

进度安排的原则是首先估算项目总规模大小，然后估算工作规模所需的时间成本，最后基于估算结果制定项目进度计划。

由于每周可能有无法预料的情况变化，因此小组在每周六都会进行讨论。对下周的任务进行明确、细分，并对MS Project中的项目规划进行更新。

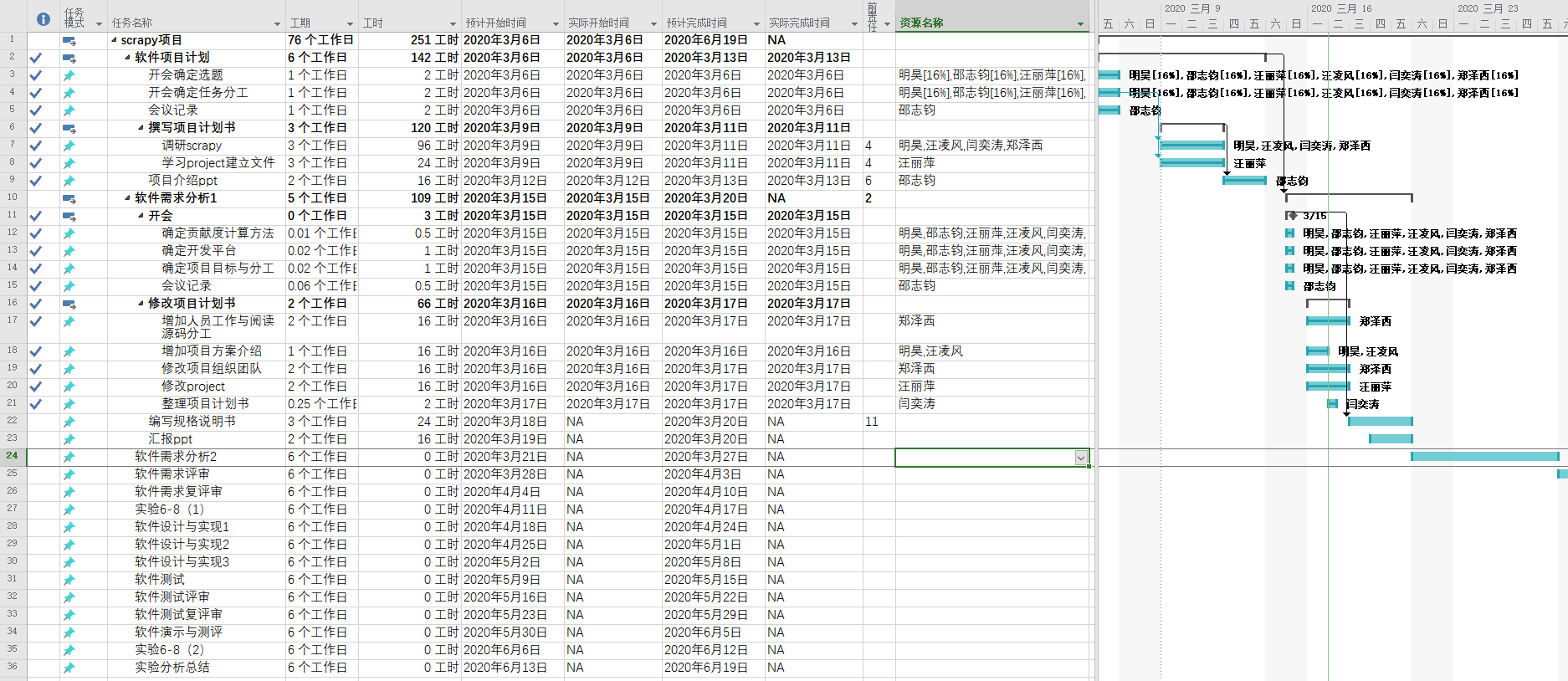


图1 总体进度安排