|  |
| --- |
| 北京航空航天大学 |
| 软件开发计划书 |
| **Scrapy** |
|  |
| SY1606324 郭炜锋 SY1606116 王益飞 SY1606322 胡 勇 SY1606416 武丁泽宇 |
| **2017/03/20**  目录  [1. 项目概述 33](#_Toc477786370)  [1.1 背景 33](#_Toc477786371)  [1.2 简介 33](#_Toc477786372)  [1.3 定义 33](#_Toc477786373)  [2. 项目内容 44](#_Toc477786374)  [2.1 Scrapy框架 44](#_Toc477786375)  [2.2 工作内容 55](#_Toc477786376)  [2.3 主要参加人员 55](#_Toc477786377)  [3. 项目计划 55](#_Toc477786378)  [3.1 工作任务分解与人员分工 55](#_Toc477786379)  [3.2 组织形式 55](#_Toc477786380)  [3.3 进度安排 66](#_Toc477786381)  [4. 参考文献 66](#_Toc477786382) |

# 项目概述

## 背景

随着网络的迅速发展，万维网成为大量信息的载体，如何有效地提取并利用这些信息成为一个巨大的挑战。搜索引擎(Search Engine)，例如传统的通用搜索引擎AltaVista，Yahoo!和Google等，作为一个辅助人们检索信息的工具成为用户访问万维网的入口和指南。但是，这些通用性搜索引擎也存在着一定的局限性。未解决局限性，定向抓取相关网页资源的聚焦爬虫应运而生。聚焦爬虫是一个自动下载网页的程序，它根据既定的抓取目标，有选择的访问万维网上的网页与相关的链接，获取所需要的信息。与通用爬虫(general purpose web crawler)不同，聚焦爬虫并不追求大的覆盖，而将目标定为抓取与某一特定主题内容相关的网页，为面向主题的用户查询准备数据资源。

## 简介

Scrapy是一个为了爬取网站数据，提取结构性数据而编写的应用框架。 可以应用在包括数据挖掘，信息处理或存储历史数据等一系列的程序中。

所谓网络爬虫，就是一个在网上到处或定向抓取数据的程序，当然，这种说法不够专业，更专业的描述就是，抓取特定网站网页的HTML数据。抓取网页的一般方法是，定义一个入口页面，然后一般一个页面会有其他页面的URL，于是从当前页面获取到这些URL加入到爬虫的抓取队列中，然后进入到新页面后再递归的进行上述的操作，其实说来就跟深度遍历或广度遍历一样。

Scrapy 使用 Twisted这个异步网络库来处理网络通讯，架构清晰，并且包含了各种中间件接口，可以灵活的完成各种需求。

Scrapy有以下特点：

* 采取可读性更强的xpath代替正则
* 强大的统计和log系统
* 同时在不同的url上爬行
* 支持shell方式，方便独立调试
* 写middleware,方便写一些统一的过滤器
* 通过管道的方式存入数据库

## 定义

专业术语定义如下：

网络爬虫：是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本；

引擎（Scrapy Engine）：用来处理整个系统的数据流处理，触发事务；

调度器（Schedule）：用来接受引擎发过来的请求，压入队列中，并在引擎再次请求的时候返回；

下载器（Downloader）：用于下载网页内容，并将网页内容返回给蜘蛛。

蜘蛛（Spiders）：蜘蛛是主要干活的，用它来制订特定域名或网页的解析规则。编写用于分析response并提取item或额外跟进的URL的类。每个spider负责处理一个或一些特定网站；

项目管道（Item Pipeline）：负责处理由蜘蛛从网页中抽取的项目，主要任务是清晰、验证和存储数据。当页面被蜘蛛解析后，将被发送到项目管道，并经过几个特定的次序处理数据；

下载器中间件（Downloader Middlewares）：处理Scrapy引擎与下载器之间的请求及响应；

蜘蛛中间件（Spider Middlewares）：处理蜘蛛的响应输入和请求输出；

调度中间件（Schedule Middlewares）：从Scrapy引擎发送到调度的请求和响应。

# 项目内容

## Scrapy框架



图2.1 Scrapy整体框架

图2.1绿线是数据流向，首先从初始URL开始，Scheduler会将其交给Downloader进行下载，下载之后会交给Spider进行分析，Spider分析出来的结果有两种：一种是需要进一步抓取的链接，例如之前分析的“下一页”的链接，这些东西会被传回Scheduler；另一种是需要保存的数据，它们则被送到Item Pipeline那里，那是对数据进行后期处理（详细分析、过滤、存储等）的地方。另外，在数据流动的通道里还可以安装各种中间件，进行必要的处理。

## 工作内容

选择Scrapy最新版本，了解Scrapy框架和用法，进行针对某网站的爬虫代码的编写，并且进行分析，具体内容如下：

* 熟悉scrapy使用方法
* 熟悉scrapy架构
* 使用scrapy设计针对某网站的爬虫
* 分析当前爬虫的性能、功能，找出可以改进的地方
* 阅读scrapy相关源码，针对当前爬虫做出适应性的修改
* 测试修改

## 主要参加人员

郭炜锋： SY1606324， 北航研究生计算机软件工程专业研一学生；

王益飞： SY1606116， 北航研究生计算机软件工程专业研一学生；

胡 勇： SY1606322， 北航研究生计算机软件工程专业研一学生；

武丁泽宇：SY1606416，北航研究生计算机软件工程专业研一学生。

# 项目计划

## 工作任务分解与人员分工

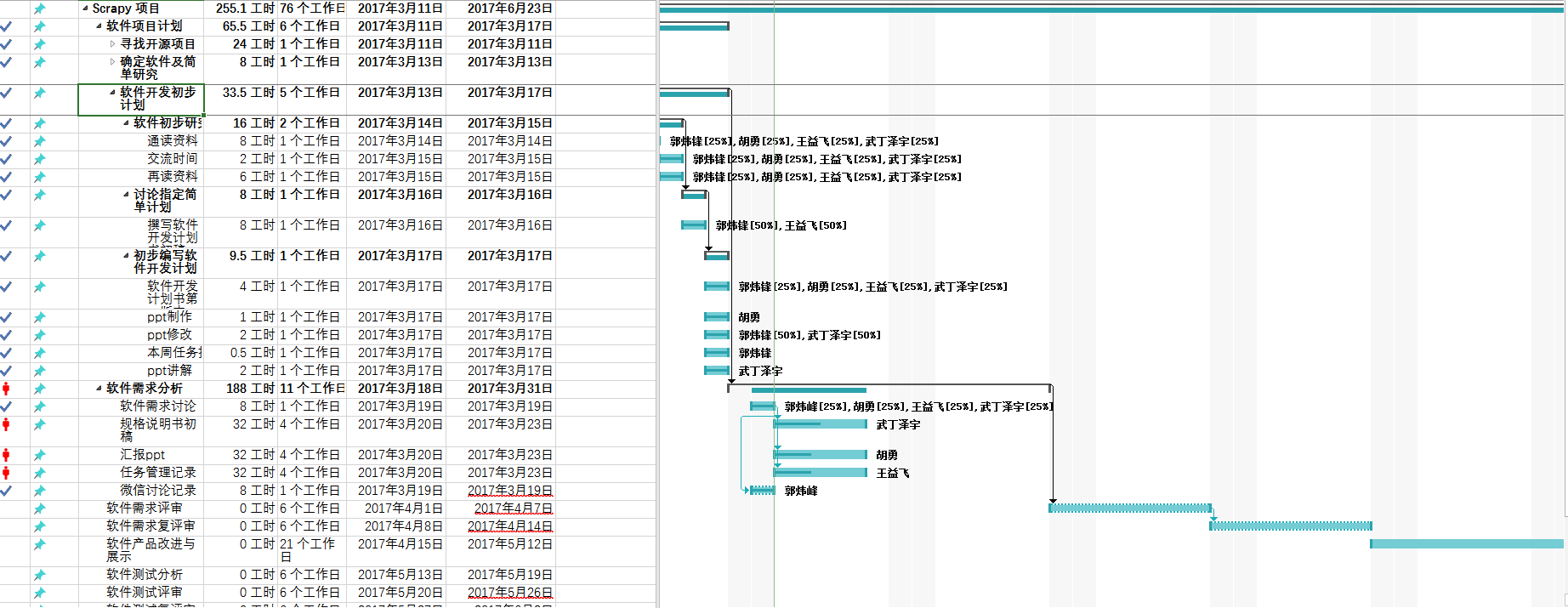
由于暂时还没有讨论出该项目的具体需求是什么，我们针对该项目要做什么具体操作，所以人员分工还需要进一步讨论中；但是其中的一个大目标已经确定了，就是针对某个网站来写爬虫代码，进而分析该项目的性能。

## 组织形式

小组会议跟分工均在微信上进行，并且随时在微信上进行问题讨论，资料分享以及各自进度汇报。

小组在微信讨论后，决定每周五晚上上课之后进行线下讨论，每人至少发言两分钟以上，来交流最近工作、学习的内容和遇到的困难等。

## 进度安排



# 参考文献

Github源码参考：<https://github.com/scrapy/scrapy.git>

Scrapy官网中的pdf学习文档：<https://media.readthedocs.org/pdf/scrapy/1.3/scrapy.pdf>

《用Python写网络爬虫》 出版社: 人民邮电出版社; 第1版 (2016年3月1日)