**项目重构的架构**

**1. 设计新的数据库支持这种模式的。**

**Information表**

10个字段 （TEXT是Sqlite的字符串文本类型）

Keyword 类型 TEXT

Url 类型 TEXT（还是用来存储联系方式页面的Url）

Name 类型 TEXT

Homepage 类型 TEXT (这个新增的，用来存储公司主页的，以免有的公司联系方式页面不存在时，Url为空)

Email 类型 TEXT

Country 类型 TEXT

Address 类型 TEXT

Tel 类型 TEXT

RawInformation 类型TEXT

SearchTimes 类型TEXT （0表示从未被搜索引擎检索过，每检索一次，此值+1，方便以后更新数据的时候有标识）

**Url表：**

5个字段 （TEXT是Sqlite的字符串文本类型）

Keyword 类型 TEXT

Title 类型 TEXT

Url 类型 TEXT

Country 类型 TEXT

Dealed 类型 TEXT（0为未处理，1为已经处理过了）

**2.黄页信息提取模块**

**2.1 重写共用母类**

* **逻辑调度引擎**
* **爬虫伪装cookie以及User-agent**
* **网页信息下载器**
* **任务队列**
* **Url入口规则生成器（包含主入口Url以及每个页面Url入口规则）**
* **列表页面解析器（若列表中已经包含了公司信息，则包含公司页面解析）**
* **公司页面解析器**
* **多线程（保证原子性，避免数据交错）**
* **数据整合与生成**
* **数据库读写**

**2.2 每个具体黄页或者其他信息端，继承母类别，重载方法**

**黄页页面解析主要分成3种：**

1. 在搜索结果页面已经包含了所有详细信息。比如Yell
2. 在搜索结果页面，比如进入黄页的公司页面，才能开始抓取信息。比如第一次做的那个网站Amarillas
3. 在搜索结果页面几乎没有什么联系信息，大部分需要google。比如192

其中，继承后，主要需要重载：

* Url入口规则生成器（包含主入口Url以及每个页面Url入口规则）
* 列表页面解析器
* 公司页面解析器
* 第2种情况，需要特别处理：先把公司页面存入数据库，然后全部存完以后开始解析。解析过程中需要及时标识哪些是处理过的。

**3.搜索引擎补全信息模块**

**3.1.取待处理队列:**

在Information表中，

SELECT \* FROM Information

WHERE **SearchTimes=’0’** AND **Email=’’**

**当搜索次数为0，并且email为空的时候，构建一个待搜索队列。**

**3.2寻找公司主页模块**

**重构原来的模块1**

**主要目标**：

* 取待处理队列
* 通过公司的关键词
* 准确获得公司主页。

**3.3 寻找公司联系信息模块**

**重构原来的模块2**

**主要目标**：

* 获得公司主页后，找寻这个主页的联系信息。
* 此时需要过滤掉无用的非公司网站，比如wikipedia.org等

**3.3 数据更新模块：**

**整合并重构以前的几个信息过滤、去重复等模块。**

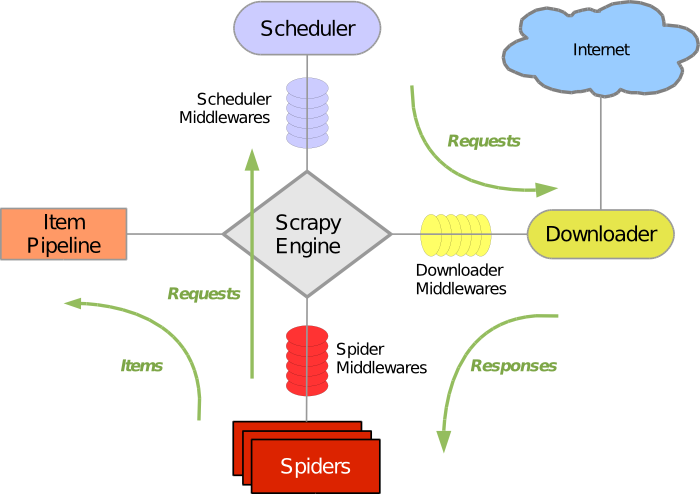
**主要目标：**

* 获得搜索引擎返回的数据，比对此数据与数据库中数据的关系，比如有多个email，则增量更新。
* 去重复
* 过滤（邮箱过滤和网址过滤）
* 补全缺失的信息

**4. 文件I/O模块**

* 输入关键词：Keywords.txt
* 输入地区：Location.txt
* 输入邮箱过滤规则：FilterMails.txt
* 输入网站过滤规则：FilterRegular.txt

**整个框架的调度图可以参考下图：**



关于参考的框架是Python一个优秀的开源爬虫，Scrapy

<http://www.scrapy.org/>