# 

**可配置化网络爬虫项目总体设计说明书**

**小组名称：**可配置化网络平台小组

**小组口号：**我们爱学习

**指导教师：**陈博老师

**文档撰写人：**吴祥龙 关玉婷

**文档撰写时间：**2017年6月20日

团队分工记录表

|  |  |  |  |
| --- | --- | --- | --- |
| **项目名称** | **学号** | **姓名** | **分工** |
| 可配置化网络爬虫 | SA17225405 | 吴祥龙 | 爬虫框架框架设计报告撰写 |
| SA17225094 | 关玉婷 | 爬虫框架框架设计报告撰写 |
| SA17225362 | 王润 | 爬虫运行设计报告撰写 |
| SA17225362 | 唐冬鑫 | 爬虫运行设计报告撰写 |

目录

[4 分布式爬虫的整体设计 2](#_Toc24232)

[4.1 整体架构 2](#_Toc13080)

[4.1.1 ReourceManager(以下简称RM) 2](#_Toc4499)

[4.1.2 Woker 2](#_Toc26861)

[4.1.4 Spider 3](#_Toc14315)

[4.1.5 Redis 3](#_Toc899)

[4.1.6 ZK集群 3](#_Toc4085)

[4.1.7 cassandra集群 3](#_Toc4637)

[4.1.8 客户端 3](#_Toc30108)

[4.2 系统的主要工作流程 3](#_Toc11240)

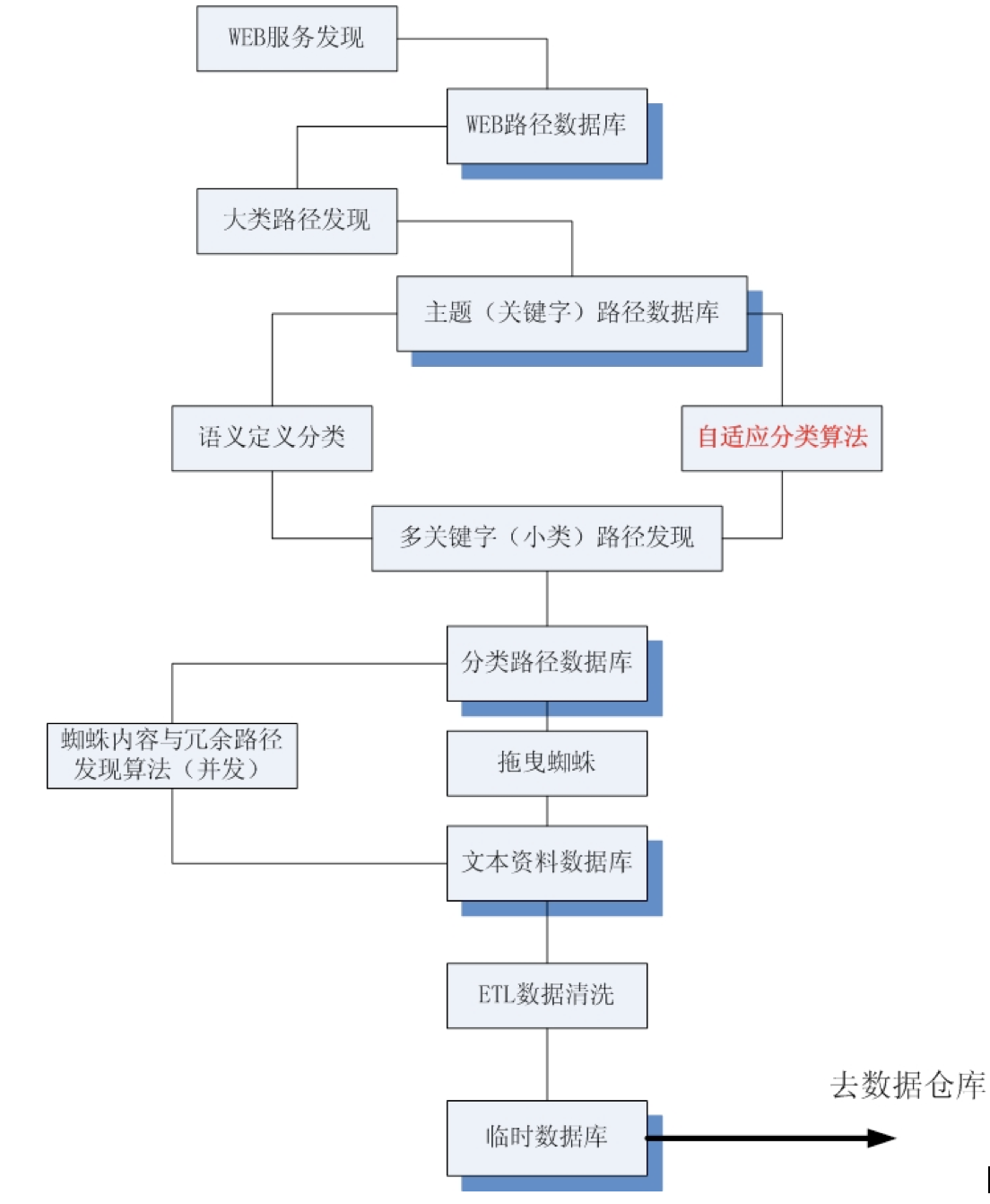
[4.2.1 爬虫任务在集群中启动的流程 4](#_Toc19099)

[调度模块活动图 8](#_Toc15852)

[4.2.2 爬虫运行的流程 8](#_Toc31020)

# 

# 1 分布式爬虫的整体设计



## 1.1 整体架构

本论文实现的系统本质是一个分布式的后台应用程序,架构图如下图所示

### 1.1.1 ReourceManager(以下简称RM)

RM管理Spider集群的资源，RM通过Zookeeper集群(以下简称ZK)来发现集群中的新的Worker节点，RM通过与Worker的信息交互，管理着集群的资源。RM负责启动新的Spider节点，并管理着他们的状态。

### 1.1.2 Woker

Worker是一台运行Spider节点的机器的抽象，Worker管理着这个机器的资源，并监控在这个Wroker上运行的Spider的状态，并把Spider状态的变化汇报给RM，如果一台Spider的状态异常，Worker会把Spider杀死，并汇报给RM

客户端

Spider

Cassandra集群

Spider

ZK集群

RM

Spider

Worker

Worker

Redis

图 4-1 整体架构图

### 1.1.4 Spider

Spider是对爬虫任务的抽象，不同的爬取任务称为一个Spider，Spider主要负责从Redis里取出网页链接，并下载，抽取网页的链接与数据，并把新的网页链接放到redis里，把数据放到Cassandra集群里。

### 1.1.5 Redis

Redis主要用来缓存爬虫任务的网页链接，Redis被当做队列来使用，每个爬虫任务在Redis里都会有一个网页链接的队列，这样保证了爬虫任务之间不会彼此影响

### 1.1.6 ZK集群

ZK集群主要是负责服务的发现，Worker启动的时候会在Zk集群里注册一个临时的节点，当这个Worker挂了这个临时节点会自动的删除，RM通过监听ZK集群的节点来发现新的Worker的加入和Worker的死亡

### 1.1.7 cassandra集群

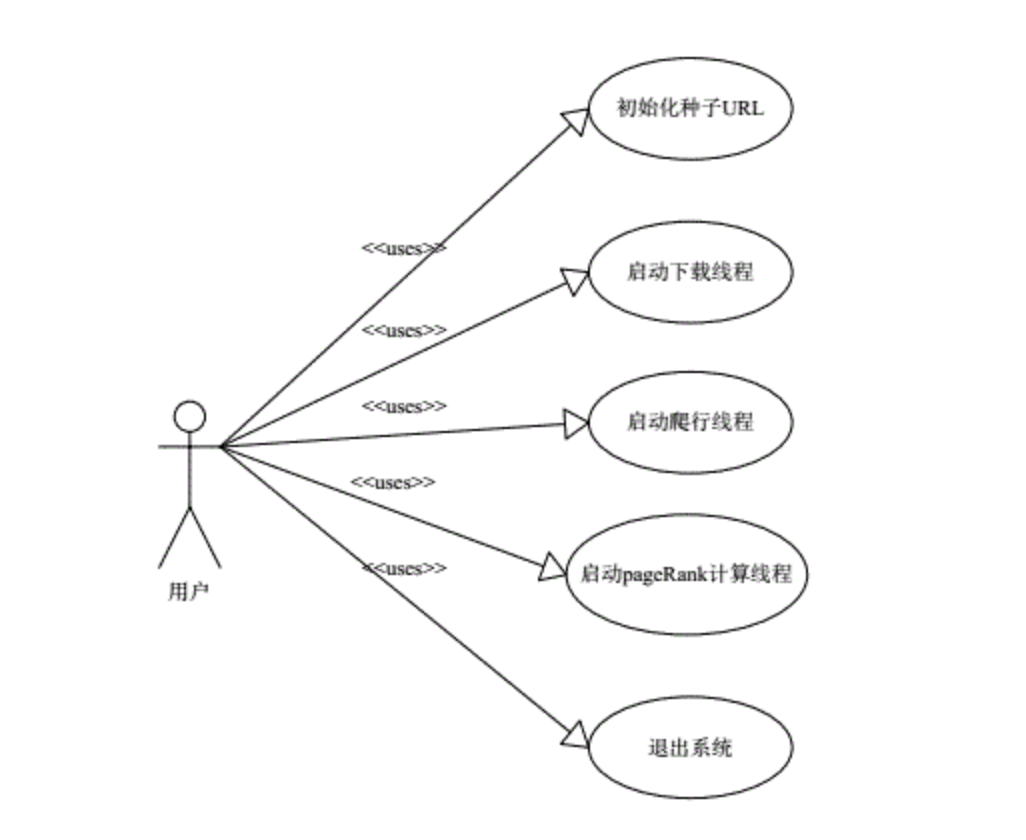
Cassandra 是一个分布式的列式的Key-Value数据库，它主要负责存储Spider爬去的数据。

### 1.1.8 客户端

客户端主要提供了对集群的访问。客户端通过连接RM 向RM发送命令，远程调用RM的方法实现对集群资源的访问。

## 1.2 系统的主要工作流程

系统用例图：



系统整体用例图

### 1.2.1 爬虫任务在集群中启动的流程

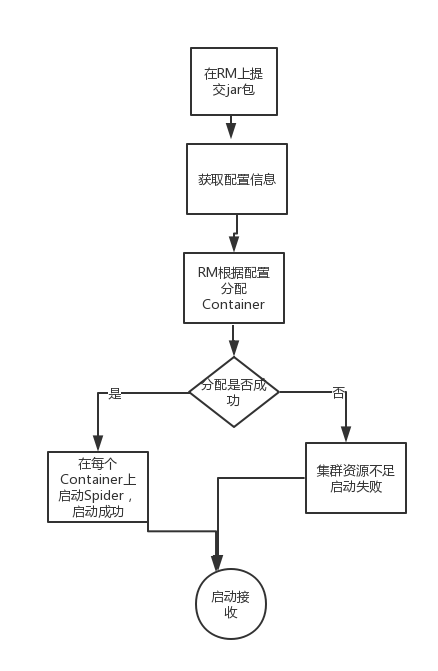
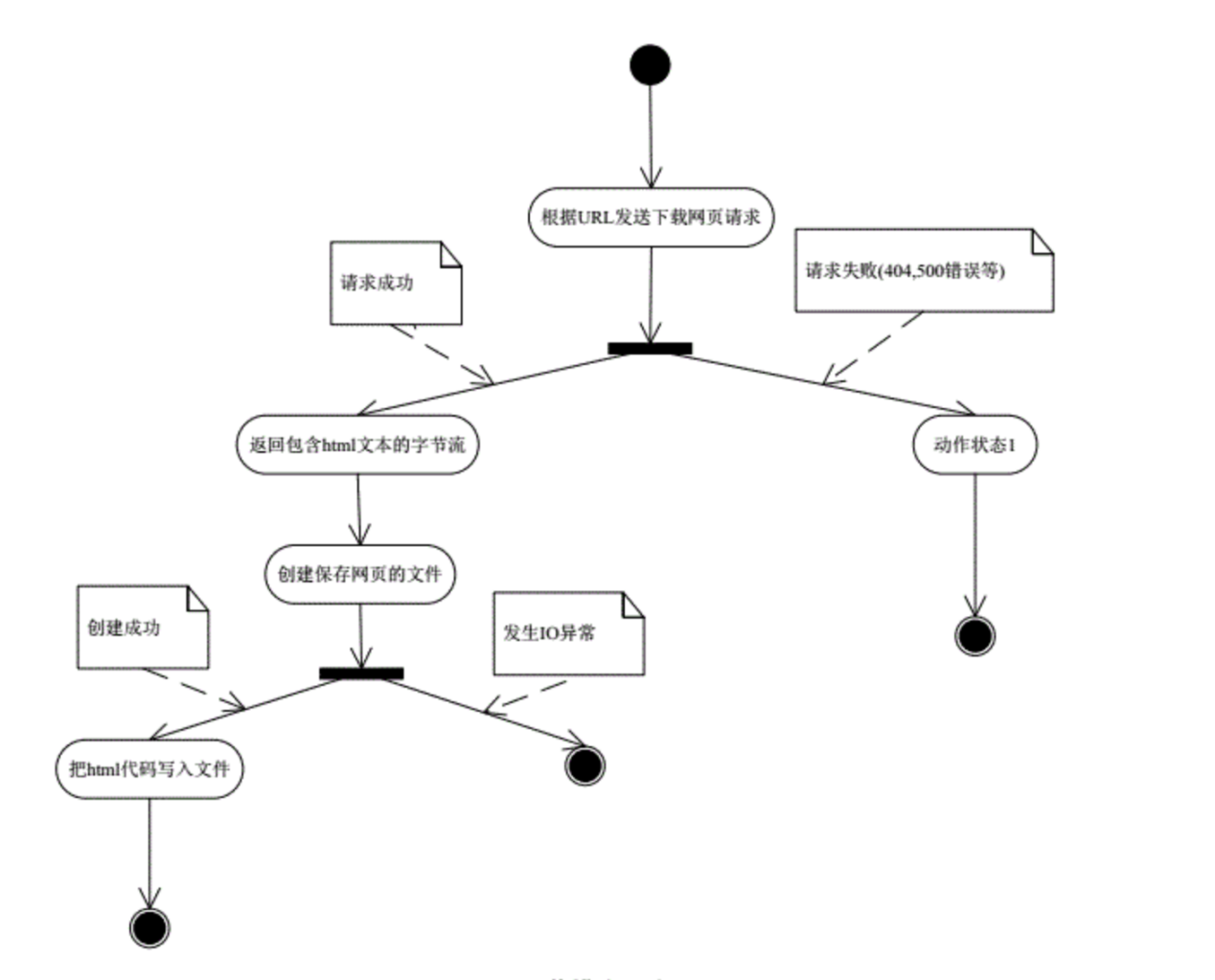


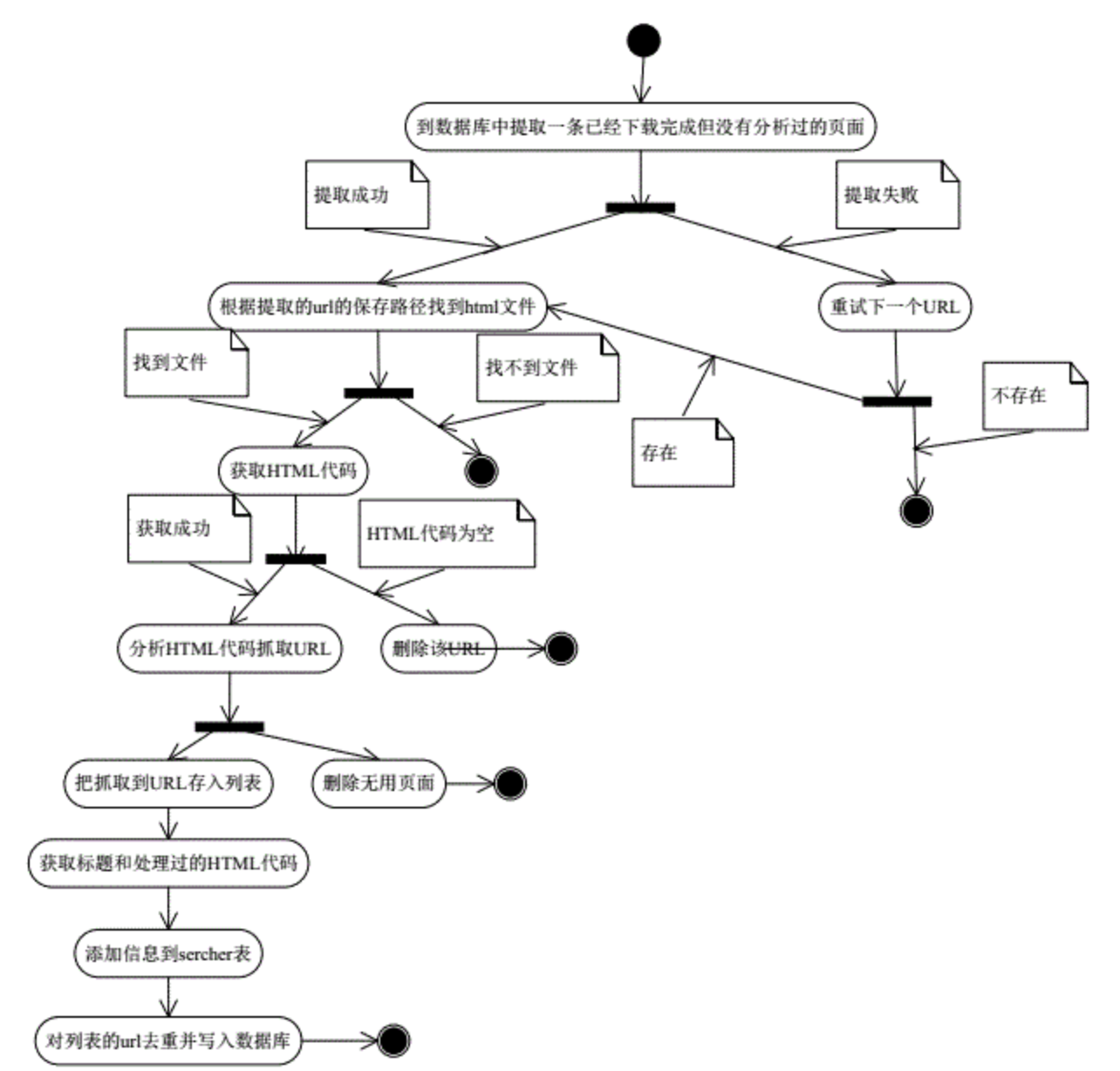
图 4-2 爬虫任务启动图

在爬虫任务启动之前，要先把爬虫的jar包上传的RM的服务器上，jar包里主要包含这个爬虫的所有配置以及爬虫爬取数据的逻辑实现，然后通过命令将jar包提交的RM上，RM读取爬虫的配置，并为爬虫的运行分配Container，如果失败说明集群的资源不足，爬虫启动失败，如果分配成功RM启动Container 爬虫启动成功。

Download模块活动图：

 Download模块活动图

Parse模块活动图：

解析模块活动图

Scheduler模块活动图：

### 调度模块活动图

### 1.2.2 爬虫运行的流程

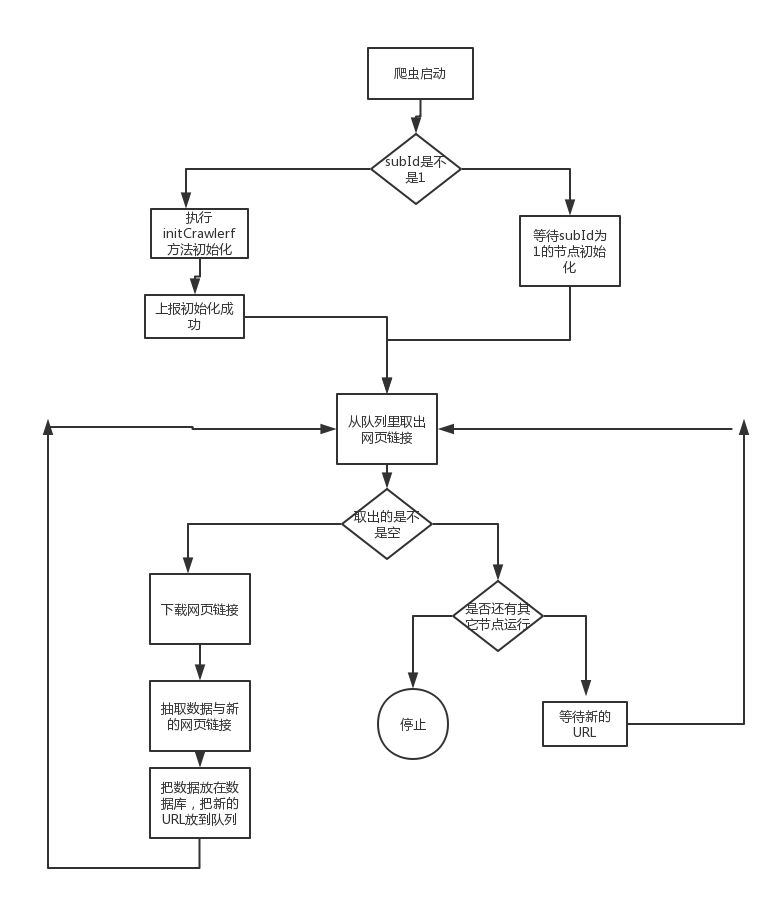
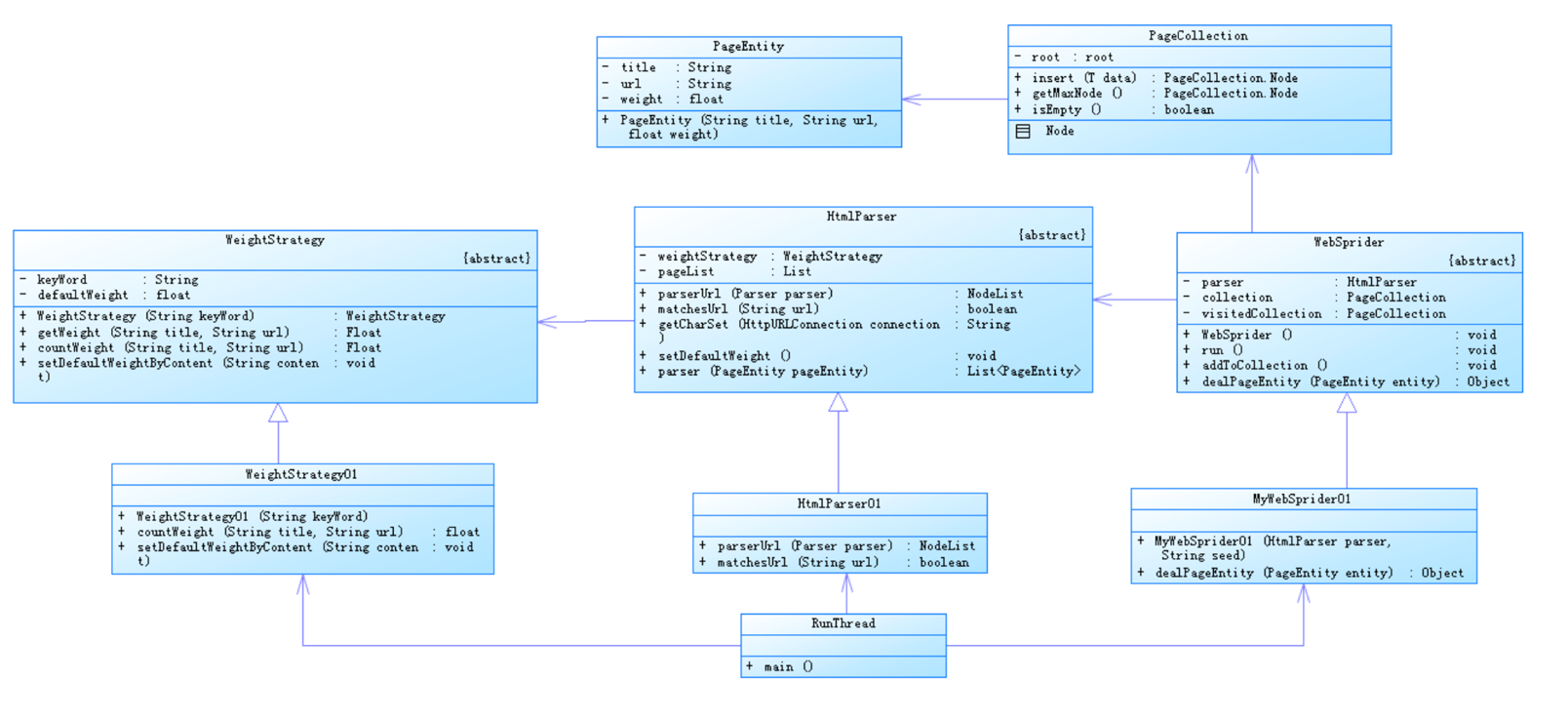


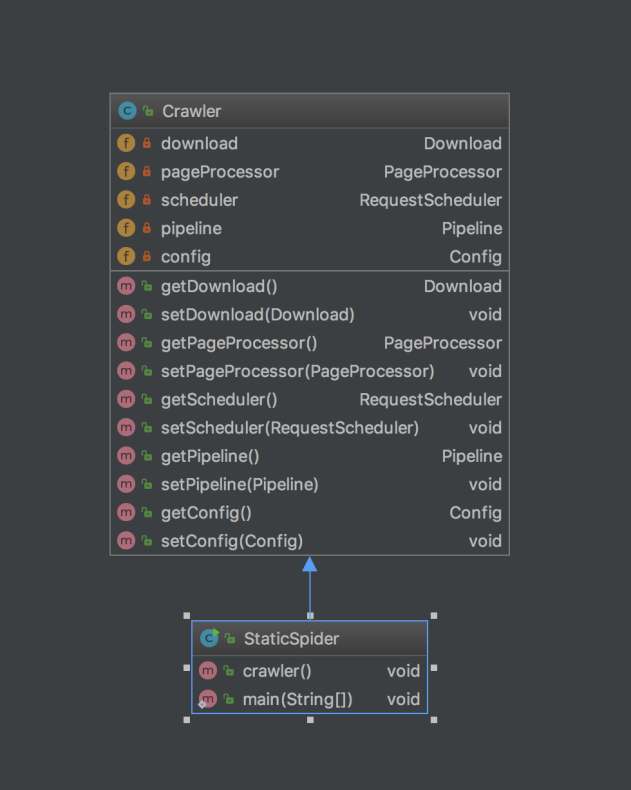
图 4-3 爬虫运行图

Worker收到RM的启动爬虫的命令后为Spider内存，并启动爬虫任务爬虫在Worker上的运行流程如上图。每个爬虫任务都存在以一个initCrawler方法，initCrawler方法是爬虫的初始化方法，因为爬虫是分布式的，一个爬虫有多个爬虫节点在运行，所以要保证initCrawler只执行一次，并且要在其它节点没有运行的时候运行。每个爬虫的爬虫节点都有一个编号叫subid，本文的做法是在爬虫初始化的时候选取subid为1的爬虫节点运行initCrawler方法，当执行完毕后会想Worker上报状态 worker通知其他节点。subId不等于1的节点等待SubId为1的节点initCrawler方法执行完毕。当初始化过程执行完毕后，爬虫会从Redis队列里取出网页链接，如果去的的链接不为空，从网络中下载网页链接对应的网页，并构造一个Dom树，然后执行数据抽取，抽取爬虫想要的数据放入数据库，抽取新的网页链接放到redis里。如果网页链接为空，判断是不是还有其它的节点在运行，如果没有爬虫运行结束如果有等待新的URL，并继续上述操作。

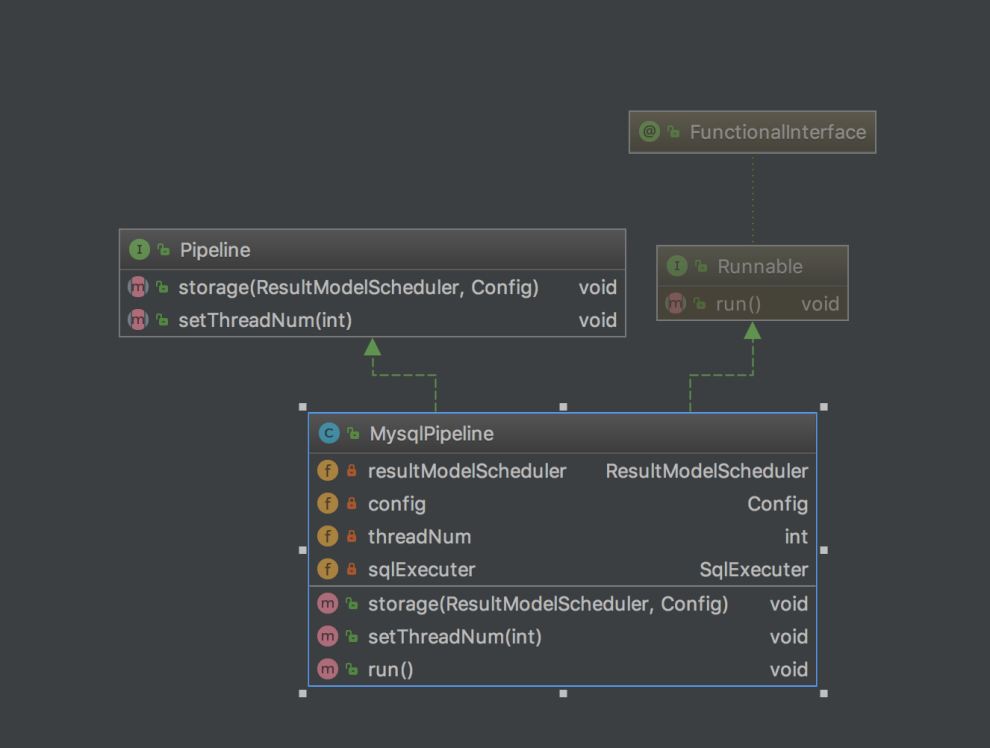
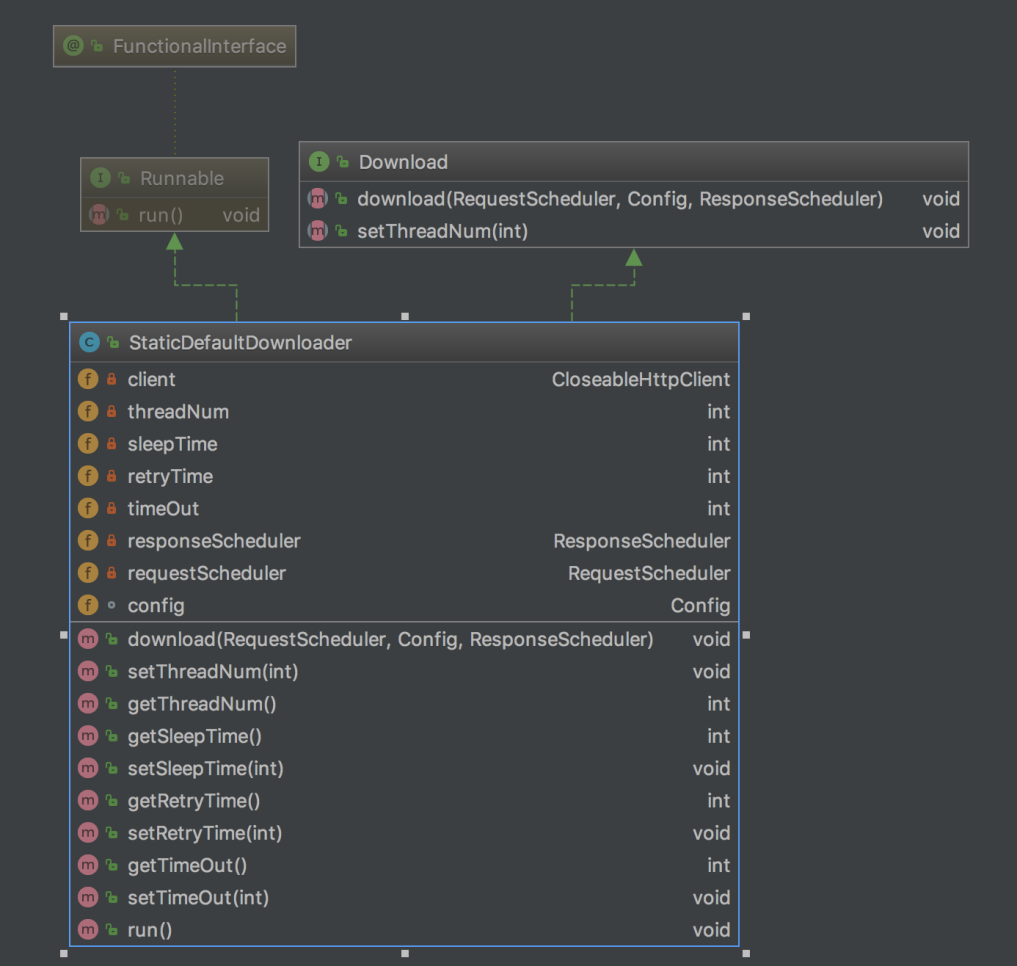
爬虫核心类图：



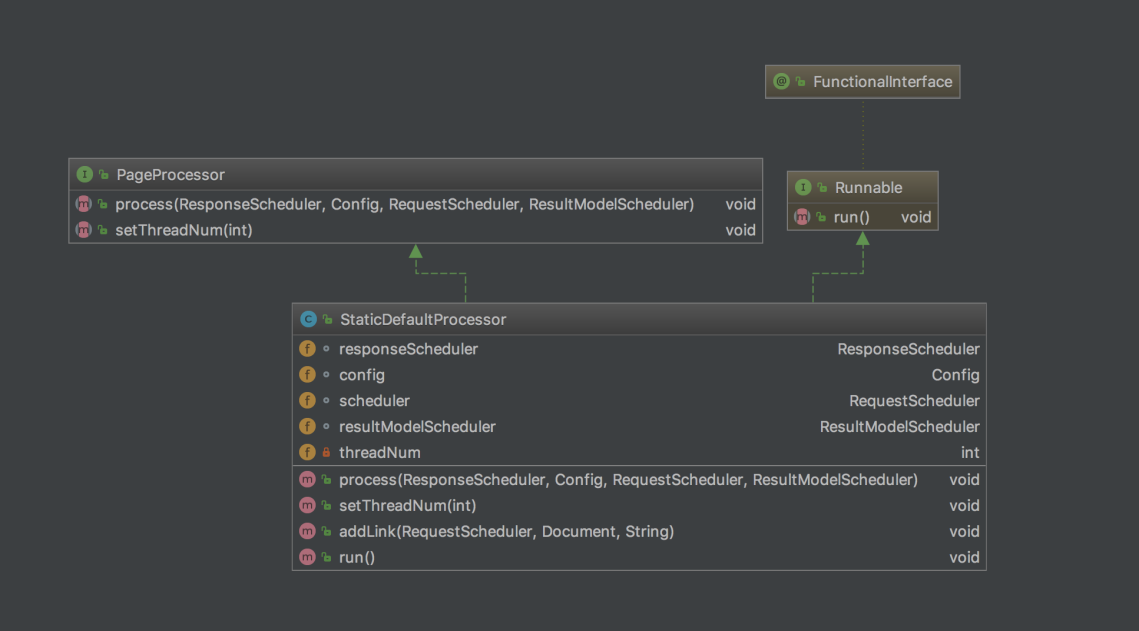
核心类图

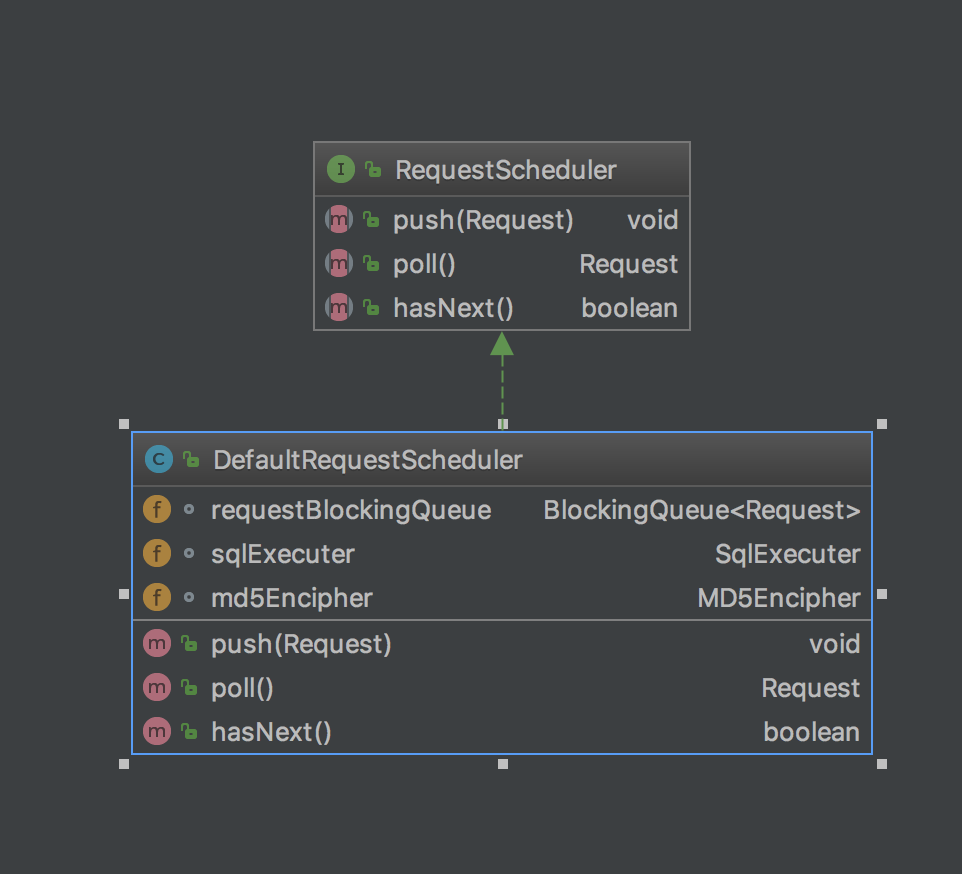
Spdier类图

Downloader类图

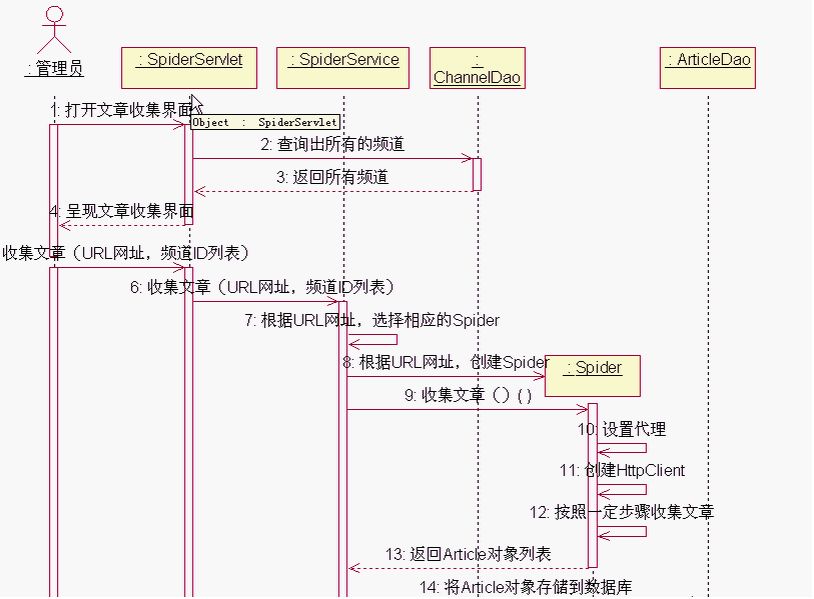


Pipeline类图:

Processor类图：

RequestScheduler类图：

系统处理流程时序图：



1.3 本章小结

本章主要对这篇文章要实现的分布式爬虫系统的架构做了一个详细的介绍，然后介绍了系统的主要工作流程。