

Seek Movies爬虫部分

体系结构文档



2017-5-28

南京大学软件学院

南京市汉口路

目录

[系统说明 2](#_Toc483818798)

[1.1系统概述 2](#_Toc483818799)

[1.2主要功能 2](#_Toc483818800)

[2. 逻辑视图 3](#_Toc483818801)

[2.1部件、连接件视图 3](#_Toc483818802)

[2.2 部件连接件描述 3](#_Toc483818803)

[2.3配置视图 6](#_Toc483818804)

[3. 开发视图 6](#_Toc483818805)

[4.进程视图 7](#_Toc483818806)

[5.部署视图 8](#_Toc483818807)

# 系统说明

## 1.1系统概述

本系统用于从百度糯米和爱奇艺这些网站上面爬取电影信息、电影票信息、影院信息，为Seek Movies的对外网站部分提供数据支持。由于要爬取的网站皆有IP访问频率上限，因此决定采用分布式架构，通过变换不同的IP地址，来绕过IP屏蔽。

## 1.2主要功能

1） URL仓库：用于保存要爬取的URL

2） 控制台：用于操控爬虫进行数据爬取，查看爬虫状态的顶层界面

3） 监控器：用于监视爬虫是否正常运行

4） 控制器：控制台的底层实现，用于控制爬虫的动作

5） HTML解析器：用于解析获取的HTML页面，同时将爬取到的URL保存到URL仓库

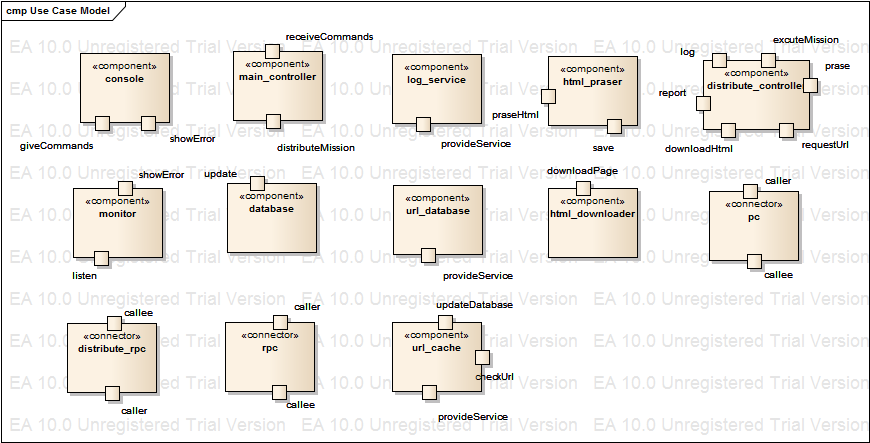
6） 数据仓库：用于保存解析后的数据

7） 页面下载器：用于下载页面

8）日志管理： 记录爬虫的状态，如果爬虫失效，可以根据任务重启

# 2. 逻辑视图

## 2.1部件、连接件视图



## 2.2 部件连接件描述

|  |  |  |
| --- | --- | --- |
| console | | |
| 类型 | component | |
| 作用 | 提供给用户操作界面，显示爬虫的工作状态 | |
| 端口 | giveCommands | 向爬虫分配任务，控制爬虫启动等等 |
| showError | 如果爬虫失效，显示爬虫的失效信息 |

|  |  |  |
| --- | --- | --- |
| main\_controller | | |
| 类型 | component | |
| 作用 | Console的逻辑层，这里包括任务分配算法，由于采用分布式爬虫，所以爬取重复的页面很可能会发生，这里的算法要降低爬虫碰撞的频率，也要保证效率，同时，用于网站存在IP屏蔽机制，要合理的安排调度算法，减少被IP屏蔽的概率 | |
| 端口 | recieveCommands | 用于接收来自console的命令，并解析命令 |
| distributeMission | 用于向爬虫分派任务 |

|  |  |  |
| --- | --- | --- |
| log\_service | | |
| 类型 | component | |
| 作用 | 用于记录爬虫的爬取日志，具体做法是每个30S，记录爬虫爬取队列中的URL，如果爬虫失效，重启的时候就可以根据日志中记录的URL爬取。 | |
| 端口 | provideService | 提供日志记录服务，一个定义比较宽泛的接口 |

|  |  |  |
| --- | --- | --- |
| html\_praser | | |
| 类型 | component | |
| 作用 | 用于解析html页面，并把解析得出的数据存入数据库 | |
| 端口 | praseHtml | 传入html页面，解析之 |
| save | 将解析得到的数据存入数据库 |

|  |  |  |
| --- | --- | --- |
| distribute\_controller | | |
| 类型 | component | |
| 作用 | 每个爬虫局部的控制器，负责调度爬虫的各个部分使之协调工作，每个模块是属于不同线程的，要考虑的线程的一系列问题 | |
| 端口 | log | 用于记录日志和读取日志 |
| report | 每隔20s向monitor发送一次heart beat，如果没有发送，那么会被认为爬虫死掉了。同时如果爬虫内部抛出exception，要通知monitor，让用户即使发现异常 |
| excuteMission | 为main\_controller提供执行接口 |
| prase | 将html数据交给html\_praser进行解析 |
| requestUrl | 从url\_cache中提取要爬取的url，同时也负责将要爬取的url放入url\_cache |
| download\_html | 将url传入html\_downloader模块，并从中获取下载的html页面 |

|  |  |  |
| --- | --- | --- |
| monitor | | |
| 类型 | component | |
| 作用 | 用于监控爬虫，如果爬虫失效，要通知用户 | |
| 端口 | listen | 监听爬虫是否失效 |
|  | showError | 向用户的控制台显示爬虫的错误信息 |

|  |  |  |
| --- | --- | --- |
| database | | |
| 类型 | component | |
| 作用 | 保存从html中解析出来的数据 | |
| 端口 | update | 更新数据库 |

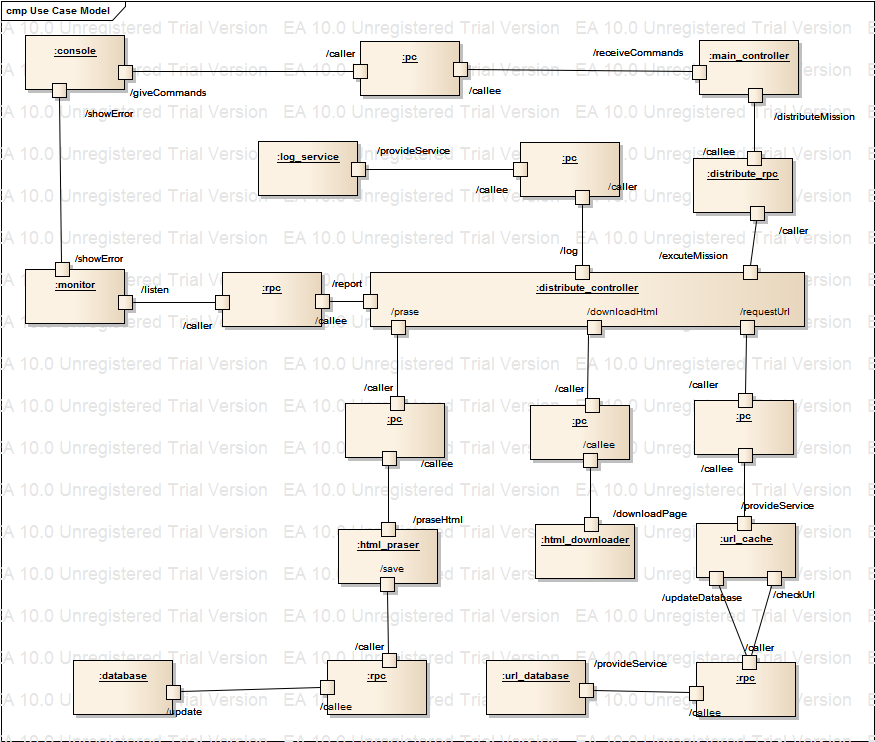
|  |  |  |
| --- | --- | --- |
| url\_database | | |
| 类型 | component | |
| 作用 | 保存爬取的页面的url，是一个url的仓库，推荐使用Berkeley DB（一种非关系型数据库，保存的形式类似于hash map，但是又比hash map省空间，同时支持高并发，在url去重的个方面能力很大，对于我们这些不懂布隆过滤器的菜鸟来说，是一个很好的下位替代品） | |
| 端口 | provideService | 向url\_cache提供url保存，url状态更新服务 |

|  |  |  |
| --- | --- | --- |
| html\_downloader | | |
| 类型 | component | |
| 作用 | 用于下载html页面，推荐使用Java + phantom 来进行网站页面的爬取，一键傻瓜式解决ajax和重定向问题 | |
| 端口 | downloadPage | 下载页面 |

|  |  |  |
| --- | --- | --- |
| url\_cache | | |
| 类型 | component | |
| 作用 | 用于保存需要爬取的url的队列，负责避免url重爬、修改url\_database中url的爬取状态等功能 | |
| 端口 | provideService | 向distribute\_controller提供一条url |
|  | updateDatabase | 如果遇到新的url，通过这个端口更新url\_database |
|  | checkUrl | 如果遇到的不是新的url(貌似不好判断是不是新的，好像可以和updateDatabase合并)，通过这个端口修改url的爬取状态 |

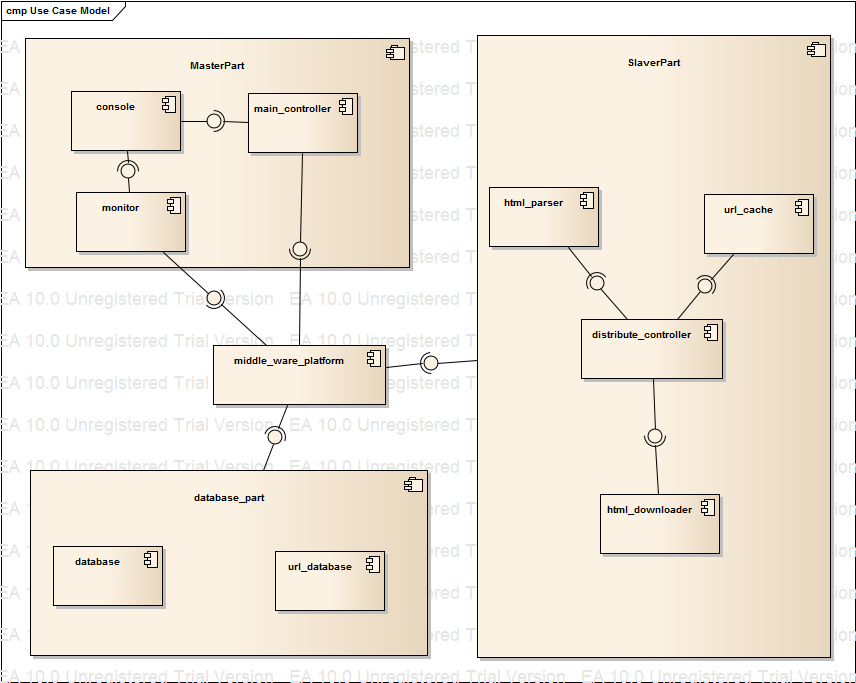
剩下3个连接件大家应该都懂。

## 2.3配置视图



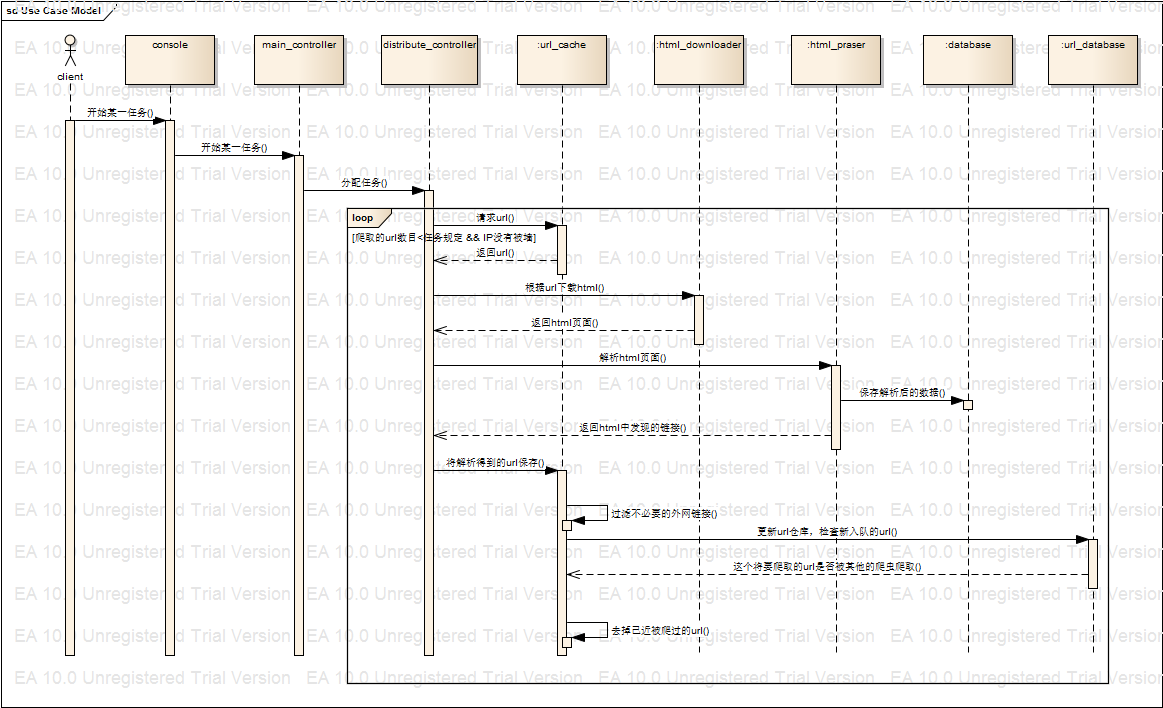
# 3. 开发视图

考虑到我们组有5个人，要负责的部分大体分为3个部分：数据展示部分（包括网页前端和后台），网页数据解析部分（实际上是体力活，估计会用到XML的相关知识），以及数据集成部分。由于数据集成部分的工作量不大（其实就是数据展示部分的数据层），所以个人人为爬虫部分2.5个人来写比较合适，由此，开发视图如下：（注：这个只是按照项目分包的情况下来分的，实际上的分工不一定这样）



# 4.进程视图

由于monitor这个太简单了，只给出任务下达到执行的视图



# 5.部署视图

为了防止被网站防火墙屏蔽，需要多个IP地址进行访问来保证效率，所以有多个从机。

