# 1.爬虫基本思路及目标

## 1.1目标

使用webmagic框架爬取CSDN博客的内容，并进行分析处理，然后存储到数据库中。

## 1.2 首页

首先将首页内容分块处理。





分析主页的链接可以看到，大致有如下四种不同的分类：

技术文章类型的分类，单独的文章链接，专家或者用户的主页链接，当前类型文章的分页情况。

在使用爬虫进行爬取的时候，主要就是抓取与博客相关的链接，同时屏蔽其他的无关链接。

## 1.3 分类和分页的抓取

分类和分页的抓取与主页的抓取具有相同的逻辑。

## 1.4 用户

进入用户博客主页链接后，目标是爬取用户的所有文章内容。

分析用户的博客主页，页面元素主要分为用户的文章列表和分页情况，这里抓取用户所有文章的逻辑是获得所有的分页情况，对每一页分别抓取页面上的文章列表。



## 1.4 文章

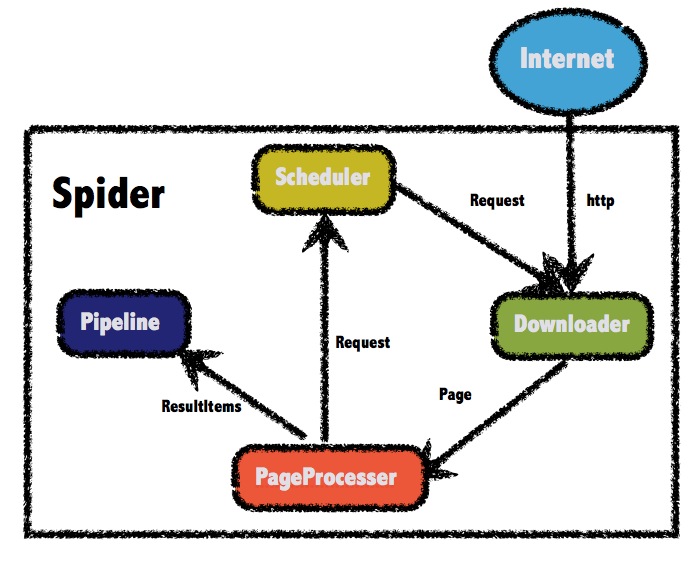
进入文章后。首先保存并处理文章内容，然后依次爬取文章中所有评论用户。对于所有的评论用户，继续使用抓取用户的业务逻辑进行处理。抓取评论用户的所有文章。



# 2．技术难点

## 2.1框架

:WebMagic.的运用



WebMagic的结构分为Downloader、PageProcessor、Scheduler、Pipeline四大组件，并由Spider将它们彼此组织起来。这四大组件对应爬虫生命周期中的下载、处理、管理和持久化等功能。WebMagic的设计参考了Scapy，但是实现方式更Java化一些。

而Spider则将这几个组件组织起来，让它们可以互相交互，流程化的执行，可以认为Spider是一个大的容器，它也是WebMagic逻辑的核心。

## 2.2分析网页架构

分析CSDN博客的整体架构，以及如何对链接进行分类处理。

## 2.3使用正则匹配网页

用户和文章的链接是未知的但是却具有一定的结构特征，通过正则可以有效的抓取出我们需要的链接。

## 2.4 如何存储下载的数据

Webmagic并没有对抓取的网页进行有效的层次结构处理，我们需要自己添加业务逻辑，对网页进行分层和分类。

同时需要考虑使用什么样的格式以及数据库存储下载的数据。

## 2.5对下载的数据归类和处理

技术实现的重难点，爬下来的数据怎么用。

## 2.6如何应对防爬机制。

 有些网站会限制url访问的频率（比如：同1个ip1分钟内只能访问某个页面N次），这种需要手动在代码里控制下节奏，比如每次http请求后，加入sleep(5000)之类的，

 有些网站会根据http请求header中的User-Agent来判断是否是同一款浏览器在恶意刷，这种可以在代码中手动弄一个User-Agent列表，把市面上的所有User-Agent全加进去，每次请求前，随机从列表中取一个User-Agent，这样看起来，貌似有很多不同的浏览器在访问，显得真实一点。

有些网站会限制IP，甚至有IP黑名单机制，我们可以在爬虫代码里设置代理服务器。

# 3.分工

分析网页架构:曹赟程，彭正宇

实现处理过程:万梦玉，彭正宇

研究数据存储方式:曹赟程

对数据进行归类和处理: 万梦玉，彭正宇,曹赟程