



(12) 发明专利申请

(10) 申请公布号 CN 102054028 A

(43) 申请公布日 2011. 05. 11

(21) 申请号 201010590806. 2

(22) 申请日 2010. 12. 10

(71) 申请人 黄斌

地址 100083 北京市海淀区二里庄北里 8 号  
楼 5 门 501 室

(72) 发明人 黄斌

(51) Int. Cl.

G06F 17/30 (2006. 01)

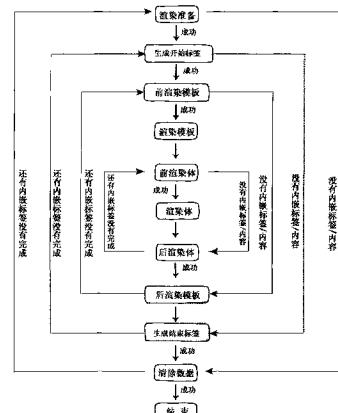
权利要求书 2 页 说明书 5 页 附图 3 页

(54) 发明名称

具备页面渲染功能的网络爬虫系统及其实现方法

(57) 摘要

本发明公开了一种具备页面渲染功能的网络爬虫系统，同时也涉及该网络爬虫系统实现页面渲染功能的方法。该网络爬虫系统包括多个信息采集器、页面分析器、URL 过滤器、页面过滤器、URL 管理器、图片生成器、URL 库和页面库，不仅能够完成一般网络爬虫的功能，还可以将网页直接进行页面渲染，并将渲染结果直接用图片格式加以保存，从而为低成本、高效率地实现页面预览功能奠定技术基础。利用本发明，当我们根据网页的地址进行检索后，不仅可以了解该页面的基本内容，能够看到其基本的显示效果，从而更多地了解整个页面的内容。



1. 一种具备页面渲染功能的网络爬虫系统,其特征在于:

所述网络爬虫系统包括多个信息采集器、页面分析器、URL 过滤器、页面过滤器、URL 管理器、图片生成器、URL 库和页面库;其中,

所述信息采集器位于所述网络爬虫系统的底层,与互联网直接进行交互以获取 Web 页面,所述页面分析器与所述信息采集器进行连接,一方面从页面内容中解析出带有链接标记的 URL,交给所述 URL 过滤器解析;另一方面将页面内容解析为文本格式,交给所述页面过滤器处理;

所述 URL 过滤器对 URL 进行限定站点范围和主题的过滤之后,存入 URL 库中;所述页面过滤器进行页面内容的冗余检测后,将检测后的页面存入页面库中;

所述图片生成器连接所述 URL 库,针对所述 URL 库中存储的 URL 生成页面对应的图片。

2. 如权利要求 1 所述的网络爬虫系统,其特征在于:

所述信息采集器从信息源出发,通过 http 协议请求,下载 Web 页面,所述页面分析器分析页面并提取链接,然后所述信息采集器再以迭代的方式访问网络。

3. 如权利要求 1 或 2 所述的网络爬虫系统,其特征在于:

所述信息采集器采用图的遍历算法搜索 Web 页面。

4. 如权利要求 1 所述的网络爬虫系统,其特征在于:

所述 URL 过滤器利用扩展元数据的语义信息,对从 Web 页面中提取出的 URL 进行主题相关性预测,按照相关链接进行采集、不相关链接直接丢弃的原则进行剪枝处理。

5. 如权利要求 1 所述的网络爬虫系统,其特征在于:

所述 URL 管理器一方面从所述 URL 库中获得 URL 列表,进行任务排列后分配给多个信息采集器;另一方面从多个信息、采集器中获得新的 URL 列表,将这些列表保存到所述 URL 库中。

6. 一种如权利要求 1 所述的网络爬虫系统实现页面渲染功能的方法,其特征在于包括如下步骤:

(1) 生成 Web 页面的开始标签;

(2) 渲染页面模板中的内容,其中每进入一个标签,都依次调用所述标签的各个生命周期阶段;

(3) 渲染 Web 页面中的体;

(4) 生成 Web 页面的结束标签;

(5) 清除数据。

7. 如权利要求 6 所述的网络爬虫系统实现页面渲染功能的方法,其特征在于:

所述步骤(2)中,调用所述标签的各个生命周期阶段是指从上层标签到下层标签的递归入口,只有下层标签渲染结束,进行调用的组件才继续后续阶段的操作。

8. 如权利要求 6 所述的网络爬虫系统实现页面渲染功能的方法,其特征在于:

所述步骤(4)中,生成结束标签的操作由控制内嵌标签执行流程的操作代替。

9. 一种如权利要求 1 所述的网络爬虫系统实现页面渲染功能的方法,其特征在于包括如下步骤:

当发现一个图片标签引用了一张图片时,向服务器发出请求;此时继续渲染后面的代码,服务器返回所述图片的文件,然后重新渲染这部分代码。

10. 如权利要求 9 所述的网络爬虫系统实现页面渲染功能的方法，其特征在于：

当发现存在一个 JavaScript 代码的 <script> 标签时，执行语句，重新渲染部分代码，然后将渲染的结果生成图片。

## 具备页面渲染功能的网络爬虫系统及其实现方法

### 技术领域

[0001] 本发明涉及一种具备页面渲染功能的网络爬虫系统，同时也涉及该网络爬虫系统实现页面渲染功能的方法，属于网络资源搜索技术领域。

### 背景技术

[0002] 据有关媒体报导，美国谷歌(google)公司在2010年10月6日推出了搜索结果可视预览功能，允许用户在搜索结果列表中直接以缩略图的形式预览每个页面。据谷歌公司有关人士介绍，“有时用户点击一个搜索结果，却发现出现的页面与其想要的页面相差甚远。于是用户只能点击返回，再去点击另一个搜索结果。这种体验很差。我们试图以提供预览的方式避免这种情况的发生。”为此，用户将在搜索结果右侧看到一个放大镜标志，点击放大镜就可以看到这个页面的缩略图预览。用户还可以向下滑动，查看所有搜索结果的预览图。

[0003] 为了满足搜索结果可视预览的要求，谷歌公司将存储几十亿个流行度较高的网页的缩略图。对于流行度较低的页面，谷歌公司也通过技术手段在不到十分之一秒的时间内生成缩略图。但是，满足上述要求所付出的硬件成本和软件成本都是巨大的。

[0004] 目前还有一些别的技术手段可以实现页面预览功能，例如使用CGI程序，抓取浏览器的图像区，利用浏览器的绘图功能生成图片。另外，在专利申请号为200910221416.5的中国发明专利申请中，公开了一种利用图像分析对互联网进行自动爬行的方法和装置。对网页组件进行视觉识别的示例性方法包括以下步骤：在网络浏览器中渲染网页以生成图像，利用机器对图像的至少一个部分进行视觉分析以检测包含可能的网页组件的区域。该示例性方法还包括步骤：自动确定检测到的网页组件的类型，并存储该网页组件类型和网页部分的位置。

[0005] 但是，现有技术中并没有利用网络爬虫系统实现页面预览功能的解决方案。网络爬虫(Web Crawler)又称为网页蜘蛛(Web Spider)、网络机器人(Web Robot)，是按照一定的规则自动抓取互联网信息的程序或者脚本组成的系统。它的工作过程可以简述如下：从预先指定的初始URL集(也称种子集)出发，从中选择一个URL，获得该URL所指向的页面，再从这个已经访问的页面中解析出新的URL，并对这些刚刚提取的URL进行分析比较，判断哪些URL还没有被访问过并将它们放入到等待访问的队列，再按照指定的策略从该等待访问队列取出下一个URL继续访问。如此重复，直到等待访问队列为空或满足停止访问条件，其过程与有向图的遍历非常相似。访问的过程中，将该网页的文本内容保存在搜索引擎的数据库中进行分析处理。

[0006] 在这些网络爬虫系统的运行过程中，普遍只将网页的内容按网页文件进行分析，抽取其中的内容。一些网络爬虫系统则更进一步，对这些内容进行简单的处理，如加以语义标注等，方便搜索引擎进行整理排序。但是，这些网络爬虫系统普遍不具备页面渲染的功能，因此并不能方便地实现搜索结果页面预览功能。

## 发明内容

- [0007] 本发明所要解决的首要技术问题是提供一种具备页面渲染功能的网络爬虫系统。
- [0008] 本发明所要解决的另外一个技术问题是提供该网络爬虫系统实现页面渲染功能的方法。
- [0009] 为实现上述的发明目的,本发明采用下述的技术方案:
- [0010] 一种具备页面渲染功能的网络爬虫系统,其特征在于:
- [0011] 所述网络爬虫系统包括多个信息采集器、页面分析器、URL 过滤器、页面过滤器、URL 管理器、图片生成器、URL 库和页面库;其中,
- [0012] 所述信息采集器位于所述网络爬虫系统的底层,与互联网直接进行交互以获取 Web 页面,所述页面分析器与所述信息采集器进行连接,一方面从页面内容中解析出带有链接标记的 URL,交给所述 URL 过滤器解析;另一方面将页面内容解析为文本格式,交给所述页面过滤器处理;
- [0013] 所述 URL 过滤器对 URL 进行限定站点范围和主题的过滤之后,存入 URL 库中;所述页面过滤器进行页面内容的冗余检测后,将检测后的页面存入页面库中;
- [0014] 所述图片生成器连接所述 URL 库,针对所述 URL 库中存储的 URL 生成页面对应的图片。
- [0015] 其中,所述信息采集器从信息源出发,通过 http 协议请求,下载 Web 页面,所述页面分析器分析页面并提取链接,然后所述信息采集器再以迭代的方式访问网络。
- [0016] 所述信息采集器采用图的遍历算法搜索 Web 页面。
- [0017] 所述 URL 过滤器利用扩展元数据的语义信息,对从 Web 页面中提取出的 URL 进行主题相关性预测,按照相关链接进行采集、不相关链接直接丢弃的原则进行剪枝处理。
- [0018] 所述 URL 管理器一方面从所述 URL 库中获得 URL 列表,进行任务排列后分配给多个信息采集器;另一方面从多个信息采集器中获得新的 URL 列表,将这些列表保存到所述 URL 库中。
- [0019] 一种网络爬虫系统实现页面渲染功能的方法,其特征在于包括如下步骤:
- [0020] (1) 生成 Web 页面的开始标签;
- [0021] (2) 渲染页面模板中的内容,其中每进入一个标签,都依次调用所述标签的各个生命周期阶段;
- [0022] (3) 渲染 Web 页面中的体;
- [0023] (4) 生成 Web 页面的结束标签;
- [0024] (5) 清除数据。
- [0025] 其中,所述步骤(2)中,调用所述标签的各个生命周期阶段是指从上层标签到下层标签的递归入口,只有下层标签渲染结束,进行调用的组件才继续后续阶段的操作。
- [0026] 所述步骤(4)中,生成结束标签的操作由控制内嵌标签执行流程的操作代替。
- [0027] 一种网络爬虫系统实现页面渲染功能的方法,其特征在于包括如下步骤:
- [0028] 当发现一个图片标签引用了一张图片时,向服务器发出请求;此时继续渲染后面的代码,服务器返回所述图片的文件,然后重新渲染这部分代码。
- [0029] 当发现存在一个 JavaScript 代码的<script>标签时,执行语句,重新渲染部分代码,然后将渲染的结果生成图片。

[0030] 本发明所提供的网络爬虫系统不仅能完成一般网络爬虫的功能,还可以将网页直接进行页面渲染,并将渲染结果直接用图片格式加以保存,从而为低成本、高效率地实现页面预览功能奠定技术基础。

## 附图说明

[0031] 下面结合附图和具体实施方式对本发明作进一步的详细说明。

[0032] 图 1 为本发明所提供的网络爬虫系统的整体组成示意图;

[0033] 图 2 为本网络爬虫系统实现网络爬虫基本功能的流程示意图;

[0034] 图 3 为本网络爬虫系统实现页面渲染功能的流程示意图。

## 具体实施方式

[0035] 如图 1 所示,本发明所提供的网络爬虫系统主要由以下各部分组成:

[0036] 1. 信息采集器

[0037] 每个信息采集器是一个网页蜘蛛 (Web Spider),处于网络爬虫系统的底层,是网络爬虫系统与海量的互联网信息(如论坛、博客、WAP、文档、音视频资料等)直接进行交互的接口部分。信息采集器的作用是获取 Web 页面。它通常从信息源(如用户查询、URL 列表或某一页面)出发,通过 http 协议请求,下载 Web 页面,页面分析器分析页面并提取链接,然后信息采集器再以迭代的方式访问网络。在本发明的一个具体实施例中,信息采集器优选采用图的遍历算法(如广度优先或深度优先策略)搜索 Web 页面。

[0038] 为保证高速获取 Web 页面中的信息,本网络爬虫系统在并行机制的基础上,对各个信息采集器采用多线程技术。在一般情况下,每个信息采集器能同时启动数百个线程进行页面信息采集。URL 管理器采取交织存取的方式管理待采集的 URL 队列,向各个信息采集器分配采集任务,因此可以保证同一个信息采集器最多只有一个线程连接同一个 Web 服务器,有效避免该 Web 服务器因访问量骤增而出现阻塞甚至宕机。

[0039] 2. 链接 (URL) 过滤器

[0040] 在 URL 库里存放的是从采集到的页面中提取出来的所有 URL,为避免采集页面出现“主题漂移”问题,这些 URL 在进入 URL 库前都必须经过主题相关性预测。我们利用扩展元数据(即 HTML Tag 如 Anchor 等信息)的语义信息,对从采集到的页面内提取出来的 URL 进行主题相关性预测,按照相关链接进行采集、不相关链接直接丢弃的原则进行剪枝处理,减少系统采集无关页面的数量,从而大量节省系统运行成本,有效提高主题信息搜索的速度和效率。链接过滤器将被预测为指向主题相关页面的链接 (URL) 入库存储,进而作为待采集 URL 由 URL 管理器分配给各个信息采集器采集该 URL 链接所指向的 Web 页面。

[0041] 3. 页面过滤器

[0042] 为进一步提高系统的查准率,需要对采集下来的页面进行主题相关性判断,也就是页面过滤。这实质上是一个文本主题分类的过程。通过去除相关性较小的页面(小于设定的阈值),提高系统的查准率。根据全信息理论,自然语言作为认识主体所表述的“事物运动状态及其变化方式”,包括形式、含义和其对认识主体的效用等三方面,分别称为事物的语法信息、语义信息和语用信息,而这三者的整体则称为“全信息”。自然语言文本具有词语同义性、词语多义性等特点,而 Web 文本是自然语言的一种特殊载体,因此在判断一篇文

本是否与系统的采集主题相关时,我们不但要关心文本的语法信息,还需要关心文本的语义准确性。本网络爬虫系统的页面过滤器以此为依据,吸收传统向量空间模型的思想,采用基于概念的向量空间法进行页面内容的过滤,通过将词汇映射到概念一级,从词所表达的概念意义层次也就是语义层次对文本进行相关性分析。

[0043] 4. 页面分析器

[0044] 页面分析器的主要功能是解析抓取下来的页面内容,可以分为两部分工作:一部分是解析出带有链接标记的 URL,交给 URL 过滤器解析,提取出链接;另一部分是将页面内容解析为文本格式,交给页面过滤器处理。

[0045] 5. URL 管理器

[0046] URL 管理器的主要功能是管理 URL 任务。一方面 URL 管理器从 URL 库中获得 URL 列表,并将它们进行任务排列后分配给多个信息采集器,另一方面 URL 管理器从多个信息采集器中获得新的 URL 列表,将这些列表以一定的策略保存到 URL 库中。

[0047] 如图 2 所示,上述的网络爬虫系统在实现网络爬虫的基本功能时,首先由 URL 管理器启动信息采集器开始 Web 页面的采集工作,并对采集的 Web 页面进行存储。然后由页面分析器进行分析,获得标记和页面两部分。其中的标记由送入 URL 过滤器进行解析,而页面部分送入页面过滤器,由页面过滤器进行内容冗余检测后,存入页面库中。Web 页面在由 URL 过滤器进行限定站点范围和主题的过滤之后,送入 URL 库中。此后,与 URL 库连接的图片生成器开始工作,针对 URL 库中存储的 URL 生成页面对应的图片。下面对此展开具体的说明。

[0048] 首先,用户输入网址向服务器发出请求,服务器返回 html 格式的 Web 页面;页面解析器开始载入 html 语言的源代码,如果发现 <head> 标签内有一个 <link> 标签引用外部 CSS 文件,则发出 CSS 文件的请求,服务器返回这个 CSS 文件;页面解析器继续载入 html 中 <body> 部分的代码,开始渲染页面。

[0049] 如图 3 所示,本网络爬虫系统实现页面渲染功能的具体步骤是这样的:

[0050] 1. 渲染准备阶段

[0051] 用于渲染前的准备操作,比如初始化一些数据;

[0052] 2. 生成开始标签

[0053] 用于生成一个 Html 文件的开始标签;

[0054] 3. 渲染模板

[0055] 该步骤主要用于渲染模板中的内容。这个阶段一般会有多个标签需要渲染,每进入一个标签,都会依次调用这个标签的各个生命周期阶段,也就是说,本处是一个从上层标签到下层标签的递归入口,只有下层标签渲染结束,进行调用的组件才会继续后续阶段的操作。

[0056] 4. 渲染体

[0057] 与渲染模板相似,也是渲染一段模板中的内容。比如对于 a 标签 (<a href = " pagelink" >this is body</a>),它的 body 是“this is body”这几个文字。

[0058] 5. 生成结束标签

[0059] 该步骤一般用于生成一个结束标签,或者控制内嵌标签的执行流程。

[0060] 6. 清除数据

[0061] 其它几个阶段并非经常用到,更多是保证生命周期的完整性。

[0062] 需要说明的是,当发现一个〈img〉标签引用了一张图片时,向服务器发出请求。此时不必等到图片下载完,而是继续渲染后面的代码;服务器返回图片文件。由于图片占用了一定面积,影响了后面段落的排布,因此需要回过头来重新渲染这部分代码;当发现存在一个 JavaScript 代码的〈script〉标签时,执行语句,重新渲染 JavaScript 执行中处理的那部分页面代码;然后由图片生成器将渲染的结果生成图片。

[0063] 上面以 html 格式的 Web 页面为例对本发明作了说明,但本发明所提供的具备页面渲染功能的网络爬虫系统并不限于处理 html 格式的页面,其它格式的 Web 页面也是可以直接处理的。

[0064] 利用本发明,当我们根据网页的地址进行检索后,不仅可以了解该页面的基本内容,更重要的是能够看到其基本的显示效果,从而更多地了解整个页面的内容。

[0065] 以上对本发明所提供的具备页面渲染功能的网络爬虫系统及其实现方法进行了详细的说明。对本领域的技术人员而言,在不背离本发明实质精神的前提下对它所做的任何显而易见的改动,都将构成对本发明专利权的侵犯,将承担相应的法律责任。

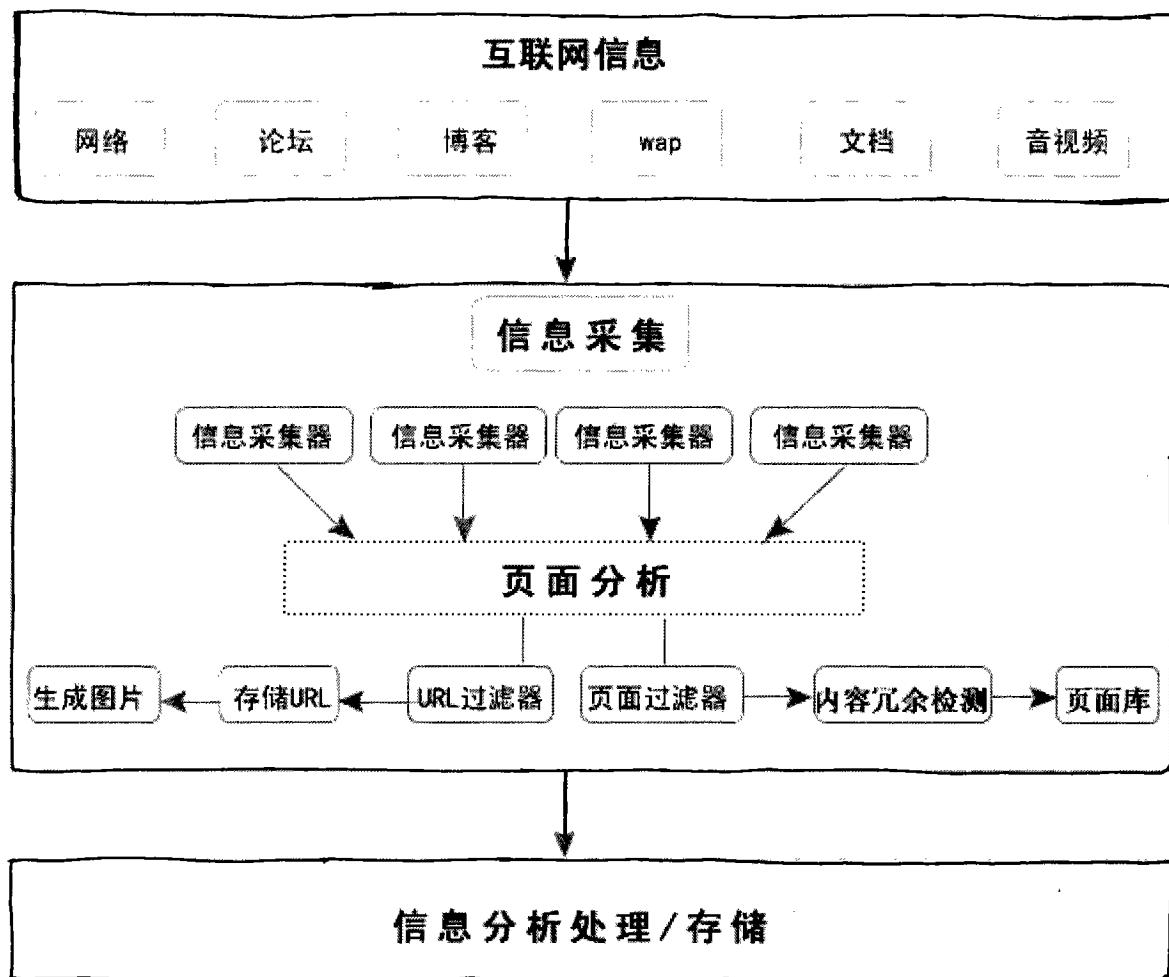


图 1

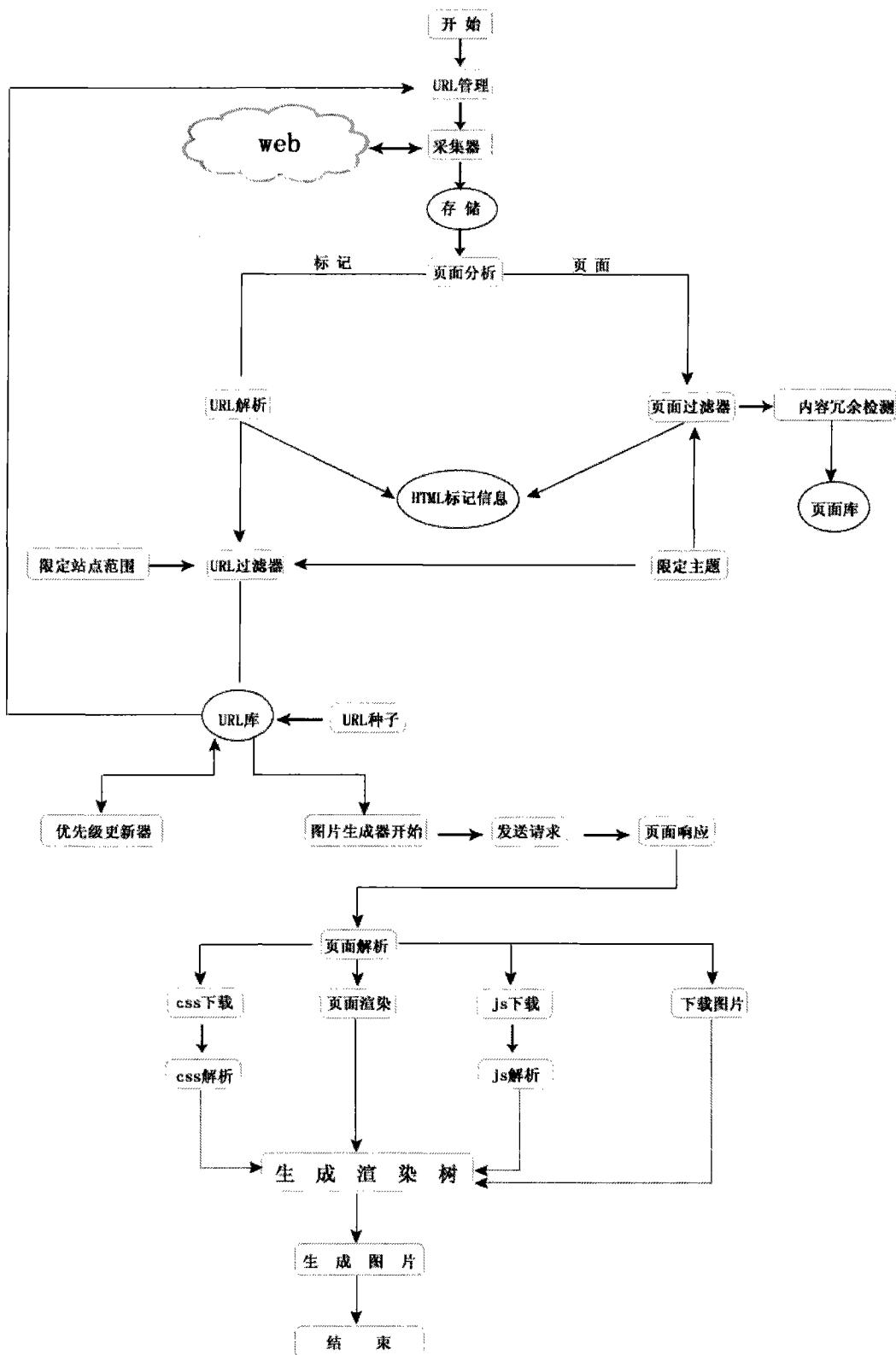


图 2

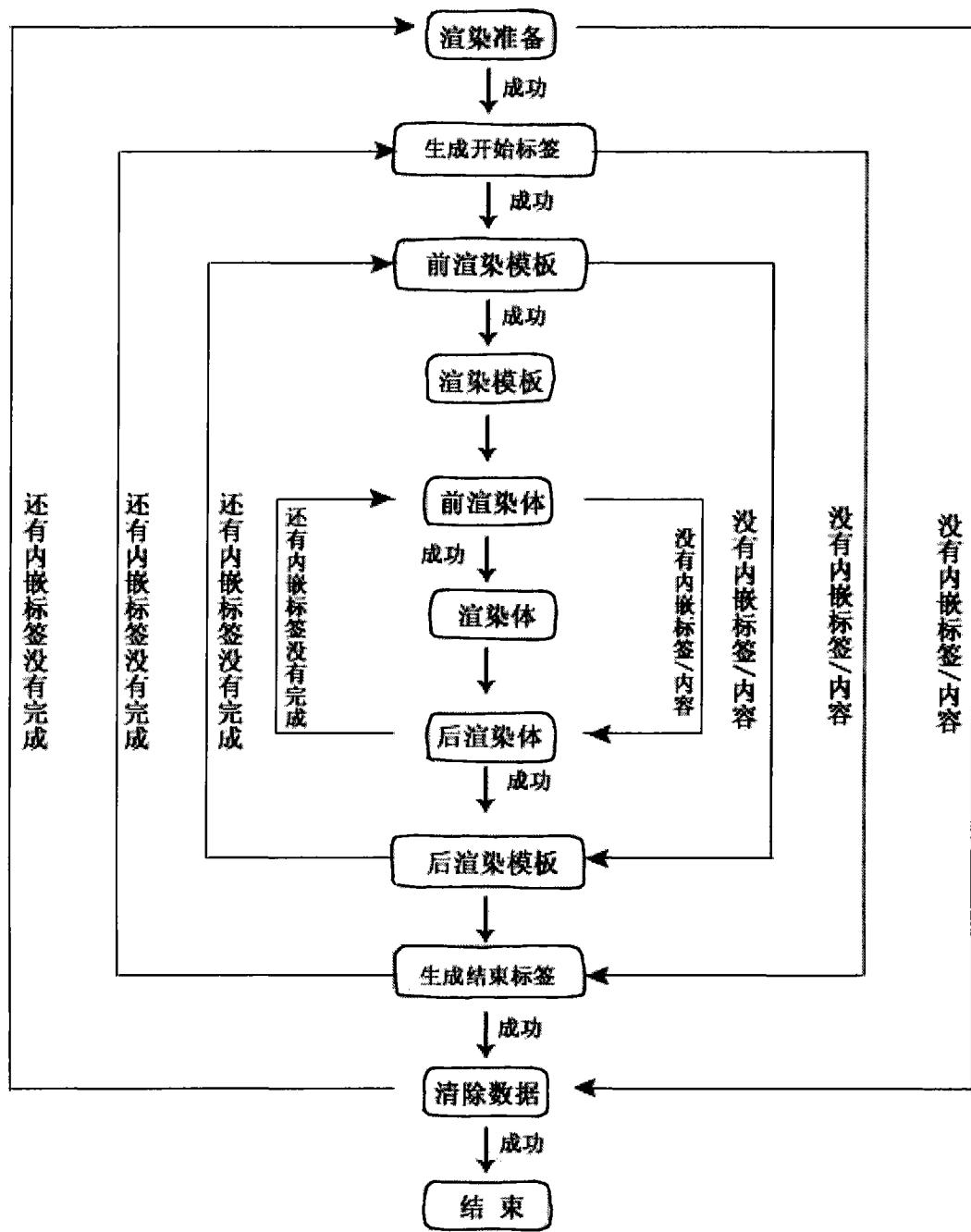


图 3