(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 118332174 B
(45)授权公告日 2024.10.29

(21)申请号 202410760744.7
(22)申请日 2024.06.13
(65)同一申请的已公布的文献号
申请公布号 CN 118332174 A
(43)申请公布日 2024.07.12
(73)专利权人 荣耀终端有限公司
地址 518040 广东省深圳市福田区香蜜湖街道东海社区红荔西路8089号深业中城6号楼A单元3401
(72)发明人 杨政良
(74)专利代理机构 深圳中一联合知识产权代理有限公司 44414
专利代理师 魏江黎
(51)Int.Cl.
G06F 16/951(2019.01)
G06F 16/958(2019.01)
(56)对比文件
CN 109213824 A,2019.01.15
CN 113934912 A,2022.01.14
审查员 邹玥

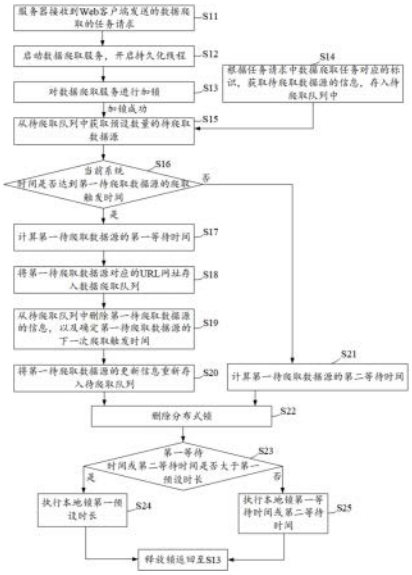
权利要求书3页 说明书19页 附图13页

(54)发明名称

数据爬取方法、系统和计算机可读存储介质

(57)摘要

本申请实施例提供了一种数据爬取方法、系统和计算机可读存储介质,该方法应用于数据爬取系统,数据爬取系统包括Web客户端和服务器,该方法包括:Web客户端显示第一页面,第一页面用于对数据爬取任务进行配置;Web客户端通过第一页面接收到对数据爬取任务的第一配置操作,以及将对数据爬取任务的配置内容发送至服务器存储,第一配置操作至少包括对数据爬取任务的爬取数据源和爬取时间间隔的配置;在Web客户端接收到启动数据爬取任务的操作的情况下,向服务器发送任务请求,请求服务器执行数据爬取任务;服务器根据爬取时间间隔,定时从爬取数据源对应的网址上爬取网页数据。由此,可以提高数据爬取的效率。



CN 118332174 B

1.一种数据爬取方法,其特征在于,所述方法应用于数据爬取系统,所述数据爬取系统包括Web客户端和服务器,所述方法包括:

所述Web客户端显示第二页面,所述第二页面用于对爬取数据源进行配置;

所述Web客户端通过所述第二页面接收到对所述爬取数据源的第二配置操作,以及将对所述爬取数据源的配置内容发送至所述服务器存储,所述第二配置操作至少包括对所述爬取数据源对应的网址的配置;

所述Web客户端显示第一页面,所述第一页面用于对数据爬取任务进行配置;

所述Web客户端通过所述第一页面接收到对所述数据爬取任务的第一配置操作,以及将对所述数据爬取任务的配置内容发送至所述服务器存储,所述第一配置操作至少包括对所述数据爬取任务的爬取数据源和爬取时间间隔的配置,对所述数据爬取任务的爬取数据源的配置包括:从所述第二配置操作所配置的爬取数据源中选择至少一个爬取数据源;

在所述Web客户端接收到启动所述数据爬取任务的操作的情况下,向所述服务器发送任务请求,请求所述服务器执行所述数据爬取任务,所述任务请求携带所述数据爬取任务的标识;

所述服务器根据所述数据爬取任务的标识,获取所述数据爬取任务对应的爬取数据源,以及将所述爬取数据源和爬取触发时间存入第一队列,其中,所述爬取触发时间为所述服务器根据当前系统时间和所述爬取时间间隔所确定的;

所述服务器从所述第一队列中获取所述爬取数据源,在当前系统时间达到所述爬取触发时间的情况下,根据所述爬取数据源对应的网址爬取网页数据;

从所述第一队列中删除所述爬取数据源的信息,以及根据所述爬取时间间隔确定所述爬取数据源的下一次爬取触发时间;

将所述爬取数据源和所述下一次爬取触发时间重新存入所述第一队列,以定时爬取所述网页数据;

在当前系统时间未达到所述爬取触发时间的情况下,所述服务器根据所述爬取触发时间和当前系统时间计算等待时间;

在所述等待时间大于第一预设时长的情况下,所述服务器执行本地锁所述第一预设时长,释放本地锁后重新从所述第一队列中获取爬取数据源;

在所述等待时间不大于第一预设时长的情况下,所述服务器执行本地锁所述等待时间,释放本地锁后重新从所述第一队列中获取爬取数据源。

2.根据权利要求1所述的方法,其特征在于,所述第一页面包括第一控件,所述Web客户端通过所述第一页面接收到对所述数据爬取任务的第一配置操作,包括:

所述Web客户端响应于用户对所述第一控件的点击操作,显示对所述数据爬取任务进行配置的第一输入控件;

接收所述用户在所述第一输入控件上输入的所述爬取数据源和所述爬取时间间隔,完成所述第一配置操作。

3.根据权利要求2所述的方法,其特征在于,在所述完成所述第一配置操作之后,所述方法还包括:

所述Web客户端在所述第一页面上显示所述数据爬取任务的配置内容和第二控件,所述第二控件用于触发所述数据爬取任务的启动和停止。

4.根据权利要求3所述的方法,其特征在于,所述Web客户端接收到启动所述数据爬取任务的操作,包括:

所述Web客户端接收到所述用户对所述第二控件的点击操作,触发启动所述数据爬取任务。

5.根据权利要求1所述的方法,其特征在于,所述第二页面包括第三控件,所述Web客户端通过所述第二页面接收到对所述爬取数据源的第二配置操作,包括:

所述Web客户端响应于用户对所述第三控件的点击操作,显示对所述爬取数据源进行配置的第二输入控件;

接收所述用户在所述第二输入控件上输入的所述爬取数据源对应的网址,完成所述第二配置操作。

6.根据权利要求5所述的方法,其特征在于,在所述完成所述第二配置操作之后,所述方法还包括:

所述Web客户端在所述第二页面上显示所述爬取数据源的配置内容和第四控件,所述第四控件用于触发对所述爬取数据源的爬取规则进行配置。

7.根据权利要求6所述的方法,其特征在于,在所述Web客户端在所述第二页面上显示所述爬取数据源的配置内容和第四控件之后,所述方法还包括:

所述Web客户端接收到所述用户对所述第四控件的点击操作,显示对所述爬取规则进行配置的第三页面;

接收所述用户对所述第三页面上第五控件的点击操作,获取并显示所述爬取数据源对应的网址的原始报文信息;

接收所述用户在所述第三页面上第三输入控件上输入的规则表达式,根据所述规则表达式对所述原始报文信息进行爬取测试。

8.根据权利要求1至7中任一项所述的方法,其特征在于,所述根据所述爬取数据源对应的网址爬取所述网页数据,包括:

所述服务器获取所述爬取数据源对应的爬取规则;

根据所述爬取规则对所述爬取数据源对应的网址中的内容进行爬取,得到所述网页数据。

9.根据权利要求8所述的方法,其特征在于,所述爬取规则包括对爬取结果所包含字段的必填性设置,在所述得到所述网页数据之后,所述方法还包括:

若所述网页数据中第一字段对应的必填性设置为必填,且所述第一字段的值为空,则所述服务器丢弃所述网页数据。

10.根据权利要求1至7中任一项所述的方法,其特征在于,在所述根据所述爬取数据源对应的网址爬取网页数据之后,所述方法还包括:

所述服务器将所述网页数据保存至存储介质中。

11.根据权利要求1至7中任一项所述的方法,其特征在于,所述数据爬取系统还包括电子设备,所述方法还包括:

所述电子设备向所述服务器发送数据请求,请求获取爬取到的所述网页数据;

所述电子设备接收并展示来自所述服务器的所述网页数据。

12.一种数据爬取系统,其特征在于,所述数据爬取系统包括Web客户端和服务器,所述

数据爬取系统用于执行如权利要求1至11中任一项所述的方法。

13.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质包括指令,当所述指令在数据爬取系统上运行时,使得所述数据爬取系统执行如权利要求1至11中任一项所述的方法。

数据爬取方法、系统和计算机可读存储介质

技术领域

[0001] 本申请涉及数据采集技术领域,具体涉及一种数据爬取方法、系统和计算机可读存储介质。

背景技术

[0002] 随着信息网络技术的高速发展,网络信息量呈爆炸式增长,但是在庞大的网络信息量中通常会有一些冗余信息或者用户并不关注的信息,因此,就需要从庞大的网络信息量中获取用户所需的信息。

[0003] 网络爬虫(或爬虫引擎)是一种自动浏览网络并抓取网页数据的程序,它可以根据一定的搜索策略从网络上过滤抓取到用户所需的信息。由于网页数据比较复杂,内容参差不齐,有时只需要抓取其中的小部分数据即可,那么,研发人员就要编写大量代码以完成网络爬虫的抓取过程。但是,这种编写代码的方式会耗费大量人力,数据爬取的效率比较低。

发明内容

[0004] 本申请提供了一种数据爬取方法、系统和计算机可读存储介质,可以提高数据爬取的效率。

[0005] 第一方面,本申请提供一种数据爬取方法,应用于数据爬取系统,数据爬取系统包括Web客户端和服务器,该方法包括:

[0006] 所述Web客户端显示第一页面,所述第一页面用于对数据爬取任务进行配置;

[0007] 所述Web客户端通过所述第一页面接收到对所述数据爬取任务的第一配置操作,以及将对所述数据爬取任务的配置内容发送至所述服务器存储,所述第一配置操作至少包括对所述数据爬取任务的爬取数据源和爬取时间间隔的配置;

[0008] 在所述Web客户端接收到启动所述数据爬取任务的操作的情况下,向所述服务器发送任务请求,请求所述服务器执行所述数据爬取任务;

[0009] 所述服务器根据所述爬取时间间隔,定时从所述爬取数据源对应的网址上爬取网页数据。

[0010] 其中,Web客户端和服务器之间可以进行通信,用户(如研发人员)可以在Web客户端上登录访问Web管理页面,并在该Web管理页面上进行数据爬取任务的配置(也即创建数据爬取任务)。在Web客户端上触发开始数据爬取任务之后,Web客户端便可以将该任务请求发送给服务器,服务器即可以在后台执行数据爬取任务,以及将数据爬取结果进行存储。

[0011] 在Web客户端上进行数据爬取任务的配置时,Web客户端可以显示第一页面(即数据爬取任务管理页面),在该页面上用户可以对数据爬取任务的爬取数据源、爬取时间间隔等信息进行配置,以及可以将配置内容发送给服务器进行存储,例如服务器将对数据爬取任务的配置内容存储在第一数据表中。在服务器进行数据爬取的过程中,可以根据配置的爬取时间间隔,定时持续的去爬取网页数据。

[0012] 在一个实现方式中,如果用户在Web客户端没有配置数据爬取任务的爬取时间间

隔,则服务器可以设置一个默认的爬取时间间隔,以根据该爬取时间间隔定时持续的去爬取网页数据。

[0013] 由此,上述数据爬取方法可以使用户在Web客户端上对数据爬取任务进行动态配置,减少编写代码的人力成本,同时服务器可以按照爬取时间间隔持续的进行数据爬取过程,提高了数据爬取的效率,也可以提高数据爬取过程的灵活性。

[0014] 结合第一方面,在第一方面的有些实现方式中,第一页面包括第一控件,上述Web客户端通过所述第一页面接收到对所述数据爬取任务的第一配置操作,包括:

[0015] 所述Web客户端响应于用户对所述第一控件的点击操作,显示对所述数据爬取任务进行配置的第一输入控件;

[0016] 接收所述用户在所述第一输入控件上输入的所述爬取数据源和所述爬取时间间隔,完成所述第一配置操作。

[0017] 其中,第一页面上可以包括第一控件(即新增控件),用户点击该第一控件后,Web客户端可以显示对数据爬取任务进行配置的第一输入控件,可以用于输入填写名称、编码、描述、爬取数据源、爬取时间间隔等内容,待填写了这些配置内容后,保存即可完成上述第一配置操作。由此,用户可以通过便捷的网页配置过程即可以完成数据爬取任务的配置,提高了数据爬取的效率。

[0018] 结合第一方面,在第一方面的有些实现方式中,在所述完成所述第一配置操作之后,所述方法还包括:

[0019] 所述Web客户端在所述第一页面上显示所述数据爬取任务的配置内容和第二控件,所述第二控件用于触发所述数据爬取任务的启动和停止。

[0020] 其中,在完成数据爬取任务的配置之后,Web客户端的上述第一页面上会显示已配置的数据爬取任务的记录,且该记录该对应有第二控件(即操作控件),用于触发任务开始和停止,也可以用于触发对数据爬取任务的编辑,即更改一些配置内容等。

[0021] 那么相应的,上述Web客户端接收到启动所述数据爬取任务的操作,包括:所述Web客户端接收到所述用户对所述第二控件的点击操作,触发启动所述数据爬取任务。

[0022] 由此,用户可以在Web客户端一键触发启动数据爬取任务,减少编写代码的人力成本,提高效率。

[0023] 结合第一方面,在第一方面的有些实现方式中,所述方法还包括:

[0024] 所述Web客户端显示第二页面,所述第二页面用于对所述爬取数据源进行配置;

[0025] 所述Web客户端通过所述第二页面接收到对所述爬取数据源的第二配置操作,以及将对所述爬取数据源的配置内容发送至所述服务器存储,所述第二配置操作至少包括对所述爬取数据源对应的网址的配置。

[0026] 因上述用户可以在Web客户端的第一页面上对数据爬取任务的爬取数据源进行配置,那么也就要提前对爬取数据源进行配置,以供后续选择。在Web客户端上进行爬取数据源(即统一资源定位符(uniform resource location,URL)数据源)的配置时,Web客户端可以显示第二页面(即URL管理页面),在该页面上用户可以对URL数据源的网址等信息进行配置,以及可以将配置内容发送给服务器进行存储,例如服务器将对URL数据源的配置内容存储在第二数据表中。

[0027] 由此,可以使用户在Web客户端上对爬取数据源进行动态配置,减少编写代码的人

力成本,提高数据提取的效率。

[0028] 结合第一方面,在第一方面的有些实现方式中,第二页面包括第三控件,所述Web客户端通过所述第二页面接收到对所述爬取数据源的第二配置操作,包括:

[0029] 所述Web客户端响应于用户对所述第三控件的点击操作,显示对所述爬取数据源进行配置的第二输入控件;

[0030] 接收所述用户在所述第二输入控件上输入的所述爬取数据源对应的网址,完成所述第二配置操作。

[0031] 其中,第二页面上可以包括第三控件(即新增控件),用户点击该第三控件后,Web客户端可以显示对爬取数据源进行配置的第二输入控件,可以用于输入填写名称、编码、描述、URL网址、所属类型等内容,待填写了这些配置内容后,保存即可完成上述第二配置操作。由此,用户可以通过便捷的网页配置过程即可以完成爬取数据源的配置,提高了数据爬取的效率。

[0032] 结合第一方面,在第一方面的有些实现方式中,在所述完成所述第二配置操作之后,所述方法还包括:

[0033] 所述Web客户端在所述第二页面上显示所述爬取数据源的配置内容和第四控件,所述第四控件用于触发对所述爬取数据源的爬取规则进行配置。

[0034] 其中,在完成爬取数据源的配置之后,Web客户端的上述第二页面上会显示已配置的爬取数据源的记录,且该记录该对应有第四控件(即操作控件),用于触发对爬取数据源进行编辑、删除、数据提取(包括爬取规则的配置)等操作。

[0035] 结合第一方面,在第一方面的有些实现方式中,在所述Web客户端在所述第二页面上显示所述爬取数据源的配置内容和第四控件之后,所述方法还包括:

[0036] 所述Web客户端接收到所述用户对所述第四控件的点击操作,显示对所述爬取规则进行配置的第三页面;

[0037] 接收所述用户对所述第三页面上第五控件的点击操作,获取并显示所述爬取数据源对应的网址的原始报文信息;

[0038] 接收所述用户在所述第三页面上第三输入控件上输入的规则表达式,根据所述规则表达式对所述原始报文信息进行爬取测试。

[0039] 其中,如果用户点击了第四控件中的数据提取,Web客户端会显示对爬取规则进行配置的第三页面,该页面上显示有第五控件(即原始报文获取控件),当用户点击该原始报文获取控件,Web客户端可以向服务器发送请求,请求服务器查询爬取数据源对应网址的原始报文,并将原始报文的数据进行显示。同时,第三页面上还显示有第三输入控件(即爬取规则输入框),用户可以在该第三输入控件上输入规则表达式,例如规则表达式可以为正则表达式或XML路径语言。待输入了规则表达式之后,可以点击爬取测试控件,即可以对根据所输入的规则表达式爬取原始报文进行测试,以测试所输入的规则表达式是否准确。由此,用户可以在Web客户端上便捷的配置或更新爬取规则,减少编写代码的人力成本。

[0040] 在一些实现方式中,Web客户端上完成了对爬取规则的配置之后,也可以将爬取规则等信息发送给服务器进行保存,服务器可以将对爬取数据源配置的爬取规则存储在第三数据表中。

[0041] 结合第一方面,在第一方面的有些实现方式中,所述任务请求携带所述数据爬取

任务的标识,在所述Web客户端向所述服务器发送任务请求之后,所述方法还包括:

[0042] 所述服务器根据所述数据爬取任务的标识,获取所述数据爬取任务对应的爬取数据源,以及将所述爬取数据源和爬取触发时间存入第一队列,其中,所述爬取触发时间为所述服务器根据当前系统时间和所述爬取时间间隔所确定的。

[0043] 其中,在上述对数据爬取任务配置完成之后,如果用户触发启动该数据爬取任务,Web客户端就会向服务器发送任务请求,该任务请求可以携带数据爬取任务对应的标识,用于使服务器根据该标识从上述第一数据表中查找对应的爬取数据源和爬取数据间隔。以及,服务器可以根据当前系统时间和爬取时间间隔计算出爬取触发时间,并将爬取数据源和爬取触发时间存入第一队列(即待爬取队列),以供后续从第一队列中读取爬取数据源进行数据爬取。

[0044] 可以理解,第一队列中可以存有不同数据爬取任务对应的爬取数据源,一个数据爬取任务对应的爬取数据源也可以有多个。

[0045] 结合第一方面,在第一方面的有些实现方式中,所述服务器根据所述爬取时间间隔,定时从所述爬取数据源对应的网址上爬取网页数据,包括:

[0046] 所述服务器从所述第一队列中获取所述爬取数据源,在当前系统时间达到所述爬取触发时间的情况下,根据所述爬取数据源对应的网址爬取所述网页数据;

[0047] 从所述第一队列中删除所述爬取数据源的信息,以及根据所述爬取时间间隔确定所述爬取数据源的下一次爬取触发时间;

[0048] 将所述爬取数据源和所述下一次爬取触发时间重新存入所述第一队列,以定时爬取所述网页数据。

[0049] 其中,在数据爬取的过程中,服务器可以先从上述第一队列中获取预设数量的爬取数据源(即作为待爬取数据源),该预设数量的爬取数据源包括上述配置的数据爬取任务对应的爬取数据源。如果当前系统时间达到了爬取数据源对应的爬取触发时间,服务器便可以根据该爬取数据源对应的网址来爬取网页数据。之后,服务器可以从第一队列中删除已爬取的爬取数据源,并计算下一次爬取触发时间,并将爬取数据源和下一次爬取触发时间重新存入第一队列,以实现定时持续的爬取网页数据,提高数据爬取效率。

[0050] 在一些实现方式中,服务器可以将所爬取的网页数据保存至存储介质中,以供后续需要时进行读取。

[0051] 结合第一方面,在第一方面的有些实现方式中,所述爬取规则包括对爬取结果所包含字段的必填性设置,在所述得到所述网页数据之后,所述方法还包括:

[0052] 若所述网页数据中第一字段对应的必填性设置为必填,且所述第一字段的值为空,则所述服务器丢弃所述网页数据。

[0053] 也即是说,用户在上述Web客户端对爬取规则进行配置时,还可以对爬取结果中包含的字段进行必填性设置,例如设置某个字段是必填项,那么,如果爬取到的网页数据中,第一字段(即任一个字段)对应的必填性设置是必填的,但是该第一字段的值又是空值,说明该第一字段不满足必填性设置的要求,则服务器可以将该网页数据进行丢弃,以使所爬取的网页数据符合用户需求。

[0054] 结合第一方面,在第一方面的有些实现方式中,所述根据所述爬取数据源对应的网址爬取所述网页数据,包括:

[0055] 所述服务器获取所述爬取数据源对应的爬取规则；

[0056] 根据所述爬取规则对所述爬取数据源对应的网址中的内容进行爬取，得到所述网页数据。

[0057] 其中，因上述服务器将爬取数据源对应的爬取规则存储在了第三数据表，因此，可以从第三数据表中获取爬取数据源对应的爬取规则，以根据该爬取规则进行数据爬取，得到上述爬取的网页数据。

[0058] 结合第一方面，在第一方面的有些实现方式中，所述方法还包括：

[0059] 在当前系统时间未达到所述爬取触发时间的情况下，所述服务器根据所述爬取触发时间和当前系统时间计算等待时间；

[0060] 在所述等待时间大于第一预设时长的情况下，所述服务器执行本地锁所述第一预设时长，释放本地锁后重新从所述第一队列中获取爬取数据源；

[0061] 在所述等待时间不大于第一预设时长的情况下，所述服务器执行本地锁所述等待时间，释放本地锁后重新从所述第一队列中获取爬取数据源。

[0062] 其中，因上述服务器从第一队列中获取了爬取数据源时，当前系统时间可能还没有达到爬取触发时间，那么就需要进行等待，以等待到系统时间达到爬取触发时间。但是又考虑到可能在等待时间内有新的数据爬取任务申请，服务器如果处于等待状态则无法执行新的数据爬取任务。因此，服务器可以设置一个第一预设时长，如果计算的等待时间大于该第一预设时长，则服务器执行本地锁第一预设时长，避免长时间加锁带来资源浪费，如果计算的等待时间不大于第一预设时长，则服务器执行本地锁等待时间，以尽快执行下一轮爬取任务。

[0063] 结合第一方面，在第一方面的有些实现方式中，所述数据爬取系统还包括电子设备，所述方法还包括：

[0064] 所述电子设备向所述服务器发送数据请求，请求获取爬取到的所述网页数据；

[0065] 所述电子设备接收并展示来自所述服务器的所述网页数据。

[0066] 其中，如果用户所持有的电子设备上安装有与上述数据爬取任务对应的业务应用，则该业务应用可以展示数据爬取结果，以及可以响应用户的搜索请求。那么，电子设备可以向服务器请求已存储的数据爬取结果(即上述网页数据)，并展示给用户查看，或者电子设备接收用户的搜索请求，从数据爬取结果中搜索到相应的内容并展示。由此，服务器可以及时将爬取到的网页数据发送给电子设备，提高用户的使用感知。

[0067] 第二方面，本申请提供一种数据爬取系统，数据爬取系统包括Web客户端和服务器，该数据爬取系统用于执行上述第一方面的技术方案中任意一种方法。

[0068] 第三方面，本申请提供一种装置，该装置包含在数据爬取系统中，该装置具有实现上述第一方面及上述第一方面的可能实现方式的功能。功能可以通过硬件实现，也可以通过硬件执行相应的软件实现。硬件或软件包括一个或多个与上述功能相对应的模块或单元。例如，接收模块或单元、处理模块或单元等。

[0069] 第四方面，本申请提供一种Web客户端，包括：一个或多个处理器，以及存储器；

[0070] 所述存储器与所述一个或多个处理器耦合，所述存储器用于存储计算机程序代码，所述计算机程序代码包括计算机指令，所述一个或多个处理器调用所述计算机指令以使得Web客户端执行第一方面的技术方案中的对应过程。

[0071] 第五方面,本申请提供一种服务器,包括:一个或多个处理器,以及存储器;

[0072] 所述存储器与所述一个或多个处理器耦合,所述存储器用于存储计算机程序代码,所述计算机程序代码包括计算机指令,所述一个或多个处理器调用所述计算机指令以使得服务器执行第一方面的技术方案中的对应过程。

[0073] 第六方面,本申请提供一种计算机可读存储介质,计算机可读存储介质中包括指令,当所述指令在数据爬取系统上运行时,使得所述数据爬取系统执行第一方面的技术方案中任意一种方法。

[0074] 第七方面,本申请提供一种计算机程序产品,计算机程序产品包括:计算机程序代码,当计算机程序代码在数据爬取系统上运行时,使得该数据爬取系统执行第一方面的技术方案中任意一种方法。

附图说明

[0075] 图1是本申请实施例提供的一例数据爬取方法的系统架构示意图;

[0076] 图2是本申请实施例提供的一例Web客户端上Web管理页面示意图;

[0077] 图3是本申请实施例提供的一例Web客户端上数据爬取任务管理页面示意图;

[0078] 图4是本申请实施例提供的另一例Web客户端上数据爬取任务管理页面示意图;

[0079] 图5是本申请实施例提供的又一例Web客户端上数据爬取任务管理页面示意图;

[0080] 图6是本申请实施例提供的又一例Web客户端上数据爬取任务管理页面示意图;

[0081] 图7是本申请实施例提供的一例Web客户端上URL管理页面示意图;

[0082] 图8是本申请实施例提供的另一例Web客户端上URL管理页面示意图;

[0083] 图9是本申请实施例提供的又一例Web客户端上URL管理页面示意图;

[0084] 图10是本申请实施例提供的又一例Web客户端上URL管理页面示意图;

[0085] 图11是本申请实施例提供的一例数据爬取方法的流程示意图;

[0086] 图12是本申请实施例提供的另一例数据爬取方法的流程示意图;

[0087] 图13是本申请实施例提供的一例Web客户端的结构示意图;

[0088] 图14是本申请实施例提供的一例服务器的结构示意图;

[0089] 图15是本申请实施例提供的一例电子设备的结构示意图。

具体实施方式

[0090] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行描述。其中,在本申请实施例的描述中,除非另有说明,“/”表示或的意思,例如,A/B可以表示A或B;本文中的“和/或”仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。另外,在本申请实施例的描述中,“多个”是指两个或多于两个。

[0091] 以下,术语“第一”、“第二”、“第三”仅用于描述目的,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”、“第三”的特征可以明示或者隐含地包括一个或者更多个该特征。

[0092] 在当前网络信息量呈爆炸式增长的环境下,用户要从庞大的网络信息量中获取到自己需要的信息是一件困难的事情,为此提出了搜索引擎的概念。搜索引擎是一个用户访

问万维网的入口,可以辅助用户搜索需要的信息,但是,考虑到不同领域、不同背景的用户往往具有不同的检索目的和需求,通过搜索引擎所返回的结果可能也会包含用户不关心的网页内容。为了解决该问题,定向抓取网页数据的网络爬虫应运而生,它可以自动下载网页数据,然后根据一定的搜索策略从网页数据中过滤抓取到用户所需的信息,具有较广泛的应用。

[0093] 在数据爬取的过程中,由于网页数据比较复杂,内容参差不齐,有时只需要抓取其中的小部分数据即可,比如,在爬取用户关注度较高的热榜内容时,只需要抓取热榜内容的标题、作者、热度值等数据,不需要抓取其他数据。因此,研发人员需要编写代码,对齐数据爬取时使用的正则表达式,并且在不同用户需求的情况下还需要更改代码,明显的,这种编写代码的方式会耗费大量人力,以及代码编写完成后的发布周期也比较长,导致数据爬取的效率比较低。

[0094] 有鉴于此,本申请实施例提供一种数据爬取方法,可以使研发人员在Web管理页面对数据爬取过程进行动态配置,减少编写代码的人力成本,以及动态配置过程可以提高数据爬取的灵活性,同时,该Web管理页面不需要进行代码发布的过程,只要能访问到该页面就可以创建数据爬取任务,提高了数据爬取的效率。

[0095] 下面我们先介绍一下本申请中数据爬取方法的应用场景,在一些实施例中,该数据爬取方法可以应用在如图1所示的系统架构上,该系统架构可以包括Web客户端、服务器和电子设备,Web客户端与服务器之间可以进行通信,电子设备与服务器之间也可以进行通信。

[0096] 其中,研发人员可以在Web客户端上登录访问Web管理页面,Web管理页面上提供有爬虫管理功能,在爬虫管理功能下提供有数据爬取任务管理、统一资源定位符(uniform resource location,URL)管理等功能。数据爬取任务管理功能可以用于创建数据爬取任务,并控制数据爬取任务的开始与停止,以及可以设置爬取时间间隔,以定时持续的进行数据爬取。URL管理功能可以用于创建URL数据源,以及对URL数据源进行配置,这里所创建的URL数据源可以供创建数据爬取任务时所选择,以选择确定要对哪个数据源进行爬取(也可以理解为对哪个URL网址进行爬取)。在Web客户端上触发开始数据爬取任务之后,Web客户端便可以将该任务请求发送给服务器,服务器即可以在后台执行数据爬取任务,以及将数据爬取结果进行存储。在一些实现方式中,数据爬取任务可以持续执行,并不断更新已存储的数据爬取结果。

[0097] 对于上述的电子设备(例如可以为手机),如果该电子设备上安装有与上述数据爬取任务对应的业务应用,该业务应用可以展示数据爬取结果以及可以响应用户的搜索请求。例如业务应用可以为智慧搜索APP,那么,电子设备可以向服务器请求已存储的数据爬取结果,并展示给用户查看,或者电子设备接收用户的搜索请求,从数据爬取结果中搜索到相应的内容并展示。在一些实现方式中,在用户通过业务应用执行了刷新操作的情况下,电子设备可以向服务器请求最新的数据爬取结果。

[0098] 因此,在上述图1的系统架构的基础上,研发人员可以在Web管理页面对数据爬取过程进行动态配置,相较于传统的编写代码的方式来说比较方便快捷。

[0099] 在另一些场景中,上述系统架构还可以用于对一些接口数据的定时获取,例如通过对接口地址的配置和获取内容的配置,可以自动获取到对应的接口数据。

[0100] 接下来,我们再介绍一下在Web客户端上对数据爬取过程进行配置的过程。以Web客户端是个人计算机(personal computer,PC)为例,如图2所示,Web客户端上可以登录访问到Web管理页面,例如Web管理页面可以为智慧搜索运营云页面。在智慧搜索运营云页面上,左边任务栏上呈现有爬虫管理控件21,点击该爬虫管理控件21后,控件下方会展示出子功能控件,包括但不限于数据爬取任务管理控件22和URL管理控件23,点击一个控件后智慧搜索运营云页面的右边部分便可以显示出相应的配置页面。

[0101] 示例性地,在研发人员点击数据爬取任务管理控件22的情况下,Web客户端上呈现有如图3所示的数据爬取任务管理页面,该页面上提供有不限于查询和新增的功能。对于查询功能,即可以基于数据爬取任务的名称和/或编码来查询已创建的数据爬取任务,这里的名称可以是创建数据爬取任务时研发人员所定义的任务名称(例如名称是小说榜),编码可以是创建数据爬取任务时研发人员所定义的编码(例如小说榜对应的编码是novel)。以图3所示为例,在名称输入框31中输入“小说榜”后点击查询控件32,当前页面上可以显示出已创建的小说榜的数据爬取任务。对于新增功能,即可以新增创建一个数据爬取任务,以图4所示为例,点击新增控件41,当前页面上可以显示出新增数据爬取任务时需要填写的内容,研发人员填写名称、编码、描述、爬取数据源、爬取时间间隔等内容之后,点击保存控件42,就可以创建成功该数据爬取任务。其中,这里的爬取时间间隔为定时数据爬取的一种方式,即在第一次数据爬取完成之后,可以在设置的时间间隔之后进行第二次数据爬取,以此类推,进行持久性的定时数据爬取。

[0102] 对于在新增数据爬取任务时填写爬取数据源的过程,如图5所示,填写爬取数据源的位置可以以一种选择控件51形式展现,当点击选择控件51,当前页面上可以显示出已创建的URL数据源列表,然后研发人员可以从该列表中选择要爬取的URL数据源。例如图5中选择了要爬取的URL数据源为热榜A和热榜B。

[0103] 在创建成功一个数据爬取任务之后,所创建的数据爬取任务便可以显示在数据爬取任务管理页面上,以所创建的数据爬取任务是对社会热榜的爬取任务为例,如图6所示,该条数据爬取任务的记录还对应有操作控件61,操作控件61用于触发任务开始、停止和编辑等不同的操作。如果研发人员点击了任务开始,即是触发该条数据爬取任务开始执行,Web客户端会向服务器发送任务请求,以由服务器执行对应的数据爬取任务。在任务开始之后,如果研发人员又点击了停止,则Web客户端会向服务器发送停止请求,以触发该数据爬取任务停止执行。如果研发人员点击了编辑,则会重新显示上述图4的页面,可以供研发人员更改名称、编码、描述、爬取数据源、爬取时间间隔等内容。

[0104] 在一些实现方式中,Web客户端在创建成功数据爬取任务之后,还可以生成该数据爬取任务对应的唯一标识,以及将该数据爬取任务对应的标识、名称、编码、描述、爬取数据源、爬取时间间隔等信息发送给服务器进行存储。可选地,Web客户端也可以将数据爬取任务的名称作为其对应的唯一标识。可选地,服务器可以将接收到的数据爬取任务对应的信息存储在第一数据表中。

[0105] 由上述图5的描述可知,在新增数据爬取任务时需要选择爬取数据源(即URL数据源),那么就要提前对爬取数据源进行配置,以供后续选择。对爬取数据源进行配置的过程可以通过上述图2中URL管理控件23对应的页面来实现,示例性地,在研发人员点击URL管理控件23的情况下,Web客户端上呈现有如图7所示的URL管理页面,该页面上提供有不限于查

询和新增的功能。对于查询功能,即可以基于URL数据源的名称和/或URL网址来查询已创建的URL数据源,这里的名称可以是创建URL数据源时研发人员所定义的URL名称(例如名称是热榜A),URL网址可以是创建URL数据源时研发人员所输入的对应网址(例如热榜A对应的URL网址是https://rebang.aaa)。以图7所示为例,在名称输入框71中输入“热榜A”后点击查询控件72,当前页面上可以显示出已创建的热榜A的URL数据源信息。对于新增功能,即可以新增创建一个URL数据源,以图8所示为例,点击新增控件81,当前页面上可以显示出新增URL数据源时需要填写的内容,研发人员填写名称、编码、描述、URL网址、所属类型等内容之后,点击保存控件82,就可以创建成功该URL数据源。

[0106] 在一些实现方式中,Web客户端在创建成功URL数据源之后,还可以生成该URL数据源对应的唯一标识,以及将该URL数据源对应的标识、名称、编码、描述、URL网址、所属类型等信息发送给服务器进行存储。可选地,Web客户端也可以将URL数据源的名称作为其对应的唯一标识。可选地,服务器可以将接收到的URL数据源对应的信息存储在第二数据表中。

[0107] 在创建成功一个URL数据源之后,所创建的URL数据源便可以显示在URL管理页面上,以所创建的URL数据源是对热榜B的数据源为例,如图9所示,该条URL数据源的记录还对应有操作控件91,操作控件91用于触发编辑、删除、数据提取和爬取数据查询等不同的操作。如果研发人员点击了编辑,则Web客户端会重新显示上述图8的页面,可以供研发人员更改名称、编码、描述、URL网址、所属类型等内容。如果研发人员点击了删除,则可以触发Web客户端将该条URL数据源进行删除。如果点击数据提取,则可以对该条URL数据源进行爬取规则的配置,因为如果上述数据爬取任务要对该条URL数据源进行爬取,通常是只需要抓取对应URL网址中的部分内容,因此就需要配置爬取规则,以抓取到需要的内容。

[0108] 其中,对URL数据源进行爬取规则配置的过程可以如图10所示,即研发人员在上述图9中点击操作控件91中的数据提取后,Web客户端可以跳转显示图10所示的页面。在该页面上,显示有URL数据源的原始来源、爬虫协议限制以及对应的URL网址,同时还显示有原始报文获取控件92。当点击该原始报文获取控件92,Web客户端可以向服务器发送请求,请求服务器查询URL数据源对应网址的原始报文,并将原始报文的数据显示在报文信息显示框93内;可以理解,这里显示的报文信息是URL数据源对应网址的全部内容,在配置了爬取规则之后,便可以从报文信息中抓取需要的内容。

[0109] 在获取到URL数据源对应网址的原始报文之后,研发人员可以在爬取规则输入框94中输入规则表达式,在一些实现方式中,规则表达式可以为正则表达式(regex),其包括普通字符(例如,a到z之间的字母)和特殊字符(称为“元字符”)的组合,在另一些实现方式中,规则表达式可以为XML路径语言(XPath),其是一种确定XML文档中某部分位置的语言,研发人员可以根据需要选择其中一种规则表达式。示例性地,图10以输入的正则表达式为例进行示出,待输入了规则表达式之后,可以点击爬取测试控件95,即可以对根据所输入的规则表达式爬取原始报文进行测试,以测试所输入的规则表达式是否准确。其中,爬取测试后抓取到的信息可以显示在爬取后报文信息显示框96内,例如图10所示的,所爬取的结果包括字段名、字段值和是否必填选项,字段名(例如title)可以由Web客户端根据规则表达式自动填写,也可以由研发人员手动填写,字段值为所爬取的原始报文中各条信息的title信息,是否必填选项表征该字段值是否必填,如果选择了“是”,则表征该字段值是必填的,假如在后续的数据爬取过程中该字段值出现空的情况,可以丢弃对应的数据爬取结果。在

爬取测试结果无异常的情况下,研发人员可以点击保存控件97,则Web客户端可以将爬取规则(包括规则表达式和对是否必填选项的设置结果等设置的规则)和爬取测试结果发送给服务器进行保存,可以理解,这里所保存的内容和上述URL数据源的标识具有对应关系,即通过URL数据源的标识可以查找到其对应的爬取规则,以供后续进行数据爬取。可选地,服务器可以将对URL数据源配置的爬取规则存储在第三数据表中,即第三数据表中存储有配置的规则表达式和对是否必填选项的设置结果。还可以理解,因URL数据源对应网址的原始报文的格式和数据可能会发生变化,因此,可以定时调整上述规则表达式,以爬取到准确的报文信息。

[0110] 通过上述的配置过程,即可以完成对URL数据源以及数据爬取任务的配置,那么,如果研发人员点击上述图6中操作控件61的任务开始,则Web客户端会向服务器发送任务请求,以由服务器执行对应的数据爬取任务。

[0111] 再接下来,我们将介绍一下服务器执行数据爬取任务的详细过程。如图11所示,该过程可以包括:

[0112] S11,服务器接收到Web客户端发送的数据爬取的任务请求。

[0113] 其中,任务请求中可以携带数据爬取任务对应的标识,例如可以是任务ID、数据爬取任务的名称等标识。

[0114] S12,服务器启动数据爬取服务,开启持久化线程。

[0115] 因上述在创建数据爬取任务时设置了爬取时间间隔,即是需要持续进行数据爬取,那么,服务器就需要开启一个持久化线程,用于后续持续的从待爬取队列中获取待爬取数据源。

[0116] 在一些实现方式中,服务器可以使用@conponet和@postconstruct注解启动持久化线程,以及通过while(true)判断逻辑使持久化线程进行循环。

[0117] S13,服务器对数据爬取服务进行加锁,如果加锁成功则执行S15。

[0118] 其中,因在实际场景中所部署的服务器可能有多个,不同服务器可以用于执行不同的服务进程,那么,如果要使当前服务器中的数据爬取服务执行当前的数据爬取任务,避免其他服务器或其他服务重复执行当前的数据爬取任务,就可以对当前的数据爬取服务进行加锁。

[0119] 在一些实现方式中,服务器可以对数据爬取服务分别添加本地锁和分布式锁,本地锁用于避免其他服务对数据爬取任务对应的数据进行并发读写,从而产生数据不一致的后果,分布式锁用于避免多个服务重复执行当前的数据爬取任务。可选地,服务器可以使用setnx命令获取分布式锁,以及设置锁的过期时间为N,例如N为1分钟,这里设置过期时间是为防止死锁,即如果当前服务器出现异常但是一直持锁的话,其他服务器也不会执行相应的数据爬取服务,导致数据爬取任务无法执行,而设置过期时间的话服务器可以在N时长之后进行解锁,使数据爬取任务继续执行。

[0120] S14,服务器根据任务请求中数据爬取任务对应的标识,获取待爬取数据源的信息,存入待爬取队列中。

[0121] 需要说明的是,S14获取待爬取的数据源信息的过程可以与上述S12-S13启动数据爬取服务等过程同步进行,即服务器可以同时进行数据爬取任务的多项准备工作。

[0122] 在S14中,因服务器接收到的任务请求中携带有数据爬取任务对应的标识,且Web

客户端在创建成功数据爬取任务之后,已将数据爬取任务对应的标识、名称、编码、描述、爬取数据源、爬取时间间隔等信息发送给服务器进行存储,那么,服务器便可以根据数据爬取任务对应的标识查找到对应的爬取数据源(即作为待爬取数据源)。又因Web客户端在创建成功URL数据源之后,也已将URL数据源对应的标识、名称、编码、描述、URL网址、所属类型等信息发送给服务器进行存储,那么服务器可以查找到数据爬取任务对应的爬取数据源的标识、URL网址等信息。示例性地,服务器可以从上述第一数据表中根据数据爬取任务的标识查找到对应的爬取数据源(即URL数据源),再从上述第二数据表中根据URL数据源的标识查找到对应的URL网址,由此可以将数据爬取任务、URL数据源和URL网址等信息对应起来。因此,服务器可以根据任务请求中数据爬取任务对应的标识,获取待爬取数据源的信息,待爬取数据源的信息包括但不限于数据源的标识、URL网址以及爬取触发时间等信息,以及将待爬取数据源的这些信息存入待爬取队列中。例如,数据爬取任务是对上述图6中社会热榜的爬取任务,其对应的待爬取数据源即热榜A和热榜B两个数据源,服务器可以将热榜A和热榜B两个数据源对应的信息存入待爬取队列中。

[0123] 其中,这里的爬取触发时间可以为系统时间+爬取时间间隔所确定的时间。示例性地,服务器接收到任务请求时的系统时间为10:00:00,爬取时间间隔为2分钟,则对应数据爬取任务的爬取触发时间应为10:02:00,服务器后续便可以按照该爬取触发时间进行数据爬取。

[0124] 在一些实现方式中,待爬取队列可以为redis的zset队列,zset队列是redis数据库中的一种有序队列,服务器在向zset队列存入待爬取数据源的信息时,可以将数据源的标识作为value,爬取触发时间作为score,按照爬取触发时间进行排序存入。例如可以按照爬取触发时间从早到晚的顺序将待爬取数据源的信息存入zset队列。

[0125] 可以理解,因研发人员可以在Web客户端于不同时间触发启动多个不同的数据爬取任务,Web客户端就要向服务器分别发送多个任务请求,以请求启动不同的数据爬取任务,那么,待爬取队列中会存入不同数据爬取任务对应的待爬取数据源的信息,且一个数据爬取任务对应的待爬取数据源也可以有多个。示例性地,Web客户端启动了3个数据爬取任务,第1个数据爬取任务对应有1个待爬取数据源(A),爬取触发时间为10:00:00,第2个数据爬取任务对应有2个待爬取数据源(B和C),爬取触发时间为10:05:00,第3个数据爬取任务对应有2个待爬取数据源(D和E),爬取触发时间为10:10:00,则待爬取数据源A、B、C、D、E的信息都会存入待爬取队列中。

[0126] S15,服务器从待爬取队列中获取预设数量的待爬取数据源。

[0127] 因服务器开启的持久化线程会持续的进行数据爬取,待爬取队列中也会存入越来越多的待爬取数据源,如果同时对全部待爬取数据源进行爬取的话无疑会增加服务器的处理量,导致负载过高,因此,服务器可以每次从待爬取队列中获取预设数量的待爬取数据源进行爬取。例如,预设数量可以为20条。

[0128] 在一些实现方式中,在待爬取队列为zset队列的情况下,服务器可以每次获取zset队列中前20条的待爬取数据源,即按照爬取触发时间从早到晚排序后的前20条待爬取数据源。

[0129] S16,服务器确定当前系统时间是否达到第一待爬取数据源的爬取触发时间,若达到则执行S17,若未达到则执行S21。

[0130] 其中,第一待爬取数据源为上述预设数量的待爬取数据源中的任意一个数据源。

[0131] 由上述描述可知,每个待爬取数据源都对应有爬取触发时间,需要达到对应的爬取触发时间时,服务器才会开始爬取对应的数据源,因此,服务器会判断当前系统时间是否达到第一待爬取数据源的爬取触发时间。

[0132] 在一些实现方式中,服务器可以使用当前系统时间减去第一待爬取数据源的爬取触发时间,得到时间差,如果该时间差大于或者等于0,说明当前系统时间已达到第一待爬取数据源的爬取触发时间,服务器需要开始执行数据爬取过程了。如果该时间差小于0,说明当前系统时间还未达到第一待爬取数据源的爬取触发时间,服务器可以继续等待。

[0133] 可以理解,因服务器获取的是预设数量的待爬取数据源,那么,就需要对每条待爬取数据源都判断是否达到待爬取数据源的爬取触发时间,对于达到爬取触发时间的多条数据源来说可以执行S17步骤,对于未达到爬取触发时间的多条数据源来说可以执行S21步骤。

[0134] S17,服务器计算第一待爬取数据源的第一等待时间。

[0135] 其中,此时的第一待爬取数据源为达到爬取触发时间的待爬取数据源,第一等待时间waittime可以使用爬取触发时间-当前系统时间来得到,所计算的第一等待时间用于后续控制本地锁的时间。

[0136] 可以理解,因当前系统时间已达到第一待爬取数据源的爬取触发时间,则第一等待时间waittime会小于或者等于0,后续可以使用waittime的绝对值来控制本地锁的时间。

[0137] S18,服务器将第一待爬取数据源对应的URL网址存入数据爬取队列,以供爬虫引擎进行数据爬取。

[0138] 即,对于达到爬取触发时间的每条第一待爬取数据源,服务器可以先将这些数据源对应的URL网址存入数据爬取队列,使爬虫引擎从该数据爬取队列中依次读取URL网址进行数据爬取。

[0139] 对于爬虫引擎进行数据爬取的过程,详见下述图12的描述,在此暂不赘述。

[0140] S19,服务器从待爬取队列中删除第一待爬取数据源的信息,以及确定第一待爬取数据源的下一次爬取触发时间。

[0141] S20,服务器将第一待爬取数据源的更新信息重新存入待爬取队列,之后执行S22。

[0142] 因上述服务器已将第一待爬取数据源对应的URL网址存入数据爬取队列,后续爬虫引擎会执行此次数据爬取任务,那么服务器便可以将第一待爬取数据源此次的信息删除,待确定了下一次爬取时间后,再将第一待爬取数据源的更新信息重新存入待爬取队列,以供下一次的爬取过程,由此可实现持续的自动定时数据爬取过程。可以理解,第一待爬取数据源的更新信息包括但不限于数据源的标识、URL网址以及下一次爬取触发时间等信息。

[0143] 在一些实现方式中,服务器确定第一待爬取数据源的下一次爬取触发时间的方式可以包括:获取第一待爬取数据源对应的数据爬取任务的爬取时间间隔,再用当前系统时间+对应的爬取时间间隔,即是下一次爬取触发时间。因此,上述持续的数据爬取过程也可理解是一种定时的数据爬取过程,定时时长为爬取时间间隔。

[0144] 还可以理解,对于每一个第一待爬取数据源,服务器都会执行上述S18-S20的步骤。

[0145] 在一些实现方式中,如果对于上述数据爬取任务,在Web客户端上未被配置爬取时

间间隔的话,服务器也可以设置一个默认的爬取时间间隔,从而以设置的爬取时间间隔来定时,持续的进行数据爬取过程。

[0146] S21,服务器计算第一待爬取数据源的第二等待时间。

[0147] 其中,此时的第一待爬取数据源为未达到爬取触发时间的待爬取数据源,第二等待时间waittime可以使用爬取触发时间-当前系统时间来得到,所计算的第二等待时间用于后续控制本地锁的时间。

[0148] S22,服务器删除分布式锁。

[0149] 其中,这里服务器删除分布式锁的目的在于防止当前服务一直占用分布式锁导致的死锁现象,例如在未达到爬取触发时间的数据爬取任务在等待被执行的过程中,对应的服务占用分布式锁,如果又进来一个新的数据爬取任务,可能导致新的数据爬取任务无法执行。因此,服务器可以先删除分布式锁,等到下一轮数据爬取过程再重新加锁。

[0150] S23,服务器判断第一等待时间或第二等待时间是否大于第一预设时长,如果大于则执行S24,如果不大于则执行S25。

[0151] S24,服务器执行本地锁第一预设时长,之后释放锁返回至S13。

[0152] S25,服务器执行本地锁第一等待时间或第二等待时间,之后释放锁返回至S13。

[0153] 也即是说,在等待下一次数据爬取的过程中,服务器会根据第一等待时间或第二等待时间执行本地锁,为避免下一次执行的时间晚于爬取触发时间,这里会判断等待时间是否大于第一预设时长(例如第一预设时长为1分钟),如果大于第一预设时长则本地加锁第一预设时长,避免长时间加锁带来资源浪费,如果不大于第一预设时长则本地加锁第一等待时间或第二等待时间,以尽快执行下一轮爬取任务。如此,在执行本地锁并释放锁之后,服务器会返回至上述S13步骤,以自动开始下一轮爬取任务。

[0154] 示例性地,假设数据爬取任务A的爬取触发时间为10:35:00,当前系统时间为10:30:30,即当前系统时间未达到爬取触发时间,计算得到第二等待时间为4分30秒。在第一预设时长为1分钟的情况下,第二等待时间大于1分钟,则服务器本地加锁1分钟。然后释放锁返回执行S13,即重新开始下一次的数据爬取过程。因数据爬取任务A还未执行,则下一次的数据爬取过程还会对数据爬取任务A进行判断,在上述本地加锁1分钟之后,当前系统时间应来到10:31:30,还是未达到爬取触发时间,计算得到第二等待时间为3分30秒,仍需本地加锁1分钟,由此以此类推。直至当前系统时间来到10:34:30,计算得到第二等待时间为30秒,第二等待时间小于1分钟,则服务器本地加锁30秒,再释放锁返回执行S13。因此时系统当前时间来到了10:35:00,达到爬取触发时间,服务器便可以将该数据爬取任务A对应的URL网址存入数据爬取队列,以供爬虫引擎进行数据爬取。

[0155] 在一些实现方式中,服务器所执行的上述步骤可以由运行在服务器内的定时任务引擎来执行。

[0156] 对于上述S18中服务器将第二待爬取数据源对应的URL网址存入数据爬取队列之后,爬虫引擎(运行在服务器内)识别到数据爬取队列有存入数据即可以开始进行数据爬取。如图12所示,以服务器为执行主体为例,该数据爬取过程可以包括:

[0157] S31,服务器开启持久化线程。

[0158] 需要说明的是,这里所开启的持久化线程与上述S12中的持久化线程并不是同一个线程,上述S12的持久化线程用于持续的从待爬取队列中获取待爬取数据源,以将达到爬

取触发时间的待爬取数据源对应的URL网址存入数据爬取队列,该S31的持久化线程用于持续的根据URL网址进行数据爬取。

[0159] S32,服务器初始化网页下载器和下载线程池。

[0160] 其中,网页下载器是一种数据下载器,用于根据URL网址下载对应的网页数据,服务器可以根据具体的数据爬取业务来初始化网页下载器或者使用默认的网页下载器。下载线程池用于并发的下载网页数据,即可以并发的下载不同URL网址对应的网页数据,服务器可以根据具体的数据爬取业务来初始化下载线程池或者创建默认的下载线程池。

[0161] S33,服务器判断下载线程池的状态是否为运行中,如果是则执行S34,如果否则结束流程。

[0162] 即在下载线程池的状态正常(运行中)的情况下,服务器可以执行后续的数据爬取过程,如果下载线程池的状态不在运行中(可能发生异常或者停止运行),则结束当前流程。

[0163] S34,服务器从数据爬取队列中获取第一待爬取数据源对应的URL网址。

[0164] S35,判断获取的URL网址是否为空,若为空则执行S36,若不为空则执行S37。

[0165] 因上述服务器将第一待爬取数据源对应的URL网址存入了数据爬取队列,则此时可以从数据爬取队列中获取第一待爬取数据源对应的URL网址。考虑到一些异常因素,如读写异常等因素,可能服务器获取到的URL网址为空,那么,服务器可以判断所获取的URL网址是否为空,如果不为空则可以正常进行数据爬取过程,如果为空则当前轮次的数据爬取无法执行,可以进行下一轮过程。例如,服务器所获取的URL网址为上述热榜A的网址。

[0166] S36,服务器执行本地锁第二预设时长,之后释放锁返回S33。

[0167] 在获取的URL网址为空的情况下,服务器可以执行本地锁第二预设时长,以避免频繁进行下一轮循环,造成资源浪费。在一些实现方式中,第二预设时长可以为1秒钟。

[0168] 可以理解,在服务器释放锁返回S33之后,再从数据爬取队列中获取URL网址时,所获取的是下一个新的URL网址。

[0169] S37,服务器通过下载线程池提交下载任务,下载URL网址对应的网页数据。

[0170] 在获取的URL网址不为空的情况下,服务器便可以触发下载URL对应的网页数据,这里可以通过上述下载线程池提交下载任务来下载URL网址对应的网页数据。

[0171] 在一些实现方式中,下载线程池中的最大线程数可以为2C+1,最小线程数可以为2C+1,C为服务器中的CPU核数,这里设置相同的最大线程数和最小线程数是为最大程度的使用CPU资源,下载线程池中的下载任务队列的最大长度可以为M(例如M为1000),以尽可能保证下载任务都提交成功。如果出现下载任务提交失败的情况,服务器也可以将对应的下载任务记录下来,例如记录在数据表中,后续可以再对该数据表中的下载任务进行重新处理。

[0172] 在一些实现方式中,下载URL网址对应的网页数据的过程可以由网页下载器来执行,具体过程可以包括:

[0173] A:网页下载器获取下载链接器。

[0174] 其中,下载链接器是与网页域名相关联的长连接,通过下载链接器可以下载想要获取的数据,例如下载链接器可以使用http连接池实现。

[0175] B:设置http请求消息。

[0176] 其中,http请求消息可以携带请求头、请求体、配置socket的超时时间

(socketTimeout)、配置连接超时时间(connectTimeout)等信息,请求头用于描述http请求消息的元数据,包含请求方法、请求协议等信息,请求体用于向网页服务器传递http请求所需的参数或内容等。

[0177] C:根据http请求消息向网页服务器请求网页数据。

[0178] D:获取网页数据流,以及将数据流转换为文本字符串。

[0179] 即在设置完成http请求消息后,网页下载器便可以向网页服务器请求对应的网页数据,一般情况下,初始获取的网页数据为网页数据流格式,这里还可以将数据流转换为文本字符串(string)。

[0180] S38,服务器获取第一待爬取数据源对应的爬取规则。

[0181] 因上述图10中配置了URL数据源的爬取规则,第一待爬取数据源是被配置的URL数据源中的一个数据源,那么服务器可以获取到第一待爬取数据源对应的爬取规则。在一些实现方式中,如果Web客户端在向服务器发送URL数据源的爬取规则等信息之后,服务器将这些信息存储在第三数据表中,那么服务器便可以从该第三数据表中获取第一待爬取数据源对应的爬取规则。

[0182] S39,判断爬取规则是否为空,若为空则结束流程,若不为空则执行S40。

[0183] 考虑到一些读写异常或URL数据源的爬取规则未完整配置的场景,如果服务器获取的爬取规则为空,则可以结束流程,如果爬取规则不为空,则可以执行后续的规则匹配过程。

[0184] S40,服务器根据爬取规则对网页数据进行匹配。

[0185] 其中,服务器可以根据爬取规则中的规则表达式(例如上述图10中输入的正则表达式),来对所获取的网页数据进行匹配。

[0186] S41,判断爬取规则是否匹配上,如果未匹配上则结束流程,如果匹配上则执行S42。

[0187] 即判断S40中使用的爬取规则是否匹配到数据,如果匹配上则对爬取到的数据进行后续处理。

[0188] S42,服务器循环获取匹配数据,以及根据爬取规则获取必填选项对应的数据。

[0189] 因一个URL网址对应的网页数据的数据量比较大,则服务器可以循环的获取匹配数据,以减少遗漏数据的概率。在一些实现方式中,服务器获取到匹配数据后,还可以按照上述规则表达式中表达式的顺序来拼接匹配数据,例如将匹配数据按照title、source的顺序拼接起来。然后,服务器还可以获取上述图10配置的爬取规则中对是否必填项所设置的结果,并获取匹配数据中必填项对应的数据。

[0190] S43,判断匹配数据是否为空,若为空则结束流程,若不为空则执行S44。

[0191] S44,判断必填项对应的数据是否为空,若为空则结束流程,若不为空则执行S45。

[0192] 也即是说,服务器会分别判断上述获取的匹配数据和必填项对应的数据是否为空,如果有数据为空的情况则可以将该条记录删除,结束当前流程。如果匹配数据和必填项对应的数据都不为空,则执行后续步骤。

[0193] S45,服务器对匹配数据进行格式处理后,保存至存储介质中。

[0194] 例如,在匹配数据是文本字符串的场景下,服务器可以将其转换为json格式,以保存在存储介质中。在一些实现方式中,服务器还可以将匹配数据填充拼接为完整的语句,再

将其保存下来,例如对于上述社会热榜的数据爬取来说,可以将title、source等信息填充拼接为完整的语句。

[0195] 除此之外,在一些实现方式中,服务器对当前URL网址对应的网页数据爬取完成之后,还可以判断该URL网址下是否有子节点信息,即是否有子页面,如果有的话还可以获取到子页面的URL网址,并继续对子页面进行数据爬取,以及将爬取到的数据也保存在存储介质中。

[0196] 在一些实现方式中,上述S38-S45对下载的网页数据进行分析的过程可以由服务器中的网页解析器来执行。

[0197] 可以理解,服务器在完成本轮的数据爬取过程之后,可以返回S33继续从数据爬取队列中获取URL网址进行持久化的数据爬取过程。

[0198] 还可以理解,依据上述图1所示的系统架构,在服务器完成数据爬取之后,如果用户所使用的电子设备上安装有与上述数据爬取任务对应的业务应用,则该业务应用就可以展示数据爬取结果以及可以响应用户的搜索请求。

[0199] 上述数据爬取方法,首先可以在Web客户端上对数据爬取任务进行动态配置,减少编写代码的人力成本,同时服务器可以按照爬取时间间隔持续的进行数据爬取过程,提高了数据爬取的效率,也可以提高数据爬取过程的灵活性。

[0200] 在一些场景上,上述对数据爬取任务进行动态配置的过程除了可以在Web客户端执行之外,也可以在服务器上执行,即研发人员直接在服务器上打开Web管理页面进行配置。在另一些场景中,在Web客户端具有一定的计算处理能力的情况下,上述数据爬取的过程也可以在Web客户端上执行。

[0201] 对于上述Web客户端,其可以是笔记本电脑、PC等设备,示例性地,图13是本申请实施例提供的一例Web客户端的结构示意图。Web客户端可以包括处理器210、存储器220和通信模块230等。

[0202] 其中,处理器210可以包括一个或多个处理单元,存储器220用于存储程序代码和数据。在本申请实施例中,处理器210可执行存储器220存储的计算机执行指令。

[0203] 通信模块230可以用于Web客户端的各个内部模块之间的通信、或者Web客户端和其他设备之间的通信等。示例性地,如果Web客户端通过有线连接的方式和其他设备通信,通信模块230可以包括接口等,例如通用串行总线(universal serial bus,USB)接口,USB接口可以是符合USB标准规范的接口,具体可以是Mini USB接口,Micro USB接口,USB Type C接口等。USB接口可以用于连接充电器为Web客户端充电,也可以用于Web客户端与其他设备之间传输数据,还可以用于连接耳机,通过耳机播放音频等。

[0204] 或者,通信模块230可以包括音频器件、射频电路、蓝牙芯片、无线保真(wireless fidelity,Wi-Fi)芯片、近距离无线通讯技术(near-field communication,NFC)模块等,可以通过多种不同的方式实现Web客户端与其他设备之间的交互。

[0205] 另外,Web客户端还可以包括显示屏240,显示屏240可以显示人机交互界面中的图像或视频等。例如可以显示上述Web管理页面,以供研发人员对数据爬取任务进行配置。

[0206] 可选地,Web客户端还可以包括外设设备250,例如鼠标、键盘、扬声器、麦克风等。

[0207] 应理解,除了图13中列举的各种部件或者模块之外,本申请实施例对Web客户端的结构不做具体限定。在本申请另一些实施例中,Web客户端还可以包括比图示更多或更少的

部件,或者组合某些部件,或者拆分某些部件,或者不同的部件布置。图示的部件可以以硬件,软件或软件和硬件的组合实现。

[0208] 对于上述服务器,其可以是单个服务器,也可以是服务器集群,示例性地,图14是本申请实施例提供的一例服务器的结构示意图。该服务器包括通过系统总线连接的处理器、存储器和网络接口。其中,该服务器的处理器用于提供计算和控制能力。该服务器的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该服务器的数据库可以用于存储上述数据爬取任务的配置数据、URL数据源的配置数据等。该服务器的网络接口可以用于与外部的设备通过网络连接通信。

[0209] 对于上述电子设备,其可以是用户使用的手机、平板电脑等,示例性地,图15是本申请实施例提供的一例电子设备的结构示意图。以电子设备是手机为例,电子设备可以包括处理器110,外部存储器接口120,内部存储器121,USB接口130,充电管理模块140,电源管理模块141,电池142,天线1,天线2,移动通信模块150,无线通信模块160,音频模块170,扬声器170A,受话器170B,麦克风170C,耳机接口170D,传感器模块180,按键190,马达191,指示器192,摄像头193,显示屏194,以及用户标识模块(subscriber identity module,SIM)卡接口195等。其中传感器模块180可以包括压力传感器180A,陀螺仪传感器180B,气压传感器180C,磁传感器180D,加速度传感器180E,距离传感器180F,接近光传感器180G,指纹传感器180H,温度传感器180J,触摸传感器180K,环境光传感器180L,骨传导传感器180M等。

[0210] 处理器110可以包括一个或多个处理单元,例如:处理器110可以包括应用处理器(application processor,AP),调制解调处理器,图形处理器(graphics processing unit,GPU),图像信号处理器(image signal processor,ISP),控制器,存储器,视频编解码器,数字信号处理器(digital signal processor,DSP),基带处理器,和/或神经网络处理器(neural-network processing unit,NPU)等。其中,不同的处理单元可以是独立的器件,也可以集成在一个或多个处理器中。

[0211] 其中,控制器可以是电子设备的神经中枢和指挥中心。控制器可以根据指令操作码和时序信号,产生操作控制信号,完成取指令和执行指令的控制。

[0212] 处理器110中还可以设置存储器,用于存储指令和数据。在一些实施例中,处理器110中的存储器为高速缓冲存储器。该存储器可以保存处理器110刚用过或循环使用的指令或数据。如果处理器110需要再次使用该指令或数据,可从存储器中直接调用。避免了重复存取,减少了处理器110的等待时间,因而提高了系统的效率。

[0213] 移动通信模块150可以提供应用在电子设备上的包括2G/3G/4G/5G等无线通信的解决方案。移动通信模块150可以包括至少一个滤波器,开关,功率放大器,低噪声放大器(low noise amplifier,LNA)等。移动通信模块150可以由天线1接收电磁波,并对接收的电磁波进行滤波,放大等处理,传送至调制解调处理器进行解调。移动通信模块150还可以对经调制解调处理器调制后的信号放大,经天线1转为电磁波辐射出去。在一些实施例中,移动通信模块150的至少部分功能模块可以被设置于处理器110中。在一些实施例中,移动通信模块150的至少部分功能模块可以与处理器110的至少部分模块被设置在同一个器件中。

[0214] 无线通信模块160可以提供应用在电子设备上的包括无线局域网(wireless local area networks,WLAN)(如无线保真(wireless fidelity,Wi-Fi)网络),蓝牙

(bluetooth,BT),全球导航卫星系统(global navigation satellite system,GNSS),调频(frequency modulation,FM),近距离无线通信技术(near field communication,NFC),红外技术(infrared,IR)等无线通信的解决方案。无线通信模块160可以是集成至少一个通信处理模块的一个或多个器件。无线通信模块160经由天线2接收电磁波,将电磁波信号调频以及滤波处理,将处理后的信号发送到处理器110。无线通信模块160还可以从处理器110接收待发送的信号,对其进行调频,放大,经天线2转为电磁波辐射出去。

[0215] 电子设备通过GPU,显示屏194,以及应用处理器等实现显示功能。GPU为图像处理的微处理器,连接显示屏194和应用处理器。GPU用于执行数学和几何计算,用于图形渲染。处理器110可包括一个或多个GPU,其执行程序指令以生成或改变显示信息。

[0216] 显示屏194用于显示图像,视频等。显示屏194包括显示面板。显示面板可以采用液晶显示屏(liquid crystal display,LCD),有机发光二极管(organic light-emitting diode,OLED),有源矩阵有机发光二极体或主动矩阵有机发光二极体(active-matrix organic light emitting diode的,AMOLED),柔性发光二极管(flex light-emitting diode,FLED),Miniled,MicroLed,Micro-oLed,量子点发光二极管(quantum dot light emitting diodes,QLED)等。在一些实施例中,电子设备可以包括1个或N个显示屏194,N为大于1的正整数。

[0217] 内部存储器121可以用于存储计算机可执行程序代码,可执行程序代码包括指令。处理器110通过运行存储在内部存储器121的指令,从而执行电子设备的各种功能应用以及数据处理。内部存储器121可以包括存储程序区和存储数据区。其中,存储程序区可存储操作系统,至少一个功能所需的应用程序(比如声音播放功能,图像播放功能等)等。存储数据区可存储电子设备使用过程中所创建的数据(比如音频数据,电话本等)等。此外,内部存储器121可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件,闪存器件,通用闪存存储器(universal flash storage,UFS)等。

[0218] 可以理解的是,本申请实施例示意的结构并不构成对电子设备的具体限定。在本申请另一些实施例中,电子设备可以包括比图示更多或更少的部件,或者组合某些部件,或者拆分某些部件,或者不同的部件布置。图示的部件可以以硬件,软件或软件和硬件的组合实现。

[0219] 上文详细介绍了本申请实施例提供的数据爬取方法的示例。可以理解的是,Web客户端、服务器和电子设备为了实现上述功能,其包含了执行各个功能相应的硬件和/或软件模块。本领域技术人员应该很容易意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,本申请能够以硬件或硬件和计算机软件的结合形式来实现。某个功能究竟以硬件还是计算机软件驱动硬件的方式来执行,取决于技术方案的特定应用和设计约束条件。本领域技术人员可以结合实施例对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0220] 本申请实施例可以根据上述方法示例对Web客户端、服务器和电子设备进行功能模块的划分,例如,可以对应各个功能划分为各个功能模块,例如检测单元、处理单元、显示单元等,也可以将两个或两个以上的功能集成在一个模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。需要说明的是,本申请实施例中对模块的划分是示意性的,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式。

[0221] 需要说明的是,上述方法实施例涉及的各步骤的所有相关内容均可以援引到对应功能模块的功能描述,在此不再赘述。

[0222] 在采用集成的单元的情况下,电子设备还可以包括处理模块、存储模块和通信模块。其中,处理模块可以用于对电子设备的动作进行控制管理。存储模块可以用于支持电子设备执行存储程序代码和数据等。通信模块,可以用于支持电子设备与其他设备的通信。

[0223] 其中,处理模块可以是处理器或控制器。其可以实现或执行结合本申请公开内容所描述的各种示例性的逻辑方框,模块和电路。处理器也可以是实现计算功能的组合,例如包含一个或多个微处理器组合,数字信号处理器和微处理器的组合等等。存储模块可以是存储器。通信模块具体可以为射频电路、蓝牙芯片、Wi-Fi芯片等与其他电子设备交互的设备。

[0224] 本申请实施例还提供了一种计算机可读存储介质,计算机可读存储介质中存储了计算机程序,当计算机程序被处理器执行时,使得处理器执行上述任一实施例的数据爬取方法。存储介质可以包括:U盘、移动硬盘、只读存储器(read only memory,ROM)、随机存取存储器(random access memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0225] 本申请实施例还提供了一种计算机程序产品,当该计算机程序产品在计算机上运行时,使得计算机执行上述相关步骤,以实现上述实施例中的数据爬取方法。

[0226] 另外,本申请的实施例还提供一种装置,这个装置具体可以是芯片,组件或模块,该装置可包括相连的处理器和存储器;其中,存储器用于存储计算机执行指令,当装置运行时,处理器可执行存储器存储的计算机执行指令,以使芯片执行上述各方法实施例中的数据爬取方法。

[0227] 其中,本实施例提供的Web客户端、服务器、电子设备、计算机可读存储介质、计算机程序产品或芯片均用于执行上文所提供的对应的方法,因此,其所能达到的有益效果可参考上文所提供的对应的方法中的有益效果,此处不再赘述。

[0228] 通过以上实施方式的描述,所属领域的技术人员可以了解到,为描述的方便和简洁,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将装置的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。

[0229] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0230] 以上内容,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以权利要求的保护范围为准。

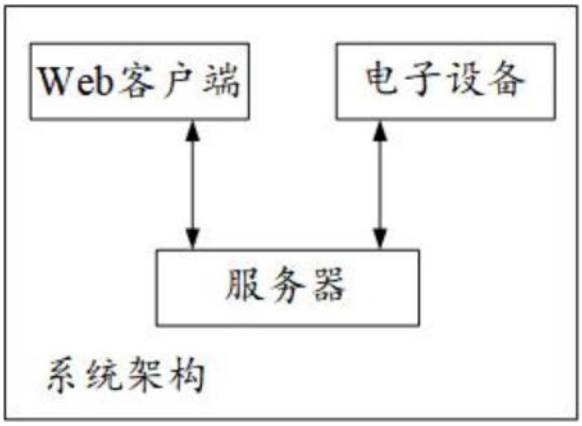


图1

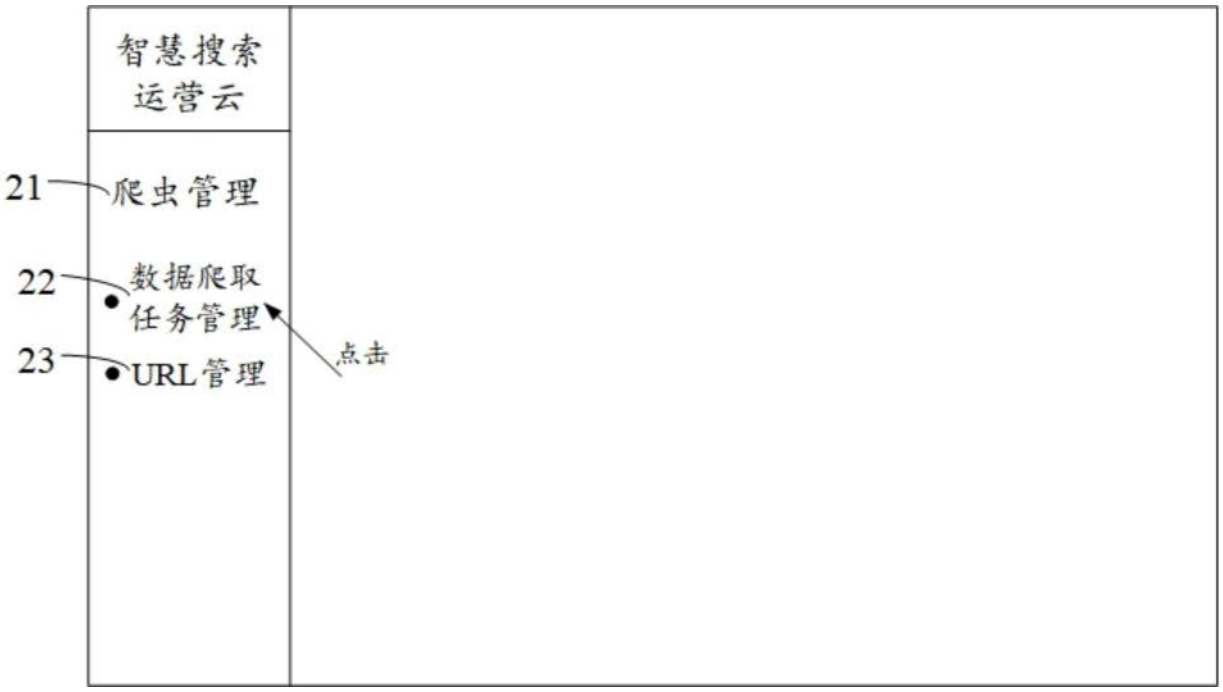


图2

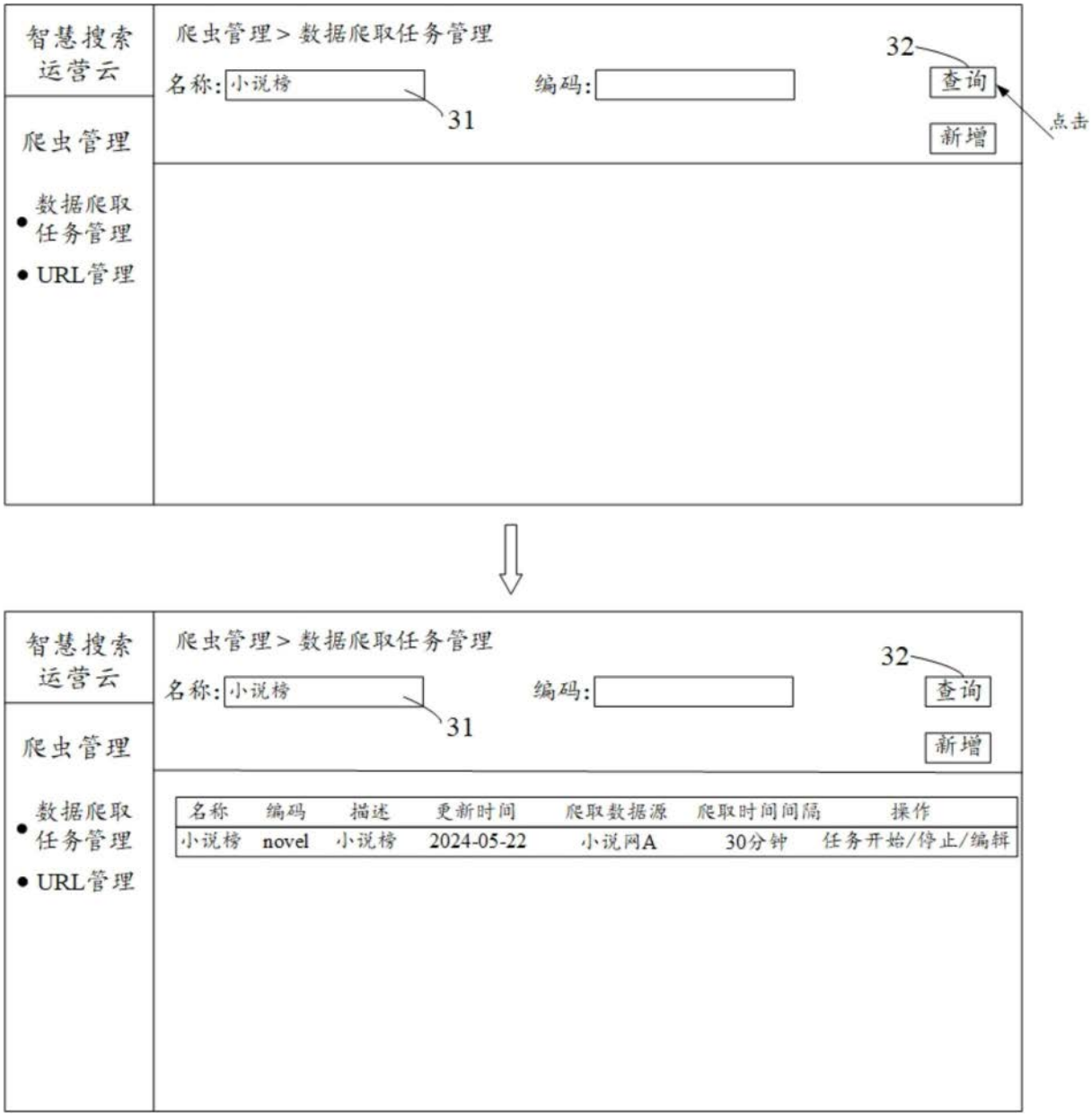

智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理>数据爬取任务管理
名称:小说榜
31
编码:
32
查询
点击
新增
智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理>数据爬取任务管理
名称:小说榜
31
编码:
32
查询
新增
名称 编码 描述 更新时间 爬取数据源 爬取时间间隔 操作
小说榜 novel 小说榜 2024-05-22 小说网A 30分钟 任务开始/停止/编辑

图3

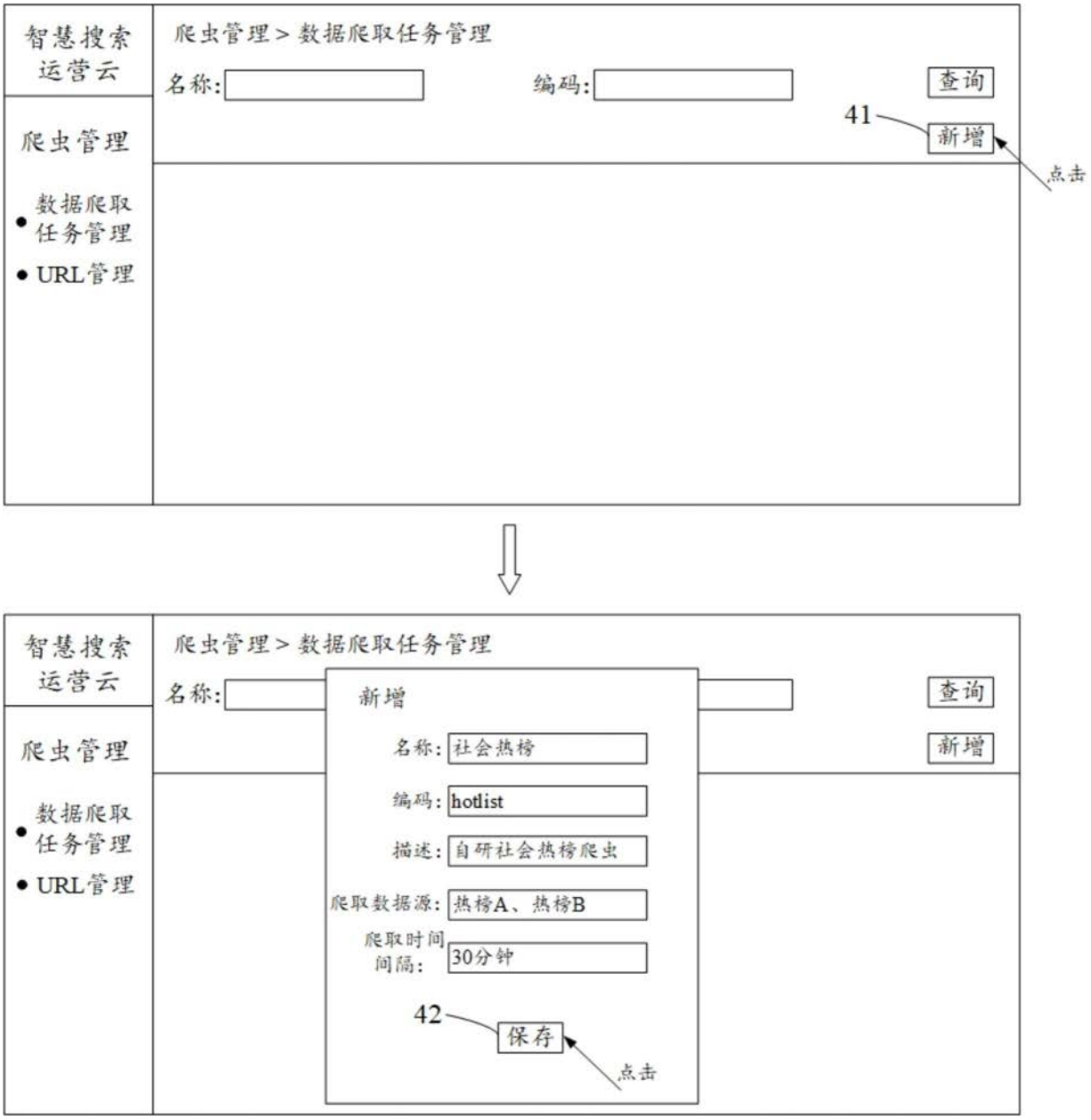

智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理 > 数据爬取任务管理
名称:
编码:
查询
41
新增
点击
智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理 > 数据爬取任务管理
名称:
查询
新增
新增
名称: 社会热榜
编码: hotlist
描述: 自研社会热榜爬虫
爬取数据源: 热榜A、热榜B
爬取时间间隔: 30分钟
42
保存
点击

图4

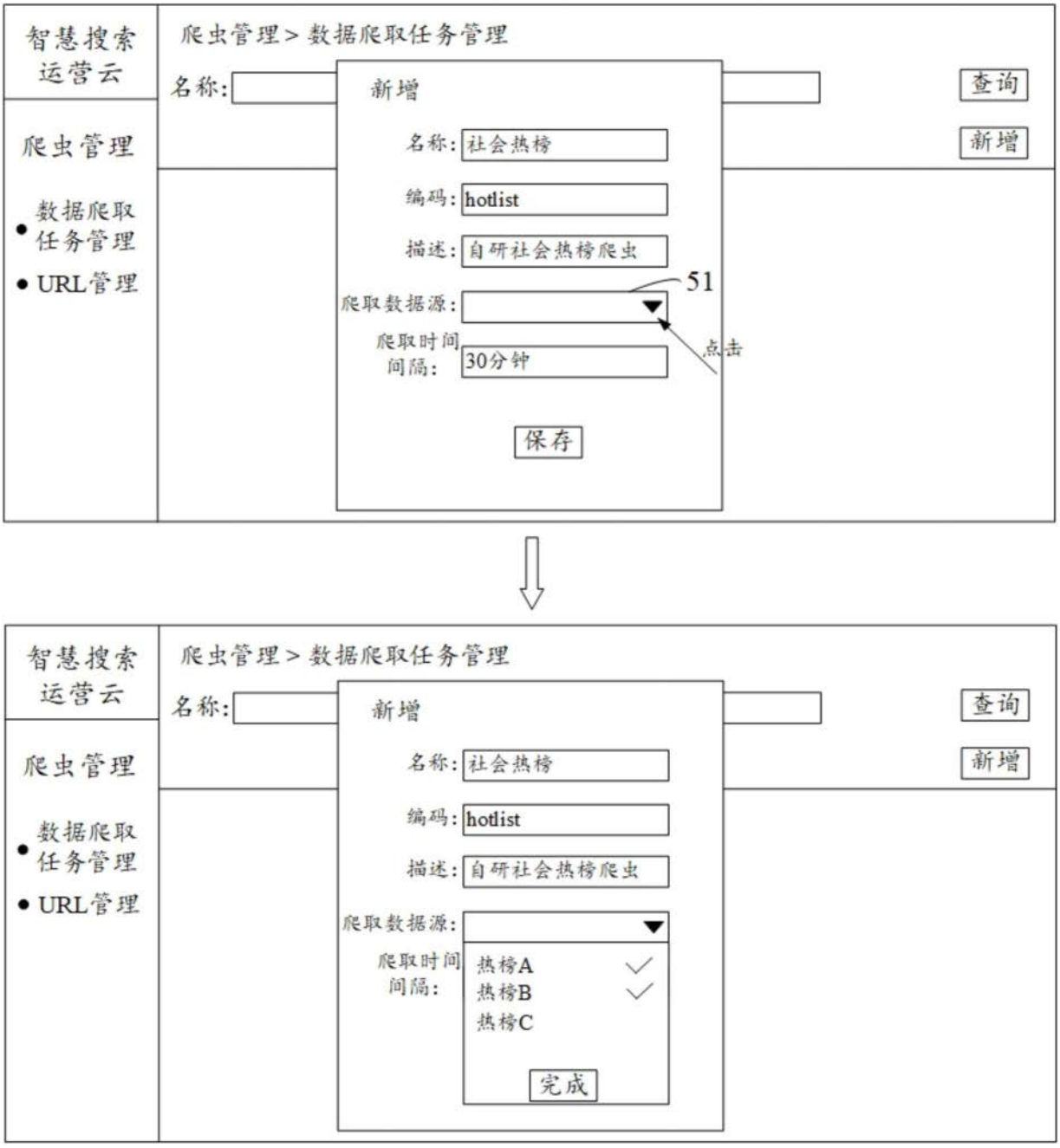

智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理>数据爬取任务管理
名称:
查询
新增
新增
名称:社会热榜
编码:hotlist
描述:自研社会热榜爬虫
爬取数据源:
51
点击
爬取时间间隔:30分钟
保存
智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理>数据爬取任务管理
名称:
查询
新增
新增
名称:社会热榜
编码:hotlist
描述:自研社会热榜爬虫
爬取数据源:
爬取时间间隔:
热榜A
热榜B
热榜C
完成

图5

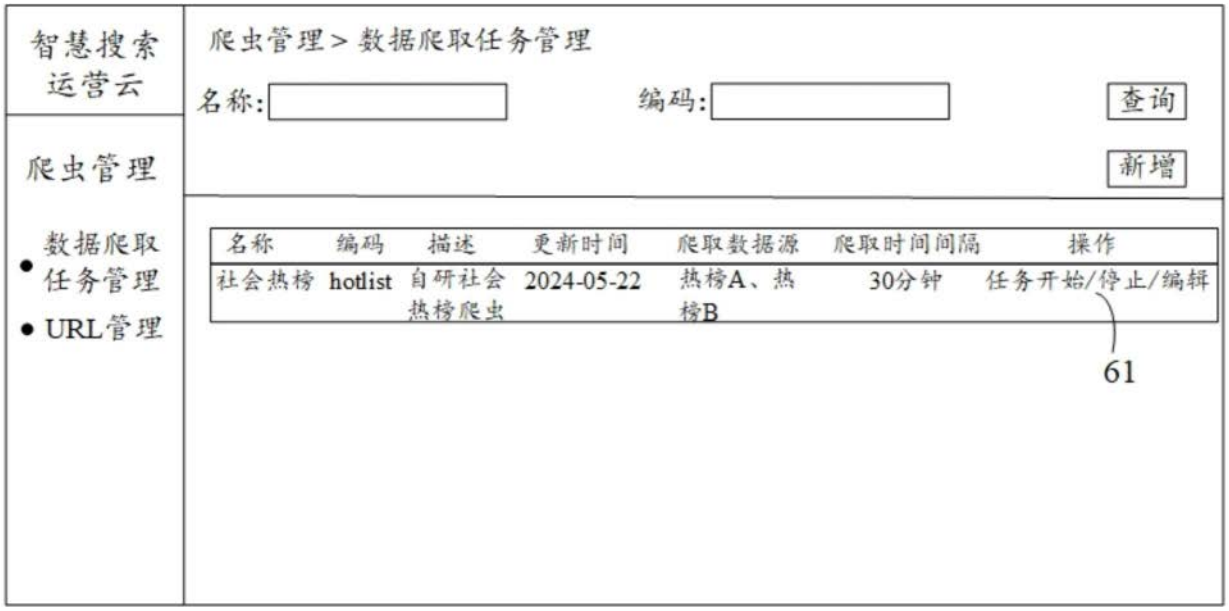

智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理 > 数据爬取任务管理
名称:
编码:
查询
新增
名称 编码 描述 更新时间 爬取数据源 爬取时间间隔 操作
社会热榜 hotlist 自研社会热榜爬虫 2024-05-22 热榜A、热榜B 30分钟 任务开始/停止/编辑
61

图6

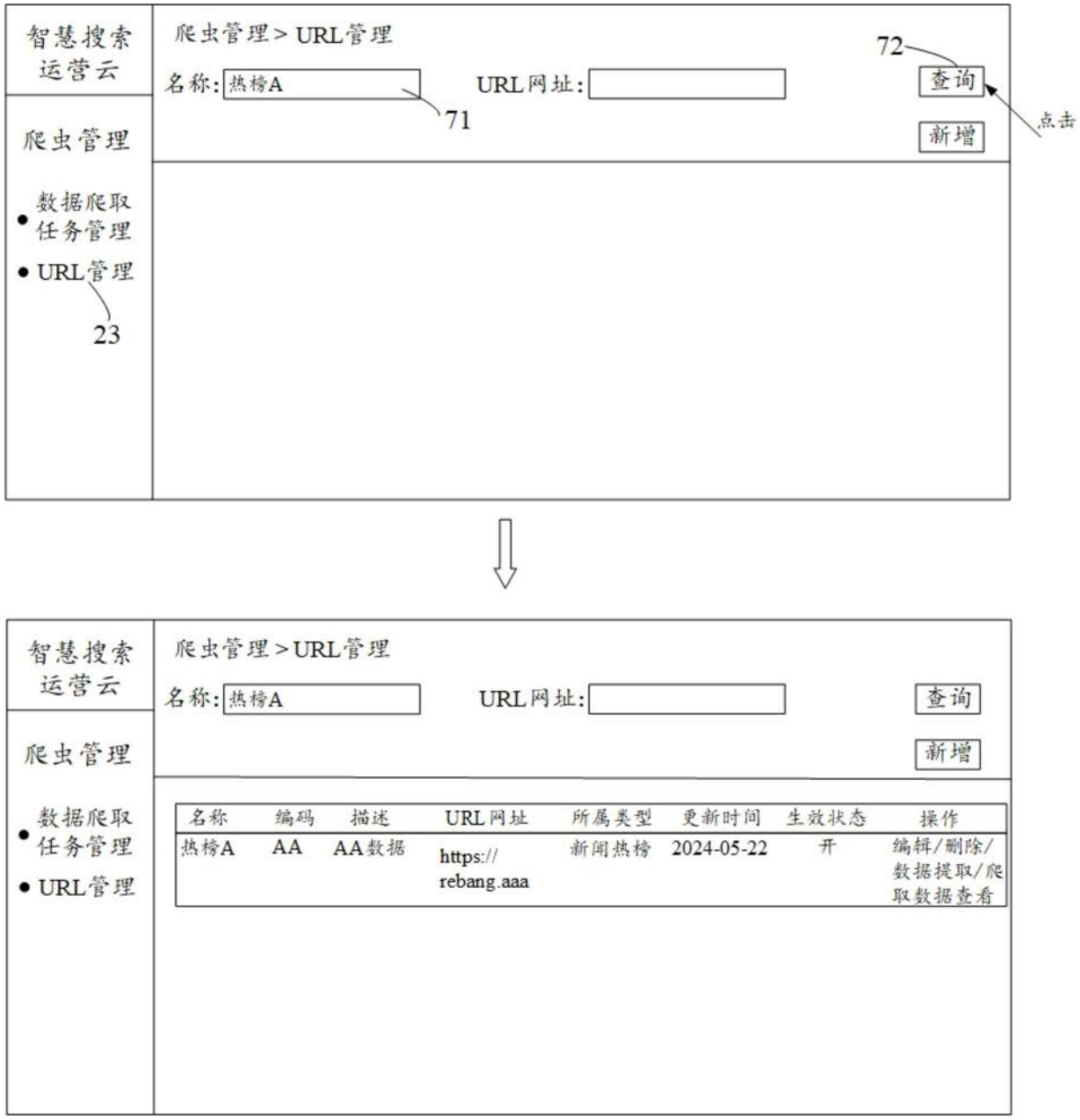

智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
23
爬虫管理>URL管理
名称:热榜A
71
URL网址:
72
查询
点击
新增
智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理>URL管理
名称:热榜A
URL网址:
查询
新增
名称 编码 描述 URL网址 所属类型 更新时间 生效状态 操作
热榜A AA AA数据 https://rebang.aaa 新闻热榜 2024-05-22 开 编辑/删除/数据提取/爬取数据查看

图7

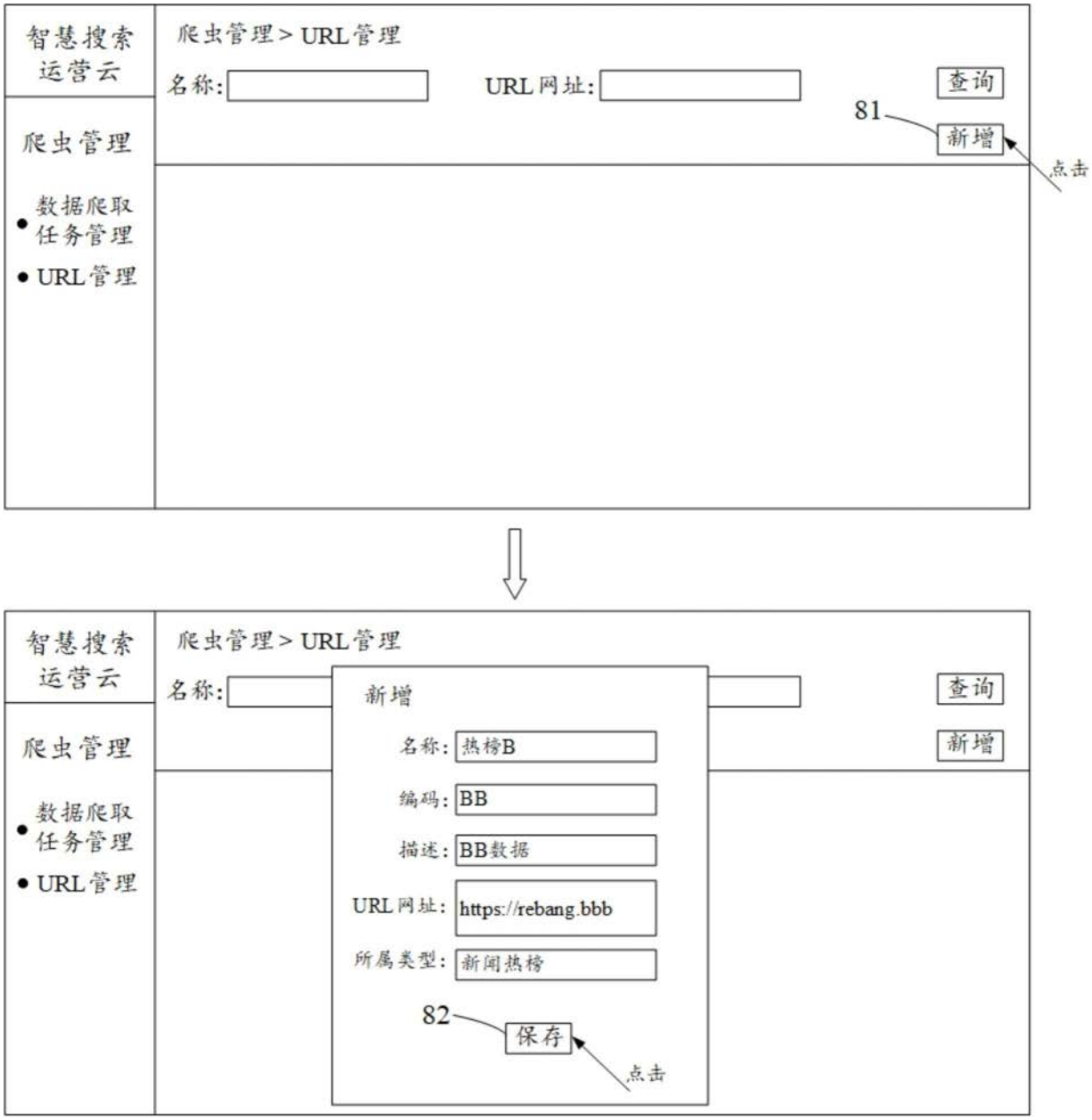

智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理>URL管理
名称:
URL网址:
查询
新增
81
点击
智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理>URL管理
名称:
查询
新增
新增
名称:
热榜B
编码:
BB
描述:
BB数据
URL网址:
https://rebang.bbb
所属类型:
新闻热榜
82
保存
点击

图8

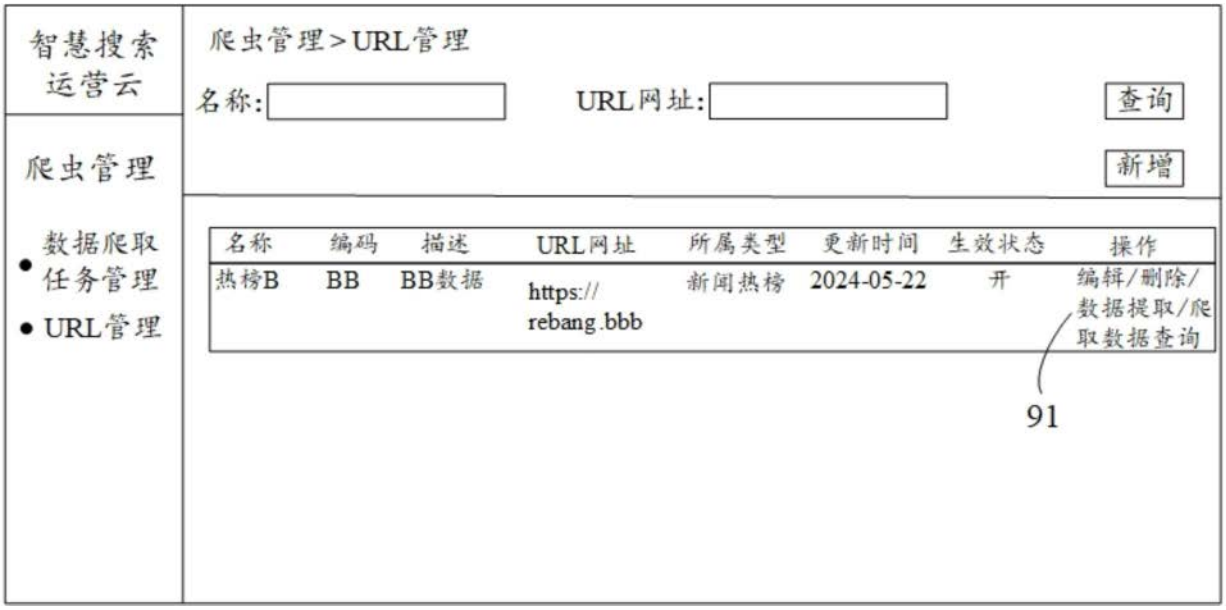

智慧搜索运营云
爬虫管理
数据爬取任务管理
URL管理
爬虫管理>URL管理
名称:
URL网址:
查询
新增
名称 编码 描述 URL网址 所属类型 更新时间 生效状态 操作
热榜B BB BB数据 https://rebang.bbb 新闻热榜 2024-05-22 开 编辑/删除/数据提取/爬取数据查询
91

图9

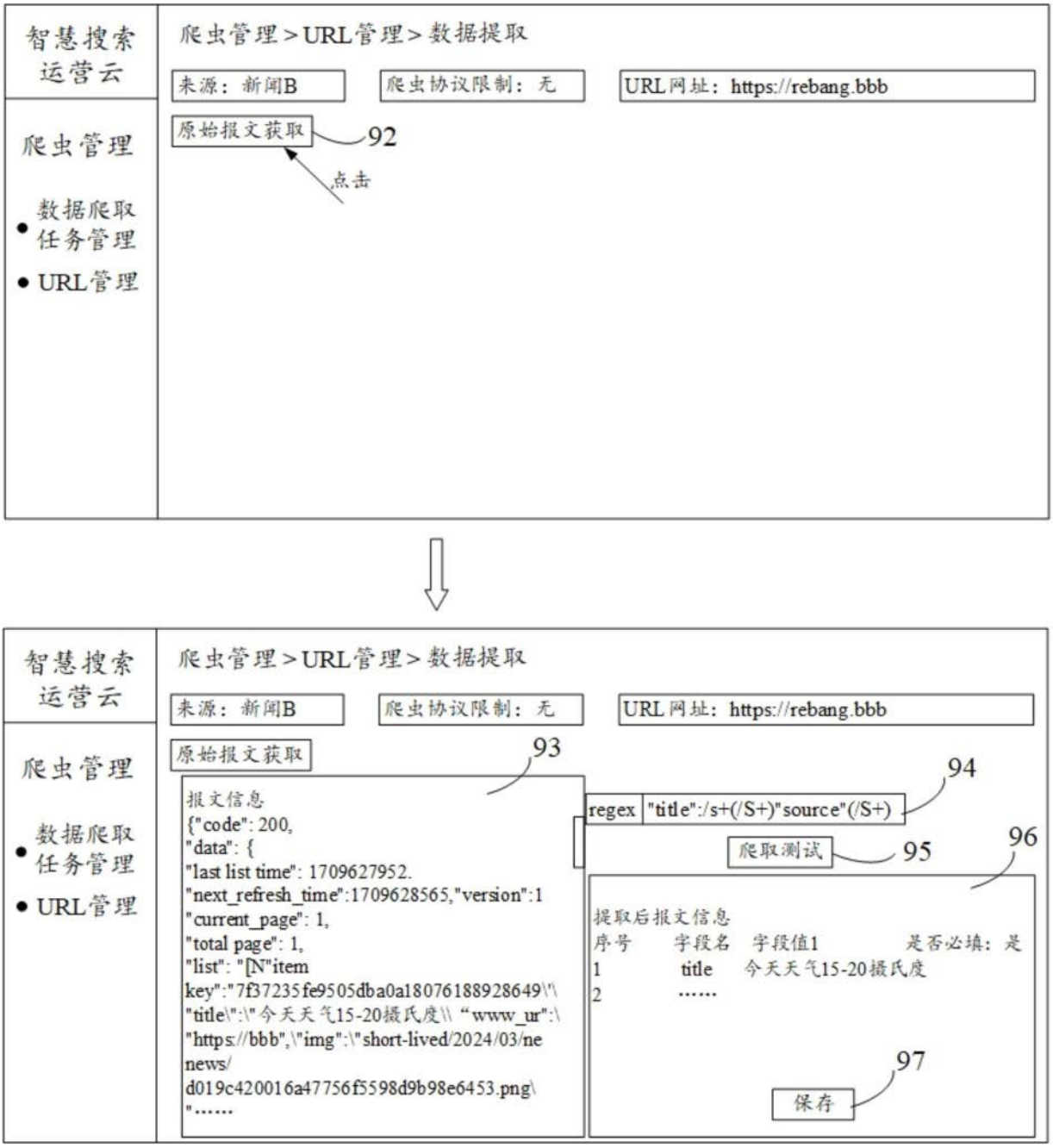

智慧搜索运营云
爬虫管理
• 数据爬取任务管理
• URL管理
爬虫管理>URL管理>数据提取
来源：新闻B
爬虫协议限制：无
URL网址：https://rebang.bbb
原始报文获取
92
点击
93
报文信息
{"code": 200,
"data": {
"last list time": 1709627952.
"next_refresh_time":1709628565,"version":1
"current_page": 1,
"total page": 1,
"list": "[N"item
key":"7f37235fe9505dba0a18076188928649\"\
"title\":\"今天天气15-20摄氏度\\ “www_ur":\
"https://bbb",\"img":\"short-lived/2024/03/ne
news/
d019c420016a47756f5598d9b98e6453.png\
"……
94
regex "title":/s+(/S+)"source"(/S+)
爬取测试
95
96
提取后报文信息
序号 字段名 字段值1 是否必填：是
1 title 今天天气15-20摄氏度
2 ……
97
保存

图10

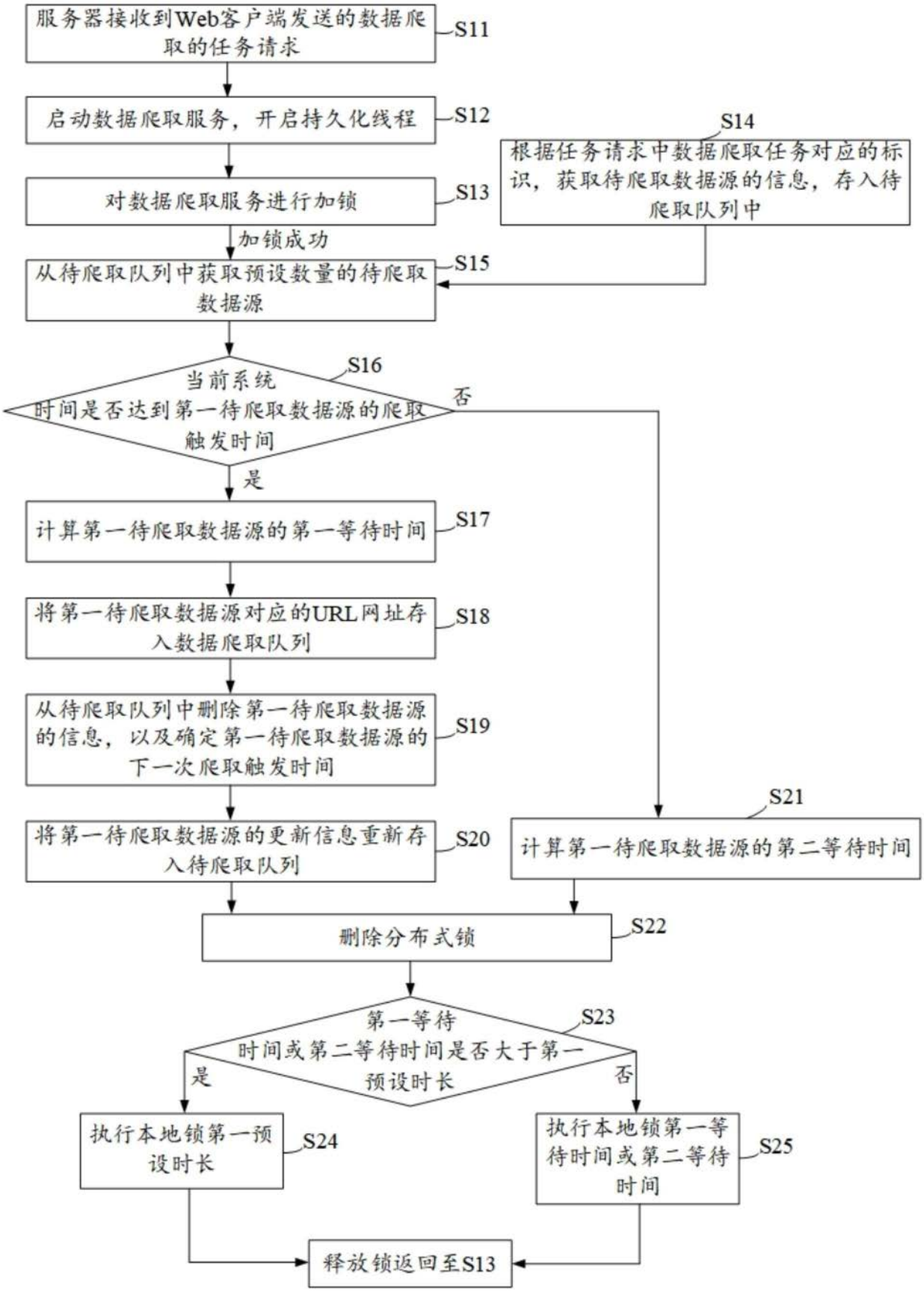

服务器接收到Web客户端发送的数据爬取的任务请求
S11
启动数据爬取服务，开启持久化线程
S12
对数据爬取服务进行加锁
S13
加锁成功
S14
根据任务请求中数据爬取任务对应的标识，获取待爬取数据源的信息，存入待爬取队列中
从待爬取队列中获取预设数量的待爬取数据源
S15
当前系统时间是否达到第一待爬取数据源的爬取触发时间
S16
否
是
计算第一待爬取数据源的第一等待时间
S17
将第一待爬取数据源对应的URL网址存入数据爬取队列
S18
从待爬取队列中删除第一待爬取数据源的信息，以及确定第一待爬取数据源的下一次爬取触发时间
S19
将第一待爬取数据源的更新信息重新存入待爬取队列
S20
S21
计算第一待爬取数据源的第二等待时间
删除分布式锁
S22
第一等待时间或第二等待时间是否大于第一预设时长
S23
是
否
执行本地锁第一预设时长
S24
执行本地锁第一等待时间或第二等待时间
S25
释放锁返回至S13

图11

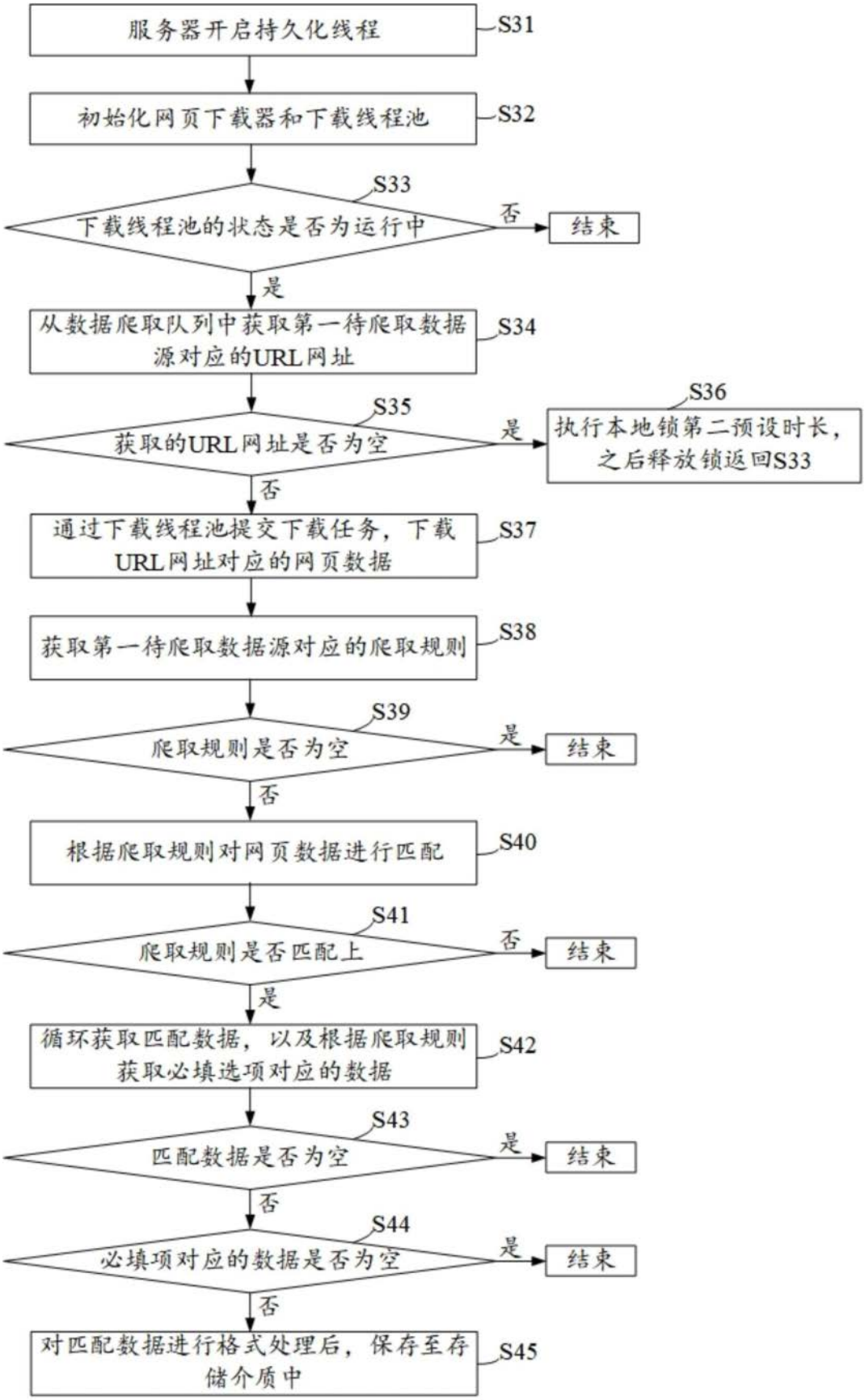

服务器开启持久化线程
S31
初始化网页下载器和下载线程池
S32
S33
下载线程池的状态是否为运行中
否
结束
是
从数据爬取队列中获取第一待爬取数据源对应的URL网址
S34
S35
获取的URL网址是否为空
是
S36
执行本地锁第二预设时长，之后释放锁返回S33
否
通过下载线程池提交下载任务，下载URL网址对应的网页数据
S37
获取第一待爬取数据源对应的爬取规则
S38
S39
爬取规则是否为空
是
结束
否
根据爬取规则对网页数据进行匹配
S40
S41
爬取规则是否匹配上
否
结束
是
循环获取匹配数据，以及根据爬取规则获取必填选项对应的数据
S42
S43
匹配数据是否为空
是
结束
否
S44
必填项对应的数据是否为空
是
结束
否
对匹配数据进行格式处理后，保存至存储介质中
S45

图12


Web客户端
存储器
220
处理器
210
通信模块
230
显示屏
240
外设设备
250

图13

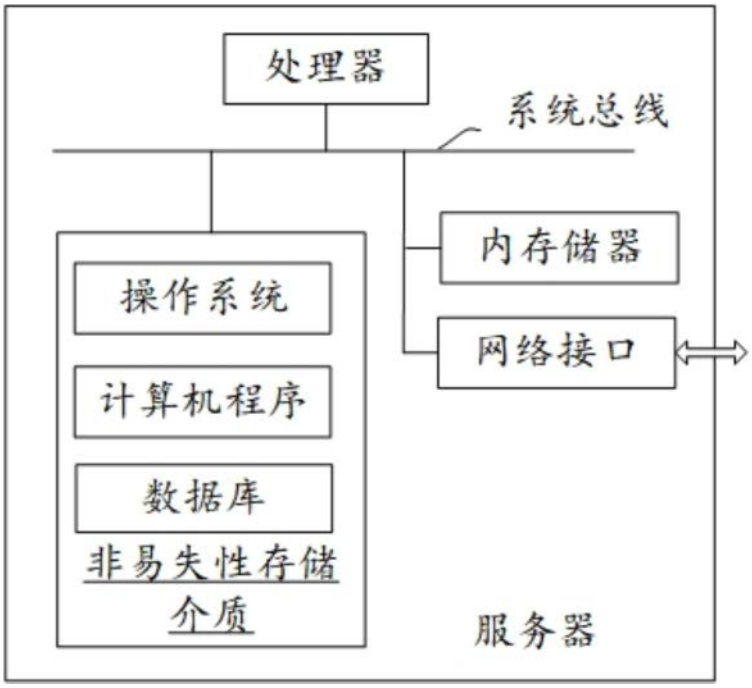

处理器
系统总线
内存储器
操作系统
网络接口
计算机程序
数据库
非易失性存储介质
服务器

图14


电子设备
天线1
天线2
移动通信模块
2G/3G/4G/5G
[150]
无线通信模块
BT/WLAN/GNSS/NFC/IR/FM
[160]
扬声器
[170A]
受话器
[170B]
麦克风
[170C]
耳机接口
[170D]
音频
模块
[170]
显示屏1-N[194]
摄像头1-N[193]
指示器[192]
马达[191]
按键[190]
内部存储器[121]
SIM卡接口1-N
[195]
外部存储器接口
[120]
处理器
[110]
传感器模块[180]
压力传感器[180A]
陀螺仪传感器[180B]
气压传感器[180C]
磁传感器[180D]
加速度传感器[180E]
距离传感器[180F]
接近光传感器[180G]
指纹传感器[180H]
温度传感器[180J]
触摸传感器[180K]
环境光传感器[180L]
骨传导传感器
[180M]
USB接口[130]
充电管理模块
[140]
电源管理模块
[141]
电池[142]
充电输入

图15