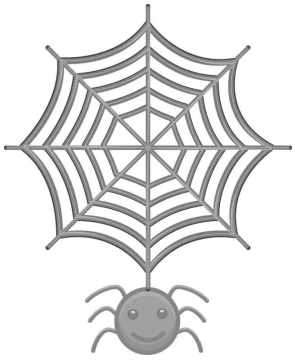
搜索引擎技术之网络爬虫

转载

算法与数学之美 于 2021-09-26 20:20:00 发布 176 收藏
文章标签： 网络 搜索引擎 python java 机器学习
原文链接：https://mp.weixin.qq.com/s?__biz=MzA5ODUxOTA5Mg==&mid=2652588580&idx=2&sn=0b56d6e9c0323d9da69f2a2898733e29&chksm=8b7fbb7fbc0832692fdec548ac9c0278e0dc70d1b4832ab21a616063ff8cd8fa9c520c3c3432&scene=126&&session
版权

阅读目录

>>>>



随着互联网的大力发展，互联网称为信息的主要载体，而如何在互联网中搜集信息是互联网领域面临的一大挑战。网络爬虫技术是什么？其实网络爬虫技术就是指的网络数据的抓取，因为在网络中抓取数据是具有关联性的抓取，它就像是一只蜘蛛一样在互联网中爬来爬去，所以我们很形象地将其称为是网络爬虫技术。其中网络爬虫也被称为是网络机器人或者是网络追逐者。

网络爬虫技术是搜索引擎架构中最为根本的数据技术，通过网络爬虫技术，我们可以将互联网中数以百亿计的网页信息保存到本地，形成一个镜像文件，为整个搜索引擎提供数据支撑。

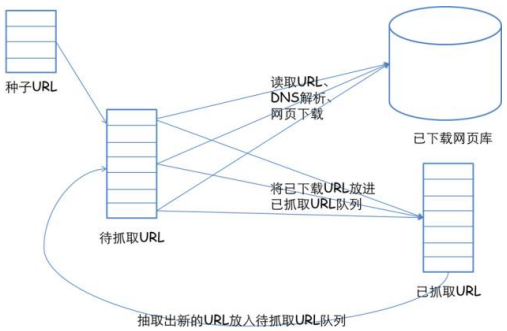
1. 网络爬虫技术基本工作流程和基础架构

网络爬虫获取网页信息的方式和我们平时使用浏览器访问网页的工作原理是完全一样的，都是根据HTTP协议来获取，其流程主要包括如下步骤：

1）连接DNS域名服务器，将待抓取的URL进行域名解析（URL------>IP）；

2）根据HTTP协议，发送HTTP请求来获取网页内容。

一个完整的网络爬虫基础框架如下图所示：



整个架构共有如下几个过程：

1）需求方提供需要抓取的种子URL列表，根据提供的URL列表和相应的优先级，建立待抓取URL队列（先来先抓）；

2）根据待抓取URL队列的排序进行网页抓取；

3）将获取的网页内容和信息下载到本地的网页库，并建立已抓取URL列表（用于去重和判断抓取的进程）；

4）将已抓取的网页放入到待抓取的URL队列中，进行循环抓取操作；

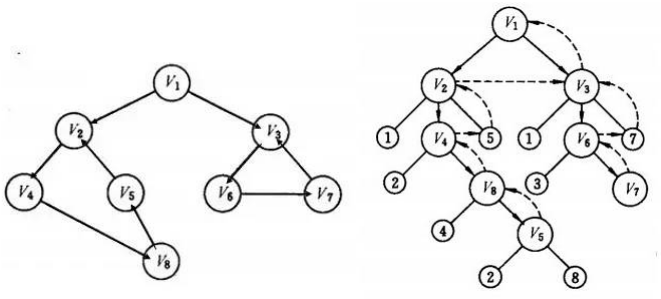
2. 网络爬虫的抓取策略

在爬虫系统中，待抓取URL队列是很重要的一部分。待抓取URL队列中的URL以什么样的顺序排列也是一个很重要的问题，因为这涉及到先抓取哪个页面，后抓取哪个页面的问题。而决定这些URL排列顺序的方法，叫做抓取策略。下面重点介绍几种常见的抓取策略：

1）深度优先遍历策略

深度优先遍历策略很好理解，这跟我们有向图中的深度优先遍历是一样的，因为网络本身就是一种图模型嘛。深度优先遍历的思路是先从一个起始网页开始抓取，然后对根据链接一个一个的逐级进行抓取，直到不能再深入抓取为止，返回上一级网页继续跟踪链接。

一个有向图深度优先搜索的实例如下所示：

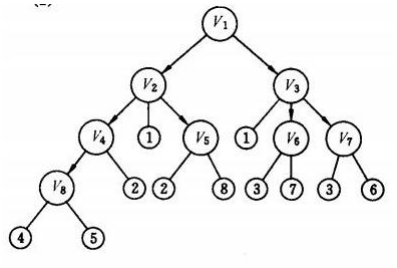


上图左图为一个有向图示意图，右图为深度优先遍历的搜索过程示意图。深度优先遍历的结果为：

$v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_8 \rightarrow v_5 \rightarrow v_3 \rightarrow v_6 \rightarrow v_7$

2）广度优先搜索策略

广度优先搜索和深度优先搜索的工作方式正好是相对的，其思想为：将新下载网页中发现的链接直接插入待抓取URL队列的末尾。也就是指网络爬虫会先抓取起始网页中链接的所有网页，然后再选择其中的一个链接网页，继续抓取在此网页中链接的所有网页。



上图为上边实例的有向图的广度优先搜索流程图，其遍历的结果为：

v1→v2 →v3 →v4→ v5→ v6→ v7 →v8

从树的结构上去看，图的广度优先遍历就是树的层次遍历。

3）反向链接搜索策略

反向链接数是指一个网页被其他网页链接指向的数量。反向链接数表示的是一个网页的内容受到其他人的推荐的程度。因此，很多时候搜索引擎的抓取系统会使用这个指标来评价网页的重要程度，从而决定不同网页的抓取先后顺序。

在真实的网络环境中，由于广告链接、作弊链接的存在，反向链接数不能完全等他我那个也的重要程度。因此，搜索引擎往往考虑一些可靠的反向链接数。

4）大站优先策略

对于待抓取URL队列中的所有网页，根据所属的网站进行分类。对于待下载页面数多的网站，优先下载。这个策略也因此叫做大站优先策略。

5）其他搜索策略

一些比较常用的爬虫搜索侧率还包括Partial PageRank搜索策略（根据PageRank分值确定下一个抓取的URL）、OPIC搜索策略（也是一种重要性排序）。最后必须要指明的一点是，我们可以根据自己的需求为网页的抓取间隔时间进行设定，这样我们就可以确保我们基本的一些大站或者活跃的站点内容不会被漏抓。

3. 网络爬虫更新策略

互联网是实时变化的，具有很强的动态性。网页更新策略主要是决定何时更新之前已经下载过的页面。常见的更新策略又以下三种：

1）历史参考策略

顾名思义，根据页面以往的历史更新数据，预测该页面未来何时会发生变化。一般来说，是通过泊松过程进行建模进行预测。

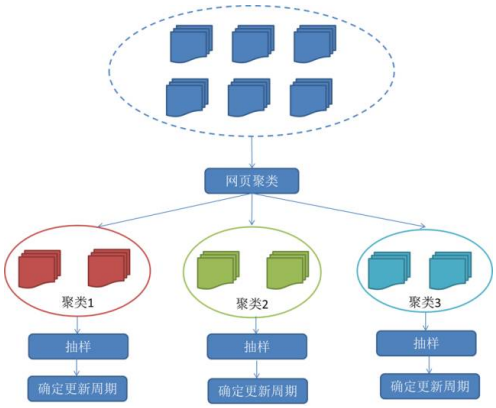
2）用户体验策略

尽管搜索引擎针对于某个查询条件能够返回数量巨大的结果，但是用户往往只关注前几页结果。因此，抓取系统可以优先更新那些现实在查询结果前几页中的网页，而后再更新那些后面的网页。这种更新策略也是需要用到历史信息的。用户体验策略保留网页的多个历史版本，并且根据过去每次内容变化对搜索质量的影响，得出一个平均值，用这个值作为决定何时重新抓取的依据。

3）聚类抽样策略

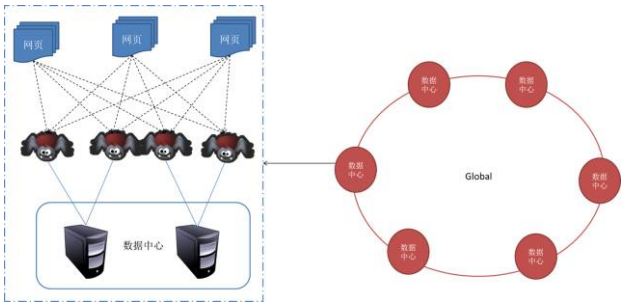
前面提到的两种更新策略都有一个前提：需要网页的历史信息。这样就存在两个问题：第一，系统要是为每个系统保存多个版本的历史信息，无疑增加了很多的系统负担；第二，要是新的网页完全没有历史信息，就无法确定更新策略。

这种策略认为，网页具有很多属性，类似属性的网页，可以认为其更新频率也是类似的。要计算某一个类别网页的更新频率，只需要对这一类网页抽样，以他们的更新周期作为整个类别的更新周期。基本思路如图：



4. 分布式抓取系统结构

一般来说，抓取系统需要面对的是整个互联网上数以亿计的网页。单个抓取程序不可能完成这样的任务。往往需要多个抓取程序一起来处理。一般来说抓取系统往往是一个分布式的三层结构。如图所示：

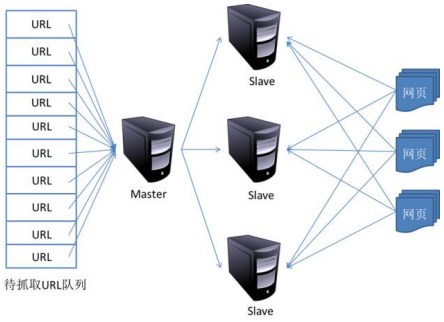


最下一层是分布在不同地理位置的数据中心，在每个数据中心里有若干台抓取服务器，而每台抓取服务器上可能部署了若干套爬虫程序。这就构成了一个基本的分布式抓取系统。

对于一个数据中心内的不同抓去服务器，协同工作的方式有几种：

1）主从式（Master-Slave）

主从式基本结构如图所示：

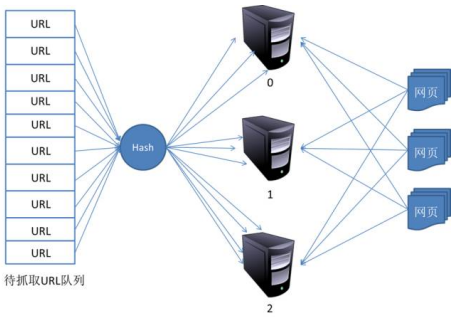


对于主从式而言，有一台专门的Master服务器来维护待抓取URL队列，它负责每次将URL分发到不同的Slave服务器，而Slave服务器则负责实际的网页下载工作。Master服务器除了维护待抓取URL队列以及分发URL之外，还要负责调解各个Slave服务器的负载情况。以免某些Slave服务器过于清闲或者劳累。

这种模式下，Master往往容易成为系统瓶颈。

2）对等式（Peer to Peer）

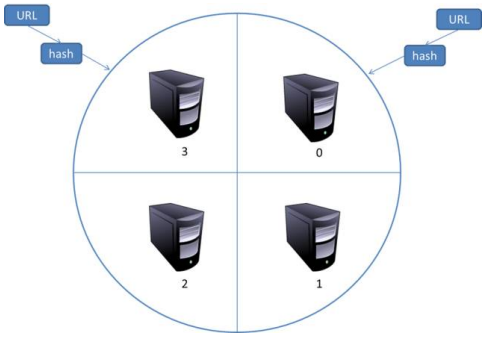
对等式的基本结构如图所示：



在这种模式下，所有的抓取服务器在分工上没有不同。每一台抓取服务器都可以从待抓取在URL队列中获取URL，然后对该URL的主域名的hash值H，然后计算H mod m（其中m是服务器的数量，以上图为例，m为3），计算得到的数就是处理该URL的主机编号。

举例：假设对于URL www.baidu.com，计算器hash值H=8，m=3，则H mod m=2，因此由编号为2的服务器进行该链接的抓取。假设这时候是0号服务器拿到这个URL，那么它将该URL转给服务器2，由服务器2进行抓取。

这种模式有一个问题，当有一台服务器死机或者添加新的服务器，那么所有URL的哈希求余的结果就都要变化。也就是说，这种方式的扩展性不佳。针对这种情况，又有一种改进方案被提出来。这种改进的方案是一致性哈希法来确定服务器分工。其基本结构如图所示：



一致性哈希将URL的主域名进行哈希运算，映射为一个范围在0-2^{32}之间的某个数。而将这个范围平均的分配给m台服务器，根据URL主域名哈希运算的值所处的范围判断是哪台服务器来进行抓取。

如果某一台服务器出现问题，那么本该由该服务器负责的网页则按照顺时针顺延，由下一台服务器进行抓取。这样的话，及时某台服务器出现问题，也不会影响其他的工作。

5. 参考内容

[1] wawlian: 网络爬虫基本原理(一)(二);

[2] guisu: 搜索引擎-网络爬虫;

[3] 《这就是搜索引擎:核心技术详解》。

作者：Poll的笔记

来源：http://www.cnblogs.com/maybe2030/p/4778134.html

—THE END—

文章推荐

☞数学 | 小学生如何诠释数学的线条美？

☞数学 | 从追女孩到找导弹，这就是数学的魅力！！

☞字节员工炸锅，薪资普降17%！

☞京东 | AI人才联合培养计划

☞世界天才大汇总

☞90后「V神」封神之路：4岁学编程，19岁创办以太坊，4年十亿身家！