**基于Hadoop分布式爬虫设计综述**

**摘要：**由于Internet规模不断扩大,包罗万象的信息资源被连接在一起,形成了一个广阔宏大的信息空间"在这个空间中,存在着海量的信息,如何快速高效和安全地让网络用户在如此浩瀚的信息海洋之中找到并获取自己所需的资源,是当前互联网发展的最大挑战之一。如今,云计算已成为当前的重要趋势之一。本文主要阐述在Hadoop分布式文件系统HDFS以及分布式计算框架MapReduce的基础上开发的分布式搜索引擎 的爬虫设计相关技术、原理、流程图。

**关键词：**云计算 分布式爬虫 Hadoop 搜索引擎

**1 引言**

随着搜索引擎的发展,搜索引擎所采用的技术也随之变得丰富和多样化,能够适应不同搜索用户以及不同搜索目的的需要。目前,搜索引擎的性能指标主要有三个:首先考虑的是规模的大小,只有规模达到一定的数量级,用户搜索结果的符合度才能够达到满足不同用户的需求程度;其次是性能,搜索引擎的网络蜘蛛必须在一个较短的时间内完成对目标网络的信息搜索,同时,能够在用户可容忍的时间段内,完成搜索结果的反馈;最后是搜索的质量,能够去掉信息重复的网页,对一些无用信息进行过滤,能够准确返回用户想要的结果。

如何从庞大的资料库中找到正确的资料,被公认为是下一代搜索技术的竞争要点"要对海量的信息进行检索,单单依靠单台计算机的处理能力远远不够,即使硬件的发展速度很快,但是根本赶不上信息的增长速度。而若采用集群计算机实现,虽然可以解决处理速度问题,但由于从网络的整体上看,该集群仍是一个结点,会严重受制于网络带宽,因此,需要采用多台计算机进行分布式协同处理"。

分布式搜索引擎是通过网络把大范围的分布、异构数据集联合起来,形成一个逻辑整体,为用户提供分布式的信息检索服务。同传统搜索引擎相比,分布式搜索引擎有以下优点:

1)各检索服务器之间协同工作,每个服务器只搜索自身自治区域内的信息资源,彼此之间只传递搜索结果信息,加快了检索速度,减轻网络及各站点的负担;

2)与网络资源本身的分布式特性相适应,增加搜索服务器方便,有良好的可扩展性;

3)索引信息化分到各个数据库中,使得各索引数据库规模小,易于管理,缩短查询响应时间。

当今，大型网站的用户多，参与度广。因此,如何有效地为如此巨大的用户群体服务,让他们参与时能够享受方便、快捷的服务,成为这些网站不得不解决的一个问题。而与此同时,凭借Google文件系统搭建起来Google服务器群,为Google提供强大的搜索速度与处理能力。于是,如何有效利用这些这种技术,为更多的企业或个人提供强大的计算能力与多种多样的服务,就是像Google这样拥有巨大服务器资源的企业在考虑的问题。

正是因为一方对计算能力的需求,而另一方能够提供这样的计算能力,云计算就应运而生。wikipedia关于云计算的定义为云计算是网格计算下的一种新的标签,它使用公用计算或其它方法来共享计算资源。云计算是依靠本机服务器或个人设备来处理用户应用程序之外的另一种选择。目前,包括Google、微软、IBM、Amazon、EMC和惠普在内的许多IT业巨头都宣布要在云计算上进行重点研究,也有了一些供企业使用的云计算案例。

目前,最著名的云计算基础设施是由Google提出来的。Google使用的云计算基础设施模式**[1]**包括四个相互独立又紧密结合在一起的系统,其包括Google建立在集群之上的文件系统GFs(GooogleFilesystem)**[2]**,针对Google应用程序的特点提出的MapReduce**[3]**编程模式,分布式的锁机制Chubby以及Google开发的模型简化的大规模分布式数据库BigTable**[4]**。

本文就是在Hadoop云计算平台的基础上完成的,阐述了在Hadoop分布式平台的基础上搭建分布式爬虫的相关研究。

**2 分布式爬虫技术背景**

**2.1 云计算**

云计算是一种全新的网络服务方式,其将传统的以桌面为核心的任务处理转变为以网络为核心的任务处理,它利用网络实现自己想要完成的一切处理任务,使网络成为传递服务、计算力和信息的综合连接,真正实现按需计算、多人协作。其基本原理为:利用非本地或远程服务器(集群)的分布式计算机,为互联网用户提供服务(计算、存储、软硬件等服务),这使得用户可以将资源切换到需要的应用上,根据需求访问计算机和存储系统,从而降低成本。云计算真正实现了按需计算,从而有效地提高了对软硬件资源的利用效。

通常,云计算(Cloudcomputing)是分布式处理(Distributed Computing)、并行处理（Parallel ComPuting）和网格计算(Gridcomputing)的改进处理,其前身是利用并行计算解决大型问题的网格计算和将计算资源作为可计量的服务而提供的公用计算。

**2.2 Hadoop 分布式平台结构**

Hadoop是大名鼎鼎的Lucene旗下的子项目,它原先是Nutch项目的组成部分,于2006年初从Nutch中分离出来成为一个独立的项目。Hadoop其实并非一个单纯用于存储的分布式文件系统,而是一个被设计用来在由普通硬件设备组成的大型集群上执行分布式应用的框架(framework)。

与前面提到的Google框架类似,Hadoop分布式平台结构包括两部分:

(l)分布式文件系统HDFS(Hadoop Distributed File System):用来在各个计算节点上存储数据,并提供了对数据读写的高吞吐率和容错性;

(2)类似于Google的Map/Reduce计算框架,它能够把应用程序分割成许多很小的工作单元,每个单元可以在任何集群节点上执行或重复执行。可见,Map/Reduce是一种简化的分布式编程模式,以让程序可以自动在普通机器组成的集群中以并行方式分布执行。

因此,Hadoop的目标是为开发分布式应用提供一个框架,而不是像OpenAFS,Coda那样为存储提供一个分布式文件系统。搜索引擎就是一种典型的分布式程序,Nuteh就是基于Hadoop开发的。基于Hadoop的分布式计算框架如下:

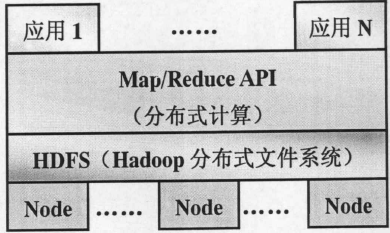


图2.1 Hadoop云技术结构

即,用户首先利用分布式文件系统HDFS将不同节点上的计算机祸合起来,给用户和应用程序提供一个共同的接口和界面,然后可利MapReduce计算框架,进行分布式计算,将一个任务“分解和结果汇总”,以在多台节点上运行,从而实现分布式编程。可见,Hadoop提供了一个分布式计算框架,就如同Java程序员可以不用考虑内存泄漏一样,Map/Reduce API也让程序员不需要关心海量数据如何被分配到多台机器上,不需要考虑机器失效的处理,不需要考虑各节点间如何协同操作共同完成任务,其简化了程序员的负担,以让不具备分布式系统经验的程序员,能够轻松地进行分布式编程。

**2.3 网络爬虫原理**

**2.3.1 搜索引擎基本技术**

随着因特网的迅猛发展、WEB信息的增加,用户要在信息海洋里查找信息,就象大海捞针一样,搜索引擎技术恰好解决了这一难题(它可以为用户提供信息检索服务)。目前,搜索引擎技术正成为计算机工业界和学术界争相研究、开发的对象。

搜索引擎以一定的策略在互联网中搜集、发现信息,对信息进行理解、提取、组织和处理,并为用户提供检索服务,从而起到信息导航的目的。按照信息搜集方法和服务提供方式的不同,搜索引擎系统可以分为三大类:目录式搜索引擎、机器人搜索引擎和元搜索引擎**[5]**。

一个搜索引擎由爬虫、索引器、检索器和用户接口等四个部分组成：

(l)爬虫(抓取模块):其功能是在互联网中漫游,发现和搜集信息。它常常是一个计算机程序,日夜不停地运行。爬虫的实现常常用分布式、并行计算技术,以提高信息发现和更新的速度。

(2)索引器:其功能是理解爬虫所搜索的信息,从中抽取出索引项,用于表示文档以及生成文档库的索引表。索引表一般使用某种形式的倒排表(InversinnList),即由索引项查找相应的文档。索引表也可能要记录索引项在文档中出现的位置,以便检索器计算索引项之间的相邻或接近关系(Proximity)。

(3)检索器:其功能是根据用户的查询在索引库中快速检出文档,进行文档与查询的相关度评价,对将要输出的结果进行排序,并实现某种用户相关性反馈机制。检索器常用的信息检索模型有集合理论模型、代数模型、概率模型和混合模型四种。

(4)用户接口:其作用是输入用户查询、显示查询结果、提供用户相关性反馈机制。主要的目的是方便用户使用搜索引擎,高效率、多方式地从搜索引擎中得到有效、及时的信息。用户接口的设计和实现使用人机交互的理论和方法,以充分适应人类的思维习惯。

**2.3.2 网络爬虫基本原理**

网络爬虫(Spider),其定义有广义和狭义之分。狭义上指遵循标准的http协议利用超链接和W七b文档检索的方法遍历万维网信息空间的软件程序;而广义的网络爬虫则是所有能遵循http协议检索Web文档的软件**[6]**。

Spider是一个功能很强的自动提取网页的程序,它为搜索引擎从万维网上下载网页,是搜索引擎的重要组成。Spider通过请求站点上的HTML文档访问某一站点。它遍历Web空间,不断从一个站点移动到另一个站点,自动建立索引,并加入到网页数据库中。网络爬虫进入某个超级文本时,它利用HTML语言的标记结构来搜索信息及获取指向其他超级文本的URL地址,可以完全不依赖用户干预实现网络上的自动爬行和搜索。

在抓取网页的时候,目前网络爬虫一般有两种策略:无主题搜索与基于某特定主体的专业智能搜索。

(1)无主题搜索主要包括:广度优先和深度优先。

广度优先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。深度优先是指网络爬虫会从起始页开始,一个链接一个链接跟踪下去,处理完这条线路之后再转入下一个起始页,继续跟踪链接。

(2)在专业智能搜索引擎中,网络爬虫的任务是获取Web页面和决定链接的访问顺序,它通常从一个“种子集”(如用户查询、种子链接或种子页面)出发,以迭代的方式访问页面和提取链接。搜索过程中,未访问的链接被暂存在一个称为“搜索前沿”(Spider Frontier)的队列中,网络爬虫根据搜索前沿中链接的“重要程度”决定下一个要访问的链接。如何评价和预测链接的“重要程度”(或称价值)是决定网络爬虫搜索策略的关键。

**2.3.3分布式网络爬虫Nutch**

Nutch是Apache基金会的一个开源项目,它原本是开源文件索引框架Lucene项目的一个子项目,后来渐渐发展成长为一个独立的开源项目。它基于Java开发,基于Lucene框架,提供Web网页爬虫功能。

**3 分布式爬虫设计**

**3.1 系统布局**

本文研究并设计的搜索引擎,是基于MapReduce的分布式搜索引擎,其分布式计算框架采用的是Hadoop**[7]**。整个系统分为应用层、搜索引擎核和云计算平台三个层次,系统架构如下图所示。

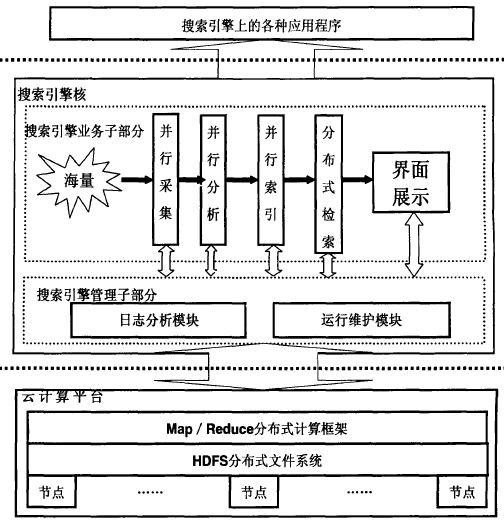


图3.1 分布式平台的系统构架

从上图可以看出,该系统以云计算平台Hadoop作为支撑,其包括分布式文件系统HDFS和分布式计MapReduce两个部分。搜索引擎核提供给各种应用程序接口。搜索引擎核操作着Hadoop的MapReduce任务,分为并行采集(分布式爬虫)、并行分析、并行索引、并行检索以及界面展示五大模块。其中,并行索引和并行检索由系统调用Lucene的核心类来完成,而本文的主要工作集中在并行采集模块上,即分布式爬虫的实现上。

**3.2 网络爬虫采集原理**

网络爬虫就是能够访问网络资源,以抓取网页信息。初始时,只需要提极少量的起始URL,网页抓取器就能够按照一定的规则沿着网页上的超级链接在网络上漫游,收集资源信息,直至遍历整个网站。如下是网络爬虫模块内部结构所示,其通过一系列的子操作完成。所有的子操作都基于MapReduce计算模型来完成。

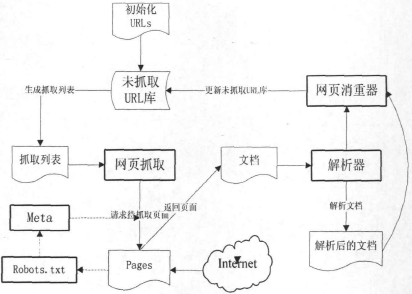


图3.2 网络爬虫内部结构

上图中的子操作分别为:

(l)建立初始化URL集,即构建“未抓取URL库”。初始URL集的建立有两种方式:超链接和用户提交。超链接是指机器人程序根据网页链接到其它网页中的超链接,从少数几个网页开始,连到其他网页的过程。用户提交是指在实际运行中,爬虫不可能爬遍所有站点,为此,用户可向搜索引擎进行提交,要求收录,搜索引擎经过核查之后,便将该网站加入到URL集合中,进行抓取。

在构建“未抓取URL库”的时候,会进行如下工作:

1）将URL集合进行格式化和过滤,消除其中的非法URL,并设定URL状态,并按照一定方法进行初始化分值;

2）将URL进行合并,消除重复的URL入口;

3）将URL及其状态、分值存入“未抓取URL库”。

(2)通过“未抓取URL库”生成“待抓取URL队列”,供抓取器进行抓取。在生成抓取队列的时候,会做如下工作:

1）从“未抓取URL库”中将URL取出并进行过滤;

2）对URL进行排序,通过域名、链接数和一种Hash算法综合进行降序排列,以使指向同一个主机上Web资源的UR此被分配到同一个抓取队列中,这样可以使得相应的MapReduce任务分配给一台机器进行执行,以防过

多的机器同时对一个主机进行抓取造成主机负担过重。

(3)抓取网页。抓取器根据抓取列表到互联网上进行相应网页的抓取,在抓取过程中遵守REP(Robots Exelusion Protocol)协议。抓取的时候,会根据robots.txt以及网页的Meta信息判断哪些是服务器定义不能索引和访问的,哪些是网页作者定义不能访问的,然后只访问能够索引的页面,同时将不能访问的URL设置成需要更新的URL。抓取的内容保存到一个临时库中。抓取过程中,页面的URL地址可能因为链接发生改变,从而需要更新URL地址;另外采用多线程的方式进行抓取,从而提高抓取速度。

(4)文档的解析。对应不同的文档调用不同的解析器,这通过Nutch的插件机制来完成。解析器对文档进行解析后,将生成的text和data保存到HDFS文件系统中。

(5)网页消重,并更新“未抓取URL库”。这里采用一部分网页解析的功能,将抓取的链接信息等,插入“未抓取URL库”和“原始网页库”中,同时消除重复的抓取内容。这样通过更新“未抓取URL库”,增加新的URL为下一轮抓取做准备。

**3.3 爬虫的基本流程设计**

本爬虫的基本需求是:对于用户给定种子URL集、抓取深度depth(即层数,种子URL在第1层,其链接URL为第2层,依次类推)、每层下载的URL数量topN等,完成网页抓取。针对该需求,本系统设置的爬虫基本流程如下:

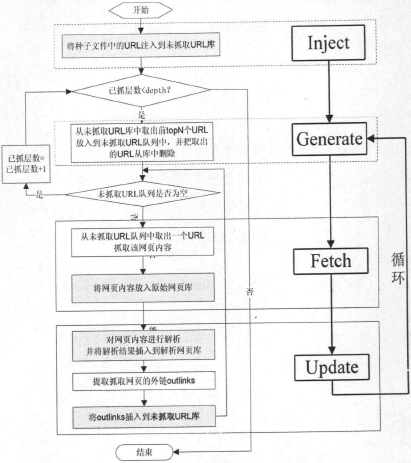


图3.3 分布式爬虫的基本流程图

该流程具体说明如下:

(1)用户将最开始的初始URL放到种子文件中,爬虫程序将该种子文件中的URL注入到“未抓取URL库”中;同时设置“已抓层次”为0。

(2)判断“已抓层数”与“要求抓取层次depth”的关系。当已抓层数小于要求抓取层次depth时,进入到(3);否则程序流程结束。

(3)从“未抓取URL库”中取出前topN个URL放入“未抓取URL队列”中,同时在“未抓取URL库”中删除这些URL。

(4)判断“未抓取URL队列”是否为空,若是,说明该层的topN个URL均已抓取完毕,爬虫开始下一层抓取,程序进入(2);若否,说明该层URL并未完全抓取,程序进入到(5)。

(5)从“未抓取URL队列”中取出一个URL,利用HTTP协议完成该URL所对应网页的抓取,并将网页内容放入“已抓取网页库”。

(6)对网页内容进行解析,提炼出所感兴趣的信息,并将这些数据插入到“解析网页库”中。

(7)将所抓取网页中的外链outlinks加入到“未抓取URL库”中。此时,一个网页的抓取流程完成,程序进入(4),进行“未抓取URL队列”中剩余URL的抓取。

从该流程图可以看出,该爬虫(Crawler)的基本过程可以分成四个独立的模块:

(1)Inject模块:根据用户所提供的初始种子URL名单,添加数据下载的入口链接,其首先读取用户给定的纯文本格式文件,获取URL列表,作为入口地址添加到“未抓取URL库”中。

(2)Generate模块:从“未抓取URL库”中获取下一步将要抓取的URL,将其放入“未抓取URL队列fetchlist”中。该模块的功能还包括对将要抓取的URL进行规范化,如对非法URL进行识别和对URL的长度进行限

制等,其也可用来完成对将要抓取URL的过滤功能,如对某些域名的限制抓取等。

(3)Fetch模块:此为爬虫的关键部分,按照“未抓取URL库”中的URL利用HTTP协议访问互联网,获取网页的具体数据,并将其存放到“原始网页库”中。

(4)Update模块:该模块与搜索引擎中的解析模块进行协作,但为了提高速度,Update模块仅调用解析模块中最简单的“链接分析”部分,已从原始网页中解析出链接outlinks。update模块将链接outlinks添加到“未抓取URL库”中,从而完成未抓取URL的更新。这样,整个网络爬虫被分成四个有机模块:Inject、Generate、Fetch和Update。如上图所示,网络爬虫从Inject出发,循环执行“Generate、Fetch、update”以完成网页抓取,循环的次数由用户所提供的“抓取深度depth”控制。

**4 结束语**

本文主要对基于Hadoop分布式爬虫进行了简单的阐述，首先介绍了搜索引擎及云计算的现状。接下来对分布式爬虫相关技术背景进行了阐述，主要是云计算、Hadoop技术平台，最后就是网络爬虫原理进行简介。最后是对本文实现的搜索引擎的分布式爬虫设计简单的介绍了系 统布局、爬虫原理、爬虫的基本流程。

**参考文献**

[1]Luiz Andre Barroso,Jeffrey Dean,UrsHolzle.Web Seareh For A Planet:The Google Cluster Architechture.IEEE ComPuter Soeiety,2003.

[2]SanjayGhemawat,HowardGobioff,Shun-Tak Leung.The Google File System,SOSP.03,Oct.19一22,2003.

[3]JeffreyDean,Sanjay Ghemawat.MaPReduee:Simplified Data Proeessing on Large Clusters,OSDI,2004.

[4]Fay Chang,Jeffiey Dean,Sanjay Ghemawat,WilsonC.Hsieh,Deborah A.Wallach,Mike Burrows,Tushar Chandra,Andrew Fikes,Robert E.Grube.OSDI,2006.

[5]蒋建洪,主要分布式搜索引擎技术的研究.科学技术与工程,Vol.7 No.10 May 2007

[6]周德翰,李舟军,高性能网络爬虫:研究综述.计算机科学,VoL36,No.8 Aug 2009

[7]吴宝贵,丁振国,基于Map瓜educe的分布式搜索引擎研究.现代图书情报技术,vol.8 2007.