浙江理工大学学报，第 29 卷，第 3 期，2012 年 5 月
Journal of Zhejiang Sci-Tech University
Vol. 29, No. 3, May. 2012

文章编号：1673-3851 (2012) 03-0404-04

一种基于云环境的 PageRank 改进算法

李　辉，张华熊

（浙江理工大学信息学院，杭州 310018）

摘　要：如何快速获取 web 信息，是基于 web 结构的挖掘技术急需解决的问题。将 PageRank 算法与云计算平台相结合，利用云计算平台分布式计算的特点极大地提高了算法运行速度。针对 PageRank 算法中大量迭代的计算过程，提出信任度的概念，根据每次计算关注点的不同，通过灵活设定网页节点的信任度，对链接矩阵进行不同信任度下的裁枝，进而进一步提高算法效率。实验结果证明了本文算法的有效性。

关键词：结构挖掘；云计算；PageRank；矩阵裁枝；信任度

中图分类号：TP393.027　　**文献标识码**：A

0　引　言

随着互联网的迅速发展，web 正在逐渐变成一种新的数据源。人们越来越频繁地在网络上发布、获取信息。web 中汇集了海量的信息，但是 web 的信息面很广，每个信息面又涵盖了巨大的信息资源库。网页信息的这些特征使得我们想要在这个信息世界快速获得所需要的知识成为一个大的难题，更为基于 web 结构的挖掘[1]技术带来了发展的契机。

由于 web 信息具有没有固定结构、动态更新快等特征，所以想要准确地定位和查找所需要的信息是极其困难的，并且对 web 页面的分析和挖掘也绝不能仅仅局限于文本。网页和普通文本的不同点主要表现在 HTML 标签和网页之间的超链接。常识告诉我们，经过标签布局的网页中大号的字体、位置放在网页前面的或者中间的信息一定是作者强调的内容，并且会出现在网页中显眼的位置。于是许多著名的搜索引擎提出了基于页面链接和基于搜索主题的算法，并将其应用于网页的排序上。Sergey Brin 和 Lawrence Page 于 1998 年提出了 PageRank 算法[2]，应用在实际的搜索引擎中，并取得了很好的效果。

然而 web 信息的海量性导致的存储问题和计算效率问题随之而来，很多网络的应用越来越追求及时性，响应速度决定了网民的搜索体验。例如在一些涉及到安全问题的账户交易中，及时性和重要性就体现的更加明显。云计算[3]的出现很好地解决了这个问题。云计算是一种新型的商业计算模型，它将计算任务分布在大量计算机构成的资源池上，使各种应用系统能够根据需要获取计算力、存储空间及其他各种软件服务[4]。本文将 web 结构挖掘算法与云计算的具体实现 MapReduce 编程[5]框架结合起来，会极大地提高对高质量网页的检索效率。

1　云计算环境下的 PageRank 算法改进

通过结构挖掘 PageRank 算法计算网页质量值，其传统方法是将网页间的链接关系直接通过公式(1)计算。但由于 PageRank 算法需要多次的、大量的迭代，所以需要将算法放在并行系统下面实现。因此本文基于开源云计算基础架 Hadoop 将 PageRank 算法与 MapReduce 框架结合起来对算法的运行效率加以研究。

web 页面节点中的页面之间的链接关系多为 A 链接指向 B 链接，A 链接指向 C 链接，A 链接指向 D 链接……实际需要的数据要求把他们解析为 A 链接对应 B、C、D…，这个过程需要部署到云计算环

收稿日期：2011－11－14

基金项目：浙江省自然科学基金资助项目(Y1100656)

作者简介：李　辉(1988－)，女，河南民权人，硕士研究生，主要从事智能信息处理方面的研究。

境下完成。最后的形式为<key, value[list]>对的形式 key:A,value:[B,C,D……]。这个过程需要在并行化为一个 MapReduce 阶段实现,下面是 PageRank 算法的网页 PageRank 值的计算公式[6]:

$$PR(A)=(1-d)+d\left(\frac{PR(P_1)}{O(P_1)}+\cdots+\frac{PR(P_n)}{O(P_n)}\right) \tag{1}$$

式(1)中参数 d 为跳转系数,Google 给出 d 的参考值为 0.85,$O(P_1P_2P_3\cdots P_n)$为页面 $P_1P_2P_3\cdots P_n$ 的链出页面数,$P_1P_2P_3\ldots P_n$ 是链向 A 的网页,$PR(A)$,$PR(P_1)\cdots PR(P_n)$各自表示网页 $P_1P_2P_3\cdots P_n$ 页面的 PageRank 值。

在使用 MapReduce 的 Map 阶段,实现源头节点 A 与其所有的链出节点(B、C、D……)的对应关系,输出为<key:源头节点,value:singler=(所有链向源头节点的链出节点列表/所有链出节点的总数目)>。在 reduce 阶段中,对应每个源头节点 key:生成一个与源头节点对应的 singler,然后拿这些中间值发送给 MapReduce 框架存储在 HDFS 中。把与 key 对应的 list[singler]中每一项加权计算,然后代入公式(1)得到每个页面新的 PageRank 值。

从上述算法中可以看到基于云计算的 PageRank 算法存在诸多的不足,首先将链接表示的矩阵直接用于计算,在实际过程中产生了大量的空间浪费,尤其当某个节点的支持链接数目很低的情况下,更是在表示链接关系的邻接矩阵中浪费了大量的存储空间。如果能有效地压缩邻接矩阵所浪费的空间,会为后面运行算法时减少算法中间结果的内存占用量产生极大的益处。

针对这一情况,本文提出一种基于矩阵表示形式的改进算法,此算法将数学中矩阵的思想[7]应用到超链接对网页节点的支持度的计算中来,把每一个页面节点表示成矩阵中的一个元素,若存在数域 N 上的 $m\cdot n$ 阶矩阵 $\boldsymbol{M}$,矩阵元素为 a_{ij}($i=1,2,\cdots,m;j=1,2,\cdots n$),以"0"和"1"代表某个节点的链接是否为其它网页节点投了一票,即是否有链接链向其它节点。如图 1 中所给出的由超链接关系转化成的邻接矩阵。

$$\boldsymbol{M}=\begin{pmatrix}0&1&1&0&1&1&0&0\\1&0&1&0&1&0&1&1\\1&1&0&1&1&1&1&0\\1&0&1&0&0&0&1&0\\0&0&1&0&0&0&0&0\end{pmatrix}$$

1:2,3,5,6
2:1,3,5,7,8
3:1,2,4,5,6,7
4:1,3,7
5:3

图 1　节点的链接关系解析成对应的链接矩阵

每一行中的矩阵元素表示所有参与为节点投票的网页节点,每一行代表单独的网页节点。将网页节点链接关系解析成邻接矩阵,制定网页节点之间的关联规则,关联关系弱的就剔除该节点,符合自定义的关联规则就保留,参加下一次迭代计算(即考察其对下一次计算的贡献力大小)。且如果网页节点中的链接数已经小于最小支持的链接数的时候,不再进行关联规则的计算,直接剔除此网页节点。

把链接矩阵每行中支持该节点的所有链接数称为该节点的信任数,信任数除以节点对应的矩阵行中的所有关联节点的数目,得到的百分比为相应该节点的信任度:

信任度=信任数/所有的关联节点数目　　(2)

假定网页节点的最小的链接支持数(Min_link)为 2,则在下面的链接矩阵中的最小支持度(信任度)为 2/8=25%。最小支持度(信任度)大于 25%的节点都属于有贡献力的节点。而不满足这项条件的就要被裁掉,不再参加下一步计算的节点。将上述的链接对应关系的矩阵表示成:

$$\boldsymbol{M}=(\boldsymbol{M}_1,\boldsymbol{M}_2,\boldsymbol{M}_3\cdots\cdots\boldsymbol{M}_8)=\begin{pmatrix}\boldsymbol{L}_1\\\boldsymbol{L}_2\\\boldsymbol{L}_3\\\boldsymbol{L}_4\\\boldsymbol{L}_5\end{pmatrix}=\begin{pmatrix}0&1&1&0&1&1&0&0\\1&0&1&0&1&0&1&1\\1&1&0&1&1&1&1&0\\1&0&1&0&0&0&1&0\\0&0&1&0&0&0&0&0\end{pmatrix} \tag{3}$$

每列代表 $\boldsymbol{M}_i$ 所有的链出节点,每行代表 $\boldsymbol{L}_i$ 每个节点所有的源头节点。

矩阵的行表示网页中的网页节点,列表示对此网页节点有贡献力的超链接网页,如果矩阵元素取值为"1",表示这个超链接网页存在链向该网页节点的链接。元素为"0",则表示不存在超链接网页链向该网页的链接项。同时可以从源头节点(行)和链出节点(列)两个方向根据定义的最小信任度的数值选择要被裁枝的行(列)。

式(3)中矩阵元素表示了链出节点和源头节点的对应关系,"1"表示 $\boldsymbol{M}_i$ 和 $\boldsymbol{L}_i$ 之间存在对应关系,反之,"0"表示 $\boldsymbol{M}_i$ 和 $\boldsymbol{L}_i$ 之间不存在对应关系。如在式(3)中,$\boldsymbol{M}_1$ 所表示的第一列只对应矩阵中 $\boldsymbol{L}_2$,$\boldsymbol{L}_3$,$\boldsymbol{L}_4$ 位置中的 3 个"1",以此类推。表 1 为依据式(3)得出的每个链出节点与其源头节点的对应关系。

表 1 链出节点与源头节点的对应关系

链出节点	源头节点	信任数
M_1	L_2,L_3,L_4	3
M_2	L_1,L_3	2
M_3	L_1,L_2,L_4,L_5	4
M_4	L_3	1
M_5	L_1,L_2,L_3	3
M_6	L_1,L_3	2
M_7	L_2,L_3,L_4	3
M_8	L_2	1

每行的网页节点的最小信任度数值计算如下：

$$Min\_link(L_1)=M_{11}+M_{12}+M_{13}+\cdots+M_{18}=4>Min\_link,$$

$$Min\_link(L_2)=M_{21}+M_{22}+M_{23}+\cdots+M_{28}=5>Min\_link。$$

同理可得出，$\boldsymbol{L}_3$，$\boldsymbol{L}_4$，$\boldsymbol{L}_5$ 的最小信任度的技术分别为 6、3、1，从而得到初始的节点集合 $\boldsymbol{C}_1=\{\boldsymbol{L}_1:4,\boldsymbol{L}_2:5,\boldsymbol{L}_3:6,\boldsymbol{L}_4:3,\boldsymbol{L}_5:1\}$。把这种思想应用到网页节点之间的关联规则中去，在每一次扫描矩阵的时候，对每一行的节点进行最小信任度的计数，并删除信任度在设定的最小信任度之下的节点，因为该节点不会对下一次生成新的节点产生作用。

裁枝算法的关联规则的步骤如下：

a)先把链接关系生成链接矩阵，计算每个节点的信任度值，并形成集合 C_1；

b)最后根据压缩后的矩阵生成网页节点的支持网页之间的关联规则：

①对于大于最小信任度的集合，生成其项数为 2 的所有非空子集 Item；

②对于生成的所有符合要求的非空子集 Item，如果对于 Item 中的的每个非空子集，剔除 Item[1] 与 Item[2]中源头节点和各自链出节点不满足最小信任度值的项；

③if $NUM\{Item[1]\bigcap Item[2]\}\geqslant Min\_link$，则关联规则的输出为 Out_link=>$Item[1]\bigcap Item[2]$，将输出 Out_link 中的每项做同样的计算，删除掉小于 Min_link 的列，其中 Min_link 为最小信任度值；

c)重复上述步骤，直到链接矩阵中所有两两源头节点的向量之间的信任度都大于 Min_link；

根据式(2)中信任度的定义，令最小信任度为 25%，即每个节点至少有数目为 2 的支持度，在$\{\boldsymbol{L}_2,\boldsymbol{L}_3,\boldsymbol{L}_4\}$的所有项数为 2 的子集中(即两个节点之间的互链接数目为 2)都满足最小信任数为 2，则经过裁枝后的节点链接$\{\boldsymbol{L}_2,\boldsymbol{L}_3,\boldsymbol{L}_4\}$是符合最小信任数的要求的。

邻接矩阵在不断地压缩，这样在每一次删除后，就减少了对生成下一层向量结果集达不到信任度的项，这样就使整个链接矩阵不断地变小，从而提高了算法的效率。

改进后的算法，裁减掉了对下次迭代贡献力不强的网页节点，不仅减少了中间结果的产生，更加提高了算法的效率，因为本文中每个矩阵元素代表的是网页中的链接，对矩阵的压缩也就是对链接节点的压缩，大大减少了算法的迭代次数和中间结果文件的大小，减少了对 HDFS[8] 的访问，减少了 I/O 的开销。

根据式(2)，信任度值由节点的信任数及其对应的所有关联节点的数目决定，在网页节点繁多的网络环境中，由文中所述的关联规则，虽然对节点的信息进行了部分的裁剪，但是整个 web 中页面的 PageRank 值计算是一个反复的迭代至收敛的过程，由于 PageRank 计算的本身存在黑洞效应，单个节点本身的入链数目与节点之间的互链接的数目相比来说，在计算上对 PageRank 值的影响较小，进而算法的失真较小，并且在算法的实际运行中减少了迭代的次数、节省了大量的存储空间。

2 实验过程及分析

2.1 实验环境的搭建

OS：Linux 操作系统。

CPU：Pentium(R) Dual-core CPU 2.5 GHz。

Memory：2.00 GB。

云集群在同一个局域网中搭建[9]，其中任选一台机器作为整个作业的 Namenode 和 JobTracker，另外 9 台作为 Datanode 和 TaskTracker，在配置文件中对 Namenode 和 Datanode 的 IP 分别进行配置。

软件环境：JDK1.6 的支持、Eclipse Helios Service Release 2 开发环境。在每台机器节点上都需要安装。

整个环境运行在 Linux 平台上，因为 Hadoop 框架的使用需要 shell 命令的支持。如果系统是 Windows，则可以安装 Cygwin 来模拟 Linux 平台环境[10]。

2.2 实验数据集的准备

本文选取了斯坦福大学网络研究平台[11]提供的 web 图集数据集作为本文的实验数据集。如表 2 所示。

表 2　Stanford 大学提供的 web 数据集

数据集名称	链接类型	数据大小/M	节点数/个	链接边数/条	描述
Web-BerkStandford	直连	105	685 230	7 600 595	来自 Berkeley 和 Stanford 的 web 图集
Web-Google	直连	71.8	875 713	5 105 039	来自 Google 的 web 图集
Web-Stanford	直连	31.3	281 903	2 312 497	来自 stanford. edu 的 web 图集

2.3　实验分析

将上述两种算法在 3 种不同规模的数据集进行实验仿真,得到算法迭代完成的时间。一种是经典的 PageRank 并行算法,另一种是基于矩阵裁枝思想进行改进的算法。在上述不同规模的数据集下,对比两者的计算效率,如表 3 所示。

表 3　算法执行时间对比

数据集边数	算法名称	
	基于 MapReduce 框架的 PageRank 算法/s	改进的基于矩阵裁枝思想的算法/s
2 312 497	15	11
5 105 039	28	23
7 600 595	38	30

从表 3 的实验数据中可以看出,改进算法由于对信任度较低的链接进行裁剪,从而减少了对计算下一级信任度影响达不到规定信任度的链接,大大降低了产生的中间结果对内存的消耗,以及每次访问 HDFS[8]产生的 I/O 消耗[12],也减少了计算每个链接节点的迭代次数,提高了算法的效率。

3　结　语

本文研究了基于云计算技术的 web 结构挖掘算法—PageRank 算法,将 PageRank 算法与 MapReduce 编程模型结合进行了算法的改进,在经典算法的基础上提出了能使算法效率得到很好增强的裁枝算法。创新点在于通过定义链接信任度的阈值,裁剪掉关联度弱的节点,从而提高算法的执行效率。

参考文献:

[1] 张　涛. Web 数据挖掘现状分析[J]. 科学之友, 2009, 17(1): 21-23.

[2] 蔡建超, 蔡　明. 搜索引擎 PageRank 算法研究[J]. 计算机应用与软件, 2008, 25(9): 59-60.

[3] 邓倩妮, 陈　全. 云计算及其关键技术[J]. 高性能计算发展与应用, 2009, 26(1): 2-6.

[4] 刘　鹏. 云计算的定义和特点[EB/OL]. (2009-05-23)[2011-10-20]. http://www.chinacloud.cn/.

[5] Lammel R. Google's MapReduce Programming Model-Revisited[J]. SCP, 2006(2): 22-24.

[6] Fu Hwai-hui, Dennis K J Lin, Tsai Hsien-Tang. Damping factor in Google page ranking[J]. Applied Stochastic Models Business and Industry, 2006(22): 431-444.

[7] 高　勋. 基于云计算的 web 结构挖掘算法研究[D]. 北京: 北京交通大学, 2010.

[8] Ibrahim S, Jin H, Cheng B, et al. Cloudlet: towards MapReduce implementation on Virtual machines[C]// Proceedings of 18th ACM International Symposium on High Performance Distributed Computing. New York: ACM Press, 2009: 65-66.

[9] Hadoop. Hadoop Cluster Setup[EB/OL]. (2010-09-12)[2011-10-20]. http://hadoop.apache.org/common/docs/r0.21.0/cluster setup.html.

[10] 李雪峰. 基于云计算环境的 web 数据挖掘算法研究[D]. 北京: 北京交通大学, 2010.

[11] Jure Leskovec. SNAP[EB/OL]. (2006-06-30)[2011-10-20]. http://snap.stanford.edu/data/index.html.

[12] 金松昌. 基于 HDFS 的多用户并行文件 IO 的设计与实现[D]. 北京: 国防科学技术大学, 2010.

An Improved Method of Pagerank Algorithm Based on Cloud Computing Environment

LI Hui, *ZHANG Hua-xiong*

(School of Informatics, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: It is urgent to solve the problem of how to obtain the information of web quickly for the technology of web structure mining. This paper applies cloud computing environment to web structure mining, making use of the characteristic of distributed computing based on cloud computing to cut down the runtime remarkably. Aiming at the iterative method for capacity computing of PageRank Algorithm, this paper brings about the concept of confidence level, according to the different attention point, through setting the confidence level of page flexibly, prunning the matrix to improve the effectiveness of algorithm. Finally, the experimental result proves to be effective.

Key words: structure mining; cloud computing; PageRank; matrix prunning; confidence level

(责任编辑: 陈和榜)