基于 Map/Reduce 的改进选择算法在云计算的 Web 数据挖掘中的研究*

方少卿[1], 周　剑[2], 张明新[2]

(1. 铜陵职业技术学院, 安徽 铜陵 244000; 2. 常熟理工学院, 江苏 常熟 215000)

摘　要: 针对目前在搜索方面的数据量大、搜索延迟的特点,提出了基于云计算的 Web 挖掘的搜索模型。采用提出的基于 Map/Reduce 模型的改进型算法,通过仿真实验验证了该算法的可行性,在一定程度上减少了搜索的代价,提高了搜索效率。

关键词: 云计算; Web 数据挖掘; Map/Reduce

中图分类号: TP391　　**文献标志码**: A　　**文章编号**: 1001-3695(2013)02-0377-03

doi:10.3969/j.issn.1001-3695.2013.02.015

Research of improvement selection algorithm in cloud-computing Web data mining based on Map/Reduce

FANG Shao-qing[1], ZHOU Jian[2], ZHANG Ming-xin[2]

(1. *Tongling Vocational Technical College*, *Tongling Anhui* 244000, *China*; 2. *Changshu Institute of Technology*, *Changshu Jiangsu* 215000, *China*)

Abstract: For the current data in the search volume, search delay characteristics, this paper presented a Web data mining method based on cloud computing search model, using the three selection algorithm based on Map/Reduce model. The simulation experiment proves the feasibility of the algorithm, it reduces the search cost, improves the efficiency of search to a certain extent.

Key words: cloud-computing; Web data mining; Map/Reduce

0　引言

随着 Internet 技术的迅猛发展,互联网上数据的更新速度呈几何级的速度增长,越来越多的新移动的存储技术广泛应用于网络平台上,如何发现有价值信息成为数据挖掘研究的热点。Web 数据挖掘是建立在对网络上的大量数据分析的基础上,利用数据挖掘算法有效地收集、选择和存储所感兴趣的信息。在日益增多的信息中发现新的概念以及之间的关系,实现信息处理的自动化,从而为商业模式过程中的数据收集、数据分析,作出对企业决策有用的数据分析是十分重要的。

1　Web 数据挖掘概述

Web 数据挖掘主要是以网络日志为研究对象,利用数据挖掘技术发现用户行为的潜在规律。目前,基于网络日志的用户行为模式研究已在网络安全、电子商务、远程教育等多个领域得到了广泛的应用,是当前的热点研究之一。Web 日志挖掘有如下的几种形式:采用关联规则分析,可获取用户页面访问行为间的关系;采用聚类分析,可将特征相似的用户或页面归并分组;采用分类分析,可对用户行为特征进行归类识别;采用频繁序列模式分析,可获取用户访问习惯。这些常用数据挖掘方法获取的用户行为模式,解决了页面自动导航、页面重要性评价以及改进网站设计、提高网站运营效益等问题。由于 Internet 本身分布广泛的特点,在 Internet 上产生的数据是海量的、分布的、异构的、动态的。这就造成了数据结构复杂度高,给 Web 数据挖掘带来了难度,从而对系统的计算能力提出更高的要求。

为了能够解决 Web 数据挖掘中的高性能、分布式的计算问题。国内外学者提出各种分布式数据挖掘平台,提高数据挖掘系统的处理能力的理论构想,但在实现过程中相对来说比较复杂。Cannataro 等人[3]提出了一种基于 Globus Toolkit 的分布式并行平台,这种平台带来了历史性的革命,它解决了传统数据挖掘不足的问题。近几年的研究主要集中在数据挖掘算法的实现和改进上,特别是 Web 数据挖掘处理的是海量数据,它所涉及和使用的挖掘算法比较复杂。在数据挖掘的过程中,挖掘算法需要多次与数据库进行交互,伴随着数据库中数据量的不断增加,数据的存储会带来新的问题,并且,随着各种不同类型的数据存储,这给挖掘算法提出了新的要求。本文将云计算

收稿日期: 2012-07-08; **修回日期**: 2012-08-15　**基金项目**: 国家自然科学基金资助项目(61173130);安徽省高校省级自然科学研究资助项目(KJ2012Z420)

作者简介:方少卿 (1965-),男,副教授,硕士,主要研究方向为 Web 数据挖掘(fangsq1965@tom.com);周剑(1981-),男,讲师,硕士,主要研究方向为数据库及系统集成、图形图像和视频信息检索、软件方法学、数据挖掘与知识发现;张明新(1962-),男,博导,中国计算机学会、电子学会高级会员,甘肃省计算机学会理事,计算机基础教学指导委员会委员,主要研究方向为数据库及系统集成、基于网络的智能化控制系统、图形图像和视频信息检索、智能信息处理、数据挖掘与知识发现.

的概念和数据挖掘服务结合起来,提供一种新的基于云计算的 Web 数据挖掘方法,可以解决 Internet 上分布广域的海量存储数据。

2 Map/Reduce 模型简述

Map/Reduce 是 Google 开发的一种并行分布式计算模型,已在搜索和处理海量数据领域得到了广泛的应用。Google 的云计算技术源于信息检索领域的海量数据的分析任务,通过对 Map/Reduce 以键/值的方式来分析处理数据集中的记录[1,2]。Map 是一个分解的过程,它将输入数据拆分为大量的数据片段,将每一个数据片段分配给一个计算机进行处理,从而具有分布式的特点;Reduce 与 Map 刚好相反,它是将分开的数据整合到一起,最后将汇总的结果进行输出。Map/Reduce 的执行由 master 和 worker 两种不同类型的节点负责,worker 负责数据处理;master 负责任务调度及不同节点之间的数据共享。其主要的流程[4,5]如图 1 所示。

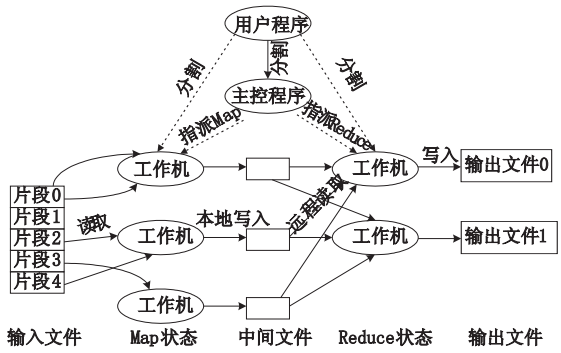

图1 Map/Reduce模型主要流程

a)用户在使用 Map/Reduce 函数时,先把输入文件分解成 M 个任务,每一个任务的大小为 16 ~ 64 MB 片段,进行不同的存储和备份。

b)在这些分派的任务中,主控程序 master 负责程序的调度与监控,是全局的统治者。它找出空闲的 worker 节点分配 M 个 Map 任务和 R 个 Reduce 任务到各计算节点。

c)将一个处理任务输入到一个 Map 工作机。它首先预处理输入的数据,并将分隔好的文件进行输入,处理后产生的关键字 key,传递给用户定义的 Map 函数。Map 函数产生的中间结果暂时缓冲到内存。

d)Map 函数产生的中间结果会在周期时间片内将中间结果自动写到本地磁盘,并需要将存储结果定时传递给主机程序 master,接着 master 负责把这些存储具体位置的信息传送给 Reduce 子任务的节点。

e)执行 Reduce 子任务的节点从 master 得到任务后,调用远程程序从 Map 工作机的本地硬盘上读取缓冲的中间数据。当 Reduce 工作机读到所有计算的中间数据,就使用中间关键字进行排序,相同关键字的信息就会汇总到一起。

f)Reduce 工作机根据每一个唯一中间关键字来遍历所有排序后的中间数据,并把关键字和相关的中间结果值集合传递给用户定义的 Reduce 函数。Reduce 函数将汇集各地的信息最终组合成一个输出文件,以便整体输出任务结果。

g)当所有 Map 和 Reduce 任务完成时,master 节点将 R 个 Reduce 结果返回给用户程序,用户程序将这些结果进行整合。

3 基于云计算的 Web 数据挖掘系统的设计与实现

3.1 概述

基于云计算的 Web 数据挖掘系统是在 Internet 上广域分布的海量数据发现数据模式和获取新的知识及规律,Map/Reduce 作为云计算的关键技术之一,其操作代表了一大类的数据处理操作方式[6]。Google 的搜索引擎选择了 Map/Reduce,并将其应用于云计算系统架构; Hadoop 在模仿 Google 时也采用了 Map/Reduce。其思想是:

a)在收集数据时,将分布在 Web 上的海量数据经过过滤、清洗、转换和合并,转换为半结构化的 XML 文件,分别保存到各个分布式文件的节点中。每一个文件复制的同时将其保存在不同的存储节点上,这样可以起到很好的备份作用,解决在 Web 数据挖掘中普遍存在的存储容量扩展和服务器突然发生故障所带来的数据丢失问题。

b)Master 负责整个数据挖掘的控制工作,服务节点(servicenode)主要完成节点之间查询的任务,创建完成之后向 master 传递信息,最后交由 master 进行汇总。

3.2 系统架构

在本文设计的基于云计算的 Web 数据系统中,节点分为四类:a)客户端节点,主要接收来自客户的请求,并根据客户提交的要求,返回结果给客户;b)主控节点,主要调度与协调算法节点之间的工作;c)算法节点,主要是负责根据主控节点针对不同的挖掘内容选择不同的计算节点;d)服务节点,主要协调算法节点与存储节点之间的关系[7]。相应地,基于云计算的 Web 数据挖掘系统分为处理层、存储器和挖掘算法层,如图 2 所示。

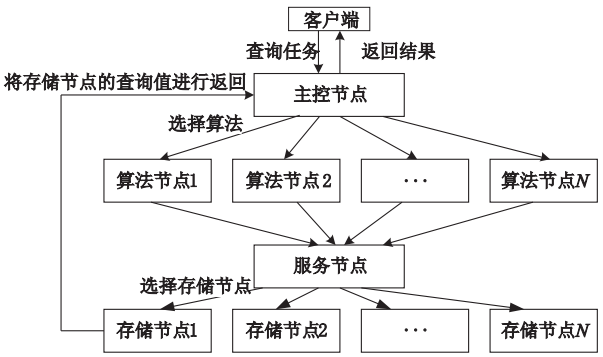

图2 基于云计算的Web系统架构

3.2.1 处理层

在 Web 数据挖掘子系统的设计中,任务调度是最重要的部分,所有 Web 挖掘由主控节点负责。Master 每隔一个周期向计算存储节点发出信号,计算存储节点是否空闲,如果计算存储节点空间就回复一个信息给 master,这样 master 就将该计算节点放入空闲列表中。Master 接到客户端的请求,根据客户端提交的任务选择相应的挖掘算法;然后向算法存储节点选择所需要的挖掘算法,算法节点会将选择的算法发送到原始数据所在的计算存储上,计算完成任务之后,就向 master 返回计算的文件数据块,然后返回给客户端。

3.2.2 数据存储层

在 Web 数据挖掘中,存储层主要的功能是将 Web 上收集到的文件解析成 XML 文件。为了防止系统瘫痪而引起数据瘫痪的问题,数据存储层还能够自动复制 XML 文件。该层主要

由数据服务节点和多个存储节点(dataNode)组成。单个存储节点可能会引起数据存储的信息丢失,所以,分布式文件系统将 XML 备份成多个以防止信息的丢失。XML 文件包括 IP 地址,通过 IP 地址可以对该 XML 文件进行访问和处理[8,9]。用户保存 XML 文件的过程如下:

a)客户端提交的任务经过主控节点变为半结构化的 XML,向数据存储节点提出保存节点,它将 XML 文件分隔成 16 ~64 MB 大小的子文件。

b)Master 查询计算存储节点是否有空闲,如果有空,就将 XML 的 IP 地址返回给客户端,并复制一份给其他的数据节点进行备用。

c)将 XML 文件的元数据写入元数据表中进行存储,以便用户可以根据结果查询子文件放入相应的数据存储节点。

3.2.3　挖掘算法层

该层存储了用于数据挖掘的各种算法。这些算法都是从传统的挖掘算法中改进过来的,这些改进的算法都是基于云计算平台的数据挖掘算法。在调用的过程中,master 获取客户端传来的任务请求,根据客户端的不同要求,采用不同的挖掘算法进行处理。本文基于 Map/Reduce 模型的三次选择算法。Map/Reduce 模型采用的是并行函数编程方式,每次 Map/Reduce 调度要首先将用户定义的 Map 函数和 Reduce 函数编译生成的结果发散到子节点上,这样就会导致任务效率的降低。为了提高效率,本文采用三次选择算法,可以有效地处理子系统的调度效率并降低网络子系统的负载。

3.3　基于云计算的 Web 数据挖掘算法

3.3.1　算法描述

数据挖掘的算法中包括关联规则、聚类、分类。其中关联规则在数据挖掘方面发挥着重要的作用,它主要包含两个步骤,即找出所有的频繁项集和在频繁项集上找出关联规则。频繁集是数据挖掘中最关键的部分,目前的方法是:首先找出频繁 1-项集 L_1,其次找出频繁 2-项集 L_2,直到某个 k 值满足 L_k 为空,最终算法结束。当求 L_k 的时候,通过 L_{k-1} 的自然连接生成候选集 C_k,然后检查 C_k 的每一个元素,满足用户自定义的最小支持度阈值就是 L_k 的元素。从 Web 上获取的数据更加广阔,所以验证 C_k 是算法中一个花费空间和时间的步骤[10]。本文采用基于 Map/Reduce 模型的三次选择算法,将云计算存储节点进行三次处理,这样确定全局频繁项集,可以提高挖掘效率。采用文献[11]中的处理方式,将预处理后的文档 d_i 作为输入的内容,数据块的集合〈client,doc-id1〉,经过发现特征词在文档中出现的频率,输出表示为〈t,〈n,f〉〉。经过 Map 函数处理之后,交给 Reduce 函数进行处理,将具有相同特征的内容进行合并,第二次进行处理的内容将分类后的特征词〈n_1,f_1〉〈n_2,f_2〉作为第三次 Map 函数的输入。同样经过第三次处理之后,比较用户通过 Web 浏览器提出的数据挖掘的服务请求,从而达到最小置信度的要求。数据经过 Map 函数将 Map$N$ 组输入(key,value)对应的中间结果(key,value),通过 Reduce 函数将相同的 key 进行合并,然后进行三次选择算法。数据经过第一次 Map/Reduce 处理的结果形成新的一轮(key_1,$value_1$),$key_1 = term_1$,$value_1$ = 〈n_1,f_1〉〈n_2,f_2〉…,作为输入数据再次经过 Map/Reduce 进行处理,key 是关键值标志,value 是局部频繁项集。在第二次进行处理时使用第一步得到的局部频繁项集,得到第二次(key_2,$value_2$),$key_2 = term_2$,$value_2$ = 〈n_1',f_1'〉〈n_2',f_2'〉…,再次生成进一步的局部频繁项集。经过同样的第三次步骤,得到全局频繁项集,从而提高挖掘处理效率。在处理算法的过程中,采用如下公式进行描述:

$$T_{ij} = F_{ij} \times e^{\log(n/t)}$$

其中:T_{ij}表示这一次的挖掘内容阈值;F_{ij}表示上一次挖掘的阈值,n 表示这次文档的数目特征词,t 表示这次文档中某关键词出现的频度;$e^{\log(n/t)}$表示上次挖掘内容的频繁项集的比值。

3.3.2　算法实现

本文采用的三次选择算法是基于 Map/Reduce 模型的算法,通过三次的选择算法得到全局频繁项集。算法实现描述如下:

```
第一次:Map
for all term1 ∈ client. doc do
frequency(t) = frequency(t) +1;
output(term1,record〈client,doc-id1,frequency(t)〉);
第一次: Reduce
input term, record(〈n1,f1〉〈n2,f2〉…)//输入数据
build list R
for all record〈n,f〉∈record(〈n1,f1〉〈n2,f2〉…)
append(R,〈n,f〉)
find(R)
output(term1, recordR)
第二次:Map
for all term2 ∈ term1 do;
for 〈clientn, frequency(t)〉∈recordR do h = h + 1
T(t,n) = frequency(t) * e^log(n/t)//进行选择
append(term2,T(〈n1,t1〉,〈n2,t2〉…))
第二次:Reduce
for〈client. doc1,T(t)〉∈T(W)
if T(W) > Y//规定的最小阈值
append(T - new〈n,T(t,n)〉);
find(T - new)
第三次:Map
for all term3 ∈ term2 do;
for〈clientn, frequency(t)〉∈recordR do h = h + 1
T(t,n) = frequency(t) * e^log(n/t) * e^log(n/t)//进行再次选择
append(term3,T(〈n1,t1〉,〈n2,t2〉…))
第三次:Reduce
for〈client. doc1,T(t)〉∈T(W)
if T(W) = Y//规定的最小阈值
find(0)//找到最小支持度阈值
```

通过进行三次 Map/Reduce 对结果进行计算,最后找到满足最小支持度的阈值,根据置信度阈值生成规则将得到的结果返回给客户端。

3.4　实验仿真与分析

在实验中,测试环境为局域网,Hadoop 平台,计算机配置为 CPU 酷睿双核 2.7 GHz,内存 2 GB DDR3,Linux 操作系统,其中一台是主节点服务器,其他三台是子节点服务器。首先所有的节点放在主节点上直接计算出执行时间;其次,采用本文所描述的算法,将数据集分解为 10 个子文件,保存在 10 个计算存储的节点上,采用三次选择算法,将存储节点传送到 2、4、6、8 这四个服务节点上,计算出时间,最后将四个存储节点上的数据备份到 2、4、6、8 这四个服务节点,计算出时间。通过这个实验,发现采用本文的算法能够提高各个节点频繁项集的时间,在运算过程中不会流失有效的关联规则,如图 3 所示。

(下转第 395 页)

的时间分别增加约 20% 和 10%,而 GT-MRTA 的性能所受影响较少(完成任务的时间约增加 3%)。随着通信失效率的增加,完成任务的时间相应增加,但总体来看,通信失效对 GT-MRTA 的性能影响较少,因此,其对通信失效具有很好的鲁棒性。

表 1 通信失效 10% 与无失效情况下 100 次完成任务的平均时间比

机器人	方法		
	集中	市场	博弈
2	1.139004	1.086873	1.045455
3	1.159251	1.079268	1.017442
4	1.290141	1.095137	1.024194
5	1.326019	1.095344	0.958246
6	/	1.088942	1.007092
7	/	1.088889	1.032995
8	/	1.065463	1.026178
9	/	1.088608	1.018519
10	/	1.092742	1.024523

表 2　通信失效 20% 与无失效情况下 100 次完成任务的平均时间之比

机器人	方法		
	集中	市场	博弈
2	1.168050	1.125483	1.098485
3	1.222482	1.142276	1.034884
4	1.363380	1.156448	1.052419
5	1.407524	1.137472	1.029228
6	/	1.168269	1.092199
7	/	1.116049	1.109137
8	/	1.119639	1.070681
9	/	1.111814	1.047619
10	/	1.139113	1.043597

5　结束语

本文对多机器人系统任务分配策略进行了形式化描述,为任务分配方案的求解提供了一种新的数学描述工具;针对多机器人系统中机器人决策之间的相互依存性,引入博弈论的思想分析了多机器人系统的任务分配问题,提出了一种基于博弈论的多机器人系统任务分配算法。实验表明该算法复杂度较低,计算量较小,鲁棒性较好,获得的任务分配方案质量较高。

参考文献:

[1] 张芳. 面向协作多移动机器人系统的再励学习方法研究[D]. 上海:上海交通大学,2002.

[2] 黎萍, 杨宜民. 多机器人系统任务分配的研究进展[J]. 计算机工程与应用,2008,44(17):201-205.

[3] 范如国, 韩民春. 博弈论[M]. 武汉:武汉大学出版社,2006.

[4] LI Ping, YANG Yi-min, LIAN Jia-le. Layered task allocation in multi-robot systems[C]//Proc of Global Congress on Intelligent Systems. 2009:62-67.

[5] SHAPLEY L. Stochastic games[C]//Proc of National Academy of Sciences of the United States of America. 1953:1095-1100.

[6] VELOSO M, STONE P. Individual and collaborative behaviors in a team of homogeneous robotic soccer agents[C]//Proc of ICMAS. 1998:309-316.

[7] DIAS M B. TraderBots: a new paradigm for robust and efficient multi-robot coordination in dynamic environment[D]. Pittsburgh, Pennsylvania: Robotics Institute, Carnegie Mellon University,2004.

[8] DIAS M B, STENTZ A. A market approach to multirobot coordination, CMU-RI-TR-01-26[R]. Pittsburgh, Pennsylvania: Robotics Institute, Carnegie Mellon University,2001.

(上接第 379 页)

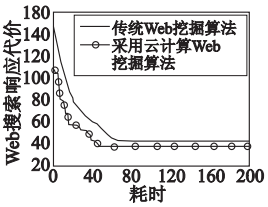

图3　Web搜索响应代价

4　结束语

在 Internet 网络日益发达的今天,更加强调数据挖掘的重要性。在数据容量越来越大的今天,如何能够更快更好地从 Web 数据挖掘中发现更多有价值的数据是现在研究的方向,特别是在云计算概念提出之后,可以将 Web 数据挖掘更好地与云计算结合起来。本文提出的三次选择算法可以更好地缩短 Web 挖掘的时间,提高挖掘效率,能够更好地面向商业应用。

参考文献:

[1] BARROSO L A, DEAN J, HOLZLE U. Web search for a planet: the Google cluster architecture[J]. IEEE Micro,2003,23(2):22-28.

[2] DEAN J, GHEMAWAT S. Map/Reduce advantages overparallel databases include storage-system independence and fine-grain fault tolerance for large jobs[J]. Communications of the ACM,2010,53(1):72-77.

[3] CANNATARO M, TALIA D, TRUNFIO P. Knowledge grid: high performance knowledge discovery service on the grid[C]//Proc of the 2nd International Workshop on Grid Computing. London: Springer-Verlag,2001:38-50.

[4] GILLICK D, FARIA A, De NERO J. Map/Reduce:distributed computing for machine learning[R]. 2006.

[5] JEFFREY D, SANJAY G. Map/Reduce: simplified data processing on large clusters[J]. Communications of the ACM,2008,51(1):107-113.

[6] 吴宝贵,丁振国. 基于 Map/Reduce 的分布式搜索引擎研究[J]. 现代图书情报技术,2007,2(8):52-55.

[7] 孙瑞锋,赵政文. 基于云计算的资源调度策略[J]. 航空计算技术, 2010,40(3):103-105.

[8] 席景科,闫大顺. Web 数据挖掘中数据集成问题的研究[J]. 计算机工程与设计,2006,27(8):1366-1368.

[9] 纪俊. 一种基于云计算的数据挖掘平台架构设计与实现[D]. 青岛:青岛大学,2009.

[10] 万至臻. 基于 Map/Reduce 模型的并行计算平台的设计与实现[D]. 杭州:浙江大学,2008.

[11] 陆小丽,何加铭. 基于 Map/Reduce 的索引数据云存储模型研究[J]. 宁波大学学报:理工版,2011,24(3):29-33.

[12] 王鹏. 云计算的关键技术与应用实例[M]. 北京:人民邮电出版社,2010.