

基于云计算技术的数据挖掘

应 毅¹, 任 凯^{2,3}, 刘正涛^{1,4}

(1 三江学院 计算机科学与工程学院, 江苏 南京 210012; 2 南京大学 金陵学院, 江苏 南京 210089;

3 中兴通讯股份有限公司 南京研发中心, 江苏 南京 210012; 4 南京航空航天大学 信息科学与技术学院, 江苏 南京 210016)

摘 要: 基于单一节点的数据挖掘系统在处理海量数据集时存在计算瓶颈, 针对该问题, 提出了一种基于云计算技术的数据挖掘方法: 将大数据集和挖掘任务分解到多台计算机上并行处理. 在对经典 Apriori 算法 MapReduce 化后, 建立了一个基于 Hadoop 开源框架的并行数据挖掘平台, 并通过对餐饮系统中点菜单的数据挖掘工作验证了该系统的有效性. 实验表明, 在集群中使用云计算技术处理大数据集, 可以明显提高数据挖掘的效率.

关键词: 数据挖掘; 云计算; Hadoop; MapReduce; Apriori 算法

中图分类号: TP39

文献标识码: A

文章编号: 1000-7180(2013)02-0161-04

Data Mining Based on Cloud-Computing Technology

YING Yi¹, REN Kai^{2,3}, LIU Zheng-tao^{1,4}

(1 College of Computer Science and Technology, Sanjiang University, Nanjing 210012, China;

2 Jinling College, Nanjing University, Nanjing 210089, China; 3 ZTE Corporation Nanjing R&D Center, Nanjing 210012, China;

4 College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: When process the massive data, there exists a calculation bottleneck in current data mining system based on single node. To solve these problems, proposed a cloud-computing technology-based data mining method. That is, the large data and mining tasks will be decomposed on multiple computers and be processed by parallel. We use open source project-Hadoop to establish a parallel data mining platform based on Apriori with MapReduce technology. It has been verified the effectiveness of system by data mining job of carte menu in catering industry. Experimental results show that, using cloud-computing technology to process large data in the cluster can significantly improve the efficiency of data mining.

Key words: data mining; cloud-computing; Hadoop; MapReduce; Apriori algorithm

1 引言

当前, 数据挖掘面临的主要挑战有两个方面: 一是怎样处理异构数据源. 计算机在各行业的急速发展使得除关系数据库以外的其它数据形式纷纷出现, 例如: 列数据库、文档数据库、Key/Value 系统、XML 数据库. 保证数据挖掘系统能有效的处理不同类型的数据源是至关重要的. 二是怎样处理海量数据. 随着时间的推移以及互联网、物联网的应用, 数据量呈指数级增长, 为了有效的处理大数据 (Big-Data), 数据挖掘算法必须是高效率的、可伸缩的.

为了解决高性能计算的问题, 国内外学者相继提出了基于集群、基于网格、基于 Agent 等各种分布式数据挖掘平台, 它们提高了数据挖掘系统的处理能力, 但都有自身难以解决的问题. 将云计算融入数据挖掘, 通过云计算的大规模并行计算, 可以解决海量数据挖掘的效率问题.

2 云计算技术

Hadoop 是一个被广泛使用的开源云计算系统^[1-4]. 使用 Hadoop, 可以方便的搭建私有云平台, 并可以根据自己的需要修改底层系统的代码. Ha-

收稿日期: 2012-05-30; 修回日期: 2012-06-22

基金项目: 中兴通讯全球云计算中心“云计算基础平台项目”和“餐饮服务系统”横向项目

doop 起源于开源网络搜索引擎 Nutch, 在借鉴了 Google 的 GFS^[5] 和 MapReduce 的设计思想之后, Hadoop 日渐成熟, 于 2008 年 1 月成为 Apache 的顶级开源项目, 并于 2011 年底发布 1.0.0 版本, 说明该产品已完全做好了应用于生产的准备. Hadoop 框架由 Java 语言编写, 运行在 Linux 操作系统之上, 由 Hadoop Common、HDFS、Hadoop MapReduce 核心部件组成. 目前, 包括 Yahoo、Intel、Facebook、《纽约时报》在内的很多公司在使用 Hadoop, 国内 Taobao 在使用 Hadoop 用于日志分析, 华东师范大学海量计算研究所在 288 核、40TB 的集群上部署了 Hadoop, 完成了多项有关海量数据方面的研究^[6].

3 基于云计算的数据挖掘平台

3.1 数据的备份

为保证数据的高可靠性, 云计算系统采用分布式存储的方式来存放数据, 即为同一份数据存储多个副本, 但在具体操作时并不是简单的冗余复制, 而是将文件按照一定的数据块大小进行切分, 再根据配置将每个文件块存储到数据节点上. 在大规模集群中, 节点失败是常见现象, 为此系统还必须提供心跳检测、错误隔离等措施来保证数据可靠性.

3.2 计算向存储迁移

毫无疑问, 数据挖掘的计算是基于大量数据的. 在分布式系统中, 稀缺的资源是网络带宽, 富裕的是

各节点的计算能力. 基于“计算迁移总是比数据迁移代价低”(Moving Computation is Cheaper than Moving Data)的思想, 云计算技术要求将计算节点和存储节点放在一起, 任务调度时, 也是尽量在保存相应输入文件块的设备上分配并执行任务, 这种方法使得大部分并行任务都在本地机器上读取输入数据, 有效的减少网络数据流量.

3.3 系统架构

基于以上思想, 本文设计了基于云计算技术的数据挖掘系统, 其整体结构如图 1 所示. 系统中的节点分为两类: 主控节点(MainCtrlNode)和工作节点(WorkerNode). MainCtrlNode 在系统中只有一个, 由 NameNode、数据仓库、JobTracker、数据挖掘算法库、SecondaryNameNode 组成. WorkerNode 在系统中可以有任意多个, 它由 DataNode、TaskTracker 组成, 负责实际的存储工作和计算工作.

NameNode 是分布式文件系统(HDFS)的主服务器, 管理文件系统的元数据, 它执行文件系统的命名空间操作, 比如打开、关闭、重命名, 还决定文件分块到 DataNode 的映射. DataNode 存储实际的数据, 负责处理客户的读写请求, 依照 NameNode 的命令, 执行数据块的创建、复制、删除等工作. 我们将数据挖掘中要用到的数据集上传至数据仓库, NameNode 会自动将数据文件分块并冗余存储到各个 DataNode. SecondaryNameNode 辅助 NameNode 处理映像文件和事务日志.

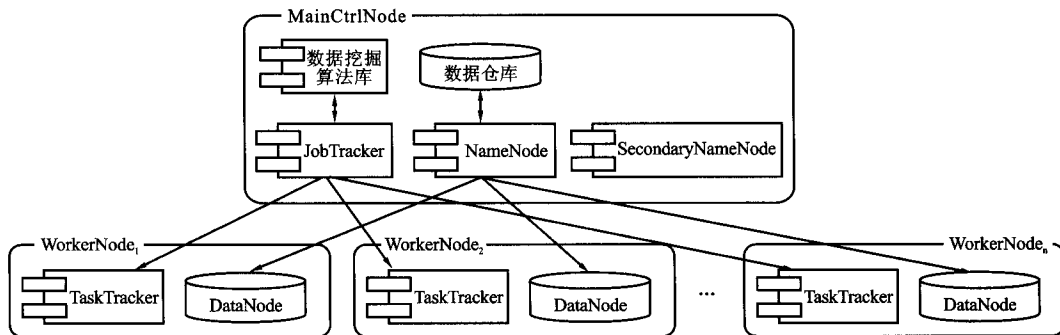


图 1 基于云计算技术的数据挖掘系统整体架构图

数据挖掘算法库负责存储数据挖掘所需的算法, 为了利用云计算并行执行的特点, 这些算法必须使用 MapReduce 编程模型重新设计实现(称为 MapReduce 化). 系统支持使用 Java、C++、Ruby 语言实现算法. 算法放入 MainCtrlNode 的算法库中, JobTracker 会自动将其分发至各 WorkerNode, 供 TaskTracker 执行时调用.

JobTracker 和 TaskTracker 采用 Master/

Slave 结构工作. JobTracker 负责调度与协调计算节点之间的工作进程, 跨 WorkerNode 分发任务, 监控任务, 重新执行已失败的任务. TaskTracker 根据应用要求结合本地数据来执行由 MainCtrlNode 分配的任务, 并把计算结果和状态信息返回给 MainCtrlNode.

上述结构具有很强的容错性, 当 WorkerNode 出现错误时, 只需要将该 WorkerNode 屏蔽在系统

外等待修复,并将该 WorkerNode 上执行的任务迁移到其他 WorkerNode 上重新执行,同时将该迁移信息通过 MainCtrlNode 发送给需要该节点处理结果的其它节点。

从功能上划分,NameNode、DataNode、数据仓库、SecondaryNameNode 形成了系统的存储部件,JobTracker、数据挖掘算法库、TaskTracker 形成了系统的计算部件。整个系统使用普通 PC 作为硬件设备,以 Hadoop 为基础,搭建在 Linux 操作系统之上。

3.4 Apriori-MR 算法

Apriori 算法是数据挖掘中发现频繁项集的经典算法。该算法使用逐层搜索的迭代方式,利用 $(k-1)$ 项集来产生 k 项集。具体包含连接步和剪枝步两个主要步骤^[7]。不失一般性,以下以 L_{k-1} 产生 L_k 为例,简单解释该算法的执行过程。

连接步:为找到 L_k ,将 L_{k-1} 与自身连接 ($L_{k-1} \bowtie L_{k-1}$),产生候选 K 项集 C_k 。

剪枝步: C_k 是 L_k 的超集,其中的各项集可能是也可能不是频繁项集,但所有的频繁 k 项集一定都包含在 C_k 中,显然,由于 C_k 中的候选项集很多,计算量非常大,基于 Apriori 性质(任何非频繁的 $(k-1)$ 项集都不是频繁 k 项集的子集),可以使用子集测试的方式压缩 C_k :如果候选 k 项集的 $(k-1)$ 项子集不在 L_{k-1} 中,则该候选也不可能是频繁的,可以从 C_k 中删除。之后扫描数据集,确定剪枝后的 C_k 中各候选项集的支持度计数,从而确定 L_k (所有支持度计数不小于最小支持度计数的候选项集就是属于 L_k 的频繁项集)。

Apriori 算法通过对数据集的多次扫描来发现所有的频繁项集,对海量数据的多次扫描以及产生的大量候选频繁项集将耗费大量时间和空间,这已成为 Apriori 算法的瓶颈。

为此,我们利用云计算并行执行的特点,对 Apriori 算法进行改进,用 MapReduce 编程模型重新设计实现,得到一个并行频繁集算法,称作基于 MapReduce 编程模型的 Apriori 算法 (Apriori base MapReduce, Apriori-MR)

不失一般性,仍以 L_{k-1} 产生 L_k 为例,解释 Apriori-MR 算法的执行过程。

Step0:将数据集、参数文件(存放最小支持度阈值)存放至 MainCtrlNode 的数据仓库中, Hadoop 框架会将其分发至各个 WorkerNode;

Step1:将 L_{k-1} 分发到各个 WorkerNode 上;

Step2:在 MainCtrlNode 上计算 C_k ($C_k = L_{k-1} \bowtie L_{k-1}$);

Step3:将 C_k 分片,若 WorkerNode 有 n 个,则将 C_k 分为 n 片,分发到相应的 WorkerNode 上,将分片后的 C_k 记作 C_{ki} ;

Step4:在各个 WorkerNode 上使用子集测试的方式并行计算 C_{ki} ,由于 L_{k-1} 已驻留到各个 WorkerNode 上,所以该计算不需要与 MainCtrlNode 交互;

Step5:扫描数据集,对剪枝后的 C_{ki} 中的每个候选计数,由于数据集已驻留到各个 WorkerNode 上,所以该计算也不需要与 MainCtrlNode 交互;

Step6:读取参数文件,将剪枝后的 C_{ki} 中各候选项集的支持度计数与最小支持度阈值进行比较,产生各自 WorkerNode 上的 L_k ,记作 L_{ki} ;

Step7:各 WorkerNode 将 L_{ki} 发送至 MainCtrlNode 汇总,得到本次迭代的 L_k ;

Step0 在整个 Apriori-MR 算法中只执行一次。Step1 是一次迭代的准备步骤,仅仅是一次文件的分发。Apriori-MR 算法的核心是 Step2~Step7,其中 Step2、Step7 在 MainCtrlNode 上执行,Step4~Step6 在各个 WorkerNode 上并行的执行,分别对应着剪枝、候选计数、与最小支持度阈值比较这 3 个最耗时的计算过程,在多个 WorkerNode 上并行执行能够大大缩短算法的执行时间。Apriori-MR 算法的执行过程如图 2 所示。

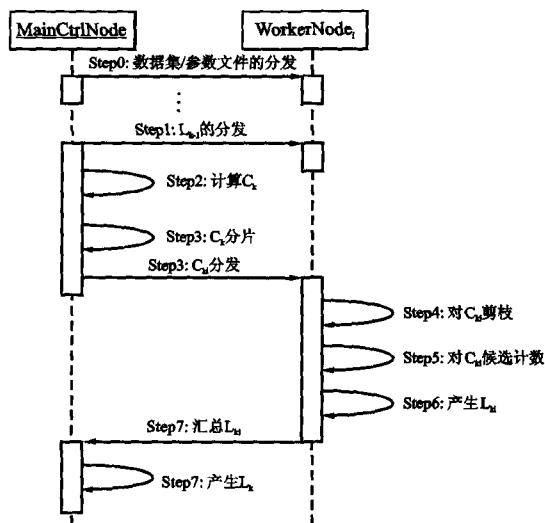


图 2 Apriori-MR 算法的时序图

4 应用实例

基于云计算技术的数据挖掘同传统数据挖掘的基本过程一致,分为数据准备、数据挖掘、结果表达

与解释 3 个阶段. 数据准备阶段又分为 3 个子步骤: 数据集成、数据选择、数据预处理. 系统由 4 台 PC 组成, 其中 1 台为 MainCtrlNode, 另 3 台为 WorkerNode. 我们以餐饮系统中的点菜单为数据集, 进行数据挖掘工作, 以期找到其中的关联规则.

我们收集了某连锁餐饮企业近 10 年的点菜单, 大约有 910000 次点菜记录, 每次点菜单记录了客人所点的菜品、酒水、点心、主食等内容.

由于时间跨度长, 所使用的技术前后不一致等原因, 这些记录有些保存在关系数据库中, 有些保存在文本文件中, 还有少部分保存在 XML 文件中. 在数据集成子阶段, 需要将这些文件合并在一起. 鉴于 Hadoop 系统在处理大文件的情况下更方便、更有效^[6], 我们决定将所有点菜单统一存储到文本文件中, 每 2 年的记录存入一个文件, 每个文件大约有 180000 行 (每一行即为一次点菜), 共 5 个文本文件.

在数据选择子阶段, 需要将噪声数据从数据文件中剔除, 比如服务费、开瓶费以及茶水、湿巾的费用, 这些信息对关联规则的挖掘结果意义不大.

在数据预处理子阶段, 主要做了以下 3 项工作: ①菜品名称的规范化: 由于行业自身特点, 菜品名称常有变化, 需要将其统一, 例如将“油爆花生米”、“盐水花生”统一记名为“花生米”; ②酒水饮料的归类: 不考虑品牌的因素, 将酒水饮料分类为啤酒、红酒、白酒、果汁类饮料、碳酸类饮料等; ③菜品编码: 为了提高处理效率, 将菜品名称从汉字转换为简单的、便于处理的字母与数字组合.

以上即为数据准备阶段所做的工作, 之后可以使用我们的系统进行数据挖掘工作. 由于使用了 Apriori-MR 算法, 执行效率得到了很大的提高.

最后得到的菜品关联规则中, 有一些是比较明显的, 是有经验的饭店管理者在工作中已经发现或是可以察觉到的, 例如: 花生米和白酒常会被同时选择; 红枣、糯米藕和木瓜被一同选中; 选择米饭的必然会选择时蔬. 另一些规则却是潜在的、不易察觉、甚至是匪夷所思的, 例如: 牛肉、乳鸽和红酒同时出现; 鸡翅和蛋挞被一同选中. 无论怎样, 挖掘出的这些关联规则在以后的菜肴推荐中会起到相当大的作用, 为餐饮企业带来切实的效益^[8].

5 结束语

本文对基于云计算技术的数据挖掘系统进行了研究, 在此基础上设计并实现了基于 MapReduce 编程模型的 Apriori 算法, 并在关联规则的挖掘方面做了测试. 与传统数据挖掘相比, 采用云计算模式成本低廉、容错性强、计算速度快、程序开发便捷、节点增加容易等. 云计算是数据挖掘中普遍适用较为理想的计算模式, 也是我们从海量数据中找到有用、可理解的知识的技术手段.

参考文献:

- [1] 胡侃, 夏绍玮. 基于大型数据仓库的数据采掘: 研究综述[J]. 软件学报, 1998(01): 54-64.
- [2] Dean Jeffrey, Ghemawat Sanjay. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2005, 51(1): 107-113.
- [3] 刘骞, 陈明. 基于改进的 Map/Reduce 及模式空间划分的数据挖掘[J]. 微电子学与计算机, 2011(08): 140-142.
- [4] 李建江, 崔健, 王聪, 等. MapReduce 并行编程模型研究综述[J]. 电子学报, 2011(11): 2635-2642.
- [5] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google File System[C]//Proceedings of the 19th ACM Symposium on Operating Systems Principles. New York: ACM Press, 2003: 29-43.
- [6] Tom White. Hadoop 权威指南[M]. 第二版. 北京: 清华大学出版社, 2011.
- [7] Jiawei Han, Micheline Kamber. 数据挖掘: 概念与技术[M]. 第二版. 北京: 机械工业出版社, 2007.
- [8] 索红军. 数据挖掘在商场决策支持中的应用研究[J]. 科学技术与工程, 2008(14): 3950-3952.

作者简介:

应 毅 男, (1979-), 硕士, 讲师. 研究方向为软件工程及数据库.

任 凯 女, (1979-), 硕士, 工程师. 研究方向为核心网及嵌入式系统.

刘正涛 男, (1975-), 博士研究生, 讲师. 研究方向为 Web 数据空间、Web 数据集成.