

云计算环境下海量分布式数据处理协同机制的研究

戴炳荣 宋俊典 钱俊玲

(上海计算机软件技术开发中心 上海 201112)

摘要 云计算环境下的数据管理和存储呈现出异构、分布式和动态等特点,为数据处理系统的设计和开发带来一定挑战,数据处理过程的协同是解决上述问题的有效手段。在协同计算模型分析的基础上,提出一种数据处理的协同机制,重点研究分布式数据的协同管理和任务的协同调度。实际的案例分析表明,该机制和方法可以大大提高并行数据处理的能力,为云计算环境下的数据处理平台的研发提供了一种可行的思路和方法。

关键词 协同计算 数据处理 协同机制 云计算

中图分类号 TP311 文献标识码 A DOI:10.3969/j.issn.1000-386x.2013.01.025

RESEARCH ON COOPERATION MECHANISM FOR MASS DISTRIBUTED DATA PROCESSING IN CLOUD COMPUTING ENVIRONMENT

Dai Bingrong Song Jundian Qian Junling

(Department of Software Service and Platform, Shanghai Development Center of Computer Software Technology, Shanghai 201112, China)

Abstract In cloud computing environment, data management and storage present the characteristics of heterogeneous, distributed and dynamic, which bring certain challenge for the design and development of data processing system. The cooperation of data processing is an effective means to solve these problems. Based on analysing the cooperation computing model, in the paper we propose a cooperation mechanism of data processing, and put the emphasis on the cooperation of distributed data management (Co-DDM) and the cooperation of task scheduling (Co-TS). Practical cases analyses demonstrate that this mechanism and method are able to improve the ability of parallel data processing greatly, it provides a feasible thought and approach for the research and development of data processing platform in cloud computing environment.

Keywords Cooperative computing Data processing Cooperation mechanism Cloud computing

0 引言

随着计算机和互联网技术的高速发展,信息呈现出爆炸式的增长模式。面对不断拓展的数据规模,如何有效地对海量信息进行存储管理、实时处理和挖掘分析,是当前数据处理领域面临的难点和重点^[1]。另一方面,在越来越广泛应用的云计算环境中,如何对异构、分布式和动态的数据进行处理,以满足不同的应用需求,是当前云计算应用领域研究的热点之一^[2]。

对云计算环境下的应用系统和平台分析可知,数据处理的方法和过程呈现出以下几个特点^[3]:1)数据处理的复杂度很高,表现在数据处理流程复杂、数据类型异构和数据算法繁多,需要通过一定的机制来协同完成特定的任务;2)数据对象的处理过程要求动态可扩展,需要根据应用需求动态地增减计算资源;3)基于云计算的编程模型要求系统的计算过程和任务调度向用户和编程人员透明,从而大幅提升系统的灵活性。因此,在云计算环境下引入任务协同调度和数据协同管理的技术和方法,是解决上述问题的有效手段。

本文结合协同计算理论和方法的研究,对云计算环境下海量数据处理的任务协同调度和数据协同管理进行了深入的研

究。针对数据处理对象、流程和任务的管理,提出了数据处理的协同模型,将其应用于海量数据处理平台的设计中并进行了验证分析,为云计算环境下的协同应用机制研究提供了一种新的思路和方法。

1 协同计算概述

协同计算是一个利用计算机技术、网络与通信技术、多媒体技术以及人机接口技术,将时间上分离、空间上分布而工作上又相互依赖的多个协作成员及其活动有机地组织起来,以共同完成某一项任务的分布式计算机环境,作为一门多学科交叉的技术,有着广阔应用前景^[4]。

1.1 协同模型分析

协同计算涉及到心理学、认知科学、社会学、组织科学等多个学科,其研究包括协同工作系统的建设、群体工作方式研究和支撑群体工作的相关技术的研究、应用系统的开发等部分,但是

收稿日期:2012-02-28。上海市科委基金项目(10510704300, 10dz1122600)。戴炳荣,硕士,主研领域:云计算,虚拟化,数据中心自动化。宋俊典,副研究员。钱俊玲,助理工程师。

整体上针对协作支持技术和协同工作技术两个基本方面^[5]。协作支持技术的研究包括通信技术、共享工作空间机制、共享信息机制、群体活动支持机制、同步机制、安全机制等;协同工作技术的研究主要包括协同工作模型、协同模式及支持协同工作相关技术等。

协作模型研究是协同计算的研究热点之一,包含群体成员进行协作、共同完成任务的模式,涉及群体成员间如何开展工作,协作时如何进行交互、操作协调、协作推进和结束协作等,主要有会话模型、会议模型、过程模型、活动模型和层次模型^[6]。

会话模型是交互双方沟通和动作协调的基础,通过特定的言语行为(如请求、许诺等)的执行来完成。会议模型是群体进行协同工作的另一种基本形式,参与者通过共享的信息空间彼此沟通,共同完成任务。过程模型将复杂的任务分成一个个小的操作步骤,活动模型对过程模型进一步扩充,将任务分成目标明确的子任务,定义子任务之间的关系,并通过任务的协同调度来完成任务。一般情况下,单一的模型不能满足对协同任务的协作方式和过程的描述,要采用多种模型混合,通过层次模型按不同层次加以描述。

1.2 协同计算的应用分析

在不同的应用环境下,协同计算的应用会有较大的差异,一般来说,协同计算的应用架构包含的基本组成元素主要有:成员角色、共享对象、协作活动和协作事件。成员角色描述群体成员在协同工作过程中所起的作用,需要依据不同的需求进行合理的成员角色划分;共享对象是在协作过程中各成员共同操作的对象;协作活动描述群体成员所进行的协作过程;协作事件则是协作进展和状态变化的指示,用于协调各成员的行为。

任务的协同调度和数据的协同管理是协同计算的关键之一,在基于协同计算应用系统的设计过程中,分层管理、灵活调度和易于扩充是系统架构设计的关键。架构的实现方式有集中式、分散式和混合式三种^[7]:

集中式架构实际上是一种客户/服务器架构。服务器集中负责整个应用系统的管理、控制和调度,以及所有与具体应用相关的程序逻辑、数据等,客户端指用户交互的输入输出。当用户要执行相关任务时,需要通过客户端向服务器发出请求,由服务器完成。集中式的协同应用架构实现简单、易维护,但协同应用对服务器有较强的依赖,很容易成为整个系统的瓶颈。

分散式架构中,协同控制的节点均处于同等地位,与系统相关的控制和管理模块分散在客户端。客户端较好的实现了系统的自治,其鲁棒性、灵活性、开放性和通用性都比较好,但数据的分布存储和操作的各自处理,使得全局的一致性较难维护,节点间的实时性同步以及用户的动态注册也较难实现。

混合式结构结合了前两种结构的优点,由服务器实现数据信息的一致性维护、用户管理和信息的转发与分发等功能,由客户端实现与用户的交互。这种结构充分利用了客户端的计算机资源,极大地减轻了服务器的工作压力,减轻了服务器的“瓶颈”效应,既能使系统具有较好的灵活性、鲁棒性和较短的响应时间,又保证了系统信息的一致性。

因此在实际应用系统的协同机制设计和开发中,一般在任务的协同调度中,采用分散式的架构设计,在数据的协同管理中采用集中式的管理方式,从而实现混合式的协同计算应用架构。

2 云计算环境下数据处理的协同机制

在云计算环境下,数据处理应用系统和平台呈现出分布式、动态、数据源异构等特点^[8]。结合协同模型和协同计算的应用分析可知,在进行海量数据处理系统和平台的设计过程中,必须结合对成员角色、共享对象、协作活动和协作事件等基础元素的分析,基于协同计算的应用框架,研究协同计算的应用架构,从而实现任务的协同调度和数据的协同管理等。

2.1 海量分布式数据处理应用架构

在云计算环境下海量数据处理的应用系统设计中,单一的协作模型和系统架构不能满足综合性的数据处理需求,必须考虑任务调度和数据管理两个层面的协同。成员角色和对象的协同管理、协作活动和协作事件的协同调度是海量数据处理应用架构的设计核心。在传统的计算机群体协作模型的基础上,通过总结和概括计算机支持的协同工作的结构特点和工作方式,提出了海量分布式数据处理的分层应用架构模型,主要分为应用接口层、调度执行层和数据管理层,如图1所示。

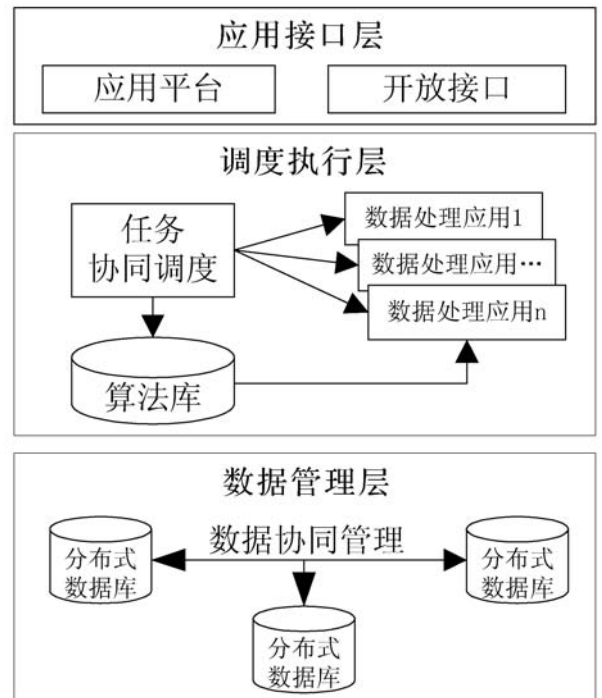


图1 海量分布式数据处理平台应用架构

由图1可知,海量分布式数据处理平台应用架构的实现方式采用了混合式的体系架构。在调度执行层,采用集中式的任务协同调度管理,根据规则和参数来调整数据处理引擎使用的数据、算法组合以及计算资源,对计算资源之间的数据交互、参数同步、任务分工进行协同。在数据管理层,对协同环境中业务数据和规则数据等进行分布式的存储,协同整个过程的数据冗余备份及容错处理,在数据处理过程中协同数据的获取提供给各数据处理应用,实现数据访问的负载均衡。

在云计算环境下的数据处理应用中,分布式数据处理系统的协同管理和任务调度的流程编排是协同机制的关键,其协同处理流程如图2所示。

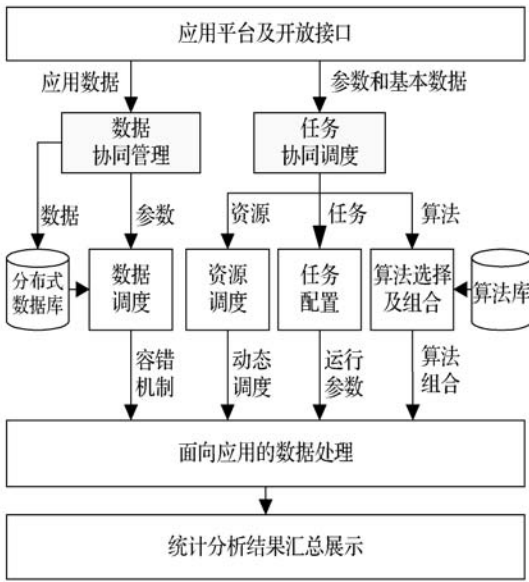


图2 协同管理和任务调度流程分析

数据协同管理的内容是分布式的数据库,其关键和核心是实现数据的分布式存储,以及分布式存储过程的容错机制。通过应用平台和开发接口,数据协同管理组件将应用数据和参数传给分布式数据库,进行协同保存,在容错机制的保障下,数据调度为数据处理应用提供分布式数据库中的数据。另一方面,任务协同调度应用采用集中式的协同管理机制,统一的分配资源、配置任务、选择和组合算法,从而实现面向应用数据处理的综合协同。

2.2 分布式数据的协同管理机制

分布式数据具有物理分布性、逻辑整体性和站点自治性三个基本特点。数据集中与自治相结合的控制机制、适当的数据冗余度和事务管理的分布性,是数据协同管理必须考虑的内容。

分布式数据的管理一般通过分布式数据库来实现。分布式数据库中的数据分别在不同的局部数据库中存储、由不同的DBMS进行管理、在不同的机器上运行,分布式数据库中每一个数据库服务器合作地维护全局数据库的一致性。对用户来说,逻辑上单个的分布式数据库实际上是由存储在多台计算机上的一组数据库组成,而用户对分布式数据库的访问是透明的,分布式数据库管理系统负责完成对数据的存储、查询调度以及整个数据库系统的事务。

在云计算应用的环境下,数据存储和管理都是分布式的,应用系统中的数据库服务器称为数据库结点,每一个数据库结点都具备数据存储、管理和协同作用,能够在每一个结点完成分布式数据对象存取、管理功能。分布式数据协同管理机制如图3所示。

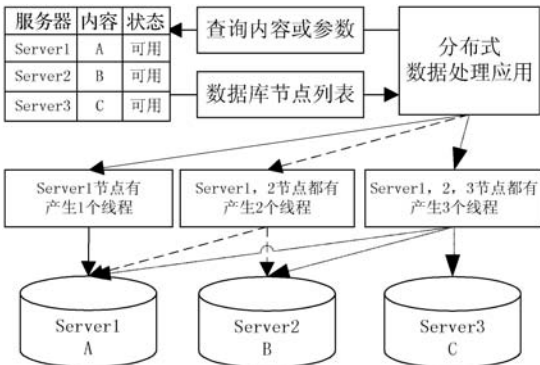


图3 分布式数据协同管理的实现机制分析

由图3可知,分布式数据协同管理的实现机制可以分为协同管理、执行和数据管理三个层次。协同管理层负责建立数据库服务器及其内容对应的映射表,实现数据的均衡存储。当分布式数据处理应用请求时,首先需要从内容和参数在映射表中查询数据节点,将获得的数据库节点列表信息返回给数据处理调度服务器。根据不同节点的分布式存储情况,数据处理调度服务器向查询到的节点发起数据处理任务,在执行层产生相应的线程,从数据管理层的数据库中查询数据。在执行查询过程中,当系统监控到某个节点失效,系统将自动选取相应备份节点继续执行该处理任务。

这种分层的分布式数据协同管理机制能够灵活地分割海量数据,对数据存储管理进行均衡存储、冗余备份,当有存储节点出现问题时能够通过相应备份节点的数据进行替代;同时,该机制也提升了海量数据的写入和读取速度,同一批数据经过分割存储到不同的节点,读取数据时,所有节点同步读取,解决了数据读取速度慢的瓶颈;通过协同管理层对节点的状态监控,能够实时了解节点健康状态,一旦出现离线情况,能够及时将任务交给其备份节点,系统的可靠性得到大大提升。

2.3 任务的协同调度机制

云计算环境下数据处理协同机制研究的另一个重点是任务的协同调度,任务协同调度是对协同计算中协作活动和协作事件基本组成元素的管理,任务的协同调度机制以任务的协同管理为核心,通过任务分配、算法选择、资源协调,生成包含动态调度、运行参数和算法组合的配置文件,作为数据处理应用的输入。任务协同调度机制和流程如图4所示。

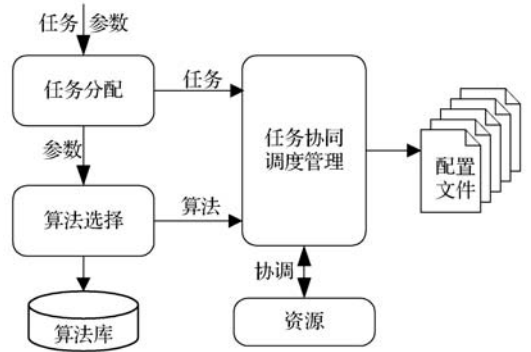


图4 动态任务协同调度的实现机制

用户通过提交的参数和基本数据由平台接口层接受,并提交给任务协同调度构件,任务分配模块负责接收这些参数,通过对请求数据的处理分析,形成系统需要执行的任务信息及算法参数,将算法参数交给算法选择模块进行算法选择,将任务信息交给任务协同调度管理模块进行综合调度排程。数据处理应用中执行的算法必须在任务协同调度管理组件中注册才能使用,算法选择模块接收到任务分配模块的参数后,将从算法库中进行匹配,并将算法的配置信息在任务协同调度管理模块中进行注册,并同步写入配置文件。配置文件中包含系统执行过程中得全局控制参数、系统任务的排程数据、任务执行过程中的算法信息、执行处理过程的计算节点信息等,配置文件通过共享白板的形式提供给各数据处理应用使用。

3 案例及分析

在制造业信息化领域,云计算技术为生产、物流、存储、销售

等方面的信息化应用带来革命性的突破。云计算技术与制造技术相结合构建的网络化制造系统,突破了空间地域对企业生产经营范围和方式的约束,为企业开展覆盖产品全生命周期的业务活动(如产品设计、制造、销售、采购、管理等)提供了便利^[9],同时在网络化的制造系统中也积累了海量数据,如何有效地对这些海量数据进行分析 and 处理,是海量数据处理平台协同机制的典型应用之一。

3.1 系统实现

本节结合某典型的制造业企业的应用需求,对现有的 ERP、SCM 系统的海量数据库进行挖掘分析,为企业的决策分析提供依据。

目前对分布式数据进行处理和计算时,主流的技术一般采用 Map/Reduce 或并行数据库技术来实现。Map/Reduce 是 Google 提出的一个软件架构,通过 Map(映射)和 Reduce(化简)的方式实现大规模数据集的并行运算^[10]。本文采用的方法是在数据库并行的技术上,引入并行协同的方式,结合基于工作流的任务协同调度,来实现并行数据库的数据处理,系统架构如图 5 所示。

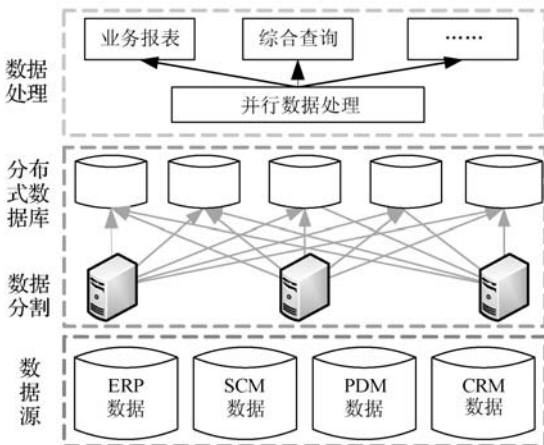


图5 实验系统架构图

由图 5 分析可知,数据源是企业的 ERP、CRM、PDM 以及 SCM 等业务系统,经过分布式数据协同管理将数据分割,存储到分布式数据库中,数据处理层以工作流为载体,通过并行的数据处理进行业务报表生成以及综合查询等。

在硬件部署方面,采用 9 台基于 Windows 2003 的服务器搭建了实验演示环境,其中包括一台应用服务器、两台数据协同管理服务器以及六台 SQL Server 2005 数据库服务器,部署结构如图 6 所示。

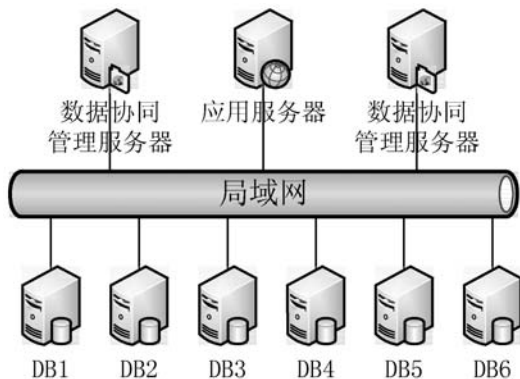


图6 实验系统部署结构图

3.2 结果分析

结合实际应用环境中的数据,模拟仿真应用系统的数据,实验数据的结构如图 7 所示,其中 T_Time 表示生产批次, T_Product 表示产品, P_Sales 表示产品销售明细, T_Store_History 表示出库历史记录。

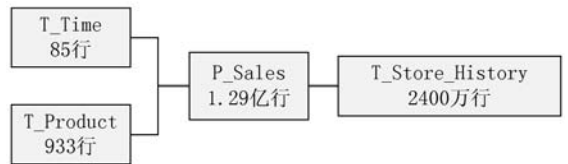


图7 实验数据关系

分别对从产品销售明细中统计总销售金额,从所有表中连接查询出每次出库中特定产品的生产批次,查询出含有某关键字的产品的销售明细,使用基于协同机制的并行计算方法,与普通单服务器的查询时间对比如表 1 所示。

表1 执行时间对比表

环境	单表求和	多表连接	模糊查询
单服务器系统	3'57''	23'47''	8'52''
数据处理平台	9''	75''	24''

从执行时间来看,在增加了 9 倍硬件资源的情况下,系统性能有了 20 倍左右的提升。从计算查询速度看,单表求和最快、对表连接最慢。从数据中看出,数据处理平台与单服务器系统的处理时间比也不尽相同,这与数据处理平台中各并行数据库中数据的关联程度以及数据的分布情况有关。

4 结语

本文结合云计算环境下应用系统的特点,提出海量分布式数据处理的应用框架,以协同计算应用架构的成员角色、共享对象、协作活动和协作事件等元素为对象,重点研究分布式数据管理机制和任务协同调度机制,并给出了数据协同管理模型和任务协同调度的方法流程。实际的应用案例和分析证明该机制可以大大提高云计算环境下海量数据处理的效率,在实际的应用环境中是有效可行的。

参考文献

- [1] 于戈,谷峪,鲍玉斌,等. 云计算环境下的大规模图数据处理技术[J]. 计算机学报,2011,34(10):1754-1765.
- [2] 程苗. 基于云计算的 Web 数据挖掘[J]. 计算机科学,2011,38(10):146-149.
- [3] 王鹏,孟丹,詹剑锋,等. 数据密集型计算编程模型研究进展[J]. 计算机研究与发展,2010,47(10):1993-2002.
- [4] Sang Woo Han, Jong Won Kim, Prinz W. A multi-agent-based management system for pervasive collaborative computing environment [C]//IEEE International Conference on Computing and Communications (PerCom 2009), 2009:1-6.
- [5] 屈正庚. 基于 CSCW 的协同设计系统的研究与应用[D]. 西安:西北大学,2011.
- [6] Han Xu, Cao Yongcun. The application of Computer-Supported Collaborative Technologies in web-based teachers Professional Development System[C]//The 5th International Conference on Computer Science and Education (ICCSE),2010,30(9):842-845.

在较低支持度下, FIUT-Stream 算法的时间开销都小于 DSTree 和 CPS-tree 算法, 并且随着支持度降低差距越来越大。

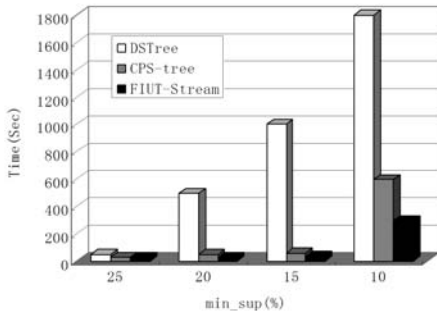


图4 在数据集 T40110D100K 下时间消耗的比较

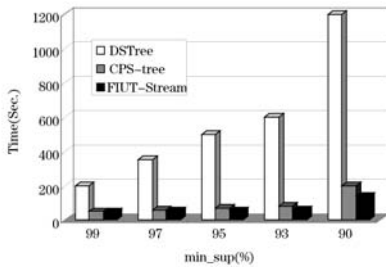


图5 在数据集 Connect-4 下时间消耗的比较

其次考虑空间开销, 设 panes = 10K 事务, 对于数据集 T40110D100K, 设定滑动窗口大小分别为 3, 5, 7, 9, 如图 6 所示, 可见算法 DSTree 需要内存最多, CPS-tree 次之, FIUT-Stream 所需内存最少。对于稠密数据集 Connect-4 实验如图 7 所示, 结果和数据集 T40110D100K 测试结果一样, FIUT-Stream 所需内存最少。DSTree 算法和 CPS-tree 算法都需要构建类 FP-tree, 需要递归挖掘 FP-tree, 空间开销大, 然而 FIUT-Stream 算法在概要结构上和挖掘算法都节约了空间开销, 因而在任意大小滑动窗口下、无论稀疏或者稠密数据集, 算法 FIUT-Stream 显示出了空间优越性。

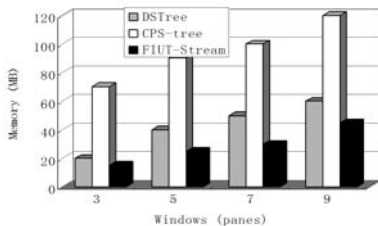


图6 在数据集 T40110D100K 下内存消耗的比较

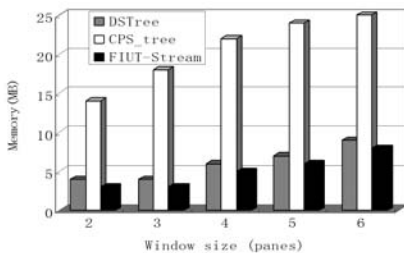


图7 在数据集 Connect-4 下内存消耗的比较

3 结语

本文提出一种新的数据流中挖掘频繁项集算法 FIUT-

Stream, FIUT-Stream 算法将数据压缩在位表结构中, 然后使用 FIUT 算法对数据进行频繁项集的挖掘, 通过引入将数据流分成等长的数据块的思想, 可以准确的挖掘出数据流中频繁项集。不但节省内存开销, 而且利用了比 FP_growth 方法效率更高的 FIUT 算法挖掘频繁项集。FIUT-Stream 算法达到时间和空间相对均衡, 是一种更有效的挖掘数据流方法。

参 考 文 献

- [1] Han J, Kamber M. Data Mining Concepts and Techniques [M]. Morgan Kaufmann, San Francisco, 2006.
- [2] Manku G S, Motwani R (2002) Approximate frequency counts over data streams [C] // Proceedings of the 28th international conference on very large data bases, Hong Kong, 2002:346 - 357.
- [3] Giannella C, Han J, Pei J, et al. Mining frequent patterns in data streams at multiple time granularities [C] // Data Mining: Next Generation Challenges and Future Directions, AAAI/MIT Press, 2004 (Chapter 6).
- [4] Chi Y, Wang H, Yu P, et al. (2004) Moment: maintaining closed frequent itemsets over a stream sliding window [C] // Proceedings of the 4th IEEE international conference on data mining, Brighton, UK, 2004:59 - 66.
- [5] Leung C K S, Khan Q I. DSTree: a tree structure for the mining of frequent sets from data streams [C] // Proc. ICDM, 2006:928 - 932.
- [6] Syed khairuzzaman. Sliding window-based frequent pattern mining over data streams [C]. 2009 Information Sciences 179, 2009:3843 - 3865.
- [7] 李国徽, 陈辉. 挖掘数据流任意滑动时间窗口内频繁模式 [J]. 软件学报, 2008, 19(10):2585 - 2596.
- [8] Yuh-Jiuan Tsay, Tain-Jung Hsu, Jing-Rung Yu. FIUT: A new method for mining frequent itemsets [J]. Information Sciences, 2009:1724 - 1737.
- [9] Tanbeer S K, Ahmed C F, Jeong B S, et al. CP-tree: a tree structure for single-pass frequent pattern mining [C] // Proc. PAKDD, 2008:1022 - 1027.
- [10] Blake C L, Merz C J. UCI Repository of Machine Learning Databases [D]. University of California-Irvine, Irvine, CA, 1998.
- [11] IBM, QUEST Data Mining Project [CP/OL]. <http://www.almaden.ibm.com/cs/quest>.
- [12] Han J, Pei J, Yin Y. Mining Frequent Patterns without Candidate Generation [C] // Proc. ACM-SIGMOD Int'l Conf. Management of Data, 2000:1 - 12.
- [13] Priya R V, Vadivel A, Thakur R S. Frequent Pattern Mining Using Modified CP-Tree for Knowledge Discovery ADMA 2010, Part I, LNCS 6440, 2010:254 - 261.

(上接第 110 页)

- [7] 王鑫印. 基于多 Agent 的数据挖掘系统任务协同研究 [D]. 苏州: 苏州大学, 2004.
- [8] 程苗. 基于云计算的 Web 数据挖掘 [J]. 计算机科学, 2011, 38(10):146 - 149.
- [9] 李伯虎, 张霖, 等. 云制造-面向服务的网络化制造新模式 [J]. 计算机集成制造系统, 2010, 16(1):1 - 7.
- [10] Google Code University. Introduction to Parallel Programming and Map/Reduce [EB/OL]. [2011 - 11 - 21]. <http://code.google.com/intl/zh-CN/edu/parallel/mapreduce-tutorial.html>.