文章编号:1001 - 9081(2008)02 - 0515 - 04

基于分布式数据缓存技术的 Web-OLAP 系统研究

曹丽娟,谢 强,丁秋林
(南京航空航天大学 信息科学与技术学院,南京 210016)
(caolijuan1135@126.com)

摘 要:为了解决在分布式环境下,Web-OLAP 系统并发访问量急剧增加导致 OLAP 服务器负担过重的问题,提出一种基于分布式数据缓存技术的 Web-OLAP 系统。给出了该系统的总体框架和分布式缓存数据的表示,并设计了分布式缓存数据的管理算法。具体的应用实例表明,该方法可以有效地提高分布式环境下 Web-OLAP 系统的访问效率,较大缩短系统的响应时间。

关键词:Web;分布式环境;OLAP;分布式数据缓存
中图分类号: TP311 **文献标志码**:A

Research of Web-OLAP system based on distributed data cache technology

CAO Li-juan, XIE Qiang, DING Qiu-lin
(*College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing Jiangsu* 210016, *China*)

Abstract: To resolve the problem of the heavy Online Analytical Processing(OLAP) server burden resulted from the sharply increasing number of the concurrent inquiry to Web-OLAP system in the distributed environment, a Web-OLAP system based on the distributed data cache technique was brought forward. The overall framework of the system and the expression of the distributed data cache were given out, and the management algorithm of the distributed data cache was designed. Finally, an application case was given. The application result shows that the method can improve the inquiry efficiency of the Web-OLAP system in distributed environment, and greatly shorten the response time.

Key words: Web; distributed environment; Online Analytical Processing(OLAP); distributed data cache

0 引言

联机分析处理(OLAP)是数据仓库中重要的分析工具,它针对特定问题进行联机数据访问和分析,专门用于支持复杂的分析操作,侧重于对决策人员和高层管理人员提供决策支持,可以根据分析人员的要求,快速、灵活地进行大数据量的复杂查询处理,并且以一种直观易懂的形式将查询结果提供给决策人员,以便准确掌握企业的经营状况,制定正确的决策方案[1]。

C/S 模式逐步被 B/S 所代替,简化了客户端,冲破了地域限制[2]。通过 Web 方式访问数据仓库,可以最大限度地发挥数据仓库及 OLAP 的优势。目前,在 Web 环境下应用数据仓库进行决策支持已经成为研究和应用的热点[3-5]。许多大型企业的总部和成员企业分布于不同的地域,在管理上,企业总部负责总体监管和决策,成员企业也有相对独立的经营权,需独立进行决策分析活动。因此,企业需要逻辑上一体的分布式决策支持系统,分布式环境下 Web-OLAP 系统更能满足其经营决策的需求。

分布式环境下 Web-OLAP 系统中,当企业总部需访问成员企业数据仓库时,查询延时加大,且在 Web 条件下,系统的并发访问用户急剧增加,大大加重了 OLAP 服务器的负担。良好的查询响应性能是人们最为关注的,即便有再强大的分析功能,人们往往也无法忍受过长的等待时间。因此,在分布式环境下 Web-OLAP 系统如何提高 OLAP 查询和分析操作效率,以便及时向用户提供分析结果是目前迫切需要解决的关键问题。为此,本文将分布式数据缓存机制应用到分布式环境下的 Web-OLAP 系统中,设计了缓存中的数据结构和对缓存的管理算法,缓解服务器的压力。

1 Web-OLAP 系统框架

基于分布式数据缓存技术的 Web-OLAP 系统的总体框架如图 1 所示。每部分均是由三层结构组成的一个 B/S 系统,三层结构分别为:表示层、逻辑层和数据层。系统应用逻辑层和数据层均呈现分布式的特征,以实现数据存储、数据组织、计算能力和计算任务的平衡分布。

与传统的三层结构相比,本文在逻辑层中引入了对象池和分布式数据缓存技术。关于对象池技术,本文利用文献[6]中研究成果。分布式缓存技术是本文的研究重点,基于分布式数据缓存技术的 Web-OLAP 系统中的缓存池呈现分布式的特征:企业总部的数据缓存池联合各成员企业的数据缓存池为总部的决策支持提供服务;各成员企业之间的数据缓存池是相对独立的,分别为成员企业自身的决策支持提供服务。本文采用动态的缓存管理策略,充分利用系统中各成员企业数据缓存空间,提高了资源利用率,运用合理的查询替换机制有效地提高了数据缓存池的访问效率。

1)表示层。

表示层即系统的客户端,只需使用常用的 Web 浏览器,它接受从服务器传来的数据,根据用户的要求组织成相应的

收稿日期:2007 - 08 - 13;**修回日期**:2007 - 10 - 23。 **基金项目**:国防基础科研基金资助项目。
作者简介:曹丽娟(1983 -),女,江苏泰州人,硕士研究生,主要研究方向:OLAP、决策支持; 谢强(1972 -),男,四川绵阳人,讲师,博士,主要研究方向:知识工程、信息系统与信息安全、人机交互; 丁秋林(1937 -),男,江西抚州人,教授,博士生导师,主要研究方向:企业信息化。

形式展现给用户。同时接受用户的输入,分析用户的动作,根据用户操作的方式和用户操作的对象,生成新的多维查询要求,发送到后台的逻辑层进行处理。

2)逻辑层。

逻辑层是应用服务器层,它是连接客户端和后台数据库的连接点,是系统的核心。其功能是接受从客户端发送过来的多维查询要求,经过一系列处理后将结果返回给客户端。本文在分布式环境下,在企业总部和各成员企业的 OLAP 决策支持系统中,引入对象池和数据缓存技术,重点在于研究总部数据缓存和各成员企业数据缓存之间的协作,提高分布式环境下 Web-OLAP 决策支持系统查询和分析操作效率,以便及时向用户提供分析结果。

3)数据层。

各成员企业的局部数据仓库从各自的操作型数据以及外部数据中抽取数据,经过清洗、转换、综合后存放在关系型数据仓库中。全局关系型数据仓库定期从各个局部关系型数据仓库进行数据综合和提取,全局数据仓库的另外一个数据源是外部数据,主要是企业总部收集的一些同企业经营活动相关的外部信息。

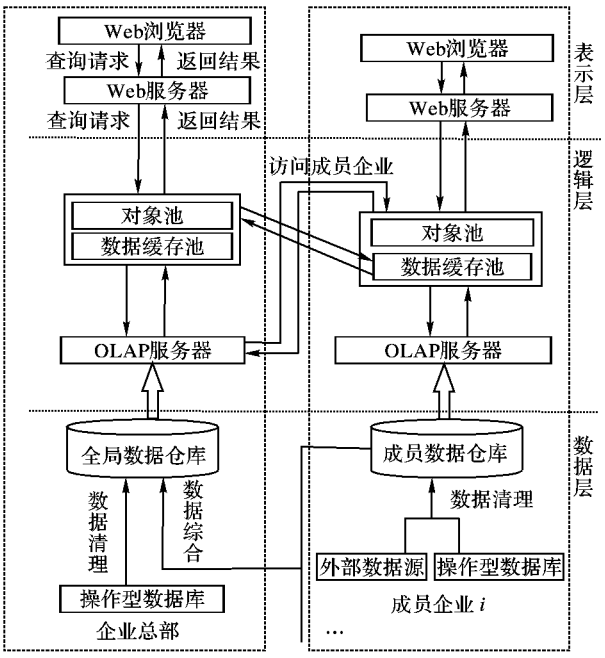

图1 系统的总体框架

2 分布式数据缓存的表示

2.1 分布式数据缓存的结构

分布式数据缓存的结构如图2所示,缓存池动态地缓存了用户提出查询的结果集,企业总部数据缓存池中多维数据集队列中的数据集,可以是从全局数据仓库经查询分析得出的结果集,也可以是存储在成员企业数据缓存池中的多维数据集。成员企业之间的数据缓存池是相对独立的,其中的多维数据集,与该成员企业对象池中的对象相对应,成员企业对用户角色的分类方式与企业总部对用户角色的分类方式相同,方便企业总部 OLAP 对象访问成员企业的数据缓存池。在执行用户查询分析的过程中,系统可以将成员企业数据缓存池中的内容按一定的策略调进企业总部数据缓存池,或在总部缓存池空间不足时将总部数据缓存池的数据退回其所属的成员企业数据缓存池,从而提高数据缓存访问效率,增大对系统资源的利用率。

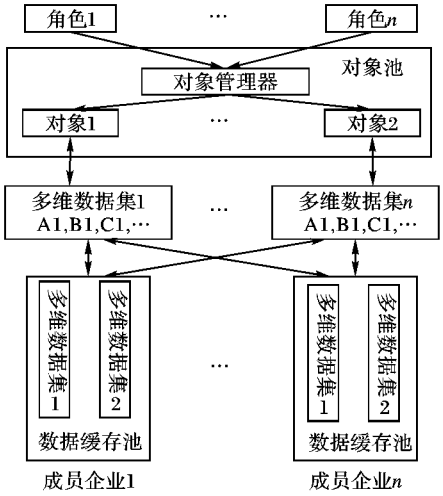

图2 分布式缓存机制系统

2.2 分布式数据缓存的表示

数据缓存池中的多维数据集以队列的方式存储,便于添加和删除。企业总部的数据缓存池中的多维数据集队列,与对象池中的某一 OLAP 对象查询的范围相对应。队列中存储数据结构分为两种形态:一种是存储的以多维数组为存储方式的数据集;另一种是存储成员企业所处位置及其数据缓存池中相应的多维数据集队列的地址,便于 OLAP 对象在查询过程中对多维数据集的访问。成员企业中,数据缓存池中的多维数据集按队列的形式进行存储,分类的方式与总部相似,供总部查询分析时访问和本地对象池中的 OLAP 对象进行查询分析。其结构包含四种类型的节点,如图3所示。

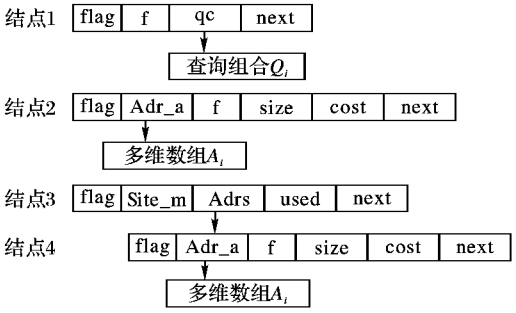

图3 缓存池中元素的数据结

节点1中:flag 为标志位,取值为"0"表示该队列为空,取值为"1"表示该队列不为空;f 表示该队列被访问的频率;qc 指向该队列中的所有查询组合 Q_i ;next 指向该队列中代表下一个查询组合的队列,若该节点为队列末节点,则 next 值为 null。

节点2中:flag 为标志位,取值为"0"表示该节点所指的多维数组是由企业总部数据仓库数据查询分析而得;Adr_a 为一地址值,指向队列中的一个多维数组;f 表示该多维数组被引用的频率;size 表示该多维数组的大小;cost 表示 OLAP 服务器重新计算该多维数组需要花费的代价;next 指向该队列中的下一个多维数组所在的节点,若该节点为队列末节点,则 next 值为 null。

节点3中:flag 取值为"1"表示该节点所指的多维数组队列是由成员企业数据仓库数据查询分析而得,此节点为队列头节点;site_m 表示该成员企业的地址;used 为地址,指向数据缓存中记录各成员企业数据缓存池使用情况(如表1所示),从而可以获得该成员企业数据缓存的使用情况;next 指向该队列中下一个多维数组所在的节点,若该节点为队列末节点,则 next 值为 null。

节点 4 中相关参数可参考节点 1:flag 为“2”表示该节点所指的多维数组是由成员企业数据仓库数据查询分析而得。next 为指向下一个与节点 4 同类型的节点,若该节点为企业总部数据缓存池中的节点,当该节点为末节点时,next 值为该成员企业数据缓存池中某多维数据集队列的地址;若该节点为成员企业数据缓存池中的节点,当该节点为末节点时,next 值为 null。

表 1　成员企业数据缓存使用

member	total	used
企业总部 1	总容量	已使用
企业总部 2	总容量	已使用
⋮	⋮	⋮
企业总部 n	总容量	已使用

在企业总部数据缓存池中 4 种节点结构均存在。在成员企业数据缓存池中只存在节点 1 与节点 4 两种结构。

3　分布式数据缓存的维护

对缓存评价的主要指标有:缓存容量、命中率、访问延时、缓存替换算法和缓存同步算法等。命中率和缓存容量之间存在一定的矛盾,缓存容量增大后,将导致缓存命中率的降低、缓存效率的下降、系统访问延时的增大。因此,缓存容量一般固定在一定的大小上,从而当缓存容量不够时,需要一定的缓存替换算法,将对系统“价值”小的缓存对象从缓存中清除,同时把“价值”较高的对象放入缓存中[7]。

3.1　分布式数据缓存维护的算法设计

企业总部与成员企业数据缓存之间的替换策略有所不同。在分布式环境下,为了充分利用各成员企业的物理资源,当企业总部数据缓存不足时,为了获得一定的存储空间,需从数据缓存中选择对系统“价值”较小的缓存对象,将其替换出来,并对其进行判断,如果该多维数据集是通过对成员企业数据仓库进行查询操作获得的,则将其替换回该成员企业的数据缓存池中;否则,将其删除。由于决策者在制定一个决策时,需反复查看,查询分析操作具有一定的重复性。下次需要再次访问该多维数据时,以从该成员企业的数据缓存池中将该多维数据集调入到企业总部的数据缓存池中。采用该种缓存替换策略,可以避免企业总部 OLAP 对象频繁访问成员企业数据缓存池中的数据集,从而提高了 OLAP 查询和分析操作效率,缩短了系统对客户提出查询分析任务的响应时间。

合理的替换策略,对缓存的效率起着决定性的作用,目前对缓存的替换算法已有许多研究,已有一些比较有效的替换策略。文献[6]在集中式操作环境下采用的评定“价值”的公式,对数据缓存中的多维数据集队列中的结果集进行替换时,已取得较好的效果,但该公式并不适合在分布式环境下使用。在分布式环境下,为了充分利用各成员企业的物理资源,本文在原有评定价值的公式中加入参数 $used(S_i)$,如式(1)所示:

$$goodness(v_i) = \frac{f(V_i) \times cost(V_i) \times used(S_i)}{size(V_i)} \quad (1)$$

其中:$0 \leqslant i \leqslant n$,$n$ 表示成员企业的个数;$f(V_i)$ 表示视图在前面查询中被引用的频率;$cost(V_i)$ 表示 OLAP 服务器重新计算视图 V_i 需要花费的代价,$size(V_i)$ 则表示该视图所占有的空间的大小;$used(S_i)$ 表示成员企业 i 的数据缓存的利用率。对于成员企业 i 而言,$0 \leqslant used(S_i) \leqslant 1$,对于从企业总部数据仓库中数据查询而得到的多维数据集,则用 $used(S_0)$ 表示,且 $used(S_0) = 1$。V_i 表示该多维数据是对成员企业 i 中的数据仓库进行查询操作得到的。V_0 表示该多维数据是从企业总部数据仓库中经查询分析操作获得的。这样,在同等计算代价、引用频率以及占用空间大小的情况下,将该多维数据 V_i 返回空间利用率较低的成员企业 i 的数据缓存中,从而增加了整个系统的资源利用率。

成员企业数据缓存池的维护方式可以采用文献[6]中集中式环境下数据缓存池的维护方法。

3.2　分布式数据缓存维护的算法详细描述

下面给出分布式数据缓存维护的算法描述(算法名称:分布式数据缓存维护算法)。

```
输入:当前查询 qc,qc 的结果视图 MVc
void DS_cache_management(que,MVc){
MV = {MV_i | i = 1,…,n}
// MV_iw 为数据缓存池中的一个多维数据集
V_in = ∅;          // 位于企业总部数据缓存池的多维数据集
V_out = ∅;          //来自成员企业数据缓存的多维数据集
if ( size(MV_c) > Main_CACHE_SIZE )
return;/ * 如果此次查询结果比企业总部缓存的总大小大,则不允许这个结果进入缓存 * /
MV = MV ∪ {MV_c} ;
if (que 结果集所操作的多维数据集对应的队列在缓存中)
  {计算 g(MV_j) (MV_j ∈ MV);
  //运用公式计算 MV 中所有视图的收益值;
  While(size(MV) > Main_CACHE_SIZE)
    { MV_i = 收益值最小的视图;
    // 选出收益值最小的视图 MV_i
    if ( MV_i = MV_c)
      Return;/ * 如果选到了 MV_c 收益值最小,那么不允许这个结果进入缓存 * /
    else
    MV = MV - {MV_i};
    if ( MV_i.flag =0)    //该视图是对全局数据仓库查询而得
    V_in = V_in ∪ {MV_i };
    //暂时不直接删除,把需要删除的视图记录起来
  Else If ( MV_i.flag =2)
  //该视图是对成员企业数据仓库查询而得
  V_out = V_out ∪ {MV_i };
  //暂时不返回成员企业,把需要返回的视图记录起来
  }
  MV = MV - V_in - V_out;/ * 从缓存中删除 V_in 中的视图,将 V_out 中的视图返回到相应的成员企业中 * /
  Update_used ( );/ * 更新记录缓存使用情况表中的内容;根据 qc 查找相应的插入位置;将 MV_c 插入该队列 * /
  }
  Else
{ if (对列数 == 限定数目)
    {for( i = 0;i < 队列限定数目;i++ )
      if( Queue[i].flag = = 0 )
        Queue = Queue - Queue[i] ;// Queue 为表示所有队列的多维数组
    }
    Else
Queue = Queue - Queue[i] // Queue[i].f 值为最小
构造该结果集对应的新队列 New_que;
New_que. next = Queue[0]. next;
Queue[0]. next = New_que;
//将 new_que 插入队列的头部更新相应项的 f 和记录
Update_used ( );          //更新记录缓存使用情况表中的内容
}
```

4 应用研究

在军工制造业中,主机制造厂为了在有限时间内生产大量武器装备,来协同其他具有特定制造能力的制造单元来共同完成任务,因而扩散制造的运作模式就在目前军工制造业中应运而生。扩散制造将武器装备产品划分成若干模块、次模块、组件、附件和零件等部分,对各部分的制造工艺进行优化与固化,然后把生产任务扩散到具有特定制造能力的制造单元,充分利用资源的整合与优化所产生的集群效应,实现武器装备在较短时间内完成大批量的生产任务。扩散制造的生产模式要求企业在网络化制造环境下能够快速响应市场需求,需要决策者能够快速制定相应的决策。我们在某扩散制造企业的决策支持系统中利用一 B/S 模式的分布式 Web-OLAP 系统。该扩散制造决策支持系统中某个扩散件一次交货合格率同期对比,如图4所示。

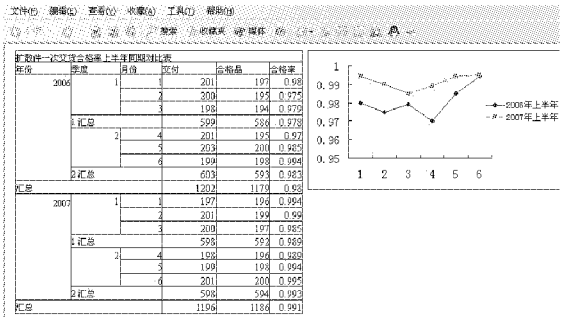

图4 扩散件一次交货合格率同期对比

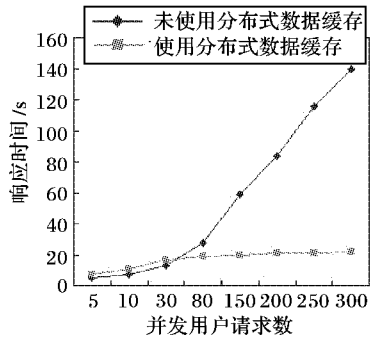

图5 并发用户请求响应时间对比

将一般情况下的决策支持系统与本文的操作结果进行对比后如图5所示,本文所采用的方法可以在用户数量增多时,较高地提高系统的响应速度。

5 结语

本文将分布式数据缓存技术引入到 Web-OLAP 系统中,通过总部数据缓存,与各分部的数据缓存协同工作,在客户端用户数据增多的情况下,减轻各服务器的查询负担,充分利用网络中的资源,缩短客户在做查询分析时的响应时间。

参考文献:

[1] 王珊. 数据仓库技术与联机分析处理[M]. 北京: 科学出版社, 1998.

[2] 王惠敏. 网络环境下对分布式决策支持系统的探讨[J]. 价值工程, 2006, 8: 88 - 90.

[3] MADEIRA H. COSTA J, VIEIRA M. The OLAP and data warehousing approaches for analysis and sharing of results from dependability evaluation experiments[C]// International Conference on Dependable Systems and Networks. [S. l.]: IEEE Press, 2003: 86 - 91.

[4] YANG W, ZHU W, LIU Y. Research of a Web-based DSS intelligent Agent over data warehouse[C]// 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI'04). Beijing: IEEE Press, 2004: 433 - 436.

[5] POWER D J, KAPARTHI S. Building Web-based decision support systems[J]. Studies in informatics and control, 2002, 11(4): 291 - 302.

[6] 于雅丽, 谢强, 丁秋林. Web 环境下基于对象池和数据缓存技术的 OLAP 系统[J]. 武汉大学学报: 工学版, 2006, 39(6): 59 - 62.

[7] 赵玉伟. WWW 中缓存机制的应用研究[D]. 武汉: 武汉理工大学, 2006.

(上接第514页)

同时应该指出,尽管改进的 χ^2 统计方法取得了一定效果,但针对整个文本分类问题的效果仍未出现明显提高。正如文献[5]所讲:文本分类问题是涉及到文本表示、相似度计算和算法决策等多种复杂技术的综合应用。

4 结语

为了解决 χ^2 统计方法提高了在指定类中出现频率较低,却又普遍存在于其他类的特征项在该类中的权重以及对低文档频的特征词不可靠两个不足之处,本文综合考虑了频度、集中度、分散度等三项指标,提出了改进的 χ^2 统计方法。经过文本分类系统的实验验证,改进的 χ^2 统计方法性能要好于传统的方法。

参考文献:

[1] 李凡, 鲁明羽, 陆玉昌. 关于文本特征抽取新方法的研究[J]. 清华大学学报: 自然科学版, 2001, 41(7): 98 - 101.

[2] 陈治纲, 何丕廉, 孙越恒, 等. 基于向量空间模型的文本分类系统的研究与实现[J]. 中文信息学报, 2004, 19(1): 36 - 41.

[3] SCHUTZE H, HULL D A, PEDERSEN J O. A comparison of classifiers and document representations for the routing problem[C]// Proceedings of the 18th ACM International Conference on Research and Development in Information Retrieval. [S. l.]: ACM Press, 1995: 229 - 237.

[4] 鲁松, 李晓黎, 白硕, 等. 文档中词语权重计算方法的改进[J]. 中文信息学报, 2000, 14(6): 8 - 13.

[5] YANG YI-MING. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval[C]// Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval. Dublin: Springer, 1994: 12 - 22.

[6] 徐凤亚, 罗振声. 文本自动分类中特征权重算法的改进研究[J]. 计算机工程与应用, 2005, 41(1): 181 - 184.

[7] 程泽凯, 陆小艺. 文本分类中的特征选择方法[J]. 安徽工业大学学报, 2004, 21(3): 221 - 224.

[8] 杨允信. 文本文件自动分类之研究[C]// 台湾地区第六届计算语言学研讨会论文集. 台湾: [s. n.], 1993.