ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, Supplement, December 2009, pp.165−175
© by *Institute of Software, the Chinese Academy of Sciences*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

分布式聚集函数支持的内存OLAP并行查询处理技术*

张延松[1,2,3+], 张 宇[3], 黄 伟[1,2], 王 珊[1,2], 陈 红[1,2]

[1](数据工程与知识工程教育部重点实验室(中国人民大学),北京 100872)

[2](中国人民大学 信息学院,北京 100872)

[3](哈尔滨金融高等专科学校 计算机系,黑龙江 哈尔滨 150030)

Distributed Aggregate Functions Enabled Parallel Main-Memory OLAP Query Processing Technique

ZHANG Yan-Song[1,2,3+], ZHANG Yu[3], HUANG Wei[1,2], WANG Shan[1,2], CHEN Hong[1,2]

[1](Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education (Renmin University of China), Beijing 100872, China)

[2](School of Information, Renmin University of China, Beijing 100872, China)

[3](Department of Computer Science, Harbin Financial College, Harbin 150030, China)

+ Corresponding author: E-mail: zhangys_ruc@hotmail.com

Zhang YS, Zhang Y, Huang W, Wang S, Chen H. Distributed aggregate functions enabled parallel main-memory OLAP query processing technique. *Journal of Software*, 2009,20(Suppl.):165−175. http://www.jos.org.cn/1000-9825/09020.htm

Abstract: A multi-node parallel main-memory OLAP system is proposed in this paper which is considered by the character of OLAP queries and the performance of main-memory database system. In this system, multi-dimensional OLAP queries with aggregate functions are distributed to each computing node to get aggregate results and final result can be available by merging all the aggregate results from multiple computing nodes. Comparing with other solutions, this system uses horizontal distribution policy to distribute massive data in multi-node with only consideration of memory capacity of computation node. According to feature of distributed aggregate function, system can improve parallel processing capacity by lazy results merging which can reduce the volume of message between nodes, the overall performance of parallel query processing can be improved. This system is easy to deploy, it is also practical with good scalability and performance for the requirements of enterprise massive data processing.

Key words: parallel query processing; main-memory OLAP DB; lazy aggregate computing

摘 要: 根据OLAP查询的特点和内存数据库的性能特征提出了由多个内存数据库组成的并行OLAP查询处

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2009AA01Z149 (国家高技术研究发展计划(863)); the Union Project with Beijing Municipal Education Commission (BMEC) on Industry-Study-Research of China (北京市教委产学研合作项目); the Renmin University of China Graduate Science Foundation under Grant No.08XNG040 (中国人民大学研究生科学研究基金); the Joint Research of HP Lab. China and Information School of Renmin University of China (中国人民大学信息学院与HP中国实验室合作项目)

Received 2009-05-01; Accepted 2009-07-20

© 中国科学院软件研究所 http://www.c-s-a.org.cn

理系统,将 OLAP 应用中的多维聚集查询分布到各个计算节点并行进行聚集计算,并将聚集计算的结果进行合并输出.与其他并行处理方法相比,该算法充分利用 OLAP DB 结构中维表远小于事实表的特性,根据数据库中事实表的数据量和节点的数据处理能力进行水平数据库分片,并根据聚集函数的可分布计算特性提高查询处理的并行度,延迟并行查询处理中的合并过程,充分利用节点的并行处理能力,减少并行查询处理过程中的数据通信量,提高系统并行查询处理性能.该算法易于实现,具有较好的可扩展性和性能,适用于企业级海量数据处理领域的需求.

关键词: 并行查询处理;内存 OLAP 数据库;延迟聚集计算

OLAP(on-line analytical processing)查询是面向海量数据的复杂多维查询,在星型结构或雪花型结构的数据仓库中,典型的 OLAP 查询中包括多表连接操作、复杂的谓词操作、order by 操作和 group-by 操作,查询结果一般是聚集计算值.OLAP 查询的主要特点是数据量大、计算量大,是一种典型的计算密集型应用.在 ROLAP(relational OLAP)模型中使用 DBMS 查询引擎来实现 OLAP 查询处理,对数据库的 I/O 性能和查询处理性能提出了更高的要求.Survey 公司的 OLAP 调查报告指出,目前 OLAP 应用中用户最关注的问题是性能,即当前的 OLAP 查询引擎在面向海量数据处理时的性能远远不能满足用户的需求.对于传统的 DRDB(disk resident database)系统而言,其主要的性能瓶颈是海量查询处理时巨大的 I/O 代价.随着计算机硬件的发展和成本的降低,大内存已成为服务器的典型特征,现代高端服务器支持的内存容量已经达到 TB 级,内存数据库 MMDB(main-memory database)技术越来越多地为企业级应用研究所关注并大量应用于高响应性要求的电信、金融等领域.与 DRDB 相比,MMDB 最显著的特征是使用内存作为数据的主要存储设备,从而在查询处理时消除了 I/O 代价,通过面向内存存储结构与 CACHE 访问性能的各种优化技术,MMDB 的性能无论在 OLTP 领域还是 OLAP 领域都远远优于传统的 DRDB.

内存数据库的高性能是建立在充足内存的前提下,当内存数据库的数据量超过系统内存容量时,内存数据库的性能会迅速下降.在企业级的 OLAP 应用领域,采用 MMDB 作为 OLAP 的查询处理引擎可以大大提高 OLAP 查询的处理性能,但由于企业级应用数据量的迅速增长,系统需要动态可扩展的处理能力,而内存容量受到服务器硬件配置、操作系统类型等多种因素的影响,难以相应地提高内存容量以适应数据的变化.目前在 OLTP 领域,内存数据库技术已得到广泛的应用,而在 OLAP 领域还没有成熟的内存数据库产品,制约内存数据库技术在高性能 OLAP 领域应用的决定因素是内存容量的扩展性难以满足实际应用需求的变化,因此,动态可扩展的内存数据库系统对于提高 OLAP 系统的性能和适用性具有重要的意义.

本文第 1 节对并行 OLAP 查询处理的相关技术作了简要的回顾.第 2 节提出基于 MMDB 的内存数据库并行 OLAP 查询处理系统的模型并给出并行 OLAP 查询算法的描述.第 3 节设计 MMDB 并行 OLAP 查询算法的性能对比测试实验,并分析实验结果.第 4 节给出论文的结论并讨论进一步的工作.

1 相关工作

1.1 并行CUBE计算

OLAP 查询根据其查询处理模式的不同分为两大类:ad-hoc OLAP 查询和基于物化视图的 OLAP 查询.由于传统的查询处理引擎难以提供高性能的实时多维查询处理能力,因此往往通过生成物化 CUBE 的方法来提高 OLAP 查询的响应性能.物化 CUBE 的计算需要大量的存储空间开销和时间开销,因此,在多个处理节点上进行并行 Data CUBE 的计算可以有效地提高 CUBE 计算的性能.

文献[1,2]通过 SN 结构的集群系统来实现 ROLAP 中的并行 CUBE 计算,通过数据分区、并行计算、分区合并等过程实现将 CUBE 的全局计算分解为在多个节点上并行执行的分区计算,并通过优化算法在多个节点之间进行负载均衡来提高系统的综合性能.实验结果表明,通过集群并行计算,CUBE 的生成速度得到显著提高.

文献[3,4]讨论了在 MOLAP(multi-dimensional OLAP)系统中应用分布式的多个服务器节点上的内存处理

© 中国科学院软件研究所 http://www.c-s-a.org.cn

基于内存多维数组的 Data CUBE 构建算法,并行处理能够有效地提高 OLAP 查询的处理性能.

文献[5]讨论了在并行 CUBE 计算中的数据倾斜问题,它指出,数据倾斜对于并行 CUBE 计算的性能具有重要的影响作用,因此需要动态的负载均衡策略来提高系统的并行计算能力.文献[6]讨论了使用 MPP 作为 OLAP 查询处理引擎的策略,查询代价评估模型需要根据查询代价分析来决策是否将查询提交给 MPP 引擎处理并进行并行查询的分发、监控与查询处理结果的组装.

1.2 分布式聚集函数

与普通的查询不同,OLAP查询是基于特定维结构和维层次的聚集计算结果,是基于特定维属性分组的聚集计算值,它反映的不是细节数据而是全局的聚集计算结果.对于SUM,COUNT,MAX,MIN等聚集函数而言,其计算过程满足结合律,即若$A=A_1\cup A_2\cup\ldots A_n$,则$SUM(A)=SUM(SUM(A_1),SUM(A_2),\ldots,SUM(A_n))$.

文献[7,8]通过增量 CUBE 维护的方法来降低 CUBE 维护的开销.当事实表改变时,记录值的变化写入差分表中,然后通过差分表计算ΔQ(通过差分表记录生成的 CUBE),将当前 CUBE 与ΔQ 合并生成基于假设数据的 CUBE.算法中要求 CUBE 的聚集操作支持分布运算(数据可以分成多个集合独立处理,然后将各个计算结果合并,标准 SQL 中 SUM,MAX,MIN 和不带 distinct 关键字的 COUNT 操作支持分布运算,而取中值操作 MEDIAN 不支持分布运算).

与基于 SN 结构的并行查询处理技术相比,第 1.1 节中介绍的算法是通过多个并行处理节点创建一个 CUBE.对于返回可分布式计算的聚集函数计算结果的OLAP查询而言,我们可以将事实表的记录分成若干个数据子集并独立地进行 OLAP 查询处理,这样相当于将一个完整的 CUBE 分解为若干个子 CUBE,多维查询可以在多个子 CUBE 上独立地并行执行,只需要在最终的结果上进行合并就可以得到与在全局数据集上的查询一致的结果.由于 OLAP 查询结果的数据量很小,因此在查询结果上的合并操作的代价很低.在 OLAP 查询中,group-by 之前与之后的数据量相差悬殊,通过将 OLAP 查询的聚集计算延迟到在各个节点完成 group-by 之后,可以提高 OLAP 查询的并行度,减少查询处理过程中的数据通信量,实现延迟聚集计算的关键是聚集计算的可分布性.

通过对 Oracle,SQL server,PostgreSQL 等数据库系统中聚集函数的分析,我们可以将OLAP查询中使用的聚集函数分为 3 类:

1. 可分布式计算的聚集函数:包括 SUM,COUNT,MAX,MIN 等.
2. 基于代数运算的可分布式聚集计算函数:包括 AVERAGE 和方差类的聚集函数.
3. 基于全局的不可分布式计算的聚集函数:包括基于全局位置的聚集函数,如 MEDIAN,TOP(n)等.

从聚集函数的使用频率来看,以 TPC-H 中的统计数据为例,在 22 个测试查询中,前两类的聚集函数的使用率为 100%,其中 SUM,COUNT 和 AVERAGE 这 3 个聚集函数的使用率达到了 95%.因此,根据 OLAP 查询中聚集函数的可分布式计算的特点,将 OLAP 查询分解为面向多个数据子集的并行查询处理可以有效地提高系统的综合处理性能并具有良好的处理能力和处理容量的扩展性.

1.3 并行计算和内存数据库技术

基于 SN 结构的并行计算能够有效地提高系统的性能和处理能力,但是其性能受数据分布策略和并行查询处理时的数据传输代价的影响.在最理想的情况下,N 个并行处理节点可以将查询的性能提升 N 倍.但在实际的查询处理时,并行查询的性能受查询处理过程的串行操作所占比率的影响,当串行化操作所占的比率较高时,查询整体的性能难以达到最优的线性加速比,当并行处理的节点增多时,查询处理性能的提升远远低于并行处理节点数增加的倍数.

连接操作的并行查询处理性能的决定因素是数据分布能否满足并行处理.在星型结构的 OLAP 应用中,多维查询往往需要事实表同时与多个维表进行连接操作,面向多个连接属性的数据分布策略将大大增加系统的复杂性,从而降低并行查询处理的性能.在数据仓库中,事实表的存储开销远远大于维表的存储开销,如在 4GB 的 TPC-H 测试数据库中,事实表 lineitem 的存储空间与其他表的存储空间之比为 69.1%:30.9%,而在更规范的星

© 中国科学院软件研究所 http://www.c-s-a.org.cn

型结构的测试数据库 SSB 中,事实表与维表存储空间开销之比为 95.5%:4.5%,因此,在 OLAP 应用中对事实表采用水平分片而对维表采用全复制方式可以大大降低系统维护的复杂性,更好地提升查询的并行度,并且将存储空间的开销控制在可以接受的程度.

在分析型数据库系统中,列存储模型比行存储模型具有更好的 OLAP 查询处理性能,典型的列存储数据库系统有 Sybase IQ,C-Store 和 MonetDB 等.列存储模型能够有效地降低查询处理时的 I/O 代价或内存带宽,能够提供更好的查询处理性能,但列存储模型的查询处理以列数据为单位,处理过程具有较强的串行性,不适合并行处理.文献[9]讨论了在列存储的内存数据库 MonetDB 中进行的扩展性研究,通过多个节点提高系统整体的内存容量,从而满足海量 OLAP 查询的内存数据存储需求.实验结果表明,在目前千兆网卡条件下,基于列存储模式的分布式查询处理时,节点之间的数据传输代价极大地影响了系统整体的查询处理性能,单纯的基于列存储模型的垂直分片策略难以有效地提高系统查询处理的并行能力.文献[10]提出高性能的并行处理数据库 ParAccel,包括内存处理技术、列存储技术、压缩技术和高速互联网络技术,提高各个节点的数据处理能力并通过高速网络降低并行查询时节点间数据传输的延迟.但查询优化器以传统的连接优化为主,并未针对 SN 结构中的数据分布进行优化,而且系统中各查询节点是一种紧密耦合状态,并非一种开放式的并行处理框架.

在性能和扩展性的权衡中我们选择了基于列存储内存数据库查询引擎的并行 OLAP 查询处理技术来解决高性能海量数据上的 OLAP 查询处理问题.列存储的内存数据库技术面向高性能 OLAP 查询处理,而基于可分布式聚集计算的 CUBE 分解与合并技术用以解决系统处理能力和处理容量问题.即用列存储的内存数据库技术弥补海量 OLAP 查询的性能缺陷,用 CUBE 分解和合并技术弥补列存储内存数据库在并行性和扩展性方面的缺陷.系统在可分布式计算聚集函数的支持下,通过将 CUBE 划分为多个子 CUBE 的方法独立进行各子 CUBE 上的 OLAP 查询并最终合并查询结果,支持不同的 SQL 引擎,具有良好的动态适应能力.

2 基于内存数据库的并行 OLAP 查询处理技术

2.1 系统模型

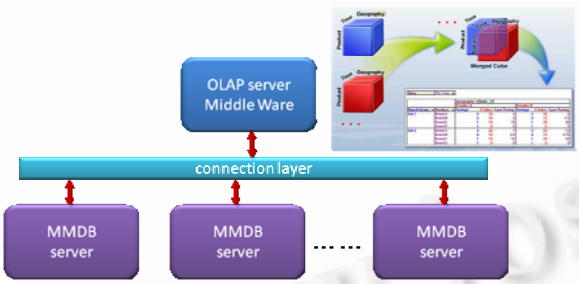


Fig.1 Framework of system

图 1 系统基本架构

如图 1 所示,在基于内存数据库的 OLAP 并行查询处理系统中,由 OLAP 中间层服务器驱动多个内存数据库查询处理节点,事实表的数据按节点的内存容量以任意顺序分布在多个节点上,当数据仓库中追加数据时可以追加到任意节点上或平均分布在各个节点上,减少了系统维护的代价和复杂度.节点的数量由事实表大小和服务器内存容量决定.OLAP 服务器通过连接层与各个内存数据库节点建立连接,向各个节点并行地提交查询处理请求,并将各个节点的查询处理结果进行合并输出.

系统的基本思想是利用聚集函数的可分布计算特性和 CUBE 结构的特点,将在单一节点上进行的海量数据上的 OLAP 查询操作转换为在多个节点子 CUBE 上的并行执行的 OLAP 查询与查询结果合并运算,从而简化系统复杂度和提高查询的并行执行能力.

2.2 聚集函数的分布式计算方法

OLAP 查询可以看作是 CUBE 空间上的切片或切块操作,对于可分布式执行的聚集函数,查询结果的每一个分组相当于一个累加器,通过将各个节点上的查询结果中相同分组的聚集值累加即可得到 OLAP 查询的最终结果.也就是说,查询执行的过程分为两个阶段,一是将查询提交到多个查询节点并行地执行;二是将查询结果合并为最终查询输出结果.

如图 2 所示,在整个查询处理过程中,各查询处理节点只需要将查询的最终结果返回给 OLAP 中间层服务

© 中国科学院软件研究所 http://www.c-s-a.org.cn

器,没有额外的通信代价.

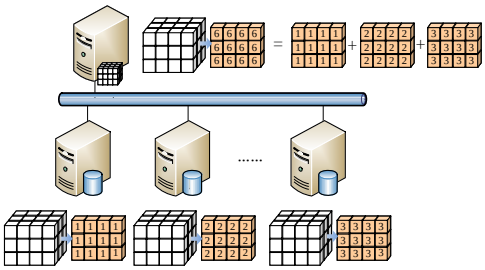


Fig.2 Computing of distributed aggregate functions

图 2 可分布式计算聚集函数的计算过程

基于代数运算的可分布式聚集计算函数可以通过可分布式计算的聚集函数的结果计算得出,假设聚集计算属性A分布在n个不同的节点上,即$A=A_1\cup A_2\ldots\cup A_n$,则

$$AVERAGE(A_1\cup A_2\ldots\cup A_n)=\sum_{i-0}^{n}SUM(A_i)\Big/\sum_{i=0}^{n}COUNT(A_i) \tag{1}$$

对于 AVERAGE 聚集函数,在第 1 阶段中,将各个查询节点上执行的查询命令中的 AVERAGE 聚集函数替换为 SUM 和 COUNT 两个聚集函数.在第 2 阶段中,分别通过合并过程计算出每个分组的全局 SUM 和 COUNT 值,并计算得出全局 AVERAGE 聚集函数值.

$$VARIANCE(A_1\cup A_2\ldots\cup A_n)=\frac{\sum_{i=0}^{n}SUM(A_i^2)}{\sum_{i=0}^{n}COUNT(A_i)}-\left(\frac{\sum_{i=0}^{n}SUM(A_i)}{\sum_{i=0}^{n}COUNT(A_i)}\right)^2 \tag{2}$$

方差聚集计算也可以转换为 SUM 和 COUNT 两个聚集函数的代数表达式,第 1 阶段在各个节点上独立地计算各分组的相应聚集计算结果,第 2 阶段在 OLAP 服务器上将各节点的聚集计算结果进行合并运算,获得各分组最终的聚集计算结果.

可代数计算的分布式聚集函数的计算可以归纳为以下规则:

1. 当聚集函数可以转换为若干个可分布式计算的聚集函数的代数表达式时,将查询中该聚集函数转换为相应的若干个可分布式计算的聚集函数,并将结果上传到 OLAP 服务器中间层,在中间层计算各个聚集函数的全局计算结果,并最后计算出最终的聚集函数结果.

2. 当转换后的聚集函数中包含代数计算的分布式聚集函数时,系统需要递归地通过多趟算法才能得到最终的结果.所增加的额外通信量为查询结果中分组的中间结果.

对于与全局位置相关的不可分布式计算的聚集函数来说,如 MEDIAN 和 TOP(n),其结果为分组中全部排序记录特定位置的记录度量值,必须在全局排序的记录序列中才能得到正确的结果,不能在分布的多个分组的排序子集中分布处理.图 3 描述了 MEDIAN 函数的处理过程,我们采用两种策略进行聚集函数的计算.

第 1 种策略是集中式计算策略,如图 3(a)所示,将分组聚集的查询操作转换为根据分组属性进行排序的操作,然后将各个节点的排序记录子集在 OLAP 中间层服务器上进行多路归并排序,并在全局的排序结果集上计算出各个分组的中值.在集中式策略中,各个查询处理结果返回给 OLAP 服务器中间层的是全部的事实表记录,可能会超出该服务器的处理能力,从而使中间层服务器成为系统性能瓶颈.

第 2 种策略是分布式计算策略,如图 3(b)所示.每个查询处理子结点在生成排序记录的同时充当临时的全局查询处理节点的角色,即将分组计算的功能分布到各个查询处理子节点上,其他查询处理子节点上该分组对应的排序记录子集发送到该节点上进行该分组的全局排序并根据全局排序的结果计算出最终的中值结果.当查询中各分组的 MEDIAN 计算在各个节点上分布地完成后,各查询子节点将最终的结果发送到 OLAP 中间层服务器并组合为最终的查询执行结果.在分布式计算策略中,分组排序、分组合并、分组 MEDIAN 结果的全局

© 中国科学院软件研究所 http://www.c-s-a.org.cn

计算都分散到各个查询处理子节点上,每个子节点只处理部分分组的排序记录,所处理的数据量不会超出节点的内存容量,而 OLAP 服务器中间层只对最终的结果进行合并处理,其数据量也远远小于整个事实表的数据量.

从查询执行时所需要的通信量来分析,两种策略的总通信量都为全部事实表记录数.在集中式计算策略中,OLAP 服务器中间层节点是通信和数据处理的瓶颈节点;而在分布式计算策略中,通信和数据处理的负载分散到各个查询处理子节点中,能够避免瓶颈节点的出现.

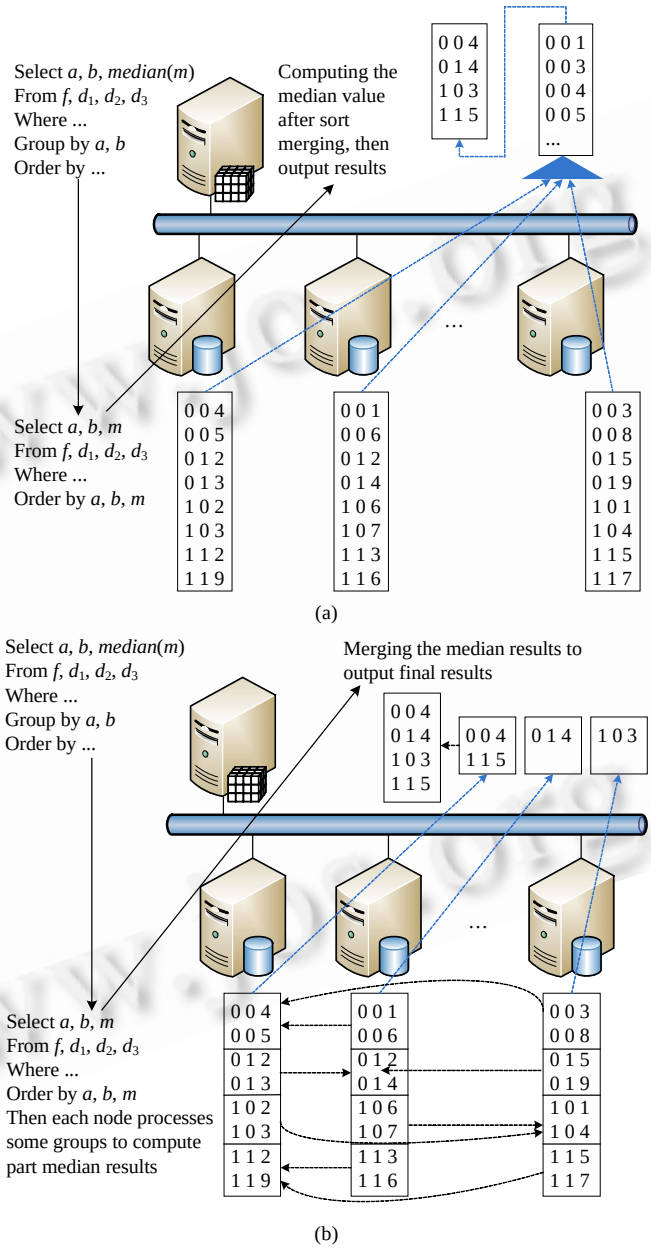


Fig.3 Computing of non-distributed aggregate functions

图 3 不可分布式计算聚集函数的计算过程

2.3 MEDIAN聚集函数的优化

MEDIAN 函数在是全局排序记录中寻找中间位置的记录,在计算出分组中的记录数目后,只需扫描到中间

位置的记录后即可得出中值,我们用 N 表示该分组的记录数量,因此扫描记录的数量为 $N/2$.在分布式计算模式下,为得到中值需要将全部 N 条记录在中间层服务器上进行全局排序后再扫描,在全局排序的分组记录中,后 $N/2$ 的记录操作为无效操作.如果采用网络通信的方式在节点间进行多路归并排序的方式扫描中值记录,虽然中值操作所需要扫描的记录数减少为 $N/2$,但又大大增加了系统间的网络通信量.基于减少分布式 MEDIAN 聚集计算通信量的目标,我们提出了迭代收缩的分布式 MEDIAN 聚集计算算法,迭代地缩小需要确定中值记录的排序记录范围,当小于阈值λ时,由中间层服务器进行最终的计算.如图 4 所示,算法描述如下:

OptDistribMEDIAN(MMDBNode $n_1,n_2,\ldots$,SQLStatement s)

1. 对n个节点上的同一分组$M_1,M_2,\ldots,M_n$,分别计算本地的MEDIAN值$m_1,m_2,\ldots,m_n$;
2. 计算该分组的全局记录数$N=N_1+N_2+\ldots+N_n$,将N'的初始值设置为$N/2$;
3. 取$m_0=\mathrm{MIN}(m_1,m_2,\ldots,m_n)$;
4. 分别计算各个节点中度量值小于m_0的记录的数量$Cn_1,Cn_2,\ldots,Cn_n$,令$N'=N'-(Cn_1+Cn_2+\ldots+Cn_n)$;
5. 若$N'<\lambda$,则将各节点上从$Cn_i(1<i<n)$之后的N'个记录子集发到中间层服务器节点进行多路归并排序并输出第N'个记录的度量值作为该分组的返回值;
6. 否则,在各节点m_0记录之后的(N'/n)个记录中令$m_i(1<i<n)$为各分组记录集中的最大值,转向 3.

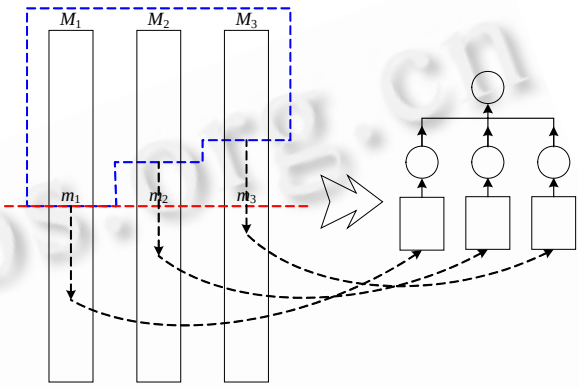


Fig.4 Optimization of MEDIAN aggregate function

图 4 MEDIAN 函数的优化算法

2.4 基于内存数据库的并行OLAP查询处理算法

在我们所提出的系统模型中,内存数据库作为 OLAP 查询的高性能执行引擎,其查询处理的数据量应小于内存容量,以满足高性能内存数据处理模式,根据数据仓库的数据量和内存容量确定了系统所需的 n 个内存服务器节点后,查询被OLAP中间层服务器并行地分发给各个内存服务器节点计算相应的子CUBE中多维切块的结果,最后将各节点上的子多维切块在中间层服务器上进行合并输出.对于不同类型的聚集函数,系统调用不同的访问接口将聚集计算分布化.完整的查询处理过程描述如下:

ParallelOLAPExecutor(MMDBNode $n_1,n_2,\ldots$,SQLStatement s)

1. 将事实表划分为 n 个子集分布到 n 个查询处理节点,维表全复制到 n 个查询处理节点上;
2. if (s 中的聚集函数为 SUM,COUNT,MAX,MIN)
3. $s\rightarrow n_1,n_2,\ldots$,将查询分发到各查询执行节点
4. *MergeResultsFromNodes*($s,n_1,n_2,\ldots$);
5. end if
6. if (s 中的聚集函数为代数可分布式计算聚集函数)
7. if (AVERAGE 或 VARIANCE)
8. $s'\leftarrow s$,转换为可分布式计算聚集函数查询;
9. $s'\rightarrow n_1,n_2,\ldots$,将查询分发到查询执行节点;
10. end if
11. if (嵌套代数分布式计算的聚集函数)
12. $s'\leftarrow s$,转换为可分布式计算聚集函数查询;
13. 多趟算法计算相关的分组聚集结果;
14. end if

© 中国科学院软件研究所 http://www.c-s-a.org.cn

15. *ComputeMergeResultsFromNodes*($s,n_1,n_2,\ldots$);
16. end if
17. if (s 中的聚集函数为 MEDIAN)
18. *OptDistribMEDIAN*($n_1,n_2,\ldots,s$);
19. *MergeResultsFromNodes*($s,n_1,n_2,\ldots$);
20. end if
21. if (s 中的聚集函数为…)
22. …
23. RETURN MergedResults.

系统中查询处理的数据量为数据仓库数据量的 $1/n$,查询处理时间为 n 个节点并行查询处理时间的最大值加上并行查询的预处理和查询结果的合并处理时间.当 OLAP 查询的输出结果中分组的数量较少时,系统之间的通信代价很小;当各个节点查询结果的总数据量超过 OLAP 中间层服务器的内存容量时,可以采用在各子节点查询结果的分组上分片传输,并被服务器以分片为单位进行查询结果的合并处理输出.

3 实验与性能分析

实验环境为 4 台 HP 服务器,其配置为:Intel Itanium-64,2CPU,主频 1.6GHz,4GB 内存,2GB Disk swap.我们通过 Eclipse3.0 开发了一个基于多线程的内存数据库并行查询处理中间层系统,连接 4 个 MonetDB 内存数据库实例,测试数据库采用 TPC-H,SSB(改进的 TPC-H,一个事实表 4 个维表结构)和来自真实应用环境的图书企业历史数据库.

1. 系统存储空间利用率

我们首先分析了采用维表全复制的分布策略时系统整体存储空间的利用率.如图 5 所示,采用维表全复制策略增加了维表在多个服务器节点上的冗余存储代价,随着节点数的增多,冗余存储代价随之增大.对于非规范星型结构的 TPC-H 数据库来说,将 order 表看作维表在节点间全复制增加了系统的存储开销,可以采用将 lineitem 和 order 表 HASH 分片的方法在节点间进行数据分布以减少系统冗余存储的代价,但在向系统中追加数据时会增加数据分布的代价;而在规范化的 SSB 数据库(将 TPC-H 数据库中的 lineitem 表和 order 合并形成单一的事实表和 4 个维表)和来自企业的真实数据库实例中,由于维表所占的存储空间比例很小(小于 5%),因此,在多个节点之间采用全复制仍然有较高的存储空间利用率(约 80%).

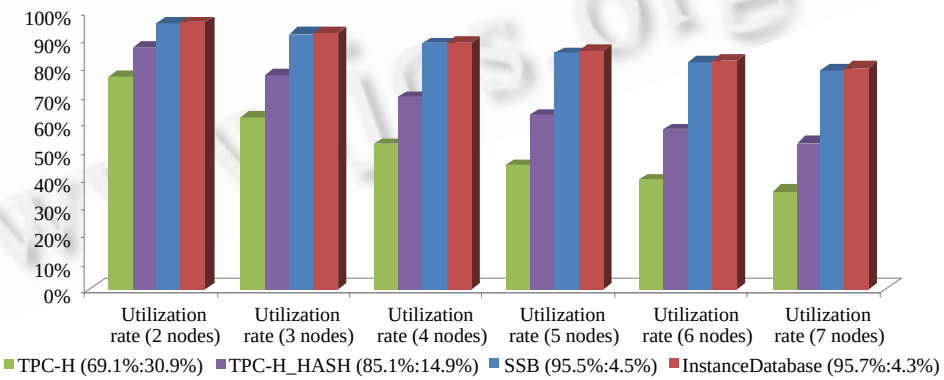


Fig.5 Utilization rate of space cost

图 5 存储空间利用率

在 SN 结构的并行数据库系统中,虽然可以避免冗余存储空间的开销,但多表连接的复杂性会导致数据分片策略的复杂度提高、并行查询处理时节点之间数据通信量的增加、降低查询的并行度.采用物化视图方式来提高 OLAP 查询的性能需要付出额外的物化视图的创建代价和存储的空间代价,完整的物化视图的存储空间

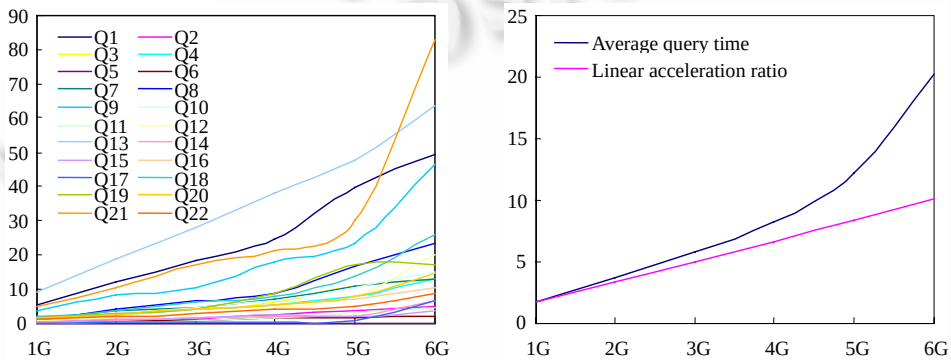
© 中国科学院软件研究所 http://www.c-s-a.org.cn

开销大大超过原始事实表的存储空间开销.在进行海量数据的 OLAP 查询处理时,我们需要权衡查询执行时间与存储空间的开销与收益,当系统能够支持高性能的 ad-hoc 查询时,我们不采用物化视图机制;当系统的查询性能不满足用户的需求时,需要采用物化视图机制提高查询处理性能,但对存储与物化视图维护提出了更高的要求.在我们的方案中采用列存储内存数据库作为高性能的 OLAP 查询处理引擎,实现高性能的内存 ad-hoc 查询处理,以较小的冗余存储代价来获得更高的查询处理性能.

2. 查询性能加速比

我们首先测试了事实表大小与查询执行时间之间的关系,作为基准的并行查询加速比参考.我们测试了数据量从 1GB 与 6GB 的 TPC-H 测试数据库中 22 个标准测试查询的执行时间.

图 6(a)为 22 个测试查询的执行时间,可以看到,查询时间随数据量的变化而相应增加.图 6(b)为平均查询时间,参考线为以 1GB 数据的平均查询时间为基准的线性加速比的查询执行时间.在 1GB~4GB 数据容量的测试中,查询平均执行时间基本符合线性增长规律;当测试数据库的数据量超过 4GB 时,已超过服务器的物理内存容量,产生了 I/O 代价,因此查询平均执行时间的增长幅度迅速提高.因此,当查询处理节点的数据量能够满足内存查询处理时,并行查询能够基本达到线性加速比的性能.



(a) (b)

Fig.6 Acceleration rate of queries

图 6 查询性能加速比

随后我们测试了将 4GB 的测试数据分布在 1 个、2 个、4 个节点上时的查询处理性能.我们从 TPC-H 的 22 个测试查询中选择了 Q1,Q3,Q10,Q12 这 4 个带有 group-by 操作的测试查询,检验采用并行 CUBE 计算模式时的查询处理性能.

如图 7 所示,Q1 和 Q12 的查询结果记录数量分别为 4 和 2,查询结果合并代价极小,并行查询总体性能接近线性加速比;而 Q3 和 Q10 查询结果记录数量分别为 1 853 和 882,由于分组记录数量较多但查询执行时间较短,导致并行查询结果的合并代价占了较大的比例,影响了并行查询总体性能的提升.当查询输出结果为低势集时,并行查询的性能随并行处理节点数量的增加而接近线性提高;当查询结果的势集较大时,虽然并行查询的时间随节点数量的增加而线性减少,但查询结果合并的代价随节点数量的增加而增加,总体查询性能受到影响.

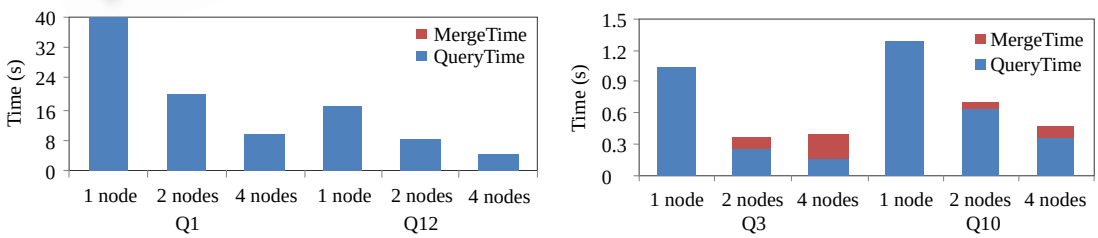
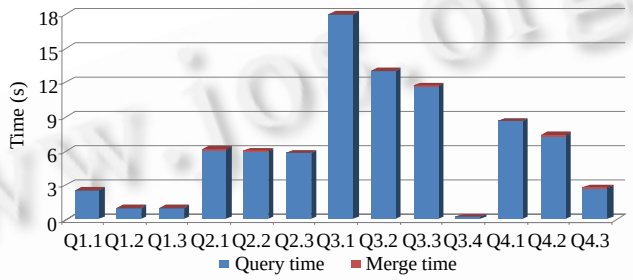


Fig.7 Time cost of query and query results merging

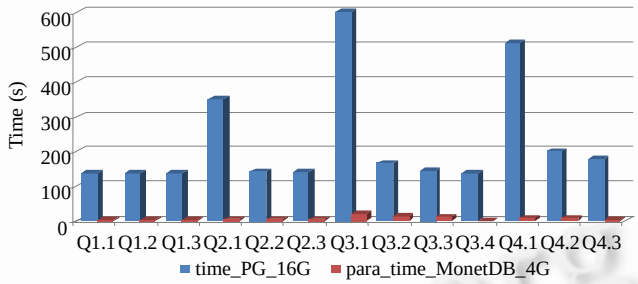
图 7 查询时间与查询结果合并时间

© 中国科学院软件研究所 http://www.c-s-a.org.cn

我们基于星型结构的测试数据库 SSB 进行了大数据量的 OLAP 查询处理测试.在实验中,我们使用了 16G 的测试数据,事实表中有 95 988 640 条事实表记录,在 13 个标准测试查询中,查询结果的行数最大为 800,最小为 1.我们设计了两组实验进行系统性能对比:第 1 组实验采用 PostgreSQL8.3.4 作为查询处理引擎;第 2 组实验采用了 4 个 MonetDB 作为查询处理引擎,每个服务器上分配 4G 数据,并行执行 13 个标准测试查询并合并 4 个查询结果.图 8(a)显示了并行内存数据库系统中查询处理的时间和查询结果合并的时间,由于 OLAP 查询的输出结果为特定属性的分组聚集结果,分组数量随着事实表规模的增加保持恒定,因此当数据量增加时,合并所占的时间比例相应降低,图 8 中合并时间所占的比例大大低于图 7 中所占的比例.16G 的测试数据超过了单一服务器上内存数据库的处理容量,我们通过多个内存数据库服务节点扩展系统的处理能力,图 8(b)显示了使用磁盘数据库 PostgreSQL 在 16G 数据上的查询处理时间和使用 4 个并行内存数据库处理节点并行地处理各自的 4G 数据并合并查询结果模式的查询处理总时间.我们可以看到,由于内存数据库查询引擎的性能远高于磁盘数据库查询引擎的性能,因此整体的查询性能优于线性加速比.



(a)



(b)

Fig.8 Query time cost of DRDB and parallel MMDBs

图 8 磁盘数据库和内存数据库并行系统查询时间

3. OLAP 查询结果势集大小对查询性能的影响

维表中不同的属性具有不同的势集,不同维层次的成员数量也有较大的差别.在并行OLAP查询处理系统中,查询结果的合并过程的时间代价由并行查询的节点数和查询结果的分组数决定,即最终的多路归并计算过程的性能由需要归并的路数和每路数据量决定.我们假定在事实表全集上的查询执行时间为T_0,系统中有n个并行查询处理节点,并行查询执行时间为T_0/n,查询结果合并的时间为$t\times n$(t为每路查询结果合并的平均时间),则在n个并行查询处理节点上的并行查询处理的时间T_{total}为$T_{total}=T_0/n+t\times n$,若满足$T_{total}<T_0$,则有:$T_0(1-1/n)> t\times n$,即$T_0>t\times(n^2/(n-1))$,当$n$较大时,则需要满足条件:$T_0>t\times n$.

基于内存数据库的并行 OLAP 查询处理算法适用于计算代价大但输出记录少的 OLAP 查询处理.

4 结束语

本文提出了在可分布式聚集计算 CUBE 上将 CUBE 划分为多个子 CUBE 并由独立的内存数据库节点进行维护,在此基础上实现多维查询在多个子 CUBE 上的并行执行,并通过子 CUBE 上结果集的合并操作实现

© 中国科学院软件研究所 http://www.c-s-a.org.cn

OLAP 查询在多个节点上的并行执行.对不可分布式计算的聚集函数进行了分析和算法设计,使算法具有良好的适应性.

本文提出的系统模型结构简单、扩展性强,可以根据系统的数据负载和服务器的处理能力动态扩展节点数量,系统的查询处理节点数量变化时不需要代价高昂的数据重分布过程,能够更好地适应应用环境的动态需求,能够充分发挥内存数据库的高性能,弥补内存数据库处理容量的缺陷.进一步的研究工作包括在系统节点数量增加时如何减少系统整体的通信开销并进一步优化 MEDIAN 等不可分布式聚集计算的聚集函数的性能以及如何处理包含子查询的复杂 OLAP 查询.

References:

[1] Chen Y, Dehne F, Eavis T. Parallel ROLAP data cube construction on shared-nothing multiprocessors. Distributed and Parallel Databases, 2004,15(3):219−236.

[2] Dehne F, Eavis T, Rau-Chaplin A. The cgmCUBE project: Optimizing parallel data cube generation for ROLAP. Distributed and Parallel Databases, 2006,19(1):29−62.

[3] Goil S, Choudhary A. Parallel data cube construction for high performance on-line analytical processing. In: Proc. of the 4th Int'l Conf. on High-Performance Computing. Washington: IEEE Computer Society, 1997. 18−21.

[4] Goil S, Choudhary A. A parallel scalable infrastructure for OLAP and data mining. In: Proc. of the Int'l Data Engineering and Applications Symp. (IDEAS'99). 1999. 178−186.

[5] Muto S, Kitsuregawa M. A dynamic load balancing strategy for parallel datacube computation. In: Proc. of the 2nd ACM Int'l Workshop on Data Warehousing and OLAP. New York: ACM, 1999. 67−72.

[6] Lu HJ, Huang XH, Li ZX. Computing data cubes using massively parallel processors. In: Proc. of the 7th Parallel Computing Workshop (PCW'97). 1997.

[7] Mumick IS, Quass D, Mumick BS. Maintenance of data cubes and summary tables in a warehouse. In: Peckham J, ed. Proc. of the ACM-SIGMOD Conf. on Management of Data. New York: ACM Press, 1997. 100−111.

[8] Lee KY, Kim M. Efficient incremental maintenance of data cubes. In: Dayal U, Whang KY, eds. Proc. of the 32nd Int'l Conf. on Very Large Data Bases. VLDB Endowment, 2006. 823−833.

[9] Zhang YS, Xiao YQ, Wang ZW, Ji XD, Huang YK, Wang S. ScaMMDB: Facing challenge of mass data processing with MMDB. In: Chen L, *et al*., eds. Proc. of the 1st Int'l Workshop on Real-Time Business Intelligence at APWeb/WAIM. 2009. 1−12.

[10] Chen YJ, Cole RL, McKenna WJ, Perfilov S, Sinha A, Szedenits E. Partial join order optimization in the ParAccel analytic database. In: Çetintemel U, Zdonik SB, *et al*., eds. Proc. of the SIGMOD 2009. 2009. 905−908.

张延松(1973－),男,山东泰安人,博士生,副教授,主要研究领域为高性能数据库,内存数据库,OLAP 应用.

张宇(1977－),女,副教授,主要研究领域为电子商务,数据仓库,OLAP 应用.

黄伟(1985－),女,硕士生,主要研究领域为内存数据库,数据库实现技术.

王珊(1944－),女,教授,博士生导师,CCF 高级会员,主要研究领域为高性能数据库,数据库与信息检索,内存数据库,视频数据库.

陈红(1965－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据仓库与数据挖掘,传感器数据管理.

© 中国科学院软件研究所 http://www.c-s-a.org.cn