海量数据分析的 One-size-fits-all OLAP 技术

张延松[1),2)] 焦 敏[1),3)] 王占伟[1),3)] 王 珊[1),3)] 周 烜[1),3)]

[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)](中国人民大学中国调查与数据中心 北京 100872)

[3)](中国人民大学信息学院 北京 100872)

摘 要 传统的 OLAP 被迅速膨胀的海量数据推动进入了大规模数据分析时代,其主要特点是存储密度大,计算强度大,需要大规模并行存储和处理能力.无论是传统的并行数据库技术还是热点的 MapReduce 技术都不得不面对海量数据在大规模并行处理环境下的性能和并行处理效率的问题.以星型模型上复杂多表连接为基础的 OLAP 算法的复杂度和并行处理过程中的数据网络传输代价都成为制约性能的重要因素.通过深入分析 OLAP 存储模型和查询负载特征,提出了对 OLAP 查询中最基础的 SPJGA-OLAP 子集在存储、查询处理、数据分布、网络传输和分布式缓存等方面面向海量数据大规模并行处理框架的优化策略和实现技术.通过对 TPC-H 和 SSB 两个工业界和学术界公认的测试标准的分析,评估了技术的可行性.提出了以内存 predicate-vector DDTA-JOIN 算法为核心的并行内存 OLAP 架构,以维表上规范化的谓词向量操作替代了多样的连接执行计划,实现以一种查询处理模型同时满足集中式处理和大规模并行 OLAP 处理的需求,充分利用现代计算机的硬件优势,最小化网络传输和 OLAP 查询处理代价.实验中分析了在 1TB 和 100TB 数据集中数据分布策略的存储代价和传输代价,通过并行 OLAP 代价模型和实际数据的实验测试验证了技术的可行性和并行处理效率.

关键词 OLAP;海量数据分析处理;谓词向量;星型模型

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2011.01936

One-size-fits-all OLAP Technique for Big Data Analysis

ZHANG Yan-Song[1),2)] JIAO Min[1),3)] WANG Zhan-Wei[1),3)] WANG Shan[1),3)] ZHOU Xuan[1),3)]

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

[2)](*National Survey Research Center at Renmin University of China, Beijing* 100872)

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

Abstract The traditional OLAP is pushed into large scale analysis era by rapidly expending big data volume. The major features are high storage density, heavy workload, large scale storage and processing capacity. Both traditional parallel database and the hot topic MapReduce technique have to face the critical issues of performance and parallel processing efficiency of big data analytical processing in large scale parallel processing framework. The performance of star schema based OLAP with star-join is limited by processing complexity and network transmission cost in parallel processing. This paper makes a deep analysis of features of storage model and workload of OLAP, proposes the optimization mechanisms and implementation technologies for the most fundamental SPJGA-OLAP subset in storage, processing, distribution, network transmission, and distributed buffering. The technical feasibility is evaluated with the commonly accepted TPC-H

收稿日期:2011-07-10;最终修改稿收到日期:2011-08-29.本课题得到国家重大科技专项基金项目(核高基项目 2010ZX01042-001-002)、国家自然科学基金项目(61070054)、中国人民大学科学研究基金(中央高校基本科研业务费专项资金,10XNI018)、中国人民大学研究生基金项目(11XNH120)资助.**张延松**,男,1973 年生,博士,主要研究方向为内存数据库、OLAP 和高性能数据. E-mail: zhangys_ruc@hotmail.com. **焦 敏**,女,1975 年生,博士研究生,讲师,主要研究方向为内存数据库、OLAP 和高性能数据库. **王占伟**,男,1985 年生,硕士,主要研究方向为内存数据库、OLAP 和高性能数据库. **王 珊**,女,1944 年生,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为高性能数据库、知识工程、数据仓库. **周 烜**,男,1979 年生,博士,副教授,主要研究方向为信息检索、高性能数据库.

industrial benchmark and SSB academic benchmark. This paper proposes the predicate-vector DDTA-JOIN centric parallel OLAP framework, replacing the diverse join execution plans with normalized predicate-vector processing, and enables one-size-fits-all OLAP model for both central processing and large scale parallel processing by making advantage of nowadays hardware, minimizing network transmission cost and processing cost. The analysis of the storage cost and network transmission cost for distribution mechanism with datasets of 1 TB and 100 TB is given. The technical feasibility and parallel processing efficiency are verified by OLAP cost model analysis and real data experiments.

Keywords　OLAP; big data analytical processing; predicate-vector; star schema

1　引　言

OLAP 是一种多维数据分析处理模型，基于关系数据库的 OLAP(Relational OLAP，ROLAP)是一种面向分析型负载的读密集型查询处理. OLAP 以星型模型和雪花型模型为存储模型，一般由一个事实表和多个维表组成，OLAP 的基本功能是切片、切块、上卷、下钻、旋转等操作，即在事实表与维表连接的基础上进行不同粒度的分组聚集计算. 在海量数据处理时代，TB 级甚至 PB 级的数据需要大规模并行计算网络的支持，巨大的存储、连接、传输和聚集归并等代价使 SQL 引擎不堪重负. SQL 引擎以传统的事务型处理为基础(OLTP)，相对于 OLAP 负载以数据计算为中心的查询处理模式显得过于复杂，一方面复杂的事务和并发机制增加了冗余的代码代价，另一方面面向大数据集的复杂多表连接操作缺乏强有力的技术支持. 以传统的并行事务处理为基础的并行数据库技术在扩展性方面受分布式事务控制机制的制约而缺乏良好的可扩展性，当前新兴的分析型数据库(如 Vertica、ParAccel、Greenplum 等)虽然面向分析型数据处理的特征优化了存储、查询处理和并行计算等技术，但其查询处理技术仍然带有 OLTP 查询处理引擎的影子，是一种由通用 SQL 引擎面向 OLAP 负载的特殊优化技术. MapReduce是一种大规模并行计算模型，它良好的扩展性使其成为海量数据大规模 OLAP 处理的候选技术方案，但 MapReduce 在解决多表连接问题时低下的性能使其难以适应复杂模型的 OLAP 处理. 因此，问题的关键是，无论是并行数据库技术还是 MapReduce 技术都没有根据 OLAP 的本质特征来创建订制式的并行存储和处理框架，优化工作难以进一步深入.

图 1 显示了 SQL 与 OLAP 的包含关系. SQL 可以看作是查询处理技术的全集，包括事务处理和分析型处理，TPC-C 和 TPC-E 是典型的 OLTP 负载. OLAP 相当于 SQL 集合中面向分析型处理的子集，以 TPC-H 为代表，查询负载以批量更新和读密集型复杂查询为特征，包含了复杂的子查询嵌套结构. SPJGA-OLAP 是本文提出的 OLAP 基本操作集，以 OLAP 中最基础的 S：选择，P：投影，J：连接，G：分组，A：聚集为主，面向 OLAP 模型标准的切片、切块、上卷、下钻、旋转等操作，排除了子查询等复杂操作. SPJGA-OLAP 是通用 OLAP 的核心功能子集.

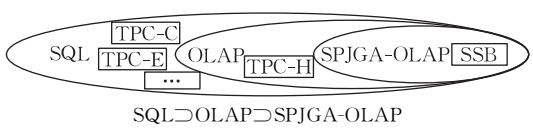

图 1　SQL 与 OLAP 的包含关系

本文的研究以 OLAP 核心的 SPJGA-OLAP 操作集上的优化为中心，提出了面向星型模型特点的以维表为中心的分布式存储模型，将事实表对维表的数据依赖规范化为 bitmap 过滤器，通过分布式维表列存储缓存策略和分组编码谓词向量技术最小化 OLAP 处理时的网络传输代价. 本文的贡献主要体现在以下几个方面：

(1)将 OLAP 最核心的操作集 SPJGA-OLAP 分离出 SQL 集合，从而使优化的目标局限于具有最大并行处理潜质的标准多表连接分组聚集计算上，简化了大规模并行计算模型的复杂度；

(2)提出了以维表为中心的海量数据分布式存储策略，以最低的负载均衡和数据更新同步代价服务于操作型 BI 需求；

(3)将 OLAP 对应的 SQL 操作分解为过滤器、分组器、聚集器，连接谓词根据模式建立内部 key-address 映射，将复杂的 SQL 简化为简单的谓词表达式和属性输入参数，支持 OLAP 向非 SQL 查询

处理引擎的迁移和与各种 SQL 引擎的融合；

(4)谓词向量技术将多表连接的数据依赖规范化为各个维表上的 bitmap 过滤器，最小化并行处理时数据依赖所产生的网络传输代价；

(5)分布式缓存机制充分利用处理节点的内存容量来优化网络传输代价，减少同步更新代价.

本文首先在第 2 节分析 OLAP 模型特征和相关研究的技术路线和成果；在第 3 节中给出 SPJGA-OLAP 模型的描述和实现技术；在第 4 节中设计并行 SPJGA-OLAP 代价模型和实验，并分析实验结果；最后给出论文的结论并讨论了进一步的工作.

2 OLAP 模型分析和相关工作

2.1 TPC-H 和 SSB 模型分析

OLAP 计算模型的复杂度取决于数据模型的特征.图 2 中显示了工业界和学术界普通采用的 TPC-H 和 SSB 标准. TPC-H 是一个双事实表结构，PARTSUPP 和 LINEITEM 都是事实表，以组合键(PARTKEY，SUPPKEY)连接，OREDER 表可以看作是 LINEITEM 事实表的辅助表，LINEITEM 表以(L_ORDERKEY，L_LINENUMBER)为主键，因此 OREDER 表与 LINEITEM 表的连接通常采用索引连接. 在 TPC-H 的 22 个标准测试查询中，查询计划树中的主要执行部分是事实表与多个维表连接的查询子树. 考虑到并行计算环境下的数据分布，由于 OREDER 表与 LINEITEM 表是 1∶4 的对应关系，通用的规则是将 OREDER 表与 LINEITEM 表按 L_ORDERKEY 进行 Hash 分布以减少并行连接时节点间的数据传输代价，提高节点的并行处理能力.但 LINEITEM 表无法同时满足与 OREDER 表和 PARTSUPP 表进行 Hash 分布的需求，TPC-H 中只有 Q9 涉及 LINEITEM 表与 PARTSUPP 表的连接操作，而 LINEITEM 表与 OREDER 表的连接数量较多，因此数据分布策略只考虑 OREDER 与 LINEITEM 表.

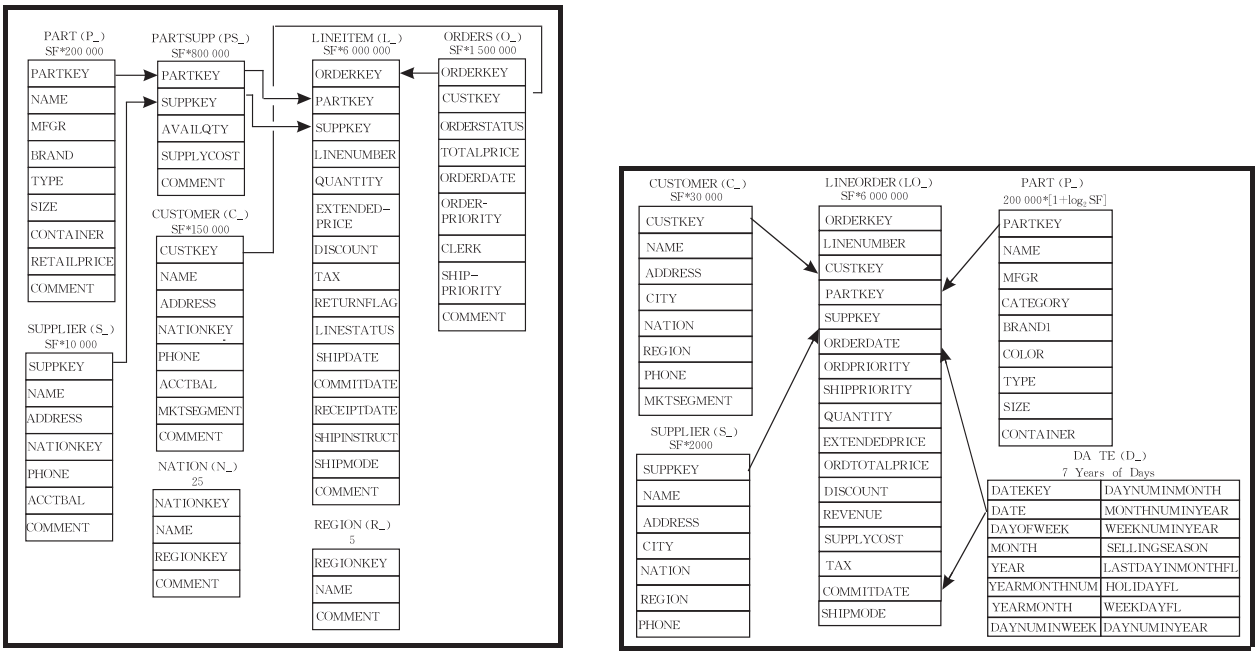

图 2 TPC-H 和 SSB 模型

TPC-H 在模式上形成雪花状结构，因此查询计划中连接操作较多.对于集中式处理模型，雪花状结构能够最小化存储代价，但对于并行计算模型，雪花状结构增加了大量的节点间数据复制或传输代价，LINEITEM 表与 OREDER 表以及其它维表上的大量连接操作在执行代价和数据分布代价上都较大.因此并行计算环境下模式设计与集中式环境下的模式设计有所不同，考虑的首要问题是复制与传输代价而不是存储代价，即需要采用物化或非规范化思想将复杂的雪花状模型简化星型模型，将聚集计算属性尽量归并到事实表中，而维表只保留基本的选择和分组属性，将 OLAP 查询计划规范化为维表上的过滤→与事实表连接→分组聚集计算模式的简单操作.只有简单的存储模型和简单的计算模型才能最大化大规模并行处理的收益.

图 2 中的 SSB 标准[1]是 TPC-H 标准的星型化

模型，目前被学术界所广泛采用. 它将模式清晰地分解为四个维表和一个事实表，消除了 TPC-H 中 LIENITEM 与 ORDER 表的巨大连接代价，消除了雪花状模型带来的复杂查询执行计划，从而使其更加适合于大规模并行计算环境下的简单数据分布.

两种模型的差异还体现在维表数据量上，以SF＝1000(Scale Factor＝1000，对应 1TB 的测试数据集)为例，TPC-H 中 5 个维表的数据总量为 50188825 KB，而 SSB 的 4 个维表数据总量为 4062216 KB，所占的比例分别为约 5%和 4‰. 维表不同的数据量决定着在大规模并行计算环境下采用什么样的数据分布与数据传输策略以及各种策略的执行效率. 我们将在后面的部分继续讨论针对模式特点进行的优化工作.

2.2　相关工作

在关系操作中，连接操作依赖于两个不同的数据集，本文将维表定义为事实表连接依赖数据集(join dependency dataset)，连接依赖数据集可以是整个维表、维表上的选择和投影子集、维表属性列或在维表上生成的 Hash 表. 当处理节点获得连接依赖数据集后，各节点即可执行并行查询处理. 因此并行数据库优化工作的核心问题是优化连接依赖数据集的复制和传输效率[2]. 例如，在 TPC-H 中可以将 LINEITEM 表与 OREDER 表按 L_ORDERKEY 和 ORDERKEY 进行 Hash 分布，保证两表在处理节点上的并行连接性能. 当 SSB 中节点数量较少时，较小的维表可以采用全复制的方式复制到每个处理节点上以支持完全并行的查询处理，其代价是冗余复制的空间代价和维表更新时较高的同步代价. 主流的数据库，如 Tera Data、Greenplum、ParAccel 等一般采用 on-the-fly 传播的方式在并行查询执行过程中动态分布连接依赖数据集. 当维表上的选择率较低且维表数据量较小时，网络传输的效率较高(Gpbs 网络的有效传输效率高于单磁盘的数据传输效率). 但 OLAP 负载与 OLTP 负载不同之处在于，OLTP 查询中选择率一般较低，以点查询为主，而 OLAP 中查询选择率很高，以范围查询为主，在 SSB 测试查询的 4 个维表上，选择率最大值分布在 1/5～6/7，因此网络传输的数据量依然较大.

针对 SSB 特征的 OLAP 查询负载，很多研究[3-7]采用最简单的维表全复制策略，包括并行数据库、DB-cluster 以及 MapReduce 模型上的研究，通过减化数据分布模型的方式简化并行计算模型，从而减少并行计算时高昂的网络传输代价. 否则，由于事实表是多外键结构，与任何一个维表的连接操作都需要将两个表按特定的连接属性在节点中重新 Hash 分布后并行处理，网络传输代价非常高昂.

文献[8]分析了当前 OLAP 的新趋势，其中操作型 BI(operational BI)的需求与传统 OLAP 中只读型数据处理的假设相冲突，因此传统 OLAP 中的物化策略、预处理策略、维表层次编码策略等失去了假设的基础，全复制策略也面临着巨大的同步更新代价. 当前解决操作型 BI 的主要技术路线是双事实表，如 SAP[8] 和 Vertica[9] 都采用了双事实表技术来同时提供分析型和操作型处理. 但从实际应用特点来看，典型的电子商务企业，如 Amazon、淘宝、阿里巴巴等，更新不仅仅体现在不断追加的交易数据，而且包括不断更新的维表数据，而维表数据的变化直接影响 OLAP 的执行结果. 双事实表技术只能解决数据迁移过程中的操作型问题.

2.3　海量 OLAP 时代模型设计原则

海量 OLAP 意味着巨大的数据存储、访问、计算、传输和同步代价，而且需要具有良好的可扩展性支持. 大规模并行计算的核心是简单的可并行计算模型和简单高效的数据分布模型. 数据仓库的基本特征是按主题组织数据，也就是说一个数据仓库的数据模型在逻辑上就是一张表，简单的数据模型能够支持 MapReduce 这样的大规模可扩展并行计算框架. 为了支持简单并行计算，我们需要维护数据模型的单一性，即以事实表为数据存储和并行处理的中心，从模式设计上缩减维表的规模，避免庞大的连接表或维表所产生的数据分布与并行连接代价.

也就是说模式设计应该从 TPC-H 的以业务逻辑为中心向 SSB 的以分析逻辑为中心的设计原则转移. 从并行计算设计上，OLAP 处理集中在易于并行计算的 SPJGA-OLAP 子集，对于 TPC-H 中复杂的迭代子查询处理，我们的原则是将其中 SPJGA 操作子树并行化，查询树中其它难以并行化的部分交给 SQL 引擎来处理，即我们的研究重点集中在并行化收益最大的 SPJGA 操作部分，不做通用的并行 SQL 查询优化.

3　SPJGA-OLAP 模型研究

3.1　存储模型

本文的研究以 SSB 模型为基础. SSB 是以事实表为中心的星型模型，我们提出了反转星型模型的并行存储模型，即以维表集中存储为中心，以事实表

水平分片为外围处理节点.反转星型模型的优点是简单,维表集中存储能够消除操作型 BI 所面临的实时更新所产生的数据复本同步代价,整个存储模型简化为以事实表为中心的分布式单表存储结构,易于数据分布和保持负载均衡,通过分布式缓存策略利用各处理节点内存来加速连接和分组操作.

图 3 所示的存储模型是并行计算框架内的逻辑存储模型,在实际应用中需要与具体的物理存储模型相结合,如在各个处理节点内采用列存储模型[10-11]、压缩等存储优化技术.

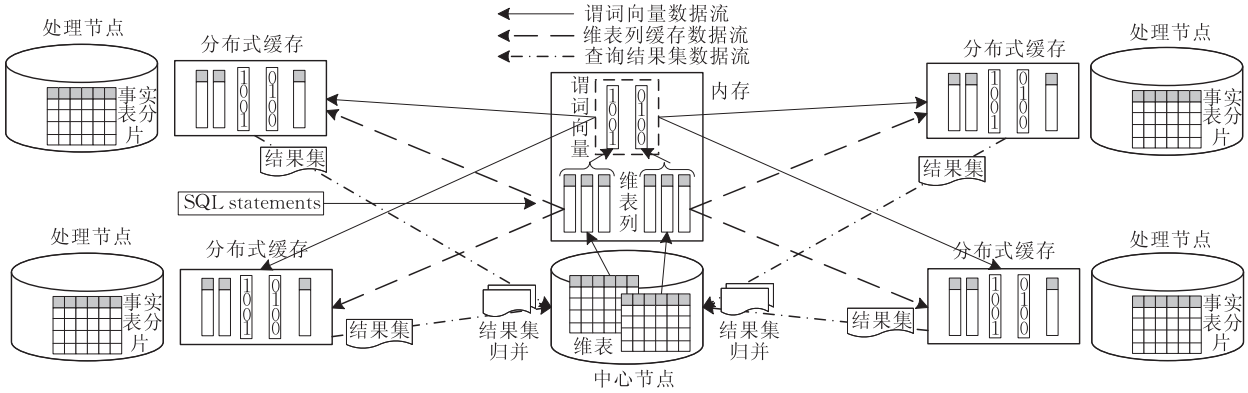

图 3 反转星型存储模型

3.2 OLAP 查询功能分解

SQL 具有复杂的语法结构,在 SPJGA-OLAP 中,SQL 语法可以简化为三类对象:过滤器、分组器和聚集器.

图 4 显示了 SSB 对应的 SQL 命令和功能分解,group-by 子句中的 *c_nation* 是查询的分组器,用于构建 group-by 操作的 Hash 表;where 子句中的谓词一部分是连接谓词,与模式中事实表与维表之间的主外键引用参照完整性约束条件相对应,维表上的谓词表达式起到过滤器的作用,在 SSB 的 SQL 命令中只包括维表属性上的直接谓词;SELECT 子句中的 SUM(…)为聚集器,用于描述事实表度量字段上的聚集计算.

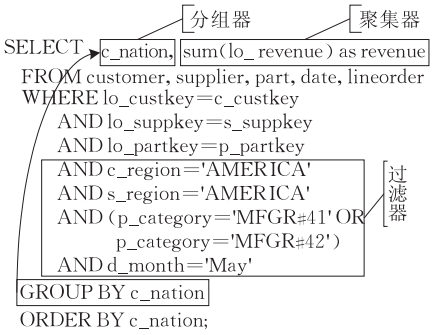

图 4 SSB 查询分解示例

因此,对于 SPJGA-OLAP,复杂的 SQL 被转换为几个标准的输入参数接口,我们可以将 SPJGA-OLAP 定义为 NoSQL 模式的 API,在算法设计和执行层面上独立于 SQL 引擎,避免基于传统的事务型查询处理引擎的设计在 OLAP 处理时的效率损失,同时也可以通过标准的 SQL 转换接口嵌入传统的 SQL 引擎中,作为 SPJGA-OLAP 类查询任务的并行处理加速器.

通过查询功能的分解,维表只起到过滤器和提供分组器的作用.我们将过滤器优化为 bitmap,即用一位来表示维表对应的记录是否满足该维表上所有谓词条件.一方面我们将事实表与维表的连接操作简化为事实表按外键属性与 bitmap 进行匹配,缩减了连接依赖数据集的大小,另一方面,通过维表主键与 bitmap 数组下标的直接映射(维表主键一般为自然序列),事实表与 bitmap 的连接简化为事实表根据外键值直接访问对应下标的 bit 位.

3.3 谓词向量并行 DDTA-OLAP 算法

在反转星型存储模型和 OLAP 查询分解的基础上,并行集群上的 OLAP 处理被分解为 4 个阶段:

(1)查询改写.将 SQL 查询改写为在每个维表上的谓词操作,为每个维表生成唯一的谓词向量(predicate-vector),谓词向量表示为与维表记录数量等长的 bitmap,每一位置 0 或 1,表示该维表记录是否满足维表上所有的谓词.

(2)谓词向量广播.通过广播的模式将中心节点生成的谓词向量传播到各处理节点的内存缓冲区中,为并行 OLAP 处理做数据准备.采用广播方式一方面降低网络传输延迟,另一方面在节点规模扩大时保持网络传输延迟的稳定性.

(3)并行 OLAP 处理.每个处理节点拥有自己独立的事实表数据分片,获得了维表谓词向量后即可独立地完成连接操作.图 5 显示了基于谓词向量

的连接操作过程. 我们在前期研究[12]中提出了 DDTA-JOIN 算法来执行 OLAP 的多表连接操作，其基本原理是将维表主键顺序化使其与内存维属性列数组的下标直接映射，从而使事实表中的维属性外键值可以直接映射到内存维属性列数组的下标，从而将复杂的多表连接操作优化为简单的按事实表外键值进行内存按地址访问操作. 我们将维表属性列进一步优化为整个维表对应一个 bitmap，事实表通过直接访问内存谓词向量 bitmap 完成在维表上的过滤操作.

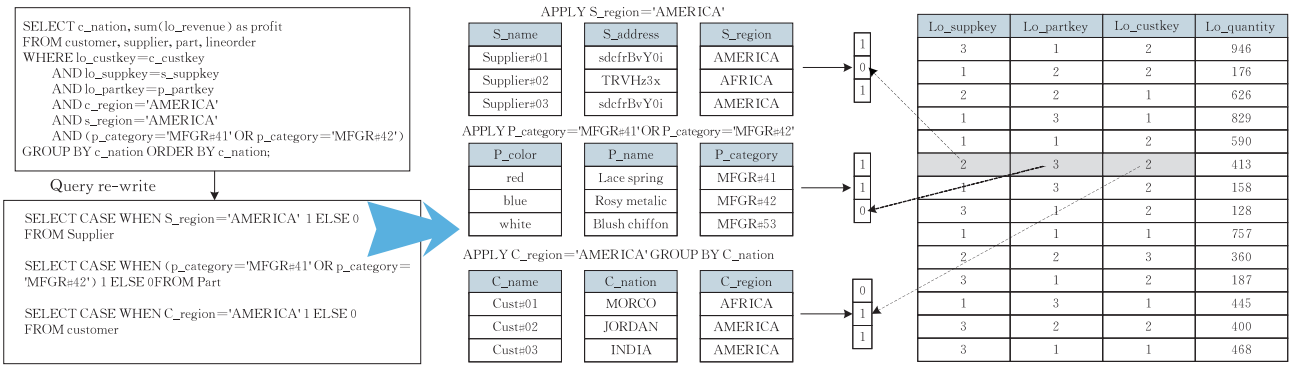

图 5　基于谓词向量的 OLAP 连接操作

图 5 表示一个标准的 OLAP 查询由 SQL 改写为对指定维表的谓词操作，并将查询结果存储为内存 bitmap 形式；在扫描事实表的过程中按事实表中对应的各个维属性外键值直接映射各个谓词向量 bitmap 指定的数据位，并根据多个位进行与操作的结果来判断当前记录是否满足连接条件，满足连接条件的记录再从内存维属性列中按照地址直接映射抽取分组属性值，传递给分组聚集器进行聚集计算. 谓词向量优化技术最小化了连接依赖数据集，减少网络传输代价.

(4) 聚集结果集归并. 在 TPC-H 和 SSB 标准中，聚集函数为可分布式聚集函数(SUM，COUNT)和代数可分布式聚集函数(AVERAGE)，因此各个并行处理节点在各自事实表分片上的聚集计算结果具有可归并性，聚集计算可以下推到并行处理节点内执行. 如果聚集函数是不可分布式聚集计算函数，如 MEDIAN、RANK、PERCENTILE 等，则必须将连接结果集汇集到中心节点后由中心节点完成最终的聚集计算任务. OLAP 查询的结果集以分组聚集计算结果为展示形式，结果集的大小取决于各分组属性的集势(cardinality，即不重复值的数量)，通常情况下远远小于连接的记录数量. 如 SSB 的 13 个测试查询中，分组聚集结果集最多只有 800 条记录，远远小于查询中满足条件的连接记录数量.

在并行处理规模较小时，我们采用集中式 Hash 归并算法来处理并行 OLAP 结果集，当并行处理规模较大时，我们采用迭代归并树算法来处理聚集归并问题(迭代归并树算法用于解决大规模集群中查询结果子集的聚集归并问题，与 Reduce 功能类似，但采用类似于 B^+-Tree 的结构，优化归并过程中的网络连接数量和网络传输代价，在本文中不做过多讨论).

通过对 OLAP 并行查询算法的优化，我们将各种 SPJGA-OLAP 负载规范化为统一的并行 OLAP 处理模型，实现 one-size-fits-all 模式的查询执行计划.

3.4　维属性列分布式缓存策略

谓词向量将维表简化为 bitmap，但 OLAP 查询中的分组属性和维表之间的谓词属性如果被压缩到谓词向量中，需要将对应属性值编码并替代谓词向量中的位编码，从而增加谓词向量的宽度.

如图 6 所示，在 customer 维表上有选择谓词“*c_region*='AMERICA'”和分组属性 *c_nation*，则维表需要提供两种类型的数据，一是谓词向量，用于标识哪条维表记录满足选择谓词条件，二是分组属性，将满足选择条件的分组属性提供给连接操作. 在此我们考虑两种谓词向量策略：

(1) 谓词向量与分组属性组合编码策略.

图 6 显示了该策略的原理. 我们将 customer 维表上的选择改写为“SELECT CASE WHEN *c_region*='AMERICA' *c_nation* ELSE 0 FROM customer”，则可以得到以分组列属性值替代过滤结果的 value-vector，如图 6 中间部分所示，然后将 value-vector 中的数据编码，根据变元数量分配适当的编码宽度，然后用编码代替 value-vector 中的原始值，形成紧凑的 key-vector，如图 6 中最后一个

框图中的部分.通过这种编码向量的方式,每个处理节点可以如图5所示的操作步骤同时完成连接过滤和分组聚集操作,本地结果集以编码作为聚集结果的分类值,最终结果在中心节点进行全局归并后再实现将分组编码通过分组编码字典表的还原过程.

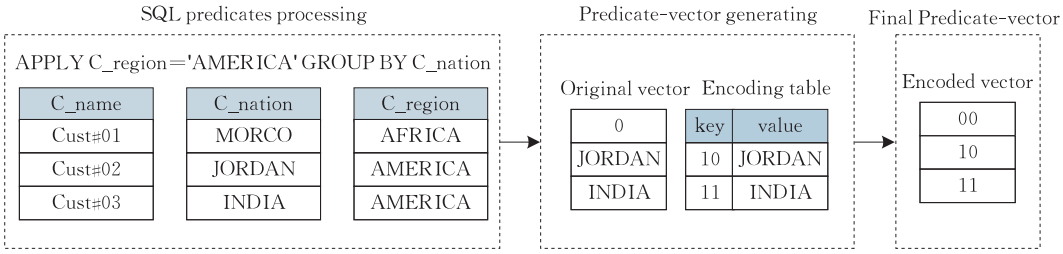

图6 谓词向量与分组属性组合编码

(2)维属性列分布式缓存策略.

当维表较小时且行数较少时,key-vector(如char-vector,可表示2^8个不同变元)数据量较小,网络传输代价差异不大.而且我们在实验中观测到,按下标地址直接访问bitmap-vector时比直接访问char-vector需要更多的CPU周期来解析位操作,因此当维表较小时,分组编码谓词向量(char-vector或短数据类型vector)能够获得较为理想的总体性能.

当维表数量较大时,bitmap-vector与char-vector数量相差8倍以上,基于bitmap的谓词向量能够更少地消耗处理节点的内存并降低网络传输延迟.在这种策略下,需要将OLAP查询所需要的维表分组属性和维表间的谓词属性增量地缓存到处理节点的内存中.如图3所示,随着查询的执行,将维表分组属性通过广播方式缓存到各处理节点的内存中形成内存维属性列,支持基于接收的谓词向量执行的DDTA-JOIN操作.各处理节点的内存相当于分布式缓存,通过中心节点的广播同步更新,缓存的维属性列在缓冲空间不足时可以根据访问频率实行LRU替换算法,缓存的维属性列也可以物化到磁盘上.在SSB中,分组属性相对比较集中,在40个维属性中只涉及到7个属性,分组属性一般为低势集数据列,可以应用字典表轻量内存压缩算法[13]大幅度降低存储的空间代价,提高网络传输和分布式缓存的效率.维表列分布式缓存策略将连接操作数据依赖集分为两个部分,一是连接过滤器,二是分组器,连接过滤器的内容随查询内容的变化而各不相同,属于查询的私有数据集,而分组属性具有共享性,即多个查询可能共享一个较小的分组属性集.对于单个查询,在维表谓词操作基础上选择的分组属性子集编码具有最小的网络传输和分布式缓存代价,但在大量并发查询负载下,完整分组属性列的分布式缓存机制能够有效地降低系统整体的传输和缓存代价.在实验中我们具体分析分布式缓存在SSB中的存储代价.

这种策略增加了维表上的谓词生成代价,增加了谓词向量的宽度和网络传输的数据量,但对处理节点的存储要求最低.缺点是编码谓词向量是查询的私有数据,分组属性在每次查询中都要先过滤再编码,没有重用性.

4 并行SPJGA-OLAP性能分析与实验

4.1 网络传输与分布式存储代价分析

我们用TPC-H和SSB的数据生成器生成1TB数据集(SF=1000),图7中统计了数据集的数据特征.我们看到在TPC-H中维表数据量约占5%,而SSB中维表数据比重为4‰;当采用谓词向量技术

TPC-H(1TB)				SSB(1TB)			
表名	大小/KB	行数	实际行数	表名	大小/KB	行数	实际行数
PART	24420265	SF*200000	200000000	PART	188523	$200000*(1+\log_2 SF)$	2193157
SUPPLIER	1414902	SF*10000	10000000	SUPPLIER	877957	SF*2000	2000000
CUSTOMER	24353654	SF*150000	150000000	CUSTOMER	2995508	SF*30000	30000000
NATION	3	25	25	DATE	228	2556	2556
REGION	1	5	5				
维表总数据量	50188825			维表总数据量	4062216		
维表总行数		SF*360000	360000030	维表总行数		$SF*32000+200000*(1+\log_2 SF)$	34193157
谓词向量大小			43MB	谓词向量大小			4MB

图7 1TB SSB数据集维表数据特征分析

时，TPC-H 中维表谓词向量总数据量为 43 MB（维表每一行映射为一位，谓词向量大小(B)＝总行数/8-bit），SSB 的谓词向量总数据量为 4 MB. 因此在两个测试标准中，并行连接操作数据依赖集的大小分别为 43 MB 和 4 MB，在当前千兆网的支持下（Gbps 网卡理论传输速度为 125 MB/s，实际测试大约在 80 MB/s的水平），谓词向量的网络传输代价为毫秒级（54 ms 和 5 ms），我们采用网络广播模式从中心节点向各个查询处理节点同步谓词向量，因此网络传输代价在网络规模扩展时也能够保持相对稳定.

通过对数据的实际分析，我们获得了在大数据集环境中并行星型多表连接操作数据依赖集的大小和网络传输代价. 处理节点上的多表连接操作被优化为按内存 bitmap 谓词向量直接访问对应位的操作，在确定事实表记录是否满足输出条件后在本地缓存的维属性分组列中抽取分组属性值完成其后的分组聚集操作. 在 SSB 中，3 个较大的维表上的选择率分别为 PART：1/1，000 ～ 2/5，SUPPLIER：1/125～1/5，CUSTOMER：1/125～1/5. 当对维表列采用轻量字典表压缩，采用 16 位压缩编码（能够表示 2^{16} 个变元）时，3 个表的维属性列所占的存储空间分别为 4 MB，4 MB 和 60 MB，可以实现维属性列内存数组化，从而支持通过事实表维属性值向维属性列数组下标的直接映射和数据访问. 在 TPC-H 中，维属性列的可能数据量将分别达到 PART：400 MB，SUPPLIER：20 MB，CUSTOMER：300 MB，分组属性的总数据量可能达到 GB 级. 当维属性上的选择率较低时，可以对分组维属性按谓词向量先进行过滤操作，然后把满足条件的维属性值组织成内存 Hash 表，实现事实表记录向分组维属性列的 Hash 连接，但当并发查询较多时，大量的 Hash 表会占据较多的内存资源，因此在大并发负载和选择率波动较大的情况下，分组维属性列适合于内存直接访问策略.

当数据集大小为 100TB 时，对于 SSB 数据集，谓词向量总量约为 382 MB，维表列大小分别是：PART：7 MB，SUPPLIER：400 MB，CUSTOMER：6 GB，总大小为 6.8 GB，即 100TB 数据集上实现内存 OLAP 处理只需要 6.8 GB 的内存，能够被大多数的计算硬件配置所满足，实现内存 DDTA-OLAP 查询处理. 对于 TPC-H 数据集，谓词向量总量约为 4.3 GB，维表列大小分别是：PART：40 GB，SUPPLIER：2 GB，CUSTOMER：30 GB，总大小为76.3 GB，超出大多中低端服务器的硬件配置，TPC-H 查询中一个维表上经常使用多个分组属性，进一步增加了内存开销. 因此，SSB 数据集比 TPC-H 更加适合于并行内存 OLAP 处理. 对于 100TB 数据集的 TPC-H 查询，可以将谓词向量内存化，维表分组属性列采用磁盘存储模式，在完成事实表与谓词向量的 DDTA-JOIN 后与相应的维表分组列进行基于磁盘的连接操作.

图 8 显示了两种磁盘维属性分组列的连接策略. 左图对应选择率较大的情况，缓存中的事实表记录根据谓词向量找到满足全部过滤条件的记录后，根据谓词向量的数组下标（事实表维属性值）直接访问磁盘中列存储的分组维属性值（列存储以定长字段存储数据，可以将维属性键值直接映射为文件中的偏移地址来直接访问对应的记录），如果采用随机访问性能更好的 SSD 硬盘则将进一步提高磁盘按位置访问性能. 右图对应选择率较小情况下的分组属性访问策略. 先通过内存中的谓词向量对磁盘分组属性列进行过滤，将满足条件的分组属性存储到内存 Hash 表中，以维属性主键（列存储中数据的偏移地址）作为 Hash key，当事实表记录根据谓词向量找到满足全部过滤条件的输出记录后，根据事实表维属性外键值与对应的维属性分组 Hash 表进行 Hash 匹配，找到对应的分组属性值，组合成输出记录，传递给后面的分组聚集器完成其后的操作.

通过对大数据集的分析，我们可以保证在较大的数据规模下（1TB 或者 100TB，1PB 的 SSB 数据集需要约 3.8GB 的内存存储谓词向量，基本能够被当前硬件配置所满足），并行 OLAP 所需要的谓词向量和分组维属性列能够控制在几个 GB 的规模，与当前服务器通常采用几十个 GB 内存的硬件配置相比，能够保证算法的内存处理特性.

4.2　谓词向量并行 DDTA-OLAP 模型代价分析

如图 3 所示，谓词向量并行 DDTA-OLAP 分为 4 个执行阶段：谓词向量创建，谓词向量广播，谓词向量 DDTA-OLAP 处理，查询结果集聚集归并. 并行 OLAP 查询的代价表示为

$$T_{\text{total}} = T_{\text{GenPredVec}} + T_{\text{BroaPredVec}} + T_{\text{DDTA-OLAP}} + T_{\text{ResMerg}},$$

其中：

$T_{\text{GenPredVec}}$：表示生成谓词向量的代价，由维表行数、谓词表达式 CPU 计算代价等因素决定；

$T_{\text{BroaPredVec}}$：由谓词向量大小决定；

$T_{\text{DDTA-OLAP}}$：由事实表分片数据量、分组维属性数量、谓词向量选择率等因素决定；

T_{ResMerg}：由节点数量和结果集大小决定.

4.3　实验平台和实验设计

我们用 8 台 PC 机进行并行 DDTA-OLAP 实

图 8 对较大的磁盘维属性分组列的访问

验测试，每台计算机的配置为 Intel® Core™2 Duo CPU E7500@ 2.93 GHz，2 GB 内存，80 GB 硬盘，操作系统为 Ubuntu 2.6.35-22. 我们使用 Open MPI v1.4.3 作为并行处理框架并实现网络广播模式的数据发布. 一个节点被用作中心节点，集中存储全部维表数据，7 个节点作为并行查询处理节点，每个节点存储本地的事实表分片. 我们通过 C++开发了谓词向量 DDTA-OLAP 实验系统，在实验中我们使用的数据集大小为 7×32 GB=224 GB，每个处理节点分配 32 GB 的事实表水平分片，实验采用分段测试的方法，即在 DDTA-OLAP 并行 SSB 查询处理过程中设置计时器分别统计每个处理阶段的时间. 谓词向量采用分组压缩编码谓词向量和 bitmap 谓词向量＋分布式分组属性缓存两种策略，分别测试了两种策略下谓词向量的生成和广播时间. 由于 SSB 测试的结果集较小且集势稳定，我们采用基于 MPI 的内存 Hash 归并算法进行全局聚集计算归并.

4.4 性能测试与分析

图 9 显示了在并行 DDTA-OLAP 查询处理过程中各个处理阶段的时间代价. 由于谓词向量较小，创建代价大约是几百毫秒，广播传输代价大约为几千毫秒，与基于磁盘 I/O 代价的本地 DDTA-OLAP 相比所占比例非常低. 采用谓词向量技术时，默认为分组属性已增量式缓存于各处理节点，因此只需要传输 bitmap 结构的谓词向量，网络传输代价最小，创建谓词向量、广播谓词向量和查询结果集归并

3 个串行负载代价占并行查询处理总代价的比例为 1.3%. 分组压缩编码谓词向量技术不需要在处理节点缓存分组属性，谓词向量采用多位编码，同时代表过滤信息和分组语义，传输代价略有提高，但串行负载在并行查询处理负载中所占的比例为 2%. 图 10 中显示了串行负载比例最大的分组压缩编码谓词向量技术，其并行处理的加速比接近于理想的线性加速比性能. 实验结果证明在磁盘 OLAP 中利用内存优化处理技术能够以极低的代价来保证复杂多表连接操作的线性处理性能和线性并行加速性能. 同时，广播谓词向量的机制也保证了集群规模扩展时网络传输代价的稳定性.

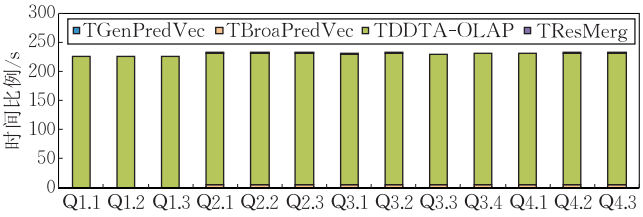

图 9　各个处理阶段的时间比例

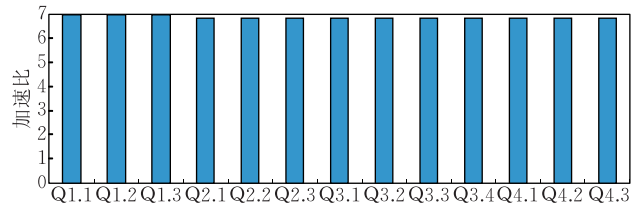

图 10　并行 DDTA-OLAP 加速比

并行加速比是衡量算法可并行性的重要指标，在实际的系统中，不同的数据库产品采用不同的查询优化策略，如列存储、索引、压缩等技术，因此绝对查询处理时间上的对比往往并不公平. DDTA-OLAP 是在查询处理层上进行的优化，不依赖于具体的底层存储技术，同样可以采用列存储来提高 I/O 性能，进而提高整体 OLAP 查询处理性能. 在高可扩展的并行 OLAP 处理框架下，进一步提高 OLAP 性能的关键是提高处理节点的性能，如采用列存储、多核并行、高性能存储设备等软/硬件技术.

图 3 所示的并行处理框架没有硬件和数据分布的限制，因此对于快速增长的数据具有良好的可扩展性，广播和谓词向量机制保持了系统规模扩大时查询处理框架在软件结构上的稳定性. 传统的并行数据库性能依赖于数据分布策略、并行连接操作时的数据重分布或数据迁移策略、查询处理引擎性能等多种因素. 为达到图 10 所示理想的线性加速比性能，一般采用对较小的维表全复制的策略，从而实现数据并行和查询处理并行，但在大规模并行计算集群中产生巨大的冗余存储代价和高昂的同步更新代价. 分析型数据库集群（如 Teradata、ParAccel 等）一般使用特殊设计的高性能硬件来控制并行查询处理时的数据迁移代价，一方面造成硬件成本的迅速提高，另一方面由硬件的依赖性产生可扩展性瓶颈问题. 本文提出的谓词向量广播技术将并行查询时的数据分布控制在非常小的常量时间范围之内，在保证存储效率和同步更新性能的同时与高代价的全复制并行查询处理机制保证相同的并行加速比性能. 同时，我们在后续的研究中采用列存储技术存储巨大的事实表，能够进一步提高并行 OLAP 的处理性能，同时保证良好的并行加速比性能和进一步提高整体查询处理性能.

5　结束语

大数据集上的 OLAP 处理必然由昂贵的高端服务器处理走向廉价的集群处理模式. 集群并行 OLAP 处理主要的瓶颈在于 OLAP 算法的可并行性，并行 OLAP 处理时的 data motion（数据迁移）代价、本地处理时的 I/O 代价、多表连接操作时的查询处理代价.

本文提出了 one-size-fits-all 模式的并行 OLAP 处理，利用现代计算机的大内存容量，将连接依赖数据集通过列存储、压缩、谓词向量等技术最小化其内存存储空间代价，使其能够满足内存直接访问，从而使不同的 SPJGA 类 OLAP 查询能够规范化到统一的 DDTA-OLAP 处理模型中，最小化 OLAP 查询处理时连接数据依赖数据集的 data motion（数据迁移）代价和基于多表连接的 OLAP 处理代价，使 OLAP 算法的并行度得以提升.

本文研究的重点是大数据集在集群环境下的并行处理框架，充分利用大容量内存简化 OLAP 算法，并实现内存 OLAP 处理，从而使并行 OLAP 算法具有可扩展性. 在此框架下，可以进一步结合列存储、内存查询处理、多核并行处理等技术进一步提高本地 OLAP 查询处理的性能，提高整体的并行 OLAP 查询处理性能.

SPJGA 类并行 OLAP 查询作为通用 OLAP 中最基础、代价最大的处理子任务，其良好的并行处理性能是提高通用 OLAP 并行处理性能的关键. SPJGA-OLAP 可以作为其它并行数据库系统中的并行处理模块来加速并行数据库的多表连接查询子任务的性能. 我们将在未来的工作中将 SPJGA-OLAP 嵌入到其它并行数据库系统中（如 PostgreSQL-XC）来提高其并行 OLAP 处理性能.

参 考 文 献

[1] O'Neil Patrick E, O'Neil Elizabeth J, Chen Xue-Dong, Revilak Stephen. The star schema benchmark and augmented fact table indexing//Proceedings of the TPCTC. Lyon, France, 2009: 237-252

[2] Han Wook-Shin, Ng Jack, Markl Volker, Kache Holger, Kandil Mokhtar. Progressive optimization in a shared-nothing parallel database//Proceedings of the SIGMOD. Beijing, China, 2007: 809-820

[3] Abadi Daniel J, Rasin Alexander, Silberschatz Avi. HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads//Proceedings of the VLDB. Lyon, France, 2009, 2(1): 922-933

[4] Lima Alexandre A B, Furtado Camille, Valduriez Patrick, Mattoso Marta. Parallel OLAP query processing in database clusters with data replication. Distributed and Parallel Databases, 2009, 25(1-2): 97-123

[5] Furtado Pedro: Model and procedure for performance and availability-wise parallel warehouses. Distributed and Parallel Databases, 2009, 25(1-2): 71-96

[6] Yang Christopher, Yen Christine, Tan Ceryen, Madden Samuel. Osprey: Implementing MapReduce-style fault tolerance in a shared-nothing distributed database//Proceedings of the ICDE. Long Beach, California, USA, 2010: 657-668

[7] Chen Songting. Cheetah: A high performance, custom data warehouse on top of MapReduce//Proceedings of the VLDB. Singapore, 2010, 3(2): 1459-1468

[8] SAP NetWeaver: A Complete Platform for Large-Scale Business Intelligence. Winter Corporation White Paper. May, 2005

[9] The Vertica Analytic Database: Rethinking Data Warehouse Architecture. Winter Corporation White Paper. May, 2005

[10] MacNicol R, French B. Sybase IQ multiplex-designed for analytics//Proceedings of the VLDB. Toronto, Canada, 2004: 1227-1230

[11] Stonebraker Michael, Abadi Daniel J, Batkin Adam, Chen Xuedong et al. C-Store: A column-oriented DBMS//Proceedings of VLDB. Trondheim, Norway, 2005: 553-564

[12] Zhang Yansong, Hu Wei, Wang Shan. MOSS-DB: A hardware-aware OLAP database//Proc of WAIM. Jiuzhaigou, China, 2010: 582-594

[13] Binnig Carsten, Hildenbrand Stefan, Färber Franz. Dictionary-based order-preserving string compression for main memory column stores//Proceedings of the SIGMOD. Rhode Island, USA, 2009: 283-296

ZHANG Yan-Song, born in 1973, Ph. D., associate professor. His current research interests include main-memory database, OLAP and high performance databases.

JIAO Min, born in 1975, Ph. D. candidate. Her current research interests include main-memory database, OLAP and high performance databases.

WANG Zhan-Wei, born in 1985, M. S.. His current research interests include main-memory database, OLAP and high performance databases.

WANG Shan, born in 1944, professor, Ph. D. supervisor. Her research interests include high performance database, data warehouse and knowledge engineering.

ZHOU Xuan, born in 1979, Ph. D., associate professor. His current research interests include IR, and high performance databases.

Background

Big data analysis requires scalable processing framework and powerful processing engine to enhance processing capability and performance. Traditional parallel databases(Teradata, ParAccel etc.) commonly employ MPP architecture for large scale processing which rely on specially designed hardware for high performance. MapReduce(Hadoop) presents a simple but high scalable framework for big data analysis, but the materialization mechanism in key/value processing makes MapReduce I/O bound, and the iterative MapReduce processing for complex processing produces large cost for processing and network transmission. In this paper, we propose a memory locality centric OLAP algorithm (predicate-vector based DDTA-OLAP) to normalize complex OLAP queries with star-join as an in-memory pipeline processing and extend this algorithm into large scale moderate cluster framework. The reverted star schema storage model is employed for efficient storage and updates, the diverse analytic processing is normalized as four stages with optimizations for data motion and processing. So the complex OLAP can be normalized as one-size-fits-all model for large scale big data analysis. This work is supported by the Important National Science & Technology Specific Projects of China ("HGJ" Projects, Grant No. 2010ZX01042-001-002), the National Natural Science Foundation of China (Grant No. 61070054), the Fundamental Research Funds for the Central Universities (the Research Funds of Renmin University of China, Grant No. 10XNI018), the Renmin University of China (Grant No. 10XNB053), and the Graduate Science Foundation of Renmin University of China (Grant No. 10XNH096 and No. 11XNH120). The target of research is to establish a memory locality centric cluster OLAP system which can provide both high performance and high scalability. We have developed ScaMMDB and ScaMMDBII for scalable main memory database and have published more than ten related papers, this work focuses on the key framework for large scale analytical processing in big data era.