一种并行处理 Skyline 查询的有效方法

黄震华[1,2] 向 阳[1] 薛永生[3] 赵 杠[4]

摘 要 Skyline 查询是近年来数据库领域的一个研究重点和热点, 这主要是因为 Skyline 查询在许多领域有着广泛的应用. 现有的工作大都集中于单处理机环境, 然而, 由于 Skyline 查询是 CPU 敏感的, 因此, 在实际应用中, 现有的方法具有很大的局限性. 基于此, 提出一种有效降低处理 Skyline 查询时间开销的并行算法 PAPSQ (Parallel algorithm for processing skyline queries). 算法有机结合多维数据对象的自身特性和通用多处理机系统的实施优点, 以 Skyline 查询搜索偏序格为底层结构, 利用多维数据对象的同胚评估值和偏序格加权技术来有效提高并行处理 Skyline 查询的效率. 实验评估表明, PAPSQ 算法具有有效性和实用性.

关键词 Skyline 查询, 并行处理, 搜索偏序格, 查询优化, 性能评估

DOI 10.3724/SP.J.1004.2010.00968

An Efficient Method for Parallel Processing of Skyline Queries

HUANG Zhen-Hua[1,2] XIANG Yang[1] XUE Yong-Sheng[3] ZHAO Gang[4]

Abstract Skyline query processing has recently received a lot of attention in database community. Most related works focus on the single processor environment. However, since skyline queries are CPU-sensitive and time costly, the existing methods have prodigious limitations in real applications. Motivated by the above fact, in this paper, we propose an efficient method for parallel processing of skyline queries, called parallel algorithm for processing skyline queries (PAPSQ). The PAPSQ algorithm seamlessly combines the speciality of multidimensional data objects and the implementary advantage of universal multiprocessor systems. Specially, the PAPSQ algorithm takes the partial order lattice of skyline queries as substrate structure, and utilizes the homeomorphism evaluation of multidimensional data objects and the weighted technology to markedly improve the performance of parallel processing of skyline queries. Furthermore, detailed theoretical analyses and extensive experiments are given to demonstrate that the algorithm is both efficient and effective.

Key words Skyline queries, parallel processing, search lattice, query optimization, performance evaluation

Skyline 查询处理技术是近年来数据库领域的一个研究重点和热点. 这主要是因为 Skyline 查询在许多领域有着广泛的应用, 如: 多标准决策支持系统[1], 城市导航系统[2], 数据挖掘和可视化[3] 以及用户偏好查询[4] 等. 给定有限规模的对象集合 $SO = O^1, \cdots, O^n$, 其中 $O^i (i \in [0, n])$ 具有 δ 维属性, 每维属性衡量它的一个子特征 (比如距离、价格等); Skyline 查询就是在 SO 中找出满足如下条件的每个对象 p: 不存在 SO 中的某一对象 r, 使得 r 在所有 δ 维上的取值均不比 p 差, 并且至少在一个维上的取值比 p 优. 显然, 不在 Skyline 查询结果中的那些对象不影响用户的最终选择.

Skyline 查询最早由 Borzsonyi 等[1] 引进到数据库领域中, 并提出两个可行的查询算法: 块嵌套循环 (Block nested loop, BNL) 算法以及分区回归 (Divide and conquer, DC) 算法. 随后, Chomicki 等[2, 5] 在 BNL 算法的基础上提出一种先进行对象排序, 再进行比较的查询方法, 即排序过滤 (Sort filter skyline, SFS) 算法. 基于索引的方法最早由 Kossmann 等[3] 提出. 在文中, 作者给出一种基于 R-树索引的计算方法, 即最邻近 (Nearest neighbor, NN) 算法. 而 Papadias 等[4, 6] 指出 NN 算法的缺陷, 并提出一种基于排序 R-树节点的方法: 分支约束 (Branch and bound skyline, BBS) 算法. BBS 算法克服了 NN 算法冗余比较节点的不足, 且比 NN 算法具有更强的剪枝能力. 实验评估表明, BBS 算法具有最好的查询效率. Sharifzadeh 等[7] 首次在空

收稿日期 2008-09-22 录用日期 2010-03-17
Manuscript received September 22 2008; accepted March 17, 2009
国家高技术研究发展计划 (863 计划) (2008AA04Z106), 国家自然科学基金 (60903032), 教育部博士点基金 (20090072120056), 同济大学青年优秀人才基金 (0800219093) 资助
Supported by National High Technology Research and Development Program of China (863 Program) (2008AA04Z106), National Natural Science Foundation of China (60903032), the Ph. D. Program Foundation of Ministry of Education of China (20090072120056), and the Outstanding Young Foundation of Tongji University (0800219093)

1. 同济大学电子与信息工程学院 上海 200092 2. 同济大学嵌入式系统与服务计算教育部重点实验室 上海 200092 3. 厦门大学信息科学与技术学院 厦门 361005 4. 复旦大学信息科学与工程学院 上海 200433
1. School of Electronics and Information, Tongji University, Shanghai 200092 2. Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, 200092 3. School of Information Science and Technology, Xiamen University, Xiamen 361005 4. School of Information Science and Engineering, Fudan University, Shanghai 200433

间数据库上研究 Skyline 查询, 并给出空间 Skyline 查询的概念. Li 等[8] 将 Skyline 查询技术引进到时间序列研究邻域[9] 来克服维度危机问题[10]. Pei 等[11] 考虑如何在不确定的数据集上进行有效的 Skyline 查询计算, 并给出一种新颖的概率 Skyline 模型来解决当对象出现的概率不确定的情况. Tao 等[12] 首次考虑当数据对象动态更新时, 如何高效维护 Skyline 查询结果集. 而 Huang 等[13] 首次考虑在多用户环境中, 如何有效处理和优化多个 Skyline 查询. 值得注意的是, 现有这些研究工作[1−8, 11−13] 均集中于单处理机环境. 然而, 由于 Skyline 查询是 CPU 敏感的, 它需要花费大量的时间开销, 特别地, 当数据量较大且 Skyline 查询个数较多时, 现有方法的效率将极其低下.

随着并行计算算法的完善以及廉价而功能强大的多处理机系统的成熟, 使得采用多处理系统来并行处理 Skyline 查询成为克服现有方法低效率的首选技术. 目前, 只有文献 [14 − 15] 考虑并行处理 Skyline 查询. Wu 等[14] 研究如何在具有内容寻址网络 (Content-addressable network, CAN) 体系结构的分布式网络[16] 中进行有效的 Skyline 查询. 作者给出一种基于递归区间划分以及动态区间编码的分布式算法 (Distributed skyline computation, DSC) 来高效返回 Skyline 集合. Cosgaya-Lozano 等[15] 考虑将原始数据集水平划分为多个子集, 并将每个子集存储于不同的处理机上. 作者给出一个两阶段的处理方法 (Parallel skyline queries, PSQ) 来有效返回最终的查询结果: 在第一阶段中, 各处理机返回局部的 Skyline 集合; 并在第二阶段中, 算法合并所有局部的 Skyline 集合, 并最终返回全局的 Skyline 集合. 然而, 文献 [14−15] 至少存在如下三个缺陷: 1) 文献 [14−15] 所给的算法需要将海量的对象从外部存储设备调入内存, 而这些对象的维度通常较多. 由于它们没有采取任何有效的对象编码机制, 因此, 用户将花费大量的时间在 I/O 开销上. 2) 因为在实际应用中, 不同的用户往往会有不同的考察角度 (即维空间), 而针对每个维空间, 用户需要提交一个该维空间上的 Skyline 查询, 然而文献 [14−15] 只考虑处理单个 Skyline 查询, 而没有考虑如何同时优化多个 Skyline 查询的处理效率. 因此在实际应用中, 文献 [14−15] 给出的算法具有很大的局限性. 3) 由于文献 [14−15] 所给出的算法只考虑处理单个 Skyline 查询, 因此, 它们的处理机分配策略不适合系统中同时存在多个 Skyline 查询的情况, 从而导致了文献 [14−15] 中的算法无法基于最佳的分配策略来降低并行处理多个 Skyline 查询的 CPU 开销.

为此, 本文从优化 Skyline 查询在多个处理机间的分组出发, 提出了一种新的并行处理 Skyline 查询的有效算法 PAPSQ (Parallel algorithm for processing skyline queries). PAPSQ 算法充分考虑了多维数据对象的存储机制和多处理机分布系统的结构特点, 在 Skyline 查询偏序格逻辑结构的基础上, 采用多维数据对象的层次联合代理和对 Skyline 查询偏序格进行加权的方法, 使得 Skyline 查询在多个处理机间的分配达到最佳状态, 从而在分割输入数据的同时, 提高了并行处理 Skyline 查询的效率. 实验评估表明, PAPSQ 算法具有有效性和实用性.

本文接下来是这样组织的: 第 1 节给出多维数据对象的层次联合代理; 第 2 节给出 Skyline 查询语义以及 Skyline 查询偏序格; 第 3 节描述 PAPSQ 算法; 第 4 节给出具体的实验评估; 最后, 第 5 节对全文工作进行总结, 并给出将来的后续研究工作.

1 多维数据对象的同胚评估值

当数据量较大, 并且对象的维数较多时, 如果直接将对象本身调入内存, 那么势必造成巨大的 I/O 开销和内存空间的消耗. 为此, 我们给出层次联合代理的概念对多维数据对象进行编码.

定义 1. 多维数据对象集合的底层结构是一同胚树 $HomTree$, 它以 ALL 为根节点的有向非循环图 (Directed acyclic graph, DAG), 可用二元组 $\Gamma = (\Psi, \wp)$ 表示. 其中, $\Psi = \{ALL, \phi_1, \cdots, \phi_n\}$ 是 Γ 中的节点集合, $\wp = \{\vartheta_{ij} | \vartheta_{ij}$表示 Γ 中有$\phi_i \rightarrow \phi_j\}$ 是 Γ 中的有向边集合.

定义 2. 假定同胚树 $HomTree$ 上的特定维度 A 的取值范围为 $\Im = \{\varpi_1, \varpi_2, \cdots, \varpi_t\}$, 它具有的同胚数为 λ, 那么它有 $\lambda + k$ 个同胚子树, 记为 $\Pi = \{\alpha^0, \alpha^1, \cdots, \alpha^\lambda\}$. 如果 $\zeta = \{\theta_1, \theta_2, \cdots, \theta_m\}$ 满足下列条件, 则称 ζ 为同胚树 $HomTree$ 相对应的第 i 个同胚子树 α^i $(0 \le i \le \lambda)$ 的胚簇:

1) $\zeta \cap \Im = \emptyset$;

2) $\prod_{i=1}^{t} \varpi_i = \zeta$;

3) $\Im(HomTree(\lambda{+}1)) \wedge \Im(HomTree(\lambda{+}2)) \wedge \cdots \wedge \Im(HomTree(\lambda + t)) \neq \emptyset$;

4) 对 $\forall \theta_x, \theta_y \in \alpha^i$ 且 $\theta_x \neq \theta_y$, 则 $\theta_x \cap \theta_y = \emptyset$.

定义 3. 假定多维数据对象集合所对应的同胚树为 $HomTree$, 那么 $HomTree$ 树节点 $\wp$ 的所有子节点组成的集合可表示为:

$$Child\_nodes(\wp) = \left\{ \wp_c | \wp \cap \Im = \wp_c \wedge \prod_{i=1}^{t} \varpi_i = \wp \wedge \wp_c \right\}$$

定义 4. 假定多维数据对象集合所对应的同胚

树为 $HomTree$, 那么 $HomTree$ 树节点 $\wp$ 的所有父节点组成的集合可表示为:

$$Parent\_nodes(\wp_p)=\left\{\wp_p|\wp_p\cap\Im=\wp_c\wedge\prod_{i=1}^{t}\varpi_i=\wp_p\wedge\wp_c\right\}$$

定义 5. 假定集合 $Child\_nodes(\wp)$ 的基数为 CA, 那么根据 Karp-Luby 随机理论[17−18], 定义 Davis-Putnam 函数 $DP(\wp)$:

$$Child\_nodes(\wp)\rightarrow\left\{0,\sum_{i=1}^{2}\frac{1}{i(i+1)},\cdots,\sum_{i=CA-1}^{CA}\frac{1}{i(i+1)}\right\}$$

定义 6. 假定集合 $Child\_nodes(\wp)$ 上 $DP(\wp)$ 函数为 $Child\_nodes(\wp)\rightarrow\{0,\sum_{i=1}^{2}\frac{1}{i(i+1)},\cdots,\sum_{i=CA-1}^{CA}\frac{1}{i(i+1)}\}$, 那么定义 $DP(\wp)$ 函数概化为 $DP(\wp)'$: $Child\_nodes(\wp\cap\Im)\rightarrow\{0,\sum_{i=1}^{2}\frac{1}{i(i+1)}\oplus\theta_1,\cdots,\sum_{i=CA-1}^{CA}\frac{1}{i(i+1)}\oplus\theta_{CA-1}\}$.

文献 [12] 研究并提出了基于某一特定维的同胚评估值, 然而 Skyline 查询通常涉及多个维度, 所以必须将某一特定维的同胚评估值扩展为能够适用于多个维度的情况.

定义 7. 假定多维数据对象集合所对应的同胚树为 $HomTree$, 它具有 λ 个同胚子树, 记为 $\alpha^0,\alpha^1,\cdots,\alpha^\lambda$, $\alpha^i(0\le i\le\lambda)$ 的 σ 个胚簇为 $\aleph(i,1),\aleph(i,2),\cdots,\aleph(i,\sigma)$, 而 $\aleph(i,j)$ 的子节点集合 $Child\_nodes(\aleph(i,j))$ 的 $DP(\aleph(i,j))$ 函数概化为 $DP(\aleph(i,j))'$, 则 $HomTree$ 上各节点的同胚评估值为

$$Ho\_Ev(HomTree,\aleph(i,j))=\begin{cases}DP(\aleph(i,j))'\aleph(i,j)\Theta DP(\aleph(i,j))(\aleph(i{+}1,j)), & \text{if } i{=}1\\ DP(\aleph(i,j))'\aleph(i{+}1,j)\Theta DP(\aleph(i,j))(\aleph(i,j{+}1)), & \text{if } i{=}2\\ DP(\aleph(i{+}1,j))'\aleph(i,j{+}1)\Theta DP(\aleph(i{-}1,j))(\aleph(i,j)), & \text{if } i{>}2\end{cases}$$

定理 1. 同胚树 $HomTree$ 上的每个节点的同胚评估值存在且唯一, 并且随着树深度的递增, 其节点的同胚评估值单调增大.

定义 8. 假定多维数据对象集合所对应的同胚树为 $HomTree$, 它上面存在某一搜索路径 Ξ 深度优先遍历 a 个节点 $node_1,node_2,\cdots,node_a$, 并且对于每个节点 $node_i(1\le i\le a)$, 它的子节点集合 $Child\_nodes(node_i)$ 所对应的 Davis-Putnam 函数概化为 $OP(node_i)'$, 那么可以定义搜索路径 Ξ 的同胚评估值为: $Ho\_Ev(HomTree,\Xi)=\prod_{i=1}^{a}(DP(node_i)'[a-i]\oplus Ho\_Ev(HomTree,node_i))$.

2 Skyline 查询语义及其偏序格

假定对象集合 Ω 共具有 K 个维度 $d_1,\cdots,d_K$, 那么每种维度的组合称为一个维空间. 不难看出, 一个包含 K 个维度的对象集合, 它最多具有 2^K-1 个维空间 ($\emptyset$ 除外), 因此系统中最多同时存在 2^K-1 个不同的 Skyline 查询. 与维空间 $V=d_1,\cdots,d_v$ 上 Skyline 查询的每个维 $d_j(1\le j\le v)$ 相关的操作涉及两个关键字 MAX 和 MIN. 为了简单起见, 把 MAX 和 MIN 这两个关键字用操作符 $>$ 和 $<$ 表示, 并用符号 $\otimes$ 通指这两个操作符, 其中, MAX (MIN) 表示该维的取值越大 (小) 越好.

定义 9. 对于维空间 $V=\{d_1,\cdots,d_v\}$ 上的两个对象 $P(d_{1p},\cdots,d_{vp})$ 和 $Q(d_{1p},\cdots,d_{vp})$, 如果它们满足下列两个条件, 那么称对象 P 在维空间 V 上支配对象 Q:

1) $\forall j(d_{jp}=d_{jq}\vee d_{jp}\otimes=\text{true}),\quad 1\le j\le v$;

2) $\exists j(d_{jp}\otimes=\text{true})),\quad 1\le j\le v$.

不难看出, 如果对象 P 在维空间 V 上支配对象 Q, 那么在 v 个度量指标的组合上, 对象 P 的各指标均不比 Q 差, 并且至少在一个度量指标上比 Q 优秀.

简单起见, 把对象 P 在维空间 V 上支配 Q, 记为 $Q\prec_V P$, 在不发生混淆的情况下, 省去字符 "V", 而写成 $Q\prec P$. 并且, 把 P 在子空间 V 上不支配 Q, 记为 $Q\not\prec_V P$.

定义 10. 设 Ω 是 K 维空间的对象全集, 如果下列条件成立, 称 $L=P_1\prec_V P_2\prec_V\cdots\prec_V P_m$ 为维空间 $V(|V|\le K)$ 上的一条支配链, 并记 $S(L)=\{P_1,P_2,\cdots,P_m\}$:

1) $\forall j(p_j\in\Omega)$;

2) $\forall j,p_j\prec_V P_{j+1}\wedge(\neg\exists Q,Q\in\Omega\wedge P_j\prec_V Q\wedge Q\prec_V P_{j+1})$;

3) $\neg\exists L',S(L')\subseteq\Omega\wedge S(L)\subset S(L')$.

定理 2. 维空间 V 上的 "$\prec$" 关系满足严格部分序.

定义 11. 设 Ω 是 K 维空间的对象全集, 并且维空间 $V(|V|\le K)$ 上存在 β 条维空间 V 上的支配链: $P_1^1\prec P_2^1\prec\cdots\prec P_{m1}^1, P_1^2\prec P_2^2\prec\cdots\prec P_{m2}^2,\cdots,P_1^\beta\prec P_2^\beta\prec\cdots\prec P_{m2}^\beta$, 则定义维空间 V 上的 Skyline 查询结果集为 $\nabla^V(\Omega)=\{P_{m1}^1,P_{m2}^2,\cdots,P_{m2}^\beta\}$.

接下来, 给出 Skyline 查询偏序格的概念. 假定用户发出 $u(u\le 2^K-1)$ 个维空间 Skyline 查询 $SKYQ(V_1),SKYQ(V_2),\cdots,SKYQ(V_u)$, 其中, $V_i(1\le i\le u)$ 表示维空间.

定义 12. Skyline 查询偏序格是一个有限非循环图 $J=(N,E)$, 节点集 N $\{V_1,V_2,V_u\}$. 而每条有

向边 $\langle V_i, V_j\rangle \in E$ 表示维空间 V_i 包含 V_j 的所有维度, 即 $V_j \subset V_i$; 且不存在节点 V_x, 使得 $V_j \subset V_x \subset V_i$ 成立. 我们把节点 V_i 称为节点 V_j 的父节点.

例如, 对象共具有 $K = 4$ 个维度 $ABCD$, 并且用户发出 11 个维空间上的 Skyline 查询, 这 11 个维空间分别为 $V_1 = ABCD, V_2 = ABC, V_3 = ABD, V_4 = ACD, V_5 = BCD, V_6 = AB, V_7 = AC, V_8 = AD, V_9 = BC, V_{10} = BD, V_{11} = CD$, 图 1 显示了这 11 个维空间 Skyline 查询构成的偏序格.

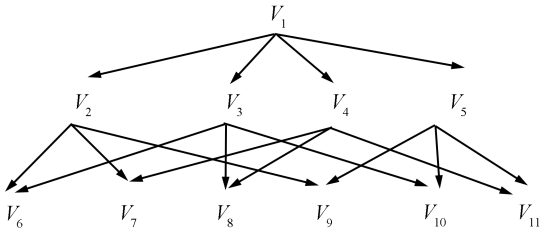


图 1　Syline 查询偏序格

Fig. 1　The partial lattice of skyline queries

定义 13. 设 $J = (N, E)$ 是加权有向连通图, $\chi_1, \chi_2, \cdots, \chi_p$ 为图 J 的 p 棵有向生成树. $\chi_i\,(1 \le i \le p)$ 中的树枝的权之和为 χ_i 的权, 记为 $W(\chi_i) = \sum_{e_jk \in \chi_i} w(e_{jk})$, $w(e_{jk})$ 为树枝 e_{jk} 的权. 如果 $W(\chi_i) = \mathrm{MIN}(\{W(\chi_1), W(\chi_2), \cdots, W(\chi_p)\})$, 则称 $\chi_i\,(1 \le i \le p)$ 为图 J 的有向最小生成树.

目前可用于有效获取最小生成树的经典算法有 3 个, 分别为 KRUSKAL 算法, PRIM 算法和 SOLLIN 算法[19].

3　PAPSQ 算法描述

近年来, 人们在 Skyline 查询方面做了很多研究工作, 并取得了一定的成果. 然而, 目前大部分工作均集中于单处理机环境[1−8, 11−13]. 由于 Skyline 查询是 CPU 敏感的, 它需要花费大量的时间开销, 特别地, 当数据量较大, 且 Skyline 查询个数较多时, 现有方法的效率将极其低下. 文献 [14 – 15] 虽然考虑了并行处理 Skyline 查询, 然而, 它们具有三个严重的性能缺陷 (详见前言), 因此, 在实际应用中, 它们具有很大的局限性.

本文从优化 Skyline 查询在多个处理机间的分组出发, 提出了一种新颖的并行处理 Skyline 查询的有效算法 PAPSQ. PAPSQ 算法包括两个阶段的工作: 在第一阶段中, 构建用于评估 Skyline 查询的模型; 而在第二阶段中, 将多个处理机分配给由第一阶段得到的模型, 并让处理机系统并行对各节点集合执行 Skyline 查询.

3.1　Skyline 查询评估模型

由第 2 节可知, 可以通过构建 Skyline 查询的偏序格 $\wp$ 来处理用户提交的多个不同维空间上的 Skyline 查询, 而且在 $\wp$ 中, 子节点的查询输入数据集是父节点的查询输出结果集.

在这一小节中, 我们给 Skyline 偏序格 $\wp$ 的每条边赋上某一权值. 由于 PAPSQ 算法的优化目标是降低并行处理多个 Skyline 查询的总时间开销, 因此在 $\wp$ 中, 用父节点产生子节点的时间开销作为权值.

在 PAPSQ 算法的第一阶段中, 使用目前比较流行的采样评估器 APA1+[20] 作为数据采样算法. 根据文献 [20] 对多种数据分布的数据集进行仿真试验, 发现 APA1+ 采样评估器能够很好地体现整个数据集的特征, 而且具有精度 $e/(e + 0.5) \approx 84.3\,\%$ 的下界保证. 另外, 依据文献 [20] 的优化建议, 选取原始数据的 1/10 作为采样数据量.

下面给出 PAPSQ 算法第一阶段的处理步骤.

算法 1. 构造 Skyline 查询评估模型

输入: 多维数据对象集合的同胚评估值文件 Z, a 个处理机的集合 $\{\Delta_1, \cdots, \Delta_a\}$, u 个维空间为 $V_1, \cdots, V_u$ 的 Skyline 查询;

输出: 各 Skyline 查询评估模型 χ;

方法:

将多维数据对象集合的同胚评估值文件 Z 水平分割成 t 个长度一致的子文件 $Z^1, \cdots, Z^t$;

For $i \in [1, t]$

　将 Z^i 调入处理机 Δ_i 所在的主存;

　根据文献 [21] 构建记为主存的位图索引 BI_i;

　构建 Skyline 查询的偏序格 $P_i = (N_i, E_i)$;

　For 偏序格 P_i 上的每条边 PE_j\{

　　依据位图索引 BI_i, 将磁盘数据的 1/10 调入主存;

　　通过有效的采样评估算法 APA1+[20] 来近似计算边 PE_j 的评估值;\}

　使用 SOLLIN 算法在多项式时间内生成总的 Skyline 查询评估模型 χ.

3.2　多处理机实施 Skyline 查询

PAPSQ 算法的第二阶段是使用 Skyline 查询评估模型 χ 来并行实施 Skyline 查询. 在 χ 的基础上, 通过多处理机系统精确获取 Skyline 集合.

定理 3. 假定分配给子 Skyline 查询评估模型 $ST_1, ST_2, \cdots, ST_m$ 的处理机总个数为 PTM, 而多处理机系统的处理机总个数为 RTM, 那么有:

$$PTM \in \left\lfloor \frac{RTM}{\sum_{i=1}^{m} API(ST_i)} \times \right.$$

$$RTM\sum_{i=1}^{m}API(ST_i)\Bigg]$$

基于定理 3, 下面给出 PAPSQ 算法第二阶段的处理步骤.

算法 2. 多处理机实施 Skyline 查询 PAPSQ-II($\chi,\{\Delta_1,\cdots,\Delta_r\}$)

方法:

/* χ 为第一阶段产生的 Skyline 查询评估模型, $\{\Delta_1,\cdots,\Delta_r\}$ 为多处理机系统中 r 个实际可用的处理机 */

将 Skyline 查询评估模型 χ 调入内存;

获取 χ 的最顶层节点 rt;

分配处理机 $\{\Delta_1,\cdots,\Delta_r\}$ 给根节点 rt, 并计算出 rt 的时间开销 $TC(rt)$;

获取 Skyline 查询评估模型 χ 的节点树 $|\chi|$;

$AMT=\sqrt{(API(\chi)-TC(rt))*|chi|}/a$;

获取 rt 的子节点集合 $\Xi(rt)=\{c_1,\cdots,c_m\}$;

For $i\in[1,m]$

  获取以 c_i 为根的子树 ST_i;

$\alpha=|chi|$;

Repeat

  $\alpha=\alpha+1$;

  $\beta=(API(ST_\alpha)-\sqrt{API(ST_{\alpha-1})})/ATM$;

  If ($\alpha>0$ and $\beta\neq 0$), Then

    分配 β 个处理机给子评估模型 ST_α;

    删除子评估模型 ST_α;

  If ($\alpha>0$ and $\beta>1$), Then

    PAPSQ-II($ST_\alpha,\{\Delta_1,\cdots,\Delta_\alpha\}$);

  Else

    $\alpha'=\alpha-1$;

    $\beta'=\beta\times 1.5$;

    $U=API(ST_\alpha)$;

    While ($\beta'=0$) 且 ($\alpha'\neq m$) Do

      $\alpha=\lfloor\alpha'/\sqrt{API(ST_{\alpha-1})}\rfloor$;

  使用 CDCA[13] 算法获取各节点的 Skyline 集合;

Until $\alpha=m$.

定理 4. 在 PAPSQ 算法的第二阶段结束时, 当 $PTM=RTM/\sum_{i=1}^{m}API(ST_i)$ 时, 有 $\beta'=\sqrt{API(\chi)/a}$; 当 $PTM=RTM\sum_{i=1}^{m}API(ST_i)$ 时, 有 $\beta'=\sqrt[3]{API(\chi)/a}$.

4 算法实验评估

在这一节中, 通过具体的实验来评估本文所给算法 PAPSQ 的有效性. 实验评估用到两类数据集是: 1) 综合数据集 (Synthetical dataset, SND), 由文献 [1] 中的数据产生器生成. SND 数据集共有 8 个维度, 每个维度的数据类型为浮点型, 占 4 个字节. 在实验中, SND 数据集的基数在 $2\times10^5\sim1\times10^6$ 间变化. 2) 实际数据集: 医学院数据集 (Chiba dataset, CDS), 为 Chiba 医学院从 1980 年开始到 1999 年为止收集的胶原质疾病统计数据. CDS 数据集包含 198 537 条记录, 每条记录总共具有 8 个维度, 每个维度的数据类型为浮点型, 占 4 个字节. CDS 数据集可以在 www.lisp.vse.cz 上下载. 在实验中, 产生 100 个维空间上的 Skyline 查询, 其中, 8 维空间 1 个、7 维空间 4 个、6 维空间 10 个、5 维空间 20 个、4 维空间 25 个、3 维空间 20 个以及 2 维空间 20 个.

与 PAPSQ 算法比较的有文献 [14] 所给的 DSC 算法以及文献 [15] 所给的 PSQ 算法. 由于系统中共有 100 个不同维空间上的 Skyline 查询, 而 DSC 算法和 PSQ 算法均是针对单个维空间来设计的, 因此在实验中, 分别各运行 100 次 DSC 算法和 PSQ 算法, 每次针对一个维空间. 在实验中, 对于 DSC 算法, 根据文献 [14] 的实现思想, 我们将数据复制在所有处理机上, 即每个处理机上的数据均是相同的. 算法根据区域间的支配关系将系统中的处理机按区域进行划分, 并对每个区域进行动态编码, 然后基于编码值将所有区域组织成一个区域访问顺序拓扑结构图, 并基于该拓朴结构图来处理相应区域内各处理机上的数据. 对于 PSQ 算法, 根据文献 [15] 的实现思想, 我们将数据等量水平分区存储在不同的处理机上, 每个处理机存储的数据不重叠. 算法通过两个阶段来返回每个固定维空间上的 Skyline 集合. 在第一阶段中, 算法获取每个处理机上的局部 Skyline 集合, 而在第二阶段中, 算法将各处理机上的 Skyline 集合传送到一台预定的处理机上, 并再次对传送过来的所有局部 Skyline 集合进行计算, 获取最终的全局 Skyline 集合.

实验的硬件设备是共享内存的集群计算机系统, 该系统里有 10 个可用处理机, 它们共享 16 G 内存. 所有程序均在 JBuilder 9 中调试通过.

4.1 综合数据集 SND 上的实验评估

在这一小节中, 具体评估本文算法 PAPSQ 在综合数据集 SND 上的性能. 实验分两组进行.

在第一组实验中, 表基数固定为 8×10^5, 处理机个数在 $2\sim10$ 间变化. 图 2 显示了该组的实验结果.

从图 2 中可以看出, 本文算法 PAPSQ 的性能在各种情况下均优于 DSC 算法和 PSQ 算法, 同时, PSQ 算法的性能优于 DSC 算法. 这主要是因为 PAPSQ 算法充分考虑了多维数据对象的存储机制和多处理机分布系统的结构特点, 并且能够使得这 100 个维空间上 Skyline 查询在多个处理机间的分配达到最佳状态. 而 DSC 算法和 PSQ 算法没有考虑这方面的因素. 另一方面, 我们注意到当处理机个数逐渐增多时, PAPSQ 算法的优越程度越来越高. 例如在图 2 中, 当处理机个数为 2

时, DSC 算法、PSQ 算法和 PAPSQ 算法的运行时间分别为 1 946.8 秒、1 804.3 秒和 1 732.4 秒, 即此时 PAPSQ 算法的运行时间为 DSC 算法 (PSQ 算法) 的 88.9% (95.8%); 而当处理机个数为 10 时, DSC 算法、PSQ 算法和 PAPSQ 算法的运行时间分别为 840.6 秒、644.5 秒和 208.3 秒, 即此时 PAPSQ 算法的运行时间为 DSC 算法 (PSQ 算法) 的 24.8%(32.3%). 这一点也说明了 PAPSQ 算法在处理机个数变化时具有较好的可扩展性.

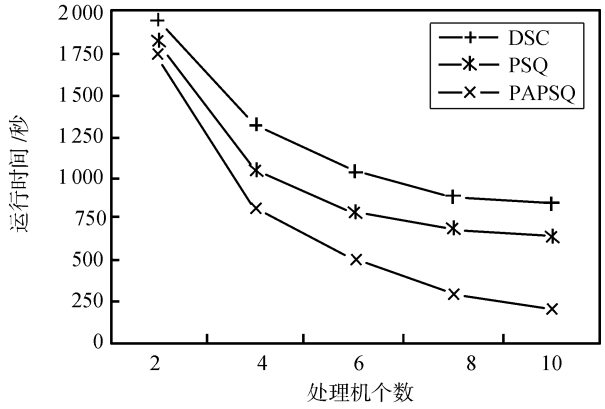


图 2　运行时间随处理机个数变化实验

Fig. 2　Runtime vs. the number of processors

在第二组实验中, 处理机个数固定为 6, 表基数在 $2 \times 10^5 \sim 1 \times 10^6$ 间变化. 图 3 显示了该组的实验结果.

与第一组实验结果类似, 在图 3 中可以看出, 本文算法 PAPSQ 的性能在表基数的各种取值下均优于 DSC 算法和 PSQ 算法, 而 PSQ 算法的性能优于 DSC 算法. 例如在图 2 中, 当处理机个数为 6 时, DSC 算法、PSQ 算法和 PAPSQ 算法的运行时间分别为 760.9 秒、688.7 秒和 352.1 秒. 同时, 当表基数逐渐增大时, PAPSQ 算法的优越程度越来越高. 这主要是因为当表基数增大时, 层次联合代理文件占原始数据表的百分比将越来越小, 因此, PAPSQ 算法能够节省大量的 I/O 开销. 例如在图 3 中, 当表基数为 2 105 时, DSC 算法、PSQ 算法和 PAPSQ 算法的运行时间分别为 468.5 秒、368.6 秒和 256.5 秒, 即此时 PAPSQ 算法的运行时间为 DSC 算法 (PSQ 算法) 的 54.7% (69.6%); 而当表基数为 1 106 时, DSC 算法、PSQ 算法和 PAPSQ 算法的运行时间分别为 1 412.4 秒、1 268.3 秒和 696.2 秒, 即此时 PAPSQ 算法的运行时间为 DSC 算法 (PSQ 算法) 的 49.2%(54.9%). 同样, 这一点也说明了 PAPSQ 算法在表基数变化时具有较好的可扩展性.

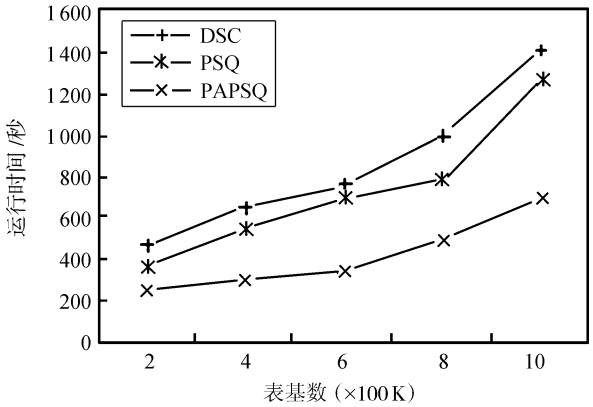


图 3　运行时间随表基数变化实验

Fig. 3　Runtime vs. the cardinality of tables

4.2　实际数据集 CDS 上的实验评估

在这一小节中, 具体评估本文算法 PAPSQ 在实际数据集 SND 上的性能. 由于实际数据集的基数固定, 为 165 937 条记录, 因此, 主要评估当处理机个数变化时, PAPSQ 算法和现有两个算法 DSC、PSQ 的运行时间. 与第 4.1 小节相同, 我们让处理机个数在 2 ∼ 10 间变化. 图 4 显示了该组的实验结果.

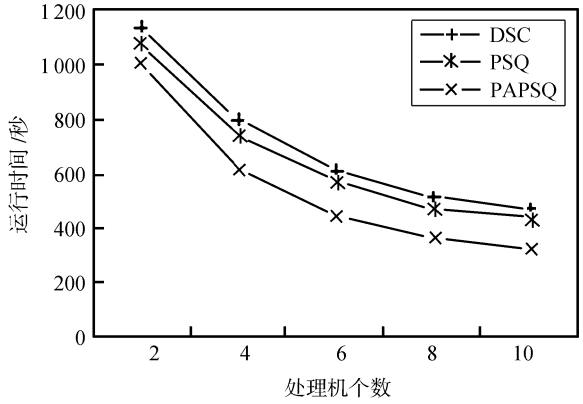


图 4　运行时间随处理机个数变化实验

Fig. 4　Runtime vs. the number of processors

从图 4 中同样可以看出, 在实际数据集上, 本文算法 PAPSQ 的性能在处理机个数的不同取值下均优于 DSC 算法和 PSQ 算法, 而 PSQ 算法的性能优于 DSC 算法. 另一方面, 与图 2 中的曲线图相比发现, 在这组实验中, 当处理机个数逐渐增多时, PAPSQ 算法的优越程度没有图 2 中 PAPSQ 算法的优越程度高. 这主要是因为, 图 2 中实验的综合数据集基数为 8×10^5, 而在这组实验中, 实际数据集的基数仅为 2×10^5. 这一点也很好验证了图 3 实验结论的正确性: 即当表基数逐渐增大时, PAPSQ 算法的优越程度越来越高.

同时, 不难看出, 图 3 中数据集基数为 2×10^5 时的实验环境 (此时处理机个数固定为 6) 与图 4 中

处理机个数为 6 的实验环境 (此时实际数据集基数为个数为 198 537) 基本相同. 然而我们发现, 在图 4 中, DSC 算法、PSQ 算法和 PAPSQ 算法的运行时间分别为 614.4 秒、573.6 秒和 441.6 秒; 而在图 3 中, 这 3 个算法分别只需要 468.5 秒、368.6 秒和 256.5 秒. 这主要是因为实际数据集 CDS 的反相关性比综合数据集 SND 高, 即实际数据集 CDS 中的 Skyline 对象所占的百分比较综合数据集 SND 大, 因此, 它需要更多的运行时间.

5 结论与将来工作

Skyline 查询处理技术是近年来数据库领域的一个研究重点和热点. 本文有机结合多维数据对象的自身特性和通用多处理机系统的实施优点, 以 Skyline 查询搜索偏序格为底层结构, 利用多维数据对象的同胚评估值和偏序格加权技术来有效提高并行处理 Skyline 查询的效率. 为证明本文所给算法 PAPSQ 的有效性和优越性, 我们分别在综合数据集和实际数据集上比较了 PAPSQ 算法和目前流行的两个算法 DSC、PSQ 的运行时间. 实验结果表明, PAPSQ 算法比 DSC 和 PSQ 算法更有效、更具有可扩展性.

将来比较重要和感兴趣的后续工作包括: 1) 设计比层次联合代理更有效的编码机制, 从而进一步降低算法的 I/O 开销和内存消耗; 2) 在构造有向最小生成树方面, 引进参数因子来平衡构造有向最小生成树的时间开销与基于树的 PAPSQ 算法时间开销.

References

1 Borzsonyi S, Kossmann D, Tocker K. The skyline operator. In: Proceedings of the 17th International Conference on Data Engineering. Heidelberg, Germany: IEEE, 2001. 421−430

2 Chomicki J, Godfrey P, Gryz J, Liang D. Skyline with presorting: theory and optimization. In: Proceedings of the International Conference on Intelligent Information Systems. Gdansk, Poland: Springer, 2005. 216−225

3 Kossmann D, Ramsak F, Rost S. Shooting stars in the sky: an online algorithm for skyline queries. In: Proceedings of the 28th International Conference on Very Large Data Bases. Hong Kong, China: VLDB Endowment, 2002. 275−286

4 Papadias D, Tao Y, Fu G, Seeger B. Progressive skyline computation in database systems. *ACM Transactions on Database Systems*, 2005, **30**(1): 41-82

5 Chomicki J, Godfrey P, Gryz J, Liang D. Skyline with presorting. In: Proceedings of the 19th International Conference on Data Engineering. Bangalore, India: IEEE, 2003. 717−719

6 Papadias D, Tao Y, Fu G, Seeger B. An optimal and progressive algorithm for skyline queries. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. San Diego, USA: ACM, 2003. 467−478

7 Sharifzadeh M, Shahabi C. The spatial skyline queries. In: Proceedings of the 32nd International Conference on Very Large Data Bases. Seoul, Korea: VLDB Endowment, 2006. 751−762

8 Li Q, Lopez I F V, Moon B. Skyline index for time series data. *IEEE Transactions on Knowledge and Data Engineering*, 2004, **16**(6): 669−684

9 Chiu B, Keogh E, Lonardi S. Probabilistic discovery of time series motifs. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington D. C., USA: ACM, 2003. 493−498

10 Papadimitriou S, Yu P. Optimal multi-scale patterns in time series streams. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. Chicago, USA: ACM, 2006. 647−658

11 Pei J, Jiang B, Lin X, Yuan Y. Probabilistic skylines on uncertain data. In: Proceedings of the 33rd International Conference on Very Large Database. Vienna, Austria: VLDB Endowment, 2007. 15−26

12 Tao Y, Papadias D. Maintaining sliding window skylines on data streams. *IEEE Transactions on Knowledge and Data Engineering*, 2006, **18**(3): 377−391

13 Huang Z, Guo J, Sun S, Wang W. Efficient optimization of multiple subspace skyline queries. *Journal of Computer Science and Technology*, 2008, **23**(1): 103−111

14 Wu P, Zhang C, Feng Y, Zhao B, Agrawal D, Abbadi A. Parallelizing skyline queries for scalable distribution. In: Proceedings of the 10th International Conference on Extending Database Technology. Munich, Germany: Springer, 2006. 112−130

15 Cosgaya-Lozano A, Rau-Chaplin A, Zeh N. Parallel computation of skyline queries. In: Proceedings of the 21st International Symposium on High Performance Computing Systems and Applications. Saskatchewan, Canada: IEEE, 2007. 12−18

16 Ratnasamy S, Francis P, Handley M, Karp R, Schenker S. A scalable content-addressable network. In: Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. San Diego, USA: ACM, 2001. 161−172

17 Koch C. Approximating predicates and expressive queries on probabilistic databases. In: Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Databases Systems. Vancouver, Canada: ACM, 2008. 99−108

18 Koch C, Olteanu D. Conditioning probabilistic databases. *Proceedings of the VLDB Endowment*, 2008, **1**(1): 313−325

19 Furfaro F, Mazzeo G, Sacca D, Sirangelo C. Compressed hierarchical binary histograms for summarizing multi-dimensional data. *Knowledge and Information Systems*, 2008, **15**(3): 335−380

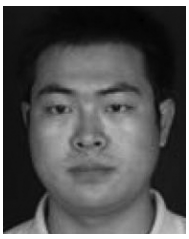
20 Jin R, Glimcher L, Jermaine C, Agrawal G. New sampling-based estimators for OLAP queries. In: Proceedings of the 22nd International Conference on Data Engineering. Atlanta, USA: IEEE, 2006. 18−27

21 Wu K, Koegler W, Chen J, Shoshani A. Using bitmap index for interactive exploration of large datasets. In: Proceedings of the 15th International Conference on Scientific and Statistical Database Management. Cambridge, USA: IEEE, 2003. 65−74

黄震华 同济大学电子信息学院讲师. 2008 年获得复旦大学计算机系博士学位. 主要研究方向为数据仓库、联机分析处理、数据挖掘与知识发现. 本文通信作者. E-mail: jukie.hung@gmail.com
(**HUANG Zhen-Hua** Lecturer at the School of Electronics and Information, Tongji University. He received his Ph. D. degree in computer science from Fudan University in 2008. His research interest covers data warehouse, on-line analysis processing (OLAP) application, data mining, and knowledge discovery. Corresponding author of this paper.)

向 阳 同济大学电子与信息工程学院教授. 主要研究方向为数据库、安全数据库、数据挖掘、XML 数据库.
E-mail: shxiangyang@vip.sina.com
(**XIANG Yang** Professor at the School of Electronics and Information, Tongji University. His research interest covers database, secure database, data mining, and XML.)

薛永生 厦门大学信息与技术学院教授. 主要研究方向为数据库、数据挖掘和知识发现.
E-mail: ysxue@mail.xmu.edu.cn
(**XUE Yong-Sheng** Professor at the School of Information Science and Technology, Xiamen University. His research interest covers database, data mining, and knowledge discovery.)



赵 杠 复旦大学信息科学与工程学院博士研究生. 主要研究方向为查询优化、数据挖掘和 XML 数据库.
E-mail: zhaogang@gmail.com
(**ZHAO Gang** Ph. D. candidate at the School of Information Secince and Technology, Xiamen University. His research interest covers query optimization, data mining, and XML.)