计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
43(12):2124~2130, 2006

语义缓存的聚集查询匹配研究

蔡建宇 吴泉源 贾 焰 邹 鹏
(国防科学技术大学计算机学院 长沙 410073)
(jianyucai@163.com)

Research on Aggregate Query Matching in Semantic Cache

Cai Jianyu, Wu Quanyuan, Jia Yan, and Zou Peng
(*School of Computer Science, National University of Defence Technology, Changsha* 410073)

Abstract queries are pervasive in massive database applications, whose execution tends to be time consuming and costly. Therefore promotion of their efficiency will largely improve the performance of the system. Semantic cache is a novel scheme for aiding query evaluation that reuses the results of previously answered queries. But little work has been done on semantic cache involving aggregate queries. This is a limiting factor in its applicability and it is mostly used in small scale database applications. In order to utilize semantic cache in massive database applications, it is necessary to extend semantic cache to support aggregate query. In this paper, query matching is identified as a foundation for answering query using semantic caches. First, a formal semantic cache model is proposed, which supports aggregate query and provides the basis for the whole research. Then the condition of query matching is presented and query matching is classified. Next, two algorithms are proposed for aggregate query matching. These two algorithms are applied to a massive database application project. Its result proves the efficiency of the algorithms.

Key words aggregate query; semantic cache; query matching

摘 要 为提高海量数据库系统的查询效率,围绕海量数据库系统中的聚集查询技术,把通常应用于小型数据库查询的语义缓存技术拓展到海量数据库的聚集查询中.首先研究了面向聚集查询的语义缓存形式化描述,在此基础上讨论了利用缓存处理查询的条件并对查询匹配进行了分类,提出并实现了包含匹配判定算法和相交匹配判定算法,最后给出了相应的实验结果.在某大型实际工程中的应用表明上述判定算法是有效的.

关键词 聚集查询;语义缓存;查询匹配

中图法分类号 TP311.13

对海量数据的查询主要是为了分析和统计.分析和统计中大量存在聚集查询.而对海量数据的聚集查询处理往往非常耗时.因此提高聚集查询的处理效率是优化海量数据库系统查询性能的关键.语义缓存机制[1-4]能够重用查询结果,缩小查询处理的数据规模是提高查询数据库效率的有效手段.然而现有的语义缓存研究[1-4]基本以"选择-投影-连接"(select project join, SPJ)类型查询为考察对象,没有考虑包含聚集、分组操作的聚集查询.不支持聚集查询使得语义缓存难以用于优化海量数据库的查询性能.为了解决这一问题,必须扩展语义缓存.语义缓存的查询匹配[5]判断缓存能否为当前查询

收稿日期:2005-05-27;修回日期:2006-05-15
基金项目:国家"八六三"高技术研究发展计划基金项目(2003AA111020,2003AA115210,2003AA115410,2004AA112020)

提供查询结果,是利用语义缓存优化查询的基础.在海量数据库应用环境下,聚集查询匹配是基于语义缓存优化聚集查询首先需要解决的问题.

目前语义缓存的查询匹配研究以 SPJ 查询为主,文献[2]提出了一个基于 Datalog 的逻辑模型,给出了确定缓存能否用于查询的准则.文献[3]以支持 SPJ 查询的形式化的语义缓存模型为基础给出了查询匹配的条件和查询匹配的算法,不支持连接、聚集查询.文献[4]引入了近似缓存描述的方法提高 SPJ 查询匹配的速度.由于聚集和分组操作的引入导致 SPJ 查询匹配研究不能直接用于聚集查询匹配.而且已有的语义缓存研究多用于小型数据库应用.文献[6-9]研究了查询与视图的匹配,但是他们的方法在利用语义特征、可操作性方面仍然存在不足.

与以往工作相比,本文的主要贡献在于以海量数据库应用为背景,基于语义缓存的研究现状,通过扩展已有的工作,对语义缓存中的聚集查询匹配问题进行了深入研究,首先给出了面向聚集查询的语义缓存形式化描述,然后提出了适合语义缓存应用的聚集查询匹配算法.

1 语义缓存的形式化描述

本文研究的查询为聚集查询,其关系代数表达式包括投影、选择、连接、聚集和分组.文献[3]的模型以传统关系代数为基础描述 SPJ 查询,不支持聚集、分组操作.下面通过扩展文献[3],给出一个面向聚集查询的语义缓存形式化描述.不失一般性,本文忽略了$\neq$比较,同时假设关系的属性类型为实数域.现做如下术语约定:

1) 比较谓词的形式为 $X\ op\ c$,X 为属性,$op\in\{\leqslant,\geqslant,<,>,=\}$,$c$ 表示常量.

2) 连接谓词的形式为 $X=Y$,X 与 Y 为连接属性.

3) 简单谓词.比较谓词和连接谓词的合称.

定义 1. 给定数据库 $D=\{R_1,R_2,\cdots,R_m\}$,其中 R_m 表示关系,语义缓存项 S 为五元组$\langle A,F,T,P,C\rangle$,其中 $T=\{R_{ik}\mid R_{ik}\in D,k=1,2,\cdots,n\}$,$A=\{a_l\mid a_l$ 为 R_k 的属性,$R_k\in D\}$,$F=\{f(b)\mid b$ 为 R_i 的属性,$R_i\in T,f\in Ag\}$,$Ag=\{\mathrm{MAX},\mathrm{MIN},\mathrm{SUM},\mathrm{COUNT}\}$.$P=p_1\wedge p_2\wedge\cdots\wedge p_j$,$p_j$ 为简单谓词.C 为该缓存项对应查询的结果,其定义为 $C=\pi_{A,F}(\sigma_P(R_{i1}\times R_{i2}\times\cdots\times R_{in}))$,具体含义是首先取得 T 中关系的笛卡儿乘积,然后利用条件 P 进行连接和选择,最后对 A 的属性进行分组运算得到 A 的属性和 F 的聚集函数组成的值.AVG 可转换为对聚集函数 SUM 和 COUNT 的计算,因此这里没有考虑 AVG.

语义缓存为一组缓存项的集合,每个语义缓存项都是聚集查询的结果.为了比较语义缓存项与查询,可按照语义缓存项的定义方法描述查询.

定义 2. 查询 Q 定义为$\langle A_Q,F_Q,T_Q,P_Q,C_Q\rangle$,其中每个组成部分的意义与定义 1 相同.

查询 Q 与语义缓存项 S 的区别在于 C_Q 是虚拟的查询结果,而 C 是缓存的实际查询结果.定义 1 和 2 为查询与语义缓存项提供了统一的描述方法.

2 聚集查询匹配

本节基于前面给出的语义缓存形式化描述解决聚集查询匹配的问题,首先讨论聚集查询匹配的相关概念和判断方法,然后给出两个聚集查询匹配的算法.

2.1 查询匹配

语义缓存与查询的匹配就是从缓存中找到可以优化查询的语义缓存项.本节基于前面给出的语义缓存形式化描述讨论查询匹配的相关概念和判定方法.

定义 3. 对于语义缓存项 $S=\langle A,F,T,P,C\rangle$ 和查询 $Q=\langle A_Q,F_Q,T_Q,P_Q,C_Q\rangle$,如果存在关系代数表达式 E,其中包含投影、选择、聚集和分组运算,$E(C)$ 是 E 的关系运算作用于 C 的结果,$E(C)\neq\varnothing$,$E(C)\subseteq C_Q$,则 S 与 Q 匹配.如果 $E(C)=C_Q$,那么 Q 完全由 S 包含.

分组运算限定只有聚集查询的分组属性集合包含于语义缓存项的分组属性集合,二者才有可能匹配.在此条件下,查询与语义缓存项匹配时它们的聚集函数之间存在多种可能的关系.二者的聚集函数集合完全等价是最直观的一种情况.此外,查询的聚集函数还可通过计算语义缓存项的聚集函数和分组属性取得结果.下面由定义 4 描述这些关系.

定义 4. 当 $f_1(a)\in F_Q$,存在 $f_2(b)\in F$,使得以下 4 个条件之一成立:

1) $f_1=f_2$,$a=b$,$f_1\neq\mathrm{COUNT}$;

2) $f_1=f_2$,$f_1=\mathrm{COUNT}$;

3) $f_1=\mathrm{SUM}$,$f_2=\mathrm{COUNT}$,$a\neq b$,$a\in A$;

4) $f_1=\mathrm{MAX}\mid\mathrm{MIN}$,$a\in A$;

则称查询 Q 的聚集函数 $f_1(a)$ 可由 S 的 F 和

A 导出,记为 $f_1(a)=h(F,A)$. 如果存在 $f_1(a)\in F_Q$,满足 $f_1(a)=h(F,A)$,则称查询 Q 的 F_Q 可由 S 的 F 和 A 导出,用 $F_Q\leftarrow D(F,A)$ 表示. 对任意 $f_1(a)\in F_Q$,满足 $f_1(a)=h(F,A)$,则称查询 Q 的 F_Q 可由 S 的 F 和 A 完全导出,用 $F_Q=D(F,A)$ 表示. 如果 $F_Q\leftarrow D(F,A)$,存在 $f_1(a)\in F_Q$,不满足 $f_1(a)=h(F,A)$,则称查询 Q 的 F_Q 可由 S 的 F 和 A 部分导出,用 $F_Q\approx D(F,A)$ 表示.

基于上述讨论,定理 1 给出判定查询匹配的条件.

定理 1. 给定语义缓存项 $S=\langle A,F,T,P,C\rangle$,查询 $Q=\langle A_Q,F_Q,T_Q,P_Q,C_Q\rangle$,$P_A$ 为出现于查询 Q 的 P_Q 中属性集合. 以下论述成立:①如果 $T=T_Q$,$A_Q\subseteq A$,$F_Q\leftarrow D(F,A)$,$P_A\subseteq A$,$P\wedge P_Q$ 可满足,则 S 与 Q 匹配. ②如果 $T=T_Q$,$A_Q\subseteq A$,$F_Q=D(F,A)$,$P_A\subseteq A$,$P_Q\Rightarrow P$,则 Q 完全由 S 包含.

证明.

1) 假定 Q 与 S 无关匹配,则不存在关系代数表达式 E,只包括投影、选择、分组、聚集,$E(C)\neq\varnothing$,$E(C)\subseteq C_Q$.

构造查询 $Q'=\langle A_Q,F_Q,T_Q,P\wedge P_Q,C'_Q\rangle$,所有满足 $P\wedge P_Q$ 的元组同时也能满足 P_Q,故 $C'_Q\subseteq C_Q$. 又 $P\wedge P_Q$ 可满足,则 $C'_Q\neq\varnothing$.

由 $T=T_Q$,$A_Q\subseteq A$,$F_Q\leftarrow D(F,A)$,$P_A\subseteq A$,$S=\langle A,F,T,P,C\rangle$,可得到 $C'_Q=\pi_{A_Q,F'}(\sigma_{P_Q}(C))$,$F'$ 为根据 $F_Q\leftarrow D(F,A)$ 得到的聚集函数集合. 故存在 $E=\pi_{A_Q,F'}\sigma_{P_Q}$,使得 $E(C)=C'_Q\subseteq C_Q$,$E(C)\neq\varnothing$.

这与前面的假设矛盾. 故 S 与 Q 匹配.

2) 假定 Q 不完全包含于 S,即存在元组 $t_Q\in C_Q$,对于 t_Q,不存在 $t\in S$,使得 $t_Q=E(t)$,E 为关系代数表达式.

构造语义缓存项 $S'=\langle A,F,T,P_Q,C'\rangle$,因为 $T=T_Q$,$A_Q\subseteq A$,$P_A\subseteq A$,$P_Q\Rightarrow P$,

所以 $C'\subseteq C$,而 $A_Q\subseteq A$,$F_Q=D(F,A)$,则 $C_Q=\pi_{A_Q,F'}(C')$,F' 为根据 $F_Q=D(F,A)$ 得到的聚集函数集合,因此对于 $t_Q\in C_Q$,至少存在元组 $t\in C'$,使得 $t_Q=\pi_{A_Q,F'}(t)$.

因为 $C'\subseteq C$,故 $t\in C$.

这与前面的假定矛盾,因此 Q 完全由 S 包含.

证毕.

当查询 Q 与语义缓存项 S 匹配时,S 不一定能够完全包含 Q. 为了区分这种情况,本文给出如下定义.

定义 5. 如果查询 Q 与缓存项 S 之间满足 $T=T_Q$,$A_Q\subseteq A$,$P_A\subseteq A$ 和以下条件之一:

1) $F_Q=D(F,A)$,$P\wedge P_Q$ 可满足;

2) $F_Q\approx D(F,A)$,$P_Q\Rightarrow P$;

3) $F_Q\approx D(F,A)$,$P\wedge P_Q$ 可满足;

则称查询 Q 与缓存项 S 相交匹配.

2.2 查询匹配判定算法

查询与缓存的匹配是依据二者之间的语义相关性判断语义缓存能否解答查询. 查询匹配包括包含匹配、相交匹配和无关匹配. 如果查询与缓存之间的关系既不是包含匹配又不是相交匹配,那么这种关系为无关匹配. 本文不再另外给出无关匹配的判定算法. 下面分别讨论包含匹配和相交匹配的判断方法.

2.2.1 包含匹配

$P_Q\Rightarrow P$ 是包含匹配判断的重要步骤. 判断 $P_Q\Rightarrow P$ 即解决 P_Q 蕴含 P 问题[10]. 下面首先讨论蕴含判断的相关定义和判断算法.

定义 6. 对于所有比较谓词 $X<c_i$,$X\leqslant c_i$ 和 $X=c_i$,$C^X_{\mathrm{up}}=\min(c_i)$,且 C^X_{up} 满足所有连接谓词 $X=Y$. 如果 X 能等于 C^X_{up},则 C^X_{up} 为封闭的,否则 C^X_{up} 为开放的. 对于所有比较谓词 $X>c_i$,$X\geqslant c_i$ 和 $X=c_i$,$C^X_{\mathrm{low}}=\max(c_i)$,且 C^X_{low} 满足所有连接谓词 $X=Y$. 如果 X 能等于 C^X_{low},则 C^X_{low} 为封闭的,否则 C^X_{low} 为开放的.

定理 2. 假定 P 为简单谓词的合取式,P 为可满足当且仅当每个变量 X 都满足 $C^X_{\mathrm{low}}<C^X_{\mathrm{up}}$ 或者 $C^X_{\mathrm{low}}=C^X_{\mathrm{up}}$,且 C^X_{low} 和 C^X_{up} 都是封闭的.

证明.

1) 充分性

构造指派 P_a,其中每个变量 X 都满足 $C^X_{\mathrm{low}}<C^X_{\mathrm{up}}$ 或者 $C^X_{\mathrm{low}}=C^X_{\mathrm{up}}$,且 C^X_{low} 和 C^X_{up} 都是封闭的. 根据 C^X_{low} 和 C^X_{up} 的定义可知 P 中所有简单谓词都能满足. 因此 P_a 为可满足的指派. P 可满足.

2) 必要性

如果 P 可满足,显然对于满足 P 的指派,X 的值满足 $C^X_{\mathrm{low}}<X<C^X_{\mathrm{up}}$ 或者 $X=C^X_{\mathrm{low}}=C^X_{\mathrm{up}}$.

因此,对 P 中每个变量 X 都有 $C^X_{\mathrm{low}}<C^X_{\mathrm{up}}$ 或 $C^X_{\mathrm{low}}=C^X_{\mathrm{up}}$ 且 C^X_{low},C^X_{up} 都为封闭的. 证毕.

定理 3. $P_Q\Rightarrow P$ 当且仅当 P_Q 不可满足,或者满足以下条件:

1) 对于 P 中任意连接谓词 $X=Y$,可在 P_Q 中

找到相同的连接谓词或者由 P_Q 的连接谓词导出 $X=Y$;

2) 对 P 中任意 $X<c, c>C_{\text{up}}^X$ 或者 $c=C_{\text{up}}^X, C_{\text{up}}^X$ 为开放的;

3) 对 P 中任意 $X<c, c<C_{\text{low}}^X$ 或者 $c=C_{\text{low}}^X$, C_{low}^X 为开放的;

4) 对 P 中任意 $X\leqslant c, c\geqslant C_{\text{up}}^X$;

5) 对 P 中任意 $X\geqslant c, c\leqslant C_{\text{low}}^X$;

6) 对 P 中任意 $X=c, c=C_{\text{up}}^X=C_{\text{low}}^X$.

其中 C_{up}^X 和 C_{low}^X 由 P_Q 得到.

证明.

1) 充分性:如果 P_Q 不可满足,则 P_Q 蕴含 P.

当 P_Q 可满足时,根据条件 1)~6)可知,P_Q 蕴含 P 中 $X=Y, X\ op\ c, op\in\{\leqslant,\geqslant,<,>,=\}$.

故 P_Q 蕴含 P.

2) 必要性:如果 P_Q 蕴含 P,那么

对于条件 1),假定对 P 中 $X=Y$,在 P_Q 中没有 $X=Y$ 或者无法导出 $X=Y$.

如果 $X>Y$,设 $X=C^X, Y=C^Y, C^X>C^Y$,这与 P 中任意 $X=Y$ 矛盾. 因此 1)成立.

对于条件 3),假设对于 $X>c$,有 $c\geqslant C_{\text{low}}^X$,设 X 的一个指派 X_1 为 C_{low}^X. 如果 $c>C_{\text{low}}^X$,则存在另外一个 X 的指派 X_2 为 C^X,满足 $c>C^X>C_{\text{low}}^X$. 这两个 X 的指派都不满足 P 中 $X>c$,与条件矛盾,因此 3)成立.

依此类推,可以证明其他条件成立. 证毕.

基于定义 6、定理 2 和 3 可以得到如下 P_Q 蕴含 P 的判断算法 IM:

算法 1. 蕴含判断 IM 算法.

输入:简单谓词合取式 P_Q 和 P;

输出:若 $P_Q\Rightarrow P$ 成立,返回 true,否则返回 false.

① 判断 P_Q 的可满足性

依次处理 P_Q 中比较谓词 $X\ op\ c$,对 P_Q 中每个变量 X 确定其 C_{low}^X 和 C_{up}^X.

依次处理 P_Q 中连接谓词 $X=Y$,使得所有相等的关系属性形成链表集合 L_Q.

按照 L_Q 中链表对相等的属性变量约束取值,对 C_{low}^X 和 C_{up}^X 加以修正.

若每个变量 X 都满足 $C_{\text{low}}^X<C_{\text{up}}^X$ 或者 $C_{\text{low}}^X=C_{\text{up}}^X$,且 C_{low}^X 和 C_{up}^X 都是封闭的,则 P_Q 可满足,执行②,否则,P_Q 不可满足,P_Q 蕴含 P,返回 true.

② 对于 P 中连接谓词 $X=Y$,搜索关系属性链表集合 L_Q,如果不能发现 $X=Y$,则返回 false.

③ 对 P 中比较谓词按照以下条件分别判断:

对 P 中任意 $X<c$,若 $c<C_{\text{up}}^X$ 或者 $c=C_{\text{up}}^X$, C_{up}^X 为封闭的,则返回 false;

对 P 中任意 $X>c$,若 $c>C_{\text{low}}^X$ 或者 $c=C_{\text{low}}^X, C_{\text{low}}^X$ 为封闭的,则返回 false;

对 P 中任意 $X\leqslant c$,若 $c<C_{\text{up}}^X$,则返回 false;

对 P 中任意 $X\geqslant c$,若 $c>C_{\text{low}}^X$,则返回 false;

对 P 中任意 $X=c$,若 $c\neq C_{\text{up}}^X$ 或者 $c\neq C_{\text{low}}^X$,则返回 false.

④ ①~③执行完毕,则 $P_Q\Rightarrow P$,返回 true.

算法 IM 的①按照定理 2 判断 P_Q 是否可满足;②③检查定理 3 的 6 个条件是否成立. 根据定理 3,算法 IM 的①~③判断 $P_Q\Rightarrow P$ 是否成立. 由此,算法 IM 的正确性得到保证.

根据定理 1 和算法 IM 可以给出包含匹配判定算法 AQCM (aggregate query containing match) 如下:

算法 2. AQCM 算法.

输入:语义缓存项 $S=\langle A,F,T,P,C\rangle$ 和查询 $Q=\langle A_Q,F_Q,T_Q,P_Q,C_Q\rangle$;

输出:若缓存项 S 包含查询 Q,则输出 true,否则,输出 false.

① 若 $T=T_Q$,则执行②,否则返回 false.

② $A_Q\subseteq A$,则执行③,否则返回 false.

③ 对于查询 Q 中每个聚集函数 $f_1(a)$,根据 f_1 类型分别处理:

若 f_1 为 MAX 或者 MIN,则搜寻 F 中 $f_2(b)$,使得 $f_1=f_2$,且 $a=b$. 若 f_2 不存在,且 $a\notin A$,则返回 false.

若 f_1 为 COUNT,则搜寻 F 中 $f_2(b)$,使得 $f_2=\text{COUNT}$. 若 f_2 不存在,则返回 false.

若 f_1 为 SUM,则搜寻 F 中 $f_2(b)$,使得 $f_2=\text{SUM}$,且 $a=b$. 若 f_2 不存在,则检查是否满足条件 $a\in A$,存在 $\text{COUNT}(b)\in F$. 若该条件满足,则继续处理 Q 中聚集函数,否则,返回 false.

④ 若 $P_A\subseteq A$,则执行⑤,否则返回 false.

⑤ 调用算法 IM,判断 $P_Q\Rightarrow P$ 是否成立.

算法 AQCM 中③检查 $F_Q=D(F,A)$ 是否成立;①~⑤依据定理 1 判断包含匹配的各项条件. 因此,算法 AQCM 的正确性成立.

2.2.2 相交匹配

定义5中2)可以通过修改算法AQCM加以处理.这里讨论1)和3)情况下的相交匹配.当查询Q与S相交时,S只能提供Q与S相交部分的查询结果.设定$Q_C=\langle A_Q,F_{QC},T_Q,P\wedge P_Q,C_{QC}\rangle$,其中$F_{QC}\subseteq F_Q$,$F_{QC}=D(F,A)$.$Q_C$为$Q$与$S$相交部分查询,可由$S$完全解答.因此需要将查询$Q$划分为缓存查询和数据库查询,这一过程称为查询剪裁[3].当查询包含连接谓词时,查询剪裁就需要对$X\neq Y$形式的谓词进行处理,使得查询更为复杂.为了便于查询剪裁和后端数据库处理查询,限定相交匹配中P_Q和P中包含的连接谓词等价.依据定义5给出相交匹配的判定算法AQOM(aggregate query overlapping match).

算法3. AQOM算法.

输入:语义缓存项$S=\langle A,F,T,P,C\rangle$和查询$Q=\langle A_Q,F_Q,T_Q,P_Q,C_Q\rangle$;

输出:若缓存项S与查询Q相交,则输出true,否则,输出false.

① 若$T=T_Q$则执行②,否则返回false.

② 若$A_Q\subseteq A$则执行③,否则返回false.

③ 对于查询Q中每个聚集函数$f_1(a)$,根据f_1类型分别处理:

若f_1为MAX或者MIN,则搜寻F中$f_2(b)$,使得$f_1=f_2$,且$a=b$.若f_2不存在,且$a\notin A$,则继续处理Q中聚集函数,否则执行⑤.

若f_1为COUNT,则搜寻F中$f_2(b)$,使得$f_2=$COUNT.若f_2存在,则执行⑤,否则继续处理Q中聚集函数.

若f_1为SUM,则搜寻F中$f_2(b)$,使得$f_2=$SUM,且$a=b$.若f_2不存在,则检查是否满足条件$a\in A$,存在COUNT$(b)\in F$.若此条件满足,则执行⑤,否则继续处理Q中聚集函数.

④ 查询Q的所有聚集函数无法由S导出,返回false.

⑤ 若$P_A\subseteq A$,则执行⑥,否则返回false.

⑥ 依次处理P_Q中谓词:

若为比较谓词$X\ op\ c$,对每个变量X确定其C^X_{low}和C^X_{up}.

若为连接谓词$X=Y$,将相等关系属性加入链表中.

⑦ 依次处理P中谓词:

若为比较谓词$X\ op\ c$,对每个变量X确定其C^X_{low}和C^X_{up}.

若为连接谓词$X=Y$,将相等关系属性加入链表中.

⑧ 比较⑥⑦中形成的链表,若二者等价,则执行⑨,否则返回false.

⑨ 利用⑦中链表对相等的属性变量修正C^X_{low}和C^X_{up}.

⑩ 若每个变量X都满足$C^X_{\text{low}}<C^X_{\text{up}}$或者$C^X_{\text{low}}=C^X_{\text{up}}$,且$C^X_{\text{low}}$和$C^X_{\text{up}}$都是封闭的,则$P\wedge P_Q$可满足,返回true,否则返回false.

算法AQOM的③根据定义4检查是否存在$F_Q\leftarrow D(F,A)$;⑥~⑩根据定理2检测$P\wedge P_Q$是否可满足.算法AQOM检验了定义5的相交匹配条件.因此算法AQOM正确性成立.

3 测试分析

StarTP是基于CORBA(common object request broker architecture)[11]开发的并行海量数据库应用平台,主要包括并行查询服务和并行加载服务.在本文提出的匹配算法的基础上,我们在StarTP的并行查询服务中实现了中间层语义缓存.

我们通过3个实验测试验证本文的匹配算法是否有效.实验环境参照StarTP的应用环境搭建,由5台数据库服务器、1台应用服务器和5台普通PC组成.每台服务器配置为两个Intel Xeon 2GHz CPU,4GB RAM,两个70GB的SCSI硬盘.服务器操作系统为Redhat Linux7.3,数据库服务器为Oracle8.1.7.PC机的配置为PⅣ 1.6GHz,256MB内存,操作系统为Win2000 Professional.所有机器通过100Mbps以太网互联.数据库的数据来源于StarTP应用的业务数据.测试查询为实际应用中常用查询.

实验1.测试对比同一数据库在不同数据规模下有缓存和无缓存时的查询响应时间.实际应用中聚集查询中分组属性具有有限取值.在此基础上形成的缓存包括的记录相对于数据库中需要处理的记录要少得多,因此基于缓存处理查询能够减轻查询处理任务.图1是一组聚集查询在有缓存和无缓存情况下的测试结果.图1说明有缓存和无缓存时的聚集查询响应时间有较大差距,基于本文研究实现的语义缓存提高了聚集查询的性能.

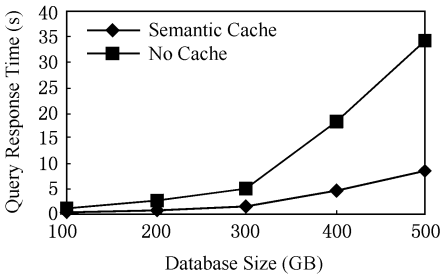

Fig. 1 Impact of semantic cache on aggregate queries.

图 1 语义缓存对聚集查询的影响

实验 2. 以相同的聚集查询集合为输入,基于语义缓存处理聚集查询,考察语义缓存规模增大时查询匹配开销与查询响应时间之间的关系. 从表 1 可以看到,随着语义缓存中语义缓存项目的增多,查询匹配的开销逐渐增大,但是查询匹配的开销占查询响应时间的百分比仍维持在比较低的水平. 海量数据库应用的聚集查询响应时间一般比较长,因此查询匹配的开销与语义缓存带来的性能改善相比可忽略不计. 只要语义缓存保持合理规模,查询匹配的开销将不会对查询响应时间有过多的影响. 实验 2 的结果表明,本文聚集查询匹配算法带来的开销是可接受的.

Table 1 Aggregate Query Matching Cost and Query Response Time with Different Scale of Semantic Cache Items

表 1 不同语义缓存规模下的聚集查询匹配开销和查询响应时间

Number of Semantic Cache Items	Query Response Time (ms)	Query Matching Cost (ms)	Query Matching Cost/Query Response Time (%)
50	187.677	0.87586	0.467
100	188.897	1.751	0.927
150	189.235	2.631	1.389
200	190.077	3.501	1.842
250	192.453	4.368	2.269

实验 3. 以相同的聚集查询集合为输入,基于语义缓存处理聚集查询,考察不同语义缓存规模下包含匹配、相交匹配和无关匹配的分布. 图 2 表明在相同条件下,随着语义缓存规模的增大,聚集查询与缓存项匹配成功的几率也逐渐增大.

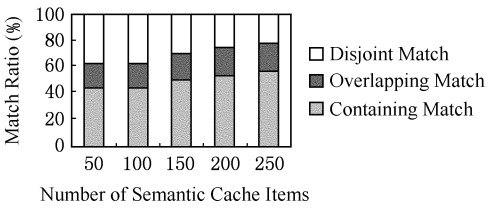

Fig. 2 Ratio of aggregate query matching with different scale of cache.

图 2 不同缓存规模下聚集查询匹配类型的比例

4 结束语

当前的语义缓存研究以 SPJ 查询为重点,没有对聚集查询进行深入研究. 而聚集查询是海量数据库应用中常见的查询类型. 查询匹配是语义缓存优化查询的关键. 为了能够在海量数据库应用环境下利用语义缓存实现查询优化,有必要扩展查询匹配. 本文形式化描述了优化聚集查询所需的语义缓存,讨论了利用缓存处理聚集查询的条件,给出了查询与缓存匹配的定义和判断算法.

本文的算法在海量数据库应用平台 StarTP 中得到了实现,测试结果验证了算法的有效性. 本文的研究已投入某大型实际工程应用. 它的数据规模达到了 TB 级,其查询业务包含大量的聚集查询. 进一步的工作包括更为复杂的聚集查询匹配、基于知识库的聚集查询匹配、语义缓存的一致性维护等.

参 考 文 献

[1] S Dar, M J Franklin, B T Jonsson, *et al*. Semantic data caching and replacement [C]. In: Proc of the 22nd Int'l Conf on Very Large Data Bases. San Fransisco: Morgan Kaufmann, 1996. 330-341

[2] Parke Godfrey, Jarek Gryz. Answering queries by semantic caches [C]. In: Proc of the 10th DEXA. Berlin: Springer-Verlag, 1999. 485-498

[3] Q Ren, M H Dunham, V Kumar. Semantic caching and query processing [J]. IEEE Trans on Knowledge and Data Engineering, 2003, 15(1): 192-210

[4] J Basu. Associative caching in client-server databases: [Ph D dissertation] [D]. San Francisco, California: Stanford University, 1998

[5] D Lee, W W Chu. Semantic caching via query matching for Web sources [C]. In: Proc of the 8th Int'l Conf on Information and Knowledge Management. New York: ACM Press, 1999. 77-85

[6] M Zaharioudakis, R Cochrane, G Lapis, *et al*. Answering complex SQL queries using automatic summary tables [C]. In: Proc of ACM SIGMOD Int'l Conf on Management of Data. New York: ACM Press, 2000. 105-116

[7] A Gupta, V Harinarayan, D Quass. Aggregate-query processing in data warehousing environments [C]. In: Proc of the 21st Int'l Conf on Very Large Data Bases. San Francisco: Morgan Kaufmann, 1995. 358-369

[8] D Srivastava, S Dar, H V Jagadish, *et al*. Answering queries with aggregation using views [C]. In: Proc of the 22nd Int'l Conf on Very Large Data Bases. San Francisco: Morgan Kaufmann, 1996. 318-329

[9] S Cohen, W Nutt, A Serebrenik. Rewriting aggregate queries using views [C]. In: Proc of the 18th Symp on Principles of Database Systems. New York: ACM Press, 1999

[10] X H Sun, N Kamel, L M Ni. Solving implication problems in database applications [C]. In: Proc of the ACM SIGMOD Int'l Conf on Management of Data. New York: ACM Press, 1989. 185-192

[11] Object Management Group. The common object request broker: Architecture and specification, version 3.0.3 [OL]. http://www.omg.org/cgi-bin/doc?formal/04-03-01, 2004-03-01

Cai Jianyu, born in 1976. Ph. D. His current research interests include distributed computing and database.

蔡建宇,1976 年生,博士,主要研究方向为分布计算和数据库.

Wu Quanyuan, born in 1942. Professor and Ph. D. supervisor. His current research interests include intelligent software and distributed computing.

吴泉源,1942 年生,教授,博士生导师,主要研究方向为智能软件和分布计算.

Jia Yan, born in 1962. Professor and Ph. D. supervisor, senior member of China Computer Federation. Her current research interests include distributed computing and database.

贾焰,1962 年生,教授,博士生导师,中国计算机学会高级会员,主要研究方向为分布计算和数据库.

Zou Peng, born in 1957. Professor and Ph. D. supervisor. His current research interests include distributed computing and operating system.

邹鹏,1957 年生,教授,博士生导师,主要研究方向为分布计算和操作系统.

Research Background

With the increase of massive database applications in critical business, how to improve aggregate query performance has become a key problem that needs to be solved urgently. Therefore efficiently executing aggregate queries is very important. Semantic cache is a novel scheme for aiding query evaluation that reuses the results of previously answered queries. But little work has been done on semantic cache involving aggregate queries. This is a limiting factor in its applicability and it is mostly used in small scale database applications. In order to utilize semantic cache in massive database applications, it is necessary to extend semantic cache to support aggregate query. Query matching provides foundation for answering query using semantic caches. Therefore, we research on aggregate querying in this paper. We propose a formal semantic cache model firstly. Based on this model, the conditions of aggregate query matching are discussed. Finally, two algorithms are proposed for aggregate query matching and applied to a massive database application project. Our work is supported by the National High-Tech Research and Development Plan of China under grant Nos 2003AA111020, 2003AA115210, 2003AA115410, and 2004AA112020.