

一种应用于 DaaS 的物化视图候选集生成算法^{*}

张水平, 程 超, 王 蓉, 张凤琴, 陈爱网

(空军工程大学 电讯工程学院 指挥自动化工程系, 西安 710077)

摘 要: 针对 DaaS 数据中心建设中物化视图选择对候选视图集的新要求和传统 MVPP 方法的不足, 提出一种新的候选视图集生成算法。该算法利用多操作变换规则进行查询优化, 然后利用算法 1 和 2 进行关系融合。实验证明, 该算法能够提高查询效率, 压缩候选视图集, 具有较高的可扩展性, 符合应用需求。

关键词: 数据即服务; 面向服务体系架构; 物化视图; 候选视图集

中图分类号: TP312; TP301.6

文献标志码: A

文章编号: 1001-3695(2012)11-4137-04

doi:10.3969/j.issn.1001-3695.2012.011.035

Algorithm to constructing view sets used for materialized in DaaS

ZHANG Shui-ping, CHENG Chao, WANG Rong, ZHANG Feng-qin, CHEN Ai-wang

(Dept. of C³I, Institute of Telecommunication Engineering, Air Force Engineering University, Xi'an 710077, China)

Abstract: For the new requirements of constructing view sets used for materialized in DaaS data center construction and shortcomings of the traditional MVPP method, this paper proposed a new algorithm for materializing view set generation. First, it used transformation rules to optimize the query, then used algorithm 1 and 2 to fuse all relationship. Experiment proves that the algorithm can improve the query efficiency, it compresses candidates view set and has high scalability. And it is accord with application requirements.

Key words: data as a service (DaaS); service oriented architecture; materialized view; candidate view set

0 引言

随着云计算技术的迅猛发展和企业信息化管理的迫切需求, 资源共享的广度不断加大, 云计算的服务类型逐渐丰富。云计算是一种商业模型, 它将计算任务分布在大量计算机构成的资源池上, 使用户能够按需获取计算力、存储空间和信息资源^[1], 简言之就是实现任务分配和按需获取。它提供的服务模式主要包括: 基础设施即服务 (infrastructure as a service, IaaS)、平台即服务 (platform as a service, PaaS) 和软件即服务 (software as a service, SaaS) 三种。其中任何一种都可以独立对外提供服务。随着 Nagy 首次提出将数据作为服务, 一种新的服务模式——数据即服务 (DaaS) 已经被许多国际组织所认可^[2]。信息系统是一个信息密集型的系统, 数据是其关注的重点, 因此 DaaS 对企业信息化管理具有重大意义。

DaaS 的目的是实现数据按需获取, 基础是数据中心的建立。视图作为数据中心的主体, 包括物理视图和虚拟视图。物理视图是将异构数据库中的数据抽取出来经过转换、存储到中心数据库, 它是一张实实在在的表; 虚拟视图是抽取数据源的模式信息到中心数据库, 通过模式操作实际的数据, 数据仍旧保留在各自数据源中。它们的优缺点主要体现在存储代价、查询代价和维护代价的不同。物理视图在本地创建副本, 数据的访问在本地执行, 查询效率高, 但由于数据存在副本, 所以必须

考虑数据的一致性问题, 进而导致视图维护代价的产生。虚拟视图只有一份数据, 不存在数据更新和数据一致性问题, 但由于实际数据存储在远端, 所以在数据查询过程中会产生一定的查询代价。由于用户对不同数据的实时性、一致性要求不同和存储空间的限制, 数据是否以物理视图的形式存储不可一概而论, 物化视图选择策略研究就是基于这一应用背景, 兼顾数据存储代价、查询代价和维护代价, 为数据中心的建立提供决策辅助。物化视图选择包括两个问题: a) 候选视图集生成问题, 也就是如何得到所有可能被物化的视图的集合; b) 物化视图选择算法。物化视图候选集生成算法是针对前者提出的。

1 需求分析和相关研究

1.1 DaaS 概述

DaaS 是 SaaS 的孪生兄弟^[3], 作为“as a service”家族成员之一, 它将数据作为一种商品提供给任何有需求的组织或个人^[4]。SOA (service oriented architecture, 面向服务的体系架构) 是一种业务驱动的、粗粒度、松耦合的服务架构, 支持对业务进行整合, 使其成为一种相互联系、可重用的业务任务或服务, 是实现 DaaS 最有效的方法。基于 SOA 的 DaaS 体系架构如图 1 所示。基础异构数据资源经过数据整合后生成符合公共语言模式的视图, 最后利用 Web service 技术将视图封装成具有公共接口的服务供用户调用, 从而实现数据资源的按需

收稿日期: 2012-04-09; 修回日期: 2012-05-16 基金项目: 陕西省自然科学基金资助项目 (2011JM8035)

作者简介: 张水平 (1956-), 女, 山西新泽人, 教授, 硕士, CCF 会员, 主要研究方向为分布式数据库、信息安全、云计算 (ech_kdy@sina.com); 程超 (1986-), 男, 湖北广水人, 硕士研究生, 主要研究方向为分布式数据库、云计算; 王蓉 (1978-), 女, 陕西西安人, 讲师, 主要研究方向为分布式数据库、信息安全; 张凤琴 (1964-), 女, 山西芮城人, 副教授, 硕士, 主要研究方向为分布式数据库、信息安全; 陈爱网 (1975-), 男, 江苏苏州人, 副教授, 主要研究方向为分布式数据库、物联网。

获取。

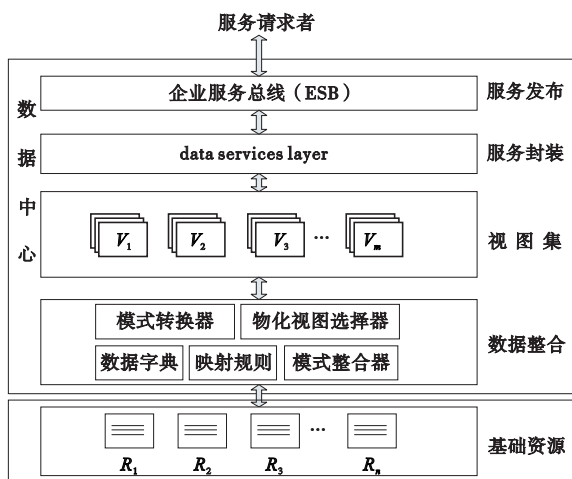


图1 基于SOA的DaaS体系架构

1.2 需求分析

本文依托陕西省自然科学基金项目,主要研究私有云中数据信息的存储和调用问题,这也是当前许多大型企业、政府机关信息化改造关注的重点。私有云中,视图集的建立主要针对常用的查询集,基于SOA架构的DaaS要求存储在数据中心的视图集应当具有以下特点:

- 为减少视图间数据的传输代价、降低物化视图选择问题的复杂度,候选视图的数量应尽可能少。
- 为满足用户不断增加的应用需求,候选视图集必须具有高可扩展性,且新增查询时,新视图集生成的时间应尽可能短。
- 视图集发生更新时,不能影响以往用户对数据的使用。

1.3 物化视图候选集生成相关研究

目前对物化视图选择问题的研究大多是在假定候选视图集已得的情况下进行的,真正关于如何生成候选视图集的研究很少。文献[5]提出将MVPP(multi-view processing plan)作为候选视图集被广泛应用,其核心思想是针对查询集遍历所有可能的查询方案,得到总开销最小的MVPP。这种方法虽然能够得到全局最优MVPP,但随着查询集数量的增加,MVPP的个数将呈指数增长。查询集发生变化时,视图集必须重新生成,必然会对以往用户对数据服务的调用产生影响。针对MVPP方法的不足和DaaS的需求,本文提出了一种新的物化视图候选集生成算法——DMVPP算法。

2 DMVPP(DaaS-MVPP)算法

2.1 相关定义

定义1 关系运算。在关系操作中,以集合代数为基础运算的数据操作语言(DML)称为关系代数语言,相应的运算就称为关系代数运算,简称关系运算。常用的基础关系运算主要包括选择(σ)、投影(π)、并($\cup$)、差($-$)和笛卡尔积($\times$)。前两类属于一元运算;后三类属于二元运算。

定义2 关系代数。由关系和其上的关系运算组成的代数称为关系代数。关系代数本身也表示一个关系。特殊地,若关系上的关系运算只包含一元运算,则称这个关系代数为一元关系代数,原关系称为一元关系代数的基,每个一元运算称为

一元关系代数的因子。

定义3 查询树^[6]。查询树是一棵树 $T=(V,E)$,其中:

- V 是节点集,每个非叶节点是关系运算,叶节点是关系名(即查询涉及的关系);
- E 是边集,两节点有边 (V_1,V_2) ,当且仅当 V_2 是 V_1 的操作分量。

特殊地,如果查询树的所有根节点都是一元关系运算组,所有叶节点都是一元关系代数,其他所有节点都是二元关系运算,则称这种查询树为标准查询树。

定义4 偏序关系。对于两个关系 E_1 和 E_2 ,如果被 E_1 所响应的所有查询都能被 E_2 所响应,反之不成立,则 E_1 和 E_2 存在偏序关系, E_1 是 E_2 的偏序,表示为 $E_1 \leq E_2$ 。

定义5 关系融合。如果两个关系代数可以转换成两个对同一个关系代数的一元操作的形式,则称这两个关系代数可融合,将它们融合的过程称为关系融合,这个共同的关系代数称为原关系代数的最大公因式。

2.2 算法理论基础

关系代数和查询树是查询操作的两种等价表示,在对查询进行优化时,可以运用以下变换规则和推论对关系代数进行等价变换。

2.2.1 多操作变换规则

设有多个关系 R,S,T ,用 U 表示一元操作, B 表示二元操作,在一定条件下存在如下规则^[7]:

- 一元操作交换律: $U_1(U_2(R)) \equiv U_2(U_1(R))$ 。
- 二元操作结合律: $(R)B_1((S)B_2(T)) \equiv ((R)B_1(S))B_2(T)$ 。
- 二元操作交换律: $(R)B(S) \equiv (S)B(R)$ 。
- 一元操作对二元操作的分配律: $U((R)B(S)) \equiv U(R)BU(S)$ 。
- 一元操作的因式分解律: $U(R)BU(S) \equiv U((R)B(S))$ 。
- 联接与笛卡尔积的等价变换: $R_1 \bowtie_{ij} R_2 \equiv \sigma_{ij}(R_1 \times R_2)$ 。

2.2.2 相关推论

根据上述等价变换规则,结合2.1节中的定义,可以得出如下推论:

推论1 两个关系 E_1 和 E_2 , $E_1 \leq E_2$ 的充要条件是存在一元操作组 F ,使得 $E_1 = F(E_2)$ 。

证明 充分性。设 $E_1 = F(E_2)$, F 为一元操作组, Q 为一个对 E_1 的一元操作组,则由一元操作交换律有

$$Q(E_1) = Q(F(E_2)) = F(Q(E_2)) \subset Q(E_2)$$

即 $Q(E_1) \subset Q(E_2) \Rightarrow E_1 \leq E_2$ 。

必要性。若 $E_1 \leq E_2$,则对于任一对 E_1 的一元操作组 Q' 有

$$Q'(E_1) \subset Q'(E_2)$$

则必然存在一元操作组 F' ,使得 $Q'(E_1) = F'(Q'(E_2))$,由一元操作交换律有

$$Q'(E_1) = Q'(F'(E_2)) \Rightarrow E_1 = F'(E_2)$$

证毕。

推论2 关系 E_1 和 E_2 , X 表示 E_1 的属性集, Y 表示 E_2 的属性集,若 $X \subseteq Y$ 且 $\pi_X(E_2) \supseteq E_1$,则 $E_1 \leq E_2$ 。

证明 设 Q 为一个对 E_1 的一元操作组,因为 $X \subseteq Y$,且 $\pi_X(E_2) \supseteq E_1$,所以 $Q(E_1) \subseteq Q(\pi_X(E_2)) \Rightarrow E_1 \leq \pi_X(E_2)$ 。

由推论 1 知: $\pi_X(E_2) \leq E_2 \Rightarrow E_1 \leq E_2$ 。

推论 3 选择操作对笛卡尔积的分配律。两个关系代数 E_1 和 E_2 , σ_F 是对 E_1, E_2 的笛卡尔积上的选择操作, 则对于关系代数 $\sigma_F(E_1 \times E_2)$:

当 F 只涉及 E_1 属性, 则 $\sigma_F(E_1 \times E_2) = \sigma_F(E_1) \times E_2$ 。

当 $F = F_1 \wedge F_2$, F_1 涉及 E_1 属性, F_2 涉及 E_2 属性, 则 $\sigma_F(E_1 \times E_2) = \sigma_{F_1}(E_1) \times \sigma_{F_2}(E_2)$ 。

推论 4 投影操作对笛卡尔积的分配律。两个关系代数 E_1 和 E_2 , $A_p (p=1, 2, \dots, n)$ 是 E_1 的属性, $B_q (q=1, 2, \dots, m)$ 是 E_2 的属性, i 是 E_1 中的域, j 是 E_2 中的域, 则

$$\pi_{A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m}(E_1 \times E_2) = \pi_{A_1, A_2, \dots, A_n, i} E_1 \times \pi_{B_1, B_2, \dots, B_m, j} E_2$$

推论 5 两个非叶子节点 E_1, E_2 , 当 E_1 的儿子节点 R_1, R_2 分别与 E_2 的儿子节点 R'_1, R'_2 可融合, 即 R_1 与 R'_1 可融合, 且 R_2 与 R'_2 可融合, 则 E_1 与 E_2 可融合。

证明 设 $E_1 = R_1 \bowtie_{ij} R_2, E_2 = R'_1 \bowtie_{i'j'} R'_2, R_1, R_2$ 为 E_1 的儿子节点, R'_1, R'_2 为 E_2 的儿子节点。

因为 R_1 与 R'_1 可融合, 且 R_2 与 R'_2 可融合, 所以存在关系代数 M, N , 使得 $R_1 = F(M), R_2 = H(N), R'_1 = F'(M), R'_2 = H'(N)$ 。其中: M 为 R_1 和 R'_1 的最大公因式; N 为 R_2 和 R'_2 的最大公因式; F, H, F', H' 为一元运算符。

由多操作变换规则 f) 可得

$$E_1 = R_1 \bowtie_{ij} R_2 = \sigma_{ij}(R_1 \times R_2) = \sigma_{ij}(F(M) \times H(N))$$

$$E_2 = R'_1 \bowtie_{i'j'} R'_2 = \sigma_{i'j'}(R'_1 \times R'_2) = \sigma_{i'j'}(F'(M) \times H'(N))$$

由推论 3、4 可得

$$E_1 = \sigma_{ij}(F(M) \times H(N)) = \sigma_{ij}FH(M \times N)$$

$$E_2 = \sigma_{i'j'}(F'(M) \times H'(N)) = \sigma_{i'j'}F'H'(M \times N)$$

得证 E_1 与 E_2 可融合, 最大公因式为 $M \times N$ 。

2.3 算法描述

DMVPP 算法的核心思想是优化查询、视图融合。具体步骤如下:

- 建立查询树。将全局查询表达式转换成查询树。
- 查询树标准化。对于查询集中的单个查询操作, 用一元操作对二元操作的分配律将一元操作向下移动, 以缩减二元操作数关系。

c) 关系融合。关系融合的目的是融合存在偏序关系的视图, 压缩候选视图集, 具体分为叶子节点关系融合和非叶子节点关系融合两步。

(a) 叶子节点关系融合。运用算法 1 对基相同的叶子节点进行关系融合。

算法 1 一元关系代数关系融合算法

输入: 两个一元关系代数 E_1, E_2 。

输出: 两个一元关系代数的最大公因式 M , 两个与原一元关系等价的以最大公因式为基的一元关系代数 E'_1, E'_2 。

设 R_1, R_2 分别为 E_1, E_2 的基, $A = \{A_i | i=1, 2, \dots, n\}, B = \{B_j | j=1, 2, \dots, m\}$ 分别为 E_1, E_2 的因子的集合, $C = \emptyset$ 。

- 若 $R_1 = R_2$, 进入步骤 2, 否则结束;
- for each A_i , 根据推论 2, if $E_2 \leq A_i(R_1), C = C \cup \{A_i\}, A = A - \{A_i\}$; end for;
- for each B_j , 根据推论 2, if $E_1 \leq B_j(R_2), C = C \cup \{B_j\}, B = B -$

$\{B_j\}$; end for;

4. 输出 $M = C_1 C_2 \dots C_p(R_1), (C_1, C_2, \dots, C_p \in C),$

$E'_1 = A_1 A_2 \dots A_i(M), (A_1, A_2, \dots, A_i \in A),$

$E'_2 = B_1 B_2 \dots B_j(M), (B_1, B_2, \dots, B_j \in B)。$

(b) 非叶子节点关系融合。运用算法 2 对以二元操作为根的查询子树根节点进行融合。

算法 2 以二元操作为根的查询子树根节点融合算法

输入: 两个以二元操作为根的查询树的等价关系代数 E_1, E_2 。

输出: E_1 和 E_2 的最大公因式, E_1 和 E_2 以最大公因式为基的关系代数。

1. E_1 的两个儿子节点是否与 E_2 的两个儿子节点分别可融合, 是则进入步骤 2, 否则结束;

2. 根据推论 5 证明的结果得到 E_1 和 E_2 的最大公因式, E_1 和 E_2 以最大公因式为基的关系代数。

2.4 应用举例

假设基础资源中有四张基本表:

学生信息表: Student(S_id, S_name, S_sex, S_home, C_id);

班级信息表: Class(C_id, C_name, C_sponsor);

成绩信息表: Grade(S_id, Co_id, G_grade, G_fullmark);

课程信息表: Course(Co_id, Co_name, Co_teacher)。

基于四张基本表的三个查询操作:

Q_1 : SELECT S_id, S_name, C_name

FROM Student, Class

WHERE Student.C_id = Class.C_id;

Q_2 : SELECT S_id, S_name, Co_name, G_grade

FROM Student as A, Grade as B, Course as C

WHERE A.S_id = B.S_id and B.Co_id = C.Co_id and Co_name =

分布式数据库';

Q_3 : SELECT S_id, S_name, C_name, G_grade, Co_name

FROM Student as A, Class as B, Grade as C, Course as D

WHERE A.C_id = B.C_id and A.S_id = C.S_id and C.Co_id = D.Co_id and G_grade ≥ 80 。

按照算法的执行步骤如下:

a) 建立查询树, 如图 2 所示。

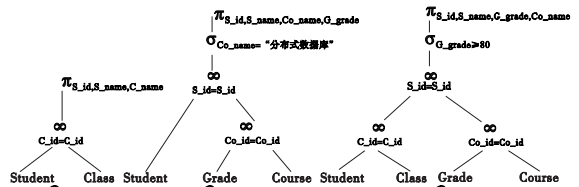


图2 查询树

b) 标准化, 如图 3 所示。

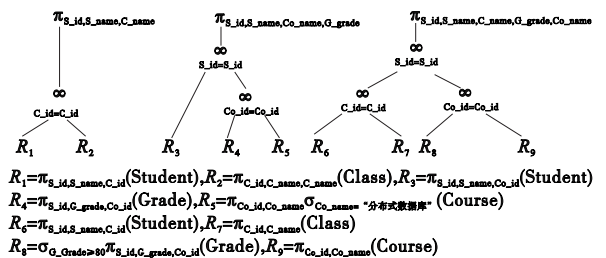


图3 标准化查询树

c) 叶子节点关系融合, 如图 4 所示。

d) 非叶子节点关系融合, 如图 5 所示。

3 算法分析

DMVPP 算法在优化查询的基础上采用算法 1 和 2 对可融合的关系进行关系融合,在确保查询较优的基础上有效缩减了候选视图集的个数和视图集生成时间。此算法的最大优点是查询集具有高可扩展性,当查询集有新增查询操作时,MVPP 方法需重新生成候选视图集,时间复杂度为指数阶 $O(2^n)$;DMVPP 算法候选视图集只进行增量更新,不会影响以往用户对数据服务的使用,时间复杂度为常数阶 $O(1)$ 。

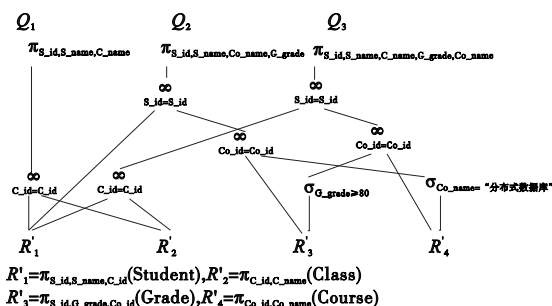


图4 叶子节点关系融合

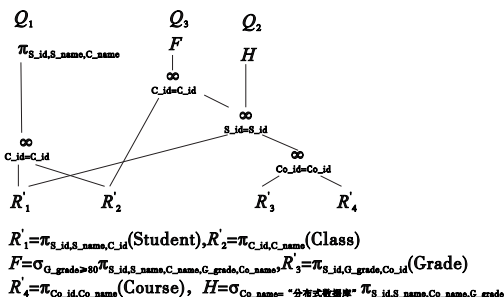


图5 非叶子节点关系融合

4 实验对比

4.1 实验设计

在实验室环境下对 MVPP 和 DMVPP 的性能进行了详尽的对比。实验网络结构如图 6 所示,节点 1 和 2 分别为支撑两个不同在线系统运行的服务器。其中节点 1 运用 Microsoft SQL Server 2000 数据库平台,包含 56 个基本表,主要反映设备的基本属性信息和生命周期;节点 2 运用 Oracle 9i 数据库平台,包含 137 个基本表,主要记录设备的状态信息和日常维护信息;视图集存储在数据中心,采用 Oracle 10g 数据库平台。

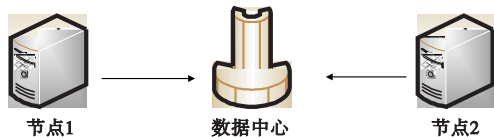


图6 实验网络结构

测试环境:

1) 数据中心

硬件平台: Xeon E5620 2.4 GHz, 8 GB DDR3; 操作系统: Windows 2003 Server; 数据库平台: Oracle 10g。

2) 节点 1

硬件平台: P4 3.0 GHz, 1 GB DDR2; 操作系统: Windows XP SP3; 数据库平台: Microsoft SQL Server 2000。

3) 节点 2

硬件平台: P4 3.0 GHz, 1 GB DDR2; 操作系统: Windows

XP SP3; 数据库平台: Oracle 9i。

4.2 实验结果对比

实验主要包括三项内容: a) 同一查询集, 候选视图集大小对比; b) 新增查询, 候选视图集生成时间对比; c) 新增查询, 原候选视图集变化情况。

实验过程主要采用人工统计, 取查询集大小分别为 10、20、40、50, 实验结果分别如图 7~9 所示。

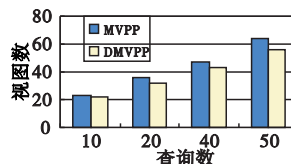


图7 候选视图集大小对比

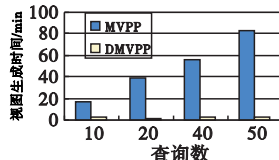


图8 新增1条查询候选视图集生成时间对比

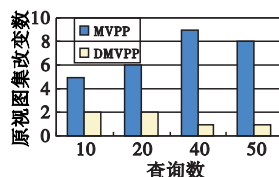


图9 新增1条查询原候选视图集变化情况对比

从实验结果可以看出, 使用 DMVPP 算法, 无论候选视图集大小还是生成时间都优于 MVPP 算法, 特别是当查询集发生变化时, 新视图集的生成时间远远低于 MVPP 算法。

5 结束语

本文提出的 DMVPP 借助多操作变换规则对查询进行优化, 按照关系融合算法对可融合关系进行融合, 有效减少候选视图个数。该算法生成的候选视图集具有很好的可扩展性, 当操作集发生变化时, 能够在确保不影响以往用户调用数据服务的基础上快速对候选视图集进行增量更新。实验证明, DMVPP 算法有效、可行。

由于本文运用的变换规则和融合算法都只是局限于 SPJ 等简单操作, 下一步可以在综合考虑主键、外键和索引的基础上对算法进行改进。

参考文献:

- [1] 刘鹏. 云计算[M]. 北京: 电子工业出版社, 2010: 1-2.
- [2] NAGY Z. Seminar on emerging trends in data communication and dissemination[EB/OL]. (2010-06-09). http://unstats.un.org/unsd/statcom/statcom_2010/Seminars/Communication/Data%20as%20a%20Service.ppt.
- [3] MACHAN D. DaaS: the new information goldmine[EB/OL]. (2010-06-09). <http://online.wsj.com/article/SB125071202052143965.html>.
- [4] Data as a service; are we in the clouds? [J]. Journal of Map & Geography Libraries, 2010, 6(1): 76-78.
- [5] YANG J, KARLAPEL K, LI Q. Algorithms for materialized view design in data warehousing environment[C]//Proc of the 23rd International Conference of Very Large Data Bases. 1997: 136-145.
- [6] 郑振楣, 于戈, 郭敏. 分布式数据库[M]. 北京: 科学出版社, 1998: 57-58.
- [7] 邵佩英. 分布式数据库系统及其应用[M]. 2版. 北京: 科学出版社, 2005: 75-76.