

多维数据概念模型的形式化描述^{*}

陆昌辉 刘青宝 邓 苏 张维明

(国防科学技术大学 信息系统与管理学院 长沙 410073)

摘 要 本文基于维层次间的部分包含关系及其传递性,对多维数据概念模型的形式化描述进行了初步研究,并首次从静态结构视图与功能结构视图两个方面对其进行了详细的阐述;同时还从基本代数操作、模型分析操作、模型维护操作三个方面对该模型支持的操作进行了定义;最后还给出了该模型的构建方法。

关键词 移动电子服务,数据仓库,数据建模,多维数据模型

The Formalization of Conceptual Multidimensional Data Model

LU Chang-Hui LUI Qing-Bao DENG Su ZHANG Wei-Ming

(College of Information System and Management, National University of Defense Technology, Changsha 410073)

Abstract This paper introduces a kind of formalization of multidimensional data model based on the partial containment relation between the dimension levels and its transitivity. It first describes the conceptual multidimensional data model from the static structure view and function structure view, and gives the definition of its basic algebra operations, model analysis operations and model maintenance operations. Finally, it also gives this model's building techniques.

Keywords Mobile electronic services, Data warehouse, Multidimensional modeling, Multidimensional data model

1 引言

多维数据建模技术是数据仓库、多维数据库、OLAP 应用的基础^[1]。随着管理信息系统的广泛应用,各大公司都积累了大量的历史操作数据,怎么利用这些历史数据来为他们的决策活动服务已成为当前的研究重点,多维数据建模技术是其关键技术之一。在多维数据模型中,维是一个非常重要的概念,由于其具有一定的层次结构,它允许人们从不同的粒度对所关心的事实进行分析,从而为各级管理人员提供辅助决策信息^[2]。在已有的多维数据模型中,在定义维层次之间的包含关系时,绝大部分研究者均默认为是完全包含关系^[5~11],然而,在空间数据中,维层次间普遍存在一种部分包含关系^[3,4],很少有人考虑这样一种部分包含关系^[4]。本文从维层次间的部分包含关系及其传递性出发,对多维数据概念模型的形式化描述进行了初步研究。

2 模型定义

2.1 静态结构视图

定义 1 (维模式) 维模式 D_{schema} 可以定义成这样一个四元组 $(T_D, \pi_D, \perp_D, ALL)$, 其中 $T_D = \{T_b, i=1, \dots, m\}$ 为其维层次类型集合; π_D 是定义在 T_D 上的一个偏序关系; ALL 是该维模式的顶层维层次,它具有最高抽象级别,含有唯一的元素 all ; $\perp_D$ 为该维的底层维层次,其粒度最细,在一个特定的维模式中, $\perp_D$ 是唯一的。

定义 2 (维) 维 D 可以定义成这样一个二元组 (C_D, P) , 其中 $C_D = \{C_b, i=1, \dots, m\}$, C_b 为该维中属于同一层次类型的维层次构成的集合,对于每一个维层次 $e_i \in C_b$, 其

相应的层次类型 T_b 可以用函数 $Type: U_i C_b \rightarrow U_i T_b$ 来表示, $Type(e_i) = T_b$; 关系 $P \subseteq Dim \times Dim \times (0, 1]$, 其中 $Dim = U_i C_b, \{i=1, \dots, m\}$ 。

对于 $(e_i, e_j, p) \in P, \{e_i \in C_b, e_j \in C_b \text{ 且 } Type(e_i) \neq Type(e_j)\}$, 则称维 D 的层次 e_i 包含在层次 e_j 中, 其中 p 为其包含度, 取值范围为 $(0, 1]$ 。为了描述上的方便, 我们把 $(e_i, e_j, p) \in P$ 简化为 $e_i \pi_p e_j$, 其具体含义如下:

(1) 当 $p=1$ 时, $e_i \pi_p e_j$ 表示维层次 e_i 完全包含在层次 e_j 中;

(2) 当 $0 < p < 1$ 时, $e_i \pi_p e_j$ 表示维层次 e_i 部分包含在层次 e_j 中, 其包含度为 p ;

另外, 我们还在维 D 上基于关系 P 定义了如下两个函数:

$Anc: Dim \rightarrow 2^{Dim}$ 其作用就是返回维层次 e_i 的直接父类层次集, 维层次 all 的直接父类层次集为空。

$Desc: Dim \rightarrow 2^{Dim}$, 其作用就是返回维层次 e_i 的直接孩子层次集, 层次类型 $\perp_D$ 为的维层次的孩子层次集为空。

定义 3 (维事实模式) 一个 n 维事实模式 S 是一个二元组 (F_{schema}, D_{schema}) , 其中 F_{schema} 为事实模式, 它表示所分析的主题, 可以用属性集 $F_{schema} = \{m_1, \dots, m_k\}$ 来表示; $D_{schema} = \{D_{schema}^i, i=1, \dots, n\}$ 是与该事实模式相连的维模式集合, 其中 n 为维的数目, 且 $D_{schema}^1 \times \dots \times D_{schema}^n \rightarrow F_{schema}$ 。

定义 4 (事实) 事实 F 的定义如下: $F = \{f \mid f = (id, m_1, \dots, m_k, e_1, \dots, e_n)\}$, 其中 id 为该事实的唯一标识, $m_i (i=1, \dots, k)$ 为事实属性, $e_j (j=1, \dots, n)$ 为相应的维值, 且 $e_j \in C_{D_j}$ 。

定义 5 (p -characterization) 为了简化起见, 我们把

^{*})Supported by the National Natural Science Foundation of China under Grant No. 60172012(国家自然科学基金)。陆昌辉 博士研究生, 主要研究领域为数据仓库技术, 数据挖掘技术。邓 苏 博士、教授, 主要研究领域为系统仿真技术和数据仓库技术。张维明 博士、教授、博士生导师, 主要研究领域为信息与决策技术, 软件工程技术。

p -characterization 记为 $f \sim_p e_k$, 其中 p 为描述度, 取值范围为 $(0, 1]$ 。其定义如下: 给定一事实 F 及维 $D = (C_D, P)$, 事实-维关系 $R \subseteq F \times \text{Dim}$, $\forall f \in F (\exists e_i \in C_D ((f, e_i) \in R) \wedge (e_i, \pi e_k) \wedge (0 < P \leq 1)) \Leftrightarrow f \sim_p e_k$ 。若 $p=1$, 我们称事实 f 被值 e_k 完全描述。

定义 6 (多维模型) 多维模型(MO)定义成这样一个四元组 (S, F, D_M, R_M) , 其中 S 为 n 维事实模式, F 为事实, $D_M = \{D_i, i=1, \dots, n\}$ 是相应的维集, $R_M = \{R_i, i=1, \dots, n\}$ 为其事实-维关系集合, 因此 $\forall i ((f, e) \in R_i \Rightarrow ((f \in F) \wedge \exists C_{D_i} \in D_i (e \in C_{D_i})))$ 。

2.2 功能结构视图

定义 7 (功能结构视图) 多维模型 MO 的功能结构视图 FV_{MO} 用一个三元组 (F_A, D_A, agt) 集合来表示。其中 F_A 为该模型的事实属性, D_A 为对相应度量属性进行分析的维度集合, 聚合类型 $\text{agt} = \{\Sigma, \Phi, C\}$ 。 Σ 表示能进行求和、求平均值、求最大值、最小值等操作的聚合; Φ 表示能进行求平均值、求最大、最小值操作, 而不能进行求和操作的聚合; C 表示只能进行计数操作的聚合。

根据事实属性的实际意义和数据类型, 不是所有的聚合函数都能应用于所有事实属性而产生有意义的聚合结果, 这个功能结构视图说明了某一事实属性可以运用相应的聚合操作(这依赖于属性聚合类型 agt) 沿着相应维集 D_M 进行有意义的聚合分析。

3 模型操作描述

由于维层次间的部分包含关系, 在对模型进行各种操作时, 它会给其结果带来一些不准确的信息, 下面就从这个角度进行详细分析。在模型的操作描述中, 对于一元运算, 设存在一个 n 维的多维对象 $M = \{S, F, D_M, R_M\}$, 其中 S 为 n 维事实模式, F 为事实, $D_M = \{D_i, i=1, \dots, n\}$ 是该多维对象的维集, $R_M = \{R_i, i=1, \dots, n\}$ 为其事实-维关系的集合。对于二元运算, 设存在两个 n 维的多维对象, $M_1 = \{S_1, F_1, D_M^1, R_M^1\}$, $M_2 = \{S_2, F_2, D_M^2, R_M^2\}$, 其中 $D_M^1 = \{D_i^1, i=1, \dots, n\}$, $D_M^2 = \{D_i^2, i=1, \dots, n\}$, $R_M^1 = \{R_i^1, i=1, \dots, n\}$, $R_M^2 = \{R_i^2, i=1, \dots, n\}$ 。设多维模型 M_1 中维 D_i 对应的维层次集合 $C_{D_i}^1 = \{C_{D_i}^{1j}, j=1, \dots, k\}$, 我们用符号 Dim_i^1 来代替 $\cup_j C_{D_i}^{1j}$, 即 Dim_i^1 为 M_1 中维 D_i 的所有层次集合。

3.1 基本代数操作

类似于传统的关系代数, 在这一部分将定义多维数据模型的一些基本代数操作(主要有交、并、差、笛卡尔集、选择、投影、连接等), 其中笛卡尔集、投影与连接操作并不受维层次间部分包含关系的影响, 其定义可以参照文[5, 6, 11]来得出。

定义 8 (维的并操作 $\cup^D$) 设维 $D_1 = (C_{D_1}, \pi_{D_1})$ 与维 $D_2 = (C_{D_2}, \pi_{D_2})$ 是模式相同的维, $D' = D_1 \cup^D D_2 = (C_{D'}, \pi_{D'})$, 其中 $C_{D'} = \{C_{D_1} \cup C_{D_2}, i=1, \dots, n\}$, 且满足下列两个条件之一:

- $\forall (e_1, e_2, p) \in C_{D'} \times C_{D'} \times (0, 1] \Rightarrow (((e_1, e_2, p) \in C_{D_1} \times C_{D_1} \times (0, 1]) \wedge ((e_1, e_2, p) \notin C_{D_2} \times C_{D_2} \times (0, 1])) \vee (((e_1, e_2, p) \notin C_{D_1} \times C_{D_1} \times (0, 1]) \wedge ((e_1, e_2, p) \in C_{D_2} \times C_{D_2} \times (0, 1])))$
- $\forall (e_1, e_2, p) \in C_{D'} \times C_{D'} \times (0, 1] \Rightarrow (\exists (p_1, p_2) \in (0, 1] \times (0, 1] \wedge ((e_1, e_2, p_1) \in C_{D_1} \times C_{D_1} \times (0, 1]) \wedge ((e_1, e_2, p_2) \in C_{D_2} \times C_{D_2} \times (0, 1]) \wedge (p = \max(p_1, p_2))))$

在上述表达式中, 其中 $C_{D'} \in \text{Anc}(C_{D_1}), C_{D_1} \in \text{Anc}(C_{D_2}), C_{D_2} \in \text{Anc}(C_{D_1})$ 。

定义 9 (多维对象的并操作 $\cup$) 设 M_1 与 M_2 两个 n 维对象, 其中 $S_1 = S_2$, 多维对象的并操作 $\cup$ 描述如下: $M' = M_1 \cup M_2$, 其中 $S' = S_1 = S_2, F' = F^1 \cup F^2, D'_M = D_M^1 \cup D_M^2 = \{D_i^1 \cup D_i^2, i=1, \dots, n\}, R'_M = R_M^1 \cup R_M^2 = \{R_i^1 \cup R_i^2, i=1, \dots, n\}$ 。

可以仿照上面并操作的描述来定义模型的交操作与差操作, 要注意的是描述维层次间部分包含关系的包含度会发生变化。

定义 10 (选择操作 σ) 给定一谓词 $q: \text{Dim}_1 \times \dots \times \text{Dim}_n \times (0, 1] \times \dots \times (0, 1] \rightarrow \{\text{true}, \text{false}\}$, $\sigma[q](M) = M' = (S', F', D_M, R_M)$, 其中 $S' = S, F' = \{f \in F \mid \exists (e_1, \dots, e_n, p_1, \dots, p_n) \in \text{Dim}_1 \times \dots \times \text{Dim}_n \times (0, 1] \times \dots \times (0, 1] (q(e_1, \dots, e_n, p_1, \dots, p_n) \wedge (\exists p'_1, \dots, p'_n) \in (0, 1] \times \dots \times (0, 1] \wedge \dots \wedge (f \sim_{p'_1} e_1) \wedge (f \sim_{p'_n} e_n) \wedge (p'_1 \geq p_1) \wedge \dots \wedge (p'_n \geq p_n)))\}$, $D_M = D_M, R_M = \{R_i', i=1, \dots, n\}$, 且 $R_i' = \{(f, e) \in R_i \mid f \in F'\}$ 。

3.2 模型分析操作

OLAP 的基本分析操作主要有钻取 (Drill-up, Drill-down)、切片、切块、旋转等, 其中切片和切块是在一部分维上选定值后, 关心度量数据在剩余维上的分布。如果剩余的维只有两个, 则是切片; 如果有三个或以上, 则是切块。切片与切块操作可以根据选择操作来描述。旋转操作只是变换维的方向, 即在表格中重新安排维的放置, 它只是展现方面的变化, 对操作结果集并没有影响。另外, 分组操作与聚合操作也是两个经常用到的 OLAP 分析操作, 下面主要对分组、聚合、钻取操作进行描述。

定义 11 (分组操作 Group) 该操作符就是根据相同的维值来对事实进行分组, 具体定义如下: $\text{Group}: \text{Dim}_1 \times \dots \times \text{Dim}_n \rightarrow 2^F$ 。如: $\forall (e_1, \dots, e_n) \in \text{Dim}_1 \times \dots \times \text{Dim}_n$, 给定 $(p_1 \times \dots \times p_n) \in (0, 1] \times \dots \times (0, 1]$, $\text{Group}(e_1, \dots, e_n) = \{f \mid (f \in F) \wedge (\exists (p'_1, \dots, p'_n) \in (0, 1] \times \dots \times (0, 1] \wedge (f \sim_{p'_1} e_1) \wedge \dots \wedge (f \sim_{p'_n} e_n) \wedge (p'_1 \geq p_1) \wedge \dots \wedge (p'_n \geq p_n))\}$ 。

由于维层次间的部分包含关系, 这就给分组操作带来了一定的不精确性。对于这种不精确性的处理, 一般有以下三种方法:

保守法: 分组结果中只包含那些被相应维值完全描述的事实, 一般用于获取分组结果的下限。

乐观法: 分组结果中包含所有被相应维值 p -characterization 描述的事实, 一般用于计算分组结果的上限。

折中法: 在定义 11 给的示例中, 这里就给出了一组描述度阈值, 凡是大于该值的事实就包含在分组结果中, 否则就丢弃掉。该方法还有另一种表示, 也就是它并没有设定一组描述度阈值, 首先像乐观法一样, 计算结果中包含所有被相应维值 p -characterization 描述的事实, 然后再用这个结果乘以一个合适的权值, 从而得出最终的分组结果, 这个权值与描述度紧密相关。

定义 12 (聚合操作 α) 为了对聚合操作进行描述, 必须给定一个聚合函数 g 和一组维的层次集合 $\{C_i \in D_i, i=1, \dots, n\}$, 同时也新增一个模式为 D_{schema}^{n+1} 的维 D_{n+1} 来保存聚合结果, 则聚合操作 α 定义如下:

$M' = \alpha[g, C_1, \dots, C_n](M) = (S', F', D_M, R_M)$, 其中: $S' = (F_{schema}(M'), D_{schema}(M'))$,

$F_{Schema}(M') = 2^{F_{Schema}(M)}, D_{Schema}(M') = \{D_{Schema}^i(M'), i = 1, \dots, n\} \cup \{D_{Schema}^{n+1}\};$

$D_{Schema}^i(M') = (T_{D_i}, \bar{\pi}_{D_i}, \perp_{D_i}, ALL_{D_i}), T_{D_i} = \{T_{b_i} \in T_{D_i} | T_{b_i} \bar{\pi}_{D_i} T_{b_i}\}, \bar{\pi}_{D_i} \text{ 为 } \pi_{D_i} \text{ 定义在 } T_{D_i} \text{ 的部分包含关系}, \perp_{D_i} = Level(C_i), ALL_{D_i} = ALL_{D_i};$

$F' = \{g(Group(e_1, \dots, e_n)) | ((e_1, \dots, e_n) \in C_1 \times \dots \times C_n) \wedge Group(e_1, \dots, e_n) \neq \Phi\};$

$D_M = \{D_i, i = 1, \dots, n\} \cup \{D_{n+1}\}, D_i = (C_{D_i}, \bar{\pi}_{D_i}^b);$

$C_{D_i} = \{C_{b_i}^b | C_{b_i}^b \in D_i \wedge Level(C_{b_i}^b) \in T_{D_i}\}, \bar{\pi}_{D_i}^b \text{ 为 } \pi_{D_i} \text{ 定义在 } C_{D_i}^b \text{ 的部分包含关系};$

$R_M = \{R_i, i = 1, \dots, n\} \cup \{R_{n+1}\},$

$R_i = \{(f', e_i) | \exists (e_1, \dots, e_n) \in C_1 \times \dots \times C_n ((f' = Group(e_1, \dots, e_n)) \wedge (f' \in F') \wedge (e_i = e_i))\},$

$R_{n+1} = \bigcup_{(e_1, \dots, e_n) \in C_1 \times \dots \times C_n} \{(Group(e_1, \dots, e_n), g(Group(e_1, \dots, e_n))) | Group(e_1, \dots, e_n) \neq \Phi\}.$

定义 13 (Drill-up); Drill-down 操作就是沿着维层次向上聚合, 获得粒度较粗的概要信息。关于其形式化定义可以参照聚合操作 α 的定义来描述, 需要注意的是它并没有新增一个维来保存聚合结果, 在这里就不再重复定义了。

定义 14 (Drill-down) Drill-down 操作就是沿着维层次向下钻取, 获取更加详细的信息。关于它的描述, 必须给定一个聚合函数 g 和一组维的层次集合 $\{C_i \in D_i, i = 1, \dots, n\}$, 同时还必须提供一个原始的多维对象, Drill-down 操作就相当于在原始的多维对象上运用相应的聚合函数来得到所要求的细节信息。关于其具体的形式化描述这里就不再重复了。

3.3 模型维护操作

关于模型维护操作的描述, 可以分为两类, 一类是模型内容的维护, 另一类就是模型定义结构的维护(也称为模式的维护)。由于模型内容的维护相对来说比较简单, 本文就不作详细讨论。

定义 15 (增加维 $ADim$) $M' = ADim(M, D_{Add}) = (S', F, D_M, R_M)$, 其中 D_{Add} 表示增加的维, 其模式定义为 D_{Schema}^{Add} , 则 $S' = (F_{Schema}(M), D_{Schema}(M')), D_{Schema}(M') = \{D_{Schema}^i(M), i = 1, \dots, n\} \cup \{D_{Schema}^{Add}\}, D_M = \{D_i, i = 1, \dots, n\} \cup \{D_{Add}\}, R_M = \{R_i, i = 1, \dots, n\} \cup \{R_{Add}\}, R_{Add} = \{(f, e) | f \in F, e = \phi\}.$

注意, 新增维的属性值初始设定为空值, 以后可以使用内容维护操作进行修改。

定义 16 (删除维 $DDim$) $M' = DDim(M, D_{Del}) = (S', F', D_M, R_M)$, 其中 D_{Del} 表示要删除的维, 其模式定义为 D_{Schema}^{Del} , 则 $S' = (F_{Schema}(M'), D_{Schema}(M')), F_{Schema}(M') = 2^{F_{Schema}(M)}, D_{Schema}(M') = \{D_{Schema}^i(M), i = 1, \dots, n\} - \{D_{Schema}^{Del}\}, F' = \{sum(Group(e_1, \dots, e_n)) | ((e_1, \dots, e_n) \in C_1 \times \dots \times C_n) \wedge Group(e_1, \dots, e_n) \neq \Phi\}, D_M = \{D_i, i = 1, \dots, n\} - \{D_{Del}\}, R_M = \{R_i, i = 1, \dots, n-1\}, R_i = \{(f', e_i) | f' \in F'\}.$

定义 17 (增加度量属性 $AMEa$) $M' = AMEa(M, Meas) = (S', F', D_M, R_M)$, 其中 $S' = (F_{Schema}(M'), D_{Schema}(M)), F_{Schema}(M') = F_{Schema}(M) \cup \{Meas\}; F' = \{f' | f \times (\phi, \dots, \phi_i)\}, (\phi_1, \dots, \phi_i) \text{ 为新度量属性的值, 在这里初始设为空, 以后可以使用内容维护操作进行修改}; R_M = \{R_i, i = 1, \dots, n\}, R_i = \{(f', e_i) | f' \in F'\}.$

删除度量属性操作 $DMEa$ 可以通过在度量属性上进行投影来得到, 在这里就不再定义了。

定义 18 (增加维层次属性 AHa) $M' = AHa(M, H_{Add})$

$= (S', F, D_M, R_M)$, 其中 H_{Add} 表示增加的维层次, 其层次类型为 T_{Add} , 则 $S' = (F_{Schema}(M), D_{Schema}(M')), D_{Schema}(M') = \{D_{Schema}^i(M'), i = 1, \dots, n\};$ 如果 $T_{Add} \in D_{Schema}^i(M')$ 则 $D_{Schema}^i(M') = (T_{D_i}, \bar{\pi}_{D_i}, \perp_{D_i}, ALL_{D_i}), T_{D_i} = \{T_{b_i} \in T_{D_i}\} \cup \{T_{Add}\}, \bar{\pi}_{D_i} \text{ 为定义在 } T_{D_i} \text{ 的部分包含关系, 如果 } \perp_{D_i} \bar{\pi}_{D_i} T_{Add}, \text{ 则 } \perp_{D_i} = \perp_{D_i}, \text{ 否则 } \perp_{D_i} = T_{Add}, ALL_{D_i} = ALL_{D_i};$ 如果 $T_{Add} \notin D_{Schema}^i(M')$, 则 $D_{Schema}^i(M') = D_{Schema}^i(M); D_M = \{D_i, i = 1, \dots, n\},$ 若 $H_{Add} \in D_i$, 则 $D_i = (C_{D_i}, \bar{\pi}_{D_i}^b), C_{D_i}^b = \{C_{D_i}^b\} \cup \{H_{Add}\}, \bar{\pi}_{D_i}^b \text{ 为定义在 } C_{D_i}^b \text{ 的部分包含关系};$ 若 $H_{Add} \notin D_i$, 则 $D_i = D_i; R_M = \{R_i, i = 1, \dots, n\}.$

类似地, 还可以定义减少维层次属性的操作 DHa 。

定理 1 上述操作对于数据仓库的概念模型是封闭的。

证明: 所谓封闭的意思就是说在原有的多维数据模型上进行这些操作得到的结果仍然为多维数据仓库模型, 从上述操作的描述中可以看出, 结果显然是正确的。

4 模型的构建

对于多维数据模型的构建, 目前主要有两类方法, 一类是对基于传统的 E/R 建模方法进行了扩展, 主要代表有 StarER 建模方法^[12]、ME/R 建模方法^[13]、DWER 建模方法^[14], 这类方法不能通过传统的 E/R 建模工具来构建相应的模型, 因此还必须专门开发了一套建模工具来支持该方法, 用户使用起来较困难。另外, 该方法对动态部分(预测查询行为)和功能视图部分的描述能力相对来说还非常弱。

另一类方法就是以软件工程领域中广泛应用的统一建模语言 UML 为基础, 针对多维数据模型的具体特点, 对 UML 进行了扩展, 通过在 UML 中定义一些新的构造类、标记值, 以及增加一些约束来指导开发人员进行多维数据概念建模^[14]。这种基于 UML 的多维数据建模方法相对其它建模方法来说, 它的一个显著优点就是建立在一种广泛接受的面对象建模语言的基础上, 从而使得开发人员进行具体应用时, 不必重新学习一些新的模型及其相关概念, 便于用户掌握。关于其建模工具, 作者在 Rose 这一广泛应用的建模工具的基础上开发了一套插件, 用户使用起来非常方便, 而且也利于该工具的推广^[14]。

结论 本文以维层次间的部分包含关系为基础, 对多维数据概念模型的形式化描述进行了初步研究, 并首次从静态结构视图与功能结构视图两个方面对多维数据概念模型进行了详细的描述; 同时还从基本代数操作、模型分析操作、模型维护操作三个方面对该模型支持的操作进行了定义; 最后还给出了该模型的构建方法。本文还存在许多不完善的地方, 有待于进一步研究, 在对部分包含关系进行推导时, 怎样利用部分包含关系给聚合操作带来的不精确性信息, 以及怎样在已有的 OLAP 系统上应用该模型等问题还有待于进一步研究。

参考文献

- 1 Firestone J M. Dimensional Modeling and E-R Modeling in the Data Warehouse [R]: [DKMS-White Paper No. Eight], June 22, 1998
- 2 段云峰, 等. 数据仓库基础 Data Warehousing Fundamentals [M]. 北京: 电子工业出版社, 2004

(下转第 120 页)

用文[1]的体系结构,可以提高系统效率。表1分析在文[1]框架下OFT、LKH的性能。而HEGKM中,当用户离开或加入,组更新只需一个组播包和一次迭代,并大大减少密钥更新量。

表1 典型密钥管理方案性能比较

指标	单用户离开				单用户加入			
	N_R	C_R	N_E	N_{step}	N_R	C_R	N_E	N_{step}
LKH	$k-1$	$2k-3$	$2k-3$	$2k-3$	k	$2k$	$2k$	$2k$
LFT	$k-2$	$k-1$	$k-1$	$k-1$	k	k	k	k
HEGKM	$m+1$	1	$k-1$	1	m	1	1	1

注: $h=\log(t/m, k=h+m)$

4.2 仿真结果

实验中,用户总数为 2^{11} ,子组数为2。实验采用改进的LKH^[3]和OFT^[4]方案,对相同的批量用户改变模式的更新效率进行比较。

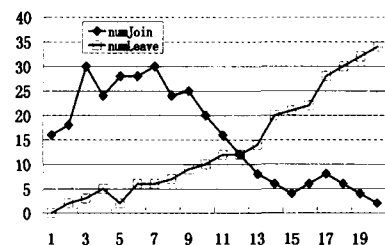


图2 用户改变模式

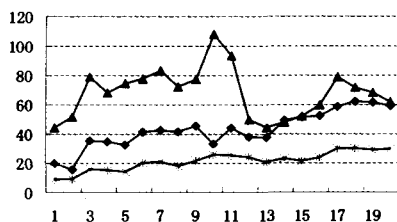


图3 时间开销比较

图2为实验中用户改变量的变化图,开始阶段加入数高于离开数,随后离开数越来越多,加入数越来越少。由于OFT将加入用户形成一族后再整体加入,因此当加入数高于离开数时,OFT的优势明显。但图3、4说明HEGKM算法的时间开销和带宽开销最小,组播带宽开销与加入用户数几乎无关。

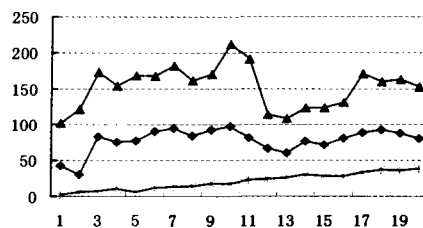


图4 带宽开销比较

结论 本文针对具有多层接入控制的安全组播,提出基于椭圆曲线密钥体制的斜树密钥管理方案,减小了系统的密钥存储开销和算法的复杂性,效率分析和仿真实验结果表明算法的效率较其它典型方案有较大提高。

参考文献

- 1 Sun Y, Liu K J R. Scalable hierarchical access control in secure group communications[A]. INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies[C], 2004, 2(7-11): 1296~1306
- 2 Wallner D, Harder E, Agee R. Key Management for Multicast: Issues and Architectures. RFC 2627, 1999
- 3 Pegueroles J, Rico-Novella F, Hernandez-Serrano J. Improved LKH for batch rekeying in multicast groups [A]. In: International Conference on Information Technology [C]. New York(USA): IEEE, 2003. 269~273
- 4 Sherman A T, McGrew D A. Key establishment in large dynamic groups using one-way function trees [J]. IEEE Trans. on Software Engineering, 2003, 29(5): 444~458
- 5 张方国, 万育民. 超椭圆曲线密码体制的研究与进展[J]. 电子学报, 2002, 30(1): 126~131
- 6 Yang Zongkai, Xu Shouzhi, Tan Yunmeng. An efficient key updating scheme for multicast key management [A]. Proceedings of the SPIE - The International Society for Optical Engineering [C], 2005, 5626(2): 1096~1104

(上接第86页)

- 3 Jensen C S, Pedersen T B. Mobile E-Services and Their Challenges to Data Warehousing. Datenbank Rundbrief, 2001, 27: 8~16
- 4 Jensen C S. Research Challenges in Location-Enabled M-Services. In: Proceedings of the Third International Conference on Mobile Data Management. Singapore: IEEE Computer Society Press, 2002. 3~7
- 5 Jensen C S, Kligys A, Pedersen T B, et al. Multidimensional Data Modeling for Location-Based Services. The International Journal on Very Large Data Bases, 2004, 13(1): 1~21
- 6 李建中, 高宏. 一种数据仓库的多维数据模型. 软件学报, 2000, 11(7): 908~917
- 7 Pedersen T B, Jensen C S. Multidimensional Data Modeling for Complex Data. In: Proc. of 15th ICDE, IEEE Computer Society, pp. Sydney, Australia, 1999. 336~345
- 8 Franconi E, Kamble A. The GMD Data Model for Multidimensional Information: A Brief Introduction. Data Warehousing and Knowledge Discovery, 5th International Conference, DaWaK 2003, Prague, Czech Republic, 2003
- 9 Trujillo J. The GOLD model: An Object Oriented multidimensional data model for multidimensional databases. In: Proceedings

- of the 16th International Conference on Advanced Information Systems Engineering (CAI SE-04), Riga, Latvia, 2004
- 10 Tsois A, Karayannidis N. MAC: Conceptual Data Modeling for OLAP. In: Proc. of the International Workshop on Design and Management of Data Warehouses (DMDW' 2001) Interlaken, Switzerland, 2001
- 11 Lechtenbörger J, Vossen G. Multidimensional Normal Forms for Data Warehouse Design. Information System, 2003, 28(5): 415~434
- 12 陈明, 吴国文, 施伯乐. 数据仓库概念模型的设计. 小型微型计算机系统, 2002, 23(12): 1453~1458
- 13 Tryfona N, Busborg F, Borch Christiansen J G. starER: A Conceptual Model for Data Warehouse Design. In: Proc. of the ACM 2nd Intl. Workshop on Data warehousing and OLAP (DOLAP' 99)
- 14 Blaschka M. FIESTA: A Framework for Schema Evolution in Multidimensional Databases: [PhD thesis, Technische]. Universität München, Germany, 2000
- 15 陆昌辉, 邓苏, 等. 基于UML的多维数据概念建模方法的研究. 系统工程与电子技术, 2004, 26(4): 522~525