ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.2, February 2007, pp.259−267
DOI: 10.1360/jos180259

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

Peer 数据管理系统中的视图维护策略*

覃 飙+, 王 珊, 杜小勇

(中国人民大学 信息学院,北京 100872)

View Maintenance Strategy in Peer Data Management Systems

QIN Biao+, WANG Shan, DU Xiao-Yong

(School of Information, Renmin University of China, Beijing 100872, China)

+ Corresponding author: Phn: +86-10-62511554, E-mail: qinbiao@ruc.edu.cn, http://www.ruc.edu.cn

Qin B, Wang S, Du XY. View maintenance strategy in peer data management systems. *Journal of Software*, 2007,18(2):259−267. http://www.jos.org.cn/1000-9825/18/259.htm

Abstract: In this paper, a strategy for view maintenance in peer data management systems (PDMSs) is proposed. First, a hybrid peer architecture is presented, in which peers prefer peer-to-peer architecture to super-peer based architecture. Then, if a view is reformulated along a semantic path, it can retrieve data from many data sources. So the peer view, local view and global view are proposed. A global view maintenance is turned into the maintenance of all related local views if join operations are confined in each local PDMS, and a query can gain the same result if it is posed over any peer in the same semantic path. Furthermore, a view may be a conjunctive query which relates to several tables in PDMSs. According to the application, Mork's rules are extended. Based on the new rule system, the update data are propagated in PDMS. According to the new rule system, a view maintenance algorithm in PDMS is proposed. Finally, simulation experiments are carried out in the SPDMS (schema-mapping-based PDMS). The simulation results show that the proposed view maintenance strategy has better performance than that of Mork's.
Key words: peer data management system; semantic path; incremental view maintenance

摘 要: 研究 Peer 数据管理系统(peer data management system,简称 PDMS)中的视图维护策略.首先提出一种混合的 P2P 架构,在该架构中,Peer 优先选择基本的 P2P 架构而不是超级 Peer 架构.如果一个视图沿着 PDMS 的一条语义链路传播,那么它通过重构能够从许多数据源检索到数据,因此扩展了视图的定义,提出 peer 视图、局部视图和全局视图;并且从一条语义链路的任意节点出发,同一查询都能够得到相同的结果.在 PDMS 中,连接操作限制在各个局部 PDMS 中,这样,全局物化视图的维护就转化为对各相关局部物化视图的维护.在 PDMS 中,一个视图可能涉及到多个表的连接,根据该应用的需要扩展了 Mork 的规则系统.根据扩展了的规则系统进行更新数据传播,同时依据该规则系统提出一种算法来维护 PDMS 中的视图.最后进行实验验证.实验结果表明,该视图维护策略比 Mork 的视图维护策略的性能要好.
关键词: Peer 数据管理系统;语义链路;增量视图维护
中图法分类号: TP311 文献标识码: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60473069, 60496325, 60503038, 60573092 (国家自然科学基金)

Received 2005-10-07; Accepted 2006-03-09

P2P 计算是当前的一个研究热点.它影响了许多领域,比如网络、分布式系统、信息系统、算法和数据库系统.通过负载分布从而避免集中式的瓶颈,P2P 系统有利于在全局范围内部署应用.研究者们近年来对 Peer 数据管理系统(peer data management system,简称 PDMS)中的数据共享问题进行了广泛而深入的研究.Halevy[1]从数据集成的角度来研究 PDMS,他在 Piazza PDMS 中提出了采用自主的、离散共享的方法来管理数据,每个 Peer 能够提供数据并且把这些数据和一个已经存在的模式联系起来,或者为其他 Peer 建立新的模式映射以方便查询,或者在已经存在的模式或数据源之间定义新的联系.Kementsietsidis[2]研究了 P2P 数据共享系统中的数据映射问题,解决了与映射表有关的语义和算法;PeerDB[3]采用一种信息检索的途径进行查询重构;Chatty Web[4]集中考虑了 gossip 协议在交换语义映射中的作用.

PDMS 中的数据一致性维护已受到越来越多的关注.Shah[5]提出了一种基于推的数据分发架构,它能够确保数据的一致性、可恢复性和高效性.Gao[6]提出了一种 CC-Buddy 的架构来维护动态数据在 P2P 系统中的一致性;对于异构数据一致性问题,CC-Buddy 在采用推方法分发数据时协调它.Mork[7]提出在大规模数据共享系统中采用物化视图,并提出了一套规则来维护物化视图.然而在 PDMS 中,随着相关数据源数据量的增加和系统的增长,如果全局视图存储在一个节点中,将很难满足 PDMS 扩展性的需要.基于这些分析和我们的研究基础[8,9],本文提出了 PDMS 中的一种物化视图维护策略.其主要贡献在于:提出了一种混合的 Peer 架构,其中 Peer 优先选择 P2P 架构而不是超级 peer 架构;然后,根据应用需要扩展了视图的定义,提出了全局视图、局部视图和 Peer 视图;如果连接操作限制在各个局部 PDMS 中,那么,全局视图的维护就转变为各个相关局部视图的维护;扩展了 Mork[7]的规则系统以便进行数据的更新管理和物化视图维护.

本文第 1 节介绍 SPDMS(schema-mapping-based PDMS)的逻辑模型.第 2 节讨论在 P2P 环境下物化视图的维护策略.第 3 节为实验描述与分析.第 4 节描述相关研究工作.最后进行全文总结.

1 SPDMS 的逻辑模型

传统的数据集成系统解决了大规模数据共享中的部分问题,而其主要局限在于需要一个全局中间模式.在 PDMS 中,各个数据源是不断变化的,并且新数据源要不断加入到这个系统中,因此,全局中间模式就成为 PDMS 的一个瓶颈[1].在 PDMS 中,基于查询需要每个 Peer 创建自己的模式,这些模式通过两两之间的映射而相互关联.这样,每个 Peer 仅需要维护少量与其密切相关的模式映射,因此,当一个 Peer 的模式发生改变时,只有少量与其密切相关的映射需要更新.当一个查询从某个 Peer 上提出来时,它通过重构能够从许多数据源检索到数据,因此我们有下面的定义.

定义 **1.1**. 与 Peer P 有模式映射关系的 Peer 组成的集合用 $APS(P)$表示,因此 $APS(P)=\{P_i|P_i\rightarrow P \text{ or } P\rightarrow P_i\}$,这里,$P_i\rightarrow P$ 表示系统建立了从 P_i 到 P 的模式映射.

定义 **1.2**. 在一个 PDMS 中,从 Peer P 中发出的查询在一组模式映射下通过重构能够到达的 Peer 所组成的集合称为该查询的传递闭包,用 P_Q^+ 表示. P_Q^+ 中所有 Peer 共同形成了该查询的语义链路,并且 P 是该语义链路的起点;该查询与其在 P_Q^+ 中其他 Peer 上重构形式的并集形成了该查询的语义.

并不是所有的 Peer 都是等价的,可以利用 PDMS 系统中 Peer 的不同能力来组建一个高效的系统架构.其中,少量 Peer 称为超级 Peer,承担着专门的责任,包括 Peer 聚合、Peer 路由和形成中间模式.我们的 SPDMS 采用了一个混合的 P2P 架构,在该架构中,Peer 更愿意采用 P2P 架构而不是超级 Peer 架构[10].如果一个查询从 Peer A 中提出,则它将在一条语义链路的 Peer 之间进行传递;如果该语义链路中的某个 Peer 处于脱机状态,则该查询将通过其超级 Peer 传递到下一个 Peer.例如,在图 1 中,如果一个查询从 UPenn 中提出并且它能够传递到 Berkeley,则其优先的路径为 UPenn→DB-Projects→UW→Stanford→Berkeley;如果 Stanford 脱机,则传递的路径为 UPenn→DB-Projects→SP→Berkeley.

在 SPDMS 中有两类超级 Peer.第 1 类超级 Peer 具有如下特性:它维护着本局部 PDMS 中 Peer 之间的模式映射,或者维护着本局部 PDMS 中 Peer 与其他局部 PDMS 中 Peer 之间的模式映射,比如 Stanford-Berkeley PDMS 中的 SP.第 2 类超级 Peer 具有如下特性:它不仅提供本局部 PDMS 的中间模式,而且具有第 1 类超级 Peer 的后

一个特性,比如 UPenn-UW-DBProjects PDMS 中的 DB-Projects.

我们用图 1 来说明如何构建 SPDMS,箭头表示 Peer 的模式之间有映射关系.如果 Stanford 和 Berkeley 之间想建立模式映射,它们有两件事情要做:第一,Stanford 把从它到 Berkeley 的模式映射注册到 SP 中,同时,Berkeley 也把从它到 Stanford 的模式映射注册到 SP 中;第二,Stanford 把 Berkeley 加入到 APS(Stanford),同时,Berkeley 把 Stanford 加入到 APS(Berkeley)中,这样,Stanford 和 Berkeley 之间的模式映射就建立起来了.

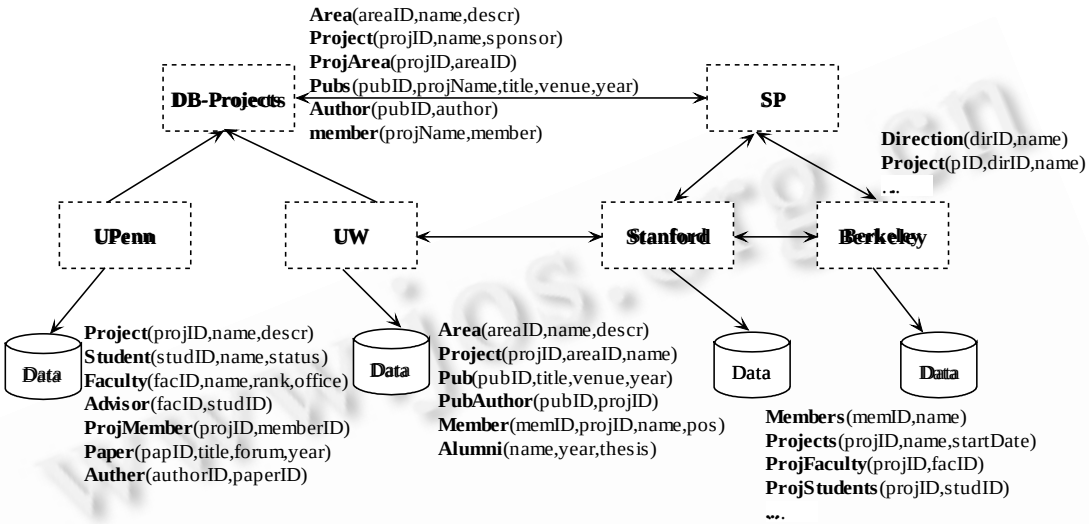


Fig.1 Architecture of SPDMS

图 1 SPDMS 的架构

如果 UW 和 Stanford 要建立映射,也有两件事情要做:第一,UW 在 DB-Projects 中注册从它到 Stanford 的模式映射,Stanford 在 SP 中注册从它到 UW 的模式映射;第二,UW 把 Stanford 加入到 APS(UW),同时,Stanford 把 UW 加入到 APS(Stanford).一旦 UW 和 Stanford 之间的模式映射建立起来了,则从该 PDMS 中任意一个 Peer 发出的查询都就能够访问所有数据源.

2 物化视图维护

2.1 物化视图的定义

假设 $P_1 \xrightarrow{V_m} P_2$ 代表从 P_1 提出的一个物化视图(V_m)能够沿着一条语义链路到 P_2 上重构.我们称 V_m 沿着语义链路的重构为其访问关系,它具有如下特性:

自反性: $P_1 \xrightarrow{V_m} P_1$.

传递性:如果 $P_1 \xrightarrow{V_m} P_2$ 和 $P_2 \xrightarrow{V_m} P_3$ 成立,那么 $P_1 \xrightarrow{V_m} P_3$ 成立.

对称性:如果 $P_1 \xrightarrow{V_m} P_2$ 成立,那么 $P_2 \xrightarrow{V_m} P_1$.

如果一个 PDMS 中的所有 Peer 形成一个集合,则一个 V_m 的访问关系形成该集合上的等价关系.类似于定义 1.2,我们有如下定义:

定义 **2.1**. 在一个 PDMS 中,从 Peer P 中发出的 V_m 在一组模式映射下通过重构能够到达的 Peer 所组成的集合称为 V_m 的传递闭包,用 $P_{V_m}^+$ 表示. $P_{V_m}^+$ 中所有 Peer 共同形成了 V_m 的语义链路,并且 P 是该语义链路的起点;V_m 与其在 $P_{V_m}^+$ 中其他 Peer 上重构形式的并集形成了 V_m 的语义.

定理 **1**. 设 A 和 B 是一个 PDMS 中任意两个 Peer ($A \neq B$),如果 $A \xrightarrow{V_m} B$ 成立,则 $A_{V_m}^+ = B_{V_m}^+$ 成立.

证明:因为 $A \xrightarrow{V_m} B$ 成立,并且 V_m 的访问关系具有对称性,所以有 $B \xrightarrow{V_m} A; \forall C \in A_{V_m}^+$,根据定义 2.1 有

$A \xrightarrow{V_m} C$;因为 V_m 的访问关系具有传递性,可得 $B \xrightarrow{V_m} C$,所以有 $C \in B^+_{V_m}$,因此 $A^+_{V_m} \subseteq B^+_{V_m}$.

用同样的方法可以证明 $B^+_{V_m} \subseteq A^+_{V_m}$. 所以有 $A^+_{V_m} = B^+_{V_m}$,定理得证. □

定义 **2.2**. 如果一个视图仅能从一个数据源检索到数据,则称其为 Peer 视图;如果从一个 Peer 发出的视图能从其所在局部 PDMS 的多个数据源检索到数据,则称其为局部视图;如果从一个 Peer 发出的视图能够从多个局部 PDMS 检索到数据或者能够从与该 Peer 不同局部 PDMS 的多个数据源检索到数据,则称其为全局视图.

假设这些视图物化,它们分别称为 Peer 物化视图(P_{mv})、局部物化视图(L_{mv})和全局物化视图(G_{mv}),并且把 G_{mv} 在各个局部 PDMS 中的部分称为它的局部实例(GL_{mv}).如果 G_{mv} 存储在一个节点中,就称为集中式 G_{mv};如果 G_{mv} 的各个 GL_{mv} 分别存储在其局部 PDMS 中,就称为分散式 G_{mv}.

一个 PDMS 可能由许多局部 PDMS 组成,例如,图 1 中的 PDMS 是由局部 Stanford-Berkeley PDMS 和 UPenn-UW-DBProjects PDMS 组成的.因为在 SPDMS 中,连接操作被限制在各个局部 PDMS 中,所以,G_{mv} 是所有相关 GL_{mv} 的并集,即 $G_{mv} = \cup_{i=1}^{n} GL_{mv_i}$,这样,分散式 G_{mv} 的维护变为各个相关 GL_{mv} 的维护.分散式 G_{mv} 在各个局部 PDMS 中都有一个邻居超级 Peer 集合($N(SP)$),其元素为其邻居超级 Peer,例如,在图 1 中,$N(SP)$= DB-Projects.在 SPDMS 中,P_{mv} 能够存储在任意 Peer 中;然而,GL_{mv} 和 L_{mv} 的定义存储在其超级 Peer 中,并且其数据存储在本局部 PDMS 中的一个有能力的 Peer 中,我们称其为 Propagation Peer.超级 Peer 维护着如下两种联系:

1) GL_{mv} 和 L_{mv} 的定义与它们的数据之间的联系;
2) GL_{mv} 和 L_{mv} 的定义与它们的数据源之间的联系.

2.2 物化视图的维护策略

在大多数情况下,对物化视图的维护从头开始计算是一种浪费,如果只计算修改后的部分,通常花费的代价更少.我们把计算与基本表变化相关部分的视图变化称为增量视图维护.一个视图如果对于基本表的更新都能够进行自维护,则称该视图是可自维护的[11];否则,该视图是不可自维护的.在 SPDMS 中,一个关系的不同版本采用不同的版本号来区分,比如我们采用 R^t 来表示关系 R 的第 t 个版本;采用版本矢量来区分同一视图的不同版本,视图 V 的版本矢量包含其依赖的各个基本关系的版本号;在此基础上,就有 updategram 和 booster[7]的定义.

定义 **2.3**. updategram 包含一组必要的改变(插入、扫除和修改),把关系从一个版本转变为其后一个版本,比如 $\mu_R^{i,j}$ 包含能使 R^i 变为 R^j 的变化;同样,$\mu_V^{\vec{i},\vec{j}}$ 包含能使 $V^{\vec{i}}$ 变为 $V^{\vec{j}}$ 的变化.

定义 **2.4**. 如果 V 是 SPJ 视图,并且其 FROM 子句中有 n 个关系,它们是 $R_1,R_2,\ldots,R_n$;D 是数据库,而 μ_{R_1} 是 R_1 的 updategram.R_2 相对于 μ_{R_1} 和 V 的 booster 是 D 中 R_2 的子集,其元素与 μ_{R_1} 中元组相关,把 booster 记为 $\beta_V(\mu_{R_1}, R_2)$.

Mork[7]研究了 SPJ 视图的维护方法,他提出一组规则来管理 updategram 和 booster,这组规则一共有 17 个,我们称其前 11 个为基本规则,这些基本规则如下所示:

A1. $\mu_R \overset{V}{\bowtie} S = \mu_{R \overset{V}{\bowtie} S}$;;

A2. $\mu_R \overset{V}{\bowtie} \beta_V(\mu_R, S) = \mu_{R \overset{V}{\bowtie} S}$;

A3. $\Pi_{attributes(S)}\left(\mu_R \overset{V}{\bowtie} S\right) = \beta_V(\mu_R, S)$;

A4. $\beta_V(\mu_R, S) \overset{V}{\bowtie} \beta_V(\mu_R, T) = \beta_V\left(\mu_R, S \overset{V}{\bowtie} T\right)$;

A5. $\left(\mu_R \overset{V}{\bowtie} \beta_V(\mu_R, S)\right) \overset{V}{\bowtie} \beta_V(\mu_R, T) = \mu_R \overset{V}{\bowtie} \left(\beta_V(\mu_R, S) \overset{V}{\bowtie} \beta_V(\mu_R, T)\right)$;

A6. 如果视图为 W,且满足 $\left(R \overset{V}{\bowtie} S\right) \subseteq \left(R \overset{W}{\bowtie} S\right)$,则 $\mu_R \overset{V}{\bowtie} \beta_W(\mu_R, S) = \mu_{R \overset{V}{\bowtie} S}$;

A7. 如果$\left(S\overset{V}{\bowtie}T\right)\subseteq\left(S\overset{U}{\bowtie}T\right)$,并且$\left(S\overset{V}{\bowtie}T\right)\subseteq\left(S\overset{W}{\bowtie}T\right)$,那么,$\beta_U(\mu_R,S)\overset{V}{\bowtie}\beta_W(\mu_R,T)\supseteq\beta_V\left(\mu_R,S\overset{V}{\bowtie}T\right)$;

A8. 如果 $i\le j<k\le l$,那么,$\mu_R^{j,k}\overset{V}{\bowtie}\beta_V(\mu_R^{i,l},S)=\mu^{j,k}_{R\overset{V}{\bowtie}S}$;

A9. 如果$(m=\max(i,k))<(n=\min(j,l))$,那么,$\beta_V(\mu_R^{i,j},S)\overset{V}{\bowtie}\beta_V(\mu_R^{k,l},T)=\beta_W\left(\mu_R^{m,n},S\overset{V}{\bowtie}T\right)$,这里,$V\subseteq W$;

A10. $\mu_R^{i,j}\cup\mu_R^{j,k}=\mu_R^{i,k}$;

A11. 如果 $V\subseteq W$, $V\subseteq X$,并且 $i<k\le j<l$,那么,$\beta_W(\mu_R^{i,j},S)\cup\beta_X(\mu_R^{k,l},S)=\beta_Y(\mu_R^{i,l},S)$,这里,$V\subseteq Y$.

在 SPDMS 中,一个视图可以是多个关系的连接,如下例所示.

例:如果$V=R_1\bowtie R_2\bowtie R_3$且 R_1 发生改变,那么 $\Delta V=\mu_{R_1}\bowtie\beta_V(\mu_{R_1},R_2)\bowtie\beta_V(\mu_{R_1},R_3)$.根据规则 A2 有

$$\Delta V=\mu_{R_1}\bowtie\beta_V(\mu_{R_1},R_2)\bowtie\beta_V(\mu_{R_1},R_3)=\mu_{R_1\bowtie R_2}\bowtie\beta_V(\mu_{R_1},R_3).$$

然而,Mork 的规则没有给出处理 $\mu_{R_1\bowtie R_2}\bowtie R_3$ 的方法,基于该应用需要,我们扩展了其规则系统,提出如下规则:

A2-1. $\mu_{R'}\overset{V}{\bowtie}\beta_V(\mu_R,R_i)=\mu_{R'\overset{V}{\bowtie}R_i}$.

这里,R'是视图 V 中包含 R 的关系子集.该规则表明:包含 R 和其他关系的 updategram 能够与 R 的 booster 交,根据该规则,$\mu_{R_1\bowtie R_2}\bowtie\beta_V(\mu_{R_1},R_3)=\mu_{R_1\bowtie R_2\bowtie R_3}$.在增加了规则 A2-1 以后,我们可以证明上面的规则 A5,其方法如下:

定理 **2**. 假如$\beta_V(\mu_R,S)$是 S 关于μ_R和视图 V 的 booster,那么如下等式成立:

$$\left(\mu_R\overset{V}{\bowtie}\beta_V(\mu_R,S)\right)\overset{V}{\bowtie}\beta_V(\mu_R,T)=\mu_R\overset{V}{\bowtie}\left(\beta_V(\mu_R,S)\overset{V}{\bowtie}\beta_V(\mu_R,T)\right).$$

证明:根据规则 A2 和规则 A2-1,我们可以得到如下等式:

$$\left(\mu_R\overset{V}{\bowtie}\beta_V(\mu_R,S)\right)\overset{V}{\bowtie}\beta_V(\mu_R,T)=\mu_{R\overset{V}{\bowtie}S}\overset{V}{\bowtie}\beta_V(\mu_R,T)=\mu_{R\overset{V}{\bowtie}S\overset{V}{\bowtie}T}.$$

根据规则 A2 和规则 A4,我们可以得到如下等式:

$$\mu_R\overset{V}{\bowtie}\left(\beta_V(\mu_R,S)\overset{V}{\bowtie}\beta_V(\mu_R,T)\right)=\mu_R\overset{V}{\bowtie}\beta_V\left(\mu_R,S\overset{V}{\bowtie}T\right)=\mu_{R\overset{V}{\bowtie}S\overset{V}{\bowtie}T}.$$

因此有$\left(\mu_R\overset{V}{\bowtie}\beta_V(\mu_R,S)\right)\overset{V}{\bowtie}\beta_V(\mu_R,T)=\mu_R\overset{V}{\bowtie}\left(\beta_V(\mu_R,S)\overset{V}{\bowtie}\beta_V(\mu_R,T)\right)$,则定理成立. □

本文重点研究如何采用上面的规则系统进行 G_{mv} 和 L_{mv} 维护,而执行计划将另文加以讨论.如果 V_m 与一个 Peer 中任何关系有关联,则称该 Peer 为 Viewing Peer.如果一个 Peer 临时存储更新数据,则称该 Peer 为 Temp Peer.如果$V_m=R_1\bowtie R_2\bowtie\ldots\bowtie R_n$,根据以上规则,系统 G_{mv} 维护策略如下:

第一,如果 R_i 发生了更新,相应局部 PDMS 的有关 Peer 就把更新数据传递给 Propagation Peer.如果该视图是可以自维护的,则仅需 Viewing Peer 把 μ_{R_i} 传递给 Propagation Peer,以便其完成增量视图维护;否则,$\Delta V=\mu_{R_i}\bowtie\beta_V(\mu_{R_i},R_1)\bowtie\ldots\bowtie\beta_V(\mu_{R_i},R_{i-1})\bowtie\beta_V(\mu_{R_i},R_{i+1})\bowtie\ldots\bowtie\beta_V(\mu_{R_i},R_n)$.故 Viewing Peer 需要把 μ_{R_i} 传递给 Propagation Peer,同时,相关 peer 需要把 $\beta_V(\mu_{R_i},R_j)$ 传递给 Propagation Peer,然后根据规则 A2-1 完成ΔV的计算,最后完成增量视图维护.

第二,如果 Propagation Peer 处于脱机状态,相应局部 PDMS 的相关 Peer 就把更新数据传递给 Temp Peer;如果该视图是可以自维护的,则仅需 Viewing Peer 把 μ_{R_i} 传递给 Propagation Peer,在其上采用规则 A10 完成ΔV的计算,即 $\Delta V=\mu_R^{i,j}\cup\mu_R^{j,k}=\mu_R^{i,k}$;否则,Viewing Peer 需要把 μ_{R_i} 传递给 Propagation Peer,同时,相关 peer 也需要把 $\beta_V(\mu_{R_i},R_j)$ 传递给 Propagation Peer,在其上完成ΔV的计算.该计算分为两步:

1) 根据规则 A2-1,有 $\mu^{j,k}_{R_1\bowtie R_2\bowtie\ldots\bowtie R_n}=\mu_{R_i}^{j,k}\bowtie\beta_V(\mu^{j,k}_{R_{i1}},R_1)\bowtie\ldots\bowtie\beta_V(\mu^{j,k}_{R_{i1}},R_i)\bowtie\beta_V(\mu^{j,k}_{R_{i1}},R_j)\bowtie\ldots\bowtie$

$\beta_V(\mu_{R_i}^{j,k}, R_n)$;

2) 根据规则 A10,有 $\Delta V = \mu_R^{i,j} \cup \mu_R^{j,k} = \mu_R^{i,k}$.一旦 Propagation Peer 连机,它将从 Temp Peer 处接受数据,完成增量视图维护.基于扩展了的规则系统,我们的视图维护算法如下:

算法 **1**. 视图维护算法.

Data_consistency_maintenance_algorithm (local view V_m, relation R_i) {

If the propagation peer is online {

If V_m is self-maintainable

Viewing peer sends μ_{R_i} to propagation peer;

Else {

Viewing peer sends μ_{R_i} to propagation peer;

Other related peer sends $\beta_V(\mu_{R_i}, R_j)$ to propagation peer;

$\Delta V = \mu_{R_i} \bowtie \beta_V(\mu_{R_i}, R_1) \bowtie ... \bowtie \beta_V(\mu_{R_i}, R_{i-1}) \bowtie \beta_V(\mu_{R_i}, R_{i+1}) \bowtie ... \bowtie \beta_V(\mu_{R_i}, R_n)$ by rule A2-1;}

} Else {

If V_m is self-maintainable {

Viewing peer sends μ_{R_i} to temp peer;

$\Delta V = \mu_R^{i,j} \cup \mu_R^{j,k} = \mu_R^{i,k}$ by rule A10;}

Else {

Viewing peer sends μ_{R_i} to temp peer;

Other related peer sends $\beta_V(\mu_{R_i}, R_j)$ to temp peer;

$\Delta V_1 = \mu_{R_i} \bowtie \beta_V(\mu_{R_i}, R_1) \bowtie ... \bowtie \beta_V(\mu_{R_i}, R_{i-1}) \bowtie \beta_V(\mu_{R_i}, R_{i+1}) \bowtie ... \bowtie \beta_V(\mu_{R_i}, R_n)$ by rule A2-1;

$\Delta V = \mu_R^{i,j} \cup \mu_R^{j,k} = \mu_R^{i,k}$ by rule A10; }

}

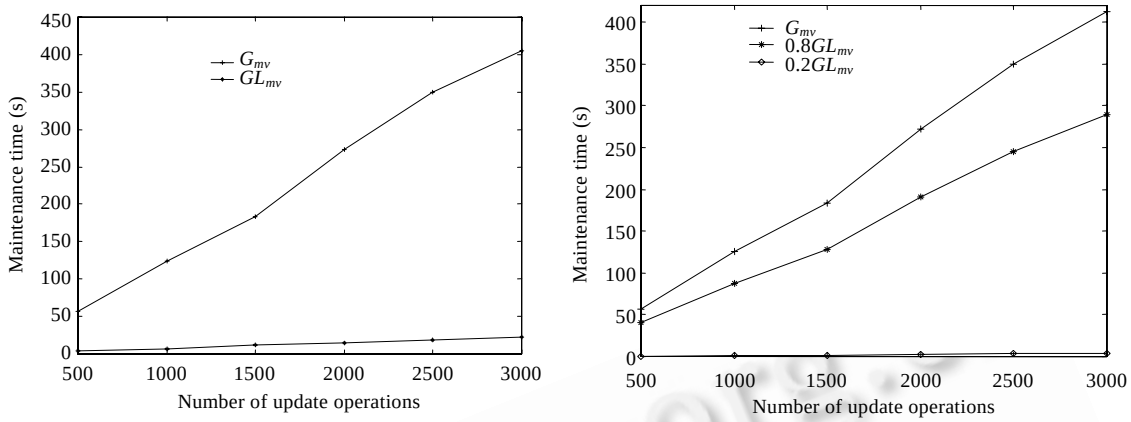
}

3 模拟实验

根据前面提到的 SPDMS 模型,我们开发了一个模拟系统,在其上做了大量的实验来比较集中式 G_{mv} 的维护效率(Mork 的维护策略)和分散式 G_{mv} 的维护效率(我们的维护策略).在我们的实验中一共有 60 个模拟节点,它们分为 5 个局部 PDMS,即 $G_{mv} = \cup_{i=1}^{5} GL_{mv_i}$;该 G_{mv} 有 12 个属性,并且共有大约 2400K 元组.实验采用的 CPU 为 PIV 1.7GHZ,内存 512MB,操作系统 Windows Server 2000,模拟程序是用 Java 编写的.

我们进行了两组实验,在第 1 组实验中,G_{mv} 均匀地分布在各个 GL_{mv} 中,实验结果如图 2(a)所示,其中,G_{mv} 和 GL_{mv} 分别表示 Mork 的维护策略和我们的维护策略所需维护时间.由该图可知,G_{mv} 明显高于 GL_{mv},这是因为每个 GL_{mv} 的数据量是 G_{mv} 的 1/5,并且更新次数是 G_{mv} 的 1/5.当系统维护各个 GL_{mv} 时,由于数据基本上在内存中,仅需要少量的 I/O 操作,因此维护时间较短.而对于维护集中式 G_{mv},由于数据量较大,需要大量的 I/O 操作,因此维护时间较长.

第 2 组实验采用 80-20 分布,即将 G_{mv}80%的数据分布在 20%的 GL_{mv} 中,并且 80%的更新操作也发生在这 20%的 GL_{mv} 上.在我们的实验中,一个 GL_{mv} 存储了 G_{mv} 中 80%的数据量,并且承担了 80%的更新操作.实验结果如图 2(b)所示,其中,0.8GL_{mv} 表示承担了 80%更新操作的 GL_{mv} 所需维护时间;而 0.2GL_{mv} 表示共同承担了 20%更新操作的各个 GL_{mv} 所需维护时间.从图中可知:0.8GL_{mv} 与 G_{mv} 相比已经有明显的下降,这是因为其数据量和更新操作的次数都比集中式 G_{mv} 要少;而各个 0.2GL_{mv} 基本上是可以忽略的,这是由于数据量小,更新操作都在内存中进行.



(a) Even distribution (a) 均匀分布 (b) 80-20 distribution (b) 80-20 分布

Fig.2 View maintenance time of G_{mv} vs. GL_{mv}

图 2 G_{mv} 和 GL_{mv} 的维护时间对比

由这两组实验可知:如果载荷均匀分布,可以充分利用 P2P 环境中的并行性,极大地提高数据一致性维护的效率;即使在采用 80-20 分布的情况下,也能有效利用 P2P 环境的并行性,较大地提高数据一致性维护的效率.在纯 P2P 架构下,Peer 有完全的自主性,当存储物化视图的 Peer 离开后,视图的一致性很难得到保障;而在超级 Peer 架构下,由于超级 Peer 具有相对稳定性,因此,视图的一致性能够满足需要.另一方面,当物化视图存储在一个节点时,其维护代价太大,并且网络传输代价也很大;如果把全局视图维护转化为各个相关局部视图的维护,则视图的维护代价和网络传输代价都能得到缓和,视图维护不会成为性能瓶颈.因此,集中式物化视图维护没有利用 PDMS 的特点,而分散式物化视图维护则充分利用了 PDMS 的优点,实验也证实了该方法的可行性.

4 相关的研究工作

Mork[7]提出了两类对象:updategram 和 booster,它们被当成系统中的第 1 类公民,然后,他导出一组规则来管理 updategram 和 booster.然而,在 PDMS 中,随着相关数据源数据量的增加和系统的增长,如果全局视图存储在一个节点中,则将很难满足 PDMS 扩展性的需要.根据应用需要,本文提出了一种新的视图维护策略.

文献[12]讲述了如何高效地计算更新,而在数据集成和数据仓库环境中,主要的研究工作集中在如何保证视图的自维护上,本文主要研究当数据源发生数据更新时,需要传输哪些数据给物化视图所在节点,以便进行视图维护.文献[13]主要研究了如何对视图定义稍微进行修改,以保证其是可自维护的.文献[14]的研究工作主要集中在为了加速数据仓库环境中视图的自维护,需要维护哪些辅助的视图上.文献[15]提出了一种版本控制集合刷新算法(version-control set refreshing algorithm,简称 VSRA),它采用增量维护、版本控制和批处理机制保证数据仓库的联机维护和数据一致性.VSRA 不仅减少了数据库和数据仓库之间的通信流量,而且提高了实体化视图的刷新效率.保证数据一致性可以采用触发器,文献[16]描述了一种算法来自动地构造触发器以维护物化视图,其基本假设是,当一个关系更新时,它知道哪些视图需要维护.在我们的视图维护策略中,当一个数据源发生更新时,其更新的数据以及与维护物化视图有关的其他数据将主动发送给视图所在的节点.

采用数据复制来确保数据的可获得性,其面临的问题是如何保证多个版本的一致性.解决该问题的策略包括:主版本策略[17],也就是当客户端对主版本进行写的时候,仅仅需要与其从节点进行交互;惰性策略[18],也就是操作松散地排序,但不能保证严格的一致性.文献[19]中提出的策略是任何一个版本都能够更新,其冲突解决方案是采用 anti-entropy 策略.在文献[20]中详细地描述了该方法的性能.在我们的视图维护策略中,各个 GL_{mv} 只在本局部 PDMS 中存在,因此,仅存在视图和数据源这两个版本之间的一致性维护问题.

5 总结和下一步研究计划

近年来,P2P 计算是一个研究热点.在 P2P 系统中,大量的节点能够联合在一起,共享它们的资源、信息和服务.本文主要研究了 PDMS 中视图的维护策略.如果视图沿着一条语义链路传播,那么,它通过重构能够从多个数据源检索到数据,因此,我们在 PDMS 中提出了全局视图、局部视图和 Peer 视图.如果一个 Peer 更新了它的数据,那么,根据扩展了的规则系统,SPDMS 采用一种基于推的传播算法来维护视图的数据一致性;由于更新操作限制在各个局部 PDMS 中,对全局视图的维护就转化为对各个相关局部视图的维护.

将来的研究计划包括如下两个方向:一方面,我们将研究如何把 L_{mv} 和 GL_{mv} 存储在与其数据源不同的局部 PDMS 中;另一方面,我们将研究面向知识的 Peer 数据集成系统.

References:

[1] Halevy AY, Ives ZG, Madhavan J, Mork P, Suciu D, Tatarinov I. The piazza peer data management system. IEEE Trans. on Knowledge and Data Engineering, 2004,16(7):787−798.

[2] Kementsietsidis A, Arenas M, Miller RJ. Mapping data in peer-to-peer systems: Semantic and algorithmic issues. In: Halevy AY, Ives ZG, Doan AH, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data. 2003. 325−336.

[3] Ng WS, Ooi BC, Tan KL, Zhou AY. PeerDB: A P2P-based system for distributed data sharing. In: Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc. of the 19th Int'l Conf. on Data Engineering. IEEE Computer Society, 2003. 633−644.

[4] Aberer K, Cudre-Mauroux P, Hauswirth M. The chatty Web: Emergent semantics through gossiping. In: Proc. of the 12th Int'l World Wide Web Conf. 2003. 197−206.

[5] Shah S, Ramamritham K, Shenoy P. Resilient and coherence preserving dissemination of dynamic data using cooperating peers. IEEE Trans. on Knowledge and Data Engineering, 2004,16(7):799−812.

[6] Gao S, Ng WS, Qian W, Zhou A. CC-Buddy: An adaptive framework for maintaining cache coherency using peers. In: Proc. of the 13th Int'l World Wide Web Conf. 2004. 330−331.

[7] Mork P, Gribble SD, Halevy AY. Managing change in large-scale data sharing systems. Technical Report, ftp://ftp.cs.washington.edu/tr/2004/04/UW-CSE-04-04-01.pdf

[8] He YJ, Shu YF, Wang S, Du XY. Efficient top-*k* query processing in pure peer-to-peer network. Journal of Software, 2005,16(4): 540−552 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/540.htm

[9] Gao Y, Cheng TY, Wang S. Certificates storage strategy and search algorithm based on Hilbert curve. Journal of Software, 2006, 17(2):305−314 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/305.htm

[10] Nejdl W, Siberski W, Sintek M. Design issues and challenges for RDF- and schema-based peer-to-peer systems. SIGMOD Record, 2003,32(3):41−46.

[11] Gupta A, Jagadish HV, Mumick IS. Data integration using self-maintainable views. In: Proc. of the 5th Int'l Conf. on Extending Database Technology: Advances in Database Technology. 1996. 140−144.

[12] Abiteboul S, McHugh J, Rys M, Vassalos V, Wiener J. Incremental maintenance for materialized views over semistructured data. In: Gupta A, Shmueli O, Widom J, eds. Proc. of the 24th Int'l Conf. on Very Large Data Bases. 1998. 38−49.

[13] Quass D, Gupta A, Mumick IS, Widom J. Making views self-maintainable for data warehousing. In: Proc. of the 4th Int'l Conf. on Parallel and Distributed Information Systems. IEEE Computer Society, 1996. 158−169.

[14] Labio W, Quass D, Adelberg B. Physical database design for data warehouses. In: Gray WA, Larson P, eds. Proc. of the 13th Int'l Conf. on Data Engineering. IEEE Computer Society, 1997. 277−288.

[15] Li ZM, Li L, Zhou XM, Wu JP. A set refreshing algorithm for data warehouse on-line maintenance, Journal of Software, 2000, 11(12):1594−1597 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/11/1594.htm

[16] Ceri S, Widom J. Deriving production rules for incremental view maintenance. In: Lohman G, Sernadas A, Camps R, eds. Proc. of the 17th Int'l Conf. on Very Large Data Bases. 1991. 577−589.

[17] Pacitti E, Simon E. Update propagation strategies to improve freshness in lazy master replicated databases. VLDB Journal, 2000, 8(3-4):305−318.

[18] Ladin R, Liskov B, Shrira L, Ghemawat S. Providing high availability using lazy replication. ACM Trans. on Computer Systems,

1992,10(4):360−391.

[19] Terry DB, Theimer MM, Petersen K, Demers AJ, Spreitzer MJ, Hauser CH. Managing update conflicts in Bayou, a weakly connected replicated storage system. In: Proc. of the 15th ACM Symp. on Operating System Principles. 1995. 172−183.

[20] Gray J, Helland P, O'Neil P, Shasha D. The dangers of replication and a solution. In: Jagadish HV, Mumick IS, eds. Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data. 1996. 173−182.

附中文参考文献:

[8] 何颖捷,王珊,杜小勇.纯 Peer to Peer 环境下有效的 Top-*k* 查询.软件学报,2005,16(4):540−552. http://www.jos.org.cn/1000-9825/16/540.htm

[9] 高迎,程涛远,王珊.基于 HILBERT 曲线的许可证存储策略及相关查找算法.软件学报,2006,17(2):305−314. http://www.jos.org.cn/1000-9825/17/305.htm

[15] 李子木,李磊,周兴铭,吴建平.一种数据仓库联机维护的集合刷新算法.软件学报,2000,11(12):1954−1957. http://www.jos.org.cn/1000-9825/11/1594.htm

覃飙(1972－),男,湖北荆州人,博士,讲师,主要研究领域为 Peer 数据管理系统,数据库系统.

杜小勇(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为高性能数据库,智能信息检索,知识工程.

王珊(1944－),女,教授,博士生导师,CCF 高级会员,主要研究领域为高性能数据库,数据仓库,知识工程.