

网格数据库物化查询缓存机制研究*

张延松^{1,2}, 张宇^{1,2}, 薛永生²

(1. 哈尔滨金融高等专科学校 计算机系, 黑龙江 哈尔滨 150036; 2. 厦门大学 计算机科学系, 福建 厦门 361005)

摘要: 提出了基于 XML Database 的网格数据库物化查询缓存机制, 提高用户查询的速度, 均衡网格负载。定义了网格数据库服务质量与数据质量的标准, 提出了物化查询选择算法 MQS, 为用户提供更好的数据服务。

关键词: 网格数据库; XML Database; 物化查询缓存机制; MQS

中图法分类号: TP311 文献标识码: A 文章编号: 1001-3695(2006)07-0242-04

Research on Grid Database Materialized Query Caching Policy

ZHANG Yan-Song^{1,2}, ZHANG Yu^{1,2}, XUE Yong-sheng²

(1. Dept. of Computer Science, Harbin Financial College, Harbin Heilongjiang 150036, China; 2. Dept. of Computer Science, Xiamen University, Xiamen Fujian 361005, China)

Abstract: In this paper, grid database materialized query caching policy based on XML Database is presented, which will improve the response time of queries and balance the load of the grid. QoS and QoD of grid database are defined to provide parameters. The MQS algorithm acts as cached materialized query selector to provide better data service.

Key words: Grid Database; XML Database; Materialized Query Caching Policy; MQS

网格计算是分布式计算的一个新的阶段, 网格需要解决在动态的虚拟组织中控制和协调对资源的共享^[1]。开放网格服务结构(Open Grid Services Architecture, OGSA)是 Global Grid Forum⁴的重要标准建议, 是以服务为中心的“服务结构”, 被称为下一代的网格结构。OGSA 网格中的各种资源都是服务, 网格服务就是要实现服务的共享。网格服务是一种 Web Service, 网格是可扩展的网格服务的集合, 即网格 = { 网格服务 }。数据库是 Internet 上最重要的数据资源载体, 在网格中集成对 Internet 上广域分布的数据库资源的数据访问服务是网格应用的重要组成部分。DAIS 工作组(Database Access and Integration Service Working Group)的主要研究内容是通过制定网格数据库服务的标准, 构造在网格环境中对数据存取和集成的中间件, 将数据库运用到网格中。本文研究在网格数据库中通过数据访问的缓存机制来提高网格数据库数据服务的效率。

关系数据库是应用得最为广泛的数据库管理系统, 数据以关系表的形式存在, 而且存在着由数据库结构不同而产生的差异性。Internet 上的数据交换标准是 XML, OGSA-DAI 采用服务的方式实现网格数据库的访问与集成, 这样就存在了一条“关系数据库 → GDS → Client”路径的 XML 数据流, 提高网格数据库数据服务的共享性和复用性可以充分发挥网格资源的优势, 提高数据访问的效率。

1 相关工作

随着网格技术的发展, 网格数据库的研究也成为了当前的热点研究问题^[2]。以 OGSA-DAI 为代表的网格访问中间件提

供了通用数据访问服务, 以 XML 格式为用户提供对网格中物理数据库的访问^[3~5]。

传统的数据库通常以 JDBC 或 ODBC 向用户应用程序提供数据接口, 其结果集面向单一的应用程序, 不能被其他应用程序所共享。OGSA-DAI 采用 Web Service 包装数据库的访问接口, 从而将对数据库的查询与可被第三方共享的 XML 结果集结合起来。通过将用户的查询结果物化为 XML 文件并缓存的策略, 可以将面向单一用户的查询变为网格中可供其他用户共享的数据资源。

动态缓存技术用于对数据访问的优化。当前很多研究集中在静、动态网页的缓存策略研究中, 文献[6]通过在应用服务器与数据库之间增加同步缓存的方法来实现一种动态调整 WebView 的物化选择方法, 文献[7]在此基础上又做了进一步的完善工作。将数据库查询物化为 XML 格式的文件并通过缓存机制进行共享可以提高系统的数据服务质量。文献[10]基于用户发布请求的相似性, 提出了频繁发布请求并且缓存中间结果的办法来提高关系数据的 XML 发布效率。在网格环境下, 物化查询缓存机制也具有其自身的特点。

2 基于 XML Database 的网格数据库 MQ 缓存机制

2.1 数据库的物化查询

数据库是一个关系集, 用户的查询产生一个结果集, 在被 Web Service 包装后形成 XML 结果集。XML 结果集可以看作是查询的物化结果, 可以以文件的形式被其他具有相同查询要求的用户所共享。我们用 $R_i = \{ R_{i1}, R_{i2}, R_{i3}, \dots, R_{in} \}$ 表示关系的集合, 用 $v_i = \{ v_{i1}, v_{i2}, v_{i3}, \dots, v_{in} \}$ 表示查询结果集, 用 $Xv_i = \{ Xv_{i1}, Xv_{i2}, Xv_{i3}, \dots, Xv_{in} \}$ 表示 XML 结果集。

定义 1 查询操作 Q , 存在 $Q(R_i) = v_i$, 其中 v_i 是对关系 R_i 进行查询操作所对应的查询结果。

收稿日期: 2005-06-26; 修返日期: 2005-08-18
基金项目: 福建省自然科学基金资助项目(A0310008); 福建省高新技术研究开放计划重点资助项目(2003H043)

定义 2 物化查询(Materialized Query, MQ), 存在 $MQ(v_i) = Xv_i$, 其中 Xv_i 是通过 Web Service 包装后形成 XML 结果集。

定义 3 查询相容度。

$$f(Q_i, Q_j) = \begin{cases} 1, & \text{如果 } Q_i(R_i) \subseteq Q_j(R_i) \\ 0, & \text{如果 } Q_i(R_i) \not\subseteq Q_j(R_i) \end{cases}$$

查询相容度用来确定两个用户查询的结果集之间是否存在包含关系。

对数据库查询结果的物化是在网格中实现数据库资源集成与共享的重要条件, 同时对物化后的查询结果的共享与一致性维护也是实现网格数据库数据缓存的重要内容。

2.2 网格数据库物化查询缓存

在 OGSA-DAIS 项目中, 网格服务包括 GDS(Grid Data Service), GDSF(Grid Data Service Factory) 和 DAISGR(DAI Service Group Registry) 三类, 可以分别将 DAISGR 看作是 GDSF 的容器, GDSF 看作是 GDS 的容器, 提供服务资源的注册、发布、发现功能。如图 1 所示, 用户所提出的网格数据服务请求通过 GDSF 创建实例化的 GDS 来完成与物理数据库的数据访问操作, 并通过 XML 数据流在用户与物理数据库之间交互, 用户的数据访问请求最终转换为通过 GDS 对网格中物理数据库的独立访问。

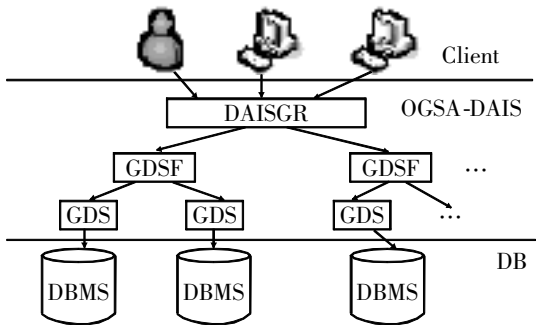


图 1 网络服务层次结构

当网格中用户的访问量激增时, 物理数据库的负载增大, 会造成数据响应能力下降, 影响数据服务质量; 而用户的数据访问具有一定的相似性, 在大部分数据库的更新周期内的数据查询结果具有可共享性, 通过物化用户查询结果并引入缓存机制可以提高网格数据库的整体数据访问性能。

如图 2 所示, 在 OGSA-DAIS 结构中引入 GDSF 主控的 XML Database 物化查询缓存机制。通过对用户查询的物化, 将用户查询的结果按一定的缓存策略缓存于 XML 数据库中, 在 GDSF 接收用户新的查询请求时, 通过对查询的比较与对缓存中物化查询的数据质量 QoD(Quality of Data) 分析来决策向用户返回缓存中的物化查询还是通过实例化的 GDS 向物理数据库提交新的查询请求。缓存中的物化查询按不同的物化策略进行更新。图 2 中的实线为用户查询的数据流向, 虚线为物化查询缓存的数据流向。数据流 为用户通过实例化的 GDS 获得数据服务; 数据流 为物化查询缓存流向; 数据流 为 GDSF 通过缓存的物化查询来满足用户的数据服务要求。

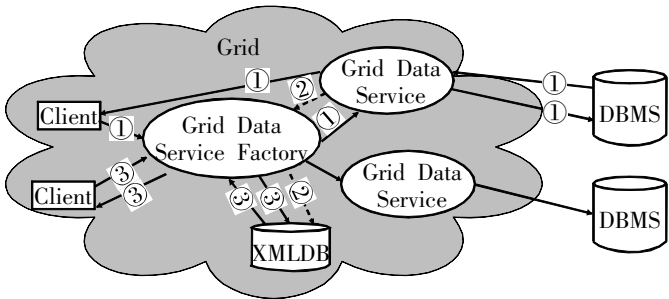


图 2 基于 XML Database 的网格数据库缓存结构

物化查询结果是以 XML 形式存在的, 物化查询的缓存机制要具有强大的 XML 处理能力。Native XML DBMS(NXD) 是以 XML 格式存储数据的 XML 数据库, 在内部保持 XML 文档的树型结构, 不需要进行数据转换, 具有较高的数据处理效率, 而且通过查询语言 XPath 和 XQuery 支持对 XML 文档的查询操作^[8,9]。以 NXD 作为物化查询缓存具有良好的 XML 操作性能。物化查询缓存管理包括物化查询结果集匹配、物化查询结果集抽取、物化查询结果集的增加、删除与更新操作。

2.3 用户查询物化策略

根据用户查询的特点, 将用户查询的物化分成三种情况:

(1) 虚拟物化。当用户的数据服务请求通过 GDS 向数据库提交时, 查询结果被包装成 XML 格式直接传递给请求用户, 查询结果不缓存, 不具有共享能力。

(2) 条件物化。根据系统物化算法对用户查询进行分析, 当满足物化条件时将用户查询结果物化为 XML 数据并缓存, 同时登记物化查询目录, 在数据满足条件时为用户提供查询缓存。当源数据更新时, 根据物化查询的访问情况进行更新或停止过期的物化查询的访问并删除该物化查询。在系统接收用户新的查询时, 通过服务质量 QoS 指标来比较网格中数据库的访问负载与物化查询缓存的访问负载, 从而选择更好的数据服务。

(3) 完全物化。在用户第一次查询时, GDS 将返回给用户的 XML 数据同时返回到物化查询缓存, 并保持与源数据的同步更新(可采用数据库服务器端驱动的完全更新或增量更新策略), 相当于建立了一个数据库的“XML 镜像”。

在三种物化策略中, 虚拟物化不具有被第三方共享的能力, 而且也不能均衡系统负载, 容易造成单点负载过重的情况。条件物化可以为第三方提供共享查询结果, 提高了系统的信息共享能力, 并有利于调整系统负载, 避免出现单点负载过重的情况。在接收用户查询请求时, 通过服务质量 QoS 指标来均衡系统负载, 提高系统的响应能力。完全物化策略需要较大的缓存代价与数据更新代价。大量访问率较低的物化查询会消耗系统较多的缓存资源并增加了系统的维护成本。采用条件物化策略, 既可以保证高访问率的用户查询的数据响应能力又可以将维护与更新成本保持在可接受的范围内。

2.4 物化查询维护与选择策略

条件物化查询策略包括物化查询维护与物化查询选择两部分功能。

2.4.1 物化查询维护

物化查询维护包括物化查询的匹配、抽取、增加、删除和更新功能。

定义 4 物化查询目录(Materialized Query Catalog, MQ-CA)。物化查询目录记录物化查询元数据信息、源数据信息、最新更新时间与访问次数等信息。记作

$$MQCA = Y_{i=1}^n \{ Xv_i, Source_i, Updatetime_Xv_i, Updatetime_S_i, access_count_i, \dots \}$$

算法 1 物化查询匹配 MQ_Match

输入: 用户查询 $Q(R_i)$
输出: 物化查询结果集 Xv_i
int flag: =0;
for 物化查询目录中每一记录
if $MQ(Q(R_i))$ 与物化查询缓存中的物化查询相同 and 物化查询的更新时间 > 源数据的更新时间
then

```
    return Xvi;
    flag:= 1;
    /* 当前缓存的物化查询 Xvi 与用户查询匹配且为有效物化查询
    时, 返回物化查询结果集 Xvi* /
    end if
end for
if ( flag =0) then
    return null;
end if
/* 当缓存中没有与查询匹配的物化查询或物化查询过期时, 返回
空集* /
当用户查询结果集为物化查询的子集时, 通过 XML Database 的查
询语言抽取出满足用户需要的物化查询结果集。
```

算法 2 物化查询抽取 MQ_Extract

```
输入: 用户查询 Q( Ri )
输出: 物化查询结果集 Xvi
if ( MQ_Match( Q ( Ri ) ) ≠ null) and ( f( Q, Q ) = 1)
    then
return XQuery( MQ_Match( Q ( Ri ) ) );
/* 若用户查询为某一物化查询子集, 则通过 XML Database 所提
供的 XQuery 查询语言在该物化查询中抽取当前查询所需要的结果子
集, 并返回给用户* /
end if
```

算法 3 物化查询增加 MQ_Add

```
输入: 用户查询 Q( Ri )
输出: null
Xv = MQ( Q( Ri ) );
//生成物化查询
MQCA_Add( Xv, time( ) );
/* 在物化查询目录 MQCA 中登记该物化查询元数据信息, 记录
物化查询更新时间和当前访问次数* /
XMLB_CREATE( Xv );
//在 XML Database 中增加该物化查询文件
```

算法 4 物化查询删除 MQ_Delete

```
输入: 物化查询 Xv
输出: null
MQCA_Delete( Xv ); /* 在物化查询目录 MQCA 中删除该物化查
询元数据信息* /
XMLB_DELETE( Xv );
//在 XML Database 中删除该物化查询文件
```

当网格中的数据库服务器进行数据更新时, 会引起部分相关的物化查询数据失效。我们令 τ 为物化查询的访问频率阈值, 物化查询在最近的一个更新周期内的访问频率大于 τ 时, 更新物化查询和物化查询目录中对应的物化查询元数据; 当最近的一个更新周期内的访问频率小于 τ 时, 删除此物化查询。

算法 5 物化查询更新 MQ_Update

```
输入: 物化查询 Xvi
输出: null
if Sourcei 产生数据更新操作 then
    UpdatetimeSi: = 数据源更新时间;
    if  $\frac{access\_count_i}{now( ) - Updatetime\_Xv_i} > \tau$  then
        更新 Xvi;
        UpdatetimeXvi: = now( );
        access_counti: = 0; //重置访问计数器
    else
        MQCA_Delete( Xvi );
    end if
end if
```

2. 4. 2 物化查询选择

物化查询缓存机制是为了提高数据的共享性, 提高系统的综合性能。用户查询质量可以从服务质量(Quality of Service, QoS) 和数据质量(Quality of Data, QoD) 两个方面进行衡量。

服务质量反映网格的系统负载情况、网络带宽、网络延迟等信息, 可以由网格信息服务—元计算目录服务(Metacomputing Directory Service, MDS) 所提供的组件—网格资源服务(Grid Resource Information Service, GRIS) 来提供。QoS 反映网格当前的性能状况, 如果网格当前的性能指标高, 则应尽量通过 GDS 从源数据库中获取用户查询的数据服务, 以提高数据的可靠性; 如果网格当前的性能指标较低, 则应考虑通过物化查询缓存机制来均衡网格负载, 以缓存的物化查询来满足用户的查询需求。我们规定 QoS 与网格资源访问性能成正比。

数据质量反映用户查询的数据可靠性。我们记作: QoD [0, 1], 我们可以通过物化查询目录 MQCA 中所对应的源数据更新时间与物化查询更新时间来确定 QoD:

```
if ( UpdatetimeXvi ≥ UpdatetimeSi) then
    QoD( Xvi, vi ) : = 1;
if ( UpdatetimeXvi < UpdatetimeSi) then
    QoD( Xvi, vi ) : = 0;
```

由前面的定义可知, 用户查询是一系列相关查询的集合, 服务器更新时所影响的查询结果也不完全相同, 对于特定的物化查询而言, 其数据质量也应该是相关查询数据质量的综合值。

定义 5 物化查询的质量系数:

$$f(Xv_{ij}, v_{ij}) = QoD(Xv_{ij}, v_{ij}) , f(Xv_{ij}, v_{ij}) \in \{ 0, 1 \}$$

质量系数用于确定用户查询中相关子查询的数据质量, 并根据用户查询集的组成确定物化查询的整体数据质量。

定义 6 用户查询整体数据质量。

$$QoD_{total}(Xv_i, v_i) = \sum_{j=1}^n a_j \times f(Xv_{ij}, v_{ij})$$

其中, a_j 是每个子查询的权重, $\sum_{j=1}^n a_j = 1$, 默认 $a_j = \frac{1}{n}$, 如需要可根据子查询的重要性设置相应的权重值。子查询权重取默认值时, $QoD_{total}(Xv_i, v_i) = \frac{1}{n} \sum_{j=1}^n f(Xv_{ij}, v_{ij})$ 。

在为 用户查询进行物化查询选择时, 我们遵循如下原则:

(1) Level_{up} 和 Level_{down} 为服务质量的上限值和下限值, Level_{accept} 为数据质量的阈值, 其值可以根据经验设置或根据网格系统的统计数据来确定。

(2) 当 QoS > Level_{up} 时, 网格系统的性能较好, 无论有无物化查询均采用实例化 GDS 数据访问方式, 为用户提供对物理数据库的直接访问, 保证数据可靠性。

(3) 当 QoS < Level_{down} 时, 网格系统的性能较差, 甚至可能无法访问源数据库, 如果此时 $1 > QoD_{total} > Level_{accept}$, 则表明虽然物化查询并不能完全保证与源数据库一致, 但从整体上可以提供较可靠的数据, 此时可以由物化查询缓存提供“保守”的物化查询供用户使用。

(4) 当 Level_{down} < QoS < Level_{up} 时, 系统可以根据物化查询选择算法来为用户提供数据服务。

在物化查询选择算法中, 首先要解决用户查询的物化问题。物化查询的目的是通过 XML Database 中缓存的共享程度较高的 XML 查询结果来提高系统的数据响应能力。根据有限资源合理使用的原则, 物化的查询应该具有较高的数据访问频率和较低的更新频率。根据前面所述的物化查询选择原则, 应该优先物化 QoS 较低的数据源所产生的查询。综合上述因

素, 我们用 $\text{evaluate}(v_i)$ 来评估用户查询的物化权重。

定义 7 查询访问率。记作 $\text{access_rate}(v_i)$, 用于计算查询结果 v_i 的访问频率。

定义 8 查询更新率。记作 $\text{update_rate}(v_i)$, 用于计算查询结果 v_i 的更新频率。

定义 9 用户查询的物化权重。

$$\text{evaluate}(v_i) = \frac{\text{access_rate}(v_i)}{\text{QoS} \times \text{update_rate}(v_i) + 0.01}$$

即, 物化权重与访问频率成正比, 与 QoS 和更新频率成反比 (分母中加上 0.01 是为了防止出现 0 值)。

2. 4. 3 物化查询选择算法 MQS

在物化查询选择算法中, 我们需要考查网格服务质量与数据质量, 根据对用户查询物化权重的评价来判断用户查询是否满足物化条件, 从而确定执行查询物化、从物化查询缓存中向用户返回物化查询数据还是通过实例化 GDS 实现对物理数据库的访问。

算法 6 MQS(Materialized Query Select)

输入: 用户查询 v_i

输出: 虚拟物化查询或条件物化查询 Xv_i

$\text{access_rate}(v_i)$; // 计算查询访问频率

if($\text{QoS}(v_i) < \text{Level}_{\text{down}}$) then

if($\text{QoD}_{\text{total}} > \text{Level}_{\text{accept}}$) then

return $\text{MQ_R}(v_i)$;

else

return null;

end if

end if

if($\text{QoS}(v_i) > \text{Level}_{\text{up}}$) then

$\text{GDS_execute}(v_i)$;

end if

if($\text{Level}_{\text{down}} < \text{QoS}(v_i)$) and ($\text{QoS}(v_i) < \text{Level}_{\text{up}}$) then

if ($\text{MQ_Match}(v_i) \neq \text{null}$) then

return $\text{MQ_Match}(v_i)$;

/* 如果物化查询缓存中存在与用户查询相匹配的物化查询, 则从物化查询缓存中向用户返回物化查询结果* /

else

if($\text{evaluate}(v_i) > \lambda$) then

$\text{GDS_execute}(v_i)$;

$\text{MQ_Add}(v_i)$;

/* 当用户查询的物化权重大于阈值 λ 时, 通过实例化的 GDS 响应用户的查询请求, 同时将该用户查询物化, 并加入到 XMLDatabase 中* /

end if

end if

if ($\text{MQ_Extract}(v_i) \neq \text{null}$) then

return $\text{MQ_Extract}(v_i)$;

/* 如果用户查询是某已缓存的物化查询的子集, 则通过物化查询抽取操作返回满足用户查询需求的 XML 子文档* /

end if

end if

3 性能分析

根据以上论述, 我们对基于 XML Database 的网格数据库物化查询缓存机制作如下总结:

(1) 基于 XML Database 的网格数据库物化查询缓存机制是 XML Database 技术、网格技术、数据库技术相结合的产物。基于 XML 的网格数据服务是实现用户查询共享的基础。

(2) 以 XML Database 为基础的物化查询的缓存管理比文件方式的缓存管理能提供更高的效率和更强大的管理能力。

(3) 服务质量的上限值 Level_{up} 可以作为更合理的物化查询决策依据。在网格性能较好时, 系统优先选择网格所提供的数据服务, 为用户提供最为可靠的数据服务; 当网格不能为用户提供较好的性能时, 系统通过物化查询的缓存机制来提高网格数据访问性能, 平衡系统负载。

(4) 服务质量的下限值 $\text{Level}_{\text{down}}$ 与数据质量阈值 $\text{Level}_{\text{accept}}$ 作为网格出现性能瓶颈而物化查询与源数据不完全一致时为用户提供数据服务的决策依据。通过综合数据质量 $\text{QoD}_{\text{total}}$ 的评估来确定当前物化查询的综合质量系数, 当大于数据质量阈值 $\text{Level}_{\text{accept}}$ 时为用户提供可参考的“保守”的物化查询结果。

(5) 与无查询缓存机制的网格数据服务相比, 物化查询缓存机制不仅能够通过缓存机制提高网格数据服务的质量、均衡网格负载, 而且通过缓存机制可以有效地解决单点失效问题, 缓存在 XML Database 中的物化查询也可以供进一步的数据集成使用。

(6) 从物化查询生成的代价上看, 当 GDS 响应用户的数据查询而产生 XML 数据流时, 只需要并行将 XML 数据流向物化查询缓存 XML Database, 不需付出额外的物化查询生成代价。

4 结论

网格技术为数据库的发展提供了更广阔的空间, 结合了网格技术与数据库技术的网格数据库系统在解决因特网环境下的数据访问与集成方面有着良好的发展前景。基于 XML Database 的网格数据库物化查询缓存机制用于解决网格环境下的用户查询共享问题, 通过物化查询选择策略来动态调整网格数据库系统的数据服务质量, 并针对不同的网格访问性能状况采取不同的访问策略, 最大限度地满足用户的数据访问需求。

参考文献:

[1] 都志辉, 陈渝, 刘鹏. 网格计算[M]. 北京: 清华大学出版社, 2002.

[2] 王珊, 张坤龙. 网格环境下的数据库系统[EB/OL]. <http://www.chinagrid.net/dvnews/show.aspx?id=301&cid=61>, 2004.

[3] A Anjomshooa, *et al.* The Design and Implementation of Grid Database Services in OGSA-DAI [EB/OL]. <http://www.ogsadai.org.uk/docs/OtherDocs/156.pdf>, 2003.

[4] M Antonioletti, *et al.* Grid Data Service Specification[EB/OL]. <http://ppewww.ph.gla.ac.uk/preprints/2002/14/2002-14.pdf>, 2003.

[5] V Raman, I Narang, *et al.* Services for Data Access and Data Processing on Grids[EB/OL]. http://www.cs.man.ac.uk/grid-db/papers/DAIS_DataServices.pdf, 2003.

[6] A Labrinidis, N Roussopoulos. Balance Performance and Data Freshness in Web Database Servers[C]. The 29th VLDB Conf. , 2003.

[7] 张欣, 陈峰, 乐嘉锦. WebView 在线物化选择方案的研究[J]. 计算机研究与发展, 2004, 41(10): 1629-1633.

[8] 庄子明. 基于 XML 的数据库技术及应用[J]. 计算机工程, 2002, 28(1): 119-121.

[9] 李由, 黄凯歌, 汤大权. XML 的数据库存储技术研究[J]. 计算机应用研究, 2002, 19(4): 60-62.

[10] 邱洋, 岳昆, 等. 利用缓存优化关系数据的 XML 发布[J]. 计算机研究与发展, 2004, 41(10): 1802-1808.

作者简介:

张延松, 讲师, 硕士, 主要研究方向为数据库技术、数据仓库、数据网格; 张宇, 讲师, 硕士, 主要研究方向为数据库技术、数据仓库、数据网格、电子商务; 薛永生, 教授, 主要研究方向为数据库技术、数据仓库、数据挖掘、网络技术等。