文章编号:1001 - 9081(2010)01 - 0005 - 05

基于对象代理数据库的微生物信息服务系统

彭智勇[1], 黄泽谦[2], 刘　俊[1], 李　越[1], 徐　波[1]
(1. 武汉大学 计算机学院,武汉 430072; 2. 武汉大学 软件工程国家重点实验室,武汉 430072)
(peng@whu.edu.cn)

摘　要:提出了一种采用对象代理数据库实现微生物数据管理的新方法。该方法将微生物菌种资源数据的共性描述信息定义为基本微生物对象,其代理对象可以用来表示微生物菌种资源分类的多样性,并定义不同菌种的特性性状信息,也可以用来定义微生物资源不同类别的扩展关联信息;通过对象更新迁移可以支持数据动态分类,利用对象视图机制可以支持模式演化,跨类查询机制则实现了高效的数据检索;实现了一个基于对象代理数据库的微生物信息服务系统。实验测试表明,该方法比传统方法更有效。

关键词:微生物信息服务系统;对象代理模型;对象代理数据库
中图分类号: TP311　　**文献标志码**:A

Microbiology information service system based on object deputy database

PENG Zhi-yong[1], HUANG Ze-qian[2], LIU Jun[1], LI Yue[1], XU Bo[1]
(1. *School of Computer, Wuhan University, Wuhan Hubei* 430072, *China*;
2. *State Key Laboratory of Software Engineering, Wuhan University, Wuhan Hubei* 430072, *China*)

Abstract: A new approach for microbiology information management based on object deputy database was proposed. It used the object deputy model for data modeling. The common descriptive information of microbes was defined as the basic object. Microbial diversity and correlated information of microbes were modeled through deputy objects. A microbiology information service system based on our object deputy database was implemented. Advanced functions of the database could facilitate information management. Update propagation supports dynamic classification of data, object views could be used for schema evolution, and cross-class query provides efficient data retrieval. The experimental result shows that this approach outperforms the traditional ones.

Key words: microbiology information service system; object deputy model; object deputy database

0　引言

微生物是一类重要的、物种丰富的生物资源和基因资源,微生物菌种资源保藏肩负着管理微生物资源和保护生物多样性的重任,也是开展微生物研究的基础[1]。目前我国保藏有10 万余株微生物菌种,分散在近 30 多个单位,现有的微生物菌种资源保藏与管理方法存在以下不足:1)信息化水平较低,各保藏单位信息交流不畅通;2)微生物菌种分散保藏,种类不全面,低水平重复;3)微生物信息检索方式落后,效率低下;4)新菌种资源信息收录不及时,难以实现高效共享。为解决上述问题,国家“十一五”规划提出建设资源丰富、面向社会开放的国家微生物菌种资源库和服务管理信息系统,实现微生物资源信息共享,进一步促进教学、科研与生产活动。

建设微生物信息系统的核心是建立微生物菌种资源数据库。国内外已存在各种不同的生物数据库,管理生物数据的方式主要有:平面文件、关系数据库、面向对象数据库与对象关系数据库[2]。平面文件通常用于存储基因序列,每一条数据记录对应一个文件,数据查询的结果也是文件形式[3]。采用平面文件管理生物数据存在的问题在于数据检索效率低,且检索结果不利于用户快速获取信息。关系数据库在生物学领域占据了重要的地位,许多生物数据库都是建立在关系数据库基础之上,然而大量的研究已经表明,关系模型并不适合管理复杂生物数据。由于难以建模数据间复杂的语义关系,采用关系数据库实现生物数据管理,数据库的模式通常较复杂、不直观,数据检索往往涉及到大量的关系连接操作,影响了数据查询的效率。与关系模型相比,面向对象模型提供丰富的语义且支持复杂数据类型定义,适合生物数据建模,因此部分生物数据库是基于面向对象数据库实现的。对象关系数据库在实现生物数据管理方面也具有相类似的优势,可以弥补传统关系数据库方式实现生物数据管理的不足。

虽然平面文件和传统的数据库能够用于建立微生物数据库,但是由于数据模型的限制,往往无法兼顾丰富语义建模、避免数据冗余、支持模式演化以及提供高效查询检索的需求。为此,本文提出一种基于对象代理数据库的微生物数据管理方法,并讨论了如何利用对象代理数据库来实现一个微生物信息服务系统。系统充分利用对象代理数据库在管理复杂生物数据方面的优势[4],可以方便地建模微生物数据间复杂的语义关系,并且避免数据冗余;可以有效地支持数据模式演化,确保系统具有良好的扩展性;可以利用跨类查询机制实现高效的数据检索。

收稿日期:2009 - 07 - 13;**修回日期**:2009 - 08 - 17。　**基金项目**:国家自然科技资源平台项目(2005DKA21208-11)。

作者简介:彭智勇(1963 -),男,湖北武汉人,教授,博士,CCF 会员,主要研究方向:复杂数据管理、Web 数据管理、可信数据管理;　黄泽谦(1983 -),男,广东普宁人,博士研究生,主要研究方向:数据库理论、科学工作流;　刘俊(1985 -),女,湖北随州人,硕士研究生,主要研究方向:生物数据管理;　李越(1987 -),男,湖北襄樊人,主要研究方向:生物数据管理;　徐波(1985 -),男,湖北随州人,硕士研究生,主要研究方向:跨媒体技术。

1 对象代理数据库

对象代理模型[5]是在传统的面向对象数据模型中通过引入代理对象和代理类的概念而提出的一个新的数据模型，其核心在于“代理”的概念。在面向对象数据模型中，属性和方法封装在一个对象中，子对象可以继承父对象的属性和方法。面向对象数据模型的封装性，无法实现对象的分割和组合，缺乏柔软性；传统的面向对象数据模型也没有提供灵活的对象视图机制。在对象代理模型中，代理对象可以选择性继承源对象的部分或全部属性、方法，同时可以根据需要增加扩展属性和扩展方法的定义；代理对象继承的属性、方法定义有切换操作，对它们的引用都会通过切换操作切换到对源对象上相应属性、方法的引用；代理对象的模式由代理类定义，对象代理代数用于创建不同的代理类；源对象和代理对象之间通过双向指针链接，更新迁移机制建立在双向指针之上，用于保证源对象和代理对象间的语义一致性。对象代理模型提供了丰富的语义，代理对象可以扮演源对象的各种角色，也可以用于实现对象视图。通过代理对象可以灵活地实现对象的分割和组合，提供了柔软性。

基于对象代理模型，我们实现了对象代理数据库管理系统，并提供对象代理数据库语言[6]用于数据定义和操作。在对象代理数据库中，每个对象和代理对象都有唯一的对象标识符；代理对象继承的属性称为“虚属性”，并不占有实际的物理存储，其值通过定义在其上的切换操作计算得到；对任何对象或代理对象的更新都会触发更新迁移，确保对象间的代理关系保持语义一致性。对象代理数据库语言采用类似标准SQL的风格，包括对象代理数据定义语言和对象代理数据操作语言。数据定义语言提供了类和代理类定义功能，一共有四种不同的代理类定义方式，包括：SELECT代理类、UNION代理类、JOIN代理类、GROUP代理类。代理类通过在普通类上声明一个代理规则创建，在代理类上还可以进一步创建代理类，在模式上形成一个代理层次网。代理对象由系统根据代理规则自动派生或消除，除了对代理对象的创建和删除操作由对象更新迁移自动完成外，在数据操作语言中，对代理对象的操作与一般对象上的操作无异。

对象代理数据库提供以下3个高级数据库功能。

1）灵活的对象视图。代理类可以用于定义对象视图。SELECT代理类用于实现对象的特化，UNION代理类用于实现对象的泛化，JOIN代理类用于实现对象的聚集，GROUP代理类用于实现对象的分组。

2）多重分类与动态分类。在对象代理数据库中，一个对象可以存在属于不同代理类的多个代理对象。由于代理类通过在源类上声明一个代理规则创建，只要源类中的对象满足代理规则，系统就会自动派生相应的代理对象。当添加、删除或更新一个对象时，更新迁移机制保证其代理对象也被相应地更新。

3）高效的跨类查询。在对象代理数据库中，由于类层次和对象层次都存在复杂的语义关系，而且对象与其代理对象间存在双向指针链接，因此可以方便地根据对象间的代理关系实现对象导航遍历。基于这一特殊机制，对象代理查询语言提供路径表达式用于跨类查询，跨类查询从功能上等价于传统的基于值的连接查询，但是基于指针的连接查询比基于值的连接查询效率更高，使用更方便。

2 微生物数据管理

微生物信息系统通常比较复杂，而且难以实现，主要是由于底层数据库的原因所造成的。微生物数据具有种类繁多、数据结构复杂、数据间存在复杂关联的特点，传统的数据库多是关系或面向对象或对象关系数据库，虽然能够用于建立微生物数据库，但是由于数据模型的限制，无法兼顾丰富语义建模、减少数据冗余、支持模式演化以及提供高效查询检索的需求。为此，我们提出基于对象代理数据库来创建微生物数据库，进一步实现微生物信息服务系统。

2.1 微生物数据特点

微生物学领域本身的复杂性决定了微生物数据具有复杂数据的特点。自然世界的微生物资源丰富多彩，微生物的生物多样性表现为物种多样性、分布广泛性和环境复杂性。微生物物种的多样性决定了微生物数据种类繁多；此外，微生物的研究涉及从宏观级别到分子级别，每种方法都有自己的分类，因此微生物的分类具有多样性。微生物资源数据库作为存储、管理微生物信息的工具，需要根据不同属种的微生物资源所具有的不同性状特征，设计不同的存储和管理方法。

微生物数据来源广泛，微生物资源数据库在集成不同来源数据时面临着结构异构和语义异构的问题，往往需要处理大量不同数据结构的数据；微生物分类的多样性决定了数据库数据结构具有复杂性的特点，因为分类数据本身就非常复杂；此外，微生物学及生物信息学的快速发展，促进了新微生物物种的发现以及微生物菌种新性状特性的鉴定，使得数据库的模式处于不断改善和丰富的状态，数据库模式的快速演化也是微生物数据结构复杂的一个表现。

微生物间复杂而多样的关联关系决定了微生物数据间存在着复杂的语义关联，例如由于微生物物种的演化，不同物种的数据虽然表现为分类不同、性状特征不同，但也存在着许多相似性和相关性；此外微生物之间还存在着生态联系，如共生、竞争等。

2.2 微生物数据建模

微生物菌种资源数据库存储与微生物学家目前正在研究、开发的微生物资源有关的数据，每条记录的信息都与一个活菌种相对应，其核心是菌种的性状信息，分为概述（共性描述）信息和特性信息。此外，每个微生物菌种还有许多扩展的关联信息，比如专利微生物除了性状信息外，还与菌种的专利信息相关联；根据国家资源信息元数据的规范，每个微生物资源都有其相应的核心元数据描述；研究人员在研究的过程中会对微生物数据添加各种注释信息，可以是多媒体类型的图片、视频、文本注释等。

2.2.1 基本对象类

所有基本微生物对象都存储在一个基本类BMO（Basic Microbe Objects）中，该基本类定义了所有微生物对象的共性描述信息，并且为每一个基本微生物对象分配一个资源编号，作为系统内部的唯一性标识。

微生物资源的共性信息包括资源的保藏编号、资源属名、资源种名等，那么BMO定义为：

```
1)  CREATE CLASS BMO (
2)      res_ID CHAR(18),                    --资源保藏编号
3)      genus VARCHAR,                      --菌种资源属名
4)      species VARCHAR,                    --菌种资源种名
5)  );
```

2.2.2 菌种特性建模

菌种特性建模是根据微生物资源的分类，定义不同菌种的特性性状描述。对于每个特殊菌种，通过BMO的一个代理类SpecStrain来定义，SpecStrain可以从BMO选择继承部分属性，同时通过扩展属性来定义具体菌种的特性性状信息，包括

有菌种的基本信息、特征特性、化学成分特性、基因信息、文献信息等。

例如对于某微生物菌种,其特性性状定义有菌落形态、营养类型等表型信息,该菌种代理类定义如下:

```
1)  CREATE SELECTDEPUTYCLASS SpecStrain(
2)    colony VARCHAR,                          --菌落形态
3)    trophic VARCHAR                          --营养类型
4)  ) AS(
5)     SELECT res_ID FROM BMO
6)     WHERE classify( genus, species) = 'SpecStrain'
7)  );
```

其中,第 1 行 SELECTDEPUTYCLASS 声明 SpecStrain 是 BMO 的 SELECT 型代理类,第 2 ~ 3 行声明了代理类 SpecStrain 的扩展属性,是菌种的特性信息定义;第 5 ~ 6 行声明了代理规则,res_ID 是从 BMO 中继承的属性,而 classify 根据 BMO 中基本微生物对象的 genus 与 species 的值,判断一个对象是否属于菌种 SpecStrain。

采用上述方式来建模菌种特性信息,既可方便地实现微生物数据的动态分类,又可有效支持系统的模式演化。当存在新的微生物资源,即有新的基本微生物对象存储到 BMO 中时,对象代理数据库的更新迁移机制能够根据各个 SpecStrain 的分类选择谓词 classify,自动将新的基本微生物对象分发到相应的菌种特性代理类中,即在该代理类中派生代理对象,从而实现微生物数据的自动分类。随着微生物研究的发展,可能会发现某个菌种存在新的特性性状,因此需要修改菌种特性的模式信息。此时,可以在不改变现有数据库模式的情况下,创建待修改 SpecStrain 代理类的代理类,将新增的特性性状建模为该代理类的扩展属性,达到模式演化的目的。

2.2.3　扩展关联信息建模

微生物的扩展关联信息具有不同的类型,包括核心元描述信息、专利信息、注释信息等。每种不同的扩展关联信息,定义为 BMO 的代理类 SpecAssociation。

以微生物注释信息为例,一个菌种资源可以存在不同类型的注释,比如图片、视频或文本记录等。假设类 Media(id, type, file)定义了注释信息,那么可以由 BMO 与 Media 的代理类 Annotation 定义微生物注释信息。

```
1)  CREATE JOINDEPUTYCLASS Annotation(
2)     desc TEXT                                 --注释描述
3)  ) AS (
4)     SELECT res_ID, file FROM BMO, Media
5)     WHERE res_ID = id
6)  );
```

其中,第 1 行 JOINDEPUTYCLASS 声明 Annotation 是 BMO 的 JOIN 代理类,第 2 行扩展定义属性"注释描述",第 4 行声明从 BMO 继承属性 res_ID,从 Media 继承属性 file,Annotation 的模式定义为(res_ID, file, desc)。

以上给出了一种定义微生物扩展关联信息的方式,对于其他类型的扩展关联信息,也可通过 BMO 的 SELECT 型代理类定义,其定义方式与 SpecStrain 类似。

通过以上方法定义扩展关联信息,可以确保系统具有良好的扩展性。与微生物资源相关联的扩展信息会随着时间发生变化,对于增加新类别的扩展关联信息,可以通过创建新的 BMO 的 SpecAssociation 代理类来实现;如果某个类别的扩展关联信息不再需要,只需要将该类别关联信息对应的 SpecAssociation 代理类删除即可,对系统其他信息管理不会发生影响。

2.2.4　建模举例

图 1 描述了一个简单的微生物数据库模型在对象代理数据库中的实现。在该例子中,基本微生物对象通过其属名和种名加词来表达,所有的基本微生物对象存储在基本对象类 Microbe 中,且每一个对象都有一个唯一的资源保藏编号(res_ID)。微生物菌种进一步被分为三类:细菌、古菌和食用菌,每一类菌种建模为 Microbe 的一个代理类,其属性除了从 Microbe 继承的微生物资源保藏编号外,涵盖了具体菌种的特性性状信息。在扩展关联信息建模方面,与微生物资源相关联的多媒体信息通过 Microbe 的代理类 Multimedia 来建模,而微生物资源的核心元数据则建模为 Microbe 的代理类 MetaData。

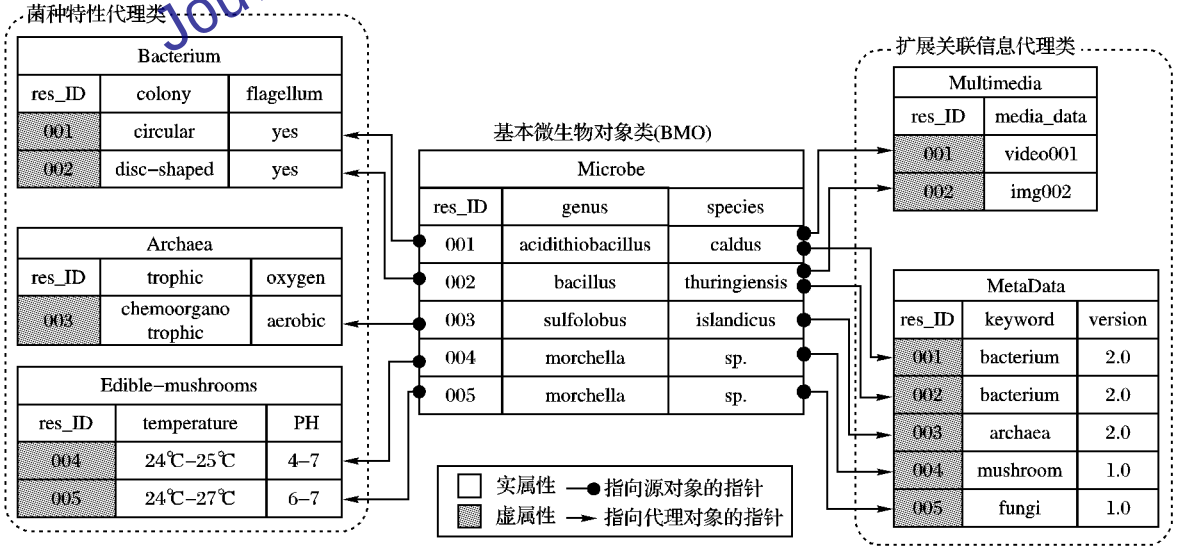


图 1　一个简单微生物数据库

2.3　微生物数据检索

微生物数据库的核心是微生物资源性状信息。在一个微生物信息系统中,典型的微生物数据检索包括:微生物共性信息查询,微生物菌种特性信息查询,微生物完整信息(完整性状信息及各种扩展关联信息)查询。

对于微生物共性信息查询,可以通过对 BMO 进行检索得到待查询信息。对于微生物特性信息查询,可以通过对某个菌种的 SpecStrain 代理类进行检索得到待查询信息。例如,在图 1 中,要查询资源编号为 003 的古菌的"营养类型",查询语句可以表示为:

```
SELECT trophic FROM Archaea WHERE res_ID = 003;
```

需要注意的是，res_ID 是从 Microbe 继承的虚属性，并不物理存储其值，这样可以避免数据冗余，当读取属性值时，通过切换操作读取其源属性的值。

如果需要查询微生物数据的完整信息，有别于传统数据库需要将多个表进行连接操作，我们可以利用对象代理数据库特殊的跨类查询机制实现查询。例如，在图 1 中，本文要查询所有细菌的完整信息，包括其所有性状信息、多媒体关联信息及其核心元数据，查询语句可以如下表示：

```
SELECT *, (bacterium -> microbe).genus,
  (bacterium -> microbe).species,
  (bacterium -> microbe -> multimedia).media_data,
  (bacterium -> microbe -> metadata).keyword,
  (bacterium -> microbe -> metadata).version
FROM bacterium;
```

3 微生物信息系统的实现

3.1 系统体系结构

基于上述方法，我们在国家科技基础条件平台建设的“教学实验用微生物资源整理、整合与共享试点平台”子项目中实现了一个微生物信息服务系统，该系统在统一菌种资源描述规范的基础上，将我国 11 所高校保藏的微生物菌株资源进行标准化整理、整合，最终实现微生物菌种资源的信息共享，其体系结构如图 2 所示。

系统以建设“教学实验用微生物菌种资源数据库”为核心进行开发，主要分为两部分：一是围绕保藏在全国 11 所高校的微生物菌种资源数据的数据采集功能；二是针对微生物菌种资源数据库的各种功能及服务。

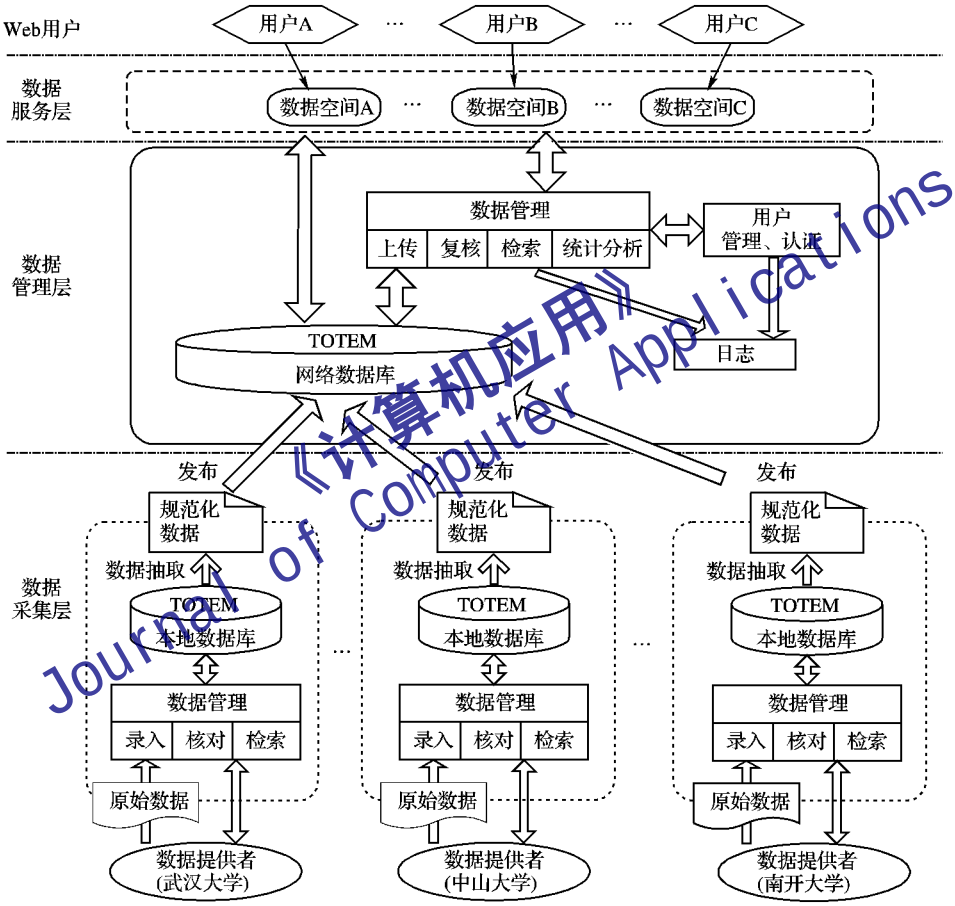


图 2 微生物信息系统体系结构

“教学实验用微生物菌种资源数据库”涵盖的微生物菌种资源包括古菌、细菌、大真菌、食用菌等 14 个菌种，本文使用自行开发的对象代理数据库管理系统 TOTEM 实现“教学实验用微生物菌种资源数据库”。首先，将属于不同菌种的微生物数据根据微生物资源共性描述规范抽取为基本微生物数据存放在 BMO 中；其次，为 14 个菌种按照其特性的不同分别创建相应的菌种特性代理类，由此形成微生物资源的性状库。此外，扩展关联信息包括微生物资源核心元数据、菌种多媒体描述、微生物专利信息等，我们分别创建 BMO 的扩展关联信息代理类，以记录微生物资源不同类别的扩展关联信息。

如图 2 所示，微生物菌种资源数据库分为本地数据库和网络数据库两部分。在数据采集阶段，由研究领域富有经验的专家负责选取数据源条件好的单位，将其所保藏的微生物资源数据入库（本地数据库）。数据源提供者在本地数据库所存储的规范化数据在通过专家评审核对后，再由网络上传发布到网络数据库中。

3.2 数据采集层

数据采集层负责微生物数据采集，建立本地微生物资源数据库。该层的系统功能包括：数据规范化、数据录入、数据核对、数据发布、数据检索等。在微生物资源数据库的建设过程中，数据源的规范化非常重要，因此在数据准备阶段，需要根据各个菌种制定的规范将原始数据进行规范化，以保证数据质量。数据采集层提供不同菌种的格式转换程序将原始数据加工转化为规范化数据。数据录入本地数据库的流程包括数据选择、整理、编码、规范化、录入、核对等步骤。用户可以基于 Web 实现数据的录入和核对，系统还提供批量的数据录入程序将大数据量的规范数据导入到本地数据库中。数据发布功能将存储于本地数据库的微生物数据上传至网络服务器

中,支持数据的选择性上传和断点续传,保证数据上传过程的鲁棒性。数据检索模块基于本地数据库所存储的数据,提供简单检索、复杂检索和逻辑检索等数据检索功能。

3.3　数据管理层

针对网络数据库——“教学实验用微生物菌种资源数据库”的数据管理层通过部署于网络服务器的系统实现,负责管理集成了 11 所高校涵盖 14 个菌种的教学实验用微生物菌种资源数据,实现采用了 B/S 体系结构,使用对象代理数据库 TOTEM 管理微生物菌种资源数据,Web 服务器采用 Apache,并用脚本语言 PHP 开发。系统除了数据库模块外,还包括用户管理模块、数据管理模块以及日志管理模块。

用户管理模块实现用户的注册、管理和认证等功能,其核心是基于角色的访问控制策略。日志管理模块处理用户浏览的记录,并把它们保存在日志文件中。数据管理模块提供数据的上传、复核、检索与统计分析等功能。

除了由数据采集层将数据由本地数据库上传至网络数据库外,数据提供者可以使用数据管理层的数据上传功能,通过 Web 页面向网络数据库上传微生物数据,并由数据约束性检查功能保证上传的数据符合相应的数据规范。领域专家还可以对网络数据库中的数据进行审核评价,数据复核功能可以对数据库数据进行检查和更新操作。通过系统的数据统计分析功能,可以随时了解数据库中的数据情况,提供基于数据源以及基于不同菌种的数据统计与分析操作。

数据检索功能实现数据的简单查询、复合查询和逻辑查询,用户既可以选择查询微生物菌种资源的性状信息,也可以查询诸如核心元数据、专利微生物信息等不同类别的关联信息。简单查询主要通过经常检索的数据项,如菌株保藏编号、属名、种名等进行检索。复合查询通过组合多个数据项进行数据检索。逻辑查询通过对多个数据项任意进行与、或、非运算检索数据。对于性状信息查询结果,既可以选择查看数据的概要信息,也可选择查看数据的详细信息,详细信息包括菌株的所有性状信息以及相关联的多媒体信息。此外,三种不同的数据查询功能都支持对数据结果进行二次查询,以精化查询结果。

3.4　数据服务层

数据服务层以数据空间[7]的形式,为用户提供个性化的数据管理服务。其功能包括数据空间管理,个性化数据扩展,数据共享。

用户可以根据自己的个性化需求,创建自己的个性化数据空间,系统会根据数据空间的模式定义,将微生物资源数据库中满足选择条件的微生物资源数据添加到数据空间中,当有新的微生物数据添加到微生物资源数据库中,系统也会将相应数据推送到用户数据空间中。数据空间实现了用户的个性化数据订阅。

个性化数据扩展包括数据自定义分类及数据自定义扩展。由于微生物的分类多样性,在现有数据库菌种分类方式的基础上,允许用户在数据空间中对数据进一步进行分类管理,例如将细菌数据分类为杆菌和球菌,并以不同的视图来查看数据空间数据。在现有数据模式定义的基础上,还可以在数据空间中实现数据模式演化,添加用户自定义数据。例如对数据空间中细菌数据模式扩展一个自定义属性,用以添加生物注释信息。

数据共享允许用户共享其自定义扩展数据,并允许用户跨数据空间查询其他用户数据空间的共享数据。例如不同用户都扩展了细菌注释信息并共享,且对相同的细菌数据添加了注释信息,那么在查询细菌性状信息时,还可以查看细菌的注释信息,并通过跨数据空间查询查看到其他用户对该数据的共享注释信息。

本文实现的微生物信息服务系统的操作界面如图 3 所示。右上方的区域为用户管理功能区域,可以修改个人信息,管理员还可进行用户管理操作。左边侧边栏为系统功能菜单选择域,除用户管理外的各功能操作都在这个区域中选择,图中展开的两个功能菜单分别是针对微生物数据管理的功能菜单以及针对数据空间管理的功能菜单。右侧的区域为具体的功能操作及信息显示区域,图 3 中显示的是查询微生物性状信息后的查询结果。

图 3　微生物信息系统界面

3.5　性能测试

为测试系统的性能,本文在对象代理数据库 TOTEM 和对象关系数据库 PostgreSQL 上分别实现一个微生物信息管理系统,前者使用本文提出的对象代理模型方法进行数据建模,而后者采用传统的关系模型进行数据建模。实验使用的服务器配置如下:4 × P3 900 MHz CPU,32 GB 内存容量,200 GB 硬盘,RedHat Enterprise Linux 4.0 操作系统,TOTEM 2.0 / PostgreSQL 8.1.2 数据库系统, Apache 2.0.54 + PHP 5.0.4 Web 服务器。

测试时,逐渐增加系统中微生物菌种资源的数量(两个测试系统具有一样的微生物菌种资源),测试在不同数据规模下两个系统消耗存储空间的多少,以及通过复杂逻辑查询显示微生物详细信息所需的查询响应时间的长短,以此评价两个系统的优劣。实验结果分别如图 4 和图 5 所示。

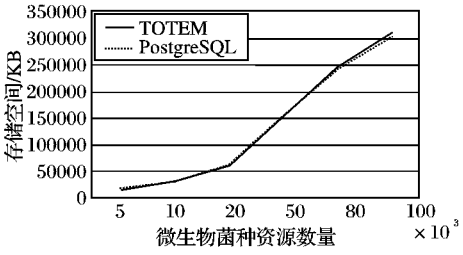


图 4　存储空间实验结果

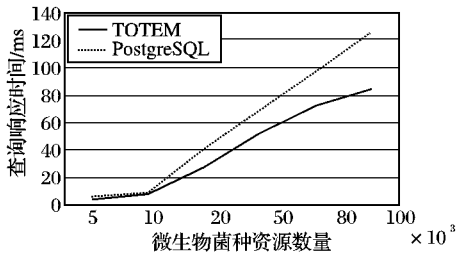


图 5　查询响应时间实验结果

从实验结果可以得出以下的结论:1)基于对象代理模型的微生物信息系统与基于关系模型的微生物信息系统消耗的存储空间基本相同,后者略占优势;2)在大数据量情况下,利

(下转第 17 页)

设节点性能服从正态分布。

图 4 描述了当节点数目从 1 到 100，方差从 0.1 到 50 变化时 *Gini* 的取值规律。可以看到，随着节点数目的增多 *Gini* 值在逐渐增大，而曲面的波动性则表明节点数目和节点性能分布是一个相互制约的关系。在这一步，我们计算出的最佳 *Gini* 值为 0.251。

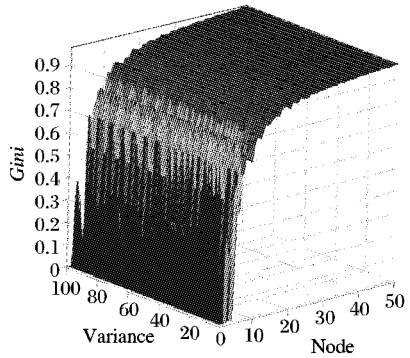
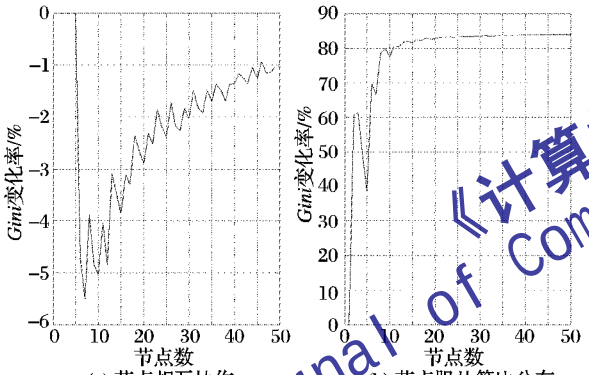


图 4　*Gini* 值在不同约束条件下的变化

现实中集群节点间往往存在相互协作的关系，图 5(a)就描述了 *Gini* 值在这种约束下的变化规律，即 *Gini* 值随协作节点数目的增多而变小，变化率为负值则表明节点协作性对 *Gini* 值的优化存在一定的副作用。



(a) 节点相互协作　　(b) 节点服从等比分布

图 5　*Gini* 值在特定约束条件下的变化率

考虑到图 4 中 *Gini* 值在不同正态分布下变化不是很大，故特别测试了不同分布对 *Gini* 值的影响。图 5(b)描述了 *Gini* 值在节点服从等比数列时的变化规律，可以看到等比数列的公比从 1 到 50 的变化过程中，*Gini* 值将近被优化了 85%，最小 *Gini* 值为 0.180。

从上述实验可以看到，在不断调整约束条件的情况下，*Gini* 值不断变小。因此本文评测和优化集群系统的方法是可行的，只需要在不同环境下设定不同的约束条件。

4　结语

本文通过分析负载均衡机制与集群节点结构的相关性，提出了一种评测和优化集群节点结构的方法。这种方法不仅仅可以应用在集群系统中，也可以应用到其他拥有多个节点的系统中，它是对软件与硬件协同发展方式的一种探索。

参考文献：

[1] HUI C C, CHANSON S T. Improved strategies for dynamic load balancing[J]. IEEE Concurrency, 1999, 7(3): 58 - 67.

[2] 景波. 实时集群计算机的体系结构研究[J]. 硅谷, 2008(14): 36 - 38.

[3] 邱钊, 陈明. Web 集群负载均衡算法比较[J]. 现代计算机, 2006(8): 61 - 63.

[4] 唐丹, 金海, 张永坤. 集群动态负载平衡系统的性能评价[J]. 计算机学报, 2004, 27(6): 803 - 811.

[5] 刘安丰, 陈志刚, 曾志文, 等. 基于数据挖掘的 Web 集群负载均衡算法[J]. 计算机工程与应用, 2003, 39(25): 59 - 61.

[6] 陈忠林, 孙雁, 王静. 分布式网络环境下的负载平衡原理及算法[J]. 四川大学学报: 工程科学版, 2003, 35(6): 97 - 100.

[7] 余海燕, 郑笑飞. 几种负载均衡解决方案的比较[J]. 信息系统工程, 2000(9): 28 - 29.

[8] 盛骤, 谢式千, 潘承毅. 概率论与数理统计[M]. 北京: 高等教育出版社, 2001.

[9] 同济大学应用数学系. 高等数学[M]. 北京: 高等教育出版社, 2002.

[10] TAN PANGNING, STEINBACH M, KUMAR V. 数据挖掘导论[M]. 范明, 范宏建, 译. 北京: 人民邮电出版社, 2008.

[11] 李中福. 计算机集群技术应用研究[D]. 新乡: 河南职业技术师范学院, 2006.

(上接第 9 页)

用对象代理模型建立的微生物信息系统在查询响应时间方面要明显优于利用关系模型建立的微生物信息系统，且随着数据规模的增大，这种优势越是明显。

4　结语

本文在分析现有微生物数据库实现技术的基础上，提出了一种基于对象代理数据库的微生物数据管理方法，通过该方法开发实现了一个微生物信息服务系统。该信息系统中的微生物菌种资源数据库使用我们自行开发的对象代理数据库管理系统 TOTEM 实现，能支持数据的动态分类，提供高效的数据检索，避免产生数据冗余，具有较好的系统性能。该系统已经在全国 11 所高校中运行，建立了微生物菌种资源丰富的“教学实验用微生物菌种资源数据库”，整理、整合了我国 11 所高校科研机构所保藏的、涵盖 14 个菌种的微生物资源。

参考文献：

[1] 李雁, 郑从义. 微生物物种多样性的保护与其资源保藏[J]. 氨基酸和生物资源, 2003, 25(3): 4 - 6.

[2] BRY F, KROGER P. A computational biology database digest: Data, data analysis, and data management[J]. Distributed and Parallel Databases, 2003, 13(1): 7 - 42.

[3] BENSON D A, KARSCH-MIZRACHI I, LIPMAN D J, *et al*. Genebank[J]. Nucleic Acids Research, 2000, 28(1): 8 - 15.

[4] PENG ZHIYONG, SHI YUAN, ZHAI BOXUAN. Realization of biological data management by object deputy database system[C]// Transactions on Computational Systems Biology V. Berlin: Springer Verlag, 2006: 49 - 67.

[5] PENG ZHIYONG, KAMBAYASHI Y. Deputy mechanisms for object-oriented databases[C]// Proceedings of the IEEE 11th International Conference on Data Engineering. Washington, DC: IEEE Computer Society, 1995: 333 - 340.

[6] ZHAI BOXUAN, SHI YUAN, PENG ZHIYONG. Object deputy database language[C]// Proceedings of the 4th International Conference on Creating, Connecting and Collaborating through Computing. Washington, DC: IEEE Computer Society, 2006: 88 - 95.

[7] FRANKLIN M, HALEVY A, MAIER D. From databases to dataspaces: A new abstraction for information management [J]. SIGMOD Record, 2005, 34(4): 27 - 33.