GDB--一种数据库网格系统*)

冯建新　于浩海　王光兴
(东北大学信息科学与工程学院　沈阳 110004)

摘　要　网格计算技术将资源有效整合，从而提供统一且灵活的资源共享，但目前主要应用于科学研究，不适于民用和商业应用。通过研究网格计算的重要内容——数据库共享技术，本文提出了数据库网格系统GDB。资源代理、中间件和XML技术的采用，使GDB能够有效整合数据库资源，解决民用和商业领域应用中所遇到的问题。实践表明，GDB有着广阔的应用前景。

关键词　网格计算，数据库，中间件，XML

GDB--a Grid Database System

FENG Jian-Xin　YU Hao-Hai　WANG Guang-Xing
(School of Information and Engineering, Northeastern University, Shenyang 110004)

Abstract　The grid computing can integrate resources effectively to get a common and flexible resource sharing. But at present, the grid computing mainly applies to the science research, not to the civil and commercial environment. Based on the research of the important content of the grid computing—the database sharing, a grid database system (GDB) is proposed in this paper. The resource proxy, the middleware and XML are adopted in GDB in order to integrate database resource effectively and solve the problems in the civil and commercial environment. Practices show that GDB has a bright prospect.

Keywords　Grid computing, Database, Middleware, XML

1　前言

网络发展要求各种应用和计算不断地扩展其实用范围。将集中式的、单机版的应用程序转换为分布式的网络程序，或者将企业或部门的资源(包括硬件和软件资源)在该企业或部门的局域网络中共享已经不是主要问题。而怎样将多个分布在不同企业或部门中的硬件和软件资源有效地整合到一起，为整个网络提供一种统一的、灵活的资源共享服务是目前网络发展急需解决的问题。

网格计算概念正是为解决这个问题而提出的。网格计算是"个体，单位和资源的动态集合中的一种灵活、安全的、相互协作的资源共享的体系结构"[1]，它将分布于不同物理和逻辑域中的各种计算资源(网络，数据，CPU，存储等)有机地整合为一个统一体，使资源共享更加强大、方便、有效。

网格计算中十分重要的部分是数据共享，而其中数据库的共享又显得极为重要。但是现有的数据库访问接口，例如ODBC和JDBC，并不适应网格计算中访问数据库的要求。为了有效地利用现有的数据库，很多研究部门提出了基于网格计算的数据库体系结构和项目，例如欧洲数据网格(EDG)的喷火计划[2]。目前网格计算技术主要应用于科学研究，但是正如Web技术的发展过程一样，网格计算的未来发展趋势必然是在广阔的民用和商用领域。因此我们提出了网格数据库GDB(Grid Database)。

GDB是一种基于中间件的网格数据库技术，它可以灵活地实现在多种应用层协议和分布式计算框架之上，例如HTTP，SMTP和SOAP[8]等，它使用灵活的数据描述语言XML[9]传输数据库数据，因此它可以方便地整合分布于不同的网络中的数据库资源。GDB的另一特点是可以提供在多种用户接口(例如ODBC和JDBC)中，这样就简化了原有的程序移植。

本文首先参照喷火项目讨论了目前网格数据库系统的不足，其次讨论GDB系统的体系结构、通信机制、工作流程及其特点，最后是本文的结论。

2　现有网格数据库系统的不足

当前大部分的网格数据库系统都应用于科学研究和科学计算中，因此多数网格数据库系统没有或很少考虑民用和商用领域的特点，导致网格技术在这些领域的推广和应用方面存在着困难。"喷火"是当前较为成熟的基于网格技术的数据库共享项目。作为EDG的一部分，它带有目前网格数据库系统的一些基本特征，下面就以它为对象讨论一下目前的网格数据库系统的不足。

在"喷火"项目中，用户直接通过HTTP+SSL访问服务端，由服务端完成认证和授权工作，如果仍然采用这种方式应用在民用和商用环境中至少存在以下三个问题。首先民用和商用环境中的用户数量明显多于科学计算和科学研究领域中的使用者，如果每一个用户的访问都要由服务端认证和授权很容易造成服务端瓶颈的产生。其次在民用和商用环境中客户终端很复杂，客户终端可能是个人电脑，也可能是移动电话，甚至是微波炉，其中某些客户终端的计算能力会很有限，根本不可能完成复杂的加密和解密算法。因此目前网格数据库系统中的安全机制显然不适用于这些终端。再者如今某些

*)基金项目：国家"九五"重点科技攻关项目基金(No. 96-B03-04-03)、国家自然科学基金资助项目(69973011)。冯建新　博士研究生，主要研究方向为分布式数据库等。于浩海　硕士研究生，主要研究方向为分布式数据库。王光兴　教授，博士导师，主要研究方向为宽带计算机网络等。

民用和商用领域的客户终端，已经有了简单而有效的认证标识(如移动电话的电话号码)，再使用另一种机制进行认证明显是一种浪费。

另外使用“喷火”系统时要求特定的接口，这就要求对现有系统重新编程。考虑到民用和商用领域应用程序的复杂性和数量，这种修改需要大量的投资，这对于运营商来说是不愿接受的。“喷火”的另一个不足是用户使用 URL 来定位所需要的数据库服务的位置，而一般的民用和商业领域的用户可能只知道需要什么样的服务而并不知道服务所在的位置。

从上面的分析可以看出成功地应用在科学计算和科学研究领域的网格数据库系统目前并不适用于民用和商用领域。

3 GDB 系统

3.1 系统结构

在有着基础性贡献的“The Anatomy of the Grid”[1]中给出了网格的四层基本结构，如图 1 所示。

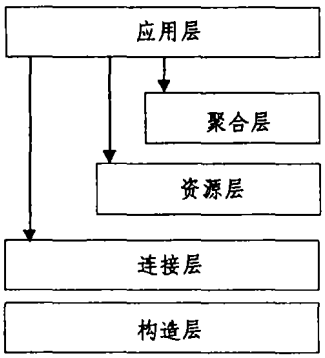

图 1 网格的体系结构

构造层处于体系结构的最低层，它的主要功能是访问实际资源，为上层利用资源提供相应的条件。它至少应该包括状态查询功能和资源管理功能，另外由于需要直接访问资源，该层必须能够直接解决异构性的问题。连接层主要解决资源间的通信、安全和认证等方面的问题。资源层构建于连接层之上，使用户可以查询和使用资源。聚合层用来解决不同资源间相互协作的问题。

需要注意的是，这四层仅仅是从概念上将网格分成了四层，在实际的网格系统中，一个组件可能实现了几个层的功能，也可能仅仅实现了一层中的某些部分的功能。一些应用也可能越过某些层。

在参考文[1]的基础上本文定义了 GDB 的体系结构。GDB 系统的体系结构主要由应用程序接口、实现绑定、客户端、通信协议、服务端、数据库和资源代理几个部分组成，如图 2 所示。

应用程序接口为 GDB 的终端使用者提供了一个应用 API 接口或直接面对终端用户的 GUI 接口，它们可以使用传统的 ODBC 或 JDBC 接口访问 GDB，以求最大可能地利用原有代码。

实现绑定层的任务就是根据用户的需求将服务绑定到特定的实现上，即 GDB 的客户端上。GDB 允许使用多种实现方法构建网格系统的中间层软件(例如 SOAP，CORBA 和 HTTP 等)以完成各种资源的整合。

客户端代表用户完成数据库访问。客户端应在所属的资源代理中注册，完成通过资源代理为用户找到合适的数据源，以类似 HTTP 的方式向服务端发出请求和接收处理服务端的响应的功能。

基于 XML 的通信协议完成连接层的通信工作，本文 3.2 节将对该通信协议进行具体介绍。

服务端作为数据库的代理，使用连接池通过本地数据库访问接口与数据库保持连接，处理客户端的请求并做出响应。

资源代理实现资源和聚合层的功能，以及用户和资源的认证工作。多个资源代理形成网络，实现数据资源的查询、认证、整合等工作。

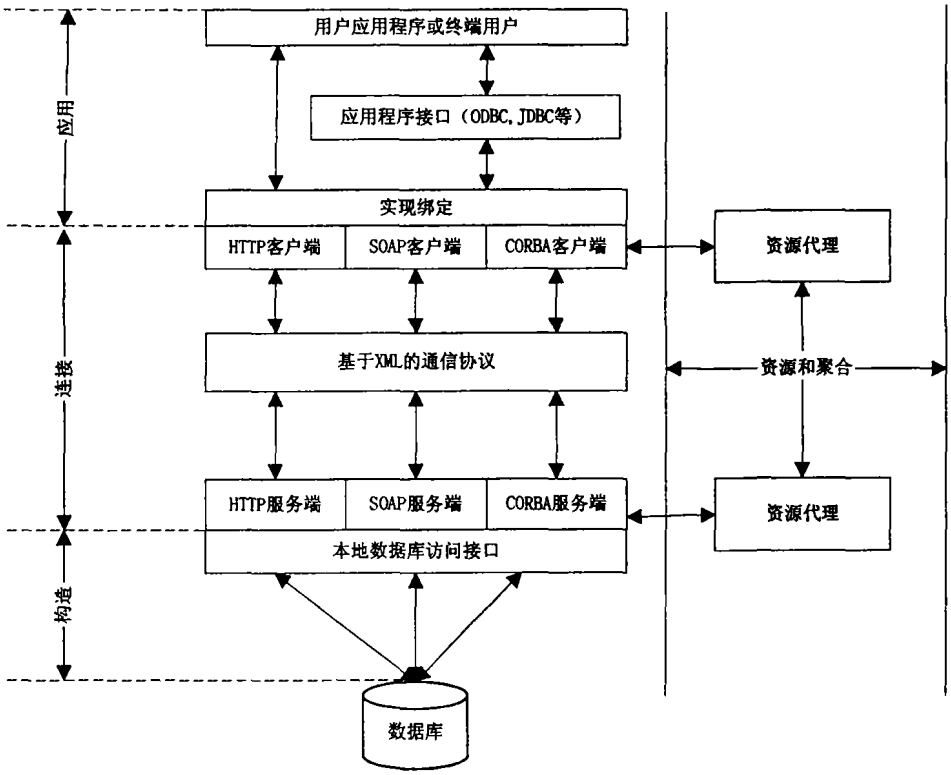

图 2 GDB 系统的体系结构

3.2 通信机制

为了灵活、高效地传输数据库请求信息和数据库响应信息，GDB 系统使用基于文本的 XML 技术。利用 XML 的可扩展性和强大的数据描述能力描述在数据库中各种不同的数据类型。GDB 的通信协议定义了命令请求、结果请求和响应三种基本消息类型以及各种子消息。下面将结合用于定义通讯协议的 XML SCHEMA 代码对通信机制进行简单的介绍。

命令请求用来描述数据库访问命令，它主要由用户信息和 SQL 命令两部分组成。用户信息就是用户的资源代理 ID 和用户 ID，用于 GDB 服务端验证用户身份。SQL 命令包括单一 SQL 命令和 SQL 命令组 2 种。单一 SQL 命令包括 SELECT，UPDATE 等标准 SQL 命令，也可以是存储过程或调用数据库的系统函数，但只能包含一条可执行的命令。SQL 命令组是一组有先后执行顺序的非查询命令的 SQL 命令，它们在一起作为一个事务要么全部执行要么都不执行。下面用 XML SCHEMA 代码描述命令请求。

```
〈element name="commandRequest"〉
  〈complexType〉
    〈sequence〉
      〈element name="proxyID" type="String"/〉
      〈element name="userID" type="string"/〉
      〈element name="command"/〉
        〈complexType〉
          〈choice〉
            〈element name="singleCommand" type
              ="SQLCommand"/〉
            〈element name="batch" type="sw:
              Batch"/〉
          〈/choice〉
          〈/complexType〉
        〈/element〉
      〈/sequence〉
    〈/complexType〉
〈/element〉
```

当数据库访问所产生的查询结果比较大时，需要客户端与服务端多次传输完成并且客户端可以根据用户的需要只索取部分的数据。客户端就利用结果请求索取数据。它主要包括用户信息和结果信息两部分。用户信息包括用户资源代理 ID、用户 ID 和序列号（将要索取的结果的序列码）。结果信息包括结果的格式和结束命令两种，结果的格式即确定使用什么方法传送结果，结束命令即不论是否所有的结果都传输结束，停止结果传输的命令。其 XML SCHEMA 代码描述如下：

```
〈element name="resultRequest"〉
  〈complexType〉
    〈sequence〉
      〈element name="proxyID" type="string"/〉
      〈element name="userID" type="string"/〉
      〈element name="continueCode" type="String"/〉
      〈group〉
        〈choice〉
          〈element name="result" type="sw:
            Result"/〉
          〈element name="close" type="string"/〉
          〈/choice〉
        〈/group〉
      〈/sequence〉
    〈/complexType〉
〈/element〉
```

响应将数据库访问的结果返回给客户端。它主要包括：序列号（结果的序列码）、结果的具体内容和异常（将把在服务端处理客户请求或者传输时出现的异常情况报告给客户）。描述响应的 XML SCHEMA 代码如下。

```
〈element name="response"〉
  〈complexType〉
    〈sequence〉
      〈element name="continueCode" type="String"/〉
      〈element name="content"〉
        〈complexType〉
          〈choice〉
            〈element name="result" type="sw:Re-
              sult"/〉
            〈element name="exception" type="sw:
              Exception"/〉
            〈/choice〉
          〈/complexType〉
        〈/element〉
      〈/sequence〉
    〈/complexType〉
〈/element〉
```

3.3 GDB 的工作流程

通过 GDB 进行一次数据访问请求、响应所构成的时序图对 GDB 的工作流程进行了介绍，如图3所示。

为了简单起见，图3省略了绑定层，和本地数据库访问接口层，以及 XML 通信等与实现相关的详细内容。

用户在访问数据库前，发出消息1激活客户端，客户端在响应激活前，先发出消息2在一个资源代理1中进行注册，注册通过后再发给用户确认消息4。从此以后将由资源代理1担保该客户端的身份。

用户发出访问数据库请求的消息5。

客户端根据用户的请求，向资源代理1发出消息6查询数据源的地址。资源代理1向已知的资源代理（包括它自己）转发该消息。

资源代理1搜集返回的消息8，并根据一定的策略（性能优先、价格优先或性价比优先）选择一个合适的资源代理 n，并随机生成一个用户 ID，该 ID 和确认信息由消息9发给被选定的资源代理 n，资源代理 n 选定一个服务端并将该服务端的地址通过消息10返回给资源代理1。

资源代理1将用户 ID 和服务端地址通过消息11返回给客户端，客户端在得到用户 ID 和服务端地址后直接通过消息12向服务端发起访问数据库的请求。

服务端先用客户端的用户 ID 验证客户端的身份（消息13），认证通过后（消息14）服务端再直接访问本地数据库（消息15、消息16），并通过消息17将结果返回给客户端。

客户端通过消息18将结果返回给用户。一次请求-响应操作结束。

这里需要注意的是为了保证安全，用户 ID 具有时效性，客户端应该定期访问资源代理得到新的用户 ID。如果用户的请求中包括了服务端的地址，那么消息7和消息8将是不必要的，资源代理可以直接通过消息9建立用户 ID 和服务端的绑定；服务端将定义通过消息21和消息22将数据库和服务端本身的最新信息注册到资源代理。

3.4 GDB 的特点

GDB 引入了资源代理。在 GDB 中资源代理完成认证、授权、管理等多重任务，并且不同的资源代理互连形成代理网络从而有效地整合各种资源。资源代理的引入至少带来了多项好处。每一个资源代理管理一定数量的客户端和服务端，这就有效地分割了整个网格数据库系统，使系统在用户和资源数量激增时仍有良好的性能。资源代理之间可以使用网格安全机制例如 GSI（Grid Security Infrastructure）[6,7]或现有的安全机制，甚至不同的资源代理之间可以使用不同的安全机制。相对安全的环境（LAN，VPN 等）促使资源代理以及它们所管理的客户端和服务端之间的认证可以相对简单并且可以充分利用系统特征和原有认证方法，这对于运算能力低的终端尤为重要。比如在一个移动电话网络中，移动电话（客户端）的电话号码就可以作为资源代理认证移动电话的唯一的标识符。

GDB 增加了中间件。用户应用程序不与数据库直接连接而是直接操作中间件的客户端，其接口可以使用传统的

ODBC 或 JDBC 接口，甚至可以提供人机图形接口。中间件的客户端代表实际客户与中间件的服务端通信以完成数据库的访问，并由其解决在跨越广域网时可能出现的各种问题。

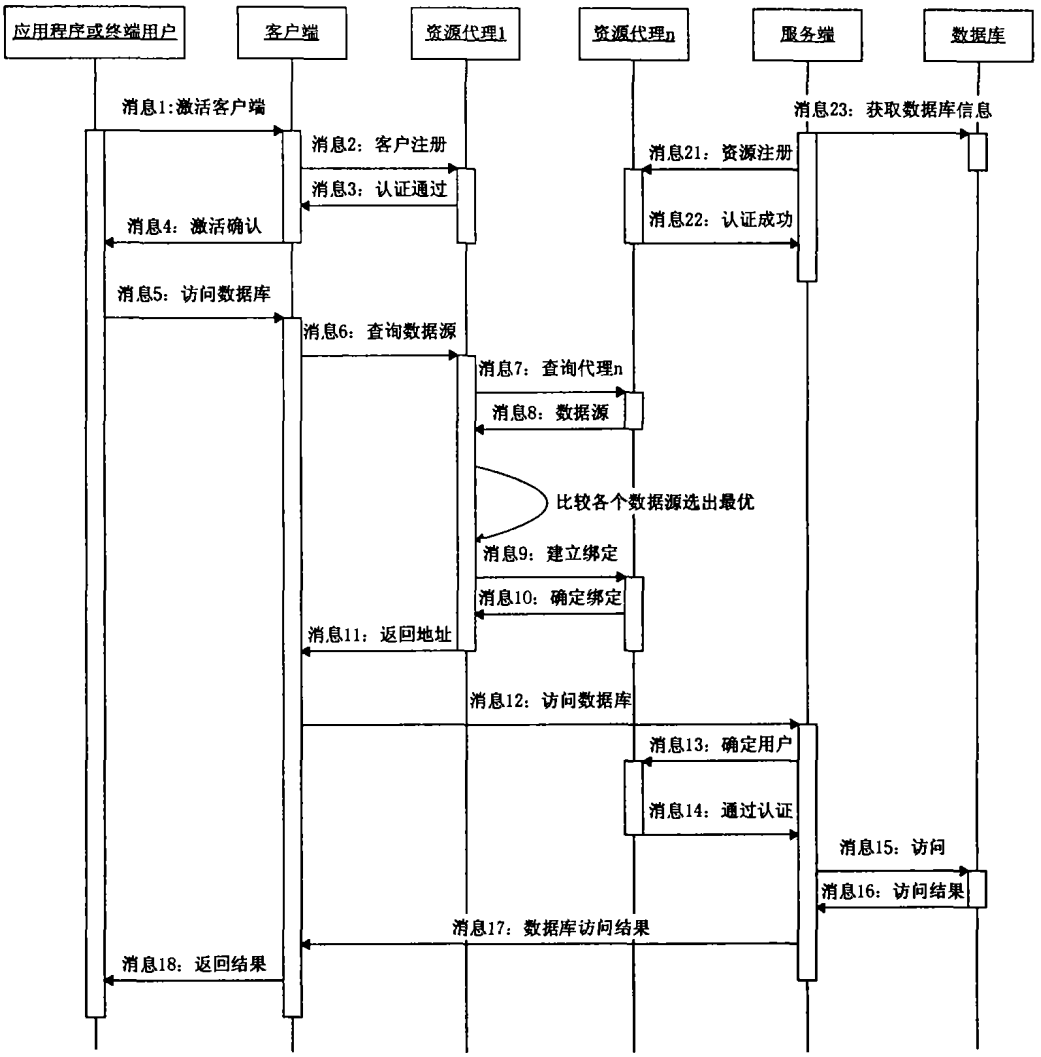

图3 GDB 系统的工作流程

GDB 基于 XML 的通信协议具有很好的可扩展性。同时由于 XML 是一种通用的数据描述语言，所以可以使用不同的实现语言，在不同操作系统上相互通信。GDB 使用 REQUEST-RESPONSE 通信模式，既简化了通信过程又提高了资源利用率，并且可以使用会话技术满足数据库访问对面向连接的要求。GDB 允许使用多种实现方法和底层通信如 SOAP，CORBA，HTTP 和 SMTP 等。

结论 GDB 实现了数据库资源的有效整合，特别值得注意的是 GDB 与其它的网格数据库系统最大的不同是它从一开始就是为民用和商用领域的应用开发的。并为此引入了一些新的特征如资源代理。因此它有更广阔的应用前景。目前，我们在一个智能电话系统数字业务平台中利用 CORBA 作为中间件实现了 GDB 的客户端和服务端，并且提供了一个基于图形界面的用户接口，使用者即智能电话用户，可以将一些查询请求(例如，高考成绩的查询)发送到智能电话系统数字业务平台，智能电话系统数字业务平台中的 CORBA 客户端将 SQL 命令交给 CORBA 服务端并由服务端完成最后的查询工作。

参 考 文 献

1 Foster I, Kesselman C, Tsudik G, et al. The Anatomy of the Grid. International Journal of Supercomputer Applications, March 2001, 15(3): 200～222

2 William B, Diana B, Wolfgang H, et al. Project Spitfire - Towards Grid Web Service Databases: [Technical report]. Global Grid Forum Informational Document, GGF5. Edinburgh, Scotland, July 2002

3 Grimshaw A S, Wulf W A, French J C, et al. Legion: The next logical step toward a nationalwide virtual computer: [Technical report No. CS-94-12]. June 1994

4 Grimshaw A S, Wulf W A. Legion- A view from 50,000 feet. In: Proc. of the Fifth IEEE Intl. Symposium on High Performance Distributed Computing. IEEE Computer Society Press, Los Alamitos, California, Aug. 1996

5 Lindahl G, Grimshaw A, Ferrari A, et al. Metacomputing-What's in it for me. White paper, http://legion.virginia.edu/papers.html. Sep. 1998

6 Utler B R, Engert D, Foster I, et al. Design and deployment of a national-scale authentication infrastructure. IEEE Computer, 2000, 33(12): 60～66

7 Foster I, Kesselman C, Tsudik G, et al. A security architecture for computational grids. In: ACM conf. on computers and security. Nov. 1998. 83～91

8 Box D, Ehnebuske D, Kakivaya G. Simple Object Access Protocol (SOAP)1.1. http://www.w3.org/TR/SOAP, May 2000

9 Bray T, Paoli J, Sperberg-McQueen C M. Extensible Markup Language (XML)1.0 (Second Edition). http://www.w3.org/TR/REC-xml, Oct. 2000

10 Roure D, ennings J D, Shadbolt N. Research agenda for the semantic grid: a future e-Science infrastructure. UK National e-Science Center, www.semanticgrid.org, Oct. 2002