

网格资源检测系统的设计与实施

Design and implementation of detecting grid resource system

杨德仁 (上海 华东师范大学 计算机应用研究所 200062)
栾 静 (上海 华东师范大学 计算机应用研究所 200062)
(乌鲁木齐 新疆师范大学 数理信息学院 830054)
顾君忠 (上海 华东师范大学 计算机应用研究所 200062)

摘要:本文首先介绍了网格、OGSI 和 Globus Toolkits。着重研究了网格编程模型、编程步骤和编程工具,分析和设计了网格资源检测系统,解析了与网格服务实施相关的几个重要文件之间的关系。最后,指出基于新标准的 Globus Toolkits 4 将对设计和实施资源检测系统提出了新挑战。

关键词:网格 资源 编程模式 实施 解析

1 引言

网格始自于 90 年代中期的 I-WAY 项目,随着开源项目 Globus 的实施及其逐步应用,网格就如同 Internet 网络那样,逐步从象牙塔走向商业应用。网格(Grid)之父 Ian Foster 指出,设计网格的目的就是让人们像使用电力那样方便地使用计算机资源如包括计算能力、存储设备和程序等,从而达到计算资源的充分共享。

目前,网格及其运行环境的构造方法及其环境的构建方法有 Tools、分布式操作系统和分布资源管理框架。其中 Globus Tools(简称 GT)是一种开源软件,也是网格的事实标准。GT 经历了 GT1、GT2、GT3 几种版本,前两种版本称为 Pre-WS(Web Service)版本,而后者是基于 WS 的版本。

开放网格服务体系结构(OGSA)是基于 WS 的面向服务的体系结构(SOA),它以服务为中心,强调的是服务实例的位置透明性和多协议绑定,支持对底层各种平台设施的集成。其规范是开放网格服务基础设施(OGSI),OGSI 是一种中间件,其核心是网格服务(在网格环境中为客户提供提供的服务),网格服务依附于接口并通过 WSDL 向外界呈现。网格服务旨在组成虚拟以便计算机充分利用闲置的资源,为集成各种系统提供平台和技术(OGSA)。网格服务遵循一套规范以实现生命周期管理、检测和通知服务状态变化的 Web 服

务。OGSI 网格服务有计算资源、存储资源、网络资源、程序、数据库等,提供了高级接口,隐蔽了底层细节,从而实现了服务与资源的彻底共享。而 OGSI 规范的实施(也是事实标准)是 GT3.x。

2 WS、OGSI 与 GT3

Web 服务是一个用 URI 识别的软件系统,其公共接口用 XML 定义和描述了一些可以通过标准化 XML 消息进行网络访问的操作。Web 服务与其他系统按 XML 消息机制交互。Web 服务是基于标准的框架,用于访问网络应用。Web 服务是程序之间通信的模型,用 SOAP 与 Web 服务通信。Web 服务具有开放性、平台和语言独立性,适于构建基于互联网的松散系统。Web 服务的一个重要特点是无状态性。

OGSI 对 Web Service 的主要改进在于引进了状态和潜在瞬间服务、服务数据、通知、服务分组、接口扩展、生命周期管理、GSH 和 GSR 等。OGSI 利用扩展的 WSDL 和 XML 模式定义,定义了编程模型,引进了状态化的 Web 服务即网格服务,定义了创建、命名和管理服务实例的生命周期的方法、通用的元数据概念,检测和异步通知状态的变化情况等,引进了标准工厂和注册接口以创建和发现网格服务。OGSI 只定义了一套服务原语,并用于构建和组合高级服务。

GT3 堆栈式体系结构如图 1 所示[5]。其中 GT3

核心是指 OGSi(网格服务); GT3 安全服务主要是访问控制; GT3 基本服务包括作业管理、索引服务和可靠文件传输服务; GT3 数据服务指复制管理; 而其他网格服务则是非 GT3 服务。

目前, 搭建 GT3. x 环境的方法已经比较成熟。

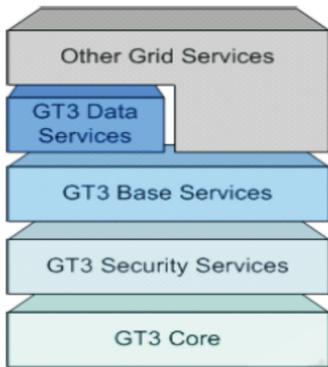


图 1 GT3 体系结构

3 网格资源检测系统设计与实施

3.1 系统需求

为了实现网格资源的实时共享, 检测网格计算机的系统资源是最基本的工作。如要进行作业调度, 就必须掌握目标计算机节点的计算资源的状况, 如计算

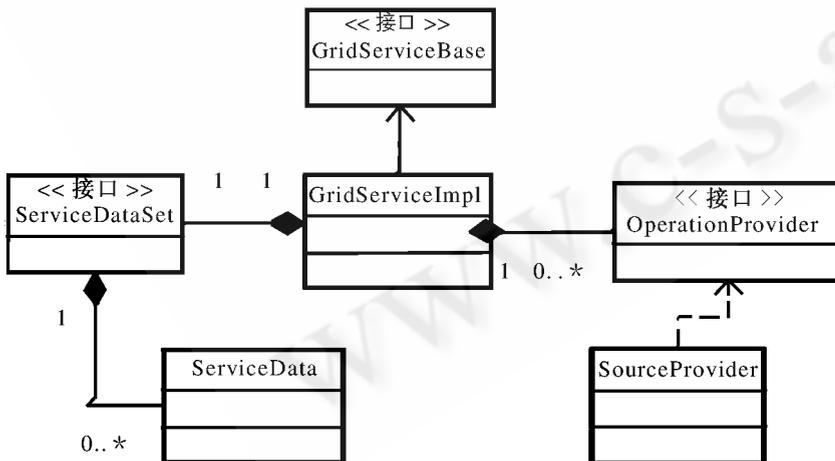


图 2 服务器端设计模式

能力、内存和磁盘空间等。OGSi 和 GT3 提供了良好的服务状态检测机制即服务数据, 可用于检测节点的资源。并且资源检测机制应该是灵活的, 能满足执行不

同作业的资源检测需求, 用户可以根据具体作业定制服务数据: 如作业 A 只需要某节点的计算能力, 而作业 B 可能需要计算能力和存储空间等。

3.2 服务数据

所谓服务数据: 就是对与网格服务相关的信息的信息的结构化描述, 使服务信息容易查询, 以便分类和索引网格服务。每个网格服务可以有任意个相关的服务数据元素。服务数据有两种即状态信息和服务元数据。其中每个元数据可以是单值或多值(数组形式), 其数据类型可以是基本数据类型或复杂数据类型。

3.3 编程模式与接口设计

如前所述, 网格服务是基于 web 服务的, 因此它的分布计算模型也是 Proxy - stub。它包括服务端编程和客户端编程两个部分, 两者通过 WSDL 服务描述文件弱耦合。WSDL 服务描述文件具体描述了服务的接口、服务的调用方法以及服务调用同底层通信协议的绑定情况。客户端用户得到 WSDL 服务描述文件后, 就可以根据该文件生成服务调用的 stub, 并通过该 stub 来完成对服务的调用, stub 用于处理数据与 XML 之间的编码、产生在服务器和客户端使用的 JAX - RPC 接口。

3.3.1 设计模式

服务器端的主要设计模式如图 2 所示。

委托模式使网格克服了 java 的单继承的限制, 实现了多重继承(详见 3.5.1)。服务数据可以用于服务发现和通知机制: 服务发现要不断查询服务数据, 有些笨拙; 而通知机制基于 observer 模式, 是灵活的编程机制之一, 本文就采用这种机制。

客户端的主要设计模式如图 3 所示。

3.3.2 核心类的实现方法

GT3. 0. x 和 GT3. 2 版本提供的系统资源实施类不同。

在 GT3. 0. x 版本中, 要用 C/C + + 编写获取资源的程序, 编译成 DLL 文件(用于 windows 平台) 或 SO 文件(用于 linux 平台), 然后用 JAVA 的 JNI 机制调用。

计算资源的方法是 ComputerMonitor(IP), 用于监

视网格计算机的可用资源如 CPU、MEM 和 DISK 等。

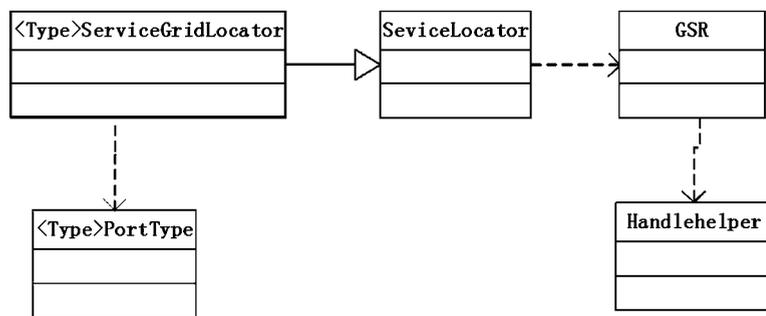


图 3 客户端设计模式

3.3.3 系统信息概念模型设计

系统的数据库模型如图 4 所示。其中 machine 中的 machine_state 字段用于软锁定计算机的资源时使用。数据库系统采用开源产品 Postgresql。

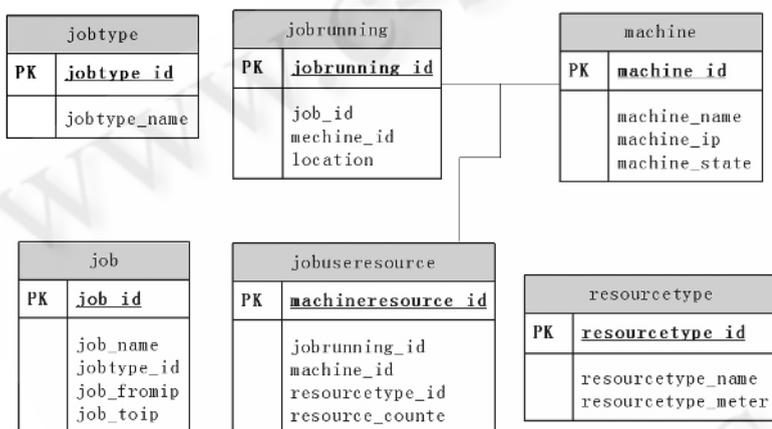


图 4 数据库设计模式

3.4 系统实施

3.4.1 编写网格服务的 5 个步骤

一是定义网格服务接口。有两种方法：直接写 WSDL 文件，或从 JAVA 接口生成 WSDL 文件。实际上，网格服务接口文件是 GWSDL 文件。GWSDL 是对 WSDL 的扩展：接口可以多继承、命名空间和服务数据。GT 提供工具自动将 GWSDL 文件生成 WSDL 文件。

GWSDL 定义了接口、操作及其命名空间（接口和操作属于此命名空间）。编写 GWSDL 文件的顺序是：根元素、命名空间、< gwsdl: portType >、< message >、< types >。

二是实施服务，用 JAVA 编程实现。有两种实施方

法，分别是继承（继承 base class）与代理（操作提供者，把接口中的操作分布在多个类中，这些类就是操作提供者，不继承 base class，只实施 OperationProvider 接口。服务部署文件将告诉容器：GridServiceImpl 提供基本网格服务功能）。相比之下，操作提供者的优点是实现了多重继承（可发布现有的 Java class）、有利于模块化、除耦和可复用设计；而其缺点是 JAVA 编码稍复杂、也要修改部署描述文件。注意在使用操作提供者时，GSDL 和 stubs classes 并不改变！只是实施不同而已，即在部署时做相应变化即可。

三是定义部署参数，编写部署文件（WSD 文件）；该文件将告诉 WEB 服务器如何和在哪里发布网格服务。

四是编译和生成 GAR 文件；这些都用到 ANT 工具（详见本文第 5 部分）。

五是部署服务：也用 ANT 工具。

3.4.2 自动编程工具简介

基于 JAVA 的 GT3 手工编程比较复杂。目前用于网格服务开发的 IDE 有 ANT、Eclipse 及其 GT3 插件。

用 ANT 实施网格服务时，有两种即自顶向下和自下而上两种方法。

自下而上需要提供的参数（文件）：只要有 JAVA 代码，经编译和打包成 JAR 文件。在 build - tools.xml 中指明 JAR 文件和接口类，利用 ANT 可以自动生成桩文件、服务定位器、部署描述符碎片、操作提供者（要委托调用现有 JAVA 代码）。

自上而下需要提供的参数（文件）：只要有 GWSDL 文件。在 build - tools.xml 中指明 GWSDL 文件路径和名称，利用 ANT 可以自动生成绑定、部分服务、桩、服务定位器、部署描述符碎片、操作提供者和实施类。但要手工为实施类添加业务逻辑。

IDE 要用到的几个重要原语：用 generateBinding 工具产生接口定义中的 wsdl: binding 和 wsdl: service。用 generateWSDL 从 java 接口产生 WSDL 文件，用 generateStubs 从 WSDL 文件产生 Stub class。

Eclipse 是新一代 IDE 开发环境，其核心采用了插

件体系结构,Eclipse 的 Java 开发工具(JDT)具有 Code Assist 功能,还集成了 Ant 插件。目前的 a 版本具有以下功能:部分生成 GWSDL 文件和 Java 实施文件(无论扩展 GridServiceImpl 或使用 Operation Providers),Java 向 GWSDL 转换的部分同步,部分提取目录结构并自动生成包结构。自动生成命名空间和包之间的映射文件 N2P,按表单选择自动生成 WSDD 文件,点击按钮生成服务(或只是桩类)的 GAR 文件。

3.4.3 服务数据的定义与实施

服务数据元素:(用 XML 表示的)结构化数据,定义方式有两种,可在 GWSDL 文件中的 <type> 标签内定义;也可先在 XSD(模式)文件中定义,XSD 文件中的基本数据类型会在同一文件内定义其他服务数据类型时被引用,最后要在 GWSDL 中引用。

如何为网格服务添加 SDE 呢?首先,是接口问题,要加新目标命名空间,另外 2 个命名空间是 SDE 类型和 OGSi 命名空间(服务数据有关的定义)。其次,引进描述数据类型的 schema 文件,注意命名空间与 XSD 文件中目标命名空间的对应关系。最后,在接口标签内增加新标签 <sd: servicedata>,指明 SDE 的性质。具有复杂数据类型的 SDE,要用 SDE javabean(ant 用 XSD 文件自动生成)与之交互。

本系统中的服务数据是状态数据即 FreeCPU、FreeMemory 和 FreeDiskVolume,都是整型。

3.4.4 GWSDL、N2P、JAVA 和 WSDD 文件之间的关系解析

在编写网格服务程序中,最容易混淆的是 GWSDL、N2P、JAVA 和 WSDD 等文件之间的关系,因此有必要做详细的分析。

XSD 文件和 GWSDL 文件的关系:如何把服务数据与服务相联系呢?GWSDL 文件首先要引入 XSD 文件,然后用 cardinality 关系把 SDE 与接口相关联。

在 GWSDL 文件中,标签 <type> 也是模式数据,它定义了 message 的元数据;而 message 定义了 gwsdl:portType 的元数据。具体是:接口内有操作,操作内有输入、输出和缺省标签,每个标签有一个消息属性,在消息属性中,除了缺省的以外,都要定义:用 message 中的 part 标签定义,而 part 标签中属性 element 的值要在 type 标签中定义。

GWSDL 文件的 gwsdl:portType name = 接口名称

(定义),。definitions name = 服务名称(定义),以便部署文件 WSDD 引用即 service name = 目录名称/服务名称。

命名空间映射文件 N2P 定义了 GWSDL 命名空间与 JAVA package 之间的映射关系。GWSDL 是语言中立的,要转换成 WSDL 文件,再用工具从 WSDL 文件生成 stub classes。具体语言通过 stub classes 访问服务接口,但 stub classes 放在那里要由映射文件 N2P 定义。总之,该文件要具体映射接口、服务数据、绑定、服务与 JAVA package 之间的关系。

在 WSDD 文件中,service 的属性 name 是服务的位置(相对目录),className 是接口或接口所在的类,baseClassName 是服务实施的类或 GridServiceImpl,schamePath 是 WSDL 文件的位置,operationProviders 是服务提供者的类名。另外一个重要参数是 service-Config,指定了索引服务配置文件的的路径,后者用于聚合网格服务实例并部署到不同的节点即计算机上。

4 GT 发展与挑战

OGSA 定义的网格服务包括了资源,并且是有状态的,这违背了服务是无状态的自然规律。基于新技术 WS-Address 的 WS-Resource Framework 标准的问世,成功地分离了服务和资源,统一了 WS 和网格服务的标准。基于 WS-Resource Framework 的 GT4 即将问世,它将具有新的编程模型,势必会对网格开发者和开发工具开发者提出新的挑战,我们拭目以待。

参考文献

- 1 Globus Toolkit 3.0, Java Programmer's Guide Core Framework [EB/OL]. http://www-unix.globus.org/toolkit/3.0/ogsa/docs/java_programmers_guide.html, 2004-05-21/2004-07-05.
- 2 Borja Sotomayor. The Globus Toolkit 3 Programmer's Tutorial [EB/OL]. http://www.casa-sotomayor.net/gt3-tutorial/download/progtutorial-pdf_0.4.tar.gz, 2004-05-21/2004-07-05.
- 3 Borja Sotomayor. A Globus Toolkit Plug-in for Eclipse [EB/OL]. <http://gt3ide.sourceforge.net/>, 2004-05-21/2004-07-05.