

基于 Web 分布式 workflow 系统可扩展性的研究与实现

祝永志

(曲阜师范大学计算机科学学院 日照 276826)

摘要 今天,传统的工作流技术已不能满足业务过程重建和自动化的日益增长的需求。本文给出了一个基于 Web 的具有高可扩展性的分布式工作流系统并给出它的设计原理和系统结构。

关键词 Web 服务,工作流,分布式系统,J2EE,可扩展性

Implementation and Examination of Distributed Workflow System with High Scalability Based on Web

ZHU Yong-Zhi

(College of Computer Science, Qufu Normal University, Rizhao 276826)

Abstract Today, tradition workflow technology can't meet the increasing needs of organizational process reengineering and automatization. This paper proposes a distributed workflow system with high scalability based on Web and gives the design principle and total structures of the system.

Keywords Web service, Workflow, Distributed system, J2EE, Scalability

1 引言

今天,商业间的竞争日趋激烈,很多公司和组织为适应变化的商业环境,不得不利用 IT 技术以支持实现的业务流程处理,这就是工作流管理系统。近年来,传统的集中式工作流控制结构,尽管其体系结构简单,但只适合于规模较小的组织机构,不能满足企业的需求。随着 Internet 的飞速发展,各企业纷纷建立自己的分布式计算环境,以使日常业务流程自动化。分布式执行结构使系统具有良好的可靠性和可扩展性,尤其适合大型企业、组织的信息协作处理。

本文给出一种基于 Web 分布式工作流管理系统结构,并对其可扩展性进行了研究,采用 J2EE 与 Web Service 技术给出了一套具体实施方案。

2 基于 Web 工作流管理系统的设计思想

Web 服务是一类可通过开放的 Internet 协议访问的软件组件,是组件技术和 Web 技术的结合。在传统的组件技术中,一个服务是一个可以通过编程接口访问的应用,客户通过紧耦合的分布式计算协议,如 DCOM、CORBA 和 RMI 等访问这些服务。这种紧耦合式的体系结构由于其自身的私有格式,尽管效率较高,但限制了服务的重要性和灵活性。

为了克服这些局限性,Web 服务协议采用了开放的体系结构。客户和服务之间通过标准的 Web 协议,如 HTTP 或 SMTP 和 XML 进行通讯,这些协议是完全独立于平台和实现,但由于 Web 协议和 XML 侧重于底层的数据传输,因此为实现 Web 服务体系中应用的互操作,需要在不同的层面上建立起一系列的标准。如一个 Web 服务是一个具有 Web 服务接口、采用 SOAP(简单对象访问协议)通讯的应用,该 Web 服务接口必须能够解释客户发来的 SOAP 请求,并且调用适当的应用逻辑来处理这个请求、返回结果。

Web 服务的实现部分可以运行于任何地点,可以采用任

何平台、任何编程语言、任何体系结构,可以是一个专门的 Web 服务应用,也可以是包装成为 Web 服务的已有应用(legacy application)。而服务的客户,也可以采用任何平台、任何语言构筑这些请求,完成对 Web 服务的访问。

3 基于 Web 工作流管理系统体系结构

3.1 系统实现方法

本文采用三层体系结构:客户层、事物处理服务层、数据库服务层。

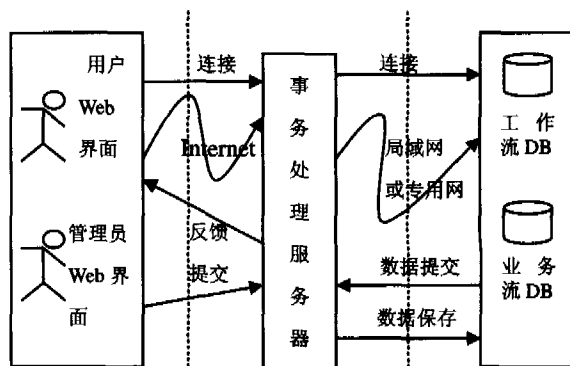


图 1 基于 Web 分布式工作流管理系统结构

图 1 采用 Internet 方式实现无地域限制,充分体现敏捷制造的优势。尽管与传统的 C/S(客户机/服务器)方式比速度较慢,但随着 Internet 的发展,基于 Web 方式是工作流管理的必然选择。

事物处理服务层作为 C/S 的中间层将原属于客户机的部分功能独立出来,达到瘦客户机目的。这有利于资料和功能集中,将资料和功能进行封装,使系统运行更加安全。

以往的分布式工作流管理系统多采用 CORBA 或 DCOM 技术实现,在 Web Service 和 J2EE 广泛使用的今天,CORBA

和 DCOM 技术已不能很好地满足企业业务过程网络化需求。J2EE 作为一个标准的体系结构,为使用 Java 进行基于 Web 的企业应用的开发与部署提供了无与伦比的支持。

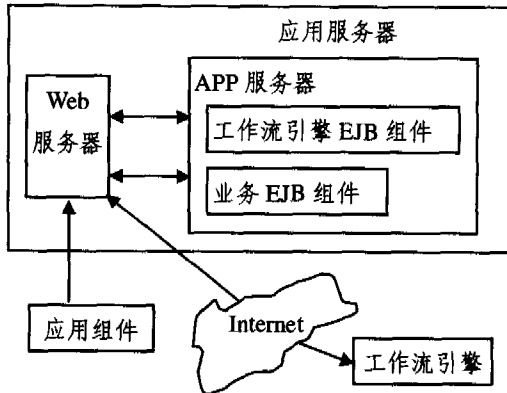


图2 事务处理服务层结构

事务处理服务层包括 Web Server 层和 App Server 层,还包括四个逻辑部件:Web 组件、APP 组件、应用组件、远程服务组件。Web 组件负责与客户交互,在需要的时候产生对 APP 组件、应用组件、远程服务组件的调用并对这些调用进行管理。APP 组件、应用组件、远程服务组件负责实际的工作执行任务并在需要的时候与数据库交互。

数据服务层主要与 APP 组件交互,提供必要的数据服务。该层主要有数据流数据库、业务数据库,分别通过工作流引擎 EJB 组件和业务 EJB 组件与应用服务器交互数据。可通过选择 ODBC(开放数据库互连)驱动程序版本和种类支持数据库系统的不断升级和更换,以提高系统性能。

EJB 组件是企业级应用设计的 Java 组件类型,是基于标准分布式对象技术、CORBA 和 RMI 的服务器端 Java 组件。EJB 组件是分布式的,提供了应用的商务逻辑部分。由于它们不涉及表示层的问题,因此必须与其它显示技术(如 Servlets),服务于 HTML 客户端的 JSP 技术,或者使用了诸如 ATW、Swing 技术的 Java 应用起作用。

按照 EJB 规范的定义,EJB 组件由 Home 接口、Remote 接口、组件类和部署描述文件构成。如欲创建一个用于计算利率的有状态会话(Stateless SessionBean)类型 EJB 组件,可将组件命名为“CalsMoney”,计算利率 EJB 组件对象命名为“CalcMoneyEJB”。

EJB 组件的 Home 接口用于定义创建、查找和删除组件对象实例的方法,有状态会话类型 EJB 组件 CalcMoneyEJB 的 Home 接口定义代码:

```
Public interface CalcMoneyHome extends EJBHome
{
Public CalcMoney create() throws CreateException, RemoteException;
Public CalcMoney create(int nRate) throws CreateException, RemoteException;
Public void remove() throw RemoteException;
}
```

在上述的描述中,EJB 组件的 Home 接口继承 EJB 接口,在该接口包括两个异构的创建组件对象实例的 create 方法,两个方法均返回该 EJB 组件的 remote 接口对象。在 Home 接口中还定义了用于 EJB 容器的组件实例池中删除组件对象的 remove 方法。该方法对应于 EJB 组件类中定义的 ej-

bRemove 方法。

EJB 组件的 Remote 接口用于定义客户端应用程序能够调用的组件商务方法。与 Java 接口定义相似,在 Remote 接口中定义的商务方法只是一个架构而没有具体的实现代码。有状态会话类型 EJB 组件 CalcMoneyEJB 的 Remote 接口定义如下:

```
Public interface calcMoney extends EJBObject {
Public float calculateMoney() throws RemoteException;
Public float calculateTax(float fRate, float fMoney) throws RemoteException;
}
```

EJB 组件 Remote 接口继承定义在 javax. ejb 包中的 EJBObject 接口。在 Remote 接口中定义了两个名称分别为 calculateMoney 和 calculateTax 的方法。两个方法均抛出 RemoteException 类型异常。由 Remote 接口中定义的方法能够抛出的异常类型可以看出 Remote 接口同样是对 Java RMI-IIOP 协议有效的远程接口。

EJB 组件类用于定义两种类型方法的实现代码:一种是 EJB 组件的生命期方法,另一种是 EJB 组件的 Remote 接口中定义的商务方法。会话类型 EJB 组件对象的组件类型实现 SessionBean 接口,实现类型 EJB 组件实现 EntityBean 接口,消息驱动类型 EJB 组件实现 MessageDrivenBean 和 MessageListener 接口(描述代码略)。

部署描述文件是 EJB 组件的组装者和部署者应用的 XML 类型文档,其主要作用是声明 EJB 组件的构成、部署过程中需要 EJB 容器提供的服务以及对于系统资源的要求等信息(描述代码略)。

3.2 分布式引擎调度策略

本系统中,没有一个中心结点负责整体流程的调度。整个业务流程的调度,实际上是由分布的工作流引擎共同完成的,如图 3。

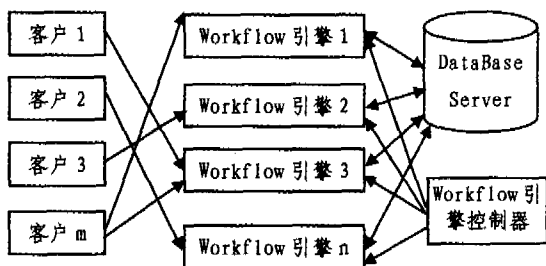


图3 workflow 管理体系结构

所谓 workflow 引擎是指 Workflow 作为应用系统的一部分,并为之提供对各应用系统有决定作用的,根据角色、分工和条件的不同决定信息传递路由、内容等级等核心解决方案。

开发一个系统,最关键的部分不是系统的界面,也不是和数据库之间的信息交换,而是如何依据业务逻辑开发出符合实际需要的程序逻辑并确保其稳定性、易维护性(模块化和结构化)和弹性(容易根据实际业务逻辑的变化作出程序上的变动,如决策权的改变、组织结构的变动和由于业务方向的变化产生的全新业务逻辑等)。

Workflow 引擎解决的就是这个问题;如果应用程序缺乏强大的逻辑层,势必变得容易出错(信息的路由错误、死循环等)。

本系统中每一个 workflow 引擎主要负责:解释过程定义、管

理活动实例信息、执行分布活动和同其它引擎通信。具体实现方法如下:

① 引擎控制器启动引擎:

(1) 当启动引擎收到客户发来的启动引擎请求后,通过 SOAP(简单对象访问协议)消息层,向过程定义存储服务发出请求,得到当前进程定义;

(2) 得到过程定义以后,启动引擎会依据定义中的分解消息,将该过程定义分解为一个节点的配置消息;

(3) 启动引擎将各个节点配置消息通过消息平台发布到参与的工作流引擎中。参与的工作流引擎得到过程配置消息后,将其保存在本地的“局部过程定义存储表”中。

② 实例启动:

启动引擎在发布完配置消息后,会依据过程定义,向第一个活动所在引擎发送数据消息(当前过程的启动参数)和控制消息(过程实例启动命令)。

③ 实例生成:

当引擎收到过程实例控制消息(过程实例启动命令)后,会检查本地实例消息表中该活动实例是否已被创建。若没有,则创建该实例并修改活动实例消息表,进行实例调度。

④ 实例调度:

当引擎接到某过程实例的数据消息和控制消息后,会检查该过程的当前活动的启动条件,即当前活动跟其它活动的数据和控制依赖关系。如果启动条件满足,则会进行活动执行,否则继续等待活动实例消息。

⑤ 活动实例执行:

(1) 根据当前要执行活动消息,分别调用具体的活动执行模块,完成活动执行;

(2) 当一个活动执行完成后,计算该活动的退出条件。如果满足退出条件,则进行活动实例销毁。

⑥ 活动实例销毁:

(1) 根据过程定义消息,得到该活动的后继节点和链接消息;

(2) 向所有的后继节点发送执行当前活动执行完毕的控制消息;

(3) 根据数据链接,向目标节点发送过程数据;

(4) 一旦将当前过程定义在本节点的所有活动都已执行完毕或者收到其它节点传来的过程结束消息,则清除活动实例消息表的当前项目,并进行日志记录。

⑦ 过程结束:

如果当前流程已执行完毕,工作流引擎会向所有参与节点发送过程结束消息,并进行相应的清理工作(流程图略)。

4 系统可扩展性

工作流管理系统(WsMS)从结构上大致分为:集中式 WsMS 与分布式 WsMS。前者是指一个过程中,所有活动的执行都集中在一个“工作流服务器”上完成,最终用户通过工作流客户端界面完成分配给他的任务。其优点是集中调度,各结点之间不直接通讯,而是通过中心结点作为中介,易于实现进程运行中活动之间的协调,易于实现集中的运行记录和

监控。系统整体设计简单,适合小规模应用。但一旦应用规模扩大,就体现出可扩展性差、中心结点负荷过重、健壮性差等不足。

本文提出的基于 Web 的分布式 WsMS,一个过程的运行分为若干阶段,每个阶段的运行都是由分布在网络上的某个处理结点完成,较好地克服了集中式 WsMS 的诸多不足。当分布式环境中的某台处理站点出现差错,仅影响那些流经该结点的过程的运行,不至于整个系统瘫痪。在应用扩展时,只需添加相应的分布式处理站点即可,有很好的可扩展性。

5 系统平台的软件、硬件配置

基于系统的总体设计,本文基于 J2EE 平台,采用 Java、XML 与 Web Service 技术,开发了基于 Web 的分布式工作流管理系统。系统的软件、硬件配置要求如下:

5.1 客户机配置

硬件环境:CPU;PII 300 以上;RAM;64M 以上,建议 128M;能正常运行 Windows 及 IE。

软件环境:Windows 98/XP/NT/2000 简体中文版+简体中文 IE6.0 以上版本。

5.2 事务处理服务器的配置

硬件环境:CPU;PIII 500 以上,建议 PIII 800 以上;RAM;256M 以上,建议 512M;DISK;800M 以上的,可用硬盘安装空间。

软件环境:Windows 2000 Server,Microsoft IIS6.0 以上。

5.3 资料处理服务器的配置

硬件环境:CPU;PIII 500 以上,建议 PIII 800 以上;RAM;256M 以上,建议 512M;DISK;800M 以上的,可用硬盘安装空间。

软件环境:Windows 2000 Server,Windows SQL Server 2000 以上。

结束语 随着应用规模和功能的不增大和调整,企业业务流程日趋复杂和敏捷,基于工作流技术的应用系统必将是未来软件技术的主要发展方向之一。本文通过分析当前工作流系统的一些不足,提出了一种基于 Web 的异构环境下的分布式工作流系统,对其具有的可扩展性进行了阐述。本文采用 J2EE 与 Web Service 技术给出了一套具体实施方案。

参考文献

- 1 刘斌,王兰郡,王浩军. 基于 CORBA 的分布式 Agent 通信构架[J]. 小型微型计算机系统,2001,22(6):728~731
- 2 龚田根,陈云,汪地,等. 基于 Web 分布式数据流管理系统的研究与开发[J]. 机电一体化,2000(4)
- 3 胡华,宋荷庆,王会进. 面向 Web 的工作流管理[J]. 计算机工程与设计,2002(4):31~33
- 4 Workflow Management Coalition (WfMC). WfMC2TC21011v3 192Feb299 Terminology & Glossary [S]. USA: WfMC,1999
- 5 谢杰华,陈震,罗源. 基于 Web 的工作流管理系统[J]. 计算机应用研究,2002(6):141~143
- 6 刘洋,葛声. 一种基于 Web 的分布式工作流系统的研究与实现[J]. 计算机工程与应用,2003(1):2~7