

OFBiz 实体引擎在 MIS 开发中的研究与应用 *

刘 智¹ 周康渠² 胡 勇³ 游中胜⁴

(重庆工学院计算机学院¹ 重庆汽车学院² 重庆 400050)

(重庆交通大学应用技术学院电子信息工程系 重庆 400042)³ (重庆师范大学学报 重庆 400047)⁴

摘要 简要介绍了 OFBiz 框架技术及其应用程序架构的发展,论述了 OFBiz 框架中实体引擎的基本设计思想,分析了 OFBiz 框架开发系统的优点,并结合具体项目实现了 OFBiz 框架技术的应用。实践表明,Ofbiz 框架的使用有效缩短了软件开发周期,开发后的系统具有良好的平台无关性、可扩展性和可移植性。

关键词 OFBiz, 实体引擎, 平台无关性, J2EE

Study and Application of Entity Engine on MIS Development

LIU Zhi¹ ZHOU Kang-Qu² HU Yong³ YOU Zhong-Sheng⁴

(College of Computer Science & Technology¹, Chongqing Institute of Technology, Chongqing 400050)²

(Dept. of Electronic & Information Engineer, Chongqing Jiaotong University, Chongqing 400042)³

(Journal of Chongqing Normal University, Chongqing 400047)⁴

Abstract In this paper, the development of OFBiz framework and its application construction was introduced briefly, And then discusses the basic designing idea of entity engine and analyzes the merits of OFBiz framework in developing MIS. Combined with concrete project, an application based on OFBiz framework is realized. It proves in practice the use of OFBiz framework in MIS developing shortens software developing cycle effectively and the system developed with OFBiz framework has good characteristic of independence, extensibility and transportability.

Keywords OFBiz, Entity engine, Independence, J2EE

1 引言

随着企业信息化的进一步推广,企业应用集成(EAI)所暴露的问题也越来越多,如信息孤岛、新平台不支持原有软件等。其中大部分问题是由于软件的可移植性、可扩充性差,造成了新系统不能在原有系统上进行扩充,原有数据库不能继续使用,从而造成了企业信息化投资的严重浪费,因此开发平台无关性应用程序,提高程序的可移植性是软件行业追求的目标之一。本文根据项目的实际特点,使用 OFBiz 框架设计并实现了某公司网络化物流管理系统,并取得了良好的效果。

2 OFBiz 简介

OFBiz(Open For Business)是一个开源项目^[1,3],它以 MVC 模式^[3,4]搭建而成,提供了创建基于最新 J2EE/XML 规范和技术标准,构建大中型企业级、跨平台、跨数据库、跨应用服务器的多层、分布式电子商务类 Web 应用系统的框架。OFBiz 提供了一整套的开发基于 Java 的 Web 应用程序的组件和工具,包括实体引擎,服务引擎,消息引擎,工作流引擎,规则引擎,应用这些组件及工具能够大大提高软件开发效率,可以大大缩短开发企业级 Web 应用系统的进度和降低成本。同时,OFBiz 所提供的系统框架,是一个基于 J2EE 结构的纯 Java 的应用程序,可以很容易和其他应用框架进行整合,因而具有良好的可扩充性。OFBiz 框架如图 1 所示。

在 OFBiz 各组件中,最底层、最关键的是实体引擎和服务引擎,其他引擎都是基于这两个引擎,是它们的具体应用。服务引擎(Service Engine)^[2]具体负责以合适的方式进行服务的定义、管理和调用,主要用于处理跨平台、跨操作系统、跨

应用系统之间的业务逻辑。实体引擎(Entity Engine)^[2]技术完全以 XML 格式来描述企业的数据库结构,以中间件的形式提供应用程序的 OR Mapping 及持久化功能,因而基于 OFBiz 的应用程序可以做到不同数据库间的无缝移植,具有良好的跨平台性。实体引擎是 OFBiz 的核心,下文主要就实体引擎技术及其数据库独立应用进行讨论。

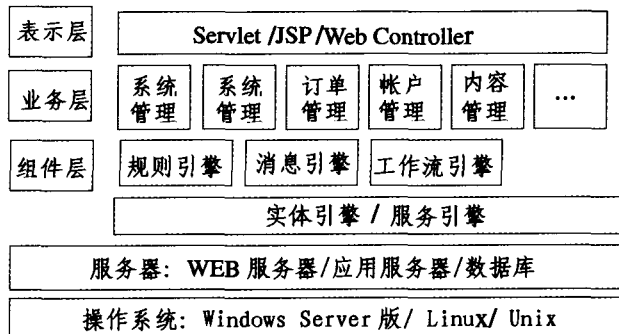


图 1 OFBiz 框架示意图

3 实体引擎技术

目前大多数持久化技术采用的是直接访问数据库的方式,当企业需要更换数据库平台时,必须修改原有应用程序,程序的可维护性不高。而在 OFBiz 中,数据库表创建、对象与数据表的映射、对象的查询等都由实体引擎封装,开发人员只需要在简单的 XML 文件中定义数据库表结构,实体引擎会自动在数据库中建表,并动态生成映射对象,用户在程序中可以只考虑对 Object 的处理,实体引擎会自动通过事务逻辑

* 本课题得到国家 863 计划项目资助,课题号:2004AA414060。刘 智 讲师,硕士,主要方向:数据库、管理信息系统。

更新到数据库中, OFBiz 实体引擎的工作原理如图 2 所示。

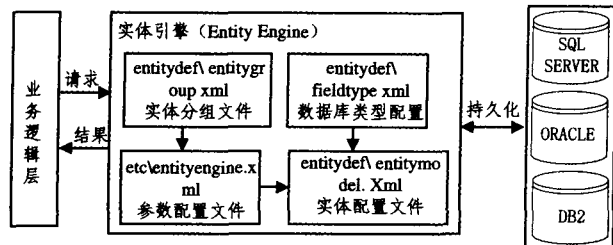


图 2 实体引擎的工作原理

实体引擎接受业务逻辑层发出的数据处理请求, 根据实体引擎配置文件和请求的信息自动地寻找相应的数据库执行持久化操作并返回操作数据到业务逻辑层。事实上, 实体引擎主要是提供了一组工具和设计模式来对现实世界中特定的实体(数据对象)进行建模和管理, 一个实体就是一个由多个数据域(fields)和该实体与其它实体之间的关系所组成的一个数据对象。实体引擎的一个主要目标是尽可能地提供一种通用的代码结构, 来消除在针对每一个实体的事物处理过程中, 所有写死(hard code)的代码。实体引擎技术主要包括以下几个要点:

1) 使用 XML 建模实体模型。实体模型的定义在 \$OFBIZ_HOME/commonapp/entitydef/目录。传统数据库中的表、视图、表之间的联系、主外键等各种约束都可以在实体模型中一一对应。有了实体模型后, 不需要编写任何 SQL 脚本, 实体引擎(entity engine)将自动建立数据库表和记录。当需要对实体做修改时, 只需修改实体模型就行了, 由实体引擎(entity engine)自动对数据库进行检查, 并对数据库结构进行相应的修改。图 3 是 OFBiz 中订单管理实体文件中的一个片断。

2) 定义通用类型。不同的数据库支持的数据类型(SQL 类型)可能会有所不同, 实体引擎通过使用中间类型做到数据库独立, 应用程序中的数据类型(JAVA 类型)和数据库中的数据类型(SQL 类型)通过中间类型进行统一。图 4 是针对 Oracle 数据库类型定义的一个通用类型片断。

```
<entitymodel>
  <! - 实体定义 ->
  <entity entity-name="OrderItem" package-name="org. ofbiz.
    commonapp. order. order"
    never-cache="true" title="Order Item Entity">
    <field name="orderId" type="id-ne"></field>
    .....
    <prim-key field="orderId"/>
    <prim-key field="orderItemSeqId"/>
    <relation type="one" fk-name="ORDER_ITEM_ORTYP"
    rel-entity-name="OrderItemType">
      <key-map field-name="orderItemSeqId"/>
    </relation>
  </entity>
  .....
</entitymodel>
```

图 3 OFBiz 订单管理实体文件片断

```
<fieldtypemodel>
  <! - General Types ->
  <field-type-def type="blob" sql-type="LONG RAW"
    java-type="java. lang. Object"></field-type-def>
  .....
  <! - Specialized Types ->
  <field-type-def type="url" sql-type="VARCHAR2(255)" java-
    type="String"></field-type-def>
  .....
</fieldtypemodel>
```

图 4 实体引擎中 Oracle 数据库通用类型定义片断

3) 定义 entitygroup.xml 文件, entitygroup.xml 主要功能

是将应用程序的实体进行分组, 从而可以在一个应用程序的不同模块使用不同的数据库平台, 或者用于处理地理上分布的数据库间的数据共享, 如图 5 所示。

```
<entitygroup>
  <entity-group group="org. ofbiz. commonapp" entity="Order-
  Header" />
  <entity-group group="org. ofbiz. commonapp" entity="Order-
  HeaderAndItems" />
  .....
</entitygroup>
```

图 5 entitygroup.xml 文件片断

4) 修改配置文件 entityengine.xml, 包括各种通用类型定义文件名称、实体文件名称、数据库连接、分组连接、用户名、密码等, 如图 6 所示。

```
<entity-config>
  .....
  <delegator name="default" entity-model-reader="main" entity-
  group-reader="main" entity-eca-reader="main" distributed-cache-
  clear-enabled="false">
    <group-map group-name="org. ofbiz. commonapp" data-
    source-name="localoracle"/>
    </delegator>
    <entity-model-reader name="main">
      <resource loader="mainfile" location="entitymodel.
      xml"/>
      <resource loader="mainfile" location="entitymodel_
      content.xml"/>
    </entity-model-reader>
    <entity-group-reader name="main" loader="mainfile" location
    ="entitygroup.xml"/>
    <field-type name="oracle" loader="mainfile" location="field-
    typeoracle.xml"/>
    <datasource name="localoracle"
      helper-class="org. ofbiz. core. entity. GenericHelperD-
      AO"
      field-type-name="oracle"
      check-on-start="true"
      add-missing-on-start="true"
      join-style="theta-oracle">
      <sql-load-path path="commonapp/db" prepend-env="of-
      biz. home"/>
    </datasource>
    .....
  </entity-config>
```

图 6 实体引擎配置文件片断

4 基于 OFBiz 框架的应用设计实例

本文以“网络化物流管理系统”项目中的软件为例, 来说明如何设计基于 OFBiz 框架的 Web 应用。网络化物流管理系统是基于重庆制造业信息化 ASP(Application Service Provider)平台为依托, 以某企业建筑公司为基础, 对公司的材料及设备的采购、库存、调拨、租赁进行全面管理的系统。根据项目需求分析, 确定该项目必须具有的特点是:

1) 平台无关, 具有良好的可移植性, 便于系统在不同环境间的移植, 包括不同的操作系统平台和不同的数据库平台;

2) 具有良好的可扩展性, 为企业以后系统功能的扩充提供接口;

3) 良好的可维护性, 系统投入使用后, 主要是由管理员承担系统维护的工作, 维护人员可能不定期变动, 这就要求系统的可维护性强。

基于项目以上特点及要求, 并且考虑到项目的开发周期短的因素, 选用 J2EE 开发环境, ORACLE 数据库平台, 并选用 OFBiz 作为系统开发框架。本文主要关注网络化物流管理系统实现的实体引擎配置部分:

1) 建立 ORACLE 数据库类型配置文件 fieldtypeoracle.xml。OFBiz 框架已经提供了 ORACLE, SQL SERVER,

(下转封三)

主机做了子进程的重复工作,这是为了减少通信量)。

算法的通信量主要有以下几部分开销:主机一次性广播离散数据给每个处理机,然后分别发送不同的函数模型,处理机每拟合一个模型后把残差反馈给主机。

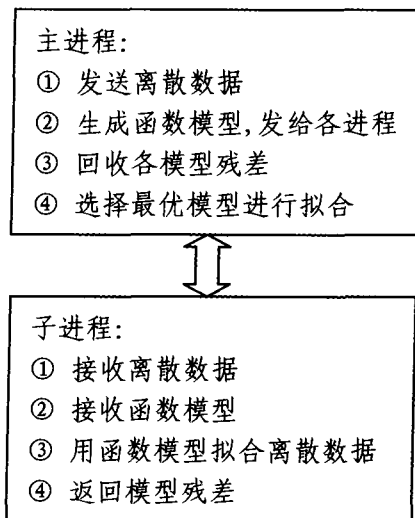


图1 并行算法工作图

从表2可以得到,三个最佳平方逼近函数中,残差的平方和最小者为指数函数,因此,主机在这三个函数模型中选择指数模型,得到最佳逼近函数 $f(x)=5.636e^{-2.891x}$ 。

当然,如果函数模型再多一些,可选择的函数模型就更广一些,也就可能得到更精确的逼近函数。

每个处理机上处理一个函数模型时,可调用现成的拟合软件,比如 Matlab 就是比较理想的拟合平台。本文并行算法实验使用了 MPI 编程调用 Matlab。

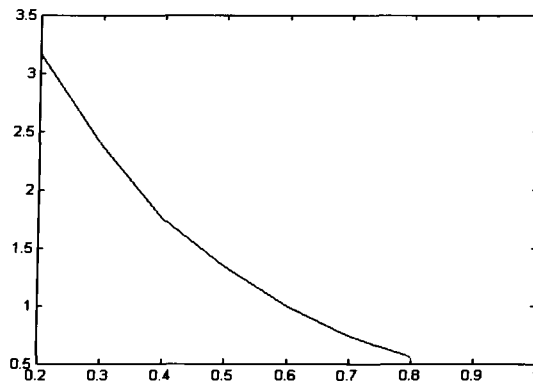


图2 离散数据的折线图

4 最优数据拟合实例

有离散数据如表1^[3]:

表1 离散数据表

x_i	0.2	0.3	0.4	0.5	0.6	0.7	0.8
y_i	3.16	2.38	1.75	1.34	1.00	0.74	0.56

把这些点依次连接起来如图2所示。

从图2的形态看,可以视为二次函数或指数函数;如果考虑数据的误差,将其视为直线也无妨。这里讨论函数形态只是为了尽量减少函数模型的数量,在实际应用时完全没有必要人为限制函数模型的范围,可以盲目搜索,但花费更多的CPU时间是盲目搜索的代价。

这三个函数模型由三个处理机都使用最佳平方逼近拟合这些数据,结果如表2。

表2 三种函数拟合的比较

函数模型	最佳平方逼近函数 $f(x)$	残差的平方和
一次函数 $y=ax+b$	$f(x)=-4.225x+3.674$	0.3047
二次函数 $y=ax^2+bx+c$	$f(x)=5.94x^2-10.17x+4.921$	8.3×10^{-3}
指数函数 $y=ae^{bx}$	$f(x)=5.636e^{-2.891x}$	8.9447×10^{-4}

参考文献

- 1 章勤,李品,肖舸.海量数据的曲线拟合并行算法及实现[J].华中理工大学学报,2000,28(10):85~87
- 2 翟振华,李凯.神经网络在函数逼近问题中的应用研究[J].计算机工程,2001(5)
- 3 丁丽娟.数值计算方法[M].北京:北京理工大学出版社,2001
- 4 都志辉,等.高性能计算并行编程技术[M].北京:清华大学出版社,2001

(上接第293页)

DB2,MYSQL等常用数据库平台的通用类型定义文件;

2)按系统模块划分,用5个实体配置文件 entitymodel_stock.xml、entitymodel_store.xml、entitymodel_dispatch.xml、entitymodel_lease.xml、entitymodel_other.xml 分别来描写采购、库存、调拨、租赁四个模块以及其他的一些实体;

3)根据实体配置文件,对 entitygroup.xml 文件进行相应的修改,即添加新定义的实体到 entitygroup.xml 文件;

4)配置 entityengine.xml 文件,主要是实体文件名称和数据库连接串参数,包括数据库访问地址,数据库名称,用户名,密码等。

在做好以上配置后,在以后的编程中就可以直接访问实体配置文件进行数据的访问,而如何完成这一过程则由实体引擎处理。特别是当用户的数据模型需要修改时,需要修改的也仅仅是实体配置文件,这就极大地减轻了程序的维护工作量。

结束语 OFBiz 框架是一种非常优秀的基于 J2EE 的 MVC 应用框架,尽管它试图把所有的基础、公共和特殊应用都涵盖进来,增加了其复杂性,阻碍了 OFBiz 的应用,但其实体引擎的思想及其所带来的良好的异构数据库间的可移植性仍然值得系统分析设计人员的学习。本文在总结了 OFBiz 框架技术及其实体引擎工作原理的基础上,结合“网络化物流管理系统”项目,真正实现了平台无关应用程序的构建,实践证明取得了良好的效果。

参考文献

- 1 <http://sourceforge.net/project/ofbiz>
- 2 <http://incubator.apache.org/ofbiz/>
- 3 冷山述,余一,陆侗.企业信息化框架 OFBiz 的研究.北京机械工业学院学报,2004(3)
- 4 王晓楠.MVC 的设计与实现.计算机系统应用,2004(3)