

# 一种新的网格环境模型——TGrid Model

齐德昱 林伟伟

(华南理工大学计算机科学与工程学院 广州 510640)

**摘要** 在分析了现有网格环境不足的基础上,提出一种新的网格环境模型——基于树形结构的网格体系与环境 TGrid,支持高性能计算,面向主题的资源共享和新一代的需求建模。它以树结构来组织网格节点和集成各种资源,实现了自底向上、多级、面向需求的资源抽象和多种资源融合。而且树型结构符合自然层次组织关系,容易实现网格系统的层次化管理,有利于减轻中心节点的负载和实现大规模应用的负载平衡,提高资源查找效率。同时,TGrid以虚拟资源的形式实现网格资源的共享,利用分布式 JVM(TJVM)虚拟网格节点上 CPU 和主存资源,利用多数据库中间件(TDOD)实现数据库级资源集成和共享,利用 Globus 网格服务(GService)实现其他软件和数据资源共享。该树型网格为日益增长的网格应用的需求提供了新的解决方案。

**关键词** 树型网格,虚拟资源,抽象,树

## TGrid Model: A New Grid Model

QI De-Yu LIN Wei-Wei

(College of Computer Engineering and Science, South China Univ. of Tech., Guangzhou 510640)

**Abstract** After analyzing the problems in current grid environments, a new tree-based grid model (TGrid) is presented in this paper. TGrid supports high performance computing, subject-oriented resources sharing and the next generation requirement modeling and it also achieves bottom-up, multi-level and demand-oriented abstraction and fusion of multiple resources. And it organizes grid nodes and integrates multiple resources in the tree structure that conforms to the natural hierarchical organization structures, easily achieves hierarchical management, also may reduce load on the central node (server) and realize load balancing for large-scale applications, improve the efficiency of resource searching. Meanwhile, resource sharing in TGrid is realized by sharing virtual resources. Virtual resources on each node may be TJVM (Tree-based distributed JVM) which virtualizes CPU and memory resources, or may be TDOD (Tree-based DOD, DOD is a multi-database middleware, which are also called "DBMS" over DBMSs) which realizes resources sharing and integration at the database level, or may be GService (Globus grid service) which encapsulates other software and data resources. TGrid provides a new solution for Grid application demands.

**Keywords** TGrid, Virtual resource, Tree

## 1 引言

随着网络技术的快速发展,网格技术已经成为当前的研究热点。网格的最终目标就是把整个因特网整合成一台巨大的“虚拟超级计算机”<sup>[1]</sup>,实现计算资源、存储资源、数据资源、信息资源、知识资源、专家资源的全面共享和协同。要构成虚拟的超级计算机的网格计算环境,其中网格体系结构是最重要的基础问题。

目前,网格的体系结构主要有两大类:五层沙漏结构和开放网格服务体系结构(OGSA)<sup>[2~5]</sup>。基于协议的五层沙漏体系结构将在异构、动态、跨管理域的分布式环境下共享各种资源的功能进行了分解,将其分别放在构造层、连接层、资源层、汇聚层、应用层中。这种体系结构的缺点是:1)针对每一种平台都需要实现沙漏瓶颈部分的连接层和资源层,要处理大量与平台相关的事务;2)五层沙漏结构的支撑系统 Globus Toolkit2<sup>[5]</sup>也只是提供了类 Unix 平台实现,且其整体结构不强,其使用的协议也未标准化或离标准化还有一个漫长的过程。OGSA 是基于服务的体系结构,在 OGSA 中,把各种计算资

源、存储资源、网络、程序、数据库等所有资源都包装成网格服务。从资源到服务,这种抽象,将资源、信息、数据等统一起来,隐藏了各种资源的异构性。网格应用要通过调用具体服务实现,而这些具体的服务都必须由用户在已有的服务的基础上根据不同需求编写。目前,OGSA 结构的支撑系统主要有 Globus Toolkits 3 和 Globus Toolkits 4<sup>[6]</sup>,虽然在标准和使用者上已经有很大进步,但要实现具体的应用还需复杂的编程工作。

此外,还有织女星网格的体系结构<sup>[7]</sup>、支持 e-Science 的网格体系结构<sup>[8]</sup>、GridLab<sup>[9]</sup>、Rajkumar Buyya 提出了经济网格及计算经济网格体系结构模型<sup>[10]</sup>、A 3-tier Grid Architecture<sup>[11]</sup>、Green Grid Architecture<sup>[12]</sup>、制造网格体系结构<sup>[13]</sup>等,这些网格体系结构大部分都基于层次化方法,其本质具有较大的一致性,主要是以 Globus 中间件做为中间层,增加特定的功能模块,以实现某一应用领域的特定应用或方便用户使用网格中间件 Globus 而提出来的。

网格体系结构研究是当前投入比较多的领域,但目前它仍然有很多不完善的地方。

\* 粤港关键领域重点突破项目(2005A10307007)、广东省重大科技专项(2003A1030404)、广东省自然科学基金(05300200)。齐德昱 教授,博士生导师,主要研究方向为网格计算、计算机体系结构等;林伟伟 博士研究生,主要研究方向为网格技术、计算机体系结构。

1)没有考虑网络拓扑:地理上分布的各个资源节点(网络节点)以什么样网络拓扑来连通才能更好地实现资源共享和协同计算等;

2)应用开发困难:大都是针对专业用户的,需要使用网络的编程语言进行复杂的编程工作,一般用户难以使用;

3)标准化程度低、抽象性不足:五层结构中,中间小,通用性较强,但两头大,是面向具体问题的,对不同的网格计算问题,构成也不同,非标准化的成分太多;在 OGSA 中,具体应用的实现是服务的集成(分布对象系统),服务的管理与互操作是标准化的,但具体的服务都是由用户按规范根据具体应用提供的,因而 OGSA 抽象出来的标准化成分太少;

4)管理问题:网格中关键节点的管理复杂,难以构造大型网格应用;

5)与网格最终目标有差距:无论是五层沙漏还是 OGSA 的网格体系结构,都距“虚拟的超级计算机”要求很远。

针对上述问题,本文提出一种基于树型网络拓扑的网格环境 TGrid。它以分层树型结构来组织各网格节点,从网络拓扑上来解决大型复杂应用负载均衡问题和保证资源的快速定位,这是当前网格体系结构最欠缺的。而且,在提出的树型网格环境中,以层次化抽象资源来实现面向主题的资源共享,并对网格节点、各种网格资源进行高度抽象和概括,然后把对任何节点资源的任何操作统一为读 Read()和写 Write()访问接口。这也正是网格本意:将分布在网络中的各种资源组织起来形成一个透明的超级计算机系统。

## 2 树型网络的概念模型

### 2.1 树型网络的思想

本文提出的树型网格环境,重点解决网格的资源拓扑和组成、大型任务的高效透明执行、系统层次化管理等问题,这也是目前流行的五层沙漏结构和 OGSA 的欠缺方面,并且使该体系结构下的网格可为上层网格计算应用提供通用易用的、结构动态的、界面单一的具有透明的超级分布计算能力的计算与资源共享环境。

树型网格以树结构来组织网格节点,并且利用分布式 JVM(TJVM)虚拟网格节点上 CPU 和主存资源,利用多数据库中间件(TDOD)实现数据库级资源集成和共享,利用 Globus 网格服务(GService)实现其他软件和数据资源共享,这样可以实现不同类型资源和面向主题资源的共享,并对网格节点、各种网格资源进行高度抽象和概括,然后把对任何节点资源的任何操作统一为读 Read()和写 Write()访问接口。其中 TJVM 是我们正在研发的分布式 JVM 项目,该项目的目标是提供一种新的面向 Java 的透明并行和分布式计算环境, TJVM 作为树型网格中的计算资源模型,用来实现计算机 CPU、主存等资源的虚拟。TDOD 多数据库是我们已经研发并实现的多数据库(DoD)中间件产品,它用来实现树型网格的分布异构多数据库的集成与共享。还有其他软件资源等可以通过使用 Globus 网格服务来封装,以实现软件资源和其他资源的共享。

### 2.2 树型网络的形式化描述

树型网格模型是基于树结构的网格资源集成模型。网格节点上资源的共享以虚拟资源的形式实现,网格中集成多种虚拟资源,为满足不同需要的资源共享和实现。以树结构来组织网格节点,以实现资源高效查找和层次化管理体系。

为了明确给出树型网格模型,首先需要相关的定义。

**定义 1** 网格节点是构成网格系统的节点。在树型网格中节点的角色有三种:资源节点、管理节点、传递节点。节点可以是其中之一或同时有几个角色。

**定义 2** 资源节点是存在若干物理资源和逻辑资源的网格节点。网格系统中存在各种各样的物理资源,包括计算资源、存储资源、数据资源、信息资源、软件资源、通信资源、知识资源、专家资源等。

**定义 3** 管理节点是负责管理其儿子节点的网格节点。

**定义 4** 传递节点是负责节点间信息传递的网格节点。

**定义 5** 虚拟资源是用来实现网格节点上的物理资源和逻辑资源的虚拟的,以便实现全面资源共享。目前我们给出的虚拟资源主要有 3 种,它们分别是 TJVM、TDOD 和 GService。

下面可以定义树型网格及得出相关定理。

**定义 6** 树型网格可以定义为一棵有很多叉树。用一个加权无向图表示,即  $G=(V, E)$ ,其中  $|V|=n$ ,为树中节点的个数,  $|E|=m$ ,为树中边的个数。 $V$  是网格节点的集合,网格  $G$  包含  $n$  个结点  $v_1, v_2, \dots, v_n$ ,  $E$  是网络连接的集合。在这棵树中,叶子节点通常是资源节点,非叶子节点通常是管理节点,某些非叶子节点也可能是传递节点,有些节点可能具有有几个角色。为了实现各种异构物理资源的共享,在树型网格中的每个资源节点中引入虚拟资源,以虚拟资源的形式实现资源共享。

**公理 1** 若网格  $G$  的物理资源集合表示为  $R$ ,  $r$  为  $R$  中的元素,资源  $r$  在网格节点  $v$  的关系表示为  $at(v, r)$ , 则

$$\forall r (r \in R \rightarrow \exists v_i (at(v_i, r))), \text{其中 } 1 \leq i \leq n.$$

**公理 2** 网格  $G$  的虚拟资源集合表示为  $A = \bigcup_{i=1}^k A_i$ , 树型网格中存在  $k$  种用来虚拟物理资源的虚拟资源,每种虚拟资源表示为  $A_i$ , 则

$A_1, A_2, \dots, A_k$  是资源  $A$  的一个划分,即对于任意  $1 \leq i, j \leq k$ ,

$$A_i \cap A_j = \Phi$$

**推论 1** 设  $u = |A|$  及  $u_i = |A_i|$ ,

$$\text{则 } u = \sum_{i=1}^k |V_i| = \sum_{i=1}^k u_i$$

因此,可将网格  $G$  的全部虚拟资源记为

$$A = \{a_1, a_2, \dots, a_u\}$$

**公理 3** 对任意虚拟资源  $a_j$ , 都能映射到相应的物理资源上,即存在 1 到多映射  $f_j: \{a_j\} \rightarrow R_j$ , 其中  $R_j \neq \Phi$  且  $R_j \subseteq R$ 。

公理 3 说明了对任何网格虚拟资源的访问最终转换为对网格物理资源的访问,也就是说我们如果需实现网格物理资源的共享即需实现网格虚拟资源的共享。

**定义 7** 虚拟资源树是由多个虚拟资源形成的一棵树,各虚拟资源所在的网格节点间形成一个网络通信拓扑树。由于一个网格节点上可能存在多个虚拟资源,所以同一个网格节点可以是多个虚拟资源树中的节点。

**定义 8(TJVM 树)** 设谓词 Node( $x$ )表示节点  $x$  是树节点, Father( $x, y$ )表示节点  $x$  是  $y$  的父亲节点, Brother( $x, y$ )表示节点  $x$  与  $y$  是兄弟, TS( $x$ )表示节点  $x$  上的 TJVM 任务集,中缀谓词“ $\in$ ”是集合的属于关系,则一棵 TJVM 树是这样一棵有限多叉虚拟资源树。它的每个节点是一个资源节点(TJVM),每个节点上至少附着一个 TJVM 任务集,中间节点也是传递节点,收集儿子节点的执行结果,其中总根节点也是管理节点,还负责 Java 程序分解,分解结果为 TJVM 任务集,

且满足如下谓词:

- a)  $\forall x (Node(x) \rightarrow \forall y (Node(y) \wedge Father(x, y) \rightarrow \forall t (t \in TS(y) \rightarrow t \in TS(x)))) \dots\dots\dots$  内射规则
- b)  $\forall x \forall y (Node(x) \wedge Node(y) \wedge Brother(x, y) \rightarrow \forall t (t \in TS(x) \rightarrow t \in TS(y))) \dots\dots\dots$  迁移唯一规则

**定义 9** GService 虚拟资源树是由多个具有虚拟资源 GService 的网格节点形成的一棵虚拟资源树。它有助于实现服务资源的快速查找和有效分配。

### 3 树型网格的资源抽象模型

树型网格的抽象模型是指对树型网格的节点、资源和资源操作等的抽象描述。由前面的概念模型部分的分析,可以得出树型网格的资源抽象层次树图(如图 1)。

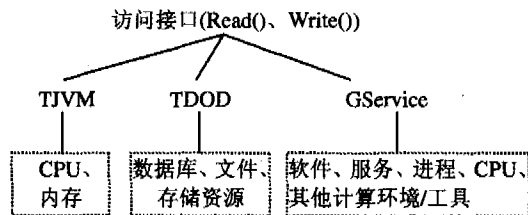


图 1 资源抽象层次树

为了更加清晰描述树型网格模型,下面用类 Java 语法给出树型网格的节点对象、资源对象、管理接口、传递接口的抽象描述。首先描述了网格节点的抽象接口 InterGNodeBase,然后在网格节点的抽象接口基础上定义网格节点对象类 GNode,再定义用来抽象网格节点上不同资源的三个虚拟资源对象类,包括 TJvmNode 类、TDodNode 类、TGlobusNode 类,对于资源节点上必然存在至少一种虚拟资源对象类。而且把对任何节点虚拟资源的操作归结为读 Read()方法和写 Write()方法,简化代码如下所示。

```

public interface InterGNodeBase //网格节点抽象接口
{
    String ID; //节点的标识
    GNodeBase father; //父亲节点
    GNodeBase son[]; //儿子节点数组
    public GNodeBase getFather(); //获得父亲节点
    public void setFather(); //设置父亲节点的值
    public GNodeBase getFirstSon(); //获得第一个儿子节点
    public GNodeBase getNextSon(); //获得下一个儿子节点
    public GNodeBase setSon(); //设置儿子节点的值
    public void addSon(GNodeBase son); //增加一个儿子节点
    public boolean isLeaf(); //判断是否叶子节点
}

public interface InterAccess //访问资源抽象接口
{
    public void write(); //将计算或任务提交给节点资源
    public Object read(); //从节点中取的计算结果
}

public class GNode implements GNodeBase //网格节点类
{
    public GNode(); //构造方法
    ID=address.getHostAddress(); //用主机 IP 初始化节点的标识
    public void Initialize(); //初始化方法
    public void destroy(); //撤消对象方法
    ...//接口 GNodeBase 的方法的实现
}

public class TJvmNode extends GNode, Access //TJVM 类
{
    ResourceCpu CPU; //CPU
    ResourceMem MEM; //内存
    TJvmNode(); //构造方法
    public void write(); //实现该方法
    { ...//提交一个 java 任务 }
    public Object read(); //实现该方法
    { ...//获得执行结果 }
}

public class TDodNode extends GNode, Access //TDOD 类
{
    SQLRequest request; //SQL 请求
    Response response; //SQL 执行结果
}
  
```

```

TDodNode(); //构造方法
public void write(); //实现该方法
{ ...//提交 SQL 请求给数据库中间件 }
public Object read(); //实现该方法
{ ...//获得执行结果 }
}

public class TGlobusNode extends GNode, Access //Globus 类
{
    String serviceadd; //服务地址
    TGlobusNode(); //构造方法
    public void write(); //实现该方法
    { ...//提交调用参数给 globus 服务 }
    public Object read(); //实现该方法
    { ...//获得执行结果 }
}
  
```

下面是管理接口,定义了管理节点与它的儿子节点的接口规范,每个管理节点都必须实现它。

```

interface InterManagerChildren
{
    ListenToChildren(Msg msg) {...}; //监听来自子节点的消息。
    CreateSon(); //创建一个儿子节点。涉及资源发现问题。
    CancelSon(); //撤销不使用的子节点
    SendMsgSon(); //发送控制命令给儿子
    ...
}
  
```

下面是传递接口,定义了传递节点与它的儿子节点、父亲节点的接口规范,每个传递节点都必须实现它。

```

interface InterDelivery
{
    ReceiveChildren(); //接收儿子节点的信息
    ReceiveFather(); //接收父亲节点的信息
    SendInfoFather(); //发送信息给父亲
    SendInfoSon(); //发送信息给儿子
    ...
}
  
```

### 4 树型网格的实现问题

树型网格的实现,涉及到许多问题。这些问题分为两大类:一类是架构和模型问题,另一类是实现算法方面问题。首先是树型网格的体系架构,这里采用树型作为网格的模型,并对其相关问题在概念模型和抽象模型部分进行了描述,树型网格模型的相关实现算法,包括物理树与逻辑树的建立、树的动态重构算法(节点加入、删除、转移等)将另有文章介绍。然后是资源管理、信息服务、安全、服务质量等问题的相关模型和其实现算法,下面我们将进行讨论这些问题。

#### 4.1 资源管理

资源管理是网格的核心功能。它主要处理资源的描述,资源的命名,资源的发现和管理等问题,要解决好这些问题首先我们必须确定好网格资源管理模型。目前网格资源管理模型主要有三类:集中式模型、分布式模型、层次化模型等。集中式模型主要从宏观上管理和控制,但可扩展性差;分布式模型可扩展性好,但欠缺全局考虑;层次化模型是两者的折中。树型网格由于它树型结构的特点更适合于采用层次化模型,可以使用树结构来管理和发现资源的变化。树型网格根据性能及网络情况,将系统分为若干资源子系统,每个子系统内部进行任务处理和资源分配管理,再将子系统作为整体进行管理。层次化管理有利于将多个不同网络的系统组织成更大系统,适应 Grid 环境,可以克服单节点瓶颈和系统管理复杂的问题。当然,对于统一的多层次管理模型还需要进行理论上的证明和完善。

#### 4.2 信息服务

网格信息服务是任何网格基础设施的重要组成部分,它需要向网格提供了对资源的发现、描述和监控机制,需要有效地对网格资源和服务进行组织和管理。建立网格信息服务使用的模型主要有:层次模型和关系模型。目前国际上比较流行的信息服务系统主要有:Globus 的 MDS<sup>[14]</sup>使用层次模型,

R-GMA<sup>[15,16]</sup>使用关系模型等。对于树型网格资源信息管理一个最好的办法是限制频繁更新和查询数据的大小,我们可以采用分离网格资源信息的获取方式,即将经常变化的网格资源动态信息利用移动 Agent 收集,实时获得资源的动态信息;而在网格信息中心只存放资源的静态信息和指向资源的描述,让用户或其它模块自主的连接资源获取动态信息,那么就能有效的改善这些问题。

### 4.3 安全问题

安全是网格中最有挑战的一个方面, Globus 项目的 GSI<sup>[2]</sup>是一个解决网格计算中安全问题的集成解决方案。在设计树型网格的安全基础设施时,除了保证网格计算安全性的同时,还尽量考虑易用性、灵活性、可移植性和可靠性;充分利用现有网络技术(如 X.509<sup>[17]</sup>、Kerberos、SSL 等),并对某些部分进行了扩充和修改;而且,树型网格面向的是大规模的、动态的环境,仍有一些安全问题有待解决,如信任代理的瓶颈问题等。

### 4.4 服务质量

服务质量是网格关键问题之一。网格计算一个主要的目标确保资源分配最佳,包括计算和存储资源可用性,安全和网络性能等<sup>[18]</sup>。在网络层上,为了保证节点间可靠的网络传输服务,节点的通讯是可以动态层次多播的方式来实现,考虑到网格树通讯的实际情况,可以对多播协议做一种改进,采用组播与单播相结合的方式。在资源方面,采用增强资源预留机制<sup>[19]</sup>和基于服务级别协定(service level agreement)来描述用户对任务、任务对资源的 QoS 需求<sup>[20]</sup>等。

## 5 模型分析

### 5.1 树型网格模型的特点

如下图 2,整个网格以树结构拓扑来组织网格节点(GN),网格节点可以动态加入和离开系统,而且,整个网格系统是一个基于树型的分层管理体系。树型网格中资源的共享以共享虚拟资源来实现,网格节点的虚拟资源可以是 TJVM、TDOD 和 GService。树型结构有利于减轻中心节点的负载,实现大规模应用的负载均衡,提高资源查找效率,实现系统的层次化管理。同时,该树型网格为网格用户提供不同程度资源抽象,以实现用户对网格资源各种需求和透明使用。例如,用户可以通过网格提交一个大型 Java 应用来完成一个复杂科学计算任务,也可以通过网格获得某项服务,如查询车票服务。总结它的主要特点如下:

- 1) 树型结构有利于减轻中心节点负载,实现任务负载均衡,提高资源查找效率;
- 2) 实现层次化管理,降低管理复杂度;
- 3) 可以实现自底向上、多级、面向需求的资源抽象和多种资源融合;
- 4) 新加入节点可以根据其网络情况来寻找父节点,提高各节点之间的网络通信性能;
- 5) 大型 Java 应用的分布透明执行,面向高性能;
- 6) 该网格体系结构的每个节点上资源分类专业化。TJVM 虚拟计算机 CPU 和内存资源, GLOBUS 网格服务虚拟各种软件和数据资源, TDOD 虚拟集成各种数据库资源;
- 7) 面向主题的资源共享和多样化资源表现形式。事实上多样性是事物的存在方式,就如同软件的多样性和计算机应用技术的多样性一样。

### 5.2 虚拟资源树的构造及分析

为了在树型网格中可以分布并行地执行大型 Java 应用,可以让多个具有 TJVM 资源的节点形成一棵虚拟 TJVM 资源树,如图 2 所示。而且,为了能高效地实现分布并行计算,在这棵虚拟 TJVM 资源树上分配任务必须满足内射规则和迁移唯一规则。对于性能方面:首先容易实现分层管理,减轻主节点负担,消除整个系统的瓶颈;随着 TJVM 树的深度的增加,可并行的 TJVM 节点资源数增加,必然可以增加任务执行的并发度,从而提高 Java 应用执行的效率。当然,这可能并不是最佳调度(性能/消耗),这方面进一步分析将另有文章介绍。

为了在树型网格中实现服务资源的快速查找和有效分配,可以让多个具有相同类型 GService 资源的网格节点构成一棵 GService 虚拟资源树,如图 2 所示。当用户申请服务资源时,将首先转向需要类型 GService 虚拟资源树,然后在树中查找最“合适”的服务资源给用户。这样避免在整个网格服务中没有目的地查找,缩小资源查找的范围,从而提高资源查找的效率。而且,在一棵资源树也更容易实现负载均衡,提高资源的利用率。

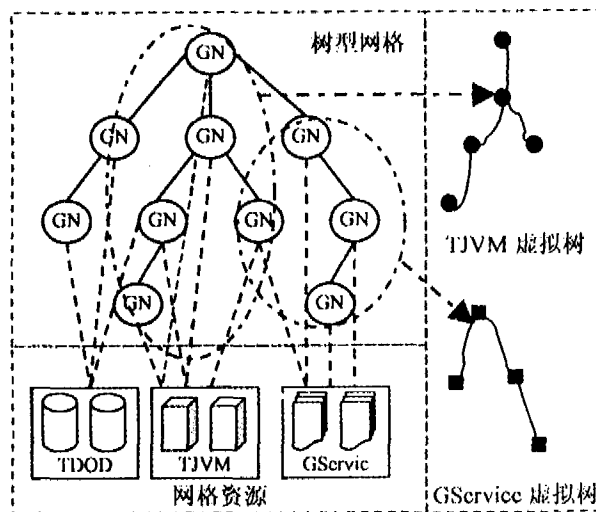


图 2 树型网格模型

**结论** 本文首先分析现有网格体系结构存在的问题与不足,提出了一种基于树型网络拓扑组织网格节点的网格环境 TGrid。在树型网格模型中,以对资源的层次化抽象来集成各种资源,并通过构造虚拟资源树来实现资源专业分类和高效利用,而且提供面向主题的资源共享和面向高性能计算,能满足不同用户的需要。文章重点给出树型网格的概念模型和资源抽象模型,并讨论了树型网格的实现问题和分析了树型网格模型。

当前,我们正在对相关理论进行模拟实验分析,并正在实现相关的算法和协议,同时也开发树型网格的原型系统。当然,还有很多具体实现问题需要进一步的研究和解决。

### 参考文献

- 1 Baker M, Buyya R, Laforenza D. Grids and Grid Technologies for Wide-Area Distributed Computing. Software - Practice and Experience, 2002, 32(15): 1437~1466
- 2 Foster I, Kesselman C, Eds. The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2004
- 3 Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid. International Journal on Supercomputing Applications, 2001, 15(3): 200~222

(下转第 15 页)

```

    </Participant>
  </Transition>
  .....
</Transitions>
</Arcs>
<! --连接 t0 和 p1 的弧, 省略了其它连接弧-->
<Arc id=a1>
  <SourceId>t0</SourceId>
  <TargetId>p1</TargetId>
  <Weight>1</Weight>
</Arc>
.....
</Arcs>
</Variables>
<! --变量 x-->
<Variable id=v1>
  <Name>x</Name><Type>int</Type>
  <Scope>global</Scope>
</Variable>
</Variables>
.....
</Pnml>

```

**结束语** 模型驱动的体系结构(MDA)是OMG提出的新方法学,是一种新的系统开发途径,强调整个系统开发过程由对软件系统的建模行为所驱动。运用MDA,从 workflow管理的特点出发,本文给出了模型驱动的工作流管理系统框架。基于该框架,我们开发了一套集成的可视化建模工具 WorkflowStudio,主要用来建立平台无关的模型,例如,过程模型、组织模型和功能模型(表单)等。这些模型可以部署到 workflow引擎上执行。通过该框架,可以初步生成基于J2EE平台的工作流引擎。正如OMG体系结构工作组的Frankel所说“MDA的潜力要得到充分的发挥还需要若干年的艰苦工作”,MDA和工作流技术本身并不成熟,这也体现在我们的理论研究和实践中。因此,在进一步的研究工作中,需要着重解决几个方面的问题,主要包括完善形式化工作流模型过程网,完善从CIM到PIM,从PIM到PSM以及从PSM到代码的转换规则定义等。

**致谢** 在此,对北京大学信息科学技术学院杨芙清院士举办,袁崇义教授和王立福教授主持的讨论班上的同学和老师表示感谢。

## 参 考 文 献

- Hollingsworth D. The Workflow Reference Model. Document Number: TC00-1003, Document Status - Issue 1.1. Workflow Management Coalition, 19-Jan-1995
- Mukerji J, Miller J. MDA Guide Version 1.0.1. Document Number: OMG/2003-06-01. Object Management Group, 12-Jun-2003
- Hur Wonchang, Jung Jae-yoon, Kim Hoontae, et al. Model-Driven Approach to Workflow Execution. International Conference on Business Process Management (BPM 2004). In: Desel J, Pernici B, Weske M. Eds. LNCS 3080, 2004. 261~273
- Breton E, Bezivin J. Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International, 8-12 Oct. 2001. 225~230
- 林嵩,赵建华,李宜东,等.从EDOC模型到J2EE程序:一个MDA工具的实现.计算机学报,2004,31(9):106~109
- Scheer A-W. ARIS - Business Process Modeling. 3rd Edition. Springer Verlag, April 2000
- van der Aalst W M P. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 1998, 8(1):21~66
- 赵文.工作流建模及模型验证技术研究:[学位论文].北京:北京大学,2004
- 袁崇义. Petri网原理与应用.北京:电子工业出版社,2005
- 赵文,胡文蕙,张世琨,等.工作流元模型的研究与应用.软件学报,2003,30(6):1052~1059
- Kleppe A, Warmer J, East W. MDA Explained: The Practice and Promise of The Model Driven Architecture. 1st Edition. Addison-Wesley Professional, April 25, 2003
- Jungel M, Kindler E, Weber M. The Petri Net Markup Language. Petri Net Newsletter, 2000, 59:24~29
- Billington J, et al. The Petri Net Markup Language: Concepts, Technology, and Tools. In: van der Aalst W, Best E, eds. Proc. Int'l Conf. Applications and Theory of Petri Nets 2003. LNCS, 2679. Springer, 2003. 483~505
- 迟美娜.基于JB-PNML的青鸟过程部署工具的设计与实现:[学位论文].北京:北京大学,2005

(上接第9页)

- Foster I, Kesselman C, Nick J M, et al. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 2002
- Introduction to Grid Computing with Globus[EB/OL]. <http://www.ibm.com/redbooks>, 2003
- The Globus Project[EB/OL]. <http://www.globus.org/>, 2004
- 徐志伟,李伟.织女星网格的体系结构研究.计算机研究与发展,2002,39(8)
- HUANG Li-Can+, WU Zhao-Hui, PAN Yun-He. A Grid Architecture for Scalable e-Science and Its Prototype. Journal of Software, 2005, 16(4):577~586
- Allen G, Davis G K, Dolkas K N, et al. Enabling applications on the grid: A Gridlab overview. International Journal of High Performance Computing Applications, 2003, 17(4):449~466
- Buyya R, Abramson D, Giddy J. A Case for Economy Grid Architecture for Service-Oriented Grid Computing. In: Proceedings of the International Parallel and Distributed Processing Symposium: 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), April 23, 2001, San Francisco, California, USA: IEEE CS Press, USA, 2001
- Ardaiz O, Sanjeevan K, Sanguera R. A 3-tier Grid Architecture and Interactive Applications Framework for Community Grids. ICCS, 2004. 67~74
- Green Grid Technical Overview. <http://grid.dartmouth.edu/docs/architecture.php>

- docs/architecture.php
- Zheng Xiao-lin, Chen De-ren, Lu Lin. Research of architecture for grid manufacturing. Proceedings of the Ninth International Conference, Computer Supported Cooperative Work in Design, 2005, 1:345~349
- Czajkowski K, Fitzgerald S, Foster I, et al. Grid information services for distributed resource sharing. The 10th IEEE Int'l Symp on High2Performance Distributed Computing, San Francis2co, California, 2001
- DataGrid Information and Monitoring Services Architecture: Design, Requirements and Evaluation Criteria: [Technical Report]. DataGrid
- Tierney B, Aydt R, Gunter D, et al. A Grid Monitoring Architecture. The Global Grid Forum GWD-GP-16-2
- Welch V, Foster I, Kesselman C, et al. X.509 proxy certificates for dynamic delegation. 3rd Annual PKI R&D Workshop, Apr 2004
- Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, 2001, 15: 200~222
- Foster I, Kesselman C, Lee C. A Distributed Resource Management Architecture that Supports Advance Reservation and Co-Allocation. In: Proceedings of the International-Workshop on QoS, 1999, 27~36
- Leff A, Rayfield J T, Dias D M. Service-Level Agreements and Commercial Grids. IEEE Internet Computing, 2003, 7(4):44~50