

推理式分布数据库管理系统 T-POREL及其实现*)

高全泉 柴兴无[✓] 徐建礼 周为群 周龙骧 TP311.13

(中国科学院数学研究所, 北京100080)

摘 要

T-POREL is a reasoning distributed Data Base management system which has been implemented on microcomputer network. The main user language of T-POREL is the universal Interactive Logic Inference Language, named Tuili. This is a more powerful, more convenient and more effective language than the well known Prolog. The basic DBMS of T-POREL is the DDBMS named C-POREL possessing fully distributed control for multi-nodes and multi-terminals(users). The reasoning of T-POREL can be made by using those data located in C-POREL and the results of reasoning can also be used to modify DDB dynamically. In This paper, the main features and basic functions of T-POREL are introduced. The coupling mode of Tuili and C-POREL and the principal ideas and strategies of implementing coupling are presented.

一、引 言

演绎数据库是AI、逻辑程序设计和DB技术相结合的一种具有演绎推理能力的数据库系统,在过去十几年里,经历了迅速全面的发展。由于它具有集人工智能、逻辑程序设计和数据库技术的实用性研究结果于一身的特特点,已成为数据库技术中最受重视的前沿研究领域之一。

演绎数据库的实现是近年来人们普遍重视的研究课题。Prolog语言在演绎数据库的实现中占有重要地位。Prolog把数据库与程序设计语言在形式上达到了完全和自然的统一,既可作为实现演绎数据库的基本工具,又能直接用于与DBMS结合构造演绎数据库

系统。近年来,为了建造原型系统,那些基于编辑程序设计和数据库联合的方法得到了广泛的研究,这些方法的主要区别在于Prolog-DBMS的耦合性质和实现耦合的技术。由于能够直接运用现有的Prolog解释器和传统的DBMS,所以这些方法可以达到快速、经济地实现之目的。

用Prolog和DBMS来构造演绎数据库系统虽然有明显的优点,但也有不足之处。当今人们普遍对Prolog寄予厚望,希望它能作为一个可以支持多种应用的有力工具,然而仅就目前的情形而论,Prolog在实际应用中的表现尚属差强人意。Prolog语言本身需要不断的完善和发展,才能与人们所期望的最

*) 国家自然科学基金和机电部电科院资助项目。

*) 中科院管理、决策、信息系统开放实验室资助项目。

终目标相适应。Prolog的最终目标应是既能适用于传统的程序设计任务,同时又能适应于AI程序设计和数据库任务的通用逻辑程序设计语言。正是由于Prolog本身能力所限,Prolog-DBMS的演绎推理及计算能力不可能有新的提高,其执行效率亦无改善,其它一些在Prolog中固有的弱点也依然存在。另一方面,DBMS对数据的支持能力也只限于集中式或并非真正分布式。

本文将介绍我们已经实现的一个属于演绎数据库范畴的推理式分布数据库管理系统T-POREL的设计及实现。T-POREL是通用逻辑推理语言Tuili[1,2]和分布式数据库管理系统C-POREL[3,4,5,]通过松散方式耦合而成的。Tuili语言是一个比Prolog功能更强、使用更方便、适应面更广且效率更高的通用逻辑推理语言。C-POREL是一个真正集成运行、完全透明的分布式数据库管理系统。较之一般的Prolog-DBMS,T-POREL无论在推理能力还是数据支持能力方面,都有较大的优势。本文提出的耦合方式和实现耦合的技术被实际用于T-POREL系统的实现,取得了预期的效果。T-POREL目前可在用以太网相联的三台UV-68020微机上运行,并已在实际应用中初步使用。

二、用于构造T-POREL和Tuili语言和C-POREL系统

1.通用逻辑推理语言Tuili

“Tuili”一词具有双重含义,既代表“Tool of Universal Interactive Logic Inference”,也是“推理”的拼音。这由陆汝钊教授在国际上首先提出并几经修改发展。Tuili已经在UV-68020微机及个人电脑(PC/XT.286、386系列)上实现[6-9]。

Tuili的设计体现了产生式系统和逻辑程序设计的有机结合。Tuili以产生式系统为主要的知识表示方法,它的知识基分为图象基、数据基、过程基和规则基。

图象基是知识元的说明部分,包括常量、变量说明、谓词说明及形式函数(相当于复

合项)说明等。

数据基的内容是已知为真的事实或实在谓词,它的内容可以动态地改变。比如,当一条规则向前匹配(以向前推理方式执行)成功,使得规则右部的谓词加入到数据基中。

过程基的内容是用Tuili以外的语言写的程序。允许使用的外部语言是C语言。若在过程基中定义了某个函数f,则f可在规则的相应语法位置上调用。

规则基由一组规则组成。每个规则代表推理的一个步骤。每个规则基表示在一个推理过程(或子过程)中允许使用的全体规则。规则基都有级别,级别高的可以启动和控制级别低的规则基,完成一个子任务。

Tuili语言系统的基本特点如下:

1) 丰富的数据类型。包括整、实、布尔、串、表、形式函数、记录、串元(由变量、常量和连接符组成的一种表达式)等基本类型和复合类型。

2) 知识库模块化。可以有任意多(理论值)个规则基和数据基。每一个规则基相当于一个Prolog程序。

3) 推理机制和搜索机制多样化。不仅有向后推理(包含了Prolog的推理功能),也有向前推理,既有深度优先搜索,又有广度优先搜索,以及根据启发式函数指导搜索的最佳优先。

4) 控制知识和领域知识表示一元化且控制结构层次化。规则基可以组成任意(理论值)深度的层次结构,下层的推理方式、搜索方式、求解目标等均受上层控制,这种控制根据运行情况动态确定。

5) 较多的非单调推理功能。通过谓词操作符以及说明具有状态性的谓词的处理,使得数据基的变化成为非单调的。

6) 事件驱动的精灵(Demon)机制。用于处理非单调性质的谓词以及处理匹配和推理的各种成功和失败等情况发生时的情形。

7) 方便用户的诸多考虑。具有简便的交互方式、自动解释和输出解、改变推理正常执行的中断机制及推理控制信息不全时的缺省规定等。

8) 齐全的系统设备。共有各类内部谓词、内部过程和内部函数等共一百多个。

9) 有效的软件接口。可以使用C语言编程,并

向下兼容Prolog的推理功能。

10) 采用编译方式实现, 应用了优化算法。编译系统及生成的目标都是C语言, 具有良好的可移植性和较高的运行效率。

2. 分布式数据库管理系统C-POREL

C-POREL是一个在微机网上真正集成的、完全透明的多终端(多用户)的系统, 其最主要的特点如下:

1) 实行完全的分布式控制, 网上所有节点一律平等, 排除任何集中控制因素。

2) 数据分布对用户完全透明, 网上任何节点均可存放数据, 用户无需关心数据具体的存放地点, 可以查询任意地点的数据。

3) 支持多节点多用户的并发事务运行。

4) 查询语言提供的功能全, 用户界面好, 可直接将结果数据以表格和图形方式输出。

C-POREL的总体体系结构如图1所示。

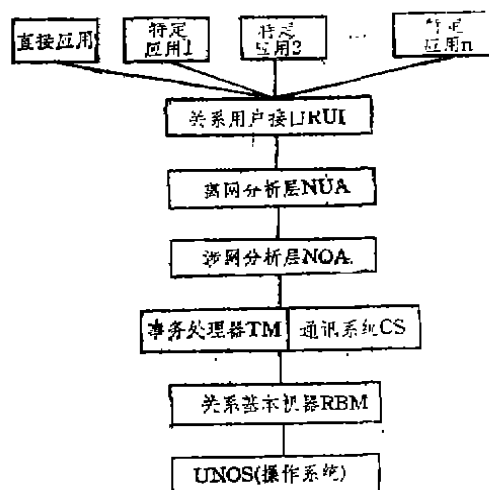


图1 C-POREL的体系结构

三、T-POREL的体系结构

Tuili语言接口与分布式数据库系统的

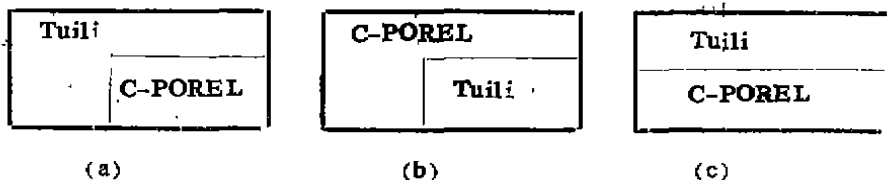


图2 不同紧密度耦合方式

耦合采用紧耦方式或采用松散耦合方式是一项需要过细考虑的策略。图2列出了三种不同紧密度的耦合方式:

图2(a)是紧耦合方式。在Tuili系统中统一地直接地纳入了分布数据库系统的功能。这种方式的优点是效率高、结构紧凑。其缺点也很显而易见, 设计复杂且不易实现, 缺乏灵活性。图2(b)也是紧耦合结构, 与(a)不同, 它在分布式数据库系统中纳入了逻辑推理功能。一种实现方法是例如增加一个Tuili用户接口。

图2(c)是松散耦合方式。Tuili系统和C-POREL系统都是独立设计和实现的系统, 它们之间再统一地结合起来。因此, 它既可单独提供推理功能和C-POREL功能, 也可提供同时需要两者紧密结合的统一功能。这种方式的优点是灵活, 设计和实现自然地符合软件工程的原则, 因而较易于划分系统的层次结

构和模块结构, 有利于高效和正确地实现。

灵活性是软件的最根本的性质, 最巨大的优势。实践证明, 包罗万象的复杂大系统不易为用户接受。例如PL/1语言和ALGOL 68语言, 它们的普及程度远远逊于稍后提出的更为单纯简洁的PASCAL语言。较早提出的分别单独具有科技计算功能的FORTRAN语言、数据处理功能的COBOL语言、表处理及符号处理功能的LISP语言, 其被广泛使用的程度要远远大于具有所有这些综合能力力的PL/1和ALGOL 68。效率只是综合因素中的一个, 效率再高但缺乏灵活性也不易于推广使用。例如, LISP机和数据库机和通用计算机相比, 其数量几乎可以忽略。

有鉴于此, 在设计推理式分布数据库系统T-POREL时, 我们采用了图2(c)的松散耦合策略。T-POREL中, 以通用逻辑推理语言Tuili作为用户主界面语言, C-POREL作

为Tuili以外的数据访问支持系统。从使用的角度来看, T-POREL是Tuili的扩充, 即在Tuili基础上扩充了应用外部的分布式数据库的能力而成。T-POREL程序中, 用户不需要显式地调用C-POREL, 所有涉及外部的、分布在计算机网上诸节点的数据由系统自动进行存取。同样, 对外部数据库的修改, 也是系统根据用户的修改方式自动进行的。T-POREL的推理是集中式的, 用户可从计算机网上的任一节点的若干终端之一进入系统, 推理中涉及到的外部数据的处理是分布式处理。T-POREL的总体结构如图3所示。

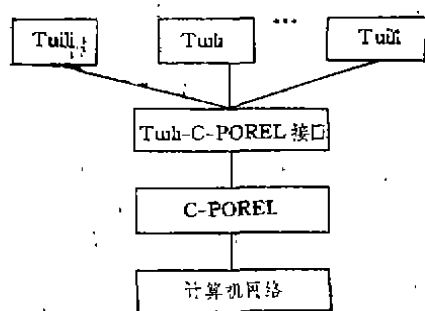


图3 T-POREL的体系结构

四、Tuili和C-POREL在语言级的结合

1. 基本原理和设计目标

Tuili以谓词为知识元, 把每个谓词看作一个图象, 以图象匹配作为推理的基本方式。谓词用于描述客体及它们间的关系, 由谓词名和一组参数组成, 谓词名表示客体间的关系, 参数表示客体本身。Tuili的数据基是一个工作空间, 用于存放推理时已知为真的谓词或经过推理产生的新的谓词。所有数据基的初态, 构成了推理时的一个封闭的已知世界。由于这类谓词不依赖其它任何条件而成立, 故称之为实在谓词或事实或数据。推理的执行过程是规则基中的规则不断直接或间接应用数据基的过程。如果仅就为推理提供已知前提或数据的作用而言, Tuili具备把这种内部的数据基拓广到通常意义下的外部(可能是分布式的, 下同)关系数据库的便

利条件。通常意义下的数据库是经关系模式组织的数据集合, 每个数据元组描述实体和实体的联系。与Prolog相同, 事实或实在谓词与数据库的关系数据元组的表示形式之间, 存在着一致的对应关系: 谓词名对应于关系名, 谓词的参数对应于数据元组的属性。虽然就表达能力和范围而言, 数据元组尚不及谓词表示, 但这并不影响把关系数据作为Tuili数据的一般性(属性的类型是参数的子类)。

通过拓广Tuili所使用的数据范围, 使得不仅在推理时可以使用自身数据基的内容, 而且, 凡在联网节点机上的数据库中的数据, 都可直接作为推理的前提使用, 可以解决用户在编写推理程序时的基本事实不充分和证据不足的问题, 使得数据库查询和推理一体化。

Tuili与C-POREL结合的另一个方面是将经过推理已经证明了的或者推演出的新结果返回到分布式数据库之中, 使得数据库中不仅包含基本的前提数据或原始数据, 而且包含有经过证明或推理获得的结论结果, 它们既能作为数据查询结果引用, 又能成为后来的推理的新起点。

2. 结合方式

从应用来看, 本质上, 用户对数据库的基本需求是查询和广义的修改(包括增、删、改)。Tuili与C-POREL结合, Tuili对C-POREL的情形也是如此。为了能在Tuili语言里应用外部数据库, 我们采用在Tuili的数据基中说明引用外部数据的数据库谓词。另一方面, 通过设置一组专门的数据库过程(它们作为系统的内部过程)来完成对数据

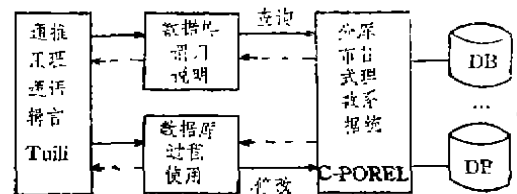


图4 Tuili与C-POREL结合方式

库的修改。数据库谓词说明和数据库过程统称为 Tuili-C-POREL 接口,它是沟通二者的桥梁,其作用如图4所示。

2.1 数据库谓词说明 一个数据库谓词说明是以外部数据引用标志“\$”开始的N元谓词,它可以在Tuili的数据基中的实在谓词的语法位置上出现。例如,在以下T-POREL程序片断中:

```
database(dbl),
$ student(name, m-grade, p-grade,
           c-grade),
eof dbl,
.....
```

dbl是一数据基, \$ student(name, m-grade, p-grade, c-grade)是一数据库谓词说明。student是一个4元谓词, name, m-grade, p-grade, c-grade是它的参数。

假设 \$P1 (S1, S2, ..., Sn) 为一数据库谓词说明,它所代表的语义如下:以n元谓词表示P1 (S1, S2, ..., Sn) 中的谓词名P1作为关系名,以它的n个参数S1, S2, ..., Sn的名字作为属性名,在分布式数据库中查询,并将查询结果作相反的转换,即查询结果中的关系名P1作为谓词名,查询结果(数据元组)中的属性值作为P1的参数值,按照Si(i=1, 2, ..., n) 在数据库谓词说明里的次序,将查询结果转换成谓词表示。例:设外部数据库中有关系student的下列数据元组:

name	age	sex	m-grade	p-grade	computer	e-grade
李明	21	男	90	85	80	100
王红	22	女	95	70	90	80
谢军	19	男	59	100	95	95

则本节前面的数据库谓词说明与以下程序片断中的实在谓词序列等价。

```
database(dbl),
student("李明", 90, 85, 100);
student("王红", 95, 70, 80);
```

• 70 •

```
student("谢军", 59, 100, 95);
eof dbl,
.....
```

不难看出,数据库谓词说明规定了所要引用的外部数据对象和Tuili与C-POREL的通讯协议,并且提供有效的数据优化途径,用户可以只选取那些与当前推理相关的数据的属性值。

T-POREL的规则中,对数据库谓词的运用与一般谓词的用法完全一致,不需区分它是内部的还是外部的。

2.2数据库过程 数据库过程是能完成对外部数据库广义修改的系统设备,与Prolog中的内部谓词的作用类似。以下给出T-POREL中提供给用户使用的数据库过程。insert-DB(P):将谓词P(p是n元谓词)作为数据元组插入到数据库中。

delete-DB("关系名", "条件表达式"):在外部数据库中,删除符合关系名是"关系名"且满足"条件表达式"所给出条件的数据元组,"条件表达式"是约束条件。

update-DB("关系名", "属性修改短语"):根据"关系名"和"属性修改短语"修改数据库中数据。

define-DB("关系名", "关键字段", "属性定义1" ..., "属性定义n"),在数据库中定义一个新的关系。

五、T-POREL程序举例

本节,我们来看一个具体的T-POREL程序。我们以求“斐波纳契”数为例,即:对用户给定的任一非负整数n,求它的fabnacci函数f(n)之值。斐波纳契函数是定义在非负整数上的一个函数,其定义为

$$f(0)=0$$

$$f(1)=1$$

$$f(n)=f(n-1)+f(n-2), n \geq 2$$

假设在外部数据库中存在着表示f(0)=0和f(1)=1的数据元组,它们由具有两个属性的关系f表示,

属性 \ 属性名	属性名	
	nv	fv
关系名		
	f	f
f	0	0
f	1	1

属性名分别是nv和fv,如上表所示:

在此前提下, T-POREL直接运用数据库中关系f的数据元组推导所要求的斐波纳契数,并将推导出来的值返回到数据库中。程序如下(由/*和*/括起来的是注解):

```
tuili <fabnaacci>; /*程序首部*/
pattern-base; /*图基*/
var n,y,z, int; /*变量说明*/
pred f(n,y); /*谓词说明*/
number-nv(n); /*谓词说明*/
true;
eof-pb;
database(dbl); /*数据基*/
$ f(nv, fv); /*数据库谓词说明*/
eof dbl;
rulebase(rb0, 0); /*0级(最高级)规则基*/
start, ?number-nv(n) !->ford^bread-
th^in (dbl) ^out (dbl) ^goals(f(n,
y)) ^run(rb1);
done, ->ford^in (dbl) ^out(dbl) ^
goals(true) ^run(rb2);
fail, ->print("fail") ^stop;
eof rb0;
rulebase(rb1, 1); /*1级规则基*/
f(n, z) ^f(n+1, y) ->f(n+2, z+y);
eof rb1;
rulebase(rb2, 2); /*2级规则基*/
?number-nv(n) ^f(n-1, y) ^f(n, z) ->
insert-DB (f(n-1), y) ^ /*数据库过
insert-DB(f(n, z)) ^stop; 程*/
eof rb2;
eof tuili; /*程序尾部*/
```

注记:本例中,由于没有用到外部语言(C),故过程基缺席。程序从0级规则基的带start标号的规则启动。首先执行咨询谓词number-nv,即:系统先在用户终端上输出“请输入一个非负整数n:”,然后等待用户输入,当用户输入一个非负整数后,number-nv的参数n便约束为这个数,并将其记忆在系统数据基systore之中。该规则的右部启动一个向前推理,以宽度优先方式执行规则基rb1,执行rb1时的输入/出数据基是dbl,那里有来自外部数据库中的数据(f(0, 0), f(1, 1)),推理目标是f(n, y)。rb1的规则是fabnaacci数(n≥2)的推导规则。这个推理成功之后,控制返回到rb0中以done

为标号的规则(暗中断发生),去执行规则基rb2,rb2中的规则作用是将方才推出来的结果即f(n-1)和f(n)之值插入到外部数据库中,rb2里,执行number-nv时,使用第一次执行该咨询过程时获得的已记忆在systore里的结果。在将f(n-1)和f(n)之值插入到外部数据库后,程序即停止。

假定用户对n的输入值是15,则在运行该程序之后,外部数据库(不必关心它的地理位置)中新增加了f(14, 377)和f(15, 610)这两个数据元组,可为别人引用。并且,若在以后需要大于15的数(设为20)的斐波纳契数,则能够以f(14)和f(15)为起点,而不必从0开始这就使推理树的高度大为降低 如图5所示

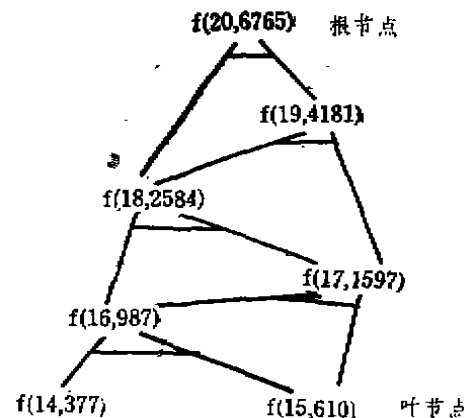


图5

六、关于T-POREL的实现

对T-POREL的实现,我们只讨论接口的实现技术,即数据库谓词说明和数据库过程的实现, Tuili与T-POREL的一般实现技术不属于这里讨论的范围,见参考文献[7, 8, 10, 11, 12]。

数据库谓词说明包括了将其转换为数据库的查询语句和数据到谓词的转换约定,其实现与Tuili和T-POREL的实现方法有关。Tuili源程序要经过Tuili→C代码→可执行代码的两级编译。这样,对于数据库谓词说明中涉及到的外部数据,不论何时从数据库查询,都需要同其它程序成份一样,经过前述两级编译后,才能在运行阶段使用。显然,若在编译之前,根据数据库谓词说明查

询数据并作转换,将其加工成标准的Tuili数据,经过上述编译后,外部数据与内部数据一样都成为Tuili数据基的实在谓词的运行时结构。这样,在运行阶段,既不需要数据查询,也不需要把查询结果动态编译成运行时的结构。基于这种思想,我们采用了静态数据查询预处理方法。具体就是,对T-POREL程序,从中挑出数据库谓词说明,将其转换成RDBL语句——一种类似于SQL的查询语句。比如,数据库谓词说明 \$student (name, m-grade, p-grade, c-grade) 将被转换成以下RDBL语句。

BEGIN

USE REL ON LEVEL 3 student,

SELECT name, m-grade, p-grade, c-grade

FROM student,

END \$

然后,用RDBL语句形成一个查询事务,调用C-POREL系统进行查询,C-POREL系统并将获得的查询结果作必要的格式转换(C-POREL系统能区别一般的查询和与Tuili合作下的查询)。系统以这种经过格式转换的查询结果取代原来的数据库谓词说明。经过这种预处理后,所有外部数据均以谓词形式作为相应数据基的一部分,因而能交给Tuili编译系统编译。由于在编译之前一次将符合要求的外部数据提取、加工完毕,故在运行阶段有较高的效率。

数据库过程是修改数据库的系统设备,犹如其它过程调用和函数调用,每执行一次,都要通过调用C-POREL系统完成所要求的动作。对它的实现,我们采用动态方法。就是,在运行阶段,每当执行到的目标是数据库过程时,即以该过程的诸参数的实际值,形成由过程名规定的对数据库修改(增、删、改、定义新关系)的RDBL语句,并将其作为一个事务交给C-POREL,C-POREL经过编译加工等处理,再由事务管理器TM启动别的进程完成对数据库的实际修改。这

种动态访问数据库的方法要求在运行阶段,用户的T-POREL进程和TM进程同时存在,通过进程通信,完成相互的合作。

七、结束语

本文简要介绍了T-POREL系统的基本特点,讨论了Tuili与C-POREL的结合方式和实现这种结合的基本方法——静态和动态访问数据库的实现方法。不论静态还是动态,都使用了查询语言RDBL,习惯上称之为松散耦合。T-POREL中,Tuili和C-POREL是基础,它们是两个具有相当规模的大型复杂系统。事实上,Tuili系统和C-POREL系统是中科院数学所计算机科学室人工智能和分布式数据库两大方向具有代表性的工作。从最初设计到计算机上实现,先后历时近六年,有几十名研究人员和研究生参加。正是因为具有Tuili和C-POREL这样的基础,才使得高性能的推理式分布数据库的实现成为可能。T-POREL的源程序包括45,000行PASCAL代码及26,700行C代码。

在结束本文之际,我们向曾经对T-POREL直接或间接作过贡献的所有人员和各有关部门谨致谢意。

参考文献

- [1] Lu Ruqian, Tuili, A Language for Expert Systems, Advances in Chinese Computer Science, I pp. 99-114, World Scientific, Singapore, 1988
- [2] 陆汝铃, 关于Tuili的设计, “计算机软件开发方法、工具和环境”, pp.229-238, 西北大学出版社, 1985年
- [3] Zhou Longxiang, C-POREL—A Distributed Relational Data Base Management System on Microcomputer Network, Scientia Sinica, Series A, Vol. XXIX, No.1, Jan.1986, pp.78-91
- [4] Zhou Longxiang, The Catalog Management Strategy of C-POREL, Advances in Chinese Computer Science, Vol. 1, 1988, pp.143-154, Wor-

面向对象的分布式OS设计模型

何炎祥 (武汉大学计算机系 武汉430072) TP311.12

刘栋臣 (东北工学院计算机系 沈阳110006)

摘

要

Object-Oriented distributed OS design model is becoming a development trend. This paper discusses the construction and design model of object-oriented distributed OS, including object concepts, object's synchronization, object's capabilities, object-oriented design methodes, and process management, memory management, communication management, device and I/O management. Finally, the differences between object-oriented distributed OS design model and process-oriented distributed-OS is presented.

一、对象及对象的构成

对象(object)是由封装起来的数据结构及其相关操作构成的集合,是一种抽象数据类型。数据结构描述了数据文件、可执行模块、进程间通信、信箱、目录、应用程序、硬件资源以及它们的控制软件等等,操作则描述了这些数据结构在操作过程中如何变化。由此,计算机系统可看作是对象的集合,而不再是由资源和进程构成。

对象由一个外部成分和一个内部成分构成。外部成分定义对象的用户视图,包括允许用户使用的操作和外部数据类型的定义。内部成分由内部说明、数据、程序、资源管理者等组成,即被操作的实际对象及其实现和表达形式。

一个对象所允许的操作规定了它的特性,定义了它对于外部世界的行为。这些操作及其内部定义的数据结构和计算的结合描

Id scientific, Singapore

[5] Zhou Longxiang, Transaction Management in Distributed Database System C-POREL, Science in China, Series A, Vol.32, No.2, Feb.1989, pp.222-233

[6] 陆汝铃,高全泉,“Tuili文本及使用手册,中国科学院数学研究所,1990,12

[7] 高全泉, Tuili(推理)语言的编译方法与实现技术,软件学报,1991.2

[8] 高全泉, Tuili实现系统里的多推理机结构,计算机学报,1991,9

[9] Gao Quonquan, An Implemented System of Knowledge Inference Language

ge Tuili-Tuili1.1, Research Report of Laboratory of Management Decision and Information Systems, Academia Sinica, MADIS, YB90-0015, 1990. 2

[10] 柴兴元, C-POREL 的关系基本机器 RBM, 中国科学院数学所硕士论文, 1987, 7

[12] 徐建礼, C-POREL中通信子系统CS的设计与实现, 中国科学院数学所硕士论文, 1988.6

[12] 周为群, 分布数据库系统C-POREL的事务管理TM的设计与实现, 中国科学院数学所硕士论文1988.6