ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.6, June 2009, pp.1406−1413
doi: 10.3724/SP.J.1001.2009.03424

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

DDS并行模型及其形式化*

刘真环[1,2+], 韦 立[2,3], 陈 艳[2], 赵荣盛[2], 王 驹[4]

[1](桂林空军学院 教研部,广西 桂林 541003)

[2](广西师范大学 数学科学学院,广西 桂林 541004)

[3](贵州大学 计算机科学与技术学院,贵州 贵阳 550025)

[4](广西师范大学 计算机科学与信息工程学院,广西 桂林 541004)

Parallel Model of DDS and Its Formalization

LIU Zhen-Huan[1,2+], WEI Li[2,3], CHEN Yan[2], ZHAO Rong-Sheng[2], WANG Ju[4]

[1](Department of Teaching and Researching, Guilin Air-Force Academy, Guilin 541003, China)

[2](College of Mathematics Science, Guangxi Normal University, Guilin 541004, China)

[3](College of Computer Science and Technology, Guizhou University, Guiyang 550025, China)

[4](College of Computer Science and Information Engineering, Guangxi Normal University, Guilin 541004, China)

+ Corresponding author: E-mail: zhhliu983@163.com

Liu ZH, Wei L, Chen Y, Zhao RS, Wang J. Parallel model of DDS and its formalization. *Journal of Software*, 2009,20(6):1406−1413. http://www.jos.org.cn/1000-9825/3424.htm

Abstract: Although the model of DDS (deadline-driven scheduler) is a classical model of real-time system, the space-condition is not included in its original framework. Based on the extension of the original framework of DDS, multi-processes task scheduling with space -constraint is investigated. By studying the parallel model of DDS, the concept of maximal separated task-set, the primary scheduling algorithm and the general scheduling algorithm are presented. In order to formalize the parallel model of DDS, the paper extend duration calculus to DC* with the idea of separation logic, which can express the space-constraint successfully, and give the formalization too.

Key words: parallel model of DDS; general scheduling algorithm; separation logic; duration calculus; formalization

摘 要: DDS(deadline-driven scheduler)模型是实时系统研究中的一个经典模型,但其原始设置中未提及空间因素.在 DDS 模型的原始设置上进行扩展,给出了 DDS 并行模型并在该模型设置下研究带空间限制的任务调度问题.提出了极大空间相容组的概念,并给出了全局调度算法和该算法可行的条件.最后还引入分离逻辑的思想对时段演算进行扩充,得到了新的形式系统 DC*,利用 DC*把 DDS 并行模型形式化.

关键词: DDS 并行模型;全局调度算法;分离逻辑;时段演算;形式化

中图法分类号: TP301 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60573010, 60663001 (国家自然科学基金); the Innovation Project of Guangxi Graduate Education of China under Grant No.2007106020701M52 (广西研究生教育创新计划)

Received 2008-05-04; Accepted 2008-07-09

时限驱动调度算法(deadline-driven scheduler,简称DDS)是实时系统中的一种经典调度算法[1],与之相关的调度问题称为DDS问题.原始的DDS模型设置情况如下[2]:

假定有m个任务$p_1,p_2,\ldots,p_m$,存在一个处理器M,

(1) 每个任务p_i均有一个相同的周期性的截止期T_i,每个周期内其必要处理时间总和为C_i($C_i \le T_i$);

(2) M 在每个瞬间只能处理一个任务.

DDS是一种非常简单的调度算法:赋予离截止期最近的任务以最高优先权,即处理器总是先处理离截止期最近的、最紧急的任务.Liu-Layland定理给出了该算法可行的充要条件[2]:

Liu-Layland 定理. DDS 算法可行的充要条件是

$$\sum_{i=1}^{m}\frac{C_i}{T_i}\le 1,$$

其中,$\frac{C_i}{T_i}$ 记为任务p_i的利用率.

DDS 模型是实时系统研究中取得的早期成果,具有很强的代表性,很多现实生活中存在的诸如网络拥塞、生产流程调度、铁路交通运输调度等问题的解决都可以借助该模型.在计算机专家、人工智能专家的观念中,一个有价值的系统应该有两部分:数学模型和形式语言.一般情况下,构建形式语言的工作往往比构造数学模型的工作更加重要和困难,因此,将 DDS 模型形式化是计算机理论界所关注的热点.

时段演算(duration calculus,简称DC)是周巢尘、Hoare和Ravn等人提出的一种实时区间时态逻辑[3],它在实时系统的形式化研究领域已经得到了一定的应用.DC在实时调度算法的可行性研究中有着巨大的作用,周巢尘在DC的形式系统中给出了Liu-Layland定理的完整的形式化证明[3].

虽然在DDS的原始设置中并没有考虑空间的因素,但在各种各样的应用领域中,空间却是一个不可忽视的关键因素,如在工业生产中必须考虑到生产某产品所必需的场地空间大小,在设计大型软件系统的时候必须考虑到程序运行占用的内存大小等等.在现实生产中,分工合作早已是习以为常的事情,多个处理器可以同时进行生产活动.因此,必须考虑对DDS的原始设置进行改进,允许有两个或者两个以上的处理器,同时引进空间概念,使DDS模型能够更准确地描述现实进程.注意到时段演算可以很好地解决任务调度的时间限制问题[3],但其并未考虑到空间因素,因此需要引入一种新的形式化工具,以便更好地在考虑空间因素的条件下完成对DDS扩充模型的形式化.

分离逻辑(separation logic)是一种研究计算机命令程序推理的形式系统[4],由Reynolds于 2002 年提出,其以Hoare逻辑[5,6]为基础,通过对Hoare逻辑的断言语言和程序语言进行扩充,使得新的形式系统能够更好地描述计算机当前的内部存储状态.随着研究的深入,分离逻辑已经被推广到了广阔的研究领域,如把分离逻辑应用到程序的并发推理研究中,形成并发分离逻辑[7];O'Hearn等人将分离逻辑抽象成一个代数结构,即抽象分离逻辑.经过几年的发展,分离逻辑已经成为一个十分重要的研究领域,分离这个概念也被应用在更广泛的领域,如情境演算和空间逻辑中[8,9].

1 DDS 并行模型中任务的调度问题

首先,我们建立 DDS 并行模型,而后在该模型的设置下研究任务的调度问题.

1.1 带空间限制的DDS并行模型

在原始的 DDS 问题设置基础上可以建立扩充的 DDS 模型,即带空间限制的 DDS 并行模型.其设置如下:

(1) 假定有m个任务$p_1,p_2,\ldots,p_m$,用$P=\{p_1,p_2,\ldots,p_m\}$来表示所有任务的集合;

(2) 给定一个固定的总工作空间 D,$|D|<\infty$;

(3) C_i,T_i的设置照旧;

(4) 每个任务p_i需要的工作空间为d_i,$d_i\le|D|$;

(5) 假定处理器个数不受限制;

(6) 每个任务都是独立的,即任意时刻其只能被一个处理器处理.

说明:以上设置(4)有相当大的余地可以修改成其他类型的模型,如可建立起并行、并发、并行+并发的多种数学模型,从而反映各种应用领域中的现实问题:

① 注意到这里的默认空间D是一维的,现实生产生活中的很多问题可以转化为该模型来解决.如举办一个大型会展活动,会展大厅的面积D是有限的,展商$p_1,p_2,\ldots,p_m$各自要求的空间是不同的常数,展期有周期性的要求.由于场地的形状可以灵活划分,所以,我们在安排展位和展期时可以将D看作是一维的.

② 如果每个任务p_i需要的工作空间是一个实时的函数$S_i(t)$,比如是一个周期性的、线性递减函数,那么整个流程将不断地释放出自由空间,调度者可以实时地调入新的任务以提高时间及空间的使用率.这类调度就可以表达为并发进程.比如,一个信息网络的总负荷是一个常数χ,各用户端p_i的负荷是实时函数$\chi_i(t)$,它是动态的.网络拥塞问题的基本数学框架即是$\sum\chi_i<\chi$[10].

③ $S_i(t)$的值域可以是各种二维、三维空间中的图形或立体拓扑图像,甚至是其他对象,如多个程序进程在资源共享中的数据集合等.引入空间因素后的实时系统框架,可以将大量实际问题表述为实时系统的数学模型.

本文讨论的即是其中最基本的,但也是最常见的一种模型.

1.2 DDS并行模型中任务的调度

在上述 DDS 并行模型设置下,接下来研究任务的调度问题.首先进行任务集合的划分,为此引入两个定义.

定义 1. 设$P=\{p_1,p_2,\ldots,p_m\}$,$Q\subseteq P$,$Q=\{q_1,q_2,\ldots,q_k\}$,若满足以下条件,则称Q为P的极大空间相容组:

1. $\sum_{i=1}^{k}\delta_i\le|D|$(这里,$q_i=p_{j_i},\delta_i=d_{j_i}$);
2. $\forall p\in P,p\notin Q,\sum_{i=1}^{k}\delta_i+d>|D|$($d$ 是 p 所需的工作空间).

定义 2. P 的 i 级极大空间相容组(i=1,2,…,k)是:

1. 设Q_1是P的一个极大空间相容组,Q_1称为P的 1 级极大空间相容组;
2. 考虑任务组$P-Q_1$(不考虑Q_1),从中又可以找到一个极大空间相容组Q_2,称Q_2为P的 2 级极大空间相容组;
3. 依此类推,可得i级极大空间相容组Q_i.

在此基础上,可以得出以下定理:

定理 1. 对$\forall p\in P$,存在P的i级极大空间相容组Q_i,使得$p\in Q_i$,从而存在一个整数k,使得

$$P=Q_1\cup Q_2\cup\ldots\cup Q_k,Q_i\cap Q_j=\varnothing\quad i,j=1,2,\ldots,k,i\ne j,$$

其中,Q_i是P的i级极大空间相容组.

证明:由定义显然得证. □

由此可知,每个任务无论其空间限制条件如何,都可以被划分至一个极大空间相容组内(在下文中,若没有特别指明,P 的 i 级极大空间相容组(i=1,2,…)将统称为 P 的极大空间相容组).

定义 3. 设Q是P的一个极大空间相容组,则一定存在一个利用率最大的任务$q_i\in Q$,对其他任务$q_j\in Q(i\ne j)$,都有$\frac{C_i}{T_i}\ge\frac{C_j}{T_j}$,称进程$q_i$为$Q$的主导任务.

定义 4. 任意给定一个极大空间相容组$Q=\{q_1,q_2,\ldots,q_k\}$,且$\frac{C_1}{T_1}\ge\frac{C_2}{T_2}\ge\ldots\ge\frac{C_k}{T_k}$,如果一个对$Q$的并行调度满足下列条件,则称其为$Q$的主调度:

(1) 在处理Q中其他任务的每一时刻,q_1都在被处理;

(2) 一般地,对$\forall i,1\le i\le k$,在q_i被处理时,$q_1,q_2,\ldots,q_{i-1}$都在被处理.

根据以上定义可知,用主调度对极大空间相容组进行调度后,可由主导任务来代表该极大空间相容组.下面证明主调度算法的存在性.

定理 2. 任意给定一个极大空间相容组 Q,存在 Q 的主调度算法.

证明:设$Q=\{p_1,p_2,\ldots,p_n\}$,$\frac{C_1}{T_1}\geq\frac{C_2}{T_2}\geq\ldots\geq\frac{C_n}{T_n}$.记$T=T_1\cdot T_2\cdot\ldots\cdot T_n$,则$T$是$p_1,p_2,\ldots,p_n$的公共截止期,因而只需安排$I=[0,T]$间的任务调度即可.$p_1,p_2,\ldots,p_n$的各个截止期$T_1,2T_1,\ldots,T_2,2T_2,\ldots,T_n,2T_n,\ldots$把$I$分成了互不相交的子区间$I_1,I_2,\ldots,I_k$,即$I_1\cap I_2\cap\ldots\cap I_k=\varnothing$,$I_1\cup I_2\cup\ldots\cup I_k=I$.对每个子区间$I_l(l=1,2,\ldots,k)$,根据各个任务利用率$\frac{C_i}{T_i}$的大小对任务进行分配.由前设$\frac{C_1}{T_1}\geq\frac{C_2}{T_2}\geq\ldots\geq\frac{C_n}{T_n}$,分别给任务$p_1,p_2,\ldots,p_n$安排$\alpha_l^1,\alpha_l^2,\ldots,\alpha_l^n$时间,满足$\frac{\alpha_l^i}{|I_l|}=\frac{C_i}{T_i}$($i=1,2,\ldots,n$).由于$\frac{C_i}{T_i}\geq\frac{C_{i+1}}{T_{i+1}}$,故$\alpha_l^1\geq\alpha_l^2\geq\ldots\geq\alpha_l^n$,先安排$\alpha_l^i$给利用率大的任务$p_i$,再在$\alpha_l^i$内安排$\alpha_l^{i+1}$,即保证$\alpha_l^i$覆盖住$\alpha_l^{i+1}$,如图 1 所示.接下来说明这样的调度安排是满足主调度要求的.

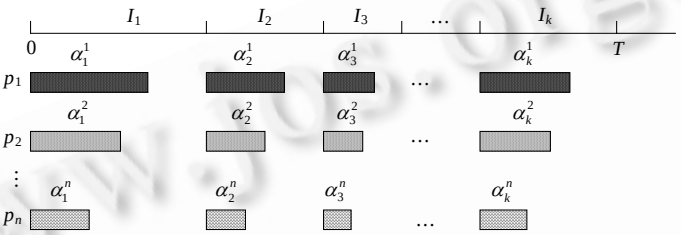

Fig.1

图 1

对任意一个任务p_i,在它的周期$[mT_i,(m+1)T_i]$内,可知它所分配到的处理时间总和等于C_i,这是因为:设$I_r\cup I_{r+1}\cup\ldots\cup I_s=[mT_i,(m+1)T_i]$,我们有$\frac{\alpha_r^i}{|I_r|}=\frac{\alpha_{r+1}^i}{|I_{r+1}|}=\ldots=\frac{\alpha_s^i}{|I_s|}=\frac{\alpha_r^i+\alpha_{r+1}^i+\ldots+\alpha_s^i}{|I_r|+|I_{r+1}|+\ldots+|I_s|}=\frac{C_i}{T_i}$,而$|I_r|+|I_{r+1}|+\ldots+|I_s|=T_i$,所以$\alpha_r^i+\alpha_{r+1}^i+\ldots+\alpha_s^i=C_i$,这就证明了这种调度方法满足了任务截止期前每个任务要求的处理时间能够被满足.又由于α_l^i覆盖住α_l^{i+1},这种调度方法满足了主调度的条件.因此,我们给出了极大空间相容组$Q=\{p_1,p_2,\ldots,p_n\}$的一种主调度算法,定理得证.

注意,当$T_i(i=1,2,\ldots,n)$是正整数时,也可以保证上述定理的正确性.证毕. □

可以发现,上述证明已经给出了一个极大空间相容组的主调度算法.

定义 5. 给出任务集合$P=\{p_i|i\in\alpha\}$,P的全局调度算法定义如下:

I. 首先按如下步骤选择、组合各个极大空间相容组:

(1) 按照利用率将任务排序,不失一般性,假定$\frac{C_1}{T_1}\geq\frac{C_2}{T_2}\geq\ldots\geq\frac{C_m}{T_m}$;

(2) 令$\{p_1\}=Q_1$,若$d_1+d_2\leq|D|$则又令$p_2\in Q_1$,否则考虑$p_3,\ldots$,直到Q_1是极大空间相容组为止,其中,Q_1是P的 1 级极大空间相容组;

(3) 对$P-Q_1$,重复步骤(1)、步骤(2)得到P的 2 级极大空间相容组Q_2;

重复上述步骤,得到P的各级极大空间相容组$Q_3,Q_4,\ldots,Q_k$.

按照上述步骤(1)~步骤(3),把任务集合P划分成 1 到k级极大空间相容组$Q_1,Q_2,\ldots,Q_k$.

II. 对每一个极大空间相容组Q_i,用定理 2 提供的主调度算法对任务进行调度.

III. 对$i=1,2,\ldots,k-1$,先处理i级极大空间相容组Q_i,再处理$i+1$ 级极大空间相容组$Q_{i+1},\ldots$,待处理完k级极大空间相容组Q_k后,再回到 1 级极大空间相容组Q_1.如此循环下去,即按照$Q_1\rightarrow Q_2\rightarrow Q_3\rightarrow\ldots\rightarrow Q_i\rightarrow\ldots\rightarrow Q_k\rightarrow Q_1\rightarrow Q_2\rightarrow\ldots$的顺序依次处理各个极大空间相容组.

注意到用全局调度算法进行调度时使用的处理器个数是不确定的,在处理第i级极大空间相容组Q_i时用到

的处理器个数是$|Q_i|$.

说明:全局调度算法较之其他任务调度方法有一个显著的特点,即它能够保证各个任务得到的处理器时间是连续的,这一点在现实工农业生产生活中任务调度问题上是具有现实意义的.如在举办大型会展活动时,展厅的空间大小是固定的,设为D;每个参展商p_i需要的展位大小也是固定的,记为d_i,其展期有周期性的需求,即在周期性截止期T_i到来之前要分配C_i的展期给p_i;全局调度算法可以保证各个参展商在每个周期内所分配到的展期是连续的.另一方面,如果任务是可切分的,则我们可以将时间及空间的使用率提高,但调度将相对复杂.

接下来,在 Liu-Layland 定理的基础上讨论全局调度算法的可行条件.为此,首先引入如下定义:

定义 6. 按定义 5 中步骤(1)~步骤(3)将任务集合P划分成 1 到k级极大空间相容组$Q_1,Q_2,\ldots,Q_k$,则称$Q_1,Q_2,\ldots,Q_k$为任务集合P的一个划分.

在此基础上,可以得出以下定理:

定理 3. 设$Q_1,Q_2,\ldots,Q_k$为任务集合P的一个划分,对每一个极大空间相容组Q_i,记 p_{i_1} 为它的主导任务,则全局调度算法可行的充要条件是主导任务组成的任务集合 $\overline{Q}=\{p_{i_1}\mid i=1,2,...,k\}$, 满足:

$$\sum_{i=1}^{k}\frac{C_{i_1}}{T_{i_1}}\le 1.$$

证明:由 Liu-Layland 定理及以上的定义和定理可以直接得出. □

2 DDS 并行模型的形式化

由于引入了空间限制,在对DDS并行模型进行形式化时,所选择的形式化工具必须能够体现出空间的要求.时段演算很好地解决了任务调度的时间限制问题,但其未考虑到空间因素.而分离逻辑则可以很好地描述涉及空间分离的问题,因此将分离逻辑中空间分离概念引入到时段演算中,就可以将DDS并行模型形式化.按此思路,将分离逻辑与时段演算结合起来,形成一个新的形式系统,记为DC*.下面将对其进行详细介绍,并以此为基础,尝试使用DC*对DDS并行模型进行形式化.

2.1 DC*的语法和语义

在完成DDS并行模型的形式化之前,首先介绍DC*的语法、语义等逻辑设置.

2.1.1 DC*的字母表

定义 7. DC*的字母表定义如下:

(1) 全局变量:$x,y,z,\ldots$;全局变量是实值变量;

(2) 时态变量:$l,v,v',\ldots$;时态变量是实值时间区间函数,其中,l 是计算时间区间长度的函数,即

$$l:\{[a,b]\mid a,b\in R\wedge a<b\}\to R \text{ 且 } l([a,b])=b-a;$$

(3) 全局函数:$f,g,h,\ldots$;全局函数不依赖于时间和时间区间;

(4) 全局关系:$G,H,\ldots$;全局关系不依赖于时间和时间区间,取布尔值{0,1};

(5) 时态命题字母:$X,Y,Z,\ldots$;时态命题字母是取值为布尔值{0,1}的时间区间函数;

(6) 空间变量:$Space,Space_i(i\in\alpha)$;它们是把时间映到空间大小的时间函数,其中,$Space(t)$表示在$t$时刻当前活动空间的大小,$Space_i(t)$表示在$t$时刻任务$p_i$所需的活动空间大小;

(7) 常量:0,1,2,…,$D,d_i(i\in\alpha)$,其中,D,d_i是空间常量且$|D|<\infty$;

(8) 状态变量:$Run_i,Std_i,Urg_{ij},\ldots$;状态变量是时间点上的布尔函数,其中:$Run_i(t)=1$ 当且仅当在t时刻任务p_i在被处理,$Run_i(t)=0$ 当且仅当在t时刻任务p_i未被处理;$Std_i(t)=1$ 当且仅当在t时刻任务p_i在向处理器提出处理请求,$Std_i(t)=0$ 当且仅当在t时刻任务p_i未向处理器提出处理请求,即此时任务p_i的处理器时间已经得到满足;$Urg_{ij}(t)=1$ 当且仅当在t时刻任务p_i比任务p_j紧急,即p_i离截止期时间比p_j离截止期时间短,$Urg_{ij}(t)=0$ 则相反.

(9) 逻辑连接词:$\neg,\vee,\wedge,\to,\leftrightarrow,\exists,\forall,\frown,*$.其中,"*"是分离合取词.

2.1.2 DC*的形成规则

DC*的形成规则与DC的形成规则类似,具体有:

定义 8. DC*的项θ递归定义如下:

$$\theta ::= x \mid l \mid c \mid \int S \mid f(\theta_1,\theta_2,...,\theta_n),$$

其中,f是一个 n 元函数符号,c 是常量,$\int S$ 是 S 的区间积分值,而 S 是一个状态表达式,其递归定义是

$$S ::= 0 \mid 1 \mid P \mid Space = Space_i \mid S_1 \vee S_2 \mid S_1 \wedge S_2 \mid \neg S,$$

这里,P是状态变量,$Space$和$Space_i$是空间变量.

项被解释为定义在区间集合上的实值函数,当区间是$[a,b]$时,项$\int S$解释为布尔值函数 S 在区间$[a,b]$上的积分值.

定义 9. 原子公式 $atom$ 的定义是

$$atom ::= \text{true} \mid c \mid X \mid R(\theta_1,\theta_2,...,\theta_n).$$

这里,c是常量,X是时态命题字母,R是一个n元关系符号,$\theta_1,\theta_2,\ldots,\theta_n$是项.

定义 10. 合式公式ϕ的定义是

$$\phi ::= atom \mid \neg\phi \mid \phi \vee \varphi \mid \phi \wedge \varphi \mid \phi^{\frown}\varphi \mid \exists x.\phi \mid \forall x.\phi \mid \phi * \varphi.$$

合式公式解释为定义在区间集合上的真值函数.

此外,DC*还沿用了DC的一些简写记号:

$\Diamond\phi \triangleq \text{true}^{\frown}\phi^{\frown}\text{true}$,它表示:对于某些子区间$\phi$成立;

$\Box\phi \triangleq \neg\Diamond(\neg\phi)$,它表示:对于所有的子区间$\phi$成立;

$\Diamond_p\phi \triangleq \phi^{\frown}\text{true}$,它表示:对于某些前置区间$\phi$成立;

$\Box_p\phi \triangleq \neg\Diamond_p(\neg\phi)$,它表示:对于任意前置区间$\phi$成立;

$[\![\ ]\!] \triangleq l = 0$;

$[\![S]\!] \triangleq \int S = l \wedge l > 0$;

$\overset{n}{\underset{i=1}{\odot}} p_i \triangleq p_1 * p_2 * ... * p_n$,即重复式分离合并句.

2.1.3 DC*的推理规则

假定实数公理系统的语言及关系符号及其全体公理在DC*中适用,又假定时段演算中的所有符号和公理、推理规则以及分离逻辑中涉及的空间分离公理、推理规则和一些自然数的公理也是在DC*中适用的.

2.2 DDS并行模型的形式化

下面使用DC*的语言对DDS并行模型进行形式化.

在t时刻,任务p_i与任务p_j是空间相容的,可以表示为

$$(Space(t) = Space_i(t) * Space(t) = Space_j(t)) \wedge Space(t) \le D.$$

在时间区间$[a,b]$,任务p_i与任务p_j是空间相容的,可以表示为

$$([\![Space(t) = Space_i(t)]\!] * [\![Space(t) = Space_j(t)]\!]) \wedge [\![Space(t) \le D]\!].$$

在t时刻,任务组$Q=\{p_i \mid i \in \beta\}$是极大空间相容的,可以表示为

$$\odot_{i\in\beta}(Space(t) = Space_i(t)) \wedge Space(t) \le D \wedge \forall j \notin \beta.\left(\sum\nolimits_{i\in\beta} Space_i(t)\right) + Space_j(t) > D.$$

在时间区间$[a,b]$,任务组$Q=\{p_i \mid i \in \beta\}$是极大空间相容的,可以表示为

$$\odot_{i\in\beta}[\![Space(t) = Space_i(t)]\!] \wedge [\![Space(t) \le D]\!] \wedge [\![\forall j \notin \beta.\left(\sum\nolimits_{i\in\beta} Space_i(t)\right) + Space_j(t) > D]\!].$$

设$Q=\{p_i \mid i \in \beta\}$是一个极大空间相容组,则$Q$存在一个主导任务,设为$p_i$,主导任务可以用下面的公式刻画:

$$\bigwedge_{j\in\beta, j\neq i}\left(\frac{C_i}{T_i}\geq\frac{C_j}{T_j}\right).$$

在完成极大空间相容概念和主导任务的刻画后,接下来实现主调度的具体刻画.

由定义 4 及定理 2 可知:存在一个主调度算法,使得极大空间相容组Q的其他非主导任务可用Q的主导任务来代表.该算法可用如下公式Γ刻画[3]:

$$\Gamma \triangleq \exists i,i_1,i_2,...,i_{k-1}\in\beta.\bigwedge_{j\in\beta,j\neq i}\left(\frac{C_i}{T_i}\geq\frac{C_j}{T_j}\right)\wedge\left(\frac{C_i}{T_i}\geq\frac{C_{i_1}}{T_{i_1}}\geq...\geq\frac{C_{i_{k-1}}}{T_{i_{k-1}}}\right)\wedge[\![Run_{i_{k-1}}\rightarrow Run_{i_{k-2}}]\!]\wedge$$
$$[\![Run_{i_{k-2}}\rightarrow Run_{i_{k-3}}]\!]\wedge...\wedge[\![Run_{i_1}\rightarrow Run_{i_1}]\!]\wedge PrR^Q\wedge Req^Q,$$

其中,

$$PrR^Q\triangleq\Box_p\bigwedge_{i\in\beta}[(T_i\,|\,l\widehat{\ }(l\leq T_i\wedge([\![\ ]\!]\vee([\![Std_i]\!]\widehat{\ }\text{true}))))\wedge((T_i\,|\,l\widehat{\ }(l\leq T_i\wedge\Diamond[\![\neg Std_i]\!]))\rightarrow(T_i\,|\,l\widehat{\ }(l\leq T_i\wedge\int Run_i=C_i)))\wedge$$
$$(\neg(T_i\,|\,l\widehat{\ }(l\leq T_i\wedge((\int Run_i=C_i)\widehat{\ }\Diamond[\![Std_i]\!])))),$$
$$Req^Q\triangleq\bigwedge_{i\in\beta}\Box_p(\int Run_i\geq\lfloor l/T_i\rfloor\cdot C_i).$$

现在考虑全局算法.首先对P的划分进行形式化.以A_i表示任务组Q_i是i级极大空间相容的,以I记P中任务的标号集,以α_i记Q_i中任务的标号集,则有

$$A_i\triangleq\bigodot_{j\in\alpha_i}([\![Space(t)=Space_j(t)]\!])\wedge[\![Space(t)\leq D]\!]\wedge\left[\!\!\left[\forall l\in\left(I-\bigcup_{s=1}^{i}\alpha_s\right).\left(\sum\nolimits_{j\in\alpha_i}Space_j(t)\right)+Space_l(t)>D\right]\!\!\right].$$

将P划分为极大空间相容组$Q_1,Q_2,\ldots,Q_k$可以形式化为

$$\bigwedge_{i=1}^{k}A_i\wedge\left(\bigcup_{i=1}^{k}\alpha_i=I\right)\wedge\forall i\neq j.(\alpha_i\cap\alpha_j=\varnothing).$$

同时,用$\Gamma_1,\Gamma_2,\ldots,\Gamma_k$分别表示极大空间相容组$Q_1,Q_2,\ldots,Q_k$的主调度算法,即

$$\Gamma_i\triangleq\exists i_1,i_2,...,i_{|\alpha_i|}\in\alpha_i.\left(\bigwedge_{j\in\alpha_i,j\neq i_1}\left(\frac{C_{i_1}}{T_{i_1}}\geq\frac{C_j}{T_j}\right)\right)\wedge\left(\frac{C_{i_1}}{T_{i_1}}\geq\frac{C_{i_2}}{T_{i_2}}\geq...\geq\frac{C_{i_{|\alpha_i|}}}{T_{i_{|\alpha_i|}}}\right)\wedge[\![Run_{i_{|\alpha_i|}}\rightarrow Run_{i_{|\alpha_i|-1}}]\!]\wedge$$
$$[\![Run_{i_{|\alpha_i|-1}}\rightarrow Run_{i_{|\alpha_i|-2}}]\!]\wedge...\wedge[\![Run_{i_2}\rightarrow Run_{i_1}]\!]\wedge Req^{Q_i}\wedge PrR^{Q_i},$$

又分别以$ShP^{\overline{Q}},PrR^{\overline{Q}},Sch^{\overline{Q}},Req^{\overline{Q}}$表示主导任务组$\overline{Q}=\{p_{i_1}\,|\,i=1,2,...,k\}$的相应公式,记$\overline{I}=\{1,2,...,k\}$,其中,

$$ShP^{\overline{Q}}\triangleq\Box\bigwedge_{i\in\overline{I}}(([\![Run_{i_1}]\!]\rightarrow[\![Std_{i_1}]\!])\wedge([\![Run_{i_1}]\!]\rightarrow\bigwedge_{j\neq i}[\![\neg Run_{j_1}]\!])),$$
$$Sch^{\overline{Q}}\triangleq\Box_p\left(\bigwedge_{i,j\in\overline{I}}Urg_{i_1j_1}\right)\wedge\left(\Box\bigwedge_{i,j\in\overline{I}}\neg[\![Urg_{i_1j_1}\wedge Run_{j_1}\wedge Std_{i_1}]\!]\right)\wedge\left(\bigwedge_{i\in\overline{I}}\Box([\![Std_{i_1}]\!]\rightarrow\left[\!\!\left[\bigvee_{j\in\overline{I}}Run_{j_1}\right]\!\!\right]\right).$$

全局调度算法可由下面的公式刻画:

$$\bigwedge_{i=1}^{k}A_i\wedge\left(\bigcup_{i=1}^{k}\alpha_i=I\right)\wedge\forall i\neq j.(\alpha_i\cap\alpha_j=\varnothing)\bigwedge_{i=1}^{k}\Gamma_i\wedge ShP^{\overline{Q}}\wedge PrR^{\overline{Q}}\wedge Sch^{\overline{Q}}\wedge Req^{\overline{Q}}.$$

3 结束语

本文是在“并行和并发实时进程及其通讯和控制系统的形式化”这一研究课题下进行研究的,是该课题研究的阶段性成果,后续还有很多工作值得进一步深入研究.分离逻辑是当今国际计算机理论界研究的新热点,经过近几年的研究已经发展出了很多研究分支,如对分离逻辑进行扩充就是一个新的研究方向.本文将分离逻辑引入时段演算中,从而推演出DC*,以便更好地形式化DDS并行模型的做法,就是在这一研究方向中所作的一种尝试.后续工作是,可以进一步研究DC*系统的可靠性、完备性等逻辑性质.此外,关于程序进程实时性的讨论也是后续研究的一个方向.

本文作者是广西师范大学逻辑及程序理论小组的成员,由于本研究小组几年前的人力、物力基础比较薄弱,在本文的研究中也体现出了作者在创新性和学术水平上的不足.在今后的工作中,我们会继续努力,争取更大的进步.

致谢 我们向对本文的工作给予支持和建议的同行表示衷心的感谢,尤其是中国科学院软件研究所周巢尘院士、张健研究员,他们不辞辛苦地访问了广西师范大学并且向我们介绍了分离逻辑、时段演算等前沿课题,给予了我们很大的鼓舞,在此再次向他们表示感谢!

References:

[1] Krisna CM, Shin KG, Wrote; Dai QH, Trans. Real-Time Systems. Beijing: Tsinghua University Press, 2004 (in Chinese).

[2] Liu CL, Layland JW. Scheduling algorithms for multiprogramming in a hard real-time environment. Journal of the Association for Computing Machinery, 1973,20(1):46−61.

[3] Zhou CC, Hansen MR. Duration Calculus : A Formal Approach to Real-Time Systems. Berlin: Springer-Verlag, 2004.

[4] Reynolds JC. Separation logic: A logic for shared mutable data structures. In: Plotkin G, ed. Proc. of the 17th Annual IEEE Symp. on Logic in Computer Science. Los Alamitos: IEEE Computer Society, 2002. 55−74.

[5] Hoare CAR. An axiomatic basis for computer programming. Comunications of the ACM, 1969,12(10):576−580, 583.

[6] Lu RQ. The Formal Semantics of Computer Languages. Beijing: Science Press,1992 (in Chinese).

[7] Brooks S. A semantics for concurrent separation logic. Theoretical Computer Science, 2007,375:227−270.

[8] Biering B, Birkedal L, Torp-Smith N. Bi-Hyperdoctrines and higher order separation logic. In: Sagiv M, ed. Proc. of the ESOP 2005: The European Symp. on Programming. Berlin: Springer-Verlag, 2005. 233−247.

[9] O'Hearn PW. Resource, concurrency, and local reasoning. Theoretical Computer Science, 2007,375:271−307.

[10] Luo WM, Lin C, Yan BP.A Survey of Congestion Control in the Internet. Chinese Journal of Computers, 2001,24(1):1−18 (in Chinese with English abstract).

附中文参考文献:

[1] Krisna CM,Shin KG,著;戴琼海,译.实时系统.北京:清华大学出版社,2004.

[6] 陆汝钤.计算机语言的形式语义.北京:科学出版社,1992.

[10] 罗万明,林闯,阎保平.TCP/IP 拥塞控制研究.计算机学报,2001,24(1):1−18.

刘真环(1982－),女,广西南宁人,硕士,主要研究领域为数理逻辑.

赵荣盛(1979－),男,硕士,主要研究领域为数理逻辑.

韦立(1981－),男,硕士,主要研究领域为数理逻辑.

王驹(1950－),男,博士,研究员,博士生导师,主要研究领域为数理逻辑,人工智能.

陈艳(1979－),女,硕士,主要研究领域为数理逻辑.