

# 基于合同网模型的多代理协作研究

张文波<sup>1,2</sup>, 赵海<sup>1</sup>, 苏威积<sup>1</sup>, 徐野<sup>1</sup>, 王金东<sup>1</sup>, 韩光洁<sup>1</sup>

(1. 东北大学信息科学与工程学院, 辽宁沈阳 110004; 2 沈阳理工大学信息科学与工程学院, 辽宁沈阳 110168)

**摘要:** 传统的合同网模型通过招标服务代理和投标服务代理之间的招投标, 可以在 Internet 这样的开放异构环境下实现任务的协作求解. 但传统合同网模型也存在一些如协作过程中网络通信量较大以及对服务代理能力和负载变化缺乏适应性问题. 为此, 本文提出了等级域和时间消耗的概念, 并将其引入到传统合同网中, 建立了一个基于等级域和时间消耗的合同网模型. 根据服务代理的能力和协作成功率, 在传统合同网功能域中划分了三个等级域, 使得服务代理间的协作大量发生在第一等级, 同时在选择服务代理时, 考虑了其时间消耗的因素. 理论分析和实验表明, 这个新模型能够降低查找协作服务范围, 降低协作求解过程中的通信量, 而且能使系统中服务代理的负载更均衡.

**关键词:** 服务代理; 协作求解; 等级域; 时间消耗; 合同网模型; 等级域-时间消耗合同网模型

**中图分类号:** TP316 **文献标识码:** A **文章编号:** 0372-2112 (2006) 05-0837-08

## Research on Cooperation of Multiple Agent Based on Contract Net Model

ZHANG Wen-bo<sup>1,2</sup>, ZHAO Hai<sup>1</sup>, SU Wei-ji<sup>1</sup>, XU Ye<sup>1</sup>, WANG Jin-dong<sup>1</sup>, HAN Guang-jie<sup>1</sup>

(1 School of Information, Northeastern University, Shenyang, Liaoning 110004 China;

2 School of Information, Shenyang Ligong University, Shenyang, Liaoning 110168 China)

**Abstract** In open environments such as the Internet agents from heterogeneous sources could exist and interact to accomplish joint activities. The general Contract Net Protocol model which works on bids invitation between a Manager agent and Contractor agents can successfully realize the cooperation among agents. But it also faces many problems such as the high network traffic and low adaptation to the changes of the contractors' capacity and load. So in this paper the concepts of grade field and time consumption are put forward and introduced into the general contract net protocol and a new model of grade field and time consumption contract net model is made. By dividing three different grades subfields according to the capability and success rate in all the service agents, the interoperation of the service agents is limited in grade field 1. The theoretic analysis and the experiments show that this new model can reduce the searching range and lighten the network traffic. Furthermore it can balance the load of the contractors.

**Key words** service agent cooperative problem-solving grade field; time consumption; contract-net grade field and time consumption contract-net model

## 1 引言

合同网模型是 Smith 等在研究分布式求解问题时提出来的, 后来被广泛地应用于多代理系统的协调中, 它把一个多代理系统看作是一个由多个节点组成的合同网, 每个节点是一个代理. 与黑板模型和多服务代理规划等分布式计算模型相比, 合同网模型以其分布式控制和实现简单的特点而应用最广; 黑板模型是一种集中控制的协作模型, 在协作的过程中, 对“黑板”的运行速度、内存空间和通信

处理能力的要求比较高, 而且当“黑板”发生故障时, 整个系统间的协作将陷入瘫痪; 多服务代理规划模型由于不适应动态变化的环境, 以及在工程实际中遇到的各种障碍, 其应用较少.

多代理系统是指由多个代理组成的系统, 它是为了解决单个代理不能解决的复杂问题, 由多个代理协调合作形成的问题求解网络<sup>[1]</sup>. 实际上, 多代理系统的研究方法是用模拟人类社会系统运作机制来提高计算机系统解决复杂问题的能力. 每个代理能够完成问题求解中的一部分

工作,能够随环境而修改自己的行为,并且与其他代理进行通信和协作<sup>[2]</sup>.

随着对其功能需求的日益增强,在执行较复杂任务时,任务的复杂性和时间紧迫性常常使其无法在规定的时限<sup>[3,4]</sup>内完成,而且经常需要人去干预和协调.为了将人的注意力回归到要完成的任务本身,使代理自主交互完成所需要的服务,就要求解决服务访问过程的透明性问题<sup>[5]</sup>,即通过代理间高效的协作来求解复杂任务.合同网模型是代理间最常使用的协作模型,研究发现,传统的合同网模型在协作求解过程中仍然存在以下缺点:

(1)传统合同网模型在进行招标时要向所有可能的任务承揽者发送标书,并等待所有的投标回答或者截止时间到来,这在承揽者较多时会带来较大的通信量和时间开销,并且浪费不相关的服务代理的计算资源;

(2)传统合同网模型无法了解和适应任务承揽方的能力及其承担的协作任务的动态变化,常常导致协作任务过于集中.一方面使得任务的等待时间增大,另一方面却导致有些任务承揽方的负载严重不均衡.

为此,本文在传统合同网模型<sup>[6,7]</sup>功能域的基础上,引入了等级域和时间开销的概念,建立了等级域-时间开销合同网模型,克服了合同网中服务代理能力<sup>[8]</sup>恒定不变的假设,减少了协作过程引起的网络通信量,避免了对不相关服务代理的时间开销,均衡了协作任务的分布.

### 2 等级域-时间消耗合同网

服务代理<sup>[9,10]</sup>是一个具有问题求解机理的计算单元或功能实体,是一个能在动态环境中灵活、自动完成某种功能的(微)计算机系统.所谓多服务代理系统 MAS(Multi-Agent System),是由多个可计算的服务代理<sup>[11]</sup>组成的集合,其中每个服务代理是一个物理的或抽象的实体,可作用于自身和环境,并与其他的服务代理进行通讯.域(Field)是一组服务代理的集合,全局域(Global Field)作为缺省域预先存在,所有的服务代理在产生时都被认为属于全局域.

#### 2.1 功能域的定义与划分

定义1 功能域(Function Field)是域中各个服务代理按照其提供的服务功能而划分的子域.

根据服务代理所提供的服务功能,域可划分为各种不同的功能域,功能域的作用就在于各个服务代理按照其提供的服务功能进行划分之后,同一个域内的各服务代理之间具有很强的相关性,协作的可能性比较大,因而在协作求解时选择协作对象的速度快,效率高.功能域的模式如图1所示.

在功能域模型中,所有的服务代理都属于全局域.全局域又按照各服务代理所提供的功能分为几个不同的功能域,域1、域2、域3等.每个域内包括一些可提供相同服务的代理.任何两个功能域之间可以相互重叠,即功能域内的某些服务代理可以分属几个不同的功能域.

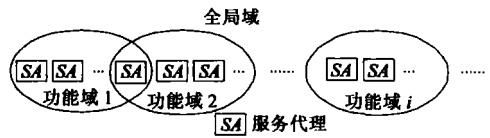


图1 功能域模型

#### 2.2 等级域的定义与等级跃迁

定义2 等级域(Grade Field)是把功能域内各个服务代理根据它们提供服务的能力和成功率分成三个不同的等级.第一等级,有服务能力且成功率较大的服务代理集合;第二等级,有服务能力且成功率较低的服务代理集合;第三等级,暂时无服务能力或服务代理集合.等级域模型如图2所示.

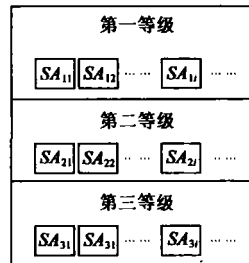


图2 等级域模型

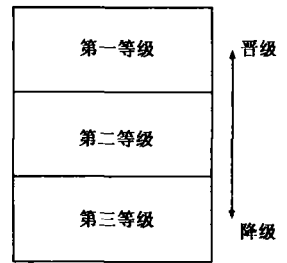


图3 等级跃迁

定义3 等级跃迁(Grade Transition)是指等级域中的服务代理根据自己能力和协作成功率的变化,进入高等级的集合(晋级)或被淘汰而进入低等级的集合(降级)的过程,如图3所示.

#### 2.3 等级跃迁的条件

服务代理的“晋级”和“降级”与两个因素有关,即各个服务代理能力的变化和跃迁规则的变化.服务代理能力的变化包括两个方面,服务代理自身能力的变化和其他服务代理能力的变化.

服务代理自身能力的变化通常是由于自身软件或硬件的升级造成的,或者是由于服务代理负载的动态变化造成任务协作求解成功率的变化而引起的,也可能是由于其他原因造成的成功率的波动(如干扰、移动等)而导致的.

其他服务代理能力的变化会对自身所处等级“地位”产生间接影响.例如,如果其他服务代理的能力和协作成功率得到了提高,则会对自己造成较大的竞争压力;如果其他服务代理的能力和成功率降低了,则会相对提高自己的竞争力.

#### 2.4 等级跃迁算法

根据各个服务代理的能力和任务协作求解的成功率的不同,划分了三个两两互不相交的等级,  $GndField1$ ,  $GndField2$ ,  $GradeField3$  令  $GradeField$  表示等级域中所有服务代理组成的集合.用  $SA$  表示招标服务代理,用  $SA_i$  表示等级域中的任意服务代理,用  $SA_{1n}$ ,  $SA_{2n}$ ,  $SA_{3n}$  分别表示这三个等级集合中的任意服务代理.

定义4 成功率是对服务代理  $SA_i$  完成任务  $Task$  情况

的一个综合的评价, 可以用符号  $Succeed\_Rate(SA_i, Task)$  来表示, 并且  $0 \leq Succeed\_Rate(SA_i) \leq 1$  鉴于篇幅限制, 本文只是简单地利用了判断是否成功的结果, 至于如何判断协作成功等问题这里不过多涉及。

为了实现等级跃迁, 本文采用了随机 TOP-N 算法<sup>[12]</sup>。TOP-N( $X$ )表示用  $X$  这种标准来度量各个服务代理, 其值的顺序排在前  $N$  位的  $SA_i$ 。文中使用成功率来度量各个  $SA_i$ ,  $N$  的值可以根据人为需要来进行设置, 这个值限定了  $GradeField1$  中  $SA_i$  的数量。文中规定  $Threshold$  是一个常量, 用来表示各个服务代理  $SA_i$  进入第一等级  $GradeField1$  的成功率的下限值。其值可以根据需要进行设置或者根据某种条件进行动态调整。文中设  $|GradeField1| \leq N$ 。如果  $GradeField1 = \phi$ , 那么应调整  $Threshold$  的值使  $GradeField1 \neq \phi$ 。

服务代理  $SA$  在  $GradeField1$  中选择协作服务代理  $SA_i$ , 求解任务  $Task$  时采用随机选择方法, 而不是选择协作成功率最大的那个服务代理, 这样既可以保证在  $SA_i$  中均衡负载, 同时又可以适应这些  $SA_i$  能力的变化。

为了能够遍历  $GradeField2$  和  $GradeField3$  每当在  $GradeField1$  中选取协作服务代理  $SA_{i1}$  进行任务  $Task$  委托 (或分配)  $|GradeField1|$  ( $|GradeField1|$  为  $GradeField1$  的势, 令  $l$  为一常数且  $l \geq 1$ ) 次后, 就在  $GradeField2$  中选取服务代理  $SA_{2i}$  一次, 并根据其完成任务的情况, 决定该服务代理  $SA_{2i}$  是否可以进入  $GradeField1$ 。

每当  $GradeField2$  中的服务代理  $SA_{2i}$  被遍历一次后, 就在  $GradeField3$  中选取服务代理  $SA_{3i}$  一次, 但是对  $SA_{3i}$  不进行任务  $Task$  的委托, 只是获取它的最新能力情况  $Can\_Do(SA_{3i}, Task)$ , 并由此决定  $SA_{3i}$  是否可以进入  $GradeField2$ 。

等级跃迁算法概述:

(1) 对等级域中的三个等级集合进行初始化:

(a) 分别把等级域中的三个等级集合  $GradeField1$ ,  $GradeField2$  和  $GradeField3$  设置为空。

(b) 如果等级域中已经有其他服务代理  $SA$ , 执行过任务  $Task$ , 那么招标服务代理  $SA$  则可以向该服务代理  $SA_i$  学习以便完成三个等级的初始化。

如果等级域中不存在对任务  $Task$  的协作求解有经验的服务代理  $SA$ , 那么  $SA$  可以在等级域中广播以便获取各个  $SA_i$  的能力。

(2) 服务代理的任务委派和等级跃迁

可以假设  $x$ ,  $y$  和  $z$  分别表示  $SA$  在三个等级中选择协作  $SA_i$  的计数, 即访问次数对集合的势的模,

$$x \bmod |GradeField1|,$$

$$y \bmod |GradeField2|,$$

$$z \bmod |GradeField3|.$$

①如果  $SA$  在  $GradeField1$  中选取  $SA_{i1}$  的次数小于  $GradeField1$  的势的  $l$  倍, 即  $x < l|GradeField1|$ , 则选择次数

计数器  $x$  增 1:  $x \leftarrow x + 1$ , 然后在  $GradeField1$  中随机选择一个  $SA_{i1}$  进行任务  $Task$  委托, 并协助其完成任务, 最后对其进行成功率评价, 并转向 ④。

②如果  $SA$  在  $GradeField1$  中选取  $SA_{i1}$  的次数已经大于等于  $GradeField1$  的势的  $l$  倍, 即  $x \geq l|GradeField1|$ , 同时  $SA$  在  $GradeField2$  中选取协作服务代理  $SA_{2i}$  的次数还小于  $GradeField2$  的势, 即  $y < |GradeField2|$ , 则重置  $GradeField2$  的选择次数计数器  $x$ , 即  $x \leftarrow 0$  并使  $GradeField2$  中的选择次数计数器  $y$  增 1, 即  $y \leftarrow y + 1$ , 然后在  $GradeField2$  中选取第  $y$  个服务代理  $SA_{2i}$  进行任务  $Task$  委托, 并协助其完成任务, 最后对其进行成功率评价, 转向 ⑤。

③如果  $SA$  在  $GradeField1$  中选取协作  $SA_{i1}$  的次数已经大于等于  $GradeField1$  的势的  $l$  倍, 即  $x \geq l|GradeField1|$ , 同时  $SA$  在  $GradeField2$  中选取  $SA_{2i}$  的次数也大于等于  $GradeField2$  的势, 即  $y \geq |GradeField2|$ , 但  $SA$  对  $GradeField3$  中各个服务代理  $ServiceAgents_i$  能力更新情况的遍历还没有结束, 即  $z < |GradeField3|$ , 则重置  $GradeField1$  和  $GradeField2$  的选择次数计数器, 即  $x \leftarrow 0$ ,  $y \leftarrow 0$  并对  $GradeField3$  的选择次数计数器  $z$  增 1, 即  $z \leftarrow z + 1$  然后在  $GradeField3$  中选取服务代理  $SA_{3i}$ , 通过通信更新其最新能力情况。如果  $GradeField3$  中的每个服务代理都被遍历过, 即  $z > |GradeField3|$ , 则重置选择次数计数器, 即  $z \leftarrow 0$  进入下一轮遍历。最后转向 ⑥。

④如果所选的服务代理  $SA_{\Delta} \in GradeField1$ , 但其对任务  $Task$  的协作成功率  $Succeed\_Rate(SA_{\Delta}, Task) < Threshold$ , 则将  $SA_{\Delta}$  从  $GradeField1$  中删除, 添加到  $GradeField2$  中。最后转向 ⑦。

⑤如果  $SA_{\Delta} \in GradeField2$ , 其对任务  $Task$  的协作成功率大于等于设定的阈值  $Threshold$ , 同时  $GradeField1$  的势小于设定的阈值  $N$ , 则将  $SA_{\Delta}$  从  $GradeField2$  中删除, 直接添加到  $GradeField1$  中, 最后转向 ⑦。

如果  $SA_{\Delta} \in GradeField2$ , 其对任务  $Task$  的协作成功率大于等于设定的阈值  $Threshold$ , 但是  $GradeField1$  的势已经达到了设定的阈值  $N$ , 并且其对任务  $Task$  的协作求解的成功率大于  $GradeField1$  中各个  $SA_{i1}$  协作求解成功率的最小值, 则首先把  $GradeField1$  中对任务  $Task$  协作求解成功率最小的服务代理迁移到  $GradeField2$  中, 然后把所选择的  $SA_{\Delta}$  从  $GradeField2$  中迁移到  $GradeField1$  中, 最后转向 ⑦。其他任何情况直接转向 ⑦。

⑥如果所选择的  $SA_{\Delta} \in GradeField3$  并且其有能力完成任务  $Task$ , 则将其跃迁到  $GradeField2$  中, 然后转向 ⑦。其他情况直接转向 ⑦。

⑦结束。

## 2.5 时间开销

定义 5 多服务代理之间协作求解的任务是由一系列的子任务组成的, 我们把子任务称为原子行为。

任务  $Task$  可以用原子行为  $AtomAction$  表示:  $Task =$

$\{AtomAction_i | AtomAction_i \in Task\} \neq \emptyset$

定义 6 任意两个任务  $Task_{k_i}$  和  $Task_{k_j}$  由任务  $Task_{k_i}$  和  $Task_{k_j}$  的所有共同原子行为组成的集合  $C$ , 称为任务  $Task_{k_i}$  和  $Task_{k_j}$  的交集, 记作  $Task_{k_i} \cap Task_{k_j}$ .

定义 7 服务代理  $SA_A$  执行协作任务  $Task_{k_i}$  的时间开销称为  $SA_A$  的执行开销, 用符号  $ExecuteT in e_{SA_A}(Task_{k_i})$  来表示.

定义 8 服务代理  $SA_A$  执行任务  $Task_{k_i}$  和  $Task_{k_j}$  的交集中的所有原子行为的时间开销称为  $SA_A$  的叠和时间开销, 用  $CooperateT in e_{SA_A}(Task_{k_i}, SA_B, Task_{k_j})$  来表示.

定义 9 服务代理  $SA_B$  执行服务代理  $SA_A$  的协作任务  $Task_{k_i}$  而需要附加的时间开销 (内存分配时间、通信时间等) 称为  $SA_B$  的附加开销, 用  $AdditiveT in e_{SA_B}(Task_{k_j}, SA_A, Task_{k_i})$  来表示.

定义 10 在服务代理  $SA_A$  和  $SA_B$  一次协商的过程中,  $SA_B$  的付出开销定义为:  
 $Payouff in e_{SA_B}(Task_{k_j}, SA_A, Task_{k_i}) = ExecuteT in e_{SA_A}(Task_{k_i}) - CooperateT in e_{SA_B}(Task_{k_j}, SA_A, Task_{k_i}) + AdditiveT in e_{SA_B}(Task_{k_j}, SA_A, Task_{k_i})$

定义 11 服务代理  $SA_B$  在  $N$  次协商之后, 付出的总开销  $OverallT in e_N(SA_B)$  定义为:

$$OverallT in e_N(SA_B) = \sum_{i=1}^N Payouff in e_{SA_B}$$

其中,  $Payouff in e_{SA_B}$  表示服务代理  $SA_B$  在第  $i$  次协商中付出的开销;  $N$  表示计算的当前时刻服务代理  $SA_B$  对还没有求解结束的任务的协商次数.

### 2.6 等级域-时间开销合同网模型

在合同网已划分的功能域的基础上, 引入等级域和时间开销, 可以画出图 4 所示的基于等级域-时间开销的合同网模型 (GT-CNM). 在该模型中, 招标服务代理 TTN 首先根据“等级域”的规则, 向第一等级集合中的所有的投标服务代理 TBN 发送任务协作求解请求, 请求的消息中包含任务  $Task$  的基本信息和相应的时间开销信息. 投标服务代理 TBN 接收到招标信息后, 在假设自己能够中标的前提下,

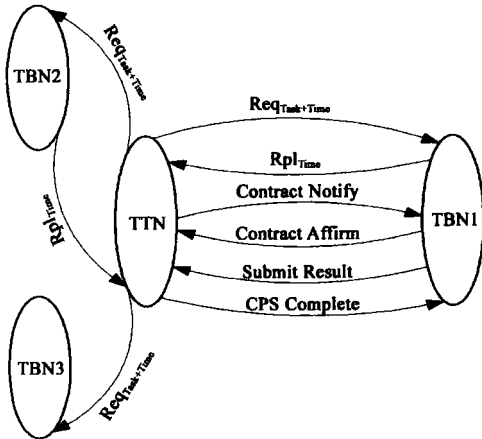


图 4 基于等级域-时间开销的合同网模型

下, 首先计算这个新的任务与正在排队执行的任务之间的交集, 然后据此计算自身的各项时间开销, 并携带这些时间开销的计算结果向 TTN 投标.

招标服务代理 TTN 根据负载均衡的原则, 在进行投标的各个服务代理 TBN 中进行时间开销的比较, 选取总开销较小的投标服务代理作为任务求解的协作对象. 如果总开销相等, 则选择付出开销较小的投标服务代理承担任务求解. 如果付出开销也相等, 则随机选择或根据偏好习惯进行. 等级域-时间开销合同网模型的协作过程如图 5 所示.

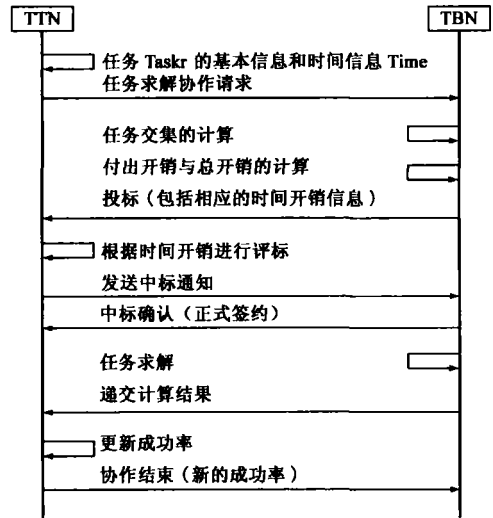


图 5 基于等级域-时间开销的合同网协作过程

### 2.7 算法描述

#### 2.7.1 协作服务代理的选取

首先我们分析协作求解中协作服务代理的选取, 共分以下几个步骤:

(1) 根据等级域模型的规则, 招标服务代理  $SA$  向  $GradeField1$  中的各个服务代理  $SA_i$  发布协作请求.

(2)  $GradeField1$  中的各个  $SA_i$  监听到该协作请求后, 计算任务集合的交集. 对任意两个任务  $Task_{k_i}$  和  $Task_{k_j}$ , 设其交集为  $C$ , 则

$$C = Task_{k_i} \cap Task_{k_j} = \{AtomAction_{AB1}, AtomAction_{AB2}, \dots, AtomAction_{ABP}, \dots\}$$

我们可以分别获得交集内各个原子行为的执行开销, 设在招标服务代理  $SA_A$  中其执行开销为:  $ExecuteT in e_{SA_A}(AtomAction_{ABi})$ ,  $i = 1, 2, \dots, p$ ; 在投标服务代理  $SA_B$  中的执行开销为:  $ExecuteT in e_{SA_B}(AtomAction_{ABi})$ ,  $i = 1, 2, \dots, p$ .

(3) 投标服务代理  $SA_B$  首先根据招标服务代理  $SA_A$  执行任务  $Task_{k_i}$  的执行时间和自身的时钟频率等信息, 计算出自己的执行开销, 即  $ExecuteT in e_{SA_B}(Task_{k_i})$ .

(4) 根据步骤 (2) 中的任务的交集, 假设服务代理  $SA_B$  中标, 计算其相应的叠和时间开销.

(5) 根据  $SA_B$  自身的情况 (内存分配时间、通信时间

等), 计算执行  $SA_A$  的协作任务  $Task_A$  而需要附加的时间开销.

(6) 然后根据  $SA_B$  的执行开销  $ExecuteFin_{e_{SA_B}}(Task_A)$ , 叠和时间开销  $CooperateFin_{e_{SA_B}}(Task_B, SA_A, Task_A)$  和附加开销  $AdditiveFin_{e_{SA_B}}(Task_B, SA_A, Task_A)$ , 计算出  $SA_B$  在这次协商中的付出开销.

(7) 最后计算服务代理  $SA_B$  在该次协商之后, 付出的总开销.

(8)  $SA_B$  向  $SA_A$  返回投标信息, 并携带这次协商后的付出开销和总开销等信息, 供  $SA_A$  在评标的过程中使用.

(9)  $SA_A$  对已经接收到的投标报文进行分析, 假设一共接收到  $L$  个投标报文  $Bidding(SA_i, OverallFin_{e_i}, PayoutTime_i)$ , 其中,  $0 \leq i \leq L$ .

在进行分析时, 首先从  $L$  个总开销  $OverallFin_{e_i}$  中选取其最小值  $OverallFin_{e_{min}}$ , 再根据负载均衡的原则把最小总开销  $OverallFin_{e_{min}}$  对应的服务代理  $SA_{overallFin_{e_{min}}}$  作为任务  $Task_A$  求解的协作服务代理  $SA_{Successful}$ . 如果这样的投标服务代理有多个, 则  $SA_A$  可以选择付出开销的最小值; 如果仍然有多个投标服务代理, 那么招标服务代理就会随机选择其中的一个投标服务代理作为任务  $Task_A$  求解的协作服务代理.

### 2.7.2 协作的执行

招标服务代理  $SA_A$  与中标服务代理  $SA_{Successful}$  之间的协作分为如下几个步骤:

(1)  $SA_A$  根据上述的评标结果向中标的  $SA_{Successful}$  发送中标信息, 开始签订任务  $Task_A$  求解的协作合同; 并向其他投标服务代理发送没有中标的通知, 终止与它们的协商.

(2)  $SA_{Successful}$  接收到中标的通知后, 立刻向  $SA_A$  返回相应的确认信息, 以便完成任务  $Task_A$  求解协作合同的签订.

(3)  $SA_{Successful}$  根据任务  $Task_A$  的情况, 分配内存, 下载任务, 通过通信获取相应的参数, 按照预定义的调度策略进行排队. 当任务  $Task_A$  获得包括处理器 CPU 在内的必要的资源时开始执行, 经过一段时间的运行, 输出相应的计算结果.

(4) 任务  $Task_A$  执行完毕后,  $SA_{Successful}$  把输出的计算结果发送给招标服务代理  $SA_A$ .

(5)  $SA_A$  接收到任务  $Task_A$  的计算结果后, 向  $SA_{Successful}$  返回计算结果的确认信息.

(6)  $SA_A$  根据相应的计算结果对  $SA_{Successful}$  进行的奖惩.

(7) 此次任务  $Task_A$  的协作求解过程结束.

## 3 理论分析和仿真验证

### 3.1 网络通信量的分析

为便于分析, 文中假设在下面的两种模型中, 招标报文均为  $V_{CNM-Inviting} = 1500$  字节, 投标报文的大小均为  $V_{CNM-Bidding} = 1000$  字节, 进一步假设在多服务代理系统中,

共有  $N$  个服务代理, 任务协作求解共发生  $F$  次, 以下计算所得的流量的单位均为字节.

传统合同网模型的网络通信量主要由招标通信量和投标通信量两部分组成, 该模型引起的网络通信量  $V_{CNM}$  为:

$$V_{CNM} = F \times (V_{CNM-Inviting} + N \times V_{CNM-Bidding}) = 1500F + 1000NF$$

而基于等级域合同网模型在进行招投标时引起的网络通信量, 主要由以下三部分组成.

#### (1) 招标前等级域形成时的网络通信量

招标服务代理  $SA$  通过广播并接收每个服务代理的能力信息所引起的网络通信量为  $V_{GT-CNM-Learn2} = 1500 + 500N + 100N$ .

其中, 1500 表示招标服务代理  $SA$  广播的报文大小; 500 表示每个服务代理向招标服务代理报告的自己的能力的大小; 100 表示招标服务代理通知每个服务代理所处等级的报文大小.

#### (2) 招投标过程的通信量

根据上面的假设, 招投标引起的网络通信量为

$$V_{GT-CNM-IB} = F \times (V_{GT-CNM-Inviting} + M \times V_{GT-CNM-Bidding}) = 1500F + 1000MF$$

其中,  $M$  为  $GradeField1$  中服务代理的数量.

#### (3) 任务协作后等级域调整而引起的网络通信量

如果设每次调整的网络通信量 (通知服务代理所处等级的变化) 为  $V_{Adjust} = 1000$  字节. 那么, 根据等级跃迁算法, 招标服务代理在任务协作求解完成后对等级域进行调整所引起的网络通信量  $V_{GT-CNM-Adjust}$  为:

$$V_{GT-CNM-Adjust} = 1000 \times \left( \frac{F}{\|GradeField1\|} + \frac{F}{\|GradeField1\| \times |GradeField2|} \right) = \frac{1000F (\|GradeField2\| + 1)}{\|GradeField1\| \times |GradeField2|}$$

综上所述, 基于等级域的合同网模型在进行招投标时引起的网络通信量为

$$V_{GT-CNM} = N \times V_{GT-CNM-Learn2} + V_{GT-CNM-IB} + V_{GT-CNM-Adjust} = 1500N + 600N^2 + 1500F + 1000MF + \frac{1000F (\|GradeField2\| + 1)}{\|GradeField1\| \times |GradeField2|}$$

至此, 我们可以画出这两个模型的网络通信量曲线, 如图 6(a) 和 (b) 所示. 其中图 6(a) 为三维图, 图 6(b) 是二维图. 需要说明的是曲线  $CNM-GT-CNM$  表示  $CNM$  模型的网络通信量与  $GT-CNM$  模型的网络通信量的差值.

从图 6(a) 和 (b) 可以看出 ( $M = 10, l = 10$ ), 基于等级域 - 时间消耗的合同网模型  $GT-CNM$  网络通信量在本地网络内的所有服务代理的个数  $N$  和任务协作求解的次數  $F$  较小时,  $GT-CNM$  模型的网络通信量大于  $CNM$  模型的

网络通信量,而且在一定的范围内,这种通信量的差距逐渐增大;然后再随着 N 和 F 的增加,两个模型的通信量差距则逐渐减小,一直到通信量相等;这主要是因为招标服务代理在初始化形成等级域的过程中引起的通信量的增加,我们把通信量差距逐渐增大又减小到 0 的过程称之为“初始化效应”。

在“初始化效应”期以后,随着 N 和 F 的增加, CNM 模型的网络通信量迅速超过了 GT-CNM 模型.需要指出的是,本节中对各种报文大小所作的假设并不会影响上面所得出的结论.

### 3.2 负载均衡的分析

对负载均衡的分析,我们通过如下的仿真试验来进行:在一个随机产生的多服务代理系统中,随机产生 25 个任务,按照模型算法的规定,分别计算出各个服务代理的总开销,来衡量其负载状况.共分如下三步:

(1)服务代理的产生.在此,我们按照嵌入式系统计算能力和存储能力,首先随机产生 5 个服务代理 SA<sub>1</sub>, SA<sub>2</sub>, …, SA<sub>5</sub>, 各个

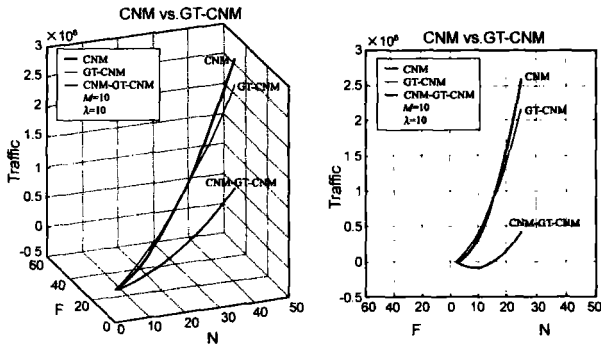


图 6 CNM 和 GT-CNM 的网络通信量

服务代理中微处理器的运算速度和内存的大小由计算机随机产生,具体如表 1 所示.

表 1 服务代理的运算速度和内存

服务代理	运算速度 (M IPS)	内存 (kB)	等级
SA <sub>1</sub>	14 18	19 34	1
SA <sub>2</sub>	10 05	68 22	1
SA <sub>3</sub>	8 57	30 27	1
SA <sub>4</sub>	6 09	54 16	2
SA <sub>5</sub>	3 79	15 08	3

(2)任务的产生.然后我们随机产生 5 组任务,每组任务包含 5 个任务,它们的执行时间 T 和内存要求也是由计算机随机产生的,具体数据如表 2 所示.

需要注明的是,执行时间是以假定的标准运算速度

8M IPS 为基础衡量的,为了便于计算,设所有的附加开销为 1ms 叠和开销为 0ms

(3)按照模型的规定,各个服务代理的任务分布情况如表 3 所示.其中,“-”表示相应服务代理没有协作任务;表中的数据为相应任务的下标.

从表 3 中可以看出,在 GT-CNM 模型中,每一组任务集合中的 Task<sub>i</sub> 都指派给了 SA<sub>i</sub>, 而 SA<sub>i</sub> 在处于 Grade 表 2 任务的执行时间和内存要求

组	Task <sub>1</sub>		Task <sub>2</sub>		Task <sub>3</sub>		Task <sub>4</sub>		Task <sub>5</sub>	
	T (ms)	M (kB)	T	M	T	M	T	M	T	M
1	49.65	21.81	89.97	9.27	82.16	25.15	64.49	17.04	81.79	11.11
2	66.14	21.08	28.44	16.39	46.92	13.34	6.47	20.83	98.83	18.63
3	58.27	23.84	42.34	28.70	51.55	15.67	33.39	26.40	43.29	5.18
4	22.59	29.39	57.98	8.14	76.03	7.56	52.98	26.27	64.05	22.11
5	20.90	4.09	37.98	0.35	78.33	26.81	68.08	5.97	46.10	8.96

表 3 各个服务代理的任务分布情况

组	SA <sub>1</sub>		SA <sub>2</sub>		SA <sub>3</sub>		SA <sub>4</sub>		SA <sub>5</sub>	
	CNM	GT-CNM	CNM	GT-CNM	CNM	GT-CNM	CNM	GT-CNM	CNM	GT-CNM
1	2	2	1, 3, 4	1, 5	5	3	-	4	-	-
2	2	2	1, 3, 4	1, 5	5	3	-	4	-	-
3	3	3	1, 2, 5	1, 5	4	2	-	4	-	-
4	2, 3	2	1, 4	1, 5	5	3	-	4	-	-
5	1, 2, 4	1, 5	3, 5	2	-	3	-	4	-	-

Field2 这说明在 GT-CNM 模型中,处于较低等级的服务代理也有机会执行任务,这完全符合 GT-CNM 模型的设计原则.如果要判断其是否进行了等级跃迁,还需做进一步的假设和计算,在此不对其进行深入的分析.

(4)根据算法的规定,我们可以分别计算出 CNM 模型和 GT-CNM 模型中各个服务代理的负载状况,然后计算出总开销的均方差,如表 4 所示.

表 4 总开销的均方差

组	CNM	GT-CNM
1	73 51	43 49
2	51 32	60 70
3	54 02	33 51
4	51 43	30 53
5	62 82	29 00
平均	58 622	39 446

从表 4 中可以看出,随机生成的 25 个任务经计算后,在 CNM 模型中总开销的均方差的平均值为 58 622 而在 GT-CNM 模型中总开销的均方差的平均值为 39 446 其各个服务代理的负载比 CNM 模型更均衡.

### 3.3 试验验证

在试验过程中,采用了实验室设计开发的基于不同厂商、不同型号的微控制器/微处理器的多个服务代理: WebiV、WebiVP、Webi4A、Internet Access Server、Netter ARM 2410 对这些服务代理的软硬件体系结构,本文不作详述,构造测试系统拓扑结构如图 7 所示.

整个系统共有 6 个投标服务代理: 3 个 WebiV, 2 个 WebiVP, Webi4A, Internet Access Server 和 Netter

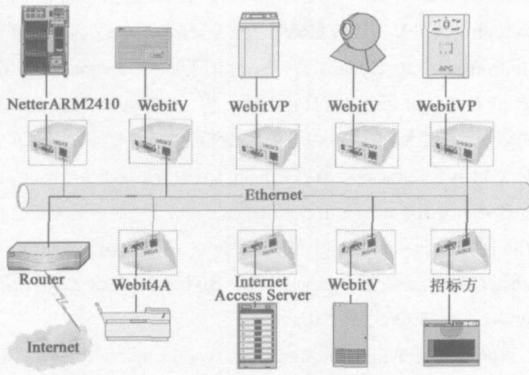


图 7 测试系统的拓扑结构

ARM 2410 虽然基于等级域 - 时间开销的合同网模型可以通过路由器 (Router) 支持广域网内服务代理间的任务协作求解, 但为了测试的方便和比较的一致性暂不考虑这种情况, 仅在局域网内进行测试。

在实际的测试过程中, 随机选择了 6 个互不相交的任务, 相应各个任务的情况如表 5 所示。

表 5 各个任务的情况

任务	运行时间 (ms)	占用内存 (kB)	附加开销 (ms)
1	38.29	29.18	3.40
2	39.59	27.26	6.99
3	86.39	44.27	2.59
4	90.88	29.46	7.26
5	71.36	32.01	4.79
6	19.53	7.62	2.10

表 6 所示为各个服务代理的能力情况。

表 6 各个服务代理的能力情况

服务代理	运行速度 (MIPS)	剩余内存 (kB)	等级
1	57.06	128.42	1
2	8.91	72.03	1
3	6.09	62.31	1
4	5.69	67.96	1
5	4.75	56.39	2
6	3.29	48.99	2
7	4.38	15.85	3
8	3.22	9.47	3

各个服务代理通过一个交换机相连接, 并接入一台美国福祿克网络公司 (Fluke Networks) 生产的集成式网络分析仪 OPV S2 NA (OptView Series II Integrated Network Analyzer) 来收集数据。收集后的通信数据通过美国电子眼数字安全公司 (eEye Digital Security) 研发的著名的网络流量分析监测工具 Iris Network Traffic Analyzer v4.07 进行分析, 察看网络上进入和发出的所有信息流的详细情况。任务的负载分布是通过编程实现的, 各个服务代理记录自己所承担过的任务, 并在查询时返回到用于数据收集的网络分析仪上。CNM 和 GT-CNM 这两个模型的网络通信量情况如图 8 所示。

通过对图 8 的分析, 可以得出如下结论:

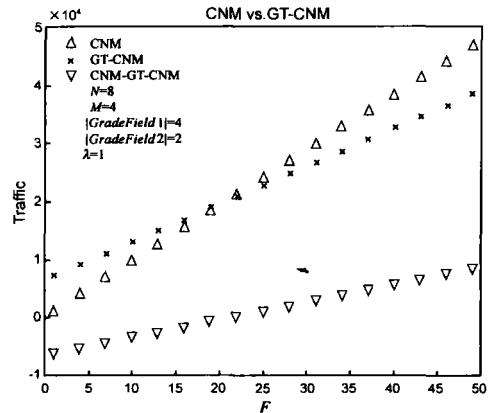


图 8 网络通信量与任务执行次数的关系

(1) 在服务代理的总数  $N$  和各个等级集合的势确定的情况下, 网络的总通信量与任务执行的次数在 CNM 模型和 GT-CNM 模型中均成正比。

(2) 在“初始化效应”期间, GT-CNM 模型在协作求解时引起的网络总通信量比 CNM 模型引起的网络总通信量多, 但在“初始化效应”期之后, GT-CNM 模型引起的网络总通信量要比 CNM 模型引起的网络总通信量少。

(3) 在本案例规定的环境下, CNM 模型和 GT-CNM 模型在进行任务协作求解协商的过程中引起的网络总通信量在执行约 21 次任务时相等。

可以看出, 这些结论符合前面的理论分析结果。

试验收集到的两个模型中各个服务代理的负载分布以及负载均方差如表 7 所示。

表 7 负载分布

服务代理	CNM 任务分布	负载	GT-CNM 任务分布	负载
1	1, 2, 5, 6	186.05	1, 2, 6	164.87
2	4	98.14	4	98.14
3	3	88.98	3	88.98
4	-	0	-	0
5	-	0	5	74.15
6	-	0	-	0
7	-	0	-	0
8	-	0	-	0
负载均方差	CNM	53.60	GT-CNM	44.28

其中, “-”表示相应服务代理没有协作任务; 表中的数据为相应任务的下标。

从表 7 中可以看出, GT-CNM 模型的任务分布比 CNM 模型的任务分布更均衡, 其负载均方差为 44.28 而 CNM 模型的负载均方差为 53.60 另外, GT-CNM 模型还兼顾到培养第二等级中的服务代理, 使其不至于永远得不到执行任务的机会而“饿死”。

但是, 我们也可以从表 7 中看出, 此时的这两个模型的负载均方差的差别不是特别大, 究其原因, 这是因为第一个服务代理的能力特别强, 无论是其运算的速度, 还是其剩余内存的大小, 都比其他的服务代理的能力高出很

多,这样,无形之中,就使得招标服务代理都倾向于把任务分配给它进行协作求解,这样做也是自然的,合理的。

需要说明的是,本文提出的基于等级域和时间开销的合同网模型和传统的合同网相比,具有初始化效应,这个过程产生的通信量比传统合同网的通信量要大,因此本文所提出的模型在 MAS 规模比较大的时候,更具有应用价值。

#### 4 结论

为了解决在传统合同网模型中多服务代理求解时存在的问题,本文提出了等级域的概念,并将其与多代理协作求解时的时间消耗一起引入到传统合同网模型中,建立了等级域-时间开销合同网模型。理论分析和仿真试验表明,该模型在多服务代理的个数和协作求解的次数较多时,其通讯量明显小于传统合同网,而且该模型中服务代理的负载分布比传统合同网模型的负载分布更均匀。

但该模型对于解决具有优先级和截止期限等特殊要求的硬实时任务的协作求解以及如何在应用中根据计算设备的运行状态来确定时间开销等问题,还需要进行更深入的研究。

#### 参考文献:

- [1] 范玉顺,曹军威.多代理系统理论、方法与应用[M].北京:清华大学出版社,2002:117-136
- [2] 哈肯.协同学[M].北京:原子能出版社,1984:1-21.
- [3] Jane W S L. ReaFit in e Systems[M]. Upper Saddle River PrenticeHall Inc., 2000:26-33
- [4] Lane D M, Mcfadzean A G. Distributed problem solving and reAction mechanisms in robot architectures[J]. Engineering Application Intelligence, 1994, 7(2): 105-117.
- [5] Uwe H, Lothar M, Martin S N, Thomas S. Pervasive Computing (Second Edition) [M]. Berlin: Springer-Verlag, 2003: 1-18.
- [6] Randall D, Reid G S. Negotiation as a metaphor for distributed problem-solving[A]. Proc. Communication in Multiagent Systems agent Communication Languages and Conversation Policies[C]. Melbourne, Australia: Elsevier Science press, 2003, 85(2): 51-97.
- [7] Sandholm T. An implementation of the contract net protocol based on marginal cost calculations[A]. Proc. Proceedings of the 12th International Workshop on Distributed Artificial Intelligence (AAAI-93) [C]. Hilden Valley, Pennsylvania USA: IEEE Computer Society, 1993: 256-262.
- [8] Jennings N R, Sycara K, Wooldridge M. A roadmap of agent research and development[J]. Journal of Autonomous Agents and Multi-agent Systems, 1998, 1(1): 7-38.
- [9] Franklin S, Graesser A. Is it an agent or just a program? a taxonomy for autonomous agent[A]. Proc. Proceedings of the Third International Workshops on Agent Theories Architectures and Language[C]. Budapest, Hungary: Springer-Verlag, 1996: 21-36.
- [10] Wooldridge M. Agent-based software engineering [J]. IEEE Proc Software Engineering, 1997, 144(1): 26-37.
- [11] Shoham Y. Agent-oriented programming [J]. Artificial Intelligence, 1993, 60(1): 51-92.
- [12] 兰少华,吴惠中,顾一禾. Agent协作中的一种选择算的实现[J]. 计算机工程, 2001, 27(12): 23-25.

#### 作者简介:



张文波 男, 1973 年生于黑龙江伊春, 东北大学信息科学工程学院博士生, 主要研究方向为复杂网络、嵌入式系统。

E-mail: zwl@neuea.com



赵海 男, 1959 年生于辽宁沈阳, 东北大学信息科学工程学院教授, 博士生导师, 主要研究方向为复杂网络、数据融合。

苏威积 男, 1975 年出生, 辽宁抚顺人, 东北大学信息科学与工程学院博士生, 主要研究方向为复杂网络。

除野 男, 1976 年出生, 辽宁沈阳人, 东北大学信息科学与工程学院博士生, 主要研究方向为复杂系统、人工智能。

王金东 男, 1976 年出生, 博士, 山东潍坊人, 主要研究领域为普适计算、嵌入式系统。

韩光洁 男, 1972 年出生, 博士, 黑龙江伊春人, 主要研究领域为普适计算、嵌入式系统。