

基于复合元胞自动机的分布式计算模型^①

朱晓敏^{②*} 耿建东^③ 陈东华^{*} 张润彤^{③***}

(^{*}北京交通大学机械与电子控制工程学院 北京 100044)

(^②科学技术部高技术研究中心 北京 100044)

(^③北京交通大学信息系统研究所 北京 100044)

摘要 针对目前的分布式计算网络仍然缺乏较高的自治管理和自主认知能力,只能提供独立状态的服务,经常出现多点故障而造成分布式网络运营成本增加、延时加剧甚至网络崩溃等问题,将元胞自动机的相关概念及模型引入到分布式网络计算中,通过修正元胞自动机的原有机制,提出了复合元胞自动机模型,并建立了基于复合元胞自动机的分布式计算模型来提高分布式网络中认知自主管理能力,从而降低分布式计算成本和提高工作效率。通过模拟程序和 GridSim 软件包验证和展示了复合元胞自动机应用于分布式计算领域的可行性及优越性。

关键词 元胞自动机,分布式计算,复合元胞,绿色网络,自主认知

0 引言

元胞自动机 (cellular automaton) 的设计思想源于生物学中自我繁殖现象,在计算机科学领域已得到广泛应用。元胞自动机模型对研究动态、并行、交互的系统十分重要^[1]。近年来随着网格计算、云计算新概念的提出,分布式计算显示出其强大的生命力。然而,目前的分布式网络一般只能提供短暂的并且有状态的服务,尽管也有一些自我管理的能力,但是仍然是缺乏较高的整体自治能力,难以有效地管理整个网络的资源,形成实质上的绿色高效的分布式网络。在某些网络中,自治主体的行为依赖于专家知识,只能对系统进行定性的判断与调整。当系统的状态发生定量改变时,无法对系统的性能参数做出连续的调整,以便使得系统性能保持在期望的范围内。特别是针对网格计算,虽然语义网格推动网格紧紧趋近它提供给用户的目标,采用统一平台来解决特定域的问题,但自治机制会更好地辅助其发挥作用^[2]。目前尚未建立完善的自治计算理论体系和研究方法体系,然而分布式系统更需要自治计算来提高大型网络环境管理效率。它本质上是复杂和动态的,所以在原有的基础上很难形成绿

色网络,而元胞自动机所依附的元胞空间类似于网络结构,同时元胞自动机独特的演化法则不同于传统的数学描述方程,其具有类似生物的自配置、自愈合、自优化、自保护、自学习的能力,正因为这样,元胞自动机为研究更为高度自治的分布式网络提供了可能。为了改善网络分布式运作环境,本文尝试将元胞自动机机制融入到分布式网络中,以实现更为高效和智能的分布式计算网络。本研究根据分布式计算的特点重新修正原始的元胞自动机的定义,提出了复合元胞自动机的概念,构建了基于复合元胞自动机的分布式计算模型,并且通过模型的模拟实验,验证了复合元胞自动机应用于分布式计算领域的可行性及优越性。

1 理论概述

在某些领域里^[3],元胞理论优于传统计算机数学理论,它能够解决传统算法难以很好解决的问题,如分布式网络的调度优化和自治计算等。元胞自动机自身衍生的模型的种类非常多,且其构建规则不固定。尽管有时规则非常简单,但是经过一定的演

① 863 计划(2009AA01Z211)资助项目。

② 女,1964 年生,博士,副教授;研究方向:复杂系统建模,人工智能以及优化控制等;E-mail: xzmzhu@bjtu.edu.cn

③ 通讯作者, E-mail: rtxhang@bjtu.edu.cn

(收稿日期:2011-09-16)

化后可以产生非常复杂的动力学行为,这些行为目前大致可以基于维数分类和基于动力学行为分类^[2]。尽管元胞自动机有着较为宽松甚至近乎模糊的构成条件,但是,作为一个数理模型,元胞自动机却有着严格的科学定义。下面给出基于集合论的定义^[2]:

设 d 代表空间维数, k 代表元胞状态并在一个有限集合 S 中取值, r 代表元胞的邻居半径, Z 为整数集, t 代表时间;假定 $d = 1$, 即元胞空间为一维空间,将整数集 Z 上的状态集 S 的分布记为 S_t , 元胞自动机的动态演化就是状态组合按照总体演化规则 F 随时间变化,可以记为

$$F: S_t \rightarrow S_{t+1} \quad (1)$$

这个动态演化又由各个元胞的局部演化规则 f 所决定。这个局部函数 f 通常被称为局部规则。对于一维元胞空间,元胞及其邻居可以记为 S_{2r+1} , 局部函数则可以记为

$$F(S_{2r+1}^t) = f(S_{-r}^t, \dots, S_0^t, \dots, S_r^t) \quad (2)$$

这样,根据式(1)和(2),就可得到一个元胞自动机的模型。同时可很容易地将 S_t 记为 S_t^d , S_{2r+1} 记为 $S_{(2r+1)}^d$, 将定义中的元胞自动机扩展到任意维空间。

Beigy^[4]等研究过具有多样学习能力的元胞自动机(cellular learning automaton, CLA)并应用于元胞网络中,证明了基于 CLA 的解决方案能产生更好的结果。Guan^[5]等提出了一种新的元胞自动机模型——基于伪随机数的自编程元胞自动机(self-programming cellular automata, SPCA),此模型能根据状态自动改变自己的传播规则,同时提高了网络工作效率。由上可见,元胞自动机在分布式计算理论上已有一定的发展。而网络的自治计算^[6,7]主要针对目前计算机网络技术快速发展而导致日益突出的分布式系统管理复杂的问题。当前,美国 Rutgers 大学 Automate 的全面目标是研究具有自治能力的、上下文意识的网格应用所需要的技术^[8]。陶丽^[9]等提出了面向自治的基于面向自治计算的 Agent 系统动态重构模型,李亚琼^[10]等提出了一种面向虚拟化云平台的全局内存优化框架,该框架能够很好地优化云平台中内存资源配置,提升整个平台的资源利用率。而张海辉^[11]等提出了 GISA2.0,实现了可扩展的网格信息模型和面向服务、支持多种监控信息聚集的层次化信息管理框架^[12]。这一些成果都为元胞应用于类似于网格的分布式网络的研究奠定了基础。

网格不仅是分布式计算概念拓展的一个领域,

也是和元胞空间较为相似的分布式网络。目前人们普遍对网格的抽象定义为“协作资源共享,在动态的多机构的虚拟结构中解决问题”^[13]。尽管分布式系统日益普及而且越来越重要,但开发人员仍然需要面对一些挑战,包括固有的复杂度、附加的复杂度以及方法和技术上的不足,对核心概念和技术持续的重新创建和重新发现。分布式系统的部署涉及到不同种类、地域上分散的、能够动态获取资源的有效管理,然而一个分布式环境的高效性在很大程度上取决于它的有效调度。因此,构建安全和高效的动态协作网络调用机制对充分利用分布式资源,减少能耗至关重要。

2 复合元胞自动机模型

本文将元胞的概念融入到分布式计算中,使元胞作为分布式系统资源的抽象单元。在构建基于元胞的分布式计算模型之前,元胞作为分布式计算的基本单位,其概念需要拓展,拓展后的元胞模型为复合元胞自动机(compound cellular automaton, CCA)的模型,以满足实际分布式系统的需要。

2.1 复合元胞模型的构建

对于实际复杂的分布式网络环境,原始的元胞模型显然不能够满足其要求,所以需要式(1)和(2)定义的元胞概念进行相应的概念拓展,构建称之为复合元胞自动机的新型元胞演化单元。

在以往的元胞自动机的应用中,元胞演化运算基本是以单个元胞作为单元,并且仅仅包含简单的状态集,而对于单元胞 C_i ,要实现自身演化和并行计算很复杂,同时也忽略了不同元胞之间通信的成本,造成实际应用的偏差。对于本文,若将元胞视作复杂网络的节点,元胞已经不再是内存的某一个孤立和简单状态集合。本文将复杂网络上的计算设备节点看作是一个复合元胞自动机,但复合元胞自动机并不是仅仅单个或多个内部元胞的相加集合,而是具有自我演化,自我管理特征的有机整体。设某复合元胞 O 在结构上是单元胞 C_i 的集合体,效仿单元胞 C_i 的状态定义,复合元胞 O 也有自己的状态,假设也为 $S_O = \{x_i\} (i = 1, 2, 3, \dots, N_O)$, 当 $S_O = \emptyset$ 时,复合元胞 O “死亡”。复合元胞不仅能表示大量的状态,还能自身执行大量的演化逻辑。

在基于复合元胞的分布式网络中,复合元胞之间能进行高级的演化,在每个复合元胞内部,子元胞在动态地演化而不受外界复合元胞的影响,最终实

现整个网络更为深层次的演化。复合元胞 O 模型中的单体多状态表示单元胞 C 的状态集合 $S_C = \{x_i\} (i = 1, 2, \dots, N_C)$, 即表示独立线程计算的特征集合。多体多状态表示的是不同的线程的不同状态共同组成的以单元胞状态集合为元素的复合元胞的状态总集合。可以把单元胞自动机概括为一个用数学形式来表示的四元组, 即

$$A_C = (L_{C\alpha}, S_C, N_C, f_C) \quad (3)$$

则复合元胞自动机的数学形式的四元组也可得到:

$$A_O = (L_{O\alpha}, S_O, N_O, f_O, M_O) \quad (4)$$

$$M_O = (\{A_{C\alpha}\}, f_{MO}) \quad (5)$$

式(3)、(4)和(5)中, A_C 、 A_O 分别代表单元胞自动机模型和复合元胞自动机模型; $L_{C\alpha}$ 、 $L_{O\alpha}$ 分别代表单元胞空间和复合元胞空间; $L_{C\alpha}$ 、 $L_{O\alpha}$ 分别为 A_C 、 A_O 的空间维数; S_C 、 S_O 分别是元胞 C 、 O 的有限离散状态的集合; N_C 、 N_O 分别表示 C 、 O 邻域内所有元胞的组合(包括中心元胞在内), f_C 、 f_O 为 A_C 、 A_O 局部状态装换函数, 也就是演化规则, 抽象形式如式(1)和(2)。 M_O 为复合元胞 O 的内部结构, $\{A_{C\alpha}\}$ 表示复合元胞 O 中的单元胞 C 的集合, f_{MO} 为复合元胞对 $\{A_{C\alpha}\}$ 管理能力。以上建立了复合元胞的基本结构, 然而, 复合元胞并非简单的元胞集合, 其能力在于如何定义 f_{MO} , 如何使之成为智能的元胞整体。

2.2 复合元胞通信分析

本节探讨通信机制在分布式网络中的演化。元胞概念结构与网络服务结构有相似性, 用于在不同种类的计算环境中建立分布式应用。参考 OGSA 网络服务结构, 我们融合了元胞的结构特点, 构建我们的元胞分布式计算通信机制。

设一个基于复合元胞的元胞自动机 A_O 的元胞空间为 V^d 。设元胞自动机的维度 d 为 2, 即为二维元胞空间 V^2 。二维元胞空间 V^2 中遍布了有限个复合元胞 $O_i (i = 1, 2, 3, \dots, N_O)$, N_O 最大为二维元胞自动机中复合元胞总数。根据复合元胞的定义, 设 M 为单个复合元胞所包含的单元胞数目, N 为单个元胞所拥有的邻居数目。不同的复合元胞 O 之间的通信通道 L 抽象为 $L = \{l_1, l_2, l_3, \dots, l_N\}$ 。设有复合元胞 $O_i = \{C_{\alpha 1}, C_{\alpha 2}, C_{\alpha 3}, \dots, C_{\alpha M}\}$ 和 $O_j = \{C_{\beta 1}, C_{\beta 2}, C_{\beta 3}, \dots, C_{\beta M}\}$, 则定义

$$L_{ij} = L(O_i, O_j) = \{l_1, l_2, l_3, \dots, l_K\} \quad (0 \leq K \leq M) \quad (6)$$

$$l_k = l(C_{\alpha a}, C_{\beta b}) \quad (1 \leq a, b \leq M; 1 \leq i, j \leq N_O; 1 \leq k \leq K; i \neq j) \quad (7)$$

式中, L_{ij} 表示 O_i 和 O_j 之间的通信通道, 视为一个复合元胞的邻居连接通道。 $\{l_1, l_2, l_3, \dots, l_K\}$ 为连接 L_{ij} 的子通道集合, 子通道 l_k 表示 O_i 和 O_j 里各自包含的活跃元胞 $C_{\alpha a}$ 和 $C_{\beta b}$ 之间的数据联系。 $C_{\alpha a}$ 和 $C_{\beta b}$ 的对应关系由复合元胞的规则动态确定。 K 表示与两活跃复合元胞 O_i 和 O_j 相关的活跃单元胞连接总数目, 即与 L_{ij} 的通道相关的复合元胞中的活跃单元胞数目。注意由同一通道 L_{ij} 两端的两个元胞 O_i 和 O_j 各自不一定包含相同数目的单元胞数目, 因为对于一个复合元胞来说, 并不是只有一个与其他复合元胞连接的通信通道 L_{ij} , 相反, 与不同的复合元胞相连接的通道都会产生一个 L_{ij} 连接。总之, 就是复合元胞内的所有单元胞通信都是通过复合元胞的通信连接实现的。综合式(6)和(7), 构建了基于元胞的分布式网络, 其网络结点的元胞的结构见图 1。

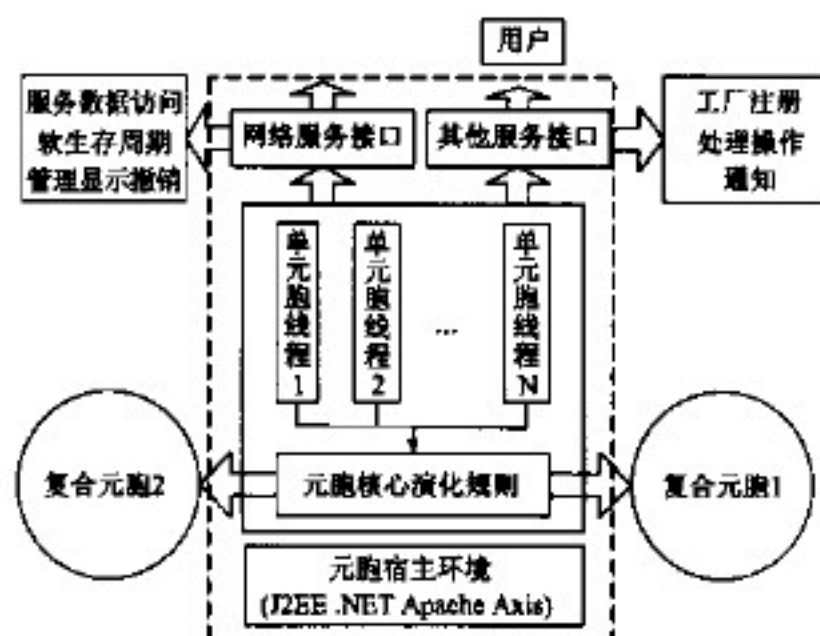


图1 分布式网络中复合元胞的构造

分布式网络的计算机被抽象成复合元胞模型, 利用复合元胞模型的结构和演化机制相融合才能发挥元胞在分布式网络的优势。

3 基于元胞的网络模型

3.1 元胞的调度机制

分布式计算网络(如网格)的调度可以定义为作业映射到多管理域的资源上的过程。Nichols^[15]等提到过的模型和架构可用于网格中自主工作流的调用, 通过优化网格, 即根据整个执行时间或者吞吐量和资源利用成本并考虑满足使用者和应用的需求, 选择最优的资源配置和调度方法。设 Q' 、 R' 分别为原始问题和汇总的总结果, Q_i 和 R_i 分别是参与分布式计算的智能设备 C_i 所下载的计算单元和通

过计算所获得运算结果, $\oplus$ 为抽象合并运算符号, 则其计算模型抽象为

$$Q' = Q_1 \oplus Q_2, \dots, \oplus Q_{N-1} \oplus Q_N \quad (8)$$

$$R' = R_1 \oplus R_2 \oplus R_3, \dots, \oplus R_{N-1} \oplus R_N \quad (9)$$

($N = 1, 2, 3, \dots, N_L$)

这类似于集中式调度。显然, 式(8)(9)代表的分解策略虽较简单, 但在利用分布式网络时存在问题, 比如过分依赖于中心服务器的协调、多任务并发时控制复杂及资源有效性不确定等^[15]。

设 Q 为需要通过分布式计算方法来解决的较大规模的问题, 则可用于分布式计算的问题 Q 可以分裂为具有相互联系的子问题 $Q_i (i = 1, 2, \dots, N)$, 一个 Q_i 中包含一个或多个单位问题 q_{ij} , 单位问题 q_{ij} 是问题 Q 的基本求解单元, 即实现元胞所能承受的运算任务。同时我们定义抽象合并运算符号 $\oplus$, 这代表在子问题之间可以规定的任何的运算法则, 以便分解问题或者组合答案。 $f(q_{ij})$ 是整个元胞的演化法则, $f_i(q_{ij})$ 为元胞的子演化法则。设 R 为问题 Q 的解, 因此可以得到问题 Q 的以下分解模型:

$$Q_i = \begin{cases} q_{i0} \oplus Q_i & (Q_i \neq null) \\ q_{i0} & (Q_i = null) \end{cases} \quad (10)$$

$$Q_i = Q_{i,1} \oplus Q_{i,2} \oplus \dots \oplus Q_{i,M-1} \oplus Q_{i,M} \quad (11)$$

($i = 0, 1, 2, \dots, N-1, N; N = 2^M; j = 1, 2, \dots, M-1, M$)

由于本文不着重探讨任务的分解策略, 于是设问题 Q 的运算结果 R 组合模型为

$$R_i = \begin{cases} r_{i0} \oplus R_i & (R_i \neq null) \\ r_{i0} & (R_i = null) \end{cases} \quad (12)$$

$$R_i = R_{i,1} \oplus R_{i,2} \oplus \dots \oplus R_{i,M-1} \oplus R_{i,M} \quad (13)$$

$$r_{i,j} = f(q_{ij}) = f_1(q_{ij}) \oplus \dots \oplus f_m(q_{ij}) \quad (14)$$

($i = 0, 1, 2, 3, \dots, N-1, N-1 = 2^M; j = 1, 2, 3, \dots, M-1, M$)

综上式(8)~(14), 简单构建了问题的分解模型。通过这个模型, 可以了解分布式计算的计算流程, 如图 2 所示。

3.2 网络模型的建立

在确定基于元胞的网络模型过程中, 采用元胞基本概念来探讨本文中基于元胞自动机的网络。

(1) 确定元胞的状态

网络中的节点可抽象为复合元胞, 每一个网络节点尽管在设备能力和具体网络服务都具有差异, 但都包含了相同的一套元胞规则。设元胞 C 的状态集合为 $\{D, F, B\}$, 是离散集合。 D 表示元胞 C “死亡”, 即元胞 C 脱离元胞空间; F 表示元胞 C 处

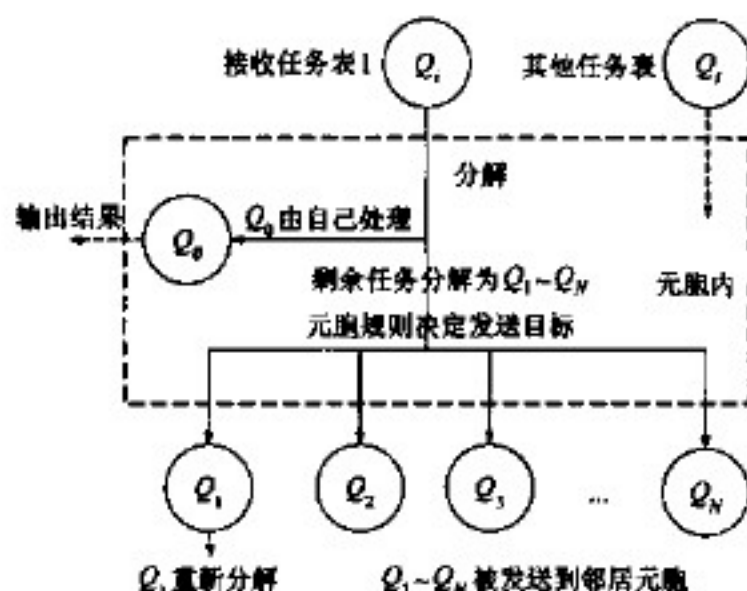


图 2 元胞任务分解流程

于空闲等待(free)中, 可以分配计算任务; B 表示元胞 C 处于单位计算的工作(busy)中, 已被锁定了, 不能再分配任务。 B 和 F 可以直接相互转化, D 和 F 可以直接相互转化, B 和 F 之间状态不能直接进行转化, 必须通在 D 和 B 中同 F 状态间接转换。 R 为元胞的值。所以我们规定在本模型中, 以复合元胞为元胞空间的基本元胞形式, 复合元胞模型如式(6)所抽象的形式。

(2) 构造元胞空间

对于应用于分布式网络计算的元胞空间, 采用复杂网络代替, 即将元胞视为复杂网络的节点。而且采用二维元胞空间, 即设元胞空间 V^2 中包含 N 个复合元胞, 即 N 个相互联系的计算机服务系统。构造的元胞空间 V^2 实质上是一个复杂的计算机网络。网络中每一个计算元胞的基本构造和演化规则都是相同的。

(3) 确定元胞邻居

设某一个复合元胞 O_i 所指向的元胞邻居集合为 $\{O_j | j \neq i, 0 \leq j \leq M\}$, 可以根据不同的实际情况决定元胞 O_i 邻居集合的个数 M , 以便实现更为灵活的演化规则。

(4) 确定演化规则

演化规则是一个局部状态转移函数。如元胞 C_i 的局部演化规则^[2]表示为

$$f: S_i^{t+1} = f(s_i^t, S_N^t) \quad (15)$$

式中 f 表示状态转移函数, S_N^t 表示 t 时刻元胞 C_i 的邻居元胞的状态。本研究引进状态检测。复合元胞的状态集包含失效状态 D , 这符合元胞空间应用于计算机分布式网络可能出现服务资源失效的问题, 造成耗费资源等待无效主机延迟返回空值。元胞自动机的演化规则可避免这样的问题, 很容易地进行

动态重连邻居。设元胞 C 中检测邻居状态方法 $CheckS(C)$ 。通过确定这样的演化规则,使得元胞的活跃能避开“死掉的元胞”,自动重连到距离较近的某个活跃的元胞上,增强分布式计算的稳定性。

为了很容易地保证重连元胞时选择元胞邻居的最优性,为元胞填入以下规则:设某元胞 C 接收到子任务 Q , Q 可分为 $q_0, Q_1, Q_2, Q_3, \dots, Q_N, q_0$ 由元胞 C 处理,子任务组 Q_i 将会被分别分配到元胞 C 的邻居中去,则此元胞在求解任务时会把连接邻居元胞所花费的时间 t_i 记录下来。设 C 将 Q_i 发送到 C_i 的途中所记录的花费的时间为 t_i , 则元胞 C 下一次接收到任务时,它会将任务优先依次发给 t_i 较小的空闲邻居中,实现动态演化。多次演化后能反映出整体网络的利用情况,更容易让元胞在未来演化中吸取经验。抽象的规则如下:

$$O_{k,i,j} = \{x \mid CheckS(x) \neq D, B; x \in O\} \quad (16)$$

式(16)中,元胞集合 $O_{k,i,j}$ 是复合元胞 $O_{k,i}$ 连接其他有效元胞的集合。其中的连接元胞可以使任何处于网络的有效元胞,包括之前已经用过的元胞,因为在复合元胞内,能同时进行相关元胞的演化。

(5) 计算元胞求解时间

设某复合元胞 $O_{k,i}$ 的处理任务总时间为 $T_{k,i}$, 为第 k 层复合元胞的第 i 个连接复合元胞所花费的时间,则当前元胞求解所获得任务花费总时间为

$$T_{k,i} = \max \{ \max_{1 \leq j \leq n} \{ T_{k,i,j} \}, P_{k,i} \} + S_{k,i} \quad (0 \leq k \leq N_k, 0 \leq i \leq N_i) \quad (17)$$

式中 $T_{k,i,j}$ ($0 \leq j \leq n$) 为复合元胞 $O_{k,i}$ 所连接一组分配该任务的元胞 $O_{k,i,j}$ 的处理时间, $P_{k,i}$ 为当前分配任务处理时间, $S_{k,i}$ 为当前元胞接收到元胞解析和分解的时间。则由此可以推断, $T_{0,0}$ 为远程任务的发起端,则分布网络中的处理时间为 $T_{0,0}$ 。

4 仿真分析

为了验证基于复合元胞自动机的分布式计算模型的可行性和优越性,通过设计一个任务扩散程序,在单台计算机模拟本文所描述的复合元胞演化时在分布式计算工作的特点。

本研究使用 JAVA 编写其模拟程序,它通过一个二维复合元胞数组 $Grid$ 来作为二维元胞空间,其维度为 $M \times N$, 包含 M 个单元胞, N 个邻居。单元胞在模拟程序里面是一个独立的计算机线程。复合元胞 $CCell$ 具有独立管理和分配它内部的 $Cell$ 集合的能力。模拟程序的工作原理如下:通过激发主程

序元胞空间的某个复合元胞 O , 将 S 个任务发送给 O , 则这个复合元胞 O 通过预先规定的演化规则自动地不断进行演化,其演化规则基本如前所述。最终,模拟程序会输出元胞空间内的元胞演化状况。

简单地,设置模拟程序参数 $MN = 10, NN = 10, M = 8, N = 8$, 将单个任务 $S_1 = 150$ 个任务发送给元胞空间内的元胞 $O(5,5)$ (下标从零开始), 则通过模拟程序最终可以得到如图 3 所示的复合元胞 O 在元胞空间演化分布示意图。

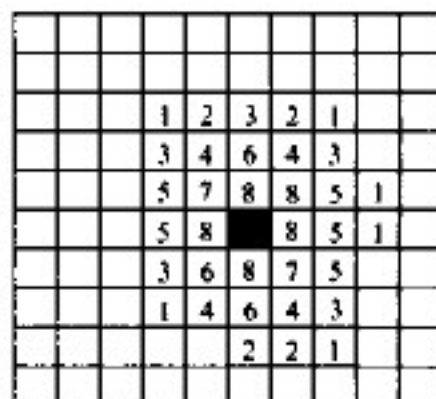


图3 单任务单复合元胞演化分布图

如图 3, 每格子代表一个复合元胞 O 。每个元胞的数字是模拟程序统计出来复合元胞内已所使用的单元胞数, 最多为 $M = 8$ 。图 3 中的深灰色元胞为开始复合元胞, 通过它按预定的规则不断演化, 使原始元胞 $O(5,5)$ 的负荷为满负荷, 即该复合元胞内的 8 个独立的单元胞同时在工作, 同时复合元胞的负荷以原始复合元胞为中心, 依次往外递减。在此种情况下, 元胞的演化方式类似于梯度函数的均匀递减方式, 但梯度的扩散方式在瓶颈网络中可能会失效。假设此时元胞空间内的两个元胞 $O(2,2)$ 、 $O(8,2)$ 和 $O(2,8)$ 同时被激发, 接收到的任务数分别为 $S_2 = 40, S_3 = 80$ 和 $S_4 = 80$, 则通过模拟程序输出其空间数据, 如图 4 所示的四任务复合元胞演化分布图。深灰色的元胞为原始激发的四个元胞, 中等灰色的是两个任务所交错的元胞区域, 浅灰色的元胞为分别受四个元胞独立激活的元胞。



图4 四任务同时工作时演化分布图

由图 4 发现,基于复合元胞自动机的分布式计算模型演化机制使得任何一台参与分布式计算的计算机都成为计算的发起者。由于复合元胞并行的特点,多个分布式计算发起者的任务可共享同一台计算机的资源,如图 4 中的 4 个原始元胞激发的中等灰色区域是 4 个分布式计算共享的区域。同时,由于元胞演化的灵活性,即使元胞到达了网络的边缘,它仍然能返回有效寻找附近空闲的元胞进行演化,使分布式计算得以进行下去。实际上,基于该模型的分布式网络不需要一个中心服务器进行协调参与分布式计算的计算机群,只需在任何一台计算机上启动分布式计算,则只要该分布计算网络中存在可用的计算机资源,该分布式计算就会自动持续,不会中断,即使其中某些计算机因故障停用,复合元胞仍然通过演化规则重连附近的复合元胞,使计算持续下去。由此,可以展示在一些具有瓶颈的复杂网络中,元胞能较灵活地利用网络完成预定的任务。

另外也可通过采用 GridSim^[16] 网络模拟软件包进行更专业的试验。前面实验仅仅用来阐述元胞运行的工作原理和证明实现基于元胞的网络资源调度的可行性和灵活性。GridSim 模拟器里面所包含的网络资源的仿真情况与实际较为相符,通过设置 GridSim 模拟器相关的参数来控制网络的规模和网络任务的分配机制,也即是元胞的扩散机制。GridSim 模拟器设置的主要参数如表 1 所示。

表 1 GridSim 模拟器主要参数设置	
主要参数	参数取值
资源数	20
用户数	10
单用户任务数	1...10 随机产生
单资源机器数	1...5 随机产生
单机器处理器数	1...4 随机产生

表 1 中有些参数是在一定范围内随机生成的,目的是为了模拟分布式网络中的不确定性。除了表 1 参数外,还有一些重要参数(如使用机器的成本、每块 PE 使用的时间、波特率等)需要设置。除了设置模拟器的相关参数外,最重要的是确定任务的分配机制,这也是基于元胞自动机的网络的最重要一步。对于基于元胞的分布式网络来说,规则越简单越好。设置好参数后,随机运行模拟器 5 次,产生了 5 组运行时的数据。经过整理后,给出了用户任务数与总时间和总成本的关系(图 5)。图 5 显示,随着用户任务数的递增,总时间和总成本基本在

快速地上升,符合预期结果。

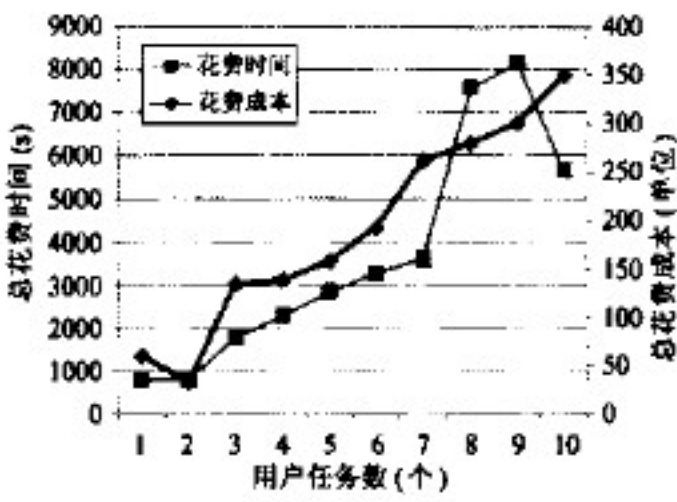


图 5 用户任务数与总时间、成本关系

我们可为 GridSim 模拟器编写任务分配规则来决定网络任务分配机制,也就是说,通过 GridSim 模拟器,可以采用元胞的机制进行任务调度。此实验通过示例性的演示说明了基于元胞自动机网络的可行性,为未来进行这方面的工作提供了必要的技术支持。

5 结 论

本文研究了元胞自动机的机制,提出了一种复合元胞的模型,并据此模型修正了原始的元胞自动机的定义,使其符合分布式网络计算的特点。基于复合元胞自动机的分布式计算模型实质上利用了元胞自动机的类似生物的演化机制和智能认知特性,它已用于分布式计算网络中。在此基础上研究如何构建绿色高效的分布式网络,增强网络的自治能力,提高分布式网络计算的效率和减少其时间成本等,可避免一些当今分布式计算中存在的问题。通过编写一个扩散模拟程序和利用 GridSim 仿真一个基于元胞的网络来对基于此模型的网络计算进行模拟运行和数据分析,可验证此模型下的网络计算的可行性。由于缺乏必要的硬件条件,本文所阐述的分布式计算并没有在真实的分布式网络中进行测试,下一步工作即是修正模拟程序,然后模拟更多的可能出现的问题,从而丰富原模型,以实现更高层次的绿色网络。

参考文献

[1] Zhu H, Wu Y H, Huang S. Cellular automata with object-oriented features for parallel molecular network modeling. *Nano Bioscience, IEEE Trans*, 2005, 4(2): 141-148

- [2] Li M Z, Baker M. The Grid: Core Technologies. USA: John Wiley and Sons, 2005. 4-10
- [3] 贾斌,高自友,李克平. 基于元胞自动机的交通系统建模与仿真. 北京:科学出版社, 2007. 1-45
- [4] Beigy H, Meybodi M R. Cellular learning automata with multiple learning automata in each cell and its applications. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans*, 2010, 40(1): 54-56
- [5] Guan S U, Tan S K. Pseudorandom number generation with self-programmable cellular automata. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Trans*, 2004, 23(7): 1095-1101
- [6] Dobson S, Sterritt R, Nixon P. Fulfilling the vision of autonomic computing. *Computer*, 2010, 43(1): 35-41
- [7] Fernandex-Carrasco L M, Terashima-Marin H, Valenzuela-Rendon M. On the path towards autonomic computing: combining swarm intelligence and excitable media models. In: *Proceedings of the 7th Mexican International Conference on Artificial Intelligence, Atizapán de Zaragoza, Mexico*, 2008. 192-198
- [8] Liu H, Parashar M, Hariri S. A component-based programming framework for autonomic application In: *Proceedings of the 1st International Conference on Autonomic Computing*, New York, USA, 2004. 10-17
- [9] 陶丽,张自力. 基于面向自治计算的 Agent 系统动态重构模型. *计算机科学*, 2007, 34(5): 147-158
- [10] 李亚琼,宋莹,黄永兵. 一种面向虚拟化云计算平台的内存优化技术. *计算机学报*, 2011, (4): 684-693
- [11] 张海辉,周兴社,杨志义. 面向多自治域网格的信息服务模型及其实现. *计算机学报*, 2008, 31(1): 65-72
- [12] Wei J, Hung T, Turner S J, et al. Architecture model for information service in large scale grid environments. In: *Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid*, Singapore, 2006. 107-114
- [13] Foster I, Kessel M C, Tuecke S. The anatomy of the grid: enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 2001, 15(3): 6-7
- [14] Nichols J, Demirkan H, Goul M. Autonomic workflow execution in the grid. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Trans*, 2006, 36(3): 353-364
- [15] Hamscher V, Schwiegelshohn U, Streit A, et al. Evaluation of job-scheduling strategies for grid computing In: *Proceedings of 1st IEEE/ACM International Workshop on Grid Computing*, Bangalore, India, 2000. 191-202
- [16] Xu H B, Liu X R, Zhang H. A grid workflow process engine: architecture and simulation. In: *Proceedings of 2009 International Forum on Information Technology and Applications*, Chengdu, China, 2009. 245-248

A distributed computing model based on compound cellular automata

Zhu Xiaomin^{*}, Geng Jiandong^{**}, Chen Donghua^{*}, Zhang Runtong^{***}

(^{*} School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044)

(^{**} The High Technology Research and Development Center, The Ministry of Science and Technology, Beijing 100044)

(^{***} Institute of Information Systems, Beijing Jiaotong University, Beijing 100044)

Abstract

The concept of cellular automation and its model were introduced into distributed computation to solve current distributed computing networks' problems of high operating cost, serious time delay, and even debacle due to the multi-point failure caused by lack of high self-management and self-awareness and only providing independent services. By amending the original cellular automaton mechanism, a compound cellular automata model was proposed, and the distributed computing model based on compound cellular automata was established to enhance distributed networks' efficiency and reduce the operating cost. The comprehensive simulations and comparisons via GridSim verified the feasibility and efficiency of the proposed compound cellular automata model in the field of distributed computing.

Key words: cellular automation, distributed computing, compound cellular, green network, self-awareness