

# 异构多核协作系统的混沌离散粒子群 NoC 映射算法

王 雷 凌 翔 胡剑浩

(电子科技大学通信抗干扰技术国家级重点实验室 成都 611731)

**摘 要** 针对异构多核片上网络(NoC)的任务映射问题,根据 IP 核的选择以及 IP 核向 NoC 平台中位置映射的两个阶段的不同特点,分别提出能耗和延时的粗略估算模型和精确计算模型。为避免离散空间搜索解落入局部最优,设计了混沌扰动机制。提出了带混沌扰动机制的改进型离散粒子群优化算法,以搜索能耗和延时优化的多目标 NoC 映射方案,该算法比传统优化算法在能耗和延时有显著的性能提高。

**关键词** 片上网络,映射,多目标优化,离散粒子群优化

**中图分类号** TP302 **文献标识码** A

## NoC Mapping for Heterogeneous Multi-core Cooperative System Based on Chaotic Discrete Particle Swarm Optimization

WANG Lei LING Xiang HU Jian-hao

(National Key Laboratory of Science and Technology on Communications, UESTC, Chengdu 611731, China)

**Abstract** Heterogeneous multi-core cooperative networks-on-chip(NoC) mapping was split into two stages: assigning the tasks to the suitable IP cores, and then mapping the IP cores to the appropriate NoC tiles. To deal with the different characteristics of these two successive stages, a coarse model and an accurate model of energy consumption or delay estimation were proposed respectively. A discrete particle swarm optimization with chaotic disturbance was proposed to solve the multi-objective NoC mapping problems, where a chaotic disturbance mechanism was designed to avoid obtaining local optimal solutions. The simulation results are better than that obtained by traditional schemes significantly.

**Keywords** NoC, Mapping, Multi-objective optimization, Discrete particle swarm optimization

### 1 引言

伴随信息科技的飞速发展,通信与多媒体应用呈现出复杂化的趋势。在应对实时性与海量数据处理的需求时,大量处理器协同工作的系统设计模式表现出极大的优势。这给片上通信能力带来了极大考验,以总线为承载基础的芯片设计模式往往不能满足需求。片上网络(Network on Chip, NoC)正是应对超大规模集成电路设计中由尺寸和通信需求等因素带来的诸多问题的一种全新方案。

NoC 设计通常包含 4 个层面的内容<sup>[1]</sup>。映射属于其中应用的建模与优化的范畴,其核心思想是在设计指标的约束下确立起应用中各任务同片上网络通信节点之间最优的绑定关系。建立一种高效的算法,在较短时间内搜索出符合设计指标的映射方案,是研究的重点和难点。

目前的研究工作已经提出了一些以功耗或延时为优化目标的 NoC 映射算法。文献[2]研究了 NoC 系统的能耗并建立了数学模型。文献[3]从避免网络拥塞的角度研究映射。文献[4]在映射过程中考虑电压岛的特性达到对全局能耗的优化。文献[5]采用基于禁忌搜索的算法求解低能耗的映射

方案。文献[6]将能耗和延时进行线性统一,采用蚁群算法实现任务映射。文献[7]将路由算法与映射综合考虑,解决面向能耗的 NoC 映射问题。这些单目标优化算法求得的 NoC 映射结果有较好的性能,但不能满足多个设计指标的复杂应用。另外,多数文献都是讨论任务到处理单元单一映射的情形,少有文献关注异构处理单元的映射问题。

复杂应用中协同运行的任务在性质上通常存在差异性,因此任务往往交付给不同类型的异构 IP 核进行处理。异构多核协作系统的 NoC 映射问题事实上分为两个阶段。首先是确立任务与 IP 核的映射关系,而后是 IP 核向网络位置的映射。根据映射两个阶段的特点,本文分别建立起了适合该阶段的能耗和延时模型,设计了混沌扰动机制和 NoC-IP 映射参考基准,改进了离散粒子群优化算法,用于能耗和延时优化的多目标 NoC 映射方案的搜索。能耗和延时指标的变化方向通常并不一致,甚至冲突。搜寻能映射方案时,只能得到关于这两个指标折中的 Pareto 解。

### 2 NoC 映射问题描述

本文研究的 NoC 映射分为图 1 所示的两个阶段执行。

到稿日期:2010-10-18 返修日期:2011-02-26 本文受国家中长期重大专项项目(2011ZX03003-003-04),国家自然科学基金项目(60873076)资助。

王 雷(1983-)男,硕士,主要研究方向为通信集成电路、无线通信,E-mail:wanglei\_26@163.com;凌 翔(1972-)男,副教授,主要研究方向为无线通信、VLSI;胡剑浩(1971-)男,教授,博士生导师,主要研究方向为无线通信、VLSI。

第一阶段中,任务图中的任务节点被分派到恰当的 IP 核上。第二个阶段中,已经同任务相绑定的 IP 核被映射到 NoC 平台,明确其在片上网络中的具体位置。复杂应用中的多个任务时常被同时分派到一个 IP 核上,以达到资源共享的目的。

本文映射算法研究中采用的 NoC 平台如图 1(右)所示。该平台使用二维网格(2D-Mesh)拓扑进行节点间互联。其路由由节点(R)与处理模块唯一对应。每个路由节点与处理模块通过网络接口(Network Interface, NI)进行数据交换。由于映射算法的研究涉及数据包传递路径的长度,因此规定 NoC 平台使用静态最短路径路由算法,其寻址简单,且结论容易被推广到其他更为复杂的平台架构。

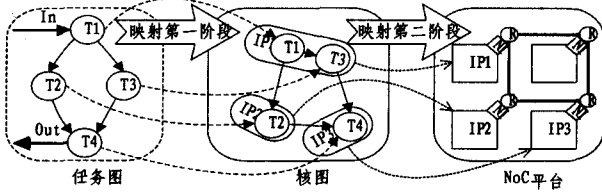


图 1 NoC 映射的两个阶段

### 3 面向能耗和延时的 NoC 映射数学模型

**定义 1** 待选 IP 核集  $IP = \{ip_1, ip_2, ip_3, \dots, ip_N\}$ , 包含 NoC 设计中可供选择的所有 IP 核。集合中的任意两元素为同质或异构资源。规定  $|IP|$  为集合中元素的个数。

**定义 2** 任务执行特征图(TPCG)  $\mathcal{G} = G(C, A)$  为有向无环赋权图。 $\mathcal{G}$  中的顶点集合  $C$  表示具体应用中的任务, 边集合  $A$  代表各任务之间的通信关系。用  $|C|$  表示所有任务节点的数量。

在 TPCG 中,  $\forall c_i \in C$ , 其属性由二元组  $\{T^i, E^i\}$  描述。 $T^i = \{t_1^i, t_2^i, \dots, t_j^i, \dots, t_{|C|}^i\}$ , 其中任意元素  $t_j^i$  表示 TPCG 中任务  $c_i$  由  $ip_j$  完成所需的时间。若  $ip_j$  不能执行任务  $c_i$ , 则记  $t_j^i = \infty$ 。 $E^i = \{e_1^i, e_2^i, \dots, e_j^i, \dots, e_{|C|}^i\}$ , 其中的任意元素  $e_j^i$  表示任务  $c_i$  由  $ip_j$  完成消耗的能量。若  $c_i$  不能由  $ip_j$  完成, 则记  $e_j^i = \infty$ 。

$a_{ij} \in A$  为顶点  $c_i$  到  $c_j$  的有向弧。 $\forall a_{ij} \in A$ , 其属性由这条边上的权值  $\omega_{ij}$  予以描述。 $\omega_{ij}$  表示协同工作的任务节点  $c_i$  向节点  $c_j$  发送的通信数据量大小。

**定义 3** 通信架构特征图(CACG)  $\mathcal{N} = N(R, P)$  为有向图, 描述 NoC 平台通信网络。其中  $r_i \in R$  表示 NoC 平台上的路由节点。 $p_{ij} \in P$  表示从  $r_i$  到  $r_j$  的路径。

映射方案  $M: C \rightarrow R$  完成所有任务  $c_i \in C$  向路由节点  $r_j \in R$  的映射。 $M \in MAP$ ,  $MAP$  为所有映射方案的集合。

$M_1: C \rightarrow IP$  完成  $c_i \in C$  向  $ip_j \in IP$  的映射。 $M_2: IP \rightarrow R$  完成  $ip_i \in IP$  向  $r_j \in R$  的映射。 $M_1, M_2$  分别表示  $M$  的第一阶段和第二阶段。

基于以上定义, 面向能耗和延时的映射问题的数学描述如下:

给定任务执行特征图  $\mathcal{G}$ 、IP 核集合  $IP$  以及通信架构特征图  $\mathcal{N}$ , 寻找映射方案  $M \in MAP$ , 使得映射后该应用在 NoC 平台上运行消耗的总能耗  $Energy$  以及总耗时  $Delay$  均尽可能小, 即:

$$V \sim \min_{M \in MAP} \{Energy(M), Delay(M)\}$$

约束条件为:

$$\forall c_i \in C, M_1(c_i) \in IP$$

$$\forall ip_n \in IP, M_2(ip_n) \in R$$

$$\forall ip_m, ip_n \in IP \text{ 且 } m \neq n, M_2(ip_m) \neq M_2(ip_n)$$

第一个约束条件保证 TPCG 中的每个任务节点均能被分配到一个 IP 核上, 而多个任务可以共同选择同一 IP 核。后两个约束条件共同确保每个 IP 核都能被唯一映射到 NoC 平台中的不同路由节点。

从数学描述可以看出, 寻找能耗和延时均优的异构多核 NoC 映射方案, 是具有两个优化目标的多目标规划问题。

#### 3.1 NoC 能耗模型

整个 NoC 系统运行中消耗的能量  $Energy$  由 IP 核在任务处理中消耗的能量  $E^{core}$  和数据传输中网络上消耗的能量  $E^{net}$  两部分构成, 即:

$$Energy = E^{core} + E^{net}$$

对于相同的数据处理量, 核上能耗只与核的类型相关(核电压、频率等条件确定)。 $\Delta$  比特数据在特定核上的处理能耗  $E_{bit}^c$  表示为:

$$E_{bit}^c = \rho_{ce} \cdot \Delta$$

式中,  $\rho_{ce}$  是与核类型相关的常数。

网络上消耗的能量可以表示为片上路由器的能耗和链路能耗两部分。单比特数据传输中的能量消耗表示为<sup>[7]</sup>:

$$E_{bit} = E_{S_{bit}} + E_{L_{bit}}$$

式中,  $E_{S_{bit}}$  是片上路由器内部交叉开关上的能量,  $E_{L_{bit}}$  是消耗在路由器之间网络链路上的能量。

除以上能耗之外, 还应该考虑源、宿节点处 NI 上的能量消耗  $E_{N_{bit}}$ 。

于是, 一比特数据从网络节点  $m(x_m, y_m)$  经过路径  $p_{mn}$  传输到网络节点  $n(x_n, y_n)$  所消耗的能量计算式为:

$$E_{bit}^{p_{mn}} = (|p_{mn}| + 1) \times E_{S_{bit}} + |p_{mn}| \times E_{L_{bit}} + 2E_{N_{bit}}$$

式中,  $|p_{mn}|$  表示路径  $p_{mn}$  长度, 也就是路径中包含的跳数。

2D-Mesh 网络中, 使用最短路径路由,  $|p_{mn}|$  的值就是  $m$  和  $n$  两节点间的曼哈顿距离  $Md_{mn}$ , 于是有:

$$E_{bit}^{p_{mn}} = (|x_m - x_n| + |y_m - y_n| + 1) \times E_{S_{bit}} + (|x_m - x_n| + |y_m - y_n|) \times E_{L_{bit}} + 2E_{N_{bit}}$$

若  $\Delta$  比特数据通过路径  $p_{mn}$ , 则能耗的计算式为:

$$E_{\Delta}^{p_{mn}} = \Delta \cdot E_{bit}^{p_{mn}}$$

该表达式为 NoC 网络通信能耗的精确计算模型。

路径尚未确定时, 对能耗进行估计可以采用网络平均路径长度。假设 2D-Mesh 网络的规模为  $X \times Y$ , 则其中任意两节点间的平均距离为:

$$Md_{ave} = \frac{1}{3} (X + Y)$$

于是, 可以得出平均路径下能耗计算式:

$$E_{\Delta}^{ave} = \Delta \cdot E_{bit}^{ave} \\ = \Delta \cdot \left[ \left( \frac{X+Y}{3} + 1 \right) \cdot E_{S_{bit}} + \frac{X+Y}{3} \cdot E_{L_{bit}} + 2E_{N_{bit}} \right]$$

该表达式为 NoC 网络通信能耗的粗略计算模型。

#### 3.2 NoC 能耗目标的计算

NoC 系统全局能耗的计算相对简单, 只需将特定映射方案下 IP 核上的能耗和网络上的能耗进行叠加。

$$E^{core} = \sum_{i=1}^{|C|} \sum_{j=1}^{|P|} k_j^i \cdot e_j^i$$

其中,  $k_j^i = \begin{cases} 1, & M_1(c_i) = ip_j \\ 0, & M_1(c_i) \neq ip_j \end{cases}$ ,  $e_j^i$  表示第  $i$  项任务由  $ip_j$  完成需要消耗的能量。

$$E^{net} = \sum_{i=1}^{|P|} \sum_{j=1}^{|P|} l_{ij} \cdot e_{ij}$$

其中,  $k_{ij} = \begin{cases} 1, & \text{存在通信 } ip_i \rightarrow ip_j \\ 0, & \text{其他情况} \end{cases}$ ,  $e_{ij}$  表示 IP 核  $ip_i$  和  $ip_j$

间通信时的网络能耗。由于安排在同一 IP 上的任务之间的通信能耗可以忽略,因此式中通信能耗  $E^{net}$  以核间通信计算。

映射第一个阶段中  $e_{ij}$  由网络能耗的粗略计算模型得到。第二个阶段中  $e_{ij}$  使用精确能耗计算模型得到。

### 3.3 NoC 延时模型

NoC 系统中的延时可以概括为系统硬件上的延时和因任务之间的约束关系带来的延时两大类。

硬件上的延时  $Delay$  主要来自 IP 核做数据处理时的时间消耗  $Delay^{core}$  和数据在片上网络中传输的时间消耗  $Delay^{net}$  两方面,可以表示为:

$$Delay = Delay^{core} + Delay^{net}$$

在规定时钟频率选定处理器类型后,完成相同的处理工作的时间取决于处理的数据量大小。 $\Delta$  比特数据在特定核上的处理能耗  $Delay_{\Delta}^i$  可以表示为:

$$Delay_{\Delta}^i = \rho_{ct} \cdot \Delta$$

式中,  $\rho_{ct}$  是与核类型相关的常数。

数据从源节点到达目的节点的延时包含 3 个部分:源、宿节点 NI 上的延时,传输路径中路由节点延时以及传输路径互联线延时。

若采用 Elmore 延时模型分析线上传输延时,对于一个具有  $n$  条线路的传输线,第  $k$  根线的传输延时  $T_k$  的归一化表达式近似表示为<sup>[8]</sup>:

$$\frac{T_k}{r_{TC_L}} = \begin{cases} (1+\lambda)\Delta_1^2 - \lambda\Delta_1\Delta_2, & k=1 \\ (1+2\lambda)\Delta_k^2 - \lambda\Delta_k(\Delta_{k-1} + \Delta_{k+1}), & 1 < k < n \\ (1+\lambda)\Delta_n^2 - \lambda\Delta_n\Delta_{n-1}, & k=n \end{cases}$$

据此计算式,对于既定的传输系统,可以认为线上延时与通信量呈线性关系。

在路由方式和工作频率确定的情况下,路由器用于处理和投递数据包的时间就只和数据量有关系。NI 的情况类似。

若  $\Delta$  比特数据从网络节点  $m(x_m, y_m)$  经过路径  $p_{mn}$  传输的网络节点  $n(x_n, y_n)$ ,传输过程中耗费的时间  $Delay_{\Delta}^{p_{mn}}$  的计算式为:

$$\begin{aligned} Delay_{\Delta}^{p_{mn}} &= \Delta \cdot [2\rho_{NI} + \rho_L |p_{mn}| + \rho_R (|p_{mn}| + 1)] \\ &= \Delta \cdot [2\rho_{NI} + \rho_L (|x_m - x_n| + |y_m - y_n|) + \rho_R (|x_m - x_n| + |y_m - y_n| + 1)] \end{aligned}$$

式中,  $\rho_{NI}$ ,  $\rho_L$ ,  $\rho_R$  分别是 NI、互联线和路由器关于传输时间的常数。 $|p_{mn}|$  是节点  $m$  和  $n$  之间的曼哈顿距离。该表达式为 NoC 网络延时的精确计算模型。

路径尚未确定时,可采用网络平均路径长度估计延时。假设 2D-Mesh 网络的规模为  $X \times Y$ ,则平均路径长度下的延时为:

$$Delay_{\Delta}^{ave} = \Delta \cdot \left[ 2\rho_{NI} + \rho_L \cdot \frac{X+Y}{3} + \rho_R \cdot \left( \frac{X+Y}{3} + 1 \right) \right]$$

上式为 NoC 网络延时的粗略估算模型。

### 3.4 NoC 延时目标的计算

NoC 系统全局延时的计算相对复杂,除了考虑硬件延时,还要考虑任务之间的约束延时,且不能简单地将各个 IP 核和每段通信链路的延时叠加。

映射第一阶段中的重点是计算完成一组基本任务的核上总处理延时。

任务图中所反映的先后约束在映射中必然存在。然而,在任务到 IP 核的分派过程中可能会引入新的约束关系。不考虑通信延时,任务执行的时间可以由甘特图表示,如图 2 所示。

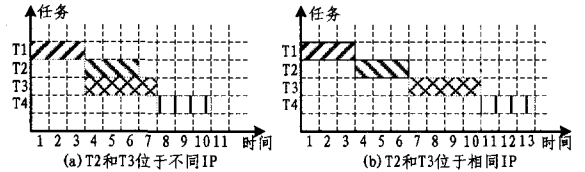


图 2 IP 核选择方式引入约束延时

图 2(a) 显示任务  $T_2$  和  $T_3$  位于不同 IP 核的情况,它们可以被同时处理。图 2(b) 中  $T_2$  和  $T_3$  位于同一个 IP 核,所以只能被先后执行,总执行时间也就比图 2(a) 多了 3 个单位时间。

选核方案同时影响甘特图中任务执行时间块的长度。IP 选择合理,可以缩短时间块长度,反之长度则会增加。

映射第二阶段中,研究重点集中在确定网络延时。此时,网络延时由任务图、任务与 IP 的绑定关系以及 IP 在 NoC 平台中的位置、通信量共同决定。映射在同一核上的任务通信延时忽略不计。可见,任务图中通信数据量大的两个任务不一定带来比通信数据量小的任务更大的延时。

为解决由通信任务图不能确定最长时间的问题,本文设计了通信时间图,即把原核通信图中边上的权值由通信量替换为通信时间。知道 IP 核之间的通信量以及通信路径的长度,利用网络延时的精确计算模型能够计算出核间通信延时。计算特定映射下的网络延时,首先作出通信时间图,在其中搜索输入节点到输出节点之间的最长路径,将路径长度(路径包含边的权值之和)作为系统网络延时。

## 4 混沌离散粒子群 NoC 映射

粒子群优化(PSO)的最初概念起源于对飞鸟群捕食行为的观察和研究,由 Kennedy 和 Eberhart 首先提出。由于其具有实现简单、收敛速度快等优点,因而在科学计算和工程应用领域得到了很好应用。PSO 在多目标优化问题的解决上也能取得优异的效果。

### 4.1 混沌离散粒子群优化

传统的 PSO 是一种在连续空间中寻找最优解的算法,不适合解决如调度、映射、路由之类建立在离散空间上的组合优化问题。

离散空间中的离散粒子群优化(DPSO)直接在离散的空间中执行搜索,寻找最优解<sup>[9]</sup>。基本思想是直接进行二进制位运算,不需要由连续空间到离散空间的映射函数做相关变换。文献[10]中提出了一种离散空间的二进制离散粒子群优化算法。算法表达式如下:

$$d_{1,i}^k = pBest_i^k \oplus X_i^k$$

$$d_{2,i}^k = gBest_i \oplus X_i^k$$

$$c_1 = rand(1, d), c_2 = rand(1, d)$$

$$V_i^{k+1} = c_1 \otimes d_{1,i}^k + c_2 \otimes d_{2,i}^k$$

$$X_i^{k+1} = X_i^k \oplus V_i^{k+1}$$

若用重量  $W$  表示一个二进制向量中的“1”的个数,则速度限制条件表示为:

$$W(V_i^k) \leq w_{max}$$

式中,常量  $w_{max}$  表示速度的最大允许重量。在迭代过程中,若发生  $W(V_i^k) > w_{max}$ ,则随机地将  $V_i^k$  中  $W(V_i^k) - w_{max}$  个位置上的“1”修改为“0”。

粒子群优化算法缺乏摆脱局部最优的机制,容易陷入局部最优解。为了使离散粒子群优化的性能更优,本文引入混沌扰动的概念,使算法搜索时能跳出局部最优。

混沌扰动的基本思想是利用混沌函数产生一个扰动向量  $\Delta X$ ,在粒子  $X_i^k$  完成位置更新后,将扰动附加到  $X_i^{k+1}$  上,得到扰动后的新位置  $\overline{X_i^{k+1}}$ 。若扰动后的位置优于之前的位置,则由此位置替换  $X_i^{k+1}$ 。扰动向量  $\Delta X$  的产生方法是:

1) 设上一次迭代中的混沌向量为  $\mu^k = (\mu_1^k, \mu_2^k, \dots, \mu_d^k)$ ,  $\mu_i^k \in [0, 1]$ ;

2) 本次迭代中的混沌向量  $\mu^{k+1}$  的任意分量  $\mu_i^{k+1}$  由 Logistic 映射给出:

$$\mu_i^{k+1} = 4\mu_i^k(1 - \mu_i^k)$$

3) 根据混沌向量的各分量确定扰动量  $\Delta X^{k+1}$  的对应分量,其关系式为:

$$\Delta x_i^{k+1} = \begin{cases} 0, & \mu_i^{k+1} < 0.5 \\ 1, & \mu_i^{k+1} \geq 0.5 \end{cases}$$

至此,一种具有混沌扰动机制的离散空间的离散二进制粒子群优化算法(简称混沌离散粒子群优化,CDPSO)已建立。

## 4.2 算法建模与数据结构设计

用混沌离散粒子群优化求解最优的 NoC 映射优化方案需要将实例的特征同算法相结合,设计恰当的数据结构,建立起适合搜索算法操作的数学模型。

### 4.2.1 从任务到 IP 核的映射算法

映射搜索的第一阶段,完成以能耗和延时最小化为目标的 IP 核选择。任务向 IP 核的分派结果用向量  $X = (x_1, x_2, \dots, x_N)^T$  表示,其含义是编号为  $i$  的任务在 IP 核  $x_i$  上完成。由于不同任务可以同时被分派到相同 IP 核,因而可以出现  $x_i = x_j$  的情况。解向量中元素的个数  $N$  和任务图中的任务数相同,  $x_i$  是特定 IP 核的编号。若将此编号用二进制数表示,并将此二进制数中包含的数据位作为行向量,则可将  $X$  表示为只含 0、1 的矩阵:

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \triangleq \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ \vdots & \ddots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NM} \end{pmatrix}$$

式中,  $x_{ij} = 0$  或  $1$ ,  $M \geq \lceil \log_2 N \rceil$ 。这里的  $X$  也就是 CDPSO 算法中的一个粒子。

在此编码方式下,CDPSO 中所有对向量的运算都扩展为对矩阵的运算。同时,原来 CDPSO 中的参数  $c_1$ 、 $c_2$  以及混沌扰动量  $\mu$  都更换为对应的矩阵形式。

CDPSO 算法迭代中计算粒子适应度时,任务间通信能耗和延时使用粗略估算模型。

目标函数的设计上,本文采用理想点法。首先分别定出能量  $Energy(M)$  和延时  $Delay(M)$  的理想期望值  $E^*$  以及  $D^*$ ,于是目标函数定义为:

$$fitness = \lambda_e (Energy(M) - E^*)^2 + \lambda_d (Delay(M) - D^*)^2$$

其中,权值  $\lambda_e$  和  $\lambda_d$  分别视具体问题而定。原多目标规划问题转化为  $\min_{M \in MAP} (fitness)$ 。

表 1 是算法的伪码描述。其中,  $fitness(\cdot)$  表示对粒子的计算适应度(即目标函数),它与能量和延时相关。而  $Chaos(\cdot)$  表示对粒子做混沌扰动操作。

表 1 CDPSO 映射算法伪码

(1) 粒子总数 = $n_p$ , 最大迭代次数 = $n_{max}$ , 随机产生 $n_p$ 个初始解 $X_1^1 \sim X_{n_p}^1$ 。
(2) 计算各个粒子的初始适应度 $fit_1^1 \sim fit_{n_p}^1$ 。设置各粒子的初始个体最优解 $pBest_1^1 \sim pBest_{n_p}^1$ 为当前初始解,找到初始全局最优解 $gBest^1$ 。
While(当前迭代次数 $i < n_{max}$ ) do
While( $j < n_p$ ) do
(3) 由 CDPSO 定义式得到第 $j$ 个粒子的新位置 $X_j^i$
(4) $X_j^i = Chaos(X_j^i)$
$fitness(X_j^i) < fitness(X_j^{i-1})$
Then $X_j^i = X_j^{i-1}$
(5) $fitness(X_j^i) < fitness(pBest^{i-1})_j$
Then $pBest_j^i = X_j^i$ 。
End of While
(6) 对于所有粒子的个体最佳值若存在 $pBest_j^i$ ,
If $fitness(pBest_j^i) < fitness(gBest^{i-1})$
Then $gBest^i = pBest_j^i$ 。
End of While
(7) 迭代结束,得到全局最优解 $gBest$ 。

在以上算法执行之后,任务图中的任务都被分派到了可选的 IP 核上。

### 4.2.2 从 IP 核到 NoC 网络平台的映射算法

NoC 映射算法执行的第二阶段,完成以能耗和延时最小化为目标的 IP 核到 NoC 平台的位置映射。第二阶段算法的初始输入是第一阶段可行解集合。这一阶段的搜索过程仍然采用混沌离散粒子群优化算法,但具体的建模过程不同。

由于路由节点必须和 IP 核建立唯一绑定的关系,IP 核到 NoC 平台路由节点的映射是单射。而 CDPSO 在完成粒子位置更新过程中粒子各分量呈现不确定性,很可能出现解向量中两个或两个以上分量相同的情况。

为了解决上述矛盾,本文设计 NoC-IP 对应表这一数据结构(见图 3 中第二列和第三列)。该表第一行的表头数值代表 NoC 平台上各路由节点的编号,第二行数数值代表与相应节点绑定的 IP 核编号。其中的  $x$  标记表示该路由节点不与 IP 核绑定,反映了 IP 核数量少于路由节点的情况。

基准 NoC-IP 对应表是指在算法执行的迭代过程始终保持不变的特殊 NoC-IP 对应表。它是每次迭代过程的一个基准参考,每次迭代中产生的新对应表都在基准 NoC-IP 对应表的基础上变换得到。

粒子仍然使用向量  $X = (x_1, x_2, \dots, x_N)^T$  来表示,其中  $N$  为 NoC 网络平台上可用路由节点的总数。向量  $X$  反映与基准 NoC-IP 对应表的相对位置关系。 $X$  中任意分量  $x_i$  的含义是将基准 NoC-IP 对应表中表头编号为  $i$  的表项同表头编号为  $x_i$  的表项内容互换(见图 3)。对应于 NoC 平台就是将第  $i$  个路由节点位置上的 IP 核同第  $x_i$  个路由节点位置上的 IP

核对调。

图3演示了算法进行3次迭代中粒子同NoC-IP对应表之间的相互关系。图中粒子的第一行是分量编号,第二行是各分量的具体数值。第一代粒子的分量1的数值为2,表示将基准NoC-IP对应表中的第1、第2路由节点处的IP核对调,得到第一次调换结果;接着扫描粒子分量2,其值为3,表示将第一次调换结果后的第2、第3路由节点处的IP对调。以此类推,直至完成粒子8个分量表示的调换操作。NoC-IP表的第一轮调换的完整过程可以记录为(1↔2)→(2↔3)→(3↔7)→(4↔3)→(5↔4)→(6↔1)→(7↔5)→(8↔2)。图4对该过程进行了较为形象的描述。简明起见,图中除了基准NoC-IP对应表之外,其他NoC-IP对应表的表头均未画出。

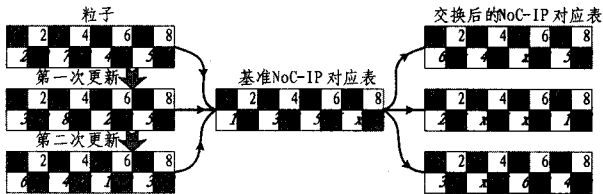


图3 粒子更新与交换规则示意

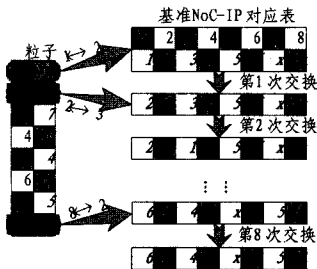


图4 NoC-IP对应表交换操作

图3中,以第一代粒子为基础的交换操作结束后,粒子会按照混沌离散粒子群算法更新自身位置,产生下一代粒子。下一代粒子再次和基准NoC-IP对应表一起确定新一轮交换后的NoC-IP对应表。粒子继续位置更新,产生新位置,结合基准NoC-IP表算法得到下一次交换后的映射表。

引入NoC-IP对应表,在抽象的粒子和NoC映射方案之间建立起联系,应对单射这一约束条件。通过评价由特定粒子确定的NoC-IP对应表即可获知此粒子的适应度。

第二阶段搜索同第一阶段的不同之处在于算法的起点和粒子适应度的计算。粒子每次位置更新后都要计算出此时粒子所对应的NoC-IP表,结合2D-Mesh网络结构的NoC平台特点,利用延时、功耗的精确模型计算粒子的适应度。

## 5 实验与结果分析

为验证本文提出的算法的性能,这里将算法应用于两个实例的NoC映射方案搜索中。第一个实例为复合多媒体系统(MMS, MultiMedia System)<sup>[7]</sup>。任务图中的40个任务将被分派到16个特定的IP核上,种类包括了DSP、通用CPU、嵌入式的DRAM以及ASIC。而IP核会进一步被映射到一个4×4规模的2D-Mesh NoC平台上。

第二个实例为视频解码系统<sup>[11]</sup>。该应用实例的任务被分派给DSP、图形处理器、RISC CPU、SDRAM等不同类型的IP核处理。映射的目标是4×4规模的2D-Mesh NoC平台。

映射由基于CDPSO的映射方案搜索算法完成。用基本

遗传算法同本文提出的算法进行效果对比。参数设置上,CDPSO包含的粒子个数为100,遗传算法中染色体个数为100,交叉变异概率分别为0.75和0.1。为了更直观地反映优化算法的效果,我们随机生成了10万种映射方案,用以同优化算法得到的映射方案相比较。

图5是CDPSO算法在映射搜索中的收敛过程。从图中可以看出,CDPSO在能耗和延时两个目标上的收敛速度都很快。观察图形可以发现,曲线上出现的起伏在一个目标的曲线的反弹阶段,另一个目标的曲线必定在下降。这反映了能耗和延时两个目标变化的不一致性,同时也正是CDPSO处理多目标优化问题的特点。

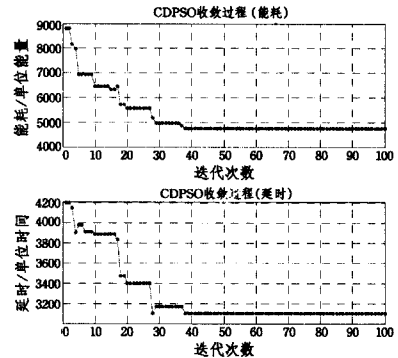


图5 CDPSO在视频解码应用NoC映射中的收敛过程

图6显示了混沌离散粒子群优化以及遗传算法搜索到实例MMS的面向能耗和延时映射方案的Pareto-front,以及随机映射点。

图7是对图6左下角的局部放大。从图7中可以看出,两种优化算法得到的Pareto-front在较少迭代次数下都能得到比大量随机映射更优的解。同时,遗传算法的结果与随机映射点的密集区距离很接近,优势并不明显。混沌离散粒子群优化得到的解在延时和能量两个方面都有很好的优化效果,其Pareto-front相距随机点集中区距离较大。CDPSO得到的映射结果在能耗上比遗传算法低23.5%,比随机映射的最好结果低32.5%。在延时方面,其结果比遗传算法低10.7%,比随机映射最好值低16.9%。

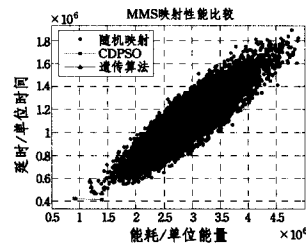


图6 MMS实例的映射结果比较

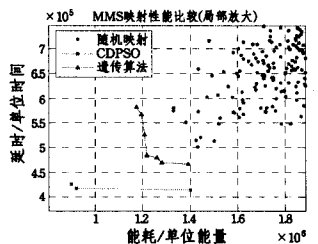


图7 MMS实例的映射结果比较局部放大

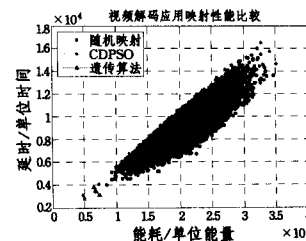


图8 视频解码应用的映射结果比较

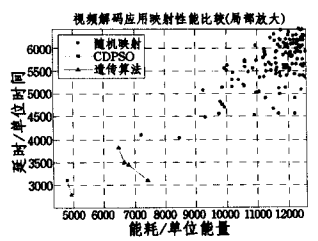


图9 视频解码应用的映射结果比较局部放大

图8中绘出了对视频解码应用随机映射得到的能耗/延时点,并且分别绘出了由CDPSO和遗传算法得到映射结果的Pareto-front。从图9局部放大的结果中可以清楚地看到,CDPSO所搜索到的映射结果较之遗传算法和随机映射都有较大的优势。图中显示的结果表明,CDPSO得到的映射方案比被随机映射得到的最好结果在能耗上低33.7%,延时上低31%,比遗传算法得到的结果在能量和延时上分别低了26.2%和10.1%。

**结束语** 本文设计了混沌离散粒子群优化算法,针对NoC映射的两阶段分别提出了能耗/延时粗略估算模型和精确计算模型,对异构多核协作系统进行以能耗和延时为优化目标的分阶段NoC映射求解。在对复杂多媒体系统进行仿真的实验中,本文提出的映射算法与随机映射和基本遗传算法相比表现出更良好的性能。

### 参考文献

[1] Marculescu R, Ogras U Y, Peh L-S, et al. Outstanding Research Problems in NoC Design, System, Microarchitecture, and Circuit Perspectives[J]. IEEE Transaction on Computer-aided Design of Integrated Circuits and systems, 2009, 28(1): 3-21

[2] Kahng A B, Li Bin, Peh L-S, et al. ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration[C]//Proceedings of Design, Automation and Test in Europe. 2009: 423-428

[3] Singh A K, Wu Ji-gang, Prakash A, et al. Mapping algorithms for NoC-based heterogeneous MPSoC platforms[C]//12th Eu-

romicro Conference on Digital System Design: Architectures, Methods and Tools. 2009:133-140

[4] Ghosh P, Sen A. Efficient mapping and voltage islanding technique for energy minimization in NoC under design constraints [C]//Proceedings of the 2010 ACM Symposium on Applied Computing. 2010:535-541

[5] 常政威,熊光泽,桑楠,等.基于电压岛的能量和可靠性感知NoC映射[J].计算机辅助设计与图形学学报,2009,21(1):19-26

[6] 杨盛光,李丽,高明伦,等.面向能耗和延时的NoC映射方法[J].电子学报,2008,36(5):937-942

[7] Hu J, Marculescu R. Energy- and performance-aware mapping for regular NoC architectures[J]. IEEE Transaction on Computer-aided Design of Integrated Circuits and Systems, 2005, 24(4):551-562

[8] Sotiriadis P P. Interconnect modeling and optimization in deep sub-micron technologies [D]. Cambridge, USA: Massachusetts Institute of Technology, 2002: 23-202

[9] 沈林成,霍霄华,牛铁峰.离散粒子群优化算法研究现状综述[J].系统工程与电子技术,2008,30(10):1986-1990

[10] Afshinmanesh F, Marandi A, RahimisKian A. A Novel Binary Particle Swarm Optimization Method Using Artificial Immune System[C]//EUROCON 2005 - The International Conference on Computer as a Tool. 2005,1:217-220

[11] Van der Tol E B, Jaspers E G T. Mapping of MPEG-4 decoding on a flexible architecture platform[C]//Proceedings of SPIE-Media Processors. 2002,4674:1-13

(上接第266页)

表3 异常运动检测结果

异常行为	预警数	正确预警数
跑动	19	15
来回运动	31	30

本文为视觉监控场景语义分析提供了一个一般化的框架,用于推理监控场景中结构化的运动规则。对于更为复杂的监控视频,例如购物商场,场景中运动对象的数量巨大,不能进行有效的检测跟踪,因此基于轨迹的场景理解技术应用具有局限性,目前直接利用视觉运动线索或使用知识模型进行场景分析已经有很多的研究成果<sup>[11,12]</sup>,未来的研究会尝试将其纳入分析框架中,使自动分析方法对监控环境限制具有更强的适应性。

### 参考文献

[1] Wang Su-yu, Shen Lan-sun. Intelligent Visual Surveillance Technology: A Survey[J]. Journal of Image and Graphics, 2009, 12(9):1505-1514

[2] Gu Jun-xia, Ding Xiao-qing. A Survey of Activity Analysis Algorithms[J]. Journal of Image and Graphics, 2009, 14(3):377-387

[3] 王昊冉,白亮,老松杨.基于图模型的足球视频语义建模方法[J].计算机科学,2011,38(6):266-269

[4] Turaga. Machine Recognition of Human Activities: A Survey [J]. IEEE Transactions on Circuits and Systems for Video

Technology, 2008, 18(11):1473-1488

[5] Xiang, Gong. Beyond Tracking: Modelling Activity and Understanding Behaviour [J]. International Journal of Computer Vision, 2006, 67(1): 21-51

[6] Wang Xiao-gang. Trajectory Analysis and Semantic Region Modeling Using A Nonparametric Bayesian Model [C]//Proceedings of IEEE Computer Society Conference on Computer Vision and Patter Recognition. Anchorage, USA, 2008:1-8

[7] Wang Xiao-gang. Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models [J]. IEEE Transactions on PAMI, 2009, 31(3): 539-555

[8] Mitchell. Machine Learning [M]. China Machine Press, 2003

[9] Wang Xiao-gang. Learning Semantic Scene Models by Trajectory Analysis [C]//ECCV 2006. Graz, Austria, 2006:110-123

[10] Jiao Bo. Research on Moving Object Detection and Tracking Methods for Intelligent Video Surveillance [D]. Changsha: National University of Defense Technology, 2009

[11] Davis J W, Bobick A F. The Representation and Recognition of Action Using Temporal Templates [C]//IEEE Conference on Computer Vision and Pattern Recognition. Santa Barbara, USA, 1997:928-934

[12] Lavee G. Video Event Modeling and Recognition in Generalized Stochastic Petri Nets [J]. IEEE Transaction on Circuits and Systems for video technology, 2010, 20(1):102-118