面向时延的 NoC 映射技术研究*

易宏波[1]，罗兴国[1]，陈　韬[1]，刘　静[1]，桑晓丹[2]
（1. 解放军信息工程大学，郑州 450002；2. 中国人民解放军 72556 部队，济南 250002）

摘　要：针对 NoC 任务映射问题中时延难以预测和启发式算法效率低的问题，提出一个时延改进模型和近邻随机遗传算法。该模型从宏观的链路负载分布和单个节点的排队时延两方面来构建 NoC 映射的时延模型，通过引入时延因子、权重系数来刻画不同映射方案对时延性能的影响，避免了 NoC 通信时延精确建模的难题。提出近邻随机思想来构建遗传算法的初始种群，并且运用该算法实现了面向时延的 NoC 映射，在达到全局最优的情况下，比经典遗传算法效率提升将近 20%。实验结果表明，该算法优于现有的经典遗传算法和随机映射方案。

关键词：片上网络；映射；时延模型；近邻随机；遗传算法
中图分类号：TP316　　**文献标志码**：A　　**文章编号**：1001-3695(2012)06-2325-04
doi:10.3969/j.issn.1001-3695.2012.06.086

Research on delay-aware NoC mapping method

YI Hong-bo[1], LUO Xing-guo[1], CHEN Tao[1], LIU Jing[1], SANG Xiao-dan[2]
(1. *PLA Information Engineering University*, *Zhengzhou* 450002, *China*; 2. *PLA* 72556 *Army*, *Ji'nan* 250002, *China*)

Abstract: Due to the delay in NoC task mapping is difficult to predict, and the low efficiency in heuristic algorithm. This paper proposed an improved delay model and nearest-neighbor random genetic algorithm(NNRGA). It constructed the NoC mapping delay model from the macroscopic link load distribution and the single node queue latency. Different mapping schemes influenced the performance of delay through importing delay factor and weight coefficient, the model avoided the difficulty to model communicating delay in NoC accurately. This paper proposed a method to construct initial population of genetic algorithm based on the thought of nearest neighbor and random. It used NNRGA to realize the delay-aware NoC mapping. And the efficiency increased by nearly 20% compared with the classical genetic algorithm when achieving the global optimum situation. the experimental results show that the algorithm is better than the classical genetic algorithm and random mapping algorithm.
Key words: NoC; mapping; delay model; nearest neighbor and random; genetic algorithm

0　引言

NoC 映射问题是一个 NP-hard 问题[1]，对于这种组合优化问题，多采用启发式算法求解，但是启发式算法往往存在停滞现象和搜索速度慢等缺陷。文献[2]采用平移交叉算子和交换变异算子，算法效率得到提升，但由于初始种群的随机性，可能会出现停滞现象。文献[3]采用排序编码，运用边重组交叉算子和倒置变异算子进行映射，能有效避免停滞现象的发生。

由于 NoC 传输时延的动态性，因此难以对时延进行精确建模。文献[4,5]基于排队理论得到通用路由结构下的时延模型；文献[6]运用排队理论，提出类似共享总线等待时间建模的方法来估计 NoC 等待时间，但计算复杂度高；文献[7]通过优化链路负载分布间接优化时延，联合能耗进行优化，但对于目标函数优化系数的确定缺少理论性；文献[8]提出平均边时延、关键路径的概念，建立了 NoC 时延模型，但对于影响时延的因素考虑不够。

针对以上问题，本文从宏观的链路负载分布和单个节点的排队时延两个角度来构建 NoC 映射下的时延模型，引入时延因子的概念，并针对 NoC 映射的实现算法，通过实验仿真确定了权重系数。针对启发式算法搜索速度慢的不足，提出一种基于近邻思想和随机思想来生成初始种群的遗传算法。

1　NoC 映射的时延模型

NoC 映射是 NoC 设计中的重要步骤，就是在给定通信任务图和拓扑结构图的前提下，针对特定的约束条件和设计目标，将通信任务分配到 NoC 拓扑结构上的对应位置。为了清晰地描述映射问题，假设任务和资源节点一一对应，即每个任务只能由处理资源完成，而每个处理资源只完成一个任务，这也是研究 NoC 映射通用的假设条件，NoC 映射优化问题就抽象成通信任务图到通信资源图之间的映射问题。为了使时延性能得到提升，本文提出的 NoC 映射时延模型从宏观的链路负载分布和单个节点的排队时延两方面考虑，对 NoC 映射时延进行优化。

定义 1　时延因子。令 latency_factor 表示为时延因子。

收稿日期：2011-10-09；**修回日期**：2011-11-15　　**基金项目**：国家“863”计划资助项目(2009AA012201)；上海市科委重大科技攻关项目(08dz501600)

作者简介：易宏波(1986-)，男，湖南湘潭人，硕士研究生，主要研究方向为片上网络(yihongbo19861021@163.com)；罗兴国(1951-)，男，重庆人，教授，博导，主要研究方向为计算机网络体系结构；陈韬(1979-)，男，湖北嘉鱼人，博士研究生，主要研究方向为高性能计算机体系结构；刘静(1984-)，男，山西人，博士研究生，主要研究方向为数据挖掘、认知流识别；桑晓丹(1985-)，女，河南人，硕士研究生，主要研究方向为片上网络映射技术.

时延因子包含两部分：a）宏观部分的链路负载分布方差 variance_link；b）单个节点部分的排队时延 latency_queue。时延因子 latency_factor 表达式如下所示：

$$\text{latency\_factor} = w_1 \frac{\text{variance\_link}}{\text{variance\_link}_0} + w_2 \frac{\text{latency\_queue}}{\text{latency\_queue}_0}$$

由于 variance_link 和 latency_queue 在量纲上存在区别，首先对 variance_link 和 latency_queue 进行归一化处理，其中 variance_link_0，latency_queue_0 为平均链路负载方差和平均排队包时延。其次，引入权重系数 w_1 和 w_2，其作用是使 NoC 时延性能在宏观和单个节点两方面得到均衡优化，平均两方面对时延的作用，取得一个平衡值，w_1 和 w_2 的值可通过实验仿真来确定。

综上所述，时延因子 latency_factor 从宏观的链路负载分布和单个节点的排队时延两个方面来反映 NoC 映射的时延效果，不同的映射结果得到的时延因子是不一样的。时延因子 latency_factor 越大，表明映射结果产生的时延越大，latency_factor 越小，表明产生的时延越小，映射效果更好。本文提出时延模型的优化目标函数，即最小化时延因子 min(latency_factor)。其中链路负载分布方差 variance_link 和排队时延 latency_queue 如式(1)所示。

链路负载方差：

$$\text{variance\_link} = \sum_{i=1}^{M} [\text{load}(l_i) - \text{load}(l)_{\text{avg}}]^2 / M \tag{1}$$

其中：l_i 为 NoC 中第 i 条链路，M 为链路总数。

链路负载计算表达式为

$$\text{load}(l_k) = \sum_{i=1}^{N}\sum_{j=1}^{N} \text{pass}_{ij}^{k} \times w_{ij} \tag{2}$$

式中：

$$\text{pass}_{ij}^{k} = \begin{cases} 1 & \text{when } p_{ij} \text{ pass by } l_k \\ 0 & \text{else} \end{cases} \tag{3}$$

w_{ij} 表示通信任务图权重值矩阵中任务 i 到任务 j 的通信量，单位是 bit。

平均链路负载为

$$\text{load}(l)_{\text{avg}} = \sum_{i=1}^{M} \text{load}(l_i) / M \tag{4}$$

排队时延由 M/G/1 模型的推导公式可得

排队时延为

$$\text{latency\_queue} = \frac{1}{\sum_{\forall s,d} x_{s,d}} \sum_{\forall s,d} x_{s,d} \times L_{s,d} \tag{5}$$

其中，$L_{s,d}$ 是任一从源节点 s 到目的节点 d 的数据包的平均时延，$x_{s,d}$(packet/s)是从源节点 s 到目的节点 d 的传输速率。

$$L_{s,d} = W_s + \sum_{(i,j)\in \Pi_{s,d}} (W_{i,j} + T) \tag{6}$$

其中：W_s 表示在源节点 s 的排队时延；T 表示平均服务时间；$\Pi_{s,d}$ 表示数据包从源节点 s 到目的节点 d 传输时经过的路由节点和相应的链路。$W_{i,j}$ 表示路由节点 i 链路 j 的排队时延。

$$W_{i,j} = N_{i,j} / \lambda_{i,j} \tag{7}$$

式中，$\lambda_{i,j}$ 表示数据包到达路由节点 i 链路 j 的速率。

$$\lambda_{i,j} = \sum_{\forall s}\sum_{\forall d} x_{s,d} R(s,d,i,j) \tag{8}$$

其中：

$$R(s,d,i,j) = \begin{cases} 1 & \text{当源节点 } s \text{ 到目的节点 } d \text{ 路线经过路由节点 } i \text{ 和链路 } j \text{ 时} \\ 0 & \text{else} \end{cases} \tag{9}$$

式中：$N_{i,j}$ 表示数据包在路由节点 i 链路 j 的平均包个数。

$$N_{i,j} = \lambda_{i,j}^2 E(B^2) / 2(1 - E(B)\lambda_{i,j}) \tag{10}$$

$$W_s = \lambda_s E(B^2) / 2(1 - E(B)\lambda_s) \tag{11}$$

其中：$E(B^2)$ 是服务时间的第二阶段[9]，$E(B)$ 是平均服务时间，$D(B) = E(B^2) - E(B)^2$。

2 近邻随机初始种群的遗传算法

针对经典遗传算法在收敛过程中易陷入局部最优和搜索效率低的问题，本文提出一种基于近邻随机思想构建初始种群的遗传算法，来提高算法运算效率。用遗传算法解决映射问题时，首先生成初始种群，然后通过遗传操作，最终获得所求问题的最优解。所以初始种群的组成对遗传算法的计算结果和计算效率存在一定影响。

在运用遗传算法解决 NoC 映射问题时，适应度值高的初始种群可以减少算法收敛到最优解的时间，提高算法效率。如果初始种群全由近邻思想构成的话，能取得较高适应度值的初始种群个体，但会降低初始种群的多样性。基于算法效率和种群多样性出发，遗传算法的初始种群由近邻种群和随机种群两部分构成。近邻随机初始种群的遗传算法操作步骤如下：

输入：fitness_latency（目标函数）、NP（种群个体数）、NG（最大进化代数）、pc（交叉概率）、pm（变异概率）。

输出：xv（目标函数最小值时的个体）、fv（目标函数的最小值）。

a）种群初始化。产生一个大小为 NP 的种群 A_0；初始化 $A_0 = 0$，$n_0 = 0$，其中 $n_0 = 0$ 是链路负载值；读取通信任务图权重值矩阵 W_{ij}；根据 W_{ij} 中通信任务间的通信关系，生成 $NP/2$ 个近邻初始种群，随机生成 $NP/2$ 个随机初始种群，对个体进行编码。

b）确定个体的适应度。用轮盘赌策略确定个体的适应度，并判断是否符合优化准则，若符合，输出最佳个体及其代表的最优解，并结束计算，否则转向 c）。

c）选择操作。依据适应度选择再生个体，适应度高的个体被选中的概率高，适应度低的个体可能被淘汰。

d）交叉操作。按照概率 pc 和一定的交叉方法产生新个体。

e）变异操作。按照概率 pm 和一定的变异方法产生新个体。

f）由交叉和变异产生新一代的种群，返回到 b）。

由于任务和资源节点是一一对应关系，染色体编码采用实值编码，一组染色体用一个数组表示，染色体中的基因和 NoC 拓扑结构资源节点的位置一一对应，通信任务图的个数小于 NoC 拓扑结构资源节点的个数。为了对 NoC 映射问题下的时延性能进行优化，对时延因子 latency_factor 取反即为算法的适应度函数。

选择是在群体中选择生命力强的个体产生新群体的过程，选择操作的主要目的是为了避免有用遗传信息的丢失，提高全局收敛性和计算效率。本文采用轮盘赌选择策略，令 $PP_i = \sum_{j=1}^{i} p_i$，$PP_0 = 0$，$PP_i$ 为累计概率，p_i 为个体的选择概率，其计算公式为 $p_i = \text{fitness\_latency}(x_i) / \sum_{i=1}^{NP} \text{fitness\_latency}(x_i)$，其中 fitness_latency($x_i$) 为个体的适应度。共轮转 NP 次（NP 为种群个体数），每次轮转时，随机产生 0 ~ 1 的随机数 r，当 $PP_{i-1} < r < PP_i$ 时选择个体。

由于 NoC 映射问题采用实值编码,且一个通信任务只映射到一个资源节点上,为了避免交叉后的重复值,本文采取的交叉和变异操作有别于传统的遗传算法。每次进行交叉和变异操作时,首先分别生成两个随机数 r_{pc} 和 r_{pm}。当交叉随机数 $r_{pc} < pc$ 时,随机选择个体中的两个交叉位,对其位置上的值进行互换,产生交叉后的新个体;当变异随机数 $r_{pm} < pm$ 时,随机选择个体中的两个变异位,对其位置进行互换,产生了变异后的新个体。交叉操作算法伪代码描述如下:

```
for i = 1:NP
    sita = rand( );% 生成一个 0:1 的随机数
     for n = 1:NP
      if sita < = PPx(n)
      SelFarther = n;% 根据轮盘赌策略确定交叉个体
        break;
        end
      end
posCut_1 = floor(rand( ) * 15) + 1;% 随机确定交叉点 1
posCut_2 = floor(rand( ) * 15) + 1;% 随机确定交叉点 2
    r1 = rand( );
    if rpc < = pc
      nx = A0(SelFarther,posCut_1);
A0(SelFarther, posCut_1) = A0(SelFarther,posCut_2);
  A0(SelFarther, posCut_2) = nx;
```

3　实验仿真与结果分析

NoC 映射使用 MATLAB_R2008b 仿真环境,基于 i7 处理器,在 Windows XP 上运行。实验采用 4 × 4 的 2D-Mesh 网络拓扑结构、XY 经典路由算法、虫孔交换方式。

实验针对 NoC 映射常用的两类通信任务图:a)随机任务通信图,由 TGFF[10] 软件生成,广泛用于构建随机状态下的任务通信图[11];b)视频对象平面解码器(video object plane decoder,VOPD)。NoC 映射技术文章多采用 VOPD 作为实际应用下的仿真实验。

由于随机映射算法被广泛用于 NoC 映射实验中对算法的比较,实验通过比较随机映射、经典 GA 和 NNRGA 三种算法在随机任务通信图和 VOPD 下达到时延最优解时的值以及运行的时间,确定时延模型的权重系数和验证 NNRGA 的性能。

具体实验步骤如下:

a)构建随机生成图和 VOPD 通信任务图;

b)确定时延因子 latency_factor 表达式中常量 $variance\_link_0$ 和 $latency\_queue_0$ 的值;

c)随机映射通信任务图到 NoC 资源节点;

d)运用经典遗传算法进行映射;

e)运用 NNRGA 进行映射。

随机任务通信图和 VOPD 实验结果分别如下。

3.1　随机通信任务图

TGFF 生成的随机通信任务图如图 1 所示。在随机任务通信图下,随机映射算法、GA、NNRGA 的仿真时延结果如图 2 所示。从表 1 和图 2 可知,随机映射算法、经典 GA 和 NNRGA 时延因子 latency_factor 的值都小于 1。与随机映射算法相比,经典 GA 的 latency_factor 降低了 29.8%,其中 variance_link 和 latency_queue 分别降低了 61.3% 和 11.4%;NNRGA 的 latency_factor 降低了 33%,其中 variance_link 和 latency_queue 分别降低了 71.5% 和 7.25%,表明该时延模型能较好地反映不同映射方案对时延性能的影响。

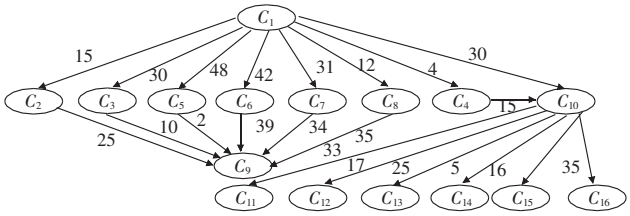

图 1　随机任务通信

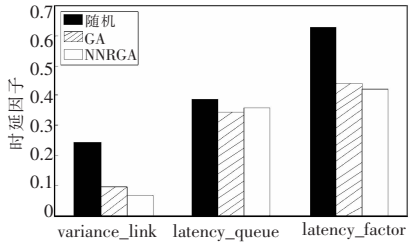

图 2　随机任务通信仿真时延柱状图

表 1　随机任务通信图实验仿真数据

算法	variance_link	latency_queue	latency_factor
随机映射	0.241 3	0.386	0.627 3
经典 GA	0.093 3	0.341 9	0.44
NNRGA	0.068 6	0.358 4	0.42

由图 2 可知,NNRGA 的 variance_link 比经典 GA 降低了 26.4%,latency_queue 增加了 4.8%,得出的 latency_factor 几乎相同,表明 NNRGA 得到的解在 variance_link 和 latency_queue 项取值不一定比经典 GA 好,但 NNRGA 能取到目标函数的最优解。NNRGA 实验取得的最优解的初始解编号是 146,该映射方案是由近邻思想生成的初始解,而 NNRGA 的运行时间相比于经典 GA 节省了 19%,表明基于近邻随机思想生成的初始种群在保证全局最优的条件下,算法效率较经典 GA 得到了显著提高。

3.2　VOPD 通信任务图

VOPD 通信任务图如图 3 所示。

在视频对象平面解码任务通信图下,GA、NNRGA 的仿真时延结果如图 4 所示。从表 2 和图 4 可知,随机映射算法、经典 GA 和 NNRGA 时延因子 latency_factor 的值都小于 1。与随机映射算法相比,经典 GA 取得的 latency_factor 降低了 40.5%,其中 variance_link 和 latency_queue 分别降低了 67.1% 和 7.1%;NNRGA 取得的 latency_factor 降低了 41.6%,其中 variance_link 和 latency_queue 分别降低了 63.2% 和 11.4%,表明该时延模型能较好地反映不同映射方案对时延性能的影响。

由图 4 可知,NNRGA 的 variance_link 比经典 GA 增加了 10%,latency_queue 降低了 4.6%,得出的 latency_factor 相近,表明 NNRGA 得到的解在 variance_link 和 latency_queue 项取值不一定比经典 GA 好,但 NNRGA 能取到目标函数的最优解。NNRGA 实验取得的最优解的初始解编号是 269,该映射方案是由随机思想生成的初始解,而 NNRGA 的运行时间相比于经典 GA 节省了 19%,表明基于近邻随机思想生成的初始种群在保证全局最优的条件下,算法效率较经典 GA 算法得到了提高。

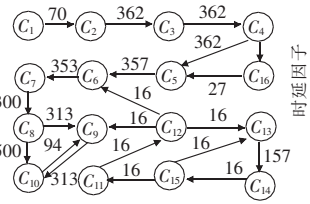

图 3 视频对象平面解码任务通信图(VOPD)任务

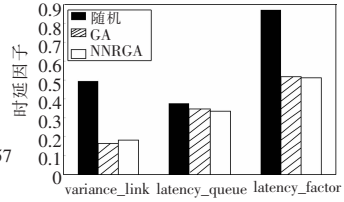

图 4 视频对象平面解码通信仿真时延柱状图

表 2 视频对象平面解码任务通信图实验仿真数据

算法	variance_link	latency_queue	latency_factor
随机映射	0.495 4	0.373 9	0.874 6
经典 GA	0.162 8	0.347 2	0.52
NNRGA	0.179 2	0.331 1	0.51

由表 1 中的 variance_link、latency_queue 数据可知,在随机任务通信图下,三种算法得到 latency_queue/variance_link 的值分别是 1.59、3.67、5.22。为了均衡负载方差和排队时延两方面对时延的优化作用,针对随机映射算法、经典 GA 和 NNRGA,时延模型权重系数 w_1 和 w_2 分别取为 1.6、3.6 和 5.2,从而能平衡两方面对时延的作用。由表 2 中的 variance_link、latency_queue 数据可知,在 VOPD 任务通信图下,三种算法得到 latency_queue/variance_link 的值分别是 0.754、2.13、1.84,针对随机映射算法、经典 GA 和 NNRGA,同理可得,时延模型权重系数 w_1 和 w_2 分别取为 0.8、2.1 和 1.8。

综上所述,本文提出的时延模型能从宏观和单个节点两方面对 NoC 时延进行优化。基于近邻随机初始种群的遗传算法和经典遗传算法相比,在运算结果和运算效率上均有提高。

4 结束语

在 2D-Mesh 拓扑结构、XY 路由算法的 NoC 实验环境下,针对 NoC 系统通信延迟难以预测的问题,本文从链路负载分布和单个节点的排队时延两方面提出了优化的 NoC 映射时延模型。同时,针对传统遗传算法效率低的问题,提出了基于近邻随机思想角度构建初始种群的算法,较传统遗传算法在运行效率上减少 20% 的时间。本文提出的时延模型与改进映射算法目前主要针对时延的单目标优化,若与能耗模型结合,也可以用来求解多目标 NoC 映射优化问题,对于 NoC 其他拓扑结构和路由算法而言,也能通过相类似的思想进行时延建模。

参考文献:

[1] HU J, MARCULESCU R. Energy-aware mapping for tile-based NoC architectures under performance constraints [C]//Pro of Asia and South Pacific Design Automation Conference. [S. l.]: IEEE, 2003: 233-239.

[2] De SILVA M V C, NEDJAH N, MOURELLE L D M. Optimal application mapping on NoC infrastructure using NSGA-Ⅱ and MicroGA [C]//Proc of the 13th IEEE International Conference on Intelligent Engineering Systems. [S. l.]: IEEE, 2009: 83-88.

[3] ARJOMAND M, SARBAZI-AZAD H, AMIRI S. Multi-objective genetic optimized multiprocessor SoC design [C]//Proc of International Symposium on System-on-Chip. [S. l.]: IEEE, 2008: 1-4.

[4] SEPULVEDA J, STRUM M, WANG J C. A multi-objective approach for multi-application NoC mapping[C]//Proc of the 2nd IEEE Latin American Symposium on Circuits and Systems. 2011: 1-4.

[5] OGRAS U Y, MARCULESCU R. Analytical router modeling for networks-on-chip performance analysis[C]//Proc of Design, Automation & Text Symposium. 2007: 1-6.

[6] ZHOU W B, ZHANG Y, MAO Z G. An application specific NoC mapping for optimized delay[C]//Proc of Design and Test of Integrated Systems in Nanoscale Technology Conference. [S. l.]: IEEE, 2006: 184-188.

[7] 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, 13(5): 937-942.

[8] MARCULESCU R, CHOU C L. Contention-aware application mapping for network on chip communication architecture [C]//Proc of IEEE International Conference on Computer Design. [S. l.]: IEEE, 2008: 164-169.

[9] BERTSEKAS D P, GALLAGER R G. Data networks [M]. [S. l.]: Prentice Hall, 1992: 1-556.

[10] DICK R, RHODES D, WOLF W. TGFF: task graphs for free [C]// Proc of Hardware/Software Codesign Conference. Washington DC: IEEE, 1998: 97-101.

[11] ORSILA H, KANGAS T, SALMINEN E, *et al*. Automated memory-aware application distribution for multi-processor system-on-chips[J]. Journal of Systems Architecture, 2007, 53(11): 795-815.