

基于改进模拟退火的 RISP 软硬件划分

朱闻博, 金同标, 殷进勇

(江苏自动化研究所, 江苏 连云港 222000)

摘要: 软硬件划分是可重构指令集处理器在软硬件协同设计中的关键问题, 已经被证明是一个 NP 难问题; 模拟退火在解决该类问题的算法中较为常用, 但在任务数变大时, 其收敛速度过慢且不一定能找到有效近似最优解, 通过将 Cauchy 分布引入扰动模型同时将其距离参数 Δy 乘上一个系数, 然后在已有代价函数的基础上提出一个更加有效的边界条件, 最后将冷却进度表的算式乘上一个权值, 以此加快算法的收敛速度; 实验结果表明, 和经典模拟退火算法相比, 新算法的收敛速度明显提高, 同时得到的解更接近最优解, 其性能优势在任务数增大时尤为明显。

关键词: 可重构指令集处理器; 软硬件划分; 模拟退火; 扰动模型; 代价函数

Hardware/Software Partitioning of RISP Based on Improved Simulated Annealing Algorithm

Zhu Wenbo, Jin Tongbiao, Yin Jinyong

(Jiangsu Automation Research Institute, Lianyungang 222000, China)

Abstract: Hardware/software partitioning is the key issue of Reconfigurable Instruction Sets Processor (RISP) in hardware/software co-design, what has been proved to be a NP problem. Simulated annealing algorithm (SA) is always the useful method to solve this kind of problems. However, the convergence speed for finding the approximately optimal solution will be slow when the count of tasks become larger. By improving the distance parameter of the perturb model using Cauchy distributing and the annealing schedule, then proposing a more effective boundary condition on the basis of the existing cost function, the speed of convergence will be accelerated. Compared to the classical simulated annealing algorithm, the final experiments indicate that using improved algorithm can significantly accelerate the convergence speed meanwhile increase the ability of finding an approximately optimal solution.

Keywords: RISP; hardware/software partitioning; simulated annealing; perturb model; cost function

0 引言

近年来, 随着硬件技术的不断发展, 可重构指令集处理器 (reconfigurable instruction sets processor, RISP) 已经被广泛地应用于高性能计算领域当中。它是将处理器核以及可重构逻辑单元相结合, 被视为下一代的处理器架构^[1]。现如今, 可重构逻辑单元多使用现场可编程门阵列 (field programmable Gate arrays, FPGA)。

软硬件划分则是可重构指令集处理器在软硬件协同设计中的关键问题, 其划分结果对于系统性能 (如系统执行时间、功耗) 等方面有决定性的影响。启发式搜索算法 (heuristic algorithm, HA) 常用以解决此类问题^[2]。其中以模拟退火算法 (simulated annealing algorithm, SA) 较为广泛, 但在问题规模变大时, 其收敛速度相对较慢, 并且不一定能找到近似最优解。

针对上述算法中存在的问题, 通过将 Cauchy 分布的扰动模型引入到模拟退火算法中且对其距离参数进行改进, 同时修改冷却进度表以及代价函数的边界条件。实验证明, 改进后算法的收敛速度大幅提高, 并且提高了找到近似最优解的能力。

收稿日期: 2014-03-28; 修回日期: 2014-04-15。

基金项目: 国家自然科学基金资助项目 (61303045); 江苏省自然科学基金资助项目 (BK2012237)。

作者简介: 朱闻博 (1989-), 男, 安徽淮北人, 硕士研究生, 主要从事可重构计算和高性能处理器方向的研究。

1 软硬件划分系统模型

1.1 系统体系结构

采用的体系结构如图 1 所示, 该体系结构包含一个通用处理器 (代表软件执行部分) 以及可重构逻辑器件 FPGA (代表硬件执行部分), 其中, 处理器被封装成 IP 核, 嵌入到可重构逻辑器件 FPGA 当中。在这种体系结构中, 硬件模块与处理器核整合在一起, 通过内部总线进行通信, 另外通过共享存储器来存储过渡数据, 并且与外部进行通信。为了研究方便, 假设硬件完全受软件控制, 同时硬件和软件之间不存在并行问题。

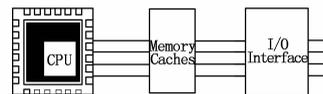


图 1 系统体系结构

1.2 系统模型

在可重构指令集处理器进行软硬件划分的过程中, 应用程序首先由 C 代码等高级语言进行描述, 然后选取适当的粒度, 之后便开始提取任务图。通常采用有向无环图 (Directed Acyclic Graph, DAG) 对任务图进行描述, 如图 2 所示。

定义 DAG 中 $D = \{V, E\}$ 。其中 V 代表图中所有节点的集合, E 表示图中所有边的集合。

有向无环图中的各个节点有 3 个重要属性, 即用一个三元

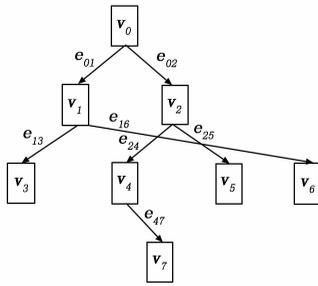


图 2 8 节点 DAG 图

组定义 $v_i = \{T_i^{sw}, T_i^{hw}, A_i\}$, 其中 T_i^{sw} 表示第 i 个任务使用软件需要的运算时间, T_i^{hw} 表示第 i 个任务使用硬件需要的运算时间, A_i 表示第 i 个任务使用硬件需要的硬件面积。

另外, 有向无环图的各个边有 2 个重要属性, 即用一个二元组定义 $e_{ij} = \{c_{ij}, t_{ij}\}$, 其中 c_{ij} 表示第 j 个节点被第 i 个节点所调用的次数, t_{ij} 表示第 i 个节点和第 j 个节点在不同划分结果下 (即一个在软件执行, 一个在硬件执行) 进行通信的所需时间。

如果用 T_i^D 表示节点 i 和其所有子节点在划分结果 D 下总的任务执行时间, 并考虑硬件不存在并行性的假设, 可得到如下的迭代公式:

$$T_i^D = t_i + \sum_{j \in V_i} c_{ij} \times T_j^D + \sum_{j \in V_i^{dif}} c_{ij} \times t_{ij} \quad (1)$$

其中当节点 i 划分到软件时, $t_i = T_i^{sw}$, 反之 $t_i = T_i^{hw}$; V_i 表示节点 i 所有子节点的集合, V_i^{dif} 表示与节点 i 划分结果不同的所有其子节点的集合。

系统总的硬件成本为:

$$A_D = \sum_{i=1}^{V_H} A_i \quad (2)$$

其中: A_D 表示划分情况 D 下总的硬件成本, V_H 表示被划分到硬件的所有节点的集合。

因此, 研究的最优划分为:

$$\min(T_0^D) \quad (3)$$

$$\text{subject to } A_D < \text{CostReq} \quad (4)$$

其中: CostReq 是由实际情况决定, 即在一定硬件成本约束下, 获取系统时间性能的最优。

2 改进模拟退火的软硬件划分算法

2.1 模拟退火算法

模拟退火最初是由 Metropolis 提出的算法, 该算法常被用来解决组合优化问题, 并且该算法已经被证明具有一定的渐近收敛性, 可以得出全局近似最优解。如下给出软硬件划分使用模拟退火算法的伪代码:

```

input: 随机产生初始划分
output: 最终划分结果
begin
给定冷却表初值
while (当前温度 > 阈值 & & 连续未接受新划分结果次数 < m)
for i = 1 to N
由扰动模型产生新的划分结果;
计算新划分与当前划分代价函数差值;
if  $\Delta < 0$  then 接受新划分为当前划分
    
```

```

else 根据接受准则决定是否接受新的划分
endif
endfor
继续降温
endwhile
end
    
```

在模拟退火得到近似最优解的过程中, 其收敛速度较慢, 目前对于模拟退火的改进主要集中在冷却进度表以及扰动模型还有代价函数改进等方面。通过改进 Cauchy 分布的参数, 同时改进算法的扰动模型和代价函数, 并且对冷却进度表进行修改使收敛速度得到改善。

2.2 改进扰动模型

在大量文献中, 模拟退火的扰动模型采用的都是随机移动, 即随机选择一个节点, 将其变更到另一个划分结果, 如果导致不满足约束条件, 则重复上述步骤, 直到产生满足约束条件的新的划分。此方法的缺点是只能在小范围内对模型进行扰动, 收敛效果不佳。因此, 文献 [3] 中提出将 Cauchy 分布作为扰动距离 y 的取值理论依据。该分布表示如下:

$$d(\Delta y) = \frac{1}{\pi} \cdot \frac{T}{T^2 + (\Delta y)^2} \quad (5)$$

可得出 $\Delta y = T \cdot \tan(\pi \cdot z)$ 。

式中, z 是随机数, 取值在 $[0, 0.5]$ 。在 DAG 图中, Δy 表示当前划分与新划分之间的距离, 即节点移至新划分后改变的节点个数。由于节点数和冷却温度在数量级上存在差异, 因此要对 Δy 进行归一化处理。可知 $\Delta y \in [0, K]$, 算法中 for 循环的时间复杂度是 $O(KN)$, 其中 $K = \text{任务数}$, $N = \text{每个温度下退火次数}$, 且 N 为定值。因此只有 K 和 for 循环的时间复杂度有关。当任务数 K 变大时, 计算时间会大幅增加。通过对 Δy 进行修正, 使 for 循环计算时间减少, 并保证 Δy 依然随 T 变化。具体方法如下:

令 $\Delta y = \sin(\omega) \cdot \Delta y$, 其中 ω 是一个很小的数。

综上所述, for 循环的复杂度变为 $O(N)$, 同时保证依旧 Δy 是 T 的函数, 在高温情况下能够进行大扰动, 并且保证可以在大范围内搜索; 在低温情况下只在当前范围内搜索, 从而加快算法收敛。产生扰动的伪代码如下:

```

begin
while ( $A_D$  不满足硬件代价的约束)
根据公式产生扰动
随机移动  $\Delta y$  个节点状态
计算硬件代价  $A_D$ 
if (不满足硬件代价  $A_D$  的约束)
back to 上一次划分
endif
end
    
```

2.3 改进代价函数

代价函数可以引导解空间的搜索方向, 目前对于解的接受存在单一和盲目性, 这也是对模拟退火算法进行改进的一个重要方向。以往代价函数规定如下:

$$\Delta = T_i - T_{i-1} \quad (6)$$

通过修改其算法的代价函数, 保证正确引导方向的前提下, 降低算法复杂度。

如图 3 所示, 假设软硬件划分的解空间是阴影部分, 横轴表示执行时间, T_{sw} 表示所有任务划分到软件后所需执行时间,

T_{hw} 表示所有任务划分到硬件后所需执行时间。纵轴表示硬件所需面积, A_D 表示硬件成本约束条件, D_1 为其中一个划分。假设其椭圆部分中有最优解, 因此该区域是算法搜索的目标。

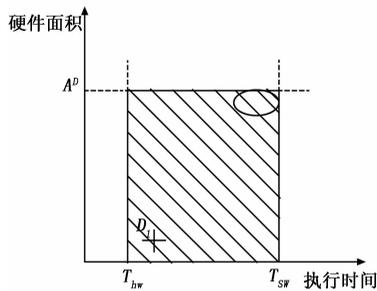


图 3 解空间示意图

以划分 D_1 为原点建立坐标系, 横轴是执行时间增量 ΔT , 纵轴是硬件面积增量 ΔA , 如图 4 所示。由于执行时间和硬件面积在数值和单位不统一, 因此先对其进行归一化处理。

由图可知, 产生的每个新划分都只会落在其中一个象限中, 假设产生的新划分为 D_2 , 即说明执行时间减少, 硬件占用面积增大。根据 Metropolis 准则, 如果新划分落在第二、三象限, 即系统时间减少, $\Delta < 0$, 则新划分以概率 1 接受; 反之新划分落在第一、四象限, 系统时间变大, $\Delta > 0$, 新划分是否被接受根据概率准则待定。

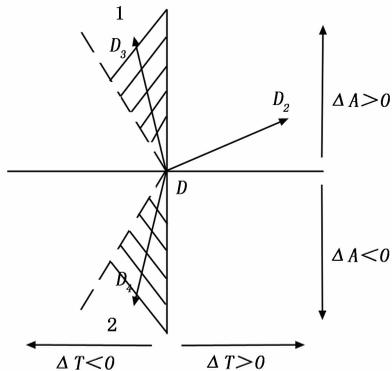


图 4 扰动点建立坐标系

在此过程中可知, 在二、三象限中有两类无效解被接受。一类如新划分 D_3 靠近正纵轴, 即新划分对减少执行时间有限的情况下反而大幅增加成本面积, 其原因在于算法只关注系统时间的减少, 而忽略成本面积的增加。另一类如新划分 D_1 靠近负纵轴, 即虽然减少硬件面积成本, 却对算法时间的减少影响甚微, 其原因在于软硬件通信时间过长。

因此设定一个边界, 如图 4 所示虚线, 即 $\Delta A = \pm \sqrt{3} \cdot \Delta T$ 。若新划分落在图中阴影部分, 对系统时间乘上一个系数 $\gamma (\gamma > 1)$, 使得 $\Delta > 0$, 从而让此情况的新划分可以被拒绝, 代价函数的伪代码如下:

```

input: 此时硬件面积  $A_i$ , 新划分执行时间  $T_i$ , 前划分执行时间  $T_{i-1}$ 
output: 代价函数差值  $\Delta$ 
begin
if  $A_i > A_D$  then
back to 前一次划分

```

```

else if 新划分位于阴影部分 then
 $T_i = \gamma \cdot T_i (\gamma > 1)$ 
else  $T_i = T_i$ 
endif
 $\Delta = T_i - T_{i-1}$ 
endif
end

```

2.4 改进冷却进度表

冷却进度表的选取对算法效率有重要影响, 在文献 [5] 中冷却进度为 $T_n = \frac{T}{n}$ 。在文献 [6] 中冷却进度为 $T_n = (\alpha)^n \cdot T$, 其中 T 表示初始温度, n 表示第 n 次降温, α 表示降温因子。由于文献 [4] 中的冷却进度下降虽快, 但可能使算法在找到近似最优解前就提前结束, 而文献 [5] 中的冷却进度虽慢, 保证算法精度, 但会使得算法收敛速度过慢, 因此通过对文献 [6] 中的冷却进度进行适当改进, 适当加快其冷却过程, 即让算式乘上一个权值 $\beta (\beta < 1)$, 以此加快冷却速度, 改进后算式如下:

$$T_n = \beta \cdot (\alpha)^n \cdot T \tag{7}$$

改进后的冷却进度介于文献 [4] 和文献 [5] 进度之间, 既能保证算法充分迭代得到近似最优解, 也能加快算法的收敛速度。

3 实验与分析

通过 TGFF 工具^[7]随机生成 DAG 图以此验证改进算法是否有效, 实验环境为 Intel Core i5-3470 3.20GHz 处理器, 3.39GB 内存, 使用 Visual Studio 2010 C++ 实现算法, 同时设定实验参数 $\omega = 0.05, \beta = 0.8$, 设定初始温度为 $T = 6000$, 硬件面积约束设定成总代价和的 0.1、0.4、0.6、0.9, 任务数分别设定为 10、40、90、160。

如图 5 所示, 即硬件面积约束为总代价和 0.9 的前提下, 算法在各个任务数下所用时间的拟合曲线。另外 SA 表示经典模拟退火算法, MySA 表示所用的改进算法。

由图 5 可知, 在任务数较小的情况下 (如任务数为 10), 两种算法的执行时间相差甚微, 但是当任务数逐渐变大时 (如任务数为 90), MySA 的收敛速度就要明显快于 SA, 当任务数达到 160 的时候, 两个算法的执行时间差异变得更加明显。

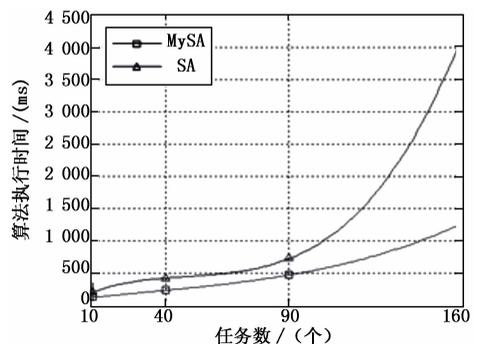


图 5 SA 和 MySA 在各个任务数下执行时间比较

表 1 表示在任务数为 160 的前提下, SA 和 MySA 两个算法在各个面积约束占比的情况下, 算法的执行时间以及相应的划分结果下系统运行所需时间。

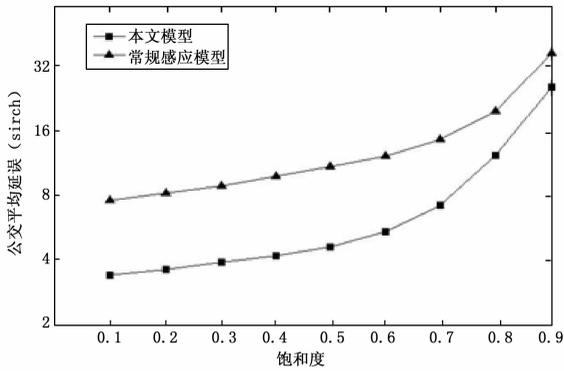


图 5 公交车辆平均延误对比图

30%以上。

5 结论

本文基于单点层制定与执行优先方案可以简化优先控制流程与优化优先算法。通过车速引导、驻站控制和信号配时以及优先请求分级错开的方式,在保证交叉口总效益最大的情况下,能使交叉口的公交平均延误时间减少幅度均高于 30%以上,有效地提高了绿灯时间利用率。

文中提出的基于单点层控制策略是一个实验原型,在今后

的研究中,笔者将结合空间优化方法和时间优化方法,同时优化道路空间资源与信号配时时间资源,后续研究应对其进行协同优化,最大程度整合时空资源,提高交通效益。

参考文献:

- [1] 高敬红, 杨宜民. 道路交通车辆检测技术及发展综述 [J]. 公路交通技术, 2012, (1): 116-119.
- [2] 凌利, 左宸, 等. 公交车辆检测技术研究进展 [J]. 现代电子技术, 2012, 21: 127-131.
- [3] 马芳. 智能公交系统中 GPS 信息质量问题解析研究 [D]. 上海: 同济大学, 2008.
- [4] 马万经, 杨晓光. 公交信号优先控制策略研究综述 [J]. 城市交通, 2010, (6): 70-78.
- [5] 邝先验, 许伦辉, 等. 交通信号公交优先控制策略及智能控制方法 [J]. 控制理论与应用, 2012, (10): 1284-1290.
- [6] NOOR M Z H, ISMAIL I, SAAID M F. Bus detection device for the blind using RFID application [A]. 2009 International Colloquium on Signal Processing & Its Applications. Kuala Lumpur [C]. Malaysia: IEEE, 2009: 247-249.
- [7] 杜云明, 周杨. 无线射频识别技术与应用研究 [J]. 自动化技术与应用, 2010, (5): 52-55.
- [8] 蔡振辉. 射频识别技术在公交站场车辆管理中的应用 [J]. 广东科技, 2010, (6): 70-71.

(上接第 2993 页)

表 1 SA 和 MySA 运行结果对比

代价约束比	算法	执行时间/ms	系统时间/ms
0.1	SA	3 310	237 468
	MySA	1 840	235 341
0.4	SA	3 420	218 567
	MySA	1310	214 236
0.6	SA	4 060	169 457
	MySA	1 330	165 583
0.9	SA	4 210	135 846
	MySA	1 270	132 194

由表 1 可知, MySA 在各种情况下的执行时间都明显比 SA 的执行时间要短(平均低于 SA 执行时间的 1/2), 证明其收敛速度要明显比 SA 快。同时, 对比最终划分结果可知, MySA 得到的最终解要优于 SA 得到的结果, 证明 MySA 找到近似最优解的能力要优于 SA。

4 结束语

针对可重构指令集处理器进行软硬件划分这一关键问题, 将 Cauchy 分布应用到经典模拟退火算法的扰动模型中, 并对其距离参数进行改进, 另外对代价函数进行改进, 给出更为有效的边界条件, 使其在执行效率和引导搜索方向上有所提高, 同时改进冷却进度表。通过对比经典模拟退火算法, 使用改进算法对算法收敛速度有明显改善, 其效果在任务数变大的情况

下更为突出, 另外改进算法也提高了搜索到全局近似最优解的能力。

参考文献:

- [1] Li Y, Callahan T, Darnell E, et al. HW/SW Co-Design of Embedded Reconfigurable Architectures [A]. Proceedings of the 37th Design Automation Conference [C]. Los Angeles: ACM, 2000: 507-512.
- [2] Noguera J, Badia R M. HW/SW Codesign Techniques for Dynamically Reconfigurable Architectures [J]. IEEE Transactions on Very Large Scale Integration Systems, 2002, 10 (4): 399-415.
- [3] 王月峰, 严卫生. 自主水下观测系统嵌入式控制器设计 [J]. 计算机测量与控制, 2013, 21 (2): 480-486.
- [4] Wu J G, Srikanthan T. Algorithmic aspects of HW/SW partitioning: 1D search algorithm [J]. IEEE Transactions on Computers, 2010, 59 (4): 532-544.
- [5] Petru E, Zebo P, Krzysztof K, et al. System Level HW/SW Partitioning Based on Simulated Annealing and Tabu Search [J]. Design Automation for Embedded Systems, 1997, 2 (5): 5-32.
- [6] Dick R P, Rhodes D L, Wolf W. TGFF: Task graphs for free [A]. Proceedings of the 6th International Workshop on Hardware/Software Co-design [C]. Seattle: ACM, 1998: 97-101.
- [7] Dave B P, Lakshminarayana G, Jha N K. COSYN: HW/SW Co-Synthesis of Heterogeneous Distributed Embedded Systems [J]. IEEE Transactions on Very Large Scale Integration Systems, 1999, 7 (1): 92-104.