

高效关键路径粒子群的软硬件划分算法

朱向华^{1,2}, 朱智林^{1,2}

(1 山东师范大学 信息科学与工程学院, 山东 济南 250014;

2 山东工商学院 信息与电子工程学院, 山东 烟台 264005)

摘 要: 软硬件划分是在满足系统约束条件下, 将系统中部分功能模块由硬件实现, 部分功能模块由软件实现, 使系统整体性能达到最优. 本文采用有向无环图对划分问题建模, 提出采用关键路径调度、自适应性的高效粒子群软硬件划分算法. 实验结果表明, 采用本文算法, 所得结果明显优于采用先来先服务的调度方法.

关键词: 软硬件划分; 关键路径; 有向无环图; 自适应; 粒子群算法

中图分类号: TP302

文献标识码: A

文章编号: 1000-7180(2013)04-0160-04

Critical Path and Efficient Particle Swarm Algorithm for Hardware/Software Partitioning

ZHU Xiang-hua^{1,2}, ZHU Zhi-lin^{1,2}

(1 College of Information Science and Engineering, Shandong Normal University, Jinan 250014, China;

2 College of Information and Electronic Engineering, Shandong Institute of Economic & Technology, Yantai 264005, China)

Abstract: Hardware/Software partitioning is the problem of which functional modules of system are implemented on hardware and which ones on software to make the system achieve optimal performance under system constraints. In this paper, a directed acyclic graph on the division of modeling, we proposed critical path scheduling, adaptive and efficient particle swarm algorithm for hardware/software partitioning. Experimental results show that the algorithms proposed in the paper can produce better solution than first-come first-served scheduling.

Key words: hardware/software partitioning; critical path; DAG; adaptive; PSO

1 引言

软硬件划分, 就是根据设计约束及系统各个部分的特点, 确定各部分的软件或者硬件实现方式, 以获得高性能、低成本的优化设计方案.

目前常见的软硬件划分方法有规划类算法和启发式算法. 如 P. V. Knudsen 等人提出采用动态规划的方法求解软硬件的划分问题^[1-2]. 一般对于小规模划分问题, 可以采用规划类算法获得最优解或近似最优解, 当划分问题规模较大时, 通常采用启发式方法来获得最优解或最优解集. 邹谊等人提出了基于遗传算法的软硬件划分算法, 在系统时间约束上没有考虑硬件模块的并行性. 刘安等提出的引入

遗传算子的划分算法复杂、运行时间长, 采用先来先服务的调度方法^[3]. 本文提出采用关键路径调度方法的粒子群划分算法, 实验表明, 本文算法优于文献^[3]的算法.

2 软硬件划分模型

在本文中, 目标系统架构模型如图 1 所示. 它有一个处理器(软件执行)和一个(或多个)硬件可编程逻辑部件(ASIC 或 FPGA), 本文在更高的抽象阶段(功能或任务级别)考虑划分和调度问题. 对于一个系统, 采用任务流图(DAG) $G = (V, E)$ 描述, 其中 $V = \{v_0, v_1, v_2, \dots, v_n, v_{n+1}\}$, 表示任务流图中所有节点的集合, 一个节点就是一个任务, 每个任务都

收稿日期: 2012-07-15; 修回日期: 2012-08-17

基金项目: 山东省自然科学基金(ZR2010FL019)

是粒度大小确定的基本调度块; $e_i \in E$ 表示任务流图中所有边的集合, $e_i = (v_i, v_j)$ 描述任务 v_i 和 v_j 之间的数据依赖或控制关系,权值表示节点之间的通信时间。

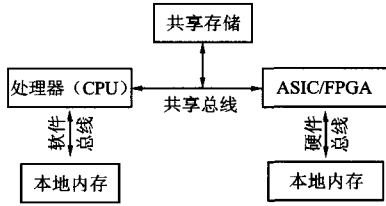


图1 目标系统架构模型

首先,对系统的时间计算进行以下说明:对于软件节点来说:① 处理器某一时间点只能处理单一任务,因此被划分到软件的节点是串行执行的。② 本地内存像任务之间的媒介,每个任务执行前从中读取数据执行后写回数据。③ 方便同硬件通信,软件的数据输出通过共享总线保存到共享存储等待硬件读取。④ 任务不能同时对存储的数据进行读或写,因此软件节点之间的通信也是串行处理的。对于硬件节点来说:① 节点之间可以并行处理。② 为防止数据丢失,数据必须被存储到硬件的本地内存中。③ 方便同软件通信,硬件的输出数据通过共享总线保存到共享存储等待软件读取。④ 硬件节点可以并行同它的前驱节点(包括软件和硬件)进行通信^[4]。

设目标划分系统由 DAG 图 $G = (V, E)$ 表示,则对任务节点 $v_i \in V$ 定义如下:

ah_i 表示节点 v_i 用硬件实现时所需要的硬件面积。

as_i 表示节点 v_i 用软件实现时所需要的软件面积。

th_i 表示节点 v_i 用硬件实现时的执行时间。

ts_i 表示节点 v_i 用软件实现时的执行时间。

ph_i 表示节点 v_i 用硬件实现时的功耗。

ps_i 表示节点 v_i 用软件实现时的功耗。

$P(v_i)$ 表示节点 v_i 所有前驱节点的集合。

$S(v_i)$ 表示节点 v_i 所有后继节点的集合。

$c(u, v_i)$ 表示节点 v_i 与节点 u 之间的通信开销。

$c(P(v_i), v_i)$ 表示节点 v_i 所有的前驱节点与节点 v_i 之间的通信开销。

$end_time(v_i)$ 表示节点 v_i 的完成执行时间。

根据系统时间计算说明,我们对 $c(P(v), v)$ 公式化定义如式(1):

$$c(P(v), v) = \begin{cases} \sum_{u \in P(v)} c(u, v), & \text{节点 } v \text{ 用软件实现} \\ \max_{u \in P(v)} \{c(u, v)\}, & \text{节点 } v \text{ 用硬件实现} \end{cases} \quad (1)$$

设 $X = (x_1, x_2, x_3, \dots, x_n)$ 是给定系统任务流图 G 的软硬件划分的一个可行解, $x_i \in \{0, 1\}$, $x_i = 1 (x_i = 0)$ 表示任务节点 v_i 由硬件(软件)实现。已知任务流图 G 的入口节点 v_{start} 和终止节点 v_{end} 都是虚节点, $end_time(v_{start}) = 0$, 则对于系统任务流图 G 的可行解 X 相应的划分系统的运行时间为 $end_time(v_{end})$ 。

定义 1 软硬件划分问题:在满足系统硬件面积约束的条件下,寻找一个最优划分方案的解或解集,使得系统的完成时间最小。系统划分问题形式化描述如式(2)所示:

$$\begin{cases} \min(end_time(v_{end})) \\ \sum_{i=1}^n ah_i x_i \leq A_{req} \\ x_i \in \{0, 1\}, i = 1, 2, \dots, n \end{cases} \quad (2)$$

3 高效离散粒子群划分算法

3.1 标准粒子群算法

粒子群算法(PSO)是一种基于群体智能随机优化算法^[5]。标准粒子群算法的第 i 个粒子的速度和位置更新由一组方程来描述:

$$V_i(t+1) = wV_i(t) + c_1 r_1 (p_i - x_i) + c_2 r_2 (p_g - x_i) \quad (3)$$

$$x_i(t+1) = x_i(t) + V_i(t+1) \quad (4)$$

式中, x_i 表示第 i 个粒子的位置, V_i 表示第 i 个粒子的速度, p_i 为第 i 个粒子的历史最优位置, p_g 为群体的历史最优位置, w 为惯性权重, c_1 和 c_2 为学习因子, r_1 和 r_2 为 $[0, 1]$ 正态分布的随机数。

为了解决离散组合优化问题, Kennedy 等又提出了二进制 PSO 算法^[5], 但是难以推广应用到离散组合优化领域。Muhammad 将自适应性引入到粒子群算法中^[7]。本文提出高效离散粒子群划分算法。

3.2 高效粒子群算法

(1) 粒子定义。每个粒子对应目标系统的一个划分方案, 粒子 i 的位置向量表示为: $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{in})$, 粒子的维数对应划分系统的任务节点数, 粒子的位置分量 x_{id} 对应任务流图中的一个任务节点, x_{id} 的取值表示节点的实现方式: 0 表示任务节点由硬件实现, 1 表示任务节点由软件实现。则粒子 i 对应的划分方案, 系统所占硬件面积 A_i 为:

$$A_i = \sum_{j=1}^n x_{ij} a_j$$

式中, a_j 表示任务 j 的硬件实现所占面积, $ph_j (ps_j)$ 分别为任务 j 硬件实现功耗(软件实现功

耗)。

(2) 算法自适应性. 自适应性机制对标准粒子群算法的优化已取得一定的研究成果, 研究表明: 在算法初期, 较大的惯性权重全局搜索能力比较强, 算法不容易陷入局部极值; 在算法末期, 较小的惯性权重收敛能力强, 使算法在可能的最优解周围精细搜索. 本文提出了具有自适应性机制的高效离散粒子群算法。

粒子群体进化过程中根据自适应性的惯性权重 w 进行速度更新, 迭代过程中的惯性权重 w 与当前迭代次数和算法最大迭代次数相关, 如公式(5)所示:

$$w_k = w_{\min} + (w_{\max} - w_{\min})k/T_{\max} \quad (5)$$

式中, w_k 表示第 k 次迭代粒子的惯性权重, $w_{\max}, w_{\min}$ 表示迭代过程中的最大(最小)变异概率, k 为当前迭代的次数, $T_{\max}$ 为最大迭代次数。

高效粒子群算法描述如下:

Step1: 初始化设置. 设置种群大小(N), 算法参数: 学习因子, 惯性权重最大值和最小值, 粒子维数, 最大迭代次数, 初始化粒子速度, 初始化粒子位置。

Step2: 对所有粒子, 初始设置 p_i 为 x_i , 初始设置 p_g 为种群中适应度值最小的 p_i 。

Step3: 每次迭代开始, 根据公式(5) 设置迭代过程中的惯性权重 w_k 的值。

Step4: 根据适应度函数求解所有粒子的适应度值。

Step5: 根据粒子的适应度值更新种群中每个粒子的 p_i 。

Step6: 根据种群中每个粒子的 p_i 更新整个种群的 p_g 。

Step7: 参照公式(3)(4), 更新每个粒子的速度和位置。

Step8: 重复 Step3 ~ Step7, 直到满足算法的迭代停止条件为止。

4 关键路径的调度算法(CPA)

任务节点的调度是软硬件划分问题中的关键环节. 软硬件划分问题, 传统上一般采用先来先服务(FCFS)的调度策略, 先来先服务的调度算法只是考虑任务节点到达的先后顺序. 在此, 考虑任务节点到达先后顺序的同时, 也注意到节点之间的通信开销对后继节点的影响. 本文提出一种基于关键路径的优先级调度策略. 根据任务节点的可执行性, 任务节点之间的调度约束如下:

(1) 根据任务节点的优先级进行调度, 优先级高的先调度, 优先级低的后调度。

(2) 只有当该任务节点的所有前驱节点都执行完后, 才能对该任务节点进行调度执行, 与任务节点的实现方式(软件实现或硬件实现) 无关。

(3) 处理器对软件节点的串行执行, 因此对于多个软件活动节点, 处理器只能调度处理一个优先级较高的软件任务节点, 待其执行完后, 处理器再调度下一个。

(4) 根据优先级调度硬件任务节点, 将它们分别交由各自的硬件模块并行处理。

对于目标系统由任务流图(DAG) $G = (V, E)$ 表示, 任务节点的优先级与其后继节点的优先级和它们之间的通信开销相关, 规定节点 $pri(v_{\text{end}}) = 0$, 其他节点优先级定义如公式(6)所示:

$$pri(u) = \begin{cases} \max_{v \in S(u)} \{ pri(v) + c(u, v) \}, & \text{节点 } u \text{ 用软件实现} \\ \max_{v \in S(u)} \{ pri(v) + c(u, v) \} + th_u, & \text{节点 } u \text{ 用硬件实现} \end{cases} \quad (6)$$

式中, $pri(u)$ 表示节点 u 的优先级; $S(u)$ 表示节点 u 的后继节点集合; $c(u, v)$ 表示节点 u 和节点 v 之间的通信时间开销; th_u 表示节点 u 的硬件实现时间。

5 实验结果及分析

图 2 中的各任务节点硬件执行时所占硬件面积, 节点序号从小到大一次对应为(4, 2, 4, 7, 5, 5, 4, 4, 3, 3, 5)。

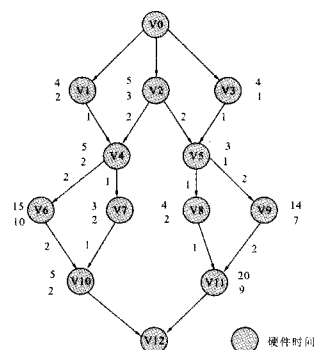


图 2 目标系统任务流程图

对于图 2 目标系统面积约束 A 为 16 的最优划分方案(0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1), 如图 3 所示. 表 1 对应该划分方案的优先级表. 采用本文关键路径优先级的调度算法调度过程如表 2 所示. 表 3 是采用文献[3]的算法(先来先服务, 对于所有软件节点, 执行时间越小的节点越早被处理器调度)。

表 1 节点优先级

	软件节点								硬件节点				结束
节点	V ₀	V ₁	V ₃	V ₄	V ₅	V ₇	V ₈	V ₁₀	V ₂	V ₆	V ₉	V ₁₁	V ₁₂
优先级	25	15	21	14	20	1	10	0	25	12	18	9	0

表 2 关键路径调度算法(CPA)

	软件节点								硬件节点			
任务节点	3	5	1	4	8	7	10		2	9	6	11
开始时间	0	7	13	17	25	30	34		0	12	14	27
通信时间	0	3	0	3	1	1	3		0	2	2	2
执行时间	4	3	4	5	4	3	5		3	7	10	9
结束时间	4	13	17	25	30	34	42		3	19	26	31

表 3 FCFS 调度算法

	软件节点								硬件节点			
任务节点	3	5	1	4	8	7	10		2	9	6	11
开始时间	0	4	8	14	19	27	39		0	14	23	27
通信时间	0	0	3	1	3	1	3		0	2	2	2
执行时间	4	4	3	4	5	3	5		3	7	9	10
结束时间	4	8	14	19	27	31	47		3	23	34	39

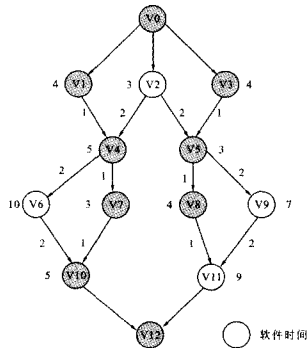


图 3 面积约束 A 为 16 的最优划分方案

在系统硬件面积约束 A 为 16 时,对于系统的同一个划分方案,采用关键路径优先级调度算法,系统的结束时间是 42,而采用文献[3]的调度策略,系统的结束时间是 47.

图 2 在关键路径调度算法(CPA)和先来先服务调度算法(FCFS)在不同面积约束下得到的划分系统的最小完成时间可由实验给出.可以看出,自适应性的粒子群算法具有更好的收敛性,更快的得到划分系统的最小完成时间,算法的执行效率更高.由随机生成节点任务数分别为 50 和 100 的 DAG 图在不同面积约束下两种调度方法所得到得最小完成时间表明,相同面积约束条件下,关键路径调度算法能够更好的得到划分系统的最小完成时间.

6 结束语

本文提出了关键路径的高效离子群软硬件划分算法,该算法具有自适应性,对惯性权重固定的粒子群算法,算法收敛性明显提高.实验结果表明,采用关键路径优先级调度算法所得目标划分系统的最小完成时间优于文献[6]的先来先服务调度算法.今后,将研究如何将本文算法应用到多处理器嵌入式系统软硬件划分问题中.

参考文献:

[1]- Knudsen P V, Madsen J PACE: A dynamic programming algorithm for hardware/software partitioning [C]//Proceeding of Internet Workshop Hardware/Software Codesign, 4th IEEE/ACM. Knudsen, Peter Voigt: IEEE, 1996: 85-92.

[2] 邹谊,庄镇泉,杨俊安,基于遗传算法的嵌入式系统软硬件划分[J]. 中国科学技术大学学报. 2004, 34(6): 724-730.

[3] 刘安,冯金富,梁晓龙,等. 基于遗传粒子群优化的嵌入式系统软硬件划分算法[J]. 计算机辅助设计与图形学报, 2010, 22(6): 927-933.

(下转第 168 页)

至服务器,提供在线直播或点播服务.在软件实现上,系统将智能手机内置摄像头采集的原始图像编码成 H.264 格式视频流,并通过 HTTP 长连接上传流媒体服务器.流媒体服务器将接收到的视频流保存成视频文件,通过视频发布模块、基于 RTMP 协议向观看者提供视频直播和视频点播服务.论文详细描述了系统实现所涉关键技术,并从视频编码和网络传输两个方面对该系统进行了性能分析.

进一步工作将对视频编码质量和网络传输性能进行优化,通过采用特定的视频编码配置,使本系统能够在网络传输不稳定的时候仍然可以观看清晰、流畅、延迟时间尚可的视频直播,进一步提升用户体验.

参考文献:

- [1] 阳永清,黄立新.基于校园网的便携式视频直播系统的实现[J].现在教育技术,2011,21(12):100-103.
- [2] 朱莹.基于 WEB 的视频直播系统的应用研究[J].电脑知识与技术,2009,5(4):968-970.
- [3] 彭凯,武娟,杨宗凯,等.基于 P2P 的流媒体直播技术研究及展望[J].计算机科学,2009,36(1):10-15.
- [4] 廖宏越,赵蓝兰,周武章.流媒体协议的研究与实现[J].计算机与数字工程,2007,35(4):106-108.
- [5] 李长洲,周曼丽.基于 Web 的流式视频直播技术[J].微型机与应用,2001,20(6):52-55.
- [6] Adobe real-time messaging protocol (RTMP) specification[EB/OL]. [2012-07-15]http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf. 2009-6.
- [7] [MS-MMSP]: Microsoft Media Server (MMS) Protocol Specification[EB/OL]. [2012-07-15]. http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/[MS-MMSP].pdf. 2012-3.
- [8] Adobe Flash Video File Format Specification[EB/OL]. [2012-07-15]. http://download.macromedia.com/f4v/video_file_format_spec_v10_1.pdf. 2010-09.
- [9] ISO/IEC 14496-10. Information technology — Coding of audio-visual objects-Part 10: Advanced Video Coding[s]. 2004-10.
- [10] ISO/IEC 14496-15. Information technology-Coding of audio-visual objects-Part 15: Advanced Video Coding (AVC) file format[s]. 2004-04.

作者简介:

杨名阳 男,(1990—),硕士生.研究方向为流媒体服务、无线网络.

张志刚 男,(1970—),硕士.研究方向为电网信息化、智能电网.

叶保留 男,(1976—),博士.副教授.研究方向为移动流媒体计算、无线网络.

(上接第 163 页)

- [4] Wangtong T, Cheung PYK, Luk W. Comparing three heuristic search methods for functional partitioning in hardware-software codesign[J]. Design Automation for Embedded Systems. 2002,6(4):25 - 449.
- [5] Kennedy J, Eberhart R. Particle swarm optimization [C]//Proceedings of IEEE Conference on Neural Networks, Australia:IEEE,1995: 1942-1948.
- [6] Kennedy J, Eberhart R. A discrete binary version of the particle swarm algorithm[C]//Proceeding of IEEE International Conference on Systems, Man, and Cybernetics. Orlando:IEEE,1997: 4104-4108.
- [7] Muhammad Ilyas MENHAS, FEI Min-rui, WANG Ling, et al. Adaptive mutation based probability binary PSO for control of ball mill pulverizing system [J]. Journal of System Simulation,2011,23(8):1568-1574.

作者简介:

朱向华 男,(1986—),硕士研究生.研究方向为软硬件协同设计.

朱智林 男,(1963—),教授.研究方向为软硬件协同设计.