

文章编号:1671-251X(2018)09-0059-08

DOI:10.13272/j.issn.1671-251x.2018020045

物联网智能感知节点 π 网低功耗软硬件划分建模

刘晓霞¹, 李芳²

(1. 四川水利职业技术学院 信息工程系, 四川 崇州 611231;

2. 重庆大学 计算机学院, 重庆 400044)

摘要:物联网智能感知节点的低功耗软硬件划分的优劣直接影响节点的续航能力和网络寿命。针对现有物联网智能感知节点的软硬件划分存在能耗较高的问题,提出了基于 π 网的物联网智能感知节点的低功耗软硬件划分模型。首先对物联网智能感知节点进行带约束定义,得到智能感知节点的约束模型;然后根据 π 网理论,建立基于 π 网的智能感知节点软硬件划分模型,实现依据软硬件IP核功耗和系统总体功耗约束下的低功耗软硬件划分,并对模型进行了演化分析。分析与实验仿真结果表明,与基于禁忌搜索算法和遗传算法的模型相比,该模型在适应度、划分执行时间和最小系统划分能耗等方面具有一定的优越性和实用性,可降低物联网智能感知节点能耗,提高其续航能力。

关键词:物联网;智能感知节点; π 网;低功耗软硬件划分;续航能力

中图分类号:TD655 文献标志码:A

网络出版地址:<http://kns.cnki.net/kcms/detail/32.1627.TP.20180823.0840.001.html>

Modeling on low power consumption hardware and software partitioning for
intelligent sensing nodes of Internet of things based on π -net

LIU Xiaoxia¹, LI Fang²

(1. Department of Information Engineering, Sichuan Water Conservancy Vocational College, Chongzhou 611231, China; 2. College of Computer Science, Chongqing University, Chongqing 400044, China)

Abstract: Advantages and disadvantages of low power consumption hardware and software partitioning for intelligent sensing nodes of Internet of Things(IoT) directly affect the endurance and network life of nodes. In view of problem of high energy consumption in hardware and software partitioning of intelligent sensor nodes of IoT, a low power consumption hardware and software partitioning model based on π -net was proposed. Firstly, the intelligent sensor nodes of IoT was defined with constraints, and the constrained model of the intelligent sensor nodes was obtained. Then, the hardware and software partitioning model of intelligent sensing nodes based on π -net was established by using the π -net theory, and the low power consumption hardware and software partitioning based on IP core power consumption of hardware and software and the overall power consumption constraints of the system were realized, and the model was analyzed for evolution. The analysis and simulation results show that the model has certain advantages and practicability in terms of fitness, execution time division and minimum system partition energy consumption compared with models based on tabu search algorithm and genetic algorithm, which

收稿日期:2018-02-27;修回日期:2018-04-28;责任编辑:张强。

基金项目:国家自然科学基金(青年科学基金)项目(61701331);四川省教育厅自然科学基金一般项目(18ZB0498);四川省水利厅2017年科研计划项目(SL2017-01)。

作者简介:刘晓霞(1976—),女,河南鄢陵人,副教授,硕士,研究方向为计算机应用技术、物联网,E-mail:903601638@qq.com。

引用格式:刘晓霞,李芳.物联网智能感知节点 π 网低功耗软硬件划分建模[J].工矿自动化,2018,44(9):59-66.

LIU Xiaoxia, LI Fang. Modeling on low power consumption hardware and software partitioning for intelligent sensing nodes of Internet of things based on π -net[J]. Industry and Mine Automation, 2018, 44(9): 59-66.

can reduce the energy consumption of intelligent sensing nodes of IoT and improve their endurance.

Key words: Internet of things; intelligent sensing node; π -net; low power consumption hardware and software partitioning; endurance

0 引言

随着物联网技术的发展和應用不断深入,智能感知节点(Intelligent Sensing Nodes, ISN)应用范围和数量不断攀升,其性能、功耗等要求越来越高。为提高 ISN 的续航能力,必须对其进行低功耗软硬件划分(Low Power Consumption Hardware and Software Partitioning, LPHSP)。在 ISN 设计和研发过程中,软硬件划分属于顶层规划与设计范畴,因此,对 ISN 进行低功耗软硬件划分是其设计和研发过程的最高层次低功耗的抽象,使其能耗达到最优化。低功耗软硬件划分对 ISN 至关重要,且有利于降低节点能耗,提高续航能力。

国内外研究者对低功耗软硬件划分算法进行了相关研究。文献[1]对双目标多选择软硬件划分进行了研究,提出了快速生成近似解的启发式算法,用定制禁忌搜索算法改进近似解算法和快速计算精确解的动态规划算法来研究软硬件划分。文献[2]对智能系统可变需求下的软硬件划分进行了研究,提出了一组增强的硬件/软件资源共享参数的划分算法。文献[3]对异构多处理器片上系统(Multi-Processor System-on-Chip, MPSoC)的软硬件划分进行了研究,提出了一种利用优化的整数线性规划算法来解决小输入的软硬件划分问题的方法。文献[4]对片上可编程门阵列作为控制器的软硬件划分进行了研究,提出了非主流排序遗传算法,并对算法进行了验证。文献[5]对复杂嵌入式系统软硬件划分进行了研究,提出了一种更加适合软硬件划分的盲优化算法,并对算法进行了验证和对比分析。文献[6]对大规模软硬件划分进行了研究,提出了一种爬山淘汰粒子群算法,并对算法进行了验证与分析。这些研究提出的低功耗软硬件划分算法、方法和策略,为研究 ISN 低功耗软硬件划分问题提供了很好的参考,但未针对物联网 ISN 进行低功耗软硬件划分研究。

π 网是一种能够处理并发事件的建模工具,融入了 π 演算和 Petri 网,是 Petri 网发展的高级阶段^[7]。鉴此,针对物联网 ISN 设计时对低功耗的要求,根据 π 网理论,建立 ISN 低功耗软硬件划分模型,并对模型进行分析和仿真实验。分析与仿真实

验结果表明,与基于禁忌搜索算法和遗传算法的模型相比,该模型在适应度和划分执行时间等方面具有一定的优越性和实用性,可降低物联网智能感知节点能耗,提高其续航能力。

1 智能感知节点约束模型

感知层作为物联网的最底层,由多种不同类型的感知设备、感知节点构成,这些设备大多为智能型设备,能够完成感知、处理和传输任务,故可抽象为 ISN,其功能需求可用任务流程图法来描述^[8]。

任务流程图法是依据待完成目标对象的要求及其主要过程,采用图表进行分析和选择实现目标所需的实际线路,描述实现顺序和过程的一种计划管理方法。智能感知节点具有一定的约束条件,特别是功耗和能耗方面的约束。为更好地描述智能感知节点,使用任务流程图来描述 ISN 划分任务。任务流程图通过有向无环图(Directed Acyclic Graph, DAG)予以描述,物联网的若干 ISN 也可以抽象为 DAG,故可用定义 1 来描述。

定义 1 若物联网 ISN 的软硬件划分问题为加权 DAG,则可定义其软硬件划分任务流程为一个五元组表,即 $PCG = \{V, E, W, F_{ec}, O_o\}$ 。

PCG 为 ISN 的加权 DAG; V 为候选的 IP 核顶点集合, $V = \{v_1, v_2, \dots, v_n\}$, 且 IP 核为具有一定功能的软件或硬件组件; E 为中顶点间有向边的集合,即 $E \subseteq V \times V, e_{ij} \rightarrow \overrightarrow{v_i v_j}, i, j \in [1, 2, \dots, n], e_{ij}$ 为顶点 v_i 到顶点 v_j 的有向边,亦表示顶点 v_i, v_j 间的依赖关系; W 为专用复杂功能组件加权集合, $W = \{P_{ro}, F_p, A_p, R_p, P_o\}$, $P_{ro}, F_p, A_p, R_p, P_o$ 分别为处理器核、FPGA 或 CPLD 核、ASIC 核、RSIC 核和其他 IP 内核; F_{ec} 为各种 IP 核功耗/能耗加权函数; O_o 为面积、执行时间等其他指标的加权函数集合。

定义 2 物联网 ISN 能耗。若 ISN 的第 i 个 IP 核的能耗主要由空闲能耗和运行能耗构成(式(1)),则 ISN 的总能耗为式(2)。

$$C_i = C_d(i) + C_r(i) \quad (1)$$

$$F_{ec} = \sum_{i=1}^n C_i(i) \leq C_g \quad (2)$$

式中: $C_d(i)$ 为第 i 个 IP 核空闲能耗; $C_r(i)$ 为第 i 个 IP 核运行能耗; C_g 为给定的功耗最大值。

由定义 1 可知,若 W 中的每个硬件处理核有本

地内存,则每个处理核上的任务间的通信即可实现。为便于进行 ISN 抽象,在此假定在 ISN 架构中的处理器数量足够,无资源竞争和临界问题,则可得到如图 1 所示的 ISN 抽象架构。

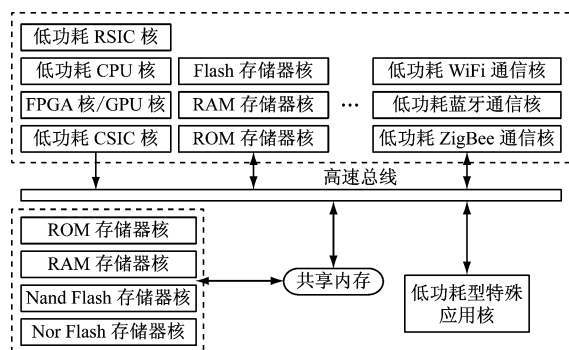


图 1 ISN 抽象架构

Fig. 1 Abstract architecture of ISN

由定义 1 可知,ISN 任务流程为 DAG 时,有向边上的数字 ω_{ij} 为顶点 v_i 到顶点 v_j 的功耗值;边 e_{ij} 除表示依赖关系外,还表示顶点 v_i 到顶点 v_j 的通信链路,此链路可能是多种总线,也可能是点到点的通信,任意 2 个顶点至少有一条通信链路直接连接。一个任务流程图中有且只有一个起始的顶点,且其入度为 0,同时,也仅有一个顶点的出度为 0,将这 2 个顶点分别称为起始顶点和终止顶点,如图 2 所示,其中 v_1 为智能节点的某种功能的软硬件划分开始, v_n 为实现该功能的软硬件划分的最优结果。

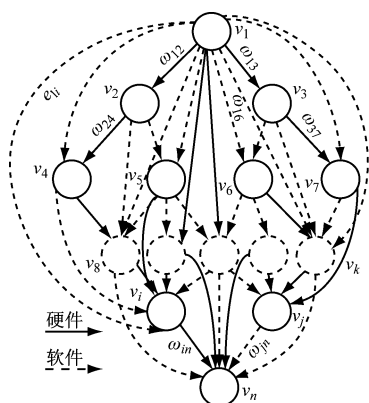


图 2 ISN 功能 IP 核任务流程

Fig. 2 IP core task flow of ISN function

由定义 1 和定义 2 可得,用任务流程图来描述 ISN 的任务划分时,在功耗等约束下,可寻求 ISN 在功耗和其他指标的最优解。图 2 即为某个 ISN 在若干划分任务下的最优解。

依据定义 1 和定义 2,可得到 ISN 软硬件划分问题的定义,即定义 3。

定义 3 物联网 ISN 软硬件划分问题定义。ISN 软硬件划分问题可定义如下:给定一个 $PCG =$

$\{V, E, W, F_{ec}, O_o\}$, 在功耗/能耗和其他指标约束下,求解每个 $v \in V$ 的任务在给定的最小功耗 $\min(F_{ec}(PCG))$ 下的划分 $P(v)$,使得式(3)成立,则 $P(v)$ 为给定 $PCG = \{V, E, W, F_{ec}, O_o\}$ 的一个最优划分。 $P(v)$ 为给定的软硬件划分函数,即 $P: v \rightarrow W$, 且 $v \in V$, 是 W 中由硬件或软件实现的任务,若 $P(v_i) = 1$, 表示节点 v 由硬件实现,同样地, $P(v_i) = 0$, 表示节点 v 由软件实现; $\min(F_{ec}(PCG)) \leq F_{ec}(PCG) \leq C_g$ 为描述的任务流程图的功耗/能耗不等式; $O_o(PCG) \leq O_d$ 为其他约束项加权不等式, O_d 为给定的加权值。

$$\begin{cases} \min(F_{ec}(PCG)) \leq F_{ec}(PCG) \leq C_g \\ O_o(PCG) \leq O_d \end{cases} \quad (3)$$

2 基于 π 网的低功耗软硬件划分模型

依据 π 网理论,可有效解决并发性的建模问题。ISN 的低功耗软硬件划分问题,存在多种要求、多种约束的硬件、软件可选择组件,是一个典型的并发性问题。本文通过建立基于 π 网的低功耗软硬件划分模型来解决 ISN 的软硬件划分问题,并对模型进行动态演化分析。

2.1 模型建立

根据 π 网理论,建立基于 π 网的物联网 ISN 的低功耗软硬件划分 LPHSP^[9] 模型,即 N_{LPHSP} 。

定义 4 物联网 ISN 的 LPHSP 模型为一个 π 网 N_{LPHSP} , $N_{LPHSP} = (A_f, S_{IPo}, D_{arc}, T_r)$, A_f 为设计中的软硬件 IP 核实现的功能集合,是模型 N_{LPHSP} 的库所集; S_{IPo} 为系统设计中软硬件 IP 核对应的功耗/能耗和其他约束项的集合,是模型 N_{LPHSP} 的变迁集,使得 $A_f \cap S_{IPo} = \phi$; $D_{arc} \subseteq (A_f \times S_{IPo}) \cup (S_{IPo} \times A_f)$, 是模型 N_{LPHSP} 的有向弧集; T_r 是定义在 $A_f \cup S_{IPo} \cup D_{arc}$ 上的一个属性函数,表示物联网 ISN 设计中的各种约束评价因子所构成的函数关系。

由定义 4 可知,物联网 ISN 软硬件划分就是在 IP 核集合上做最优划分,该划分保证在 D_{arc} 和 T_r 条件下代价最小。

定义 5 约束评价因子函数 T_r 为一个八元组,用来表示各种评价因子间的相互关系,即

$$T_r = (C_L, T_r, Q, F_C, F_D, ID, T_i, B) \quad (4)$$

式中: $C_L = \{c_{l_1}, c_{l_2}, \dots, c_{l_n}\}$, 是有限的库所集,在此表示各种 IP 核的功耗/能耗和其他约束项的集合,即 $c_{l_i} = S_{IPi} \cup S_{IPoi}$, 其中 S_{IPi} , S_{IPoi} 分别为第 i 个 IP 核的功耗/能耗和其他约束项; $T_r = \{T_{r_1}, T_{r_2}, \dots, T_{r_n}\}$, 为有限迁移集合,且 $C_L \cup T_r \neq \phi$, $C_L \cap T_r = \phi$,

表示由硬件 IP 核库所到软件 IP 核的变迁集合,反映 ISN 的动态迁移; $Q=\{q_1, q_2, \dots, q_n\}$, 是有限的数据变迁集, q_i 为第 i 个数据变迁, 且 $C_L \cap T_r = \phi$, $C_L \cap Q = \phi$, $Q \cap T_r = \phi$, 在此表示各 IP 核的成本; $F_C \subseteq (C_L \times T_r) \cup (T_r \times C_L) \cup (C_L \times Q)$, 是控制流的弧的集合; $F_D \subseteq (C_L \times Q) \cup (Q \times C_L) \cup (C_L \times T_r)$, 是数据流的弧的集合, 表示硬件、软件 IP 核间的动态关系; $ID: C_L \rightarrow X$ (X 为非负整数集合), 是位置集合上的标识向量, 表示各种约束因子的优先级别; $T_i: T_r \rightarrow T_r'$, 为迁移所需的时间, 表示模型 N_{LPHSP} 迁移为下一个变迁后的 π 网模型 N'_{LPHSP} 所需的时间; B 表示其他评价因子。

定义 4、定义 5 定义了物联网 ISN 软硬件划分 LPHSP 模型, 其动态迁移规则参见文献[7]。

由此, 可得到模型 N_{LPHSP} 的软硬件划分算法步骤如下:

第 1 步: 输入、输出和约束条件设置。输入设置: ISN-DAG 及模型 N_{LPHSP} 的初始条件、候选 IP 核库所、约束条件; 输出设置: 符合约束条件的划分集合, 并进行最优化处理; 约束条件设置: 按照式(3)设定。

第 2 步: 初始网。按照输入的 ISN-DAG 及约束, 得到初始 π 网。

第 3 步: 初始网演算。依据 π 网理论, 在初始输入设置下进行初始 π 网演算。

第 4 步: 子网划分。对初始 π 网进行变迁的使能与激发, 得到中间 π 网 N'_{LPHSP} ; 依据约束条件进行库所、变迁的删减, 得到最小网 N''_{LPHSP} ; 在 N''_{LPHSP} 中, 计算变迁的最小时间及其对应的子网, 直到得到

子网 N'''_{LPHSP} 中没有被 2 个变迁同时覆盖的变迁, 否则继续进行子网划分。

第 5 步: 得到 N_{LPHSP} 模型。经过第 3、4 步, 在约束下, 对得到的子网 N'''_{LPHSP} 进行评估, 若符合约束条件, 则得到模型 N_{LPHSP} , 否则返回第 3 步。

第 6 步: 模型 N_{LPHSP} 的效用评估。对模型 N_{LPHSP} 进行效用度计算, 并将其与给定的约束条件中的效用度值进行比较, 若其值小于约束中的效用度值, 则返回第 1 步。

第 7 步: 输出划分结果。符合约束条件的划分, 输出划分结果, 启用最优化处理。

第 8 步: 得到最优的划分结果。从所有符合约束条件的划分输出中, 采用多目标最优化处理方法, 得到最优的结果, 并以此作为物联网 ISN 的软硬件划分的最后结果输出; 否则返回到第 1 步。

按照以上步骤, 得到物联网 ISN 软硬件划分模型 N_{LPHSP} , 如图 3 所示。图 3 中, 左边为模型 N_{LPHSP} 的库所、变迁在约束条件下进行软硬件划分的过程; 右边为模型 N_{LPHSP} 进行划分之后, 得到选中的可选 IP 核构成的 ISN 模型。图 3 中, $Data_i$ 为第 i 个库所的功耗/能耗等约束条件集合; E_i 表示 A_i 软硬件 IP 核实现的功能集合, 是模型 N_{LPHSP} 的库所集; t_i 为变迁集合, 是模型 N_{LPHSP} 中的 S_{IP_0} , 表示划分中的软硬件 IP 核对应功耗/能耗和其他约束项的集合; FM_i 为划分后得到的功能模块; C_{ji} 为第 i 个功能模块进行软硬件划分时得到的多种 IP 核集合; e_1, e_2 表示对划分效用评估后重新返回模型 N_{LPHSP} 的某个步骤。

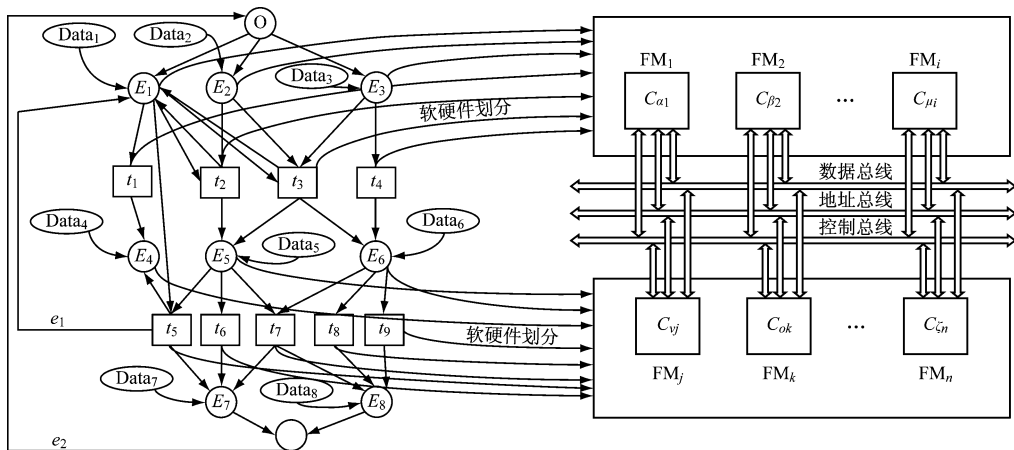


图 3 模型 N_{LPHSP} 的划分结果与功能模块的映射关系

Fig. 3 Partitioning results between N_{LPHSP} model and mapping relation of functional modules

2.2 模型动态演化分析

应用 π 网理论, 得到物联网 ISN 的 LPHSP 模

型 N_{LPHSP} , 还需利用 π 网理论对 N_{LPHSP} 模型进行动态演化分析。动态演化即为模型 N_{LPHSP} 增加新的库

所或服务组件时,模型 N_{LPHSP} 仍然能够保持良好的性能;若模型 N_{LPHSP} 中某个库所发生了更新,或者被新的组件取代,则为更新演化。在此主要对模型 N_{LPHSP} 进行库所增加和更新的演化分析。

若模型 N_{LPHSP} 的库所 A_i 有新库所 s 加入,可用如下进程进行描述,即进程 NewPlace 可描述为

$$\text{newPCreate}(i_d, s) = \text{PCreate}(i_d, s) \quad (5)$$

$$\text{regPlace}(i_d, s) = (i_d, s)(\text{reg}(i_d, s)) \quad (6)$$

$$\text{conregplace}(x, y) = (x, y)(\text{reg}(x, y)) \quad (7)$$

式(5)为库所创建进程,即创建了一个标志符为 i_d 、服务为 s 的候选库所;式(6)为库所注册进程,即库所通过通道 reg 传递其标志符 i_d 和服务 s 到候选库中;式(7)为库所在候选库中的注册进程,即库所通过通道 reg 获取新库所的注册信息 x 和 y ,同时更新其知识库 K_p 。只有式(5)一式(7)同时成立时,才能进入模型。

模型 N_{LPHSP} 的库所 A_i 有新库所 s 加入后,需要参与模型 N_{LPHSP} 的各种变迁,可用 NewPlaceP 进程来描述变迁的加入,即

$$\text{ReqserPro}(i, r, l) = \bar{i} \langle a \rangle r(z) \bar{z} \langle l \rangle \quad (8)$$

$$\text{QueserPro}(i, r, p) = i(y) \langle [y = a] r \langle p \rangle + [y \neq s] \overline{\text{subs}} \langle y \rangle \rangle \quad (9)$$

$$\text{DelivserPro}(p, s) = p(x) \bar{x} \langle s \rangle \quad (10)$$

式(8)为请求服务的进程,通道新候选 IP 核通过通道 i 发送请求 a 到库所,然后通过通道 r 等待库所回复,一旦收到应答标志 z ,就通过 z 发送请求服务地址 l 到库所。式(9)为连接件中相应的服务查询进程, p 为进程;式(10)为服务提供进程。只有式(8)一式(10)同时成立时,才能进入模型。

模型 N_{LPHSP} 的库所 A_i 有新库所 s 加入后, s 为不完备时,则用 π 网的弱时间等价关系可将式(9)改写为

$$\text{QueserPro}(i, r, p, R_i) = i(y) \langle [y = a] r \langle p \rangle + [y \neq s] \overline{\text{subs}} \langle y \rangle \rangle Q_i \quad (11)$$

式中 R_i 是在定义 5 的八元组上对时间的抽象与进程 Q_i 的弱时间等价。

由此,模型 N_{LPHSP} 可进行库所添加,而且加入新库所后,新库所能够和与之适应的服务组件进行交互,从而参与模型 N_{LPHSP} 的变迁。

若模型 N_{LPHSP} 的库所 A_i 要进行变化或升级,则需对原库所进行语义更新和扩展更新^[10]。对模型 N_{LPHSP} 的库所 A_i 进行语义更新时,其更新规则为规则 1。

规则 1 设候选 IP 核 $C_{p,0}$ 的接口为 Ψ , 组件

$C_{p,1}$ 的接口为 Ω , 若 $\Psi \approx \Omega$ 成立,并 $\exists \beta$ 和 M , 有 $\beta\Psi + M \approx \beta\Omega + M$ 成立,则称 $C_{p,1}$ 可使 $C_{p,0}$ 进行保持语义更新。

若模型 N_{LPHSP} 的库所 A_i 需进行扩展更新,利用规则 2 对原有库所进行扩展更新。

规则 2 设候选 IP 核 $C_{p,0}$ 的接口为 Γ , $C_{p,1}$ 的接口为 Π , 若 Γ 和 Π 使以下条件同时成立:① $f_n(\Gamma) \subseteq f_n(\Pi)$, 即 Γ 的自由名字是 Π 自由名字的子集;

② 如果 $\tau\Gamma \rightarrow \Gamma'$, 那么 $\exists \Pi', \Pi \Rightarrow \Pi'$; ③ 如果 $\Gamma \xrightarrow{x(z)} \Gamma'$, 那么 $\exists \Pi', \Pi \xrightarrow{x(z) \dots x_i(z_i)} \Pi'$; ④ 如果 $\Gamma \xrightarrow{x < y >} \Gamma'$,

那么 $\exists \Pi', \Pi \xrightarrow{x < y > \dots x_i < y_i >} \Pi'$, 则进程 Π 对 Γ 进行了扩展性更新,标记为 $\Gamma < \Pi$ 。故 $C_{p,1}$ 对 $C_{p,0}$ 进行了扩展性更新。

综上所述可知,当模型 N_{LPHSP} 有新的 IP 核加入时,经过模型 N_{LPHSP} 的库所创建进程、库所注册进程和库所在候选库中的注册进程而融入模型 N_{LPHSP} , 成为模型 N_{LPHSP} 的一部分。因此,模型 N_{LPHSP} 能够对库所 A_i 进行增加和更新,即模型 N_{LPHSP} 能够进行动态增加和更新演化。

3 实验与仿真

以具体物联网 ISN 为例,对模型 N_{LPHSP} 进行验证与优化^[11]。

以温度湿度感知采集节点为例,用模型 N_{LPHSP} 对其进行描述,并按照模型步骤进行低功耗软硬件划分。候选 IP 核在此以硬件核和软件核来区分,并按照功能模块予以分类描述,即模型 N_{LPHSP} 的库所由硬件库所和软件库所构成,且库所又按照功能来区别。多参数感知采集节点的库所 $A_i = \{A_{HW}, A_{SW}\}$, 其中 A_{HW} 为硬件 IP 核,且 $A_{HW} = \{\text{Proc}, \text{Mem}, \text{Com}, \text{Sen}, \dots\}$, $\text{Proc}, \text{Mem}, \text{Com}, \text{Sen}$ 分别为候选的处理器核、存储器核、通信核、传感器核,而每个候选硬件核又对应多个选项,如 Proc 可为 MCS-51 核、ARM7TDMI 核、moto68K 核和 ARM Cortex Mx 核等, Sen 为按照感知信号的不同进行分类的候选传感器核集合,比如温度传感器核可为 DHT11、金属热电阻、热电偶、DS18B20、铜电阻和热敏电阻等; A_{SW} 为软件 IP 核,即能够实现一定功能的软件核,如模块初始化程序、板级驱动程序、操作系统、网络协议和应用软件模型等。模型 N_{LPHSP} 的变迁为 S_{IP_0} , S_{IP_0} 为库所 IP 核对应的功耗和其他性能指标约束集合。由此,温度湿度感知采集节点 N_{LPHSP} 模型划分如图 4 所示。

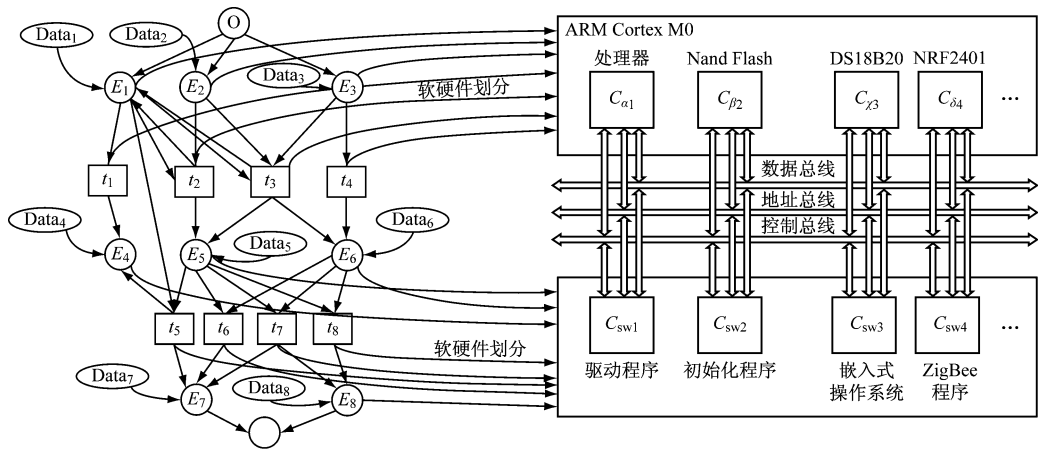


图 4 温度湿度感知采集节点 N_{LPHSP} 模型划分

Fig. 4 Partitioning of N_{LPHSP} model of temperature and humidity sensing acquisition node

实验测试时,ZigBee 模块选用 F8913D,WiFi 选用 CC2540,蓝牙采用 RKi6000,处理器使用 Cortex M 系列,软件 IP 核采用标准 C 语言编写的功能核。测试时采用运行测试和源代码级程序分析相结合的方式,综合得到实验数据。在测试过程中,每种算法使用相同的面积、体积和其他参数,具体实验得到的数据见表 1。

利用表 1 数据作为属性 DAG-PCG,并输入 N_{LPHSP} 模型、基于遗传算法^[12-13]的模型、基于蚂蚁优化算法的模型^[14]和基于禁忌搜索算法^[15-16]的模型,利用 Matlab 编程仿真每个模型的适应度、划分运行时间和最小系统划分能耗,得到如图 5 所示的比较曲线。

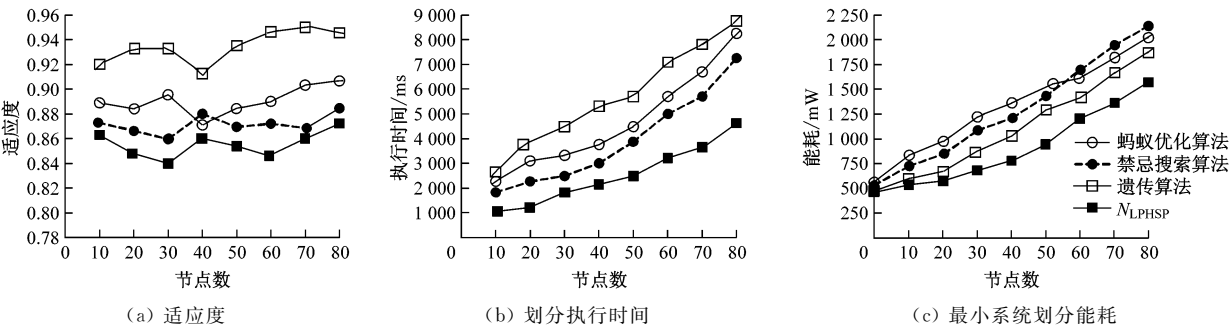


图 5 模型 N_{LPHSP} 与其他算法模型的适应度、执行时间和最小系统划分能耗的比较曲线

Fig. 5 Comparison curves of N_{LPHSP} model and other algorithms model in fitness, partitioning execution time and minimum system partition energy consumption

本实验在一定的约束条件下优化其软硬件划分。从图 5(a)可看出,与其他算法相比,当候选 IP 核增加时,模型 N_{LPHSP} 的适应度曲线总的趋势是在增加,但适应度值较其他算法小,说明更加容易适应候选 IP 核数量的增加。从图 5(b)可看出,在相同候选 IP 核下,模型 N_{LPHSP} 软硬件划分执行时间比其他算法少,说明其运行速度比较快。从图 5(c)可看出,随着候选 IP 核数量的增加,各模型执行时能耗

均在增加,但模型 N_{LPHSP} 的能耗增加量比较小,说明模型 N_{LPHSP} 在能量消耗方面具有优越性。

实验及仿真结果表明,与基于蚂蚁优化算法、禁忌搜索算法和遗传算法的模型相比,模型 N_{LPHSP} 在适应度、划分执行时间和最小系统划分能耗等方面具有一定的优越性和实用性^[17],可降低物联网 ISN 能耗,提高其续航能力。

表1 候选IP核参数及其实验数据

Table 1 Parameters of candidate IP core and its experimental data

功能模块	IP核	运行时功耗/能耗		其他指标		
				执行时间/ μ s	适应度	可用性
处理器	TI MSP432	95 μ A/MHz		—	0.999 7	0.999 1
	STM32L476	100 μ A/MHz		—	0.999 6	0.999 3
	Atmel SAML21	35 mA/MHz		—	0.999 2	0.999 2
	Cortex M0	35 mA/MHz		—	0.999 9	0.999 5
	⋮	⋮		⋮	⋮	⋮
存储器	ROM	150 μ A		10~20	0.999 9	0.999 6
	RAM	170 μ A		10~20	0.999 8	0.999 5
	Nor Flash	9 mA		10~20	0.999 3	0.999 4
	Nand Flash	9 mA		10~20	0.999 4	0.999 6
传感器	DHT11	3 mW		15	0.999 5	0.999 7
	DS18B20	2.6 mW		21	0.999 4	0.999 9
	湿度传感器	1.8 mW		32	0.999 2	0.999 5
通信模块	WiFi 模块	发送	72 mW	—	0.999 1	0.992 6
		接收	58.8 mW			
	ZigBee 模块	发送	96.1 mW	—	0.999 5	0.993 2
		接收	91.7 mW			
	蓝牙	发送	60 mW	—	0.999 2	0.994 5
软件IP核		接收	60 mW			
	C_{SW1}	113 μ W		637	0.995 9	0.934 3
	C_{SW2}	8 μ W		422	0.954 1	0.943 4
	C_{SW3}	21 μ W		483	0.921 3	0.920 1
	C_{SW4}	231 μ W		749	0.956 2	0.957 1
	C_{SW5}	251 μ W		698	0.934 2	0.934 1
	C_{SW6}	342 μ W		972	0.967 1	0.967 3
	C_{SW7}	207 μ W		653	0.957 3	0.931 4
	C_{SW8}	239 μ W		549	0.923 5	0.936 7
	C_{SW9}	213 μ W		497	0.991 2	0.923 5
⋮	⋮	⋮		⋮	⋮	⋮

4 结语

物联网 ISN 的低功耗软硬件划分的优劣直接影响节点的续航能力和网络寿命。首先对物联网 ISN 进行带约束定义,得到了 ISN 的约束模型;然后根据 π 网理论,建立了基于 π 网的物联网 ISN 软硬件低功耗划分模型,实现了依据软硬件 IP 核功耗和系统总体功耗约束下的低功耗软硬件划分,并对模型进行了演化分析;最后,与基于禁忌搜索算法和遗传算法等的模型进行了对比实验。分析与实验仿真结果表明,与基于禁忌搜索算法和遗传算法等的

模型相比,该模型在适应度、划分执行时间和最小系统划分能耗等方面具有一定的优越性和实用性,可降低物联网 ISN 能耗,提高其续航能力。

参考文献(References):

[1] SHI W J, WU J G, LAM S K, et al. Algorithms for bi-objective multiple-choice hardware/software partitioning [J]. Computers & Electrical Engineering, 2016, 50(3): 127-142.

[2] WANG R, HUNG W, YANG G, et al. Uncertainty model for configurable hardware/software and resource partitioning [J]. IEEE Transactions on

- Computers, 2016, 65(10): 3217-3223.
- [3] OUYANG Aijia, PENG Xuyu, LIU Jing, et al. Hardware/software partitioning for heterogenous MPSoC considering communication overhead [J]. International Journal of Parallel Programming, 2017, 45(4): 1-24.
- [4] BAHRI I, IDKHAJINE L, MONMASSON E, et al. Original articles: optimal hardware/software partitioning of a system on chip FPGA-based sensorless AC drive current controller [J]. Mathematics & Computers in Simulation, 2013, 90(2): 145-161.
- [5] ZHANG Tao, ZHAO Xin, AN Xinqi, et al. Using blind optimization algorithm for hardware/software partitioning[J]. IEEE Access, 2017, 5(99): 1353-1362.
- [6] 鄢小虎, 何发智, 陈壹林. 求解大规模软硬件划分问题的爬山淘汰粒子群算法[J]. 东南大学学报(自然科学版), 2017, 47(2): 225-230.
- YAN Xiaohu, HE Fazhi, CHEN Yilin. Elimination particle swarm optimization with hill climbing algorithm for solving large-scale hardware / software partitioning problem [J]. Journal of Southeast University (Natural Science Edition), 2017, 47(2): 225-230.
- [7] 曹木亮, 吴智铭, 杨根科. 一类新型的模块化高级 Petri 网—— π 网[J]. 上海交通大学学报, 2004, 38(1): 52-58.
- CAO Muliang, WU Zhiming, YANG Genke. π -nets: a new modular high level Petri net [J]. Journal of Shanghai Jiaotong University, 2004, 38(1): 52-58.
- [8] SHA E, LI W, ZHUGE Q F, et al. Power efficiency for hardware/software partitioning with time and area constraints on MPSoC [J]. International Journal of Parallel Programming, 2015, 43(3): 381-402.
- [9] RIABI M, MANAI Y, HAGGÈGE J. Hardware/software partitioning approach for embedded system design based on genetic algorithm [J]. International Journal of Scientific Research & Engineering Technology, 2015, 3(3): 20-25.
- [10] TALEB M, LECLERCQ E, LEFEBVRE D. Model predictive control for discrete and continuous timed Petri nets[J]. International Journal of Automation & Computing, 2018(2): 1-14.
- [11] CHAKRABORTY D, GUHA D, DUTTA B. Multi-objective optimization problem under fuzzy rule constraints using particle swarm optimization[J]. Soft Computing, 2016, 20(6): 2245-2259.
- [12] 侯能, 何发智. 混合并行两步调整遗传策略的软硬件划分算法[J]. 华中科技大学学报(自然科学版), 2017, 45(12): 39-45.
- HOU Neng, HE Fazhi. Hybrid parallel genetic strategy with two-step adjustment for HW/SW partitioning [J]. Journal of Huazhong University of Science & Technology (Natural Science Edition), 2017, 45(12): 39-45.
- [13] 朱峰军. 多选择软硬件划分问题的算法研究[D]. 天津: 天津工业大学, 2014.
- [14] 潘颖, 阮文惠. 基于改进蚁群算法的嵌入式系统软硬件划分[J]. 现代电子技术, 2017, 40(3): 164-166.
- PAN Ying, RUAN Wenhui. Embedded system hardware and software partition based on improved ant colony algorithm [J]. Modern Electronics Technique, 2017, 40(3): 164-166.
- [15] 高健, 李涛. 三种软硬件划分算法的比较分析[J]. 计算机工程与设计, 2007, 28(14): 3426-3428.
- GAO Jian, LI Tao. Comparison and analysis of three hardware/software partitioning algorithms [J]. Computer Engineering and Design, 2007, 28(14): 3426-3428.
- [16] 周立秋. 可重构系统中的一种动态软硬件划分算法[D]. 长沙: 湖南大学, 2009.
- [17] 焦素敏, 王彩红, 王学梅. 应用于大规模 SoC 软硬件划分的多种群协同进化方法研究[J]. 计算机应用研究, 2011, 28(10): 3690-3693.
- JIAO Sumin, WANG Caihong, WANG Xuemei. Multi-group co-evolutionary method research: applied in large scale SoC hardware/software partitioning [J]. Application Research of Computers, 2011, 28(10): 3690-3693.