软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(2):419−438 [doi: 10.13328/j.cnki.jos.004536]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

分布式嵌入式系统的自适应能耗管理和分析*

范贵生[1,2], 虞慧群[1], 陈丽琼[3], 刘冬梅[1]

[1](华东理工大学 计算机科学与工程系,上海 200237)

[2](计算机软件新技术国家重点实验室(南京大学),江苏 南京 210093)

[3](上海应用技术学院 计算机科学与信息工程系,上海 200235)

通讯作者: 陈丽琼, E-mail: lqchen@sit.edu.cn

摘 要: 以降低分布式嵌入式系统整体能耗为目标,立足设备属性及其关系,从系统的启动设备集和设备动态供电电压两个方面着手,提出一种基于 Agent 的自适应能耗管理及其分析方法.在此基础上,给出分布式嵌入式能耗网(DE-Net),并利用 DE-Net 模型对系统的基本组件进行建模,根据组件间关系形成能耗模型,以刻画系统的执行流程和能耗属性.最后,利用 CTL 描述系统性质,并借助 Petri 网的操作语义来验证方法的正确性和有效性.具体实例应用及实验结果表明:该方法能够有效地降低分布式实时系统的能耗,正确描述能耗自适应调整过程,简化建模和分析过程,对开发具有低能耗 DES 具有重要的理论意义和实用价值.

关键词: 分布式嵌入式系统;多 Agent;能耗;自适应;Petri 网

中图法分类号: TP316 **文献标识码**: A

中文引用格式: 范贵生,虞慧群,陈丽琼,刘冬梅.分布式嵌入式系统的自适应能耗管理和分析.软件学报,2014,25(2):419−438. http://www.jos.org.cn/1000-9825/4536.htm

英文引用格式: Fan GS, Yu HQ, Chen LQ, Liu DM. Adaptive energy consumption management and analysis for distributed embedded system. Ruan Jian Xue Bao/Journal of Software, 2014,25(2):419−438 (in Chinese). http://www.jos.org.cn/1000-9825/4536.htm

Adaptive Energy Consumption Management and Analysis for Distributed Embedded System

FAN Gui-Sheng[1,2], YU Hui-Qun[1], CHEN Li-Qiong[3], LIU Dong-Mei[1]

[1](Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

[2](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210093, China)

[3](Department of Computer Science and Information Engineering Shanghai Institute of Technology, Shanghai 200235, China)

Corresponding author: Chen Li-Qiong, E-mail: lqchen@sit.edu.cn

Abstract: In order to reduce the overall energy consumption of the distributed embedded system(DES), this paper proposes an adaptive energy management and analysis method based on the attributes of devices and their relationships by considering the boot device set and the dynamic supply voltage of devices. The study results in the distributed embedded energy consumption net (DE-Net) which is used to model the basic components of the embedded system. The model of energy consumption is then formed to characterize the execution process and the attributes of energy consumption. Finally, CTL is used to express the basic properties of DES. The operational semantics of Petri nets helps verify the correctness and effectiveness of the proposed method. The example and experimental results show that this method can correctly describe the adaptive energy consumption of DES and simplify the modeling and analysis process, which has important theoretical significance and practical value for developing the low energy consumption of DES.

Key words: distributed embedded system; multi-agent; energy consumption; adaptive; Petri net

* 基金项目: 国家自然科学基金(61173048, 61300041); 高等学校博士学科点专项科研基金(20130074110015); 上海市教育委员会科研创新项目(12YZ166); 中央高校基本科研业务费专项基金(WH1314038); 上海市科委重大项目(12510503800)

收稿时间: 2013-05-07; 修改时间: 2013-09-29; 定稿时间: 2013-12-05

© 中国科学院软件研究所 http://www.jos.org.cn

随着网络技术的迅速发展,基于嵌入式的应用已经广泛应用于交通、能源、汽车、航天航空、医疗保健等领域.分布式嵌入式系统(distributed embedded system,简称 DES)采用分布式体系结构,通过各种无线传感设备,把物体与互联网相连接,进行信息交换和通信,以达到智能化管理的目的.然而,DES 复杂结构所带来的能耗成为限制其发展的瓶颈,使得 DES 的能耗问题的重要性呈现逐步上升趋势.特别是在电池供电的嵌入式系统中,由于其自身特点的限制,其发展也非常依赖低功耗、高能效技术的研究与应用.鉴于 DES 应用的社会效益和经济价值,如何设计和分析 DES 的能耗问题,已成为软件工程领域具有挑战性的研究课题[1].

DES 自适应能耗管理是根据 DES 中各设备的能耗数据,利用公式或理论模型来推演设备的能耗状况,进一步获得整个软件的动态能耗模型,并制定能耗策略.其目标是:在系统运行过程中,实时分析能耗信息,并根据预先设定好的策略对设备状态和属性进行动态的调整,以更好地满足系统节能的需求.在硬件给定的情况下,通过软件技术降低系统能耗的问题得到广泛重视,开发低能耗软件成为系统设计者的重要目标之一[2].然而,DES 的结构类型多种多样,行为特征纷繁复杂,对自适应能耗管理与分析提出了新的挑战:

- 首先,实现 DES 的硬件平台丰富多样、应用领域及需求千变万化,对软件设计提出了新的挑战.硬件平台的多样性极大地影响到软件系统的设计方法,从基本组件抽象到组件之间的调度、通信和协调,都需要新的语言来进行描述.用于描述 DES 结构的建模语言必须具备系统功能性质的规范和分析能力,以及动态能耗调整和表达能力.此外,DES 中计算成分和物理实体结构、行为和交互关系复杂,需要有新的技术建立 DES 结构、成分的抽象以及组合关系,以统一的方式提供成分建模和模型组合的手段,提高能耗模型的可扩展性.
- 其次,DES 构成组件复杂多样,传感器、处理器、执行器等组件具有确定性与随机性共在的能耗特征,静态分析和优化技术难以适应 DES 的发展.DES 部署环境可能发生动态改变,导致组件的位置、执行操作等属性存在随机性.因此,需要提出新的 DES 能耗策略,以满足系统运行时限和功能性需求为前提,根据当前系统组件部署情况和属性,自适应地启动和休眠设备,动态调整设备的供电电压以降低系统的整体能耗.
- 最后,DES 动态行为特征复杂,为其能耗模型的性质验证带来困难.DES 具有复杂能耗和空间特征、多层次的结构特征,因此需要研究新的验证方法,利用能耗模型的执行语义和实际映射,立足自适应能耗策略的目标和范畴,对系统正确性进行验证和分析,以保障 DES 能耗建模和分析的有效性.

为了解决 DES 的自适应能耗管理与分析的问题,传统的能耗管理方法需要扩充新成分才能适应低能耗的要求,以得到比较合理的节能目标.多 Agent 系统(MAS)由多个自主或半自主的 Agent 组成,每个 Agent 封装了与之相关的要素,并根据特定的目标和环境状态,动态地与其他 Agent 通信获取信息,互相协作完成整个问题的求解[3].以 Agent 为能耗调节和管理的实体,以组织抽象 Agent 为基础构造的 DES 自适应能耗技术是对 DES 结构、能耗行为特征更自然的描述,更有利于人们分析、理解和优化 DES 能耗.然而,基于 Agent 的软件系统缺乏对模型的精确定义,尤其是不能为系统提供有效的推理和验证方法,因此,可借助形式化方法对基于 Agent 的 DES 能耗进行描述和研究,逐步扩充动态描述 DES 基础组件及其交互行为、时限约束以及能耗状态的机制,从而建立 DES 能耗管理与分析的理论体系.Petri 网作为一种直观的图形建模工具和一种具有丰富数学基础的形式化模型[4],可以广泛应用于描述和研究并发、异步和分布式特征的系统.

本文围绕 DES 的自适应能耗管理和分析过程面临的挑战和新需求,提出一种基于 Agent 的 DES 自适应能耗管理与分析方法.首先,立足 DES 结构、硬件部署、设备属性等信息,以降低系统整体能耗为目标,提出基于 Agent 的 DES 自适应能耗管理策略,用于动态计算启动设备集和设备的电压;其次,给出分布式嵌入式能耗网(DE-Net)的语法和支撑机制,利用它构建 DES 基础组件的模型库,并根据实际执行流程和部署情况生成相应的能耗模型;最后,采用 CTL 将系统和模型性质转化成一些 CTL 公式,并借助 Petri 网的操作语义及模型状态空间验证方法的正确性和有效性.

本文第 1 节给出相关工作.第 2 节给出系统的框架和自适应能耗需求.第 3 节给出基于 Agent 的自适应能耗管理策略.第 4 节是分布式嵌入式系统的能耗模型构建.第 5 节提出 DES 能耗值分析和模型验证方法.第 6 节通

© 中国科学院软件研究所 http://www.jos.org.cn

过仿真实验说明方法的有效性.最后是结论和下一步工作.

1 相关工作

高效地使用能量来最大化节点生存寿命或提高能量效率,是 DES 面临的挑战之一.国内外研究学者已经意识到研究低能耗的必要性,并提出了一些方法来降低 DES 的能耗.文献[5]提出了一种利用 BP 神经网络在体系结构级估算软件能耗的模型,该能耗模型对 5 个体系结构级软件特征量进行度量,使用 BP 神经网络拟合出软件特征量与嵌入式软件能耗的非线性函数关系.为了提高可靠性和能效,文献[6]提出一种基于链路质量和能耗的路由协议,该协议根据端到端的链路质量评估机制、剩余能量和跳数进行路由选择.Senn 等人在文献[7]中针对具体的体系结构分析与设计语言 AADL,研究基于 AADL 模型的嵌入式系统能耗评估精化方法,并根据设计流程 AADL 组件所采用的规范精化过程,提出不同的精化层次.基于 EDF 和速率调度算法,文献[8]提出了 DES 新的空闲时间管理技术——服务率比例空闲时间分布,以减少 DES 的能耗.该方法同时引入容错机制,以便利用动态松驰机制来维护断点并提供故障回滚功能.Mahapatra 等人提出了一种 DES 的能耗优化调度模型,以处理不同时间的任务运行,并在系统任务集调度中采用一种新的空闲时间分配方案[9],每个节点上的活动组件会周期性地确定服务速率,并根据不同网络节点所观测到动态交通情况来使用电压调节.文献[10]提出了一种基于质量信息驱动的物联网感知层能耗管理方法.该方法能兼容底层所使用的协议,且保证能耗的有效性,而无需牺牲所关注的质量属性.这些方法对更高抽象级别的能耗研究相对较少,需要进一步分析各软件层次中影响系统能耗的关键因素,并考虑软件优化方法对整个系统能耗的影响.

Agent 是指驻留在某一环境下,能持续自主地发挥作用,具有驻留性、反应性、社会性、主动性等特征的计算实体.近年来,基于 Agent 的自适应技术技术引起学术界及工业界的共同关注.文献[11]提出了一种基于多 Agent 的上下文感知学习系统,并采用开源代码 LMS 及其组件加以实现,最后将该方法应用到学习管理系统以验证功能的正确性.基于 Agent 的模型驱动的软件架构用于解决软件的自适应问题[12].该方法解决传统面向对象方法中的静态组件和行为特征,可以增强模型的复用性和灵活性,使得对模型的维护可以转化为对实际软件系统的维护.Chang 等人采用谓词变迁网对多 Agent 进行描述.该方法比较全面地考虑可扩展的多 Agent 建模,并解决 Agent 的动态属性、通信及协调的问题[13].文献[14]提出一种协调组件用于对多 Agent 系统的资源共享进行建模.该方法定义了协调组件的 4 个层次及其相关的 4 个步骤,以便解决资源共享的复杂过程,着色 Petri 网及其相关理论用于对过程进行分析.这些方法从不同方面对基于 Agent 的系统进行分析,并取得了一定的效果.然而,它们没有涉及如何构建自适应软件实体,以支持自适应系统中实体的自主性,并主动地获取环境信息,从而实现系统能耗自适应的目标.此外,这些工作没有考虑系统能耗的实时变化,在系统的结构和行为发生变化时,它们不能在线地自动调整系统的能耗并提供合理能耗管理方案,因而不能满足 DES 能耗的稳定性要求.

形式化方法在计算机软硬件领域的重要作用逐渐为人们所认同,它有助于增加软件开发人员对系统的理解,便于及时更正设计中的错误和缺陷,是保证 DES 能耗自适应方法正确性及提高系统可靠性的重要手段.已有多种形式化方法用于嵌入式软件系统规范和验证.基于状态机的形式化模型适合对系统结构和行为的描述,例如,文献[15]提出一个基于环境建模的物联网服务提供框架,在此框架中,环境实体作为一个重要的概念被引入,用于刻画物理世界中各种物体的属性和行为.该方法以时间自动机为工具,对环境实体和物联网服务的行为进行建模,并用时间自动机网络来刻画服务的组合和它们与环境实体的交互.文献[16]从构件化嵌入式软件体系结构出发,采用基于路径的系统能耗分析评估方法,在嵌入式系统架构设计阶段对其能耗特性进行分析与评估.在此评估体系中,软件体系结构应用进程代数语言 CSP 进行形式化描述,能耗特性在构件接口级别定义.Lanese 等人采用进程演算对物联网进行建模,并引入互模拟的两个概念[17],其中一个用于捕捉终端用户的行为,另一个则用于进行组合推理.我们提出一种策略驱动的可靠嵌入式系统建模与分析方法[18],采用面向方面思想提取可靠性保障策略相关关注点.通过构造关注点模型,并利用编织机制,将关注点模型动态地集成为一个完整的嵌入式系统可靠模型.由于 DES 能耗自适应在资源竞争、空间和环境感知等特性方面与计算实体区别很大,要求建模语言具有较强的表达能力,以便对系统交互行为进行精确的刻画.

© 中国科学院软件研究所 http://www.jos.org.cn

2 系统框架和自适应能耗需求

本节以 DES 自适应能耗管理和分析为驱动,介绍系统的整体框架,包含各部分研究内容及其关系,并基于实例分析分布式嵌入式系统自适应能耗需求,提取相关的组件及其属性,为后续的能耗管理奠定基础.

具体的系统框架如图 1 所示.该框架主要从分布式嵌入式系统的能耗需求分析、模型构建、能耗分析与优化、能耗值计算和验证这 4 个方面展开研究:

- 需求分析阶段:分析 DES 的能耗需求及其组件,如传感器、执行器和 Agent 等.从应用所需完成功能的角度出发分析其执行流程,给出相应的任务集、任务与设备间映射关系等需求信息,为后续的研究奠定基础.
- 能耗模型构建阶段:采用 Petri 网对 DES 中的传感器、执行器、Agent 等组件进行建模,并刻画环境信息及交互行为以形成系统的模型库;其次,对模型库中的基础组件进行属性配置,并根据任务属性、任务与设备的映射关系集成各基础组件的模型,以构成系统的能耗模型.
- 能耗分析和优化阶段:基于设备完成任务集的实际含义,计算出系统的可能路径.分析每个执行路径的能耗,包括最大能耗、最小能耗、当前能耗等信息.同时,根据 DES 的结构特征和运行机理,提出能耗管理策略,包括设备启动和休眠规则、电压调整方案等;最后,根据执行路径的能耗信息和能耗管理策略来计算系统的执行决策.
- 能耗值计算和验证阶段:根据能耗模型中库所、变迁等的实际映射,从任务、设备等角度着手计算系统的能耗值.基于模型的执行语义,利用 Petri 网的相关理论分析和验证模型的正确性、策略的有效性.

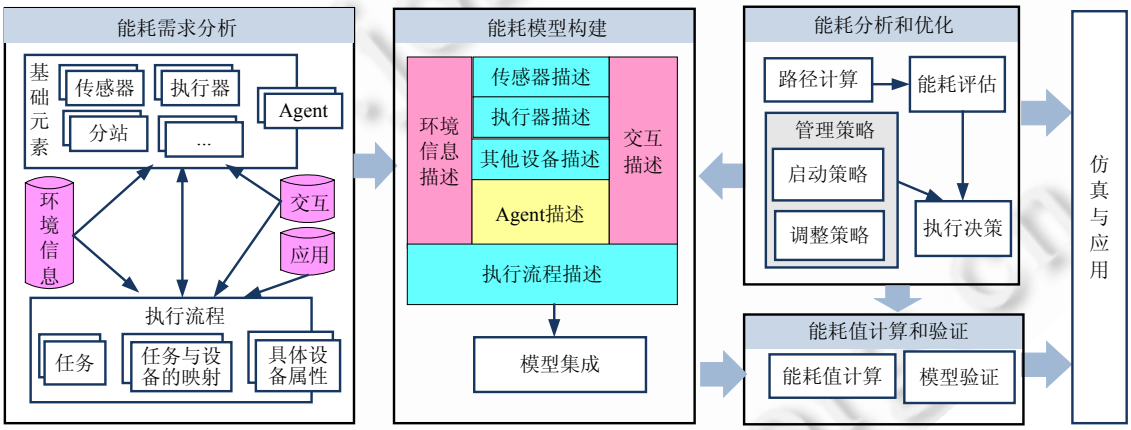

Fig.1 System architecture

图 1 系统框架

DES 主要是由传感器节点、执行器节点等设备及其交互连接而成,其中,传感器主要用于将被测物理量转换为电信号输出,执行器是将控制信号转换为被控物理量的装置.分站用于对传感器输入的信号和通信设备传输来的信号进行处理,控制执行器工作.主机主要用来接收监测信号、人机对话、输出控制、控制打印输出、联网监测等.本文在分布式嵌入式设备中加入 Agent 组件,用于分析和控制系统能耗.Agent 可用于获取和分析环境信息(设备的状态、能耗等),并根据一定的策略对环境结构进行指导和修改.

由于 DES 的功能是由多个独立运行的子功能构成,本文将每个子功能称为任务.在分布式嵌入式系统中,每个任务会有多个可用设备与之对应.即,有多个设备可以完成该功能,但是设备的属性不同,比如位置或性能不同.首先需要分析 DES 的需求,包括任务集、可用设备集、任务间关系等.通常,一个系统设备会有多种活动状态,本文仅考虑设备的工作状态和休眠状态.设备在各个状态的能耗可以根据数据手册或者通过测试和仿真得到.本文将设备、Agent、子系统间的交互定义为连接件,即,用连接件实现系统的连接和通信,并且将连接件看作

© 中国科学院软件研究所 http://www.jos.org.cn

与设备同级的组件,因此,连接件也可以看作是连接设备.由于本文的研究切入点是设备的能耗管理和分析,在分析连接件时仅考虑其能耗属性,没有对其进行深入优化.一个设备可以有多个操作,每个操作对应设备可行的一个事件.连接件通过描述设备、子系统间接口的交互行为实现设备间的交互.下面将从设备、Agent、连接件等方面对 DES 展开需求分析.

定义 1. DES 的能耗需求是一个八元组:$\Xi=(AE,AG,AL,RE,TW,RA,OP,D)$,其中,

(1) 设备集 $AE=\{ae_1,ae_2,\ldots,ae_n\}$,$ae=(type,op,ra,ep,ei)$,其中,$type,op$ 是设备的类型和可能执行操作;$ra(ae)=(V_{\max},V_{\min},nw_{i,j})$是 ae 的属性函数,分别描述设备的最大、最小供压、具体位置;$ep(ae)=(etp,enp)$是操作的执行时间和单位能耗值,假设设备的启动操作能耗较大;$ei=(ty,func)$是设备接口的集合,接口具有类型、功能规约两个属性.

(2) 连接件集 $AL=\{al_1,al_2,\ldots,ag_n\}$,$al=(data,oa,nl,li)$,其中,$data$ 是连接件的传输内容;oa 是连接件关联的设备或 Agent 集合,假设$|oa|=2$;nl 是一个固定值,表示连接件的能耗;li 用于描述连接件的输入输出接口,假设与 AL 对应的每个设备都有两个接口,用来负责该设备的输入和输出.连接件用于实现设备、子系统间的交互.此外,连接件的能耗也可以设置成动态变化的.

(3) 代理集 $AG=\{ag_1,ag_2,\ldots,ag_n\}$,$ag=(eg,lg,om,op,gi)$,其中,$eg\subseteq AE,g\subseteq AL$ 是 ag 管辖的设备和连接件集;$om:AG\times RA\rightarrow\{Start,Sleep,reV,raV\}$是代理对设备的可执行操作;$op$ 是代理的可能操作集合;gi 是 ag 的接口集合,一个 ag 具有多个输入输出接口,如接收设备的数据、与其他 ag 的交互接口等.

(4) 任务集 $TK=\{tk_1,tk_2,\ldots,tk_n\}$,$tk=(et,tp,rl)$,其中,$et:TK\rightarrow AE^*$表示任务 tk_i 的可用设备集,$tp:TK\rightarrow op^*$表示完成所需操作的集合;rl 是任务间的关系函数,用于刻画系统的执行流程,主要考虑顺序(>)、选择(+)、并行关系(||).设备可以完成多个任务,一个任务也可以被多个设备调用.记集合 $VW(ae_i,TK_j)=\{tk_k|ae_i\in et(tk_k)\wedge tk_k\in TK_j\}$为设备 ae_i 在任务集 TK_j 的作用集.

(5) D 是 DES 的截止时限,具体单位可根据实际情况设置.

在 DES 中,设备通过执行一系列操作来实现其功能.传统的能耗分析方法通常将能耗特性建立在设备级别的实体上,然而对于不同的应用场景,设备的执行操作和频率可能不同,即,不同场景中同一个设备的能耗不同.因此,以设备为基本能耗单元进行能耗分析可能会造成较大的误差.鉴于此,本文将能耗特性的描述建立在操作级别上.

例 1:本文用一个矿井监控系统来说明 DES 的需求模型.矿井监控系统主要包括安全保障、自动报警、人员定位、自动报警和考勤管理等子系统,不同子系统的执行流程不一样.在矿井监控系统中,各个子系统通过传输消息达到通信和交互,因此是一种分布式嵌入式系统.其中,人员定位子系统的执行流程是:人员 me_i 随身携带的射频卡进入读卡器的工作区域,射频卡被激活后($tk_{i,1}$)(未进入读卡器工作区域,射频卡不工作,处于休眠状态),将人员编码加密信息发射出去($tk_{i,2}$);读卡器发送信号以启动射频卡 tk_1,同时接收射频卡发来的无线信号 $tk_{i,3}$,然后汇总传输给分站 tk_2.经分站接收($tk_{i,4}$)、处理($tk_{i,5}$)后,提取出人员编码相关信息 $tk_{i,6}$,经数据传输接口送至地面监控计算机,地面监控器将接受的数据进行汇总(tk_3)、分析(tk_4),完成矿井人员自动跟踪定位管理和考勤.这里,将设备简化为射频卡、读卡器、分站、地面监控器这 4 种设备.根据设备的地理位置设置 3 个 Agent,每个 Agent 下有一个分站,用以执行 Agent 的操作.假设所有连接件的能耗均为 2mw,其中,时间单位 TTU 为 2ms,单位能耗的单位为 mw.射频卡的最大/最小供电电压分别为 3.6V 和 2.3V,读卡器为 5V 和 2.85V,分站为 3.3V 到 6V.地面监控器主要由电脑主机组成,所以不考虑其电压调整.射频卡的操作流程是:启动($op_1(1,3)$)后经过低噪声放大、混频处理、放大和滤波($op_2(1,3)$),转换信号后输出信号($op_3(2,2)$).射频卡主要完成任务 $tk_{i,1}(op_1)>tk_{i,2}(op_2,op_3)$.读卡器的工作原理比较复杂,包含两个并行模块,分别用以发送信号来启动射频卡($op_4(1,2)$)、接受射频卡发来的信号($op_5(1,2)$),并汇总传输到分站($op_6(2,2)$).读卡器的业务流程为$(tk_1(op_4)||tk_{i,3}(op_5)>tk_2(op_6))$.分站工作原理是:首先从读卡器上采集射频卡发送的数据信息($op_7(1,3)$),然后将信息传输到 RAM 芯片缓存起来($op_8(2,2)$),另一端单片机将缓存的数据进行抽取($op_9(1,3)$),通过总线传输给井上的地面监控器($op_{10}(2,5)$).地面监控器是由一个主机构成,主要接受各个分站传输上来的数据($op_{11}(1,2)$)、汇总数据($op_{12}(3,3)$)、分析数据,以形成最终报表($op_{13}(4,5)$).

© 中国科学院软件研究所 http://www.jos.org.cn

地面监控器的流程为 $tk_3(op_{11},op_{12}))>tk_4(op_{13})$.整个监控系统从人员身上佩戴的射频卡发出信号到处理完毕的时间是 30.假设系统现在有 40 个人员在井下,某一时刻人员的具体分布为(25,25,30,31,33,42,44,45,65,102,103,104,104,105,105,125,122,140,175,150,25,114,31,133,42,140,36,163,202,203,224,254,305,306,125,122,240,275). 整个矿井的线路长 500 m,读卡器每隔 20 m 安装一个,每 100 m 设立一个分站,每个读卡器的接受范围为±15.

分析矿井监控系统,可以发现存在如下问题:(1) 由于井下人员是移动的,系统需要根据当前井下人员分布动态调整启动读卡器、分站以降低系统能耗.当读卡器和分站工作区域的射频卡数量发生改变时,设备的执行操作集也会发生改变,不同操作的工作频率导致其能耗不同.因此,有必要根据设备的执行操作来动态地调整设备的电压,以降低设备的能耗.如何设计一个自适应的能耗管理策略,是 DES 需要解决的重要问题.(2) 随着系统的实施,设备的数量、布局经常需要改变,如某个读卡器故障,或者增设读卡器等.因此,设计一个灵活、可重用的模型是 DES 能耗设计首先要解决的问题.

针对 DES 面临的两个问题,本文拟采取如下两个措施:(1) 在确保 DES 功能的前提下,根据当前设备的位置和执行情况,尽可能地减少启动的设备集.基于 DES 的时限,根据设备执行的操作动态地调整启动设备的电压,进而降低系统的能耗.同时,在分站上引入 Agent 思想,用以实现对能耗的自适应管理.(2) 设备、连接件、Agent 的建模过程借鉴模板的思想,以建立系统的模型库.当用户需求发生改变时,只需更改具体映射,进而提高模型的可重用性.

3 基于 Agent 的自适应能耗能耗管理策略

由于 DES 除了实现基本的目标(即时间约束)以外,往往还具有更加严格的能耗约束.因此,设计者不仅要考虑如何降低能耗,而且要保证系统在满足时限约束的前提下尽可能地降低其能耗这一新问题.本文根据 DES 结构和领域特征,基于 Agent 研究 DES 的自适应能耗管理策略,包括启动和调整两个子策略,其中,启动策略主要根据系统的执行流程,基于设备的当前状态和属性动态计算启动设备集;而调整策略则从系统运行时限着手,计算启动设备的动态电压.

3.1 启动策略

根据系统的执行流程,基于设备的当前状态评估其能耗值.当设备周围没有事件发生时,则将部分模块置于空闲状态,即,设备的能耗比较低.按照一定的规则将这些设备上的处理操作转移到可替换设备,原设备调到低消耗状态(休眠状态).假设设备 ae 完成任务 tk 一次成功运行所需执行的操作序列称为设备 ae 的路径,如射频卡为了完成任务 $tk_{i,2}$ 的路径为 op_2,op_3.由于设备通常是由某个初始操作开始,运行到某个终止操作结束,根据完成功能的不同,有些设备可能有多个初始操作和多个终止操作.由于设备初始和终止时的特性是系统关心的内容,在具有多个初始操作的设备中,通过添加启动操作 ini 将多个初始操作转化为一个唯一的初始操作 ini.同理,引入正常结束 end 操作和超时结束 fo 操作.此外,为了统一描述设备的执行路径,在只有 1 个初始操作的设备中也添加一个 ini 操作表示设备的开始运行.因此,设备的一次成功运行表示为执行了一个以 ini 操作开始、以 end 或 fo 操作为结束的操作序列,即,设备执行的一个路径.ini,end,fo 操作在设备实际运行描述中是不存在的,它所执行的操作是一个空操作,本文规定这些操作的能耗和运行时间为 0.由于设备完成不同任务所调用的操作不同,对应的能耗也会有差异.此外,设备完成同一个任务时,其选择路径不唯一,因此其能耗也会不同.设设备 ae_i 完成任务集 TK_j 的路径集合为 $Lat(ae_i,TK_j)=\{Lat_1,Lat_2,\ldots,Lat_n\}$,其中,路径 $Lat_k=\{op_{k,1},op_{k,2},\ldots,op_{k,f}\}$.路径 Lat_i 的能耗值为

$$en(Lat_k)=\sum_{op_{k,l}\in Lat_k} enp(op_{k,l})\times etp(op_{k,l}).$$

设备 ae_i 完成任务集 TK_j 的平均能耗值为

$$avg_en(ae_i,TK_j)=\frac{\sum_{Lat_k\in Lat(ae_i,TK_j)} en(Lat_k)}{|Lat(ae_i,TK_j)|}.$$

© 中国科学院软件研究所 http://www.jos.org.cn

如,射频卡为了完成任务 $tk_{i,2}$ 的能耗为 7.

定义 2. 设Ξ是 DES 需求模型,$\forall ae_i,ae_j\in AE,\forall ae_i,ae_j\in AE$ 是系统两个不同的设备,$ext(ae_i,ae_j)$为两个设备的距离.

(1) 若 $VW(ae_i,TK)\cap VW(ae_j,TK)=\varnothing$,则称 ae_i,ae_j 在任务集 TK 下是无关的,记作 $tc_i\infty_{0,TK}tc_j$;否则,记作 $tc_i\infty_{TK}tc_j$;

(2) 若 $VW(ae_i,TK)\subset VW(ae_j,TK)\wedge ext(ae_i,ae_j)\leqslant\xi$,其中,$\xi$是距离的阈值(根据设备的辐射范围设置),则称 ae_j 覆盖 ae_i,记作 $ae_j\nabla_{TK}ae_i$;否则,记作 $ae_j\Delta_{TK}ae_i$;

(3) 若 $ae_j\nabla_{\mathrm{TK}}ae_i$ 且 $avg_en(ae_i,VW(ae_i))>avg_en(ae_j,VW(ae_j))$,则称 ae_j 在任务集 TK 下完全覆盖 ae_i,记作 $ae_j\nabla_{all,TK}ae_i$;

(4) 若 $VW(ae_i,TK)=VW(ae_j,TK)\wedge ext(ae_i,ae_j)\leqslant\xi$,则称 ae_j 等同 ae_i,记作 $ae_i\approx_{TK}ae_i$.

两个设备是无关的,是指其作用集不相交;两个设备完全覆盖,则指设备的作用集之间存在属于关系.在实际运行中,虽然有些设备具有相同的功能,但是由于其地理位置不同,导致执行的能耗也会有所差异.因此,在定义覆盖和等同关系时,不仅需要考虑设备的作用集,而且还需要考虑设备的间隔距离.分析矿井监控系统可以发现,射频卡间是无关的.设人员定位的距离阈值为 20,发现读卡器 ae_2^r 的作用集:

$$VW(ae_2^r,TK)=\{tk_1,tk_{1,3},tk_{2,3},tk_{3,3},tk_{4,3},tk_{5,3},tk_{21,3},tk_{23,3},tk_2\}\subset$$
$$VW(ae_3^r,TK)=\{tk_1,tk_{1,3},tk_{2,3},tk_{3,3},tk_{4,3},tk_{5,3},tk_{6,3},tk_{7,3},tk_{8,3}tk_{21,3},tk_{23,3},tk_{25,3},tk_{27,3},tk_2\},$$

且 $ext(ae_2^r,ae_3^r)=20$, 所以, $ae_3^r\nabla ae_2^r$.进一步计算两个设备的平均能耗值,可知 ae_3^r 完全覆盖 ae_2^r. 同理,可以分析其余设备间的关系.

DES 的能耗需求为$\Xi=(AE,AG,AL,RE,TW,RA,OP,D)$,初始化当前设备集 $AE_{now}=AE$,当前任务集 $TK_{now}=TK$,启动设备集 $AE_{ac}=\varnothing$,设备 ae 的执行任务集为 $TK(ae)=\varnothing$,则其启动策略是:

(1) 候选设备集计算:$\forall ae_i\in AE_{now}$,若$\exists ae_j$ 使得 ae_j 在任务集 TK 下完全覆盖 ae_i,则 $AE_{now}=AE_{now}-ae_i$.

(2) 启动设备集计算:$\forall ae_i\in AE_{now}$,若 $VW(ae_i,TK_{now})=\max\{VW(ae_j,TK_{now})\},ae_j\in AE_{now}$,则 $AE_{ac}=AE_{ac}\cup ae_i$, $AE_{now}=AE_{now}-ae_i,TK_{now}=TK_{now}-VW(ae_i,TK_{now}),TK(ae_i)=VW(ae_i,TK_{now})$.若 $TK_{now}\neq\varnothing$,则继续转步骤(2);若 $TK_{now}=\varnothing$,则转步骤(3).

(3) 输出 $AE_{ac},TK(ae_i)$.进一步根据任务间关系增加相应的连接件,输出启动连接件集合 La_{ac}.

利用启动策略可以计算出系统的启动设备集合,其原理是,根据实际设备的能耗随着执行任务数的增加不呈线性增加.因此,可以充分使用设备完成多个任务,减少系统的启动设备数,从而减低系统的整体能耗.Agent 根据系统需要完成的功能,利用启动策略动态计算启动设备集和连接件集合.

3.2 调整策略

由于 DES 整体有时限要求,但没有细化到各个设备,因此有必要分析设备的局部时限要求,以降低设备的供应电压.下面根据关键任务集提出调整策略,并计算设备的局部时限和动态供电电压.

设任务 tk_i 在设备 ae_g 中有操作集合 $op(tk_i,ae_g)=\{op_1,op_2,\ldots,op_k\}$,则关键任务集的计算方法为:

① 始化关键任务集 KC,初始化 $KC=\{tk_i|\forall tk_j\in TK\rightarrow RL(tk_i,tk_j)\notin\{||,+\},tk_j\in TK\}$,$temp=TK\text{-}KC$,转步骤②.

② 调整关键任务集,$\forall tk_i\in temp,tempc_i=\{tk_j|RL(tk_i,tk_j)\in\{||,+\},tk_j\in TK\}\cup tk_i$,若 $et_i=\max\left\{\sum_{op_f\in op(tk_j,ag_e)}etp(op_f)\right\}$,

则 $KC=KC\cup tk_i,temp=temp-tempc_i$,若 $temp\neq\varnothing$,则转步骤②;否则,输出 KC.

关键任务集就是 DES 中运行时间最长的一条执行路径,下面基于此调整任务的运行时间和供电电压.设关键任务集为 KC,tk_i 由设备 ae_g 执行,具体调整策略为:

(1) 计算关键路径集的运行时间:$tc_{\max}=\sum_{tk_j\in KC}\sum_{op_f\in op(tk_j,ag_e)}etp(op_f)$.

(2) 计算空闲时间:$D_{rem}=D-tc_{\max}$.

© 中国科学院软件研究所 http://www.jos.org.cn

(3) 计算任务 tk_i 的原始运行时间 $tc_i=\sum_{op_f\in op(tk_i,ag_e)} etp(op_f)$.

(4) 计算任务 tk_i 的新运行时间:

$$tc_i':tc_i'=\begin{cases} tc_i+D_{rem}\times\dfrac{\sum_{op_f\in op(tk_i,ag_e)} etp(op_f)}{\sum_{tk_j\in KC}\sum_{op_f\in op(tk_j,ag_e)} etp(op_f)}, & tk_i\in KC \\ tc_j', & tk_i\notin KC\wedge tk_j\in KC\wedge rl(tk_i,tk_j)\in\{+,||\} \\ tc_i, & \text{其他} \end{cases}.$$

(5) 计算任务 tk_i 运行时间的调整率:$etc_i=\dfrac{tc_i}{tc_i'}$.

(6) 计算任务 tk_i 新的供电电压:$V_{dd}'^{i}=V_{\min}^{g}+\dfrac{V_0^g\times etc_i}{2}+\sqrt{\left(V_{\min}^{g}+\dfrac{V_0^g\times etc_i}{2}\right)^2-(V_{\min}^{g})^2}$,

其中,$V_0^g=\dfrac{(V_{\max}^g-V_{\min}^g)^2}{V_{\max}^g}$.

利用调整策略,Agent 可以调整设备的动态电压以适应任务的运行,在保证时限的前提下,降低系统的总体能耗.进一步地,可以按照比例计算各个操作的运行时间 etp'.

4 布式嵌入式系统的 DE-Net 模型

能耗模型是能耗管理和分析的基础.本节围绕分布式嵌入式系统的能耗需求和领域特征,首先给出 DE-Net 网的语法和执行语义,并利用 DE-Net 网刻画分布式嵌入式系统的系统结构和基础组件,以构成分布式嵌入式系统的能耗模型,为系统的分析和验证奠定模型基础.

4.1 DE-Net模型基础

Petri 网作为一种图形化建模工具和一种具有丰富数学基础的形式化模型,可以广泛应用于描述和研究并发、异步和分布式特征的系统,因此,Petri 非常适合描述分布式嵌入这种松耦合的系统.下面介绍一些系统运行的基本概念,其余的概念见文献[4].

定义 3. 十一元组$\Omega=(PN,IO,D,A_T,A_F,\lambda,\Gamma,TI,TA,PI,PA)$称作分布式嵌入式能耗网(distributed embedded net,简称 DE-Net),其中,

(1) $PN=(P,T,F)$是一个基本 Petri 网,其中,P,T,F 分别表示库所、变迁和弧的有限集,且两两不相交;

(2) $IO\subset P$ 是一类特殊的库所,称为Ω的接口,用虚线圆圈表示;

(3) D 为非空有限的个体集,规定 f_D,f_S 分别为 D 上给定的公式集和符号集;

(4) $A_T:T\to f_D$,对于 $t\in T,A_T(t)$中的自由变量必须是以 t 为一端的有向弧上自由变量;

(5) $A_F:F\to f_S$,若$(p,t)\in F$ 或$(t,p)\in F$,则 $A_F(p,t)$或 $A_F(t,p)$为 n 元符号集,缺省为空;

(6) $AT:T\to N^*\times N^*$是变迁的属性函数,$\forall t_i\in T,AT(t_i)=(\lambda_i,en_i)$描述 t_i 的优先级和单位能耗,默认值为 0.本文假设:λ_i 越小,变迁 t_i 的优先级越大;

(7) $Ct:P\to N^*\times N^*$是库所的属性函数,描述了库所的延迟时间和能耗,默认为 0;

(8) $M_0:P\to f_S$ 是Ω的初始标识,$p\in P,M_0(p)$不含任何自由变量;

(9) $\Gamma=\{\Gamma_i|i\in N^*\}$是 DE-Net 的有限集,其中每个组件称为$\Omega$的一个页面;

(10) $TI\subset T$ 是替代节点的集合,其中每个页面对应一个替代节点,用实心矩形表示;$PI\subset P$ 是端口节点的集合,端口节点描述替代节点的输入和输出库所,用实心圆圈表示;

(11) $TA:TI\to\Gamma$是一个页分配函数,即,为每个替代节点分配一个页面;PA 是端口映射函数,主要是把端口节点映射到对应页面的输入和输出接口.

© 中国科学院软件研究所 http://www.jos.org.cn

DE-Net 模型是对确定的时间延迟进行建模,当变迁的触发发生冲突时,以竞争来选择触发的变迁,将库所 P_i 的延迟时间记作 ct_i.DE-Net 模型主要是对设备、连接件、Agent 或整个分布式嵌入式系统进行建模,其中,库所、变迁和接口分别描述组件的所处位置、可能操作和输入输出参数.个体集 D 主要用来描述 DES 中的资源,本文将 DES 的资源和信息抽象成个体 $d_i=(it,i,RW_i)$,其中,$it\in\{e,g,l,d\}$说明该个体所描述的对象,e,g,l,d 分别表示可用设备集、Agent 集、连接件、数据包.i 表示个体的标记,如,设备 ae_j 在设备集 AE 中所处位置为 j.当 it 为 d 时,RW_i 表示数据包的内容;否则,RW_i 为描述对象的属性,如能耗、电压.设 $k\#(d_i)$表示个体 d_i 中的第 k 个组件,而把系统中的普通数据包统一抽象成个体φ.如果没有无特殊说明,则 DE-Net 模型中出现的个体均为φ.因为设备没有下一层组件,所以其模型的Γ,TI,TA,PI,PA 均为空.而 Agent 模型中每个页面代表一个设备,且其接口非空;而分布式嵌入式系统中每个页面表征一个 Agent.任意$x\in(P\cup T)$,集合${}^{\bullet}x=\{y|y\in(P\cup T)\wedge(y,x)\in F\}$和$x^{\bullet}=\{y|y\in(P\cup T)\wedge(x,y)\in F\}$分别对应于 x 的输入和输出;任意 $t_i\in T$,将变迁 t_i 的输入/输出弧$(A_F(t_i,p)/A_F(p,t_i))$以及 $A_T(t_i)$中出现的自由变量集记作 $FV(t_i)$.

图 2 描述了一个简单的 DE-Net 模型Ω_1,该模型中有一个页面(虚线框).Ω_1 中的替代节点 T_4 分配给页面 Ta,也就是说,该节点描述了页面 Ta 的内部操作.相应地,端口节点 P_4,P_6 分别映射为页面 Ta 的输入和输出接口 P_8,P_9.

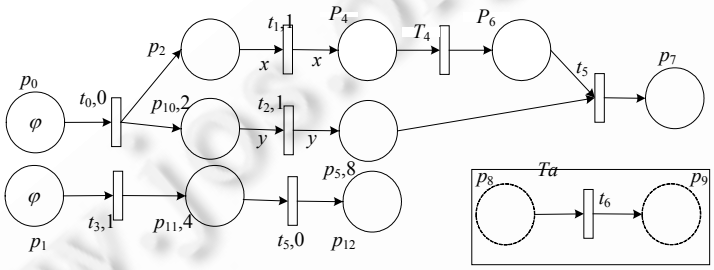

Fig.2 DE-Net model Ω_1

图 2 DE-Net 模型Ω_1

在各库所中,个体的分布状况称为 DE-Net 模型的标识,记作 $M.\forall p\in P,M(p)=\{(d_i,q_i),\ldots,(d_k,q_k)\}$,其中,$\{d_i,d_j,\ldots,d_k\}$为标识 M 下库所 p 中存放的个体集,而$\{q_i,q_j,\ldots,q_k\}$为相应的数量.

定义 4. 设Ω为 DE-Net 模型,在θ时刻,模型处于标识 M.设$\forall p_i\in P$,在标识 M 下库所 p_i 有 j 个个体,d_i^k 为 p_i 的第 k 个个体$(0<k<j)$,ξ_k 为个体 d_i^k 的产生时刻,ct_i 为库所 p_i 的延迟时间,称向量:

$$TS(p_i)=(TS_i^1,TS_i^2,\ldots,TS_i^j)$$

为库所 p_i 的所有个体等待时间,其中,$TS_i^k=\max\{ct_i-(\theta-\xi_k),0\}$.

称 $TS_i(d_i^k)$ 为个体 d_i^k 的等待时间,若库所 p_i 中只有 1 个个体,则直接用 $TS(p_i)$表示.$TS_i(d_i^k)=m$ 表示系统还要再等 m 时间单位才可以使用个体 p_i^k,$TS(p_i^k)=0$ 表示该个体可用.为了减少分析的复杂度,本文假设:若库存的时间延迟大于 0,则该库所中只存放同一种个体.记 $TS(M,\theta)$为θ时刻标识 M 下库所的等待时间集合.如Ω_1 在全局时间 1 时,库所 P_2,P_{10},P_{11} 中分别有个体 d_1,d_2,d_3,且 $TS(P_2)=TS(d_1)=0,TS(P_{10})=1,TS(P_{11})=3$.记 M^a 为 M 中可用的个体分布,而 M^u 标记 M 中不可用的个体分布.$|M^a(P_i)|=k$ 表示在库所 P_i 中有且仅有 k 个个体的等待时间为 0,$|M^u(P_i)|=k$ 表示在库所 P_i 有 k 个个体不可用.如Ω_1 在全局时间 1 时,$M_1(P_2)=(d_1,1),M_1(P_{10})=(d_2,1),M_1(P_{11})=(d_3,1)$,但是只有库所 P_2 的个体 d_1 可用.因此:

- 在全局时间 1 时,$M_1^a(P_2)=(d_1,1),M_1^u(P_{10})=(d_2,1),M_1^u(P_{11})=(d_3,1)$;
- 在时刻 2 时,$M'=M_1$,而 $M_1'^a(P_2)=(d_1,1),M_1'^a(P_{10})=(d_2,1),M_1'^u(P_{11})=(d_3,1)$.

设Ω为 DE-Net 模型,二元组 $S=(M,TS)$称为Ω在θ时刻的状态,其中,M 为标识,描述了系统的资源分布;TS 为在θ时刻标识 M 的等待时间,刻画了系统的时间特性.初始状态 $S_0=(M_0,TS_0)$,TS_0 为一个零向量(在初始状态下所

© 中国科学院软件研究所 http://www.jos.org.cn

有令牌都是可用的).如,$S_1=(M_1,TS_1)$是Ω_1在时刻 1 的状态,其中,$TS_1=\{0,1,3\}$.

二元组 $S=(M,TS)$为Ω在θ时刻的状态,根据状态的定义可知,有两种方式可以改变状态 S:(1) 时间的消逝.在时刻$\theta+\omega(\omega>0)$,由于系统中令牌的等待时间发生改变,促使系统到达新的状态 S',记作 $S[\omega\rangle S'$.(2) 变迁替换的触发.变迁 t_i 替换的触发会产生新的标识,从而系统进入新的状态 S',记作 $S[t_i\langle d_1,d_2,\ldots,d_n\rangle\rangle S'$.

若Ω_1在时刻 1 处于状态 $S_1=(M_1,TS_1)$下,则不触发任何变迁,系统在时刻 2 处于另一个状态 $S_2=(M_2,TS_2)$,其中,$M_1=M_2,TS_1\neq TS_2$.若Ω_1在时刻 1 触发 t_1,则系统进入新的状态 $S_3=(M_3,TS_3)$,其中,$M_1\neq M_3,TS_1\neq TS_3$.将 S 经过时间ω后再触发替换 $t_i\langle d_1,d_2,\ldots,d_n\rangle$到达 S',$S[\omega\rangle S'$且 $S'[t_i\rangle S''$,记为 $S[(t_i\langle d_1,d_2,\ldots,d_n\rangle,\omega)\rangle S''$.$\forall t\in T$,若 $FV(t_i)=\{x_1,x_2,\ldots,x_n\}$,则个体集$\{d_1,d_2,\ldots,d_n\}$满足 $d_i\in\{M^a(p)|p\in{}^\bullet t\cup t^\bullet\}$且 d_i 对应于变量 x_i,将个体 $d_1,d_2,\ldots,d_n$ 分别替换 $x_1,x_2,\ldots,x_n$ 所得到变迁 t 的实例称为变迁 t 的一个替换,记作 $t\langle x_1\leftarrow d_1,x_2\leftarrow d_2,\ldots,x_n\leftarrow d_n\rangle$,简写为 $t\langle d_1,d_2,\ldots,d_n\rangle$.替换主要是对 t 的输入/输出弧以及 $A_T(t)$中出现的所有自由变量绑定相应的个体.记 $A_T(t)\langle d_1,d_2,\ldots,d_n\rangle$和 $A_F(p,t)\langle d_1,d_2,\ldots,d_n\rangle$分别描述将个体 $d_1,d_2,\ldots,d_n$ 替换到公式 $A_T(t)$和输入弧上谓词 $A_F(p,t)$所得到的值.若替换 $t\langle d_1,d_2,\ldots,d_n\rangle$使得 $A_T(t)\langle d_1,d_2,\ldots,d_n\rangle=\text{true}$,则称 $t\langle d_1,d_2,\ldots,d_n\rangle$是 S 下变迁 t 的可行替换.记变迁 t 在状态 $S=(M,TS)$下的所有可行替换集合为 $VP(S,t)$.设变迁 t 的自由变量集 $FV(t_i)$为空,若$\forall p\in{}^\bullet t,M(p)\neq\varnothing$,则变迁 t 在 S 下肯定存在可行替换.对应可行替换的执行过程为:从 t 的每个输入库所 p_i 中任选一个可用个体 d_i 即可.

设Ω为 DE-Net 模型,$S=(M,TS)$为Ω的一个状态,若 $VP(S,t)\neq\varnothing$,则称变迁 t 在状态 S 下有发生权,记作 $S[t\rangle$.在 DE-Net 模型中,变迁 t 在状态 S 下是有发生权的,当且仅当 t 在状态 S 下存在可行替换.将状态 S 下所有有发生权的变迁集合记为 $ET(S)$.

定义 5. 设Ω为 DE-Net 模型,$S=(M,TS)$为Ω的一个状态,对于变迁 $t_i\in ET(S)$,如果 t_i 满足下列条件:$\beta_i\leqslant\min(\beta_j)$,其中,$t_j\in ET(S)$,则称变迁 t_i 在状态 S 下的触发是有效的.

将状态 S 下所有可以有效触发的变迁集合记为 $FT(S)$.状态 S 下 t_i 的触发是有效的,当且仅当 t_i 在状态 S 下有发生权,且变迁集 $ET(S)$中不存在比 t_i 优先级高的变迁.在状态 S 下通过有效触发变迁 t_i 的一个可行替换 $t_i\langle d_1,d_2,\ldots,d_n\rangle$到达新状态 S'的过程记为 $S[t_i\langle d_1,d_2,\ldots,d_n\rangle\rangle S'$.设$\Omega$为 DE-Net 模型,$M$ 为Ω的一个标识,对于$\forall t_i,t_j\in FT(S)$,若:${}^\bullet t_i\cap{}^\bullet t_j=\varnothing$,则称 t_i,t_j 在状态 S 下是并发;否则,称变迁 t_i,t_j 在 M 下冲突.若两个变迁间存在并发关系,则无论触发哪个变迁都不会影响另一个变迁的触发.将 S 下并发的变迁集合称为 S 的最大并发集,记为 $MT(S)$.集合 $H(S)=\{t\langle d_1,d_2,\ldots,d_n\rangle|t\in MT(S),t\langle d_1,d_2,\ldots,d_n\rangle\in VP(S,t)\}$称为 S 的一个最大触发集.

定义 6. 设Ω为 DE-Net 模型,$S=(M,TS)$为Ω的一个状态,系统通过有效触发 $H(S)$中所有变迁到达新的状态 S',记作 $S[H(S)\rangle S'$,称 S'是 S 的可达状态.S'分别按如下规则计算:

$$\forall t_i\langle d_1,d_2,\ldots,d_n\rangle\in H(S),\forall p_j\in{}^\bullet t_i\cup t_i^\bullet:M'(p_j)=M(p_j)-A_F(p_j,t_i)\langle d_1,d_2,\ldots,d_n\rangle+A_F(t_i,p_j)\langle d_1,d_2,\ldots,d_n\rangle.$$

在 DE-Net 模型中,可达状态按照触发变迁的可行替换进行相应的调整.如果存在触发序列 $H_1,H_2,\ldots,H_k$和状态序列 $S_1,S_2,\ldots,S_k$ 使得 $S[H_1\rangle S_1[H_2\rangle S_2\ldots S_{k-1}[H_k\rangle S_k$,则称 S_k 是从 S 可达的.从 S 可达的所有标识集合记为 $R(S)$,规定 $S\in R(S)$.基于上述触发规则,从初始状态 S_0 出发构造模型的可达状态.

4.2 DE-Net模型的构建

下面针对 DES 的需求模型Ξ,采用 DE-Net 模型对系统的基础组件分别进行建模,进而根据执行流程构造 DES 的能耗模型.为了区别输入接口和输出接口,对于输入接口用上标 I 标示,输出接口用上标 O 标记.另外,在设备、Agent、连接件的变迁和库所前标注对应的组件.如,设备 ae_i 的开始变迁 t_{in} 表示为 $ae_i.t_{in}$.如果变迁是为了描述数据流程而引入,则不标注.

设备 ae_i 的 DE-Net 模型为 Ω_{ae_i},如图 3 所示,其中,引入库所 $p_i^I,p_{ie}^I,p_o^O,p_{dt}^O$ 分别描述设备的启动输入、休眠输入、执行输出、超时输出这 4 个接口,引入变迁 t_i,t_e,t_{ie},t_{fo} 分别表示设备的启动、完成、休眠和超时处理操作.设备中的操作 op_i 运行过程如下:首先,引入变迁 $t_{a,i}$ 刻画操作的执行,$ct(p_{a,i})=etp(op_i)$表示操作的执行时间,$en(t_{a,i})=enp(op_i)$;若引入库所 $p_{r,i}$ 用于标记操作的可执行,若设备发生超时或休眠,即$|M(p_{p,i})|\neq0$,则调用变迁 $t_{p,i}$,使得操作处于中断位置 $p_{p,i}$.变迁获得重新执行($p_{r,i}$),则调用变迁 $t_{r,i}$ 使得操作处于可用位置($p_{w,i}$).引入库所 p_c 用于

© 中国科学院软件研究所 http://www.jos.org.cn

标记设备处于运行位置,引入库所 p_d 用来控制设备的运行时限,因此,设置 $Ct(p_d)$为设备的动态运行时限.若时限截止时设备还未运行成功,则调用变迁 t_{fo} 输出超时故障 p_{dt}^O;若设备运行成功,则调用变迁 t_e 消除设备运行标记并输出结果到 p_o^O,同时,设备进入结束位置 p_e.

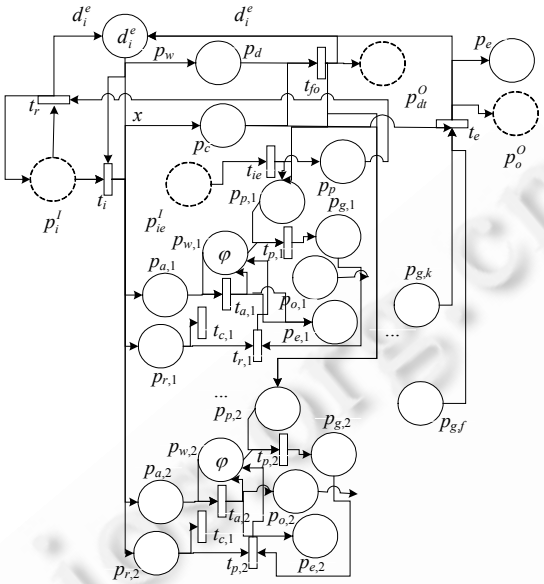

Fig.3 DE-Net model of device

图 3 设备的 DE-Net 模型

在 DES 能耗模型中,连接件 la_i 的 DE-Net 模型 Ω_{la_i} 如图 4 所示.设连接件对应的两个组件为 A 和 B,则对 A,B 分别引入两个接口用于接收和发送,对应库所描述组件的消息缓存,其中,库所 p_{po} 中存放连接件的解析策略.连接件接收到消息后触发变迁 t_{re},并按照 d_i^p 的规定来确定解析消息的策略.引入变迁 t_{se1},t_{se2},用于发送消息到目的端,$en(t_{se1})=en(t_{se1})=nl(la_i)$.

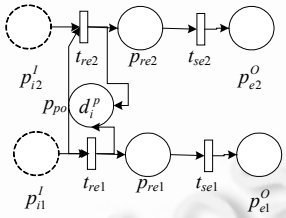

Fig.4 DE-Net model of connector

图 4 连接件的 DE-Net 模型

在 DES 能耗模型中,代理 ag_i 的 DE-Net 模型 Ω_{ag_i} 如图 5 所示,其中,深色框内表示代理内设备和连接件的执行,即 Agent 的环境.Agent 从环境中获取信息,根据自适应策略计算启动设备(即,对环境执行一定的控制).引入变迁 t_{st} 启动设备,而变迁 t_{ns} 使得其他设备进入休眠状态.变迁 t_e 表示 Agent 设备的结束.在收到 Agent 发出的指令后,设备会做出相应的调整和修改,若收到启动指令则设备进入运行位置.当设备收到休眠指令时,无论其处于等待执行、初始还是执行位置,设备都会调用相应的变迁进入休眠位置.若 Agent 下有设备运行超时,则启动超时处理,使得所有管辖的设备都进入休眠位置,并将超时信息上报系统.

© 中国科学院软件研究所 http://www.jos.org.cn

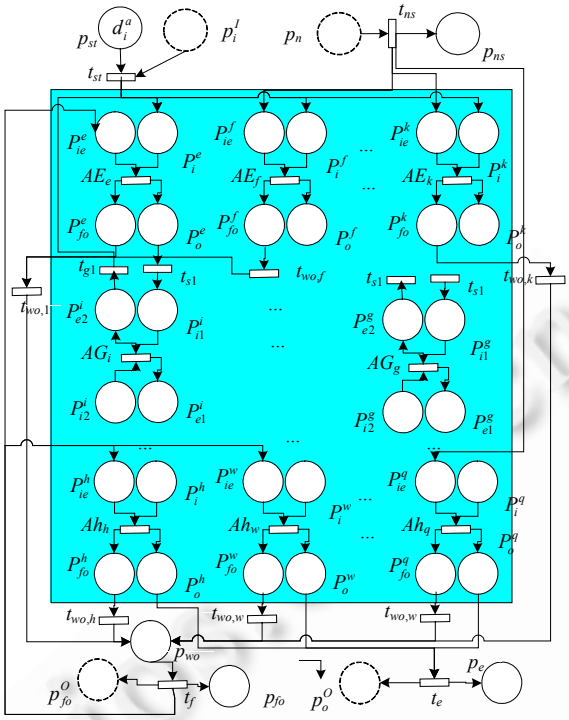

Fig.5 DE-Net model of agent

图 5 Agent 的 DE-Net 模型

下面根据 DES 的需求,构建 DES 的能耗模型.设需求模型Ξ的能耗模型Ω构造步骤如下:

(1) 根据各个设备的属性,构造所有设备的 DE-Net 模型.在上层模型中用 P_i^i,P_{ie}^i,P_o^i,P_{fo}^i 表示设备 ae_i 的启

(2) 动输入、休眠输入、输出、超时输出接口;进一步地,根据 Agent 与设备的关系建立 Agent 的能耗模型.

(3) 引入变迁 t_s 和库所 p_s 分别描述整个系统的开始操作和位置,对整个系统进行初始化;引入变迁 t_e 和库所 p_e 分别描述整个系统的结束操作和位置.依据 Agent 关系,通过对应连接件的模型来组合 Agent 模型,以形成整个系统的能耗模型.引入变迁 t_{dt}、库所 p_a,p_c,p_{dt} 描述系统的时限控制.

(4) 利用自适应能耗管理策略,依次计算启动设备集和设备的动态电压,并对模型中的库所进行初始化设置.

(5) 设置初始标识 $M_0(p_s)=\varphi$,同时,设置变迁 t_{dt} 和系统级别所有变迁的优先级为 4.本文将变迁优先级分为 5 个等级,可以根据实际需要适当地调整优先级的等级.

利用上述步骤,可以构建 DES 的能耗模型,为后续的系统分析和模型验证奠定基础.

5 基于模型的系统能耗值计算和验证

本节首先分析能耗模型上变迁和库所的实际含义,并根据分布式嵌入式系统基础组件属性及其关系,对组件的能耗值进行分析和计算;其次,基于构建好的能耗模型,将系统性质转化为模型上的一些 CTL 公式,并进一步验证方法的正确性和有效性.

5.1 系统能耗值计算

基于构建的能耗模型,从分布式嵌入式系统结构中的任务、设备、代理和系统这 4 个角度着手,给出具体

© 中国科学院软件研究所 http://www.jos.org.cn

的能耗值计算公式,并映射到模型.

由于 DES 任务采用不同设备完成功能,会直接导致对应能耗模型可达状态的不同.设任务 tk_i 在设备 ae_j 上通过调用操作 $OP(ae_j,tk_i)$执行,下面从能耗模型的角度对 DES 常见的能耗值进行计算和分析.

(1) 任务 tk_i 的能耗

从公式 $\frac{e_{V_{dd}}}{e_{V_{\max}}}=\frac{\alpha\times C_L\times f_{V_{dd}}\times V_{dd}^2}{\alpha\times C_L\times f_{V_{\max}}\times V_{\max}^2}=\frac{1}{etc}\times\frac{V_{dd}^2}{V_{\max}^2}$ 可得,任务 tk_i 对应的操作 $op_f\in OP(ae_j,tk_i)$的新单位能耗为

$$enp'(op_f)=enp(op_f)\times etc_i\times\frac{V_{dd}'^{i}}{V_{\max}^{j}}.$$

操作新单位能耗和运行时间主要表现为能耗模型中对应变迁和库所的属性.任务 tk_i 的功能是由操作集合 $OP(ae_j,tk_i)$中的所有操作共同完成的,因此,任务 tk_i 的能耗 $EN(tk_i)=\sum_{op_f\in OP(ae_j,tk_i)}enp'(op_f)\times etp'(op_f)$.

基于任务的能耗可以分析系统中消耗最大的任务和最小的任务,以了解系统整体能耗的分布,进而为能耗分析奠定数据基础.映射到模型中,为任务对应路径的能耗.

(2) 设备的能耗性能

设备 ae_j 的总能耗 $EN(ae_j)=\sum_{tk_i\in TK(ae_j)}EN(tk_i)$,则设备执行任务的平均能耗为

$$avg_EN(ae_j)=EN(ae_j)/|TK(ae_j)|.$$

分析系统中各个设备的总能耗和平均能耗,可以将节能分析进一步定位到总能耗和平均能耗较大的设备.设备的总能耗映射到能耗模型中,为设备对应可达路径上变迁的能耗之和.

(3) 代理和系统的能耗值

代理的能耗主要由其所管辖的设备和连接件的能耗组成,因此,代理 ag_j 的能耗值为

$$EN(ag_j)=\sum_{ae_j\in AE_{ac}\cap eg(ag_i)}EN(ae_j)+\sum_{la_j\in AE_{ac}\cap lg(ag_i)}nl(la_i);$$

系统的能耗值为

$$EN=\sum_{ag_i\in AG}nl(ag_i)=\sum_{ae_j\in AE_{ac}}EN(ae_j)+\sum_{la_j\in AE_{ac}}nl(la_i).$$

利用上述能耗指标,可以从多方面对 DES 的能耗进行定量分析.此外,还可以利用模型中资源的分布动态获得正在运行的设备和操作集合、已经运行结束的设备和操作集合、超时的设备等信息.

5.2 模型验证

计算树逻辑(computation tree logic,简称 CTL)是时态逻辑的一种.CTL 引入以下算子描述未来的不确定性: always(用 G 表示),sometimes(用 F 表示),next(用 X 表示),until(用 U 表示),all paths(用 A 表示),exists a path(用 E 表示).CTL 公式语法可以表示为(使用 BNF 范式描述):

$$\varpi::=\text{TRUE}|\text{FALSE}|pf|\neg\varpi|\varpi\wedge\varpi|\varpi\vee\varpi|\varpi\rightarrow\varpi|AX\varpi|EX\varpi|AF\varpi|EF\varpi|AG\varpi|EG\varpi|A(\varpi U\varpi)|E(\varpi U\varpi),$$

其中,pf 属于原子公式集合.本节基于分布式嵌入式系统能耗模型的可达状态集,利用 CTL 描述系统的性质,借助 Petri 网的操作语义验证方法的正确性和有效性.

设模型Ω是利用 DE-Net 模型对 DES 需求Ξ进行建模所得到的能耗模型,$R(\Omega)$表示对应的可达状态集,S 是Ω的一个可达状态,ϖ是一个使用 CTL 描述的公式:性质可以描述为$\Omega,S|=\varpi$,表示 S 状态满足公式ϖ.

能耗模型主要刻画 DES 的功能性需求,因此,有必要分析能耗模型能否正确描述嵌入式系统的执行流程.嵌入式系统的执行流程是由多个任务按照一定的关系组合而成,因此,可以从能耗模型能否正确描述任务间的关系着手.

定理 1. 设模型Ω是利用 DE-Net 模型对 DES 需求Ξ进行建模所得到的能耗模型,$R(\Omega)$是对应的可达状态集,$\forall S\in R(\Omega),\forall tk_i,tk_j\in TK$,设 $tk_i\in TK(ae_f),tk_j\in TK(ae_g)$,则有:

(1) 若 $RL(tk_i,tk_j)$=>且$\Omega,S|=\varpi_2$,则$\Omega,S|=AX\varpi_1$,其中,ϖ_1 为任务 tk_i 得到执行,ϖ_2 为任务 tk_j 得到执行;

© 中国科学院软件研究所 http://www.jos.org.cn

(2) 若 $RL(tk_i,tk_j)=+$,则$\Omega,S_0|=AG\varpi_3$,其中,ϖ_3 为$|M(ae_{f\bullet}p_{e,o})|+|M(ae_{g\bullet}p_{e,p})|\leqslant 1, op_o\in op(tk_i), op_p\in op(tk_j)$;

(3) 若 $RL(tk_i,tk_j)=||$,则$\Omega,S|=\varpi_4$,其中,ϖ_4 为$|M(ae_{f\bullet}p_{e,o})|=|M(ae_{g\bullet}p_{e,p})|=1\vee|M(ae_{f\bullet}p_{e,o})|=|M(ae_{g\bullet}p_{e,p})|=0$.

证明:

(1) 因为 $RL(tk_i,tk_j)=>$,若 $f=g$,即 tk_i,tk_j 由同一个设备 ae_f 完成,则从设备的建模过程可知,存在变迁 $ae_{f\bullet}t_{i,j}$,将任务 tk_i 的运行结果输出到 tk_j 的输入.因此,任务 tk_i 在状态 S 得到执行,则存在状态 $S'\in R(S)$,使得 $ae_{f\bullet}t_{i,j}\in ET(S)$.因此有$\Omega,S|=AX\varpi_1$.否则,若两个任务分别由不同的设备完成($f\neq g$),根据系统建模过程可知,会引入连接件 la_f用于完成任务 tk_i 到任务 tk_j 的交互.因此,任务 tk_i 在状态 S 得到执行,则存在状态 $S'\in R(S)$,使得连接件 la_f得到执行.同理,存在状态 $S'\in R(S)$,任务 tk_j 得到执行.因此有$\Omega,S|=AX\varpi_1$.

(2) 反证法,设子命题(2)不成立.即,$\exists S\in R(S_0)$,使得$|M(ae_{f\bullet}p_{e,o})|+|M(ae_{g\bullet}p_{e,p})|>1$,因为在任意一个设备模型中,库所 $p_{e,o}$ 保存操作 op_o 运行结束的标记,所以$|M(ae_{f\bullet}p_{e,o})|\leqslant 1, |M(ae_{g\bullet}p_{e,p})|\leqslant 1, |M(ae_{f\bullet}p_{e,o})|=|M(ae_{g\bullet}p_{e,p})|\leqslant 1$.因为 $RL(tk_i,tk_j)=+$,所以不存在某个状态使得两个任务同时执行,与上述假设推导结论矛盾,所以假设不成立.即,子命题(2)成立.

(3) 同理可证,子命题(3)成立. □

定理 1 说明:能耗模型可以正确地描述 DES 的业务执行流程,因此可以使用能耗模型在系统设计前期演示和分析 DES 的功能性需求.由于系统的节能主要通过自适应能耗管理策略来实现,因此有必要分析能耗模型对能耗管理策略的正确执行.能耗模型对启动和调整策略的正确执行,主要体现在根据计算好的启动设备集和动态单位能耗,实现设备的启动和休眠控制.

定理 2. 设模型Ω是利用 DE-Net 模型对 DES 需求Ξ进行建模所得到的能耗模型,$R(\Omega)$是对应的可达状态集, $\forall S\in R(\Omega),\forall ae_i\in AE$,系统的启动设备集 AE_{ac},则:

(1) 若 $ae_i\in AE_{ac}$,$\Omega,S|=\varpi_1$,其中,ϖ_1 为$|M(ae_{i\bullet}p_e)|=1$;

(2) 若 $ae_i\notin AE_{ac}$,$\Omega,S|=\varpi_2$,其中,ϖ_2 为$|M(ae_{i\bullet}p_p)|=1$.

证明:

(1) 若 $ae_i\in AE_{ac}$,不妨设 $ae_i\in eg(ag_j)$,即 ae_i 属于 Agent 的管辖范围.

在 ag_j 建模过程中会设置: $ag_{j\bullet}P_i^i\in ag_{j\bullet}t_{st}^{\bullet}$,所以存在 $S_1\in R(S_0)$,使得 $M_1(ag_{j\bullet}p_i^I)\neq\varnothing$且 $M_1(ag_{j\bullet}p_{ie}^I)=\varnothing$,因为 $M_0(ag_{j\bullet}p_{st})=(d_j^a,1)$且 $ag_{j\bullet}p_{st}^{I\bullet}=ag_{j\bullet}t_{st}$,所以 $M_1(ag_{j\bullet}p_{st})=(d_j^a,1)$.因为$^{\bullet}ag_{j\bullet}t_{st}=\{ag_{j\bullet}p_i^I, ag_{j\bullet}p_{st}\}$,$ag_{j\bullet}p_i^{I\bullet}=ag_{j\bullet}t_{st}$,所以存在 $S_2\in R(S_1)$,使得 $M_2(ag_{j\bullet}P_i^i)=(\varphi,1)$,即,$|M(ae_{i\bullet}p_i^I)|=1$.从设备的建模过程可知,变迁 $ae_{j\bullet}t_i$ 可以有效触发,而 $ae_{j\bullet}t_{ie}$ 没有发生权,所以,设备会触发变迁进入 $ae_{j\bullet}t_i$ 运行位置,进一步地,执行各个操作到达结束位置($ae_{i\bullet}p_e$),因此$\Omega,S|=\varpi_1$,其中,ϖ_1 为$|M(ae_{i\bullet}p_e)|=1$.

(2) 同理,从 Agent 对设备执行休眠操作的建模过程可证子命题(2)成立. □

定理 2 说明,能耗模型可以正确地描述 Agent 与设备之间的交互,包括获得设备信息、Agent 启动和休眠设备.在实际运行中,设备的启动和休眠不一定是固定不变的,比如,设备 ae 在某次运行中需要启动,在新的环境中(系统设备分布发生改变时)可能需要休眠.因此,有必要分析在能耗模型中设备被休眠后是否可以被重新启动,已经运行的设备是否可以被中断.

定理 3. 设模型Ω是利用 DE-Net 模型对 DES 需求Ξ进行建模所得到的能耗模型,$R(\Omega)$是对应的可达状态集, $\forall S\in R(\Omega),\forall ae_i\in AE$,则:

(1) 若$|M(ae_{i\bullet}p_p)|=1$,则$\Omega,S|=\varpi_1$,其中,ϖ_1 为$|M(ae_{i\bullet}p_e)|=1$;

(2) 若$|M(ae_{i\bullet}p_{\mathrm{p},j})|=1$,$\Omega,S|=\varpi_2$,其中,$\varpi_2$ 为$|M(ae_{i\bullet}p_{g,j})|=1$.

证明:

(1) $\forall ae_i\in AE$,因为$|M(ae_{i\bullet}p_p)|=1$,所以在状态 S 下,设备 ae_i 处于休眠位置.设经过一段时间后,ae_i 得到重新启动,即,存在 $S_1\in R(S)$有$|M_1(ae_{i\bullet}p_i^I)|=1$.下面证明 ae_i 的休眠不会影响再次启动,主要从两个方面证明:

首先证明 ae_i 可以重新进入运行位置,即,调用操作.因为 $ae_{i\bullet}p_i^{I\bullet}=\{ae_{i\bullet}t_i, ae_{i\bullet}t_r\}$,$^{\bullet}ae_{i\bullet}t_r=\{ae_{i\bullet}p_i^I, ae_{i\bullet}p_p\}$,所以

© 中国科学院软件研究所 http://www.jos.org.cn

$ae_i\bullet t_r$ 具有发生权,且 $ae_i\bullet t_r$ 的优先级为 0.所以状态 S_1 下,变迁 $ae_i\bullet t_r$ 的触发是有效的.不妨设 S_1 通过触发变迁 $ae_i\bullet t_r$ 到达状态 S_2,则有 $|M_2(ae_i\bullet p_w)|=|M_1(ae_i\bullet p_i^I)|=1$,因为 ${}^\bullet ae_i\bullet t_i=\{ae_i\bullet p_i^I, ae_i\bullet p_w\}$,所以 $ae_i\bullet t_i$ 在状态 S_2 下的触发是有效的,即,ae_i 可以重新进入运行位置.

其次证明所有的操作都可以正常执行,$\forall op_f\in op(ae_i)$ 在 ae_i 被休眠时刻可能处于运行位置 $p_{a,f}$ 或结束位置 $p_{e,f}$.因为 ${}^\bullet ae_i\bullet p_p=ae_i\bullet t_{ie}, ae_i\bullet t_{ie}^\bullet=\{ae_i\bullet p_p, ae_i\bullet p_{p,1}, ae_i\bullet p_{p,2},\ldots,ae_i\bullet p_{p,n}\}$,所以在状态 S 下,有:

$$|M(ae_i\bullet p_{p,1})|=|M(ae_i\bullet p_{p,2})|=\ldots=|M(ae_i\bullet p_{p,n})|=|M(ae_i\bullet p_{p,n})|=|M(ae_i\bullet p_{p,n})|=1.$$

因为 $\forall op_f\in op(ae_i)$ 在状态 S 下有 $|M(ae_i\bullet p_{w,f})|=1$,${}^\bullet ae_i\bullet t_{p,f}=\{ae_i\bullet p_{w,f}, ae_i\bullet p_{p,f}\}$,且变迁 $ae_i\bullet t_{p,f}$ 的优先级等于 0,所以 $ae_i\bullet t_{p,f}$ 在 S 下可以有效触发.不妨设 S 通过有效触发 $ae_i\bullet t_{p,f}$ 到达状态 S_3,因为 $ae_i\bullet t_{p,f}{}^\bullet=ae_i\bullet p_{g,f}$,所以 S_3 下有 $|M_3(ae_i\bullet p_{g,f})|=1$,操作可以正常执行.即,$ae_i$ 通过执行一些操作后,可能到达结束位置 p_e.因此,$\Omega,S|=\varpi_1$,其中,ϖ_1 为 $|M(ae_i\bullet p_e)|=1$.

(2) 同理可以证明子命题(2)成立. □

定理 3 说明:当 Agent 需要重新启动某个已经被休眠的设备时,设备及其操作都可以正常执行,完成相应的功能.同时,当操作处于运行位置时,对应设备得到休眠指令,操作可以被挂起,进入到中断位置.在 DES 能耗建模过程中,自适应能耗管理策略是核心部分,因此,有必要分析自适应策略的有效性.

定理 4. 设 DES 需求为 Ξ,利用自适应策略计算出系统的启动设备集为 AE_{ac},任务 tk_i 的运行时间和供电电压分别为 tc_i, V_{dd}^i,则:

(1) 任意一个可以完成系统功能的设备集为 AE_f,则有 $EN(AE_{ac})\leqslant EN(AE_f)$;

(2) 在启动设备集 AE_{ac} 下,调压后的系统能耗小于等于调压前的能耗.

证明:

(1) 反证法:设存在设备集 AE_f,利用 AE_f 可以完成系统功能且 $EN(AE_{ac})>EN(AE_f)$.根据假设有两种情形:

- 情形 1:存在设备 $ae_i\in AE_{ac}$ 可以完成 $ae_j\in AE_f$ 的功能,其余设备相同.根据假设有 $EN(ae_i)>EN(ae_j)$.基于设备间关系可得,若设备 ae_i 和 ae_j 的距离在阈值范围内,则 ae_j 完全覆盖 ae_i.根据启动策略可知,系统会选择 ag_j 替代 ae_i.所以假设不成立,否则,若设备 ae_i 和 ae_j 的距离不在阈值范围内,则虽然 $EN(ae_i)>EN(ae_j)$,但是由于 ae_j 的地理位置比较远,所以选择 ae_j 会导致传输能耗变大,最终执行的能耗增加.所以假设不成立.
- 情形 2:存在设备集 $ae_{j,1},ae_{j,2},\ldots,ae_{j,k}$ 可以完成 ae_i 的功能,但是 $EN(ae_i)>EN(ae_{j,1})+EN(ae_{j,2})+\ldots+EN(ae_{j,k})$.根据分布式嵌入式系统可知,设备的启动和休眠的能耗差距比较大,因此,将多个任务分散到不同设备执行比将任务集中在一个设备上能耗大.因情形 2 与实际相悖,所以假设不成立.

综上所述,若任意一个可以完成系统功能的设备集为 AE_f,则有 $EN(AE_{ac})\leqslant EN(AE_f)$.

(2) 设 op_f 是系统执行过程中调用的一个操作,op_f 的新能耗为

$$en'(op_f)=enp'(op_f)\times etp'(op_f)=\left(enp(op_f)\times\frac{1}{etc_i}\times\frac{V_{dd}'^i}{V_{\max}^j}\right)\times(etp(op_f)/etc_i)$$

$$=enp(op_f)\times etp(op_f)\times\frac{1}{etc_i^2}\times\frac{V_{dd}'^i}{V_{\max}^j}=en(op_f)\times\left(\frac{tc'}{tc}\right)^2\times\frac{V_{dd}'^i}{V_{\max}^j}\leqslant en(op_f).$$

即,通过电压调整操作的能耗得到降低.系统的整体能耗等于执行的各操作能耗之和,因此,调压后的系统能耗小于等于调压前的能耗. □

定理 4 说明:通过自适应能耗的两个子策略,DES 的整体能耗可以逐步得到降低.

6 应用案例

本节利用上述煤矿监控系统来说明方法的有效性和正确性,并通过 4 个仿真实验来分析能耗管理策略的适用范围和有效性.

© 中国科学院软件研究所 http://www.jos.org.cn

根据实际需要,每个人身上的射频卡都是必须启动的,从射频卡的完成功能可以发现,这些射频卡都是无关的,所以,采用启动策略不会休眠射频卡.下面采用启动策略计算人员定位系统的启动读卡器集.具体步骤如下:

Step 1. 根据作用集定义计算 AE_{now} 中读卡器 ae_2^r 的作用集为 $VW(ae_2^r,TK)=\{tk_1,tk_{1,3},tk_{2,3},tk_{3,3},tk_{4,3},tk_{5,3},tk_{21,3},tk_{23,3},tk_2\}$.同理可以计算读卡器 $ae_3^r,ae_4^r,ae_6^r,ae_7^r,ae_8^r,ae_9^r,ae_{10}^r,ae_{11}^r,ae_{12}^r,ae_{13}^r,ae_{14}^r,ae_{15}^r,ae_{16}^r$ 的作用集,而其余设备的作用集均为 0.排序后,$AE_{now}=\{ae_3^r,ae_7^r,ae_2^r,ae_6^r,ae_8^r,ae_{14}^r,ae_9^r,ae_{11}^r,ae_4^r,ae_{13}^r,ae_{16}^r,ae_{10}^r,ae_{12}^r,ae_{15}^r\}$.根据启动策略(1)计算候选备用集 $AE_{now}=\{ae_3^r,ae_7^r,ae_6^r,ae_8^r,ae_{14}^r,ae_9^r,ae_{11}^r,ae_4^r,ae_{13}^r,ae_{16}^r,ae_{12}^r\}$.

Step 2. 利用启动策略(2)逐步计算,以得到 $AE_{ac}=\{ae_3^r,ae_7^r,ae_6^r,ae_{14}^r,ae_8^r,ae_9^r,ae_{11}^r,ae_{13}^r,ae_4^r,ae_{16}^r,ae_{12}^r\}$.启动设备 ae_3^r,ae_7^r 的执行任务集分别为 $TK(ae_3^r)=\{tk_1,tk_{1,3},tk_{2,3},tk_{3,3},tk_{4,3},tk_{5,3},tk_{6,3},tk_{7,3},tk_{8,3},tk_{21,3},tk_{23,3},tk_{25,3},tk_{27,3},tk_2\}$,$TK(ae_7^r)=\{tk_1,tk_{14,3},tk_{15,3},tk_{16,3},tk_{17,3},tk_{22,3},tk_{24,3},tk_{35,3},tk_{36,3},tk_2\}$.同理可以获得启动设备 $ae_4^r,ae_6^r,ae_8^r,ae_9^r,ae_{11}^r,ae_{12}^r,ae_{13}^r,ae_{14}^r,ae_{16}^r$ 的执行任务集.

针对读卡器分析发现:随着人员的移动,读卡器间的关系可能会发生改变.人员越密集,读卡器的使用率就越高.而分站和地面监控器间关系都是无关的,所以不会对其进行休眠.分析人员定位系统发现,执行路径的最大运行时间等于 21,所以还有 9 个空闲时间用于调整.按照调整策略计算射频卡、读卡器和分站上各个操作的新运行时间、供电电压分别为(1.43,3.35),(1.43,3.35),(2.8,3.35),(1.43,4.57),(1.43,4.57),(2.85,4.57),(1.43,5.45),(2.85,5.45),(1.43,5.45),(2.85,5.45).利用能耗模型的构建步骤,可以建立人员定位的能耗模型,如图 6 所示.

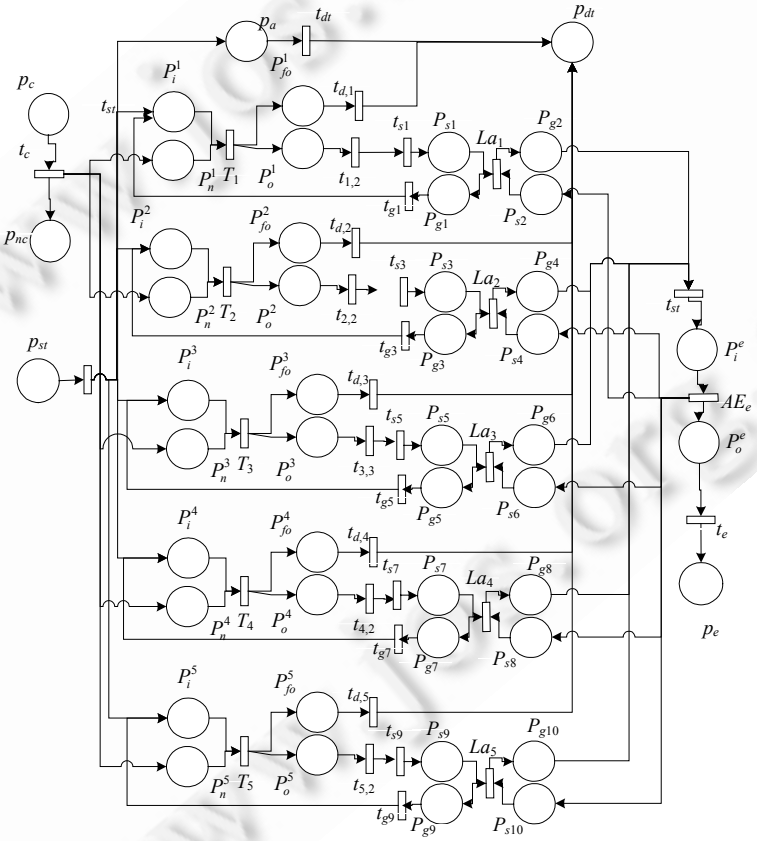

Fig.6 Energy model of personnel oriented system

图 6 人员定位系统的能耗模型

利用 Petri 网相关工具验证模型的相应性质,包括执行流程的正确性、Agent 与设备间交互的正确性、启动

© 中国科学院软件研究所 http://www.jos.org.cn

策略和调整策略等性质.同理,可以对煤矿监控系统进行建模和分析.

为了有效评估方法的有效性,本文设计了仿真实验以评估方法的效果,藉以说明方法的实用性.首先,随机产生 700 个射频卡(主要需要随机生成射频卡的位置)、3 000 个读卡器、10 个分站、1 个地面监控器作为 DES 的物理资源.随机生成 960 个任务构成分布式嵌入式系统的任务资源.每个设备都包含名称、位置、供电电压参数等属性.每个任务至少包含任务名称、任务功能、可用设备等基本信息.同时,如果没有特殊说明,下面仿真中的矿井系统均包含 10 个任务、30 个射频卡、200 个读卡器.

实验 1 的目的是说明采用启动策略的有效性.其具体实验步骤如下:

(1) 取 400 个射频卡、260 个读卡器、50 个分站、10 个地面监控器分为 10 组,每组对应为一个矿井系统,依次对每个矿井软件进行步骤(2)、步骤(3).

(2) 根据系统的执行流程,选择 10 组设备集来完成系统的功能,基于各设备集计算出每个软件的整体能耗.

(3) 利用启动策略计算系统的启动设备集,基于启动设备集计算出每个软件的整体能耗.

实验 1 的实验结果如图 7(a)所示,分析该图可知:

(1) 设备集的能耗会随着系统设备的属性不同而发生改变,如 $s2$ 能耗明显较低,而 $s3$ 在所有设备集下的能耗均较高.

(2) 采用启动策略后,系统的整体能耗得到降低,最高情况下降低了 34.5%,最低降低了 3%.分析最低情况发现:在该组系统中,射频卡的分布比较散,所以启动设备只比其中一组设备减少了 1 个设备,所以效果比较差.

综上所述,在人员集中情况下使用启动策略的效果会比较明显.

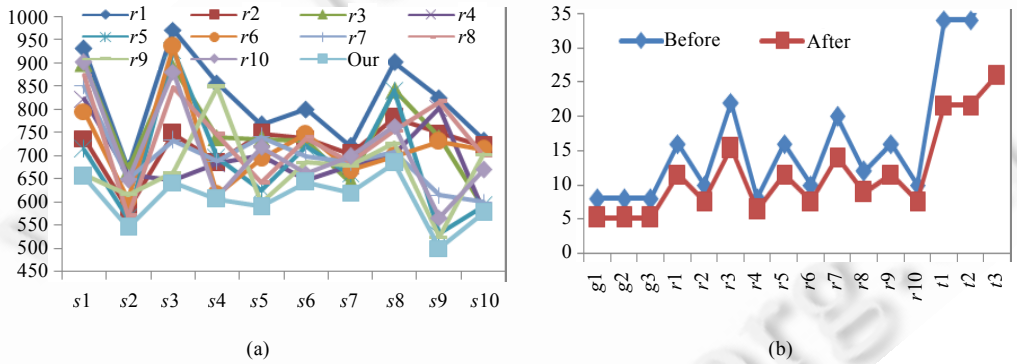

Fig.7 Simulation results

图 7 仿真结果

实验 2 的目的是说明采用能耗管理策略的有效性.其具体实验步骤如下:

(1) 取 40 个射频卡、26 个读卡器、5 个分站、1 个地面监控器组成一个人员定位系统.

(2) 采用启动策略计算该系统的启动设备集合.

(3) 分别计算着启动设备的调压前能耗和调压后能耗.

实验 2 的实验结果如图 7(b)所示,其中,射频卡仅计算了 3 个:$g1$,$g2$,$g3$,读卡器主要启动 10 个,而分站也启动 3 个.通过分析可知:

(1) 虽然调整策略通过调节射频卡的动态电压可以降低射频卡的能耗,但每个射频卡的能耗是一样的.

(2) 调整策略在读卡器上的效果比较明显,但是不同读卡器的效果不一样.分析其原因,主要是由于读卡器工作范围内的射频卡数量有差异造成.

(3) 同样地,调整策略对于分站也具有一定的效果.

© 中国科学院软件研究所 http://www.jos.org.cn

实验 3 的目的是分析能耗模型状态空间与系统设备增加的关系.其具体实验步骤如下:

(1) 取 20 个射频卡、26 个读卡器、5 个分站、1 个地面监控器组成一个人员定位系统的基本结构.

(2) 系统以 10 为单位增加射频卡,即 30,40,50,60,70,80,90,然后采用启动策略计算该系统的启动设备集和能耗模型的可达状态个数.

(3) 系统以 5 为单位增加读卡器,即 26,31,36,41,46,51,56,然后采用启动策略计算该系统的启动设备集和能耗模型的可达状态个数.

(4) 以 1 为单位增加分站,即 6,7,8,9,10,11,12,根据读卡器的位置平均分布分站,然后采用启动策略计算该系统的启动设备集和能耗模型的可达状态个数.

(5) 以 1 为单位增加 Agent,即 26,31,36,41,46,51,56,根据读卡器的位置平均分布 Agent,然后采用启动策略计算该系统的启动设备集和能耗模型的可达状态个数.

从实验 3 的计算结果可以看出:所有情况下,能耗模型的状态空间均为 32,即,增加射频卡、读卡器、分站和 Agent 对能耗模型的状态空间均无影响.分析其原因可知:能耗模型在执行过程中,通过引用最大触发集来计算可达状态,而系统中设备都是并行运行的,所以已有设备个数增加不会影响模型的状态空间.比如,增加射频卡虽然系统可能需要启动更多的读卡器,但是由于读卡器之间是并行执行,所以能耗模型的状态个数仍然保持不变.若系统此时增加一个转换器(有 3 个顺序执行的操作),则可以发现,系统的模型状态空间会增加到 37.所以,增加设备类型可以导致能耗模型的状态空间增加,增加幅度由设备的操作及其关系决定.通过实验 3 可知,本文所提出的模型可以适合大型的 DES 的能耗建模与分析.

实验 4 的目的是分析能耗模型状态空间与系统设备故障或休眠的关系,具体实验步骤如下:

(1) 取 40 个射频卡、26 个读卡器、5 个分站、1 个地面监控器,对应为一个矿井系统.

(2) 随机选择 5 个射频卡,系统在运行过程中需要临时将这些射频卡置于临时休眠,然后,采用启动策略计算该系统的启动设备集和能耗模型的可达状态个数.重复 10 次第(2)步,因为每次休眠的射频卡位置可能不同.

(3) 设系统目前射频卡位置随机移动一次位置,计算系统新的启动设备集和能耗模型的可达状态个数.

实验 4 的计算结果可以看出:当系统在运行过程中需要临时改变设备的状态时,会导致能耗模型状态空间的增加.虽然休眠的射频卡位置不同会造成系统的启动设备集不同,但是能耗模型的可达状态个数一直是 34,与休眠射频卡的位置无关.而当射频卡位置发生改变时,可以发现,启动设备集的个数会动态改变,但是能耗模型的可达状态个数没有发生改变,仍然是 34.

7 结束语

随着计算机软硬件的迅速发展与网络的广泛应用,越来越多的 DES 呈现出自适应特征.它们代表了软件形态和复杂性特征的变化,从而对支持这类系统开发的软件工程技术提出了新的要求和挑战.如何动态地构建和分析 DES 自适应能耗,也逐渐成为软件界的研究热点.与现有工作相比,应用本文所提出的方法可以达到如下效果:

(1) 给出分布式嵌入式系统的自适应能耗管理策略.该策略根据当前设备的分布和系统的业务流程,动态地计算启动设备集.在确保系统功能性需求的前提下,通过减少启动设备来达到降低系统的能耗.同时,根据系统的运行时限及设备运行操作的属性,动态地计算操作的运行时间和供电电压,以降低启动设备的能耗值,从而降低系统整体能耗.仿真实验结果表明,能耗管理策略可以有效降低系统的整体能耗.

(2) 正确刻画了分布式嵌入式系统动态和静态特性,采用 Petri 网对分布式嵌入式系统的设备、连接件、Agent 及其交互进行建模,所构造的模型可以刻画分布式嵌入式系统业务流程.在建模过程中,通过构建基础组件的模型库,可以根据实际需求对 Agent、设备、连接件进行增减,并动态地织入能耗管理策略,提高了能耗模型的可扩展性.

© 中国科学院软件研究所 http://www.jos.org.cn

(3) 利用 CTL 将系统的性质转换为公式,基于能耗模型的执行语义分析系统各部分的能耗值,并从理论上证明策略和模型的正确性.

基于 Agent 的软件开发具有易理解、更加模块化以及更易于复用等特性.目前,我们正着手完善 Petri 网及其扩展的形式化语法和语义,并进一步研究 DES 的行为特征.同时,开发相应的支撑工具,形成一套更具有实际应用价值的分析方法.

References:

[1] He C, Zhu XM, Guo H, Qiu DS, Jiang JQ. Rolling-Horizon scheduling for energy constrained distributed real-time embedded systems. Journal of Systems and Software, 2012,85(4):780−794. [doi: 10.1016/j.jss.2011.10.008]

[2] Dam T, Koen L. An adaptive energy-efficient MAC protocol for wireless sensor networks. In: Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems. Los Angeles: ACM, 2003.171−180. [doi: 10.1145/958491.958512]

[3] Antonio C, Massimo C, Salvatore G, Seidita, V. Agent-Oriented software patterns for rapid and affordable robot programming. Journal of Systems and Software, 2010,83(4):557−573. [doi: 10.1016/j.jss.2009.10.035]

[4] Girault C, Valk R. Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications. Berlin: Springer-Verlag, 2003.

[5] Liu XB, Guo B, Shen Y, Xiong B, Wang JH, Wu YS, Liu YB. Embedded software energy modeling method at architecture level. Ruan Jian Xue Bao/Journal of Software, 2012,23(2):230−239 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4145.htm [doi: 10.3724/SP.J.1001.2012.04145]

[6] Machado K, Rosário D, Cerqueira E, Loureiro A, Neto A, de Souza J. A routing protocol based on energy and link quality for Internet of things applications. Sensors, 2013,13(2):1942−1964. [doi: 10.3390/s130201942]

[7] Senn E, Lanrem J, Juin E, Dignet JP. Refining power consumption estimations in the component based AADL design flow. In: Proc. of the IEEE Conf. on Specification, Verification and Design Language. IEEE, 2008. 173−178. [doi: 10.1109/FDL.2008.4641441]

[8] Acharya S, Mahapatra R. A dynamic slack management technique for real-time distributed embedded systems. IEEE Trans. on Computers, 2008,57(2):215−230. [doi: 10.1109/TC.2007.70789]

[9] Mahapatra RN, Zhao W. An energy-efficient slack distribution technique for multimode distributed real-time embedded systems. IEEE Trans. on Parallel and Distributed Systems, 2005,16(7):650−662. [doi: 10.1109/TPDS.2005.78]

[10] Sun ZW, Liu CH, Bisdikian C, Branch JW, Yang B. QoI-Aware energy management in Internet-of-things sensory environments. In: Proc. of the 9th Annual IEEE Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks. IEEE, 2012. 19−27.

[11] Mahkameh Y, Ardeshir B. A context-aware adaptive learning system using agents. Expert Systems with Applications, 2011,38(4):3280−3286. [doi: 10.1016/j.eswa.2010.08.113]

[12] Liang X, Des G. Agent model: software adaptivity using an agent-oriented model-driven architecture. Information and Software Technology, 2009,51(1):109−137. [doi: 10.1016/j.infsof.2008.02.002]

[13] Chang L, He XD, Shatz SM. A methodology for modeling multi-agent systems using mested Petri nets. Int'l Journal of Software Engineering and Knowledge Engineering, 2012,22(7):891−926. [doi: 10.1142/S0218194012500246]

[14] Lian J, Shatz S, He X. Flexible coordinator design for modeling resource sharing in multi-agent systems. Journal of Systems and Software, 2009,82(10):1709−1729. [doi: 10.1016/j.jss.2009.04.054]

[15] Li LX, Jin Z, Li G. Modeling and verifying services of Internet of things based on timed automata. Chinese Journal of Computers, 2011,34(8):1365−1377 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011.01365]

[16] Zhang TT, Wu X, Li CD, Dong YW. On energy-consumption analysis and evaluation for component-based embedded swith CSP. Chinese Journal of Computers, 2009,32(9):1876−1883 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2009.01876]

[17] Lanese I, Bedogni L, Felice M. Internet of things: A process calculus approach. In: Proc. of the 28th Annual ACM Symp. on Applied Computing. ACM, 2013. 1339−1346. [doi: 10.1145/2480362.2480615].

© 中国科学院软件研究所 http://www.jos.org.cn

[18] Fan GS, Yu HQ, Chen LQ, Liu DM. Strategy driven modeling and analysis of reliable embedded systems. Ruan Jian Xue Bao/ Journal of Software, 2011,22(6):1123−1139 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4026.htm [doi: 10. 3724/SP.J.1001.2011.04026]

附中文参考文献:

[5] 刘啸滨,郭兵,沈艳,熊冰,王继禾,伍元胜,刘云本.嵌入式软件体系结构级能耗建模方法.软件学报,2012,23(2):230−239. http://www.jos.org.cn/1000-9825/4145.htm [doi: 10.3724/SP.J.1001.2012.04145]

[15] 李力行,金芝,李戈.基于时间自动机的物联网服务建模和验证.计算机学报,2011,34(8):1365−1377. [doi: 10.3724/SP.J.1016.2011.01365]

[16] 张滕滕,吴晓,李长德,董云卫.基于 CSP 的构件化嵌入式软件能耗分析与评估方法研究.计算机学报,2009,32(9):1876−1883. [doi: 10.3724/SP.J.1016.2009.01876]

[18] 范贵生,虞慧群,陈丽琼,刘冬梅.策略驱动的可靠嵌入式系统建模及分析方法.软件学报,2011,22(6):1123−1139. http://www.jos.org.cn/1000-9825/4026.htm [doi: 10.3724/SP.J.1001.2011.04026]

范贵生(1980－),男,上海人,博士,助理研究员,主要研究领域为软件工程,面向服务计算,形式化方法.
E-mail: gsfan@ecust.edu.cn

陈丽琼(1982－),女,博士,副教授,主要研究领域为嵌入式系统,形式化方法.
E-mail: lqchen@sit.edu.cn

虞慧群(1967－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程,可信计算与安全,形式化方法.
E-mail: yhq@ecust.edu.cn

刘冬梅(1970－),女,博士,高级工程师,主要研究领域为软件工程,面向服务计算.
E-mail: dmliu@ecust.edu.cn

© 中国科学院软件研究所 http://www.jos.org.cn