

基于时间自动机的嵌入式系统调度分析工具

于 淼¹, 李 允², 桂盛霖², 罗 蕾²

(1. 西南交通大学信息科学与技术学院, 成都 610031; 2. 电子科技大学计算机科学与工程学院, 成都 610054)

摘 要: 为验证嵌入式实时系统开发过程中任务集的可调度性, 设计并实现一种嵌入式系统调度分析工具。提出通用任务模型, 建立任务与事件到达自动机和任务状态自动机的状态关系映射, 利用基于模型检测的时间自动机可达性方法判定系统的可调度性。仿真实例结果表明, 该工具的分析准确性较高。

关键词: 形式化方法; 时间自动机; 可调度性; 嵌入式实时系统; 任务模型

Schedule Analysis Tool for Embedded System Based on Timed Automata

YU Miao¹, LI Yun², GUI Sheng-lin², LUO Lei²

(1. School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, China;

2. School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

【Abstract】 In order to verify the schedulability property of the task set in embedded real-time system, this paper designs and implements a schedulability analysis tool. It abstracts general task model in the first place and defines the logical mapping of task models in system to states of two types of timed automata. Based on model checking theory, this tool determines whether this system can be scheduled by timed automata reachability method, and finally tests the accuracy of that method through two simulation examples.

【Key words】 formalization method; timed automata; schedulability; embedded real-time system; task model

DOI: 10.3969/j.issn.1000-3428.2012.03.095

1 概述

嵌入式实时系统已被广泛应用于涉及安全与正确性的关键领域。开发这类系统的成本较高, 如果能够在系统开发过程中越早发现问题, 就可以避免很多人力物力资源的浪费。

由于不同的设计需求, 嵌入式实时系统的开发存在不同的度量标准, 如资源消耗、代码规模、内存预算、成本控制等。特别是对实时属性的关注, 使得在开发过程中都会提供一个详细的系统描述和性能评估。实时性的含义要求系统对外部事件的响应和任务的执行都必须在限定的时间内完成。为满足在网络和多处理器等复杂环境下嵌入式系统中任务执行的实时要求, 可调度性的验证则极为重要。通过这样的保证, 设计者可以确定目标嵌入式系统任务在不超过截止时间的前提下准确实现。为此, 本文提出一种基于时间自动机的嵌入式系统调度分析工具, 用于验证嵌入式系统的可调度性。

2 系统模型

嵌入式实时系统功能性与非功能属性的日益扩展, 导致了设计复杂度的爆炸式增加。与此同时, 软件与硬件协同在较高系统层次的抽象方法也得到广泛研究。基于软件工程的思想, 建模是一个可以定制结构化系统模板并且提供快速验证的方法。它在保证系统正确性和合理性的前提下, 提高了开发效率和能力。

2.1 抽象模型需求

抽象模型是嵌入式实时系统模型的核心体现, 它具备可论证性, 不包含具体的实现细节。不同的模型反映不同的系统层次抽象, 同时也代表系统中各个元素的不同视角。为了突出对系统可调度性的研究, 多数的模型都包含以下内容:

(1) 抽象各种软件组件和硬件组件作为任务单元; (2) 分派通信

和计算资源以满足任务的时间限定; (3) 预测各个任务模型的最坏响应时间以及端到端延迟; (4) 对整个系统模型的分析结果做出判定。

2.2 任务模型和资源需求

任务^[1]是嵌入式实时系统的执行单元。系统详细指定每个任务资源的分配。在抽象模型中, 任务按照分布情况被分为两类, 内部事件任务和外部事件任务, 前者是普通任务被绑定在处理器内部, 后者表示处理器外系统环境中的任务。为了满足嵌入式实时系统的稳定性和处理能力, 本文在设计工具时考虑对多处理器环境和任务同步触发的支持, 但不涉及处理器的细节结构, 如算术逻辑单元(Arithmetic Logic Unit, ALU)、寄存器和控制器, 而把处理器看做一个整体的组件。在 2.1 节通用抽象模型需求的基础上, 任务模型还应包含如下详细属性:

(1) 任务优先级约束: 根据具体的调度策略, 任务通过可抢占或不可抢占的方式获得不同级别处理器使用权限。

(2) 任务执行时间: 任务需要的处理器运行时间。

(3) 调度策略: 不同的任务集在处理器上的调度规则, 分为静态调度和动态调度, 静态调度如单调速率调度(Rate-monotonic Scheduling, RMS) 算法; 动态调度如最早截止时间优先(Earliest Deadline First, EDF)算法。

基金项目: 国家自然科学基金资助项目(90718019); 国家“863”计划基金资助项目(2007AA010304)

作者简介: 于 淼(1985—), 男, 硕士, 主研方向: 嵌入式系统, 实时软件建模; 李 允, 副教授、博士; 桂盛霖, 博士; 罗 蕾, 教授、博士生导师

收稿日期: 2011-07-15 **E-mail:** miao-miaoyu@163.com

(4)缓冲队列: 将事件消息存放在缓冲区域, 待调度器实时读取队首事件, 实例化对应任务至就绪状态。

(5)偏移量: 任务到达时刻的初始相位。

(6)端到端延时: 指从某个模块的输入端到一个输出端所经历的时间。

(7)相对截止时间: 每个任务必须在该时间限前完成, 取决于软硬实时系统的不同抽象需求。

(8)周期性任务: 周期性任务意味着任务到达释放是周期性时刻, 通常认为周期值与截止时间相等。

(9)非周期任务: 任务的到达时间是不确定的, 可能很长, 也可能任务根本不到达。

(10)零星性任务: 它的属性介于周期任务与非周期任务之间, 虽然该任务也是不确定时刻到达, 但是要求它的最小到达时间间隔为确数。

(11)依赖关系: 在单处理器内部任务或多处理器间任务中存在的多种复杂的触发和约束关系。

2.3 系统模型实例

在嵌入式实时系统中, 普遍会分布多个任务, 任务间也有复杂的依赖关系, 图 1 为一个典型系统实例。该系统模型有 7 个任务, 其中, 任务 1~任务 4 绑定在处理器 1 上使用同一种调度策略 EDF, 而任务 5 和任务 6 绑定在处理器 2 上使用同一种调度策略 RMS, 任务 7 则是外部事件任务, 与任务 4 有外部事件触发关系。在处理器 1 内部, 任务 3 对任务 2 有内部事件触发关系。同时, 模型中处理器 1 与处理器 2 间还存在任务 1 对任务 5 的依赖关系。

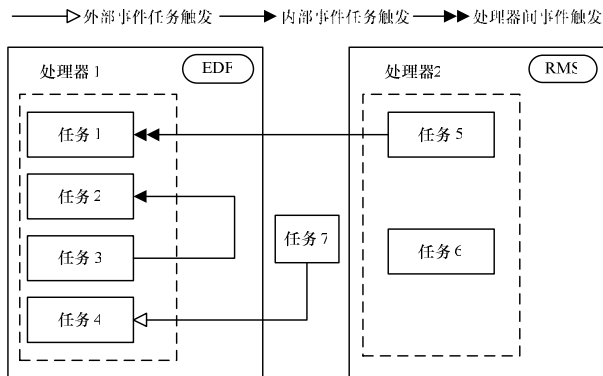


图 1 系统模型实例

3 工具设计

目前的系统性能分析方法大致可分成 2 类: 仿真方法和形式化分析方法。

(1)基于仿真的方法的优点如下: 1)可将结果直接反馈给用户, 通过随时改变模型的特征和参数, 用户可以直观追踪系统中任务状态变化的过程。2)仿真方法可以适用范围较广, 如系统组件间存在复杂的交互, 这种情况很难被抽象并建立数学模型, 却可以通过动态仿真来准确表达。然而, 仿真技术的主要缺点是不能满足可能性的完全覆盖, 尤其是系统复杂度较高的情况下, 会存在边界问题以及许多不确定行为, 使得用户在有限的仿真时间内不易找到精确的性能最好与最坏阈值。当仿真时刻停止时, 对目标系统的测试过程也随之停止, 后续的任务状态都成为未知, 所以通常仿真时间需要被延长以尽量减少在仿真结束后出现超出截止时间的风险。

(2)基于形式化的方法是系统性能分析研究的热点, 这种方法基于数学模型, 通过明确的定义与严密的证明, 描述系统的行为。基于模型检测的时间自动机方法就是一种形式化

的分析方法。

时间自动机^[2]是在传统的有限状态自动机上增添对时间的约束机制, 来表达状态通过时间跃迁而变化, 从而满足实时性分析。实时系统中多任务的时钟表达总会出现空间组合爆炸的问题, 时钟带^[3]是一种关于时钟约束的描述组合, 可以尽量避免上述问题。

本文设计的调度分析工具利用有界差分矩阵(Difference-bound matrices, DBM)^[4]有效表示时钟带。DBM 支持时钟增长、时钟重置、时钟相交、时钟重置与标准化操作, 满足时钟与自动机有效关联。

本文对每个任务定义 2 种时间自动机: 事件到达自动机和任务状态自动机。

(1)事件到达自动机是在时间自动机理论上对系统中任务类型的描述, 如周期任务、零星任务和非周期任务。当时钟到达指定时刻时, 会产生同步事件, 并分派一个任务实例。

图 2 为一个周期任务事件到达自动机实例。

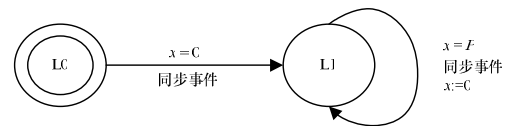


图 2 周期事件到达自动机实例

当时钟 x 为 0 时刻时, 初始位置 $L0$ 迁移到位置 $L1$, 并且将释放同步事件, 该事件会对相应任务的状态自动机进行同步。当时钟 x 增长为周期值 P 时, 位置 $L1$ 迁移到 $L1$, 同时再次释放同步事件, 并且将时钟 x 值赋值为 0, 重新计时, 继续循环。

(2)任务状态自动机是对任务被激活后, 内部状态跃迁的描述, 如图 3 所示。其中, 到达状态表示任务到达, 等待同步事件激活; 就绪状态表示任务就绪, 等待分配处理器资源。执行状态表示任务执行, 占用处理器资源; 错误状态表示任务超时, 未完成任务执行时间。

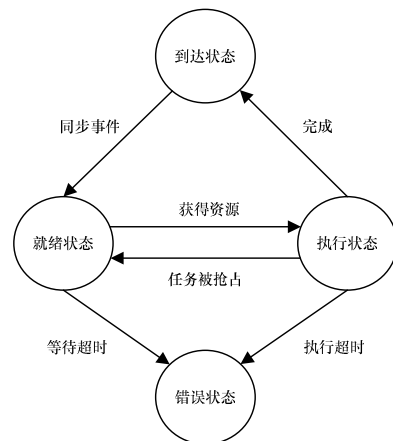


图 3 任务状态自动机内部跃迁描述

当任务捕捉到同步事件时, 由到达状态迁移至就绪状态; 当任务获得处理器资源时, 由就绪状态迁移至执行状态; 当有高优先级任务出现时, 由执行状态被抢占为就绪状态; 当出现超时, 则任务进入不可调度的错误状态; 当执行完成时, 恢复到就绪状态。

通过前期的验证^[5], 充分判定系统可调度性问题可以等效转换成时间自动机状态可达性问题。因此, 系统调度分析工具的设计把事件到达自动机和任务状态自动机视为一个节

点结构,随着时间流逝,动态构造调度分析树,通过访问可能到达的节点,比较每个节点 DBM 与状态信息,来判断该状态是否可达。如果判断任务已经来到错误状态,则立刻停止遍历,该系统不可调度;否则,直到所有可能节点已被遍历,验证该系统可以调度。调度分析树就是一种树形数据结构,用于存储分析的节点,初始根节点是任务集合中每个任务自动机的初始状态,其他叶子节点为时间约束下任务时间自动机的状态组合。可调度性即可表达成一系列到可达状态的合法分析路径。

本文工具支持如下各种调度分析算法,所有的算法都体现在任务状态自动机中就绪状态和执行状态之间的转换(获得资源或任务被抢占),当任务状态需要切换的瞬时操作中依据具体优先级规则判断抢占情况:

(1)单调速率调度(RMS)算法是最优的静态优先级调度算法,该算法定义为任务的周期值越小,任务的优先级越高,任务的周期值越大,其优先级越低。

(2)截止时间单调调度(DMS)是在单调速率调度算法的基础上发展起来的,其中任务的优先级按照截止时间来分配,截止时间值越短的任务优先级越高,反之,任务优先级越低。

(3)最早截止时间优先(EDF)算法满足动态性且拥有较高的 CPU 使用率,该算法定义任务距离截止时刻越近,任务的优先级越高,任务距离截止时刻越远,任务的优先级越低。

(4)固定优先级调度(FPS)算法和动态优先级调度算法相比,其具有实现简单、调度开销小、稳定性强等优点,所有任务的优先级在设计时考虑综合因素后确定,且在运行过程中保持不变。

为验证调度分析结果的准确性,本文工具提供了单步执行和连续执行的仿真功能,以保证对结果的校验。单步仿真原理是基于时间自动机的单步时间增长流逝,将每一步的时间流逝值与后继自动机边约束进行判断,选择可以通过的所有后继自动机状态中的一个。连续仿真设定持续时间值,动态判断单步仿真过程并逐个显示结果。

4 性能验证

本节列举 2 个典型的嵌入式实时系统模型^[6],通过与仿真方法的比较以验证调度工具的准确性。

4.1 复杂触发模型

复杂出发模型如图 4 所示,其中有 2 个处理器,任务 1 和任务 2 绑定于处理器 1,任务 3 和任务 4 绑定于处理器 2。任务 1 与任务 2、任务 2 与任务 3、任务 3 与任务 4 存在触发的关联关系。一个输入事件流经过任务 1 处理后作为任务 2 的输入,以此类推,直到经过任务 4 后形成输出。

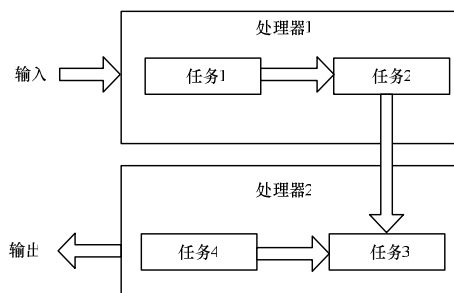


图 4 复杂触发模型

该模型参数如图 5 所示,其中,任务 1 的执行时间从 12 ms~24 ms 变化,分别对模型进行调度方法和仿真方法分析,端到端延时结果一致。

事件流输入周期: 30 ms											
处理器 1: 固定优先级					处理器 2: 固定优先级						
执行时间/ms					任务 2: 2		任务 3: 3		任务 4: 8		
优先级					任务 1>任务 2			任务 3>任务 4			
任务 1 执行时间/ms					12	14	16	18	20	22	24
输入端到输出端延时 /ms	调度方法				25	27	29	31	33	35	37
	仿真方法				25	27	29	31	33	35	37

图 5 复杂触发模型参数集

4.2 数据依赖模型

数据依赖模型如图 6 所示,其中只有处理器 1,任务 1~任务 3 都以固定优先级调度策略分布在其上。输入流 1 经过任务 1 处理后形成输出 1,输入流 2 则分别要经过任务 2 和任务 3。在这个模型中,任务 3 的执行时间不但依赖于任务 2 的触发关系,还要考虑到任务 1 的抢占因素。

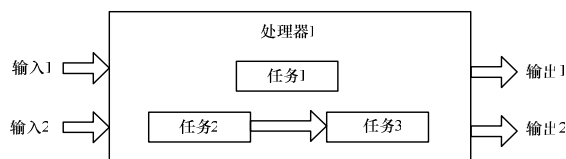


图 6 数据依赖模型

该模型的参数如图 7 所示,其中,任务 1 的执行时间从 12 ms~24 ms 变化,分别对模型进行调度方法和仿真方法分析,端到端延时结果仍然一致。

事件流输入 1 周期: 70 ms				事件流输入 2 周期: 40 ms				
处理器 1: 固定优先级								
执行时间/ms		任务 2: 15				任务 3: 5		
优先级		任务 1> 任务 2>任务 3						
任务 1 执行时间/ms		12	14	16	18	20	22	24
输入端 2 到输出端 2 延时/ms	调度方法	32	34	36	38	40	57	59
	仿真方法	32	34	36	38	40	57	59

图 7 数据依赖模型参数集

5 结束语

本文基于时间自动机理论,设计并实现了嵌入式实时系统的调度分析工具,该工具可以验证系统的可调度性,在开发早期阶段对系统性能进行分析。同时,通过对多处理器和外部环境事件触发的支持,加强了工具对任务模型的处理能力。列举对 2 个典型实例模型,利用与仿真方法的比较,确保分析结果的正确性。下一步将对时间自动机形式化方法作进一步研究,以改进调度性工具的分析能力和效率。

参考文献

- [1] 武海燕,晏立. 嵌入式实时软件的任务构造[J]. 计算机工程, 2010, 36(7): 39-41.
- [2] Fersman E, Mokrushin L, Pettersson P, et al. Schedulability Analysis of Fixed Priority Systems Using Timed Automata[J]. Theoretical Computer Science, 2006, 354(2): 301-317.
- [3] 陆少鹏. 嵌入式基于模型驱动验证及软件生产线的研究与实现[D]. 成都: 西南交通大学, 2010.
- [4] 孙全勇. 时间自动机及其应用研究[D]. 哈尔滨: 哈尔滨工程大学, 2007.
- [5] Gui Shenglin, Luo Lei. UCAS: A Schedulability Analysis Tool for AADL Models[C]//Proc. of EUC'08. Shanghai, China: IEEE Press, 2008.
- [6] Perathoner S, Wandeler E, Thiele L, et al. Influence of Different System Abstractions on the Performance Analysis of Distributed Real-Time Systems[C]//Proc. of EMSOFT'07. New York, USA: [s. n.], 2007.

编辑 金胡考