

# 信息物理系统实时任务 WCET 的研究

刘 辉, 张立臣, 许 阳

(广东工业大学 计算机学院, 广东 广州 510006)

**摘 要:**信息物理系统(CPS)是最近几年才出现的一个新的交叉领域的研究概念,它被普遍认为是计算机信息处理技术史上的下一次革命,将会改变人与现实物理世界之间的交互方式,具有广泛的应用前景。简要介绍了 CPS 的概念、一些新的特性。研究了 CPS 实时性方面的最坏执行时间(WCET)分析的组成部分、获取方法和计算算法,并比较了几种算法的优劣,列举了这一领域一些研究进展,讨论了 WCET 分析这一领域中存在的问题,给出了将来的研究方向。

**关键词:**信息物理系统;WCET 分析;可调度分析

**中图分类号:** TP302 TP316

**文献标识码:** A

**文章编号:** 1673-629X(2012)04-0126-04

## WCET Analysis on Real Task of Cyber Physical Systems

LIU Hui, ZHANG Li-chen, XU Yang

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China)

**Abstract:** Cyber physical system (CPS) is a new concept of cross-over research fields in recent years. It is commonly believed as the next revolution in information technology area. It will change the way of interaction between people and the physical world, and has a bright applied future. Introduce the concept and features of the CPS. Research the WCET analysis of real time task in CPS, compare the several computation means of the WCET, list some research in this field. Discuss the existed problems of WCET analysis, give the future work what is left to do.

**Key words:** cyber physical systems; WCET analysis; schedulability analysis

## 0 引 言

信息物理系统是一种深度嵌入式实时系统,集传感、控制、计算及网络技术于一体,是通过网络将信息系统与物理系统连接在一起而构成的一种大型的、异构的、分布式实时系统<sup>[1]</sup>。它与传统的系统相比具有一些新的特性<sup>[2-5]</sup>:1)每个物理部件都具有计算和通信能力;2)网络具有多样性,网络规模巨大;3)动态重组和重配置;4)在任意时空规模上控制循环是封闭的,人有可能也在该闭环中;5)操作必须是可靠的;6)分布式管理和控制;7)时间特性;8)容错性。在实时系统中,任务调度是保证系统实时性的关键因素之一。不同于传统紧密耦合的嵌入式实时系统,CPS是一种分布式的、异构的深度嵌入式实时系统。一个任务在传统系统中的执行时间容易被预测,而在 CPS 中,因

为异构而变得不易被预测。嵌入式软件 WCET 边界的确定,对下一代嵌入式实时系统<sup>[6]</sup>是一个极为重要的问题。嵌入式系统的软件设计需要严格遵守编码标准,以确保安全性和可靠性。其中一个例子是航空电子设备中 RTCA DO-178B 标准要求覆盖测试(语句、分支和条件)。实时嵌入式系统一个很重要的附加要求是可预测的软件组件的时序行为,特别是硬实时嵌入式系统必须满足的时序约束。空中客车公司,要求他们的供应商能够为目前正在研发的飞机(空中客车 380 和波音 787)上部署的软件提供 WCET 的界限。

目前国内从事 CPS 研究的学者和机构不多,西北工业大学计算机学院的周兴社和董云卫等提出了一种 CPS 时间行为建模方法<sup>[7]</sup>,大连理工大学软件学院的夏锋博士于 2009 年初创立了 CPS 研究组<sup>[8]</sup>,从事 CPS 网络协议、计算算法、软件平台等方面研究,但这些机构都没有从 CPS 系统实时性方面展开研究,因此对 CPS 中 WCET 的研究就很有必要。

## 1 WCET 分析概述

早在 1986 年, Kligerman E 等人就最先提出了 WCET 分析的概念。奥地利 Vienna 技术大学的研究

收稿日期:2011-08-17;修回日期:2011-11-21

基金项目:国家自然科学基金项目(60373000,604740722);国家自然科学基金重大发展计划项目(90818008);广东省自然科学基金项目(07001774,04009465)

作者简介:刘 辉(1986-),男,河南南阳人,硕士,研究方向为实时系统;张立臣,教授,硕士生导师,研究方向为实时系统和分布式系统。

者给出的定义是<sup>[9]</sup>:计算程序代码块的执行时间的上限,且同时保证安全性和精确性,代码块的执行时间是指占用 CPU 的时间。程序实际的 WCET 除了自身因素外,还受到上下文环境的影响,不同环境下的执行时间很有可能不一样,因此得到精确的 WCET 是很困难的,只能获取一个估计值。根据 WCET 分析得到的 WCET 估值,可以为系统设计师确定任务的预计执行时间总值提供参考,是用来对实时系统的实时性和可靠性进行验证的,所以 WCET 分析必须满足四个原则:表现性、高效性、有效性和精确性。表现性是说选用的策略能对各个分析过程得到的信息进行处理;高效性指分析策略在一定执行时间内只占用适量内存就可完成任务;有效性指不能低估 WCET,即任意程序段的 WCET 估值应不小于实际运行时的 WCET 值;精确性要求提供可接受的高估值,即 WCET 估值与实际运行时的 WCET 值的误差要小于一定的范围。

## 2 获取 WCET 估值的方法

WCET 分析包括静态分析、动态测量和混合方法共 3 种方法。动态测量方法主要应用在工业上,测量工具有模拟器、逻辑分析仪和示波器等。该方法的思路是直接在目标环境中运行程序,通过尽可能多的测试用例来得出程序的 WCET。常用的策略有遗传进化算法<sup>[10]</sup>、以生成动态测试用例为目标的优化算法<sup>[11]</sup>、数学统计方法<sup>[12,13]</sup>和随机性测试方法。动态测试的结果和以下三个因素有关:处理器初始状态、输入数据的控制转移指令产生的多执行路径和处理器在访问输入数据时的处理器状态,因此测量方法得到的结果不十分可靠,它只有在控制流路径变化少的程序中,才获得可靠的 WCET 上界。

静态分析方法不实际执行程序,针对系统的软件特性进行分析,在特定的硬件平台上建立合适的模型,通过分析所建模型得出程序的 WCET。目前工业中的处理器发展很快,尤其是缓存技术和流水线技术更是日新月异,所以基于复杂处理器特性来获取 WCET 的方法变得很复杂。由于用该方法得到的 WCET 值满足四原则中的有效性,即 WCET 的估值不低于实际值,所以静态分析成为 WCET 分析中的常用方法。

混合方法,顾名思义就是把静态和动态方法相结合<sup>[14,15]</sup>,一般有两种结合方式:或者先静态分析,然后动态测量;或者先动态测量,在此基础上进行分析。目前对这一方法的研究还不是很成熟。

因为动态测量方法是在程序运行时进行的,需要考虑许多无法确定的因素,所以很难通过测量得到精确的 WCET 值,经常是低于实际的 WCET 值。而静态方法是事前分析程序结构,不需考虑物理特征,所以不

仅能够获取单个任务的执行时间,而且对一些无法实际执行的程序段也适用。一般情况下,静态分析考虑情况比较全面,这也导致获得的 WCET 值比实际值偏高。传统的 WCET 分析把每条底层指令的 CPU 时间视为定值,故而得到的程序运行时间也是一个确定的值。然而,随着缓存技术、分支预测等技术的发展,再加上浮点型指令在不同机器上的运行时间的不确定性,使得程序的 WCET 也不再是固定值,而是在一个时间范围内波动,即代码段的 WCET 值符合一定的概率分布,因此一些学者就提出了概率 WCET 的概念。静态和动态方法的分析结果比较如图 1 所示。

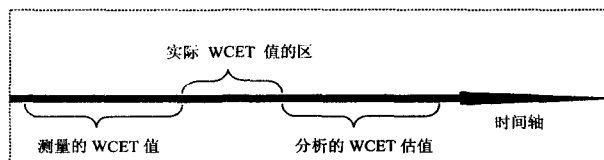


图 1 静态分析和动态测量的 WCET 值的范围

## 3 WCET 分析的过程

静态方法是 WCET 分析中的主流方法,主要包括:软件层分析、硬件层分析、在软硬件分析基础上计算 WCET 值<sup>[16]</sup>。软件分析阶段只在高级语言环境中针对目标程序的结构分析,找出影响执行时间的因素,如循环迭代次数、任何情况下也不能执行的路径等信息,从而确定最坏情况下指令序列;硬件分析阶段考虑硬件特性如缓存、流水线等对目标系统的影响,在汇编环境下对系统建立模型;计算最坏情况下目标指令的 WCET 值。

文献[17]对静态分析<sup>[18]</sup>的过程进行了分类,分为三个正交面,如图 2 所示。

X轴-表示层次;Y轴-程序流;Z轴-时间模型

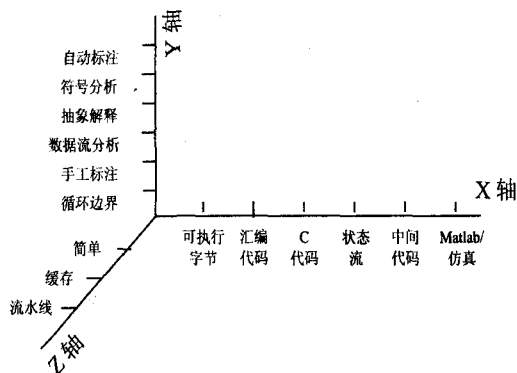


图 2 静态分析的详细表示

其中,X 轴表示的是源代码层的影响因子,主要表现在高级环境中,比如状态流是诸如在 STATEMATE 系统实施的状态图描述的语义<sup>[19]</sup>。Y 轴表示的是程序的语法和语义层,是控制程序流向的信息,包括循环

次数、调用关系以及其他约束信息。Z 轴表示的是考虑硬件元素对时间的影响所建立的模型,如缓存和流水线技术对目标系统执行时间的影响,该模型是程序运行在目标机上的行为。静态 WCET 分析的三个部分联系紧密、不易分开讨论。Z 轴的不同层次对 Y 轴的要求是不同的,X 轴上不同的表示层次能够提供不同的 Y 轴上的程序流事实的信息。总之,静态分析方法的过程是连贯的,它们互相依赖、互相影响。

## 4 计算 WCET 的算法

在完成软件层分析和底层硬件分析后,根据上述两步的分析结果,依据一定的算法计算程序的 WCET 值。一般常用的 WCET 计算策略主要有三种:列举所有隐藏执行路径、树结构、寻找最长路径。

### 4.1 列举所有隐藏执行路径

该法是目前使用最多的求解方法,该方法根据对程序流的执行计数变量进行赋值来建立迭代模型,这个计数变量的值反映了执行程序中的节点总数。而每个实体有一个计数变量  $x$  和一个时间变量  $t$ ,其乘积  $x \cdot t$  即为一个实体的执行时间。使用数学逻辑约束建立目标函数进行最大化求解来得到 WCET 估计值,即  $WCET = \maximize \{Xi * Ti, \text{对任一实体 } i\}$ 。

实际计算时,采用了时间图,它与域图对应。该方法的优点在于分析中不需列举所有的程序路径,因此对程序结构没有要求,能处理任何形式的程序;主要缺点是不能指出具有最大执行时间的路径,因此无法针对特定路径进行 WCET 值的优化。

### 4.2 树结构

该法根据为程序语句定义的规则确定该语句的 WCET,从下到上遍历程序的语法树结构来计算最终 WCET 值。计算时采用的模型是树模型。优势是设计简单,计算 WCET 的代价也较低。主要问题表现在不能表达任意的连接关系,只能适用于良好结构的程序,难以包含进分支依赖关系,不能给出显式路径,难以进一步做底层分析。主要针对的是复杂指令的处理器。

### 4.3 寻找最长路径

该法利用递归算法,遍历程序中所有可能的执行路径,同时计算所有路径的执行时间并记录,然后找到具有最长执行时间的路径来确定最终的 WCET。计算时采用的模型是有向图。该类方法在针对 WCET 值进行软件技术优化场合应用广泛,缺点是当存在很多可行路径时,其时间复杂度随循环、分支数量呈指数级增长,效率较低。

### 4.4 其他方法

文献[20]利用生成函数方法,针对复杂指令处理器中指令的执行时间固定的特性,利用数据流框架来

获得目标程序的 WCET 值。如果已知程序控制流程图的边执行频率,该方法对 WCET 的分析计算可以达到完全精确。

文献[21]扩展以前的研究,通过计算循环迭代次数,给出以循环迭代次数为参数的符号化公式,即最坏情况循环执行时间=最坏情况路径时间\*路径的数目+循环迭代次数。

文献[22]利用 Newton-Gregory 插值多项式来处理嵌套循环的迭代次数,从而可以得到计算程序 WCET 的参数化公式。

## 5 结束语

文中介绍了 CPS 的基本概念及特性,重点从时间特性研究了 WCET 分析技术。经过近三十多年的发展,WCET 分析技术的研究在各类研究机构中产生了深远的影响,甚至在实际的工程领域中开始崭露头角,成为热门的研究点。然而硬件的发展日新月异,新的特性总是不断的出现,使得 WCET 分析总是落后于硬件新特性的发展。因此,接下来的一个研究方向就是解决具有先进体系结构处理器的时间模型。另一方面,信息物理系统的调度分析也需要程序的 WCET 参数,这也是未来的研究方向。单纯的静态或动态方法很难使 WCET 估值精确,还需研究新的混合处理方法,大大提高 WCET 分析的可信度。

### 参考文献:

- [1] 谭朋柳,舒 坚,吴振华.一种信息-物理融合系统体系结构[J].计算机研究与发展,2010(22):312-316.
- [2] Lee E A. Cyber-physical systems-are computing foundations adequate? [C]//Position Paper for NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap. Austin, TX: Cyber-Physical Systems Workshop, National Science Foundation, 2006.
- [3] Tan Ying, Goddard S, Perez L C. A prototype architecture for cyber physical systems [J]. ACM SIGBED Review, 2008, 5 (1): 1-2.
- [4] Richard W, Gabriel P. A software architecture for next generation cyber-physical systems [EB/OL]. [2010-07-10]. <http://www.CS.bu.edu/~richwest/papers/weM-cps.pdf>.
- [5] Lee E A. Computing foundations and practice for cyberphysical systems; a preliminary report [R]. Berkeley: University of California, Berkeley, 2007.
- [6] Wegener J, Mueller F. A comparison of static analysis and evolutionary testing for the verification of timing constraints [J]. Real-Time Systems, 2001, 21(3): 241-268.
- [7] Wang Hanbo, Zhou Xingshe, Dong Yunwei, et al. Modeling timing behavior for cyber-physical systems [C]//Proc of Int Conf on Computational Intelligence and Software Engineering.

- Piscataway, NJ: IEEE, 2009: 1-4.
- [8] Xia Feng. Cyber-physical systems research group[EB/OL]. [2010-03-10]. <http://www.cpschina.org/>.
- [9] Puschner P, Burns A. Guest Editorial: A Review of Worst-Case Execution-Time Analysis[J]. Real-Time Systems, 2000, 18(2-3): 115-128.
- [10] Wegener J, Grochtman M. Verifying Timing Constraints of Real-Time Systems by Means of Evolutionary Testing[J]. Real-Time Systems, 1998, 15(2): 275-298.
- [11] Tracey N, Clark J, Mander K. The Way Forward for Unifying Dynamic Test Case Generation: The Optimization-Based Approach[C]//Proc of International Workshop on Dependable Computing and Its Applications (IFIP'98). Johannesburg, South Africa: [s. n.], 1998: 12-14.
- [12] Petters S M. How Much Worst Case is Needed in WCET Estimation[C]//2nd International Workshop on Worst-Case Execution Time Analysis. Technical University of Vienna, Austria: [s. n.], 2002.
- [13] Bemat G, Colin A, Petters S M. WCET Analysis of Probabilistic Hard Real-Time Systems[C]//Proc of the 23rd Real-Time Systems Symposium. [s. l.]: [s. n.], 2002.
- [14] Kiltler R, Puschner P, Wenzel I. Measurement-Based Worst-Case Execution Time Analysis Using Automatic Test-Data Generation[C]//Proc of the 4th International Workshop on Worst-Case Execution Time (WCET) Analysis. Catania, Sicily, Italy: [s. n.], 2004.
- [15] Betts A, Bemat G. Issues Using the Nexus Interface for Measurement-Based WCET Analysis[C]//5th International Workshop on Worst-Case Execution Time Analysis. Palma de Mallorca, Spain: [s. n.], 2005.
- [16] Engblom J. Processor Pipelines and Static Worst-Case Execution Time Analysis[D]. Uppsala, Sweden: Acta Universitatis Upsalienensis, 2002.
- [17] Vivancos E, Healy C, Mueller F, et al. Parametric Tuning Analysis[C]//Workshop on Languages, Compilers, and Tools for Embedded Systems. Utah: ACM Press, 2001: 88-93.
- [18] Kirner R, Puschner P. Classification of WCET Analysis Techniques[C]//Proc of 8th IEEE International Symposium on Object-oriented Real-time Distributed Computing. Seattle, WA: [s. n.], 2005: 190-199.
- [19] Harel D, Naamad A. The STATEMATE Semantics of Statecharts[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 1996, 5(4): 293-333.
- [20] Shaw A C. Reasoning about time in higher level language software[J]. IEEE Transactions on Software Engineering, 1989, 15(7): 875-889.
- [21] Blieberger J. Daut-flow framework for worst-case execution time analysis[J]. Real-Time Systems, 2002, 22: 183-227.
- [22] van Engelen R A, Gallivath K, Walsh B. Parametric Timing Estimation with the Newton-Gregory Formulae[J]. Journal of Concurrency and Computation: Practice and Experience, 2004, 18(11): 1435-1463.

(上接第125页)

#### 4 结束语

文中尝试在 VxWorks 操作系统环境下实现嵌入式故障恢复系统各模块原型。通过测试表明,该原型具有基本的故障恢复能力。对在 VxWorks 系统下,利用系统方法提高软件系统的容错能力具有一定的参考意义。进一步的工作主要是对检查点设置<sup>[14]</sup>过程、故障更进一步的细分和针对各类型恢复策略进行优化,并完善原型系统的任务恢复能力。

#### 参考文献:

- [1] 孙 栓,赵 敏,戴 维.微小卫星星载计算机存储容错技术研究[J].计算机技术与发展,2008,18(8):148-151.
- [2] 王 霆,长宁宁,王艳利.分布式高可靠性星载计算机系统研究与实现[J].科技信息,2008(10):57-58.
- [3] 历海燕,赵志国,李新明.分布式嵌入式环境下的容错软件设计[J].计算机应用研究,2006(7):142-143.
- [4] 罗国庆.嵌入式软件开发[M].北京:机械工业出版社,2003.
- [5] 黄启军,李晓光.基于 S3C44B0X 微处理器的 VxWorks BSP 开发与设计[J].计算机技术与发展,2006,16(7):45-47.
- [6] 周启平,张 杨,吴 琼.VxWorks 开发指南与 Tornado 实用手册[M].北京:中国电力出版社,2004:251-254.
- [7] 李小群,张文君,潘远明,等.基于 RTEMS 的软件容错系统设计[J].计算机应用研究,2009,26(3):911-913.
- [8] 张卫民.航天飞控软件的二维容错体系结构设计[J].计算机工程,2008,34(5):266-267.
- [9] 李宏亮,文 梅,张春元,等.高可用实时系统中故障检测及故障恢复技术的研究[J].计算机工程与科学,1999,21(6):81-84.
- [10] Shirvani P P, McCluskey E J. Fault-tolerant Systems in A Space Environment: The CRC ARGOS Project[R]. [s. l.]: [s. n.], 1998.
- [11] 苟冬荣,刘海清.双机容错计算机系统的设计与实现[J].计算机工程,2008,34(15):255-257.
- [12] Deconinck G. Software-implemented Fault Tolerance and Separate Recovery Strategies Enhance Maintainability[J]. IEEE transactions on reliability, 2002, 51(2): 158-165.
- [13] Huang Yennun, Kintala C. Software Fault Tolerance in the Application Layer[M]. [s. l.]: John Wiley & Sons Ltd., 1995: 237-247.
- [14] 鄢喜爱,杨金民,田 华.双机容错系统中最佳检查点间隔的分析[J].计算机工程,2007,33(5):283-284.