

分布式实时嵌入式系统端到端性能确保

廖 勇, 熊光泽, 陈旭东, 桑 楠, 朱清新

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】针对具有端到端截非周期任务模型的分布式实时系统,提出了一种ICA算法,构造了一个兼顾非周期实时任务个体性能和总体性能的分布式控制环。该算法能动态调整任务的QoS,从概率上确保了任务的端到端截止时间,并能通过确保CPU利用率来提高系统吞吐量。实验结果证明ICA能为具有端到端非周期任务模型的分布式实时嵌入式系统提供更好的性能确保。

关键词 分布式实时嵌入式; 反馈控制; 实时系统; 调度
中图分类号 TP316 文献标识码 A

End-to-End Performance Guarantee for Distributed Real-Time Embedded Systems

LIAO Yong, XIONG Guang-ze, CHEN Xu-dong, SANG Nan, ZHU Qing-xin

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

Abstract This paper proposes an Integrated Control Algorithm (ICA) for the Distributed Real-Time Embedded Systems (DRES) that employs the end-to-end aperiodic task model. ICA features a distributed feedback loop that relates the per-task metric to the aggregate metric. It is able to dynamically determine the QoS (Quality of Service) of incoming tasks and guarantee the end-to-end deadlines of accepted tasks, at the same time enforce the desired utilization bound on multiple processors. Finally, experiments demonstrate that ICA is capable of providing better performance guarantees for DRES that employs the end-to-end aperiodic task model.

Key words distributed real-time embedded; feedback control; real-time systems; scheduling

近年来,分布式实时嵌入式系统(Distributed Real-Embeckled Systems, DRES)应用日趋广泛^[1],如过程控制系统、自动监控系统以及具有多层结构的WEB服务。而端到端任务模型^[1-2]在该类系统中大量应用:每一个任务由多个子任务组成,子任务之间有着前驱/后继关系,并分阶段依次运行于不同的CPU上,如:一个监控任务需要经过以下三个子任务的处理:传感器采样、数据传输、本地数据显示,而每个子任务分别运行在不同处理机上。大多数这样的系统因具有动态性和不确定性^[1, 3-4]而归属于软实时系统。开发此类系统首先要考虑的是:如何在系统任务数目和载荷不确定的情况下确保非周期任务的个体性能,即:端到端截止时间。

为此,本文提出了一种集成控制(Integrated Control Algorithm, ICA)算法。ICA集成了非周期任务许可控制机制与反馈控制机制,前者根据非周期任务的可调度性上限^[5]接收任务,能满足接收任务

的端到端截止时间;后者能确保系统中多个CPU的利用率、提高系统吞吐量。在集成控制中,本文利用了瞬态利用率(Synthetic Utilization)^[5]和实测利用率,前者用于许可控制以确保任务的个体性能;后者用于衡量CPU是否过载,是否保持在设定值,从而进行反馈控制。由于瞬态利用率小于等于实测利用率^[5],因此,它们之间存在着一定的函数关系(见第3.1节),可以动态调整到达任务的瞬态利用率使得实测利用率稳定在设定值,这样,系统的总体性能和任务的个体性能就可得以兼顾。此外,在反馈控制中,本文提出了一种新的(Quality of Service, QoS)自适应机制:将任务的端到端截止时间分为不同的QoS级别,当系统过载时,降低任务的QoS,反之则提高任务的QoS。

1 任务模型

本文考虑了一种由 M 个CPU($\{P_i | 1 < j < M\}$)组

收稿日期:2005-10-20

基金项目:国防科研基金;国家863计划资助项目(2003AA1Z2210)

作者简介:廖勇(1976-),男,博士生,主要从事分布式实时系统、实时调度等方面的研究。

成的分布式实时嵌入式系统，CPU通过高速通信接口(如VME)相互连接，处理器之间的通信开销可以忽略不计^[1]。令端到端的非周期任务为 $\{T_i\}$ (i 为任意正整数)，任务之间彼此独立、可抢占。每个任务 T_i 由一组子任务 $\{T_{ij} | 1 < j < n_i\}$ 组成 ($1 < n_i < M$)，并依次分配到多个CPU上执行，每个子任务 T_{ij} 只有在其上一个子任务 $T_{i(j-1)}$ 执行完毕之后才能执行。子任务 T_{ij} 被分配到哪个CPU中执行由应用系统确定，如由两个处理器、三个任务组成的系统如图1所示。图中 T_1 只有子任务 T_{11} 在 P_1 上执行， T_2 有两个子任务 T_{21} 和 T_{22} 分别在 P_1 和 P_2 上执行， T_3 也只有子任务 T_{32} 在 P_2 上执行。 T_i 的到达时间用 A_i 表示，端到端的相对截止时间用 D_i 表示， T_{ij} 在 P_j 上的估计执行时间用 C_{ij} 表示(注意： C_{ij} 可为0，意味着 T_i 没有子任务在 P_j 上执行)。

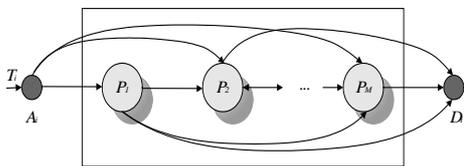


图1 端到端非周期任务模型

用 l_q 表示 T_i 所期望确保的端到端截止时间的QoS级别。每个任务定义 N 个QoS级别($N \geq 2$)。 l_q 越高， T_i 分配到的截止时间就越短。如果 $l_q=0$ ，指 T_i 被拒绝。令 $D_i(l_q)$ 为QoS级别 l_q 所对应的端到端相对截止时间， $D_{ij}(l_q)$ 为子任务 T_{ij} 的相对截止时间(有关将 $D_i(l_q)$ 分配到子任务的策略将在第3.3节介绍)。由于任务请求的利用率是QoS级别 l_q 的单调递增函数，所以接收任务

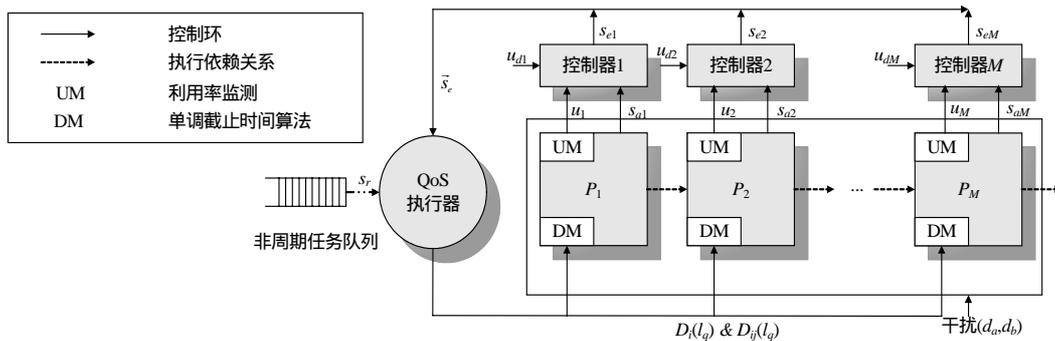


图2 集成控制结构

图2为整个系统包括 M 个分散的控制器(Controllor)，每一个根据自己的CPU参数进行控制，它利用反馈控制理论，并器根据 u_j 和 s_{ej} 间的关系计算出估计瞬态利用率 s_{ej} ($j = 1, 2, \dots, M$) (见3.1节)；然后， s_{ej} 传递给执行器(Actuator)进行QoS调整和许可控制(见3.3节)，这样便可将CPU利用率稳定在某一设定值。另一方面，在许可控制中，只有当到达任务的

的 l_q 将会影响到CPU的实测利用率。这样，在反馈控制和许可控制下，通过调整 l_q ，CPU的实测利用率便可保持在设定值，同时接收任务的端到端截止时间也在概率上可得以满足。

任务根据如下方式进行同步：令第一个子任务 T_{i1} 的到达时间等于父任务 T_i 的到达时间，后继子任务 T_{ij} ($j = 2, 3, \dots, M$)的到达时间由其前驱子任务 $T_{i(j-1)}$ 的截止时间确定，即： $A_{ij}(l_q) = d_{i(j-1)}(l_q)$ ($d_{i(j-1)}(l_q)$ 是 $T_{i(j-1)}$ 的绝对截止时间)，这样，子任务间的前驱/后继关系便可得以维护。

2 集成控制结构

图1为非周期任务的端到端性能控制是一个分布式控制问题，需要同时确保任务端到端截止时间及多个处理器的利用率，因此，经典控制方式已不适用，为此，本文提出了能兼顾任务个体性能和总体性能的综合控制结构。图2中， u_j ($j = 1, 2, \dots, M$)为处理器 P_j 的实测利用率； s_{aj} ($j = 1, 2, \dots, M$)为实测瞬态利用率； s_{ej} ($j = 1, 2, \dots, M$)为估计瞬态利用率，由控制器计算得出； u_{dj} ($j = 1, 2, \dots, M$)为实测利用率设定值，是开发人员期望的CPU利用率，它反映了CPU的繁忙程度； s_r 为任务队列中所有任务的请求瞬态利用率； d_a 反映任务的估计执行时间和实际执行时间的偏差所产生的干扰，正是 s_r 和 d_a 这两个因素导致了系统的不确定性；另一个干扰 d_b 由实测利用率和实测瞬态利用率的非线性关系所产生。

瞬态利用率不超过其可调度上限(58.6%)时文献[1]证明：当采用DM调度策略时，只要能确保处理器上的瞬态利用率小于58.6%，则非周期实时任务的可调度性就一定能够得到满足。由于篇幅有限，本文不予详述，任务才能被接收，这样便可从概率上确保任务的端到端截止时间。

另外，图2中的每个CPU都由自己的调度器负责

底层的任务调度,本文采用DM(Deadline Monotonic)算法作为调度策略。每个CPU还有一个利用率监控器,负责在每个采样周期测量 u_j 和 s_{aj} 值。采样周期是根据系统性能要求,并最终通过实验确定^[1]。

3 系统模型

3.1 单CPU控制

由于ICA采用的是分布式控制结构,每个CPU的控制是独立进行的。首先讨论单CPU控制:因为瞬态利用率与实测利用率是不相等的,因此,假设两者之间的关系为: $u = G_b s_a$ 。进一步假设估计瞬态利用率和实测瞬态利用率的关系为: $s_a = G_a s_e$,其中, G_a 和 G_b 反映系统的固有属性。由于两者均为非线性、时变参数,因此有必要将它们线性化,令:

$G_a = K_a, G_b = K_b$, 其中, K_a 和 K_b 是由实验确定的常数。该线性化方法与文献[1]的方法类似,本文不予详述,这样: $u = K_b s_a, s_a = K_a s_e$ 。

为了对系统不确定性因素做出自适应调整,并确保任务个体性能,ICA构造了两个反馈控制环,双环反馈控制如图3所示。图中 $e_1 = u_d - u$, $e_2 = s_d - s_a$ (s_d 瞬态利用率的设定值 $s_d < 58.6%$), 控制环1用于保证实测利用率 u 保持在其设定值 u_d , 控制环2用于消除估计瞬态利用率 s_e 和实测瞬态利用率 s_a 之间的误差。令: $\frac{du}{dt} = k_1 e_1, \frac{ds_a}{dt} = k_2 e_2$, 其中, $k_1 e_1, k_2 e_2$ 分别表示实测利用率和实测瞬态利用率的变化比率。经Laplace变换,可得:

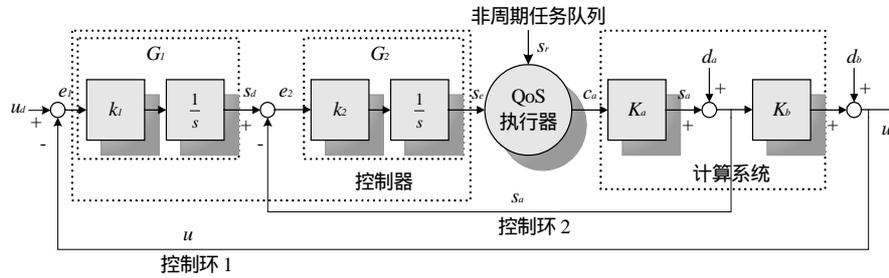


图3 双环反馈控制

$$G_1(s) = \frac{u(s)}{e_1(s)} = \frac{k_1}{s}, \quad G_2(s) = \frac{s_a(s)}{e_2(s)} = \frac{k_2}{s} \quad (1)$$

则二阶系统模型为:

$$\frac{u(s)}{u_d(s)} = \frac{k_1 k_2 K_a K_b}{s^2 + k_2 K_a s + k_1 k_2 K_a K_b} \quad (2)$$

3.2 分布式实时嵌入式系统的控制

$$\text{令: } F_j(s) = \frac{u_j(s)}{u_{dj}(s)} = \frac{k_{1j} k_{2j} K_{aj} K_{bj}}{s^2 + k_{2j} K_{aj} s + k_{1j} k_{2j} K_{aj} K_{bj}},$$

其中($j=1,2,\dots,M$), 则分布式实时系统的控制为:

$$\bar{u}(s) = \bar{F}(s) \bar{u}_d(s) \quad (3)$$

3.3 执行器(QoS actuator)设计

系统中只有一个QoS执行器如图2所示,但是,正是该执行器从端到端的角度进行许可控制和QoS自适应,才使得非周期实时任务个体性能和总体性能得以确保。当一个新任务到达系统时,执行器根据控制器的输出 \bar{s}_e 和每个CPU的实测瞬态利用率判断是否能接收它。如果能,则让该任务进入系统;否则,降低任务的QoS,直到任务的可调度性得到满足。一种极端情况是:任务降级到最低级别时其可调度性仍然得不到保证,此时,该任务将被拒绝。然后,执行器根据式(4)的分配原则将父任务的端到端截止时间 $D_i(l_q)$ 分配至其子任务上。

$$\frac{C_{i1}}{D_{i1}(l_q)} : \frac{C_{i2}}{D_{i2}(l_q)}, \dots, \frac{C_{iM}}{D_{iM}(l_q)} = s_{e1} : s_{e2}, \dots, s_{eM} \quad (4)$$

这样,执行器接收任务的瞬态利用率 c_a 为(注:当 $s_r < s_e$ 时,到达载荷将不受图3的控制环控制,直接进入系统):

$$c_a = \begin{cases} s_e & s_r > s_e \\ s_r & s_e < s_r \end{cases} \quad (5)$$

4 实验结果

仿真实验考虑由三个CPU构成的分布式实时嵌入式系统。一个任务产生器模拟非周期实时任务的到达,让任务到达服从泊松分布,任务端到端截止时间分为三个QoS级别。第一个级别在[300, 500] ms服从均匀分布;第二个级别和第三个级别分别是第一个级别的1/3和1/5,任务的执行时间在[6, 15] ms内服从负指数分布。参数 k_1, k_2, K_a, K_b 的值分别为0.18、0.195、0.8、1.4。实验模拟非周期任务的执行,每个任务的子任务都将依次在三个CPU上执行,仿真持续400 s,并让每个CPU的到达载荷在50%~190%间变化。图4绘出了在采用许可控制不采用许可控制的情况下三个CPU的瞬态利用率的变化(浅色虚线为非周期任务的可调度上限58.6%),

图5记录了在该情况下非周期任务的端到端截止时间错过率。可见,当系统载荷为动态、不确定时,ICA能将截止时间错过率控制在一个用户可以接受的范围内(本实验为3%,图5所示),以实现在概率上确保非周期任务的端到端截止时间;此外,即使是在采用了许可控制的情况下,仍然会有任务错过其端到端截止时间(图5的第1个图),这是由 G_a 和 G_b 非线性关系造成的被控系统与实际系统之间的误差而引起,而这种现象在软实时系统中是允许的。

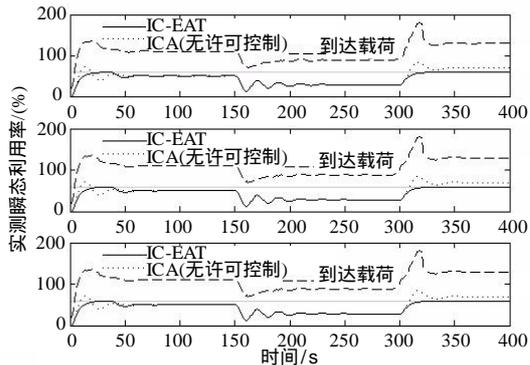


图4 非周期任务的执行

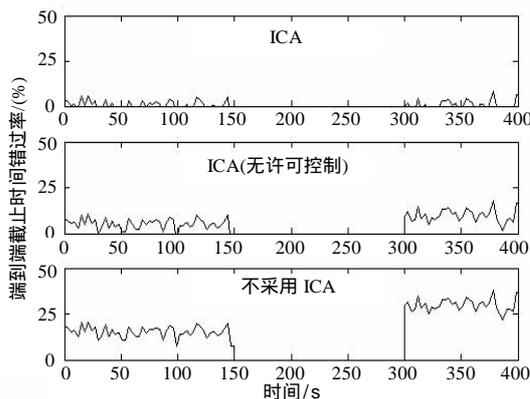


图5 端到端截止时间错过率

5 结论

本文针对具有端到端非周期任务模型的分布式实时嵌入式系统提出了一种ICA算法,它兼顾非周期实时任务的个体性能和总体性能,并将任务估计执行时间与实际执行时间之间的偏差、非周期任务到达作为引起系统不确定的两个外部因素。ICA能通过许可控制从概率上确保任务的端到端截止时间,并能通过反馈控制和QoS自适应提高系统吞吐量。实验结果证明,ICA能为系统提供更好的性能确保。

参考文献

- [1] LU Chen-yang, WANG Xiao-ru, Xenofon Koutsoukos. end-to-end utilization control in distributed real-time systems[C]//International Conference on Distributed Computing Systems, Tokyo, Japan, [s.n.], 2004: 456-466.
- [2] SUN J, LIU J W S. Synchronization protocols in distributed real-time systems[C]//International Conference on Distributed Computing Systems, Hongkong:[s.n.],1996: 38-45.
- [3] LUI Sha, ABDELZAHER T F, KARL-Eric Arzen, et al. Real time scheduling theory: a historical perspective[J]. Journal of Real-time Systems, 2004, 33(5): 101-155.
- [4] LIAO Yong, CHEN Xu-dong, SANG Nan, et al. Optimal oeward based adaptive CPU resource allocation for computing devices in pervasive environment[J]. Journal of Information and Computational Science, 2005, 2(1):75-80.
- [5] ABDELZAHER T F, SHARMA V, LU Chenyang. A utilization bound for aperiodic tasks and priority driven scheduling[J]. IEEE Transactions on Computers, 2004, 53(3): 334-350.

编辑 刘文珍