嵌入式实时系统中基于检验点检测的电压分配技术

李国徽[1] 杨 兵[1] 胡方晓[1] 许华杰[2] 杜建强[3]

[1](华中科技大学计算机学院 武汉 430074)

[2](上海第二工业大学计算机与信息学院 上海 201209)

[3](江西中医学院计算机系 南昌 330006)

摘 要 嵌入式实时系统的实时特性、高度的系统可靠性和低系统能耗对综合考虑系统的容错和节能提出了要求.研究了嵌入式实时系统中如何达到上述三大目标的非周期任务调度和电压分配问题.在基于检验点容错技术的可调度性检测基础上,提出了对于系统非周期任务的基于调度性检测的电压分配算法 CST-VA,该算法在保证任务实时性的同时提高了系统的可靠性,节省了系统的能耗.模拟实验表明,该算法比现有电压分配算法更适合于嵌入式实时系统.

关键词 嵌入式实时系统;非周期任务;容错技术;能量管理;电压分配

中图法分类号 TP316 **DOI 号**: 10.3724/SP.J.1016.2009.02403

A Voltage Allocation Technique with Checkpoint-Based Schedulability Test in Embedded Real-Time Systems

LI Guo-Hui[1] YANG Bing[1] HU Fang-Xiao[1] XU Hua-Jie[2] DU Jian-Qiang[3]

[1](*School of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan* 430074)

[2](*School of Computer and Information, Shanghai Second Polytechnic University, Shanghai* 201209)

[3](*Department of Computer Science, Jiangxi University of Traditional Chinese Medicine, Nanchang* 330006)

Abstract Fault-tolerance through software/hardware redundancy as well as power management through frequency and voltage scaling has been well studied separately in the context of embedded real-time systems. However, the real-time characteristic, the high-level reliability and the low power consumption make the combination of fault-tolerance and power management necessarily for embedded real-time systems. This paper proposes a new method for aperiodic task scheduling and voltage allocation called voltage allocation with checkpoint-based schedulability test (CST-VA). CST-VA not only guarantees the timing constrains, also provides higher reliability through checkpointing and minimizes power consumption via optimal voltage allocation technique. Simulation results show that CST-VA has better performance compared with the existing voltage allocation technique and is more appropriate for embedded real-time systems.

Keywords embedded real-time systems; aperiodic tasks; fault-tolerance; power management; voltage allocation

---

收稿日期:2007-11-25;最终修改稿收到日期:2009-10-12.本课题得到国家自然科学基金(60873030)、国家"八六三"高技术研究发展计划项目基金(2007AA01Z309)和中央高校基本科研业务费专项资金资助.**李国徽**,男,1973 年生,博士,教授,博士生导师,主要研究领域为现代数据库理论及集成技术、实时系统、移动实时计算、软件工程方法与技术等. E-mail: guohuili@mail.hust.edu.cn. **杨 兵**(通信作者),男,1975 年生,博士,主要研究方向为移动实时数据库系统. E-mail: yangbing@126.com. **胡方晓**,男,1983 年生,博士研究生,主要研究方向为实时系统的容错、节能技术、移动实时数据库的事务处理. E-mail: humizi911@163.com. **许华杰**,男,1976 年生,博士,讲师,主要研究方向为传感器网络. **杜建强**,男,1968 年生,教授,主要研究领域为医学图像处理.

1 引 言

嵌入式实时系统的广泛应用,使得嵌入式实时系统得到了深入的研究.在多数实际的嵌入式应用中,实时特性、高度的系统可靠性和低系统能耗成为设计的主要目标.因此,利用冗余的容错技术[1-2]和利用改变频率、电压的能量管理[3-4]在嵌入式实时系统中分别得到了深入的研究.但是,对于综合考虑系统容错和节能的研究较少.

嵌入式系统通常工作于恶劣环境,这就对系统的可靠性提出了要求,使得容错技术变得必要.容错计算是指用户程序和系统软件在出现错误的情况下,仍能正确执行[5].它主要通过任务的重新执行或组件冗余来实现.在实时系统中,典型的容错是通过以下3步完成的:在线错误检测、记检验点和回滚恢复.

记检验点是有代价的,同时会增加任务的执行时间.在不出现错误的情况下,记检验点可能会使得该任务错过截止期;而当错误发生时,记检验点能减少任务重新执行所需的时间,从而可能使得该任务满足截止期要求.频繁的记检验点可以缩短重新执行的时间,但是会增加任务总的执行时间;而不频繁的记检验点在不出现错误时对系统没有多少影响,但当错误出现时会使得重新执行的时间变长.因而,记检验点的频率应同检验点的代价及回滚时间等因素综合考虑.

嵌入式系统通常又是严重的能量受限于电池的工作寿命,这对系统的能耗提出了要求,使得能耗管理变得必要.通常有两类动态的能耗管理技术:一类是对系统中的组件在其空闲时有选择的进行关闭或减慢[6];另一类是对系统的工作电压、工作频率的动态调整,即所说的动态电压调整(Dynamic Voltage Scaling,DVS)技术[7].

DVS技术是一种新型的节能机制,它根据应用的负载变化动态改变系统的运行电压和时钟频率,从而极大地降低系统能耗.目前,DVS技术已成为在系统工作时降低能耗的有效而流行的策略[3-4,7].

从实时特性考虑,实时系统的空闲时间(slack time)可以用来记检验点,以提高系统的可靠性;也可以用来进行动态电压调整,以降低系统的能耗.从系统可靠性考虑,为了减轻由高危险温度(high die temperature)对可靠性带来的影响,我们可以利用动态能量管理技术(如DVS技术)来处理,也可以采用容错技术(如记检验点技术)来处理.从容错和节能的关系考虑,工作电压的降低可以起到节能的效果,但工作电压的降低也会影响电路的稳定性,增加出错的可能性.

依据以上几个因素,我们应该同时考虑嵌入式实时系统的容错和节能.在此之前,容错技术和DVS技术通常是作为分离的主题在文献中出现的.例如,文献[1-2,5]等讲述了记检验点的容错技术而没有考虑节能的效果;而文献[3-4,7]等研究了DVS技术而没有考虑容错的能力.目前也只有文献[8-10]等少量文章综合考虑这两个方面,而且他们大多考虑周期性任务的处理,对于非周期任务研究的很少.

在文献[8]中,作者假设在任务执行过程中错误至多只出现一次,而且工作电压是可以连续变化的.同时,他们也没有考虑在记检验点和状态恢复过程中发生错误的情况.在文献[9-10]中,研究者对文献[8]中不合理的假设进行了改进,提出了处理至多k个错误的方案,但文献[9]是针对软实时系统的,文献[10]虽对文献[9]进行了改进,使之适应于硬实时系统,但其可调度性检测算法过于复杂.并且这些文献多关注周期性任务的处理,对非周期性任务的研究较少.

本文给出了一种简便的、考虑容错能力的非周期任务调度的可调度性检测方法CST(Checkpoint-based Schedulability Test),并将此方法与DVS技术结合,提出了一种嵌入式实时系统非周期任务的基于调度性检测的电压分配算法CST-VA(Voltage Allocation with Checkpoint-based Schedulability Test).

本文第2节给出问题的描述;第3节讲述考虑基于检验点容错的可调度性检测;第4节描述综合考虑容错和DVS技术的非周期任务调度及电压分配算法.相应的模拟实验及数据分析在第5节进行表述;第6节对文章进行总结,给出结论.

2 问题描述

本节在给出本文研究的假设前提的基础上,通过以下的形式化定义,对文章要解决的问题进行了描述.

定义1. 任务调度及电压分配问题包括系统的实时任务集J、系统的离散工作电压集V和离散时钟速度集S.任务集J是有限个非周期任务的集

合,记为 $J=\{J_1,J_2,\cdots,J_n\}$,其中 n 为系统中任务的个数.电压集 V 是系统中有限个离散工作电压的集合,记为 $V=\{V_1,V_2,\cdots,V_m\}$,其中 m 为系统中离散工作电压的个数.速度集 S 是系统中有限个离散时钟速度的集合,记为 $S=\{S_1,S_2,\cdots,S_m\}$,其中 S_i 是与工作电压 V_i 相对应的时钟速度.记 S 中最大的时钟速度为 f.

定义 2.　$S_i(t)$ 是 t 时刻任务 J_i 的处理器时钟速度,它满足 $S_i(t)\in S$.

定义 3.　一个非周期任务 J_i 是一个四元组,记为 $J_i=(A_i,D_i,R_i,k_i)$,其中,A_i 表示任务 J_i 的到达时间,D_i 表示任务 J_i 的截止期,且满足 $A_i\leq D_i$,R_i 表示在任务 J_i 执行不出现错误的情况下,CPU 以最大时钟速度 f 执行时所需的时钟周期数,k_i 表示任务 J_i 在执行中需要容忍的错误个数.

定义 4.　记 Cs 为记检验点的代价,是指保存检验点状态所需要的代价,用时钟周期数来衡量.

定义 5.　记 Cr 为状态恢复的代价,是指取出一个检验点所需要的代价,用时钟周期数来衡量.

为了研究的方便,我们提出以下假设:

(1) 任务集 J 中的任务是可抢占的.

(2) 任务集 J 在不出现错误的情况下是可调度的.

(3) 同一任务的检验点间隔是相同的.

(4) 错误为暂时性的[11-12],而且错误发生后能被及时检测到.

(5) 电压集 V 是非降序排列的,即 $V_1\leq V_2\leq\cdots\leq V_m$.

为了简便,我们以一个任务的分析为例.一个任务总的执行时间可以分为 3 个部分:有效计算的时间、记检验点所需要的时间和取回检验点所需要的时间[10].同样,一个任务在其执行过程中发生错误的情况也可以分为 3 种:在有效计算过程中出错、在记检验点的过程中出错和在取回检验点的过程中出错.

假设 m 个检验点被均匀地记录于一个任务,那么一个任务在出现 k 个错误时的最坏执行时间 $WCE(m)$ 是当这 k 个错误都发生在记检验点的过程中的情况下得到的[10].这时 $WCE(m)$ 可表示为

$$WCE(m)=\frac{R}{f}+m\frac{Cs}{f}+k\frac{(Cs+Cr)}{f}+k\frac{R}{\frac{f}{m+1}} \tag{1}$$

其中,R/f 是任务在不出现错误时的最坏执行时间;mCs/f 是记检验点的时间代价,$k(Cr+Cs)/f$ 是发生 k 次错误时回滚的时间代价;$kR/[f(m+1)]$ 是发生 k 次错误时任务重新执行的时间代价.

此时,理论上的空闲时间可以表示为 $S=D-A-WCE(m)$.我们的目标就是为系统中的每个任务 J_i 找到一个合适的 m 值,使得每个任务在理论上有最大的空闲时间用于 DVS 技术的应用,从而达到节能的目的.由上述分析,我们所研究的问题可描述如下.

任务调度及电压分配问题. 给出系统任务集和电压集的实例,为每个任务找到一个合适的记检验点的个数,使得实时任务可调度,并为相应的每个任务分配最佳的工作电压,使得系统总的能耗最小.

3　基于检验点的可调度性检测

要使得任务在理论上有最大的空闲时间,由上节的分析可知,必须使得式(1)取最小值.由数学方法可知,当 $m=\sqrt{kR/Cs}-1$ 时,式(1)最小.由于 m 所表示的实际含义,它必然是个非负整数,为此,我们给出下面的定义.

定义 6.　定义函数 F 使得 $F(WCE(m))=m$.

定义 7.　任务的最佳检验点个数,记为 m^*,它由下式确定:

$$m^*=\begin{cases}F(\min(WCE(\lfloor m\rfloor),WCE(\lceil m\rceil))), & \text{当}m>0\text{时},\\ 0, & \text{否则}.\end{cases}$$

由定义 7 可知,当计算出的 m 值为正时,我们取 $\sqrt{kR/Cs}-1$ 的上界整数和下界整数中使得 $WCE(m)$ 更小的那个 m 作为任务的最佳检验点个数;当计算出来的 m 值为负数或 0 时,任务的最佳检验点个数则取 0.

另外,我们应该看到,增加检验点的个数对系统的影响是双向的.一方面,增加检验点会增加记检验点的代价,这使得总的执行时间增长.另一方面,增加检验点可以缩短重新执行的时间,这使得总的执行时间减少.当一个任务由 m 个检验点增加到 $(m+1)$ 个检验点时,此时减少的总的执行时间为 $kR/[f(m+1)]-kR/[f(m+2)]=kR/[f(m+1)\cdot(m+2)]$.而此时增加的总的执行时间为 Cs/f.当过分增加检验点个数时,会使得总的执行时间增加的部分要大于减少的部分.因而,我们应该找到一个界限,使得在此范围内增加检验点的个数,能使得任务总的执行时间在出错的情况下得以减少.

此外,对于实时系统,每个任务都必须满足截止

期的要求.即在不出现错误和出现错误的情况下都应在截止期内完成执行.因此,我们得到嵌入式实时系统中各任务可调度的必要条件:

$$A+R/f+mCs/f+k(Cs+Cr)/f+kR/[f(m+1)]\leq D \quad (2)$$

$$A+R/f+mCs/f\leq D \quad (3)$$

$$Cs/f-kR/[f(m+1)(m+2)]\leq 0 \quad (4)$$

其中,不等式(2)和(3)分别保证任务在出现错误和不出现错误时,其执行满足截止期要求.不等式(4)保证记检验点的个数使得任务总的执行时间在出错的情况下得以减少.

引理 1. 如果不等式(2)和(3)都有解,则不等式(2)的解必包含于不等式(3)的解之中.即,如果不等式(2)的解为 $m_1\leq m\leq m_2$,不等式(3)的解为 $m\leq m_3$.则有 $m_2<m_3$.

证明. 假设 $m_2\geq m_3$.由式(3)知 $A+R/f+m_3Cs/f=D$.

又由式(2)知:$A+R/f+m_2Cs/f+k(Cs+Cr)/f+kR/[f(m_2+1)]=D$.

对于同一任务,因为 $m_2\geq m_3$,且上面两个等式成立,则必有 $k<0$.

而 k 表示任务可以承受的错误的个数,即 $k\geq 0$.两者相矛盾.

故假设不成立.从而此引理得证. 证毕.

引理 2. 不等式组(2)~(4)与不等式组(2)、(4)同解.

证明.

情形 1.当不等式(3)无解时

对于非负整数 k,不等式(2)必无解.

此时,前后两个不等式组都无解.

情形 2.当不等式(3)有解时

由引理 1 可知,不等式(2)的解必然包含于不等式(3)的解之中.

此时,前后两个不等式组也同解.

综上情况,引理得证. 证毕.

定理 1. 不满足不等式组(2)、(4)的任务必然是不可调度的.

证明.

情形 1.当不等式(2)无解时

即任务在出现 k 个错误时,其执行不能满足该任务的截止期要求,是不可调度的.

此时,不论不等式(4)是否有解,不等式组都无解.

情形 2.当不等式(4)无解时

即不能通过增加检验点来减少任务出错时的总的执行时间,也是不可调度的.

此时,不论不等式(2)是否有解,不等式组都无解.

综上情况,定理得证. 证毕.

另一方面,由于各任务间竞争系统资源,使得它们之间存在内在的关联,为此,我们需要考虑它们对系统可调度性带来的影响.由于我们假设任务集在不出现错误时是可调度的,为此,我们只需要考虑出现错误的情况下,各任务间的可调度性.

定义 8. 任务 J_i 的利用率,记为 u_i,它由下式确定:$u_i=WCE(m_i^*)/(D_i-A_i)$,其中 $WCE(m_i^*)$ 是任务 J_i 在出现错误时的最坏执行时间,A_i 和 D_i 分别是任务 J_i 的起始时间和截止期.

定义 9. 系统的利用率,记为 U,它由下式确定:$U=\max\{u_1,u_2,\cdots,u_n\}$,其中 n 为系统中非周期任务的个数,u_i 为各任务的利用率.

综合上述两个方面,对于硬实时系统,我们利用基于检验点的容错技术来检测任务级的可调度性,利用系统的利用率来检测系统级的可调度性.

此处,我们给出嵌入式实时系统基于检验点容错技术的可调度性检测算法 CST(Checkpoint-based Schedulability Test).

```
Procedure CST(J, f)
BEGIN
1. for each J_i in J, do {
1.1. calculate m* for each J_i;
1.2. solving the in-equations (2) and (4) for each J_i;
1.3. if (there is a J_i of which the in-equations have no
       solution or there is a m* which is not contained in
       the solution of that J_i)
       return (the task set is unschedulable);
1.4. else calculate u_i for J_i;
   }
2. calculate U;
3. if (U≤1)
     return (the task set is schedulable);
4. else return (the task set is unschedulable);
END
```

在算法 $CST(J,f)$ 中 J 是系统中非周期任务的集合,f 是系统中离散时钟速度集 S 中的最大时钟速度.算法的第 1 句进行任务级的调度性检测.其中,第 1.1 句为每个任务计算任务的最佳检验点个数.第 1.2 句进行必要条件的检查.第 1.3 句进行任务级调度的检测,并给出不可调度的返回信息.第

1.4 句为每个任务计算任务利用率.第 2 句计算系统的利用率.第 3 句进行系统级的调度性检测.第 4 句处理不能通过系统级调度检测情况的返回信息.算法 $CST(J,f)$的时间复杂度为 $O(n)$,其中 n 为系统中非周期性任务的个数.

4　电压分配算法

对于给定任务集 J 和系统的离散工作电压集 V,如何为每个任务分配最佳的工作电压,使得系统消耗的能量最小,这一直是被研究人员所关注的问题.文献[4]虽然给出了任务间电压分配问题的一种最优解,但是没有考虑系统容错的情况.为此,我们提出了一种考虑系统容错的任务集电压分配算法.

我们的首要目标是确保实时任务的截止期要求,即使是在出现 k 个错误的情况下.这就需要用到上述第 3 节的可调度性检测方法.其次的目标是使得系统总的能量消耗最小.此处,我们给出嵌入式实时系统基于检验点检测的电压分配算法 CST-VA (Voltage Allocation with Checkpoint-based Schedulability Test).

Procedure $CST\_VA(J,V,f)$
BEGIN
1. $CST(J,f)$;
2. if(the task set is schedulable){
2.1. for each J_i in J, do{
2.1.1. calculate the new R_i for each J_i in the presence of m^* checkpoints for the task;
2.1.2. generate an optimal continuous voltage for each J_i;
2.1.3. if (there is a task whose optimal continuous voltage is higher than V_m)
return(there is no feasible schedule);
}
3. return(the discrete voltage allocation and schedule);
END

在算法 $CST\text{-}VA(J,V,f)$中 J 是系统中非周期任务的集合,V 是系统中离散工作电压的集合,f 是系统中离散时钟速度集 S 中的最大时钟速度.算法的第 1 句调用 $CST(J,f)$算法对系统任务集进行可调度性检测,即检测系统在保证各任务截止期要求的情况下,是否能通过为每个任务记合适的检验点来提高系统的可靠性.算法的第 2 句是对可调度情况的进一步处理.其中,第 2.1.1 句计算每个任务在记最佳检验点个数 m^* 的情况下的最坏执行时间 R_i.第 2.1.2 句利用计算得到的 R_i,为每个任务计算它的理想分配电压.(这个理想分配电压不一定是系统所提供的离散工作电压集 V 中的电压.)第 2.1.3 句检查所得到的理想分配电压是否超出了系统所提供的离散工作电压的范围.如果超出了电压范围,则认为该任务不可调度.第 3 句是利用第 2 句中得到的信息,给出每个任务的离散工作电压和调度信息.算法的时间复杂度为 $O(n\log^2 n)$,其中 n 为系统中非周期性任务的个数.

这里需要指出,在进行了 CST 后必须在 CST-VA 中再次进行可调度性检测.这是因为,当我们采用记检验点的容错技术并满足了 CST 时,任务在考虑容错情况下所需 CPU 周期的最坏执行个数 R_i 会有一定程度的增加,这会直接影响理想电压的取值.当我们发现有任务的理想电压超过系统提供的工作电压集 V 中的最大值 V_m 时,就不能为该任务分配合适的工作电压.而对于小于 V 中最小值 V_1 的理想电压,我们可以利用零电压和 V_1 来处理.

对于如何得到一个任务的理想电压以及如何将该理想电压转换为两个相邻的系统所提供的离散工作电压,限于篇幅就不在本文详述,具体信息请阅读文献[4].

5　模拟实验分析

本节将 CST-VA 算法和文献[4]中提出的 Alloc-vt算法以及文献[8]中提到的 FT-Only,FT+EC 算法进行性能比较.分别从可靠性和能耗特性两个方面进行比较.

表 1 给出了系统任务集的一个实例,其中 A_i 表示任务 J_i 的到达时间;D_i 表示任务 J_i 的截止期;R_i 表示在任务 J_i 不出现错误的情况下,CPU 以最大时钟速度执行时所需的时钟周期数;k_i 表示任务 J_i 在执行时需要容忍的错误个数.为了研究的简便,我们假设时钟速度与工作电压成线性关系,而且能耗与时钟速度的平方成正比.系统的工作电压集 $V=\{3.0V,5.0V,7.0V,9.0V,12.0V\}$,对应的时钟速度集 $S=\{30MHz,50MHz,70MHz,90MHz,120MHz\}$.为了量化系统能耗,我们将功率函数 $P(s)$ 规范化为

表 1　任务集的一个实例

任务	A_i	D_i	R_i	k_i
J_1	0	11	150	4
J_2	3	8	120	2
J_3	5	8	180	1
J_4	9	11	80	1

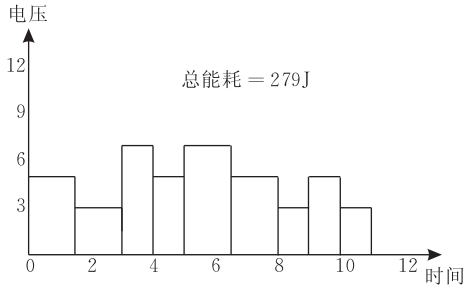
P(10MHz)=1J/s.假设记检验点的代价 $Cs=6$M 个时钟周期,状态恢复的代价 $Cr=6$M 个时钟周期.

5.1 CST-VA 算法与 Alloc-vt 算法的比较

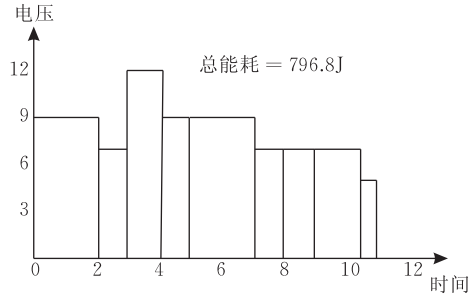
该任务集实例在两种算法下的理想电压由表 2 给出.同时,图 1 给出了此任务集实例在两种算法下具体的电压分配情况.其中图 1(a)是不考虑错误出现情况下 Alloc-vt 算法[4]的分配结果,图 1(b)是考虑系统容错情况下 CST-VA 算法的分配结果.

表 2 任务集实例在两种算法下所对应的理想电压值

任务	Alloc-vt 算法的电压/V	CST-VA 算法的电压/V
J_1	3.75	8.05
J_2	6.0	10.7
J_3	6.0	8.4
J_4	4.0	6.5



(a) Alloc-vt算法的分配结果



(b) CST-VA算法的分配结果

图 1 任务集实例在两种算法下的电压分配情况

如图 1 中所示,当不考虑错误时任务集工作于较低的工作电压{3.0V,5.0V,7.0V},当考虑容错时任务集工作于较高的工作电压{5.0V,7.0V,9.0V,12.0V}.这同理论分析是一致的.因为,当考虑容错时,各任务的最坏执行时间会有所增加,为了保证任务的实时特性,必然会工作于较高的工作电压下.

对应于两种电压分配方法下系统的能耗分别为 279J 和 796.8J.可以看到,我们所提出的算法对系统能耗有相应的增加,这同理论分析是一致的.因为,当考虑容错时,各任务工作于相对较高的电压,所消耗的能量自然有所增加.但是这种增加是有限的,而且能有效提高系统的可靠性.对于该任务集实例,各任务分别可以承受 11 个、2 个、4 个和 5 个错误的出现.

图 2 给出了任务集实例中各任务随 k 值不同对最佳检验点个数 m^* 带来的影响.当 k 变大时,相应的 m^* 随之变大,当 k 大到一定值时,找不到对应的 m^* 值.对一个任务,当 k 值增加时,表示该任务必须承受的错误个数增加,为保证任务的可靠性,相应的记检验点的个数必然会增加.然而当 k 值增大到一定程度后,利用添加检验点已经不能在任务截止期内保证其可靠性.这同理论分析也是一致的.我们可以从第二、三部分看到,m^* 是 k 的一个单调增函数,同时 $WCE(m)$ 也会随 k,m 的增加而增加,当 k,m 增到一定程度时,任务必然会超过其截止期要求.

在此,可以从图 2 中反映第 4 节所说明的需要在 CST-VA 算法中再次进行可调度性检测的问题.例如,对于任务 J_2,当 k 值为 3 时,由 CST 可得到相应的 m^* 值为 7,但在 CST-VA 算法中,对于 $k=3$,$m^*=7$,计算所得的理想电压值 $V=12.15$V,比系统工作的最大电压还大,是不可调度的.造成这一问题的原因在于任务的可抢占性.在 $3\leq t\leq 5$ 时,任务 J_2 执行,而当 $5\leq t\leq 8$ 时,任务 J_3 抢占执行,导致任务 J_2 的实际执行时间只有 2 个时间单位.

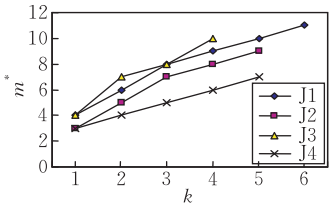


图 2 检验点个数随错误个数的变化关系图

图 3 给出了记检验点的代价 Cs 对检验点个数带来的影响.当 Cs 变大时,相应的 m^* 随之变小,当 Cs 变大到一定程度时,找不到对应的 m^* 值.当记检验点的代价变大时,从任务的实时特性考虑,应减少记检验点的个数.然而当 Cs 增大到一定程度后,利用添加检验点已经不能在任务截止期内保证其可靠

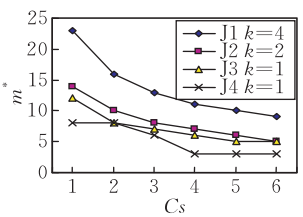


图 3 检验点个数随记检验点代价的变化关系图

性.这同理论分析也是一致的.我们从第 2、3 节可以看到,m^* 是 Cs 的一个单调减函数,同时 $WCE(m)$ 也会随 Cs 的变大而增加,当 Cs 增大到一定程度时,任务必然会超过其截止期要求.所幸,目前的研究表明,Cs 远小于各任务的 R_i[8,10].

从上面的数据可以看到,在增加了有限的系统能耗的情况下,CST-VA 算法比 Alloc-vt 算法较大程度地提高了系统的可靠性.

5.2　CST-VA 算法与 FT-Only、FT+EC 算法的比较

另一方面,我们将 CST-VA 算法与只考虑容错而不考虑节能的 FT-Only 算法[8]以及同时考虑容错和节能的 FT+EC 算法[8]进行了比较.表 3 给出了采用 3 种不同方法时所记的检验点个数(为了同文献[8]中的算法进行比较,此时考虑各任务的 $k_i=1$).其中,在 FT-Only 算法下的检验点个数是在任务工作于最大速度时所需检验点的最小个数(其取值的计算参考文献[8]).在 FT+EC 算法下的检验点个数如文献[8]所述计算得到.在 CST-VA 算法下的检验点个数是我们利用本文提出的方法计算得到的最佳检验点个数.表 3 同时给出了在 3 种方法下任务的能量消耗,我们记任务在 FT-Only 算法下的能量消耗为 E_1,在 FT+EC 算法下的能量消耗为 E_2,在 CST-VA 算法下的能量消耗为 E_3.

表 3　任务集在 3 种不同算法下所记的检验点个数及消耗的能量

任务	FT-Only	E_1/J	FT+EC	E_2/J	CST-VA	E_3/J
J_1	1	367.2	1	226.8	4	119.2
J_2	1	295.2	1	185.6	3	162
J_3	2	338.4	3	228.8	4	214.2
J_4	1	199.2	2	98.8	3	86

从表 3 的数据可以看到,在提供相同的系统可靠性时,CST-VA 算法比 FT-Only,FT+EC 算法能有效地节省系统能耗.同时,由于 FT-Only 和 FT+EC 算法的不足,CST-VA 算法可较大程度地提高系统的可靠性.

6　结　论

嵌入式实时系统中容错和节能的综合研究正在吸引众多研究者的兴趣.其特点在于,系统既要求满足实时特性,又要求一定的容错能力,同时还要求使得系统的能耗最小.本文重点研究了嵌入式实时系统中如何达到上述 3 大目标的非周期任务调度和电压分配问题.在基于检验点容错技术的可调度性检测基础上,提出了对于系统非周期任务的基于检验点检测的电压分配算法 CST-VA.该算法在保证任务实时性的同时提高了系统的可靠性,也保证了系统能耗的节省.模拟实验表明,CST-VA 比 Alloc-vt 较大程度地提高了系统的可靠性;比 FT-Only 和 FT+EC 在相同可靠性前提下降低了系统能耗,同时比两者有更好的可靠性,更适合于嵌入式实时系统的需要.

对于进一步的工作,我们有以下几个方向的考虑.(1)为了问题的简化,我们在本文中考虑各任务都具有相同的平均开关电容(average switched capacitance),在今后的研究中我们会考虑具有不同平均开关电容的任务集的容错调度和节能电压分配.(2)在以后的工作中我们会考虑将本文提出的对非周期任务的处理与周期性任务的处理统一起来.(3)在将来的工作中我们会研究非均匀检验点策略是否会对本文所描述的问题有更好的解决办法.

参　考　文　献

[1] Lee H, Shin H, Min S. Worst case timing requirement of real-time tasks with time redundancy//Proceedings of the 6th International Conference on Real-Time Computing Systems and Applications. Hong Kong, China, 1999: 410-414

[2] Pradhan D K. Fault Tolerance Computing: Theory and Techniques. Old Tappan, NJ: Prentice Hall, 1986

[3] Ishihara T, Yasuura H. Voltage scheduling problem for dynamically variable voltage processors//Proceedings of the 1998 International Symposium on Low Power Electronics and Design. New York, USA, 1998: 197-202

[4] Kwon W C, Kim T. Optimal voltage allocation techniques for dynamically variable voltage processors//Proceedings of the 2003 Design Automation Conference. California, USA, 2003: 125-130

[5] Siewiorek D, Swarz R. Reliable Computer Systems: Design and Evaluation. Natick, MA: Peters A K, Ltd., 1998

[6] Benini L, Bogliolo A, Micheli G D. A survey of design techniques for system-level dynamic power management. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2000, 8(3): 299-316

[7] Hong I, Kirovski D, Qu G et al. Power optimization of variable-voltage core-based systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1999, 18(12): 1702-1714

[8] Melhem R, Mosse D, Elnozahy E N. The interplay of power management and fault recovery in real-time systems. IEEE Transactions on Computers, 2004, 53(2): 217-231

[9] Zhang Y, Chakrabarty K. Energy-aware adaptive checkpointing in embedded real-time systems//Proceedings of the 2003 Design, Automation and Test in Europe Conference and Exhibition. Paris, France, 2003: 918-923

[10] Zhang Y, Chakrabarty K. A unified approach for fault tolerance and dynamic power management in fixed-priority real-time embedded systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2006, 25(1): 111-125

[11] Castillo X, McConnel S, Siewiorek D. Derivation and calibration of a transient error reliability model. IEEE Transactions on Computers, 1982, 31(7): 658-671

[12] Dupont E, Nicolaidis M, Rohr P. Embedded robustness ips for transient-error-free ics. IEEE Transactions on Design and Test of Computers, 2002, 19(3): 54-68

LI Guo-Hui, born in 1973, Ph. D., professor, Ph. D. supervisor. His current research interests include advanced database technology, real-time systems, mobile real-time computing and software engineering.

YANG Bing, born in 1975, Ph. D.. His current research interests focus on mobile real-time database systems.

HU Fang-Xiao, born in 1983, Ph. D. candidate. His current research interests include combination of fault-tolerance and power management in real-time systems, transaction processing in mobile real-time databases.

XU Hua-Jie, born in 1976, Ph. D., lecturer. His main research interests focus on issues in sensor networks.

DU Jian-Qiang, born in 1968, professor. His main research interests focus on medical imaging.

Background

The work is supported in part by the National Natural Science Foundation of China under the grant No. 60873030; the National High Technology Research and Development Program (863 Program) of China under the grant No. 2007AA01Z309; and Special Funds of Central Colleges for Basic Scientific Research and Operating Expenses.

The problem of task scheduling and voltage allocation, which should minimize the total processor energy consumption while guaranteeing timing constraints of tasks, is a representative requirement for embedded real-time systems. At present, researchers are interested in the trade-off between system reliability and energy consumption for real-time systems. The slack time in real-time systems can be used by recovery schemes to increase system reliability as well as by frequency and voltage scaling techniques to save energy. Moreover, the rate of transient faults also depends on system operating frequency and supply voltage. So, it gives us the new challenge for considering the combination of system reliability and energy consumption for the problem of task scheduling and voltage allocation.

Fault tolerance through redundancy as well as energy management through frequency and voltage scaling has been well studied separately in the context of real-time embedded systems. However, there are relatively less researches addressing the combination of fault tolerance and energy management. Furthermore, most of the existing methods are concerned about periodic task set only.

In many real-time embedded systems, tasks can be classified as periodic tasks and aperiodic tasks. Aperiodic tasks can be processed with general scheduling algorithms. However, in many cases, it is more efficient to process aperiodic tasks with separate algorithms which make use of the knowledge that aperiodic tasks may themselves have a variety of timing requirements.

This paper focuses on the variable voltage allocation problem for aperiodic task set in embedded real-time systems based on earliest deadline first (EDF) scheduling strategy. First, the authors present a checkpointing-based schedulability test, and with this test, provide a deterministic guarantee and get the optimal number of checkpoints that a task should be inserted if possible. The optimal number of checkpoints can help the task to guarantee the timing constraints and minimize the worst case execution time in the present of transient faults.

Based on the checkpointing-based schedulability test, the authors propose a voltage allocation method through dynamic voltage and frequency scaling (DVFS) technique. The method not only guarantees timing constraints, but also provides higher system reliability through checkpointing and minimizes energy consumption via optimal voltage allocation techniques.

This paper contributes on the problem of task scheduling and voltage allocation focusing on aperiodic tasks for real-time embedded systems. The simulation results show that the proposed method has significantly better performance compared with the existing voltage allocation technique and is more appropriate for embedded real-time systems.