混合关键级任务资源需求的概率性分析

张 通,郑 浩,朱长昊,张凤登

(上海理工大学 光电信息与计算机工程学院,上海 200093)

摘 要:混合关键级系统可调度性分析通常基于最坏情况执行时间,导致系统资源的过度预置和低关键级模式下的分析过于苛刻。为解决资源过度预置的问题,简化低关键级模式下的可调度性,针对单核处理器上EDF调度的混合关键级零星任务集,提出随机需求约束函数模型。分析了混合关键级系统的概率性资源需求,得到相应的可调度性条件,并在尽可能增加低关键级任务预算的同时设计出系统可调度性测试算法。实验结果表明,与之前的可调度接受率相比,该方法提高了32%,在降低pDBF模型算法复杂度基础上仍使可调度性得到显著改善。

关键词:混合关键级;需求约束函数;概率性分析;可调度性;调度算法

DOI:10.11907/rjdk.211368 **开放科学(资源服务)标识码(OSID)**:

中图分类号:TP301 **文献标识码**:A **文章编号**:1672-7800(2022)001-0156-08

Probabilistic Analysis of Resource Requirements for Mixed-Criticality Tasks

ZHANG Tong, ZHENG Hao, ZHU Chang-hao, ZHANG Feng-deng

(*School of Optical-Electrical & Computer Engineering*, *University of Shanghai for Science & Technology*, *Shanghai* 200093, *China*)

Abstract: The mixed-criticality systems' schedulability analysis is normally on the basis of worst case execution time (WCET), which leads to the problem of over-provisioning for resources and causes quite pessimistic analysis in low-criticality mode, In order to solve the problem of over-provisioning for resources and simplify the schedulability analysis method in low critical level mode. In this paper we propose a probabilistic Demand Bound Function (pDBF) model for the sporadic task sets scheduled by EDF scheduling policy on one processor. we present how to analyze and calculate the pDBF of the mixed-criticality system with an example, and the corresponding schedulability conditions are derived. An algorithm of schedulability testing with respect to sporadic task systems is designed with concerning maximizing the execution-budget of low-criticality task. Evaluations illustrate that our model can significantly improve the schedulability, meanwhile, reducing the complexity of the testing algorithm through the existing method. The experimental results show that the schedulability acceptance rate can be improved by 32% compared with the previous deterministic analysis. The proposed method can reduce the complexity of the analysis algorithm in pDBF model, and also improve the schedulability significantly.

Key Words: mixed-criticality; demand bound function; probabilistic analysis; schedulability; scheduling algorithm

0 引言

混合关键级系统[1-4](Mixed-Criticality Systems,MCS)在运行资源宽裕的条件下,系统不再单纯为保证安全关键性任务(高关键级任务)运行,而是尽力保证非安全关键性任务(低关键级任务)的服务质量。任务在MCS中运行一般通过执行预算[5-6](Execution Budgets)控制,所有任务运行时都不会超过预算。任务执行预算通常根据任务的最坏情况执行时间(Worst Case Execution Time,WCET)进行安排,这导致低关键级任务不必要地被赋予了一定的安全关键性。本文将概率性最坏情况执行时间[7-8](probabilistic Worst Case Execution Time,pWCET)引入传统的MCS模型,通过概率性实时分析方法研究系统的可调度性[9]。概率性实时分析中将任务资源调度失败视为系统失效,高关键级任务错过截止期概率可设定在某个极其低的水平(如

收稿日期:2021-03-12

基金项目:上海市自然科学基金项目(15ZR1429300)

作者简介:张通(1992-),男,上海理工大学光电信息与计算机工程学院硕士研究生,研究方向为分布式实时系统;郑浩(1996-),男,上海理工大学光电信息与计算机工程学院硕士研究生,研究方向为FPGA嵌入式设计、实时以太网;朱长昊(1996-),男,上海理工大学光电信息与计算机工程学院硕士研究生,研究方向为多核处理器实时调度研究;张凤登(1963-),男,博士,上海理工大学光电信息与计算机工程学院教授,研究方向为汽车电子与现场总线。本文通讯作者:郑浩。

10^{-9}/h);低关键级任务则允许更高的水平(如10^{-6}/h)。现实中安全关键性系统也存在随机性行为,如在先进硬件架构多级缓存中的数据随机替换策略或倒车泊车雷达中的随机频率声波[10]。

本文提出了概率性需求边界函数模型(probabilistic Demand Bound Function,pDBF),通过分析系统整体的资源需求过载概率来进行可调度性分析。考虑任务执行预算与pWCET的联系,说明在MCS的不同模式下如何得到pDBF,丰富了MCS模型;设计了针对混合关键级零星任务集的可调度性测试算法,分析了算法复杂度以及系统其他参数对可调度性的影响。

1　相关工作

文献[11]提出了概率性时间需求分析方法(Probabilistic Time Demand Analysis, PTDA),针对弱实时或混合实时系统进行研究;文献[12]开创性地针对WCET采用概率性分析方法,以避免系统资源过度预置;以概率性最坏情况响应时间分析为代表,文献[13-15]针对固定任务优先级调度策略下的周期性任务系统采用不同于pWCET的设定,这里假设任务所有可能执行时间的概率分布已知。在考虑任务间概率性抢占下,计算概率性最坏情况响应时间。相关文献中任务模型拓展至包含pWCET与pMIT。另外就是一类以实时接口(real-time interface)分析模型为基础的研究,文献[16]基于此提出了概率性实时演算框架(probabilistic real-time calculus),同时考虑了pWCET与pMIT,并同样基于需求边界函数提出了针对EDF调度的可调度性充分条件,但该框架仅仅从pWCET或pMIT的最极端部分分析出发,从而得到的是可调度性的下界。本文主要关注任务最坏情况执行时间,用pWCET表示,采用类似已有文献的计算原理。

除了围绕pWCET分析外,还涉及任务概率性最小到达间隔或概率性截止期[17];文献[18]将一段时间内任务到达次数用随机变量表示,缺点是系统模型信息不如其他模型参数丰富。pWCET的获取可以使用静态概率性时间分析(Static Probabilistic Timing Analysis,SPTA)方法。上述研究假设任务与任务之间的概率性参数独立;pWCET一般以整数值的离散分布形式给出,本文同样遵循该设定。

针对固定任务优先级调度的MCS,文献[19]采用了类似文献方法,通过计算每个超周期(hyperperiod)内所有任务的错过截止期概率得到可调度的分析结果;此外文献[20]改进混合关键级系统中SMC与AMC策略,计算所有模式下任务的错过截止期概率;针对EDF调度,文献[21]根据pWCET设定了预算超出概率,改善了系统的可调度性;部分研究将概率性实时分析引入能耗约束的MCS调度设计中,文献[22]提出了基于测量估计的概率性分析来估计任务最坏情况下的能量消耗(Worst-Case Energy Consumption,WCEC),但没有结合具体的能耗资源管理算法;Bhuiyan等[23]基于概率性方法提出了能耗影响的速率控制,并结合动态电压频率调节技术(Dynamic Voltage and Frequency Scaling,DVFS)实现了系统能耗最小化;关于多核处理器平台MCS,文献[24]基于概率性的DAG分析量化MCS中低关键级任务;Zeng等[25]在同构多核处理器平台上,提出了概率性混合关键级任务模型,得到相应的任务分区算法PPDC,使低关键级任务运行得到提升。本文提出的概率性需求边界函数模型,可计算任务系统在调度周期内的资源需求过载概率,在避免资源过度预置的同时改善系统可调度性。

2　符号与模型定义

2.1　符号定义与概念

pWCET为整数取值的离散型随机变量,可用概率质量函数(Probability Mass Function,PMF)表示,即$f_i(c_i)=P(\mathcal{C}_i=c_i)=p_i$,i=0,1,…,k,或记为:

$$C_i=\begin{pmatrix} c_0=c^{min} & c_1 & \cdots & c_k=c^{max} \\ p_0 & p_1 & \cdots & p_k \end{pmatrix}, \tag{1}$$

规范假设$c_0<c_1<\cdots<c_k$,$\mathcal{C}_i$的范围为$[c^{min}, c^{max}]$。p_i须满足$p_i\geqslant 0$,i=1, 2, …, k,且Σk i=1p_i=1。概率分布函数(Cumulative Distribution Function,CDF)表示随机变量小于某个值的概率和,即$F_i(x)=\Sigma x\ y=x^{min}f_i(y)$,$x\in[c^{min},\infty)$。两个相互独立随机变量X与Y的卷积和定义为P(Z=z)=$\Sigma\infty$ k=$-\infty$P(X=k)P(Y=z-k),记为Z=X$\otimes$Y;相应卷积差记为Z=X$\ominus$Y=X$\otimes$(-Y);若干部分概率和为1的随机变量分布的合并称为联合,记为Z=X$\oplus$Y。

随机变量X_1和X_2间的比较,若对任意x两个随机变量的CDF满足$FX_1(x)\leqslant FX_2(x)$,则称X_1大于等于X_2,记为$X_1\geqslant X_2$,在CDF曲线上X_1始终位于X_2下方;反之称X_1小于等于X_2,记为$X_1\leqslant X_2$。约定x与y最大值为$\llbracket x\rrbracket_y$,x与y最小值为$\llbracket x\rrbracket^y$。

2.2　任务系统模型

在单处理器硬件平台上,任务系统由一组个数为n的零星任务集$\tau=\{\tau_1,\tau_2,\cdots,\tau_n\}$构成。单个任务$\tau_i$用($T_i$, D_i, $\mathcal{C}_i$, L_i)元组定义,分别表示任务周期、概率性最坏情况执行时间、相对截止期以及任务关键级。系统实际运行是通过设置执行预算的方式,所有任务在低关键级模式下执行预算为B_i。本文限制为双关键级系统,即$L_i=\{LO, HI\}$。任务τ_i的第j个作业记为$J_{i,j}$。任务$D_i\leqslant T_i$,$c^{max}\leqslant D_i$。任务与任务之间不存在顺序或互斥关系;任务不同作业之间相互独立。

定义低关键级任务子集$\tau^L=\{\tau_i\in\tau|\ L_i=LO\}$,高关键级任务子集$\tau^H=\{\tau_i\in\tau|\ L_i=HI\}$。定义任务$\tau_i$的平均利用率与系统$\tau$的平均利用率为:

$$U_{\tau_i}^{avg}\overset{def}{=\!=}\bar{C}_i/T_i, \tag{2}$$

$$U_{\tau}^{avg}\overset{def}{=\!=}\sum\nolimits_{\tau_i\in\tau}U_{\tau_i}^{avg}, \tag{3}$$

系统从低关键级模式开始运行,当任务作业实际执行时间未超过执行预算B_i时,系统处于低关键级模式;当高关

键级作业执行超过其B_i时，提升系统关键级模式。上述系统模式切换是基于内部触发的机制，还有一种外部触发机制[26]，后者认为外部事件也能导致系统被动地提升关键级。模式切换后所有低关键级任务作业会被抛弃，并且在高关键级模式下不会释放；系统运行在高关键级模式时，若出现调度空闲，则系统关键级回落。

3 概率性需求边界函数

3.1 概率性需求边界函数定义

对于零星或周期任务，任意时间范围Δ内，任务τ_i的需求边界函数[27]（Demand Bound Function，DBF）为：

$$dbf(\tau_i,\Delta)=\left\lfloor\frac{\Delta+T_i-D_i}{T_i}\right\rfloor\cdot C_i,\forall\Delta\geqslant 0, \tag{4}$$

其中，C_i是任务WCET，从而系统τ的需求边界函数为：

$$\mathrm{dbf}(\tau,\Delta)=\sum_{\forall\tau_i\in\tau}\mathrm{dbf}(\tau_i,\Delta),\forall\Delta\geqslant 0, \tag{5}$$

对于固定速率单处理器，任意时间范围Δ内系统提供资源可用供给边界函数[28]（Supply Bound Function，SBF）来描述，即：

$$\mathrm{sbf}(\tau,\Delta)=\Delta,\forall\Delta\geqslant 0 \tag{6}$$

定理1：若任务系统τ在任意时间范围Δ内的需求边界函数总是不大于系统的供给边界函数，则在EDF算法调度下任务系统τ是可调度的，即满足：

$$\mathrm{dbf}(\tau,\Delta)\leqslant\mathrm{sbf}(\tau,\Delta),\forall\Delta\geqslant 0 \tag{7}$$

定义1：若任务的最坏情况执行时间使用pWCET描述，则任意时间范围Δ内，该任务的需求边界函数簇可以用概率性需求边界函数（pDBF）描述如下：

$$\mathrm{pdbf}(\tau_i,\Delta)=\overset{nx}{\otimes}C_i,\forall\Delta\geqslant 0 \tag{8}$$

其中，$\overset{nx}{\otimes}\mathcal{C}_i=\mathcal{C}_i\otimes\ldots\otimes\mathcal{C}_i$表示对$\mathcal{C}_i$卷积和$nx$次，$nx=\lfloor(\Delta+T_i-D_i)/T_i\rfloor$。整个任务系统的pDBF为：

$$\mathrm{pdbf}(\tau,\Delta)=\bigotimes_{\forall\tau_i\in\tau}\mathrm{pdbf}(\tau_i,\Delta),\forall\Delta\geqslant 0 \tag{9}$$

通过例1说明pDBF暂时不设任务关键级。将DBF分析中使用的WCET设为表1中黑色方框内数值，此时系统利用率大于1，在EDF算法下不可调度[29]。

Table 1 Model task set of example 1-pDBF

表1 例1pDBF模型任务集

τ	C_i	D_i	T_i
τ_1	$C_1=\begin{pmatrix}1 & 2\\ 0.9 & 0.1\end{pmatrix}$	5	5
τ_2	$C_2=\begin{pmatrix}1 & 3\\ 0.9 & 0.1\end{pmatrix}$	8	8
τ_3	$C_3=\begin{pmatrix}2 & 4\\ 0.8 & 0.2\end{pmatrix}$	10	10

以Δ=10为例，按式（1）和式（2），任务系统DBF为11，任务系统不可调度；同样对于Δ=10，各个任务的pDBF分别为：

$$\mathrm{pdbf}(\tau_1,10)=\begin{pmatrix}2 & 3 & 4\\ 0.81 & 0.18 & 0.01\end{pmatrix}, \tag{10}$$

$$\mathrm{pdbf}(\tau_2,10)=\begin{pmatrix}1 & 3\\ 0.9 & 0.1\end{pmatrix}, \tag{11}$$

$$\mathrm{pdbf}(\tau_3,10)=\begin{pmatrix}2 & 4\\ 0.8 & 0.2\end{pmatrix}, \tag{12}$$

整个任务系统的pDBF为：

$$\mathrm{pdbf}(\tau,10)=\begin{pmatrix}5 & 6 & 7 & 8 & 9 & 10 & 11\\ 0.5832 & 0.1296 & 0.2178 & 0.0468 & 0.0188 & 0.0036 & 0.0002\end{pmatrix} \tag{13}$$

图1表示一个超周期内任务系统的DBF和pDBF。任务系统pDBF所有可能值如图中灰度部分所示，对应概率越大则灰度越深。随着时间范围的增大，出现极端资源需求情况的概率越来越小。pDBF模型仍然包含了DBF模型中所有信息，任务系统在Δ=10时发生资源需求过载的概率（Demand Overload Probability，DOP）为0.000 2，如果允许任务系统以不大于某个概率阈值H_T发生资源需求过载，如0.001，那么例1的可调度性将放宽。

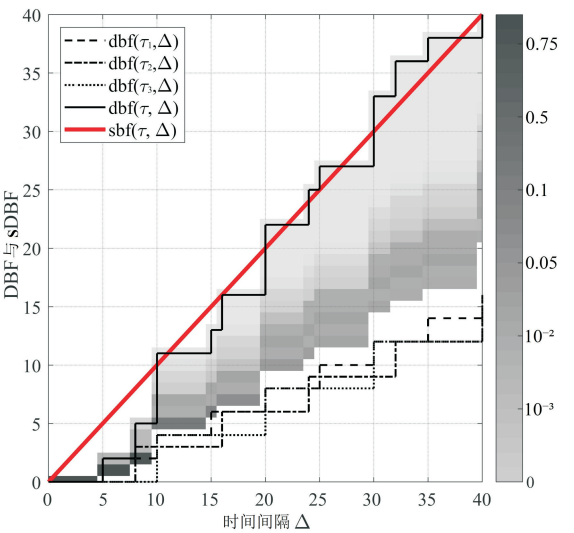

Fig. 1 The pDBF of the example task set and the compared DBF

图1 示例任务集的pDBF和与其相比较的DBF

3.2 可调度性分析

先给出几个相关定义。某个任务$\tau_i=(T_i, D_i, \mathcal{C}_i)$的第j次作业记为$J_{i,j}=(t_{i,j}, r_{i,j})$，$t_{i,j}$表示作业释放时刻，$r_{i,j}$表示作业响应完成时刻，如果$J_{i,j}$完成时刻大于其绝对截止期时刻，即$r_{i,j}>t_{i,j}+D_i$，则所有可能的$r_{i,j}$对应的概率之和称为$J_{i,j}$错过截止期概率（Deadline Miss Probability，DMP），即$\mathrm{DMP}_{i,j}=\mathrm{P}(\mathrm{R}_{i,j}>\mathrm{t}_{i,j}+\mathrm{D}_i)$，$\mathrm{R}_{i,j}$表示作业$\mathrm{J}_{i,j}$所有可能最坏响应完成时刻所构成的随机变量。

定义2：对任意时间范围Δ，若任务系统τ的需求边界函数大于供给边界函数，则称任务系统发生需求过载；任务系统pDBF大于SBF这部分概率和的最大值，称τ为在Δ内的需求过载概率，记为$\mathrm{DOP}_{\tau,\Delta}$，即：

$$\mathrm{DOP}_{\tau\Delta}=\left[\!\left[P(\mathrm{pdbf}(\tau,t)>t)\right]\!\right]_0,\forall\Delta\geqslant 0,t\in[0,\Delta], \tag{14}$$

下面证明一个引理，将$\mathrm{DOP}_{\tau,\Delta}$与任务作业的$\mathrm{DMP}_{i,j}$联系起来。

引理1 若任务系统τ在任意时间范围Δ内的需求过载

概率 $DOP_{\tau,\Delta}$ 不大于某个概率阈值 H_T,则在该时间范围Δ内所有任务作业错过截止期概率也一定不大于 H_T,即满足:

$$H_T \geqslant DOP_{\tau,\Delta} \geqslant DMP_{i,j}, \forall \Delta \geqslant 0, \forall \tau_i \in \tau, \quad (15)$$

其中任务作业须满足 $t_{i,j}+D_i \leqslant \Delta$。

证明:首先考虑 $\Delta \geqslant max\{D_1, D_2, \cdots, D_n\}=D^{max}$ 的情况。设 τ 在Δ内的 $DOP_{\tau,\Delta} \leqslant H_T$ 但大于0,根据定理1,任务作业错过截止期的可能性是必然存在的。将系统报告出现调度失效的最早时刻记为 $t_f \in (D^{max}, \Delta]$,则 $H_T \leqslant DOP_{\tau,tf} \leqslant DOP_{\tau,\Delta}$。在 t_f 时刻之前根据EDF调度策略,以 t_f 为绝对截止期的任务作业将在所有之前已释放作业中被调度。设这部分作业为 $J_{r_{i,j}=t_f}=\{J_1, J_2, \cdots, J_m\}$(来自不同任务省去作业序号下标),此时错过截止期的作业全部或至少一个来自 $J_{r_{i,j}=t_f}$。以 t_f 之前时刻为绝对截止期的任务作业,自然在 t_f 之前就完成调度或者报告调度失败,这里分两种情况讨论:

情况1:只有作业 J_1 在 t_f 时刻可能错过截止期,J_1 错过截止期的概率为 DMP_1,显然 $DOP_{\tau,tf}=DMP_1$。

情况2:可能错过截止期的作业有k(≤m)个,错过截止期概率分别为 $DMP_1, DMP_2, \cdots, DMP_k$。由于作业绝对截止期都相同,所以任务ID最小的作业尽管已经被优先调度,但仍可能以 DMP_1 概率错过截止期,其他优先级更低的任务将肯定得不到调度而错过截止期,所以有 $DMP_1 \leqslant DMP_2 \leqslant \cdots \leqslant DMP_k$。任何无法完成调度的作业最后都会造成DOP不为0,所以 $DOP_{\tau,tf}=max\{DMP_1, DMP_2, \cdots, DMP_k\}=DMP_k$。

$\Delta < max\{D_1, D_2, \cdots, D_n\}$ 相当于缩小任务集范围,上述分析同样适用。综上所述,$H_T \geqslant DOP_{\tau,\Delta} \geqslant DOP_{\tau,tf} \geqslant DMP_{i,j}, \forall \Delta \geqslant 0, \forall \tau_i \in \tau$。

将例1中任务截止期进一步设值为 $D_1=3, D_2=7, D_3=7$。图2考虑Δ=8这一情况,t_7 时刻就是该情况下的 t_f,此时任务 $J_{r_{i,j}=t_f}=\{J_1, J_2\}$,分别属于 τ_2 与 τ_3。此时 $DMP_2=0, DMP_3=0.02$, $DOP_{\tau,\Delta}=0.0226$。

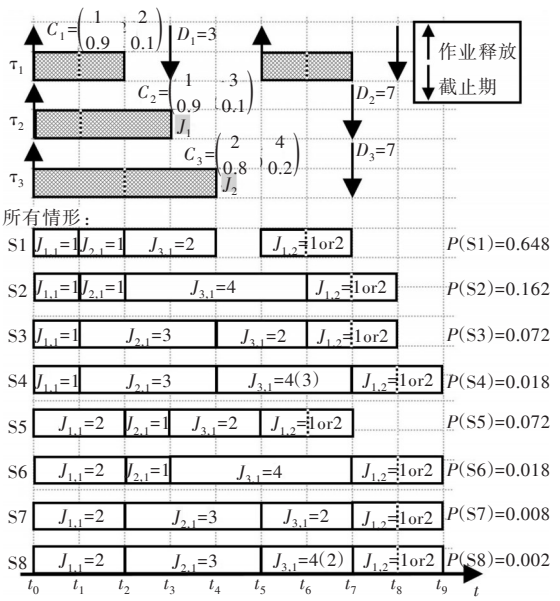

Fig. 2　An example of the relationship between system DOP and task DMP

图2　系统DOP与任务作业DMP的关系说明示例

综上,DOP比基于DMP的分析更加可靠,由此得出可调度性结论:若在任意时间范围Δ内,任务系统 τ 的概率性需求边界函数pDBF可以在其DOP不大于某一阈值 H_T 的条件下满足Δ内的资源供给,则认为系统是可调度的。

4　混合关键级任务的概率性需求边界函数

MCS的任务调度常采用EDF-VD[30]的调度策略。利用缩短后的虚拟截止期 $DLO\ i$ 来弱化模式切换给系统DBF带来的影响。对于 $\tau_i \in \tau^H$,有 $DLO\ i \leqslant D_i, DLO\ i \geqslant C_i$。首先提出MCS的可调度性命题:

命题1:对于一个使用pWCET参数的混合关键级系统任务集 τ,在EDF算法调度下,若系统在高关键级模式以及低关键级模式下,其资源需求过载概率都不大于给定的可调度性概率阈值 H_T,则认为任务系统是可调度的,即:

$$LO-Condition: DOP_{\tau,\Delta} \leqslant H_T, \forall \Delta \geqslant 0, \quad (16)$$

$$HI-Condition: DOP_{\tau^H,\Delta} \leqslant H_T, \forall \Delta \geqslant 0, \quad (17)$$

证明:若任务被允许以极其微小概率 H_T 错过其截止期,则等同于系统运行平均失效时间必须大于设计者给定的系统运行预期寿命。为保证高关键级任务在所有模式下都有相同程度的可靠性保证,H_T 的所有模式是统一的。基于这样的系统可靠性保证,结合引理1,原命题为真。

4.1　根据任务执行预算重构任务的pWCET

图3为重构任务的pWCET概率分布,在低关键级模式下pDBF中使用 C LO i,在高关键级模式下仍使用完整的 C_i,这体现预算 B_i 对系统的控制作用。对于 $\tau_i \in \tau^L$,一但运行超过 B_i 就会被调度器中止,所以重构将超出 B_i 部分的概率堆叠到 B_i 处;对于 $\tau_i \in \tau^H$,运行超过 B_i 会使系统提升关键级,但这部分需求考虑在高关键级模式分析中,所以重构将这部分概率堆叠到0。任务 B_i 的设置存在天然矛盾性:B_i 设置过小,低关键级任务运行的服务质量将恶化,高关键级任务更易预算超支;B_i 设置过大将损害低关键级模式的可调度性。

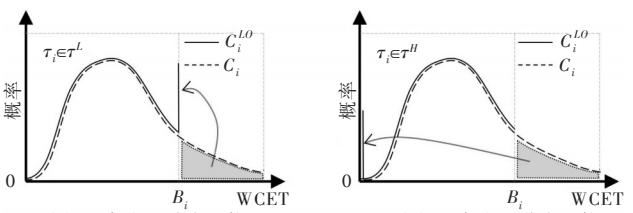

Fig. 3　Reconstructing C LO i based on task pWCET for pDBF computation of task system

图3　根据任务的pWCET重构C LO i以用于任务系统的pDBF计算

4.2　结转作业资源需求分析

任何高关键级任务在系统发生模式切换时,都可能存在结转任务 $J\vdash i$,可将 $J\vdash i$ 视为在模式切换时释放的新作业。

在保证高关键级作业在可调度条件下,必须进行最坏情况分析。假设 $J\vdash i$ 会被尽可能迟地释放,即 $J\vdash i$ 在原来低关键级模式下调度时恰好在截止期前完成,但 $J\vdash i$ 在系统切换为高关键级模式后其执行时间大于 B_i 的概率一定大

于0，即τ_i超过执行预算B_i的概率。

如图4所示，$J\vdash i$可能包含的情况有3部分：①作业在模式切换前可能已经执行；②在剩余调度窗口$l\vdash i$内为原低关键级模式下剩余的执行需求；③由于模式切换而导致突然增加的执行需求（这种情况一定存在）。下面将例1中任务τ_2拓展为例2（见表2），并分析$J\vdash i$的执行需求$C\vdash 2$。

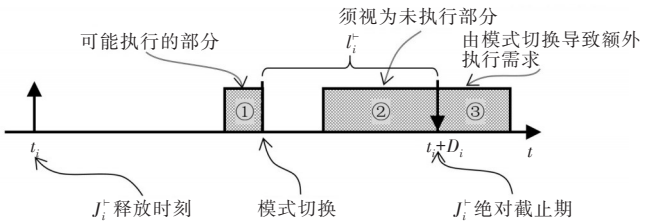

Fig. 4 Worst case scenario of carry-over job

图4 结转作业可能出现的最坏情况

Table 2 Model task τ_2 of example 2-pDBF

表2 例2-pDBF模型任务τ_2

τ	L_i	C_i	B_i	D_i	T_i
τ_2	HI	$C_2=\begin{pmatrix}1 & 3 & 5\\ 0.9 & 0.09 & 0.01\end{pmatrix}$	3	8	8

$J\vdash 2$在未发生模式切换时，其截止期前至多满足B_2个单位的资源需求。$l\vdash 2$满足$0\leqslant l\vdash 2<D_i=8$，以$l\vdash 2=2$为例，对$C_2$各个部分分别分析。对于$c_1=1<l\vdash 2$，从最坏情况考虑，把它归类到如图4所示的第②部分，得到$C,\vdash,head\ 2$；对于$l\vdash 2<c_2=3\leqslant B_2$，由于在剩余执行窗口内至多执行$l\vdash 2$个单位，则至少有1个单位已经被执行了，即包含了图4中的①、②部分，得到$C,\vdash,mid\ 2$；对于$l\vdash 2<B_2<c_3=5$，由于至多保证B_2个单位执行需求被满足，所以有$c_3-B_2=2$个单位无法得到满足，包含在第①、②、③部分中。得到$C,\vdash,tail\ 2$如下：

$$C_2^{\vdash,head}=\begin{pmatrix}1\\0.9\end{pmatrix},C_2^{\vdash,mid}=\begin{pmatrix}2\\0.09\end{pmatrix},C_2^{\vdash,tail}=\begin{pmatrix}2\\0.01\end{pmatrix}, \tag{17}$$

$$C_2^{\vdash}=C_2^{\vdash,head}\oplus C_2^{\vdash,mid}\oplus C_2^{\vdash,tail}=\begin{pmatrix}1 & 2\\0.9 & 0.1\end{pmatrix}, \tag{18}$$

但无论$l\vdash 2$多长，肯定包含第③部分，这导致无法满足H_T的约束，而EDF-VD调度算法可缓解这一问题。

为了给第③部分执行预留空间，采用EDF-VD调度算法为每个高关键级任务设置虚拟截止期$D\mathrm{LO}\ i(C_i\leqslant D\mathrm{LO}\ i\leqslant D_i)$。如图5所示，使用虚拟截止期后$J\vdash i$获得更长调度窗口。

引理2（结转作业的执行需求）：对于高关键级任务$\tau_i\in\tau^{\mathrm{H}}$的作业，其在低关键级模式以及高关键级模式下的EDF调度分别根据$D\mathrm{LO}\ i$与D_i进行。若在低关键级模式下，系统的资源需求可以得到满足，则模式切换后其结转作业$J\vdash i$的剩余调度窗口长度为$l\vdash i\geqslant 0$，则有：

（1）若$l\vdash i<D_i-D\mathrm{LO}\ i$，表示该作业在模式切换前已经完成，不用进行作业结转。

（2）若$l\vdash i\geqslant D_i-D\mathrm{LO}\ i$，表示该作业必然为一个结转作业，即在模式切换后该作业仍有执行需求$C\vdash i$，且$C\vdash i$如式（19）、式（20）所示，其中$l\vdash' i=l\vdash i-(D_i-D\mathrm{LO}\ i)$。

证明：对于$l\vdash i<D_i-D\mathrm{LO}\ i$，表示模式切换发生在虚拟截止期$D\mathrm{LO}\ i$之后，所以作业在模式切换前已完成，不用作业结转；对于$l\vdash i\geqslant D_i-D\mathrm{LO}\ i$，当$0\leqslant l\vdash i'<B_i$时，由于$l\vdash i'$不足以完成$B_i$个执行需求，所以必然有部分需求已执行，$C\vdash i$须扣除这部分，所以$C\vdash i$包含①、②、③部分；对于$B_i\leqslant l\vdash i'<D\mathrm{LO}\ i$，须把整个执行预算内的需求放在$l\vdash i'$内考虑，不包含第①部分，$C\vdash i$包含第②、第③部分。如图5所示。

$$C_i^{\vdash}=C_i^{\vdash,head}\oplus C_i^{\vdash,mid}\oplus C_i^{\vdash,tail}$$

$$\begin{cases}C_i^{\vdash,head}=\begin{pmatrix}c_1 & \cdots & c_m\\ p_1 & \cdots & p_m\end{pmatrix},c_1<\cdots<c_m<l_i^{\vdash}{}'\\ C_i^{\vdash,mid}=\begin{pmatrix}l_i^{\vdash}{}' \\ \sum_{j=m+1}^{n}p_j\end{pmatrix},l_i^{\vdash}{}'\leqslant c_{m+1}<\cdots<c_n<B_i,0\leqslant l_i^{\vdash}{}'<B_i\\ C_i^{\vdash,tail}=\begin{pmatrix}l_i^{\vdash}{}'+c_{n+1}-B_i & \cdots & l_i^{\vdash}{}'+c_k-B_i\\ p_{n+1} & \cdots & p_k\end{pmatrix},B_i\leqslant c_{n+1}<\cdots<c_k\end{cases} \tag{19}$$

$$C_i^{\vdash}=C_i,B_i\leqslant l_i^{\vdash}{}'<D_i^{\mathrm{LO}} \tag{20}$$

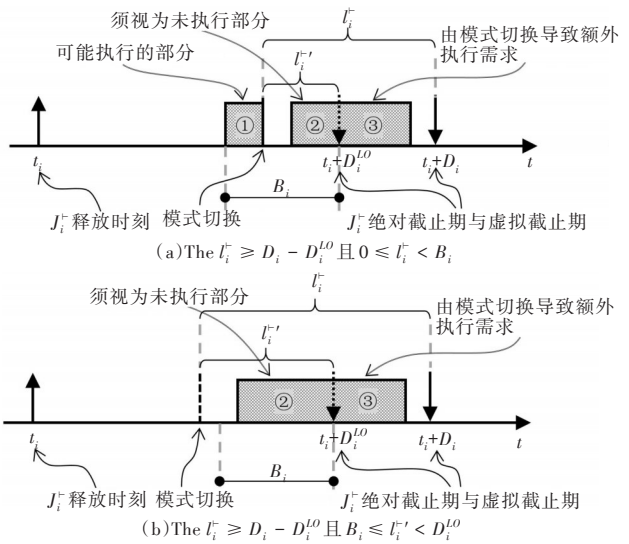

Fig. 5 By using virtual deadline to left more scheduling windows for J⊢2

图5 通过使用虚拟截止期为J⊢2空出更多的调度窗口

4.3 不同模式下系统的pDBF

在低关键级模式下，系统成为标准的零星任务系统，其低关键级模式下任务系统的pDBF为：

$$\mathrm{pdbf}^{LO}(\tau,\Delta)=\bigotimes_{\forall\tau_i\in\tau}\left(\overset{nx}{\otimes}C_i^{LO}\right),\forall\Delta\geqslant 0, \tag{21}$$

其中，$nx=\lfloor(\Delta+T_i-D\mathrm{LO}\ i)/T_i\rfloor$。

$l\vdash i$越长表示越可能存在结转作业。如图6所示，为了更多地考虑任务作业，可将完整的任务作业在Δ内尽可能往后排，以空出尽可能大的$l\vdash i$，长度为$\Delta\bmod T_i$。当$l\vdash i<D_i-D\mathrm{LO}\ i$时，$J\vdash i$不存在；$B_i+(D_i-D\mathrm{LO}\ i)\leqslant l\vdash i<D_i$时，$l\vdash i$可以容纳一个完整作业，$J\vdash i$等同一个正常作业；$D_i-D\mathrm{LO}\ i\leqslant l\vdash i<B_i+(D_i-D\mathrm{LO}\ i)$时，须考虑有一个结转作业情况。综上，高关键级模式下pDBF可表示为：

$$\mathrm{pdbf}^{\mathrm{HI}}(\tau^{\mathrm{HI}},\Delta)=\bigotimes_{\forall\tau_i\in\tau^{\mathrm{HI}}}\mathrm{pdbf}^{\mathrm{HI}}(\tau_i,\Delta), \tag{22}$$

其中

$$\mathrm{pdbf}^{\mathrm{HI}}(\tau_i,\Delta)\overset{\mathrm{def}}{=\!=}\begin{cases}\left(\overset{nx}{\otimes}C_i\right)\otimes C_i^{\vdash} & ,D_i-D_i^{\mathrm{LO}}\leqslant l_i^{\vdash}<B_i+D_i-D_i^{\mathrm{LO}}\\ \overset{nx}{\otimes}C_i & ,\text{其他}\end{cases} \tag{23}$$

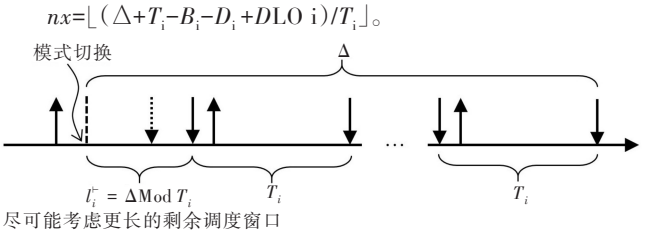

Fig. 6　The worst case distribution of high critical job

图 6　高关键级任务作业的最坏排布情况

5　实验结果与分析

5.1　生成实验任务集

随机实验任务集生成参数设置如下:①任务关键级 L_i:CP 表示生成任务为高关键级任务的概率,默认为 CP = 0.5;任务周期 T_i 设置为 $25\cdot\omega$,汽车或航空器中常见的实时任务周期一般为 20~1 000ms,所以 ω 随机选择[1, 40]范围的整数;②任务平均利用率 $Uavg\tau_i$ 基于 UUnifast 算法[31]生成;③任务 $\mathcal{C}_i$:根据 $\bar{C}_i=T_i\cdot Uavg\tau_i$ 以及随机生成的 WCET 参数 $cmax$ i∈$[1.1\bar{C}_i, 2\bar{C}_i]$,按内推插值法生成序列($c_0, c_1, \cdots, cmax$);$\mathcal{C}_i$ 没有具体分布,唯一的限制是在分布上呈递减趋势,这里按指数型衰减生成;④任务执行预算 B_i:根据任务执行概率阈值 P_T 选择,即满足 $P(\mathcal{C}_i\geqslant B_i)=P_T$,$P_T$ 默认为 10^{-5};⑤任务截止期 D_i、虚拟截止期 DLO i:$D_i=T_i$;DLO i 在$[B_i, T_i]$内随机选取。

5.2　算法复杂度实验

为尽可能彻底测试所提出的可调度性测试算法的时间复杂度,将任务集的任务个数在 2~15 范围内变化,任务 pWCET 长度在 2~15 范围内变化。任务集平均利用率 $Uavg\tau$ 从 0.05~1.05 随机产生;可调度性概率阈值 H_T 在实验开始时从$[10^{-6}, 10^{-5}]$中随机选择,实验结果如图 7 所示。z 轴表示算法测试所用时间,空间中每个实验数据点取 100 个随机任务集平均测试时间;算法运行时间与任务集个数以及 pWCET 参数长度呈指数关系,任务集个数对运行时间影响更突出,这是因为测试算法的外循环次数为 HP+1,内循环次数为任务集中任务个数,基本运算操作为对随机变量的卷积和,所以前者影响更大。综上所述,当测试任务集平均利用率大于 1 时,算法运行在线性时间复杂度上,其他情况下则运行在伪多项式时间复杂度上。所以应尽量减小超周期,比如通过取幂值方式。

对于临界时刻释放的周期或零星任务系统,尽管采用了 pWCET 参数,但在一个超周期内分析同样有效。测试范围至少应为一个超周期,因为在 DBF 模型分析下,l_{max} 指 DOP>0 的最早时刻,而在 pDBF 分析下,指 DOP>H_T 的最早时刻;另外算法是基于系统不同模式下的平均利用率判定,这利用了随机过程中的更新报酬定理[32]。

算法 1:可调度性测试算法

输入:MCS 零星任务集 $\tau=\{\tau_1, \tau_2, \cdots, \tau_n\}$,可调度性概率阈值 H_T,任务系统超周期 $HP=lcm\{T_1, T_2, \cdots, T_n\}$;

输出:可调度性分析结果("schedulable or not schedulable")。

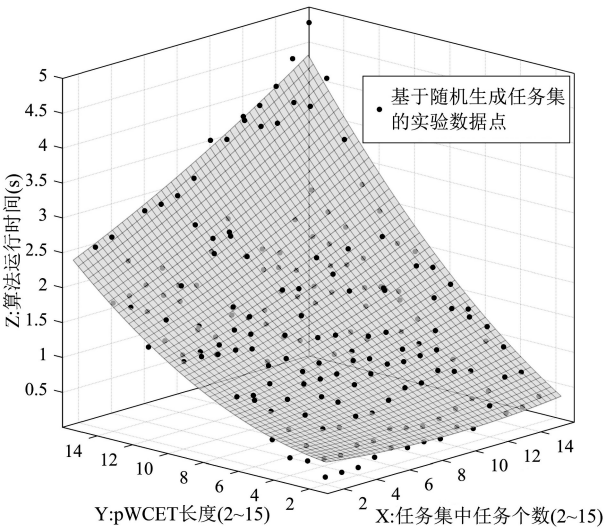

Fig. 7　Executing time of schedulability testing algorithm

图 7　可调度性测试算法运行时间

ULO avg←$CalcuTaskUavg(\tau^L)$; /*LO-模式平均利用率*/

UHI avg←$CalcuTaskUavg(\tau^H)$; /*HI-模式平均利用率*/

if ULO avg> 1||UHI avg> 1 **then return**("not schedulable");

end if

for t := 0 **To** HP **do**

　for i := 1 **To** n **do**

Lr ←t ModT_i;/*最大剩余调度窗口长度*/

if Lr ==DLO i **then**

pdbfLO τ←pdbfLO $\tau\otimes\mathcal{C}$LO i; /*LO-模式系统 10.pDBF*/

end if

if L_i ==HI&&D_i-DLO i≤ Lr ≤B_i+D_i-DLO i **then**

if Lr ==B_i+D_i-DLO i **then**

pdbfHI_full τ←pdbfHI_full $\tau\otimes\mathcal{C}$ i;

else

pdbfHI_⊢ τ←pdbfHI_⊢ $\tau\otimes\mathcal{C}\vdash$ i; /*结转作业 pDBF*/

end if

end if

end for

pdbfHI τ←pdbfHI_full $\tau\otimes$ pdbfHI_⊢ τ; /*HI-模式系统 pDBF*/

$DOP_{\tau,t}$←$FindMaxDop$(pdbfLO τ, pdbfHI τ);

if $DOP_{\tau,t}>H_T$ **then return**("not schedulable");

end if

end for

return("schedulable");

5.3　仿真实验

所有任务的截止期等于其周期,下面分析 pWCET 参数长度、系统可调度性概率阈值 H_T 以及系统高关键级任务概率 CP 等系统参数对可调度性性能的影响。任务集包含 10 个任务,系统平均利用率以 0.05 的步长从 0.05 变化到 1.0,每个实验数据点测试 100 个随机任务集。

(1)改变任务 pWCET 参数长度。使用重抽样技术[33],在不降低参数可靠性前提下缩减任务的 pWCET 参数长度,当长度抽样为 1 时使用传统的 DBF 方法分析。

如图8所示，$Uavg\tau$较小时(0.05~0.55)，pWCET长度对于可调度性没有显著改善。而当利用率增大时，确定性分析(虚线)的可调度性状况迅速恶化，但在不同pWCET长度下的可调度性仍有较大提升。但是当pWCET长度大于8后，这种提升就变得十分有限。

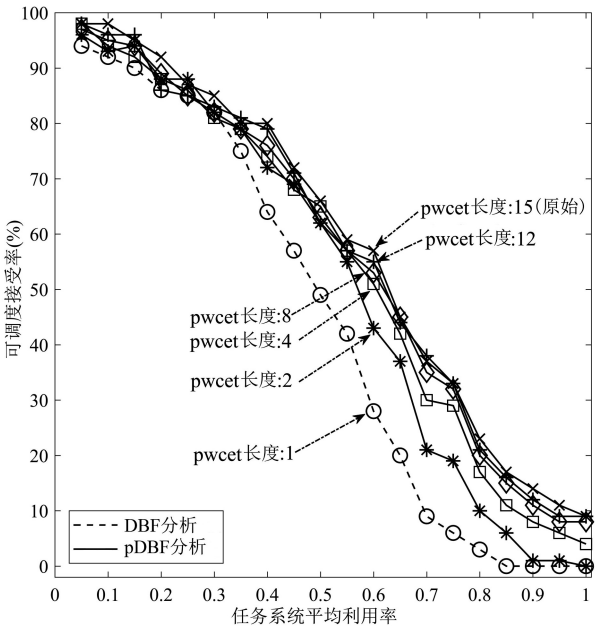

Fig. 8 Changing the pWCET parameter length of the task (by re-sampling)

图8 改变任务的pWCET参数长度(通过重抽样方法)

(2)改变系统可调度性概率阈值H_T。施加越严格的可调度性概率阈值，系统越可靠。

图9中，H_T越宽松，系统的可调度性越好，但是在系统平均利用率特别小(0.05~0.25)或特别大(0.95~1.00)时，这种表现并不突出。

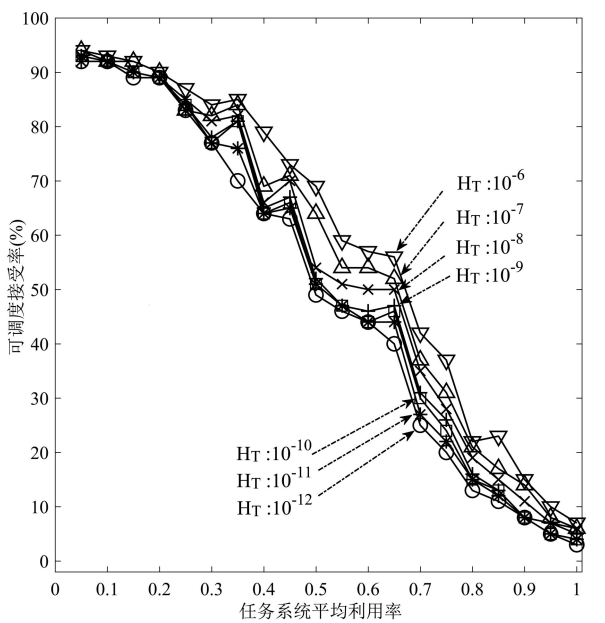

Fig. 9 Changing the probability threshold of system schedulability H_T

图9 改变系统可调度性概率阈值H_T

(3)改变系统高关键级任务概率CP。提高系统高关键级任务的比例会增大系统高关键级模式下的pDBF。

从图10可以发现，在改变$Uavg\tau$的同时，不同的CP可调度性性能之间相对地位几乎没有改变，说明CP对于可调度性的影响比较独立。

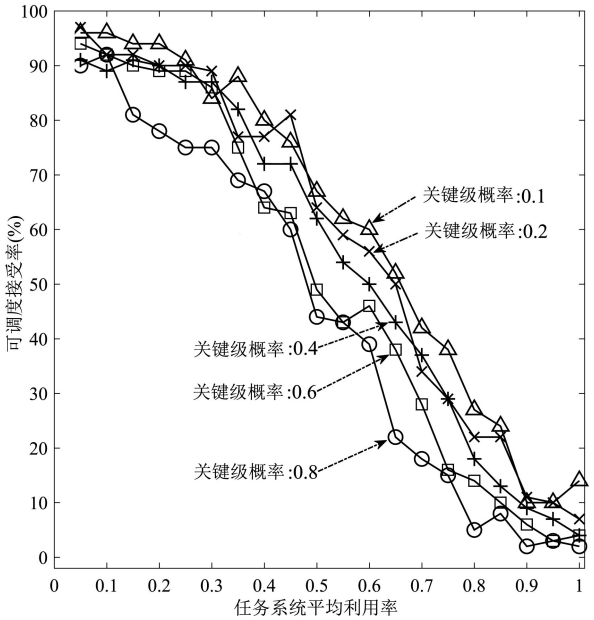

Fig. 10 Changing the system high critical level task probability CP

图10 改变系统高关键级任务概率CP

6 结语

传统MCS默认依据任务的WCETs来安排执行预算，这种方法会产生资源过度预置问题，而一般MCS低关键级任务执行的可靠性要求不像高关键级任务要求那么高。本文首先分析并提出了概率性需求边界函数(pDBF)模型，分析了MCS不同模式下的pDBF，尤其是系统模式切换时结转作业的执行需求。实验表明，通过对C_i重采样或者选择合适的H_T，可调度接受率可以提高32%，同时降低了pDBF模型下分析算法的复杂度。未来可采用实时接口分析(real-time interface analysis)方法，将本文方法拓展至固定任务优先级调度系统中。

参考文献：

[1] BURNS A, DAVIS R I. A survey of research into mixed criticality systems[J]. ACM Transactions on Embedded Computing Systems, 2017, 50(6):82-95.

[2] VESTAL S. Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance[C]. Tucson: 28th IEEE International Real-Time Systems Symposium, 2007.

[3] ZENG L N, XU C, LI R F, et al. Scheduling algorithm for mixed-criticality jobs based on dynamical demand boundary[J]. Journal of Software, 2020,31(11):3657-3670.
曾理宁，徐成，李仁发，等. 一种基于动态需求边界的混合关键级作业调度算法[J]. 软件学报，2020,31(11):3657-3670.

[4] JALIL B, SARAVANAN R, ARVIND E, et al. Combining task-level

and system-level scheduling modes for mixed criticality systems[C]// IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications,2019:136-145.

[5] HU B, THIELE L, HUANG P C, et al. FFOB: efficient online mode-switch procrastination in mixed-criticality systems [J]. Real-Time Systems, 2019, 55(3):471-513.

[6] MAXIM D, DAVIS I R, CUCU G L, et al. Probabilistic analysis for mixed criticality systems using fixed priority preemptive scheduling [C]//25th International Conference on RTNS, 2017: 237-246.

[7] CUCU-GROSJEAN L, SANTINELLI L, HOUSTON M, et al. Measurement-based probabilistic timing analysis for multi-path programs [C]//24th Euromicro Conference on Real-Time systems, 2012: 91-101.

[8] ALTMEYER S, CUCU GROSJEAN L, DAVIS I R. Static probabilistic timing analysis for real-time systems using random replacement caches[J]. Real-Time Systems, 2015, 51(12):77-123.

[9] STEFAN DRASKOVIC, REHAN, AHMED, et al. Schedulability of probabilistic mixed-criticality systems[J]. Real-Time Systems,2021, 57(7):1-46.

[10] MAXIM D, CUCU-GROSJEAN L. Response time analysis for fixed-priority tasks with multiple probabilistic parameters[C]//Proceeding of IEEE 34th RTSS, 2013: 224-235.

[11] TIA S T, DENG Z, SHANKAR M, et al. Probabilistic performance guarantee for real-time tasks with varying computation times[C]// Proceeding of the Real-Time Technology and Applications Symposium, 1995: 164-173.

[12] EDGAR S, BURNS A. Statistical analysis of WCET for scheduling [C]//Proceeding of IEEE 22nd RTSS, 2001: 215-224.

[13] DÍAZ L J, GARCÍA F D, KIM K, et al. Stochastic analysis of periodic real-time systems[C]//Proceeding of IEEE 23rd RTSS, 2002: 289-300.

[14] DÍAZ L J, LÓPEZ M J, GARCÍA M, et al. Pessimism in the stochastic analysis of real-time systems: concept and applications[C]//Proceeding of IEEE 25th RTSS, 2004:197-207.

[15] LÓPEZ M J, DÍAZ L J, ENTRIALGO J, et al. Stochastic analysis of real-time systems under preemptive priority-driven scheduling[J]. Real-Time Systems, 2008, 40(5): 180-207.

[16] SANTINELLI L, CUCU-GROSJEAN L. A probabilistic calculus for probabilistic real-time systems[J]. ACM Transactions on Embedded Computing Systems, 2015, 14(3):52-67.

[17] PALOPOLI L, FONTANELLI D, MANICA N, et al. An analytical bound for probabilistic deadlines [C]//Proceeding of IEEE 24th ECRTS, 2012: 179-188.

[18] KACZYŃSKI A G, BELLO L L, NOLTE T. Deriving exact stochastic response times of periodic tasks in hybrid priority-driven soft real-time systems[C]// Proceeding of IEEE 12th ETFA, 2007: 101-110.

[19] DRASKOVIC S, HUANG P C, THIELE L. On the safety of mixed-criticality scheduling[C]//Portugal:Proceeding of 2016 WMC, 2016.

[20] ABDEDDAÏM Y, MAXIM D. Probabilistic schedulability analysis for fixed-priority mixed criticality real-time systems[C]//Proceeding of AMC 20th DATE, 2017: 596-601.

[21] GUO Z S, SANTINELLI L, YANG K C. EDF schedulability analysis on mixed-criticality systems with permitted failure probability[C]// Proceeding of IEEE 21st RTCSA, 2015: 187-196.

[22] REGHENZANI F, MASSARI G, FORNACIARI W. A probabilistic approach to energy-constrained mixed-criticality systems[C]//2019 IEEE/ACM Intl Symposium on Low Power Electronics and Design, 2019: 1-6.

[23] BHUIYAN A, REGHENZANI F, FORNACIARI W, et al. Optimizing energy in non-preemptive mixed-criticality scheduling by exploiting probabilistic information[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, 39(11): 3906-3917.

[24] MEDINA R, BORDE E, PAUTET L. Availability enhancement and analysis for mixed criticality systems on multi-core[C]//Design, Automation and Test in Europe Conference and Exhibition, 2018: 1271-1276.

[25] ZENG L N, XU C, LI R F. Partition and scheduling of the mixed-criticality tasks based on probability[J]. IEEE ACCESS, 2019, 128(7): 87837-87848.

[26] HUANG L D, LI R F. The scheduling analysis of real-time tasks after event-triggered criticality level transition[J]. Journal of Computer Research and Development, 2017, 54(1):184-191.
黄丽达, 李仁发. 事件触发关键级提升的实时任务可调度性分析[J]. 计算机研究与发展, 2017, 54(1), 184-191.

[27] BARUAH K S, MOK K A, ROSIER E L. Preemptively scheduling hard-real-time sporadic tasks on one Processor[C]//Proceeding of IEEE 11th RTSS, 1990: 182-190.

[28] EKBERG P, WANG Y. Bounding and shaping the demand of generalized mixed-criticality sporadic task systems[J]. Real-Time Systems, 2013, 50(1): 48-86.

[29] ZHANG F D. Distributed real-time system[M]. Beijing: Science Press, 2014.
张凤登. 分布式实时系统[M]. 北京:科学出版社,2014.

[30] BARUAH S,BONIFACI V,D'ANGELO G,et al. The preemptive uniprocessor scheduling of mixed-criticality implicit-deadline sporadic task systems[C]//Proc of IEEE 24th ECRTS,2012:145-154.

[31] BINI E, BUTTAZZO G C. Measuring the performance of schedulability tests[J]. Real-Time Systems, 2005, 30(5):129-154.

[32] ROSS M S. Introduction to probability models[M]. 11th ed. Singapore: Elsevier, 2014:433-441.

[33] MAXIM D, HOUSTON M, SANTINELLI L, et al. Re-sampling for statistical timing analysis of real-time systems[C]//Proceeding of ACM 20th RTNS, 2012: 111-120.

(责任编辑:杜能钢)