

信息物理融合系统实时数据服务

邵亚丽¹, 陈海华², 张立臣³, 陈晓纯¹

(1. 广东理工学院 信息工程系, 广东 肇庆 526000;

2. 广东财经大学华商学院 数据科学学院, 广东 广州 511300;

3. 广东工业大学 计算机学院, 广东 广州 510000)

摘要:信息物理融合系统(CPS)是一个智能化决策和控制系统,在大数据时代背景下能提供实时感知、信息服务及动态控制功能,这意味着它必须是安全的、可靠的、实时的、高效的。由此实时性显得尤为重要,它是保证CPS安全、可靠、高效的基础。对CPS的实时性可以从实时数据服务、实时控制、数据更新事务和控制事务的协同调度角度等进行研究。该文主要从实时数据服务对其展开研究,通过比较周期性和非周期性数据更新事务的实时调度算法的可调度性和CPU利用率,对算法进行改进。仿真结果表明改进的DESH-SC和DESH-SA能明显减少调用开销,DF-LALF的调度成功率较高,而调度开销较小。

关键词:信息物理融合系统;实时性;数据服务;实时控制;调度算法

中图分类号:TP302.1

文献标识码:A

文章编号:1673-629X(2021)05-0113-06

doi:10.3969/j.issn.1673-629X.2021.05.020

Real-time Data Service of Cyber-physical System

SHAO Ya-li¹, CHEN Hai-hua², ZHANG Li-chen³, CHEN Xiao-chun¹

(1. Department of Information Engineering, Guangdong Polytechnic College, Zhaoqing 526000, China;

2. School of Data Science, Huashang College of Guangdong University of Finance and Economics,
Guangzhou 511300, China;

3. School of Computer, Guangdong University of Technology, Guangzhou 510000, China)

Abstract: Cyber-physical system (CPS), as an intelligent decision-making and control system, can provide real-time perception, information services and dynamic control functions in the context of the big data era, which means it must be safe, reliable, real-time and efficient. Therefore, the real-time is particularly important, which is the basis for ensuring the safety, reliability and efficiency of CPS. The real-time of CPS can be studied from the perspective of real-time data services, real-time control, collaborative scheduling of data updating transactions and controlling transactions. We mainly study the real-time data service and improve the algorithm by comparing the schedule ability and CPU utilization of the real-time scheduling algorithm of periodic and aperiodic data update transactions. The simulation shows that the improved DESH-SC and DESH-SA can significantly reduce the call overhead, while DF-LALF has a higher scheduling success rate and less scheduling overhead.

Key words: cyber-physical system; real-time; data service; real-time control; scheduling algorithm

0 引言

CPS是一个多维复杂系统,通过计算、通信和控制等相关技术的相互协作,将信息世界和物理世界进行深度融合,实现实时感知、信息服务以及动态控制的功能。作为下一代智能化、网络化、机电一体化控制系统的核心技术,CPS具有重大研究价值。大数据时代,CPS面临着重大挑战,它必须能够实时感知、实时通

信、实时控制,因此实时性是CPS的重要特征。该文从CPS的数据管理层面研究了它的实时性方法,并对这些方法进行研究、改进、仿真证明。

CPS是一个具有清晰架构和使用流程的技术体系,而非一项简单的技术,对工业数据具有实时分析、实时处理的能力。它包含数据的整个处理流程(采集、聚合、解析、排序、分析、预测、决策和分发等),并

收稿日期:2020-06-19

修回日期:2020-10-21

基金项目:国家自然科学基金(61572142,61370082);广东省自然科学基金2016年广东省特色创新(自然科学)项目(2016KTSCX165);大学生创业(项目)(CXCY2019020);2017年广东省教育厅青年创新人才项目(2017KQNCX235)

作者简介:邵亚丽(1991-),女,助教,研究方向为自动化技术、机器学习。

在分析过程中充分考虑各种特征和要求,例如逻辑、流程关系、目标等。在类似自动汽车、智能电网、建筑控制和医疗健康等 CPS 基础设施中,监视和控制至关重要。传统的控制系统的设计关注点如不确定性、物理干扰、鲁棒控制和自适应控制不同,CPS 反映了控制系统与计算技术的集成水平。受工业控制系统的启发,如图 1 所示,该文对 CPS 中的控制服务进行分层设计,展示了控制服务的不同设计层面,包括架构层、特定控制设计关注点和云那部分架构。在该架构中,主要关心延迟和实时控制操作。CPS 的实时行为是多种不同操作的结合,如传感、处理、通信和执行。这迫切需要一个将这些领域的现有研究成果和新的解决方案包含进来的统一实时操作理论,以保证 CPS 系统的实时性操作。为保证实时数据库中的数据新鲜度和系统控制质量,该文从物理控制层出发,设计有效的算法来执行和调度传感器的更新和控制任务。

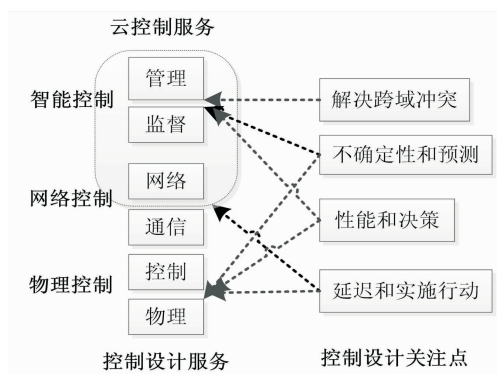


图 1 CPS 控制系统的通用框架及设计关注点

为了保持时间有效性,需要生成传感器更新事务,以周期性地更新系统环境中实时数据对象的值^[1]。传感器更新事务具有无限数量的周期性作业,这些作业具有固定长度的周期和相对截止时间。目前已有许多学者为维持实时数据的有效性和新鲜度进行了广泛的研究工作^[2-7]。其中一些使用周期性更新事务,而其他的一些处理非周期性更新事务。第二类,如文献[6-7],主要设计用于软实时系统,解决如何在运行时调度更新事务,以最大化实时数据对象的新鲜度,同时最小化它们对应用程序的实时事务处理的影响。尽管这些方法已被证明能有效地实现更好的平均性能,但是它们不能为实时事务的执行提供数据有效性保证。相反,使用周期性更新事务的那些方法的性能目标是对数据有效性提供保证。该文在已有的周期性调度算法的基础上提出的改进算法在有效保证数据有效性的同时,最小化更新工作量。

1 CPS 实时数据服务

通常 CPS 是深嵌于物理世界,用以将搜集到的动

态物理测量结果(实时数据)存储到实时数据库(RTDBS)。接着其子系统就能够对这些实时数据进行查询,然后执行控制操作。作为一个实时控制系统,保证 CPS 子系统查询数据的新鲜度是保证控制任务的正确及时执行的前提,因此显得尤为重要。与传统关系数据库中的数据不同,实时数据有时间语义,即采样值只在特定的时间段内有效^[8-11]。大数据时代,CPS 呈现出多模行为并且系统中的更新和控制工作负载动态变化,为了保证系统数据的新鲜度和控制质量,同时减少系统和网络开销,CPS 如何高效处理和调度传感器的更新和控制任务是个亟需解决的问题,因此有必要设计解决这类问题的算法。

定义 1(实时数据):一个实时数据对象 X_i 可以用一个三元组表示, $X_i = (\text{Val}, V_i, R_i)$, 其中 Val 、 V_i 、 R_i 分别指数据对象的当前值、有效期间隔、采样时间。

定义 2(数据的时间有效性):在任意时刻 t , 采样的实时数据对象 X_i 的数据值在 $t + V_i$ 内是暂时有效的,之后它是无效的或过时的。则称该实时数据是时间有效的。

定义 3(抖动时间):即传感器产生的数据更新事务 τ_i 到达实时数据库系统的时间和采样时间的差值^[12]。

实时数据对象的有效性间隔的实际值取决于应用以及相应实体的动态属性^[8]。许多 RTDBS 的重要设计目标之一是保证实时数据的新鲜度,访问无效数据值可能会严重影响系统的实时功能^[3-5,13]。

2 实时系统调度

为了方便理解,本节给出常用符号及其定义。 $T = \{\tau_i\}_{i=1}^m$ 是一组周期性更新事务 $\{\tau_1, \tau_2, \dots, \tau_m\}$, $X = \{X_i\}_{i=1}^m$ 是一组实时数据对象。 V_i 是 X_i ($1 \leq i \leq m$) 的有效间隔长度。事务 τ_i ($1 \leq i \leq m$) 更新相应的数据对象 X_i 。 $d_{i,j}$ 、 $f_{i,j}$ 和 $r_{i,j}$ 分别表示 τ_i 的第 j 个作业 $J_{i,j}$ 的绝对截止时间、完成时间和采样时间。在此假设(1)系统是同步的,即所有更新事务的第一个作业在同一时刻释放;(2)作业的采样时间等于释放时间;(3)更新事务的截止时间是绝对截止时间;(4)事务必须每个周期执行一次;(5)调度算法是抢占式的。以下是该文包含的几种调度算法:

Half-Half (HH) 的基本思想是:为保持实时数据对象的有效性,在保证更新事务的周期和相对截止时间之和不大于是该事务的有效时间间隔长度的前提下,规定的周期等于其相对截止时间,且都等于有效间隔的一半^[9,14-15],即 $P_i = D_i = V_i / 2$ 。

More-Less (ML) 的基本思想是:采用周期性事务模型,在保证更新周期和相对截止时间的和不大于是更

新数据对象的有效期间隔的前提下,设置更新事务的周期大于相应更新数据的有效期间隔的一半,即 $P_i + D_i \leq V_i$, $D_i \leq V_i/2 \leq P_i$ 。

ML 的提出是为了最小化更新工作负载,它使用单调截止时间(deadline monotonic, DM)^[3]来调度周期性更新事务。

对于可调度的 ML,事务 τ_i ($1 \leq i \leq m$) 应当满足三个约束条件:

(1)有效约束:事务 τ_i 的周期和相对截止时间的和总是小于或等于 V_i ,即 $D_i + P_i \leq V_i$,如图 2 所示。

(2)截止时间约束:更新事务的周期需大于其更新的实时数据的有效期间隔长度 $1/2$,而它相应的截止时间小于或等于该实时数据对象的有效期长度的 $1/2$ 。对于即将被调度的事务 τ_i ,其截止时间 D_i 必须大于或等于 τ_i 的最坏情况执行时间 C_i ,即 $P_i \geq V_i/2$, $D_i \leq V_i/2$, $I_i + C_i \leq D_i \leq P_i$ 。

(3)可调度性约束:对于给定的一组更新事务 $T = \{\tau_i\}_{i=1}^m$,使用截止时间单调调度算法来调度事务。如果对于 $i < j$, τ_i 的优先级高于 τ_j 的优先级,则必须满足 $\sum_{j=1}^i (\lceil D_i P_j \rceil * C_j) \leq D_i$ ($1 \leq i \leq m$),其中 $\lceil D_i P_j \rceil$ 表示比 τ_i 优先级高的事务 τ_j 所抢占的次数,因此该式指的是事务 τ_i 的第一个作业的响应时间,也就是 τ_i 的最大响应时间。

Deferrable Scheduling algorithm for Fixed Priority transactions (DS-FP) 的基本思想是:以非周期性硬实时事务模型为基础,在确保实时数据对象更新时间有效性的同时,尽可能推迟当前更新事务作业 $J_{i,j}$ 的后继作业的释放时间 $r_{i,j+1}$,从而增加同一事务两个连续作业的时间间隔,以降低 CPU 的工作负载^[6,15-16]。其动态地为更新事务的作业分配周期和相对截止时间。

类似的,DS-FP 算法中更新事务 τ_i ($1 \leq i \leq m$) 也需满足三个约束条件:

(1)有效性约束: $J_{i,j}$ 的释放时间和 $J_{i,j+1}$ 的完成时间之间的最大间隔不能超过 τ_i 的有效期间隔 V_i ,即:

$$d_{i,j+1} - r_{i,j} \leq V_i$$

(2)截止时间约束:后继作业 $J_{i,j+1}$ 的相对截止时间不大于其执行时间,且不大于其周期,即:

$$C_i \leq d_{i,j+1} - r'_{i,j+1} \leq r'_{i,j+1} - r_{i,j}$$

(3)可调度性约束:更新事务集 $T = \{\tau_i\}_{i=1}^m$,假设已知执行时间 C_i 和有效期 V_i ,则需要满足:

$$\sum_{j=1}^i (\lceil D_i P_j \rceil * C_j) \leq D_i$$
 ($1 \leq i \leq m$)

和 ML 一样,DS-FP 也是基于最短有效期优先(Shortest Validity First, SVF)为事务分配优先级,即按实时数据对象的有效期的长度来分配优先级,有效期

越短优先级越高,这有利于松弛度 $(V_i - C_i)$ 较小的事务。

虽然 DS-FP 是一种能有效减少处理器利用率同时保证有效性约束的算法,但是受到在线调度开销的影响,线调度开销在时间复杂性方面远高于 ML。此外,DS-FP 的调度开销取决于事务的数量和更新数据的有效性时长。因此,在事务集变大时,DS-FP 的代价可能变得非常大。和以任务周期的最小公倍数为其超级周期的周期性调度相反,DS-FP 算法通常没有这样的自然超级周期。因此引入两个具有超级周期的可延迟调度算法(deferrable scheduling with hyperperiod, DESH):DESH-SC()和 DESH-SA,用于从 DS-FP 算法中离线构造周期调度,可减少在线调度的时间复杂性。这两个 DESH 算法具有以下两个特征:

特征 1:调度满足有效性约束。

特征 2:在线调度的时间复杂度为 $O(1)$ 。

DESH-SC 的基本思想是:搜索可以无限重复而不违反 DS-FP 调度的超级周期。如果找不到超级周期,则 DESH-SC 返回。DESH-SC 算法由两部分组成:一个用于离线找到超级周期的算法和一个在线调度事务的算法。一旦发现超级周期,后者就变得不重要了,因为它只需要重复超级周期的调度。

DESH-SA 的基本思想是:为一组满足有效性约束的事务 T 离线构造一个超级周期调度 S_H ,假设 S_H 调度的第一个超级周期的长度为 $\|S_H\|$ 。如果可以通过调整时间段 $[0, \|S_H\|]$ 内的 DS-FP 调度来构建 S_H 的第一个超级周期,则可以通过每个 $\|S_H\|$ 时间单元无限重复 S_H 的第一个超级周期来构建完整的 S_H 调度。因此,和 DESH-SC 类似,DESH-SA 算法由两部分组成:用于离线构建超级周期的算法和一个用来在线调度事务的算法。

由于 DS-FP 是一个固定优先级调度算法,限制了需要使用动态优先级调度的系统对它的应用。为了克服该缺陷,提出一种称为具有最小实际松弛优先的可延迟调度的动态调度算法(DS-LALF),以保证实时数据对象的有效性。作业的实际松弛是在作业错过它的截止时间之前通过考虑要执行较高优先级作业所需要的空闲时间来衡量该作业允许的空余时间。

DS-EDF 是文献中最先研究动态优先级系统中的可延迟调度。在 DS-EDF 中,每个更新事务的所有第一个作业的释放时间被初始化为零,且它们的截止时间被分配为相应的有效性长度。将被调度的更新作业按照它们的截止时间的升序顺序放入队列 Q_{EDF} 中。然后,总是最先调度 Q_{EDF} 中具有最早截止时间的作业。对于每个更新事务的第一个作业,它们从指定的释放时间 0 开始调度,而所有其他作业从它们的截止

时间向后调度,并通过考虑从调度中的较高优先级作业的总抢占来推出它们的释放时间。一旦作业 $J_{i,j}$ 完成,其下一个作业 $J_{i,j+1}$ 的截止时间 $d_{i,j+1}$ 被设为 $r_{i,j+1} + V_i$,并将 $J_{i,j+1}$ 放入 Q_{EDF} 队列中。当作业是不可调度时,该算法失败,即计算到它的释放时间小于它前面作业的截止时间。

虽然 DS-EDF 使用比 DS-FP 更灵活的优先级分配,但它的可调度性并不比 DS-FP 好。与 DS-EDF 类似,DS-LALF 是 DS-FP 的扩展。两个连续更新作业之间的间隔是根据在运行时中的较高优先级作业的总抢占来确定的,而不是使用最坏情况的响应时间。然而,与根据其最后截止时间为每个更新作业 $J_{i,k}$ 分配优先级的 DS-EDF 不同,DS-LALF 使用实际松弛,即在它前面的更新工作 $J_{i,k-1}$ 完成后,根据时间段 $[d_{i,k-1}, d_{i,k}]$ 中的可用空闲时隙数来决定作业的优先级。实际松弛在更新作业中考虑了上述可调度性约束,它更好地指示了作业的紧急程度,并且更适应于调度,特别是当作业的负载很重时。

3 仿真结果

实验都是利用 MATLAB 下的 simulink 工具实现。

3.1 ML 和 DS-FP 的性能比较

第一组实验主要用于比较算法 ML、DS-FP、DESH-SC 和 DESH-SA 在处理一组相同更新事务的工作负载。第二组实验,主要研究 DS-FP 和 ML 的调度成功率。

CPU 工作负载的比较。

表 1 中给出了本实验使用的参数及它们的值。实时数据对象的数量从 50 变化到 300,且假设每个实时数据对象的有效性间隔长度范围为 4 000 ms 到 8 000 ms。对于更新事务,假设每个事务更新一个实时数据对象,且每个事务的 CPU 时间从 5 ms 到 15 ms 均匀变化。根据这些参数随机生成更新事务。

表 1 实验参数设置

参数	数值
更新 X_i 的 CPU 时间 C_i	[5,15]
实时数据对象数 N_T	[50,300]
数据对象 X_i 的有效性长度 V_i /ms	[4 000,8 000]

ML 和 DS-FP 产生的更新事务的 CPU 工作负载的定量比较结果如图 2 所示。

由图 2 可知,ML 中的 CPU 工作负载始终低于 DS-FP 的 CPU 工作负载。且随着更新事务量的增加,差异越大,当事务量为 300 时,差异达 16.8%,这是因为 DS-FP 以较低速率自适应地采样实时数据对象。注意 DESH-SC 的曲线仅在更新事务量为 10 到 35 的范

围内绘制。这是因为在实验中,DESH-SC 只能在更新事务数量不超过 35 时找到其超级周期。由图还能发现,DESH-SC 和 DESH-SA 的 CPU 工作负载几乎和 DS-FP 的一样,但是 DESH-SA 的工作负载始终低于 ML。这是因为 DESH-SA 中截止时间调整对增加系统总体工作负载几乎无影响。即使系统中的事务数量高达 300,DESH-SA 只会比 DS-FP 多不到 1% 的 CPU 工作负载。这是因为在实验中,DS-FP 忽略了在线调度开销。DS-FP 的实际 CPU 工作负载应高于图中所示的值。

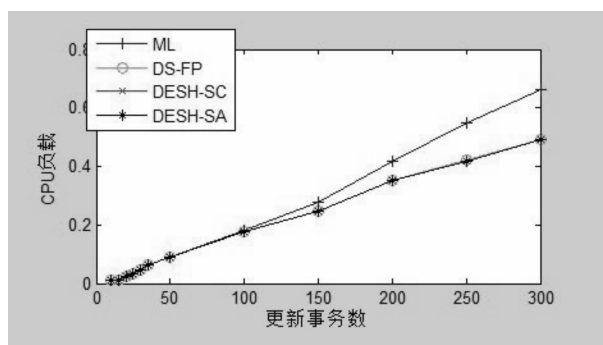


图 2 CPU 负载比较

实验表明 DS-FP 下的 CPU 负载少于 ML 中的 CPU 负载。

3.2 可调度性比较

表 2 给出了本实验使用的参数及它们的值。同样地更新事务根据表中的参数随机生成。 γ 表示的一组事务 T 的密度因子定义为 $\sum_{i=1}^m C_i/V_i$ 。

表 2 实验参数设置

参数	数值
更新 X_i 的 CPU 时间 C_i	[1,10]
实时数据对象数 N_T	[1,15]
数据对象 X_i 的有效性长度 V_i /ms	[20,200]

该组实验,比较 DS-FP 和 ML 在各种 CPU 利用率下的调度成功率。在实验中,系统中有 10 个更新事务,且密度因子在 $[0.5, 0.72]$ 间变化,实验过程中通过固定 C_i 和减少 V_i 来增加密度因子。对每个点运行 200 次,记录它们的平均值。图 3(a)为 DS-FP 和 ML 的调度成功率结果图。可以发现 DS-FP 在可调度性方面始终优于 ML。且可由 ML 调度的所有任务集也均可由 DS-FP 调度。当密度因子为 0.58 时,ML 的成功率下降到 0.85 以下。而在 DS-FP 中只有当密度因子大于 0.63 时,才会发生。此外,当密度因子为 0.64 时,大多数事务集不能由 ML 调度,而 DS-FP 的成功率仍然在 0.55 左右。图 3(b)描述了在 ML 和 DS-FP 下可调度的任务集对应的 CPU 利用率,表明 ML 始终高于 DS-FP 的 CPU 利用率,且当密度因子增加时它

们的差异增加。当密度因子为 0.67 时,差异达到 18.6%,此时 ML 的调度成功率接近 0。实验表明 DS-FP 的性能比 ML 好。

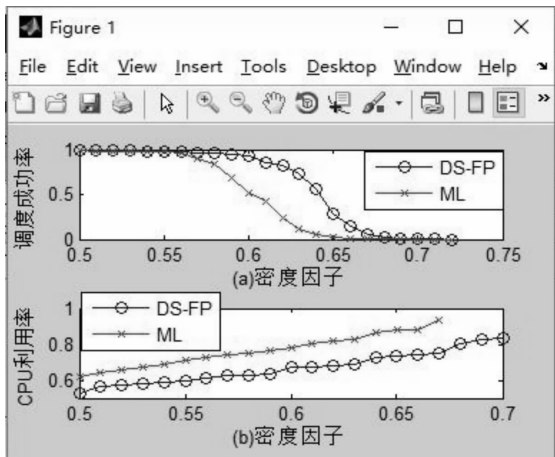


图 3 ML 和 DS-FP 的成功率及 CPU 负载对比

3.3 DS-LALF 的性能评估

以下实验评估 DS-LALF 算法中更新事务占用的 CPU 利用率及调度的成功率。实验的参数设置同表 2,结果如图 4 所示。

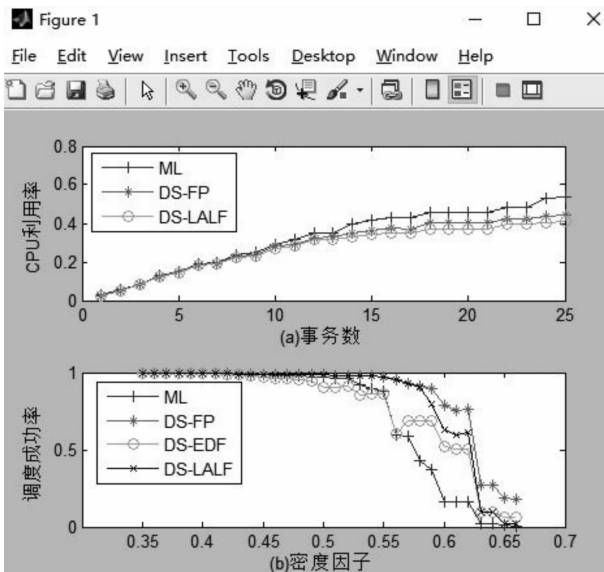


图 4 调度成功率和 CPU 利用率的比较

由图 4(a)可知,DS-FP 和 DS-LALF 的 CPU 利用率始终低于 ML,特别是在更新事务的数量较大时。当系统中有 24 个更新事务时,DS-FP 在 ML 上的改善为 9.5%,DS-LALF 在 ML 上的改善为 11.2%。且可以发现 DS-LALF 的 CPU 利用率始终略低于 DS-FP 的,并明显低于 ML,特别是当更新事务数更大时。这表明在 CPU 利用率方面,DS-LALF 的性能优于 DS-FP 以及 ML,因为它根据可用时隙数根据它们的紧急程度自适应地调度作业。

在图 4(b)中通过改变密度因子来比较在不同更新工作负载下这几个算法间的可调度性。该图描述了

由 5 个更新事务组成的系统的密度因子从 0.35 变化到 0.65 时,它们的调度的成功率,其中执行时间总是设置为最坏情况值。对每个点运行 200 次,图中结果是它们的平均值。

由图 4(b)可得,DS-LALF 的调度成功率始终高于 ML 和 DS-EDF,但略低于 DS-FP。当密度因子为 0.55 时,ML 和 DS-EDF 的成功率下降到 0.85 以下。而在 DS-FP 和 DS-LALF 中,只有当密度因子分别为 0.6 和 0.59 时发生。此外,当密度因子为 0.63 时,几乎所有事务集都不能由 ML 和 DS-EDF 调度,而 DS-FP 和 DS-LALF 的成功率分别仍在 0.27 和 0.1 左右。还能发现 DS-EDF 的调度成功率随不同的密度因子值波动较大。当密度因子值较小时,它的可调度性与 ML,DS-LALF 和 DS-FP 类似,因为工作负载小。但是,当密度因子较高时,与 DS-LALF 和 DS-FP 相比,DS-EDF 的可调度性不稳定,原因是引入的动态调度机制以使用最早截止时间优先来调度更新作业可能使一些作业在被 CPU 长时间服务后错失了它们的截止时间。

4 结束语

CPS 的实时数据服务是其实时性研究的重要部分,该文主要谈论了固定优先级环境下,实时数据库中周期性和非周期性事务的实时调度算法问题,分析研究了几个较为经典的周期性事务实时调度算法 HH 和 ML 以及非周期可延迟调度算法 DS-FP,它们都能很好地维持实时数据时间有效性。此外,研究了基于 DS-FP 的新的和实用的方法,DESH-SC 和 DESH-SA,它们将在线调度开销减少到 $O(1)$ 。与 ML 相比,DS-FP 大大降低了处理器工作负载。因此,当 RTDBS 使用 DS-FP 去跟踪环境变化时,DS-FP 可以提高应用程序事务的性能。DESH-SA 是一种最小化传感器更新工作量并保证有效性约束的非常有效的方法,并且在时间和空间复杂性方面具有高效性。

DS-LALF 是 DS-FP 的增强方法,能在保持实时数据对象有效性的同时,进一步降低 DS-FP 的在线计算开销并能适用于需要动态优先级分配的系统。DS-LALF 与 DS-FP 应用相同的可延迟调度原理以尽可能晚地推迟更新作业的释放时间,最大化来自同一更新事务的两个连续更新作业之间的时差,最小化由更新事务引起的 CPU 工作负载。DS-LALF 不使用最短有效性优先策略分配更新事务的优先级,而根据更新事务在运行过程中的实际松弛度为它们分配优先级。与 DS-FP 相比,DS-LALF 具有更低的在线计算开销和更低的更新工作负载。最后通过大量的仿真实验进

行了论证。

参考文献:

- [1] 向军,李国徽,杨兵,等. 移动实时数据库 QoS 管理和更新事务调度算法[J]. 计算机科学,2008,35(9):103-107.
- [2] SINGH Y J, SINGH Y S, GAIKWAD A, et al. Dynamic management of transactions in distributed real-time processing system[J]. International Journal of Database Management Systems,2010,arXiv:1005.5435:161-170.
- [3] XIONG Ming, HAN Song, CHEN Deji, et al. DESH: overhead reduction algorithms for deferrable scheduling[J]. Real-Time Systems,2010,44(1-3):1-25.
- [4] WANG Peng, XIANG Yang, ZHANG Qi. A service-oriented architecture of cyber-physical system and study on two key issues[J]. Journal of Donghua University: English Edition, 2012,29(4):299-304.
- [5] HAN Song, LAM K, WANG Jiantao, et al. On co-scheduling of update and control transactions in real-time sensing and control systems: algorithms, analysis, and performance[J]. IEEE Transactions on Knowledge & Data Engineering, 2013,25(10):2325-2342.
- [6] XIANG J. QoS management for real-time and mobile database systems based on imprecise computation: QoS management for real-time and mobile database systems based on imprecise computation[J]. Journal of Computer Applications,2008,28(7):1709-1712.
- [7] 白天,范波. 实时更新与复制事务的周期与截止期分派[J]. 湖南理工学院学报:自然科学版,2016,29(2):20-23.
- [8] HUANG Wanqi, DAI Wenbin, WANG Peng, et al. Real-time data acquisition support for IEC 61499 based industrial cyber-physical systems[C]//IECON 2017 - 43rd annual conference of the IEEE industrial electronics society. Beijing:IEEE,2017:6689-6694.
- [9] 邢黎闻. 孙优贤院士论工业信息物理融合系统[J]. 信息化建设,2018(1):10-11.
- [10] KUO S Y, CHOU Y H, CHEN C Y. Quantum-inspired algorithm for cyber-physical visual surveillance deployment systems[J]. Computer Networks,2016,117:5-18.
- [11] LIU Jia, KOU Tianyou, CHEN Qian, et al. On wireless network infrastructure optimisation for cyber-physical systems in future smart buildings[J]. International Journal of Sensor Networks,2015,18(3/4):148-160.
- [12] YOU M, LIU Q, SUN H. New communication strategy for spectrum sharing enabled smart grid cyber-physical system[J]. IET Cyber Physical Systems Theory & Applications, 2017,2(3):136-142.
- [13] 李仁发,谢勇,李蕊,等. 信息-物理融合系统若干关键问题综述[J]. 计算机研究与发展,2012,49(6):1149-1161.
- [14] WANG S, ZHANG G, SHEN B, et al. An integrated scheme for cyber-physical building energy management system[J]. Procedia Engineering,2011,15:3616-3620.
- [15] 杨帆,刘彦,李仁发,等. 一种基于数据驱动的 CPS 建模方法研究[J]. 计算机学报,2016,39(5):961-972.
- [16] YUCELEN T, HADDAD W M, FERON E M. Adaptive control architectures for mitigating sensor attacks in cyber-physical systems[J]. Cyber-Physical Systems,2016,2(1-4):24-52.
- (上接第 112 页)
- programmable custom computing machines. Salt Lake City: IEEE,2011:117-124.
- [6] 王鹏,刘万和,刘锐,等. 基于 SystemVerilog 可重用测试平台的实现[J]. 电子技术应用,2015,41(2):61-64.
- [7] 高虎,郑军,田曾昊. FPGA 板级自动化仿真测试环境框架设计[J]. 微电子学与计算机,2017,34(12):94-98.
- [8] 徐潮. 多 FPGA 原型验证平台关键模块的设计与实现[D]. 长沙:国防科学技术大学,2015.
- [9] 陈锐,门永平,杨文强,等. 基于 UVM 的 FPGA 软硬件联合仿真验证技术研究[J]. 空间电子技术,2017,14(1):38-42.
- [10] LEMON R, COULTER B, ARCHIBALD S, et al. Interface test adapter development & maintenance using a continuity/insulation automatic test station in large scale test systems [C]//IEEE AUTOTESTCON. Anaheim, CA, USA: IEEE, 2009:107-112.
- [11] 何庆. 无线传感器网络节点 SoC 的原型验证[D]. 武汉:华中科技大学,2015.
- [12] BITTNER R, RUF E. Direct GPU/FPGA communication via PCI express[J]. Cluster Computing,2014,17(2):339-348.
- [13] JUSTIN J, LINDSTROM L, JAIN A. Using a plug-in model to simplify and enhance ATE test software capabilities[C]//IEEE AUTOTESTCON. CA:IEEE,2012:105-107.
- [14] TOUBA N A. Survey of test vector compression techniques[J]. IEEE Design & Test of Computers,2006,23(4):294-303.
- [15] 高虎,封二强,赵刚. 基于 Testbench 的 FPGA 实物自动化测试环境设计[J]. 电子技术应用,2018,44(4):48-51.