HLA/RTI 下周期与非周期任务调度的实时性改进

刘述田 戴树岭 张亚琳

(北京航空航天大学 虚拟现实技术与系统国家重点实验室, 北京 100191) (中国航空无线电电子研究所, 上海 200241)

摘 要: 为了把高层体系结构的标准应用到实时系统中,必须确保运行时支撑环境的实时性.联邦成员间相关的任务之间有优先顺序约束关系,在联邦成员内调度运行时较难提供可预测的响应,因而难以保证所有任务尤其是非周期任务的实时性.以任务调度理论的角度在联邦成员内部综合调度周期与非周期任务运行的 D-EDF (Double-Earliest Deadline First)策略,既能舍弃部分冗余数据使周期性任务在截止时间前高效完成,又能调度非周期性任务规则的运行提高实时响应速度,使得联邦成员可以高效地处理有优先顺序约束关系的任务,进而改进了运行时支撑环境的实时性.最后证明了 D-EDF 调度策略的可行性.

关 键 词: 高层体系结构; 运行时支撑环境; 实时; 任务调度; 优先顺序约束

中图分类号: TP 391.9

文献标识码: A **文 章 编 号**: 1001-5965(2014)01-0110-05

Improvement of HLA/RTI real-time performance by scheduling periodic and aperiodic tasks

Liu Shutian Dai Shuling

(State Key Laboratory of Virtual Reality Technology and Systems, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Zhang Yalin

(Chinese Aeronautical Radio Electronics Research Institute, Shanghai 200241, China)

Abstract: In order to apply high level architecture (HLA) in some real time systems, real-time performance of run-time infrastructure (RTI) must be guaranteed. Because of precedence constraints of correlated tasks between federates, it is hard to provide predictable response for all tasks, and it is also difficult to ensure the real-time performance of all tasks, especially aperiodic ones. The double-earliest deadline first (D-EDF) strategy of how to schedule overall periodic and aperiodic tasks inside a federate was discussed from the point view of task scheduling theory. The policy can not only discard some redundant data to ensure the periodic tasks completed effectively before their deadlines, but also schedule aperiodic ones running regularly to improve their real-time responding speed. It makes the system effectively handle the tasks with precedence constraints and consequently improves the real-time performance of RTI. The feasibility of the D-EDF scheduling strategy was proved.

Key words: high level architecture (HLA); run-time infrastructure (RTI); real-time; task scheduling; precedence constraints

高层体系结构(HLA, High Level Architecture)作为分布式仿真的 IEEE 标准,已经被大量的军用和民用系统所采用,至今已经发展到 IEEE 1516-2010[1]. 实时系统是一类其正确性不仅依赖于逻辑计算结果,而且依赖于得到这些结果的时间的系统. 有很多学者尝试把 HLA/RTI (Run-Time Infrastructure)与实时性结合起来研究,因为在实际应用中有很多情形需要在实时分布式仿真

收稿日期: 2013-03-26; **网络出版时间**: 2013-11-20 10:28

网络出版地址: www.cnki.net/kcms/detail/11.2625.V.20131120.1028.001.html

作者简介: 刘述田(1978 -),男,辽宁普兰店人,博士生,liushutianmail@163.com.

系统中按照 HLA 规则开发 RTI. 然而 HLA 并没有把实时性作为一项要求放在规则当中,按照其开发的系统并不能满足实时性要求.

研究者主要在以下 4 个方面提出研究方法对 RTI 的实时性加以改进:①把服务质量(QoS,Quality of Service)机制融合到 HLA 的规则当中,增强系统的可预测性[2],QoS 机制提供差异化服务,保证了网络应用能够根据需求占用带宽,为 HLA/RTI 通信提供了更大的灵活性以及更高效的资源利用率;②在实现 RTI 的本地节点上,文献[2-4]使用多线程及线程池提高任务队列的处理效率,使得系统能够同时处理多个任务队列并且降低了创建与销毁线程的系统开销,增强了系统的可预测性与扩展性;③使用了全局调度、可抢占调度等调度策略,使得系统能够及时处理最紧迫的任务,避免任务错过截止时间[3-4],文献[5]提出以图论的方法研究优先顺序限制条件下的任务调度问题;④文献[6-7]提出了专用通信协议以提高 HLA 的实时性能,但是成本较高,没有被广泛采用.

系统的实时性就是要求系统能够对任意的任务具有可以预测的响应,亦即系统能够在有限的时间内对任务请求完成响应并运行. 上述方法使得 HLA/RTI 应用能够更好地利用系统资源,在联邦成员数量增长时有更好的扩展性,以及更好的 RTI 服务反应能力,这些都显著增强了 RTI 的实时性. 但是这些方法在通信协议、操作系统层次考虑实时性问题,而没有在联邦成员这一层面考虑如何进行任务调度. 多数文献虽然提出了不少方法增强实时性,但是却没有对调度策略的可行性进行分析与证明[2-5],其调度策略因而比较缺少说服力. 周期性任务的时间属性在运行前就已明确,这样在指定的时间点所有的时间限制条件都容易满足,因而是易于预测的[8];但是非周期任务在未开始时并不能确定该任务到达的时间,因此非周期任务给系统实时性带来较大的不确定性,调度非周期任务的运行也相对较难,然而其却在较大程度上影响了对系统实时性的判断. 本文将从在联邦成员内部任务调度这一角度入手,进行周期与非周期任务的综合实时调度,讨论如何增强 HLA/RTI 的实时性,提出任务调度策略以及对这些策略进行调度可行性分析.

1　HLA/RTI 通信模型

RTI 作为仿真系统架构的中间件,必须能够为系统中的仿真节点即联邦成员提供统一的标准,使得它们能够互相通信、互操作以确保行为保持一致. 依据 HLA 的规则,联邦成员间的通信必须通过 RTI 来实现,RTI 是 HLA 的核心. 如图 1 所示,所有的联邦成员交换信息都必须经过 RTI,联邦成员和 RTI 组成一个不受限制的、可扩充的分布式仿真系统.

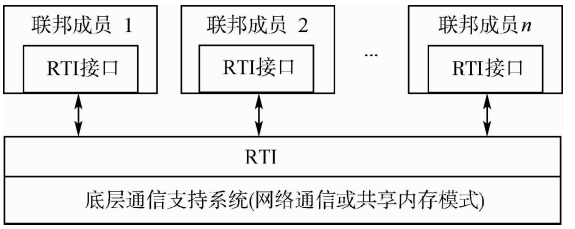


图 1　HLA/RTI 通信模型[3]

联邦成员在每个仿真循环中每隔一个时间间隔通过 RTI 服务,如 sendInteractions( ) 等向其他联邦成员发送信息,如图 2 所示.



图 2　联邦成员任务间优先顺序约束

现实世界中,使用数据的任务只能在产生该数据的任务完成之后发生,这种数据约束关系称为优先顺序约束. 每个联邦成员接收与发送数据以与其他联邦成员进行交互,这些数据是由周期性或非周期性任务产生的,联邦成员处理任务的实时性主要体现在如何处理这些具有优先顺序约束的周期性与非周期性任务.

2　周期任务调度及可行性分析

2.1　任务约束关系定义

实时系统中,通常对一系列周期性任务进行抽象,为每个任务实例分配优先权,在运行时根据优先权进行调度. 周期性任务集合 S 中的一个任务 T_i 可以用一个四元组进行描述[9]:

$$T_i = \langle P_i, E_i, r_i, d_i \rangle$$

任务 T_i 每隔周期 P_i 释放一个实例 $T_i[k]$, $k \in \mathbf{N}$; E_i 表示任务 T_i 的最差情形下的执行时间; $r_i[k]$ 表示任务实例 $T_i[k]$ 的释放时间; $d_i[k]$ 表示任务实例 $T_i[k]$ 的截止时间,如图 3 所示.

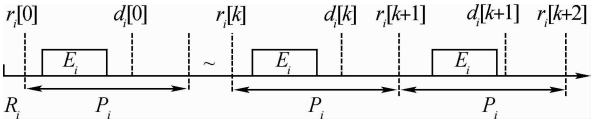


图 3　周期性任务符号定义

定义 1　如果一个调度策略使得任务集合能够满足所有时间限制条件而执行完毕,则称这个调度策略是可行的,该任务集合是可调度的.

定义 2　设 S 为一个任务集合, $T_i, T_j \in S$, 如果必须在 T_i 执行完毕后才能开始执行 T_j, 则称 T_i

优先于 T_j,表示为 $T_i > T_j$,其中符号 $>$ 定义为关系“优先于”.

定义 3 设 S 为一个周期性任务集合,T_i,$T_j \in S$,P_i 和 P_j 分别是 T_i 和 T_j 的周期,$M_{ij} \subseteq \mathbf{N}^2$,对于 $\forall (m,n) \in M_{ij}$,若有 $T_i[m] > T_j[n]$,则称 $T_i > T_j$ 为扩展优先顺序约束. 若 $m = n$ 且 $P_i = P_j$,则称 $T_i > T_j$ 为简单优先顺序约束.

这个关系描述了无限个任务实例的约束关系,注意到 S 是一个周期性任务集合,可以用循环的形式来描述任务实例的约束关系. 对于 $\forall m$,$n \in \mathbf{N}$,用 l_{mn} 表示 m 和 n 的最小公倍数,用 Γ_n 表示区间 $[0,n]$ 内的整数的集合.

定义 4 设 S 为一个周期性任务集合,T_i,$T_j \in S$,P_i 和 P_j 分别是 T_i 和 T_j 的周期,$p = l_{P_iP_j}$,对于 $M_{ij} \subseteq \Gamma_{p/P_i} \times \Gamma_{p/P_j}$,若 $\exists q \in \mathbf{N}$,使得

$$M'_{ij} = \left\{ (m',n') \mid (m',n') = (m,n) + \left(q\frac{p}{P_i}, q\frac{p}{P_j} \right) \right\} \tag{1}$$

则称 $T_j > T_i$ 为周期性扩展优先顺序约束.

这个定义可用图 4 详细解释,其中给出了几种不同周期的任务可能的优先顺序约束关系.

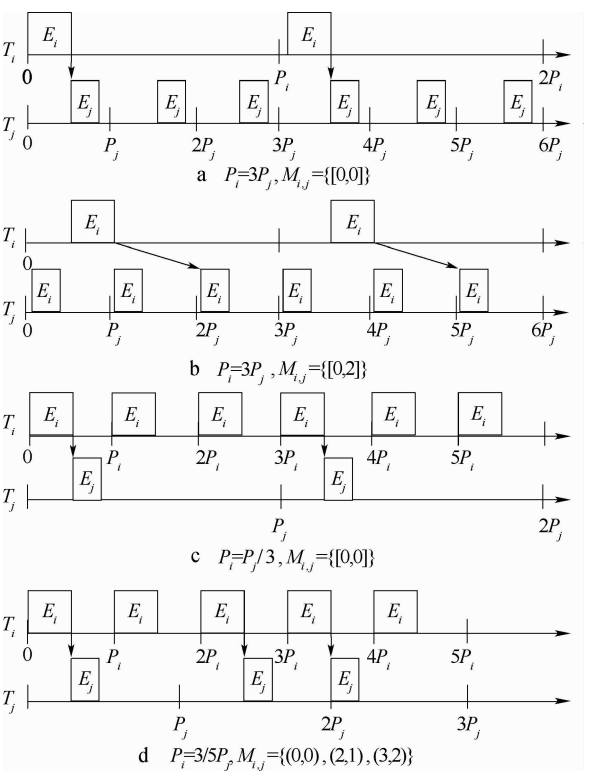


图 4 周期性扩展优先顺序约束关系[10]

例如,图 4a 有 $M_{ij} = \{(0,0)\}$,$T_i[0] > T_j[0]$,$T_i[1] > T_j[3]$,$T_i[2] > T_j[6]$,…;图 4c 有 $M_{ij} = \{(0,0)\}$,$T_i[0] > T_j[0]$,$T_i[3] > T_j[1]$,$T_i[6] > T_j[2]$,…;图 4d 有 $M_{ij} = \{(0,0),(2,1),(3,2)\}$,$T_i[0] > T_j[0]$,$T_i[2] > T_j[1]$,$T_i[3] > T_j[2]$,$T_i[5] > T_j[3]$,$T_i[7] > T_j[4]$,$T_i[8] > T_j[5]$,….

2.2 任务调度方法与分析

多数分布交互式仿真系统,以时间步进(time-stepped)的模式[11]进行时间推进,绝大部分任务都是时间驱动的周期性任务. 任务与任务之间的同步关系体现在传入传出数据队列的处理方式上. 为了增强 RTI 的实时性,加快对联邦成员间的信息处理速度,逐渐放弃了早期 RTI 的阻塞式处理方式,而采用了非阻塞式[12]. 非阻塞式处理方式虽然加快了信息处理速度,但是也增加了对数据处理的不确定性,进而可能严重影响系统处理任务的效率. 如图 4c 所示,任务 2 的周期是任务 1 的周期的 3 倍,在通常情形下,任务 1 在任务 2 的一个周期内发送 3 个数据,处理器接收到数据就会唤醒一个任务来处理这个数据,这样就会造成任务多执行了 2 次,使系统做了重复的工作,造成效率的降低,在频率差别更大的情形下效率下降得更明显. 如果按照定义 3 的方式来定义任务之间的约束关系,由图 4 可以发现,不管传入的信息周期与本联邦成员长短关系如何,每个周期都至多只执行一次处理信息的任务,这样就提高了系统的效率.

考虑每一个任务的信息约束队列,见图 5,其长度与任务周期有如下关系:

$$l = \lceil (1/P_2 - 1/P_1)\Delta t \rceil + l_0 \tag{2}$$

按照上述处理方式,队列缓冲区的长度不短于 $L = \lceil P_1/P_2 \rceil$,否则就会产生数据溢出.

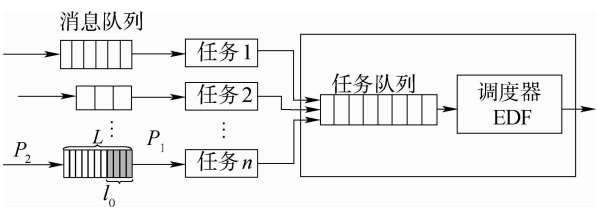


图 5 信息队列与任务队列

单处理器的多个周期性任务的调度策略已经有非常成熟的算法,文献[13]证明了动态优先权调度策略,如最早截止时间优先(EDF, Earliest Deadline First)算法能够获得最高 100% 的 CPU 利用率,并且该算法是最优的. 在 RTI 环境下,单处理器下任务间的约束关系转化为联邦成员间任务间的优先顺序约束关系和时间约束关系. 以任务的角度来看,传入的信息和发送的信息可以视为不同联邦成员的任务之间约束信号,该信号唤醒了任务实例进而执行.

每个联邦成员的任务队列处理如图 5 所示,在任务处理过程中,比较任务就绪队列中每个任务的截止时间,最小的截止时间任务获得最高的优先权,优先运行.

根据文献[13],可得定理 1.

定理 1　设 S 为一个周期性任务集合 $\{T_i \mid 1 \leqslant i \leqslant n, n \in \mathbf{N}\}$,$E_i$ 是任务 T_i 的最差情形下的计算时间,P_i 是任务 T_i 运行周期. 如果 S 是可调度的,当且仅当

$$\sum_{i=1}^{n} \frac{E_i}{P_i} \leqslant 1 \tag{3}$$

引理 1　设 S 为一个周期性任务集合 $\{T_i \mid 1 \leqslant i \leqslant n, n \in \mathbf{N}\}$, E_i 是任务 T_i 的最差情形下的时间, Δt 是步进式联邦成员的运行步长. 如果 S 在步进式联邦成员上按照本节前述的调度方式是可调度的,当且仅当

$$\sum_{i=1}^{n} E_i \leqslant \Delta t \tag{4}$$

证明　充分性. 易见,若 $\sum_{i=1}^{n} E_i \leqslant \Delta t$,则说明在每一个步长内,所有任务都可以完成,由 EDF 的最优性可知, S 是可调度的.

必要性. 若 S 是可调度的,由步进式联邦任务(如图 5)及本节前述的调度方式,每个任务在一个步长内至多只执行一次,按照最多情形计算,有 $P_i = \Delta t$,由定理 1,若 S 是可调度的,则有

$$\sum_{i=1}^{n} \frac{E_i}{P_i} = \sum_{i=1}^{n} \frac{E_i}{\Delta t} \leqslant 1 \tag{5}$$

即

$$\sum_{i=1}^{n} E_i \leqslant \Delta t \tag{6}$$

如果 $P_i > \Delta t$,按照上述 EDF 算法,在某些步长时间内联邦成员有更多的空闲计算时间,式(6)必然成立.　**证毕**

这说明,若一个周期性任务集合在整个时间轴上按照上述调度方式是可调度的,则该任务集合每一个任务的实例在步长 Δt 内必须是可调度的.

若要确保任务的可预测性,一是要确定每个周期处理传入信息队列的某个特定数据. 根据定义 3,只要确定了 M_{ij},那么就可以按照这个选择方式进行处理. 二是要确定该数据在队列中的等待时间. 任意一个任务实例 T_i 的最长响应时间可以表示为

$$t_i = E_i + \sum_{\forall j \in h(i)} \left\lceil \frac{t_i}{P_j} \right\rceil E_j \tag{7}$$

式中, t_i 为任务 T_i 的最长响应时间; E_i , E_j 分别为任务 T_i , T_j 的最长计算时间; P_j 为任务 T_j 的周期; $h(i)$ 为同一任务队列优先权比任务 T_i 高的任务集合.

根据引理 1,若 t_i 大于本联邦成员的步长 Δt ,就说明本联邦成员不能处理完毕所有的任务,系统处理能力不足,则数据队列随着时间的推进不断地增长,达不到实时的性能要求. 另一个侧面也说明,在这种情形下任何一种任务调度策略都是不可行的,因为 EDF 算法是最优的[13].

3　D-EDF 算法及分析

交互的任务属于典型的非周期任务,这种类型的任务由于人在回路中,其实时性要求往往较强,要求事件必须在截止时间前得到响应,否则系统则会给人以严重失真的感觉,甚至会因此做出错误判断.

非周期任务集合 Q 的一个任务 J_i 可以用一个三元组来描述[14]:

$$J_i = \langle E_i, A_i, d_i \rangle$$

式中,任务 J_i 在绝对时刻 $A_i[k]$ 到达一个实例 $J_i[k]$, $k \in \mathbf{N}$; E_i 为任务 J_i 的最差情形下的执行时间; $d_i[k]$ 为任务实例 $J_i[k]$ 的绝对截止时间. 这些定义与周期任务相似,可以参考图 3.

周期任务与非周期任务的混合任务的调度算法通常采用“服务器”调度策略[15-16]. 这种策略把非周期性任务当作周期性任务处理,加快对非周期任务的响应速度.

时间步进式联邦的特点是,所有的任务实例要么在一个周期内完成,要么在下一个周期内完成,不能跨越 2 个周期. 以 $s_i[k]$ 表示实例 $J_i[k]$ 的开始执行时间, $f_i[k]$ 表示实例 $J_i[k]$ 的执行完成时间,若 $J_i[k]$ 在第 m 个周期内完成,则有

$$s_i[k] \geqslant m\Delta t \wedge f_i[k] \leqslant (m+1)\Delta t \tag{8}$$

$$f_i[k] \leqslant d_i[k] \tag{9}$$

根据周期性任务在每一个周期的运行情况,事先确定每个周期内非周期任务可以占用的最大比率 U_{A} ,故第 m 个周期可以调度的非周期任务集合 $F(m)$ 必须满足如下条件:

$$\sum_{\forall i \in F(m)} E_i \leqslant \Delta t \cdot U_{\mathrm{A}} \tag{10}$$

则可以保证第 m 个周期可以调度的任务满足式(8),若要同时满足式(9),则必须对 $F(m)$ 使用 EDF 调度算法.

上述采用 2 次 EDF 调度算法的策略,命名为 D-EDF(Double-EDF)算法.

定理 2　给定一个周期性任务集合 $S = \{T_i \mid 1 \leqslant i \leqslant n, n \in \mathbf{N}\}$ 处理器利用率为 U_{P} ,以及一个非周期性任务集合 $Q = \{J_i \mid 1 \leqslant i \leqslant m, m \in \mathbf{N}\}$ 处理器利用率为 U_{A} ,如果整个任务集合是 D-EDF 可调度的,当且仅当

$$U_{\mathrm{P}} + U_{\mathrm{A}} \leqslant 1 \tag{11}$$

证明 D-EDF 按照第 2 节的策略产生一个唯一的调度任务列表 X，并且 X 是周期性的，其处理器利用率 $U = U_P + U_A$，即若 X 是可行的，当且仅当 $U \leqslant 1$. 注意到定理 1，则可证明 $U \leqslant 1$，即 $U_P + U_A \leqslant 1$. **证毕**

4 结束语

将任务调度策略应用到 RTI 中，在联邦成员这一层次上综合调度周期与非周期任务的运行，提高了 RTI 的实时性，使得 HLA 可以应用到实时系统中，提高开发效率. 具有优先顺序约束的任务调度的可预测性在很大程度上依赖于数据传输的确定性，即确定了发送数据的长度与时间就可以确定收到数据的长度与时间. 但是由于分布式应用环境的特殊性，网络通信有很多不确定性，网络环境需要能够保证通信延迟的上限或者能够提供 QoS 通信，以调度远端进程满足任务的截止时间，减少或者消除任务调度的抖动. 任务调度也要考虑通信延迟的不确定性，在提高任务调度的可预测性的同时也要减少数据顺序不一致带来的不确定性，这些都是以后可以研究的内容.

参考文献（References）

[1] IEEE 1516-2010 Standard for modeling and simulation (M&S) high level architecture (HLA): framework and rules[S]

[2] Zhao Hui, Georganas N D. HLA real-time extension[C]//Distributed Simulation and Real-Time Applications, Fifth IEEE International Workshop on, DS-RT 2001. Cincinnati: IEEE, 2001: 12 - 21

[3] Guan Li, Zou Ruping, Zhu Bin. An HLA/RTI architecture based on multi-thread processing[J]. Jornal of China Ordance, 2010, 6(3): 182 - 188

[4] Boukerche A, Lu K. A novel approach to real-time RTI based distributed simulation system[C]//38th Proceedings of Simulation Symposium Proceeding ANSS'05. Washington DC: IEEE Computer Society, 2005: 267 - 274

[5] Adelantado M, Siron P. Towards an HLA run-time infrastructure with hard real-time capabilities[C]//International Simulation Multi-Conference. Ottava: Pierre Siron, 2010: 12 - 14

[6] Mike D B, Zyda M, Watsen K, et al. Virtual reality transfer protocol (vrtp) design rationale[C]//Proc of the Sixth IEEE International Workshop on Enabling Technologies. MIT: IEEE Computer Society Press, 1997: 18 - 20

[7] Zhao Hui. HLA streaming and real-time extensions[D]. Ottawa: University of Ottawa, 2002

[8] Tang H. Combining hard periodic and soft aperiodic real-time task scheduling on heterogeneous compute resources[C]//2011 International Conference on Parallel Processing (ICPP). Taipei: IEEE, 2011: 753 - 762

[9] Buttazzo G C. Periodic task scheduling[M]. New York: Springer, 2011: 9 - 118

[10] Forget J, Boniol F, Grolleau E. Scheduling dependent periodic tasks without synchronization mechanisms[C]//16th IEEE Real-Time and Embedded Technology and Applications Symposium. Stockholm: IEEE, 2010: 301 - 310

[11] Mclean T, Fujimoto R, Fitzgibbons B. Middleware for real-time distributed simulations[J]. Concurrency and Computation: Practice & Experience-Distributed Simulation and Real-Time Applications, 2004, 16(15): 1483 - 1501

[12] Fujimoto R, Mclean T, Perumalla K, et al. Design of high performance RTI software[C]//Fourth IEEE International Workshop on Distributed Simulation and Real-Time Applications. San Francisco: IEEE, 2000: 89 - 96

[13] Liu C L, Layland J W. Scheduling algorithms for multiprogramming in hard real-time environments[J]. Journal of the ACM, 1973, 20(1): 46 - 61

[14] Buttazzo G C. Aperiodic task scheduling[M]. New York: Springer, 2011: 53 - 78

[15] Spuri M, Buttazzo G C. Scheduling aperiodic tasks in dynamic priority systems [J]. Real-Time Systems, 1995, 10(2): 179 - 210

[16] Marchand A, Silly-Chetto M. Dynamic real-time scheduling of firm periodic tasks with hard and soft aperiodic tasks[J]. Real-Time Systems, 2006, 32(1): 21 - 47