ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.10, October 2006, pp.2040−2047
DOI: 10.1360/jos172040

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

异构分布式实时仿真系统的容错调度算法*

刘云生+, 张 童, 张传富, 查亚兵

(国防科学技术大学 机电工程与自动化学院,湖南 长沙 410073)

A Fault-Tolerant Scheduling Algorithm for Heterogeneous Distributed Real-Time Simulation Systems

LIU Yun-Sheng+, ZHANG Tong, ZHANG Chuan-Fu, ZHA Ya-Bing

(School of Mechatronics and Automation, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: Phn: +86-731-4576403, E-mail: lysliberty@yahoo.com.cn

Liu YS, Zhang T, Zhang CF, Zha YB. A fault-tolerant scheduling algorithm for heterogeneous distributed real-time simulation systems. *Journal of Software*, 2006,17(10):2040−2047. http://www.jos.org.cn/1000-9825/17/2040.htm

Abstract: Heterogeneous distributed real-time simulation system is a special kind of real-time system. This paper solves its fault-tolerant problem based on the CSP (checkpoint-based spare processor) model, which is an improvement of the traditional SP (spare processor) model with checkpoint mechanism. Firstly, two propositions are put forward based on the characters of simulation system. Secondly, the Worst Case Response Time of the simulation tasks are analyzed based on Markov chains and the schedulability analysis rules for simulation task are presented. In the end of the paper, a fault-tolerant scheduling algorithm CSP-RTFT is proposed and simulated. The results show that the algorithm can achieve better stability, higher task accept ratio than SP-RTFT which is based on SP model, whereas the resource utilization ratio is lower than those based on PB model.

Key words: heterogeneous distributed simulation system; fault tolerance; real-time scheduling; Markov chain; worst case response time

摘 要: 异构分布式实时仿真系统是一类特殊的实时系统,基于改进的 SP(spare processor)容错模型(checkpoint-based spare processor,简称 CSP)对其容错问题进行了研究.首先,根据仿真系统的特点提出了两个命题,这是后续工作的基础;而后,基于 Markov 链对仿真任务的最坏反应时间进行了分析,并提出了仿真任务的可调度性分析规则;最后,基于 CSP 容错模型和上述可调度分析规则提出了异构分布式实时仿真系统的容错调度算法 CSP-RTFT.算法的仿真结果表明:该算法较之基于 SP 模型的算法 SP-RTFT 可获得更好的稳定性、更高的任务接收率;缺点是资源利用率比 PB 模型下的算法要低.

关键词: 异构分布式仿真系统;容错;实时调度;Markov 链;最坏反应时间
中图法分类号: TP391 文献标识码: A

* Supported by the Defense Pre-Research Project of the 'Tenth Five-Year-Plan' of China under Grant No.51404010403KG0155 (国家"十五"国防预研基金)

Received 2005-05-07; Accepted 2005-12-31

任何成功的新技术从实验、论证到应用、推广,由于使用环境、用户需求的变化及相关技术的发展,都需要经历一个不断调整的过程.1983 年开始出现的先进分布仿真技术(ADS),从最初的 SIMNET 逐步演化到目前广泛采用的 HLA(high level architecture)及将来可能采用的 XMSF(extendable modeling and simulation framework)[1,2],其发展历程也是如此.分布式仿真系统作为分布式系统的一种,必然要解决可靠性问题.例如:若把分布式仿真系统作为一个实时决策辅助系统嵌入到 C4ISR 中,为指挥、参谋人员提供实时决策支持,那么,由于仿真系统崩溃或可靠性方面的缺失导致的决策延迟或错误,其后果将很可能是灾难性的.但一直以来,仿真系统的可靠性问题并没有得到足够的重视.由于近年来 ADS 在军事、国民经济领域内的作用和地位越来越重要,及在实际使用过程中的一些经验、教训,该问题才引起足够的重视并成为研究的热点[3−5].

容错是提高系统可靠性的主要手段之一,在分布式仿真系统中,容错问题的解决主要包括上层对仿真系统相关体系结构的扩展,协议的修订,以及底层对具体容错模型、容错调度算法的研究.实际上,上层通过扩展体系结构和修订协议所获得的容错性能的提升,最终还是要通过底层具体的容错模型、调度算法来实现.也就是说,对容错模型、容错调度算法的研究是提高系统容错性能的基础.本文将重点对具体的容错模型、调度算法进行研究.此外,随着集群及网格技术[6]的发展,系统中异构资源将不可避免.因此,在对具体调度算法的研究中,将考虑资源异构的影响.

1 相关工作

文献[3−5]对分布式仿真系统的容错问题进行了初步研究:文献[3]提出并利用网络服务及语义网技术实现了一个分布式资源管理系统 DRMS,该系统可用来解决 HLA 仿真系统的容错问题;文献[4]对采用乐观时间推进机制的仿真系统的回卷协议进行了研究;Björn Möller[5]对 HLA 仿真系统中可能出现的故障、HLA 的容错扩展及仿真系统的容错设计模式进行了系统的研究.但上述这些研究大都局限于系统体系结构这一层次,而没有涉及到底层具体的容错模型及调度算法.实际上,容错调度一直都是分布式系统的研究热点,在这方面已经做了大量的工作.

对同构环境中实时容错调度的研究已经相对成熟:文献[7,8]对周期任务的调度进行了研究;文献[9]就动态、非周期、可抢占的实时任务的调度问题进行了研究,提出并验证了利用副版本重叠和调度时间回收技术来提高任务集可调度性的可行性;文献[10]就周期任务和软非周期任务(soft aperiodic tasks)的混合调度问题进行了研究,提出了一种通过调整软非周期任务的截止时限而将其转化为硬实时周期任务的方法.

对异构环境中容错调度的研究相对较少.实际上,由于集群、网格技术的发展,异构系统容错调度在最近几年才成为热点.同构系统和异构系统调度的最大区别就是:在同构系统中,同一任务在所有处理机上的运行时间、处理机发生故障的频率相同;而在异构系统中则不然.这主要是由于异构系统中不同的软、硬件配置所致.文献[11]基于可靠性代价研究了任务的分配问题,但是它不具有实时特征.文献[12]基于可靠性代价提出了一种实时容错调度算法.

上述工作的研究对象都是普通分布式实时系统,没有考虑分布式实时仿真系统的特点,不能直接用来解决其容错调度问题.本文将在上述工作的基础上,根据分布式实时仿真系统的具体特点研究其调度问题.

2 模型描述

2.1 系统模型

HLA 仿真系统由 N 个互联的异构处理机构成,处理机可以是 PC、集群及工作站等.为了表示系统中处理机的这种异构性,我们引入处理机超集:$\Omega=\{P_1,P_2,\ldots,P_l\}(1\le l\le N)$,其中,$P_i$ 表示同构(类)处理机的集合.$P_i=(p_1,p_2,\ldots,p_k)$,$|P_i|$表示该类处理机的数量.系统中发生的故障类型为失败停(fail-stop),本文仅考虑由于硬件不稳定及断电等造成的处理机停止响应或崩溃等这类硬件故障,假定软件及网络是无错的,在任意时刻系统中只能发生一个故障.每个处理机中连续两次故障的时间间隔服从均值为 $1/\lambda_i$ 的负指数分布[13],处理机间的故障是独立

的.$p_i=(\Delta_i,\lambda_i)$,其中,Δ_i 表示已分配到处理机 p_i 上的任务的集合.

2.2 任务模型

首先给出分布式实时仿真系统中任务及任务集的定义:所谓仿真任务就是成员[1]进程,一个成员就是一个仿真任务,暂不考虑一个成员包括多个进程的情况;一个仿真应用所包括的所有仿真成员就构成了该仿真应用的任务集;本文用 $V=[v_1,v_2,\ldots,v_m]$来表示一个仿真任务集,定义每个仿真任务 v_i 为一个 3 元组 $v_i=(a_i,d_i,c_i)$,a_i,d_i,c_i 分别表示任务的到达时间、截止时限和执行时间.由于系统异构性的影响,同一 v_i 在不同的异构处理机上的 c_i 是不同的,我们为描述这种差异,为每个任务 v_i 定义一个计算向量 $C_i=[c_{(i,1)},c_{(i,2)},\ldots,c_{(i,l)}]$,$c_{(i,j)}$表示任务 v_i 在处理机类 P_j 上的执行时间.任务之间无约束关系(根据文献[14],有前驱关系的任务集可转化为相互独立的任务集).

上述仿真任务特别适用于实时决策辅助的仿真任务,其到达时间随战场态势动态变化,无规律可循,我们将其归为非周期任务.当然,不排除有些仿真任务属于周期性任务,如在某些仿真系统中,需要重复运行同一仿真任务以产生大量的实验数据,本文暂不研究这类周期性仿真任务的调度问题.

2.3 容错模型

SP(spare processor)[15]是一种广泛采用的容错模型.在该模型中,当处理机发生故障时,一般利用重新执行来进行故障恢复.但若在任务运行时对其定期执行检查点,则可极大地减少由故障导致的计算损失.因为当发生故障时,受影响的任务只需最多回卷一个检查点间隔即可[16],显然,这样会增强整个任务集的可调度性.根据上述分析,本文将用检查点机制改进 SP 模型.下面首先给出 CSP 的具体定义:

定义 1. 基于检查点的空闲处理机模型(checkpoint-based spare processor,简称 CSP).

所谓 CSP 是指一种容错模型.在该模型中,系统提供一台或数台空闲的备用处理机,当系统正常运行时,每隔一定的时间对系统执行一次检查点.当系统中有节点发生故障时,则把发生故障的处理机上的任务利用相应检查点文件在与其同构的备份处理机上进行恢复.

这里之所以在同构处理机上进行故障恢复的原因是:检查点文件本质上是任务进程空间的一个快照(snapshot),而不同类型的处理机对进程空间的管理不同,这就导致不同类型的处理机产生的检查点文件很可能不能通用.所以在故障发生后,只能在同构的处理机上进行故障恢复.当然,目前异构系统的进程迁移技术已经有所进展,但与实用还有很大差距,在这里暂不考虑这种技术.

本文采用一种协同检查点协议方式来保存系统状态.该协议为保证系统状态的一致性,在进行保存时仿真任务间需要进行某种方式的协同.当发生故障时,所有仿真任务都要回卷.关于协同检查点协议的具体论述可参见文献[17],在此不再赘述.

此外,由于需要预留备份处理机,所以在进行任务分配时系统中只有 $M(M<N)$台处理机可用,其他$(N–M)$台处理机用作备份机.显然,若利用该模型进行容错,只要提供足够数量的备份处理机,则系统在执行过程中可以容忍发生 $n(n>1)$次故障.

3 仿真任务可调度性分析

任务可调度性分析大致可分为两类:一类基于处理机利用率;一类基于任务的最坏反应时间 WCRT(worst case response time)[18].本文将基于 WCRT 分析任务的可调度性.

最坏反应时间 r_i 定义为任务完成时间与任务释放时间在最坏情况下的差值[18].这里所谓的最坏情况,就是考虑与其他任务的同步、通信开销、高优先级任务抢占、在执行过程中发生处理机故障等.

下面对仿真任务的 WCRT 进行分析.由于分布式实时仿真系统有别于一般的分布式实时系统,而这些特殊性将会影响仿真任务的 WCRT 分析,因此,我们就其特殊性提出如下命题.

命题 1. 同一处理机上的所有仿真任务应按照一定的调度算法交叉执行.

同一处理机上的不同仿真任务之间是并行的,且存在通信、同步关系,而某一时刻 CPU 上只能运行一个任务,所以操作系统的调度算法必须保证所有任务可以交叉执行以保证上述通信及同步.显然,该算法应该是抢占

式的.

实际上,由于不同仿真成员所需解算模型的差异(导致计算量不同)及成员间的数据依赖关系,各个仿真任务之间不可能采用严格的交叉执行(如时间片轮换调度算法).但从总体上来看,为了保证同一处理机上任务之间的通信和同步,这种交叉执行又是必须的(尽管有时不是严格的交叉执行).所以,命题 1 反映了操作系统调度算法的总体特征.下面根据该命题就仿真任务的 WCRT 分析提出命题 2.

命题 2. 处理机 p_j 上的仿真任务 v_i 的最坏反应时间为

$$r_i = b_i + i_i + \sum_{v_k \in \Delta_j}(c_k + f_k) \tag{1}$$

其中,b_i 表示 v_i 的通信、阻塞开销(由成员间同步及数据交换导致的阻塞);i_i 表示在 v_i 执行过程中高优先级任务的影响;第 3 项表示 p_j 上的所有仿真任务在一定失效模型下的运行时间之和,其中 f_k 表示容错开销.

根据命题 1,在不考虑任务切换及通信、阻塞开销的前提下,每个仿真任务的 r_i(WCRT)是该处理机上所有仿真任务在一定失效模型下的运行时间之和.但是在实际仿真系统中,由于在大部分仿真任务间存在频繁的数据交互,所以还要考虑通信及同步、数据交换导致的阻塞的影响.如果该处理机还运行有其他高优先级任务,则在分析仿真任务的 r_i 时,还应考虑这些高优先级任务的影响.

此外,为表述方便,下文中将根据具体情况分别采用 WCRT 和 r_i 来指代任务的最坏反应时间.

3.1 仿真任务的最坏反应时间分析

文献[18]对单处理机中任务的 r_i 进行了分析,但是其相关结论是在假定存在一个最小故障发生间隔的前提下获得的,属于确定性分析的范畴.而实际上,由于来自软、硬件方面不确定性及工作环境的影响,所谓故障的最小间隔很难获得.基于上述原因,本文将基于我们在文献[16]中所做的工作来对仿真任务的 r_i 进行重新研究.在文献[16]中,我们基于 Markov 链对仿真任务的实际执行时间进行了研究.

根据文献[16]的结论,考虑检查点、故障恢复开销的影响,并假设节点故障服从泊松分布的前提下,完成一个检查点间隔实际需要的时间 *tInt* 为

$$tInt = \frac{1}{M\lambda}(\mathrm{e}^{M\lambda(T+Ch+R)} - \mathrm{e}^{M\lambda R}) \tag{2}$$

其中,M 是系统中运行仿真任务的处理机的数量;λ是所有处理机故障率的平均值;T 是检查点间隔;Ch 是检查点开销;R 是故障恢复开销.为便于分析,假定λ,Ch,R 为常量[16].

根据式(1),考虑不同仿真任务通过 RTI(run-time infrastructure[1])进行的通信、阻塞开销及高优先级任务影响,仿真任务 v_i 在处理机 p_p 上的 WCRT 为

$$r_i = b_i + \sum_{v_j \in hp(p)} \left\lceil \frac{r_i}{t_j} \right\rceil o_j + \sum_{v_k \in \Delta_p \cup v_i} \left\lceil \frac{c_k}{T} \right\rceil tInt \tag{3}$$

其中,第 1 项表示通信、阻塞开销;第 2 项表示高优先级任务的影响,$hp(p)$表示 p 上高优先级任务的集合,o_j,t_j 分别表示相应运行的时间、周期(这里考虑相对复杂的周期性高优先级任务的影响,对于非周期性高优先级任务,用 i_i 替代该项即可);第 3 项表示已分配到该处理机上仿真任务及该任务本身对其 WCRT 的影响.

由于 r_i 是一个非减量,具体求解可以通过如下迭代方程来实现:

$$r_i^{n+1} = b_i + \sum_{v_j \in hp(p)} \left\lceil \frac{r_i^n}{t_j} \right\rceil o_j + \sum_{v_k \in \Delta_p \cup v_i} \left\lceil \frac{c_k}{T} \right\rceil tInt \tag{4}$$

假定 $r_i^0 = c_i$,当 $r_i^{n+1} = r_i^n$ 时,就可以获得最终解,即 $r_i = r_i^n$.显然,如果 $r_i^{n+1} > d_i$,则该任务不可调度,迭代停止.

由式(3)可见:在高优先级任务及通信、阻塞开销的影响确定的前提下,WCRT 主要取决于第 3 项,即故障的平均发生频率λ、检查点间隔 T、检查点开销 Ch 等参数.由于已经假定λ,Ch,R 为定值,所以,WCRT 主要取决于 T,而根据文献[16],可以得到 T 的最优解.根据该最优解执行检查点,则可获得 *tInt* 的最小值,显然,此时也可得到 v_i 在处理机 p 上的最小 WCRT.由于容错仿真系统在实际运行过程中会基于这个 T 的最优解进行状态保存,所以,与此相对应,算法也将基于任务的最小 WCRT 进行可调度分析.

当基于重新执行(即传统的 SP 模型)进行故障恢复时,由于在此过程中不做检查点,所以,检查点间隔就等效于实际的执行时间,且 $Ch=R=0$.这里,直接给出 SP 模型中仿真任务 v_i 在处理机 p_p 上的 WCRT 的求解迭代方程:

$$r_i^{n+1}=b_i+\sum_{v_j\in hp(p)}\left\lceil\frac{r_i^n}{t_j}\right\rceil o_j+\sum_{v_k\in\Delta_p\cup v_i}\frac{1}{M\lambda}(e^{M\lambda c_k}-1) \tag{5}$$

下面给出仿真任务的可调性分析规则.

3.2 可调度性分析规则

根据命题 2,在任务分配过程中,当处理机上有新的仿真任务加入时,将导致该处理机上所有仿真任务的运行时间之和增大.也就是说,后加入的任务会影响先加入任务的可调度性.所以,每当有新任务加入时,已经分配到该处理机上的所有任务的可调度性都要重新分析(这也是分布式仿真系统的可调度性分析有别于一般实时系统之处).基于上述分析,提出基于 CSP 模型容错的分布式仿真系统的可调度性分析规则:

在分配 v_i 到 p_p 时,根据式(3)重新计算该处理机上所有任务的最坏反应时间(包括 v_i),若此时该处理机上所有仿真任务的最坏反应时间都小于其相应截止时限,即$\forall v_k\in(\Delta_p\cup v_i),r_k<d_k$,则 v_i 在处理机 p_p 上可调度;否则不可调度.

实际上,上述调度准则不仅适用于仿真任务,而且适用于一类实时任务调度,这类任务的特点是:同一处理机中实时任务之间存在交互、同步,任务需交叉执行.下面根据上述分析,给出相应的容错调度算法.

4 算法描述

调度算法分为两部分:任务分配算法和发生故障后的调度算法.下面,我们首先给出基于 CSP 模型进行容错时的任务分配算法 CSP-RTFT.

在分配 v_i 时,CSP-RTFT 首先根据式(3)来计算该任务在系统中 M 台处理机上的 WCRT,将 M 台处理机按照 WCRT 的大小进行非减排序,从拥有最小 WCRT 的处理机开始,根据上述可调度性规则进行可调度分析,而后将该任务分配到第 1 个满足可调度性分析的处理机上:若所有的处理机都不满足,则对 v_i 的分配失败,算法返回 sch_failed;若所有任务都满足可调度性分析,则算法返回 sch_success.其伪代码如下所示:

```
(输入:V,Ω,C;输出:sch_success 或者 sch_failed,Ω){
    for (i=1 to m) {
        根据式(3)计算 v_i 在所有处理机上的 WCRT,并将所有处理机按照 WCRT 的大小进行非减排序;
        for (j=1 to M){
            根据式(3)重新计算Δ_j 中原有任务的 WCRT;
            if (∀v_k∈(Δ_j∪v_i),r_k<d_k){
                p(v_i)=p_j; /*将 v_i 分配到 p_j 上*/
                break;
            }
        }
        if (没有找到合适的处理机)
            return (sch_failed);
    }
    return (sch_success);
}/*算法结束*/
```

系统中发生故障后的调度非常简单:将故障机上的任务,利用相应的检查点文件,在备份的同构处理机上回卷即可恢复.由于采用协同检查点协议进行状态保存,此时,其他节点上的仿真任务也需要回卷.

SP 模型下的容错调度算法 SP-RTFT 与上述算法类似,只是任务 WCRT 的计算是根据式(5),这里不再赘述.

5 算法仿真

本节将以算法的稳定性、任务接收率及处理机利用率为评价标准来比较 CSP-RTFT 算法和 SP-RTFT 算法.

仿真参数设定如下:系统由 5 类处理机构成,每类处理机包括 3 台同构处理机,在 CSP 及 SP 中,都是预留 1 台作为备份处理机.异构处理机的故障率不同,同构处理机的故障率相同.假设所发生的故障为短暂故障,发生故障的处理机在修复后又可以作为其他 2 台处理机的备份机重新加入到系统中;仿真任务总数为 30 个,同一仿真任务在同构处理机上的运行时间相同,在异构处理机上的运行时间服从[3000,3500]s 的均匀分布,不同仿真任务的通信、阻塞开销服从[10,300]s 的均匀分布.暂不考虑高优先级任务的影响(实际上,在仿真系统中,处理机都是专用的,除必需的系统进程以外,一般不会有其他无关高优先级进程).具体的仿真步骤为:

(1) 设定仿真任务集的截止时限为 15000s,任务的截止时限与任务集的截止时限相同;

(2) 随机生成每个任务的在 5 类处理机上的计算向量 C_i;

(3) 分别基于 CSP-RTFT 算法和 SP-RTFT 算法对任务进行调度.

在调度时,根据文献[16]提供的最佳检查点间隔来计算任务的最坏反应时间,λ取所有处理机故障率的平均值.此外,为比较两种算法的性能,在对仿真任务的分配过程中没有考虑截止时限的影响.

5.1 算法的稳定性

首先分析算法的稳定性,即不同的 λ 对任务 WCRT 的影响.

取不同的λ对同一组任务进行调度,调度中所采用的处理机平均故障率分别为:0.0000588,0.0000061,0.0000005(每台处理机的故障率不予列出),调度结果如图 1~图 3 所示.根据 CSP 模型,由于发生故障前后故障机、备份机所运行的任务相同,所以在分析中不对故障机和备份机加以区分.

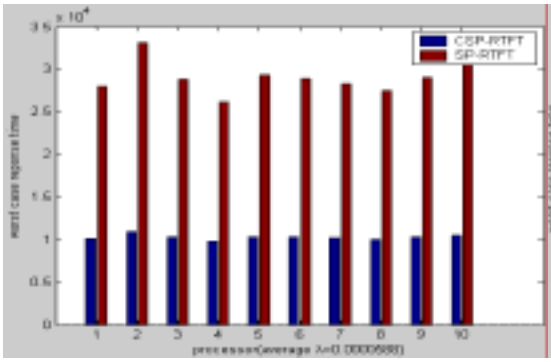

Fig.1 The WCRT with high failure rate

图 1 高故障发生频率时的 WCRT

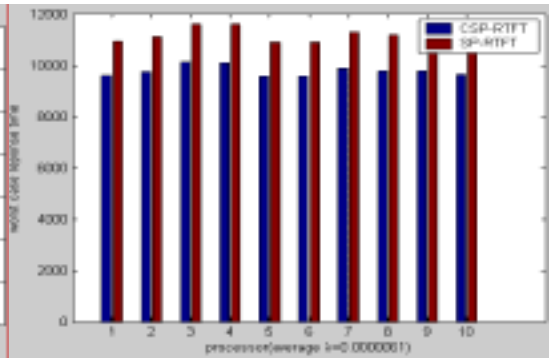

Fig.2 The WCRT with medium failure rate

图 2 中故障发生频率时的 WCRT

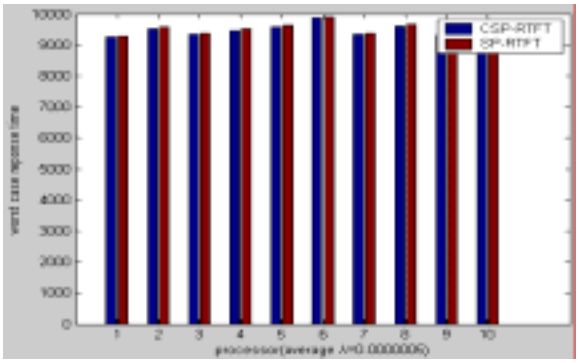

Fig.3 The WCRT with low failure rate

图 3 低故障发生频率时的 WCRT

由图可见:对于同一任务集,不同的故障发生频率(λ)对采用 SP 模型进行容错任务的 WCRT 影响非常大,而对采用 CSP 模型进行容错任务的 WCRT 影响则相对较小.这主要是因为每次发生故障后,在 SP 模型下故障机上的所有任务都要重新执行的缘故.当故障发生频率较低时,对任务的 WCRT 影响还比较小,但如果故障发生频率很高,频繁地重新执行则会导致任务的 WCRT 成倍地增长,这一点从图 1、图 3 的对比中可以很明显地看出来.在 CSP 模型下,检查点间隔会根据不同的故障发生频率,依据文献[16]的结论进行调整:当故障频率很高时,检查点间隔随之变小;当故障发生不是很频繁时,则检查点间隔会适度增大.由于每次发生故障最多只损失一个检查点间隔的计算时间,这种检查点设置策略显然会在一定程度上抵消不同的故障频率给任务的 WCRT 带来的影响,使任务的 WCRT 相对稳定.所以,CSP-RTFT 算法不易受系统故障率变化的影响,具有更好的稳定性.

5.2 任务的接收率及处理机利用率

如图 1 所示,当λ较大时,在 SP-RTFT 算法中,所有的任务都会在截止时限后完成,任务的接收率为 0;在 CSP-RTFT 算法中,所有的任务都会在截止时限前结束,任务的接收率为 100%.所以,CSP-RTFT 算法具有更高的任务接收率.当λ变小时,两种算法中任务的最坏反应时间都会有所降低,这会提高两种算法的任务接收率.但是比较而言,λ的变化对 SP-RTFT 算法任务的接收率的影响远大于对 CSP-RTFT 算法的影响,具体原因见上节.

就处理机利用率而言,在假设短暂失效的恢复时间可以忽略的前提下,系统中一直有 5 台处理机作为备份机,所以处理机的利用率为 10/15=67%.显然,CSP 模型加之资源异构性的影响极大地降低了系统中处理机的利用率.

应该注意到:由于每个任务的 c_i 不同,所以,在每个处理机上任务的 WCRT 应该会有所不同.但是,由于任务同步、通信的影响又决定了这种差别不会很大.所以,基于上述分析及命题 2,我们不对处理机上所有任务的完成时间及该处理机上每个任务的 WCRT 加以区分.

此外需要说明的是,由于对通信、阻塞开销难以准确地度量,在算法仿真过程中,我们假设其服从一个均匀分布,这可能会在一定程度上影响算法仿真的结果.但由于该部分开销对任务 WCRT 的影响要远小于任务执行时间的影响,所以对仿真评估结果的影响不会很大.

6 结 论

由上可见,CSP-RTFT 算法比 SP-RTFT 算法具有更好的稳定性及更高的任务接收率;与 PB[9]模型下同一任务只可以容忍一次故障相比[12],CSP 模型可以容忍多次故障.当然,在 PB 模型中,可以通过多次复制获得任务的多个副版本,而后将这些副版本调度在不同的处理机上以容忍多次故障,但这样,相关的调度问题就非常复杂.尽管 CSP 较 SP 或 PB 模型下的调度算法有如上优势,但是 CSP 会造成严重的资源浪费,在异构系统中,这种浪费更加明显;而在 PB 模式下,处理机可以互为备份,资源利用率相对较高,在本文的后续部分将对 PB 模型下的容错调度算法进行研究.

总之,CSP 模型较适合于计算资源充裕的环境,相应的任务调度比较简单;如果资源很紧缺,难以提供足够的备份处理机,则 PB 模型会更占优势,但此时任务调度的复杂度也会有所增加.本文的后续工作将对基于 PB 模型的容错调度算法及通信、阻塞开销对仿真任务 WCRT 的影响的精确度量进行研究.

致谢 在此,我们向对本文的工作给予大力支持和建议的国防科学技术大学三院军用仿真技术研究室的黄柯棣教授、邱晓刚主任、黄健副主任及乔海泉博士表示感谢.

References:

[1] Huang KD, *et al*. System Simulation Technology. Changsha: Publishing House of National University of Defense Technology, 1998. 307−326 (in Chinese).

[2] Brutzman D, Zyda M, Pullen JM, Morse KL. Extensible modeling and simulation framework (XMSF): Challenges for Web-based modeling and simulation. Proc. of the Technical Challenges Workshop, Strategic Opportunities Symp. Monterey, 2002. http://www.MovesInstitute.org/xmsf

[3] Eklöf M, Moradi F, Ayani R. A framework for fault-tolerance in HLA-based distributed simulations. In: Kuhl ME, Steiger NM, Armstrong FB, Joines JA, eds. Proc. of the 2005 Winter Simulation Conf. Orlando: IEEE Press, 2005. 1182−1189.

[4] Damani OP, Garg VK. Fault-Tolerant distributed simulation. In: Alberta B, ed. Proc. of the 12th Workshop on Parallel and Distributed Simulation. Washington: IEEE Press, 1998. 38−45.

[5] Möller B, Karlsson M, Löfstrand B. Developing fault tolerant federations using HLA evolved. In: Proc. of the 2005 Spring Simulation Interoperability Workshop. San Diego, 2005. http://www.sisostds.org/siw/05spring/readlist.htm

[6] Xu ZW, Feng BM, Li W. Grid Computing Technology. Beijing: Publishing House of Electrnics Industry, 2004. 1−19 (in Chinese).

[7] Han CC, Shin KG, Wu J. A fault-tolerant scheduling algorithm for real-time periodic tasks with possible software faults. IEEE Trans. on Computers, 2003,52(3):362−372.

[8] Manimaran G, Murthy CSR. A fault-tolerant dynamic scheduling algorithm for multiprocessor real-time systems and its analysis. IEEE Trans. on Parallel and Distributed Systems,1998,9(11):1137−1152.

[9] Ghosh S, Melhem R, Mossé D. Fault-Tolerance through scheduling of aperiodic tasks in hard real-time multiprocessor systems. IEEE Trans. on Parallel and Distributed Systems, 1997,8(3):272−284.

[10] Ripoll I, Crespo A, García-Fornes A. An optimal algorithm for scheduling soft aperiodic tasks in dynamic-priority preemptive systems. IEEE Trans. on Software Engineering, 1996,23(6):388−400.

[11] Kartik S, Murthy SR. Task allocation algorithms for maximizing reliability of distributed computing systems. IEEE Trans. on Computer, 1997,46(6):719−724.

[12] Qin X, Jiang H, Swanson DR. An efficient fault-tolerant scheduling algorithm for real-time tasks with precedence constraints in heterogeneous systems. In: Proc. of the 31st Int'l Conf. on Parallel Processing (ICPP 2002). 2002. 360−368. http://www.cs.nmt.edu/~xqin/pubs/icpp02.pdf

[13] Vaidya NH. A case for two-level recovery schemes. IEEE Trans. on Computer, 1998,47(6):656−666.

[14] Dinatele M, *et al*. Dynamic end-to-end guarantees in distributed real-time systems. In: Proc. of the Real-Time Systems Symp. 1994. 67−73. http://citeseer.ist.psu.edu/cache/papers/cs/529/http:zSzzSzretis.sssup.itzSzpaperszSzrtss1994.pdf/ dinatale94dynamic.pdf

[15] Lauzac S, Melhem R, Mossé D. Adding fault-tolerance to P-fair real-time scheduling. In: Proc. of the Workshop on Embedded Fault-Tolerant Systems. Boston: IEEE Press, 1998. 34−37.

[16] Liu YS, Zhang CF, Zhang T, Zha YB, Huang KD. The analysis of best checkpoint interval of distributed simulation system using Markov chains. Journal of National University of Defense Technology, 2005,27(5):73−78 (in Chinese with English abstract).

[17] Chandy M, Lamport L. Distributed snapshots: Determining global states of distributed systems. ACM Trans. on Computing Systems, 1985,3(1):63−75.

[18] Punnckkat S. Schedulability analysis for fault tolerant teal-time systems [Ph.D. Thesis]. York: University of York, 1997.

附中文参考文献:

[1] 黄柯棣,等.系统仿真技术.长沙:国防科学技术大学出版社,1998.307−326.

[6] 徐志伟,冯百明,李伟.网格计算技术.北京:电子工业出版社,2004.1−19.

[16] 刘云生,张传富,张童,查亚兵,黄柯棣.基于 Markov 链的分布式仿真系统最佳检查点间隔研究.国防科技大学学报,2005,27(5):73−78.

刘云生(1976－),男,山东日照人,博士生,主要研究领域为分布式仿真容错,网格计算.

张传富(1973－),男,博士生,主要研究领域为系统建模与仿真,仿真网格.

张童(1978－),女,博士生,主要研究领域为系统建模与仿真,仿真网格.

查亚兵(1968－),男,教授,博士生导师,主要研究领域为系统建模与仿真,实验分析与仿真应用,仿真网格.