文章编号:1001 – 2486(2005)05 – 0073 – 05

基于 Markov 链的分布式仿真系统最佳检查点间隔研究*

刘云生,张传富,张 童,查亚兵,黄柯棣
(国防科技大学机电工程与自动化学院,湖南 长沙 410073)

摘 要:基于 HLA 的分布式仿真系统作为一类特殊的分布式系统,其容错一般基于回卷恢复实现,在回卷恢复中检查点间隔的不同会对系统性能产生很大的影响。分析了分布式仿真容错与一般分布式系统容错的异同,根据不同仿真进程对仿真结果的重要程度对其进行了分类,定义并利用 Markov 链分析了采用回卷恢复时分布式仿真系统的可用度,得到了系统最大可用度对应的检查点间隔的求解等式,通过一组试验数据验证了该最佳检查点间隔求解等式的正确性。

关键词:分布式仿真;容错;可用度;Markov 链;检查点间隔
中图分类号:TP391.9 文献标识码:A

The Analysis of Best Checkpoint Interval of Distributed Simulation System Using Markov Chains

LIU Yun-sheng, ZHANG Chuan-fu, ZHANG Tong, ZHA Ya-bing, HUANG Ke-di
(College of Mechatronics Engineering and Automation, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: HLA-based simulation system, regarded as a special kind of distributed system, often adopts rollback recovery to realize fault tolerance. Checkpoint interval is an important rollback recovery parameter that will seriously influence system performance. Firstly, we analyze the differences of fault tolerance between HLA – based distributed simulation system and the general distributed system. And then according to the different degrees of the importance of the simulation process to the simulation result, we classify simulation processes into trivial parts and critical parts. Furthermore, the availability of distributed simulation system, which adopts rollback recovery mechanism, has been defined and analyzed through the utilization of Markov chain. As a result, we achieve an equation, by which the checkpoint interval in the best system availability can be figured out. The correctness of this conclusion has also been testified through a set of experimental data.

Key words: distributed simulation; fault tolerance; availability; Markov chains; checkpoint interval

基于 HLA 的分布式仿真系统(HBDSS)作为一类相对特殊的分布式系统,必然要解决其容错问题[1],HBDSS 的容错可基于回卷恢复(rollback recovery)实现。回卷恢复所要解决的核心问题就是如何在系统正常运行时对其执行检查点以获得必要的恢复信息。这里除了要解决全局检查点的一致性问题、in-transit 消息问题[2]外,还必须要确定检查点间隔[2]。因为若检查点间隔过小,则需频繁执行检查点,从而会给系统在无错运行(failure-free)时带来很大的开销;如果检查点间隔太大,则当系统发生故障时,又会造成大量的计算损失。

执行检查点的基本目的是提高系统的可用度,分布式仿真系统可根据系统的类型对检查点间隔的设置进行一定程度的优化,比如文献[3]就提出了一种针对并行离散仿真系统的检查点间隔设置策略,但是这种策略只适用于特定类型的仿真系统,不具有通用性且实施难度较大,比如这种设置策略就不能用于基于时间步长推进的仿真系统。

1 HBDSS 的特点

HBDSS 和普通分布式系统的区别在于 HBDSS 中存在 RTI 这一类中间件系统,它通过两个进程

* 收稿日期:2005 – 04 – 19
基金项目:国家部委基金资助项目(51404010403KG0155)
作者简介:刘云生(1976—),男,博士生。

(RTIexec、fedexec)和一个函数库(libRTI)提供了一系列服务功能来处理联邦运行时的互操作和管理联邦运行,在一定程度上实现了"软总线"功能。

HBDSS中主要有三类仿真进程:RTIexec、fedexec、联邦成员进程。我们将运行这三类进程的处理机称为仿真节点,一般而言,每个仿真节点上会运行一个或数个仿真进程。显然,任一仿真节点出现故障,都将导致运行在该节点上的仿真进程的进程空间损坏,从而导致仿真不能继续。当然,失效仿真节点上运行的进程不同,则引发的后果不同。比如若运行RTIexec、fedexec的节点出现故障,则不能创建新的联邦执行,成员不能正常地加入退出联邦执行及进行交互等;若运行联邦成员进程的节点出现故障,轻则会影响仿真的真实性,重则将使整个系统不能进行正常的仿真时间推进。所以为防止由于某仿真节点故障而使整个仿真重新启动,降低计算损失,必须按照一定检查点间隔来保存所有仿真进程的状态。

HBDSS的检查点协议异于普通分布式系统。在普通分布式系统中,只需要处理由消息传递所导致的进程状态的不一致及in-transit消息问题,而在HBDSS中,不仅要处理上述问题,而且还要保证所保存的成员状态和对应的RTI状态是一致的。

总之,HBDSS中RTI这一特殊组件的存在,使其运行机制异于普通的分布式系统,但是从容错的角度来看,两类系统中降低由节点故障所造成的计算损失的手段是一致的,都需要根据一定的检查点协议定期地保存系统状态,其区别主要在于检查点协议本身的复杂程度不同,而非检查点的执行频率。所以,可以借鉴分布式系统最佳检查点间隔的研究成果来研究HBDSS的最佳检查点间隔。

2 系统模型及假设

仿真系统由 M 个由网络相连的处理机(仿真节点)组成,所有处理机都可以通过网络访问一个中心稳定存储器;全局检查点的一致性由一种协同检查点协议[2]保证;系统中发生的故障类型为失败停(fail-stop),在本文中仅考虑由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,且在任意时刻系统中只能发生一个故障。每个处理机中发生的故障的次数服从泊松过程,处理机中连续两次故障的时间间隔的概率分布函数为 $1-e^{\lambda t}$,处理机间的故障是独立的。系统中有冗余的处理机可用于故障恢复。此外,假定检查点开销[2] C、恢复开销[2] R 都是常量。

由前所述,在HBDSS中,不同仿真节点(进程)的故障对系统的影响不同,而有时为了满足仿真的实时性要求,并没有足够的时间用于回卷。所以,有时即使某些仿真进程由于节点崩溃而停止响应,也要维持仿真系统的继续运行,以得到一个大致可用的结果。基于此,根据故障后果对系统影响程度的不同,将系统中的仿真进程分为两类:

(1)次要仿真进程:该类进程对仿真的正常推进等不起作用。包括一些数据记录成员等。假设该类进程所占仿真节点的数量为 G。

(2)关键仿真进程:是保证仿真的正常运行、真实性、实时性等不可或缺的进程。包括RTIexec、fedexec以及大部分联邦成员进程。

假设运行次要仿真进程的处理机发生故障后,系统不需要回卷,在后续分析中,如不做特殊说明,所分析的进程都指关键仿真进程。

3 仿真系统可用度研究

3.1 仿真系统可用度定义

为准确定义系统的可用度,先对如下两类时间进行区分:

(1)有效时间(useful time):仿真任务启动到任务完成期间进行仿真运算的时间。

(2)无效时间(useless time):仿真任务启动到任务完成期间,除进行仿真运算外的其它时间,包括系统执行检查点的时间、故障恢复时间等。

区分两类时间的标准是系统是否进行仿真运算,而不是系统是否继续运行。比如在某处理机出现故障后进行回卷恢复时,虽然系统继续运行,但是此时系统进行的工作却不是仿真运算,所以此时系统处于无效时间。

假设仿真任务运行期间有效时间的和为 U,无效时间的和为 NU。因有效时间和无效时间不会交叠,这样仿真任务总运行时间为 $STime = U + UN$。U 实际上就是在不采用容错措施且运行过程中无故障发生时,仿真任务的理论运行时间,假设 U 已知;$STime$ 就是在采用容错措施且有故障发生的前提下,完成该仿真任务所需要的实际运行时间。

根据可靠性理论,定义分布式仿真系统的可用度

$$A = U/STime \tag{1}$$

假设检查点间隔为 T,设 $\mu = STtime/T$。这样,仿真任务的运行时间便被分为「μ」个检查点间隔。计算出每个检查点间隔的运行时间 $tInt$ 后,完成该仿真任务所需要的时间就是

$$STime = \sum_{1}^{[\mu]} tInt \tag{2}$$

将其代入(1)式,便可以获得仿真系统的可用性。

3.2 相关工作

分析可用度有三种基本方法:(1)传统的数学统计分析[4]。采用这种方法对分布式系统的开销率(与可用度类似)进行了分析。该方法的缺点是非常复杂,不便于实施。(2)基于 Markov 链的系统可用度分析[5]。当系统的状态有限时,这种分析方法是首选,但是当系统的状态繁多时,该种分析技术则是捉襟见肘。(3)基于 Petri Net 的系统可用度分析。文献[6]利用 Petri Net 对系统的可用度及相关参数进行了评测。实际上 Petri Net 和 Markov 链有其内在的一致性,但 Petri Net 更适合于对大型系统的相关参数进行分析。

在分布式仿真可用度的分析中所涉及到的系统状态有限,所以本文选择基于 Markov 链对系统的可用度进行分析。

3.3 基于 Markov 链的分布式仿真系统可用度分析

在提供了基于检查点的容错机制的前提下,在仿真进程执行一个检查点间隔的过程中,可能的状态有:正常状态 1;出错状态 2;完成状态 3。其状态转换过程如图 1 所示。

在该检查点间隔中,若一切正常,则系统完成该检查点间隔中的仿真运算并转移到状态 3;若发生故障,则系统状态从 1 转移到 2,然后利用保存在全局稳定存储介质上的检查点文件进行故障恢复,恢复完成后系统继续运行以完成该检查点间隔,系统状态最终转移到 3。若在故障恢复时又发生故障,则仍利用原检查点文件进行故障恢复。

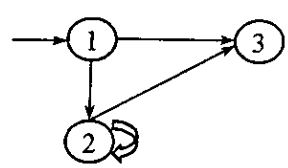

图 1 状态转换图

Fig.1 State transition

从上述分析可见,仿真系统运行过程中一个检查点间隔的平均实际运行时间可以通过该 3 个状态的 Markov 链来求解,将每个检查点间隔的实际运行时间相加就可以得到该仿真任务的总体实际运行时间。

设从状态 X 到 Y 的转移概率为 P_{XY},相应的权重为 W_{XY}。其中 P_{XY} 表示这种转移的可能性,W_{XY} 表示在状态转移到 Y 之前处于 X 的持续时间。接下来将分析每个状态转换中涉及到的转移概率和权重,并求得一个检查点间隔的实际运行时间。

(1)从状态 1 到状态 3

只有当所有的运行关键仿真进程的处理机都没发生故障时,整个系统的状态才会转移到 3。若任一处理机发生故障,所有仿真进程都需回卷;根据故障模型,处理机不发生故障的概率为 $e^{-\lambda(T+C)}$($T+C$)是在无故障时完成一个检查点间隔所需要的时间,则($M-G$)个处理机都不发生故障的概率即转移概率 $P_{13} = \prod_{1}^{(M-G)} e^{-\lambda(T+C)} = e^{[-(M-G)\lambda(T+C)]}$,权重为 $W_{13} = T + C$。

(2)从状态 1 到状态 2

当有处理机发生故障时,系统状态会转移到 2,显然,系统中发生故障的概率 $P_{12} = 1 - P_{13}$。

下面来求解相应的权重。根据假设,发生故障的概率密度为($M-G$)$\lambda e^{-(M-G)\lambda t}$。在已知故障发生的前提下,对应的条件概率密度为($M-G$)$\lambda e^{-(M-G)\lambda t}/(1-e^{-(M-G)\lambda(T+C)})$。对应的权重也就是:$W_{12}$

$$= \int_0^{T+C} t \frac{(M-G)\lambda e^{-(M-G)\lambda t}}{1-e^{-(M-G)\lambda(T+C_L)}} \mathrm{d}t = \frac{1}{(M-G)\lambda} - \frac{T+C}{e^{(M-G)\lambda(T+C)}-1}。$$

(3)从状态2到3

当状态转移到2后,由于需要首先进行故障恢复,则完成该间隔一共所需时间为 $T+C+R$。类似于 P_{13},在恢复期间及随后的检查点间隔内所有($M-G$)台处理机都不发生故障的概率为 $P_{23}=e^{-(M-G)\lambda(T+C+R)}$,相应权重为 $W_{23}=T+C+R$。

(4)从状态2到状态2

如果发生该种状态转移,说明在故障恢复期间又发生故障,所以 $P_{22}=1-P_{23}$。类似于 W_{12},相应的权重为 $W_{22}=\frac{1}{(M-G)\lambda}-\frac{T+C+R}{e^{(M-G)\lambda(T+C+R)}-1}$。

在得到了上述值后,根据文献[7]完成一个检查点间隔所需要的时间就是

$$tInt = P_{13}W_{13} + P_{12}[W_{12} + W_{22}P_{22}/(1-P_{22}) + W_{23}] \tag{3}$$

将相应的值代入(3)式,则可得到完成一个检查点间隔所需要的时间为

$$tInt = \frac{1}{(M-G)\lambda}(e^{(M-G)\lambda(T+C+R)} - e^{(M-G)\lambda R}) \tag{4}$$

将上式代入(2)式,而后再将其带入(1)式,则可得系统可用度为

$$A = \frac{T}{\frac{1}{(M-G)\lambda}(e^{(M-G)\lambda(T+C+R)} - e^{(M-G)\lambda R})} \tag{5}$$

显然,当系统的可用度最大时,对应的检查点间隔就是最佳检查点间隔。由于在上式中只有 T 一个变量,则最佳检查点间隔可以通过求解 $\frac{\partial A}{\partial T}=0$ 获得。

通过对上式求导可得

$$e^{(M-G)\lambda(T+C)}[1-(M-G)\lambda T] = 1 \tag{6}$$

对某个仿真任务的最佳检查点间隔就可以通过求解上式获得。由于 $\frac{\partial^2 A}{\partial T^2}<0$,所以在给定条件下系统可用度有唯一的极大值。

4 数值分析

利用一组假定的试验数据分析、验证不同的检查点间隔(单位为s)对系统可用度的影响,并验证本文关于最佳检查点间隔的结论。

采用的分析数据如表1所示。仿真任务理论运行时间为5040s,检查点开销和恢复开销都为2s,仿真节点的数量为100,次要仿真进程所占处理机数量为10。通过计算,不同检查点间隔、不同 λ 下的可用度如图2所示。

由图2可见,检查点间隔的不同对系统的可用度的影响非常大。其主要原因是当检查点间隔过小时,由于需要频繁执行检查点,而每次执行检查点都会有一定的开销,这会降低系统的可用度;而当检查点间隔过大时,则一旦发生故障,系统会回卷到上一检查点所保存的系统状态,损失的计算量就会很大,同样会减低系统的可用度。此外,存在最佳的检查点间隔,基于该最佳检查点间隔执行检查点,可以使系统获得最大的可用度。λ 对系统的可用度亦有影响。λ 越小,系统的可用度越高,反之,系统的可用度越低。

系统最佳可用度所对应的参数表1所示。

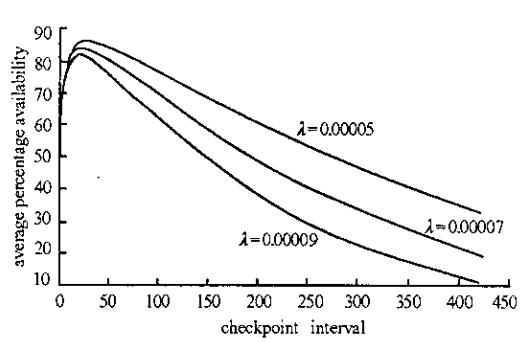

图2 系统的可用度图

Fig.2 Availability of system

表 1 数值分析结果图

Tab.1 Result of numerical analysis

检查点间隔 T	最佳可用度	λ
28s	86.39%	0.00005
23.33s	83.89%	0.00007
21s	81.73%	0.00009

将相应的数值代入式(6),则可得等式近似成立。这从另外一个侧面验证了式(6)的正确性,所以,对于给定仿真任务,其最佳检查点间隔可以通过式(6)得到。

5 结 论

利用 Markov 链对分布式仿真系统的可用度进行了分析,得到了系统最大可用度对应的检查点间隔的求解等式,为分布式仿真系统容错的检查点间隔的选择提供了依据。在分析中假定检查点开销、恢复开销为常量,而在实际的系统中,会随负载的不同而不同,这可能会影响到分析的精度。此外,μ 的向上取整对分析结果的也会有一定影响。

参 考 文 献:

[1] 刘云生,等.分布式仿真系统容错机制研究[J].系统仿真学报,2005,17(2).

[2] Elnozahy M, Alvisi L, Wang Y M, et al. A Survey of Rollback-Recovery Protocols in Message-Passing Systems[R]. Tech. Rep. No. CMU-CS-96-181, Dept. of Computer Science, Carnegie Mellon Univ, ftp://ftp.cs.cmu.edu/user/mootaz/papers/S.ps, 1996.

[3] Fleischmann J, Wilsey P A. Comparative Analysis of Periodic State Saving Techniques in Time Warp Simulators[A]. In Proceedings of the 9th Workshop on Parallel and Distributed Simulation(PADS '95)[C], Lake Placid, NY, USA, June 14-16, 1995, 50-58.

[4] Vaidya N H. A Case for Two-level Distributed Recovery Schemes[A]. In ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems[C], Ottawa, May 1995.

[5] Vaidya N H. A Case for Two-Level Recovery Schemes[J]. IEEE Transactions on Computers, 47(6), 1998.

[6] Park Gyung-Leen, Youn Hee Yong, Choo Hyun-Seung. Optimal Checkpoint Interval Analysis Using Stochastic Petri Net[A]. 2001 Pacific Rim International Symposium on Dependable Computing[C], December 17-19, Seoul, Korea, 2001.

[7] Trivedi K S. Probability and Statistics with Reliablility, Queueing and Computer Science Applications[M]. Prentice Hall, 1988.