

基于同步和异步时间管理的混合时间管理算法

龚爱珍, 艾丽蓉, 王 琼

(西北工业大学 计算机学院, 陕西 西安 710072)

摘 要:基于 HLA 的并行分布式仿真是目前仿真的一种趋势。HLA 的接口规范中定义了六类服务,其中时间管理服务是保证联邦成员之间正确交互的基础。时间管理服务保证了在仿真过程中所发生的事件在逻辑上的正确性,以及所发送的消息在逻辑上的有序性。实现该管理机制有两种方法:保守时间推进算法和乐观时间推进算法。而保守时间推进又可以通过异步时间管理算法或同步时间管理算法来实现,异步和同步算法各有优缺点,这使得它们的运用具有一定的局限性。文中全面分析了异步和同步时间管理算法的优缺点,在此基础上提出了一种将两者结合,进行优缺点互补的混合时间管理算法,以此来提高保守时间推进算法的效率,使其能更好地运用于 RTI 中。并通过实验验证了该混合算法的有效性。

关键词:异步时间管理;同步时间管理;混合时间管理;并行分布式仿真

中图分类号:TP391.9

文献标识码:A

文章编号:1673-629X(2011)07-0032-04

A Hybrid Time Management Algorithm Based on Both Asynchronous and Synchronous Time Management

GONG Ai-zhen, AI Li-rong, WANG Qiong

(Dept. of Computer, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Parallel distributed simulation based on HLA is a trend now. HLA defines six service groups. TM (Time Management) service group ensures correct time advancement of a federation. It guarantees the logical validity of events created or sent in a simulation. There are two ways to realize this TM mechanism: conservative TM and optimistic TM. A conservative TM algorithm can be either asynchronous or synchronous. However, both asynchronous and synchronous algorithms have their own drawbacks and thus cannot be used for all federation scenarios. It proposes a hybrid TM algorithm by combining asynchronous and synchronous algorithms to solve their respective disadvantages. This has greatly enhanced the efficiency of conservative TM, which makes it work better in RTI. Experimental results show the effectiveness of the hybrid algorithms.

Key words: asynchronous TM; synchronous TM; hybrid TM; parallel distributed simulation

0 引 言

为了增强仿真应用间的互操作性和可用性,以及实现跨地域联合仿真,使得分布式仿真成为了必然。HLA 已于 2000 年被 IEEE 确定为 1516 标准,成为国际分布仿真的通用标准。HLA 的接口规范中定义了六类服务^[1-3],其中时间管理服务是保证联邦成员之间正确交互的基础。时间管理服务保证了在仿真过程中所发生的事件在逻辑上的正确性,以及所发送的消息在逻辑上的有序性。实现该管理机制有两种方法:保守时间推进算法和乐观时间推进算法^[3-5]。保守时

间推进机制可通过同步和异步时间管理算法两种方法来实现^[6,7],异步算法在仿真推进是根据联邦成员的 GALT(Greatest Available Logical Time)值做出局部推进决定,而同步算法则是根据联邦的 GALT 做出全局推进决定。这使得异步算法存在“时间蠕变”问题,而同步算法则存在会被某些低频率仿真成员拖延了仿真推进的问题。文中提出了将两者结合的混合算法来解决各自的缺点。并用实验验证混合算法的有效性。

1 传统时间管理算法介绍

1.1 GALT 介绍

GALT 在时间推进服务中起着重要的作用。联邦成员的 GALT 表示该联邦成员的最大安全时间推进值,将来不会再接收到时戳值(time stamp,即 TS)小于该值的 TSO 消息。联邦的 GALT 定义为联邦成员的 GALT 的最小值,它的含义是指在联邦最小的局部时钟(指联邦成员的仿真时间)和它之间建立一个事件

收稿日期:2010-12-07;修回日期:2011-03-15

基金项目:国家自然科学基金(60273087);西北工业大学基础研究基金(W018101);陕西省自然科学基金(SJ08F25)

作者简介:龚爱珍(1986-),女,江西南昌人,硕士研究生,研究方向为人工智能与计算机理论;艾丽蓉,副教授,研究方向为软件工程、信息安全与智能信息处理。

窗,所有联邦成员的所有 TS 小于或等于 GALT 的事件均可以并发的执行。计算公式如下^[1-3]:

$$GALT_i = \begin{cases} \text{Min}(T_i + \text{Lookahead}_i) & \text{如果联邦成员 } F_i \\ & \text{是时间受限的成员, } F_i \text{ 是所有能给 } F_i \\ & \text{发送 TSO 消息的联邦成员} \\ \infty & \text{如果联邦成员 } F_i \text{ 不是时间受限的} \\ & \text{成员} \end{cases}$$

联邦的 $GALT = \text{Min}(GALT_i), i = 1, 2, \dots, n$ 。

在仿真过程中,各联邦成员定时报告本地时间信息,时间管理算法就是根据所报告的时间信息来计算 GALT 值,根据计算的 GALT 结果来决定仿真的推进点。

1.2 异步时间管理算法

该方法中允许每个 LP(Logical Process)——即联邦成员,可以根据各自的 GALT 值作出一个局部的决定,这使得每个 LP 可以独立地、异步地决定安全事件的范围。

假设两两相邻的 LP 之间均有一条单向链路(链路是单向的,但是两个 LP 之间的消息是双向的,因此存在输入/输出两条链路),通过这些链路,LP 之间可以相互发送消息。每一个链路保证发送的消息是按 TS 非递减次序的,这样的话,每个 LP 的 GALT 值则为任何输入链路上的最小未处理消息的 TS。当所有输入链路均不为空时,LP 找出所有输入链路中 TS 最小的消息进行处理,并将 LT 设置为该消息的 TS。

但是当某条输入链路为空时,此时有两种处理方式:一种方式是把该链路上的最后接收的消息的 TS 用于计算 LBTS;很明显这种方式不可用,因为该链路上最后接收的消息的 TS 很明显不是最新的,而与实际最新的 TS 相差多少则取决于该链路上消息发送的频率。第二种方式是该 LP 被阻塞直到有新的消息;这种方式容易造成死锁。例如 LP1 等 LP2, LP2 等 LP3, LP3 等 LP1,这就发生了死锁。解决这种死锁的方法很多,较典型的算法有:空消息法、命令驱动的空消息法、死锁的探测和恢复机制等。其中空消息法应用较广^[8,9]。而在空消息法运用过程中又会产生一个新的问题——“时间蠕变”。

空消息法中^[10,11],LP1 发给 LP2 时戳为 $LT + \text{Lookahead}$ 的空消息,是前者对后者作出的承诺:以后 LP1 发给 LP2 的消息时戳不会小于 $LT + \text{Lookahead}$ 。利用空消息,接收方可以安全处理其它事件。在空消息法中,LP 只能以 LH (Lookahead) 步长推进直到遇到能处理的非空消息为止,而当 LH 比较小的时候就会有“时间蠕变”的现象出现,例如:图 1 所示矩形框中的表示消息。首先,LP1 处理 TS=4 的消息,然后将 LT=4,发

送 TS=4+2 的空消息给 LP2 和 LP3。LP2 在接收到 LP1 空消息后将 LT=6,然后发送 TS=6+2 的空消息给 LP1 和 LP3。LP1 接收到 LP2 发送的空消息后,将 LT=8,然后发送 TS=8+2 的空消息给 LP2 和 LP3...按照此步骤,以后到 LP1 和 LP2 以步长 LH(此例中 LH 为 2)将 LT 推进到 40 时才能处理 TS=40 的那个消息。这个就是“时间蠕变”问题。

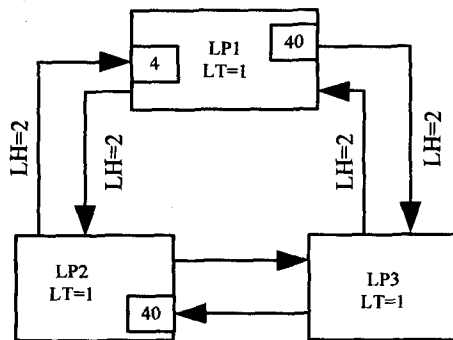


图1 产生“时间蠕变”的仿真

1.3 同步时间管理算法

该方法中,每个 LP 采用执行—阻塞—执行的方式进行推进。基本方式是:首先,每个 LP 一进入阻塞,报告局部事件中最小的未处理 TS+LH。当所有 LP 都进入阻塞时,每个 LP 的 GALT 就可以计算出来,从而算出全局的 GALT 并通知所有的 LP。计算出最新的 GALT 后,LP 可以处理所有 TS 与全局 GALT 相等或者小的事件。当 LP 运行超过了安全科处理事件的范围时,它就会再次进入阻塞阶段,并重复这个过程。此方法中解决了异步时间管理中的死锁问题和“时间蠕变”问题。同样以图 1 为例,LP1 处理完 TS=4 的消息后,报告 TS=40+2,LP2 报告 TS=40+2,LP3 中没有未处理消息,报告 TS=∞,所以全局 GALT=42,这样 LP1 和 LP2 都可以处理 TS=40 的消息了。而不用像异步中以 LH 为步长,一步步推进。

然后,同步时间管理算法中存在至少两个问题:一个是瞬时消息的处理方式;例如 LP1 已经阻塞,但是 LP2 并不知道 LP1 此时的状态,LP2 给处于阻塞状态下的 LP1 仍然发送了信息,然后进入阻塞,这种情况下的消息称为瞬时消息。该情况下对 LP1 的 GALT 值计算是不准确的,只把 LP1 进入阻塞前收到的消息考虑进去了,而没有把瞬时消息考虑进去。瞬时消息的出现明显影响全局 GALT 的计算,Fujimoto 使用缓存阻塞和由 Mattern 提出的运行消息计数器已经解决了这个问题^[9]。另一个问题是:因为全局 GALT 是需要根据所有的 LP 的报告来计算的,如果其中一个 LP 的报告时间频率比较低的话,将严重影响所有的 LP 的时间推进。这是同步时间管理算法中的主要缺点,也是需要解决的问题。

2 混合算法

从以上对异步和同步时间管理算法的描述可以知道:异步时间管理算法是将 LT 推进为新处理事件的 TS,而同步时间管理算法中,是所有 LP 被阻塞后报告下一待处理事件 TS+LH,然后取全局 MIN(TS+LH)作为 LT 的推进值。因此在设计时主要用到时间推进服务是 NMR(Next Message Request)和 NMRA(Nest Message Request Available)^[12]。

混合算法的主要思想是:

1) 将异步时间管理发送的时间信息标识为 AI(asynchronous information), $AI = LT + LH$;同步中的时间信息标识为 SI(synchronous information), $SI = TS(\text{of next message}) + LH$ 。

2) 每个 LP 均发送 AI 和 SI,分别利用 AI 和 SI 来计算 AIGALT 和 SIGALT。AIGALT 的计算和异步中的 GALT 计算方式一样, SIGALT 的计算和同步中的 GALT 计算方式一样。

3) LP 的 $GALT = \max(AIGALT, SIGALT)$ 。

在计算 SIGALT 时,为了保证所有 LP 的 SI 信息是同一轮交互里面的,在此加入了一个 SINUM 来进行识别,每完成一次交互, SINUM 自动加 1。

在混合算法中, GALT 的计算需要同时依赖于 AIGALT 和 SIGALT, AIGALT 和 SIGALT 两者间的互补解决了各自方法中原有的问题。当 LP 要进行时间推进时,对外发送 UI 和 SI 信息,接收到信息以后是否允许其推进,按图 2 的流程图来进行处理。

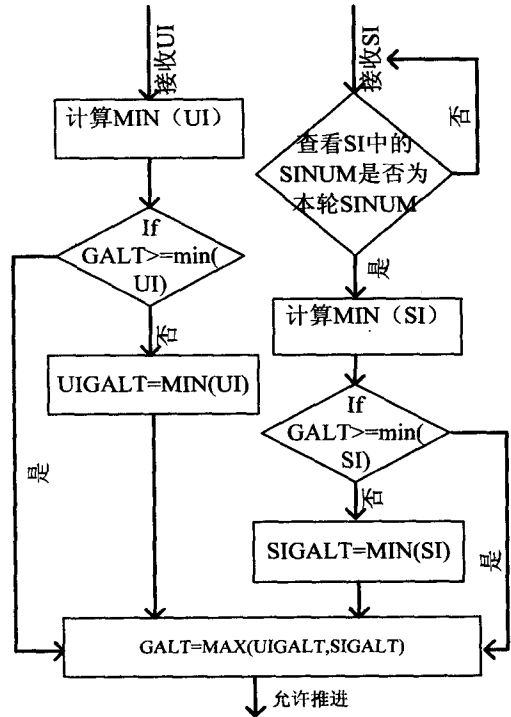


图 2 对 UI 和 SI 的处理流程
混合算法解决了异步时间管理算法中的“时间蠕

变”问题。当 LH 值较小是,只要保证 SI 消息能及时报告,利用混合算法中的第三条计算公式,就可以将 GALT 值推进到下一个待处理事件 TS,和同步时间管理算法效果一样,而避免了以步长蠕变的问题。

混合算法同样解决了同步时间管理算法中低频率报告时间信息的 LP 所带来的影响。对于低频率报告时间信息的 LP,可以默认它的 LH 值比较大,不影响全局的安全推进,这样的话直接利用其它 LP 的 SI 信息来计算出不够精确的 SIGALT。但是由于利用 AI 信息可以精确计算出 AIGALT 值,而根据 $GALT = \max(AIGALT, SIGALT)$ 计算出来的效果和使用异步时间管理算法一样。

3 实验及结果

以上针对混合算法在解决异步时间管理算法和同步时间算法缺点的有效性方面进行了理论分析,同时用实例验证了该理论的正确性及可行性。在 windows 平台下用 MAKRTI 开发了两个实例进行测试。

在实例 1 中创建两个联邦成员,两个联邦成员有相同的 LH,两者之间以时间增量 10 来相互发送 NMR,当接收到 TAG 后,仿真推进并再次发送 NMR,不断重复操作。实验中不断改变 LH 值的大小,来测试三种算法在不同 LH 值下进行一次交互所需要的平均时间。表 1 为实验结果。

表 1 LH 值对仿真结果的影响

LH	同步(ms)	混合(ms)	异步(ms)
0.2	45	48	340
0.5	42	48	155
0.8	40	50	140
1.0	40	44	80
3.0	40	44	47
5.0	40	42	42

从表 1 中可以看出,当 LH 值比较小的时候,由于“时间蠕变”的影响,异步时间控制算法的运行效率明显比同步和混合差很多。而在混合算法中,由于 SI 的作用,使得其效果和同步时间管理算法类似,非常稳定。

在实例 2 中,主要测试低频率报告时间信息的联邦成员对时间推进的影响效果。仍然创建了两个联邦成员。成员 1 报告频率较高,时间增量为 10, LH 为 10。每次交互后演示 200ms 来进行仿真处理。成员 2 报告频率较低,时间增量为 100, LH 为 100,每次交互后延时一段时间再发送时间信息。本测试中,测试了成员 2 的延时从 200ms ~ 1200ms 变化中,三种算法每次交互所需要的平均时间。表 2 为实验结果。

表2 低频率报告成员对仿真运行的影响

低频率成员的报 告频率(ms)	同步(ms)	混合(ms)	异步(ms)
200	220	220	220
400	238	218	218
600	253	219	221
800	268	219	223
1000	290	220	220
1200	308	223	218

从表2中可以看到,随着低频率报告成员频率的降低,同步时间管理算法的效率严重受影响,而混合和异步时间管理算法的运行则比较稳定。

4 结束语

在分布式仿真中,传统的同步时间管理算法和异步时间管理算法都有各自的缺点存在,使其自身的使用受到了限制。而文中的混合算法则很好地解决了这个问题。考虑到实际仿真中的不确定性,LH值较小的情况和低频率报告成员出现的情况都有可能出现,因此,该混合算法在实际仿真运用中有较实用的价值。

参考文献:

- [1] Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society. IEEE std 1516-2000, high level architecture (HLA)-framework and rules[S]. USA: The Institute of Electrical and Electronics Engineers, Inc, 2000.
- [2] Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society. IEEE std 1516, 1-2000, high level architecture (HLA) - federate interface specification [S]. USA: The Institute of Electrical and Electronics Engineers, Inc, 2001.
- [3] 周彦,戴剑伟. HLA 仿真程序设计[M]. 北京:电子工业出版社, 2002.
- [4] 赵沁平,周忠,吕芳. 大规模分布节点的仿真时间同步算法[J]. 中国科学, 2008(8): 1139-1156.
- [5] 华伟亮,孙少斌. HLA/RTI 中时间管理机制及实现[J]. 系统仿真技术, 2009(3): 208-213.
- [6] Fujimoto R M, Weatherly R M. HLA Time Management and DIS[C]//14th Workshop on Standards for the Interoperability of Distributed Simulation. [s.l.]: [s.n.], 1996.
- [7] Fujimoto R M, Weatherly R M. Time Management in the DoD High Level Architecture[C]//10th Workshop on Parallel and Distributed Simulation. [s.l.]: [s.n.], 1996.
- [8] 李革,刘宝宏,张耀程. 并行于分布仿真系统[M]. 北京:电子工业出版社, 2010.
- [9] 蔡吉森,汪厚祥. 保守 PDES 中时间管理问题研究[J]. 计算机工程与设计, 2007(14): 3536-3538.
- [10] Fujimoto R M. Parallel discrete event simulation[J]. Communication of the ACM, 1999, 33(10): 30-35.
- [11] 欧阳伶俐,宋星,卿杜症,等. HLA 时间管理与 PDES 仿真算法研究[J]. 系统仿真学报, 2000(3): 237-240.
- [12] 李琳娜. RTI 时间管理 NERA 分析[J]. 煤炭技术, 2009, 28(12): 183-185.

(上接第31页)

跟它后面结合的字,跟它前后结合的字,独立的字四种类别。分类是在一定的语境下利用互信息的数学统计方法完成的,其方法简单、实用、正确率高。另外,采用t-测试的方法初步解决了占歧义86%以上的交叉歧义问题。当然文中也存在很多不足,比如对歧义的处理、未登录词鉴别的能力还有待提高,仍需进一步的研究,以期提高分词的正确率。

参考文献:

- [1] 罗桂琼,费洪晓,戴弋. 基于反序词典的中文分词技术研究[J]. 计算机技术与发展, 2008, 18(1): 81-83.
- [2] 赵姝,张燕平,张媛,等. 基于交叉覆盖算法的入侵检测[J]. 计算机工程与应用, 2005, 41(1): 141-143.
- [3] 刘慧婷,倪志伟,李建洋,等. 基于交叉覆盖算法的时间序列模式匹配[J]. 计算机应用, 2007, 27(2): 425-427.
- [4] 刘政怡,龚建成,吴建国. 基于交叉覆盖算法的中文分词[J]. 计算机工程, 2006, 32(19): 183-184.
- [5] 吴涛,张铃. 基于 Rough 集的交叉覆盖算法设计[J]. 计算机科学, 2003, 30(3): 57-59.
- [6] 李新福,赵杰,梁巍. 基于互信息的宋史语料库词表的提取[J]. 河北大学学报(自然科学版), 2006, 26(5): 557-560.
- [7] 陈平,刘晓霞,李亚军. 基于字典和统计的分词方法[J]. 计算机工程与应用, 2008, 44(10): 144-146.
- [8] 岳涛. 汉语自动分词技术的最新发展及其在信息检索中的应用[J]. 情报技术, 2005(4): 55-60.
- [9] 费洪晓,康松林,朱小娟,等. 基于词频统计的中文分词的研究[J]. 计算机工程与应用, 2005(7): 67-70.
- [10] Huang C, Yo T, Simon P, et al. Realistic and Robust Model for Chinese Word Segmentation[C]//Proceedings of ROCLIN G. [s.l.]: [s.n.], 2008.
- [11] Huang C, Simon P, Hsieh S, et al. Rethinking Chinese Word Segmentation: Tokenization, Character Classification, or Wordbreak Identification[C]//Proceedings of the Association of Computational Linguistics Annual Meeting (ACL). [s.l.]: [s.n.], 2007.
- [12] Sang Nong, Zhang Tianxu, Shi Weiqiang. Characteristics of contrast and application for small target detection[C]//SPIE Data and Processing of Small Target. [s.l.]: [s.n.], 1998: 123-131.