基于协调组的 RTI时间管理方法

吕 芳 周 忠 魏 晟 吴 威

(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)

摘 要: 针对目前运行支撑环境(RTI, Run2Time Infrastructure)时间管理方法中存在的时间推进效率低及网络带宽占用量大的问题,提出基于协调组的 RTI时间管理方法并将其应用于 BH RTI 2. 3时间管理服务. 将仿真系统中节点划分为不同的协调组并设置协调者,通过分组局部协调逐步达到全局的时间同步. 实验结果表明,该方法在仿真节点时间推进速度及网络带宽占用量方面优于同等条件下的其他 RTI时间管理方法.

关 键 词: 高层体系结构; 运行支撑环境; 时间管理; 协调组; 时戳下限

中图分类号: TP 391

文献标识码: A 文 章 编 号: 100125965(2008) 0921084204

Time management of RTI based on negotiation group

L Fang Zhou Zhong Wei Sheng Wu Wei

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

Abstract Aimed at the problems existed in the time management of run2time infrastructure(RTI) at the present time, a method of time management of RTI based on negotiation group was brought forward and applied in the time management service of BH RTI 2. 3, which was conducted to solve the problems such as low time advance speed and high network bandwidth engross. This method divided the node in the simulation system in2 to different coordination groups and set the coordinator, which attained global time synchronization through the coordination of local synchronization. The experiment outcome indicates this method preponderates over other methods in both time advance speed and network bandwidth engross on even ground.

Key words: HLA(high level architecture); RTI(run2time infrastructure); time management; negotia2 tion group; LBTS(lower bound time stamp)

高层体系结构(HLA, High Level Architec2 ture)是分布式仿真的国际标准, 运行支撑环境(RTI, Run2Time Infrastructure)提供了 HLA接口规范中定义的各项服务,其中时间管理主要解决分布式仿真中的事件一致性问题[1- 2].

时间管理也称时间同步,管理分布式仿真中各类消息的发送和接收,保证仿真过程的逻辑正确性. 目前,国内外一些 RTI软件已经实现了时间管理服务,国外有 DMSO RTI系列[3], Pitch公司的 pRTI[4], MAK公司的 MAK RTI,国内有北京航空航天大学的 BH RTI,国防科技大学的 KD2RTI, StarLink[5]等.

这些 RTI实现的时间管理方法可以分为基于单一节点协调和基于分布多节点协调 2类. 前者通过设置一个协调节点来负责仿真中盟员时间信息的维护与计算并处理盟员的时间推进,如 pRTI, MAK RTI及 DMSO RTI随着仿真规模的增大,单一的协调节点成为系统瓶颈,存在时间推进速度低、网络带宽占用量大的问题. 后者将仿真过程中时间信息的计算与维护完全分布于参与仿真的各个节点上,节点之间通过发送消息来进行同步协调,如 BH RTI 2. 2, Starlink等. 它避免了单一节点成为系统瓶颈的问题,但为了维护各个节点数据的一致性,存在协调报文占用大量网络带宽的问题.

本文提出了基于协调组的 RTI时间管理方

收稿日期: 2007209213

基金项目: 国家 863计划资助项目(2006AA01Z331); 国家自然科学基金资助项目(60603084)

作者简介: 吕 芳(1983-), 女, 内蒙古乌兰浩特人, 硕士生, lvfang@ vrlab buaa edu cn

法,实现了 BH RTI 2 3的时间管理,将仿真过程中的时间信息的维护与计算根据职能分配到不同的节点上.

1　HLA时间管理相关概念

1) 前瞻值　L,设仿真节点逻辑时间为 T,且该节点预测产生下一事件的时间不会早于 T + L,则称 L为该节点的/前瞻值0.

2) 时戳下限　LBTS (Lower Bound Time Stamp),节点的最大安全时间推进值. HLA 标准文档中[3]定义 LBTS为: $LBTS_i = \min\{T_j + L_j\}$, i X j,其中 T_j和 L_j分别为节点 j的当前仿真时间和前瞻值.

3) 输出时间　OPT(OutputTime),也称为有效逻辑时间 (ELT, Effectire Logic Time),由 Frederick 引入. 指仿真节点能够发送的事件时戳的下界,计算公式为: OPT= T + L, T为仿真时间.

2　基于协调组的时间管理模型

基于协调组的时间管理模型将分布式系统中的各个分布节点分作不同的协调组并进行角色分配,每组中选出 1个协调节点. 仿真过程中,节点以组为单位,通过协调节点进行时间同步协调,最终达到全局的时间同步. 同时,根据各种节点对报文的可靠性要求的不同,优先保证关键节点的报文传输可靠.

各个节点根据职能的不同可分为 3种角色:仿真节点、局部协调者和全局协调者. 其结构见图 1. 可见,仿真节点与局部协调者,局部协调者与全局协调者之间均为多对一的关系. 虚线框 a所示是一个局部协调组,它由若干个仿真节点和一个局部协调者组成. 虚线框 b所示是全局协调组,由若干个局部协调者和一个全局协调者组成.

每个仿真节点都具有输出时间 OPT及时戳下限 LBTS,局部协调者和全局协调拥有自身的最小时间和次小时间. 仿真节点的 LBTS计算过程如下:设系统中存在 m 个协调节点,每个协调节点负责 n_m个仿真节点的时间同步协调. 将它们分作 m 组,对任一节点 N 拥有仿真时间 t和输出时间 R 每个协调组可得集合 8,因此得

$$8_1 = \{R_{11}, R_{12}, R_{13}, \ldots, R_{1n_1}\}$$
$$8_2 = \{R_{21}, R_{22}, R_{23}, \ldots, R_{2n_2}\}$$
$$\vdots$$
$$8_m = \{R_{m1}, R_{m2}, R_{m3}, \ldots, R_{mn_m}\}$$

对节点集合 8_k, 1[k[m,通过各个节点的承诺时间值 R可以得出组内节点承诺时间值 R的最小值和次小值,计算公式表示为

$$\begin{cases} K_k = R_{ki} \quad R_{ki} \text{ I } 8_k \\ D_k = R_{ki} \quad R_{ki} \text{ I } (8_k - K_j) \end{cases} \quad P\ j(j \text{ X } i \text{ y } R_{ki} [R_{kj})$$

进而得到集合 5 = {K_1, D_1, K_2, D_2, K_3, D_3, , , K_m, D_m}. 令 $L_k = K_k$, $L_{k+m} = D_k$,可得

$$5 = \{L_1, L_2, L_3, L_4, \ldots, L_{2m-1}, L_{2m}\}$$

对于集合 5,可确定其最小值 W和次小值 N,它们的计算公式为

$$\begin{cases} W = L_k \quad L_k \text{ I } 5 \\ E = L_k \quad L_k \text{ I } (5 - W) \end{cases} \quad P\ j(j \text{ X } k \text{ y } L_k [L_j)$$

通过全局计算的结果,得到承诺时间最小的节点集合 5_{min}以及变量值 (K_k, D_k)最小的协调节点集合 C_{min}:

$$5_{min} = \{N_i \mid R_i = W\}$$
$$C_{min} = \{C_k \mid K_k = W \text{C } D_k = W\}$$

最终确定的变量值 W和 N就是要求的各个节点的可以推进的时间的上限,对于承诺时间值最小的节点,其上限为 N,其他节点的上限为 W,因此可以得到各个节点的 LBTS取值为

$$LBTS_i = \begin{cases} W & N_i \notin 5_{min} \\ N & N_i \text{ I } 5_{min} \end{cases}$$

在协调过程中,各个分布的节点之间通过消息进行通讯. 这些信息可以分作 2类:如图 2,一类是由被协调节点发送至协调节点的包含节点输出时间 R以及局部协调者计算得到的 K, D的信息,称之为 R信息;另一类是由协调节点发送至被协调节点的包含了用于计算 LBTS的 W值和 N值的信息,简称为 W2N信息.

通过对模型中各个节点特征的研究,得出结论:局部协调者和全局协调者要求 R信息必须可靠,最小的 (W, N)所在的局部协调者及输出时间最小的仿真节点对 W2N信息要求可靠,其他的节点对 W2N信息可靠性没有要求.

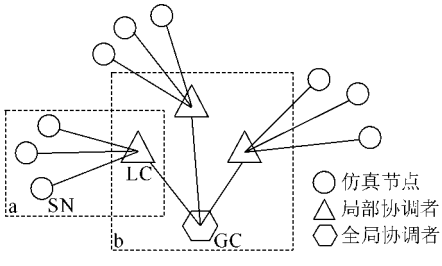


图 1　协调组划分示意图

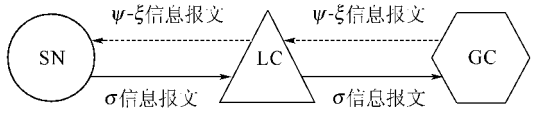


图 2　节点之间信息传递图

3 基于协调组的 BH RTI时间管理

BH RTI是北京航空航天大学虚拟现实技术与系统国家重点实验室自主开发的 RTI软件, 同时支持 HLA 1.3和 IEEE 1516标准. BH RTI 2.2实现了一种分布式的时间管理服务, 具有较高的推进效率, 但存在时间管理控制报文占用大量带宽的问题[6]. BH RTI 2.3采用基于协调组的 RTI时间管理方法, 主要方案如下.

3.1 协调组模型映射

将基于协调组的时间管理模型应用于 BH RTI的时间管理服务, 系统总体结构如图 3.其中, 盟员对应于模型中的仿真节点, RTI服务器中 RTI_TM对应于局部协调者, 时间管理服务器对应于全局协调者. 图 3中矩形虚线框部分所示局部协调组, 可同时处理 1 024个盟员节点的同步; 椭圆形虚线框部分所示为全局协调组, 可同时处理 512个局部协调组的时间同步. 仿真盟员, RTI_TM, 时间管理服务器的具体结构及功能如图 4所示. [1] 仿真盟员节点. 盟员节点负责具体仿真逻辑的执行, 它维护了用于仿真的接受序(RO, Receive Order)队列、时戳序(TSO, Time Stamp Order)队列、节点的当前逻辑时间、输出时间以及 LBTS值. 其中, RO队列中事件按照接收序存放, TSO队列中事件按照其时戳由小到大顺序存放.

° RTI_TM. RTI_TM作为局部协调者, 负责协调其管理范围内的各个仿真盟员的时间推进. 它维护了其协调范围内的各个盟员的输出时间列表 OptArray, LBTS列表 LbtsArray以及最小时间仿真盟员的集合 MinFedSet, 同时它还具有属性值局部最小时间 LMT和局部次小时间 LST. » 时间管理服务器. 时间管理服务器作为全局协调者, 负责协调整个仿真系统中的所有 RTI时间管理模块的时间推进. 它维护了所有 RTI的局部最小时间值和局部次小时间值的列表 RtiTimeArray, 最小 RTI节点集合 MinRtiSet, 同时具有属性值全局最小时间 GMT和全局次小时间 GST.

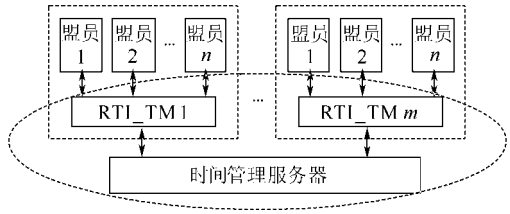


图 3 基于协调组的 BH RTI时间管理结构

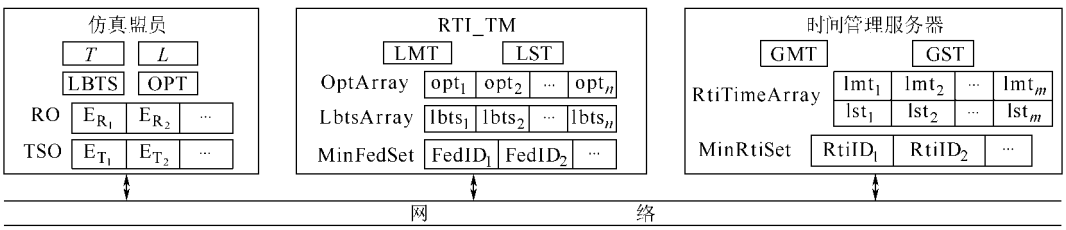


图 4 各角色节点内部结构图

3.2 节点的时间推进算法

节点的时间推进是时间管理算法的关键部分, 仿真盟员节点、RTI_TM以及时间管理服务器三者进行同步协调的过程如图 5所示. 仿真盟员提出时间推进请求, 计算输出时间, 向 RTI服务器节点发送 R报文, 通知 RTI自己当前的输出时间; RTI服务器接收来自仿真盟员的时间信息, 更新本地保存的各个盟员的输出时间值, 判断更新的输出时间是否对 RTI服务器节点的局部最小、次小时间值(LMT, LST)产生了影响, 如果没有影响则不做处理, 继续等待下一报文, 否则计算(LMT, LST), 发送 R报文给时间管理服务器, 通知新的局部时间值; 时间管理服务器接收来自 RTI服务器的局部时间信息, 更新本地保存的 RTI服务器的时间特征值, 判断更新的(LMT, LST)是否对时间管理服务器节点的特征值(GMT, GST)产生了影响, 如果没有则不做处理, 继续等待后续报文, 否则根据系统中的所有 RTI的特征值计算得出全局特征值(GMT, GST); 时间管理服务器计算得出新的全局特征值(GMT, GST)之后, 向所有 RTI节点组播 W2N报文, 通知它们最新的全局时间信息, 然后确定集合 MinRtiSet, 对任一属于该 RTI服务器的节点, 发送 TCP的 W2N报文; 对于任一 RTI服务器节点, 如果属于 MinRtiSet, 那么它可能会收到组播和 TCP的相同报文. 如果收到了重复报文, 则丢弃, 然后对收到的协调信息进行处理. 根据全局特征值(GMTS, GSTS)计算其管理范围内各个仿真盟员的 LBTS值, 向其协调范围内的仿真盟员发送组播报文, 通知他们最新的承诺逻辑时间值(PLT, Promise Logic Time)值, 确定集合 MinFedSet, 对于任一个属于 MinFedSet集合的仿真盟员, 发送 TCP的 W2N报文.

仿真盟员收到来自 RTI的协调报文, 更新自

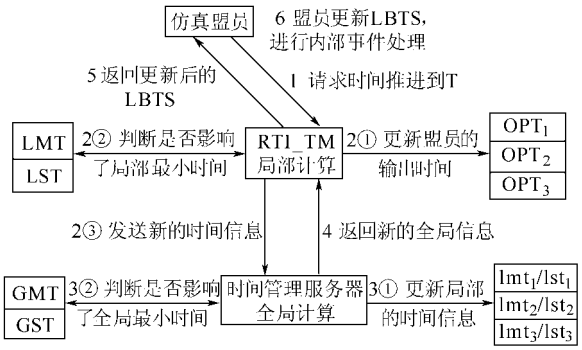


图 5　节点时间推进过程

身 LBTS值. 如果请求推进的时间大于 LBTS, 则继续等待, 否则, 进行时间推进并处理节点队列中的事件, 然后继续下一次推进.

4　实验与结果分析

通过对 HLA /RTI时间管理的研究, 设计了 RTI时间管理的测试指标, 将应用本文时间管理方法的 BH RTI 2. 3与其他同类 RTI软件进行时间管理的性能对比测试. 实验的环境如表 1.

表 1　实验环境表

项目	配置
网络	100M 以太网, 华为 Quidway S3928
主机	Pentium 4 3. 40GHz 内存 1GB, windows XP, 11台
软件	RTI软件: BH RTI 2 3, BH RTI 2. 2, DMSO RTI 1 3NGv6, MAK RTI 3. 0, pRTI 1516LE 测试程序: DMSO Benchmark 网络带宽测试软件: IRIS v4 07 1

测试指标包括时间推进速度、空载网络带宽占用量和时间同步盟员规模. 时间推进速度是指仿真节点在单位时间内间推进次数, 单位为次 /s 空载网络带宽占用量是指仿真运行过程中, 在没有事件报文的情况下, 单位时间内 RTI时间推进所占用的网络带宽量, 单位为 MB. 时间同步盟员规模是指盟员进行时间同步推进时, 盟员数量与系统推进速度的关系. 其中, 时间推进速度和盟员规模测试采用美国国防部支持的 DMSO Bench2 mark测试平台, 带宽占用采用网络带宽测试软件: IRIS v4. 07. 1.

图 6为仿真主机数量为 1Q 盟员数量分别为 10, 20, 30, 50, 100时的时间推进速度的测试结果. 仿真盟员的 L为 1, 进步长为 1Q 实验结果表明, 随着节点规模的增大, 各 RTI时间推进速度降低. MAK RTI不支持规模大于 50个盟员的时间管理, DMSO RTI不支持规模大于 100个盟员的时间管理. BH RTI 2 3的时间推进速度最高, 且下降的幅度也较小.

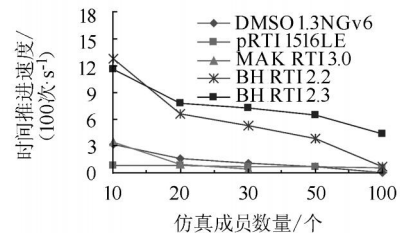


图 6　时间推进速度试验结果统计图

图 7是在图 6的实验环境下, 对 RTI时间管理服务运行时的网络带宽占用量进行测试的实验结果统计. 由于所用的 pRTI为限制版, 一次仿真运行只能调用服务接口 100次[4], 无法对其进行准确测量, 因此该组实验中没有包含 pRTI, DMSO RTI和 MAK RTI均不支持盟员数量为 100的仿真规模. 从图中可以得出 BH RTI 2 3一直保持较低的空载网络带宽占用.

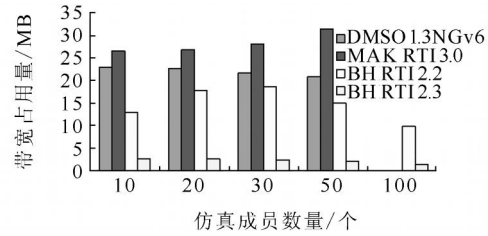


图 7　空载网络带宽占用量实验结果统计图

5　结 束 语

本文对 RTI中时间管理服务进行研究, 提出一种基于协调组的 RTI时间管理方法, 描述了方法的基本思想并介绍了该方法在 BH RTI 2 3中的具体实现. 对本文的方法进行实验, 结果表明该方法在节点时间推进速度及网络带宽占用量方面优于同等条件下的其他 RTI, 与现有 RTI相比可以支持更大规模的仿真.

参考文献 (References)

[1] Fujimoto R M. Parallel and distributed simulation systems[C] // Proceedings of the 2001 Winter Simulation Conference. Piscat2 away, N J: IEEE, 2001: 147- 157

[2] Fujimoto R M. Time management in the high level architecture [J]. Simulation, 1996, 71(6): 60- 67

[3] DMSO. RTI 1. 32next generation programmer. s guide version5 [EB/OL]. [2006- 02- 15]. http: //www. dmso. mil

[4] Fujimoto R M. Exploiting temporal uncertainty in parallel and distributed simulation [C] //13th Workshop on PADS. Los Alamitos: IEEE, 2004: 395- 401

[5] Zhang Yan, Zhou Zhong, Wu Wei. A hierarchical time manage2 ment mechanism for HLA2based distributed virtual environment [J]. Journal of Computational Information Systems, 2006, 1(2): 7- 15

(责任编辑: 吴　琼)