

文章编号:1001-9081(2006)04-0989-03

分布式实时仿真系统高精度时间同步技术研究

薛芳侠¹, 闫了了¹, 谢虹¹, 姚益平², 刘海业¹

(1. 63892 部队, 河南 洛阳 471003; 2. 国防科技大学 计算机学院, 湖南 长沙 410073)

(xuefangxia@tom.com)

摘要:在采用 PC 机和 Windows 操作系统的局域网上用软硬件相结合的方式实现高精度、高效率的时间同步是提高分布式实时仿真系统置信度、降低系统硬件成本的关键。给出了一种局域网时间对齐(LTS)算法和一种基于 PC 机自身资源的时钟构造方法。LTS 算法在局域网环境下比目前广泛使用的 NTP(网络时间协议)算法有更好的时间对齐效果,新时钟在性能上远远优于 Windows 系统时钟。最后设计了一个完整的时间同步方案,该方案达到了高精度、高效率时间同步和降低硬件成本的目的。

关键词:分布式;实时仿真系统;时间同步

中图分类号: TP393 **文献标识码:** A

Study on technique of high time coherence used in real-time distributed interactive simulation

XUE Fang-xia¹, YAN Liao-liao¹, XIE Hong¹, YAO Yi-ping², LIU Hai-ye¹

(1. 63892 Crops of PLA, Luogang HeNan 471003, China;

2. College of Computer, National University of Defense Technology, Changsha Hunan 410073, China)

Abstract: In LAN which use PC and Windows operating system, accurate and effective time coherence using the pattern that software cooperate with hardware is one of the keys for real-time distributed interactive simulation system to improve fidelity and decrease cost. A brief time synchronization algorithms in LAN (LTS algorithms) was bring forward, and a new clock was designed based on resource of PC. In LAN, LTS algorithms has better time adjust effect. The new clock is better than the system clock of Windows operating system in performance. At last, a integrated scheme of time coherence was designed, and the purpose to get accurate and effective time coherence and to decrease cost was realized.

Key words: distributed; real-time simulation; time coherence

0 引言

在分布式实时仿真系统中,联网仿真的计算机之间的时间同步必须达到一定的精度,否则可能引起仿真逻辑关系的混乱,严重时会导致仿真试验彻底失败。分布式实时仿真系统一般由一个局域网或分布在异地的几个局域网组成,普遍采用的时间同步模式是分层混合时间同步模式,即在每个局域网内设置一台带有时统设备的计算机作为时间服务器,局域网内其他计算机(客户机)采用软件算法和时间服务器取得同步。

目前,PC 机因为较高的性价比,Windows 2000/NT 操作系统也因为其强大的功能和众多的商品软件和软件开发工具的支持,已成为构建分布式仿真系统的首选硬件设备和操作系统,但 PC 机在 Windows 2000/NT 操作系统下的时间分辨率较低,因此 PC 机系统时钟对有较高时间同步要求的分布式交互仿真系统来讲是无法使用的,同时因为 Windows 2000/NT 操作系统的实时性较差,扩大了传统网络时间对齐算法的对齐误差。此外,对于分布式实时仿真系统,选择时间对齐算法时必须考虑算法的效率,即每次时间对齐所占用的 CPU 时间要尽量少,以减少时间对齐对实时仿真运行的冲击,而传统

的时间对齐算法在效率上不能很好的满足这一要求。

本文针对以上问题,对使用 PC 机和 Windows 2000/NT 操作系统的分布式实时仿真系统的时间同步问题展开研究,重点讨论局域网时间对齐算法和客户端时钟设计。

1 分布式实时仿真系统网络时间对齐算法

本节介绍两种时间对齐算法,一种是适用于广域网的 NTP(Network Time Protocol,网络时间协议)时间对齐算法,一种是适用于局域网的 LTS(LAN Time Synchronization,局域网时间对齐)算法。

1.1 NTP 网络时间对齐算法

在目前已广泛使用的网络时间对齐算法中,NTP 被公认是一种高精度时间对齐算法。在 NTP 支持的三种时间传送模式(多播模式、客户/服务器模式和对称模式)中客户/服务器模式的时间对齐精度最高,时间对齐过程如图 1 所示。

T_1 : 客户方发送查询请求时间(以客户方时间系统为参照)

T_2 : 服务器收到查询请求时间(以服务器时间系统为参照)

T_3 : 服务器发送回复时间信息包时间(以服务器时间系

收稿日期:2005-10-28;修订日期:2006-01-13

作者简介:薛芳侠(1972-),男,河南南阳人,工程师,硕士,主要研究方向:分布交互式仿真技术;闫了了(1966-),男,河南人,高工,主要研究方向:指挥自动化技术、仿真试验技术;谢虹(1965-),男,湖北人,研究员,博士,主要研究方向:电子战仿真;姚益平(1963-),男,教授,博士,主要研究方向:软件工程、分布式仿真和虚拟现实;刘海业(1980-),男,河北人,工程师,主要研究方向:计算机网络攻防技术。

统为参照)

T_4 :客户方收到时间信息包时间(以客户方时间系统为参照)

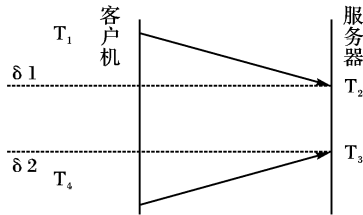


图1 NTP算法(客户/服务器模式)时间对齐过程

δ_1 :请求信息从客户机进程传送到时间服务器进程所消耗的时间

δ_2 :回复信息从时间服务器进程传送到客户机进程所消耗的时间

设服务器标准时间和客户机本地时间之差为 θ , θ 即为客户机所需调整的时间:

$$\begin{cases} T_2 = T_1 + \theta + \delta_1 \\ T_4 = T_3 - \theta + \delta_2 \\ \delta = \delta_1 + \delta_2 \end{cases}$$

解上述方程可得:

$$\begin{aligned} \theta &= ((T_2 + T_3) - (T_1 + T_4) - (\delta_1 - \delta_2))/2 \\ &= ((T_2 - T_1) - (T_4 - T_3))/2 - (\delta_1 - \delta_2)/2 \end{aligned}$$

因为 δ_1 和 δ_2 在时间对齐时无法准确测量,NTP算法假设请求和回复在网上传播的时间相同,即 $\delta_1 = \delta_2$,则:

$$\begin{cases} \theta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \\ \delta = (T_2 - T_1) + (T_4 - T_3) \end{cases}$$

据此,客户方即可通过 T_1, T_2, T_3, T_4 计算出时差 θ 去调整本地时钟。

NTP算法的第一项误差为网络传输时间波动引起的误差,大小为 $(\delta_1 - \delta_2)/2$,即对时请求从客户进程传送到服务器进程的传输时间和回复信息从服务器进程传送到客户进程的传输时间之差的一半,主要受网络负载、服务器及客户机CPU负载的影响。对于高精度的时间对齐,NTP算法一是采用数据过滤机制减少此项误差,即每次对时进行 n (一般取 $n = 8$)次查询,得到 n 组 (δ, θ) 值,采用最小值算法(取 δ 最小的 θ 值)或中值算法(取 θ 的平均值)来获取较为精确的 θ 值,二是在软件实现时要求赋予客户机时间对齐线程较高的优先级(一般应设为实时优先级),以减小客户机CPU负载对该项误差的影响。

NTP算法的第二项误差是一项隐含误差,即客户机时钟分辨率造成的误差。算法中 T_1 和 T_4 是以客户方时间系统为参照的,且在 θ 的计算中是相加关系,因此客户机时钟分辨率造成的误差大小可达到客户机时钟分辨率的大小,也就是说假设客户机时钟分辨率为1ms的话,则客户机时钟分辨率造成的误差可达1ms之多。

1.2 局域网时间对齐算法

局域网时间对齐(LAN Time Synchronization, LTS)算法是作者在文献[3]提出的算法基础上,运用NTP算法的思想,针对分布式实时仿真系统设计的一种网络时间对齐算法。因为仅适用于局域网,因此将其称为局域网时间对齐算法。这种算法实质上是在局域网环境下对NTP算法的一种改进,其对时过程如图2所示。

T_1 :客户方发送查询请求时间(以客户方时间系统为参照)

T_2 :服务器收到查询请求时间(以服务器时间系统为参照)

T_3 :客户方收到时间信息包时间(以客户方时间系统为参照)

δ_1 :请求信息从客户机进程传送到时间服务器进程所消耗的时间

δ_2 :回复信息从时间服务器进程传送到客户机进程所消耗的时间

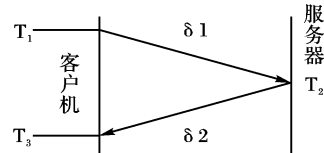


图2 LTS算法时间对齐算法示意图

设服务器标准时间和客户机本地时间之差为 θ , θ 即为客户机所需调整的时间,则有: $T_2 = \theta + \delta_1 + T_1$

$$T_3 = T_2 + \delta_2 - \theta$$

解上述方程组可得: $\theta = T_2 - (T_3 + T_1)/2 + (\delta_2 - \delta_1)/2 \approx T_2 - (T_3 + T_1)/2$

LTS算法也要求赋予客户机时间对齐线程较高的优先级。和NTP算法的两项误差相对应,LTS算法第一项误差为网络传输时间波动引起的误差,大小也是 $(\delta_1 - \delta_2)/2$,但减少此项误差的方法与NTP算法不同,LTS算法减少该项误差的方法是依据对局域网数据传输时间的分析测试结果设计的。大量测试表明,在不同的网络负载下,网络传输时间的最小值基本维持不变,而最大值变化剧烈。

根据网络传输时间的变化规律, $\delta_2 + \delta_1$ 越是接近2倍的网络最小传输时间, $\delta_1 - \delta_2$ 也会越小,LTS算法减小 $(\delta_1 - \delta_2)/2$ 的方法就是以 $\delta_1 + \delta_2$ 来估算 $(\delta_1 - \delta_2)/2$ 的范围,而 $\delta_1 + \delta_2$,即 T_1 与 T_3 时刻之间时间段的长度可以用PC机性能计数器精确测量(测量精确度高达 $0.3\mu s$)。假定某局域网最小网传输时间是 Y ,要求 $(\delta_1 - \delta_2)/2$ 不大于 X ,根据数学推导结果,只要 $\delta_2 + \delta_1 < 2X + 2Y$,即可保证 $(\delta_1 - \delta_2)/2$ 不大于 X 。这样就可以将 $2X + 2Y$ 作为门限值,当 $\delta_2 + \delta_1$ 小于门限值时即认为是一次成功的时间对齐,可以用时间对齐所获得数据调整客户机时钟,当 $\delta_2 + \delta_1$ 大于等于门限值时认为是一次不成功的时间对齐,所获得数据舍弃不用。根据在100M局域网上对网络传输时间的测试数据推算,当将门限值设定为0.4ms时可保证LTS算法第一项误差 $(\delta_1 - \delta_2)/2$ 不大于0.1ms,经实际测试验证,一般不超过2次对时请求,即可保证有一次成功的时间对齐。在实际工程使用中,可以以一定的时间间隔发送对时请求,直至 $\delta_2 + \delta_1$ 小于门限值,这种时间对齐方式对实时仿真系统非常有利,因为客户机时间对齐线程的优先级较高,如果连续进行时间对齐的话,势必会影响到实时仿真运行的实时性。

NTP算法的第二项误差是由客户机时钟分辨率造成的误差,而LTS算法可以绕开客户机时钟的分辨率问题。对于LTS算法,当判定时间对齐成功时,可以在对齐过程结束时刻(T_3 时刻)调整客户机时钟。

在 T_3 时刻将客户机时钟调整为:

$$\begin{aligned} T_3 + \theta &= T_3 + T_2 - (T_3 + T_1)/2 \\ &= T_2 + (T_3 - T_1)/2 \end{aligned}$$

$$= T_2 + (\delta_2 + \delta_1)/2$$

T_2 是从时间服务器获取的时间(以服务器时间系统为参照), $\delta_2 + \delta_1$ 可以用 PC 机性能计数器进行精确测量。这样就绕开了客户机时钟分辨率对时间对齐精度的影响,使时间对齐精度变得与客户机时钟分辨率无关。

从上述对 NTP 算法和 LTS 算法的分析比较来看,在使用 PC 机和 Windows 操作系统的局域网环境下,LTS 算法可以达到更高的时间对齐精度,且效率更高,更适合于分布式实时仿真系统。

2 基于 PC 机性能计数器的高性能时钟设计

实际测试表明在 Windows2000/NT 操作系统下,PC 机系统时钟时间分辨率只有 15 ~ 16ms,时间漂移率约为 60ms/h。如果在分布式实时仿真系统中使用系统时钟,不管时间对齐有多精确,计算机之间的时间同步精度都不可能超过 16ms,即不可能超过系统时钟的分辨率。因此,要做到高精度的时间同步必须为 PC 机构建新的高性能(高时间分辨率、低时间漂移率)时钟。

在 PC 机上有一个和 CPU 集成在一起的被称为性能计数器的计数器,其计数频率(计数器计数值每秒的增加数)约为 3.6MHz,并且 Windows 操作系统也提供了读取性能计数器计数值的接口函数:QueryPerformanceCounter,因此如果利用 PC 机性能计数器构造新的时钟,时间分辨率可高达 0.3 μ s(性能计数器的计数周期)。但性能计数器记录的是系统启动以来的计数累计值,因此初始时间需要利用时间对齐算法,通过网络从时间服务器获取。本文构造的基于性能计数器的新时钟的时间计算模型如下:

首先设计三个变量:

StartTime 初始时间。对新时钟进行时间对齐时从时间服务器获取的“标准时间”,即 LTS 算法中的“ $T_2 + (\delta_2 + \delta_1)/2$ ”,其误差取决于时间对齐算法精确度。

StartTick 初始性能计数器计数值。即 StartTime 时刻性能计数器计数值。

freque 函数计算使用的性能计数器频率值。

读取时间信息的全局时间函数:

GefTime()

```

{
    通过 QueryPerformanceCounter 函数获取性能计数器当前计数值 EndTick
    当前时间
    Time = StartTime + (EndTick - StartTick) / Freque; (1)
}
    
```

设时间对齐时刻(StartTime 时刻)性能计数器频率为 $freque(0)$,任一时刻 x 性能计数器频率为 $freque(x)$,则对新时钟的时间计算公式(1)进行变形推导后可得:

$$Time = StartTime + t + \frac{freque(0) - freque}{freque}t + \int_0^t \frac{freque(x) - freque(0)}{freque} dt \quad (2)$$

t 为时间对齐后至取时时经历的真实时间,从式(2)可知,新时钟时间误差有三项:第一项是 StartTime 带来的误差,即时间对齐误差;第二项是由新时钟时间计算使用频率值 $freque$ 和时间对齐时刻性能计数器的真实频率 $freque(0)$ 的偏差造成的;第三项误差是由性能计数器频率的随机漂移引起的。第二项和第三项误差取决于新时钟自身的性能。

对于性能计数器频率,Windows 操作系统提供了一个 API 函数 QueryPerformencFrequency, Microsoft MSDN 对该函数的解释是可以该函数获取性能计数器的频率,但从理论上分析其返回值不可能是一个准确的值,因为性能计数器频率实质上是单位时间内性能计数器计数值的增加量,要准确的测量其频率需要准确的时间,而单台 PC 机是无法获取准确时间的。经过在同一型号多台 PC 机上的测试发现,QueryPerformencFrequency 取得的是一个固定值,因此作者认为 QueryPerformencFrequency 函数获得的只是性能计数器的标称值,该标称值与用时统设备测得的频率值的偏差达 80 ~ 200Hz,这一偏差造成的时钟漂移率将达 80 ~ 200ms/h。因此在工程应用中要获得高精度的时间同步,必须实测客户机性能计数器的频率值。

在客户/时间服务器模式的局域网中,要测量客户机性能计数器的频率,需要通过网络,利用时间服务器的标准时间进行测量。对分布式实时仿真系统来讲,在仿真开始前,时间服务器、客户机和网络都处于空闲状态,在这种情况下,网络传输时间具有很强的对等性,即数据包从时间服务器进程传输到客户机某进程的时间和数据包从客户机进程传输到时间服务器进程的时间近似相等。利用这种特性,作者设计了一种在局域网环境下测试客户机性能计数器频率的算法,简称网络测频算法,测频工作应在仿真试验开始前进行,其时序如图 3 所示。

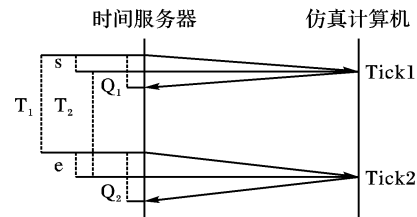


图 3 网络测频算法时序图

测频开始时,时间服务器向客户机发送一个测频开始信号,客户机接收到时间服务器测频开始信号后立即读取本机性能计数器计数值 $Tick_1$,并将该值作为反馈数据发送给时间服务器,时间服务器接收到仿真计算机应答信号后计算测频开始信号往返延时 Q_1 ,等待一段时间(T_1)后,时间服务器向客户机发送测频结束信号,客户机接收到测频结束信号后立即读取本机当前性能计数器计数值 $Tick_2$,并作为反馈数据发送给时间服务器,时间服务器接收到应答信号后计算测频结束信号往返延时 Q_2 ,客户机频率计算的公式如下:

$$freque = (Tick_2 - Tick_1) / (T_1 - (Q_2 - Q_1) / 2)$$

在 T_1 取 1000s 的情况下,大量测试表明,这种网络测频算法的测频精确度达 0.3Hz。

对于引起第三项误差的性能计数器频率的随机漂移问题,作者也在多台 PC 机上进行了大量的测试,测试结果表明性能计数器频率非常稳定,漂移率都在 0.04Hz/h 左右,按照累积 24h 计算,新时钟频率漂移最大不超过 0.96 Hz,因此其时间漂移率最大值不超过 0.96ms/h。

总之,新时钟和系统时钟相比,其优越性体现在两个方面:一是时间分辨率,新时钟时间分辨率为 0.3 μ s,系统时钟时间分辨率为 16ms;二是时间漂移率,系统时钟为 60ms/h,新时钟最大值不超过 0.96ms/h。

(2)对数据进行预处理有利于神经网络的学习^[10]。由于该模型的输入数据范围已经为 $[0,1]$,因此,可以对它进行标准化处理。输出数据范围在 $[-3,3]$ (区分度在 $[0,2]$),可以进行归一化处理。

(3)为寻找网络的最佳宽度,可以用同一批验证数据来检验网络的拟合误差。假设其网络验证误差与隐层神经元宽度间的关系近似一凸函数,可采用黄金分割法来寻找该网络的最佳宽度^[11]。

(4)Perrone 和 Cooper 证明^[12],将神经网络集成运用于回归估计时,如果采用简单平均,且各网络的误差期望为互相独立的随机变量时,则集成的泛化误差为各网络泛化误差平均值的 $1/S$,其中 S 为集成中子网的数目。由于本模型类似于回归估计,所以可以训练多个 GRNN 网络。测试时,网络的最后输出取各网络输出的累加平均。

本研究中,训练样本的 M,N 分别为250,500;验证样本的 M,N 分别为80,100,测试样本的 M 固定为50, N 分别取30、50、80、100、170。Monte Carlo 模拟次数 K 和神经网络集成个数 S 均取30。表1列举了每次测试的均方根误差 RMSE。

表1 相同被试数不同项目数的 RMSE 值

测试项目数	难度 RMSE	区分度 RMSE
30	0.2560	0.1576
50	0.2200	0.1564
80	0.2058	0.1785
100	0.2061	0.1770
170	0.1989	0.1868

6 结语

介绍了一种新的基于 GRNN 的 IRT 项目参数估计建模方法,并结合实际问题对模型进行了多方面的改进,力图利用 GRNN 来学习 CTT 参数统计值和 IRT 参数值之间的隐含关系。实验结果表明:(1)运用 GRNN 进行 IRT 项目参数估计是可行的;(2)在小样本情况下,与现有的 IRT 参数估计软件

相比,该方法的参数估计误差更小^[13];(3)与文献[2]的结果相比仍然存在差距,导致差距的原因除了研究的对象和问题不同外,另一方面,也可能是因为计算机模拟生成的样本质量还不能完全代表实际样本空间,这是值得进一步研究的问题。

参考文献:

- [1] 漆书青,戴海崎,丁树良.现代教育与心理测量学原理[M].南昌:江西教育出版社,1998.313-347,124-177.
- [2] 余嘉元.基于联结主义的连续记分 IRT 模型的项目参数和考生能力估计[J].心理学报,2002,34(5):522-528.
- [3] 谭云兰,丁树良,辛锐铭.基于 IRT 模型的 BP 神经网络估计[J].计算机工程与应用,2004,40(17):56-57.
- [4] 余嘉元.项目反应理论及其应用[M].南京:江苏教育出版社,1992.62-157.
- [5] 董长虹. Matlab 神经网络与应用[M].北京:国防工业出版社,2005.124-126.
- [6] 闻新,周露,李翔,等. Matlab 神经网络仿真与应用[M].北京:科学出版社,2003.285-289.
- [7] LIM RG, DRASGOW F. Evaluation of Two Methods for Estimating Item Response Theory Parameters When Assessing Differential Item Functioning[J]. Journal of Applied Psychology, 1990, 75(2): 164-174.
- [8] 朱德全,宋乃庆.现代教育统计与测评技术[M].重庆:西南师范大学出版社,1998.227-255.
- [9] 高祖新,尹勤.实用统计计算[M].南京:南京大学出版社,1996.130-140.
- [10] 谢庆生,尹健,罗延科.机械工程中的神经网络方法[M].北京:机械工业出版社,2003.88-92.
- [11] 卢险峰.最优化方法应用基础[M].上海:同济大学出版社,2003.67-72.
- [12] PERRONE M, COOPER L. When networks disagree: Ensemble method for neural networks, Artificial Neural Networks for Speech and Vision [M]. New York: Chapman & Hall, 1993. 126-142.
- [13] 谭云兰,丁树良,辛锐铭.基于 IRT 模型的 BP 神经网络降维法参数估计及其应用[J].江西师范大学学报(自然科学版),2004,28(6):485-488.

(上接第 991 页)

3 分布式实时仿真系统高精度时间同步方案

本文设计的分布式实时仿真系统高精度时间同步方案为分层混合时间同步方案。在分布式实时仿真系统的每个局域网内设置一台带有 GPS 时统设备的 PC 机作为时间服务器,局域网时间服务器之间通过 GPS 时统取得同步;局域网内其他计算机通过 LTS 算法与时间服务器取得同步,客户机采用本文设计的新时钟为仿真进程提供时间。客户机性能计数器频率在仿真开始前,利用本文设计的网络测频算法进行测量。图4是仅在仿真开始前进行一次时间对齐,客户机时间与 GPS 标准时间偏差的连续测试图(横轴为时间轴,单位为 s;纵轴为时间偏差轴,单位为 ms)。

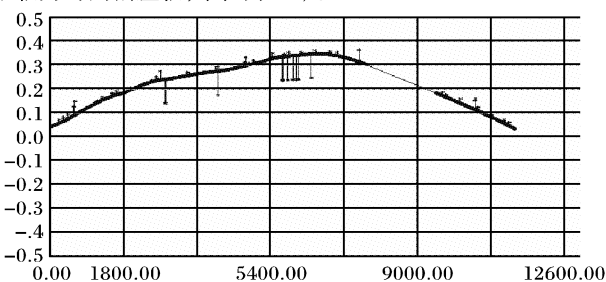


图4 客户机时间偏差曲线

从测试结果看,本方案只在仿真开始前进行一次时间对齐即可保证在3个小时内客户机时间与标准时间的偏差不超过0.4ms。我们在C3I仿真试验系统的时间同步软件开发中将LTS算法的时间对齐间隔设置为30min,连续24h的测试表明时间同步误差最大值不超过0.4ms。

4 结语

本文设计的新时钟无论从时间分辨率方面还是从时间飘移率方面来讲,其优越性都是PC机系统时钟所无法比拟的。本文给出的LTS算法在局域网环境下具有比NTP算法更好的时间对齐效果,并且降低了对网络资源的占用。总之,本文设计的分层混合时间同步方案时间同步精度高,资源占用少,成本低,在保证时间同步精度的前提下,可以将时间对齐对实时仿真运行的冲击降到很低的水平,是分布式实时仿真系统理想的时间同步方案。

参考文献:

- [1] 薛芳侠. C3I 仿真试验系统时间同步技术研究[D]. 国防科技大学, 2004.
- [2] 孙娜,熊伟,丁宇征.时钟同步的研究与应用[J].计算机工程与应用,2003,39(27):177-179.
- [3] 黄四牛,熊新平. DIS 中时钟同步问题研究[J]. 系统仿真学报, 2000, 12(5): 9-10, 17.