

# 局域网环境下的时钟同步与保持

李学军, 谢剑薇, 于凤坤

(装备指挥技术学院 信息装备系, 北京 101416)

**摘要:**针对局域网环境下系统仿真的时钟同步需求,提出了一种时钟同步与保持方法。首先建立客户端与授时服务器的网络连接,通过多次网络通信,得到客户端与授时服务器时间差的不等式组,通过解不等式组求出时间差的有效范围,这样有效地提高了时钟同步的精度。然后通过两个校准时间参考点对系统的脉冲频率进行校准,使得已同步的时钟得以保持,从而避免了在仿真应用中频繁地进行同步,节省了时间开销。实验结果表明,这种方法的同步误差在 $50\mu\text{s}$ 左右,且可以在较长时间内保持。

**关键词:**时钟同步;时钟保持;脉冲频率;时间参考点

**中图分类号:**TP393

**文献标识码:**A

**文章编号:**1673-629X(2009)04-0227-04

## Clock Synchronization and Retaining in Local Area Network

LI Xue-jun, XIE Jian-wei, YU Feng-kun

(Department of Information and Equipment, Academy of Equipment

Command and Technology, Beijing 101416, China)

**Abstract:** A clock synchronized algorithm based on the local area network is proposed. Firstly, establish connection between client and time server, and send time stamps between them. From the time stamps, can get the inequalities about the time difference between the client and the time server, and calculate the probable field from the inequalities. Experiments show that the field is very little, so the synchronization accuracy is approved. The clock frequency is adjusted by testing two clock reference points. This lead to the synchronized clock is retained, and a lot of synchronization operation is avoided, so the time cost is saved. Experiments show that synchronization tolerance of the algorithm is about  $50\mu\text{s}$ , it can be retained in a long time.

**Key words:** clock synchronization; clock retaining; clock frequency; clock reference point

### 0 引言

时钟同步(Clock Synchronization)是系统仿真的关键技术之一,目前仍然是一个比较活跃的研究领域<sup>[1]</sup>。时钟同步方法大致分为硬件同步和软件同步两类。在硬件同步方面,比较典型的方法有四时钟系统、交互连接策略和多级同步算法等<sup>[2~4]</sup>。硬件同步的优点是同步精度高,可达 ns 级,一般用于对同步精度要求很高的场合。硬件同步的成本相对较高,因此在许多仿真应用中采用软件同步方法。软件同步法一般利用标准的网络通信,通过发送包含时间戳的同步消息包,经过同步算法的处理来实现同步,具有成本低、伸缩性好等优点。根据对网络延迟的不同理解,软件同步算法分为概率同步算法、确定性同步算法和统计同步算法三

类。概率同步算法<sup>[5]</sup>假设网络延迟为一个随机变量,且不知道该随机变量的分布,其缺点是在不能确知网络延迟的分布时,不能确保在一轮同步中的效率。确定性算法<sup>[6]</sup>假设系统在网络通讯延迟存在上限,它们保证任意两个时钟之间的差值在一个确定的范围之内。概率同步和确定性算法都要求通过某种其它途径事先获得网络中的消息延迟的一些先验信息。统计算法<sup>[7]</sup>是基于对通讯延迟的分布情况,进行同步算法的设计,它只能以很高的概率保证单次同步精度。美国特拉华大学 Mills 教授提出了网络时钟同步协议 NTP (Network Time Protocol)<sup>[8]</sup>。NTP 使用层次式时间分布模型,所能取得的准确度依赖于本地时钟硬件的精确度和对设备以及进程延迟的严格控制。在此基础上,王世华等人提出了 ETSA(End-to-end time synchronization)算法<sup>[9]</sup>,能够较准确完成时钟同步任务,同时不存在同步失败现象,其同步精度能够限定在毫秒级。

软件同步方法由于同步精度相对较低,同步过程

收稿日期:2008-08-03

基金项目:国家 863 项目(2002AA716133)

作者简介:李学军(1967-),男,教授,博士,CCF 高级会员,研究方向为计算机应用。

要占用 CPU 时间和网络带宽等缺点,其应用范围受到了一定程度的限制。文中提出了一种局域网络环境下的软件同步和时钟保持方法,同步精度可达  $50\mu\text{s}$  左右。在提高同步精度的同时,采用了时间保持策略,不需要频繁进行同步,节省了 CPU 时间开销和网络带宽。

## 1 算法的基本思想

计算机软件系统的时钟由时钟中断来维持,每隔一定时间发一次中断信号,系统的时间计数器增 1。Windows 系统时钟中断的间隔一般为  $15\sim 50\text{ms}$  之间,这一间隔决定了系统时间的刻度,系统时间的精度不高于这一刻度。Windows 系统的 `GetCurrentTime()` 函数用来获取系统的当前时间,连续用 `GetCurrentTime()` 函数读取时间,会发现许多次函数调用的返回值相同,然后突然跳到下一个时间刻度。因此用 `GetCurrentTime()` 函数所读取的时间精度不优于  $15\text{ms}$ 。

Windows 系统可以设定时间中断,通过 `OnTimer()` 函数来触发需要在指定时间执行的操作,但这一方法也受系统时间刻度的影响。

微机系统另有一套刻度更高的计时机制,采用主板上的晶振脉冲发生器和计数器来计时,但由于初始时间和脉冲频率不准确,仍然不能得到准确的系统时间。文中通过软件方法,来校准系统的初始时间和脉冲频率,从而可以提供更为精确的系统时间。为了便于阐述算法,先引入几个基本概念:

(1) 本地系统时间:指的是本地计算机时钟所指示的时间,可以用 Windows API 函数 `GetCurrentTime()` 读取。该时间可能与统一的协调世界时间(UTC, Universal Time Coordinate)有较大差距。

(2) 仿真标准时间:仿真系统中所采用的时间标准,一般采用 UTC,也可以自定义仿真的起始时间和快慢。

(3) 脉冲频率和时间计数值:计算机主板上有一个晶振脉冲发生器,以很短的时间间隔均匀地发出脉冲信号,称该振脉冲发生器每秒钟发出脉冲的次数为脉冲频率。系统中还有一个计数器,从计算机开机时开始计数,每收到一个脉冲信号计数器的值增 1,称任一时刻该计数器的值为当时的时间计数值。时间计数值反映了从开机到当时的时间长短。

(4) 原始脉冲频率和校准脉冲频率:不同计算机的脉冲频率可能有很大差别,Windows 系统下可以用 `QueryPerformanceFrequency()` 函数获取计算机的脉冲频率,但用该函数获取的脉冲频率一般是不准确的,误差大小与晶振材料特性、环境温度、计算机老化程度等

许多因素有关。文中将通过软件方法对误差进行校正,为便于区别,称用 Windows 函数读取的脉冲频率为原始脉冲频率,通过软件校正后的脉冲频率为校准脉冲频率。

(5) 时间参考点( $C, T$ ):指的是系统中的某一时刻,该时刻的时间计数值为  $C$ ,时间为  $T$ 。这个时间  $T$  可以是仿真标准时间,也可以是本地系统时间或自定义时间。若客户端采用本地系统时间作为  $T$ ,则相应的时间参考点与服务器不同步,可通过文中算法与服务器进行同步,将时间  $T$  校正为与服务器同步的时间  $T'$ ,这时时间参考点( $C, T'$ )称为校准时间参考点。

(6) 本地时间:给定一个时间参考点之后,则可以通过该参考点、脉冲频率和时间计数值计算出当时的时间,称这一时间为本地时间。时间参考点是本地时间计时的起点,设时间参考点为( $C_0, T_0$ ),脉冲频率为  $f$ ,计数值为  $C$ ,则本地时间  $t$  的计算公式为:

$$t = T_0 + (C - C_0)/f \quad (1)$$

本地时间的精度取决于时间参考点和脉冲频率的精度。本地时间的刻度为脉冲频率的倒数,例如文中实验所用计算机的频率为  $2810050000$ ,因此时间刻度为  $0.36\text{ns}$ 。

实现时钟同步的基本思想是,先指定一台授时服务器,设该服务器的时间为仿真标准时间。系统中的其它计算机通过多次网络通信,获取多个时间点的服务器仿真标准时间,将它们与客户端的本地时间进行对照,求出客户端与服务器端的时间差,从而获得客户端的校准时间参考点,对本时间进行校正。然后通过两个校准时间参考点,计算客户端的校准脉冲频率。依据校准时间参考点、校准脉冲频率,即可根据任一时刻客户端的时间计数器值,计算出该时刻客户端的校准时间,从而达到时钟同步的目的。

## 2 时钟同步

互联网上有许多时钟同步软件,采用的是单次网络通信,或多次通信取平均值的方式。由于网络通信时间延迟的随机误差较大,且分布没有规律,因此同步误差较大。文中采用多次通信方式,用解多个不等式的方法求最优解,相当于取多次网络通信中延时最小的一次作为计算依据,其结果比单次通信更可信。

同步原理如图 1 所示。 $P_i$  是客户端向服务器请求时间同步的时间点, $Q_i$  是服务器收到请求后发送仿真标准时间的时间点,同步算法如下:

(1) 设置同步通信次数  $N$  的初值,一般可取  $200\sim 1000$ ;

(2) 需要同步的客户端建立与授时服务器的网络

连接(使用基于连接的通信方式时网络延时相对较小);

(3) 客户端读取本地计算机的脉冲频率  $f$ ;

(4) 客户端读取当前本地时间  $T_0$  和当前计数值  $C_0$ , 构成一个时间参考点  $(C_0, T_0)$ ;

(5)  $i = 0$ ;

(6) 客户端读取当时的计数值  $C$ , 按式(1) 计算本地时间存入  $P[i]$ , 并向授时服务器发出时钟同步请求;

(7) 授时服务器收到请求后立即读取当时的仿真标准时间, 并发送给客户端;

(8) 客户端收到授时服务器的仿真标准时间后保存到  $Q[i]$ ;

(9)  $i = i + 1$ ;

(10) 若  $i < N$ , 则转(4);

(11) 依据  $P[i]$  和  $Q[i]$  计算客户端与服务器的时间差。

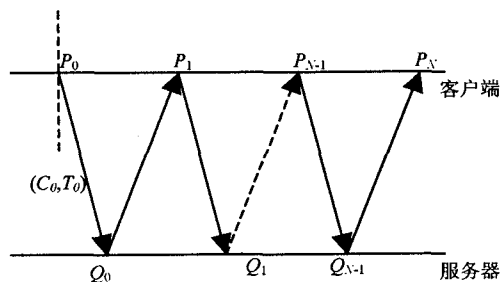


图1 时钟同步原理图

设客户端的时钟比服务器慢  $T$  秒, 则与客户端本地时间  $P[i]$  对应的仿真标准时间为  $P[i] + T$ , 由于网络事件是严格按时间顺序进行的, 则可得到以下一组不等式:

$$P[i] + T < Q[i], (i = 0, 1, \dots, N-1) \quad (2)$$

$$P[i] + T > Q[i-1], (i = 1, 2, \dots, N-1) \quad (3)$$

由式(2)得:  $T < Q[i] - P[i]$

即:  $T < \text{Min}(Q[i] - P[i])$

由式(3)得:  $T > Q[i-1] - P[i]$

即:  $T > \text{Max}(Q[i-1] - P[i])$

所以:  $\text{Max}(Q[i-1] - P[i]) < T < \text{Min}(Q[i] - P[i])$

设  $T_{\min} = \text{Max}(Q[i-1] - P[i])$

$T_{\max} = \text{Min}(Q[i] - P[i])$

则:  $T \in (T_{\min}, T_{\max})$

由于上式中的  $P[i]$ ,  $Q[i]$  已通过实验得到, 因此容易求出  $T_{\min}$ ,  $T_{\max}$ , 则可取其平均值为客户端与服务器的时间差  $T$ , 即:

$$T = (T_{\min} + T_{\max})/2$$

这时的同步误差不超过  $(T_{\max} - T_{\min})/2$ 。

通过在局域网上进行多次测试, 发现  $(T_{\min}, T_{\max})$  的范围很小, 一般为  $100\mu\text{s}$  左右, 以其平均值作为校正值, 则校正后的时间误差为  $50\mu\text{s}$  左右。

在求出客户端与服务器的时间差  $T$  之后, 就可以求得客户端的一个校准时间参考点:  $(C_0, T_0')$ , 其中

$$T_0' = T_0 + T$$

这样以后可用这一校准时间参考点计算本地时间, 计算公式为:

$$t = T_0' + (C - C_0)/f \quad (4)$$

### 3 时钟保持

采取以上时钟同步措施后, 在一定的时间范围内可根据公式(4)计算客户端较精确的本地时间, 从而实现时钟同步。然而计算机系统的脉冲频率误差较大, 公式(4)中用到了客户端的脉冲频率  $f$ , 因此经过一段时间的运行之后, 时间漂移会逐步加大, 从而超出允许的误差范围。为了实现客户端长时间与授时服务器同步, 采取了对客户端的脉冲频率进行校准的方法, 使得本地时间的误差保持相对稳定, 不会随着时间的推移而不断扩大。

具体方法是, 先测得客户端的一个校准时间参考点  $(C_0, T_0')$ , 让系统运行一段时间之后, 再测得另一个校准时间参考点  $(C_1, T_1')$ , 可以由这两个参考点计算出本地计算机的实际频率为:

$$f' = (C_1 - C_0)/(T_1' - T_0')$$

得到校准脉冲频率之后, 本地时间的计算公式可由公式(4)改为公式(5):

$$t' = T_0' + (C - C_0)/f' \quad (5)$$

由于  $T_0'$ ,  $T_1'$  本身仍存在一定误差, 因此时间间隔  $(T_1' - T_0')$  也会存在误差, 这一误差的大小是稳定的, 不大于它们同步误差的 2 倍, 因此增加这两个参考点的时间间隔可以减小相对误差, 从而提高校准脉冲频率的精度。  $T_0'$ ,  $T_1'$  的时间间隔越长, 校准脉冲频率越精确, 一般可取 2~8 小时。按这一方法校准之后, 可以保证在这一时间间隔里的漂移不大于同步允许误差。

### 4 测试结果

笔者使用 Visual C++ 语言实现了文中所述的时钟同步和保持算法, 并应用于某国防预研项目中, 解决了该项目中仿真系统的时钟同步问题。表 1 是对实验

室三台计算机采用该算法的时钟同步情况,测试时与服务器的通信次数为 1000 次。从表中可以看出,本算法的同步精度可以达到  $50\mu\text{s}$  左右。从表 1 还可以看出,在没有校准脉冲频率时,时间漂移较大,一般为几十毫秒,校准脉冲频率之后,时间漂移缩小到几十微秒。

表 1 三台计算机的同步测试结果(时间单位: $\mu\text{s}$ )

	PC1	PC2	PC3
原始脉冲频率	2810050000	802930000	2992580000
同步精度	52.2	54.8	45.5
3 小时漂移	26712	-47260	20131
校准脉冲频率	2810056950	802926486	2992585578
校准后 3 小时漂移	38.5	47.8	-32.3

运用本算法编制的同步软件,还应用于课题组网络应用程序调试中。运用此软件同步网络应用程序的各计算机之后,程序开发人员可以清晰地看到不同计算机之间各种网络请求和响应的时序,从而快速定位程序中的错误。

## 5 结束语

提出了一种局域网环境下的时钟同步与保持算法。通过解不等式组的方法,有效提高了同步精度。通过测试两个校准时间参考点,校准了脉冲频率,使已同步的时钟得以保持,从而避免了频繁地进行同步,节省了时间开销。实验结果表明,这种方法的同步误差

在  $50\mu\text{s}$  左右,且可以在较长时间内保持。本方法是系统仿真应用经济实用的时钟同步方案。

## 参考文献:

- [1] Huang Jiung-yao, Tung Ming-Chih, Wang Kuei Min, et al. Smart Time Management -- the unified time synchronization interface for the distributed simulation[J]. Computer Standards & Interfaces, 2005(27):149-161.
- [2] Farrugia E, Simon R. An efficient and secure protocol for sensor network time synchronization[J]. Journal of System and Software, 2006(79):147-162.
- [3] 赵卫华,杨孟飞,袁 艺. 时钟同步方法研究综述[J]. 控制工程, 2005(5,6):11-19.
- [4] 孙 娜,熊 伟,丁字征. 分布式网络系统中时钟同步的实现[J]. 计算机工程, 2003, 29(14):136-138.
- [5] 李明国,宋海娜,庄钊文. 基于概率同步算法的计算机外时钟同步系统设计与仿真[J]. 计算机仿真, 2002, 19(3):95-97.
- [6] 李明国,宋海娜. 计算机时钟同步技术研究[J]. 系统仿真学报, 2002, 14(4):477-480.
- [7] 宁 伟,张 波,王树梅,等. 基于分布式系统实现时钟同步的统计平均算法[J]. 计算机工程, 2005, 31(5):47-49.
- [8] 傅 磊,戴冠中. 一类网络化控制系统的时延分析及时钟同步方法[J]. 计算机应用研究, 2007, 24(7):264-269.
- [9] 王世华,胥布工,刘云辉. 网络实时系统端对端的时钟同步研究及算法实现[J]. 信息与控制, 2007, 36(3):352-356.

(上接第 226 页)

控制两个摄像机的基于双目立体视觉的鞋楦数字化系统,论述了其组成、工作原理和实现的关键技术。在实现过程中,结合脚部模型的特点,利用交互式方法能够获得反映脚型特征的关键数据后,采用基于视差范围约束和动态搜索的图像配准算法进行图像匹配,然后得到三维空间的坐标数据,并且通过数据拟合技术进行拟合。实验结果表明,三维重建得到了比较好的重建效果,达到了实用的要求。

## 参考文献:

- [1] 郑嫦娥,张 辉,叶佩青,等. 鞋楦数字化加工系统的研究与开发[J]. 组合机床与自动化加工技术, 2003(2):3-5.
- [2] 罗 胜. 基于机器视觉的鞋模数字化及类似方法对比[J]. 工程设计学报, 2007, 14(1):57-61.
- [3] 何 海,汤春林,孙华燕. 双目立体视觉在模型姿态监测中的应用研究[J]. 计算机技术与发展, 2007, 16(11):238-240.
- [4] 王荣本,张明恒,石德乐. 双目视觉技术在目标测量中的应用[J]. 公路交通科技, 2007, 24(2):122-125.

- [5] Kim Sungjin, Choi Seungioon. The Coil recognition system for an unmanned crane using stereo vision[C]// The 30th Annual Conference of the IEEE industrial Electronics Society. Busan: [s. n.], 2004:1235-1239.
- [6] Fusiello A. Uncalibrated euclidean reconstruction[J]. Image and Vision Computing, 2000, 18(7):555-563.
- [7] 王明平,宋丽梅. 基于计算机视觉的车架号采集系统[J]. 计算机技术与发展, 2008, 18(4):239-241.
- [8] Wang C C L, Chang T K K, Yuen M M F. From laser-scanned data to feature human model: a system based on fuzzy logic concept[J]. Compute-Aid Design, 2003, 35(3):241-253.
- [9] Jebara T S, Pentland A. Parameterized structure from motion for 3D adaptive feedback tracking of faces[C]//CVPR. [s. l.]:[s. n.], 1997.
- [10] Brunig M, Niehsen W. Fast Full-Search Block Matching[J]. IEEE Trans CSVT, 2001, 11(2):241-247.
- [11] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two frame Stereo correspondence algorithms[J]. International Journal of Computer Vision, 2002, 47(1/2/3):7-42.