

基于分布式仿真系统的时钟同步技术研究

黄学进, 崔鑫, 余婷

(中国航发控制系统研究所, 江苏 无锡 214063)

摘要: 为了推进时钟同步在分布式仿真系统中的实际应用, 识别了时钟同步系统的具体要求, 在对比多种时钟同步方案后, 选择 NTP 方案并进行优化; 在介绍 NTP 时钟同步的基本原理基础上, 进行了时钟同步精度影响因子分析, 创造性的提出了构建高精度逻辑时钟、不等式法优化网络回路往返时延的不对称性、时钟晶振频率在线补偿 3 种优化方法, 并建立了时钟同步系统; 实验结果表明, 设计的时钟同步系统的同步精度优于 1 ms, 且平均校准周期达到 5 h 左右; 该时钟同步系统能封装为 DLL, 可灵活集成到具体项目中。

关键词: 时钟同步; NTP; 逻辑时钟; 不等式法优化; 时钟晶振频率补偿; 分布式仿真

Research on Clock Synchronization Technology Based on Distributed Simulation System

Huang Xuejin, Cui Xin, Yu Ting

(AECC Aeroengine Control System Institute, Wuxi 214063, China)

Abstract: In order to promote the practical application of clock synchronization in distributed simulation system, the specific requirements of clock synchronization system are identified. After comparing various clock synchronization schemes, NTP scheme is selected and optimized. On the basis of introducing the basic principle of NTP clock synchronization, this paper analyzes the influence factors of clock synchronization accuracy, creatively puts forward three optimization methods: constructing high-precision logic clock, optimizing the asymmetry of network loop round-trip delay by inequality method, and on-line compensation of clock crystal frequency, and establishes the clock synchronization system. The experimental results show that the synchronization accuracy of the designed clock synchronization system is better than 1ms, and the average calibration period is about 5h. The clock synchronization system can be encapsulated as a DLL, which can be flexibly integrated into specific projects.

Keywords: clock synchronization; NTP; logic clock; inequality optimization method; compensation of crystal oscillator frequency; distributed simulation

0 引言

时钟同步是指调整系统中网络设备或计算机的时钟, 使之与其它时钟源的时间保持一致或者偏差限定在足够小的范围内的过程^[1]。时钟同步是分布式仿真系统的核心技术之一, 在分布式仿真系统中, 数据采集、模型处理、人机交互等都分布在不同的节点上, 因此进行时钟同步的目的是维护一个全局一致的物理或逻辑时钟, 以使得系统中的信息、事件及各节点与时间有关的行为有一个全局一致的解釋^[2]或者是提供一种仿真时间的推进机制, 以确保节点发送和接收消息在时间逻辑上是完全正确的。分布式仿真系统往往具有节点操作系统类型多 (Windows 系统、Linux 系统、RTX 实时系统、LabVIEWRT 实时系统等)、仿真周期小 (毫秒级)、结构灵活、分布范围广的特点, 因此需要时钟同步系统具备兼容多种操作系统、同步精度高 (优于 1 ms)、灵活度高、成本低, 同时能考虑系统负载, 本文提出的时钟同步系统时钟校准周期要求大于 1 h。目前

常用于时钟同步的技术方案有: GPS (global positioning system) 时钟同步技术^[3]、IRIG-B (interrange instrumentation group - B) 时钟同步技术^[4]、NTP (network time protocol) 时钟同步技术^[5]、PTP (precision time protocol) 时钟同步技术^[6]等。各种方案应用于分布式仿真系统中各有优缺点, 如表 1 所示。

表 1 时钟同步方案对比

同步方案	同步精度	优点	缺点
GPS	微秒级	同步精度高, 覆盖范围广	需要 GPS 接收机, 成本较高, 且接收机必须无遮挡
IRIG-B	十微秒级	同步精度较高, 稳定性好	需要专用设备和电缆, 成本高, 组网不灵活
NTP	1~50 ms	组网灵活, 成本低	同步精度较低
PTP	纳秒级	同步精度很高	需要 1588 协议设备支持, 成本较高, 且部署不灵活

收稿日期: 2021-04-07; 修回日期: 2021-04-13。

作者简介: 黄学进 (1984-), 男, 湖北咸宁人, 硕士, 高级工程师, 主要从事航空发动机测控仿真方向的研究。

引用格式: 黄学进, 崔鑫, 余婷. 基于分布式仿真系统的时钟同步技术研究[J]. 计算机测量与控制, 2021, 29(6): 212-218.

从表中可以看出, NTP 同步与文章的同步设计目标最为接近, 只是同步精度略有不足。近年来, 许多文章对 NTP 进行了改进和优化。谌华金^[7]通过 CPU 时间戳计数器 (RDTSC, read time stamp counter) 实现了同步精度 1 ms 以内, 但是同步周期最长只有 200ms, 对原系统的 CPU 和网络负载影响较大。侯重远^[8]利用交换机的 IEEE802.1p 优先级调度功能, 将同步精度提高到 10~100, 但除了需要交换机支持调度功能外, 操作系统必须为实时操作系统 (如文中使用的 VxWorks), 不能兼容非实时 Windows 系统。

黄沛芳^[9]通过对信息包回路时延设置阈值进行数据筛选, 只保留时延 0.5 ms 以内的数据, 将同步精度提升到 250, 但这需要提前获取网络时延分布信息, 且同步间隔只有 1 min。因此, 本文基于 Windows 系统, 在 NTP 协议基本原理上, 通过逻辑时钟构建、网络时延不对称性优化、时钟频率补偿三方面进行优化, 实现设定的分布式仿真系统时间同步目标: 同步精度优于 1 ms 且校准时间间隔大于 1 h。该时钟同步方法基于 UDP 通讯, 研究结果同样可以推广到实时操作系统中。

1 NTP 基本原理

NTP 是由 RFC1305 定义的时钟同步协议, 最早是由美国特拉华大学的 David L. Mills 教授在 1982 年提出^[10], 在实际的应用过程中, 可以采用 NTP 的简化版本 SNTP (简单网络时间协议)。这里主要介绍 NTP 以客户机/服务器模式进行通信的过程, 如图 1 所示。客户机首先要生成一个 NTP 查询信息包, 通过网络发送到时间服务器; 服务器收到后, 根据本地时间生成一个 NTP 时间信息包发送给客户机^[11]。两个信息包都带有发送和接收的时间戳, 根据这 4 个时间戳 $T_1 T_2 T_3 T_4$ 来确定客户机和服务器之间的时间偏差和网络时延。

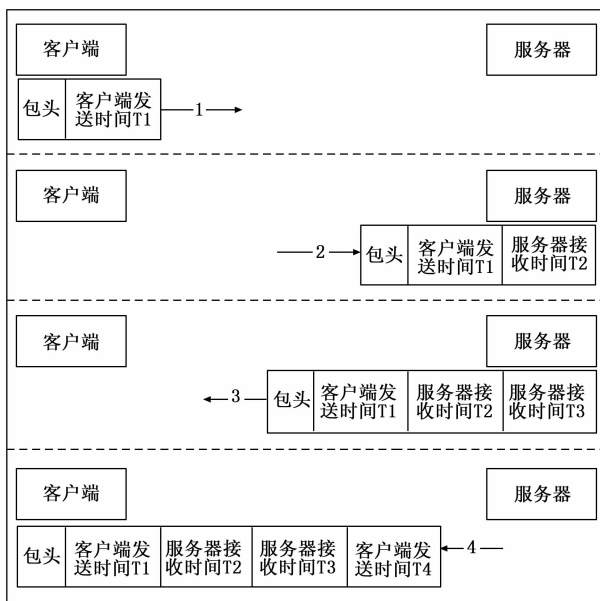


图 1 NTP 通讯过程

根据 NTP 通讯过程, 可以抽象出模型如图 2 所示^[12]。

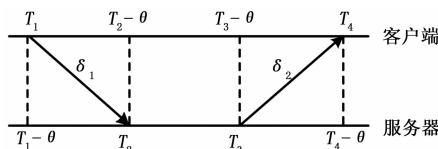


图 2 NTP 通讯模型

记客户端时间落后服务器 θ , 请求信息在网上传播耗时为 δ_1 , 回复信息在网上传播耗时为 δ_2 , 网络传输的往返延时而为 δ , 则有:

$$\begin{cases} T_2 = T_1 + \theta + \delta_1 \\ T_4 = T_3 - \theta + \delta_2 \\ \delta = \delta_1 + \delta_2 \end{cases} \quad (1)$$

4 个未知数, 3 个方程, 无法获得解析解。假设在相同的网络条件下, 客户端与服务器通讯往返回路对称, 网络时间延迟相等, 即存在:

$$\delta_1 = \delta_2 \quad (1)$$

记: 客户端落后服务器时间估计值为 $\hat{\theta}$ (θ 为真实值)。这时, 可求解得到服务器和客户端之间的时间偏差 $\hat{\theta}$:

$$\begin{cases} \hat{\theta} = \frac{(T_2 - T_1) + (T_3 - T_4)}{2} \\ \delta = (T_2 - T_1) - (T_3 - T_4) \end{cases} \quad (3)$$

2 时钟同步精度影响因子分析

根据 NTP 基本原理设计 NTP 客户端和服务器进行时钟同步测试实验, 具体设计如下:

- 1) 采用 Windows 系统时间作为时间戳;
- 2) 通讯网络使用内部局域网;
- 3) 固定周期进行时钟校准, 如表 2 所示;
- 4) 实验过程中每秒进行同步精度测试并记录数据。

实验数据整理如图 3 所示。

表 2 系统时间下同步精度实验设计

	时间戳	校准周期/min
实验 1	系统时间	5
实验 2	系统时间	20

实验结论: 1) 基于 Windows 系统时间进行同步, 短时间波动量约 15 ms, 因此同步精度最高不会优于 15 ms; 2) 长时间看, 校准周期内, 同步偏差逐渐增大, 分析认为这是服务器和客户端之间时钟晶振频率不同导致。

分析时钟同步过程如图 4 所示, 客户端和服务器获取来源于晶振的系统时间, 在经过 Windows 下的 TCP/IP 协议和网卡及网络传输后, 实现时间戳信息交换。因此, 结合 NTP 基本原理和时钟同步实验结果, 可以将引入误差的环节分为 3 个部分: 时间获取、时间传输、晶振偏差。

2.1 时间获取

Windows 时间系统主要包括实时时钟 (RTC)、基本输入输出系统 (BIOS) 时钟和操作系统 (OS) 时钟三部

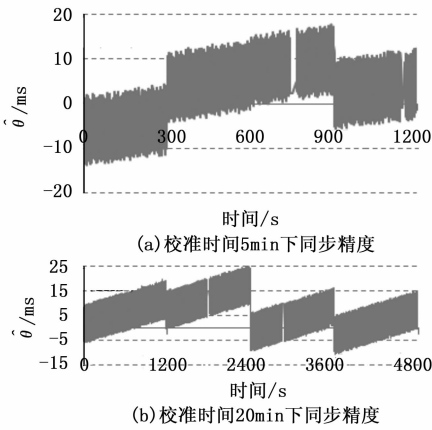


图 3 系统时间下时钟同步精度变化

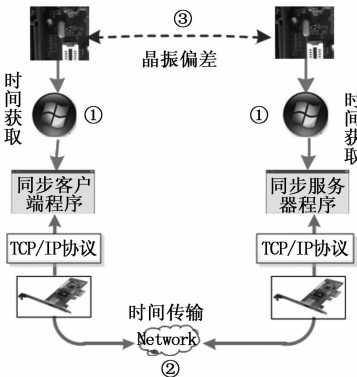


图 4 时钟同步精度影响因子

分^[13]。RTC 基于晶振和计时电路的硬件时钟。BIOS 时钟硬件部分对 RTC 时间脉冲分频产生中断信号，然后通过中断响应程序进行软件计时从而产生系统时钟^[14]。BIOS 时钟每秒产生 64 个中断信号，中断周期约为 15.6 ms，即系统时钟的时钟精度约为 15.6 ms。因此 Windows 系统常用的多个与时间有关的 API，其时间分辨率只能达到 15 ms，当高频率调用读取时间时，会发现多次返回时间相同，然后突然跳到下一个时间刻度，出现“时间停止”和“时间跳跃”现象，跳跃量约为 15.6 ms，如图 5 所示。

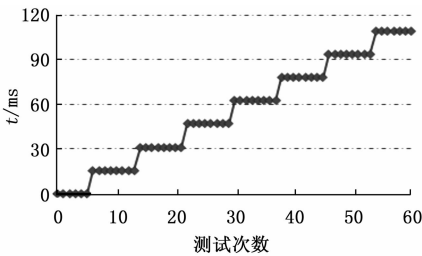


图 5 系统时间分辨率测试

因此，在 Windows 非实时系统下进行高精度时钟同步时，首先必须获取高精度的时间戳，这是进行高精度时钟同步的基础。

2.2 时间传输

时间传输部分详细过程如图 6 所示，通讯传输延迟的

不确定性主要受协议栈不确定性时延、网络路由不确定性时延和操作系统进程调度等方面影响^[15]。

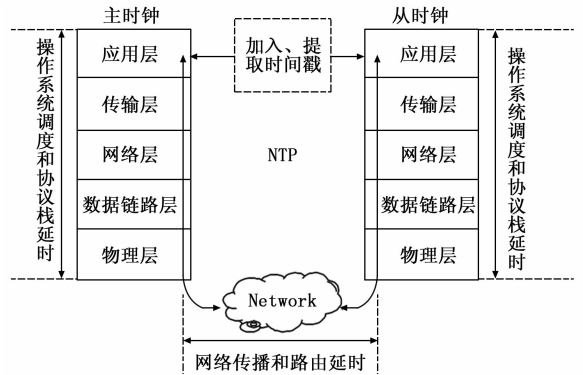


图 6 NTP 通讯传输过程及其不确定性时延

协议栈不确定性时延。TCP/IP 五层网络架构包括应用层、运输层、网络层、数据链路层和物理层，NTP 是应用层的协议，下层使用运输层的 UDP 协议。根据 UDP/IP 协议，时钟同步报文的传输需要逐层进行。传输过程受到处理器处理速度、操作系统工作方式等种种因素的影响，时钟同步报文在应用程序中开始发送到报文到达物理传输介质的时间一般无法控制，从而会引入误差。

网络路由不确定性时延。网络路由时延包括固有的链路时延、路由器处理和排队时延。报文在往返过程中不同的路由选择会导致信号的往返时延不对称。进行信息传输时，发送方首先将信息发送到本地网卡的缓冲区，然后在网络空闲时发送到接收方网卡的缓冲区，因此在网络负载高、任务繁忙的时候，接收方收到发送方信息的时间具有很大的不确定性，这就是网卡的缓存效应。因此在局域网内进行同步时，时间延迟经常是不对称的，公式计算出的时差时间差也是不准确的^[16]。

操作系统进程调度。Windows 系统采用时间片轮转机制对进程进行调度，进程调度中内存管理、UDP 协议的封装、网卡控制器里的排队等，任何一次中断或任务调度都会打断协议栈的封装或解析过程。这些操作都将会引入不可忽略的延迟波动，达到几微秒至几毫秒，造成时间延迟的不可预测。

因此，若要降低网络传输延迟对同步精度的影响，需要在处理时间戳的过程中考虑如何降低消息包在网络传输过程中不对称通信链路带来的影响。

2.3 晶振偏差

分布式仿真系统中，长时间下时钟的准确度取决于晶振频率准确度。但是受温度变化、电磁干扰、芯片老化等因素影响，晶振频率会发生小幅度波动，其中温度对晶振频率影响最大^[17]。由于工艺和材料的原因，即使同一生产线上标称频率相同的石英晶体，其实际频率也是不同的，实际频率与标称频率偏差率从量级到量级不等。以量级为例，时钟 1 h 误差达 360 ms。因此，即使在每次时钟同步过程中，从节点的授时精度很高，但由于频率偏差带来的守

时能力差, 随着时间的推移, 从节点和主节点的时钟同步精度也会逐渐变差。Veitch D 等人^[18]通过实验得出, 短时间内, 时钟偏差随时间的变化基本可以认为是线性的, 但长时间下, 时钟偏差随时间的变化是非线性的。

要提高从节点的守时能力, 主要有两个思路: 一是更换时钟源, 选取高精度、高稳定度、守时能力更强的晶体振荡器。晶体振荡器的稳定度等级从一般晶体振荡器、温度补偿型晶体振荡器、恒温晶体振荡器、铷原子振荡器逐步提高^[19], 同时晶体振荡器的成本将变得十分昂贵; 另外的思路则是采用数学算法对时钟进行建模, 以主节点的频率作为基准对从节点进行频率偏差补偿, 提高从节点的守时精度。

3 时钟同步优化

3.1 构建高精度逻辑时钟

对于一个系统而言, 时钟可分为两种: 物理时钟和逻辑时钟。物理时钟又称为绝对时钟或者墙上时钟, 对应于现实世界中的时间, 通常通过计算机内置的时钟芯片可以获得物理时钟。逻辑时钟是指自行建立的时钟, 独立于物理时钟, 但一般和物理时钟有一定的换算关系^[20]。分布式仿真系统中, 只要求系统仿真的时间和真实的时间有一定关系, 而不要求和真实时间完全相同, 因此可使用逻辑时钟。

高精度逻辑时钟的构建主要有两种方案: 1) 基于系统硬件的高分辨率性能计数器。它从系统上电开始, 以其晶振频率进行计数, 可通过相应的接口函数获取其当前计数值和晶振频率; 2) 计算机 CPU 中, 有一个 CPU 时间戳计数器, 记录了自 CPU 上电后的时钟周期数, 通过机器指令 RDTSC (read time stamp counter) 可以读取这个值^[22]。但是, 多核计算机情况下读取的 RDTSC 指令并不准确, 原因包括: 同一主板上每个核的 TSC 是不同步的, 读取的 RDTSC 可能在不同 CPU 上; CPU 时钟频率可能变化, 例如某些节能模式或超频^[21]。因此, 本文选择基于高分辨率性能计数器构建逻辑时钟。

高分辨率性能计数器提供函数 QueryPerformanceFrequency 用于获取频率, 提供函数 QueryPerformanceCounter 用于获取计数器的当前值 u , 利用两次获得的计数之差及时钟频率, 就可算出经历的精确时间^[22]:

$$\Delta t = \frac{\Delta u}{f} \quad (4)$$

取初始时刻计数值为 u_0 , 对应的系统时间为 t_0 , 当前计数值为 $u(t)$, 建立的逻辑时钟时间为 T , 根据式则有:

$$T = \Delta t + t_0 = \frac{u(t) - u_0}{f} + t_0 \quad (5)$$

经测试, 该逻辑时钟的实用时钟分辨率约为 2, 远小于同步精度指标, 可用于时钟同步, 如图 7 所示。

基于逻辑时钟设计时间同步客户端和服务器进行同步测试实验, 具体设计如下:

- 1) 采用高精度逻辑时钟作为时间戳;
- 2) 通讯网络使用内部局域网;

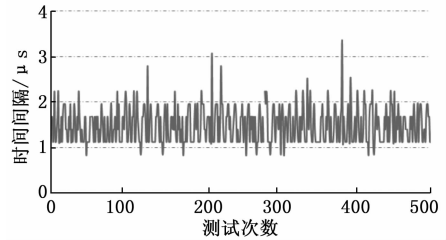


图 7 逻辑时钟分辨率测试

- 3) 固定周期进行时钟校准, 如表 3 所示;
- 4) 实验过程中每秒进行同步精度测试并记录数据。实验数据整理如图 8 和图 9 所示。

表 3 系统时间下同步精度实验设计

	时间戳	校准周期/min
实验 1	逻辑时钟	5
实验 2	逻辑时钟	20

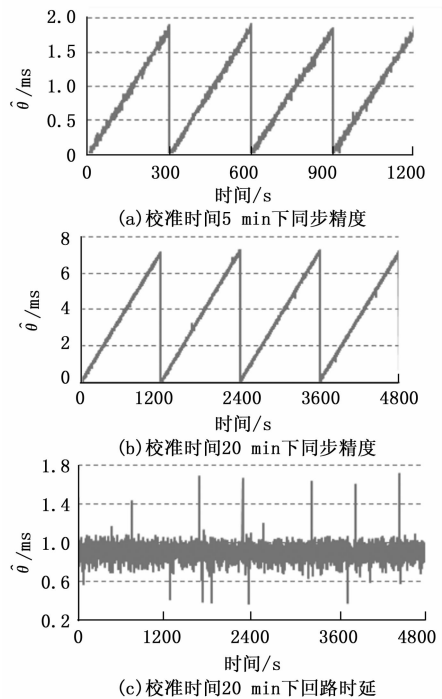


图 8 高精度时钟下同步实验结果

根据实验数据, 可以得出如下结论:

- 1) 校准周期内, 同步精度变化总体呈线性变化趋势, 变化速率约 0.36 ms/min;
 - 2) 每 5 min 校准下同步精度达到 1.8 ms, 每 20 min 校准下同步精度达到 7.2 ms。因此在本次测试环境下, 要达到 1 ms 同步精度, 则校准间隔至多只能 160 s;
 - 3) 同步精度局部存在约 0.5 ms 的毛刺, 无毛刺时存在 0.2 ms 的瞬时误差;
 - 4) 回路时延大部分在 0.8~1 ms 之间, 部分甚至超过 1.4 ms, 时延高的时候, 同步精度更差。
- 因此, 建立的逻辑时钟有效, 但还需要进一步优化。

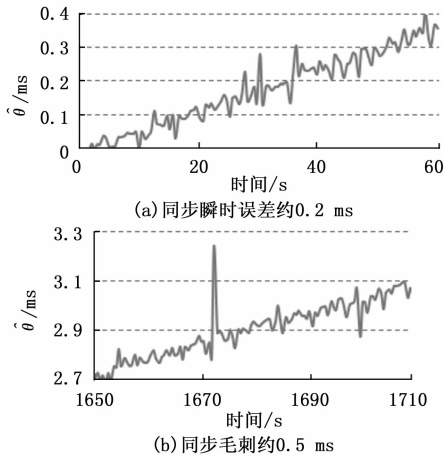


图 9 局部同步瞬时误差和毛刺

3.2 不等式法优化回路时延不对称性

根据式 (1) 和式 (3), 真实值 θ 和估计值 $\hat{\theta}$ 偏差 (授时精度) 为^[23]:

$$|\hat{\theta} - \theta| = \left| \frac{\delta_1 - \delta_2}{2} \right| \quad (6)$$

其中: δ_1 和 δ_2 为随机量, 大小在 $(0, \delta)$ 之间。因此, 网络时延不对称性越高, 授时精度越差。

根据上节测试, 同步精度变化速率为 0.36 ms/min , 即 100 ms 内时间偏差约 $0.6 \mu\text{s}$, 对同步精度 1 ms 而言可忽略。因此, 短时间内 (100 ms 左右) 连续多次通讯, 可以认为过程中 θ 不变, 如图 10 所示。不等式法就是根据记录的多组数据 $(T_{n1} T_{n2} T_{n3} T_{n4})$ 进行优选, 寻求最优数据, 计算 $\hat{\theta}$ 。

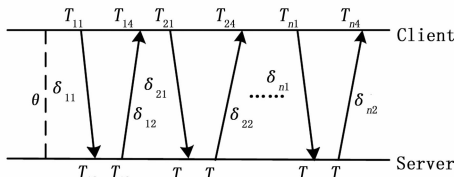


图 10 不等式法通讯过程

对应每一次往返通讯过程都有:

$$\begin{cases} T_{n2} - T_{n1} = \theta + \delta_{n1} \\ T_{n3} - T_{n4} = \theta - \delta_{n2} \end{cases} \quad (7)$$

由于 δ 为传输延迟, 都是正值, 因此可得:

$$\begin{cases} T_{n2} - T_{n1} = \theta + \delta_{n1} > \theta \\ T_{n3} - T_{n4} = \theta - \delta_{n2} < \theta \end{cases} \quad (8)$$

记 δ_{k1} 是 $\{\delta_{11}, \delta_{21}, \dots, \delta_{n1}\}$ 中的最小值, δ_{m2} 是 $\{\delta_{12}, \delta_{22}, \dots, \delta_{n2}\}$ 中的最小值, 可得:

$$\begin{cases} \min(T_{n2} - T_{n1}) = \theta + \delta_{k1} = T_{k2} - T_{k1} \\ \max(T_{n3} - T_{n4}) = \theta - \delta_{m2} = T_{m3} - T_{m4} \end{cases} \quad (9)$$

因此, 可以将 θ 范围缩小为:

$$(T_{m3} - T_{m4}) < \theta < (T_{k2} - T_{k1}) \quad (10)$$

此时取真值范围的中值作为时钟偏差估计值, 即:

$$\hat{\theta} = \frac{(T_{k2} - T_{k1}) - (T_{m4} - T_{m3})}{2} \quad (11)$$

此时估计值和真值的关系为:

$$\hat{\theta} = \theta + \frac{\delta_{k1} - \delta_{m2}}{2} \quad (12)$$

因此, 不等式法优化的物理意义为: 从多次通讯数据中, 取出往返耗时分别是最小的时候, 使用它们的差值作为时钟偏差估计值 $\hat{\theta}$ 。对比式 (12) 和式 (6), 这就将授时误差来源从单次通讯中往返时延的不对称性, 转化为多次通讯中往返分别最小时延的不对称性。显然, 最小时延更接近固有时延, 受网络波动的影响较小, 更具有对称性。

再次设计时间同步客户端和服务器进行对比实验, 具体设计如下:

- 1) 采用高精度逻辑时钟作为时间戳;
- 2) 固定周期 20 min 进行时钟校准;
- 3) 网络使用内部局域网, 采用不等式法优化回路往返时延的不对称性;
- 4) 实验过程中每秒进行同步精度测试并记录数据, 如图 11 和图 12 所示。

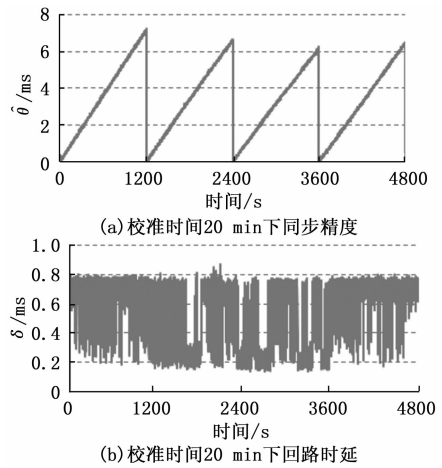


图 11 不等式法下同步实验结果

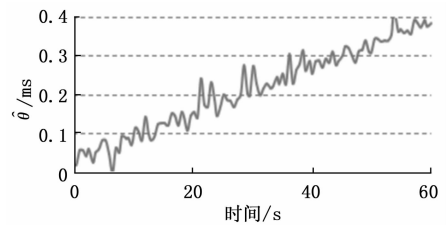


图 12 不等式法下局部瞬时误差

根据实验数据, 可以得出如下结论:

- 1) 回路时延比优化前明显减小, 基本在 0.8 ms 以内;
- 2) 同步精度毛刺消失, 瞬时误差下降到 0.1 ms ;

因此, 经过不等式法优化, 回路往返时延不对称性对同步精度的影响明显减小, 同步精度线性变化的趋势得到进一步加强。

3.3 时钟晶振频率在线补偿

3.3.1 补偿系数模型

将客户端在 t 时刻的逻辑时钟表示为 $c(t)$, 服务器逻辑时钟表示为 $s(t)$, 定义客户端和服务器时间差为 $\theta(t)$, 根据式有:

$$c(t) = \frac{\Delta u_c(t)}{f_c} + t_0 \quad (13)$$

$$s(t) = \frac{\Delta u_s(t)}{f_s} + t_0 \quad (14)$$

$$\theta(t) = s(t) - c(t) \quad (15)$$

其中 t_0 为高精度时钟建立时的系统时间, 是一个常量。理论上:

$$\Delta t = \Delta s(t) = \Delta c(t) \quad (16)$$

$$\frac{\Delta u_s(t)}{f_s} = \frac{\Delta u_c(t)}{f_c} \quad (17)$$

但随着时间的进行, 客户端和服务器的时间变化量并不相等, 原因在于受环境条件的影响 (如温度、电压、老化), 时钟频率不断的产生漂移。分布式仿真系统中, 并不需要系统中时间与真实时间完全一致, 只需要保持相同的时间基准即可。因此, 本文以服务器时钟为基准, 对客户端进行晶振频率补偿。

记客户端时钟晶振补偿系数为 α , 补偿后频率为 \tilde{f}_c , 则有:

$$\tilde{f}_c = \alpha \cdot f_c \quad (18)$$

因此有:

$$\Delta t = \frac{\Delta u_s}{f_s} = \frac{\Delta u_c}{\tilde{f}_c} = \frac{\Delta u_c}{\alpha \cdot f_c} \quad (19)$$

当 $\alpha=1$ 时, 表示客户端真实时钟晶振频率与标称频率相同, 与服务器时钟快慢相同; 当 $\alpha<1$ 时, 表示客户端真实时钟晶振频率小于标称频率, 即本身时钟相对服务器时钟较慢; 当 $\alpha>1$ 时, 表示客户端真实时钟晶振频率大于标称频率, 即本身时钟较快。

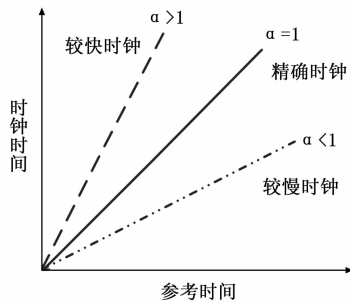


图 13 补偿系数与时钟相对快慢的关系

3.3.2 补偿系数在线求解与迭代

记 $C(t)$ 为客户端补偿后的理想时钟, 则有:

$$C(t) = s(t) \quad (20)$$

根据式 (13)、式 (19) 和式 (20), 可得:

$$C(t) = \frac{\Delta u_c(t)}{\alpha f_c} + t_0 + \theta_0 \quad (21)$$

其中: θ_0 为建立时刻客户端与服务器的时间差, 是一个常量。

因此, 根据式 (15) 和式 (21):

$$\theta(t) = C(t) - c(t) = \frac{1-\alpha}{\alpha} \cdot \frac{\Delta u_c(t)}{f_c} + \theta_0 \quad (22)$$

结合式 (13), 可得:

$$\theta(t) = \frac{1-\alpha}{\alpha} \cdot c(t) - \frac{1-\alpha}{\alpha} \cdot t_0 + \theta_0 \quad (23)$$

将 $\theta(t)$ 对 $c(t)$ 求导, 可得:

$$k = \frac{d\theta(t)}{dc(t)} = \frac{1-\alpha}{\alpha} \quad (24)$$

即:

$$\alpha = \frac{1}{k+1} \quad (25)$$

因此, 理论分析可得出, 若从时钟相对主时钟是严格稳定的, 即时钟晶振频率补偿系数为常数, 那么时钟偏差 $\theta(t)$ 是按线性变化的。

在此基础上, 本文采用最小二乘对斜率 k 进行拟合, 削弱测量噪声影响, 从而得到补偿系数 α 。即每过段时间记录一次当前时间 t_i 和时间偏差 $\hat{\theta}_i$, 当达到设定的条件时, 对记录的数据采用最小二乘拟合。最小二乘拟合公式为:

$$k = \frac{n \sum (t_i \hat{\theta}_i) - \sum t_i \cdot \sum \hat{\theta}_i}{n \sum t_i^2 - (\sum t_i)^2} \quad (26)$$

而由于随着时间的推移, 晶振频率会发生漂移, 当前补偿得到的频率过一段时间后已经不够准确了, 需要再次进行频率补偿。因此, 需要继续研究时钟频率补偿系数在线迭代问题。

记 α_i 为第 i 次晶振补偿系数, $C_i(t)$ 为客户端第 i 次迭代时的理想时钟 (与服务器时间保持一致)。则进行第 i 次迭代时, 参照式 (20) 和 (21) 有:

$$s(t) = C_i(t) = \frac{\Delta u_i(t)}{\alpha_i f_c} + t_0 + \theta_0 \quad (27)$$

此时客户端逻辑时钟 $c(t)$ 为上次迭代时的理想时钟, 即:

$$c(t) = C_{i-1}(t) = \frac{\Delta u_c(t)}{\alpha_{i-1} f_c} + t_0 + \theta_0 \quad (28)$$

同理可解得:

$$\theta(t) = \left(\frac{\alpha_{i-1}}{\alpha_i} - 1 \right) \cdot [c(t) - t_0 - \theta_0] \quad (29)$$

将 $\theta(t)$ 对 $c(t)$ 求导, 可得:

$$k = \frac{d\theta(t)}{dc(t)} = \frac{\alpha_{i-1}}{\alpha_i} - 1 \quad (30)$$

即:

$$\alpha_i = \frac{\alpha_{i-1}}{k+1} \quad (31)$$

应用新的 α_i 即可实现补偿系统的动态迭代, 从而使系统能够维持高精度时钟同步。

4 实验结果与分析

应用高精度逻辑时钟、不等式法优化回路时延不对称性和时钟晶振频率在线补偿与迭代 3 种方法, 设计时间同步客户端和服务器的同步实验, 客户端工作过程如图 14 所示:

- 1) 创建逻辑时钟, 使用不等式法与服务器同步一次;
- 2) 等待 0.5 s 后, 使用不等式法获取当前时间偏差 $\hat{\theta}$ 和当前客户端时间 t , 并记录为一组数据;
- 3) 判断当前 $\hat{\theta}$ 是否超过性能指标 γ , 设置 $\gamma=0.9$ ms。若

没有, 则重复步骤 2); 若超过性能指标 γ , 则转到步骤 4);

4) 使用记录的数据, 采用最小二乘拟合得到补偿系数 α , 更新客户端时钟并清空记录的数据, 再转步骤 2)。

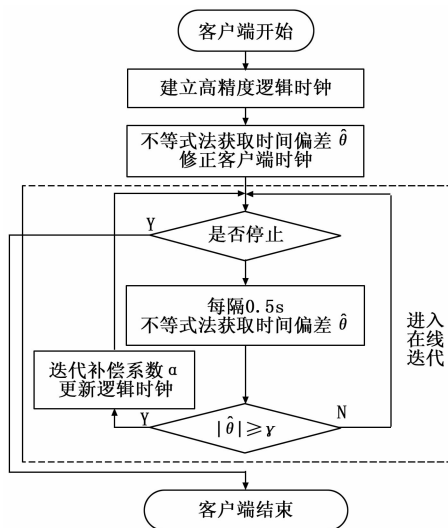


图 14 客户端工作流程

经过约 60 h 的长试实验, 结果如图 15 所示, 从图和数据中可以得出: 1) 实验在 137s 后进入在线迭代环节, 初次拟合得到补偿系数为 $\alpha = 0.999\ 993\ 35$; 2) 时钟补偿系数 α 迭代次数达到 12 次, 平均可以维持时间达到 5 h; 3) α 在 0.999 993 05~0.999 993 79 之间波动, 由于标称频率为 3.58 MHz, 所以实验过程中实际晶振频率与标称频率之间偏差量在 22.2~24.9 Hz 之间波动, 波动量约 2.7 Hz。

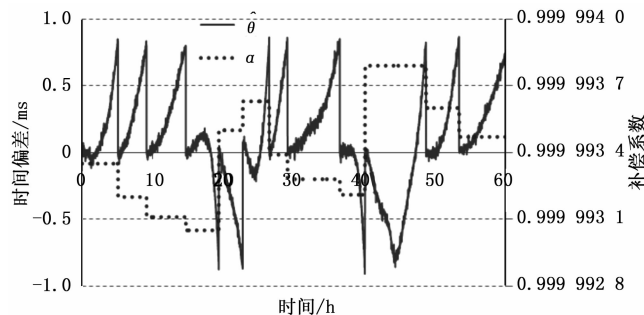


图 15 频率补偿在线迭代下同步效果

5 结束语

本文在介绍 NTP 基本原理的基础上进行了时钟同步精度影响因子分析, 并针对性采用了构建高精度逻辑时钟、不等式法优化回路时延不对称性、时钟晶振频率在线补偿 3 种解决策略, 进行了时钟同步优化。通过长试实验, 验证了优化后的时钟同步系统, 能有效的实现时钟同步精度优于 1 ms 且平均校准周期达到 5 h。项目应用时, 可以将时钟同步客户端封装为 DLL, 集成到各分布式仿真系统软件中实现时钟同步, 具有较好的工程应用价值。

参考文献:

[1] 刘占超, 房建成, 刘百奇. 一种改进的高精度 POS 时间同步方

法 [J]. 仪器仪表学报, 2011, 32 (10): 2198-2204.

[2] Lee H K, Lee J G, Jee G I. Calibration of measurement delay in global positioning system/strapdown inertial navigation system [J]. Journal of Guidance Control & Dynamics, 2002, 25 (2): 240-247.

[3] 张红亮. 分布式系统时钟同步技术的研究与应用 [D]. 长沙: 国防科学技术大学, 2002.

[4] 贾振华, 许晓冬, 吕德东, 等. 基于 IRIG-B 码的测控时间系统的设计 [J]. 电子设计工程, 2017 (12): 77-80.

[5] 陈源, 刘明波, 侯孝民, 等. 基于 IRIG-B 码的软件化测控系统时间同步方法研究 [J]. 遥测遥控, 2012 (2): 53-57.

[6] Mills D L. Internet time synchronization: the network time protocol [J]. IEEE Transactions on Communications, 1991, 39 (10): 1482-1493.

[7] 谌华金, 罗双喜, 何佳洲. 利用 CPU 时间戳实现局域网时间同步的新机制 [J]. 计算机应用, 2009, 29 (11): 2920-2924.

[8] 侯重远, 江汉红, 芮万智, 等. 面向交换式电力监测网的 NTP 同步精度提高方法 [J]. 电力自动化设备, 2013, 33 (1): 148-152.

[9] 黄沛芳. 基于 NTP 的高精度时钟同步系统实现 [J]. 电子技术应用, 2009 (7): 122-124.

[10] Yamashita T, Ono S. A statistical time synchronization method for frequency-synchronized network clocks in distributed systems [J]. IEEE Transactions on Information & Systems, 2004, 87 (7): 1878-1886.

[11] 黄长江. 局域网时间同步应用研究 [D]. 北京: 中国科学院研究生院, 2006.

[12] 徐强, 汪芸. 基于 SNTP 同步模型的时钟同步系统 [J]. 计算机应用与软件, 2007, 24 (9): 54-56.

[13] 马素刚, 孟伟君. 基于 SNTP 的网络时钟软件设计与实现 [J]. 西安邮电学院学报, 2009, 14 (3): 98-101.

[14] 陈朝福, 王磊. 局域网环境的网络时间协议性能测试 [J]. 计算机应用, 2012, 32 (4): 943-945.

[15] 张国军, 温春林. 基于 SNTP 的多控制系统时钟同步方案的设计与应用 [J]. 仪器仪表标准化与计量, 2013 (2): 38-41.

[16] 魏丰, 孙文杰. IEEE-1588 协议时钟同步报文的精确时间标记方法研究 [J]. 仪器仪表学报, 2009 (1): 162-169.

[17] 李晓珍, 苏建峰. 基于 IEEE1588 高精度网络时钟同步的研究 [J]. 通信技术, 2011, 44 (3): 105-107.

[18] Veitch D, Ridoux J, Korada S B. Robust synchronization of absolute and difference clocks over networks [J]. IEEE/ACM Transactions on Networking (TON), 2009, 17 (2): 417-430.

[19] 吴歆馨. 基于 IEEE1588 高精度网络时钟同步系统的研究 [D]. 武汉: 华中科技大学, 2007.

[20] 张城. 基于 IEEE1588 协议的网络同步时钟技术的研究 [D]. 杭州: 浙江大学, 2013.

[21] 李晓珍. 基于 IEEE1588 的网络时间同步系统研究 [D]. 北京: 中国科学院研究生院, 2011.

[22] 王刚, 黄飞, 乔纯捷, 等. 分布式网络时钟同步研究 [J]. 仪器仪表学报, 2008, 29 (11): 2399-2403.

[23] 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.