

分布式节点群的参量差分时钟同步算法

凌云翔¹, 范林军¹, 王涛², 张国华¹

(1. 国防科学技术大学 信息系统工程重点实验室, 湖南 长沙 410073; 2. 不来梅大学 数学与计算机科学学院, 德国 不来梅 28359)

摘要: 结合分布式系统多平台性、通用性、阶段同步性等新特征, 提出一种新的分布式节点群参量差分逻辑时钟模型(LCPD)及其同步算法。该模型不修改本地时间, 将不确定的网络延迟考虑到差分频率中, 通过差分时钟计数值的递归演算和差分频率的收敛校准, 获取各节点稳定的计数值, 以此作为衡量全网时钟同步的标志。算法采用参考广播、锁相环和协同同步原理, 步骤分为零点校对和差分校准。仿真实验表明: 该算法一旦同步就不再失步, 具有较好的收敛性和稳定性, 可满足大规模、异构分布式仿真系统的同步需求。

关键词: 计算机应用; 分布式仿真系统; 时钟同步; 参量差分; 分布式节点群; 逻辑时钟

中图分类号: TP393 **文献标志码:** A **文章编号:** 1000-1093(2013)01-0051-09

DOI: 10.3969/j.issn.1000-1093.2013.01.010

A Clock Synchronization Algorithm Based on Parametric Difference for Distributed Node Clusters

LING Yun-xiang¹, FAN Lin-jun¹, WANG Tao², ZHANG Guo-hua¹

(1. Science and Technology on Information Systems Engineering, National University of Defense Technology, Changsha 410073, Hunan, China; 2. Faculty of Mathematics and Computer Science, University of Bremen, Bremen 28359, Germany)

Abstract: By analyzing some new features of distributed systems such as multi-platform, generality, phase synchronization et al, a novel logic clock model based on parametric difference (LCPD) and its synchronization algorithm for distributed node clusters (SADNC in short) were proposed, In LCPD, the indeterminate network delay is considered into the changing parametric difference frequency (PDF) and the local clock value is not modified. By converging on the dynamic PDF and calculating the counting values of LCPD recursively, all distributed nodes are synchronized in the whole network once each node's counting value is equivalent and stable. The steps of SADNC algorithm consist of zero-point time calibration and synchronization of LCPD, in which some ideas of methods such as parametric difference broadcast, phase-locked loop and cooperative time synchronization are adopted. The simulation experimental results show that the stability and convergence of SADNC are excellent and all nodes hardly can be out of synchronization once they have been synchronized, which can meet the time synchronization requirement of large-scale, heterogeneous distributed simulation systems.

Key words: computer application; distributed simulation system; clock synchronization; parametric difference; distributed node cluster; logic clock

收稿日期: 2011-12-28

基金项目: 国家自然科学基金项目(60875048, 61005055)

作者简介: 凌云翔(1971—)男, 教授, 博士生导师。E-mail: yxling@tom.com;

范林军(1985—)男, 博士研究生。E-mail: lxfan_nudt@163.com

0 引言

随着普适计算、网格计算、物联网等新兴分布式应用技术与能源、传感器、片上系统等相关技术不断走向成熟^[1-4],越来越多的分布式系统以新颖的工作方式、全新的体系结构和动态多变的网络拓扑出现。可以预见,下一代分布式系统将超越传统,具有以下几个特点^[5]:普适性、动态组合性、冗余性、阶段同步性等。

然而,一个基本的、至关重要的问题自分布式系统诞生之日便困扰着开发人员——时钟同步。针对传统分布式系统时钟同步,先前学者们取得了大量的研究成果,提出了许多协议与算法^[6-11]。基于 GPS 的时钟同步^[6]属于硬同步,其同步精度高,效率高;但其穿透性差,价格昂贵,不适合大规模分布式部署。Mills 教授在 1985 年提出的网络时间协议^[7](NTP)由于其鲁棒性、可扩展性、灵活、便捷部署等诸多优点而作为网络时间协议标准。特别是 GPS 和 NTP 相结合^[10]以及用于工控领域的精密时间协议(PTP)^[12],更能达到微秒级、纳秒级的同步精度。但 NTP 由于不确定的网络时延、频繁的数据交换、耗费带宽等缺点而在非传统分布式网络中很少采用^[13],PTP 硬件协助同步代价太高。虽然不少学者对传统同步协议进行改进^[14-15],但依旧存在网络时延、时钟偏移等问题。

无线传感器网络(WSN)是新兴分布式系统的典型代表,研究者提出很多针对 WSN 的优秀同步算法^[16-20],这为我们提供了借鉴的思路。Jeremy 等提出参考广播同步^[16](RBS),其通过缩短同步的关键路径,减少报文传输延迟和发送中断等待时间等不确定性,排除发送和访问时间,来提高同步精度。但 RBS 不适合点对点网络,且算法的收敛时间较长。为提高同步精度和算法的可扩展性,学者们又提出了传感器网络时间同步协议^[17](TPSN)和泛洪时间同步协议^[18](FTSP)。但 TPSN 能耗高,不适合移动部署;FTSP 占用过多的带宽,且基于 MAC 层时间戳计数,通用性差。Servetto 等^[19]和 Werner-Allen 等^[20]分别提出了萤火虫算法(RFA)和协同同步算法(CTS),以“互同步”、“协作同步”的全新思路来解决 WSN 中时间同步问题,由于 RFA 良好的收敛性能、CTS 低同步误差等优点,经完善后它们很可能作为未来 WSN 的同步方案。由于传统以太网和多

任务操作系统模式下工作的群同步过程存在如网络时延、系统调度时延、打包解包时延、函数调用时延等不确定性因素,因此,WSN 中算法的思想和数据处理方式在通用的分布式时钟同步方法中并不适用。

考虑到下一代分布式系统的全新特征,传统时间同步方法将面临诸多棘手问题(如更长的网络时延、带宽瓶颈、同步节点的不确定性等)。目前,针对异构、通用的分布式平台的普适型时间同步协议与算法几乎很少。本文针对新型分布式系统的特点,从时钟同步本质需求出发,在深入研究包括传统网络和新型 WSN 等在内的经典同步方法的优势和不足的基础上,提出了仿真系统下基于参量差分的逻辑时钟(LCPD)模型及其分布式节点群同步算法(SADNC),仿真结果表明 LCPD 模型的可行性和 SADNC 算法的收敛性、稳定性和可操作性。旨在从一个全新的角度诠释和解决分布式仿真系统中的时钟同步问题。

1 相关概念

1.1 分布式节点群

分布式系统是一个相对笼统的概念,目前并没有严格的定义。文中泛指涉及分布式节点访问的应用,且其具体形式依赖于相应的设计架构或工作模式。由于依托网络在结构、拓扑和规模等因素上的差异,相应背景下的分布式系统的定位不同,因此,其对应节点群的构成也不尽相同。一种简单典型的分布式节点群构成是一组面向同类应用的具有类似参数(精度、稳定性等)的仿真节点。文中相关时钟同步的讨论主要针对满足上述构成的分布式节点群。图 1 描述了几类异构环境下分布式节点群的架构。

1.2 同步的范畴

一般地,同步是指两个或两个以上随时间变化的量在变化过程中保持一定的相对关系^[21]。由同步的定义可见,同步本质上是一种时间相关的保持关系,即要求同步对象在整个作用过程中应当始终保证其相对关系的顺序一致性。

集中式系统的同步主要采用信号量、锁、临界区等解决方案^[22-24],而在分布式仿真系统中,由于系统的异构性和物理上的分散性,同步则主要倾向于

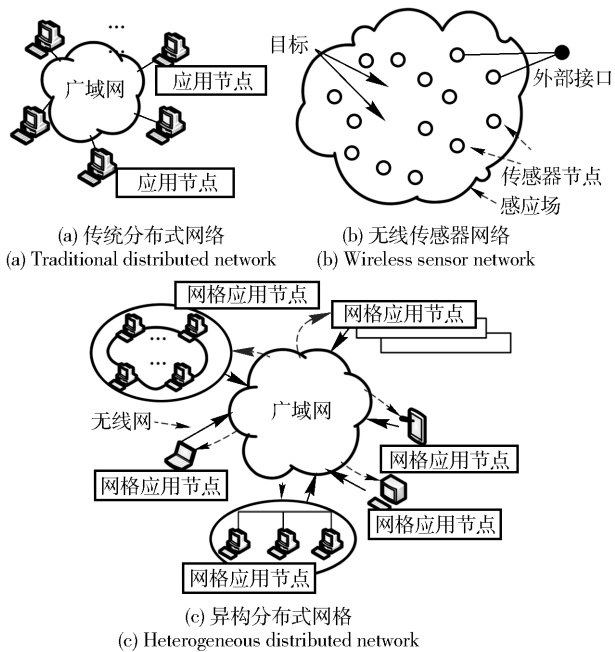


图 1 异构分布式节点群

Fig. 1 Heterogeneous distributed node clusters

考虑时间相关的同步解决方法。因此,分布式仿真系统中的主要同步对象是各分布节点的时间,可以由所有节点共同维护或由特定节点维护的逻辑时间,也可以是由每个节点各自维护的本地时钟的指示时间。

一般地,时间同步划分是按照同步的需求层次进行的,分为排序、相对同步、绝对同步 3 类^[6,16]。排序,即通过事件排序来实现事件发生先后顺序的判断,是最简单的同步方式;相对同步,侧重于不直接修改本地时间值,在保持各节点的本地时钟连续运行(不跳变)的基础上,通过获取其他节点如时钟偏移和时钟漂移等信息,实现不同节点的本地时间值之间的转换,达到同步的目的;绝对同步,则是系统内存在一个所有节点需要向其对准的时间参考基准,由于非基准节点要保证本地时间与标准一致,本地时间可能因为时间协议修改导致本地时间发生跳变。

2 LCPD 模型

LCPD 模型基于参量差分思想和逻辑时钟概念,将不可预知的网络延迟考虑到差分频率的调整中,它并不修改本地计算机的物理时钟,旨在提供一种逻辑时间戳,保证分布式节点群随仿真推进的时间顺序关系。文中是在文献[25]研究成果的基础

上,进一步阐述和验证模型与算法的优缺点与可行性。

2.1 LCPD 原理

有两个基本假定:1) 每个时钟节点本身在所需精度范围内是稳定的。2) 网络状态在统计学上是稳定的。一般地,计算机物理时钟的模型为

$$T(t) = T(t_0) + R(t - t_0) + \frac{1}{2}D(t - t_0)^2, \quad (1)$$

式中: t 是当前物理时间; $T(t)$ 是从时间 t_0 开始到当前时间 t 的时钟偏移量; R 是时钟频率偏移; D 是由于计算机的可变频率振荡器老化引起的漂移率。通常情况下,时钟频率偏移 R 受温度的影响较大,其误差变化范围从几百万分之一到几万分之一。由(1)式右边第 3 项引起的误差主要由其前两项决定。从上面的分析也可以看出,引起时钟偏差的主要因素是可变频率振荡器的频率漂移。假定时钟是稳定的,即时钟频偏 R 为固定值,那么计算机物理时钟的计时模型为

$$C(t) = \alpha t + \beta, \quad (2)$$

式中: $C(t)$ 为计算机物理时钟的计时结果; α 为计算机时钟标称频率与实际频率的比值; β 为计算机时钟计时初始偏移。

一般计算机的物理时钟实际频率不可修改,因此,依靠直接校正 $C(t)$ 的同步方法在获得同步后的计算机物理时钟在一定时间后必然会失步,且会造成计时结果的跳变而失去本地时间的连续性。由于计算机时钟标称频率的不准确性,不少同步方法将重点放在如何测得更加准确的实际频率值,再由此计数差除以测得频率来计算时间。这类方法测得频率的精度和校正对计算时间的影响很大(如频率的细微改变可能导致计算时间的剧烈跳变和回溯)。

既然物理时钟的频率不可调,时间为防止跳变也不能调,那么我们就构造一个计时结果和频率可以调整的时钟,即仿照计算机物理时钟的计时模型构造虚拟时钟,称作差分时钟,

$$P_{k,n+1} = P_{k,n} + F_{k,n} \Delta T, \quad (3)$$

这是一个递推计时模型,其中: $F_{k,n}$ 为差分时钟频率; $P_{k,n}$ 为差分时钟计数值, $k = 1, 2, 3, \dots, n$ 表示时钟节点个数, $n = 1, 2, 3, \dots$ 表示步进次数; $|\Delta T| = T_{s+1} - T_s$ 为逻辑时间间隔, T_s 表示某个特定的逻辑时刻。 $P_{k,n}$ 可简记为 P 。理想的 LCPD 对应于物理时刻 t , 各节点应当具有相同的差分计数值 P ; 同时,

LCPD 具有良好的反演特性,即如果某一时刻各节点差分计数值均为 P ,则可推断他们处于逻辑时间的同一时刻 T 。

2.2 LCPD 特征

LCPD 模型不同于传统的物理时钟模型,其特征如下:

1) 不修改本地物理时钟。LCPD 并不改变计算机本地物理/系统时钟,能够避免因频繁校正物理时钟导致操作系统及时钟相关的应用混乱。

2) 严格顺序关系。差分时钟将同步对象转移到 LCPD 的计数值 P 上,即同步的目的是令所有节点的差分时钟理论上在任何步进周期内保证严格全局顺序关系。

3) 即时时戳时钟。差分计数值 P 在系统调用瞬间即时生成。

4) 混合时钟色彩。LCPD 通过差分频率和计数值在“逻辑”上运行,表明其具有逻辑时钟特征;而稳定后的 LCPD 又可通过其与物理时钟的严格函数关系换算,实现与物理时钟类似的功能。

5) 良好的收敛性与稳定性。LCPD 将锁相环原理引入到分布式仿真系统时钟同步中,通过调整差分频率,在各节点差分时钟间形成闭环控制,可实现 LCPD 的快速收敛和趋于稳定,从而将原本分散于各个节点的不确定性因素通过各节点差分时钟频率的不断校正平均到整个网络中来。

6) 仿真节点动态组合。收敛的 LCPD 允许分布式节点群频繁地动态组合与分解。

7) 提供同一节点对不同时间分辨率的仿真应用。

2.3 SADNC 算法

2.3.1 单跳网方案

SADNC 算法参考 NTP 中具有良好跟踪性能的锁相环系统,RBS 中有利于群同步的广播和差分思想,协同同步算法中依靠空间平均而非传统的时间平均的多播预测思想,萤火虫算法中“互同步”思想。

低时延的具有简单平面拓扑结构的单跳网是基于参量差分的分布式时钟群同步方法的最为理想的运行环境,依靠 LCPD 实现单跳网的同步主要分零点定时、差分对准两类操作。

1) 零点定时过程

零点定时即同步过程的初始化阶段,旨在为之后的差分对准提供一个相对可供信赖的基准。定时

过程采用与协同同步方法类似的方法,即由同步发起者发起同步,按照同步发起者发送 w 个同步包,其他接收节点根据接收情况推算第 $w+1$ 个同步包的发送时刻 T_s ,所有节点在 T_s 时刻将自己的 LCPD 计数值 $P_{k,0}$ 置零,差分频率 $F_{k,0}$ 置为初始频率 F_0 ,即所有节点 LCPD 均有

$$P_{k,0} = 0, F_{k,0} = F_0, k = 1, 2, 3, \dots, m.$$

图 2 描述了 SADNC 算法零点定时过程。

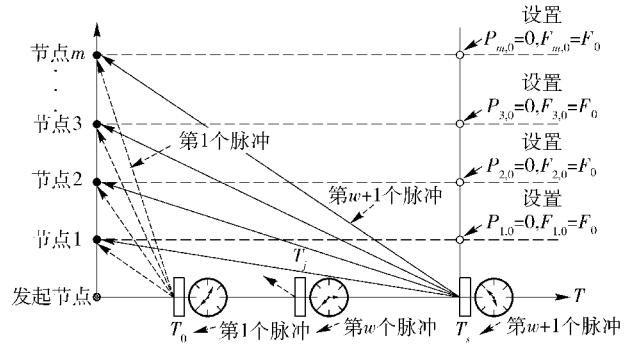


图 2 SADNC 算法零点定时过程

Fig. 2 Zero-point time calibration of SADNC

2) 差分校准过程

差分校准为主要的同步过程,在这个过程中需保证存在第三方节点担任或系统内存在若干节点轮流担任参考信标发送者,同步节点收到参考信标后即读取当前本地 LCPD 的计数值,而后所有同步节点互换 LCPD 数据,而后根据接收到的时钟数据进行本地 LCPD 差分频率的校正。图 3 描述了 SADNC 算法同步校准过程。相关 SADNC 算法计算公式如下

差分时钟计数

$$P_{k,n} - P_{k,n-1} = F_{k,n-1} \times \frac{C_{k,n} - C_{k,n-1}}{f'_k}, \quad (4)$$

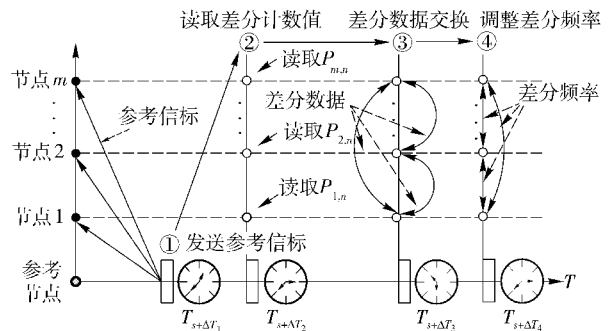


图 3 SADNC 算法同步过程

Fig. 3 Clock synchronization of SADNC

式中: $P_{k,n}$ 为第 k 个节点 n 次步进时差分时钟的计数值; $F_{k,n-1}$ 为第 k 个节点 $n-1$ 次步进时差分时钟频率; $C_{k,n}$ 为物理时钟计数值; f_k 为计算机物理时钟标称频率。

差分时钟计数值的均值

$$\bar{P}_n = \frac{1}{m} \sum_{k=1}^m P_{k,n}, \quad (5)$$

式中: \bar{P}_n 为差分校准过程中互换信标中所有节点计数值的平均值; m 为同步节点个数。

差分时钟频率的校正

$$F_{k,n} = F_{k,n-1} \times \frac{\bar{P}_{n-1}}{P_{k,n-1}}. \quad (6)$$

差分时钟频率的收敛

$$F'_{k,n} = (FT_{k,n} + FB_{k,n})/2, \quad (7)$$

式中: $FT_{k,n}$ 为第 k 个节点 n 次步进时差分频率的峰

值; $FB_{k,n}$ 为谷值; $F'_{k,n}$ 为收敛后的差分频率。

SADNC 算法执行同步收敛操作的整个步骤用伪代码表示如算法 1 所示。

2.3.2 多跳扩展方案

由于采用基于多播/广播差分的方式工作,为防止出现广播风暴,必须限制同步单跳网的规模使其不能过于庞大;另一方面,为解决更大规模的分布式仿真系统同步问题,则需要引入超节点实现树形分层结构,逐级渐进同步。首先由顶层网络发起单跳网 LCPD 同步,进而由各级子网的超节点依次激发各级子网的同步行为,从而引发全网的 LCPD 同步过程,经过一定时间的运行后实现全网基于 LCPD 的同步,同时,随着网络规模的扩大,单个节点对于全网 LCPD 的稳态影响将越来越小,从而保证全网同步的稳定性。

```

1  for each calibrating step  $i$  ( $i \in [1, n]$ ) do
2    while(  $|P_{j,i} - \bar{P}_i| \neq 0, j \in [1, k]$  ) do
3      /* 当且仅当每个节点的差分计数值与所有节点计数值的均值相同时算法终止 */
4      Compute  $C_{j,i}$  for all nodes at real time  $t$  by Eq. (2) /*  $C_{j,i}$  为物理时钟计数值 */
5      Compute  $P_{j,i}$  for all nodes by Eq. (4) /*  $P_{j,i}$  为差分时钟计数值 */
6      Calculate the average value  $\bar{P}_i$  by Eq. (5) /*  $\bar{P}_i$  为所有节点计数值的均值 */
7      Adjust the frequency of LCPD  $F_{j,i}$  by Eq. (6)
8      for each synchronous node  $j$  ( $j \in [1, k]$ ) do
9        Find the peak value  $FT_{j,i}$ ; /* 寻找差分频率的峰值 */
10       Find the lowest value  $FB_{j,i}$ , similarly /* 寻找差分频率的谷值 */
11       if  $FT_{j,i} \neq 0$  &&  $FB_{j,i} \neq 0$ 
12         Execute the convergent operation by Eq. (8) /* 差分频率的收敛 */
13       endif
14     endfor
15   endwhile
16 endfor

```

算法 1 SADNC 算法同步收敛过程

Algorithm 1 Synchronziation and convergence process of SADNC

2.4 SADNC 算法的局限性

任何方法都不是完美的,都存在其理论上的局限性。本文同步算法的核心是通过各节点 LCPD 快速收敛来实现全网并行同步,因此, LCPD 的性能直接决定同步算法效果。由 LCPD 工作原理可知, LCPD 通过网络状态的统计分布特性来实现各节点不确定性向全网系统确定性

转移,将各节点不确定的因素分布到全网各节点差分频率中去。因此,可以预见在网络规模特别小的情况下, LCPD 的全网稳定性并不理想。 LCPD 模型基于计算机物理时钟是稳定但不准确的假设,与实际情况并不完全相符,因此,各节点实际物理时钟的稳定性会为 LCPD 性能带来一定的影响。

3 测试与分析

3.1 测试条件

实验测试在 Windows XP 平台下,基于 MATLAB R2009a,采用 Intel dual-core E5400 2.70 GHz 和 2 GB 内存的配置。设定计算机物理时钟模型为 $C_{k,n} = f_{k,n} \times t$,其中 $C_{k,n}$ 为第 k 个节点第 n 次校准的物理时钟计数值, $f_{k,n}$ 为时钟真实频率, t 为物理时间。测试分为 3 组节点,每组节点数量依次为 10、20、30。测试中每组节点的真实频率、节点数量、仿真步长、初始差分频率和计数值分别设定如表 1 所示。

表 1 仿真实验配置

Tab. 1 Configuration of simulation experiments

组	物理时钟频率/Hz	数目	步长/s	F_0 /Hz	$P_{k,0}$
1	980,1 020,990,1 005,1 000,	10	0.1	1 000	0
	995,1 012,998,1 010,1 002				
2	980,1 020,990,1 005,1 000,	20	0.1	1 000	0
	995,1 012,998,1 010,1 002,				
	1 010,996,979,1 008,1 000,				
3	992,1 014,1 003,988,1 002	30	0.1	1 000	0
	980,1 020,990,1 005,1 000,				
	995,1 012,998,1 010,1 002,				
	1 010,996,979,1 008,1 000,				
	992,1 014,1 003,988,1 002				
	989,1 015,996,1 005,1 007,				
	997,1 014,999,1 019,1 002				

3.2 测试原理

考虑到在实际条件下难以达到测试的绝对同步性,即很难实现全部测试节点在相同的物理时刻进行时钟数据采集,本文采用仿真的方式对时间同步方法进行测试与分析。

一般地,针对同步方法的测试主要集中于同步精度、稳定性和同步代价等问题上。

同步精度方面,由 LCPD 的工作原理可知 LCPD 的理论同步精度为计数值 P 的单位分辨率,即 $1/F$ 。

同时,其同步精度的绝对上限值为计算机物理时钟真实周期 $1/f$,因此,在仿真条件下对本文提出的同步方法的同步精度测试意义并不大。

同步代价方面,算法核心基于 RBS 的广播差分思想,单次群同步的算法复杂度为 $O(n^2)$ 。

稳定性方面,由于本文提出的同步方法旨在通过 LCPD 的计数值 P 为分布式系统提供一个可以作

为全网快速同步且稳定的同步依据,因此,需要通过 LCPD 理想状态下的仿真来检验其收敛特性和作为同步依据的可行性,相应的考察值选定为各节点 LCPD 的频率和计数值与均值的偏离数据。

3.3 结果与分析

由以上的测试原理分析,对基于参量差分的分布式时钟群同步方法的测试集中考察其 SADNC 理想状态下频率的收敛特性和计数值与均值的偏离程度。我们的仿真分别针对原始的 SADNC 和改进的 SADNC 进行。基本 SADNC 随仿真进行的变化数据如图 4 所示。

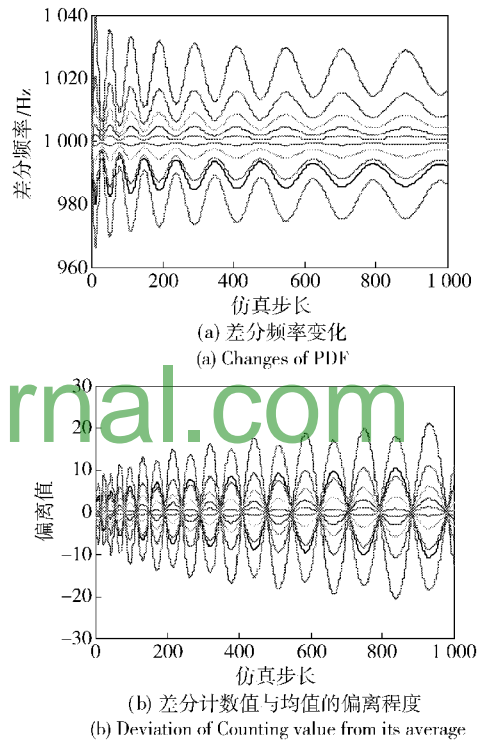


图 4 原始 SADNC 随仿真进行的收敛性能(节点 = 10)

Fig. 4 Convergence Performance of original SADNC for 10 nodes

由图 4 可以看出,基本 SADNC 即便在理想设定的条件下,其差分频率的收敛趋势也令人很不满意,同时,各节点的差分时钟计数值与全网均值的偏离随着时间的累积也会越来越大。很显然,这并不是我们设想的 LCPD。由于各节点差分计数值 P 在真实的物理时刻 t 并没有收敛到一个稳定值,即意味着它无法作为有效的同步依据。由于 LCPD 计数值是由差分频率同时间的乘积累加而得的,且为保证其连续性,不应当直接进行修改。另外通过观察图 4 左边部分,可以发现各节点 LCPD 的频率变化

曲线呈渐增周期性振动的趋势,这表明原始 SADNC 算法各节点差分频率校正效率不高,跳变放缓,且无法稳定在某个值上,算法效率偏低。因此,通过对差分频率振动曲线的波峰和波谷值求平均来加速收敛,是较为可行的。对由此改进的 SADNC 进行仿真测试,相关仿真数据如图 5、图 6 所示。图 5、图 6 分别表示当节点等于 10、20、30 时,改进的 SADNC 算法的良好收敛性能:各节点的 LCPD 在仿真进行到大约 55 步(5.5 s)时即基本达到全网稳定;相应地,各节点的 LCPD 计数值与均值在同一物理时刻 t 的偏离也快速收敛为 0。通过观察图 5 右边部分和图 6,我们发现:相比 20 个节点,当节点为 10 和 30 时,其 SADNC 的收敛性更好;20 个节点的 SADNC 在仿真步长处于 70~90 步时,稳定性有所降低。这表明在较少和较多节点的情况下,SADNC 具有较好的收敛效果与稳定性。

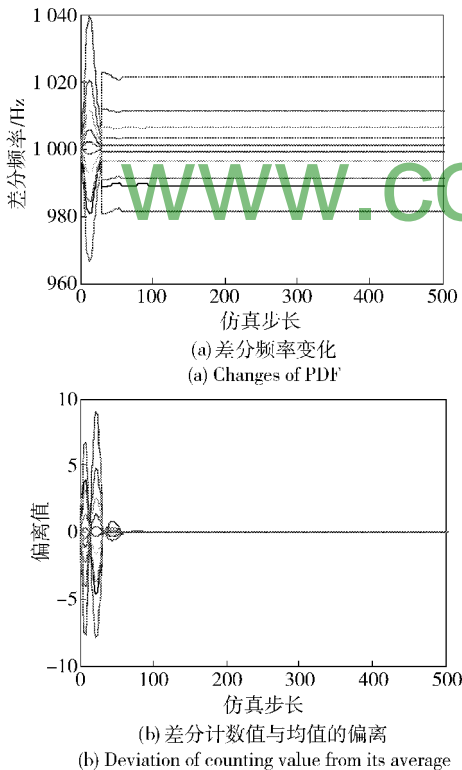


图 5 改进 SADNC 收敛性能(节点 = 10)

Fig. 5 Convergence Performance of improved SADNC for 10 nodes

图 7 仿真数据表明了当仿真步长在 80~100 步和 100~1 000 步时差分时钟计数值与均值的偏离程度。图 7(a)表明:当节点较少 ($N = 10$) 和较多 ($N = 30$) 时,算法的收敛效果明显,在仿真进行到

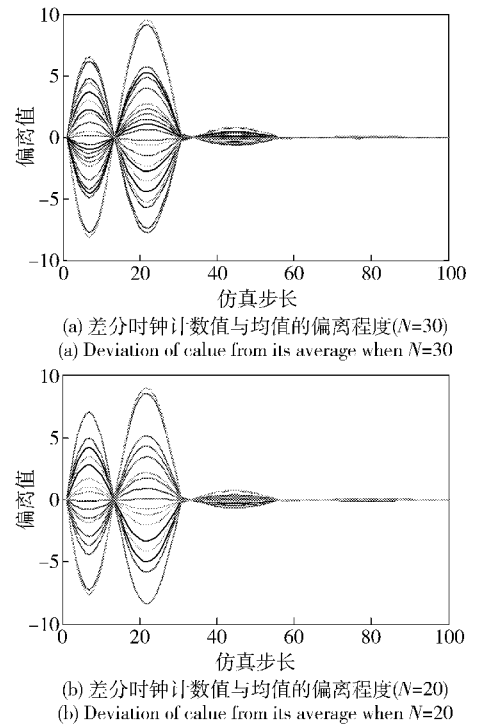


图 6 改进 SADNC 收敛性能(节点 = 20, 30)

Fig. 6 Convergence Performance of improved SADNC for $N = 20, 30$

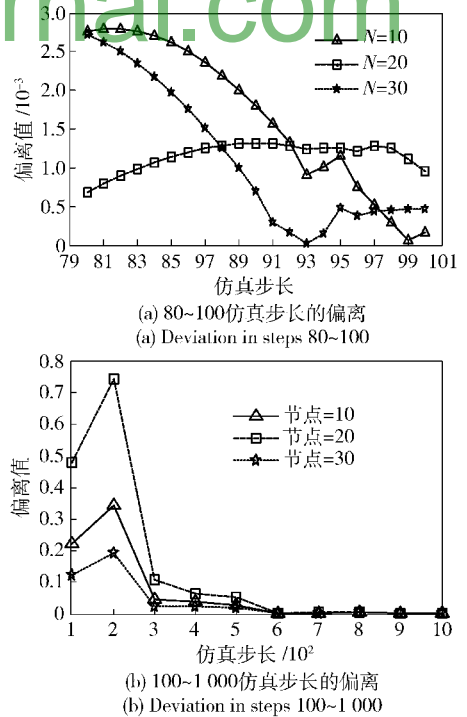


图 7 不同节点数下的 SADNC 同步性能

Fig. 7 Convergence Performance of improved SADNC for different nodes

9.54 s ($N = 10$) 达到 $\leq 10^{-3}$ 的计数偏离, 30 个节点

时只需 8.9 s, 而 20 个节点却需要 10 s。图 8(a) 直观展示了达到此同步偏离误差各节点组所需时间对比。图 7(b) 说明: 随着同步的不断推进 (步长 ≥ 100), 节点数量越多, 算法收敛效果越明显。这表明改进的 SADNC 算法适合大规模分布式节点群的逻辑时钟同步。图 8(b) 表明改进的 SADNC 算法良好的稳定性, 一旦全网节点实现同步后, 所有分布式节点就不再失步, 为仿真推进提供一个稳定的时戳。

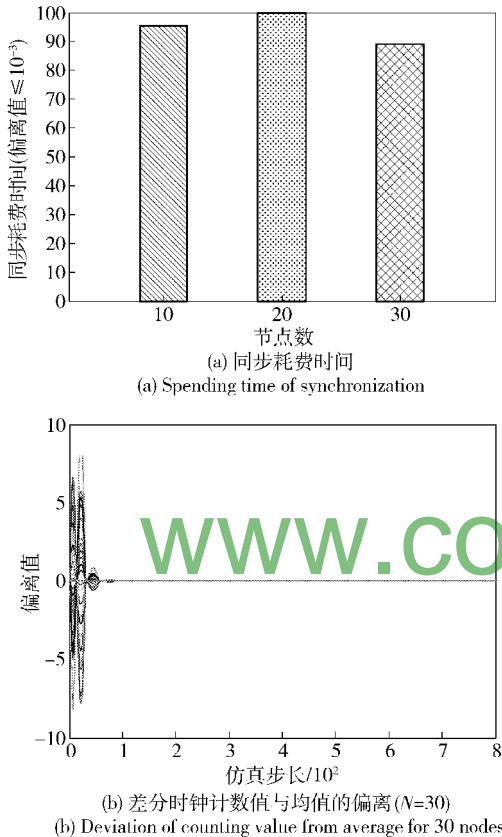


图 8 SADNC 同步算法耗费时间与稳定性分析
Fig. 8 Spending time and stability analysis of improved SADNC

通过前后两组仿真实验对比, 采取极点平均来加速逼近的收敛过程在仿真/理想条件下效果十分理想。改进的 SADNC 具有良好的收敛特性和稳定性, 稳定后的 LCPD 可以作为大规模分布式仿真系统时间同步的有效依据。

4 结论

本文针对新型分布式系统对时钟同步新的要求, 提出一种普适的、通用的基于参量差分的逻辑时钟同步模型 (LCPD) 及其同步算法 (SADNC), 并在

仿真实验条件下, 验证了该模型及其算法的有效性和科学性。LCPD 时钟模型将不确定的网络延迟考虑到差分频率的不断校准过程中, 通过差分频率与逻辑时间的乘积, 递归计算得到前后两次步进的差分计数差值而构建, 是一种不同于传统物理时钟的逻辑时钟模型。SADNC 算法基于 LCPD 模型, 通过对差分频率不断校正与收敛操作, 实现全网所有节点的差分计数值趋于同一稳定值, 以此作为时间同步的依据。仿真实验表明, 改进的 SADNC 算法具有良好的收敛性和稳定性。该算法由于同步机制差异, 不适用于 HLA/RTI 时间管理, 但已应用于自行开发的脱离 HLA 架构的服务化分布仿真支撑环境。

参考文献 (References)

- [1] Wang W, Xu Y, Khanna M. A survey on the communication architectures in smart grid [J]. *Computer Networks*, 2011, 55 (15): 3604 - 3629.
- [2] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing [J]. *Communications of the ACM*, 2010, 53(4): 50 - 58.
- [3] Zhu F, Carpenter S, Kulkarni A. Understanding identity exposure in pervasive computing environments [J]. *Pervasive and Mobile Computing*, 2012, 8(5): 777 - 794.
- [4] Atzori L, Iera A, Morabito G. The internet of things: a survey [J]. *Computer Networks*, 2010, 54(15): 2787 - 2805.
- [5] Strassburger S, Schulze T, Fujimoto R. Future trends in distributed simulation and distributed virtual environments: results of a peer study [J]. *International Series in Operations Research & Management Science*, 2009, 133: 231 - 261.
- [6] Sterzbach B. GPS-based clock synchronization in a mobile, distributed real-time system [J]. *Real-Time Systems*, 1997, 12: 63 - 75.
- [7] Mills D L. Internet RFC-958, network time protocol (NTP) [S]. US: Internet Engineering Task Force, 1985.
- [8] Dana P H. Global positioning system (GPS) time dissemination for real-time applications [J]. *Real-Time Systems*, 1997, 12: 9 - 40.
- [9] Mills D L. Internet RFC-1305, network time protocol specification, implementation and analysis [S]. US: Internet Engineering Task Force, 1992.
- [10] Loy D. GPS-linked high accuracy NTP time processor for distributed fault-tolerant real-time systems [D]. Vienna: Vienna University of Technology, 1996.
- [11] Gotzhein R, Kuhn T. Black burst synchronization (BBS)-a protocol for deterministic tick and time synchronization in wireless networks [J]. *Computer Networks*, 2011, 55 (13): 3015 - 3031.

- [12] Eidson J C, Fischer M, White J. IEEE-1588™ standard for a precision clock synchronization protocol for network measurement and control systems[C]//Proceedings of the 2nd ISA/IEEE Sensors for Industry Conference. Houston: IEEE, 2002: 243 – 254.
- [13] Mills D L. Internet time synchronization: the network time protocol[J]. IEEE Transactions on Communications, 1991, 39(10): 1482 – 1493.
- [14] Lenzen C, Sommer P, Wattenhofer R. Optimal clock synchronization in networks[C]//Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems. Berkeley: ACM Press, 2009: 225 – 238.
- [15] Rhee I K, Lee J, Kim J, et al. Clock synchronization in wireless sensor network: an overview [J]. Sensors, 2009, 9(3): 56 – 85.
- [16] Hoepman J H, Larsson A, Schiller E M, et al. Secure and self-stabilizing clock synchronization in sensor networks [J]. Theoretical Computer Science, 2011, 412(40): 5631 – 5647.
- [17] Ganeriwal S, Kumar R, Srivastava M B. Timing-sync protocol for sensor networks[C]//Proceedings of the 1th International Conference on Embedded Networked Sensor Systems. Los Angeles: ACM Press, 2003: 138 – 149.
- [18] Maróti M, Kusy B, Simon G, et al. The flooding time synchronization protocol[C]//Proceedings of the 2th International Conference on Embedded Networked Sensor Systems. Baltimore: ACM Press, 2004: 39 – 49.
- [19] Geoffrey W-A, Geetika T, Ankit P, et al. Firefly-inspired sensor network synchronicity with realistic radio effects[C]//Proceedings of the 3th ACM Conference on Embedded Networked Sensor System. San Diego: ACM Press, 2005: 142 – 153.
- [20] Hu A, Servetto S D. On the scalability of cooperative time synchronization in pulse-connected networks [J]. IEEE Trans on Information Theory, 2006, 52(6): 2725 – 2748.
- [21] Wikipedia. Synchronization [EB/OL]. (2011-12-13) [2012-07-05]. <http://en.wikipedia.org/wiki/Synchronization>.
- [22] Wikipedia. Critical_section [EB/OL]. (2011-12-03) [2012-07-05]. http://en.wikipedia.org/wiki/Critical_section.
- [23] Fahmy S, Balakrishnan S, ElMekkawy T. Deadlock prevention and performance oriented supervision in flexible manufacturing cells: a hierarchical approach [J]. Robotics and Computer-Integrated Manufacturing, 2011, 27(3): 591 – 603.
- [24] Wikipedia. Semaphore [EB/OL]. (2011-10-20) [2012-07-05]. <http://en.wikipedia.org/wiki/Semaphore>.
- [25] 王涛. 基于混合时钟的轻量级分布式仿真支撑环境研究 [D]. 长沙: 国防科技大学, 2011.
WANG Tao. Research on light weight support environment for distributed simulation based on compounded clock [D]. Changsha: National University of Defense Technology, 2011. (in Chinese)