

基于 NTP 的 ORBUS 时间智能同步系统

丁永红, 张洪胜*

(淮南联合大学信息工程学院, 安徽 淮南 232001)

摘要: 基于 NTP 的 ORBUS 时间同步系统实现网络同步需要静态绑定若干个同步源, 非根同步源存在不稳定特性。为了提高同步网的生存能力, 构建了自组织同步系统的有向图模型, 根据有向图模型设计了 ORBUS 的智能时间同步系统, 提出了智能时间同步系统的核心算法智能层级搜索算法 ISS。最后实现了分布式计算环境中的智能时间同步系统。

关键词: ORBUS; 智能同步; 自组织; ISS

中图分类号: TP393

文献标识码: A

文章编号: 1672-755X(2020)02-0024-05

Intelligent Synchronization Research on ORBUS Time System Based on NTP

DING Yong-hong, ZHANG Hong-sheng*

(Huainan United University, Huainan 232001, China)

Abstract: The ORBUS time synchronization system based on NTP needs static binding of several synchronization sources to achieve network synchronization, and non-root synchronization sources have unstable characteristics. In order to improve the viability of the synchronization network, the directed graph model of the self-organizing synchronization system has been constructed, the intelligent time synchronization system of ORBUS is designed based on the directed graph model, and the core algorithm of the intelligent time synchronization system, that is, the intelligent stratum search algorithm ISS, is proposed. Finally, the intelligent time synchronization system in the distributed computing environment is realized.

Key words: ORBUS; intelligent synchronization; self-organization; ISS

随着科学技术的不断发展, 网络技术的重要性日益明显。网络时间同步作为网络技术的重要环节, 在关乎国民生计的许多领域扮演着重要角色, 如计费系统、交易系统、卫星发射系统、核能利用系统等, 涉及经济、军事、政治等各个方面。网络时间同步可以从网络应用、网络管理和网络安全等方面对其加以影响^[1-2]。

云计算、大数据以及人工智能是目前网络发展的主要方向, 分布式计算需要分布式网络得以支撑, 分布式网络是一个异构平台, 为了适应这种异构平台, 基于 CORBA 规范^[3]开发的 ORBUS 系统应运而生。ORBUS 系统的时间系统确保在分布式环境中时间同步, 开发了基于 NTP 的时间同步系统。由于是基于 NTP 的时间系统, 故存在基于层级的同步结构, 其同步的鲁棒性主要是通过选取多个同步源, 同时提供同步参考量, 即使有一个或者几个节点出现失效, 至少保留一个同步源, 也能够保持同步的可能。如果完全失去同步源则同步系统就处于瘫痪状态, 严重时会导致灾难性后果。这就是本文要解决的问题, 在原先的

收稿日期: 2020-03-05

基金项目: 安徽省教育厅自然科学研究重点项目(KJ2017A584)

作者简介: 丁永红(1975—), 男, 安徽安庆人, 讲师, 硕士, 主要从事分布式计算的研究。

通信作者: 张洪胜(1968—), 男, 安徽淮南人, 副教授, 硕士, 主要从事人工智能的研究。

时间同步系统的基础上构建一个自组织模块,使得所有具有同步模块的异构平台能够自动搜索同步源,形成新的层级同步系统,达到智能时间同步。

1 网络时间协议 NTP 和 ORBUS 时间系统

1.1 网络时间协议 NTP

分布式计算中的时间同步问题是一个重要的研究课题,而目前采用的协议主要是网络时间协议 NTP (Network Time Protocol)^[4]。网络时间协议 NTP 的时间同步的主要过程是:首先设置根节点,主要是一个或多个相互备份的网络服务器,其本地的守时系统与 UTC 保持同步;其次在全球各个地区选择一些合适的节点,使其与这些根节点同步,我们称这些节点为上层时间服务器;最后其他网络上的节点可以选择其中若干个这样的时间服务器作为时间同步源,同时这些同步过的节点还可以为下层节点提供时间同步服务。NTP 协议中时间同步网的组建是在整体部署时人为地确定各个时间同步服务器的 IP 地址,以及同步服务器的数量、请求同步间隔等参数,从而构成一个以根节点为树根的树形多层次化结构网络。

1.2 ORBUS 时间系统

ORBUS 系统^[5]由东南大学自主开发,符合 CORBA 规范。ORBUS 时间系统也遵从其中的时间服务规范,并实现了计数器事件的处理(the Time Event Service)和主机的基本时钟同步(the basic Time Service)等服务,在这几个服务里面定义了五个接口,分别是 TIO、UTO、TimerEventHandler、TimeService 和 TimerEventService,它们实现了基本时间服务和对基本时间服务的创建和管理等操作。在此基础之上引入了 NTP 的时间同步技术,在服务进程 UTO 对象里面设计 ORBUS_set_str_tp_dp_ep() 函数,计算出实现 NTP 时间同步的相关参数,其中包括 strum 层级参数。客户进程绑定若干服务器地址,通过 UTO 对象引用获取服务端包括层级在内的同步所需参数,然后调用多个算法完成同步。

ORBUS 时间系统是 ORBUS 系统的一个组件,ORBUS 系统是可以跨软硬件异构平台的中间件,具有很好的扩展性和可伸缩性,所以其时间同步服务的完善,能够真正在分布式网络环境中实现时间同步。但其同步时间源的绑定存在单点失效问题,所以有进一步对其研究的必要。

2 自组织同步模型

在自然环境中自组织^[6]是一个很常见的现象,比如鸟群的整体移动、生命体的自我修复、病毒的自我复制进化等。自组织是个体在满足特定规则下,异步地、自动地协同工作,整体呈现出自适应的特性,宏观目标是由微观行为作用的结果。我们可以把时间同步网看成是一个自组织系统,认为每个同步节点就是一个 ORBUS 时间同步进程,以 UTC 时间服务器为中心,也称之为根节点,按照特定规则异步地、自动地协同工作,以 UTC 为基准调整各个同步节点的时钟,最终实现同步网的时间同步的宏观目标。基于这个思想,我们要完成 ORBUS 系统时间同步的自组织具体方法如下:每个同步节点执行相同的函数调用,当周围节点状态发生改变时,自动选择最优的参数配置时间同步网,使得同步网具有自主参数优化能力、抵抗恶意攻击的能力,以及自动搜索发现同步源的自愈能力。

为了更好地说明,把时间同步网用有向图 $G = \langle V, NS, WS, E \rangle$ 来表示,其中 V 是同步网中内节点和外节点的集合,表示为 $NS \cup WS = V$, NS 即内节点集合,包括根节点和直接或间接同步于根节点的节点, WS 即外节点集合,包括同步网中目前还没有同步根节点或其他同步服务器节点的节点, $NS \cap WS = \Phi$; $E \in NS \times NS$, 即有 $(u, v) \in E$, u, v 均为内节点, u 是 v 的同步源,有向边 (u, v) 表示节点的同步关系。则有向图 G 中根节点一定是同步源,入度 0;内节点至少有一个同步源,可以作为同步源的候选节点;内节点当失去同步源时转化为外节点;节点的同步源数量即入度,被同步的节点数量即出度。

有向图 G 中任意节点 v 的状态转换可以抽象地表示时间同步系统中节点自组织过程,如图 1 所示:首先网络中普通节点有时间同步需求,需要调用自组织的 ORBUS 时间同步进程,进入有向图,成为外节点;其次自主检测有关状态是否处于正常同步,如果是则继续同步,如果否则外节点 v 在 G 中筛选出合适的内节点集合 $BEST(v)$, 开始搜索同步源;然后 v 修改自己的 ORBUS 时间系统同步参数,与 $BEST(v)$

同步,此时外节点成为内节点,随后内节点 v 在 ORBUS 时间系统中执行时间同步, v 同步于 $BEST(v)$ 中的最佳同步源时钟;最后由于同步系统进程终止或者所有同步源的退出,导致节点 v 转为外节点而退出有向图。

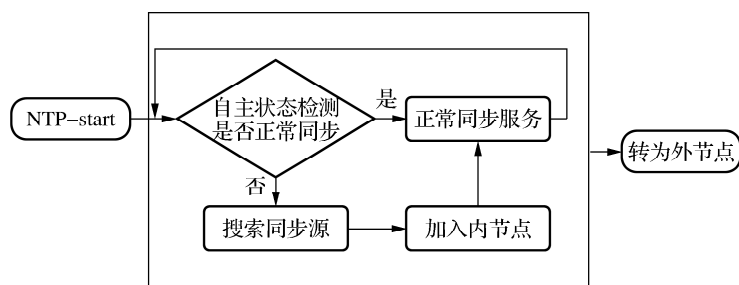


图 1 时间同步系统中节点自组织过程

3 分布式系统资源搜索算法

3.1 分布式系统资源搜索算法分类

分布式系统资源搜索或者资源发现是一个传统问题。自组织系统与资源搜索密切相关,自组织系统能否自愈取决于搜索方法。分布式资源搜索包括结构化网络资源搜索和非结构化网络资源搜索,而非结构化网络资源搜索还可以进一步分为盲目搜索(Blind Search)和信息搜索(Informed Search)。Flooding 算法和 Random Walk 算法是盲目搜索的两个特例。Flooding 算法就是广播。Random Walk 算法,一般随机找寻 k 个邻居,称之为 k -Random Walker 算法,即随机向 k 个邻居节点发送搜索消息,收到搜索请求的节点也随机向其 k 个邻居发起搜索,有效降低了搜索消息量,但在 TTL 一定的情况下会遗漏部分有效资源, k 的经验值为 16~64。信息搜索是根据任何能提高搜索成功率的信息进行定位。APS(Adaptive Probabilistic Search)自适应概率搜索是基于 k -Random Walker^[7] 的一种信息搜索算法,其资源的定位是依靠节点在搜索某类资源时记录的参数,其对应缓存在本地索引列表里。对于信息搜索算法具有以下共性:第一,对资源分类,方便资源定位;第二,资源定位信息要随资源变化而动态更新;第三,网络中所有节点需要预存目标资源的子集信息。ORBUS 时间同步系统通过 APS 提高同步精度。

3.2 智能层级搜索算法

在信息搜索算法中,可以根据资源分布的层次来确定资源的定位,ORBUS 时间同步系统中,各个节点都有一个重要信息就是层级,在搜索资源时也就归结到搜索层级值,把希望获取的层级值称为层级期望 SE 。这种搜索算法定义为智能层级搜索算法 ISS(Intelligent Stratum Search)。

有同步需求的一个普通节点,对同步网一无所知,首先是激活时间同步进程,然后通过两种途径接入同步网成为外节点:第一,如果同网段内有内节点,则广播请求信息,内节点做出响应,建立连接;第二,如果同网段内没有内节点,本地索引列表里需要预置有可靠连接的内节点集,一般为根节点地址,与内节点集建立初始连接。这里内节点集一定在安全区,也是同步网有效的前提。外节点搜索资源的定位信息层级期望 SE 是寻求同步的目标内节点的重要参数,而同步网的层级关系是逐层递减的,在建立连接后,为了降低通信成本和 CPU 开销,请求消息转发选择时只向离层级期望 SE 值近的节点转发。

在时间同步网络中,各个节点既是请求端,又是服务端,是对等关系,所以分析任意节点的请求转发过程都是一样的。下面分析节点 x 收到请求消息 Q 的情况,假设搜索内节点的层级期望是 Q . SE , x 的内节点集是 $NS(x) = \{b_1, b_2, \dots\}$,则算法伪码:

Algorithm ISS(Q)

$NS(x) = \{b_1, b_2, \dots\}$ //定义内节点,即同步系统内正常同步节点

$if(x.T = Q.SE) answer(Q, NSA), return endif$ //如果 x 的层级就是期望值,返回 x 的 IP

$if(Q.TTL = 0)$ 丢弃 Q , return endif //如果 TTL 倒计时归零,终止转发

$if(redundancy_kill)$ 丢弃 Q , return endif //如果接收到的消息重复,丢弃

$Q.TTL - 1$ //生存期 - 1

$Z(x) = \{bj | abs(bj.T - Q.SE) \leq abs(x.T - Q.SE), bj \in NS(x)\}$ //在 x 的内节点集里,查找比自己在层级

```

上更接近于请求节点的层级期望值的内节点
if  $Z(x) \neq \Phi$  for  $\forall y \in Z(x), passto(Q, y)$  //如果找到更接近的层级值的节点集合,转发 Q
else
for  $\forall y \in NS(x), passto(Q, y)$  //以上没有,为保证搜索的成功率则向所有内节点广播
endif

```

其中, $passto()$ 为消息转发; $abs()$ 为绝对值; $x.T$ 为 x 的层级; $reduncy_kill$ 为冗余消息检测; $Q.NSA$ 为 x 的 IP 地址; $answer(Q.NSA)$ 为回答请求消息。

4 ORBUS 时间智能同步系统的设计与实现

4.1 智能同步系统的设计

智能同步系统是一个自组织系统,在分布式计算中是一个再同步的过程,要完成系统自愈需要设计请求同步模块和被请求同步的模块,另外还需要一个管理全局的控制模块。控制模块主要控制节点的四种状态的转换。第一,正常状态。此时属于有向图中的内节点,与同步源保持正常同步,定期检测同步源的有效性,如果同步源丢失进入等待状态。第二,等待状态。机器的重启、网络掉线等可能引起同步源的暂时丢失,为了降低同步代价,设置计时器 A,等待同步恢复,因为在计时器 A 复位之前也可能同步源自愈,网络恢复正常,如果计时器 A 复位,转至搜索状态。第三,搜索状态。设置计时器 B,启动搜索同步源模块,选择同步源模块,实现自动配置,找到同步源,则转到正常状态,如果计数器 B 复位,期间没有找到同步源,进入根搜索状态。第四,根搜索状态。设置计数器 C,启用本地索引列表里面的根节点,帮助本节点获取层级期望 SE 的同步源,找到期望同步源,返回正常状态,如果计数器 C 复位前没有找到期望同步源,重回等待状态。

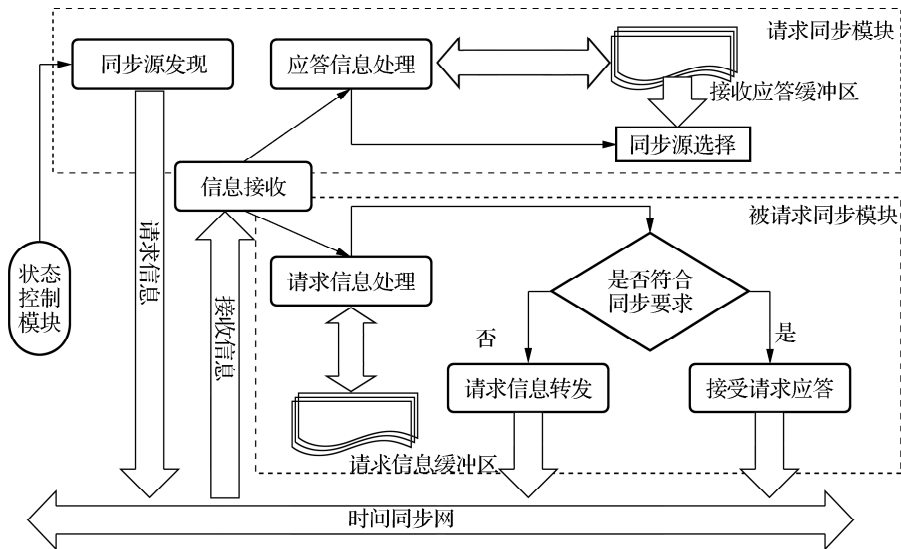


图 2 智能同步系统的设计流程

控制模块除了协调四种状态的转换之外,还要负责监测同步源、启动同步模块、主导各个模块之间的配合、修改相关同步参数,以及与 ORBUS 时间系统交互操作。智能同步系统的整个设计流程如图 2 所示。当同步源发现模块启动后,调用 UTO 对象绑定广播地址变量,获取所有同步源的应答信息,经处理推入应答信息缓冲区,调用同步源选择函数,直至有效同步源队列满载,得到最优同步源完成时间同步过程。智能同步系统为了同时具有为其他节点服务的机制,被同步模块设计为:当节点收到其他节点的绑定连接信号时对请求的信息进行处理,如果合法且第一次到达,即推入请求信息缓冲区,以备后续信息的比较,到达的信息依据请求层级期望 SE 调用智能层级搜索算法 ISS 函数,如果符合期望值,则本节点发回 IP 等参数,接收同步请求,如果不符合,则向离期望值更近的节点转发,为下一个对等内节点做好准备。

4.2 智能同步系统的实现

本系统是在 NTP 的 ORBUS 时间同步系统^[8]的基础上进行设计和开发,我们把主控函数以及智能同步系统的同步模块和被同步模块中的相关函数和过程都添加到 ORBUS 系统的 UTO 对象中,原同步系统的客户端需要绑定的固定服务器地址由同步源选择函数输出 IP 提供,其他 NTP 的同步机制正常执行,从而完成智能同步的应用。

同步源端部分代码如下:

```

universal_time()函数中的主要操作:
query_process();//调用请求处理过程
ORBUS_set_IP(NSA);//如果当前节点层级值符合请求节点层级期望 SE,则从本地读入 IP 地址给 NSA(参见 4.2 节)
提供客户端调用的接口:
universal_time();//完成同步源端操作的主要函数
ORBUS_get_IP(p1);//获得同步源 IP 信息
请求端部分代码:
timeServVar1=CosTime::TimeService::bind("TimeSourceTest",p1);//绑定同步源,p1 为同步源地址
timeServVar->universal_time(); //调用同步源的 universal_time()函数
ORBUS_get_IP(p1);//获得同步源 IP 信息,进入有效同步源备份

```

在分布式环境中实现同步源的同步,是依托 ORBUS 时间系统的 UTO 对象完成,同步源端通过 ORBUS_set_IP(NSA)提供符合请求端期望的本地 IP,请求端通过 `timeServVar1=CosTime::TimeService::bind("TimeSourceTest",p1)`;逐个绑定本网段的所有准同步源,调用同步源端的 ORBUS_get_IP(p1)获取所有符合条件的同步源 IP,请求端通过同步源选择模块得到唯一同步源 IP,以及邻居节点地址、根节点地址等信息。此绑定动作在基于 NTP 的 ORBUS 时间同步系统实际同步之前,受主控模块状态控制启动,p1 地址根据状态不同选取本网段内的所有地址,然后基于原 ORBUS 的时间同步系统完成时间同步。

5 结 语

基于 NTP 的 ORBUS 的智能时间同步系统,除了本地索引列表里备份根节点地址以外,不需要任何人工参与,能有效重构网络结构,用户无需知道机器的同步源。而且原系统要求高精度同步,需要提供多个同步服务器,智能同步系统只需同步选择最优的一台服务器,一定程度上节省了服务资源。本系统还可以根据同步源的 CPU 空闲程度、同步进程的活跃程度来构造层级期望,根据同步源的请求频率等来判定同步能力,从整体上提高智能同步系统的精度。

参考文献:

- [1] Jean-Marc B. Time synchronization over networks using convex closures[J]. ACM Transactions on Networking, 2000, 8(2): 265 - 277
- [2] 包秀国. 开放层次式系统的生存性增强技术研究[D]. 哈尔滨: 哈尔滨工业大学, 2004
- [3] 汪芸. CORBA 技术及其应用[M]. 南京: 东南大学出版社, 1999: 68 - 80
- [4] Mills D. Network time protocol (version 3): specification, implementation and analysis[R]. Newark: Network Working Group Report RFC-1305, University of Delaware, 1992
- [5] 东南大学计算机科学与工程系网络教研究. 奥贝软总线 ORBUS 1.1 使用手册[M]. 南京: 东南大学出版社, 1999
- [6] 包秀国, 胡铭曾, 张宏莉. 一种自组织时间同步网[J]. 通信学报, 2004, 25(1): 150 - 156
- [7] Lv Q, Cao P, Cohen E, et al. Search and replication in unstructured peer-to-peer networks[C]. Princeton: Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 2002: 258 - 259
- [8] 丁永红, 徐强, 汪芸. 基于 NTP 的 ORBUS 时间同步系统[J]. 东南大学学报(自然科学版), 2006, 36(3): 477 - 482

(责任编辑: 湛 江)