分布式容错系统的同步化策略

欧阳珣 李 榕 (华中理工大学 计算机学院 430074)

摘要:分布式容错的时间同步化系统虽然在我们国内还没有得到广泛的应用,但是随着计算机技术的发展,分布式环境中各计算机的时间同步化问题显得越来越重要。目前国际上有多种时间同步化的软件,它们的运行机制不同且各有特点,本文就目前流行的几种分布式容错系统的时间同步化策略进行了较详细的研究,并得出了相应的结论。

一、引言

随着网络技术的发展,各类数据传输要求严格同步,因此在数据传输中的同步和控制等问题上如何精确地掌握错误模式与影响程度是一个相当重要的问题。但目前分布式系统的时间同步存在着技术上的困难与限制。

1.无法达成真正的时间同步化。因为在分布式环境中,每个节点都有自己的计时器,而每个计时器的进行速率又不可能全部一致,很难达成所有系统节点时间上的绝对同步;

2.分布式系统中的时间同步化往往要参考其他节点的时间来进行修正,而要取得参考时间的资料,必须要通过通信网络的传递。实际通信网络的信息传递一定会有时间上的延迟(propagation delay),这样在时间同步上我们又增加了考虑的因素。

3.参考时间在分布式环境中传输的正确性也会严重影响系统时间同步。

为了解决上述问题,许多科研人员投入了相当大的力量并制定了有关的协议和规则,其中 NTP(Network Time Protocol)和 SNTP(Simple Network Time Protocol)就是应用最多的两种协议。

而随着计算机网络技术的发展与主从结构(client/server)观念的提倡,比集中处理(centralized processing)更有弹性,成本更低的分布式处理结构(distributed processing)就为众多系统所采用。因此"分布式系统容错"(fault tolerance in distributed systems)也就提到议事日程上来。分布式系统中主要的元件如:处理器(processors)、通信链路(communication links)、计时器(clock),资料储存装置(nonvolatile storage)及软件等。而这些都是单一独立的元件,并且这些典型的元件通常不会考虑容错。但事实上,在分布式系统中,这些元件都是以相互合作的方式在运作,不可能脱离其他元件而独立,因而这些元件的容错能力决定了分布式系统中的容错能力,也是分布式系统容错主要的研究范畴。而"时间同步化"(time synchronization)正是分布式系统容错中相当重要的课题。

二、分布式同步化策略

"时间同步化"的相关研究及论文发表和论述,最早出现于 1970 年左右,当时大部分研究都是与实时系统(real time system)有关,因为许多实时系统所需的控制机制,如投票(Voting)与同步回复(synchronized)等等,都假设在相当严格的时间同步化之下,其中之一的论述就是 Ellingson E. 和 Kulpinski R. 两人发表的《Dissemination of System Time》(1973)。从此之后时间同步化渐渐在分布式计算(distributed computing)领域中展现其重要性,并受到广泛的注意。

时间同步化的算法基本上可分为:"硬件时间同步化"(hardware time synchronization)和"软件时间同步化"(software time synchronization)两种。硬件时间同步化需要特殊的硬件设备来完成,软件时间同步化则不需要额外特殊的硬件设备。

关于分布式容错的时间同步化策略与算法相当多,根据 Manfred J. 和 Douglas M. Bough 的分类,基本上可分为以下三大类型:

1.主仆式及时间服务器的时间同步化策略

主仆式及时间服务器时间同步化策略是分布系统中,最直接最简单,也是最广泛被应用的方法。其主要策略是在系统中建立一个或若干个具有高可信度与有效度的时间服务器,而系统中其他的节点则透过通信网路与服务器直接地撷取正确的时间来修正各自的时间,进而达到整个系统中各节点时间上的同步。

主仆式时间同步化策略最主要的应用表现在 NTP (Network Time Protocol)及其相关的 SNTP(Simple Network Time Protocol)协议的制定,而 NTP 与 SNTP 也成为当今网际网络中许多时间同步化机制的标准。另外运用时间服务器式的时间同步化策略关键就是 Novell Netware 4.01 网管系统,它也是采用的主仆式结构。Netware 4.01 将其系统区分为内部时间同步化(internal time synchronization)与外部时间同步化(external time synchronization),主要是建立几种不同类型的时间服务器,依其不同的功能而分层管理,如图 1。主仆式时间服务器式时间同步化策略优点是原理和设备简单,但缺点是容错能力低、集中式管理易受瓶颈效应(bottle－neck effect)影响等。

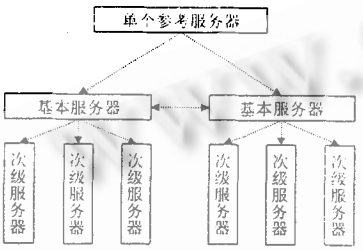

图 1　Netware 4.01 时间服务器式时间同步化结构图

2.拜占庭协议式时间同步化策略

拜占庭协议(Byzantine Agreement/Consensus)在分布式系统容错领域中是相当基本且重要的容错技术,它的算法就是在系统中由一节点充当发送者(transimitter),由它发送一特定信息给所有的结点,非错误节点接收到此信息一经处理后会将它再正确地广播给其他剩余节点,每个节点透过彼此信息的交换,通过多数决定(majority)方式可以知道哪些节点发生错误,之后各节点再轮流当发送者,不断递回交换信息,直至所有错误被找到为止,如图 2。

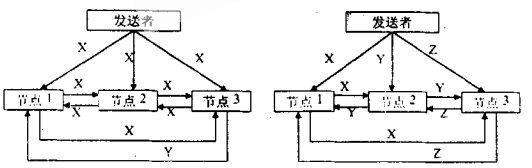

图 2　拜占庭协议示意图

拜占庭协议具有如下特性:(1)它适用于侦测所有分布式系统中的任何错误(arbitrary fault)包括信息遗失或节点错误等等;(2)它要执行 m 次递归程序以决定 m 个错误;(3)它的容错能力可以达总节点数三分之一的错误个数,每个节点对外连接数需大于错误个数两倍。

拜占庭协议也被使用于时间同步化的策略上,其优点为:(1)容错能力强;(2)适用于任何型态的系统错误;其缺点是:(1)需要相当多的信息交换量(message exchange),从而加重了整个网络负载;(2)适用于特定信息的交换,对于时间同步化中需交换非特定的参考时间信息,有其应用上的限制。

3.收敛函数式时间同步化策略

收敛函数式时间同步化策略的算法为搜集系统中各个节点的参考时间,通过一个有效的收敛函数计算所收集的参考时间并从中决定同步化时间,再根据其结果来调整各自的时间。它的基本步骤如下:

(1)从其他的节点收集参考时间值;

(2)考虑影响因素进而估算这些时间估计值,并获得一正确时间值;

(3)运用一收敛函数计算这些时间估算值,并获得一正确时间值;

(4)根据正确时间值来修正各自的时间。

综合以上论述,我们认为主仆式及时间服务器时间同步化策略是比较好的一种策略。在我们设计的简单网络时间协议(Simple Network Time Protocol,简称 SNTP)就是采用的主仆式及时间服务器时间同步化策略。

三、网络时间通信协议的设计

为达到网络的同步,我们的设计采用 SNTP,要求达到的精度为 0.5 秒。SNTP 是采用主仆式同步策略,而且 SNTP 来源于 NTP,即网络时间协定(Network Time Protocol,简称 NTP)是目前在国际网络中使用最广泛的时间同步化机制。它是由 David L. Mills 于 1985 年首次提出,后来历经修改有《NTP Serwsion 2,1989》《NTP Version 3,1992》等版本,最新的版本是《NTP Version 4,1996》。其主要的特点是向使用中的客户端或服务器端的计算机与远端的参与时间源,例如无线电广播、卫星或是数据传送,提供一种机制(mechanism)与结构(structure)进行时间同步化。它能使得调整后客户端的时间准确度在局域网络(LAN)中达到几毫秒,在广域网络

(WAN)中达几十毫秒。目前许多国际网络(Internet)的作业系统,如 UNIX 等,都是采用 NTP 作为其时间同步化机制的标准。

NTP 最主要的目的,在于建立一种与单一或若干个可信赖的参考时间服务器相连接时彼此达成时间同步化时的通信协定,以期时间同步化的结果能达到高精度与可信度的要求,NTP 是建立在 Internet Protocol 与 User Datagram Protocol 之上,属于一种非通路连接(Connectionless)的传输机制。NTP 是以子网络同步化(subnet synchronization)的方式来进行的,如图 3 所示。节点表示子网络服务器(subnet server),它们是以阶层式管理来规划时间同步化,箭头代表参考时间信息的传递方向,线段表示备份连接。如图 3(a)中当阶层数 2 的节点与阶层数 1 的节点连接发生错误,连接到该节点阶层数为 3 的所有节点必须通过备份连接到另一阶层数 2 的非错误节点,该节点也必须都降为阶层数 3,整个结构重新规划,如 3(b)所示,所以基本上 NTP 的子网同步化是一种树状结构:

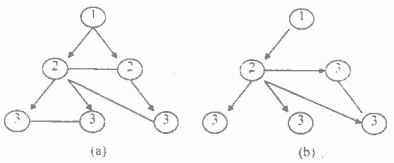

图 3 NTP 的子网同步化图

整个 NTP 结构模式如图 4 所示。

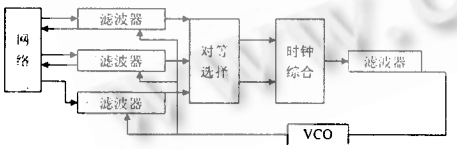

图 4 网络时间协议模式

NTP 虽然提出子网同步化阶层式结构来弥补其主仆式时间同步化容错能力的不足,但是树状结构的容错能力不够,而收敛函数式时间同步化有较佳的容错能力,但是却因收敛函数式时间同步化策略需要大量的参与时间信息,因此在效率上被限制而难以达到,为求 NTP 的设计简单化,故 NTP 采用主仆式时间同步化策略。

NTP 网络时间协议作为一种主仆式同步化策略,具有精度高的优点,达到 1/50ms。它采用工程算法来提高自身的同步精度,得以用在 GPS 全球定位系统中。但目前大多数使用环境中 NTP 显得太复杂,且不需要达到如此之高的精度。

SNTP 则是一种很好的选择,它是对 NTP 的一种简化,它所具有的精度为≤1 秒,并且 NTP 可以兼容 SNTP,换言之它们两者之间是透明的,这样使用 SNTP 的客户和服务器也可以用使用 NTP 的服务器和客户相互通信。

我们设计的 SNTP 系统是用在分布式环境中的,它由服务器和客户端组成。由客户端向服务端发出同步请求,服务端收到请求后立即向客户端传递时间信息。在我们设计中,我们采用时间戳的方法,即将客户端发出请求的时刻,服务端收到请求的时刻,服务端响应的时刻和客户端收到信息的时刻分别定义为 T_1,T_2,T_3 和 T_4 四个时间戳,在信息传递的时候协议记下时间戳。由时间戳可以确定时间传输误差 $\delta(\triangle)$和时钟误差 $\theta(\Theta)$。

$$\delta=(T_4-T_1)-(T_3-T_2)$$

$$\theta=[(T_2-T_1)+(T_3-T_4)]/2$$

我们在设计中采用反复记录时间戳,求平均值的方法来提高精度。

结论:

通过研究分布式时间同步化策略,并将其应用在我们设计中,达到了既定的 0.5 秒的精度。

参考文献

[1] Mills, D. L. Network Time Protocol(NTP). DARPA Network /working Group Report RFC-958, M/A-COM Linkabit. September 1985.

[2] Mills, D. L. Network Time Protocol(Verstion1)-specification and implementation. DARPA Network Working Group Report RFC-1059, University of Delaware, Jyly 1988.

[3] David L. Mills," Simple Network Time Protocol (SNTP)", Network Working Group Report RFC-1361, August 1992

(来稿时间:1999 年 8 月)