基于 NS 的分布式并行网络模拟器

李 越,钱德沛
(西安交通大学计算机与工程系,陕西西安 710049)

摘 要: 网络模拟是测试和评价网络协议性能的重要手段.随着计算机网络的发展,网络模型变得更大更复杂,使网络模拟往往要运行很长时间.本文以在网络研究领域广泛使用的网络模拟器 ns 为基础,应用并行离散事件模拟技术(PDES)修改其串行事件调度机制,扩展网络模型库,实现了 ns 在工作站机群环境下的并行执行,为大规模网络的模拟提供了支持.

关键词: 网络模拟; 并行离散事件模拟; ns
中图分类号: TP393.01 **文献标识码**: A **文章编号**: 0372-2112 (2004) 02-0246-04

A Distributed Parallel Network Simulator Based on NS

LI Yue, QIAN De-pei
(*Department of Computer Science and Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi* 710049, *China*)

Abstract: Network simulation is widely used in network research to evaluate the performance of network protocols. With the development of computer networks, network models grow in size and complexity, so execution time of simulation can be unbearably long. A parallel network simulator which runs on network of workstations is presented, the parallel network simulator is designed based on the popular sequential network simulator ns. Parallel discrete event simulation (PDES) techniques are applied to modify the event scheduling mechanism in ns. Some necessary extensions are also made to ns model library. Performance measures including speedup and memory consumption are evaluated at last.

Key words: network simulation; parallel discrete event simulation; ns

1 引言

随着计算机网络的快速发展,在网络模拟中建立的网络模型变得更大更复杂,使得大规模网络的模拟往往要运行很长时间.在主流工作站上对一个包含几百个节点的千兆网进行包一级的模拟,真实网络几分钟的行为就需要模拟器运行几个小时.并行离散事件模拟技术[1](PDES)将大的网络模型划分为子模型并行执行,从而大大提高网络模拟的运行效率.

NS[2](Network Simulator)是源代码开放的串行网络模拟器,被广泛地应用在网络研究领域,以测试和评价协议的性能.作为 973 海量信息系统概念性试验和验证平台的一部分,我们应用 PDES 技术修改 ns 的串行事件调度机制,扩展 ns 的网络模型库,实现了 ns 在工作站机群环境下的并行执行.从而使已有的各种基于 ns 的网络模型可以方便地重用.本文首先介绍并行离散事件模拟的概念,然后给出基于 ns 的并行网络模拟器的体系结构和实现中的若干关键技术,最后是性能分析和结论.

2 背景

串行离散事件模拟保存一个事件表,其中的每个事件具有一个时戳.事件调度器不断从事件表中获取最小时戳的事件处理.保证了所有事件按照时间顺序执行.

PDES 可以看成是一组逻辑过程 LP (Logical Process) 的集合,每个 LP 对应一个串行离散事件模拟.事件表按照某种规则划分给每一个 LP.在运行模拟时,如果 LP 还是简单地按照串行离散事件模拟那样处理事件,就有可能造成事件执行的乱序,我们看图 1 所示的例子.

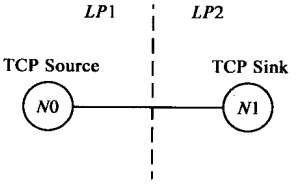

图 1 包含两个节点的网络模型

网络模型由两个节点 $N0$ 和 $N1$ 构成,并分别划分给两个 $LP(LP1, LP2)$. TCP 数据源绑定在 $N0$ 上,宿绑定于 $N1$. 假定 $LP1$ 在模拟时间 1.1 执行一个向 $N1$ 发送数据包的事件,执行

收稿日期:2002-12-25;修回日期:2003-10-13
基金项目:973 国家重点基础研究项目(No. G1999032710);863 高技术项目(No. 2001AA112120)

完成后会产生一个新的超时事件,插入到事件表中,比方说时戳为 8.1. 如果这时 $LP1$ 象串行模拟那样马上执行该事件,势必又会产生时戳为 15.1 的超时事件,这将永远继续下去. 显然与实际的网络行为不同. 事实上正确的事件执行顺序应该是 $LP2$ 收到来自 $LP1$ 的事件后发送应答,该应答事件会在 8.1之前被 $LP1$ 执行,从而将超时事件从事件表中删除.

这是 PDES 为保证模拟结果的正确性必须要解决的问题,即时间同步问题 - 每个 LP 都必须按照时间顺序处理事件[1]. 时间同步的方法分为保守同步和乐观同步两类.

保守同步基于阻塞(blocking),基本思想是必须先确认某事件是"安全"的,然后才能处理该事件."安全"是指不会收到比该事件具有更小时戳的事件. 早期的保守同步机制是 null message 算法[3],它假定 LP 之间具有固定不变的连接结构,通信系统是 FIFO 的. 并采用发送 null message 避免死锁. 另一种保守同步机制基于并行计算模型 BSP(Bulk Synchronous Parallel)[4],用全局的同步操作来决定可以"安全"执行的事件,避免了发送 null message. LP 反复执行一个包含以下两步的处理过程:

® 全局同步,计算时间下限 T_b,即将来可能会收到消息的最小时戳;

® 处理所有时戳不超过 T_b 的事件.

全局同步采用路障机制(barrier). 当某一 LP 进入路障后,它会一直阻塞到所有其他 LP 都进入路障为止. 当所有 LP 进入路障后,启动计算时间下限 T_b. 为了计算 T_b,时间提前量(lookahead)的概念非常重要,含义是从事件在某个 LP 上被处理,到它对另一个 LP 产生影响的一段模拟时间. 在图 1 中,$LP1$ 和 $LP2$ 之间的 lookahead 可以认为是数据包从 $N0$ 传到 $N1$ 经历的模拟时间. 如果 $LP1$ 此时的模拟时钟是 T,那么据此 $LP2$ 可以计算出 $T_b = T$ + lookahead. lookahead 是根据模拟的特定模型决定的.

乐观同步的提出是为了避免保守同步中的阻塞,基本思想是每个 LP 可以"大胆"向前执行,如果发生错误(收到时戳小于当前模拟时钟的事件)用回滚机制(rollback)纠正. 常见的乐观同步机制是 TW(Time Warp)算法[5].

保守同步的缺点在于没有最大限度地利用模型的并行性,乐观同步则需要存储大量的模型状态信息. 二者适用于不同的应用. 通常前者更适合具有很好 lookahead 的模型.

3　并行网络模拟器的体系结构

并行网络模拟器以 ns 为基础,体系结构如图 2 所示. ns 的模型库中包含了大量的网络协议模型和流量模型,与实际网络类似,模型库分为物理链接层、网络层、传输层和应用层. 建模者利用模型库生成各种网络模型进行模拟. 在并行网络模拟器中,将网络模型按拓扑结构划分,每个 ns 的实例只模拟整个网络模型的一部分(即子模型). 最优的划分策略应该兼顾 lookahead 和负载均衡. 为了模拟更大规模的网络,每个 ns 中只定义被模拟的那一部分模型,所以必然需要扩展 ns 现有的模型库. 比如在图 2 中,连接节点 $R1$ 的远程链路不可能利用现有的模型库生成,所以需要在物理链接层增加远程链路模型. 同样需要对网络层和传输层做相应的扩展. 这方面将在第 5 节做详细的论述.

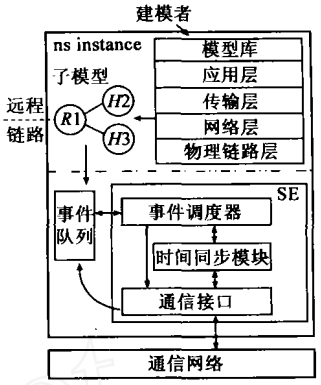

图 2　并行网络模拟器的体系结构

并行网络模拟器的事件调度过程由模拟引擎(SE)完成,它包括事件调度器、时间同步、通信接口三个模块,如图 2 所示. 事件队列包含本地事件和外部事件. 本地事件由本地子模型产生,外部事件是其他 ns 实例发送的事件,通过通信接口接收. 事件队列中的事件按时戳先后排序,事件调度器抽取时戳最小的事件,调用相应的处理函数,在处理过程中,可能向其他的 ns 实例发送消息. 当事件调度器发现没有安全事件可以处理时,会调用时间同步模块的相关函数启动路障同步(barrier synchronization)并计算时间下限 T_b. 时间同步模块将计算结果返回,事件调度器开始新一轮的事件调度、处理过程. 在第 4 节将论述具体的细节. 通信接口是一组负责不同 ns 实例之间交换消息的 API 函数,能够保证可靠通信并且避免通信死锁. 需要区分事件消息和同步消息的概念,前者是向其他 ns 实例发送外部事件的消息. 后者是在路障中为了同步发送的消息.

4　路障同步和计算时间下限 T_b

为了计算将来可能会收到事件消息的时戳下限 T_b,需要考虑过渡消息(transient message)的影响. 过渡消息是指已经发出但尚未到达目的地的事件消息. 如果在路障同步中没有考虑过渡消息,则有可能发生这样的情况:一个 LP 完成路障同步并计算得到 T_b,开始了新一轮的事件处理. 这时收到某个 LP 在进入路障同步之前发送的事件消息(该消息是在已完成的 T_b 计算中没有考虑到的过渡消息),其时戳有可能比当前的模拟时钟小,这是 PDES 保守同步机制不允许的.

此外由第 2 节 LP 的两步处理循环可以发现,在一个循环中发送的事件消息只有在下一个循环中才会被处理. 基于以上的考虑,我们设计并行网络模拟器的事件调度过程如下,其中 N_i 是 LP_i 中下一个待处理事件的时戳,LA_i 是 LP_i 的时间提前量.

```
SE Execution Loop:
while (there are events to process) {
    receive messages generated in previous cycle;
    catch transient messages;
    Tb = min(Ni + LAi);
    process events with time stamp <= Tb
}
```

路障同步可以采用不同的算法,如广播型,树型或蝶型[6].在实现中,路障同步过程通常和过渡消息的捕获同时进行.文献[7]给出了一种检测系统中是否存在过渡消息的方法.LP 维护两个计数器,一个记录其进入路障之前发送事件消息的个数,另一个记录已经收到事件消息的个数.LP 之间交换计数器信息.当所有的 LP 进入路障,并且发送数的总和等于收到数的总和时,表明所有的过渡消息已经被捕获.缺点是 LP 只有收集到所有的计数器信息后才能判断是否存在过渡消息.如果发送计数器和接收计数器不相等(即存在过渡消息),那么所有 LP 只能重复该过程.如果过渡消息很多,同步的开销就非常大.我们在蝶型路障同步中用向量计数器替代上述算法的发送接收计数器,保证经过两轮处理所有的过渡消息能够被捕获.描述如下:

LP_i 维护一个向量 MV_i,如果 LP_i 向 LP_k 发送一个事件消息,则计算 $MV_i[k] = MV_i[k] + 1 (i \neq k)$,如果 LP_i 收到一个事件消息,计算 $MV_i[i] = MV_i[i] - 1$.算法分为两轮,在每一轮中按照蝶型方式[6]交换 MV_i.以 8 个 LP($LP_0, LP_1 \cdots LP_7$)为例,第一轮中 LP_3 进入路障后,首先向 LP_2 发送 MV_3,并等待 LP_2 的回复.收到 MV_2 后计算 $MV_3 = MV_3 + MV_2$.接下来 LP_3 与 LP_1, LP_7 进行类似的操作,完成第一轮的处理.第一轮结束以后,所有 LP 进入路障,不会再发送事件消息.在第二轮中,如果 $MV_i[i] > 0$,说明 LP_i 有过渡消息未收到,LP_i 会等待直至 $MV_i[i]$为零.然后将 MV_i 清零,并与相关 LP 交换 MV_i.第二轮结束后,所有 LP 的 MV 被清零,所有过渡消息被捕获.从上面的过程可以看出不论有多少过渡消息,同步消息数最多是 $2n\log n$,其中 n 是 LP 的数目.

计算 T_b 可以在上述算法的第二轮进行,即每个 LP 的下一个待处理事件的时戳和时间提前量之和的最小值.在目前的实现中,时间提前量是一个固定的常数,即不同网络子模型之间远程链路延迟的最小值加上包长与带宽的商.将在第 5 节论述远程链路模型.

5 模型库扩展

5.1 物理链接层

NS 模型库的物理链接层主要包含节点模型和链路模型,用于构造网络拓扑.在物理链接层增加远程链路模型,为网络建模者提供不同 ns 实例上网络节点的互连.远程链路模型由 ns 中的链路模型继承而来,包含带宽、延迟、队列管理等属性.由于远程链路的一个端节点不在本地定义,所以定义远程链路时除指定本地端节点外,还需要指定一个标识.初始化时,通过交换消息,定义在不同 ns 实例上具有相同标识的远程链路被认为是互连的.在模拟过程中,数据包经过远程链路传输实际上是向相应 ns 实例发送事件消息,该事件消息中包含完整的数据包信息.

5.2 网络层

NS 缺省采用一种抽象的路由策略,用集中计算路由的方式替代了节点间路由信息的交换.在并行网络模拟器中,每一个 ns 实例只定义整个网络模型的一部分,由于缺乏全局的网络拓扑信息,该路由策略不能直接应用.而且大规模网络的模拟会消耗过多的内存,其中很大一部分被路由表占用.这是因为每一个节点的路由表包含到所有其他节点的路由信息,使其占用的空间以 n^2 的速率增长.文献[8]给出了一种在相关节点上按需计算路由的方法,由于节点不必保存路由表,所以节省了大量的内存空间.这种方法实际上是对网络模型的进一步抽象,忽略了模型中不感兴趣的细节.

实际上大规模网络的拓扑结构通常是分层的,我们在并行网络模拟器中采用了分层路由策略.图 3(左)给出了一个分层(两层)的拓扑结构,所有的节点被划分为 4 个彼此分离的簇.节点地址采用 $A.B$ 的形式,A 为簇号,B 为节点号.每个簇包含一个或多个边界节点(如 $C2$ 的边界节点 2.2,2.4).每个 ns 实例独立地计算本地簇的簇内路由,通过交换消息得到由边界节点构成的簇间虚拓扑(图 3 右),计算簇间路由.从而得到整个网络的路由信息.分层路由策略可以减小节点的路由表,从而减少了模拟时占用的内存空间.

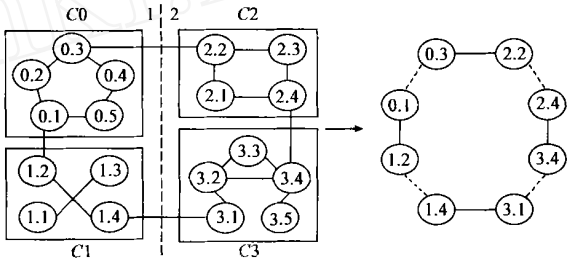

图 3 分层路由策略

5.3 传输层

传输层包含各种 TCP 和 UDP 协议模型.建立网络模型时需要在数据源和宿之间建立逻辑连接,比如 $ns connect $Tcp-Source $TcpSink,该命令将源和宿的端口号保存在数据包中.如果数据源和宿定义在不同的 ns 实例上,逻辑连接就无法用上述方法建立.为了在 ns 实例之间建立远程逻辑连接,我们将远端的数据源或宿绑定在节点的某个端口上,端口号由建模者显式得指定.建立远程逻辑连接时指定目标节点和端口号,即 $ns connect $LocalTcpSrc $DstNodeAddr $DstPort.

6 相关工作

近年来国外对并行网络模拟的研究非常活跃.UCLA 开发了一个基于 C 的并行模拟语言 Parsec,支持多种不同的时间同步算法.GloMoSim[9]是运行在 Parsec 之上的网络模型库,主要包含各种无线通信协议.Georgia Tech 使用网络描述语言 Ted 建立网络模型,运行在基于乐观同步算法的并行模拟内核 GTW[10]之上.但是这些系统都需要网络建模者学习新的语言来描述模型,使它们并没有得到广泛的应用.本文的并行网络模拟器以在网络研究领域得到广泛应用的 ns 为基础,为建模者将已有工作移植到并行提供了方便.

Ferenci 提出了基于 RTI-KIT 和模型代理(Model Proxy)的通用并行模拟体系结构[11],并以此为基础实现了 ns 的并行版本.RTI-KIT 是一个支持美国国防部高层体系结构(HLA)开发的工具包.模型代理定义了 HLA 联邦成员之间交互的规则.缺点是需要建模者手工计算和配置子模型之间的路由,这

对于大规模网络的模拟会相当麻烦. 本文的并行网络模拟器用分层路由策略扩展 ns 模型库,建模者只需要做少量的配置,能够支持大规模网络的模拟,具有很好的可扩展性.

7　性能分析

实验平台由四台浪潮 NP320 工作站组成,每一台的处理器为 PentiumIII 1.13GHz,内存 1.5GB,用 100Mbps Ethernet 互连. 实验中生成了包含 160 个节点的网络模型,每个工作站模拟 40 个节点. 用 ns 附带的工具 itm 生成需要的网络拓扑,其输入是用户自定义的脚本文件,输出为 SGB 文件,用 sgb2ns 转换为 ns 可读的 tcl 文件. 实验中分别用不同的配置参数生成网络模型,将并行模拟结果与单机 ns 的串行模拟结果作对比,如表 1 所示. 表 1 中的时间是实际模拟运行的时间,没有包含初始化和建立网络模型的时间. 可以看出,随着时间提前量(lookahead)的增加,加速比提高. 这是因为在每个处理循环中得到更大的时间下限 T_b,所以有更多的"安全"事件被处理. 本地逻辑连接数增加时,本地负载增大,在具有相同 T_b 的前提下,同样有更多的事件被处理,所以有更大的加速比.

表 1　加速比测试结果

Lookahead(ms)	8			13			20		
本地逻辑连接数	2	6	10	2	6	10	2	6	10
串行(s)	660	816	958	642	795	933	620	776	915
并行(s)	376	343	374	310	307	340	279	276	313
加速比	1.76	2.38	2.56	2.07	2.59	2.74	2.22	2.81	2.92

如前文所述,为了支持大规模网络的模拟,采用了分层路由策略. 图 4 是对网络规模的测试结果. 实验中生成了包含 1,000～10,000 个节点的网络模型. 曲线 1 描述了单机上用 ns 模拟占用的内存数,采用了 ns 缺省的路由策略. 曲线 2 是并行模拟的测试结果,曲线 2 的数据是四台工作站占用内存数的总和.

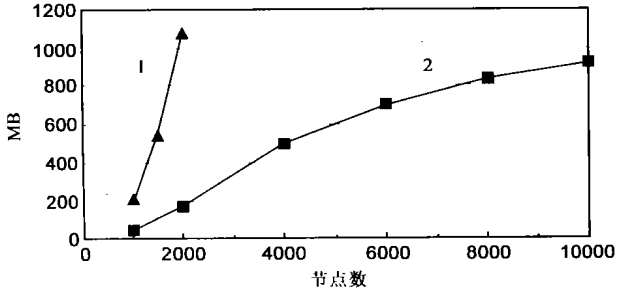

图 4　网络规模的测试结果

8　结论

网络模拟已经成为测试和评价网络协议和算法性能的重要手段. 为了使模拟大规模复杂网络成为可能的同时,省去网络研究人员学习和熟悉新的模拟系统和模拟语言的过程. 我们构造了基于网络模拟器 ns 的分布式并行网络模拟器. 因为对 ns 只做了有限的修改,所以已有的网络模型得以方便地重用,并为大规模网络的模拟提供了有力的支持.

参考文献:

[1] Fujimoto R. Parallel discrete event simulation[J]. Communications of ACM,1990,33(10):30 - 53.

[2] Kevin Fall. NS Manual[R]. A Collaboration between researchers at UC Berkeley,LBL,UC Berkeley,USC/ ISI,2002.

[3] Chandy K M,J Misra. Distributed simulation:A case study in design and verification of distributed programs[J]. IEEE Transactions on Software Engineering,1979,SE-5(5):440 - 452.

[4] L G Valient. A bridging model for parallel computation[J]. Communications of ACM,1990,33(8):103 - 111.

[5] Jefferson D R. Virtual time[J]. ACM Transactions on Programming Languages and Systems,1985,7(3):404 - 425.

[6] Eugene D. The butterfly barrier[J]. International Journal of Parallel Programming,1986,15(4):295 - 307.

[7] Fujimoto R. Parallel and distributed simulation[A]. Winter Simulation Conference Proceedings[C]. phoenix:ACM SIGSM,1999. 122 - 131.

[8] Polly H. Minimizing routing state for light-weight network simulation [R]. Swiss Federal Institute of Technology,2001.

[9] Bagrodia R,X Zeng,M Gerla. GloMoSim:A library for the parallel simulation of large wireless networks[A]. Proceedings of the 12th Workshop on Parallel and Distributed Simulation[C]. Banff,Alberta,Canada:ACM,1998. 154 - 161.

[10] Das S R,R Fujimoto. GTW:A time warp system for shared memory multiprocessors[A]. Winter Simulation Conference Proceedings[C]. Orlando:ACM SIGSM,1994. 1332 - 1339.

[11] L Ferenci. An approach for federating parallel simulators[A]. Proceedings of the 14th Workshop on Parallel and Distributed Simulation[C]. Bologna,Italy:ACM,2000. 63 - 70.

作者简介:

李　越　男,1978 年生于西安,现为西安交通大学计算机系博士研究生,主要研究方向为网络模拟,网络性能评价.

钱德沛　男,1952 年生于上海,西安交通大学计算机系教授,博士生导师,研究方向为计算机体系结构,计算机网络等.