

一种基于移动 Agent 的网络性能管理系统及性能分析^{*}

章 韵¹ 王汝传^{1,2} 姜 波¹

(南京邮电大学计算机学院 南京 210003)¹

(南京大学计算机软件新技术国家重点实验室 南京 210093)²

摘 要 针对传统的基于 SNMP 在网络管理中的不足,设计了一种基于移动 Agent 的网络性能管理系统,此系统融合移动 Agent 技术,解决了传统网络管理的管理资源占用、处理能力瓶颈和灵活性差等问题,提高了网络管理的效率。主要描述了本系统的架构模型及工作原理,并同传统的基于 SNMP 的网络性能管理系统进行了理论和实验比较,结果验证了本系统的优越性。

关键词 移动 Agent,网络性能管理,响应时间

Research and Implementation of Network Performance Management System Based on Mobile-agent

ZHANG Yun¹ WANG Ru-chuan^{1,2} JIANG Bo¹

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)¹

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)²

Abstract To solve the disadvantage of the traditional SNMP network management, this paper designed a new network performance management system based on mobile-agent. The simulation shows that the system improves the efficiency of network management system, by resolving the issues of traditional SNMP network management such as management resource consumption, unskillful and so on.

Keywords Mobile agent, Network performance management, Response time

1 引言

当人类社会进入 21 世纪时,一个显著的特征就是社会的网络化。网络不但为人们提供了方便的通信方式、丰富的共享资源,更为重要的是提供了一个综合的、自动的、分布式的处理平台。作为网络的重要组成部分,网络管理系统的建设和发展也面临着新的机遇和挑战,传统的基于 SNMP 的网络性能管理系统在扩展性、灵活性等方面已经不适应网络的发展,而移动 Agent 的智能性、迁移性等各种特性,在一定程度上可以解决传统的网络性能管理系统所遇到的问题。本文设计并实现了一种基于移动 Agent 的网络性能管理系统,并对此系统的管理性能进行了理论和实验分析。

2 传统的网络性能管理系统

传统的性能管理系统主要有基于 SNMP^[1]或者 CMIP^[2]的,而在实际应用中,目前主流的网络管理大多都是基于 SNMP 协议的,所有基于 SNMP 的网管体系的结构都基本相同^[3],构成要件主要包括:网络管理站 NMS、被管设备 DEVICE、管理信息库 MIB、管理代理 SNMP-agent、NMS 同被管设备之间的管理协议 SNMP。其结构如图 1 所示。

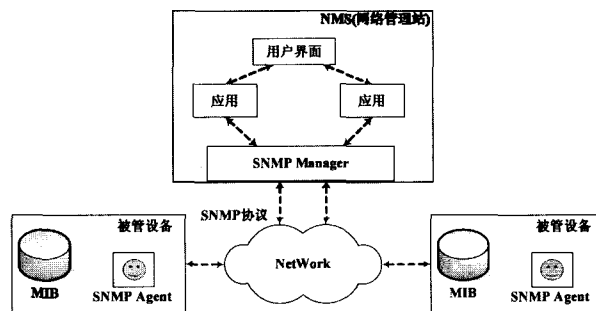


图 1 基于 SNMP 的网络管理体系结构图

从图 1 可以看到,集中式管理体系的管理应用程序 Management application 在 NMS 上运行,它通过各 SNMP-agent 对网络上的各种设备进行监测和控制,并对从各个节点上传的状态变量值和其它数据进行运算、分析,然后进行管理决策。SNMP-agent 以静态程序代码的形式驻留在网络设备上,结构简单。当 NMS 要执行某项任务时,SNMP-agent 会按 NMS 的 Polling(轮询)指令,按一定的时间间隔将 MIB 中状态变量值上传,但它本身并没有决策管理的能力。NMS 通过访问这些设备的 MIB 来对它们的状态进行监控。从根本上

^{*} 本课题得到国家自然科学基金(60573141 和 60773041),江苏省自然科学基金(BK2005146),江苏省高新技术研究计划(BG2005038、BG2006001),国家高科技 863 项目(2006AA01Z201,2006AA01Z219),2006 江苏省软件专项,南京市高科技项目(2007 软资 106,2007 软资 127),现代通信国家重点实验室基金(9140C1101010603),江苏省计算机信息处理技术重点实验室基金(kjs050001,kjs06006)和江苏省高校自然科学基金研究计划(07KJB520083)资助。章 韵 副教授,主要研究方向是计算机软件、计算机网络、信息安全、移动代理和虚拟现实技术等;王汝传 教授,博士生导师,主要研究方向是计算机软件、计算机网络、信息安全、移动代理和虚拟现实技术等;姜 波 硕士研究生,研究方向为信息网络、计算机网格、信息安全与对等计算等。

来说,SNMP 协议最核心的原理是由一个网络管理站负责轮询驻留在各网络设备上的代理。

而这种方案本身不可避免地存在着如下的不足:

(1)网络性能指标的计算量都存在于网络管理站一端,网络管理站的处理能力成为整个网络管理系统的瓶颈;

(2)网络管理站要定期频繁地向各被管设备发出请求报文,加大了网络中的管理通信开销;

(3)管理信息(MIB)是预先定义并且是标准化的,如果想进一步修改是很困难的,灵活性和扩展性不够。

利用移动代理的智能性、迁移性和自主性等特点^[4,5]可以解决上述问题,为此本文将移动 Agent 引入网络性能管理,提出了一种新的网络性能管理架构,并将其实现,最后进行了定性定量的分析。

3 基于移动 Agent 的网络性能管理系统

本文基于移动 Agent 的网络性能管理系统的主要结构(如图 2 所示)包括,网络管理中心(负责整个性能管理系统的运行维持,直接同操作者进行交互)、被管设备(其上具有移动 Agent 运行环境 MCC——Mobile-Agent Container Context,可以驻留运行移动 Agent)和数据库(主要维护网络管理系统中被管设备的地址、性能信息、状态等信息)。在本文系统中的大部分任务主要由三个移动 Agent 类的协作来完成,分别是:HomeMobilet, AnalysisMobilet 以及 PerforwatchMobilet,其主要结构和功能将在下面作详细的描述。在传统的基于 SNMP 的网络管理中,大多网络管理中心是基于轮询方式的,由于每次性能数据的采集,都要求管理站发出性能数据性能请求,这种方式对网络带宽的要求比较高,在先前有人设计的基于移动 Agent 的网络管理系统中,是基于代理串行迁移的^[6],如果对于较大的网络,这种系统对响应时间的要求达不到。本文的系统采用基于驻留代理主动传送性能数据的设计思想,这样既节省了网络性能管理系统的通信开销,又具有较高的响应时间。

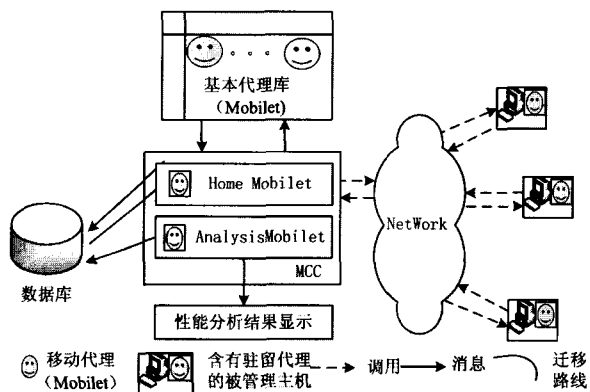


图 2 基于移动 Agent 的网络性能管理系统结构图

3.1 系统主要代理类及组件设计

在本文设计的网络性能管理系统中,移动代理系统利用了 MAP 系统,其主要的设计目标是 3 个代理类以及提取主机性能的组件的实现,下面分别对其功能进行描述。

1) HomeMobilet

HomeMobilet 的主要功能关系架构如图 3 所示。在本文所设计的网络性能管理系统中,HomeMobilet 是最基本的功能代理,在网管站运行的最底层代理,它负责网络性能管理系统的启动,网络被管设备地址的提取,监控代理 PerforWatch-

Mobilet 的生成及派发,性能数据的接收和存储,调用 AnalysisMobilet 根据需求生成性能报告。它同本网络性能管理系统中的多个模块产生关联:从数据库提取被管设备的地址信息和往其中储存被管设备的性能数据信息以及当前的采样频率信息;当需要管理某一网络设备时,生成监控代理 PerforWatchMobilet,并将其派发;当用户想查看性能分析结果时,调用分析代理 AnalysisMobilet,生成性能分析报告显示给用户。

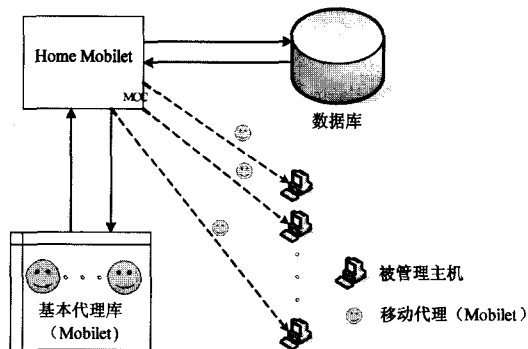


图 3 HomeMobilet 的主要关系图

2) AnalysisMobilet

主要负责根据用户的要求,对数据库中的数据进行分析,分析当前被管设备的性能状况,生成直观的图形来显示此设备的性能状况。

3) PerforWatchMobilet

除了上面两种静态代理外,还定义了一种动态代理 PerforWatchMobilet。此类代理为驻留代理,即是说当此类代理迁移到目标主机后就驻扎在此主机中,接收管理站命令主动向管理站发送性能数据信息。另外,此监控代理还可以接收网管站 HomeMobilet 的管理信息,主要是调整数据采集频率或者终止其任务的执行的命令信息。

4) GetPCInfofSC

在本文基于移动 Agent 的网络性能管理系统中,一个最基本的单元功能就是如何取得被管理主机的数据。在此本文利用 MAP 系统提供的组件接口来设计开发了相应的组件 GetPCInfofSC 及其组件代理,以提供本系统所需要的性能数据(主要包括 CPU 使用率、内存使用率、网络流量等)。

3.2 系统执行流程

本文所设计的网络性能管理总的基本流程如下(参见图 4)。

(1)首先启动网络性能管理系统,即启动已经由 MAP 构建好的基本代理 HomeMobilet,然后查询数据库,读取被管设备的地址等信息。

(2)HomeMobilet 根据被管设备的数目及地址,产生移动 Agent(PerforWatchMobilet),并将选择采集的设备信息、变量及性能指标的算法等相关信息动态加载到移动 Agent 执行代码中,最后将移动 Agent 发送到各被管设备中去。

(3)移动 Agent 按照管理者的要求,迁移到目的被管设备并驻留在此设备中。

(4)PerforWatchMobilet 向该被管设备的组件代理发出请求,采集被管设备与性能相关的数据信息并加以处理,然后定期向 HomeMobilet 发送处理过的性能信息(转步骤 6)。

(5)HomeMobilet 接收 PerforWatchMobilet 发送来的性能数据信息,并将其按一定的格式存储于本地数据库。

(6)PerforWatchMobilet 接收 HomeMobilet 发送来的更改数据采集时间间隔或终止任务的控制信息,并执行相应的任务(转步骤 4)。

(7)用户查看某被管设备的性能数据,并选择相应的格式。

(8)HomeMobilet 生成 AnalysisMobilet,对选择设备的性能数据进行分析,并得到要求格式的性能报告。

(9)将所得到的性能报告,显示到直观的界面上。

可以看出,本文所设计的基于移动 Agent 的网络性能管理系统,从数据采集、数据分析以及数据显示都和移动 Agent 相关,并且很多任务的完成需要多个代理的相互协作来共同完成。

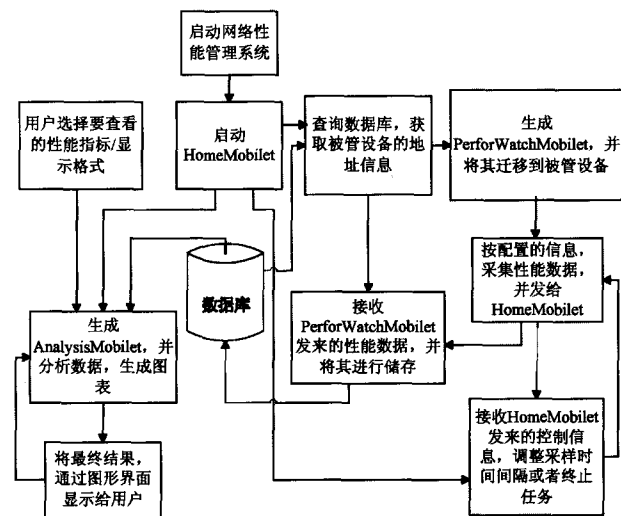


图 4 网络性能管理的执行流程

4 系统性能理论分析

对于本文所设计的基于移动 Agent 的网络性能管理系统,本文将其与传统的基于 SNMP 的网络管理系统进行对比,以验证本文设计的合理有效性。

4.1 网络流量的分析

NMS 的网络流量是指 NMS 在网络层与所有被管节点的数据字节数,网络流量是查看一个网络性能的重要因素,在构建一个网络性能管理系统,并实时查看网络性能时,应该做到最大程度上对网络的流量不要造成太大的影响,因此有必要对网络中的信息流量予以分析。

1) 基于 SNMP 协议的网络管理系统的网络流量分析

目前基于 SNMP 的性能管理无论是集中式网管还是客户端/服务器模式的分布式网管,在数据采集方面基本上采用的都是定时轮询模式。

这种模式的一般过程是,由管理者向分布在各个被管设备上的 SNMP 代理发出 Get/Get next 请求,SNMP 代理接收到该 get 请求后,将被请求的 MIB(management information base) 变量值以 get_response 报文的形式返回给管理者,再由管理者对数据进行存储、计算、分析,然后显示在图形化界面上。这种方法实现起来比较简单,但是却带来了不可避免的性能问题,它表现在网络流量占用方面的不足主要表现如下:因为若要获取一个 MIB 变量表的数据,则需要不断进行这种 get/response 报文的交互,因为要得到表中的下一个数据,必须从上一个返回的 get_response 报文中获得下一个数据的

OID 标识,这样即是说每隔一段很短的时间,管理者就要向各个被管设备发出轮询,同时接到请求的 SNMP 代理都要向管理者返回相应数据,这必然导致网络中的管理通信数据流量的增加,即是说很多 SNMP 报文占用了网络的带宽,此外,很多性能指标只靠获取一个 MIB 变量,无法完成指标的计算,NMS 通常要进行几次查询才能得到计算性能指标的所有 MIB 变量。例如要计算某端口的流入错误率,需要采集 if InEr2rors,if InUcast Pkts 和 if InNUcast Pkts 3 个 MIB 变量,NMS 只是盲目地收集所有原始数据,实际处理的往往只用到部分数据,从而造成了浪费,这使得网络通信数据流量翻倍。

2) 基于移动 Agent 协作的网络管理系统的网络流量分析

用本文所设计的移动 Agent 方案采集网络性能管理数据,相对传统的 SNMP 的单纯的轮询方式,在采集性能指标时,更有针对性,由于性能数据由被管设备上面的监控代理主动发送,这就减少了同网络管理站的交互流量,同时传输的是监控代理处理以后的数据结果,如获取某端口的错误率,那么 NMS 获得的直接就是某端口的错误率,这相对于 SNMP 方式中多次传送多个 MIB 变量然后加以选择处理才能得到指标的方式可以节省更多的管理通信开销,即减小由于网络管理而占用的额外带宽。

3) 两种方式的比较

在基于 SNMP 的网络管理方式中,假定网络中有 n 台被管设备,NMS 向每台被管设备发出一条 SNMP get/ get next request 请求报文的平均大小为 $getP$,被管设备 $Device_i(i=1,2,\dots,n)$ 向 NMS 返回的响应报文大小为 reP_i ,则 NMS 对网络中所有设备进行一次 MIB 变量的获取,其带宽消耗(数据流量)为:

$$B_{SNMP} = \sum_{i=1}^n (getP + reP_i) \quad (i=1,2,\dots,n)$$

若计算某一性能指标,平均需要采集 m 个 MIB 变量,则计算一个性能指标所产生的数据流量为:

$$B'_{SNMP} = m \times \sum_{i=1}^n (getP + reP_i) \quad (i=1,2,\dots,n) \quad (1)$$

本文中基于移动 Agent 的数据采集,采用驻留代理监控的形式,当系统启动后,代理迁移到被管设备后就不再迁移,因此在数据采集的执行过程中可以不考虑移动 Agent 的大小,监控代理会动态调整采样间隔,根据时间间隔定期地向 NMS 发送性能数据,节省了 NMS 的控制通信信息,而且监控代理对性能数据进行处理,得到结果以后再传送给 NMS,这也压缩了性能数据的大小,从而节省了 NMS 与每个设备进行通信的数据流量。

假定与上面相同的环境,在系统正常运行以后,NMS 只发送一些控制信息,由于监控代理会定期向 NMS 发送性能数据,因此不需要每次获取性能数据 NMS 都向被管设备发送请求报文。

由于监控移动 Agent 会在被管设备上将运算处理原始数据,将处理得到的结果直接带回给 NMS,这样,对于所监控的同一个性能指标,假设对于被管设备 $Device_i(i=1,2,\dots,n)$,设其发送的结果报文大小为 $D_i(i=1,2,\dots,n)$,则在本文的方案中,获取一个性能指标所产生的数据流量为:

$$B_{MA} = D_1 + D_2 + \dots + D_n \quad (i=1,2,\dots,n) \quad (2)$$

在传统的轮询算法中,对于一个性能指标,平均需要采集 m 个 MIB 变量,则可以知道 $D_i < reP_j * m(i,j=1,2,\dots,n)$,比较公式(1)和(2),显然易见:

$$B_{MA} < B'_{SNMP}$$

则易知,本文中所给出的方案可以在很大程度上降低由于网络性能管理本身所带来的带宽消耗开销。

4.2 响应时间的分析

一般情况下,评价一个网络应用系统的服务性能,响应时间(Response Time)是一个主要的指标。而网络性能管理系统的响应时间主要取决于如下几个方面:远程节点的 CPU 的运算能力和内存容量、系统所在网络的带宽、消息的类型以及应用的复杂程度等等。如果一个性能管理系统没有良好的响应时间,那么这个系统的合理有效性就要大打折扣。

响应时间是指从发出请求到接收到所有结果之间的时间间隔。一般地,网络性能管理系统总的响应时间由几个部分组成: $T = \text{原始数据的处理时间 } T_d + \text{网络延迟时间 } T_l + \text{网络传输时间 } T_s$ 。设网络中有 n 台被管设备,网络设备到网络管理站的平均网络带宽为 τ ,两台设备间的平均网络延迟为 δ ,网络管理站(NMS)的处理能力为 η ,网络中设备的平均处理能力为 μ 。

1) 基于 SNMP 协议的性能管理系统的响应时间分析

根据上一节中的分析,可以知道此种方式的性能管理系统中传输的数据流量为: $B'_{SNMP} = m \times \sum_{i=1}^n (getP + reP_i)$ ($i=1, 2, \dots, n$), 则其传输时间为:

$$T_s = \frac{B'_{SNMP}}{\tau} = \frac{m \times \sum_{i=1}^n (getP + reP_i)}{\tau} \quad (n=1, 2, \dots, n)$$

则在网络上面的所花费的时间为: $T_s + T_l = \frac{B'_{SNMP}}{\tau} =$

$$\frac{m \times \sum_{i=1}^n reP_i}{\tau} + n * \delta \quad (n=1, 2, \dots, n)$$

原始数据处理时间为:

$$T_d = \frac{B'_{SNMP}}{\eta} = \frac{m \times \sum_{i=1}^n (getP + reP_i)}{\eta} \quad (n=1, 2, \dots, n)$$

则得到总的响应时间为:

$$T = T_d + T_l + T_s = \frac{m \times \sum_{i=1}^n (getP + reP_i)}{\tau} + n * \delta + \frac{m \times \sum_{i=1}^n reP_i}{\eta} \quad (n=1, 2, \dots, n) \quad (3)$$

2) 基于移动 Agent 的网络性能管理系统的响应时间分析

同样,也根据上一节中的分析知道,完成一次性能数据采集的数据流量为:

$$B_{MA} = D_1 + D_2 + \dots + D_n \quad (i=1, 2, \dots, n)$$

则其传输时间为:

$$T_s = \frac{B_{MA}}{\tau} = \frac{\sum_{i=1}^n D_i}{\tau} \quad (i=1, 2, \dots, n)$$

在基于移动 Agent 的网络管理系统中原始需要处理的数据量和上面的同样为: $m \times \sum_{i=1}^n reP_i$, 那么其处理时间为:

$$T_d = \frac{m \times \sum_{i=1}^n reP_i}{n * \mu} \quad (i=1, 2, \dots, n)$$

则得到本文中所提的网络性能管理系统中一次性能数据采集处理的总响应时间为:

$$T' = T_d + T_l + T_s = \frac{m \times \sum_{i=1}^n D_i}{\tau} + n * \delta +$$

$$\frac{m \times \sum_{i=1}^n reP_i}{n * \mu} \quad (n=1, 2, \dots, n) \quad (4)$$

比较 T 与 T' 两个公式,则很容易发现,二者的网络延迟大体一致,网络传输时间后者明显快于前者;当网络中设备的数量达到一定程度时,后者的原始数据处理时间明显地要短于前者,所以易见,在同等情况下,本文提出的网络性能管理系统的响应时间要优于传统的基于 SNMP 的网络性能管理系统。

5 系统性能实验分析

本文设计了实验来验证上节的理论分析,对于基于 SNMP 的传统网络性能管理和本文提出的基于移动 Agent 的网络性能管理两种不同的网络性能管理方式,在一个子网内连续管理采集多台网络设备的数据,每台网络设备采集 5 个不同的性能指标数据。

测试方法:管理站分别使用本文实现的网络性能管理系统和基于 SNMP 的传统轮询的网络性能管理系统,对数据流量的测量使用 Sniffer 软件对网络站 MAP 平台端口和 UDP161(即 SNMP 服务器的端口^[7])进行监控。获取响应时间的方法则是分别使用各自编程语言中的相应的时间函数来获取时间差。

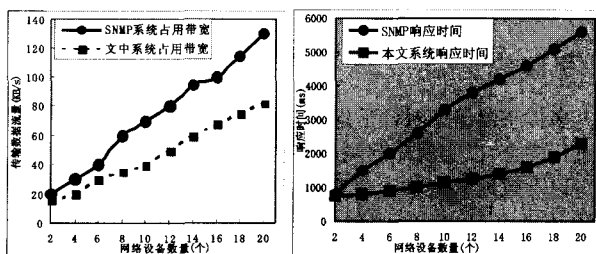


图 5 两种不同网络管理模式对比图

测试步骤:对被管设备提取相应的性能数据,测量数据流量和响应时间,改变被管设备的数目再进行测量。

网络环境:10Mbit 以太网网络。

应用平台:Windows。

分别对比两种网络对于网络带宽的占用情况,可以得到如图 5 和图 6 所示的测试结果,可以清楚地看出本文提出的网络性能管理系统,在执行同样网络性能管理任务的情况下,要比传统的基于 SNMP 的网络性能管理系统节省大量的带宽,并且响应时间也比基于 SNMP 的网络性能管理系统短很多。

结束语 文中提出的基于移动 Agent 的网络性能管理方案,可以减少网络管理的数据流量,缩短管理响应时间,避免传统 SNMP 轮询带来较大的网络管理通信开销,以及由此引起的网络抖动甚至短暂的阻塞。本文提出的自适应数据变化规律的数据采集算法在网络性能管理中对数据进行拟和时,可以在较小的管理通信开销下,得到更加精确的拟和曲线。除此之外,增强了网络中各节点的计算能力,使网络管理的计算功能不再集中在管理者一方,而是将这些计算功能合理地分配到网络中的其他网元设备,减轻了管理者的负载,利用移动 Agent 的永续机制,使系统更加健壮,从而提高整个网络管理的分布性和可靠性。本系统在设计与实现中主要考虑系统性能问题,对移动 Agent 安全性考虑得较少。这是整个系统

(下转第 191 页)

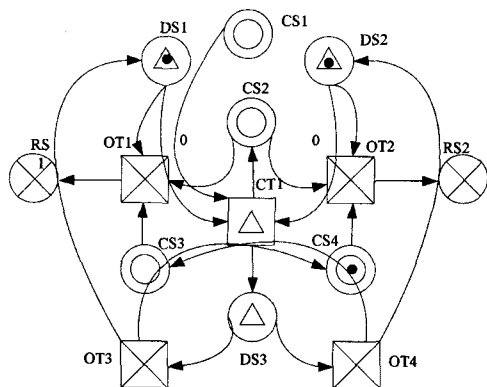


图2 斐波那契数列计算器的数据流 Petri 网模型

在图2所给出的斐波那契数列计算器数据流 Petri 网模型中,数据驻留站库所 DS1, DS2, DS3 表示存储器;数据拷贝引用操作变迁 OT1 表示加法器;辅助控制库所 CS1 表示控制器;数据对象引用操作 OT1 和 OT2 负责清除存储区当中的过时数据,而数据对象引用操作 OT3 和 OT4 负责将新产生的数据复制到相应的存储区当中。根据数据流 Petri 网系统的定义,各种库所与变迁之间的流关系权已经是确定的,但为了阅读方便,在图中没有标识的流关系的权为 1,标识为“0”的权为零。下面以求解第 5 项斐波那契数列的值为例,简单说明斐波那契数列计算器数据流 Petri 网模型系统的运行过程。表 1 中的系统标识列表(有关 Petri 网标识的相关理论请参考文献[2])记录了斐波那契数列计算器数据流 Petri 网模型的完整运行过程。

表 1 斐波那契数列计算器数据流 Petri 网模型标识列表

| M | CS1 | CS2 | CS3 | CS4 | DS1 | DS2 | DS3 | RS1 | RS2 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 3 | 0 | 0 | 1 | 1/1 | 1/1 | 0 | 0 | 0 |
| 1 | 2 | 1 | 0 | 1 | 1/1 | 1/1 | 1/2 | 0 | 0 |
| 2 | 2 | 0 | 0 | 0 | 1/1 | 0 | 1/2 | 0 | 1 |
| 3 | 2 | 0 | 1 | 0 | 1/1 | 1/2 | 0 | 0 | 1 |
| 4 | 1 | 1 | 1 | 0 | 1/1 | 1/2 | 1/3 | 0 | 1 |
| 5 | 1 | 0 | 0 | 0 | 0 | 1/2 | 1/3 | 1 | 1 |
| 6 | 1 | 0 | 0 | 1 | 1/3 | 1/2 | 0 | 1 | 1 |
| 7 | 0 | 1 | 0 | 1 | 1/3 | 1/2 | 1/5 | 1 | 1 |
| 8 | 0 | 0 | 0 | 0 | 1/3 | 0 | 1/5 | 1 | 2 |
| 9 | 0 | 0 | 1 | 0 | 1/3 | 1/5 | 0 | 1 | 2 |

如表 1 所示,求解第 5 项斐波那契数列的数据流 Petri 网模型在 CS1(5-2=3)的控制下,系统经历了 9 个状态的变化,完成了三次求和运算。如果将数值 1 带入 DS1, DS2 的初始状态中(a/b, a 代表标识数, b 代表数值),经过三次求和计算,得到第 5 项斐波那契数列的数值—“5”。

结束语 本文从分析数据资源的特点出发,重点介绍了一种基于 Petri 网原理的新型的数据流程建模方法——数据

流 Petri 网,给出了数据流 Petri 网模型的形式化定义和图形表示,并以求解斐波那契数列第 n 项数值为例,详细说明了数据流 Petri 网模型在数据流程建模上的表达能力和建模特点。数据流 Petri 网是一种新型的数据流建模方法,基于数据流 Petri 网模型的数据流分析和系统检测等相关技术将成为进一步研究的重点。

参考文献

- [1] 乐晓波, 陈黎静. Petri 网应用综述[J]. 长沙交通学院学报, 2004, 20(2): 51-55
- [2] 袁崇义. Petri 网原来与应用[A]. 电子工业出版社, 2005
- [3] Lee J, Member S. Modeling Uncertainty Reasoning With Possibilistic Petri Nets[J]. IEEE Transaction on Systems man and Cybernetics-Part B: CYBERNETICS, 2003, 33(2)
- [4] Shen R L. Reinforcement Learning for High - Level Fuzzy Petri Nets [J]. IEEE Transaction on Systems Man and Cybernetics-Part B: Cybernetics, 2003, 33(2)
- [5] Adamyan A, He D. Sequential Failure Analysis Using Counters of Petri Vet models[J]. IEEE Transaction on Systems man and Cybernetics-Part A: Systems and Humans, 2003, 33(1)
- [6] Zhang Zhiming, Wang Yue, Tao Ran, et al. Resource Allocation Using Timed Petri Vets and Heuristic Search[J]. Journal of Beijing Institute of Technology, 2000(9):148-154
- [7] 罗军舟, 顾冠群, 谢俊清. Petri 网的协议分析器[J]. 计算机学报, 1997(3):206-212
- [8] 丁正已, 谢磊. Petri 网在多处理器系统性能分析中的应用[J]. 信息工程大学学报, 2000(12):58-60
- [9] 丁彩虹, 黄文虎, 姜兴渭. Petri 网在基于规则系统一致性问题的检查中的应用[J]. 机械强度, 1999(9):170-173
- [10] 周必水, 倪慧莉. Petri 网在时序建模中的应用[J]. 杭州电子工业学院学报, 2000(8):1-8
- [11] 曹阳, 张维明, 沙基昌, 等. Petri 网在通信网络仿真建模中的应用[J]. 计算机仿真, 2001(5):38-41
- [12] 李慧芳, 范玉顺. 基于 Petri 网仿真的批处理过程动态调度[J]. 系统仿真学报, 2002(7):928-931
- [13] 陈胜, 程景云. 基于 Petri 网的工作流建模及仿真[J]. 计算机辅助工程, 2000(5):34-37
- [14] 田国会, 贾磊, 路飞, 等. 离散事件系统的面向对象实时时态着色 Petri 网模型[J]. 系统工程理论方法应用, 2001(10):206-208
- [15] Ichikawa A, Yokoyama K, Kurogi S. Control of Event - Driven Systems-Reachability and Control of Conflict-Free Petri Nets. Trans. Soc. Instrum. Control. Eng. (Japan), 1985, 21(4): 324-330
- [16] David R, Alla H. Continuous Petri Nets // Proceedings of 8 th European Workshop on Application and Theory of Petri Nets. Zaragoza, Spain, 1987:275-294

(上接第 125 页)

原型转化为实际应用时需要进一步研究的问题。

参考文献

- [1] Case J, Fedor M, Schoffatal M. A simple network management protocol(SNMP)[S]. RFC1157, 1990
- [2] 张国鸣, 唐树才, 薛刚逊. 网络管理实用技术[M]. 北京: 清华大学出版社, 2002
- [3] 杨家海, 任宪坤, 王沛瑜. 网络管理原理与实现技术[M]. 北京:

清华大学出版社, 2000

- [4] Pham V A, Karmouch A. Mobile software agent: an overview [J]. IEEE Communication Magazine, 1998, 36(7): 26-37
- [5] Wooldrige M. Intelligent Agents: Theory and Practice[J]. The Knowledge Engineering Review, 1995, 10(2): 115-152
- [6] 郭学理, 肖岗, 曹成里. 基于智能代理的网络管理系统研究[J]. 小型微型计算机系统, 2001, 22(9):1051-1054
- [7] 谢希仁. 计算机网络(第四版)[M]. 电子工业出版社, 2003