

事件驱动的分布式自适应网络管理系统

丁振国 王静懿 赵永杰

(西安电子科技大学计算机学院 西安 710071)

摘要 借鉴 CORBA 事件服务的方式,提出并实现了基于事件驱动模型的分布式自适应网络管理系统(EDDM)。该系统在功能上具有强可伸缩性,可实现海量网络事件的缓冲、过滤和分发;同时分布式架构具有良好的自适应性,根据 BP 神经网络算法可动态调整管理节点的负载,允许管理站的加入和退出,并及时根据实际需要调整系统的管理层次结构。实验结果表明,EDDM 可有效缩短故障响应时间,减小网管数据占有的网络带宽。

关键词 网络管理,分布式,事件驱动,神经网络,自适应

中图分类号 TP393 **文献标识号** A

Event-driven Based Distributive and Adaptive Network Management System

DING Zhen-guo WANG Jing-yi ZHAO Yong-jie

(Department of Computer Science, Xidian University, Xi'an 710071, China)

Abstract Referring to CORBA event services scheme, the paper proposed an event-driven based distributive and adaptive network management system named EDDM. The system has very good flexibility in functions, realizing the buffering, filtering, and distributing capabilities of massive network events. Meanwhile distributed architecture has fine adaptability. The load and number of the management nodes can be dynamically adjusted by BP neural network algorithm. Then the management architecture can be changed according to the actual needs. Experiments show that EDDM can effectively reduce the latency of response and the bandwidth occupied by management data.

Keywords Network management, Distributed system, Event-driven, BP neural network, Adaptive system

1 引言

现有网络管理方案面临的一个问题是缺乏统一的管理流程,包括功能流程和业务流程。从功能流程的方面来说,现有的网络管理方案大多定义了单个管理操作和这些操作的对象,没有指出具体流程。缺乏统一的网络管理流程,会严重制约网管系统功能的有效发挥^[1]。面临的另一个问题是基于 SNMP 的网络管理系统通常采用集中式网络管理模型,不适合应用于较大规模的网络系统,大量的 SNMP 协议数据单元 PDU 向管理站汇集会影响正常业务的运行^[2]。

针对上面的问题,本文提出了基于事件驱动模型的自适应分布式网络管理系统(EDDM)。该系统基于 SNMP 网管协议,应用事件驱动模型来解决网络管理的业务流程。例如,系统得到一个“service failed”的事件告警后,会自动启动通知管理员以及修复的功能,或者将此故障信息显示在拓扑图上,供管理员查看以及进行手工修复。事件将整个系统的各个独立运行的功能模块协同起来,使系统功能在结构上具有强的可伸缩性。EDDM 也是具有中心节点的分布式结构,其重要特点是:(1)自适应性,通过增加管理站的数量可以方便地扩大被管网络的规模,允许管理站动态加入和退出;(2)负载平

衡能力,系统可以根据管理站实现负荷的情况向管理站动态分派任务。

通过上述技术,EDDM 解决了网络管理业务流程,实现了各功能模块以松耦合的方式协同工作;也解决了 SNMP 集中式网络管理的带宽和处理瓶颈,增加了系统的顽健性和管理大规模网络的能力;同时方案具备从集中式网管直接升级为分布式的优点。

2 EDDM 模型

2.1 基本设计思想

2.1.1 事件驱动模型

网络管理系统采用 OSI 规定的 FCAPS 模型,即网络配置管理、性能管理、故障管理、账务管理和安全管理功能^[3]。整个网络管理过程包括对被管理对象信息的采集、处理、通告、报警等。根据事件驱动模型^[4],事件的生产者采集被管对象上的信息,生成相应的事件后发送到事件管理器。事件管理器将事件进行缓冲过滤存档,再向已订阅该事件的事件消费者发送该事件,接着由事件的消费者对事件进行处理,如图 1 所示。这样构建的一个基于事件驱动构架的网络管理系统,会使程序和系统响应更灵敏,也便于事件的生产者和消费

到稿日期:2010-08-20 返修日期:2010-11-01 本文受中央高校基本科研业务费专项资金资助项目(JY10000917003),综合网络管理系统(QTQXA20091QB05)资助。

丁振国(1959—),男,博士,教授,硕士生导师,主要研究方向为计算机网络与远程教育技术,E-mail:dingzhenguo@126.com;王静懿(1987—),女,硕士,主要研究方向为分布式处理,E-mail:sand-pearl@163.com(通信作者);赵永杰(1987—),男,硕士,主要研究方向为网络事件处理。

者功能的增加和删除,使系统具备了良好的伸缩性。

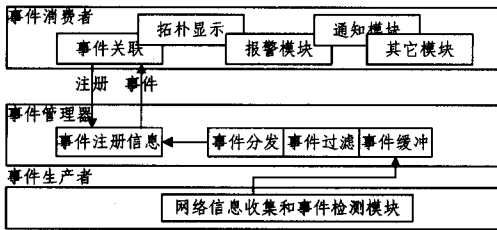


图1 事件驱动的管理模型

2.1.2 自适应分布式管理

采用 ICE 技术思想,维护一种分布式、动态、自治的监测网络结构。开始阶段,被管理设备比较少,由管理服务器独立监测全部设备。随着被管理设备数量的增加,激活部分性能良好、运行稳定的被管理设备的监测属性,将本该由管理服务器完成的轮询任务进行任务分配,最终由激活的被管理设备代替完成。当被管理设备监测到管理范围中有故障发生时,通过异步告警通知管理服务器,由管理服务器进行处理。此模型将管理服务器从轮询任务中解脱出来,使它的负载与管理设备的数量呈非线性增长。它要做的事情从不断地收集所有设备的数据转变成了将要轮询的任务分配出去,由被监测设备进行自治监测。当设备状态发生变化时,进行必要的任务调度,如图 2 所示。

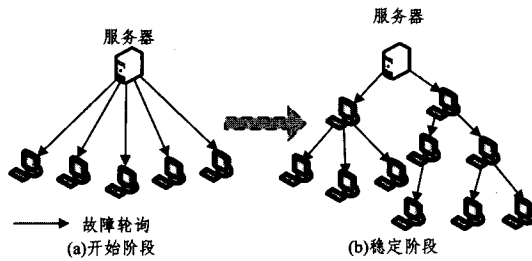


图2 监测网络结构的动态调整

2.2 系统框架

2.2.1 管理站基础结构

NMS 的监控、控制和数据收集功能可以由一组称为守护程序的并发任务来处理。各个守护功能模块通过事件的形式松耦合在一起,如图 3 所示。

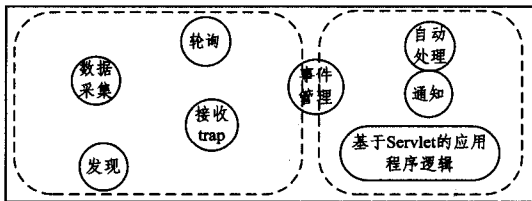


图3 管理站基础框架

(1)事件生产者

①轮询:此模块被认为是网络管理进程中的核心功能。它是一个按需开启的组件,通常在网络发现单元开启后立即被开启,对指定服务或者应用程序进行轮询。

②数据采集:用来采集节点的通信情况,比如 IP 或者 ARP 包的发送、接收情况等。

③接收 trap:用来接收被管设备发送过来的 trap 信息。

④发现:用来发现新的节点。在系统开始发现和分析网络设备时,它被认为是第一个被激活的组件。通过事件处理模块,获取的设备数据被返存到数据库中。

(2)事件管理器

事件管理器从事件生产者处接收事件,将事件进行缓冲、过滤和存档,再发送给已订阅该事件的事件消费者。

(3)事件消费者

①自动处理:自动处理模块可以调用、配置单元模块来自动处理一些已经注册过的事件。这个模块接收来自事件控制器发送的事件后,根据该事件的唯一标识来自动执行配置操作,例如开启、关闭某些服务等。

②通知:通知的守护程序监听一些自己感兴趣的事件。当这些事件发生时,可以各种方式,包括飞信、邮件、鸣笛等通知用户,以使用户及时处理这些事件。

(3)事件类型和来源

事件通道处理的事件类型和来源包括①被管对象(路由器、主机、应用服务等)主动发送过来的事件,定义的类型有链路 up/down、主机性能/故障相关告警(如负载过重、硬盘空间不足)等。②数据采集模块定期轮询发现的事件,如应用服务无响应或者服务关闭等。每类事件由 UEI(Unique Event Identifier)唯一标识。

2.2.2 分布式管理站结构

分布式模型分为 3 部分:数据存储、中心管理站和动态管理站。中心管理站是整个监测系统的核心,维护着整个系统稳定、高效运行。一方面它对节点管理的任务进行调度,维护着虚拟的监测网络;另一方面完成已接收到的信息的处理。动态管理站是向一些性能较高的被管理设备植入代理与远程对象服务,使其具备监测与被监测的双重身份,如图 4 所示。

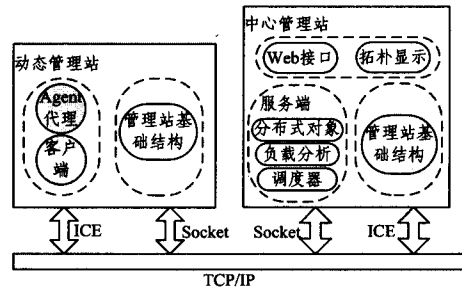


图4 分布式监测系统整体框架

(1)中心管理站

中心管理站作为集中式管理的管理站运行,也可以作为分布式管理站的中心管理站运行。除了具备接收动态管理站的事件外,还可以根据动态管理站的内存和 CPU 利用率进行负载分析,使用调度器对监控任务进行重新调度。这些功能集中放在服务端的守护程序里。服务端守护程序主要包括负载分析、调度器以及分布式对象 3 个部分。

①负载分析:负载分析由 BP-神经网络部分和动态负载部分组成。BP-神经网络主要对设备的稳定性和负载的振幅进行长期训练,其结果作为任务调度时的重要依据。动态负载部分主要实时记录具体设备当前 CPU、内存的使用情况,也是任务调度的一个重要依据。

②调度器:调度器是服务端守护程序的核心,也是中心管理站的核心。一方面,消息接收器收到消息后交由调度器进行处理,由调度器更新负载分析模块及数据库的相关信息,同时决定由故障处理中的哪一个组件进行下一步处理。另一方面,监测任务的分配与再调度也由调度器完成。当有新添加的监测任务时,调度器结合负载分析模块,选出一个最合适的

动态管理站,通过远程对象,将新添加的任务分配给该动态管理站,从而达到分配的目的。当整个系统出现负载不均时,调度器通过远程对象,将超载节点的任务转移到欠载节点,从而进行再调度,实现了整个系统的负载均衡。

③分布式对象:采用 ICE 技术构建的对象,主要对动态管理站进行远程控制,包括动态管理站所管理的节点范围、守护程序的状态等。例如可以远程启动或者关闭事件生产者/消费者中的守护程序。

(2) 动态管理站

动态管理站虽然是被中心管理站管理的,但是可以在中心管理站的允许下独立地对被管范围进行监控、处理、通知。它与中心管理站的交互内容都包括通过 socket 向管理站发送指定事件,以及通过 ICE 协议接收管理站实现远程控制,比如启动或者关闭某个守护程序,或者添加或者删除被管理范围等。

①Agent 代理:此代理为守护进程,用来收集当前 PC 的内存和 CPU 的运行数据。

②客户端:客户端同样也是守护程序,用于向中心管理站发送事件以及接收服务器端 ICE 控制的守护程序。

3 神经网络负载均衡算法描述

目前负载均衡算法可以分为静态和动态两种,但主要还是集中在动态算法上^[5]。动态负载均衡是分布式系统研究的一个重要方面,其主要目的是把各项网管任务根据管理站的负载情况均衡地分派下去。通常的解决方案是根据每个动态管理者的资源情况,包括 CPU 利用率、内存空间和网络带宽等因素来综合考虑、权衡任务的分派^[6-8],或者根据管理者或被管对象的数量进行任务分派^[9]。由于任务的调度本身是 NP 难问题^[10],因此在本网络管理系统中采用了启发式算法:根据动态管理站的 CPU 和内存利用率进行动态调度,采用适当的策略把不同的任务分配到相应的动态管理站上去运行。

3.1 问题描述

在本系统中,由于是将监测任务分配到 PC 上,因此由节点的不稳定性造成的影响会比较严重。将管理任务分配给不稳定的节点,会因为该节点突然关机或者运行某个进程占用了大量 CPU 和内存,而使中心管理站不得不对该节点的任务进行重新调度,这种现象称为“抖动”。减小不必要的抖动,是调度算法需要解决的问题。

在本系统中,任何一个节点都可以用一个正整数 C 来抽象描述其处理能力,称 C 为该节点的能力值;处理能力 C 的最小阈值用正整数 $T(T < C)$ 来描述,而阈值用正整数 α 来描述。于是,当前节点的实际负载 L 和处理能力阈值 T 与 α 存在如下函数关系:

$$F(L, T, \alpha) = \begin{cases} 0, & \text{当 } L < T - \alpha \text{ 时,表示当前节点为空闲节点} \\ 1, & \text{当 } L \in [T - \alpha, T + \alpha] \text{ 时,表示当前节点为负载适中节点} \\ 2, & \text{当 } L > T + \alpha \text{ 时,表示当前节点为超载节点} \end{cases}$$

定义 1(平衡指标) 设在某区间 Ω 上有 N 个设备需要被监测,已知它们之间的处理是完全并行的,且花费的 CPU 时间固定但不均等。定义第 i 个设备需要被监测的故障任务量为 $q_i (1 < i < n)$, q_i 要消耗的 CPU 执行时间为 c_i , 在第 j 个节点执行 q_i 所要消耗的 CPU 执行时间为 $c_{ij} \mu_j$ (μ_j 为影响因

素的权重)。

设所有动态管理站的集合为 S , 令 $m = |S|$, 则每个节点的监测任务集合为

$$Q_i = \{q_k \mid 1 < k < n \text{ 且 } k \neq i\} Q_1 \cup Q_2 \cup Q_3 \cup \dots \cup Q_m = \Omega \text{ 且 } Q_i \cap Q_j = \emptyset, i \neq j$$

在第 i 个节点上,执行完一次监测任务所要消耗的 CPU 执行时间为 $T_i = \sum_{j \in Q_i} C_{ij} \mu_j$ 。

系统全部执行完所有监测任务所需要的时间为 $T_{\max} = \max_{i=1}^m (T_i)$ 。

这里给出了 CPU 执行时间的平衡指标为

$$\eta = \frac{\sum_{i=1}^m T_i}{T_{\max} \times P}$$

可见, $0 < \eta \leq 1$, η 越逼近 1, 各节点完成监测任务的时间越相近。如果 η 很小, 则说明有的节点上监测任务过早地完成了, 而有的节点上监测任务消耗的时间相对较长。在这种情况下就需要进行任务调度。

定义 2(稳定性) 如果某一动态管理站在单位时间 α 内, 负载未超过 β 且未发生关机或故障, 则认为该动态管理站在该时间内的状态是稳定的。在本系统中, 将稳定的动态管理站占总节点的比例称为该系统的稳定性。

一维网络故障监测任务调度问题: 给定任意 m 个动态管理站 p_1, p_2, \dots, p_m , 要求寻找一种任务分配结构, 将监测任务 Ω 分解为 m 个子区域 Q_1, Q_2, \dots, Q_m , 依次分配给 m 个动态管理站, 由它们负责处理各自包含的子区域内的设备监测, 使得所有动态管理站的当前负载小于 β , $\eta \geq \delta$, 并尽量保证系统的稳定性。

3.2 算法描述

调度器实现的任务分配过程如下:

(1) 从数据库取出 CPU 使用率和内存使用率, 从而计算出每个动态管理站的负载。数据库中的消息是由接受 Trap 的守护程序即时更新的。

(2) 按照负载权重对动态管理站队列进行排序。本文采用的负载公式为

$$L_{\text{总}} = 0.5L_{\text{CPU}} + 0.5L_{\text{MEM}}$$

(3) 筛选掉 CPU 占用率大于阈值或内存占用率大于阈值的动态管理站。

(4) 用动态管理站编号和当前时间作为输入, 结合保存的权值信息文件, 采用 BP 神经网络计算其输出。

(5) 将稳定性大于阈值的动态管理站放入主要队列, 将小于等于阈值的动态管理站放入次要队列。分配任务时, 先从主要队列中选取节点, 主要队列为空时, 再从次要队列中选取, 分配任务的次要节点不超过总数的十分之一, 如图 5 所示。

学习过程描述如下:

(1) 工作信号正向传播: 输入信号从输入层经隐藏层传向输出层, 在输出端产生输出信号, 这是工作信号的正向传播。在信号的向前传递过程中, 网络的权值固定不变, 每一层神经元的状态只影响下一层神经元的状态。如果在输出层不能得到期望的输出, 则转入误差信号反向传播。

(2) 误差信号反向传播: 网络的实际输出与期望输出之间的差值即为误差信号。误差信号由输出端开始逐层向前传

播,这是误差信号的反向传播。在误差信号反向传播的过程中,网络的权值由误差反馈进行调节。通过权值的不断修正使网络的实际输出更接近期望输出[7]。

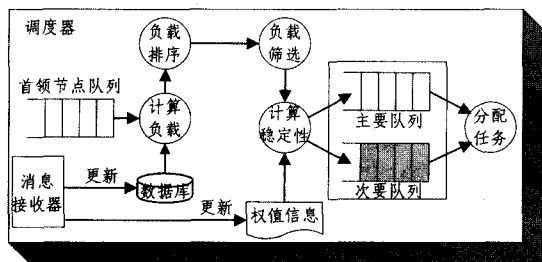


图5 任务分配模型

4 性能分析

4.1 响应时间测试

测试指标:当 EDDM 监测模型首领节点数为 1、2、3 时,SNMP 集中式监测模型发现该故障的响应时间随任务个数的变化情况。

测试方案:在被管理端,模拟一个网络连接故障,通过轮询组件来监测故障的发生。

具体测试过程如下:

(1) 在测试基于 SNMP 监测模型的性能时,首先制定 10 个网络连接故障监测任务,其中 9 个被监测设备网络连接正常,1 个无法连接。将网络无法连接设备的任务排在任务列表的最后来模拟最坏情况。依次增加网络连接正常的任务个数为 19,29,⋯,99 来模拟拥有 20 台,30 台,⋯,100 台被管理设备的网络环境。

(2) 在测试 EDDM($N=1$)监测模型的性能时,首先选取一台设备作为首领节点,如(1)所述制定监测任务,并将任务分配给首领节点执行。依次增加网络连接正常的任务个数,来模拟拥有 20 台,30 台,⋯,100 台被管理设备的网络环境。总响应时间为中心管理站分配任务到成功接收故障消息的总时间。

EDDM($N=2$)和 EDDM($N=3$)选取的首领节点的个数分别为 2 个和 1 个,其它步骤与(2)一致。

测试结果如图 6 所示。从图中可以看出,EDDM($N=1$)模型的总响应时间长于 SNMP 集中式模型,两种模型的增长幅度基本一致。EDDM($N=2$)模型的总响应时间的增长幅度小于 SNMP 集中式监测模型。刚开始 EDDM($N=2$)的响应时间大于 SNMP 集中式模型,当任务个数超过 30 后,小于 SNMP 集中式监测模型。EDDM($N=3$)模型的总响应时间小于 EDDM($N=1$)和 EDDM($N=2$),在开始阶段 EDDM($N=3$)模型的总响应时间大于 SNMP 集中式监测模型,当任务个数超过 22 后,其响应时间小于 SNMP 集中式监测模型。

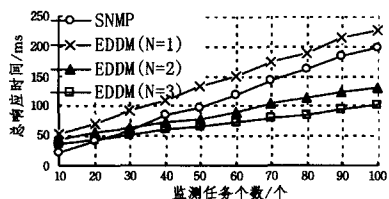


图6 总响应时间变化情况

实验结果表明,EDDM 监测模型的响应时间随着首领节点个数的增加,其增幅逐渐减小。当任务个数达到某一值时,EDDM($N>1$)监测模型的响应时间将小于 SNMP 集中式监测模型。EDDM($N>1$)监测模型在监测任务数较多的情况下有效地缩短了故障发现响应时间。

4.2 流量测试

测试指标:在轮询周期和任务数量不变的情况下,EDDM 监测模型与 SNMP 集中式监测模型的中心管理站流量随时间的变化情况。

测试方案:以读取每个被管理设备上的 sysDescr(OID 为 1.3.6.1.2.1.1.1.0)变量值作为管理任务,指定轮询周期为 5s,任务数为 100 个。使用 Sniffer 软件对网络管理站的特定端口进行流量监控。

图 7 给出了 SNMP 集中式监测模型在监测被管理设备时中心管理站的网络流量随时间的变化情况。从图中可以看出,网络流量随时间呈周期性波动,其峰值为 200 Packets,波峰之间的距离为 5s。波峰与任务个数有关,波间距与轮询周期相关。图 8 给出了 EDDM 监测模型在监测被管理设备时中心管理站的网络流量随时间的变化情况。图中有两处明显的波动,第一处是由中心管理站与数据库通信,获取设备信息和任务信息造成的。第二处是由中心管理站与动态管理站通信,将监测任务分发给动态管理站造成的。

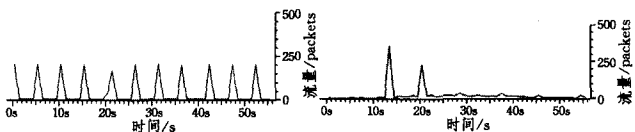


图7 SNMP 集中式监测模型流量变化 图8 EDDM 监测模型流量变化

测试结果表明,EDDM 监测模型除了在数据库通信阶段和分配任务阶段的流量高于 SNMP 集中式监测模型外,其余时间流量均低于 SNMP 集中式监测模型。EDDM 监测模型有效地减小了网络流量。

结束语 本文提出并实现了基于事件驱动模型的分布式自适应网络管理系统(EDDM)。基于事件驱动模型实现管理站的基础结构,事件的生产者采集被管对象上的信息,生成相应的事件后发送到事件管理器。事件管理器将事件进行缓冲过滤存档,再向已订阅该事件的事件消费者发送该事件。事件生产者和消费者的功能由守护程序实现。各个守护程序之间是松耦合的关系,易于添加和删除,所以系统在功能上具有强的可伸缩性。同时,分布式架构各管理模块依据基础管理模型扩展其它管理功能,中心管理站根据 BP 神经网络算法以 CPU 和内存为参数,动态调整下层管理节点的负载。系统的管理层次结构根据需要会在集中式和多层式之间变动,可最大限度地有效管理被管节点。实验结果表明,EDDM 可有效缩短故障的响应时间,减小网管数据占有的网络带宽。

参考文献

- [1] 杨家海,吴建平,王沛渝. 一个基于事件驱动的网络故障管理系统的设计与实现[J]. 计算机工程与应用,2003,23:4
- [2] 孙延涛,杨芳南,石志强. 负载均衡的分布式网络管理系统[J]. 通信学报,2009,3:34
- [3] ISO/IEC 10040. Information technology-Open Systems Inter-

[4] Yang J, Wu J, You Y. A Web-based, Event-driven Network Management Architecture[C]// 4th Asian-Pacific Conference on Communications, 1999:1214

[5] 李克清. 一种动态自适应的网格平衡调度算法[J]. 武汉大学学报, 2006(2):69

[6] Gavalas D, Greenwood D, Ghanbarim, et al. Hierarchical network management: a scalable and dynamic mobile agent-based approach [J]. Computer Network, 2002, 38(8):693-711

[7] Zhu Y, Chen T, Liu S. Models and analysis of trade-offs in distributed network management approaches [A] // Integrated Network Management Proceedings [C]. Seattle, WA, USA,

[8] Yoshihara K, Isomura M, Horiuchi H. Dynamic load balancing for distributed network management [A]// Integrated Network Management[C]. 2003:277-290

[9] Giladi R, Gat M. Meta-management of dynamic distributed network managers [A] // IFIP/IEEE SOM [C]. Texas, USA, 2000:119-131

[10] Sinnen O. Task scheduling for parallel systems [M]. Hoboken, New Jersey: John Wiley & Sons, 2007:89

[11] 高隽. 神经网络原理及仿真实例[M]. 北京:机械工业出版社, 2003:112

(上接第 84 页)

上^[11]对 LCESM 与 LPS^[7]在传感器监听成本和匹配次数上进行了实验比较。

• 实验环境

会议室均匀分布着 20 个温度传感器,每个温度传感器以 5 次/s 的速度产生数据。10 个移动应用在时间段 $T=60s$ 内发出订阅请求:订阅最近的温度传感器。且在 T 内,每个移动应用随机地沿着不同曲线移动,应用每隔 2s 需要更换事件源。

• 实验结果

如图 8 所示,LPS 中的所有温度传感器在时间 T 内不断产生数据。而在 LCESM 下,在时间 T 内有些温度传感器被多个应用共享,平均只有 6~7 个温度传感器产生数据,监听成本比 LPS 低很多。如图 9 所示,在 LCESM 下,只有在当前订阅需求和事件源匹配失效的情况下才进行重新匹配,与 LPS 中一次发布对应一次匹配的情况相比,匹配次数大幅度减少。随着时间的推移,总匹配次数增加缓慢。而 LPS 中总匹配次数随着时间的增加沿呈斜率为 100 的直线增加,增加速度快。

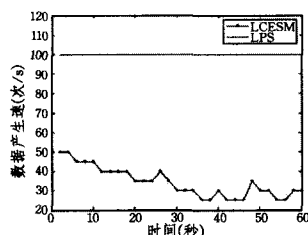


图 8 上下文数据产生速率比较

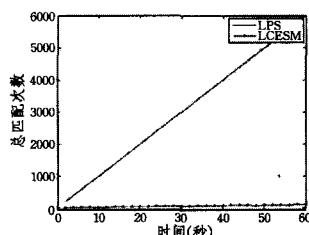


图 9 总匹配次数比较

结束语 本文给出一种位置敏感的上下文事件订阅机制(LCESM),该机制包括位置敏感订阅语言 LACL 和动态绑定方法。基于 LACL 定义了位置事件和事件源事件,两者分别使系统具有位置感知能力和事件源感知能力;为适应普适计算的动态性,动态绑定方法完成上下文事件源到订阅的自动映射,将事件与订阅的匹配有效地转化为事件源与订阅的匹配,从而减少了匹配次数和提高了系统性能。此外,绑定 agent 与订阅的动态绑定使系统具有良好的可扩展性和性能,绑定 agent 对订阅的动态操作使系统避免了多余的上下文事件的监听成本和网络通信量,更好地支持了事件源与订阅的动态绑定,为应用提供了满意的、位置更敏感的上下文信息服

务。

下一步需要从以下几方面改进:1) 改进 LACL 语言,使之具有更强的表达能力,以支持更复杂、位置敏感的复合上下文订阅。2) 需要针对不同计算环境的特点,解决如何处理订阅、事件等语言的扩展问题,解决适应不同应用程序需求的服务器模块的管理问题,解决不同计算环境下的异构协议问题。

参 考 文 献

[1] Unhelkar B, Murugesan S. The Enterprise Mobile Applications Development Framework[C]//IT Pro. May/June 2010

[2] 马建刚,黄涛,汪锦岭,等. 面向大规模分布式计算发布订阅系统核心技术[J]. Journal of Software, 2006, 17(1):136-147

[3] Huang Yong-qiang, Hector G-M. Publish/Subscribe in a Mobile Environment[J]. Wireless Networks, 2004, 10:643-652

[4] Fiege L, Mühl G, Felix C, et al. Modular event-based systems [J]. The Knowledge Engineering Review, 2002, 17(4):359-388

[5] Fiege L, Gartner F C, Kasten O, et al. Supporting Mobility in Content-based Publish/Subscribe Middleware[C]// Endler M, Schmidt D, eds. LNCS 2672. 2003:103-122

[6] Cugola G, de Cote J E M. On Introducing Location Awareness in Publish-Subscribe Middleware [C] // Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops(ICDCSW'05). 2005

[7] Eugster P T, Garbinato B, Holzer A. Location-based Publish/Subscribe[C]//Proceedings of the 2005 Fourth IEEE International Symposium on Network Computing and Applications (NCA'05). 2005

[8] 闫新庆,尹周平,熊有伦. 支持移动客户应用的分布式订阅/通知框架[J]. 计算机科学, 2008, 35(6)

[9] 彭禹,苑洪亮,吴泉源. 基于内容的发布订阅中支持订阅者移动的事件迁移算法研究[J]. 计算机研究与发展, 2007, 44(Suppl.):67-72

[10] Meier R, Cahill V. On Event-based Middleware for Location-aware Mobile Applications[J]. IEEE Transactions on Software Engineering, 2010, 36(3)

[11] Liu Han, Ye Jian, Zhu Zhen-min, et al. Research of Mobile Applications-oriented Peer-to-Peer Context-aware Mechanism[C]// International Conference on Computer Application and System Modeling (ICCASM 2010). Shanxi, Taiyuan, October 2010: 121-127