移动 agent 应用于网络管理

On Mobile – Agent Applying in the Network Management

王志勇　杨振启　（曲阜师范大学计算机科学学院、曲阜师范大学网络中心　276826）

摘要：本文介绍了基于移动 Agent 和 snmp 的网络管理系统的设计与实现，并给出了部分代码。Web 环境下开发，使得管理人员可以脱离网管站地理位置的束缚。分布式的思想，减轻了网络和主机负担。

关键词：网络管理　移动 Agent　SNMP

1　基于移动 agent 网络管理框架

移动 agent 是一个具有自主移动、自主学习的实体。它具有一定的智能，一定的对周围环境的适应能力，并能执行对环境产生影响的行为。在人工智能领域中，Agent 是具有自主性、协同性、相应性、预动性。在分布式系统中，它就是一个能够移动的实体。这两个概念相互补充，共同构成了 Agent。Agent 有在网络中移动就需要有它的发起端和接收端。这两个终端都要配有 agent 的运行环境，这样才能完成 agent 的动作。

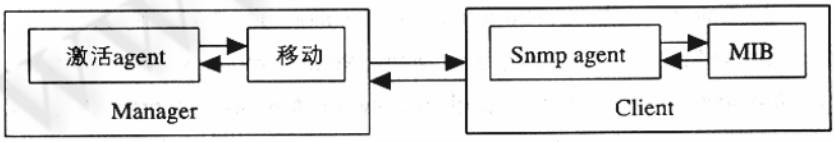

图 1　移动 agent 网管系统的框架

简单网络管理协议（SNMP）是目前网络管理的一个事实上的标准。以前有很多网络管理协议，他们都配有非常复杂的指令集，通过指令的操作实现复杂的管理操作。随着网络的发展，SNMP（基于简单指令集的协议）逐渐成为主流。它有 3 个版本：v1、v2c、v3。SNMP 的指令有：get – request、get – next – request、get – bulk – request、response、set – request、inform – request、trap。归结起来就是存和取两种操作。SNMP 通过对管理信息库（MIB）的存取实现对设备的管理。管理信息库（MIB）是由设备系统软件维护的、能反映设备详细状态的对象的集合。MIB 是一个树形结构，每一个节点都有自己的 OID，通过 OID 我们可以很方便的存取每一个对象的内容。如果改变了某个对象变量的内容，系统软件便可以根据我们的设置改变系统配置，从而达到管理的目的。

下面给出用移动 agent 进行网络监控管理的软件框架，可以让我们从总体上对这类软件有个大概的认识，见图 1。

网管站的任务是激活移动 agent 并且在 agent 被派遣以前完成对 agent 目的列表的初始化，agent 便可以根据此列表完成自动迁移。另外，对 agent 返回的数据处理和可视化。在这里为了表达软件的功能我们把网管站称为 Manager，把被管理的设备称为 Client。也可使用其他称谓。

2　系统实现

2.1　基本思想

在这里给出了程序的流程图，帮助我们理解程序的基本思想；分别是管理端和被管理端程序流程，见图 2。

程序按照分布式的观点，把轮训式网络管理的必须要在一台机器上完成的工作，分配到多台机器。相当于把复杂度分配给了各个站点，包括管理站点和被管理站点。这样便可以保证机器的性能及减小网络的负担（系统 agent 中可以把冗余的信息处理简化，使传输的信息量尽可能减少，所以网络负担减小）。

2.2　详细设计

由于有的设备能安装移动 agent 执行环境（如 pc），有的则不能安装移动 agent 执行环境（如路由器）。针对网络中存在这两种设备，我们设计了一个

移动 agent 的网络管理系统模型。构成图如图 3。

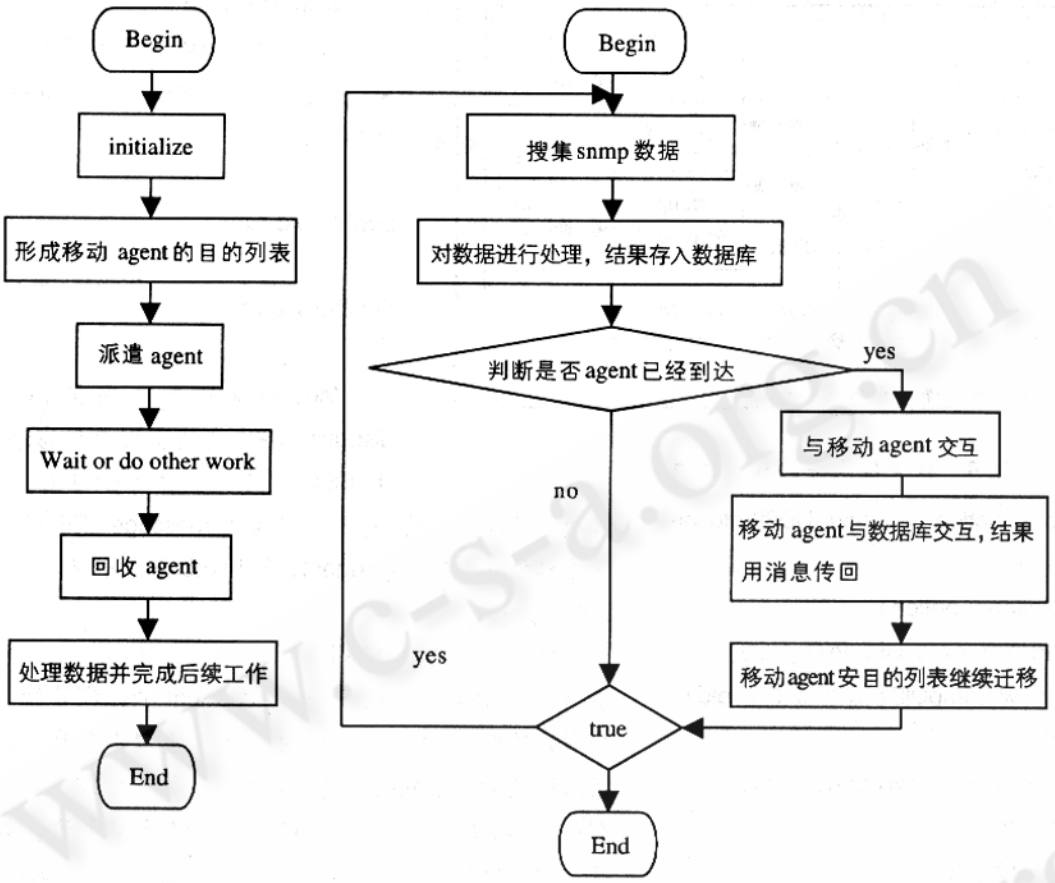

图 2　管理端和被管理端程序流程

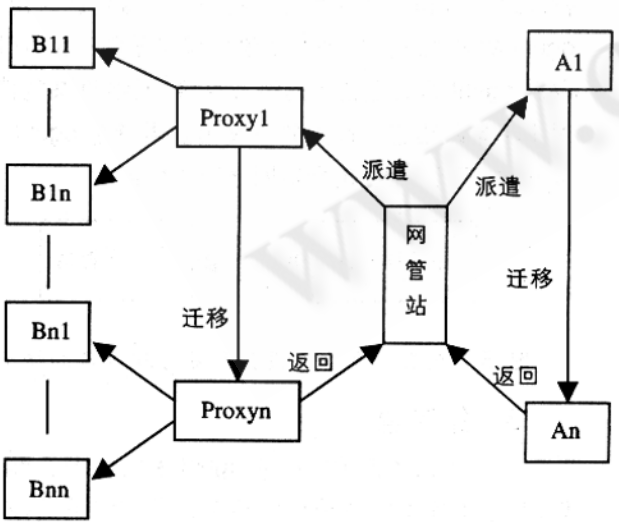

图 3　系统构成图

其中 An 代表的是具有移动 agent 执行环境的网络设备，Bnn 代表不具备移动 agent 执行环境的网络设备。因为它不具备移动 agent 的执行环境，所以要管理这种设备就需要有能安装移动 agent 执行环境的设备作为代理（Proxyn），在代理上装有移动 agent 的执行环境，并且有一个系统 agent 运行，见图 4。

以上为被管理设备的程序模块的构成示意图，其中 snmp 数据采集主要的功能是查询被管理设备的 snmp agent 的管理信息库（MIB），把信息送与系统 agent，agent 收到信息后，便对信息进行处理，处理后把数据存入数据库，同时信息采集部分也可根据需要与数据库进行交互。当移动 agent 到达后，移动 agent 可对数据库查询。关于移动 agent 的结构，文献[3]中已经给出了详细的模型可以参考。

2.3　部分实现

与 snmp agent 交互，现在有很多开发包可以选择，比如 NET－SNMP、adventnet snmp api 等等，我们采用的是 advent snmp api；可以采用的语言也有很多 perl、php、c、java 等等。为了便于扩充和便于移植，我们采用 java 作为开发语言：

```
SnmpAPI api = new SnmpAPI( );
api. setDebug( true );
session. open( );
SnmpPDU pdu = new SnmpPDU( );
```

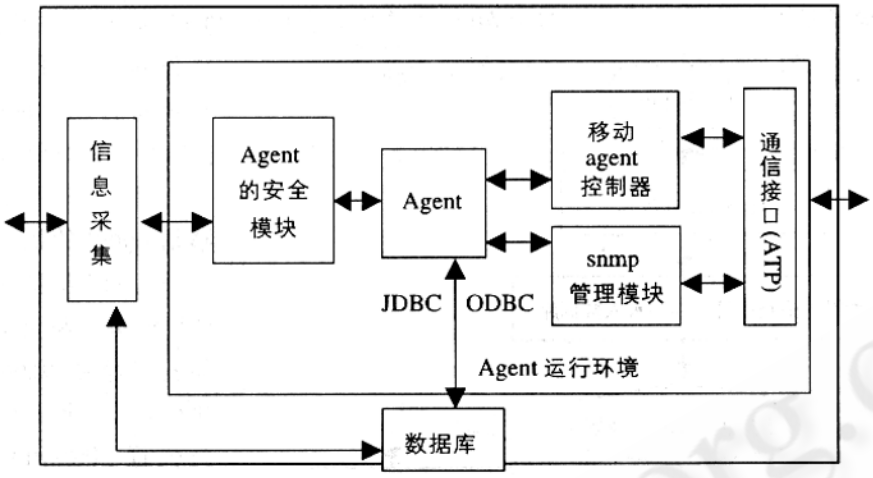

图 4　被管理设备的程序模块

UDPProtocolOptions option = new UDPProtocol-Options(远程主机地址);

pdu.setProtocolOptions(option);

pdu.setCommand(SnmpAPI.GET_REQ_MSG);

SnmpOID oid = new SnmpOID(需要查询的 oid);

pdu.addNull(oid);

SnmpPDU result = session.syncSend(pdu);

把得到的结果存入数据库;

继续进行下面的数据采集;

以上实现了 snmp 的数据采集操作,为了实现其他操作可以在以上程序的基础上添加其他代码,从而实现更复杂的功能。

对于移动 agent 我们这里采用 Aglets 进行开发,并采用 Sequential Itinerary 模式,下面的代码部分实现了网管站端的系统 agent:

public void onCreation(Object init) {

定义移动 agent 目的列表;

……

URL origin = getAgletContext().getHostingURL();

//创建移动 agent,进行移动

getAgletContext().createAglet(getCodeBase(), MobileAgent 在 CodeBase 中的位置,new MobileAgent(origin, destinations));

}

public boolean handleMessage(Message msg)

{

判断 Message 的种类;

解析 Message 的内容;

根据 Message 来了解相应信息(概念)或产生相应动作(agent action);

}

public void run()

{

doOtherWorks();

}

MobileAgent 类在内部调用 dispatch(URL)方法实现自身移动,用 JDBC 完成与数据库的交互,实现数据的简单处理,把最终结果用 Message 的形式发送给网管站端的系统 agent,通过 web 显示出来。

2.4 讨论与分析

用这种方式也并不是完美无缺的,其中一个缺点是,移动 agent 的体积问题,移动 agent 的体积要做一个很好的控制,不能太大。体积太大不仅不会减轻网络负担还会增加网络传输的开销。关于这个问题我们在系统 agent 中有一个模块是把 SNMP 信息进行处理,使其尽可能的简化,这样就可以减少在移动 agent 中携带的信息量。并且采用消息机制,把一部分数据通过消息发回网管站。另一个问题是当网络中的设备很少时,可以取一个极限情况来分析:网络中只有一个被管理设备和一个网管站。我们可以看到这时用移动 agent 的方式与用轮训方式管理网络差别不大,所以说移动 agent 方式适合于网络节点多,网络复杂的情况,对于小型网络它的优点并不突出。再就是 agent 的密度问题,密度过大,也会对网络造成负担。

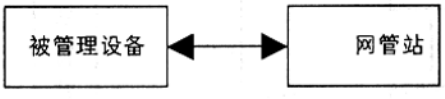

图 5　只有一个管理设备和被管理设备的情况

3　结束语

以上给出了详细的移动 agent 系统模型及部分实现,基于本模型的系统可以用来进行网络的监控和管理,特别是进行流量的监测,能很好的减轻网络负担。

(下转第 31 页)

（上接第 28 页）

参考资料

1 http://www.adventnet.com.cn/documents/snmpapidocment.

2 张云勇、刘锦德编著，移动 agent 技术，清华大学出版社，2003。

3 李碧蓉、肖德宝，基于智能移动 Agent 的网络管理思想模型的研究，小型微型计算机系统，2001。

4 D. Grimshaw Artificial Intelligence topics with agents, fall 2001。

5 马俊涛、刘积仁，Mobile Agent 体系结构及关键技术讨论[J]，小型微型计算机系统，1998。