

Internet 发展方向之一:主动网络^{*})

Active Networks: A Promising Direction in Internet

陈晓林 李 冀 陆桑璐 陈贵海 谢 立

(南京大学软件新技术国家重点实验室 南京 210093)

Abstract The architecture of traditional networks has limited the development of new protocols and applications and the evolution of Internet infrastructure. The emergence of active networks provides a possible approach. Active network is a programmable network in which user can insert customized programs into the switches of the networks. As a kind of novel network architecture, active networks are advantageous in creating and tailoring new network services and improving the flexibility of networks, which makes it one of the promising directions in the Internet. In this paper, we discuss the background, principles and architecture of active networks, as well as the applied fields and the problems to be solved in the future.

Keywords Active networks, Service, Execution Environment, Node OS, Active package

1. 引言

随着 Internet 的迅速发展,用户对网络服务的需求变化快,在需要高带宽的实时多媒体通信和只需低带宽的无线移动计算两方面的需求都很显著。尽管 Internet 的带宽增长速度十分迅速,但由于目前 Internet 体系结构的限制,网络提供商无法针对用户需求变化而及时地提供相应服务,主动网络(Active Network)的出现为解决这一难题提供了新的思路^[1,2,4]。

2. 主动网络的产生背景及基本思想

传统层次型网络体系结构,对等层间的通信需要采用相同的协议。为了能使各种计算机互联,就要制定一些标准协议。新协议要能广泛应用于 Internet 都需要一个前提:通过标准化过程成为一个标准协议。然而,满足用户新需求的新协议从开发出来到成为一个标准化协议再到广泛应用这段时间,是相当长的,IPV6 就是一个典型的例子。当新协议广泛应用时,用户需求可能已发生变化,使它实际上不能满足用户需求。所以说,传统网络体系结构的特点大大限制了新协议和新服务应用到 Internet 的速度和 Internet 基础设施的进化速度。

主动网络标准化的只是通信模型,而不是一个个的通信协议。通信模型同时支持多种协议而不是仅支持标准化了的协议,这样,新协议、新服务可立即应用到整个主动网络,而无需经过工业界的标准化过程。主动网络的“主动”表现在将用户定义的计算引入到路由器/交换器。传统的路由器/交换器只对经过的包做存储、转发,对包内的语义不做分析、理解,而主动路由器/交换器除对经过的主动报文做存储、转发,还将对主动报文内的数据进行处理甚至执行主动报文内携带的程序。用户(网络提供商或普通端用户)可以向主动路由器/交换器插入用户定义的程序,这些程序可以:①动态引入、配置、定制新服务;②协同端应用计算。

3. 主动网络体系结构

主动网络由互联的主动节点组成,这些主动节点为主动

报文中的“小程序”提供执行环境,当主动报文到达主动节点时,报文中的程序或代码就被激活执行,改变报文自身的内容或主动节点的环境状态,从而实现报文处理或网络服务的配置。一个主动节点由执行环境 EE(Execution Environment),节点操作系统 Node OS(Node Operating System)组成^[5]。传统网络体系结构定义了网络节点间如何相互协调来提供网络服务,而主动网络只定义了主动节点可编程接口的基本功能,而不涉及可编程接口的实现和采用什么编程语言。另外,为了简化 EE 能移植到多种低层操作系统,还定义了 EE-Node OS 的接口。

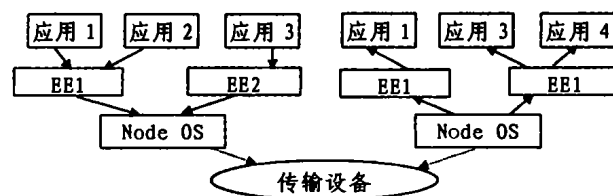


图1 主动节点体系结构

3.1 主动节点体系结构

EE 负责用户-网络接口的所有方面,包括用户发送的主动报文的语法、语义,提供的编程模型和抽象,地址和命名机制。每个 EE 向用户提供一些 API,例如一个扩展的 Java 虚拟机。多个 EE 可以运行在一个主动节点上,Node OS 管理可用的资源和控制资源竞争,EE 访问节点资源都必须通过 Node OS,Node OS 还实现一个安全策略数据库和一个执行引擎(enforcement engine)来控制安全使用资源,另外,Node OS 支持对 EE 有用的抽象,如路由表。用户通过与 EE 的交互来获取服务。一方面,用户可动态获取并组合 EE 提供的服务,另一方面,用户可通过 EE 提供的 API 来编写自定义的服务,并将新服务加载在 EE 上。

所有主动节点的操作模型都是相同的,如图2所示^[1]。包到达物理链路后(如 Ethernet, ATM),Node OS 根据包头将包送到合适的逻辑通道,每一个逻辑通道做相应的协议处理(如 TCP, UDP, IP),然后将包递交给 EE 解释和处理或者递交给逻辑输出通道。

^{*})本文得到国家 863 高技术项目(863-306-ZT02-03-01)的资助。陈晓林 在职硕士生;李 冀 硕士生;陆桑璐 副教授;陈贵海 教授;谢立 教授,博士生导师,研究领域均为分布式处理和并行计算。

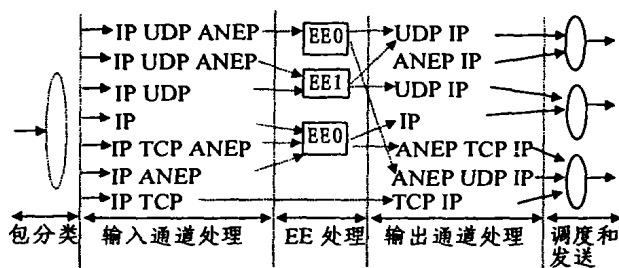


图2 主动节点的操作模型

3.2 执行环境(EE)

EE类似于操作系统中的“shell”程序，负责实现用户-网络接口。每种EE定义了一个用于解释、处理主动报文的虚拟机，可以将IP的实现看作是一个简化的EE。典型的EE实例有：(1)MIT开发的ANTS^[6]。ANTS在网络节点上，提供一个扩充的Java虚拟机，用于解释、执行主动报文中用户定义的Java程序。ANTS中，每个主动报文称为一个Capsule，每个Capsule由一段程序和数据封装而成。协议是用户自定义服务和代码分发的单位，它将相关的一些Capsule(这些Capsule的代码合作完成某任务)分成组，每类Capsule有自己的协议标识和类型标识。当Capsule到达主动节点后，ANTS检查其协议标识和类型标识，然后在代码缓存(Code Cache)中查找是否有处理该Capsule所需的全部代码，如没有则通过按需加载代码的方式向上一个节点请求下载代码。处理该Capsule所需的全部代码都加载后，运行Capsule中用户定义的Java程序。ANTS还提供软状态缓存(soft-state cache)给Capsule中定义的用来存储Capsule中的数据，计算所需的一些状态信息等。(2)SwitchWare。SwitchWare用PLAN^[7]向用户提供一个可编程接口，主动报文内的移动代码用PLAN写成。PLAN提供强类型和静态类型检查并保证任何程序都能终止。主动报文注入到网络后，报文传递路径中的SwitchWare节点动态加载主动报文内的代码，进行安全审计和程序验证，执行主动报文中的程序，报文传递路径由主动报文中的移动代码控制。另外，SwitchWare提供向网络节点显式加载代码，但这与前述主动报文的传送相分离并要接受严格的安全性检查。

3.3 节点操作系统(Node OS)

Node OS接口定义了4个主要的抽象：线程池、存贮池、通道和流(flows)。前三个封装的是三种系统资源：计算、通道和带宽，第四个用于前述三者的控制和调度。创建一个线程池需要指定3个参数：线程池的最大线程数目、循环周期(circle rate)和调度策略。通道分为输入、输出和直通通道(cut through)。流是Node OS中身份验证、访问控制、资源调度的主要抽象。一个典型的流包括一个输入/输出通道、一个线程池、一个存贮池。主动报文通过输入通道到达EE后，EE使用流中分配的线程和内存来处理主动报文，处理完后将其送到输出通道。流可以包含子流，形成一个流层次。流层次结构中，根节点对应于Node OS，第一级节点对应于EE，其余节点对应于EE创建的子流。这样的结构可以支持复杂的主动网络应用。Node OS的实例如Bowman^[8]，可运行于SunOS(5.5版及以上)和Linux(内核2.2.0及以上)。

3.4 互操作性

主动网络并不是要取代现在的Internet，而是通过隧道技术将主动网络叠加在Internet上，通过Internet实现互操

作性。ANEP^[9]封装主动报文，ANEP的报头定义了处理该主动报文的EE类型，ANEP封装的报文进一步被IP封装，通过IP路由到达目的主动节点。在从源主动节点到目的主动节点的路径中，非主动路由器对主动报文的处理同对IP报文的处理。

4. 应用领域

主动网络主要可应用于以下4个领域：

1. 可编程的网络基础设施：为了能适应用户需求的变化，网络提供商希望能针对用户需求快速开发运用新网络服务和动态管理网络服务，并且在增加新服务，删除、升级、组合现有服务的同时不影响当前提供的服务。主动网络是一种可编程的网络，非常适合用来解决网络基础设施的可编程性、开放性、可扩展性问题。

2. 网络管理：传统的网管方法是：网管工作站向被管设备发探询消息来获取被管设备状态，然后根据探询到的信息再做出反应。这种方法所有的智能都集中在网管工作站，易造成通信和处理瓶颈，并严重制约了跟踪解决问题的快速性而且效率不高。主动网络为解决上述问题提供了新方法，主动探询报文探询到被管设备有问题时，可就地做出采取何种动作的决定并通知被管设备。

3. 无线移动计算：由于无线瘦客户机只需低带宽，所以可以将计算引入到支持无线瘦客户机的网络基础设施中，这些计算一方面用于解决无线瘦客户机的漫游和远程访问服务问题，另一方面可为无线瘦客户机提供个性化服务及与无线瘦客户机协同计算。

4. 分布式端到端应用：典型的例子有组播。传统的端到端差错控制和流量控制方法由于无法利用组播树而很难解决NACK/ACK风暴(Negotiate Acknowledge/Acknowledge Implosion)、选择重发、分布恢复负担、拥塞控制、健壮性这五个问题^[11]，主动网络的出现使组播协议可利用组播树中的主动路由器协同端实现差错控制和流量控制，从而给上述五个问题的解决提供了一条新途径。其它例子有诸如MIT开发的在线拍卖系统^[10]等。主动网络的支持可以有效提高一些分布式端到端应用的效率。

结束语 主动网络有很多优点：对于网络提供商，可缩短运用新协议/服务的时间；对于端用户或第三方，可动态引入、定制、配置服务及协同端应用计算；对于研究者，可在现有的Internet上实验新协议/服务而不影响Internet原有的服务。但主动网络方面的研究刚起步，还有很多问题需要解决。涉及到的方向有：EE中的可编程计算模型；如何保证主动路由器在为用户提供自定义服务时并能快速路由转发IP包；能提供QoS的Node OS；EE和Node OS的安全性；主动网络在前述4个领域的应用等。

参考文献

- 1 Tenenhouse D L, Wetherall D. Towards an Active Network Architecture. In: Proc. Multimedia Comp. and Networking 96, MMCN' 96, San Jose, CA, Jan 1996
- 2 Tenenhouse D L, et al. A Survey of Active Network Research. IEEE Communications Magazine, 1997, 35(1): 80~86
- 3 Calvert K L. Directions in Active Network. IEEE Communications Magazine, 1998, 36(10): 72~78
- 4 Smith J M. Programmable Network Selected Challenges in Computer Networking. Computer, 1999(Oct.): 40~42
- 5 Smith J M, et al. Activating networks, a progress report. Computer, 1999, 324: 32~41

分布式 Web 服务器技术综述

A Survey on Distributed Web Server Techniques

马晓星 吕建

(南京大学计算机科学与技术系 软件新技术国家重点实验室 南京 210093)

Abstract With the explosive growth of the World Wide Web, many popular Web sites are faced with the challenge of the overload of tremendous requests. The best way out is using distributed Web server systems for their good scalability and low costs. In this paper, we try to give a comprehensive survey on the underlying techniques of the distributed Web server systems: request-dispatching mechanisms, load-balancing algorithms, Web content replication and distribution schemas and other important aspects.

Keywords Web Server, Distributed System, Survey

World Wide Web (WWW, 或简称 Web) 是一个基于 Internet 的超文本分布信息系统。自十来年前诞生于欧洲粒子实验室以来, 它以难以置信的速度迅速普及, 现今许多重要的信息服务都通过 Web 来提供。随着 WWW 规模的爆炸性增长, 许多 Web 站点的访问量都大大超出了普通 Web 服务器所能承担的范围。Web 服务器的过载会导致服务质量下降(响应时间过长, 请求丢弃率过高等), 甚至会使服务器宕机。

解决这一问题的直接办法是升级 Web 服务器的硬件, 采用处理能力更强的计算机系统, 如使用更快的 CPU、SMP 并行体系结构等。然而, 这种方法的扩展性有限, 性价比不高。更好的办法是使用分布式 Web 服务器系统。而若使用地理上分布的 Web 服务器系统则不仅能扩展 Web 服务器的客户请求处理能力, 而且还可以节约网络带宽这一宝贵资源。

迄今为止, 分布式 Web 服务器系统已经有许多成功的应用^[1]。然而, 要让一组 Web 服务器像一个服务器一样可靠并且高效地工作, 有不少问题需要仔细考虑, 其中某些问题还有待进一步研究。V. Cardellini 等人在文[2, 18]中讨论了几种不同的分布式 Web 服务器及其负载均衡机制。本文将从一个更宽广的视角对各种分布式 Web 服务器技术做一个更全面的介绍和分析, 特别是那些文[2, 18]中未曾收入的新近的工作。

1. 分布式 Web 服务器

1.1 WWW 与 Web 服务器

WWW 采用 Client/Server 的基本结构, 其基本工作原理非常简单。促使 WWW 获得全球性广泛流行的关键技术因素有三: 一是一个简易友好的超文本语言 HTML, 二是一个满足全球唯一性的命名规则 URL (更准确地说应是 URI (统一资源标识符, 参见 RFC1630)。URL 是 URI 的一种形式, URI 还包括 URN (统一资源名) 等。本文只涉及到 URL), 三是一个轻便有效的通信协议 HTTP。

HTML, 全称为超文本标记语言, 是有史以来最成功的标记语言——有人估计 WWW 上有 1000 亿个 HTML 页面。HTML 本身规模较小且易于移植, 可以很容易地在各种计算平台实现。HTML 可以在二维空间上展现文本或其他对象。HTML 也很简单易学, 易于为用户接受。然而最重要的一点是 HTML 的超文本特性: HTML 页面中可以任意嵌入指向其他文档的指针(称为超链接), 用户只需点击一下鼠标, 就可以顺着超链接访问到也许位于地球另一端的另外一个页面。这构成了 WWW 的基础。

作为一个全球性的公共信息系统, WWW 需要一个系统的规则来命名其上的各个资源(包括 HTML 文档和其他对象), 每个资源的名字应是全球唯一的。统一资源定位符 URL 利用服务器的名字来区分不同服务器上的资源, 对于同一个服务器上的资源则用类似于 UNIX 文件系统的树形目录来命名。URL 还指明了访问该资源的协议和服务器端口, 对于 WWW 来说通常是 HTTP 协议, 端口缺省为 80。

超文本传输协议 HTTP^[9] 是一个用于分布协作超媒体信息系统的应层协议, 是 WWW 上 Web 服务器、代理服务器、浏览器等实体进行通信的主要协议。一个简单的 HTTP 交互的工作过程如下: 浏览器首先和服务器建立一个 TCP 连接, 再从这个通道向服务器发出一个 HTTP 请求, 服务器再从这个通道发回他对这个请求的 HTTP 响应, 最后连接关闭。HTTP 的一个重要特点是无状态, 也就是各次 HTTP 交互之间是相互独立的。这意味着不论这些 HTTP 交互在实际内容上是否相关, 都可以将它们分散到多个镜像服务器上进行处理, 而对用户来说其效果与一台服务器是一样的。

实际上用户访问一个 WWW 资源的过程可能稍微复杂一些。图 1 展现一个典型的访问过程。用户要访问 <http://www.w3.org/History.html>, 他的浏览器首先要获得 www.w3.org 的 IP 地址, 于是就向本地的 DNS 发出一个域名解析请求; 本地的 DNS 发现 www.w3.org 不是本地的域名, 且本

6 Wetherall D, et al. ANTS: A toolkit for building and dynamically depolying network protocols. In: Proc. IEEE OpenArch' 98, San Francisco, CA; Apr. 1998
7 Hicks M. PLANA—Programming Language for Active Network Architecture. <http://www.cis.upenn.edu/~switchware/papers/plan.ps>, 1997
8 Merugu S, et al. Bowman: A Node OS for Active Networks. IEEE INFOCOM' 2000, 2000

9 Alexander D S, et al. Active Network Encapsulation Protocol (ANEP). <http://www.cis.upenn.edu/~switchware/ANEP/docs/ANEP.txt>, 1997
10 Legedza U, et al. Improving the performance of distributed applications using active network. IEEE INFOCOM' 98, 1998
11 陈晓林, 等. 一个基于主动网络的大规模可靠组播协议的设计和实现. 电子学报, 录用