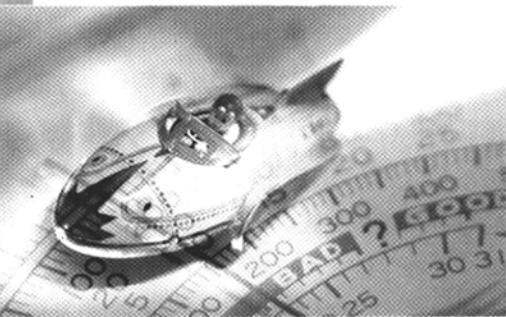


Jini 与 EJB 技术比较及其应用



朱茜 陈福民 徐冰 (上海 同济大学计算中心 200092)

摘要: 本文在研究分布式计算必要性的基础上, 通过对 Jini 结构特征的探讨, 比较了 Jini 和 EJB 的异同点, 并就其提出了一种综合应用的实例。

关键词: 分布式计算 Java Jini EJB (Enterprise JavaBeans)

1 分布式计算及 Jini 技术的提出

随着 Internet 的日益发展, 推动 WEB 发展的公司纷纷把认识转向“通用网络”的概念。由此而来, PC 的增长率会下降, 因为并不是每个人都想在他们的办公桌上放一台安装了 Windows 或 MacOs 的个人计算机, 而是希望能够通过手机或电视机来访问 Internet。目前广泛应用的 WAP 手机和机顶盒就满足了这种需求。但是, 它们基本上只是将 PC 的功能移植到了另外的设备上。

而“分布计算”不仅将 WEB 的标准能复制到嵌入式设备 (embedded device) 中, 而且还能将该设备提供给 Internet 上的其他实体。

近年来, Java 以它优秀的平台搭建和集成技术, 以及自己独有的“一次编写, 到处运行”的特点, 给网络带来一股新气象。纵然用 Java 可以编写在各种不同平台上运行的代码, 但是, Java 并未能正面应对各种不同平台的集成和综合问题。例如想把数字相机接到某台计算机上, 就必须有相应的驱动程序, 就像 Windows 上的驱动程序在 UNIX 环境下不能用, 甚至不同版本的 Windows, 有时也要求不同的驱动程序。

针对这个问题, Sun 公司推出了它的另一核心技术——Jini。建立在 Java 软件基础结构之上的 Jini 技术可以使所有类型的数字设备, 在不需要大量的计划、安装或人为干预的情况下, 非常简单地连接到任意自发网络 (impromptu network) 上, 从而使得对新的网络服务的访问和发送变得就象接入电话一样简单, 并共同运行在一个社区 (community) 中。每个设备可提供该社区中的其他设备可能需要使用的服务。这些设备还提供了它们自己的用户接口或程序接口, 这就保证了可靠性和兼容性。

虽然加入 Jini 的服务既可以是硬件也可以是软件, 但 Jini 最终真正的魅力在于使硬件方便、自由地连入网络。Jini 电子设备是需要有一定的计算能力和存储能力的。因而它应有自己的处理器、存储器以及显示设备等专用的功能硬件。根据目前硬件的发展趋势: 高性能的处理器可以使用在最简单的设备中; 适当的存储器能与高性能处理器协同工作; 网络的带宽继续增加, Jini 假设 100Mb/s 到 1Gb/s 的网络将会很快普及; 存储设备将继续提高单位存储量与价格之比; 消费电子产品数字化, 这样 Jini 系统将不断消除消费电子产品、计算机、网络、Web 及娱乐消费设备之间的界限。

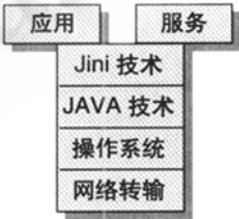


图 1

2 Jini 的技术架构

有别于以前的客户/服务器概念, Jini 提出了客户/服务 (client/service) 的模式。即网络上的服务动态地组成能完成一定功能的“联盟” (federation), 客户提出某种请求, 服务就可以完成相应功能。客户与服务是相对的, “客户”也可以向 Jini 系统中其他成员提供服务, 这时它也就成了“服务”; “服务”也可以有它的请求, 这时它就是“客户”。

在运行着的 Jini 系统中三个主要的角色: 服务 (例

如打印服务)、客户(使用服务的人或程序等)、Lookup 服务(它在 client 和 service 之间扮演代理、中介、定位的角色)。还有一部分就是连接这三部分的网络,该网络将运行 TCP/IP 和 Java RMI 协议。

工作过程如下图 2 所示:

第一阶段是服务用 Lookup 服务来注册自己。在这阶段不能用 RMI,因为它还不能定位 Lookup 服务,所以它使用运行在端口 4160 上的特殊 TCP 协议,发送一个 512 字节的小信息包。每个 Lookup 服务都将在这个端口监听。当 Lookup 服务在该端口接到一个请求后,它就利用 RMI 把一个对象发回到 Lookup 服务端。该对象就是 registrar 担当 Lookup 服务的代理功能,在服务端的虚拟机上运行。服务端需要使用 Lookup 服务都要通过这个代理注册器。这包括把 service 的拷贝存到 lookup 上。

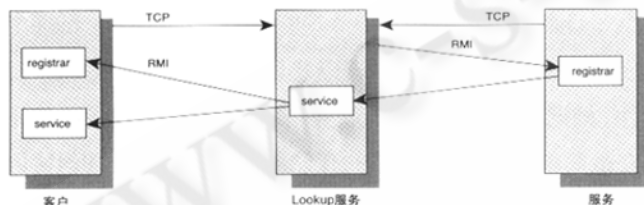


图 2

另一方面,客户通过相同的机制从 Lookup 服务上获得一个 registrar。但是这次和上面做的有些不同,它要求 service 要被拷贝给客户。在该阶段,有一个在服务提供者主机上运行的原始服务,同时还有一个存储在 Lookup 服务上的服务拷贝和一个运行在客户 JVM 中的服务拷贝。客户可以直接向运行在它自己 JVM 中的服务对象请求服务。

3 Jini 与 EJB 的比较

Jini 技术让我们想起 javasoft 公司在更早时期推出的 EJB (Enterprise JavaBeans) 技术。

EJB 是一种基于构件(Component)的服务器开发模型,也采用客户/服务结构。其基本思想是为构件开发提供一个框架/Framework,满足规范的构件可以方便、灵活地放在服务器端的 EJB 框架中,扩展服务器的功能,而不需要考虑移植等的细节,类似硬件的即插即用。此种技术建立在 Java 语言基础上,因此只需代码的一次实现而随处可用;而利用框架的设计思想,使开发者可方便地使用分布式对象的特点。

这两种技术都扩展了 Java 语言的应用,都采用客户/服务体系,分布式结构。但应用角度不同,我们可以以此为出发点,详细比较一下这两种新技术的不同点:

(1) EJB 的开发使用通常是在三层结构(客户机/应用服务器/数据库服务器)上。EJB 把所有应用程序的商务逻辑在服务器端执行,这样,EJB 构成了一种‘瘦’客户、‘肥’服务的模式;Jini 的应用中,客户只和一个或多个能提供给它服务的设备相连,所以,而 Jini 是一种点对点的‘客户——服务’模式。Lookup 服务一旦把提供服务的对象返回到客户,客户就可以在本地执行这个对象的方法,这样,服务处理就被分布化了。

(2) EJB 中服务器的对象协议用 JMS API 隐藏 EJB 代码,JMS API 提供了一种更高层次事件/消息通信层和其他过程通信。如果需要更直接的通信方式,EJB 就要显式存取对象协议,典型的对象协议就是 CORBA/IIOP 或 DCOM,如图 3 所示:

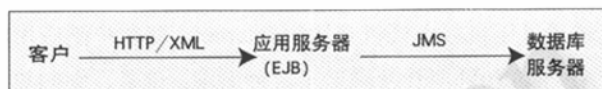


图 3

而 Jini 技术如前所述,客户用 RMI 和 Lookup 服务连接,服务用‘发现/加入’协议宣布它的存在,并在 Lookup 服务中登记它的对象,如图 4 所示:

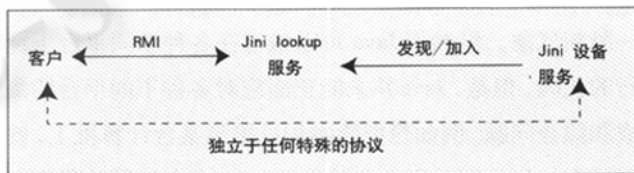


图 4

综上所述,Jini 和 EJB 技术的不同点可列为下表:

比较方面	Jini	EJB
系统模型	点对点连接 客户、服务可以是任何设备	三层结构(C/S/S) ‘瘦’客户、‘肥’服务
所需典型平台	支持 JAVA 的设备	JAVA 应用服务器、 JAVA 数据服务器
典型的拓扑结构	分布式的客户	分布式的服务企业
传输协议	TCP、RMI、设备的查找/加入	IIOP (CORBA)
优势	智能化地使用设备提供的服务	编写一次 EJB,可在各个企业中运行

可见,两种技术各有优势,完全可以综合应用,以一个客户携便携式电脑出差为例。

① 酒店的 lookup 服务器运用 Jini 技术,提供打印机、传真机等设备。

② 客户方的配置:客户同样用 lookup 服务器提供外部的安全存储点(诸如大容量硬盘),作为夜间备份;提供办公室的打印机,传真机;提供家中收邮件的 PC 等一系列客户所需设备。

③ 客户通过相应网络,下载 applet 来收集每天的开销情况,服务器端用相应的 servlet 来支持它的应用。

④ 通过验证, servlet 用 JNDI 来生成并定位一个用来把每天的开销情况更新到数据库的 EJB。

⑤ 客户可以把有关开销的报表存储在安全存储点中,在酒店提供的 Jini 打印机上打印出来,传真一份到其办公室的 Jini 传真机上。

4 Jini 技术的应用

以上例来看,客户通过查询某个 Lookup Service 来定位服务。为了实现这一点,首先必须找到 lookup service。另一方面,打印机等服务要在 lookup service 上注册,首先也要找到 lookup service。所以客户和服务的第一步都是要查找 lookup service。查找 lookup service 有两种方法:unicast (一对一) 和 broadcast (广播)。

显然,客户已经知道了 lookup service 所在的机器的准确地址,就可以用一对一的查找。编程使用包 net.jini.core.discovery 中的类 LookupLocator,它有两个构造函数如下:

```
package net.jini.core.discovery;

public class LookupLocator {

    LookupLocator(java.lang.String url)
        throws java.net.MalformedURLException;

    LookupLocator(java.lang.String host,int port);
}
```

第一个构造函数中的 URL 必须是这种形式的 "jini://host/" 或者 "jini://host:port/"。如果不指定端口号 (port),缺省端口是 4160。以上例来调用

```
import net.jini.core.discovery.LookupLocator;

public class InvalidLookupLocator
{
    static public void main(String argv [])
    {
```

```
        new ValidLookupLocator();
    }

    public ValidLookupLocator()
    {
        LookupLocator lookup;
        try
        {
            lookup = new LookupLocator("jini://hotel.com",
4160);

            System.out.println("Hotel lookup creation succeeded");
        }
        catch(java.net.MalformedURLException e)
        {
            System.err.println("Hotel lookup failed: " + e.toString());
        }
    }
}
```

5 总结

Jini 刚发布之时,美国的 Sun 公司已和日本的索尼、夏普、东芝、日立,荷兰的飞利浦等八家公司,联合研制开发基于 Jini 技术的家电产品。日本爱普生公司已成功开发出 Jini 打印机,安装在打印机上的 Jini 代码容量为 600kB。日本佳能公司也在 Jini 发布会上演示了 Jini 多功能彩色打印机试制品,并且在年内实现商品化。在硬盘驱动器 HDD 方面,世界三大著名的 HDD 厂家中有两家 Quantum, Seagate 已经推出支持 Jini 的 HDD 试制品。纵观当今的 IT 界,具有与 Jini 同样的技术的竞争对手寥寥无几,象 Microsoft 的 MTS 和 Millennium。但是, Jini 与其他技术相比有个不可比拟的优势,就是可以在适应 Java VM 的所有计算机上执行,这是它难以超越、不可战胜的法宝。■

