

Jini 核心机制——查找服务概述

程文英, 帅春燕, 吕杨

(昆明理工大学 信息工程与自动化学院, 云南 昆明 650051)

摘要: Jini 查找服务是 Jini 核心机制之一, 提供了在分布式 Jini 中服务的中央注册机制. 任何服务都是由查找服务发现并确定, 系统用户与系统的联系也是通过查找服务实现的. 本文对查找服务的作用、工作原理、使用方法进行了详细介绍.

关键词: Jini; 查找服务; 服务

中图分类号: TP393.04 **文献标识码:** A **文章编号:** 1007-855X(2003)01-0095-04

Jini Core Mechanism—Summarization of Lookup Service

CHENG Wen-Ying, SHUAI Chun-Yan, LU Yang

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650051, China)

Abstract: Jini lookup service is one of Jini core mechanism, which provides the service of the central register mechanism in the distributed Jini system. Any service is found and defined by lookup service, and by lookup service the communication between client and system is also implemented by it. In this paper, the important role, working principle and applied methods of lookup service are introduced in great detail.

Key words: Jini; lookup service, service

0 引言

Jini 技术的即插即用的基础结构主要由三组协议组成: 查找(Lookup)、发现(Discovery)和加入(Join). 查找服务是分布式 Jini 系统的基本组成部分, 提供了在分布式 Jini 系统中服务的中央注册机制. 在 Jini 系统中, 程序找到服务的首选方法是通过查找服务, 而且查找服务也是提供给管理员和用户各种访问接口的基础.

Jini 体系结构最重要的概念是服务, 一个服务是一个实体, 它能被人、程序或其它服务使用, 它可以是一次计算、存储、和另一个用户交流的通道、软件过滤器、硬件设备或另一个用户. Jini 系统中成员间的联合是为了对服务共享访问. 一个 Jini 联合不应被简单看成是客户机和服务器的集合, 或者是用户和程序的集合, 或者是程序和文件的集合; 它是由一组完成某一特定任务的服务组成, 服务也可以利用其它服务来完成某一任务. Jini 系统提供一种机制, 以在分布式系统中实现对服务的构造、查找、通信和使用, 同时这种机制使用服务协议(即一套 Java 接口)完成服务间的通信. 通过查找服务发现和确定服务, 同时还起到连接系统和系统使用者的作用.

1 查找服务工作原理

查找服务(Lookup service)是 Jini 系统的中心机制, 任何服务都是由查找服务发现并确定, 系统用户与系统的联系也是通过查找服务实现的. 查找服务相当于网络中所有服务的公告牌, 它把指定某服务所提供性能的接口与实现该服务的对象集对应起来. 查找服务中的对象也可以包含其它查找服务, 从而构成层次式的查找服务. 查找服务是为了用户在网络中查找和调用服务, 而服务是由它的接口类型(用 Java 编程

收稿日期: 2002-09-24.

第一作者简介: 程文英(1976~), 女, 在读研究生; 主要研究方向: 嵌入式系统及计算机控制.

语言编写)或其它属性描述的。

当 Jini 系统中的客户需要服务时,它根据服务的类型或属性向查找服务查询合适的服务,查找服务经查询后把结果返回给客户,当客户决定使用某一服务时,查找服务将把所需服务的服务对象的拷贝发送给客户(如图 1)。

客户通过该服务对象与服务提供者进行直接联系,调用服务.服务对象与服务提供者之间的私有协议,对用户隐藏了实现的细节(如图 2)。

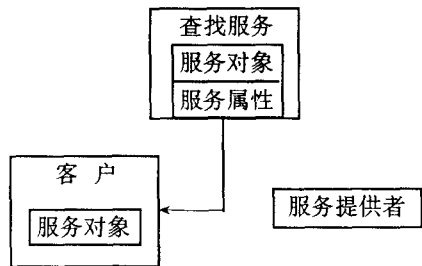


图 1 查找过程

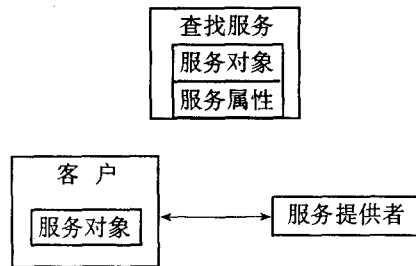


图 2 服务过程

1.1 查找服务的模型

查找服务保存了一个服务表,表中的每个元素代表了系统中可用的一项服务,其中保存了访问这项服务所需要的代码和一系列可扩充的描述服务的属性.打个比方,查找服务就像电话簿,Jini 系统中的服务就像电话簿中的用户,查找服务中保存的信息就是用户的电话号码.也就是说,凡是需要访问 Jini 系统中的服务,就必须知道查找服务中保存的“电话号码”.但是查找服务又不同于电话簿,例如查找服务保存的是一个动态的“电话簿”,即系统中的服务可以在“电话簿”中主动加入和删除自己的“电话号码”,查找服务完全支持这种机制。

在服务注册时,服务提供设置自身的用户界面,这个服务界面可以作为服务的属性保存到查找服务中,管理员可以轻松地设置服务,不用忙着到处找各种各样的驱动程序和管理软件。

当服务在查找服务中注册后,会产生一个通知事件,系统管理员可以利用查找服务中的事件回调机制来接到这个通知。

在服务的有效期内,如果服务遇到问题需要引起管理员的注意,查找服务将自动增加一个属性来指明问题,并将更改后的结果保存到查找服务中.然后,查找服务的事件回调机制向管理员发出通知,告知服务遇到问题.有了这个有力的机制,大大减轻了系统管理员的负担。

1.2 定位查找服务

客户通过查询某个查找服务来定位服务,为了实现这一点,首先必须找到查找服务.一个服务要在查找服务上注册,首先也要找到查找服务.所以客户查找服务和注册的第一步都是要搜索查找服务.搜索查找服务有两种方法: unicast(唯一)和 broadcast(广播).一对一查找(Unicast discovery),一般用于已知查找服务的准确地址,当你已经知道了查找服务所在的机器,就可以用一对一的查找;广播式查找(Broadcast discovery),如果查找服务的位置不知道,那就要用广播式查找了.一对一查找是只有已知地址的查找服务响应请求,而广播式查找是在网上全面搜索,网上能收到信息的查找服务都应该响应请求。

2 使用查找服务

Jini 查找服务为服务的提供者和使用者的提供了一组丰富但简单的 API.服务和客户都要使用查找功能,服务使用查找来存储自己的服务代理,通过为服务附加属性来描述这些代理;客户使用查找来搜寻服务,而且能在新的服务加入时得到通知,或者在查找的服务集发生变化时请求事件。

客户和服务都使用发现机制来寻找查找服务.客户一般直接使用发现的 API,而服务可以利用一些更高层的 API 来管理发现.当要向 Jini 中加入一新的服务时,服务提供者,比如一个设备或软件,首先向网络上发送请求,使任何的查找服务能识别它(如图 3).一旦查找服务发现这一服务后,此服务将上载一个服务对象,登记到查找服务中(如图 4).服务对象包含 JAVA 程序界面,此界面中有实现服务所需的方法及

有关属性. 服务提供者需有能力发现查找服务, 当然服务提供者也可交第三方代理查找.

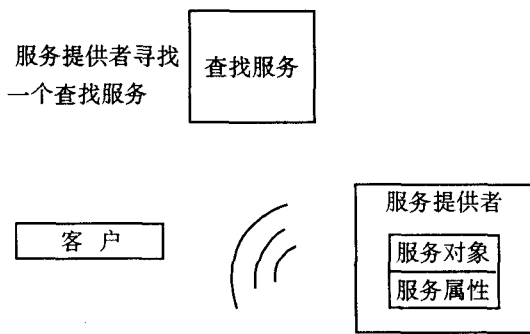


图 3 查找过程

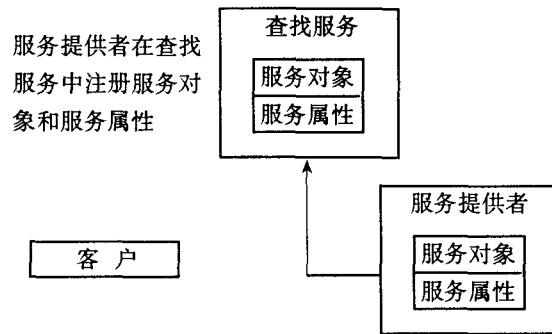


图 4 注册过程

2.1 服务使用查找服务

服务提供者发布自己的服务代理, 包括获得服务 ID 以及使用加入协议与查找服务交互. 每个服务都有一个唯一标识自己的 ID, 以区别于其它服务. 服务 ID 是 128 位长的标识符, 可唯一确定一个服务, 服务 ID 中的 60 位是把系统时钟从 1582 年以来的时间用 100 ns 的时间单位来表示, 加上许多随机噪声所得, 另外的 68 位是根据查找服务所在的唯一主机地址 (通常是 Ethernet 地址) 实现随机值得到. 当服务第一次连到查找服务时, 查找服务为它分配一个服务 ID. 服务 ID 要保持唯一, 服务在运行过程中以及在所有注册的查找服务上都要使用相同的 ID. 要求服务在每个查找服务中使用相同的 ID, 就可以判断具有相同 ID 的两个 ServiceItems 是代表相同的服务或设备, 以便于资源的有效利用或防止资源的使用发生冲突.

服务提供者与查找服务进行通讯, 服务提供者把自己的服务对象和服务属性注册到查找服务中, 也就是把服务对象和服务属性发送到查找服务中, 这个过程采用的是加入协议 (Join). 加入协议是一系列确定服务如何与发现和查找服务交互的规范. 加入协议的规则是通过限制何时以及怎样连接查找服务、如何存储属性和组以及连接哪些查找服务等, 使服务可能免于维护. 服务的最基本工作是在任意通过发现找到的 ServiceRegistrar (服务注册器) 上调用 register() 来发布代理, 但为了使服务能清晰地与其群体交互和使服务更易于管理和维护, 需要使用加入协议. 服务在使用加入协议时需要做很多记录, 服务需要永久 (即使是经过重起也无妨) 记住多项记录, 包括服务 ID、成员的组的集合、直接连接的查找服务集、以及附带的任意属性等.

2.2 客户使用查找服务

客户应用程序, 只使用服务, 不提供服务对象和属性. 客户可以搜寻一个查找服务中注册的任意服务, 最基本的是基于服务代理所实现的接口、基于附加到代理上的属性或直接使用服务 ID 的搜寻方式. 除了搜寻以外, 客户还可以请求查找服务在新的服务出现、原有服务改变或先前注册的服务退出等情况下向自己发通知. 大部分情况下, 客户与查找服务的交互局限于两个操作: 搜寻服务和请求事件.

2.2.1 搜寻服务

向查找服务提交一个描述被搜寻服务的模板, 查找服务返回与模板匹配的服务代理. ServiceRegistrar (服务注册器) 支持两种不同形式的 lookup() 方法来执行搜寻. 两种 lookup() 方法的格式如下:

```
Object lookup(ServiceTemplate tmpl)
    Throws RemoteException;
ServiceMatches lookup(ServiceTemplate tmpl, int maxMatches)
    Throws RemoteException;
```

第一个方法适用于使用模板机制可完全指定所需服务的客户, 如果有一个或多个服务匹配, 查找服务总是选择一个作为匹配的服务, 这个服务的代理对象被返回到调用者; 而另一个适用于客户使用模板不能完全指定要搜寻的服务的情况, 或用于“浏览器”风格的使用, 它们希望找到所有可用的服务. 这个方法不再只是返回一个服务代理, 而是返回一个 ServiceMatches 对象, 其中包含了一个与查询相匹配的全部 Ser-

viceItem 对象(包括代理、服务 ID 和属性)组成的数组。

2.2.2 从查找服务请求事件

通过搜寻可以查到哪些服务已在一个特定的查找服务中注册,请求查找服务可以在有新的服务出现时向客户发送事件,这个过程称为请求事件.通过请求事件,不仅在新的服务出现时客户可得到通知,在已有的服务退出或者服务上的属性集发生变化时,客户也可以被通知.客户还可以说明希望哪些变化出现时才被通知.

(1) 指定何时发送事件

使用 ServiceTemplate 来说明感兴趣的服务类型.在请求发送事件时,客户要提供一个特殊的标志说明如何使用模板,此标志可以是三个可能值中的任意值或全部,分别指示在哪些情况下发送事件,控制何时发送事件的“变化”标志的定义如下:

①出现与提供的模板相匹配的新的服务:

```
int TRANSITION_MATCH_NOMATCH = 1 < < 0;
```

②过去与模板相匹配的服务不再匹配:

```
int TRANSITION_NOMATCH_MATCH = 1 < < 1;
```

③与模板匹配的服务虽仍然匹配,但发生了某些变化:

```
int TRANSITION_MATCH_MATCH = 1 < < 2
```

第一种情况用于寻找特定类型的服务,或是具有特定关联属性的服务,客户希望在有这样的服务注册时通知.第二种情况用于检测服务从查找服务中注销的情况(可能是由于服务崩溃或租借过期),或者是服务的某个属性消失的情况.第三种用于使客户可随时观察到群体的最新状态.

(2) 使用 Notify()来请求事件

客户通过在 ServiceRegistrar 上调用 Notify()方法来请求事件.Notify()方法的定义如下:

```
EventRegistration Notify(ServiceTemplate tmpl, int transition,
    RemoteEventListener l, MarshalledObject data,
    Long LeaseDuration)
    Throws RemoteException;
```

Notify()方法的参数中 ServiceTemplate 描述匹配的模板,transition 标志指出如何使用模板,transition 的可取值已由上述指定何时发送事件的整数标志指定,RemoteEventListener 是一个 RMI 远程接口,用于处理从查找服务中接收的消息,MarshalledObject 指定一个任意已预先序列化的 Java 对象,希望在事件发送的同时返回,LeaseDuration 是为事件请求的租借提出的持续时间.

Notify()方法返回 EventRegistration 对象,这是一个容器类,包含了事件源(对于从查找服务返回的事件,产生事件的源是 ServiceRegistrar)、事件注册的租约和其他一些数据.

3 结束语

上面详细介绍了 Jini 查找服务的原理和使用,这有利于对 Jini 核心机制的理解,也有助于开发和调试 Jini 应用程序.

参考文献:

- [1] [美]W.Keith Edwards 著,王召福等译.Jini 核心技术[M].北京:机械工业出版社,2000.17~70.
- [2] 黄志伟.Jini 技术[OL].www.einfoally.com/webcn/Jini2.htm.
- [3] 三千胡杨,王昊.Jini 技术基础结构[OL].www.hooyang.com/hy_tc/lantools/t_java/java_20010602_0003.htm,2001/06/02.
- [4] 尹湛.新一代分布式计算平台——Jini[OL].www.china-pub.com/computers/emook/0623/info.htm,2001/01/02.
- [5] Jini 技术综述[OL].http://tech.enet.com.cn/zhuanti/jdeveloper/doc/jini/1.html.