

基于移动 Agent 的分布式网络并行计算方法 A Method for Parallel Computing of Distributed Network Based on Mobile Agent

石润华 邹莹 钟诚*
Shi Runhua Zou Ying Zhong Cheng

(广西大学计算机与信息工程学院 南宁 530004)
(Coll. of Comp. & Info. Eng., Guangxi Univ., Nanning, 530004)

摘要 介绍移动 Agent 和基于移动 Agent 并行计算的基本思想和一个实际的移动 Agent 支持系统——Aglet 系统的基本原理。在此基础上, 以一个并行求和程序为例, 介绍基于移动 Agent 的并行计算的具体实现方法, 为移动 Agent 的实际应用提供参考。

关键词 并行计算 分布式网络 移动 Agent

中图分类号 TP301.6

Abstract The mobile Agent and its parallel computing are introduced. Based on the theory of a mobile Agent supporting system, a method for parallel computing of distributed Network, exemplified by a parallel sum process, is given. This method can be some reference values in the application of mobile Agent.

Key words parallel computing, distributed network, mobile Agent

1 移动 Agent 和基于移动 Agent 的并行计算

代理 (Agent) 一词最早出现在人工智能研究领域。从用户的角度来说, 它是一种程序, 代表和实现用户的意志和行为。它的功能是通过用户的允许来实现的。移动代理 (Mobile Agent) 由 General Magic 公司于 1990 年代初在商业系统 Telescript 中提出, 用来支持编程语言级的可移动性。可以将移动 Agent 定义为: 一个运行于开放、动态网络环境中封装良好的计算实体, 它代表用户自主地在网络上移动, 并完成指定任务^[1]。移动 Agent 能够在异构网络中, 在自己的控制下从一台主机移动到另外一台主机上继续运行。它可以在任意点暂停执行, 将自身传送到一台新的主机上, 并且在新主机上从暂停点继续执行。在每一台新的主机上移动 Agent 都可以直接同服务资源进行交互, 完成任务后再将结果同自身传回源主机。移动 Agent 实际上可理解为一段可移动的程序, 它具有跨平台持续运行、自我控制移动能力, 模拟

2003-01-13 收稿。

* 中国科学技术大学在读博士。

人类的行为和关系，并能提供只能服务一定人类的程序。与 Client/Server 技术以及 Applets 和 Servlets 移动代码相比，它具有自主性、移动性、协作性和安全性等特点。

Internet 上的主机分布在不同地点，硬件、软件环境各不相同，它们的计算能力也相差甚远。但它们都有着如下特征：平台的异构性和平等性，资源链接的开放性和动态性，使用方式的个性化和灵活性，网络连接环境的不可预测性。因此，要在网络环境下实现分布式并行计算就需要一种动态、自治、适应异构网络的计算模式。传统的 Client/Server 计算模式静态结构缺乏网络适应性，数据移动需增加网络负载，不具有网络协同的适应性。移动 Agent 能较好地解决这些问题，将会成为新一代的网络计算模式。

基于移动 Agent 的分布式并行计算除了具有传统的 Client/Server 计算模式的网络上每个主机既可作为客户端又可作为服务器端的优点外，更有着自身的优点：通过移动 Agent 到资源所在地，Agent 能够直接和资源进行交互而不需要通过网络传递中间数据，有效地节约带宽和降低响应时间，能够在低带宽、高延迟、不可靠的网络中应用；移动 Agent 可以在异地自主运行，可以根据当前主机计算能力和网络负载情况，自主判断是停留还是移动到新的主机上继续执行，对于网络环境具有很好的适应性，而且相互间有很好的协作性。

在基于移动 Agent 实现的分布式网络并行计算环境中，网络中的每一台主机均可作为并行计算环境中的成员，主机与主机之间地位平等，每台主机可生成一个或多个 Agent，并派遣 Agent 进行通信或计算；可以将大型或超大型的计算任务划分成多个可并行化的子任务交付各个移动 Agent，Agent 移动到各个主机上进行并行计算，任务完成后返回，或把计算结果用消息传递的方式送回。在此种计算环境中，通常需要一台主机作为划分任务、分配任务、监控任务执行以及回收任务的调度者。其计算模式见图 1。

2 移动 Aglets

Aglets 定义为基于 Java 的自治软件代理。它由 IBM 日本公司提出，是一种完全采用 Java 开发的移动代理计算技术，并提供实用的 Aglets Workbench 平台让用户开发或执行移动代理系统。Aglets 由“Agent”与“Applet”两个字合成，简单地说就是具有 Agent 行为的 Java applet 对象。它继承了 Applet 的代码移动特性，但与 Applet 不同的是它在代码移动的过程中携带状态一起移动，在异地可自主安全运行。

Aglets 系统提供一个环境界面 (AgletContext) 来实现 Aglet 和主机之间的交互。Aglet 使用环境界面来获取环境 (its aglets host) 信息，给环境或是在同一环境里的活动 Aglets 传送消息。它提供方法操纵和管理在一个环境里正在运行的 Aglets。而 Aglet 与 Aglet 之间的沟通可用消息传递的方式来传递消息对象 (message object)。此外，基于安全上的考虑，Aglets 并非让外界直接存取其资讯，而是透过一个代理者 (Aglet Proxy) 提供一致的界面与外界沟通，如图 2 所示。这样做的好处是：Aglet 所在的位置会透明化，也就是若想要与远端的 Aglet 沟通时，只要在本地主机环境界面上产生对应远端 Aglet 的代理者，并与此代理者沟通即可，不必直接处理网路连接与通讯的问题。环境界面和代理者共同管理 Aglet 的基本行为：产生

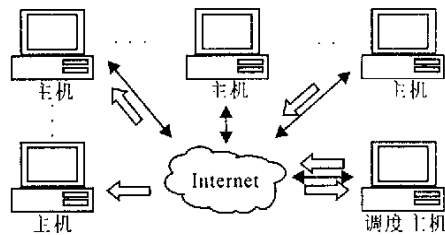


图 1 基于移动 Agent 的分布式网络并行计算模型
◁表示 Agent 移动方向。

Aglets, 召回远端的 Aglets, 复制 Aglets, 分派 Aglets 到远端工作站, 暂停、唤醒 Aglets, 以及移除 Aglets 等。

Aglets 系统提供一个服务器应用程序 Tahiti, 一旦它成功运行, 主机将作为代理服务器使用。通过赋给 Tahiti 不同端口号, 你可以在单个微机上运行多个服务器进程。Tahiti 提供一个用户界面来监视、新建、复制、派遣、消除 Aglets, 设置代理服务器的存取、进出权限等。

3 基于移动 Agent 的并行计算的实现方法

运行环境: 1 台 IBM Pentium 4, 操作系统为 Win2000。2 台 DELL Pentium 3, 1 台系统为 Win2000, 1 台为 Linux。使用 Aglets-2.0.1 版本。并配装 J2sdk1.4.0。

3.1 安装 Aglets-2.0.1

以下均以 Win2000 下安装与设置为例, Linux 的安装与设置过程相近。

(1)解压 Aglets-2.0.1.zip 包(解压后存放在 f:\Aglets-2.0.1)。

(2)修改 f:\Aglets-2.0.1\cnf\Aglets

2.0.1.props 配置文件。主要作如下修改:

1)设置 Aglets-2.0.1 的 home 路径

Aglets.home=F:\Aglets-2.0.1

2)设置主机名: maf.finder.host=srh

(srh 是用户电脑主机名)。

(3)修改环境变量; 打开“我的电脑”→

“控制面板”→“系统”→“高级”→“环境变量”

1)修改变量 CLASSPATH, 在其值的最后加上 “; f:\Aglets-2.0.1\lib\Aglets-2.0.1.jar”。

2)重新编辑 path 变量, 在其值的最后加上 “; f:\Aglets-2.0.1\bin”。

(4)运行 “f:\Aglets-2.0.1\bin\ant.bat”, 然后屏幕提示把文件 “f:\Aglets-2.0.1\bin\java.policy 和 .keystore” 复制到 “C:\Documents and Settings\Administrator\” 下。

(5)运行 “f:\Aglets-2.0.1\bin\Agletsd -f f:\Aglets-2.0.1\cnf\Aglets-2.0.1.props”。(启动 Tahiti: Aglets 服务器窗口管理程序)

(6)以后每次要启动 Aglets 服务器窗口管理程序-Tahiti, 只需直接在命令行下输入: agletsd。(user name: aglet_key, password: aglets)

3.2 并行求和程序的实现

要求计算从 1 到 1000 间所有整数和。我们把这一求累加和分成 2 个子任务。一个子任务求从 1 到 500 之间所有整数和, 另一个子任务求从 501 到 1000 之间所有整数和。IBM 主机作为调度、管理 Aglets 主服务器, 2 台 DELL 主机作为计算节点。首先在主 Aglets 服务器上生成一个主 Aglet-sum, 它是 Aglet 类的子类。由 Aglet-sum 再生成 2 个新的子 Aglet-sum1 和 sum2, 并发地调度它们到远端计算节点, 其中每个 Aglet 携带一个求和子任务。一旦完成求和任务, 它们可以携带计算结果返回到 Aglets 主服务器, 也可以把计算和作为消息参数传递给主 Aglet-sum 接收、处理消息参数, 并把它们相加得到最终结果, 实现并行求和。

3.2.1 方法介绍

下面介绍程序用到的主要方法: 1)public void onCreate(java.lang.Object init): 与 Applet

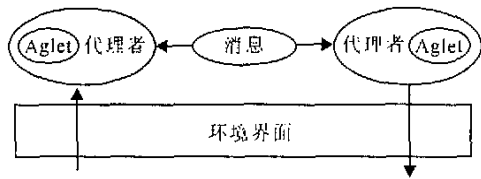


图 2 Aglet 基本功能界面

的 `init()` 方法一样, `onCreation(Object init)` 方法对 Aglet 初始化。2) `public final AgletContext getAgletContext()`: 获取正在执行的 Aglet 所处的环境—AgletContext 对象, 环境负责 Aglet 与主机通信。3) `public AgletProxy getAgletProxy(AgletID id)`: 在当前环境中获取由 id 标识的 Aglet 所属的代理者, 代理者负责 Aglet 与 Aglet 之间的通信。4) `public AgletProxy createAglet(java.net.URL codeBase, java.lang.String code, java.lang.Object init)`: 创建一个 Aglet 类的实例, 新建的 Aglet 代码既可在本地也可在远端主机上, 由第 1 个参数指定, 其代码 (.class 程序) 名为第 2 个参数给定。第 3 个参数为传递初始对象, 可以是 AgletID, 代理者等类型对象。5) `public AgletProxy dispatch(java.net.URL address)`: 由代理者对象调用, 并把属于代理者对象的 Aglet 派送到远端主机。9) `public java.lang.Object sendMessage(Message msg)`: 代理者对象调用, 表示向此代理者对象传送消息对象 msg。10) `public boolean handleMessage(Message message)`: 由 Aglet 调用, 对发送来的消息对象进行接收、响应和处理。11) `public void run()`: Aglet 线程的进入点。相当于 Java 程序的 `main()` 方法。

3.2.2 类实现

(1) Aglet-sum 类的实现。Aglet-sum 继承 Aglet 类, 重载了 3 个 Aglet 类方法: `onCreation(Object o)`, `handleMessage(Message msg)` 和 `run()`。在 `onCreation(Object o)` 中我们创建两个子类 Aglet-sum1 和 Aglet-sum2, 并调度子 Aglet 到计算节点并发求和。`run()` 方法等待两个子 Aglet 发送消息, 直到 `handleMessage(msg)` 处理完 2 个子 Aglet 发回的消息后才被激活, 最后把子 Aglet 的计算和相加, 得最终结果并在终端显示。

(2) 编写 Aglet-sum1 和 Aglet-sum2 类的实现。Aglet-sum1 和 Aglet-sum2 继承 Aglet 类, 重载了 2 个 Aglet 类方法: `onCreation((AgletID) id)` 和 `run()`。在 `onCreation((AgletID) o)` 方法中, 通过主 Aglet 传递来的参数 (AgletID) o (主 Aglet 的 ID 号, 系统随机产生一个惟一 ID 号来标识每个 Aglet), 来得到主 Aglet 的代理者。在 `run()` 方法中, 计算求和, 并把结果作为消息传递参数发送给主 Aglet 的代理者。

3.2.3 运行

编译 3 个 Java 程序, 在命令行下输入: `javac aglet-sum.java` 对 Aglet-sum.java 进行编译生成 Aglet-sum.class。同理生成 Aglet-sum1.class 和 Aglet-sum2.class。使用 Aglets 系统提供的 Tahiti 服务器应用程序运行 Aglet-sum.class, 立即生成 2 个子 Aglet-sum1 和 Aglet-sum2, 并自动把它们派遣到相应远程主机执行, 主 Aglet 等待 2 个子 Aglet 发送计算好的结果。在我们的应用实例中, 2 个子 Aglet 执行完求和计算后, 把结果 125250 和 375250 作为消息传递参数发回主 Aglet, 主 Aglet 最后把两个求和结果相加 $125250+375250$, 并把最终结果 500500 (即求和结果) 输出到屏幕显示。

4 结束语

移动 Agent 技术是人工智能与分布式计算技术相结合的产物, 有着很强的网络适应能力。通过移动 Agent 技术, 我们可以把网络上分散的、闲置的、异构的资源很好地利用起来, 在 Internet 上实现分布式网络并行计算。Aglets 系统完全由 Java 语言支持, 实现简单, 开发容易。

(下转第 140 页)

- 2 干道尊,汤嵘嵘,谭玉钧.中华鳖生化组成的分析 I:一般营养成分的含量及肌肉脂肪酸的组成.水生生物学报,1997,21(4):299~305.
- 3 Montano M, Navarro J C. Fatty acids of wild and cultured penaeus vannamei larvae from Ecuador. Aquaculture. 1996. 142: 259~268.
- 4 Ota T, Takagi T. A comparative study on the lipid class composition and fatty acid composition of sweet smelt *Plecoglossus altivelis*, from marine and freshwater habitat. Bull Fac Fish Hokkaido Univ. 1977, 28(1): 47~56.
- 5 Castell J D. 1979-Review of lipid requirements of finfish. Finfish nutrition and fish feed technology. Schriften BFF Hamburg/Germany, 14/15(2): 59~84.
- 6 李荷芳,郝斌,王辉亮等.野生对虾与养殖对虾脂肪酸组成和含量的比较.海洋科学,1993,(1):41~46.
- 7 李爱杰.中国对虾营养研究进展.上海水产大学学报,1988,7:16~23.
- 8 Bell M V, Henderson R J, Sargent J R. The role of polyunsaturated fatty acids in fish. Comp Biochem Physiol. 1986, 83 B(4): 711.

(责任编辑:邓大玉 曾蔚茹)

(上接第111页)

本文为我们在 Internet 上开展分布式网络并行计算提供范例,开拓了并行计算的应用领域。

参考文献

- 1 陶先平,吕建.基于 Internet 的软件 Agent 技术.南京:南京大学出版社,1998.
- 2 Bill Venners. Solve real problems with aglets, a type of mobile agent. [Http://www.javaworld.com/javaworld/jw-05-1997/jw-05-hood_p.html](http://www.javaworld.com/javaworld/jw-05-1997/jw-05-hood_p.html), 2002-12-06.
- 3 Bill Venners. The architecture of aglets. [Http://www.javaworld.com/javaworld/jw-04-1997/jw-04-hood_p.html](http://www.javaworld.com/javaworld/jw-04-1997/jw-04-hood_p.html), 2002-12-06.
- 4 胡朝晖,陈奇,瑞钊.移动 Agent 系统综述.计算机应用研究,2000,10:1~4,27.
- 5 王静翊. Agent (Mobile Agent). [Http://jspzone.myrice.com/article/java/aglet/aglet_brief.htm](http://jspzone.myrice.com/article/java/aglet/aglet_brief.htm), 2002-12-10.
- 6 钟诚,汪学明,陈旭. JAVA 语言及其应用.重庆:重庆大学出版社,2001.

(责任编辑:黎贞崇 邓大玉)