

基于 Android 手机的远程访问和控制系统

耿东久¹, 索岳², 陈渝², 文军¹, 吕勇强²

(1. 电子科技大学 计算机科学与工程学院, 成都 610054; 2. 清华大学 计算机科学与技术系, 北京 100084)

(gengdj.1984@gmail.com; wenjun@uestc.edu.cn)

摘要: 普适计算是一种新的计算模式, 其本质特点是物理集成和自发互操作。而智能空间正是普适计算的这两个本质特点的一种具体而集中的体现, 智能手机在其中可以获得增强个性化的服务。提出了一个基于 Android 手机的远程访问和控制系统, 该系统基于开放服务网关规范(OSGI)的智能网关技术。首先介绍了整个系统结构框架, 然后详细分析了各个模块的具体实现, 最后在真实设备上进行了测试。实现了 Android 手机利用周围的服务, 对其他设备的普适访问和远程控制。

关键词: 智能空间; 普适访问; 远程控制; Android 手机

中图分类号: TP311.5; TP316.5 **文献标志码:** A

Design and implementation of Android phone based access and control in smart space

GENG Dong-jiu¹, SUO Yue², CHEN Yu², WEN Jun¹, LÜ Yong-qiang²

(1. College of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu Sichuan 610054, China;

2. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: As a new computing mode, the most essential features of ubiquitous computing, embodied in intelligent space, are physical integration and spontaneous interoperation. Due to them, Intelligent phone can get personalized service. A remote access and control system was proposed in this paper, which was based on the Open Service Gateway Initiative (OSGI) intelligent gateway technology. Firstly, the system architecture was introduced, then the realization of each module was analyzed, finally, testing on actual devices was carried out. Using surrounding services, Android mobile phone realizes the universal access and remote control on other devices.

Key words: smart space; universal access; remote control; Android mobile phone

0 引言

随着普适计算的蓬勃发展, 出现了一种全新的计算模式。该模式脱离了传统的键盘、鼠标, 以自然和隐含的方式与信息系统进行交流, 在更多不在桌面环境下发生的日常活动中, 成为了一种主流模式, 人们可以随时随地、透明地获取数字化的服务, 体现了信息空间与物理空间的融合。智能空间是上述两个特征在房间、建筑尺度上的集中体现, 是一个在其中嵌入了计算、信息设备和多模态传感器的工作空间, 它的一个重要特性就是游牧服务(Cyber Foraging, CF)^[1-2]。用户带入空间的移动无线手持设备(例如手机)可以充分利用其周围基础设施中资源相对较丰富的设备, 进行信息处理。同时其上运行的模块也可以与空间中的其他模块进行交互和协作, 以获得与当地空间相关的各种服务。Android 是一种全新的开源手机操作系统, 具有强大的应用层 API 和丰富的传感器功能, 其开放的平台有利于开发者开发出各类应用软件, 并且无缝地结合 Google 优秀的服务^[3]。

本文介绍了一种基于 Android 的远程控制和访问系统, 在系统中 Android 手机利用周围无线网络资源与其他设备进行自发交互, 例如进行远程无线鼠标、无线键盘、文件共享、文

件传输操作, 远程遥控 UPnP 设备等, 为智能空间中其他具有传感和计算能力的设备识别丰富的上下文提供了可能, 又为实现计算对人的注意力的透明提供了重要的途径。

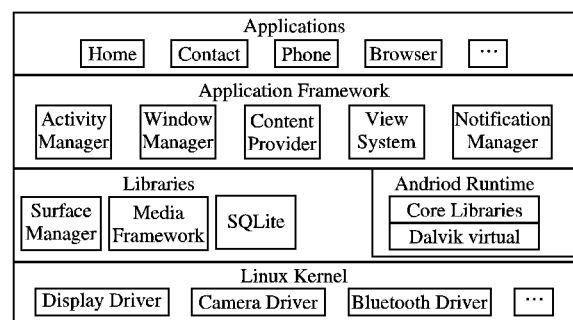


图1 Android 框架

1 Android 的系统架构

Android 操作系统自顶向下分成 4 个层次, 即应用层、应用框架层、组件库层和虚拟机和 Linux 内核层。其构架如图 1 所示^[4]。

1) 应用层。Android 操作系统同一系列核心应用程序包一起发布, 其核心应用程序包括 E-mail 客户端、SMS(短信)

收稿日期: 2010-08-12; 修回日期: 2010-09-15。 基金项目: 国家 863 计划项目(2009AA011906)。

作者简介: 耿东久(1984-), 男, 河北衡水人, 硕士研究生, 主要研究方向: 计算机网络、操作系统; 索岳(1983-), 男, 北京人, 博士, 主要研究方向: 普适计算、智能空间; 陈渝(1972-), 男, 广东湛江人, 副教授, 博士, CCF 会员, 主要研究方向: 操作系统、分布式系统; 文军(1966-), 男, 四川成都人, 副教授, 主要研究方向: 数据库; 吕勇强(1980-), 男, 北京人, 助理研究员, 主要研究方向: 集成电路计算机辅助设计、数字医疗。

序、日历、Google 地图、网页浏览器、联系簿等,目前所有的应用都是使用 Java 语言编写的。

2)应用框架层。开发者通过使用核心应用程序来调用 Android 框架提供的 API,这个应用程序结构被设计成方便复用的组件。任何的应用程序都可以公布它的功能,其他的应用程序可以使用这些功能(涉及到系统安全问题的功能将会被框架禁止)。该应用程序重用机制使用户可以方便地替换程序组件。隐藏在每个应用后面的是一系列的服务和系统,其中包括:

①丰富和可扩展的视图。用于构建应用,包括列表、网格、文本框、按钮,甚至内嵌的 Web 浏览器。

②内容提供者。为应用提供数据或者实现应用间的数据共享。

③资源管理器。提供非代码的资源支持和访问,包括本地字符串、图形和布局文件。

④通知管理器。使应用程序可以在状态栏上显示告警信息。

⑤活动管理器。管理应用的生命周期,以及提供常用的导航回退功能。

3)在组件库层。Android 包含了一套 C/C++ 函数库,主要包括 libe、Media Framework、WebKit、SGL、OpenGL ES、FreeType、SQLite 等,它们被应用于 Android 系统的各种组件中,这些功能通过 Android 应用框架展现给开发人员。

4)运行时环境。Android 包括了一个核心库,该核心库提供了 Java 编程语言核心库的大多数功能。每一个 Android 应用程序都在它自己的进程中运行,拥有一个独立的 Dalvik 虚拟机实例。Dalvik 被设计成可以同时高效地运行多个虚拟系统。Dalvik 虚拟机执行后缀为 dex 的可执行文件,该格式文件针对小内存使用做了优化。同时虚拟机是基于寄存器的,所有的类都经由 Java 编译器编译,然后通过 SD 中的“dx”工具转化成“.dex”格式。Dalvik 虚拟机依赖于 Linux 内核的一些功能,比如线程机制和底层内存管理机制^[4]。

5)Linux 内核层。Android 底层是一个基于 Linux 2.6.23 内核开发的独立操作系统。主要是添加了一个名为 Goldfish 的虚拟 CPU 以及 Android 运行所需的特定驱动代码。该层用来提供系统的底层服务,包括安全机制、内存管理、进程管理、网络堆栈及一系列的驱动模块。作为一个虚拟的中间层,该层位于硬件与其他的软件层之间。

2 系统总体设计

系统的总体设计思路是在智能空间中部署一台 PC,负责提供服务并与用户的 Android 手机通信。手机作为管理其他设备的管理器,提供管理界面,例如远程控制设备、文件操作。这实际上是一个 C/S 结构的系统,PC 作为服务器,Android 手机作为客户端。系统的总体结构如图 2 所示。

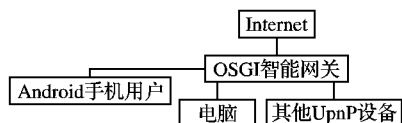


图2 系统总体结构

在该系统中,OSGI 智能网关连接了内部网络和外部网络,所有的内部设备与该网关相连,它主要由 OSGI Framework 以及各种 Bundle 组成,Bundle 之间互相通信,使设备能够了解彼此的状态,并进行相互的操作。用户携带 Android 智能

手机进入空间后,手机端连入网络,动态获取一个 IP 地址,搜索网络中的可用设备服务,发现可用服务后与之进行交互。Android 手机与 PC 之间通过 Socket 进行通信^[5]。

手机端的按键通过脚本文件定义成为各种击键命令或者热键组合命令(消息),PC 端解析该命令,调用相应的应用程序,进行远程控制,同时 PC 端会返回一些状态信息^[6],例如某个目录中所有文件的列表。当在焦点窗体响应命令(消息)显示按键字符时就是无线键盘,其设备管理框架如图 3 所示。

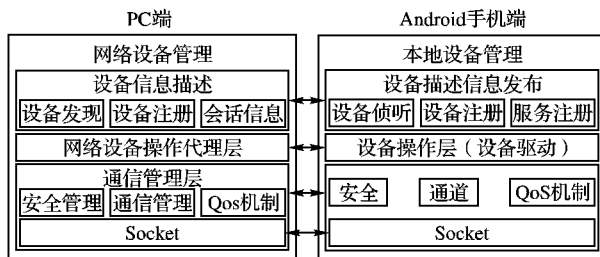


图3 设备管理框架

3 系统的具体实现

3.1 Android 手机终端模块的设计

当手机启动访问控制模块时,会新建一个线程^[8]。在此线程内,首先会创建 MulticastClient 对象,对 HTTP 端口、UDP 端口、SSDP 端口以及监听时间间隔进行设置;然后建立组播 Socket,设置 TTL 为 255,发送报文“connect”,等待服务器端的响应:服务器收到组播 UDP 包后,取得源端 IP 地址,并向手机端发送消息;手机接收到返回的消息之后,通过 handler 发送“找到 Server”的提示,并提取 Server 端的 IP 地址,完成查找过程^[7]。整个查找过程采用组播的方式,服务器端的响应时间大约为 3 s,可以返回多个 Server,便于与多个设备进行交互。接着建立 TCP 连接,在 TCP 的连接中指定了要连接的服务器端口号,当服务器的端口已被占用时,会返回端口已被占用的出错信息,这样并发用户数为 1,避免了相互之间的干扰。

Android 手机通过 CommandControl 类与 Server 进行消息(命令)的传递,它本身接受本地网络、远程网络信息两个参数,每次 TCP 连接都会新建一个 TcpClient 类对象,TcpClient 类对象封装了 TCP 连接,设有监听器 netDataListener,每次接收到报文后,都会调用 netDataListener 的 DataRecieved 方法,对报文进行响应处理。其中 CommandControl 类定义了一组请求操作,包括 FileCtrl 命令、MouseCtrlr 命令、KeyCtrl 命令等。这样手机端可向服务器端发送各种命令,主要命令如下:

```
BASE_LIST_REQ( CommandType. R3MOTE_SERVER),
LIST_REQ( CommandType. R3MOTE_SERVER),
LIST_REPLY( CommandType. R3MOTE_SERVER),
MEDIA_INFO_REQ( CommandType. R3MOTE_SERVER),
MEDIA_INFO( CommandType. R3MOTE_SERVER),
RUN( CommandType. R3MOTE_SERVER),
SERVER_ERROR( CommandType. R3MOTE_SERVER),
SUCCESS( CommandType. R3MOTE_SERVER),
AUTH_REQ( CommandType. R3MOTE_SERVER),
AUTH_REPLY( CommandType. R3MOTE_SERVER),
PLAY_DVD( CommandType. R3MOTE_SERVER), MOUSE_MOVE_REQ(
CommandType. R3MOTE_SERVER), MOUSE_CLICK_REQ(
CommandType. R3MOTE_SERVER), KEYBOARD_EVENT_REQ(
CommandType. R3MOTE_SERVER), MOUSE_WHEEL_REQ(
CommandType. R3MOTE_SERVER),
UPDATE_SERVER_REQUEST( CommandType. R3MOTE_SERVER)
```

```

SERVER), SHOW_ALL_FILES_REQ( CommandType. R3MOTE_
SERVER), SHOW_PLAYABLE_FILES_ONLY_REQ
( CommandType. R3MOTE_SERVER), LAUNCH_URL_REQ
( CommandType. R3MOTE_SERVER),
PLAY( CommandType. MEDIA_PLAYER),
PAUSE( CommandType. MEDIA_PLAYER),
...

```

为了减少资源占用,CommandControl 采用单例的形式,只与一个 Server 进行通信。当尝试与新的 Server 进行通信时,CommandControl 会关闭之前的连接,网络具有很大的吞吐量。多处使用子线程处理界面控件和耗时的操作。这样用户在与 Server 进行交互的同时,还可以进行本地的操作,系统被阻塞的时间很短。

为了交互的方便,使用 Android 的传感器来触发事件。例如当把手机上的文件传输到 PC 上时,把原本面朝上平置的手机往右旋转 45° 到 90° 之间时,触发文件传输的事件。首先注册一个方位感知传感器监听器。监听函数为 public void onSensorChanged(int sensor, float values[]), sensor 选用方向传感器,置横向旋转角 (value[2]) 在 45° 与 90° 之间。Android 手机本身不支持 samba 功能,为了传输方便,使用开源项目——JCIFS。JCIFS 是一个完全用 Java 开发的 SMB 客户端库,它紧密遵循通用 Internet 文件系统 (Common Internet File System, CIFS) 规范支持 Unicode、命令管道、批处理、线程化调用者的 IO 复用、加密认证和完整事务。在手机端引用该库,可以访问 Server 端的共享目录,并进行文件的上传和下载的操作。资源的 url 定位形式为:smb:// {user} : {password} @ {host} / {path}。

3.2 Server 端的模块设计

Server 端启动的时候,会加入到一个组播地址中,然后初始化 Socket,对规定的端口进行侦听。当请求连接的报文到来时,performAuthentication 会对用户的合法性进行验证,Server 端向手机端发送一个验证包,等待手机端的响应;当收到的响应报文数据与预期的数据相等时,验证通过,否则拒绝连接。连接成功后,会解析收到的报文。由 handleReceiveData 类处理收到的不同命令,并且发送 SendPacket 与手机端进行交互。

```

Public synchronized void handleReceiveData( AbstractPacket packet,
TcpConnection connection) {
    LOGGER.info("Received command: " + packet.toString());
    Command command = packet.getCommand();
    if (command == Command. BASE_LIST_REQ)
    {
        returnPacket = null;
        ...
    }
    else if (command.getCommandType() == CommandType. MEDIA_
_PLAYER)
    {if (command == Command. CLOSE)
        {
            ...
        }
        else if (command == Command. PLAY)
        {
            ctrlPoint.operate("powerOn");
        }
        else if (command == Command. STOP)

```

```

{
...
}
}

```

整个系统的运行流程如图 4 所示。

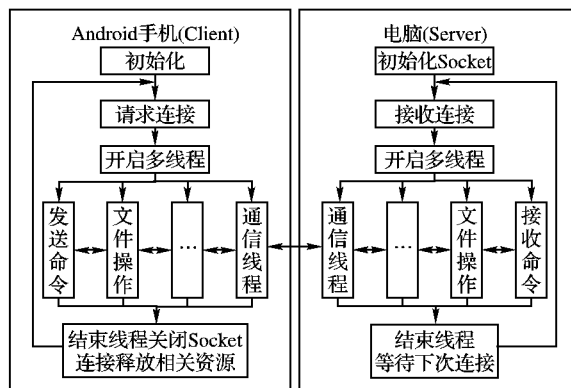


图4 系统运行流程

为了提高程序的响应速度,多处使用子线程,这样异步处理各种不同事件,避免了事件之间的盲等。当前没有处理任务时可将处理器时间让给其他任务^[8],占用大量处理器时间的任务可以定期将处理器时间让给其他任务,并可随时停止某个任务,设置各个任务的优先级,以此优化性能,提高了程序的运行效率,减少了服务器的响应时间,提高了 CPU 和网络 I/O 的利用率。

4 实现结果

系统采用了 C/S 的结构来进行实现,服务器与客户端通过 Socket 进行网络通信。在 PC 端使用 Java 的 OSGi 框架,手机的 Server 做成一个可热插拔的 Bundle,该手机 Bundle 只有被加载的时候才会启动,因此具有很高的性能,实现环境为 Eclipse。Android 手机端的启动、访问控制及文件操作过程如图 5 所示。

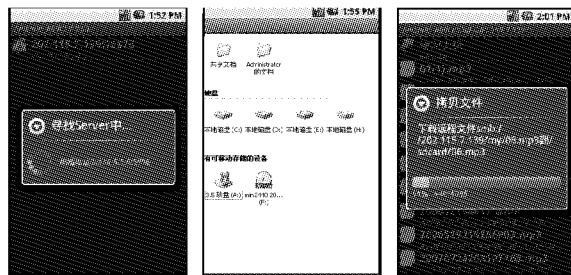


图5 实现结果

5 结语

经过在 HTC Magic G2 上反复测试,手机连接 Server 的时间小于 5 s,远端 Server 对手机控制命令的响应时间小于 3 s,基本上完成了手机在智能空间的普适访问和控制功能。系统在设计与实现中,已经加入了与其他设备进行交互的接口,下一步将准备实现多种设备之间的远程互操作^[9]。如将 UPnP 的协议移植到手机上,这样就可以自动联网,发现其他设备的服务,进而实现对整个智能空间中的 UPnP 设备进行控制,这也正是项目组正在深入研究和探讨的问题。

参考文献:

- [1] BALAN R, FLINN J, SATYANARAYANAN M, et al. The case for cyber foraging[C]// Proceedings of the 10th Workshop on ACM SIGOPS European Workshop. New York: ACM, 2002: 411-428.

(下转第 571 页)

本文方法在图规划算法的基础上加入了完全规约图、广义笛卡儿积的求解,因此执行时间要大于反向随机求得可行解的经典图规划算法,但随着规模的增大,DLG 具有较高的稳定性;对于大规模的 Web 服务库,能够在较短时间内完成自动组合,满足用户对组合服务质量的要求。

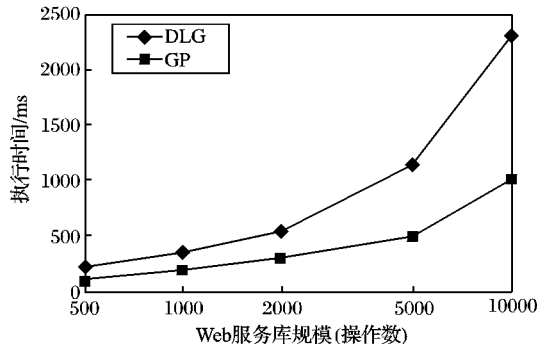


图3 不同 Web 服务库规模下服务请求处理时间

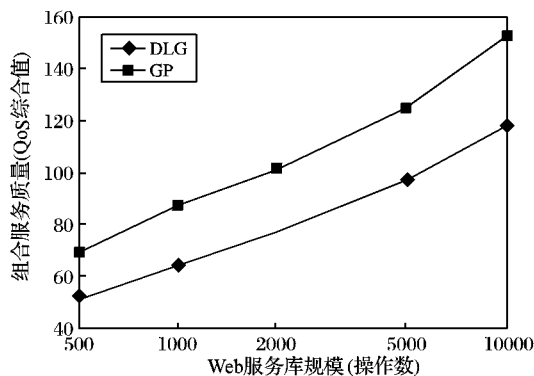


图4 不同 Web 服务库规模下组合 Web 服务质量

4.3 组合服务质量的仿真实验

由于单个服务请求的随机性,故在不同规模 Web 服务库下,测试了 200 个服务请求的平均服务质量,每个 Web 服务的质量由小于 20 的随机正整数生成。由定义 1, QoS 综合值越小表示质量越高;图 4 显示了随着规模的增大, QoS 综合值不断增大,这与实际是相符的,因为随着规模的增大,操作库中参数数目增大,随机生成的服务请求需要更多的服务操作,因此组合服务的 QoS 综合值不断增大;DLG 求得的组合服务质量明显优于图规划算法,从而能够提高组合的成功率。

以上两组实验结果表明,基于有向层次图的 Web 服务组合方法,能够在较短时间内解决大规模 Web 服务操作库的 Web 服务组合问题,得到近似质量最优的组合服务序列。

5 结语

为了解决 Web 服务自动组合问题,本文提出了基于有向

层次图的服务组合方法,从用户输入开始,通过 Web 服务操作库建立参数节点之间的层次关系,避免了循环搜索,提高了 Web 服务之间的并发性,能够得到具有最小阶数的最优 Web 服务组合序列,该序列近似全局最优的,且不存在冗余的 Web 服务。仿真实验结果表明,本文方法能够在较短时间内求得最优组合服务,有效地满足大规模 Web 服务操作库的用户请求,提高服务组合的成功率。

本文主要求解 Web 服务的组合序列,没有考虑 Web 服务的组合验证、Web 服务组合失败后的补偿以及用户对组合服务某些单一 QoS 属性的要求,这些都是下一步要解决的问题。

参考文献:

- [1] 倪晚成,刘连臣,吴澄. Web 服务组合方法综述[J]. 计算机工程, 2008, 34(4): 79-81.
- [2] VOSSSEN T, BALL M, LOTEM A, et al. On the use of integer programming models in AI planning [C] // Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. New York: ACM, 1999: 304-309.
- [3] 李喜彤,范玉顺. Web 服务流程相容性和相似性分析[J]. 计算机学报, 2009, 32(12): 2429-2437.
- [4] 廖军,谭浩,刘锦德. 基于 Pi-演算的 Web 服务组合的描述和验证[J]. 计算机学报, 2005, 28(4): 635-643.
- [5] 雷晖晖,段振华. 一种基于扩展有限自动机验证组合 Web 服务的方法[J]. 软件学报, 2007, 18(12): 2980-2990.
- [6] 李鹏,战德臣,刘国忠,等. 一种面向用户的 Web 服务组装方法[J]. 计算机应用, 2009, 29(11): 3120-3123.
- [7] 龚小勇,朱庆生,武春岭. 基于位置矩阵 QoS 感知的 Web 服务组合研究[J]. 计算机应用, 2008, 28(8): 2170-2172.
- [8] HASHEMIAN SV, MAVADDAT F. A graph-based framework for composition of stateless Web services[C] // Proceedings of the European Conference on Web Services. New York: ACM, 2006: 75-86.
- [9] HASHEMIAN SV, MAVADDAT F. A graph-based approach to Web services composition[C] // Proceedings of the Symposium on Applications and the Internet. Washington, DC: IEEE Computer Society, 2005: 183-189.
- [10] 曹利培,刘静,缪淮扣. 基于图的 Web 服务组合优化的研究[J]. 计算机科学, 2007, 34(2): 95-99.
- [11] 刘家茂,顾宁,施伯乐. 基于 Mediator 的 Web Services 无回溯反向链动态合成[J]. 计算机研究与发展, 2005, 42(7): 1153-1158.
- [12] 刘峰,谭庆平,杨艳萍. 基于深度优先搜索的 Web 服务合成算法[J]. 计算机工程与科学, 2006, 28(12): 80-82.
- [13] 邓水光,吴健,李莹,等. 基于回溯树的 Web 服务自动组合[J]. 软件学报, 2007, 18(8): 1896-1910.
- [14] YANG YAN PING, TAN QIN PING, LIU FENG, et al. Web service composition algorithm based on fix-point theorem[C] // International Federation for Information Processing. New York: Springer-Verlag, 2005: 293-303.

(上接第 561 页)

- [2] XIE WEIKAI, SHI YUANCHUN, XU GUANYOU, et al. Smart platform: A software infrastructure for smart space[C] // IEEE International Conference on Multimodal Interfaces. Washington, DC: IEEE Computer Society, 2002: 429-434.
- [3] 潘永高,钟亦平,张世永. 基于网关的 J2ME Jabber 系统研究[J]. 计算机工程, 2005, 31(19): 108-110.
- [4] 龙亚券,黄璞,吴胜. 超大容量 NAND Flash 文件系统——YAFFS2 在 Linux 下的实现[J]. 北京电子科技学院学报, 2007, 15(2): 80-84.
- [5] YUAN M J. J2ME 移动应用程序开发[M]. 梁超,王延华,译. 北

京:清华大学出版社, 2004.

- [6] YUAN M J, SHARP K. Series 40 可扩展应用程序开发[M]. 周良忠,译. 北京:人民邮电出版社, 2006.
- [7] Google 开发人员访谈 Android 平台[EB/OL]. [2010-06-01]. <http://www.builder.com.cn/2007/1113/628807.shtml>.
- [8] 张海燕. Java 多线程技术在手机互联网中的应用[J]. 农业网络信息, 2008(3): 97-99.
- [9] WEISER M. The computer for the twenty-first century[J]. Scientific American, 1991(9): 94-100.