

一个面向分布式桌面计算环境的超瘦客户端

张跃冬, 朱定局, 宋振华, 孙毓忠

(中国科学院计算技术研究所国家智能计算机研究开发中心, 北京 100080)

摘要: 瘦客户端技术通过程序逻辑与用户界面的分离以及处理的集中来达到提高资源利用率、降低系统成本、提高系统安全性等目的。该文介绍了一个适合面向分布式桌面计算环境的超瘦客户端系统——SmartTerm, 分析了 SmartTerm 设计中面临的一些技术问题, 提出了解决方案, 并且以 RFB 协议为基础提供了一个 SmartTerm 的设计实现实例。SmartTerm 可以应用到网格计算和普适计算环境下的网络终端、移动设备、远程控制设备等诸多领域。

关键词: 瘦客户端; 远程显示; 分布式计算; 桌面环境

Ultra-thin-client for the Distributed Desktop Environment

ZHANG Yuedong, ZHU Dingju, SONG Zhenhua, SUN Yuzhong

(National Research Center for Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

【Abstract】 Thin-client computation advocates the separation of application GUI and the processing logic as well as the centralization of the processing, hence to enhance the resource utilization, decrease the system cost, and improve the system security. This paper introduces SmartTerm, an ultra-thin-client for the distributed desktop-like computing environment. It discusses the technological issues in designing the SmartTerm system, and proposes solutions. A design and implementation instance of the SmartTerm system is presented basing on the RFB protocol. The SmartTerm system can be applied in many systems under the grid computing and ubiquitous computing environment, such as grid terminals, mobile devices, remote control device, etc.

【Key words】 Thin-client; Remote display; Distributed computation; Desktop environment

1 概述

瘦客户端(Thin-client)是从提高系统资源利用率、降低信息系统总体成本、增强系统安全性等目的出发而提出的一种计算技术。其根本思想在于将应用程序的执行逻辑和界面显示部分分离, 将应用程序的逻辑处理部分集中到后端处理能力强的服务器上进行。通过多个用户终端共享同一个服务器的处理能力来达到提高系统资源利用率、降低信息系统成本、增强系统的安全控制能力的目的。

网格计算、普适计算等计算模式的兴起进一步推动了瘦客户端技术的发展。网格计算^[1]被认为是未来一段时间计算模式发展的趋势, 它主张在动态、多团体的虚拟组织中的协同资源共享和问题解决。普适计算^[2], 也称泛在计算, 它主张计算机本身从人类视线中的消失。上述计算模式将从应用前景和技术需求两个方面影响瘦客户端系统的发展: 一方面, 由于资源的分布化、用户终端的泛在化, 瘦客户端系统将获得更广泛的应用场景; 另一方面由于客户端的便携性、移动性等要求, 瘦客户端系统需要在技术上进一步改进。

针对未来分布式计算环境对终端系统的需求, 本文提出了一种可以为用户提供桌面式计算环境的超瘦客户端(Ultra-thin-client, SmartTerm)。所谓“超瘦”是指瘦客户端只是一个网络挂载的终端设备, 仅包括一个或者一组具备网络通信能力的人机交互部件, 它只负责用户的界面显示和输入转发。其它如应用处理、数据存储、用户环境集成等工作都在后端的能力服务器上进行, 如图 1 所示。超瘦客户端系统的主要优势可以总结为结构简单、低成本、低功耗、支持异构平台。

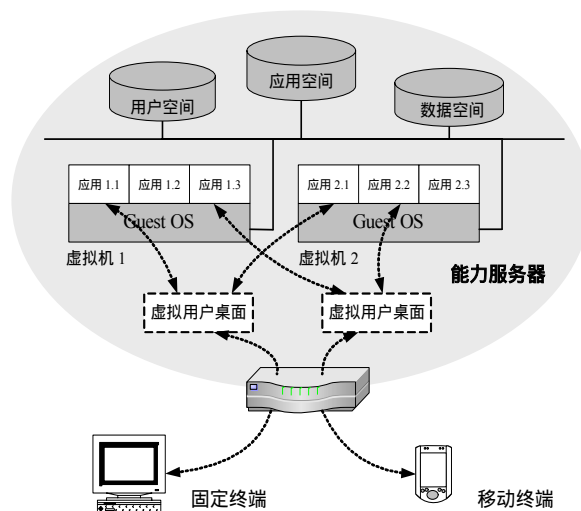


图 1 一个分布式桌面计算系统的实例

本文主要讨论了 SmartTerm 系统的设计和实现中的一些关键问题, 提出了解决方案, 并且以 RFB^[3] 协议为基础, 提供了一个 SmartTerm 系统的实现实例, 并对实现实例进行了分析和评价。

基金项目: 中国科学院百人计划基金资助项目

作者简介: 张跃冬(1976 -), 男, 博士生, 主研方向: 服务器体系结构和远程设备访问方面的研究; 朱定局, 博士生; 宋振华, 硕士生; 孙毓忠, 研究员、博导

收稿日期: 2006-04-30 **E-mail:** ydzhang@ncic.ac.cn

2 SmartTerm 的设计问题分析

2.1 显示数据的捕获及表示方法

当前的瘦客户端系统中,显示数据的捕获及表示主要分为两种方式。第1种方式以X11、ICA、RDP等协议及相关系统为代表,这类瘦客户端系统采用一种高层次的、和系统平台相关的命令来表示显示数据。这种方式的特点是编码效率高,但不支持异构平台,另外终端所需的软硬件支撑比较庞大。第2种方式以VNC^[4]系统为代表,这类系统采用一种低层次的基于底层设备命令和位图拷贝的方式来捕获和表示显示信息。相对于第1种方式而言,这种方式平台的相关性弱,但是数据通信量比较大。对于超瘦客户端来说,上面第2种数据捕获和表示方式更为合适,这是因为:(1)这种方式不需要依赖特定的操作系统平台语义进行解码;(2)基于底层设备命令的处理方式使得数据处理路径得到简化;(3)随着网络通信的进一步发展,网络带宽将不是瓶颈,事实上RFB协议目前已经可以在低带宽网络上顺畅运行。

2.2 数据传输

在数据传输上,SmartTerm 主要面临以下问题:(1)在数据传输上,服务器和客户端的主从关系如何?(2)瘦客户端与服务器的通信是否采用强同步的方式?(3)通信数据是否有状态?在以上问题上,SmartTerm 系统的最佳选择如下:SmartTerm 系统采用主从关系,显示输出设备采用服务器端主动的方式,而用户输入设备则采用客户端主动的方式,这是从输入/输出行为的本身特征作出的选择。SmartTerm 与服务器之间的通信采用弱同步、无状态的方式,这样才能更好地实现网格计算和普通计算环境下部件分布化、可移动,部件之间松散耦合,系统动态部署等目标。

2.3 客户端结构

SmartTerm 客户端的设计遵循两个原则:(1)SmartTerm 的定义,即只包括一个或者一组具备网络通信能力的人机交互设备;(2)与显示协议及服务器端平台兼容。针对个人用户的 SmartTerm 而言,一组人机交互设备主要包括输入设备(键盘、鼠标等)和显示设备。为了使这组设备的网络挂载能力得到提高,SmartTerm 还应该包含一个通信部件。在 SmartTerm 中,采用 Linux 帧缓冲区设备规范来实现对硬件设备的抽象,而这种抽象使得硬件数据接口和传输协议中的数据表示比较接近,只需要在二者之间作一些转化处理,简化了客户端软件的结构和运行开销。

3 基于 RFB 协议的 SmartTerm 设计

3.1 Linux Framebuffer 设备

帧缓冲区(framebuffer)是Linux把显存抽象后的一种设备。它允许上层应用程序在图形模式下直接对显示缓冲区进行读写操作。这种操作是抽象的、统一的,用户不必关心物理显存的具体细节。Linux帧缓冲区设备,目前已经广泛应用到了Linux平台的图形界面显示中。另外,通过和嵌入式Linux操作系统及相应开发系统配合,Linux帧缓冲区设备已经应用到了移动设备和手持设备中^[5]。

3.2 RFB 协议

RFB 协议全称为远程帧缓冲区(Remot Frame Buffer)协议,是用于 VNC 系统中的一个远程显示协议。由于 RFB 协议工作在帧缓冲区层次上,因此它对于几乎所有的窗口系统和应用都适合。RFB 协议显示部分基于一条简单的显示原语“在给定的 x, y 位置上显示一个矩形像素数据”,通过一系列矩形区域显示的更新来达到一次帧缓冲区更新。

RFB 协议中的帧缓冲区和 Linux 平台上的帧缓冲区设备虽然属于同类概念,但是二者并不等同。RFB 协议中的帧缓冲区是一种在软件层次上的虚拟,在服务器端其数据来源不是 Linux 帧缓冲设备;在客户端其数据也不能直接写入帧缓冲区设备。

3.3 SmartTerm 结构

基于 RFB 的 SmartTerm 设计的总体思想:以 RFB 协议和 Linux Framebuffer 设备为基础,建立可以直接读写 Linux Framebuffer 设备及标准输入设备(键盘、鼠标)的 RFB 协议处理引擎。该引擎的运行不依赖以往瘦客户端系统所必需依赖的特定操作系统语义或者软件库,从而实现 SmartTerm 的真正“超瘦”目标。

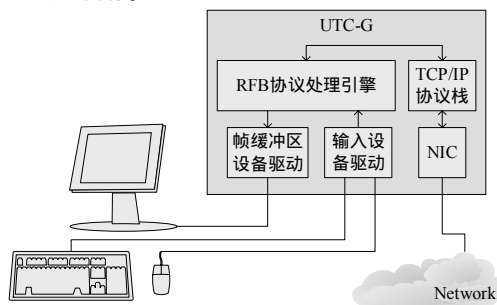


图 2 SmartTerm 的结构

SmartTerm 的结构如图 2 所示,主要包括 5 个部分:网络接口, TCP/IP 协议栈, RFB 协议处理引擎, Framebuffer 设备驱动和输入设备驱动。网络接口包括网络接口卡及其驱动程序,完成 SmartTerm 与服务器的物理连接功能。TCP/IP 协议栈部分完成对 TCP/IP 协议的支持。在网络层选用 TCP/IP 协议是因为 TCP/IP 协议是一个得到广泛应用的、能保证数据可靠有序传输的协议。事实上,其他可以保证数据可靠有序传输的网络层协议也可以应用到 SmartTerm 中。帧缓冲区设备驱动是遵循 Linux 帧缓冲区显示设备规范的驱动程序。输入设备驱动包括鼠标和键盘的驱动程序。

RFB 协议处理引擎是 SmartTerm 的核心部分,主要完成以下功能:(1)通过 ioctl 函数从帧缓冲区设备中读取帧缓冲区设置,根据配置信息完成 RFB 客户端配置;(2)通过一系列的 RFB 消息交换完成 SmartTerm 与服务器端 RFB 通道的初始化;(3)读取服务器发送来的 RFB 消息,并将其翻译成相应的帧缓冲区读写命令,绘制屏幕信息;(4)处理输入设备驱动程序的事件,将相应的输入事件翻译成 RFB 消息,发送给服务器。

3.4 服务器端支持

由于当前的 SmartTerm 实现采用 RFB 协议作为显示协议,因此包括各种版本的 VNC 服务器在内的任何支持 RFB 协议的服务器都可以作为 URC-G 的服务器软件。在图 1 所示的分布式桌面计算环境中,采用了一个经过改造的 Linux 平台 VNC 服务器和一个经过改造的 GNOME 桌面系统作为我们的分布式计算环境接入网关。该网关可以将分布在多个任务节点上的用户应用集成到统一的虚拟用户桌面中提供给客户端。

4 实验及讨论

目前,在 Linux 平台上,VNC 厂商提供了基于 X11 的客户端软件。在实验中,在相同的平台上对 SmartTerm 和 X11 VNC Viewer 从软件运行开销和系统性能两个方面进行了对比。实验平台的客户端采用 Redhat Linux 9.0(内核版本 2.40.8)

操作系统,硬件配置为 CPU 奔腾 III 850,内存 SDRAM PC133 384M;客户端通过 10M 以太网与服务器连接。

对于软件运行开销,从内存消耗和 CPU 占用两个方面进行了考察。图 3 对比了空载、网页浏览、和流媒体放映时二者的平均内存消耗和 CPU 占用情况。

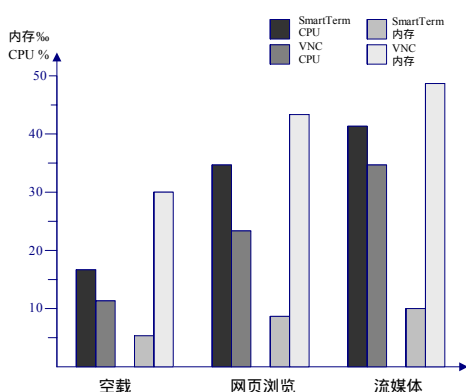


图 3 资源消耗对比

在性能方面,对比了一些典型应用的启动和操作显示延迟,具体如表 1 所示。

表 1 应用启动和操作延迟对比

延迟项	SmartTerm	VNC
GNOME 启动	8.5s	10s
Mozilla 启动	2s	2.5s
网页切换	0.5s	0.8s
OpenOffice	7s	8s
Office 文档切换	1s	1.5s

通过以上对比可以看出,SmartTerm 比以往的 VNC 客户端在内存消耗上有显著优化,显示延迟有一定降低。这主要是因为基于帧缓冲区设备的客户端系统减少了 RFB 消息的处理和拷贝过程,使得处理过程中的内存消耗和处理时间都得到了降低。但是,在以上的测试中 SmartTerm 的 CPU 消耗高于 VNCViewer。这主要是因为基于 X 的 VNCViewer 通过 Xfree86 系统充分利用了显示设备的硬件加速功能,将一部分图像处理操作放到了显示设备上,而 Linux 帧缓冲区设

备不具备这一功能,将所有的图像处理都放在了机器的 CPU 上。由于现在的多数嵌入式系统和便携设备的显示系统不具备或没有较强的硬件加速功能。因此这一现象不会在这类系统上出现,因此,对于嵌入式系统和便携设备来说,SmartTerm 的 CPU 消耗依然有一定的优势。

基于 RFB 设计实现的 SmartTerm 系统存在的一个比较大的缺陷是:RFB 协议的显示数据更新是基于客户端请求(Pull)模式的,不符合我们原本提出的系统设计目标。这将是我们在下一步工作中需要解决的一个问题。

5 总结

本文针对网格计算和普适计算对客户端的需求而提出了超瘦客户端的概念。所谓超瘦客户端是指客户端只是一个或者一组具备网络通信能力的人机交互设备。SmartTerm 设计和实现中的主要问题在于传输协议设计或选取、服务器端显示数据的捕获方式选择、客户端结构设计等。本文在 RFB 协议的基础上完成了一个 SmartTerm 系统的设计及实现实例。通过实验可以看到,SmartTerm 在瘦客户端结构简化、成本削减、低功耗支持等方面有相当的优势。另外,由于客户端软件结构的简化,SmartTerm 在性能上也比以往的类似瘦客户端系统有所提升。下一步,我们将通过协议和服务器端软件的改进来解决显示数据更新模式问题。目前 SmartTerm 已经应用到国家智能计算机研究开发中心的能力服务器实验系统中。

参考文献

- 1 Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations[J]. International Journal of High Performance Computing Applications, 2001, 15(3): 200-222.
- 2 Roy W, Trevor P, Gaetano B, et al. Disappearing Hardware[J]. IEEE Pervasive Computing, 2002, 1(1): 36-47.
- 3 Tristan R. The RFB Protocol[Z]. AT & T Labs Cambridge Whitepaper, 2005.
- 4 Richardson T, Stafford F Q, Wood K R, et al. Virtual Network Computing[J]. IEEE Internet Computing, 1998, 2(1): 33-38.
- 5 Trolltech. Qt/Embedded Whitepaper[Z]. 2004. <http://www.trolltech.com>.

(上接第 103 页)

参考文献

- 1 Gnutella[Z]. <http://gnutella.wego.com/>.
- 2 El-Ansary S, Alima L, Brand P, et al. Efficient Broadcast in Structured Peer-to-Peer Networks[C]//Proc. of International Workshop on Peer-to-Peer Systems. 2003.
- 3 Gupta I, Kermarrec A, Ganesh A. Efficient Epidemic-style Protocols for Reliable and Scalable Multicast[C]//Proc. of IEEE Intl. Symp. Reliable Distributed Systems. 2002.
- 4 Jelasity M, Preuss M, Paechter B. A Scalable and Robust Framework for Distributed Applications[C]//Proceedings of the IEEE Congress on Evolutionary Computation. 2002.
- 5 Montessor A. A Robust Protocol for Building Superpeer Overlay Topologies[C]//Proceedings of the 4th International Conference on Peer-to-Peer Computing, Zurich, Switzerland. 2004.
- 6 Ratnasamy S, Francis P, Handley M, et al. A Scalable Content Addressable Network[R]. Berkeley, CA, TR-00-010, 2000.
- 7 Rowstron A, Druschel P. Pastry: Scalable, Decentralized Object Location and Routing for Large-scale Peer-to-peer Systems[C]//Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms. 2001.
- 8 Stoica I, Morris R, Karger D, et al. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications[C]//Proc. of the ACM SIGCOMM'01 Conference, New York, 2001. ACM Press, 2001.