doi:10.3969/j.issn.1673-4807.2014.04.016

基于龙芯的 Windows 软件运行环境的设计与实现

苏 煜[1],张 涛[1],靳国杰[2],荣 辉[2],王金双[1],赵 敏[1]
(1.解放军理工大学 指挥信息系统学院,江苏 南京 210007)
(2.龙芯中科技术有限公司,北京 100098)

摘 要: 龙芯处理器在国产处理器市场占有重要地位.由于龙芯采用的是 MIPS 指令集且主要支持 Linux 操作系统,然而目前在该平台上的应用软件还相对匮乏,且很多传统 Windows 软件也无法在该平台上直接运行,这使得龙芯处理器的推广与应用受到了影响.针对这一问题,提出了一种在龙芯平台上基于 Wine 来提供 Windows 虚拟执行环境,并通过将 Wine 和 Bochs 模拟器相结合完成 X86 指令集到 MIPS 指令集翻译的解决方案.该方案能够实现 Windows 程序在零修改情况下直接运行在龙芯平台上,可支持 Delphi,C,C++ 等多种语言开发的程序,以及网络、数据库等多种功能.测试表明:该系统运行稳定,但还存在不可忽略的性能影响.

关键词: 龙芯; Wine; Bochs; 指令集翻译
中图分类号: TP391 **文献标志码:** A **文章编号:** 1673-4807(2014)04-0387-05

Design and implementation of Windows software environment based on Loongson

Su Yu[1], Zhang Tao[1], Jin Guojie[2], Rong Hui[2], Wang Jinshuang[1], Zhao Min[1]
(1. College of Command Information Systems, PLA University of Science and Technology, Nanjing Jiangsu 210007, China)
(2. Loongson Technology Corporation Limited, Beijing 100098)

Abstract: The Loongson processor occupies an important position in the domestic market. Since Loongson uses the MIPS instruction and supports the Linux operating system, the application software on the platform is relatively scarce, and many traditional Windows software cannot be run directly on the platform. It makes the popularization and application of Loongson processor affected. To solve this problem, a Loongson platform is proposed based on Wine to provide Windows virtual execution environment. The translation from X86 instruction to the MIPS instruction is achieved by combining the Wine and Bochs simulator. Windows program of zero-modified can directly run on Loongson platform in the proposed scheme. The scheme can support software programmed by Delphi, C, C++ and implement functions of network and database. Tests show that the system is stable, but it also has the performance impact which cannot be ignored.

Key words: Loongson; Wine; Bochs; ISA translation

核心软硬件的自主化对于国家信息安全至关重要.目前,国产处理器主要有龙芯、飞腾等;操作系统主要代表是中标麒麟;数据库主要有人大金仓、神通和达梦等.龙芯处理器采用 MIPS 指令集,运行类 Unix 操作系统,由于该平台上应用软件还相对匮乏,且 Windows 程序无法直接在该平台上运行,这对传统应用的移植以及该平台的推广与应用造成一定的影响.

传统的软件移植的主要类型通常是二进制移植(移植可执行文件)和源代码移植(移植源语言表达式)[1].源代码移植是通过对软件源代码进行分析,找出其相关于目标环境的代码,并进行修改,使其符合环境的运行要求,但有些时候源代码无法获得.二进制移植是指可执行文件通过向目标环境移植,达到在目标环境正常运行的移植方法.虚拟化技术是常用的二进制移植方法,但目前龙芯处理器的性能

收稿日期: 2014-08-20
作者简介: 苏煜(1991—),男,硕士研究生,研究方向为系统软件安全.E-mail:barton_sue@163.com

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

还难以克服当前虚拟化技术带来的性能影响.

因此,文中提出了在龙芯处理器上交叉编译 Wine 软件,使其提供 Windows 应用软件运行环境的轻量级解决方案,并通过 Wine 与 Bochs 模拟器相结合实现了 X86 指令集向龙芯 MIPS 指令集的解释执行,从而实现了 Windows 程序向龙芯平台的零修改迁移.

1 相关技术的研究与展望

指令集仿真是解决不同指令集之间软件二进制移植的主要方法. 当前,对指令集的仿真主要分为两种方式:一种是指令解释执行,一种是二进制翻译[2]. 解释执行通过对源指令进行取指、译码和仿真,实现源指令集体系结构 ISA(instruction set architecture)向目标 ISA 的映射. Bochs 便是一个开源的指令集解释器,它可以通过对不同设备进行模拟,实现应用对虚拟化的不同要求,而不需要对整个系统进行仿真. 二进制翻译是把源指令块翻译成执行同样功能的目标指令,因为是指令块之间的对应,相对于解释执行,效率会更高. QEMU(quick EMUlator)是目前较为先进的支持多元多目标平台的动态二进制翻译系统[3]. 为了实现 Windows 应用程序在跨操作系统,跨指令集的平台下运行,不仅需要对用户指令进行翻译,同时还需要仿真 Windows 的系统调用. 单纯的使用指令集仿真无形当中加重了虚拟化的负担,影响了使用效率. 因此,将 Wine 编译到 MIPS 环境,由 Wine 来提供 Windows 应用程序所需的部分 API,同时通过 Bochs 实现在 MIPS 环境下用户指令和部分 Windows 原生动态库的解释执行,构建 Bochs + Wine 方式提高运行效率. 未来可考虑实现 QEMU + Wine 的方式来进一步优化系统,提高效率.

采用分层思想的计算机体系结构存在 ISA 和 ABI(application binary interface)[4] 两个重要接口. ISA 由用户 ISA 和系统 ISA 组成,ABI 主要由系统调用接口和用户 ISA 组成[5]. X86 与 MIPS 分别有不同的 ISA 且相互不兼容. 同时为了提高软件在不同 ISA 的运行效率,面临着系统调用跨平台执行的性能优化问题.

2 系统的设计与实现

2.1 系统架构

为了迎合自主化的需要,该虚拟结构运行在搭载 Linux 操作系统的龙芯 3A 处理器上. 龙芯 3A 是国家自主研制的基于 MIPS 指令集的处理器. 在该平台上运行 X86 指令的 Windows 应用程序既跨体系结构又跨操作系统平台. 本系统采用开源软件 Wine 和 Bochs 来完成此任务,整体设计方案如图 1.

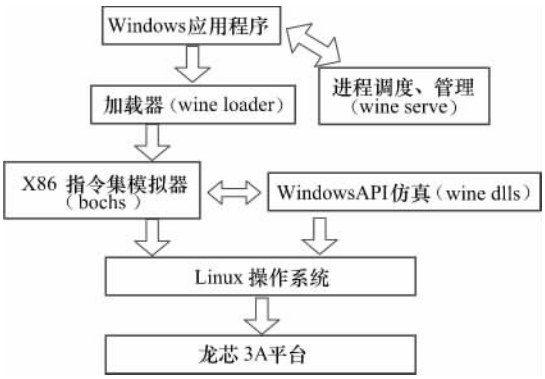


图 1 跨平台中间件的整体设计方案

Fig. 1 Overall design of cross platform middleware

该系统通过虚拟化技术构造一个进程虚拟机,该虚拟机通过 Wine 提供并实现应用程序运行所需的动态库. 结合 X86 模拟器对应用程序调用的 Windows 原生动态库以及 Wine 的初始化指令进行翻译执行,通过 X86 模拟器实现 Wine 在不同 ISA 平台下对 Windows 系统库的兼容.

虚拟化是构建一个将虚拟的客户系统映射到真实的主机系统上的同态. 通过虚拟化技术,可以实现依赖于客户系统的软件在主机系统上良好运行. 虚拟化过程包含两个部分:①把虚拟资源或状态映射成底层机器中的真实资源;②使用真实机器上的指令和/或系统调用来执行虚拟机的指令和/或系统调用规定的活动[6]. 虚拟机便是通过虚拟化技术,构建一个介于主机和客户系统的中间件实现仿真平台的设计架构. 通过虚拟机 ABI 或 ISA 的仿真,实现可执行文件的跨平台移植.

2.2 ABI 仿真虚拟机 Wine

Wine 通过 Window API 仿真实现 Windows 软件在类 UNIX 系统中运行[7]. 与一般的模拟器不同,它并不提供虚拟 cpu 以及 I/O 设备的仿真,而是通过 Winelib 把 Windows API(应用程序接口)用标准的 Unix 和 X 接口重新实现(图 2). Windows 应用程序

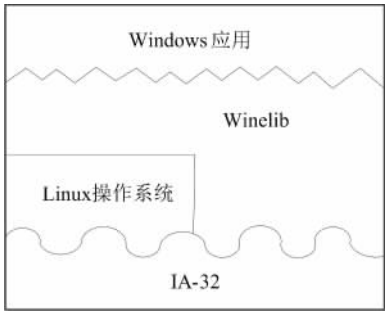


图 2 Wine 实现相同指令集下的虚拟机架构

Fig. 2 Virtual machine architecture of Wine implementation

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

的运行主要依赖于 NTDLL,KERNEL32,GDI32 和 USER32 4 个动态库. Wine 已经基本实现该 DLL 动态库与 UNIX API 的对应. Wine 与 Windows NT 架构对应关系如图 3[8].

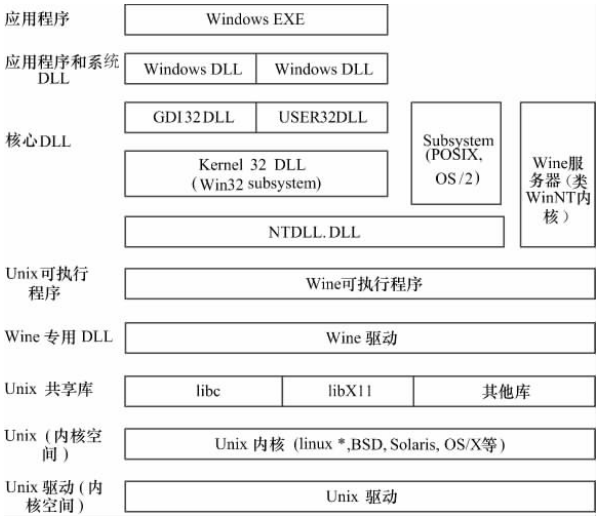


图 3　Wine 架构以及与 Windows NT 架构的对应关系

Fig. 3　Corresponding relation between Wine architecture and Windows NT architecture

2.3　ISA 仿真虚拟机 Bochs

Bochs 是开源的 X86 仿真平台. 它能够支持多种不同指令集实现跨指令集的 X86 仿真. Bochs 通过在不同的操作系统与指令集和目标应用之间构建一个虚拟的仿真层(图 4),将目标软件执行的代码指令解释为主机可识别指令. Bochs 可以模拟 Intel X86 CPU、通用 I/O 设备和客户 BIOS,还能够模拟运行大部分的操作系统,如 Linux,DOS,Windows 95/98,Windows NT/2000/XP 等[9]. Bochs 能够被编译和使用为多种模式,通常使用 Bochs 来提供完整的 X86 PC 模拟器,包括 X86 处理器,硬件设备和内存.

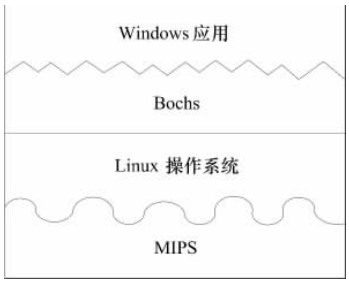


图 4　利用 Bochs 实现在 MIPS 架构下的 Windows 应用

Fig. 4　Application of Windows in MIPS framework using Bochs

2.4　Wine 和 Bochs 交互设计

由上文可以得出,Windows 应用程序在运行中,可能会调用 Wine 内置动态库,或者 Windows 原生动态库. 在本系统中,由于应用程序和 Windows 原生动态库是 X86 指令集的二进制制品,他们都要通过 Bochs 来模拟执行.

系统在运行 Windows 应用程序时,整个进程地址空间中存在着如下几种形态的线性区:①Windows 应用程序,PE 格式,X86 指令;②Windows 原生动态库,PE 格式,X86 指令;③Wine 内置动态库,ELF 格式,MIPS 指令;④Bochs 模块,ELF 格式,MIPS 指令. 相应的,系统在运行时,有两种执行形态:①Wine 内置动态库和 Bochs 模块,在龙芯 CPU 上直接执行;②Windows 应用程序和 Windows 原生动态库,通过 Bochs 模拟执行.

系统在运行时,需要在线性区之间进行交互,由一个线性区到另一个线性区的切换往往通过函数调用. 根据线性区类型和函数调用方式,文中把交互分为如下 4 种类型,并针对每种类型,进行相应的接口设计.

1)从 X86 指令线性区调用到 X86 指令线性区. X86 指令的线性区都是在 Bochs 上执行,这种类型的调用方式在 X86 指令内部已经完成,并不需要多余的操作.

2)从 MIPS 指令线性区调用到 MIPS 指令线性区. MIPS 指令的线性区是在龙芯 CPU 上直接执行,这种类型的调用方式也不需要二进制指令外的多余的操作.

3)从 X86 指令线性区调用到 MIPS 指令线性区. X86 指令的线性区是在 Bochs 上执行的,在本系统中,这种类型的调用存在于两个地方:Bochs 对指令 JMP 和 CALL 的模拟,系统需要在这两处地方进行相应的判断,如果判断为到 MIPS 指令线性区的调用,则跳转到 MIPS 指令执行.

4)从 MIPS 指令线性区调用到 X86 指令线性区. 回调就是这种类型的一个典型. 当回调发生时,一部分回调会跳转到 Wine 内置动态库,另一部分会跳转到 X86 指令部分. 除了回调,Wine 的内置动态库中也会有向 X86 指令部分的调用.

系统设计了相应的调用方式,如图 5 所示.

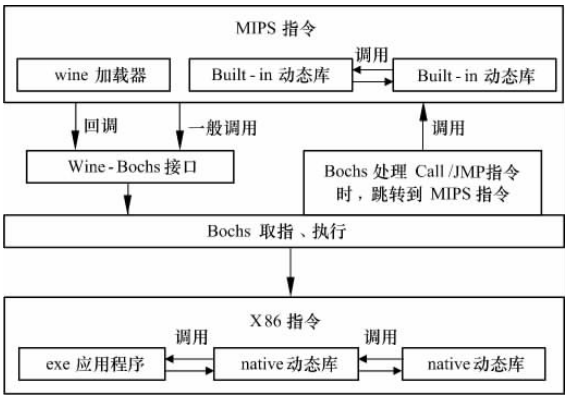


图 5　Wine 和 Bochs 交互设计框图

Fig. 5　Interaction design of Wine and Bochs

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

2.5 Wine 和 Bochs 执行的双向调用机制实现

2.5.1 X86 指令线性区调用 MIPS 指令函数

系统开始运行时,会统计所有 MIPS 动态库中 API 的名字、地址、参数个数等,并把它们记录在数据表中. Bochs 在对指令 JMP 和 CALL 模拟时,首先获得要跳转到的指令的地址,该地址存储在 EIP (extended instruction pointer) 寄存器中,系统判断该 EIP 是指向 X86 指令空间,还是 MIPS 指令空间. Bochs 在初始化时,遍历了该进程空间中所有的 MIPS 动态库,查表获取每个 MIPS 动态库的基地址和大小. 系统通过将 EIP 的值和该数据表中地址空间范围比较,来确定该 EIP 是否指向 MIPS 指令空间. 如果该 EIP 属于 MIPS 指令空间,则进行相应的处理,跳转到 MIPS 函数执行;否则继续按照 Bochs 的方式执行.

2.5.2 MIPS 指令线性区调用 X86 指令函数

这种类型的调用主要分为两种情况:①回调函数;②PE 格式的动态库在加载完成后,要执行它的入口函数,执行入口函数需要进入 X86 指令空间.

在 Bochs 中,本文设计了开启 Bochs 模拟执行的接口. 每个接口都是一个 X86 指令空间的函数调用,根据调用的参数个数设计不同的接口类型. 接口内部的实现中,首先要找到 Bochs 的虚拟 CPU,然后进行相应的现场保护,之后进行函数调用,调用完成后返回.

2.6 多线程的实现

Wine 中的程序通过 createThread 创建新的线程,由 Wineserver 进行线程调度,并发起线程的执行. 新创建的线程从一个入口开始执行,线程可能会执行到 X86 指令,需要 Bochs 来模拟执行.

由于每个线程都要有一个独立的模拟执行部件,而且在执行的时机上是物理并发的,因此需要创建多个 Bochs 模拟 CPU,每个模拟 CPU 用于执行一个线程中的 X86 指令.

3 系统评估

本文测试评估时基于龙芯 3A 硬件平台实现了一个原型系统. 实验系统的运行环境为龙芯 3A 8 核心 1Ghz 处理器,DDR800 2G 内存的服务器,中标麒麟高级服务器版 V6.0 操作系统. 系统评估实验设置了系统功能和系统性能两种类型的测试,其中:

1)系统功能测试:通过在原型系统中运行 Delphi,VC ++ 开发的简单单线程程序和多线程程序,以及复杂的 MIS 系统程序来评估系统功能的完善性.

2)系统性能测试:以 Windows 下运行记事本程序的响应时间作为基准,通过本原型系统运行 Windows 下编译的记事本程序和 Linux 下通过 Wine 交叉编译生成的记事本程序的响应时间来评估系统性能.

3.1 系统功能测试

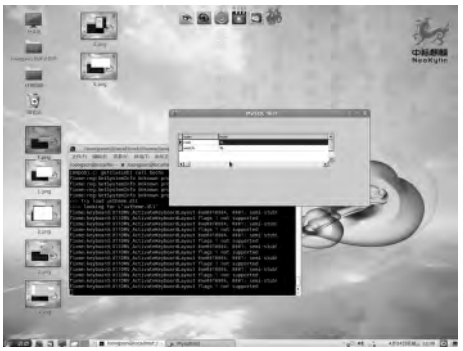
采用虚拟执行环境运行一个 delphi 程序,该程序绘制基本的 delphi 控件,并多线程的运行(图 6a)),采用该虚拟执行环境运行贪吃蛇,如图 6b). 采用该虚拟执行环境运行一个数据库客户端,该客户端采用 ODBC 的方式访问远程数据库服务器(如图 6c)). 该数据库客户端可以正常读写服务器里的数据.



a) 运行 delphi 程序



b) 运行 VC 程序



c) 运行数据库客户端

图 6 功能测试

Fig. 6 Function test

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

3.2　系统性能测试

获得 Windows 记事本可执行文件 notepad. exe,在 Linux 终端上使用命令 Wine notepad. exe 运行 notepad. exe 程序(图 7). 测试该应用在运行速度、响应时间、稳定性等性能是否能够达到一般用户的使用要求. 测试中与 Windows 平台进行比较,测试实例如表 1,测试结果如表 2.



图 7　运行 notepad. exe

Fig. 7　Run the noteped. exe

表 1　测试实例

Table 1　Test case

项目	实例 1	实例 2	实例 3
机器平台	联想 ThinkPad T60 1951 - IXU	本系统平台	本系统平台
操作系统	Windows XP	中标麒麟	中标麒麟
处理器	酷睿双核 T2300,主频 1667MHz	龙芯 3A,主频 900MHz	龙芯 3A,主频 900MHz
应用程序	Windows 上编译生成的 PE 文件 notepad. exe	Linux 上通过 Wine 生成的 ELF 文件 notepad. exe. so	Windows 上编译生成的 PE 文件 notepad. exe
执行方式	直接鼠标双击 EXE 文件	使用命令 Wine notepad. exe. so	使用命令 Wine notepad. exe

表 2　测试结果

Table 2　Test results

测试项目	实例 1	实例 2	实例 3
记事本启动时间	即刻启动	3 s	4 s
200 次窗口打开与关闭时间	10 s	33 s	43 s

3.3　测试结果分析

经系统功能测试,该虚拟环境支持 Delphi,C,C ++ 等多种语言开发的程序,以及支持网络、数据库等多种功能. 经系统性能测试,系统运行 notepad. exe 基本满足使用要求,但需要进一步改进. 同时,用 oprofile 软件分析影响虚拟环境性能的函数,为以后的优化提供了参考.

4　结论

文中设计的零修改软件移植中间件可以很好地完成 Windows 程序跨平台移植,基本实现了简单程序(notepad. exe)在龙芯平台上的运行. 通过优化结构,目标实现 Office 软件在龙芯平台上的正常运行. 这将使国家机关在自主化平台上的办公得以实现. 同时,可以基于该系统架构进行信息流的分析与追踪,提高系统安全保障.

参考文献(References)

[1] 陈丹丹. 软件移植实践[D]. 杭州:浙江大学,2007.

[2] Cifuentes C, Malhotra V. Binary translation: static, dynamic, retargetable [C] // *Proceedings of International Conference on Software Maintenance*. Monterey CA USA: IEEE - CS Press. 1996: 340 - 349.

[3] 陈乔, 蒋烈辉, 董卫宇, 等. 基于动态二进制翻译技术的仿真器研究[J]. 计算机工程, 2011, 37(20): 277 - 279.
Chen Qiao, Jiang Liehui, Dong Weiyu, et al. Simulator research based on dynamic binary translation technology [J]. *Computer Engineering*, 2011, 37(20): 277 - 279. (in Chinese)

[4] 黄聪会,陈靖,罗樵,等. 面向二进制移植的虚拟化技术[J]. 计算机应用研究,2012,29(11):4185 - 4187.
Huang Conghui, Chen Jing, Luo Qiao, et al. Virtualization technology for binary migration [J]. *Application Research of Computer*, 2012, 29(11): 4185 - 4187. (in Chinese)

[5] Smith E, Nair R. The architecture of virtual machines [J]. *Computer*, 2005, 38(5): 32 - 38.

[6] Smith J E, Nair R. 虚拟机:系统与进程的通用平台[M]. 北京: 机械工业出版社,2009:4 - 7.

[7] Wine User's Guide. Chapter 1 Introduction. [DB/OL]. http://www.winehq.org/docsen/wineusr-guide.html#introduction.

[8] Wine Developer's Guide. Chapter Ⅱ. Wine architect [DB/OL]. http://www.winehq.orgsite/docs/winedev-guide/×2884.

[9] Bochs User Manual. Chapter 1 Introduction to Bochs [DB/OL]. http://bochs.sourceforge.net.

(责任编辑: 童天添)

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net