

文章编号: 1004-4353(2018)02-0170-04

基于 Android 与 Linux 的云终端实现方案

杨雄, 舒晨, 黄小君, 刘雨婷
(福州大学 至诚学院, 福建 福州 350002)

摘要: 针对 Android 平台桌面虚拟化体验不理想的问题, 提出一种基于 Android 系统与 Linux 系统相融合的云终端实现方案. 首先利用 Libhybris 解决 Android 系统与 Linux 系统的兼容性问题, 然后实现 2D 图形显示和高清视频重定向功能, 最后通过基于 RK3188 的云终端平台验证该方案的可行性. 研究表明, 该方案实现的云终端在 2D 图形显示效率、多媒体性能和 USB 设备重定向等方面具有良好的特性, 具有一定的市场应用价值.

关键词: Android; Linux; 云终端

中图分类号: TP316 **文献标识码:** B

Cloud terminal implementation based on Android and Linux

YANG Xiong, SHU Chen, HUANG Xiaojun, LIU Yuting
(Zhicheng College, Fuzhou University, Fuzhou 350002, China)

Abstract: In order to solve the problem of unsatisfactory desktop virtualization experience on the Android platform, a cloud terminal implementation scheme integrating the Android system and the Linux system is proposed. First we use Libhybris to solve the Android system and Linux system compatibility issues, and then achieve 2D graphics display and high-definition video redirection capabilities, and finally through the RK3188-based cloud terminal platform to verify the feasibility of the program. The research results show that the cloud terminal implemented by the scheme has good features in terms of 2D graphics display efficiency, multimedia performance and USB device redirection, and has a certain market application value.

Keywords: Android; Linux; cloud client

Android 是目前最流行的操作系统之一, 云终端的选型也从传统 PC、专用云终端扩展到基于 Android 平台的 ARM 设备上, 但目前主流桌面虚拟化协议 Citrix 的 ICA^[1]、VMware 的 PCOIP^[2] 和 Red Hat 的 SPICE^[3] 对 Android 系统的支持非常有限, 因而在 ARM 云终端上更多的是使用 Linux 系统. 然而, ARM 厂商却更青睐于 Android 系统, 为 Android 系统提供了功能齐全的闭源驱动程序, 而不再提供 Linux 系统的硬件驱动程序. 目前, 国外对云终端的相关研究主要集中在 X86 架构的设备, 而国内对 Android 平台云终端的研究多局限于对桌面虚拟化协议的优化. 例如: 袁野等^[4] 提出了一种基于 Android 的云桌面客户端的实现方案, 该方案能够使客户端满足 2D 图形渲染, 但当鼠标移动较快时会出现明显的鼠标跳跃现象, 而且不支持多媒体和 USB 设备重定向功能. 王宁等^[5] 提出了一种基于 SPICE 协议的 Android 云桌面客户端的实现方案, 该方法仅能够支持连接云桌面功能, 不能支持多媒体应用和 GPU 加速. 杨子超^[6] 提出了一种基于 Android 的

RDP 客户端的实现方案,虽然该 RDP 客户端通过模拟硬件鼠标功能使用户无需改变在 PC 上的使用习惯,但该方案的显示性能对终端设备要求较高,在虚拟桌面中打开新程序或者文件时,由于桌面更新区域较大,导致 CPU 处理不及时而出现延迟和刷屏现象,并且该方案没能够利用 Android 平台的 GPU 硬件解码功能. 综上所述,针对基于 Android 云终端的研究虽然取得了一定的成果,但仍存在用户操作体验不佳、核心功能缺失和硬件资源无法充分使用等问题. 基于此,本文提出一种将 Android 系统与 Linux 系统相融合的云终端实现方案,并通过实验验证本文方案的有效性.

1 云终端系统的实现

1.1 驱动接口转换

Android 系统架构和 Linux 系统类似,如图 1 所示. 虽然 Android 系统与 Linux 系统都使用 Linux 内核,但是基于 Glibc 类库^[7]的 Linux 系统应用无法直接调用 Android 系统基于 Bionic Libc 的闭源驱动库^[8],即两者互不兼容;因此,需要通过 Libhybris 进行驱动接口转换,将基于 Bionic 的系统驱动库中的符号替换成与 Glibc 相兼容的版本符号后, Linux 系统中的图形显示框架和多媒体框架才能够对驱动接口进行调用. Libhybris 的工作原理如图 2 所示.



图 1 Android 系统架构

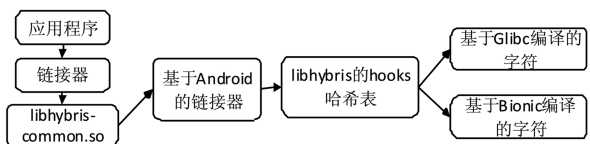


图 2 Libhybris 原理图

Libhybris 的 hooks 哈希表的关键代码如下:

```
hybris/common/hooks.c:
static struct_hook hooks[] =
{
    {"property_get",property_get},
    {"property_set",property_set},
    (...)
}

void * get_hooked_symbol(char * sym)
{
    struct_hook * ptr=&hooks[0];
    static int counter=-1;
    while (ptr->name!=NULL)
    {
        if (strcmp(sym,ptr->name)==0)
        {
            return ptr->func;
        }
        ptr++;
    }
    (...)
}
```

1.2 2D 图形显示的实现

目前,基于 Android 云终端的显示效率低下是由于其显示输出没有调用 GPU 的硬件加速而采用 CPU 软件渲染,因此若要提高 2D 图形显示效率就必须使用 Android 系统的硬件加速功能. 为此,本文利用 Linux 系统的显示框架 Wayland,通过 Libhybris 调用 Android 系统原生显示服务 SurfaceFlinger^[9]来实现基于硬件加速的应用程序界面渲染. Wayland 框架如图 3 所示.

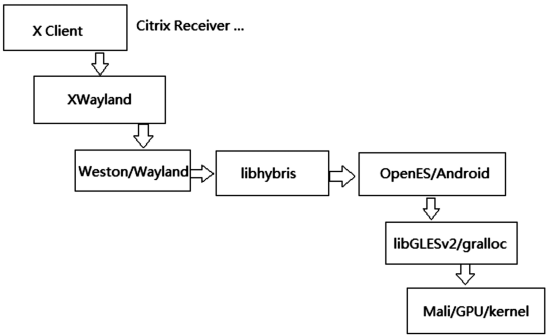


图 3 Wayland 框架图

Wayland 调用 SurfaceFlinger 服务的关键代码如下:

```
SfClient * sf_client_create_full (bool egl_support)
{
    SfClient * client=new SfClient();

    client->client=new android::SurfaceComposerClient();

    (...)

    client->egl_support=egl_support;
    if (egl_support) {
        client->egl_display=eglGetDisplay(EGL_DEFAULT_DISPLAY);
        (...)
    }

    return client;
}
```

1.3 高清视频硬解码的实现

基于 Android 云终端的多媒体性能不足主要是由于采用 CPU 软解码来实现视频解码,占用了大量 CPU 时间;因此,若要提高云终端的多媒体性能,就必须通过硬解码来实现高清视频(1 080 P)的流畅播放.为此,本文利用 Linux 系统的多媒体框架 Gstreamer^[10],通过基于 Libhybris 的插件 Gst-Hybris 调用 Android 系统的媒体硬解 OMX 模块来实现在低 CPU 使用率下基于硬解码的高清视频流畅播放. Gstreamer 框架如图 4 所示.

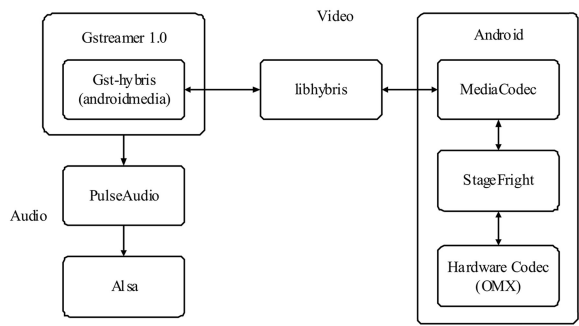


图 4 Gstreamer 框架图

实现 2D 图形显示和高清视频硬解码后,将 Android 系统与 Linux 系统相融合,即在 Android

系统的底层驱动之上构建 Linux 系统的图形应用. Android 系统与 Linux 系统融合的框架如图 5 所示.该融合系统保留 Android 系统原生的 HAL 及相关服务,系统启动时首先进入 Android 系统并开启相关服务,然后 chroot 切换进入 Linux 系统,最后启动云终端图形界面和 ICA 协议客户端.系统启动流程如图 6 所示.

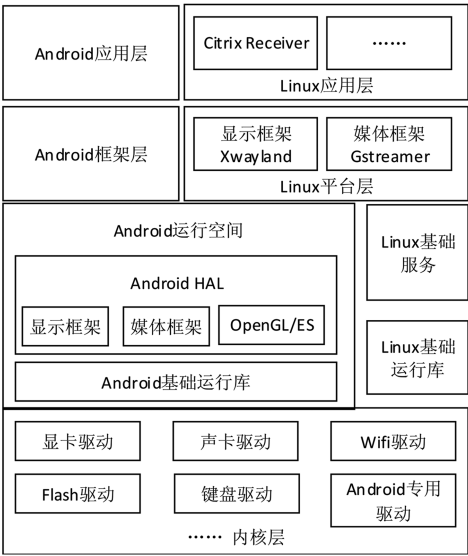


图 5 融合系统框架图

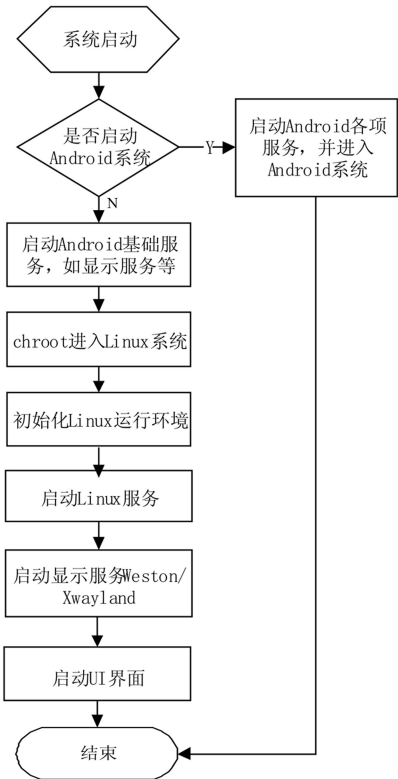


图 6 系统启动流程图

2 测试与分析

测试平台基于瑞芯微的 RK3188 处理器芯片,该芯片主频为 1.5 GHz,支持 OpenGL ES 2.0 和 1920×1080 分辨率.平台搭载 Android 4.4.2 和基于 Ubuntu 14.04 构建的 Scratchbox2 文件系统.测试平台如图 7 所示.

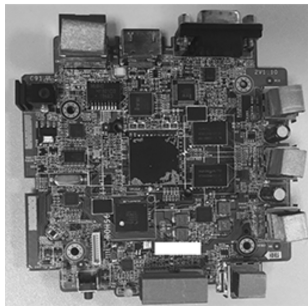


图 7 基于 RK3188 的云终端

对云终端的验证主要包括:2D 图形显示效率、多媒体性能和 USB 设备重定向功能.

在 Citrix 虚拟桌面环境中,利用 Citrix 的测试工具收集帧数,依次对各种典型场景进行验证.结果显示,该方案每秒更新帧数大约为 20,未出现卡顿现象,具有良好的 2D 图形显示效率.以典型的 Excel 滚屏场景为例,该方案云终端帧数稳定,无刷新迟滞现象.另外,本方案中基于 Citrix Receiver 的开放接口(H.264)能够实现 JPEG 硬解码功能.因此本文方案既能充分发挥该平台的 GPU 性能,又可以提高用户体验和降低带宽.

以 Citrix 的认证视频为测试源,利用 Gstreamer 的 gst-launch 命令验证系统本地多媒体播放性能.测试结果显示,该方案支持流畅播放 1080 P 高清视频.另外,在虚拟桌面中,通过 Citrix 的多媒体重定向功能也能够调用 GPU 的硬解码功能实现 1080 P 高清视频的流畅播放.

启用 USB 设备重定向策略后,USB 设备能够从云终端重定向到 Citrix 虚拟桌面,获得与本地系统访问设备的相同体验,说明该方案支持通用 USB 设备重定向功能.

3 结束语

本文提出的基于 Android 的云终端,通过 Libhybris 使 Android 系统与 Linux 系统相兼容,以 Android 系统的驱动为基础构建 Linux 系统应用,将 Android 系统的驱动优势与 Linux 系统的图形应用优势相结合,解决了传统基于 Android 的云终端所存在的功能缺失和体验不理想的问题,扩大了基于 ARM 的云终端的选型范围.但因为 Android 系统与 Linux 系统的双显示器输出机制不同,所以目前该测试平台仅支持单显输出功能,今后我们将进一步研究 RK3188 厂商提供的 Linux 内核源码,以实现双显示器输出功能.

参考文献:

- [1] Li J, Jia Y, Liu L, et al. CyberLiveApp: A secure sharing and migration approach for live virtual desktop applications in a cloud environment[J]. Future Generation Computer Systems, 2013, 29(1): 330-340.
- [2] Calyam P, Patali R, Berryman A, et al. Utility-directed resource allocation in virtual desktop clouds[J]. Computer Networks, 2011, 55(18): 4112-4130.
- [3] 徐浩,兰雨晴.基于 SPICE 协议的桌面虚拟化技术研究及改进方案[J].计算机工程与科学,2013,35(12):20-25.
- [4] 袁野,何嘉.基于 Android 的云桌面客户端关键技术研究及实现[J].计算机应用,2013,33(a01):26-28.
- [5] 王宁,王浩杰,屈喜龙.基于 SPICE 协议的 Android 云桌面客户端研究[J].湖南工程学院学报(自然科学版),2016,26(2):33-36.
- [6] 杨子超.基于 Android 的 RDP 客户端的设计与实现[D].成都:电子科技大学,2013.
- [7] Block F, Dewald A. Linux memory forensics: dissecting the user space process heap[J]. Digital Investigation, 2017, 22:S66-S75.
- [8] 李钊.面向 64 位 UniCore 体系结构的 C 语言函数库的移植与优化[D].北京:北京大学,2012.
- [9] 江帆,贺也平,周启明. SurfaceFlinger 在 X Window 系统环境下的运行方案[J].计算机系统应用,2017, 26(10):95-101.
- [10] 宫健.基于 Gstreamer 的嵌入式流媒体传输系统的研究与实现[D].南京:南京邮电大学,2016.