

基于 SPICE 协议的高性能虚拟桌面系统优化设计

张倩¹ 徐聪¹ 李楠¹ 高树论¹ 郑凌兵¹

ZHANG Qian XU Cong LI Nan GAO Shulun ZHENG Lingbing

摘要

针对高性能应用场景下,基于 SPICE 协议的虚拟桌面普遍存在体验效果不佳的问题,文章提出了一种融合 GPU 虚拟化技术的 SPICE 协议优化方案。核心在于将 Windows DXGI 截屏技术与 SPICE 协议深度结合,设计并实现了一套完整的虚拟桌面系统。通过搭建高负载虚拟桌面测试环境进行验证,结果表明:优化后的系统显著提升了传输性能,可有效适配需要 GPU 加速的高性能应用场景,解决了传统 SPICE 虚拟桌面的体验瓶颈,保障用户获得最佳使用体验;同时,整套系统设计合理、运行稳定,具备良好的可行性与应用价值。

关键词

虚拟桌面; SPICE; GPU 虚拟化

doi: 10.3969/j.issn.1672-9528.2026.01.004

0 引言

随着虚拟桌面(VDI)技术的成熟,逐渐成为企业办公的核心工具。用户对体验的要求不断提升,期望 VDI 能媲美高端图形工作站,提供流畅的高清视频、3D 建模等高性能应用支持。为此,VDI 架构需升级强化图形渲染能力,以满足日益增长的 GPU 计算需求。

目前业内主流的虚拟桌面传输协议^[1]主要有 VMWare 的 PCoIP 协议、Citrix 的 ICA 协议^[2]、Microsoft 的 RDP 协议和 RedHat 的 SPICE 协议^[3]等。其中 SPICE 协议是唯一完全开源的协议,且传输采用 TCP 协议安全性较高不易丢帧,但是 SPICE 协议的不足之处也很明显,如视频处理能力差、带宽占用过大、服务器响应延迟、不支持 3D 显示等。业界众多厂商基于开源 SPICE 协议进行二次开发,针对特定场景优化 SPICE 协议的不足,形成关键技术能够自主掌控的虚拟桌面产品。

在高性能应用场景下,针对视频播放 SPICE 协议默认采用 MJPEG 压缩算法,其对每一帧进行处理,压缩质量较好但压缩效率较大,对带宽要求较高,不适合高清视频资源的网络传输。文献[4]提出帧率与图像算法结合的视频区域帧测算法获取视频播放区域并使用 MPEG4 压缩算法,减少视频传输冗余,降低视频传输带宽;文献[5]提出基于 SPICE 协议的视频重定向方案,通过视频流的方式将视频传送到客户端,由客户端进行解码播放,从而解决 3D 视频播放的问题;文献[6]提出针对虚拟桌面高端应用场景如专业 3D 软件、实时监控等,采用 GPU 透传的方式提高桌面云用户的用户体验,但是虚拟机独占 GPU 成本较大。这些文献一部分针

对 SPICE 的视频处理算法进行优化,一部分结合 SPICE 协议改进增加功能模块优化视频处理方式。本文对 SPICE 协议进行深入研究,主要解决 SPICE 协议在高清视频传输及 3D 显示等高性能场景下桌面体验效果不佳的问题,通过结合基于 KVM^[7]的 GPU 显卡虚拟化技术,提出基于 GPU 虚拟化技术的 SPICE 协议优化方案,构建满足用户最佳体验的虚拟桌面系统^[8],从而支持 OpenGL 和 DirectX 标准及主流的 3D 应用程序,满足用户对 GPU 在图形、计算机人工智能方面的需求,提供良好的开源解决方案。

1 SPICE 协议优化

SPICE 协议由 4 个核心组件构成。SPICE Guest 负责捕获操作系统及应用程序的图形指令;SPICE Server 以动态库(libspice)形式为 QEMU 提供协议支持;SPICE Client 用于连接和操作远程虚拟机;SPICE Protocol 定义各组件间的交互规则和数据格式。如图 1 所示。

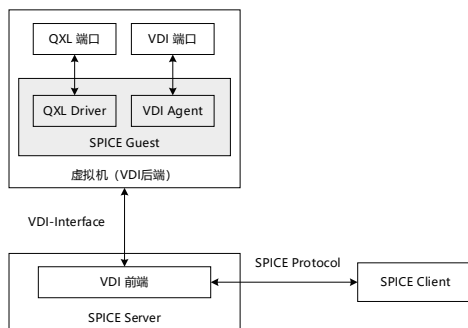


图 1 SPICE 协议组成

SPICE 协议默认同虚拟显卡 QXL 设备对接,通过 QXL 驱动捕获图形指令,经封装和压缩后由 SPICE Server 传输至 Client。但在 3D/ 高清视频等高性能场景, QXL 的 CPU 模拟

1. 中国电子科技集团公司第五十二研究所 浙江杭州 311121

方案难以满足需求。为此,本文基于 NVIDIA vGPU 技术,通过 GPU 加速实现高效图形处理,优化虚拟桌面系统性能。

1.1 视频传输分析

基于 SPICE 协议的常规图像流传输流程如图 2 所示,当帧间隔 ≤ 50 ms 时,系统采用 MJPEG 视频编码进行帧内压缩,并通过 UDP 传输至客户端。然而该方案存在局限性:

- (1) MJPEG 仅压缩帧内冗余,保留帧间冗余,压缩效率不足;
- (2) 高清视频数据量剧增,在有限带宽下易引发丢帧和画面卡顿问题。

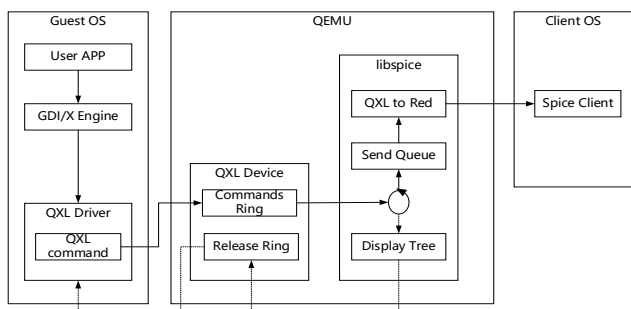


图 2 SPICE 协议图像流

在高性能应用场景下,当虚拟桌面启用 GPU (如 NVIDIA vGPU) 时,物理 GPU 将取代 QXL 成为主显卡,导致 SPICE 协议因无法获取 QXL 输出而出现黑屏问题。典型如 NVIDIA T4 等计算型 GPU 虽支持加速,但缺乏显示输出功能。因此需优化 SPICE、QXL、GPU 之间的协同机制,保持 QXL 作为虚拟显示设备,确保 GPU 渲染的图像通过 QXL 设备显示。

1.2 优化方案设计

针对高性能虚拟桌面应用场景,提出基于 Windows DXGI 截屏技术与 SPICE 协议结合的优化方案。如图 3 所示。

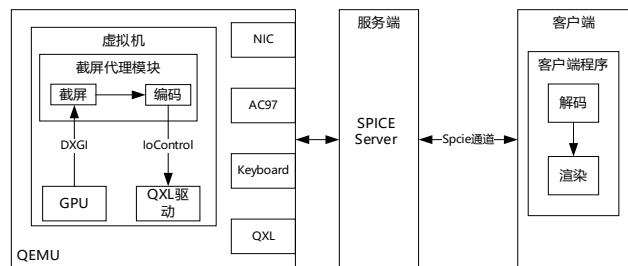


图 3 SPICE 协议优化结构图

DXGI 屏幕复制支持 Window8 及以上产品,通过直接在 GPU 层捕获屏幕数据,大幅降低 CPU 计算负载,释放虚拟机资源用于关键业务应用。工作流程采用三段式优化设计:

- (1) 应用程序输出画面由 DXGI 接口直接捕获;
- (2) 专用截屏代理模块实时进行高效压缩编码;
- (3) 处理后的数据流经 SPICE 协议通道直接传输至客户端,客户端获取到视频数据后进行解码和渲染显示。

该架构最大优势在于完全规避了服务端额外的画面渲染开销,相比传统方案大幅提升传输性能,特别适用于需要 GPU 加速的高负载虚拟桌面环境。

1.3 关键步骤设计

1.3.1 QXL 驱动优化

WDF (windows driver foundation) 驱动程序框架是微软为 Windows 操作系统提供的一种驱动程序开发模型。本文将原生 QXL 驱动改用 WDF 框架进行重新编码,同时修改 QXL 驱动的 Vendor ID 和 Device ID,操作系统将识别 QXL 设备为普通 PCI 设备,而非显示设备,从而解决在 GPU 显卡下的 QXL 驱动无法加载的问题。同时定义一些 IO 控制码用于和截屏代理模块的通信:

- (1) IOCTL_SURFACE: 主表面创建控制码;
- (2) IOCTL_BITBLT: 位块传输控制码;
- (3) IOCTL_CUSOR_POSITION: 设置光标位置控制码;
- (4) IOCTL_CUSOR_SHAPE: 设置光标形状控制码。

1.3.2 数据捕获

截屏代理模块采用 DXGI 技术进行图像数据捕获,而不需要对每个像素进行 CPU 处理。截屏代理模块首先创建 D3D 设备,并调用 DuplicateOutput 方法来创建 DXGI 输出复制对象,然后周期性调用输出复制对象 AcquireNextFrame 方法捕获当前桌面的图像数据,存在 DXGI_OUTDUPL_FRAME_INFO 类型的结构体变量 FrameInfo 中,截屏代理模块调用 DeviceIoControl 函数下发 IOCTL_SURFACE 控制码,用于主表面创建。

同时,截屏代理模块从 FrameInfo 变量中直接提取光标位置信息,调用 DeviceIoControl 函数下发 IOCTL_CUSOR_POSITION 控制码用于光标位置设置。除了光标位置,在虚拟机桌面环境中,光标类型对于用户体验也非常重要。单色光标 (MONO) 在显示简单的文本或图形时可能足够用,但在需要更精细控制光标与背景融合的场景下,透明度光标 (ALPHA) 则更为合适。截屏代理模块调用输出复制对象的 GetFramePointerShape 方法获取当前桌面光标的形状信息,然后调用 DeviceIoControl 函数下发 IOCTL_CUSOR_SHAPE 控制码用于设置光标形状为单色光标或透明度光标。

最后,截屏代理模块从 FrameInfo 变量中提取执行 BITBLT 操作所需的所有信息,例如源和目标的矩形区域、颜色深度、位图数据等,调用 DeviceIoControl 函数下发 IOCTL_BITBLT 控制码用于位块传输。当窗口内容更新时,BITBLT 可以用来快速刷新屏幕。

1.3.3 数据传输

优化后的 QXL 驱动程序采用原始的 SPICE 协议通道 (channel) 进行数据传输。主通道 (SPICE_CHANNEL_MAIN) 用于传输基本的显示数据,其他通道包括用于音频、剪贴板、USB 设备等。SPICE 协议对发送到客户端的图

像使用无损压缩算法，以避免重要显示对象的破坏。同时，SPICE 实现了客户端图像缓存，以减少向客户端的冗余传输。服务器和客户端之间的缓存是同步的，服务器会根据缓存容量决定是否添加或删除缓存项。

1.3.4 客户端渲染

客户端接收到 SPICE 协议消息后，根据消息内容更新本地的显示，客户端可跨平台，支持 Windows、Linux 等操作系统。

2 虚拟桌面系统设计

2.1 基于 OpenStack 架构桌面服务端

本文基于开源 OpenStack^[9] 项目构建桌面服务端环境，通过 VDI 虚拟化方式实现远程虚拟桌面。如图 4 所示。

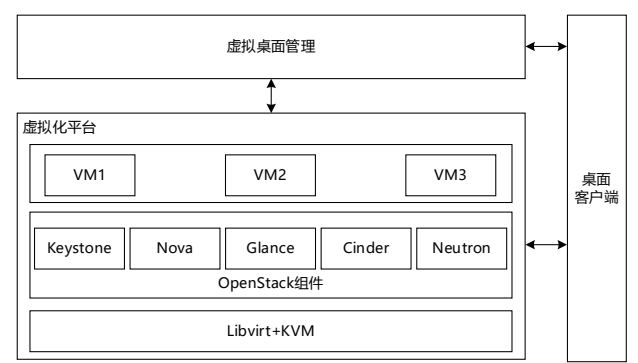


图 4 OpenStack 桌面服务端架构

（1）虚拟化平台

虚拟化平台为虚拟桌面提供了基础设施服务，基于开源 OpenStack 项目进行二次开发，构建虚拟化资源池，包括计算资源、存储资源和网络资源，提供资源调用接口。

（2）虚拟桌面管理

虚拟桌面管理提供用户管理、桌面调度、策略管理以及云终端管理等服务，合理分配资源。

（3）桌面客户端

云终端是显示和操作虚拟桌面的终端设备，通过 SPICE 协议进行虚拟机桌面访问，在云终端中需要部署客户端软件程序，用于用户登录和访问虚拟桌面。

2.2 基于 Electron 框架桌面客户端

本文采用 Electron 框架，Electron 是一个使用 JavaScript、HTML 和 CSS 构建桌面应用程序的框架。基于 Chromium 浏览器内核和 Node.js，使之成为一个能够跨平台（Win、Mac、Linux）运行 JavaScript 代码的二进制可执行程序。如图 5 所示。

Electron 是跨平台客户端开发框架，其内嵌的操作界面使用 VueJS 进行开发。针对本地管理操作（如启动终端管理），使用 NodeJS 的 childprocess 执行 CLI 子进程执行。基于 Electron 框架的桌面客户端软件界面设计如图 6~7 所示。

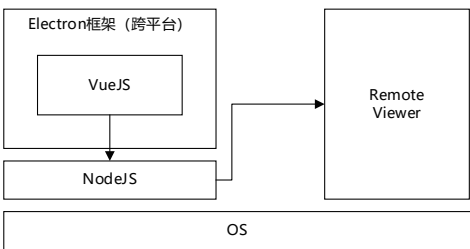


图 5 虚拟桌面客户端开发框架



图 6 客户端登录

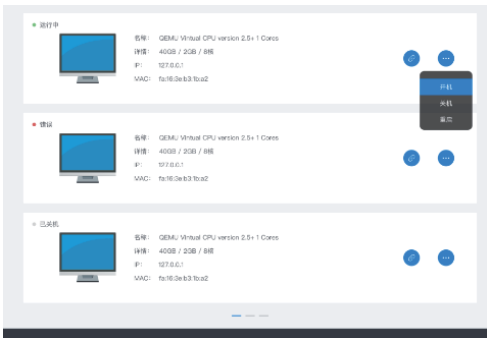


图 7 虚拟桌面列表

3 实验分析

3.1 实验环境

本文虚拟桌面系统测试环境包括测试搭建的硬件环境和软件环境。硬件环境主要由运行虚拟化平台的服务器设备和 SPICE 客户端设备组成，硬件环境如表 1 所示。

表 1 硬件环境

设备名称	参数
GPU 服务器	CPU: Intel 至强银牌 4114 处理器 ×2 (2.2 GHz, 10 cores)
	内存: 128 GB 硬盘: 128 GB 操作系统: Centos8 GPU 显卡: NVIDIA Tesla T4, 16 GB 显存
客户端	CPU: Intel J4125 处理器 ×1 (2 GHz, 4 cores) 内存: 8 GB 硬盘: 128 GB 操作系统: Ubuntu18.04

在客户端上部署桌面客户端软件，在 GPU 服务器上搭建 all-in-one 的虚拟化平台 OpenStack 环境，提供虚拟机管理软件，软件环境如表 2 所示。

表 2 软件环境

软件名称	软件版本
QEMU	6.0.0
SPICE	0.15.0
SPICE-protocol	0.14.3
SPICE-gtk	0.39

3.2 实验结果及分析

本文针对高性能场景设计测试用例，对比普通虚拟桌面和 vGPU 虚拟桌面的测试效果。

(1) 播放 1080P 高清视频

在普通虚拟桌面和 vGPU 虚拟桌面上分别播放 1080P 高清视频，相比普通虚拟桌面，vGPU 虚拟桌面 1080P 视频在窗口移动情况下播放无卡顿和延迟、播放界面清晰。如图 8 所示。

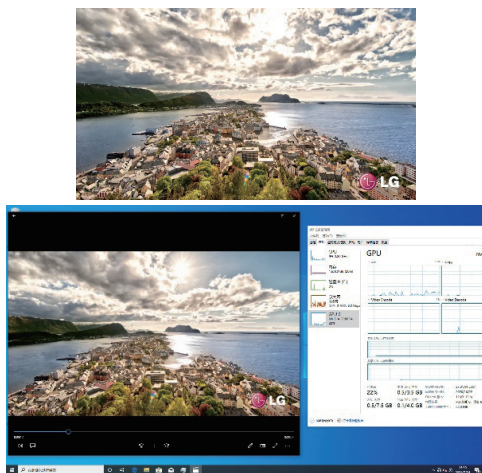


图 8 vGPU 虚拟桌面测试结果

(2) FurMark 压力测试

采用 FurMark 测试工具进行性能测试，尤其是 OpenGL 基准的测试工具，通过持续 3D 渲染测试显卡功耗和性能稳定性。在 vGPU 虚拟桌面上运行 FurkMark 工具，在显卡满载情况下能够平稳运行。如图 9 所示。

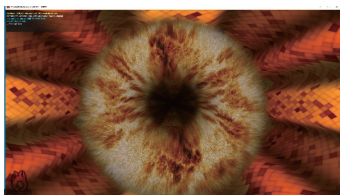


图 9 FurMark 测试结果

4 结论

本文简单介绍了虚拟桌面的应用场景及主流的传输协议，重点分析了 SPICE 协议的架构及原理。针对高清视频播放及 3D 应用程序等高性能应用场景提出了基于 GPU 的优化方案，提供了更加流畅的视频播放体验及更好地支撑了 3D 应用程序。同时对虚拟桌面系统进行了设计实现，基于虚拟桌面系统进行了测试与分析，验证了整套方案的可行性。

下一步的优化方向包括：（1）持续利用 GPU 的性能进行视频硬编码优化，进一步提升视频流畅度和用户体验；（2）优化网络传输效率，自适应调整传输带宽；（3）优化支持多个显示器数据显示。

参考文献：

- [1] 李康,陈清华.分析对比 VDI 常用的几种远程桌面协议[J]. 信息系统工程, 2020(1): 111-113.
- [2] 曹伟峰.基于 CITRIX 的桌面虚拟化在教学中的应用方案[J]. 信息技术与信息化, 2017(7): 79-82.
- [3] 杨声远,吴朋朋,雍文龙.SPICE 桌面协议的研究与优化[J]. 计算机时代, 2023(11): 108-111.
- [4] 侯文慧,王俊峰.面向云桌面协议的视频区域帧测算法[J]. 计算机应用, 2018, 38(5): 1463-1469.
- [5] 彭高贤.云桌面 3D 视频处理技术研究及优化[J]. 电子技术与软件工程, 2020(19): 180-181.
- [6] DONG Z L, WANG Z B, YUAN J K. Research and application of GPU pass-through based on KVM in graphics workstation[C/OL]//Conference: 2018 3rd International Workshop on Materials Engineering and Computer Sciences (IWMECS2018).Berlin:Springer,2018[2025-12-22].https://www.researchgate.net/publication/324500926_Research_and_application_of_GPU_pass-through_based_on_KVM_in_graphics_workstation.DOI:10.2991/iwmeecs-18.2018.76.
- [7] ALGARNI S A, IKBAL M R, ALROOBAEA R, et al. Performance evaluation of Xen, KVM, and Proxmox hypervisors[J]. International journal of open source software and processes (IJOSSP), 2018, 9(2):39-54.
- [8] WANG J J, SONG T,WU J W,et al. FastDesk:a remote desktop virtualization system for multi-tenant[J]. Future generations computer systems, 2018, 81: 478-491.
- [9] 武佳宁.基于 OpenStack 的虚拟云桌面系统建设方案[J]. 微型电脑应用, 2017, 33(5): 31-34.

【作者简介】

张倩(1990—)，女，江苏扬州人，硕士研究生，研究方向：计算机技术、大数据。

徐聪(1986—)，男，浙江杭州人，硕士研究生，研究方向：计算机技术、大数据。

李楠(1982—)，男，山西长治人，硕士研究生，研究方向：计算机技术、人工智能。

高树论(1985—)，男，浙江温州人，硕士研究生，研究方向：计算机技术、人工智能。

郑凌兵(1986—)，男，浙江台州人，硕士研究生，研究方向：计算机技术、大数据。

(收稿日期：2025-07-07 修回日期：2026-01-13)