Research on Key Technologies of RDV5-Based Cloud Desktop for University Computer Labs

Zhiyuan Wu[1,a], Yinqian Cheng[2,b,*]

[1]*School of Artificial Intelligence, China University of Geosciences (Beijing), Beijing, China*
[2]*Information Network Center, China University of Geosciences (Beijing), Beijing, China*
[a]*wuzy@cugb.edu.cn*, [b]*chengyq@cugb.edu.cn*
[*]*Corresponding author*

Abstract: *With the deepening of informatization in higher education, the operation and maintenance of public computer labs face challenges such as low system deployment efficiency, difficult software environment maintenance, and significant data security risks. Cloud desktop technology offers an effective solution to these problems. This paper investigates the RDV5 cloud desktop system, systematically analyzing its VOI (Virtual OS Infrastructure) architecture based on four design principles: centralized storage, distributed computing, disk-network convergence, and data offloading. Key technologies including system architecture, network boot process, master disk and copy-on-write mechanism, multi-level caching with filter drivers, and offline hot-switching are elaborated through diagrams and descriptions. Combined with the practical deployment requirements of university computer labs, the application value of this system in unified management, rapid deployment, automatic restoration, and data security is analyzed.*

Keywords: *cloud desktop, VOI architecture, PXE network boot, virtual disk, university computer lab*

1. Introduction

As higher education informatization deepens, public computer labs in universities have become essential infrastructure for teaching, research, and examinations. However, traditional stand-alone management approaches face increasingly severe challenges: (1) low system deployment efficiency, requiring individual installation and configuration for each terminal; (2) difficult software environment maintenance, with heterogeneous software requirements across different courses; (3) significant data security risks, with uncontrolled local data storage; and (4) slow fault recovery, requiring manual reimaging of failed machines[1,2].

Cloud desktop technology provides an effective solution to these problems. Currently, three mainstream architectures exist[3,4]: VDI (Virtual Desktop Infrastructure), which uses server-side centralized computing; IDV (Intelligent Desktop Virtualization), which runs local virtual machines on terminals; and VOI (Virtual OS Infrastructure), which runs the OS directly on bare-metal terminals without a virtualization layer. Among these, VOI achieves the best performance and compatibility since the operating system runs natively on the terminal hardware, eliminating virtualization overhead[5], but VOI's main disadvantage is that computing and data reside on the terminal side, resulting in weaker centralized management, lower security isolation than VDI, and greater reliance on terminal hardware standardization and local maintenance.

This paper investigates the RDV5 cloud desktop system developed by Ruiqi Information Technology, which implements a VOI architecture based on four design principles: centralized storage, distributed computing, disk-network convergence, and data offloading. The core technology is the proprietary I-Driver virtual disk, which operates at the block level between the file system and physical disk layers[6,7]. The system has been deployed at scale in multiple universities, with deployment scales ranging from hundreds to over a thousand terminals. A comparison with VDI and IDV is provided in Table 1.

Table 1. Comparison of mainstream cloud desktop architectures.

Feature	VDI	IDV	VOI
Computing Location	Server centralized	Terminal local VM	Terminal bare-metal
Virtualization Layer	Server Hypervisor	Terminal Hypervisor	None
Network Dependency	High	Medium	Low
Performance	Server-limited	Virtualization overhead	Near physical PC (no virt. layer)
Peripheral Compat.	Poor	Medium	Good
Offline Capability	Not supported	Supported	Supported

2. System Architecture Overview

The RDV5 system adopts a three-layer architecture as shown in Fig. 1, consisting of the terminal layer (distributed computing), the network layer (Gigabit Ethernet), and the server layer (centralized storage). The terminal layer comprises PC terminals with local CPU, memory, and GPU resources that execute the operating system natively. The network layer carries multiple functions including PXE boot, block-level network storage (I-Driver), P2P data offloading, and management channels. The server layer hosts the RDV5 server running four core services: image management, terminal management, DHCP/TFTP boot services, and block-level storage (I-Driver Target)[8].

By executing locally on the terminal, the system eliminates server-side virtualization overhead. Under typical configurations, its interactive performance is close to that of a physical machine with equivalent specifications.

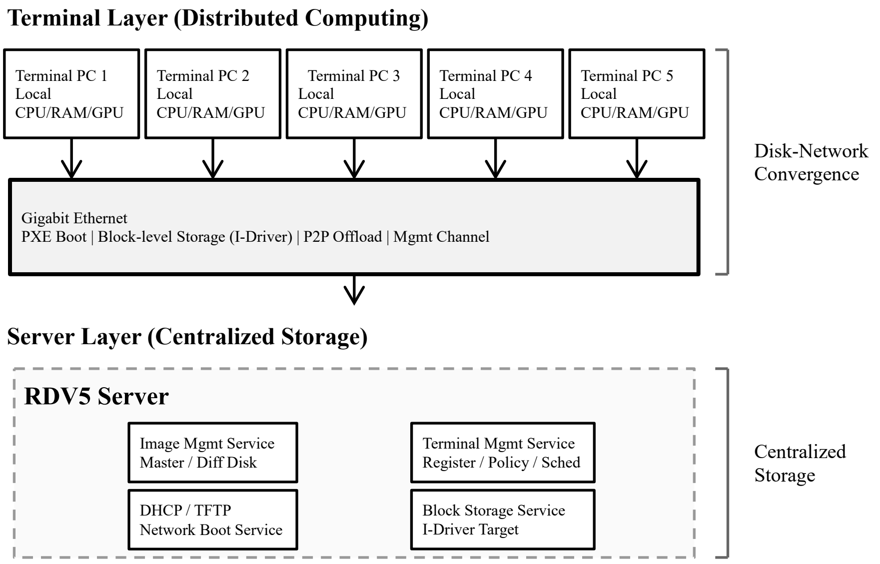

Fig. 1. Overall system architecture of RDV5.

2.1. I-Driver Virtual Disk Technology

I-Driver is a proprietary core technology developed by Ruiqi. It inserts a virtualization layer between the Windows file system layer and the physical disk layer, encapsulating SCSI commands within TCP/IP for transmission across the network. Similar to iSCSI[9,10] in concept, I-Driver is deeply optimized for cloud desktop scenarios, integrating cache management, differential disk support, and P2P data transfer capabilities.

2.2. UEFI Local Management Mode

In addition to the standard PXE network boot, RDV5 supports a UEFI local management mode that pre-installs the boot loader in the local EFI partition. This mode eliminates the need for DHCP/TFTP broadcast and is suitable for cross-VLAN or DHCP-restricted network environments.

3. Network Boot and Driver Loading

The complete terminal boot process is illustrated in Fig. 2 as a sequence diagram between the terminal PC and the RDV5 server. The process is divided into two phases: the UEFI phase and the Windows phase.

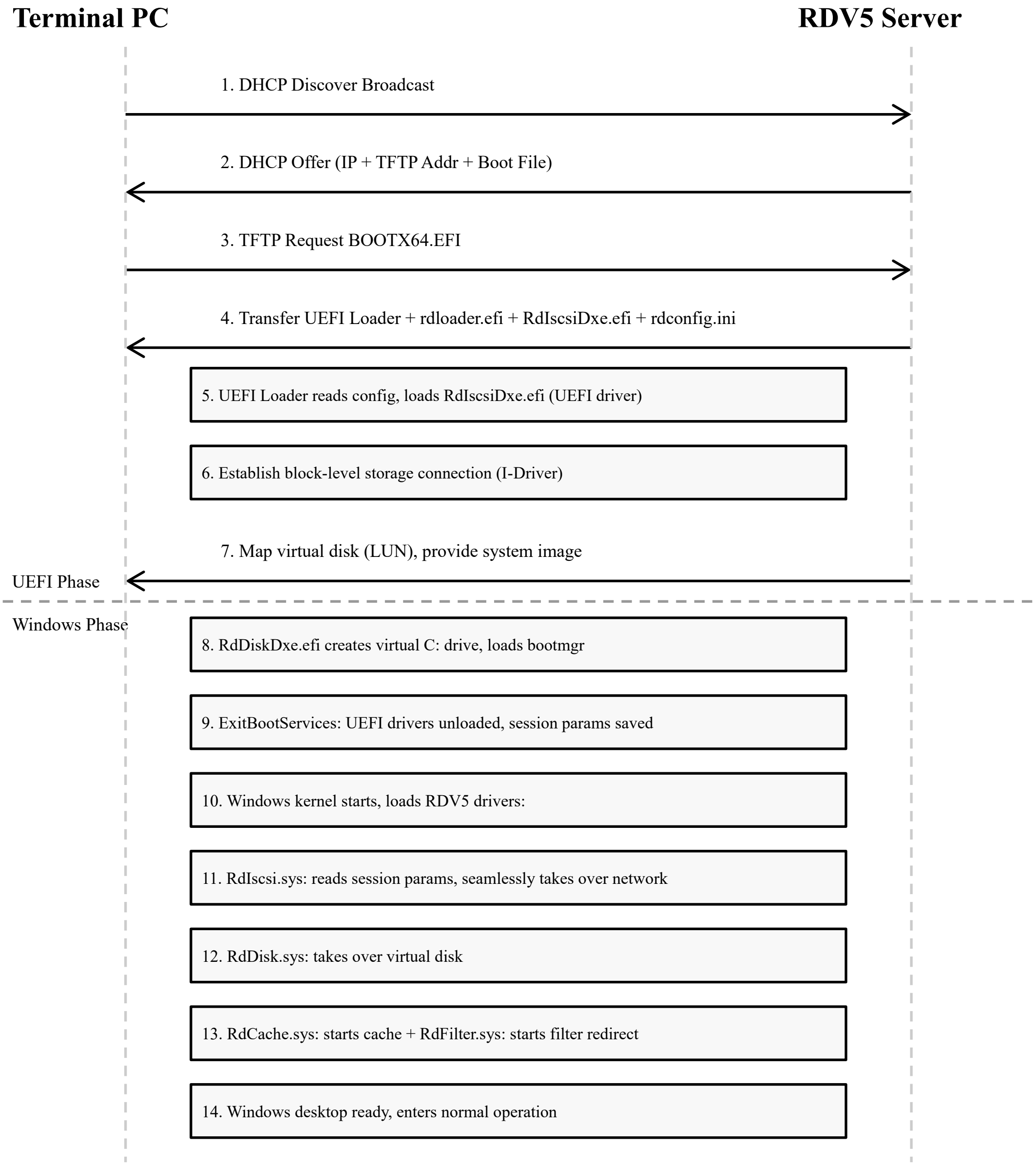

Fig. 2. Network boot and virtual disk mounting process.

3.1. PXE Boot and Boot File Loading

The boot sequence begins with a standard PXE process: (1) the terminal broadcasts a DHCP Discover message; (2) the server responds with a DHCP Offer containing the IP address, TFTP server address, and boot file name; (3) the terminal requests BOOTX64.EFI via TFTP; (4) the server transfers the UEFI boot loader along with rdloader.efi, RdIscsiDxe.efi, and the configuration file rdconfig.ini[6].

3.2. Two-Stage Driver Loading Mechanism

A fundamental challenge in diskless booting is the bootstrap paradox: the system image resides on the virtual disk, but the virtual disk driver itself is part of that image. RDV5 solves this through a two-stage driver loading mechanism[8].

Stage 1 (UEFI phase): The UEFI loader reads the configuration file and loads RdIscsiDxe.efi, which establishes the block-level network storage connection (I-Driver). RdDiskDxe.efi then creates the virtual C: drive and maps the LUN to provide the system image, enabling bootmgr to load.

Stage 2 (Windows phase): After ExitBootServices unloads the UEFI drivers (with session parameters preserved), the Windows kernel starts and loads four RDV5 drivers in sequence: RdIscsi.sys reads the saved session parameters and seamlessly takes over the network connection; RdDisk.sys takes over the virtual disk; RdCache.sys activates multi-level caching; and RdFilter.sys starts the file system filter for I/O redirection.

4. Image Management and Copy-on-Write

RDV5 employs a master-disk/differential-disk architecture with a copy-on-write (COW) mechanism, as illustrated in Fig. 3. This design enables efficient image sharing while providing per-terminal isolation[11,12].

4.1. Master Disk Sharing and Differential Disk Isolation

The master disk contains the complete operating system and pre-installed software, stored on the server as a read-only shared image. All terminals access the same master disk. Each terminal is assigned an independent differential disk that records only write modifications. Differential disks can be stored in three locations depending on persistence requirements: memory (lost on logout), local hard disk (optionally retained), or server (permanent roaming).

4.2. Read and Write Paths of Copy-on-Write

For read operations, the system first checks the differential disk; on a hit, data is returned directly; on a miss, the system reads from the master disk. For write operations, the COW mechanism copies the original block to the differential disk before modification, ensuring the master disk remains unmodified[13].

4.3. System Restoration

System restoration is achieved by simply discarding the differential disk, instantly reverting the terminal to the initial master disk state. In university computer labs, the differential disk is automatically cleared on each reboot, ensuring a clean environment for the next user session.

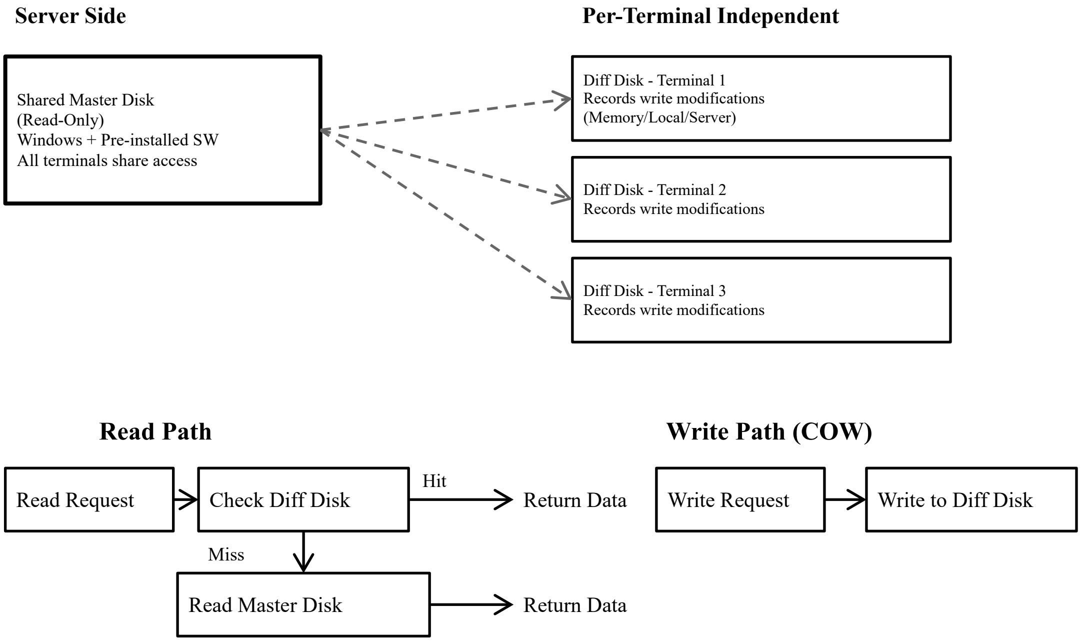

Fig. 3. Master disk, differential disk, and copy-on-write mechanism.

5. Multi-level Cache and Filter Driver

To minimize network I/O latency and support offline operation, RDV5 implements a multi-level caching architecture integrated with a Windows Minifilter driver, as shown in Fig. 4.

5.1. Minifilter Driver

RdFilter.sys is built on the Windows Minifilter framework[14]. It registers pre-operation callbacks to intercept IRP (I/O Request Packet) requests. Based on the cache index, it determines the data location and redirects the I/O accordingly: a cache hit results in a local read, while a cache miss triggers a network read followed by local cache backfill.

5.2. Three-Level Cache Architecture

The cache system consists of three levels with increasing latency: Level 1 (Memory Cache) provides microsecond-level access, dynamically allocating 30%-50% of available RAM with LRU eviction for the hottest data; Level 2 (Disk Cache) operates at millisecond-level access using a hidden partition of 30-50 GB, supporting on-demand caching and offline boot; Level 3 (Network Read) fetches data from the I-Driver block storage at 100ms-level latency, backfilling the local cache after each read. For a typical 60 GB system image, only 13-20 GB (22%-33%) actually needs to be cached.

5.3. Cache Management Strategy

The cache management employs an LRU algorithm with O(1) complexity using a hash table and doubly-linked list. A three-tier priority scheme is used: core system files are never evicted, frequently used applications receive priority retention, and temporary files are eligible for eviction. Additional optimizations include intelligent prefetching, P2P cache transfer to distribute server load, and incremental synchronization.

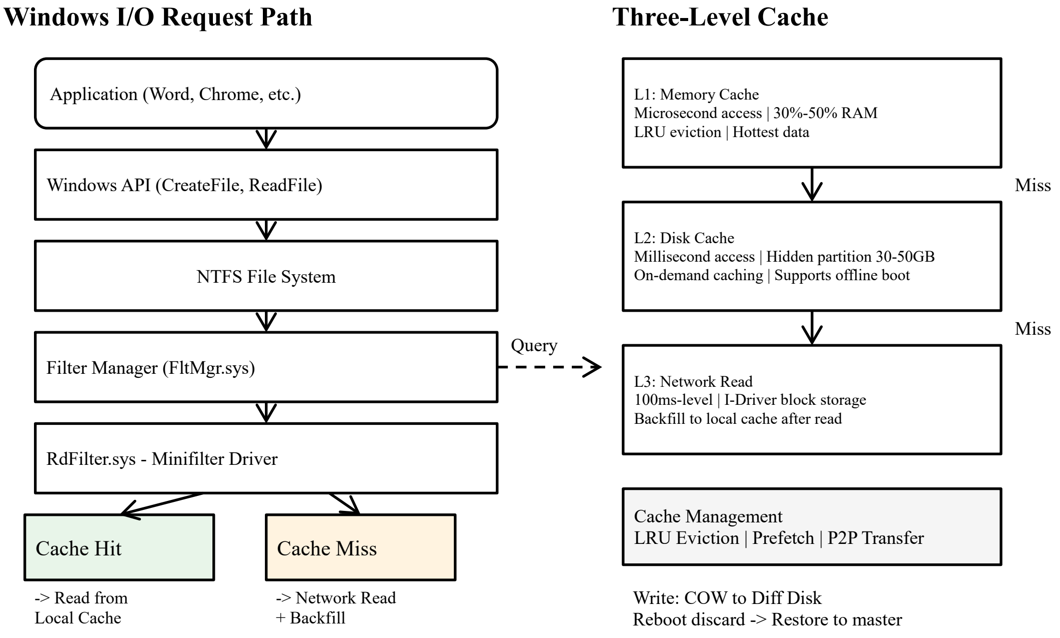

Fig. 4. Multi-level cache and filter driver architecture.

6. Working Mode Analysis

RDV5 supports three working modes: diskless mode, disk-equipped mode, and hybrid mode. In diskless mode, the terminal has no local storage and relies entirely on network I/O. In disk-equipped mode, the terminal has a local SSD (typically 256 GB) partitioned into an EFI partition, a cache partition (30-50 GB), and a user data partition (D: drive). The hybrid mode combines elements of both. A detailed comparison of these three modes is presented in Table 2.

Table 2. Comparison of three working modes.

Feature	Diskless Mode	Disk-equipped Mode	Hybrid Mode
Boot Time	60-120s (full network)	15-30s (local cache)	30-60s
Runtime Perf.	Network-dependent, I/O >10ms	I/O <1ms, <5% diff from PC	I/O <1ms, <5% diff from PC
Network Req.	Sustained >=100 Mbps	Boot >=100 Mbps, run <10 Mbps	10-100 Mbps (elastic)
Offline Cap.	Not supported	Supported	Partial
Hardware Cost	Low	Medium	Medium
Use Case	Stable network labs	General purpose	Unstable network

Among the three modes, disk-equipped mode is the most practical for university computer labs, offering fast boot times, offline capability, and near-native performance. It has been widely adopted across university deployments.

7. Offline and Hot-Switch Mechanism

A key advantage of the disk-equipped VOI architecture is its ability to operate offline and seamlessly switch between online and offline modes. The complete process is illustrated in Fig. 5.

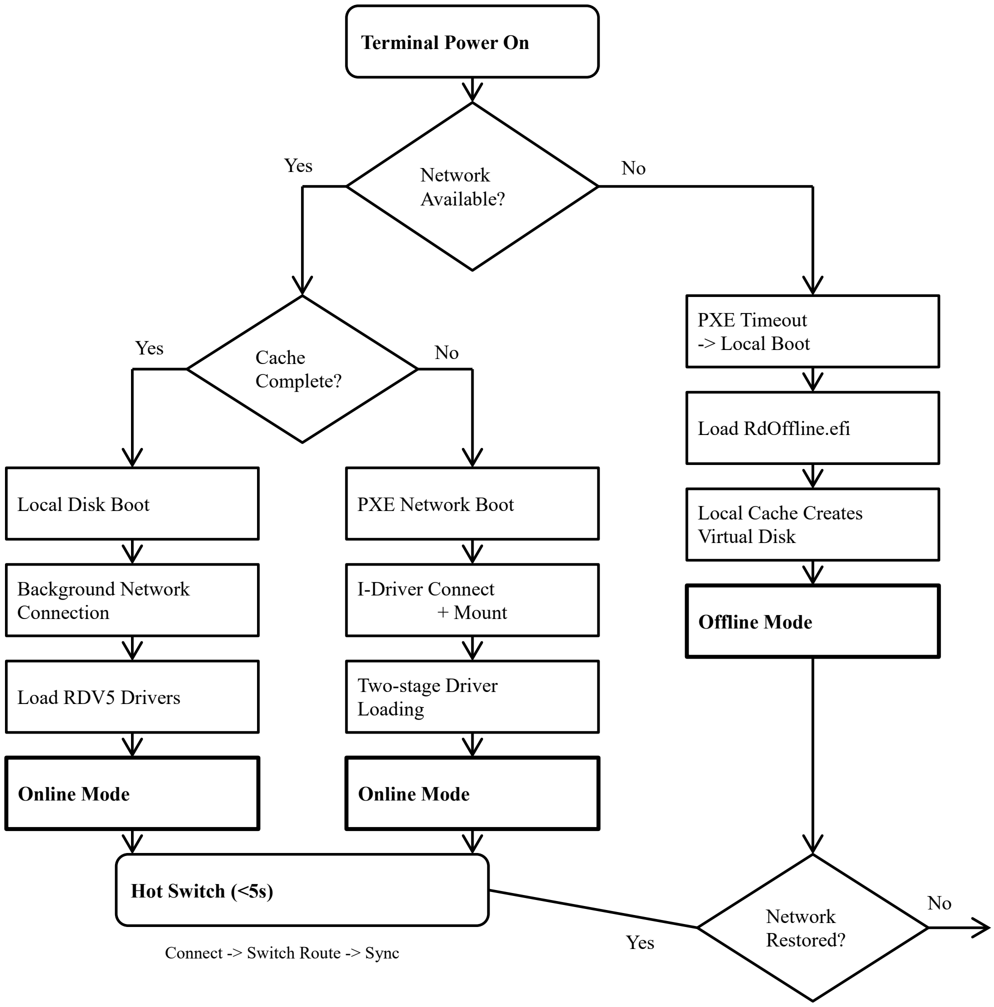

Fig. 5. Working modes and offline hot-switch process.

7.1. Online Boot Path

When the network is available and the local cache is incomplete, the terminal follows the standard PXE boot path: PXE network boot, I-Driver connection establishment, two-stage driver loading, and online operation. When the cache is complete, the terminal boots from the local disk and establishes a background network connection for cache synchronization.

7.2. Offline Boot Path

When the network is unavailable, the PXE process times out and the terminal falls back to local boot. RdOffline.efi is loaded from the local EFI partition, which creates a virtual disk from the local cache and boots into offline mode. This requires approximately 5 GB of cached boot-essential files to function.

7.3. Network Recovery Hot-Switch

When the network is restored during offline operation, the system performs a hot-switch to online mode in less than 5 seconds. The process involves establishing a background network connection, atomically switching the mode flag, and performing incremental synchronization. The transition is transparent to the user. Small updates (<500 MB) are handled automatically, while large updates are recommended to be applied upon reboot.

8. University Lab Deployment Practice

8.1. Deployment Scheme

A typical deployment for 100 terminals consists of one RDV5 server storing multiple system images (e.g., Windows 10 Education with Office, Visual Studio, Python, MATLAB, and AutoCAD). Each terminal is equipped with a 256 GB SSD in disk-equipped mode, connected via Gigabit Ethernet. The server manages image distribution, terminal registration, and policy configuration.

8.2. User Data Management

The system employs a temporary profile strategy for user data management. User profiles are created as temporary profiles that are automatically deleted upon logout, preventing data accumulation on the system drive. For persistent student work, folder redirection to the local D: drive is recommended. The recommended file storage arrangement is summarized in Table 3.

Table 3. File storage strategy for university computer labs.

Storage Location	Content	Persistence	Permission
C: (Shared Image)	System files, pre-installed software	Restored on reboot	Student read-only
%USERPROFILE% (Temp Profile)	Temporary profile	Deleted on logout	Student write (temp)
D: (Local Disk)	Student work, personal data	Persistent	Student writable

8.3. Deployment Effectiveness

The deployment demonstrates several key benefits: (1) Unified management: a single administrator can manage thousands of terminals; (2) Rapid deployment: PXE combined with P2P distribution enables fast image rollout; (3) Automatic restoration: differential disk and COW mechanisms ensure a clean state on each reboot; (4) Data security: system data is centralized on the server with temporary profiles preventing data leakage; (5) Performance assurance: VOI local computing meets the demands of CAD, video editing, and programming courses; (6) Offline availability: disk-equipped mode ensures continued operation during network outages.

9. Conclusion

This paper has presented a systematic analysis of the key technologies in the RDV5 cloud desktop system through five architectural diagrams and process flows: (1) the three-layer VOI system architecture; (2) the network boot and two-stage driver loading process; (3) the master-disk/differential-disk copy-on-write mechanism; (4) the multi-level cache and Minifilter driver architecture; and (5) the working modes and offline hot-switch process.

The VOI architecture, with its convergence of disk and network technologies, provides a highly efficient and reliable solution for university computer lab management. The system has been deployed at scale across multiple universities, demonstrating its practical value in unified management, rapid deployment, automatic restoration, and data security. As the trend toward VOI, VDI, and IDV convergence continues, such systems will play an increasingly important role in higher education IT infrastructure.

References

[1] C. Liu, H. Xie, Z. Shi, "A maintenance scheme for public computer laboratories based on disk-network dual standby," China University Teaching, 2015.

[2] T. Chen, "Design and application of laboratory teaching environments based on cloud desktop convergence mode," Frontiers of Modern Education, vol. 5, no. 5, 2024.

[3] A. He, Z. Zhou, T. Wang, "Construction scheme for university computer labs based on cloud desktop technology," Digital Technology and Application, vol. 40, no. 6, pp. 161-163, 2022.

[4] X. Guo, "Research on cloud desktop-based computer lab management mode," Software, 2022.

[5] J. Huang, "Research on the application of cloud desktop technology in university computer laboratory management," Information and Computer, 2019.

[6] A. Mitra, A. Ghosh, S. C. Sethuraman, D. P. V S, "Colaboot: A Cloud-based Diskless PC Booting Mechanism," arXiv preprint arXiv:2408.17045, 2024.

[7] E. D. Lazowska, J. Zahorjan, D. R. Cheriton, W. Zwaenepoel, "File Access Performance of Diskless Workstations," ACM Transactions on Computer Systems, vol. 4, no. 3, pp. 238-268, 1986.

[8] A. Mitra, H. Haroon, A. R. Shah, M. E. Rasooli, B. I. D. Nikolaev, T. Balli, "/dev/SDB: Software Defined Boot -- A novel standard for diskless booting anywhere and everywhere," arXiv preprint arXiv:2601.20629, 2026.

[9] X. He, Q. Yang, M. Zhang, "A Caching Strategy to Improve iSCSI Performance," in Proc. IEEE LCN, 2002.

[10] IETF, "RFC 7143: Internet Small Computer System Interface (iSCSI) Protocol (Consolidated)," 2014.

[11] W. Xiao, Y. Liu, Q. Yang, J. Ren, C. Xie, "Implementation and Performance Evaluation of Two Snapshot Methods on iSCSI Target Storages," in Proc. IEEE MSST, 2006.

[12] D. T. Meyer, G. Aggarwal, B. Cully, G. Lefebvre, M. J. Feeley, N. C. Hutchinson, A. Warfield, "Parallax: Virtual Disks for Virtual Machines," in Proc. EuroSys, 2008.

[13] N. Xiao, B. Chen, Z. Cai, "A Demand-driven Virtual Disk Prefetch Mechanism for Seamless Mobility of Personal Computing Environments," The Journal of Supercomputing, vol. 63, no. 1, pp. 150-170, 2013.

[14] Microsoft, "Filter Manager Concepts," Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/windows-hardware/drivers/ifs/filter-manager-concepts.