A Performance Analysis of System Rollback Techniques

Antônio Drumond Cota de Sousa, Henrique Cota de Freitas

Computer Architecture and Parallel Processing Team (CArT)
Department of Computer Science – Pontifícia Universidade Católica de Minas Gerais
30.535-901 – Belo Horizonte – MG – Brazil

adcsousa@sga.pucminas.br, cota@pucminas.br

Abstract. *System restorations are crucial for maintaining operational continuity, yet the lack of comparative performance data makes selecting the right rollback tool a significant challenge. This paper addresses this gap by conducting a quantitative performance analysis of four popular techniques: NixOS Generations, Snapper, Timeshift (in RSYNC mode), and Windows System Restore. We evaluated each tool based on two key metrics: rollback duration and snapshot storage size, using a controlled virtualized environment. Our results reveal significant performance disparities, with rollback speeds differing by up to a factor of four, indicating that filesystem-integrated, copy-on-write tools are demonstrably more efficient for performance-critical applications.*

1. Introduction

A system restoration, or rollback, can be defined as returning a system to a previously recorded state, called a restore point or snapshot (Yun et al., 2008). This is useful for maintaining system availability and operation continuity, providing a reliable way to quickly recover from software failures, human error, or other disruptive events. For critical systems where uptime is imperative, the ability to revert to a known-working state is invaluable.

A wide range of system rollback tools exists, each with distinct performance characteristics and specific operating system or filesystem dependencies. Consequently, selecting the appropriate tool is a critical upfront decision, as migrating a production environment to accommodate a different tool later can be costly and impractical.

A review of the existing literature reveals a significant gap: there is a lack of comparative performance data on system rollback tools. This lack of quantitative data makes informed decision-making difficult. Therefore, the primary objective of this paper is to address this problem by conducting a comparative analysis of four distinct rollback techniques. This paper benchmarks the restoration time and storage overhead of four popular and architecturally diverse solutions: NixOS System Generations (Dolstra and Löh, 2008), Timeshift (RSYNC mode) (Linux Mint, 2025), Snapper (openSUSE, 2025), and Windows System Restore (Microsoft, 2025).

This research makes two primary contributions. First, it provides a quantitative result comparing the restoration time and storage overhead of key rollback tools, revealing

Antônio Sousa is an undergraduate student in Computer Science at PUC Minas.

The authors would like to thank the National Council for Scientific and Technological Development of Brazil (CNPq - Codes 311697/2022-4 and 402837/2024-0) and PUC Minas FIP 2024/30947.

performance differences of up to four times between them. Second, it presents a detailed analysis of the architectural trade-offs and requirements between these approaches, producing a set of practical guidelines to help engineers and system administrators select the most suitable technology for their specific operational requirements.

This paper is organized as follows. Section 2 presents related works. Section 3 provides the necessary background. Section 4 details the methodology. Section 5 discusses the experiments and analyzes the results. Finally, Section 6 concludes the paper.

2. Related Work

Aiming to improve the reliability and fault tolerance of mobile devices, Lin et al. (2001) introduced a novel rollback recovery system for PalmOS, a dominant mobile operating system at the time of publication. Although the platform is now outdated, the lightweight checkpointing mechanisms described by the authors share foundational similarities with modern tools, such as Windows System Restore. The work's primary contribution was demonstrating the feasibility of a low-overhead recovery system in a resource-constrained environment.

In the context of data center environments, where Virtual Machine Clusters (VMCs) are common, system failures can significantly impact operational continuity. To address this, Cui et al. (2014) developed HotRestore, a restoration tool capable of rolling back a VMC to a previously recorded state in mere seconds. The authors' work demonstrates the role of rapid system restoration in reducing the operational overhead caused by such failures in large-scale virtualized environments.

Addressing the performance overhead of data protection features in most open source storage systems, Yang et al. (2019) proposed S3R5, a storage system architecture based on the Redirect on Write (RoW) principle. The authors compared the snapshot restoration performance of S3R5 with that of Ceph, a widely used Copy-on-Write (CoW) based storage platform. Their results demonstrated that RoW implementation in S3R5 reduced the rollback time by up to 44.40%.

In summary, the discussed articles highlight the relevance of System Rollback Tools in various critical computational environments. Despite this established importance, a direct analysis of modern rollback tools and their underlying architectural trade-offs is largely absent from existing research. Our work addresses this gap by providing a comparative performance analysis of such tools. We evaluated restoration time and disk usage by snapshots, while also examining the architectural principles and practical requirements of each approach.

3. Background

System restore, or rollback, is a process which allows the restoration of critical operating system and application files during a crash event (Yun et al., 2008). Four system rollback tools that employ different techniques will be compared in this article, and each of them will be explained in the following sections

3.1. NixOS system generations

NixOS is a Linux distribution configured and managed entirely declaratively, in contrast to the imperative approach used by most other systems (Dolstra and Löh, 2008). To allow

users to easily recover from upgrades or mistakes that render the system non-bootable or unstable, the system creates system profiles, called **generations** whenever changes are made to the system. Each generation is an immutable snapshot of the system.

This process is initiated when a user runs the command `nixos-rebuild switch`. This command evaluates and builds the system configuration present in `/etc/nixos/configuration.nix` and stores the result in `/nix/var/nix/profiles/system/`. Then, an activation script is run, updating the current system environment into the new profile. Each system profile is linked to a numbered generation in the bootloader menu, which can be selected at startup. This enables users to rollback the system into any non-garbage-collected previous generation.

3.2. Snapper + BTRFS

BTRFS (B-Tree File System) is a Linux filesystem that utilizes a Copy on Write (CoW) mechanism. This feature is leveraged by tools like Snapper to create and restore highly space efficient system snapshots (openSUSE, 2025).

The CoW principle dictates that when data is modified, changes are written to a new block, rather than overwriting the original location. BTRFS implements this using a CoW-friendly B-Tree data structure (Rodeh et al., 2013). Whenever a node in the tree is changed, the path between that node and the tree's root is duplicated, and modifications are applied only to the copied nodes (Rodeh, 2008). This is very effective for creating snapshots, as it only requires creating a new root pointer to the existing unmodified tree. Subsequent changes to system files will cause new blocks to be written, diverging from the snapshot without altering its original blocks, as shown in Figure 1. Restoring the system from a snapshot is similarly straightforward, as it simply involves making the snapshot's root the new default filesystem root.

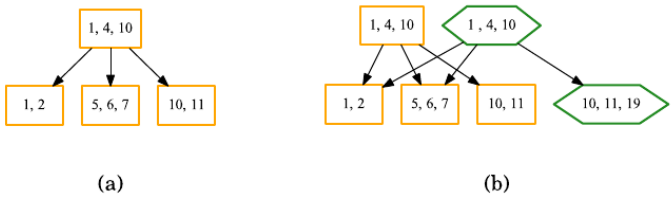

Figure 1. (a) A basic B-tree and (b) Inserting key 19 (Rodeh et al., 2013).

3.3. Timeshift + RSYNC

Timeshift is a system restoration tool for Linux. In its rsync mode, which can be used on any Linux distribution or filesystem, snapshots are created by making copies of system files. While the initial snapshot is a full copy of all files, subsequent ones reduce disk space usage by using hard links for any unchanged files from previous snapshots. Timeshift utilizes the RSYNC utility to reconstruct the copied files in their original locations (Linux Mint, 2025).

The rsync algorithm, on which the utility is based, was designed to efficiently synchronize files, originally between computers on a network (Tridgell, 2025). The algorithm works by dividing the destination file into a series of non-overlapping fixed size blocks. It calculates a weak 32-bit checksum and a strong 128-bit MD4 checksum for each of those

blocks. It then searches through the origin file to find all blocks of data that have the same weak and strong checksum as one of the blocks in the destination file. Finally, it builds a copy of the origin file at the destination by using a sequence of instructions, where each instruction is either literal data that is missing from the destination file, or a reference to one of its existing blocks (Tridgell and Mackerras, 1996).

3.4. Windows System Restore

Volume Shadowcopy Service (VSS) is a framework in Microsoft Windows that is used by applications to create point-in-time snapshots, known as shadow copies. This service is the underlying technology for many Windows features, including the native Windows System Restore tool. VSS coordinates three key components:

1. **Requester:** Applications that initiate the snapshot process (e.g. the System Restore utility).
2. **Writers:** Components that collect data and ensure data consistency by preparing their associated application for the snapshot, by flushing caches or freezing I/O operations.
3. **Providers:** Software or hardware components responsible for creating, maintaining and restoring shadow copies themselves. Windows includes a default software provider that uses a Copy on Write (CoW) mechanism.

To create a snapshot, a requester asks VSS to signal the writers, which prepare their data. The provider then creates the shadow copy containing the state of the selected volume. When restoring a system, the CoW provider uses the stored data to revert the volume to the state captured in the snapshot (Microsoft, 2025). The interaction between VSS and its Writers, Requesters, and Providers is shown in Figure 2.

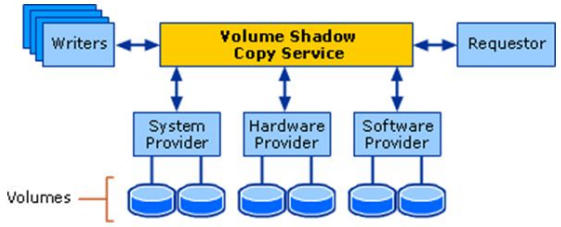

Figure 2. Coordination of Writers, Requestors and Providers (Microsoft, 2025).

4. Methodology

4.1. Computational Environment

All experiments were conducted on a single machine, which contained an Intel Core i7-1165G7 processor with 4 physical cores and 8 threads, 16GB of DDR4 memory, and 500GB of SSD storage. The operating system used was Linux NixOS 24.11 (Vicuna). For the virtual machines, which did not run simultaneously, each was configured with 2 CPU cores, 4GB of RAM, and a 64GB virtual disk.

4.2. Procedures

This study adopted an experimental approach to measure and compare the performance of different system rollback tools. In addition to the previously mentioned tools, the following software was used to conduct the experiments in a controlled environment:

- `Quickemu`: A front end for `Qemu` (A popular type 2 hypervisor), used to create and manage virtual machines for testing.
- `vmstat`: A Linux utility used to monitor system resource usage. It was employed to detect system-level anomalies (e.g., high CPU *steal time*) that could interfere with test validity.
- `BTRFS Assistant`: A graphical interface for `Snapper`, used to streamline the creation of BTRFS snapshots.

The experiments consisted of the following: `Quickemu` was used to create four virtual machines, one with NixOS (Linux), two with Fedora (Linux), and one with Windows 10. On these virtual machines, the respective restoration tools were configured: NixOS with its System generations, one Fedora VM with `Snapper`, the second Fedora VM with `Timeshift`, and Windows 10 with its native `System Restore`. For each configuration, 10 system rollbacks were executed. During each rollback operation, `vmstat` was running on the host system to monitor resource usage and ensure no system-level anomalies interfered with results. Figure 3 provides a test diagram.

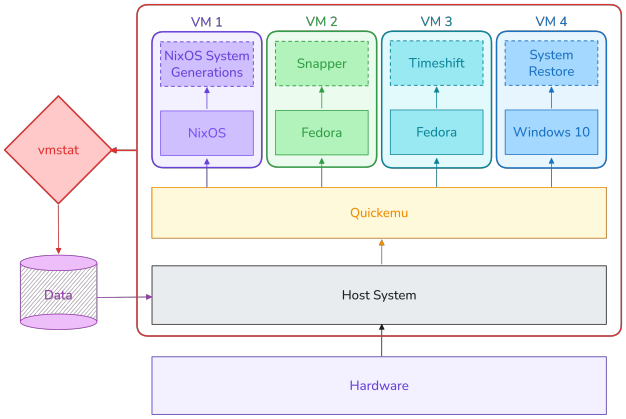

Figure 3. Test environment diagram.

To measure the disk space occupied by snapshots, a different technique was required for each restoration tool. For NixOS and Timeshift, the space used by each snapshot was determined by directly measuring the size of its corresponding storage directory. For Snapper, the `snapper list` command was used, which provides the precise disk space for each snapshot. A different approach was necessary for Windows System Restore, as it does not provide a direct method to measure individual snapshot sizes. Instead, it only reports the total space consumed by all snapshots. Therefore, the `vssadmin list shadowstorage` command was executed after each snapshot was created. The size of each individual snapshot was then calculated by measuring the increase in the total space consumed.

5. Experiments and Results

5.1. Restoration Time

Figure 4 illustrates the rollback durations for each tool, highlighting a significant performance gap between them. NixOS generations and Snapper were the fastest, exhibiting

low median rollback times of approximately 20 and 26 seconds, respectively. Timeshift occupied the middle of the spectrum, with a median duration of 54 seconds. In contrast, Windows System Restore was the slowest and least consistent tool, showing a much higher median time of approximately 88 seconds and the widest performance variance.

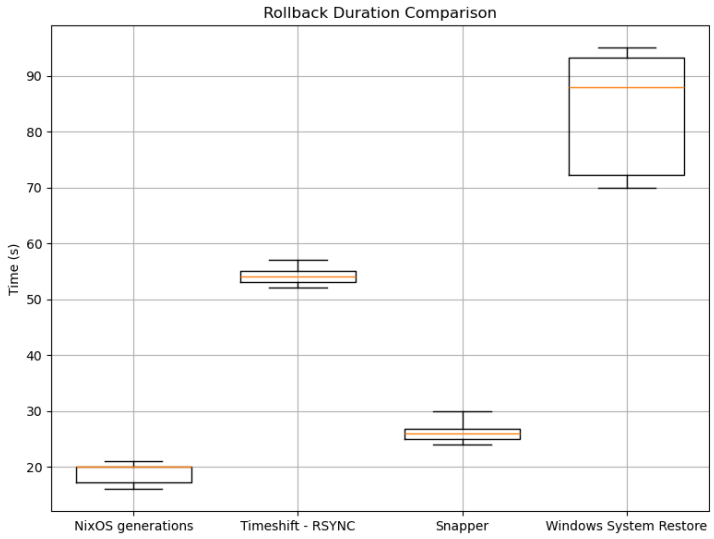

Figure 4. Time comparison of different system restoration techniques.

NixOS and Snapper demonstrated the highest efficiency. The rollback mechanism for both is primarily a metadata operation. In the case of Snapper, this involves re-pointing the filesystem's root to a different BTRFS subvolume, an almost instantaneous change. For NixOS, a rollback is functionally identical to a standard reboot into a different, pre-existing system configuration, where the system environment is constructed by the activation script with symlinks to the Nix Store. Consequently, for both tools, the measured rollback time is dominated by the system's reboot sequence rather than the restoration operation itself.

In contrast, Timeshift's performance is limited by its file-based approach. A restoration requires the physical transfer of data from the snapshot location to the live filesystem. While the RSYNC algorithm minimizes the amount of blocks transferred, the disk I/O operation is inherently more time and resource intensive than the metadata-only changes performed by Snapper and NixOS.

Finally, the lower relative performance of Windows System Restore can be attributed to the architectural overhead of its Volume Shadow Copy Service (VSS) framework. The restoration process requires complex coordination between multiple software layers (the requester, writers, and provider), each of which adds latency. This intricate orchestration makes the process fundamentally slower compared to the more direct methods used by the other tools.

5.2. Snapshot Size

Table 1 exhibits the average space taken by each individual snapshot for each of the tools.

Timeshift recorded the largest average snapshot size at 11.32 GB. This is because its file-based approach creates full copies of files, without compression. This large footprint, however, is alleviated by not creating copies of unchanged files. In consequence, after the initial snapshot is created, subsequent ones will occuppy less disk space.

Timeshift's file-based approach yielded the largest average snapshot size, at 11.32 GB. While its initial snapshot is a full, uncompressed copy of all relevant files, the storage footprint of subsequent snapshots is reduced by using hard links, rather than making duplicate copies.

In contrast, Snapper and Windows System Restore demonstrated more space-efficient results (13.92 MB and 30 MB, respectively). Both tools leverage their underlying filesystem's Copy on Write (CoW) capabilities. This means snapshot only stores altered disk blocks between system states, rather than entire files.

NixOS had the lowest footprint, with each generation averaging only 80 kB. This low size is because NixOS generations do not contain programs or dependencies themselves, but rather directions to packages located in the system's Nix Store. As packages can be shared across all generations, the snapshot size reflects only the changes in configuration scripts and symlinks needed to construct the system environment.

Table 1. Average disk space occupied by system snapshot per restoration tool.

	NixOS System Generations	Timeshift	Snapper	Windows System Restore
Average space occupied by Snapshot	80 kB	11.32 GB	13.92 MB	30 MB

6. Conclusion

The primary objective of this study was to analyze popular system rollback tools. Our experiments revealed significant disparities in both rollback duration and snapshot size. The results consistently showed that metadata-based approaches, such as NixOS generations and Snapper, offered the fastest restoration times and the lowest storage overhead. In contrast, the file-based approach employed by Timeshift's rsync mode resulted in high disk usage and moderately slower restorations. Lastly, Windows System Restore was the least efficient in restoration performance.

These findings are significant for system administrators and DevOps engineers. For performance-critical systems, our results indicate that modern copy-on-write tools, such as Snapper, or recovery-focused operating systems, like NixOS, are the most effective strategies. While Timeshift offers greater filesystem flexibility, it comes at a clear performance cost. The data suggests that Windows System Restore may be unsuitable for enterprise environments that require swift and predictable recovery.

It is important to note the limitations of this study. The experiments were conducted on virtual machines with a single hardware configuration and an NVMe SSD. Performance characteristics, especially for I/O-heavy tools such as Timeshift, may differ on other storage media such as HDDs. Furthermore, this analysis was limited to four representative tools; other solutions, such as native ZFS snapshots, were not included. Future

work could evaluate the resilience of these restoration tools in critical failure scenarios, such as filesystem corruption, interrupted snapshots, or partial disk failure.

References

L. Cui, J. Li, T. Wo, B. Li, R. Yang, Y. Cao, and J. Huai. HotRestore: A fast restore system for virtual machine cluster. In *28th Large Installation System Administration Conference (LISA14)*, pages 10–25, Seattle, WA, Nov. 2014. USENIX Association. ISBN 978-1-931971-17-1. URL https://www.usenix.org/conference/lisa14/conference-program/presentation/cui.

E. Dolstra and A. Löh. Nixos: a purely functional linux distribution. *SIGPLAN Not.*, 43(9):367–378, Sept. 2008. ISSN 0362-1340. doi: 10.1145/1411203.1411255. URL https://doi.org/10.1145/1411203.1411255.

C.-Y. Lin, S.-Y. Kuo, and Y. Huang. A checkpointing tool for palm operating system. In *2001 International Conference on Dependable Systems and Networks*, pages 71–76, 2001. doi: 10.1109/DSN.2001.941392.

Linux Mint. Timeshift: A system restore tool for Linux. https://github.com/linuxmint/timeshift, 2025. Accessed: 2025-07-01.

Microsoft. Microsoft learn - volume shadow copy service (vss). https://learn.microsoft.com/en-us/windows-server/storage/file-server/volume-shadow-copy-service, 2025. Accessed: 2025-06-06.

openSUSE. snapper: A tool for filesystem snapshot management. https://github.com/openSUSE/snapper, 2025. Accessed: 2025-07-01.

O. Rodeh. B-trees, shadowing, and clones. *TOS*, 3, 02 2008. doi: 10.1145/1326542.1326544.

O. Rodeh, J. Bacik, and C. Mason. Btrfs: The linux b-tree filesystem. *ACM Trans. Storage*, 9(3), Aug. 2013. ISSN 1553-3077. doi: 10.1145/2501620.2501623. URL https://doi.org/10.1145/2501620.2501623.

The Linux man-pages project. vmstat(8) - Report virtual memory statistics. https://linux.die.net/man/8/vmstat, 2025. Accessed: 2025-07-06.

A. Tridgell. rsync. https://rsync.samba.org/, 2025. Accessed: 2025-06-10.

A. Tridgell and P. Mackerras. The rsync algorithm. Technical report, The Australian National University, 1996. URL http://hdl.handle.net/1885/40765. Accessed: 2025-06-10.

H. Yang, Y. Yang, and Y. Tu. S3r5: A snapshot storage system based on row with rapid rollback, recovery and read-write. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 2111–2118, 2019. doi: 10.1109/HPCC/SmartCity/DSS.2019.00292.

S.-M. Yun, A. Savoldi, P. Gubian, Y. Kim, S. Lee, and S. Lee. Design and implementation of a tool for system restore point analysis. In *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 542–546, 2008. doi: 10.1109/IIH-MSP.2008.256.