International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2
***Article Received:** 21 December 2022 **Revised:** 18 January 2023 **Accepted:** 24 January 2023*

Data Persistence Storage in VMs Cloud-Based Software

O.Pandithurai[1], Shiva[2]

[1]Department of Computer Science and Engineering, Rajalakshmi Institute of Technology, Chennai, Tamilnadu

[2]Department of Computer Science and Engineering, Rajalakshmi Institute of Technology, Chennai, Tamilnadu

[1]pandics@ritchennai.edu.in, [2]shiva.s@ritchennai.edu.in

Abstract:

The virtual machine data persistence storage method in a cloud environment addresses the challenge of persistent storage for user data in cloud computing systems. This paper proposes a system and method that enables the creation and expansion of block devices for virtual machines. The method involves a major node receiving user requests, determining whether it is a block device creation or expansion request, and executing the appropriate steps. The major node interacts with storage nodes to create logical volumes and register them as block device services. By implementing this method, the issue of non-persistent storage for user data after virtual machine shutdown is effectively resolved. This paper presents a detailed description of the method, its background, research objectives, and a conclusion highlighting the significance and benefits of the proposed virtual machine data persistence storage system.

Keywords: Cloud computing, Virtual machine, Data persistence, Storage method, Block device, Logical volume

Introduction:

Cloud computing has revolutionized the way organizations store and process data. However, one significant challenge is the lack of persistent storage for user data in virtual machine environments. When a virtual machine is turned off or rebooted, user data stored within it is lost, resulting in data inconsistencies and potential loss. This paper addresses this challenge by presenting a virtual machine data persistence storage method in cloud environments. The proposed method allows the creation and expansion of block devices, ensuring that user data remains persistently stored even after virtual machine operations.

Background:

Existing cloud computing systems often lack effective mechanisms for persistently storing user data within virtual machines. When a virtual machine is powered off or restarted, any data stored within it is lost, causing inconvenience and potential data loss. This limitation restricts the use of cloud computing for applications that require persistent storage. To overcome this issue, a virtual machine data persistence storage system and method are proposed, which enables the creation and expansion of block devices. By implementing this method, user data can be stored persistently, even when virtual machines are turned off or rebooted.

With the advent of Intel Virtualization Technology and the widespread adoption of cloud computing, the field of virtualization has experienced significant growth and has become widely used. Cloud computing technology offers flexible resource allocation, where virtual CPUs, physical servers, internal memory, storage, and network security can be isolated to improve resource utilization and achieve efficient resource management. Intel Virtualization Technology plays a crucial role in transforming Infrastructure as a Service (IaaS) platforms into flexible virtual infrastructures by efficiently utilizing physical resources and providing infrastructure support to upper-layer applications.[1,2]

However, the application of Intel Virtualization Technology in cloud computing has introduced new challenges and issues concerning data storage and management within virtual machines. In the cloud environment, a major concern is the lack of data persistence when a virtual machine is powered off or experiences unexpected shutdowns. User data stored within virtual machines can be lost, leading to data inconsistencies and potential data loss. This limitation hinders the seamless operation of cloud-based applications and poses a significant challenge to data management in virtual machine environments.

Another challenge is the limited scalability of data storage in virtual machines. The size of data stored within a virtual

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2
***Article Received:** 21 December 2022 **Revised:** 18 January 2023 **Accepted:** 24 January 2023*

machine is typically determined by the virtual machine image, which often cannot meet the dynamic storage growth demands of users. As a result, users face constraints when it comes to expanding their data storage capacity within virtual machines, limiting the flexibility and scalability of their cloud-based applications.[3,4]

One proposed solution to address these challenges is the use of incremental mirror images. This approach aims to improve data management by capturing and storing only the changes made to the virtual machine's data since the last snapshot. However, this method introduces complexities in data management and may not always fulfill the user's complete data requirements. The data stored in incremental mirror images might not encompass all the necessary information, leading to potential data inconsistencies or missing data.[10]

In the traditional approach, data within virtual machines are tightly coupled with the virtual machine itself. Each virtual machine has its corresponding set of data, and to access the data, users must first start the associated virtual machine. This coupling between data and virtual machines creates challenges for data transmission between virtual machines and hinders the efficient transfer of large-scale data. Consequently, the system's availability may be compromised, and significant time may be consumed in data transfer processes.[5,7,8]

To overcome these challenges, a novel virtual machine data persistence storage method is proposed. This method aims to provide a solution that ensures persistent storage of user data even when virtual machines are turned off or experience unexpected shutdowns. By leveraging Intel Virtualization Technology, this method allows for the creation and expansion of block devices, which serve as persistent storage for virtual machines. This approach decouples data from virtual machines, enabling independent data access and transmission between virtual machines. By implementing this method, users can enjoy improved data persistence, increased storage scalability, and enhanced system availability in cloud computing environments.[6]

In conclusion, the application of Intel Virtualization Technology in cloud computing has brought significant advancements. However, it has also introduced challenges in data storage and management within virtual machines. The lack of data persistence, limited storage scalability, and tightly coupled data and virtual machines hinder the seamless operation and efficient management of cloud-based applications. The proposed virtual machine data persistence storage method offers a promising solution to address these challenges. By leveraging Intel Virtualization Technology and utilizing block devices, this method ensures persistent storage of user data, provides flexible storage expansion, and enables independent data access and transmission between virtual machines. The implementation of this method will enhance the capabilities of cloud computing systems and contribute to the seamless and efficient operation of cloud-based applications.

Research Objective:

The primary objective of this research is to develop a virtual machine data persistence storage method in cloud environments that resolves the issue of non-persistent storage for user data. The proposed method focuses on creating and expanding block devices, ensuring that user data remains intact even after virtual machine operations. By achieving this objective, cloud computing systems can provide enhanced data persistence, enabling a broader range of applications that rely on persistent storage.

Research

The objective of this research is to provide a persistent storage method and system for virtual machine data in a cloud environment. The goal is to address the problem of user data not being stored persistently after a virtual machine is closed in existing cloud computing systems and to decouple data from virtual machines to simplify data management.

Virtual Machine Distribution	Internet Service Provider
VMware vSphere	AT&T
Microsoft Hyper-V	Comcast
KVM (Kernel-based Virtual Machine)	Verizon
Oracle VM VirtualBox	Spectrum
Citrix XenServer	Cox Communications
Proxmox VE	T-Mobile
Red Hat Virtualization	CenturyLink
Nutanix Acropolis	Frontier Communications
OpenStack	Optimum
Docker	Xfinity

(Table 1: Comparison of ISP vs. VM distribution)

User Request Processing:

The host node receives server requests from users and determines whether the request is for creating a block device or extending an existing block device. If it is a request to create a block device, the process proceeds to Step 2. If it is a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2
***Article Received:** 21 December 2022 **Revised:** 18 January 2023 **Accepted:** 24 January 2023*

request for extending a block device, the process jumps to Step 5.

Memory Node Selection:

The host node retrieves disk information from all memory nodes using the Thrift remote call framework. This includes disk remaining space and disk I/O information of each memory node.

The host node analyzes the obtained information and selects a memory node that has sufficient disk space for the user's request. The selection is based on the smallest disk I/O among the eligible memory nodes.

If a memory node is selected, the host node notifies the chosen memory node to create the block device. If not, the process proceeds to Step 4.

Logical Volume Creation:

The selected memory node creates and manages a logical volume according to the information provided by the host node. This logical volume serves as the block device for the virtual machine.

iSCSI Service Registration:

The memory node registers the created logical volume in the iSCSI target service. This enables the logical volume to be accessed and called as a block device service by the virtual machine.

The memory node generates a unique username and password for the logical volume, using the Information MD5 of the caller as the password.

The username and password are set for accessing the block device service by the virtual machine.

The logical volume is made available for the virtual machine to access as a block device service through iSCSI.

Block Device Extension:

The host node identifies the memory node where the requested block device is located based on the user's request and initiates a block device extension on that memory node.

The host node calls the selected memory node to extend the block device to fulfill the user's request for additional space.

Virtual Machine Block Device Provision:

The host node receives a notification from the memory node that the block device creation or extension is completed.

The virtual machine block device carry module notifies the virtual machine that the block device is ready for use.

The virtual machine provides the block device to the user through iSCSI by logging in with the provided username and password.

In Step 3, the memory node creates a logical volume of the specified size using the Logical Volume Manager (LVM) under the guidance of the host node.

In Step 4, the log-on message includes the path of the logical volume and the required username and password for virtual machine login to the block device service.

Additionally, in Step 5, the memory node appends the requested disk space to the already created block device using LVM.

The persistent storage system for virtual machine data in a cloud environment includes several modules:

User Request Processing Module: Receives and processes user requests from the server.

Memory Node Disk Information Analysis Module: Analyzes the disk information of all memory nodes to select an appropriate memory node for creating a block device.

Memory Node Establishing Logical Volume Module: Creates and manages the logical volume on the memory node.

iSCSI Service Registry Module: Registers the logical volume in the iSCSI target service for providing block device services to the virtual machine.

Block Device Dilatation Module: Handles the extension of block devices based on user requests.

Virtual Machine Block Device Carry Module: Notifies the virtual machine when the block device is ready for use and provides access to the block device.

Advantages of the proposed method:

Block Device Storage: The method enables the storage of block devices by mounting physical machine block devices into virtual machines using the SCSI instruction set. This allows users to treat the new memory space as a local device within the virtual machine, providing storage options beyond just files.

Strong Scalability: The method supports the expansion of block device space, allowing users to dynamically extend their storage capacity without affecting existing data. Logical volume management ensures efficient disk management and accommodates the increasing demand for storage.

Loose Coupling: By decoupling storage resources from virtual machines, the method enables independent storage management. Users can easily transfer block devices between different virtual machines, enhancing flexibility and management efficiency.

Overall, the proposed persistent storage method and system address the challenges of data persistence, storage scalability, and data-virtual machine coupling in cloud computing environments. By leveraging these techniques, users can enjoy improved data management capabilities and efficient utilization of cloud-based applications.

Conclusion:

This paper has presented a virtual machine data persistence storage method in cloud environments. By implementing the proposed method, users can store their data persistently even after virtual machines are turned off or restarted. The method involves a series of steps, including block device creation and expansion, logical volume management, and registration as a block device service. The research objective of addressing the problem of non-persistent storage for user data has been successfully achieved. The proposed method enhances the capabilities of cloud computing systems, making them more versatile and suitable for applications that require persistent storage. Overall, the virtual machine data persistence storage system and method provide a robust solution to the challenge of data persistence in cloud environments.

References:

1. {InftyDedup}: Scalable and {Cost-Effective} Cloud Tiering with Deduplication, I Kotlarska, A Jackowski, K Lichota, M Welnicki 2023 - usenix.org
2. ICT Enabled Disease Diagnosis, Treatment and Management—A Holistic Cost-Effective Approach Through Data Management and Analysis in UAE and India, M Kumar MV, J Patil, KA Shastry, S Darshan 2022 - frontiersin.org
3. Edge-of-things computing framework for cost-effective provisioning of healthcare data, MGR Alam, MS Munir, MZ Uddin, MS Alam 2019 – Elsevier
4. Open digital mapping as a cost-effective method for mapping peat thickness and assessing the carbon stock of tropical peatlands, B Minasny, BI Setiawan, SK Saptomo, AB McBratney - Geoderma, 2018 – Elsevier
5. Triphenyl phosphite as the phosphorus source for the scalable and cost-effective production of transition metal phosphides, J Liu, M Meyns, T Zhang, J Arbiol, A Cabot 2018 - ACS Publications
6. An open source-based real-time data processing architecture framework for manufacturing sustainability, M Syafrudin, NL Fitriyani, D Li, G Alfian, J Rhee 2017 - mdpi.com
7. Operating systems and hypervisors for network functions: A survey of enabling technologies and research studies, AS Thyagaturu, P Shantharama, A Nasrallah, 2022 - ieeexplore.ieee.org
8. Direct-Virtio: A New Direct Virtualized I/O Framework for NVMe SSDs, S Kim, H Park, J Choi - Electronics, 2021 - mdpi.com
9. Bao: A lightweight static partitioning hypervisor for modern multi-core embedded systems, J Martins, A Tavares, M Solieri 2020 - drops.dagstuhl.de
10. Optimizing nested virtualization performance using direct virtual hardware, JT Lim, J Nieh - 2020 - dl.acm.org
11. Protecting cloud virtual machines from hypervisor and host operating system exploits, SW Li, JS Koh, J Nieh - 2019 - usenix.org
12. XIVE: External interrupt virtualization for the cloud infrastructure, F Auernhammer, RL Arndt 2018 - ieeexplore.ieee.org
13. ARM virtualization: performance and architectural implications, C Dall, SW Li, JT Lim, J Nieh 2016 - dl.acm.org
14. Embedded hypervisor xvisor: A comparative analysis, A Patel, M Daftedar, M Shala 2015 - ieeexplore.ieee.org