Index Based Deduplication File System in Cloud Storage System

S.Sasikala, V.Geetha

***Abstract*— In a virtualized cloud situation each occurrence of a guest operating system turns on a virtual machine, get into virtual hard disks denoted as virtual disk image files in the host operating system. Because these image files are deposited as regular files from the external point of view, assisting up VM's data is mainly done by taking snaps of virtual disk images. In a virtualized cloud computing environment, recurrent snapshot backup of virtual disks advances hosting reliability but storage request of the operations is massive. Though dirtybitbased technique can identify unmodified data between versions, full deduplication with fingerprint evaluation can remove more redundant content at the cost of computing resources. So in this project we proposed a LIQUID framework to introduce the deduplication scheme in both file and block level. And propose a deduplication file system with low storing consumption and high-performance IO, which fulfills the necessities of VM hosting. Finally we spread our work to address the problem of authorized data deduplication. And this project we can implement Distributed Hash Table techniques in data server and also comprise trust based scheme between P2P data sharing file system.**

***Index Terms*— DeDuplication, LIQUID Framework, Virtual Machine.**

I. INTRODUCTION

Cloud computing is the delivery of calculating as a facility somewhat other than a product, whereby the shared resources and information are providing to computers and other devices as a usefulness (like the electricity grid) over a network. Cloud computing depend on sharing of resources to achieve consistency and economies of scale, similar to a utility on a network. On the establishment of cloud computing is the comprehensive concept of converged infrastructure and shared services.

Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the efficiency of the shared assets. Cloud resources are generally not only shared by multiple users but are also vigorouslytransferred per demand. This can work for allotting resources to users. For example, a cloud computer capability that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a diverse application (e.g., a web server). This method should maximize the use of calculating power thus reducing conservational damage as well since less power, rack space, air conditioning etc. are necessary for a variety of tasks. Through cloud computing, numerous users can access a single server to recover and update their data without acquiring licenses for dissimilar applications.

II. LITERATURE REVIEW

Jiansheng Wei, Hong Jiang, Ke Zhou, Dan Feng [01] focus on scalable high-throughput exact duplication approach, called MAD2, to eliminate duplicates both at the file level and at the chunk level in backend storage of network backup services. MAD2 utilizes on-disk Hash Bucket Matrix to preserve fingerprint locality and integrates in-memory Dual Cache to capture and exploit locality. In addition, MAD2 employs Bloom Filter Array to efficiently identify unique incoming fingerprints and indicate where a duplicate may reside. By employing a DHT-based Load-Balance technique to distribute file recipes and chunk contents among multiple storage nodes in their backup sequences, MAD2 further enhances performance with a balanced load.

Chunqiang Tang [02] focus on FVD which is a holistic virtual disk solution for both Cloud and non-Cloud environments. A design principle of FVD is to make all functions orthogonal so that each function can be enabled or disabled individually. The purpose is to support diverse use cases without being burdened with the overhead of all functions. Using copy-on-write, copy-on-read, and adaptive prefetching, FVD supports instant VM creation and instant VM migration, even if the VM image is stored on direct attached storage.

Chun-Ho Ng, Mingcao Ma, Tsz-YeungWong, Patrick P. C. Lee, and John C. S. Lui [03] focus on LiveDFS, a live deduplication file-system that is designed for VM image storage in an open source cloud with commodity configurations. LiveDFS respects the file system design layout in Linux and allows general I/O operations such as read, write, modify, and delete, while enabling inline deduplication. To support inline deduplication, LiveDFS exploits spatial locality to reduce the disk access overhead for looking up fingerprints that are stored on disk. It also supports journaling for crash recovery. LiveDFS is implemented as a Linux kernel driver module that can be deployed without the need of modifying the kernel source. We integrate LiveDFS into a cloud platform based on OpenStack and evaluate the deployment. In this work, we mainly focus on deduplication on a single storage partition.

Chung Pan Tang, TszYeung Wong, Patrick P. C. Lee [04] focus on CloudVS, an add-on system that provides version control for VMs in an open-source cloud that is deployed with

Manuscript received March 22, 2015.

S.Sasikala, PG Student STET Women's college, mannargudi
V.Geetha, Head of CS department, STET Women's college, mannargudi

commodity hardware and operating systems. CloudVS is built on redundancy elimination to build different

VM versions, such that each VM version only keeps the new and modified data chunks since the prior versions. It propose a simple tunable heuristic and several optimization techniques to allow CloudVS to address different performance trade-offs for different deployment scenarios.

BogdanNicolae, John Bresnahan, Kate Keahey and Gabriel Antoniu [05] focus on efficient management of VM images, such as image propagation to compute nodes and image snapshotting for check-pointing or migration, is critical. The performance of these operations directly affects the usability of the benefits offered by cloud computing systems. This paper introduced several techniques that integrate with cloud middleware to efficiently handle two patterns: multi-deployment and multi-snapshotting. It propose a lazy VM deployment scheme that fetches VM image content as needed by the application executing in the VM, thus reducing the pressure on the VM storage service for heavily concurrent deployment requests. Furthermore, we leverage object versioning to save only local VM image differences back to persistent storage when a snapshot is created, yet provide the illusion that the snapshot is a different, fully independent image.

SaskoRistov, MarjanGusev and AleksandarDonevski [06] focus on security assessments of OpenStack cloud services and four virtual machine instances with different operating systems Fedora, Ubuntu, CentOS and Windows. The experiments addressed the security vulnerabilities both from inside and outside the OpenStack cloud. The results of the assessments proved hypothesis that cloud multi-tenant environment raises new security vulnerabilities risks from inside the cloud, both for the tenants and the OpenStack cloud provider. Inside vulnerabilities ubsume the outside vulnerabilities for the cloud node and each operating system, which proves the hypothesis.

III. DEDUPLICATION

In computing, data deduplication is a specialized data compression technique for removing duplicate copies of repeating data. Related and somewhat identical terms are intelligent (data) compression and single-instance data storage. This method is used to increase storage operation and can also be functional to network data transmissions to diminish the number of bytes that must be sent. In the deduplication process, distinctive chunks of data or byte patterns are recognized and kept during a process of examination. As the examination continues, other chunks are matched to the stored copy and whenever a match take place, the redundant chunk is switched with a small reference that points to the stockpiled chunk. Assumed that the same byte pattern may occur dozens or even thousands of times (the match frequency is reliant on the chunk size), the amount of data that must be deposited or transferred can be greatly reduced.

This type of deduplication is dissimilar from that achieved by typical file compression tools, such as LZ77 and LZ78. While these tools recognize short repeated substrings inside individual files, the determined of storage-based data deduplication is to inspect large volumes of data and identify large segments such as entire files or large sections of files that are identical, in order to accumulate only one copy of it. This copy may be additionally flattened by single-file compression procedures. For illustration a typical email system contain 100 instances of the same 1 MB file attachment. Each time when the email platform is supported , all 100 instances of the add-on supplements are saved, demanding 100 MB storage space. Through data deduplication, only one occurrence of the attachment is actually stored; the succeeding instances are referenced back to the saved copy for deduplication ratio of roughly 100 to 1.

IV. WORKING

Cloud resource allocation

The virtualization is used to provide increasing number of servers on virtual machines (VMs), decreasing the number of physical machines required while conserving isolation between machine instances. This method better employs server resources, permitting many different operating system instances to run on a small quantity of servers, saving both hardware acquisition costs and operational costs such as energy, management, and cooling. Individual VM instances can be distinctly managed, letting them to serve a wide variety of purposes and preserving the level of control that many users want. In this module, clients store data into data servers for future usages. Then data servers store data in Meta servers.

Deduplication scheme

Deduplication is a tools that can be used to reduce the amount of storage necessary for a set of files by finding duplicate “chunks” of data in a set of files and storing only one copy of each chunk. Subsequent needs to store a chunk that exists precise in the chunk store are done by simply copying the identity of the chunk in the file’s block list; by not storing the chunk a second time, the system stores less data, thus decreasing cost. In this module, we introduce fingerprint scheme to identify the chunks that differ, both in fixed-size and in variable-size chunking use cryptographically in order to secure content hashes such as MD5 or SHA1 to identify chunks, thus permitting the system to swiftly discover that newly generated chunks that are previously in stored instances.

File system analysis

In this module, we first split VM disk images into chunks, and then analyze different sets of chunks to define both the amount of deduplication possible and the source of chunk similarity. It use the term disk image to denote the logical conceptcomprising all of the information in a VM, while image files denotes to the actual files that create up a disk image. A disk image is always connected with a single VM; a monolithic disk image comprises of a single image file, and a spanning disk image has further image files, each restricted to a particular size. Files are stockpiled in data server with block id and is can be monitored by Data servers. Data servers are mapped by means of Meta servers.

Data sharing components

In this module, we examine data sharing components and Meta server in LIQUID accountable for handling all data servers. It interchanges a regular heartbeat message with each data server, in order to keep an up to date idea of their health status. The Meta server exchanges heartbeat messages with data servers in a round-robin manner. This method will be slow to notice failed data servers when there are numerous data servers. To speedup failure detection, whenever a data server or client meets connection problem with another data server, it will send an error signal to the Meta server. A dedicated contextual daemon thread will immediately send a heartbeat message to the problematic data server and defines if it is alive. This mechanism confirms that failures are detected and handled at an early stage. The round-robin approach is still required since it could detect failed data servers even if no one is interactive with them.

P2P trust management

This module, we analyze P2P trust management system using bloom filter array types. A Bloom filter is a space-efficient probabilistic data construction that is used to test whether an element is a member of a set. False positive comparisons are possible but false negatives are nothaving a Bloom filter that has a 100% recall rate. Every client preserves connections to a set of peer clients pursued by the Meta server, and occasionally updates its copy of peer clients' Bloom filters. When appealing a data block by its fingerprint, it checks presence of the fingerprint between peer clients' Bloom filters in a random order, and efforts to fetch the data block from a peer if its Bloom filter contains the requested fingerprint. If the data block is not found between peers, the client will go back to draw the data block from data servers.

Evaluation criteria

Deduplication is an effective approach to lessen storage demands in environments with large numbers of VM disk images. Asdeduplication of VM disk images can save 80% or more of the space requisite to store the operating system and application environment,we explored the influence of many factors on the effectiveness of deduplication. We exhibited that data localization have little impact on deduplication ratio. Yet, factors such as the base operating system or even the Linux distribution can have a major impact on deduplication effectiveness. Thus, we recommend that hosting centers suggest "preferred" operating system deliveries for their users to ensure maximal space savings. If this preference is followed subsequent user activity it will have little effect on deduplication effectiveness.

V. ARCHITECTURE DIAGRAM

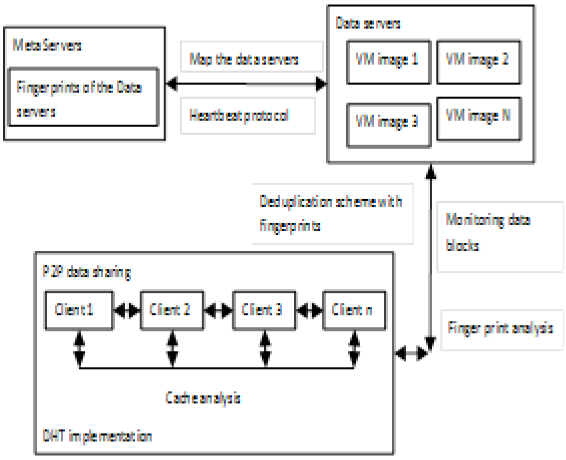

Fig1.1 Deduplication Scheme with Fingerprint

VI. CONCLUSION

Virtualization is widely used in servers to proficiently provide many reasonably separate execution environments while decreasing the need for physical servers. While this method saves physical CPU resources, it still consumes large amounts of storing because each virtual machine (VM) instance needs its own multi-gigabyte disk image. Besides, existing systems do not support block sharing between disks images, relying on techniques such as overlays to build multiple VMs from a single "base" image is efficient method. As an alternative, we suggest the use of deduplication to both lessen the total storage necessary for VM disk images and risen the ability of VMs to share disk blocks. To test the efficiency of deduplication, we showedwidespreadestimations on different sets of virtual machine disk images with chunking approaches. Astonishingly, the deduplication ratio of different issues within a given lineage does not rest on heavily whether the releases are successive. Deduplication uses to diminish the amount of storage disbursed by the VM disk images. However, our conclusions are built on real-world disk images, not images generated for deduplication testing thus, we rely on that these findings will generalize well to a wide array of VM disk images. In future we extend the framework with trust based peer to peer system to share cache files and provide Distributed hash table techniques to improve performance of data servers.

REFERENCES

[1]Jiansheng Wei, Hong Jiang, Ke Zhou, Dan Feng,"MAD2: A Scalable High-Throughput Exact Deduplication Approach for Network Backup Services", IEEE 2010.

[2] Chunqiang Tang,**"**FVD: a High-Performance Virtual Machine Image Format for Cloud**"**, June 2011.

[3] Chun-Ho Ng, Mingcao Ma, Tsz-YeungWong, Patrick P. C. Lee, and John C. S. Lui**, "**Live Deduplication Storage of Virtual Machine Images in an Open-Source Cloud", Proceeding Middleware11 Proceedings of the 12th ACM/IFIP/USENIX international Conference on Middleware, 2011.

[4] Chung Pan Tang, TszYeung Wong, Patrick P. C. Lee,” CloudVS: Enabling Version Control for Virtual Machines in an Open-Source Cloud under Commodity Settings”, IEEE 2012.

[5] BogdanNicolae, John Bresnahan, Kate Keahey and Gabriel Antoniu,”Going Back and Forth: Efficient Multideployment and Multisnapshotting on Clouds”,The20th International ACM Symposium on High-Performance Parallel and Distributed Computing HPDC 2011.

[6] SaskoRistov, MarjanGusev and AleksandarDonevski ,” OpenStack Cloud Security Vulnerabilities from Inside and Outside”, The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization , 2013.

S.Sasikala, PG Student STET Women’s college, mannargudi

V.Geetha, Head of CS department, STET Women’s college, mannargudi