ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

An Accessible Non-redundant Storage for Extensive Virtual Machine Implementation

Hemalatha.A

PG Student, Department Of Computer Science and Engineering,Parisutham Institute of Technology and Science,

Thanjavur, Tamilnadu, India

Abstract: In cloud computing and services, virtual machine powers with its features by providing resources and various infrastructure which is actually expected by large businesses and organizations. A challenging problem yet cannot be solved is the storage file system of a virtual machine instance. The existing deduplication concepts to reduce the VM storage are under DEDE and SAN clusters. In case of extensive VM hosting, all the existing techniques are unable to convince since they have inadequacy to cost. Our proposed system addresses accessible deduplication storage exclusively for Extensive VM implementation. Design of this system offers rapid creation of instances of VM including transmission of data in peer-to-peer networks. A wide range of features like very less utilization of VM storage, cloning of VM instances quicker and using copy on read techniques peer caches are managed to avoid bottlenecks.

Keywords: Deduplication, Cloud computing, virtual machine, storage, peer to peer, Copy-on-Read.

I. INTRODUCTION

Cloud Computing is widely considered as potentially the next dominant technology in IT industry. It offers simplified system maintenance and scalable resource management with Virtual Machines (VMs). As a fundamental technology of cloud computing, VM is being a hot research topic in recent years. The overhead of virtualization has been well addressed by hardware advancement in CPU industry, and by software implementation improvement in hypervisors themselves.

Typically in the platform of cloud computing, there are template images to create new VMs. In most cases, a limited number of combinations of OS and software applications will be used. Preparing template images for such combinations would be enough for most users' needs, which is a common practice adopted by cloud computing pioneers like Amazon EC2.

A virtual machine image is called as an instance of a cloud provided storage for a client, which is a virtual representation of a computer's hard disk. This image should be made accessible from host machine to boot a virtual machine in a cloud environment.

A. Issues in virtual machine storage

✓ A virtual server and desktop patterns are unpredictable that the time of happening and its arrival of great volume often.

✓ Whereas once one physical server drew upon its own dedicated disks, now many virtual servers simulate one box and can often demand I/O simultaneously from sharable storage.

✓ Ensuring the optimal delivery of resources being a big challenge in delivering virtual machine storage

In this paper, we have proposed a distributed file system particularly designed to simultaneously address the above problems faced in large-scale VM deployment. Its client side breaks VM images into small data blocks, references them by their fingerprints (calculated during a deduplication process), and uses deduplication techniques to avoid storing redundant data blocks. The deduplicated data blocks are then saved to a group of data servers, and the set of fingerprints is saved to a meta server.

When a VM image is being accessed, then a client downloads its set of fingerprints from the meta server, data blocks are fetched from data servers and peer clients in a P2P fashion, and exports an integrated VM image layout to hypervisors of system. The P2P data block transfer protocol of proposed system reduces requests directly issued to data servers, uses DAS on each host machine more effectively, and guarantees high scalability of the whole system.

The major impacts of this paper are listed as follows.

✓ We propose a deduplication storage with high-performance IO and low storage consumption, which satisfies the requirements of VM hosting.

✓ We provide a P2P data block sharing scheme, which is highly scalable for extensive deployment of VM images.

✓ We develop additional techniques for reducing network IO and expediting VM creation, including fast VM image cloning, copy-on-read and on-demand data block fetching, for high availability with the technique of fault tolerance.

✓ Therefore, the technique used in our proposed system be a file system which achieves good performance in handling multiple challenges of VM creation.

II. BACKGROUND

Our analysis reads various effects and use of implementing deduplication to the environment of extensive virtual machines

The growing number of VMs being deployed leads to increased burden on the underlying storage systems. For ensuring that advanced VM features like migration and high availability could work fluently, VM images need to be accessible from more than one host machine. This tends to the common practice to store VM images on shared network storage such as Network-attached storage (NAS) and SAN, for the ephemeral storage the direct-attached storage (DAS) of host machine is used. An approach has the issue that network storage systems usually cost several times more than DAS, and they have high demand on network IO performance. The critical need to store thousand VM images would be an extremely challenging problem for network storage systems because of the significant scale of storage consumption.

Studies have shown that the storage consumption issue brought by a large number of VM images could be addressed by deduplication techniques [6],[7], which have been extensively used in archival systems [8]. Existing systems have made efforts to address this issue on a SAN cluster by deduplication of data [9]. It is operated in a decentralized approach, therefore the deduplication is done at the running VMs in host machines, and unique data blocks are then stored over the SAN clusters. Though, SANs are very expensive, and thus difficult to satisfy the extensive need of VM image storage in the future.

There are a lot of papers, which focus on distribute file systems. GFS, HDFS, and OpenStack provide high availability by replicating stored data into multiple chunk servers.

In proposed system, every client is also a replica storing data blocks frequently used. This indicates that proposed system has high fault tolerance capability. Even if all data servers are down, a client would still be able to fetch data block from peer clients. Moreover, compared with these systems, our system has reduced storage consumption by 44 percent at 512KB data block size, eliminating the influence of back-up copy. Meanwhile, the I/O performance loss is just less than 10 percent.

Lustre is a parallel and distributed file system, generally used for cluster computing. The architecture of Lustre file system is similar to proposed system, but there are several important differences. Proposed file system has reduced storage consumption by using deduplication technology, and solved the bottleneck problem of metadata server owing to our P2P data block sharing scheme among all client nodes.

Amazon's Dynamo used DHT to organize its content. Data on the DHT are split into several virtual nodes, and are migrated for load balance in unit of virtual nodes. Proposed system follows this approach by splitting data blocks into shards according to their fingerprint, and management of data in unit of shards. In addition to the common distributed storage systems, HYDRAstor and MAD2 propose effective distributed architectures for deduplication. The former uses distributed hash table to distribute data, and the latter uses Bloom filter Array as a quick index to quickly identify non-duplicate incoming data. However, both of them focus on scalable secondary storage, which is not suitable for VM images storage. LiveDFS enables deduplication storage of VM images in an open-source cloud; however, it only focuses on deduplication on a single storage partition, which is difficult to handle the problems in distribute systems.

A. VM image formats

The VM images has two basic formats, they are

- ✓ Raw image format
- ✓ Sparse image format

The byte by byte copying of contents from the disks to a regular file is stated as raw image format of virtual machines.

The construction of complex mapping between blocks in physical disks and data blocks in VM images is termed as sparse image format.

B. Data Deduplication

Data deduplication is the process of comparing two objects as files or blocks and removing all redundant data available in it. By removing duplicated data blocks the storage consumptions can be reduced.

C. Deduplication Approaches

Its main goal is to improvise the storage utilization by eliminating the duplicate copies of data. The process of deduplication includes the identification of unique blocks of data using analysed fingerprint from their content. If any identical fingerprint is found then that is a redundant data block. The duplicated block is replaced with a reference to the stored block without copying it again. This technique is more effective than conventional compression tools.

Deduplication can be performed on the basic unit as a file, or sections inside that file. There are two methods to break a file into sections are fixed size chunking and variable size chunking [8].

The method of splitting the original file into block of the same size is termed as fixed size chunking. Whereas, variable size chunking is a complex method, by calculating Rabin fingerprint on a file content and detect natural boundaries in a file.

The archival systems, data that are rarely accessed uses variable size chunking. For virtual machine images, fixed size chunking is good enough in the measurement of deduplication ratio.

D. Uses of Data-Deduplication

The summarization of basic advantages of de-duplication is,

- ✓ Increase in storage efficiency
- ✓ Lowers the hardware costs
- ✓ Reduces the costs for backups
- ✓ Decreases the disaster recovery costs

E. Limitations

The disk images are mainly used to store OS and application files. User generated data could be stored directly into the disk image, but it is suggested that large pieces of user data should be stored into other storage systems, such as SAN. Temporary data could be saved into ephemeral disk images on DAS.

As in physical world, one hard drive cannot be attached to multiple machines at the same time. Assumption is aimed at achieving higher deduplication ratio and better IO performance for temporary data

DOI 10.17148/IJARCCE.2015.4230

III. DEDUPLICATION STORAGE FOR VMS

Proposed system consists of three components, i.e., a single meta server with hot backup, multiple data servers, and multiple clients. Each of these components is typically a commodity Linux machine running a user-level service process.

VM images are split into fixed size data blocks. Each data block is identified by its unique fingerprint, which is calculated during deduplication process. Proposed System represents a VM image via a sequence of fingerprints which refer to the data blocks inside the VM image.

The meta server maintains information of file system layout. This includes file system namespace, fingerprint of data blocks in VM images, mapping from fingerprints to data servers, and reference count for each data block. To ensure high availability, the meta server is mirrored to a hot backup shadow meta server.

The data servers are in charge of managing data blocks in VM images. They are organized in a distributed hash table (DHT) fashion, and governed by the meta server. Each data server is assigned a range in the fingerprint space by the meta server. The meta server periodically checks the health of data servers, and issues data migration or replication instructions to them when necessary.

It acts as a transparent layer between hypervisors and the deduplicated data blocks stored in our system. The client is a crucial component, because it is responsible for providing deduplication on VM images,

P2P sharing of data blocks, and features like fast cloning. When starting a new VM, client side of proposed file system fetches VM image meta info and data blocks from the meta server, data servers and peer clients, and provides image content to hypervisors. After the shutting down of VMs, the client side uploads modified metadata to meta server, and pushes new data blocks to data servers, to make sure that the other client nodes can access the latest version of image files.

A. Algorithms

The algorithms used in our system for generating signatures and finger print calculation are,

✓ Bloom Filtering Algorithm
✓ Hashing Algorithm (MD5, SHA1)

Bloom Filtering Algorithm, a Bloom filter is like a hash table, and simply uses one bit to keep track whether an item hashed to the location. The fingerprints of all their data blocks are compacted into Bloom Filter. ‘K’ different hash functions must also be defined, each of which maps a fingerprint to one of the m array positions randomly.

When adding a new fingerprint into the Bloom filter, the k hash functions are used to map the new fingerprint into k bits in the bit vector, which will be set as 1 to indicate its existence. To query for a fingerprint, we feed it to each of the k hash functions to get k array positions.

A Bloom filter uses an array of ‘m’ bits, all set to 0 initially, to represent the existence information of ‘n’ fingerprints ‘K’ different hash functions must also be defined, each of which maps a fingerprint to one of the m array positions randomly.

Notations in bloom filter implementations are,

✓ S is a set of n elements.
✓ Set of k hash functions with range {1 . . . m} (or {0 . . . m – 1}) .
✓ m-long array of bits initialized to 0.
✓ k hash functions $h_1 . . . h_k$
✓ We could use SHA1, MD5, etc.

To get a family of size k, $h_i(x)$ = MD5(x + i) or MD5(x || i) would work.

If any of the bits at these 0th position, then the element is definitely not in the set; otherwise, if all are 1, then either the element is in the set, or the bits have been to 1 by chance during the insertion of other elements.

Bloom filters have false positives. The probability of false positive is given as

$$(1-(1-1/m)^{kn})^k \simeq (1-e^{-kn/m})^k \tag{1}$$

Hashing Algorithms, a group of characters as a key taken in hash function and maps it to a value of a certain length as a hash value. The hash value is smaller than the original string which is the representative of the string of characters.

Hashing is process of indexing and locating items in databases because it is easier to find the shorter hash value than the longer string. Hashing is also used in encryption. The hashing algorithms used in proposed systems are,

✓ MD5
✓ SHA-1 (Secure Hash Algorithm)

Proposed System offers fault tolerance by mirroring the meta server, and by replication on stored data blocks. When the meta server crashes, the backup meta server will take over and ensure functionality of the whole system.

Replicas of data blocks are stored across data servers, thus crashing a few data servers will not impair the whole system.

B. Fixed Size Chunking

Proposed file system chooses fixed size chunking instead of variable size chunking. This decision is made based on

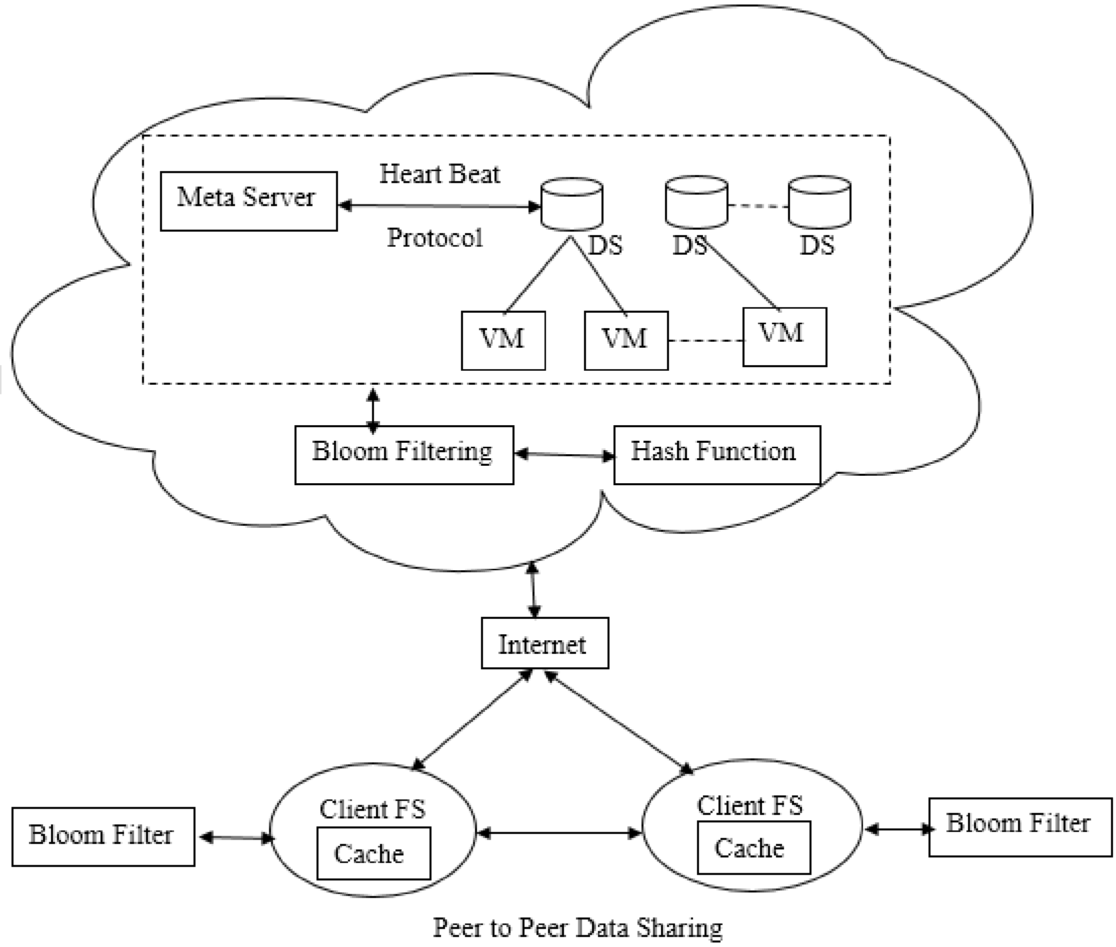

Fig. 1 Architecture of deduplication storage for virtual machine images

the observation that most x86 OS use a block size of 4 KB for file systems on hard disks. Fixed size chunking applies well to this situation since all files stored in VM images will be aligned on disk block boundaries. Moreover, since OS and software application data are mostly read-only, they will not be modified once written into a VM image. The main advantage if fixed size chunking is its simplicity. Storing data blocks would be easy if they have the same size, because mapping from file offset to data block could be done with simple calculations. Previous study has shown that fixed size chunking for VM images performs well in measure of deduplication ratio.

Proposed file system is compiled with block size as a parameter. This makes it more adaptive to choose different block size under different situation. Based on our experience, it is advised to use a multiplication of 4 KB between 256 KB and 1 MB to achieve good balance between IO performance and deduplication ratio.

Deduplication systems usually rely on comparison of data block fingerprints to check for redundancy. The fingerprint is a collision-resistant hash value calculated from data block contents. MD5 and SHA-1 are two cryptography hash functions frequently used for this purpose.

The probability of fingerprint collision is extremely small, many orders of magnitude smaller than hardware error rates [12]. So we could safely assume that two data blocks are identical when they have the same fingerprint.

C. Heartbeat Protocol

The meta server in proposed system is in charge of managing all data servers. It exchanges a regular heartbeat message with each data server, in order to keep an up to date vision of their health status.

The meta server exchanges heartbeat messages with data servers in a round-robin fashion. This approach will be slow to detect failed data servers when there are many data servers. To speedup failure detection, whenever a data server or client encounters connection problem with another data server, it will send an error signal to the meta server.

A dedicated background daemon thread will immediately send a heartbeat message to the problematic data server and determines if it is alive. This mechanism ensures that failures are detected and handled at an early stage. The round-robin approach is still necessary since it could detect failed data servers even if no one is communicating with them.

DOI 10.17148/IJARCCE.2015.4230

D. P2P Data Block Sharing

One advantage of our system is its P2P data block sharing scheme. It alleviates burden on data servers by sharing data blocks among all client nodes in a peer-to-peer fashion, eliminating network IO bottlenecks

E. On-Demand Data Block Fetching

Proposed File System uses the copy-on-read technique to bring data blocks from data servers and peer clients to local cache on demand as they are being accessed by a VM. This technique allows booting a VM even if the data blocks in the VM image have not been all fetched into local cache, which brings significant speedup for VM boot up process. Moreover, since only the accessed portion of data blocks are fetched, network bandwidth consumption is kept at a low rate, which is way more efficient than the approach of fetching all data blocks into local cache and then booting the VM.

F. Fast Cloning for VM Images

The common practice for creating a new VM is by copying from a template VM image. Most VM images are large, with sizes of several GB. Copying such large images byte-by- byte would be time consuming. It provides an efficient solution to address this problem by means of fast cloning for VM images.

The impact of many factors on the effectiveness of deduplication. We showed that package installation and the Linux distribution can have a major impact on deduplication effectiveness. Thus, we recommend that hosting centres suggest “preferred” operating system distributions for their users to ensure maximal space savings. If this preference is followed subsequent user activity will have little impact on deduplication effectiveness. We found that, in general, 40% is approximately the highest deduplication ratio if no obviously similar VMs are in-evolved. However, while smaller chunk sizes provide better deduplication.

G. Characteristics of proposed system

The properties of proposed system on deduplication storage for virtual machine images are listed below,

✓ It avoids additional disk operations incurred by local file system by organizing data blocks into large lumps.

✓ It supports instant VM image cloning by copy-on-write technique, and provides on-demand fetching through network, which enables fast VM deployment.

✓ P2P technique is used to accelerate sharing of data blocks, and makes the system highly scalable.

✓ Periodically exchanged Bloom filter of data block fingerprints enables accurate tracking with little network bandwidth consumption.

✓ Caching frequently modified parts of image blocks will avoid running expensive deduplication algorithms frequently, thus improves IO

IV. CONCLUSION

We explored that the proposed system, which is a deduplication file system with good IO performance and a rich set of features. This is achieved by caching frequently accessed data blocks in memory cache, and only run deduplication algorithms when it is necessary. Deduplication of VM disk images can save 80% or more of the space required to store the operating system and application environment; it is particularly effective when disk images correspond to different versions of a single operating system performance.

V. FUTURE WORK

In future to develop a Multi-Dimensional Vector in Bloom Filtering algorithm, this technique improves the data storage and avoids the duplication in large file system efficiently. Mainly these techniques to adapt for any bit operating system with Scheduling on large scale file system on Map Reduce Knowledge.

ACKNOWLEDGMENT

I would like to thank my guide Prof. P.Ramadoss, Asst. Prof., Computer Science and Engineering Department, Parisutham Institute of Technology and Science, Thanjavur for his help and guidance to enable me to propose this system

REFERENCES

[1] Ng, M. Ma, T. Wong, P. Lee, and J. Lui, *Live Deduplication Storage of Virtual Machine Images in an Open-Source Cloud*, in Proc. Middleware, 2011.

[2] M. Juric, Notes: Cuda md5 Hashing Experiments, May 2008. [Online]. Available: http://majuric.org/software/cudamd5/C.

[3] Xingjun Zhang ; Guofeng Zhu ; Yueguang Zhu., *An Undirected Graph Traversal Based Grouping Prediction Method for Data De-duplication.*, Published in: Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference on 1-3 July 2013

[4] Jingxin Feng and Jiri Schindler., *A Deduplication for Host-side Caches in Virtualized Data Center Environments.*, Posted by CTOEDITOR In Publications on May 11, 2013

[5] C. Tang., *Fvd: A High-Performance Virtual Machine Image Format for Cloud*. in *Proc. USENIX Conf. USENIX Annu. Tech*. Conf., 2011, p. 18

[6] K. Jin and E.L. Miller, *‘‘The Effectiveness of Deduplication on VirtualMachine Disk Images,’’* in Proc. SYSTOR, Israeli Exp. Syst. Conf., New York, NY, USA, 2009, pp. 1-12.

[7] A. Liguori and E. Hensbergen, *‘‘Experiences with Content Addressable Storage and Virtual Disks,’’* in Proc. WIOV08, San Diego, CA, USA, 2008, p. 5.

[8] B. Zhu, K. Li, and H. Patterson, *‘‘Avoiding the Disk Bottleneck in the Data Domain Deduplication File System,’’* in Proc. 6th USENIXConf. FAST, Berkeley, CA, USA, 2008, pp. 269-282

[9] A.T. Clements, I. Ahmad, M. Vilayannur, and J. Li, *‘‘Decentralized Deduplication in San Cluster File Systems,’’* in Proc. Conf. USENIX Annu. Techn. Conf., 2009, p. 8, USENIX Association.

[10] Ion Stoica Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan, *“Chord: A Scalable Peer to peer Lookup Service for Internet Applications”*, MIT Laboratory for Computer Science, 2013.

[11] M. McLoughlin, The qcow2 Image Format, Sept. 2011. [Online]. Available: http://people.gnome.org/markmc/qcow-image-format.html

[12] S. Quinlan and S. Dorward, *‘‘Venti: A New Approach to Archival Storage,’’* in Proc. FAST Conf. File Storage Technol., 2002, vol. 4, p. 7.

BIOGRAPHY

A.Hemalatha received B.Tech (IT) from Periyar Maniammai college of Technology for Women, Anna University in 2005 and received MBA (Technology Management), Anna university in 2010. She is currently persuading m.e – computer science in parisutham institute of technology and science, anna university.

DOI 10.17148/IJARCCE.2015.4230