ISSN Online: 2319 – 9253
Print: 2319 – 9245

A Study on Multi snap shooting in cloud computing

Swati Gupta [1] **Sangeeta Gupta** [2]

1. Asst.Prof. Dept. of CSE, B.V.Raju Institute of Technology, Hyderabad
2. Asst.Prof., Dept. of CSE, B.V.Raju Institute of Technology,Hyderabad

ABSTRACT

Cloud computing refers to the use and access of multiple server-based computational resources via a digital network , In cloud computing, applications are provided and managed by the cloud server and data is also stored remotely in the cloud configuration. Cloud computing has transform the way we think of acquiring resources ,by allowing users to lease computational resources from the cloud provider's datacenter for a short time by deploying virtual machines (VMs) on these resources. We proposes a new model to raises new challenges. First create vm's instances. Next take a snapshot of many images and transfer them to persistent storage to support management tasks, such as suspend-resume and migration. We use snapshotting hypervisor independent to ensure a maximum compatibility with different configurations. This paper addresses these challenges by proposing a virtual file system specifically optimized for virtual machine image storage. It is based on a lazy transfer schemes.

Keywords: virtual machines, cloud, snapshotting , image files, middle ware

I. INTRODUCTION

The Infrastructure as a Service cloud computing has emerged as a viable alternative to the acquisition and management of physical resources. With IaaS, users can lease storage and computation time from large datacenters. Leasing of computation time is accomplished by has complete control over the configuration of the VMs using on-demand deployments, IaaS leasing is equivalent to purchasing dedicated hardware but without the long-term commitment and cost. The on-demand nature of IaaS is critical to making such leases attractive, since it enables users to expand or shrink their resources according to their computational needs, by using external resources to complement their local resource base.

This problem is particularly acute for VM images used in scientific computing where image sizes are large. A typical deployment consists of hundreds or even thousands of such images. Conventional deployment techniques broadcast the images to the nodes before starting the VM instances, a process that can take tens of minutes to hours, not counting the time to boot the operating system itself.

II. EXISTING SYSTEM

The huge computational potential offered by large distributed systems is hindered by poor data sharing scalability.

We addressed several major requirements related to these challenges. One such requirement is the need to efficiently cope with massive unstructured data (organized as huge sequences of bytes - BLOBs that can grow to TB) in very large-scale distributed systems while maintaining a very high data throughput for highly concurrent, fine-grain data accesses.

The role of virtualization in Clouds is also emphasized by identifying it as a key component. Moreover, Clouds have been defined just as virtualized hardware and software plus the previous monitoring and provisioning technologies.

Cloud Computing is a “buzz word” around a wide variety of aspects such as

ISSN Online: 2319 – 9253
Print: 2319 – 9245

deployment, load balancing, provisioning, and data and processing outsourcing.

DISADVANTAGE

To give an less performance and storage space. Network traffic consumption also very high due to non concentrating on application status.

It is not possible to build a scalable, high-performance distributed data-storage service that facilitates data sharing at large scale.

III. PROPOSED SYSTEM

A distributed virtual file system specifically optimized for both the multi deployment and multi snapshotting patterns. Since the patterns are complementary, we investigate them in conjunction. Our proposal offers a good balance between performance, storage space, and network traffic consumption, while handling snapshotting transparently and exposing standalone, raw image files (understood by most hypervisors) to the outside.

We introduce a series of design principles that optimize multi deployment and multi snapshotting patterns and describe how our design can be integrated with IaaS infrastructures.

We show how to realize these design principles by building a virtual file system that leverages versioning-based distributed storage services.

ADVANTAGE

A good balance between performance, storage space, and network traffic consumption, while handling snapshotting transparently and exposing standalone, raw image files

SYSTEM IMPLEMENTATION: MODULES

- Cloud infrastructure
- Application state maintenance
- Application access pattern
- Aggregate the storage
- Image mirroring
- Striping the image
- Optimize multi snapshotting
- Zoom on mirroring

1. CLOUD INFRASTRUCTURE

IaaS platforms are typically built on top of clusters made out of loosely-coupled commodity hardware that minimizes per unit cost and favors low power over maximum speed . Disk storage (cheap hard-drives with capacities in the order of several hundred GB) is attached to each machine, while the machines are interconnected with standard Ethernet links. The machines are configured with proper virtualization technology, in terms of both hardware and software, such that they are able to host the VMs. In order to provide persistent storage, a dedicated repository is deployed either as centralized or as distributed storage service running on dedicated storage nodes.

2. APPLICATION STATE MAINTENANCE

The VM deployment is defined at each moment in time by two main components: the state of each of the VM instances and the state of the communication channels between them (opened sockets, in-transit network packets, virtual topology, etc.). To saving the application state implies saving both the state of all VM instances and the state of all active communication channels among them. While several methods have been established in the virtualization community to capture the state of a running VM (CPU registers, RAM, state of devices, etc.), the issue of capturing the global state of the communication channels is difficult and still an open problem.

3. APPLICATION ACCESS PATTERN

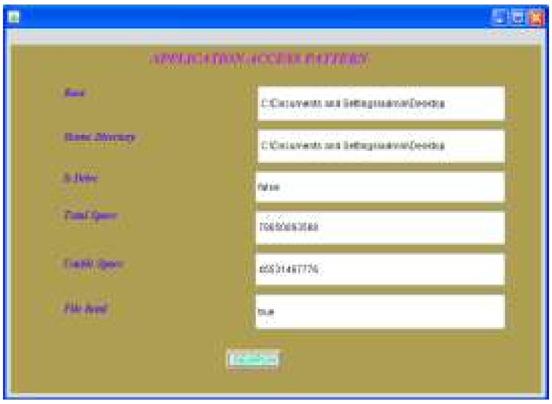
A VM typically does not access the whole initial image. For example, it may never access some applications and utilities that are installed by default with the operating system. In order to model this aspect, it is useful to analyze the life-cycle of a VM instance, it will based on Three phases. They are boot, application and shutdown.

4. AGGREGATE THE STORAGE

In most cloud deployments, the disks locally attached to the compute nodes are not exploited to their full potential. Most of the time, such disks are used to hold local copies of the images corresponding to the running VMs, as well as to provide temporary storage for them during their execution, which utilizes only a small fraction of the total disk size.

5. IMAGE MIRRORING

A new VM needs to be instantiated; the underlying VM image is presented to the hypervisor as a regular file accessible from the local disk. Read and write accesses to the file, however, are trapped and treated in a special fashion. A read that is issued on a fully or partially empty region in the file that has not been accessed before (by either a previous read or write) results in fetching the missing content remotely from the VM repository, mirroring it on the local disk and redirecting the read to the local copy. If the whole region is available locally, no remote read is performed. Writes, on the other hand, are always performed locally.



6. STRIPING THE IMAGE

Each VM image is split into small, equal-sized chunks that are evenly distributed among the local disks participating in the shared pool. When a read accesses a region of the image that is not available locally, the chunks that hold this region are determined and transferred in parallel from the remote disks that are responsible for storing them. Under concurrency, this scheme effectively enables the distribution of the I/O workload, because accesses to different parts of the image are served by different disks.

7. OPTIMIZE MULTISNAPSHOTTING

Saving a full VM image for each VM is not feasible in the context of multi snapshotting. Since only small parts of the VMs are modified, this would mean massive unnecessary duplication of data, leading not only to an explosion of utilized storage space but also to unacceptably high snapshotting time and network bandwidth utilization.

8. ZOOM ON MIRRORING

One important aspect of on-demand mirroring is the decision of how much to read from the repository when data is unavailable locally, in such way as to obtain a good access performance. A straightforward approach is to translate every read issued by the hypervisor in either a local or remote read, depending on whether the requested content is locally available. While this approach works, its performance is questionable. More specifically,

ISSN Online: 2319 – 9253
Print: 2319 – 9245

many small remote read requests to the same chunk generate significant network traffic overhead (because of the extra networking information encapsulated with each request), as well as low throughput (because of the latencies of the requests that add up).

VII. CONCLUSION

Finally we conclude that our investigation shows the 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. To introduced several techniques that integrate with cloud middleware to efficiently handle two patterns: multi deployment and multi snapshotting.

REFFERENCE

[1] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. Commun. ACM, 53:50–58, April 2010.

[2] Bar-Noy and S. Kipnis. Designing broadcasting algorithms in the postal model for message-passing systems. In SPAA '92: Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures, pages 13–22, New York, 1992. ACM.

[3] P. H. Carns, W. B. Ligon, R. B. Ross, and R. Thakur. Pvfs: A parallel file system for Linux clusters. In Proceedings of the 4th Annual Linux Showcase and Conference, pages 317–327, Atlanta, GA, 2000. USENIX Association.

[4] B. Claudel, G. Huard, and O. Richard. Taktuk, adaptive deployment of remote executions. In HPDC '09: Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing, pages 91–100, New York, 2009. ACM.

[5] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's highly available key-value store. In SOSP '07: Proceedings of 21st ACM SIGOPS Symposium on Operating Systems Principles, pages 205–220, New York, 2007. ACM.

[6] M. Gagn´e. Cooking with Linux—still searching for the ultimate Linux distro? Linux J., 2007(161):9, 2007.

[7] J. G. Hansen and E. Jul. Scalable virtual machine storage using local disks. SIGOPS Oper. Syst. Rev., 44:71–79, December 2010.

[8] M. Hibler, L. Stoller, J. Lepreau, R. Ricci, and C. Barb. Fast, scalable disk imaging with Frisbee. In ATC '03: Proceedings of the 2003 USENIX Annual Technical Conference, pages 283–296, San Antonio, TX, 2003.

[9] Y. J´egou, S. Lant´eri, J. Leduc, M. Noredine, G. Mornet, R. Namyst, P. Primet, B. Quetier, O. Richard, E.-G. Talbi, and T. Ir´ea. Grid'5000: A large scale and highly reconfigurable experimental grid testbed. International Journal of High Performance Computing Applications, 20(4):481–494, November 2006.

[10] K. Keahey and T. Freeman. Science clouds: Early experiences in cloud computing for scientific applications. In CCA'08: Proceedings of the 1st Conference on Cloud Computing and Its Applications, 2008.