Containers: A New Approach for Virtualization in Cloud Computing

SHWETA SHINDE[1], SHRUTI TOPLE[2], SATISH KUMBHAR[3]
[1, 2, 3]Department of Computer Engineering and Information Technology, College of Engineering Pune, Shivajinagar, Pune, INDIA
[1]shindess08.it@coep.ac.in, [2]topless08.it@coep.ac.in, [3]ssk.comp@coep.acin

ABSTRACT

Virtualization is one of the major building blocks of cloud architecture. Hypervisors which are widely used for virtualization, simulate host hardware thus decreasing performance of the virtual machines. In this paper we propose an alternative solution to achieve virtualization in cloud. Here we explore operating system virtualization based on lightweight separation of Operating System resources by the kernel. Technologies such as Linux Containers (LXC), OpenVZ or Zap pods for Linux, Jails for FreeBSD, and Zones for Solaris create separate resource namespaces called as 'containers' to implement operating system virtualization. Containers share the same kernel thus avoiding hardware emulation consequently increasing the performance. We also put forward a container based cloud architecture which preserves multi-tenancy, measured servicing, resource pooling, availability, etc. characteristics of cloud computing. Conclusively we present typical scenarios where this model is applicable.

Keywords: Cloud Computing, Containers, Hypervisor, Operating System Based Virtualization

1. INTRODUCTION

Cloud computing is not a quantum leap but a progressive evolution. It can be viewed as a parallel integration of prevailing technologies. Virtualization, distributed computing, utility computing together forms what is termed as cloud computing. Today, every business organization however small is interested in leveraging cloud in order to reap the benefits it claims to offer. Sharing of resources, on demand availability of required services, pay-as-per-use, etc. are some of the advantages attracting the organizations and motivating them to migrate to the cloud [1]. This sudden growth of cloud computing has opened new doors for researchers. Many studies are being carried out by universities and industries which focus on various aspects of this emerging trend of computing. The main focus of research includes performance enhancement, security, cost reduction and load-balancing. Multiple tenants share underlying hardware resources of the host in cloud computing. Each tenant is given a virtual machine. Virtualization is used to facilitate this sharing [9]. Hypervisor based virtualization is the most commonly used technology, but it suffers performance degradation. The main reason for this is the hardware emulation required to create virtual machines. Separate operating system runs on every virtual machine. Citrix® XenServerTM, VMware® ESXiTM, Microsoft® Hyper-V hypervisor, KVM and Virtualbox are some of the examples of hypervisors.

This paper proposes performance enhancement solution for cloud by using operating system virtualization which is also known as container based virtualization. The term ‘Containers’ is used hereafter to refer to a virtual machine created using operating system virtualization. Container virtualization uses kernel of the underlying host operating system referred as node. The containers do not have their own separate kernel. Each container has just a root file system of its own. This contains basic libraries, user level directories essential for the working of a container. Each container can have its own isolated namespace of network, files, process IDs, etc. Resources allocated to each container can be configured on demand to serve the incoming client requests. Linux Containers (LXC) [2], OpenVZ [3], Linux-VServer or Zap pods for Linux, Jails for FreeBSD, Zones for Solaris, iCore Virtual Accounts for Windows XP, WPARs for AIX, HP-UX Containers (SRP) for HPUX and Parallels® Virtuozzo are technologies available today for container virtualization.

The rest of the paper is organized as follows. In Section 2, discusses the major building blocks of cloud architecture in brief. Hardware, virtualization, automated management, provisioning of resources and security are identified as the important blocks. Concept of virtualization is introduced in section 3. It describes the two main types of virtualization currently available. Here, first discussions are done about the widely used hypervisor based virtualization and then container based operating system virtualization. Further, comparison of these two cutting edge technologies is done to show how OS based approach is superior when performance is the main concern. Section 4 describes how container based approach can be applied to realize cloud architecture and to preserve its characteristics of multi-tenancy, measured servicing, resource pooling, availability, etc. Applications of this model are also presented here. Section 5 discusses conclusions and future directions of research work that can be done with respect to the proposed idea.

2. BUILDING BLOCKS OF CLOUD

Following Sub-Sections discuss the major building blocks of cloud architecture.

2.1 Hardware

The main factor in increasing budget of computer organization is costly hardware required for its functioning. For starting a new organization or scaling its existing scope, tremendous investment needs to be done for purchasing, retaining or upgrading the existing hardware. Moreover, it is never utilised to the fullest capacity and demands high maintenance cost and skilled staff.

In cloud computing, the service providers are armed with abundant hardware. Organizations do not purchase any high end hardware. All they do is specify their requirements to cloud service provider. From then onwards, it's the service providers concern to make sure that requested hardware is provided to the customer.

The question is how do cloud service providers manage to meet dynamically changing hardware requirements of all its customers? The key is provisioning of hardware it has on shared basis among its clients. Hardware is provided in such a way that every customer thinks he has all of it to himself, but in reality it is multiplexed among multiple customers. This ensures maximum possible utilization of hardware. This essentially saves cost for both the service provider as well as the customers. Moreover maintenance cost, power, cooling, etc. are also managed by service providers, thus relieving customer from these tasks. As a result customer organization can focus more on its main stream business, rather than worrying about infrastructure and its maintenance.

2.2 Virtualization

Virtualization can be defined as a methodology of dividing the resources of computer hardware into multiple execution environments. This is done by applying one or more concepts or technologies such as hardware and software partitioning, time sharing, partial or complete machine simulation, emulation, etc. The concept of virtualization was first devised in the 1960s. It was then spearheaded in IBM to help split large mainframe machines into separate 'virtual machines'.

Virtualization block comes above the hardware block in cloud computing. Virtualization consists of a host system that provides one or more virtualized machine (guest system) environments [4]. Each customer is allocated with a virtual machine using virtualization techniques. Virtual machines are nothing but sandboxes. Security and isolation is essential so that applications on different virtual machines execute in an independent way. All requests from guest systems are serviced by host system. Host is the only entity allowed to control the hardware. Guest systems cannot directly access the hardware.

Hypervisor and containers are two techniques to achieve virtualization. Hypervisors need special hardware which supports virtualization. On the other hand containers do not require special hardware support. We will be discussing these two techniques in greater depth later in section 3.

2.3 Automated Management

At the heart of successful cloud implementations is the move beyond mere virtualization of infrastructure. It is a further step assuring cloud service delivery and management along with virtualization. Cloud computing is inherently distributed in nature thus making manual management a tedious task. Automating request, delivery, and management of cloud infrastructure is the next level after virtualization. Through the standardization and automation of shared computing resources, cloud computing speeds the delivery of services. Automated management in cloud computing paradigm ensures that the computing resources are evenly distributed among all the customers as per the requirement. Network bandwidth, disk space, CPU cycles, memory, etc. are some of the shared resources that need to be managed dynamically.

Node management, cluster management, fault tolerance, replication, synchronization, backup, recovery, load balancing, live migration, security are the important activities which demand automation when considered in context of cloud computing [5]. Automated management is also concerned with continuous monitoring of all the activities performed and usage of the resources by the customers. This assists in tracking the utilization done by each customer in order to charge them accordingly. Automation also helps in improving the quality of service by taking corrective actions from time to time.

2.4 Security

One of the trepidations on part of an individual, a business, a government agency or any other entity using cloud computing is anonymity of location where data is being stored. Cloud providers must ensure that the service is trustworthy and data is entirely safe and secure. Only then cloud computing will get wide acceptance among organizations. When considering security in cloud, we can broadly divide it in two major types. Firstly, data on the network which connects customer and cloud must be securely exchanged. This also includes securing data transfers within the cloud infrastructure. Secondly, data which is at rest on various storage devices in the service provider's data center must be protected. Most of the service providers today employ various encryption techniques to protect the data. Other major security aspects to be considered are data integrity, access control, authorization, authentication, data confidentiality and privacy, and legal compliances [6]. Some of the solutions deployed today use Virtual Private Networks, private cloud, firewall, securing management databases, logging and monitoring tools to overcome the security challenges posed by cloud.

2.5 Provisioning

Providing rapid and on demand service is backbone of the cloud philosophy. Provisioning which literally means taking care of delivery of services to customers (tenants) is thus essentially one of the building blocks of cloud. Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Storage as a Service (StaaS) are some of the service models offered by cloud [7].

3. TYPES OF VIRTUALIZATION

In this section, we briefly discuss the two major types of virtualization techniques.

3.1 Hypervisor Based Virtualization

First type of virtualization technique is hypervisor based virtualization. Hypervisor, which is sometimes referred to as virtual machine monitor (VMM), allows several operating systems (Guest OS) to share single hardware of host. Hypervisor controls the host processor and resources. It is responsible for allocating resources and ensuring that the guest operating systems function independently without interfering with each other. This method requires special hardware support to form a thin base layer which redirects all instructions coming from virtual machines to the underlying hardware layer. In this model, each guest (Virtual Machine) is a completely installed Operating System right from the kernel to libraries, applications and so on. In case of paravirtualization, access to hardware does not require virtualizing all drivers. But when using full virtualization, complete hardware needs to be virtualized resulting in significant loss of performance. The current options of hypervisors like vSphere, Hyper-V, XenServer, etc. place an upper limit on amount of resources to be allocated per client. This makes it impossible for guest machines to fully utilize the available resources of host.

3.2 Container based OS Virtualization

A container-based system provides a shared, virtualized OS image consisting of a root file system, a (safely shared) set of system libraries and executables. Every guest system (container) shares same host kernel but has its own set of processes, memory, network, etc. and hence completely avoids emulation of hardware. Process of container is assigned a separate user permission, networking, filesystem name space from its parent [8].

Approach used for virtualization with containers differs completely from KVM and Xen technologies for system virtualization. Containers started out with existing Linux process management and added isolation mechanism. Chroot functionality of UNIX forms the foundation of containers, wherein root directory itself is changed consequently enabling execution of applications in independent sandboxes. Linux Containers (LXC), OpenVZ, Jails, etc. add resource management and namespaces to chroot, yielding a scalable system virtualization mechanism. Containers support large number of instances on a single host and still preserve performance thus providing lightweight and low density virtualization. A new container is created by specifying a configuration file which contains name of the container, network configuration, capabilities, resource requirements and permissions for devices to be allocated to the container. This configuration file is referred while booting up a container. The operations possible on a container are create, start, stop, freeze, unfreeze, destroy, restart and execute.

4. CONTAINERS AND CLOUD

The benefits of container virtualization can be used to implement a better strategy to address the performance and scalability challenges of cloud computing. Fig. 1. gives an architecture to realize cloud environment by using container virtualization.

The architecture depicts a typical cloud comprising of geographically distributed data centres of cloud service provider. A datacenter consists of multiple nodes simultaneously serving various clients connected to the cloud via the internet. All these nodes are connected to a node controller which has the responsibility of managing them. Node controllers of all datacenters of service provider communicate with each other continuously for global management of the cloud. Every node is a high end server machine which acts a host for container virtualization. Cloud nodes are virtualized using technologies such as Linux Containers (LXC) [2], Jail, OpenVZ [3], Linux VServer, Virtuozzo, etc. to create multiple containers. Host has a dedicated hardware with an operating system installed on it. Containers created on a node share the kernel of host operating system.

Every customer using cloud has different requirements in terms of memory, CPU, network bandwidth, etc. To meet such diverse demands of millions of users, we allocate a separate container to every user. The container is customised by changing the configuration parameters of container as per the client's requirements. Dynamic change can be incorporated by tuning the configurations. Even though multiple clients use same kernel of host, containers are isolated from each other thus ensuring secure multi tenancy. All the node controllers are responsible for maintaining consistency and availability of the containers. This can be achieved using the check pointing, live migration, snapshotting features of the containers. Monitoring of the containers assists in meeting the pay-as-per-use characteristic of cloud computing [10].

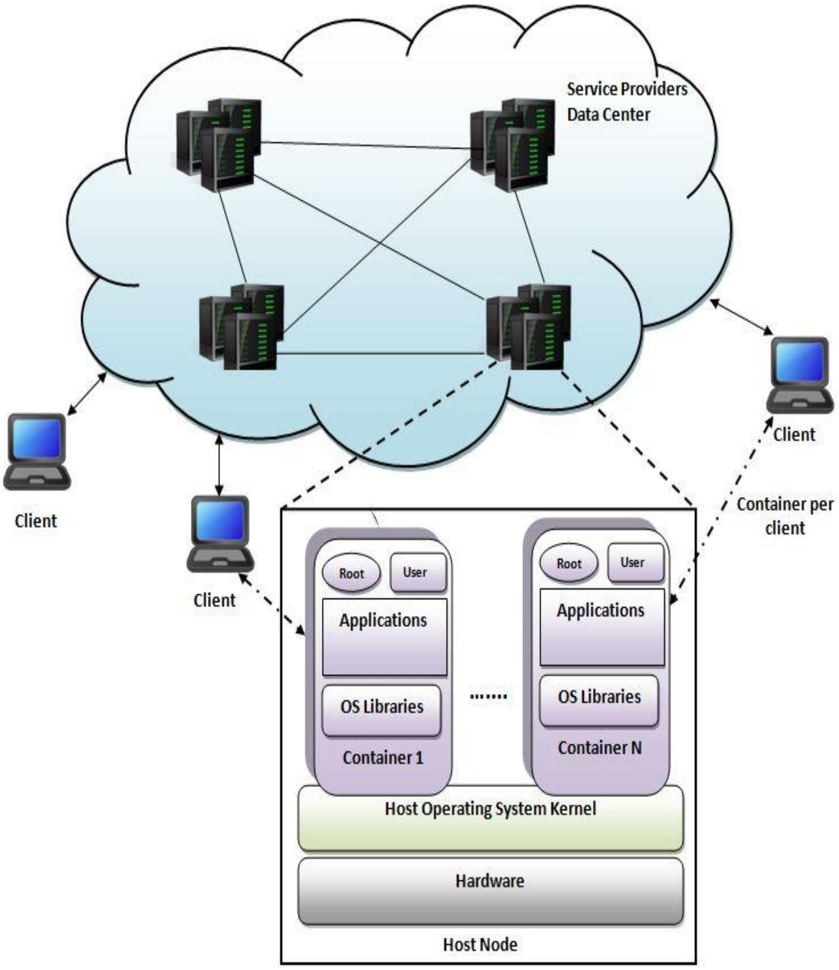

Fig.1. Architecture of container based cloud.

As of now, applications of containers are scenarios where the guest instances to be consolidated use the same kernel type. The fact that same kernel is shared among containers hinders their usage to areas which demand different kernel for each container. To address this technical limitation, generally service providers deploy different operating systems on various nodes. For example, one host node runs Linux kernel while the other runs Windows kernel. Another approach to tackle this limitation is to use hypervisors to create virtual machines of different kernels on the host node. This virtual machine (VM) can then be used as host OS for creating containers. By this approach even one host node is sufficient to meet the requirement of containers with different kernels. In short, we do container based OS virtualization of a VM created using hypervisor based virtualization.

A typical application of container based cloud is Storage as a service [11]. As data storage services on cloud do not require different operating system types, this purpose can be served by using containers.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced container based operating system virtualization technology as an option for virtualization - one of the main pillars of cloud computing [9].We have also proposed cloud architecture which uses containers to provide service to multiple tenants. We further discussed the noticeable usage scenarios where this method can be best applied. In a nutshell, containers surely are promising solution to ameliorate cloud computing.

Containers are still one of the less explored areas from cloud computing perspective. Work needs to be done in order to make containers more secure and manageable so as to compete with mature hypervisor technology [12].

ACKNOWLEDGEMENTS

We acknowledge Mr. Sandeep R. Patil and Mr. Deepak R. Ghuge of IBM India Storage Lab for the contribution & guidance throughout the research work.

REFERENCES

[1] Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen and Zhenghu Gong (2010), The Characteristics of Cloud Computing, *Proc. Parallel Processing Workshops (ICPPW), 39th International Conference*.

[2] Linux Containers (LXC), http://lxc.sourceforge.net/, 2012.

[3] OpenVZ, http://wiki.openvz.org/, 2012

[4] Theera Thepparat, Amnart Harnprasarnkit, Douanghatai Thippayawong, Veera Boonjing, Pisit Chanvarasuth (2011), A Virtualization Approach to Auto-Scaling Problem, *Proc. Information Technology: New Generations (ITNG), Eighth International Conference*.

[5] Ramgovind S, Elo MM, Smith E (2010), The Management of Security in Cloud Computing, *Proc. Information Security for South Africa (ISSA).*

[6] Minqi Zhou, Rong Zhang, Wei Xie, Weining Qian, Aoying Zhou (2010), *Proc. Security and Privacy in Cloud Computing: A Survey, Semantics Knowledge and Grid (SKG, Sixth International Conference.*

[7] Malathi, M. (2011), Cloud computing concepts, *Proc. Electronics Computer Technology (ICECT), 3rd International Conference*.

[8] Stephen Soltesz, Herbert Ptzl, Marc E. Fiuczynski, Andy Bavier, Larry Peterson(2007), Container-based Operating System Virtualization:A Scalable, High-performance Alternative to Hypervisors, *Proc. 2nd ACM SIGOPS/EuroSys European Conference onComputer Systems*.

[9] Hanfei Dong, Qinfen Hao, Tiegang Zhang, Bing Zhang (2010), Formal Discussion on Relationship between Virtualization and Cloud Computing, *Proc. Parallel and Distributed Computing, Applications and Technologies (PDCAT).*

[10] Peter Mell and Tim Grance(2009), *The NIST Definition of Cloud Computing*, Version 15.

[11] Robert Haas, Anil Kurmus, Moitrayee Gupta, Roman Pletka and Christian Cachin, A Comparison of Secure Multi-tenancy Architectures for Filesystem Storage Clouds, IBM Research Zurich.

[12] Jianfeng Yang, Zhibin Chen (2010), Cloud Computing Research and Security Issues, *Proc. Computational Intelligence and Software Engineering (CiSE).*

AUTHORS BIOGRAPHY

Ms. Shweta Shinde is a senior undergraduate student at the Department of Computer Engineering and Information Technology at College of Engineering, Pune. Her research interests include Storage Systems, Virtualization Technologies, Cloud Computing, Distributed Systems and Object Oriented Programming.

Ms. Shruti Tople is a senior undergraduate student at the Department of Computer Engineering and Information Technology at College of Engineering, Pune. Her research interests include Storage Systems, Virtualization and Operating Systems.

Mr. Satish Kumbhar received his M.Tech in Computer Engineering from College of Engineering, Pune. He is working presently as Assistant Professor in Department of Computer Engineering and Information Technology at College of Engineering Pune. He has 7 years of experience in teaching. He has taught various subjects in Computer Science and Engineering like Microprocessor Techniques, Data Communication and Networking, Human Computer Interface, Probability and Statistical Inference, Data Structure and Algorithm, Linux and Xen Virtualization, Advanced Computer Network etc.