Challenges in Real-Time Virtualization and Predictable Cloud Computing

Dushyant Singh, Assistant Professor

Department of Computer Science and Engineering, Vivekananda Global University, Jaipur

Email Id- dushyant.singh@vgu.ac.in

ABSTRACT: In the last decade, cloud computing and desktop virtualization have transformed general-purpose computer applications. Reduced operational expenses, server consolidation, flexible design, and elastic resource supply are all benefits of the cloud paradigm. Despite cloud computing's success in general-purpose computing, current cloud computing and virtualize has significant difficulties in supporting new soft real-time applications such online video streaming, cloud-based gaming, and telephony management. Cloud computing, especially the usage of public clouds, offers technical, environmental, and economic benefits by enabling many under-utilized systems to be merged onto fewer physical servers. Cloud computing data centers, like a massive physical machine, provide infrastructure, platform, and software applications as services to customers. In an open, shared, and virtualized computer environment, these applications require real-time performance. This article examines the technical difficulties of supporting real-time applications in the cloud, reviews recent advances in real-time virtualization and cloud computing technologies, and proposes future research paths to allow cloud-based real-time applications.

KEYWORDS: Cloud Computing, Operating System, Real-Time Application, Technology, Virtualization.

1. INTRODUCTION

The widespread availability of low-cost high-speed Internet connections through DSL and, more recently, optical technologies, along with unparalleled connectedness via cellular and wireless technologies, is allowing an unavoidable transition to distributed computing architectures. As the PC fades, paving the way for a new age of distributed computing, applications that depend only on physical resources and data accessible in the local personal computer (PC) are slowly but steadily becoming part of history. This is encapsulated in the newly growing concept of Cloud Computing, in which cloud providers rent resources on-demand and pay-per-use. Cloud computing data centers, like a massive physical machine, provide infrastructure, platform, and software applications as services to customers. IaaS (infrastructure as a service), PaaS (platform as a service), and SaaS (software as a service) are the terms used to describe these services. Cloud apps can operate and distribute their calculations and data over as many nodes as they require, and they can access massive quantities of data stored on the premises of cloud data centers. Briefly, cloud computing enables the next generation of computing services, which are strongly oriented toward massively distributed and on-line computing, as well as a new paradigm of on-demand high performance computing (HPC) that is available to everyone, from anywhere, at any time[1]–[4].

Real-time systems are anticipated to develop in this way when new application domains join the cloud world, owing to the enormous potential and increased usefulness that this paradigm may offer. Military distributed control systems used to remote surveillance, early reaction and warning systems, autonomous vehicles with enhanced intelligence from the sensor cloud, and cloud gaming, among others, are examples of both the hard and soft sides of real-time systems.

Cloud computing, especially the usage of public clouds, offers technical, environmental, and economic benefits by enabling many under-utilized systems to be merged onto fewer physical servers. A cloud provider can scale on hundreds of thousands of clients (a.k.a., tenants) with constantly changing workload needs, re-optimizing the infrastructure in a fully automated (or semi-automatic) manner as required, and delivering high levels of availability and dependability. Virtualization, especially machine virtualization, is one of the most significant technologies that allowed this paradigm change in computing[5], [6].

Computer virtualization (also known as processor virtualization) enables a single physical machine to mimic the behavior of many machines, allowing different and heterogeneous operating systems (referred to as guest OSes) to be hosted on the same hardware. A virtual machine monitor (VMM), often known as a hypervisor, is a software infrastructure capable of executing such emulation that runs on (and has complete control over) the physical host.

Multiple operating systems that might otherwise leave their underlying hosts underutilized may be transferred to the same physical resources thanks to virtualization in data centers. This allows for a decrease in the number of physical hosts needed, as well as better exploitation at greater saturation levels, resulting in cost and energy savings. Cloud computing's multi-tenant nature has a big impact on the increasingly complex and demanding user needs for cloud infrastructures. Users expect real-time and interactive apps and services, in addition to on-line storage. This is also evident in already-available futuristic goods, such as lightweight computers that are almost incapable of accomplishing anything locally without linked to the "Cloud."

Applications in a high-performance cloud computing (HPCC) environment have considerably higher temporal needs; therefore, performance characteristics such as resource guarantees and timely supply of results become essential. Due to I/O overhead and jitter of the necessary length of executed instructions, it is currently an active research field to match such requirements with virtualized systems. Furthermore, although most activities or tasks are assigned to a particular core, they often have synchronization dependencies with other activities. Combining cloud computing with real-time is a difficult issue that requires consideration of many factors, including efficient access to the physical platform[7], [8].

Although real-time hypervisors generally enable applications to access the actual machine, it is apparent that the hardware in virtualized environments for cloud computing is typically not directly accessible by the user-level application software layers.

Utilizing high-performance methods, it may be feasible to enhance service response times from a cloud platform using existing technologies. The primary issues are on the multitenancy of cloud computing systems, which process the requests of many independent users on heavily crowded servers. WindRiver, Acontis Technology, SysGO, OpenSynergy, LynuxWorks, or Real Time Systems Gmbh are examples of commercial real-time hypervisors (some of which include hierarchical scheduling) for safety critical systems of various sources.

However, owing to performance and compatibility issues at the low-level execution layer, they are unlikely to become standard cloud technology in the near future.

The mainstream and real-time versions were developed with distinct goals in mind. Real-time hypervisors, for example, were created to maintain temporal isolation and determinism rather than to maximize throughput of user requests and provide statistical assurances on service contracts.

Cloud computing was not designed with demanding real-time applications in mind, since they usually operate in confined settings. This article focuses on the major difficulties that cloud-computing technologies provide when used in soft real-time applications. Online video streaming (e.g. Netflix on Amazon EC2), cloud-based games, and telephony administration are examples of soft real-time domains. Due to their extremely dynamic workloads that need elastic resource allocation, these applications may greatly benefit from cloud computing. The industry for cloud-based gaming is gaining traction. Netflix, which is hosted on Amazon EC2, and both Xbox and Playstation are going to provide cloud-based gaming. For example, Microsoft's Xbox One game system allows for cloud-based calculation of environmental components, while Sony just purchased Gaikai, a significant open cloud gaming platform. As more latency-sensitive games and players migrate to the cloud, meeting different latency requirements in the cloud-computing environment becomes more essential. According to user research, networked games need a quick reaction time, even as low as 100 milliseconds for first-person shooter games. Virtualized network functions are implemented in private or hybrid clouds of network operators, signaling a significant change in the telecommunications industry from hardware-based provisioning of network services to a software-based provisioning paradigm. IP Multimedia Subsystem (IMS) components, for example, are typically designed and calibrated to run on specific hardware platforms with exact real-time and reliability requirements, based on a target maximum workload specification, such as the maximum number of supported subscribers or call attempts per second. Matching the same requirements in a virtualized context where a plethora of Virtual Machines (VMs) share the same physical hardware for providing a plethora of services with highly heterogeneous performance requirements to independent customers/end-users presents a number of challenges, some of which can be addressed as summarized in this paper[9], [10].

For soft real-time applications, bypassing system software and exposing the hardware directly to apps is neither necessary nor productive. Incorporating real-time scheduling rules into virtualization platforms, on the other hand, provides a direct advantage, since the system may provide real-time performance to applications in a hierarchical way. In this article, we discuss the issues that arise when soft real-time workloads are mixed with the needs of distributed and virtualized physical infrastructures, such as in cloud computing. We discuss the concept of service level agreements and the difficulties of including real-time characteristics into them. The article presents a review of soft real-time applications that need specific levels of service level agreements in terms of real-time performance but do not require hard real-time performance guarantees. We go through some of the options for integrating a real-time model into a virtualized application paradigm. We discuss various methods to HPCC, as well as the difficulties posed by network communication, which necessitates I/O access, resulting in additional delays; we also offer some ideas for increasing the network's efficiency.

1.1 Virtualization and cloud computing:

The existence of its enabler, namely virtualization technologies, offers a variety of advantages to the cloud computing paradigm. To begin, this section gives a high-level summary of cloud computing's advantages. In reality, the virtualization type chosen is always a compromise between the closeness of control and access to the actual underlying hardware platform, the performance provided to the hosted program, and the development and deployment flexibility.

Before going through the various kinds of virtualization (which are covered in this part), it's important to understand the architecture of virtual machines, which is quite similar to the design of a hardware platform.

1.2 Virtualization's Advantages:

Virtualization technology provides programs with an abstract view of the underlying hardware platform and resources through interfaces. Virtualization, as stated in [10] and [11], offers a number of advantages for cloud computing:

- Isolation of functional execution. The hypervisor is in charge of protecting virtual machines (VMs) and, as a result, applications running on various VMs. Within their virtual machine, users may be given rights without jeopardizing the isolation or host integrity.
- Virtualization allows for the creation of highly specialized and customized environments, including operating systems, libraries, and run-time execution environments. Virtualization, in reality, provides functional separation, allowing different activities to run on the same physical hardware.
- Easier to handle. Customized runtime environments may be created, moved, and shut down in a highly flexible manner, depending on the requirements of the hardware provider.
- The coexistence of old and modern applications. Virtual machines (VMs) help legacy applications maintain binary compatibility in run-time environments.
- Virtualized environments may be used to test and debug parallel programs since a completely distributed system can be simulated on a single physical host.
- Hypervisors and their live-migration capabilities make it possible to improve the dependability of hosted virtualized applications while keeping them independent of the underlying hardware in a smooth and transparent way.

Nonetheless, cloud computing's multi-tenancy characteristic, along with its greater degree of consolidation, is one of the reasons posing many difficulties that have yet to be adequately addressed. The increasing sharing of physical resources across numerous software components and applications hosted on behalf of different customers makes providing consistent and predictable performance levels to each of them more challenging. Virtual machines (VMs) may, in fact, run in parallel on a virtualized platform, vying for physical resources that are planned by an underlying hypervisor. Within VMs, the processor is further given to tasks or threads according to the guest OS particular scheduling rules.

Despite the disadvantages of the above-mentioned feature of server consolidation, real-time applications in the cloud may profit from the same benefits as general-purpose apps.

The creation of a private cloud for remote control and operation of an industrial automation facility is an example. Sensors may be used to monitor plant operations, and the data can be saved and analyzed in a private cloud. In addition, the operating logic of certain machines/instruments may be outsourced to a private cloud, which simplifies plant floor maintenance and remote control. The virtual nodes latencies as well as the communication jitter must be managed due to the real-time restrictions of certain operations.

1.3 Architectures of virtual machines:

1.3.1 Interfacing of hardware and software:

Despite being implemented on more complicated hardware architectures, software systems continue to develop. This is feasible because computer systems are designed in a hierarchical manner with well-defined interfaces that divide layers of abstraction neatly. It's common knowledge that well-structured and specified interfaces make it easier for engineering teams to create separate subsystems for both software and hardware systems. A good illustration of this is the instruction set architecture (ISA).

The sheer use of interface, however, has certain drawbacks. A binary format of a program, for example, is bound to a particular ISA and operating system libraries. This was discovered to be restrictive, especially in an ultra-connected environment where the ability to quickly move code from one computer to another had an obvious advantage. Machine virtualization solves this restriction by adapting a system's interface and visible resources to an underlying, potentially different, actual system's interface and resources. The aim of virtualization technology is not to conceal or simplify a system's underlying elements. The aim of virtualization is to link a virtual resource to a real one by using the interfacing abstraction of real resources or subsystems.

1.3.2 Virtual execution environment development:

The fidelity with which they implement the different interfaces provided within the architecture of a computer system, i.e., hardware or instruction set architecture, operating system interface, application binary interface, or application programming interface, determines the approach used to develop virtual execution environments. In the case of JVM, for example, a unique instruction set architecture is given, which includes a dynamic translator or interpreter for code written in a high-level language such as Java.

As a result, a virtual machine (VM) is a software implementation of a physical execution platform, machine, or computer that can execute programs in the same way that the actual machine can. A virtual machine (VM) abstracts a hardware platform, an operating system, and even a set of libraries and programs to execute. The VM then runs additional software layers, which may result in time delays.

Virtual environment design may be done in two ways: via hardware partitioning or through hypervisor technology. Hardware partitioning splits a physical computer into several "partitions," each of which may run its own operating system. These divisions are usually made using coarse allocation units.

This kind of virtualization allows for hardware consolidation, but it lacks the resource sharing and emulation advantages that hypervisors provide.

1.3.3 Types of hypervisor

The hypervisor's design and implementation are critical because it has a direct impact on the throughput of virtual machines, which may be very near to that of native hardware.

Hypervisors are the preferred technology for system virtualization because they provide for a great deal of freedom in the definition and management of virtual resources. Dividing and/or virtualizing platform resources accomplish Hypervisor replication. There are two kinds of hypervisors, as described in:

- Type 1 (bare metal) software that operates on the actual hardware platform. It virtualizes key hardware components and divides them into many separate, isolated partitions. It also offers basic control and communication functions between partitions. Because virtual environments employing type 1 hypervisors are near to the hardware and may utilize hardware resources directly, they are ideal for real-time systems.
- Type 2 hypervisors: those that operate on top of a host operating system. Because they operate in a traditional operating system environment, these hypervisors are referred to as hosted hypervisors. The hypervisor layer is a differentiated software layer that operates on top of the host operating system (which runs directly above the hardware), while the guest operating system runs at a separate level.

2. DISCUSSION

The author has discussed about the challenges in real-time virtualization and predictable cloud computing. Cloud computing is a relatively new paradigm for dynamically lifecycle management computing services located in data centers that are heavily invested in virtualization technology, allowing server quantities of material and efficient resource usage in general, despite the fact that it has been a reality for several years. Important improvements in machine virtualization, networking, data analysis, and storage have led to the widespread use and acceptance of these paradigms in many areas during the past few decades.

Real-time application domains are still lagging behind in cloud computing adoption due to their stringent timing constraints and the necessity for predictability assurances. Consolidating the predictable features that real-time virtualization technology will be able to provide in the future is a difficult issue that needs real-time virtualization technology. There are still some significant obstacles and difficulties in the way of its complete acceptance by real-time domains, particularly hard real-time applications.

3. CONCLUSION

The author has concluded about the challenges in real-time virtualization and predictable cloud computing. Reduced operational expenses, server downsizing, flexible system design, and elastic resource supply are all benefits of the cloud paradigm. Despite cloud computing's success in general-purpose computing, current cloud computing and virtualization software has significant difficulties in supporting new soft real-time applications such online streaming content, cloud-based gaming, and telephony management. In an open, shared, and virtualized computer environment, these applications require real-time performance. Like a large physical machine, cloud computing data centers offer infrastructure, platform, and application programs as a service to clients. These services are referred to as IaaS (infrastructure as a service), PaaS (platform as a service), and SaaS (software as a service). Cloud applications may use as many nodes as they need to run and disseminate their computations and data, and they can access enormous amounts of data housed on the grounds of cloud data centers.

REFERENCES

[1] "Virtualization in Cloud Computing," *J. Inf. Technol. Softw. Eng.*, 2014, doi: 10.4172/2165-7866.1000136.

[2] K. S. K. Devi, G. S, and D. R, "Virtualization in Cloud Computing," *IJARCCE*, 2018, doi: 10.17148/ijarcce.2018.71122.

[3] W. Ding, B. Ghansah, and Y. Wu, "Research on the Virtualization technology in Cloud computing environment," *Int. J. Eng. Res. Africa*, 2016, doi: 10.4028/www.scientific.net/JERA.21.191.

[4] M. García-Valls, T. Cucinotta, and C. Lu, "Challenges in real-time virtualization and predictable cloud computing," *J. Syst. Archit.*, 2014, doi: 10.1016/j.sysarc.2014.07.004.

[5] N. Khan, A. Shah, and K. Nusratullah, "Adoption of Virtualization in Cloud Computing," *Int. J. Green Comput.*, 2016, doi: 10.4018/ijgc.2015010104.

[6] F. Lombardi and R. Di Pietro, "Secure virtualization for cloud computing," *J. Netw. Comput. Appl.*, 2011, doi: 10.1016/j.jnca.2010.06.008.

[7] A. P. M and M. T. Sathiyabama, "Virtualization in Cloud Computing," *Int. J. Trend Sci. Res. Dev.*, 2018, doi: 10.31142/ijtsrd18665.

[8] A. Oludele, E. C. Ogu, K. 'Shade, and U. Chinecherem, "On the Evolution of Virtualization and Cloud Computing: A Review," *J. Comput. Sci. Appl.*, 2014, doi: 10.12691/jcsa-2-3-1.

[9] R. M. Sharma, "The Impact of Virtualization in Cloud Computing," *Int. J. Recent Dev. Eng. Technol.*, 2014.

[10] M. Singh, "Virtualization in Cloud Computing- a Study," in *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, 2018, doi: 10.1109/ICACCCN.2018.8748398.