A Review paper on Real Time Challenges in Cloud Computing

Sitaram Gupta, Assistant Professor

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

Email Id- sitaram.gupta@vgu.ac.in

ABSTRACT: Virtualization is becoming more important in real-time and embedding systems design to provide flexibility and separation across separate run-time environments. At the platform level, it is necessary to develop direct methods to convert cloud service level agreements (SLAs) to resource assignment and enforcement. SLAs usually deal with platform resources like as bandwidth, latency, and memory. A real-time viewpoint, on the other hand, must enter the picture to offer actual execution methods that ensure and enforce execution and temporal isolation in order to satisfy the timing constraints of real-time applications. Virtualization of resources should be kept at a similar distance from application execution and management of computing resources at the platform level. Providing an efficient operational management framework, particularly for resources in the form of networks, may be particularly challenging in this area. In this vein, network services virtualization, including virtualized Base Stations, is becoming more important. As a result, there is a growing need to manage the temporal patterns of virtualized software in order to make it more predictable.

KEYWORDS: Cloud Computing, Infrastructure, Network, Technology, Virtualization.

1. INTRODUCTION

In essence, both the real-time and cloud infrastructure communities want to ensure that allocated resources or contractual service standards are met between users and the implementation platform, or between virtual machines and the platform. However, a cloud-computing expert who usually comes from the network and distributive systems arena or system software may have a different perspective than a real-time systems expert. As a result, we offer some conceptual clarity and have developed a word mapping [1]. Virtualization is becoming more important in real-time and embedding systems design to provide flexibility and separation across separate run-time environments.

At the platform level, it is necessary to develop direct methods to convert cloud service level agreements (SLAs) to resource assignment and enforcement. SLAs usually deal with platform resources like as bandwidth, latency, and memory. A real-time viewpoint, on the other hand, must enter the picture to offer actual execution methods that ensure and enforce execution and temporal isolation in order to satisfy the specified requirement of real-time applications. Virtualization of resources should be kept at a similar distance from application execution and management of computing resources at the platform level. Providing an integrated resource management framework, particularly for resources in the form of networks, may be particularly challenging in this area. In this vein, network services virtualization, including virtualized Base Stations, is becoming more important. As a result, there is a growing need to manage the temporal behavior of virtualized software in order to make it more predictable. To put it another way, there is a growing need for real-time virtualisation methods that enable the creation of real-time virtual computers (RT-VMs)[2]–[5].

1.1 Execution platform control and access:

RT-VMs are an important technology, which enables virtualized applications to satisfy QoS limitations (in terms of, for example, throughput, latency, memory, computational, and connection capabilities) as specified in contractual agreements between customers and providers, which are codified in appropriate SLAs. Cloud computing is still in its infancy, with best-effort methods dominating the landscape despite their challenges in establishing predictable behavior. As a result, real-time and QoS standards for cloud apps are less stringent, and QoS in actual SLAs is virtually non-existent. However, the situation is quickly changing, and the full potential of cloud platforms will only be realized if real-time design methods are widely used, allowing for predictable cloud software execution. With more and more engaging services accessible on-line, this is anticipated to be driven by increasing and rapidly changing consumer demands and expectations[6], [7]. The function and design of a real-time hypervisor are particularly important in integrating real-time in virtualization. The following are the particular difficulties of such a task:

- The hypervisor handles interrupt translation. Hardware interrupts are converted to software interrupts and sent to the operating system, which must determine the priority of the interrupts. Once an interrupt has been queued for delivery to a VM, the hypervisor scheduler must determine when to schedule the

VM so that it can handle the waiting interrupt (s). This may happen after a few tens of milliseconds if there are more ready-to-run VM virtual CPUs than accessible physical CPUs.

- Time availability, Applications that need to manage time should have access to timers and clocks. It is also required by guest actual working systems to execute traditional resource management functions including resource accounting, enforcement, and prediction. However, the hypervisor's scheduling of VMs may have a significant impact on guest operating systems' perceptions of time, resulting in more unpredictable behavior.
- When dealing with multi-processor or multi-core VMs, the hypervisor may schedule the numerous virtualized CPUs (vCPUs) on the underlying physical CPUs in such a manner that the vCPUs do not proceed at the same pace. In addition, not all vCPUs will be scheduled at the same time. This may introduce additional challenges that must be addressed in the software design; for example, in a guest operating system kernel, a spin-lock intervention is attempted, assuming that the other vCPUs will free the lock in a few milliseconds, when it has actually been scheduled out by the hypervisor.
- Unpredictability of network performance is one of the most difficult problems in distributed real-time virtualized settings, particularly in the presence of virtualized networking setups. Tuning a hypervisor configuration to meet specific performance objectives for virtualized network functions, such as those described in [83], is an intriguing area of research.

VMs must be scheduled for operation in the same manner that OS systems schedule threads. The hypervisor (or similar virtualization infrastructure) schedules the execution of the VMs using scheduling characteristics that are usually specified statically. A RT-VM will adjust its characteristics according to the real-time parameters required or the QoS criteria for real-time scheduling so that it is scheduled ahead of others. This may be coupled with hierarchical scheduling algorithms/policies that create time divisions to execute groups of tasks that might be VMs in real-time scheduling theory. ARINC-653 defines temporal divisions for virtual machines running on the same physical platform to prevent temporal and spatial interference[8]. The hypervisor is thus in charge of managing the hardware resources and enforcing the spatial and temporal separation of the guest operating systems in cloud computing for real-time embedded settings. As a result, para-virtualization is the approach that best suits the needs of embedded real-time systems. It has a quicker execution time and a more user-friendly interface. In the case of para-virtualization, the guest operating system must be modified[9].

1.2 Resource management and real-time scheduling:

For real-time tasks, virtualization poses a variety of challenges. First, since the functionality of each virtual machine (VM) is dependent on the amount of resources (e.g., compute, storage, or networking) used by other VMs, the increasing degree of resource sharing across different operating systems makes it harder to execute software in predictable ways. Worse, numerous VMs using the same physical hosts and connections are often hosted on behalf of distinct and independent tenants in a cloud computing environment (customers). Furthermore, dynamic VM formation and migration generates potentially large and bursty workloads, which may significantly degrade the performance of VMs that share the same physical resources. This makes the issue of temporal interferences across VMs even more essential and crucial, since the performance of one VM is dependent not only on the burden imposed by the same tenant inside the cloud infrastructure, but also on the workload imposed by VMs from other tenants. This scenario, although tolerable in the early stages of cloud computing, is not likely to continue long, as users develop trust in the technology and their still rudimentary needs change. Some of the most significant contributions to real-time scheduling and resource management for QoS-sensitive virtualized workloads are listed below[10].

1.2.1 Resource management in a distributed environment:

Real-time scheduler managing techniques and algorithms in general may offer temporal execution guarantees. A number of contributions on resource management designs, such as emerged for centralized systems as a way to incorporate QoS-based resource management to trade-off execution quality by the allocated resources. Specific methods based on QoS levels for real-time job management have also been developed. Figure 1 shows the increased complexity in the architectural definition of resource managers (RMs).

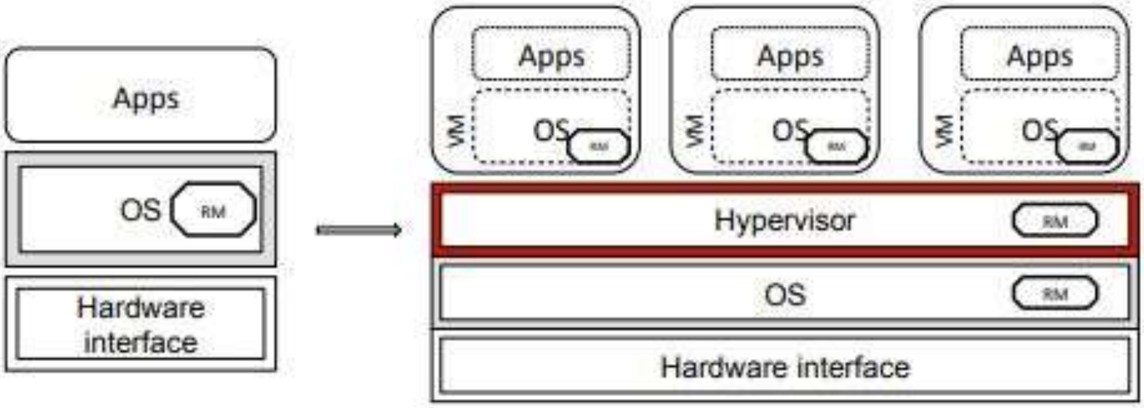

Figure 1: the above figure shows the increased complexity in the architectural definition of resource managers (RMs).

The presence of a resource broker that ensures application resources and contracts is clearly shown in these contributions. Resource managers operate in a partly centralized/partially distributed manner in dispersed settings. All of these concepts must be tweaked to work with real-time virtualization technologies. There is no one place where the resource manager may be found. In order to determine the architectural arrangement of the various modules of a resource-provisioning manager that is also distributed within the node itself, it will have to depend on the concepts of hierarchical scheduling and the related realizations like ARINC-653.

The shift to a more sophisticated design in the architecture of resource management, i.e., the placement of schedulers. The virtual machines' temporal divisions will be defined and arbitrated by the host operating system's scheduler (or partitions). There will be a local scheduler in guest OSs that, in the case of real-time systems, must be designed to be synced with the host/global scheduler. Actual implementations may vary depending on a variety of factors, such as the location of the scheduling queues, access to the processor time, which varies depending on the kind of hypervisor utilized, and so on.

1.2.2 Virtual machine scheduling in real time:

Using appropriate scheduling algorithms at the virtualization or host operating system level, the issue of performance isolation in cloud computing, particularly the one of managing interferences at the CPU level may be partly addressed, recommends using an EDF-based scheduling method for Linux on the host to schedule virtual machines in order to isolate virtualized software on the CPU level (VMs). Unfortunately, the authors utilize a scheduler integrated inside a user-space process (VSched), which results in excessive context switch overheads. Furthermore, in the presence of a VM that blocks and unblocks, such as owing to I/O, VSched cannot effectively provide temporal isolation. The author had looks at virtual machine performance isolation, focusing on the use of different scheduling rules offered in the Xen hypervisor. Dunlap also suggested a number of improvements to the Xen type of payment in order to solve problems with temporal isolation and fairness among the CPU shares allocated to each VM. Specifically, focuses on automatic on-line CPU allocation modification in order to keep VM performance constant. In order to execute the required QoS control loops, the framework must go beyond the standard IaaS business model, requiring application-specific metrics. Furthermore, the authors do not discuss how dynamic resource distribution is seen from the viewpoint of resource planning and prior reservations. In contrast to the heuristic-based methods described before, using rigorous real-time scheduling algorithms to schedule virtual machines is a potential approach.

1.2.3 Alternatives that are not dependent on virtualization of machines:

A number of papers recommended a disruptive strategy, severing ties with today's practice of virtualized cloud infrastructures and suggesting entirely new methods to provide real-time performance in distributed cloud infrastructures. These are based on very distinct ideas about software architecture. Several writers have stated that, in comparison to today's well-established virtualized infrastructures, lighter software designs enabling cloud applications are required. In today's world, it is common to find functionality replicated at several levels, such as CPU scheduling, packet filtering/routing, memory allocation, and security features at the hypervisor and guest operating system layers. This causes inertia in the software component management, reducing agility, flexibility, and, in certain cases, performance. For instance, Media Cloud has been suggested as a new platform for multimedia processing. It is built on the concept of very light and almost stateless media processing components, whose placement may be rapidly determined and changed at runtime for specific multimedia flows between sources and destinations.

For example, according to, Media Cloud can support the instantiation of media processing functions, i.e., service components on distributed cloud resources, in a few milliseconds, and the reassignment of media processing components from one processing resource to another at service run-time in 2 to 3 milliseconds. Such adaptability, made possible by the platform's exclusive emphasis on the concept of an application-specific cloud, is unquestionably a potential enabler for real-time workloads and adaptive resource management strategies. Similar reasons may be found in, which proposes the new Osprey Operating System for predictable cloud computing, which envisions a revolutionary operating system architecture for cloud computing. In addition, the newly proposed Mirage project, which revisits the concept of library operating systems for implementing high-performance network applications for cloud computing and mobile platforms with a substantially reduced software stack optimized at compile-time, follows a similar line of thinking. Because of the smaller size of these cloud apps, they can be more nimble and responsive in terms of changing the system at runtime to address performance problems that would be difficult with traditional VMs.

Finally, emphasizes that it is inevitable to lose processing resources in the hierarchical composition of several real-time schedulers in order to provide schedulability guarantees for hosted real-time workloads. Alternatively, the authors suggest that OS-level virtualization, such as that provided by Linux containers (LXC), could be a useful way to maintain the most important benefits of virtualization (such as isolation from a security standpoint), while achieving schedulability of the overall real-time workload in a simpler and more efficient manner. There are varieties of additional situations in which OS-level virtualization is being studied, such as network function virtualization and others, but a comprehensive explanation is beyond the scope of this article. Finally, some of the most difficult challenges in cloud computing are due to the need to provide SaaS offerings with stable and predictable service levels, which requires the interaction of a large number of business players, including multiple SaaS and IaaS providers, as well as Network Service Providers. Establishing appropriate SLA models among various stakeholders may be a problem in and of itself, including both technical and economic issues that need to be addressed in the future.

1.3 Network communication difficulties:

In real-time systems, the networking is a generator of temporal unpredictability. Theoretical models of real-time networking calculate message exchange response times and overall system schedulability based on the assumptions that the network is highly predictable, no messages are lost, noise effects are absent, wired technology is used, and the system is closed so that no dynamic interference caused by additional tasks or applications can be considered once the system is closed. The time-triggered design is used in most real-time networks for critical situations. A similar concept has been effectively used in a variety of areas, most notably in the automobile industry. Proprietary protocols, such as the 1535 bus-based aerial data transmission systems, have been utilized in various application fields such as avionics, depending on business standards and target system field (i.e., civil, military, or aerospace). The nature of network traffic and networking protocols in cloud computing settings displays a particular behavior, making complete prediction impossible. Instead, Quality of Service (QoS) provisions are used.

2. DISCUSSION

This article examined some of the issues and difficulties associated with achieving real-time virtualization as a first step toward providing an idealized map of the current situation, outlining the components required at all levels to make it a reality. The issues discussed include hypervisor structure and function, various kinds of virtualization and their percentile rank, general resource management problems and VM schedulability in relation to hierarchical scheduling, and the critical role of the network in virtualization technology. For the latter, this article has presented several HPCC community methods for bypassing inefficiencies caused by protocol execution inside the various levels of the software stack. To map these domains, a vocabulary mapping between cloud and decentralized real-time systems has been established. Finally, the author have discussed a few potential approaches that will need attention from many groups in order to achieve the real-time cloud.

3. CONCLUSION

To provide flexibility and isolation across separate run-time environments, cloud computing has become more and more important and embedding systems design. Direct methods to translate data center service level agreements (SLAs) to resource undertaking and enforcement must be developed at the platform level. Station resources like as traffic, latency, and memory are typically covered by SLAs. In order to satisfy the delay requirements of real-time applications, a real-time viewpoint must enter the picture to provide actual method of execution that ensure and enforce successful implementation and temporal isolation. At the platform level, resource virtualization should be kept at a narrow angle from assessment execution and resource management.

In this area, providing an efficient production guideline, especially for resources in terms of connections, may be particularly difficult. Virtualization of network services, including virtual machines Cell Towers, has become more important in this regard.

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.