

(An ISO 3297: 2007 Certified Organization) Website: <u>www.ijircce.com</u> Vol. 4, Issue 12, December 2016

Heuristic Based Mapping of Workflows and VM Allocation in Cloud Systems

Rekha A Kulkarni, Dr. Suhash. H Patil, Dr N. Balaji

Asst. Professor, PICT, Pune, India

Professor, BVU, COE, Pune, India

Professor, JNTUK Vijaynagarum, Andhra Pradesh, India

ABSTRACT: Advancements in networking and hardware design areas have become responsible for advanced computing paradigms like utility computing and cloud computing. In utility computing the hardware and software resources are concentrated in large data centers and users can pay as they consume computing, storage and communication resources. Cloud computing is a way of having utility computing with the help of large amount of resources and service providers.

The uncertainties in the cloud are imposed by the Computational Power and the Communication Cost. In real cloud computing environments, task execution times usually cannot be reliably estimated, and the actual values are available only after tasks have been completed. This may be contributed to the following reasons. Tasks usually contain conditional instructions under different inputs. This can be interpreted that tasks in parallel applications may contain multiple choice and conditional statements, which will lead to different programbranches and loops. Different branches or loops make the task computation with large differences, which will leaddirectly to the same task in the case of different data input may also lead to different task execution times. Tasks to be executed parallel the dependencies among the tasks have to known with the help of directed acyclic graph.TheVMs' performance in clouds varies over the time. This can be contributed to the following fact. With the advanced virtualization technology, multiple VMs can simultaneously share the same hardware resources (e.g., CPU, I/O, network, etc.) of a physical host. Such resource sharing may cause the performance of VMs subjecting to considerable uncertainties mainly due to resource interferencebetween VMs.

In this paper we study the impact of task and VM uncertainties on scheduling methods and propose the usages of heuristic approach to improve VM allocation to satisfy the timing requirements of workflows.

KEYWORDS: Virtualization, Real time scheduling, VM allocation, Heuristic, DAG.

I. INTRODUCTION

Cloud computing provides a distributed computing environment in which there is a pool of virtual, dynamically scaleable heterogeneous computing and storage platforms. The computing power and the storage are provided as Services on demand to external user over the internet. Cloud computing aims at making computing as a service whereby shared resources, software, and information are provided to users who are requesting the service as a utility over a network. Basic services are rented by users for their complex applications with various resource requirements which are usually modeled as workflows. Better services imply higher costs. Services are consumed based on Service-Level Agreements, which define parameters of Quality of Service in terms of the pay-per-use policy.

The cloud provides the services in terms of three delivery models SaaS software as a service IaaS infrastructure as a service PaaS Platform as a service. Deployed as public private and hybrid clouds. In the heterogeneous environment of cloud handling real time tasks poses challenges of processor scheduling and load balancing to meet the deadlines. Implementation of real-time tasks poses a lot of problems due to the unpredictability of the tasks involved and due to the lack of complete task knowledge prior to the execution process.



(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

Virtualization: In cloud VM are responsible for execution of the given task. The workflow which has number of tasks has to be run using the VMs which are available or provided by the service provider. The tasks have to be mapped to VMs.

The cloud paradigm offers advantages through reduction of operation costs, server consolidation, flexible system configuration and elastic resource provisioning. One of the most important technologies that enabled this paradigm shift in computing is virtualization, and particularly machine virtualization. Machine virtualization (also referred to as processor virtualization) allows a single physical machine to emulate the behavior of multiple machines, with the possibility to host multiple and heterogeneous operating systems (called guest operating systems or guest OSs) on the same hardware. A virtual machine monitor (VMM), or hypervisor, is the software infrastructure running on (and having full control of) the physical host and which is capable of running such emulation.

Real-time scheduling of virtual machines: The problem of performance isolation in cloud computing, and especially the one of controlling the interferences at the computing level, can be partially mitigated by using proper scheduling algorithms at the hypervisor or host operating system level. [19] Concerning the isolation of virtualized software on the computing level, proposes to use an EDF-based scheduling algorithm for Linux on the host to schedule virtual machines (VMs). Unfortunately, the authors make use of a scheduler built into a dedicated user-space process (VSched), leading to unacceptable context switch overheads. Furthermore, VSched cannot properly guarantee temporal isolation in presence of a VM that blocks and unblocks, e.g., as due to I/O. investigates the performance isolation of virtual machines, focusing on the exploitation of various scheduling policies available in the Xen hypervisor. Furthermore, Dunlap proposed variousenhancements to the Xen credit scheduler in order to address various issues related to the temporal isolation and fairness among the CPU share dedicated to each VM. Precisely, focuses on automatic on-line adaptation of the CPU allocation in order to maintain a stable performance of VMs. The framework needs to go beyond the common IaaS business model, in that it needs application-specific metrics to run the necessary QoS control loops. Also, the authors do not address how the dynamic resource allocation is considered from a resource planning and advance reservation perspective. In contrast to heuristics- based solutions discussed above, a promising approach is to employ rigorous real-time scheduling algorithms to schedule virtual machines. We provide an overview of some ideas on one of the key problems in cloud computing such as the improvement of the virtualization technology to offer strict guarantees over the contracted resources. One other challenge will be the real-time live migration where VMs containing real-time applications are transferred between different physical servers. Though there are many parameters or constraints involved in practical workflow scheduling settings, deadline and time slot are two crucial ones in cloud computing.

II. BACKGROUND

Virtualization Technologies :

[20] One of the main technologies used in the Cloud Computing is virtualization. Clouds leverage various virtualization technologies (e.g., machine, network, and storage) to provide users an abstraction layer that provides a uniform and seamless computing platform by hiding the underlying hardware heterogeneity, geographic boundaries, and internal management complexities.

From architectural perspective, virtualization approaches are categorized into the following two types: Hosted Architecture and Hypervisor-Based Architecture. In the hosted architecture The virtualization layer is installed and run as an individual application on top of an operating system and supports the broadest range of underlying hardware configurations. Example of such architecture includes VMware and VirtualBox.

In the Hypervisor-Based Architecture: The virtualization layer, termed Hypervisor is installed and run on bare hardware and retains full control of the underlying physical system. It is a piece of software that hosts and manages the VMs on its Virtual Machine Monitor (VMM) components. The VMM implements the VM hardware abstraction, and partitions and shares the CPU, memory, and I/O devices to successfully virtualize the underlying physical system. In this process, the Hypervisor multiplexes the hardware resources among the various running VMs in time and space sharing manner. VMware ESXi ,Xen Server and KVM are examples of this kind of virtualization. Since Hypervisors have direct access to the underlying hardware resources rather than executing instructions via operating systems as it is



(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

the case with hosted virtualization, a hypervisor is much more efficient than a hosted virtualization system and provides greater performance, scalability, and robustness.

The Industry and research community have come up with the following three types of alternative virtualization techniques:

Full Virtualization: This type of virtualization technique provides full abstraction of the underlying hardware and facilitates the creation of complete VMs in which guest operating systems can execute. Full virtualization is achieved through a combination of binary translation and direct execution techniques that allow the VMM to run in Ring 0.

Paravirtualization: Different from the binary translation technique of full virtualization, Paravirtualization works through the modification of the OS kernel code by replacement of the non-virtualizable instructions with hyper calls that communicate directly with the hypervisor virtualization layer.

Hardware Assisted Virtualization: In response to the success and wide adaptation of virtualization, hardware vendors have come up with new hardware features to help and simplify virtualization techniques. Intel Virtualization Technology (VT-x) and AMD-V are first generation virtualization supports allow the VMM to run in a new root mode below Ring 0 by the introduction of a new CPU execution mode. With this new hardware assisted feature, privileged and critical system calls are automatically trapped by the hypervisor and the guest OS state is saved in Virtual Machine Control Structures (VT-x) or Virtual Machine Control Blocks (AMD-V), removing the need for either binary translation (full virtualization) or paravirtualization. KVM also provides full virtualization with the help of hardware virtualization support. It is a modification to the Linux kernel that actually makes Linux into a hypervisor on inserting a KVM kernel module. One of the most interesting KVM features is that each guest OS running on it is actually executed in user space of the host system. This approach makes each guest OS look like a normal process to the underlying host kernel.

Virtual Machine Migration Techniques:One of the most prominent features of the virtualization system is the VM Live Migration (Clark et al., 2005) which allows for the transfer of a running VM from one physical machine to another, with little downtime of the services hosted by the VM. Almost all the modern virtualization environments offer VM live migration feature, including Xen Server, VMware ESX Server (through VMotion (Nelson, Lim, & Hutchins, 2005)), KVM, Microsoft Hyper-V, Oracle VM VirtualBox, and OpenVZ.

III. RELATED WORK

Cloud computing provides a distributed computing environment in which there is a pool of virtual, dynamically scaleable heterogeneous computing and storage platforms. The computing power and the storage are provided as Services on demand to external user over the internet. Cloud computing aims at making computing as a service whereby shared resources, software, and information are provided to users who are requesting the service as a utility over a network. One of the key technologies which plays an important role in Cloud data-center is **resource scheduling**. Real time processing on the cloud systems requires satisfying timing constraints of real time systems. In the scenario of time dependent application deadline misses are undesirable. One of the challenging scheduling problems in Cloud data center is to consider allocation and migration of re-configurable virtual machines (VMs) and integrating features of hosting physical machines.

ShuoLiu,Gang Quan and Shangpin Ren [1] have discussed about The real-time tasks scheduled non-preemptively with the objective to maximize the total utility. The most unique characteristic of their approach is that, different from the traditional utility accrual approach that works under one single time utility function (TUF), have two different TUFs—a profit TUF and a penalty TUF—associated with each task at the same time, to model the real-time applications for cloud computing that need not only to reward the early completions but also to penalize the abortions or deadline misses of real-time tasks.

Ji wang, WeidongBao [2] have discussed about a fault-tolerant mechanism which extends the primary-backup model to incorporate the features of clouds. They have proposed an elastic resource provisioning mechanism in the fault-tolerant context to improve the resource utilization. On the basis of the fault-tolerant mechanism and the elastic resource provisioning mechanism, designed novel fault-tolerant elastic scheduling algorithms for real-time tasks in clouds named FESTAL, aiming at achieving both fault tolerance and high resource utilization in clouds.



(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

experiments injecting with random synthetic workloads as well as the workload from the latest version of the Google cloud tracelogs are conducted by CloudSim to compare FESTAL with three baseline algorithms, i.e., Non-Migration-FESTAL (NMFESTAL), Non-Overlapping-FESTAL (NOFESTAL), and Elastic First Fit (EFF).

Carlos Vázquez and Ginés Moreno [3] Cloud Scheduler Assisted by a Fuzzy Affinity-Aware Engine. Recent advances on declarative paradigms have introduced expressive resources based on fuzzy logic. These are very useful for increasing the flexibility of the so-called fuzzy logic programming paradigm, resulting in highly expressive languages where the treatment of uncertainty and approximate reasoning is performed in a natural, efficient way. Therefore, making It extremely useful for decision-making scenarios dealing with high levels of uncertainty.

Bruno Moura*, Yan Soares*, Leticia Sampaio*, Renata Reiser*, Adenauer Yamin*and Mauricio Pilla [4] the authors have studied the necessity of making a system robust to the uncertainties of the different measures taken from the computing infrastructure in Grids. In particular, taking relevant information to tasks scheduling from a set of heterogeneous sources and dynamic behaviour has been the focus of a significant number of studies., Their paper aims to approach a module for task scheduling in Grid Computing, using a taken decision system under uncertainty regime. The proposed module uses the fuzzy logic, called fGrid, to treat uncertainties.

In this paper, we are proposing a heuristic based on the dynamic data to have the mapping of tasks with the available VMs and in what way the VM live migration can affect the meeting of deadlines in soft real time tasks like video streaming.

IV. PROPOSED SYSTEM

In the real-time processing of workflows on the cloud systems meeting of deadlines possesses two uncertainty issues. First one is because ofthe uncertainty incurred in the computations times of tasks and the second one is because of the sharing of resources among the VMs which may require VM live migrations to be carried out during workflow execution. In our approach, we are trying to solve these uncertainties with the help of heuristics in order to meet the deadline of the task as QOS parameter to be considered as a part of SLA.

During VM live migration process, additional network traffic is generated during the whole migration period since hypervisor need to transfer in-memory states of the running VM to the target machine. Furthermore, VM migration causes unavailability of hosted applications due to the VM downtime factor. As a consequence, VM living migration is identified as an expensive data center operation that should not be triggered very often. While network-aware VM migration strategies opt for optimizing overall network usage and reduce the inter-VM communication delays through migrating communicating VMs into nearly hosts, most of the strategies do not consider the associated VM migration overheads and resulting application performance degradation.

VM placement problem is in fact an NP-complete problem since it requires combinatorial optimization to achieve the goals. As a consequence, most of the research works attempt to solve the problem through heuristic methods so that the algorithms terminate in a reasonable amount fime. Such heuristics are not guaranteed to produce optimal placement decision; however, from time constraints perspective exhaustive search methods that guarantee the generation of optimal solutions are not practical, especially considering the scale of modern data centers.

No physical node with enough resource is found to host a VM and No physical path with enough bandwidth is found to be allocated. VM placement strategies utilizing both VM resource requirements information and inter-VM traffic load can come up with placement decisions that are more realistic and efficient. resource contention will have adverse effects on application performance, thus leading to SLA violations and profit minimization. Therefore, it is important to understand the behavior and resource usage patterns of the hosted applications in order to efficiently place VMs and allocate resources to the applications. Utilization of historical workload data and application of appropriate load prediction mechanisms need to be integrated with VM consolidation techniques to minimize resource contentions among applications and increase resource utilization and energy efficiency of data center.



(An ISO 3297: 2007 Certified Organization)

Website: <u>www.ijircce.com</u>

Vol. 4, Issue 12, December 2016

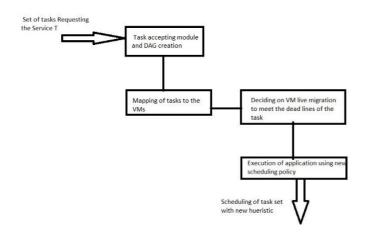


Fig 1 : Architecture model of proposed system

V. ALGORITHM STEPS OF THE PROPOSED SYSTEM

- Step 1 : Accept the workflow w consisting of set of tasks
 - $W = \{t1, t2, ..., tn\}$
- Step 2: Generation of DAG to identify tasks dependencies.
- Step 3: Find suitable VM configuration to map the tasks.
- Step 4: Check for VM live migration using the heuristic to meet the deadlines.
- Step 5: Execute the application or the workflow on the VMs.

Step 6. Check for the performance analysis of the heuristic.

VI. EXPERIMENTAL SETUP AND RESULTS

For our experimental setup we are using OpenNebula Platform as cloud platform to set up private cloud and Amazon EC2 for public cloud services. OpenNebula has a flexible architecture and is easy to customize, and it provides a set of tools and service interfaces that are handy for the integration. We have also integrated with other Cloud plat-forms such as Amazon EC2 **OpenNebula** is a fully open-source toolkit to build IaaS private, public and hybrid Clouds, and a modular system that can implement a variety of Cloud architectures and can interface with multiple datacenter services. OpenNebula provides storage, network, virtualization, monitoring, and security technologies to deploy services as virtual machines on distributed infrastructures, combining both datacenter resources and remote Cloud resources, according to allocation policies.

VII. CONCLUSION

Meeting the deadline for Resource and Timing Constraints is a Challenge in a cloud system. While working with shared environment users will compete for resources to meet their timing deadlines. In such a scenario, resource scheduling plays a critical part in meeting an application's expectations. In large clusters, tasks complete at such a high rate that resources can be reassigned to new jobs on a timescale much smaller than job duration.



(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

The job to be scheduled next according to fairness might not have data on the nodes that are currently free. In a heterogeneous environment like cloud where processing speeds are likely to vary among nodes, high processing nodes are expected to complete more tasks. In such a scenario, the concept of equally distributing data in order to provide data locality is likely to create network bottleneck. For heterogeneous systems, effective data placement strategies are required in order to ensure efficient task scheduling. So the proposed system tries to select a scheduler at run time which can meet the timing requirements based on the information available locally and globally.

With the help of heuristic, it is possible to address the uncertainties involved while dealing with large number of processors. With the help of configuration changes to OpenNebulait is possible to create an infrastructure for meeting the deadlines in real time tasks.

REFERENCES

[1] Shuo Liu Gang Quan, Shangping Ren, "On-line Scheduling of Real-time Services for Cloud Computing ",2010 IEEE 6th World Congress on Services.

[2] Ji Wang, WeidongBao, Xiaomin Zhu, Laurence T. Yang, andYang Xiang, Senior Member, IEEE" FESTAL: Fault-Tolerant Elastic Scheduling Algorithm for Real-Time Tasks in Virtualized Clouds" IEEE transactions on computers, vol. 64, no. 9, september 2015

[3] Carlos Vázquez and Ginés Moreno Luis Tomás and Johan TordssonA" Cloud Scheduler Assisted by a FuzzyAffinity-Aware Engine",

[4] Bruno Moura*, Yan Soares*, Leticia Sampaio*, Renata Reiser*, Adenauer Yamin*and Mauricio Pilla" fGrid: Uncertainty Variables Modeling for Computational Grids using Fuzzy Logic".

[5]K. Liu, H. Jin, J. Chen, X. Liu, D. Yuan, and Y. Yang, "A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost- Constrained Workflows on Cloud Computing Platform." J. of High Performance Computing Applications, 24:445–456, Nov 2010.

[6]J. Gu, J. Hu, T. Zhao, and G. Sun, "A new resource scheduling strategy based on genetic algorithm in cloud computing environment," J. Comput., vol. 7, no. 1, pp. 42-52, 2012.

[7]A.Oprescu and T. Kielmann, "Bag-of-Tasks Scheduling under Budget Constraints." in 2nd International Conference on Cloud Computing, Los Angeles, USA, 2009.

[8]A.Oprescu and T. Kielmann, "Bag-of-Tasks Scheduling under Budget Constraints." in 2nd International Conference on Cloud Computing, Los Angeles, USA, 2009.

[9]S. H. Jang, T. Y. Kim, J. K. Kim and J. S. Lee, "The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing," IJCA, vol. 5, no. 4, pp. 157-162, December 2012.

[10]H. Ye, J. Wong, G. Iszlai and M. Litoiu, "Resource Provisioning for Cloud Computing," Proceedings of CASCON 2009, November 2009.

[11]T.Henzinge, A. V. Singh, V. Singh, T. Wies and D. Zufferey, "Flex-PRICE: Flexible Provisioning of Resources in a Cloud Environment," In CLOUD, 2010.

[12]M. Paul and G. Sanyal, "Task-scheduling in cloud computing using credit based assignment problem," International Journal on Computer Science and Engineering (IJCSE) 3.10, pp. 3426-3430, 2011.

[13]S.Xue and W. Wu, "Scheduling workflow in cloud computing based on hybrid particle swarm algorithm," TELKOMNIKA Indonesian Journal of Electrical Engineering 10.7, pp.1560-1566, 2012.

[14]X. Wu, M. Deng, R. Zhang, B. Zeng and S. Zhou, "A task scheduling algorithm based on QoS driven in cloud computing," Advances in Engineering Software, Procedia Computer Science, 17, pp. 1162-1169, 2013. [15]B. M. Xu, C. Y. Zhao, E. Z. Hu and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," Advances in Engineering

Software, 42(7), 419-425, 2011.

[16]I. A. Moschakis and H. D. Karatza, "Performance and cost evaluation of Gang Scheduling in a Cloud Computing system with job migrations and starvation handling," In: Proceedings of IEEE Symposium on Computers and Communications (ISCC), pp. 418-423, 2011.

[17]S.Ghanbari and M. Othman, "A priority based job scheduling algorithm in cloud computing," Procedia Engineering, 50, pp. 778-785, 2012.

[18]L. F. Bittencourt and E. R. M. Madeira, "HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds"

[19] Marisol García-Valls a, ît, Tommaso Cucinottab, Chenyang Lu" Challenges in real-time virtualization and predictable cloud computing", Journal of Systems Architecture 60 (2014) 726-740.

[20] MdHasanulFerdaus Rodrigo N. CalheirosManzurMurshedRajkumarBuyya, "Chapter 2 DOI: 10.4018/978-1-4666-8213-9.ch002 Network-Aware Virtual Machine Placement and Migration in Cloud Data Centers