Resource Allocation to Minimize Cost in Cloud System

M.Markco[1] ,D.Radha[2], K.Raju[3]

[1]Assistant Professor, [2]Assistant Professor, [3]PG Scholor
[1] E.G.S.Pillay Engineering College, [2]SembodaiRukmaniVaratharajan Engineering College, [3]Madha Engineering College
[1]Nagapattinam,[2]Sembodai,[3]Chennai

Abstract: *This project virtual machine (VM) technology being increasingly mature, compute resources in cloud systems can be partitioned in fine granularity and allocated on demand. By formulate a deadline-driven resource allocation problem based on the cloud environment facilitated with VM resource isolation technology, and also propose a novel solution with polynomial time, which could minimize users' payment in terms of their expected deadlines. By analyzing the upper bound of task execution length based on the possibly inaccurate workload prediction, further propose an error-tolerant method to guarantee task's completion within its deadline. Validate its effectiveness over a real VM-facilitated cluster environment under different levels of competition.*

Keywords: *virtual machine, Optimal Resource Allocation, Earliest Deadline First.*

1. INTRODUCTION

Cloud computing platforms are rapidly emerging as the preferred option for hosting applications in many business contexts. Starts up companies are relying on public cloud infrastructures to deploy their applications, which help reducing their initial costs. Larger companies are also adopting clouds, either public clouds for expanding their existing infrastructures or rapid deployment of test environments, or private clouds for dynamic on-demand provisioning of virtual resources among their internal divisions. All the resources provisioned by cloud system are supposed to be under a payment model. Each task's workload is likely of multiple dimensions. First, the compute resources in need may be multi-attribute (such as CPU, disk-reading speed, network bandwidth, etc.), resulting in multidimensional execution in nature. Second, even though a task just depends on one resource type like CPU, it may also be split to multiple sequential execution phases, each calling for a different computing ability and various price on demand, also leading to a potentially high-dimensional execution scenario.

2. LITERATURE REVIEW

In the elasticity of a utility matches the need of businesses providing services directly to customers over the Internet, as workloads can grow (and shrink) far faster than 20 years ago.. From the cloud provider's view, the construction of very large datacenters at low cost sites using commodity computing, storage, and networking uncovered the possibility of selling those resources on a pay-as-you-go model below the costs of many medium-sized datacenters, while making a profit by statistically multiplexing among a large group of customers. From the cloud user's view, it would be as startling for a new software startup to build its own datacenter as it would for a hardware startup to build its own fabrication line Cloud Computing users, relieved of dealing with the twin dangers of over-provisioning and under-provisioning our internal datacenters.[1]

The Clouds do not have a clear and complete definition in the literature yet, which is an important task that will help to determine the areas of research and explore new application domains for the usage of the Clouds. To tackle this problem, the main available definitions extracted from the literature have been analyzed to provide both an integrative and an essential Cloud definition. Although their encompassing definition is overlapped with many grid concepts Virtualization is the key enabler technology of Clouds, as it is the basis for features such as, on demand sharing of resources, security by isolation, etc. Usability is also an important property of Clouds. Also, security enhancements are needed so that enterprises could rely sensitive data on the Cloud infrastructure. Finally, QoS and SLA enforcement will also be essential before ICT companies reach high levels of confidence in the Cloud. NGG and

ISSN: 2278 – 1323

OGF efforts are highly devoted to this task, enforcing standardization to enable a Cloud federation that can then deal with the required massive scalability.[2]

The issue of isolation from misbehaving VMs is an important one to consider, especially for a commercial hosting environment. Xen,OpenVZ and Solaris Containers. Their results highlight differences between major classes of virtualization systems – full virtualization like VMware Workstation, para virtualization like Xen and operating system level virtualization like Solaris Containers and Open VZ. Full virtualization completely protected the well-behaved VMs in all of our stress tests. Para virtualization offers excellent resource isolation as well. In their Xen tests, the well-behaved VMs suffered at most 1.7% degradation for the disk intensive test with many other tests showing only slight, but repeatable degradation. With operating system level virtualization the need for resource controls, either as a default or through proper configuration, was clear. Without them, well-behaved and misbehaving workloads both suffered. Strong resource isolation clearly can be added to operating system level virtualization. As in the case of Solaris and OpenVZ, the operating system can be modified to implement new resource scheduling algorithms that enforce resource isolation across VMs.[3]

There has been much prior work on task scheduling that considers resource requirements that address how much resource tasks consume. By addresses the performance impact of task placement constraints. Task placement constraints impact which resources tasks consume. Task placement constraints, such as characteristics specified by the Condor Class Ads mechanism, provide a way to deal with machine heterogeneity and diverse software requirements in compute clusters. Experience at Google suggests that task placement constraints can have a large impact on task scheduling delays. Although our data is obtained from Google compute clusters, the methodology that we develop is general. In particular, the UM metric applies to any compute cluster that employs a Class Ads style of constraint mechanism. [4]

3. OPTIMAL RESOURCE ALLOCATION

Traditional job scheduling is often formulated as a kind of combinatorial optimization problem or queue-based multiprocessor scheduling problem. That is, most of the existing deadline-driven task scheduling solutions from single cluster environment confined in LAN to the Grid computing environment suitable for WAN are also strictly subject to the queuing model under which a single machine's multiple resources cannot be further split to smaller fractions at will. Some successful platforms or cloud management tools leveraging VM resource isolation technology include Amazon EC2 and Open Nebula. Traditional optimization problems are often subject to the precise prediction of task's characteristic (or execution property), which is nontrivial to realize in practice. Accordingly, as the state of the art, further analyze our algorithm's optimality approximation ratio given the possibly wrong predictions of tasks' execution properties.

By formulate demand for computing power and other resources as a resource allocation problem with multiplicity, where computations that have to be performed concurrently are represented as tasks and a later task can reuse resources released by an earlier task. By present an algorithm (Optimal Resource Allocation) with a proof of its approximation bound that can yield close to optimum solutions in polynomial time. Enterprise users can exploit the solution to reduce the leasing cost and amortize the administration overhead.

3.1 Load Distribution:

The Process distributed incoming task to available system resources and achieving good load balance in a fully decentralized and heterogeneous cloud environment. Allocate resource for task with its resource requirements that can minimize a task's execution time. Each user needs to precisely predict the execution property (i.e., workload ratio) for task, before constructing the resource allocation with minimized payment for its execution under a user-specified deadline.

3.2 Payment Minimization:

Each task's execution may involve multidimensional resources, such as CPU and disk I/O. Upon receiving the request from user, the scheduler checks the pre collected availability states of all candidate nodes, and estimates the minimal payment of running the task within its deadline on each of them. The host that requires the lowest payment will run the task via a customized VM instance with isolated resources. Specifically, the VM will be customized with such a CPU rate and disk I/O rate that the task can be finished within its deadline and its user payment can also be minimized meanwhile. Finally its computation results will be returned to users.

3.3 Fault Tolerance:

Cloud systems usually do not provision physical hosts directly to users. If the resources provisioned

ISSN: 2278 – 1323

are relatively sufficient, by guarantee task's execution time always within its deadline even under the wrong prediction about task's workload. Each task can be guaranteed to be finished within its original deadline even though task properties cannot be predicted accurately.

4. NEW PROPOSED SCHEME

The optimal solution to the problem with unbounded capacities. Focus on such a question: what is the final upper bound of task execution length as compared to its predefined deadline, when running it using the resource vector allocated with inaccurately predicted workload information .It implement a web service-based prototype that can compute a set of combined matrix operations. Each matrix operation is called by some user task through a web service API and each task is executed in a VM container. Algorithm is evaluated on such a real cluster environment. Users can submit their computation request by editing their mathematical formulas. By make use of Parallel Colt to perform math computations, each consisting of a partially ordered set of operations. Parallel Colt is such a library that can effectively calculate complex matrix-operations like matrix multiply, in parallel via multiple threads.

4.1 System Architecture:

The user that requires the lowest payment will run the task via a customized VM instance with isolated resources. Specifically, the VM will be customized with such a CPU rate and disk I/O rate that the task can be finished within its deadline and its user payment.

The user that requires the lowest payment will run the task via a customized VM instance with isolated resources. Specifically, the VM will be customized with such a CPU rate and disk I/O rate that the task can be finished within its deadline and its user payment. It can also be minimized meanwhile .Finally its computation results will be returned to users, running two VMs that are allocated with half of the total physical resources, so its availability vector. If there are no workloads being executed simultaneously for a particular task, its total execution time will be the sum of the individual processing times on different dimensions.

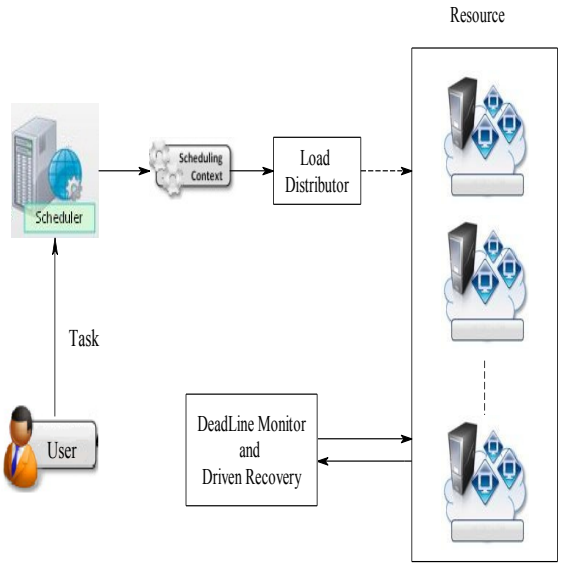

Figure 1: System Design

The user that requires the lowest payment will run the task via a customized VM instance with isolated resources. Specifically, the VM will be customized with such a CPU rate and disk I/O rate that the task can be finished within its deadline and its user payment. It can also be minimized meanwhile .Finally its computation results will be returned to users, running two VMs that are allocated with half of the total physical resources, so its availability vector. If there are no workloads being executed simultaneously for a particular task, its total execution time will be the sum of the individual processing times on different dimensions. If the execution of the workloads overlaps, however, the task's completion time would be shorter. Accordingly, it's final execution .For simplicity, denote task execution time as where denotes a constant coefficient .Load distributor which distribute the resource which is under the control of dead line recovery and monitor.

5. CONCLUSIONS AND FUTURE WORK

An efficient algorithm (Optimal Resource Allocation) to determine the optimal resource, aiming to minimize user's payment on task and also Endeavour to guarantee its execution deadline meanwhile in addition analyze the approximation ratio for the expanded execution time generated by the algorithm to the user-expected deadline, under the possibly

ISSN: 2278 – 1323

inaccurate task property prediction. When the resources provisioned are relatively sufficient, guarantee task's execution time always within its deadline even under the wrong prediction about task's workload characteristic.

The approximation ratio for the expanded execution time generated by our algorithm to the user-expected deadline, under the possibly inaccurate task property prediction. When the resources provisioned are relatively sufficient, By guarantee task's execution time always within its deadline even under the wrong prediction about task's workload characteristic. In the future, plan to integrate our algorithms with stricter/original deadlines into some excellent management tools like Open Nebula, for maximizing the system-wide performance .Some queuing policies like earliest deadline first(EDF) will be studied to further reduce user payment especially in the short supply situation. More complex scheduling constraints like the compatibility and security issue will also be taken into account.

References

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A.Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing, "Technical Report UCB/EECS-2009-28, EECS Dept., Univ.California, Berkeley, Feb. 2009.

[2] L.M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition," SIGCOMMComputer Comm. Rev., vol. 39, no. 1, pp. 50-55, 2009.

[3] D. Gupta, L. Cherkasova, R. Gardner, andA.Vahdat, "Enforcing Performance Isolation across Virtual Machines in Xen," Proc.ACM/IFIP/USENIX Int'l Conf. Middleware (Middleware '06), pp. 342-362, 2006.

[4] J.N. Matthews, W. Hu, M. Hapuarachchi, T. Deshane, D. Dimatos, G. Hamilton, M. McCabe, and J. Owens, "Quantifying the Performance Isolation Properties of Virtualization Systems," Proc.Workshop Experimental Computer Science (ExpCS '07), 2007.

[5] F. Chang, J. Ren, and R. Viswanathan, "Optimal Resource Allocation in Clouds," Proc. IEEE Int'l Conf. Cloud Computing, pp. 418-425, 2010.

[6] S. Di, D. Kondo, and W. Cirne, "Characterization and Comparison of Cloud versus Grid Workloads," Proc. 14th Int'l Conf. Cluster Computing, pp. 230-238, 2012.

[7] H. Khazaei, J.V. Misic, and V.B. Misic, "Modelling of Cloud Computing Centers Using m/g/m Queues," Proc. Int'l Conf. Distributed Computing Systems Workshops (ICDCS), pp. 87-92, 2011.

[8] D. Milojicic, I.M. Llorente, and R.S. Montero,"Opennebula:ACloud Management Tool," IEEE Internet Computing, vol. 15, no. 2,pp. 11-14, Mar./Apr. 2011

[9] V. Petrucci, O. Loques, and D. Mosse´, "A Dynamic Optimization Model for Power and Performance Management of Virtualized Clusters," Proc. First Int'l Conf. Energy-Efficient Computing and Networking (e-Energy '10), pp. 225-233, 2010.

[10] J.E. Smith and R. Nair, Virtual Machines: Versatile Platforms For Systems and Processes. Morgan Kaufmann, 2005.

D.Radha was born in Tamilnadu, India in 1988. She received her B.TECH. degree in Information Technology from Anna University, Chennai in 2009. She completed her M.Tech. (Mainframe Technology) in Anna University, Regional Centre - Coimbatore. She is currently working as an Assistant Professor in the Department of Computer science and engineering at Sembodai Rukmani Varatharajan Engineering College, Nagapattinam.

M.Markco was born in Tamilnadu, India in 1988. He received her B.E. degree in Computer Science and Engineering from Anna University, Chennai in 2009. He completed his M.E. (Network Engineering) in Anna University, Regional Centre – Coimbatore. He is currently working as an Assistant Professor in the Department of Computer science and engineering at E.G.S.Pillay Engineering College, Nagapattinam.

ISSN: 2278 – 1323