ISSN: 2320-1363

Allocation of Resources To Virtual Machines In Cloud Environment For Reducing The Costs

A.Ajaykumar*1, G.Swapna Rani*2

[1]**M.Tech Student, Dept of CSE, CMR Institute of Technology, Kandlakoya Medchal, Hyderabad, India**

[2]**Assistant Professor, Dept of CSE, CMR Institute of Technology, Kandlakoya Medchal, Hyderabad, India**

ABSTRACT:

The trend of using virtual machines is very high, according to the technology of cloud computing, it has grown very huge. Computing of resources in cloud system can be partitioned in granules and shared on demand. This demonstrated my paper in three ways: Initially, prepared a deadline-driven resource allocation difficulty based on the cloud environment facilitated with Virtual machine resource isolation technology, and also propose a new solution with polynomial time, which could minimize user's costs in terms of their predictable deadlines. Secondly, analyzing the upper bound of task execution length based on the possible inaccurate workload prediction, then further proposes an error-tolerant technique to assurance of tasks completion within its deadline. Finally, validated its efficiency over actual Virtual Machine-facilitated cluster environment under different kinds of levels. In my research, by tuning algorithmic input deadline based on our original bound, task execution size can always be limited within its deadlines. I plan to integrate my algorithms with original deadlines with a tool like open nebula for the system performance. In my system, server nodes split the resources and distributed over the virtual machines. This will help to reduce the time complexity as well as task will be completed within deadlines.

KEYWORDS: Server Nodes, Resource Allocation, Virtual Machines Sharing, Cost Reduction.

ISSN: 2320-1363

I.INTRODUCTION

Cloud computing has evolved as a compelling model for the deployment of ease-of-use virtual environment on the Internet. One typical feature of clouds is its pool of easily accessible virtualized resources like hardware, platform or services that can be dynamically reconfigured to adjust to a variable load. Here load is can be termed for scale. All the resources provisioned by cloud system are supposed to be under a payment model, in order to avoid users over demand of their resources against their true needs. 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 prices on demand, also leading to a potentially high-dimensional execution scenario. The resource allocation in cloud computing is much more complex than in other distributed systems like Grid computing platform. In a Grid system, it is improper to share the compute resources among the multiple applications simultaneously running atop it due to the inevitable mutual performance interference among them. Whereas, cloud systems usually do not provision physical hosts directly to users, but leverage virtual resources isolated by VM technology.

According to the elastic resource usage model, I have intension design a resource allocation algorithm with high prediction-error tolerance facility, also reducing users' costs which subject to their expected deadlines since the idle physical resources can be randomly Partitioned and allotted to new tasks, the VM-based isolatable resource sharing could be very flexible. This implies the feasibility of finding the optimal solution through convex optimization strategies, un like the outmoded Grid model that relies on the independent resources like the number of physical cores. However, i check that it is in viable to openly resolve the essential and

ISSN: 2320-1363

necessary condition to find the optimal explanation through novel approach.

This paper discusses the concept of Cloud Computing. Cloud computing means storing and accessing data and programs virtually instead of your computer's hard drive. In this cloud system Cloud computing is an emerging computing paradigm in which resources of the computing infrastructure are provided as services over the Internet. This document pays much attention to the Grid paradigm, as it is often confused with Cloud technologies. In this cloud system, the cloud server nodes are described. From these server nodes the request for the tasks are distributed to virtual machines to accomplish the task within its deadlines

These processes can be fulfilled with novel resource allocation algorithm. We further defined the efficiency of our solutions by implementing a set of complex web services that are based on complex matrix-operations, over a real cluster environment with 70 virtual machines. In a Grid arrangement, it is offensive to allocate the considered resources among the multiple applications simultaneously running atop it due to the inevitable mutual performance interference among them. Whereas, in cloud systems typically do not stipulate physical hosts openly to users.

II.SYSTEM ARCHITECTURE

The system architecture depicted in fig.1.Consists of users, data owners, Cloud server, cloud server nodes, and virtual machines. The data owner will share his data into the cloud server then the user will keep request to download the data from cloud server. Cloud server will give permissions to download the data to authorized user only. The user who keep request to cloud server will only get the permissions to access the data. Then the cloud server takes the task and sends it to its specified server nodes. These server nodes split the task and send to its specified virtual machines to accomplish the task. Task is nothing but the available resource. And how the resource is allocated to the virtual machines is totally based on novel resource allocation algorithm. Design goals evolved from the system architecture are discussed in next section.

III.DESIGN OF CLOUD SYSTEM

Cloud computing has seemed as a enthralling paradigm for the arrangement of ease-to-use virtual environment on the Internet. One typical feature of clouds is its easy available of virtualized resources such

ISSN: 2320-1363

as hardware, platform or services. These can be dynamically reconfigured to adjust a flexible load (scale). All the resources provisioning by cloud system are made-up to be under a payment model, in order to avoid users' over request of their resources against their true essentials. Each task's workload is likely to be 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 commercially provided cloud. We provide a evaluation and study of each of these systems. We begin with a small Synopsis comparing the present unprocessed aspect set of this project.

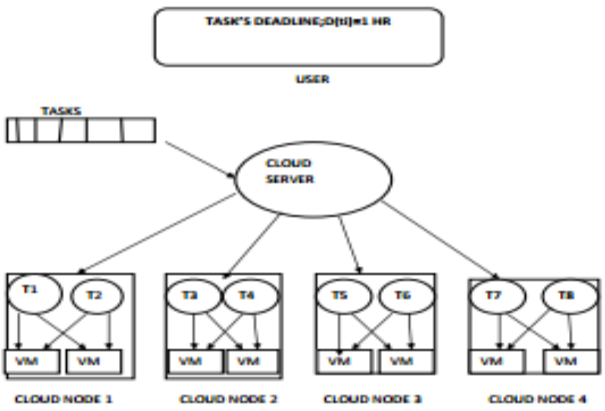

Fig.1. System Architecture

CPU, it can also be split into several sequential implementation phases, each calling for a dissimilar computing ability and various costs on demand, also leading to a theoretically high-dimensional execution scenario.

3.1 ALLOCATION OF RESOURCES USING SCHEDULING POLICIES

Every task will be handled based on their importance or in terms of First-Come-First-Serve (FCFS) program. Sometimes when the tasks priorities are of the same. Then each task's execution may involve multidimensional resource sharing process, such as CPU and disk I/O. It's a data extraction task, for example, generally I need to load a huge set of data from disk before or else in the middle of its computation. Ultimately, such a job may store its calculation results onto the local disk or a public server through network system.

3.2 CALCULATION OF PREDICTABLE TIME FOR EXECUTION

The process in executing such a task is denoted 'ti'. If suppose the task's execution times cost on computation and disk processing are expected as 4 and 3 hours, individually. Continuously receiving the request, the scheduler checks the availability states of all requested candidate nodes, and then estimates the marginal payment of running the task within its

ISSN: 2320-1363

deadline to each of them. The cloud that requires the lowest payment will run the task via a custom-built 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 completed within its deadline and its user cost can also be reduced. Finally, its computation results will be returned to users at their respective deadlines.

3.3 OPTIMAL RESOURCE ALLOCATION

This can minimize the optimal solution for minimizing the payments. It is nontrivial to do so, because the last situation cannot be solved directly. Although, we achieved a novel algorithm with polynomial time complexity to share resource, which can be proved to satisfy the condition. This algorithm is aimed based on such a innovation: if this is not considering the limit of resource capacities, then the problem can be directly resolved using Lagrangian multiplier method. Based on the convex optimization theory [2], [3], the Lagrangian function of the problem could be expressed as (1), where and are matching Lagrangian multipliers. Note that is a constant defined in (1) and r is the abbreviation of r (ti) as stated above

$$Fl(R)=\frac{1}{R}(\sum_{k=1}^{R}b_k r_k)(\theta\sum_{K=1}^{R}\frac{l_k}{r_k})+\lambda(\theta\sum_{K=1}^{R}\frac{l_k}{r_k}-D)+\sum_{K=1}^{R}\mu_k(r_k-a_k)$$

(1) Where, R=Execution Dimension, Bk=Price Vector, Rk=Resource Vector, Lk=Workload Vector, D=Deadline, Ak=Available Vector. According to theKarush-Kuhn-Tucker conditions that is necessary and appropriate condition of the optimization.

$$\lambda \geq 0, \mu_K \geq 0, K=1,2...\mathrm{R}$$

$$\sum_{I=1}^{R}\theta\frac{l_I}{r_I}\leq D$$

$$\lambda(\sum_{I=1}^{R}\theta\frac{l_I}{r_I}-D)=0$$

$$r_K \leq a_K(P_S), K=1,2....R; S=1,2,....\mathrm{n}$$

$$\mu_K(\mathrm{r}_K-a_K(P_S))=0, K=1,2....R; S=1,2...\mathrm{n}$$

$$\frac{dF1}{dr_k}=\frac{1}{R}((\sum_{i=1}^{R}b_i r_i).\frac{-l_k}{r_k^2}+b_k.\sum_{i=1}^{R}\frac{l_i}{r_i}+-\lambda\frac{l_k}{r_k^2}+\mu_k)=0 \quad \ldots.2)$$

K=1,2….R Where, R=Resource, Bk=Price Vector, Rk=Resource Vector, Lk=Workload Vector, D=Deadline, Ak=Available Vector

ISSN: 2320-1363

- Input: $D(t_i)$; Output:execution node p_s , $r^*(t_i)$
- $\Gamma = \Pi$, $C = D(t_i)$, $r^* = \phi$ (empty set);
- Repeat
- $r_\Gamma^*(t_i, p_s) = \text{CO-STEP}(\Gamma, c)$;
- on Γ^*
- $\Omega = d_k / d_k \in \Gamma \,\&\, r_k^{(*)}(t_i, p_s) > a_k(p_s)\}$;
- $\Gamma = \Gamma \setminus \Omega$ /*Γ take away Ω^* /
- $C = C - \theta \sum_{d_k \in \Omega} \frac{f_k}{a_k}$ /* Update C* /
- $r^*(t_i, p_s) = r^*(t_i, p_s) \cup (r_k^{(*)} = a_k(p_s) | d_k \in \Omega \,\&\, a_k(p_s)$
- is d_k's upper bound};
- until ($\Omega = \phi$);
- $r^*(t_i, p_s) = r^*(t_i, p_s) \cup r_\Gamma^*(t_i, p_s)$
- end for
- Select the smallest p(ti) by traversing the candidate solution set;
- Output the selected node ps and resource allocation r*(ti,ps);

3.4 OPTIMALITY ANALYSIS WITH INACCURATE INFORMATION:

This result was based on strong conditions. That is exactly task's workload vector. That is nothing but each user needs to surely identify the execution property (i.e., workload ratio) for his/ her job, before making the resource allocation with minimized payment for its execution under a deadline specified by the user. In some cases, the execution property could be simply predicted and measured exactly. For example, we can have the own workload ratio among the data to be read/written from/ to disk and those to be downloaded/uploaded via network connection by relating their data sizes. In many other cases, however, the property of execution cannot be estimated, such as computation-intensive requests whose execution times highly depending on the CPU cycles to consumed.

IV.RESULTS AND DISCUSSION

-This is the login screen which is used to enter the details of the user, owner, and cloud owner.

-Initially the owner will login with his credentials and upload some file into the server.

-Then the user will login into the server and he will keep request for some files.

-Then finally the cloud server owner will give permissions to accept the files to download the requested files.

This all will happen from this login page..

FIG 1Login Page

ISSN: 2320-1363

Description:

This graph discusses to measure the performance of the OAA and DER (Deadline Extension Ratio). The tasks which are inputted are shown in the X axis and the deadline extension ratio results are measured in Y axis. The main experimental results are shown by the original deadline D (i.e., D=D) in this algorithm. From Fig. 1, the task's performances times cannot be recurrently guaranteed to be performed within their deadlines in the worst case so many tasks (1-45) are submitted continuously. Specifically, the system availability is relatively high the average value of deadline extension ratio is near to 0.15.

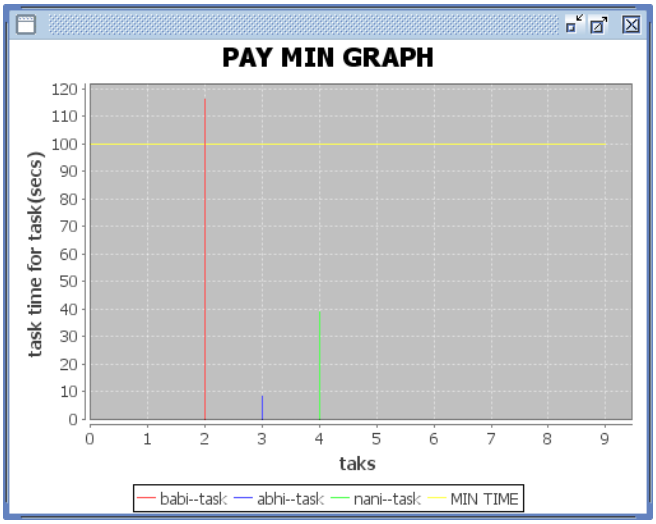

FIG2Performance with Deadline D= α*D

Description:

In this graph the deadline extension ratio D is set to aexact deadline (α*D). Number of tasks are denoted by 'm' are submitted up to 45, all task's execution times can be kept nearby to 2.00 times as high as their specific deadlines (D) are at poor by increasing the task's, task'scompletingtimes cannot be always definite because of the limited resource measurements, but the mean level is stillkept abnormally lower than 2.25, it means that most of the tasks can still meet the QoS .By this the most of the tasks are completed within its deadlines.

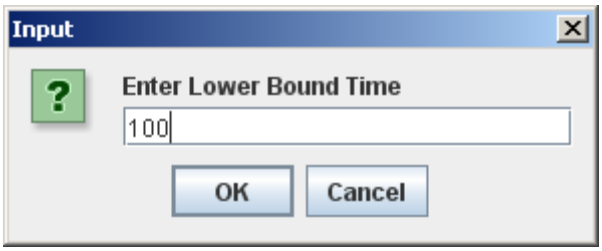

FIG3 Performance with Lower Bound

Description:

According to the analysis the optimal algorithm was based on the inaccuracy of predicted workload. From the above graph we can observe that the prediction method can surely make the lower bound the workload that is nothing but α value will be set close0 to 2, where α is defined is always lower.

ISSN: 2320-1363

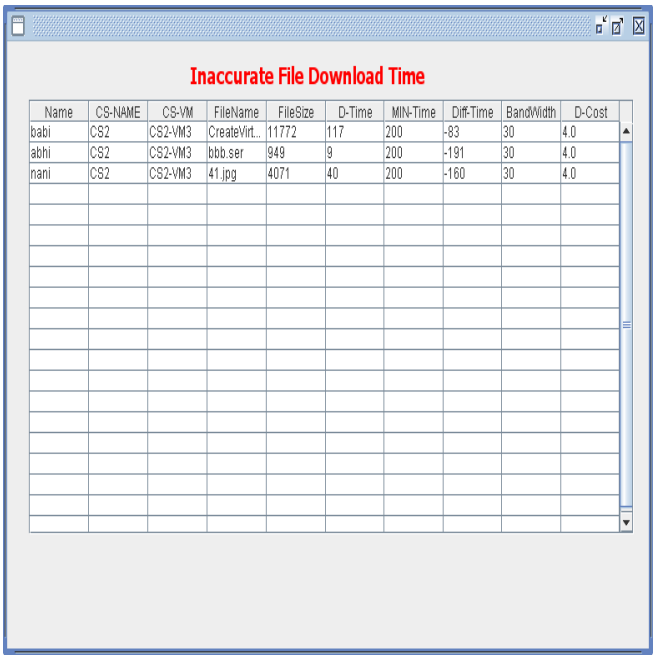

FIG4 Performance with Upper Bound

Description

In this the optimal algorithm is based on the inaccuracy of the workload predicted. From this Fig4, we can observe the prediction method used toensure the lower bound of the workload predicted.

RESOURCES ALLOCATION IN CLOUD SYSTEM

In cloud system how the resources will be allocated and moved to virtual machines shown in this screen. when the user requests for any resources it is equally shared among all the server nodes and equally distributed among them.it is clearly shown in figure

CONCLUSION

We conclude this paper withthe propose of novel resource allocation algorithm for cloud system that supports VM-multiplexing technology, aiming to minimize user's payment on specified task and also endeavor to guarantee its execution deadline.When the resources provisioned are comparatively satisfactory, this can assure task's implementation time always within its deadline even under the wrong prediction about task's workload features. In the future I plan to integrate my algorithms innovative Deadlines into some first-rate management tools like Open Nebula, for maximizing the system-wide performance. A few queuing methodologies like earliest deadline first (EDF) will be considered to advance decrease user payment cost particularly in the short supply circumstances.Design analyses show that the proposed scheme

ISSN: 2320-1363

satisfiesthedesired security requirements and it guarantees efficiency as well.

REFERENCES

[1]Sheng Di, Member, IEEE, and Cho-Li Wang, Member, IEEES.Di is currently a post-doctor researcher at INRIA, Grenoble, France, and C.L. Wang is with the Department of Computer Science, The Universityof Hong Kong, Hong Kong.

[2]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.

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

[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] E. Imamagic, B. Radic, and D. Dobrenic, "An Approach to Grid Scheduling by Using Condor-G Matchmaking Mechanism," Proc.28th Int'l Conf. Information Technology Interfaces, pp. 625-632, 2006.

[6]Amazon elastic compute cloud (EC2). http://aws.amazon.com/ec2/.

[7] Eucalyptus Home Page.http://www.eucalyptus.com/.

[8] Nimbus Home Page. http://www.nimbusproject.org/.

[9] Open Nebula Home Page. http://www.opennebula.org/.

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

[11]K. Fraser, S. Hand, T. Harris, I. Leslie, and I. Pratt.TheXenoserver Computing Infrastructure. TechnicalReport UCAM-CL-TR-552, University of Cambridge, Computer Laboratory, Jan. 2003.

[12]BECHTOLSHEIM, A. Cloud Computing and Cloud Networking.talk at UC Berkeley, December 2008.

[13] Cloudera, Hadoop training and support [online]. Available from: http://www.cloudera.com/.

[14] R. Creasy IBM Journal of Research and Development.Vol. 25, Number 5. Page 483.Published 1981.The Origin of the VM/370 Time-Sharing System.