ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2013

An Enhanced Cost Optimization of Heterogeneous Workload Management in Cloud Computing

[1]Sudha.C

Assistant Professor/Dept of CSE, Muthayammal College of Engineering,Rasipuram, Tamilnadu, India

Abstract: Load Balancing is a Computer Networking method to distribute workload across multiple Computers to achieve the Optimal Resource Utilization by maximizing the Throughput, and minimizing the Response time, and also avoids overloading among the systems. The Workload comprises both transactional and long-running analytic computations. These computations may bring new performance management challenges due to the different nature of a heterogeneous set of mixed workloads. The different nature imposes the need for new scheduling mechanisms to manage co-located heterogeneous set of applications such as to run a web application and a batch job on the same physical server. The proposed system implements the Improved Cost Effective Technique(ICET) which decides, whether the resources should be leased from the public Cloud and aggregated to the private Cloud, that will provide enough processing power to execute a workflow within a given execution time. This technique is used to reduce the cost within the desired execution time. ICE Techanique works by choosing the tasks based on their dependency. Hence it does not implement any sequential order. Therefore Cloud Harmony Search Optimization (CHSO) algorithm has been proposed to include the sequential selection among the tasks. This algorithm doesn't consider the dependency of tasks. It finds the optimal solution based on the mixed workload placement matrix and generates the best new placement vector that enhances accuracy and intersection rate of harmony memory. The required simulations are done in the Cloud Simulator to investigate the performance of the algorithm.

Keywords : Workflow, Harmony search, Workload Management, Placement Vector.

I. INTRODUCTION

Cloud Computing Support on demand Self Service, Pay-for-use and dynamically scalable storage services over the Internet. One of the Cloud Computing service is an Infrastructure as a Service.IaaS Provides Infrastructure components as a repository,firewalls, networks,Load balancing and other computing resources to their customers. Infrastructure as a Service is sometimes referred to as Hardware as a Service. Features and functionalities of IaaS are Instant VM Launching, Live VM Migration, Load balancers, Bandwidth Provisioning, Virtualized Computational & Storage Volumes, VM, Storage, Network Management. Load balancing is a process of applying the total load to the individual nodes of the shared system to enhance the response time on the job and to make effective resource usage, at the similar time. Extract a state in which some of the nodes are overloaded and some others are loaded. The intention of workload management develops the performance considerably,to have a backup plan when the system fails and to maintain the system durability. The Workload can be Memory capacity, interruption, CPU demand. At the same time, data centers will convey an unparalleled degree of heterogeneous workload combination,hosting in the same infrastructure distributed services from many different users.

Several Load balancing algorithms are provided in this area so far some examples of which will be discussed here. Dynamic genetic algorithm for load balancing in distributed system.They simulated that the characteristics of tasks can be defined in advance[8], Combination of FCFS and genetic algorithms are small no of jobs in the queue.The aim of this algorithm is to minimize execution time of the selected task considering the data existing in the scheduler. Once the queue has more task available,genetic algorithm used for load balancing. Ant colony Optimization (ACO) and another based on Particle swarm optimization (PSO) algorithm [9].

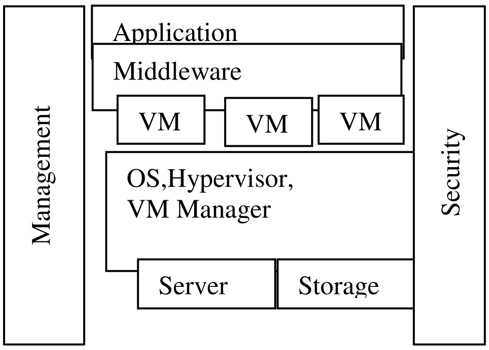

Fig .1 Cloud Computing Architectural blocks

The Performance of these algorithms has been compared in terms of makespan and load balancing level. The find results shows the proposed PSO algorithm has a good performance as compared with the proposed ACO algorithm. Maximum level of workload management is 84%.

II. RELATED WORK

A. Dynamic Application Placement Technique

The dynamic placement technique [3] has been proposed that allows the management to high level objective of mixed long running and transactional workloads in virtualized environments. Dynamic Application Placement Technique supply performance predictions with value to job completion time on a control cycle which within the job execution time. At the same time the actual utility complete by a job can only considered at completion time,the technique to predict the utility that each job in the system will finish given particular allocation,the concept of hypothetical utility for which they proposed assume that all jobs can be placed all together,and in which the available CPU power may be allocated among all jobs so that wanted utility is balance among all nodes.

B. PALB Algorithm

Power Aware Load Balancing (PALB) algorithm [4] proposed in the previous paper,maintains the state of all compute nodes based on utilization percentage. The algorithm has three basic cases. It does collect the utilization percentage of each active compute node, In the case that all compute node 'n' are above 80% utilization,all of the applicable compute nodes are in operation or else,the new virtual machine is to compute node with maximum utilization. The threshold of 80% utilization was taken since when 20% of the resources are available. Using PLAB, an organization needs to construct local clouds using Eucalyptus would be able to save on the cost of energy. This is due to the fact that Eucalyptus does not account for consumption of power when applying its default workload balancing technique.

C. Heterogeneous Workload

The Existing system [1] aims to make equal placements in terms of Relative Performance function. The use of RPF provides uniform workload particular performance models that allow proper placement decisions across various workloads . Optimization Objective of the request given from various customer applications. Here Relative Performance Function value calculated every application. The main objective of the Application Placement Controller is to find the finest feasible placement applications as well as an equivalent load distribution that optimizes the global objective of the system. Application placement Technique is used to manage Long Running jobs. Limitation of Existing Systems are the goals are basically defined in terms of average response time or throughput over a short time interval,Non interactive workloads typically require calculation of a schedule for an extended period of time and the cost of the infrastructure and usage of electrical energy is high.

Performance Model for workloads

We are given a set of jobs. With each job 'n' we associate following information

Resource usage: A resource usage explains the resource requirements of a job and is given at job submission time. The parameters used each job, the amount of CPU cycles for each job, and maximum and minimum CPU speed, memory requirement

Relative Performance Function

$$\frac{T_d - T_c}{T_d - T_d^{Start}}$$

Here T_d = Desired completion time
T_c = Actual job completion time
T_d^{Start} = Desired Start Time.

Evaluating *Placement Decision*

Let M be a Placement and P_n be the amount of CPU power allocated to application 'n' in placement M. For applications that are not placed, $P_n = 0$.

To calculate the relative performance of application n given placement M calculated at time t_{now} for a control cycle of length L, we calculate a hypothetical relative function at a time t_{now} + L. For each job,we increase amount of work that's done over L with allocation P_n. We use the so-obtained hypothetical relative function from placement matrix X and Y. This presumption helps us balance batch work execution over time.

III. OUR CONTRIBUTION

A. Improved Cost Effective Technique

The Improved cost effective technique is based on the following general steps

1. Primary schedule:schedule the workflow in the private cloud P.
2. While the deadline is smaller than the makespan means select tasks to reschedule, pick resources against the public clouds to construct the Hybrid cloud H, Reschedule the selected task in Hybrid cloud.

The Technique process two main steps are the choice of tasks to reschedule and the choosing of resources from the public cloud to make the Hybrid Cloud. While the former determine which tasks can have their execution time reduced by using more powerful resources from Public Cloud, and next determines the performance costs difficult in the new schedule. In This paper we analyze different kind of conditions to select which tasks will rescheduled if the expected execution time is violated in the Primary schedule and find how many resources to request from public cloud considering their cost and performance, task dependencies costs are check, since communication time between tasks in the same processor is to be null. More no of cores means lower cost per core. The Technique General Steps are the Initialize work flow Scheduling, the technique checks if resources from the private cloud are required based on the deadline. Compare makespan with deadlines, while deadline violated,the technique iterates until the deadline is met Select the node n_i task such that Priority P_i is maximum and n_i is not in the rescheduling group G,added to node n_i to the group G, which is composed of all nodes are rescheduled from the private cloud to the public cloud. An iteration is repeated until the number of selected cores reaches the no of clusters in G. The selected Resource adds to the hybrid cloud. Reschedule the selected tasks in Hybrid cloud.

B.Cloud Harmony Search Optimization

A Harmony search algorithm is a progress algorithm introduced by Geen et al. In 2001 for the first time . This algorithm assumes musical improvisation process. Each output is called a harmony in this algorithm. All the data kept in Harmony Memory(HM). Generates the new solutions based on three basic principles like harmony memory consideration,pitch adjusting and random selection[6][7].

Fitness Function

One of the key feature of the load balancing algorithm is the fitness function, which indicates the amount of fitness function for each workload.

$$Load(L_i) = CurrentLoad(L_i) + \sum_{j=0}^{n} \frac{Tasklength(j)}{PP(L_i)} \qquad \text{---------------------------------(1)}$$

$$+ \sum_{j=0}^{n} \frac{Osizeof(j) + Isizeof(j)}{PW(L_i)}$$

Where, L_i is the desired resource, current load() is the current load of each resource, task length gives the processing requirement of each take, $PP(L_i)$ processing power of each resource, PW is the bandwidth between each resource with scheduler, Isize and Osize offer the size of input and output files. Makespan is the completion time of the final task allocated to resources which is given by Eq.(2), ‘n’ is the no of resources.

Makespan = max(load(L_i)) i={1,2,3…n} --- (2)

The resource utilization ratio of desired resource load to makespan which is given by Eq.(3). Average Utilization of each resource is stated in Eq.(4) as the criterion effective on the fitness of a solution.

U(L_i)= load(L_i)/Makespan. -- (3)

$$\text{Avg (U)} = \frac{\sum_{k=1}^{n} U(k)}{N} \qquad \text{-- (4)}$$

CHSO parameters

In this paper, PAR,HMCR and BW in are considered as functions of harmony fitness[8] generated in the previous iteration which are described by Eq.(5),Eq.(6) and Eq.(7) ,respectively. HMCR will be increased if the harmony of former iteration shows a worse fitness versus the best harmony available, the existing harmonies in the memory are used at a higher probability and changes are less probable, When the resource has extra load and a task has been assigned, then the task will be converted to the resource with the lowest workload.

HMCR(i) ={ $HMCR_{min}$ +($HMCR_{max}$ – $HMCR_{min}$)×$fitn_{best}$/$fitn_{worst}$} -------------------- (5)
PAR(i) = {PAR_{max} – (PAR_{max} – PAR_{min})×$fitn_{best/}$ $fitn_{worst}$ } -----------------------(6)
BW(i) = {bw_{max} –(bw_{max} – bw_{min}) ×$fitn_{best}$/$fitn_{worst}$ } ------------------------------ (7)

CHSO Algorithm Steps:

Step1 :Initialize random Harmonic memory parameters.
Step2:Generate harmony memory with random placement vector according to uniform distribution of each workload.
Step3:Workload Scheduling parameter initialize
Memory requirement,CPU Speed,Length of work Schedule
Step4:Generate a new placement vector for each workload.

With the probability harmony memory considering rate (0≤hmcr≤1),pick the stored value from harmony memory.
With the probability of particular range,pick a random value within the allowed range.
Step5:Perform different kinds of calculation needed if the value in Step 4 came from Harmony memory
With probability Pitch adjusting rate(0≤Par≤1);
Step 6:If new placement vector is better than worst vector in Harmony memory replacing the worst with new placement vector.
Step 7: Repect from Step 2 to Step 6 until termination criterion(e.g.maximum iterations) is satisfied so that all workloads is placed.

IV. SIMULATION RESULT

A cloud computing environment including of users and more resources has been taken into account for the purpose of simulation. The used simulation tool kit CloudSim. The CloudSim implements a variety of workflows for batch jobs and includes a performance model for transactional workloads. It considers a fertilized system, in which VM control mechanisms such as live migration,resume are used to construct application placement.

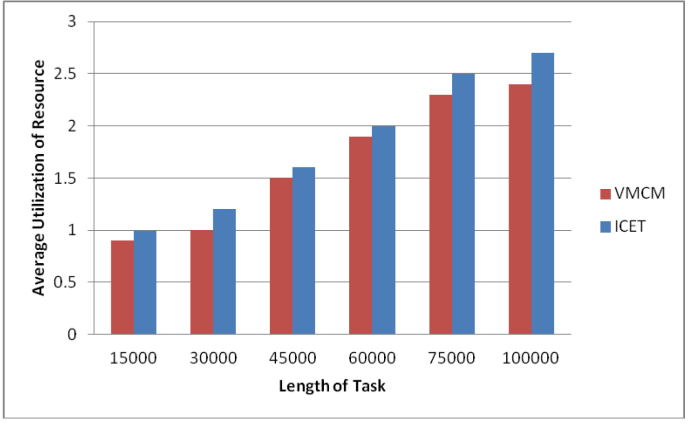

Fig. 2 Average utilization of Resources versus Length of tasks for ICET and VMCM

We compared our Improved Cost Effective Technique to existing VM Control Mechanism(VMCM).While average resource utilization of it is 2.76% higher than that of VMCM.
The cost of these mechanisms, with respect to Execution time they take to complete heterogeneous workload in a cloud environment. We compared our Improved Cost Effective Technique to existing VM control mechanism. ICET can reduce cost while achieving the established desired execution time and also implement CHSO to find best cost effective optimal solution for ICET.

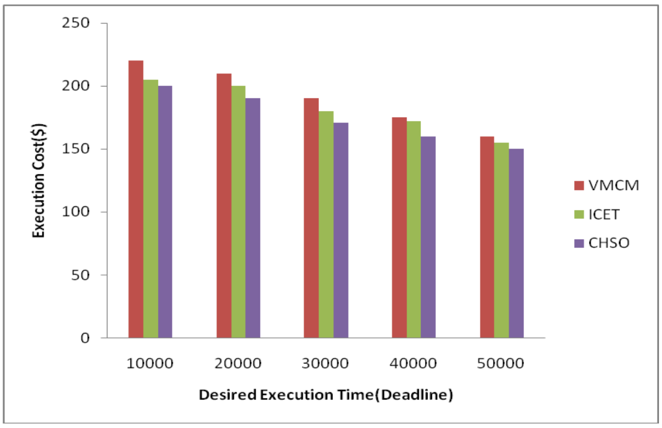

Fig .3 Execution Cost Comparison graph

V. CONCLUSION AND FUTURE WORK

The System has been carried out and integrated with a business application server middleware, which give the support for executing placement decisions. Our System is managed high level application goals and takes into report the application satisfaction with how well the goals are met.

we proposed an Improved cost effective technique used to speed up the execution of workflow submit to desired execution time but also minimize cost compared to the greedy approach, it provides makespans as low as the user needs.As a result, the user is able to control costs by changing the desired workflow execution time deadline.The technique makes it flexible to the use with different requirements.We implement Cloud Harmony Search Optimization for solving Optimal problems in heterogeneous workload management in the cloud computing environment. Our algorithm discovery fitness for each placement vector matrix .For future work to solve tough multiobjective and multicriterion NP-hard optimization problems.

REFERENCES

[1] David Carrera,Malgorzata Steinder,Ian Whalley,Members,IEEE:Autonomic Placement of Mixed Batch and Translational Workloads:IEEE Transactions on parallel and distributed system,VOL 3,NO. 2,FEBRUARY 2012.

[2]FatemehHeydari and hadi Shahriar Shahhoseini:"Dynamic and Adaptive Loadbalancing using Harmony Search in Grid Computing"IJCSNS International Journal of Computer Science and Network Security,Vol.12 NO.9,Sep 2012.

[3] David Carrera,Malgorzata Steinder:"Managing SLAs of Heterogeneous Workloads using Dynamic Application Placement. "

[4] Jeffrey M. Galloway,Karl L. Smith,Susan S. Vrbsky:Power Aware Load Balancing For Cloud Computing,WCECS 2011.

[5] Chen, J., Pan, Q. and Li, H., "Harmony search algorithm with dynamic subpopulations for scheduling identical parallel machines," Sixth International Conference on Natural Computation, Vol. 5, pp.2369-2373, 2010.

[6] Mahdavi, M., Fesanghary, M. and Damangir, E., "An improved harmony search algorithm for solving optimization problems," Mathematics and Computation, Vol. 188, pp.1567-1579, 2007.

[7] Rizos Sakellariou and Henan Zhao:Hybrid Heuristic for DAG Scheduling on Heterogeneous Systems. pp. 111-123 IEEE Computer Society.

[8] Zomaya, A.Y., Teh, Y.H.," Observations on using genetic algorithms for dynamic Load Balancing," IEEE Transactions on Parallel and Distributed Systems, Vol. 12, Issue 9, pp. 899-912, 2001.

[9] Ludwig3T, S.A., 3TMoallem3T, A., "Swarm Intelligence Approaches for Grid Load Balancing, " 3TJournal of Grid Computing3T,3T Vol. 9, No. 33T, pp.0T279-301, 2011.

[10] Geem, Z.W., Kim, J.H. and Loganathan, G.V., "A new heuristic optimization algorithm: harmony search," Simulation, Vol. 76, pp.60-68, 2001.