Enhanced Automatic Control Virtual Machine for Cloud Using Real Time Task Oriented Algorithm

Ms.M.Sowmeena[1], Mr.C.Ramesh[2]

[1]PG Scholar, Department of Computer Science and Engineering,
[2]Assistant professor, Department of Computer Science and Engineering,
Vivekanandha College of Technology for Women,
Elayampalayam, Tiruchengode.
Tamilnadu, India.

***Abstract*— The enterprise aims to upgrade the working of applications in the unify environment by balancing the memory pages of virtual machines. The MEB system is inconsequential and can be perfectly integrated into user space without interfacing with virtual machine monitor operation. A Global scheduling algorithm based on the dynamic criterion to determine the optimum allocation memory. They virtualization technique enables multiple virtual machines (VMs) to be placed on the same physical hosts and supports the live migration of virtual machine between physical hosts based on the performance requirements. When virtual machine does not use all the provided resources, they can be logically resized and consolidated to the minimum number of physical hosts. While idle nodes can be switched to sleep or hibernate mode to eliminate the idle energy consumption and thus reducing the total energy consumption (TEC) in cloud data centers. Cloud can achieve the same level of computing power a supercomputer does but at a much-reduced cost. Cloud is like a virtual supercomputer. The efficient task scheduling considers the completion time of tasks in a cloud environment. ASJS is mainly intended to decrease job's completion time. This project works on ASJS algorithm. This algorithm mainly concentrates on allocating suitable VM with the suitable task. The computing power of each resource is defined as the product CPU speed and available CPU percentage and the transmission power of each cluster is defined as the average bandwidth between different clusters. ASJS status of each resource in the cloud as parameters to initialize the cluster score of each bundle.**

***Keyword:* virtual machine, EASJS, ASJS, Cluster, Resource.**

I. INTRODUCTION

Cloud computing is a type of Internet-based tally that maintains mutual computer processing assets and data to computers and other devices for use. Cloud computing is the phrase used to describe different scenarios in which computing resource is delivered as a service over a network connection (usually, this is the internet). One of the key characteristics of cloud computing is the resilience that it offers and one of the ways that elasticity is offered through scalability. Cost benefits must be extended though, as the operation in question will have to purchase/rent and uphold all the crucial software and hardware. A hybrid cloud grants a team to maximize their competence by utilizing the public for non-sensitive operations while using a private setup for sensitive or mission-critical operations.
Cloud computing exhibits the following key characteristics

- Agility
- Application programming interface
- Cost reduction
- Device and location independence
- Maintenance

II. RELATED WORKS

Christopher Clark et al [2] Migrating operating system instances across distinct physical hosts is a useful tool for administrators of data centers and clusters: It allows a clean separation between hardware and software, and facilitates fault management, load balancing, and low-level system maintenance. The design options for migrating OSes running services with liveness constraints.The administrator selects a minimum and a maximum bandwidth limit. The first pre-copy round transfers pages at the minimum bandwidth. Each subsequent round counts the number of pages dirtied in the previous round, and divides this by the duration of the previous round to calculate the dirtying rate. The bandwidth limit for the next round is then determined by adding a constant increment to the previous round's dirtying rate have empirically determined that 50Mbit/sec is a suitable value. It terminate pre-copying when the calculated rate is greater than the administrator's chosen maximum, or when less than 256KB remains to be transferred. By integrating live OS migration into the Xen virtual machine monitor we enable rapid movement of interactive workloads within clusters and data centers. The dynamic network-bandwidth adaptation allows migration to proceed with minimal impact on running services, while reducing total downtime to below discern able thresholds. Feature of this study is moving the contents of a virtual machines memory from one physical host to another can be approached in any number of ways.

Armando Fox [3] Jobs are scheduled as soon as it arrives. Because a grid environment is heterogeneous with different types of resources, online mode heuristic scheduling algorithms are more appropriate for grid environment. Dynamic FPLTF scheduling algorithm (DFPLTF) is based on the fastest processor to Largest Task First scheduling algorithm it can be converted to the fastest processor to Largest Task First scheduling algorithm more flexible for grid environment. More Fit Task First Scheduling Algorithm (MFTF) mainly attempts to assign the most suitable resource to the task by a value called fitness. The range of fitness is from 100,000 to 0 and is defined where Wi is the workload of job “i”, “Sj” is the CPU speed of resource “j” and “Ei” is the expected execution time of job “i”.“Ei” is determined by the user or estimated by the system. Another dynamic scheduling situation in grids is called workflow scheduling. A workflow consists of a set of task to be executed. These tasks may have precedence relation or there may have data that need to be transferred from one task to another. Therefore, workflow scheduling is grid is more difficult than a simple scheduling for a set of independent tasks.

D.Saha [4] the authors stated that most parallel jobs cannot be fully complemented. In a homogeneous parallel machine-one in which all processors are identical the serial fraction of the computation has to be executed at the speed of any of the identical processors, limiting the speedup that can be obtained due to parallelism. The insubstantial uses Markov chain occupying models to check out the deed of static and dynamic processor assignment arrangement for heterogeneous architectures. Parallel jobs are assumed to be described by acyclic directed task graphs. A new static processor assignment policy, called Largest Task First Minimum Finish Time (LTFMFT), is introduced. The analysis shows that this policy is very sensitive to the degree of heterogeneity of the architecture and that it outperforms all other policies analyzed. Three dynamic assignment disciplines are compared and it is heterogeneous environments, the disciplines that perform better are those that consider the structure of the graph, and not only the service demands of the individual tasks. Finally, static and dynamic processor assignment disciplines are compared in terms of performance.

Syed Nasir Mehmood Shah et al [5] describe the authors stated that a ‘Grid' is an infrastructure for resource sharing. It is used for large-scale data processing, many of the applications being scientific ones. Reliability, efficiency, and effectiveness in resource utilization are the desired characteristics of grid scheduling systems. In this paper, they proposed two more flavors of multilevel hybrid scheduling algorithms; i.e. the dynamic multilevel hybrid scheduling algorithm using median and the dynamic multilevel hybrid scheduling algorithm using square root. They have three key features. First, they favor the shortest job for execution. Second, they execute the job on the support of a dynamic time quantum, to moderately distribute processor time among grid jobs. A third feature is that they always execute the longest job, thus avoiding starvation. ‘Scheduling’ is described by the grid scheduling dictionary project as follows: the third phase includes job execution. In the second phase, the selection of the best match of jobs to resources is an NP-complete problem. In this paper, they presented a solution to the fixed time quantum problem by proposing two more flavors of the multilevel hybrid scheduling algorithm. Scalability testing is a significant success factor for the design and development of a grid scheduling algorithm. Scalability means a measure of optimizing the application to use more processing power given to it in the form of additional processors or cores.

Chee Shin Yeo et al [6] describe have declared that the grid computing enables the virtualization and dynamic delivery of computing services on demand to realize utility computing. With this new outsourcing service model, the user is able to define their service needs through service level agreements (SLAs) and only have to pay when they use the services. It then describes two appraisal methods that are simple and visceral: (i) separate and (ii) integrated risk analysis to analyze the effectiveness of resource management policies in achieving the objectives. Evaluation results based on simulation successfully demonstrate the applicability of separate and integrated risk analysis to assess policies in terms of the objectives. Thus, a wholesale computing service will appearance two the new challenges: (i) what is the intention or goals it needs to attain in regulation to backing the utility computing model and (ii) how to gauge even if these aspirations are carried out or not. Different users have distinctive needs for various jobs and thus demand specific Quality of Service (QoS) for completing these jobs. A user can negotiate the QoS terms and conditions with commercial computing service provider before formally outlining the confirmed negotiations in a Service Level Agreement (SLA) such as the number of processors, memory size, disk storage size and runtime estimate.

III. METHODOLOGY

When science and technology advance, the problem encountered becomes more computing power. In contrast to the traditional notion of using supercomputers, grid computing is proposed. Distributed computing supports resource sharing. Parallel computing supports computing power. The grid can achieve the same level of computing power as a supercomputer does but at a much-reduced cost. In order to utilize the power of grid computing completely, we need an efficient job scheduling algorithm to assign jobs to resources. The project considers Adaptive Scoring Job Scheduling algorithm (ASJS) which aims to decrease job’s completion time. We consider not only the computing power of each resource in the grid but also the transmission power of each cluster in a grid system.

A. ADD CLUSTER

In this module, the cluster id is given with ATP (Average Transmission Power) and ACP (Average Computing Power) and CS (Cluster Score) value set to zero. The details are saved in ‘cluster’ table.

B. ADD RESOURCE

In this module, the resource id, Resource Name, IP Address, CPU MHz, CPU MHz available, Load percent, CP (available computing power) and storage capacity details are keyed. The details are saved in ‘Resources’ table.

C. ASSIGN CLUSTER TO RESOURCE

In this module, the cluster id is fetched from 'Clusters' table and resource it is fetched from 'Resource' table. The id is selected from combo boxes and is saved in 'Cluster Resource' table.

D. ADD JOB

In this module, the job id, name, required RAM in MHz, required hard disk storage in MB, CPU MHz, and network bandwidth is keyed and saved into 'job' table.

E. CLUSTER SCORE CALCULATION FOR DATA INTENSIVE AND COMPUTATION INTENSIVE STRATEGY

In this module, the cluster score is calculated based on the following formula. Where CS is the cluster score for cluster "i", a and b are the weight value of ATP and ACP respectively, the sum of α and β is 1, ATP and ACP are the average transmission power and average computing power of cluster "i" respectively. ATP means the average available bandwidth the cluster "i" can supply to the job and is defined as- Where Bandwidth_available is the available bandwidth between cluster i and cluster j, m is the number of clusters in the entire cloud system. Similarly, ACP means the average available CPU power cluster "i" can supply the job and is defined as- Because the transmission power and the computing power of a resource will actually perturb the performance of job execution, these two factors are used for job scheduling. Since the bandwidth between resources in the same cluster is usually very large, we only consider the bandwidth between different clusters.

F. CLUSTER SCORE CALCULATION WITH STORAGE CAPACITY

In addition to the existing formula, the storage capacity is also calculated as Average Transmission Power and so the sum of α, β and γ are 1. All other calculations are used in the same scenario as above module. The job is split into tasks with α, β and γ values for each subtask. So one cluster is assigned to one task and others cluster for other tasks. Likewise, jobs are considered as replica units and so more cluster are assigned for each job.

ENHANCED ADAPTIVE SCORING JOB SCHEDULING ALGORITHM (EASJS)

EASJS aims to decrease job's completion time. We consider not only the computing power of each resource in the cloud but so the transmission power of each cluster in a cloud system. The computing power of each resource is defined the product CPU speed and available CPU percentage and the transmission power of each cluster is defined as the average bandwidth between different clusters. ASJS uses the status of each resource in the cloud as a parameter to initialize the cluster score of each cluster. The cluster score of each cluster will be adjusted by applying the local update and global update. The system will submit a job to the most appropriate resource according to the scores.

CLUSTER SCORE

The cluster score is calculated based on the following formula. Where CS is the cluster score for cluster "i", "a" and "b" are the weight value of ATP and ACP respectively, the sum is 1, ATP and ACP are the average transmission power of cluster i respectively. ATP means the average available bandwidth the cluster "i" can supply to the job and is defined as: Where bandwidth available is the available bandwidth between cluster "i" and cluster "j", m is the number of clusters in the entire cloud system. Similarly, ACP means the average available CPU power cluster "i" can supply the job and is defined as: Where CPU_speed is the CPU speed of resource kin clusters "i", the load is the current load of the resource k in cluster "i", n is the number of resources in cluster "i". Since the bandwidth between resources in the same cluster is usually very large, we only consider the bandwidth between different clusters. Local update and global update are used to compose the score.

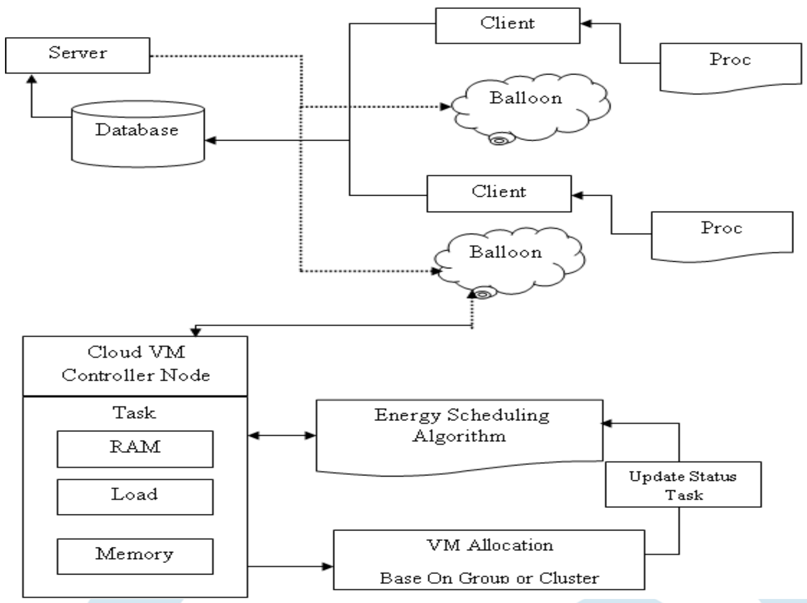

FIG .1 CONTROLS VIRTUAL MACHINE OF REAL-TIME TASK

IV CONCLUSION

Global update rule updates the prestige of each resource and cluster after a job is completed by a resource. It supplies the job scheduler the newest information of all resources and clusters such that the job scheduler can select the fittest resource for the next job. The preliminary results show that Enhanced Adaptive Scoring Job Scheduling is capable of decreasing completion time of jobs and the performance of Adaptive Scoring Job Scheduling is better than other methods. In the future, Enhanced Adaptive Scoring Job Scheduling can be applied to real grid applications. This project focuses on job scheduling. The project can be modified to consider the division of file and the replica strategy in data-intensive jobs. Jobs are independent of this project, but they may be carried out in the future.

V. RESULTS AND DISCUSSION

"Table 1.1" specify the resources allocation for GESA clustering and EASJS algorithm. The desk contains a number of resources allocations for GESA and EASJS Algorithm.

Table 1 Performances Analysis- Resources Allocation

S. No	Number of Resource [n]	GESA [n]	EASJS [n]
1	30	20	25
2	40	30	35
3	50	40	45
4	60	50	55
5	70	60	65
6	80	70	75
7	90	80	85
8	100	90	95
9	110	100	105
10	120	102	110

"Fig 1.1" describes the resources allocation for GESA clustering and EASJS algorithm. The table contains a number of resources allocations for GESA and EASJS Algorithm.

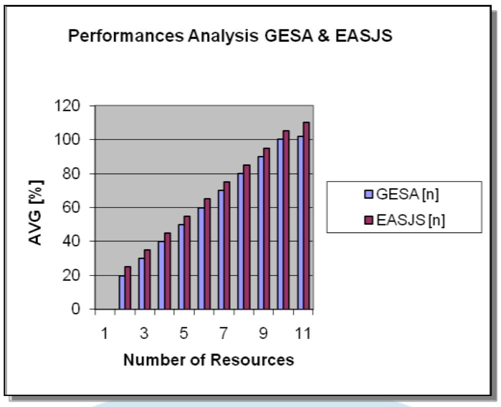

Fig 1.1 Performances Analysis- Resources Allocation

"Table 1.2" describes the resources allocation time complexity for GESA clustering and EASJS algorithm per seconds. The figure contains a number of resources allocation time details for GESA and EASJS algorithm.

S. No	Number of Resource [n]	GESA [Sec]	EASJS [Sec]
1	100	122	118
2	200	143	138
3	300	152	153
4	400	164	156
5	500	173	166
6	600	155	146
7	700	146	142
8	800	172	171
9	900	143	142
10	1000	167	163

Fig 1.2 Performances Analysis- Resources Allocation Time

"Fig 1.2" describes the resources allocation time complexity for GESA clustering and EASJS algorithm per seconds. The figure contains a number of resources allocation time details for GESA and EASJS algorithm.

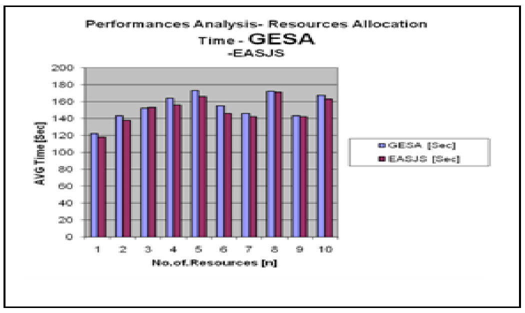

Fig 1.2 Performances Analysis- Resources Allocation Time

VI. FUTURE ENHANCEMENTS

The system is very flexible and user-friendly, so the maintenance based on the changing environment and requirement can be incorporated easily. Any changes that are likely to cause failures are prevented with security and preventive measures could be taken. The coding is done in understandable and flexible method program which helps easy changing. Since MS-SQL server and .NET are very flexible tools, the user can easily incorporate any modular program in the application. The project provides the best assistance in document management. The application becomes useful if the below enhancements are made in future. If the application is designed as web service, it can be integrated into many websites. Non-synchronized documents can be alerted to synchronize. The application is developed such that above said enhancements can be integrated with current modules. The application becomes useful if the below enhancements are made in future. If the application is designed as web service, it can be integrated into many websites. In addition, the same concept will in another service model of PaaS and SaaS. The thesis contribution will be implemented in the real time in a cloud environment. In addition, the future enhancement will plan to analyze the load balancing work in the cloud area. The application is developed such that above said enhancements can be integrated with current modules.

REFERENCES

[1] Wei-Zhe Zhang,Hu-Cheng xie and Ching-Hsien Hsu," Automatic memory control of virtual Machines on a consolidation server,vol 5, ,2017.
[2] Christopher Clark.C, K .Fraser, S. Hand ,J.G .Hansen, E.Jul, C.Jul, C .Limpach, I .Pratt and A. Warfield, "Live migration of virtual Machines", in proc .2nd Conf. Symp. Netw. Syst. Des.implementation-Volume 2, 2005, pp. 273-286.
[3] Armando Fox, Rean Griffith, Anthony,D. Joseph, Randy Katz, Andy Konwinski, David Patterson, "Online mode heuristic Scheduling algorithm", 2012.
[4] D.Saha, D.Menasce , S.Porto, Static and dynamic processor scheduling disciplines in heterogeneous parallel architectures,Journal of parallel and distributed computing, Vol.28, No.1, 1995, pp.1-8.
[5] Syed Nasir Mehmood Shah, Ahmad Kamil bin mahmood, Alan Oxley,dynamic multilevel Hybrid scheduling algorithms for Grid Computing, process computer science 4, 2011, pp. 402-411.
[6] Chee Shin Yeo, Maheswaran.M, S.Ali,H .J.Siegel, D.Hensgen, R.Frund, Integrated risk analysis for a commercial computing Service in utility computing, the journal of parallel and distributed computing 59, 2013, pp.107-131.
[7] Tudor Ioan Salomie, G.Alonso, T.Roscoe and K.Elphinstone,"Application level ballooning for efficient server consolidation", in Proc.8th ACM Eur.Conf.Comput.Syst. 2013, pp. 337-350.
[8] Zhang.W,Cheng.T,H.He and A.M.K.Cheng,"LVMM: A lightweight virtual machine memory management architecture for virtual computing Environment", in proc.Int.Conf.Uncertainty reasuring knowl.Eng,2011,Vol.1,pp.235-238.
[9] Zhang.W, He.H, G.Chen and J.Sun, "Multiple virtual machines resource scheduling", Appl. Math.Inf.Sci, vol.7, no.5, pp.2089-2096, 2013.
[10] Zhao.W, Z.Wang and Y.Luo, "Dynamic memory Balancing for virtual machines", ACM SIGOPS Oper.Syst.Rev.,vol.43, 2009, pp.43-47.