ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 10, October 2017

DATA STAGING AND CACHING USING GREEDY AND GENETIC ALGORITHM IN CLOUD

Priyanka Garg, Poonam Rana
Krishna Institute of Engineering & Technology, Ghaziabad

ABSTRACT

We propose an optimization model based on total migration coststaging and caching, down time and operational cost in any dynamic cloud environment. This model can be applied to pack the virtual machines tightly on to the minimum number of physical nodes. We used genetic algorithm to solve the optimization problem as the packing problem is priority on VM. We also propose load balancer framework making use of the Greedy-genetic algorithm (Greedy-GA) method. We found out in our experimental results that our model outperforms existing models in terms of minimizing the number of active nodes, reduction in energy consumption and improving pool utilization.

Keywords: VM, LIZA, CPU etc.

I.INTRODUCTION

In a virtual machine, virtualization is software that separates physical infrastructures to create various dedicated resources. It is the fundamental technology that powers cloud computing. [Cloud Computing: A Small Business Guide] "Virtualization software makes it possible to run multiple operating systems and multiple applications on the same server at the same time," said Mike Adams, director of product marketing at VMware, a pioneer in virtualization and cloud software and services. "It enables businesses to reduce IT costs while increasing the efficiency, utilization and flexibility of their existing computer hardware." The technology behind virtualization is known as a virtual machine monitor (VMM) or virtual manager, which separates compute environments from the actual physical infrastructure.

A. Network Virtualization

Network virtualization is the key to the current and future success of cloud computing. In this article, we explain key reasons for virtualization and briefly explain several of the networking technologies that have been developed recently or are being developed in various standards bodies. The Private Cloud is a model or architecture and frequently exhibited as being the answer for all our computing issues. It is particular and secure Cloud based environment which can be accessed and managed by the organization. It is nearer to the more traditional model of individual local access networks (LANs) that is utilized as a part of the past by big business however having the added advantages of virtualization. This can be also called either "Internal" or "corporate" or "venture" Cloud and it will be ensured by firewall. Cloud computing present's new open doors and new challenges. One of those challenges is the means by which safety is approach in the Private Cloud and hybrid Cloud. By considering a design approach to Hybrid and Private Cloud security, we would have the capacity to know the serious ideas, standards and blueprints that gives a fruitful safety achievement of Hybrid Cloud and Private loud.

B. Need of virtualization

There are many reasons to virtualize resources. The five most common reasons are:

1 Sharing: When a resource is too big for a single user, it is best to divide it into multiple virtual pieces, as is the case with today's multi-core processors. Each processor can run multiple virtual machines (VMs), and each machine can be used by a different

user. The same applies to high-speed links and large-capacity disks.

2 Isolation: Multiple users sharing a resource may not trust each other, so it is important to provide isolation among users. Users using one virtual component should not be able to monitor the activities or interfere with the activities of other users. This may apply even if different users belong to the same organization since different departments of the organization (e.g., finance and engineering) may have data that is confidential to the department.

3 Aggregation: If the resource is too small, it is possible to construct a large virtual resource that behaves like a large resource. This is the case with storage, where a large number of inexpensive unreliable disks can be used to make up large reliable storage.

4 Dynamics: Often resource requirements change fast due to user mobility, and a way to reallocate the resource quickly is required. This is easier with virtual resources than with physical resources.

5 Ease of management: Last but probably the most important reason for virtualization is the ease of management. Virtual devices are easier to manage because they are software-based and expose a uniform interface through standard abstractions.

II. RELATED WORK

Scheduling performed using several algorithms are studied in the literature recently. In [2], the author proposes an image caching mechanism to reduce the overhead of loading disk image in virtual machines. The author in [3] presents a dynamic approach to create virtual clusters to deal with the conflict between parallel and serial jobs. In this approach, the job load is adjusted automatically without running time prediction. In [4], the author describes a suspend/resume mechanism is used to improve utilization of physical resource. The overhead of suspending/resume is modeled and scheduled explicitly. In [5] present a planner-guided strategy for multiple workflows. It ranks already tasks and decides which task should be scheduled first. However if there are new lower rank workflows coming continuously, the higher rank task will not be scheduled to execute. In a massive scalable cloud, this situation will become true. This algorithm only considers the execution time. The author in [6] describes the Minimum Execution Time (Met) assigns each job in arbitrary order to the nodes on which it is expected to be executed fastest, regardless of the current load on that node. Met tries to find good job-node pairing, but because it does not consider the current load on a node it will often cause load imbalance between the nodes and not adapt application in the heterogeneity computer systems. On scheduling in grid computing environment, genetic algorithm could be adopted to solve the NP-hard problem. The author in [7] represents an extensive study on the usefulness of GAs for designing efficient Grid schedulers when make span and flow time are minimized. A first-come-first-served and a Genetic algorithm are used for the load balancing strategy [8]. It gives GA based task scheduling for optimum decision. The author in [9] introduces the Task and Virtual Machine (TVM) scheduler, which schedules VMs in cloud and task on VMs. By scheduling both, task and VMs, it is possible to define virtual computing systems that go beyond the limitation imposed by the availability of resources. The Virtual machines are instantiated in every host in the network prior to the execution of the application. Tasks composing application have requirements that can limit the choice of resources for execution.

II.PROBLEM IDENTIFICATION

The problem is to determine the number of physical machines, deciding on the locations of running virtual machines with minimizing costs comes from virtual machine rearrangement subject to capacity constraints on physical machines.

III. SYSTEM MODEL

Theoreticallycommunication, cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing have five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.

A cloud operating system is a new category of software that is specifically designed to holistically manage large collections of infrastructure – CPUs, storage, networking – as a seamless, flexible and dynamic operating environment.

Analogous to the operating system that manages the complexity of an individual machine, the cloud operating system manages the complexity of a datacenter. Although alternative approaches may be pursued, TekLeap believes that virtualization is the key underpinning technology to enable cloud computing.

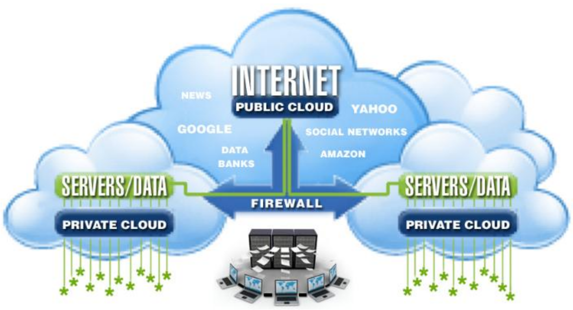

Figure 1: virtual cloud computing system

Each user has their own keyboard, screen, setting, application, and data files, so their experience is just like they were working at an independent PC. Virtual desktop share the excess power of standard PCs and make computing simple affordable, and energy-efficient.

IV. PROPOSED IMPLEMENTATION

The proposed system the Greedy Genetic Algorithm (Greedy-GA) module is integrated into MALAB. In this system, we have Data centers that comprise of hosts. Each of the hosts has one or more Processing Elements (PE). On these hosts, we have various VMs running. These VMs have one or more job running on them. In server, user jobs are directly represented as server operator. The Server operators have various requirements. The processing power requirement of each user is represented using Million Instructions per Second (MIPS). In the proposed architecture, the Greedy-GA module takes the host list and the VM list and produces an optimal mapping. The Greedy-GA module divides the entire processing among the various members that run in parallel in the module.

The Greedy-Genetic Algorithm (Greedy-GA) attempts to overcome the limitations of the packet aggregation-based mechanism approaches. According to Jian et al. [16], the major contributing factor towards premature convergence is the mutation operator. In this we attempt to reduce the chances of premature convergence by using a self-adjusting mutation operator. Generally, in GA approaches, the mutation rate, that is, the probability of mutation is static. The value of this parameter of GA is defined at the beginning of the GA and remains constant throughout. As a variation to this traditional GA, Jian et al. vary the rate of mutation. Thus here, the mutation probability is dynamic. It is defined to be dependent on a parameter called population differentia. Population differentia is a ratio that is used to indicate the rate at which the different individuals differ from each other. This parameter guides the probability of mutation. The use of this self-adjusting probability of mutation ensures that no premature convergence takes place. In their approach the degree at which 2 individuals, say "A" and "B" differ from each other is given by:

$$d(A, B) = \sum_{j=0}^{l-1} A_j \oplus B_j \quad (1)$$

Where l gives the length of the chromosome. Thus, the total rate at which each of the individuals differ from the rest of the population can be defined as:

$$Population\ differentia = \sum_{i=0}^{N} \sum_{j=0}^{N} \frac{d(A_i, B_k)}{(N-1) \cdot (N-1) \cdot l} * 100 \quad (2)$$

Where N is the population size.

The outline of the GREEDY -GA is described in Algorithm 1. The basic idea in GREEDY -GA is that we divide the entire population into members. In traditional GA, we take an entire population. The various operators of GA, selection, crossover and mutation are applied at once to the entire population across all the generations. Researchers have proved that these steps are the most time-consuming steps in GA. In Greedy -GA, by dividing the population into members and then processing each of these members in parallel, we attempt to enhance the speed of GA.

When employed in a distributed parallel system, the processing of each member may be carried out in parallel, thus greatly reducing the total runtime. Our problem of VM allocation does not define such neighbours. So here, in order to construct the members, we perform simple mutations. The resulting chromosomes which vary, though only slightly, from each other, are placed in the same member. The processing time is further reduced by destroying the members which do not offer any hope of obtaining better individuals. Each member is processed "k" times. If no better individual has been encountered till then, we destroy the member and take the next member. If at least one better individual has been generated from the processing of the current member, then we continue processing the member for "W" iterations. The values of "k" and "W" are determined through experimental evaluations. For each individual in the population, we assess the quality of the individual by calculating the fitness value associated with it. In GA, the fitness value is generally a function of the objectives that we take into consideration. In the proposed approach, the objective that we take into consideration is the physical resource utilization. The algorithm that we used to calculate the fitness value of each individual is outlined in Algorithm 2. An additional precaution has to be taken while employing member gene approach to the VM allocation problem. It should be ensured that all the chromosomes satisfy the constraints. To ensure this, we implement a separate function where each individual is checked for feasibility. In case the individual is found to be infeasible, an attempt is made to transform the infeasible solution into a feasible one. Algorithm 3 takes as input an individual and returns a chromosome representing a feasible assignment.

Algorithm 1 Outline of the Greedy-Genetic algorithm

---

1. Input: Lists of hosts and VMs
2. Initialize the list of hosts and Vms.
3. Initialize the values of parameters of GA and the number of Greedy machine to be constructed.
4. Randomly initialize the population.
5. Compute the fitness values of each chromosome in the population. Calculate the population differentia.
6. Perform crossover and mutation.
7. Select the Greedy heads as the best individuals from the current population.
8. for all Greedy∈ Population do
9. repeat
10. Perform mutation on the family head and insert mutated chromosome into Greedy machine.
11. Compute the fitness value of the chromosome obtained after mutation.
12. if fitness of the mutated chromosome is greater then
13. Add the mutated chromosome to the population.
14. Set flag as true
15. end if
16. until family size
17. repeat
18. Perform crossover and mutation on the current LAST IN machine to get the next generation of the current family.
19. Until 'k' times if flag=true else 'W' times if flag=false
20. Select the fittest individual from the population to get the best solution.
21. end for

The process of on-line resource allocation in cloud data centers include four main phases:

(1) Determining when a host is considered as being overloaded;

(2) Determining when a host is considered as being underloaded;

(3) Selection of VMs that should be migrated from an overloaded host; and

(4) Finding the new placement of the VMs selected for migration [31].

In this research, a reference scenario consisting the best policies reported in [31] for the aforementioned phases are extended by applying our new heuristics. More precisely, this study takes advantage of a resource management solution including Local Regression (LR) for the first phase, a simple method (SM) for the second phase, Minimum Migration

Time for the third phase (MMT), as well as Power &Greedy value & Number of VM Migrations Allocation Genetic Algorithm policy for the fourth phase along with the Enhanced Optimization (EO) policy proposed in our previous study [32] in which the following procedure is adopted for the whole resource management process.

First, the new incoming VMs that have just requested resources are placed on appropriate PMs using GA policy. After that, a resource optimization cycle is repeated in which PMs′ Resource utilizations information is gathered at every management scheduling interval.

In the first step, PMs are searched one by one to find overloaded PMs until there is no more blank machine. Resource utilization values of each PMs are predicted based on the resource utilization history of PMs, using Local Regression (LR) prediction algorithm [31]. If the prediction algorithm forecasts for a PMs that utilization of either one of its resource types will be more than 100%, then this PMs is determined to be an overloaded PMs. After that, VMs residing on overloaded PMs are selected for migration based on Minimum Migration Time (MMT) policy until the elimination of hot spots [31].

In the following step, selected VMs are categorized based on their utilization. Then, a resource allocation procedure is executed for the sorted VMs to find their migration destination using GA allocation policy. GA policy finds the PMs that both have enough resource to host the VM as well as the least power increase after allocation of a VM. If the control system finds a proper destination for a VM, then it is added to the migration list. Following that, underloaded PMs are determined.

In each searching step to find underloaded PMs, the PMs with the least resource utilization is selected as a candidate of being underloaded. VMs from underloaded PMs are added to the migration list until the controlling system cannot find any underloaded PMs. In the following step, selected VMs from underloaded PMs are sorted based on their utilization. If the control system can find proper PMs as migration destinations for all the VMs residing on an under loaded PMs using GA-GREEDY policy, then all its VMs are added to the VM's migration list.

V.RESULT

We proposed a concept for network virtualization scheme that can be used on both non-OpenFlow networks and Open Flow Networks. This scheme will evaluate the CPU problem first before establishing the network virtualization. It will evaluate the average of link utilization from origin sites to candidate remote sites. We hope that this evaluation will decrease the downtime of the process such as moving network address reservation to the remote site. The simulation is done in MATLAB to visualize the performance of the proposed model. Experimentation is done on CPU times for three techniques comparison average result is shown. This is to ensure the validity of the results produced. The comparison of the proposed GREEDY -GA based approach.

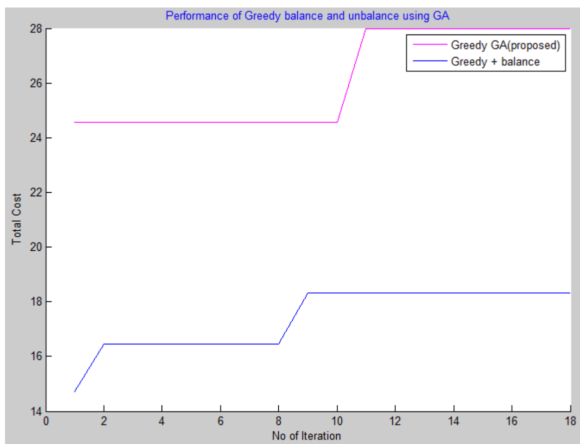

Figure 2: Total cost of the Greedy and GA-Greedy on number of iteration

Experimentation is done on CPU times for three techniques comparison average result is shown. This is to ensure the validity of the results produced. The comparison of the proposed GA-GREEDY based approach.

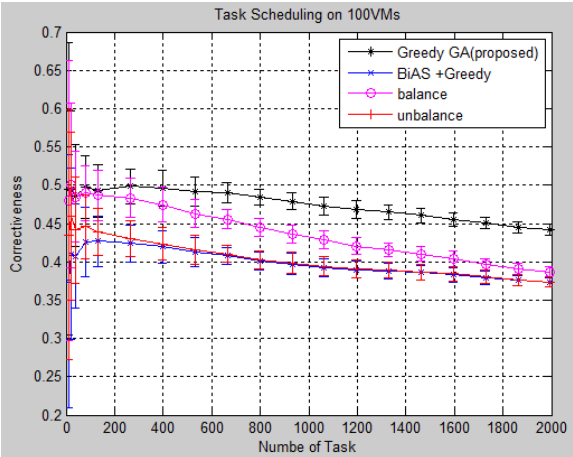

Figure 3: Task scheduling on 100 VMs

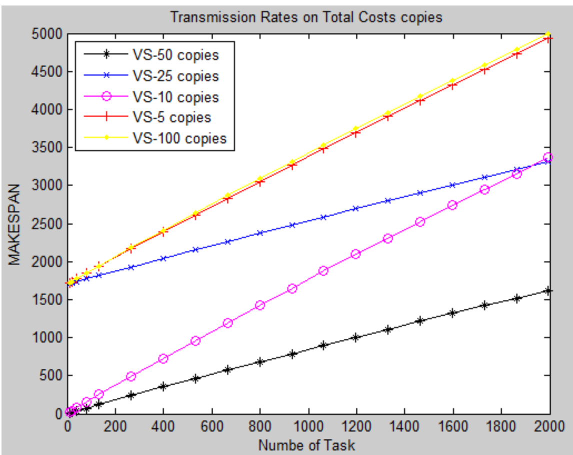

Figure 4: Transmission rates on total costs as Make span on number of task

VI.CONCLUSION

This paper has addressed the issue that current cloud network virtualization techniques lack capability to provide fine-grained QoS management without sacrificing scalability by introducing a staging and caching virtualization concept and a new network virtualization. The proposed GREEDY -GA network virtualization platform nicely solves the problems that cause the scalability issue. Cloud computing system is used to scale the application by maximizing the concurrency and using the resources efficiently. The algorithm takes time utilization and resource utilization into consideration and hence results in high signification.

REFERANCES

[1] ManelBourguiba, Ines El Korbi "Improving Network I/O Virtualization for Cloud Computing" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 3, MARCH 2014

[2]. W. Emeneker and D. Stanzione, "Efficient Virtual Machine Caching in Dynamic Virtual Clusters." In
SRMPDS Workshop, ICAPDS, 2007.

[3]. N. Fallenbeck, H. Picht, M. Smith, and B. Freisleben, "Xen and the art of cluster scheduling," in Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, 2006.

[4]. B. Sotomayor, R. Llorente, and I. Foster, "Resource Leasing and the Art of Suspending Virtual Machines," in 11th IEEE International Conference on High Performance Computing and
Communications, pp. 59–68.

[5]. Zhifeng Yu and Weisong Shi, "A Planner-Guided Scheduling Strategy for Multiple Workflow Applications," icppw, pp.1 -8, International Conference on Parallel Processing - Workshops, 2008.

[6]. LeeCY ,PiramuthuS.Tsai YK, "Job shop Scheduling with a genetic algorithm and machine learning"
InrJ.Pred Res.1997 35(4):1171-1191.

[7]. J. Carretero, F. Xhafa, "A genetic algorithm based schedulers for grid computing systems," International Journal of Innovative Computing, Information and Control, vol.3, no. 6, Dec. 2007.

[8]. H.J. Braun, T. D.and Siegel, N. Beck, L.L. Blni, M. Maheswaran, A.I. Reuther, J.P. Robertson,M.D. Theys, and B. Yao, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," Journal of Parallel and Distributed Computing, vol. 61, no. 6, pp.810–837, 2001.

[9]. SaswatiSarkar "Optimum Scheduling and Memory Management in Input Queued Switches with Finite Buffer Space", IEEE 1373, 2003.

[10] R, Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud Computing and emerging IT platforms: vision, hype, and reality for delivering IT services as the 5th utility, Future Generation of Computer Systems, 25 (2009), 599– 616.

[11] Gartner identifies the Top 10 Strategic Technologies for 2008", Gartner, 2007.

[12] "Report to Congress on Server and Data Center Energy Efficiency", Public Law 109-431,U.S.Environmental Protection Agency, ENERGY STAR Program, Aug 2, 2007.

[13] Energy Efficiency in Data Centers: Recommendations for Government- Industry Coordination U.S. EPA's National Data Center Energy Efficiency Strategy Workshop, Jul 8 to Oct 16, 2008.

[14] IBM Energy management White paper on Realizing the value of the green data centerbyintegrating facilities and IT' May 2008.

[15] "Five big questions about cloud computing" By RWME staff, Published: December 27, 2009 on http://www.rwme.net/v1/inside.aspx?v=1&nid=900&sec=news&cat=1

[16] Jian, Z., Wang, S.. Study on self-adjusting of gene migration genetic algorithm. Journal of Xi'an Jiaotong University 2002;11