International Journal of Research in Engineering, Science and Management
Volume-2, Issue-1, January-2019
www.ijresm.com | ISSN (Online): 2581-5792

Energy Conservation by Consolidating VMs in Cloud Environment

Jagruti R. Desale
PG Student, Department of Computer Engineering, SSVPS B. S. Deore College of Engineering, Dhule, India

***Abstract*: Energy utilization has been a principle worry to the earth as the size of cloud server becomes distinctly bigger across world because of the simplicity of web use and handling on cloud. As a result, the energy utilization develops quickly. Additionally, they contribute in to the natural downsides like carbon emanation. Virtualization advancements give the capacity to exchange virtual machines between the physical machines utilizing live VM relocation in distributed computing. ACS Dynamic server consolidation is a productive approach for energy protection in cloud by diminishing the aggregate number of dynamic physical machines. Since VM consolidation issue is entirely NP-hard hence ACS VM consolidation reduces energy consumption of data centers with Minimum SLA violation and number of VM migration.**

***Keywords*: ACS, Cloud Computing, Energy Efficient, SLA, VM consolidation.**

1. Introduction

Distributed computing is the conveyance of processing as an administration as opposed to giving as an item, whereby shared assets, software's, and data are given to the PCs and different gadgets as an utility (like power lattice) over a system (ordinarily the web). The underlying thought of cloud computing is that associations don't deals with their IT framework, however have it conveyed as an administration by a Cloud Service Provider (CSP). Due to the quick development of distributed computing in the Information Technology scene, numerous definitions have risen. Distributed computing is connected with another worldview for giving distinctive processing assets, more often than not are tended to form three key angles which are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Likewise, the four organization models are private cloud, community cloud, public cloud and hybrid cloud [1]. There is essentially two ways broadly utilized as a part of which the power and size can lessen that are Virtual Machine (VM) union and Dynamic Server Provisioning. In proposed approach, the VMs are at first designed PMs in light of Based Fit Decreasing (BFD) calculation. High energy utilization does not mean high working expenses but rather increments higher carbon outflows. Dynamic server provisioning approaches spare power by driving just a base measure of assets expected to fulfill the workload prerequisites. Subsequently, extra servers are brought disconnected or put into low-control mode if the workload request diminishes. The previous couple of years, there are different endeavors [1]. The ACS VM consolidation algorithm plot proposes energy conservation of data centers with accomplishing the fancied level of Quality of Service (QoS) between cloud suppliers and their clients

A. VM Consolidation

VM consolidation is the one of the technique used to reduce the energy consumption of data centers. The number of active PMs by migrating and consolidating the VMs into lessen number of physical machines. To periodically characterize an application's resource requirements and resource usage is the main challenge in consolidation. Consolidation includes two strategies: VM placement and VM migration. In VM Placement process it helps in selecting the appropriate host for the given VM. Selecting the proper host which best suits VM is one of the fascinating assignment in the relocation procedure, since wrong determination of host can expanded the quantity of movement, asset wastage and energy utilization. In this way using VM placement energy saving can be perform by shutting down some servers.

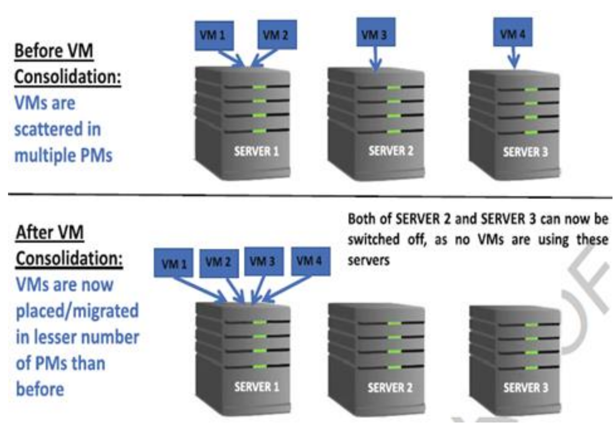

Fig. 1. Overview of VM migration

VM Migration is the another strategy of VM consolidation which carried out after the initial VM placement in order to lessen the number of running physical machines by migration of few VMs and consolidate them into lessen number of PMs. There are four steps in VM machine migration process. First, to choose the PM which is overload or under loaded, second to choose one or more VM, third to select the PM where chosen

VM can be found and last the VM movement. Consolidation mainly classifies into two types- Static Consolidation and Dynamic Consolidation. In static consolidation when VM is require then no migration takes place, new VM is placed to PMs for processing. While in Dynamic Consolidation when necessity occurs, the VMs are migrated from one PM to another. Figure 1 shows the overview of VM consolidation using VM migration [3].

2. Related work

VM consolidation techniques have been very attractive to reduce energy costs and increase resource utilization in virtualized data centers. Consequently, a good amount of research works have been done in this area and depending on the modeling techniques used, different problem solving techniques are proposed.

Beloglazov and Buyya [4] proposed the overload detection algorithm based on an idea of setting CPU utilization threshold categorizing the non overload and overload states of the host. It compares the current CPU utilization of the host with the defined threshold. If the threshold is exceeded, the algorithm detects a host overload. An example of static CPU utilization threshold based algorithm is the averaging threshold-based algorithm (THR). But as the fixed values of utilization thresholds are suitable for an environment with static and predictable workloads, in which different types of applications can share a PM. The system should able to adjust the utilization threshold depending on the workload patterns exhibited by the applications. Therefore, this approach is not suitable in dynamic workload.

Beloglazov and Buyya proposed in the later work [2] based on a statistical analysis of historical data collected during the lifetime of the VMs an auto-adjustment of the utilization thresholds. It defines to adjust the value of upper utilization threshold depending on the strength of deviation of the CPU utilization. The higher the deviation, then lower is the value of the upper utilization threshold the more likely that the highest CPU utilization and cause an SLA violation. Example algorithms are Median Absolute Deviation (MAD) and Interquartile Range (IQR) and Local Regression (LR). These Adaptive utilization threshold algorithms are more robust than static CPU utilization threshold algorithms in case of dynamic environment. But, it provides poor prediction of host overloading.

M. Ferdaus [3] proposed the AVVMC based VM consolidation approach that focuses on balanced resource utilization of servers across different computing resources CPU, memory and network I/O with the goal of minimizing power consumption and resource wastage. It uses Ant Colony Optimization metaheuristic with balanced usage of computing resources based on vector algebra. However, it has migration and re-configuration overheads.

The resource utilizations of VMs continue to vary over time due to dynamic workloads. So according to workload it is necessary to adapt and optimize the VM placement periodically. ACS VM consolidation is optimization technique for dynamic virtual machine consolidation to reduce energy consumption with satisfying QoS requirements by reducing SLA violation and number of VM migrations [1].

3. System architecture

Cloud Dynamic server provisioning approaches spare energy by utilizing a decreased measure of assets required for fulfilling the workload necessities. Subsequently, extra servers are turned off or put into a low power mode when the workload request diminishes.

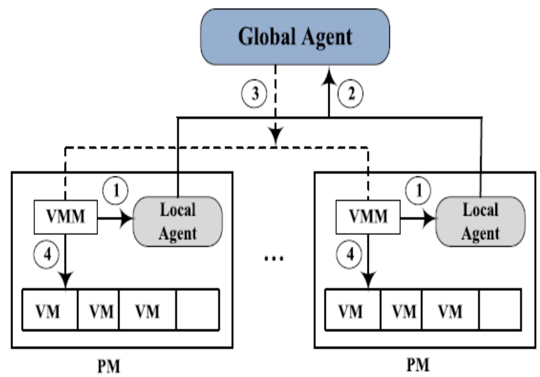

Fig. 2. System architecture

The ACS is applied to VM consolidation problem [1]. Figure 2 shows the multi-agent system architecture for dynamic VM consolidation. In this approach, the targeted system is an IaaS environment, represented by a large-scale data center consisting of N heterogeneous physical machines. Each node is characterized by the CPU performance which is defined in Millions Instructions per Second (MIPS), amount of RAM and network bandwidth. The type of the environment implies no knowledge of application workloads and time for which VMs are provisioned. Multiple independent clients submit requests for provisioning of M number of heterogeneous VMs characterized by requirements to processing power defined in MIPS, amount of RAM and network bandwidth. The fact that the VMs are managed by independent users implies that the resulting workload created due to combining multiple VMs on a single physical node is mixed. The mixed workload is formed by various types of applications, such as HPC and web-applications, which utilize the resources simultaneously. The users establish SLAs with the resource provider to formalize the Quality of Service (QoS) delivered. The provider pays a penalty to the users in cases of SLA violations. The system is tiered comprising local agent and global agent as shown in Figure 2. The local agent resides on each physical. Their objective is the continuous monitoring of the physical node's CPU utilization resizing the VMs according to their resource needs. The global manager resides on the master node and collects information from the local managers to maintain the overall view of the utilization of resources. The global manager issues commands for the optimization of the VM placement and builds global best migration plan using proposed ACS VM consolidation algorithm and sends command to the VMM. After receiving command VMMs perform resizing and migration of VMs as well as changes in power modes of the nodes [1].

4. ACS VM consolidation algorithm

Optimization metaheuristic algorithm is computational methods that take inspiration from the foraging behavior of some ant species. In Ant Colony Optimization, a number of artificial ants build solutions to the considered optimization problem by choosing feasible solution components and exchanging information on the quality of these solutions via pheromone [1].

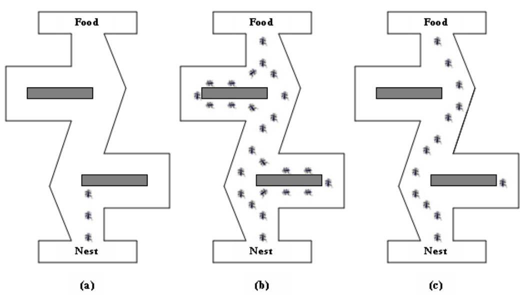

Fig. 3. Ant colony behaviors

The ACS VM consolidation creates a set of tuples T from which each tuple t consist of 3 elements- source PM, destination PM and VM to migrate as given in (1)

$$t = (p_{so}, v, p_{de}) \qquad (1)$$

A. Objective function

Migration plan is the output of the VM consolidation algorithm, which results into the lessen number of active PMs without compromising their performance. As the main objective of this algorithm is to reduce the number of active PMs, the objective function can be defined as number of shutdown or sleeping PMs. The objective function is,

$$f(M) = |P_s|^{\gamma} + \frac{1}{|M|} \qquad (2)$$

Where, $= |P_s|^{\gamma}$is set of PMs in sleep mode and M is migration plan.

When selected migration plan is prefer, ACS VM consolidation algorithm selects migration of VM to the already active PMs. Hence when it is not possible to migrate VMs to already active PMs in that case PMs in sleeping mode can be activated. The PM is said to be in sleeping mode when it is not hosting any VMs. The set of sleeping PMs can be defined as-

$$P_s = \{ \forall_{\mathrm{p}} \in P \mid V_p = \emptyset \} \qquad (3)$$

Where, V_p=set of VMs running on PM p.

B. Stochastic state transition rule

In ACS VM consolidation approach, the pheromone is deposited on the tuples defined in (1). Each nA ants uses a stochastic state transition rule to choose the next tuple to traverse by ants. This state transition rule in ACS approach is called as pseudo-random-proportional-rule. According to this rule, an ant k chooses a tuple s to traverse next by applying (4) Where, τ is amount of pheromone, η is heuristic value

$$s = \begin{cases} \arg max_{u \in T_k} \{ [\tau_u].[\eta_u]^{\beta} \}, & if\ q \le q_0 \\ S, & otherwise, \end{cases} \qquad (4)$$

The probability $\mathrm{p_s}$ of an ant k to choose tuple s to traverse next is defined as-

$$\mathrm{p_s} = \begin{cases} \frac{[\tau_s].[\eta_s]^{\beta}}{\sum_{u \in T_k}[\tau_u].[\eta_u]^{\beta}}, & if\ s \in T_k, \\ 0, & otherwise \end{cases} \qquad (5)$$

C. The heuristic value

The heuristic value of a tuple s is defined as

$$\eta_s = \begin{cases} (|C_{p_{de}} - (U_{p_{de}} + U_v)|_1)^{-1}, & if\ U_{p_{de}} + U_v \le C_{p_{de}} \\ 0, & otherwise \end{cases} \qquad (6)$$

The heuristic value does VM migrations that result in a minimize under-utilization of PMs. The constraint $U_{p_{de}} + U_v \le C_{p_{de}}$ prevents migrations that would result in the overloading of the destination PM. The stochastic state transition rule in (4) and (5) prefers tuples with a higher pheromone concentration and which result in a higher number of released PMs [1].

D. Global and local pheromone evaporation rule

ACS also uses a global and a local pheromone trail evaporation rule. The global pheromone trail evaporation rule is applied at the end of iteration when all ants complete their migration plans. The global update rule is defined as-

$$\tau_s = (1 - \alpha).\tau_s + \alpha\,.\Delta^{+}_{\tau_s} \qquad (7)$$

Where, $\Delta^{+}_{\tau_s}$is additional pheromone given to the set of tuples from global best migration plan and it is defines as-

$$\Delta^{+}_{\tau_s} = \begin{cases} f(M^{+}), & if\ s\ \epsilon\ M^{+} \\ 0, & otherwise \end{cases} \qquad (8)$$

The local pheromone trail update rule is applied on a tuple when an ant traverses the tuple when making its migration plan. It is defined as

$$\tau_s = (1 - \rho)\,.\,\tau_s + \rho\,.\,\tau_0 \qquad (9)$$

Where, τ_0 is initial pheromone level which is define as-

$$\tau_0 = (|M|.|P|)^{-1} \qquad (10)$$

In ACS the pseudo-random-proportional-rule and the global pheromone trail update rule are helps to search more accurate.

The pseudo random proportional rule gives tuples with highest heuristic value and higher pheromone level. So that ants can search other quality solution path in near about approximate of the thus far global best solution [1].

E. Algorithm: Ant Colony System (ACS) VM consolidation

BEGIN
Create set of tuples;
DO
FOR each *Ti* do
FOR each ant k do
FOR each tuple t do
Generate random variable;
IF $q > q_0$ then
Compute probability to choose next tuple to traverse by formula (5);
END IF
Choose next tuple to traverse by formula (4);
Add t to temp migration plan;
Apply local pheromone rule on t;
Update $U_{p_{de}}, C_{p_{de}}$;
IF temp migration plan>best score Then
Add t to migration plan of ant k;
Else
Subtract t from migration plan of ant k;
End if
Add temp migration plan to set of migration plans;
END FOR
Select global best migration plan;
Apply global update rule;
END FOR
END

5. Result analysis

This Section is devoted to evaluate the performance of ACS VM Consolidation algorithm. This paper is focus on reducing energy consumption of data centers with minimizing SLA violation and number of VM migration. Result analysis was conducted on HP PC with Intel i2 CPU and 4 GB of memory running window 7 and Cloud Sim 3.0. To evaluate the efficiency of ACS VM Consolidation algorithm, the experimental environment is set up using the Cloud Sim toolkit. Cloud Sim is a discrete event simulator for implementation and evaluation of resource provisioning and VM consolidation techniques for different applications.

Table 1
Comparison values with ACS Algorithm

Parameter	LRA	AVVMC	ACS
SLA	25	18	15
Energy Consumption	0.16	0.09	0.03
VM migration	6	5	3

We simulated a data center comprising Random heterogeneous PMs and selected two server configurations in Cloud Sim: HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores 1860 MHz, 4 GB), and HP ProLiant ML110 G5 (Intel Xeon 3075, 2 cores 2660 MHz, 4 GB). Dual-core CPUs are sufficient to evaluate resource management methods that are designed for multi-core CPU architectures. Moreover, it is important to simulate a large number of servers for performance evaluation of VM consolidation methods. To evaluate the efficiency of ACS VM Consolidation approach, three parameters are measured: SLA Violations, Energy Consumption and number of VM Migrations. The results are based on a random workload. In the random workload, the users submit requests for the provisioning of heterogeneous VMs randomly.

A. ESLA violation

The users establish SLAs with the resource provider to formalize the Quality of Service (QoS) delivered. QoS which describes performance level of system in terms of maximum response time and minimum throughput of the system to the end users. The ACS VM Consolidation prevents SLA violation using prediction of overloaded PMs and using heuristic value in equation (6) ensures that destination PM does not become overloaded when a VM migrates on it.

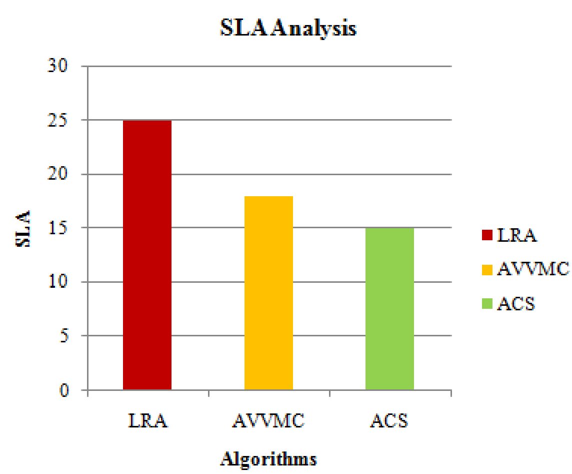

Fig. 4. Comparison graph for SLA

Fig. 4 shows comparison of ACS algorithm with existing algorithms LRA and AVVMC. It shows ACS algorithm reduces number of SLA violation.

B. Energy consumption

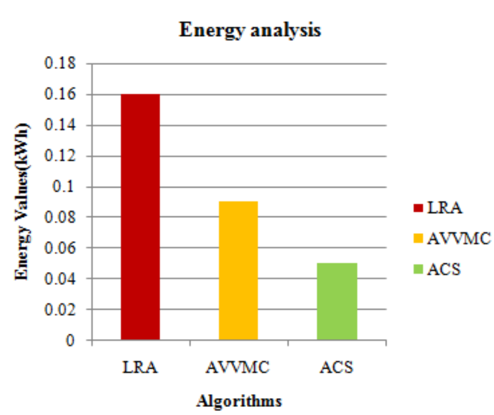

Fig. 5. Comparison graph for Energy Consumption

The energy consumption of a PM depends on the utilization of CPU, memory disk and network card. But most studies show that CPU consumes more power than memory, disk storage and network interface. Therefore, resource utilization of PM is usually represented by its CPU utilization.

Fig. 5 shows comparison between ACS algorithm with LRA and AVVMC. It shows that ACS VM consolidation outperforms existing VM consolidation approaches in terms of energy consumption because the defined objective function in equation (2) in ACS tries to maximize number of dormant PMs by packing VMs into the lessen number of PMs that have enough capacity.

C. Number of VM migrations

Live VM migration is a costly operation that includes some amount of CPU processing on the source PM, the link bandwidth between the source and destination PMs, the downtime of the services on the migrating VM, and the total migration time. Therefore, one of our objectives was to minimize the number of migrations. ACS VM consolidation approach reduces number of migration because it creates a migration plan that has requires the minimum number of migrations.

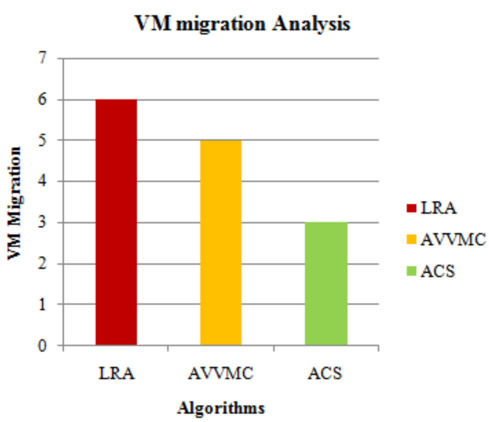

Fig. 6. Comparison graph for VM Migration

Fig. 6 shows comparison of ACS approach with existing LRA and AVVMC algorithms. From figure ACS shows minimum number of migrations compared to others.

6. Conclusion

Cloud data centers provides on demand delivery of computing resources on internet. As there is increase in size of data centers across the world, it consumes more energy for their operation which results into CO2 emission. Because of the increase in size of data centers and inefficient resource usage energy conservation technique like ACS VM Consolidation is required. The propose ACS VM consolidation algorithm reduces the energy consumption of data centers as compare with existing algorithms like LRE and AVVMC. It consolidates VMs into reduced number of PMs, which also helps in minimizing SLA violation and Number of VM migrations. Hence as comparing with the existing algorithms like LRA and AVVMC, the ACS VM Consolidation algorithm reduces energy consumption of data center with minimum number of migrations and minimum SLA violation.

References

[1] Adnan Ashraf, Tapio Pahikkala, Fahimeh Farahnakian, "Using Ant Colony System to Consolidate VMs for Green Cloud Computing," *IEEE Transaction on Service Computing*, vol. 08, pp. 187-198, March/April 2015.

[2] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristic for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Comput.: Prac. Exp*, vol. 24, pp. 1397–1420, 2012.

[3] M. Murshed, R. Calheiros, and R. Buyya M. Ferdaus, "Virtual machine consolidation in cloud data centers using ACO metaheuristic," *Proc. 20th Int. Eur. Conf. Parallel Process*, vol. 8632, pp. 306-317, 2014.

[4] J. Abawajy, and R. Buyya A. Beloglazov, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Comput. Syst.*, vol. 28, pp. 755-768, 2012.

[5] A. Ashraf, "Cost-efficient virtual machine management: Provisioningsioning, admission control, and consolidation," Abo, Finland, Europe: TUCS Disseratios, Oct 2014.

[6] R. Buyya, Y. C. Lee, and A. Zomaya A. Beloglazov, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Adv. Comput.*, vol. 82, pp. 47-111, 2011.

[7] M. Harchol-Balter, R. Raghunathan A. Gandhi, "AutoScale: Dynamic, robust capacity management for multi-tier data centers," *ACM Trans. Comput. Syst.*, vol. 30, no. 04, pp. 14:1-14:26, 2012.

[8] M. Dorigo and L. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evolutionary Comput.* , vol. 01, pp. 53-66, Apr. 1997.

[9] G. Di Caro, and L. M. Gambardella M. Dorigo, "Ant algorithms for discrete optimization," *Artif. Life*, vol. 05, pp. 137-172, Apr. 1999.