Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-020-02026-7

ORIGINAL RESEARCH

Cloud management architecture to improve the resource allocation in cloud IAAS platform

J. Srinivasan[1] · C. Suresh Gnana Dhas[2]

Received: 20 February 2020 / Accepted: 24 April 2020

Abstract
Cloud computing (CC) is an attractive emerging technology due to offering services based on-demand by the process of virtualization. Since CC platform offers services based on-demand it has been widely used in the field of various emerging IT infrastructure. In cloud platform each application is run in individual virtual machine (VM) for execution of services within the host. Since cloud platform operates on on-demand service it need to cope with multiple application in single time hence it is necessary to adopt an effective approach for balancing memory utilization in cloud network. For effective utilization of available memory existing approaches uses probability distribution method for allocating resources in cloud platform but still there exists a lack of utilization of available memory in cloud platform. This paper aims to develop an effective approach for dynamic memory allocation in VM in cloud platform. For memory allocation among VM in cloud platform proposed approach uses cloud vertical elasticity manager (CVEM), memory reporter (MR), memory over subscription granter (MOG). The MOG uses a scheduler to allocate the memory in a dynamic way inside a host. Finally, we adopt host elasticity rule to balance the available memory to allocate dynamically the memory inside an available host in cloud.

Keywords Cloud vertical elasticity manager · Memory reporter · Memory over subscription granter · Dynamic memory allocation

1 Introduction

In recent years, cloud computing (CC) is emerging information technology (IT) service delivery model offers essential characteristics like elasticity, resource on demand, broad network access, resource pooling, and measured service (Beltran et al. 2016). These characteristics makes the CC services as an emerging technology. Various technologies evolved in CC services includes distributed computing, virtualization, service-oriented design, utility computing, storage, networking, etc. (Baranwal and Vidyarthi et al. 2015). Out of which, the resource allocation (RA) algorithms is considered as an important target in recent era (Yuvaraj and Suresh Ghana Dhas 2018; Yuvaraj et al. 2013, 2018, 2019; Sivaram et al. 2019).

The target is regarded as a group of services that understands a service in the form of a separable job onto VM groups. The resource usage is optimized to offer better quality of service (QoS) and it further limits the migrations. Various solutions in this perspective is developed for the purpose of RAs and this includes bin packing algorithms. Even though there is no clear agreement within the community on that aspects of the matter area unit, the necessity on an algorithmic model requires such consideration (Beaumont et al. 2016).

The streamlined edges of central desk-based operation, commonplace protected installation, data security and simpler clients control on virtual desktop-clouds (VDCs) (Yan 2011) have now become increasingly obvious to the business community virtualisation by Desktop-as-a-Service (DaaS) offers. Above all, in virtual desktops (VDs), the expected client QoE is prone to network loss and cannot withstand disruptive patterns of cross traffic (Fiedler et al. 2010; Calyam et al. 2014).

✉ J. Srinivasan
jsrinivasan.bu@gmail.com

C. Suresh Gnana Dhas
sureshc.me@gmail.com

1 Research and Development Centre, Bharathiar University, Coimbatore, Tamilnadu 641046, India

2 Department of Computer Science and Engineering, Ambo University, P. O. Box 19, Ambo, Ethiopia

Published online: 24 May 2020

In (Molto et al. 2016), a system is designed that is integrated with CMP to automatic supply of vertical snap in order for adapting to the VM memory size to its present consumption of memory and this includes migration to overload. The VM-per-host allows increased relation while maintaining the VM service, since the memory is inflated dynamically (Beltran 2016; Calyam et al. 2013; Ficco et al. 2016, 2018; Galante and Bona 2012; Lin et al. 2013; Mell and Grance 2011).

1.1 Impact of VM in cloud

In (Beaty et al. 2009) the authors form the event of snap policies through formalized and dependable. This approach imposing snap through the dynamic representation and on-line quantitative verification of Markov call Processes (MDP) cloud probabilistic model checking.

In second stage planned concrete snap models and connected snap policies. The policies for each real and artificial data sets in clusters of NoSQL databases (Naskos et al. 2014).

From analyzes it is ascertained that planned approach improves upon the progressive in considerably increasing user-defined utility values and decreasing user-defined threshold violations. This job given a proper, probabilistic model checking based approach using cluster of VMs. The planned MDP snap models and associated probability policies that think about the dynamic representation of such models.

1.2 Impact of dynamic memory allocation

For increasing convenience and recognition of cloud storage as a service (STaaS) alternatives to ancient on-line video recreation models, that think about expensive content delivery networks (CDNs) has been planned in (Barba-Jimenez et al. 2016).

This can be a powerful commitment for a VoDservice, as a result of ancient cloud approaches typically target a best paradigm optimizing performance, cost, and information measure, among different parameters.

1.3 Impact of CSP in cloud

Another developed a unique tool which will capture important quality metrics like web Utility and repair latent period, which may be accustomed quantify VDC platform readiness. The CSPs is permitted by the tool, where the researchers verify RAschemes in two modes: Run Simulation and Run Experiment (Calyam et al. 2014).

The GraspCC feed provides an almost optimum estimation of the quantity for each workflow of VMs. GraspCC-fed expands to include research working channels in every single provider and unified clouds a previously planned GRASP aid for requests (Coutinho et al. 2015). The GraspCC-fed tests are combined with the SciCumulus operating stream system for clustering. So GraspCC-Fed is an important tool for users in nursing that will allow you to determine for the digital cluster the best configuration of parallel cloud-based science work flows (Yuvaraj et al. 2018, 2019).

In past researches, it is therefore difficult to manage routing complexities within the cloud network specialized throughout network optimization techniques for handling the degradation in performance. The research group has built RA schemes such as (Deboosere et al. 2012; Beaty et al. 2009) to extend the QoS over user defined protocol variations with VDCs (Calyam et al. 2014).

A limited tools in the CC environment limits the number and type of activities scheduled on a computer. Various The set of jobs of all types are scheduled for simultaneous utilization on a server, which is called initialization. The external modules on a database has the capability of handling the servers. The overall cloud capacity region is then Hermann Minkowski addition of all server capability regions (Maguluri et al. 2014).

This paper focus on CC services, which provide technology as a product. Users in this method submit applications on VMs for services. Various application specifies the resources that it requires processor power, storage, etc. Initial requests are sent to the cloud service provider and scheduled on physical machines called the servers.

The novelty and the contribution of paper is stated below: The MR, CVEM and MOG experiments were integrated in the database management platform in which CVEM measure the space allocation of the VM periodically and updates it to the cloud platform. MR measures storage consumption and free memory is transferred to the cloud platform based of data. MOG measures the user amount of memory and then the experiments namely CVEM, MR and MOG are built into cloud platform, where a planner is configured to automatically delegate storage space to VM. Additional rule of elasticity is adopted to handle the allocation on cloud memory.

2 Cloud management platform

Topology of the CC center into consideration is shown in below figure one. These techniques shut down a server pool to save lots of owner if its servers don't seem to be presently serving any job.

3 Proposed architecture

The aim of this research is to improve the CC performance through incorporation of VM for memory allocation. To improve memory allocation VM utilizes cloud vertical elasticity manager (CVEM), memory reporter (MR) and memory oversubscription granter (MOG) in cloud platform. The most downside in cloud platform is expressed in below Fig. 1.

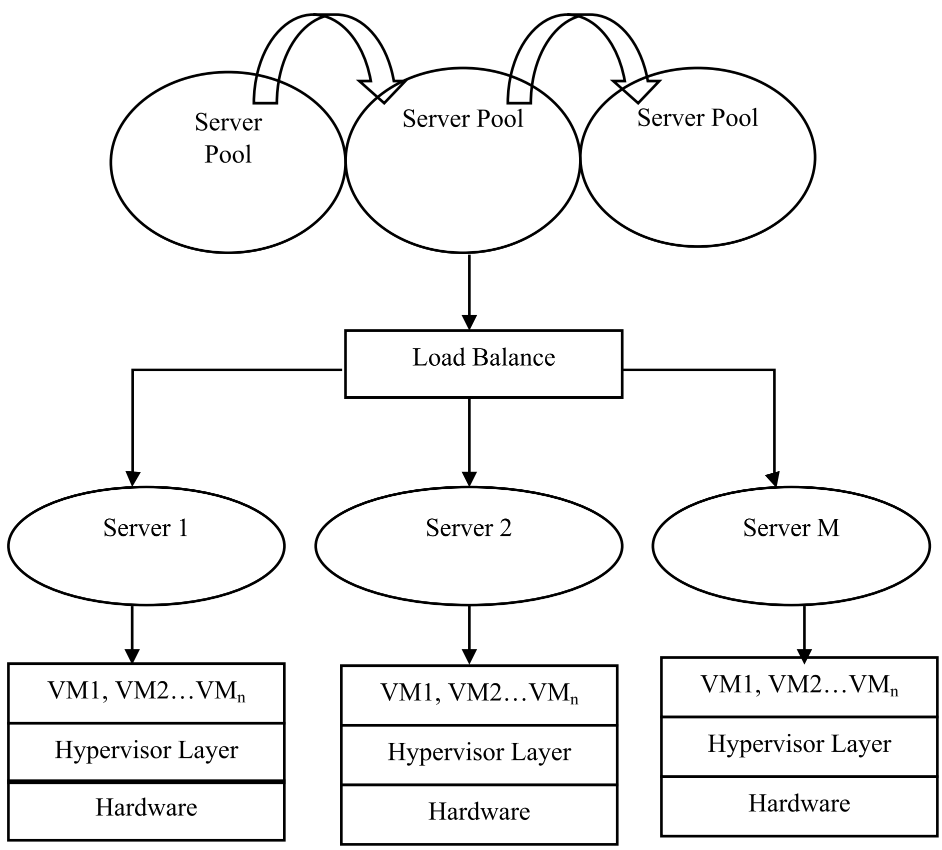
At first, in KVM, memory requirement are highly supported by the hypervisor total content. In second stage, OpenNebula (Molto et al. 2016) related to the cloud management platform supply through significant management of VMs cycle with consideration of infrastructure in physical layer. As per the (Badger et al. 2012) distinct mechanism is adopted for the optimization of energy consumption on total VM unit that mitigates the over subscription issues evolved in memory allocation:

(i) Memory loss which occurs in hypervisor resources by the loaded constant VMs running at physical host of system;
(ii) In case of VMs the amount of offline migration is evaluated through physical machine loading capability;
(iii) Migration of resources memory based on-demand in VM performs effectively in physical machine through loaded VM one;
(iv) Disks allocated for streaming transforms VMs minimal portion to disk which are native in VM to start off in another machine located in physical layer, and
(v) Memory of network uses another machine for memory allocation to swap the file. In this research memory allocation migration techniques are integrated in CMP process. The analysis illustrated that migration techniques are supported by hypervisors for many OS such as UNIX and Windows. This allows system to integrate simply for on-demand cloud network addition to VMs with resettled host in physical layer for standard service infrastructure in CMP.

The Cloud VAMP design for on-demand memory allocation includes 3 components such as:

- *Cloud vertical elasticity manager (CVEM)* CVEM evaluate the memory required for VMs to complete the particular tasks and dynamically allocates the allotted memory through the formulation of customized rules. The design of rules is based on agents related to CMP and hypervisors process in the system which access the on-demand physical nodes in Cloud. The CVEM perform live migration in VM to restore overloaded memory allocation.
- ***Memory reporter (MR)*** In MR, the agent runs within the VM and provides detailed report about system observance for free space, used memory size and swap file utilization for application lies in the VM. Those data

Fig. 1 Topology of cloud center



are highly accessible by CVEM with CMP's integrated observance system or third-party observance system hopping.

- ***Memory over subscription granter (MOG)*** A system with CMP related to memory number relies on host physical layer which takes consideration of CMP of the computer hardware.

The concept of Open Nebula implementation it integrates significantly on the integration process. Based on the installation Open Nebula performs installation for majority of services located in node at front-end whereas VMs are interior operating nodes deployment scenario. In case of KVM hypervisor are put in for analysis. The Cloud VAMP design involves Python-based model for analysis. For analysis CVEM runs on concern of observance data for usage of particular memory within the infrastructure of VM. The MR agent effectively reported about memory utilization through adoption of querying member for entire VM free memory due to utilization of disk space. Through the adoption of mechanism of contextualization in Open Nebula VM runs dynamically. The CVEM dynamically access the centralized concerning observance data for memory usage with deployment based on-demand cloud (Praveenchandar et al. 2020).

4 Mathematical design

In hardware design effective memory allocation strategy is evaluated through over subscripted in the host memory which gets changed in version of the KVM VM manager (VMM). The VMM calculated the memory purloined for individual host and provides instruction and guidelines to physical host based on the total avail memory quantity of the CVEM which are reclaim by different VMs runs within the physical host. In the present case, KVM memory utilization decides CVEM which can be shrink or enlarge for the allotted memory in VMs with counting memory usage. The allotted memory (AM) at present to VM splits according to Used memory (UM) of the running applications within the device and free memory (FM) and the resultant memory is denoted as AM = UM + FM. The application of vertical elasticity rules in Cloud VAMP presented in existing research jobs focused on MOG for the 200$^{\text{th}}$ unallocated memory in VM. The aim is to manage the free memory availability based on the application running through VM which requires huge memory sharing capability. Through application of elasticity rule vertically solely trigger the free memory share of the smaller VM than eighteen or higher than 100 and 20th MOP in the cloud. This facilitates the solely performance over substantial changes used by the VM which removes gratuitous memory changes fluctuations. Under this circumstance Cloud VAMPP dynamically adopts memory size of VM (1),

$$AM = UM \times (1 + MOP) \tag{1}$$

In above equation AM in VM is denoted as newly allocated memory, in this AM is allotted to VM by hypervisor. The UM denotes the application of used memory by VM. The MOP of 200 denoted the elasticity rule triggered solely by the VM free memory.

The application of elasticity rule vertically in VM employs the 2500 MB and downstream capability value of 1200 MB for allotted memory. On the total available memory 1000 MB are utilized by the appliances and 200 MB were used as free memory (FM) which offers MOP value of 20% in UM. When the UM exceeds increase memory threshold (IMT) or decrease memory threshold (DMT) as per elasticity rule vertical focused on 200th FM in VM. In order to withstand oscillations elasticity rule does not trigger or alter the changes in memory consumption by the elasticity rules.

According to rule of elasticity mechanism is complemented based on occurred thrashing intervals in VM. The size of memory is increases ought to intervals occurrence in VM in huge amount with devastating effects which thrashes the performance of application (Beltran 2016). Through the adoption of exponential backoff strategy significant mechanism is adopted (Naskos et al. 2014). Within the VM free memory is not accessible for every 500th memory location almost all memory.

In case free memory is not available within the VM than 500th different memory is available for present assigned allotted memory. The memory allocation in VM facilitates fast increase in avoiding thrashing attainable performance of the VM. In case if the memory is of high demand than equal memory allocation is performed by hypervisor in VM. At last, for final interval of observation leads to memory shortage within the VM which resulted in performance of running application in VM which request quick memory request at higher rate in the Cloud VAMP for allotted VM memory where VM allocates most of the memory available. In allotted VM memory final steps are corrected in Cloud VAMP with reduction of rule engaged in memory allocation through system of self-regulation.

5 Scheduler design

Hardware changes are effectively monitored by the KVM-VMM with respect to changes in hardware with consideration of physical host of memory oversubscribe. In market the memory quantity lies within the host accordance with the system observance based on the physical memory quantity obtained via observance system particularly and shares data through memory quantity which are apart from the free memory lies within the VMs. The hardware lies in the VMs are memory reduction and estimated the memory quantity of the VMs with the one host due to memory host minus required memory of deployed VMs as shown in Fig. 2.

$$HostVM_N = Host_N - \sum_{VM_{host}} VM_N \tag{2}$$

Using this approach, the computer hardware can act as if the hosts had a lot of memory on the market for the VMs and can attempt to deploy new VMs within the physical host notwithstanding the full quantity of memory requested by the VMs is larger than the physical memory on the market at the destination host. The worth of O is designed for the on-premises Cloud so as to extend the degree of memory over subscription. The proportion thus a price of 1/3 means no memory over subscription are going to be introduced by Cloud VAMP this implies that the total of allotted memory of all the VMs of a number within the on-premises Cloud can ne'er exceed the on the market memory of that host. Cloud VAMP can attempt to perform the maximum amount over subscription as doable to reclaim all the free memory from the VMs to alter most over subscription, since the CMP computer hardware can allot extra VMs to the underlying hosts. Notice that this could need to migrate VMs a lot of often if applications begin difficult extra memory. Also, below no circumstances Cloud VAMP can reclaim memory being employed from the VMs since that will have a dramatic impact on the performance of applications. In the end, this parameter ought to be properly fine-tuned betting on the necessities of the on-premises Cloud.

As a final remark, sort of applications that need low latency responses might like to not be live migrated to alternative hosts, which could have a bearing on its performance and therefore the level of service expected by the consumer. If a Cloud supplier desires to run applications that area unit terribly sensitive to performance, this may be supported by allocating the VMs that run those applications to a set of hosts that may not be monitored by Cloud VAMP. This way, the allotted memory to those VMs get reduced and applications can run based on the requested resources while not being migrated to alternative hosts. Also, notice that the goal of Cloud VAMP not allot a lot of resources to extend the performance to reclaim the unused resources. It is necessary to considered that allotted memory of a VM that's presently not being employed application for alternative VMs to use it. Of course, betting on the memory consumption patterns, the applying may need the additional memory back and this may introduce a performance penalty. In the end, these techniques is more focuse for a particular on-premises Cloud betting on the employment and application characteristics.

$$PE_i = LinearPower\left(\frac{\left(P_{\max} - P_{\min}\right) \times U_i}{100}\right) \tag{3}$$

where $P_{\max}$ and $P_{\min}$ is the maximum and minimum power consumed by PDi respectively. Utilization of Data Center can be calculated by

$$U_i = \left(\frac{(Total\_VM - Allocated\_VM)}{Total\_VM}\right) \tag{4}$$

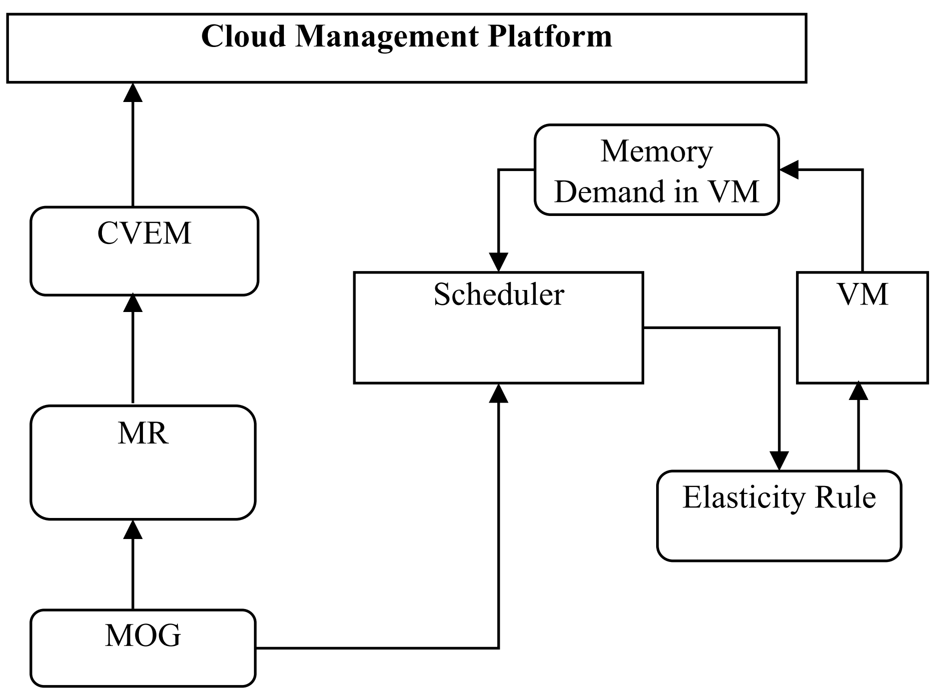


Fig. 2 Proposed system architecture

6 Results and Discussion

For simulation this research utilizes CloudSim 3.0 module is employed. CloudSim 3.0 offers simulation in cloud and power model simulation in predefined performance. The CloudSim power package with for simulation of VM aware allocation strategy for measuring power and fault with formulated algorithm program is evaluated with consideration of several test cases which involves linear power model and three serves ranges from S1–S5. Serve utilization is relies directly on power model. The developed algorithm is tested using the basic DVFS (dynamic voltage and frequency scaling) with respect to programming with effective power management scheme in servers. For proposed approach testing is run for requests count 600, 800, 1200 and 1400. The other configuration for server are presented as follows:

The evaluation of proposed approach is evaluated through consideration of VM which supports the several number of applications for constant VM applications. The cloud infrastructure classifies provision of infrastructure from the cloud and application for cloud relies on VM with introduction of vital edges for deploying multiple back-end process applications. In case of multi-event scenario one VM may accustomed for multiple scenario for several number of users which provides drastic variation in pattern of memory consumption which comprises running of single application within one VM. The cloud VAMP have tendency to adopt for effective managing of VM allotted memory or within collection of VM where memory is employed active counts.

In RAM the exact performance of VM (VMI) is presented for physical host factor with respect to on-premises cloud.

In VM1 cloud VAMP minimizes allotted RAM value since memory utilization of VM boosting is incredibly minimal (where no application are run in the allocated space).

The process is put within the instrument initially deployed where supported by Apache cloud application server which leads to rise in VM1 allocation memory requested by cloud VAMP.

Through constant image process second instrument is deployed which has tendency to increases memory by means of minimal sharing among two pages lies within two VM.

Due to reduced memory of VM1 enough free obtainable memory is lies within the physical machine to host another VM which can be VM2 defined as computer hardware standard Open Nebula which is running in the same physical host of the network.

A range of memory-intensive application for 3rd dead VM instrument introduces VM1 memory pressure. The application is based on the artificial benchmark application based on United States of America for regulation of memory allocation pattern in order to get the delineate performance through application of MFLOPS.

The memory of VM1 increases in cloud through VAMP which may often leads to memory overload with migration strategy through contingency arrange of VMs. The migration of VM involves transmission from physical host to aptitude of VM memory when it is not overload.

In physical host when enough memory is free then VM1 offers lot of memory space which considers memory size of VM not able to increase for first allocated memory quantity level of VM.

The sequence of events occurred in cloud server is distributed evenly with the cloud which is on-premises basis. The Fig. 3 illustrated the $64,000 VM allocation strategy in the VM with manual host factor in VM machine. In analysis, deployment of VM is performed via RAM size of 4 GB with time instant 1:00:00 which relies on event A. In second stage, Cloud VAMP identifies sufficient free memory with VM and allocate memory factor which slightly higher than that of 500 MB data for event B. AT 1:01:57 the VM tends to perform installation for packages installed under manual processing which demands for further memory and leads to drastic periodic increase in VM allotted memory of VM. In time 1:04:44 manual VM strategy is deployed for manual factor deployed in event C with increase in VM memory needs which often leads to increase in allotted memory. The VM allotted memory observed at time 1:06–1:07 are relies on VM steady state where manual square is measured is not in active state of the VM performed with the allocated memory for clear observation of memory allocation techniques. In the time period 1:07:36 causes deployment of VM in second devices with introduction of pressure memory with VM leads to allotted memory periodic increase with consistent need for memory where the domestic host the memory-intensive for the Java application server.

The above Fig. 4 demonstrated that proposed approach performs effectively based on the application which impacts effectively on over subscription and on-demand migration on VM for the specific application shows that up to 15 VMs in proposed techniques delivers the chief attributes effectively through ultimate thrashing and secondly it offers migration measures. However the transient reduction is incredibly insufficient for long term application which

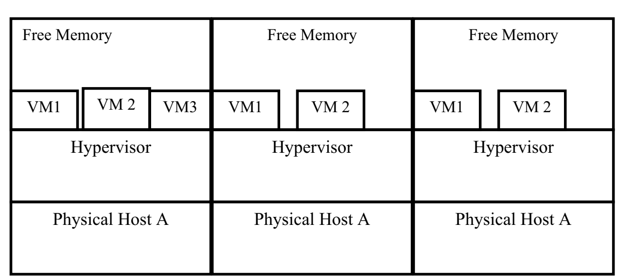


Fig. 3 VM memory allocation of IaaS cloud

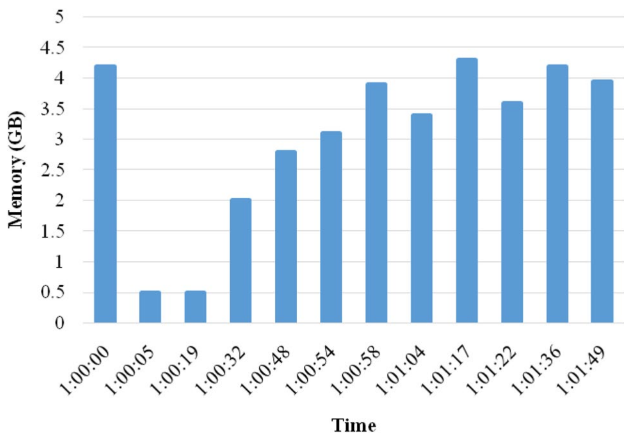


Fig. 4 Comparison of time vs memory

can be negligible. Additionally, Cloud VAMP fine tune the application in to consideration and forestall the expense of thrashing without wasting memory further. The MOP evaluation reduces the worth based on the consideration of level at infrastructure. As a result, memory allocation application offers low latency response which will not migrate on-demand with consideration of hosts alternatively which can be effective on its performance based of the expected service level of the consumer.

In case if Cloud service provider (Fig. 5 and Table 1) run application than its square measure significantly sensitive in its performance which need to support VM allocation strategy to run those application for set host and monitor Cloud VAMP. In this way, memory allocation in VM will not reduce the resource requested for application which cannot be migrated towards alternative hosts in to consideration. Further the Cloud VAMP is appropriate with extension of resource performance with minimal reclaim or effective utilization of unused resources. In other words, this can be denoted as target memory of the hypervisors which supports

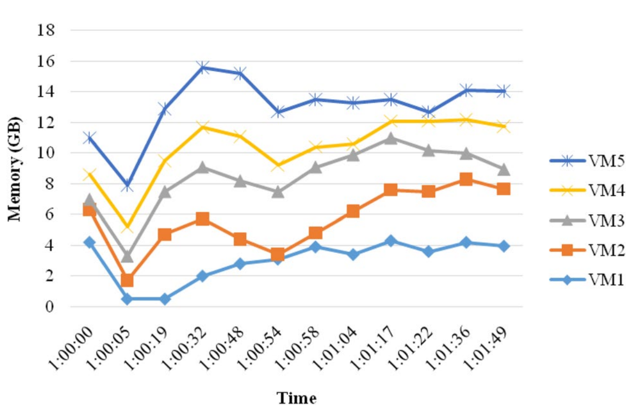


Fig. 5 Comparison of VMs time vs memory

Table 1 VMs memory allocation

Host	VM1	VM2	VM3	VM4	VM5
Physical memory	16	16	16	16	16
Allotted memory	6	4.5	8	5.3	12
Free space	10	11.5	8	10.7	4

dynamic management of resources based on the appliance performance. It deals with VM allotted memory which deals with utilization of VM memory utilization.

In this memory consumption pattern is effectively managed based on the application memory requirement with introduction of performance cost over the VM. At the end, it can be concluded that for on-demand cloud service VM performs effectively for based on application characteristics.

7 Conclusion

This paper proposed a Cloud VAMP architecture for customized memory allocation technique with consideration of over subscription for on-demand memory allocation in cloud. For effective memory allocation this paper adopts vertical elasticity in VM with consideration of downtime and on-demand migration with consideration of over subscription scenario in to consideration. Through the concept of memory leveraging and availability of on-demand migration capability in the KVM hypervisor the performance is evaluated. The evaluation exhibited that Cloud VAMP technique in Cloud management platform dynamically increase and decreases the VMs allocated memory which effectively fit with the VM running application. Further in this research generic architecture is presented which deploys various CMPs with implementation of open-source platform in Open Nebula which is used in this research. The analysis is performed through consideration of elastic virtual clusters in horizontal and vertical manner with consideration of different jobs under multi-agent scenario. The cloud VAMP reclaim ability of unused memory from those VM with enabling over subscription temporarily in CMPs with increase in consolidation ration of VM—per-host with reduction on application which runs. The on-demand migration provides advantage of service level restoration under memory overload scenario.

The proposed approach can be effectively employed in real-time applications such social media, YouTube for dynamic management of burst of data. In future, this research can be improved through incorporation of security scheme for improving performance of dynamic memory allocation.

References

Badger L, Grance T, Patt-Corner R, Voas J (2012) CC synopsis and recommendations. NIST Spec Publ 800:146

Baranwal G, Vidyarthi DP (2015) A fair multi-attribute combinatorial double auction model for RA in CC. J Syst Softw 108:60–76

Barba-Jimenez C, Ramirez-Velarde R, Tchernykh A, Rodríguez-Dagnino R, Nolazco-Flores J, Perez-Cazares R (2016) Cloud based video-on-demand service model ensuring quality of service and scalability. J Netw Comput Appl 70:102–113

Beaty K, Kochut A, Shaikh H (2009) Desktop to cloud transformation planning. In: Parallel and distributed processing, 2009. IPDPS 2009. IEEE International Symposium on IEEE, pp 1–8

Beaumont O, Eyraud-Dubois L, Lorenzo-del-Castillo JA (2016) Analyzing real cluster data for formulating allocation algorithms in cloud platforms. Parallel Comput 54:83–96

Beltrán M (2016) BECloud: a new approach to analyse elasticity enablers of cloud services. Future Gener Comput Syst 64:39–49

Calyam P, Rajagopalan S, Selvadhurai A, Mohan S, Venkataraman A, Berryman A, Ramnath R (2013) Leveraging OpenFlow for resource placement of virtual desktop cloud applications. In: Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on IEEE, pp 311–319

Calyam P, Rajagopalan S, Seetharam S, Selvadhurai A, Salah K, Ramnath R (2014) VDC-analyst: design and verification of virtual desktop cloud RAs. Comput Netw 68:110–122

Coutinho RDC, Drummond LM, Frota Y, de Oliveira D (2015) Optimizing virtual machine allocation for parallel scientific jobflows in federated clouds. Future Gener Comput Syst 46:51–68

Deboosere L, Vankeirsbilck B, Simoens P, De Turck F, Dhoedt B, Demeester P (2012) Cloud-based desktop services for thin clients. IEEE Internet Comput 16(6):60–67

Ficco M, Esposito C, Chang H, Choo KKR (2016) Live migration in emerging cloud paradigms. IEEE CC 3(2):12–19

Ficco M, Esposito C, Palmieri F, Castiglione A (2018) A coral-reefs and game theory-based approach for optimizing elastic cloud RA. Future Gener Comput Syst 78:343–352

Fiedler M, Hossfeld T, Tran-Gia P (2010) A generic quantitative relationship between quality of experience and quality of service. IEEE Netw 24(2):36–41

Galante G, Bona LCED (2012) A survey on CC elasticity. In: Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and CC. IEEE Computer Society, pp 263–270

Lin W, Peng B, Liang C, Liu B (2013) Novel RA model and algorithms for CC. In: Emerging intelligent data and web technologies (EIDWT), 2013 fourth international conference on IEEE, pp 77–82

Maguluri ST, Srikant R, Ying L (2014) Heavy traffic optimal RA algorithms for CC clusters. Perform Eval 81:20–39

Mell P, Grance T (2011) The NIST definition of Cloud Computing. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf. Accessed 1 Sept 2011

Moltó G, Caballer M, de Alfonso C (2016) Automatic memory-based vertical elasticity and oversubscription on cloud platforms. Future Gener Comput Syst 56:1–10

Naskos A, Stachtiari E, Gounaris A, Katsaros P, Tsoumakos D, Konstantinou I, Sioutas S (2014) Cloud elasticity using probabilistic model checking. arXiv preprint arXiv:1405.4699.

Praveenchandar J, Tamilarasi A (2020) Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-01794-6

Sivaram M, Mohammed AS, Yuvaraj D, Porkodi V, Manikandan V, Yuvaraj N (2019) Advanced expert system using particle swarm optimization based adaptive network based fuzzy inference system to diagnose the physical constitution of human body. In: International conference on emerging technologies in computer engineering. Springer, Singapore, pp 349–362

Yan L (2011) Development and application of desktop virtualization technology. In: Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on IEEE, pp 326–329

Yuvaraj N, Suresh Ghana Dhas C (2018) High-performance link-based cluster ensemble approach for categorical data clustering. J Supercomput. https://doi.org/10.1007/s11227-018-2526-z

Yuvaraj N, Vivekanandan P (2013) An efficient SVM based tumor classification with symmetry non-negative matrix factorization using gene expression data. In: 2013 International conference on information communication and embedded systems (Icices). IEEE, pp 761–768

Yuvaraj N, Raja R, Dhas C (2018) Analysis on improving the response time with PIDSARSA-RAL in ClowdFlows mining platform. EAI Endorsed Trans Energy Web 5(20):1–4. https://doi.org/10.4108/eai.12-9-2018.155557

Yuvaraj N, Kousik NV, Jayasri S, Daniel A, Rajakumar P (2019) A survey on various load balancing algorithm to improve the task scheduling in cloud computing environment. J Adv Res Dyn Control Syst 11(08):2397–2406