

Multi-dimensional Affinity Aware VM Placement Algorithm in Cloud Computing

Nilesh Pachorkar¹, Rajesh Ingle²

Abstract

One of the challenging problems in cloud computing is the efficient placement of virtual machines having varying resource requirements over the physical machines of fixed capacity. The placement of virtual machines can be done using different approaches. One of the approaches is reducing virtual machine placement problem to multidimensional bin packing problem which is a NP-Hard problem. As the VM placement is a multidimensional problem, it is necessary to consider multiple dimensions such as CPU usage, memory, network bandwidth, etc. together while allocating virtual machines to physical machines. By considering the memory and network affinity among virtual machines during the VM placement, we can significantly reduce the memory usage and network overhead. Existing algorithms do not consider these affinities together while performing the VMs' placement. In this paper we proposed a multidimensional affinity aware VM placement algorithm which can optimize the performance as compare to the existing algorithms.

Keywords

Cloud computing, virtualization, page sharing, memory affinity, network affinity, VM placement.

1. Introduction

Virtualization is the key feature of cloud computing that enables multiple virtual machines to share the resources on a single physical machine which in turn improves resource utilization of data center. Virtualization enables the live migration [9] of virtual machines (i.e. relocating a VM from one host to another without taking it down) which helps in maintaining the promised SLA to the cloud consumer and also for balancing load across physical servers in the data centers.

Nilesh Pachorkar, Department of Computer Engineering, P.I.C.T. College, Pune, India.

Rajesh Ingle, Department of Computer Engineering, P.I.C.T. College, Pune, India.

Live migration is used in server consolidation i.e. consolidating VMs on fewer physical servers. Virtual machine monitors (also called hypervisors) such as VMware [10], Xen [11] are widely used for management of virtual machines (VMs) and facilitates the creation, destruction and migration of virtual machines. Due to the ability of hosting multiple virtual machines on the same physical machine, new challenges have been introduced such as mapping of virtual machines (VMs) to physical machines (PMs), load balancing amongst all physical machines in the data center. The placement of VMs to PMs is subject to many constraints such as the static and dynamic resource requirements of VMs, security requirements, availability requirements [1]. Most of these constraints are specified or defined in the Service Level Agreement (SLA) associated with the each customer's VM. Existing VM placement algorithms uses different approaches such as Constraint programming, Bin packing, stochastic integer programming, Genetic algorithm, etc. to solve the problem of VM placement in cloud computing. In this paper we considered the Bin packing approach in the proposed multidimensional affinity aware VM placement algorithm to solve the problem of VM placement.

There are different approaches for placement of VMs on the PMs in a cloud infrastructure such as Constraint Programming, bin packing, Genetic algorithm, stochastic integer programming (SIP), etc.

Constraint programming approach can be used when inputs such as resource requirements of VMs, SLA requirements are available [7]. In this approach, as the number of constraints increases, the time taken to generate optimal solution also increases.

Bin packing approach can be used for dynamic VM placement where the demand is highly variable [8]. It is heuristic based approach. Therefore, it may not generate an optimal solution but it will always generate a good solution in considerable amount of time. Multi-dimensional bin packing approach can also be used for solving VM placement problem.

The stochastic integer programming approach is used for modelling optimization problems involving

uncertainty [13]. In cloud computing, there is a uncertainty in resource demand and the cost of resources is also uncertain. This approach is used in [3] to optimize the resource provisioning which helps to decrease the resource provisioning cost for the customer.

The genetic algorithm is a search heuristic that can be used to find approximate solutions to the optimization problems. Genetic algorithm uses techniques which are inspired by natural evolution. The genetic algorithms do not scale well for larger systems [14]. The Genetic algorithm approach can be applied to the VM placement problem. Nakada et al. [14] uses genetic algorithm to solve the problem of virtual machine packing problem.

Affinity of VMs can be defined as the dependency between VMs [4]. There are different types of affinity in cloud computing such as communication affinity or network affinity, memory affinity, data affinity, and user defined affinity [4]. The main contributions of this paper are listed below:

- The survey of existing VM placement algorithms including non-affinity-aware VM placement algorithms as well as affinity-aware VM placement algorithms.
- The multi-dimensional affinity-aware VM placement algorithm is proposed which find out the trade-off between memory and network affinity between VMs and perform the VM placement by considering both memory and network affinity between VMs.
- The affinity-aware VM placement system is proposed which consist of memory tracer, network tracer, and control plane. This system can improve the application's performance by decreasing the network overhead and memory usage.

The rest of the paper is organized as follows. The survey of existing VM placement algorithms is presented in the section 2. The analysis of existing affinity aware VM placement algorithms is given in section 3. Then the VM placement problem is described in the section 4. The proposed system is explained in detail in section 5 and the section 6 concludes this paper.

2. Literature Review

Recently there is a lots of research has been done in the field of VM placement problem and it is still going on. The VM placement algorithms can be divided into two categories: One is affinity aware VM placement algorithms and Second is non-affinity aware VM placement algorithms i.e. VM placement algorithms that do not consider the affinity between VMs while placing VMs onto the PMs.

Some of the VM placement algorithms [1], [2] considered the continuous stream of VM deployment requests that arrives in the cloud environment. Wubin Li et al. [2] assumes the exact duration for each deploy set. But, the behaviour of deploy requests and their duration is unpredictable. Calcavecchia et al. [1] presented a VM placement technique called Backward Speculative Placement (BSP) that can optimize the VM placement under a continuous stream of deploy requests. This system considers the dynamic nature of VM deployment requests as compare to the [2] which consider the exact duration for each deploy set. Authors used the BSP technique in two scenarios. First, it is used to handle the continuous stream of VM deployment requests. Second, it is used in a periodic optimization where current VM placement is re-optimized. BSP technique used the multi-dimensional approach for the VM placement. BSP technique does not considered the network relationship between VMs belonging to the same set.

Chaisiri et al. [3] proposed an Optimal Virtual Machine Placement (OVMP) algorithm for the VM placement across multiple cloud providers. The authors considered two payment plans i.e. reservation plan and on-demand plan that cloud providers can offer to cloud consumers for resource provisioning. OVMP algorithm uses Stochastic Integer Programming (SIP) approach to minimize the resource provisioning cost in a cloud computing environment. The OVMP algorithm takes into account the probability distribution of future resource demands and resource pricing.

Very few existing VM placement algorithms considered the affinity between virtual machines. The VM placement algorithms proposed in [4], [5] and [6] considered the affinity between virtual machines while hosting virtual machines (VMs) onto the physical machines (PMs). Jianhai Chen et al. [4] proposed a VM placement algorithm called affinity aware grouping for allocation of VMs (AAGA). This algorithm identifies the network affinity between groups of VMs and tries to colocate those VMs onto

the same physical machine. This improves the application performance and also reduces the network overhead. The authors also defined some rules for performing affinity-aware grouping of VMs. A decentralized affinity-aware migration technique is proposed in [5] that identifies the network affinity between pairs of VMs. This affinity is then used for VM placement which is done with the help of distributed bartering algorithm. In [6], a system called Memory Buddies is developed that can identify the VMs with higher page sharing potential and colocate them onto the same physical host. The authors also developed a memory fingerprinting technique to identify VMs with high page sharing potential. The system uses VM colocation algorithm that identify VMs with high page sharing potential and colocate them onto the same host.

3. Analysis of VM Placement Algorithms

In this section, we have provided some analysis of the existing affinity aware VM placement algorithms [5, 6]. In [5], J. Sonnek et al. considered the network affinity for the placement and migration of VMs. The algorithm proposed by the J. Sonnek et al. shows up to 42% improvement in the application's runtime as compare to the no-migration technique and achieves up to 85% reduction in the network communication cost. While in [6], T. Wood et al. considered the memory affinity between VMs for the placement of VMs. The authors have developed a memory fingerprinting technique which helps to find out the memory affinity between different VMs. The system proposed in [6], can increase the data center capacity by 17% when VMs with high page sharing potential are consolidated.

4. Problem Statement

In VM placement problem, there are 'n' virtual machines and 'm' physical machines given. The resource requirement of each VM across each dimension such as CPU, memory, network bandwidth, etc. is given. Each physical machine has certain limited resources across each of these dimensions. Here, the objective is to place the VMs on the minimum number of PMs such that the total demand of all VMs hosted on a single PM should not

exceed its capacity across any dimension. This VM placement problem can be mapped to a multi-dimensional bin packing problem, also known as vector bin packing problem. After initial placement of VMs on different PMs, the proposed system will find out the memory and network affinity between different VMs which are hosted on different PMs. After determining the affinities between VM pairs, the proposed system will try to place these VMs on the same PM. So, the problem statement is to develop a VM placement algorithm that will consider the memory and network affinity between VMs while placing the VMs on the physical machines (PMs).

5. Proposed System

Consider a data center with multiple physical machines (PMs). Each PM runs a hypervisor and one or more VMs. Applications are running on one or multiple VMs. Initially the VMs are placed using some variant of First-FIT-Decreasing (FFD) heuristic such as FFDProd or FFDAvgsum [12] which are used for multi-dimensional bin packing problem. The bin packing problem is a NP-hard problem [8]. But, it can be solved by using many heuristics such as First Fit, Best Fit, Worst Fit, etc. The multidimensional bin packing problem is also a NP-hard problem [12] which can be solved by using heuristics such as FFD (First Fit Decreasing) and its variants like FFProd, FFDAvgsum, etc. As the applications are running on multiple VMs, the different components of an application on those VMs are communicating with each other. Thus, there is communication dependency (i.e. network affinity) between VMs. This communication dependency can be obtained from the network traffic between the VMs while the application is executing on them [4]. This communication dependency between VMs is also called as network affinity. [4, 5] shows that if such VMs having network affinity between them are colocated on same PM then it can reduce the communication overhead of physical network and hence improves the performance of application. Content based page sharing mechanism (CBPS) is employed in many hypervisors, such as VMware ESX, Xen, etc. which identifies the VMs' memory pages with identical content and then consolidates these memory pages into a single memory page.

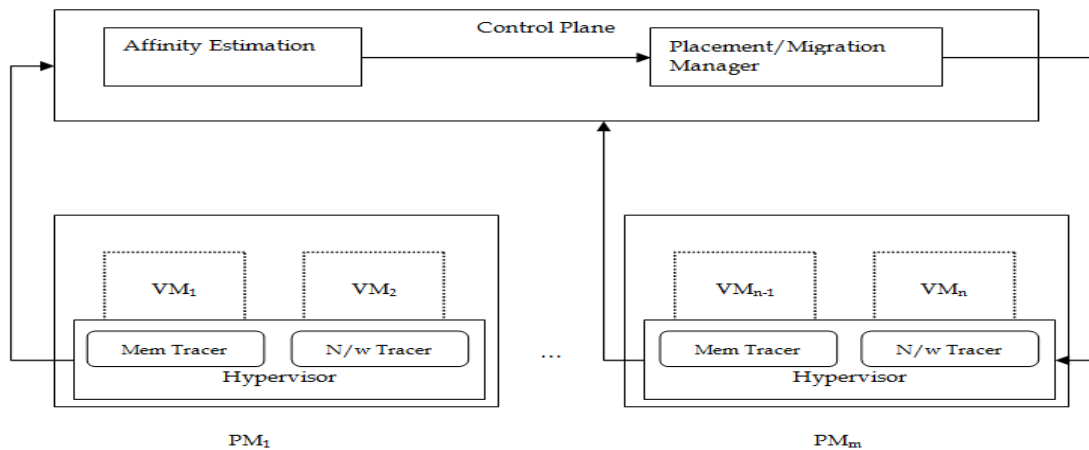


Fig.1 Proposed System Architecture

But, this mechanism is applicable only to the VMs which are hosted on the same PM. In cloud environment, there can be VMs hosted on different PMs having identical pages and it is necessary to host these VMs on the same PM to take the full benefit of CBPS [6]. This limitation of CBPS is overcome by the memory buddies system which is proposed in the [6]. The affinity aware algorithms proposed in [4], [5] and [6] consider only the single affinity i.e. network affinity or memory affinity. Neither algorithm considers both these affinities together. As the VM placement is the multi-dimensional problem, it is necessary to consider multiple dimensions such as CPU usage, memory and network bandwidth. In the proposed algorithm, we have considered the memory and network affinity together while performing the VMs placement on the physical hosts. The proposed system architecture is shown in fig. 1. There are n virtual machines which are to be placed on the m physical machines. The memory tracer (mem tracer) and network tracer (n/w tracer) are the tools that reside at the each hypervisor level.

The network tracer tool will analyse the traffic between VMs which are hosted on different physical machines by using traffic analyser tool like TCPdump. The network tracer tool will analyse the traffic for all VMs that are hosted on the same PM. Then it will send the statistics of each VM to the control plane. It is assumed that all storage of the data center is on the network file system. The memory tracer tool is similar to the Memory Buddies [6] tool which finds the page sharing potential between different VMs. The memory tracer tool residing at the hypervisor will create the memory fingerprint for all memory pages of each VM on the same host. With

the help of these memory fingerprints, the affinity estimation module can compute the number of pages with identical content between two VMs [6]. From this the memory affinity between different VM pairs can be found out. The control plane is running on one of the VM of the physical machine. The affinity estimation module will receive the memory usage and network usage statistics of each VM from memory tracer and network tracer tools located at each hypervisor. Then affinity estimation module will identify the communication dependency between VM pairs. This will give us the network affinity between VM pairs.

The memory and network affinity between VM pairs found by affinity estimation module are given as input to the proposed VM placement algorithm which runs in the VM placement/migration manager. The placement manager uses the heuristics based on these memory and network affinity between VMs. This heuristic is used with FFD (First Fit Decreasing) algorithm to place the VMs on the available PMs. In this way, the VMs with memory and network affinity between them will get placed on the same physical machine. This may reduce the communication overhead in the data center and also the memory usage in the data center. This will also help to improve the application's performance.

6. Conclusion

After studying the existing affinity-aware VM placement algorithms, we found that there is a need of multi-dimensional affinity-aware VM placement algorithm which will consider the multiple affinities such as memory and network affinity between VMs

while placing the VMs on the PMs. The existing affinity aware algorithms have not considered the memory and network affinity together which is required to be considered as the VM placement is the multi-dimensional problem. If we consider only one dimension while placing the VMs then it may cause overhead along other dimension. So, in the proposed algorithm we have considered both the memory and network affinity together while placing the VMs. The proposed multi-dimensional affinity aware VM placement algorithm can optimize the resource provisioning mechanism while reducing the communication overhead and memory usage in the data center. This will help to improve the performance of the applications which are hosted on multiple virtual machines in the data center.

References

- [1] N. Calcavecchia, O. Biran, E. Hadad, E. and Y. Moatti, "VM Placement Strategies for Cloud Scenarios," IEEE 5th International Conference on Cloud Computing (CLOUD), vol., no, pp.852,859, 24-29 June 2012.
- [2] W. Li, J. Tordsson, and E. Elmroth, "Virtual Machine Placement for Predictable and Time-Constrained Peak Loads", Proceedings of the 8th international conference on Economics of grids, clouds, systems, and services (GECON'11) Vol. 7150, Springer-Verlag, pp. 120-134, 2011.
- [3] S. Chaisiri, B.S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," Services Computing Conference", APSCC 2009 IEEE Asia-Pacific , vol., no., pp.103,110, 7-11 Dec. 2009.
- [4] J. Chen, K. Chiew, Deshi Ye, L. Zhu, and W. Chen, "AAGA: Affinity-Aware Grouping for Allocation of Virtual Machines," IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), vol., no., pp.235,242, 25-28 March 2013.
- [5] J. Sonnek, J. Greensky, R. Reutiman, and A. Chandra, "Starling: Minimizing Communication Overhead in Virtualized Computing Platforms Using Decentralized Affinity-Aware Migration," 39th International Conference on Parallel Processing (ICPP) , vol., no., pp.228,237, 13-16 Sept. 2010.
- [6] T. Wood, G. Tarasuk-Levin, P. Shenoy, P. Desnoyers, E. Cecchet, and M.D. Corner, "Memory buddies: exploiting page sharing for smart colocation in virtualized data centers" Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments March 11-13, 2009, Washington, DC, USA.
- [7] Hien Nguyen Van, Tran, F.D., Menaud, J.-M., "Autonomic virtual resource management for service hosting platforms," ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD '09., vol., no., pp.1,8, 23-23 May 2009.
- [8] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," 10th IFIP/IEEE International Symposium on Integrated Network Management, vol., no., pp.119,128, May 2007.
- [9] C. Clark, K. Fraser, A. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines", Proceedings of the Symposium on Networked Systems Design and Implementation, pp.273-286, May 02-04, 2005.
- [10] VMware hypervisor, <http://www.vmware.com/products/esxi-and-esx/overview> (as of 12 December 2013).
- [11] Xen hypervisor, <http://www.xenproject.org> (as of 12 December 2013).
- [12] R. Panigrahy, K. Talwar, L. Uyeda, & U. Wieder, "Heuristics for vector bin packing" (2011), <http://research.microsoft.com/apps/pubs/default.aspx?id=147927>, unpublished.
- [13] Stochastic Programming, <http://stoprog.org/index.html?spintroduction.html> (as of 12 December 2013).
- [14] H. Nakada, T. Hirofuchi, H. Ogawa, and S. Itoh, "Toward virtual machine packing optimization based on genetic algorithm", Proceedings of International Symposium on Distributed Computing and Artificial Intelligence, vol. 5518, pp. 651-654, Springer, June 2009.



cloud computing.

Nilesh Pachorkar received the B.E. degree in computer engineering from Pune University, Pune, India in 2012. He is currently pursuing M.E. computer as PG Scholar in the department of computer engineering, PICT, Pune University, Pune, India. His research interest includes distributed systems and



Rajesh Ingle is a professor of computer engineering, PICT, Pune. He received Ph.D. CSE from IIT Bombay, B.E. Computer from PICT, Pune University, and M.E. Computer from COEP, M.S. Software Systems from BITS, Pilani. His research interests include distributed system security, grid

middleware, cloud computing, multimedia networks and spontaneously networked environments. He has more than 20 research publications in conferences and Journals.