International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Unified Provisioning Framework for Dynamic Virtual Machine Placement Optimization in Cloud Data Center

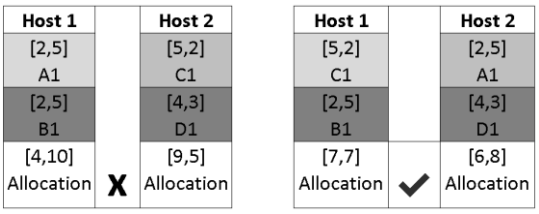
Darshan Shah, M. Vinayakmurthi, Anand Kumar

Abstract: ***Current techniques for Virtual machine placement in cloud data center is avoiding multiple resources due to its complexity and hardness of the problem. Due to this each Virtual Machine Placement increased overall frequency of server consolidation and migration. In this paper, we have overcome these limitations by providing local search-based unified approximation framework which utilized multiple resources of server and reduced the frequency of server consolidation and migration. The framework is evaluated on Azure cloud data center benchmark data sets and it has surpassed existing methods with improvement by 32% in overall virtual machine placement.***

Index Terms**: *Virtual Machine Placement, Cloud Data Center, Multi-dimensional resource optimization.***

I. INTRODUCTION

The cloud computing is one of the most cost-effective models to process data intensive jobs. Virtual machine hosted over cloud data center helps different application and services to execute on demand basis to process such data for analytical and decision making. Each individual virtual machine consumes different types of resources. For example, virtual machine hosts with machine learning consumes more CPU than normal VM. Due to such diversity and demand, it is required to build a framework that consider different machines resources which utilize it with maximum capacity. It is highly desirable and profitable in terms of logistics perspective that all the resources like memory, CPU, network bandwidth can become fully used on a machine and prevent any imbalance resource allocation. Imbalance allocation increases server maintenance activities like migration and consolidation. Current placement techniques or frameworks using heuristic based techniques with no performance guarantee. In later section such widely used techniques are highlighted with their limitations. First, most of current techniques during placement going with slot-based approach in which resources are divided into different slots corresponding to some amount of memory and cores then later use heuristics or greedy algorithms to place them over different hosts. Second, they also ignored different resources like disk storage and network bandwidth requirements during placement and due to this, VMs are either over provisioned or under provisioned. Below two figures describe the problem briefly in which first case it has placed with imbalance resource utilization but second has considered all the resources of the machine hence it considered a right placement for the machine. There are three main service components in Cloud Computing. Platform as a Service (PaaS), Software as a Service and Infrastructure as a Service (IaaS). Out of them IaaS is the most important and widely used component by users. This also equally important for cloud provider to manage it well for cost management. In IaaS, cloud provider manages infrastructure resources such as servers, storage, networking hardware. The users used these cloud services to get the benefits of Pay as You Go model in which they have to pay only what they used and consumed. For example, if any server is not required then it can be shut down and user doesn't have to pay any cost for it.



Imbalance Placement **Balance Placement**

Figure 1: Virtual Machine Placement

If Virtual machine placement is considered as generalized assignment problem, then it is nothing but a bin packing problem. So, in literature of Virtual Machine problem is often modeled as bin packing problem in which virtual machines are considered as items and machines as bins and objective is to minimize bins consumption by covering maximum items [4],[5],[6],[7],[8]. When it is model as Vector bin packing which is nothing but specialization of bin packing problem, all vectors are as resources and bins as machines and objective is to place all vectors as near as to its diagonal [9], [10]. However, such modeling as Vector bin packing (VBP) is not give optimum result due to hardness hence such reduction is not directly applying to the problem addressed in this paper. The bigger limitation of VBP is that all the items are assumed to be known in advance which is not possible in real scenario because in VMP requests are also online as well as offline mode.

Revised Manuscript Received on 30 July 2019.
* Correspondence Author
Darshan Shah*, Computer Science Department, Reva University, Bangalore, India.
Dr. M. Vinayakmurthi, Computer Science Department, Reva University, Bangalore, India.
Dr. Anand Kumar, M.S. Engineering College, Bangalore, India.

This paper presents Unified Provisioning Framework (UPF) for Virtual machine placement in which dynamic environment is considered which helps to handle any type of request and no need to change the algorithm or configuration of the environment. The proposed framework is using graph coloring concepts which adaptively place VMs in both offline and online mode efficiently. While we have focused on multi resource balance schedule, optimum resource consumption is the important parameter to consider for efficient virtual machine placement. The technique available in [1] doesn't provide the performance guarantee hence it is not applicable in the real time environment. UPF helps to provider better resource management and leads towards optimum performance and its impact on allocation is quite noticeable during empirical evaluation. Our work makes following advancement in Virtual machine placement problem,

- Unified approach in which multiple resources are allocated efficiently in both offline and online mode comparing to existing framework in which separate solution is required for each mode.
- Proposed framework is based on mathematical model which provides performance guarantee.
- It is considering multiple resources during placement and adaptive to dynamic environment of cloud data center which helps to reduce frequency of server consolidation and migration.
- The framework has been evaluated in real time bench mark data set of azure cloud data center and compare to state-of-the-art placement techniques it improves resource balance by 32%. It also offers a synthesis between balance allocation and performance.

II. LIMITATION OF EXISTING TECHNIQUES

Matrices based technique is most common technique for VM placement. In paper [4] a weighted sum of resources is used and paper [9] used mathematical function of resources. It is quite important to accurately estimates the current utilization and total capacity available in dynamic cloud data center environment. Generally, those details are collected by logs and monitoring tools. It is also required overhead estimation model for the application to find accurate resources needed for the virtual machines. In [1] author used micro benchmark to identify overhead over server after placement. In the paper [6] discussed different performance prediction models of applications running and used by the different cloud users. Methodology used in [9] employs forecasting by using time and cost parameters to fulfil changes in the resource requirement. Considering problem addressed in this paper, we only discussed related techniques in detail as below.

A. Sandpiper

Sandpiper used volume-based approach to migrate and place VMs. [3] It detects when VM is under provisioned and then place or migrate VM. If we consider PM with 3 resources CPU (c), network bandwidth (nb) and memory(m) then total capacity of PM to corresponding PM is calculated by Sandpiper as below,

$$\frac{1}{1-c} * \frac{1}{1-nb} * \frac{1}{1-m}$$

It then calculates exploitable volume based on calculation of (1- c) * (1- nb) * (1- m) which gives total capacity remained in the server. When any hotspot (under provisioned) PM is detected then it ordered PMs in the decreasing order of their volume. Then VM placement algorithm place VM by identifying the highest Volume Size ratio. The overall approach is migrating VM as Worst Fit Algorithm with greedy approach. This algorithm seems to be good for placement but having limitation in resource utilization. Let us understand it with example. Without losing the generality, reduce problem to 2D resource utilization problem. Using above equation let us take the values for CPU and mem as below,

The requirement of VM is 0.13 mem and 0.2 CPU. Consider above availability as per sandpiper algorithm a VM will be placed in the PM2 is selected. The selection is not optimum because PM1 has more CPU intensive capacity than PM2.

Below figure give more visual insight on placement performed by sandpiper,

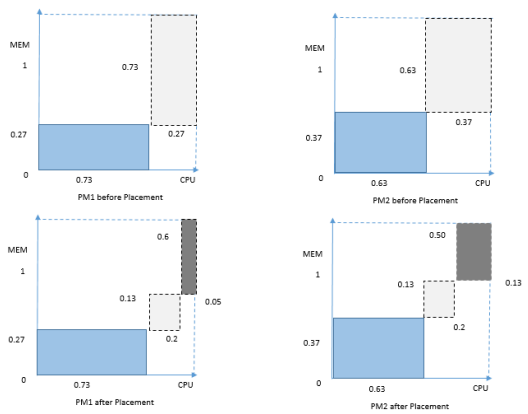


Figure 2: Sandpiper – Virtual Machine Placement

B. Vector-dot

Vector dot is another novel vector [2] based scheme which based on the idea of identifying dot product. To derive dot product, it has used Resource Utilization Vector (RUV) of PM and Resource Requirement Vector (RRV) of VM and out of it chooses lowest dot product of all available list. The approach identifies complementary VM from the available list. For example, A new VM request has high CPU requirement but low memory utilization so in that case the algorithm check best PM which has the low CPU resource consumption but high memory consumption. This approach seems to be good but has a drawback. Let us understand by below example, The RUV of PM1 is 0.5 and 0.4 and PM2 0.3 and 0.4 respectively for CPU and MEM. The RRV of VM is 0.1 and 0.2 so dot product between RRV (VM) and RUV(PM1) is 0.13

whereas for PM2 it has 0.13 as dot product. So as per the vectorDot PM2 is the final choice and the reason is the VM has complementary resource requirement for PM1. For example, VM needs more memory than CPU and PM2 which ended up by providing more memory and less CPU.

This clearly shows that it is a misfit because PM 2 used up more memory and less CPU so VM does not have the complementary resource requirement with respect to PM2. This factor is the downside of the vectorDot approach and suitable for optimum VM placement scheme.

C. Novel Vector based Heuristic

In the paper [1] authors have derived new approach of VM placement by deriving all resources as vectors and represented resource as cube. This approach adopts large variable neighborhood search in which it has various neighborhood and apply search for VM placement. As shown in the below diagram, it has created a resource cube for 3 different types if resource and the same way if it can create four different types of resources as octagon. For example, a VM with 3 resources requirements will creates hexagon with (0,0,0) and (1,1,1) as Normal resource hexagon.

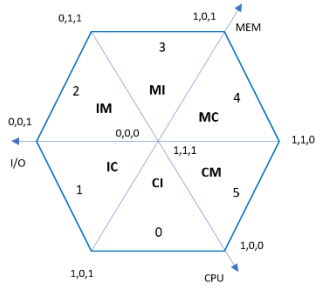


Figure 3: Novel Vector Based approach for Virtual Machine Placement [1]

The resource is represented as resource hexagon with equal parts as six triangles. It uses three resource matrices to identify resource utilization as Low, Medium and High. A Resource Utilization Vector (RUV) of a PM and Resource Requirement Vector (RRV) of VM in the Planar Resource Hexagon (PRH) gives information about resources utilization. Above techniques needs more PM to visit and adopts best fit approach based on vectors available. This technique does not consider fact that some suitable hosts are in a severely underutilized state and would be better off being powered down and having their virtual machines consolidated on another host. The first host is severely off balance with 90% of its CPU complement in use and 5% of its RAM complement in use. The second host is slightly off balance with 10% of its CPU complement in use and 5% of its RAM complement in use. A virtual machine that would use 10% of a host's CPU complement and 95% of a host's RAM complement now needs to be placed. The virtual machine should be placed on the first host to use all the host's available resources. If the virtual machine were to be placed on the second host, only the second host's RAM complement would be exhausted. Additionally, 80% of the second host's CPU complement would be rendered unusable. (A virtual machine cannot utilize only CPU resources). Consequently, preference as a target should be given to off balance hosts. The same this technique is not providing performance guarantee and hence not suitable for the ideal VM placement scheme.

III. UNIFIED PROVISIONING FRAMEWORK

In this section, we describe Unified Resource Provisioning framework, begin with reduction to graph coloring to build a model and then shows main algorithm used for placement and how it helps to achieve better model for VM placement with balance resource optimization. Finally, combine all we show overall architecture and implementation of our Unified Provisioning framework for Virtual machine placement with balanced utilization of multiple resources.

A. Reduction to graph coloring problem

The framework used graph coloring connects to build its mathematical model. The lower and upper bound are derived using reduction to graph coloring problem. In this section we show lower and upper bound achieved in the framework.

To achieved bound we have extended the technique available in the [8] and reduced the problem instances to graph coloring technique and then presents the main algorithm used in the framework.

Our results are as below,

3.1.1. Algorithm build a graph for each VM allocated to machine.

3.1.2. Each node will be mark as colors in the graph network. This will represent the virtual allocation of the VM.

3.1.3 Algorithm will identify right node to mark with unique adversary color such way that it will not rom the clique. If clique is identified then it avoids the node and mark other suitable node with color.

3.1.4 Now this virtual mapping is used to allocate VM to machine.

3.1.5 The whole process completes in the time

log d/ log log d in which d represents the size of the total VM allocated to machine.

B. Reduction rules used in graph coloring

To get performance bound, we introduced a reduction technique in the framework. Generally, idea is to build a virtual model of server based on divide into different slots and then defined number of colors require to build a graph such that overall number of monochromatic cliques will reduce. Our reduction rules grantees that, any optimal solution for the placement can be extended into an optimal solution for the original graph by coloring all the required vertices. For each vertex (VM request), it treats as a game with below three steps,

1) Defined number of colors and allocate each colors to slot.
2) Encode each vertex with unique colors available in the slot.
3) Use algorithm to place the vertex in the graph such a way that minimize the monochromatic clique.
4) Use adversary strategies to color the graph.

C. Main algorithm used for VM placement in framework:

In the unified provisioning framework, use above graph coloring reduction and have four phases to identify correct placement for virtual machine.

Step 1. Initialization components and services.

Step 2. Identify right node without building the clique in the graph.

Step 3. Use this virtual mapping for allocation.

Step 4. Provision VM with all required configuration.

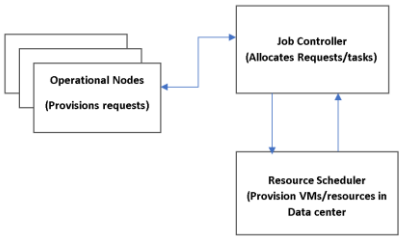


Figure 4: Unified Provision Framework

IV. EXPERIMENTAL EVALUATION AND RESULT

Unified Provisioning Framework is evaluated using our prototype implementation on a 25 data sets collected from private data center running azure cloud platform. To understand and evaluate performance a large trace is collected from production environment.

A. Infrastructure

The data centre configured with Azure pack in the private cloud consists of different windows servers running over machine with 64 cores, 32 GB of memory, 1TB hard drives and 1Gbps Network Interface card. We also report experiments on a small size server in which each machine has 6 cores, 16GB of memory, 3 1TB disks and a 1Gbps network interface card.

B. Workload:

To test our framework, production environment different work load types are taken from and instances are generated in the environment [12]. The multiple resources like CPU, memory and network bandwidth are taking into consideration and randomly picked up data set from available choices. Balanced CPU-to-memory ratio which is generally used for general purpose. Then workload is generated for types which are Compute optimized which has high CPU to memory configuration is available. The same way Memory optimized, and Storage optimized virtual machines workloads are also generated. We have avoided very large size configuration which is used for high performance computing and GPU because of environment set up restrictions.

C. Highlights:

1. Unified Provisioning Framework improves resource optimization and balance allocation by about 28% in our deployment and by up to 32% when replaying using production traces in simulation. This result is consistent during the testing and results are promising from avoiding resource underutilization and overallocation.
2. Framework incurred efficiency but also incurred slight loss compare in which it has targeted to reduce the overall number of sizes. But this loss is not more than 3.56% compared to our estimated loss before evaluating framework which was 10%.

D. Experimental Results

Unified Provisioning framework (UPF) is compared with First Fit Decreasing (FFD), Sandpiper, Vector Dot and Novel vector-based approach (NVM).

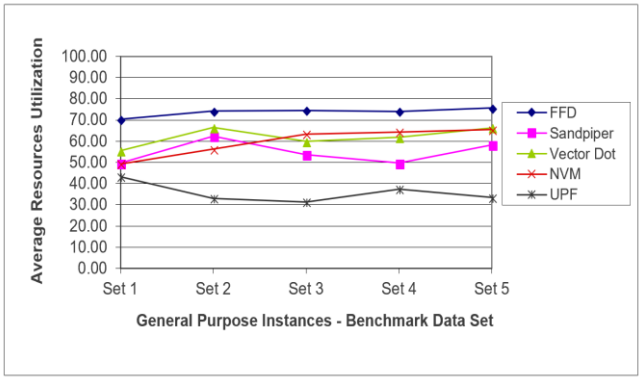


Figure 5: Test result of general-purpose instances

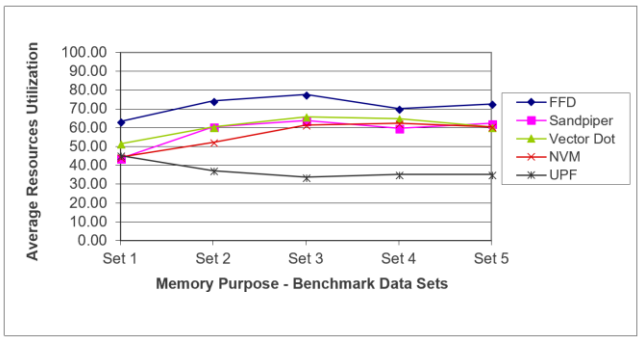


Figure 6: Test result of Memory purpose instances

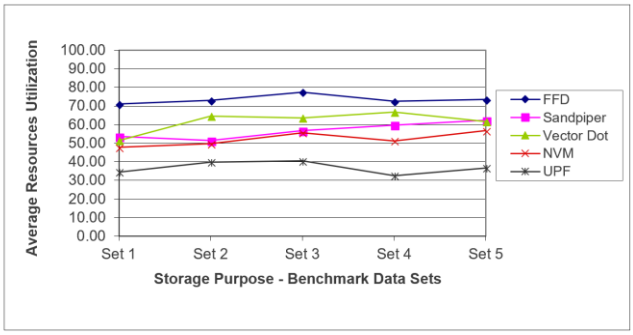


Figure 7: Test result with storage instances

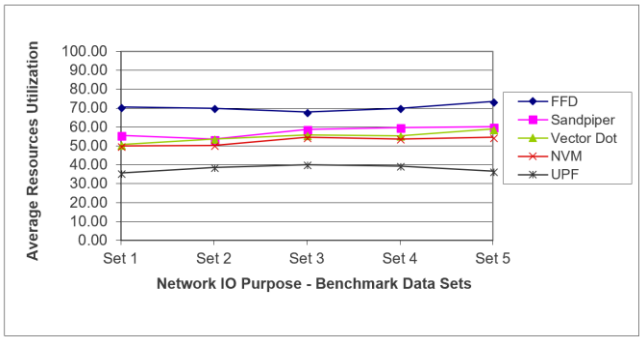


Figure 8: Test result with network I/O instances

It is quite difficult to maintain scalability during large instances testing. To estimate overheads, experiments ran with different types of virtual machines instances and different types of machine configurations. An additional network overhead was observed due to expanding size of instances but that was negligible. Further, the memory usage at the different servers is observed in profiler was very slightly;

the timeline graphs (not shown) are dominated by garbage collection of .NET framework.

V. CONCLUTION

In this paper, we have seen importance of Virtual machine placement in cloud environment with limitation of existing techniques. Such limitation has encouraged us to design and develop a unified framework for VM placement which can handle multiple resources and also works well at the enterprise level. We have used graph coloring reduction to understand complexity of the problem and to derive better bounds in our solution. The framework has been evaluated on four different types of instances and they have shown positive result and improved performance by 32%. In future, we are planning to include hard constraints in the framework like minimizing provisioning time, load balance and precedence constraints and evaluate with large size instances.

REFERENCES

1. M. Mishra and A. Sahoo, Title: "On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach," *2011 IEEE 4th International Conference on Cloud Computing*, Washington, DC, 2011, pp. 275-282. doi: 10.1109/CLOUD.2011.38"
2. A. Singh, M. Korupolu and D. Mohapatra, Title: "Server-storage virtualization: Integration and load balancing in data centers, 2008 SC - International Conference for High Performance Computing, Networking, Storage and Analysis, Austin, TX, 2008, pp. 1-12. doi: 10.1109/SC.2008.5222625"
3. Wood T., Cherkasova L., Ozonat K., Shenoy P. (2008) Title: "Profiling and Modeling Resource Usage of Virtualized Applications. In: Issarny V."
4. Schantz R. (eds) Middleware 2008. Middleware 2008. Lecture Notes in Computer Science, vol 5346. Springer, Berlin, Heidelberg
5. Yossi Azar , Ilan Reuven Cohen , Seny Kamara , Bruce Shepherd, Title: "Tight bounds for online vector bin packing, Proceedings of the forty-fifth annual ACM symposium on Theory of computing, June 01-04, 2013, Palo Alto, California, USA [doi>10.1145/2488608.2488730]"
6. Nikhil Bansal , Arindam Khan, Title: "Improved approximation algorithm for two-dimensional bin packing, Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, p.13-25, January 05-07, 2014, Portland, Oregon"
7. Nikhil Bansal , Kang-Won Lee , Viswanath Nagarajan , Murtaza Zafer, Title: "Minimum congestion mapping in a cloud, Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing, June 06-08, 2011, San Jose, California, USA [doi>10.1145/1993806.1993854]"
8. Nikhil Bansal, Tjark Vredeveld, and Ruben van der Zwaan, Title: "Approximating vector scheduling: Almost matching upper and lower bounds, in LATIN, 2014, pp. 47--59."
9. N Bansal, M Eliáš, A Khan - Proceedings of the twenty-seventh annual 2016 - dl.acm.org Title: "Improved approximation for vector bin packing"
10. V. Natu and T. N. B. Duong, Title: "Secure Virtual Machine Placement in Infrastructure Cloud Services," 2017 IEEE 10th Conference on Service-Oriented Computing and Applications (SOCA), Kanazawa, 2017, pp. 26-33".
11. Weijia Song, Zhen Xiao, Qi Chen, Haipeng Luo, Title: "Adaptive resource provisioning for the cloud using online bin packing", IEEE Transactions on Computers, vol. 63, no. 11, pp. 2647-2660, 2014. James, Reinders.(2007). Intel Threading Building Blocks: Outfitting C++ for Multi-core Processor Parallelism. O'Reilly Media, Inc. Sebastopol, CA, 2007".
12. https://docs.microsoft.com/en-us/azure/virtual-achines/windows /sizes

AUTHORS PROFILE

Darshan Shah Research scholar and currently working as cloud technology lead in the Fortune 500 company. He has and worked for clients and companies like Microsoft, Dell R&D, Accenture and Cognizant. He has more than 11 years of industry experience and 3 years of research experience in High Performance Computing, Grid Computing and Optimization. He has published and presented his research works in more than 5 national and international conferences.

Dr. M. Vinayakmurthi Professor, Ph.D. in "Computational Fluid Dynamics-Mathematics" from Bangalore University, M.Sc. in Mathematics, B.Sc. in Mathematics from Bharathidasan University and also B.Ed. degree in Mathematics from Anamalai University He has 26 years of teaching experience in UG, PG and Ph.D. and more than 45 research papers in the reputed journals. His areas of Interest include Discrete Mathematics, Probability and Statistics, Operation Research, System Simulation and Modeling, Finite Automata Theory, Analysis and Design of Algorithms, Computer Graphics, Data Mining and Data Warehouse.

Dr. Anand Kumar Dean (Academics) and Professor in department of CSE of MS Engineering College, Bangalore and have more than 14 years of experience. His interested research areas are Evolutionary Computing, Genetic Algorithms and Network Design. He has authored more than 25 research papers in International Journal and International - National Conferences. I serve the editorial board of many reputed International Journals and associated with several International Organization.