Improved Mathematical Model for Virtual Machine Placement Optimization with Resource Constraints

[1]Darshan Shah, [2]Dr. M. Vinayakmurthi, [3]Dr. Anand Kumar

ABSTRACT-- *Virtual Machine placement problem is a hard-combinatorial optimization problem in cloud data center. In this paper we have described problem in detail with its importance and challenges. We have also proved that the problem is NP complete when multiple dimensional resources are considered during placement. To optimize such a complex problem, we have developed new MILP mathematical model by including various constrains. An empirical evaluation of this model has surpassed existing state-of-the-art MILP and branch and bound models and shown evaluation on different data sets have shown zero percentage gaps for more than 70% instances.*

Keywords-- *Virtual Machine Placement, Mathematical Model, Multi-dimensional resource optimization, Cloud data center.*

I. INTRODUCTION

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.

In IaaS, Virtual Machine (VM) is one of the widely used component. Because of its versality, user can host their application, store data and also installed any software or configure operating system based on requirement and it has increased a lot in past few years. Below data from Cisco index [1] clearly shows that Cloud resource consumptions and significance of effective resource management by provider,

---

[1]Computer Science Department, Reva University, Bangalore, India.
[2] Computer Science Department, Reva University, Bangalore, India.
[3] M.S. Engineering College, Bangalore, India.

Received: 22 Sep 2019 | Revised: 13 Oct 2019 | Accepted: 15 Jan 2020

Global data center workloads and compute instances in millions							
	2016	2017	2018	2019	2020	2021	CAGR 2016-2021
Traditional data center workloads and compute instances	42.1	41.4	40.8	39.1	36.2	32.9	-5%
Cloud data center workloads and compute instances	199	262	331	393	459	534	22%
Total data center workloads and compute instances	242	304	372	432	495	567	19%
Cloud workloads and compute instances as a percentage of total data center workloads and compute instances	83%	86%	89%	91%	93%	94%	-

Figure 1: Cisco Cloud Report [1]

The VMs runs in cloud data center as a layer of virtualization which creates an abstraction in the cloud environment. This virtualized abstraction creates a challenge in effective management and monitoring. For example, a server running multiple VMs can have many requirements such as regular update, security patches, critical application uptime and as well as performance of CPU intensive jobs. To understand it better let us take an example. A server with the configuration to perform high memory usage creates an issue for other VMs which also needs the high memory in the same server. So, the right placement of VM is very important so that similar needs conflicted and creates an issue.

Below are few major challenges of efficient VM placement in the cloud data center,

i. In a virtual environment, total system memory should be assigned to individual virtual machines (VMs) in accordance with their memory usage. Under provisioning memory to a VM can hinder performance, forcing a guest VM to resort to paging – that is, storing and retrieving data from secondary (or slower) storage.

ii. Inefficient VM placement often leads Virtual Sprawl. In which VMs on administrator loses the control over the network reaches and unable to manage them effectively or it can also create the situation where VMs may demands unnecessary high amount of machine resources.

iii. The various types of servers and storages are used by users and they all are inter connected in the cloud network and this is another area where virtualization can encounter potential problems. Bottlenecks can occur when multiple VMs contend for network access on the same physical server, prompting user complaints about application access or performance problems.

There are also very important environmental parameters affect in effective resource management such as number of users used the resource, location, neighborhood, dependency of between service, total cost, scalability, performance moving or migrate cost of server, the networking within the organization, overall demands of the user or organization.

To address above major challenges and important environmental parameters related to resource management, we designed Virtual Machine placement problem with multiple resource dimensions and with different constraints. A mathematical model is designed and developed which will help to solve the problem as enterprise level rather solving just individual domain level.

Our major contributions are as follows,

1. Analyzed the complexity VM placement problem which will help to understand the complexity of the problem and better model design.

2. Designed and developed Mathematical model for Virtual Machine Placement problem with practical environmental constraints which helps to utilize machine resources and VM management challenges.

3. Experimental evaluation of mathematical model with large instance sizes and comparison with state-of-the-art models and algorithms.

This paper is managed into different sections, the section 2 discussed related work, and the section 3 formally described and analyzed proposed problem. Section 4 described mathematical model along with constraints used in the model. Section 5 showed experimental evaluation of model with current models and compare the results and section 6 concludes the paper.

II. LITERATURE STUDY

Due to problem complexity various mathematical programming-based solutions are applied to get better result in Virtual Machine placement problem. In the paper [1] authors have applied Integer linear programming (ILP) formulation to choose best host. The overall mechanism is to build different test data points and then compare it with optimum solutions which will help them to give better bound of the solution. In [2] multi-Objective Mixed integer linear programming (MILP) model and main constrain is to reduce the number of VM rejection by resource wastage. In [3] author has used Branch and bound algorithm which operates directly on the map function instead using binary variables. The algorithm works with partial solutions, in which map function defined for a subset of the VMs, and then do the traversal over the space of partial solutions in a tree-like manner. Child nodes from the tree are obtained by selecting a VM that is not yet mapped and trying to map with all the PMs available with enough capacity. Once such child nodes are identified from the partial solution they are mapped with the appropriate host. In paper [4], communication costs among different services is minimized during virtual machine placement. The VM placement problem is reformulate quadratic assignment problem as MILP model which solved the problem by obtaining global optimum solution. The two main matrices are used, Cost of communication and bandwidth consumption cost. Later these two matrices are used to formulate and build the constraints. In the paper [6], VM placement problem is solved using MIP and consider anti-colocation as objective to make sure that its virtual disks are spread across physical disks of the Physical Machine.

III. VIRTUAL MACHINE PLACEMENT PROBLEM DISTRIPTION AND ANALYSIS

To understand Virtual machine placement problem let us take very well-known bin packing problem. In bin packing each item must be packed bins such way that all items are packed and total bin size must be minimized. Now if we extend this to more special case known as vector bin packing problem in which each item contains vectors and item must be place in bins such a way that all vectors are diagonal to each bin and total bin size must be minimized. The same way in virtual machine placement problem each VM known as item and bin as machine and resources as vectors and overall goal is to place them in minimum number of machines such a way that all resources are utilized (vectors diagonal to bin) of that machine and total machines sized minimize (bin size minimized). Below diagram shown the graphical view of the problem,

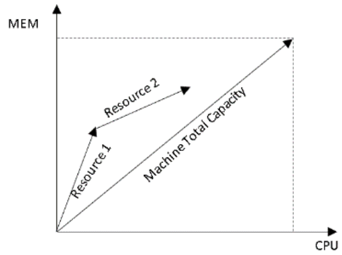

Figure 2: VM Placement Problem with multiple resources

Each VM request follows specific flow in cloud computing environment. Cloud computing is an on-demand computing platform in which user can submit any type of VM and with any number of capacities. This complete placement follows specific procedure to provision(place) in the cloud data center. Below is such flow of the VM placement in the data center.

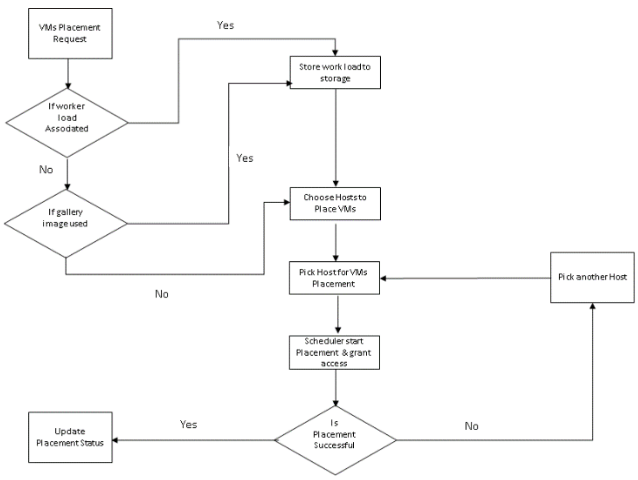

Figure 3: Virtual Machine Placement Flow

3.1 Complexity of Virtual Machine Placement

The Virtual Machine Placement (VMP) problem is NP complete with large instance size. To understand its complexity let us reduce this problem to 3SAT problem.

Theorem 1: Virtual Machine Placement problem with multiple resources is NP complete problem.

3SAT $\leq p$ VMP

Lemma 1: VMP problem is in NP.

Proof:

Let us have a set of VM instances know as *S* with resources *R*, and set of Machines M, now a verifier can contain *k* such instances in *O(k)* time such a way $s \in S$ VMs are assigned to $m \in M$ machines in the polynomial time without violating the capacity of machines. The same way a verifier can identified *No* such instances in the polynomial time $s \in S$ which are not allocated to $m \in M$ machines.

Lemma 2: VMP problem is NP complete.

Proof:

To prove that VMP is NP complete let us reduce this problem to 3SAT problem which is NP Complete problem in following manner. The 3SAT problem have instances with

formula $\alpha = C1 \wedge .... Cm$ in which each term is containing values *x1, x2.... xn* and its negation is represented as *x1~, x2~.... xn~*. Using this now let's constructed instance as *(V, k)* of VMP problem as below, Let *P* represents the set of unique terms available in the β such a way terms $pi \in \beta$ in each clause which creates a VM instances in machine *mi*. Since there are multiple such clauses available as *x1* and *x1~* which indicates the allocation and not allocation.

For any such VM instance of *vi* with allocation as with clause *x1* and without allocation with *x1~*. Apply this connection to three such VMs. This we called a 'clause gadget'. At the end, for all the variables in the clause *xi* add connection to *x1* and other to *x1~*. Now verify the connection, each clause associated with *x1* available as allocated with *Yes* instances of 3SAT and *x1~* instances as *No* instances. This way it shows the Virtual Machine problem is NP complete problem. The whole process runs in the polynomial times.

This complexity analysis helps later to build better model and configuration of different constraints within the model which helps to reduce the gaps in the large size instances.

IV. MATHEMATICAL MODEL FOR THE VIRTUAL MACHINE PLACEMENT PROBLEM

Virtual Machine placement is formulated as a mathematical model to be sure there are no ambiguities and performance issues after placement. In this section, a problem is modeled as mixed integer programming with below placement binary variables and continuous variables.

4.1 Decision Variables

$Xvm \in (0, 1)$ in which 1 if VM is placed to machine $m \in M$ else 0.

$Yvm \in (0, 1)$ in which 1 if VM is indicates virtual machine $v \in V$ changed from one place to another place.

$Omr \in \mathbb{R}+$ represents utilization of machine $m \in M$ over the resource $r \in R$.

$Bvm \in \mathbb{R}+$ represents whether virtual machine $v \in M$ has neighborhood $n \in N$.

$load\ mr \in \mathbb{R}+$ load cost for a machine $m \in M$ for the $r \in R$.

$loadcostmr \in \mathbb{R}+$ is total load cost.

$balancebm \in \mathbb{R}+$ is balance cost $b \in B$ for machine $m \in M$.

$balancecost \in \mathbb{R}+$ is total balance cost.

$mmcost \in \mathbb{R}+$ is machine move cost for $m \in M$.

$tmcost \in \mathbb{R}+$ is total machine move cost for $m \in M$.

totalcost $\in \mathbb{R}+$ is total cost of machine usage during virtual machine placement which needs to be minimized.

4.2 Hard Constraints

The list of below hard constrains needs to be satisfied during Virtual Machine placement. These constraints help to manage environment parameters which we discussed in the first section of the paper.

4.2.1 Volume Constraints

Let us represent *R* as a set of resources be part of different machines $m \in M$ as *C (m, r)* and capacity of resources $r \in R$

for machine $m \in M$ and *R (v, r)* the requirement of resource $r \in R$ for the machine $v \in V$. Then, given placement *M*, the resource utilization of machine *m* for with the resource *r* is defined as below,

$$O\ (m, r)\ = \sum_{v=1}^{n} R(v, r)$$

Virtual Machine will be provisioned over the machine only if sufficient capacity is available at the machine with all the required resources. This can be represented as below,

$$\forall\, m \in M, r \in R, O\ (m, r) \leq C\ (m, r)$$

4.2.2 Clash Constraints

There are different types of Virtual machines available such that similar pair of machines should be allocated at the same machine sets. This constraint makes sure that such virtual machines are disjoints.

$$\forall\, v \in V, (vi, vj) \Rightarrow M(vi) \neq M(j)$$

4.2.3 Expansion Constraints

Let us represent *L* as set of available in the data center such a way that location $l \in L$ represents as collection of machines.

Here a collection of locations represents as disjoint set as,

$$\forall e\ \in, E \sum_{l \in L}^{n} min\ (1, |\{\, v|\ m\ (v) \in l\}|)\ \geq expansionMin\ (v)$$

4.2.4 Dependency Constraints

Let us represent N as neighborhoods, which contains collection of machines. All the neighborhoods are representing has as disjoint sets.

If virtual machines *v1* is depends on *v2* due to SLA cloud subscription, then each such request should be fulfilling accordingly such that they placed at right location due to interdependency. This constraint can be formulated as below,

$\forall\, vi \in V, \exists\, vj \in V$ and $n \in N$ such that

$M(vi) \in n$ and $M(vj) \in n$

4.3 Objective

The overall objective is to improve machines utilization with all its associated resources without violating above mentioned constraints. The total objective cost is constructed by combining various cost such as Load cost, Balance Cost and Machine move cost.

4.3.1 Load Cost

Let S (m, r) be a load cost constraint on the machine m ∈ M for resources r ∈ R. The load cost if

$$\alpha 1 = \max(0, \mathrm{O}(\mathrm{m}, \mathrm{r}) - \mathrm{S}(\mathrm{m}, \mathrm{r})$$

then for each resource and corresponding to the used capacity as per the load defined,

$$LoadCost(r) = \sum_{m=1}^{n} \alpha 1$$

4.3.2 Balance Cost

A correct placement should have all required resources for the VM. For example, machine required memory, but CPU cores are not enough then such machine is not appropriate for the virtual machine placement. This constraint takes care of such allocation and states a kind of 'dependency' between inter dependent resources.

Let B be the set of dependencies between the resources. These dependencies are defined by pair of resources and targeted dependencies. For example, we have $b = (r1, r2, t) \in B$ and $B \subseteq \mathbb{R}^2 \times \mathbb{N}$. Let *F (m, r)* be a free space

available in the resource $r \in R$. The machine m ∈M then,

$$F(m, r) = C(m, r) - O(m, r)$$

The balance cost $b \in B$ is

$$BalanceCost(b) = \sum_{m=1}^{n} \max\big(0, T(b) * F\big(m, r1(b)\big) - F(m, r2(b)\big)$$

4.3.3 Machine Move Cost

The cost of moving machine m from *M0 (v) to M(p) = m (if M (v) = M0 (v)* then such moving cost is equal to zero. The moving cost of machine is overall sum of all machines moved. Then its cost is,

$$MMC = \sum_{m=1}^{n} M(mi, mj)$$

4.3.4 Total Objective Cost (TC)

$$TC = Load\ Cost + Balance\ Cost + Machine\ Move\ cost$$

4.4 Mixed Integer Programming (MIP) Mathematical Model

$$\text{Minimize}\ (Z) = Z1 = \sum_{r \in R}^{n} Cr \sum_{m \in M}^{n} \alpha mr \qquad (\textit{load cost})$$

$$+\ Z2 = \sum_{(r1,r2) \in B}^{n} C(r1,r2) \sum_{m \in M}^{n} \alpha(r1,r2) \qquad (\textit{balance cost})$$

$$+\ Z3 = \sum_{v \in V} vi \sum_{m \in} mi \qquad (\text{machine move cost})$$

s.t.

$$\sum_{v \in V}^{n} Rvr\ Xmv \leq \text{Cmr} \qquad \forall m \in M, r \in R$$

$$\sum_{v \in V}^{n} Rvr\ Xmv + \sum_{v \in V}^{n} Rvr\ Xmv \leq \text{Cmr} \qquad \forall m \in M, r \in R$$

$$\sum_{v \in V}^{n} xvm \leq 1 \qquad \forall v \in V$$

$$LC = \sum_{v \in V}^{n} Rvr\ Xvm - \text{SCmr} \qquad \forall m \in M, \quad r \in R$$

$$BC = m,r1,r2 \geq Tr1,r2\ \left(Cm,r1 - \sum_{v \in V}^{n} Rvr1\,.Xvm\right) - \left(Cm,r2 - \sum_{v \in V}^{n} Rvr2\,.Xvm\right)$$

$$\sum_{m \in M}^{n} xvm = 1 \qquad \forall m \in M$$

$$\sum_{l=L} E \geq \text{SpreadMin}\ v \qquad \forall v \in V$$

$$\sum_{m \in M} E \leq 1 \qquad v \in L, l \in L$$

$$\sum_{l=L} E \leq \text{Xpm} \qquad \forall v \in V$$

$$X_{pm} \in \{0,1\}$$

V. EXPERIMENTAL EVALUATION AND RESULTS

Our mathematical model for Virtual Machine placement problem is evaluated using our prototype implementation on a 10 data sets collected from private data center running azure cloud platform.

5.1 Infrastructure

System running windows 10 operating system, 16 GB RAM, intel i7 6500 CPU @2.50GHz 2 cores and 4 logical processors.

5.2 Workload

The multiple resources like CPU, memory and network bandwidth are taking into consideration and randomly picked up the data set from the 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.

Test Set	VM	PM	Resources	Load	Locality	Dependency	Balance
T_1	100	20	5	20	20	5	20
T_2	200	20	10	40	20	5	20
T_3	300	20	10	40	20	5	20
T_4	400	20	10	40	20	5	20
T_5	500	20	10	40	20	5	20
T_6	1000	50	15	100	20	10	50
T_7	2000	75	15	150	20	15	50
T_8	3000	75	15	150	20	15	100
T_9	4000	100	20	200	20	20	125
T_10	5000	125	20	250	20	20	150

Figure 4: Test Instances

5.3 Highlights

1. Our mathematical model has shown good positive results and first 7 test instances are without gap and rest 3 has very minor gap.

2. Current MILP model and branch and bound model has gaps in most of the instances and unable to reach at the optimum level.

5.4 Experimental Results

The simulation result shown in the below figure [5] that our model has shown good result for top 7 instances with zero percentage gap for VM placement problem. Also, it greatly reduces the resource consumption without violating any constraints. Furthermore, in the figure [6] we have evaluated it with MILP formulation without new constraints and Branch and Bound formulation. Both results are not optimally and produced the gap. For the case of MILP, it went out of memory due to large size of variables. Branch and bound without any parallel branching shows that it unable to identify the optimum within a given time period.

Below is the result of all ten instances used for evaluation of our mathematical model.

Test Set	Upper Bound	Lower Bound	Gap(%)
T_1	44306501	44306501	0
T_2	666530829	666530829	0
T_3	583125515	583125515	0
T_4	242404632	242404632	0
T_5	427678196	427678196	0
T_6	54350836	427677796	0
T_7	559888659	559888659	0
T_8	2272487840	1007955933	11.77
T_9	2680231407	1680231407	21.45
T_10	1680231407	307041984	32.69

Figure 5: Result with our model

Test Set	MILP Gap (%)	Branch & Bound Gap (%)
T_1	0	0
T_2	6.78	0
T_3	11.67	56.77
T_4	22	12.56
T_5	67.45	3.5
T_6	34.54	12.55
T_7	56.77	45.34
T_8	33.35	23.56
T_9	23.56	55.34
T_10	45.67	67.77

Figure 6: Result of MILP with limited constraints & Branch and Bound

VI. CONCLUTION AND FUTURE WORK

In this paper we have provided improved mathematical model by adding more complex constraints for VM placement problem. We have described the problem and analyzed its complexity by reducing it to 3 SAT problem. With our new model machine utilization is improved compare to current MILP models and branch and bound. A model is then tested with mid-size to large size instances and found that more than 70% instances have shown zero percentage gap and rest are very less gap. This is good positive outcome compare to existing models where first they are in limited constraints and has gaps. Apart from this, model is general enough to extend further to accommodate other constraints and objectives. Our next goal is to improve our result for very large size input and reduce the gap shown in the experimental result.

REFERENCES

1. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html
2. "REGAIEG, Rim & Koubaa, Mohamed & Osei-Opoku, Evans & Aguili, Taoufik. (2018). Multi-Objective Mixed Integer Linear Programming Model for VM Placement to Minimize Resource Wastage in a Heterogeneous Cloud Provider Data Center. 10.1109/ICUFN.2018.8437036.
3. Bartók, Dávid & Mann, Zoltan. (2015). A branch-and-bound approach to virtual machine placement.
4. Lin, Ming-Hua & Tsai, Jung-Fa & Yi-Chung, Hu & Su, Tzu-Hsuan. (2018). Optimal Allocation of Virtual Machines in Cloud Computing. Symmetry. 10. 756. 10.3390/sym10120756.
5. Mayank Mishra and Anirudha S., Title: "On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach," 2011 IEEE 4th ICCC, Washington,DC,2011,pp.275-282.doi: 10.1109/CLOUD.2011.38"
6. 'Y. Xia', 'M. Tsugawa',' J.A B. Fortes' and S. Chen, "Large-Scale VM Placement with Disk Anti-Colocation Constraints Using Hierarchical ecomposition and Mixed Integer Programming," in IEEE TPDS, vol. 28, no. 5, pp. 1361-1374, 1 May 2017"
7. 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"

8. Wood T., Cherkasova L., Ozonat K., Shenoy P. (2008) Title: “Profiling and Modeling Resource Usage of Virtualized Applications. In: Issarny V.”
9. “Schantz R. (eds) Middleware 2008. Middleware 2008. Lecture Notes in Computer Science, vol 5346. Springer, Berlin, Heidelberg”
10. “V. Naatu and ‘TNB’ Duong, Title: "Secure Virtual Machine Placement in Infrastructure Cloud Services," 2017 IEEE Conference (SOCA), Kanazawa, 2017, pp. ‘26-33’”.”
11. “https://docs.microsoft.com/en-us/azure/virtual-achines/windows/sizes”