International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 8, August 2017)

Heuristic Based Resource Reservation Strategies for Public Cloud

Aravind C Bijjaragi[1], Ms Ranjana B. N.[2]

[1]Department of Studies in Computer Science and Engineering, VTU Belagavi, Karnataka, India
[2]Assistant Professor Department of Studies in Computer Science and Engineering, VTU Belagavi, Karnataka, India

***Abstract*— AT present in Cloud Arena Cloud service Providers (CSPs) use unalike pricing representations for their offered services. Few of them are appropriate for short term requirement however others for the long term requirements. Resources are accessible on reservation scheme, and on demand scheme. Reservation-based pricing model is suitable for a Cloud service User's (CSUs) extensive term claim resources, needs practical data for searching optimal number of resources for prior reservations, to reduce over-all cost (Overprovision situation). Conversely on-demand pricing model costing more than reserved one. Therefore, some optimization strategies are required to minimize the resource usage charges from cloud user views.**

A lot more methodologies exists, to give solutions to resource reservation claims and most of them have integer programming problem (IPP), which is NP in nature. In this work we derive heuristic methodological algorithm to find some near optimal solution to the problem initially by putting restrictions, later these restrictions are removed. We used Best-fit heuristic to achieve sub-optimal solutions, in this presenting work. Best-Fit heuristic algorithm attempts to approximate Bin-Packing strategy; which we have chosen in comparison with Exact Virtual Machine allocation proposal. Allocation and migration algorithms are used to minimize overall data-center power consumption.

***Keywords*— Amazon EC2, Best-fit, Bin-packing, Cloud IaaS manager, Cloudlet, Cost optimization, Energy aware VM scheduler, Energy estimation, Extended bin-packing, Global broker, IPP, peList, pesNumber, Public cloud, resource reservation.**

I. INTRODUCTION

AN advance in distributed computing technology provides an eagle view to computing world. Practices of dedicated access to computers replaced by on-demand accesses to resources shared between different organizations and many individuals. Basically we find two types of clouds for deployment, Amazon's EC2: Computing (e.g., processing or memory) instances created and are provided on-demand. Google's Map Reduce: Providing computing capacity on demand. One important dimension at which we escalate in this work is the resource provisioning plan.

Cloud providers have two kinds of cloud resource provisioning schemes: on-demand strategy and advance or long-term reservation strategy. Advanced resource reservation strategy has many specialties for sharing of resource materials. It gives simple strategy for resource planning and reservation in future time and above an increased requirement that resources get allocated when demanded. Although advance reservation strategy is more advantageous, we focus mostly on current demand plan.

At present the cloud service users (CSUs) face major challenge due to various pricing variations offered by the cloud service providers (CSPs). The objective is to design an optimal resource reservation heuristic, in cloud data centers, while considering different dimensions of the problem such as resource provisioning plan, cloud service model, etc.

II. RELATED WORK

We review related works in this section on heuristic based resource reservation strategies for public cloud.

Application-centric cloud architecture that provides a generalized framework for auto deploying, providing scalability, sharing, robust and high availability of cloud based applications. The goal of this architecture is achieving a cost effective, fault tolerant and auto-scalable web-application deployment across cloud-providers in [1]. In [2], Author states Cloud-Compute-Commodities (C3) pricing function as option pricing problem, and details cloud resources price model using a continuous-time approach; and address uncertainty inherent constraints in achieving required quality-of-service (QoS) through the use of technological and economic principles. In [3] financial option-based market-model is introduced for a federation of cloud providers, which helps providers to increase profit and mitigate risks. In [4] The Author formulated the revenue maximization problem as a finite-horizon stochastic-dynamic program, with stochastic demand, also characterizes optimality conditions for the stochastic problem. Author extended the model to the case with nonhomogeneous demand. And conducted an asymptotic analysis on this more general but difficult problem.

In [5], Author describes the optimal resource allocation using mixed integer programs. It proposes the light-weight optimal-solutions for problem. Works on an objective of whole processing time and reduces total cost with a given budget, and it also reduces total processing time. In [6], author considers the case of a single cloud provider and addresses best match customer demand. In particular, author model this problem as a constrained discrete-time optimal control problem, it becomes a challenging problem to determine the optimal way to allocate resources to optimize total revenue while minimizing energy cost. In [7], author proposed a cost model that takes into account the user's partial utility specification when the provider needs to adjust resources between VMs. CloudSim was extended to support scheduling model. Several simulation scenarios with synthetic and real workloads are presented, using datacenters with different dimensions regarding the number of servers and computational capacity.

III. Design And Implementation

A. System Architecture

It illustrates basic structure of program and data components which are needed to construct a computer-based-system. It takes into consideration that the architectural style, the structure and properties of the components that makes the system and the interrelationship that happen among all participants of a system. It represents the entire system as a whole along with their sub components.

Figure 3.1 depicts the system architecture model depicting the proposed energy efficient migration and allocation algorithms (contributing to scheduling), an energy availing estimator and a cloud manager (handling infrastructure resource instantiation and management).

Each module is briefly described to set the stage for the analytical modeling for validating into optimal resource reservation in clouds.

Cloud IaaS manager maintains cloud resources and handle cloud user queries, fetch and store images in storage spaces, and also manages VM scheduling.

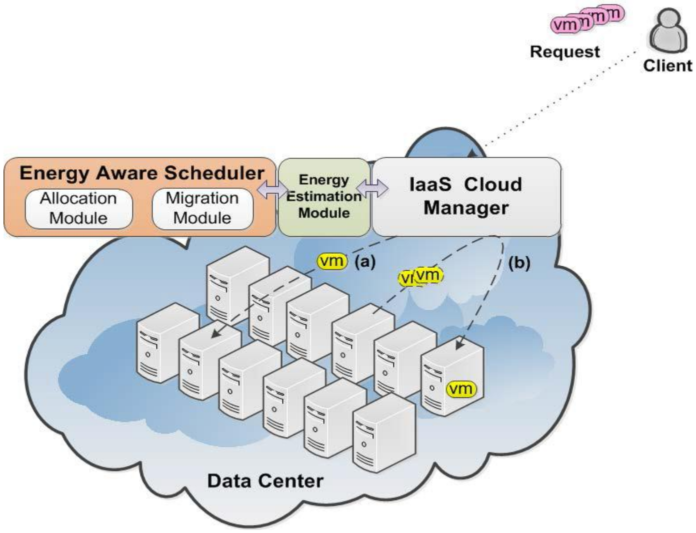

Figure 3.1 Proposed System Architecture Model

Energy estimation is a negotiator between cloud infrastructure and the energy (aware) scheduler.

Energy-aware VM scheduler is in-charge for VM (energy aware) placement, and data-center is main concern in energy consumption optimization model. This scheduler has two modules: i) Allocation ii) Migration. Allocation module initiates VM placement using Exact-VM allocation algorithm. Migration module dynamically consolidates VM that minimizes the number of activated servers and in-active servers are put into sleep mode or switched off. We adopted bin-packaging algorithm for prime placement of user requests and to follow with dynamic consolidation once a sufficient number of departures have occurred. Migration algorithm handles dynamic consolidation (i.e. regrouping of VMs to free as many servers) to put VM to sleep mode or in switch-off mode.

B. Implementation Modules

Cloudlet Models the cloud-based application services which are commonly deployed in the data centers. Cloudlets are the basic programming element or entity, which is similar to user application program, requiring cloud environment to start its activities.

We modeled Cloudlets in our Simulation based environment (Cloudsim) by approaching well known terminologies such as;

1. Cloudlet id – Module name or identifier.
2. pesNumber – Number of requiring processor elements.
3. Userid – User identifier of particular cloud application.
4. Length –Length of the application in Million instruction unit. Every application have pre-assigned instruction length
5. File size – Additional attachment size in Megabytes (MBs).
6. Output size – The output requirement in Megabytes (MBs).

Virtual Machine (VMs) Virtual machines are cloud entities, which models the virtualized resources in cloud environment. VM Scheduler determines how processing of host machines are get assigned to VMs (i.e. Number of processing elements (cores) will be get allotted to each virtual machine, and how much of processing core's capability efficiently be qualified for individual virtual machine).

The terminologies used in terms of virtual machine are:

1. *Vmid* – Virtual machine name or identifier.
2. *pesNumber* – Number of processing elements in Number.
3. *Mips* – Virtual machine capability of processing in terms of million instruction per seconds.
4. *Size* – Pre-assigned Virtual machine image size in Megabytes (MBs).
5. *Ram* – Virtual machine Random access memory requirement in Megabytes (MB).
6. *Bw* – Pre-assigned network bandwidth requirement in Mega Heartz (MHz).
7. *VMM* – Virtual machine monitor such as Xen and others.

Datacenter is an important entity in Cloud based simulation environment, which is being optimized for power consumption as well cost optimized by using Dynamic Voltage / Frequency Scaling (DVFS) used for VMs migration and consolidation methodologies. We also emphasized with the governance methodologies such as; Conservative, Performance, On-demand, User-spaces.

While modeling Powerdatacenter instantiating we use following terminologies;

1. *MaxPower* – Power requirement of the data center to operate in Watts
2. *Powerpercent* – Pre-assigned Power utilization factor in percentage
3. *Mips* – Pre-assigned datacenter capability in terms of processing of instructions i.e. million instruction per second.
4. *Ram* – Pre-assigned random access memory requirement for the datacenter including all hosts in it in terms of Megabytes (MBs).
5. *Storage* – Secondary memory capability required in datacenter consolidation of host nodes in terms of Megabytes (MBs).
6. *Bw* – Network requirement of the Datacenter in terms of Mega Heartz (MHz).
7. *Frequency* – Pre-assigned processor capability in Mega Heartz (MHz). Should be mentioned in increasing order in array initialization.

Host nodes in datacenter are initialized using following terms or parameters;

1. Hostid – Host node identifier.
2. Ram – Host node main memory capability.
3. Bw – Host node Bandwidth capability.
4. Storage – Individual Hosts Secondary storage capability.
5. peList – Number of Virtual resources from Host node.
6. Scheduler type – Virtual machine scheduler type i.e. of either Time shared or space shared.

Datacenter characteristics also need to be initialized such as;

1. *Architecture* – System architecture used in data center for the host nodes, such as x86 or x64.
2. *Os* – System operating system used in data center, such as Linux, Windows, etc.,
3. *Time zone* – Time zone of the datacenter workplace, in India it is (+5:30).
4. *VMM* – Virtual machine monitor adapted in datacenters, such as Xen.
5. *Cost* – Pre-assigned CPU cycle prices per unit time.
6. *CostperMemory* – Pre-assigned Main memory price unit per time.
7. *CostperStorage* – Pre-assigned Secondary memory price unit per time.
8. *CostperBw* – Pre-assigned Main memory price unit per time.

Broker is the mediator entity or an agent, between cloud service user and cloud service provider. This has responsibility of assigning cloudlet to virtual machine and by making use of heuristic approach fitting these virtual machines with servers or host nodes to serve the power as well as cost optimization purposes.

Brokers with or without events have local brokers meant for individual or bulk number of VMs are getting their server or host nodes.

Global broker is an important entity in our study; we applied an energy-efficient resource-allocation heuristic strategy here to gain in optimizing cost and power consumptions. We applied the Best-fit bin packaging heuristic in optimizing power as well cost optimizations.

Advance resource reservation offers simple way for resource planning in future. In Reservation plan the resources could be reserved earlier and the resource availability is ensured in future. There are two kinds of resource allocation they are; static and dynamic allocation. Static resource allocation is performed initially when requests arrive. Dynamic resource allocation is used to manage resources are derived for homogeneous data centers that surround observing abilities and probes. The dynamic resource allocation or consolidation controlled by Virtual Machine live migration and aims to minimize number of activated or used servers.

The Exact Virtual Machine allocation is an Extended Bin packing methodology over the presence of accepted settings stated in form of limitations or variations. Intension here is to keep Virtual Machines (VMs) to server or node hosting the Virtual Machine depends on power-usages. In consideration to ‘n’; number of requested Virtual Machines (VMs), we define the number of servers, m, available in the data center. The servers are assumed to have same power consumption limit: Pj; Max. While in execution, VMs of an individual server ‘j’ described by their present power usages Pj; current. Because our prime concern is in reducing energy usages of data-centers, so we define key decision variable ej for each server j that is set to 1 if server j is selected to host Virtual Machines (VMs), 0 if it is not. In addition, we define the bivalent variable xij to indicate that VMi has been placed in server j and set xij to 1; xij = 0 otherwise (if not).

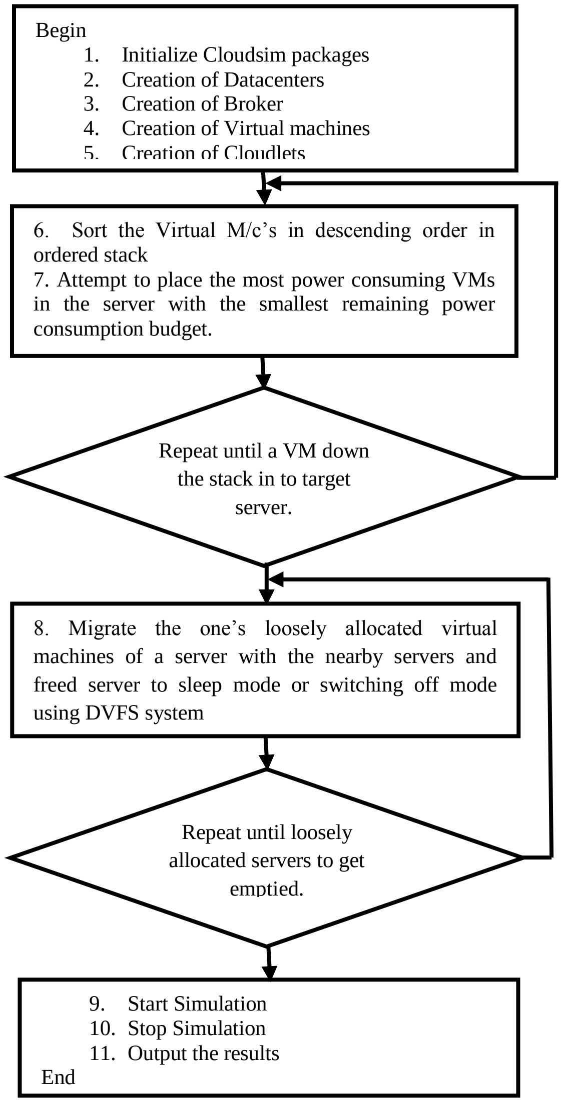
Begin
1. Initialize Cloudsim packages
2. Creation of Datacenters
3. Creation of Broker
4. Creation of Virtual machines
5. Creation of Cloudlets
6. Sort the Virtual M/c’s in descending order in ordered stack
7. Attempt to place the most power consuming VMs in the server with the smallest remaining power consumption budget.
Repeat until a VM down the stack in to target server.
8. Migrate the one’s loosely allocated virtual machines of a server with the nearby servers and freed server to sleep mode or switching off mode using DVFS system
Repeat until loosely allocated servers to get emptied.
9. Start Simulation
10. Stop Simulation
11. Output the results
End

Figure 3.2 Flow of Best-fit Heuristic with Migration Process (DVFS)

We have implemented the Best-fit heuristic to accomplish sub-optimal solutions. Heuristic algorithm anticipated to realize energy-efficient VM placement consists of following two steps:

Best-Fit Heuristic Algorithm

1. Sorting the requested Virtual Machines (VMs) in falling order of power usages, which constructs ordered stack employed in further steps for packing Virtual Machines in available servers.
2. The sorted Virtual Machines held initiating from top of stack and trying to insert most power requiring VMs in server machine with lowest remaining power usage with low-priced machine, till a VM downcast stack in target server. The practice reprises up until every VM in stack are positioned, and packed as a lot as possible in utmost full servers. Tends to servers at liberty for sleeping mode or switched-off mode.

Best-Fit heuristic methodology attempts to near Bin-Packing method, in contrast with Exact Virtual Machine (VMs) allocation proposal. The allocation algorithms are shared with migration methodology to reduce complete data-center power consumption. The Best-Fit heuristic methodology chosen because it is well-known to attain worthy suboptimal performance related with traditional Bin-Packing heuristic.

C. Results and discussion

The heuristic based resource reservation algorithm implemented using java SDK v6, the proposed methodology has been executed on Intel Core i3CPU @ 2.20GHz processor with 6GB RAM.

An IPP is NP-Hard and cannot work satisfactorily beyond six months period. Amazon EC2 considered for implementing and evaluating efficient resource reservation algorithm. Therefore, the Amazon EC2 contracts for one-year and three-years have been scaled down to one-month and three-months respectively, using the following equations to keep the hourly discount of reserved VM over on-demand VM unchanged.

Reservation1month = Reservation1year / 12

Reservation3month = (Reservation3year /36) * 3

We are using one kind of VM in our simulation, which can handle multiple types also; we determine demands for such VMs to apply our heuristics, to find out amount of reservation. Properties of such VMs in our simulation are listed in table 3.1.

Table 3.1
Properties of VM used in simulation

Property	Value
Type of Virtual Machine (VM)	Standard large (Linux)
Reservation Cost (one month)	$20.25
Reservation Cost (three months)	$32.00
On-demand Usage Cost	$0.24 / hour
Reserved VM (1 month) Usage Cost	$0.136 / hour
Reserved VM (3 months) Usage Cost	$0.108 / hour

We are experimenting with four different sets of demand data, which is uniformly distributed through workload.

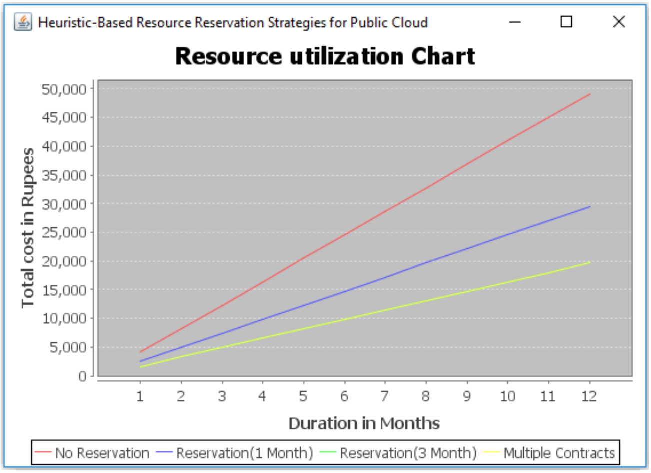

Figure 3.2: Total cost for reserving strategies

We listed out the costs for duration going from one to twelve months is shown in figure 3.2. The comparison of heuristic based strategies with multiple contracts, and single contract for one month period and with single contract for three-month duration. Total cost with no-reservation is also shown in the figure 3.2.

We observed following things from figure 3.2.

1) In comparison with no reservation, heuristic based reservation strategies made significant reduction in total costs.

2) Total cost reduces more in longer duration contracts than lesser periods
3) Demand data distributed (uniform or un-uniform) over a variety of reservation strategies are not affecting any performance metrics.

A relative performance is evaluated of system with Amazon's EC2 instances without any sort of scaling (scaled down) measures. We are not considering the IPP because of its constraint that it can't be used beyond six months duration.

The standard rating of Amazon's EC2 instance for one year contract is $243 and for three years it is $384 and discount for one year as $104. The equation □ Rk / αk □ evaluated for one year contract results to 2336 hours and for 3 year contract equate to 2909 hour when discount with $132. No reservation is made beyond 2336 hours for 1 year and 2909 hours for 3 years which is floor to the equation we considered.

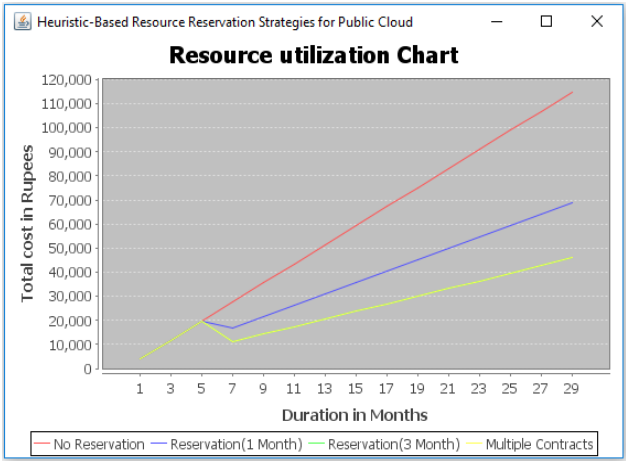

Figure 3.3 Total cost for Reservation strategies

IV. CONCLUSION AND FUTURE SCOPE

Integer Programming Problem (IPP) method generate very outsized numeral of variables thus does not conclude in resource reservation problem operating for periods even for six months. In this project, the system provides near optimal solution for using the cloud resources in finding power and cost optimizations. The linear time heuristic works on hourly basis for more than 3 years without any difficulty.

Some of the uncertain characteristics are not covered in this project like spot pricing scheme need to be covered in future time.

REFERENCES

[1] S. Khatua, A. Ghosh, and N. Mukherjee, "Application-centric Cloud management," in Proc. 9th IEEE/ACS Int. Conf. Comput. Syst. Appl., 2011, pp. 9–15.

[2] B. Sharma, R. K. Thulasiram, P. Thulasiraman, S. K. Garg, and R. Buyya, "Pricing cloud compute commodities: A novel financial economic model," in Proc. 12th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput., May 13–16 , 2012, pp. 451–457.

[3] A. N. Toosi, R. K. Thulasiram, and R. Buyya, "Financial option market model for federated cloud environments," in Proc. 5th IEEE Int. Conf. Utility Cloud Comput., Nov. 2012, pp. 3–12.

[4] X. Hong and L. Baochun, "Dynamic cloud pricing for revenue maximization," IEEE Trans. Cloud Comput., vol. 1, no. 2, pp. 158–171, Jul.–Dec. 2013.

[5] H. Menglan, L. Jun, and B. Veeravalli, "Optimal provisioning for scheduling divisible loads with reserved cloud resources," in Proc. 18th IEEE Int. Conf. Netw., 2012, pp. 204–209.

[6] Qi Zhang, Quanyan Zhu, Raouf Boutaba, "Dynamic Resource Allocation for Spot Markets in Cloud Computing Environments,", in Proc. 4th IEEE Int. Conf. Utility & Cloud Computing, 2011, pp.178–185.

[7] Jose Simao and Luıs Veiga, "Partial Utility-Driven Scheduling for Flexible SLA and Pricing Arbitration in Clouds,", IEEE Trans. Cloud Comput., vol. 4, no. 4, pp. 158– 171, Oct.–Dec. 2016.

[8] S. Khatua, P.K. Sur, R.K. Das and N. Mukherjee, "Heuristic-Based Resource Reservation Strategies for Public Cloud,", IEEE Trans. Cloud Comput., vol. 4, no. 4, pp. 392– 401, Oct.–Dec. 2016.