REVIEW ON LEASE SCHEDULING IN HAIZEA FOR IAAS CLOUD

Aarti Vig[1], Ritika Keswani[2]
[1]PG Scholar, Dept. of CSE, Institute of Technology & Management, Gwalior (M.P)
[2]Assistant Professor, Dept. of CSE, IPS-CTM, Gwalior (M.P)
Email: sweetyaarti.vig@gmail.com[1], ritika.keswani12@gmail.com[2]

Abstract

Cloud Computing is the technique of dynamic virtual provisioning of computing resources like processing power, storage, networking and Information Technology infrastructure through the Internet on pay per use basis from service providers. Customers can request for resources on the cloud from anywhere on the planet, by sending Lease through Internet to cloud service provider. Leases are resource provisioning abstraction which contains all required resources, the provisioning time of those resources, budget of customer etc. in terms of number of nodes, capacity of nodes, and duration of nodes. The service provider, then accepts all leases coming from customers, creates virtual machines for them, schedules those virtual machines, with the help of a lease manager. Lease manager make use of their algorithms for lease management, which also includes scheduling of leases. Haizea is an open-source virtual machine based lease management software. It allows user to send request in the form of leases. The algorithm which haizea make use of to search resources in its sample database is based on linear search. The time complexity of linear search algorithm to search a list of n elements is n. Since IaaS service provider at global level have extremely large number of resource in their datacenter. The time taken by linear algorithm rises proportionally with rise in resources. This research work is about development of an efficient algorithm based on binary search, since binary search takes less time than linear search.

Keywords: Best-Effort, Advanced Reservation, Immediate Leases, Economy Based Lease, Deadline Sensitive Lease, Linear search, Binary search.

I. INTRODUCTION

Cloud services are classified into three categories, on the basis of computing resources present in computer science ; they are infrastructure as a service which provides computing hardware resources as a service, platform as a service which provide software as a computing platform in the form of services to the customers , the computing platform provided in this service consist of tools and APIs which is used to create, configure and deploy the softwares without the complexity of buying and managing the platform resource. Software as a service which provides software applications such as email application, CRM etc to the customer as a service.

This survey is mainly about scheduling of leases and searching of resources in IaaS service provider’s data center, which uses hardware virtualization to provide computing resources to its customers. An open source virtual machine based lease management agent used here is Haizea, which enables user to send request in the form of leases. Haizea has its own lease format. Customers make use of this format and send request in terms of number of nodes, capacity of nodes, and duration of nodes[1]. Haizea takes scheduling decisions based on the coming leases and its scheduling algorithms. Haizea can take decision regarding scheduling of virtual machines but it cannot instruct the hypervisor to implement those decisions. Haizea requires an interface between it and hypervisor which can instruct hypervisor, this interface is known as virtual infrastructure manager such as Opennebula. Haizea can schedule leases in both simulation mode and Opennebula mode[1][2][3][4]. Haizea works in three types of mode, Opennebula mode, Unattended simulation mode and Interactive simulation mode. When haizea works in Opennebula mode, it generates virtual machine management decisions, Opennebula accepts those decisions and implements them by instructing hypervisor. Haizea supports three types of leases [1]:

- Advanced Reservation Lease
- Best-Effort Lease
- Immediate Lease

II. ADVANCED RESERVATION LEASE.

A customer who wants resources for certain fixed duration in future he makes use of advanced reservation lease. In case of advanced reservation lease the request for resources may look like “Need 10 Unix servers, each with 1.2 GHz processing capability and 50 GB memory for 2 months starting from January”. Here demand for 10 nodes of Unix with certain capabilities customer is for some fixed duration in future. Advanced reservation leases have higher priority and cannot be preempted in between of execution of lease[1].

III. BEST-EFFORT LEASE

When a customer request resources in the form of a Best-Effort lease, it means that, he needs resources as soon as they are available to service provider, but if the resources are not available currently, he is willing to wait until the resources are available. Best-Effort leases are of two types preemptible best-effort lease and non-preemptible best-effort lease.

When an advanced reservation lease arrives, which has higher priority then a best-effort lease, haizea first checks whether that lease can be fulfilled in future, if not that lease will be directly rejected and if yes then haizea checks whether it need to preempt any lease in order to schedule that advanced reservation lease. If haizea need to preempt certain lease to schedule advanced reservation lease, then haizea do that by stopping less priority best-effort lease and put it back in the queue and resume that best-effort lease again, after the completion of advanced reservation lease. Preemptible best-effort leases are used for performing batch jobs, which are non-interactive in nature The only downside of this lease is that they can preempted by higher priority leases like advanced reservation and immediate leases[1].

On the other hand if customer request resources through non-preemptible best-effort lease, by that he means, he can wait for the requested resources to become available but he don’t want them to be preempted in between of the lease execution by any high priority lease.

IV. BEST-EFFORT WITH DEADLINES

The advanced reservation leases are of higher priority than best-effort leases. Therefore when advanced reservation lease arrives preemptible best-effort lease get preempt. If advanced reservation leases keeps on coming best-effort leases will get starved for long period of time. The solution to this problem is that a deadline is set for every preemptible best-effort lease, which means that the best-effort lease needs to be executed before the set deadline. This indirectly set a limitation on number of advanced reservation leases that can preempt a best-effort lease during the latter’s execution time [7].

V. IMMEDIATE LEASE

Customer who need resources now, immediately, and don’t want to reserve them in future, the customer can do this by sending request in the form of Immediate Lease. If service provider may not be able to provide required resources immediately, the customer will change his requirement, according to what service provider is offering, but he will not ask to reserve the resources for future. If service provider cannot provide any requirement coming from customer side, the Lease will finally be dropped[5].

VI. ECONOMY BASED LEASES

In all the previously defined lease formats there is a major drawback. In any of the three leases shown above customers have no provision to tell anything about his/her budget customer may have to pay more then his budget since service provider has no way to know how much a customer is willing to invest. It may be possible that customer will be allocated with the resources having much higher price whereas low price resources would be available. Also he/she don’t have any opportunity to bargain for resources, a customer can ask for resources only once in a lease. If demanded resources are not available customer don’t have any other chance to ask for optional resources in the same lease. Because of this reason service provider will loose customer if he is not able to fulfill demand as per lease. Moreover a customer can never know the current market prices of the resources service provider is having, since the database of service provider doesn’t have any provision to show the market prices. The problem is for both sides, customer and service provider, both of them suffers considerable lose[6].

```
<lease-request arrival="00:00:00">
<lease preemptible="true">
<nodes>

<node-set number of nodes="1"
duration=”10:00:00” budget=”200”>
<res type="CPU" amount="100"
weight=”0.6”/>

<res type="Memory" amount="1024"
weight=”0.4”/>
</node-set>

<node-set number of nodes="1"
duration=”10:30:00” budget=”400”>
<res type="CPU" amount="100"
weight=”0.8”/>
<res type="Memory" amount="1024"
weight=”0.2”/>
</node-set>

</nodes>

<start>
</start>

<software>

<disk-image id="foobar.img" size="1024"/>
</software>

</lease>
</lease-request>
```

Procedure 1: Modified Lease Format[7]

This problem is solved in two modules. In module 1 lease format is changed. In module 2 an algorithm for filtering one node option out of two given in lease and searching resources in database on basis of customer’s budget and resources cost is created. For simulation purpose lease is shown as an XML file called Lease Workload Format (LWF). In order to include the facility to show budget of the customer, the lease format is changed. As shown in procedure1 two new tags are added along with the other tags showing the capacity of the resources. One of the newly added tags will contain the budget which a customer is willing to invest. This tag is named as budget tag. The other tag is named as weight tag. This tag is used to show the demanded importance or desired weight of the resource. In order to negotiate over capacity and budget with service provider lease format is changed to have two choices of demanded nodes rather than one. If service provider is unwilling to fulfill first choice, customer can ask for second choice or service provider will have an option to fulfill second choice in case he can not fulfill first choice. As shown in procedure2, the database of the service provider is also changed, here again a new tag is added, called as cost. This tag contains current market price of the resources. As a result prices of the resources will now be visible to customer. Haizea will accept leases in changed format, take out all parameters including budget and weight, and start searching for appropriate or feasible nodes available. It may possible that one or more nodes will satisfy the requirement. That means more number of alternative solutions may be possible. In order to find most optimal node out those nodes, multi criteria decision making method is used and here comes the role of weight tag. Weight is the desired importance of the requested node. A matrix is made with alternatives as rows and weight as columns and the node showing minimum deviation from positive ideal solution and maximum deviation from negative ideal solution will be selected as most optimal solution[9]. The database of service provider is shown

```
<site>
<resource-types names="CPU Memory"/>
<nodes>
<node-set number of nodes="4">
<res type="CPU" amount="100"
cost=”100”/>
<res type="Memory" amount="1024"
cost=”100”/>
</node-set>
<node-set number of nodes="4">
<res type="CPU" amount="100"
cost=”100”/>
<res type="Memory" amount="1024"
cost=”100”/>
</node-set>
<node-set number of nodes="4">
<res type="CPU" amount="100"
cost=”100”/>
<res type="Memory" amount="1024"
cost=”100”/>
</node-set>
```

```
<node-set number ofnodes="4">
<res type="CPU" amount="100"
cost=”100”/>
<res type="Memory" amount="1024"
cost=”100”/>
</node-set>
</nodes>
</site>
```

Procedure2: Sample Modified database of service provider[7]

Suppose service provider shortlist m alternatives which satisfies the capacity required by customer and budget of customer than the next job of service provider is to find the most optimal alternative out of those m. The method used here is TOPSIS

Multi Criteria Decision Making

First of all an evaluation matrix is created with m alternatives and n criteria, here criteria is weight given with each resource type, therefore we have a matrix $(x_{ij})_{m\ x\ n}$. The matrix $(x_{ij})_{m\ x\ n}$ is then normalized to a named,

$R = (r_{ij})_{m\ x\ n}$.

Then weighted normalized decision matrix is calculated say

$T = (t_{ij})_{m\ x\ n} = (w_j r_{ij})_{m\ x\ n}$, i =1,2,.....m , j = 1,2,.....n.

Then the most optimal solution is determined as:

So = max{tij}

VII.LIMITATIONS IN CURRENT ALGORITHMS

The algorithm which haizea make use of to search resources in its sample database is linear. The time complexity of linear algorithm to search a list of n elements is n. Since IaaS service provider at global level have extremely large number of resource in their datacenter. The time taken by linear algorithm rises proportionally. At present service provider have to set the current market prices of the resources manually[9].

VIII. PROPOSED SOLUTION

Our research work is to design an algorithm which will make use of binary search instead of linear search since the time complexity of binary search algorithm to search an array of size n is log n the searching time based considerably low even though the number of resources in the database is too high.

For this we add a parameter named “demand” with each resources in the database this demand parameter states the current market demand of that particular type of resources .the resources in the database are arranged in the decreasing order of their demand in the market, when leases arrives to haizea, haizea attaches a value with the demanded resources according to their market demand and make use of that value to search in its database in binary fashion.

REFERENCES

[1] Borja Sotomayor, 1.0 Beta 2, *http:// haizea.cs.uchicago.edu/haizea_manual.pdf.*

[2] Mehta H.K., Gupta Eshan, “Economy Based Resource Allocation In IaaS Cloud”, *International Journal of Cloud Applications and Computing”, Volume 3, Issue 2, pp. 1-11, 2013.*

[3] B. Sotomayor, K. Keahey, and I. Foster, "Combining Batch Execution and Leasing using Virtual Machines," *Proc. 17th Int'l Symp. High Performance Distributed Computing* (HPDC 08), ACM Press, 2008, pp. 87–96.

[4] Llorente IM, Rubén S. Montero, and Ian Foster , “Virtual Infrastructure Management in Private and Hybrid Cloud”, *Internet Computing, IEEE,* pp: 14-22, *Volule 13, Issue 5, 2009.*

[5] Eymann, T., Reinicke, M., Streitberger, W., Rana, O., Joita, L., Neumann, D., & Zini ts. Multiagent and Grid Systems, “leasing in cloud systems using the opennebula engine” *In Proceedings of the Workshop on Cloud Computing and its Application*, pp. 297–307.

[6] Abramson, D., Buyya, R., & Giddy, J.. “A computational economy for grid computing and its implementation” *in the Nimrod-G resource broker. Future Generation Computer Systems, 18(8), 061– 1074. doi:10.1016/S0167-739X(02)00085-7.* (2002).

[7] Nathani, A., Chaudhary, S., & Somani, G. “Policy based resource allocation in IaaS cloud”, *Future Generation ComputerSystems,28(1),94–103.doi:10.1016/j.future .2011.05.016* (2012)

[8] Ardaiz, O., Artigas, P., Eymann, T., Freitag, F., Navarro, L., & Reinicke, M.. “The catallaxy approach for decentralized economy-based allocation in grid resource and service

markets*",AppliedIntelligence,25(2),131–145.doi:10.1007/ s10489- 006-9650-9.* (2006).

[9] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms:

[10] Akhani, J., Chuadhary, S., & Somani, G. "Negotiation for resource allocation in IaaS cloud", *In Proceedings of the Fourth Annual ACM Bangalore Conference (p. 15).* (2011, March).

[11] Sotomayor, B. (2007). "A resource management model for vm-based virtual workspaces". *Masters Thesis, University of Chicago.*

[12] Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, pp 25(6), 599–616. doi:10.1016/j. future.2008.12.001.

[13] Sotomayor, B., Keahey, K., Foster, I., & Freeman, T. . "Enabling cost-effective resource leases with virtual machines". *In Proceedings of the Hot Topics Session in ACM/IEEE International Symposium on High Performance Distributed Computing*. (2007).

[14] Sotomayor, B., Montero, R. S., Llorente, I. M., & Foster, I. . "Capacity leasing in cloud systems using the opennebula engine". *In Proceedings of the Workshop on Cloud Computing and its Application.* (2008).