User Defined Time and Cost Constraints Based Resource Allocation algorithm in Cloud

Swati Khemka, Mehul Mahrishi
Department of Computer Science and Engineering,
Swami Kesvanand Institute of Technology, Management & Gramothan, Jaipur, India
Email: swati.khemka001@gmail.com

Received 15 July 2017, received in revised form 5 August 2017, accepted 10 August 2017

Abstract: **Cloud computing enabled on-demand provisioning of computational resources over the internet. It represents a distributing computing mechanism by utilizing the high speed network. Cloud service providers always try to increase profit, while cloud user wants to reduce budget and meet deadline with satisfying all the requirements. Since both do not exchange much information with each other, this makes the resource allocation in cloud a tedious job.**

In this paper we introduce a time and cost weightage based resource allocation algorithm, i.e. User Defined Time and Cost Constraints Based Resource Allocation algorithm which decide the selection of resource either on the basis of cost or on the basis of time. The time and cost constraints are submitted by user along with the request. We implemented this algorithm using CloudSim. Experimental results show that our approach either reduces execution cost or execution time depending on user requirements.

Keywords: Cloud Service Provider (CSP), Cloud Computing, Resource allocation.

1. INTRODUCTION

As the technology is changing rapidly, Cloud Computing is becoming more advance. Every passing moment is adding something in cloud. Because of this continuous advancement in cloud, it is hard to give a standard definition of cloud. Cloud computing is totally lays on two important concepts: one is Abstraction and the other one is Virtualization. Cloud abstracts internal details of cloud system implementation and location of datacenters from users and developers. Resources are virtualized to scale multiple users on a single resource[1].

Cloud computing is a fast growing area in which resource allocation is an important technology. Cloud has two distinct parties: service providers and users that have their own interests. Generally service providers provide a little information regarding workload characteristics and detailed resource states. They do not share the complete information regarding resource availability and how much load the resources have at any instant. As a result, Users are only able to make requests for resources based on the little information. In same manner, users do not share complete information of their applications and workload that are going to be executed on the virtual machines(VMs). Due to this, service providers place VMs in a random fashion, which in turn results in ineffective resource utilization and degraded performance.

Resource allocation is a way of assigning resources to the cloud user. Resource allocation mainly solution of some questions: what types of resources are? How many resources are? and when to provide these resources available for users?. The cloud user makes a request by deciding the type and quantity of resources. For efficient execution of application, resource type should be meeting the characteristics of workload and the quantity should be meeting the parameters like response time, deadline, etc. In Cloud computing, where resources are requested and returned by users dynamically, it's not sufficient to decide what type of resource should be allocated, decision making for when to provide resource also becomes important. Now-a-days there are numbers of options for the user for one type of requirement specification, but selection of appropriate CSP becomes difficult. So researchers have developed multiple techniques for appropriate resource allocation. These techniques try to satisfy user and optimize resource utilization.

2. RELATED WORK

A minimum cost maximum flow(MCMF) algorithm for dynamic resource allocation is provided [2]. This algorithm is introduced to overcome the problem of scalability in modified Bin- Packing algorithm. Resource allocation problem is represented in the form of directed graph. An autoregressive method is used for prior knowledge of varying resource demands. That helps in settings up pricing of resources dynamically, which avoids request rejection and enhance profits for resource providers. This algorithm forecasts new user requests and supports scalability.

A Greedy Particle Swarm Optimization(GPSO) is a searching method which is combination of two methods, one is greedy search method and other is particle swarm optimization. Actually virtual design advisor(VDA) works to overcome the problem of performance optimization of DBMS which is on VMs. That requires calibration of query optimizer's tuning parameters which evaluate the cost of database workloads running in virtual machines with varying resource allocation. A greedy heuristic method is adopted by VDA but that contains deception in local optima. GPSO algorithm prevents deception in local optima and provides cost minimization [3].

As we know providers always try to increase revenue but the same time demands fluctuates, it becomes essential to reconfigure VMs to suit the requests. To deal with it a market based dynamic resource allocation approach[4] is introduced. Prices can be adjusted according to market condition and demand pattern.

Resource allocation problem in Distributed cloud for context aware applications is a complicated task because there are many alternative computers with varying capacities. So, providing a suitable one, within time to charge profit to the providers become complicated. So a giFIFO scheduling algorithm dynamically allocates resources to all the context aware applications (i.e. each request that arrives at a datacenter contains a context id). Final allocation of endserver is based on compatibility check of context id and id of endserver. It minimizes response time and maximizes profit charged by the CSPs[5].

Cloud provide services to many users at the same time and different users have different quality of service (QoS) requirements, the scheduling strategy should be developed for multiple workflows with different QoS requirements. [6] So the Author defined a Multiple QoS Constrained Scheduling Strategy of Multi-Workflows (MQMW) to address the problem. The strategy can schedule multiple workflows which are started at any time and the QoS requirements are taken into account.

A cost and time optimization algorithm for allocating resources to service requests by considering multiple clouds, using the concept of RAINBOW framework in such a way that the user's requirements are met with minimum cost. Consumer can submit their additional requirements (i.e. cost, time, security etc) along with requests. These requirements are used in resource utilization. [7]

A SLA based resource allocation algorithm is introduced. It uses QoS parameters (such as service initialization time, arrival time and penalty time) from the prospective of users and providers that help in dealing with user requests. It minimizes SLA violations and infrastructure cost.[8]

Another dynamic resource allocation algorithm is using Haizea resource lease manager. It is based on deadline sensitive policy that reduces the number of request rejection by swapping and preemption of consecutive leases when no idle resource is there. It reduces cost also.[9]

A dynamic resource allocation method in self oriented cloud in which each request is in the form of lightweight query message. That makes resource searching easy. Optimal resource allocation is done using Karush Kuhn Tucker conditions under user budget. It maximizes resource utilization by combining with proportional share model. [10]

An agent based resource management and discovery strategy is proposed [11]. It is basedBloom filter. Bloom filter contains information of resources and it sends this information to related databases via related agents. When user requests arrive they go to broker agents and the required information for resource searching is there which reduces time for discovery and data transfer work.

Priority based VM allocation algorithm executes high priority job first. When a high priority job arrives and if no VM is free for allocation. It avoids creation of new VM by preempting low priority job and allocates that VM to high priority job [12].

In this research we are trying to improve resource allocation in user's point of view. User can elaborate its request by entering additional parameters. Users can be benefited by providing more efficient resources. In cloudSim no such type of resource allocation strategy is implemented yet.

3. PROPOSED ARCHITECTURE

We proposed a time and cost weightage based resource provisioning algorithm i.e. User Defined Time and Cost Constraints Based Resource allocation algorithm(TCBR) which decides the priority of allocation either on the basis of cost or on the basis of time. The time and cost constraints are submitted by user along with the request. It has different modules that are under three layers as shown in fig.1.

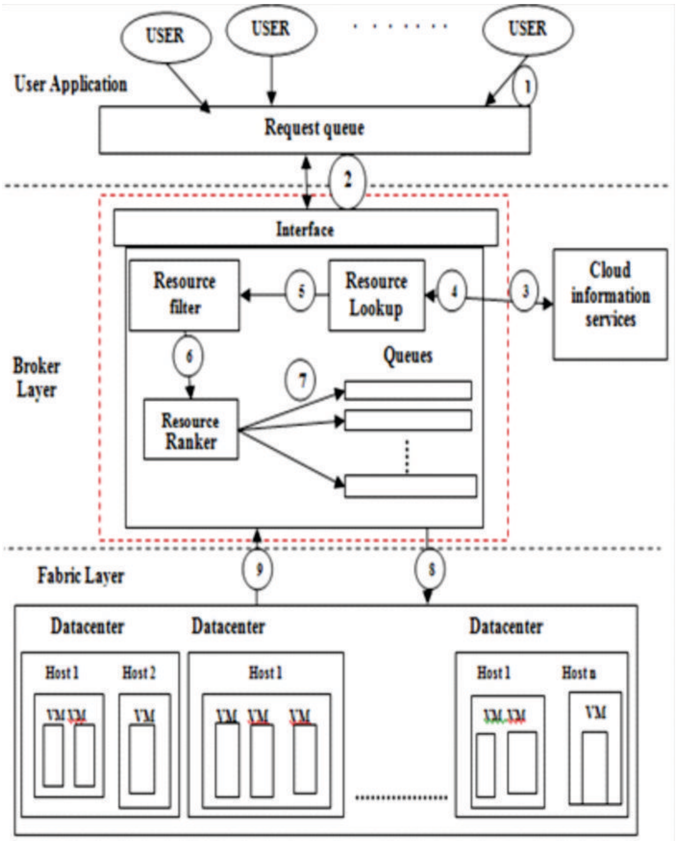

Fig.1 Architecture of TCBR Allocation

3.1 User Application Layer

User application layer or user interface is where cloud users interact with cloud. It is where a user access cloud environment by sending requests. User requires the accessing device and internet connectivity. Devices like mobile phone, laptop, workstations, pc, etc. are used for accessing the cloud. Cloud provides the facility of concurrent accessing; in other word multiple users can access the services of cloud at a time.

3.2 Broker Layer

On this layer a user request directly handled. It is responsible for resource provisioning because most of the resource allocation strategies are implemented on this layer. In cloud computing environment services are provided by CSPs. There are multiple service providers on the cloud who provide services on cloud. But the cloud users don't directly contact with the services providers. An intermediate always works between cloud users and service providers. This intermediate is called cloud resource broker. The broker's role may simply be to save user's precious time. The broker researches services from different CSPs and understands the work processes, provisioning needs, budgeting and data management requirements of users.

3.2.1 Cloud Information Services

A Cloud Information Service (CIS) is an entity that provides cloud resource registration, indexing and discovery services. Resource broker can contact this class for resource discovery, which returns a list of registered resource Ids.

3.2.2 Resource Lookup

It is an internal part of the broker, where it works for resource searching. It looks for the available resources when a user request arrives for this is contact to CIS. Each resource provider resister itself to provides resources as a service. Resource lookup is responsible for supporting resource provisioning in cloud. It passes the information of available resources to the resource filter.

3.2.3 Resource Filter

Resource filter is also a part of broker layer. It sorts the resource according to the requested constraints. It makes resource selection easy. It provides support to the resource ranker and reduces ranker's work.

3.2.4 Resource Ranker

Resource ranker is to provide ranking to the resources. It works as a selector to select an appropriate resource. For infrastructure level it tries to reduce infrastructure spend and improve maintains and scaling.

3.3 Fabrication Layer

It is where resources are resided. It contains like datacenter, hosts and VMs etc.

3.3.2 Datacenter

Datacenter is basically a large group of networked servers. A datacenter is responsible for storing and managing the data and information. It is simply a purpose built warehouse for large numbers of computers. Data center architectures and necessities can differ in a major way.

3.3.2 Host

Host is connected with datacenter. Host executes actions related to management of virtual machines (e.g., creation and destruction). A host has a defined policy for provisioning memory and bandwidth, as well as an allocation policy for Pe's to virtual machines.

3.3.3. Virtual Machines(VMs)

For cloud computing environment virtualization is a very important technology. Virtual machine is virtualization of a real physical machine. Virtual machines share physical machine means on a single computer system we can create multiple virtual machines. It enhances the capacity of a real computer. It provides same look and feel like a real machine.

Our proposed work is on broker layer, where functionality of this layer is enhanced by adding modules: resource lookup, resource filter and resource ranker.

Workflow of TCBR Architecture

1. The user entity creates a request that contains application description with QoS requirements, sends it to request queue.
2. User request goes to the broker from request queue.
3. Broker's Resource Lookup module interacts with the CIS entity to identify contact information of resources.
4. CIS provides a list of available resources to the Resource Lookup.
5. Resource lookup gives this list to the Resource filter.
6. Filtering of the resources is done on the basis of QoS requirements using User Defined Time and Cost Constraints Based Resource Allocation Algorithm.

   QoS constraints(w_t, w_c)

   Where w_t = time weightage factor

   w_c = cost weightage factor

   Estimated execution time (t) = Cloudlet length/MIPS of VM

   Estimated execution cost (c) = (subscriptioncost *t) + costperbw + costpermry +costperstorage
7. For each resource calculate

   Threshold value (TH) = wtt + wcc

   Min(TH)

   Resource Ranker selects a resource according to minimum thresholdand passes the request.
8. Then resource is assigned to request.
9. When the execution is completed, the resource will be free for other requests.

Fig.2 Workflow of TCBR architecture

4. EXPERIMENTAL SETUP AND RESULTS

This section evaluates the effectiveness of TCBR algorithm. In our experimental environment we are considering multiple VMs, multiple, user requests and a broker. Each user request(Cloudlet) enters its request with time weightage and cost weightage parameters. Here we are taking 6 user requests, whose requested time weightage and cost weightage parameters are shown in table 1.

Table 1 User's Requested QoS Parameters

User request (cloudlet)	Time weightage parameter (wt)	Cost weightage parameter(wc)
C0	0.8	0.2
C1	0.7	0.3
C2	0.4	0.6
C3	0.9	0.1
C4	0.2	0.8
C5	0.3	0.7

To execute these requests we are taking 10 VMs with processing speed varying from 10000 million instructions per second(MIPS) to 30000 MIPS and various costs as shown in Table II.

Table 2: VM configuration

VM (ID)	MIPS	Sub-scription cost ($)	Cost per Band-width($)	Cost per Memory ($)	Cost per Storage ($)
0	10000	0.015	0.012	0.02	0.001
1	20000	0.030	0.024	0.04	0.002
2	30000	0.045	0.036	0.06	0.003
3	15000	0.023	0.013	0.025	0.001
4	20000	0.030	0.025	0.042	0.002
5	30000	0.046	0.037	0.062	0.003
6	10000	0.011	0.010	0.015	0.00075
7	15000	0.016	0.020	0.030	0.0015
8	15000	0.018	0.025	0.037	0.0018
9	20000	0.022	0.025	0.043	0.0021

For each user request selection of VM is on the basis QoS parameters, if wt>0.5 than VMs are filtered on the basis of estimated execution time(t) otherwise VMs are filtered on the basis of estimated execution cost©.

$$t = Cloudlet\ length/MIPS\ of\ VM$$

$$c = (subscriptioncost * t) + costperbw + costpermry + costperstorage$$

Selection of one appropriate VM will be on the basis of threshold (TH) value. The VM with minimum threshold will be selected.

$$Threshold\ value\ (TH) = w_t t + w_c c$$

Table 3 Allocated VMs

Cloudlet	Cloudletlength (MI)	Allocated VM
C0	6516373	Vm2
C1	17838839	VM5
C2	13320993	VM9
C3	11481502	VM1
C4	21213274	VM7
C5	18307528	Vm4

We have studied a number of existing techniques of resource allocation in cloud. We will compare the existing resource allocation method of CloudSim with the proposed TCRB resource allocation model. In existing CloudSim user request is assigned first available VM but in our proposed algorithm user request is assigned best available VM.

First experiment result shows improvement of performance in response of execution time for proposed model over the existing model. It reduces the execution time of job by providing better resources or we can say fast processing resources.

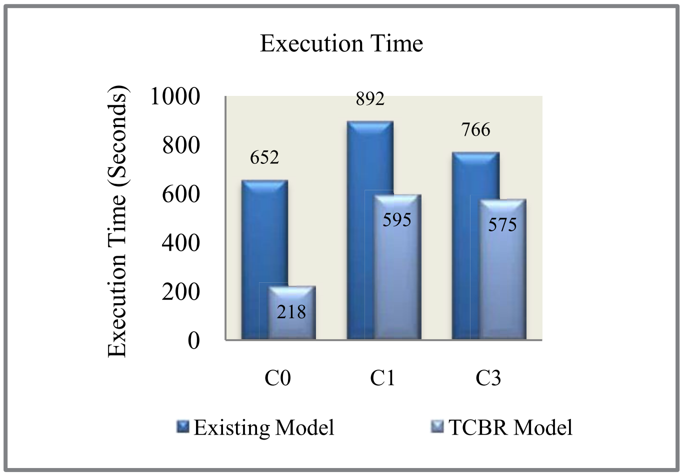

Fig. 3 Execution Time of Tasks for Existing and TCBR Model

The second part of results determines cost optimization over existing model. In this performance improvement is gained by reducing the execution cost.

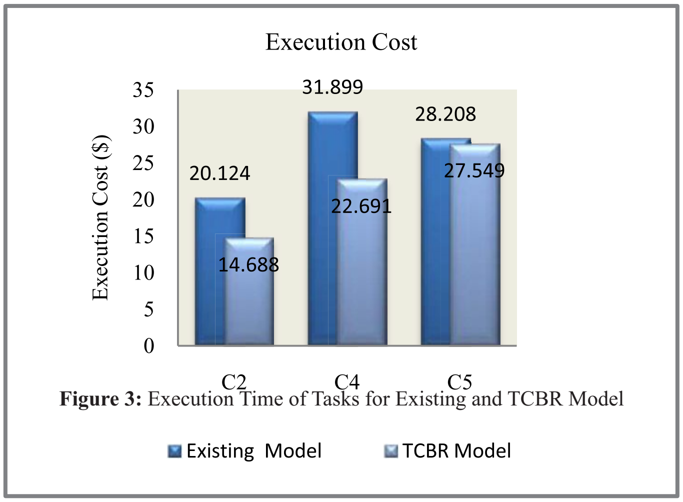

Figure 3: Execution Time of Tasks for Existing and TCBR Model

Fig. 4 Execution Cost of Tasks for Existing and TCBR Model

5. CONCLUSION AND FUTURE SCOPE

As a solution, we proposed a time and cost weight-age based resource allocation algorithm i.e. User Defined Time and Cost constraints based Resource Allocation algorithm. The time and cost constraints are submitted by user along with the request. This allocation method provides support for the resource provisioning in the cloud computing environment. The results shows that we are now able to provide better resource selection to cloud user as user now can specify its request with specific QoS constraints.

In future besides time and cost parameters we can add more functional and nonfunctional parameters that user can enter for increase interaction with provider like data security, reliability etc.

We can also provide estimated waiting time and the user can be intimated with the same.

REFERENCES

[1] Barrie Sosinsky, "Cloud Computing Bible", ISBN: 978-0-470-90356-8, Wiley publishing, Inc., Dec 2010.

[2] Makhlouf Hadji, Djamal Zeghlache, "Minimum Cost Maximum Flow Algorithm for Dynamic Resource Allocation in Clouds", IEEE5th International Conference on Cloud Computing, ISBN: 978-0-7695-4755-8, June 2012 , pp. 876-882.

[3] Radhya Sahal, Sherif M. Khattab, Fatma A. Omara, "GPSO: An Improved Search Algorithm for Resource Allocation in Cloud Databases", International Conference on Computer Systems and Applications(AICCSA), ISBN 978-1-4799-0792-2, May 2013, pp. 1-8.

[4] Qi Zhang, Eren Gurses, Raouf Boutaba, Jin Xiao "Dynamic Resource Allocation for Spot Markets in Cloud Computing Environments", Hot-ICE'11 Proceedings of the 11th USENIX conference on Hot topics in management of internet, cloud, and enterprise networks and services, March 2011.

[5] Keerthana Boloor, Rada Chirkova, Yannis Viniotis, Tiia Salo, "Dynamic request allocation and scheduling for context aware applications subject to a percentile response time SLA in a distributed cloud", 2nd IEEE International Conference onCloud Computing Technology and Science (CloudCom), ISBN: 978-0-7695-4302-4 , Dec. 2010, pp. 464-472.

[6] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing", IEEE International Symposium on Parallel and Distributed Processing with Applications, ISBN: 978-0-7695-3747-4, Aug 2009, pp. 629-634.

[7] Venkatarami Reddy Chintapalli, "A Deadline and Budget Constrained Cost and Time Optimization Algorithm for Cloud Computing", Advances in Computing and Communications, Communications in Computer and Information Science Volume 193, ISBN: 978-3-642-22726-4, 2011, pp 455-462.

[8] Linlin Wu, Saurabh Kumar Garg and Rajkumar Buyya, "SLA-based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments", 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2011, pp 195-204.

[9] Thangaraj P, Soundarrajan S, Mythili A, "Resource Allocation Policy for Infrastructure as a Service in Cloud Computing", International Journal of Computer Science and Management Research",, 2013, pp 1645-1649.

[10] Sheng Di and Cho-Li Wang, "Dynamic Optimization of Multiattribute Resource Allocation in SelfOrganizing Clouds", IEEE Transactions on Parallel and Distributed Systems, 2012,pp 464- 478,.

[11] Rojia Nikbazm, Ahmadi, M." Agentbased Resource Discovery in Cloud Computing using Bloom Filters", 4th IEEE International eConference on Computer and KnowledgeEngineering (ICCKE), ISBN 978-1-4799-5487-2, pp. 352-357, Oct 2014.

[12] Saraswathi AT, Kalaashri.Y.RA , Dr.S.Padmavathi , "Dynamic Resource Allocation Scheme in Cloud Computing" Procedia Computer Science Vol -47 , pp. 30 – 36, 2015.

◆ ◆ ◆