International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 2, No. 1, 2015, pp. 97-111.
ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

A mechanism to improve the throughput of cloud computing environments using congestion control

Mohammad Farajizadeh[a] , Naser NematBakhsh[b]

[a] *Department of Computer, Yasooj Branch, Islamic Azad University, Yasooj, Iran.*

[b] *Department of Computer, Yasooj Branch, Islamic Azad University, Yasooj, Iran.*

Abstract

Cloud computing as a new model for the delivery of applications, platforms or computing resources, including processing, memory, bandwidth, etc. to customers in return for the used payment method emerged. Therefore it has become important to share cloud resources, so as to provide them economically. In cloud computing services, multiple types of resources, such as processing ability, bandwidth and storage, need to be allocated simultaneously. If there is a surge of requests, a competition will arise between these requests for the use of cloud resources. This leads to the disruption of the service and it is necessary to consider a measure to avoid or relieve congestion of cloud computing environments. The basic issue here is to provide a new congestion control method for cloud computing environments which reduces the size of required resource for congested resource types instead of restricting all service requests as in Conventional methods. In this research, congestion control implemented in allocation level of available resources in the datacenters to virtual machines And we evaluated its impact on the cloudlets response time and throughput of cloud computing environments by comobining it with a new cloudlets allocation strategy. In order to simulate the proposed mechanism named CCMICC, CloudSim3.0.3 software has been used. It is demonstrated by simulation evaluations that CCMICC can allocate more resources to virtual machines compared with the conventional methods and relieve the congestion. the proposed method can improve the throughput of cloud computing environments as well as service quality.

Keywords: Cloud Computing, Resource Allocation, Congestion Control, throughput

Introduction:

Cloud computing is a computing method in which resources with dynamic development capability and mainly in virtual form are prepared and delivered on the internet by service providers [1]. Cloud computing has been created as a new model for delivery of applications, platforms, or computing resources such as processing ability, memory, band width, etc. to the customers in method of paying in return to using services [2]. When demands in a resource are more than its current capacity, the congestion occurs [1]. Therefore, congestion control is related to resource allocation in a network. So that the network can operate at an acceptable level of efficiency when demand exceeds the capacity of available resources [2]. Since, Applying the appropriate algorithms and mechanisms could lead to optimum allocation of resources and consequently to service quality promotion And considering that, perfect recognition of the operational environment and precise model designed for that environment can help apply an appropriate mechanism. As a result, in this section we first introduce the matter, research necessity, research background, and research goals. Then, in section 2, the model of cloud resources allocation will be explained. In section 3, the method to decrease the amount of requested resources will be discussed in congestion situations. Section 4 is devoted to the introduction of cloudlets allocation strategy and the proposed mechanism is introduced in section 5. Simulation details are discussed in section 6. Simulation results are investigated and analyzed in section 7 and finally conclusion is provided in section 8.

1-1- *Problem Statement*

As cloud computing services rapidly expand their customer base, it has become important to share cloud resources, so as to provide them economically. In cloud computing services, multiple types of resources, such as processing ability, bandwidth and storage, need to be allocated simultaneously. If there is a surge of requests, a competition will arise between these requests for the use of cloud resources. This leads to the disruption of the service and it is necessary to consider a measure to avoid or relieve congestion of cloud computing environments. The basic issue here is to provide a new congestion control method for cloud computing environments which reduces the size of required resource for congested resource type instead of restricting all service requests as in Conventional methods. In a method that provided in the past the effect of reducing the size of the requested resources on the congested resources type is not specified and also The congestion control happen at cloudlets to virtual machines (VMs) allocation level But we want to implement congestion control in allocation level of existing resources in the datacenters to virtual machines. then combine it with a cloudlets allocation strategy and evaluate its impact on the cloudlets response time and throughput of cloud computing environments and finally improve the effectiveness of previous methods.

1-2- *The importance and necessity of research*

The recent research and experiments has demonstrated that the sharing of cloud resources Can be considered as an important factor in congestion control and new congestion control methods based on the resource allocation for cloud computing environments can handle more requests compared with the conventional methods and relieve the congestion. Therefore according to the existing research and experiments and also due to the weakness in this area is expected to solve the problem of resource allocation in order to congestion control in cloud computing environments is a topic of discussion. And network congestion control for all types of media as one of the most active areas of research in the past decade has been proposed.

1-3- *literature and research background*

Shin-ichi Kuribayashi in 2011 proposed a new congestion control method for cloud computing environments which reduced the size of required resources for congested resource type instead of

restricting all service requests as in the existing methods. In this method multiple types of resources, such as processing ability and bandwidth, allocated to requests simultaneously. Kuribayashi proposed the user service specifications for the proposed congestion control method, and clarified the algorithm to decide the optimal size of required resource to be reduced, based on the load offered to the system. It is demonstrated by simulation evaluations that the proposed method can handle more requests compared with the conventional methods and relieve the congestion [3]. The problem of this method was that implemented for the allocation of virtual machines to the cloudlets And by reducing the size of requests resulting in increased response time of cloudlets, the completion time of cloud services and finally reduced throughput.

Another study by Sourav Banerjee et al. in 2015, introduced a new cloudlet allocation policy with suitable load balancing technique that helps in distributing the cloudlets to the virtual machines (VMs) equally likely to their capacity which makes the system more active, alive, and balanced. This reduces the completion time of the cloudlet(s) as well as reduces the makespan of the VM(s) and the host(s) of a datacenter. Eventually, this proposed work improves the QoS[*]. The experimental results obtained using CloudSim 3.0.3 toolkit extending few base classes are compared and analyzed with several existing allocation policies [4].

1-4- *Research Objectives*

The main goals of this research are to decrease the loss probability for resource allocating to the virtual machines, reduces the response time of the cloudlet(s) and finally to increase throughput of cloud computing environments and resource performance In terms of congestion with the following suppositions.

Suppositions:

- the physical facilities for providing cloud computing services are distributed over multiple datacenters
- the services are non-delay
- multiple types of resources, allocate to requests simultaneously.

1-5- *Definition of technical terms*

Cloud computing: cloud computing has emerged as a new sample to deliver applications, platforms, or computing resources (including processing ability, memory, band width, etc.) to customers in a payment for use manner. The model is highly reasonable, Because the customer pays for his actual use without tolerating extra expenses. Furthermore, this model is extremely flexible as a pack can be used based on the customer's need.

Virtual machine: Based on receiving requests to system, virtual machines can be dynamically applied or stop working on a physical machine. These virtual machines are able to simultaneously run the applications on the basis of different operating systems on a machine or the physical host [5].

Host: The hosts or physical machines, in fact, provides hardware infrastructure for creating and running virtual machines [5].

Virtual machine placement: The decision to place a particular VM to a particular host is known as VM placement. Key challenge here is to maximize the number of cohosted VMs while still maintaining the required performance level [6].

Resource Allocation: allocating resources in cloud environments means finding an appropriate physical host to virtual machines placement which include users requests. Resource managing is a duty of cloud providers. Cloud consumers' requests for service are sent to the cloud from everywhere in the world [7].

---

[*] Quality Of Service

Congestion: Internet "Congestion" occurs when the aggregate demand for a resource exceeds the available capacity of the resource. Resulting effects from such congestion include long delays in data delivery, wasted resources due to lost or dropped packets [8].
Congestion Control: Congestion control is concerned with allocating the resources in a network such that the network can operate at an acceptable performance level when the demand exceeds or is near the capacity of the network resources. Although resource allocation is necessary even at low load, the problem becomes more important as the load increases because the issues of fairness and low overhead become increasingly important. Without proper congestion control mechanisms, the throughput may be reduced considerably under heavy load [9].

2- Cloud resource allocation model

In this Study, the resource allocation model that Proposed in [3] and [10] is used. This resource allocation model for a cloud computing environment is such that multiple resources taken from a common resource pool are allocated simultaneously to each request for a certain period. we consider two resource types: processing ability and bandwidth, for the preliminary evaluation. It is assumed that the physical facilities for providing cloud computing services are distributed over multiple centers.
The cloud resource allocation model that incorporates these assumptions is illustrated in Figure 1 Each center has servers (including virtual servers), which provide processing ability, and network devices which provide the bandwidth to access the servers. The maximum size of processing ability and bandwidth at center j (j=1,2,..,k) is assumed to be Cmaxj and Nmaxj respectively. When a service request is generated, one optimal center is selected from among k centers, and the processing ability and bandwidth in that center are allocated simultaneously to the request for a certain period. If no center has sufficient resources for a new request, the request is rejected [3].

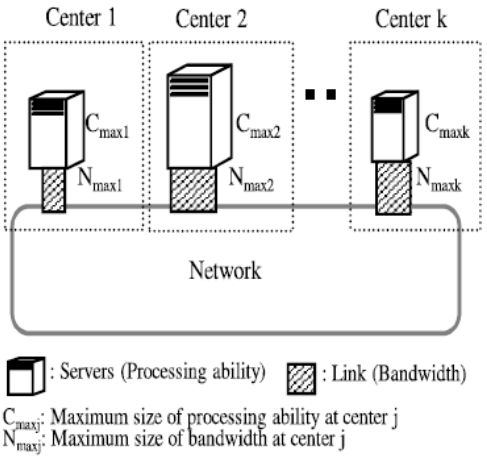


Figure (1) resources allocation model for cloud computing services

3- The method used to control congestion

In this study, we used congestion control idea Proposed in [3] which reduces the size of required resource for congested resource type, instead of restricting all service requests when a specific types of resource is congested. The aim of this method to reduce the requests loss probability and to increase the resources performance [3].

3-1- ***User service parameters:***

C_r: Size of required processing ability
C_v: Threshold value for processing ability
N_r : Size of required bandwidth
N_v : Threshold value for bandwidth
C_0 : Acceptable minimum size of processing ability (=q* Cr)
N_0 : Acceptable minimum size of bandwidth (=q* Nr)
C_{r0}: Finally determined size of processing ability ($C_0 \leq C_{r0} \leq C_r$)
N_{r0}: Finally determined size of bandwidth ($C_0 \leq C_{r0} \leq C_r$)
Q : Maximum reduction coefficient ($0<q\leq 1.0$)
This coefficient indicates how much reduction in the size of required resource can be accepted when its resource type is congested. The value of q should be the same for both processing ability and bandwidth [3].

3-2- ***procedure flow to decide the values of Cr0, Nr0 in the case where only processing ability is congested:***

Figure 2 show the procedure flow to decide the values of Cr0, Nr0 in the case where only processing ability is congested.

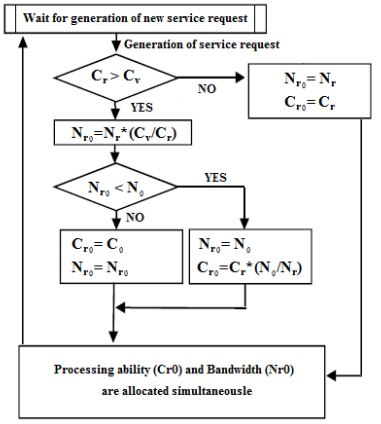


Figure (2) procedure flow to determind the values of Cr0, Nr0 in the case where only processing ability is congested.

3-3- ***procedure flow to decide the values of Cr0, Nr0 in the case where only band width is congested:***

Figure 3 show the procedure flow to decide the values of Cr0, Nr0 in the case where only band width is congested.

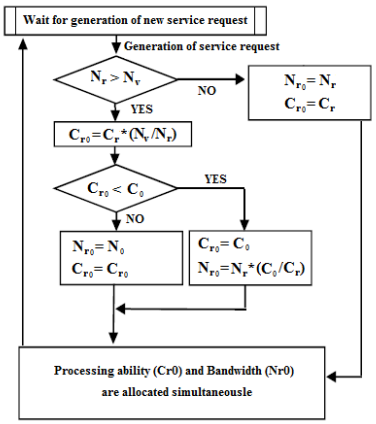


Figure (3) procedure flow to determind the values of Cr0, Nr0 in the case where only bandwidth is congested.

3-4- *procedure flow to decide the values of Cr0, Nr0 in the case both processing ability and band width is congested:*

Figure 4 show the procedure flow to decide the values of Cr0, Nr0 in the case both processing ability and band width is congested.

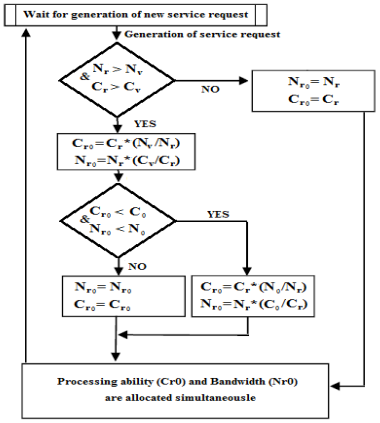


Figure (4) Procedure flow to determind the values of Cr0, Nr0 in the case where both processing ability and bandwidth are congested.

4- CloudLets Allocation Strategy:

In the provided mechanism, has been used the cloudlets allocation strategy proposed in[4] To improve the quality of service that we will continue to explain it.

4-1- *Cloud Resource Allocation Policy:*

In this allocation policy, all the VMs are sorted in descending order according to their MIPS (Million Instruction per Second) and stored in a list. Then, calculate the maximum load capacity (MLC) of each VM by using the Eq.(1).

$MLC_j(\%) = (MIPS_j / \text{Total MIPS of all VMs}) * 100$ (1)

The batches of cloudlets are stored in a list. The size of each cloudlet (cl_size) is calculated in term of percentage using Eq.(2).

$$cl_size_i(\%)=(MI_i / \text{Total MI of all Cloudlets}) * 100 \quad (2)$$

The cloudlets are sorted into descending order according to their Million Instruction (MI) value and placed them in a list. Then, the cloudlets are assigned to the VMs guided by the First Come First Serve (FCFS) policy following the process in which the highest capacity VM will be allocated first, and after allocation, the remaining load capacity (RLC) of that VM will be compared with the MLC of the other VMs in the vm_list by using the Eq.(3).

$$RLC_j(\%)=MLC_j-cl_size_i \quad (3)$$

4-2- *Important Parameters of Cloudlet allocation strategy:*

Execution time (Exec): Execution time is defined as the ratio of MI (Million Instruction) of the cloudlet to the MIPS (Million Instruction Per Second) of the allotted VM. The calculation of the execution time of a cloudlet is mentioned in the Eq. (4).

$$Exec_i=MI \text{ of } Cloudlet_i / MIPS \text{ of } VM_j \quad (4)$$

Starting time (ST): Starting time is defined as the completion time of the previous cloudlets in the same VM. Whenever a cloudlet is assigned to a VM for the first time then its starting time is assumed to be zero. When the other cloudlets are assigned to that VM the completion time of the previous cloudlet on that VM will be treated as the starting time of next assigned cloudlet.

Completion time (CT): It is defined as the sum of the execution time and the completion time of the previous cloudlet allocated in the same VM. The calculation of the completion time of a cloudlet is mentioned in the Eq. (5).

$$CT_i=Exec_i + CT_{i-1} \quad (5)$$

Makespan of VM (MSVM): makespan of VM is used to measure the efficiency of a VM. The calculation of the makespan of a VM is mentioned in the Eq. (6).

$$MSVM_j= CT \text{ of Last Allocated Cloudlet into } VM_j \quad (6)$$

Makespan of Host (MSH): The makespan of a host is calculated as the maximum makespan of the VM from the allotted VMs in that host. The calculation of the makespan of a VM is mentioned in the Eq. (7).

$$MSHj= \max(MSVM_j) \quad (7)$$

4-3- *Pseudo code of cloudlets allocation mechanism:*

This section presents the pseudo code of cloudlets allocation algorithm introduced in [4] that was discussed in previous section.

Load Balancing Algorithm

Input : n Number of VMs presented in the vm_List
m Number of Cloudlets presented in the cloudlet_List
Output: Allocate the m Number of Cloudlets to n Number of VMs

1. Initialy allocate the VMs into the Host in the DC and sort them into ascending order according to their MIPS value. Sort the VMs in vm_List.
2. Calculate Maximum Load Capacity (MLC) of each VM present in vm_List.
3. The Batch of Cloudlets is Stored in cloudlet_List.

4. Calculate the Size of Each Cloudlet (cl_size).
5. Sort the Cloudlets in Descending order According to the Size of Cloudlets.
6. Allocate the Cloudlets to the VMs According to the following steps
7. For m Number of Cloudlet and n Number of VMs
   7.1. Find the Maximum Load Capacity (MLC) VM from the VMs present in the vm_List.
   7.2. Allocate the Cloudlet to the Corresponding VM which has Maximum MLC value.
   7.3. Decreased the MLC value of the VM by the size of the assigned Cloudlet (cl_size) into the particular VM.
   7.4. Update the MLC value of the VM according to the value of RLC.
8. Return the step 7 Until all Cloudlets Present in cloud_list are Allocated to their Corresponding VM.
9. Calculate Starting Time, Execution Time and Completion Time for Each Cloudlet.
10. Find the MakeSpan of N Number of VMs (MSVM).
11. Find the MakeSpan of the Host with N Number of VMs

5- Pseudo-code of CCMICC mechanism:

This section presents the pseudo-code of CCMICC mechanism. This combined mechanism is a mechanism which consists of two phases. In the first phase apply allocation of virtual machines and congestion control. In the second phase apply allocation of cloudlets to virtual machines.

Congestion Control Mechanism In Cloud Computing (CCMICC)

Input : RequestedVM_List , DC_List And Cloudlet_List
Output : Allocate the VMs To DCs and Allocate the VMs To Cloudlets

1. Get RequestedVMs_List And DC_List And Cloudlet_List
2. Select a DC with the most available resources from DC_List
3. Send RequestedVMs_List To SelectedDC For Creating and Resource Allocating
4. If (Some RequestedVMs Not Created Due To Lack Of Resources) Then
   4.1. If (DC_List Is Empty) Then
      4.1.1. Make RemainingDC_List Of DataCenter That Have The Available Resources
      4.1.2. Sort RemainingDC_List Order By Descending
      4.1.3. Select a DC with the most available resources From RemainingDC_List
      4.1.4. Send Congested VMs To Selected DC For Resource Allocating By Presented Congestion Control Mechanism that presented In Section 3
      4.1.5. If (Some VMs Not Created And RemainingDC_List Not Empty)
         4.1.5.1.1. Remove Selected DC From RemainingDC_List
         4.1.5.1.2. Repeat From 4.1.3
   4.2. Else
      4.2.1. Remove Selected DC with the most available resources From DC_List
      4.2.2. Repeat From 1
5. If (VMCreated_List Not Empty) Then
   5.1. Allocate All Cloudlets To Created VMs By Load Balancing Algorithm that presented in Section 4

6- Simulation

present study, Simulated by CloudSim3.0.3 through 3 scenarios. In the first scenario a situation is considered that there is no congestion in datacenters and requests are able to obtain all required resources. In the second scenario assumed that a situation in which required resources are congested and no congestion controlling is performed. In this case, congested requests are losted and some virtual machines

are not created. in the third scenario requests are encountered with lack of required resources as well as congestion that via CCMICC mechanism is controlled.

6-1- *General conditions of simulation*

In this experiments, we are considered ten Cloudlets and three virtual machines in a host. Properties of the cloudlets and virtual machines are considered as showed in the following tables. Moreover, It is assumed that there is only one datacenter with only a host. In this study, decreasing coefficient (q) is assumed to 0.7 so that when the creation requests of virtual machines in datacenters are congested, the requested amount of congested resource is 30% decreased and then, if possible, is created in appropriate datacenter and required resources are allocated to the virtual machine. This resource decrease occurs in conditions that due to lack of required resources the request must have been lost, and this strategy is a method for avoid the loss of the creation of a requested virtual machine and infact to request of resources allocation in the proposed mechanism. The amount of q can be a number between 0 and 1($0 < q \leq 1$). when q is 1, it means that reduce the size of congested resource is zero percent and virtual machine creation request do not experience any decrease in size. Table 1 Shows virtual machines with their processing ability by million instructions per second, band width size, their memory capacity, the number of processors, and scheduling policy of cloudlets that are sorted by MIPS in ascending order.

Table 1- Properties of VMs

VM_id	MIPS	BW	RAM	PEs Number	Cloudlet Scheduler
0	300	1000	250	1	Space Shared
1	600	15000	500	1	Space Shared
2	400	2000	1000	1	Space Shared

Table 2 represents the maximum load capacity of each VM, calculated in percentage and stored in decreasing order according to their MIPS.

Table 2- Maximum load capacity of each VM

VM_id	MIPS	Capacity Of VMs [(MIPS/Total MIPS Of All VMs)×100] (%)
1	600	46
2	400	31
0	300	23

Table 3 represents ten cloudlets that sorted and stored in decreasing order according to their sizes (MI).

Table 3- Size (%) of each sorted cloudlet

Cld_id	MI	Capacity Of Cloudlets [(MI/Total MI Of All VMs)×100] (%)
7	17000	15
1	16000	14
4	15000	14
6	12000	11
2	11000	10
8	10000	9

0	9000	8
5	8000	7
9	7000	6
3	6000	5

The cloudlets are allocated into the VMs in FCFS manner and follow this proposed policy. Each of the cloudlet will be assigned to the VM whose MLC is maximum. After allocation, the remaining load capacity (RLC) of VMs will be calculated using Eq. (3).

6-2- *Senario 1*

In this experiment we considered the properties of the host so that the size of the host processing ability considered to 4000 mips and placement of virtual machines in the datacenter is not congested. Table 4 Shows a host with their processing ability in million instructions per second, band width size, their memory capacity and scheduling policy of VMs.

Table 4- properties of the host in a situation that VMs are not congested

HOST_id	MIPS	BW	RAM	VM Scheduler
0	4000	20000	2048	Time Shared

Table 5 shows the final list of cloudlets that allocated to virtual machines in a situation that Virtual machines are not congested.

Table 5- final list of cloudlets allocated to VMs in a situation that VMs are not congested

Cld_id	Cl_Size(%)	Vm_id	MLC(%)	RLC(%)
7	15	1	46	31
1	14	1	31	17
4	14	2	31	17
6	11	0	23	12
2	10	1	17	7
8	9	2	17	8
0	8	0	12	4
5	7	2	8	1
9	6	1	7	1
3	5	0	4	-1

Table 6 shows execution, Completion and start time of cloudlets that allocated to virtual machines in a situation that Virtual machines are not congested.

Table 6- execution, Completion and start time of cloudlets allocated to VMs in a situation that VMs are not congested

Cld_id	MI	Vm_id	MIPS	Exec	ST	CT
0	9000	0	300	30	0	30
1	16000	1	600	27	0	27
2	11000	1	600	18	27	45
3	6000	0	300	20	30	50
4	15000	2	400	38	0	38

5	8000	2	400	20	38	58
6	12000	0	300	40	50	90
7	17000	1	600	28	45	73
8	10000	2	400	25	58	83
9	7000	1	600	12	73	85

Table 7 shows makespans of three virtual machines so that the placement of virtual machines in the host is not congested.

Table 7- makespans of three virtual machines in a situation that VMs are not congested

VM_id	MSVM(MakeSpan Of VMs)
0	90
1	85
2	83

Table 8 shows makespan of only host of datacenter in a situation that Virtual machines are not congested.

Table 8- makespans of host in a situation that congestion is not occurred

HOST_id	MSH(MakeSpan Of Host)
0	90

6-3- *Senario 2*

In this experiment, We have considered host properties such that virtual machines placement in datacenter has encountered with congestion and lack of resources. In this case congestion is occurred but congestion is not controlled. Table 9 Shows a host with their processing ability in million instructions per second, band width size, their memory capacity and scheduling policy of VMs.

Table 9- properties of the host in a situation that congestion is not controlled

HOST_id	MIPS	BW	RAM	VM Scheduler
0	1250	20000	2048	Time Shared

Table 10 shows the final list of cloudlets that allocated to virtual machines in a situation that virtual machines placement in datacenter has encountered with lack of resources and creation of Virtual machines is congested.

Table 10- final list of cloudlets allocated to VMs in a situation that congestion is not controlled

Cld_id	Cl_Size(%)	Vm_id	MLC(%)	RLC(%)
7	15	1	60	45
1	14	1	45	31
4	14	2	40	26
6	11	1	31	20
2	10	2	26	16
8	9	1	20	11
0	8	2	16	8
5	7	1	11	4
9	6	2	8	2
3	5	1	4	-1

Table 11 shows execution, completion and start time of cloudlets that allocated to virtual machines in a situation that virtual machines placement in datacenter has encountered with congestion and lack of resources.

Table 11- execution, Completion and start time of cloudlets allocated to VMs in a situation that congestion is not controlled

Cld_id	MI	Vm_id	MIPS	Exec	ST	CT
7	17000	1	600	28	0	28
4	15000	2	400	38	0	38
1	16000	1	600	27	28	55
2	11000	2	400	28	38	65
6	12000	1	600	20	55	75
0	9000	2	400	23	65	88
8	10000	1	600	17	75	92
9	7000	2	400	18	88	105
5	8000	1	600	13	92	105
3	6000	1	600	10	105	115

Table 12 shows makespans of three virtual machines so that the placement of virtual machines in the host is congested and congestion is not csontrolled.

Table 12- makespans of three virtual machines in a situation that congestion is not controlled

VM_id	MSVM(MakeSpan Of VMs)
0	The virtual machine because of congestion is not created
1	115
2	105

Table 13 shows makespan of only host in datacenter when virtual machines placement in datacenter has encountered with lack of resources and congestion is not controlled.

Table 13- makespans of host in a situation that congestion is not controlled

HOST_id	MSH(MakeSpan Of Host)
0	115

6-4- *Senario 3*

In this experiment, We have considered host properties such that virtual machines placement in datacenter has encountered with lack of resources and congestion. In this case that congestion is occurred, the size of the host processing ability considered 1250 and decreasing coefficient (q) is assumed 0.7. This means that the requested size of congested resources as much as 30 percent is reduced. Then, if possible, the allocation of resources applied. In this experiment, congestion is controlled by using CCIMCC mechanism. Table 14 Shows a host with their processing ability in million instructions per second, band width size, their memory capacity and scheduling policy of VMs.

Table 14- properties of the host in a situation that VMs are congested

HOST_id	MIPS	BW	RAM	VM Scheduler
0	1250	20000	2048	Time Shared

Table 15 shows the final list of cloudlets allocated to virtual machines in a situation that virtual machines placement in datacenter has encountered with lack of resources and congestion is controlled by using CCIMCC mechanism**.**

Table 15- final list of cloudlets allocated to VMs in a situation that congestion is controlled

Cld_id	Cl_Size(%)	Vm_id	MLC(%)	RLC(%)
7	15	1	50	35
1	14	1	35	21
4	14	2	33	19
6	11	1	21	10
2	10	2	19	9
8	9	0	17	8
0	8	1	10	2
5	7	2	9	2
9	6	0	8	2
3	5	2	2	-3

Table 16 shows execution, Completion and start time of cloudlets allocated to virtual machines in a situation that virtual machines placement in datacenter has encountered with lack of resources and congestion is controlled by using CCIMCC mechanism.

Table 16- execution, Completion and start time of cloudlets allocated to VMs in a situation that congestion is controlled

Cld_id	MI	Vm_id	MIPS	Exec	ST	CT
7	17000	1	600	28	0	28
4	15000	2	400	38	0	38
8	10000	0	210	48	0	48
1	16000	1	600	27	28	55
2	11000	2	400	28	38	65
6	12000	1	600	20	55	75
9	7000	0	210	33	48	81
5	8000	2	400	20	65	85
0	9000	1	600	15	75	90
3	6000	2	400	15	58	100

Table 17 shows makespans of three virtual machines so that the placement of virtual machines in the host is congested and congestion is controlled by using CCIMCC mechanism.

Table 17- makespans of three virtual machines in a situation that VMs are congested

VM_id	MSVM(MakeSpan Of VMs)
0	81
1	90
2	100

Table 18 shows makespan of only host in datacenter when congestion is controlled by using CCIMCC mechanism.

Table 18- makespans of host in a situation that congestion is occurred

HOST_id	MSH(MakeSpan Of Host)
0	100

7- Results Discussion

Figure 5 and table 19 represent the comparison of virtual machines Throughput and Table 20 shows throughput of host in situations that congestion occurred and is controlled with q=0.7 in compared with congestion happened but not controlled, i.e. q=1.

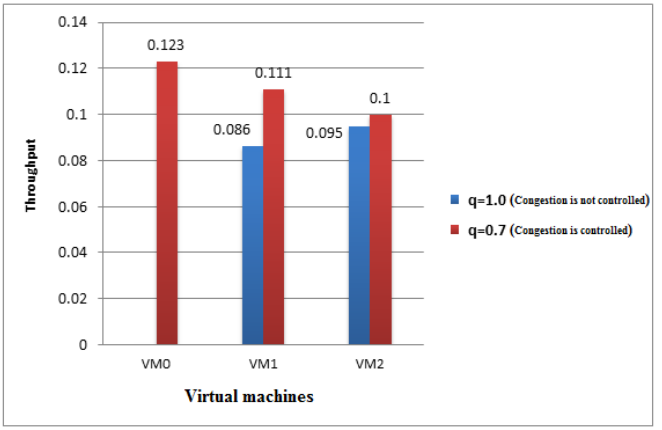


Figure (5) comparison of virtual machines Throughput between congestion controlled and non-controlled states

Table 19- comparison of virtual machines Throughput between congestion controlled and non-controlled states

VM_id	Proposed Strategy in [4]	CCMICC
0	0	0.123
1	0.086	0.111
2	0.095	0.100

Table 20- comparison of virtual machines Throughput between congestion controlled and non-controlled states

HOST_id	Proposed Strategy in [4]	CCMICC	ROI_ Throughput(%)
0	0.095	0.123	29.5

8- Conclusions

This study using the idea introduced in [3] in order to reduce the requests loss and combining it with load balancing technique introduced in [4], provided a mechanism to improve the load balancing method in [4] as much as possible. It has also increased system performance. In this study the idea of reducing resources size is used to allocate resources to virtual machines. The evaluations of simulation results have demonstrated that CCMICC mechanism is able to allocate more resources in congestion situations and control the congestion through creating more virtual machines by means of decrease the size of requested resources. It has been demonstrated by simulation evaluations that CCMICC can handle more cloudlets compared with the cloudlets allocation approach, which is introduced in [4] and can relieve the congestion. As a result, Throughput of cloud datacenters is improved and finally quality of services in cloud computing environments is increased. As the simulation results clearly illustrate CCMICC mechanism in compared with cloudlets allocation approach, which is introduced in [4], increases 29.5 percent the Throughput of datacenter.

In the future, we will evaluate the impacts of the number of users, the number of datacenters and The fairness in the resources allocation on the performance of the CCMICC mechanism.

References:

Ankur-Kumar, Sharma, 2014. Vplacement: Congestion Aware Virtual Machine Placement System, Unpublished master's Thesis, Carnegie Mellon University, Pittsburgh, America.

Banerjee, S., Adhikari, M., Kar, S. and Biswas, U., 2015. Development and Analysis Of a New Cloudlet Allocation Strategy For QOS Improvement in Cloud, Arab Journal of Science and Engineering, 40(5):1409-1425.

Ebrahimi, F., Farahi, A. and Farhodinejad, A., 2012. An overview of the resource allocation methods and its importance in a cloud computing environment, The first Payam Noor University national conference on information technology and computer networks, Payam Noor university of Tabas, Iran, February 14,2012.

Farhadi, A. and yazdian-verjani, A., 2013. Intelligent placement of virtual machines in the cloud-based datacenters using a grouping genetic algorithm, 11th Iranian Conference on Intelligent Systems, University of Kharazmi, Iran, February 27th & 28th, 2013.

Jain, Raj, 1990. Congestion Control in Computer Networks: ISSUES AND TRENDS, IEEE Network Magazine, 4(3):24-30.

Kuribayashi, Shin-Ichi, 2011. Proposed Congestion Control Method for Cloud Computting Environments, Journal of Computer Networks and Communication, 3(5):161-176.

Kuribayashi, Shin-Ichi, 2013. Joint Multiple Resource Allocation Method for Cloud Computing Services with Different QOS to Users at Multiple Locations, International Journal of Computer Network and Communications, 5(5):1-5.

Motavaselalhagh, F. and Safi-Esfahani, F., 2013. Provide an efficient algorithm to allocate resources based on service level agreements in cloud computing environment, The Fifth Iranian International Conference on Electrical and Electronic Engineering, Islamic Azad university of Gonabad, Iran, August 20th & 21th & 22th, 2013.

Pirhadi, M. and Godarzi, R., 2012. Cloud computing: architecture, advantages, challenges and propose solutions, The first national conference on information technology and computer networks, Islamic Azad university of Tabas, Iran, February 14, 2012.

Yaghmaee-Moghaddam, M.H., 2010. A Fuzzy Active Queue Management Mechanism for Internet Congestion Control, Third International Workshop on Advanced Computational Intelligence, Suzhou, China, August 25th & 27th, 2010.