International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Fog Node Allocation with Contigent Round Robin Algorithm

Dr. V. Shoba [1], *Dr. D. Bhuvaneswari* [2]

Assistant Professor, Department of Computer Science, SRM Arts and Science College

Abstract

One of the main components of cloud computing is data centers. The most challenging issues related to data centers is high energy consumption. The most important energy wasters in the cloud data centers is idle hosts. They use hardware resources as busy hosts and have low utilization. It means the hosts consume a large amount of energy when being at light load or even idle, compared with the time of having maximum use of their resources.

Introduction

The cloud data centres reduce energy consumption by using virtualization techniques. This technique enables multiple virtual machines (VMs) located on a single physical machine or host. So, the performance of the cloud data centres can be improved and the number of hardware resources can be possibly reduced. Live-migration is another benefit of virtualization that enables easy transferring of a VM from a host to another. It transfers running VMs from one host to another without interrupting the operation of the VMs. The VMs migrate to another host, when the number of VMs on a specific host is decreasing whereas the idle host goes to sleep or shutdown mode. This technique decreases the number of active hosts called "host consolidation."

Host consolidation includes 4 stages:

(1) detecting overloaded hosts from which some VMs should be migrated.

(2) selecting the VMs that should be migrated from an overloaded host.

(3) detecting under-loaded hosts which should migrate all their VMs and

(4) placing VM on destination host for migrating VMs.

Host consolidation consist of M is the number of available physical machines, and capacity of their resources such as memory, CPU, and bandwidth of the network is specified. N is the number of VMs whose needs should be met by memory, CPU, and bandwidth capacity. A good VM to host mapping is placing VMs on suitable hosts so that minimizing the number of the used physical machines. Also, total needs of resources for the located VMs in the host should not exceed the capacity of the physical machine. . In fact, the placement of a VM means finding a suitable physical machine for the VM regarding the minimum number of the hosts. Also, the service providers should be able to provide high-quality services to increase utilization and decrease energy consumption.

Ensuring the quality of services that have been defined in the service level agreement) is one of the important necessities for cloud computing environment. It is possible to save energy consumption substantially using placement techniques and create a trade-off among increasing utilization, decreasing energy consumption, and SLA. The VM placement problem is similar to bin packing program which is one of the NP-hard problems; hence, to solve the corresponding optimization problem, approximation algorithms are typically used. Energy consumption is reduced in the cloud data centres by dynamic placement of VMs. The employed policies in the proposed approach select a host with minimum energy usage to decrease both energy consumption and Service Level Agreement Violation . The proposed approach considers to the minimum correlation coefficients between the selected VM and the VMs running on the host, future load of the VMs. To predict the load of very next future of VMs to make decisions, an ensemble prediction algorithm is used. T

The proposed approach handles heterogeneous VMs effectively and does not require any information about the applications running on VMs. The main contributions of this research can be summarized as follows:

- We introduce the proposed framework for VM placement in order to reduce energy consumption in cloud data centres.
- We proposed an algorithm for VM placement in cloud data centres in order to reduce energy and SLAV using a novel algorithm with a mixture of approaches such as learning automata theory, ensemble prediction algorithm, and correlation.
- We consider minimum correlation coefficients between the current VM and VMs of the destination host and energy consumption of destination host after allocation, at the time of placement.

Allocation of Load to Fog Servers

User base properties

Number of UB's	Regions	Requests user/h	Request size (KB)	Peak hours start GMT	Peak End GMT	Avg. peak users	Avg. off peak users
1	0	60	20	3	9	1000	100
3	2, 4, 5	60	200, 300, 400	3	9	1000	100
3	1, 2, 5	60, 70, 80	100, 100, 100	3	9	1000	100
2	0, 3	60	200, 900	3	9	1000	50
4	1, 3, 2, 5	60	1, 100, 20, 100	3	9	1000	100

DC configurations values

Datacenter	No. of VMs	Image size	Memory	BW
DC1	10	10,000	512	1000
DC2	50	10,000	512	1000
DC3	80	10,000	512	1000
DC4	80	10,000	512	1000
DC5	10	10,000	512	1000
DC6	50	10,000	512	1000

Load balancing and grouping factor configuration

Grouping factors	Value/details
User grouping factors in userbases	500
Request grouping factors in datacenters	250
Executable instruction length per request (bytes)	5000
Load balancing policy across VMs in a single datacenter	contingent Round robin

Performance Metrics

- **Arrival Time-** which process are arrived in the ready queue
- **Completion Time**- which process completed its execution

  Max=$\sum_{i=1}^{n}$ (task complete + task execute+task weight)
- **Burst Time**-time required to process for CPU execution
- **Turn Around Time**- time difference between completion time and arrival time of a process
- **Waiting Time**- time difference between turn around time and burst of a process
- **Response Time** : Total response time(TRT)= $\sum_{i=1}^{k}$ FT + $\sum_{j=1}^{l}$ FT+ $\sum_{k=1}^{r}$ FT

  T=no.of task/ max(average execution time i) where i=0…n
- **Execution Time** of VM(ETVM)

  ETVM1 → $\sum_{i=0}^{T}$ execution time – start time (where T= 11 ,07, and 04 no of tasks)
- **Average execution time** =$\sum_{i=0}^{n}$ EVTM/ no of tasks (or) cloudlets

  3490+2897.5+3192.5/30=319.33ms

  (FT→ Finishing Time) 36150+12160+9040=57350 (where k=19,l=7 , r=4 & n=30)
- **Average response time**==$\sum_{i=0}^{n}$ FT_i / no of tasks (or) cloudlets 57350/30=1911.67
- **Estimate Time Quantum for Iteration One**: *Odd Round Iteration*

  C= Floor (5/2) =2: ETQ= BTime[C] = 15 m-sec.
- **Estimate Time Quantum for Iteration Two**: *Even Round Iteration*

  ETQ= FLOOR (6/2): ETQ=6 m-sec.

- **Estimate Time Quantum for Iteration** : *Single Task Iteration*

  ETQ=2 m-sec.
- The cloud network through put is defined as
- the average rate of successful data transfer through cloud environment
- It is a rate the users receive for their request and measured (KB/sec)
- The latency is the time taken for virtual machine
- from one device to another memory state of the VM to new server
- It measured in terms of milliseconds (ms)

Capacity of virtual machine

C vm=p*q

where

p→ processing speed of processor in MIPS ,

q→ no of CPU are busy to execute task

CONTIGENT ROUND ROBIN ALGORITHM

Initialize the Request in Queue as LQ (List of Cloudlets)

PTime (Process / Execution Time)

N: Number of process in LQ [] at any point of time

Initialize CBT=0 // Current Burst Time

Initialize NR=1 //Number of Round in RR order

PNR: $_{NR}$ th Cloudlet in the Queue

ETQ: Enhanced Time Quantum

Step: 1 Check the status of Queue

If (LQ is Empty) then

Assign the new task to Queue

Else

Append the task to the end of the Queue in FIFO order

End if

Step 2: Rearrange the Queue Task in increasing order of execution time of each task

Step 3: If (LQ is not NULL) then

calculate ETQ

If (N>1)

If (N%2==1) then

C= FLOOR (N/2)

ETQ=LQ[C]. PTime[C]

Else

C=N/2

ETQ=LQ[C]. PTime[C]

End if

Else

ETQ= LQ[First]. PTime[First]

End if

Call: ProcessExec (LQ [], ETQ)

Step 4: If (N>=NR) then

Set NR++ and goto Step 3

Else

Stop

Step 5: Append the New Arrival Task at the end of the Queue and Arrange the LQ in increasing order.

Step 6: Estimate the AWT, ATAT and Number of Context Switching.

Step 7: End

Function: ProcessExec (LQ [], ETQ)

Begin

i=0

while i>=count(LQ)

if (LQ[i].PTime[i] <=ETQ) then

Remove LQ[i] after execution of its reaming burst time

Else

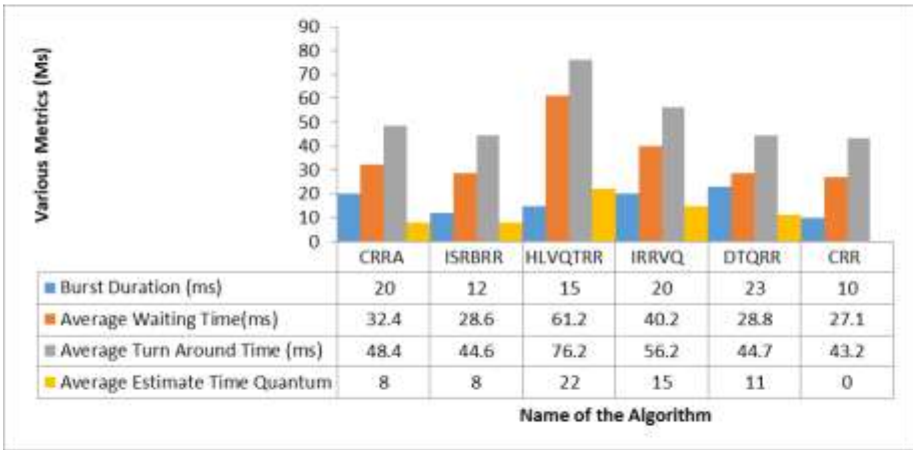
LQ[i].BTime[i] = LQ[i].BTime[i]-ETQ (Reduce the Execution Time of Current Process)

i++

End if

End

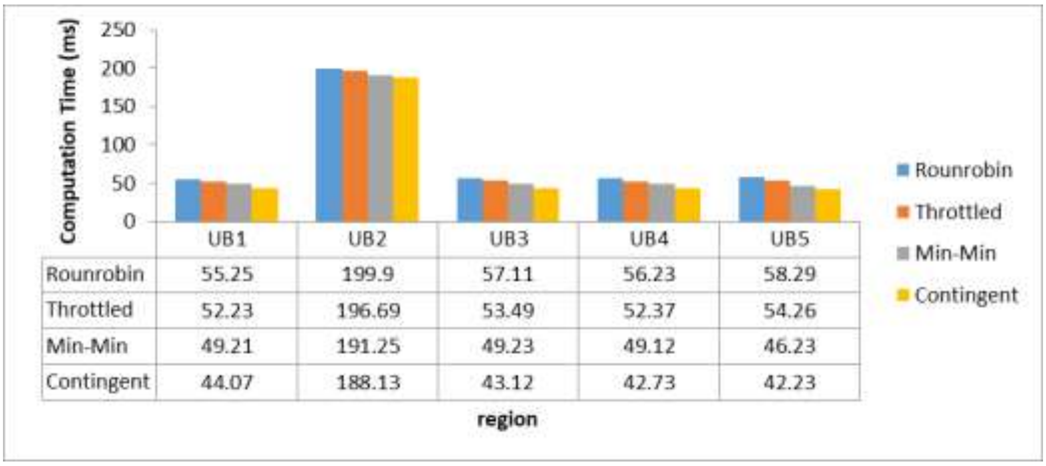
Performance Analysis for Metrics



	CRRA	ISRBRR	HLVQTRR	IRRVQ	DTQRR	CRR
Burst Duration (ms)	20	12	15	20	23	10
Average Waiting Time(ms)	32.4	28.6	61.2	40.2	28.8	27.1
Average Turn Around Time (ms)	48.4	44.6	76.2	56.2	44.7	43.2
Average Estimate Time Quantum	8	8	22	15	11	0

The performance of CRR is better compared to other metrics. The SR BRR performs better followed by HLVQTRR, CRRA, IRRVQ and DTQRR.

Performance Evaluation –Completion time of Contingent Round Robin



	UB1	UB2	UB3	UB4	UB5
Rounrobin	55.25	199.9	57.11	56.23	58.29
Throttled	52.23	196.69	53.49	52.37	54.26
Min-Min	49.21	191.25	49.23	49.12	46.23
Contingent	44.07	188.13	43.12	42.73	42.23

CloudSim

Programming and Implementation:

This section entails a description of the main steps of java code

implementations required for datacenter simulation. In the present study, the following

data center model was simulated in the CloudSim.

************** Length of Instruction from the (Task and Workloads) Section ************** Cloudlet Length 5000000

# of Task CPU: 1

Input file size: 100000

Output file size: 300000

************** Each Host ******************************

Memory RAM: 32 GB

Bandwidth: 8 Mbps

Storage (SSD/HDD): 2000 GB

************** Each VM (Virtual Machine) **************

Disk : 20 GB

Memory RAM : 1 Gb

VM MIPS: 1000 VM

Bandwidth: 1 Mbps

# of VM CPU: 1

Data center 1 specifications

Each Hosts

CPU Quad cores (Each core has 1000 mips)

Memory RAM 32 GB of memory

Storage 50 GB of storage (50000)

Bandwidth 8 mbps (2000 kbits/s)

Number of Data center brokers: 1

20 Cloudlets (task and workloads)

Length of Instruction 5000000 length of instruction

Input file size 100000 kb input file size Output file size 300000 kb

output file size CPU core 1

5 Virtual machines

Storage 20 GB

Memory 1 GB RAM

Virtual CPU 1 (each with 1000 mips CPU speeds)

Cloudlets Scheduler Timeshared

Each host of the data center model has the following specifications.

- Quad cores of 1000 MIPS
- 32GB memory RAM
- 20GB storage
- 8 mbps bandwidth.
- A single data center Broker
- 20 Cloudlets measuring 5000000 in instruction length
- 100000 kb file input size
- 300000 kb output file size.

Further, the database model comprises of

- 20 VMs
- Storage capacity of 20GB
- 1GB memory RAM
- 1 virtual CPU installed with 1000 MIPS speed
- Space-sharing Cloudlets Scheduler.

Network usage (Fog and Cloud delay)

	Config 1	Config 2	Config 3	Config 4	Config 5
Fog	10978.72	23018.24	44915.68	92318.56	183169.12
Cloud-only	166344	332589	664999	1050079	1101319

VM properties

VM ID	VM MIPS	MEMORY	NO.of.CPU	VMM
0	1000	1000	512	Xen/Windows
1	800	1000	512	Xen/Windows
2	600	1000	256	Xen/Windows
3	400	1000	512	Xen/Windows
4	200	1000	256	Xen/Windows

Parameters to configure VM

VM ID	Memory (Mb)	Bandwidth(mb)	No of core	Rate(MIPS)
0	4096	1024	4	32400
1	2048	1024	2	16200
2	1024	1024	2	8100

Task properties

Task ID	Length	File Size	Output File	No.of CPU
0	116228	300	300	1
1	52118	300	300	1
2	290543	300	300	1
3	197914	300	300	1
4	33734	300	300	1

Comparison among all load balancing algorithm

User bases	Rounrobin	Throttled	Max-Min	New one
UB1	55.25	52.23	49.21	44.07
UB2	199.90	196.69	191.25	188.13
UB3	57.11	53.49	49.23	43.12
UB4	56.23	52.37	49.12	42.73
UB5	58.29	54.26	46.23	42.23

Comparison between request servicing time (ms) of data centers

User bases	Rounrobin	Throttled	Max-Min	New one
DC1	0.68	0.45	0.32	0.25
DC2	0.57	0.44	0.41	0.38
DC3	0.48	0.43	0.32	0.27
DC4	0.45	0.48	0.44	0.34

Overall response time was 534750.87ms on average with a minimum of 564.98ms and maximum of 106218.92ms

Response time based on region

User bases	Avg(ms)	Min(ms)	Max(ms)
UB1	532309.45	564.98	1062618.97
UB2	533961.01	34633.81	1028050.50

On an average UB1 had slightly faster response time than UB2

Data centre request servicing times for DC1 was 532309.45ms on average with a minimum of 500.24ms and a maximum of 1062540.24ms

The Overall cost of the virtual machine was $61.39 which included VM cost of $0.88 and $60.51

Status record

Virtual Machines	Tasks	Starting Time	Finish Time	CPU cycles allotted	Status of Tasks	Status of VM	Capacity of VM
Vr2	Tl2	11:01	11:19	18250	Completed	Underloaded	16396
Vr3	T11	11:06	11:20	32295	Completed	Normal	0

Assumption of cost:

Cost per 1 Gb of data transfer was $0.11

Cost of VM per hour (1 Gb to 100 mips) was $0.11

Memory cost was $0.05 and storage cost was $0.1

Execution Time of VM(ETVM)

ETVM1 → $\sum_{i=0}^{T}$ execution time – start time (where T= 11 no of tasks)

ETVM1 → $\sum_{i=0}^{T}$ execution time – start time (where T= 07 no of tasks)

ETVM1 → $\sum_{i=0}^{T}$ execution time – start time (where T= 04 no of tasks)

Average execution time =$\sum_{i=0}^{n}$ EVTM/ no of tasks (or) cloudlets

3490+2897.5+3192.5/30=319.33ms

Response Time

Total response time(TRT)= $\sum_{i=1}^{k}$ FT + $\sum_{j=1}^{l}$ FT+ $\sum_{k=1}^{r}$ FT (FT→ Finishing Time)

36150+12160+9040=57350

(where k=19,l=7 , r=4 & n=30)

Average response time==$\sum^{n}_{i=0}$ FT_i / no of tasks (or) cloudlets

57350/30=1911.67

Throughput

T=no.of task/ max(average execution time i) where i=0…n

Makespan(maximum completion time of all tasks)

Max=$\sum_{i=1}^{n}$ (task complete + task execute+task weight)

Capacity of virtual machine

C vm=p*q where p→ processing speed of processor in MIPS , q→ no of CPU are busy to execute task

Conclusion

The problems of participation between the threshold and the center and also a few existing levels of facet registration are shown. The new approach effectively increases the data centers energconsumption rate and average network flow transfer time. Contingent Round Robin Load Balancing is another approach that is available to facilitate improvements to the allocation of job between workstations at runtime. Future research improvements include Hypervisor Security and the detection of Whole VM failures such as DC identity, VM migration and consolidation of VMs. Identifying Secure and Authenticate Edge data Center in Fog. Global distribution and achieving minimal downtime of resources. In fog VM migration ratio is high and QoS is not ensured. Heterogeneity of IoT device reduces efficiency of fog nodes.

References:

1. **Simar Preet Singh, Anand Nayyar, Harpreet Kaur ,"Dynamic Task Scheduling Using Balanced Vm Allocation Policy For Fog Computing Platforms " , Big IoT Data Analytics in Fog Computing,Vol 20,No.2 ,2019.**
2. **Siminesh.C.N,Grace.M.K.E,Ranjitha.K, " A proactive flow admission and re- routing scheme for load balancing and mitigation of congestion propagation in sdn data plane", Int. J. Comput. Netw. Commun,Volume 10, 2019.**
3. **Lawanya shri.M, Santhi. K, Gangadevi.E, Meenatchi. S, Navaneethan.C,"Secure Virtual Machine Migration Technique using Blockchain", International Journal of Psychosocial Rehabilitation, Volume 24,Issue 8,2020.**
4. **Huang.L, Shen.Q,Shao.W, "Congestion Aware Fast Link Failure Recovery of SDN Network Based on Source Routing", KSII Trans. Internet Inf. Syst. 2017, 11, 5200–5222. Sustainability, 12, 4255 26 of 28 , 2020.**
5. **Istabraq A.I, Emad H, Hemiary A.I ,"Virtualized Fog Network with Load Balancing for IoT based Fog-to-Cloud",International Journal of Informatics Visualization, 10.30630/joiv.4.3.374,Volume 3,2020.**
6. Balaji K et.al,"An energy efficient load balancing on cloud computing using adaptive cat swarm optimization", Materials Today: Proceedings (Science Direct), https://doi.org/10.1016/j.matpr.2020.11.106

6. Dr.A.Senthil Kumar, Dr.G.Suresh, Dr.S.Lekashri, Mr.L.Ganesh Babu, Dr. R.Manikandan. (2021). Smart Agriculture System With E – Carbage Using Iot. International Journal of Modern Agriculture, 10(01), 928 - 931. Retrieved from http://www.modernjournals.com/index.php/ijma/article/view/690 , January 2021.

7. Dr. R. Manikandan, Dr Senthilkumar A. Dr Lekashri S. Abhay Chaturvedi. "Data Traffic Trust Model for Clustered Wireless Sensor Network." INFORMATION TECHNOLOGY IN INDUSTRY 9.1 (2021): 1225–1229.