A Survey on Various Load Balancing Algorithms in Cloud Computing

Jalpa M Ramavat¹, Prof Rikin Thakkar²

^{1, 2} Department of Computer Engineering ¹ VGEC,Ahmedabad,India ² SOCET-Ahmedabad

Abstract- The use of cloud environment is growing day by day. The small businesses are using cloud for their day to day need of resources because cloud provide on demand and pay per use services. The business which are of low budget and not be able to setup wide infrastructure for recent technologies, Cloud computing is blessing for them. As the need increases, managing load at cloud is the biggest challenge that the cloud provider has. Distributing equal load in different node which may be geographically at different location is major issue. Various load balancing algorithms are there for even distribution of load. Again load balancing will improve the parameters like cost, response time , through put etc. Also Load balancing is a big aspect in terms of power utilization and resource utilization.

In this paper we will discuss various static as well as dynamic load balancing algorithms with their advantages and disadvantages.

Keywords- Cloud computing, Load balancing, on demand access

I. INTRODUCTION

Cloud Computing is one of the emerging and important technology now a days. It provides on demand access to resources []. So the users have to pay for what they use. It will be adopted by more and more users, Industries, organizations -. The need to manage billions of user requests such that all requests will satisfy in proper time with less cost is very important. It will happen when all available resources utilize fairly.

The load balancing is the mechanism through which proper resource utilization is possible by distributing load equally among available resources. Also it will improve parameters like response time, throughput, cost etc.

This paper will discuss various available load balancing algorithms and their pros and cons.

The paper is arranged as follows. Section 2 gives brief idea about load balancing. Section 3 covers Literature survey of Load balancing algorithms. Section 4 contains the metrics for comparing the discussed algorithms and section 5 is conclusion of whole paper.

II. LOAD BALANCING

It is the process through which we can achieve fair resource distribution between tasks and improves the performance of cloud [8]. Highly loaded datacenter utilizes more power. Load balancing will also improve the power efficiency of datacenter and limit the wastage of power.

Load Balancing algorithms are mainly categorized in two types Static and Dynamic.

Static algorithm uses the current state of node [2].It will not bother about the previous state of node. The user requirements and available resources are predefined. Run time changes in requirements and resources are not allowed [1]. They are easier to implement and more suitable for homogeneous environment [4].

Dynamic algorithms use previous as well as current state of node to distribute the load [2]. The user requirements and Resources can be changed at run time. They are suited in homogeneous as well as heterogeneous environments [4].

III. LITERATURE REVIEW

Priyadarashini et al [2] propose the comparative study of Static algorithm like robin and weighted round robin algorithm and dynamic algorithms like FCFS, Throttled , Modified Throttled and Particle Swam Optimization Algorithm. In round robin the request are assigned in circular queue. Each job will be assigned to available VM for some fix time period after that VM will be moved at the end of queue.Weighted Round Robin will assign weight to each node. So requests are received depending on weight. Throttled Load Balancing algorithm group VM according to the request they can handle and request assigned to the VM which can handle it by searching suitable VM.In Modified Throttled the second search will be started from the VM that is next to previously allotted VM.In FCFS the request are queued as they came. Load balancer will assign the first request to VM by considering Load of VM. In Particle Swam Optimization the pbest for each particle is calculated and compared with its previous fitness value and if new is greater than old is updated by new one. From all pbest the best value is assigned to Gbest and than particle position is updated and new velocity is calculated. They simulate above dynamic algorithm in cloudsim.

Geethu Gopinath et al [3] done comparative analysis of Min –Min and Max – Min algorithms. Min –Min is simple to implement. It will calculate minimum completion time for all nodes. Then the task having minimum completion time is chosen and assign to the respective node. Max- min chooses the task having maximum completion time to run on node. It will run short task concurrently with the long one. They simulate and compare both algorithm using cloud sim.

Shridhar et al [4] propose hybrid scheduling algorithm that combine Divide and conquer approach and throttled algorithm. The algorithm is having two pass. In first pass it uses divide and conquer approach for dividing task to available resource handler and assign task to available RH. Every time request is completed, the status of the RH is given to load balancer for next allotment. In second pass for next allotment the request will be allotted to available RH which was not used recently. Priority of request is also considered by load balancer. The proposed algorithm will be compared with throttled on simulator cloud sim.

Navtaj Shingh et al [5] propose algorithm uses the Improved Max min and ant colony approach. They sort the virtual machine according to improved max min and then calculate the execution time of submitted task on each resource. They use Ant approach for calculation of execution time. Task having maximum execution time is assigned to the resource having minimum completion time. Then that task is removed from task set. They simulate the proposed algorithm in cloud sim and compare it with improved max min algorithm.

Reena Panwar et al [6] propose Dynamic Load Management algorithm. They consider dynamic set of virtual machines. When new request comes they check for best suited virtual machine. Once the request is bound they remove that VM index from group of available VM, so it will not be considered for any future request until it finishes assigned workload and become available again. As the algorithm will not every time considers an overloaded VM again and again for scheduling so has less overhead. The author compares this algorithm with optimal VM Load balancing algorithm and simulates result in cloud analyst.

Surbhi Kapoor et al [7] propose cluster base approach for load balancing. They group the VM in clusters by using K Mean Clustering by considering three resource types as parameters i.e CPU processing speed, Memory and Network bandwidth. Load balancer will then assign the request to the appropriate VM of the chosen cluster by looking into the list of cluster and change the status to Available. The Proposed Algorithm is compared with throttled and modified throttle algorithm.

Mohamed et al [8] propose the load balancing architecture for cloud computing based on multiple cluster. The main three load balancing elements are: Main load balancer, Local load balancer and authentication element. MLB maintain the table of clusters with their processing capacity and match client request particular cluster. LLB uses scheduling algorithm to perform load balancing within cluster. They add authentication layer to authenticate user and also grant priority to user and his job.

Akhil Goyal et al [9] discuss firefly algorithm by using behavior of firefly. Threshold value is set for all virtual nodes. It will maintain index table for VMS. When request came the index table is searched for least loaded VM. Any VM will not get load more than threshold value. It focuses on Energy consumption which is a key research issue in cloud computing environment.

Ravi Shankar et al[10] propose Power Aware Resource Allocation Policy for Hybrid cloud. It passes data center list and request queue as argument to algorithm. It will first check the length of request queue and exit if queue is empty. Then compute the power efficiency of data center of public as well as private cloud and sort them. Private request is having higher priority than public request. If the request is private, it will be assigned to the high power efficient private datacenter which can satisfy the request. If high power efficient private datacenter is not available than assign request to public cloud datacenter having high power efficiency. If the request is public then it will be assigned to high power efficient public datacenter. If it was not available than assign to the private datacenter that is having minimum power efficiency.

IV. ANALYSIS METRICS OF EXISTING ALGORITHMS

Aultori hmsResultAdvanueg geDisadvanu gearashin e i et al [2]arashin voir (CTS and particle swarmresponse time and cost compared i dading more response time.no.no.[2]Noficiency in load balancing, more improper response i et al 2]Simple to improper response time.Sufair in nature, poor Load balancing, Poor more responseSufair in nature, poor Load balancing, Poor more response time.Sufair in sponse response time.Notified and simulated and simulated inclouding in takes then other two algorithm in terms of response se time and cost.Notified terms of response se time and cost.Static in insplement. nature, response ime and cost.Priyad 12]Robel response and weighted and weighted and weighted and weighted and weighted in the static is ecand robin.Improve response response time and suitableStatic in response response time and suitableSimulated in response response to algorithm in terms of response responseStatic in nature, response response to algorithm in terms of response that mand found ind count response response to algorithm in terms of response to algorithm in terms of response response performs sector in responseStatic in nature, response response pe	Table 1. Analysis Of Some Load Balancing Algorithms				Priyad	Particl	Modified	Less	Priority is	
InvestInves	Author	Algorit	Result	Advantage	Disadvanta	arashin	e	throttled,	response	not
Priyad arashin i et al [2]Rou robinlow simple or implement halancing, more responseSimple to implement halancing, Poor nessource utilization, more responseSimple to implement halancing, Poor responseSimple to implementSimple to implementSimple to implementSimple to implementSimple to implementSimple to response time.Simple to implementSimple to implementSimulatelin implementSimple to implementSimple to implement <th< td=""><td></td><td>hms</td><td></td><td>-</td><td>ge</td><td>i et al</td><td>Swarm</td><td>FCFS and</td><td>time and</td><td>considered</td></th<>		hms		-	ge	i et al	Swarm	FCFS and	time and	considered
Pryad response i ct al [2]Kouin efficiency more response time, improper resource time, improper resource response i ct al rotal i ct al rotal i ct al rotalSimulation response time, improper improper response time, improper response timeStatic response response timeStatic response response timeStatic response response timeStatic response response timeStatic response response timeStatic response response timeMore time response timeStatic response timeStatic response response timeStatic response response timeStatic response response timeStatic response response timeStatic response response timeStatic response response timeStatic response response timeStatic response response timeStatic response required to static response timeMore time required to response timeMin response required to response timeMin response required to responseMin response required to response time responseMin response required to response timeMin response required to response timeMin response response required to response timeMin response response required to response timeMin response required to response timeMin response required to response timeMin response required to response timeStatic response required to response time and response time </td <td>D : 1</td> <td>D 1</td> <td></td> <td></td> <td></td> <td>[2]</td> <td>Optimi</td> <td>Particle</td> <td>cost</td> <td>, Doesn't</td>	D : 1	D 1				[2]	Optimi	Particle	cost	, Doesn't
arashin i et al [2]iod bad halancing, more response time, improper esource management.implement poor Load balancing, Poor response time, improper response ti et al et al i et al i et al i et al [2]Modified more response improper esource management.Simple on static static static staticModified and Found that PSO perform better than other two algorithm in terms of response ti et al (2)Modified more response time, i et al (2)Privad additionModified balancing, nore response time, i et al (2)Modified management.Resource management.Modified balancing, Poor response respons	Priyad	Round	low	Simple to	Static in		zation	swarm	compare to	consider
i et al [2]Index halancing, more response time, in proper management.Simulate halancing, more response time, impersonse timeare compared halancing, more response time utilization, more response timestatic in cloudsim and found that PCPS perform algorithm in algorithm in terms of response term of algorithm in term of term of<	arashin	robin	efficiency in	implement	nature,			optimization	Modified	resource
[2] more response time, improper response time, improper response tine, improper response tine, improper response tine, improper response tine, improper response improper response improper i et al [2]Weigh ted round robinBetter simplementStatic indenentand simulated incource response immroper response immroper improper response immroper i et al (2)More time response response immroper i et al (2)More time response response time, i et al (2)Static i more response respon	i et al		load		poor Load			are compared	throttled	specific
Image: series of the series	[2]		balancing,		balancing,			and simulated	and FCFS	demand of
Privad i et al i i e			more		Poor			in cloudsim		task.
Imagetime, improper resource management.utilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more response timeutilization, more moreutilization, more response timeutilization, more moreutilization, more response timeutilization, more response time the VM take poor.utilization, more more time time take the VM take poor.utilization more time take time take time take time take time takeMan take time take time take timeutilization, more take time take timeutilization, more take time take timeutilization take time take time take timeutilization take take take take take take takeutilization take take take take take take takeutilization take take take takeutiliza			response		resource			and found		
Image: Source management.more response timemore response timemore response timemore response timemore response timemore response timemore response time and cost.more response time and cost.more response time and cost.Min cloud sim and cost.Simple to mature, and cost.Static in analyze, and cost.Priyad rankinThrott end algorithm is et al (2)Thrott algorithm is to algorithm is to algorithm is to algorithm in the rai algorithm in the response time and weighted at round robin.Min et al (3)Simulated in cloud sim implement, and that mumber of performment is depended upon cloud performs the second of algorithm the second discussed static the response time and weighted at the second algorithm the second is static to algorithm the second discussed state to cost is state to the vortex and make span.Compare to response time and task is required.Simulated in cloud sim of algorithm is in the response time and task is required.Simulate in cloud sim of algorithm is the second discussed time and cost is state to resources time and cost is previous data algorithm cost is previous data and found the vM that and found the vM that and found the cost compare to or the vM that and found exist the vM that and found the cost compare to the vM that and found the cost compare to the vM that and found that and found that and found that the second of the vM that and found that for the vM that and found the compare to to the vM that and found that the vM that and found the total and found that for the vM that and found that the vM that and found the total and found that the vM that and found that for			time,		utilization,			that PSO		
Privad I 21Weigh round round round robin i et al 121Simple to round robin round robinSimple to response response response response required to algorithm in time te al and Weigh edBetter assumption response response response response response required to algorithm in the t al algorithm is the t al i at al compare to round robinSimple to response response required to suitable time, good suitable to ite al algorithms i.eMore time response required to suitable time, good suitable to ite al algorithms i.eMore time response required to suitable time, good suitable to ite al (2) it has statie that or ound robinMore time response time and cost is port PSO the response time and cost is port to ite al (2) it has statied from the VM that statied from the VM that statied from the VM that and found and found and found cost is provious)Compare to response time, Less port to source time, Less provious)Compare to response time, Less port to source to FCFS.Compare to response time, Less portCompare to response time, Less portSimulated in cost is portSimulated in response time, Less portSimulated in response time, Less portSimulated in response time, Less portSimulated in response time, Less portSimulated in response time and cost is portSimulate in response time and cost is portSimulate in response time and cost is portSimulate in response time			improper		more			nerform		
Priyad arashin i et al round robinBetter implementSimple to implementStatic implement			resource		response			better than		
Privad arashin i et al i ct al i et al i et al i et al i et al [2]Better compare to round robinSimple to implementStatic implementStatic implementMore time required to search witableMin Geethu (Gopina (Min (Min Min Min Min MinSimulated in cloud sim and cost.Simple to implement, mature, priorPriyad arashin i et al [2]Throttl algorithms i.e and Weighted and Weighted and Weighted arashin i et al [2]Import manageme and Weighted and Weighted and Weighted arashin i et al [2]Import manageme and Weighted and Weighted arashin i et al [2]Compare to static algorithms is extremed the VM that is extremed is extremed and found econd algorithms the second discussed time, Less previously allotted VM, simulated in cloud sim and found ecost is previously allotted VM, allotted VM, and found ecost is previously allotted VM, and found ecost is previously allotted VM, and found ecost compare to response time and found cost implement, and found ecost is previously allotted VM, and found cost implement to cost compare to response time and found cost is previously allotted VM, and found ecost is previously allotted VM, and found ecost compare to response time cost compare to response time cost compare to response time cost compare to response time cost compare to response time cost, less time and Found poor response time cost, less time and found ecost is poor response time cost, less time and cost, less t			management.		time			other two		
Introduct i et al [2]Deficit of round robinSimple to implementStatic implementOnly and required to static and weighted nat namageme and Weighted i et al [2]More time required to static algorithms i.e loadGeethu suitable (Gipina suitable (Gipina (Gipina (Gipina (Gipina (Gipina (Gipina) (Gipina)Min simulated in cloud sim and found indered more indered (Gipina)Simulated in more more (Gipina)Simulated in more nature, priorSimulated in nature, prior (Gipina)Simulated in more nature, priorSimulated in nature, priorSimulated in nature, prior (Gipina)Simulated in more more (Gipina)Simulated in more nature, priorSimulated in nature, priorSimulated in nature, prior (Gipina)Simulated in more nature, priorSimulated in nature, priorSimulated in nature, priorSimulated in nature, priorSimulated in nature, priorSimulated in nature, priorSimulated in resources utilization and nake span.Simulated in in tetterSimulated in in priorSimulated in more more tetterSimulated in in priorSimulated in in in tetterSimulated in in priorSimulated in in in tetterSimulated in in priorSimulated in in in tetterSimulated in in in tetterSimulated in in in tetterSimulated in in in tetterSimulated in in in tetterSimulated in in in tetter <t< td=""><td>Privad</td><td>Weigh</td><td>Better</td><td>Simple to</td><td>Static</td><td></td><td></td><td>algorithm in</td><td></td><td></td></t<>	Privad	Weigh	Better	Simple to	Static			algorithm in		
anama (2)compare to robinimpendent responsemerce responseCompare to responseSimulated in responseSimulated in responseSimulated in responseSimulated in nature, and that nature, and that i et al [2]Modifi responseSection responseStatic in nature, of responseStatic in nature, of responsePriyad [2]Modifi ret al edBetter as round robinCompare to static number of static num drobinCompare to static responseCompare to static responseCompare to static responseCompare to static responseCompare to static responseCompare to static responseCompare to response time and cost is por.Modifi responseStatic in nature, and task is required.Priyad a rashin i et al [2]Modifi response time and cost is s next to previously allotted VM tast referoCompare to static time, Less por.Compare to PSO the response time and cost is por.Geethu Affin AffinMin Min Min Performs betterSimulated in inglement, and make span.Simulated in response time and task is porPriyad arshin i et al [2]FCFSSimulated in cloud sim and found betterSimulate in cost, Less response time and found time and foundCompare to resourcesCompare to PSO it has por response time and cost.Compare to PSO it has por task is porSimulated in resourceSimulate in r	arashin	ted	compare to	implement	Stude			terms of		
I chara [2]Ionard Ionard (2]Ionard Ionard (2]Ionard Ionard (2]Ionard Ionard (2]Ionard Ionard (2]Ionard Ionard (2]Ionard (2)Ionard <b< td=""><td>i et al</td><td>round</td><td>round robin</td><td>implement</td><td></td><td></td><td></td><td>response time</td><td></td><td></td></b<>	i et al	round	round robin	implement				response time		
Interprised and compare to i et alImprove compare to staticImprove response required to searchGeet me GopinaMin MinSimulate in cloud sim and foundStatic in implement, nature, prior[2]Improve staticload searchsuitable witableGeet mi (1)Min (1)Simulate in cloud sim and foundmore more priormature, prior[2]Modifi ed compare to resume edBetter as staticCompare to staticCompare to staticCompare to resourcesCompare to priorMin the tal (1)Min mature, resourcesMin mature, resourcesMin mi the tal (2)Modifi the second the secondCompare to staticCompare to resourceCompare to resourcesMin mi required.[2]Priyad edModifi the second the VM that sarart of previously allotted VM. tansferCompare to resourceCompare to resourceCompare to resourceCompare to resourceSimulated in resourceSimulate in resource<	[2]	robin	Tourie Toom					and cost		
Priyad arashin i et al i et al i et al [2]Enter as compare to staticImprove response responseMore time required to sourcesMin GopinaSimulated in MinSimulated in colud sim and foundSimuleton manageme suitableMin to et al GopinaMin MinSimulated in colud sim and foundSimuleton more prforSimulated in implement, and weighted and weighted and weighted and weighted arashin edImprove responseMore time suitableMin to et al GopinaMin th et al [3]Min th et al (1]Simulated in and found of algorithm is depended is depended (1]Simulated in mumber of resourcesSimulated in th et al (1]Simulated in mumber of resourceSimulated in (1]Simulated in mumber of resourceSimulated in (1]Simulated in (1]	[2]	TODIII						and cost.		
arashin i et al i et al i et alcompare to time, good staticresponse time, good scarchGopina th et al th et al (3)Mincloud sim cloud sim thatimplement, more prior[2]Implement and Weighted and Weighted and Weighted arashin i et al i et al i et al (2)Nodifi Better as scarchCompare to staticSompare to Promate PromateMin Min performance performanceHaks then performance i depended mupon cloudNodifi Min Min upon cloud performs tetrms of number of number of Heavier tasks are there then and make started from the VM that Response time cost, Less cloud sim tet VM the VM that Response cloud sim the VM that Response cloud sim the VM that Response cloud sim the VM that Response cloud sim the VM that cost, Less cloud sim cloud sim to FCFS.Compare to static from time, Less proviculy dataGeetha Response the thet al thet al <td>Priyad</td> <td>Throttl</td> <td>Better as</td> <td>Improve</td> <td>More time</td> <td>Geethu</td> <td>Min</td> <td>Simulated in</td> <td>Simple to</td> <td>Static in</td>	Priyad	Throttl	Better as	Improve	More time	Geethu	Min	Simulated in	Simple to	Static in
i et al [2]static algorithms i.e algorithms i.e 	arashin	ed	compare to	response	required to	Gopina	Min	cloud sim	implement,	nature,
[2]algorithms i.eloadsuitable[3]thatnumber ofknowledgeRound robinand WeightedntVM.Performanceof algorithmtasks thenresourcesPriyadModifiBetter asCompare tocompare toStaticPSO theis dependedMin Minand task is[2]edThrottlThrottlaalgorithmsresponsetime andcost isFor sourcefor mumber ofresources[2]edthe seconddiscussedtime andcost ispor.for mumber ofresourceresource[2]edthe secondtime, Lesspor.for mumber ofresourceresourceresource[2]edtransfercost ispor.for masterfor mumber ofresourceresourceis next totime, Lesspreviouslydatacost sisfor datafor mumber ofsimulated incost compare tosimulated incost, Lesscost unsfercost ompare tofor datafin oreimplement,nature soand foundconsumingcost ompare tofin orefor dagorithmtask sthenresourcesfit et aland foundconsumingcost masterfin orefor dagorithmtask stheni et aland foundcost masterfin orefor dagorithmtask sthenresourcesfit et aland foundfingement.performancefor dagorithmtask sthenresources<	i et al		static	time, good	search	th et al		and found	more	prior
Round robin and Weighted round robin.manageme ntVM.Performance resourcesheavier of algorithm is dependedheavier of algorithm is dependedof resourcesPriyad i et al [2]Modifi ed compare to ti et al [2]Better as compare to the second the second is next to algorithm the VM that is next to cost is previously alloted VM.Compare to response time. Less to next to to algorithmCompare to response time and cost is poor.Compare to response time and cost is poor.If More response time and cost is poor.If More response time and cost is poor.If More response time and cost is poor.Max- formance the VM that are there then and make span.Simulated in nature so priorPriyad arashim i et al [2]Robin and found to FCFS.Compare to time cost compare to FCFS.Compare to response time cost compare to FCFS.Compare to response time and cost.Simulated in formance cost compare to FCFS.Simulate on to for transfer to for transfer to for transfer to FCFS.Simulate on to for transfer to for transfer to for transfer to for transfer response time cost compare to FCFS.Compare to PSO it has poorMax- to for the All the tal is depended upon cloud that that that that that that thatSimulate in response time and to for taks then transfer response time cost.Compare to PSO it has poor response time and cost.	[2]		algorithms i.e	load	suitable	[3]		that	number of	knowledge
Priyad arashin i et al i i et al i			Round robin	manageme	VM.			performance	heavier	of
Priyad arashin i et al i et alModifi Better asCompare to staticCompare to PSO the responseSompare to PSO the responseis depended upon cloud Environment.Min Min Performs the secondall ottak is required.[2]edThrottled as ascarch will be stated from the VM that all ottakalgorithms (2] it has the VM that stated from the VM that all ottakCost is poor.is meand the VM that the VM that all ottakCost is poor.Geethu the VM that the VM that all ottakMaxe the VM that transferGeethu the tal the VM that the Cost Less the VM that the Cost LessMaxe the VM that the VM that the VM that the VM that the VM that the Cost LessMaxe the VM that the tal the tal the tal the tal the tal the tal the talMaxe the tal the tal the tal the tal the tal the tal the talMaxe the tal the tal <td></td> <td></td> <td>and Weighted</td> <td>nt</td> <td></td> <td></td> <td></td> <td>of algorithm</td> <td>tasks then</td> <td>resources</td>			and Weighted	nt				of algorithm	tasks then	resources
Priyad arashin i et al [2]Modifi 			round robin.					is depended	Min Min	and task is
Instruct arashin i et al [2]edd eddCompare to static algorithmsPSO the response time and cost is poor.Environment. If Morebetter in terms of number of Heavier tasks utilization are there then market modified[2]edfrom te second[2] it has terms of Less port.cost is poor.resource terms of number of previously dataPSO the response time, Less poor.Fee then and make span.market modified[2]it has terms of terms of 	Privad	Modifi	Better as	Compare to	Compare to			upon cloud	Performs	required.
Interm i et al i et al [2]Compare to functionalgorithms algorithms discussed time and cost is poor.If More number of number of Heavier tasks are there then and make Min Min performs better.If More resource utilization are there then and make Min Min performs[2]edform the second the VM that is next to alloited VM. transfer cloud sim i et al i et al i et alLess time, Less previously alloited VM.poor.If More response time, Less previously alloited VM.Geethu transferMax- forma the tal i the tal [3]Simulated in simulated in cloud sim implement, formaSimulated in priorSitic in nature so perior the tal [3]Priyad arashin i et al [2]FCFSSimulated in cloud sim timeSimple to implement, provintedCompare to poorIf more formaSimple to forma formaSimple to forma formaSimple to forma formaSimple to forma formaSimulated in forma formaSimulated in formaSimulated in forma formaSimulated in forma formaSimulated in forma forma formaSimulated in forma formaSimulated in forma formaSimulated in forma formaSimulated in forma formaSimulated in formaSimulated in forma formaSimulated in forma formaSimulated in forma formaSimulated in formaSimulated in formaSimulated in formaSimulated in forma formaSimulated in formaSimul	arashin	ed	compare to	static	PSO the			Environment.	better in	
TerkingInformed useRegrammeRegrammeRegrammeResponseInformed usenumber of the secondresource tuilization[2]edthe second(2) it has tesscost is poor.poor.Heavier tasks are there then better.and make main[4]is next to previously allotted VM.time, Less dataLesspoor.Heavier tasks are there then better.and make main[5]Simulated in cloud sim and found less VM and cost compare to to FCFS.compare to implement.Geethu formed timeMax- formationSimulated in formationSimulate of main thet al formationSimulated in formationSimulate of main formationPriyad [2]FCFSSimulated in betterSimple to implement.Compare to implement.Compare to poorIf number of formationResource formationformation formation[2]FCFSSimulated in betterSimple to implement.Compare to poorIf number of formationresources formationrequired.[2]FCFSSimulated in betterSimple to implement.Compare to poorFormation formationCompare to modified the and cost.If number of formationrequired.[2]FCFSSimulated in compare to modified throtted.Simple to implement.Compare to modified time and cost.If number of formationResponse ter formationResponse ter formationRespo	i et al	Throttl	Throttled as	algorithms	response			If More	terms of	
TendColorInterformanceInterformationHeavier tasksutilizationsearch will be started fromLesspoor.Interformationare there thenand makethe VM that is next toLesspoor.InterformationMin Minspan.previouslydataInterformationbetter.InterformationSimulated inSimulated inallotted VM.transferInterformationGeethuMax-Simulated inSimple toStatic inSimulated in cloud simcost, LessInterformationGopinaMincloud simimplement,nature soand foundconsuming less VM and cloud simcompare toInterformationand foundIf morepriordata transfer cost compare to FCFS.ThrottledInterformationIf andNinand foundResourcesPriyad arashin i et alFCFSSimulated in modifiedSimple toCompare to implement.If anyPerformationBetter in and task siteresources[2]FCFSSimulated in and FoundDynamic in implement.PSO it has poorpoorIf anyResource are moreand make and makeIt any[2]betternatureresponse time and compare to modifiedCompare to implementIf anyAnyAny[2]FCFSSimulated in implementSimple to implementCompare to implementIf anyAnyAny[2]better <td>[2]</td> <td>ed</td> <td>the second</td> <td>discussed</td> <td>time and</td> <td></td> <td></td> <td>number of</td> <td>resource</td> <td></td>	[2]	ed	the second	discussed	time and			number of	resource	
Search Will ceLesspoor.are there thenand makestarted from the VM that is next toResponsepoor.HeHeMin Min nspan.is next totime, Lesstime, LessPerformsbetter.HeHeallotted VM.transferFormationcost, LessGeethuMax-Simulated inSimple toStatic inSimulated in cloud simcost, LessImage: static different timecost, LessGeethuMax-Simulated inSimple toStatic inand found less VM and cost compare to to FCFS.Throttledcompare toIfIfmorepriorPriyad arashin i et al [2]FCFSSimulated in to FCFS.Simple to implement.Compare to PSO it has poorCompare to PSO it has poorIfIfnumber of performaceMax Min and Maxinand task is resourcesPriyad arashin i et al [2]FCFSSimulated in ponarie to modifiedSimple to natureCompare to PSO it has poor response time cost.Compare to response time and cost.IfImmet response time and cost.IfImmet response time and cost.IfImmet response time and cost.IfImmet response time and cost.Immet response time and cost.Immet response time and cost.Immet response time and cost.Immet response time and cost.Immet response time and cost.Immet response time an	[2]	ea	search will be	[2] it has	cost is			Heavier tasks	utilization	
Privad arashin i et al [2]FCFSSimulated in cloud sim and FoundSimple to implement, implement, operationMin Min Performs better.span. Performs better.Privad arashin i et al [2]FCFSSimulated in cloud simCompare to implement, operationGeethu GopinaMax- Min Min GopinaSimulated in Min Min Cloud simStatic in nature so priorPrivad arashin i et al [2]FCFSSimulated in cloud simCompare to implement, operationGeethu Max- GopinaMin Min Min MinSimulated in motifiedStatic in nature soPrivad arashin i et al [2]FCFSSimulated in cloud simSimple to implement, priorCompare to motifiedMax- FSO it has poor response time and cost.Geethu Max- fieldMax- fieldSimulated in fieldSimele to response time and cost.FCFSSimulated in fieldCompare to motifiedMin fieldFCFSSimulated in fieldSimple to fieldCompare to motifiedFCFSSimulated in fieldSimple to fieldCompare to fieldIf number of fieldFCFSFCFSPrivad fieldFCFSSimulated in fieldSimple to fieldCompare to fieldCompare to fieldFCFSFCFSFCFSFCFSPrivad fieldFCFSSimulate in fieldSimple to fieldCompare to fieldFCFSFCFSFCFSFCFSPrivad fie			started from	Less	poor			are there then	and make	
Privad arashin i et al [2]FCFSSimulated in simulated in compare to cloud simSimple to time consuming timeCompare to periorGeethu for thet al thet alPerforms better.Simulated in implement, nature soStatic in nature soPrivad arashin i et al [2]FCFSSimulated in simulated in compare to cloud simCompare to implement, natureGeethu for thet al thet alMax- cloud simSimulated in implement, and foundStatic in nature soPrivad arashin i et al [2]FCFSSimulated in simulated in betterSimple to implement, natureCompare to PSO it has poor response time and cost.Compare to response time and cost.Max- cost.Simulated in sidependedSimulated in taks then securesPrivad arashin i et al [2]FCFSSimulated in cloud sim implement, natureCompare to response time and cost.Compare to response time and cost.Compare to response time and cost.If number of response time and cost.FCFSPrivad i et al [2]FCFSSimulated in sponse time and compare to modified throttled.Simple to response time and cost.Compare to response time and cost.If one response time and cost.Performs response time and cost.Performs response time and cost.Performs response time and cost.Performs response time and cost.Performs response time and respo			the VM that	Response	poor.			Min Min	span.	
Privad arashin i et alFCFSSimulated in simulated in cloud simSimple to transferSimple to transferGeethu GopinaMax- MinSimulated in cloud simSimple to implement, and foundStatic in nature so priorPriyad arashin i et alFCFSSimulated in cloud simCompare to implement, thet alCompare to implement, thet alIf more and foundSimulated in priorSimulated in of thet alSimulated in and foundSimulated in priorNine thet alSimulated in and foundSimulated in priorNine priorNine priorSimulated in priorSimulated in priorNine priorSimulated in priorSimulated in prior			is next to	time Less				Performs		
Privat allotted VM.transfer transferGeethuMax- GeethuSimulated in Coud simStatic in nature so of cloud simSimulated in cloud simcost, Less timecost, LessFeethuMax- GopinaSimulated in MinSimple to cloud simStatic in nature so priorand found data transfer cost compare to to FCFS.compare to ThrottledFeethuMax- feethuSimulated in and foundSimple to of algorithmstatic in is dependedPriyad arashin i et al [2]FCFSSimulated in cloud simSimple to implement.Compare to PSO it has poorCompare to FSO it has poorFESimulated in seponse time cost.Simple to implement.Compare to FSO it has poorFESimulated in seponseSimple to implement.Compare to poorFEFESimulated in seponseSimple to implement.Compare to poorFEFESimulated in implement.Simple to poorFEFEFESimulated in implement.Compare to poorFE			previously	data				better.		
Privad arashinFCFSSimulated in coud simSimple to cost, LessGoeina thet al (3]MinCloud sim and foundimplement, priornature so priorPrivad arashin i et al [2]FCFSSimulated in compare to to FCFS.Simple to thet al cost, LessCompare to thet al cost, LessGopina thet al [3]Mincloud sim and foundIf more priorpriorPrivad arashin i et al [2]FCFSSimulated in cloud simSimple to implement.Compare to PSO it has poor response time and cost.Compare to poorIf nore thatPrior thatMin thatnumber of of algorithm tasks then upon cloudResources required.Priyad arashin i et al [2]FCFSSimulated in cloud sim implement.Simple to poor response time and cost.Compare to response time and cost.Compare to response time and cost.If number of termsIf more response time and cost.If number of response time and to span.If number of response time and cost.If number of response time and t			allotted VM	transfer		Geethu	Max-	Simulated in	Simple to	Static in
Privad arashin i et al [2]FCFSSimulated in cloud sim and found data transfer to FCFS.Simulated in time compare to modified timeCompare to response time and cost.Compare to response time and cost.Min th et al [3]and found thet al [3]If more and foundprior knowledge performance of algorithm is dependedMin muther soPrivad arashin i et al [2]FCFSSimulated in cloud sim and FoundSimple to implement.Compare to poor response time and cost.Compare to poor response time and cost.Compare to poor response time and cost.Compare to poor response time and cost.Nin the t al poor response time and cost.Nin the t al poor then Max then Max poor then Max and make span.Nin Performs span.Nin the to the to the make then Max then Max then Max then Max then Max then MaxNin Performs span.Nin the to the to t			Simulated in	cost Less		Gonina	Min	cloud sim	implement	nature so
Privad and found less VM and data transfer to FCFS.Compare to Throttled[3]In the band of that thatIn three priorprior knowledge of of algorithm is dependedKnowledge of of algorithm tasks then upon cloudPresources resourcesPrivad arashin i et al [2]FCFSSimulated in cloud sim and FoundSimple to implement. poor response time compare to modified throttled.Compare to poor response time and cost.Compare to poor response time and cost.[3]In the output to that thatIn the output to knowledge performance is depended to AX Min tasks then terms of terms of terms of terms of terms of terms of terms of time and cost.[3]Privad are more modified throttled.Simple to implement. poor response time and cost.Compare to response time and cost.[4]In the output to that the Max the Max the Max the Max the Max the Max the MaxIn the output to the Max the Max <b< td=""><td></td><td></td><td>cloud sim</td><td>time</td><td></td><td>th et al</td><td></td><td>and found</td><td>If more</td><td>prior</td></b<>			cloud sim	time		th et al		and found	If more	prior
Priyad arashinFCFSSimulated in (data transfer to FCFS.Simple to implement.Compare to (arashinCompare to (arashinManuel of performance (of algorithm)Inductor of tasks then (asks then)Rubber of (of of of algorithm)Inductor of (lighter)Rubber of (of (asks then)Priyad arashin i et al [2]FCFSSimulated in implement.Simple to (poor (coud sim)Compare to (poor (poor (poor))Performs (poor)Performs (poor)required.[2]better (compare to) (compare to) (poor)poor (poor)Performs (poor)performs (poor)Performs (poor)Performs (poor)[2]better (poor)naturepoor (cost.)poor (poor)poor (poor)poor (poor)poor (poor)poor (poor)[2]better (poor)natureresponse (cost.)poor (cost.)poor (poor)poor (poor)poor (poor)poor (poor)[2]better (poor)naturecost.)poor (poor)poor)poor (poor)poor)[2]better (poor)naturecost.)poor)poor)poor)poor)[2]better (poor)naturecost.)poor)poor)poor)poor)[2]better (poor)naturecost.)poor)poor)poor)poor)[2]better (poor)naturecost.)poor)poor)poor)poor)[2] <td></td> <td></td> <td>and found</td> <td>consuming</td> <td></td> <td>[3]</td> <td></td> <td>that</td> <td>number of</td> <td>knowledge</td>			and found	consuming		[3]		that	number of	knowledge
Privad arashin i et al [2]FCFSSimulated in cost compare to FCFS.ThrottledCompare to is implement.of algorithm is dependedtasks then Max Min and task is required.Privad arashin i et al [2]FCFSSimulated in cloud sim implement.Simple to implement.Compare to poor response time and cost.Compare to implement.FCFSSimulated in implement.Compare to poor response time and cost.FCFSSimulated in implement.Compare to poor response time and cost.FCFSSimulated in implement.Compare to poor response time and cost.FCFSSimulated in implement.Compare to poor response time and cost.FCFSSimulated in implement.Compare to poor response time and cost.FCFSFCFSSimulated in poor response time and cost.FCFSFCFSFCFSFCFSFCFSSimulated in compare to modified throttled.Simple to poor response time and cost.FCFSFCFSFCFSFCFSFCFSSimulated in compare to modified throttled.Simple to cost.FCFSFCFSFCFSFCFSFCFSSimulated in compare to modified throttled.Simple to cost.FCFSFCFSFCFSFCFSFCFSFCFSSimulated in compare to modified throttled.FCFSFCFSFCFSFCFSFCFSFCFSFCFSFCFSSimulate modified throttled.FCFS			less VM and	compare to		[~]		performance	lighter	of
Privad arashinFCFSSimulated in implement.Simple to poor response time compare to modified throttled.Compare to poorSimulated in poor responseCompare to poor responseMax Min upon cloudMax Min performs required.Privad i et al [2]FCFSSimulated in implement.Simple to poor responseCompare to poor responseEnvironment.better in terms of terms of[2]better modified throttled.natureresponse cost.max max modifiedand make[2]implement poor response time throttled.cost.max max max maxand make[2]implement poor response time throttled.cost.max poor max cost.and make[2]implement poor response time throttled.cost.max poor responseand make[2]implement response time throttled.cost.max poor responseand make[3]implement response time throttled.cost.max poor responseand make[4]implement responsecost.better modified throttled.implement responsecost.[5]implement responsecost.implement responsecost.implement response[6]implement responsecost.implement responseimplement responseimplement responseimplement response[6]implement responsecost.implement 			data transfer	Throttled				of algorithm	tasks then	resources
Priyad arashin i et al [2]FCFSSimulated in cloud sim and FoundSimple to implement. poor response time compare to modified throttled.Compare to poor response time and cost.Is depended upon cloudPerforms Performs required.required.Priyad arashin i et al [2]FCFSSimulated in oloud sim and FoundSimple to poor response time and cost.Compare to poor response time and cost.If number of performs terms of terms of terms of terms of are more then Max better then Min Performs span.If number of terms			cost compare					is depended	Max Min	and task is
PriyadFCFSSimulated in cloud simSimple to implement.Compare to PSO it hasEnvironment.better in If number ofIterns ofi et aland FoundDynamic in naturepoorlighter tasksresource[2]betternatureresponsetime and cost.Min Performsspan.better then modifiedcost.better then Min Min.LoudLoud			to FCFS					upon cloud	Performs	required
PriyadFCFSSimulated in cloud simSimple to implement.Compare to PSO it hasIf number of lighter tasksterms of resource are more[2]and FoundDynamic in naturepoor responsepoor time and cost.it me and cost.it me and better then better then[2]modified throttled.cost.cost.modified throttled.it me and cost.min Performs Min Min.								Environment	hetter in	required.
arashin i et al [2]cloud sim and Foundimplement. Dynamic in naturePSO it has poor response time and cost.lighter tasks are more then Max better then better then Min Min.resource utilization then Max	Priyad	FCFS	Simulated in	Simple to	Compare to			If number of	terms of	
i et al [2] and Found better nature response time and compare to modified throttled. better better nature better	arashin		cloud sim	implement.	PSO it has			lighter tacke	resource	
[2] better nature response response time time and time and compare to cost. better then modified throttled. better	i et al		and Found	Dynamic in	poor			are more	utilization	
response time compare to modified throttled. time and cost. time and cost. throttled. th	[2]		better	nature	response			then May	and make	
compare to modified throttled. cost. better then Min Min. span.			response time		time and			Min Performs	snan	
modified throttled.			compare to		cost.			better then	Span.	
throttled.			modified					Min Min		
			throttled.							

Shridh	Hybrid	Simulated in	Better	Doesn't	Reena	Dyna	They	Improve	It doesn't
ar et al	of	CloudSim	resource	Consider	Panwar	mic	compare this	resource	consider
[4]	Divide	and it	utilization	priority of	et al	Load	algorithm	utilization,	Priority of
	and	compared	compare to	task,	[6]	Manag	with optimal	data	tasks. It
	Conqu	with	Throttled,	Doesn't		ement	VM Load	processing	doesn't
	er and	Modified	also has	consider			balancing	cost and	consider
	Modifi	Throttled	better	resource			algorithm and	response	resource
	ed	algorithm.	Execution	specific			simulate	time.	specific
	Throttl	They found	time	demand of			results in		demand of
	ed	that the		user.			cloud analyst.		task.
		proposed					It will		
		algorithm					Improves		
		reduces					response		
		execution					time, data		
		time by					processing		
		9.972% and					time and data		
		efficient load					transfer cost.		
		balancing as					It will make		
		compare to					Proper		
		Modified					resource		
		Throttled					utilization.		
		algorithm.							
					Surbhi	K-	It will be	It considers	It Doesn't
Navtaj	Combi	They	Better	Considered	Kapoor	mean	simulated on	the	Consider
Shingh	nation	simulate on	Load	limited	et al	Cluster	Cloudsim and	resource	the Priority
et al	of	CloudSim	balancing	number of	[7]	ing	result shows	specific	of Task,
[5]	Max-	and found	and	resources			that the	demand of	Load
	Min	that the total	Response	and VMs,			proposed	the task. It	balancing
	and	processing	as compare	Didn't			algorithm	has less	within
	Ant	cost and	to	considered			improves the	overnead	cluster is
	Colon	completion	Improved	cost model.			parameters	as scanning	not
	у	time of	Max min,				time Response		considered.
	Algorit	proposed					time,	done in	
	hm	algorithm is					time Make	only	
		less than that					Spop and	Cluster It	
		of Improved					span and	ciuster. It	
		max min					compare to	in in	
		algorithm.					throttled and	heterogene	
							modified		
							throttled	Environme	
							algorithm	nt. It	
								improves	
								Resource	
								utilization.	
								Response	
								time	
								-	
								,Execution	
								,Execution time, Make	

Akhil	Firefly	Simulate	Efficient in	Difficult to
Goyal		algorithm on	terms of	implement
et al		CloudSim	Energy	in
[9]		and	consumptio	heterogene
		Compared	n. Also	ous
		with PSO	response	environme
		algorithm.	time and	nt. Priority
		Result shows	resource	is not
		that the	utilization	considered.
		firefly	is good	
		algorithm has	compare to	
		less response	PSO	
		time and		
		good		
		processor		
		utilization as		
		compare to		
		PSO. The		
		energy		
		consumption		
		using firefly		
		is 10 to 14 %		
		less compare		
		to PSO.		
Ravi	Power	Simulated in	Good	Doesn't
Shanka	Aware	CloudSim	Resource	consider
r et	Resour	and	utilization,	resource
al[10]	ce	compared	minimize	specific
	allocat	with	power	demand of
	ion	DVFS(Dyna	consumptio	task , More
	Policy	mic Voltage	n in hybrid	suitable in
		and	cloud	Hybrid
		Frequency		cloud
		Scaling). The		environme
		proposed		nt.
		algorithm		
		maximizes		
		throughput		
		and has less		
		power		
		consumption.		

V. CONCLUSION

Load Balancing play an important role in cloud computing. It provides proper resource utilization and improves the response time and cost that lead the customer satisfaction. It also reduces the power consumption which is the biggest challenge in green computing. We have discussed almost fourteen load balancing algorithm and their pros and cons. The all discussed algorithm behave differently in different condition. The use of algorithms is dependent upon cloud environment, cloud size and user requirements.

V. ACKNOWLEDGMENT

I would like to thanks Prof Rikkin Thakkar for his valuable support and suggestions.

REFERENCES

- Garima Rastogi and Dr Rama Sushil, "Analytical Literature Survey on Existing Load Balancing Schemes in Cloud Computing." International Conference on Green computing and Internet of Things, 2015.
- [2] Priyadarashini Adyasha Pattanaik, Sharmistha Roy and Prasant Kumar Pattnai, "Performance Study of Some Dynamic Load Balancing Algorithms in Cloud Computing Environment." 2nd International Conference on Signal Processing and Integrated Networks (SPIN),2015,pp.619-624.
- [3] Geethu Gopinath P P, Shriram K Vasudevan," An indepth analysis and study of Load balancing techniques in cloud computing environment", In 2nd International Symposium on Big Data and Cloud Computing(ISBCC'15),2015, pp. 427-432.
- [4] Shridhar G.Domanal and G. Ram Mohana Reddy, "Load Balancing in Cloud Environment using a Novel Hybrid Scheduling Algorithm", IEEE International Conference on Cloud Computing in Emerging Markets, 2015, pp. 37-42.
- [5] Navtej singh and Rajwinder kaur, "Dynamic Combination of Improved Max-Min and Ant Colony Algorithm for Load Balancing in Cloud System", 6th ICCCNT – 2015.
- [6] Reena Panwar and Dr. Bhawna Mallick, "Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm", IEEE, 2015, pp. 773-778.
- [7] Surbhi Kapoor and Dr. Chetna Dabas, "Cluster Based Load Balancing in Cloud Computing", IEEE, 2015.
- [8] Mohamed Belkhouraf, Ali Kartit and Hassan Ouahmane, Hamza Kamal Idrissi, Zaid Kartit and Mohamed El Marraki, "A secured load balancing architecture for cloud computing based on multiple clusters", IEEE, 2015.
- [9] Akhil Goyal,Navdeep S Chahal, "Bio Inspired Approach for Load Balancing to Reduce Energy Consumption in

Cloud Data Center", International Conference on Communication, Control and Intelligent Systems (CCIS),2015.

- [10] Ravi Shankar Jha and Punit Gupta, "Power Aware Resource Allocation Policy for Hybrid Cloud", Third International Conference on Image Information Processing,2015,pp.336-341.
- [11] [11] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", Second Symposium on Network Cloud Computing and Applications, IEEE, 2012, pp. 137-142.
- [12] [12] Aya A. Salah Farrag, Safia Abbas Mahmoud, EI Sayed M. EI-Horbaty, "Intelligent Cloud Algorithms for Load Balancing problems: A Survey", iEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS'15),2015.pp-210-216