Monika Sahu et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 4, Issue 2, June 2017, pp. 165-171

Analyzing Cloud Load Balancing Algorithms & Their Comparison

Monika Sahu[1], Dr. Vivek Jaglan[2]

[1]Department of C.S.E., Bhagwant University Ajmer Rajasthan

[2]Associate Professor, Amity University, Gurugram

***Abstract*— Cloud computing is a radical innovation which gives figuring as-a-benefit as opposed to giving processing as an item. It arrangements processing assets on request by paying for part of time we are utilizing them. Idea of cloud diminishes general cost and administration endeavors through proficient usage of different assets. It gives us a stage from where different assets like handling, memory, speed, data transmission and so on can be pooled on request premise by paying for them. Distributed computing has grown excessively yet at the same time it has different research challenges like security, unwavering quality, accessibility, stack adjusting, control administration and information movement investigation and so forth which require analysts prompt consideration. Load adjusting is one of the hot research regions these days. It is the procedure of consistently dispersing workload on geologically circulated server farms. Different load adjusting calculations as of now exist for proficient usage of assets. This paper primarily concentrates on idea of load adjusting, existing burden adjusting calculations, different load adjusting measurements and future work toward this path.**

***Keywords*— Cloud Computing, Resource Scheduling, Resource Allocation**

I. INTRODUCTION

Cloud computing [1] is an innovation that gives Information Technology as a support of its clients. The customers/clients can get to the administrations from cloud condition through the web and remote servers. The remote server keeps up the information and applications for enrolled customers. Diverse sorts of clients, for example, association, buyers and organizations utilize distinctive sorts of cloud administrations, for example, information stockpiling, applications and restrictive programming without contributing tremendous sum. The design of Cloud Computing comprises of two fundamental parts got back to frond end and end associated with each other with the assistance of Internet association. The cloud goes about as back end and customer goes about as front end some portion of cloud design. Figure 1 beneath demonstrates the distinctive sorts of administrations gave by distributed computing to various sorts of clients.

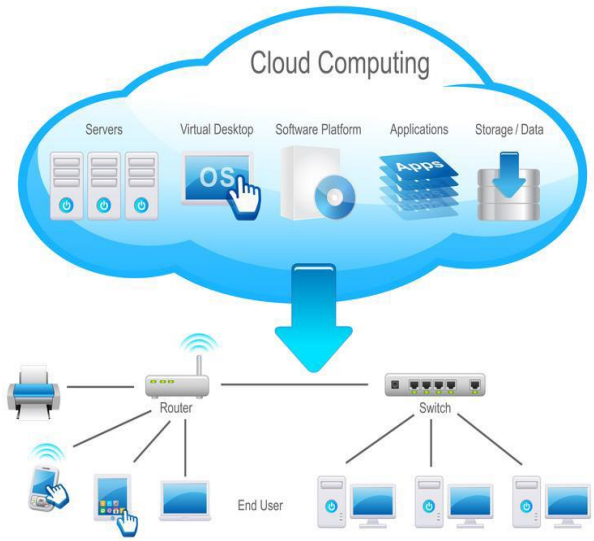

Figure 1: Cloud Computing

As appeared in figure 1 changed sorts of administrations are remote servers, virtual desktop, Software stage, applications and online stockpiling. Diverse sorts of clients can get to the above assets through their mobiles, tablet and desktop PCs in the wake of interfacing with Internet.

Cloud computing encourages us to use the offices of numerous applications, for example, web conferencing, recreations, E-mail on the Internet. With distributed computing we can make and design the applications on the web. It additionally causes us to redo the applications as per customers' need. We can store over information on the web and recover the information whenever anyplace. It evacuates the prerequisite for customers/clients to be in an indistinguishable area from the genuine equipment that stores information or data. A client can get to assets/administrations of distributed computing utilizing different equipment gadgets subsequent to making the association with the Internet [2].

Cloud gives us colossal capacity and offers distinctive sorts of assets, for example, rapid server, runtime stage, applications and programming. Because of shortage and huge cost of cloud assets there is requirement for legitimate booking and distribution of cloud assets. The cloud assets must be designated in productive and compelling way with the goal that aggregate cost of assets ought to be limited. This paper gives audit of various cloud planning techniques utilized as a part of cloud condition. It likewise give presentation of proposed planning approach with the goal that rare assets of cloud condition are dispensed in productive, compelling and temperate way.

II. RESOURCE SCHEDULING & ALLOCATION

Two fundamental performers include in distributed computing are cloud suppliers and the cloud clients. Mists suppliers i.e., cloud itself build up the cloud server farms and different assets to be utilized by cloud clients. Cloud clients i.e., end clients can really utilize the cloud assets and pay as indicated by their use. The essential correspondence

(connection) between cloud suppliers and cloud clients can be effortlessly comprehended utilizing Figure 2 underneath [3].

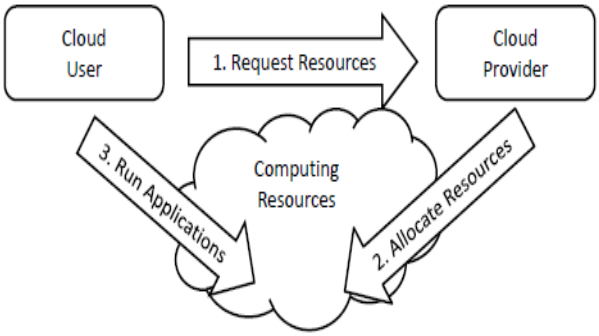

Figure 2: Interaction between cloud actors

The communication steps are recorded underneath:

1. The cloud client starts a demand for a particular assets to the cloud supplier.
2. On getting the demand the cloud supplier checks for the accessibility of particular asset.
3. If asset exists at that point dole out the asset to the asking for client.
4. The client now uses the administrations of relegated assets to play out a particular assignment or application.
5. When no more administration is required then the client discharges the asset, pay for the asset and shuts the association.
6. The supplier now plan and allot the asset to other asking for customers. [4, 5, 6].

One intriguing part of the distributed computing condition is that these on-screen characters or say players are by and large from various association and districts with their own particular need and interests. "The fundamental objective of cloud suppliers is to create however much income as could reasonably be expected with least speculation on cloud framework." To accomplish this target the cloud suppliers have different virtual machines to be utilized by numerous customers to achieve most extreme benefit.

Vitality proficient Cloud assets portion comprises in distinguishing and relegating assets to every approaching client ask for in such a way, that the client necessities are met, that the slightest conceivable number of assets is utilized and that server farm vitality productivity is enhanced. Figure 3 demonstrates the asset designation and planning plan for distributed computing.

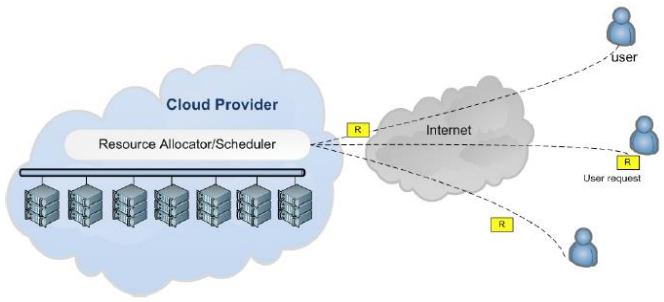

Figure 3: Resource Allocation & scheduling in Cloud Computing

A. Resource Allocation

In [7] creators stated "Asset distribution includes choosing what, what number of, where, and when to make the asset accessible to the client. Normally, clients choose the sort and measure of the asset compartments to ask for then suppliers put the asked for asset holders onto hubs in their datacenters. To run the application effectively, the sort of asset compartment should be very much coordinated to the workload qualities, and the sum ought to be adequate to meet the imperatives i.e., work must be finished before its due date. In a flexible domain like the Cloud where clients can demand or return assets powerfully, it is additionally critical to consider when to make such alterations."

B. Job Scheduling

In [8] creators stated "Once the asset holders are given to the client, the application settles on a booking choice. Much of the time, the application comprises of numerous occupations to which the apportioned assets are given. The activity scheduler is in charge of allotting favored assets to a specific employment with the goal that the general processing assets are used adequately. The application likewise needs to ensure each activity is given sufficient measure of assets, or what's coming to its. Such a planning choice turns out to be more perplexing if the earth is heterogeneous."

III. Literature Review

Existing load balancing algorithms

In existing writing different load adjusting calculations exist which are utilized as a part of various situations. Every calculation has a few advantages and disadvantages. Which calculation is utilized as a part of a specific circumstance it relies on different variables like size of cloud, kind of demand, measure of accessible assets and limit of hubs and so forth. Load adjusting calculations are arranged in different routes as talked about above, however extensively they are characterized in two classes static and dynamic.

Static Algorithms

Different static load adjusting calculations effectively existing in writing are examined here.

First Come First Serve

In FCFS forms are executed by their entry time. To start with process entering framework is executed first. This calculation is exceptionally easy to execute, non-versatile and non-preemptive approach. At the point when another activity touches base in framework it needs to sit tight for finishing of as of now executing process. Be that as it may, it doesn't consider some other criteria with the exception of landing time for execution assessment [24].

- Min-Min

In Min-Min calculation least execution time of all undertakings for a specific asset is figured, at that point least

of all least execution times is chosen and this base time is utilized for designation of assignment on comparing machine. Execution time of every single other undertaking for that machine is refreshed by including least execution time of this assignment to their execution time and this undertaking is expelled from rundown of errands to be executed on that machine. This procedure is proceeded until the point that all assignments are proficient.
Preferred standpoint of this calculation is that it tries to execute shorter occupations right off the bat. Be that as it may, now and then this calculation prompts starvation [25].

 Max-Min

Max-Min calculation is like Min-Min calculation aside from that most extreme time rather than least time among least execution times of all errands is chosen for assignment of undertaking on a specific machine. Execution time of every other errand for that machine is refreshed by including execution time of this assignment to their execution time and this errand is expelled from rundown of undertakings to be executed on that machine. This procedure is proceeded until the point that all errands are proficient [26]. This calculation enhances makespan however doesn't consider QoS factor.

 Load adjusting Min-Min (LBMM)

This calculation utilizes Min-Min calculation as its premise. LBMM has been created to defeat the restriction of Min-Min calculation. Min-Min calculation considers least execution time of each errand however it doesn't consider add up to heap of every hub. Because of which a few hubs get over-burden and other under stacked. LBMM expands reasonableness in stack adjusting [27].

 Opportunistic Load Balancing (OLB)

Entrepreneurial load adjusting calculation principle objective is to keep every hub occupied. This calculation relegates assignment to presently accessible hub arbitrarily without considering its present workload and execution time coming about into vast execution time. Assuming more than one hub is accessible, one hub is chosen subjectively. Be that as it may, this calculation is exceptionally basic and gives enhanced asset usage for unforeseen errands by appointing them to the hubs haphazardly [27].

 OLB+LBMM

This is a progressive approach which consolidates the advantages of both OLB and LBMM to enhance framework execution. OLB calculation keeps every hub occupied and LBMM executes each assignment in least time in this way influencing general consummation to time least. In this two stage stack adjusting calculation, first stage utilizes OLB to appoint undertakings to benefit administrator and second stage employments LBMM to disseminate subtasks to appropriate hubs by considering least execution time of each subtask [27]. Constraint of this calculation is that it can't handle CPU bound and information yield bound procedures.

 Minimum Execution Time

This calculation doles out the undertakings haphazardly to the hubs which seem to execute assignment in least time without considering current heap of framework. This calculation doesn't consider current heap of framework which may lead stack awkwardness [27].

 Minimum finishing Time

This calculation doles out the assignments arbitrarily to the hubs with least consummation time. This calculation is superior to least execution time calculation. It considers both fruition time and additionally current heap of the hub [27].

 Round Robin

In this heap adjusting calculation undertakings are alloted to the hubs in roundabout request by utilizing recipe i= (i+1) mod n, where n is add up to number of hubs and I is the chosen hub. To start with hub is picked indiscriminately and others are chosen in roundabout form. This calculation does not consider current heap of the framework and may cause stack awkwardness. A few hubs get over-burden and others under stacked. This calculation functions admirably when all hubs have same load. Be that as it may, it"s not a decent decision when hubs have extraordinary stack [28].

 Weighted Round Robin

Weighted Round Robin calculation conquers the restrictions of Round Robin calculations by doling out weights to the hubs in light of their handling power. Essential idea of this calculation is same as that of Round Robin with weights appointed to all hubs. This calculation considers current load status of every hub and hubs with high weight handle a bigger number of undertakings than the hubs with bring down weight [28].

 Randomized calculation

This calculation allots the errands to hubs arbitrarily without considering any parameter. This calculation here and there prompts stack lopsidedness as it doesn"t consider present and past heap of the framework. It is useful for circumstances when all hubs have measure up to measure of load. It is a basic calculation without between process correspondences [29].

 Threshold Algorithm

Fundamental point of this calculation is to decrease between process correspondence. At the point when a hub is made, stack is appointed to that hub promptly. Hubs are chosen locally for assignment of errands. Hubs having load are ordered into three classes over-burden, under-stacked and medium. Two limits for characterizing load are tupper and tlower.
Overloaded>tupper
tlower<Medium<tupper
Under-loaded<tlower

At first, all hubs are in under-stacked state. At whatever point a demand arrives it is satisfied locally until the point when it surpasses limits. Whenever limits are crossed it sends messages to remote hubs and illuminates them about current load status. A remote hub is then chosen for assignment allotment. Preferred standpoint of this calculation is that it expels overheads of between process correspondence and remote process memory access with enhanced execution [29].

Dynamic Algorithms

Dynamic calculations consider current heap of framework and also stack at run time to make effective load adjusting. These calculations are appropriate for heterogeneous condition and adjust as indicated by changing necessities at run time. These calculations make productive usage of assets while including different overheads like high between process correspondences and expanded multifaceted nature. Dynamic calculations are intricate in nature and in addition hard to actualize. Different dynamic load adjusting calculations effectively existing in writing are examined here.

 Shortest Job First (SJF)

SJF is least complex and non-preemptive type of load adjusting. This calculation appoints need to the occupations in view of their sizes and right off the bat apportions assets to the activity with least size. Key of this calculation is size of the activity in light of which stack is dispersed on various hubs. This is a non-preemptive approach implies once a procedure has been distributed, it won't leave assets some time recently its finish. This calculation enhances framework execution by giving lower reaction time and pivot time as analyzed to round-robin and FCFS [30].

 Equally Spread Current Execution Load (ESCE)
This calculation keeps up a line for approaching occupations and a rundown of undertakings on every single virtual machine. The data about heap of virtual machine is utilized to choose its status as ordinary, over-burden or under-stacked. In light of status of the virtual machines assignments are allotted to the slightest stacked machine from work line by work chief [31]. In the event that a virtual machine is over-burden and another is under-stacked, this calculation exchanges stack from over-burden machine to under-stacked machine for appropriate use of assets. Assignments are spread consistently finished every single virtual machine, so this calculation got name similarly spread current execution [32]. This calculation gives proficient utilization of assets with computational overheads.

 Throttled

Throttled calculation keeps up a file list for keeping data about VM‟s. This table has two sections one for keeping VM‟s id and second to store their status data. VM‟s status can be accessible or occupied and at first all VM‟s status are accessible. At whatever point client ask for arrives, it is gotten by datacenter controller and exchanged to stack balancer to choose fitting VM from record list. Load balancer checks entire record list for nothing virtual machine that can meet user‟s prerequisites also, returns chose VM‟s id to the datacenter controller. On the off chance that no VM is discovered free, stack balancer returns - 1 to datacenter and datacenter include the activity in work line. Datacenter at that point distributes occupation to the chose VM and give this data back to stack balancer. Load balancer at that point refreshes its file table for status of virtual machine. On the off chance that the datacenter faces any issue in allotting employment to VM, it returns negative input to stack balancer and load balancer doesn't refresh file table [32]. This is one of the effective calculations yet it works in same equipment setups of every virtual machine. Improved Throttle Algorithm
Improved throttled overcomes limitations of throttled algorithm by considering different hardware configurations of VM‟s.
Due to different hardware configurations, VM‟s have different capacities. Improved throttled algorithm uses more runtime parameters like response time and current load of system for job allocation to VM‟s. Its overall working is same as that of throttled but it provides minimum overheads. Improved throttled provides better performance by reducing no of request rejections [32].
 Biased Random Sampling
This calculation utilizes a chart for stack adjusting, where hubs speak to servers and edge headings speak to accessibility of assets. An internal edge speaks to a free asset and outward speaks to apportioned one. Distribution of occupation to a server is spoken to by evacuation of internal edge and fulfillment of employment by server is spoken to by formation of internal edge. Employment designation is done through irregular inspecting. An irregular walk is performed and it begins from hub which gets ask. At each stage a new hub is chosen from neighborhood. Lastly chose hub is utilized for work execution. A limit esteem is set for walk length. In the event that walk length is more prominent than edge, work is executed by that hub generally walk length is expanded through arbitrary examining. This calculation gives enhanced execution and can be enhanced more by biasing towards hubs with a few particular attributes [33]. Constraint of this calculation is that it can't function admirably in heterogeneous populace [25].

 Active Clustering

Dynamic grouping calculation depends on standard of joining the like hubs together. The principle thought is to consolidate all hubs together having comparative qualities. Technique of this calculation is as at first a hub is assigned haphazardly as initiator.
Initiator at that point chooses a relational arranger from its neighborhood with condition that go between ought to be of various sort. This relational arranger at that point frames a connection amongst initiator and matchmaker's neighbor whose sort is same as that of initiator. Go between at that point dispenses with it connect from initiator. This calculation give better throughput because of colossal accessibility of hubs [33].
 Honey Bee Foraging

This is a nature propelled delicate figuring way to deal with tackle stack adjusting issue in view of organic conduct of bumble bees. This is a decentralized load adjusting approach. In bumble bee rummaging honey bees search for the nourishment source at whatever point they found the nourishment they return back to their hives and advise different honey bees through waggle move. The force and day and age of waggle move decides quality and amount of nourishment source. Subsequent to gaining from waggle move bumble bees either abuse a similar sustenance source in the event that it is accessible or they scout for another nourishment source [24]. For stack adjusting at runtime, different servers are gathered together under virtual server (VS) each having its own particular administration line. At whatever point a server forms a demand from its line, it ascertains benefit of serving demand. This benefit relates to nature of nectar appeared in waggle move by bumble bees. High estimation of benefit implies utilize same hotspot for additionally undertaking allotments generally another server is looked. This calculation is appropriate in heterogeneous condition and accomplishes effective asset usage. Fundamental weakness of this calculation is that expansion in stack measure prompts corruption in execution [25] [23]. This calculation works just when framework is in adjusted state else it sits tight for framework to accomplish adjusted state [25].

 Genetic Algorithm (GA)

GA is a developmental system in view of primary of characteristic choice. This calculation is utilized to take care of improvement issues. GA for the most part comprises of three operations: determination, hybrid and transformation. For cloud stack adjusting GA is utilized to discover ideal processors for designation of undertakings. In GA most importantly an underlying populace is chosen indiscriminately and the individual individuals from populace are known as chromosomes. These chromosomes are in encoded shape which can be parallel, tree or numeric encoding. At that point Fitness estimation of entire populace is computed and chromosomes with high wellness are chosen for mating pool. At that point hybrid operation is connected on chosen chromosomes for producing new posterity. Single point hybrid is utilized of course. After hybrid transformation operation is connected to enhance nature of posterity and default change likelihood is 0.05. After transformation operation new posterity is acquired and its wellness is assessed. This procedure is rehashed until posterity with wanted qualities is gotten [19] [26]. This calculation limits makespan while enhancing QoS. In any case, detriment of this calculation is that it might fall in neighborhood optima.

 Improved GA

Fundamental GA has an impediment that after age of populace it considers all chromosomes of populace for producing posterity whether they are fit or not, which prompts increment in stack and also execution time of hubs. To beat this issue Improved GA has been proposed. It's essential working is same as that of GA aside from that it decreases beginning populace. After populace age, fittest people are chosen for following stage. And afterward hybrid and change operations are connected on chose chromosomes bringing about decreased cost and execution time [35]. 2

 Ant Colony Optimization (ACO)

This is heuristic approach for stack adjusting. Fundamental idea of ACO is that ants move looking for sustenance and store pheromone on its pathway and in the wake of discovering nourishment source moves in reverse bearing, likewise saving pheromone on its way back. Different ants take after its way by detecting pheromone affidavit on the way. In this way the likelihood of a subterranean insect picking a way is proportional to grouping of pheromone [27]. A similar idea is utilized for stack adjusting in cloud. Ants create from the ace hub and move in various ways to think about over-burden and under stacked hubs in cloud. This data about load status is refreshed in database instantly. While moving if the insect finds an under stacked hub, it will proceed with its development in forward heading, else it will begin moving in reverse course to the past hub [24] [28]. The subterranean insect gets executed at whatever point it finds the objective hub.

 Ant Lion Optimizer (ALO)

ALO is nature propelled calculation in view of conduct of swarm knowledge. This calculation considers two creatures' antlions and ants. Ants influence arbitrary strolls and antlions to chase for ants. ALO has five stages 1) irregular stroll of ants 2) working of traps by antlions 3) catching ants in trap 4) finding targets 5) reconstructing trap. In this calculation ants move in seek space utilizing irregular walk and antlions burrow pits for getting ants. These antlions sit at base of pits and sit tight for ants. The edges of pits are sharp enough to fall at base of pit. Antlions devour got ants and reproduce pit for next prey. Amid irregular stroll of ants, their wellness is assessed and if any subterranean insect discovered superior to antlions, their positions are traded [29]. Primary favorable position of ALO is that it can deal with huge inquiry space and stays away from nearby optima [19].

 Stochastic Hill Climbing (SHC)

Stochastic Hill Climbing is variety of Hill Climbing calculation. It's a unified, delicate registering approach. Essentially two techniques are utilized for taking care of advancement issues finish and deficient strategies. Finish strategies certification to give arrangement of resolvable issues however require exponential time in most pessimistic scenario. Then again inadequate techniques don't ensure to take care of all issues yet give attractive outcomes for every resolvable issue. SHC is one of the inadequate issues. SHC is the nearby streamlining issue and consistently moves in bearing of expanding esteem. Subsequent to coming to at top esteem (most astounding from all neighbors) this calculation stops. SHC chooses a move among numerous moves by considering prospect of slopes steepness. Wellness of each

move is assessed by a few criteria to move towards substantial move. This procedure is rehashed until halting criteria is fulfilled or arrangement is gotten [30]. This calculation indicates better reaction time when contrasted with FCFS and RR.

 Multi Cluster

It's a semi-concentrated engineering with bunches of assets. This engineering comprises of three sections stack balancer, group of assets and confirmation component. Load balancer comprises of two sorts of balancer principle stack balancer (MLB) and neighborhood stack balancer (LLB). Primary load balancer is focal controlling expert for adjusting load on various group of assets. At whatever point a customer ask for arrives then MLB dispense it to proper bunch of assets while keeping up a table of groups alongside their handling abilities. LLB is for adjusting the heap locally inside a group. MLB maps solicitations to various bunches and LLB maps it to a specific asset inside group. LLB keeps up an occupation line for up and coming employments and asset table for keeping data about accessibility of assets. LLB perform reasonable load adjusting by considering weight of every hub. MLB, LLB together perform productive load adjusting by wiping out single purpose of disappointment. On the off chance that MLB falls flat anytime of time, the principal LLB mindful about disappointment can deal with its work and passes this data about hub inability to different hubs. Verification component is for security reason. Just validated customers can take a shot at this engineering. This heap adjusting design is unpredictable yet gives greater security and accessibility [22].

IV. CONCLUSION

These days utilization of cloud is expanding at exponential rate. Different huge organizations are now utilizing cloud for various assets without taking any cerebral pain of their execution. Web-based social networking destinations like Facebook, Twitter are generally utilizing cloud for putting away information, pictures, recordings and so forth. Indeed, even little organizations are heading towards cloud for different administrations. Because of wide utilization of cloud, information on cloud is additionally expanding exponentially and we require a few measures to deal with this information. Different load adjusting calculations have been proposed as of now. In this paper we have talked about different effectively existing static and dynamic load adjusting calculations. Static calculations are basic, simple to actualize and gives great outcomes in a few circumstances. In any case, they are not all that much predominant these days since they rely on past condition of framework and not consider current parameters. Then again dynamic calculations consider current situation with framework and give ideal load adjusting when contrasted with static calculation. Be that as it may, they are mind boggling and hard to execute. In this paper we have talked about different dynamic load adjusting calculations, each have their own particular advantages and disadvantages. Every calculation has been produced by considering distinctive issues at the top of the priority list. So we can't say which one is better and which not, everything relies upon parameters, condition and application we are utilizing. Here and there a calculation gives better outcome when contrasted with another, occasionally not. Everything relies upon circumstance in which we are utilizing it. In future we can create a smart load adjusting calculation by joining delicate processing approaches like neural, fluffy, GA and so forth. This will give us a keen method for adjusting load on cloud with powerful use of different administrations.

REFERENCES

[1] B. Hayes, "Cloud Computing," *Commun. ACM*, vol. 51, no. 7, pp. 9–11, Jul. 2008.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," *NIST special publication*, vol. 800, no. 145, p. 7, 2011.

[3] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, "Cloud Computing, A Practical approach"

[4] D. Warneke, O. Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud", IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, pp 985 - 997, June 2011, DOI: http://doi.ieee computersociety.org/10.1109/TPDS.2011.65.

[5] V. Vinothina, Dr. R. Sridaran, Dr. Padmavathi Ganapathi, "Resource Allocation Strategies in Cloud Computing", International Journal of Advanced Computer Science and Applications [IJACSA], Vol. 3, No.6, 2012. ISSN: 2158-107X (Print), DOI: 10.14569/issn.2156-5570.

[6] Sowmya Koneru, V N Rajesh Uddandi, Satheesh Kavuri, "Resource Allocation Method using Scheduling methods for Parallel Data Processing in Cloud", International Journal of Computer Science and Information Technologies[IJCSIT], Vol. 3(4), 2012, pp 4625 - 4628 4625, ISSN: 0975-9646.

[7] Thangaraj P, Soundarrajan S, Mythili A, "Resource allocation policy for IaaS in Cloud computing", International Journal of Computer Science and Management Research, Vol 2, Issue 2, pp 1645 - 1649, February 2013, ISSN 2278-733X.

[8] Mohd Hairy Mohamaddiah, Azizol Abdullah, Shamala Subramaniam, and Masnida Hussin, "A Survey on Resource Allocation and Monitoring in Cloud Computing", International Journal of Machine Learning and Computing, Vol. 4, No. 1, February 2014.

[9] Garg, S.K., C.S. Yeo, A. Anandasivam and R. Buyya, 2009. Energy-efficient scheduling of HPC applications in cloud computing environments. Comput. Sci. Distributed, Parallel Cluster Computing.

[10] Li, B., J. Li, J. Huai, T. Wo and Q. Li et al., 2009, "EnaCloud: An energy-saving application live placement approach for cloud computing environments", Proceedings of the International Conference on Cloud Computing, Sept. 21-25, IEEE Xplore Press, Bangalore, pp: 17-24. DOI: 10.1109/CLOUD.2009.72

[11] Gupta, P.K. and N. Rakesh, 2010. "Different job scheduling methodologies for web application and web server in a cloud computing environment", Proceedings of the 3rd International Conference on Emerging Trends in Engineering and Technology, Nov.

19-21, IEEE Xplore Press, Goa, pp: 569-572. DOI: 10.1109/ICETET.2010.24

[12] Yang, B., X. Xu, F. Tan and D.H. Park, 2011, "An utilitybased job scheduling algorithm for cloud computing considering reliability factor", Proceedings of the 2011 International Conference on Cloud and Service Computing, Dec. 12-14, IEEE Xplore Press, Hong Kong, pp: 95-102. DOI: 10.1109/CSC.2011.6138559

[13] Li, J., J. Peng and W. Zhang, 2011, "A scheduling algorithm for private clouds", J. Convergence Inform. Technol.", 6: 1-9. Li, J., M. Qiu, J. Niu, W. Gao and Z. Zong et al., 2011.

[14] Sindhu, S. and S. Mukherjee, 2011, "Efficient task scheduling algorithms for cloud computing environment". Commun. Comput. Inform. Sci., 169: 79-83. DOI: 10.1007/978-3-642-22577-2_11

[15] Paul, M. and G. Sanyal, 2011, "Task-scheduling in cloud computing using credit based assignment problem", Int. J. Comput. Sci. Eng., 3: 3426-3430, 2011.

[16] Vijindra and Sudhir Shenai. A, "Survey of Scheduling Issues in Cloud Computing", 2012, ICMOC-2012, 1877-7058, Elsevier Ltd, Doi: 10.1016/j.proeng.2012.06.337, page no: 2881 – 2888.

[17] Neetu Goel, Dr. R.B. Garg, "A Comparative Study of CPU Scheduling Algorithms", International Journal of Graphics & Image Processing |Vol 2|issue 4|November 2012.

[18] Monica Gahlawat, Priyanka Sharma, "Analysis and Performance Assessment of CPU Scheduling Algorithms in Cloud using Cloud Sim", International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 5 – No. 9, July 2013.

[19] Swachil Patel, Upendra Bhoi, "Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review", International Journal Of Scientific & Technology Research VOLUME 2, ISSUE 11, NOVEMBER 2013.

[20] Vignesh V, Sendhil Kumar KS, Jaisankar N, "Resource Management and Scheduling in Cloud Environment", International Journal of Scientific and Research Publications, Volume 3, Issue 6, June 2013.

[21] Dilshad H. Khan, Prof. Deepak Kapgate, "Efficient Virtual Machine Scheduling in Cloud Computing", International Journal of Computer Science and Mobile Computing, Vol.3 Issue.5, May- 2014, pg. 444-453

[22] Lipsa Tripathy, Rasmi Ranjan Patra, "Scheduling In Cloud Computing", International Journal on Cloud Computing: Services and Architecture (IJCCSA) ,Vol. 4, No. 5, October 2014.

[23] Nima Jafari Navimipour and Farnaz Sharifi Milani, "Task Scheduling in the Cloud Computing Based on the Cuckoo Search Algorithm", *International Journal of Modeling and Optimization, Vol. 5, No. 1, February 2015.*

[24] S.Sujan, R.Kanniga Devi, "A Dynamic Scheduling Scheme for Cloud Computing", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 3, March 2015.*

[25] Sagnika Saha, Souvik Pal and Prasant Kumar Pattnaik, "A Novel Scheduling Algorithm for Cloud Computing Environment", Advances in Intelligent Systems and Computing 410, DOI 10.1007/978-81-322-2734-2_39, © Springer India 2016.

[26] Sushil Kumar Saroj, Aravendra Kumar Sharma, Sanjeev Kumar Chauhan, "A Novel CPU Scheduling with Variable Time Quantumbased on Mean Difference of Burst Time", International Conference on Computing, Communication and Automation (ICCCA2016)

[27] Akilandeswari. P and H. Srimathi, "Survey on Task Scheduling in Cloud Environment", I J C T A, 9(37) 2016, pp. 693-698 © International Science Press.

[28] Shridhar Domanal, Ram Mohana Reddy Guddeti, and Rajkumar Buyya, "A Hybrid Bio-Inspired Algorithm for Scheduling and Resource Management in Cloud Environment", IEEE Transactions On Services Computing, VOL. X, NO. X, JULY 2016.

[29] B. Rajasekar, S. K. Manigandan, "An Efficient Resource Allocation Strategies in Cloud Computing", International Journal of Innovative Research in Computer and Communication Engineering Vol. 3, Issue 2, February 2015.

[30] Sumita Bose, Jitender Kumar, "An Energy Aware Cloud Load Balancing Technique using Dynamic Placement of Virtualized Resources", Advances in Computer Science and Information Technology (ACSIT) Volume 2, Number 7; April – June, 2015 pp 81 – 86.

