Analysis of Various Optimization Algorithm of Cloud Computing

Archana Dubey
Research Scholar
Global Institute of Technology Jaipur

Mr. Rajesh Rajaan
Assistant Professor
Global Institute of Technology Jaipur

Abstract - The load balancing is carried out to offer efficient service performance for users. For this, the availability and usage of the entire cloud system is increased instead of the final purpose. Cloud date centers as compared to the requested amount as this kind of requirements are of real-time and strict. In the processing cycle, the total resource volume, that every user requested, is integrated. There is rare possibility for a resource amount for accessing the current total amount of available computing resource in a cloud data center. Thus, the maximization of presence and performance potential of the cloud data center at all the time is not required. The various load balancing cloud computing techniques are reviewed in this review paper.

Keywords - Cloud Computing, load balancing, Optimization algorithms

I. INTRODUCTION

Cloud Computing (CC) is an interesting and useful valuable research direction subsequent to utility computing, grid computing and distributed computing in recent times. The services of infrastructure, platform and software for users are attained from this platform. The CC assists the users with on-demand services through Internet. IaaS (Infrastructure as a Service) is the base of cloud environment. An enormous number of physical hosts are employed in the cloud data center to provide services to the user. This task cannot be accomplished efficiently to the physical host with huge remaining resource amount every time due to the change in the rest of the resource amount of every physical host at continuous level. A hypothesis is considered that the tasks, which the users requested, are executed every time to a physical host whose selection is done randomly [1]. This physical host is unable to handle the task in case of higher volume of resource of the submitted task in comparison with the rest of the resource volume of the selected physical host. The execution of this task is failed because of it. In case of similarity between the requested resource amount of a task and rest of the resource amount of the physical host which executes the task, the time consumption to handle the task becomes longer. If the task requests are received to cloud data center at continuous level, the imbalance of load of cloud data center is occurred. Consequently, the computing outcomes cannot response the users on time and in efficient manner.

The service performance can be ensured by executing the tasks in CC data centers in accordance with the greatest load demands to the corresponding hosts. Thus, many physical hosts are not working most of the time and thus, the computing resource is wasted. Recently, the major concern of cloud data centers is to balance the load [2]. The optimal physical host must be selected in the process of deploying tasks in effective way so that the load of cloud data centers can be balanced. Different techniques has fundamental objective to balance the load quickly using an algorithm cycle. If the optimal load balancing policy is utilized in excess to deal with the implementation issue, the reduction in efficacy and maximization of users waiting time is taken place. The load balancing is carried out to offer efficient service performance for users. For this, the availability and usage of the entire cloud system is increased instead of the final purpose. Moreover, the efficient service performance can be easily achieved due to the availability of higher volume of computing resource in cloud date centers as compared to the requested amount as this kind of requirements are of real-time and strict. In the processing cycle, the total resource volume, that every user requested, is integrated. There is rare possibility for a resource amount for accessing the current total amount of available computing resource in a cloud data center. Thus, the maximization of presence and performance potential of the cloud data center at all the time is not required [3]. There is no need to guarantee that the optimal load balancing effect have to reached subsequent to each algorithm cycle in real time till the entire system has affected the long-term optimal load balancing.A number of bio-inspired algorithms prove their efficiency in load balancing systems such as ant colony, bee colony, simulated annealing, and firefly. Simulated annealing is one of the simplest stochastic algorithms. This algorithm depends on the properties of the metal annealing cycle. A novel solution x_j is acknowledged with a probability based on the existing solution or state x_i,

$$p(x_i \to x_j | x(t) = x_i) = \frac{1}{Z} \exp\left[-\frac{1}{T(t)} \max\{0, f(x_j) - f(x_i)\} \right],$$

Here, f is the minimization of objective function. Also, Z refers to a normalization factor. Various researchers have used this algorithm to provide a solution of very complex issues. However, generating new strategies$x_j$from existing solutions may rely upon the implementation and problem to be solved. Whatever the generation method, such generations of new strategiescreate a Markov chain or, more precisely, a random walk. Hence, the main operator aims at generating new strategies through random walks, and the resulting randomization behaves as a mutation or exploratory search tool. Selection is obtained by testing whether a strategy is being improved [4]. Ant algorithms, particularly the ACO (Ant Colony Optimization) devised by M. Dorigo, imitator the foraging conduct of social ants. Primarily, most ant algorithms use pheromones as a chemical messenger and concentration of pheromone concentration indicates the quality solutions to the area of concern. With the utilization of perspective, solutions of the pheromone concentration that assigned marks to routes and paths with the more pheromone concentrations as improved answers to different queries e.g., discrete combinatorial issues. Taking a closer look at ACO [5], it is noticed that random pathway generation is predominantly mutation, whereas selection based on pheromone offers a solution for choosing quicker routes. The ant algorithms do have an explicit crossover. Following equation defines the probability of ants in a network problem at a specific node i forselecting the route from node i to node j.

$$p_{ij} = \frac{\phi_{ij}^{\alpha} d_{ij}^{\beta}}{\sum_{i,j=1}^{n} \phi_{ij}^{\alpha} d_{ij}^{\beta}}$$

Where, α > 0 and β > 0 are the affected parameters, $\emptyset_{ij}$denotes the pheromone concentration on the route amid $i$whiled_{ij}indicates the requirement of the parallel route. The choice is somewhat based on a priori knowledge of the route, therefore, distances_{ij} is generally applied to make $d_{ij} \propto$ 1/s_{ij}.However, bee algorithms generally don't use pheromone. The ABC (artificial bee colony) (ABC) optimization algorithm divides the bees in a colony into three categories namely, employed bees (forager bees), onlooker bees (observer bees), and scouts. Both scout bees and employed bees perform Randomization and are mostly mutation. Selection depends on honey or objective. Also, the explicit crossover is absent [6].

Firefly algorithm (FA) has extremely powerful and robust detection capability and uses large number of resources. In 2008, Shin-Xi Yang developed the firefly algorithm (FA). This algorithm depends on the flashing pattern and conduct of tropical fireflies. FA is simple, adaptable and user-friendly optimization algorithm. The motility of a firefly i is attracted to another, more striking (brighter) firefly $j$can be determined as follows:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha \epsilon_i^t$$

Here, the second term is the result of attraction. β_0 denotes the attractiveness at zero distance r = 0 [7]. The third term is randomization, and α represents the randomization parameter, and ϵ_i^t refers to a vector of random numbers obtained from a Gaussian distribution at time t. Some works also employ the randomization in regard to ϵ_i^tthat can be protracted to other distributions without any trouble.

II. LITERATURE REVIEW

Chang-an Ren, et.al (2019) presented a new firefly algorithm based cloud computing resource scheduling optimization scheme [8]. This work initially developed a statistical model based on VM (Virtual Resource) resource scheduling issue. Afterward, an improved firefly algorithm, termed as SFDA (selective elimination and decision domain strategy of firefly algorithm) was presented in this work by taking into account the optimum time period and load operation. This approach was used to search the best approach. Eventually, this work evaluated the efficiency of the developed scheme using the CloudSim tool. The results of the tests indicated that presented scheme was capable enough to achieve efficient scheduling solution and ensured the balanced load of VM score to satisfy the requirements of the clients.

Fatemeh Ebadifard, et.al (2018) provided an optimal scheme based on firefly algorithm to provide the solution of task scheduling issue in the cloud computing scenario [9]. The introduced algorithmic approach along with reducing the maximum task execution time also increased the use of VMs (Virtual Machines) by considering the potential of all VMs and applying the suitable technique for allocating the requests for VMs. Therefore, this work defined a behavioral framework of the task scheduling algorithm on the basis of model analyzing schemes. The tested outcomes acquired by applying the presented technique confirmed the efficacy of the presented approach in terms of resource usage and makespan.

Yi Zhang, et.al (2019) developed a new firefly algorithm (NFA) by applying two steps [10]. The first step was concerned with presenting a distance-based mapping operator according to the distance between a firefly and the brightest one so that the relationship between a firefly and a solution could be mapped. The next step made use ofacompetent heuristic for evaluating the objectives of the solution. The tested outcomes obtained by applying the aforementionedapproaches revealed that the NFAapproach performed superior to the benchmark firefly algorithmic approach and the currentmost optimal algorithm with regard to scheduling efficiency and computational efficacy.

A. Francis Saviour Devaraj, et.al (2020) presented a hybrid approach consisting firefly and IMPSO (Improved Multi-Objective Particle Swarm Optimization) scheme [11]. This approach was termed as as FIMPSO. The main objective of FF (Fierfly) algorithm was to make the search space minimum while the IMPSOapproach was concerned with identifying the improved response. The introduced FIMPSO algorithmic approach obtained average load for creating and improving the crucial strategies. The results of the extensive simulations revealed that the developed model outperformed the existing approaches in terms of performance. This algorithmic approach generated efficient results in terms of93% of memory usage, 67% of reliability, and 72% of throughputwith a make span of 148, which was better than all other compared techniques.

Mainak Adhikari, et.al (2020)proposed a new Firefly algorithm (FA) based workflow scheduling scheme that considered many contradictory objectives such as cloud servers, makespan, resource usage, and reliability [12]. The major objective of firefly algorithm was to discover an appropriate cloud server for every workflow satisfying its preferences during thebalancing of load and resourcesused by the cloud servers. The extensive simulations were carried out to evaluate the proposed scheduling approach over Google cluster traces. The results of analysis proved the supremacy of the developed approach over the standard algorithms with regard to various Quality-of-Service (QoS) metrics.

MahyaMohammadiGolchi, et.al (2019) defined Cloud computing scenario for providing the efficient services [13]. In this work, a hybrid algorithm comprising firefly and IPSO (Improved Particle Swarm Optimization) algorithms was presented for obtaining the improved average load for creating and optimizing the valuable parameters including competent resource usage and the response time of tasks correspondingly. This work also delivered some pointers to evaluate the performance of the presented hybrid approach. In the results, the presented approach performed better than its counterparts and showed flexible behavior to minimize the average load based on multi-objective optimization.

Weiwei Xia, et.al (2018) formulated the issue of joint resource distribution as a combinatory optimization issue [14]. This work provided a solution of this issue by considering three evolutionary algorithm named GA (Genetic Algorithm), ACO-GA (Ant Colony Optimization with Genetic Algorithm), and OGA (Quantum Genetic Algorithm). This work also proposed a mapping strategy between the resource allocation matrix and the chromosome of GA, ACO-GA, and QGA, searched the existing radio and cloud resource pairs according to the resource availability matrixes for ACO-GA, and encoded the difference value amid the distributed resources and the minimal resource need for QGA for decreasing the time complexity. The results of wide-ranging simulationsrevealed that the presented techniques significantly performed better than the state-of-the art algorithms in regard to certain parameters.

2.1 Comparison Table

Author	Year	Description	Outcome
Chang-an Ren, Yinzhen Huang, Qingyun Luo, Xiaocui Li	2019	Presented a new firefly algorithm-based cloud computing resource scheduling optimization scheme. This work initially developed a statistical model based on VM (Virtual Resource) resource scheduling issue.	The results of the tests indicated that presented scheme was capable enough to achieve efficient scheduling solution and ensured the balanced load of VM score to satisfy the requirements of the clients.
Fatemeh Ebadifard, Saeed Doostali, SeyedMortezaBabamir	2018	Provided an optimal scheme based on firefly algorithm to provide the solution of task scheduling issue in the cloud computing scenario.	The tested outcomes acquired by applying the presented technique confirmed the efficacy of the presented approach in terms of resource usage and makespan.
Yi Zhang, Junlong Zhou, Lulu Sun, Jingjing Mao, Jin Sun	2019	Developed a new firefly algorithm (NFA) by applying two steps. The first step was concerned with presenting a distance-based mapping operator according to the distance between a firefly and the brightest one so that the relationship between a firefly and a solution could be mapped.	The tested outcomes obtained by applying the aforementioned approaches revealed that the NFA approach performed superior to the benchmark firefly algorithmic approach and the current most optimal algorithm with regard to scheduling efficiency and computational efficacy.
A. Francis Saviour Devaraj, Mohamed Elhoseny, K. Shankar	2020	Presented a hybrid approach consisting firefly and IMPSO (Improved Multi-Objective Particle Swarm Optimization) scheme. This approach was termed as as FIMPSO.	This algorithmic approach generated efficient results in terms of 93% of memory usage, 67% of reliability, and 72% of throughput with a make span of 148, which was better than all other compared techniques.
Mainak Adhikari, Tarachand Amgoth, Satish Narayana Srirama	2020	Proposed a new Firefly algorithm (FA) based workflow scheduling scheme that considered many contradictory objectives such as cloud servers, makespan, resource usage, and reliability.	The results of analysis proved the supremacy of the developed approach over the standard algorithms with regard to various Quality-of-Service (QoS) metrics.
MahyaMohammadiGolchi, ShidehSaraeian, MehrnooshHeydari	2019	In this work, a hybrid algorithm comprising firefly and IPSO (Improved Particle Swarm Optimization) algorithms was presented for obtaining the improved average load for creating and optimizing the valuable parameters including competent resource usage and the response time of tasks correspondingly.	In the results, the presented approach performed better than its counterparts and showed flexible behavior to minimize the average load based on multi-objective optimization.

Weiwei Xia, Lianfeng Shen	2018	Formulated the issue of joint resource distribution as a combinatory optimization issue. This work provided a solution of this issue by considering three evolutionary algorithm named GA (Genetic Algorithm), ACO-GA (Ant Colony Optimization with Genetic Algorithm), and OGA (Quantum Genetic Algorithm).	The results of wide-ranging simulations revealed that the presented techniques significantly performed better than the state-of-the art algorithms in regard to certain parameters.

III. CONCLUSION

Cloud Computing (CC) is an interesting and useful valuable research direction subsequent to utility computing, grid computing and distributed computing in recent times. The services of infrastructure, platform and software for users are attained from this platform. The various optimization algorithm like ant colony, bee colony etc are reviewed in this paper for the load balancing. It is analyzed that optimization algorithms give high performance as compared to other algorithm

IV. REFERENCES

[1] Karan D. Patel, Tosal M. Bhalodia, "An Efficient Dynamic Load Balancing Algorithm for Virtual Machine in Cloud Computing", 2019, International Conference on Intelligent Computing and Control Systems (ICCS)

[2] Guilin Shao, Jiming Chen, "A Load Balancing Strategy Based on Data Correlation in Cloud Computing", 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)

[3] Pradeep Kumar Tiwari, Sandeep Joshi, "Dynamic weighted virtual machine live migration mechanism to manages load balancing in cloud computing", 2016, IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)

[4] Narayan Joshi, Ketan Kotecha, D B Choksi, Sharnil Pandya, "Implementation of Novel Load Balancing Technique in Cloud Computing Environment", 2018, International Conference on Computer Communication and Informatics (ICCCI)

[5] P. Geetha, C.R. Rene Robin, "A comparative-study of load-cloud balancing algorithms in cloud environments", 2017, International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)

[6] T. Deepa, DhanarajCheelu, "A comparative study of static and dynamic load balancing algorithms in cloud computing", 2017, International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)

[7] Hussain A Makasarwala, PrasunHazari, "Using genetic algorithm for load balancing in cloud computing", 2016, 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)

[8] Chang-an Ren, Yinzhen Huang, Qingyun Luo, Xiaocui Li, "Resource Scheduling in Cloud Computing Based on Firefly Algorithm", 2019, IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)

[9] Fatemeh Ebadifard, Saeed Doostali, SeyedMortezaBabamir, "A Firefly-based Task Scheduling Algorithm for the Cloud Computing Environment: Formal Verification and Simulation Analyses", 2018, 9th International Symposium on Telecommunications (IST)

[10] Yi Zhang, Junlong Zhou, Lulu Sun, Jingjing Mao, Jin Sun, "A Novel Firefly Algorithm for Scheduling Bag-of-

Tasks Applications Under Budget Constraints on Hybrid Clouds”, 2019, IEEE Access

[11] A. Francis Saviour Devaraj, Mohamed Elhoseny, K. Shankar, “Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments”, 2020, Journal of Parallel and Distributed Computing

[12] Mainak Adhikari, Tarachand Amgoth, Satish Narayana Srirama, “Multi-objective scheduling strategy for scientific workflows in cloud environment: A Firefly-based approach”, 2020, Applied Soft Computing

[13] MahyaMohammadiGolchi, ShidehSaraeian, MehrnooshHeydari, “A hybrid of firefly and improved particle swarm optimization algorithms for load balancing in cloud environments: Performance evaluation, 2019, Computer Networks

[14] Weiwei Xia, Lianfeng Shen, “Joint resource allocation using evolutionary algorithms in heterogeneous mobile cloud computing networks”, 2018, China Communications, Volume: 15, Issue: 8