ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

A comparative study of various optimization techniques for cloud brokerage systems

Chandradeep Bhatt,
Asst. Professor, Department of Comp. Sc. & Info. Tech., Graphic Era Hill University, Dehradun, Uttarakhand India 248002
DOI: 10.48047/jcdr.2021.12.02.92

Abstract

The term "cloud computing" refers to a concept of resource sharing that enables ubiquitous, convenient, and on-demand network access to a shared pool of configurable computer resources that are provided by commercial providers in accordance with certain service level agreements. The use of cloud computing is significant for a number of reasons, including data analysis and storage. A cloud broker acts as a go-between for their customers and the various service providers. Requests can be made by the client to the internet broker. The cloud broker is responsible for matching the client's request with the various offerings that are made available by the service's provider. The problem of cloud brokerage can be formulated as a multi-objective optimization challenge, with the following three goals in mind: decreasing the amount of time it takes to respond to requests from customers; limiting the amount of energy that is consumed; and maximizing the amount of money that is made by the cloud broker. To overcome this problem using various optimization techniques can be compared. The performance of the cloud brokerage system is compared with that the multi-objective particle swarms optimization, genetic algorithm, and random search algorithm and ant colony optimization.

Keywords: Cloud Computing, Cloud Broker, Multi-objective Optimization, Random search, genetic algorithm, ant colony optimization.

1. Introduction

Cloud computing has evolved as an effective and cost-effective way to meet the ever-increasing demands placed on information technology in today's world. This is due to the fact that the world is moving in the direction of quicker and more efficient computing processes. The term "cloud computing" refers to a form of computing that is designed to be beneficial not just to end users but also to organizational and other enterprise users. This requires the utilization of scientific procedures, which call for the processing of a significant amount of data, which can be both time-consuming and expensive.

Cloud computing is a services-oriented computing paradigm that has profoundly impacted computing by offering three web-based services. These services are known as Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS) [1]. Shared virtualized cloud resources, which are in general a collection of multiple proprietary processes housed in a virtual environment known as a virtual machine, are utilized in order to carry out the provision of these large-scale services. In a cloud context, virtualized computational

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

capabilities are used to the process of provisioning resources based on demand. From the perspective of a deployment model, cloud computing may be broken down into the following categories:

The public cloud refers to the location where cloud service providers make their information technology (IT) capabilities available to any and all customers via the internet.

A private cloud is an environment in which information technology capabilities are made available to a limited number of customers who are associated with a company. The provider of cloud services could be an internal technological organization (that is, the same organization as the end user), or it could be an independent third party.

The term "hybrid cloud" refers to a situation in which an organization creates an environment by utilizing a mix of both private and public cloud services in order to meet its needs.

The internal cloud is a subset of the private cloud model. In this model, the cloud refers to a capacity of information technology that is provided as a service by an organization to its own business operations.

The term "external cloud" refers to the capability of information technology that is provided as a service to a company but is not housed by that company's own organization. A public or private cloud that is hosted on the internet by a third party is referred to as an external cloud.

The work that has been spent on research has been directed on the lack of cloud interoperability that is a barrier to the adoption of cloud computing due to the lack of vendor involvement in the problem. It's a well-known truth that having the capability to shift workloads and data seamlessly between private and public clouds can boost performance and availability while also cutting expenses.

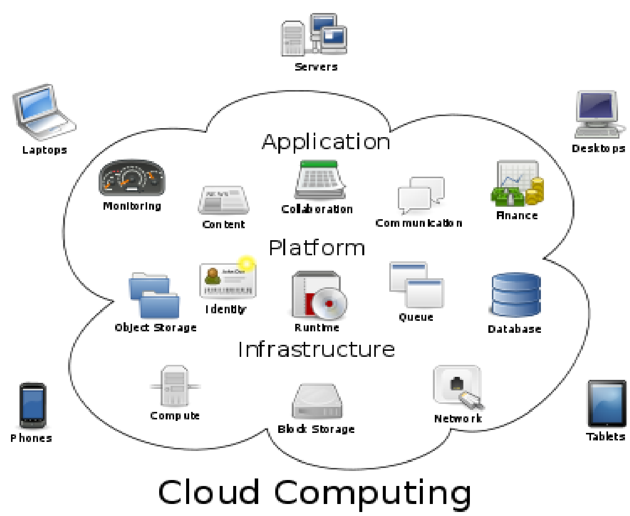

Fig1 model of cloud computing

2. Related Works:

According to k. Yildirim's [2008] description, the sensing broadcasting of a sensor network is what defines the level of supervision or tracking that an area receives from sensors. Connectivity is a crucial criteria that must be

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

met in order to demonstrate how nodes in a sensor network can efficiently communicate with one another. They have an interest in an initial deployment method that will enhance the broadcasting area of a wireless sensor network while maintaining connectivity between nodes. This is under the condition that all given hotspot zones will be covered by the sensors. [5]

A genetic algorithm technique is presented by Yu and Buyya[2010] in to handle scheduling optimization challenges in workflow applications with two quality of service constraints (deadline and budget). The utility computing model is the subject of discussion in all of the publications that have come before. In this particular paradigm, the client and the provider engage in conversation with one another in a one-on-one setting. [8]

According to Xu et al. [2010], the placement of virtual machines was stated as being defined as a multi-objective combinatorial optimization problem with the goal of simultaneously optimizing potentially competing objectives. One of the goals is to lower the costs associated with energy consumption while another is to increase the efficiency with which multifunctional resources are utilized. Methods of intelligent search are utilized in order to locate near-optimal solutions that have a reasonable runtime. In this work, it was proposed to make use of a modified evolutionary algorithm for the purpose of efficiently searching for global optimal solutions and a fuzzy-logic based assessment approach for the purpose of merging several objectives. In order to properly deal with the potentially enormous solution space for big-scale data canters, a modified genetic algorithm has been proposed and developed as a possible solution. In order to achieve a balance between the competing priorities, the algorithm employs a method known as fuzzy multi-objective assessment. The proposed method was evaluated by simulation-based studies in terms of its performance, scalability, and resilience. The comparison between the proposed method and well-known algorithms for bin-packing and methods focusing on a specific objective revealed that the proposed method exhibited greater performance.[9]

A different Cloud Intermediate Framework was proposed by Guo et al. in 2016. This framework brings together user subletting and different cloud providers. This intermediate framework is compatible with a variety of cloud services and offers consumers streaming processing services that may be accessed on demand. In the meantime, they came up with an excellent pricing approach for this intermediary architecture and gave it the name Pricing-Repurchasing. To begin, the intermediary has the ability to repurchase the sparse capacity at a dynamic rate per load. This fee is determined by the length of time and the quantity of sparse resources that the users now possess. Second, the intermediary has the ability to charge consumers varying amounts of money depending on the quantity of time and computer resources that they rent. They intend to establish an effective pricing and subletting plan for the middleman that optimizes its total revenue, taking into consideration any necessary refunds to the users. This will be done within the context of this framework. As a result, the implementation of this pricing model presents significant obstacles for the formulation of a policy that maximizes revenue for the intermediary.[11]

Sun et al. (2016) conducted a study in which they compared several cloud scheduling approaches found in the literature. They then identified the common aspects shared by all of the approaches as well as the disparities that existed between each approach based on the various cloud scheduling levels. The findings of the comparison that were provided reveal that there are significant weaknesses in the offered schedulers, almost no Pareto multi-objective solution algorithms, and very few genuine trials confronting the real cloud restrictions. [12]

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

In their study, Zhang et al. (2016) looked at three of the most fundamental QoS parameters: cost, time, and dependability. These factors are essential for a grid application, yet their individual qualities couldn't be more dissimilar from one another. [13] An application for a process can be submitted by a user, along with fundamental QoS requirements such as the bottom bound of reliability, the deadline, and the budget. In addition, the user may find that optimizing one of the QoS parameters is to their liking. Therefore, the purpose of the ACO method that has been developed is to locate a workable schedule that not only fulfills all of the user-defined QoS criteria but also optimizes the QoS parameter that is most chosen by the user. In order to accomplish this goal, seven instance-based heuristics have been developed to direct the search behavior of ants. An application for a process can first be submitted by a user in an abstract form, which corresponds to the abstract specification level. The grid system must then choose and configure the various application components of the application in order to produce an abstract workflow. The order in which tasks are carried out is specified by the abstract workflow. As was just discussed, in OSGA, the tasks that make up a workflow are given their own individual Web service instances, which are provided by GSPs. The scheduler will only select a single service instance for the task's actual execution, despite the fact that the implementation of a task may be supported by multiple service instances, each of which is given by a unique GSP and has its own unique set of quality of service attributes. As a result, the objective of the third level is to map the tasks included in the abstract workflow to the respective service instances in order to provide a concrete workflow.

The cloud broker has to find the best configuration between the clients and the cloud service providers. We use the set $U = u_1, \dots, u_n$to denote N clients and the set $S = s_1, \dots, s_n$to denote M service providers in the model. Each service provider has a limited capacity for handling the requests from clients and the total number of handling requests in the service provider needs to be greater than the number of requests from the a client. In order to describe the process of service providers, we introduce a binary variable b_{ij} with $i = 1, \dots, N \wedge j = 1, \dots M$ as follows:

Clients are expected to complete their jobs in a minimal time when they submit requests to the cloud broker and service providers. Therefore, we consider the response time of requests from clients. SetL_{ij} as the latency between clients i and service provider j. It can be measured as$L_{ij} = CT - AT$, where CT is the current time and AT is the arrival time of a request from client i at service provider j. When the service provider receives a request from a client, the service provider has to spend time T_j to execute the request. Thus, the first objective is the minimization of the response time (RT) of requests. It is formulated as follows:

$$RT = \sum_{i=1}^{N} \sum_{j=1}^{M} b_{ij} \left(L_{ij} + T_j \right) \quad (1)$$

Through the intermediary of the cloud broker, a customer submits his or her request to a service provider. The cloud broker is responsible for managing the client's needs and locating the optimal solution for those needs. In the meantime, it is anticipated that the broker will earn a profit from the undertaking. P_i is regarded to be the price from customer i, and C_j is considered to be the cost of service provider j. As a result, the profit of the brokerage is considered to be the second aim. Therefore, the second purpose is to ensure that the cloud broker generates the highest possible profit, denoted by the following formula:

$$P = \sum_{i=1}^{N} \sum_{j=1}^{M} b_{ij}(P_i - C_i) \quad (2)$$

For the service provider to successfully carry out the request made by the customer, the task must be finished with the smallest possible amount of energy used. As a result, we believe that the consumption of energy is an essential issue in the context of CC as an objective. Assume that E_j represents the amount of energy that service provider j utilized in order to carry out the task. The following is a breakdown of the overall energy consumption produced by all of the service providers:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{M} b_{ij}.E_j \quad (3)$$

The last objective is to minimize the total energy consumption of the system. With the above three objectives, the optimization problem of the cloud broker is consider as a single-objective and multi-objective problems

3. Cloud brokerage systems

The creation of a protected cloud management platform by a cloud service broker is intended to make the provisioning of complicated cloud services to cloud users more straightforward. They make it possible for customers to exploit the cloud provider's full potential in all aspects of their business. They ensure compliance with the appropriate IT standards and handle service level agreements between cloud providers in an efficient manner.

A cloud agent is another name for this specific kind of cloud broker. A broker's job is to save the buyer time by doing research on the available services from a variety of vendors and by offering information on how to achieve business goals in the most effective manner. consumers are provided with a set of application program interfaces (API) and user interfaces (UI) by a broker. These API and UI interfaces hide the technical details and make it possible for consumers to use services that are acquired from a single vendor.

Roles of Cloud Broker:

1. ***Cloud Service Intermediation:*** The value added services on top of existing cloud platforms and it is provided by the intermediation broker.
2. ***Aggregation:*** An aggregation broker ensures the interoperability and security of data between systems and brings the multiple services
3. ***Cloud Service trade:*** A cloud service trade provides flexibility and by offering multiple services

.

Cloud brokering model:

The N customers, M service providers, and one cloud broker make up the constituent parts of the cloud brokering paradigm. The cloud broker is the point of contact for requesting the customer. The cloud broker is responsible for matching the requests made by customers with the available services provided by service

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

providers.In a sense, the cloud broker anticipated being able to locate the optimal transaction between the customers and the service providers that would result in the highest possible profit. On the other side, when customers make requests of service providers, they are required to reduce the amount of time it takes for those requests to be fulfilled. In addition, a decrease in the amount of energy that is consumed by cloud computing. Because of the increasing expansion of cloud services, one of the most critical challenges is the reduction of energy use.

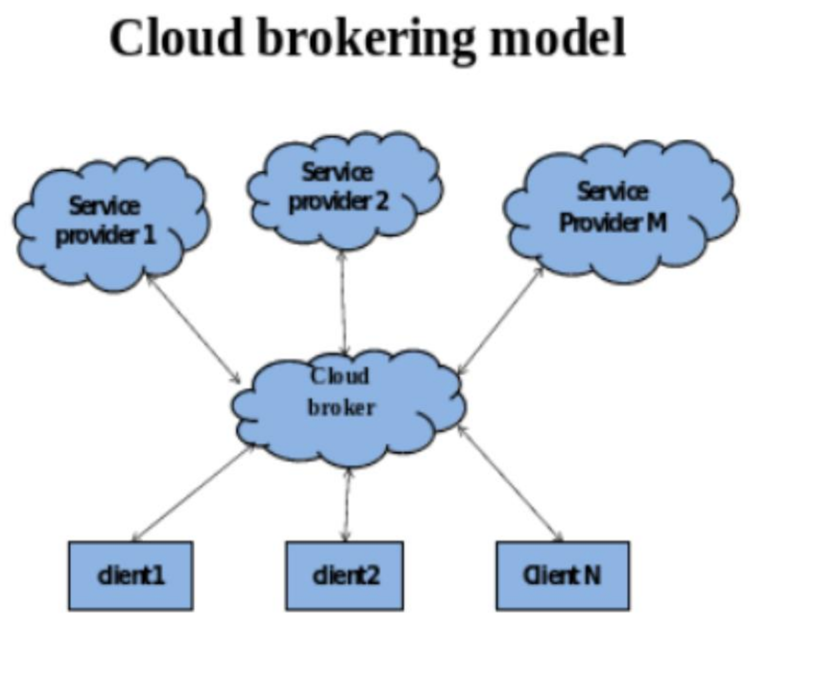

Fig 1 Cloud brokering model

2. Problem statement

In the not-too-distant future, internet connectivity will be required for an enormous number of objects and pieces of data. This will provide a formidable problem. It is necessary to establish an infrastructure in order to control and maintain such a large number of connected things. Managing the cloud broker as an intermediate of infrastructure as a service in cloud computing is necessary to get around this obstacle. In cloud computing, a request from a client can occasionally be sent in by way of the cloud broker. The cloud brokering process involves matching the requests made by customers with the offerings made by service providers. To some extent, it is the job of the cloud broker to locate the transaction that will yield the most possible profit for both the client and the service provider. On the other side, when customers make requests of service providers, they are required to reduce the amount of time it takes for those requests to be fulfilled. Additionally, a decrease in the amount of energy that is consumed by cloud computing. These issues can be discussed in this paper in order to find a solution to these issues by employing a variety of different optimization strategies to overcome these issues.

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

3. Optimization techniques

3.1 Genetic algorithm

Genetic Algorithms (GAs) are search based algorithms based on the concepts of natural selection and genetics. GAs is a subset of a much larger branch of computation known as **Evolutionary Computation**.

In computer science, there is a large set of problems, which are **NP-Hard**. What this essentially means is that, even the most powerful computing systems take a very long time (even years!) to solve that problem. In such a scenario, GAs proves to be an efficient tool to provide **usable near-optimal solutions** in a short amount of time.

3.2 Random Search Algorithm

The wind farm layout optimization problem was addressed by JuFeng et al. (2016) with the proposal of a novel multi-objective random search technique. This strategy creates a continuous variable by establishing the positions of the wind turbines, and it is able to optimize both the quantity of turbines and the placements of those turbines in the wind farm at the same time. There are two goals that are being considered. The Jensen wake model and the local wind distribution are used in conjunction with the wake effects to compute the total power production, which is one goal. The other goal is to optimize the total power output. The second goal is to cut down on the overall length of the electrical cables. Prim's algorithm is used to determine the length, which is considered to be the entire length of the smallest spanning tree that connects all of the turbines. [14]

Within the framework of this technique, populations are generated at random. It will do random checks on the details of the population. The fitness value of the particles can be verified at each iteration, and the value of fitness can be updated in each particle. It's time to bring the particles' positions and velocities up to date. Since this iteration has reached the best possible solution, the procedure can now be finished.

3.3Ant colony algorithm

Ants of certain species forage (initially) at random, and after they locate suitable food, they head back to their colony while leaving pheromone trails behind them. If more ants discover this route, it is quite likely that they will not continue to wander aimlessly but will instead continue to follow it, eventually returning to it and strengthening it if they are successful in finding food.

When one ant discovers a good (that is, short) path from the colony to the food supply, additional ants are more likely to follow that path, and positive feedback eventually leads to all of the ants following a single path. This is the overall result.

3.4 Single objective Particle Swarm Optimization (PSO)

Particle swarm optimization (**PSO**) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, and moving these particles around in the search-space according to the particle's position and velocity.

The single objective particle swarm optimization to analyze the cloud brokerage problems. It will give the optimal solution for the cloud brokerage. In the beginning, we consider the three objective s as a single objective optimization problem, and we study utility function of the cloud broker as follows;

$$\underset{U,S}{\text{minimise}} U = \omega_1 \mathrm{RT} + \omega_2 E - \omega_3 P \quad (5)$$

$$\text{Subject to } \sum_{i=1}^{N} x_{\mathrm{ij}} \leq A_j \, x_{\mathrm{lk}} \quad (6)$$

$$\sum_{i=1}^{N} b_{\mathrm{ij}} \geq R_i \quad (7)$$

$$A_j, R_i \geq 0; \text{ i=1,…..,N; j=1,…..M} \quad (8)$$

Where $\omega_{1,}$ $\omega_{2,}$ and ω_3 are the weighting factors of the response time of requests, the profit of the cloud broker and total energy consumption of the system, respectively. The sum of weights equal to one ($\omega1+\omega2+\omega_3 =1$). RT is the response time of the system that is calculated using Eq. (2). E is the total energy consumption of the system that is calculated using Eq. (4). P is the profit of the cloud broker that is calculated using Eq. (3). Hence, the optimization problem of the cloud broker is to minimize the utility function as follows:
Where Ri is the number of the requests from client i and Aj is the capacity of the service provider j. Constraint (7) indicates whether the total number of the requests from client to service provider j is less than or equal to the capacity of service provider j.

Particle swarm optimization, sometimes known as PSO, is a population-based optimization technique that was originally presented in 1995 [10]. It was conceived after the social behavior of bird swarming served as its primary source of motivation. The PSO algorithm is made up of a group of alternative answers that gradually evolve in order to get closer and closer to a practical answer to a problem. It is used to search the space of a particular problem in order to determine the parameters that are necessary in order to maximize a certain aim. It is possible to implement and apply it in order to find solutions for a variety of function optimization issues.

PSO is initialized by creating a group of random particles and then searches for optimum solution in the problem space by updating generations.

$$v_{\mathrm{lk}}(t+1) = w v_{\mathrm{lk}}(t) + c_1 r_1 [p_{\mathrm{lk}} - x_{\mathrm{lk}}(t)] + c_2 r_2 [g_{\mathrm{lk}} - x_{\mathrm{lk}}(t)] \quad (9)$$

We consider that the search space is dimensional. In every iteration, each particle is updated by following two position values, personal best (pBest), is the best position achieved so long by particle and global best (gBest), is the best position found by the neighbours of particle.

Where C_1, C_2 are the learning factors called the coefficient of the self-recognition component. w is an inertia weight .r_1 and r_2 are the random numbers that are uniformly distributed in the interval 0to 1.
After calculating the updated velocity, the positions of the particles are updated as follows:

$$x_{\mathrm{lk}}(t+1) = x_{\mathrm{lk}}(t) + v_{\mathrm{lk}}(t+1) \quad (10)$$

Algorithm 3 Procedure of single particle swarm pimization
Main
-initialize the parameter of the particle swarm
-randomly se the position and velocity of all particles
-set each particle's pBest to the particle position
-set gBest to the randomized particle position
Repeat
  For each particle
  Calculate fitness value by Eq, (5)
End
Choose the particle with the best fitness value of all particles as the gBest
  For each particle
  Calculate particle velocity by Eq.(9)
  Update particle position by Eq.(10)
End
  if fitness of the new position> pBest
then
  set pBest as a new position
  select the particle position with the best fitness value as gBest
  until the termination criterion is met
End

Where l is the number of the particles and k denoted the dimension of the particles. The PSO terminates its optimization process when the number of iterations reaches the maximum limit or the minimum error is not satisfied.

3.5 Multi objective particle swarm optimization

In multi-objective optimization problems, there are multiple goals that must be addressed all at once in order to be successful. The need to simultaneously minimize the effects of not one but two or more objective functions, which might sometimes be in direct competition with one another, arises in a great number of applications. Because of the presence of several criteria in such problems, the concept of optimality of a solution needed to be rethought, which led to the development of the idea of Pareto optimality. For instance, while optimizing the shape, distinct Pareto optimal solutions correspond to different structure configurations that have the same level of fitness but are distinguished by their individual features. Therefore, it is essential to locate the greatest possible number of such solutions, while also ensuring that there is sufficient diversity in the features that they share. This is a goal that should be given high priority.The goal of the multi-objective particle swarm optimization is to locate a set of solutions known as the Pareto set. It provides the best possible solution. The first thing that happens is a random generation of the initial swarm, and after that, a set of gBest is initialized by employing non-dominated particles from the swarm. The collection of gBest has been placed in a separate archive for safekeeping.

At each iteration, Ag Best is selected and the positions of the particles are updated. The turbulence

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

operators are applied in MOPSO after updating the position. The set of gBest is updated after all the processes of all the particles have finished. The MOPSO terminates its processes when the number of iterations reaches the maximum limit or the minimum objective function error is satisfied. It will give optimal solution for the cloud brokerage problems with the faster convergence and maximum profit can be earned in this techniques. It also can be minimizing the response time and energy consumption of the cloud broker.

Algorithm 4 Procedure of Multi-objective particle swarm optimization

Main

-initialize the parameter of the particle swarm

-randomly se the position and velocity of all particles

-set each particle's pBest to the particle position

-set gBest to the randomized particle position

-Initialize the set of gBest in an archive

Repeat

  For each particle

  select gBest

   Calculate particle velocity for (9)

   Update particle position for (10)

  calculate the fitness value for (2), (3)and(4)

  -Turbulence operators

  if fitness of the new position> pBest

then

  set pBest as a new position

  -select the particle position with the best fitness value as gBest

  -Update the set of gBest in an external archive

  until the termination criterion is met

End

3. Discussion

In cloud brokering, problems are recognized and analyzed, and a variety of optimization strategies are considered. Table 3 provides an examination of the amount of time required to complete the assignment with the assistance of a variety of different optimization strategies. The performance of various optimization strategies is compared in figure 2, which shows that this comparison is dependent on the time period.

The parameters of reaction time, energy consumption, and profit made by the cloud broker are shown to be analyzed in Tables 4, 5, and 6. These tables demonstrate that several optimization strategies were compared using

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

these factors. The iteration that is being utilized here falls anywhere between 0 and 250. Figures 3, 4, and 5 illustrate, respectively, the reaction time, energy usage, and profit made by the cloud broker at the conclusion of each iteration. According to the results of the multi-objective particle swarm optimization, the optimization strategies lead to a rise in profits while simultaneously leading to a decrease in response times and energy consumptions. MOPSO was compared to other optimization strategies, including ant colony optimization, genetic algorithms, random search, and single-objective PSO.

It demonstrates that MOPSO is the superior solution for the cloud broker's reaction time, energy usage, and profit. Random search involves selecting the particles at random, achieving slower convergence with the ant colony, and increasing the number of times the search is performed. Ant colony offers the greatest solution, despite the fact that reaching that solution requires a longer amount of time and more iterations. Both PSO and GA rely on information sharing among members (particles) of the population in order to improve their overall performance. This is another area in which PSO and GA are comparable to one another.The complexity of GA is higher, and the rate of convergence is slower. On the other hand, single-PSO has a strong global searching capability with a single objective as the utility function, but MOPSO has a faster convergence rate and provides the best optimal solution.

Table 2 Parameters

Algorithm	**Parameters**	**Values**
GA	Size of the population	20
	Probability of crossover	0.8
	Probability of mutation	0.02
	Scale for mutations	0.1
RS	Size of the population	20
	Self-recognition coefficient	1.49
	Social coefficient	1.49
	Inertia weight	0.90→ 0.1
ACO	Number of Ants	2 ↔16
	Static threshold point	0.85→ 0.95

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

	Pheromone decay parameter	0.1→ 0.4
PSO	Swarm size Self-recognition coefficient Social coefficient Inertia weight	20 1.49 1.49 0.90 to 0.1
MOPSO	Swarm size Self-recognition coefficient Social coefficient Inertia weight	20 1.49 1.49 0.9 to 0.1

Table 3 Task Vs Time

Time/Task	*ACO*	*RS*	*GA*	*SPSO*	*MOPSO*
5	*19*	*18*	*16.5*	*16*	*15*
10	*39*	*37*	*36*	*34.5*	*33*
15	*55*	*53*	*52.5*	*51*	*49*
20	*69*	*65.2*	*64*	*63*	*61*

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

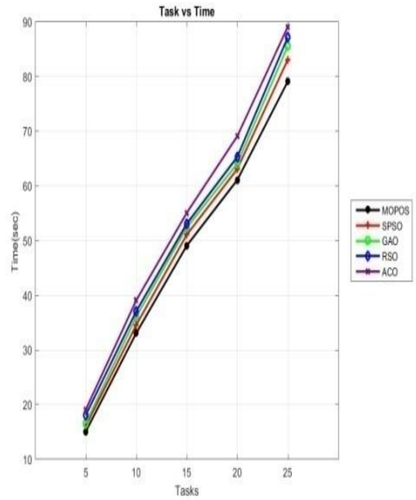

Fig 2 Time excution at the each task

Table 4 Response Time (sec) Vs Iteration

Algorithm/Iteration	***50***	***100***	***150***	***200***	***250***
MOPSO	*200*	*130*	*110*	*60*	*40*
SPSO	*230*	*180*	*160*	*90*	*60*
GA	*240*	*230*	*210*	*170*	*120*
RS	*260*	*260*	*250*	*240*	*230*
ACO	*260*	*250*	*240*	*230*	*210*

Table 5 Profit of Cloud Broker (%) Vs Iteration

Algorithm/Iteration	***50***	***100***	***150***	***200***	***250***
MOPSO	*15*	*15*	*15*	*15*	*15*
SPSO	*14*	*14*	*14*	*14*	*14*
GA	*7*	*6*	*8*	*4*	*5*
RS	*13*	*13*	*13*	*13*	*13*
ACO	*12*	*12*	*14*	*12*	*1*

Table 6 Energy Consumptions (units) Vs Iteration

Algorithm/Iteration	***50***	***100***	***150***	***200***	***250***
MOPSO	*160*	*130*	*50*	*30*	*10*
SPSO	*220*	*170*	*120*	*80*	*50*
GA	*260*	*180*	*160*	*140*	*100*
RS	*270*	*260*	*250*	*245*	*250*
ACO	*260*	*260*	*230*	*225*	*150*

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

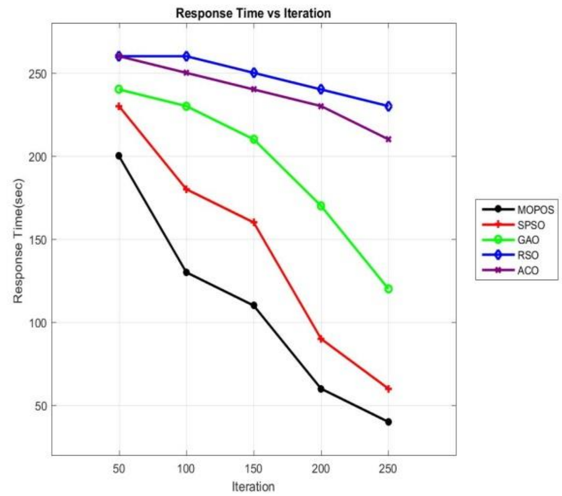

Fig 3 The response time at the end of each itertion

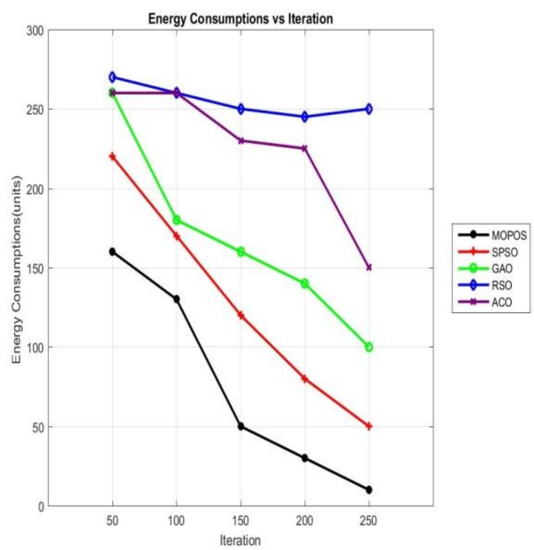

Fig 4 The energy consumption at the end of each itertion

Journal of Cardiovascular Disease Research

ISSN:0975 -3583,0976-2833 VOL12, ISSUE 02, 2021

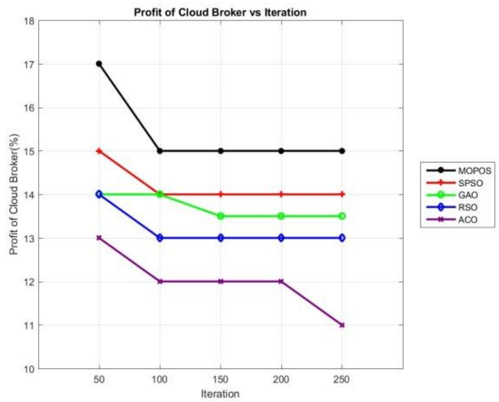

Fig 5 The profit of the cloud broker at the end of each itertion

4. Conclusion and Future work

In this method, numerous optimization strategies in cloud brokering systems in cloud environments are analyzed. A cloud broker's job could consist of choosing the service provider who offers the best possible option for optimization strategies. These optimization strategies aim to minimize response time while simultaneously reducing energy usage and maximizing the broker's profit. Comparisons are made between the genetic algorithm, the random search method, and the ant colony optimization process using the multi objective PSO algorithm. According to the findings of the comparisons, the performance of selecting cloud providers is greatly improved when multiple objective PSO algorithms are used. Experiments will be run on the system using a variety of cloud models and optimization strategies in the work that will be done in the future. The data stored in the cloud will be secured in a secure manner using various encryption methods. The profit of the broker will be increased to its maximum potential if automatic data security is provided.

References

1. [1] R. Buyya, J. Broberg, and A. M. Goscinski, Cloud computing: principles and paradigms. John Wiley & Sons, 2010.
2. K. Yildirim, T. Kalayci, and A. Ugur, “Optimizing coverage in a kcovered and connected sensor network using genetic algorithms,” inProceedings of the 9th WSEAS International Conference on Evolutionary Computing. World Scientific and Engineering Academy and Society (WSEAS), 2008
3. Z. Sun, Y. Zhang, Y. Nie, W. Wei, J. Lloret, and H. Song, “Casmoc: a novel complex alliance strategy with multi-objective optimization of coverage in wireless sensor networks,” Wireless Networks, pp. 1–22, 2016.
4. Towards Secure Cloud Bursting, Brokerage and Aggregation Srijith K. Nair[1], Sakshi Porwal[2], Theo Dimitrakos[1], Ana Juan Ferrer[3], Johan Tordsson[4], Tabassum
5. K. Yildirim, T. Kalayci, and A. Ugur, “Optimizing coverage in a kcovered and connected sensor network using genetic algorithms,” in Proceedings of the 9th WSEAS International Conference on Evolutionary Computing. World Scientific and Engineering Academy and Society (WSEAS), 2008.

6. H. Li, M. Dong, K. Ota, and M. Guo, "Pricing and repurchasing for big data processing in multi-clouds," in IEEE Transactions on Emerging Topics in Computing. 2016, 10.1109/TETC.2016.2517930.
7. J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in 2010 IEEE/ACM Int'l Conference on Green Computing and Communications (GreenCom) & Int'l Conference on Cyber, Physical and Social Computing (CPSCom), 2010
8. R. Chaukwale, S. S. Kamath. "A modified ant colony optimization algorithm with load balancing for job shop scheduling," in Proc. 201315th International Conference on Advanced Computing Technologies (ICACT), IEEE, 2013, pp. 1-5.
9. Y. Wei, Y. CHEN. "Cloud Computing Task Scheduling Model Basedon Improved Ant Colony Algorithm," in Proc. Computer Engineering,vol.41, no.2, pp. 12-16, 2015.
10. Y. Zha, Y. Li. "Task scheduling in cloud computing based on improved ant colony optimization," in Proc. Computer Engineering and Design,vol.34, no. 5, pp. 1716-1719, 2013.
11. R. Van den Bossche, K. Vanmechelen, J.Broeckhove. "Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds," in Proc. 2011 IEEE Third International Conference on Cloud Computing Technology and Science, Baltimore, MD, 7-10 Nov. 2011,pp. 320-327.
12. W. DAUN, X. FU, F. WANG, B. WANG, H. HU. "QoS constraints task scheduling based on genetic algorithm and ant colony algorithm under cloud computing environment," in Proc. Journal of Computer Applications, vol.34, no. S2, pp.66-69, 2014.
13. G. Han , W. Que , G. Jia , L. Shu . "An Efficient Virtual Machine Consolidation Scheme for Multimedia Cloud Computing," in Proc. Sensors, Vol. 16, No.2, pp. 246-258, 2016.
14. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)