International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol.8, No.1, 2021, pp.19-30.

ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

A Robust Strategic Theory based Load Balancing and Resource Allocation in Cloud Environment

M. Bala Krishna [a]

[a] *Assistant Professor of Computer Science, Maharani's Science College for Women, Mysuru, India.*
E-mail: balakrishna.mysore@gmail.com

Abstract

An on demand elastic service is cloud computing. At particular time, based on clients requirements, software, information, shared resources and other devices are provided in this. For supporting cost effective usage of computing resources, it is designed using distributed computing and virtualization advances and it enhances resource scalability. Based on requirement, business outcomes can scale up or down its resources. On-demand resource allocation challenges are created by customer demand management.

To minimize energy, Dynamic Particle Swarm Optimization (DPSO) model is developed in available research works, where CPU utilization are regulated while operating at maximum frequency. Geographically distributed resources like computers, storage etc. are owned by self interested organizations or agents are available in large-scale computing systems of cloud computing. In their own benefit, this resource allocation algorithm can be manipulated by these agents and severe degradation in performance are produced due to its selfish behaviour and its efficiency is also very poor. To solve this kind of problem, a strategy is designed for developing a resource allocation protocols with load balancer in first phase of this research work. In this agents are forced to follow the rules and tell truth. In heterogeneous distributed systems, to solve load balancing problem, a truthful strategy is designed using this strategic theory. Optimal allocation algorithm based on Improved Elephant Herd Optimization (IEHO) is proposed where a truthful payment scheme is admitted by output function which satisfies voluntary participation. Good performance is exhibited by proposed approach as indicated in experimentation results.

Keywords: Improved Elephant Herd Optimization (IEHO), Strategic Theory, Load Balancing, Resource Allocation, Cloud Computing.

DOI: 10.9756/IAJSE/V8I1/IAJSE0803
Received: 07 January 2021/Revised: 17 January 2021/Accepted: 11 February 2021/Published: 31 March 2021

Introduction

The large scale distributed computing's new paradigm is cloud computing, where data are moved away to large data centres from portable PCs and desktop [1]. Scalable IT resources like service and application, infrastructure are provided by this. Using this, they operate, over the Internet, on pay-per-use basis for adjusting capacity quickly and easily. In demand, changes can be accommodated using this and organization's capital cost in hardware and software are also avoided using this [2].

Thus, for a shared computing resources pool like services, applications, storage, servers and networks, a framework to enable required on-demand network access is given by cloud computing. With minimum service provider's interaction or management effort, these resources are de-provisioned and provisioned in a fast manner [3]. Availability is further promoted using this. Because of cloud computing's exponential growth, it adopted widely in industry and there exist a rapid data centres expansion.

In cloud computing environment, virtualization methods are adapted and resources are deployed in those environment. Different services are allowed to access these resources. Cloud computing modernizes current IT industry and based according to subscription, all resources are accessed [4].

Network based service is a cloud computing and it is classified into four classes, namely, Software-as-a-service (SaaS), Platform-as-a-service (PaaS), Database-as-a-service (DaaS) and Infrastructure-as-a-service (IaaS). Services addressing consumers directly are IaaS. Hardwares are provided by this and it can be used by organization [5]. Accessing of applications by consumers are provisioned and presented by other three services.

Any hardware or software that may be used to perform a task or a job is referred to as resource allocation. Scheduling is the act of allocating some time, resource, or slot to a specific mission. As a consequence, services are scheduled depending on the arrival of activities that must be attended to or actions taken on [6].

In any given situation, the only part we can manage is the money and their distribution, while the tasks continue to arrive outside of our control. When several tasks need the same resource in related time slots to accomplish their job, resource management is important.

There are several allocation algorithms implemented from which a system selects its own predefined algorithm for allocating resources among tasks with the least amount of conflict [7]. Turn-around time, make-span, and total wait time are examples of efficiency metrics. Shortest Career First (SJF) prioritizes the job with the shortest lifetime.

Priority Queue takes an extra parameter, priority, into account when scheduling tasks. The method of organizing, managing, and handing over available resources from the virtual server to the physical computer is known as resource scheduling [8]. Where a scheduling process necessitates more power than a single server can provide, a load balancing technique distributes workloads across many servers to improve performance.

Load balancing, on the other hand, is a data networking technique for distributing workload through multiple machines or a computer cluster, network connections, central processing units, disk drives, or other resources in order to optimize resource efficiency, mitigate reaction time, and prevent overload [9].

Instead of using a single component with load balancing, using multiple components with load balancing can improve reliability by redundancy. Dedicated software or hardware, such as a multilayer switch or a Domain Name System processor, is typically used to provide load balancing. One of the most critical facets of cloud computing is load balancing.

It's a process that uniformly distributes the diverse local workload over all nodes in the cloud, avoiding instances when certain nodes are fully loaded while others are idle or performing no work. It leads to a high customer retention and resource usage ratio [10], thus enhancing the system's overall efficiency and resource usefulness. It also means that all computational resources are spread evenly and effectively.

It also avoids system bottlenecks that can arise as a result of load mismatch. When one or more components of a program fail, load balancing aids in the service's continuity by applying fair-over, or the provisioning and de-provisioning of device instances without fail [11]. Load balancing helps to enhance efficiency by sharing the load across these different tools (network links, central processing units, disk drives).

Overloading can be avoided using this and response time, throughput enhancement and optimum resource utilization are also obtained using this. Various load balancing algorithms are used for distributing load in various systems. Two major classifications are used in load balancing algorithms, which are given by,

- Depending on how charge is distributed and how processes are allocated to nodes (system load);
- Depending on node's information status (System Topology).

It is highly required to set up a load balancing mechanism, where resources are provided based on user request for achieving their satisfaction and it provides better performance as well as better service quality. For reducing response time and for maximizing resources usage, load balancing is performed where strategic theory is used for allocating tasks to virtual machines.

The rest of the work is structured as follows: In Section 2, a detailed literature on associated work is elaborated. Section 3, specifics with proposed method and the detailed strategic theory are evaluated. Section 4, presented the results and discussion. In Section 5, outlined with conclusion and future work.

Literature Review

In cloud computing, various load balancing and resource allocation techniques disadvantages and advantages are discussed in this section.

A hybrid load balancing algorithm by combining Grey Wolves Optimization algorithms (GWO) and Teaching-Learning-Based Optimization (TLBO) is proposed by Mousavi et al [12]. Across virtual machines, loads are balanced well to maximize throughput and this can rectify the local optimum trapping problem.

On eleven test functions, benchmarked this hybrid algorithm and for verifying the results, conducted a comparative study with GWO, Biogeography-based optimization (BBO) and particle swarm optimization (PSO). In load balancing, for evaluating proposed algorithm's performance, simulated the hybrid algorithm and presented the experimental results.

A hybrid bacterial swarm optimization (BSO) algorithm is proposed by Jeyakrishnan et al [13]. Over entire search space, global seek is achieved using Particle Swarm Optimization (PSO) and BFO algorithm is used for achieving local search. For tackling a scheduling problem, a novel idea is proposed in this work using hybrid load balancing methods.

Available dynamic ADS, PSO and SA algorithms are overtaken by projected algorithms as demonstrated in experimental results. Resource utilization, make-span are maximized while minimizing operational cost using this hybrid technique.

A hierarchical edge-cloud software defined network (HECSDN) controller system design is developed by Lin et al [14] and it has three features. First, in cloud and edge, for devices, for locating its computational tasks based on traffic loads, a flexible mechanism is provided by this system architecture via computational resources sharing.

Proposed architecture's queuing model can be designed which is a second feature. Network designers are enabled to estimate design's performance quickly using networking architecture's model description. In experimental setup, it does not require more cost and time.

In HECSDN architecture, for various applications, proposed an effective load-balancing algorithm for satisfying QoS requirements and performing fairness allocation which are derived from queuing model. Even in large-scale SDN, effective performance can be produced using this newly-developed multi-tier controller system, without degrading overall performance. Moreover, in transient periods, this system highly stable even under highly fluctuating flow arrivals.

In a cloud computing environment, to solve scheduling and resource allocation problem, a simulated-annealing load balancing algorithm is presented by Fan et al [15]. Load balancing can be achieved using this technique as shown in experimental results and when compared with basic simulated-annealing and round robin algorithms, this produces a better performance.

A hierarchical control framework is presented by Leontiou et al [16] whose objective is to compromise antagonistic objectives inside a data centre. Virtual machines admission control and resource allocation problems are tackled simultaneously by local control level. Virtual machines placement in physical servers cluster and incoming request's load balancing are addressed by upper level. Towards incoming requests fluctuations, user requirements and system constraints are satisfied using these two control layer's cooperation as shown in numerical results.

An algorithm by considering various SLA parameters like required CPU time, network bandwidth, memory and Pre-emptable task execution is proposed by Pawar et al [17]. In fierce condition of resource contention, resources are utilized in a better manner using this algorithm as shown in experimental results.

For for QoS constrained resource allocation, two step process is processed by Wei et al [18]. Without considering resource assignments multiplexing, optimum problem of every participants are solved independently in the first stage. For solving independent optimization, proposed a binary Integer Programming technique. In second phase, designed an evolutionary mechanism, various participants initial optimal solution's multiplexed strategies are changed by reducing its losses in efficiency. In evolutionary mechanism, fairness and optimization are considered in algorithms. If resource allocation game has feasible solutions, then there exist a Nash equilibrium as demonstrated.

In a cloud computing environment, for task-oriented resource allocation, a model is introduced by Ergu et al [19]. Pair wise comparison matrix method is used for ranking resource allocation task. User preferences and available resources are given by Analytic Hierarchy Process. Based on task's rank, computing resources are allocated.

Further, for enhancing consistency ratio and identifying inconsistent elements, an induced bias matrix is used when conflicting weights in different tasks are assigned. For validating proposed technique, introduced two illustrative examples.

In cloud computing environment, current network resource allocation strategies critical evaluation and its possible applicability are presented by Arfeen et al [20]. Network resource allocation strategies consistent optimization and network awareness are focused in this research and identifies the issues which required further investigation by research community. In cloud computing, for network resource allocation, tailored active measurements based framework is also proposed. In shaping management strategies and resource allocation in vital applications like cloud computing as well as in computing future Internet application protocols and architecture, a dominant role is played by changing optimality criteria along with dynamic user requirements, traffic considerations and network topology as concluded.

On IaaS, according to VMs load, a dynamic resource allocation technique is proposed by Inomata et al [21] and it is abbreviated as DAIaS. Users are enabled by this technique for dynamically adding or deleting one or more instances according to user specified load and conditions. For evaluating DAIaS efficiency and effectiveness, a prototype is implemented by this. On a real cloud service like Amazon EC2, for extracting prototype, an experimentation is performed.

An adequate utility function based two-tier resource management technique is presented by Minarolli et al [22]. Utility function has local controller, where CPU shares are dynamically allocated to virtual machines for maximizing local node utility function and global controller which initiates virtual machines live migrations to other physical nodes for maximizing global system utility function. In cloud computing environments, proposed approach's benefits are shown in experimental results.

However, there are some problems in these techniques like data availability and IT resources use optimization which affects cloud performance. Some processors may be overloaded and some may be idle. Sometimes user requests a more computing power than his machine for performing its tasks while requested resource is processing other tasks and can let user waiting long time, and even more so, this CPU load can cause slowdown of other tasks and influence execution time.

It is highly critical to manage this complexity. Load distribution is assumed as a good solution to this problem. So, this work focused on tackling this problem via resource allocation based strategic theory.

Proposed Methodology

The major objective is for designing a strategy which utilizes optimum load balancing and resource allocation algorithms. Load's fraction that is allocated to every computer for minimizing expected execution time is computed using this optimal algorithm. Every computer's utilization defines incurred cost it. Every agent is asked by this strategy for reporting its processing rate and proposed optimal algorithm is used for computing allocation. Payments are handed to computers by strategy after the computation of allocation. Following are the major contribution of this research work.

1) A strategy is designed for developing a resource allocation protocols with load balancer in first phase. In this agents are forced to follow the rules and tell truth.
2) In heterogeneous distributed systems, to solve load balancing problem, a truthful strategy is designed using this strategic theory.
3) Optimal allocation algorithm based on Improved Elephant Herd Optimization (IEHO) is proposed where a truthful payment scheme is admitted by output function which satisfies voluntary participation.

Resource Management System Modeling

There is a large scaled as well as distributed data centre in every cloud provider and it includes heterogeneous physical servers. Numerous computational resources provided as a pay-per-use business models. Users are allowed to apply for a virtual machines by providers in Infrastructure-as-a-Service and they will be charged based on occupied time [23]. On a physical server, hypervisor, VMware, Xen are used for creating VM.

On VMs cluster, high performance applications are deployed by cloud users for accomplishing their missions (Map Reduce or web services jobs), which are termed as jobs in this work. Possible VM group types are offered by cloud providers in general and are used for simplifying user selection. The CPU cores count, storage size, memory size and other resources quantities are specified for defining every group.

Since there is a heterogeneity among various users regarding required VMs and it may be varied with respect to time. There is need to dynamically adjust resource allocation decision by providers. For this, in cloud, a resource management system is designed.

- **Resource Management System**

On a complex and distributed cloud system, effective as well as fair resource allocation strategy is required for resource management system. This should have centralized control and physical resources must be coordinated by this. Proposed cloud resource management system's architecture is illustrated in Figure 1. There are four components in this resource management system namely, control center (CC), infrastructure management (IM), cloud environment monitor (CEM) and register center (RC). These components are described as,

1) **RC:** For management and connection in cloud data centre, every physical server must register its information on RC.
2) **CEM:** Information like host names, physical servers IP addresses are retrieved using this component and its statuses like shutdown, running and starting are monitored by this. CPU consumption, disk storage and memory are also monitored using this.
3) **IM:** Virtualized infrastructures like virtual machines creation and releasing are managed and deployed using this.
4) **CC:** Computing Center- about resource allocating, highly appropriate decision is provided by this.

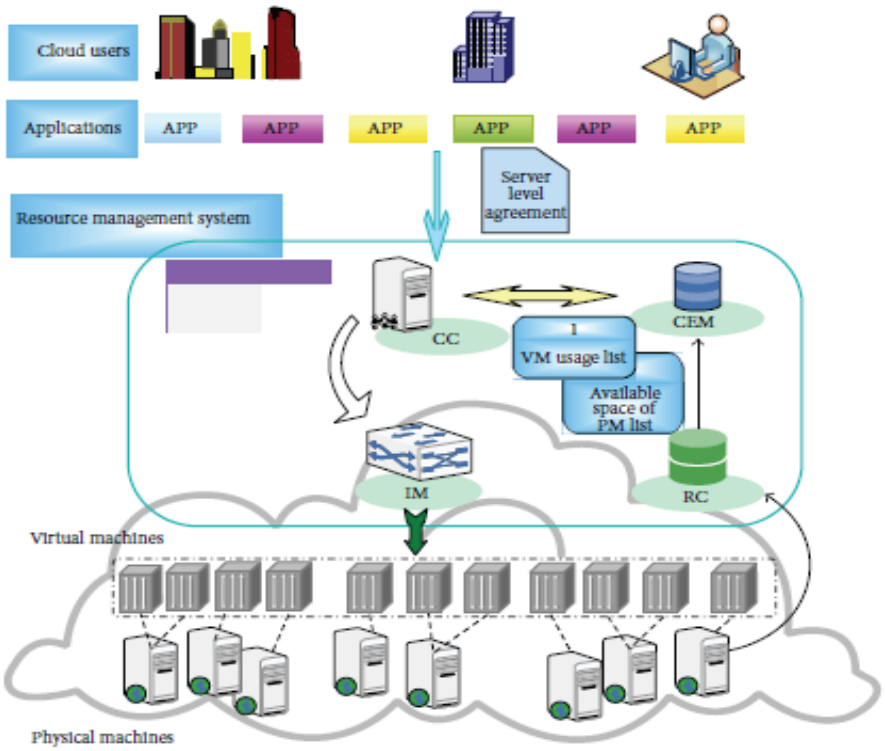


Figure 1: A Framework of Cloud Resource Management System

In RC, for physically registered servers, resource consumptions and status are monitored using CEM. If a new physical server is willing to join in cloud, information like IP, MAC address are registered in RC. If a service request is send by users to cloud, resources requirements in this request are received by CC. According to the collected information by CEM, an intelligent resource allocation decision is made by CC. For managing physical servers, IM executes allocation decision and virtual machines are placed.

In cloud environment, along with load balancing, an optimal resource allocation is proposed in this work. Load balancing problem corresponds to mapping of limited resource quantity to independent users for completing its jobs. In a time-slotted paradigm, resources are allocated in this resource management system. Current time slot's user request which is arriving dynamically are recorded and at the start of next time slot, for resource allocation, they will be served. Time slot's every start is termed as decision moment.

Strategic Design Concepts

Few important strategic design concepts are introduced in this section. For one parameter agents, strategic design problems description is concentrated in this. Distributed system model is illustrated in Figure 2. There will be some private data which represented using a single real valued parameter for every agent in this strategy design problem type.

- **Definition (Strategy Design Problem):** Following characterizes a strategy design problem of one parameter agents.
  1) Allowed outputs, finite set Λ. Output is a vector, $\lambda(b) = (\lambda_1(b), (\lambda_2(b), \ldots., (\lambda_n(b))\lambda(b)\epsilon\Lambda$, based on agents' bids $b = (b_1, b_2, \ldots..b_n)$, it is computed. Here, b_i is a value (bid) reported by agent i to strategy.
  2) There is a privately known parameter t_i termed as its true value for every agent i, (i=1,...,n). Cost incurred by every agent is based on its true value and output and is represented as $cost_i(t_i, \lambda(b))$.
  3) Profit maximization is a major objective of every agent. Agent i's profit is given by $profit_i(t_i, b) = P_i(b) - cost_i(t_i, \lambda(b))$, where, payment handed by strategy to agent i is represented as P_i.
  4) Selection of an output λ which optimizes specified cost function $g(t, \lambda)$ is a major objective of this strategy.

Following specific form of cost functions are assumed: $cost_i(t_i, \lambda(b)) = t_i, \lambda_i(b)$, i.e., t_i, indicates cost per r unit load.

- **Definition (Strategy):** Two functions are used for characterizing strategy:
  1) Output function $\lambda(b) = (\lambda_1(b), (\lambda_2(b), \ldots., (\lambda_n(b))$. Input to this function is agents' vector bids $b = b_1, b_2, \ldots \ldots b_n)$ and produces an output $\lambda\epsilon\Lambda$.
  2) Payment function $P(b) = (P_1(b), (P_2(b), \ldots., (P_n(b))$ which produces payment handed by strategy to every agent.

Notation: In remaining part of this paper, bids vector is represented as b_{-i} and it does not include agent's bid i. Vector b is expressed as (b_{-i}, b_i).

- **Definition (Truthful strategy):** if for each type t_i's agent i and for each bids b_{-i} of other agents, agent's profit is maximized when declares her real type t_i. (i.e., truth-telling is a dominant strategy).

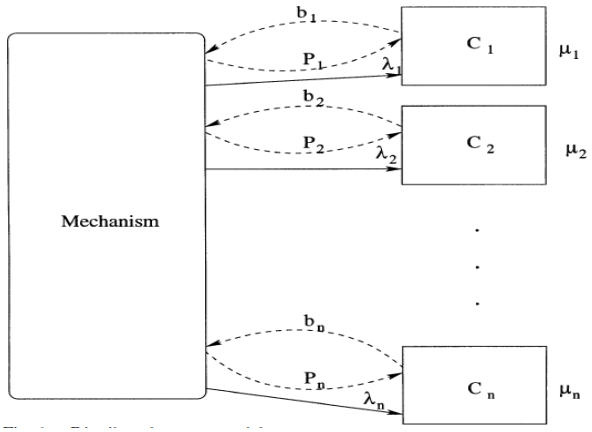


Fig. 2: Distributed System Model

- **Definition (Truthful Payment Scheme):** If there exist a payment function P, a truthful payment scheme is admitted by an output function so that strategy is truthful. Truthful agent's profit should always be a non-negative which is strategy's desirable property. By making a participation in strategy, agents are aiming for a profit.
- **Definition (Voluntary Participation strategy):** A strategy satisfies voluntary participation condition for each true values t_i, agent i, and other agents' bids b_{-i}, if $profit_i(t_i, (b_{-i}, t_i)) \geq 0$ (i.e., truthful agents never incur loss).

Major objective is for designing a truthful strategy which minimizes system's overall expected response time. Computation of allocation algorithm and payment scheme are concentrated in this work, which minimizes overall expected response time based on computer bids b_i and all computers are motivated for reporting its true values t_i.

Designing the Strategic Theory

The framework is used for designing load balancing strategy. In this, a single real valued parameter is used for representing every agent's true data. For obtaining a truthful strategy, this technique must find an output function which satisfies two conditions. Minimization of $D(\lambda)$ and bids decreasing. In addition, a strategy must be framed which satisfies voluntary participation. Payment function which satisfies voluntary participation must be computed for guaranteeing this property. Over all jobs $D(\lambda)$, output function $\lambda(b)$ which minimizes expected execution time needs to be computed at first and feasible allocation should be produced. In bids, there will be a decrease in this output function.

- **Definition (Feasible Allocation):** A feasible allocation $\lambda = (\lambda_1, (\lambda_2, \ldots\ldots, \lambda_n)$ is a load allocation and following conditions are satisfied by this:
    1) Positivity: $\lambda_i \geq 0, i = 1, \ldots\ldots, n$;
    2) Conservation: $\sum_{i=1}^{n} \lambda_i = \phi$;
    3) Stability: $\lambda_i < \mu_i, i = 1, \ldots\ldots, n$.

Following nonlinear optimization problem is solved for obtaining optimal load allocation.

$$\min_{\lambda} D(\lambda) \tag{1}$$

subject to the constraints defined by conditions i)-iii).

Over a convex feasible region defined by conditions i)-iii), convex function needs to be minimized to obtain solutions to this problem. In this case, for optimality, it is necessary and sufficient to have first order Kuhn–Tucker conditions.

Assume $\alpha \geq 0, \eta_i \geq 0, i = 1, \ldots\ldots, n$, as Lagrange multipliers. Lagrangian function is

$$L(\lambda_1, \lambda_2, \ldots\ldots, \lambda_n, \alpha, \eta_i, \ldots\ldots\eta_n) = \sum_{i=1}^{n} \lambda_i F_i(\lambda_i) - \alpha\,[\sum_{i=1}^{n} \lambda_i - \phi;] - \sum_{i=1}^{n} \eta_i \lambda_i \tag{2}$$

For this problem, optimum solution is $\lambda_i, i = 1, \ldots\ldots, n$ as implied by Kuhn–Tucker conditions, if and only if there exists $\alpha \geq 0, \eta_i \geq 0, i = 1, \ldots\ldots, n$ such that

$$\frac{\partial L}{\partial \lambda_i} = 0 \tag{3}$$

$$\frac{\partial L}{\partial \alpha} = 0 \tag{4}$$

$$\eta_i \lambda_i = 0,\ \eta_i \geq 0, \lambda_i \geq 0, i = 1, \ldots\ldots, n. \tag{5}$$

These conditions become

$$\lambda_i F_i'(\lambda_i) + F_i(\lambda_i) - \alpha - \eta_i = 0, i = 1, \ldots\ldots, n \tag{6}$$

$$\sum_{i=1}^{n} \lambda_i = \phi \tag{7}$$

$$\eta_i \lambda_i = 0,\ \eta_i \geq 0, \lambda_i \geq 0, i = 1, \ldots\ldots, n. \tag{8}$$

Optimum resource allocation algorithm is obtained by solving these conditions. For solving these conditions, proposed an Improved Elephant Herd Optimization (IEHO) algorithm in this work.

Optimization Steps to Select the Optimal Allocation Using Improved Elephant Herding Optimization (IEHO)

A novel metaheuristic nature-inspired optimization algorithm is Elephant herding optimization (EHO) and it is illustrated as follows [24]:

1) Clan Operator

In every clan, under the matriarch's leadership, all elephants live together. Therefore, in clan ci, for every elephant, matriarch ci is having influence on its next position. In clan ci, for elephant j, following expression is used for updating it.

$$x_{new,ci,j} = x_{ci,j} + \alpha \times (x_{best,ci} - x_{ci,j}) \times r \quad (9)$$

Where, in clan ci, for elephant j, newly updated position is expressed as $x_{new,ci,j}$ and its old position is expressed as $x_{ci,j}$. On $x_{ci,j}$, matriarch ci's influence is determined by a scale factor α ∈ [0, 1]. Matriarch ci is represented as $x_{best,ci}$, in clan ci, it is a fittest individual elephant and r ∈ [0, 1]. In every clan, following expression is used for updating fittest elephant.

$$x_{new,ci,j} = \beta \times (x_{center,ci}) \quad (10)$$

Where, dth dimension is indicated as 1 ≤ d ≤ D and total dimension is expressed as D. In clan ci, elephants count is represented as nci, individual elephant$x_{ci,j}$'s d th dimension is represented as d. The clan centre ci's $x_{center,ci}$ is computed using D as expressed in 10.

2) Separating Operator

At every generation, separating operator is computed as:

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1) \times rand \quad (11)$$

Where, individuals elephants position's lower bound is given by x_{min} and upper bound is expressed as x_{max}, in clan ci, worst elephant individual is given as $x_{worst,ci}$, A stochastic distribution is represented as rand ∈ [0, 1], it has uniform distribution and its value lies between 0 to 1. There are three limitations in original-+ elephant herding optimization (EHO). They are,

1) Unjustified convergence towards origin due to update operator affects exploration.
2) Unbalanced exploitation/ exploration trade-off.
3) Initial elephants position's skewed distribution is produced due to separating operator.

In original EHO, fast unjustified convergence towards origin is a major problem. For overcoming this problem, an improved elephant herding optimization (IEHO) is proposed in this work. Clan operator is updated in EEHO for achieving exploitation and exploration. In clan *ci*, individual having maximum fitness is selected as matriarch (*m*) in every generation at time *t*.

$$m_i^t = \underset{x \in x_i}{arg\,max\, F(x)} \quad (12)$$

Where, in clan *i*, individual elephants set is given by x_i.

- **Clan Updating Operator**

In clan *i*, every elephant *j* has old position $x_{i,j}^t$. Clan matriarch m_i^t , has an affect on its new position $x_{i,j}^{t+1}$ and it is given in expression (13).

$$x_{i,j}^{t+1} = x_{i,j}^t + \alpha \times \left(m_i^t - x_{i,j}^t\right) + \beta \times \left(c_i^t - x_{i,j}^t\right) + \gamma \times r \quad (13)$$

Where, scale factor is given by $\alpha \in [0,1]$, on elephant's new position, clan matriarch's influence is computed by this, another scale factor is given by $\beta \in [0,1]$, which determines elephant's tendency of moving towards clan centre. Elephant's tendency of random walk is determined using another scale factor called $\gamma \in [0,1]$. A random vector with uniform distribution is expressed as $r = (2 \times rand - 1)(x_{max} - x_{min})$, individuals elephants position's lower bound is given by x_{min} and upper bound is expressed as x_{max}. Clan's centre is expressed as c_i^t and expression 14 is used for computing this.

$$c_i^t = \frac{1}{n_i} \times \sum_j x_{i,j}^t \quad (14)$$

Original EHO uses $m^{t+1} = \beta c^t$ for matriarch update operator which produces an unjustified convergence towards origin as shown in Figure 3.

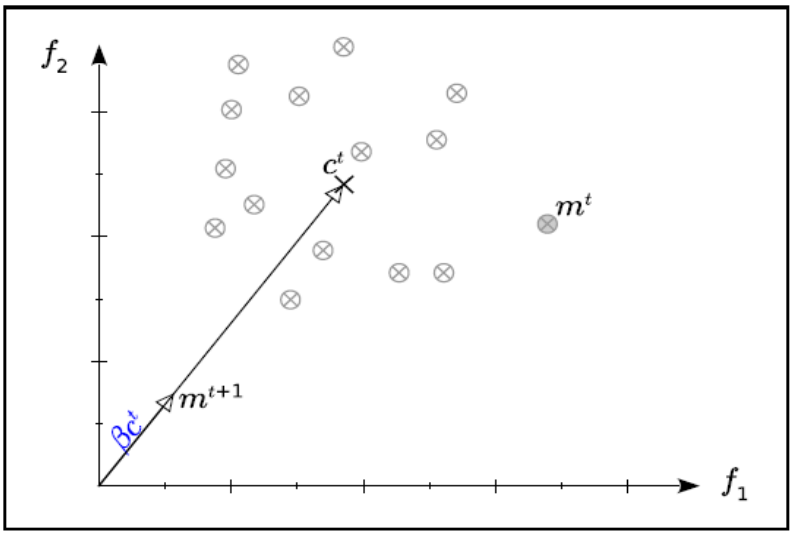


Figure 3: Biased Matriarch Update Operator in the Original EHO

An allocation $\lambda(b) = (\lambda_1(b), (\lambda_2(b), \ldots., (\lambda_n(b))$ is computed in this algorithm. For expected execution time, overall optimum is provided by this. Bids reported by computers are used for computing this optimum. This optimum may not be same as "true optimum" computed when all computers declare its true values, if different values other than its true values $(t_i = \frac{1}{\mu_i})$ are declared by some computers. Therefore, worst performance is exhibited if some computers lies. Overall expected execution time will be high in those condition and output function is minimized, expressed as OPTIM. Now, a truthful strategy is admitted by this output function.

Results and Discussion

On a Dell Optiplex9010 with JDK 1.7, simulations are run. Following assumptions are made for minimizing simulation complexity. In simulations, two resource types namely memory and CPU are considered. Various resources, predicted maximum consumption is indicated using every job request submitted by a user and VMs cluster of same type are used for handling those requests. For every time slot, total resources amount provided are estimated previously by cloud provider.

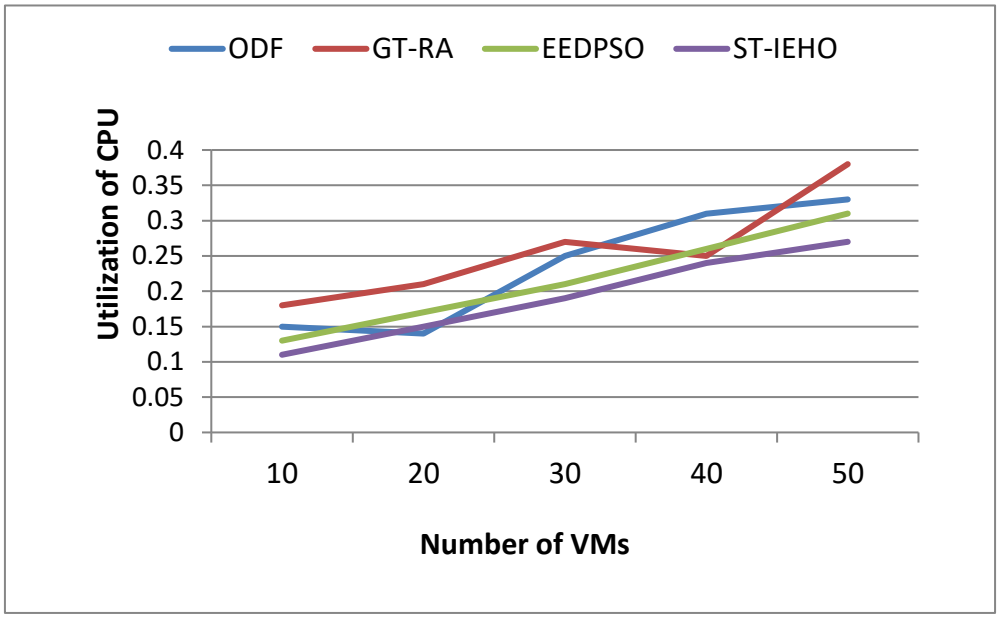


Figure 4: Comparison of CPU Utilization between the Proposed and Existing Methods

Resource utilization comparison of GT-RA, ODF, EEDPSO and proposed ST-IEHO is shown in Figure 4. Best CPU utilization and resource utilization are provided by proposed ST-IEHO scheme. Cloud data centre's energy efficiency can be enhanced using this proposed scheme.

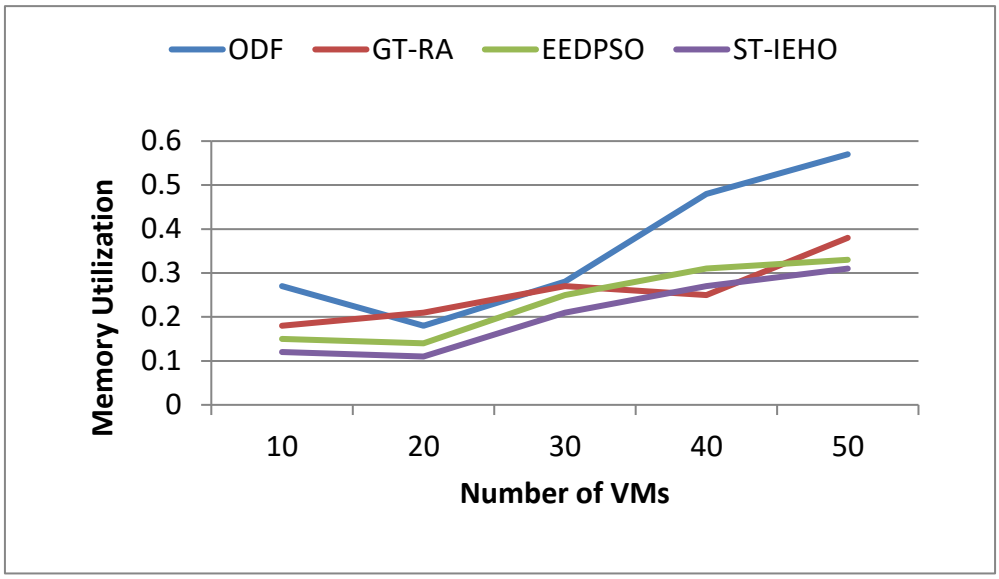


Figure 5: Comparison of Memory Utilization between the Proposed and Existing Methods

Increase in virtual machines order increases memory utilization as shown in Figure 5. In traditional heuristic algorithms like GT-AR, ODF and EEDPSO techniques, there is a linear variation in memory utilization as shown in that figure. With same virtual machines count, high energy efficiency is given by proposed ST-IEHO, when compared with GT-AR, ODF and EEDPSO techniques.

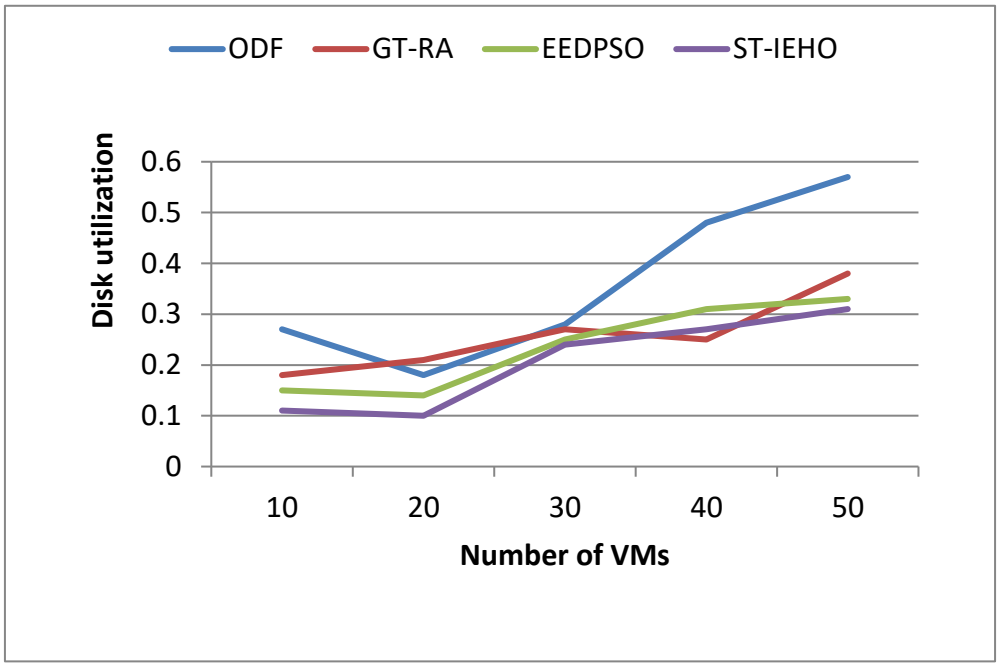


Figure 6: Disk Utilization between the Proposed and Existing Methods

Using these four algorithms, investigated the disk usage. Figure 6 shows the results of it. When compared with GT-RA, ODF and EEDPSO, better energy efficiency is exhibited by proposed ST-IEHO technique which is highly important parameter of cloud service provider.

Conclusion

For resource allocation with selfish agents, designing protocols problems are investigated in this research work. In strategic design theory, solving this kind of problem is a major objective. In heterogeneous distributed systems, to solve load balancing problem, a truthful strategy is designed using this theory. A truthful payment scheme is admitted by output function using this allocation algorithm based on proposed IEHO and it satisfies voluntary participation. In cloud environment, best allocation is provided by resource allocation protocol based on proposed strategy theory and its effectiveness is shown by experimentation results. Further, to enhance scheduling task very effectively, hybrid learning mechanism is focused in this work.

References

Sidhu, A.K., & Kinger, S. (2013). Analysis of load balancing techniques in cloud computing. *International Journal of computers & technology*, *4*(2), 737-741.

Ghanbari, Z., Navimipour, N.J., Hosseinzadeh, M., & Darwesh, A. (2019). Resource allocation mechanisms and approaches on the Internet of Things. *Cluster Computing*, *22*(4), 1253-1282.

Jung, G., & Sim, K.M. (2012). Location-aware dynamic resource allocation model for cloud computing environment. *In International Conference on Information and Computer Applications (ICICA), IACSIT Press, Singapore*.

Abirami, S.P., & Ramanathan, S. (2012). Linear scheduling strategy for resource allocation in cloud environment. *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, *2*(1), 9-17.

Gouda, K.C., Radhika, T.V., & Akshatha, M. (2013). Priority based resource allocation model for cloud computing. *International Journal of Science, Engineering and Technology Research (IJSETR)*, *2*(1), 215-219.

Lee, G., Tolia, N., Ranganathan, P., & Katz, R.H. (2010). Topology-aware resource allocation for data-intensive workloads. In *Proceedings of the first ACM asia-pacific workshop on Workshop on systems*, 1-6.

Sonkar, S.K., & Kharat, M.U. (2015). A survey on resource management in cloud computing environment. *International Journal of Advanced Trends in Computer Science and Engineering*, *4*(4), 48-51.

Chieu, T.C., Mohindra, A., & Karve, A.A. (2011). Scalability and performance of web applications in a compute cloud. *In IEEE 8th International Conference on e-Business Engineering*, 317-323.

Khan, H., Janjua, K., Sikandar, A., Qazi, M.W., & Hameed, Z. (2020). An Efficient Scheduling based cloud computing technique using virtual Machine Resource Allocation for efficient resource utilization of Servers. *In International Conference on Engineering and Emerging Technologies (ICEET)*, 1-7.

Park, J.H., Kim, H.W., & Jeong, Y.S. (2014). Efficiency sustainability resource visual simulator for clustered desktop virtualization based on cloud infrastructure. *Sustainability*, *6*(11), 8079-8091.

Cao, Y., & Ro, C.W. (2012). Adaptive scheduling for QoS-based virtual machine management in cloud computing. *International Journal of Contents*, *8*(4), 7-11.

Mousavi, S., Mosavi, A., & Varkonyi-Koczy, A.R. (2017). A load balancing algorithm for resource allocation in cloud computing. In *International conference on global research and education*, *Springer, Cham*, 289-296.

Jeyakrishnan, V., & Sengottuvelan, P. (2017). A hybrid strategy for resource allocation and load balancing in virtualized data centers using BSO algorithms. *Wireless Personal Communications*, *94*(4), 2363-2375.

Lin, F.P.C., & Tsai, Z. (2019). Hierarchical edge-cloud SDN controller system with optimal adaptive resource allocation for load-balancing. *IEEE Systems Journal*, *14*(1), 265-276.

Fan, Z., Shen, H., Wu, Y., & Li, Y. (2013). Simulated-annealing load balancing for resource allocation in cloud environments. *In International Conference on Parallel and Distributed Computing, Applications and Technologies*, 1-6.

Leontiou, N., Dechouniotis, D., Denazis, S., & Papavassiliou, S. (2018). A hierarchical control framework of load balancing and resource allocation of cloud computing services. *Computers & Electrical Engineering*, *67*, 235-251.

Pawar, C.S., & Wagh, R.B. (2012). Priority based dynamic resource allocation in cloud computing. In *2012 International Symposium on Cloud and Services Computing*, 1-6.

Wei, G., Vasilakos, A.V., Zheng, Y., & Xiong, N. (2010). A game-theoretic method of fair resource allocation for cloud computing services. *The journal of supercomputing*, *54*(2), 252-269.

Ergu, D., Kou, G., Peng, Y., Shi, Y., & Shi, Y. (2013). The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment. *The Journal of Supercomputing*, *64*(3), 835-848.

Arfeen, M.A., Pawlikowski, K., & Willig, A. (2011). A framework for resource allocation strategies in cloud computing environment. *In IEEE 35th Annual Computer Software and Applications Conference Workshops*, 261-266.

Inomata, A., Morikawa, T., Ikebe, M., Okamoto, Y., Noguchi, S., Fujikawa, K., & Rahman, S.M.M. (2011). Proposal and evaluation of a dynamic resource allocation method based on the load of VMs on IaaS. *In 4th IFIP International Conference on New Technologies, Mobility and Security*, 1-6.

Minarolli, D., & Freisleben, B. (2011). Utility-based resource allocation for virtual machines in cloud computing. *In IEEE symposium on computers and communications (ISCC)*, 410-417.

Xu, X., & Yu, H. (2014). A game theory approach to fair and efficient resource allocation in cloud computing. *Mathematical Problems in Engineering*, *2014*.

Wang, G.G., Deb, S., & Coelho, L.D.S. (2015). Elephant herding optimization. *In 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, 1-5.