

# Cauchy Particle Swarm Optimization (CPSO) based Migrations of tasks in a virtual machine

Vadivel R (✉ [Vadivel032015@gmail.com](mailto:Vadivel032015@gmail.com))

Hindustan Institute of Technology and Science: Hindustan University

SUDALAIMUTHU T

Hindustan Institute of Technology and Science: Hindustan University

---

## Research Article

**Keywords:** cloud computing, resource allocation, PCA, Optimization, migration count

**Posted Date:** March 22nd, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-315891/v1>

**License:**   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

**Version of Record:** A version of this preprint was published at Wireless Personal Communications on August 17th, 2021. See the published version at <https://doi.org/10.1007/s11277-021-08784-7>.

# Cauchy Particle Swarm Optimization (CPSO) based Migrations of tasks in a virtual machine

1.Mr.R.Vadivel, vadivelcse@gmail.com

2.Dr.T.SUDALAIMUTHU, sudalaimuthut@gmail.com

Assistant Professor in the Department of Computer Science and Engineering, Adithya Institute of Technology, Coimbatore, India.

Associate Professor in the Department of Computer Science and Engineering, Hindustan Institute of Technology and Science, Chennai, India.

## Abstract:

Cloud computing technology helps to resolve the problem in storage management by providing virtual resources to the end users. But, the overloading of virtual machines results in degradation of performances as well as it increases in the energy consumption of the virtual machines. Several techniques were used to determine the workloads of the cloud and then apply the migration algorithm for efficient utilization of resources. But, the process depends on the past outputs and only few step ahead predictions. Most of the techniques allocate the resources based on all the attributes. This results in higher processing time for the allocation. Hence, in this, an attribute based resource allocation is proposed to allocate and utilize the resources effectively based on the user demands. The modified Principal component analysis and relief is used for the attribute selection. Then, the selected attribute is processed with the hybrid Cauchy particle swarm algorithm for the allocation of resources. The proposed method is tested google cluster dataset and its performance is evaluated in terms of migration count and power consumption. The proposed method performance is compared with the automated migration technique (ALM) and forecast based migration technique (CF-LA). The proposed method outperforms both the existing technique by reducing the power consumption and the migration count between the virtual machines. Hence, the proposed MPCA and relief based CPSO is best for allocating the resources dynamically in the cloud.

**Keywords:** cloud computing, resource allocation, PCA, Optimization, migration count

## 1. Introduction

With today's technology, cloud computing is the greatest gift to human society. That's because cloud computing helps people store valuable information on virtual storage devices for free or at minimal cost. This is primarily a benefit to the enterprise. Resources are both physical and virtual servers, services, network devices, and software programs. Use a special program to access them.

Cloud storage is used by all end users because it has free data storage space and easy access

anytime, anywhere. As a result, the demand for cloud storage continues to grow. This type of storage plays a greater role in storing information from social media accounts, information from websites, and customer information from banks.

Cloud computing offers three types of services. The first service is to provide virtual resources for storing and processing data, similar to a data center. This service is called Infrastructure as a Service. The second is to use virtual resources with development mode. This service is called Platform as a Service. The third is to provide users with application-oriented services, which are called software as services.

Virtualization is the concept of operating different environments using the same physical system. The system used for virtualization is the host, and the new environment running on it is called a virtual machine. The concept of virtualization is to reduce the cost of individual hardware setups to run processes. In cloud computing, virtualization processes effectively utilize resources and improve services.

Virtual machines allow users to use resources efficiently, but frequent use and operation of a single virtual machine can overwhelm the system. To overcome this problem, the concept of migration has been introduced. In migration, the data is a shift from one server to another. It can be run in live or offline mode. Offline mode only runs before or after the process of reducing utilization.

Alnajdi et al. (2016) conducted a survey of the techniques in resource allocation in cloud computing in dynamic nature. Here, the allocation of the cloud resources can be done in two ways: one is static and the other is dynamic. The dynamic approach is difficult and require more processing time as compared to static. Based on dynamic,

three types of resource allocation model were discussed. The three models were based on agreement between providers and users, based on market and based on its utilization. Then, challenges in the allocation process were discussed and the future modifications also discussed in [1].

Nehru et al. (2016) developed an algorithm to perform dynamic allocation of resources using an auction technique [2]. Here, the auction takes place when the user demand increased. At that time, the service provider sold its resources based on bidding. The highest bid user can avail the resources. This approach can favor the providers, but user require tactic to utilize the resources effectively at minimum cost.

Pradhan et al. (2016) discussed different scheduling based algorithms in cloud computing for allocating the resources. The algorithms were round robin, and shortest job first. To improve the scheduling process, the round robin is modified and used for allocating the resources [3]. This modified round robin algorithm able to meet the customer requirements.

Verma et al. (2016) proposed a prediction based approach to allocate the resources to the users [4]. Here, the available users were called as tenants. Based on the usage of their resources, their requirements are classified as increase or decrease. This prediction is made with the help of machine learning algorithm like SVM, tree etc. Based on this prediction, the available resources shared among the users and also reserve for the future users.

Vakilinia et al. (2016) proposed a technique for the virtual machine placements or the migrations between the virtual machines [5]. Here, the migration is based on minimizing the power required for the migration. It is achieved with the help of a column based scheduler algorithm and call back approach to place the virtual machines for processing the data.

Induraisk et al. (2016) published a complete overview of the techniques used in the dynamic resource allocation in the cloud. It classified the techniques based on the load and temporal behavior of the load. The techniques used in the load based resource allocation were discussed in detail [6].

Neethu and Babu (2016) proposed a market based technique for providing the cloud resources to the users [7]. Here, the auction is used as the technique in IaaS layer. The modified paddy field algorithm used for determine the best service provider and the consumer can increase the revenue

of the providers and utilize their resources effectively,

Mousavi et al. (2017) proposed a hybrid heuristic approach for providing the services to the user in cloud computing effectively [8]. Here, the combination of grey wolf optimization and teaching learning based optimization is used for the allocation of resources. This hybrid combination works to solve the complete implementation of the tasks with the available resources. Mousavi et al (2017 a) proposed an algorithm for the resource allocation in the cloud computing [9]. But, the load balancing concept is used for the resource allocation process. The load balancing is used as the fitness function for the hybrid approach proposed in [8].

Seng et al. (2016) utilized the multiple objective genetic algorithm for allocating the resources in the cloud [10]. Here, first the requirement of sources is predicted. Based on the prediction and the available sources as input, the multiple objective genetic algorithm will determine the optimal allocation of resources to fulfil all the operations requested by the user.

Wang et al. (2017) proposed an improved stackelberg game theory approach for the resource allocation in the cloud [11]. Here, the providers are considered as leaders and users are considered as followers. Due to the multiple leaders and followers, it proposed a multiple leader and follower based stackelberg game theory for resource allocation. This approach able to provide a balance for profit between the leaders and follower in allocating and utilizing the sources. Yuan et al. (2017) also utilized the game theory approach for the resource allocation in the Geo-distributed cloud centres [12].

Singh and Bhasin (2017) utilized the ant colony algorithm for allocating the resources in cloud [13]. The allocation based on utilizing the available resources effectively for completing the task queued in the cloud.

Most of the techniques allocate the resources based on all the attributes. This results in higher processing time for the allocation. Hence, in this, a single attribute based resource allocation is proposed to allocate and utilize the resources effectively. The modified Principal component analysis is used for the attribute selection. Then, the selected attribute is processed with the Cauchy based particle swarm algorithm for the allocation of resources. The contributions of this paper is as follows:

- Single attribute for the resource allocation.
- Reduce the frequent migrations between the virtual machines.
- Reduce the power consumption of the machines to prolong its activity.

Paper is organised in the following format: Recent techniques used in resource allocation is discussed in the next section 2. Section 3 describes the working of the proposed method. Section 4 discuss the results of the proposed method and its performance comparison with the existing techniques. Section 5 summarize the proposed method and section 6 extends the proposed method with its future perspective.

## 2. Literature survey

Zhang et al. (2018) proposed a machine learning based concept in auction based resource allocation technique in cloud [14]. It utilized the Linear ALLOC algorithm for learning the information from the dataset for allocating the resources. This approach requires knowledge about the resources and user demand in all conditions for better resource allocation.

Vakilina (2018) proposed a technique for minimizing the power consumption in the cloud datacentre [15]. Because the frequent migration in the cloud centre results in depletion of energy in the machines. It results in high cost for the maintenance and interrupted services. This problem is minimized with the help of linear quadratic programming based optimization. This process first minimize the power consumption and reduce the migration between the machines. The technique has to be performed in reverse order to improve its performance.

Ali and alam (2018) proposed a technique based on the availability of resources [16]. Here, the prior knowledge about the cloud resources is available for allocating the resources to the users. The task with minimal processing time will be allocate for the processing. It results in larger queue for the long time consumption process and the load of the cloud will not be balanced.

Jarewah et al., (2018) proposed a technique for virtual machine placements and its migration process based on the prediction of host overload information [17]. Here, the logistic regression and mean absolute deviation is used for the prediction of the overload scenario in the cloud. Based on the prediction, the data centre can place the new virtual machines or implement the migration algorithm for avail the machine for new requests. This approach

requires large dataset to understand the loading conditions of the host.

Leontiou et al. (2018) implemented a two stage model for the allocation of resources to the users in the cloud [18]. Here, first accepting and allocation of tasks to the virtual machine is performed using local controller. Then, the global controller performs the balancing between the tasks operations in virtual machine. This two level controller is designed with the help of fuzzy logic. This method able to control the operations individually. Only limited users can access the resources.

Alkhalaileh et al. (2019) proposed a technique for the allocation of resource in dynamic nature for hybrid mobile based cloud computing [19]. Here, first the modelling of the cloud and user requirements is performed. That is, it performs the modelling of tasks and resources available in the cloud. Then, its energy requirement and cost for the resources are considered for the optimization process. Then, the data offloading technique is used to find the optimal resource allocation for the tasks. This technique relies on the modelling of information for effective utilization of resources. Saidi et al. (2019) surveyed different techniques used in the allocation of resources in the cloud to the user.

Muthulakshmi and somasundaram (2019) utilized the optimization algorithm for the resource allocation [21]. Here, the tasks, priority of the tasks, and distance between the server and client are considered for the allocation process. Based on these parameters, the available resources are allocated using the combined approach of simulated annealing and artificial bee colony algorithm. The algorithm can be improve its performance by considering the time consumption of the tasks.

Devarasetty and reddy (2019) also utilized the optimization algorithm for the allocation process [22]. But, here, the fitness function for the allocation is to meet the user demands. Due to this, the user gets more benefits as compared to the providers. The algorithm should be in balanced manner between the user and provider for effective utilization.

Gentry et al. (2019) proposed a technique based on the deadline of the tasks for scheduling the cloud resources [23]. Here, first the probability for all the incoming tasks complete its process is estimated with the available resources. Based on that model, the low probability tasks can be drop or defer for future process. Naha et al. (2020) also

proposed a technique based on the deadline of the tasks for resource allocation but in fog cloud [24]. These techniques can avail the sources but the loss of tasks is higher.

Belgacem et al. (2020) utilized the bioinspired algorithm for the scheduling of the dynamic demands of the user in the cloud [25]. Here, the objective function of the algorithm is multiple. The scheduling of task should increase the resource availability in the cloud and also cheaper for the user. Agarwal (2020) also implemented the multiple objective based scheduling concept in the cloud [26]. But, the lion optimization algorithm is used for the implementation. This approaches tries to create a balance between the user and provider. This results in poor usage of resources.

### 3. Existing method

Forsman et al. (2015) proposed an automated migration for allocating the resources in the cloud [27]. Here, the migration algorithm is framed for the live migration between the virtual machines to limit the power consumption and utilize the resources effectively. But, the frequent migrations results in high consumption of power. Paulraj et al. (2018) proposed a live migration technique for the virtual machines in the cloud [28]. Here, the migration is based on clustering the loads of the server and learning the workload of the server using combined forecast technique. The clustering and learning helps to balance the load between the virtual machines. But, the underload server gets

shock by frequent migration process. This technique has the following limitations.

- User requests are not considered in dynamic nature in both the techniques.
- Migration count can be reduce using optimization technique.
- It consider all the attributes for resource allocation and it may results in more processing time for allocation.

The above shortcomings were overcome with the single attribute based optimized resource allocation using CPSO algorithm in cloud for dynamic tasks.

### 4. Resource allocation using CPSO

This section describes the working of the proposed hybrid optimization method to allocate the available cloud resources effectively to the users to improve user experience and profit for the cloud providers. Due to the scarcity of physical sources, the migration process takes place through the virtualization concept. But, in-migration, resource allocation is a difficult job. Several techniques and their shortcomings discussed in the previous section. Based on that, a hybrid optimization-based technique proposed for resource allocation in the cloud to improve its resource efficiency. The block diagram for the proposed method working scenario shown in figure 1.

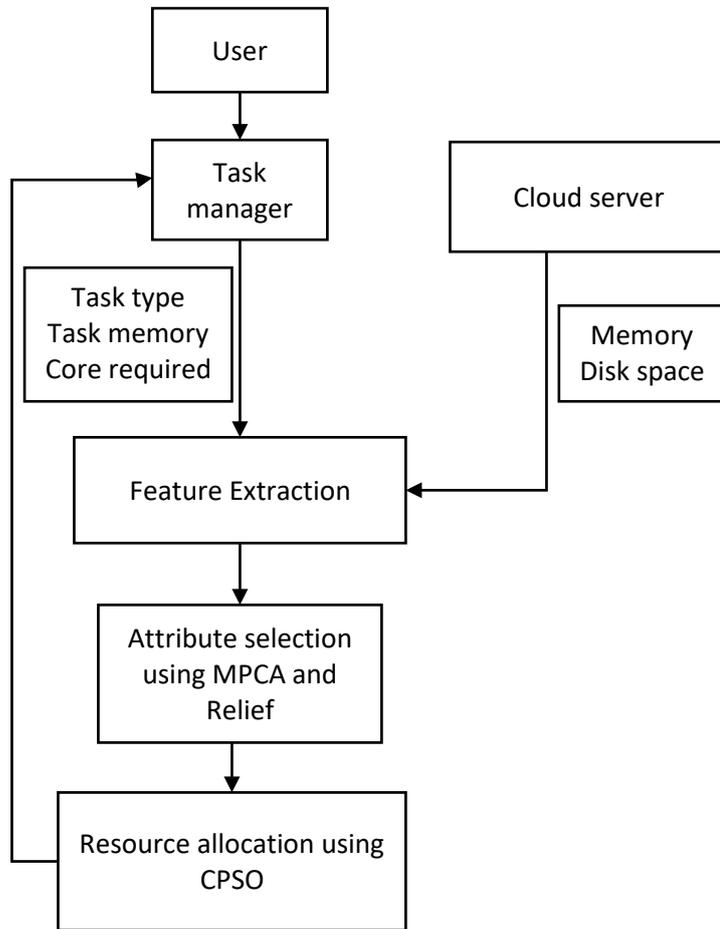


Figure 1. Proposed CPSO

Step 11: end

Steps in resource allocation using proposed CPSO technique.

Step 1: Receives request from user.

Step 2: Sort the user requests based on the requirements like CPU resource, memory and Disk space.

Step 3: The cloud available resource information is also collected in parallel manner.

Step 4: Once the information is collected, the cloud manager will select the dominant attribute in allocating the resource is selected using the Modified Principal component and relief based analysis.

Step 5: CPSO begins

Step 6: search for the best resource allocation slots by minimizing the fitness function.

Step 7: if (max iterations reach)

Step 8: Resource allocation starts

Step 9: else

Step 10: repeat from step 6.

The working of individual algorithm in resource allocation is explained in detail below.

#### a. User request

In this, the users will request the cloud service provider to allocate the resources for their applications.

#### b. Task Manager

The task manager will receive the request and collect the requirements of the users like disk space, CPU resources and approximate time period for processing to run their application.

#### c. Feature extraction

In this phase, the requirements of the each user is extracted and stored with their ID temporarily for resource allocation process.

#### d. Cloud service providers

The cloud service providers will perform multiple operations to allocate the resources to the user.

##### i. Resource collection

In this, the available resources are updated timely and on-going process are also included for allocating the resources.

##### ii. Attribute selection using Modified PCA and relief for resource allocation

In this, the dominant attribute selected for both the user features and cloud resources. In this, the attribute selection is performed by combination of modified Principal component analysis and Relief algorithm. The selection process were based on the relation between the predictor attributes like CPU resource, disk were sorted based upon its computational time.

##### a. Modified Principal component analysis

Here, the dimensionality of the dataset reduced into a lesser dimension with the help of modified Principal component analysis. Here, the PCA utilized the kernel to solve the non-linear dimensions. It uses the Gaussian kernel [29]. The steps in the modified principal component analysis is as follows:

Consider the dataset has N- dimensional feature and it's transformed into non-linear format as  $\phi(X)$ . The N-dimensional space converted into a D-dimensional space with a condition  $D \gg N$ . Based on this, the data point  $X_i$  in the dataset projected into a point in  $\phi(X_i)$ . This transformation is achieved with the help of Gaussian Kernel based PCA,

Let assume that the Sum of new feature vectors for n elements has zero mean.

$\frac{1}{n} \sum_{i=1}^n \phi(X_i) = 0$	1
--	---

The size of the covariance of the new feature space will be  $D \times D$  and equation 2 gives the covariance of the new feature space.

$C(D) = \frac{1}{n} \sum_{i=1}^n \phi(X_i) \phi(X_i)^T$	2
---	---

The eigenvalues and Eigen vector for the above equation 2 is as follows in equation 3.

$C(D)v_k = \lambda_k v_k$	3
---------------------------	---

The term k has values from 1, 2 and 3 to D in the new feature space. Substitute the covariance value from equation 2 in equation 3.

$\frac{1}{n} \sum_{i=1}^n \phi(X_i) \{\phi(X_i)^T v_k\} = \lambda_k v_k$	4
--	---

The value of  $v_k$  given in equation 5.

$v_k = \frac{1}{n} \sum_{i=1}^n a_{ki} \phi(X_i)$	5
---	---

The above equation substituted in equation 4 and it can be rewritten as in equation 6,

$\begin{aligned} \frac{1}{n} \sum_{i=1}^n \phi(X_i) \phi(X_i)^T \frac{1}{n} \sum_{j=1}^n a_{kj} \phi(X_j) \\ = \lambda_k \frac{1}{n} \sum_{i=1}^n a_{ki} \phi(X_i) \end{aligned}$	6
---	---

The equation 6 is multiplied with  $\phi(X_i)^T$  and it becomes

$\begin{aligned} \frac{1}{n} \sum_{i=1}^n \phi(X_i) \phi(X_i)^T \frac{1}{n} \sum_{j=1}^n a_{kj} \phi(X_j) \phi(X_i)^T \\ = \lambda_k \frac{1}{n} \sum_{i=1}^n a_{ki} \phi(X_i) \phi(X_i)^T \end{aligned}$	7
---	---

The term  $\phi(X_i) \phi(X_i)^T$  is the kernel and it is noted as in equation 8.

$\phi(j) \cdot \phi(X_i)^T = k(X_i, X_j)$	8
---	---

Replace the equation 7 with the kernels and it becomes as follow in 9

$\frac{1}{n} \sum_{i=1}^n k(X_i, X_i) \frac{1}{n} \sum_{j=1}^n a_{kj} k(X_i, X_j)$ $= \lambda_k \frac{1}{n} \sum_{i=1}^n a_{ki} k(X_i, X_i)$	9
--	---

In matrix format, the equation 9 becomes

$K^2 a_k = \lambda_k n K a_k$	10
-------------------------------	----

The terms  $K$  is the kernel matrix and it is same as in equation 8. And  $a_k$  is the  $n$ - dimensional column vector of the dataset and is denoted in equation 11

$a_k = [a_{k1}, a_{k2}, \dots, a_{kn}]^T$	11
---	----

In this, the kernel is Gaussian and it is given in equation 12.

$k(X, Y) = \exp(-\ X - Y\ ^2 / 2\sigma^2)$	12
--	----

Then, the final reduced dimensional vector  $Y$  is given by equation 13.

$Y_k(X) = \phi(X_i)^T v_k = \sum_{j=1}^n a_{kj} k(X_i, X_j)$	13
--	----

### b. Relief based feature reduction

Relief will calculate a feature score for each feature, and then apply it to the ranking and select the feature with the highest score for feature selection. Alternatively, these scores can be used as feature weights to guide downstream modelling. The relief feature scoring is based on the recognition of the feature value differences between the closest pairs of instances. If a feature value difference ("hit") is observed in adjacent pairs of instances with the same category, the feature score will decrease. Alternatively, if a feature value difference ("missing") is observed in pairs of adjacent instances with different category values, the feature score increases. The weight for the feature vector is given in equation 14

$W_A = w_A - (x - hit)^2 + (x - miss)^2$	14
--	----

Based on the above equations 1 to 14, the dominant attribute in the dataset determined for the allocation process of virtual machines in the cloud. This approach does not required any normalization or linearization process to perform feature reduction.

### iii. Resource allocation using CPSO

In this, based on the dominant attributes, the resources allocated to the user using hybrid optimization approach without disturbing the on-going process of the cloud users. Here, the improved version of the particle swarm optimization is used. The position update of the particle swarm optimization is replaced by Cauchy mutant operator to improve its global best solution by improving the local best solution. The steps in the PSO is as follows:

Step 1: PSO initialization with parameters and its optimal solution for processing.

Step 2: Evaluate fitness function for all particle swarms.

Step 3: Determine the best solution in the iteration.

Step 4: Overall best solution after the completion of iterations

Step 5: Velocity update.

Step 6: Position update using Cauchy mutant operator.

Step 7: if iter < max iter

Step 8: repeat step2

Step 9: else

Step10: Output the optimal allocation of resources for users.

#### a. Initialization

Here, the parameters for both the particle swarm were initialized. The initialization parameters composed of number of iterations, objective function, lower boundary and upper boundary of the available resources and other constant parameters for the position and velocity updates.

#### b. Fitness function evaluation

In this, the fitness function is to allocate the cores to the tasks with lesser computational time. It is based on the number of tasks core required to perform its tasks. The fitness function is shown in equation 15.

$FF = \min(\text{Core processing}_{time}, \text{migration count})$	15
--	----

#### c. Velocity update

First, the swarms evaluated its solution based on the objective function. The best solution of each swarm in that iteration is calculated and it is called as local best solution (LBS). Once the iteration completes, the local solutions were sorted in ascending and the top solution is the global best

solution. After obtaining the global best solution, the velocity of the swarms updated using equation 16.

$$v(t+1) = v(t) + L1 * R1 * (LBS - P_{CPSO}(t)) + L2 * R2 * (GBS - P_{CPSO}(t)) \quad 16$$

The terms L1 and L2 are the learning coefficients and its value is 1.5 and 2.0 respectively. The terms R1 and R2 are random numbers. The term  $P_{CPSO}(t)$  is the position of the swarms using Cauchy mutation operator

#### d. Position update using Cauchy operator

Here, the Cauchy mutation technique is used. The mutation technique is used to improve the solutions for a problem. This helps to improve the position of the swarms. The position update for the PSO with MGA is as follows in equation 17.

$$P_{CPSO}(t) = \max(P_j(t)) + \min(P_j(t)) - P_{i,j}(t) \quad 17$$

The term  $P_j(t)$  is the position of the particle and max and min represents the maximum and minimum solution of the particle in the iterations.

Based on the above position of swarms, the global best solution (GBS) of the PSO will also vary and it is given in equation 18

$$GBS(t) = GBS(t) + W_{CPSO}(t) * R(\text{minimum}, \text{maximum}) \quad 18$$

The term R is the random number its value lies between the boundaries of the search space and  $W_{CPSO}(t)$  is the Cauchy mutant operator and is given in equation 19

$$W_{CPSO}(t) = \frac{\sum_{j=1}^{pop\ size} v[j][t]}{pop\ size} \quad 19$$

The term indicates the velocity of the  $j^{\text{th}}$  particle in the iteration t. Based on this, the global and position of the swarms updated.

#### e. Termination

The algorithm continue its process till it reaches the criteria. For PSO, the minimal solution and maximum iteration is the criteria

Once all the iterations completed, the optimal allocation of the cloud resources for the user task determined.

### 5. Implementation and discussion:

The proposed resource allocation scheme is were tested with the help of JAVA programming under windows10 setup.

The testing is conducted on the google cluster dataset. The dataset comprises of CPU resources, task utilization time period and also it contains new resource request and their requirements. In this, the attributes from the new request and already available resources were used for the processing.

First, the missing information in the dataset will be removed to extract the user and Cloud available resources from the dataset.

Time	ParentID	TaskID	JobType	NrmTaskCores	NrmTaskMem
90000	757745334	1488529826	0	0	0.0311296
90000	975992247	1488529821	0	0	0
90000	1468458091	1488529832	1	0.021875	0.00235309
90000	1460281235	1488529840	0	0	0
90000	1164728954	1488529835	0	0.003125	0.0016384
90000	1288997448	1488529848	0	0.003125	0.0049152
90000	1488529845	1488529847	1	0.003125	0.000719232
90000	1263655469	1488529844	2	0	0
90000	1164728954	1488529851	0	0.003125	0.0016384
90000	981883546	1488529849	3	0.003125	0.0838861
90000	1263655469	1488529853	2	0	0
90000	757745334	1488529860	0	0	0
90000	1487094655	1488529866	0	0.003125	0.00382931
90000	1458411965	1488529868	0	0	0.000402513
90000	1390006181	1488529872	1	0	0
90000	1164728954	1488529877	0	0.003125	0.0016384
90000	1288997448	1488529880	0	0.0125	0.000992832
90000	1433362210	1488529881	0	0	0.024576
90000	1263655469	1488529882	2	0.015625	0.0251658
90000	1288997448	1488529889	0	0.003125	0.0049152
90000	1488529887	1488529894	2	0	0
90000	1213243701	1488529902	0	0	0.000336337

Figure 2. Google cluster dataset

Once the data cleaning is completed, the attribute selection takes place using modified Principal component analysis. The MPCA will select the major attribute for resource allocation process. The selection is based on the relation between the attributes like job type, number of task cores, memory space and time consumed for the processing. Based on this criteria, the proposed MPCA selects the task core as the optimal attribute for the resource allocation with respect to time.

After the attribute selection, the resource allocation process begins with the proposed hybrid optimization technique. In this, resources allocation is based on minimizing the fitness function by both the Cauchy based particle swarm optimization. The number of iterations required by the proposed algorithm to allocate the resources shown with the help of convergence curve in figure 3.

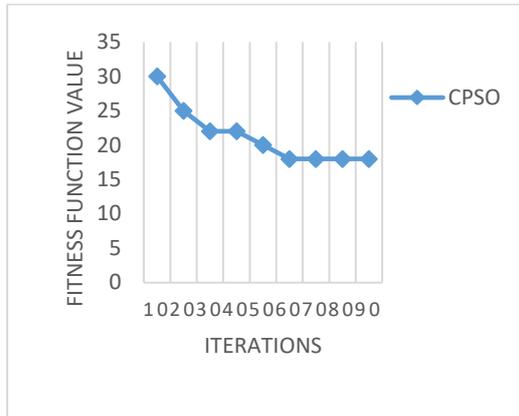


Figure 3. Convergence curve for CPSO

The CPSO helps to allocate the resources for the new tasks effectively by minimizing time for processing. To evaluate the proposed method resource allocations, the following two parameters were used.

- Migration count
- Power consumed

Because resource allocation is an important task, but frequent migrations results in high power consumption and reduce the cloud server efficiency. Hence, this two parameters were evaluated after the resource allocation process and is compared with the existing techniques like automated migration (ALM) [27] and combined forecast based live migration (CF\_LA) [28].

#### i. Migration count

Resource allocation is a difficult and an important task in the cloud service. Because the resource allocation is to share the available resources effectively without affecting the on-going process. While allocating the resources, migrations occurs due to the shifting of the tasks. This migrations results in pausing the process for some time. Therefore, a resource allocation scheme should use lesser migration for better process.

In this, the migration is determined by counting the number of migrations takes place while allocating the resources to the new request. This migration count is calculated for the proposed

method and is compared with the existing technique and is shown in figure 4.

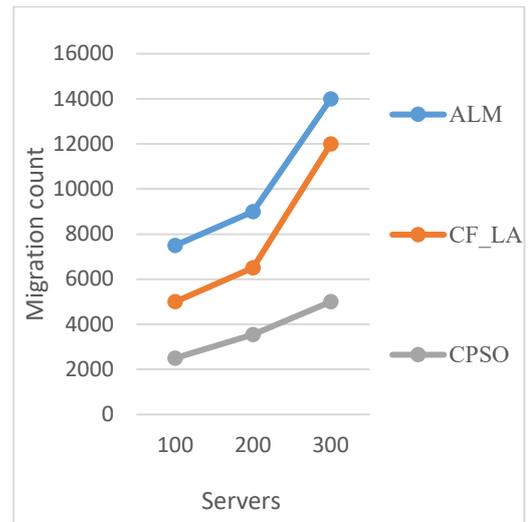


Figure 4. Migration count comparison

The proposed hybrid optimization technique can reduce the migration count effectively as compared to the other approaches. It is due to the attribute selection process and proposed CPSO. Because the attribute used for resource allocating is based on the time factor and it helps to reduce the frequent migrations.

#### ii. Power consumption

For effective operations, the power consumption also should be in optimal range. It can achieved with the help of resource allocating and limiting the frequent migrations. Generally, the term power consume is the amount of energy consumed by the servers in a particular time period t. It can be realized using the equations below

$E_{cons} = E_{cons} * T$	20
---------------------------	----

$P_{cons} = K * P_{max} + (1 - K) * P_{max} * U_t$	21
--	----

$P_{cons}$  is the power consumed at time t,  $P_{max}$  is the power consumed when the server is occupied to its maximum level, K is the power consumed by the idle server and  $U_t$  is the utilization of resources at time T. Based on this, the comparison of the power consumption between the proposed and existing techniques shown in figure 5.

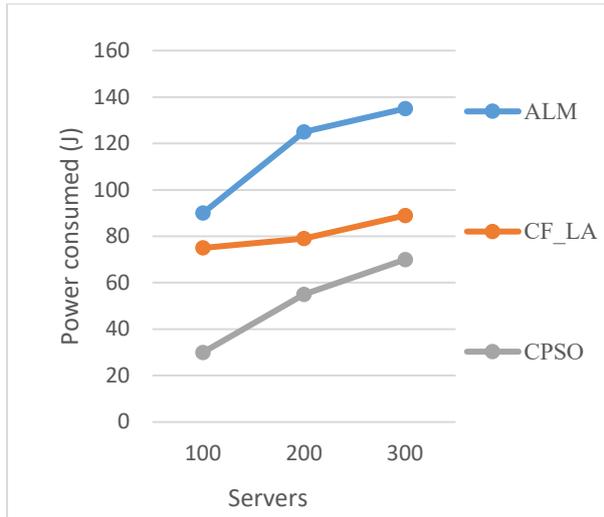


Figure 5. Power consumption comparison

Due to the lesser migration count, the proposed method can minimize its power consumption effectively as compared to the existing techniques and is depicted in the figure 5.

Based on the individual performance in resource allocation and also after resource allocation, the proposed method can outperform the existing techniques. Hence, the proposed MPCA based CPSO is the best resource allocation technique for the jobs in cloud.

The proposed approach Combined attribute selection based CPSO is compared hybrid PSO-MGA in terms of waiting time. Because the hybrid PSO-MGA also utilized the Cauchy mutation for position update with the help of genetic algorithm [31]. Due to the dual optimization process, the wait time is more and it reduce the efficiency of the processing of request. Therefore, in this, the resource allocation is improved in terms of multiple attribute selection process and the particle swarm position is directly updated using Cauchy mutation process. This modifications helps to reduce the wait time for processing the requests and it is shown in figure 6.

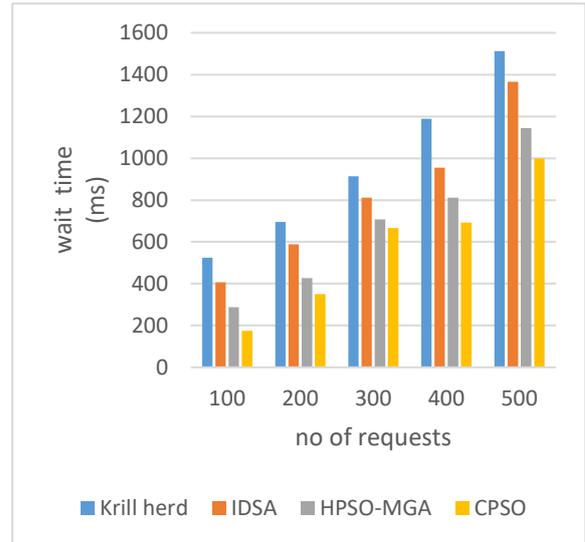


Figure 6. Wait time comparison

Figure 6 reduce the wait time for the request using the proposed technique as compared to the other existing techniques. Therefore, the proposed method is best in terms of services and also in power management.

## 6. Conclusion

In today's world, the cloud storage is used by all end users due to the freely available data storage area and easy access at any time and any place. Due to this, the demand of the cloud storage keeps on increasing. While utilizing the cloud storage, the virtual machine will run out of the resources for performing higher operations. To overcome this problem, the cloud migration concept is introduced. This migration is to perform both the utilization of resources as well as to minimize the energy consumption of the resources. Due to this, the proposed algorithm not only implements the resource allocation, but also reduces the power consumption and migration between the virtual machines. Along with limited the violations of services. The proposed method is implemented on the google dataset and evaluated in terms of the power consumption and number of migrations. The proposed method outperforms the existing method by following measures

- Number of migrations are reduced, which reduces the shock of the systems.
- Allocation of the resources done in minimal iterations
- The power consumption is also minimized.
- User services is improved with reduced wait time

Conflict of interest:

There is no conflict of interest.

Funding Information:

There is no funding information.

Availability of data and material:

There is no availability of data and material.

Author's contribution:

There is no author's contribution.

Code Availability:

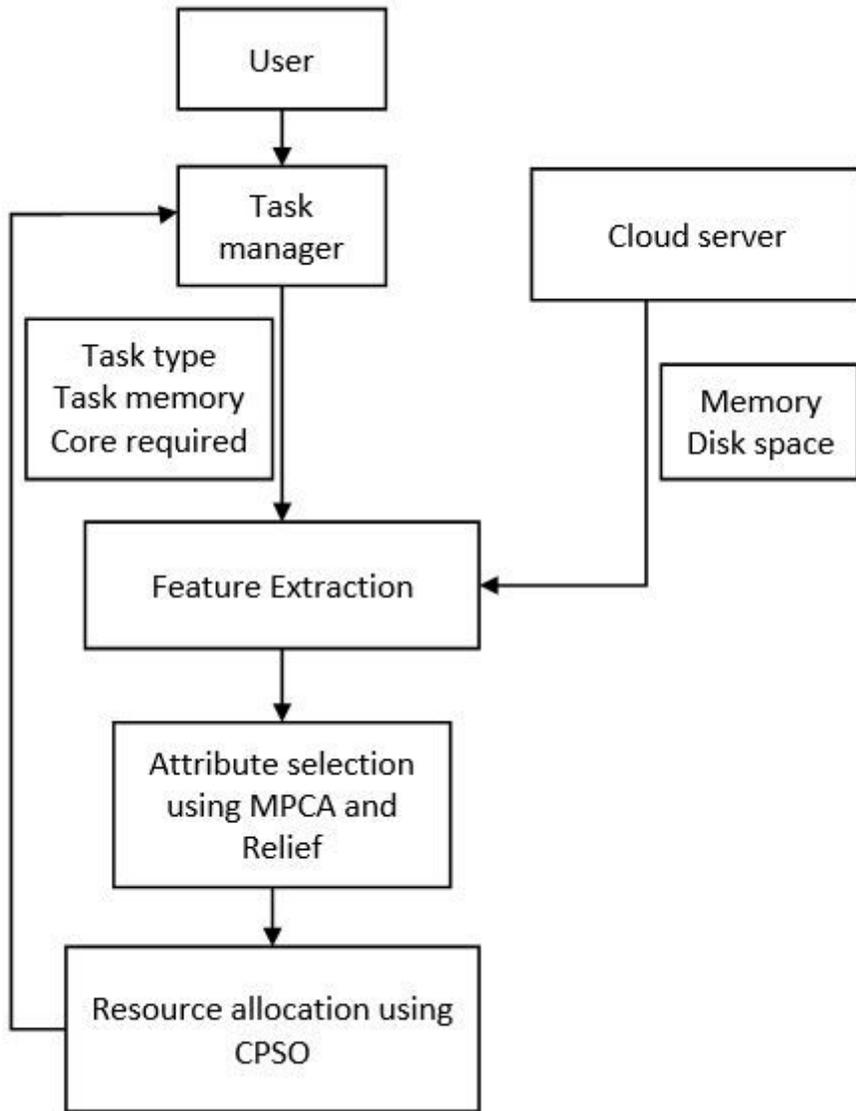
There is no code availability.

References

1. Alnajdi, S., Dogan, M., & Al-Qahtani, E. (2016). A survey on resource allocation in cloud computing. *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, 6(5).
2. Nehru, E. I., Shyni, J. I. S., & Balakrishnan, R. (2016, March). Auction based dynamic resource allocation in cloud. In *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)* (pp. 1-4). IEEE.
3. Pradhan, P., Behera, P. K., & Ray, B. N. B. (2016). Modified round robin algorithm for resource allocation in cloud computing. *Procedia Computer Science*, 85, 878-890.
4. Verma, M., Gangadharan, G. R., Narendra, N. C., Vadlamani, R., Inamdar, V., Ramachandran, L., ...& Buyya, R. (2016). Dynamic resource demand prediction and allocation in multi-tenant service clouds. *Concurrency and Computation: Practice and Experience*, 28(17), 4429-4442.
5. Vakiliinia, S., Heidarpour, B., & Cheriet, M. (2016). Energy efficient resource allocation in cloud computing environments. *IEEE Access*, 4, 8544-8557.
6. Indrusiak, L. S., Dziurzanski, P., & Singh, A. K. (2016). *Dynamic Resource Allocation in Embedded, High-Performance and Cloud Computing*. River Publishers.
7. Neethu, B., & Babu, K. R. (2016, September). Dynamic resource allocation in market oriented cloud using auction method. In *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)* (pp. 145-150). IEEE.
8. Mousavi, S., Mosavi, A., Varkonyi-Koczy, A. R., & Fazekas, G. (2017). Dynamic resource allocation in cloud computing. *Acta Polytechnica Hungarica*, 14(4), 83-104.
9. Mousavi, S., Mosavi, A., & Varkonyi-Koczy, A. R. (2017, September). A load balancing algorithm for resource allocation in cloud computing. In *International conference on global research and education* (pp. 289-296). Springer, Cham.
10. Tseng, F. H., Wang, X., Chou, L. D., Chao, H. C., & Leung, V. C. (2017). Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm. *IEEE Systems Journal*, 12(2), 1688-1699.
11. Wang, Y., Meng, S., Chen, Y., Sun, R., Wang, X., & Sun, K. (2017). Multi-leader multi-follower Stackelberg game based dynamic resource allocation for mobile cloud computing environment. *Wireless Personal Communications*, 93(2), 461-480.
12. Yuan, X., Min, G., Yang, L. T., Ding, Y., & Fang, Q. (2017). A game theory-based dynamic resource allocation strategy in geo-distributed datacenter clouds. *Future Generation Computer Systems*, 76, 63-72.
13. Singh, H., & Bhasin, A. (2017). Efficient Resource Management Technique for Performance Improvement in Cloud Computing. *Indian Journal of Computer Science and Engineering (IJCSE)*, 8(1).
14. Zhang, J., Xie, N., Zhang, X., Yue, K., Li, W., & Kumar, D. (2018). Machine learning based resource allocation of cloud computing in auction. *Comput. Mater. Continua*, 56(1), 123-135.
15. Vakiliinia, S. (2018). Energy efficient temporal load aware resource allocation in cloud computing datacenters. *Journal of Cloud Computing*, 7(1), 1-24.
16. Ali, S. A., & Alam, M. (2018). Resource-Aware Min-Min (RAMM) algorithm for resource allocation in cloud computing environment. arXiv preprint arXiv:1803.00045.
17. Jararweh, Y., Issa, M. B., Daraghme, M., Al-Ayyoub, M., & Alsmirat, M. A. (2018). Energy efficient dynamic resource management in cloud computing based on logistic regression model and median absolute deviation. *Sustainable Computing: Informatics and Systems*, 19, 262-274.
18. 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.
19. Alkhalaileh, M., Calheiros, R. N., Nguyen, Q. V., & Javadi, B. (2019, May). Dynamic resource allocation in hybrid mobile cloud computing for data-intensive applications. In *International Conference on Green, Pervasive, and Cloud Computing* (pp. 176-191). Springer, Cham.
  20. Saidi, K., Hioual, O., & Siam, A. (2019, November). Resources allocation in cloud computing: a survey. In *International Conference in Artificial Intelligence in Renewable Energetic Systems* (pp. 356-364). Springer, Cham.
  21. Muthulakshmi, B., & Somasundaram, K. (2019). A hybrid ABC-SA based optimized scheduling and resource allocation for cloud environment. *Cluster Computing*, 22(5), 10769-10777.
  22. Devarasetty, P., & Reddy, S. (2019). Genetic algorithm for quality of service based resource allocation in cloud computing. *Evolutionary Intelligence*, 1-7.
  23. Gentry, J., Denninnart, C., & Salehi, M. A. (2019, May). Robust dynamic resource allocation via probabilistic task pruning in heterogeneous computing systems. In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (pp. 375-384). IEEE.
  24. Naha, R. K., Garg, S., Chan, A., & Battula, S. K. (2020). Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment. *Future Generation Computer Systems*, 104, 131-141.
  25. Belgacem, A., Beghdad-Bey, K., & Nacer, H. (2020). Dynamic resource allocation method based on Symbiotic Organism Search algorithm in cloud computing. *IEEE Transactions on Cloud Computing*.
  26. Agrawal, S. Multi-Objective Optimization for Dynamic Resource Provisioning in a Multi-Cloud Environment using Lion Optimization Algorithm.
  27. Forsman, M., Glad, A., Lundberg, L., & Ilie, D. (2015). Algorithms for automated live migration of virtual machines. *Journal of Systems and Software*, 101, 110-126.
  28. Paulraj, G. J. L., Francis, S. A. J., Peter, J. D., & Jebadurai, I. J. (2018). A combined forecast-based virtual machine migration in cloud data centers. *Computers & Electrical Engineering*, 69, 287-300.
  29. Wang, Q. (2012). Kernel principal component analysis and its applications in face recognition and active shape models. arXiv preprint arXiv:1207.3538.
  30. Wang, H., Li, H., Liu, Y., Li, C., & Zeng, S. (2007, September). Opposition-based particle swarm algorithm with Cauchy mutation. In *2007 IEEE congress on evolutionary computation* (pp. 4750-4756). IEEE.
  31. Ramasamy, V., & Pillai, S. T. (2020). An effective HPSO-MGA optimization algorithm for dynamic resource allocation in cloud environment. *Cluster Computing*, 23(3), 1711-1724.

# Figures



**Figure 1**

Proposed CPSO

Time	ParentID	TaskID	JobType	NrmlTaskCores	NrmlTaskMem
90000	757745334	1488529826	0	0	0.0311296
90000	975992247	1488529821	0	0	0
90000	1468458091	1488529832	1	0.021875	0.00235309
90000	1460281235	1488529840	0	0	0
90000	1164728954	1488529835	0	0.003125	0.0016384
90000	1288997448	1488529848	0	0.003125	0.0049152
90000	1488529845	1488529847	1	0.003125	0.000719232
90000	1263655469	1488529844	2	0	0
90000	1164728954	1488529851	0	0.003125	0.0016384
90000	981883546	1488529849	3	0.003125	0.0838861
90000	1263655469	1488529853	2	0	0
90000	757745334	1488529860	0	0	0
90000	1487094655	1488529866	0	0.003125	0.00382931
90000	1458411965	1488529868	0	0	0.000402513
90000	1390006181	1488529872	1	0	0
90000	1164728954	1488529877	0	0.003125	0.0016384
90000	1288997448	1488529880	0	0.0125	0.000992832
90000	1433362210	1488529881	0	0	0.024576
90000	1263655469	1488529882	2	0.015625	0.0251658
90000	1288997448	1488529889	0	0.003125	0.0049152
90000	1488529887	1488529894	2	0	0
90000	1213243701	1488529902	0	0	0.000336337
90000	1164728954	1488529904	0	0	0

Figure 2

Google cluster dataset

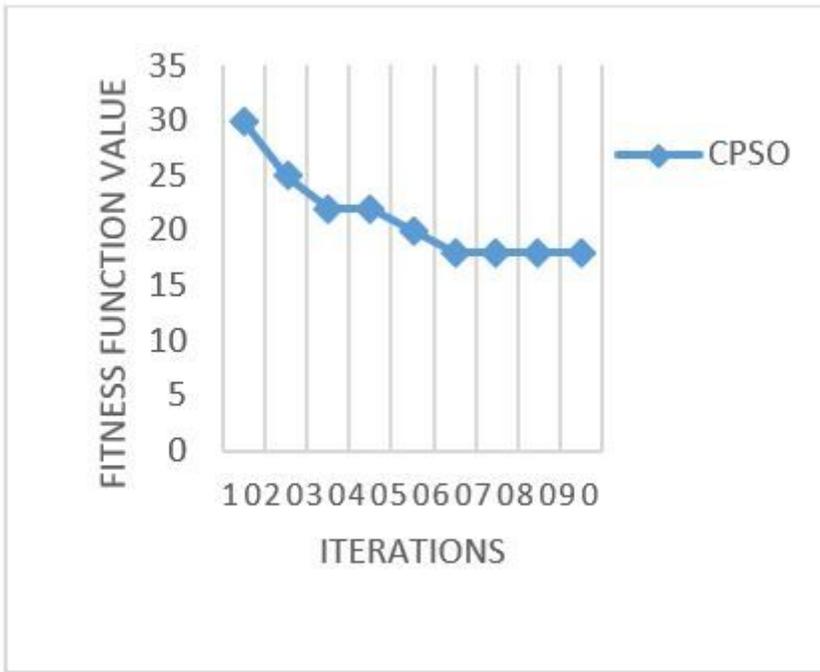


Figure 3

Convergence curve for CPSO

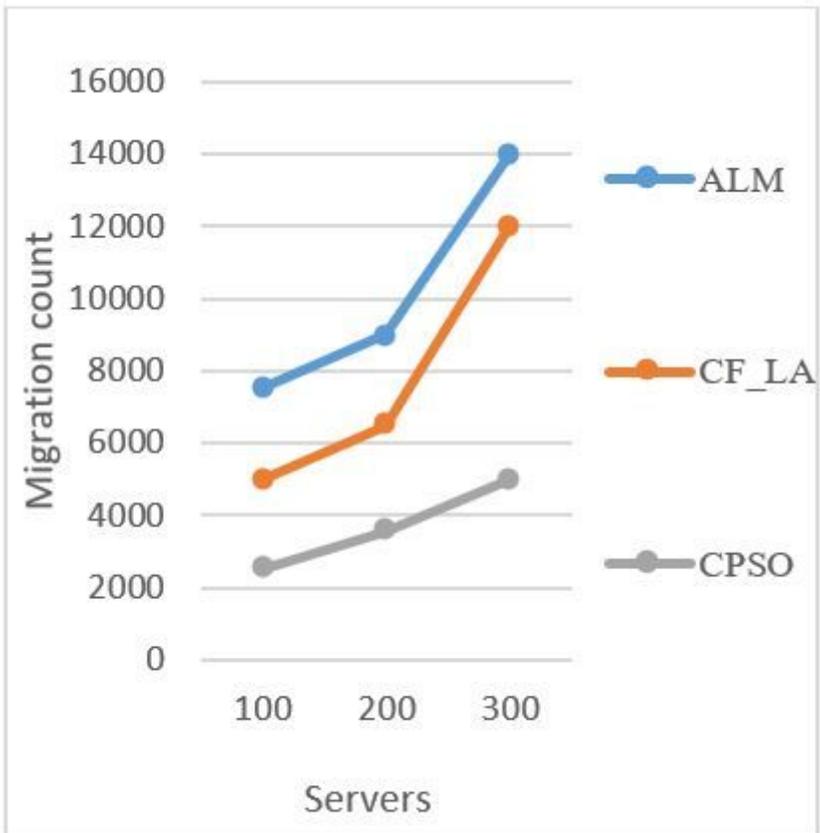
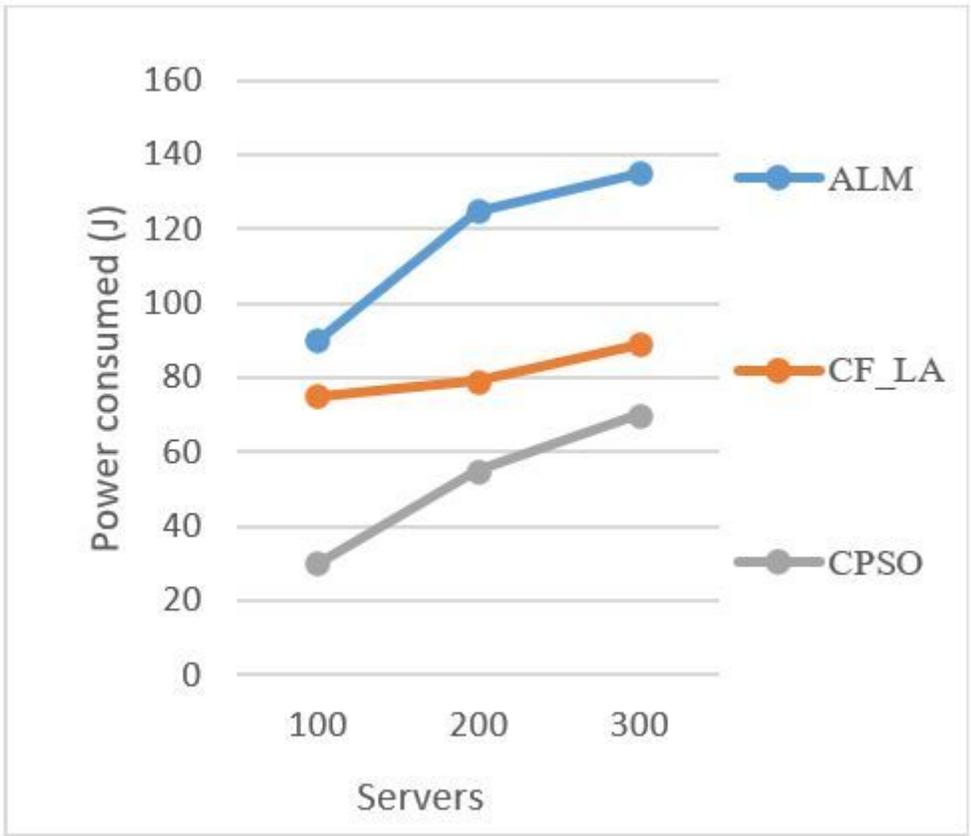


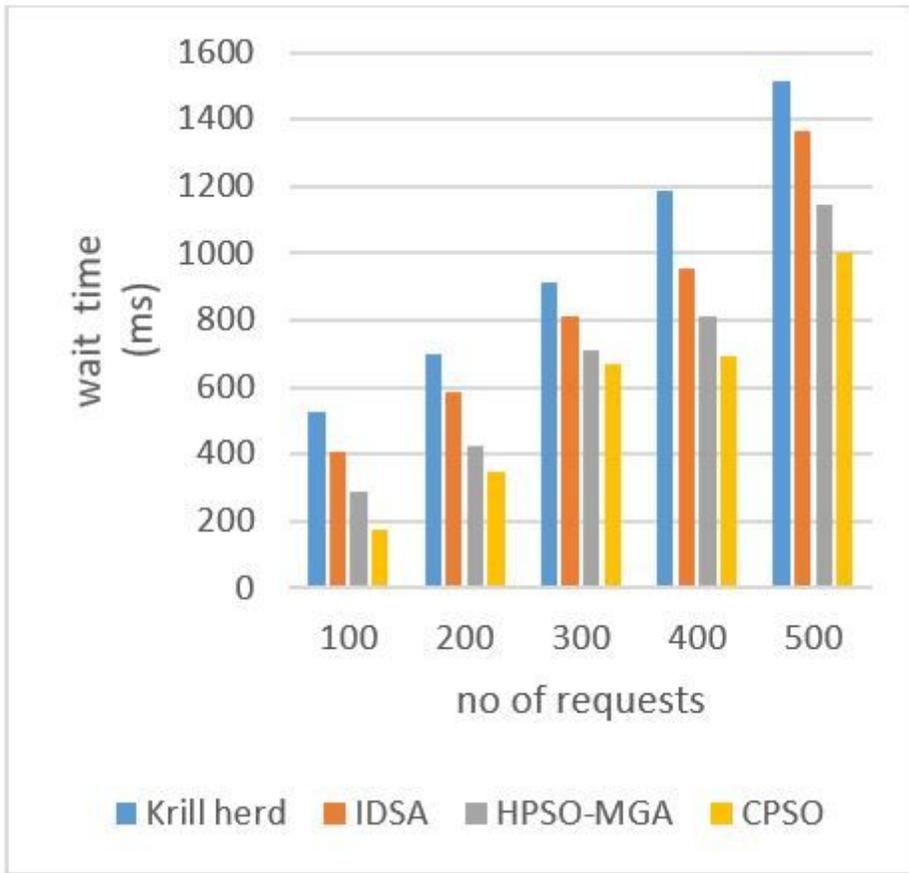
Figure 4

Migration count comparison



**Figure 5**

Power consumption comparison



**Figure 6**

Wait time comparison