ISSN (Online) : 0974-5645
ISSN (Print) : 0974-6846
**Indian Journal of Science and Technology,** *Vol 8(S3), 182–188, February 2015* DOI: 10.17485/ijst/2015/v8iS3/60501

# A Position Balanced Parallel Particle Swarm Optimization Method for Resource Allocation in Cloud

**R. S. Mohana***

Computer Science and Engineering Department, Kongu Engineering College, Erode, TamilNadu, India; mohanaamethesh@gmail.com

## Abstract

**Objective**: The main objective of this research is to allocate the resources with high profit and achieve high user satisfaction level in the cloud computing environment.

**Methods**: An innovative technique called Position Balanced Parallel Particle Swarm Optimization (PB-PPSO) method is introduced for allocating resources. The main intent of PB-PPSO is to find the optimized resources for the set of tasks with less make span and minimum price. The set of rules are generated from the optimized resources for the training process. In the testing process, the resources are allocated to the new users by learning the rules from the training process.

**Results**: PB-PPSO method shows high profit when compared to the existing methods such as Support Vector Machines (SVM) and Artificial Neural Network (ANN). In the PB-PPSO method, the optimized set of resources is determined for the set of tasks by using the particle swarm optimization algorithm. Then the rules are generated for the classification process. If the arrival rate of users is 500, the total profit is 720$ and the response time is 78ms. Based on the comparison and the results from the experiment shows the proposed approach works better than the other existing systems with high profit and less average response time.

**Conclusion**: The findings demonstrate that the PB-PPSO is presented and this method has high efficiency in terms of total profit and average response time for allocating the resources for the users.

**Keywords:** Artificial Neural Network, Cloud Computing, Position Balanced Parallel Particle Swarm Optimization, Resource Allocation, Support Vector Machines

## 1. Introduction

Cloud Computing is a new model that obtain the computer resources over the internet. Allocating efficient resources for the users with high profit is a serious concern in the cloud computing environment. Different methods are suggested for resource allocation. Xiong[1] suggested Service level Agreement (SLA) based allocation of resources in the cloud computing environments. In the service computing, resource allocation is typically related with a Service Level Agreement that is a set of quality of services and a price determined between users offered a cloud service request model with service level agreement constraint. Li[2] presented a new optimization algorithm is used for profit-driven service request scheduling based on dynamic reuse that acquires the personalized SLA distinctiveness of user requirements and current system workload.

Garg[3] proposed a novel meta-scheduling heuristics algorithm to handle the trade-off between overall execution time and cost and decrease them concurrently on the basis of a tradeoff factor. The meta-scheduling algorithms include Min-Min Cost Time Trade-off (MinCTT), Suffrage Cost Time Trade-off (SuffCTT), and Max-Min Cost Time Trade-off (Max-CTT). The scheduling problem mainly concerns that to decrease the cost and time of using resources for all the users across the community, is establish to be NP-hard due to its combinatorial nature.

*Author for correspondence

Reig[4] proposed a novel method to utilize an online prediction system that contains a fast systematic predictor and adaptive machine learning based predictor. The mechanism consists of two folds: facilitate the cloud to non-expert users by means of using service level metrics and support providers to do a proficient exploitation of their resources by using the resources left by web applications to perform jobs in a proficient way. To accomplish a service-level metric Machine Learning techniques are used in a Self-Adjusting Predictor which predicts the required resources.

Yeo[5] suggested a share distribution method called LibraSLA that is based on their service level contract. This method considers the service of admit new jobs into the cluster. To improve resource utilization, cluster Resource Management Systems (RMSs) require being aware of these necessities in utility-driven cluster computing. Jaideep[6] presented a learning based opportunistic algorithm that efficiently brings Map Reduce in the Software as a Service paradigm. To reduce overloading of resources Admission control scheme is very important to meet user service demands in the utility driven cloud environment. In the cloud computing, the cloud based services and the Map Reduce standard is augmented which makes the problem of admission control intriguing. Dhingra[7] suggested a new optimization method called Bacterial Foraging that incessantly optimize the resource allocation for enhancing the energy efficiency of the data centre. The above mentioned methods are less efficient for resource allocation.

Chitra[8] suggested a method called Optimum Session Interval based Particle Swarm Optimization (OSIPSO) in semantic web usage mining. This method is used to recognize the optimized session time by using Particle Swarm Optimization (PSO) method. The advantages of PSO are there is no overlapping and mutation computation. Rahmati[9] suggested a method called Comprehensive Learning Particle Swarm Optimization (CLPSO) algorithm to resolve the highly inhibited multi-objective Optimal Power Flow (OPF) problem that is used in power systems. This paper proposes the application of PSO and CLPSO to solve the multi-objective OPF problem.

In the existing research, two machine learning techniques are suggested such as SVM and ANN[10]. The intent of the Machine learning method is to build a distributed system for resource monitoring and prediction. This method includes learning-based methods for the optimization of the prediction of resources. But the limitation is the ANN is over fitting or under fitting problem with indelicate parameters. SVM is not suitable for huge number of tasks. So, in the proposed research PB-PPSO method is introduced for allocating resources with higher user satisfaction level. By using the PSO algorithm, the optimized resources are identified for the group of tasks and generate the rules. If a new user request the resources, the efficient resources are alocated by learning the rules. Section 1 briefly explains the previous work for resource allocation in cloud computing. Section 2 presents the existing research and Section 3 presents the proposed research. Section 4 explains the numerical results. Section 5 describes the conclusion and future work.

## 2. Support Vector Machine (SVM) and Artificial Neutral Network (ANN)

The SVM and ANN[10] are used for resource monitoring and allocation of resources. The main motivation of the machine learning method is to build a distributed system for resource monitoring and prediction. The machine learning methods such as Artificial Neural Network (ANN) and Support Vector Machine (SVM) are used for regression computation. These two methods can be used for modelling resource state prediction.

### 2.1 Working of SVM

Support Vector Machines (SVM) is a supervised learning method in with related learning algorithms that examine data and identify patterns which are used for categorization. The SVM is used to classify the success of ROI whereas a resource is provided to the users. The input given to the SVM training is set of metrics from the users (Deadline, Budget, Input File Size and Request Length) and the other associated information of IaaS provider (Service Initiation Time, Price, Input Data Transfer Price, Output Data Transfer Price, Processing Speed, Data Transfer Speed). In the SVM training process, the mapping of user necessities to the resources needs to be investigated by using the SLA based technique[11].

The input consists of aforesaid details denoted as matrix and represented by $x_i$ and $w$ denotes the weight value matrix whose product is summation with bias value to give the class value. This is given by,

$$x_i.w + b = 0 \quad (1)$$

This equation denotes a central classifier margin. This can be bounded by soft margin at one side using the following equation.

$$x_i.w + b = 1 \quad (2)$$

The input given to the SVM is plotted as data points in the graph. In the training process, the weight value is attuned so that the expected outcome is attained i.e.., Profit / ROI denoted as binary value true with "1" as per the equation $x_i.w + b = 1$ and "0" denotes the Loss in ROI. The weight value of successful ROI is used for testing phase. During testing, if the user gives a new request $x_{i+1}$ is require to be examined with previously acquired w with bias value b. If the result is in 1 then allocation process followed during testing will lead to profit otherwise incur a loss. Thus the classified output is given by,

$$y_{i+1} = \begin{cases} x_{i+1}.w+b=1, & \text{Profit} \\ x_{i+1}.w+b=0, & \text{Loss} \end{cases} \quad (3)$$

This is for when the minimum error is zero and may vary according to initial setting of parameters.

### 2.2 Working of ANN

The similar "n" metrics used in SVM are taken here as input and fed in to "n" node of input layer parallel. These input metrics are evaluated in numerous configurations for acquiring a better weight matrix that provide a superior result with minimal error at output layer. During training, for the given input the weight matrix is accustomed to acquire the preferred result say "1" as Boolean value denoting true in the profit. In the testing process, the updated weight matrix is utilized. During testing process, the input of request is evaluated with weight values and the ultimate result at the output layer determines the success (profit) and failure (loss) of the allocation scenario in accordance to the training details as same as SVM.

## 3. Position Balanced Parallel Particle Swarm Optimization (PB-PPSO) Method

In this proposed research, Position Balanced Parallel Particle Swarm Optimization (PB-PPSO) method is introduced for efficient allocation of resources. In this method, PSO is used for finding the optimized resources for the set of tasks. In the optimization algorithm, each particle has set of tasks and set of resources which begins with arbitrary initialization of particle's position and velocity. Every particle in the swarm behavior has two specifications: a position which denotes the suggested location and a velocity which means the speed of moving. The particle in the swarm negotiates over the entire search space and memorizes the best position found. The communication is takes place between the particles so that they adjust their locations and velocities based on solutions discovered by others. The position of the particle is scored by the fitness.

The fitness is computed by the objective functions. The main objective function is to provide task assignments that will accomplish minimum make span and minimum price for the users. Based on the fitness value, the particle is quantified as a good solution. During the execution of the PSO algorithm, the best fitness value is considered as the individual best fitness value. Comparing the entire particles in the swarm, the best fitness value is called global fitness value. Furthermore, the weighted mean value is computed local best and global best positions for reducing the computational time. So, at every time the particle position and velocity is updated. Finally, the global best is identified for the entire swarm. So, the optimized solution is identified which has set of tasks and set resources which has minimum make span and minimum cost. According to the optimized solution, the rules are generated in the training process. In the training process, there are two class labels. One is profit and another one is loss. The label "profit" is assigned for the less fitness value of the particle at every iteration. (i.e. tasks running in these set of resources takes high make span and high cost). The label "loss" is assigned for the high fitness value of the particle at every iteration. So, the training process is completed. If a new user gives request, the rules are learned and provide the efficient resources.

**Algorithm 1: Position Balanced Parallel Particle Swarm Optimization (PB-PPSO) Algorithm**

Input: Training samples and class labels
Output: Resource allocation

1. Initialize N number of particles with set of tasks and allocate the resources randomly, a position of particle is denoted by $X_i$ and velocity is denoted as $V_i$.
2. *pbest* represents the best well-known position of particle i and gbest signifies the best position of the entire swarm
3. Particle position is initialized as $X_i$
4. For every particle i=1, 2… N

5. Compute the fitness value for each particle
6. // Fitness computation
7. *Fitness* = *Min*(Makespan, total cost)
8. If the fitness value is higher than the *pbest*
9. Set the present value as the new pBest
10. Until a termination criterion is met
11. Select the particle with best fitness value of all particles as the gbest
12. //Computation of weighted mean value
13. W pBest$_{ij}$ (t) = $(m_1(t), m_2(t), ....m_3(t) = 1/M_j \Sigma_{\downarrow}(i=1)^{\uparrow} M \equiv a_{\downarrow}(i,j)\ P_{\downarrow}(i,j)^{\uparrow} t, 1/Mj \Sigma_{\downarrow}(i=1)^{\uparrow} M \equiv \square a_{\downarrow}(i,j) P_{\downarrow}(i,j)^{\uparrow} t, ...1/Mj \square \Sigma_{\downarrow}(i=1)^{\uparrow} M \equiv a_{\downarrow}(i,n)\ P_{\downarrow}(i,n)$
14. Wgbest (t) = $(M_1(t), M_2(t), .... M_3(t)) = \left( \frac{1}{M}\sum_{j=1}^{M} b_{i,1} G_{j,1}^{t}, \frac{1}{M}\sum_{i=1}^{M} b_{i,2} G_{j,2}^{t}, ..... \frac{1}{M}\sum_{i=1}^{M} b_{i,n} G_{j,n}^{t} \right)$
15. // Calculation of particle velocity
16. $V_i(t+1) = wv_i(t) + c_1 r_1[\hat{x}_i(t) - x_i(t)] + c_2 r_2[g(t) - x_i(t)]$ //Where, the index of the particle is represented by i, $v_i(t)$ is the velocity of particle i at time t, $x_i(t)$ is the position of particle i at time t, parameters w, c1, and c2 are coefficients
17. Update particle position and velocity
18. $x_i(t+1) = x_i(t) + v_i(t+1)$
19. Until some stopping condition is met
20. Generate the rules and assign class labels
21. Particles which has less fitness are assigned to "loss" at every iteration
22. Particles which has high fitness are assigned to "profit" at every iteration
23. // Testing Process
24. If a new task is submitted, learn the rules
25. Assign the resources for the tasks

## *Description*

In this algorithm, N number of particles is randomly initialized. Each particle has set of tasks and set of resources. The tasks are randomly allocated to the resources. The position of the particle is denoted as $X_i$ and velocity is denoted as $V_i$. The well-know position of the particle is represented as *pbest* and gbest denotes the best position of the entire swarm. For every particle in the swarm, the fitness is computed. The fitness is computed by the objective functions. The completion time of tasks allocated to resource j is evaluated as follows:

$$t_{complete}(j) = \frac{\left(\sum_{k \in A_j} T_k\right)}{C_j} 1 \le j \le m \quad (4)$$

In this equation $A_j$ is the set of task indexes which are assigned to resource j. $T_k$ and $C_j$ represents the size of the task i and processing speed of the resource j, respectively. The Make span is computed by

$$Makespan = Max\{t_{complete}(j)\} 1 \le j \le m \quad (5)$$

The second objective function is the total price that must be minimized. Suppose $w_j$ denotes unit price for resource j. Therefore, the execution cost of task i on resource j can be computed using the following equation:

$$Price(j) = t_{complete}(f) \times w_j \quad (6)$$

Then, the total cost of the scheduling is calculated as follows:

$$Total\ cost = \sum_{1 sjsm} Price(f) \quad (7)$$

The fitness is computed by,

$$Fitness = Min(\text{Makespan, total cost}) \quad (8)$$

The objective is to reduce the make span and the total cost for the execution of the task. If the fitness value is higher than the *pbest*, the present value is set as the *pbest* . Among all the particles, this algorithm computes the best fitness value that is called gbest. The weighted mean value is calculated for the local and global positions. It is usual, as in other evolutionary algorithm that relates elitism with the particles' fitness value. The greater the fitness, the more significant the particle is. Describing it properly, rank the particle in descending order according to their fitness value first. After that, allot every particle a weight coefficient $\alpha_i$ linearly diminishing with the particle's rank, which is, the closer the best solution, the higher its weight coefficient is. Equation (9) and (11) describes the weighted mean value for the local and global best positions.

W pbest$_{ij}$ (t) = $(m_1(t), m_2(t), ....m_3(t) = 1/M_j \Sigma_{\downarrow}(i=1)^{\uparrow} M \equiv a_{\downarrow}(i,j)\ P_{\downarrow}(i,j)^{\uparrow} t, 1/Mj \Sigma_{\downarrow}(i=1)^{\uparrow} M \equiv \square a_{\downarrow}(i,j) P_{\downarrow}(i,j)^{\uparrow} t, ...1/Mj \square \Sigma_{\downarrow}(i=1)^{\uparrow} M \equiv a_{\downarrow}(i,n) P_{\downarrow}(i,n)^{\uparrow} t)$

Wgbest (t) = $(M_1(t), M_2(t), .... M_3(t)) =$

$$\left(\frac{1}{M}\sum_{j=1}^{M} b_{i,1}G_{j,1}^{t}, \frac{1}{M}\sum_{i=1}^{M} b_{i,2}G_{j,2}^{t}, \ldots\ldots \frac{1}{M}\sum_{i=1}^{M} b_{i,n}G_{j,n}^{t}\right) \quad (10)$$

The particle velocity is evaluated as,

$$V_i(t+1) = \mathrm{w}v_i(t) + \mathrm{c}_1 r_1[\hat{\mathrm{x}}_i(t) - \mathrm{x}_i(t)]$$

$$+ \mathrm{c}_2 r_2[\mathrm{g}(t) - \mathrm{x}_i(t)] \quad (11)$$

In the equation (11), the index of the particle is represented by i, $v_i(t)$ is the velocity of particle i at time t, parameters w, c1, and c2 are coefficients.

The particle position is evaluated as,

$$x_i(t+1) = \mathrm{x}_i(t) + \mathrm{v}_i(t+1) \quad (12)$$

In the equation (12), $x_i(t)$ is the position of particle i at time t. This can be continued until the stopping condition is met. Finally, the set of rules are generated for the training process. The rules are nothing but the optimized solution for the resource allocation. The set of tasks with particular constraints (Deadline, Budget, Input File Size and Request Length) are assigned to the particular set of resources. So, there is less make span and less total cost is acquired. In the training process, the class labels are assigned. The particles with less fitness value means the tasks executing in the set of resources takes high Make span and high cost at a particular iteration. So, the class label "loss" is assigned to the attributes vales of the tasks. At the same way, the particles with high fitness value means the tasks executing in the set of resources takes less Make span and less total cost at a particular iteration. So, the class label "profit" is assigned to the attributes vales of the tasks. If the user submits the new task, the rules are learned from the training samples and allocate the resources.

## 4. Numeric Results

To implement the resource allocation techniques, CloudSim is used that is a Cloud environment simulator. The performance of the proposed PB-PPSO compared to the existing methods like SVM and ANN in the user and resource provider perspectives. In the user's point of view, observe the number of requests accepted and also how fast the user request is processed. (Called average response time). In the experimental results, three performance metrics such as total profit in $, Average Response Time in seconds and total number of initiated VM are compared.

Figure 1 shows that the PB-PPSO method achieves high profit and initiating the least number of VMs when arrival rate is increased from 100 to 500. When the user request number is increased, the total profit is increased in proposed method when compared to the existing system. This is because when the number of requests is increased, the number of users being accepted is increased too by utilizing initiated VMs.

Table 1 shows that total profit for the existing and the proposed system for the Variation in user request number. If the variation in user request number is 500, the total profit is y 750$ in the PB-PPSO, 745 $ in the ANN and 730$ in the SVM.

Table 2 shows that average response time for the existing and the proposed system for the Variation in user request number. If the variation in user request number is 500, the average response time is 77 secs in the PB-PPSO, 81 secs in the ANN and 85 secs in the SVM.

Figure 2 shows that the PB-PPSO method achieves smaller response time and accepts more number of users with less number of virtual machines. The average response time is taken in seconds. Compared to the existing systems like SVM and ANN, the average response time is less in the PB-PPSO method.

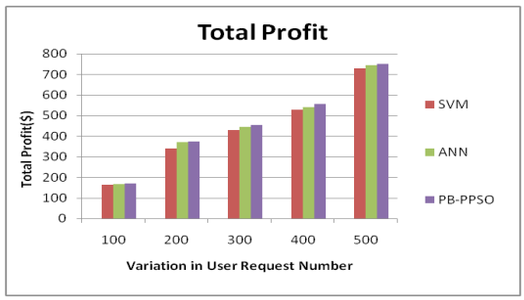


**Figure 1.** Total profit.

**Table 1.** Total profit Vs variation in user request number

| Total Profit($) | | | |
|---|---|---|---|
| Variation in user request number | SVM | ANN | PB-PPSO |
| 100 | 165 | 165 | 165 |
| 200 | 340 | 370 | 375 |
| 300 | 430 | 445 | 455 |
| 400 | 530 | 540 | 555 |
| 500 | 730 | 745 | 750 |

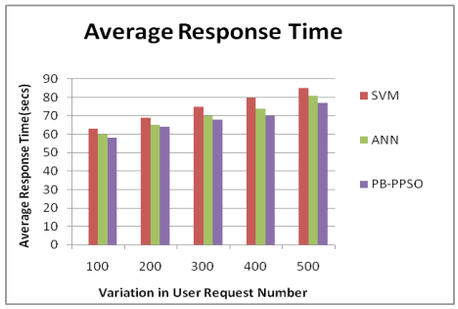


**Figure 2.** Average response time.

**Table 2.** Average response time Vs variation in user request number

| Average Response time (Secs) | | | |
|---|---|---|---|
| **Variation in user request number** | **SVM** | **ANN** | **PB-PPSO** |
| 100 | 63 | 60 | 58 |
| 200 | 69 | 65 | 64 |
| 300 | 75 | 70 | 68 |
| 400 | 80 | 74 | 70 |
| 500 | 85 | 81 | 77 |

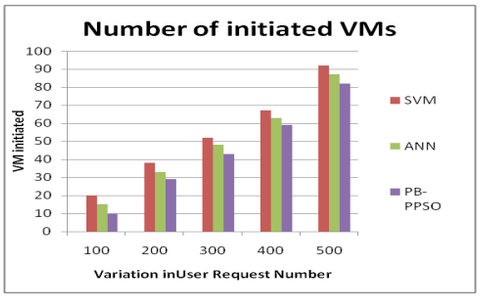


**Figure 3.** Number of initiated VMs.

**Table 3.** Number of initiated VMs Vs Variation in user request number

| Number of initiated VMs | | | |
|---|---|---|---|
| **Variation in user request number** | **SVM** | **ANN** | **PB-PPSO** |
| 100 | 20 | 15 | 10 |
| 200 | 38 | 33 | 29 |
| 300 | 52 | 48 | 43 |
| 400 | 67 | 63 | 59 |
| 500 | 92 | 87 | 82 |

Figure 3 shows that in the PB-PPSO method initiates least number of virtual machines when the arrival rate increases when compared to the existing methods.

Table 3 shows that Number of initiated VMs for the existing and the proposed system for the Variation in user request number. If the variation in user request number is 500, the Number of initiated VMs is 82 in the PB-PPSO, 87 in the ANN and 92 in the SVM.

## 5. Conclusion

Efficient resource allocation is an important concern in the cloud computing environment. In the existing research, the machine learning methods are suggested such as SVM and ANN. But the problem is ANN has over fitting or under fitting problem with indelicate parameters. SVM is not suitable for huge number of tasks. So, in the proposed method PB-PPSO method is introduced which finds the optimized soluting for allocating the resources. Using the optimization algorithm, the resources are identified to the set of tasks which takes minimum make span and minimum total cost. Based on this the rules are generated for the training process. In the testing, if a new user gives a request, the resources are allocated by learning the rules.

For future work, there are still some confronts in scalability, heterogeneity, SLA management automation, multiple QoS metrics which require to be explored further.

## 6. References

1. Xiong K, Perros H. SLA-based resource allocation in cluster computing systems. Proceedings of 17th IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2008); Alaska, USA: 2008.
2. Lee YC, Wang C, Zomaya AY, Zhou BB. Profit-driven service request scheduling in clouds. Proceedings of the International Symposium on Cluster and Grid Computing (CCGrid 2010); Melbourne, Australia: 2010.
3. Garg SK, Buyya R, Siegel HJ. Time and cost trade-off management for scheduling parallel applications on utility grids. Future Generat Comput Syst. 2009; 26(8):1344–55.
4. Reig G, Alonso J, Guitart J. Deadline constrained prediction of job resource requirements to manage high-level SLAs for SaaS cloud providers. Barcelona, Spain: Tech. Rep.

UPC-DAC-RR, Dept. d'Arquitectura de Computadors, University Politécnica de Catalunya; 2010.
5. Yeo CS, Buyya R. Service level agreement based allocation of cluster resources: Handling penalty to enhance utility. Proceedings of the 7th IEEE International Conference on Cluster Computing (Cluster 2005); Boston, MA, USA: 2005.
6. Jaideep DN, Varma MV. Learning based Opportunistic admission control algorithms for map reduce as a service. Proceedings of the 3rd India Software Engineering Conference (ISEC 2010); Mysore, India: 2010.
7. Dhingra A, Paul S. Green Cloud: heuristic based bfo technique to optimize resource allocation. Indian J Sci Technol; 2014; 7(5):685–91.
8. Chitra S, Kalpana B. Optimum session interval based on particle swarm optimization for generating personalized ontology. Indian J Sci Technol. 2014 Aug; 7(8):1137–43.
9. Rahmati M, Effatnejad R, Safari A. Comprehensive Learning particle swarm optimization (clpso) for multi-objective optimal power flow. Indian J Sci Technol. 2014 Mar; 7(3):262–70.
10. Mohana RS, Thangaraj P. Machine Learning approaches in improving service level agreement-based admission control for software-as-a-service provider in cloud. J Comput Sci. 2013; 9(10).
11. Wu L, Garg S, Buyya R. SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments. Australia: Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne; 2011 p. 1280–99.