

# Resource Allocation in Cloud Environment Using Approaches Based Particle Swarm Optimization

Vahid Asadzadeh Chalack  
Department of Computer  
Azad Islamic University  
Germi, Iran

Seyed Naser Razavi  
Computer Engineering  
Department, Faculty of  
Electrical and Computer  
Engineering, University of  
Tabriz, Iran

Sajjad Jahanbakhsh Gudakahriz  
Computer Engineering  
Department of Computer  
Azad Islamic University  
Germi, Iran

---

**Abstract:** It is obvious that in emerging computing paradigms such as cloud computing systems, scheduling is one of the main phases to take advantages of capabilities. The cloud computing environment is a dynamic environment which allows services to be shared among many users. Scheduling methods of traditional systems are ill-suited for the cloud computing systems, and this new environment requires new methods tailored to its specifications. In this paper, we developed multiple algorithms for task scheduling in cloud computing systems. These algorithms are based on the particle swarm optimization (PSO) algorithm, which is a technique inspired by collective and social behavior of animal swarms in nature, and wherein particles search the problem space to find an optimal or near-optimal solution. The algorithms were developed with the aim of minimizing Makespan, Flowtime and the task execution cost simultaneously. Simulation and test results show the better efficiency of the proposed methods than other similar algorithms.

**Keywords:** Resources Allocation, Cloud computing, Particle swarm optimization, Makespan, Flowtime.

---

## 1. INTRODUCTION

Cloud computing is an internet-based computing paradigm that provides a new framework for provision, consumption and delivery of IT services (including software, information and shared computing resources). Cloud computing allows the IT resources to be provided through a flexible and scalable internet-based method at the moment and the scale that is demanded by the user. A cloud computing provider offers online commercial applications through a web browser or other software. Applications and information are stored in the servers and users are given access on request. Details stay hidden from the users and they need no expertise or knowledge on cloud infrastructure technology to use it. In a distributed system, some nodes may become heavily loaded while others remain inactive or underutilized. The task of distribution scheduler is to schedule processes to nodes (processors) in an optimal way. While there is an implicit distinction between task scheduling and task allocation, these two terms revolve around one problem. In terms of resources, the problem is how to allocate processors to the processes. From a user perspective, the problem is how to schedule the processes in the processors. Studies have shown that nature-inspired heuristic optimization methods are most effective for such applications. Most of these methods try to minimize the total time of execution. Collective Intelligence is an artificial intelligence method based on collective behavior, which has been used by many researchers to develop heuristic collective intelligence optimization algorithms, such as ant colony optimization (ACO) algorithm, particle swarm optimization (PSO) algorithm and firefly algorithm. Among the abovementioned algorithms, PSO is widely regarded as the best because of features such as rapid convergence, insensitivity to initiation values, flexibility and high tolerance to error. The major drawback of PSO however is in its local search, which often converges to local optima and lead to globally suboptimal solutions. It has been proven that this algorithm can be improved through combination with other methods.

## 2. LITERATURE

There are various types of task scheduling algorithm. The main goal of a scheduling algorithm is to achieve high computing performance and best system throughput. Traditional scheduling algorithms cannot operate in cloud environment (because of overhead costs), thus providers have resorted to heuristic or hybrid algorithms to fill this gap [1]. Effectiveness of task scheduling has a direct effect on the quality of cloud, thus many algorithms have been developed to resolve this particular problem [2]. In some studies, algorithms have been developed to optimize the resource efficiency. In this section, we review some of these scheduling algorithms. Efficient virtual machine allocation and task scheduling is one of the key issues of the cloud systems. ACO-based load balancing algorithm has been proposed for solving the load balancing issue of virtual machines in the task scheduling process. This algorithm can adapt to dynamic cloud environment and decrease task scheduling execution time while providing a balanced load for the virtual machines of data centers [3]. Dynamic load balancing strategy has been developed based on genetic algorithm. This algorithm increases scheduling speed, reduces switching between processors, selects the task to be executed and even identifies the details of the tasks when used in advanced state. Adaptability threshold of this algorithm contributes to the load balancing of dynamic processors [4].

The time-cost balancing algorithm has been developed to provide a number of functions based on cloud computing capabilities. The example of these functions is the compression of workflow aimed to reduce execution time and cost based on the user input information [5]. This algorithm tries to efficiently map tasks to resources in the cloud. The main phase of the algorithm includes the use of improved bee colony algorithm to assign priorities to tasks and then the use of an algorithm for task grouping based on their priorities. This scheduling algorithm computes the cost for acquisition of resources and the performance of calculations to complete workflow tasks. In this algorithm, the ratio of acquisition of

the resources to the cost of efficient communication to perform workflow tasks had a significant improvement. [9]

A heuristic workflow scheduler algorithm based on the group optimization of tasks is proposed. The heuristic workflow scheduling algorithm was designed based on the group optimization of tasks, heuristic approach of applications according to the resources in the cloud with the aim of reducing computation times and reducing data transfer times. The algorithm includes two main components: the use of heuristic algorithm in order to discover right resources and the use of group optimization algorithm for correct mapping of these resources to tasks. The experimental results show that the use of this algorithm will save costs and properly distribute the workload on resources. [9]

### 3. Scheduling Problem

The task scheduling problem consists of N tasks and M machines. Each task must be processed by one of the M virtual machines such that in the end the overall scheduling duration would be minimized. The proposed algorithm is focused on the Quality of Service parameters makespan, flowtime and task execution cost. Each task can be executed on just one resource and cannot be paused before the end. The algorithm utilizes the ETC matrix model explained in [6]. Since the proposed scheduling algorithm is static, the expected time for executing task j on resource i is assumed to be predetermined and set within ETC matrix [i, j].

Completion\_Time [i,j] is equal time that job j be completed on resource i and is computed as follows.

$$(1) \text{Completion\_Time}[i,j] = \text{ETC}[i,j]$$

Makespan: maximum completion\_Time [i,j], that is computed as follows.

$$(2) \text{Makespan} = \text{Max} (\text{Completion\_Time}[i,j]) \quad 1 \leq j \leq N, \quad 1 \leq i \leq M$$

Flowtime: sum of the completion time of jobs [i, j] over all resources, that is computed by follow equation.

$$(3) \text{flowtime} = \sum_{i=1}^m \sum_{j=1}^n \text{completion\_Time}[i,j]$$

The goal of scheduling in the proposed algorithm is to submit each of the jobs to each of the resources to minimize makespan and flowtime of the jobs at last.[10]

### 4. The proposed scheduling algorithm

In the proposed scheduling algorithm, the population diversity factor is used to alter the inertia weight for solving the task scheduling problem in cloud computing. Before presenting the algorithm, the parameters required for solving the scheduling problem with PSO algorithm need to be assessed.

#### 4.1 Representation of Particle Swarm

One of the important issues of using PSO algorithm for solving task scheduling problem is how to turn a scheduling problem into a solution or in other words how to form a mapping between the solution and the particles of PSO algorithm. In the PSO scheduling algorithm, each particle is a possible solution for task allocation problem. Each particle vector has a length of N (N is the number of input tasks); and each element inside this vector is a random integer between 1 and M (M is the total number of resources).

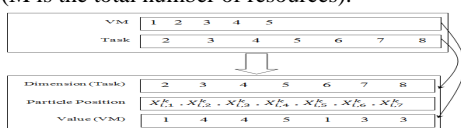


Figure 1. The mapping of tasks to resources

#### 4.2. Generating the Initial Population of Particles

In the proposed method, the initial population is generated at random. For this purpose, algorithm generates a random integer between 1 and M, representing the number of resource on which the task at hand will be executed. Randomness helps maintaining population diversity and gives all members of population an equal chance of being selected. Each particle vectors is as long as the number of tasks. The size of particle population determines the number of candidate solutions or the size of search in the problem space.

	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>
Particle <sub>1</sub>	VM <sub>2</sub>	VM <sub>5</sub>	VM <sub>3</sub>	VM <sub>2</sub>	VM <sub>1</sub>
Particle <sub>2</sub>	VM <sub>1</sub>	VM <sub>4</sub>	VM <sub>5</sub>	VM <sub>3</sub>	VM <sub>2</sub>
Particle <sub>j</sub>	VM <sub>1</sub>	VM <sub>5</sub>	VM <sub>5</sub>	VM <sub>4</sub>	VM <sub>2</sub>

Figure 2. Typical Particle

#### 4.3 Using the Population Diversity Factor for Changing the Inertia Weight

One of the problems of this algorithm is the premature convergence caused by rapid information exchange between particles which itself is because of lack of diversity in the particle population. Thus, one way of improving the overall performance of PSO algorithm is to maintain population diversity by improving indices such as inertia weight. In this method, we have used a regulator represented by kb [7, 8].

During the search, the regulator improves the static weight (w) by controlling the population diversity through negative feedback. The regulator index is shown with e and is expressed as follows:

$$e(t) = \frac{D_i(t) - D_0(t)}{D_i(t)} = \frac{D_0(t-1) - D_0(t)}{D_0(t-1)}$$

Where D<sub>0</sub>(t-1) is the extent of diversity at step t-1 and D<sub>0</sub> is the extent of diversity at step t.

Using the regulator index kb, static weight is defined as follows:

$$w(t) = kb \times e(t) = kb \left( 1 - \frac{D_0(t)}{D_0(t-1)} \right)$$

The suitable range for kb is between 0.9 and 1.6, and 1.4 is a good choice for most situations. In this version of method, the static weight will be adjusted according to population diversity (its decrease or increase). During the search, when population diversity D<sub>0</sub> decreases rapidly, inertia weight will increase to improve global search; when D<sub>0</sub> decreases very slowly, a small inertia weight w will be introduced to improve local search; and when D<sub>0</sub> fluctuates as a step progresses, a negative inertia weight will be used to reduce particle diversity. With this technique, the balance between global and local search will be maintained and the search will not converge to local optima. Diversity is measured using the following equation.

$$diversity(s(t)) = \frac{1}{n_s} (x + a)^n = \sum_{i=1}^{n_s} \sqrt{\sum_{j=1}^{n_s} (x_{ij}(t) - \hat{x}_j(t))^2}$$

Where (X<sub>j</sub>)(t) is the average of j-th dimension of all particles.

## 5. Simulation

In this section, the efficiency of the tasks scheduling algorithm presented in the previous section is evaluated. The objective of this algorithm is to schedule several independent tasks in a cloud environment. These tasks belong to an application that the user has requested to be executed on the cloud environment. Along with the task, the user specifies the QoS parameters, i.e. the time-cost optimization strategy that needs to be respected by the system. For example, the user can demand that cloud computing system should execute the application in the shortest time possible. Through simulation, the efficiency of time optimization algorithm can be compared with other algorithms.

### 5.1. Cloud computing environment model

A simulated environment for cloud computing includes one or more users and a number of resources. Users at any time may enter cloud computing and offer their application to run. This program includes a number of independent works, each of which can run on any desired resource. The user entity, after login, will be assigned its own broker entity, which performs its application scheduling and then returns results to the user after assigning tasks to resources and completing their implementation.

### 5.2. User model

Often in simulation experiments, the environment is modeled as single-user. The single-user version does not mean that only a user is logged in the system from simulation start to finish, but it means that the scheduling of any two users does not interfere with each other. For example, if a user is logged in at time 1500 and his application implementation finishes at time 4000 and another user is logged in at time 6000, then the system is called single-user. For comparison purposes, the system is presented as single-user.

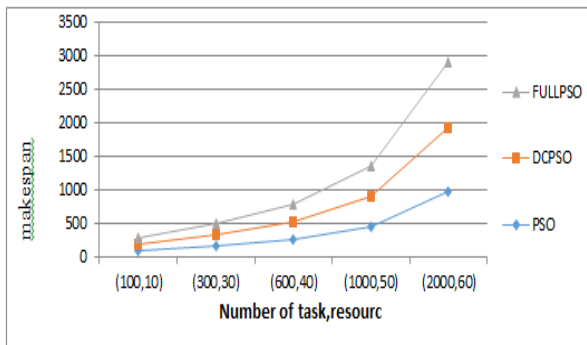


Figure 3. Diagram of makespan

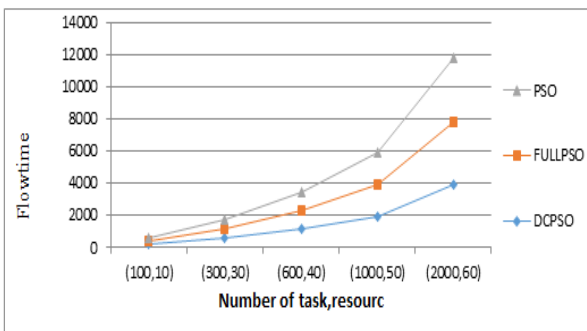


Figure 4. Diagram of flowtime

## 6. Conclusion

This paper provided a new method of resources allocation for cloud computing environment. The proposed algorithm utilized two high speed PSO-based methods which improved resource efficiency, selection of the best resource for task allocation, and minimized Makspan, while improving Flowtime parameter. The said parameters were simulated accurately. The simulation results showed the better performance of the proposed methods than other similar alternatives.

## 7. Suggestions

The future works are suggested to focus on development of a mechanism for partitioning applications or large problems into smaller sub-problems or tasks; as devising a mechanism for high speed and accurate portioning depending on task length (which is a determining load balancing factor in distributed systems especially cloud computing systems) will lead to improved speed and accuracy of scheduling and resource allocation, and thereby the increased efficiency of cloud computing system.

## 8. REFERENCES

- [1] Elzeki O. M. , M. Z. Rashad and M. A. Elsouid, "Overview of Scheduling Tasks in Distributed Computing Systems", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231 - 2307, Vol. 2, Issue-3, pp. 470-475, 2012. Casavant, T.L., Kuhl, J.G. 1988 A taxonomy of scheduling in general-purpose distributed computing systems. IEEE Transactions on Software Engineering.
- [2] Pardeep Kumar and Amandeep Verma," Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Vol 2, Issue 5, May 2012,pp:111-114.
- [3] Tinghuai Ma, Ya Chu, Licheng Zh., Otgonbayar A., "Resource Allocation and Scheduling in Cloud Computing: Policy and Algorithm", IETE Technical, Vol. 31, No. 1, 2014.
- [4] Bilgaiyan S., Sagnika S., Das M., "An Analysis of Task Scheduling in Cloud Computing using Evolutionary and Swarm-based Algorithms", International Journal of Computer Applications (0975 – 8887), Vol. 89 , No.2, pp. 11-18, 2014.
- [5] Selvarani S, Sadhasivam G., "Improved Cost-based Algorithm for Task Scheduling in Cloud Computing", Computational Intelligence and Computing Research (ICCIC), pp. 1-5, 2010.
- [6] Bittencourt LF, Madeira E., "A Cost Optimization Algorithm for Workflow Scheduling in Hybrid Clouds", In Journal of Internet Services and Applications, pp. 207-227, 2011.
- [7] J. Kennedy and R.Mendes, "Neighborhood topologies in fully-informed and best of neighborhood particle swarms,"vol. 12, pp. 50, 2003.
- [8] R.NEVES, "Watch Thy Neighbor Or How The Swarm Can Learn Form its Environment. " vol. 10, pp. 50, 202.
- [9] Dillon T,Wu Ch,Chang E.2010.Cloud Computing: Issues and Challenges.In Advanced Information Networking and Applications (AINA).24th IEEE International Conference on,pp:27-33.

- [10] Sajjad, Asadzadeh. Ch., Seyed Naser, R., Ali H.,(2014),  
“Job Scheduling on the Grid Environment using Max-  
min Firefly Algorithm”, International Journal of  
Computer Applications Technology and Research, Vol3.  
Issue 1, p.p 63-67