

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Parallel Particle Swarm Optimization for Task Scheduling in Cloud Computing

M. Vidhya¹, N.Sadhasivam²

P.G. Student, Department of CSE, Bannari Amman Institute of Technology, Sathyamangalam, India

Assistant Professor, Department of CSE, Bannari Amman Institute of Technology, Sathyamangalam, India

ABSTRACT: Cloud computing is the internet based computing where sources are accessed via online. These services have the ability to extend the provisioning of resources based on users demand. The user applications are submitted to the virtual machines for processing. So the mapping of user tasks to virtual machines plays a major role in efficient provisioning of resources. The task scheduling problem can't be solved in polynomial time. So for solving this kind of problem, the heuristics algorithms like particle swarm optimization are used. To minimize the execution time of particle swarm optimization, the parallel version of the algorithm is used.

KEYWORDS: cloud computing, task scheduling, particle swarm optimization, mapping, user tasks

I. INTRODUCTION

Cloud computing is the large collection of pool of resources which are assigned to users based on demand. The cloud computing also referred as the internet based computing where the services are accessed through internet via simple Application Programming Interface (API) or web client program. Cloud computing is defined by Rajkumar Buyya et al. in [1] as a pool of services that are delivered to users based on their demand similar to traditional utilities such as water, electricity, gas, and telephony. In this model, the users access services without regard to where the services are hosted and how they are delivered.

Cloud computing is a new paradigm for distributed computing. The new emergence of cloud computing technologies provides method to deal with complex applications which are the applications of great deal of data and needing high performance applications. Clouds have been define to be a type of parallel and distributed system consisting of inter-connected and virtualized computers. Distributed computing systems have emerged as a powerful platform for providing higher computational power required by a wide range of applications.

The performance of a parallel application on distributed system is highly dependent on both the characteristics of application such as execution cost and communication cost between tasks etc. and the features of platform on which it is implemented such as computation capacity of the processors, number and type of processors, memory size, communication bandwidth, etc. To effectively exploit distributed systems, an important challenge is the mapping of tasks to processors in order to achieve objectives like either load balancing, minimizing the communication costs, or combination of both for computationally demanding tasks with diverse computing needs.

The task scheduling is defined as mapping of user tasks to virtual machines in order to improve performance. The resources should be used effectively without affecting the service parameters. In run time the number of users can be increased, so the tasks need to be scheduled according to increase in proportion. Therefore, there is a need for better algorithms to schedule tasks on these systems. The task scheduling is the NP-complete problem where the execution time of an algorithm is not in polynomial time. The heuristic algorithms are used for solving NP-complete problems. There are lots of algorithms for solving task scheduling such as Simulated Annealing, Genetic algorithm etc. The Particle Swarm Optimization algorithm is proved better for distributed environment compared to genetic algorithm. The particle swarm optimization is extensively used because of its ease of implementation and faster convergence.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

II. RELATED WORK

The task assignment is a one of the core research area in cloud computing. The task assignment is responsible to select the best suitable resources in a cloud, by taking some constraints such as cost, time, and bandwidth etc. The cloud service providers should have high performance servers for dealing compute intensive applications and large pool of storage space to deal with data intensive applications. In [6], the cloud task scheduling is described as the process of mapping tasks to available resources on the basis of characteristics of tasks and the requirements of clients. The resources should be utilized efficiently without affecting the service parameters of cloud. The Scheduling process in cloud is categorized into three stages namely the resource discovery where the data center broker discovers the resources and their status information, selection of resource based on certain parameters and task submissions where the tasks are submitted to selected resources. In [9], Pandey et al. describes non-linear model which assigns data to nearest site in order to minimize data transfer cost and data communication cost.

In [7], Kennedy et al. describes the particle swarm optimization (PSO) based on birds flocking behavior where each bird are treated as particles. All the particles are moving around in search space where each particle has certain position and velocity. Each particle updates its velocity based on the best flying experience of its own and the best experience among all particles. This process is repeated until certain criteria meets either maximum number of iterations or the swarm converges. In [12], Shi et al. empirically measures the particle swarm optimization to illustrate advantages and disadvantages of PSO. The PSO algorithm converges quickly and to improve adaptive inertia weight is implemented. In [17], Zhi et al. describes the adaptive particle swarm optimization to improve the performance of PSO. In this the elitist learning is used to automatically control the inertia weight, acceleration coefficients and other algorithm parameters to provide better performance.

In [10], Salman proposed the particle swarm optimization algorithm for task assignment problem where the results are compared with other heuristic algorithms such as genetic algorithm. The results prove that the algorithm is better suitable for task assignment problem. In [4], double meta-heuristic algorithm such as particle swarm optimization and differential evolution are used for single machine scheduling problem. Here the small position value rule is used to convert the continuous optimization problem into discrete optimization problem. The double meta-heuristic improves the quality of solution and performance. In [16], a particle swarm optimization algorithm is used to solve task assignment problem in grid computing. The objective is to minimize the completion time of tasks where the results are proved better than the genetic algorithm. In [8], the PSO is used to assign compute intensive tasks to resources. To convert continuous optimization technique to discrete the small position value rule is used. The results prove that the PSO is better suitable to cloud computing

In [15], a modified PSO algorithm is proposed to task scheduling problem in order to reduce execution time and computation cost. The experiments are evaluated using different parameters such as time, speed and efficiency. The results proved the modified PSO surpass ordinary PSO. The problem with original PSO algorithm is premature convergence and getting stuck into local optimum. Cheng et al. proposed gradient descent method with particle swarm optimization to avoid premature convergence and swarm stagnation. In [3], the hybrid PSO is proposed to avoid deadlock in job shop scheduling problem which caused due to the use of PSO. The hybrid PSO uses the other methods to solve the local searching such as Tabu Search, Simulated Annealing. In [11], the improved particle swarm optimization algorithm is used to reduce the tasks average running time and raises the availability rate of resources. The proposed scheme uses PSO combined with Simulated Annealing to avoid getting stuck into local minimum.

In [13], the parallel executions of tasks on heterogeneous multiprocessor are considered. The genetic algorithm is used to provide efficient mapping of tasks to resources, the sequence of execution of tasks and the optimal configuration of system. In [14], the hybrid tabu search-PSO is proposed to balance between local and global exploration. In all of the exiting works, to overcome premature convergence and local optimum different variants of PSO are proposed. The algorithms discussed so far introduces complexity to original PSO algorithm. So in [5], Gonsalves et al. proposed a parallel version of PSO to improve premature convergence and local optimum without introducing any complex process. In this work, the parallel PSO is applied for task scheduling problem in heterogeneous environment.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

III. PROBLEM FORMULATION

To formulate a task assignment scheme, the set of tasks T_i where $i=\{1,2,3,\dots,n\}$ and the set of computational resources R_j where $j=\{1,2,3,\dots,m\}$ are considered. The set of nodes considered are heterogeneous where the processing ability and the processor's memory are different. So the tasks execution time will differ based on the computing resource on which it is assigned. If task i is assigned to computational node j , then $x_{i,j}$ will be set to 1, otherwise it is 0. The objective is to find the optimal mapping of tasks to resources which minimizes the job completion time.

IV. TASK SCHEDULING BASED ON PARALLEL PARTICLE SWARM OPTIMIZATION

The task scheduling is the core research area in cloud computing. There are lots of algorithms to provide efficient mapping of tasks to resources. The particle swarm optimization is one of optimization technique used in task scheduling. But it suffers from premature convergence and getting stuck into local optimum. To overcome this parallel version of particle swarm optimization is introduced. The algorithm has two phases. One is multi-evolutionary phase in which the swarm is randomly divided into k sub-swarms. Each sub-swarm evolves independently. Each particle updates its position based on its own experience and swarm's best experience. This process is repeated for certain iterations. After that the swarms are merged to form single evolutionary phase. In which the swarm best of each sub-swarm is compared to determine the global best. Each particle is updated based upon personal best, swarm best and the global best. This process is repeated for certain iterations. The parallel version of the algorithm is easy to implement and provides better assignment compared to original PSO. The figure 1 shows the parallel particle swarm optimization. The algorithm steps are as follows:

1. Randomly generate K number of independent swarm populations so as to be uniformly distributed over the entire decision space.
2. Evaluate the fitness of the particles in each individual swarm. In the minimization problems, the fitness of a particle is inversely proportional to the value of the function.
3. Determine the particle-best ($pbest$) and the swarm best ($sbest$) of each individual swarm.
4. Update the velocity and position of each particle in each swarm according to (1) and (2).
5. Allow the individual swarms to evolve independently through N iterations (i.e., repeat Steps 2 through 4).
6. Determine the global-best ($gbest$) by comparing the swarm-best ($sbest$) of all the swarms. For minimization problems, the $gbest$ in a given iteration is given by the following equation:

$$gbest = \max(sbest_1, sbest_2, sbest_3, \dots, sbest_k) \quad (1)$$

7. The independent swarms are interacted using $gbest$ value. The velocities of particles are updated according to the following equation

$$\begin{aligned} v_{i,j}(t) = & wv_{i,j}(t-1) + c_1r_1(p_{i,j}(t-1) - x_{i,j}(t-1)) \\ & + c_2r_2(p_{s,j}(t-1) - x_{i,j}(t-1)) \\ & + c_3r_3(p_g(t-1) - x_{i,j}(t-1)) \end{aligned} \quad (2)$$

where $x_{i,j}$ is the position of i th particle in j th swarm, $v_{i,j}$ is velocity of i th particle in j th swarm, $p_{i,j}$ is the $pbest$ of i th particle in j th swarm, $p_{s,j}$ is the swarm best of j th swarm, p_g is the global best among all the sub-swarms, c_1, c_2, c_3 are acceleration parameters and r_1, r_2, r_3 are the random variables.

8. The position of all the particles are updated according to the following equation

$$x_{i,j}(t) = x_{i,j}(t-1) + v_{i,j}(t) \quad (3)$$

9. The steps 2 to 8 are repeated for M iterations and the output is the best optimal assignment of data blocks

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

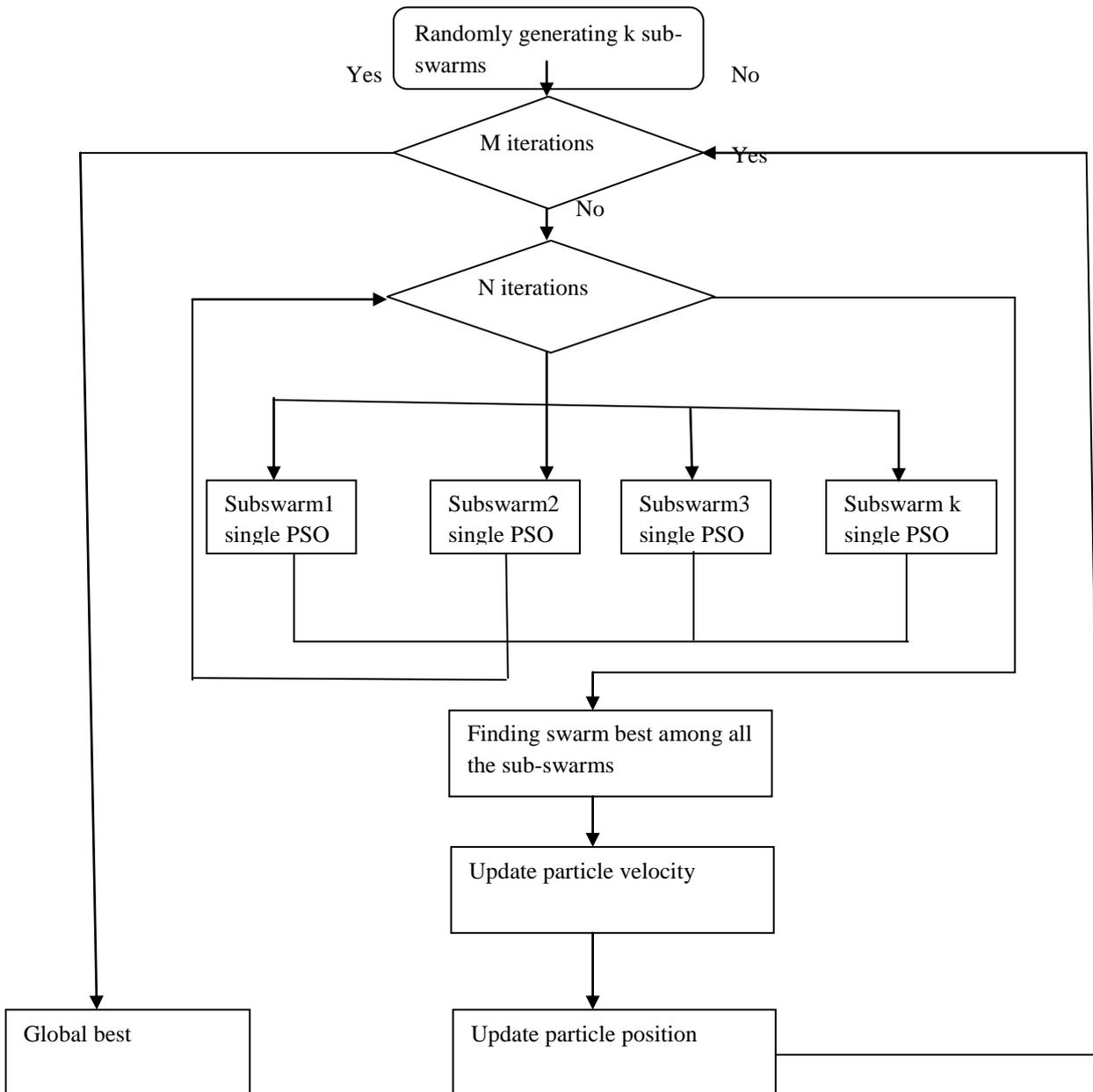


Fig.1Flow diagram of parallel particle swarm optimization

V. EXPERIMENTAL RESULTS

The cloudsim tool is used to conduct the experiments which simulate the cloud environment. The particle swarm optimization and parallel particle swarm optimization algorithms are used for task scheduling with the same environment, their performance and results are compared. The parameter settings for PSO are inertia weight $w=0.7$, acceleration parameters $c1=1.46$, $c2=1.26$. The parameter settings for PPSO are inertia weight $w=0.7$, acceleration parameters $c1=1.4$, $c2=1.4$ and $c3=1.4$. The population size for both algorithms are 50. The number of iterations is 100. The execution time and results for PPSO is better than PSO.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

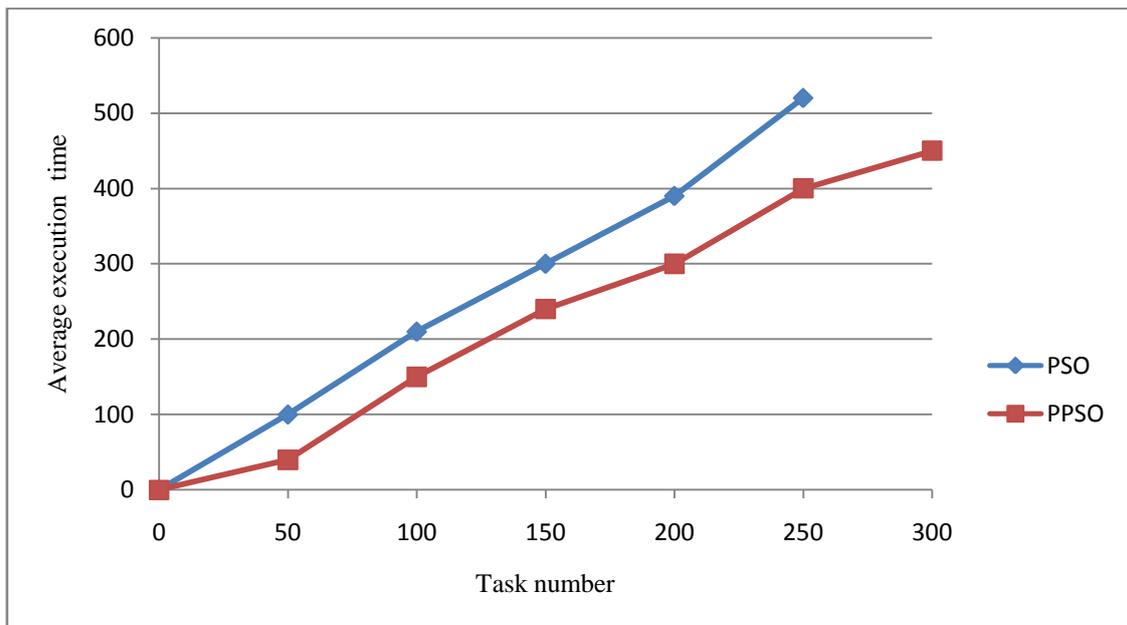


Fig.2 Average execution time of PSO and PPSO

The results prove that parallel particle swarm optimization reduces the execution time of tasks in cloud environment. In future work, the algorithm is opted to minimize costs in addition to minimizing execution time. The dynamic load balancing is also considered in future.

REFERENCES

- [1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Generation Computer Systems*, Vol. 25, No.6, pp.599–616,2009 .
- [2] Cheng-Hong Yang, Sheng-Wei Tsai, Li-Yeh Chuang, and Cheng-Huei Yang, "A modified particle swarm optimization for global optimization", *International Journal of Advanced Computer Technology*, Vol. 3, No.7, pp. 169-189,2011
- [3] Cun-li Song, Xiao-bing Liu, Wei Wang, and Xin Bai, A hybrid particle swarm optimization algorithm for job-shop scheduling problem, *International Journal of Advancements in Computing Technology*, Vol. 3, No. 4, pp. 79-88,2011
- [4] M. Fatih Tasgetiren, Yun-Chia Liang, Mehmet Sevklı, and Gunes Gencyilmaz, "Particle swarm optimization and differential evolution for single machine total weighted tardiness problem", *International of Production Research*, pp.4737-4754,2006
- [5] T. Gonsalves, and A. Egashira, "Parallel swarms' particle swarm optimization", *Proc. of the Applied Computational Intelligence and Soft Computing (ACIS)*, 2013.
- [6] A. Jangra, and T. Saini. "Scheduling Optimization in Cloud Computing" ,*International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 3,pp. 62-65,2013
- [7] J. Kennedy and R.C. Eberhart, "Particle swarm optimization", *Proc. IEEE Int. Conf. on Neural Networks*, Vol. 4, pp.1942– 1948, 1995.
- [8] Lizheng Guo, Shuguang Zhao, Shigen Shen, and Changyuan Jiang, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm", *Journal of Networks*, Vol.7, No.3, 2012.
- [9] S. Pandey, A. Barker, K. K. Gupta, and R. Buyya, "Minimizing Execution costs when using globally distributed cloud services", *24th IEEE International Conference on Advanced Information Networking and Applications*,2010
- [10] A. Salman, "Particle swarm optimization for task assignment Problem", *Microprocessors and Microsystems*, Vol. 26, No.8, pp.363–371, 2002.
- [11] Shaobin Zhan and Hongying Huo, " Improved PSO-based Task Scheduling Algorithm in Cloud Computing", *Journal of Information and Computational Science*(3821–3829) , Vol.9,No.13 ,2012
- [12] Y. Shi, R.C. Eberhart, "Empirical study of particle swarm optimization", *Proc. IEEE Congr. Evol. Computing*, pp.1945-1950.
- [13] K. Sunitha, and Mrs. P.V. Sudha, "An efficient task scheduling in distributed computing systems by improved genetic algorithm", *International Journal of Communication Network Security*, Vol.2, No.2, 2013
- [14] Yu-dong Zhang, Le-nan Wu, "A hybrid TS-PSO optimization algorithm", *Journal of Convergence Information Technology, JCIT*, Vol. 6, No. 5, pp. 169-174 .2011
- [15] Zahraa Tarek, Magdy Zakria and Fatma A. Omara, " Pso Optimization algorithm for Task Scheduling on The Cloud Computing Environment", *International Journal of Computers and Technology*, Vol. 13, No. 9,2014
- [16] L. Zhang, Y.H. Chen, R.Y Sun, S. Jing, and B. Yang, "A task scheduling algorithm based on PSO fro Grid Computing", *International Journal of Computational Intelligence Research*, pp.37-43, 2008.
- [17] Zhi-hui Zhan, Jun Zhang, Yun Li, and Henry Chung, "Adaptive particle swarm optimization", *IEEE Transactions on System, Man, and Cybernetics*, Vol. 39, No. 6, pp. 1362-1381,2006