ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Workflow Scheduling Using Hybrid Discrete Particle Swarm Optimization (HDPSO) In Cloud Computing Environment

Purnima Devi[1], Mala kalra[2]

[1]PG Scholar, Dept. of Computer Science and Engineering, NITTTR, Panjab University, Chandigarh, India

[2]Assistant Professor, Dept. of Computer Science and Engineering, NITTTR, Panjab University, Chandigarh, India

ABSTRACT*:* Most of the scientific applications nowadays are represented in the form of workflows. The tasks in these workflows require lot of computation power and communication between tasks. Cloud Computing is the most recent technology which provides different computing resources dynamically based on Pay-Per-Use model. So cloud is an ideal platform for executing these workflows. As the mapping of workflow tasks to resources is an NP-hard problem, workflow scheduling is quite challenging in this environment. So, the objective of this research is effective scheduling of workflows in cloud environment. Particle Swarm Optimization has been extensively used in literature survey for solving many optimization problems. Particle Swarm Optimization algorithm gives quicker convergence and obtains closer solutions. To get better performance further, we have used Min-Min algorithm to create initial population. So this research proposes a hybrid of PSO and min-min algorithm for solving workflow scheduling problem in cloud environment focusing on makespan.

KEYWORDS*:* Cloud Computing, Workflow Scheduling, Hybrid DPSO, Makespan.

I. Introduction

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud Computing includes computational and storage services as pay-as-you-go model. In Cloud computing hardware and software are managed by third party and these services can be used by individuals or for businesses. Cloud Computing is very fascinating to business holders as it eliminates the overhead of provisioning and permits the businesses to begin from small and expand their resources only if there is a huge rise in demand of services [1][2]. Cloud computing is an expansion of parallel computing, distributed computing and grid computing. A cloud system can be deployed using models known as public [4], private [3], and community cloud and Hybrid clouds.

A. Workflow Scheduling

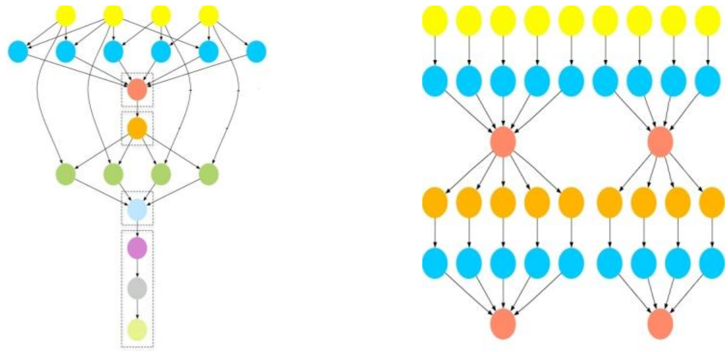
Scheduling is the process of finding out the appropriate resources to allocate to the tasks or jobs. Scheduling is done effectively by taking into mind certain QoS constraints such as budget constraints, deadline constraints, and high throughput. According to a simple classification [5], job scheduling algorithms can be categorized into two main groups; Batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithm. In BMHA, Jobs are queued and collected into a set when they arrive in the system. By On-line mode heuristic scheduling algorithm, jobs are scheduled when they arrive in the system. As the cloud environment is heterogeneous, and the speed of each processor varies frequently, the On-line mode heuristic scheduling algorithms are best for a cloud environment. Scheduling can be broadly classified into two categories: independent task scheduling and workflow scheduling. Independent tasks are those which don't require any correspondence between the tasks, there is no relation between tasks. Each task executes independently on cloud resources. In workflow scheduling we have to follow some order to execute workflow tasks, we cannot provide virtual machine to the child task before completion of its parent task. To decide appropriate workflow scheduling method task dependency plays an important role. Figure 1 shows the structure of two scientific workflow applications: Montage and Inspiral.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015



(a) **(b)**

Figure 1 DAG of Montage (a) and Inspiral (b) Workflows [6]

II. RELATED WORK

Scheduling dependent tasks in a workflow application also become very difficult as mapping of the tasks is done by considering the precedence constraints between various tasks [7].This is done to make execution faster, reduce the cost for the customers and make full utilization of compute resources. Research work in [8] discussed an algorithm to reduce the expectation time for searching targets. A traditional PSO algorithm with a random factor tackle premature convergence problem, and achieve a significant improvement in multi-robot system. Results demonstrate the feasibility, robustness, and scalability of proposed method than previous methods. Total time taken to complete the tasks is less and stable than traditional PSO-based method which results in low power consumption. Author in [9] introduced a new algorithm called DVFS-MODPSO for workflow scheduling in distributed environments such as cloud computing infrastructures. DVFS-MODPSO simultaneously optimizes several conflicting objectives namely, the makespan, cost and energy in a discrete space. Abdi et al. [10] proposed a Modified Particle Swarm Optimization (MPSO) in which they generated initial population using SJFP to improve its performance. The performance of proposed work is illustrated using different QoS parameters computation cost and makespan. Authors in [11] used PSO for scheduling tasks in cloud environment. This process reduced execution time and improves resource utilization. Ant-colony optimization [12] is combined with min-min to perform scheduling in grid environment focusing on makspan. Authors proposed a hybrid approach by combining DPSO and min-min to perform task scheduling in grid environment, which reduce makespan and flow time [13]. The authors proposed GHPSO [14] by combining crossover and mutation of genetic algorithm with PSO, which helped in achieving better performance than standard PSO in terms of cost and makespan.

III. PROPOSED WORK

Workflow scheduling is a major challenging issue in cloud computing. Formally, a workflow is represented by a directed acyclic graph (DAG), G (T, E), where T represents finite set of tasks of workflow and E denotes the edges between the tasks of the workflow. The main aim of the cloud workflow scheduling process is to map each task to a suitable resource with an objective of optimizing the makespan of the workflow. Since it is not possible to find all possible mappings in a reasonable time, we try to reach near the best solution. Cloud workflow scheduling with an optimum makespan is an N-P complete problem. Therefore, the purpose of workflow scheduling process is reformulated to discover the task-to-resource mappings for each task of a workflow by approximating the solution.

From the literature survey, we have found that a lot of work has been done in area of workflow scheduling in cloud environment. Hybrid Discrete PSO algorithm has been used to solve independent task scheduling in grid environment [13]**.** In this work, we are going to use HDPSO to resolve workflow scheduling problem in cloud environment. When PSO algorithm is applied for scheduling of jobs, it gives faster convergence and obtains quicker solutions [15]. For most global optimal problems PSO works better. It takes less computation time to get same or even better results from already existing algorithms. Figure 2 shows the proposed algorithm.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Algorithm HDPSO

1. **Inputs :** Any Workflow
2. **Output :** Execution Schedule
3. Generate initial population using Min-Min algorithm.
4. Apply fitness function to evaluate each particle in initial population.
5. Find out best position of each particle and global best position of particles, pbest and gbest respectively for the initial population.
6. Update the velocities $V_k(t)$ for every particle at the current iteration according to Eq. 1.
   $V_k(t+1) = w*V_k(t) + r_1.c_1.[P_{best,k}(t) - X_k(t)] + r_2.c_2[G_{best}(t) - X_k(t)]$ $\quad k=1,2,3….n \quad (1)$
   Where k is the number of particles, $P_{best,k}(t)$ is the position of the particle with the best fitness value at the previous iteration, $X_k(t)$ is the position of particle at the previous iteration and $G_{best}(t)$ is global best position among all the particles, c_1 and c_2 are constant values, r_1 and r_2 are random values between 0 and 1.
7. Update the position of particle according to Eq. 2.
   $X_k(t+1) = X_k(t) + V_k(t)$ $\quad k=1,2,3….n \quad (2)$
8. Stop if maximum number of iterations is reached and output best solution else goes to step 4.

Figure 2 Proposed Algorithm

The Flow chart of proposed algorithm is shown in Figure 3.

A. Fitness Function

In this research, the fitness function is based on makespan, i.e. the completion time of last task. For each machine we add execution time of all tasks assigned to that particular machine and the largest runtime of all machines will represent makespan .Makespan = max{Fi} where Fi is the finish time of the last task.

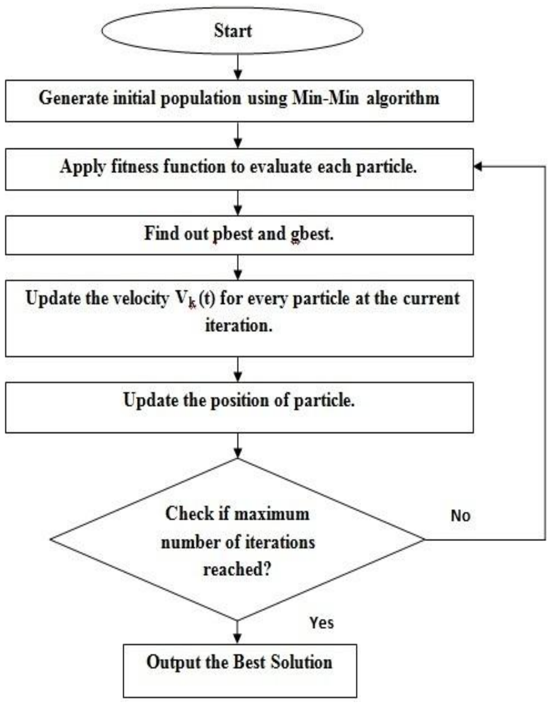


Figure 3: Flowchart of Proposed Algorithm

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

IV. SIMULATION EXPERIMENTS

A. Experimental Environment and Parameter Settings

In this section, we will discuss the experimental setup for evaluating the performance of proposed algorithm. The experiment is performed in a simulated environment using workflowSim simulator [16] with Eclipse IDE. For simulation we need to set some parameters for WorkflowSim and HDPSO, Table 1 shows the parameters setting for HDPSO and Table 2 shows parameters settings for WorkflowSim toolkit. Table 3 shows the specification of VMs.

Table 1 Parameter Settings for HDPSO

Parameters	Values
Population size	100
Maximum iteration	10
Inertia Weight (W)	0.5
Acceleration Coefficient (C1)	1
Acceleration Coefficient (C2)	1
Maximum Velocity (Vmax)	4

Table 2 Parameter Settings for WorkflowSim

Type	Parameters	Values
Datacenter	Number of Datacenters	1
	Number of Hosts	20
	Type of Manager	Time- Shared
VM	Number of VMs	5
	MIPS	400-2000
	VM memory (RAM)	512
	Bandwidth	1000
	Policy	Time-Shared
Task	Number of Tasks	10 to 1000

B. Result Analysis

We consider Montage and Inspiral real workflows for performance analysis of proposed algorithm. The proposed algorithm is compared with PSO in terms of makespan. Table 4 and 5 shows the results for montage and inspiral workflow respectively. Figure 4 and 5 shows the graphical representation of achieved results for montage and inspiral workflow respectively. Table 6 shows the result of varying number of VMs and keeping number of tasks constant, its graphical representation is shown in Figure 6.

Table 3 Specification of VMs (5)

VMs(5)	Processing Speed (MIPS)
VM1	400
VM2	800
VM3	1200
VM4	1600
VM5	2000

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Table 4 Comparison of proposed algorithm with Discrete PSO for Montage Workflows

Workflow	Number of Tasks	HDPSO	PSO
Montage_25	25	140.208	152.349
Montage_50	50	257.442	275.561
Montage_100	100	586.786	664.601
Montage_1000	1000	6017.119	6299.851

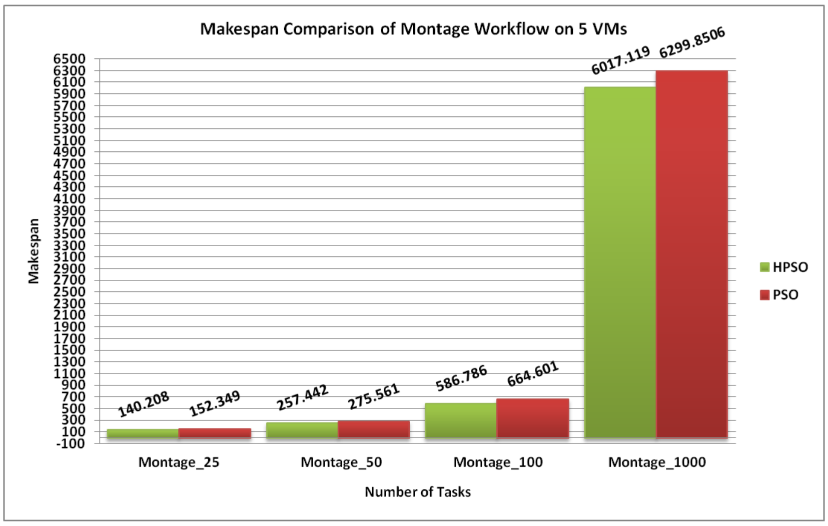


Figure 4 Performance Analysis of HDPSO and PSO for Montage Workflow with 5 VMs

Table 5 Comparison of proposed algorithm with PSO for Inspiral Workflows

Workflow	Number of Tasks	HDPSO	PSO
Inspiral_30	30	3583.789	4463.851
Inspiral_50	50	6918.604	7647.716
Inspiral_100	100	13979.26	14477.67
Inspiral_1000	1000	115327.2	120209.6

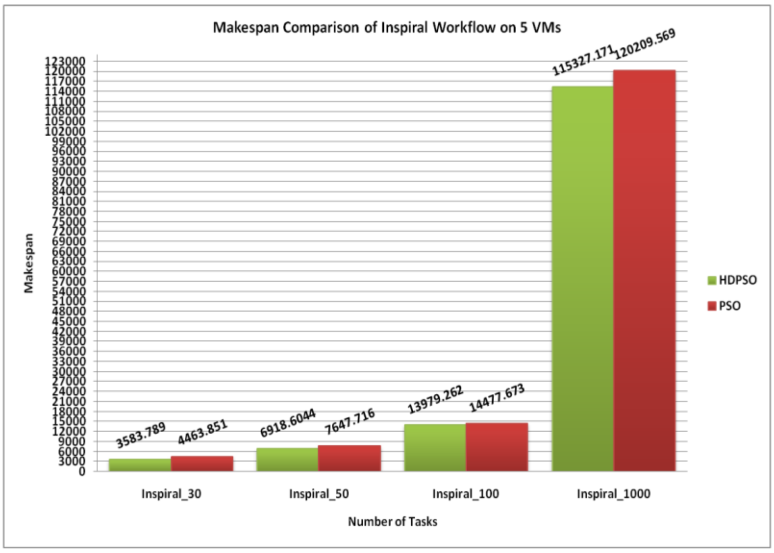


Figure 5 Performance Analysis of HDPSO and PSO for Inspiral Workflow considering 5 VMs

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Table 6 Makespan for Fixed Number of Tasks (Inspiral_50) and Varying no of VMs.

Number of VMs	HDPSO	PSO
5	6918.604	7647.716
10	5941.963	6532.496
15	4950.368	5209.952
20	785.5525	798.8067

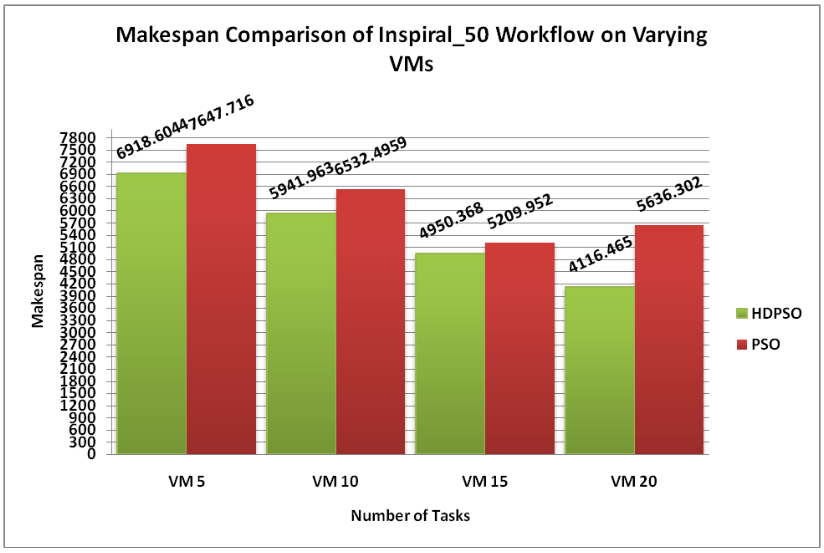


Figure 6 Performance Analysis of HDPSO and PSO for Inspiral_50 Workflow varying number of VMs

Simulation results show that HDPSO gives lesser makespan as compared to PSO. From the above Tables and Graphs we can observe that HDPSO performs better then PSO in all three cases, (i) when considering Montage Workflow (ii) when using Inspiral workflow (iii) when varying VMs for Inspiral workflow keeping number of tasks constant. PSO has a drawback that its local search capability is not good which is overcome in propose work by combining Min-Min with PSO.

V. Conclusion And Future Work

Cloud workflow scheduling with an optimum makespan is an N-P complete problem. Therefore, the purpose of workflow scheduling process is to reformulate the task-to-resource mappings for each task of a workflow by approximating the solution. This research work presents a workflow scheduling algorithm based on hybrid of DPSO and Min-Min to minimize the total makespan of the tasks in workflow submitted to the cloud computing environment. Simulation results show that HDPSO gives lesser makespan as compared to PSO. From the results we can observe that performance of HDPSO is better than PSO.
We focused on only one QoS parameter that is makespan. This research work can be extended by adding more QoS parameters such as cost, load balancing and reliability etc. Moreover we used simulation environment for performing experiments, in future this work can be implemented on real cloud.

REFERENCES

[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," Journal of Internet Services and Applications, Springer, pp. 7-18, April 2010.
[2] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A.Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research," Journal of Internet Computing, IEEE, Volume 13, No. 5, pp. 10-13,
[3] L. Savu, "Cloud Computing: Deployment Models, Delivery Models, Risks and Research Challenges," International Conference on Computer and Management (CAMAN), pp. 1-4, May 2011.
[4] P. Metri, and G. Sarote. "Privacy Issues and Challenges in Cloud Computing," International Journal of Advanced Engineering Sciences and Technologies (IJAEST), Volume 5, pp.1-6, 2011.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

[5] Y. H. Lee, S. Leu, and R. S. Chang, "Improving Job Scheduling Algorithms in a Grid Environment," Future Generation Computer Systems, Elsevier, Volume 27, No. 8, pp. 991–998, October 2011.

[6] https://confluence.pegasus.isi.edu/display/pegasus/ WorkflowGenerator-/(Last accessed on: 15.04.2015)

[7] L. Liu, M. Zhang, Y. Lin, L. Qin, "A Survey on Workflow Management and Scheduling in Cloud Computing", IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 838-846, 2014.

[8] Y. Cai, Z. Chen, and H. Min, "Improving Particle Swarm Optimization Algorithm for Distributed Sensing and Search," IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 373 - 379, October 2013.

[9] S. Yassa, R. Chelouah, H. Kadima, and B. Granado, "Multi-Objective Approach for Energy-Aware Workflow Scheduling in Cloud Computing Environments," Hindawi Publishing Corporation The Scientific World Journal, Volume 2013, pp. 1-13,September 2013.

[10] S. Abdi, S. A. Motamedi and S. Sharifian, "Task Scheduling using Modified PSO Algorithm in Cloud Computing Environment," International Conference on Machine Learning, Electrical and Mechanical Engineering, Dubai (UAE), pp. 37-41, January 2014.

[11] G. Zhao, "Cost - Aware Scheduling Algorithm Based on PSO in Cloud Computing Environment," International Journal of Grid and Distributed Computing, Volume 7, No.1, pp. 33-42, March 2014.

[12] D. Kaur, and S. Singh, "An Efficient Job Scheduling Algorithm using Min-Min and Ant Colony Concept for Grid Computing," International Journal Of Engineering And Computer Science, Volume 03, No. 07 , pp. 6943-6949 , July 2014.

[13] M. Karimi, and H. Motameni , "Tasks Scheduling in Computational Grid using a Hybrid Discrete Particle Swarm Optimization," International Journal of Grid Distribution Computing, Volume 6, No.2, pp. 29-38, April 2013.

[14] S. J. Xue, and W. Wu, "Scheduling Workflow in Cloud Computing Based on Hybrid Particle Swarm Algorithm," Indonesian Journal of Electrical Engineering, TELKOMNIKA Volume 10, No. 7, pp. 1560-1566, November 2012.

[15] K. Vivekanandan, and D. Ramyachitra, "A study on scheduling in grid", International Journal on Computer Science and Engineering," Volume 3, No. 2, pp. 571-584, February 2011.

[16] W. Chen, E. Deelman, "WorkflowSim: A toolkit for simulating scientific workflows in distributed environments", IEEE 8th International Conference on E-Science (e-Science), pp. 1 – 8, Oct. 2012.