

(A High Impact Factor, Monthly, Peer Reviewed Journal) Visit: <u>www.ijirset.com</u> Vol. 7, Issue 3, March 2018

# Deadline Constrained Cloud Computing Resources Scheduling by hybrid of Particle Swarm Optimization with Ant Colony Optimization

Nancy Tanwar<sup>1</sup>, Gautam Kumar<sup>2</sup> M.Tech Scholar, Dept. of CS, SDDIET, India<sup>1</sup> Asst. Professor, Dept. of CS, SDDIET, India<sup>2</sup>

**ABSTRACT:** This paper presents the concepts of three evolutionary algorithms i.e., genetic algorithm, ant colony optimization and particle swarm optimization algorithm. An evolutionary algorithm copies the way how evolution occurs in the nature. There are various types of evolutionary algorithms. This paper focuses on Genetic, ACO and PSO algorithms. Genetic algorithm provides solution to various optimization problems. It follows the principle of "survival of the fittest". Various problems such as knapsack problem, TSP (travelling salesman problem) can be solved using genetic algorithm. Ant colony optimization is a heuristic algorithm which follows the behaviour of ants i.e., the way ants seek food in their environment by starting from their nest. Particle swarm optimization algorithm(PSO) is also an optimization algorithm which also uses a method of searching using some heuristics.

### I. INTRODUCTION

With rapid growth in Information Technology, more and more workflow systems are adopting cloud as their execution environment. It becomes progressively competitive on how to efficiently manage various workflows. Workflow is one of the most challenging problems in cloud to execute it while minimizing time as well as cost incurred by set of heterogeneous resources over the cloud simultaneously.

Nowadays, Cloud computing is a growing area in distributed computing that deliver dynamically adaptable services on demand over the internet through virtualization of hardware and software. The biggest advantage of the cloud is its flexibility to lease and release resources as per the user requirement. Furthermore, the cloud provider offers two types of plans namely short-term plan on demand and long-term reservation plan. It has got intelligent infrastructure i.e. Transparency, Scalability, Monitoring and Security

Scheduling of workflows require huge computation and communication cost. It is the process of mapping inertdependent tasks on the available resources such that workflow application can complete its execution with user defined quality of service. This work target random workflow requests overtime, so it must schedule workflow execution without any knowledge of future requests.

#### II. EXISTING WORK

This section gives a brief review of the work carried out by various researchers in this field. Work has been done by researchers in **Workflow on Cloud.** Various aspects of the problem were studied. Performance effective and low complexity task scheduling for heterogeneous computing [9] Haluk et al. has proposed two approaches for scheduling a workflow task in heterogeneous computing environment namelyHEFT and Critical path on a Processor. They work on the same line with slight differences. HEFT uses the upward rank for task prioritization; the other uses the combination of upward and downward ranks. The latter uses the critical path and assigns the tasks on the critical path to the processor which will give minimum EFT. HEFT is better than other algorithms in same domain because of its high



(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

### Vol. 7, Issue 3, March 2018

efficiency in terms of makespan and robust nature. A Hybrid Heuristics- Genetic Algorithm for Task Scheduling in Heterogeneous Processor Networks H2GS [7]

**Kaur et al.** proposed a new Modified Genetic Algorithm for scheduling the tasks in private cloud for minimizing the makespan and cost. In MGA, initial population is generating using SCFP (Smallest cloudlet to Fastest Processor), LCFP (Longest cloudlet to Fastest Processor) and 8 random schedules. Two-point crossover and simple swap are used. This gives the good performance under the heavy loads. A Performance Effective Genetic Algorithm for Task Scheduling in Heterogeneous Systems (PEGA) [17] Ahmad et al. proposed an effective genetic algorithm called PEGA, which is capable of providing the optimal results in large space with less time complexity. The direct chromosome representation is used having two parts. The right half is made by the b-level (upward rank) which give the better results in terms schedule length when compared with randomly generated population. Two-fold cross over is used in which single and two-point crossover are executed one after the other to enhance the quality and the convergence speed of the solution. The author has concluded that the PEGA provides the better schedule with smaller makespan and low time complexity.

**Arabnejad et al.** proposed a new list-based scheduling technique named PEFT for heterogeneous distributed computing which gives the better results than HEFT in terms of makespan. It has the same time complexity as that of HEFT. It consists of task prioritizing phase and processor selection phase. Most of the list-based scheduling allocates a task to the processor on which it gives the smallest EFT for the current task. They do not consider the impact of children on the current task and hence sometimes lead to the poor decisions. Optimistic EFT is calculated to assign a processor for a task. The aim is to ensure that the children tasks of the current task are finished earlier. The author has shown that PEFT does not give good results if all the tasks lies on critical path. Deadline based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds [22]

**Maria et al.** have used PSO metaheuritic technique to optimize the scientific workflow on IaaS cloud. The authors have incorporated the basic features of IaaS cloud such as unlimited heterogeneous resources, flexibility, dynamic provisioning of resources, VMs boot time as well as leased time and performance variation of VMs. The idea is to combine the resource provisioning along with the scheduling strategy with a objective to minimize the execution cost while staying in the defined deadline. Task scheduling using NSGA II with Fuzzy Adaptive Operators for Computational Grids [23]

#### III. PROPOSED TECHNIQUE PRESENT WORK

#### 4.1 Basic ACO Algorithm

The Ant algorithm was introduced by Dorigo M. in 1996 based on the real ant behavior and it's a new heuristic algorithm to solve combinational optimization problems. Investigations have shown that ants have the ability to find food in an optimal path between the food and nest. With the ant motion some pheromone is released on ground, previous laid trail is encountered by isolated ant is being detected and follow with a higher probability. The ant's probability of choosing the way depends upon the pheromone concentration on that way. Higher the pheromone concentration, higher will be the probability of that way adoption. An optimal way can be found by utilizing this positive mechanism of feedback. The steps of ant colony optimization algorithm are given below:

#### Algorithm 1

Step 1: Parameters is set; pheromone trails are initializing.Step 2:On path segments, theVirtual trail is accumulated.Step 3: ACO - Construct Ant SolutionsFrom node i to node j an ant will move with probability



(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: <u>www.ijirset.com</u>

Vol. 7, Issue 3, March 2018

$$\mathbf{P}_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}$$

Where,

On edge i, j the amount of pheromone is  $\tau_{i,j}$ To control the influence of  $\tau_{i,j}\alpha$  is a parameter In edge i, j (typically  $1/d_{i,j}$ )  $\eta_{i,j}$  is the desirability To control the influence of  $\eta_{i,j}\beta$  is a parameter **Step 4:** ACO - Pheromone Update According to the equation amount of pheromone is updated  $\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta \tau_{i,j}$ Where, On a given edge i, j the amount of pheromone is  $\tau_{i,j}$   $\rho$  is the rate of pheromone evaporation is  $\rho$ The amount of pheromone deposited is  $\Delta \tau_{i,j}$ , typically given by  $\Delta \tau_{i,j}^{k} = \begin{cases} \frac{1}{L_{k}} & \text{if ant k travels on edge i, j} \\ 0 & \text{otherwise} \end{cases}$ 

Where,

The cost of the  $k^{th}$  ant's tour (typically length) is  $L_k$ .

#### 4.2 Proposed Workflow Scheduling ACO Algorithm

With the above given ant algorithm characteristic utilization, the task can be scheduled. Similarly, new task can be carried out with the utilization of previous task scheduling result. The basic ideas of ACO algorithm is inherited in Workflow Scheduling-ACO algorithm for the reduction of execution time and cost.

Firstly, input the workflow to the workflow simulator and parse the task from this workflow. Pareto distribution is followed by the task. On VMs, there is a pareto distribution of ants at the beginning and then,  $VM_i$  pheromone values are initialized:

 $VM_b_i$ 

$$\tau_i(0) = p_N U M_i \times p_M I P S_i +$$

Where

 $p_NUM_i \leftarrow$  number of VM<sub>i</sub> processor

 $p_MIPS_i \leftarrow$  million instructions per second of each VM<sub>i</sub> processor

 $VM_b_i \leftarrow VM_i$  communication bandwidth ability

Choosing VMs rule for next task: For next task, VM<sub>i</sub> choose by k-ant with probability defined as:

$$P_{i}^{K}(T) = \begin{cases} \frac{|\tau_{i}(T)|^{\alpha}|c_{i}|^{\beta}|lb|^{\gamma}}{\sum[\tau_{K}(T)]^{\alpha}[c_{K}]^{\beta}[lb]^{\gamma}} & \text{if } i \in 1, 2 \dots \dots n \\ 0 & \text{otherwise} \end{cases}$$

Where

 $\tau_i(T) \leftarrow$  pheromone value of VM<sub>i</sub> at time T  $c_i \leftarrow$  VM<sub>i</sub> computing capacity

 $C_i$  can be defined as:

$$c_i = p_N U M_i \times p_M I P S_i + V M_b_i$$

 $lb_i \leftarrow VM_i$  load balancing factor for minimizing the degree of imbalance defined as:

$$lb_i = 1 - \frac{et_i - Avg_et}{et_i + Avg_et}$$

Where

 $Avg_et \leftarrow$  virtual machine average execution time in the optimal path last iteration  $et_i \leftarrow$  expected execution time of VM<sub>i</sub> task



(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

### Vol. 7, Issue 3, March 2018

 $et_i$  is defined as:

 $et_i = \frac{total\_TL}{c_i} + \frac{Input\_FS}{VM\_b_i}$ 

Where

 $total_TL \leftarrow$  total length of task submitted to VM<sub>i</sub>

 $Input_FS \leftarrow$  task length before execution

 $\alpha, \beta$  and  $\gamma \leftarrow$  parameters controlling the relative weight of pheromone trail along with VMs computing capacity and load balancing. Once heavily loaded are some VMs becoming bottleneck in cloud influences the given task set makespan. The load balancing factor lb<sub>i</sub> is defined in the ant algorithm for improving the capacity of lead balancing. Bigger the lb<sub>i</sub>, higher will be the probability of choosing means VM<sub>i</sub> comprehensive ability is greater now.

Updating Pheromone: Ant Let  $\tau_i(T)$  at any time T be the VM<sub>i</sub> pheromone intensity. The update of the pheromone is given by:

$$\tau_i(T+1) = (1-\rho) \times \tau_i(T) + \Delta \tau_i$$

Where

 $\rho \in [0,1]$   $\leftarrow$  decay coefficient of pheromone trail

The past solution impact will be less if value of  $\rho$  is greater. The  $\Delta \tau_i$  value is defined as:

After the completion of ant tour, updating the local pheromone on VM visited and  $\Delta \tau_i$  value is given as:

 $\Delta \tau_i = 1/t_{iK}$ 

Where

 $t_{iK} \leftarrow$  K-ant searched shortest path length at i<sup>th</sup> iteration

In case, the current optimal solution is found by the ant while completing its tour, larger intensity pheromone is laid on its tour and updating the global pheromone on VM visited and  $\Delta \tau_i$  value is given as:

$$\Delta \tau_i = d/t_{oi}$$

Where

 $t_{op} \leftarrow$  current optimal solution

d←encouragement coefficient

If the function is optimized, then we analysis the cost and time of that function.



Fig shows System model



(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 3, March 2018

### **IV. SIMULATIONS & DISCUSSIONS**

RESULTS OF BAT AND ACO USING -SIPHT							
Ensemble	BAT			ACO			
Size	TET	TEC	Response Time	TET	TEC	Response Time	
2	24.06	5723.855	0.01543485	4.59	4584.063	0.03872953	
4	40.16	10011.35	0.02542084	11.26	8605.69	0.06121714	
6	39.04	13521.19	0.0222077	15.13	12200.43	0.05127205	
8	55.98	13404.39	0.02757845	32.83	14277.19	0.05764191	
10	89.61	17549.09	0.02727286	27.77	16900.17	0.0473407	
12	64.5	20561.6	0.02593575	34.34	19939.58	0.04683594	
14	54.64	21713.64	0.02712497	34.06	22367.37	0.05854424	
16	61.21	25855.99	0.02958881	53.59	30163.12	0.06301173	
18	93.3	33493.56	0.03669985	74.36	33303.33	0.05073513	
20	71.03	34154.99	0.03308249	60.55	33057.34	0.05341777	

Table 4.1: Comparison table of BAT and ACO using SIPHT









Figure 4.3: Comparison graph of Response time of BAT and ACO using SIPHT



(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: <u>www.ijirset.com</u>

Vol. 7, Issue 3, March 2018

RESULTS OF BAT AND ACO USING -MONTAGE						
Ensemble	BAT			ACO		
			Response			Response
Size	TET	TEC	Time	TET	TEC	Time
2	0.0	0.0	0.0	0.0	0.0	0.0
4	17.09	447.3718	0.00036851	1.73	299.958	0.00346107
6	19.69	628.3048	0.00071784	1.82	356.9556	0.00483133
8	21.95	730.9887	0.01030986	3	496.4608	0.08346628
10	28.47	1093.207	0.01084365	3	496.4608	0.08346628
12	28.89	1017.366	0.01437245	6.54	1147.339	0.08482576
14	31.15	1329.824	0.01859333	7.61	1131.746	0.09339701
16	33.2	1288.644	0.01599708	10.52	1480.906	0.09690665
18	34.07	1538.399	0.02065969	8.66	1278.957	0.10229598
20	35.3	1418.877	0.01837339	10.4	1745.794	0.11267924

Table 4.2: Comparison table of BAT and ACO using MONTAGE



Figure 4.4: Comparison graph of TET of BAT and ACO using MONTAGE



Figure 4.5: Comparison graph of Response time of BAT and ACO using MONTAGE



(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: <u>www.ijirset.com</u>

### Vol. 7, Issue 3, March 2018

RESULTS OF BAT AND ACO USING -GENOME						
Ensemble	BAT			ACO		
			Response			Response
Size	TET	TEC	Time	TET	TEC	Time
2	137.78	7549.552	0.00008362	23.2	9915.513	0.00009375
4	201.57	42511.93	0.00063674	73.24	27030.74	0.00122507
6	370.15	28155.45	0.00013048	133.83	42569.68	0.00018726
8	284.46	58872.21	0.00132634	315.65	41656.28	0.00136613
10	365.5	77034.56	0.00151176	745.24	86234.89	0.00171455
12	423.76	66004.98	0.00141466	546.78	88012.68	0.00153053
14	486.7	82720.56	0.00147464	317.76	93654.41	0.00243139
16	530.13	113448.6	0.00167419	401.95	106579.9	0.00171186
18	419.1	119935.5	0.00157754	791.04	123916.8	0.00167645
20	746.6	138725.4	0.00113138	491.21	139920	0.00214002

 Table 4.3: Comparison table of BAT and ACO using GENOME



### Figure 4.6: Comparison graph of TEC of BAT and ACO using GENOME



Figure 4.7: Comparison graph of Response time of BAT and ACO using GENOME



(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: <u>www.ijirset.com</u>

Vol. 7, Issue 3, March 2018

RESULTS OF BAT AND ACO USING -CYBERSHAKE							
	BAT			ACO			
Ensemble			Response			Response	
size	TET	TEC	Time	TET	TEC	Time	
2	0	0	0	0	0	0	
4	19.57	822.6585	0.00033389	2.6	687.9981	0.00239637	
6	25.25	1136.987	0.00076269	2.36	950.981	0.00397693	
8	27.05	1545.962	0.00624708	5.92	1518.366	0.03944654	
10	28.85	1883.18	0.01026533	4.87	1689.733	0.08423868	
12	31.78	1729.39	0.00984849	8.08	1724.117	0.05825278	
14	36.95	2530.267	0.01088677	9.96	2463.681	0.06189383	
16	34.34	2106.646	0.01465981	9.25	2731.877	0.06826968	
18	38.77	2311.253	0.01292667	11.78	1989.96	0.05372413	
20	39.69	2541.908	0.0086382	10.62	1982.903	0.03366692	

Table 4.4: Comparison table of BAT and ACO using CYBERSHAKE







Figure 4.9: Comparison graph of Response time of BAT and ACO using CYBERSHAKE



(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: <u>www.ijirset.com</u>

Vol. 7, Issue 3, March 2018

RESULTS OF BAT AND ACO USING -LIGO							
Ensemble	BAT			ACO			
			Response			Response	
size	TET	TEC	Time	TET	TEC	Time	
2	0.0	0.0	0.0	0.0	0.0	0.0	
4	45.91	2762.633	0.00044014	7.61	1467.243	0.0007262	
6	66.89	4191.678	0.00059422	25.09	4022.422	0	
8	36.66	4883.178	0.00526892	44.34	5699.963	0.00994162	
10	69.82	5304.774	0.00589453	16.98	4708.369	0.01883875	
12	101.39	7331.103	0.00617638	28.65	6271.538	0.01486582	
14	59.96	7729.967	0.00749991	34.7	7550.524	0.01505755	
16	150.38	10608.93	0.00524406	48.58	9948.208	0.01277547	
18	89.77	11384.08	0.00669903	70.95	11188.03	0.01097674	
20	151.22	13094.86	0.00582672	89.9	12582.79	0.00994354	

Table 4.5: Comparison table of BAT and ACO using LIGO



Figure 4.14: Comparison graph of TET of BAT and ACO using LIGO





### **V. CONCLUSION**

In this paper, a review to three evolutionary algorithms, i.e., Genetic algorithm, Particle Swarm Optimization(PSO)andAnt colony optimization(ACO) is given. Genetic Algorithm copies the way evolution occurs in the nature, while ACO is based on the behaviour of ants and PSO is based on the behaviour of flock of birds and fishes. All these algorithms can be used to solve various optimization problems



(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

#### Vol. 7, Issue 3, March 2018

#### REFERENCES

[1] O. Zhang, L Cheng and R. Boutaba, "Cloud computing: state-of-the-art and research challenges", Journal of internet services and applications, Springer, vol. 1, no.1, pp. 7-18, May 2010.

[2] K. Bessai, S. Youcef, A. Oulamara, C. Godart and S. Nurcan, "Bi-criteria workflow tasks allocation and scheduling in cloud computing environments", IEEE 5th International Conference on Cloud Computing (CLOUD), pp. 638-645, June 2012.

[3] Mell, Peter, and Tim Grance, "The NIST definition of cloud computing", Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, pp. 20-23, 2011.

[4] E.N. Alkhanak, S.P. Lee, R. Rezaei and R.M. Parizi, "Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues", Journal of Systems and Software, Elsevier, vol. 113, pp. 1-26, March 2016.

[5] M.I. Daoud and N. Kharma, "A high performance algorithm for static task scheduling in heterogeneous distributed computing systems", Journal of Parallel and distributed computing, Elsevier, vol. 68, no. 4, pp. 399-409, April, 2008. [6] G. Wang, Y. Wang, H. Liu and H. Guo, "HSIP: A Novel Task Scheduling Algorithm for Heterogeneous Computing", Scientific Programming, Hindawi Publishing Corporation, March 2016.

[7] M. I. Daoud and N. N. Kharma, "A hybrid heuristic-genetic algorithm for task scheduling in heterogeneous processor networks", Journal of Parallel and Distributed Computing, Elsevier, vol. 71, no. 11, pp. 1518–1531, 2011.
[8] Y. Xu, K. Li, J. Hu and K. Li, "A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues",

Information Sciences, Elsevier, vol. 270, pp. 255 - 287, 2014.

[9] H. Topcuoglu, S. Hariri and M.Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing", IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 3, pp. 260-74, March 2002. 24

[10] S. Smanchat and K. Viriyapant, "Taxonomies of workflow scheduling problem and techniques in the cloud", Future Generation Computer Systems, Elsevier, vol. 52, pp.1-12, November 2015.

[11] X. Liu, D. Yuan, G. Zhang, W. Li, D. Cao, Q. He, J. Chen and Y. Yang, "The design of cloud workflow systems", Springer Science & Business Media, November 2011.

[12] Talbi and E. Ghazali, "Metaheuristics: from design to implementation", John Wiley & Sons. , vol. 74, 2009.

[13] J. Yu, R. Buyya and K. Ramamohanarao, "Workflow scheduling algorithms for grid computing", Metaheuristics for scheduling in distributed computing environments, Springer Berlin Heidelberg, pp. 173-214, 2008.

[14] R.J. Moraga, G.W. DePuy and G.E. Whitehouse,"Metaheuristics: A Solution Methodology for Optimization Problems", Handbook of Industrial and Systems Engineering, CRC Press, FL. 2006.

[15] K.F. Man, K.S. Tang, and S. Kwong, "Genetic algorithms: concepts and applications", IEEE transactions on Industrial Electronics, vol. 43, no. 5, pp. 519-34, October 1996.

[16] S. Kaur and A. Verma, "An efficient approach to genetic algorithm for task scheduling in cloud computing environment", International Journal of Information Technology and Computer Science (IJITCS), vol. 4, no. 10, pp. 74, September 2012.

[17] S. G. Ahmad, E. U. Munir, and M. W. Nisar, "PEGA: A performance effective genetic algorithm for task scheduling in heterogeneous systems", IEEE 14th International Conference on High Performance Computing, pp. 1082–1087, 2012.

[18] C. Wang, J. Gu, Y. Wang, and T. Zhao, "A hybrid heuristic-genetic algorithm for task scheduling in heterogeneous multi-core system", in Algorithms and Architectures for Parallel Processing (Y. Xiang, I. Stojmenovic, B. Apduhan, G. Wang, K. Nakano, and A. Zomaya, eds.), Springer Berlin Heidelberg ,vol. 7439 of Lecture Notes in Computer Science, pp. 153-170, 2012.

[19] S. Abrishami, M. Naghibzadeh, and D.H. Epema, "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds", Future Generation Computer Systems, Elsevier, vol. 29, no. 1, pp. 158-69, January, 2013. 25

[20] A Verma and S. Kaushal, "Deadline constraint heuristic-based genetic algorithm for workflow scheduling in cloud", International Journal of Grid and Utility Computing, Inderscience Publishers Ltd, vol. 5, no. 2, pp. 96-106, January 2014.

[21] Hamid, Arabnejad and JG. Barbosa, "List scheduling algorithm for heterogeneous systems by an optimistic cost table", IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 3, pp. 682-94, March 2014. [22] M.A. Rodriguez and R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds", IEEE

Transactions on Cloud Computing, vol. 2, no. 2, pp. 222-35, April , 2014. [23] R. Salimi, H. Motameni, and H. Omranpour, "Task scheduling using NSGA II with fuzzy adaptive operators for computational grids", Journal of Parallel and Distributed Computing, Elsevier, vol. 74, no. 5, pp. 2333-2350, May 2014.

[24] Delavar, A. Ghorbannia and Y. Aryan, "HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems", Cluster computing, Springer, vol. 17, no. 1, pp. 129-137, 2014.

[25] M. Kalra and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing", Egyptian Informatics Journal, Elsevier, vol. 16, no. 3, pp. 275-295, 2015.

[26] A. Verma and S. Kaushal, "Cost-Time Efficient Scheduling Plan for Executing Workflows in the Cloud", Journal of Grid Computing, Springer, vol. 13, no. 4, pp. 495-506, December 2015.