Scheduling Scientific Workflow Tasks in Cloud Using Swarm Intelligence

V.Vinothina
Deptarment of Computer Science
Kristu Jayanti College Autonomous
Bengaluru, India.
vinothina.v@kristujayanti.com

Abstract— **One of the state-of-the art techniques of Cloud Computing is the type of distributed computing system and derives its features. It has been unanimously used by all the organizations as its yields enormous benefits and its features. The scalability and heterogeneity features make the Cloud most suitable for computing scientific workflow tasks as the workflow comprises thousands of tasks and deals with huge amount of data. Many scheduling algorithms have been proposed using different methods to compute the workflow tasks in cloud with different objectives such as minimal makespan, minimal cost, maximal resource utilization etc. In spite of that this paper proposes an algorithm namely Improved Workflow Scheduling using ACO (IWSACO) with variance in WFSACO (WorkFlow Scheduling using Ant Colony Optimization) using one of the swarm intelligence techniques of ACO to obtain better performance.**

Keywords— Ant Colony Optimization; Workflow Applications; Scheduling; Cloud Computing component; Virtual Machine;

I. INTRODUCTION

The Cloud computing is a kind of new computing paradigm that works for solving the new problem which combines the different computers to constitute a big computing system to execute some large tasks [1]. One of the major challenges in cloud computing is task scheduling and efficient use of cloud resources. Generally, the task scheduling problem is NP-hard and obtaining an optimal solution is a challenging task. The scheduling problem is complex in cloud due to heterogeneous, dynamic and self governing natures of cloud. Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) are the deployment models of cloud allows the users to host their own applications in the cloud using the policy of pay-per-use. Since cost involved in cloud usage, users have to utilize the cloud efficiently.

Workflow applications are common in science and engineering and their structure/composition is known in advance [2]. It may be a service, application or a module. The resource requirement of each task in the workflow depends on its functional capacity and data input. Virtual servers of Cloud reduce the users' cost in purchasing, operating and maintaining a physical computing infrastructure [3]. Moreover, virtualization technology enables the use of multiple virtual machines on one physical machine. This results in more optimization of sharing and better utilization of physical resources [4].

Scientific applications are mainly used by scientists for their research purposes which are made up of coarse-grained and precedence constrained tasks. Scheduling scientific workflow tasks is the problem of scheduling tasks in scientific applications and mapping each task to suitable resources based on some performance impact factors [5]. The factors such as makespan, resource cost and resource utilization mainly depend upon the algorithm and practices used for scheduling and allocation of resources to the tasks.

The nature of ACO (Ant Colony Optimization) algorithm such as robustness and self -adaptability can just match the characteristics of cloud computing. The ACO also used in grid computing task scheduling but doesn't get good performance. This paper proposes certain basic steps for scheduling scientific workflow tasks using ACO in cloud. ACO algorithms are probabilistic models inspired by the social behaviour of ants. Initially ACO has been applied for travelling salesman problem. Since then, it has been successfully applied to NP-hard combinatorial problem [6].

Hence the author's previous work [16] used ACO approach to minimize the completion time of scientific workflow. The objective of this paper is to propose an algorithm to overcome the complexity involved in WFSACO algorithm [16]. However usage of cloud is based on pay-per-use. Therefore, resource cost and resource usage are also considered as important factors affecting the performance. This paper is organized as follows: A few similar works have been discussed in Section II. The proposed scheduling environment has been explained under Section III. The steps for scheduling workflow tasks have been furnished in section IV. The evaluated results are given in section V followed by conclusion in section VI.

978-1-5090-4997-4/17/$31.00 ©2017 IEEE

II. Related Work

Many scheduling algorithms has been proposed for task scheduling in various environments such as single processor

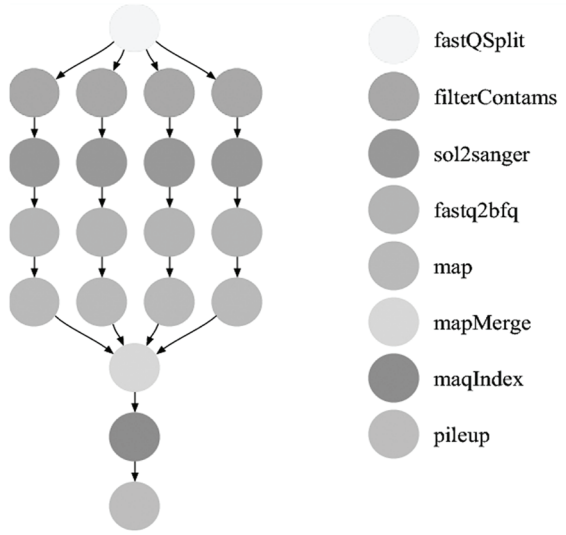


Fig. 1. Montage Scientific Workflow Structure [20]

system, multiprocessor homogeneous system, multiprocessor heterogeneous system, homogeneous distributed system and heterogeneous distributed system with different objectives. In this section, some related works which have been carried out for task scheduling are discussed in short.

Achar et.al [15] proposed an optimal scheduling algorithm which used the tree based data structure of virtual machine tree (VMT). The simulated results provide better results than traditional algorithm. The PBTS (Partitioned Balanced Time Scheduling) algorithm proposed by Byun*et al*. [7] schedules the tasks with the objective of cost and execution budgets. The proposed algorithm however focus on parallelizable workflow tasks with precedence constraint, for which much fine- grained time information is needed. The algorithms based on critical path method has been proposed in grid system for scheduling scientific workflow applications [8] [9] without considering the resource utilization and cost.

DET algorithm proposed by Yuan et.al [10] uses dynamic programming approach and an iterative procedure to distribute deadline for critical tasks and non-critical tasks respectively. Then the execution cost is minimized using a local optimization procedure. The study made by Xiaotang Wen et.al [11] used ACO for resource scheduling and PSO algorithm to improve resource utilization.

Resource constrained project scheduling problem using ACO is proposed by Yumiao Zhou et.al [12] which lacks in making comprehensive use of several priority rules. Hybrid algorithm using ACO and Cuckoo search for job scheduling has been proposed by R.G Babukartik et.al [13] which minimizes the makespan. But the algorithm did not consider resource cost and resource utilization. PACO (Period ACO) based independent task scheduling algorithm in cloud computing is proposed by Weifeng Sun st.al [14] performed well in terms of makespan and load balance of whole cloud cluster. The proposed approach considered makespan as heuristics data. Workflow Scheduling using ACO algorithm proposed by vinothina et.al [16] mainly emphasized on makespan. The algorithm scheduled the tasks by considering Virtual Machine (VM) capacity and completion time of the tasks assigned to particular VM. But the proposed approach considers task length and number of child tasks to obtain better schedule.

III. Scheduling System

Workflow applications are represented by Directed Acyclic Graph (DAG) in which set of tasks (t_1,t_2,…t_n) are represented by set of vertices and precedence constraints are represented by set of directed edges (e1,e2,…en). Edge from e_i to e_j indicates that $task_i$ should complete execution before $task_j$ can start. Fig.1 depicts the basic structure of montage scientific workflow application which is used in astrophotography to assemble astronomical images as mosaics to save original input images.

Each task in the work flow is characterized by its input files for processing, output files to be send to the child tasks, runtime, level of task, child tasks and parent tasks. The tasks are processed by the set of VMs and the speed of each VM is represented in MIPS (million instructions per second). The cost of the workflow is defined by execution time and communication time. Execution time is the makespan of workflow i.e. the time required to schedule and executes the workflow in cloud. The communication time is the time required to transfer the input and output data between the tasks. Resource cost involves the use of compute resources (CPU, memory) and communication resources (bandwidth) per second in cloud.

IV. Improved Workflow Scheduling using ACO

ACO (Ant colony optimization) was proposed by M.Dorigo [17] which is based on the social food seeking behaviour of ants shown in Fig.2 having the objective of finding shortest path between two points. Since then it has been successfully applied to several NP-hard combinatorial optimization problems [18].The proposed approach uses ACO for mapping of tasks to resources so that all the tasks can complete its computation within minimal time.

A. Complexity in WFSACO Algorithm

WFSACO algorithm is designed for public cloud environment due to heterogeneity nature of computational resources in a cloud datacenter and heterogeneity natured tasks in the workflow. This algorithm clustered the tasks level wise and then scheduled to available VMs level wise. After clustering the tasks, the tasks are ordered in four different

ways: Topological order, Longest task First, Shortest task first and random manner. The number of ants equal to the number of ways ordering tasks.

Suppose the number of levels in a workflow is 50 then for each level the ordering procedures has to be applied. In addition to that each ordering takes its own time. The scheduling time complexity of the algorithm increases as the number of tasks in each level increases. Hence the following problems need to be addressed to improve the performance of WFSACO.

1. Scheduling/ordering at each level of workflow which may increases the makespan.
2. After mapping tasks in a level to VMs, the VM which would complete its tasks earlier must wait for other VMs as this algorithm follow level wise scheduling.
3. VM with more capacity is assigned with more tasks as per Transition Probability (TP) formula. This makes the lower capacity VM sits idle.

Hence to overcome the above mentioned problem, the ACO based algorithm WFSACO is redesigned as IWSACO. The work flow of IWSACO is given in Fig, 3. The steps are described as follows.

B. *IWSACO Algorithm*

1. Initially, the submitted workflow as directed acyclic graph is parsed and the tasks are arranged level wise in order based on the input task length. For example, in level 1, there are four tasks t_1, t_2, t_3 and t_4 with file size 700 MB, 400MB, 800MB and 300MB respectively. Then the order of task would be t_3, t_1, t_2 and t_4.

2. If two tasks have same task length, then number of child tasks will be considered to order the tasks. Again if there is any tie, on first come first served basis tasks are ordered. Likewise all the tasks are ordered and numbered sequentially.

3. The number of ants is constantly set to one because the ant maps the task with VM which will give minimal makespan using TP formula. The ant acts as a scheduling agent and the scheduling order is same for all the tasks in workflow. Hence the number of ant is one.

4. To minimize the makespan, the ant tries scheduling tasks with different VMs. Hence the number of iteration is equal to the number of VMs. The number of VMs is created from the available computing capacity (in mips) in a cloud host.

5. For the transition probability Formula, the

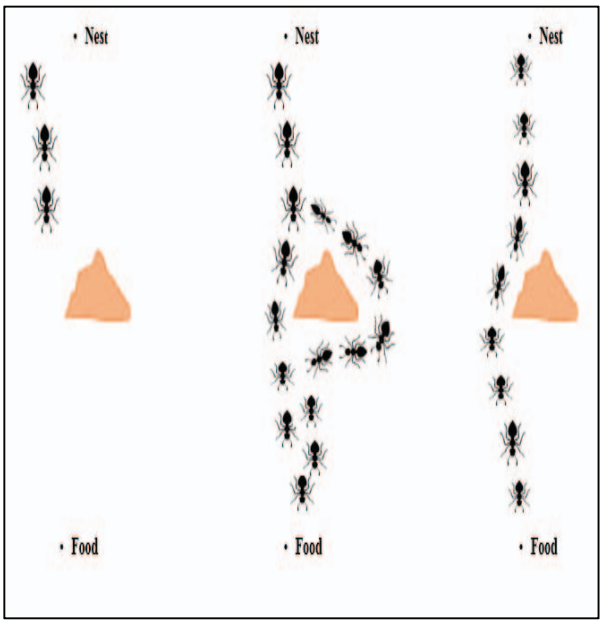
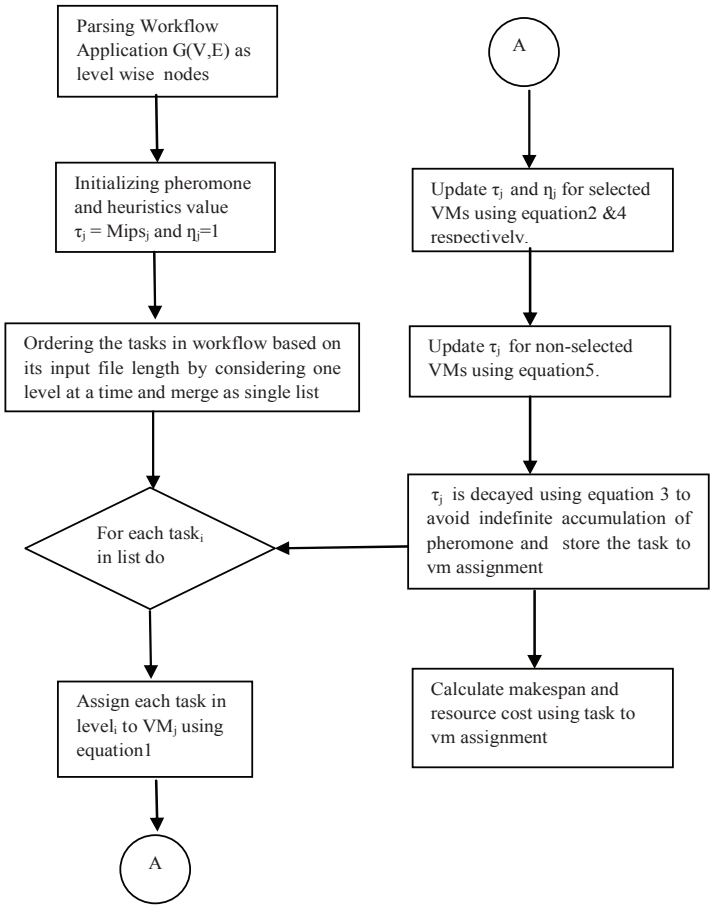


Fig. 2. Social Food Seeking Behavior of ant "unpublished"[19]



Fig, 3. Workflow of IWSACO

pheromone and heuristics value are chosen for each VM.

6. The improved approach lets the following assumption:
   Pheromone τ_j = computing capacity of VM_j (in mips)
   Heuristics (η_j) = inversely proportional to the completion time of last task assigned to that VM_j.
   Initially $\tau_j = Mips_j$ and $\eta_j=1$.

7. The transition probability (TP) of assigning a $task_i$ to a VM_j is calculated using equation 1 as in [16]

$$TP_{ij} = \frac{(\tau_j)^{\alpha} * (\eta_j)^{\beta}}{\sum_{j \in n} (\tau_j)^{\alpha} * (\eta_j)^{\beta}} \quad (1)$$

   where α represents the importance of computing capacity and β represents the importance of completion time of task in our algorithm. Initially, α=1 and β=1. n=available VMs in a host.

8. After the task chooses a VM, the pheromone and heuristics value is updated on selected VMs using the following equations 2 and 4 respectively as in [16] so that other task will not choose the selected VM till the completion of assigned task.

$$\tau_{j+1} = (1-\rho)\ \tau j + \Delta\tau_j \quad (2)$$

   To prevent infinite accumulation of pheromone, the pheromone trail decay coefficient ρЄ [0, 1] is used in ACO. Here the computing capacity is decayed and assigned to local pheromone updating factor $\Delta\tau_j$ and the value of $\Delta\tau_j$is given by equation 3.

$$\Delta\tau_j = 1-((CT_{ij}-vm_{avg})/vm_{sum}) \quad (3)$$

   Where CT_{ij} is the completion time (makespan) of last $task_i$ being assigned to vm_j, vm_{avg} is the average completion time of all vms and vm_{sum} is the sum of makespan of all vms. The heuristics value is updated as follows

$$\Pi_j = 1/CT_{ij} \quad (4)$$

9. The pheromone value for non selected vms are set using equation 5.

$$\tau_{j+1} = \tau_j * \Delta\tau_j \quad (5)$$

10. The steps from 7 to 9 are repeated for all the tasks in the scientific workflow.

11. As per schedule, the tasks are mapped to VMs. The resource cost can be calculated using makespan, cost of computational resource and bandwidth per second. The resource usage can be estimated by finding the idle time slot in each virtual machine.

Since all the tasks are scheduled one time much of the makespan time as per WFSACO algorithm will be reduced using IWSACO. All the VMs are in use throughout the computation of scientific workflow. The significant factor need to be evaluated is precedence constraint of tasks. But through empirical study that factor can be analyzed and corrected.

V. Experimental Results

To evaluate the algorithm, the scientific community's realistic workflows such as Montage (astronomy), Epigenomics (biology) and SIPHT (biology) are used which are available as XML files[20]. CloudSim [21] simulator is used to set up public cloud environment. The workflow is submitted in a single host of cloud datacenter with necessary input files. The algorithms IWFSACO and WFSACO are evaluated using 5 virtual machines. The most significant metrics Makespan is considered for evaluation. The results are shown in Table 1.

Workflow Application-Number of Tasks	Makespan In WFSACO (ms)	Makespan in IWSACO (ms)
Montage-25	58.5	56.8
Montage-50	122.42	105.3
Epigenomics-24	2455.3	2100.2
Epigenomics-47	3805.2	3102.7
SIPHT-30	168.4	162.5
SIPHT-60	249.7	199.2

Table.1 Algorithms Results

The results have shown that IWSACO algorithm given better makespan than WFSACO. Obviously, IWSACO takes less scheduling time than WFSACO as ordering and scheduling is done once for all tasks in workflow. Moreover VMs need not wait for other VMs to complete its task before moving to next level.

VI. Conclusion

The improved version of WFSACO presented in this paper for workflow scheduling using ACO minimizes the makespan as well as reduce the complexity of WFSACO. First, tasks in each level of the workflow are ordered based on task length and number of its child tasks. The ordered tasks are mapped to resources using ACO. To minimize the makespan, pheromone levels and heuristics data are updated for each machine, based

on transition probability as per the proposed approach assumption. The comparison of empirical results of the proposed approach with that of other existing algorithms is being worked out with other metrics such as scheduling time and resource cost. Based on the results, the redesigning of the proposed algorithm will be our future work.

REFERENCES

[1] Weifeng Sun, Ning Zhang, Haotian Wang, wenjuan Yin, Tie Qiu, PACO: A Period ACO_based Scheduling Algorithm in Cloud Computing, International conference on Cloud Computing and Big Data.482-486,2013.

[2] RaghavendraAchar,P.SanthiThilagam, Shwetha D, PoojaH,Roshni and Andrea, "Optimal scheduling of Computational Task in Cloud using Virtual Machine Tree", 2012 Third International Conference on Emerging Applications of Information Technology,2012,pp.143-146.

[3] Xiangyu Lin and Chase Qishi Wu, "On Scientific Workflow Scheduling in Cloud under Budget Constraint", 2013 42nd International Conference on Parallel Processing, pp.90-99

[4] C.Hoffa, G.Mehta, T.Freeman, E.Deelman, K.Keahey,B.Berriman and J.Good, "On the use of cloud computing for scientific workflows", in Proc. of the 4th IEEE Int.Conf.oneScience, Washington, DC USA, 2008, pp.640-645.

[5] SaeidAbrishami,MahmoudNaghibzadeh and Dick H.J Epema," Cost-Driven Scheduling of Grid Workflows Using Partial Critical Paths", IEEE Transactions on Parallel and Distributed Systems, Vol.23, No.8, Aug 2012.

[6] Tang R, Qin Y.Zhang L, "Research on Heuristics Logistics Distribution Algorithm Based on Parallel Multi-ant colonies", Journal of Software, pp-612-619, 2011.

[7] E KByun, Y S Kee, J S Kim and S Maeng, "Cost Optimized provisioning of elastic resources for application workflows", Future Generation Computer Systems, vol.27, no.8,pp.1011-1026, Oct.2011.

[8] Tianchi Ma and RajkumarBuyya , "Critical path and priority based algorithms for scheduling workflows with Parameter Sweep tasks on Global Grids", 17th International Symposium on Computer Architecture and High Performance Computing, IEEE Computer Society.

[9] M Rahman, S Venugopal and RajkumarBuyya, "A Dynamic Critical Path Algorithm for Scheduling Scientific workflow Applications on Global Grids", 3rd International conference on e-Science and Grid Computing, pp.35-42.

[10] Y.Yuan, X.Li,Q.Wang and X.Zhu, "Deadline Division-Based Heuristic for cost-Optimization in Workflow Scheduling", Information Sciences, vol.179,no.15, pp.2562-2575,2009.

[11] Xiaotang Wen, Minghe Huang and Jianhua Shi, "Study on Resource Scheduling Based on ACO algorithm and PSO algorithm in Cloud Computing", 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science, 2012, pp-219-222.

[12] Yumiao Zhou, QingshunGuo and RongweiGan, "Improved ACO Algorithm for Resource –Constrained project Scheduling Problem", International conference on Artificial Intelligence and Computational Intelligence, 2009 pp.358-365.

[13] R.G Banukartik and Dhavachelvan, "Hybrid Algorithm using the advantage of ACO and Cuckoo search for Job Scheduling", International Journal of Information Technology and Convergence and Services vol.2, Aug 2012, pp.25-34.

[14] Weifeng Sun, Ning Zhang, Haotian Wang, Wenjuan Yin and Tie Qiu, "PACO: A Period ACO_Based Scheduling Algorithm in Cloud Computing:, International Conference on Cloud computing and Big Data, pp,482-486, 2013.

[15] RaghavendraAchar,P.SanthiThilagam, Shwetha D, PoojaH,Roshni and Andrea, "Optimal scheduling of Computational Task in Cloud using Virtual Machine Tree", 2012 Third International Conference on Emerging Applications of Information Technology,2012,pp.143-146.

[16] Vinothina.V and Dr.R.Sridaran, "Scheduling Scientific Workflow Based Applications using ACO in Public Cloud", International Journal of Engineering and Technology, Vol.7, No.6, pp.1994-2000, 2016.

[17] M.Dorigo, C.Blum, "Ant Colony Optimization Theory: A survey", in Theorectical computer science 344(2-3) 2005,pp.243-278, 2005.

[18] Tang R, Qin Y.Zhang L, "Research on Heuristics Logistics Distribution Algorithm Based on Parallel Multi-ant colonies", Journal of Software, pp-612-619, 2011.

[19] Vinothina.V, "Design and Development of Effective Algorithms for Computing Scientific Workflow Based Applications in cloud Environment", Thesis, 2016.

[20] https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator.

[21] http://www.cloudbus.org./cloudsim