

Improvised Particle Swarm Optimization Technique for Workflow Balancing in Cloud

Babita Rani Radwal
Department of Computer Science and Engineering
Uttarakhand Technical University Dehradun, India

Sanjay Kumar, PhD
Department of Computer Science and Engineering
Uttarakhand Technical University Dehradun, India

ABSTRACT

Our research focuses on improvised Particle Swarm Optimization Technique for workflow balancing in cloud (IPSO-WF). The suggested technique assigns a cost to each task based on the resource requirement, the algorithm takes the four linear VMs (Virtual Machines) into deliberation before assign the job to it. The swarm searches for the VM meeting the rate, the work is assigned to the selected VM and resources updated. Since resources are allotted and VM engaged with a work the rate must needed to updated which has been disregarded in most of the research work however the planned algorithm updates the resources and rate is calculated again and again every instance when the work is assigned giving it a more realistic costing.

The suggested work has been tested on 25, 50 and 75 VMs for L-ACO, B and BM. All the procedures perform best when more VMs are allotted as lesser VMs takes more resources resulting about loss of energy and time too. The acquired results shows that through all the systems are competitive but the suggested technique performs much better in the terms of time and energy.

Keywords

Load balancing; Workflow; PSO; ACO

1. INTRODUCTION

Cloud Computing, a modern approach in today's world. It provides on-demand facilities like software, infrastructure, platform, hardware, and storage etc. dynamically to the end user as stated in to the "pay per use" model by using virtualized resources in excess of internet. It can host discrete applications such as business, social networks and technical applications.

While CC also provides few services like IaaS, PaaS and SaaS etc. to end users but due to freshness of CC, it also undergoes from some types of research issues like security, performance, database management, workflow scheduling, server consolidation, VM migration, and fault tolerance etc. Workflow scheduling is major concern for scientific applications among all of these issues. Present workflow scheduling techniques in the cloud and grid environment focused on several QoS parameters like cost, CPU time, reliability and makespan etc.

The inception of smart IOT devices has seen a standard shift of the necessity of the cloud and with more and more applications using cloud the requirement of WF scheduling has increased many folds. ACO with its discrete variants has been utilized over the previous year's not only for networking but scheduling and the load balancing as well. However, new robust and efficient technique is required for balancing the load of the cloud.

2. MOTIVATION

The research work focuses on IPSO-WF (improved Particle Swarm Optimization Technique for workflow) in the cloud. In the planned work assigns a charge to every task based on the resource requirement, the system takes the four linear Virtual Machines (VMs) into consideration before assign the task to it. The swarm searches for the VM meeting the charge, our suggested work is allotted to the selected VM and resources updated. Since the resources are allotted and VM engaged with a work the charge need to be updated which has been ignored in most of the research work however the suggested innovation updates the resources and cost is recalculated every time the work is assigned giving it a more realistic costing.

3. RELATED WORK

MengXu et al (2009) [3] focused on makespan and cost, time, success ratio parameters in his work. This approach calculates the mean run time and means execution cost of all the workflows and schedules the task first having a slightest covariance of time and payment to optimize both the makespan and cost. The procedure scheduled WF at dynamic interval and decrease the performance time and cost. Conduct further evaluations with a realistic tested and study how to add more QoS constrained (reliability, availability, etc) to workflows were the main concern with the approach.

S.Pandey et al (2010) [4] wished-for a PSO based heuristic technique which considers the computation and transmission cost as the scheduling parameters. The approach calculates the average computational and average transmission cost of each task and schedules tasks to the resource with minimized cost.

K. Liu et al, Y. Yang et al, (2010) [5] Execution value and time and planned any other booking calculation for Innovative change exhaustive cost-requirement. It was utilized to restrict the fee under certain patron assigned due dates. Empowers the bargains of execution fee and time. The principal concern with the system was to apply our reserving calculation to some certifiable applications, for example, protection guaranteeing work procedure frameworks.

S.Pandey et al (2010) [4], works for Resource utilization, time parameters and projected a heuristic scheduling strategy primarily based on PSO. Which was once used for three times price savings as compared to BRS and excellent distribution of workload onto resources. Main problem with the strategy was to integrate PSO based heuristic into workflow management gadget to schedule workflows of actual purposes such as intelligence imaging analysis, EMO, and others.

Zhangjun Wu et al, Zhiwei Ni et al, Dong Yuan et al and (2011)[10] works on completion time and scalability and planned Market-oriented hierarchical scheduling strategy. In which the Overall jogging fee of cloud workflow system will be reduced. It can be used to optimize each makespan and cost

concurrently. The fundamental trouble with strategy was once ultimate solution for the global task-to-task mission may now not be found.

J. Durillo et al (2012) [8] focused on makespan and cost. The task having highest rank is scheduled first. MOHEFT [26] extends HEFT in the cloud environment by using the SPEA2* approach for bi-objective scheduling criteria to optimize both makespan and the economic cost in the cloud environment.

Ghanbari et al (2012) [7] planned a scheduling policy for priority based job through considering makespan as the scheduling parameter in the cloud environment by considering three levels of Analytical Hierarchy Process which are represented with three types of job priority which are objective level, scheduling level and job level.

Zhangjun Wu et al (2013) [10] includes GA, ACO and PSO heuristics for the job level and resource level scheduling known as the project of task-to-service scheduling by bearing in mind makespan, rate and resource utilization as the scheduling parameters. This method basically deals on task to its service assignment. In glimpse of this all task of distinct WF instances mapped to cloud services in cloud market globally. These were based on functionally and non functional QoS requirements. Results for parameters compared in ACO,GA, PSO approaches and find out that ACO is performing better than others.

Yuli Yang et al (2013) [9] schedules the task by authenticating resource disappointment possibility for the duration of the diffusion of the task with security and reliability constraint. The main aim on the adjustment b/w the execution price and time taken by the task. The result displays that the approach is much better than the traditional technique for WF and also observe most trusted efficient in execution flow.

Behzad, Shahram et al, Reza Fotohi et al (2013) [6] focus their investigation work on priority and time quantum parameters. By using these parameters they worked to reduce ordinary waiting time and response time. The approach was suggested to estimate the minimum no of computing hosts these host are required for executing the WF within a customer specific computation time. BTS stores worthy performance with WF which have MPI i.e parallel task.

Agarwal et al, and Saloni Jain (2014) [11] worked to reduce run time by working on cloud size and priority parameters. By choosing added task & run time can be reduced. Planned policy tested in grid environment and the time difference can be observed by testing it in a cloud environment was the major contention with this approach.

Longhua Feng et al (2014)[12] chipped away at different parameters, for example, time execution, data transfer capacity inclination, Expectation time inclination, JEF capacity and capacity result to enhance QoS and diminish makespan. Change should be possible in the arranged calculation to diminish culmination time and to acquire decency was the principle worry with the procedure.

Shahab et al, and Zarintaj et al (2015) [13] worked on heuristics PSO based approach for scheduling. In their research they worked on number of particles in Swarm, number of iterations, learning rate with regard to individual ability and learning rate with regards to Social ability parameters to reduce makespan and to improve memory utilization. Approach implementation with hybrid heuristic was the major concern.

N. Moganarangan et al, Babukarthik et al, S Bhuvaneswari et al, S. Basha, S., and Dhavachelvan et al (2016) [14] planned a combined technique of ACO and cuckoo search policy. They worked on energy consumption and makespan parameters. In their work they focused on voltage scaling factor for minimizing energy consumption, Makespan of their planned algo based on number of task compared with ACO.

N.K. Chien et al, and N.H. Son et al, (2016)[15] focused on a approach for LB and worked on processing time and response time parameters. Their approach was based on method of calculating the EOS time in heterogeneous cloud environment. The results of their work improves response time and processing time.

Zhicheng Cai et al, Xiaoping Li et al, Ruben Ruiz et al and Qianmu Li et al (2017)[16] planned a scheduling algorithm (DDS) for bag-of-tasks workflow with stochastic task completion time. The policy works to fulfil the workflow deadline with the help of some of task completion time expectation. Which is used to estimate real task completion time.

Qingsheng Zhu et al (2014)[17], advised a approach for reducing the cost and success ratio. Which was a combined technique and these approaches were L-ACO and Provis. By combining both of the approaches the optimization of cost for workflow scheduling with deadline constraints is done. The main objectives of the strategy was success ratio and cost.

4. RESEARCH METHODOLOGY

PSO and ACO for WF preparation is adapted for improved completing time, LB, and cost minimization, it is advised intended for improved LB performance. The major purpose of this study work is to schedule dissimilar workflows in an well-organized and the wide-ranging carrying out time, and in general assessment would be reduce the intended scheme would be power well-organized also. In outlook of this work, it is advocate a novel approach bottom on PSO that are an get better PSO for workflow assessment in cloud. The purpose of it heuristic scheduling is used to reduce the already exist resource that is not make use of appropriately at whatever time there are no one tasks accessible for scheduling. It is additional robust and efficient method for evaluation the load of the cloud. The intended procedure allocates which have charge to every task based on the reserve condition, however, algorithm get the four linear VMs into contemplation previous to allocate the task to it. Swarm investigate has been for the VM meeting the cost, the work is allotted to the certain VM and resources updated. Since the resources are selected and VM occupied with a occupation the cost necessary to be updated which has to be uncared for in mainly of the study work therefore the considered procedure updates the resources and cost is recalculated every occasion the work is allocate generous it a more sensible assessment.

4.1 Algorithm IPSO-WF (Improved PSO for Workflow)

Step 1: Start

Step 2: loop i from 1 to Vms/4

Step 3: Calculate required bandwidth(rBw)
Calculate energy (renergy)

Step 4: if i is less than equal to n-4 then

Step 5: $vm \leftarrow \min(\{vm(i).CP, vm(i+1).CP, vm(i+2).CP, vm(i+3).CP\})$

Step 6: elseif i is equal to 68 then

Step 7: $vm \leftarrow \min([vm(i).CP, vm(i-(Vms-4)).CP, vm(i-(Vms-3)).CP, vm((Vms-1)-i).CP])$

Step 8: elseif i is equal to 70 then

Step 9: $vm \leftarrow \min([vm(i).CP, vm(i-(Vms-1)).CP, vm(i-(Vms-2)).CP, vm((Vms-1)-3).CP])$

Step 10: end if

Step 11: $storedCostPerMem \leftarrow Vms(vm). \text{ costPerMem}$
 $storedcostPerBw \leftarrow Vms(vm). \text{ costPerBw}$

Step 12: calculate RcostPerMem

Step 13: if required cost per memory is less than the storedCostPerMem

Step 14: if required cost per bw is less than the storedCostPerBw

Step 15: update the resources

Step 16: allot the virtual machine

Step 17: end if

Step 18: else

Step 19: loop back to get optimal resources

Step 20: end

Step 21: end loop

Step 22: Stop

here:

required cost per memory = $(Ram/Vms) / \text{renergy}$

required cost per bandwidth = $(\text{costPerBw}/Vms)/rBw$

4.2 WORKING

The virtual machines are allotted resources with the cost of operation.

As the network starts and VMs are activated the required bandwidth of the request and required energy of the request is calculated. Four continuous Virtual machines are taken into consideration, Virtual machine with Minimum cost is selected and resources updated. If loop comes to the last VM then first four virtual machines are selected, if it comes to second last then last and first three are selected and if third last then last three and the first virtual machine is selected.

5. RESULTS AND ANALYSIS

5.1 Parameters used to analyse proposed Algorithm

Energy: This parameter shows that impact on the battery consumption of the sensor, including the energy consumed due to sensing communication and manipulative tasks.

Time: In this parameter we can show the extreme capacity of time which is occupied by the job may use to processor or to find out how much overall execution interval occupied by a job for complete its task.

Cost-effectiveness: Cloud computing is cost-effective ability of the information allocation on the cloud. But, there might also be a hidden value associated with the process which includes- facts recovery, application modification, software insurance etc. For instance, this is no longer fee effective to

use a couple of SaaS to resolve one task. The interoperability in cloud computing may additionally include the chance of an enlarge in value in the transition of the solution from one cloud to another.

5.2 Mathematical formulations

Users can charge on the basis of number of interval intervals in which they have used the resources and any partial utilization is rounded to a full-interval interval for example 0.4 hour is rounded to 1 hour

Calculating Execution Time: When task t_i is allotted to service S_i , the completion instance can be calculated as:

$$ET_tL = W_i/P(S_i)$$

Calculating Cost: if S_i is leased by the user from LST_i (Lease start time) to LFT_i (Lease finish time) the required cost is calculated by:

$$E C_i = [(LFT_i - LST_i) / T_i] \times C(S_i)$$

Where S_i is service, T_i is time interval and $C(S_i)$ is cost and $P(S_i)$ is processing capacity.

Calculating Response Time: The response time, RT is calculated with the help of the following formula-

$$RT = F_t - Ar_t + TD$$

where Ar_t is the arrival time of enduser request and F_t is the finish time of enduser request.

Calculating Transmission Delay: The transmission delay, TD is calculated with the help of using the following formulas

$$TD = TI + Tt$$

where TD is the transmission delay TI is the network latency.

Tt is the time which is used to transfer the size of data of a single request from source to destination location. TD is considered to be same in every case and hence it is considered as zero.

5.3 Results for proposed algorithms

For better LB we compare our determined scheme with three different algorithms L-ACO, PSO and DDS (Delay based dynamic) on different VMs (25,50, and 75).

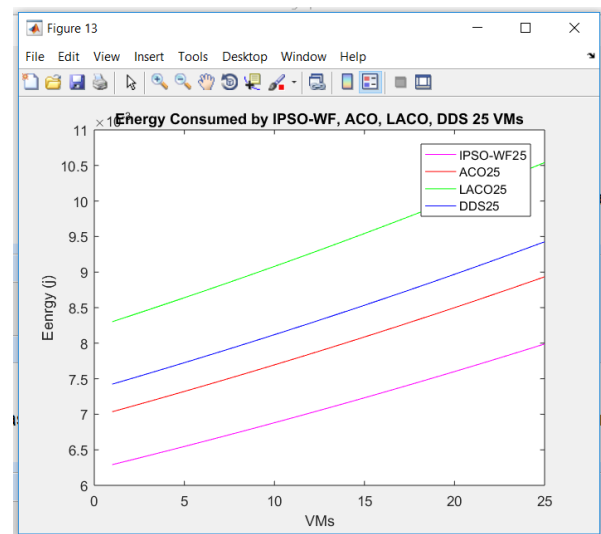


Fig. 1. Energy consumed by Proposed IPSO-WF in compared to existing algorithms on 25 VM

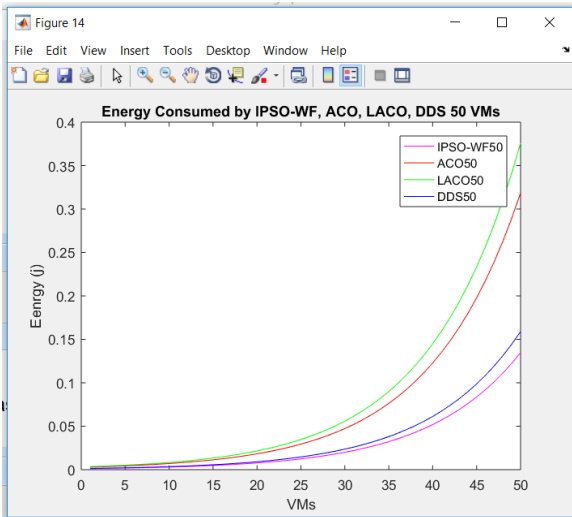


Fig. 2. Energy consumed by Proposed IPSO-WF in compared to existing algorithms on 50 VM

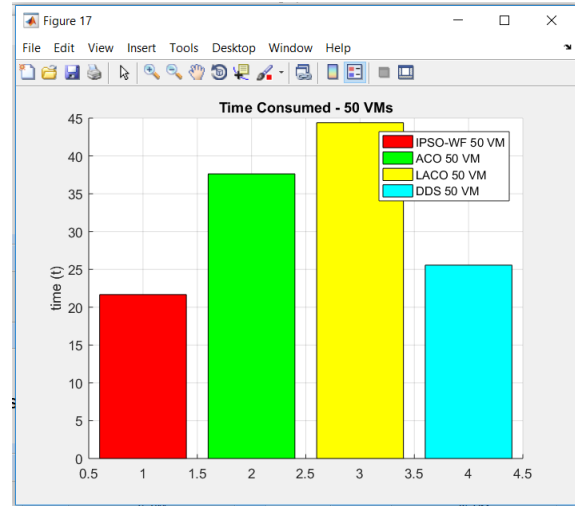


Fig. 5. Time consumed by Proposed IPSO-WF in compared to existing algorithm on 50 VM

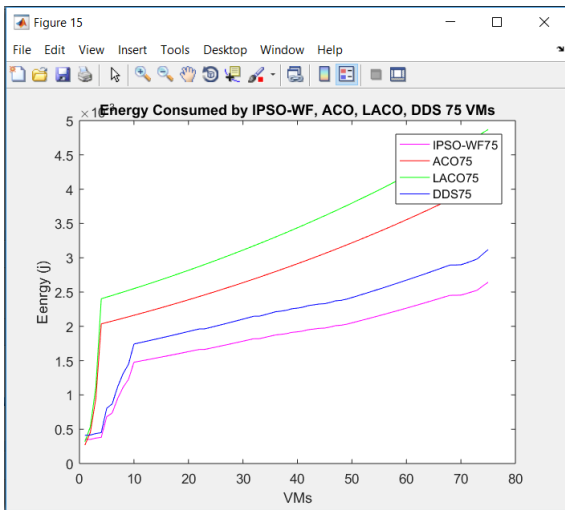


Fig. 3. Energy consumed by Proposed IPSO-WF in compared to existing algorithms on 75 VM

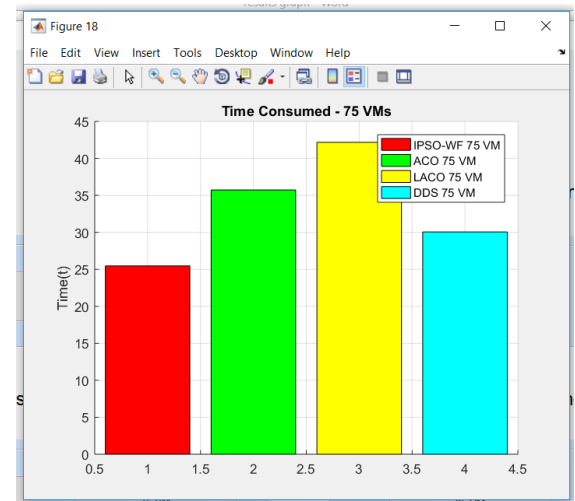


Fig. 6. Time consumed by Proposed IPSO-WF in compared to existing algorithm on 75 VM

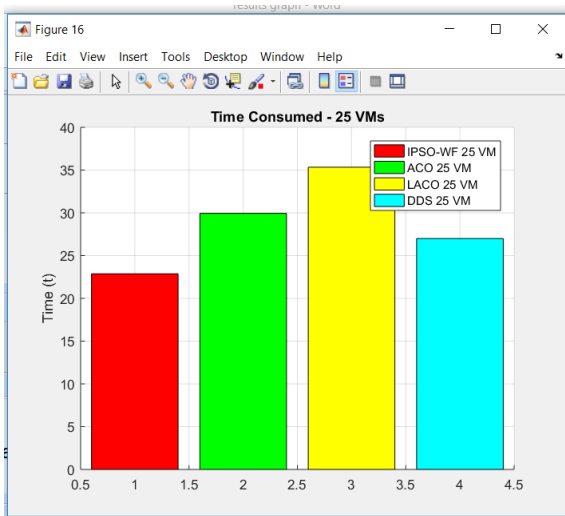


Fig. 4. Time consumed by Proposed IPSO-WF in compared to existing algorithm on 25 VM

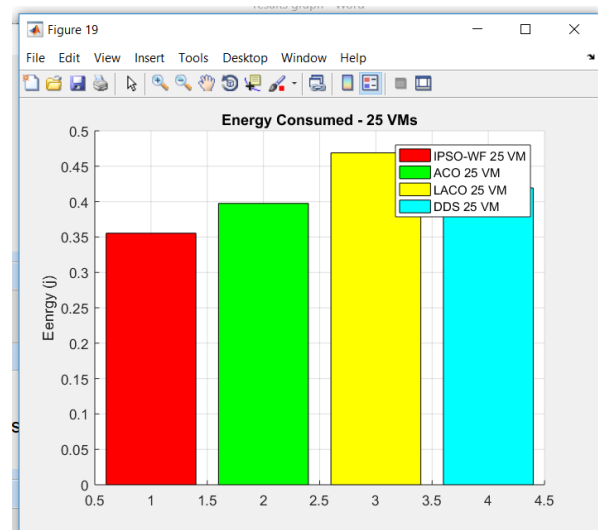


Fig. 7. Energy consumed by Proposed IPSO-WF in compared to existing algorithm on 25 VM

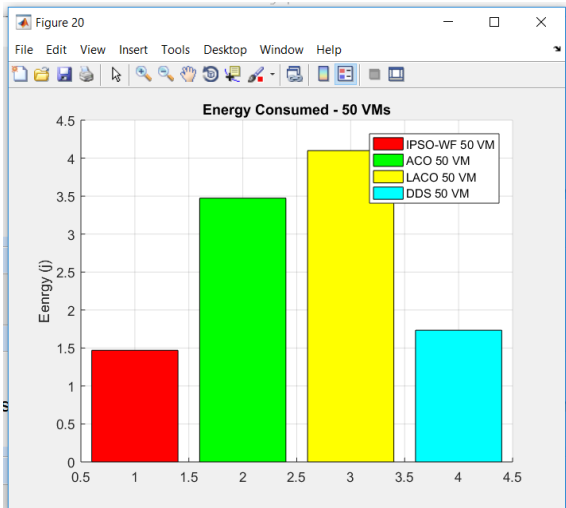


Fig. 8. Energy consumed by Proposed IPSO-WF in compared to existing algorithm on 50 VM

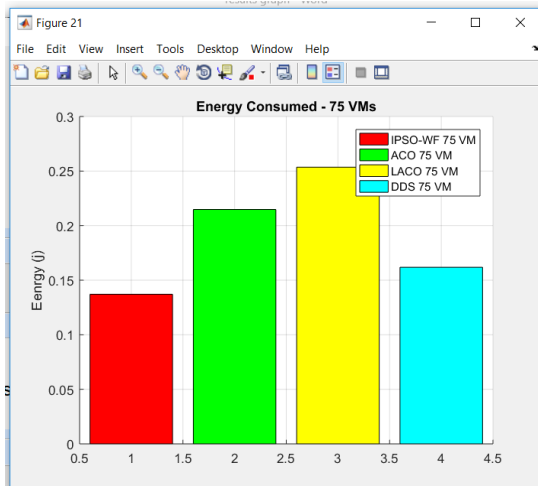


Fig. 9. Energy consumed by Proposed IPSO-WF in compared to existing algorithm on 75 VM

Comparison of proposed IPSO-WF is done with existing algorithms ACO, DDS, and LACO. All the compared results shows that our recommended IPSO-WF consumed less energy and time on all VM (25, 50, 75) is minimum, so the recommended schemes give the better results in compared to previously used schemes.

6. SECTIONS

Present scheduling strategies concentrated on the time success ratio and the cost. The main aim of these schedulers are reduction of the overall finishing time and price of the workflow but no consideration was given on the resource utilization. In aspect of this research work we focus on improvised PSO for workflow balancing in the cloud. The result analysis of this work shows that respons instance of the structure is reduced as matched with the other systems.

In this work we assigns a cost to every task on the bases of resource requirement, the scheme takes the four linear VMs into consideration before assign the task to it. The suggested technique updates the resources and cost can also be is recalculated each interval the work is assigned giving it a more realistic costing. The results obtained shows that through all the strategies are competitive but the planned system implements improved and much better results in the

terms of time and energy. In this proposition, we displayed a modified version of PSO algorithms. IPSO-WF algorithm also compared with DDS, ACO and L-ACO, algorithms. IPSO-WF approach outperformed the DDS, ACO and L-ACO algorithms in terms of time and energy consumption. MATLAB tool is used to implementation purpose

7. REFERENCES

- [1] M. Wiecek, R. Prodan, T. Fahringer, "Scheduling of scientific workflows in the ASKALON grid environment", SIGMOD Record, vol. 34, no. 3, pp. 56–62, 2005.
- [2] G. Singh, C. Kesselman, E. Deelman, "Optimizing grid-based workflow execution", Journal of Grid Computing, vol. 3, no. 3-4, pp. 201-219, 2005.
- [3] M. Xu, L. Cui, H. Wang, Y. Bi, "A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing", IEEE, 2009.
- [4] S. Pandey, L. Wu, S. Guru, R. Buyya "A particle swarm optimization based heuristic for scheduling workflow applications in cloud computing environments", pp. 400–407, IEEE, 2010.
- [5] K. Liu, Y. Yang, J. Chen, X. Liu, D. Yuan, H. Jin "A Compromised-Time- Cost Scheduling Algorithm in Swin DeW-C for Instance-intensive Cost-Constrained Workflows on Cloud Computing Platform", International Journal of High-Performance Computing Applications, vol.24 no.4 445-456, May 2008.
- [6] E.-K. Byun, Y.-S. Kee, J.-S. Kim, E. Deelman, S. Maeng "Bts: Resource capacity estimate for time-targeted science workflows", Journal of Parallel and Distributed Computing, vol. 71, no. 6, pp. 848–862, 2011
- [7] S. Ghanbari, M. Othman, "A priority-based job scheduling algorithm in cloud computing", International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE), pp.778-785, 2012.
- [8] J. Durillo, H. Fard, R. Prodan "MOHEFT: A multi-objective list-based method for workflow scheduling", pp. 185-192. IEEE, 2012.
- [9] Y. Yang, X. Peng "Trust-Based Scheduling Strategy for Workflow Applications in Cloud Environment", pp. 316-320. IEEE, 2013.
- [10] Z. Wu, X. Liu, Z. Ni, D. Yuan, Y. Yang "A Market-Oriented Hierarchical Scheduling Strategy in Cloud Workflow Systems", no. 1, pp. 256-293, Springer US 2013.
- [11] Agarwal, Dr, Saloni Jain "Efficient optimal algorithm of task scheduling in cloud computing environment", arXiv preprint arXiv: 1404.2076 (2014).
- [12] Li, Ji, Longhua Feng, Shenglong Fang "A greedy-based job scheduling algorithm in cloud computing", Journal of Software 9.4 (2014): 921-925.
- [13] TAREGHIAN, Shahab, Zarintaj BORNAEE "A new approach for scheduling jobs in cloud computing environment", Cumhuriyet Science Journal 36.3 (2015): 2499-2506.
- [14] Moganaragan, N., Babukarthik, R. G., Bhuvaneshwari, S., Basha, M. S., Dhavachelvan "A novel algorithm for reducing energy consumption in cloud computing

environment: Web service computing approach”, Journal of King Saud University-Computer and Information Sciences, 28(1), pp. 55-67.

- [15] Chien, N. K., Son, N. H., Loc, H. D. “Load balancing algorithm based on estimating finish time of services in cloud computing”, pp. 228–233, IEEE.
- [16] ZhichengCai, Xiaoping Li, Rubén Ruiz, Qianmu Li “A delay-based dynamic scheduling algorithm for bag-of-task workflows with stochastic task execution times in

clouds”. Future Generation Computer Systems vol71,pp. 57–72(2017)

- [17] Quanwang Wu, Fuyuki Ishikawa “Deadline-constrained Cost Optimization Approaches for Workflow Scheduling in Clouds”, IEEE, 2017
- [18] Babita Rani Radwal, Sanjay Kumar “Dynamic scheduling with task completion time estimation methods in cloud”, IJCSE, Vol.6, Issue.3, March 2018