EAI Endorsed Transactions
on Scalable information Systems
Research Article EAI.EU

SLA based Workflow Scheduling algorithm in Cloud Computing using Haris Hawks optimization

Sudheer Mangalampalli[1, *], Ganesh Reddy Karri[1], Kiran Sree Pokkuluri[2], K Varada RajKumar[3], Ganti Naga Satish[4]

[1]School of Computer Science and Engineering, VIT-AP University, Amaravati, India
[2] Department of Computer Science and Engineering, Shri Vishnu Engineering College for Women, Bhimavaram, A.P., India
[3]Department of Computer Science and Engineering (AIML), MLRIT Hyderabad, Telangana, India
[4]Department of Computer Science and Engineering, BVRIT HYDERABAD, Telangana, India

Abstract

Task Scheduling is crucial facet in cloud paradigm as virtual resources need to be provisioned to the variable requests coming onto cloud console from various users and more over that tasks are depends on each other which creates a workflow which is a difficult task for cloud service provider to provision these tasks over appropriate VMs. Inefficient mapping of tasks to VMs increases makespan and lead to violation of SLA between users, cloud provider. In this paper, we modeled a SLA based workflow scheduling algorithm focuses on minimization of makespan and SLA violations. This algorithm developed using Harris hawks optimization. Experimentation carried out using workflowsim. Random workload fed as input to algorithm and it is evaluated against existing baseline approaches and simulation results revealed that our proposed approach minimizes makespan and SLA violations over existing approaches by 40% and 43% respectively.

Keywords: makespan, SLA Violation, Haris Hawks optimization, PSO, ACO, GA

Received on 11 July 2023, accepted on 04 September 2023, published on 28 September 2023

doi: 10.4108/eetsis.4005

[1]Corresponding author. Email: author@emailaddress.com

1. Introduction

Cloud Computing paradigm changed the perspective of IT industry in terms of deployment of IT infrastructure virtually and providing all resources virtually as per the SLA laid in between customers and cloud providers. All virtual resources in cloud paradigm provided to customers in a customized manner based on their subscriptions and as per the pricing model they had chosen for their purposes [1]. To provide high quality services for customers and for provisioning resources to customers on demand cloud providers need efficient scheduling algorithm which follows SLA. Many existing scheduling algorithms are developed using various metaheuristic models in [2], [3], [4], [5] but still there is a chance of improvement for task scheduling as it is NP-hard problem. To tackle scheduling process in cloud paradigm efficiently, this res search paper proposes SLA based workflow scheduling approach by considering priorities of tasks. This workflow scheduling algorithm developed using Haris Hawks optimization which is a nature inspired algorithm based on cooperative behavior nature of Harris hawks [6].

The main contributions of manuscript are presented below.

1. Developed workflow scheduling approach which considers priorities of tasks, carefully maps workflows onto appropriate VMs.
2. Modeled workflow scheduling algorithm using Haris Hawks optimization.
3. Simulations conducted on workflowsim and compared against PSO, GA, ACO algorithms and addressed parameters i.e., SLA violations, makespan.

2. Related Works

In [7], a workflow scheduling approach is formulated for minimization of makespan and cost in cloud paradigm. A variant of PSO is used to model scheduling approach. Simulations conducted on workflowsim, compared over baseline mechanisms FCFS, HEFT. simulation results shown PPTS-PSO shows impact on above mentioned parameters. In [8], authors focused on improvement of resource efficiency and satisfaction of users using a hybridized approach. Authors hybridized PSO and GA to get balance over global and local search. It simulated over Cloudsim and finally evaluated against baseline algorithms. From results, PGA proved that it overcomes limitations of existing approaches for above parameters. In [9], a scheduling algorithm is formulated based on hybridization of GA and gravitational search algorithms. Simulation carried out on Cloudsim, evaluated against baseline algorithms. From results, GAGELS proved it outperforms existing approaches by addressing execution time, resource utilization. In [10], task scheduling mechanism is formulated for improvement of makespan, resource utilization by employing a machine learning approach which optimizes parameters. Simulation carried out in MATLAB and evaluated over existing approaches. From results it evident that machine learning mechanism added with metaheuristic approaches improved above mentioned parameters. In [11], a multi adaptive learning strategy formulated using PSO for tackling scheduling in cloud paradigm. This strategy works based on variety of two particles i.e. ordinary, best particles in which best particles population is to be raised and ordinary particles population is to be decreased to achieve optimization. Experimentation carried out on MATLAB and evaluated against SOTA approaches. Results proved MALPSO achieved improvement of execution time. In [12], authors devised task scheduling algorithm focused on minimization of consumption of energy, start time efficiency, makespan. This approach modeled using AINN-PSO to model effective resource provisioning. It evaluated against SOTA approaches. Results proved that AINN-PSO is dominant in minimizing energy consumption, start time, makespan. In [13], authors used whale optimization to model scheduling algorithm addresses makespan, migration time, energy consumption. For this approach, authors calculated task priorities carefully to schedule workflows onto precise virtual resources. Simulation carried out on workflowsim and evaluated over standard nature inspired algorithms. Results proved that existing Pwhale generates scheduling models well enough rather than existing approaches. In [14], authors presented a workflow scheduling algorithm focused to minimize execution time, energy and total cost. This approach modelled by using artificial algae approach in which scheduling takes place based on two phases i.e., high prioritized tasks are created using bottlenecks and then schedule tasks using weighted sum fitness function. Simulation carried out in workflowsim and compared against baseline approaches. Results revealed that MAA dominates other approaches for above mentioned parameters.

Table 1. Summary of Task and Workflow Scheduling algorithms in cloud paradigm

References	Algorithm	Parameters
[7]	PPTS-PSO	Makespan, cost
[8]	PGA	User Satisfaction, Resource efficiency
[9]	GAGELS	Execution time, Average Resource utilization
[10]	PSO-DRL, Firefly-DRL	Makespan, RU
[11]	MALPSO	Execution time
[12]	AINN-PSO	Makespan, energy, start time
[13]	Pwhale	Makespan, energy consumption, migration time.
[14]	MAA	Execution time, Energy Consumption, cost

From table1, we can clearly observe that many existing authors used nature inspired, metaheuristic approaches for tackling scheduling problem but still authors haven't focused on SLA violation. It is important criteria from facets of both cloud user, provider. Therefore, in this research we focused on precise scheduling of workflows onto appropriate virtual resources using Haris Hawks optimization to address metrics i.e., SLA violation, makespan.

3. Problem Definition, Proposed System Architecture

In this section, we defined workflow scheduling problem by assuming a workflow i.e., DAG as $g = (v, e)$ where v indicates set of tasks are represented here $t^k = \{t^1, t^2 \dots \dots t^k\}$. e indicates dependency which is interdependency in workflows. workflows are running on v^n virtual resources, they are indicated as $v^n = \{v^1, v^2, \dots .., v^n\}$. virtual resources reside in physical machines as $p^i = \{p^1, p^2, \dots . p^i\}$. Physical machines sitting in datacenters as $dc^j = \{dc^1, dc^2, \dots . dc^j\}$. From above statement, problem is defined as t^k interdependent tasks to be scheduled on v^n virtual machines to address SLA violation, makespan.

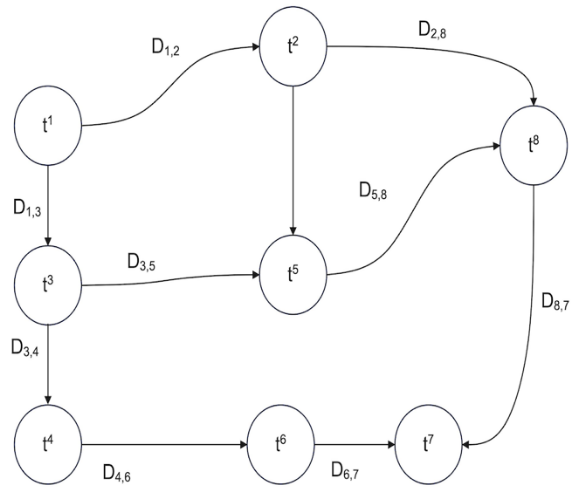

Figure 1. Workflow for simulation

Figure 1 indicates workflow used in our simulation and t^k indicates number of tasks, D^{mn} indicates connection between different tasks as it is a workflow.

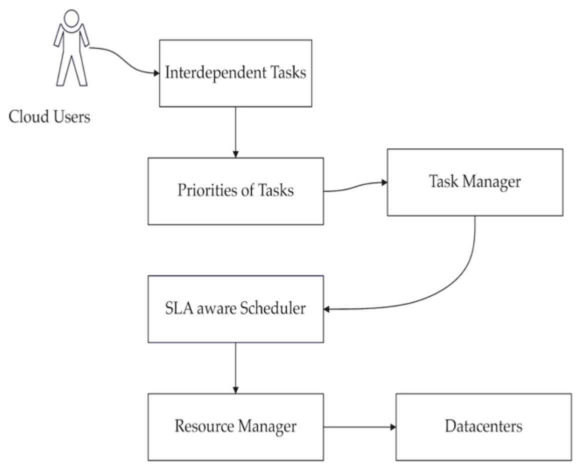

Fig. 2. Proposed System architecture for SLA aware Scheduler

Initially in our proposed system architecture various cloud users submitted requests onto cloud console and in turn those interdependent tasks are submitted to task manager and before submission to task manager task priorities evaluated for tasks depends on size, VM processing capacities. From the captured priorities tasks are appropriately scheduled to precise virtual resources by considering underlying resources available in resource manager.

3.1 Mathematical Modelling

This subsection clearly discusses about mathematical modelling of SLA aware scheduler. The main objective of this SLA aware scheduler is to map workflows accurately based on task priorities to suitable virtual machines while minimizing makespan, percentage of SLA violations. To calculate all incoming priorities of interdependent tasks below eqn.1. need to be used.

$$t^{pr} = t_k^{size} / v_n^{proc} \quad (1)$$

Where t_k^{size} indicates size of the task, v_n^{proc} indicates processing capacity of virtual resource. Earlier in this subsection, we have already mentioned that main objective of this scheduler is to improve makespan and it is calculated using below eqn.2.

$$m^k = av^n + e^k \quad (2)$$

Where av^n indicates availability of virtual resources, e^k indicates execution time for considered tasks. The second objective need to be addressed using SLA aware scheduler is to minimize percentage of SLA violations and it is to be calculated using below eqn.3, 4 and 5.

$$TA = \frac{1}{s}\sum_{i=1}^{s} \quad VioT^p / Act^p \quad (3)$$

$$PerDeg = \frac{1}{z}\sum_{x=1}^{z} \quad PerDeg^r / TotCap^r \quad (4)$$

SLA violations always depends on total active time of hosts, performance degradation of corresponding resource. It can be evaluated using eqn.5.

$$SLA^{violations} = TA * PerDeg \quad (5)$$

After calculation of makespan, SLA violations from eqns.2. and 5 we have formulated a fitness function which minimizes makespan, SLA Violation using SLA aware scheduler. It is calculated using below eqn.6.

$$f(x) = \sum \quad x\,(m^k(x)\,), min(SLA^{violations}(x)) \quad (6)$$

3.2 Haris Hawks Optimization

Haris Hawks optimization is a meta heuristic approach, and it is inspired from Haris hawks and proposed in [15]. In this approach, these Haris Hawks catches rabbits using different phases i.e., exploration, exploitation and transition phase from exploration to exploitation. Initially, Haris Hawks tries to catch prey based on whether they are in the close group or far away from them. Therefore, position vector is calculated using following eqn.7.

$$y(t+1) = \{y(t) - R_1|y_{rand}(t) - 2R_2y(t)| \quad Q \geq 0.5 \ (y_{rab}(t) - y_m(t)) - R_3(lb + R_4(ub - lb)) \quad Q < 0.5 \tag{7}$$

After identifying position vector of Hawk, average position of a Hawk bird can be calculated using below eqn. 8.

$$y_m(t) = \frac{1}{N}\sum_{i=1}^{N} \quad y_i(t) \tag{8}$$

Here in this algorithm, prey energy needs to be calculated as follows using eqn.9.

$$en = 2en_0(1 - \frac{t}{T}) \tag{9}$$

Where en is energy required for a prey to escape from hawk, T is maximum number of iterations, en_0 is initial energy of prey and this energy level lies in between [-1,1]. When en_0 decreases i.e., from 0 to -1 it weakens and if en_0 value is increasing from 0 to 1 it strengthens its energy and it tries to escape from Hawk. Transition from exploration to exploitation depends on two factors i.e., energy for prey to escape, probability of chance to escape from hawk. Probability of chance to escape from a hawk indicated as R and if $R < 0.5$ there is a chance of successful escape and if $R \geq 0.5$ then there is no possibility to escape from hawk. Now from these factors leads to exploitation phase and that exploitation is done in different ways i.e., soft besiege, hard besiege, soft besiege with incremental dives, hard besiege with incremental dives [15]. If $R \geq 0.5$ and $|en| \geq 0.5$ then soft besiege occurs and hawks encircle prey to catch it and exhaust their energy. It is calculated using following below eqns. 10,11.

$$y(t+1) = \ \delta y(t) - en|My_{rab}(t) - y(t)| \tag{10}$$

$$\delta y(t) = y_{rab}(t) - y(t) \tag{11}$$

If $R \geq 0.5$ & $|en| < 0.5$ then hard besiege starts and hawks encircle prey hardly and exhaust their energy. It is calculates using below eqn.12.

$$y(t+1) = \ y_{rab}(t) - en|\delta y(t)| \tag{12}$$

There are another two exploitation phases i.e., soft besiege with constructive dives and hard besiege with constructive dives. These two phases work based on previous movement of hawk. For soft besiege with constructive dives below eqn.13 to be used.

$$x = y_{rab}(t) - en|My_{rab}(t) - y(t)| \tag{13}$$

In this phase, movement of hawk is compared with previous movement and if it is not properly laid its movement then hawk will encircle prey abruptly. It is modeled by using following eqn.14.

$$z = x + a * lf(dim) \tag{14}$$

Where, $lf(x) = 0.01 * \frac{\vartheta * \eta}{|\gamma|^{\{\frac{1}{\alpha}\}}}$ (15)

$$\eta = \left(\frac{\left(\frac{\pi\alpha}{2}\right)}{\partial\frac{(1+\alpha)}{2} * \alpha * 2^{\frac{\alpha-1}{2}}}\right)^{1/\alpha} \tag{16}$$

Update of hawks in soft besiege with constructive dives calculated using below eqn.17.

$$y(t+1) = \{x, \ if \ f(x) < f(y(t)) \ z, \ if \ f(z) < f(y(t)) \tag{17}$$

In eqn.17 x and z are calculated using eqns. 13, 14.

Update of hawks in hard besiege with constructive dives calculated using below eqn.18.

$$y(t+1) = \{x, \ if \ f(x) < f(y(t)) \ z, \ if \ f(z) < f(y(t)) \tag{18}$$

In eqn.18. x, z are calculated using eqns. 19, 14.

$$x = y_{rab}(t) - en|My_{rab}(t) - y_m(t)| \tag{19}$$

Where $y_m(t)$ is calculated using eqn.8.

4. Proposed SLA based workflow scheduling algorithm using Haris Hawks Optimization

Input: Considered workflow $g = (v, e)$, number of tasks $t^k = \{t^1, t^2 \dots \dots t^k\}$, number of VMs $v^n = \{v^1, v^2, \dots., v^n\}$

Output: Efficient mapping of workflows onto precised VMs.

---

1. Start
2. Fitness function initialization.
3. Initialization of Haris Hawk population in random manner.
4. Calculate fitness using eqn.6.
5. Calculation of task priorities using eqn.1.
6. If $|en > 1|$ then begin exploration phase
7. vector updation by eqn.7.
8. If $|en > 1|$ then it is in exploitation phase
9. If $R \geq 0.5$ && $|en| \geq 0.5$ begin soft prey catching, then
10. vector updation by eqn.10.
11. Elseif $R \geq 0.5$ && $|en| < 0.5$ then begin hard prey catching then
12. vector updation by eqn.12.
13. Elseif $R < 0.5$ && $|en| \geq 0.5$ then begin soft prey catching with constructive steps then
14. vector updation by eqn.17.
15. Elseif $R < 0.5$ && $|en| < 0.5$ then begin hard prey catching with constructive steps then
16. vector updation by eqn.18.
17. Evaluate makespan, SLA violation percentages using eqns.2,5 check whether they are minimized or not.
18. If they are minimized update those parameters as best otherwise repeat process until all iterations completed.
19. End

---

5. Simulation and Results

In this section we discussed how we have conducted simulation on workflow simulator [16] and generated results using proposed SLA based task scheduling algorithm using Haris Hawks optimization. For entire simulation we used random workload consists of 100 -1000 tasks which ran for 100 iterations. We have used 8GB RAM, 256 GB Hard disk, M1 core chip processor.

5.1 Calculation of makespan

Makespan calculated by considering 1000 tasks and we compared proposed scheduler with existing SOTA approaches.

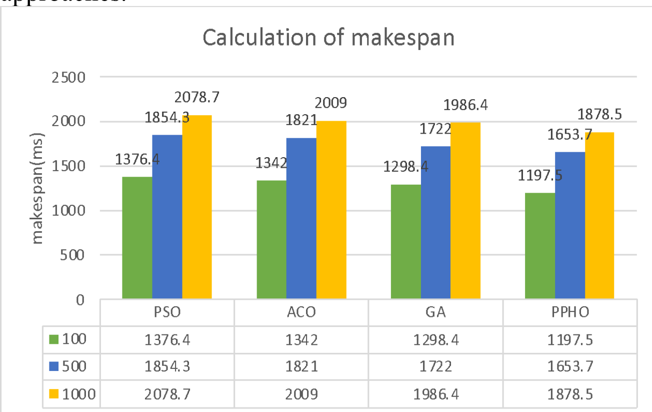

Fig. 3. Evaluation of makespan

Makespan obtained for PSO for 100,500, 1000 tasks is 1376.4, 1854.3, 2078.7 respectively. For ACO with 100,500, 1000 tasks it is of 1342, 1821,2009 respectively. For GA with 100,500,1000 tasks, it is of 1298.4,1722,1986.4 respectively. For PPHO with 100, 500,1000 tasks it is of 1197.5, 1653.7,1878.5 respectively. From above Fig.3. it is evident that SLA aware task scheduler dominates all other baseline mechanisms in view of makespan.

5.2 Calculation of SLA Violation

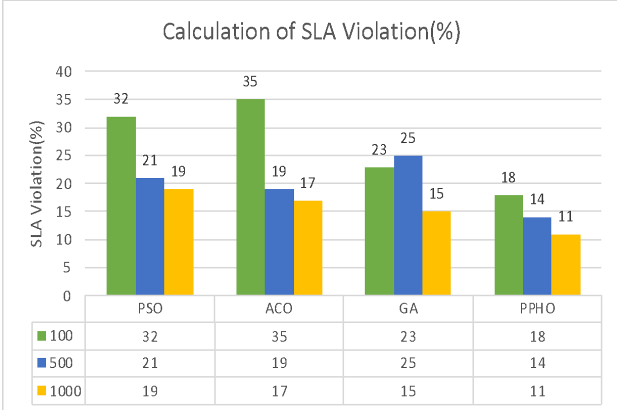

Fig. 4. Evaluation of SLA Violation

For PSO with 100, 500, 1000 tasks SLA violation percentage is of 32, 21,19 respectively. For ACO with 100,500, 1000 tasks it is of 35,19,17 respectively. For GA with 100,500, 1000 tasks it is of 23,25,15 respectively. For PPHO with 100,500, 1000 tasks it is of 18,14,11 respectively. From above Fig.4. it is evident that SLA aware task scheduler dominates all other baseline mechanisms in view of SLA Violation.

6. Conclusions and Future Work

Task scheduling is prodigious challenge in cloud paradigm. Tasks are interdependent on each other, and processing capacities of all tasks vary from each other and to precisely schedule these workflows onto appropriate virtual resource is a challenging issue. Many of existing researchers developed various workflow scheduling approaches using metaheuristic approaches but there is chance to improve scheduling process in cloud paradigm. In this research, we have used Haris hawks optimization as a methodology to tackle scheduling approach. Workflowsim used as a simulation environment to model scheduler. Our proposed SLA based scheduling approach evaluated SOTA approaches, and it dominates over existing algorithms and minimized SLA violations, makespan. In future, we are planning to design a scheduler by employing a machine learning approach to tackle scheduling in an effective manner.

References

[1] Mangalampalli, Sudheer, et al. "Cloud Computing and Virtualization." *Convergence of Cloud with AI for Big Data Analytics: Foundations and Innovation* (2023): 13-40.

[2] Pirozmand, Poria, et al. "An improved particle swarm optimization algorithm for task scheduling in cloud computing." *Journal of Ambient Intelligence and Humanized Computing* (2023): 1-15.

[3] Elcock, Jeffrey, and Nekiesha Edward. "An efficient ACO-based algorithm for task scheduling in heterogeneous multiprocessing environments." *Array* 17 (2023): 100280.
[4] Imene, Latreche, et al. "A third generation genetic algorithm NSGAIII for task scheduling in cloud computing." *Journal of King Saud University-Computer and Information Sciences* 34.9 (2022): 7515-7529.
[5] Mangalampalli, Sudheer, Ganesh Reddy Karri, and Utku Kose. "Multi Objective Trust aware task scheduling algorithm in cloud computing using Whale Optimization." *Journal of King Saud University-Computer and Information Sciences* 35.2 (2023): 791-809.
[6] Heidari, Ali Asghar, et al. "Harris hawks optimization: Algorithm and applications." *Future generation computer systems* 97 (2019): 849-872.
[7] Talha, Adnane, and Mohammed Ouçamah Cherkaoui Malki. "PPTS-PSO: a new hybrid scheduling algorithm for scientific workflow in cloud environment." *Multimedia Tools and Applications* (2023): 1-24.
[8] Shao, Kaili, Ying Song, and Bo Wang. "PGA: A New Hybrid PSO and GA Method for Task Scheduling with Deadline Constraints in Distributed Computing." *Mathematics* 11.6 (2023): 1548.
[9] Praveen, S. Phani, et al. "A Hybrid Gravitational Emulation Local Search-Based Algorithm for Task Scheduling in Cloud Computing." *Mathematical Problems in Engineering* 2023 (2023).
[10] Shingne, Harshala, and R. Shriram. "Heuristic deep learning scheduling in cloud for resource-intensive internet of things systems." *Computers and Electrical Engineering* 108 (2023): 108652.
[11] Pirozmand, Poria, et al. "An improved particle swarm optimization algorithm for task scheduling in cloud computing." *Journal of Ambient Intelligence and Humanized Computing* (2023): 1-15.
[12] Srivastava, Ankita, and Narander Kumar. "An energy efficient robust resource provisioning based on improved PSO-ANN." *International Journal of Information Technology* 15.1 (2023): 107-117.
[13] Mangalampalli, Sudheer, Ganesh Reddy Karri, and G. Naga Satish. "Efficient Workflow Scheduling algorithm in cloud computing using Whale Optimization." *Procedia Computer Science* 218 (2023): 1936-1945.
[14] Shukla, Prashant, and Sudhakar Pandey. "MAA: multi-objective artificial algae algorithm for workflow scheduling in heterogeneous fog-cloud environment." *The Journal of Supercomputing* (2023): 1-43.
[15] Heidari, Ali Asghar, et al. "Harris hawks optimization: Algorithm and applications." *Future generation computer systems* 97 (2019): 849-872.
[16] Chen, Weiwei, and Ewa Deelman. "Workflowsim: A toolkit for simulating scientific workflows in distributed environments." *2012 IEEE 8th international conference on E-science*. IEEE, 2012.