PSO-Based Workflow Scheduling: a Comparative Evaluation of Cloud and Cloud-Fog Environments

Dineshan Subramoney and Clement Nyirenda

November 1, 2021

PSO-Based Workflow Scheduling: A Comparative Evaluation of Cloud and Cloud-Fog Environments

Dineshan Subramoney, Clement Nyirenda
Department of Computer Science,
Faculty of Natural Sciences,
University of the Western Cape,
Cape Town 7535, South Africa
[1]cnyirenda@uwc.ac.za
[2]dineshansubramoney@gmail.com

***Abstract*—This paper presents a comparative evaluation of cloud and cloud-fog environments for workflow scheduling using the classical Particle Swarm Optimization (PSO) algorithm. This paper also presents a weighted sum objective function based on three objectives: makespan, cost and energy. The recently proposed FogWorkflowSim is used as the simulation environment for the cloud and cloud-fog architectures. The results obtained for two well-known scientific workflows (Montage and CyberShake) show that the incorporation of the fog layer for the execution of workflows has the potential to improve processing efficiency and reduce energy consumption, motivating the cloud-fog computing paradigm. Future work will focus on the evaluation of other types of workflows such as Epigenomics, LIGO and SIPHT as well as increasing the number of tasks in a workflow.**

***Index Terms*—Workflow scheduling, Fog Computing, Cloud Computing, Particle Swarm Optimization**

I. Introduction

As cloud computing [1] becomes more and more established in the ICT industry, the possibility of conducting large-scale scientific computations, that could not be done on traditional computing systems is becoming a reality. Scientific workflows [2]–[4] are some of the data-intensive scientific applications that are really benefiting from the cloud computing revolution. In a nutshell, scientific workflow is characterized by interdependent tasks and computations that are aimed at achieving some scientific objectives. The cloud infrastructure offers a suitable platform for executing scientific workflows because these applications involve complex data and are also characterized by long sessions of computation. Furthermore, the cloud infrastructure offers other crucial attributes to workflow computations such as cost efficiency, high speed, accessibility, manageability, elasticity, and virtualization capabilities.

Scientific workflows are typically described as a directed acyclic graph (DAG), where the nodes are tasks and the edges denote the task dependencies [4]. The scheduling of these tasks for execution on cloud virtual machines presents huge challenges because of high computation and communication costs [5]. Population-based techniques such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) have been used for scheduling the tasks on the cloud infrastructure [5]–[10]. Currently, the PSO approaches seem to be the most favourite amongst these population-based techniques as evidenced by a recent extensive survey on PSO-based approaches [9]. The interest in PSO is sparked by its fast convergence and short run time.

While most of the aforementioned works have focused on workflow scheduling in the traditional two-tier cloud infrastructure, very few [10]–[12] have focused on the emerging three-tier infrastructure, which incorporates the fog devices between the cloud and the end devices. Fog computing extends the cloud computing model and acts as an intermediate layer. Any device with computing, storage and networking capabilities can be considered a fog device. Among other things, fog devices are characterised by the following advantages, compared to their cloud counterparts: lower latency, improved user experience, higher security, and energy efficiency.

A simulation toolkit, known as Fogworkflowsim [13], for workflow scheduling in fog computing was recently proposed. This work was inspired by the WorkflowSim [14], which has been used for simulating workflow scheduling in cloud environments for close to a decade. In [12], a comparative evaluation of population-based optimization algorithms for Workflow Scheduling in cloud-fog Environments has been conducted. The work developed a weighted sum objective function that incorporates makespan, cost, and energy consumption and proceeded to implement the Genetic Algorithm (GA), PSO, Differential Evolution (DE), and the PSO-GA algorithms in the Fogworkflowsim [13] toolkit.

The paper builds on the work in [12] by exclusively focusing on the PSO algorithm's workflow scheduling performance in two architectures: the traditional cloud environment and the emerging three-tier cloud-fog environment. To the best of our knowledge, no work has so far compared PSO workflow scheduling performance under the two environments. The work is preliminary in nature as it uses only two types of scientific workflows within the Pegasus framework, namely: Montage and CyberShake. Makespan, energy consumption, and cost are used as performance metrics.

The rest of the paper is organised as follows. Section II briefly describes the standard PSO algorithm. Section III presents the workflow Scheduling basics as well as the objective function, that was used in this work. Section IV presents the PSO optimization process for workflow scheduling, while Section V presents the performance evaluation. Section VI concludes the paper and presents future work.

II. Particle Swarm Optimization Algorithm

Particle swarm optimization (PSO) is a population-based stochastic optimization algorithm introduced by Kennedy and Eberhart [15]. The technique is used to solve optimization problems by emulating the social behavior of bird flocking, fish schooling and other animal societies that cooperate and share information to improve their position without relying on a leader. In this technique, a population of individuals, that are candidate solutions represented as particles, move in a given solution space according to its current position X_i^k, and current velocity V_i^k for the k^{th} iteration. The quality of each particle is measured using a defined fitness function depending on the optimization problem. Each particle's movement is based on its best known personal position $pBest_i$, and also moves towards the best known global position $gBest$ for the entire swarm. This process leads the swarm to the best position over a number of iterations in the search process. The particle's velocity and position are described below:

$$\begin{aligned} V_i^{k+1} = \omega V_i^k + c_1 r_1 (pBest_i - X_i^k) \\ + c_2 r_2 (gBest_i - X_i^k) \end{aligned} \tag{1}$$

$$X_i^{k+1} = X_i^k + V_i^k \tag{2}$$

where ω is the inertia weight, r_1 and r_2 are random numbers between (0,1), and c_1 and c_2 are the learning factors.

III. Workflow Scheduling Basics and Objective Function

This section begins by defining the workflow scheduling problem. Then, the weighted sum based objective function is presented for workflow scheduling.

A. The Concept of Workflows and Problem Formulation

The workflow application is represented as a Direct Acyclic Graph (DAG), defined by $G = (T, E)$, where T denotes the set of n tasks $\{t_1, t_2, \ldots, t_n\}$ and E is the set of edges, representing the dependencies between pairs of tasks [6], [7], [10], [12]. An edge can be illustrated by $d_{ij} =< t_i, t_j >\in E$, where d_{ij} is a positive value representing the output data from task t_i to t_j. Therefore, the execution of t_j cannot start until t_i has completed. A task t_i with no parent is known as a $start$ task and a task t_j with no child is known as an end task.

Workflow application scheduling in the cloud and cloud-fog environments are defined here as the problem of assigning computing resources with different characteristics to tasks of the workflow application, in order to minimize the total completion time, cost and energy of the workflow execution.

B. The Weighted Sum Objective Function

There are m computational resources which are of two types, namely cloud and fog servers. In the cloud setup, only cloud servers are considered, whereas the cloud-fog approach includes both cloud and fog servers. The source end devices are not considered as computational resources since the aim of this work is to evaluate the incorporation of the fog layer to the existing cloud paradigm. Next, the mathematical formulas are defined for makespan, cost and energy consumption, and finally the presentation of the weighted sum based objective function.

1) Makespan: The makespan MS is calculated as follows:

$$MS = max\{FT_{t_i}, t_i \in T\} - min\{ST_{t_i}, t_i \in T\} \tag{3}$$

where ST_{t_i} and FT_{t_i} are the starting time and finishing times respectively for task t_i in a workflow.

2) Cost: The computation cost and communication cost are considered in this work. The computational cost [10] when using computing resource r is

$$CE_i^r = pr * (FT_{t_i} - ST_{t_i}), \tag{4}$$

where pr is the unit processing cost. The communication cost for a particular task refers to the cost of sending a task output of size d_{ij} from the resource processing task i to the resource allocated to process task j.

$$CC_{ij} = trc_{ij} * d_{ij}, \tag{5}$$

Therefore, the total cost TC is

$$TC = \sum_{i=1}^{n} \sum_{j=1}^{n} CC_{ij} + \sum_{r=1}^{m} \sum_{i=1}^{n} CE_i^r. \tag{6}$$

3) Energy Consumption: The energy consumption model [16] is constructed using active E_{active}, that is energy consumed when a task is being executed, and idle E_{idle}, the energy used when a resource is idling. The active energy is calculated by using

$$E_{active} = \sum_{i=1}^{n} \alpha f_i v_i^2 (FT_{t_i} - ST_{t_i}), \tag{7}$$

where α is a constant; f_i and v_i are the frequency and voltage for the resource on which task i is being executed. The energy consumed during the idle period [16], [12] is determined by using

$$E_{idle} = \sum_{j=1}^{m} \sum_{idle_{jk} \in IDLE_{jk}} \alpha f_{min\ i} v_{min\ i}^2 L_{jk}, \tag{8}$$

where $IDLE_{jk}$ is a set of idling slots on resource j, $f_{min\ i}$ and $v_{min\ i}$ refer to the frequency and lowest supply voltage on resource j respectively; L_{jk} is the duration of idling time for $idle_{jk}$. The total energy TE consumed for the execution of the entire workflow is

$$TE = E_{active} + E_{idle}. \tag{9}$$

Therefore, using the three aforementioned objectives, the weighted sum objective function is defined by:

$$F(M) = w_1 * MS_{norm} \; + \; w_2 * TC_{norm} \\ + \; w_3 * TE_{norm}, \tag{10}$$

where M is the assignment of the n tasks of a workflow to the m available computing resources. MS_{norm}, TC_{norm} and TE_{norm} are the normalized makespan, total cost and total energy respectively; w_1, w_2 and w_3 are the coefficient weights. Equal weighted coefficients are used here to obtain a balanced contribution of the three objectives since all are equally important in a good solution. Normalization is used here to eliminate any biases in the objective function and is described in our previous paper [12].

IV. The PSO Optimization Process for Workflow Scheduling

This section firstly describes the encoding of the particle for PSO and how this mapping is used to generate a task-resource schedule. The second part describes the PSO optimization process.

A. Description of the Particle for PSO

The workflows used in this work can be scheduled for execution at the fog server or at the cloud server. Each computational resource has its own computational capacity, power and bandwidth.

Since the scheduling of workflow tasks in a computing environment is a discrete problem, we use natural numbers to encode the individuals for the PSO algorithm. The individuals of the PSO are represented by the particles that are mappings of task-resource schedules. The dimension or length of each particle is n which is the total number of tasks in the workflow. Each position in the particle is a positive integer representing the task number. The value assigned to this position is the server ID that is allocated to execute the task. The ID numbers are selected from the servers available on the respective architecture tier. Suppose a workflow has 10 tasks which are scheduled for execution on 5 available servers. In this instance, the particle's length is 10 and each element is an integer between 1 and 5. An example task assignment of this particle can be expressed as {3,3,2,4,5,3,2,1,5,1}. This particle's possible schedule for the cloud and cloud-fog environments are illustrated in Table I and Table II.

Algorithm 1 illustrates the PSO optimization process. Parameters N and G denote the number of particles and the number of generations respectively. while the other parameters have already been defined in Section II. The algorithm starts with the initialization of N, c_1, c_2, ω, and G. It then proceeds by creating N particles, each of which is evaluated by running the respective workflow scheduling on the Fogworkflowsim [13] toolkit. The fitness function value is evaluated and the personal best and the global best values are determined. After that the algorithms goes into iterative process for G generations. It updates the global best and the personal best values whenever it gets better values.

TABLE I
Example of the Task-Resource Allocation using only the Cloud Layer

Layer	Cloud	Cloud	Cloud	Cloud	Cloud
Server ID	1	2	3	4	5
Assigned Task	8,10	3,7	1,2,6	4	5,9

TABLE II
Example of the Task-Resource Allocation on the Cloud and Fog Layers

Layer	Fog	Fog	Cloud	Cloud	Cloud
Server ID	1	2	3	4	5
Assigned Task	8,10	3,7	1,2,6	4	5,9

Algorithm 1: Particle Swarm Optimization (PSO) Algorithm

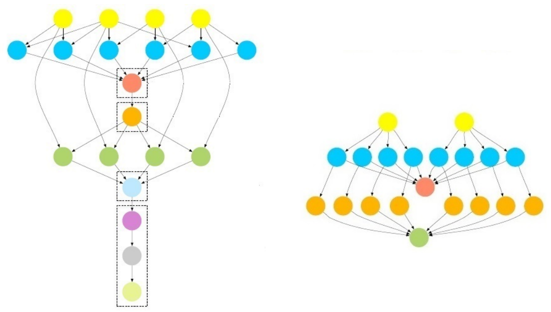
Input: N, c_1, c_2, ω, and G;
Output: $gBest$ and $F(gBest)$;
Randomly generate N particles;
$F(gBest) \leftarrow 0$;
for $i \leftarrow 1, N$ **do**
    Invoke the Fogworkflowsim workflow scheduler.;
    Compute the fitness function value, $F(x_i)$, for particle i, by using the Weighted Sum Objective Function from Section III (B) ;
    $pBest_i \leftarrow x_i$;
    $F(pBest_i) \leftarrow F(x_i)$;
    if $F(x_i) > F(gBest)$ **then**
        $gBest \leftarrow x_i$;
        $F(gBest) \leftarrow F(x_i)$;
$t \leftarrow 0$;
while $t \leq G$ **do**
    for $k \leftarrow 1, N$ **do**
        (1) Update v_k and x_k by using Eq. 1 and Eq. 2;
        (2) Invoke the Fogworkflowsim workflow scheduler. ;
        (3) Compute the fitness function value, $F(x_k)$, by using the Weighted Sum Objective Function from Section III (B);
        if $F(x_k) > F(pBest_k)$ **then**
            $pBest_k \leftarrow x_k$;
            $F(pBest_k) \leftarrow F(x_k)$;
        if $F(x_k) > F(g^b)$ **then**
            $gBest \leftarrow x_k$;
            $F(gBest) \leftarrow F(x_k)$;
    $t \leftarrow t + 1$;

V. Performance Evaluation

In this section, the workflow models are presented along with the simulation setup on the FogWorkflowSim tool [13], and finally, a discussion on the simulation results.

A. Workflow Models

This work uses two well-known scientific workflows that have been studied extensively in research, namely Montage and CyberShake [17]. The graphical representation of the workflows are shown in Fig. 1. The Montage workflow, created by NASA/IPAC, represents an astronomy application that combines multiple input images to create custom mosaics of the sky. It has a more sequential structure with a pipeline of tasks, while CyberShake requires more parallel processing of tasks for characterizing earthquake hazards threatening a region. Together, these workflows are composed of a variety of structures that provide a good basis for performance evaluations.



(a) Montage (b) CyberShake

Fig. 1. Scientific workflows [17]

B. Simulation Environment

The simulations are done using the FogWorkflowSim simulator. The simulator is executed using the Eclipse Java IDE on a computer with 64-bit Windows 10 operating system, Intel(R) Core(TM) i5- 5200U CPU @ 2.20GHz and 16 GB RAM. The population size of PSO is set to 50. The PSO learning factors C1 = C2 = 2. The inertia weight is 1. The number of iterations is 100. The weighted coefficients w_1, w_2 and w_3 are equal. The two scientific workflows with 500 tasks each are used as input where the workflow is a DAG XML file representation of the workflow generated by Pegasus [18]. The simulations are performed 10 times for each workflow and environment setup. The number of cloud servers and fog servers are 10 and 6, respectively. The characteristics of each server on the two tiers along with the parameter settings for the simulation environment are shown in Table III.

TABLE III
PARAMETER SETTINGS OF SIMULATION ENVIRONMENT

Parameters	Fog Server	Cloud Server
Processing rate (MIPS)	1000	2000
Task execution cost ($)	0.48	0.96
Communication cost ($)	0.01	0.02
Working power (mW)	700	1700
Idle power (mW)	200	1200
Uplink bandwidth (Mbps)	500	300
Downlink bandwidth (Mbps)	800	500

C. Simulation Results

In Figs. 2-4, the results for makespan, cost and energy consumption for the Montage workflow for 500 tasks are illustrated. The makespan is lower for the cloud-fog environment, as expected, due to the 6 additional fog servers used in the simulation. The cost metric is significantly better in the cloud-fog layers. This is likely due to the reduced processing and data transfer costs associated with the fog layer. The energy consumption is also reduced in cloud-fog as the larger size of the cloud requires more energy to remain online, and utilizing the fog with the cloud enables more efficient and distributed processing of the workflow tasks.

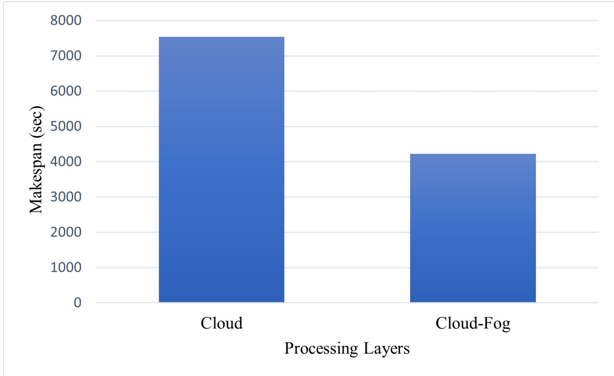


Fig. 2. Makespan for Montage

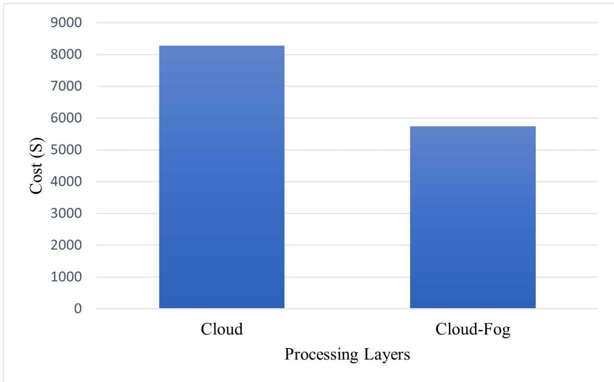


Fig. 3. Cost for Montage

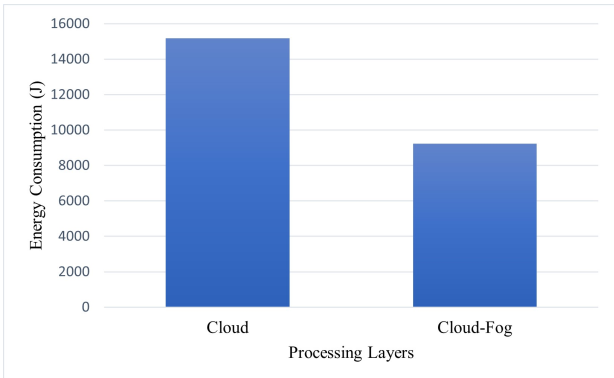


Fig. 4. Energy consumption for Montage

In Figs. 5-7, the results for makespan, cost and energy consumption for the CyberShake workflow for 500 tasks are illustrated. The performance metric comparative results for the cloud and cloud-fog are similar to what was observed for the Montage workflow, however, the metric values are significantly higher. This is because the CyberShake workflow task sizes and task runtimes are much higher compared to Montage.

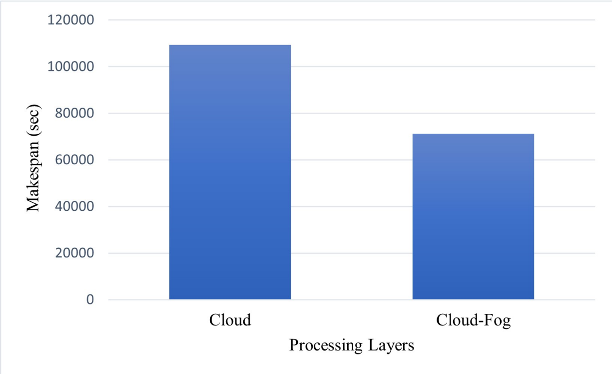


Fig. 5. Makespan for CyberShake

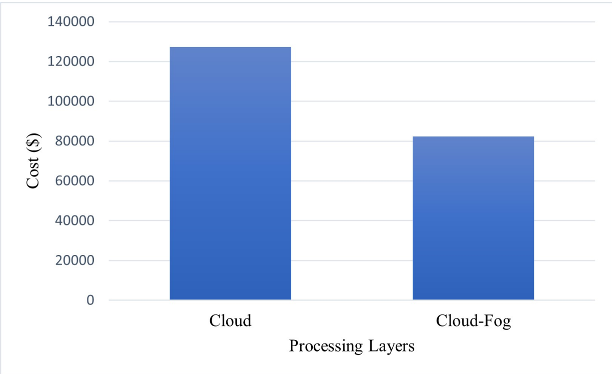


Fig. 6. Cost for CyberShake

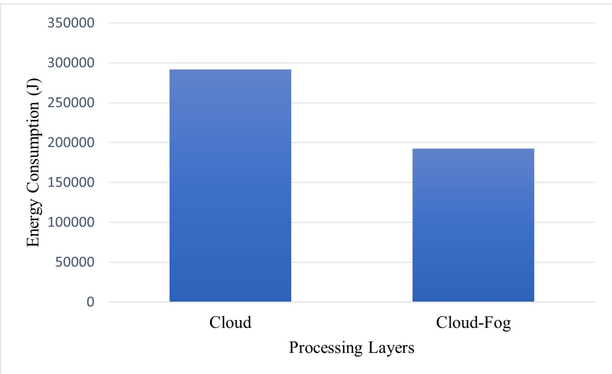


Fig. 7. Energy consumption for CyberShake

VI. Conclusion and Future Work

This paper has presented a comparative evaluation of workflow scheduling in cloud and cloud-fog environments using the classical PSO algorithm on the FogWorkflowSim simulator. A weighted sum objective function made up of makespan, cost and energy is described for workflow scheduling. Results show that the cloud-fog environment performed better than the cloud especially in terms of overall cost and energy consumption. Therefore, the incorporation of the fog layer for the execution of workflows has the potential to improve processing efficiency justifying the benefits of the emerging cloud-fog computing paradigm.

In future, the number of tasks in each of the workflows will be increased. Other types of workflows such as Epigenomics, LIGO and SIPHT will be evaluated. The number of fog servers, optimization objectives will be increased, and deadline and budget constraints will be incorporated.

References

[1] R. Buyya, S. Pandey, and C. Vecchiola, "Cloudbus toolkit for marketoriented cloud computing," *Proceedings of the 1st International Conference on Cloud Computing*, vol. 5931, pp. 24–44, 2009.

[2] W. Song, F. Chen, H.-A. Jacobsen, X. Xia, C. Ye, and X. Ma, "Scientific workflow mining in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2979–2992, 2017.

[3] Y. Zhao, Y. Li, I. Raicu, S. Lu, C. Lin, Y. Zhang, W. Tian, and R. Xue, "A service framework for scientific workflow management in the cloud," *IEEE Transactions on Services Computing*, vol. 8, no. 6, pp. 930–944, 2014.

[4] M. A. Rodriguez and R. Buyya, "A taxonomy and survey on scheduling algorithms for scientific workflows in iaas cloud computing environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28(10), p. e4041, 2017.

[5] A. Verma and S. Kaushal, "A hybrid multi-objective particle swarm optimization for scientific workflow scheduling," *Parallel Computing*, vol. 62, pp. 1–19, 2017.

[6] A. Singh *et al.*, "A multi-objective workflow scheduling algorithm for cloud environment," in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*. IEEE, 2018, pp. 1–6.

[7] A. M. Manasrah and H. Ba Ali, "Workflow scheduling using hybrid ga-pso algorithm in cloud computing," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[8] M. Farid, R. Latip, M. Hussin, and N. A. W. Abdul Hamid, "A survey on qos requirements based on particle swarm optimization scheduling techniques for workflow scheduling in cloud computing," *Symmetry*, vol. 12, no. 4, p. 551, 2020.

[9] M. Masdari, F. Salehi, M. Jalali, and M. Bidaki, "A survey of pso-based scheduling algorithms in cloud computing," *Journal of Network and Systems Management*, vol. 25, no. 1, pp. 122–158, 2017.

[10] Y. Xie, Y. Zhu, Y. Wang, Y. Cheng, R. Xu, A. S. Sani, D. Yuan, and Y. Yang, "A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment," *Future Generation Computer Systems*, vol. 97, pp. 361–378, 2019.

[11] V. De Maio and D. Kimovski, "Multi-objective scheduling of extreme data scientific workflows in fog," *Future Generation Computer Systems*, vol. 106, pp. 171–184, 2020.

[12] D. Subramoney and C. N. Nyirenda, "A comparative evaluation of population-based optimization algorithms for workflow scheduling in cloud-fog environments," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 760–767.

[13] X. Liu, L. Fan, J. Xu, X. Li, L. Gong, J. Grundy, and Y. Yang, "Fogworkflowsim: An automated simulation toolkit for workflow performance evaluation in fog computing," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2019, pp. 1114–1117.

[14] W. Chen and E. Deelman, "Workflowsim: A toolkit for simulating scientific workflows in distributed environments," *8th IEEE International Conference on E-science*, pp. 1–8, 2012.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization in: Neural networks," in *Proceedings IEEE International Conference on 1995*, 1942, pp. 1942–1948.

[16] S. Yassa, R. Chelouah, H. Kadima, and B. Granado, "Multi-objective approach for energy-aware workflow scheduling in cloud computing environments," *The Scientific World Journal*, vol. 2013, 2013.

[17] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.-H. Su, and K. Vahi, "Characterization of scientific workflows," in *2008 third workshop on workflows in support of large-scale science*. IEEE, 2008, pp. 1–10.

[18] The Pegasus website, [Online]. Available: https://pegasus.isi.edu/. [Accessed 22 May 2021].

Clement N. Nyirenda received his PhD in Computational Intelligence from Tokyo Institute of Technology in 2011. His research interests are in Computational Intelligence paradigms such as Fuzzy Logic, Swarm Intelligence, and Artificial Intelligence and their applications in Communications.

Dineshan Subramoney received his Bachelors Degree (Honors) in Computer Science from the University of KwaZulu-Natal. He is currently working on his MSc in Computer Science at the University of the Western Cape. His research interest is in Workflow Scheduling using Population-based Algorithms such as Particle Swarm Optimization.