34th IEEE/ACM International Conference on Automated Software Engineering (ASE 2019), San Diego, CA, USA, November 11-15, 2019.

FogWorkflowSim: An Automated Simulation Toolkit for Workflow Performance Evaluation in Fog Computing

Xiao Liu[1], Lingmin Fan[2], Jia Xu[2], Xuejun Li[2], Lina Gong[2], John Grundy[3], Yun Yang[4]
[1]School of Information Technology, Deakin University, Geelong, Australia
[2]School of Computer Science and Technology, Anhui University, Hefei, China
[3]Faculty of Information Technology, Monash University, Melbourne, Australia
[4]School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia
xiao.liu@deakin.edu.au; xjli@ahu.edu.cn; john.grundy@monash.edu; yyang@swin.edu.au

***Abstract*—Workflow underlies most process automation software, such as those for product lines, business processes, and scientific computing. However, current Cloud Computing based workflow systems cannot support real-time applications due to network latency, which limits their application in many IoT systems such as smart healthcare and smart traffic. Fog Computing extends the Cloud by providing virtualized computing resources close to the End Devices so that the response time of accessing computing resources can be reduced significantly. However, how to most effectively manage heterogeneous resources and different computing tasks in the Fog is a big challenge. In this paper, we introduce "FogWorkflowSim" an efficient and extensible toolkit for automatically evaluating resource and task management strategies in Fog Computing with simulated user-defined workflow applications. Specifically, FogWorkflowSim is able to: 1) automatically set up a simulated Fog Computing environment for workflow applications; 2) automatically execute user submitted workflow applications; 3) automatically evaluate and compare the performance of different computation offloading and task scheduling strategies with three basic performance metrics, including time, energy and cost. FogWorkflowSim can serve as an effective experimental platform for researchers in Fog based workflow systems as well as practitioners interested in adopting Fog Computing and workflow systems for their new software projects. (Demo video: https://youtu.be/AsMovcuSkx8)**

Keywords—Fog Computing, Workflow, Performance Evaluation, Task Scheduling, Simulation Toolkit

I. INTRODUCTION

With the emergence of Internet of Things (IoT), many smart devices are being developed with rapidly increasing number of smart IoT applications [1]. However, a large number of IoT applications which is response sensitive cannot be supported effectively by Cloud services over the Internet [2]. To address the limitation of Cloud Computing in supporting smart applications, Fog Computing (sometimes called Edge Computing) has been proposed as a latest computing paradigm [3]. Fog Computing is to create an extra layer of Fog Nodes between the layer of End Devices and the layer of Cloud Servers, which can provide computation resources much closer to the End Devices [4]. As a result, the requirement of real-time response can now be met.

Consider a real-world example. Fig. 1 describes a heart rate monitoring application in a smart healthcare system which illustrates the collaboration between Fog Nodes and Cloud Servers. The End Device layer collects the data of patients by various kinds of sensors in real time. The Fog Node layer can process the data from the sensors to get heart rate data, and then transmit the data to the Cloud Server for storage. It can also make emergency call to the doctor once abnormal heart rate is detected [5]. Then the doctor can inquire the patient's heart rate record from both Fog Nodes and Cloud Servers for diagnosis. In such a system, numbers of Fog Nodes are processing the data received from different patients in parallel to effectively relieve the pressure on data communication and computation of Cloud Servers [6]. Also, it can timely respond to the patient's emergency even when the connection to Cloud severs is down.

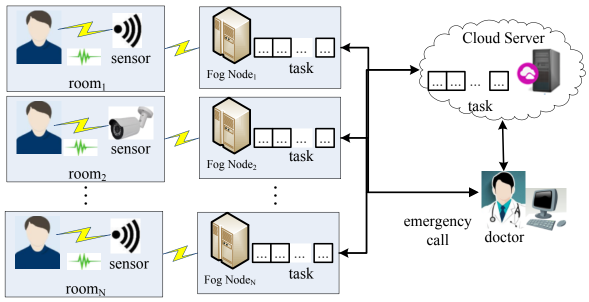

Fig. 1. A scenario of heart rate monitoring

This Fog Computing environment includes three layers of resources and as each has unique characteristics, resource management in Fog Computing is a very challenging issue. It is also very difficult and expensive to create a real-world Fog Computing environment for research purposes due to its complexity in networks and resources. Workflow systems are a major tool for process automation and underly most automated software applications, such as product lines, business processes, scientific computing, and many smart applications [7]. The study of Fog-based workflow systems is an important topic. However, since Fog Computing is still in its infancy, there is no existing simulation toolkit which can simulate the running of workflow applications in a Fog Computing environment and facilitate the evaluation of different resource and task management strategies.

To address this problem, we propose "FogWorkflowSim" a comprehensive, automated simulation toolkit for Fog based workflow systems. FogWorkflowSim integrates the simulation of Fog Computing and workflow systems. Specifically, the major capabilities of FogWorkflowSim include: 1) it can automatically simulate the complex network topology and various types of computing resources in the Fog; 2) it can automatically execute user submitted workflow applications; 3) it can automatically evaluate and compare the performance of different computation offloading and task scheduling strategies for Fog workflows.

II. Related Work and Motivation

Currently, there are many Cloud workflow simulation tools. CloudSim and WorkflowSim are the two most representative projects. CloudSim is an extensible simulation toolkit which can model and simulate the Cloud environment as well as the Cloud applications [8]. However, without the integration of a workflow system, CloudSim can only support the execution of simple independent tasks. Hence，WorkflowSim extends CloudSim to include the capability of running workflow applications [9]. It incorporates workflow scheduling and execution, and supports workflows with different structures.

Since Fog Computing is still in its infancy, there are only a few simulation toolkits, e.g. iFogSim [10]. iFogSim is a simulation toolkit for the Fog environment and it supports the modelling of Fog network topology and Fog resources in different layers. iFogSim simulates the running of independent computation tasks and supports computation offloading strategies which determine the specific layer that the computation tasks will be offloaded to. However, iFogSim cannot support workflow applications as there are too many complicated tasks and configuration dependencies.

We wanted to investigate the problem of how to integrate the simulation of Fog Computing with workflow systems so as to address the challenging issues of resource and task management in the Fog.

III. FogWorkflowSim System Design

The foundation of FogWorkflowSim is the simulation of a complex Fog Computing environment and complex workflow system. Instead of building these from scratch, FogWorkflowSim inherits the functions of iFogSim [10] and WorkflowSim [9]. We integrated the functions of iFogSim with WorkflowSim, and modified resource allocation strategies to achieve integration of the two systems.

Fig. 2 shows the three layers of FogWorkflowSim: the Fog Computing environment layer, the workflow system layer and the resource management layer. Each layer is responsible for specific functions to facilitate the operations of higher levels. The three layers are described below.

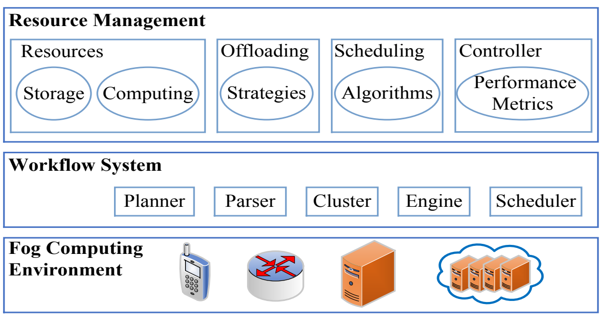

Fig. 2. The Architecture of FogWorkflowSim

A. Fog Computing environment layer

The Fog Computing environment has three layers: the End Device layer, the Fog Node layer and the Cloud Server layer. We use the Fog Device class to represent all types of resources in the Fog. It can be used to simulate different devices by changing the specifications for the hardware such as the computation capacity, storage capacity, uplink/downlink bandwidth, and so on. It also provides the interface for the management and allocation of these hardware resources. Methods in this class also include the simulation of running allocated workflow tasks, the simulation of computation and storage resources and so on.

B. Workflow system layer

The workflow system layer consists of the Planner module, the Parser module, the Cluster module, the Engine module, and the Scheduler module. The Planner module is responsible for the start of simulation. The Parser module is responsible for parsing the input workflow file in XML format into Task class which represents workflow task in the system. The Cluster module is responsible for clustering several tasks into a job according to a specific clustering algorithm. The Engine module is to submit jobs to the Scheduler module according to the dependencies of tasks, re-schedule failed tasks and end the simulation if all tasks are completed. The most important module is the Scheduler module. This serves as the entry of the workflow scheduling algorithm, and it is also responsible for the creation of Virtual Machines (VM) and the submission, update and return of workflow tasks.

C. Resource management layer

The resource management layer is divided into Resource module, Offloading module, Scheduling module and Controller module. The Resource module is a virtualized pool for computation and storage resources of various types. Offloading and Scheduling modules are extensible libraries for various computation offloading strategies and task scheduling algorithms respectively in the system. The Controller module is a base for the models of various performance metrics that are used to evaluate the performance of running workflow applications.

IV. System Functions

The input to FogWorkflowSim is workflow files in XML format. These can be generated by popular tools such as WorkflowGenerator [11]. The system also supports customized workflows that need to be converted to XML format. The user interface of FogWorkflowSim has three pages: the main page, the Fog environment setting page and the algorithm setting page. These are shown in Fig. 3 and Fig. 4. Below, we review the main system functions:

Setting of Fog Computing environment. In Fig. 3, the red box marked 1 is for configuring the Fog Computing environment where the amount of devices in each layer can be set. The user can define detailed settings in the left page of Fig. 4 by clicking the ‘More Details’ button.

Configuration of Fog resources. The total amount of Fog resources is determined when the Fog Computing environment is initialized. Each FogDevice can manage its own computation, storage and other types of resources.

Selection of computation offloading strategies, workflow scheduling algorithms and their optimisation objectives. Box number 2 in Fig. 3 is a panel where the computation offloading strategy, scheduling algorithm and optimisation objective can be selected. MinMin, MaxMin, FCFS and RoundRobin algorithms are used only for the optimisation of time. PSO (Particle Swarm Optimisation) and GA (Genetic Algorithm) are metaheuristics based algorithms used for optimisation objectives (time, energy and cost). Box 2 in Figure 4 is for setting the parameters of PSO and GA algorithms respectively. The user can set the parameters either by filling in manually or importing them.

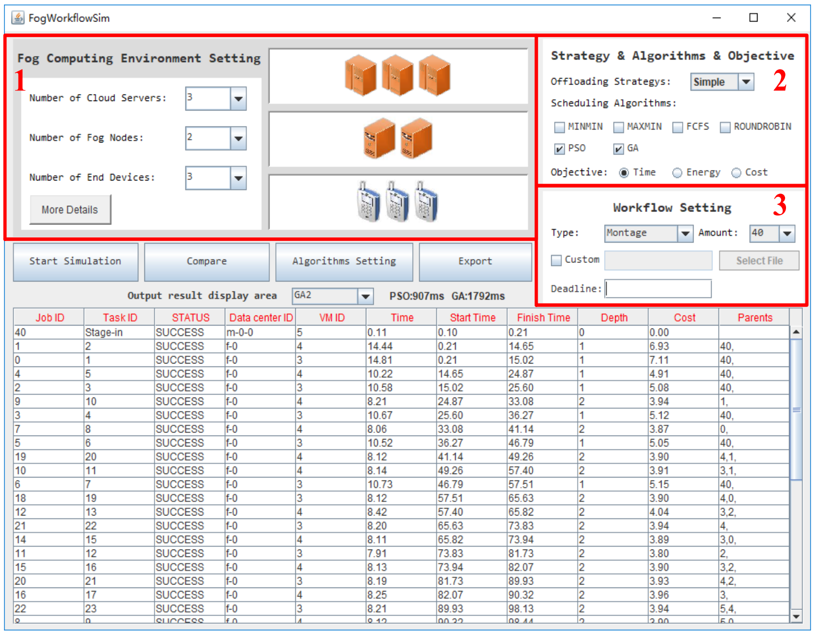

Fig. 3. Main page of FogWorkflowSim

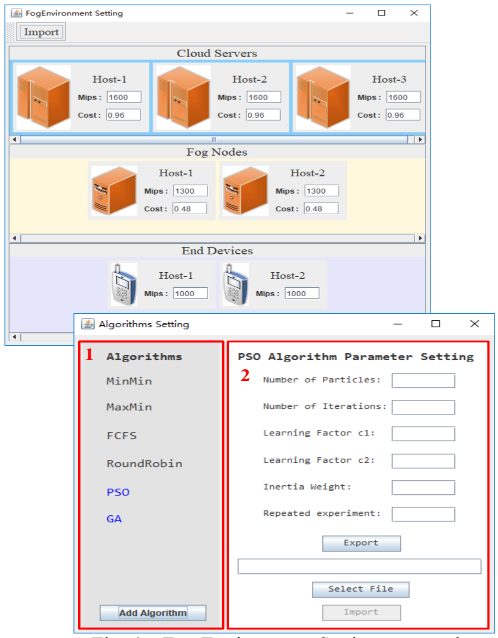

Fig. 4. Fog Environment Setting page and Algorithm Setting page of FogWorkflowSim

Setting of workflow applications. The box marked 3 in Fig. 3 is for setting the workflows where user can choose the workflow structure, the amount of workflow tasks and the deadline of workflow (as a very important quality constraint for workflow applications). In addition, user's own workflow files can be imported.

Extension support of performance metrics. A library stores performance metrics such as time, energy and cost, it also supports user defined performance metrics.

Extension support for computation offloading strategies. A library stores the computation offloading strategies such as All-in-Fog (namely without the use of Cloud Servers), All-in-Cloud (namely without the use of Fog Nodes), and Simple (reduce energy consumption of the End Device under response time constraints) as we proposed in [5]. The system also supports extension of user defined computation offloading strategies.

Extension support for workflow scheduling algorithms. There are many workflow scheduling algorithms in the library including MinMin, MaxMin, FCFS, RoundRobin, PSO and GA. In addition to these built-in scheduling algorithms, FogWorkflowSim also supports the extension of user submitted algorithms. The customized algorithms can be imported to the system when user clicks the 'Add Algorithm' button in the box marked with number 1 in Fig. 4. Due to the space limit, please visit our project website[1] to see the detailed instructions for how to extend the library with user submitted algorithms.

V. Evaluation

In our experiments, Fog based workflow can achieve a significant reduction (around 39% to 58%) in time and cost compared with Cloud based workflow (namely if All-in-Cloud is selected as the computation offloading strategy). This is consistent with our previous work and other related work. Therefore, to simplify the demonstration and focus on the evaluation of different task scheduling algorithms, the experiment in this section was carried out with the same Simple computation offloading strategy [5].

A. Experimental Environment and Parameter Settings

FogWorkflowSim runs on a desktop computer with the following configurations: Intel core i7, dual-core 3.2GHz CPU, 16 GB RAM, and Microsoft Windows 10 OS. FogWorkflowSim is developed in Java JDK 1.8.

TABLE I. Parameters Settings of Fog Computing Environment

Parameters	Mobile Device	Fog Node	Cloud Server
MIPS	1000	1300	1600
Working Power (mW)	700	0	0
Idle Power (mW)	30	0	0
Data Transmission Power (mW)	100	0	0
Data Receiving Power (mW)	25	0	0
Task Execution Cost ($)	0	0.48	0.96

FogWorkflowSim supports different workflow structures with different number of tasks. In our experiments, we use the Montage workflow to compare the impact of different number of workflow tasks on the task scheduling results. The parameter settings for the Fog Computing environment are described in TABLE I. Please note that the power for Fog Nodes and Cloud Servers are not considered as our focus is on the energy consumption of End Devices. Using the similar settings as in previous work [12], the uplink and downlink bandwidths of the End Device layer are 20Mbps and 40Mbps respectively. In the simulation, we consider the situation with 2 Fog Nodes, 3 Cloud Servers and 1 End Device in the Fog environment.

In the current version of FogWorkflowSim, we implemented six representative task scheduling algorithms. For PSO algorithm, the particle number is 30. The learning factors C1 and C2 are both 2. The inertia weight is 1. For GA algorithm, the population size is 50. The rates of crossover and mutation are 0.8 and 0.1 respectively. The iteration numbers of PSO and GA are both 100.

[1] Project at GitHub: https://github.com/CCIS-AHU/FogWorkflowSim.git

B. Experimental Results and Analysis

A summary of experimental results on time, energy and cost for all six algorithms is shown in Fig. 5. The detailed results on task execution time are shown in Fig. 6. The results of task execution time for two metaheuristics based algorithms (PSO and GA) are always better than other heuristics scheduling algorithms. With the increase of the number of tasks, the performance gap between GA and PSO is growing. For example, when the number of tasks is 20, the task execution time of GA is 11.94% lower than that of PSO. When the task number becomes 100, it is 16.13% lower. Therefore, it shows that the GA is the most efficient task scheduling algorithm for time optimisation.

Fig. 5. Summary of experimental results

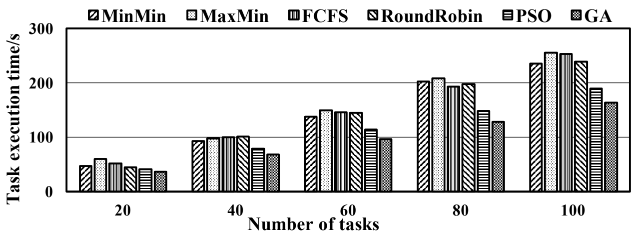

Fig. 6. Comparison of the task execution time

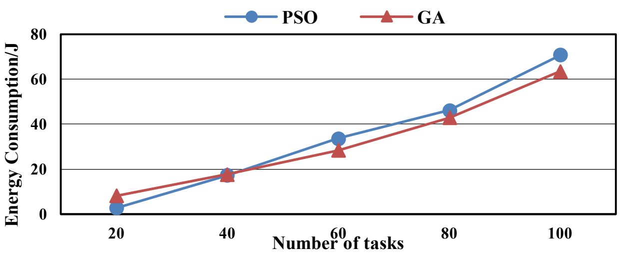

Fig. 7. Comparison of the energy consumption

The results on energy consumption of End Devices are shown in Fig. 7. When the task number is smaller than 40, the energy consumption of GA is higher than PSO. However, when it is larger than 40, the energy consumption of GA is always lower. So, in general, GA is better in energy optimisation. Fig. 8 shows the results on task execution cost. The execution cost of GA algorithm is always lower. With the increase of the number of tasks, the performance gap between GA and PSO is growing as well. This shows that GA is the better one for cost optimisation.

In summary, the experimental results demonstrate the effectiveness of FogWorkflowSim, that can serve as a useful experimental platform for researchers in Fog based workflow systems as well as practitioners interested in adopting Fog Computing and workflow systems for their new software projects.

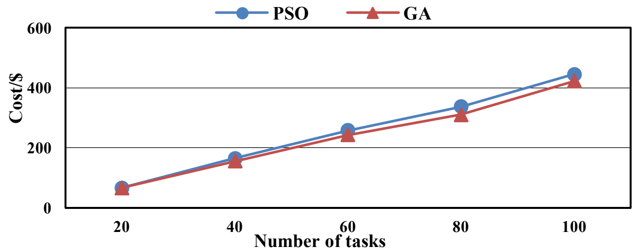

Fig. 8. Comparison of the task execution cost

VI. Conclusions and Future Work

In this paper, we have demonstrated automated simulation results with different workflow scheduling algorithms (such as MinMin, MaxMin, FCFS, Round Robin, PSO and GA) for different performance metrics (such as time, energy and cost). Our results demonstrate that FogWorkflowSim is an effective automated simulation toolkit for the simulation and performance evaluation of Fog workflows. In the future, we are planning to extend FogWorkflowSim to support for task scheduling by taking into the consideration of tasks with different priorities and the mobility of End Devices.

References

[1] S. Li, L. D. Xu and S. Zhao, "5G Internet of Things: A Survey," *Journal of Industrial Information Integration*. vol. 10, 2018.

[2] T. Chen, Q. Ling, Y. Shen and G. B. Giannakis, "Heterogeneous Online Learning for "Thing-Adaptive" Fog Computing in IoT," *IEEE Internet of Things Journal*. vol. 5, 2018.

[3] A. A. Mutlag, M. K. A. Ghani, N. Arunkumar, M. A. Mohamed and O. Mohd, "Enabling Technologies for Fog Computing in Healthcare IoT Systems," *Future Generation Computer Systems*. vol. 90, 2019.

[4] H. Zhang, Y. Zhang, Y. Gu, D. Niyato and Z. Han, "A Hierarchical Game Framework for Resource Management in Fog Computing," *IEEE Communications Magazine*. vol. 55, 2017.

[5] J. Xu, X. Li, R. Ding and X. Liu, "Energy Efficient Multi-Resource Computation Offloading Strategy in Mobile Edge Computing," Computer Integrated Manufacturing Systems. vol. 25, 2019.

[6] X. Liu, Y. Yang, D. Cao and D. Yuan, "Selecting Checkpoints along the Time Line: a Novel Temporal Checkpoint Selection Strategy for Monitoring a Batch of Parallel Business Processes," *Proc. International Conference on Software Engineering*, 2013.

[7] N. A. Moketar, M. Kamalrudin, S. Sidek, M. Robinson and J. Grundy, "An automated collaborative requirements engineering tool for better validation of requirements," *Proc. 31st International Conference on Automated Software Engineering (ASE)*, 2016.

[8] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software: Practice and Experience*. vol. 41, 2011.

[9] W. Chen and E. Deelman, "WorkflowSim: A Toolkit for Simulating Scientific Workflows in Distributed Environments," *Proc. IEEE International Conference on e-Science*, 2012.

[10] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh and R. Buyya, "iFogSim: A toolkit for Modeling and Simulation of Resource Management Techniques in the Internet of Things, Edge and Fog Computing Environments," *Software: Practice and Experience*. vol. 47, 2017.

[11] G. Juve. "WorkflowGenerator." [Online]. Available: https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator.

[12] S. Cao, X. Tao, Y. Hou and Q. Cui, "An Energy-Optimal Offloading Algorithm of Mobile Computing Based on HetNets," *Proc. International Conference on Connected Vehicles and Expo*, 2015.