Chapter 11
Fuzzy-Based Workflow Scheduling in Multi-Cloud Environment

J. Angela Jennifa Sujana, R. Venitta Raj, and T. Revathi

11.1 Introduction

Nowadays with the introduction of many Internet of Things (IoT)-based real time applications, the need for integrating those applications with the cloud environment has become essential. Also, there is a great need for accessing the data from IoT-based smart applications for doing meaning analysis for inferring useful information and automation of the process. At the same time, the user may want to use different cloud services for their application. In such situations, multi-cloud environment is the only solution. Hence, with multi-cloud environment the user or the developer is having enormous freedom in choosing the best cloud service provider from a set of cloud providers.

Multi-cloud computing aggregates large pool of resources and share them among vast cloud users. It is a gifted technique for systems integration. Multi-cloud computing can also be defined as a new version of collaborative environment, in which scalable and virtualized resources are provided as a service over the Internet [1]. In general, the service workflow is organized as the collection of services to ease the requirements and automation of large-scale distributed systems [2]. Although most commercial cloud services are operated and owned by distributed and heterogeneous organizations in multi-cloud computing environments, the inherent uncertainty and unreliability of large-scale organizations often pose threats to the operation of workflow applications. Hence, in addition to execution time and cost factors, we need to think about trust factor.

J. A. J. Sujana (✉) · R. V. Raj · T. Revathi
Department of Information Technology, Mepco Schlenk Engineering College,
Sivakasi, Tamil Nadu, India
e-mail: ang_jenefa@mepcoeng.ac.in; venittaraj@mepcoeng.ac.in; trevathi@mepcoeng.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_11

Initially this multi-cloud environment is not supported for the cloud users, because of the vendor lock-in problem [3]. The Open Virtualization Format (OVF) [4, 5] is a vendor-independent format which supports portability and deployment with different vendors. With the collaborative effort of Dell, Microsoft, HP, IBM, VMWare, and XenSource, the system images can be imported and deployed on multiple platforms, thus enabling cross-platform portability. The introduction of this platform-independent virtualization format is one reason for the emergence of multi-cloud environment. Also, users and developers started to use multi-cloud based on their optimal quality criteria satisfaction.

In many of the applications which use the cloud services, the Quality of Service (QoS) plays a vital role. The QoS mainly depends on the user request and the application being used. It can be noted many enterprise applications and IoT-based smart applications can be represented as workflows [6]. Rajganesh et al. [7] have proposed a service context-aware cloud broker which uses the service details from the contextual information of cloud services and computes service similarities on the basis of QoS values. Each task in the workflow represents a module in the enterprise application. Normally these workflows are represented as Directed Acyclic Graph (DAG). Each node in the DAG represents the task in the workflow. Fuzzy decision-based models can be adopted for multi-cloud environments [8, 9]. A fuzzy logic-based intelligent cloud broker is proposed to find the imprecise state of the inexperienced cloud user when deciding the infrastructure service requirements [10]. An intelligent cloud broker which uses the MapReduce framework for the effective pre-processing of cloud users' feedback was proposed by Rajganesh Nagarajan et al. [11]. Also, they propose a fuzzy logic-based trust evaluation system for accepting the user's feedback in terms of fuzzy linguistic [12].

This chapter focuses on finding the suitable cloud service provider based on the Quality of Service of the user or developer of enterprise applications and IoT-based smart applications. The main QoS parameters considered are time, cost, and trust. This multi-objective QoS-based workflow scheduling is based on fuzzy model. The fuzzy membership function is defined for each objective, and collectively the decisions are made. Also, hence this chapter proposes a multi-objective QoS-based workflow scheduling in multi-cloud environment with fuzzy logic with the fuzzy logic-based workflow scheduling (FLWS) algorithm.

11.2 System Architecture

The system architecture of the proposed fuzzy-based workflow scheduling in multi-cloud environment is represented in Fig. 11.1. This chapter envisages a fuzzy-based multi-objective model for dispatching the tasks in the workflow to suitable cloud provider in accord with the user Quality of Service (QoS). The user will submit the workflow, which will be handled by the QoS-based Resource Management layer. This module gets the workflow from the user and the QoS weightage for each objective. The objectives addressed are time, cost, and trust. This QoS-based Resource Management module can also be viewed as an agent, helping in the user for

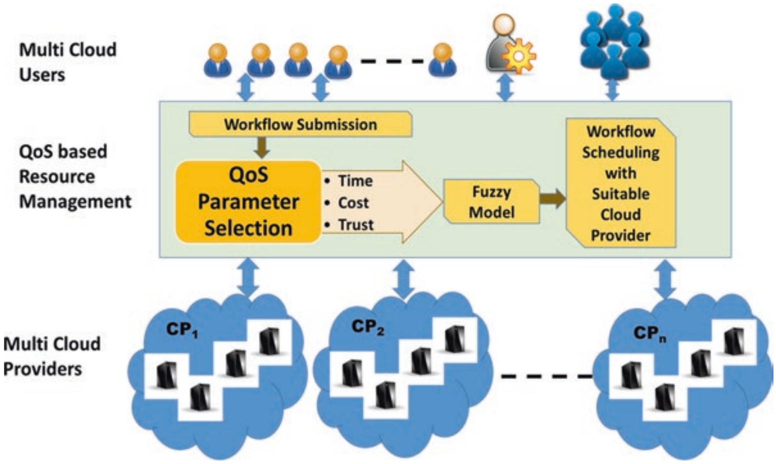

Fig. 11.1 System architecture

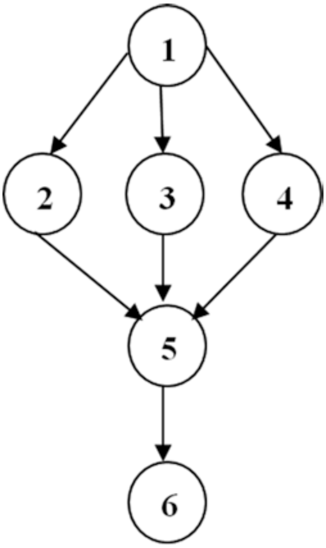

Fig. 11.2 Sample workflow

identifying the suitable services for them. The fuzzy model is used to make the decision in selecting the suitable cloud provider for the workflow. Here we provide a fine-grained selection module. That is, the user can select different services for each task in the workflow from different cloud providers. For each task the best suitable cloud provider's service will be considered, and it will be scheduled accordingly.

Figure 11.2 represents a sample workflow, which can be represented as the Directed Acyclic Graph (DAG) [13]. The sample workflow consists of six tasks represented by the nodes, and the edges between the tasks represent the dependency among the tasks. The edges represent the data transfer time. If both the ancestor and descendant task get executed in the same virtual machine, then the data transfer time between these tasks becomes zero.

11.3 Multi-objective Fuzzy Decision Model

In this chapter, we introduce fuzzy-based decision model for selecting the suitable cloud provider in multi-cloud. The clouds' providers considered in the multi-cloud environment are denoted as CP = $\{cp_1, cp_2, \ldots, cp_n\}$. The workflow is modelled as a directed acyclic graph, W = (TK, E), where TK denotes the tasks' set TK = $\{tk_1, tk_2, \ldots, tk_m\}$ and each task $tk_i \in$ TK, $1 \leq i \leq m$ needs a cloud service from the cloud provider. Similarly E represents the set of communication link edges between tasks, and each edge $e(i,j) \in E$ represents that the task tk_j depends on the task tk_i, i.e., task tk_i should be executed prior to the execution of the task tk_j. The task with an in-degree 0 is designated as tk_{entry} entry task, and the task without-degree 0 is designated as tk_{exit} exit task. The parent tasks of the task tk_i are represented *ancestor*(tk_i), and the child tasks of a task tk_i are represented by *descendant*(tk_j). In multi-cloud environment the provider has enormous services. The user pays for the services.

This aids in selecting the best optimal cloud service provider for the tasks in the workflow according to the user's QoS demand. The fuzzy decision model will consider three objectives, namely, time, cost, and trust.

Let n be the total no. of candidate cloud providers in multi-cloud. There are a set of S_j^i services, which are available for each task in the workflow. More specifically S_j^i denotes the service from cloud provider cp_j, delivered for task tk_i. Hence for the same task tk_i there will be "m" number of services available from the cloud providers CP. These services may have varied processing capabilities, processing time, and prices. Let t_j^i be the total processing time and C_j^i be the total processing cost for the ith task on the jth service.

To get optimized solution on multi-objective, we propose fuzzy model as follows. Using max-min as the operator [14], the membership function of objectives can be formulated by separating each objective into its maximum values and minimum values. The membership function for time and cost objectives is given by Eq. (11.1) and for trust is given by Eq. (11.2).

$$u(x) = \begin{cases} 1, Z_K \leq Z_K^{\min} \\ \dfrac{Z_K^{\max} - Z_K(x)}{Z_K^{\max} - Z_K^{\min}}, Z_K^{\min} \leq Z_K \leq Z_K^{\max} \\ 0, Z_k^{\max} \leq Z_k \end{cases} \tag{11.1}$$

$$u(x) = \begin{cases} 0, Z_l \leq Z_l^{\min} \\ \dfrac{Z_l(x) - Z_l^{\min}}{Z_l^{\max} - Z_l^{\min}}, Z_l^{\min} \leq Z_l \leq Z_l^{\max} \\ 1, Z_l^{\max} \leq Z_l \end{cases} \tag{11.2}$$

Here the workflow scheduling problem is considered as a multi-objective planning problem which focuses in optimizing the conflicting objectives [15]. The following notations are used to represent the objectives. The deadline of a task is denoted by D, budget that can be spent by B, and trust level by Tr. The time taken by jth service assigned for ith task is denoted by t_j^i. Let total number of tasks be n and total number of cloud provider services available for the ith task be $m_{i.}$ To denote assigning the ith task to jth service is given by y_j^i; if the jth service is assigned for ith task, then $y = 1$; else $y = 0$.

The objectives are thus described as follows in Eq. (11.3a), (11.3b), and (11.3c).

$$Minimize: Z_1 = \sum_{i=1}^{n}\sum_{j=1}^{m_i} y_j^i c_j^i \tag{11.3a}$$

$$Minimize: Z_2 = \sum_{i=1}^{n}\sum_{j=1}^{m_i} y_j^i t_j^i \tag{11.3b}$$

$$Maximize: Z_3 = \sum_{i=1}^{n}\sum_{j=1}^{m_i} y_j^i tr_j^i \tag{11.3c}$$

$$\text{Subject to}: Z_1 \;''\; B, Z_2 \;''\; D, Z_3 \;''\; Tr$$

where Z_1 and Z_2 represent the minimization objective for time and cost and Z_3 represents the maximization objective for trust. Our proposed fuzzy logic-based workflow scheduling (FLWS) model considers three different objectives related to time, cost, and trust constraint [10].

11.3.1 Membership Function for Trust Evaluation

The general trust metric, which is the combination of direct trust (DT) and recommendation trust (RT), can be defined as given in Eq. (11.4),

$$Tr(S_i) = w_i * \mathrm{DT}(S_i) + (1 - w_i) * \mathrm{RT}(S_i) \tag{11.4}$$

where the weight w_i assigned for DT is calculated by

$$w_i = 1 - \frac{1}{e^k} \tag{11.5}$$

where k is the number of times the ith service is used by the client user. The direct trust (DT) is calculated by Eq. (11.6).

$$\mathrm{DT}(S_i) = \frac{n_i + 1}{N_i + 2} \tag{11.6}$$

where DT(S_i) is the direct trust of the ith service which the active user experiences based on the history and RT(S_i) is the recommendation trust of ith service by other users. w_i is the weight of the direct trust and recommendation trust which is calculated as in Eq. (11.5).

Recommendation trust value can be calculated as given in Eq. (11.7). Here we consider the rating given by other users for a cloud provider's service. The user can record a rating in the 1–5 scale. Value 1 for poor service and value 5 for the best service from the cloud provider. A sample rating table with five users for four cloud providers is shown in Table 11.1. The missing values are represented by ? and it is calculated by the recommendation trust. The recommendation trust is calculated as the weighted sum of the user's rating for the service.

$$\mathrm{RT}\left(S_i\right) = avg\left(v_a\right) + \frac{\sum_{i=1}^{n} w_{ai}\left(v_{ij} - \overline{v_i}\right)}{\sum_{i=1}^{n}\left|w_{ai}\right|} \tag{11.7}$$

$avg(v_i)$ can be calculated as follows:

$$avg\left(v_i\right) = \frac{1}{\left|S_i\right|} * \sum_{j \epsilon S_i} v_{ij} \tag{11.8}$$

$avg(v_i)$ be the average rating given by user i. v_{ij} is the rating by user i to the jth cloud provider's service. We use the *Pearson Correlation Coefficient (PCC)* for calculating the similarity between user a and i. This can be calculated as using the Eq. (11.9).

$$w_{ai} = \frac{\sum_{j \in S}\left(v_{aj} - \overline{v_a}\right)\left(v_{ij} - \overline{v_i}\right)}{\sqrt{\sum_{j \in S}\left(v_{aj} - \overline{v_a}\right)^2\left(v_{ij} - \overline{v_i}\right)^2}} \tag{11.9}$$

Membership function for trust calculation is formulated as given in Eq. (11.10).

$$U_{tk_i}\left(tr_x\right) = \begin{cases} \dfrac{tr_x^i - tr_{\min}^i}{tr_{\max}^i - tr_{\min}^i}, & if\ tr_{\min}^i \le tr_x^i \le tr_{\max}^i \\ 0, & if\ tr_{\min}^i = tr_x^i \\ 1, & if\ tr_{\max}^i = tr_x^i \end{cases} \tag{11.10}$$

Table 11.1 Sample ratings by the users for the cloud services

	Us_1	Us_2	Us_3	Us_4	Us_5
CP_1	?	2	3	4	5
CP_2	3	4	5	3	?
CP_3	1	5	?	2	2
CP_4	5	1	2	5	5

11.3.2 Membership Function for Execution Time

Using max-min as the operator membership function for execution time is calculated as given in Eq. (11.11).

$$U_{tk_i}\left(t_x\right)=\begin{cases}\dfrac{t_{\max}^{i}-t_{x}^{i}}{t_{\max}^{i}-t_{\min}^{i}}, & if\, t_{\min}^{i}\le t_{x}^{i}\le t_{\max}^{i}\\ 0, & if\, t_{\max}^{i}=t_{x}^{i}\\ 1, & if\, t_{\min}^{i}=t_{x}^{i}\end{cases}\tag{11.11}$$

11.3.3 Membership Function for Execution Cost

Similarly, the membership function for cost the user has to pay for the service is calculated as given in Eq. (11.12).

$$U_{tk_i}\left(c_x\right)=\begin{cases}\dfrac{c_{\max}^{i}-c_{x}^{i}}{c_{\max}^{i}-c_{\min}^{i}}, & if\, c_{\min}^{i}\le c_{x}^{i}\le c_{\max}^{i}\\ 0, & if\, c_{\max}^{i}=c_{x}^{i}\\ 1, & if\, c_{\min}^{i}=c_{x}^{i}\end{cases}\tag{11.12}$$

11.3.3.1 Fuzzy Decision

To convert to crisp model from the fuzzy model, we use max-min as the operator as follows:

$$Maximize \quad \lambda_i$$

$$U_{tk_i}\left(t_x\right)\ge\lambda_i, U_{tk_i}\left(c_x\right)\ge\lambda_i, U_{tk_i}\left(tr_x\right)\ge\lambda_i \tag{11.13}$$

The overall satisfaction degree can be defined as the minimum of overall satisfaction of the given membership values for the trust, time, and cost.

The overall satisfaction by the fuzzy decision is given in Eq. (11.14).

$$\lambda_j^i=\min\left\{U_{tk_i}\left(t_x\right), U_{tk_i}\left(c_x\right), U_{tk_i}\left(tr_x\right)\right\} \tag{11.14}$$

The selection of cloud provider’s service for task is given by the maximum of all degrees of satisfaction, as defined in Eq. (11.15),

$$\lambda_k^i = \max\left\{\lambda_j^i\right\} \tag{11.15}$$

However, the user may want to specify their own preference among the objectives time, cost, and trust. This can be done by adding the weight factor for each objectives time, cost, and trust. Thus, we introduce a weight factor for objectives time, cost, and trust. They are represented as w_t, w_c, and w_{tr},respectively. The final selection of the cloud providers' service is done by the weighted arithmetic mean operator and given in Eq. (11.16).

$$Maximize U * W' \tag{11.16}$$

where W is the weight factor for objectives time, cost, and trust. $W = \{w_t, w_c, w_{tr}\}$. The sum of the weight factor must be 1. i.e. $w_t + w_c + w_{tr} = 1$ and the weight factor values must be in the interval [0,1].

11.4 Schedule Primitives

To schedule the workflow with minimum execution time, we use the Earliest Completion Time (ECT) of a task with a cloud provider as the heuristic technique. In this regard the following primitives are considered.

The execution time of the task tk_i on the jth CP is termed as $\mathrm{ET}(tk_i, cp_j)$, and it is calculated using Eq. (11.17). The computation capacity of the service S_j in cloud provider cp_j is denoted as $x(cp_j)$. The execution time of a task tk_i is denoted by $x(tk_j)$.

$$\mathrm{ET}\left(tk_i, cp_j\right) = \frac{x\left(tk_i\right)}{x\left(cp_j\right)} \tag{11.17}$$

The average execution time of the task tk_i is given in Eq. (11.18).

$$\mathrm{ET}\left(tk_i\right) = \frac{\sum_{j=1}^{m} \mathrm{ET}\left(tk_i, cp_j\right)}{m} \tag{11.18}$$

The communication cost of an edge $e(i,j)$ can be calculated as given in the Eq. (11.19). Since the services are available from different cloud providers, there will be some latency in hosting the workflow tasks' in the virtual machines of the cloud providers. The late(cp_j) denotes the latency that a cloud provider will have for executing any task. The $x(e(i,k))$ is the amount of data transferred between the two tasks i and k. bw_k is the bandwidth of the link between the cloud providers.

$$\mathrm{CC}\left(tk_i, tk_j\right) = \sum_{j=1}^{m} \mathrm{late}\left(cp_j\right) + \frac{x\left(e\left(i,j\right)\right)}{\sum_{k=1}^{n} \mathrm{bw}_k} \tag{11.19}$$

The Earliest Start Time (EST) is the earliest possible start time of a task on a cloud provider cp_j. This can be computed by the Eq. (11.20)

$$EST\left(tk_i, cp_j\right) = max\{ReadyTime\left(tk_i, cp_j\right), \\ \max_{t_m \in \mathrm{ancestor}(t_i)} \left\{ECT\left(tk_m, cp_j\right) + CC\left(tk_m, tk_i\right)\right\} \tag{11.20}$$

$$\mathrm{ReadyTime}\left(tk_{\mathrm{entry}}, cp_j\right) = 0$$

The ReadyTime(tk_i, cp_j) denotes the time in which the jth VM is ready for executing the task tk_i. The same cloud provider cp_j can be used to execute multiple tasks, provided it has completed its earlier task and ready for executing the next task and the maximum satisfaction level. The ready time of the entry task is considered as 0. The Earliest Completion Time of a task is computed by considering the earliest start time of a task and the execution of the task on a CP. The EST of a task is computed by finding the maximum of the ready time of a CP for the task and the earliest end time of the entire ancestral task along with the data transfer time (Communication Cost). If any two tasks are allotted with the same cloud provider CP, then the communication cost of those two tasks CC(tk_m, tk_i) = 0.

The Earliest Completion Time (ECT) of a task is the earliest possible completion time of a workflow task on a cloud provider. This is computed by the Eq. (11.21)

$$\mathrm{ECT}\left(tk_i, cp_j\right) = \mathrm{EST}\left(tk_i, cp_j\right) + \frac{x\left(tk_i\right)}{x\left(cp_j\right)} \tag{11.21}$$

Makespan is the one representing the schedule length, and it is found by considering the ECT of exit task and it is denoted in Eq. (11.10). The objective of the schedule is to minimize this makespan.

$$\mathrm{makespan} = \mathrm{ECT}\left(tk_{\mathrm{exit}}, cp_j\right) \tag{11.22}$$

11.5 Fuzzy Logic-Based Workflow Scheduling

The fuzzy logic-based workflow scheduling (FLWS) model is given in the Algorithm.

Algorithm: The FLWS Algorithm

```
Algorithm: The FLWS Algorithm.
Input: Input: W = (TK, E)  //W - Workflow DAG ,
    Set of Cloud Providers- CP,
    QoS Parameters – (time, cost, and trust) values as tuple
Output: A Schedule $Sch$ with workflow tasks $tk_i \in TK$ mapped to Cloud Provider $cp_j \in CP$
$Sch = \emptyset$ // $Sch$ is the schedule
Calculate $ECT(tk_i, cp_j)$ using equation (21).
Calculate $U_{tk_i}(tr_x), U_{tk_i}(t_x)$ and $U_{tk_i}(c_x)$ using equation (10), (11) and (12) respectively
$TaskList \leftarrow tk_{entry}$
While $TaskList \neq \emptyset$ do
    for all available Cloud Provider $cp_j$ providing service for task $tk_i$ do
        Find $ECT(tk_i, cp_j)$value using insertion-based scheduling policy
        Compute $\lambda_k = \max\{\sum_{j=0}^{2} U_{Z_{j(x)}}^{T_i} * w_j\}$
    end for
    $Sch = Sch \cup Map(tk_i, S_k)$
    Update TaskList with next level of tasks from W
end while
Return $Sch$
```

The algorithm fuzzy logic-based workflow scheduling (FLWS) gets the Workflow W as DAG, the set of Cloud Providers CP that are readily providing services for the tasks in the workflow, and the QoS Parameters values as integer for time, cost, and trust as tuple. Initially the schedule **Sch** is initialized as NULL. Calculate Earliest Completion Time ECT(tk_i, cp_j) for all the tasks in each Cloud Provider providing the service. The QoS Parameters values for (time, cost, and trust) are got as tuple for each service in Cloud Providers. With these values $U_{tk_i}\left(tr_x\right), U_{tk_i}\left(t_x\right)$ and $U_{tk_i}\left(c_x\right)$ can be calculated. Then for generating the schedule initially the first task is considered. Then with the loops at step 5 and 6, the decision-making process is repeated for all the tasks. The ECT value is computed with insertion-based scheduling policy. Step 8 does the calculation for making the fuzzy decision for finding the cloud service best suited for the task as per the QoS requirement of the user. The mapped service to the task Map(tk_i, S_k) is appended to the schedule Sch. These steps are repeated until all the tasks are done with. Finally, the schedule Sch is given as the output.

11.6 Results and Discussions

To illustrate the working of our fuzzy logic-based workflow scheduling (FLWS) model, we present a table with sample values for the six tasks in workflow shown in Fig. 11.2. Table 11.2 shows the sample QoS Parameters (time, cost, and trust) values as tuple for four Cloud Providers. The steps are worked out for the first task in the workflow.

Step 1

Using max-min operator, compute vectors of $(U_{T_i}(t_x), U_{T_i}(c_x), U_{T_i}(tr_x))$ for the first task. Here $t^1_{\max} = 4, t^1_{\min}=3$; $c^1_{\max}=5$, $c^1_{\min}=2$; $tr^1_{\max}=5$, $tr^1_{\min}=2$. For this task three cloud providers are providing their services. Hence the calculations are as shown below.

$$U^1_{tk_1} - \left(\frac{4-4}{4-3} \quad \frac{5-5}{5-2} \quad \frac{3-2}{5-2}\right) = \left(0 \quad 0 \quad 1/3\right)$$

$$U^2_{tk_1} - \left(\frac{4-3}{4-3} \quad \frac{5-2}{5-2} \quad \frac{5-2}{5-2}\right) = \left(1 \quad 1 \quad 1\right)$$

$$U^3_{tk_1} - \left(\frac{4-3}{4-3} \quad \frac{5-4}{5-2} \quad \frac{2-2}{5-2}\right) = \left(1 \quad 1/3 \quad 0\right)$$

Table 11.2 Sample QoS parameter values from multi-cloud providers for their services

	(Time, cost, and trust) values by cloud providers			
Workflow task no.	CP_1	CP_2	CP_3	CP_4
1	(4, 5, 3)	(3, 2, 5)	(3, 4, 2)	–
2	(3, 2, 1)	–	(3, 2, 5)	(2, 1, 1)
3	(4, 7, 3)	(5, 6, 4)	(6, 5, 5)	(7, 4, 4)
4	(2, 3, 6)	(3, 4, 1)	–	(6, 3, 4)
5	(5, 2, 1)	–	(4, 6, 5)	(6, 2, 7)
6	–	(4, 3, 2)	(3, 5, 4)	(5, 2, 2)

Step 2
The selection of cloud provider's service for task is computed by the maximum value of all three objectives based on weight factor. Maximize $U * W'$. As given in step 8 (Algorithm FLWS) do the calculations. Simply this can be denoted as $(U_{T_i}(t_x)\ U_{T_i}(c_x)\ U_{T_i}(tr_x))*\ (w_t w_c w_{tr})'$. Here four cases are presented. In case 1 all the three objectives are given the same weightage. In case 2, time is given the priority and the other two factors are not considered. Similarly, in case 3, cost is given the priority and the other two factors are not considered. Also, in case 4, trust is given the priority and the other two factors are not considered.

Case 1
Consider $W = (1/3\ 1/3\ 1/3)$ to the set $W = (w_t w_c w_{tr})$ for test.

$$\begin{pmatrix} U^1_{tk_1} \\ U^2_{tk_1} \\ U^3_{tk_1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1/3 \\ 1 & 1 & 1 \\ 1 & 1/3 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} = \begin{pmatrix} 1/9 \\ 1 \\ 4/9 \end{pmatrix}$$

As per the results got, it could be understood that the second cloud provider's service **CP$_2$** is the best service for the first task considering equal weightage for all the three objectives time, cost, and trust.

Case 2
Let $W = (1, 0, 0,)$,

$$\begin{pmatrix} U^1_{tk_1} \\ U^2_{tk_1} \\ U^3_{tk_1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1/3 \\ 1 & 1 & 1 \\ 1 & 1/3 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

As per the results, it is noted that both second and third cloud provider's service **CP$_2$ and CP$_3$** are the best service for the first task considering time alone among the three objectives time, cost, and trust.

Case 3
Let $W = (0, 1, 0,)$,

$$\begin{pmatrix} U^1_{tk_1} \\ U^2_{tk_1} \\ U^3_{tk_1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1/3 \\ 1 & 1 & 1 \\ 1 & 1/3 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1/3 \end{pmatrix}$$

When cost alone is considered among the three objectives, it is noted that cloud provider's service **CP$_2$** is the best service for the first task.

Case 4

Let $W = (0\ 0\ 1)$,

$$\begin{pmatrix} U_{tk_1}^1 \\ U_{tk_1}^2 \\ U_{tk_1}^3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1/3 \\ 1 & 1 & 1 \\ 1 & 1/3 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1 \\ 0 \end{pmatrix}$$

When trust alone is considered among the three objectives, it is noted that cloud provider's service $\mathbf{CP_2}$ is the best service for the first task.

The performance graph is also shown below. The algorithm is implemented in Cloudsim. In the simulation experiments, graphs are generated using DagGen library. DagGen is a synthetic task graph generator tool. This tool generates random and synthetic task graphs for the purpose of simulation. This is useful, for instance, to evaluate scheduling algorithms that must be tested over a wide range of application configurations. The comparison of FLWS, Minimum Critical Path (MCP), and greedy-cost models is presented by using the DAG generated. The experiments are done to sequentially test the three algorithms with different sizes. The number of tasks varies from 10 to 100 with an increment of 20. The comparisons of execution time and cost of FLWS, MCP, and greedy cost are shown in Figs. 11.3 and 11.4, respectively. As for the FLWS algorithm, time and cost are both considered simultaneously, which enables the user to compromise requirements to yield a genuinely better solution. MCP is good with execution time among the algorithms but with the highest cost. Conversely, greedy cost has the least cost but takes the longest execution time. As for the FLWS approach, time and cost are both considered simultaneously, which enables the user to get a better solution with their QoS.

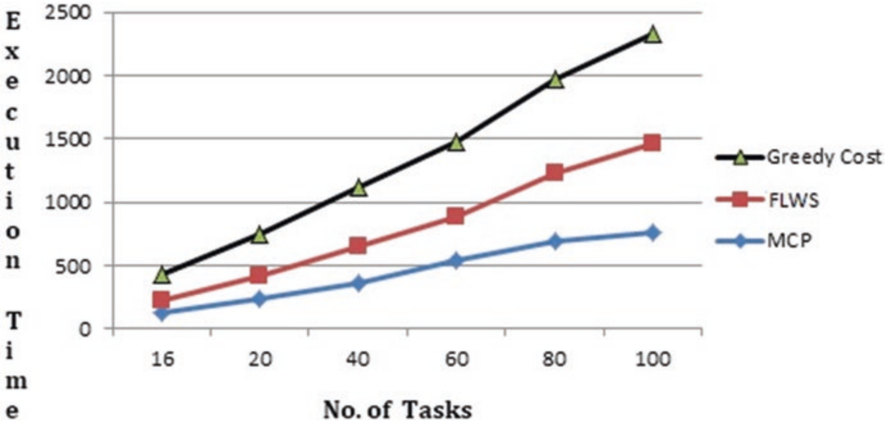

Fig. 11.3 Comparison of execution time

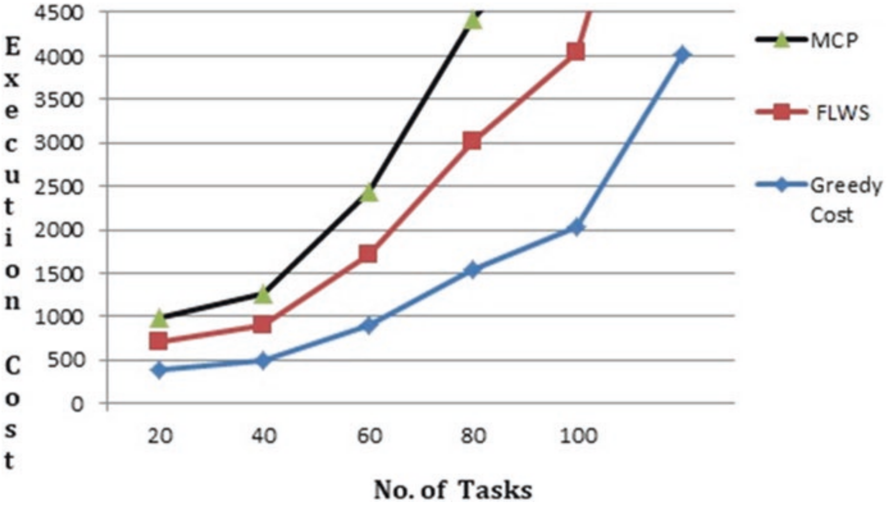

Fig. 11.4 Comparison of execution cost

11.7 Conclusion

This chapter discusses the problems related to workflow scheduling (WFS) in multi-cloud computing environment. A fuzzy logic-based decision for solving the multi-objective problem is proposed. The fuzzy logic-based workflow scheduling (FLWS) provides an optimal schedule for the given workflow with user QoS being satisfied in terms of time, cost, and trust. Thus, this FLWS algorithm provides cooperative model for workflow scheduling in multi-cloud environments along with the user quality of service being achieved.

References

1. Diaz-Montes, J., Diaz-Granados, M., Zou, M., Tao, S., & Parashar, M. (2018). Supporting data-intensive workflows in software-defined federated multi-clouds. *IEEE Transactions on Cloud Computing, 6*, 250–263. https://doi.org/10.1109/TCC.2015.2481410
2. Ranaldo, N., & Zimeo, E. (2009). Time and cost-driven scheduling of data parallel tasks in grid workflows. *IEEE System Journal, 3*, 104–120.
3. Buyya, R., Christian, V., & Tamarai Selvi, S. (2013). Mastering cloud computing. TMGH.
4. Crosby, S., et al. (2010). Open virtualization format specification. Distributed management task force. [online document] Retrieved from https://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf
5. Hwang, K., Fox, G. C., & Dongarra, J. G. (2012). *Distributed and cloud computing, from parallel processing to the internet of Things*. Morgan Kaufmann Publishers.
6. Tan, W. A., Sun, Y., Li, L. X., Lu, G. Z., & Wang, T. (2014). A trust service-oriented scheduling model for workflow applications in cloud computing. *IEEE Systems Journal, 8*, 868–878. https://doi.org/10.1109/JSYST.2013.2260072

7. Nagarajan, R., & Thirunavukarasu, R. (2020). A service context-aware QoS prediction and recommendation of cloud infrastructure services. *Arabian Journal for Science and Engineering, 45*, 2929–2943. https://doi.org/10.1007/s13369-019-04218-6
8. Farid, M., Latip, R., Hussin, M., & Abdul Hamid, N. A. W. (2020). Scheduling scientific workflow using multi-objective algorithm with fuzzy resource utilization in multi-cloud environment. *IEEE Access, 8*, 24309–24322. https://doi.org/10.1109/ACCESS.2020.2970475
9. Sujana, J. A. J., Revathi, T., & Rajanayagam, S. J. (2020). Fuzzy based security-driven optimistic scheduling of scientific workflows in cloud computing. *IETE Journal of Research, 66*(2), 224–241. https://doi.org/10.1080/03772063.2018.1486740
10. Nagarajan, R., & Thirunavukarasu, R. (2019). A fuzzy-based decision-making broker for effective identification and selection of cloud infrastructure services. *Soft Computing, 23*, 9669–9683.
11. Nagarajan, R., Thirunavukarasu, R., & Shanmugam, S. (2018). A fuzzy-based intelligent cloud broker with MapReduce framework to evaluate the trust level of cloud services using customer feedback. *International Journal of Fuzzy Systems, 20*, 339–347. https://doi.org/10.1007/s40815-017-0347-5
12. Nagarajan, R., Kumaran, S. S., & Thirunavukarasu, R. (2017). A fuzzy logic based trust evaluation model for the selection of cloud services. IEEE international conference on computer communication and informatics (ICCCI), Coimbatore. https://doi.org/10.1109/ICCCI.2017.8117686.
13. Sujana, J. A. J., Revathi, T., & Malarvizhili, M. (2015). Scheduling of scientific workflows in cloud with replication. *Applied Mathematical Sciences, 9*, 2273–2280.
14. Zadeh, L. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.
15. Xu, L. (1988). A fuzzy multi-objective programming algorithm in decision support systems. *Annals of Operations Research, 12*, 315–320.