

Optimal Cost for Time-Aware Cloud Resource Allocation in Business Process

Rania Ben Halima, Slim Kallel, Walid Gaaloul, Mohamed Jmaiel

▶ **To cite this version:**

Rania Ben Halima, Slim Kallel, Walid Gaaloul, Mohamed Jmaiel. Optimal Cost for Time-Aware Cloud Resource Allocation in Business Process. 2017. hal-01525209

HAL Id: hal-01525209
https://hal.science/hal-01525209

Preprint submitted on 19 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal Cost for Time-Aware Cloud Resource Allocation in Business Process

Rania Ben Halima*†, Slim Kallel†, Walid Gaaloul*, Mohamed Jmaiel†‡
*Telecom SudParis, UMR 5157 Samovar, Univ. of ParisSaclay, France
Email: {rania.ben_halima,walid.gaaloul}@telecom-sudparis.eu
†ReDCAD Laboratory, University of Sfax, Tunisia
Email: {rania.benhalima,slim.kallel,mohamed.jmaiel}@redcad.tn
‡Digital Research Center of Sfax, Tunisia
Email:mohamed.jmaiel@enis.rnu.tn

***Abstract*—Cloud Computing infrastructures are being increasingly used for running business process activities due to its high performance level and low operating cost. The enterprise QoS requirements are diverse and different resources are offered by Cloud providers in various QoS-based pricing strategies. Furthermore, business process activities are constrained by hard timing constraints and if they are not executed correctly the enterprise will pay penalties costs. Therefore, finding the optimal Cloud resources allocation for a business process becomes a highly challenging problem. While optimizing the Cloud resource allocation cost, it is important to respect activities QoS requirements and temporal constraints and Cloud pricing strategies constraints. The aim of the present paper is to offer a method that assists users finding the optimal pricing strategy for Cloud resource used by business process activities. Basically, we use a binary/(0-1) linear program with an objective function under a set of constraints. In order to show its feasibility, our approach has been implemented and the results of our experiments highlight the effectiveness of our proposed solution.**

Keywords. Business Process, Cloud Computing, Optimization, Pricing Strategy, Time

I. Introduction

Cloud Computing has recently emerged as a new computing paradigm in many application areas comprising office and enterprise systems [1]. It offers a scalable and on-demand access to various types of VM with different amount of CPU cores, memory and disk at different prices. That is why Cloud Computing is being an attractive infrastructure used by enterprise to deploy their business processes.

With the purpose of increasing their profits and attracting more clients, Cloud providers propose different pricing strategies. For instance, Amazon offers three pricing models: on-demand, reserved and spot instance which are charged variously. The cost of each pricing model depends on some parameters such as time, region, etc. In addition, business process activities are constrained by hard timing constraints.

While there exist works on resource allocation and management in the BPM context [2], [3], [4], only few authors have paid attention to process scheduling in Cloud [5], [6]. In addition, one key perspective when dealing with Business Process Management (BPM) is time [7]. However, to the best of our knowledge, optimizing business process cost enriched with temporal constraints and activities' penalty costs while combining various pricing strategies is not yet handled.

The diversity of the pricing models offered by Cloud providers and the business process activities requirements (temporal, QoS, penalty cost) make the optimization issue more complex. For that, in this work, we propose a new approach for discovering an optimal pricing strategy for time-aware Cloud resource allocation that has the lowest price. Concretely, we propose to find from a set of pricing models, the strategies that guarantee to the enterprise to have the less expenditure without violating temporal constraints of both: activities and Cloud provider pricing models.

The remainder of this paper is organized as follows. In Section II we present definitions related to business process using different Cloud resource pricing strategies. Section III motivates the problem with a real use case from France Telecom Orange labs. Afterwards, in section IV we describe our approach for an optimal pricing strategy for resource allocation in business process. In Section V, the experimental evaluation results are analysed. A review of the related work is given in section VI. Finally, our conclusions and future work are presented.

II. Basic Concepts

In this section, we present the main concepts and definitions related to Cloud resource pricing strategies, the process models and the Cloud resource allocation.

A. Cloud resource pricing strategies

In Cloud Computing, there are various pricing models under which instances can be acquired. Each Cloud provider proposes *on-demand* instances that can be purchased at a fixed cost per hour. This instance type is offered by all the Cloud providers (Amazon, Microsoft Azure, Google, etc). Furthermore, other pricing models are proposed in cheaper costs and are based on temporal perspective. For instance, Google proposes *per-minute* billing strategy. Otherwise, Microsoft virtual machines are billed in *pre-paid subscriptions*. Whereas, two pricing models offered by Amazon specify the instances costs based on temporal factor:

- *reserved instance* is a reservation of resources and capacity for 1 year up to 3 years. With reserved instances, a significant discount up to 30% compared to on-demand instances can be offered to a consumer.
- *spot strategy* offers to consumer the opportunity to bid for spare unused Amazon EC2 instances. Using this strategy type instead of on-demand type claims to consumer to save up to 90% in resources costs. The spot price fluctuates based on the supply and demand of available EC2 capacity[8]. To get the spot instance, clients specify the maximum price that they are willing to pay per instance hour based on the spot price history available via the Amazon EC2 API and the AWS Management Console [3]. If the spot price overcomes the bid price the spot instances will be interrupted. Therefore, Amazon proposes spot instances with predefined duration in hourly increments up to six hours in length.

We define $\mathcal{R}$ the set of Cloud resources where $\mathcal{R} = \{R_i, 1 \leq i \leq n\}$. Each resource R_i that has a memory of RAM_i and a CPU of CPU_i can be taken from a provider Pr_j where $j \in \{1, \cdots, p\}$. Each provider Pr_j proposes a set of different pricing models ($\mathbf{St_j} = \{St_{jk}, 1 \leq k \leq s\}$) for each resource R_i.
The formal definition of a Cloud pricing model is given in Definition 1.

Definition 1: (*Cloud pricing model*) A Cloud pricing model St_{jk} is defined as a triplet St_{jk}=(T_{jk},TC_{jk},c_{jk}) where:

- T_{jk} is the St_{jk} strategy type of the provider Pr_j
- TC_{jk} defines the temporal constraints imposed by the strategy St_{jk} of the provider Pr_j
- c_{jk} is the unit hour cost proposed by strategy St_{jk} of the provider Pr_j

The temporal constraint TC_{jk} is defined by the pricing model and presents the time span allowed for using a Cloud instance with a defined cost c_{jk}. It can be:

- relative temporal constraint: duration constraint is expressed in terms of time interval [MinAvR, MaxAvR] where $0 \leq MinAvR \leq MaxAvR \leq \infty$
- absolute temporal constraint: specifies the start and finish time of the instance temporal availability:
  - Start Using No Earlier Than (SUNET), Finish Using No Earlier Than (FUNET)
  - Start Using No Later Than (SUNLT), Finish Using No Later Than (FUNLT)

For more details, we refer interested reader to [3].
In table I and table II, we present a set of Cloud resources $\mathcal{R}$={R_1=m4.xlarge, R_2=r3.large, R_3=m3.2xlarge, R_4=F2, R_5=F4} offered by Pr_1=Amazon and Pr_2=Microsoft. Each resource R_i has a RAM_i, CPU_i and a different cost defined by the Cloud provider in various pricing models. For instance, R_1=m4.xlarge proposed by Pr_1=Amazon has a memory of RAM_1=4 GB and a CPU of CPU_1=32 GB. This instance is offered by Pr_1 in different pricing models St_{1k} (where $k \in \{1, \cdots, 4\}$) such as St_{13}=(spot, [1h,6h], 0.142$).

In our work, we assume that the size of the transferred data between instances is small. Therefore, the data transfer time is about some seconds which is negligible compared to activities duration expressed in hours. Furthermore, we consider that the instance operating system is Linux and the availability zone of the instances is us-east-1a.

B. Business process model

A business process model defines the relationship between a set of activities that are needed to achieve a business goal of an organization. More formally, a process model is represented as a directed graph where nodes are tasks, gateways or events and edges are control dependencies [3].

Definition 2: (Business Process Model) A business process model is a tuple ($\mathcal{A}$,E,F,$\mathcal{R}$,D,F,C) where:

- $\mathcal{A}$ is the set of activities where $\mathcal{A} = \{a_q, q \in \{1, \cdots, r\}\}$;
- $F : \mathcal{A} \longrightarrow T$ assigns temporal constraints to activities;
- $\mathcal{R}$ is the set of used Cloud resources;
- $E \subseteq \mathcal{A} \times \mathcal{A}$ is the set of edges;
- $D \subseteq \mathcal{A} \times \mathcal{R}$ is the set of relations between activities and resources;
- $\mathcal{C} : \mathcal{R} \longrightarrow C_i$ is a function used to compute the process cost

As described in Definition 2, business process activities are constrained by hard timing constraints T and need a set of Cloud resources $\mathcal{R}$ to be executed. The function $\mathcal{C}$ is utilized to compute the process cost (the Cloud resources cost based on the pricing models presented in section II-A and the penalty cost added if an activity is interrupted).

Activities temporal constraints: They can be relative and/or absolute [9]:

- relative such as duration which restricts the time span allowed for executing an activity and it is expressed in terms of a time interval $[MinD, MaxD]$ with $1 \leq MinD \leq MaxD$
- absolute specifies the start and finish times of process activities such as MSAT (Must Start At), SNET (Start No Earlier Than), FNET(Finish No Earlier Than), etc.

For instance, in Table III the activities a_1, a_5 and a_9 in the service supervision process presented in Fig. 1 have a temporal duration equal to [1h,2h]. For more details, we refer interested readers to [3].

For the process model, at the moment of resource allocation, the selection of the resource, the provider and the Cloud pricing models for each VM is done to run the process activities without violating temporal and QoS constraints. Let P be a process model, and $\mathcal{R} = \{R_i, 1 \leq i \leq n\}$ is the set of Cloud resources in different pricing models. Each a_q in P needs a minimal RAM of RAM_q and a minimal CPU of CPU_q to be executed. Furthermore, each Cloud resource R_i has a memory of RAM_i and a CPU of CPU_i. Pr is the set of Cloud providers Pr_j (where $j \in \{1, \cdots, p\}$) that have a set of pricing models $\{St_{jk}, k \in \{1 \cdots s\}\}$.
The resource allocation for the process activities is formally given in Definition 3.

TABLE I: Virtual Machine Instance Properties by Amazon EC2

VM	RAM	CPU	On-demand	Reserved (no upfront)	Spot predefined duration	Spot non-predefined duration
m4.xlarge	4 GB	32 GHz	0.215$/h	0.147$/h	0.129$/h [0h,1h]	0.0491$/h [06pm,01am$^{(+1)}$]
					0.142$/h [1h,6h]	0.0386$/h [01am,06pm]
r3.large	2 GB	16 GHz	0.166$/h	0.105$/h	0.096$/h [0h,1h]	0.0225$/h [03am,10pm]
					0.102$/h [1h,6h]	0.0381$/h [10pm,03am$^{(+1)}$]
m3.2xlarge	8 GB	30 GHz	0.532$/h	0.380$/h	0.293$/h [0h,1h]	0.0787$/h [10am,9pm]
					0.372$/h [1h,6h]	0.0863$/h [09pm,10am$^{(+1)}$]

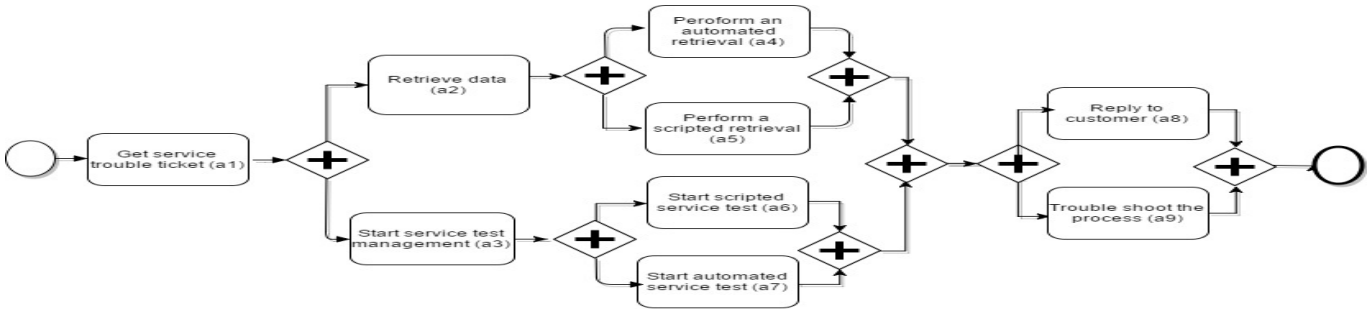


Fig. 1: A service supervision process

TABLE II: Virtual Machine Instance Properties by Microsoft

Instances	RAM	CPU	On-demand
F2	4 GB	32 GHz	0.128$/h
F4	8 GB	64 GHz	0.256$/h

Definition 3: (*Resource allocation AL*): The allocation of Cloud resources $\mathcal{R}$ for the activities $\mathcal{A}$ in the process model P is a function AL: $\mathcal{A} \longrightarrow \mathcal{R}$ that assigns for each activity $a_q \in \mathcal{A}$ an R_i that have RAM_i and CPU_i from the provider Pr_j and the strategy St_{jk} where $RAM_q \leq RAM_i$ and $CPU_q \leq CPU_i$ (where $i \in \{1, \cdots, n\}$ and $q \in \{1, \cdots, r\}$).

In table IV, we present the possible resource allocations for each activity while taking into consideration the resources properties and the activities QoS requirements (RAM, CPU). For instance, the activity a_1 needs a minimal memory of RAM_1=4 GB and a CPU of CPU_1=32 GB can be assigned to R_1=m4.xlarge offered by Pr_1=Amazon. The strategy type of R_1 can be St_{13}=(spot, [1h,6h], 0.142$).

III. Motivating Example

Our research is motivated by a real business process (Fig. 1) of the Telco operator Orange, one of our industry partners. We use BPMN to design the process as it is one of the most popular business process modeling language [10], [11], [12]. In Table III, we present the activity temporal duration [3]. For instance, the execution of activities a_1, a_5 and a_9 takes from 1 hour to 2 hours. Furthermore, activities need virtual machines to be executed as shown in Table III. For instance, activities a_1 and a_2 require a virtual machine with a minimal memory of 4 GB and a minimal CPU of 32 GB. Therefore, in table IV are presented the possible resource allocations. Besides, some activities in the process are critical and if they will not be executed correctly a penalty cost should be paid by the enterprise. For example, in Table III, a_4 has a penalty cost equal to 0.2$ defined by the Cloud provider.

As mentioned in section II, different instance types are offered by Cloud providers. Tables I and II show the Cloud instances properties and their associated costs in various pricing strategies. In fact, Table I (respectively Table II) presents Amazon EC2 (respectively Microsoft Azure) instances. For example, R_1=m4.xlarge has a memory of RAM_1=4 GB and a CPU of CPU_1=32 GHz and its unit hourly cost is $\{$0.215$, 0.147$, 0.142$, 0.0386$$\}$ in $\{$on-demand, reserved, spot predefined duration, spot non-predefined duration$\}$ strategy. In addition, as it is presented in Table IV various allocations are possible to run activities. For example, a_1, a_2, a_4, a_7, a_8 and a_9 can use R_1=m4.xlarge from Amazon or R_4=F2 from Microsoft. Otherwise, a_5 may be performing in R_3=m3.2xlarge from Amazon or R_5=F4 from Microsoft. Whereas, only R_5=F4 from Microsoft can be allocated for a_3.

Cloud resources are offered in different pricing strategies. That is why each activity can use a Cloud instance in a pricing strategy proposed by the Cloud provider. In Table V we present two possible choices taken from a set of possible resource allocation. For example, in the first choice a_1 uses R_1 from Pr_1=Amazon as a St_{14}=spot non-predefined duration

TABLE III: Process activities requirements

Activities	a1	a2	a3	a4	a5	a6	a7	a8	a9
Durations	[1h,2h]	[2h,3h]	[1h,1h]	[1h,4h]	[1h,2h]	[2h,3h]	[1h,1h]	[1h,1h]	[1h,2h]
Penalties	0.7$	0$	0$	0.2$	0$	0$	0$	0$	0$
RAM	4 GB	4 GB	8 GB	2 GB	8 GB	8 GB	2 GB	4 GB	4 GB
CPU	32 GHz	32 GHz	64 GHz	16 GHz	30 GHz	30 GHz	16 GHz	32 GHz	16 GHz

TABLE IV: Resource allocation

Cloud Providers	**Instances**	**Activities**
Amazon	m4.xlarge	a_1 & a_2 & a_4 & a_7 & a_8 & a_9
Amazon	r3.large	a_4 & a_7
Amazon	m3.2xlarge	a_4 & a_5 & a_6 & a_7 & a_9
Microsoft	F2	a_1 & a_2 & a_4 & a_7 & a_8 & a_9
Microsoft	F4	a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9

TABLE V: Two Possible Choices

Activities	**First Choice**	**Second Choice**
a_1	R_1 from Pr_1 from St_{14}	R_1 from Pr_1 from St_{11}
a_2	R_1 from Pr_1 from St_{13}	R_1 from Pr_1 from St_{13}
a_3	R_5 from Pr_2 from St_{21}	R_5 from Pr_2 from St_{21}
a_4	R_2 from Pr_1 from St_{14}	R_2 from Pr_1 from St_{14}
a_5	R_3 from Pr_1 from St_{14}	R_3 from Pr_1 from St_{14}
a_6	R_3 from Pr_1 from St_{13}	R_3 from Pr_1 from St_{13}
a_7	R_2 from Pr_1 from St_{13}	R_2 from Pr_1 from St_{13}
a_8	R_4 from Pr_2 from St_{21}	R_4 from Pr_2 from St_{21}
a_9	R_3 from Pr_1 from St_{11}	R_3 from Pr_1 from St_{11}
Process cost	C=4.4218$	C=4.0746$

strategy and a_8 uses R_4 from Pr_2=Microsoft as St_{21}=on-demand strategy. The only difference between the two choices presented in the Table V is the strategy of the instance R_1 performing a_1. We compute the process cost for each choice based on the equation 1. Indeed, the cost is (i) the sum of the unit hourly cost c_{ijk} of R_i from Pr_j in strategy St_{jk} by the activity duration d_q (we assume that $d_q = MaxA_q$) and (ii) the sum of activities penalties costs.

$$C = \sum_{q=1}^{|\mathcal{A}|} \sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} d_q c_{ijk} + \sum_{q=1}^{|\mathcal{A}|} p_q \quad (1)$$

We notice that, in spit of the use of the cheapest strategy for R_1, the process cost of the second choice is lower. This is due to the add of activity a_1 penalty cost since there is a risk of interruption for R_1 while it is considered as a spot instance with non-predefined duration. So, it will be primordial to find the optimal pricing strategy for each Cloud resource while in some cases using a more expensive strategy is better than using a cheaper one with an interruption risk especially for critical activities. Indeed, in order to find the optimal cost of the process using various Cloud providers pricing strategies, one needs to find the Cloud resources that do not violate QoS constraints and to find the best strategy that does not violate activities temporal constraints and to take into account the penalty cost. In other words, one has to find the allocation that has the optimal cost. Optimality is expressed in terms of two criteria: (i) minimizing the use of expensive Cloud instances while guaranteeing the activities requirements (ii) minimizing the penalty cost of critical activities. Therefore, in next section we aim at presenting our method to solve this issue.

IV. The Linear Program

This section presents our proposed binary linear program to find the optimal Cloud resource allocation cost of a process model P. First, the necessary inputs, decision variables are introduced then, the linear program is presented (constraints and objective function). The inputs are:

- the set of activities $\mathcal{A}$ in P, the set of the activities' QoS, $QoS_{\mathcal{A}} = \{QoS_{a_q} : a_q \in \mathcal{A}\}$, the set of activities temporal constraints $TC_{\mathcal{A}}$=$\{TC_{a_q} : a_q \in \mathcal{A}\}$;
- the set of Cloud resources $\mathcal{R} = \{R_i, \forall i \in \{1, \cdots, n\}\}$ required by activities $\mathcal{A}$.
- the set of Cloud providers $Pr = \{Pr_j, \forall j \in \{1, \cdots, p\}\}$ and the set of Cloud pricing models St_{jk}=(T_{jk},TC_{jk},c_{jk}) where $k \in \{1, \cdots, s\}$ of each provider Pr_j.

In the following, we present the decision variables of our mathematical model:

$$X_{ijkq} = \begin{cases} 1 & \text{if } R_i \in \mathcal{R} \text{ is from } Pr_j \text{ in strategy } k \text{ and} \\ & \text{is assigned to activity } a_q, \\ 0 & \text{otherwise} \end{cases}$$

$$V_q = \begin{cases} 1 & \text{if } a_q \in \mathcal{A} \text{ uses a spot instance with non} \\ & \text{predefined duration and its penalty is} \\ & \text{not null} \\ 0 & \text{otherwise} \end{cases}$$

The objective function of the model: (i) selects the suitable Cloud resource for each activity (ii) selects Cloud providers and Cloud pricing models for each Cloud resource. So, the objective function seeks to select the resources that respect the constraints and achieves a minimum total Cloud resources cost.

$$MinC = \sum_{q=1}^{|\mathcal{A}|} \sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} d_q c_{ijk} X_{ijkq} + \sum_{q=1}^{|\mathcal{A}|} p_q V_q \quad (2)$$

The total execution cost includes the sum of resources activities allocation costs in order to execute the process and the sum of penalties costs added when a spot instance with non-predefined duration is used for critical activities. The allocation cost is given by the multiplication of the execution time of the activity d_q by the R_i utilization cost. The cost of R_i is the hourly unit strategy cost c_{ijk} proposed by a provider Pr_j. The second cost is given by the addition of each activity penalty cost in case of using a spot instance having a risk of interruption. This penalty cost is defined by the enterprise.
The objective function is subject to the following sets of linear constraints:

1) *Resource constraints*: The resource constraints ensure that the resources' capacities in processing and memory satisfy the activity requirements.

$$\sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} min(RAM_i) X_{ijkq} \geq RAM_q V_q, \quad \forall q \in \{1, \cdots, r\} \quad (3)$$

$$\sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} min(CPU_i) X_{ijkq} \geq CPU_q V_q, \quad \forall q \in \{1, \cdots, r\} \quad (4)$$

2) Activities temporal constraints: In order to ensure the respect of activities temporal requirements, constraint 5 guarantees that each activity starts (S_q) each execution after the maximal end time (F_r) of all its predecessors executions.

$$max(\mathrm{F}_r) \leq S_q \ : \ \forall a_q \in A \text{ and } r < q \quad (5)$$

3) *Pricing strategy constraints:* The following constraints identify the time span allowed to use a resource R_i ($[MinAvR_i, MaxAvR_i]$) and also guaranty that R_i is available from the start until the end time of the activity requiring it.

$$\sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} MinAvR_i X_{ijkq} \geq MinD_q V_q, \quad \forall q \in \{1, \cdots, r\} \quad (6)$$

$$\sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} MaxAvR_i X_{ijkq} \geq MaxD_q V_q, \quad \forall q \in \{1, \cdots, r\} \quad (7)$$

$$\sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} SUNET(R_i) X_{ijkq} \leq S_q V_q, \quad \forall q \in \{1, \cdots, r\} \quad (8)$$

$$\sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} FUNET(R_i) X_{ijkq} \geq F_q V_q, \quad \forall q \in \{1, \cdots, r\} \quad (9)$$

4) *Interruption constraint:* This constraint is used to ensure the addition of the penalty cost when the instance used has an interruption risk.

$$\sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} X_{ijkq} str_k = V_q : \forall a_q \in A, p_q > 0 \text{ and } str_k = 1 \quad (10)$$

5) *Placement constraint:* This constraint 11 ensures that each task uses an only one instance type, Cloud provider and pricing strategy.

$$\sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{p} \sum_{k=1}^{s} X_{ijkq} = 1 : \forall a_q \in A \quad (11)$$

6) Assignment constraint: Each instance type, Cloud provider and strategy pricing is used by an only one activity.

$$\sum_{q=1}^{|\mathcal{R}|} X_{ijkq} = 1 : \forall i \in \{1, \cdots, n\}, \forall j \in \{1, \cdots, p\}, \quad \forall k \in \{1, \cdots, s\} \quad (12)$$

7) *Binary constraints:* We are dealing with a linear program, therefore, we add equations (13-14) to impose that the decision variables should be either 0 or 1 (binary variables).

$$V_q \in \{0, 1\}, \quad q \in \{1, \cdots, r\} \quad (13)$$

$$X_{ijkq} \in \{0, 1\}, \quad \forall q \in \{1, \cdots, r\}, i \in \{1, \cdots, n\}, \quad j \in \{1, \cdots, p\}, k \in \{1, \cdots, s\} \quad (14)$$

In Fig. 1, a_1 requires a minimal RAM_1=4 GB and a minimal CPU_1=32 GB then its assigned resource should be the one having the minimal RAM_i and minimal CPU_i and satisfying the activity requirements (Equations (3) and (4)). In addition, a_2 and a_3 are the successors of a_1 then they might start after the finish of a_1 (equation (5), i.e, $F_1 \leq S_2$ and $F_1 \leq S3$). Furthermore, while a_1 has a temporal duration ($MinD_1$ =1

hour and $MaxD_1$=2 hours), then R_i offered by the provider Pr_j in strategy St_{jk} should satisfy the temporal constraints of the pricing strategy. For instance, if R_i=m4.xlarge proposed by Pr_1=Amazon and is from the strategy St_{13}=spot with predefined duration, therefore, $MinD_1$=1 hour $\geq$ $MinAvR_1$ =1 hour and $MaxD_1$=2 hours $\leq$ $MaxAvR_1$=6 hours and X_{1131}=1 (Equations (6) and (7)).
However, if the resource R_1 is taken from Pr_1=Amazon and is from the strategy St_{14}=spot with non predefined duration, therefore, $SUNET(R_1)$ <= S_1 =08am and $FUNET(R_1) \geq F_1$ =10am (Equations (8) and (9)). Some activities in the example presented in Fig. 1 have penalties costs that should be added if they did not finish their execution. For instance, a_1 has a penalty cost, therefore, if a_1 uses a spot instance with non predefined duration its penalty cost should be considered when computing the optimal cost (Equation (10)). Besides, a_1 can use only one Cloud instance R_1 from Pr_1=Amazon from St_{13}=spot strategy and this R_1 performs only a_1 then X_{1131}=1 (Equations (11), (12)).

V. Evaluation

In this section, we investigate through numerical experiments the behavior of our proposed approach. First, in section V-A, we evaluate the efficiency (i.e., converging quickly to optimal solution) and the performance (i.e., a reasonable computation time) of our solution. Second, in section V-B, we evaluate the scalability of our approach.

For the analysis, we implemented the proposed optimization problem using IBM-ILOG Cplex Optimization Studio V12.6.3 on a laptop with a 64-bit Intel Core 2.3 GHz CPU, 6 Go RAM and Windows 7 as OS. For the first experiment, the data inputs are the service supervision process and the set of Cloud resources presented in section III. As presented in Table III, the process activities have temporal and QoS requirements and two tasks have penalty costs. The Cloud providers considered are Amazon and Microsoft. Whereas, for the second experiment, the data inputs are defined randomly from the ranges presented in Table VI. For instance, the number of providers should be in (1,2).

TABLE VI: Data Input Ranges

Information	Type	Range
Providers' number	integer	$[1, \cdots, 2]$
Amazon strategies' number	integer	$[1, \cdots, 4]$
Microsoft strategies' number	integer	$[1, \cdots, 1]$
CPU number	integer	$[16, \cdots, 64]$
RAM amount	double	$[2, \cdots, 8]$
Compute price	double	$[0.01\$, \cdots, 0.532\$]$
Requirement in CPU	integer	$[8, \cdots, 64]$
Requirement in RAM	double	$[2, \cdots, 8]$
Activities' number	integer	$[2, \cdots, 90]$
Activities' durations	integer	$[1, \cdots, 5]$
Penalty cost	double	$[0\$, \cdots, 1\$]$

TABLE VII: Experiments 1 Results

Group	Minimal	Average
G1	3.9199$	2.903$
G2	3.7976$	2.5719
G3	3.9619$	2.93027$
G4	3.9256$	2.5719$
G5	3.8664$	2.5719$
G6	3.739$	2.5719$
G7	3.8211$	2.4309$
G8	3.91425$	2.5719$

A. Experiments 1: Variability-based Evaluation

An existing classical approach can be adapted for our problem. In fact, one can follow the following steps: (i) searching a collection of possible allocations respecting the linear program constraints (ii)computing the cost of each one and (iii) selecting the allocation having the optimal cost.

That is why, we propose for 10 users divided in 8 groups $\mathcal{G} = \{G_i, i \in 1, \cdots, 8\}$ to select randomly 4 possible allocations from a collection of 10. The number of user of each G_i is equal to $i + 2$ where $i \in \{1, \cdots, 8\}$. The groups submit their response one by one.

As shown in Table VII, we notice the variation of the average process cost. Three minimal values presented the minimal value for each group. The lowest minimal cost is only got for the seventh group. This value is the result of our implemented linear program. Therefore, this naive solution gives the optimal cost for our problem but it is labour intensive and time consuming. In fact, to get the optimal resource allocation many users and selections are needed. Furthermore, if the number of activities is big it will be so hard to get manually the optimal solution. Using our approach we can rapidly define for each resource the best strategy to use in order to optimize the allocation cost.

B. Experiments 2: Scalability

In the second experiment, we change the number of process activities, the maximum number of instances proposed by each Cloud provider in various pricing strategies. First, we consider that some activities have penalty costs. The results are shown in Table VIII. So, compared to a naive approach, our approach discover rapidly (the processing time $t_{cp} = 7.546s$) the solution for our optimization problem. Second, we consider that the activities penalty costs are null. We remark that the

TABLE VIII: Experiments 2 Results

Nb Activities	Nb Providers	Nb Strategies	Nb VM Types	Proposed LP $p_q > 0$	Proposed LP $p_q = 0$	Percentage
2	1	4	2	ob=0.39\$ t_{cp}=1.6s	ob_p=0.3718\$ t_{cp_p}=1.54s	4.66% 4%
3	2	5	3	ob=0.659\$ t_{cp}=1.786s	ob_p=0.3718\$ t_{cp_p}=1.693s	4.73% 5%
4	2	14	5	ob=0.918\$ t_{cp}=1.95s	ob_p=0.8178\$ t_{cp_p}=1.820s	11% 6.66%
5	2	14	5	ob=1.0752\$ t_{cp}=2.216s	ob_p=0.9552\$ t_{cp_p}=2.03s	11.16% 8.39%
9	2	14	5	ob=2.4309\$ t_{cp}=2.701s	ob_p=2.1309\$ t_{cp_p}=2.31s	13.45% 14%
30	2	56	19	ob=9.02\$ t_{cp}=3.7s	ob_p=7.678\$ t_{cp_p}=3.05s	14.87% 18%
60	2	112	38	ob=18.04\$ t_{cp}=5.15s	ob_p=15.14\$ t_{cp_p}=4.2s	16.07% 18.44%
90	2	169	50	ob=26.343\$ t_{cp}=7.546s	ob_p=22.05\$ t_{cp_p}=5.9s	16.29% 21.81%

objective function and the response time are lower than the one considering penalty cost. To compare the execution time and the process cost between considering that activities have penalties costs and have not, we compute the percentage based on the equation 15. In fact, the enterprise can save up to 16.29% in process cost when activities penalty costs are null. Furthermore, the solution is discovered more quickly (21.81%).

$$percentage = 100\% - \frac{ob}{ob_p} \quad \text{and} \quad percentage = 100\% - \frac{t_{cp}}{t_{cp_p}}$$

VI. Related Work

There exist previous research work on the optimal Cloud resource allocation. Wang et al [13] developed two distributed algorithms in order to optimize the data-center net profit with deadline-dependent scheduling by jointly maximizing revenues and minimizing electricity costs. In [14], [15], [16] authors propose approaches to optimize cloud resources cost and the processing time by considering different pricing schemes such as on-demand and reserved models. Those works do not deal with business process models that use Cloud resources. Rather our approach optimizes process cost while considering different Cloud pricing models.

A cost-efficient deployment approach for elastic business process management systems is proposed by [17]. In fact, a mixed integer linear programming technique is applied to deal with an optimization problem while considering the data transfer communication requirements. Goettelmann et al. in [18] propose three algorithms in order to optimize the business process deployment into various public clouds while taking into account the data transfer time and cost.

The optimal process execution cost has been addressed for single process models such in [19] where authors propose a method to predict the execution path of a business process, estimate and optimize the cost of the used Cloud resources based on pricing strategies. In [20], a provisioning approach of Cloud resources for dynamic workflows is proposed. The authors propose an algorithm to allocate Cloud resources for a dynamic workflow which takes into account some constraints. Then, they extend their work in order to support the dynamic changes of workflow. However, those works do not consider that the enterprise uses Cloud resources from different Cloud providers in various pricing models.

The issue of selecting the vendor or the pricing model is addressed in various contexts. For instance, in [21], [22]authors proposed linear programming models in order to decide who is the best vendor that should be selected to satisfy a set of requirements.

In [23], [24] authors propose a mixed integer programming model to allocate human resources while considering employee availability as a major constraint of employee scheduling. While our purpose in this paper is to discover which is the best pricing strategy for each Cloud resource to satisfy business process activities and to minimize Cloud resource allocation and activities penalty costs.

When selecting the pricing model for each VM, the process designer should verify the matching between activities temporal constraints and Cloud instances temporal availabilities. This step was the subject of our recent work [3]. Nevertheless, we have not consider that activities can have penalty cost. The

latter can be added especially when the task uses the cheapest instance, spot instance with non-predefined duration, that can be interrupted. In other words, sometimes choosing a more expensive pricing model for Cloud instances is better than using a cheaper one with a risk of interruption. Consequently, if we just verify that the resource allocation is temporally consistent when using cheapest instances this does not guarantee that the process has an optimal cost. That is why, optimizing process cost enriched with temporal constraints and using different pricing strategies is required by the process designer.

Different authors focus on the area of configurable process models. Some authors define methods to derive the best process variants from a configurable process model. For instance, in [25] variants are derived based on domain constraints and business rules. Rekik et al [26] define an approach for optimal configurable process deployment into a Cloud federation. However, they do not handle the issue of optimal allocation while considering pricing strategies.

VII. Conclusion

In this paper, we presented an approach that helps an enterprise optimizing its business process cost deployment into the Cloud. We mapped the problem to discover the best pricing model for each Cloud resource required to execute business process activities. This problem is solved through a binary linear program with an objective function under a set of constraints. The latter satisfy the activities (temporal, QoS) requirements, the Cloud resources constraints and the pricing strategies constraints. The experimental results show that our approach seems to perform very well both in terms of cost minimization, performance, efficiency and scalability.

As future work, we aim at extending our approach with the notion of configurable business process. Furthermore, we plan to propose an approach to optimize the business process cost at run time. Finally, we intend to propose a scheduling approach to match the business process activities temporal requirements with Cloud pricing strategies temporal constraints in order to optimize the resources costs.

Acknowledgment

This work is partially supported by the OCCIware research and development project[1] funded by French Programme d'Investissements d'Avenir (PEA)

References

[1] O. Givehchi, H. Trsek, and J. Jasperneite, "Cloud computing for industrial automation systemsa comprehensive overview," in *Emerging Technologies & Factory Automation, IEEE 18th Conference on*. IEEE, 2013, pp. 1–4.

[2] S. Boubaker, W. Gaaloul, M. Graiet, and N. B. Hadj-Alouane, "Event-b based approach for verifying cloud resource allocation in business process," in *Services Computing, IEEE International Conference on*. IEEE, 2015, pp. 538–545.

[3] R. B. Halima, S. Kallel, K. Klai, W. Gaaloul, and M. Jmaiel, "Formal verification of time-aware cloud resource allocation in business process," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2016, pp. 400–417.

[4] Z. Huang, W. M. van der Aalst, X. Lu, and H. Duan, "Reinforcement learning based resource allocation in business process management," *Data & Knowledge Engineering*, vol. 70, no. 1, pp. 127–145, 2011.

[5] S. Schulte, P. Hoenisch, C. Hochreiner, S. Dustdar, M. Klusch, and D. Schuller, "Towards process support for cloud manufacturing," in *Enterprise Distributed Object Computing Conference, IEEE 18th International*. IEEE, 2014, pp. 142–149.

[6] P. Hoenisch, S. Schulte, S. Dustdar, S. Venugopal *et al.*, "Self-adaptive resource allocation for elastic process execution." in *IEEE CLOUD*, 2013, pp. 220–227.

[7] S. Cheikhrouhou, S. Kallel, N. Guermouche, and M. Jmaiel, "The temporal perspective in business process modeling: a survey and research challenges," *Service Oriented Computing and Applications*, vol. 9, no. 1, pp. 75–85, 2015.

[8] Online, "Amazon ec2," https://aws.amazon.com/ec2/pricing/.

[9] S. Kallel, "Specifying and monitoring non-functional properties," Ph.D. dissertation, Dissertation, Darmstadt, Technische Universität Darmstadt, 2011, 2011.

[10] N. N. Chan, W. Gaaloul, and S. Tata, "Assisting business process design by activity neighborhood context matching," in *International Conference on Service-Oriented Computing*. Springer, 2012, pp. 541–549.

[11] ——, "Context-based service recommendation for assisting business process design," in *International Conference on Electronic Commerce and Web Technologies*. Springer, 2011, pp. 39–51.

[12] M. Sellami, W. Gaaloul, and S. Tata, "Functionality-driven clustering of web service registries," in *Services Computing (SCC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 631–634.

[13] W. Wang, P. Zhang, T. Lan, and V. Aggarwal, "Datacenter net profit optimization with deadline dependent pricing," in *Information Sciences and Systems, 46th Annual Conference on*. IEEE, 2012, pp. 1–6.

[14] Q. Li and Y. Guo, "Optimization of resource scheduling in cloud computing," in *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2010.

[15] M. Hu, J. Luo, and B. Veeravalli, "Optimal provisioning for scheduling divisible loads with reserved cloud resources," in *18th IEEE International Conference on Networks*. IEEE, 2012, pp. 204–209.

[16] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Transactions on Services Computing*, pp. 164–177, 2012.

[17] P. Hoenisch, C. Hochreiner, D. Schuller, S. Schulte, J. Mendling, and S. Dustdar, "Cost-efficient scheduling of elastic processes in hybrid clouds," in *IEEE 8th International Conference on Cloud Computing*. IEEE, 2015, pp. 17–24.

[18] E. Goettelmann, W. Fdhila, and C. Godart, "Partitioning and cloud deployment of composite web services under security constraints," in *Cloud Engineering, IEEE International Conference on*. IEEE, 2013, pp. 193–200.

[19] T. Mastelic, W. Fdhila, I. Brandic, and S. Rinderle-Ma, "Predicting resource allocation and costs for business processes in the cloud," in *World Congress on Services, New York City, NY, USA*, 2015, pp. 47–54.

[20] F. Fakhfakh, H. H. Kacem, and A. H. Kacem, "A provisioning approach of cloud resources for dynamic workflows," in *Cloud Computing (CLOUD),IEEE 8th International Conference on*. IEEE, 2015, pp. 469–476.

[21] H. S. Kilic, "An integrated approach for supplier selection in multi-item/multi-supplier environment," *Applied Mathematical Modelling*, vol. 37, no. 14, pp. 7752–7763, 2013.

[22] E. A. Demirtas and Ö. Üstün, "An integrated multiobjective decision making process for supplier selection and order allocation," *Omega*, vol. 36, no. 1, pp. 76–90, 2008.

[23] S. M. Al-Yakoob and H. D. Sherali, "Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations," *Annals of Operations Research*, vol. 155, no. 1, pp. 119–142, 2007.

[24] M. Afilal, H. Chehade, and F. Yalaoui, "The human resources assignment with multiple sites problem," *International Journal of Modeling and Optimization*, vol. 5, no. 2, p. 155, 2015.

[25] N. Assy and W. Gaaloul, "Extracting configuration guidance models from business process repositories," in *International Conference on Business Process Management*. Springer, 2015, pp. 198–206.

[26] M. Rekik, K. Boukadi, N. Assy, W. Gaaloul, and H. Ben-Abdallah, "A linear program for optimal configurable business processes deployment into cloud federation," in *IEEE International Conference on Services Computing, San Francisco, CA, USA*, 2016, pp. 34–41.

[1] http://www.occiware.org