Simulation of Configurable Resource Allocation for Cloud-Based Business Processes

Mehdi Ahmed-Nacer, Kunal Suri, Mohamed Sellami, Walid Gaaloul

Simulation of Configurable Resource Allocation for Cloud-Based Business Processes

Mehdi Ahmed-Nacer[*†], Kunal Suri[†‡], Mohamed Sellami[§] and Walid Gaaloul[†]

[*]University of Sciences and Technology Houari Boumediene, Algiers, Algeria
[†]Telecom SudParis, UMR 5157 Samovar, Université Paris-Saclay, Evry, France
[‡]CEA, LIST, Laboratory of Model Driven Engineering for Embedded Systems (LISE), P.C. 174, Gif-sur-Yvette, France
[§] LISITE Laboratory. ISEP Paris, France
{firstname.lastname}@telecom-sudparis.eu, m.ahmed.nacer@usthb.dz, mohamed.sellami@isep.fr

Abstract—**Cloud computing has enabled provisioning of scalable and virtualized resources to organizations in an ubiquitous and on-demand manner. Likewise, configurable process modeling has enabled organizations to reuse their existing knowledge by sharing a reference process model between different tenants that can be customized according to specific needs. Nevertheless, there are some limitations in this context, (i) it is not cost-effective to make use of real cloud infrastructure such as, Amazon EC2 for experimenting and analyzing the deployment of the best process variant, (ii) frequent changes in the configurations during experimentation using a real cloud setup involves various manual adjustments which is a time-consuming task, (iii) certain conditions prevailing in the internet-based cloud environments are beyond the control of the practitioners involved in such analysis. Thus, making use of simulation environments is one of the good alternative for testing and analyzing the best process variants with optimal resource allocation. Building upon our previous work, we propose in this paper, (i) a unified description model that allows the representation of process activities along with their possible cloud resource allocation alternatives, (ii) a methodology for simulating a configurable process along with its cloud resource consumption, and (iii) an extension of CloudSim framework in order to realize the simulation of the configurable resource allocation for the cloud-based business processes.**

Index Terms—**Simulation; Configurable; Business Process; Resource Allocation; Cloud**

I. INTRODUCTION

According to the National Institute of Standards and Technology (NIST), cloud computing represents a model that enables a cloud provider to share their computing resources to users who can access them in an ubiquitous, convenient and on-demand way with minimal management effort [1]. Likewise, *configurable process modeling* helps an organization to reuse its existing knowledge by integrating all possible process variations of the same business process in a given domain into one customizable model, thus providing a generic reference model [2]. In other words, it extends a regular process model with variation points, known as *configurable elements* which allows to have multiple design options for the process. This configurable process model enables practitioners to derive the appropriate process variants depending on their specific needs [3]. Thus, the configurable process modeling technique empowers an organization having different branches in different geographical locations or having different departments executing the same process with small differences,

to share and reuse the process knowledge and derive new processes with minimal efforts. Moreover, the concept of configurable process modeling is highly complementary to cloud computing since both allow the sharing of control-flow or resource elements among multi-tenant business processes [4]. Thus, provisioning business processes in the cloud helps an enterprise to adopt agile, flexible and cost-effective business solutions along with reducing the process development and maintenance costs.

To take advantage of cloud infrastructure for process deployment, an organization first needs to derive the *best process variant* (at design time) among the large number of possible process variants from its configurable process model. The best process variant can be expressed in terms of different functional and non-functional requirements such as, time, cost, availability, etc. making it a multi-criteria optimization problem [5]. Moreover, in contrast to the classical configurable process modeling technique that are based on the use of only control-flows, the practitioners involved in making use of cloud-based configurable processes need to keep the cloud resource perspectives additionally in the mind. In fact, this is due to two main reasons, on the one hand, they need to define a resource allocation for each activity and the dependencies that might exist between different resources. Since each tenant has its own specific requirements in terms of resource consumption, the resource allocation has to be taken into account while creating process variants. Thus, classical configurable process models have to be extended with resource configuration facilities. On the other hand, the deployment of a cloud-based configurable process model should also take into account the variable, dynamic and economic nature of the cloud ecosystem (e.g. pay-as-you go model, dynamic resource assignment and release, variable and dynamic cloud offers, etc.). It would be inefficient if all the allocated process resources along with their dependencies and actions are instantiated, while some of them are not used properly as there were no proper techniques to explicitly model the dependencies during the process configuration stage.

Thus, the objective of our research is to enable the users to derive the *best process variant* from its configurable process model keeping the cloud resource perspective in mind and to simulate its deployment in a cloud environment. In absence of

a simulation framework, real cloud infrastructures have to be rented for the time needed to analyze the best configuration which is neither good from cost perspective nor from the perspective of managing frequent changes in the resource allocation policy during the experiments. For the same reason, we propose a unified description model that allows the representation of business process activities along with their possible cloud resource allocation alternatives. We then provide a methodology for simulating a configurable business process with its associated cloud resource allocation to the process activities. Additionally, we provide an extension to the CloudSim framework in order to realize the simulation of the configurable resource allocation for cloud-based business process. Furthermore, our work will assist the users to avoid deployment failures in the real cloud infrastructure due to wrong resource allocation policies.

The remainder of the paper is organized as follows: Section II motivates the problem with a real-world case from *Orange France*, a telecommunication operator. In Section III, we provide the background about CloudSim and our approach on configurable cloud resource allocation. In Section IV, we describe our approach. In Section V, we present the experimentation and implementation along with the analysis of the results. In Section VI, we discuss the work done in context of configurable resource allocation in business process development for cloud environment. Finally, in Section VII we summarize the proposed work and presents an outlook on future work.

II. MOTIVATING EXAMPLE

Our research is motivated by a real configurable business process provided by one of our industrial partners, *Orange France*, a French telecommunications provider. The configurable process depicted in Figure 1[1] represents steps involved in addressing a customers complaint due to service quality drop. The process is modeled with the configurable Business Process Model and Notation (c-BPMN). In a configurable modeling language, the configurable elements can be the activities and the gateways and are modeled with a thick line [3]. For example, the activities $a11$, $a12$ and $a14$ and the gateways $XOR1$, $XOR2$, $XOR3$, $XOR8$, $OR4$ and $OR5$ are configurable. This process is used to derive variants that are then deployed into a cloud infrastructure.

The process starts upon receiving a complaint through a trouble ticket (activity $a1$) after which three sub-processes may arise (connected by the configurable XOR), (i) data retrieval: an expert executes the activities so that customer data are recuperated (sub-process starts with the activity $a2$), (ii) service test management: the expert executes activities to detect anomalies related to the complaint (sub-process starts with the activity $a3$), and (iii) resource test management: activities are executed to remotely setup the necessary parameters for complaint analysis (sub-process starts with the activity $a4$).

In the sub-process starting with activity $a2$, the expert can configure one of the three following steps, they are, perform an automated retrieval (activity $a5$) or perform a manual retrieval (activity $a6$) or perform a scripted retrieval (activity $a7$). In the sub-process starting with the activity $a3$, the expert can configure either one or both the following steps, they are, start a scripted service test (activity $a8$) or start an automated service test (activity $a9$). In the sub-process starting with activity $a4$, the experts can configure the following tasks, start a manual resource test (activity $a10$), start a resource test via a script (activity $a11$), start an automated resource test (activity $a12$). Once the various steps involved in the before mentioned sub-process are completed then, either a troubleshooting is done (activity $a13$) or the expert can configure the ticket escalation step (activity $a14$).
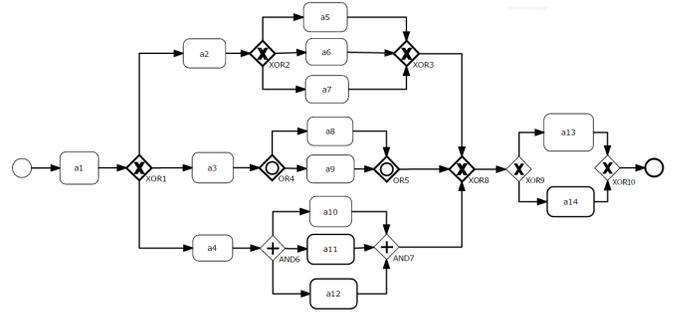

Fig. 1: Configurable Service Supervision Process

An example of one of the process variant derived from the configurable process (Figure 1) is illustrated in Figure 2. It depicts a process version without the *Resource Test Management* sub-process. In this variant the activity $a1$, $a2$, $a3$, $a5$, $a7$, $a8$, $a9$ and $a13$ are depicted wherein all of them need a Central Processing Unit (CPU) cores to execute their applications which is provided through the Virtual Machine (VM) resource. The activities also need a network to communicate with other applications which is included as a *shareable* resource. Furthermore, the activities $a1$ and $a13$ need an elastic storage resource, that for security issues is not shared with other activities or instances. The parameters of resource (capacity, cost, etc.) and the type of elastic operator (vertical or horizontal) are specified in the *unified descriptor model* (section IV-A).

Suppose that *Affiliate A* configures the process as shown in Figure 2. Indeed, as the CPU core is not elastic, thus if the execution of the activities exceeds the capacity of the CPU core, then execution will fail. However, since the storage is elastic, thus if the storage capacity needed by the activities exceed the capacity of the storage resource provisioned initially, then a new storage will be created and added to the process. However, this will make the overall cost of the resource allocation to increase. Suppose that another *Affiliate B* configures the process variant as shown in Figure 3 and the business process result in having activities $a1$, $a2$, $a4$, $a5$, $a7$, $a10$, $a11$, and $a13$. Since the activities $a10$ and $a11$ are

executed concurrently (connected via *AND*), when this process gets deployed, a new CPU will be provisioned and used for each activity as the CPU core is not an elastic resource. This additional resources will also increase the total cost of the resource consumed. Furthermore, a detailed example on the cloud resource allocation to business process activities is provided in Section IV.
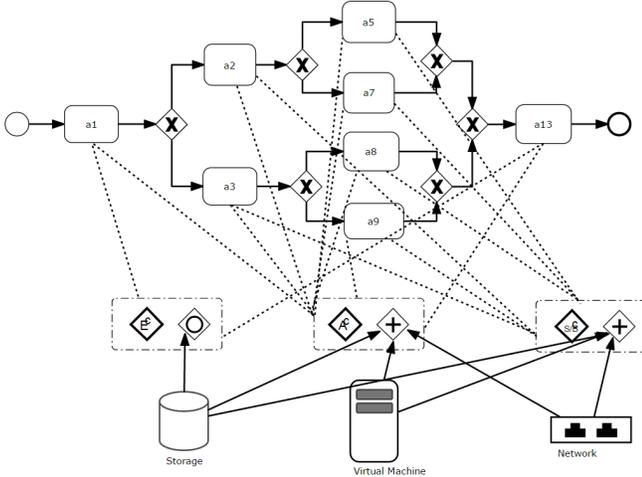

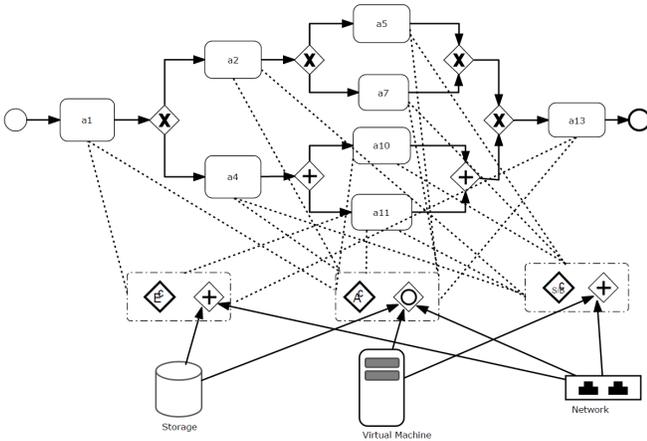Fig. 2: Process Variant 1 and its Allocated Cloud Resources Derived From the Configurable Process (Fig. 1)

Fig. 3: Process Variant 2 and its Allocated Cloud Resources Derived From the Configurable Process (Figure 1)

III. BACKGROUND

In this section, we present some definitions about configurable cloud resource allocation (III-A, III-B, III-C) for multi-tenant business process development [6] and CloudSim framework (III-D).

As mentioned before, while flexibility at the level of control-flow perspective in business processes has been widely defined in the literature, the flexibility at the level of the resource perspective has been neglected. In order to model such variability, we have extended configurable process modeling tech-

nique with configurable resource allocation to allow tenants to customize their needed resources in the context of cloud computing [6]. To this end, we have categorized two main properties to specify the variable allocation of cloud resources required by the cloud process providers which are: (i) the desired resources and their properties w.r.t the structure of cloud resource management API provided by the *Open Cloud Computing Interface* (OCCI)[2] and (ii) the desired resource behavior in terms of shareability and elasticity. Therefore, in this section we identify the three main operators related to the configuration of the resource properties and their behavior, which are, (i) configurable resource assignment operator, denoted as A^c (section III-A), (ii) configurable resource elasticity operator, denoted as E^c (section III-B) and (iii) configurable resource sharing/batching operator, denoted as $(S/B)^c$ (section III-C).

A. Configurable Resource Assignment Operator

The A^c operator enables a cloud tenant to model a variable number of resources allocated to its process activity. It possesses two configuration parameters: (i) the configurable type that can be a configurable inclusive choice (OR^c), a configurable exclusive choice (XOR^c), or a configurable parallel flow (AND^c), and (ii) the range which specifies the minimal and maximal number of the resources that can be allocated from each resource type (i.e., compute, storage, and network). In our example in Figure 4, three resources $network2$, $network1$ and $compute3$ are linked to activity $a1$ through an OR^c. In this way, a tenant may configure the OR^c to an XOR associated to $network2$ and $network1$ in order to specify that either $network2$ or $network1$ can be allocated to activity $a1$. The allocation decision between $network1$ and $network2$ is then left to the run-time depending on the environment requirements, availability of the resources, etc.

B. Configurable Resource Elasticity Operator

The E^c operator allows the modeling of the variability at the elasticity level that a cloud tenant may require regarding the anticipation of the workload of its activities. It has two configuration parameters: (i) the configurable type used to model the set of resources to be elastic and (ii) the elasticity type (i.e. vertical, horizontal, or hybrid). Returning to our example in Figure 4, either of the $network1$ or $network2$ can be elastic (they are connected through an XOR^c). A tenant may configure the XOR^c to a XOR in order to specify that only $network2$ or $network1$ can be elastic.

C. Configurable Resource Sharing/Batching Operator

The S/B^c enables the modeling of the variability regarding the shareability property of resources. A resource can be shared between different activities (i.e. shareable), or between different activity instances (i.e. batch). This operator has two configuration parameters: (i) the configurable type used to model the number of instances/activities that share the resource, (ii) the shareability type (i.e. in a shareable, batch

or hybrid manner). For instance in Figure 4, an AND^c is used to connect the resource *compute*3 to the activities *a*1, *a*13 and *a*14. Since an AND^c can be only configured to an *AND*, the tenant can specify that only *a*1 and *a*13 can share simultaneously *compute*3.
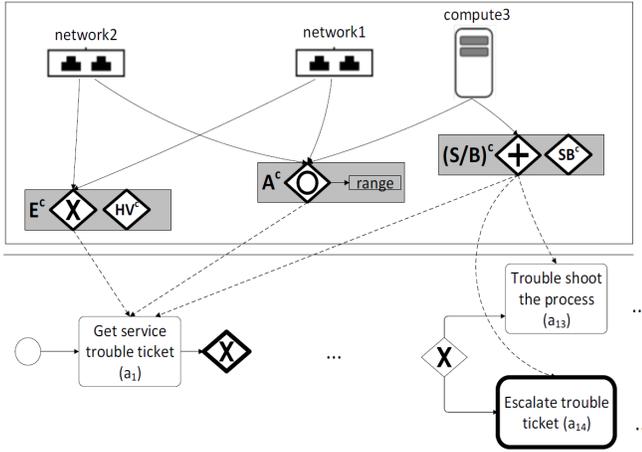

Fig. 4: Configurable Resource Operators

D. Cloud Simulation Tools

To develop and analyze new cloud environment with the help of the simulators, it is first required to understand the existing landscape of cloud simulators along with their pros and cons. Such a study, will allow us to define the different resource that can be allocated to the business process.

In the literature, many cloud simulators exist such as, iCanCloud [7], Green-Cloud [8], CloudSim [9], SimGrid [10], GridSim [11] and many others. However, due to variety of characteristics involved in cloud computing as shown in Table I, one particular existing cloud simulator clearly isolates the multi-layer cloud abstractions i.e. SaaS, PaaS, and IaaS. This cloud simulator is called CloudSim[3] [9]. It supports more functionalities as compared to other simulation tools, along with being extensible. Additionally, CloudSim is an open source framework, completely written in Java and there having many extensions being developed and published over years such as, [12], [13], [14]. Furthermore, in [15] the author performed a critical evaluation on various cloud computing simulators and concluded that CloudSim is the best choice for doing research related to study of performance evaluation of cloud resources via simulation.

Simulator	Programming Language	Extensible	IaaS	PaaS	SaaS
iCanCloud [7]	C++	Yes	Yes	HPC	No
GreenCloud [8]	C++, OTcl	No	Yes	No	No
CloudSim [9]	Java	Yes	Yes	Yes	No
SimGrid [10]	C	Yes	Yes	No	No
GridSim [11]	Java	Yes	Yes	No	No

TABLE I: Simulator Evaluation

[3]http://www.cloudbus.org/cloudsim/

Furthermore, CloudSim uses six main concepts for expressing cloud resources, they are , (i) *Datacenter*, (ii) *Host*, (iii) *Vm*, (iv) *Pe* (Processing Element), (v) *Cloudlet*, and (vi) *Storage*. In our work, we make use of the same terminology for expressing cloud resources.

IV. METHODOLOGY

In this section, we describe the approach used in our work for simulating the deployment of business process variants in a cloud provider using CloudSim. We provide a Unified Description Model (UDM) containing relationship between the activities, operators, and cloud resources. This UDM is parsed to create various business process variants. These variants along with their allocated resources are then simulated in the extended version of CloudSim developed by us. Our approach will enable practitioners to analyze the cost and execution time for the best cloud resource allocation policy and the best deployment scenario for each process variant.

A. Unified Description Model

In this section, we present *Unified Description Model* (UDM) which describes cloud resource allocation for a configurable business process. We will illustrate this model using an example shown in the Figure 5. The corresponding UDM is shown as an eXtensible Markup Language (XML) in Listing 1. The details about the resource allocation for the activities shown in the Figure 5 are provided in section IV-B2.

The UDM is divided into three parts, (i) activities, (ii) operators, and (iii) cloud resources.

1) Activities: All activities have a mutual dependencies in their relation. For instance, in Listing 1, lines (4-9) depicts that the activity *a*1 has a sequential dependence with activities *a*2, *a*4, *a*10, *a*11. In the UDM each activity has a unique identifier and requested capacity (such as, CPU defined in terms of millions of instructions per second (MIPS) rating, memory size, bandwidth etc). In practice executing these activities shall consume cloud resources offering atleast the requested capacity. These activities are connected to resources with operators that describes their behavior.

2) Operators: Operators express the behavior of the desired resource. As described in section III, we identify three main operators related to the configuration of the resource properties and their behavior: *Assignment*, *Elasticity*, and *Shareability*. For each operator a configurable type is assigned. These configurable types can be either a configurable OR^c, a configurable AND^c or a configurable XOR^c. Each operator takes a set of activities as input. For example, in Listing 1, line 31-34, Share operation has *a*10 and *a*11 activities as input. It is configurable with *AND* operator and has a *sharing* type.

3) Cloud Resource: Cloud resource allocation takes into account two main parameters, (i) the desired resource and (ii) the operators. The desired resources are the resources used by CloudSim tool. In this paper, we consider the following resources offered by CloudSim: (i) Processor (*Pe*) characterized by millions of instructions per second (MIPS), (ii) Storage characterized by its total size capacity, (iii) Virtual

machines (*VMs*) characterized by size, MIPS, number of *Pe* (PesNumber) and memory (ram), and *Network* characterized by its topology file from which the latency network traffic is simulated.

The cloud resources are configurable through operator A^c (Section III-A). Each resource has a set of output identifiers corresponding to the operators. The resources can be *elastic* and/or *sharable*. For instance, in Listing 1, in line 41-54, the VM, Storage and Processor (CPU) are configurable (connected with assignment operator id=23). In line 46-53, both VM (id=33) and Processor (id=32) are *sharable* (connected with share operator id=22), while at line 42, the Storage (id=31) is *elastic* (connected with id=21).
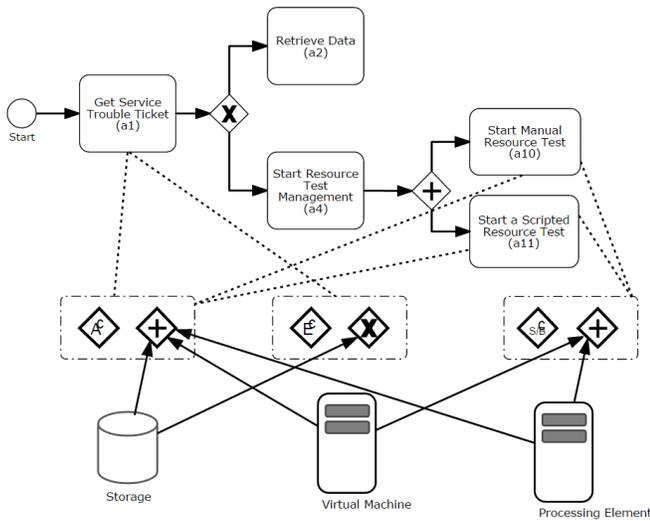
Fig. 5: Process Fragment showing Cloud Resource Allocation

```
<?xml version="1.0" encoding="UTF−8"?>                              1
<RacineFile>                                                        2
<Activites>                                                         3
<a1 id="1" size="200" mips="100,1000" ram="10" BW="15" mem="50,75"> 4
        <a2 id="2">Sequentiel</a2>                                  5
        <a4 id="4" >Sequentiel</a4>                                 6
        <a10 id="10" >Sequentiel</a10>                              7
        <a11 id="11">Sequentiel</a11>                               8
</a1>                                                               9
<a2 id="2" mips="200,1000" ram="10" BW="25" mem="50,100">           10
        <a4 id="4">Conditionnel</a4>                                11
        <a10 id="10">Conditionnel</a10>                             12
        <a11 id="11">Conditionnel</a11>                             13
</a2>                                                               14
<a4 id="4" mips="300,1000" ram="15" BW="15" mem="20,100">           15
        <a2 id="2">Conditionnel</a2>                                16
        <a10 id="10">Sequentiel</a10>                               17
        <a11 id="11">Sequentiel</a11>                               18
</a4>                                                               19
<a10 id="10" mips="200,1000" ram="20" BW="65" mem="40,100">         20
        <a11 id="11">Parallel</a11>                                 21
</a10>                                                              22
<a11 id="11" mips="150,1000" ram="10" BW="25" mem="15,90">          23
        <a10 id="10">Parallel</a7>                                  24
</a11>                                                              25
</Activites>                                                        26
<Operateur>                                                         27
<Elasticity id="21" configurable="yes" name="Elasticity" type="H" variantes=  28
    "XOR">
<in>1</in>                                                          29
</Elasticity>                                                       30
<Share id="22" configurable="yes" name="Share" type="sharing" variantes=      31
    "AND">
<in>10</in>                                                         32
<in>11</in>                                                         33
</Share>                                                            34
<Assignment id ="23" configurable="yes" name="Assignment" variantes=          35
    "AND">
<in>1</in>                                                          36
<in>10</in>                                                         37
<in>11</in>                                                         38
</Assignment>                                                       39
</Operateur>                                                        40
<Ressources>                                                        41
<Storage id="31" name="Storage" sizeTotal="100">                    42
<out>21</out>                                                       43
<out>23</out>                                                       44
</Storage>                                                          45
<Processor id="32" name="Processeur" mips="60">                     46
<out>23</out>                                                       47
<out>22</out>                                                       48
</Processeur>                                                       49
<VM id="33" mips="1000" name="VM" size="600" PesNumber="900" ram="            50
    512">
<out>23</out>                                                       51
<out>22</out>                                                       52
</VM>                                                               53
</Ressources>                                                       54
</RacineFile>                                                       55
```

Listing 1: Unified Description Model based on Figure 5

B. Simulation of Configurable Process Models

Figure 6 summarizes the steps involved in the simulation of a configurable process model. The CloudSim tool takes the UDM as input. This model represents the configurable process model. From this unified description model, various process variants can be generated. All the possible variants are stored in a database so that CloudSim can search in the database and retrieves the variant for simulation. According to the operators (*AND*, *OR*, *XOR*) in the process variant, various instances can be further generated. CloudSim is responsible for parsing these variants in order to generate the instances. Afterwards, CloudSim parses the variants again to configure a resource allocation for the selected process.

However, CloudSim is not designed for managing business process models. Additionally, CloudSim lacks the ability for supporting *elasticity* operator. Thus, to create a configurable resource allocation for the process variants, we need to extend CloudSim for including these abilities to it.
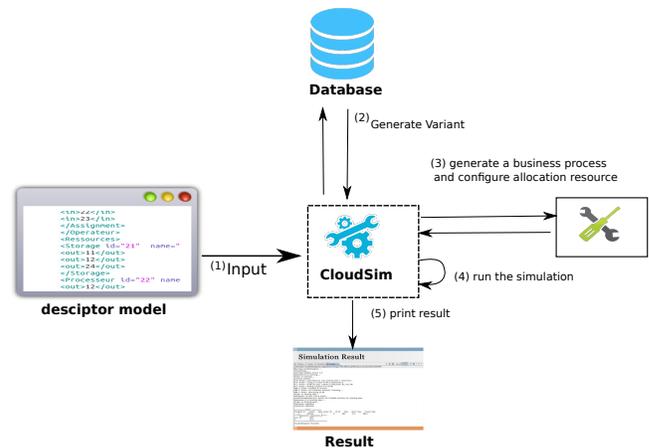

Fig. 6: Simulation of Configurable Process Models

In the following sections, we detail how CloudSim constructs the process and how it configures the resource allocation for the simulation purposes.

1) Construction of Business Process Variants: In order to generate a business process variant from the description model, we extend CloudSim for generating all the possible variants from the configurable process model i.e. the UDM. These process variant models shall be stored so that they can be used to create the process instances.

Once the variant is selected, CloudSim constructs a process instance by making use of the dependencies between the activities. These dependencies manage the scheduling of the workflow processing. According to the configurable operator between activities, the decision for the branching could be:

- *Sequential*: the execution of one activity follows the execution of another activity. This pattern is represented in CloudSim by *SpaceShare* policy
- *Conditional*: only the branches that satisfy the predetermined criteria are executed. Depending on the number of branch chosen, this pattern can represented in CloudSim by *TimeShare* or *SpaceShare* policies
- *Concurrent*: all activities in the same branch are executed concurrently. This pattern is represented in CloudSim by *TimeShare* policy.

Parsing the UDM allows us to construct the process variant for simulation purposes. In listing 1 (corresponding to process fragment shown in Figure 5), many instance for simulation can be generated. For example, $< a1, a2, \text{etc}>, < a1, a4, a10, a11, \text{etc} >$, etc.

2) Resource Allocation: For simplicity, we consider an activity as a finer abstraction of an application service being hosted in the virtual machine (VM). Plus, all VMs are hosted in the same Host.

To clearly illustrate the resource allocation we use Figure 5 that shows a simple *Storage*, *VM* and *Pe* provisioning scenario. In this figure, a host with one CPU core receives request for hosting one *VM*. This host has a hard drive storage. All resources are connected to the configurable resource assignment operator A^c. *VM*, *Pe* and *Storage* are connected to *sharable* operator through AND^c. In addition, *Storage* is *Elastic* through XOR^c.

Suppose that we simulate the instance $< a1, a4, a10, a11 >$ at first. The activities $a10$ and $a11$ are connected to the cloud resource *Pe*, *Storage* through AND^c. A tenant may configure the AND^c to an *AND* associated to *Pe* in order to specify that the *Pe* is shared between $a10$ and $a11$. Since $a1$ and $< a10,a11 >$ are executed sequentially, we configure CloudSim to share *Pe* between $a1$ and $< a10,a11 >$ in $space-shared$ mode. Thus, the activities $a10$ and $a11$ are put in a queue until the execution of $a1$ is complete. In the same context, $a10$ and $a11$ activities share the *Pe* concurrently. Thus, in CloudSim, we configure resource (*Pe*) in $time-shared$ policy.

Moreover, we consider a process will not go into infinite loops. Thus, in a process instance having a loop, we consider it to undergo a random number of iterations.

V. EXPERIMENTATION

In order to evaluate the usefulness and effectiveness of our approach, we performed experiments using a real dataset of configurable business process from our industrial partner, *Orange*, a French telecommunication provider. Different variants of business process for Voice over Internet Protocol (VoIP) are defined and used by *Orange*. These variants and their allocated cloud resources are created manually and described separately.

In section I, we already established the need of performing simulations over the use of real cloud. However, to illustrate the advantage of conducting the simulation of configurable resource allocation and the usefulness of CloudSim tool, we configured the process illustrated in Figure 1 into two variants i.e. Figure 2 and Figure 3 (discussed in section II). The first variant is represented by Figure 2, wherein the resources are allocated sequentially. While, in the second variant i.e Figure 3, the resources are allocated and consumed concurrently.

The experiments were conducted following the methodology described in section IV, i.e. (i) parsing the UDM to create business process variants, (ii) allocating CloudSim resources to the process activities, (iii) changing the capacity associated with the activity in terms of MIPS and memory size, and (iv) computing execution time and the cost of cloud resource consumption.

The experiments were conducted on a Intel(R) Core(TM) machine having the following configuration: 2.00GHz CPU with 8 GB of RAM running a standard Ubuntu Linux version 14.04 and JDK 1.8. Additionally, we adjusted the CloudSim properties as follows: architecture as x86, operating system as Linux, cost of usage of storage resource as $0.001 and the cost of usage of memory resource as $0.05. As said before, for each experiment, we manually changed the resource capacity for some activities and observed the change in performance based on execution time and costs. Furthermore, for in our current experimentation setup, we assume that all the resources are present in the same datacenter. In other words, we do not treat the multi-cloud allocation scenario. Likewise, we do not perform a complete load simulation in the current experimentation setup and we consider it as one of our future work. The associated UDM and the source code of this implementation will be available at our lab's URL[4].

A. Results Analysis

The experimentation results are based on a series of experiments performed that were conducted in two ways as explained in Section V-A1 and Section V-A2.

1) First experiment: In this experiment, we execute all process instances that were configured from the *same process variant* based on different cloud configurations. Then, for each of these process instances we compute the storage cost and the execution time of the instance. Table II presents the simulation result for all the instances that were configured from *Process Variant 1* as shown in Figure 2. Overall, we were able to configure six unique process instances were from the Process

---

[4]http://www-inf.it-sudparis.eu/SIMBAD/tools/simulprocessconfig

Variants 1		Resource			Capacity		Result	
	Activities	Pe	VM	Storage	Mips	Storage	Max. Cost	Exec. Time
instance₁	a1	-	-	x	100 000	1000	$0.021	200 μs
	a2	x	x	-				
	a5	x	x	-				
	a13	-	-	-				
instance₂	a1	-	-	-	100 000	2000	$0.042	200 μs
	a2	x	x	-				
	a7	x	x	-				
	a13	-	-	x				
instance₃	a1	-	-	x	500 000	1000	$0.021	1 ms
	a3	x	x	-				
	a8	x	x	-				
	a13	-	-	x				
instance₄	a1	-	-	x	500 000	2000	$0.042	1 ms
	a3	x	x	-				
	a9	x	x	-				
	a13	-	-	x				

TABLE II: Simulation result of configurable resource allocation

Variant 1. In Table II, for each instance, we specify which resource is allocated for which activity, the resource capacity allocated to these activities and provide the cost of resources and the execution time.

The first two instance (instance₁ and instance₂) have two activities that consume a processor (Pe) over the VM. Indeed, the activities $< a2, a5 >$ and $< a2, a7 >$ allocate one VM and one Pe. Each activity executes 100,000 instructions. Since the execution are made sequentially, the total execution time of these activities is 200 μs. For the memory cost, instance₁ needs 1000 *mo* of memory capacity, while instance₂ need 2000 *mo*. Since each Storage has a capacity of 1000 *mo* and the Storage is elastic, a new Storage was created for instance₂. In this case, the execution of instance₁ costs $0.021 while instance₂ costs $0.042. Afterward, we increased both the size of instruction and the activities that use Storage. In both instances, instance₃ and instance₄, two of their activities need Storage and execute 500,000 instructions. The activities are executed sequentially. Thus, CloudSim will put activities a8 and a9 in the queue until the execution of a3 is completed. The overall execution time is proportional to the number of instructions executed by the activities. However, we remark that the storage cost does not increase as compared to instance₁ and instance₂, even if in instance₃ and instance₄ two activities (a1 and a13) consume memory while only one activity was consuming storage in instance₁ and instance₂. This is due to the sequential execution. Indeed, the same Storage resource was re-used and a new storage resource was not are created, thus bringing down the costs.

2) The second experiment: In this experiment, we execute the *same resource configuration* over all process instances configured from the two process variant. In this case, we observed the result of each variant and made a comparison between them. Table III presents the total execution time and the maximum storage cost for all instances configured from variant 1 and variant 2, using three different cloud resource configuration. In variant 2, (for storage 2000) a10 and a11 activities are executed concurrently. For these activities, CloudSim uses Time-Share policy (see section III-D). Since the activities are executed concurrently, the total execution

time is reduced as compared to variant 1. In this case, variant 2 will be chosen by the user as it is better as the simulation shows compared to variant 1. In the other hand, the memory cost increases in variant 2, since three activities allocate memory resource (a1, a11 and a13). Therefore, based on memory requirement, variant 1 will be chosen by the user. However, when the simulator executes variant 2 and runs concurrently 500,000 instructions for each activity, it exceeds the total capacity of the CPU core and the execution fails. Indeed, as the CPU core is not elastic in our experiment thus, it proves that a proper resource allocation is a must in the cloud environment.

From these results, we can observe the advantages of performing simulations. Through simulations, the user will have the possibility to estimate the cost and performance before deploying the processes on the real cloud. In this way, it will reduce the deployment costs and help in avoiding failures. In addition, the user will be able to choose the optimal or the best configurable resource allocation for business process variant according to their specific criteria.

	Capacity		Result	
	Mips	Storage	Max. Cost	Exec. Time
Variant 1	100 000	1000	$0.021	100 μs
Variant 2	100 000	1000	$0.021	80 μs
Variant 1	200 000	2000	$0.021	200 μs
Variant 2	200 000	2000	$0.084	160 μs
Variant 1	500 000	2000	$0.048	500 μs
Variant 2	500 000	2000	-	failure

TABLE III: Simulation result of configurable resource allocation

VI. RELATED WORK

In literature various research studies have been addressing the topic related to the deployment of service-based business processes in cloud environments. Authors in [16], [17] have put forward the benefits and drawbacks of blending Business Process Management (BPM) and Cloud Computing. However, not much work has been done around the handling of resource perspective in BPM as compared to the control-flow perspective. Moreover, much of the research work done in terms

of resource perspective has been focused on human resource behavior and thier allocation in context of BPM [18], [19], [20], [21]. For instance, in [22], [23] the authors highlight the value of human resources as a social compute unit to resolve incidents in IT service organization. While authors in [24] propose a resource-efficient scheduling algorithm for business processes and cloud-based computational resources. A complete graphical notation for the assignment of human resources to process activities is introduced by authors in [25]. Contrarily to our work, these proposals lack the work for supporting the allocation of cloud resources to process activities w.r.t cloud properties such as the elasticity and shareability, and simulating such the resource behavior in cloud platforms.

Additionally, configurable process modeling approaches have been proposed as a way to model the variability in process models and allow for configuration facilities [26]. They represent an efficient solution for modeling multi-tenant business processes [4]. Many configurable process modeling languages [3], [27], [28], [29] have been proposed for enabling customizable process constructs. However, these works have focused on the variability modeling in the control flow perspective and overlooked the resource perspective. Recently, some works have addressed the configuration of the resource perspective in business processes [30], [31], [32]. However, they studied the resources in a generic way and did not address the specificity of resources in cloud environments. For instance, La Rosa et al. [30] have proposed the configurable integrated EPC (C-iEPC) with features for capturing resource, data and physical objects. They realize the configuration of these elements using configurable connectors borrowed from the control-flow perspective aiming to model the variable resource allocation. Contrarily to our approach, they focus on human resources and do not support cloud resources behavior for e.g., resource sharing and resource elasticity. Moreover, in our work, we also make use of configurable connectors to model the variability in cloud resources allocation but we adapt these to the cloud context with taking into account the specificity of the cloud features.

However, over the past few years there have been some studies for formalizing and modeling cloud resources in context of cloud-based business processes. In [33] the authors provide a formal definition for ensuring a correct and consistent cloud resource allocation in business process modeling using Event-B based formalism. In [34] the authors formalized temporal constraints on cloud resources used in process activities in cloud-based business processes. In [35] the authors provide a formalization to describe cloud resources in context of social business processes. In [36] the authors describe the benefit of explicitly modeling cloud resources along with their energy efficient properties for configurable business processes. In [37] the authors provided a linear programming technique to find optimal process variants from configurable process model deployed on a cloud federation which is a good technique for NP-complete problems. Contrarily to our work, these proposals lack the work for supporting generalized scenarios of cloud resource allocation to process activities and to simulate the behavior of process variants in cloud infrastructure.

VII. Conclusion and Future Work

A configurable process model needs to be configured to derive individual process variants that suits the specific requirements of an organization. However, applying these approaches to the Cloud Computing domain can be a challenging task. Specifically, as the mechanism to configure and integrate cloud resources has been hardly considered in configurable process modeling. Additionally, it is not cost effective to make use of real cloud infrastructure for conducting experimentation to find the optimal cloud resource allocation policy. Not to forget, the creation of tests on real infrastructure involving frequent changes, which is a time consuming task. Thus, in this paper, we bridge the gap by proposing a Unified Description Model (UDM) that allows to explicitly model the resource allocation alternatives in context of the multi-tenant process models, including concepts of elasticity and shareability. We then studied the existing cloud simulator landscape to find CloudSim toolkit as the most suitable tool for our work. However, as CloudSim lacks the ability to bind activities with their resources, we extended CloudSim for introducing two main features, (i) to construct a configurable process model: parsing the description model, extracting the activities and their capacity and constructing the multi-tenant process models, and (ii) allocate resource for activities: according to the operators type, each activity consume one or more resources

We then performed simulation experiments and computed the performance in terms of time and the cost of the allocated cloud resources. The experimentation results showed that our approach allows finding of the best configuration of resources to be allocated to the configurable process model.

As our future work, we will extend our study to support other kinds of cloud resource such as, network and to integrate this approach in widely used process modeling tool such as, Signavio. It will allow the users to graphically model a configurable resource allocation and to run the simulation using a simple clic. We also plan to perform load simulation for business processes running on cloud using CloudSim. Another direction of our future work is to extend the configurable resource allocation to multi-Cloud scenario. This will allow organizations to find best provider for hosting their processes along with finding the best configuration of resource allocation.

ACKNOWLEDGEMENTS

References

[1] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.
[2] F. Gottschalk, W. M. van der Aalst, and M. H. Jansen-Vullers, "Configurable process modelsa foundational approach," in *Reference Modeling*. Springer, 2007, pp. 59–77.

[3] M. Rosemann and W. M. van der Aalst, "A configurable reference modelling language," *Information Systems*, vol. 32, no. 1, pp. 1–23, 2007.

[4] W. M. Van Der Aalst, "Business process configuration in the cloud: How to support and analyze multi-tenant processes?" in *Web Services (ECOWS), 2011 Ninth IEEE European Conference on*. IEEE, 2011, pp. 3–10.

[5] 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.

[6] E. Hachicha, N. Assy, W. Gaaloul, and J. Mendling, "A configurable resource allocation for multi-tenant process development in the cloud," in *International Conference on Advanced Information Systems Engineering*. Springer, 2016, pp. 558–574.

[7] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente, "icancloud: A flexible and scalable cloud infrastructure simulator," *Journal of Grid Computing*, vol. 10, no. 1, pp. 185–209, 2012.

[8] D. Kliazovich, P. Bouvry, Y. Audzevich, and S. U. Khan, "Greencloud: a packet-level simulator of energy-aware cloud computing data centers," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–5.

[9] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities," in *High Performance Computing & Simulation, 2009. HPCS'09. International Conference on*. IEEE, 2009, pp. 1–11.

[10] H. Casanova, A. Legrand, and M. Quinson, "Simgrid: A generic framework for large-scale distributed experiments," in *Computer Modeling and Simulation, 2008. UKSIM 2008. Tenth International Conference on*. IEEE, 2008, pp. 126–131.

[11] R. Buyya and M. Murshed, "Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," *Concurrency and computation: practice and experience*, vol. 14, no. 13-15, pp. 1175–1220, 2002.

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

[13] M. Bux and U. Leser, "Dynamiccloudsim: Simulating heterogeneity in computational clouds," *Future Generation Computer Systems*, vol. 46, pp. 85–99, 2015.

[14] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*. IEEE, 2010, pp. 446–452.

[15] A. Bashar, "Modeling and simulation frameworks for cloud computing environment: A critical evaluation," in *International Conference on Cloud Computing and Services Science*, 2014, pp. 1–6.

[16] E. Duipmans and D. L. F. Pires, "Business process management in the cloud: business process as a service (bpaas)," *University of Twente*, 2012.

[17] M. Wang, K. Y. Bandara, and C. Pahl, "Process as a service distributed multi-tenant policy-based process runtime governance," in *Services computing (scc), 2010 ieee international conference on*. IEEE, 2010, pp. 578–585.

[18] C. Cabanillas, A. Norta, M. Resinas, J. Mendling, and A. Ruiz-Cortés, "Towards process-aware cross-organizational human resource management," in *Enterprise, Business-Process and Information Systems Modeling*. Springer, 2014, pp. 79–93.

[19] E. Kajan, N. Faci, Z. Maamar, A. Loo, A. Pljaskovic, and Q. Z. Sheng, "The network-based business process," *IEEE Internet Computing*, vol. 18, no. 2, pp. 63–69, 2014.

[20] N. N. Chan, W. Gaaloul, and S. Tata, "Assisting business process design by activity neighborhood context matching," in *Service-Oriented Computing - 10th International Conference, ICSOC 2012, Shanghai, China, November 12-15, 2012. Proceedings*, 2012, pp. 541–549. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-34321-6_38

[21] ——, "Context-based service recommendation for assisting business process design," in *E-Commerce and Web Technologies - 12th International Conference, EC-Web 2011, Toulouse, France, August 30 - September 1, 2011. Proceedings*, 2011, pp. 39–51. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23014-1_4

[22] M. Z. Candra, H.-L. Truong, and S. Dustdar, "Provisioning quality-aware social compute units in the cloud," in *International Conference on Service-Oriented Computing*. Springer, 2013, pp. 313–327.

[23] B. Sengupta, A. Jain, K. Bhattacharya, H.-L. Truong, and S. Dustdar, "Who do you call? problem resolution through social compute units," in *International Conference on Service-Oriented Computing*. Springer, 2012, pp. 48–62.

[24] P. Hoenisch, S. Schulte, and S. Dustdar, "Workflow scheduling and resource allocation for cloud-based execution of elastic processes," in *Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on*. IEEE, 2013, pp. 1–8.

[25] C. Cabanillas, D. Knuplesch, M. Resinas, M. Reichert, J. Mendling, and A. Ruiz-Cortés, "Ralph: a graphical notation for resource assignments in business processes," in *International Conference on Advanced Information Systems Engineering*. Springer, 2015, pp. 53–68.

[26] M. La Rosa, W. M. van der Aalst, M. Dumas, and F. P. Milani, "Business process variability modeling: A survey," 2013.

[27] R. Mietzner and F. Leymann, "Generation of bpel customization processes for saas applications from variability descriptors," in *Services Computing, 2008. SCC'08. IEEE International Conference on*, vol. 2. IEEE, 2008, pp. 359–366.

[28] F. Gottschalk, W. M. P. V. D. Aalst, M. H. Jansen-Vullers, and M. L. Rosa, "Configurable workflow models," 2008.

[29] D. Ciuksys and A. Caplinskas, "Reusing ontological knowledge about business processes in is engineering: process configuration problem," *Informatica*, vol. 18, no. 4, pp. 585–602, 2007.

[30] M. La Rosa, M. Dumas, A. H. Ter Hofstede, and J. Mendling, "Configurable multi-perspective business process models," *Information Systems*, vol. 36, no. 2, pp. 313–340, 2011.

[31] A. Kumar and W. Yao, "Design and management of flexible process variants using templates and rules," *Computers in Industry*, vol. 63, no. 2, pp. 112–130, 2012.

[32] A. Hallerbach, T. Bauer, and M. Reichert, "Capturing variability in business process models: the provop approach," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 22, no. 6-7, pp. 519–546, 2010.

[33] M. Graiet, A. Mammar, S. Boubaker, and W. Gaaloul, "Towards correct cloud resource allocation in business processes," *IEEE Transactions on Services Computing*, 2016.

[34] 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.

[35] E. Hachicha, W. Gaaloul, and Z. Maamar, "Social-based semantic framework for cloud resource management in business processes," in *Services Computing (SCC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 443–450.

[36] E. Hachicha, K. Yongsiriwit, and W. Gaaloul, "Energy efficient configurable resource allocation in cloud-based business processes (short paper)," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2016, pp. 437–444.

[37] 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 *Services Computing (SCC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 34–41.