IJICTR
International Journal of Information & Communication Technology Research

Volume 10- Number 3 – Summer 2018 (50 -59)

Efficient Policy-Aware Deployment of Business Process in Cloud Federation

Hamidreza Nasiriasayesh
Information Technology (IT) Department
ICT Research Institute (Iran Telecom Research Center (ITRC))
Tehran, Iran
hr-nasiri@itrc.ac.ir

Alireza Yari *
Information Technology (IT) Department
ICT Research Institute (Iran Telecom Research Center (ITRC))
Tehran, Iran
a_yari@itrc.ac.ir

Received: 9 October, 2017 - Accepted: 13 March, 2018

***Abstract*—The use of cloud computing to implement business processes is becoming increasingly important because users can benefit from the economic and technical advantages of this technology. The concept of the Business Process as a Service (BPaaS) is a new solution in the use of specific business processes as a medium for aligning information technology and business. However, managing and deploying business processes on heterogeneous Cloud providers is still a challenge for organizations due to interoperability concerns. This paper suggests an algorithm for optimizing the resource allocation of the business process in Extended Federated BPaaS model in accordance with the requirements of the user's policy. The developed model has been compared with other popular models supporting the service/business process policy and shows that the proposed model can effectively execute business processes with regards to infrastructure and data transfer costs. To evaluate the model, we implemented this algorithm on WorkflowSim tool, taking into account the challenges of data transfer in the cloud federation environment. The results of the experiments highlight the algorithm's efficiency compared with initial deployment in a way that the proposed optimization mechanism reduces the cost of transferring data between clouds and can improve performance of workflow execution in cloud federation in terms of makespan time and cost.**

***Keywords*—BPaaS, Cloud Federation, Policy, Business Process, Resource Allocation, Optimization**

I. INTRODUCTION *

A business process or a business workflow is known as a high-level model for a set of business activities. One of the main characteristics of the business process is that it is based on the business rules of the organization. The business process represents an end-to-end process of the organization, so whenever the dependency between tasks is met then the specific goal of the organization is achieved [1], [2].

Business Process Management (BPM) or WorkFlow Management (WFM) systems automatically support business processes. Different business processes and changes on them, due to changing market conditions, are inevitable because companies need to respond to market changes. As a result, organizations are trying to invest in significant projects for the development of business processes, resulting in high costs and also implementation time, which often prevents fast response to them [3], [4].

Due to the change in the process request patterns and in order to apply the requested processes, the business process management system is constantly changing. In private clouds, we have fluctuation in resource demanding leads either over-provisioning or under-provisioning. This is common especially when the process steps are deployed by the SaaS [5].

Complex business processes need variety of services that a single cloud often cannot afford. In addition, dissatisfied agreements between cloud providers and consumers may result from lack of availability of a cloud service.

Recently, a lot of research has been done on the migration of BPM systems to the cloud. However, existing implementations mainly address only single clouds. In other words, the implementation of process-based applications in a centralized environment with several clouds is still challenging [6].

Although there are cases in which the cloud cannot execute the business process as effective as the user requires, a process-oriented application in a cloud is only suitable for business processes that are not data and compute-intensive. For instance, some process-oriented tasks may have certain infrastructure requirements not supported by the cloud. Extensive scientific workflows have a large number of tasks that require processing and storage capacities beyond the cloud size [3], [7].

Users of business processes like any other cloud clients prefer to have the high performance and low cost. A federated-cloud BPaaS as a solution, which runs business processes using various clouds, is suitable in terms of quality and cost.

Therefore, the cloud federation is considered as the perfect solution as the next development of cloud computing. The Federation of Clouds offers timely capabilities and delivers the elasticity of applications so that QoS is guaranteed under the dynamic environment by changing workloads. This can provide support for elastic provisioning capabilities to address changes in user demand. Research focuses on the challenges of the federal cloud computing environment such as the Job scheduling [8], [9]–[12]

Federated cloud service providers that are linked to provide resilient resources to users which can reduce the challenges of Service Level Agreements (SLAs) [13], [14]. Therefore, the cloud federation can offer the following benefits:

i. performance guarantees;
ii. availability guarantees;
iii. convenience; and

* Corresponding Author

iv. dynamic workload distribution.

Using several clouds for workflows, is the issue of service combining [15], where services provided by different clouds are cooperate for a goal, here, executing tasks of a business process. Similarly, we call such multi Cloud Services as a composite service. The topic of research with cloud computing services is not only the choice of individual services, but also more challenges, the impact of service composition. Cloud selection for a workflow is a complex matching problem, in which both the software requirements on the infrastructure and the quality of services and properties of cloud providers, including service prices, availability and response time, should be taken into consideration. By using these parameters and in addition to the effect of the combination, it is not easy to decide which clouds to execute the workflow with the result of payment, with the specified cost, but with policy enactment assurance.

The expansion of the deployment of workflow in multiple clouds is typically limited to optimizing the assignment of tasks to predefined Clouds resources, with the aim of minimizing data transmission, runtime, and cost [15]. These approaches do not support Cloud Clouds SLA selection, which can also greatly affect performance and deployment costs. Enterprises face difficulties in identifying and selecting the process providers, as they should consider costs, vendor lock-in and security aspects.

Briefly, in this paper we have the following contributions:

- An extended framework to support the deployment of business process on a federated Cloud environment.
- Policy-based matching approach in federated clouds
- Optimization of workflow execution in the federated Clouds

II. RELATED WORK

In this section, we discuss the works related to the deployment of service-based processes as well as business processes in the cloud environment.

Cloud computing is a computing concept based on early technologies such as Grid and utility computing tools and support various service models: software as a service (SaaS), platform as a service (PaaS), infrastructure As a Service (IaaS) In this section of the paper, we describe the work related to Cloud Flow Management in various cloud types, and in particular the deployment of automated service deployment in heterogeneous clouds, and also cover policy models in service computing.

In Grid computing, it is essential that a type of coordinator is responsible for managing tasks between network sites because the network resources are heterogeneous [6]. This is known as the Workflow Management System (WfMS). The most important tasks of these systems are the description of the workflow, resource discovery, job planning, data transfer and monitoring. Comprehensive terminology and accurate comparison of WfM systems by Yu and Buyya are presented in[16].

In recent years, researchers have actively studied the scientific workflows in the field of distributed computing. Examples of workflow services like UNICORE[1], as well as the DAGMan component developed in the Consdor toolkit as an extension for the Globus toolkit[2] are middlewares for the purpose of integration of workflow engine[17].

Cloud computing has greatly expanded in managing business processes in grid computing. Although the cloud brings a lot of technical benefits such as provenance and elasticity to workflow developers, it creates new problems for the organization.

One of the common ways to set up business processes in cloud is using WfMSs, for example Kepler WfMS [17], [18], which has been expanded for the purpose of using Amazon's EC214 service. Article [19] suggests market-based cloud-based architecture. A broker as a Market-Maker is the main component of this architecture which acts as an interface between the user and the cloud resources, and an engine based on QoS requirements, provides tasks between cloud resources. Aneka platform using the EC2 service shows the proper performance of this architecture in implementing cloud-based workflows [20], [21].

The mOSAIC[3] project is designed to streamline cloud development, including workflow. It provides a framework that includes the Cloud Agency and the API. The cloud agency component meets the best resource settings that satisfy the SLA of the software. The mOSAIC API is a programming model that supports the cloud federation. Implementing an agent based on this framework is available and evaluated by actual programs and clouds. One of the reasons for this framework for workflow developers is that the program needs to be rebuilt so it can be compatible with the mOSAIC API.

Many approaches ensure that they support heterogeneous clouds, however their actual implementation only uses homogeneous cloud environments. In addition, these approaches often ignore the transmission of data and network-related communications between business process activities, while it is very important. In addition, the selection of cloud resources required to deploy a business process is often based on the requirements of SLA, while non-functional needs are ignored.

A business process is not a web service or business activity. In a business process, activities are connected through the flow control structure, and decisions in a service can lead to more changes or other actions in other services. The policy model needs to consider the business process and flow rather than just the service. For example, canceling a process is different from service cancellation.

The XACML models [22] and WS-Policy [23] are two prominent models as standard specifications. They can be considered in a family, because both focus on the limitation of services. In fact, they really do not accept the basic requirements mentioned above. More work on these models still covers the constraints aspect.

Article [24] offers an architectural solution to deal with the process as a service topic. This architecture includes an architecture style called Service Process Architecture (SPA) designed to expand the service-oriented architecture style. The SPA style defines the principle for developers to achieve customization and processing compatibility. This architecture consists of three main parts: the specification of the policy, the coordination framework and the enhanced AOP. Policy specifications will help users identify their business policy needs in the cloud environment. The purpose of the framework is to coordinate the use of policies defined for different users during process execution in the cloud. Finally,

[1] https://www.unicore.eu/

[2] https://www.globus.org/

[3] https://www.mosaic-fp7.eu/

the AOP has enhanced and is responsible for extensibility of the framework.

SPA architecture can expand service oriented architecture and affect SOA principles. Using this architecture, a comprehensive solution is presented for the process as a service, while this architecture is only applicable to clouds, however, the federal cloud considerations are not considered. In addition, implementation of this model is complicated, because consumers are responsible for overseeing compliance with business process policies.

In [25], the authors propose a pure Petri method to provide platform-based resources achieve their proper execution in clouds. While in our work, we focus on adaptive deployment of service-based processes with IaaS resources needed to meet the user policy requirements and QoS. In [26], authors challenge the deployment of business processes to a cloud infrastructure from the perspective of the process owner. The proposed approach consists of three main phases: forecasting the implementation of the process, allocating resources, and estimating the cost. Compared to our work, only the time for re-run and reuse is considered, while no strategy for collecting communication costs has been created. In addition, they consider deploying to a cloud provider and not responding to different demands for cloud computing resources.

In [27], the authors offer an adaptive resource allocation strategy to implement the process in the cloud environment. In fact, based on previous knowledge about the current and future trends, cloud resources are automatically allocated or released. The authors focus on avoiding the under and over provision, however, nothing about the reduction of communication costs in the process has been mentioned. In [28], a Mixed Integer Linear Programming (MILP) approach is used to ensure the cost-effective planning of an elastic process into a hybrid cloud. The proposed model considers data transfer to reduce the total cost of assistance. Also, in [29], authors suggest a linear program (LP) to determine a customizable business process to apply optimal methods to a cloud federation. The proposed model considers the cost of calculating and transferring data. However, in [30], the authors claim that the LP is not suited to resolving the deployment of large-scale applications in computing infrastructure. In fact, the amount of time and resources needed to solve the problem increases significantly with the size of a given workflow. Therefore, the discoveries are needed to provide a faster and efficient solution.

Given the above issues, the BPaaS federation model has been expanded. It is based on a component that allows users to select the most appropriate cloud resources based on their policy requirements, service level agreements, and costs. With the help of the Federal BPaaS, various clouds have been identified for implementing a business process with respect to policies. The proposed model is able to manage the movement of data between different tasks in the business process. In this work, the simulation results show that our model is successful in providing efficiency in terms of time and cost in executing the workflow. In addition, due to the high level of this model, it can be coordinated with other existing cloud infrastructures.

III. SAMPLE BUSINESS PROCESS SCENARIO

In this section, we will introduce a motivational scenario of the financial industry that is increasingly facing a huge volume of transactions and has currently an enormous transition in banks and financial institutions. The most important goal is to respond to the user's needs efficiently, which includes an optimization process for business processes.

We consider a large financial company with a variety of users and is made up of numerous business processes that include a variety of software-based services. These services provide short-term operations, but critical times like processing credit card transactions and billing, into long-time intensive computations such as suspicious modeling and risk analysis, and large-scale analytical processes for financial institutions such as insurance. These processes typically contain a large amount of data, require complicated parallel computing, and are often rarely needed. Keeping a private cloud for the computing capacity to run these models in particular or periodically can be expensive, especially when resources may be used for a large part of the time. The Federation provides the clouds needed for private clouds, helps them optimize their use and achieve significant performance, and allows them to respond to ambitious modeling risks and potentially responsive variations. (Such as accepting new regulatory requirements) faster.

There are many processes used in a wide range of domains in organizations. A shopping process can be used in retail applications. In many companies the recruitment process is used. Therefore, there is a great potential to share many service-based business processes between the parties.

The scenario shown in Figure 1 shows a general service-based process delivered by BPaaS providers. In fact, there are organizations looking for a business process, and we know them as process consumers. However, due to the differences in the trading policies of different organizations, the use of business processes is difficult for other organizations.

We use the service-based process shown in Figure 1. We are considering the process to be deployed in a federated environment.

All activities of the process are public service process and available on the Internet. These services belong to different organizations that are provided as SaaS. As shown in Figure 1, the business process consists of 9 activities, each supported by a public service provided by a cloud provider. Each policy of the activity is defined by the process client.

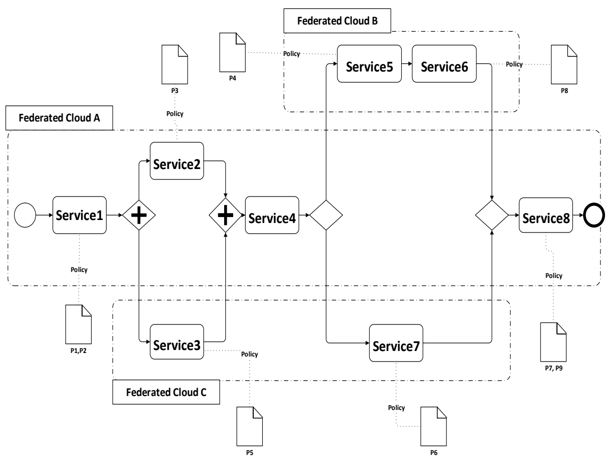

Fig. 1. Business Process in BPEL

As shown in Fig 1., the business process consists of 8 numbers of software services. Services deal with a large amount of data, as well as precise models that need to be transferred from one service to the next. This transition takes time to be considered by the BPMS. The enterprise maintains a fixed set of computing resources as a private cloud, which is enough to provide all software-based services at an off-peak time. At the peak times, when there are multiple parallel processes, it is necessary to lease additional resources or bring the process to the federal clouds to meet the needs of resources for services. Figure 1 provides a snapshot of the implementation of the federal cloud process that runs Service1, Service2, Service4, Service8 on Cloud A and Service3 and Service7 running in the federal Cloud C, and finally, Service5 and Service6 must be run in the Federated Cloud B, which are members of a cloud federation, as there are fewer resources in the private cloud. The capabilities to transfer data in a cloud is very good, so data transfer from user data only needs a short time. Conversely, the capabilities to transfer data between members of the federation is limited, because they are transmitted through the Internet. This transfer of data between the cloud also issues the cost of data transfer.

The process that is supposed to be deployed in a cloud must meet policies of the organization. Table 1 contains examples of policies that need to be implemented in the business process for the consumer of process.

TABLE I. SAMPLE BUSINESS POLICIES

Cloud Service	Policy	Sample Policy Description
Service1	P1,P2	- Transactions for any order from abroad are not allowed. - Orders less than 5 are not allowed.
Service2	P3	- Central Bank of Iran should process payment transactions.
Service3	P5	- If the card processing fails, the service should be retried. The maimum numbers of retry is 9 for the last 4 minutes and less than 3 for the last 60 seconds.
Service4	-	-
Service5	P4	-The processing time spent for the credit card should not be more than 650 ms.
Service6	P8	- Any business partner with low security (>6) can not receive the information of accounts for customers.
Service7	P6	- Any Order with total price over the 1800 euros is shipped without any charges.
Service8	P7,P9	- The system should issue a formal receipt for costomer - A notification is sent to the costomer due to the failure in the issuing receipt

The policy model in this work is similar to the XACML specification. In fact, three levels for the structure (Rule, Policy, Policy Set) are taken into consideration. By using this model, we can have nested policies for different scenarios. This is intended for many organizations, because this feature is essential for them to develop their own business policies. Since the rules are the main elements of a policy, the user must define the various categories of rules required to cover various aspects of business policy.

The goal is to avoid costly data transfer when adopting the process in line with the alignment of business process policies. Since the completion of the process is critical to the production process, each process is set to have a deadline, and when the execution of the process does not meet this deadline, the cost of the penalty could increase. The amount of process vision and the exact execution time of the process can also be found in a variety of areas, such as the financial industry. Therefore, work in hand can also be applied to other areas [31].

IV. EXTENDED FEDERATED BPAAS MODEL

A. Extended Federated BPaaS Model

The Extended Federated BPaaS is designed as a mediator between process consumers and cloud service providers. It is an extension of our previous model [32] .The Extended Federated BPaaS Model Architecture is illustrated in Fig. 2. The model is developed to help process consumers find the right cloud provider to execute their business processes and ensure that their corresponding policies are enacted. during the execution of the workflow process.

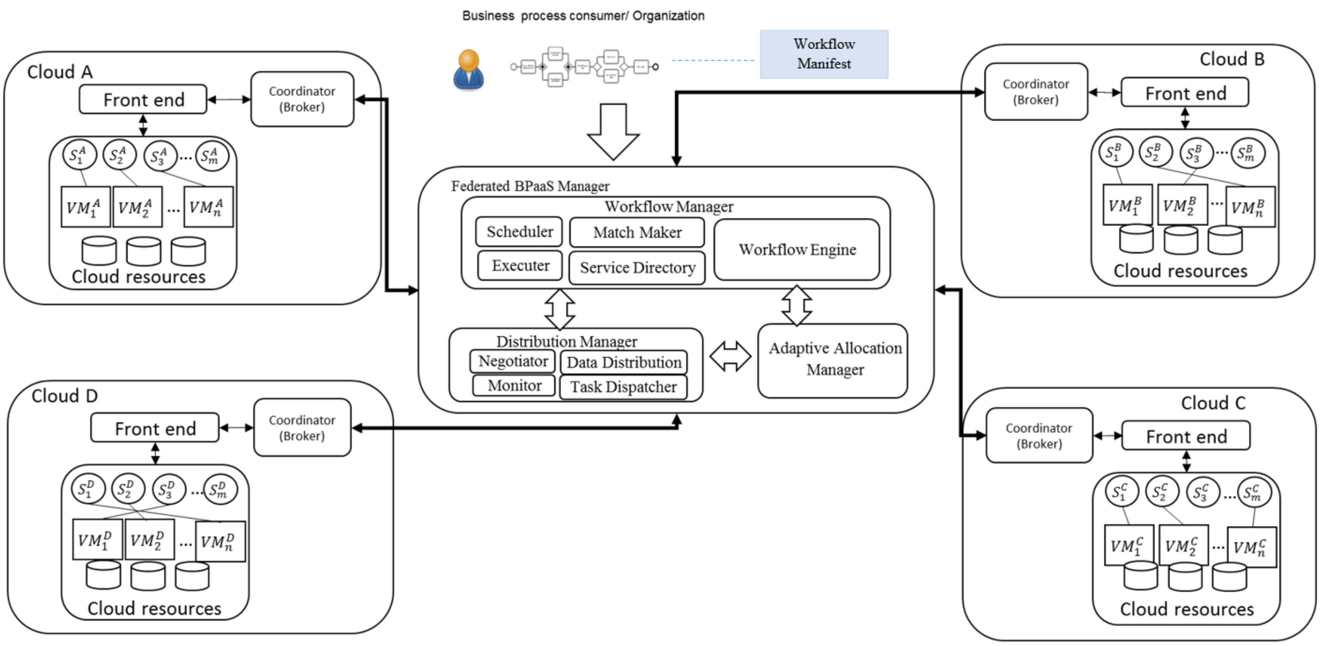

Fig. 2. Architecture of Extended Federated BPaaS Model

The workflow manifest is owned by the user of the business process and the key element in the Extended Federated BPaaS Model. It simply identifies the business process structure in terms of business activities (services) that are supposed to be deployed as services in the cloud federation. For each of these activities in this process, a reference is assigned to a policy that fully defines policies and QoS. Moreover, the manifest includes input and output information.
Main components of the Extended Federated BPaaS Model are as the following:

-Federated *BPaaS Manager (FBM)*:

This component is the heart of the model and is a business process management system. The responsibilities of this component are accomplished by three main components: 1) *Workflow Manager*, 2) Distribution *Manager*, 3) Adaptive Allocation *Manager*.

1) *Workflow Manager*:

Workflow Manager control and plan the execution of the business process. The Workflow engine receives the business process steps with the Service Level Objectives (SLO) in the form of a user request, and decomposes the business process. Then, Match Maker searches for the service directory to find services that match the workflow tasks to the services. This mapping is useful to find out what services are running, so it is distinguished from this traffic information manager to direct service call requests to the VM containing this service to allocate resources. In addition, the service call is limited by the QoS specified in the SLO.

The workflow executer begins to execute a workflow and its services. According to the workflow request, workflow / service planning, and query result from the distribution manager, the selection of services for the workflow stages is done. The response time of the service is measured by the executer. This function is to identify the deviation from the expected behavior at the time of execution, which results in penalties. The Workflow Manager logs service calls and response time. In case of deviation, the workflow manager can re-plan the execution of the workflow.

2) Distribution *Manager*:

The *Distribution Manager* component promises the implementation of policies determined through several components. The Monitor component collects information from the cloud coordinator. The information includes the use of resources in the cloud and the latest status of the task execution task. In the case of SLA compromise, the monitor component decides to transfer the service to another cloud within the federation. Negotiator component keeps the latest status of cloud resources. Some tasks require processing of input data. The data distribution component is responsible for coordinating the transmission of data to the tasks. The task distribution is done by the Task Dispatcher Component. This component delivers the tasks to the resources to be executed.

3) Adaptive Allocation *Manager*:

Adaptive Allocation *Manager* is responsible for the efficient allocation of resources to services. Whenever a service is required to be migrated from a VM in a cloud, the Adaptive Allocation *Manager* find the best destination VM in a cloud with respect to the minimum transfer of data between cloud as well as enforcing the business process policy compliance.

-*Cloud* Coordinator:

This component is shown in Figure 2 and is responsible for cloud management of the membership in the Federation of Clouds in an organizational context based on QoS by negotiating a contract. The cloud coordinator provides the planning environment, management, and deployment for business processes. The cloud service is transferred to the cloud community by implementing core resource management features such as scheduling, allocation, and monitoring.

The scheduling and allocation components will check if there are enough resources to perform tasks, and then allocate the appropriate VM based on the QoS workflow.

Policy Engine: The policy engine maintains the information for SLAs and policies. This component is responsible for accounting for the use of services. Users can introduce multiple QoS to choose more options for running business processes, but this complicates the finding of optimal resource allocation solutions. In the federation, the schedule, re-allocation and migration of VMs is required dynamically to respond to the SLA, so the cloud coordinator facilitates access to availability, rules, pricing rules, and SLAs.

The monitoring component is responsible for overseeing deployed services and their capacity adjustment, a number of VM instances and allocating their resources such as memory and processors to ensure that the SLA is consistent with and consistent with business policies.

B. Policy-Based Computing Approaches

Since policy-based computing is very important in the proposed model, it would be better to compare this model with other policy-based approaches in the area of systems that are dependent on service-oriented architecture. Table 2 provides a comparison of policy-based approaches such as WS-Policy, XACML, SPA approaches and Extended Federated BPaaS model. In the XACML approach, consumer policy requirements are not provided, so policies are defined only by process providers. On the other hand, in the WS policy approaches, only policies adopted by the two sides are applied. Furthermore, the WS policy approach focuses on services, not business processes. Process consumers are able to define their policies in the process in the approach of SPA. Providers can also set policies for internal users. Both XACML and SPA support policy implementation in a single cloud, but no one supports the implementation of policy in a few clouds, especially in the cloud federation environment. The proposed model is suitable for organizations because it supports business processes every day and effectively prevents vendor-lock in problem.

TABLE II. COMPARISON OF POLICY-BASED COMPUTING APPROACHES [32]

Policy-based Computing approach/model	Focus of Policy	Support of Policy monitoring for Provider	Support of Policy monitoring for Consumer	Single Cloud support	Federated Cloud support
XACML	Service	Yes	-	Yes	-
SPA approach	process	Yes	Yes	Yes	-
WS-Policy	Service	Yes	Yes	-	-
Extended Federated BPaaS	process	Yes	Yes	Yes	Yes

V. SYSTEM MODEL

At this point, it should be noted that the execution of the business process is assigned to a time period t, and we use the parameter t as index for considering the time dependent variable. This allows us to apply the scheduling and resource provisioning generation using this model at a desired point of time. This point of time is when a business process step fails to enact the determined policy. Other cases could be considered, for example, in cases where missing information or new information is available. Such as additional process requests, leased or released resources, or an unexpected error, such as a VM connection failure.

Definition 1: Process Provider

We consider multiple process models, and therefore state the set of process models with P where:

$p_i \in P = \{p_1, \dots, p_n\}$, p_i indicates one process model of Process Provider.

Each process includes tasks that is used when the work in the process is not broken down to a finer level of Process detail, and therefore we consider the set of tasks within a process as follows:

$t_{p_i}^j \in T_{p_i} = \{t_{p_i}^1, \dots, t_{p_i}^m\}$, $t_{p_i}^j$ indicates one task model within the process model p_i .

In a certain time period t, a set of process instances are considered for execution in the cloud federation, where:

$p.ins_{p_i}^k \in PI_{p_i} = \{p.ins_{p_i}^1, \dots, p.ins_{p_i}^p\}$, $p.ins_{p_i}^k$ indicates one certain process instance of process model p_i in the cloud federation.

Each process instance includes task instances that is used to fulfil the process, and therefore we consider the set of task instances within a process instance as follows:

$ti_{p.ins_{p_i}^k}^l \in TI_{p.ins_{p_i}^k} = \{ti_{p.ins_{p_i}^k}^1, \dots, ti_{p.ins_{p_i}^k}^q\}$, $ti_{p.ins_{p_i}^k}^l$ indicates one task within the process instance $p.ins_{p_i}^k$.

Definition 2: Cloud Provider

A cloud provider C is defined as a pair C = (Sc, VMc), where:

– $S_c = ((S_1,R_1), \dots, (S_i,R_i))$ represents the service models of cloud service provider C where S_i is the service and the R_i is the set of policy rules supported by the service S_i.

– $VM_c = \{VM_1, \dots, VM_j\}$ represents the set of VM types provided within cloud provider C which vary in their configuration and their cost per billing time unit and data transfer.

Each VM type has a specific resource, e.g., available CPU and RAM as well as bandwidth capacity.

Each VM type is defined as a tuple $VM = (C_p, R_p, D_p, C_c, R_c, D_c)$ where C_p is the price of CPU, R_p is the price of RAM storage, D_p is the price of data transfer, C_c is the max capacity for CPU, R_c is the max capacity for RAM storage and D_c is the max capacity for data transfer rate.

Definition 3: Cloud Federation

A cloud Federation is defined as $F = (C_F, T_F)$, where:

- $C_F = \{C_1, \dots, C_i\}$, represents the cloud providers participating in the cloud federation F.

- T_F = (TR, TC), where the TR is the data transfer rate between clouds in the federation and the TC is the data transfer cost.

Due to a specific instance of a particular process, it does not necessarily call corresponding service instances because the cloud federation tries to minimize the total cost, that is, the cost of renting a VM. Therefore, at the beginning of the t period, it may not yet be necessary to start a process because its due date in the future is sufficient and the execution of the process may lead to a subsequent optimization period.

VI. OPTIMIZATION MODEL AND ADAPTIVE ALLOCATION IN FEDERATED CLOUDS

A) Simple Workflow Policy Matching Algorithm

We implemented a simple workflow policy matching Algorithm called SWPMA. Depending on the list of services that is requested and the candidate list of candidates, SWPMA iterates through the list of services and randomly selects a candidate for each, which meets all the requirements of the policy. Therefore, for each selected provider, there must be a policy respectively inside user defined ranges in the workflow manifest.

Additionally, the algorithm examines the capacity of the current data center whether the load allows the deployment of the requested service type. However, the result set may be empty that means non e of the cloud providers meet the requested criteria.

The following pseudo-code describes the operation of the SWPMA algorithm using the system model presented in the previous section.

```
Input : T(task t and its children) , SP , VM = (Cp_MAX, Rp_MAX, Dp_MAX) , Rtask
For each task in T
For each service provider p in SP Do
  If task is Deployable in p Then
    If VM = (Cp_MAX) <= VM = (Cp)  & VM = (Rp_MAX) <= VM = (Rp) &
     VM = (Dp_MAX) <= VM = (Dp) & Rtask ⊂  Sc (R)
    Add(p, CList)

    Endif
  Endif
Endfor

If size of CList ≠ 0
    Then
       p_new = Choose random p from CList
       RandomAllocationMap.add(task, p_new)
     Else return null
Endif
Endfor

Output : RandomAllocationMap
```

Fig. 3. Simple Workflow Policy Matching Algorithm (pseudo-code)

B) OPTIMIZATION MODEL

Described SWPMA algorithm is flexible in adapting SLA features. Hence, it cannot handle the use, because functional characteristics are more important than the non-functional or selecting providers, while reducing overall costs. Additionally, network connectivity between clouds and traffic costs are ignored in the implementation. Therefore, we use an economic utility-based matching algorithm that meets the needs of customer policy needs and their QoS settings. The main strategy of the tool algorithm is to optimize the user cost for the quality of the requested service using an optimization model.

The workflow is executed and managed by the Federated BPaaS manager. When an step of a workflow fails to fulfill the policy of that task, it is required to re-schedule and allocate that task and subsequent tasks (T) to appropriate cloud providers in the cloud federation regarding the cost of data transfer between clouds. The optimization model for the purpose of cost reduction is as follows:

$$\text{Min} \quad \sum_{task \in T} \sum_{vm \in VM_C} (Cp, Rp, Dp) * n(vm, t) + DS_{Cost} + DT_{Cost}$$

The term DS_{Cost} and DT_{Cost} represent the cost of data storage and data transfer respectively for the migration of the task. As intra-cloud data transfer is free of charge, the data transfer cost will be 0 in the same cloud. Data transfer cost is calculated as follows:

$$DT_{Cost} = \begin{cases} T(src) + T(dst) & Inter\_Cloud \\ 0 & Intra\_Cloud \end{cases}$$

Fourth term in the formula is for determining the priority of the task. The current time t is subtracted from DL_{pi} which represents the deadline of the process, so the task with a closer time to the deadline has the higher priority for assignment.

VII. EVALUATION

We evaluated the proposed SWPMA algorithm and the optimization model using a real Montage application by performing a series of simulation experiments. The simulation environment used and the results gathered in this section are presented in this section (Fig. 4).

To evaluate, we use a broker-based simulation framework to examine the deployment and scheduling of workflow in multi-cloud environments. This framework helps users find the Cloud service suitable for implementing and executing their work applications. Its main component is Match-Maker, which performs the matching process to select target clouds for deployment. Task scheduler assigns workflow tasks to the selected cloud resources. The architecture also includes a data manager to manage data transfers during the execution of the work. All communications with cloud providers provided by the standard interface provided by hosted Inter-cloud Gateways. A Workflow outsourced from the client side, delivers the workflow tasks to the Federated BPaaS Manager with respect to their execution order and data flow dependencies. We use WorkflowSim [33], a CloudSim-based version of the Pegasus WfMS [34] as a workflow engine.

To execute workflows using this framework, the Workflow Engine, in the first step, receives a description of the workflow and policy requirements and QoS from the user. After parsing the workflow, workflow engine implements different clustering techniques to reduce the number of workflow tasks. The workflow and the user needs are sent to the Broker. In the following, Match-Maker selects cloud resources that can fit the user's requirements using different matching methods. After that, all the VMs and the requested cloud storage are selected in Clouds, the Workflow Engine transmits the input data from the client to the cloud storage and then begins to publish the workflow tasks according to their executable instructions. During the implementation, the planner assigns each task to a VM with respect to different planning policies, while the data manager manages the transfer of cloud-to-cloud data. A duplicate catalog stores data catalogs by mapping work files to their central data centers. Eventually, the results of the execution are transferred to the cloud storage and can be retrieved.

To evaluate, we implemented the proposed SWPMA algorithm using the Java code as new matching policies in the Match-Maker Broker component.

For all simulation experiments, we have configured 15 cloud datacenters. Each cloud consists of 45 physical hosts, which are evenly divided between three different host types with 12 to 24 CPU cores. All 15 modeled datacenters provide computational services with different VM settings. In addition, 12 of them have Cloud Storage service.

To have an experimental evaluation of our match-making approach with a real case study, we will consider using a Montage workflow using the previously described simulation framework. Montage workflows mapped by Pegasus to the NSF Cyber-Infrastructure are characterized by tens of thousands of executable tasks and the processing of thousands of images.

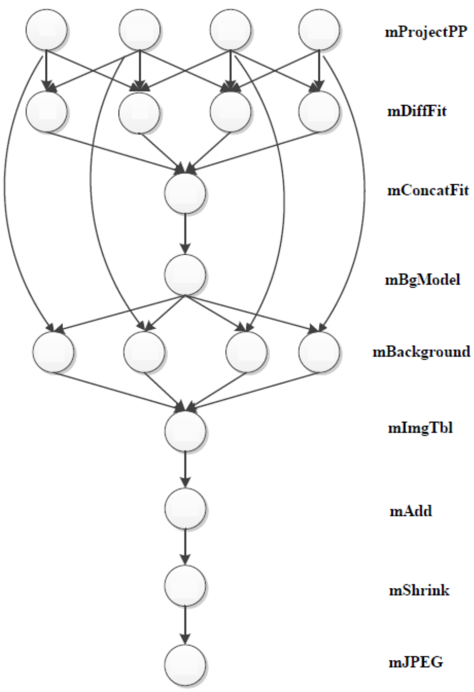

Fig. 4. Montage Workflow DAX (18 jobs) example

For all of our experiments, we use real-time Montage XML trace (1000 jobs) generated by Pegasus WfMS from a workflow production process.

Job descriptions, including runtime information and input/output inf ormation, have been imported with Workflow Parser of WorkflowSim. To reduce scheduling overhead, we configured the clustering engine to use vertical clustering as a hybrid technique. Here, the sequential tasks of each vertical level are merged into a cluster task. Additionally, we configured the Workflow engine to run a maximum of five tasks to the server at each scheduling period (the default value used by Pegasus). As a planning policy, we first use the simple Round Robin programmer who takes the workflow schedule tasks to the first VMs in the workflow regardless of the location of the data center. Clearly for data-intensive workflows, we can use more sophisticated scheduling policies such as scheduling based on data locality [35]; using data locality to improve the performance of the execution of the Montage astronomical workflow [36] in the cloud.

The scenario of using a Montage workflow deployment using our simulation framework includes the following steps:

First, the user provides his functional requirements for deploying a workflow with requests from 10 to 45 VM and a cloud storage to store data. We assume that all VMs located in the same data center are connected to a shared storage. After getting QoS and user budget requirements, the Workflow engine sends a user request to the Cloud Service Broker (1) to select the appropriate Clouds for deployment based on the policy of matching the configuration. Then the requested resources are deployed (2), the Workflow engine transmits the input from the client to the cloud storage and then starts the execution of the task (3). Finally, the output data is stored when storing the cloud, when the execution is completed and can be obtained from the customer.

To collect simulation results, we repeated 20 times in the same host, and then we calculated the average value. The results of the simulation are shown in the figure 5 and figure 6.

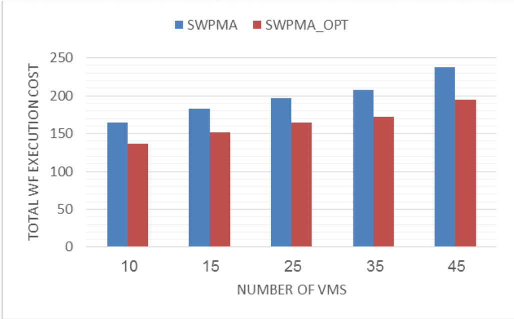

Fig. 5. Total WF execution cost with SWPMA and SWPMA_OPT for different VM numbers

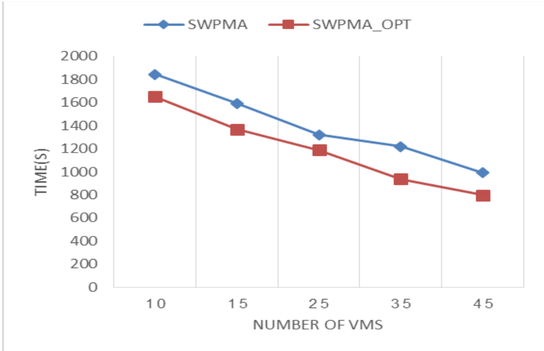

Fig. 6. WF execution time with SWPMA and SWPMA_OPT for different VM numbers

TABLE III and IV illustrates the improvement in makespan and cost for workflow execution in different scenarios.

TABLE III. COST REDUCTION IMPROVEMENT BY SWPMA_OPT VS SWPMA

VM numbers	SWPMA	SWPMA_OPT	Cost Reduction %
10	165	137	17%
15	183	152	17%
25	197	165	16%
35	208	172	17%
45	238	195	18%

Results of two algorithms reveals that the average cost reduction in workflow execution is 17% which is considerable in large scale execution of business processes.

TABLE IV. MAKESPAN IMPROVEMENT BY SWPMA_OPT VS SWPMA

VM numbers	SWPMA	SWPMA_OPT	Makespan Improvement %
10	1842	1650	10%
15	1592	1370	14%
25	1320	1188	10%
35	1220	938	23%
45	992	799	19%

TABLE IV contains the data for makespan improvement in workflow execution. For 35 number of VMs the improvement in makespan time is 23% and 19% improvement for 45 VMs. It shows that although the makespan time is decreased in comparison to SWPMA, the improvement rate could be reduced by increasing VMs as the overhead of data migration could increase.

VIII. Discussion

As it is illustrated in the figure 5 and 6, as the total cost decreases the execution time for workflow increases. So, leasing les s VMs causes high make span time and if we need faster execution of workflows we should lease more number of the VMs. Moreover, the SWPMA_OPT algorithm shows an improvement in the cost and make span in comparison with the SWPMA algorithm which uses a random approach to re-allocation of task to services when it fails to fulfill the workflow policy.

The Extended Federated BPaaS responds effectively to the concerns of business process consumers using SOA-based computing. Organizations can leverage their business processes into federal BPaaS management with different forecasts such as business policies and QoS. The consumer is not worried about finding the right service in the cloud. In fact, managing the workflow for matching services from the directory service and scheduling task schedules is the responsibility of workflow manager. The distribution manager monitors the compliance of the business process and examines the appropriate measures to ensure the implementation of the policy during the execution of the workflow. Therefore, the business process consumer assures that the implementation of business process is aligned with the specified business strategy and the distribution of tasks in an affordable manner.

A Cloud Service Provider that is a member of the Cloud Federation uses the Cloud Coordinator with the BPaaS Federal Director. The Cloud Coordinator plans local applications locally and helps cloud providers to improve the use of resources in the cloud. In addition, monitoring the execution of the task, and in the event of a violation of QoS or policy, the distribution manager will assign tasks to other services.

IX. Conclusion

In this paper an Extended Federated BPaaS Model was proposed for deploying business processes in the cloud federation environment. Based on a mediator, the proposed framework helps cloud users to get cloud resources considering their policies and tasks' QoS requirements. The developed model has been compared with other models that support business policy, and suggests that the proposed model can efficiently deploy processes of the customers on the federated clouds. We focused on the adaptive allocation of resources that actively distribute the tasks to achieve better makespan time and data transfer cost.

To evaluate the model, we implemented it the WorkflowSim tool with considering the data transfer challenges in the Cloud federation environment. We implemented a simple workflow policy matching Algorithm called SWPMA that considers the business policy of the process but does not consider any optimization. This algorithm was improved by a proposed optimization moled. The results of experiments show that proposed optimization mechanism reduces the data transfer between the Clouds and improves the makespan for workflow execution. Furthermore, the results reveals increasing the number of vms has limitation and rate of improvement could decrease.

For the future works, we could implement AI approaches to improve the performance of the model to have a better non-functional attributes.

References

[1] E. E. Mon, M. M. Thein, and M. T. Aung, "Clustering Based on Task Dependency for Data-intensive Workflow Scheduling Optimization," *Proc. 9th Work. Many-Task Comput. Clouds, Grids, Supercomput.*, pp. 20–25, 2016, DOI: 10.1109/MTAGS.2016.07.

[2] J. Lehner, "Personal BPM-Bringing the Power of Business Process Management to the User.," *Zeus*, no. February, pp. 19–20, 2015.

[3] P. Grefen, R. Eshuis, N. Mehandjiev, G. Kouvas, and G. Weichhart, "Internet-based support for process-oriented instant virtual enterprises," *IEEE Internet Comput.*, vol. 13, no. 6, pp. 65–73, 2009, DOI: 10.1109/MIC.2009.96.

[4] T. Leadership *et al.*, "Business Process Technology and the Cloud : defining a Business Process Cloud Platform," *Proc. 21st Int. Prod. Dev. Manag. Conf.*, vol. 393, no. September, pp. 195–203, 2012.

[5] J. M. Galloway and S. Vrbsky, "A CLOUD ARCHITECTURE FOR REDUCING COSTS IN LOCAL PARALLEL AND DISTRIBUTED VIRTUALIZED CLOUD ENVIRONMENTS," 2013.

[6] Y. Zhao *et al.*, "A Service Framework for Scientific Workflow Management in the Cloud," *IEEE Trans. Serv. Comput.*, vol. 1374, no. 6, pp. 1–1, 2014.

[7] S. S. Chauhan, E. S. Pilli, and R. C. Joshi, "A broker based framework for federated Cloud environment," *2016 Int. Conf. Emerg. Trends Commun. Technol. ETCT 2016*, 2017.

[8] A. Gupta, P. Dhyani, O. P. Rishi, and V. Pathak, "Service Request Approach for e-Governance using Federation of Cloud," no. 5, pp. 597–601, 2018.

[9] N. Anwar and H. Deng, "Elastic Scheduling of Scientific Workflows under Deadline Constraints in Cloud Computing Environments," *Futur. Internet*, vol. 10, no. 1, p. 5, 2018.

[10] Z. Wen, J. Cala, P. Watson, and A. Romanovsky, "Cost Effective, Reliable and Secure Workflow Deployment over Federated Clouds," *IEEE Trans. Serv. Comput.*, vol. 10, no. 6, pp. 929–941, 2017.

[11] K. Dar, A. Taherkordi, R. Rouvoy, and F. Eliassen, "Adaptable service composition for very-large-scale internet of things systems," *Proc. 8th Middlew. Dr. Symp. - MDS '11*, pp. 1–6, 2011.

[12] C. L. B and N. Assy, "Adaptive Deployment of Service-based Processes into Cloud Federation," vol. 10570, pp. 275–289, 2017.

[13] M. R. M. Assis, L. F. Bittencourt, and R. Tolosana-Calasanz, "Cloud federation: Characterisation and conceptual model," *Proc. - 2014 IEEE/ACM 7th Int. Conf. Util. Cloud Comput. UCC 2014*, pp. 585–590, 2014.

[14] F. Jrad, J. Tao, I. Brandic, and A. Streit, "SLA enactment for large-scale healthcare workflows on multi-Cloud," *Futur. Gener. Comput. Syst.*, vol. 43–44, pp. 135–148, 2015.

[15] *M. Papazoglou, Web Services: Principles and Technology, Addison-Wesley, 2008.* .

[16] F. Jrad, "A Broker-based Framework for Multi-Cloud Workflows Steinbuch Centre for Computing," pp. 61–68.

[17] P. Korambath *et al.*, "Deploying kepler workflows as services on a cloud infrastructure for smart manufacturing," *Procedia Comput. Sci.*, vol. 29, pp. 2254–2259, 2014.

[18] J. Wang, M. AbdelBaky, J. Diaz-Montes, S. Purawat, M. Parashar, and I. Altintas, "Kepler + CometCloud: Dynamic scientific workflow execution on federated cloud resources," *Procedia*

Comput. Sci., vol. 80, pp. 700–711, 2016.

[19] S. Pandey and D. Karunamoorthy, "WORKFLOW ENGINE FOR CLOUDS," pp. 321–344, 2011.

[20] R. N. Calheiros, C. Vecchiola, D. Karunamoorthy, and R. Buyya, "The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds," *Futur. Gener. Comput. Syst.*, vol. 28, no. 6, pp. 861–870, 2012.

[21] J. Zhou, C. Sun, W. Fu, J. Liu, L. Jia, and H. Tan, "Modeling, design, and implementation of a cloud workflow engine based on aneka," *J. Appl. Math.*, vol. 2014, 2014.

[22] X. Analysis, W. Ellis, D. Hersh, X. Analysis, W. Ellis, and D. Hersh, "XACML 3.0 Analysis," pp. 0–18, 2015.

[23] "eXtensible Access Control Markup Language (XACML) Version 3.0- OASIS Standard- 22 January 2013." [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.

[24] M. X. Wang, K. Y. Bandara, and C. Pahl, "Process as a service - Distributed multi-tenant policy-based process runtime governance," *Proc. - 2010 IEEE 7th Int. Conf. Serv. Comput. SCC 2010*, pp. 578–585, 2010.

[25] "Yangui, S., Klai, K., Tata, S.: Deployment of service-based processes in the cloud using petri net decomposition. In: Meersman, R., Panetto, H., Dillon, T., Missikoff, M., Liu, L., Pastor, O., Cuzzocrea, A., Sellis, T. (eds.) OTM 2014. LNCS, vol. 8841, pp."

[26] "Mastelic, T., Fdhila, W., Brandic, I., Rinderle-Ma, S.: Predicting resource allocation and costs for business processes in the cloud. In: 2015 IEEE World Congress on Services, pp. 47–54, June 2015."

[27] P. Hoenisch, S. Schulte, S. Dustdar, and S. Venugopal, "Self-adaptive resource allocation for elastic process execution," *IEEE Int. Conf. Cloud Comput. CLOUD*, pp. 220–227, 2013.

[28] P. Hoenisch, C. Hochreiner, D. Schuller, S. Schulte, J. Mendling, and S. Dustdar, "Cost-Efficient Scheduling of Elastic Processes in Hybrid Clouds," *Proc. - 2015 IEEE 8th Int. Conf. Cloud Comput. CLOUD 2015*, pp. 17–24, 2015.

[29] "Rekik, M., Boukadi, K., Assy, N., Gaaloul, W., Ben-Abdallah, H.: A linear program for optimal configurable business processes deployment into cloud federation. In: 2016 IEEE International Conference on Services Computing (SCC), pp. 34–41, June 2016."

[30] "Verbelen, T., Stevens, T., Turck, F.D., Dhoedt, B.: Graph partitioning algorithms for optimizing software deployment in mobile cloud computing. Future Gener. Comput. Syst. 29(2), 451–459 (2013). special section: Recent advances in e-Science."

[31] P. Hoenisch, C. Hochreiner, D. Schuller, S. Schulte, J. Mendling, and S. Dustdar, "Cost-Efficient Scheduling of Elastic Processes in Hybrid Clouds (PDF Download Available).pdf."

[32] H. Nasiriasayesh and A. Yari, "A Model For Policy-aware Execution of Business Process in Federated Clouds," in *2018 9th International Symposium on Telecommunications (IST'2018)*, 2018.

[33] W. Chen and E. Deelman, "WorkflowSim: A toolkit for simulating scientific workflows in distributed environments," *2012 IEEE 8th Int. Conf. E-Science, e-Science 2012*, 2012.

[34] E. Deelman *et al.*, "Pegasus, a workflow management system for science automation," *Futur. Gener. Comput. Syst.*, vol. 46, pp. 17–35, 2015.

[35] F. Jrad, J. Tao, I. Brandic, and A. Streit, "Multi-dimensional Resource Allocation for Data-intensive Large-scale Cloud Applications," *Proc. 4th Int. Conf. Cloud Comput. Serv. Sci.*, pp. 691–702, 2014.

[36] J.-S. Vöckler, G. Juve, E. Deelman, M. Rynge, and B. Berriman, "Experiences Using Cloud Computing for a Scientific Workflow Application," *Proc. 2nd Int. Work. Sci. cloud Comput.*, pp. 15–24, 2011.

AUTHORS' INFORMATION

Hamidreza Nasiriasayesh received his B.Sc. degree in Computer Engineering, Software from the Sharif University of Technology (SUT), Tehran, Iran, in 2008 and the M.Sc. degree in Information Technology Engineering from Shahid Beheshti University (SBU), Tehran, Iran in 2011. He is currently doing research in Information Technology Research Faculty of Iran Telecom Research Center (ITRC). His research mainly focuses on Cloud Computing, Service Computing and Simulation of Distributed Systems, with emphasis on Cloud Federation. Currently, he is working on Workflow Management System in Federated Clouds.

Alireza Yari received his B.Sc. degree in Control System Engineering in 1993 from the University of Tehran (UT), Iran, and M.Sc. and Ph.D. degrees in Systems engineering in 2000 from Kitami Institute of Technology, Japan. He is currently doing research in Information Technology Research Faculty of Iran Telecom Research Center (ITRC). His research interests include cloud computing and data centers. He is also working on application of cloud computing in data-intensive applications, such as web search engine.