A Framework for Data Protection in Cloud Federations

Lena Mashayekhy, Mahyar Movahed Nejad, and Daniel Grosu
Department of Computer Science
Wayne State University
Detroit, MI 48202, USA
{mlena, mahyar, dgrosu}@wayne.edu

***Abstract*—One of the benefits of cloud computing is that a cloud provider can dynamically scale-up its resource capabilities by forming a cloud federation with other cloud providers. Forming cloud federations requires taking the data privacy and security concerns into account, which is critical in satisfying the Service Level Agreements (SLAs). The nature of privacy and security challenges in clouds requires that cloud providers design data protection mechanisms that work together with their resource management systems. In this paper, we consider the privacy requirements when outsourcing data and computation within a federation of clouds, and propose a framework for minimizing the cost of outsourcing while considering two key data protection restrictions, the trust and disclosure restrictions. We model these restrictions as conflict graphs, and formulate the problem as an integer program. In the absence of computationally tractable optimal algorithms for solving this problem, we design a fast heuristic algorithm. We analyze the performance of our proposed algorithm through extensive experiments.**

***Keywords*-data protection; federation formation; virtual machine placement; cloud computing.**

I. Introduction

In recent years, many companies have been migrating or building their business into clouds. One of the major concerns for such companies and potential individual users in adopting cloud services is their data protection. Therefore, widespread adoption of cloud services depends on several technological challenges such as guaranteeing data privacy and security. Given the importance of data privacy, customers' fear of sensitive data leakage should be the primary concern when providing cloud services.

Privacy regulations such as the Fair Information Principles [1] are applicable to cloud services. In addition, users project their expectations of confidentiality and control in the Service Level Agreements (SLAs). However, cloud providers are accountable for meeting the privacy expectations of the users, and cloud providers should limit the access to individuals' confidential data and to companies' commercially sensitive data.

One of the benefits of clouds is the possibility of dynamically enhancing their resource capabilities by forming federations with other cloud providers in order to rapidly scale-up when required. However, forming cloud federations presents a host of new challenges resulting from the current lack of efficient cloud federation formation mechanisms. One of the major challenges is that a federation formation mechanism needs to address the data protection concerns that arise from outsourcing computations [2]. A key aspect of cloud federations, however, is that their infrastructure is shared among cloud providers and it is off the premises of a single cloud provider. Therefore, there exists a significant threat to data privacy and security associated with remote storage and processing of data.

In this paper, we consider the privacy requirements when outsourcing data and computation within a federation of clouds. We design a data protection framework for cloud federations that minimizes the cost of outsourcing. The benefits of employing our framework on a cloud federation are threefold. First, it helps align the cloud services with the users' concerns regarding their data protection. Second, it reduces the cost of upgrading the system software to support future data protection needs. Third, it avoids future costs and penalties resulting from leakage of sensitive data.

Virtualization is a major breakthrough enabling cloud providers to abstract the physical infrastructure, and to hide the complexity of underlying resources. On the other hand, the ever-growing demand for cloud computing services places the virtual machine (VM) management at the heart of cloud providers decision making process. The cloud provider creates a pool of virtualized resources which are offered to clients as VM instances. When the demand exceeds the capacity of available resources, the cloud provider can scale-up by forming a cloud federation with other cloud providers, and flexibly mapping and moving VMs (using VM migration technologies [3]) to the other cloud providers. In this study, we focus on data protection when the computation is outsourced to other cloud providers within the federation via such VM migrations, and propose a framework to minimize the total cost of outsourcing while considering two restrictions, as follows. First, there are several limitations for a cloud provider in assigning VMs to other cloud providers. Such limitations could be due to trust and reliability issues, or due to the geographical location of the other cloud provider. The location of a cloud provider can arise privacy and security issues since transferring a VM over to specific regions (e.g., crossing national boundaries), raises legal concerns. Therefore, we consider such restrictions, called *trust restrictions*, during the VM assignment and migration.

The trust restrictions specify the VMs that should not be assigned to specific cloud providers. Second, if several VMs are co-located on the same cloud provider they can reveal sensitive information. However, the user or the cloud customer requires through agreements that such information be accessible only to the main cloud provider and not to others. The cloud provider is accountable for protecting such information, and revealing it is against the agreement. Therefore, we consider such restrictions on co-locating VMs, called *disclosure restrictions*. These disclosure restrictions, specify which VMs can never be co-located on the same cloud provider when outsourcing the cloud services. This represents another level of data protection (after the encryption level) which subsequently reduces the need to encrypt all data.

A. *Our Contribution*

We propose a framework for data protection in a federation of clouds that considers the data privacy and security restrictions while minimizing the cost of outsourcing the computation to the cloud service providers that are part of the federation. We model the two key data privacy restrictions in cloud federations as two conflict graphs, one graph representing the conflicts between VMs and cloud providers, while the other, representing the conflicts among VMs. We then formulate the data protection problem in cloud federations as an integer program. In the absence of computationally tractable optimal algorithms for solving this problem, we design a fast heuristic algorithm. Our proposed algorithm incorporates a novel VM placement strategy in order to find close to optimal solutions, minimizing the cost of outsourcing while satisfying the data protection constraints. We provide a comprehensive assessment through extensive performance analysis experiments and compare the obtained solutions with the optimal solutions.

B. *Related Work*

A cloud architecture that allows a cloud to build a federation with other clouds was introduced by Celesti et al. [4]. Their model considers that a cloud provider is unable to fulfill its users' requests and forwards the requests to other clouds. Mashayekhy and Grosu [5], addressed the problem of federation formation in clouds and designed a coalitional game-based mechanism that enables the cloud providers to dynamically form a cloud federation maximizing their profit. Mashayekhy and Grosu [6] investigated the problem of federating resources in grids by employing coalitional game theory. In addition, they studied the problem of federating resources in grids considering trust relationship among grid service providers [7]. Li et al. [8] investigated profit maximization strategies in cloud federations, where VMs are sold through auctions. They proposed a truthful mechanism for trading VMs within a federation. Samaan [9] proposed an economic model based on repeated games, to regulate capacity sharing in a cloud federation, where each provider aims at maximizing its profit. Bruneo [10] proposed performance evaluation techniques based on stochastic reward nets for federated clouds to predict and quantify the cost-benefit of a strategy portfolio and the corresponding quality of service (QoS) experienced by users. None of the above mentioned studies considered the data protection constraints in a federation of clouds. VM allocation in clouds has been studied extensively. Bin et al. [11] proposed a VM placement approach considering multiple data privacy constraints without considering the cost of outsourcing. In our previous studies [12], [13], we proposed truthful mechanisms for VM allocation in clouds such that their profit is maximized and the resources are utilized efficiently. Although these studies considered the cost when allocating the VM, they did not consider the data protection constraints.

Existing approaches to preserve privacy of stored data-sets in clouds are mainly based on encryption and anonymization. Data anonymization refers to hiding the privacy-sensitive information such as identities. Zhou et al. [14] proposed a framework called Prometheus, that automatically separates sensitive data from nonsensitive data independent of the specific applications. They proved that Prometheus guarantees the privacy-preserving feature. Dou et al. [15] proposed a privacy-aware cross-cloud service composition method that protects the privacy such that a cloud is not required to unveil all of its transaction records. Their method uses history records associated with a service's past transactions. Zhang et al. [16] proposed a method for protecting the data privacy in hybrid clouds, called Sedic. Sedic automatically partitions a MapReduce computing job in terms of data security levels, and then assigns nonsensitive data to a public cloud. Encrypting all data-sets in clouds is not only time consuming but also very costly. Zhang et al. [17] proposed a privacy leakage upper bound constraint-based approach to identify intermediate data-sets that need encrypting. Zhang et al. [18] proposed a scalable two-phase approach to anonymize large-scale data sets. In the first phase, original data-sets are partitioned into a group of smaller data-sets, where they are anonymized in parallel, producing intermediate results. In the second phase, the intermediate results are aggregated, and further anonymized to achieve consistency. Our proposed framework provides an additional layer of data protection by implementing the restrictions related to data security and privacy and at the same time minimizes the cost of outsourcing.

C. *Organization*

The rest of the paper is organized as follows. In Section II, we describe the data protection problem in cloud federations. In Section III, we present the proposed algorithm that solve the data protection problem in federations of clouds. In Section IV, we evaluate the properties of the proposed algorithm by extensive experiments. In Section V, we summarize our results and present possible directions for future research.

II. Data Protection in Cloud Federations Problem

In this section, we describe the model of the system, and the data protection problem.

A. System Model

We first describe the system model considering that a cloud provider P_0 wants to outsource its workload consisting of a pool of N VMs to other cloud providers. We consider a federation of cloud providers $\mathcal{F} = \{P_0, P_1, P_2, \ldots, P_M\}$ that are available to provide services. Each cloud provider $P_j \in \mathcal{F}$ has restricted computing capacity, denoted by R_j, available to provide to other cloud providers. Each provider P_j incurs cost when providing resources. For a cloud provider P_j, we denote by c_j, the cost associated with each VM instance executed on P_j, and by m_j, the cost associated with VM migration from P_0 to P_j.

Cloud provider P_0 does not have enough resources to fulfill the requested VMs, and needs to outsource some of the requested VMs to other cloud providers within the federation in order to execute the jobs and more importantly, to minimize its cost while satisfying the data protection restrictions imposed by the users. Therefore, the outsourcing decisions have to be made considering both data protection restrictions and cost minimization. As we mentioned in the introduction section, there are two data protection restrictions that have to be considered when outsourcing VMs within a federation of cloud providers: (i) the *trust restrictions*, specifying that some VMs cannot be outsourced to specific cloud providers; and, (ii) the *disclosure restrictions*, specifying that some VMs cannot be outsourced to the same cloud provider. We model these two restrictions as two *conflict graphs*.

To model the trust restrictions, we consider a graph $H(V \cup \mathcal{F}, A)$, where V and $\mathcal{F}$ are the sets of VMs and cloud providers, respectively, representing the vertices in the graph, and A is a set of edges $< i, j >$ representing the conflict between $VM_i \in V$ and $P_j \in \mathcal{F}$. If there is an edge between a VM and a cloud provider, it specifies that the VM cannot be assigned to that cloud provider.

To model the disclosure restrictions, we consider a graph $G(V, E)$, where V is a set of VMs representing the vertices in the graph, and E is a set of edges $< i, j >$ representing the conflict between VM_i and $VM_j \in V$. If VM_i and VM_j cannot be assigned to the same cloud provider, we consider them as a conflicting pair of VMs. If there is no edge between two VMs, these VMs can be assigned to the same provider. However, if there exists an edge between two VMs, these two VMs should not be assigned to the same provider.

B. Data Protection Problem

We define the data protection in cloud federations (DPCF) problem as follows. We consider that P_0 is the cloud provider that wants to outsource N VMs to other cloud providers part of the federation, in order to execute the application of its users. The cloud provider's goal is to minimize its cost while allocating VMs to participating clouds in the federation considering the trust and disclosure restrictions, specified by the the conflict graphs defined in the previous section.

We define an indicator variable δ_{ij}, $\forall i \in V, \forall j \in \mathcal{F}$, that characterizes the conflict between VM$_i$ and cloud provider P_j, and implicitly characterizes the trust restrictions, as follows:

$$\delta_{ij} = \begin{cases} 1 & \text{if VM}_i \text{ can be assigned to } P_j, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

If $\delta_{ij} = 0$, then it specifies that VM$_i$ should never be assigned to cloud provider P_j.

We define decision variables x_{ij} and y_j as follows:

$$x_{ij} = \begin{cases} 1 & \text{if VM}_i \text{ is assigned to } P_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

$$y_j = \begin{cases} 1 & \text{if there is a VM assigned to } P_j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We formulate the DPCF problem as an Integer Program, called IP-DPCF, as follows:

$$\text{Minimize } \sum_{i=1}^{N}\sum_{j=1}^{M} c_j x_{ij} + \sum_{j=1}^{M} m_j y_j \tag{4}$$

Subject to:

$$\sum_{i=1}^{N} x_{ij} \le y_j R_j, \qquad \forall j \in \mathcal{F} \setminus P_0 \tag{5}$$

$$\sum_{j=1}^{M} \delta_{ij} x_{ij} = 1, \qquad \forall i = 1, \ldots, N \tag{6}$$

$$x_{ij} + x_{kj} \le 1, \quad \forall < i, k > \in E, j = 1, \ldots, M \tag{7}$$

$$x_{ij} \le y_j, \qquad \forall i = 1, \ldots, N, \, \forall j \in \mathcal{F} \setminus P_0 \tag{8}$$

$$x_{ij} = \{0, 1\}, \qquad \forall i = 1, \ldots, N, \, \forall j \in \mathcal{F} \setminus P_0 \tag{9}$$

$$y_j = \{0, 1\}, \qquad \forall j \in \mathcal{F} \setminus P_0 \tag{10}$$

Objective function (4) represents the total cost of outsourcing all VMs to the federation. The total cost includes the execution and migration costs of outsourced VMs. Constraints (5) ensure that the allocation of VMs to each provider does not exceed the available capacity of that cloud provider. Constraints (6) guarantee that each VM is assigned to exactly one cloud provider that does not have any conflict with, according to the trust restrictions. Constraints (7) ensure that VMs assigned to a cloud provider do not have conflict, according to the disclosure restrictions. Constraints (8) ensure that a cloud provider is a member of the set of providers to which VMs are outsourced, that is, there exists at least a VM that is assigned to that cloud provider. Constraints (9) and (10) represent the integrality requirements for the decision variables. IP-DPCF determines the assignment of all VMs that cloud provider P_0 wants to outsource to the federation, minimizing the total cost of outsourcing, while satisfying the trust and disclosure restrictions.

III. Algorithm for Solving DPCF

In this section, we introduce our proposed algorithm that solves the DPCF problem. We first describe our proposed VM partitioning algorithm, called VMPA, which uses the conflict graph of disclosure restrictions to determine a partitioning of the VMs. We then describe our proposed algorithm, called DPCFA, that solves the DPCF problem. DPCFA algorithm uses the VM partitioning algorithm (VMPA) in order to minimize the cost of federation formation while satisfying the trust and disclosure restrictions.

A. VM Partitioning Algorithm

In this section, we propose the VM Partitioning Algorithm (VMPA). VMPA partitions the VMs in a way that conflicting VMs are not assigned to the same cloud provider. VMPA uses a conflict graph $G'(V', E')$ which is a subgraph of the conflict graph G, $G' \subseteq G$, as an input. The algorithm builds a max-heap $\mathcal{V}$ to order the VMs in V' based on their number of conflicting VMs, denoted by d_i. The max-heap has two main functions associated with it: (i) enqueue(), that inserts a VM along with its priority into the heap; and (ii) extractMax(), that extracts the VM with the highest priority. A VM with the highest priority, i.e., with the most conflicts is always at the top of the heap. The algorithm also creates a subset S_0 of VMs that do not have any conflicts (i.e., $d_i = 0$). Considering S_0 is critical for the algorithm in order to minimize cost. We will discuss this in more details in the next subsection. The algorithm extracts the VM with the highest priority (i.e., with the highest number of conflicts), and assigns it to S_1, where $K = 1$ tracks the number of current subsets without considering S_0. The assignment of the rest of the VMs in $\mathcal{V}$ based on their priorities is as follows: For each of the current subsets, it checks if there exists any conflicting VMs. The algorithm assigns the VM to the first subset that does not have any conflicting VMs. If there is no subset without conflict, it exists, and creates a new subset for that VM. The result of the partitioning of the VMs is S_0 along with other subsets S_i. Then, the algorithm sorts the partitioned VMs based on the number of VMs in each subset such that S_1 is the largest subset. Finally, the algorithm returns the sets S_i, which represent the partitioning of the set of VMs. The VMs that are part of a set S_i do not conflict with each other, and can be assigned to the same cloud provider. These sets will be used in our proposed algorithm DPCFA, described in the next subsection.

Algorithm 1 VMPA: VM Partitioning Algorithm

```
Input: $G'(V', E')$
Create an empty max heap $\mathcal{V}$
for all $i \in V'$ do
    $d_i \leftarrow$ the number of connected VMs to VM$_i$
    if $d_i > 0$ then
        $\mathcal{V}$.enqueue($i, d_i$)
    else
        $S_0 \leftarrow S_0 \cup \{i\}$
$(i, d_i) = \mathcal{V}$.extractMax()
$S_1 = \{i\}$
$K = 1$
while $\mathcal{V}$ is not empty do
    $(i, d_i) \leftarrow \mathcal{V}$.extractMax()
    flag ← FALSE
    for all $k = 1, \ldots, K$ do
        for all $j, < i, j > \in E', j \in S_k$ do
            flag← TRUE
            break
        if ! flag then
            $S_k \leftarrow S_k \cup \{i\}$
            break
    if flag then
        $K = K + 1$
        $S_K = \{i\}$
Sort $S_1, \ldots, S_K$ based on descending order of their size
Output: $S_0, S_1, \ldots, S_K$
```

B. Algorithm for Solving DPCF

In this section, we propose the Data Protection for Cloud Federations Algorithm, called DPCFA. DPCFA requires the two conflict graphs as input. First, $G(V, E)$ represents the conflict graph for the disclosure restrictions, where V is the set of VMs and E is the set of conflict edges between two VMs. If $< i, j >$ exists, then VM_i and $VM_j \in V$ cannot be assigned to the same provider. Second, $H(V \cup \mathcal{F}, A)$ represents the conflict graph for the trust restrictions, where $V \cup \mathcal{F}$ is the set of vertices representing the VMs and cloud providers, and A is the set of conflicting edge between a VM and a cloud provider. If $< i, j >$ exists, then $VM_i \in V$ can be assigned to $P_j \in \mathcal{F}$.

We define the cost metric γ_j for cloud provider P_j based on its VM execution cost along with its average migration cost as follows:

$$\gamma_j = c_j + \frac{m_j}{R_j} \tag{11}$$

Besides the actual VM execution cost, the cost metric should consider the estimated migration cost. In doing so, our proposed cost metric considers the average migration cost for each VM based on the available capacity R_j of the cloud provider. This is due to the fact that when migrating a batch of VMs, the cloud provider pays only once for the migration cost. As a result, the average migration cost is a reasonable estimate.

DPCFA creates a min-heap $\mathcal{F}^q$, in order to keep the cloud providers ordered based on their cost metric, γ_j. The min-

Algorithm 2 DPCFA: Data Protection for Cloud Federations Algorithm

```
**Input:** Conflict graphs $G(V,E)$ and $H(V \cup \mathcal{F}, A)$
Create an empty min-heap $\mathcal{F}^q$
**for all** $j \in \mathcal{F} \setminus P_0$ **do**
    $\gamma_j = c_j + \frac{m_j}{R_j}$
    $\mathcal{F}^q$.enqueue($j, \gamma_j$)
    $\hat{R}_j = R_j$
**while** $V$ is not empty **do**
    **if** $\mathcal{F}^q$ is empty **then**
        Infeasible solution
        **break**
    $(j, \gamma_j) \leftarrow \mathcal{F}^q$.extractMin()
    $G'(V',E') \leftarrow$ non-conflicting VMs with $P_j$ based on the updated graph $H$
    $\{S_0, S_1, \ldots, S_K\}$=VMPA($G'$)
    **if** $|S_1| + |S_0| > 0$ **then**
        $y_j = 1$
    **else**
        **continue**
    **if** $R_j - |S_1| \geq 0$ **then**
        **for all** $i \in S_1$ **do**
            $x_{ij} = 1$
            Remove $i$'s adjacent edges from $E$
        $V = V \setminus S_1$
        $\hat{R}_j = R_j - |S_1|$
        **while** $\hat{R}_j > 0$ and $|S_0| > 0$ **do**
            $i \leftarrow$ a VM $\in S_0$
            $x_{ij} = 1$
            $\hat{R}_j = \hat{R}_j - 1$
            $S_0 = S_0 \setminus \{i\}$
            $V = V \setminus \{i\}$
            Remove $i$'s adjacent edges from $E$
    **else**
        $\bar{S} \leftarrow$ choose the first $R_j$ VMs in $S_1$
        **for all** $i \in \bar{S}$ **do**
            $x_{ij} = 1$
            Remove $i$'s adjacent edges from $E$
        $V = V \setminus \bar{S}$
        $\hat{R}_j = 0$
**Output:** $x$,$y$
```

heap has two main functions associated with it: enqueue() and extractMin(), where the former inserts a cloud provider along with its cost metric into the heap, and the latter extracts the cloud provider with the minimum cost metric. A cloud provider with the minimum cost is always at the top of the heap.

The algorithm starts the assignment process of VMs to cloud providers by choosing a cloud provider P_j with the minimum value for the cost metric γ_j. Then, it creates $G'(V', E')$ based on non-conflicting VMs with P_j in H. DPCFA calls the VMPA algorithm (described in the previous subsection) to partition the VMs in V' such that within each subset there is no conflicting VMs. Note that S_1 is always the largest subset. If cloud provider P_j has enough capacity to host the VMs in S_1, the algorithm assigns S_1 to P_j, and updates the current available capacity of P_j. DPCFA assigns S_0 considering the updated remaining capacities of the cloud provider. Note that VMs in S_0 do not have conflicts with any other VMs. This feature of our proposed algorithm makes use of the remaining capacities of the cloud providers with less cost. This novel VM placement leads to close to optimal solutions for the DPCF problem since it tries to assign as many VMs as possible to the cloud providers with the lowest cost. Note that in each iteration, DPCFA creates S_0 specific to each cloud provider to make the most use of VMs not having trust restrictions for that cloud provider, while those VMs do not have any disclosure restrictions. If cloud provider P_j does not have enough capacity, the algorithm assigns all capacity of P_j to the first R_j VMs in S_1, and removes them from V. Note that assigning the first R_j VMs is critical in obtaining close to optimal solutions since in VMPA the VMs are added to the subsets in order of their number of conflicting VMs. That means, a VM with the highest number of conflicts is considered first. Removing the first R_j VMs from the set of VMs makes the partitioning of the remaining VMs effective. Finally, the algorithm returns the assignment of VMs to the cloud providers as an output.

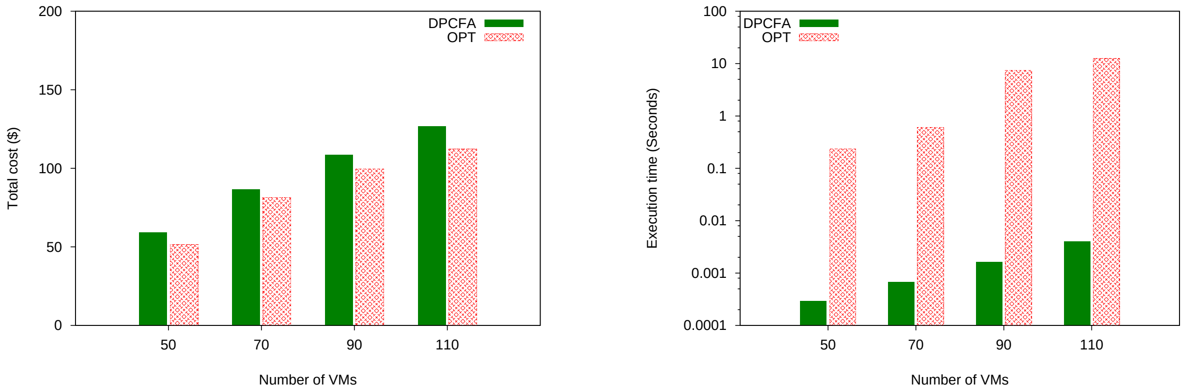
The time complexity of DPCFA is polynomial in the number of VMs, the number of cloud providers, and the number of conflicts among VMs and between VMs and cloud providers.

IV. Experimental Results

We perform extensive experiments in order to investigate the properties of the proposed algorithm, DPCFA. We compare the performance of DPCFA with that of OPT, where OPT obtains the optimal solution by solving IP-DPCF (Equations (4) to (10)). We implemented OPT using IBM ILOG CPLEX Optimization Studio Multiplatform Multilingual eAssembly. Since IBM ILOG CPLEX could not find solutions for DPCF instances with large numbers of VMs and cloud providers, we present two classes of experiments, small-scale and large-scale, to analyze the performance of the proposed DPCFA algorithm. In the small-scale experiments, we compare the results of DPCFA and OPT for DPCF instances with small number of cloud providers participating in the federation and small number of VMs that need to be outsourced. In the large-scale experiments, we compare the results of DPCFA and OPT for DPCF instances with large number of cloud providers participating in the federation and large number of VMs that need to be outsourced. DPCFA and OPT algorithms are implemented in C++ and the experiments are conducted on AMD 2.93GHz hexa-core dual-processor systems with 90GB of RAM which are part of the Wayne State Grid System. In this section, we describe the experimental setup and analyze the experimental results.

A. Experimental Setup

To analyze the performance of our proposed algorithm, we use a random graph model, the Erdös-Rényi model [19], for



(a)

(b)

Figure 1: DPCFA vs OPT (small-scale experiments): (a) Costs; (b) Execution time.

the conflict graph G. An Erdös-Rényi graph (m, p) is a graph constructed by connecting nodes randomly, where the graph has m nodes. The probability of having an edge in the graph is p for any pair of nodes, and it is independent from every other edge. That means, all graphs with m nodes and e edges have equal probability. Based on the parameter $p \in [0, 1]$, the graph can be sparse or complete. Since the Erdös-Rényi graph is a connected graph, leading to a conflict graph with all VMs having conflicts, we generate a smaller graph for the conflicted VMs and add the rest of the VMs. For the small-scale experiments, we consider instances of DPCF having the total number of VMs, $N = \{50, 70, 90, 110\}$ and eight cloud providers, where $m = \{30, 50, 70, 90\}$ is the number of conflicted VMs for each case, respectively, and $p = 0.05$. For the large-scale experiments, the number of VMs are $N = \{1000, 2000, 3000, 4000\}$, the number of cloud providers is 32, where $m = 60\%$ of the VMs have conflicts, and $p = 0.05$. The values of the execution costs and migration costs, are drawn from a uniform distribution over the Microsoft Azure prices [20]. In all cases, we choose the extra large VM configurations of Microsoft Azure for our experiments, with 8 cores and 14 GB RAM. The execution cost of a VM is between $[0.7, 1.1]\$$, while the migration cost is between $[2, 8]\$$. The capacity of the cloud providers available within the federation are generated based on the number of VMs. For the small-scale experiments, we use a uniform random distribution with variance 50 and mean 90, 90, 135, 180. For example, for an experiment with 50 VMs, the cloud providers' capacities are generated within the range $[40, 140]$. For the large-scale experiments, we use a uniform random distribution with variance 160 and mean 320, 640, 1280, and 2560, to generate the cloud providers' capacities. Note that the size of one VM is 8. For the small-scale experiments, we create the conflict graph H by choosing a random number between 0 and 3 to represents the number of conflicts between a VM and cloud providers. For example, a VM with 3 conflicts means that it cannot be assigned to 3 out of the 8 cloud providers. For the large-scale experiments, we create the conflict graph H by choosing a random number between 0 and 8 to represents the number of conflicts between a VM and the cloud providers.

B. *Analysis of Results*

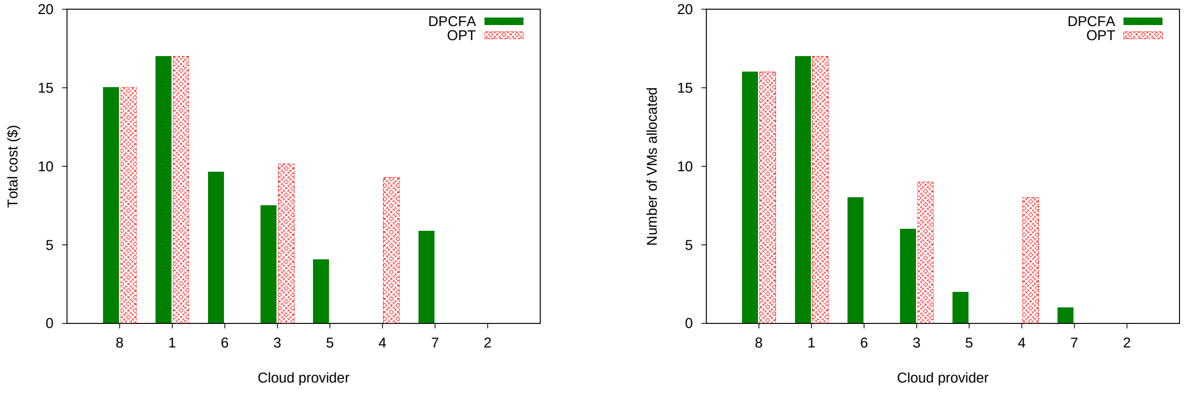
In the following, we analyze the results for each of the two classes of experiments.

1) Small-scale experiments: We analyze the performance of DPCFA and OPT by considering instances of DPCF with 8 cloud providers and four small sets of VMs that need to be outsourced ($N =$ 50, 70, 90, and 110).

Fig. 1a shows the total cost that the cloud provider incurs for outsourcing all the VMs to the federation. As shown in the figure, the cost achieved by using DPCFA is very close to that achieved when using OPT. For example, for 70 VMs, the total cost obtained by OPT is \$81.53 while the total cost obtained by DPCFA is \$86.59. This results in 6% optimality gap.

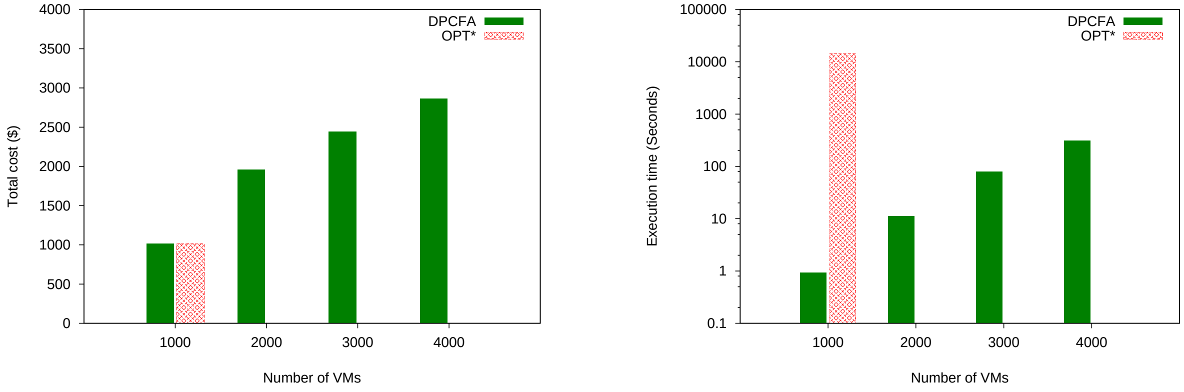
Fig. 1b shows the execution time of the algorithms. DPCFA is very fast, being three to four orders of magnitude faster than OPT, while obtaining close to optimal costs.

We analyze the results of the smallest instance with 50 VMs in more details. In Fig 2, we present the VM placement on each cloud provider and their cost for the case of 50 VMs available for outsourcing. Fig. 2a shows the cost that the cloud provider should pay to each cloud provider participating in the federation. Fig. 2b shows the number of VMs allocated to the cloud providers participating in the federation. Note that the cloud providers are sorted based on the cost metric on horizontal axis, i.e., P_8 is the cloud provider with the smallest cost metric, while P_2 is the provider with the largest cost metric. As a result, DPCFA chooses all lowest cost cloud providers to assign the VMs with the exception of P_4, where the remaining set of VMs has conflicts with it. Note that DPCFA and OPT choose the same allocation to the first two cloud providers with the



(a) (b)

Figure 2: DPCFA vs OPT with 50 VMs: (a) Costs; (b) VMs allocated.



(a) (b)

Figure 3: DPCFA vs OPT (large-scale experiments): (a) Costs; (b) Execution time. (*OPT was not able to determine the allocation when the number of VMs are 2000, 3000, and 4000 in feasible time, and thus, there are no bars in the plots for those cases)

lowest cost, and do not assign any VMs to the cloud provider with the highest cost.

2) Large-scale experiments: OPT could not find the solutions even after 8 hours for DPCF instances with 2000, 3000, and 4000 VMs. As a result, we are not able to compare the results of DPCFA with that of optimal solution for those cases.

Fig. 3a shows the total cost of the VM placement. For 1000 VMs, OPT finds the optimal assignment of VMs with total cost of $1010.89, while DPCFA obtains the total cost of $1017.68. The results show that DPCFA performs very well when the problem scales. The optimality gap in this case is less than 0.2%. Fig. 3b shows the execution time of DPCFA and OPT. For the obtained results, DPCFA is four orders of magnitude faster than OPT.

From all the above results, we conclude that DPCFA determines VM placements that give solutions close to the optimal while satisfying the trust and disclosure restrictions. It also requires small execution times, making it a suitable candidate for deciding the outsourcing and placement of VMs within a federation of clouds.

V. Conclusion

The benefits from using cloud services should not come at the cost of compromising the privacy and security of users' data. Data protection in terms of legal compliance and user trust are major issues in clouds, and they should be a top priority when designing cloud systems. On the other hand, the ever-growing demand for cloud services along with the demand dynamics require cloud providers to scale-up when needed. A practical platform to cope with such demand is the cloud federation. In this paper, we proposed a data protection framework for cloud federations that minimizes the cost of outsourcing the computation under data protection constraints. We proposed to represent the data protection restrictions as conflict graphs, and model the data protection problem as an integer program. In the absence of computationally tractable optimal algorithms for solving this problem, we designed a fast heuristic algorithm incorporating a novel VM placement strategy in order to find close to optimal solutions. The results from extensive performance analysis for small-scale and large-scale sets of experiments showed that our proposed algorithm is capable of finding close to optimal solution very fast. For future work, we plan to extend this study and integrate it into existing online dynamic federation formation mechanisms.

Acknowledgment

This research was supported in part by NSF grants DGE-0654014 and CNS-1116787.

References

[1] F. T. Commission, “Privacy online: Fair information practices in the electronic marketplace: A federal trade commission report to congress.” 2000.

[2] S. Pearson, “Taking account of privacy when designing cloud computing services,” in *Proc. of the IEEE ICSE Workshop on Software Engineering Challenges of Cloud Computing*, 2009, pp. 44–52.

[3] J. Zheng, T. Ng, K. Sripanidkulchai, and Z. Liu, “Pacer: A progress management system for live virtual machine migration in cloud computing,” *IEEE Transactions on Network and Service Management*, vol. 10, no. 4, pp. 369–382, 2013.

[4] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, “How to enhance cloud architectures to enable cross-federation,” in *Proc. of the 3rd IEEE Intl. Conf. on Cloud Computing*, 2010, pp. 337–345.

[5] L. Mashayekhy and D. Grosu, “A coalitional game-based mechanism for forming cloud federations,” in *Proc. of the 5th IEEE Intl. Conf. on Utility and Cloud Computing*, 2012, pp. 223–227.

[6] ——, “A merge-and-split mechanism for dynamic virtual organization formation in grids,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 540–549, 2014.

[7] ——, “A reputation-based mechanism for dynamic virtual organization formation in grids,” in *Proc. 41st IEEE Intl. Conf. on Parallel Processing*, 2012, pp. 108–117.

[8] H. Li, C. Wu, Z. Li, and F. Lau, “Profit-maximizing virtual machine trading in a federation of selfish clouds,” in *Proc. of the IEEE INFOCOM*, 2013, pp. 25–29.

[9] N. Samaan, “A novel economic sharing model in a federation of selfish cloud providers,” *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2013.

[10] D. Bruneo, “A stochastic model to investigate data center performance and qos in iaas cloud computing systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 560–569, 2014.

[11] E. Bin, O. Biran, O. Boni, E. Hadad, E. K. Kolodner, Y. Moatti, and D. H. Lorenz, “Guaranteeing high availability goals for virtual machine placement,” in *Proc. of the 31st IEEE Intl. Conf. on Dist. Comp. Syst.*, 2011, pp. 700–709.

[12] L. Mashayekhy, M. M. Nejad, and D. Grosu, “A truthful approximation mechanism for autonomic virtual machine provisioning and allocation in clouds,” in *Proc. of the ACM Cloud and Autonomic Computing Conf.*, 2013, pp. 1–10.

[13] M. M. Nejad, L. Mashayekhy, and D. Grosu, “A family of truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds,” in *Proc. of the 6th IEEE Intl. Conf. on Cloud Computing*, 2013, pp. 188–195.

[14] Z. Zhou, H. Zhang, X. Du, P. Li, and X. Yu, “Prometheus: Privacy-aware data retrieval on hybrid cloud,” in *Proc. of the IEEE INFOCOM*, 2013, pp. 2643–2651.

[15] W. Dou, J. Chen, X. Zhang, and J. Liu, “Hiresome-ii: Towards privacy-aware cross-cloud service composition for big data applications,” *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2013.

[16] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan, “Sedic: privacy-aware data intensive computing on hybrid clouds,” in *Proc. of the 18th ACM Conf. on Computer and Communications Security*, 2011, pp. 515–526.

[17] X. Zhang, C. Liu, S. Nepal, S. Pandey, and J. Chen, “A privacy leakage upper bound constraint-based approach for cost-effective privacy preserving of intermediate data sets in cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1192–1202, 2013.

[18] X. Zhang, L. T. Yang, C. Liu, and J. Chen, “A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 363–373, 2014.

[19] P. Erdös and A. Rényi, “On random graphs, I,” *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.

[20] WindowsAzure: Purchase Options - Pricing. [Online]. Available: http://www.windowsazure.com/en-us/pricing/calculator/