Efficient Online Virtual Machines Migration for Alert-Based Disaster Resilience

Omran Ayoub, Obed Huamani, Francesco Musumeci and Massimo Tornatore
Politecnico di Milano, Department of Electronics, Information and Bioengineering, Milan, Italy
Email: {*firstname.lastname*}@polimi.it

***Abstract*—Several recent weather-based disasters had very negative impacts on cloud networks, causing Data Center (DC) shutdown, consequent data-loss and intolerable downtime of cloud services. This has put the reactive disaster-resilient design of cloud networks on top the agenda of several cloud DC operators. DC operators are investigating approaches to avoid downtime of cloud services in case a DC is affected by a disaster. Thanks to virtualization most cloud services run on Virtual Machines (VMs) hosted by DCs, so it is possible to keep these services alive if the VMs are evacuated (namely, migrated) before the disaster from a DC affected by the disaster to a DC in a safe location, in an online technique. This technique is known as online "VM migration", which results without or with a minimal service downtime. In this paper, we present an Integer Linear Programming (ILP) model for efficient online VMs migration in case of an alerted disaster (e.g., most weather-based disasters, as hurricanes) such as to avoid service downtime. The ILP performs scheduling and assigns route and bandwidth to the migration of VMs towards a safe DC within an alert time, with the objective of maximizing the number of VMs migrated and minimizing service downtime, network resource occupation and migration duration. We present a comparative analysis of offline and online migration strategies such as to quantify the trade-off between downtime, network resource utilization and migration duration. Moreover, we investigate the impact of the memory dirtying rate on the online migration process, i.e., the number of VMs evacuated and network resource occupation.**

I. Introduction

In recent years, data center virtualization and cloud computing have been undergoing an exponential growth, as cloud infrastructures are increasingly hosting enterprise and public-Internet services. It is expected that as high as 94% of data center (DCs) workloads will be processed by cloud DCs in 2021 [1]. Virtualization is a main contributor to this growth, as it enables the sharing of computing resources across diverse locations, and allows network and DC operators to efficiently exploit their infrastructure, i.e., the cloud network. A cloud network is composed of a number of geographically-distributed and interconnected DCs. These DCs are built over physical servers based on general-purpose hardware where services are provisioned through the use of Virtual Machines (VMs) such that a single VM could support a cloud service. These cloud networks have an indispensable role in delivering of latency-sensitive and bandwidth-hungry services to end users possible. As any disruption of service is a major concern, it is crucial that cloud networks are always resilient to data-loss and to service disruptions. The unexpected impact recent weather-based disasters had on cloud networks and particularly on service downtime made the disaster-resilient cloud network design an important issue. For example, in 2012, hurricane Sandy flooded New York taking down a significant number of DCs for days, causing permanent damage in users data and temporary service disruption. However, the occurrence of some natural events is sometimes predictable, as an alert can be issued, e.g., some weather-based disasters such as hurricanes, floods or tornadoes. Therefore, such alerts can be exploited to perform proper inter-DC VM migration and data evacuation from the DCs in danger towards safer locations. In this paper, we focus on the inter-DC VM migration for weather-based disaster resiliency, which has never been, to the best of our knowledge, investigated previously.

A. Related Work

Some studies have already proposed inter-DC data-evacuation techniques for weather-based disaster resilience considering an evacuation deadline (i.e., time of occurrence of the disaster), e.g., Refs. [2] [3] focus on the maximization of data evacuated from DC located in a disaster zone. Our work is different than all existing studies on the topic, since we consider the *online* migration of VMs, meaning that the amount of data transferred is not known a priori, since in online VM migration, the total data transferred depends on the bandwidth assigned and the rate at which the VM memory gets modified. To the best of our knowledge, previous disaster-resilient technique did not investigate the online VM migration problem, as their aim was only to maximize data transferred and not account also for minimization of service downtime. Similarly, some studies have addressed the online VM migration problem with the aim of improving the migration process by minimizing the downtime or the VM migration duration (Refs. [4],[5]), improving the network utilization (Refs. [6], [7]). More specifically, Ref. [8] proposed a quantitative model for the migration duration and downtime of VM migration over a wide area network and Ref. [9] presented a theoretical analysis of the necessary bandwidth to satisfy the total migration time and the downtime constraints of a single VM migration. Furthermore, Refs. [7] and [10] studied serial and parallel migration strategies for multiple VMs. With respect to these works, our problem presents higher complexity as we consider an alert time (i.e., the evacuation deadline). The alert time constraints the migration to be performed in given amount of time, and thus it imposes a minimum bound on the migration bandwidth assignment. Moreover, we consider the scheduling problem of the VMs migration along with the routing and bandwidth assignment.

B. Paper Contribution

In this paper, we focus on online VM migration for alert-based disaster resilience in a distributed DC infrastructure. We propose an Integer Linear Programming (ILP) model which assigns VMs to destination DCs (i.e., to DCs not affected by the disaster), performs scheduling and assigns route and migration bandwidth for the VMs migration with the objective of maximizing the number of VMs migrated online. Our model considers the minimization of the network resource occupation through minimizing the amount of bandwidth utilized per link during the duration of the VM migration process as a secondary objective.

The rest of the paper is organized as follows: in Sec. II we present background information of VM migration. In Section III we introduce the ILP model for efficient VM migration for disaster-resiliency. Section IV presents the case study and shows illustrative numerical results. Section V concludes the paper.

II. Background on VM Migration

For several purposes, DC operators already perform intra-DC VM migration, i.e., migration of VMs within a DC, or inter-DC VM migration, i.e., migration of VMs among inter-connected DCs [11]. Although intra-DC VM migration was first applied, mainly to conserve energy through consolidation of VMs in fewer physical servers, inter-DC VM migration gained recently more importance for purposes such as load balancing and enhancement of quality of experience [12], improving the overall network energy consumption [13].

VM migration requires transferring all VM data, i.e., disk, memory and processors states, from a source to a destination server. A possible baseline approach for VM migration is the *offline VM migration*, which consists in halting the VM at the original server, transferring all its data, and then re-activating it at the destination server. However, in this approach, the VM cannot be accessed while the migration process is taking place, which results in service interruption (i.e., service downtime) which might be intolerable in some cases [14]. More precisely, given a VM with size V^M = 10 Gbit to be transferred, if migration is performed using a migration bandwidth B = 1000 Mbit/s, the VM migration duration will be = T_{mig} = V^M/B = 10 seconds, and this roughly coincides with the service downtime. While such a migration duration may be short enough to migrate the VM before a disaster occurrence, the 10-seconds service downtime remains intolerable.

To avoid this drawback, DC operators and service providers resort to the concept of *online, or live VM migration* [15], where the data transfer process takes place while the VM is still running, thus significantly reducing service downtime [16], [17]. However, during the online migration, the VM memory is "*dirtied*" (i.e., modified) due to users activity, and therefore additional information (the *dirtied* data) needs to be transferred in an iterative process together with the original VM state to make sure the VM at the destination server

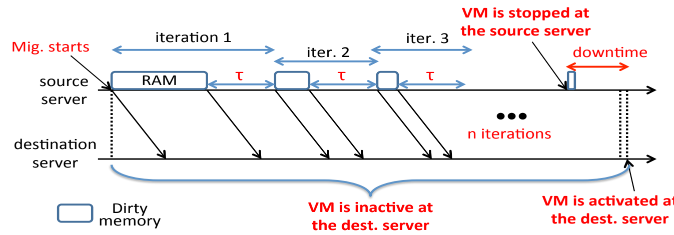


Fig. 1. Example of the iterative-copy phase during a VM migration.

is synchronized with the one at the source server. A main drawback of this approach is the increase in network resource occupation (e.g., bandwidth assigned and connection duration) with respect to the offline VM migration. Moreover, since memory pages are modified during the online VM migration, a non-linear relationship between the bandwidth assigned to migrate a VM and the migration duration arises, meaning that a baseline bandwidth assignment of online migration process might significantly penalize the migration duration and/or the network resource occupation. The parameter capturing this aspect is called *memory dirtying rate* D, the rate at which the VM memory is modified during the migration process. It might vary based on the type of VM, its hosted applications, as well as the number of served users and their activity. We refer to D in bit/s. Note that, unlike the offline migration, the total migration duration T_{mig} not only depends on the provisioned bandwidth, B_{mig}, but also on the dirtying rate D.

Figure 1 shows how online migration works. The first iteration is used to transfer the original VM memory to the destination, while following iterations are used to transfer the "dirtied" memory, i.e., the memory blocks that were modified by users. The duration of each iteration depends on the amount of memory dirtied during the previous iteration and the migration bandwidth assigned. Moreover, an inter-iteration delay τ is also shown in the figure, which is due to end-to-end network delay, processing delay at either end, or a combination of both. The iterative copy phase stops when a specific stop condition is met. The stop condition could be a predefined number of iterations or the amount of dirtied memory low enough to meet with a targeted maximum downtime. During the final stop-and-copy phase, the VM is stopped at its source, remaining dirtied memory is copied, and the network is re-configured before bringing the VM up at the destination location.

In addition, we note that the VM migration process cannot converge in case the bandwidth assigned for the migration is less than the dirtying rate D[1]. In other words, the lower bound for the migration bandwidth is the VM dirtying rate, D. However, in our evaluations, we assume, as in Ref. [18], that the minimum migration bandwidth possible is B_{min} = $1.2 \cdot D$. Similarly, we set a limit on the maximum migration bandwidth B_{min} we can allocate when $\frac{\Delta T_{mig}}{\Delta B} \leq 0.1\%$, i.e., when the advantage in terms of migration duration is very low.

[1]We assume that D is constant for the entire duration of a VM migration. This can be interpreted as a worst-case scenario, where D is the maximum possible value for that specific VM.

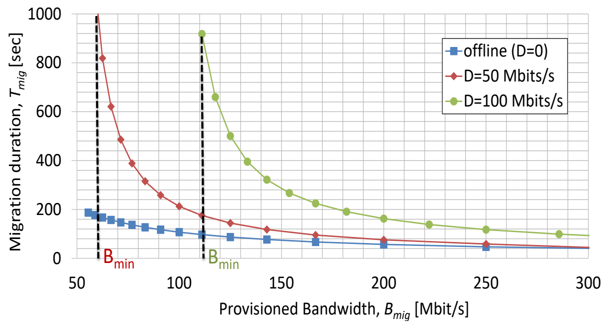


Fig. 2. Function points curves for a VM of size V^M = 10 Gbit and variable dirtying rates: D = 0 (offline), D = 50 Mbit/s and D = 100 Mbit/s.

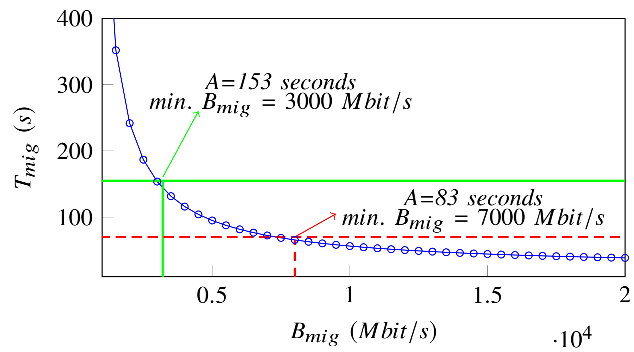


Fig. 3. Example of the effect of the alert time (evacuation time) on the set of possible bandwidth values to perform the migration for a VM of size = 40 GB and dirtying rate D = 0 (offline) and D = 500 Mbit/s.

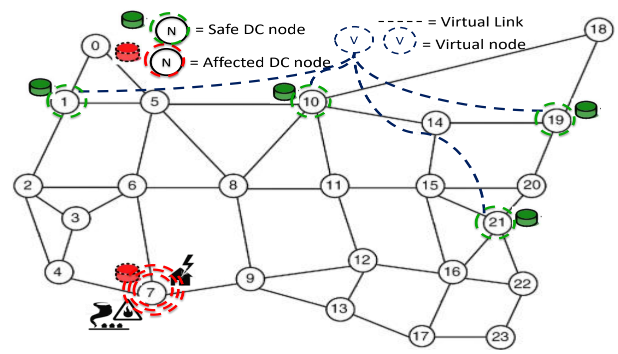


Fig. 4. USA24 Network Topology. DC nodes, virtual node and affected DC are highlighted.

The dependency between T_{mig} and B_{mig} is represented in the function-points curves shown in Fig. II. The 3 curves correspond to the migration of a VM with size V^M = 10 Gbit and for different dirtying rate values, i.e., D = 0 (offline migration), D = 50 Mbit/s and D = 100 Mbit/s. We used the model in [6] to obtain these curves, which in the following will be referenced to as *function-points curves* (refer to Ref. [18] for a detailed explanation). On one hand, we note that as the provisioned bandwidth increases, the advantage in terms of total migration duration is reduced. On the other hand, reducing B produces a drastic increase in the total migration duration, due to the high number of iterations needed for the migration. For all these reasons, the choice of the bandwidth to be associated to the VM migration is not trivial.

Moreover, we show in Fig. 3 the function-points curve and an example of the set of possible bandwidth values to perform VM migration in a weather-based disaster scenario for deadlines of 153 and 83 seconds for $D = 500\ Mbit/s$. First, we highlight how the deadline imposes a lower bound on the bandwidth values applicable to perform the migration process. For example, for an alert time of 153 seconds, the minimum value of migration bandwidth which can be considered is *min*. B_{mig} = 3000 Mbit/s while when the deadline becomes more strict, e.g., 83 seconds (represented in a dashed line in Fig. 3), a smaller set of bandwidth values (with much higher values) shall be assigned to perform the migration with *min*. B_{mig} = 7500 Mbit/s. Thus, the alert time imposes more constraints on the management of the migration process of all VMs more difficult, specially when the number of VMs to be migrated online is high.

In addition, as the amount of network resources occupied due to online VM migration is the product of T_{mig}, B_{mig} and the number of links traversed, and since a non-linear relationship exists between T_{mig} and B_{mig} (due to the dirtying rate), an efficient migration bandwidth assignment, which guarantees evacuation of VMs within the alert time, is needed to avoid excessive and undesired network resource occupation. Therefore, efficient migration bandwidth assignment and scheduling of VMs migration are decisive to maximize the number of VMs migrated, reduce service downtime and minimize network resource occupation.

III. ILP-based Disaster-Resilient Online VM Migration

A. Problem Statement

The problem is addressed in this paper is referred to as *"Alert-based Online Migration for Disaster-Resilience"*. The problem can be stated as follows, **given** a physical wide area network topology (as in Fig. 4), a DC affected by a disaster, a set of candidate destination DCs, a set of VMs running at the affected DC, their function-point curves and an alert time (i.e., evacuation deadline), we **decide** i) destination DC of a VM, ii) migration bandwidth of the VM and iii) scheduling (i.e., starting time) of all VMs migration with the **objective** of 1) maximizing number of VMs migrated, 2) minimizing service downtime (i.e., maximizing online VM migration when possible), 3) minimizing network resources and 4) minimizing average migration downtime.
In this paper we address the problem via an ILP formulation which is described in the following.

B. Sets and Parameters

- Graph $\mathcal{G} = (N, E)$ models the physical network topology, where N represents the set of nodes and E the set of bidirectional high-capacity optical links of capacity C.
- The subsets N_s, N_d and $N_t \subseteq N$ represent the affected DC node, the set of candidate DC locations which can host VMs and the set of transit nodes, respectively.

- V is the set of VMs to be migrated from the affected DC, each with size S_v.
- For each VM $v \in V$, there are two sets, B^v and D^v, representing the online migration bandwidth and migration duration values, corresponding to the function-points curve of VM v. $b^{on}_{i,v}$ and $d^{on}_{i,v}$ represent the bandwidth value and the migration duration of point i of the function-points curve of VM v.
- T is the set of time intervals, each with an integer index k_t.
- A, an integer parameter representing the alert time (evacuation deadline).
- C, an integer parameter representing the amount of storage capacity available at the destination DC to host VMs.
- α, β, γ and δ: constant values utilized to set the priority of the different terms in the objective function.
- M: a large integer value.

C. Variables

- x_v: binary variable equal to 1 if VM $v \in V$ is migrated online, 0 otherwise.
- y_v: binary variable equal to 1 if VM $v \in V$ is migrated offline, 0 otherwise.

Note that two different binary variables are used to specify if a VM is migrated online or offline as there exist as well a third possibility which is a VM not being migrated.
- $l_{v,i}$: binary variable indicating if point i of the function-points curve of VM v (represented by set B_v and D_v) is assigned to the migration of VM $v \in V$.
- b^{on}_v and d^{on}_v: integer variables (≥ 0), representing the bandwidth value assigned to the online migration of VM $v \in V$ and its corresponding migration duration, respectively.
- b^{off}_v and d^{off}_v: integer variables (≥ 0) corresponding to the bandwidth value assigned and the migration duration of the offline migration of VM $v \in V$, respectively. Note that d^{off}_v is calculated by the ratio $\frac{S_v}{b^{off}_v}$ in case y_v equal to 1 whereas it is equal to 0 otherwise.
- ψ_v: integer variable (≥ 0) representing the downtime of VM $v \in V$. **Note that the $\psi_v = d^{off}_v$ in case VM v is migrated offline (i.e., in case $y_v = 1$) whereas $\psi_v = 0$ otherwise.**
- r_v and e_v: integer variables (≥ 0) representing the starting time and the ending time of the migration process of VM $v \in V$, respectively.
- ω_v and λ_v: integer variables (≥ 0) representing the bandwidth assigned and the duration of the migration process of VM $v \in V$, independent of the type of the migration process, respectively.
- $w_{v,t}$: binary variable equal to 1 if $v \in V$ is migrated at time slot $t \in T$.
- $h^{i,j}_v$: binary variable equal to 1 if the migration process of $v \in V$ is using link $(i,j) \in E$.
- $z^{i,j}_{v,t}$: binary variable equal to 1 if the migration process of $v \in V$ is performed at time slot $t \in T$ using link $(i,j) \in E$.

D. Objective Function

The objective function of the ILP is expressed as follows:

$$\min\Bigg(\sum_{v\in V}\big(-\alpha(x_v+y_v)+\beta\psi_v+\sum_{(i,j)\in E}\sum_{t\in T}\gamma z^{i,j}_{v,t}\omega_v+\delta\lambda_v\big)\Bigg)$$

The first term of the objective function accounts for the number of VMs migrated, either online or offline. Although x_v and y_v have the same weight in the objective function, we implicitly give priority to the online migration (i.e., we maximize $\sum x_v$) as we minimize, as a second objective, the service downtime, ψ_v. In fact, minimizing the total service downtime for all the migrated VMs corresponds to performing online VM migration as much a possible. Note that the first term is multiplied by a negative sign as the optimization objective is to be minimized, whereas we want to maximize the number of VMs migrated. The third term, $\sum_{(i,j)\in E}\sum_{t\in T} z^{i,j}_{v,t}\cdot\omega_v$, accounts for the average Resource Occupation (RO). Note that this term can be represented as a unique variable that can be linearized. Finally, the fourth term accounts for the migration duration for VM v, represented by λ_v.

α, β, γ and δ in the objective function are positive constants and are utilized to set the priorities of the different terms. However, the priority of the first two terms is not changed as the main objective is to maximize the number of VMs migrated and minimize service downtime, we consider two different objectives, namely: i) *RO-minimized* where we prioritize the minimization of the average network resources required to perform the migration of all VMs, represented by RO_{avg}, and ii) *T-minimized* where we prioritize the minimization of the migration duration of each of the VMs, λ_v, and therefore the average migration duration of all VMs, represented by $T_{mig,avg}$.

E. Constraints

1) ***Migration Process and Bandwidth Constraints****:* Constraint 1 guarantees that a VM can be migrated once at maximum (either offline or online). Const. 2 assures that if a VM is migrated online, a function point corresponding to a bandwidth value is chosen of the function-points curve of VM v. Correspondingly, Consts. 3 and 4 assign the bandwidth and the migration duration to an online migration of VM v. In case VM v is migrated offline, Const. 5 assigns a bandwidth value to an offline migration process and equality 6 calculates the migration duration accordingly.

$$x_v + y_v \leq 1, \quad \forall v \in V \tag{1}$$

$$\sum_{i\in I} l_{v,i} = x_v, \quad \forall v \in V \tag{2}$$

$$b^{on}_v = \sum_{i\in I} b^{on}_{i,v}\cdot l_{v,i}, \quad \forall v \in V \tag{3}$$

$$d^{on}_v = \sum_{i\in I} d^{on}_{i,v}\cdot l_{v,i}, \quad \forall v \in V \tag{4}$$

$$b^{off}_v \leq M\cdot y_v, \quad \forall v \in V \tag{5}$$

$$d_v^{off} = \frac{S_v}{b_v^{off}}, \quad \forall v \in V \tag{6}$$

For clarity, we show in Eqns. 7 and 8 the calculation of ω_v and λ_v, the bandwidth value and the migration duration of the migration process of VM v, independent if migrated online or offline.

$$\omega_v = \sum_{i \in I} b_{i,v}^{on} \cdot l_{v,i} + b_v^{off}, \quad \forall v \in V \tag{7}$$

$$\lambda_v = \sum_{i \in I} d_{i,v}^{on} \cdot l_{v,i} + \frac{S_v}{b_v^{off}}, \quad \forall v \in V \tag{8}$$

*2) **Scheduling Constraints**:* Const. 9 guarantees that the ending time of any VM migration has to be before the alert time and Const. 10 guarantees that the ending time of a VM migration is after the starting time by a duration exactly equal to the migration duration.

$$e_v \leq A, \quad \forall v \in V \tag{9}$$

$$e_v = r_v + \lambda_v, \quad \forall v \in V \tag{10}$$

Note that the scheduling problem dealt with is different than the traditional ones since the migration duration is not known a priori and is, in fact, a decision variable in the problem. For this reason, we introduce two binary variables, $n_{v,t}$ and $m_{v,t}$. Consts. 11-14 guarantee that $n_{v,t} = 1$ for $k_t \geq r_v$ and that $m_{v,t} = 1$ for $k_t \leq e_v$. Consts. 15 and 16 assure that a VM is migrated at a time t if $n_{v,t} \cdot m_{v,t} = 1$.

$$k_t - r_v + 1 \leq M \cdot n_{v,t}, \quad \forall v \in V, t \in T \tag{11}$$

$$e_v - k_t \leq M \cdot m_{v,t}, \quad \forall v \in V, t \in T \tag{12}$$

$$A - r_v = \sum_{t \in T} n_{v,t}, \quad \forall v \in V \tag{13}$$

$$e_v = \sum_{t \in T} m_{v,t}, \quad \forall v \in V \tag{14}$$

$$w_{v,t} \leq n_{v,t} \cdot m_{v,t}, \quad \forall v \in V, t \in T \tag{15}$$

$$\sum_{tinT} w_{v,t} = \lambda_v, \quad \forall v \in V \tag{16}$$

$$z_{v,t}^{i,j} \geq w_{v,t} \cdot h_v^{i,j}, \quad \forall v \in V, t \in T, (i,j) \in E \tag{17}$$

Constraint 17 guarantees that $z_{v,t}^{i,j} = 1$ if v is migrated at time instance t (i.e., for $w_{v,t} = 1$) and is utilizing link (i,j) $\in$ E (i.e., for $h_v^{i,j} = 1$).

*3) **Flow and Capacity Constraints**:* Constraints 18, 19 and 20 show the flow constraints for the source DC (affected DC), destination DCs and the transit nodes. Note that the VM migration process may possibly utilize different paths at different time instants throughout the migration process but the destination DC has to be always the same. Finally, Const. 21 and 22 represent the link capacity and the DC capacity constraints, respectively.

$$\sum_{(n,j) \in E : n \in N_s} z_{v,t}^{n,j} = w_{v,t}, \quad \forall v \in V, t \in T \tag{18}$$

$$\sum_{(i,d) \in E : d \in N_d} z_{v,t}^{i,d} = w_{v,t}, \quad \forall v \in V, t \in T \tag{19}$$

$$\sum_{(i,j) \in E : i \in N_t, j \in N_t} z_{v,t}^{i,j} = w_{v,t}, \quad \forall v \in V, t \in T \tag{20}$$

$$\sum_{t \in T} \sum_{v \in V} \omega_v \cdot z_{v,t}^{i,j} \leq C_{i,j}, \quad \forall (i,j) \in E \tag{21}$$

$$\sum_{vinV} k_{v,d} * S_v \leq A, \quad \forall d \in N_d \tag{22}$$

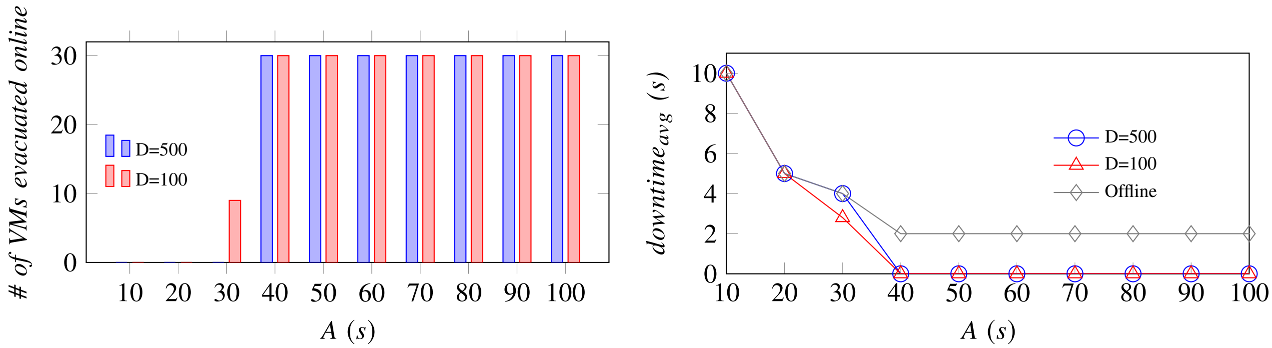
IV. Illustrative Numerical Results

A. Case Study

The topology considered in this study is the USA-24 network shown in Fig. 4, constituted by $|N| = 24$ nodes and $|E| = 43$ bidirectional links, each with $C = 100$ Gbit/s capacity in both directions. We consider 5 DC locations in which one is affected by a weather-based disaster (highlighted in Fig. 4). Note that the results are general and do not depend on the choice of the affected DC. We consider 30 VMs each of a size of 40 *GB*. Indeed, a DC may host thousands of VMs however we consider such a case study to compare the proposed strategies. In the evaluations, we consider different values of the dirtying rate with the objective of analyzing the effect of the dirtying rate on the VM evacuation process. The values of the dirtying rate D considered are 100 Mbit/s and 500 Mbit/s assuming cases in which around 3000 and 15000 memory pages of a VM are modified per second. Moreover, we consider short disaster alerts and thus we perform the evaluations for an alert time A ranging from 10 to 100 seconds.

We evaluate the performance of *Alert-based Online Migration* strategies, namely, *RO-minimized* and *T-minimized*, and compare them to a benchmark *Offline* strategy, where all VMs are migrated in an offline manner such as to perform the evacuation in a time as short as possible but on the account of causing a service outage of the services run by the VMs. We consider the following metrics for our evaluation:

- Total number of VMs migrated
- Average downtime per VM $downtime_{avg}$
- Average migration duration $T_{mig,avg}$
- Average migration bandwidth $B_{mig,avg}$
- Average data transferred DT_{avg}, $DT_{avg} = B_{mig,avg} \cdot T_{mig,avg}$



(a) # VMs migrated vs. alert time

(b) Downtime vs. alert time

Fig. 5. The number of VMs evacuated online (a) and (b) the average downtime for the online strategies and the offline approach for D = 100 Mbit/s and D = 500 Mbit/s as a function of the alert time (s).

- Average resource occupation RO_{avg} representing the average amount of network resources occupied to perform the migration of one VM.
- Average number of hops traversed $hops_{avg}$

Note that both strategies, *RO-minimized* and *T-minimized* have the same behavior in terms of the number of VMs migrated and $downtime_{avg}$ as the variables corresponding to these metrics are prioritized in both strategies.

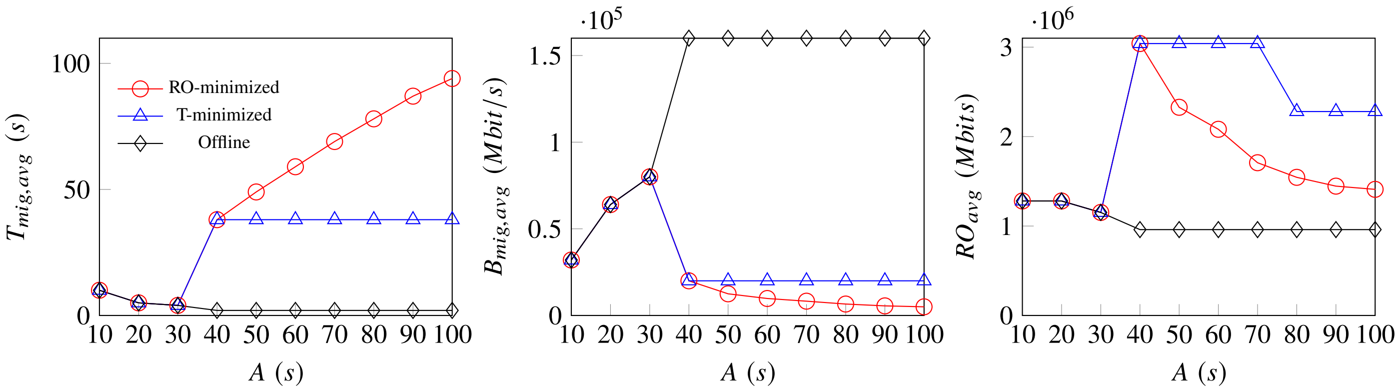
B. Discussion

1) Effect of dirtying rate on downtime: Figure 5(a) shows the number of VMs migrated online (out of the 30 VMs) for the *Online* strategy for $D = 100$ $Mbit/s$ and $D = 500$ $Mbit/s$ as a function of the alert time. Note that here we do not differentiate between *RO-minimized* and *T-minimized* as both strategies exhibit the same performance in terms of number of VMs migrated and service downtime. For both values of D, none of the VMs is migrated online for $A \leq 20$ s. This is because an online migration of a VM in such a stringent alert time requires high migration bandwidth which the network capacity is not able to accommodate for all VMs. For $A = 30$ s, 9 VMs are migrated online (while 21 VMs migrated offline) for $D = 100$ $Mbit/s$, while none of the VMs is migrated online for $D = 500$ $Mbit/s$. This demonstrates that higher dirtying rate (i.e., higher user activity) affects the online migration process as it requires higher bandwidth and longer migration duration to be performed. Moreover, for $A \geq 30$ s and for both values of D, all 30 VMs are migrated online. This shows that, given a number of VMs, their size, dirtying rate and network capacity, there exist a threshold on the minimum amount of time required to perform online migration for all VMs to be migrated from a DC affected by a disaster. Moreover, in Fig. 5(b) we show the average downtime per VM, $downtime_{avg}$, for the online strategy (for $D = 100$ $Mbit/s$ and $D = 500$ $Mbit/s$) and the *Offline* strategy. As expected, we see that for $A \leq 20$ s, all strategies have an equal $downtime_{avg}$ as only offline VM migration is possible. However, even when all the VMs are migrated offline, when A increases, $downtime_{avg}$ decreases. This is because a higher alert time allows utilizing higher migration bandwidth values for the offline VM migration and performing more efficient scheduling. Moreover, for $A \geq 30$ s, the online strategy with $D = 100$ $Mbit/s$ exhibits lower $downtime_{avg}$ than the case where $D = 500$ $Mbit/s$, as a lower D allows to migrate some VMs online. Finally, as seen in Fig. 5(a), for $A \geq 40$ s, all VMs are migrated online thus exhibiting no service downtime ($downtime_{avg} = 0$), i.e., eliminating service downtime. Note that for the *Offline* strategy, $downtime_{avg}$ remains constant and greater than 0 even for higher values of A as this is the minimum amount of time to perform the offline migration of the VM.

2) RO-minimized vs. T-minimized: Now, we compare the *RO-minimized*, *T-minimized* and *Offline* strategies in terms of $T_{mig,avg}$, $B_{mig,avg}$, RO_{avg}, DT_{avg} and $hops_{avg}$ considering $D = 500$ $Mbit/s$. We concentrate our analysis for $A \geq 30$ s, as for values of $A < 30$ s the behavior of the different strategies is similar as only offline VM migration is performed. Fig. 6(a) shows that, for $A = 40$ s, $T_{mig,avg}$ increases up to around 40 seconds for both *RO-minimized* and *T-minimized* online strategies. We notice that, for $A = 40$ s, the performance of both strategies coincide, due to the fact that for the given alert time and the given function-point considered (where we considered a maximum bandwidth value possible, namely, B_{max}), to perform online VM migration of all VMs, all VMs migration need to be performed at the maximum provisioned bandwidth, i.e., B_{max}.

However, for $A \geq 50$ s, $T_{mig,avg}$ increases progressively for *RO-minimized* while it remains constant for *T-minimized*. This is because the *RO-minimized* strategy tends to utilize lower bandwidth values (as seen in Fig. 6(b)) if this provides lower RO while *T-minimized* utilizes the maximum migration bandwidth value possible for online migration to maintain a minimal $T_{mig,avg}$. Indeed, we see in Fig. 6(c) that RO_{avg} for *RO-minimized* decreases progressively for higher values of A as this strategy utilizes migration bandwidth values which result in lower RO. On the contrary, yet surprisingly, RO_{avg} decreases slightly for *T-minimized* for $A \geq 80$ s however it remains greater than that of *RO-minimized*. As for the *Offline* strategy, it utilizes high bandwidth values to minimize $T_{mig,avg}$ to 2 seconds (which coincides with the downtime) and as expected, it exhibits the lowest RO.

To examine in detail the reason behind the behavior of

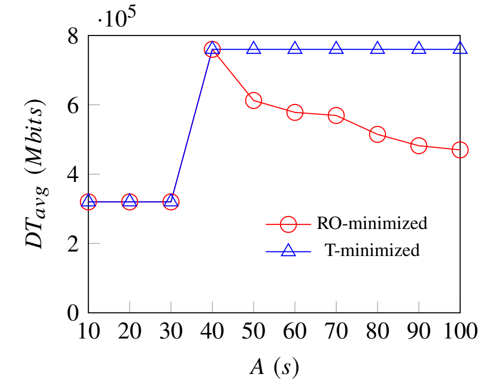


(a) $T_{mig,avg}$ *vs. alert time* (b) $B_{mig,avg}$ *vs. alert time* (c) RO_{avg} vs. *alert time*

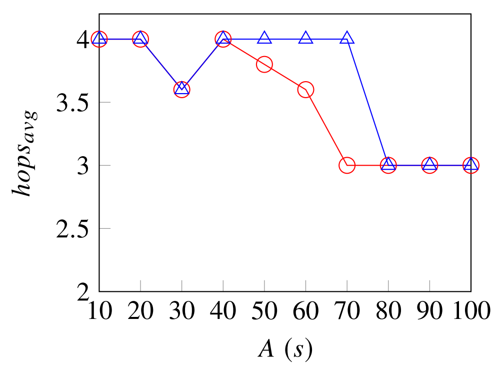
Fig. 6. The average migration duration (a), average migration bandwidth assigned (b) and (c) average resource occupation per VM as a function of the alert time for D = 500 $Mbit/s$.

RO-minimized and *T-minimized*, we show in Figs. 7(a) and 7(b) the two contributions of RO_{avg}, i.e., DT_{avg} and $hops_{avg}$, respectively. As expected, DT_{avg} decreases for *RO-minimized*, as, for increasing A, there is more opportunity to use function points which allows RO-minimization. However, DT_{avg} remains constant for *T-minimized*, as this strategy utilizes the maximum bandwidth value possible to perform online VM migration while minimizing $T_{mig,avg}$. In Fig. 7(b) we see that $hops_{avg}$ decreases more evidently for higher values of A in case of *RO-minimized* with respect to *T-minimized*. This is due to the fact that *RO-minimized* strategy saves on network resources and tends to use lower bandwidth values in case this makes it possible to migrate VMs to closer DCs, and thus to save more on network resources through occupying resources on fewer network links. On the contrary, *T-minimized* utilizes the maximum bandwidth value possible to guarantee that the VM migration is performed with the minimum $T_{mig,avg}$ and this does not allow to migrate VMs towards close DCs due to the presence of bandwidth bottlenecks in the proximity of closest DCs. On one hand, this shows that as *RO-minimized* strategy saves on network resources (around 40% with respect to *T-minimized*) through assigning migration bandwidth values which allow more efficient scheduling of VMs such as to migrate VMs towards closer DCs, thus minimizing RO. On the other hard, minimizing $T_{mig,avg}$ is achieved on the account of utilizing more network resources. This shows that there exist trade-offs between minimizing the downtime and network resource occupation on one side and minimizing the VM migration duration and the network resource occupation on another side.

3) Effect of dirtying rate on Resource Occupation: To examine more in detail the effect of D on the migration process, we compare the performance of *RO-minimized* strategy for D = 100 $Mbit/s$ and D = 500 $Mbit/s$. First of all, we see in Fig. 8(a) that DT_{avg} in the case D = 100 $Mbit/s$ increases to its peak at A = 30 s while in the case D = 500 $Mbit/s$ it reaches its peak at A = 40 s, meaning that online VM migration is performed earlier for the case when D is lower. Then, as A increases, DT_{avg} decreases for both values of D. However, it decreases more rapidly for D = 100 $Mbit/s$, which in turn



(a) DT_{avg} vs. *alert time*



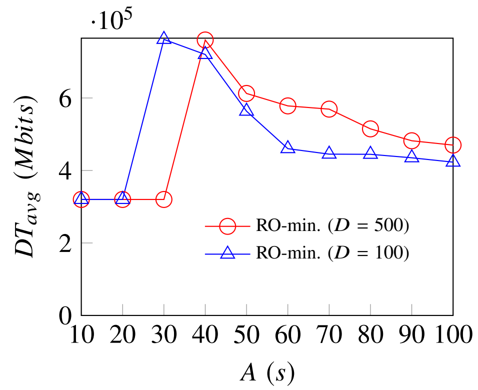
(b) $hops_{avg}$ *vs. A*

Fig. 7. The average data transferred per VM migration (a) and (b) average number of hops as a function of the alert time for D = 500 $Mbit/s$.

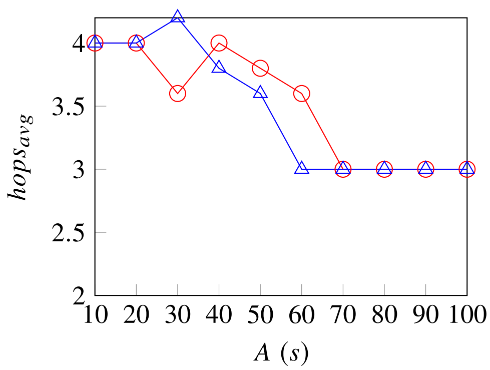
allows to migrate all VMs to the closest DC, thus achieving $hops_{avg}$ = 3 hops at a lower value of A than in the case where D = 500 $Mbit/s$ (as see in Fig. 8(b)). This is because a lower dirtying rate allows for more flexibility in assigning efficient (i.e., with lower RO) migration bandwidth values and thus efficient scheduling of VMs migration.

V. Conclusion

In this paper we proposed an ILP model for efficient online VMs migration for disaster resiliency in an inter-



(a) DT_{avg} vs. $alert\ time$



(b) $hops_{avg}$ vs. $alert\ time$

Fig. 8. The average data transferred per VM migration (a), (b) average number of hops and (c) average resource occupation per VM migration as a function of the alert time for the $RO-minimized$ strategy for $D = 500\ Mbit/s$ and $D = 100\ Mbit/s$.

data center network. With the proposed ILP, we model the routing and bandwidth assignment as well as the scheduling of the VMs migration from a DC affected by a disaster to other DCs within an alert time. Specifically, we proposed two strategies with the objective of maximizing the number of VMs evacuated and minimizing the average service downtime while minimizing the network resource occupation, *RO-minimized*, or while minimizing the migration duration, *T-minimized*. We performed evaluations to quantify the trade-off between service downtime, migration duration and overall network resource occupation. Moreover, we investigated the effect of the memory-dirtying rate on the online migration process. Results show that, i) given the size of the VMs, the dirtying rate, the time available and the network capacity available, there exist a threshold on minimum amount of time required to perform evacuate all VMs online, thus eliminating service downtime, ii) our proposed strategy, *RO-minimized*, is capable of assigning route and migration bandwidth and scheduling VMs efficiently such as to migrate VMs to the closest data centers and eventually saving on the overall network resource occupation and that iii) the dirtying rate is a decisive parameter to be considered due to its effect on the online migration process specially in an alert-based disaster resilient scenario where the duration available to perform the migration is limited.

ACKNOWLEDGMENT

This article is based upon work from COST Action CA15127 (Resilient communication services protecting end-user applications from disaster-based failures - RECODIS) supported by COST (European Cooperation in Science and Technology).

REFERENCES

[1] V. Cisco, "Cisco visual networking index: Forecast and methodology 2016–2021." 2017.

[2] S. Ferdousi, M. Tornatore, M. F. Habib, and B. Mukherjee, "Rapid data evacuation for large-scale disasters in optical cloud networks," *Journal of Optical Communications and Networking*, vol. 7, no. 12, pp. B163–B172, 2015.

[3] S. Ferdousi, F. Dikbiyik, M. F. Habib, M. Tornatore, and B. Mukherjee, "Disaster-aware datacenter placement and dynamic content management in cloud networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 7, pp. 681–694, 2015.

[4] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper, "Predicting the Performance of Virtual Machine Migration," in *IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2010.

[5] M. Bari, M. Zhani, Q. Zhang, R. Ahmed, and R. Boutaba, "CQNCR: Optimal VM Migration Planning in Cloud Data Centers," in *IFIP Networking Conference, 2014*.

[6] U. Mandal, P. Chowdhury, M. Tornatore, C. U. Martel, and B. Mukherjee, "Bandwidth Provisioning for Virtual Machine Migration in Cloud: Strategy and Application," *in IEEE Transactions on Cloud Computing*, 2016.

[7] O. Ayoub, L. Pace, F. Musumeci, and A. Pattavina, "Dynamic Routing and Bandwidth Assignment for Live Virtual Machines Migrations," in *20th International Conference on Optical Network Design and Modeling (ONDM)*. Cartagena, 2016.

[8] W. Cerroni, "Multiple Virtual Machine Live Migration in Federated Cloud Systems," in *IEEE Conference on Computer Communications Workshops (INFOCOM Workshops)*, 2014.

[9] J. Zhang, F. Ren, and C. Lin, "Delay Guaranteed Live Migration of Virtual Machines," in *IEEE Conference on Computer Communications (INFOCOM)*, 2014.

[10] G. Sun, D. Liao, V. Anand, D. Zhao, and H. Yu, "A New Technique for Efficient Live Migration of Multiple Virtual Machines," *Future Generation Computer Systems*, vol. 55, pp. 74–86, 2016.

[11] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding Data Center Traffic Characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 92–99, 2010.

[12] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," in *10th IFIP/IEEE International Symposium on Integrated Network Management*, pp. 119–128, 2007.

[13] A. Gupta, U. Mandal, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Cost-Efficient Live VM Migration Based on Varying Electricity Cost in Optical Cloud Networks," *Photonic Network Communications*, vol. 30, no. 3, pp. 376–386, 2015.

[14] P. D. Patel, M. Karamta, M. Bhavsar, and M. Potdar, "Live Virtual Machine Migration Techniques in Cloud Computing: A Survey," *International Journal of Computer Applications*, vol. 86, no. 16, 2014.

[15] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 273–286.

[16] K. Ye, X. Jiang, D. Huang, J. Chen, and B. Wang, "Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments," in *2011 IEEE International Conference on Cloud Computing (CLOUD)*. Washington DC, pp. 267–274, 2017.

[17] T. Wood, K. Ramakrishnan, J. Van Der Merwe, and P. Shenoy, "Cloudnet: A Platform for Optimized WAN Migration of Virtual Machines," *University of Massachusetts Technical Report TR-2010-002*, 2010.

[18] O. Ayoub, F. Musumeci, M. Tornatore, and A. Pattavina, "Efficient routing and bandwidth assignment for inter-data-center live virtual-machine migrations," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 3, pp. B12–B21, 2017.