The final publication is available at Springer via http://dx.doi.org/10.1007/s11107-015-0498-y

Orchestrating Connectivity Services to support Elastic Operations in Datacenter Federations

Adrian Asensio*, Marc Ruiz, and Luis Velasco
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain.
*Corresponding author: aasensio@ac.upc.edu

Abstract—Datacenter federations are able to manage appropriately the green energy resources available in each datacenter (DC) thanks to their geographically distributed infrastructure, thus reducing energy expenditure. Scheduling algorithms can compute virtual machine migration, transferring huge amounts of raw data from one DC to another to minimize operational costs and ensuring a certain Quality of Experience (QoE). Because green energy availability greatly depends on weather conditions, in this work we present a statistical model to improve green solar energy availability estimation accuracy and we use it in a mixed integer linear programming (MILP) formulation to compute optimal virtual machine placement. Optical connections can be used to provide connectivity services of enough capacity to support those migrations. In particular, elastic optical networks can provide connections with multi-granular bitrate, which can be adapted on demand. DC resource managers can request optical connections and control their capacity. However, that scheme involves the resource managers to implement algorithms and interfaces to deal with network specifics and complexity. To solve that issue, in this paper we propose coordinating transfer-based inter-DC connectivity services; inter-DC connectivity is requested in terms of volume of data and completion time. We analyze cost savings when each connectivity model is applied in a DC federation. For the sake of a compelling analysis, exhaustive simulation experiments are carried out considering realistic scenarios. Results show that the notification-based model can save up to 20% of energy costs and more than 40% of communication costs in the evaluated scenarios.

***Keywords*: Federated datacenters, Energy costs minimization, Inter-datacenter networks.**

1 INTRODUCTION

The huge energy consumption of datacenters (DC) requires an elastic resource management, e.g. by turning servers off when they are not used or turning them on to satisfy increments in the demand. Thanks to virtualization, jobs (e.g., web applications) can be encapsulated in virtual machines (VM) and run in the most proper server according to their performance goals. The local resource manager can migrate VMs from one server to another looking for reducing energy consumption while ensuring the committed quality of experience (QoE) [1]. In that regard, the live-migration technique allows migrating VMs from one server to another without stopping them resulting in reduced downtimes.

Large Internet companies, such as Google, have their own IT infrastructures consisting in a number of large DCs placed in geographically diverse locations to guarantee the appropriate QoE to users; DCs are interconnected through a wide area network [2]. Using such infrastructure, workloads can be moved among DCs to take advantage of reduced energy cost during off-peak energy periods in some locations while using green energy when it is available in some other locations. Servers are turned off when they are not used, thus minimizing their energy expenditure. Nonetheless, smaller independently operated infrastructures cannot perform such elastic operations; notwithstanding, they can cooperate by creating DC federations [3] to increase their revenue from using IT resources that would otherwise be under-utilized, and to expand their geographic coverage without building new DCs.

Within a single DC, VM migrations for consolidation and load balancing purposes are commonly automated using *schedule-based* algorithms running in the local resource manager. These algorithms target at optimizing some utility function, ensuring quality of experience and service availability; its outcome is the set of VMs to be activated, stopped or migrated in the local DC. When a DC federation is created, scheduling algorithms need to consider not only local workload and resources but also those in the rest of federated DCs and compute VM migration towards remote DCs, as well as within the local DC.

In a recent global cloud index study [4], Cisco forecasts DC traffic to quadruple over the next years, reaching 554 EB per month by 2016. Two main components of traffic leaving DCs can be distinguished: traffic among DCs (DC2DC) and traffic between DCs and end users (DC2U). The former includes VM migration to manage the cloud elastically, whilst the latter is associated to applications, such as web, email, etc. Network providers can facilitate federated DCs interconnection by allowing them to request connections' setup on demand with the desired bitrate, while tearing down those connections when they are not needed. In the last years, huge standardization work has been done defining control plane architectures and protocols to automate connection provisioning. The IETF is defining the Application-Based Network Operations (ABNO) architecture [5], which is based on standard components such as the active stateful Path Computation Element (PCE).

Since DC2DC traffic is generated by VM migration among DCs, the connectivity required between two DCs highly varies along the day, presenting dramatic differences in an hourly time scale. In [6], we proposed to use flexgrid-based optical network to interconnect DCs, since that technology provides fine and multiple granularity. In flexgrid optical networks the available optical spectrum is divided into frequency slices of fixed spectrum width. Optical connections can be allocated into a variable number of these slices, and its capacity can be dynamically managed by allocating or releasing slices provided that the spectrum allocated to an optical connection remain contiguous [7].

1.1 Related work

In this work we assume a scenario where federated DCs interconnected by a flexgrid-based network, are strategically placed around the globe to provide worldwide high QoE services. Each DC obtains its required energy from green sources and from the electric grid (*brown energy*). Green energy can cover a given percentage of the total energy required in a DC (*green coverage*), while the rest of required energy is covered by the electric grid. In our previous work [8], we stated the Elastic Operations in Federated Datacenter for Performance and Cost Optimization (ELFADO) problem and studied two approaches to orchestrate a DC federation: distributed and centralized. We showed that the centralized approach takes full advantage from green energy availability in the federated DCs helping to minimize energy and communication costs (cost per GB transmitted through optical connections) while ensuring QoE. Thus, here we assume a centralized federation orchestrator computing periodically the global optimal placement for all the VMs in the federated DCs.

When deploying energy-efficient DCs, their internal architecture must be kept in mind. A certain number of switches are needed to provide connectivity between servers in the DC and to interface the DC with the Internet. Consequently, according to the DC architecture being adopted, a corresponding power is consumed, which basically depends on the number and type of switches used. Several intra-DC architectures have been studied in literature (see [9] for a detailed survey). Among them, the so-called *flattened butterfly* architecture has been identified as the most power-efficient DC architecture, since its power consumption is proportional to the number of currently used servers. However, the most widely-deployed architecture for DC is the so-called *fat-tree* topology [10], which is based on a hierarchical structure where large higher-order switches represent the interface of the DC towards the network infrastructure, and are connected to the servers via a series of lower-order switches, providing the intra-DC connectivity.

Many papers can be found in the literature addressing the energy expenditure minimization in DC management [11]-[14]. In [11], the authors propose scheduling workload in a DC coinciding with the availability of green energy, consolidating all the jobs on time slots with solar energy available, increasing green energy consumption up to 31%. Authors in [12] present a DC architecture to reduce power consumption, while guaranteeing QoE. They consider online-monitoring and VM placement optimization achieving energy savings up to 27%. Some works, e.g. [13], refer to the problem of load balance DC workloads geographically, following green energy availability, to reduce the amount of brown energy consumed focusing mainly on wind energy and the capability of store energy. Other authors focus on the importance of counting as “energy expenditure” every element in the DC, not only computing machinery. The author in [14] remarks the idea that all IT equipment counts when consuming energy, also the fluctuation of green energy production and energy transportation are important factors.

As elastic operations for VM migration require huge bitrate to be available among DCs for some time periods, the inter-DC network can be based on the optical technology and must provide automated interfaces to set-up and tear down optical connections with the required bitrate. Some works consider optical networks to interconnect DCs. For instance, routing algorithms considering both routing and scheduling are presented in [15] and energy savings are shown with respect to solving routing and scheduling problems separately. Aimed at minimizing the CO2 emissions of data centers by following the current availability of renewable energies, authors in [16] devise routing algorithms for connections supporting cloud services.

Some works using flexgrid networks to interconnect DCs are currently appearing in the literature. Authors in [17] propose an application controller that interfaces an OpenFlow controller for the flexgrid network, similarly to the approach followed by Google [2]. Notwithstanding, some network operators are supporting ABNO in the IETF, so there is a lack of consensus on the architecture; in this work we assume IETF’s architecture, supported by major European network operators [18].

1.2 Contributions

Because weather conditions severely impact green solar energy availability, in this paper we present a model to estimate the amount of green solar energy available in each location as a function of the specific time period and the weather conditions. Next, we leverage on that model to extend the centralized ELFADO problem defined in [8]. Once computed, resource managers are in charge of performing those VM migrations.

After the federation scheduler schedules the next period, local resource managers can start performing VM migration. To that end, they interface the interconnection network control plane as a way to automatize connection set-up and tear down with the required bitrate. Even though local resource managers can request optical connections set-up, tear down, and adapt their capacity on demand, the flexgrid interconnection network supports additional traffic for different services and clients. Therefore, competition for network resources could lead to connections capacity being reduced or even blocked at requesting time. In that case, resource managers can either perform connection request retries, similar to I/O polling in computers, to increase the bitrate of already established connections or set-up new ones, although without guarantees of success, resulting in a poor cloud performance.

To alleviate to some extent the dependency between cloud management and network connectivity, in this paper we propose a novel connectivity model named as *notification-based*. An abstraction layer on the top of ABNO, the Application Service Orchestrator (ASO), could be deployed. The ASO implements a northbound interface to request transfer operations. Those applications' operations are transformed into network connection requests. The northbound interface uses application-oriented semantic, liberating application developers from understanding and dealing with network specifics and complexity.

The rest of this paper is organized as follows. Section 2 describes a power model for intra-DC architecture and the proposed architecture for the federated DC, proposes a model to estimate the amount of green solar energy availability and presents a stochastic mixed integer linear programming (MILP) formulation that uses the proposed model to minimize energy expenditure costs. Section 3 is based on our previous work in [19]. It describes the architecture for the federated DC to interface the ASO, which, in turn, interfaces ABNO in charge of the interconnection network and presents two connectivity models, polling-based and notification-based, taking advantage of ABNO and ASO respectively. Illustrative results are provided in Section 4 to compare polling-based and notification-based connectivity models. We show that the latter reduces both energy and communication costs compared to the former. Finally, Section 5 concludes the paper.

2 MINIMIZING ENERGY EXPENDITURES

2.1 Datacenter power model

Power consumption in DCs comes mainly from IT devices, P_{IT}, which includes both the servers and the switches connecting them, and from non-IT equipment, $P_{non\text{-}IT}$, such as cooling, power supplies and power distribution systems. Although P_{IT} can be easily estimated, it is difficult to evaluate the power consumption of non-IT devices since it depends on factors which cannot be easily estimated, such as the geographical location or the building hosting that DC. An indirect way to estimate $P_{non\text{-}IT}$ is to consider the Power Usage Effectiveness (PUE) metric [20]. PUE can be used as a measure of the energy efficiency of a DC and quantifies the amount of power consumed by non-IT equipment in that DC. Therefore, if P_{IT} and PUE can be estimated for a given DC, the total power consumed in a DC can be computed as $P_{DC}=PUE*P_{IT}$. To compute P_{IT}, the power consumed by both the servers and network equipment must be considered. The power consumption of a server depends on the CPU utilization as function of the load, and can be estimated as

$$P_{server}(load) = P_{server\text{-}idle} + (P_{server\text{-}max} - P_{server\text{-}idle})*load \quad (1)$$

where *load* represents the ratio between the current load and the maximum capacity of the server, whilst $P_{server\text{-}idle}$ and $P_{server\text{-}max}$ represent the power consumed by the server when it is idle and when it operates at its maximum capacity, respectively [15]. The power consumed by network equipment depends on the specific architecture of the DC. In this work, we assume the *fat-tree* architecture [10], which consists of three switching layers; from top to bottom: *Core*, *Aggregation* and *Edge*. The aggregation and edge layers together with the servers are organized in a number of clusters M. Each cluster has $M/2$ edge switches, $M/2$ aggregation switches, and $M^2/4$ servers. In addition, there are $M^2/4$ M-port core switches, each having one port connected to each cluster, whilst each cluster is connected to every core switch. Then, power consumption of the IT devices in the DC can eventually be computed as follows, where P_{core}, P_{agg}, and P_{edge} denote power consumption of core, aggregation, and edge switches, respectively and $load_s^i$ the load in server s, which is in cluster i.

$$P_{IT} = \frac{M^2}{4} \cdot P_{core} + \sum_{i=1}^{M} \left[\frac{M}{2} \cdot \left(P_{agg} + P_{edge}\right) + \sum_{s=1}^{M^2/4} P_{server}(load_s^i) \right] \quad (2)$$

2.2 Minimizing energy expenditures in a DC federation

A first optimization to reduce energy expenditures is to perform consolidation, placing VMs so as to load servers as much as possible and switching off those servers that become unused. To further reduce energy consumption, consolidation can be performed by taking into account clusters structure, and switching on/off clusters as single units. Those servers in switched on clusters without assigned load remain active and ready to accommodate spikes in demand.

In addition, DC federations can perform elastic operations, migrating VMs among DCs aiming at minimizing operational costs by taking advantage from available green solar energy in some DCs and off-peak cheap brown energy in others DCs while ensuring the desired QoE level. The latency experienced by the users of a service can be used as a measure of QoE level.

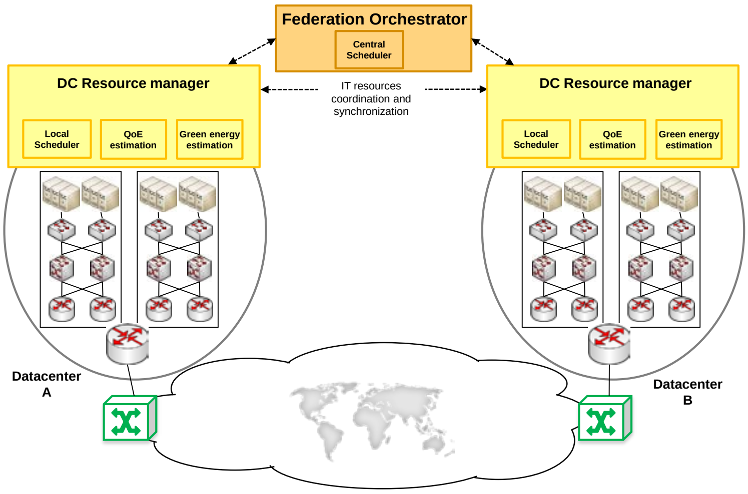

Fig. 1. Architecture for orchestrating datacenter federations.

We assume that a *federation orchestrator* (Fig. 1) computes periodically the global optimal placement for all the VMs in the federated DCs so as to minimize operational costs whilst ensuring QoE. That centralized architecture has proved to reduce operational costs, when dominated by energy and communication costs [8]. The orchestrator computes VM placement and communicates that to each DC resource manager. To solve the scheduling problem some data must be available, such as an estimation of QoE perceived by the users, the amount of green energy available in each DC and the cost of brown energy, among others. QoE can be estimated by a specialized module inside each resource manager. The cost of brown energy comes from the contract that each DC has with the local power supply company, which varies with the time of day. Finally, the amount of green energy that will be likely available in the next period can be predicted using historical data and weather forecast [21]. Then, each local resource manager migrates VMs sending its data to some remote DC.

In line with [8], let us assume that green solar energy covers some proportion β_d of the maximum energy consumption of DC d (Max_Energy_d), i.e., in the scenario where all servers are at the highest load, all switches are active, and the *PUE* value is the highest one. The amount of green energy available in DC d at period t can be estimated as $g_d(t) = \beta_d * \delta_d(t) * Max\_Energy_d$, where $\delta_d(t)$ is the normalized availability of green energy as a function of the time of day in the location of DC d.

Therefore, once the VMs to be placed in each DC in the next period are known, the federation orchestrator can compute precisely the amount of workload in each DC, compute the green energy available and thus, the optimal VM placement.

2.3 Statistical model for green solar energy availability estimation

Note that the accuracy of $\delta_d(t)$ highly impacts on the accuracy of the green solar energy availability estimation $g_d(t)$. However, the value of $\delta_d(t)$ is related not only to the current season, but also to the weather conditions, which makes it hard to be aggregated into a single value. Plots in Fig. 2 illustrate this fact for a DC placed in Berlin (data obtained from [22]-[24]). As observed, all the days in the same week behave similarly, i.e. peak values are at the same time and around a mean value (seasonal effect), although steep changes from one hour to the next and between two consecutive days (weather effect) can be clearly observed. In view of this, we decided characterizing the green energy availability as a function of three variables: the day of the year (index i), the hour of the day (index j), and the weather conditions (modeled as *weather levels*, index k). Therefore, we use $\delta_d(i,j,k)$ hereafter.

Fig. 3a shows observed δ_d values along a year at 2 pm; i.e. data pairs $\{i, \delta_d(i,2,*)\}$, where * represents any value of the variable (weather level in this case). Five weather levels with different markers are plotted; each level shows a similar shape but different scale. In light of Fig. 3a, it can be concluded that weather conditions can be likely predicted as weather levels. Hence, our proposed statistical model to estimate $\delta_d(i,j,k)$ consists of a set of curves, one for each weather level k, as a function of the day i. In addition, a set of coefficients to scale predictions for every hour j are necessary. With this approach, multiple stationary effects (daily, yearly) can be easily managed.

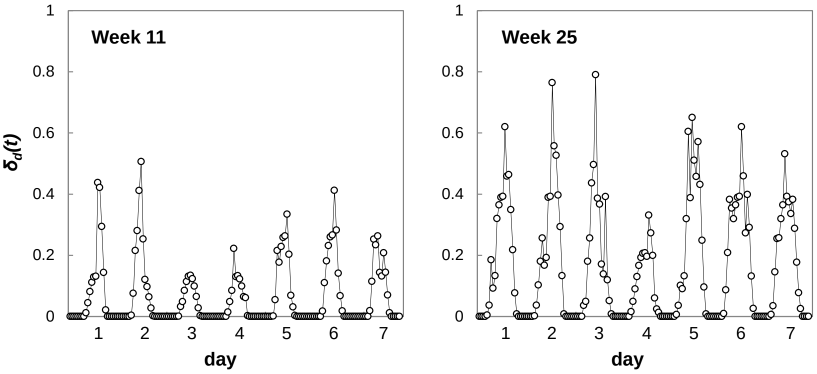

Fig. 2. $\delta_d(t)$ as a function of the day for two different weeks.

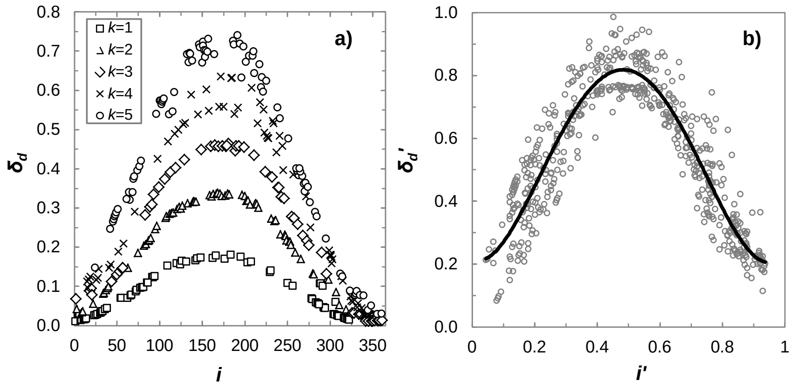

Fig. 3. (a) δ_d as a function of the day (at 2 p.m.). (b) δ_d' for k=3.

Before applying curve fitting to each weather level, data must be properly normalized. To that end, a tuple with three coefficients is needed for every observation belonging to the same hour j (i.e. $\delta_d(*,j,*)$): {$firstDay_j$, $lastDay_j$, $peak_j$}, where the first two coefficients store the range of days where δ_d takes non-zero values and the last coefficient stores the maximum δ_d observed . The tuple for data in Fig. 3a is {1, 365, 0.747}. Note that those tuples are remarkably different for different hours, e.g. the tuple at 7 a.m. for the same location is {88, 294, 0.255}.

Next, data pairs {i, $\delta_d(i,*,*)$} need to be normalized following equations (3) and (4), obtaining normalized data pairs {i', $\delta_d(i',*,*)'$} in the continuous range [0, 1]. For illustrative purposes, let us normalize the observation in Fig. 3 for k=3 and i=100, with δ_d=0.351. From the tuple of coefficients characterizing data in Fig. 3a and equations (3) and (4), i'=(100-1)/(365-1)=0.27 and δ_d'=0.351/0.747=0.47. Normalized data pairs for k=3 are represented in Fig. 3b, together with the curve that better fits the average trend. After a preliminary analysis, we concluded that a degree 4 polynomial is enough accurate for fitting. Therefore, the statistical model shown in equation (5) consists in a polynomial with 5 coefficients that receives as input the normalized day, the hour, and the weather level and returns the normalized green energy availability. Since the polynomial does not perfectly match all the observations, a Gaussian error ε centered in 0 with variance σ^2 is considered as part of the model.

$$i' = \frac{i - firstDay_j}{lastDay_j - firstDay_j} \tag{3}$$

$$\delta_d(i', j, *)' = \frac{\delta_d(i', j, *)}{peak_j} \tag{4}$$

$$\delta_d(i', j, k)' = \sum_{n=0..4} b_{nk} \cdot [i']^n + \varepsilon(0, \sigma^2) \tag{5}$$

Using the proposed statistical model, $\delta_d(i,j,k)$ for a given day i, hour j, and weather level k can be estimated as follows: *a*) the day is first normalized using eq. (3), thus obtaining i'; *b*) $\delta_d(i',j,k)'$is estimated using eq. (5); and *c*) $\delta_d(i,j,k)$ is obtained from $\delta_d(i',j,k)'$ using eq. (4) properly reverted. As a result, the probability density function of the green energy availability follows a Gaussian distribution with mean $\delta_d(i,j,k)$ and variance σ^2.

Note that the expected weather level is an input of the model that might be estimated by other means. When estimations are done between consecutive hours, it is clear that a time series model for predicting next weather level from past observations is a valuable option. However, in this work we take advantage of a basic but time-dependent model consisting in a matrix of transition probabilities between levels in consecutive hours, estimated from the available data. The matrix stores the probability that weather changes from level k to level k' in the next hour j+1; an example of transition probabilities matrix is shown in Section 4.

In the next section, the proposed methodology is used to generate instances for solving a new approach of the centralized ELFADO problem defined in [8] that we call stochastic ELFADO (STC-ELFADO).

2.4 Stochastic ELFADO formulation

For the sake of completeness, the ELFADO problem statement is reproduced in the following:

<u>*Given*</u>:

- a set of federated datacenters D.
- the set of optical connections E that can be established between every pair of DCs,
- a set of VMs $V(d)$ in each datacenter d,
- a set of client locations L, where nl_l is the number of users in location l to be served in the next period,
- PUE_d, brown energy cost c_d, and green energy available in datacenter d for the next period,
- the data volume u_v and the number of cores nc_v of each VM v,
- energy consumption of each server as a function of the load and each switch,
- the QoE q_{ld} perceived in location l when served from a VM placed in datacenter d,
- a threshold th_v for the QoE required at any time for accessing the service in virtual machine v.

<u>*Output*</u>*:* the datacenter where each VM will be placed the next time period;

<u>*Objective*</u>*:* Minimize energy and communications cost for the next time period ensuring the performance objective for each service.

We propose to solve the stochastic approach for the ELFADO problem by means of discrete probability scenarios (see [25]). In view of the statistical model presented above, a scenario for each weather level k can be easily obtained: the green energy available is estimated from the polynomial model and its probability comes from the transition probability from the current to the next k level. Then, the brown energy needed at each scenario is different and the cost of each scenario is weighted in the objective function by the transition probability. In line with [11], we assume the cost of green energy is zero to prioritize its use against brown energy.

In addition, we consider that each DC is connected to the flexgrid inter-DC network through a switch equipped with bandwidth variable transponders of a given capacity (e.g. 100 Gb/s). To limit the total time of the migration process (*maxMigrationTime*), we limit the amount of data (GB) that can be transferred by a given optical connection in *maxMigrationTime*, which depends on the effective connection throughput (i.e., capacity without headers).

The notation needed for the MILP formulation is as follows:

Sets

D	set of federated DC, index d.
E	set of optical connections, index e.
V	set of VMs, index v.
$V(d)$	set of VMs currently placed in DC d.
L	set of client locations, index l.
$K(d)$	set of probability scenarios in DC d

Users and performance

q_{ld}	QoE (in terms of delay in ms) experienced by users accessing from location l to DC d.
nl_l	number of users in location l.
th_v	on average QoE threshold to be guaranteed for those users accessing VM v.

DC architecture and VMs

M	number of clusters per DC.
ns	number of cores per server.
u_v	size of VM v in GB.
nc_v	number of cores needed by VM v.

Connections

u_e	Max volume of data (GB) that can be conveyed through connection e, computed as: throughput_e * *maxMigrationTime*.
c_e	cost per GB transmitted through connection e.

Energy

PUE_d PUE for DC d.

c_d brown energy cost per kWh in DC d.

$w(\cdot)$ energy consumption of element $(\cdot)$, $w(\cdot) = P(\cdot)$*1h

g_{dk} amount of green energy available in DC d under probability scenario k

p_{dk} probability of scenario k in DC d

The decision variables are:

x_{vd} binary, 1 if VM v is placed in DC d; 0 otherwise.

y_{dk} positive real, energy consumption in DC d in probability scenario k

z_e positive integer, GB to be transferred through optical connection e.

γ_d positive integer with the number of servers operating with some load in DC d.

ρ_d positive integer with the number of clusters switched on in DC d.

b_d positive real, total energy consumption in DC d

Finally, the formulation of the stochastic ELFADO based on introducing discrete probability scenarios, is as follows:

$$(\text{STC}-\text{ELFADO})\ \text{minimize} \sum_{d\in D} c_d \cdot \sum_{k\in K(d)} p_{dk} \cdot y_{dk} + \sum_{e\in E} c_e \cdot z_e \tag{6}$$

s.t.

$$\frac{1}{\sum_{l\in L} nl_l} \cdot \sum_{l\in L}\sum_{d\in D} nl_l \cdot q_{ld} \cdot x_{vd} \le th_v \quad \forall v \in V \tag{7}$$

$$\sum_{d\in D} x_{vd} = 1 \quad \forall v \in V \tag{8}$$

$$\gamma_d \ge \frac{1}{ns} \cdot \sum_{v\in V} nc_v \cdot x_{vd} \quad \forall d \in D \tag{9}$$

$$\rho_d \ge \frac{4}{M^2} \cdot \gamma_d \quad \forall d \in D \tag{10}$$

$$b_d = PUE_d \cdot \left(\begin{array}{c} \frac{M^2}{4} \cdot w_{core} + \\ \frac{M}{2} \cdot \left(w_{agg} + w_{edge} \right) \cdot \rho_d + \\ w_{server-max} \cdot \gamma_d + \\ w_{server-idle} \cdot \left(\frac{M^2}{4} \cdot \rho_d - \gamma_d \right) \end{array} \right) \quad \forall d \in D \tag{11}$$

$$y_{dk} \ge b_d - g_{dk} \quad \forall d \in D, k \in K(d) \tag{12}$$

$$z_{e=(d_1,d_2)} = \sum_{v\in V(d_1)} u_v \cdot x_{vd_2} \quad \forall d_1, d_2 \in D, d_1 \ne d_2 \tag{13}$$

$$z_e \le u_e \quad \forall e \in E \tag{14}$$

The objective function (6) minimizes the total cost for every DC in the federation, which consists on the energy costs (weighted by the probability of each scenario), and the communication costs for the VMs that are moved between DCs. Constraint (7) guarantees that each VM is assigned to a DC if the on-average QoE perceived by the users does not exceed the given threshold. Constraint (8) ensures that each VM is assigned to exactly one DC. Constraint (9) computes, for each DC, the amount of servers where some VM is to be placed, whereas constraint (10) stores the number of clusters that will be switched on. Constraint (11) computes the total energy consumption in each DC (see power model in Section 2.1). Constraint (12) stores the brown energy consumption for each scenario in each DC computed as the difference between the effective energy consumption and the amount of green energy available in the next period in each DC. Constraint (13) computes the amount of data to be transferred from each DC to some other remote DC and constraint (14) assures that the capacity of each optical connection is not exceeded.

Scheduling algorithms such as STC-ELFADO run periodically (e.g. each hour) and compute the proper VM placement aiming to reduce energy expenditures, while ensuring the maximum migration time. It is clear that to minimize expenditures, a VM could be migrated every hour, which can be avoided adding new constraints to the MILP formulation. Notwithstanding, the statistical model for green solar energy availability estimation, presented in Section 2.3, introduces short-term solar energy predictability that indirectly avoids abrupt changes in the prediction, preventing VMs to be continuously migrated.

3 ORCHESTRATING CONNECTIVITY SERVICES FOR FEDERATED DATACENTERS

Each DC local resource manager needs to interface not only other DCs to coordinate VM migration, but also the ABNO controlling the interconnection network to request optical DC2DC connections' set-up and tear down. In this paper, we propose that local resource managers request connectivity services through an ASO, which coordinates services and interfaces the SDN controller inside the ABNO to request operations on the network. The general architecture is presented in Fig. 4.

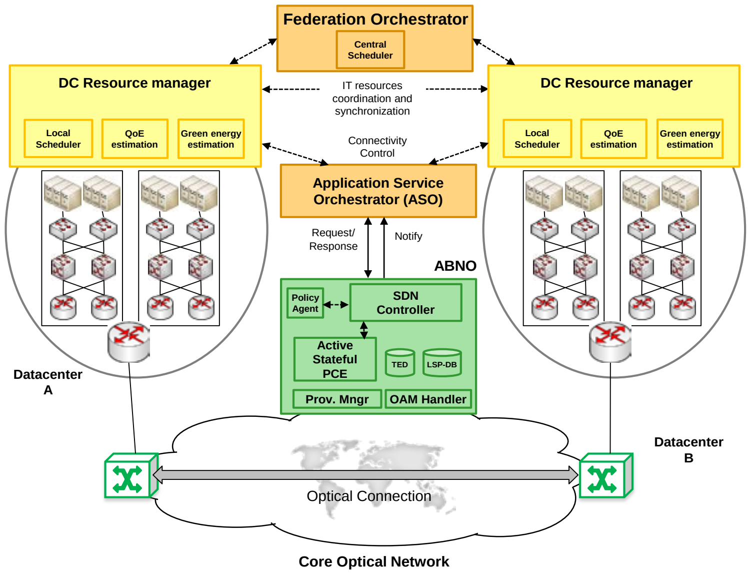

Fig. 4. ASO implementing a northbound interface for the application layer and request/response and notifications towards the ABNO.

Scheduling algorithms such as STC-ELFADO run periodically (e.g. each hour) and thus, VM migrations are required to be performed within each period; more specifically, the migration process should be completed within *maxMigrationTime*. The live-migration process for each individual VM migration can be done in short times and thus downtimes are virtually zero when both servers are in the same DC. However, when the migration process involves two different DCs, the downtime directly depends on the throughput of the path connecting the servers [26] as a result not only of the delay, but also because some of the already migrated memory pages are modified in the current server (known as dirty pages). Therefore, from the DC federation viewpoint, the offered connectivity service is better when the VMs can transferred fast to their proper locations. Nonetheless, the network operator wants to reuse its resources as much as possible, so it is required to assess a connectivity model which can fit with users and operators requirements. In this section we compare two connectivity models: *i*) *polling-based*, and *ii*) *notification-based*.

In the polling-based model, each local resource manager controls connectivity to remote DCs so as to perform VM migration in the shortest total time. The source resource manager requests label switched paths (LSP) set-up, tear down, as well as elastic operations to the SDN controller in the ABNO architecture (Fig. 5a). After checking local policies, the SDN controller forwards the requests to the active stateful PCE, which performs LSP operations on the controlled network. It is worth noting that, although applications have full control over the connectivity process, physical network resources are shared with a number of clients and LSP set-up and elastic spectrum increments could be blocked as a result of lack of resources in the network. Hence, applications need to implement some sort of periodical retries to increase the allocated bandwidth until reaching the required level. These retries, could impact negatively on the performance of the inter-DC control plane and do not ensure achieving higher bandwidth.

In the notification-based model, applications request transfers instead of connectivity; an ASO is deployed in between the network control plane and DC resource managers, as a new stratum to provide an abstraction layer to the underlying network. The ASO implements a northbound interface to request transfer operations. Those applications' operations are transformed into network connection requests. The northbound interface uses application-oriented semantic, liberating application developers from understanding and dealing with network specifics and complexity.

Thus, resource managers request transfers using its native semantic: amount of data to be transferred, DC destination, completion time, etc. The ASO is in charge of managing inter-DC connectivity; if not enough resources are available at requesting time, notifications (similar to interruptions in computers) are sent from the PCE to the ASO each time specific resources are released. To that end, a subscription service has been implemented in the ABNO architecture: the ASO subscribes to a notification service in the SDN controller; when some resource in the set of specified links is released the SDN controller sends a notification message towards the subscribed ASO. Upon receiving a notification, the ASO takes decisions on whether to increase the bitrate associated to a transfer. Therefore, we have effectively removed the polling mechanism and migrated to a notification-based transfer mode.

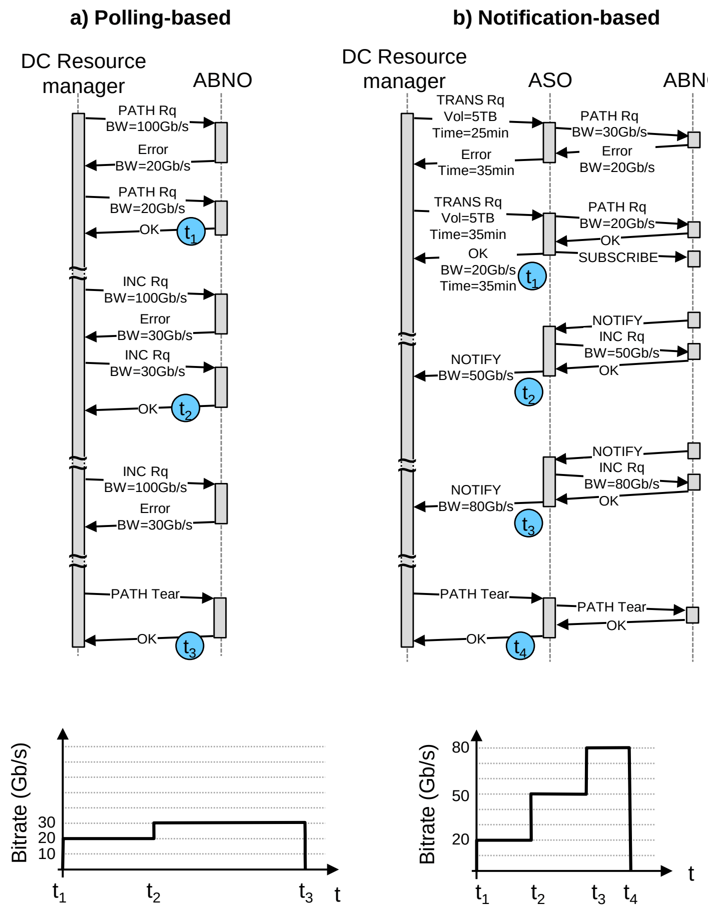

Fig. 5. Connection models and bitrate under each model.

In the proposed notification-based model (Fig. 5b), the source resource manager sends a transfer request to the ASO specifying the destination DC, the amount of data to be transferred, and the required completion time. Upon its reception, the ASO requests the SDN controller to find the greatest spectrum width available, taking into account local policies and current service level agreements (SLA) and sends a response back to the resource manager with the best completion time. The source resource manager organizes data transfer and sends a new request with the suggested completion time. A new connection is set up and its capacity is sent in the response message; besides, the ASO requests the SDN controller to keep it informed when resources become available in the route of that LSP. Algorithms deployed in the SDN controller monitor spectrum availability in those physical links, so when resource availability allows increasing the allocated bitrate of some LSP, the ASO performs elastic spectrum operations to ensure committed transfer completion times. Each time the ASO modifies the bitrate of a connection, a notification is sent to the source resource manager with the new throughput. The resource manager then optimizes VM migration as a function of the actual throughput while delegating ensuring completion transfer time to the ASO. Fig. 5 shows the bitrate of the connections against time under both models. It is worth noting that the notification-based model takes advantage of the use of subscriptions and notify messages, being able to reduce time-to-transfer remarkably.

Table 1 Algorithm for transfer request

INPUT *source, destination, dataVol, rqTime, slotWidth*
OUTPUT *Response*

1: *minBitrate* ← translateAppRequest (*dataVol, rqTime, slotWidth*)
2: *netResp* ← requestConnection (*source, destination*, *minBitrate*)
3: **if** *netResp==KO* **then**
4:     *maxBitrate* ← getMaxBitrate (*source* , *destination*)
5:     *minTime* ← translateNetResponse (*maxBitrate, dataVol*)
6:     **return** {*KO, minTime*}
7: requestSubscription (*netResp.connId.route*)
8: *time* ← translateNetResponse (*netResp.connId.bitrate, dataVol*)
9: **return** {*OK, netResp.connId, netResp.connId.bitrate, time*}

Table 1 presents the algorithm that we implemented in the ASO for transfer requests. It translates requested data volume and completion time, into a required bitrate, taking into account frequency slot width in the flexgrid network (line 1). Next an optical connection request is sent towards the SDN controller, specifying source and destination of the connection and the bitrate (line 2). In case of lack of network resources (lines 3-6), the maximum available bitrate between source and destination DCs is requested to the SDN controller, its result translated into the minimum completion time, which is used to inform the requesting DC resource manager. If the connection could be established, the ASO requests a subscription to the links in the route of the connection, so as to be aware of available resources as soon as they are released in the network (line 7). Finally, the actual completion time is recomputed taking into consideration the connection's bitrate and both, bitrate and time, are communicated back to the requesting DC resource manager.

4 ILLUSTRATIVE RESULTS

In this section, we firstly validate the stochastic approach for the ELFADO problem comparing its performance against that of the deterministic version presented in [8]. Next, we use STC-ELFADO on a worldwide scenario to compare the performance from using the polling-based and the notification-based connection models.

For evaluation purposes, we developed resource managers in an OpenNebula-based cloud middleware emulator. The federation orchestrator solving the centralized STC-ELFADO model was implemented as a stand-alone module in Java. Federated DCs are connected to an ad-hoc event-driven simulator developed in OMNET++. The simulator implements the ASO and the flexgrid network with ABNO and SDN controller on the top, as described in Fig. 4. Finally, the algorithm described in [27] for elastic spectrum allocation was implemented.

Brown energy cost for each DC was estimated from their respective local electric company rates (e.g. Europe's Energy Portal and U.S. Department of Labor) and green energy coverage was obtained from U.S. Department of Labor. Servers in DCs are assumed to be HP ProLiant DL580 G3, equipped with four processors, 2 cores per processor, with $P_{server\text{-}idle}$ = 520W and $P_{server\text{-}max}$ = 833W. We consider a different type of switch, and thus a different power consumption value, for each layer of the intra-DC architecture. We selected the Huawei *CloudEngine* switches series. Table 2 details switching capacity and power consumption of the switches.

Table 2 Characteristics of Huawei CloudEngine switches.

Layer	Model	Sw. capacity	P(W)
Core	12812	48 Tb/s	16200
Aggregation	6800	1.28 Tb/s	270
Edge	5800	336 Gb/s	150

4.1 Stochastic ELFADO validation

To compare the performance of the deterministic and the stochastic ELFADO approaches, we consider a European topology consisting of 10 DCs, each with a number of clusters M=10. For each location, a statistical model of the form presented in section 2.3 is estimated from the data available in [22]-[24]. Table 3 and Table 4 show the parameters used for data normalization and the coefficients of the polynomials respectively for a datacenter placed in Berlin. Note that the range between *firstDay* and *lastDay* as well as that for the *peak* increase for mid-day hours. In addition to this, Table 5 stores the weather transition probabilities between consecutive hours. Highest probabilities are observed in the diagonal and cells close to it, being this representative of a natural daily weather evolution, where changes during short time periods are usually small. The validation process of this model allows concluding that most of the 95% of the predictions provide residuals within a common accepted standard error of ± 2 ($\sigma^2 \approx 0.002$). The model provides similar performance for all the considered locations.

Table 3 Normalization parameters

	j							
	5	**6**	**7**	**8**	**9**	**10**	**11**	**12**
firstDay$_j$	139	109	88	56	10	1	1	1
lastDay$_j$	216	256	294	328	365	365	365	365
Peak$_j$	0.0106	0.0987	0.2548	0.421	0.5764	0.7094	0.7997	0.8516
	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**
firstDay$_j$	1	1	1	2	36	70	103	136
lastDay$_j$	365	365	365	365	317	287	261	235
peak$_j$	0.8556	0.8138	0.7283	0.6047	0.4564	0.2946	0.1362	0.0256

Table 4 b_{nk} polynomial coefficients

	n=4	***n=3***	***n=2***	***n=1***	***n=0***
k=1	3.737	-7.443	3.759	-0.054	0.016
k=2	7.175	-14.136	7.166	-0.208	0.075
k=3	9.227	-17.895	8.661	0.001	0.124
k=4	11.283	-21.782	10.319	0.247	0.133
k=5	11.847	-21.920	8.497	1.823	-0.049

Table 5 Weather probability transitions (***j→j+1***)

	k=1	***k=2***	***k=3***	***k=4***	***k=5***
k=1	0.7298	0.1992	0.0498	0.0151	0.0060
k=2	0.2427	0.4976	0.1445	0.0689	0.0463
k=3	0.1069	0.3152	0.3261	0.1793	0.0725
k=4	0.0712	0.1346	0.2000	0.3154	0.2788
k=5	0.0054	0.0261	0.0534	0.1786	0.7364

For evaluation, a total load of 5,000 VMs is distributed among DCs. Under the stochastic approach, one scenario per weather level k and DC were considered (50 scenarios in total), with expected g_{dk} and p_{dk} obtained from the proposed statistical model. Under the deterministic approach, only one scenario per DC with the expected average green energy availability was assumed.

Aiming at evaluating the quality of the obtained solutions, we computed the *Expected Value of Perfect Information* (*EVPI*), which assumes that the real amount of green energy is perfectly known in advance, being this the benchmark for computing energy costs and savings.

We solved the ELFADO problem using CPLEX [28] at every hour of 120 consecutive days of spring and summer time (2880 problem instances in total). To estimate likely costs of each ELFADO solution, a simulation was run to obtain green energy availability from the probability distribution obtained with the $\delta_d(t)$ statistical model, thus emulating a real-life behavior; costs were eventually computed from that simulated green energy values.

Fig. 6a shows the energy cost increment with respect to the cost of EVPI solutions as a function of daytime hours. As clearly observed, the main differences between strategies appear in central hours, when $\delta_d(t)$ takes the highest values and when fluctuations induced by weather variations are more evident. The stochastic approach obtains solutions much closer to EVPI compared to the deterministic scheme. Specifically, a maximum difference circa 5% with respect to EVPI is observed. The benefits of this stochastic approach can be also observed in Fig. 6b, which plots cumulative cost increment; a promising 30% of the total cost increment with respect to that provided by the deterministic approach is achieved by only using few discrete probability scenarios. In view of this, it is clear that the stochastic approach for solving ELFADO provides high-quality solutions with a computational complexity similar to that of the deterministic approach.

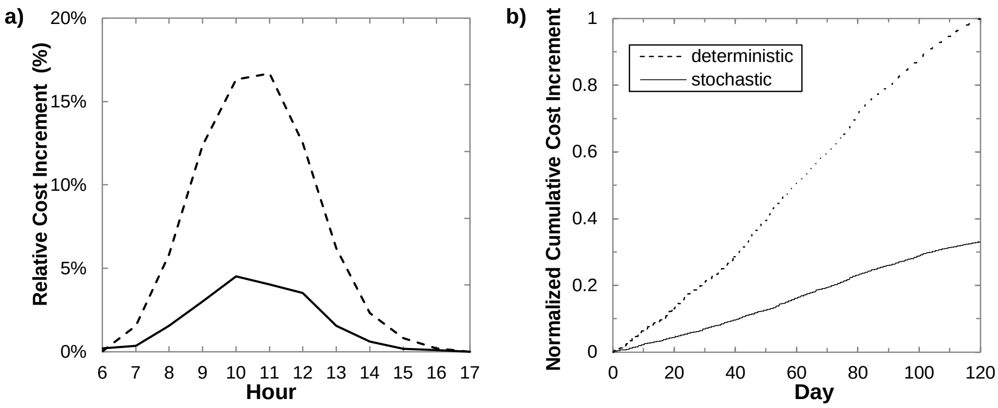

Fig. 6. Energy Cost Increment w.r.t. EVPI: a) relative per hour, b) cumulated per day

4.2 Polling-based and notification-based performance evaluation

After the stochastic approach for ELFADO problem has been validated, we move to a global 11-node topology, where locations are used as source for DC2U traffic collecting user traffic towards the set of DCs. The set of DCs consists of five DCs strategically located in Taiwan, India, Spain, and Illinois and California in the USA (Fig. 7). A global telecom operator provides optical connectivity among DCs, which is based upon the flexgrid technology. The number of users in each location was computed considering Wikipedia's audience by regions that was scaled and distributed among the different locations in each region. Latency was computed according to Verizon's data (See http://www.verizonbusiness.com).

In line with [10], DCs are dimensioned assuming a fat-tree topology with a maximum of M=48 clusters with two levels of switches and $M^2/4$=576 servers each. The number of VMs was set to 35,000, with individual image size of 5 GB; we assume that each VM runs in one single core. Initially, all VMs are distributed among all DCs equally.

An integer number of clusters is always switched on, to support the load assigned to the DC; those servers without assigned load remain active and ready to accommodate spikes in demand. Green cover was set to ensure, at the highest green energy generation time, a proportion of energy β_d when all VMs run in DC d.

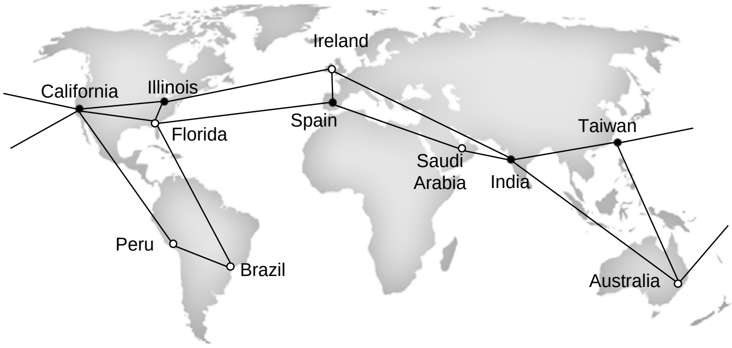

Fig. 7. Worldwide scenario considered in our experiments.

We consider that each DC is connected to the flexgrid inter-DC network through a switch equipped with 100Gb/s bandwidth variable transponders. Therefore, the actual capacity of optical connections is limited to that value. To compute the real throughput, we consider headers for the different protocols, i.e. TCP, IP, and GbE. The capacity requested for optical connections was computed to guarantee that VM migration is performed in less than 40 minutes. Note that transponders capacity together with maximum scheduled transfer time limit the amount of VMs that can be moved between two DCs to about 4,620.

Finally, a dynamic network environment was simulated for the scenario under study, where background incoming connection requests arrive following a Poisson process and are sequentially served without prior knowledge of future incoming connection requests. Background traffic competes with the one generated by the federated DCs for network resources. The bitrate demanded by each background connection request was set to 100 Gb/s. The holding time of connections is exponentially distributed with the mean value equal to 2 hours. Source/destination pairs are randomly chosen with equal probability among all nodes. Different values of offered network load were considered by changing the arrival rate while keeping the mean holding time constant.

Fig. 8 plots daily energy and communication costs as a function of the normalized background traffic intensity. We observe a clear increasing trend when the background traffic increases, as a consequence of connections' initial capacity decreases from 55 Gb/s to only 12 Gb/s on average. To try to increase that limited initial connections' capacity, elastic capacity increments need to be requested. The results obtained when each connectivity model is applied are however different. Both models behave the same when the background traffic intensity is low or high, which is as a consequence of the percentage of VMs that could not be migrated in the scheduled period (see Fig. 9a). When the intensity is low, there are enough resources in the network so even in the case that elastic connection operations are requested, both models are able to perform scheduled VM migration in the required period. When the background intensity is high connections requests are rejected or are established with a reduced capacity that is unlikely modified. As a result, a high percentage of scheduled VM migrations could not be performed.

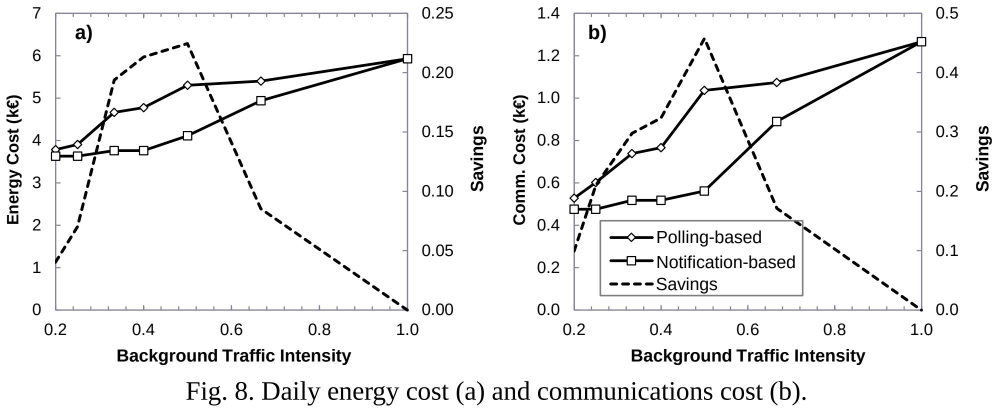

Fig. 8. Daily energy cost (a) and communications cost (b).

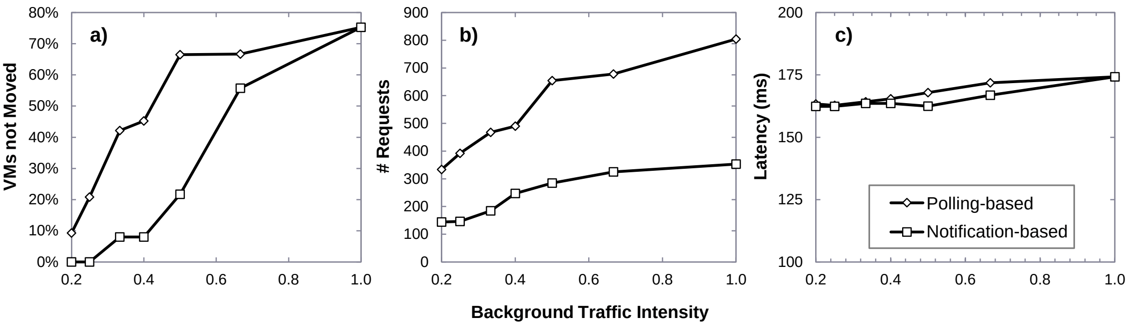

Fig. 9. Percentage of VMs not moved as first scheduled (a), number of connection requests (b), and latency experienced by users (c).

However, when the background load increases without exceeding 5% of total blocking probability, the behavior of the analyzed connectivity models is different; the proposed notification-based model provides a constant energy and communication costs until the normalized background load is greater than 0.4, in contrast to the remarkable cost increment provided using the polling-based model. In fact, costs savings as high as 20% and 45% in energy and communications, respectively are obtained when the notification-based model is applied with respect to those of the polling-based. When the normalized background load increases from 0.4, the lack of resources starts affecting also the notification-based model and, although costs savings reach their maximum for a load of 0.5, energy and communication costs start increasing and relative savings decreasing.

It is also interesting to see the total number of requests generated when each connectivity model is used. Fig. 9b plots the amount of requests for set-up, elastic capacity increment or decrement, and tear down that arrive to the SDN controller. When the polling-based model is used, the number of requests is really high compared to that number under the notification-based model. However, since the requests are generated by the DC resource managers without any knowledge of the state of the resources, the majority of those requests are blocked as a result of lack of resources. Such high utilization of the network resources is the target for the network operator. In contrast, in the notification-based model, elastic capacity increment or decrement requests are generated by the ASO, which knows that some resources in the route of established connections have been released and elastic capacity operation could be successfully applied. In this case, the amount of requests is much lower but many of them are successfully completed (although some few can be also blocked). Regarding latency, both models are able to provide similar performance, as shown in Fig. 9c. This fact, however, is as a result of the scheduler algorithm that focuses at guaranteeing the committed QoE.

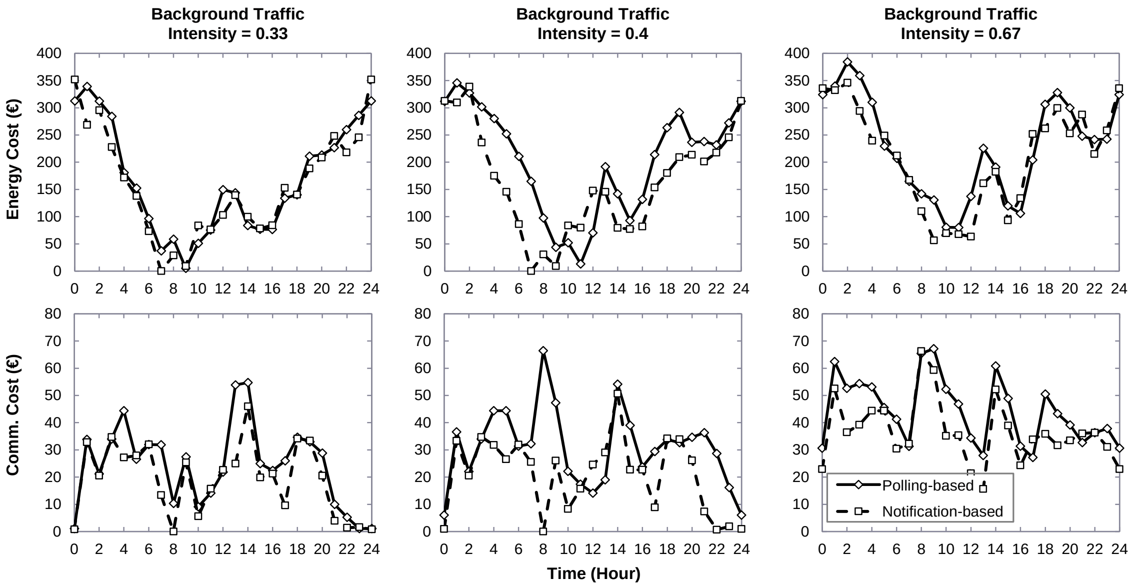

Fig. 10. Hourly costs for several background traffic intensities

Finally, Fig. 10 illustrates hourly variation in the energy and communication costs when the polling-based and the notification-based models are applied, for three different background traffic loads. The behavior of both models is basically the same and slight hourly energy cost savings can be appreciated, although they are clearly evident for the intermediate load. In contrast, there are some periods with a totally different behavior between polling-based and the notification-based models, especially in the intermediate load. That is as a consequence of that VMs can be placed in those locations so as to minimize cost in the notification-based model so no new migrations are required, whereas massive migrations need to be done polling-based, which further increases communications needs.

5 CONCLUDING REMARKS

A statistical model has been proposed for improving the accuracy of green energy availability estimation in a given location. Taking advantage of the proposed statistical model and extending the centralized ELFADO problem, the STC-ELFADO problem has been presented; the statistical model has been used in the STC-ELFADO MILP formulation to compute optimal VM placement in a datacenter federation. Both, the statistical and the deterministic approaches have been compared and results showed that the estimation accuracy in the STC-ELFADO results in energy costs reduction.

Then, ASO implementing a northbound interface with application-oriented semantic has been proposed as a new abstraction layer between DC resource managers and the SDN controller in the control plane of flexgrid-based interconnection networks. Each resource manager can request transfer operations specifying the destination DC, the amount of data to be transferred and the desired completion time.

The above connectivity model, named notification-based, has been compared against the model, named polling-based, where the local resource managers are in charge of requesting connections directly to the SDN controller. The polling-based model needs periodical retries requesting to increase connection's bitrate, which do not translate into immediate bitrate increments and could have a negative impact on the performance of the network control plane.

Energy and communication costs and QoE in a DC federation were analyzed. Some green solar energy is available in each of the locations as a function of the time, whilst the cost of brown energy shows differentiated on/off peak costs. A federation orchestrator computes periodically the global optimal placement for all the VMs in the federation so as to minimize operational costs whilst ensuring QoE.

From the results, we observed that when the network operates under low and medium traffic load costs savings as high as 20% and 45% in energy and communications, respectively can be obtained when the notification-based model is applied with respect to those of the polling-based. Besides, both connectivity models allow scheduling algorithms to provide the committed QoE.

Finally, the proposed notification-based model opens the opportunity to network operators to implement policies so as to dynamically manage connections' bitrate of a set of customers and fulfill simultaneously their SLAs.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 317999 IDEALIST project and from the Spanish MINECO SYNERGY project (TEC2014-59995-R).

REFERENCES

[1] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic Resource Management Using Virtual Machine Migrations," IEEE Commun. Mag, vol. 50, no. 9, pp. 34-40, 2012.

[2] X. Zhao, V. Vusirikala, B. Koley, V. Kamalov, and T. Hofmeister, "The Prospect of Inter-Data-Center Optical Networks," IEEE Commun. Mag., vol. 51, no. 9, pp. 32-38, 2013.

[3] I. Goiri, J. Guitart, and J. Torres, "Characterizing Cloud Federation for Enhancing Providers' Profit," in Proc. IEEE Cloud, 2010.

[4] Cisco, Global Cloud Index, 2012.

[5] D. King, A. Farrel, "A PCE-based Architecture for Application-based Network Operations," IETF draft, Jan. 2015.

[6] L. Velasco, A. Asensio, J.Ll. Berral, V. López, D. Carrera, A. Castro, and J.P. Fernández-Palacios, "Cross-Stratum Orchestration and Flexgrid Optical Networks for Datacenter Federations," IEEE Network, vol. 27, no. 6, pp. 23-30, 2013.

[7] L. Velasco, A. Castro, M. Ruiz, and G. Junyent, "Solving Routing and Spectrum Allocation Related Optimization Problems: from Off-Line to In-Operation Flexgrid Network Planning," (Invited Tutorial) IEEE/OSA Journal of Lightwave Technology (JLT), vol. 32, no. 16, pp. 2780-2795, 2014.

[8] L. Velasco, A. Asensio, J. Ll. Berral, E. Bonetto, F. Musumeci, V. López, "Elastic Operations in Federated Datacenters for Performance and Cost Optimization," Elsevier Computer Communications, vol. 50, pp. 142-151, 2014.

[9] Y. Zhang, and N. Ansari, "On Architecture Design, Congestion Notification, TCP Incast and Power Consumption in Data Centers," IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 39-64, 2013.

[10] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in Proc. ACM SIGCOMM, 2008.

[11] I. Goiri, K. Le, T. Nguyen, J. Guitart, J. Torres, and R. Bianchini, "GreenHadoop: Leveraging Green Energy in Data-Processing Frameworks," in Proc. EuroSys, 2012.

[12] L. Liu, H. Wang, X. Liu, X. Jin, W. He, Q. Wang, and Y. Chen, "GreenCloud: A New Architecture for Green Data Center," in Proc. ICAC-INDST, 2009.

[13] Z. Liu, M. Lin, A. Wierman, S. Low, L. Andrew, "Geographical Load Balancing with Renewables," in Proc. ACM Greenmetrics, 2011.

[14] J. Pierson, "Green Task Allocation: Taking into Account the Ecological Impact of Task Allocation in Clusters and Clouds," River Publishers Journal of Green Engineering, vol. 1, no. 2, pp. 129–144, 2011.

[15] J Buysse, K. Georgakilas, A. Tzanakaki, M. Leenheer, B. Dhoedt, and C. Develder, “Energy-Efficient Resource-Provisioning Algorithms for Optical Clouds,” IEEE/OSA Journal of Optical Communications and Networking, vol. 5, no. 3, pp. 226-239, 2013.
[16] M. Gattulli, M. Tornatore, R. Fiandra, and A. Pattavina, “Low-Emissions Routing for Cloud Computing in IP-over-WDM Networks with Data Centers,” IEEE Journal on Selected Areas in Communications, vol. 32, no. 1, pp. 28-38, 2014.
[17] J. Zhang, et al. “First demonstration of enhanced Software Defined Networking (eSDN) over elastic Grid (eGrid) Optical Networks for Datacenter Service Migration,” in Proc. OSA OFC PDP5B.1, 2013.
[18] L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. López, “In-Operation Network Planning,” IEEE Communications Magazine, vol. 52, no. 1, pp. 52-60, 2014.
[19] L. Velasco, A. Asensio, J.Ll. Berral, A. Castro, V. López, "Towards a Carrier SDN: An example for Elastic Inter-Datacenter Connectivity," (Invited Paper) OSA Optics Express, vol. 22, pp. 55-61, 2014.
[20] The Green Grid: www.thegreengrid.org
[21] N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy, “Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems,” in Proc. SECON, 2010.
[22] US Department of Energy. US Energy Information Administration. 2009. Web site http://www.eia.doe.gov/.
[23] US Department of Energy, http://apps1.eere.energy.gov/buildings/energyplus/weatherdata_about.cfm
[24] D. King, W. Boyson, J. Kratochvil, “Photovoltaic Array Performance Model,” Sandia National Laboratories Report, SAND2004-3535, 2004.
[25] A. Shapiro, D. Dentcheva, A. Ruszczynski, “Lectures on Stochastic Programming: Modeling and Theory,” MPS-SIAM, 2009.
[26] E. Harney, S. Goasguen, J. Martin, M. Murphy, and M. Westall, “The Efficacy of Live Virtual Machine Migrations over the Internet,” in Proc. VTDC, 2007.
[27] A. Asensio, M. Klinkowski, M. Ruiz, V. López, A. Castro, L. Velasco, J. Comellas, “Impact of Aggregation Level on the Performance of Dynamic Lightpath Adaptation under Time-Varying Traffic,” in Proc. IEEE ONDM, 2013.
[28] CPLEX, <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>.