Cost-Aware VM Placement across Distributed DCs using Bayesian Networks

Dmytro Grygorenko*, Soodeh Farokhi*, and Ivona Brandic

Faculty of Informatics, Vienna University of Technology, Austria.
{firstname.lastname}@tuwien.ac.at

Abstract. In recent years, cloud computing providers have been working to provide highly available and scalable cloud services to keep themselves alive in the competitive market of various cloud services. The difficulty is that to provide such high quality services, they need to enlarge data centers (DCs), and consequently, to increase operating costs. Hence, leveraging cost-aware solutions to manage resources is necessary for cloud providers to decrease the total energy consumption, while keeping their customers satisfied with high quality services. In this paper, we consider the cost-aware virtual machine (VM) placement across geographically distributed DCs as a multi-criteria decision making problem and propose a novel approach to solve it by utilizing Bayesian Networks and two algorithms for VM allocation and consolidation. The novelty of our work lays in building the Bayesian Network according to the extracted expert knowledge and the probabilistic dependencies among parameters to make decisions regarding cost-aware VM placement across distributed DCs, which can face power outages. Moreover, to evaluate the proposed approach we design a novel simulation framework that provides the required features for simulating distributed DCs. The performance evaluation results reveal that using the proposed approach can reduce operating costs by up to 45% in comparison with First-Fit-Decreasing heuristic method as a baseline algorithm.

Key words: cloud computing, Bayesian networks, MCDA, simulation.

1 Introduction

The emergence of big data centers (DCs) causes power consumption issues for cloud providers while they usually use energy plans which are not optimal [1]. In other words, how to achieve cost-optimized solutions to run geographically distributed (geo-distributed) cloud DCs is a major challenge in the era of rising electricity costs and environmental protection on the one hand, and high expectation of cloud customers in terms of quality of service (QoS) on the other hand [2]. From a cloud provider point of view, the designated challenge introduces the necessity of a multi-criteria decision making solution involving several external factors such as power-outages in DCs, weather conditions, and electricity prices, as well as internal factors, such as resource demands and the usage

* The first two authors have contributed equally to this paper.

of different cooling modes. All of these factors can influence the VM placement decision under some levels of uncertainty.

Although various techniques have been devised for efficient cloud resource management, an effective solution for governing cloud resources in geo-distributed DCs is still an open issue. The current work suffers from short comings such as: ignoring the expert knowledge and thereby loosing important information for building efficient system models; partially addressing cloud management problems, i.e., virtual machines (VM) placement [3, 4], temperature-aware energy usage [5], VM migration [6]. In other words, there is a lack of research that tries to model a combination of these problems while taking into consideration their interconnections and dependencies.

In this paper, we propose a new approach to reduce the cloud operating costs taking the cloud provider point of view by proposing a VM placement approach that is applied across distributed DCs. The proposed approach consists of the VM allocation and consolidation algorithms. Each of these algorithms uses a similar Best-Fit-Decreasing (BFD) heuristic that utilizes certain utility function for assessments of the most optimal decision. The proposed approach includes three steps: (i) constructing a Bayesian Network (BN) [7] to represent expert domain knowledge on cloud infrastructure management; (ii) applying Goal Question Metric (GQM) method [8] to define the underlying measures for the chosen criteria based on the BN's output; (iii) applying a method, called multi-criteria decision aid (MCDA)[9], to create the utility function as the final decision making indicator. For the evaluation, we compare our approach with two VM placement baseline algorithms, namely First-Fit-Decreasing heuristic (*FFD*) which supports both allocation and migration, as well as a First-Fit VM allocation approach with no migration strategy, named *NoM*. Evaluation is performed using a proposed cloud simulation framework, named *CloudNet*.

Contributions of the paper are twofold. First, we propose an approach to reduce the cloud operating cost by applying VM placement across geo-distributed DCs. It leverages the cloud expert knowledge and models them in a BN. The outputs of BN reasoning are further utilized in a utility function built based on GQM and MCDA methods and used in the proposed VM allocation and consolidation algorithms. Second, due to the lack of necessary features for evaluating of geo-distributed DCs, we propose and design *CloudNet* as a novel cloud simulation framework.

The remainder of this paper is organized as follows. Section 2 briefly presents the related work, while Section 3 brings up the challenges of VM placement across geo-distributed DCs. Section 4 describes formal definition of the VM placement problem and proposes a cost-aware solution for it. *CloudNet* is introduced in Section 5. While Section 6 includes discussion on the evaluation metrics and results, Section 7 concludes the paper.

2 Related Work

In this section we first give an overview of existing approaches on cloud management and then focus on existing cloud simulation frameworks.

Cloud Management Approaches. Beloglazov et al. [10] propose a green cloud solution that allows not only to minimize operating cost but also to reduce environmental impact. Li et al. [2] present a consolidation and forecast-based resource provisioning algorithm that utilizes BNs. N. Calcavecchia et al. [11] consider dynamic nature of the incoming stream of VM allocation requests and propose a technique called Backward Speculative Placement (BSP) that projects the past demand behaviour of a VM on a candidate target host. Y. Song et al. [12] state that the key improvement of resource utilization and service throughput depends on using an optimized dynamic resource allocation method. They propose a two-tier resource allocation mechanism consisting of local and global resource allocation with feedback to provide capacities of concurrent applications. A recent research by D. Lučanin et al. [13] introduces the usage of the location- and time-dependent factors to leverage them in distributed DCs to enable flexible energy-efficient cloud management via SLA models. J. Altmann et al. [14] propose a cost model along with a model-based service placement optimization algorithm. Their approach takes into consideration the total cost of all possible service placement options in federated hybrid cloud environments and identifies the optimum placement decision.

Despite the mentioned work, there is not enough research attention on identification of causal relationships hidden in expert knowledge which can enable a more cost-aware VM placement across geo-distributed DCs. This motivated us to work on modeling such relationships using BNs and applied methods such as GQM and MCDA on the designed model.

Cloud Simulation Frameworks. *CloudSim* [15] provides a tool kit for modeling and behaviour simulation of various cloud components such as DCs, physical machines (PMs), and VMs. It includes typical cloud features, i.e., VM allocation, cloud federations, and dynamic workloads. Its main usage is the evaluation of cloud resource provisioning strategies in a controlled simulated environment. *D-Cloud* [16] is a dedicated test environment build upon Eucalyptus. It allows to simulate different regular faults in a cloud environment, and to inject them into host operating systems. *PreFail* [17] is a framework for systematic and efficient failure exploration, and validation of correctness of cloud recovery protocols. However, none of the existing tools provides features for simulating geo-distributed cloud DCs. Hence, we design a cloud simulation framework named *CloudNet* (see Section 5) to provide such features.

3 Challenges of VM Placement across Distributed DCs

The main goal of a cloud provider is to minimize the operating costs of running infrastructure while meeting Service Level Agreements (SLAs) with customers. Cloud providers tend to distribute their DCs all over the world in order to cover specific customer requirements and improve the performance of their services. However, for supporting the VM placement across distributed DCs, cloud providers need to handle several challenges in order to achieve a cost-aware solution: (i) each region has its own electricity market that directly effects energy costs. Global electricity price comparison [18] shows quite big price differences that can dynamically change in various countries. Moreover, due to the different weather in various regions, temperature-aware management of distributed DCs can greatly reduce energy cost, specially the cooling cost. More precisely, DCs in cold regions have smaller partial power usage effectiveness (pPUE) rate [5] or broadly speaking consume less energy to cool their infrastructures (e.g., see Figures 2a, and 2d); (ii) power outages can lead to big issues for a cloud provider. Statistics of electrical outages [19] reports the countries with frequent power outages in spite of a low energy price, hence, it might be impossible to guarantee some QoS metrics such as availability in such regions; (iii) decision making regarding the live VM migration is directly affected by factors such as VM RAM size, bandwidth of the migration link, and Dirty Page Rate (DPR), which effects the migration period; (iv) the trade-off between reducing the energy cost (including power and cooling) of DCs on the one hand, and keep the customers satisfied in terms of QoS on the other hand, is a multi-criteria decision making problem for cloud provides. For instance, switching on and off VMs, or frequent migration of VMs can lead to SLA violations and consequently penalty costs that can invert the effect of such actions on cost efficiency.

In general, VM placement across distributed DCs with highly dynamic environment injects a lot of uncertainty about various internal and external factors that makes it a challenging multi-criteria decision problem. Therefore, it needs effective solutions to reduce the operating costs without QoS degradation.

4 The VM Placement Approach

In this section, we first formalize the VM placement problem and then present the proposed algorithms for cost-aware solution across distributed DCs by utilizing BNs to deal with uncertainty.

4.1 Problem Formulation

In this section, we introduce a model of the cloud aspects which are used in our proposed solution.

VM States. At each point of time t each VM can operate within two possible sets of states, either already allocated to a PM, $allocated(t)$, or has to be allocated, $waiting(t)$. The set of all VMs is called $all(t)$, where $all(t) = waiting(t) \bigcup allocated(t)$. The set $migrated(t)$ defines a set of VMs that are being migrated to other PMs at time t, where $migrated(t) \subseteq allocated(t)$, i.e., all VMs of this set are currently under migration. At each execution step, a VM placement method should find a target PM for: (i) all VMs in the set $waiting(t)$; (ii) the VMs from the set $allocated(t)$ that their current allocation is not optimal enough based on a calculated utility value.

Resources. In our modeling, a DC consists of M distinct PMs. Each PM m is defined with a certain set of resources R. Each resource r has a known limited capacity C_{mr}, where $m \in \{1..M\}$ and $r \in \{1..R\}$. We define the binary variable

$x_{ij}(t)$ that indicates if a VM v_i is allocated to a PM j at time t. Equation 1 states that each VM from the set $allocated(t)$ is allocated exactly to one PM.

$$\sum_{j=1}^{M} x_{ij} = 1, \quad \forall\ v_i \in allocated(t) \tag{1}$$

Each VM v_i has its specifications that define upper bound of each resource $max(vr_{ir}(t))$ required by it at any point of time. During each execution step, a VM requires a certain amount of resources vr_{ir} that is considered during decision making process of the VM placement. Since these resources will not be necessarily provisioned for the VM, we introduce the amount of resources $vp_{ir}(t)$ that are provided for the VM. This value can be less (in case of the VM downtime) or equal to the resources required by the VM $vr_{ir}(t)$. Equation 2 guarantees that the amount of the provisioned resources for all VMs allocated to a PM does not exceed the overall capacity of the PM.

$$\sum_{i\ \in\ allocated(t)} x_{ij}(t) \cdot vp_{ir} \leq C_{jr}, \quad \forall\ j = 1..M,\ r = 1..R \tag{2}$$

Moreover, Equation 3 states how the utilization U_{jr} of a PM j and certain resource r with allocated VMs can be computed:

$$U_{jr} = \sum_{i \in allocated(t)} x_{ij}(t) \cdot vp_{ir}, \quad \forall\ j = 1..M,\ r = 1..R \tag{3}$$

Note that the resource which we more focus on in this work is CPU.

Live VM migration. In Equation 4, we define the binary variable $y_{ij}(t)$ that indicates a VM v_i is under migration to a PM j at time t. This equation states that each VM from the set $migrated(t)$ can be migrated exactly to one PM.

$$\sum_{j=1}^{M} y_{ij} = 1, \quad \forall\ v_i \in migrated(t) \tag{4}$$

In our model, we assume that the migration of a VM will not affect the resources of a target PM until the migration is completed. Equation 5 states how DPR depends on the RAM size of a migrated VM:

$$dpr_i(t) = f(vr_{iram}), \quad \forall\ v_i \in migrated(t) \tag{5}$$

where $dpr_i(t)$ is the DPR of the VM and f is a custom defined functional dependency. For simplicity we assume f is a certain linear function. The amount of migrated RAM of VM i to another PM, $migratedRAM_i(t)$, is computed by Equation 6, where $bw(t)$ is a bandwidth speed rate between the source and the target PMs and $\Delta(t)$ is the period when the VM has been under migration.

$$migratedRAM_i(t) = \frac{bw(t) \cdot \Delta(t)}{dpr_i(t)} \tag{6}$$

Energy consumption and costs. We utilize a commonly used technique for power saving, namely *Dynamic Voltage and Frequency Scaling* (DVFS) [20]. DVFS allows to adjust the frequency of a microprocessor and thereby to reduce power consumption. In our model, energy consumption of a certain PM j is defined by CPU utilization and is stated in Equation 7, where f is the power specification of the PM:

$$W_j = f(U_{jCPU}) \cdot \Delta(t) \tag{7}$$

Energy consumption of a DC is the sum of all included PM energy consumptions plus energy consumption for cooling. As originally was modeled in [5], overall DC's energy consumption is defined in Equation 8:

$$W_{DC} = \sum_{i=1}^{n} W_i \cdot pPUE_{DC}(T) \tag{8}$$

where $pPUE_{DC}(T)$ is the pPUE rate of a DC at temperature T. Energy costs of a DC for a given period of time depend on energy price at that period and the amount of consumption. In our model, energy costs are defined in Equation 9, where P_{DC} is the energy price at the DC location.

$$C_{DC} = W_{DC} \cdot P_{DC} \tag{9}$$

4.2 Decision Model

Building a decision model is started with the definition of objectives and an appropriate set of actions that allow to achieve the goal. The goal of a cloud provider is to reduce its operating cost while satisfying its customers in terms of QoS. In our model, the set of possible decision actions are *Allocate VM*, *Migrate VM*, *Switch-on PM*, and *Switch-off PM*. Afterwards, we identify a set of criteria which are important to be considered from the cloud provider's point of view during the VM Placement. Each criterion is a function of a certain quantitative measurement of a cloud infrastructure. Table 1 contains the list of criteria used in our model.

There are several assumptions based on which we define the migrated list, *migrated*(t). We assume that penalty costs are relatively high for all requests (e.g., see Table 4), hence the cloud provider should avoid placement of VMs to PMs where their SLAs can be violated with a high possibility. Therefore we define *VM unavailability* (g_1) that can be computed according to Equation 10:

$$g_1 = \frac{\sum(downtime\ duration\ v)_i}{billing\ period}, \quad \text{where } v \in allocated(t) \tag{10}$$

where the numerator is the duration of VM downtime i during the billing period. Higher value of this criterion increases the possibility of SLA violation.

The second criterion, *PM power consumption* (g_2), directly influences energy costs of the cloud provider. Equation 11 states the calculation of g_2 for a certain PM j:

$$g_2 = (W_{max} - W_j \cdot pPUE_{DC}(T))/W_{max} \tag{11}$$

where W_{max} is a constant that defines the maximal utilized power of a PM by considering of energy consumption for cooling. While W_j is the PM power consumption (Equation 7), $pPUE_{DC}(T)$, as introduced in Equation 8, is the pPUE rate of the DC at temperature T. Equation 11 utilizes the pPUE rate of a DC where a certain PM is hosted. Indeed, we define g_2 as a function where values closer to 1 are preferred over the values close to 0.

PM CPU utilization (g_3) is an indicator for efficient energy consumption. Although a cloud provider tends to utilize as less resources as possible, it should consider the risk of higher CPU demands than the PM capacity which may lead to QoS degradation.

Runtime load balancing in the cloud is performed via live VM migration. Since lower *VM migration duration* (g_4) value decreases the period of VM reallocation, it allows more efficient usage of cloud resources.

We define *Energy price* (g_5) as another factor that explicitly impacts energy costs of a cloud provider. Equation 12 defines the calculation of this criterion:

$$g_5 = (P_{max} - P_{target\ PM})/P_{max} \tag{12}$$

where P_{max} defines the maximal energy price over all geo-distributed DCs managed by a certain cloud provider, and $P_{target\ PM}$ is the energy price of a DC which hosts the target PM where the VM is going to be migrated to.

Table 1: The list of criteria used for the VM placement problem in our model.

criteria	**abbreviation**	**related equations**
VM unavailability	g_1	Eq. 10
PM power consumption (incl. cooling)	g_2	Eqs. 7, 11
PM CPU utilization	g_3	Eqs. 1, 2, 3
VM migration duration	g_4	Eqs. 4, 5, 6
Energy price	g_5	Eq. 12

In summary, while some of these criteria can be directly measured or observed (e.g., g_2, g_3, g_5), others may depend on hidden factors (e.g., g_1, g_4). Since some criteria may depend on hidden factors, this induces a level of uncertainty during the decision making process from the cloud provider's point of view. Therefore utilizing BNs would be a proper way to handle such issues and reason about these levels of uncertainty. In the remaining of this section, BN model and MCDA technique, which are used in decision making process of our approach, will be introduced.

Bayesian Network. Bayesian Networks are graphical models that represent variables of interest (e.g., object features, event occurrences) and probabilistic dependencies among them via direct acyclic graph. The main benefit of such models lays in possibility to simulate the mechanism of exploring causal relations between key factors using Bayes theorem. This theorem explains the probability of an event based on the conditions related to the event [21].

MCDA. Although a BN model can be efficiently used to aid decision making by observing the value of uncertainty corresponding to each node, the VM placement problem, which is addressed in this work, is a multi-criteria decision problem, so utilizing the BNs alone is insufficient and using methods such as MCDA is necessary. While deciding about multiple criteria, each alternative can

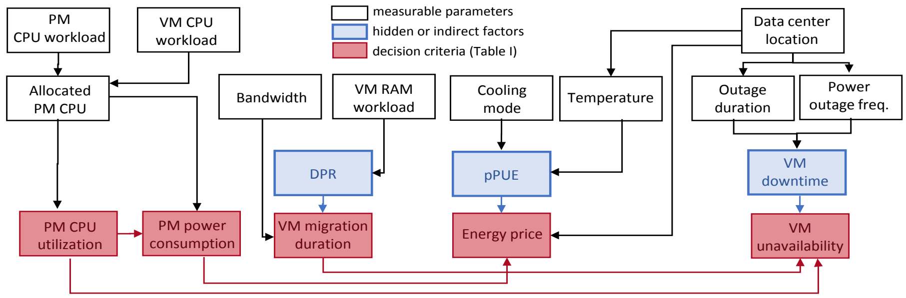


Fig. 1: A snapshot of the designed Bayesian Network.

be in conflict with the others, therefore all criteria should be combined simultaneously and be evaluated considering their preferences. More specifically, MCDA is a mean to combine measured results and rank all alternatives. These alternatives are evaluated based on a set of criteria. MCDA allows sophisticated and flexible utilization of BNs in decision making analysis [22].

4.3 Phases

The proposed VM placement approach works in the following three phases which will be explained in this section:

1. designing the BN to represent expert domain knowledge on cloud infrastructure management;
2. using GQM method to define the underlying measures for the chosen criteria based on the BN's output;
3. applying MCDA method to create the utility function as the final decision making indicator.

Phase 1: designing Bayesian Network. As the first phase of the proposed approach, a BN depicted in Figure 1 is constructed. This figure represents a simplified snapshot[2] of the BN. The structure of the BN was defined based on the extracted knowledge of cloud management experts. There are three types of nodes in this network. The white nodes define the parameters that can be directly measured at runtime. The red nodes denote the criteria which were presented in Table 1 and influence on the VM placement decision making. The values of the red nodes are further used as the inputs of the utility function. The blue nodes are the hidden factors that indirectly affect the decision making. The probabilistic dependencies between nodes are determined based on the experts knowledge, PM specifications, and the provided temperature and outage statistics.

Furthermore, in order to react earlier to runtime situation (i.e., having proactive behavior) and to make the optimum decision about each action in the cloud environment, predicted resource workload values are used in the BN model (Figure 1). For forecasting of these data, we use the three **policies** introduced in Table 2. As defined in this table, each policy use a different technique to predict the future workload.

[2] A complete snapshot of the designed BN: `https://goo.gl/Gt4DX6`

Table 2: The workload prediction policies used in BN as input.

policy name	**abbrev.**	**definition**
Last Workload	BN-LW	next workload value equals to the last one
Trend Workload	BN-TW	values follows a certain linear trend
Linear Regression Workload	BN-LRW	applying linear regression on historical data

Phase 2: using GQM method. While the BN was constructed at the first phase, the second phase is utilizing GQM method to define the underlying measures for the chosen criteria. Table 3 represents the mapping of each criterion value $\in [0, 100]$ to its corresponding utility value $\in [0, 1]$, where 0 denotes the worst value and 1 indicates the best value. The mapping for g_3 was found empirically, while for the other criteria, we consider 10% increment of their value as 0.1 increment of the corresponding utility value. In case of g_3, if the value exceeds 100% (i.e., over-usage of CPU), the selected action which leads to this condition will be immediately rejected.

Table 3: Mapping the criteria values $\in [0, 100]$ to the utility values $\in [0, 1]$.

g_1,g_2,g_4,g_5 [%]	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
utility value	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1

g_3 [%]	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
utility value	0.125	0.25	0.375	0.5	0.625	0.750	0.875	1	0.66	0.33

Phase 3: applying MCDA. As the third phase, once values for each criterion are computed, MCDA is applied in order to combine the values for each possible action, namely allocation or migration to some PM, and rank the results. Each criterion g_i is defined with a weight w_i that represents its relative importance in the context of the given decision problem. Equation 13 defines the utility function $U(a)$ of an action a:

$$U(a) = \sum w_i \cdot g_i(a) \tag{13}$$

The utility function $U(a)$ is used to evaluate the benefits of the possible actions. A VM is either allocated or migrated to a PM with the highest utility value.

4.4 Algorithms

We propose a VM Placement algorithm based on the defined utility function (Equation 13) to leverage the most important criteria of cloud infrastructures such as energy prices, resources consumption, VM availability, and penalty costs of SLA violation. VM placement algorithm implies the usage of the BN model based on the application of MCDA approach. VM placement can be divided into allocation of incoming VM requests to PMs (i.e., the *Allocate VM* action) and consolidation of the current running VMs (i.e., the *Migrate VM* action). The VM allocation is triggered when a new VM request arrives, while the VM consolidation is applied periodically on running infrastructure at each interval. Both VM allocation and consolidation problems can be reduced to a Bin-Packing problem. In our work, we use a modification of BFD heuristic [23] that will be introduced in this section.

Allocation Algorithm. As presented in Algorithm 1, first VMs $\in waiting(t)$ are sorted in decreasing order by their SLA priorities. In our approach we support three SLA priority levels, namely *Gold*, *Silver*, *Bronze*, which define the priority of resource allocation for a VM. A VM with a *Gold* SLA has the highest priority for the resource allocation and consequently has the highest penalty cost (e.g., see Table 4) in case of SLA violation. Afterwards, based on the computed utility value for each PM according to Equation 13, a PM with the highest utility value is chosen as the target PM for allocating the VM. If the chosen PM is off, the action *Switch-on PM* is applied (Lines 12-13).

Algorithm 1: allocation algorithm.

```
    input  : pmList, vmList: vm ∈ waiting(t)
    output: vmAllocationMap
1  vmList.sortDecreasingSLAPenalty();
2  foreach vm ∈ vmList do
3      maxUtility ← 0 ;
4      allocateToPm ← NULL;
5      foreach pm ∈ pmList do
6          utility ← computeUtility(pm,vm);                    ▷ Eq. 13
7          if utility > maxUtility then
8              allocateToPm ← pm;
9              maxUtility ← utility;
10     if allocateToPm ≠ NULL then
11         vmAllocationMap.put(vm, allocateToPm);
12         if allocateToPm.isSwitchedOff() then
13             allocateToPm.switchOn();
14 return vmAllocationMap;
```

Consolidation Algorithm. The consolidation of the running VMs is performed in two phases. **First**, we detect the VMs that need to be consolidated, based on the calculated utility value for each PM using different prediction workload policies introduced in Table 2. **Second**, we migrate them according to Algorithm 2 similar to the allocation algorithm (Algorithm 1). Migration of a VM to a certain PM is triggered, if the utility value of that PM is higher than the utility value of the PM where the VM is currently allocated in. Algorithm 2 triggers *Switch-off PM* action, if there is no allocated VMs on this PM (Lines 13-15).

5 CloudNet, a Novel Simulation Framework

CloudNet is a novel framework, proposed in this work, which allows cloud providers to simulate their infrastructure in a repeatable and controllable way, in order to find the performance bottlenecks, and evaluate the different management scenarios under real world data traces. The most important feature of *CloudNet* that distinguishes it from the other similar frameworks is the ability of simulating distributed DCs while taking into consideration the energy and

Algorithm 2: consolidation algorithm.

```
input  : pmList, vmList: vm ∈ allocated(t) \ migrated(t), vmAllocationMap
output: vmMigrationMap
vmList.sortDecreasingSLAPenalty();
foreach vm ∈ vmList do
    maxUtility ← 0 ;
    migrateToPm ← NULL;
    foreach pm in pmList do
        utility ← computeUtility(pm,vm);                ▷ Eq. 13
        if utility > maxUtility then
            migrateToPm ← pm;
            maxUtility ← utility;
    currentPm ← vmAllocationMap.get(vm);
    if migrateToPm ≠ currentPm then
        vmMigrationMap.put(vm, migrateToPm);
foreach pm ∈ pmList do
    if pm.hasNoVMs() then
        pm.switchOff();
return vmMigrationMap;
```

cooling costs, power outages, weather temperature, and energy prices. It also supports several SLA priority levels with different penalty costs. For the evaluation of our VM placement approach, we designed *CloudNet* to be able to simulate the management of geo-distributed DCs with frequent power outages. To facilitate the reproduction of our research, we released the source code of *CloudNet*[3]. More implementation details can be also found in [25].

Table 4: *CloudNet* general configuration.

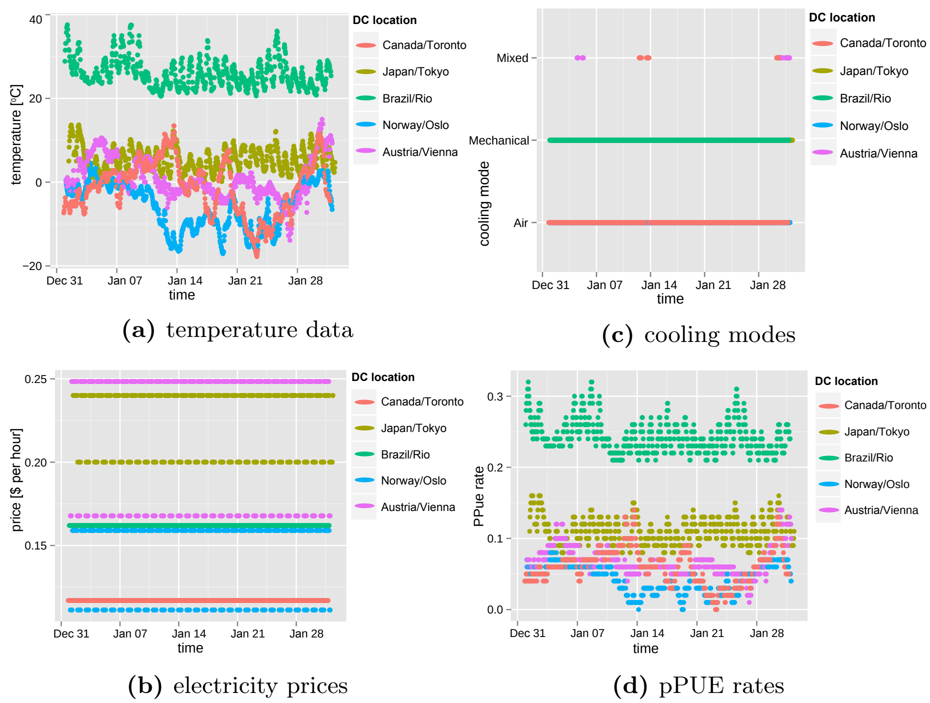
#DCs	#PMs	#VMs	Bronze.SLA penalty	Silver.SLA penalty	Gold.SLA penalty
5	25	25	0.05$/% violation	0.1$/% violation	0.2$/% violation

Table 5: *CloudNet* geo-temporal input parameters for each DC.

Data center	Brazil	Canada	Norway	Austria	Japan
Day/Night switch hours (hour)	8-23	8-23	8-23	6-22	8-23
Day energy price ($/kWh)	0.162	0.117	0.159	0.2484	0.24
Night energy price ($/kWh)	0.162	0.117	0.1113	0.1678	0.20
SAIDI (min/month)	1101.6	220	218	39	6

To evaluate the proposed algorithms based on the regional electricity prices and temperature differences, we first configured *CloudNet* as summarised in Table 4. In our simulation, we have one VM type (1000MIPS, 768MB RAM) and

[3] `https://github.com/dmitrygrig/CloudNet`



(a) temperature data

(c) cooling modes

(b) electricity prices

(d) pPUE rates

Fig. 2: Real world data traces used as inputs for the chosen DCs and evaluation period. The sources of the exposed data are as follows: a [24], b [18], c and d [5].

one PM type in terms of resource capacity (3000MIPS, 4GB RAM). Furthermore, each VM has an availability-related SLA metric which is defined as the time of the overall downtime per billing period. Note that the used billing period is one month in our work. As mentioned before, our approach supports three SLA priority levels, namely *Gold*, *Silver*, *Bronze*, with different SLA penalty costs which are shown in Table 4.

For the evaluation, we setup *CloudNet* with five distributed DCs in different time zones. Each location has regional electricity prices and temperature values which are time-dependent. The temperature changes can cause the necessity of using different cooling modes and directly effect the pPUE rate. The geo-temporal input parameters for each DC are shown in Table 5.

The chosen locations for DCs allow to evaluate the proposed algorithms considering a combination of various real world input data such as electricity price, power outage statistics, cooling models, and temperature. Conducting the evaluations, we simulate one month (from January 1 till February 1, 2013) operation of running distributed DCs, with management interval of 1 hour. We use the following real data traces as the input parameters of *CloudNet*:

- **Temperature data**. We retrieved the real temperature data traces for the chosen period and locations from the public web service, `Forecast.IO` [24] with the granularity of 1 hour (see Figure 2a).

– **Cooling modes**. We simulated Emerson's DSE^{TM} cooling system, described in [5]. This system has three different cooling modes: *Air*, *Mechanical*, and *Mixed*. One mode is switched to another one when the outside temperature is changed. In our evaluation, we switch *Air* mode to *Mixed* after temperature exceeds 12°C and *Mixed* to *Mechanical* after exceeds 18°C. Figure 2b depicts the switching between various modes of cooling system.
– **Electricity prices**. Electricity prices for the chosen locations are defined using statistics in [18]. Some locations such as Austria have different pricing models for day and night as shown in Table 5. Figure 2c shows changes of electricity prices for different locations.
– **Power outage statistics.** We obtained the data traces of the power outages corresponding to the chosen locations and period for our simulation from [19]. As shown in Table 5, the electric measure *system average interruption duration index* (SAIDI) is utilized. Note that we used the real values of a year for the simulation period of a month in order to better show the ability of our approach in handling more unreliable DCs in terms of power outage.
– **PM power specification**. We use data traces of *SPECpower benchmark* (HP ML110[4]) to define power specification for each PM in simulated DCs.

Figure 2d shows the values of the pPUE rate of the chosen DCs. In general, as shown in Figures 2a, and 2d, a lower temperature drastically decreases pPUE and hence is more energy efficient due to the lower cooling cost.

6 Evaluation

In this section the proposed cost-aware VM placement approach is evaluated using the designed BN based on data extracted from the real-world traces in the presented simulated environment (Section 5).

6.1 Baseline Algorithms

The proposed VM placement approach under three introduced workload prediction policies (see Table 2) is evaluated in comparison with the two baseline approaches, FFD and an approach named NoM that follows a First-Fit VM allocation strategy, but does not support VM migration. The rationale behind having such a baseline is to show that the migration does not have inverse effect on a cost-aware VM placement algorithm in terms of SLA violation. All approaches have the same input parameters and configurations. The goal of the evaluation is to show the ability of the proposed approach to use the designed BN model which includes the extracted information about the cloud infrastructure in order to perform more efficient decisions concerning placement of VMs across distributed DCs. An approach is said to be better if the total cost including the energy cost and the penalty cost of the SLA violation is minimised. To get comparison results with FFD approach, we used it under two resource allocation policies. The first policy, *First-Fit-Decreasing Agreed* (FFD-A), statistically

[4] `https://goo.gl/ZvmK6o`

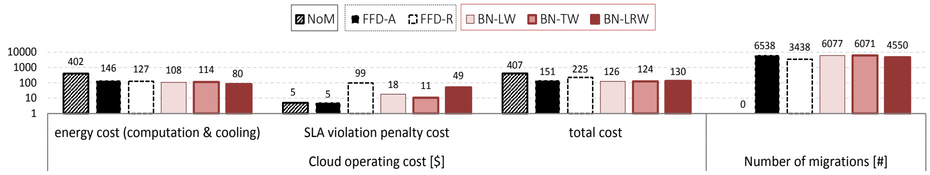


Fig. 3: Aggregated evaluation results throughout one month of simulation.

allocates the amount of resources agreed by the SLA, while the second policy, *First-Fit-Decreasing Requested* (FFD-R), is more dynamic and it allocates the amount of resources which are required by the VM at runtime.

6.2 Evaluation Metrics

The following metrics are considered as the evaluation metrics: (i) *energy cost* that presents the total operating costs includes the computation cost and the cooling costs of all DCs; (ii) *SLA Violation penalty cost* that is a penalty cost and has to be paid by the cloud provider in case of SLA violation; (iii) *number of migrations* that represents the migration actions triggered during the simulation whether to avoid the SLA violation due to the power outage or due to the consolidation. In general, the migration action should be applied by a cloud provider if it has appreciable impact on the operating cost.

6.3 Results

Figure 3 shows the aggregated evaluation results obtained during the whole simulation run, one month. The usage of the proposed approach (reddish plots) leads to better results with all the three used policies, while under BN-TW policy it has the best results. BN-TW policy improves costs usage by up to 69% (124$ vs. 407$ total cost) in comparison with NoM approach. This improvement is by up to 45% (225$ vs. 124$ total cost) in comparison with FFD-R which has less number of migrations, and by up to 18% (151$ vs. 124$ total cost) in comparison with FFD-A which has more number of migrations.

The results reveal that the usage of a more enhanced prediction policy, e.g. linear regression (BN-LRW policy) in our approach, increases the cost efficiency in the terms of energy costs (incl. computation and cooling), however it has more SLA violations in comparison with other used policies (i.e., BN-LW, BN-TW).

Moreover, the results in the case of NoM approach in comparison with FFD and the proposed approach indicate the necessity of supporting migration strategy while managing distributed DCs in order to decrease the operating cost. This is because of including the knowledge about the dynamic geo-temporal input parameters and their influence on the energy cost such as temperature, electricity price, etc. Furthermore, since by using *CloudNet* we could simulate DCs with frequent power outages, approaches like NoM even with a high energy cost, still suffer from SLA violation as they cannot handle situations like a power outage.

In comparison with FFD approach (i.e., FFD-A and FFD-R) the proposed VM placement approach gained less energy cost while keeping the penalty cost under control in a way that the total operating cost is less under all used policies. The reason is due to the utilization of the prediction workload policies, using the extracted knowledge about the cloud management (i.e., modeling power outage in cloud DCs) modeled as BN, the effectiveness of MCDA applied on BN reasoning, and supporting different SLA models. In summary, the proposed cost-aware VM placement approach under all workload prediction policies achieved better results in terms of both energy and total costs in comparison with the two baseline approaches.

7 Conclusion and Future Work

In this paper, we proposed a novel approach for cost-aware VM placement across distributed DCs to reduce the energy and penalty costs paid by a cloud provider. This approach creates a decision model using Bayesian Networks, and then applies an MCDA method along with two proposed algorithms for the VM allocation and consolidation. For the evaluation, we focused on geographically distributed DCs where the cloud infrastructure experiences frequent power outages, and the cloud provider aims to decrease the total cloud operating cost (i.e., power, cooling, and violation penalty) while keeping the customers satisfied in terms of less SLA violation. The proposed approach was evaluated in a novel simulation framework, which provides the features of distributed DCs, by comparing the results with two baseline algorithms, namely NoM and FFD. The simulation results showed that the proposed approach decreased the energy cost by up to 69% in comparison with NoM approach and by up to 45% compared to the FFD approach.

We plan to extend this work along several directions: (i) enhancing VM placement by using *Kalman Filtering* as a workload prediction technique; (ii) utilizing *hybrid Bayesian Networks* to be able to use the analogous data and make the BNs' parameters more precise. The future extensions will lead to make even more cost-optimized VM placement decisions.

Acknowledgement

This work was partially supported by the adaptive distributed systems doctoral college and the HALEY project at Vienna University of Technology, and the Vienna Science and Technology Fund (WWTF) through the PROSEED grant.

References

1. McKendrick, J.: Cloud Computing's Hidden Green Benefits `http://www.forbes.com/sites/joemckendrick/2011/10/03/cloud-computings-hidden-green-benefits`.
2. Li, J., Shuang, K., Su, S., et. al: Reducing Operational Costs through Consolidation with Resource Prediction in the Cloud. In: International Symposium on Cluster, Cloud and Grid Computing, IEEE (2012) 793–798
3. Li, K., Zheng, H., Wu, J.: Migration-based Virtual Machine Placement in Cloud Systems. In: International Conference on Cloud Networking (CloudNet), IEEE (2013) 83–90

4. Masoumzadeh, S.S., Hlavacs, H.: Integrating VM Selection Criteria in Distributed Dynamic VM Consolidation Using Fuzzy Q-Learning. In: International Conference on Network and Service Management (CNSM), IEEE (2013) 332–338
5. Xu, H., Feng, C., Li, B.: Temperature Aware Workload Management in Geo-distributed Datacenters. In: SIGMETRICS Performance Evaluation Review (PER). Volume 41., ACM (2013) 373–374
6. Akoush, S., Sohan, R., et. al: Predicting the Performance of Virtual Machine Migration. In: International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), IEEE (2010) 37–46
7. Premchaiswadi, W.: Bayesian Networks. InTech (2012)
8. Basili, V.R., et. al: The Goal Question Metric Approach. In: Encyclopedia of Software Engineering. Wiley (1994)
9. Fenton, N., Neil, M.: Making Decisions: Using Bayesian Nets and MCDA. Knowledge-Based Systems **14**(7) (2001) 307–325
10. Beloglazov, et. el: Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. Future generation computer systems (FGCS) **28**(5) (2012) 755–768
11. Calcavecchia, N.M., Biran, O., et. al: VM Placement Strategies for Cloud Scenarios. In: International Conference on Cloud Computing (CLOUD), IEEE (2012) 852–859
12. Song, Y., Sun, Y., Shi, W.: A Two-Tiered On-Demand Resource Allocation Mechanism for VM-Based Data Centers. Transactions on Services Computing (TSC) **6**(1) (2013) 116–129
13. Lučanin, D., Jrad, F., et. al: Energy-Aware Cloud Management Through Progressive SLA Specification. In: Economics of Grids, Clouds, Systems, and Services (GECON), Springer (2014) 83–98
14. Altmann, J., Kashef, M.M.: Cost model based service placement in federated hybrid clouds. Future Generation Computer Systems (FGCS) **41** (2014) 79–90
15. Calheiros, R.N., Ranjan, R., et. al: CloudSim: a Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. Software: Practice and Experience **41**(1) (2011) 23–50
16. Banzai, T., Koizumi, H., et. al: Design of a Software Testing Environment for Reliable Distributed Systems Using Cloud Computing Technology. In: International Conference on Cluster, Cloud and Grid Computing, IEEE (2010) 631–636
17. Joshi, P., Gunawi, H.S., Sen, K.: PreFail: a Programmable Tool for Multiple-Failure Injection. In: SIGPLAN Notices. Volume 46., ACM (2011) 171–188
18. online: E-Pricing `http://en.wikipedia.org/wiki/Electricity_pricing`.
19. online: World Electrical Outages Statistics `http://www.nationmaster.com/country-info/stats/Energy/Electrical-outages/Days`.
20. Kim, W., Gupta, M.S., et. al: System Level Analysis of Fast, per-core DVFS using On-chip Switching Regulators. In: International Symposium on High Performance Computer Architecture (HPCA), IEEE (2008) 123–134
21. Pollino, C., Henderson, C.: Bayesian Networks: A Guide for their Application in Natural Resource Management and Policy. Technical report (2010)
22. Vincke, P.: Multicriteria decision-aid. Multi-Criteria Decision Analysis (MCDA) **3**(2) (1994) 131
23. Yue, M.: A Simple Proof of the Inequality FFD (L)$<$11/9 OPT (L)$+$ 1,for all l for the FFD Bin-packing Algorithm. Acta mathematicae applicatae sinica (AMAS) **7**(4) (1991) 321–331
24. online: Forecast Weather Web Service `http://www.forecast.io`.
25. Grygorenko, D.: Cost-based Decision Making in Cloud Environments using Bayesian Networks. Master thesis, Vienna University of Technology, Austria (2014)