

 Open access • Journal Article • DOI:10.1109/TCC.2015.2459710

Exploiting Renewable Sources: When Green SLA Becomes a Possible Reality in Cloud Computing — [Source link](#)

Sabbir Hasan, Yousri Kouki, Thomas Ledoux, Jean-Louis Pázat

Institutions: French Institute for Research in Computer Science and Automation, École des mines de Nantes

Published on: 01 Apr 2017 - IEEE Transactions on Cloud Computing (IEEE)

Topics: Green computing, Service-level agreement, Cloud computing, Energy consumption and Virtualization

Related papers:

- [Smart management of renewable energy in clouds : from infrastructure to application](#)
- [Eco-Aware Online Power Management and Load Scheduling for Green Cloud Datacenters](#)
- [GreenCassandra: Using renewable energy in distributed structured storage systems](#)
- [Enabling Green Energy Awareness in Interactive Cloud Application](#)
- [The green cloud effective framework: An environment friendly approach reducing CO 2 level](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/exploiting-renewable-sources-when-green-sla-becomes-a-si726jt8op>



HAL
open science

Exploiting Renewable sources: when Green SLA becomes a possible reality in Cloud computing

Md Sabbir Hasan, Yousri Kouki, Thomas Ledoux, Jean-Louis Pazat

► To cite this version:

Md Sabbir Hasan, Yousri Kouki, Thomas Ledoux, Jean-Louis Pazat. Exploiting Renewable sources: when Green SLA becomes a possible reality in Cloud computing. *IEEE Transactions on Cloud Computing*, IEEE, 2017, 5 (2), pp.249-262. 10.1109/TCC.2015.2459710 . hal-01187907

HAL Id: hal-01187907

<https://hal.archives-ouvertes.fr/hal-01187907>

Submitted on 15 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exploiting Renewable sources: when Green SLA becomes a possible reality in Cloud computing

Md Sabbir Hasan, Yousri Kouki, Thomas Ledoux, and Jean Louis Pazat

Abstract—While the proliferation of Cloud services have greatly impacted our society, how green are these services is yet to be answered. Although, demand escalation for green services has grown due to societal awareness, the approaches to provide green services and establish Green SLAs remain oblivious for cloud or infrastructure providers. The main challenge for cloud provider is to manage Green SLAs with their customers while satisfying their business objectives, such as maximizing profits by lowering expenditure for green energy. Since, *Green SLA* needs to be proposed based on the presence of green energy, the intermittent nature of renewable sources makes it difficult to be achieved. In response, this paper presents a scheme for green energy management in the presence of explicit and implicit integration of renewable energy in data center. More specifically we propose three contributions: i) we introduce the concept of virtualization of green energy to address the uncertainty of green energy availability, ii) we extend the Cloud Service Level Agreement (CSLA) language to support *Green SLA* by introducing two new threshold parameters and iii) we introduce *greenSLA* algorithm which leverages the concept of virtualization of green energy to provide per interval specific *Green SLA*. Experiments were conducted with real workload profile from PlanetLab and server power model from SPECpower to demonstrate that, *Green SLA* can be successfully established and satisfied without incurring higher cost.

Index Terms—Cloud Computing, Sustainable Computing, Renewable Energy, Green Service Level Agreement, Cross-layer SLA



1 INTRODUCTION

IN 2007, data centers in Western Europe consumed a whopping 56 terawatt-hours (TWh) of power per year. According to the EU, this figure is likely to almost double to 104TWh by 2020¹. This translates into high carbon emissions as most of the electricity comes from fossil fuel or coals, only exception in France where 85% electricity is produced by nuclear plants, causing relatively lower carbon footprint than any other country. Still, the amount of carbon footprint is nowhere near to zero.

Subsequently, similar to other large consumers of power, data centers find themselves increasingly pressured either by legislation or by public opinion to find options to reduce their carbon footprint. Therefore, demands for green products and services are ever increasing as social awareness for being green is hiking. In response, the service providers have started to set sustainability goals and create initiatives to reduce carbon emissions by using renewable sources to their services. Although several research efforts have been made to reduce energy consumption by designing/implementing server consolidation, hardware with better power/performance tradeoffs, workload migration and software

technique for energy aware scheduling, still the goal for alleviating carbon footprint is being underachieved. Given the circumstances, explicit or implicit integration of renewable energy to the data center can be the only way to reduce carbon footprint at an acceptable level. Besides that, demand for green services is ever increasing, thus integrating renewable sources to the data center left no choice. Few smaller green cloud providers, e.g., GreenCloud², Green House Data³ and academic researchers [1] integrated renewable sources to the data center explicitly which offers green computing services. However, renewable sources are known to be very intermittent in nature, thus providing Green services or running Servers and Virtual machines (VMs) only by on-site renewable energy becomes very unrealistic. Moreover, some research efforts have also explored how to incorporate off-site renewable energy to data center, as the best location for producing renewable energy does not always have the best potential to build a data center. Transporting the off-site energy is arduous as wheeling charge imposed by the Grid might be more than the expectation and power losses through transmission lines are inevitable. Besides the above explicit involvement, some implicit options for reducing carbon footprint also exist through renewable energy certificate (REC) and power purchasing agreement (PPA). As example, tech giant companies like Google and Apple are investing to nearby big wind and solar plants through PPA to reduce their carbon footprint. On the other hand, REC is a tradable commodity proving that, electricity was generated using renewable sources. Therefore, purchasing of a green certificate equals to purchasing a claim that the certificate owner consumed energy from the renewable portion of the

- S. Hasan is with ASCOLA (EMN-INRIA, LINA) and MYRIADS (IRISA-INRIA) Research Groups. INSA de Rennes, France.
E-mail: Firstname.Lastname@inria.fr
- Y. Kouki and T. Ledoux are with ASCOLA Research Group EMN-INRIA, LINA. Ecole des Mines de Nantes, France.
E-mail: Firstname.Lastname@inria.fr
- J. L. Pazat is with the MYRIADS Research Group, INSA, INRIA, IRISA. Rennes, France.
E-mail: jean-louis.pazat@irisa.fr

1. European Commission, Code of Conduct on Data centers Energy Efficiency- Version 1.0, October 30, 2008

2. <https://www.greencloud.com/>

3. <http://www.greenhousedata.com/>

whole energy grid [2].

Considering all the above implicit and explicit green energy integration option to data center, any service providers are yet to propose Service Level Agreement (SLA) based on green energy availability with their infrastructure. Haque et al. [1] first proposed *Green SLA* based on green energy availability from on-site renewable plant where environmental conscious clients can ask for differentiated *green* services with varied green energy requirement. However, providing green service can be terminated or no additional green service request will be entertained when green energy is not available, which quantifies that, there is no evidence of having formal contracts establishment between Software-as-a-Service (SaaS) providers/clients and Infrastructure-as-a-Service (IaaS) providers.

To address this problem, we investigated *Cloud Energy Broker* in our previous work [4] and now are extending our research to propose *Green SLA* based on the availability of green energy by introducing the concept of Virtualization of green energy. The energy can be virtually green for a specific period of time if abundance of green energy is available aperiodically in shorter time interval along with the deficit of green energy in rest of the time frame. Therefore, the virtualization concept can increase the greenness of energy, rather increasing the amount of green energy. Concretely, when the availability of green energy is more than demand, we use the whole portion of available green energy but characterize the interval as surplus interval. When green energy is insufficient to meet the demand, we nullify the degraded interval with the surplus interval. We use the term *virtualization* because we nullify a degraded interval (lack of green energy) with a surplus interval (excessive green energy than demand), but from the client's or SaaS provider's perspective, they realize both the interval as ideal interval (when supply meet the demand), though the green energy was not present instantaneously rather present virtually. Figure 1 shows the visualization of our approach. In our proposed approach, *Green SLA* can be monitored and evaluated per interval (time slot) rather per-request or application. Furthermore, we extend CSLA [3], the Cloud Service Level Agreement language allowing to define SLA in any language for any Cloud service (XaaS) to support *Green SLA* based on the concept of virtualization of green energy. The proposed new version expresses a Service Level Objective (SLO) using two thresholds and supports a new class: 'the surplus' that is characterized to nullify degraded interval. Besides, the nullification process reflects the instability of green energy as well. Furthermore, we introduce *greenSLA* algorithm which leverages the concept of virtualization of green energy and CSLA extension to validate our proposal. For experiment purpose, real workload profile from Planet-Lab and server power model from SPECpower benchmark⁴ were traced for conducting experiments. Results demonstrate that, *Green SLA* can be successfully established and satisfied without incurring higher cost for total energy expenditure.

The rest of this paper is organized as follows. Section 2 illustrates the background and motivation, which is used to ease the understanding of our approach. Section 3 describes

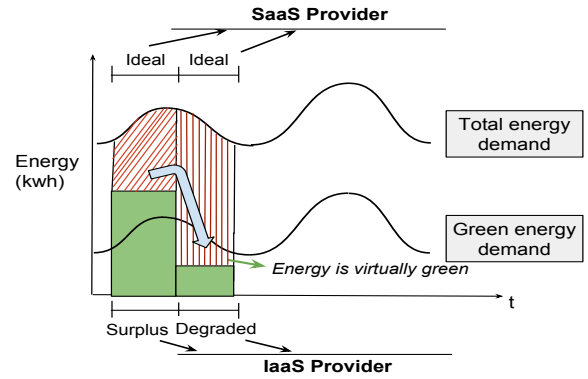


Fig. 1: Concept of Virtualize green energy

the system model and the algorithmic solution. The results obtained from experimental evaluation are presented and discussed in Section 4. Section 5 provides a selection of relevant work related to this paper. Finally, Section 6 concludes this paper and provides some discussion on future work.

2 BACKGROUND AND MOTIVATION

The Cloud architecture is usually composed of several XaaS layers and SLAs are characterized at various levels in this stack to ensure the expected QoS for different stakeholders. As shown in Figure 2, an end-user is a client of the SaaS provider, which is itself a client of the IaaS provider and as well as for Energy-as-a-Service (EaaS) provider. We investigated Service level objective (SLO) for every layer and presented in the Figure 2. Availability of resource and response time are treated as the basic SLO between end client and SaaS provider. On the other hand, availability of physical resources were the only objective so far in between the SaaS and IaaS layer, but we proclaim to create a new class of SLO, which should be treated based on green energy availability. Depending on the requirement from the upper layer, IaaS provider can have a formal contract with EaaS providers to have a portion of brown and green energy to fulfill its contracted SLA to the upper layers e.g., SaaS provider, end client. Therefore, we discuss how SLA can be established based on green energy, what are the green energy sources we can exploit and how we can take leverage the concept of virtualization of green energy to satisfy the *Green SLA*.

2.1 Green power-driven SLA

Due to the dynamic nature of the Cloud, SLA between consumers and providers emerge as a key aspect and SLA enforcement becomes an important challenge. Today's research is more concentrated on Workload-driven SLA rather than Power-driven SLA and *Green* power-driven SLA. Usually, Workload-driven SLA depends on end-users criteria such as availability, response time, throughput, etc. In contrast, Power-driven approach implies, shifting or scheduling the deferrable workloads to the time period when the price of electricity is lower or migrating workloads to the different region (data center) where the electricity price is cheaper than the origin with respecting the deadline. On the contrary, *Green* power-driven SLA can be realized as: end-users

4. https://www.spec.org/power_ssj2008/

or SaaS providers shift their workloads in a renewable/green energy powered data center having an agreement with IaaS provider that some portion of their workload should run in a greener environment. Existing literature does not provide a clear idea about the advantages and disadvantages of different integration option of renewable energy sources in data centers. Although some research [5][6][7][8] have explored the opportunity of integrating renewable sources in data center, but lacks the explanation of how SLA should be established between IaaS and SaaS providers based on the green energy availability.

To address this problem, we propose a green power driven SLA framework established between SaaS and IaaS provider stating that, IaaS provider provides infrastructure with proportional e.g., 30% green energy availability. For instance, IaaS provider will have a formal contract with SaaS provider to provide green infrastructure based on a business model. In Figure 2, SLA_{S-I} contract used for showing two SLO of this layer, namely availability of physical resource and availability of green resource. Point to be noted that, a substantial amount of research [9][10][11][12][17][18] has been already done both in industry and academia about efficient dynamic consolidation of PM, migration of VM and scalability issues in Cloud infrastructure. In position to these existing research, our work can be seen as complementary to their research since reducing energy consumption in infrastructure level and associating green sources can reduce carbon footprint in data center from the global point of view. Therefore, we are concerned about how to make a data center green and what are the options to make it possible by penetrating available green energy sources based on *Green SLA*, i.e., service level objectives about green resource/energy. Indeed, we argue that *Green SLA* should be established by taking into account the presence of green energy rather just reducing the energy consumption in the infrastructure level.

2.2 Green energy sources

From Figure 2, it can be shown that, the energy layer of the Cloud computing infrastructure consists of a single Grid where several *Green Energy-as-a-Service* (GEaaS) providers from REC market and green energy provider from spot market are connected. Additionally, on-site renewable energy plant can be associated to the data center internally or externally through the same Grid.

- *REC market*: We exploit the REC market where multiple GEaaS providers produce energy and feed to the Grid. Considering the Cloud infrastructure as a stack of XaaS layers, a IaaS provider will buy a portion of green energy dynamically from those GEaaS providers to supply green computing services to the SaaS providers or their clients. As renewable energy sources are very intermittent in nature, the renewable energy-feeding price would be very different from one to another provider depending on the location of site, availability of sources (wind speed, solar irradiation, etc.) and capacity factor of the plant. Committing to a single provider might result unavailability of required green energy requirement for certain time frame thus ensuring certain percentage

of green energy availability in data center can not be met. Moreover, in deregulated electricity market, prices vary significantly during the day depending on the energy generation and demand in the wholesale market [20]. As a consequence, most of the energy distribution company introduce different non-flat tariffs. Day-ahead pricing (DAP) is one of the many pricing methodology, which is widely used in deregulated electricity pricing market. In our previous work [4], we consider different GEaaS providers, which update DAP with information including probable generation of Green energy, price and availability. Usually, GEaaS provider add their green energy production to the Grid and IaaS provider consume the purchased energy from the same Grid, since there is only a single Global Grid in European electricity market. How IaaS and GEaaS provider can have a formal contract regarding energy procurement, is elaborately discussed in our previous work [4] as well. Therefore, we purchased green energy before hand by forecasting the green energy demand of SaaS providers.

- *Spot green energy market*: Usually spot market possess lesser amount of energy than regular energy or electricity market and price tends to be higher than traditional or different non-flat tariffs. Spot market is very important for consumer (IaaS provider), if the real-time energy/power demand is excessive than the forecasted demand. Moreover, the actual demand cannot be known accurately in advance and any forecasting technique provides at least some error statistics. So, for fulfillment of *Green SLA*, IaaS provider needs to purchase green energy from spot market if it is required.
- *On-site renewable energy*: Due to the growing demand of green services, most of today's green data centers adopted on-site renewable energy plant e.g., wind turbine, solar panel to meet the green energy demand. Nevertheless, the perfect place for constructing a renewable energy plant might not have the true potential to build a data center due the intermittent nature of the renewable sources. But having a small-scale renewable plant always gives the advantage to incorporate green energy to data center to fulfill at least the partial green energy demand if there is not sufficient amount of energy in the REC or Spot green energy market.

2.3 From energy storage to green energy virtualization

As renewable power sources are very intermittent in nature, hence predicting the amount of renewable energy production ahead of real time might demonstrate greater error statistics in data center. Nonetheless, excessive production of renewable energy can go to waste and sometimes might imbalance the Grid if the production becomes greater than the capacity. The later case doesn't apply if the data center has a small-scale renewable source e.g., wind turbine, solar plant. One way to overcome the challenge is to use energy storage or battery to store this superfluous green energy which can be discharged later for peak shaving of

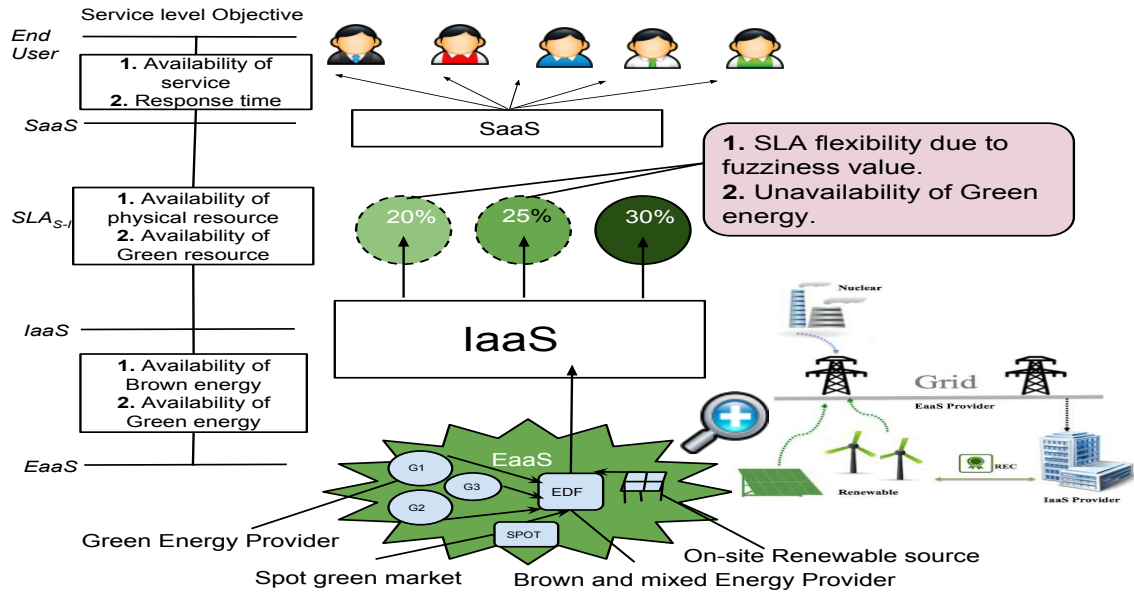


Fig. 2: Cross layer SLA

data center power demand or for fulfillment *Green SLA* between IaaS and SaaS providers when renewable energy is needed but not available. Energy storage incurs additional costs to data centers cap-ex and op-ex, hence it is not an attractive solution for small-scale data centers. Moreover, storages have finite capacities to recharge energy and their lifetime is a decreasing function of *depth-of-discharge* (DoD) and charge/discharge cycles [22]. Usually DoD refers, how much energy the battery has delivered.⁵ Therefore, if the production of renewable energy is above the capacity of storage, remaining energy goes to waste. Even the state of the art batteries have 80-85% [35] efficiency on charging and discharging capabilities, which implies 28-36% loss of energy.

Then, how to manage unavailability of green resources at run-time if the storage approach has several drawbacks, if the on-site renewable energy or spot market productions are insufficient? Our proposal is to smooth out the differences between deficit and surplus of green energy production during a certain time window with the objective to obtain an summation superior to a certain threshold, which we refer as *Virtualization of green energy*. Interestingly, this concept does not increase the total energy consumption rather increases the *greenness of energy* used in data center. In this way *real* energy storage is not needed and neither of the portion of renewable energy is wasted. Furthermore, total expenditure for energy purchasing can be reduced as no energy goes to waste and additional cost for using storage is not needed. Obviously, *Green SLA* between IaaS and SaaS providers can be fulfilled if the time-slot length is adapted. For example, if IaaS provider has established a SLA to SaaS provider to have some portion of green energy available for each time slot e.g., $T=30/60$ minutes, it is possible to satisfy the contract by using virtualization concept of green energy, which is elaborately explained in the next section.

5. It is not recommended to fully discharge batteries to 100%, otherwise it would shorten the life-cycle of batteries.

3 GREEN SLA ENFORCEMENT

In this section, we first summarize our previous work and then sketch out the Cloud Service Level Agreement language (CSLA)[3] to show how it has been extended to support *Green SLA* and the concept of virtual energy. Then, we describe the model parameters and investigate the goal for cost reduction of spot green market as well as total energy expenditure by proposed algorithmic solution based on *Green SLA*.

3.1 Previous work

Due to two time-scale green energy market (REC and Spot) and aperiodic spikes of workload we have divided our proposed work in two phase named planning and run-time phase. In planning phase [4], we proposed a *Cloud energy broker*, which can adjust the availability and price combination to buy green energy dynamically 24 hours before from the REC market to make data center green for a specific (as a example 30%) portion. We investigated a simplified power model from where we could formulated and predicted power demand of a data center for next 24 hours by evaluating 7 days of real data, traced from PlanetLab [21]. Our monitoring window length was one hour that provided almost accurate predicted information with fewer error statistics. We also have taken a realistic consideration that Green energy providers can publish a day ahead green energy generation and price per hour, which is a common practice at European electricity and energy market along with smart-grid environment. Our energy broker tried to maximize of using renewable energy under strict budget constraint whereas it also tries to minimize the use of brown energy by capping the limit of overall energy consumption of data center. Furthermore, how we manage the energy procurement and management in the run-time phase based on our proposed *Green SLA* approach, is discussed and analyzed in the following sections.

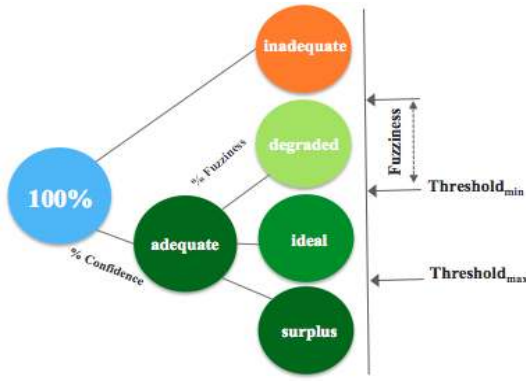


Fig. 3: SLO evaluation in CSLA

3.2 CSLA language

CSLA[3], the Cloud Service Level Agreement language, allows to define SLA in any language for any Cloud service (XaaS). CSLA addresses intrinsically (i) QoS uncertainty in unpredictable and dynamic environment; (ii) the cost model of Cloud computing. Indeed, CSLA allows the expression of sophisticated Service Level Objectives (SLOs) with new features such as confidence and fuzziness to deal with QoS uncertainty: (i) the fuzziness defines the acceptable margin degree around the threshold of an expression; (ii) the confidence defines the percentage of compliance of clauses.

In order to evaluate an objective (SLO), an initial evaluation enables to classify the interval as ideal (i.e., threshold is respected), degraded (i.e., threshold is respected using fuzziness margin) or inadequate (i.e., threshold is not respected even with fuzziness margin). We distinguish two concepts: (i) per-interval evaluation, in which the evaluation is performed at the end of each interval; (ii) per-request evaluation, in which the objective is evaluated for each request. A final evaluation, at the end of the time window, allows one to verify an objective (SLO) by applying the fuzziness and confidence percentages to the initial evaluation. The final evaluation enables the identification of non-accepted/accepted degradation and inadequate cases, that is, that will/will not result penalties. In other words, the final evaluation absorbs or notifies the violations.

In this paper, in order to allow energy virtualization (since this concept is based on a certain time frame), we consider per-interval evaluation. Further, we extend CSLA language to support the *Green SLA* by taking the advantage of the concept of virtualization of green energy. First, we define an SLO by using two thresholds $threshold_{min}$ and $threshold_{max}$ (see Figure 3). Secondly, beyond the $threshold_{max}$, we consider the intervals as surplus meaning that, excessive green energy was present in that interval. Finally, we add an intermediate step in the evaluation process. This step consists to update the intervals classes using the nullification of degraded intervals by surplus intervals i.e., for each surplus interval we translate a degraded interval to ideal interval.

Listing 1: CSLA example.

```

1 <csla:terms>
2 <csla:objective id="GreenResourceSLO" actor="IaaSProvider">
3 <csla:expression metric="Gr" comparator="in" threshold-min="25"
  threshold-max="30" unit="%" monitoring="Mon-1" Confidence=
  "91,66" fuzziness-value="5" fuzziness-percentage="18,18"/>

```

```

4 </csla:objective>
5 </csla:terms>
6 <csla:penalties>
7 <csla:Penalty id="p-Gr" objective="GreenResourceSLO" condition="
  violation" obligation="provider">
8 ...
9 </csla:Penalty>
10 </csla:penalties>

```

In the example (see Listing 1), we focus on only one SLO about the percentage of green resources (lines 1-4). The SLO states that the percentage of green resources should be guaranteed between 25 and 30, with confidence, fuzziness and percentage fuzziness of 91,66%, 5% and 18,18%, respectively. In concrete terms, it means that the percentage of green resource measured within an observation period may be i) lower then 20% in 8.34% of the observation periods, ii) between 20% and 25% in 16,66% (91.66% of 18,18%) of the observation periods and iii) greater or equal to 25% in 75%. A violation of the *GreenResourceSLO* implies a penalty that depends on the green percentage not respected (lines 7-9).

Using this objective for an evaluation window of 24 intervals, we accept 22 adequate intervals (18 ideal and surplus, 4 degraded) and 2 inadequate.

3.3 Supply side characteristics

We consider our system operates in discrete time model. From day ahead REC market, IaaS provider purchase green energy for next suitable time period e.g., 12 hours or 24 hours. Furthermore, for evaluation and validation of SLA by CSLA language we divide before mentioned time frame in t ($t = 12$ or 24 hours time period). Moreover, the total time is divided into J ($J \in \mathbb{N}^+$) coarse-grained time slots of each length of T , accordance with the length of the day-ahead REC market, e.g., minutes, hours in Figure 4. In addition, each fine-grained time slots τ , ($\tau = 30$ minutes) are treated as monitoring window and t where, $t = jT$ ($j = 1, 2, \dots, J$), can be defined as evaluation window for SLA validation in our model. IaaS provider purchase green energy from single or multiple REC providers (discussed in our prior work) in day ahead REC market for next 12 to 24 hours. So, for each fine-grained time slot, we define $d(\tau)$ where, $\tau \in (t, t + T - 1)$ is purchased with P_{max} upper bound price. As we integrate on-site renewable power source in our model, we consider $r(\tau)$ amount of renewable energy is produced and added to each fine-grained time slot. As renewable energy sources are very intermittent in nature, we suggest $r(\tau)$ to be $r(\tau) \geq 0$ for each fine-grained time slot meaning, renewable energy will not be available in some slots due to the sporadic nature of the source. Hence, the supply side consists of two independent parts, e.g., $U(\tau) = d(\tau) + r(\tau)$. As we consider, two time scale renewable energy market, if the demand of renewable/green energy is greater than the supply, IaaS provider has to buy additional energy in real-time from REC market which tends to have higher price on average than day-ahead or long-term ahead market similar to real-time electricity market. This additional energy is regarded as spot energy. We define $s(\tau)$ amount of renewable energy needs to be purchased from real-time spot renewable energy REC market at price $\beta(\tau)$ ($0 < \beta(\tau) \leq \beta_{max}$) in each fine-grained time slot if required.

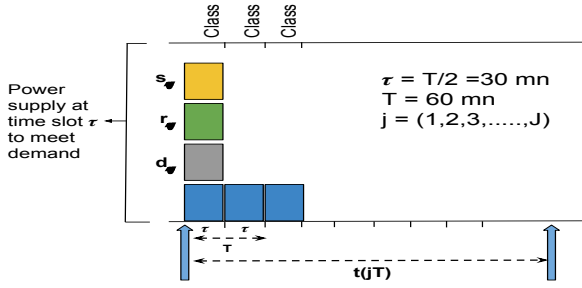


Fig. 4: Supply side characteristics

3.4 Virtual energy model

At each fine-grained time slot, workload arrives with the requirement of green energy percentage e.g., 30%, that needed to be served and we define the request process as $e(\tau)$. We assume, non renewable energy can be drawn anytime from Grid if there is deficit of renewable energy in spot renewable energy market. Discussing about energy drawing from Grid market is out of the scope of this paper. Considering the demand and supply side, the ideal condition would be meeting exact demand from the supply side : $U(\tau) + s(\tau) = e(\tau)$ or $U(\tau) + s(\tau) \geq e(\tau)$ suggesting, supply might exceed the actual green energy demand in some slots. In our model, this superfluous energy will neither be stored in a storage nor be wasted, rather will be used as virtual energy in the data center. This additional energy will increase the percentage of green energy of the total energy. Hence, we characterize $b(t)$ as the summation of all available green energy $[U(t) + s(t)]$ present in the slot and define superfluous or virtual energy $v(\tau)$ as:

$$v(\tau) = \begin{cases} v & \text{if } b(\tau) \geq thr_{max} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

3.5 Cost Minimization Problem of Spot Energy

As described in previous sub section, we consider, green energy demand, available green energy purchased from day-ahead renewable energy market and on-site plant, market price of spot energy from a vector $(e(\tau), U(\tau), \beta(\tau))$ and $e(\tau), \beta(\tau)$ is *i.i.d* over slots with some unknown probability distribution. Furthermore, $U(\tau)$, $e(\tau)$ and $\beta(\tau)$ are deterministically bounded by finite constraints U_{max} , e_{max} and β_{max} , so that: $0 \leq U(\tau) < U_{max}$, $0 \leq e(\tau) \leq e_{max}$, and $0 < \beta(\tau) \leq \beta_{max}$, $\forall t$

Now letting $Q(t)$ represent the total green energy request in the queue on slot t , we will have following update equation,

$$Q(t+1) = \max[Q(t) - U(t) - s(t), 0] + e(t) \quad (2)$$

Here $s(t)$ is a decision variable chosen in every slot t to stabilize the $Q(t)$ depending on the current state of the queue and vector $(e(t), U(t), \beta(t))$. We define an upper bound s_{max} for $s(t)$ as $0 \leq s(t) \leq s_{max}$. Hence, our objective is to design a flexible and robust control policy for time varying systems to formalize the stochastic cost optimization problem for spot energy is mentioned below:

$$\text{minimize } Cost_{av} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\beta(\tau)s(\tau)\} \quad (3)$$

$$\text{subject to } \bar{Q} < \infty, \quad (4)$$

$$0 \leq s(\tau) \leq s_{max}, \quad \forall \tau \quad (5)$$

where, \bar{Q} is the time average expected queue backlog, defined as:

$$\bar{Q} = \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{Q(\tau)\}$$

Since the virtual energy can not be present in every slot, the current control decision is coupled with the future decisions. As example, the deficit of green energy in some slots may be larger and hence IaaS provider has to pay penalty to SaaS provider or to the end client. We preferred Lyapunov optimization over dynamic programming to solve this optimization problem as dynamic programming requires significant statistics of demand and supply probabilities [23]. In contrast, Lyapunov framework has been proven to be efficient to design control algorithms for before mentioned constrained optimization without requiring a priori knowledge of demand and cost statistics.

We define an energy degraded aware virtual queue $X(t)$ and energy inadequacy aware virtual queue $Y(t)$ to measure the backlog of energy deficits in the queue by tracking the number of slots when energy deficiency and virtual energy is present. Moreover, for $X(t)$, having energy deficiency and presence of virtual energy can not occur simultaneously in a slot. We propose two threshold parameters thr_{min} and thr_{max} , where $thr_{min} < e(t)$ and $e(t) < thr_{max}$. For instance, in our framework, thr_{min} and thr_{max} are 25% and 30% respectively. The idea behind introducing the threshold parameters is to analyze whether any of the two events have occurred or not in the slot. When the supply side has lesser amount of energy than thr_{min} value to meet green energy demand in a slot, the slot is considered as an energy deficit slot. Hence, we characterize $b(t)$ such a way that, it can be either in between the value of thr_{min} and $thr_{min} - fuzzinessValue$ or lower than $thr_{min} - fuzzinessValue$ or above the value of thr_{min} and thr_{max} . Furthermore, in case of energy inadequacy aware virtual queue, the maximum allowable slots when the value of $b(t)$ lies on $b(t) < (thr_{min} - fuzzinessValue)$, can be defined as N_{max} . Therefore, the functionality of $b(t)$ will depend on thr_{min} , thr_{max} and $fuzzinessValue$ which is constructed as:

$$f(b(t)) = \begin{cases} surplus(v(t)) & \text{when } b(t) \geq thr_{max} \\ ideal & \text{when } thr_{min} \leq (b(t) \leq thr_{max}) \\ degraded & \text{if } (thr_{min} - b(t)) \leq fuzzinessValue \\ inadequate & \text{otherwise} \end{cases} \quad (6)$$

So, the functionality of $b(t)$ indicates that when there is excessive green energy available in a slot that demanded energy, we characterize those superfluous energy slots as surplus energy slots. Furthermore, the update equation for energy degradation and energy adequacy aware virtual queue will be:

$$X(t+1) = \max[X(t) - \gamma V(t), 0] + \gamma b(t) \quad (7)$$

$$Y(t+1) = \max[Y(t) - N_{max}, 0] + \gamma b(t) \quad (8)$$

where, γ is a counter, which adds values to corresponding parameter e.g., $v(t), b(t)$ whenever it is present in the queue. So, γ can be represent as $\gamma\{0, 1\}$. In other sense, $\gamma V(t+1) = [\gamma V(t) + (V(t) > 0)]$ and $\gamma b(t+1) = [\gamma b(t) + (b(t) > 0)]$, which explains broadly the construction of equation (7) and equation (8).

3.6 Lyapunov Optimization

We define, $\Theta(t) = [Q(t), X(t)]$ as the concatenated vector of actual and virtual queues. Moreover, the quadratic Lyapunov function is $L(\Theta(t)) = \frac{1}{2}[Q^2(t) + X^2(t) + Y^2(t)]$. So, the T slot conditional Lyapunov drift is interpreted as:

$$\Delta(\Theta(t)) = \mathbb{E}[L(\Theta(t+T)) - L(\Theta(t)) | \Theta(t)] \quad (9)$$

Following the Lyapunov framework of drift-plus-penalty algorithm [37], our algorithm designed to observe the current queue states $Q(t), X(t), Y(t)$ and the current vector $(e(\tau), U(\tau), \beta(\tau))$ and to make a decision on $s(t)$ where $0 \leq s(t) \leq s_{max}$, to minimize an upper bound on the following expression in every T slots:

$$\Delta(\Theta(t)) + V \mathbb{E} \left\{ \sum_{\tau=t}^{t+T-1} \beta(\tau) s(\tau) \right\}$$

where, V is defined as a control variable as $V > 0$ which is chosen accordingly to IaaS providers goal to give different weights that affect operational cost and energy deficiency trade-off. A large deficit of energy can reduce the operational cost, but can have negative effects on green energy requirement in data center resulting high percentage of SLA violation contracted between IaaS and SaaS provider. So, our approach consider to minimize a weighted sum of drift and penalty.

Theorem 1 (drift-plus-penalty bound). Let $V > 0, T \geq 1$ and $t = jT, \tau \in [t, t+T-1]$. For any control policy that satisfies $0 \leq s(t) \leq s_{max}$ for all t and the demand backlog is $Q(t) < Q_{max}$, the drift-plus-penalty satisfies:

$$\begin{aligned} & \Delta(\Theta(t)) + V \mathbb{E} \left\{ \sum_{\tau=t}^{t+T-1} \beta(\tau) s(\tau) | \Theta(t) \right\} \\ & \leq BT + V \mathbb{E} \left\{ \sum_{\tau=t}^{t+T-1} \beta(\tau) s(\tau) | \Theta(t) \right\} \\ & + \mathbb{E} \left\{ \sum_{\tau=t}^{t+T-1} Q(\tau) [e(\tau) - U(\tau) - s(\tau)] | \Theta(t) \right\} \quad (10) \\ & + \mathbb{E} \left\{ \sum_{\tau=t}^{t+T-1} X(\tau) [\gamma b(\tau) - \gamma v(\tau)] | \Theta(t) \right\} \\ & + \mathbb{E} \left\{ \sum_{\tau=t}^{t+T-1} Y(\tau) [\gamma b(\tau) - N_{max}] | \Theta(t) \right\} \end{aligned}$$

where, B is a finite constant and compute the bound on above drift-plus-penalty expression which is defined as:

$$B = \frac{1}{2} \max[(b_{max} - V_{max})^2 + (b_{max} - N_{max})^2 + (U_{max} - s_{max})^2 + e_{max}^2]$$

3.6.1 Dynamic Algorithm

We minimize the right hand side of drift-plus-penalty at each fine-grained time slot $\tau \in [t, t+T-1]$ by observing queue statistics $Q(\tau), X(\tau), Y(\tau)$, green energy demand $e(\tau)$, renewable energy production $r(\tau)$, spot green energy price $\beta(\tau)$ and choosing $s(\tau)$ according to the following optimization:

$$\begin{aligned} \text{minimize} \quad & \sum_{\tau=t}^{t+T-1} s(\tau) [V\beta(\tau) - Q(\tau)] \\ & + \sum_{\tau=t}^{t+T-1} [(X(\tau) + Y(\tau))(\gamma b(\tau) - \gamma v(\tau))] \\ \text{subject to} \quad & 0 \leq s(\tau) \leq s_{max}, \quad \forall \tau \\ & \sum_{\tau=t}^{t+T-1} \gamma b(\tau) \leq N_{max} \end{aligned}$$

3.6.2 Algorithmic solution

The proposed Lyapunov framework runs Algorithm 1 in the background in a repetitive manner to ensure *Green SLA* according to the runtime context, namely the demand, the budget, the spot market and SLA. The Algorithm 1 triggers the purchasing method. More importantly, it indicates the real situation to purchase energy from spot green market. It gives an edge over only choosing V parameter by procuring energy only when it is necessary to fulfill the contracted SLA.

- *interval class==inadequate*: Line 2 of Algorithm 1 describes the *interval.class* for being at $((thr_{min} - b(t)) > fuzziness \text{ value})$ and if $(\gamma b(t) < N_{max})$, we update the slot as green energy inadequate slot. Otherwise, we purchase energy referring to line 6 by triggering Algorithm 2.
- *interval class==degraded*: Line 9 of Algorithm 1 indicates *interval.class* for being at $((thr_{min} - b(t)) \leq fuzziness \text{ value})$ and if any previous slot posses virtual energy $v(t)$, we nullify one degraded slot. Contrarily, line 17 triggers to Algorithm 2 for purchasing green energy if it is feasible.
- *interval class==ideal*: The *interval.class* in the line 20 indicates, if the value of $b(t)$ lies between thr_{min} and thr_{max} , no procurement is needed. But if previous degraded slots exceeded the threshold number (deg_{max}), we need to move to Algorithm 2 to purchase green energy. The number of degraded slots might be greater than the targeted deg_{max} , only if there is unavailability of green energy in the spot green market.

At the end, the algorithm updates the current intervals/slots status to either inadequate/degraded/ideal or surplus. We propose one method to purchase green energy from the spot green energy market named *buySpotMinCost* (see Algorithm 2). As the label suggests, the *buySpotMinCost* insists to minimize the cost by purchasing green energy to switch from one class to the next in the order (inadequate, degraded, ideal, surplus). In addition, the purchasing decision is based on the penalty model, cost and available

quantity of energy in the spot green energy market. It can be observed from Algorithm 3 that, we support only static penalty model in this work. We buy only if the available green energy in the sport market is able to switch to the demanded class. Since it is a linear time algorithm, the average-case complexity and worst-case time complexity is $O(n)$.

Algorithm 1: greenSla

Input: $interval.class, \beta(\tau), P_{spot}, Av_{spot}, N_{max}, ina_{curr}, deg_{curr}, idl_{curr}, sur_{curr}$
Output: $s(\tau), ina_{curr}, deg_{curr}, idl_{curr}, sur_{curr}$

```

1  $s(\tau) = 0;$ 
2 if  $interval.class == inadequate$  then
3   if  $ina_{curr} < N_{max}$  then
4      $ina_{curr} ++;$ 
5   else
6      $s(\tau) = buy(\beta(\tau), P_{spot}, Av_{spot});$ 
7      $interval.class = updateClass(s(\tau));$ 
8      $(ina_{curr}, deg_{curr}, idl_{curr}, sur_{curr}) = update(interval.class)$ 
9 else if  $interval.class == degraded$  then
10  if  $sur_{curr} > 0$  then
11     $idl_{curr} + = 2;$ 
12     $sur_{curr} --;$ 
13  else
14    if  $deg_{curr} < deg_{max}$  then
15       $deg_{curr} ++;$ 
16    else
17       $s(\tau) = buy(\beta(\tau), P_{spot}, Av_{spot});$ 
18       $interval.class = updateClass(s(\tau));$ 
19       $(ina_{curr}, deg_{curr}, idl_{curr}, sur_{curr}) = update(interval.class)$ 
20 else if  $interval.class == ideal$  then
21  if  $deg_{curr} > deg_{max}$  then
22     $s(\tau) = buy(\beta(\tau), P_{spot}, Av_{spot});$ 
23     $interval.class = updateClass(s(\tau));$ 
24     $(ina_{curr}, deg_{curr}, idl_{curr}, sur_{curr}) = update(interval.class)$ 
25  else
26     $idl_{curr} ++;$ 
27 else if  $interval.class == surplus$  then
28   $sur_{curr} ++;$ 
29 return  $s(\tau), ina_{curr}, deg_{curr}, idl_{curr}, sur_{curr}$ 

```

Algorithm 2: buySpotMinCost

Input: $interval.class, interval.val, \beta(\tau), P_{spot}, Av_{spot}, penaltyModel$
Output: $class, cost$

```

1 if  $interval.class == inadequate$  then
2   if  $deg_{curr} < deg_{max}$  then
3      $b = buySpot(Degraded, penaltyModel);$ 
4   else if  $deg_{curr} > deg_{max}$  then
5      $b = buySpot(Surplus, penaltyModel);$ 
6   else
7      $b = buySpot(Ideal, penaltyModel);$ 
8 else if  $interval.class == degraded$  then
9   if  $deg_{curr} > deg_{max}$  then
10     $b = buySpot(Surplus, penaltyModel);$ 
11   else
12     $b = buySpot(Ideal, penaltyModel);$ 
13 else if  $interval.class == ideal$  then
14   $b = buySpot(Surplus, penaltyModel);$ 
15  $class, cost = update(b);$ 
16 return  $class, cost$ 

```

Algorithm 3: buySpot

Input: $class, penaltyModel$
Output: buy

```

1  $buy = false;$ 
2 if  $penaltyModel == static$  then
3   if  $demandedClass(class).isPossible()$  then
4     if  $cost(class) < Penalty(class)$  then
5        $buy = true;$ 
6 return  $buy$ 

```

4 EVALUATION

This section presents the results obtained from some experiments. In order to evaluate the proposed approach, we first describe our experimental environment. Then, we present cost analysis, SLA validation and how penalty model can influence the purchase decision and reduction of total expenditure in results section. Furthermore, insights and critical analysis are presented in discussion section.

4.1 Experimental Testbed

4.1.1 Real world power and workload traces

For a datacenter, Power usage effectiveness (PUE) is defined as the ratio of the data centers total power consumption to the data centers power consumption at the computer servers [20]. Therefore, we consider a data center which has an average PUE of 1.77. Though some of the state-of-the-art techniques claim to have reduced this value closer to 1.20, still most of today's data center have higher PUE values than 1.7 [24]. Therefore, in the planning phase [4], for transforming CPU utilization to power consumption, we traced CPU utilization for 7 days of 30 servers from PlanetLab [21] where CPU utilization has been traced for 500 different servers from across the world. As, building precise analytical models for modeling power consumption by modern multi-core CPUs makes a complex research problem, instead of using an analytical model of power consumption by a server, we utilize real data on power consumption provided by the results of the SPECpower benchmark.

We have selected two server configurations with multi-core CPUs published in November 2013: Dell Inc. PowerEdge M620 (Intel Xeon E5-2660 8coresX2.2 GHz,64 GB), and IBM NeXtScale nx360 M4 (2 Intel Xeon E5-2600,10 coresX2.2 GHz,256 GB). The configuration and power consumption characteristics of the selected servers are shown in Table 1. So, we model $total\ data\ center\ power(t) = Server\ power\ consumption(t) \times PUE\ value$. In addition, we use OpenForecast⁶ to forecast power demand for next 24 hours based on last 7 days power consumption which was traced. As our goal is to make data center 30% green, we scale down the power requirement demand to 30% and dynamically buy the required green energy from day-ahead REC market from multiple providers. Furthermore, power requirements were transformed to energy requirement (power integrated over time), as energy is purchasable in Grid and REC market but not the power (at which rate energy is transmitted).

6. <http://www.stevengould.org/software/openforecast/index.shtml>

TABLE 1: Power consumption by the selected servers at different load levels in Watt

Servers	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Dell Inc PowerEdge M620	688	1151	1322	1494	1671	1848	2061	2289	2499	2765	3239
IBM NeXtScale nx360 M4	550	873	999	1123	1251	1380	1525	1673	1887	2116	2404

We have presented 3 kinds of workload in Figure 9(a). The first workload is a real workload traced from Planetlab for 30 servers, which can be seen as interactive jobs (e.g., airline booking, e-commerce site), the second workload is more characterized as an on/off pattern (e.g., scientific application or batch jobs for same modeled server). However, the third workload is created with greater forecasting error statistics comparing to our predicted power workload in the planning phase by OpenForecast to evaluate how we can still propose a solution to fulfill green energy requirement based on *Green SLA*. The predicted green workload and above mentioned workload's characteristics is presented in Table 2. Furthermore, we take advantage of the local solar irradiation data to calculate the amount of on-site green energy presented in the Figure 9(b). As the spot market data is not available, we produce synthetic data to validate our experiment presented in Figure 9(c). There could be multiple energy consumers who might need green energy from the spot market, hence all the energy present in the spot market will not be available for a single consumer, which makes a realistic assumption.

TABLE 2: Workload characteristics

Workload	Mean	Standard deviation	Variance
Predicted Green workload	17.75	0.93	0.86
Experimental workload 1	20.26	2.98	8.89
Experimental workload 2	19.15	3.47	12.05
Experimental workload 3	23.95	4.24	18.01

4.1.2 Cost function and algorithms for comparison

Finding market prices of each kWh produced by green sources are extremely difficult as most of the today's wind or solar power infrastructure or plants receive enormous incentives either from government or different policy making organizations. Hence, to model a realistic price for energy in the day-ahead REC market, we investigate information of cap-ex, op-ex, levelized cost, fixed O&M cost, variable O&M cost of different sources of energy (e.g., Nuclear, Wind, Solar, Hydro etc)⁷ and find that the ratio of energy consumption cost between nuclear/brown and green energy is 1:1.68 approximately [4]. Since renewable sources are intermittent in nature, we consider the price of green energy sold at REC market will be in the range of 0.19-0.25 cents/kWh, which is 31.57% in price variation, while the price of Nuclear or mixed energy provided by EDF⁸ is 0.14 cents/kWh. As prices tend to be higher in the spot market, we have made an assumption that green spot market price can be 30-35% higher than the normal or day-ahead REC market. We compare our *greenSLA* algorithm with purchase at deadline approach and an energy storage approach that stores excessive on-site renewable and other abundant energy. Recent empirical

studies shows that, the charging/discharging efficiency of a storage is $\eta = 80\%$ and cost per cycle is approximately 0.1 euro [35]. In addition, we use fixed penalty value 1.5 euro/interval for IaaS provider if *Green SLA* is violated. We will analyze why do we choose this value and how it affects to the total energy cost in the result section. So, we define total energy cost as C_g, C_p, C_s respectively for *greenSLA*, "purchase at deadline" and "using energy storage" as follows:

- 1) C_g = grid energy cost + REC energy cost + spot green energy cost + penalty if violated SLA.
- 2) C_p = grid energy cost + REC energy cost + spot green energy cost
- 3) C_s = grid energy cost + REC energy cost + spot green energy cost + storage charging and discharging cost.

4.2 Results

In this section we describe cost analysis, SLA evaluation based on our proposed *greenSLA* algorithm, impact of control parameter V , impact of penalty to the total energy expenditure and robustness of our approach in detail.

4.2.1 Cost analysis

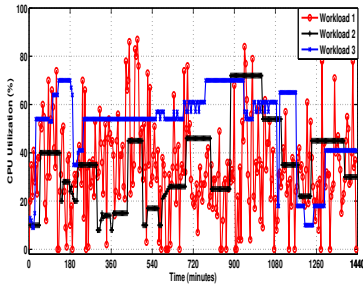
In Figure 6, we compare the cost for 24 hours between our proposed *greenSLA* algorithm and other two approaches namely "purchase at deadline" and "using energy storage". It is rationale that, purchasing green energy in every slot when it is needed incurs higher cost for purchase at deadline approach. Furthermore, the storage incurs additional costs due to charging the excessive energy and discharging the remaining energy in some slots. Besides, the storage does not discharge the exact amount of energy that has been charged earlier. From Figure 6(a) and 6(b), we calculate that, *greenSLA* achieves 4.47% and 4.17% cost reduction for workload 1 and 3.29% and 2.1% for workload 2 comparing to other two approaches. For workload 3, it has been noticed that (shown in Figure 6(c)), some time slots experience greater green energy inadequacy, hence *greenSLA* algorithm was forced to choose penalty for few slots. In some other slots, *greenSLA* choose penalty over buying green energy from spot market, as the cost for buying green energy was slightly higher in terms of total expenditure. Since purchasing green energy option is limited in spot market for a single consumer, other two approaches cannot meet the exact demand. Nevertheless, *greenSLA* performs better by reducing 5.9% and 3.54% cost comparing to other approaches for workload 3. In terms of buying spot energy, Figure 7(c) shows, purchase at deadline approach incurs 8.17%, 6.26%, 15.62% expenditure of total energy cost for workload 1, 2 and 3, the storage performs better by incurring 7.48%, 4.35% and 12.45% for respective workloads. In contrast, *greenSLA* significantly reduce the expenditure for spot green energy by only incurring .65%, .71% and 4.75%

7. http://www.eia.gov/forecasts/aeo/pdf/electricity_generation.pdf

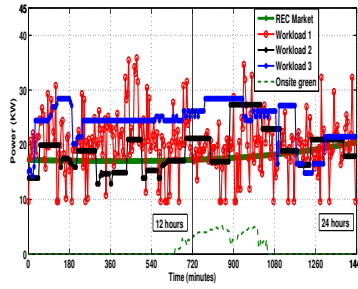
8. <http://entreprises.edf.com/entreprises-45638.html>

TABLE 3: SLA between IaaS provider and its consumers

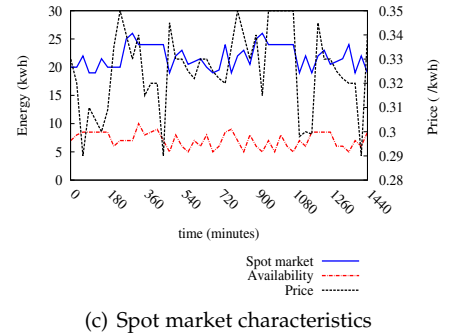
service	metric	oper.	$thr_{min}(\%)$	$thr_{max}(\%)$	fuzz.	% of fuzz.	conf.	penalty (euro/interval)
energy	green (Gr)	\geq	25	30	5	91.66	18.18	1.5



(a) Workload

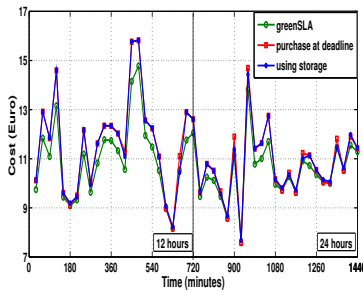


(b) Power Workload

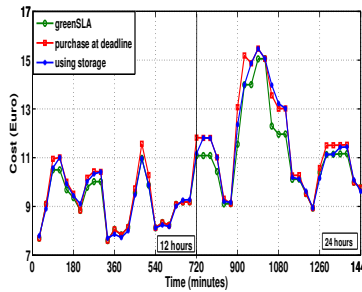


(c) Spot market characteristics

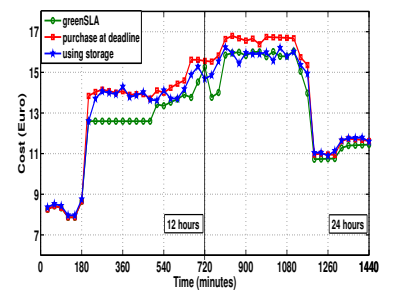
Fig. 5: Experimental Testbed



(a) Cost analysis for workload 1

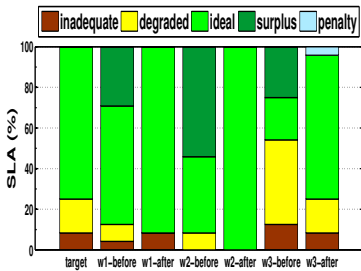


(b) Cost analysis for workload 2

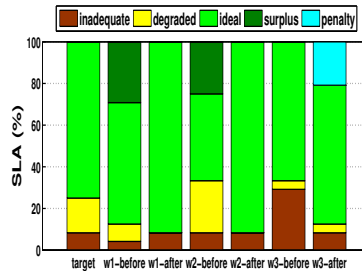


(c) Cost analysis for workload 3

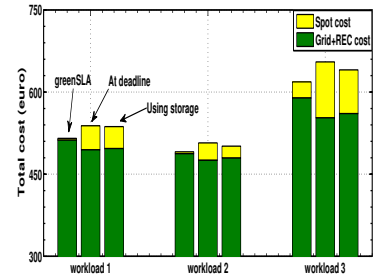
Fig. 6: Cost Analysis



(a) SLA validation for 12 hours

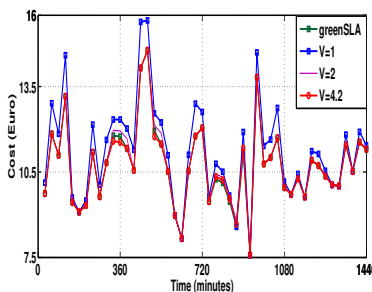


(b) SLA validation next 12 hours

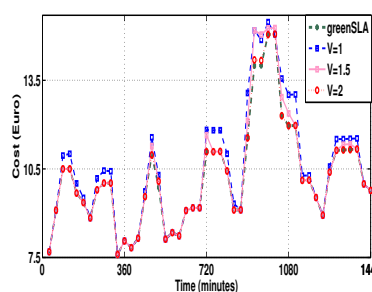


(c) Energy Cost

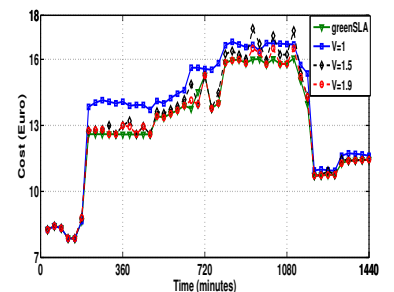
Fig. 7: SLA Validation and Energy Cost



(a) V-values for Workload 1



(b) V-values for Workload 2



(c) V-values for Workload 3

Fig. 8: Impact of parameter V

of total expenditure for above mentioned workloads. In our understanding, the concept of virtualizing the green energy leverages the process of reduction the total green energy expenditure by our algorithm than other two approaches.

4.2.2 SLA validation

Figure 7(a) and 7(b) show the comparison of targeted and achieved SLA based on *greenSLA* algorithm, which is evaluated and validated through CSLA. For our experiment, we fix observation window $\tau = 30$ minutes and evaluate every 12 hours as evaluation window suggesting, we evaluate 24 intervals at a time. We present a example of CSLA in section 3.2 having a SLA contract to provide 18 ideal intervals, 4 degraded intervals and 2 inadequate intervals out of 24 intervals or slots of green energy. The Table 3 summarizes the SLA between the IaaS provider and its consumers (SaaS providers). When SaaS provider establishes a SLA with IaaS provider for having 30% green energy to run their workload, the Ideal intervals ranges from 25% to 30% for green energy in CSLA framework. We argue on the fact that, it is not possible to provide or measure exactly 30% green energy in each timeframe. Besides, running same workload over and over again in a server shows slightly different power consumption profile. For workload 1, *greenSLA* achieves exactly the target for first 12 hours, but shows better performance attaining 91.67% of ideal and 0% of degraded interval comparing to the goal of providing 75% and 16.66% of ideal and degraded interval respectively. The algorithm performs even better for batch jobs type workload 2, by providing 91.67% and 100% of ideal interval for first 12 hours and later 12 hours. Although the mean green energy demand for workload 1 and workload 2 deviates by 14.14% and 7.88% comparing to our predicted demand, *greenSLA* still managed to fulfill SLA by greater percentage, thus our algorithm is robust to inaccurate prediction information in terms of SLA validation. Due to the insufficient amount of green energy in the green spot market, *greenSLA* fails by 4.17% and 8.33% to meet SLA for workload 3 in two timeframe but still managed to incur lower cost than other two approaches, even though the algorithm chose to provide penalties in 6 intervals.

4.2.3 Impact of control parameter V

As shown in Figure 8, to simulate 3 kinds of workload, we fix t to be 24 hours and each fine grained timeslot as $\tau=30$ minutes. We conduct experiments with different V values ranging from 1 to 5 and realized that, as the V value increases, it reduces the total energy cost. However, fixing larger value of V can violate contracted SLA. We see that, for workload 1, when $V = 4.2$, the control parameter performs well by reducing cost close to *greenSLA* and can satisfy the targeted SLA shown in 8(a). Nevertheless, if the value is increased by fraction, reduction of cost becomes larger but violates SLA. So this quantitatively indicates that, our proposed Lyapunov framework can approach very close to *greenSLA* within a diminishing gap of $\mathcal{O}(1/V)$. Moreover we perceive that, the value of V can not be fixed ahead since it depends on the characteristics of the workload and SLA parameters. Figure 8(b) and 8(c) shows that, the same value of V can incur different level of costs. So, choosing the appropriate value is essential to make a trade-off between cost

reduction and maintaining *Green SLA*. From our experiment, we find that the value of $V = 2$ and $V = 1.9$ can incur costs near to *greenSLA* for workload 2 and workload 3.

4.2.4 Impact of penalty

For our experiment, we choose static penalty per interval meaning, if IaaS provider fails to meet the demand of providing green energy beyond the SLA, the provider pay a fixed amount per interval. It is well understandable that, choosing a penalty value is extremely difficult and it depends on the business model of IaaS provider. Though we fixed the value of penalty as 1.5 euro/interval, Figure 9(c) shows that how the other penalty value affects the total expenditure for workload 3. In case of workload 1 and 2, *greenSLA* does not incur any penalty as it achieves the targeted SLA. We consider penalty value ranges from 1 euro/interval to 1.75 euro/interval by a factor of 0.25. We realize that, even increasing the penalty value by 0.25 euro/interval, the total expenditure increases only by 0.48%, 0.93%, 1.05% for 1.25, 1.50, 1.75 euro/interval. Hence, we choose 1.5 euro/interval penalty value which affects below 1% to the total green energy expenditure and certainly gives the flexibility to the IaaS provider.

4.3 Discussion

In this paper, we provide a *greenSLA* approach to show how SLA can be established and validated based on Green energy. Due to the intermittent nature of green sources and price diversity in electricity market, we consider that, the REC and Spot market energy price variation fluctuates in a range in section 4.1.2. We observe that, with 31.57% (.19-.25 cents/kwh) price variation in REC market; the maximum cost difference fluctuates within [-4.13%, 6.63%] for 3 kinds of workload, whose mean and standard deviation (see Table 2) varies significantly than the predicted workload. Figure 9(a) shows the total cost curve in respect to different REC market energy price and energy demand variation. So, with the increase of energy price in REC market, the total expenditure increases slightly but expenditure can be increased significantly if the energy demand variation is large. The rationale is that, the cost reduction through using *greenSLA* depends on the application workload. If the workload is more predictable, the cost reduction could be larger. Furthermore, Figure 9(b) indicates, the maximum cost difference fluctuates within [-1.43%, 1.30%] in respect to our proposed solution, while spot market has 25% (.28-.35 cents/kwh) price variation. Section 4.2.2 shows that, *greenSLA* managed to fulfill SLA with greater percentage, even workload 1 and 2 has significant deviation of mean in terms of green energy demand. Therefore, *greenSLA* is robust and reliable to the energy prices in two time-scale market and energy demand, even though they have certain turbulence in variation. Furthermore, we only present one method to buy green energy from the green spot market based on proposed *Green SLA*. Our idea can be easily extendable to other methods required by IaaS provider to meet different goals and establishment for different SLA based on availability of green energy. Furthermore, this paper only proposes static penalty in case of SLA violation, but dynamic penalty can be integrated into the model as CSLA supports dynamic penalty

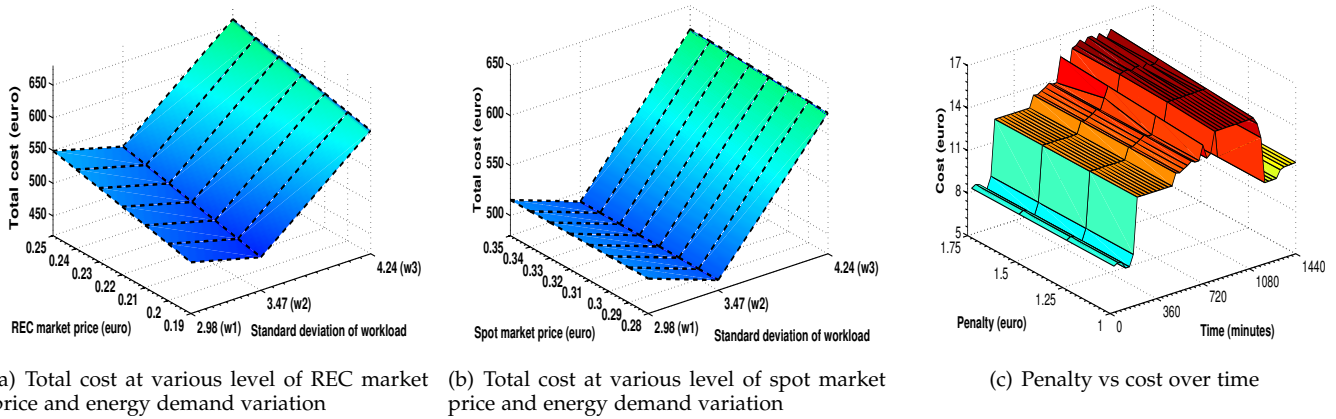


Fig. 9: Impact of different energy prices and penalty to total cost

modeling. Moreover, we do not propose an optimal solution as optimal solution for reducing cost of green energy can be varied depending on the workload pattern, on-site green energy generation and green spot market characteristics. From our experiments we observe that, it is possible to validate *Green SLA* with proposed spot market characteristics. Nonetheless, providing penalty in most of the intervals/slots when green energy is not available seems a little unrealistic. In case of low availability of green energy, we recommend that, efficient VM migration framework in IaaS level [36] and dynamic resource scaling and functionality degradation of services [3] can reduce energy consumption as well as green energy requirement in data center.

5 RELATED WORK

In recent years, a plethora of work has addressed the problem of reducing carbon footprint and energy procurement cost by integrating different renewable options in Cloud data centers. In the following, we first survey the literature on *Green SLA* and then focus on proposals on reducing energy procurement or management cost.

5.1 Green SLA

Laszewski et al. [13] investigated different green IT metrics that considered environmental impacts as part of the SLA for building GreenIT-as-a-Service (GaaS) which can be reused as part of a SaaS and IaaS framework. The framework provides how green metrics (i.e., DCiE, PUE, DCeP, SWaP) can be taken into consideration to reduce energy consumption in different layer of Cloud computing to provide greener services. Later, Wang et al [14] proposed green SLA based on energy aware scheduling of resources by exploiting dynamic voltage and frequency scaling (DVFS) technique. Their green SLA contract definition phase creates some green SLA templates where user might accept some performance or QoS degradation in order to reduce power consumption for their task completion. While these specifications are ought to build green services, *Green SLA* must be established using green energy sources to reduce carbon footprint. Klingert et al. [25] introduced the notion of *Green SLA*, focusing more on optimizing energy per job based

on known hardware and software techniques. Furthermore, how applications might specify preferences/requirements for these techniques were discussed. However, their work neither provides specification of SLA nor proposes validation process based on available green energy. Generally, *Green SLA* should be able to guarantee that the leased resources to SaaS providers or end clients via IaaS provider are environmental friendly. To this point, Haque et al. [1] considered an SLA that specifies the proportion of green power that the IaaS provider should use to run the job (e.g., $x\%$ of the job should run on green power). However, their proposed *Green SLA* approach is per application/job specific, where every job can have different green energy requirement. In contrast, we advocate the *Green SLA* as per interval specific, which is not dependent to the application/job. Moreover, their approach is more based on power distribution and control infrastructure. We try to ensure that, a data center can be proportionally green for the whole day by exploiting available green energy sources and markets for establishing a formal contract between SaaS provider/end client and we explained how the *Green SLA* can be formulated and validated through CSLA language. Ahmed et al. [26] addressed green SLAs for geographically distributed data center by shifting workloads towards most green energy efficient data center (following renewable) by penetrating VM migration. Their goal was to reduce carbon emission per job by finding most green data center among the virtual data center owned by the same provider while maximizing revenue. Again their work, including [27], lacks proper definition of *Green SLA* and its real use case in any business model.

While above researchers focused on addressing *Green SLA*, substantial amount of research has been done in coordinated multi-level power management [15], energy-efficient resource management [9][16], energy-efficient network-aware scheduling [19] and prominent green architecture [17] for Green data center. Recently, Basmadjian et al. [38] summarized the existing proposals and solutions that deal with the energy consumption and its reduction possibilities in every aspect of a Cloud data center. In position to above works, our research exploits incorporating green energy to establish *Green SLA*, which can be harmonious and advantageous to

existing trends and techniques used in modern data center to reduce carbon footprint.

5.2 Multi-source energy management and cost

Another stream of research has been found on multiple timescale energy market, prices and scheduling of energy in smart grid. Deng et al. [22] [29] studied optimal energy procurement to reduce the operational cost from long term ahead and real time market under intermittent renewable energy supplies with the leverage of UPS to reduce operational cost. Nair et al. [30] proposed similar goal using dynamic programming without taking into account the storage facility. On the contrary, we introduce spot energy market and investigate energy management of a data center by procuring energy depending on the *Green SLA* to reduce the operational cost of data center. We also introduce the concept of virtualization of energy, which take advantage over energy storage and shown to be efficient for small-scale data center. In addition, recent researches [31][32][33][34] show cost reduction in data center by distributing request process across multiple data centers by leveraging both time diversity and location diversity of electricity prices in grid environment. Contrarily, we study how to reduce cost depending on *Green SLA* aware energy management and procurement in single data center rather than how to distribute requests across data centers.

6 CONCLUSION

In this paper, we have presented a new scheme for green energy management based on *Green SLA*. Due to the intermittent nature of renewable sources, *Green SLA* was not introduced per interval specific based on green energy before. To overcome this difficulty, our proposal included three novel features: (i) the concept of virtualization of green energy ii) the extension of CSLA language that supports *Green SLA* and iii) the *greenSLA* algorithm, which leverages the concept of virtualization of green energy. Results demonstrate that, by smartly exploiting the renewable sources, *Green SLA* can be a possible reality in Cloud Computing. Throughout the paper we argue that, *Green SLA* should be established based in the presence of green energy, thus carbon footprint can be reduced and social responsibility for being *Green* can be achieved. Moreover, any large power consuming infrastructure similar to Cloud computing, which has large power demand variation over time and has aspiration to reduce carbon footprint by using green energy, can easily extend our proposed research work. We intend to continue this work in several ways. Firstly, we would like to define dynamic penalty models and pricing policies. Secondly, we continue to propose new strategies to purchase green energy from spot energy market. Finally, we expect to support negotiation phase to improve customer satisfaction levels.

ACKNOWLEDGEMENT

This work is supported by the EPOC project within the Labex CominLabs (<http://www.epoc.cominlabs.ueb.eu/>).

REFERENCES

- [1] M. Haque, K. Le, I. Goiri, R. Bianchini, and T. Nguyen, "Providing green slas in high performance computing clouds," Proc. International Green Computing Conference (IGCC), 2013.
- [2] C. Ren, D. Wang, B. Urgaonkar, and A. Sivasubramaniam, "Carbon-aware energy capacity planning for data centers," Proc. IEEE 20th International Symposium on Modeling, Analysis, Simulation of Computer and Telecommunication Systems (MASCOTS), Aug. 2012.
- [3] Y.Kouki, F. Alvares, S. Dupont and T.Ledoux, "A Language Support for Cloud Elasticity Management," Proc. IEEE/ACM CCGrid, Chicago, USA, 2014.
- [4] MS Hasan, Y. Kouki, T. Ledoux, JL. Pazat, "Cloud Energy Broker: Towards SLA-driven Green Energy Planning for IaaS Providers," Proc. IEEE International Conference on High Performance Computing and Communications (HPCC), Aug. 2014.
- [5] Y. Gao, Z. Zeng, X. Liu, and P. R. Kumar, "The Answer Is Blowing in the Wind: Analysis of Powering Internet Data Centers with Wind Energy," Proc. INFOCOM (Mini Conference), Apr. 2013.
- [6] I. Goiri, K. Le, T. Nguyen, J. Guitart, J. Torres, and R. Bianchini, "GreenHadoop: Leveraging Green Energy in Data-Processing Frameworks," Proc. EuroSys, Apr. 2012.
- [7] Z. Liu, M. Lin, A. Wierman, S. Low, and L. Andrew, "Geographical Load Balancing with Renewables," Proc. ACM GreenMetrics, Jun. 2011.
- [8] C. Stewart and K. Shen, "Some Joules Are More Precious Than Others: Managing Renewable Energy in the data center," Proc. Workshop on Power Aware Computing and Systems, 2009.
- [9] A. Beloglazov, J. Abawajy, and R. Buyya, Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing, Future Generation Computer Systems, vol. 28, no. 5, pp. 755768, May 2012.
- [10] A. Verma, Pmapper: Power and Migration Cost Aware Application Placement in Virtualized Systems, Proc. Ninth ACM/I-FIP/USENIX International Conf. Middleware (Middleware), 2008.
- [11] M.S. Hasan and E.N. Huh, "Heuristic based energy-aware resource allocation by dynamic consolidation of virtual machines in cloud data center," KSII Transactions on Internet and Information Systems, 7(8):18251842, 2013.
- [12] A. Rheddane, N. Palma, F. Boyer, F. Dumont, J. Menaud, A. Tchana, "Dynamic Scalability of a Consolidation Service," Proc. International Conference on Cloud Computing, Industry Track, Jun 2013, United States.
- [13] G. von Laszewski, L. Wang, "GreenIT Service Level Agreements." Proc. Service Level Agreements in Grids Workshop, colocated with IEEE/ACM Grid Conference, 2009, Banff, Canada.
- [14] Wang, Lizhe, Gregor Von Laszewski, Jai Dayal, and Fugang Wang, "Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS." Proc. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 368-377. 2010.
- [15] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, X. Zhu, "No "Power" Struggles: Coordinated Multi-level Power Management for the Data Center," Proc. ACM SIGARCH Computer Architecture News. 36, 4859, 2008.
- [16] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-Efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," Proc. 2010 International Conference on Parallel and Distributed Processing techniques and Applications (PDPTA 2010), Las Vegas, USA, July 12-15, 2010.
- [17] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, Y. Chen, GreenCloud: a new architecture for green datacenter, Proc. 6th ACM International conference industry session on Autonomic computing and communications industry session, pp. 2938, ICAC-INDST 09. 2009.
- [18] D. Kliazovich, P. Bouvry, S. U. Khan, "GreenCloud: a packet-level simulator of energy-aware cloud computing data centers," The Journal of Supercomputing 62.3: 1263-1283, 2012.
- [19] D. Kliazovich, P. Bouvry, and S. U. Khan, "Dens: Data center energy-efficient network-aware scheduling," Proc. IEEE/ACM International Conf on Green Computing and Communications (Greencom), China, 2010.
- [20] M. Ghamkhari and H. Mohsenian, "Energy and performance management of green data centers: A profit maximization approach," IEEE Transactions on Smart Grid, 4(2):10171025, June, 2013.

- [21] K. Park, V. S. Pai, "Comon: a mostly-scalable monitoring system for planetlab," Proc. SIGOPS Operating Systems Review, volume 40, pages 6574. ACM, 2006.
- [22] W. Deng, F. Liu, H. Jin, C. Wu, and X. Liu, "MultiGreen: Cost Minimizing Multi-source data center Power Supply with Online Control," Proc. ACM e-Energy, 2013.
- [23] M. J. Neely, A. S. Tehrani, A. G. Dimakis, "Efficient Algorithms for Renewable Energy Allocation to Delay Tolerant Consumers," Proc. 1st IEEE International Conference on Smart Grid Communication, 2010.
- [24] J. Yuventi, R. Mehdizadeh, "A critical analysis of power usage effectiveness and its use in communicating data center energy consumption," Energy and Buildings, 64(0):90-94, 2013.
- [25] C. Bunse, S. Klingert, T. Schulze, GreenSLAs: Supporting Energy-Efficiency through Contracts, Energy Efficient Data Centers," ser. Lecture Notes in Computer Science, vol. 7396, 2012.
- [26] A. Amokrane, F. Zhani, Q. Zhang, R. Langar, R. Boutaba, G. Pujolle, "On Satisfying Green SLAs in Distributed Clouds," Proc. IEEE/IFIP International Conference on Network and Service Management (CNSM), Nov, 2014.
- [27] A. Amokrane, Mr. Zhani, R. Langar, R. Boutaba, G. Pujolle, "Greenhead: Virtual Data Center Embedding Across Distributed Infrastructures," IEEE Transactions on Cloud Computing, vol. 1 (1) 2013.
- [28] P. X. Gao, A. R. Curtis, B. Wong, "Its not easy being green," ACM SIGCOMM Computer Communication Review, 2012.
- [29] W. Deng, F. Liu, H. Jin, C. Wu, "SmartDPSS: Cost-Minimizing Multi-source Power Supply for data centers with Arbitrary Demand," Proc. IEEE ICDCS, 2013.
- [30] S. Nair, S. Adlakha, A. Wierman, "Energy Procurement Strategies in the Presence of Intermittent Sources," Proc. ACM SIGMETRICS, p. 85-97, 2014.
- [31] Y. Guo, Z. Ding, Y. Fang, D. Wu, "Cutting Down Electricity Cost in Internet Data Centers by Using Energy Storage," Proc. IEEE GLOBECOM, Dec. 2011.
- [32] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-electricity-market Environment," Proc. IEEE INFOCOM, Mar. 2010.
- [33] Y. Yao, L. Huang, A. Sharma, L. Golubchik, M. Neely, "Data Centers Power Reduction: A Two Time Scale Approach for Delay Tolerant Workloads," Proc. IEEE INFOCOM, Mar. 2012.
- [34] Y. Zhang, Y. Wang, and X. Wang, "GreenWare: Greening Cloud-Scale Data Centers to Maximize the Use of Renewable Energy," Proc. ACM Middleware, Dec. 2011.
- [35] R. Uргаonkar, B. Uргаonkar, M. Neely, A. Sivasubramanian, "Optimal Power Cost Management Using Stored Energy in Data Centers," Proc. ACM SIGMETRICS, June, 2011.
- [36] F. Hermenier, X. Lorca, J.M. Menaud, G. Muller, J. Lawall, "Entropy: a consolidation manager for clusters," In: Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments (VEE), 2009
- [37] L. Georgiadis, M. Neely, L. Tassiulas, "Resource Allocation and Cross-layer Control in Wireless Networks." Now Pub, 2006.
- [38] R. Basmadjian, P. Bouvry, G. Costa, L. Gyarmati, D. Kliazovich, S. Lafond, L. Laurent, H. Meer, J.M. Pierson, R. Pries, J. Torres, T. Trinh, S. U. Khan, "Green data centers", Large-scale Distributed Systems and Energy Efficiency: A Holistic View. John Wiley & Sons. P 159-196, 2015



Md Sabbir Hasan received a Masters degree in Computer Science from Kyung Hee University, South Korea and B.SC in Electrical and Electronics Engineering from Islamic University of Technology, Bangladesh in 2013 and 2009 respectively. Currently, he is a PhD student at INSA de Rennes. He is a member of ASCOLA and MYRIADS INRIA teams. His research interest include Green SLA aware resource management in Cloud and energy-aware Cloud Computing.



across multiple laaS.

Yousri Kouki received a PhD degree in Computer Science from Ecole des Mines de Nantes in 2013. He is currently a Cloud Architect & Researcher at Linagora, Paris. His work and research interests include Cloud Computing, Service Oriented Computing, Autonomic computing and Green-IT. His professional contributions include a solution for Cloud Service Level Management. His present activities aim at designing and implementing an open PaaS solution that supports the Lifecycle of Cloud Applications



program committees.

Thomas Ledoux received a PhD from University of Nantes in 1998 in which his main contribution was the design of a reflective middleware named OpenCorba. He is a senior assistant professor at Ecole des Mines de Nantes and member of the INRIA/LINA Ascola team. Currently, he is investigating self-adaptive component-based applications for large-scale distributed systems and/or Green SLA for Cloud Computing. He heads several national projects for the Ascola team and serves on a number of conference



niques for High Performance Computing.

Jean-Louis Pazat received a PhD degree in Computer Science from University of Bordeaux (France) in 1989. He is currently Professor at the engineering school INSA de Rennes and is the scientific director of the Large Scale Systems Department at IRISA Rennes. He is also member of the MYRIADS Inria/IRISA research team. His research interests include middleware, dynamic adaptation techniques, energy efficiency and security in Clouds and distributed systems. He previously worked on parallelization tech-