The SAGITTA approach for optimizing solar energy consumption in distributed clouds with stochastic modeling

Benjamin Camus, Fanny Dufossé, Anne-Cécile Orgerie

The SAGITTA approach for optimizing solar energy consumption in distributed clouds with stochastic modeling

Benjamin Camus[1], Fanny Dufossé[2], and Anne-Cécile Orgerie[3]⋆

[1] Inria, IRISA, Rennes, France, benjamin.camus@inria.fr,
[2] Inria, CRIStAL, Lille, France, fanny.dufosse@inria.fr,
[3] CNRS, IRISA, Rennes, France, anne-cecile.orgerie@irisa.fr

Abstract. Facing the urgent need to decrease data centers' energy consumption, Cloud providers resort to on-site renewable energy production. Solar energy can thus be used to power data centers. Yet this energy production is intrinsically fluctuating over time and depending on the geographical location. In this paper, we propose a stochastic modeling for optimizing solar energy consumption in distributed clouds. Our approach, named SAGITTA (Stochastic Approach for Green consumption In disTributed daTA centers), is shown to produce a virtual machine scheduling close to the optimal algorithm in terms of energy savings and to outperform classical round-robin approaches over varying Cloud workloads and real solar energy generation traces.

Keywords: Data centers, distributed clouds, energy efficiency, renewable energy, scheduling, on/off techniques.

1 Introduction

The rapid increase of demand for Internet services leads Cloud providers to build more and more data centers for hosting these services. The data centers that constitute the Cloud infrastructures are usually geographically distributed for security reasons or to offer lower latency for their clients. This infrastructure increase comes with a dramatic growth of the power consumption globally drawn by data centers. As an example, in 2014, data centers in the U.S. consumed an estimated 70 billion kWh, representing about 1.8% of total U.S. electricity consumption [17].

To reduce this impact, Cloud providers resort to renewable energy sources which are either on-site or off-site [21]. Such energy sources are mostly intermittent by nature (wind, sun, etc.) with high variations, and periods of time without any production (during night for instance for photovoltaic panels). Energy storage devices can help to overcome this issue. But, they constitute a costly investment and they intrinsically lose part of the energy stored [22]. Thus, without storage, renewable energy has to be consumed upon production or it is

⋆ Corresponding author.

wasted. In this context, optimizing renewable energy consumption requires to know local availability for the distributed cloud infrastructure, in order to adequately allocate computing resources to incoming user requests. The goal is to geographically distribute the workload among the data centers so that, it fits at best the on-site renewable energy production that is variable and not known.

Here, we consider the problem of scheduling workload across multiple data centers for minimizing renewable energy loss. To solve this issue, we propose SAGITTA: a Stochastic Approach for Green consumption In disTributed daTA centers. SAGITTA uses a stochastic approach for estimating renewable energy production, and greedy heuristics for allocating resources to the incoming user requests and switching off unused servers. While SAGITTA was first introduced in [3], the original SAGITTA algorithm did not take into account the switching on and off energy costs. Here these costs are integrated in SAGITTA's algorithm and all the simulations have been redone. This chapter extends the first SAGITTA study [3] with the following contributions:

- modification of SAGITTA's original version to take into account the switching on and off energy costs;
- proof of local optimality of SAGITTA;
- proposition of an optimal algorithm based on dynamic programming to solve the problem;
- simulation results exploring the influence of green energy forecast and green energy production on SAGITTA and study of its scalability;
- performance comparison between SAGITTA and the optimal algorithm;
- study on the exactness of our green power production forecast.

Our simulation-based results show the efficiency of SAGITTA compared to classical allocation approaches. Indeed, compared to the optimal solution, SAGITTA consumes 5.2% more energy overall, while a classical round-robin solution consumes 12.9% more energy overall than optimum.

The remainder of the paper is structured as follows. Related work is presented in Section 2. A formal definition of the problem is given in Section 3. Section 4 details the SAGITTA approach. Section 5 exhibits an optimal algorithm for the considered problem. A simulation-based evaluation is conducted, simulation conditions are described in Section 6 and results are provided in Section 7. Future work is discussed in Section 8.

2 Related work

Cloud infrastructures consist in geographically distributed data centers which are linked through communication networks [23]. With the emergence of the Future Internet and the dawning of new IT models such as cloud computing, the usage of data centers, and consequently their power consumption, increases dramatically. As an example, for 2010, Google used 900,000 servers which consumed 1.9 billion kWh of electricity [11]. Other major Cloud companies present similar figures and similar issues [10].

Virtualization technology and its ability to pool resources through transparent sharing should have minimized worldwide data center consumption. But, the energy consumption of state-of-the-art servers grows inexorably as they embed more and more powerful cores and advanced features and technologies. Consequently, the global data center consumption keeps increasing rapidly [17]. This situation raises major environmental, economic and social concerns.

The first way to save energy at a data center level consists in locating it close to where the electricity is generated, hence minimizing transmission losses. For example, Western North Carolina, USA, attracts data centers with its low electricity prices due to abundant capacity of coal and nuclear power following the departure of the region's textile and furniture manufacturing [9]. This region has three super-size data centers from Google, Apple and Facebook with respective power demands of 60 to 100 MW, 100 MW and 40 MW [9].

Other companies opt for greener sources of energy. For example, Quincy (Washington, USA) supplies electricity to data facilities from Yahoo, Microsoft, Dell and Amazon with its low-cost hydro-electrics left behind following the shutting down of the region's aluminum industry [9]. Several renewable energy sources like wind power, solar energy, hydro-power, bio-energy, geothermal power and marine power can be considered to power up super-sized facilities.

In spite of these approaches, numerous data facilities have already been built and cannot be moved. Cloud infrastructures, on the other hand, can still take advantage of multiple locations to use green sources of energy with approaches such as follow-the-sun and follow-the-wind [8]. As sun and wind provide renewable sources of energy whose capacity fluctuates over time, the rationale is to place computing jobs on resources using renewable energy, and migrate jobs as renewable energy becomes available on resources in other locations. However, the migration cost, in terms of both energy and performance, may be prohibitive [2].

Within the data center itself, a range of technologies can be utilized to make cloud computing infrastructures more energy efficient, including better cooling technologies, temperature-aware scheduling [7], Dynamic Voltage and Frequency Scaling (DVFS) [18], and resource virtualization [19]. The use of Virtual Machines [1] brings several benefits including environment and performance isolation; improved resource utilization by enabling workload consolidation; and resource provisioning on demand. Nevertheless, such technologies should be analyzed and used carefully for actually improving the energy-efficiency of computing infrastructures [13].

One of the most efficient techniques for saving energy in not fully utilized data centers consists in shutting down unused resources as switched off resources consumes less power than idle ones [14]. The number of switched off resources can be increased by consolidation techniques [14]. However, switching on and off resources consumes time and energy and these costs need to be taken into account in order to effectively ensure energy savings [15].

Concerning green energy integration, Ren *et al.* have proposed an online scheduling algorithm which optimizes the energy cost and fairness among different data centers subject to queuing delay constraints [16]. While their work is based on a distributed Cloud model similar to ours, they aim at minimizing the cost of the consumed electricity, instead of the wasted renewable energy in our case. Tripathi *et al.* have presented a mixed integer linear programming formulation for capacity planning while minimizing the total cost of ownership [21]. Their model schedules demand considering the availability of green energy and its price variation to lower the total cost of ownership. Finally, a literature review of renewable energy integration in data centers can be found in [6].

3 Problem

First, the Cloud model and assumptions are described in Section 3.1 similarly to [3]. Then, Section 3.2 proposes the problem formulation.

3.1 Cloud model

We consider a distributed Cloud infrastructure comprising several data centers geographically distributed and powered by the regular electrical grid on one side and on-site photovoltaic panels (PV) on the other side. The user management of the Cloud is assumed to be centralized. This Cloud model is shown on Figure 1.

Incoming users requests can arrive at any time. Each request requires to be computed by a virtual machine (VM) located on any of the data centers. Each data center hosts a given amount of homogeneous servers.

3.2 Problem formulation

We consider a system of M data centers spread over a large area. A data center DC_i is characterized by its number S_i of servers. Servers are considered homogeneous over the different data centers, in term of computing capabilities and energy consumption.

As for the application model, we consider identical VMs submitted at unpredictable rate. The VMs are supposed to be executable in less than one time slot. We can thus describe both computing and memory requirement of VMs by the number C of VMs that a server can complete in a single time slot. We consider that a server consumes at full capacity a power of P_s.

The main difference with previous paper [3] concern the energy consumption model. In [3], the model only considered the processing cost, that is the consumption of servers during the processing time. The switching ON/OFF costs were only used to evaluate the performance of resulting algorithms. In this paper, we take into account the switching ON/OFF costs in the basis model. More precisely, we consider that the switches are done at beginning of the time slots, and that the switching time is negligible compared to the time slot duration. For example, the duration to turn on will not impact the number of VM executed in the remaining time. We compute in this paper the energy at the scale of the time slots, and we do not evaluate if the energy is used mostly at beginning or at the

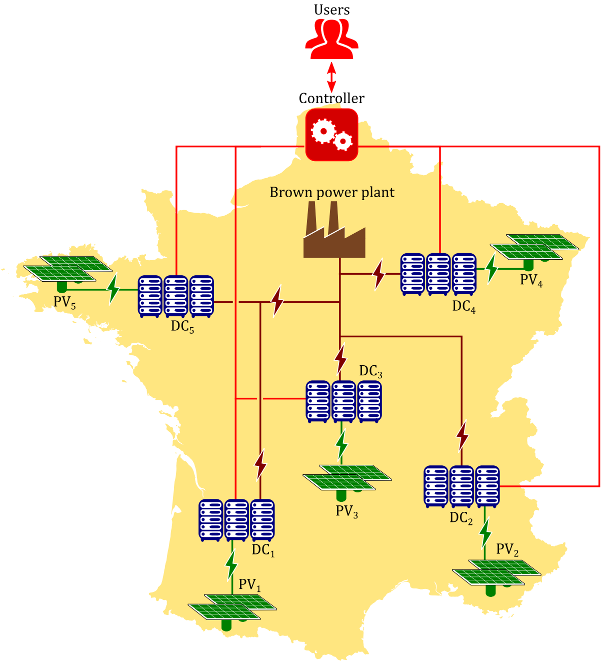

Fig. 1. Considered cloud model from [3].

end on a time slot. Therefore, in our model, the energy consumption, likewise the energy production, is smoothed on the time slot.

Consequently, the energy consumption of DC_i has three possible component. The first part is the power consumption of each server, and is proportional to the number of servers ON at current time slot t on DC_i, $U_i(t)$. The second part is the costs of switching on servers, proportional to the number of servers turned on $U_i^+(t) = \max(0, U_i(t) - U_i(t-1))$ and the last on the cost of shutting down servers, proportional to $U_i^-(t) = \max(0, U_i(t-1) - U_i(t))$. The total power consumed by the system is thus

$$\sum_{i=1}^{M} P_s \times U_i(t) + P_{ON} \times U_i^+(t) + P_{OFF} \times U_i^-(t).$$

This power requirement is to be compared with the green power produced at each data center. We model the green power available at time slot t in data center DC_i as a random variable $G_i(t)$ that follows a truncated normal distribution of mean $Eg_i(t)$ and standard deviation $p_i(t)$, with lower limit 0. Thus, the brown power consumed at time slot t in DC_i is equal to

$$max(0, P_s \times U_i(t) + P_{ON} \times U_i^+(t) + P_{OFF} \times U_i^-(t) - G_i(t)).$$

Our problem consists in allocating VMs to data centers, in order to minimize the consumption of brown energy. VMs are allocated by time slots. Then, our objective is to turn ON the adequate number of servers on the best locations for this criteria. We denote $N(t)$ the number of waiting VMs at time slot t. We thus need to have enough servers ON for all waiting VMs at time slot t: $\sum_{i=1}^{M} U_i(t) \geq N(t)/C$. All these notations are summarized in Table 1.

Table 1. Table of Notations

Notation	Definition
Constants	
M	Number of data centers
DC_i	Data center number i
S_i	Number of servers in data center i
C	Maximum number of VMs in parallel on a server
P_s	Maximum power consumption of a server
Variables	
$N(t)$	Number of incoming VMs for time slot t (input)
$U_i(t)$	Number of machines ON at current time slot on data center i (output)
$G_i(t)$	Random variable of the green power produced at time slot t
$Eg_i(t)$	Expected green power generation at data center i during time slot t (input)
$p_i(t)$	Standard deviation of green power generation on data center i (input)
w	Workload portion (number of VM): $0 < w \leq N(t)$ (input)
$Ec_i(P,t)$	Expected brown consumption of data center i with power P at time slot t

4 SAGITTA

In this section, we present our approach named SAGITTA: a Stochastic Approach for Green consumption In disTributed daTA centers. The details for computing the expected green and brown consumption are provided in Section 4.1. SAGITTA's algorithms are presented in Section 4.2 and its local optimality is demonstrated in Section 4.3.

4.1 Expected green and brown consumption

We now evaluate the expected brown power consumption of data center DC_i at time t, consuming power P, $Ec_i(P,t)$. We first evaluate the density function of the random variable of the green power generation of DC_i at time t $G_i(t)$.

Let X be a random variable following a normal distribution of parameters $Eg_i(t)$ and $p_i(t)$, density function

$$\phi(t) = \frac{1}{p_i(t)\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-Eg_i(t)}{p_i(t)}\right)^2}$$

and distributive function

$$\Phi(t) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{t - Eg_i(t)}{p_i(t)\sqrt{2}}\right)\right).$$

Then, for $x > 0$,

$$P(G_i(t) < x) = P(X < x | X > 0) = \frac{P(0 < X < x)}{P(X > 0)}$$

and the density function of $G_i(t)$ equals $\phi_i(t) = \frac{\phi(t)}{P(X>0)}$.

Let $B_i(t)$ be the random variable of the brown consumption of DC_i at time slot t. Then, as shown in [3], we have:

$$Ec_i(P,t) = (P - Eg_i(t))\frac{\Phi(P) - \Phi(0)}{1 - \Phi(0)} - p_i(t)^2\frac{\phi(0) - \phi(P)}{1 - \Phi(0)},$$

with $\phi(x) = \frac{1}{p_i(t)\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-Eg_i(t)}{p_i(t)}\right)^2}$ and $\Phi(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x-Eg_i(t)}{p_i(t)\sqrt{2}}\right)\right)$.

In the following, if the considered time slot is clear, we simply note $Ec_i(P)$ for the expected brown power consumption.

4.2 Algorithms description

Our SAGITTA approach uses several algorithms to take decisions and allocate VMs to physical servers. These algorithms are designed to determine at any time slot on which data center to turn ON and OFF servers. At each time slot, our constraint is to turn ON the minimum number of servers that allows for executing all requested VMs, that is $\lceil N(t)/C \rceil$.

Algorithm 1: General algorithm

```
if $\sum_{1 \leq i \leq M} U_i(t) < \lceil \frac{N(t)}{C} \rceil$ then
  Switch on decision;        (Algorithm 2)
else if $\sum_{1 \leq i \leq M} U_i(t) > \lceil \frac{N(t)}{C} \rceil$ then
  Switch off decision;       (Algorithm 3)
end if
Transfer decision;       (Algorithm 4)
for $1 \leq i \leq M$ do
  Let $U_i(t)$ servers on and fill them, switch off the rest;
end for
```

The general algorithm (Algorithm 1) is designed as follows. It first determines if the number of servers available is under or over the requested number. If there is not enough servers ON, Algorithm 2 determines the location of servers to switch on. If some servers are unnecessary, Algorithm 3 determines where servers should be shut down. These decisions are done regarding the expected green energy production in the different data centers. More precisely, Algorithm 2 compares the expected extra cost in brown energy consumption c_i induced by an additional server ON on any data centers, and selects the data center with minimum expected extra cost. The variable $U_i(t)$ is then incremented, but the servers are only switched on at end of Algorithm 1, when all decisions are taken on any data centers. The same way, Algorithm 3 selects one by one the servers to switch OFF to maximize the expected gain.

Algorithm 2: Switch on decision

```
for $1 \leq i \leq M$ do
    $P_i = U_i(t-1) \times P_s$
    $U_i(t) = U_i(t-1)$
    if $U_i(t) < S_i$ then
        Compute $c_i = Ec_i(P_i + P_{ON} + P_s) - Ec_i(P_i)$;
    else
        $c_i = +\infty$;
    end if
end for
while $\sum_{1 \leq i \leq M} U_i(t) < \lceil \frac{N(t)}{C} \rceil$ do
    Find $j$ such that $c_j = \min_{1 \leq i \leq M} c_i$;
    $U_j(t)++$;
    $P_j += P_{ON} + P_s$
    Recompute $c_j$;
end while
```

Finally, Algorithm 4 evaluates if the brown power consumption could be reduced by transferring the available processing power from one data center to another. More precisely, the algorithm determines some location where a fixed number of servers is turned off, and a new location where the same number of servers is turned on. One server is selected for switch OFF on the data center of maximum gain and another one to switch ON on the data center of minimum cost, if the gain on the first data center exceed the cost on the second one. The costs and gain are computing with respects to previous preallocation. For example, the cost of turning a server on is not the same if Algorithm 3 has decided to turn off a server at previous step, that is if $U_i(t) < U_i(t-1)$ or not. If no such decision was taken, the cost c_i is the same as in Algorithm 2, $c_i = Ec_i(P_i + P_{ON} + P_s) - Ec_i(P_i)$, that includes the cost to turn on and the cost of processing. In the other case, turning on a server correspond to cancel a decision to turn off a server, that is $c_i = Ec_i(P_i - P_{OFF} + P_s) - Ec_i(P_i)$. The same holds for gain g_i to turn off a server.

Algorithm 3: Switch off decision

```
for $1 \leq i \leq M$ do
    $P_i = U_i(t-1) \times P_s$
    if $U_i(t) > 0$ then
        Compute $g_i = Ec_i(P_i) - Ec_i(P_i + P_{OFF} - P_s)$;
    else
        $g_i = -1$;
    end if
end for
while $\sum_{1 \leq i \leq M} U_i(t) > \lceil \frac{N(t)}{C} \rceil$ do
    Find $j$ such that $g_j = \max_{1 \leq i \leq M} g_i$;
    $U_j(t) - -$;
    $P_j + = P_{OFF} - P_s$
    Recompute $g_j$;
end while
```

After running Algorithm 4, general Algorithm 1 applies all these decisions. The selected number of servers are turned ON and OFF and all VMs are allocated to available servers.

4.3 SAGITTA local optimality

We demonstrate in this section, that Algorithms 2, 3 and 4 are locally optimal. More precisely, at time slot t, with respect to the normal laws $(\mathcal{N}(Eg_i, p_i)^2))_{1 \leq i \leq M}$, these algorithms select the best servers to turn on or off, to minimize the expect brown power consumption. We first consider Algorithm 2.

Theorem 1. *Algorithm 2 is locally optimal with respect* $(\mathcal{N}(Eg_i, p_i)^2))_{1 \leq i \leq M}$.

Proof. We are studying the execution of SAGITTA during one unique time slot, so we do not precise t in this proof. For sake of simplicity, we denote in this proof $c_i(P) = Ec_i(P + P_{ON} + P_s) - Ec_i(P)$. We compare Algorithm 2 to any selection of servers to turn on. We do not compare to a configuration where some servers are turned on and others turned off.

We first prove that the function $c_i(P)$ is increasing, and then that this property induces the local optimality of Algorithm 2.

$$Ec_i(P) = P \times \frac{\Phi(P) - \Phi(0)}{1 - \Phi(0)} - \frac{\int_0^P x\phi(x)\mathrm{d}x}{1 - \Phi(0)}$$

Algorithm 4: Transfer decision

```
for $1 \leq i \leq M$ do
  if $U_i(t) > U_i(t-1)$ then
    $g_i = Ec_i(P_i) - Ec_i(P_i - P_s - P_{ON})$;
  else
    $g_i = Ec_i(P_i) - Ec_i(P_i + P_{OFF} - P_s)$;
  end if
  if $U_i(t) < U_i(t-1)$ then
    $c_i = Ec_i(P_i - P_{OFF} + P_s) - Ec_i(P_i)$;
  else
    $c_i = Ec_i(P_i + P_{ON} + P_s) - Ec_i(P_i)$;
  end if
end for
while $\max_{1 \leq i \leq M} g_i > \min_{1 \leq j \leq M} c_j$ do
  Find $k$ such that $g_k = \max_{1 \leq i \leq M} g_i$;
  Find $l$ such that $c_l = \min_{1 \leq j \leq M} c_j$;
  $U_k(t) - -$;
  $U_l(t) + +$;
  Recompute $g_k$ and $P_k$;
  Recompute $c_l$ and $P_l$;
end while
```

with $\phi(x) = \frac{1}{p_i\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-Eg_i}{p_i}\right)^2}$ and $\Phi(x) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{x-Eg_i(t)}{p_i(t)\sqrt{2}}\right)\right)$. Let first evaluate the derivative function of $Ec_i(P)$.

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}P}Ec_i(P) &= \frac{\Phi(P)-\Phi(0)}{1-\Phi(0)} + P \times \frac{\frac{\mathrm{d}}{\mathrm{d}P}\Phi(P)}{1-\Phi(0)} - \frac{\mathrm{d}}{\mathrm{d}P}\frac{\int_0^P x\phi(x)\mathrm{d}x}{1-\Phi(0)} \\
&= \frac{\Phi(P)-\Phi(0)}{1-\Phi(0)} + P \times \frac{\frac{\mathrm{d}}{\mathrm{d}P}\Phi(P)}{1-\Phi(0)} - \frac{\mathrm{d}}{\mathrm{d}P}\frac{\int_0^P x\phi(x)\mathrm{d}x}{1-\Phi(0)} \\
&= \frac{\Phi(P)-\Phi(0)}{1-\Phi(0)} + P \times \frac{\phi(P)}{1-\Phi(0)} - \frac{P\phi(P)}{1-\Phi(0)} \\
&= \frac{\Phi(P)-\Phi(0)}{1-\Phi(0)}
\end{aligned}$$

Then,

$$\frac{\mathrm{d}}{\mathrm{d}P}c_i(P) = \frac{\Phi(P + P_{ON} + P_s) - \Phi(P)}{1 - \Phi(0)}.$$

The function Φ is strictly increasing, as it is the cumulative distributive function of law $\mathcal{N}(Eg_i, p_i)^2)$. Then, the derivative of function $c_i(U)$ is strictly positive, and $c_i(U)$ is strictly increasing. Intuitively, the more servers are turned on on a data center, the higher is the ratio of brown power consumption of this data center.For sake of simplicity, we denote $P_i(U)$, the power consumption corresponding to U servers on on data center DC_i. Function P_i is clearly strictly increasing.

We demonstrate now that the local optimality of Algorithm 2 can be deduced from this property. We consider here the location of servers to turn on, and not the possible transfers. Let $(U_i^{opt})_{1\leq i\leq M}$ be the optimal choice for total brown power production, and $(U_i^{alg})_{1\leq i\leq M}$ the decision of Algorithm 2.

First denote that by hypothesis, we have $\sum_{1\leq i\leq M} U_i^{opt} = \sum_{1\leq i\leq M} U_i^{alg}$. Suppose that for some i and j, $U_i^{opt} > U_i^{alg}$ and $U_j^{opt} < U_j^{alg}$.

Consider the step of Algorithm 2 at which the last server was decided to be turned on on DC_j. Let U_i be the number of servers this algorithm had decided to turn on on DC_i at this step. By definition of Algorithm 2, $c_j(P_j(U_j^{alg}-1)) \leq c_i(P_i(U_i))$. As proven earlier, $c_i(P_i(U_i)) \leq c_i(P_i(U_i^{alg})) \leq c_i(P_i(U_i^{opt}-1))$. As $(U_i^{opt})_{1\leq i\leq M}$ is optimal, then $c_i(P_i(U_i)) = c_i(P_i(U_i^{alg})) = c_i(P_i(U_i^{opt}-1))$ and as c_i and P_i are strictly increasing, $U_i^{alg} = U_i^{opt}-1$. Thus, $c_j(P_j(U_j^{alg}-1)) = c_i(P_i(U_i^{opt}-1))$ and so, the decision to turn on a last server on DC_i or DC_j does not impact the expected brown power consumption.

As this property holds for all possible differences between $(U_i^{opt})_{1\leq i\leq M}$ and $(U_i^{alg})_{1\leq i\leq M}$, we can conclude that both selections have the same expected brown power consumption and that $(U_i^{alg})_{1\leq i\leq M}$ is optimal.

□

Theorem 2. *Algorithm 3 is locally optimal with respect* $(\mathcal{N}(Eg_i,p_i)^2))_{1\leq i\leq M}$.

Proof. This demonstration is very similar to the previous one, so we use similar intermediate results. Let $g_i(P) = Ec_i(P) - Ec_i(P+P_{OFF}-P_s)$. Then, $\frac{\mathrm{d}}{\mathrm{d}P}g_i(P) = \frac{\Phi(P)-\Phi(P+P_{OFF}-P_s)}{1-\Phi(0)}$, $g_i(P)$ is strictly increasing.

Now, let $(U_i^{opt})_{1\leq i\leq M}$ be the optimal choice for total brown power production, and $(U_i^{alg})_{1\leq i\leq M}$ the decision of Algorithm 3. Suppose that for some i and j, $U_i^{opt} > U_i^{alg}$ and $U_j^{opt} < U_j^{alg}$. As in previous proof, we denote $P_i(U)$ the power consumption on data center DC_i corresponding to U servers on.

Consider the step of Algorithm 3 at which the last server was decided to be turned off on DC_i. Let U_j be the number of servers the algorithm had decided to turn on on DC_j at this step. By definition of Algorithm 3, $g_j(P_j(U_j^{alg}+1)) \geq g_i(P_i(U_i))$. We have $g_i(P_i(U_i)) \geq g_i(P_i(U_i^{alg})) \geq g_i(P_i(U_i^{opt}+1))$. By optimality, we obtain $g_i(P_i(U_i)) = g_i(P_i(U_i^{alg})) = g_i(P_i(U_i^{opt}+1))$. As $g_i(P_i(P))$ is strictly increasing, $U_i^{alg} = U_i^{opt}+1$ and the decision to turn off a last server on DC_i or DC_j does not impact the expected brown power consumption.

As in the previous proof, we obtain that $(U_i^{alg})_{1\leq i\leq M}$ is optimal. □

Theorem 3. *Algorithm 4 is locally optimal with respect* $(\mathcal{N}(Eg_i,p_i)^2))_{1\leq i\leq M}$.

Proof. Let $(U_i^{opt})_{1\leq i\leq M}$ be the optimal choice for total brown power production, and $(U_i^{alg})_{1\leq i\leq M}$ the decision of Algorithm 4. Suppose that for some i and j, $U_i^{opt} > U_i^{alg}$ and $U_j^{opt} < U_j^{alg}$. As previously, $P_i(U)$ is the power consumption on data center DC_i for U servers on.

By definition of Algorithm 4, $c_i(P_i(U_i^{alg})) \geq g_j(P_j(U_j^{alg}))$. Moreover, we know $g_i(P_i(U_i^{opt})) = c_i(P_i(U_i^{opt}-1)) \geq c_i(P_i(U_i^{alg}))$ and $g_j(P_j(U_j^{opt}+1)) \leq$

$g_j(P_j(U_j^{alg}))$. Thus, $g_i(P_i(U_i^{opt})) \geq c_j(P_j(U_j^{opt}))$, $U_i^{opt} - 1 = U_i^{alg}$ and $U_j^{opt} + 1 = U_j^{alg}$. The decision to turn on a last server on DC_i and DC_j does not impact the expected brown power consumption.

By induction, we obtain that $(U_i^{alg})_{1 \leq i \leq M}$ is optimal.

□

We have proven that our algorithms are locally optimal, it means that there are optimal only regarding the current time slot. It is clear that the only point in the model that impacts this locality is the switching ON/OFF costs. Without these costs, the power consumption on current time slot does not depend on previous time slot. Thus, we obtain the following theorems.

Theorem 4. *Algorithm 2 is optimal without switching ON/OFF costs, with respect* $(\mathcal{N}(Eg_i, p_i)^2))_{1 \leq i \leq M}$.

Theorem 5. *Algorithm 3 is optimal without switching ON/OFF costs, with respect* $(\mathcal{N}(Eg_i, p_i)^2))_{1 \leq i \leq M}$.

Theorem 6. *Algorithm 4 is optimal without switching ON/OFF costs, with respect* $(\mathcal{N}(Eg_i, p_i)^2))_{1 \leq i \leq M}$.

5 Computing the optimal

We propose in this section an optimal algorithm that allows to compute the minimal power consumption for the whole experiment, knowing the workload trace and the green power production. This algorithm is then employed to evaluate the performance of our algorithms in the simulations. Obviously it cannot be used to allocate VMs, as it is based on complete data knowledge.

This algorithm (Algorithm 5) is a dynamic programming algorithm based on the concept of configurations. We consider as a configuration, a possible state of the platform, described by the number of servers on on each data center. More formally, configuration c at time slot t is defined as $c = (k_1, \ldots, k_M)$, where k_i is the number of servers on at time slot t on data center DC_i We defined as $P(c,t)$ the minimal power consumption of a schedule for the t first time slots, with k_i servers on on DC_i for each i at time slot t.

We denote $L(t)$ the set of pairs $(c, P(c,t))$, for all the possible configurations for time slot t. Notice that the total number of servers on during a time slot is directly related to the workload, thus $\sum_i k_i = \lceil N(t)/C \rceil$. As the complexity of this algorithm directly depends on the size of $L(t)$, this property permits to strongly reduce the execution time of experiments simulating this algorithm.

This algorithm computes recursively the set $L(t)$ for each time slot t, based on $L(t-1)$. More precisely, it first computes the possible configurations, that is the set of tuples $c = (k_1, ..., k_M)$ such that for all i, $k_i \leq S_i$ and $\sum_i k_i = \lceil N(t)/C \rceil$. Then, for each of these tuples, it computes $P(c,t)$ using $L(t-1)$. To that end, it considers each possible configuration c' at time slot $t-1$ and computes the power consumption at time slot t, based on the number of servers to turn on

Algorithm 5: Optimal algorithm

```
$L(t)$: list of values $(k_1, ..., k_M, P)$ where $k_i$ is the number of servers ON on
  $DC_i$ at current time slot and $P$ the lowest brown power consumed with this
  final configuration;
for $1 \leq t \leq t_{max}$ do
    forall $j_1, j_2, \cdots, j_M$ such that $\forall k, 0 \leq j_i \leq S_i$ and $\sum_i j_i = \lceil N_t/C \rceil$ do
        $P = +\infty$;
        forall $(k_1, ..., k_M, P_{t-1}) \in L(t-1)$ do
            if $P_{t-1} \leq P'$ then
                $P' = 0$;
                for $1 \leq i \leq M$ do
                    $P_i = 0$;
                    if $k_i > j_i$ then
                        $P_i + = (k_i - j_i) \times E_{OFF}$;
                    else
                        $P_i + = (j_i - k_i) \times E_{ON}$;
                    end
                    $P_i + = j_i \times P_S$;
                    $P' + = max(0, P_i - GP(t))$
                end
                $P' + = P_{t-1}$;
                if $P' < P$ then
                    $P = P'$;
                end
            end
        end
        Add $(j_1, ..., j_M, P)$ in $L(t)$;
    end
end
```

and off during this time slot and the power consumption for the servers on. When adding to $P(c', t-1)$, we obtain the minimum power consumption for t time slots with configuration c' at time $t-1$ and c at time t. We can then take the minimum for all $(c', P(c', t-1)) \in L(t-1)$ and we obtain $P(c,t)$. $L(0)$ is initialized with one unique tuple in the configuration c, $k_i = 0$ for all i and $P(c, 0) = 0$, this means all servers off, and no power consumed yet.

6 Validation framework

We evaluate our algorithm through a modeling and simulation (M&S) process. In the following, we first give an overview of the whole cloud implementation model (Section 6.1). We then detail our implementation of the data centers (Section 6.2), of the green power production (Section 6.3), of the cloud workload (Section 6.4), of the algorithm implementation (Section 6.5), and the different simulations performed (Section 6.6).

6.1 Simulation overview

The whole cloud implementation model is described in Figure 2. We simulate data centers using the DCSim (Data Center Simulator) discrete-event M&S tool [20]. This simulator provides the power consumption of each data center as a function of time.

We implement our algorithm in an ad-hoc way using the Java language into a simulated cloud controller. This simulator receives as inputs the green power production for each data center as well as the cloud workload (i.e. the number of VMs to deploy on the cloud for each time slot). Based on these inputs and on SAGITTA's algorithms, the controller generates for each server the VM allocation and the instructions which are directly sent to the simulated data center manager.

Note that we do not explicitly model the brown power production as we assume it to be infinite (at the scale of the cloud). We also ignore the telecommunication network as we assume it to have negligible impact on the system functioning (we assume network to be oversized for our scenario), and an almost constant power consumption over time if no energy-saving technique is applied [14]. Finally, we do not take into account here the energy consumed by the data centers' cooling systems.

In order to perform the simulations, we connect all these heterogeneous models using the MECSYCO (Multi-agent Environment for Complex-SYstem-CO-simulation) M&S platform [4, 5] which is based on the DEVS (Discrete-EVent System specification) formalism [24]. We have defined a DEVS interface for DCSim, and implemented it in MECSYCO.

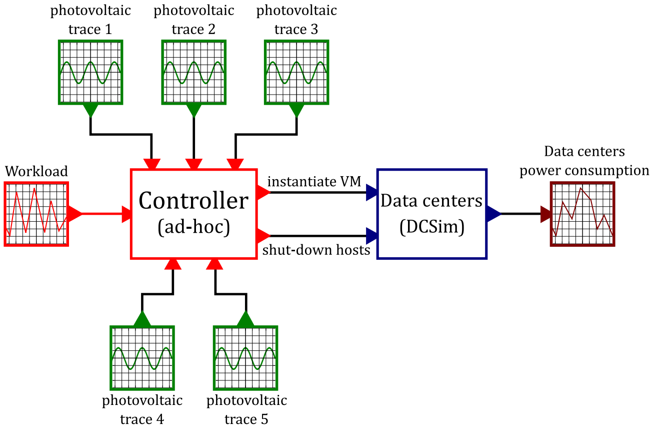

Fig. 2. Bloc diagram view of the cloud model from [3].

6.2 Data center simulation

Our cloud consists in five homogeneous data centers composed of five clusters. Each of these clusters contains 80 homogeneous nodes, so overall, the cloud comprises a total of 400 servers. The characteristics of each server are based on the Taurus servers of the French experimental testbed Grid'5000[4]. These Taurus servers are equipped with 2 Intel Xeon E5-2630 CPU with 6 cores each, 32GB memory, 598GB storage and a 10 Gigabit Ethernet interface. In order to determine the power consumption of each node, we implement the power model of [12], which is based on real measurements made on Taurus nodes. These measurements notably state that a Taurus server consumes 8W when powered OFF, 97W when idle, and 220W at 100% CPU load (i.e. $P_s = 220W$ for our algorithm).

Within this cloud, we deploy homogeneous VMs that are equivalent to the Amazon EC2 "large" flavor[5] - i.e. each VM requires 4 CPU cores, 8GB memory and 80GB storage. Hence, three VMs can be simultaneously running on one node. For the sake of simplicity, we assume that, when deployed, a VM always works at full capacity. In the same way, we neglect the delays for the VM to start/stop. All the VMs are automatically deleted at the end of each time slot. A time slot lasts five minutes in our simulations.

6.3 Green power production

In order to feed the controller during the simulation, we use real recordings of green power production and real workload traces. We get the former from the Photovolta project[6] of the University of Nantes. These recordings correspond to the power produced by a single Sanyo HIP-240-HDE4 photovoltaic panel updated every five minutes over one week. In order to have heterogeneous trajectories between data centers (and thus to represent solar irradiance differences between sites spread across a country), we select recordings starting at different dates, namely: 4th of September 2016, 2nd of February 2014, 8th of June 2014, 22nd of June 2015 and 21st of December 2014. We consider here that 30 photovoltaic panels (for a surface of $165.6m^2$) are installed at each data center. Then we scale these photovoltaic signals accordingly.

6.4 Workload input

We use the normalized ClarkNet HTTP trace of [20] for our cloud workload, shown in Figure 3. This workload trace spans over one week. We scale this workload to 98% of the cloud total capacity (i.e. the maximal workload peak represents 98% of the total computing capacity of the cloud). The trace peaks are synchronized with the photovoltaic signal ones to have proper day-night cycles in our simulation.

[4] https://www.grid5000.fr

[5] https://aws.amazon.com/ec2/

[6] http://photovolta2.univ-nantes.fr

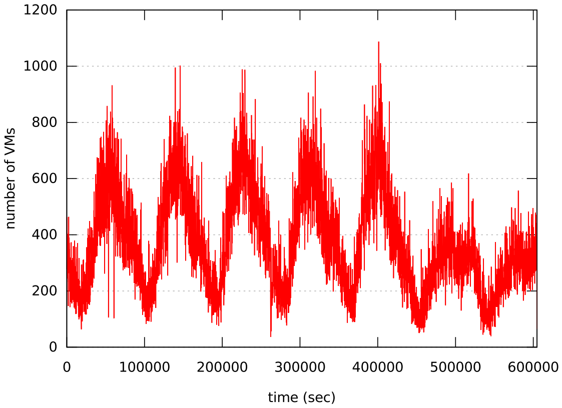

Fig. 3. The input workload used in the experiments from [3].

6.5 Algorithm implementation

The controller implementing our SAGITTA approach is run at each time slot (i.e. each five minutes). It saves all the data received from the green power sources during the current day. The controller computes at each time slot the standard deviations $p_i(t)$ using this history. It computes each expected green power production $Eg_i(t)$ by averaging a reference green power production trajectory (the Photovolta project recording of the 20th of August 2013 in our case which is the day with the best yield) scaled according to the last green power production received from i. More precisely, we denote $P_{ref}(t)$ the green power production at corresponding hour the day of reference (see Figure 4). We obtain the following formula:

$$Eg_i(t) = max\left(0, PV_i(t-1) + \frac{P_{ref}(t) - P_{ref}(t-1)}{2}\right).$$

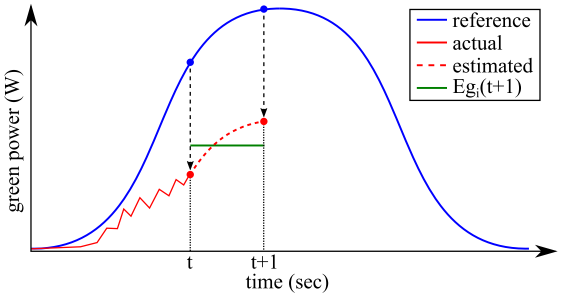

Fig. 4. Expected green power production computation for a time slot from t to $t+1$ from [3].

Note that we consider with this formula that $Eg_i(t)$ is equal to the average between the green power production received at $t-1$ and the one estimated at t. Thus, we take into account that the green power trajectory changes *during* the time slot, and not only at its beginning.

In order to minimize the number of ON/OFF cycles for the servers, the controller fills in priority the hosts already ON. Therefore, from a time slot to the next one, the controller keeps trace of the employed servers.

6.6 Simulated approaches

We compare SAGITTA performance against two Round-Robin inspired algorithms:

- **Round-Robin-VM** distributes the VMs fairly between the data centers regardless their green power production.
- **Round-Robin-DC** starts filling with VMs the first data center (in an arbitrary predefined order). If this data center becomes full, the algorithm starts using the next one, and so on.

Like SAGITTA, these two algorithms employ in priority the nodes already ON.

As the performance of Round-Robin-DC strongly depends on the order of the data centers, we test two opposite configurations corresponding to the best and the worst possible contexts. To define these contexts, we sort the photovoltaic traces according to the total amount of green energy they provide. We assign then the traces to the data centers following this order. The best context corresponds to the case where the photovoltaic traces are sorted in a decreasing order. Thus, the first data center (i.e. the one filled in priority) will be supplied by the best photovoltaic power trajectory. The worst context corresponds then to the case where the traces are sorted in an increasing order (i.e. the data center with the worst green power supply will always be filled first).

7 Results

Based on the simulation framework described in the previous section, several experiments were run to validate our proposed approach. The simulations are performed in order to compare SAGITTA against state-of-the-art approaches (Section 7.1). The influence of the green energy forecast is analyzed (Section 7.2). Various green production scenarios are studied to estimate the impact of green energy location on SAGITTA's performance (Section 7.3). The scalability of SAGITTA is evaluated by increasing the number of data centers (Section 7.4). SAGITTA is compared to the optimal approach (Section 7.5). Finally, we study the exactness of the green power production forecast (Section 7.6).

7.1 Energy consumption with SAGITTA vs. round-robin approaches

The second set of simulation integrates the switching ON/OFF costs and estimates their impact on the algorithms' energy consumption to reflect this point. Following the data collected by [15] on the Taurus cluster, we add a static energy consumption penalty of 5.28 Wh (consumed in 150 seconds) for each switch-ON command, and 0.56 Wh (consumed in 10 seconds) for each switch-OFF command sent. As shown in Table 2, even when considering these penalties, simulations show that SAGITTA performs better than the other solutions with a difference of at least 10%.

Table 2. Total cumulative brown energy consumption.

	SAGITTA	**Round-Robin-VM**	**Round-Robin-DC**
Best	2.77 MWh	3.38 MWh	3.02 MWh
Worst	2.77 MWh	3.38 MWh	4.4 MWh

Figure 5 shows the power consumption over time of each data center in the simulated cloud using SAGITTA. This figure also shows the number of transfers made by Algorithm 4 - a negative (respectively positive) value meaning that the algorithm switches off (respectively on) hosts. This plot highlights the usefulness of the transfer algorithm. For instance, at time 173,700 s. which corresponds to early morning, DC 2 starts producing green energy slightly earlier than DC 0. SAGITTA takes then advantage of this situation by performing 19 transfers from DC 0 to DC 2. Transfers are highly correlated with discontinuities in the green power production trajectories. Thus, the transfer decision may enable adapting the VM allocation, and consequently the energy consumption, to unforeseen increases and decreases of the green power production. In the absence of transfer, the switch on and off decisions enable adapting the DC workload to their green power production - i.e. the data centers with higher power production are generally more used than the others.

For the sake of simplicity, in the following, we will consider the best case for the Round-Robin-DC algorithm (with data centers ranked by their overall green energy production). All the simulations in the next sections also include the switching ON/OFF costs.

7.2 Influence of the green energy forecast

One basis of the SAGITTA approach is the green energy production forecast. The value $Eg_i(t)$, namely the expected PV production in DC_i at time slot t is computed regarding the electricity production at time slot $t-1$. This approach permits a simple computation for the value $Eg_i(t)$ to parametrize the probability law of green energy production. However, this formula estimates the electricity production regarding only the previous time slot, despite of the high volatility of

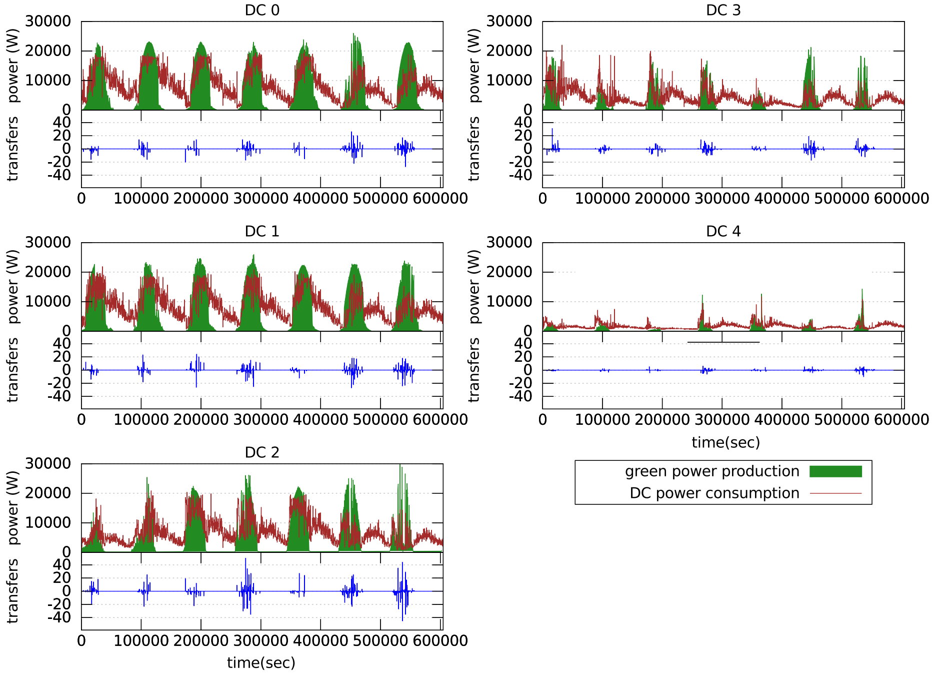

Fig. 5. Power consumption per data center with SAGITTA and transfer decisions.

solar energy. We experiment in this section an evaluation of $Eg_i(t)$ on a sliding window of PV production values. We target here the optimal size of the window, and the weight to give to the values of the different time slots of the window.

We propose several solutions to determine $Eg_i(t)$ on a sliding window of size s. For the sake of simplicity, we denote $g_i(t) = PV_i(t) - P_{ref}(t)$, with $P_{ref}(t)$ the daily production at same hour, the day of reference. We then make a weighted average value of values $g_i(t)$, with weight p_i:

$$Eg_i(t) = max\left(0, \frac{PV_i(t-1) + \frac{\sum_{k=1}^{s}(g_i(t-k)\times p_{s-k})}{\sum_{k=1}^{s} p_k} + P_{ref}(t)}{2}\right).$$

The first variant CST1 uses constant weigths $p_k = 1$ for recent and old values. In the second variant ADD1, the values of p_k increase linearly: $p_k = k + 1$. Finally, the values of p_k are multiplied by 2 at each step in PROD1: $p_k = 2^k$. In these variants, the computation includes values corresponding to the night, when $PV_i(t)$ and $P_{ref}(t)$ are both null. This impacts the estimation with useless values. Then, in variants CST2, ADD2 and PROD2, all values $g_i(t)$ corresponding to $P_{ref}(t) = 0$ are removed from the computation. Results of these computations are detailed in Figure 6.

The first unexpected result is the very low values of the optimal size of the sliding window. Regardless of the variant, the best size of the window is always

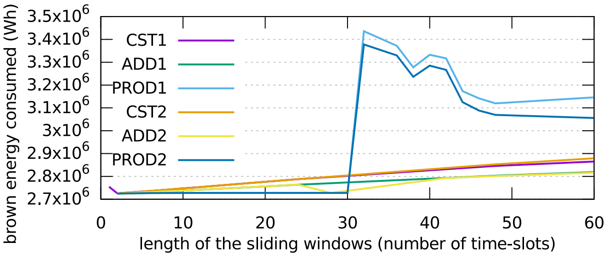

Fig. 6. Influence of Eg_i estimation from [3].

2, with a slight reduction of the brown energy consumed. The good performance of algorithms PROD1 and PROD2 can be related to the large weight given to the earliest production values in the computation. The weight given to early values has indeed a large impact on the variants' performance.

7.3 Influence of green energy production

Cloud providers need to adequately dimension their on-site photovoltaic panels (PVs). This issue involves a trade-off between the financial cost of installing and operating PVs, and the financial gains they are bringing in terms of green energy produced and thus, electricity that has not to be bought from the regular grid.

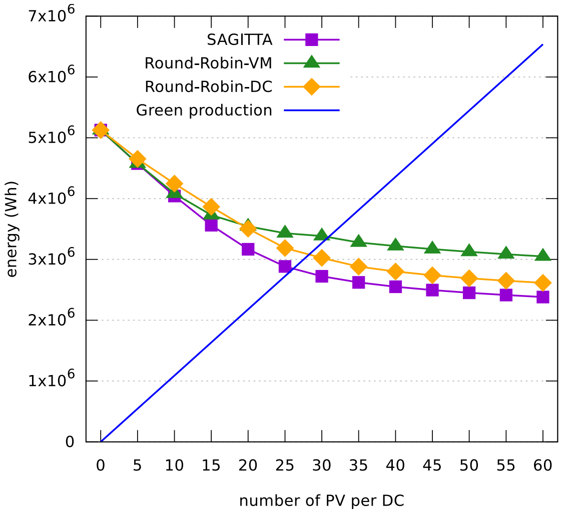

Fig. 7. Influence of green energy production on brown energy consumption from [3].

We perform a set of experiments to determine the influence of green energy production on SAGITTA performance. As shown in Figure 7, the number of

PVs varies per data center and the total brown power consumption is recorded over one week. We can see that, as soon as green energy is available, SAGITTA consumes clearly less brown energy than the other approaches.

Figure 7 also shows that up to about 25 photovoltaic panels, the brown energy consumption curves have a steeper slope, leading to higher gains per photovoltaic panels. For more than 25 photovoltaic panels, the energy gains are lower per added panel. When reaching 45 panels, the green energy production exceeds the total energy consumption of the data center (represented by the case with 0 panel). However, this production is concentrated during the day (as shown in Figure 5), whereas the workload, and consequently the energy consumption, spans over the day and the night. Thus, when reaching a number of photovoltaic panels whose production covers most of the Cloud energy consumption during daylight, adding panels can only save the energy consumption peaks at the beginning and the end of the day (when panels produce less energy), and their buying cost can thus exceed the monetary gains they generate.

Table 3. The considered cloud scenarios with increasing number of data centers.

Number of data centers	5	10	15	20	25	30	35	40
Total number of nodes	400	400	400	400	400	400	400	400
Number of PV per data centers	30	14	9	7	6	5	4	3

7.4 Scalability of SAGITTA

In order to check if the SAGITTA's energy savings scale up, we simulate the power consumption of distributed clouds with a larger number of data centers. For these different clouds, we progressively increase the number of data centers, and so the number of green power sources (still taken from the Photovolta project), while maintaining the same total number of nodes (and so an unchanged input workload). The total photovoltaic energy production is also kept as steady as possible by progressively decreasing the number of photovoltaic panels per data centers. Yet we decided not to consider fractions of panels, so the number of panels slightly varies between the scenarios to keep whole numbers. The compositions of these clouds are summed up in Table 3.

As shown in Figure 8, the simulation results discloses that SAGITTA scales up: it maintains its energy gains in larger clouds, and always consumes less brown energy than the other approaches. From a computing time point of view, in our simulation environment, it takes 9 seconds to execute SAGITTA over the whole workload trace (representing one week) for the case with 5 data centers, and 28 seconds for the case with 40 data centers. While this computing time is increased by a factor of 3 (when increasing the data center number by a factor of 8), it still remains inconsequential for the scalability of SAGITTA.

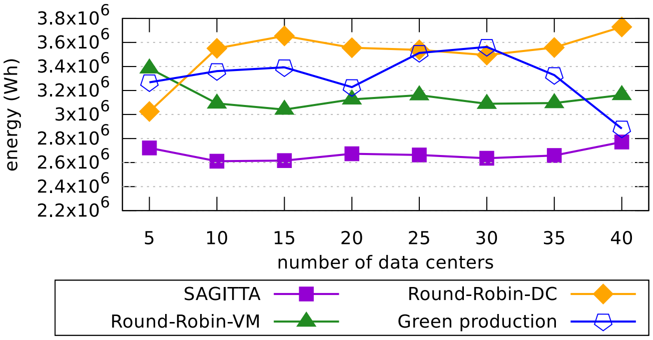

Fig. 8. Brown energy consumption of SAGITTA with increasing number of data centers.

7.5 Comparison with the optimal

We compare SAGITTA with the optimal solution when considering ON/OFF switches penalties. In order to compute this optimal solution, we implement the algorithm 5 in python. Due to the high level of computing resources required, we parallelize and distribute the first **forall** loop of the algorithm. For each time slot, we use 30 hosts of the Grid'5000 platform to run in parallel the algorithm. Even with this optimization, we only were able to compute the optimal solution for a cloud composed of 5 data-centers of 20 hosts. The algorithm took about 2 weeks to perform 1 week of simulation.

The results are shown in Tables 4 and 5. We can see that:

- SAGITTA is very close to the optimal solution although it requires way lesser computing resources than the optimal algorithm.
- SAGITTA still performs better than the other two algorithms.

Table 4. Total cumulative energy consumptions over one week when considering ON/OFF penalties.

	optimal	**SAGITTA**	**Round-Robin-VM**	**Round-Robin-DC**
Best	649,259Wh	666,238Wh	822,204Wh	733,304Wh
Worst	649,259Wh	666,238Wh	822,204Wh	1,086,626Wh

7.6 Exactness of the green power production forecast

The Table 6 shows the difference between the green power production predicted by SAGITTA and the actual ones. The Root-Mean-Square Deviation (RMSD) and the Normalized RMSD (NRMSD) are given in order to allow comparing SAGITTA with other future green power prediction models. We can see that

Table 5. Percentage of cumulative energy consumptions over the optimal when considering ON/OFF penalties.

	SAGITTA	Round-Robin-VM	Round-Robin-DC
Best	5.2%	26.6%	12.9%
Worst	5.2%	26.6%	67.4%

SAGITTA makes an average error of only 311.3W, which corresponds by comparison to the power consumed by 1.4 working hosts. This is a relatively small error when considering that the green production of each DC ranges from approximately 0W to 25,000W. This demonstrate that the prediction model of SAGITTA is accurate.

Table 6. Differences between the green power production predicted by SAGITTA and the actual ones.

	DC1	DC2	DC3	DC4	DC5	Average
Average	243.22W	332.96W	602.51W	303.71	74.44W	311.37W
Standard deviation	618.96W	884.74W	1456.11W	863.1W	270.21W	921.97W
RMSD	665.03W	945.32W	1575.84W	914.97W	280.27W	973.13W
NRMSD	2.66%	3.72%	6.22%	4.66%	2.14%	3.83%

8 Conclusion

In this chapter, we consider the problem of optimizing the green energy consumption of a geographically distributed cloud equipped with on-site photovoltaic panels. We tackle this challenge by distributing the cloud workload (composed of virtual machines) among the different data-centers.

We propose here a new version of the SAGITTA (a Stochastic Approach for Green consumption In disTributed daTA centers) approach which is based on this strategy. SAGITTA relies on a stochastic modeling of the expected green energy production in order to adequately allocate virtual machines to the data centers. The approach also switches off unused servers to save energy. In the new version, SAGITTA now natively takes into account of the energy costs arising from these on/off switches. This extension of SAGITTA is more robust and offers slightly better results.

We have proven in this paper that SAGITTA is locally optimal -i.e. that SAGITTA is optimal regarding the current time slot. In order to evaluate the performance of SAGITTA when considering the on/off switches costs, we also proposed a dynamic programming algorithm for computing the optimal energy consumption of the whole experiment. We conducted a simulation-based evaluation using real workload traces, wattmeter measurements on testbed servers, and real production traces from photovoltaic panels. We compared SAGITTA with two round-robin algorithms which do not consider the green energy production for allocating virtual machines in the cloud. The results show that SAGITTA can allocate virtual machines in a more energy-efficient way than these traditional approaches. Moreover, SAGITTA exhibits good results in term of brown energy consumption with a difference of only 5.2% with the optimal solution computed by our dynamic programming algorithm. The simulations also show that SAGITTA can adapt to different green energy production patterns as it outperforms traditional approaches in all these cases. Finally, we shown that SAGITTA can smoothly scale with the number of data centers belonging to the cloud.

In future work, we plan to extend SAGITTA by integrating the ability to dynamically migrate virtual machines and the energy production from one site to another. We also want to adapt SAGITTA to continuous (i.e. non time-slotted) workloads. Finally we want to integrate in our simulation the impact of network devices on the energy consumption.

Acknowledgments

This work has been supported by the Inria exploratory research project COSMIC (Coordinated Optimization of SMart grIds and Clouds).

References

1. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the Art of Virtualization. In: ACM Symposium on Operating Systems Principles (SOSP). pp. 164–177 (2003)
2. Callau-Zori, M., Samoila, L., Orgerie, A.C., Pierre, G.: An experiment-driven energy consumption model for virtual machine management systems. Tech. Rep. 8844, Inria (Jan 2016)
3. Camus, B., Dufossé, F., Orgerie, A.C.: A stochastic approach for optimizing green energy consumption in distributed clouds. In: International Conference on Smart Cities and Green ICT Systems (SMARTGREENS). pp. 47–59 (April 2017)
4. Camus, B., Galtier, V., Caujolle, M., Chevrier, V., Vaubourg, J., Ciarletta, L., Bourjot, C.: Hybrid Co-simulation of FMUs using DEV&DESS in MECSYCO. In: Symposium on Theory of Modeling & Simulation - DEVS Integrative M&S Symposium (2016)
5. Camus, B., Paris, T., Vaubourg, J., Presse, Y., Bourjot, C., Ciarletta, L., Chevrier, V.: MECSYCO: a Multi-agent DEVS Wrapping Platform for the Co-simulation of Complex Systems. Research report, LORIA (Sep 2016)
6. Deng, W., Liu, F., Jin, H., Li, B., Li, D.: Harnessing renewable energy in cloud datacenters: opportunities and challenges. IEEE Network 28(1), 48–55 (January 2014)
7. Fan, X., Weber, W.D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. In: ACM International symposium on Computer architecture (ISCA). pp. 13–23 (2007)

8. Figuerola, S., Lemay, M., Reijs, V., Savoie, M., St. Arnaud, B.: Converged Optical Network Infrastructures in Support of Future Internet and Grid Services Using IaaS to Reduce GHG Emissions. Journal of Lightwave Technology 27(12), 1941–1946 (2009)
9. How dirty is your data? Greenpeace report (2011)
10. Katz, R.H.: Tech Titans Building Boom. IEEE Spectrum 46(2), 40–54 (Feb 2009)
11. Koomey, J.: Growth in Data Center Electricity Use 2005 to 2010. Analytics Press (Aug 2011)
12. Li, Y., Orgerie, A.C., Menaud, J.M.: Opportunistic Scheduling in Clouds Partially Powered by Green Energy. In: IEEE International Conference on Green Computing and Communications (GreenCom) (Dec 2015)
13. Miyoshi, A., Lefurgy, C., Van Hensbergen, E., Rajamony, R., Rajkumar, R.: Critical power slope: understanding the runtime effects of frequency scaling. In: ACM International conference on Supercomputing (ICS). pp. 35–44 (2002)
14. Orgerie, A.C., Dias de Assunção, M., Lefèvre, L.: A Survey on Techniques for Improving the Energy Efficiency of Large-Scale Distributed Systems. ACM Computing Surveys (CSUR) 46(4), 47:1–47:31 (2014)
15. Rais, I., Orgerie, A.C., Quinson, M.: Impact of Shutdown Techniques for Energy-Efficient Cloud Data Centers. In: International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP). pp. 203–210 (Dec 2016)
16. Ren, S., He, Y., Xu, F.: Provably-Efficient Job Scheduling for Energy and Fairness in Geographically Distributed Data Centers. In: IEEE International Conference on Distributed Computing Systems (ICDCS). pp. 22–31 (2012)
17. Shehabi, A., Smith, S., Horner, N., Azevedo, I., Brown, R., Koomey, J., Masanet, E., Sartor, D., Herrlin, M., Lintner, W.: United States Data Center Energy Usage Report. Tech. rep., Lawrence Berkeley National Laboratory (2016)
18. Snowdon, D., Ruocco, S., Heiser, G.: Power Management and Dynamic Voltage Scaling: Myths and Facts. In: Workshop on Power Aware Real-time Computing (2005)
19. Talaber, R., Brey, T., Lamers, L.: Using Virtualization to Improve Data Center Efficiency. Tech. rep., The Green Grid (2009)
20. Tighe, M., Keller, G., Bauer, M., Lutfiyya, H.: DCSim: A data centre simulation tool for evaluating dynamic virtualized resource management. In: Workshop on Systems Virtualization Management. pp. 385–392 (Oct 2012)
21. Tripathi, R., Vignesh, S., Tamarapalli, V.: Optimizing Green Energy, Cost, and Availability in Distributed Data Centers. IEEE Communications Letters (2016)
22. Wang, D., Ren, C., Sivasubramaniam, A., Urgaonkar, B., Fathy, H.: Energy storage in datacenters: What, where, and how much? In: ACM SIGMETRICS/PERFORMANCE. pp. 187–198 (2012)
23. Wang, L., Tao, J., Kunze, M., Castellanos, A., Kramer, D., Karl, W.: Scientific Cloud Computing: Early Definition and Experience. In: IEEE International Conference on High Performance Computing and Communications (HPCC). pp. 825–830 (2008)
24. Zeigler, B., Praehofer, H., Kim, T.: Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press (2000)