Energy Usage and Carbon Emission Optimization Mechanism for Federated Data Centers

Dang Minh Quan[1], Andrey Somov[2], and Corentin Dupont[2]

[1] Institute of Information Technology for Economic,
National Economic University, Vietnam
quandm@upb.de
[2] CREATE-NET, Trento, Italy
{asomov,cdupont}@create-net.org

Abstract. This work addresses the problem of high energy consumption and carbon emissions by data centers which support the *traditional* computing style. In order to overcome this problem we consider two allocation scenarios: *single allocation* and *global optimization* of available resources and propose the optimization algorithms. The main idea of these algorithms is to find a server in the data center with the lowest energy consumption and/or carbon emission based on current status of data center and service level agreement requirements, and move the workload there. The optimization algorithms are devised based on Power Usage Effectiveness (PUE) and Carbon Usage Effectiveness (CUE). The simulation results demonstrate that the proposed algorithms enable the saving in energy consumption from 10% to 31% and in carbon emission from 10% to 87%.

Keywords: data center, traditional computing, power consumption, green computing, resource management.

1 Introduction

Until recently, the key performance indicator of a data center was its performance. However, the growing number of IT services and large scale tasks, resulting in higher power consumption [1] and carbon emission, have forced the ICT community to concern the energy efficiency of data centers carefully [2].

Several energy-aware approaches and resource management techniques have been introduced to tackle power consumption problem in the data center domain from different points of view. Many approaches are focused on workload consolidation in order to decrease the number of servers by switching them off/sleep and, therefore, to reduce the power consumption [3] [4]. Some techniques, in contrast, put efforts in finding the solutions of optimal workload placement in order to minimize the cooling systems energy consumption [5] while similar research works investigate the opportunity to reduce the power consumption of cooling systems by using intelligent scheduling or choosing the optimum temperature of cold air [7] [8]. The biological algorithm in [6] determines more power efficient servers within a data center facility and moves workload there.

J. Huusko et al. (Eds.): E2DC 2012, LNCS 7396, pp. 129–140, 2012.

The objective of this research work is twofold: to reduce the *power consumption* and *carbon emission* of a federated data center with the traditional mode of computation. To achieve this objective we propose the optimization algorithms for the single allocation of virtual machines and for global resources optimization.

The paper is organized as follows: Section 2 describes the problem formulation. Sections 3 and 4 present the algorithm for single allocation request and global optimization respectively. The simulation results based on different scenarios and the data centers configurations are shown in Section 5. Finally, we conclude and discuss our future work in Section 6.

2 Problem Formulation

We assume that we have a set of data centers D including N centers. Assume that in each data center d_l we have a set of servers S_l. Each server $s_{li} \in S_l$ is characterized with the number of cores and the amount of memory ($s_{li} \cdot nr\_Core$ and $s_{li} \cdot nr\_RAM$, respectively). As servers use central storage infrastructures (such as SAN, NAS or iSCSI external storage arrays), we do not have to care about storage in each server.

Each server s_{li} has a set of running virtual machines V_{li} including k_{li} virtual machines. Each virtual machine $v_j \in V_{li}$ is characterized with required number of virtual CPU and amount of memory ($v_j \cdot r\_vCPU$ and $v_j \cdot r\_RAM$, respectively) and the average CPU usage rate computed in % and the amount of memory ($v_j \cdot a\_Urate$, $v_j \cdot a\_RAM$).

With each server s_{li} the following constraints (1)-(4) have to be met. The total usage rate of a certain number of CPUs on a certain number of VMs can not exceed the safe performance factor for this certain number of cores:

$$\sum_{j=1}^{k_{li}} v_j \cdot r\_vCPU * v_j \cdot a\_Urate <= k * s_{li} \cdot nr\_Core \tag{1}$$

where k is the safe performance factor, $k < 1$.

The average total memory used by the VMs on one server cannot exceed the amount of total available memory of the server:

$$\sum_{j=1}^{k_{li}} v_j \cdot a\_RAM <= s_{li} \cdot nr\_RAM \tag{2}$$

The total number of VMs with required number of virtual CPUs is less or equal to number of cores with predefined maximum number of virtual CPUs on board:

$$\sum_{j=1}^{k_{li}} v_j \cdot r\_vCPU <= s_{li} \cdot nr\_Core * \max vCPUpCore \tag{3}$$

where *maxvCPUpCore* is maximum number of virtual CPUs per Core.

The number of the server's VMs can not exceed the maximum number of VMs, *maxVMpServer*, allowed for the server:

$$k_{li} <= \max VMpServer \tag{4}$$

2.1 Allocation Request

Assume we assign a new virtual machine with requirement ($v.r\_vCPU$, $v.r\_RAM$) to server s_{li} as virtual machine v_{k+1}. The constraints above have to be met also with the new VM, so we have the following constraints.

$$\sum_{j=1}^{k_{li}} v_j \cdot r\_vCPU * v_j \cdot a\_Urate + v_{k+1} \cdot r\_vCPU <= \\ <= k * s_{li} \cdot nr\_Core \tag{5}$$

$$\sum_{j=1}^{k_{li}} v_j \cdot a\_RAM + v_{k+1} \cdot r\_RAM <= s_{li} \cdot nr\_RAM \tag{6}$$

$$\sum_{j=1}^{k_{li}} v_j \cdot r\_vCPU + v_{k+1} \cdot r\_vCPU <= \\ <= s_{li} \cdot nr\_Core * \max vCPUpCore \tag{7}$$

$$k_{li} + 1 <= \max VMpServer \tag{8}$$

Before assigning the new VM to the server s_{li}, its energy consumption, E_{li0}, is

$$E_{li0} = f(U_{li0}) \tag{9}$$

and CPU usage rate is

$$U_{li0} = \frac{\sum_{j=1}^{k_{li}} v_j \cdot r\_vCPU * v_j \cdot a\_Urate}{s_{li} \cdot nr\_Core} \tag{10}$$

The total energy TE_{li0}, that data center used for the server s_{li} before assigning the new VM is

$$TE_{li0} = PUE_l * E_{li0} \tag{11}$$

The total CO_2 emission, C_{li0}, due to the server s_{li} in the data center before assigning the new VM is

$$C_{li0} = CUE_l * E_{li0} \tag{12}$$

After assigning, the energy consumption is

$$E_{li1} = f(U_{li1}) \tag{13}$$

where CPU usage rate is

$$U_{li1} = \frac{\sum_{j=1}^{k_{li}} v_j \cdot r\_vCPU * v_j \cdot a\_Urate + v_{k+1} \cdot r\_vCPU}{s_{li} \cdot nr\_Core} \tag{14}$$

If the goal is in optimizing energy, we have to pick the server s_{li} in a way that minimizes (E_{li1}-E_{li0}).

CO_2 emission is

$$C_{li1} = CUE_l * E_{li1} \tag{15}$$

If the goal is in optimizing CO_2 emission, we have to pick the sever s_{li} in a way that minimizes (C_{li1}-C_{li0}).

2.2 Global Optimization Request

When moving a virtual machine v_{li} from center d_l to center d_k, we have to consider the following constraints:

- Transfer time must be less than T
- Energy spent for the transfer must be less than E_t

After redistributing all running virtual machines over D, each server s_{li} has an updated set of running virtual machines V_{li1} including k_{li1} virtual machines. The new arrangement must also satisfy the above constraints (1)-(4).The total energy used in the federated data center is then

$$E_1 = \sum_{l=1}^{N} f(U_{li1}) \text{ with } i \in [1,n], l \in [1,N] \tag{16}$$

where n is the number of servers and N is the number of data centers, $f(U_{li1})$, and CPU usage rate is as follows

$$U_{li1} = \frac{\sum_{j=1}^{k_{i1}} v_j .r\_vCPU * v_j .a\_Urate}{s_{li} .nr\_Core} \tag{17}$$

CO_2 emission is

$$C_1 = \sum_{l=1}^{N} f(U_{li1}) * CUE_l \text{ with } i \in [1,n], j \in [1,N] \tag{18}$$

We have to rearrange the workload in a way that minimizes the CO_2 emission C_1.

If the goal is minimizing energy consumption, we also minimize C_1 but with CUE_l replaced by PUE_l.

3 Algorithm for Single Allocation Request

The main idea of the proposed algorithm is to go through each server in each data center to see the consumed energy and CO_2 emission if we assign the VM to that server of the data center. From those collected data we will select the server of the data center having the smallest CO_2 emission. The algorithm for minimizing energy is similar.

The algorithm for single allocation request (see Fig. 1) is presented in Table 1. In order to satisfy the VM allocation request the algorithm tries to find a server with minimum energy overhead. To do so, the algorithm takes into consideration the information about data center (*data center description*), including the CUE, and considers the acting SLA. Besides, the algorithm communicates with the power calculator for the evaluation of required power resources. Based on all these parameters, the algorithm, finally, proposes the candidate server for the allocation.

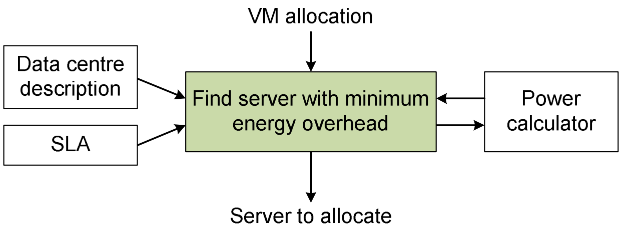

Fig. 1. Single allocation request diagram

The pseudo code provides the algorithm (see Table 1) explanation in more details and is as follows:

Table 1. Single allocation request for federated traditional data center

```
Input: model of all data centers in the federation, constraints,
characteristics of the incoming VM
Set b_meet=0
For each data center d1
{
        Step 0: The servers of d1 are in an array A1
        Step 1: Calculate power consumption P1 and CO2
        emission C1 of d1
        Step 2: Set i=0
        Step 3: Check if the resources of the server
        at array index i meet the requirement
        Step 4: If not go to Step 8
        Step 5: Calculate power consumption P1i and CO2
        emission C1i of the data center if the VM is
        deployed on the server at array index i
        Step 6: Save the tuple (1,i, C1i- C1)
        in a list L
        Step 7: Set b_meet=1
        Step 8: i++
        Step 9: Repeat the process from Step 3 until i
        exceeds the size of the array
}
```

Table 1. (*continued*)

```
If b_meet==1
       Go through the list L to find the tuple having
       the smallest C1i- C1
       Output: the server in the data center having
       the smallest C1i- C1
If b_meet==0
       Output: no solution
```

4 Algorithm for Global Optimization Request

This section describes the algorithm for global optimization request (see Fig. 2).

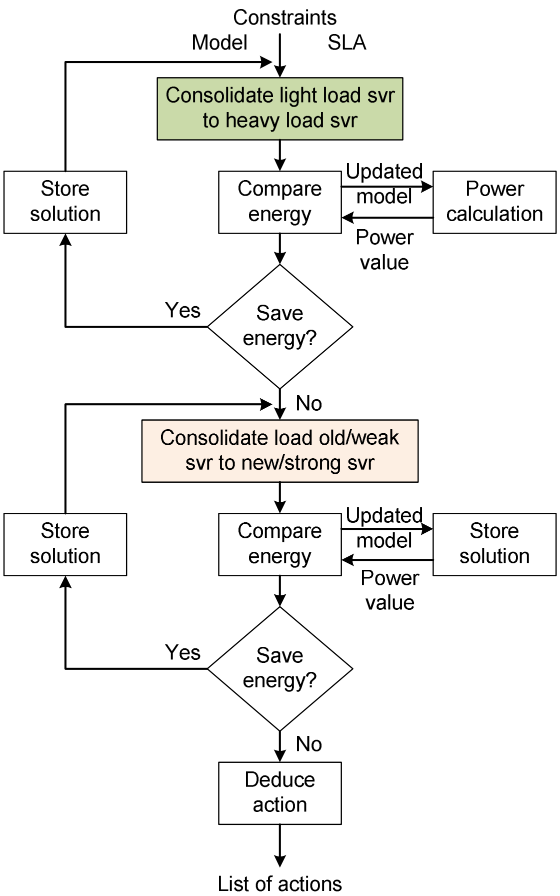

Fig. 2. Global optimization flow diagram

The main idea of the algorithm is to move all the heavy loading VMs to the servers which have the best rate of CO_2 emission or the largest computational horsepower. The VMs which cannot conveniently be moved out of the old data centers are rearranged inside the same data center to achieve the smallest possible CO_2 emission value. The same mechanism can be applied to optimize the energy by replacing CUE by PUE.

The algorithm for global allocation request for federated traditional data center is as follows:

Table 2. Global allocation request for federated traditional data center

```
Input: model of all data centers in the federation, constraints
Step 0: Form list of all VMs LV running in all data centers
Step 1: Form list of all servers LS in all data centers
Step 2: Sort the list LV in descending order of the actual CPU
usage A.
The actual CPU usage A of a VM v is calculated with
A= v.r_vCPU * v.a_Urate
Step 3: Sort the list LS in ascending order of the maximum CO2
emission E per core of the server.
The maximum CO2 emission E per core of the server s is calculated
with
E= s.max_Power*CUE/s.nr_Core
Step 4: For each server in the list LS, we use the traditional
pack algorithm to reassign VMs in the list LV to the server; then
remove reassigned VMs out of LV
Step 5: If there are still VMs in the list LV, we return the empty
exchange list
Step 6: From the new assignment, form the exchange list
Output: Exchange list containing VMs that should be moved to other
servers
```

Table 3. Pack algorithm for traditional data center

```
Input: The server, list of VMs LV, constraints about transfer time

Algorithm:
Step 1: Determine the number of free resources in the server
(fCPU, fMem)
Step 2: (where nCPU and nMem is the number of CPU and size of RAM
available on the server, respectively):
For each VM in LV
{
    if((fCPU*k < nCPU*u_rate) && (fMem < nMem))
    {
       Calculate the transfer time TF from old data
       center to the one
       If TF meets the transfer time constraints
              Put VM into list L1
              Update the amount of free resources:
              fCPU -= nCPU*u_rate
              fMem -= nMem
    }
}
Output: L1
```

5 Simulation Results

This section presents the simulation scenario and results for the federated traditional data center case.

5.1 Simulation Scenario

The simulation is done to study the saving rate of the resource allocation mechanism in different resource configuration scenarios. To do the simulation, we use 4 server classes with single core, duel cores, quad cores and six cores. The main parameters for each server class are presented in Table 4.

Table 4. Server configuration

Server type (i)	Nr. CPU	Nr. Cores	P_{idle} CPU (W)	f_i (GHz)	RAM (MB)	Nr. Fan	Disk (MB)	P_{max} (W)
1	1	1	7.57	2.0	1000	4	400	102.22
2	1	2	9.88	2.0	2000	4	500	103.39
3	1	4	20.14	2.2	4000	5	800	171.70
4	1	6	22	2.4	6000	6	1000	229

We generated 3 resource configuration scenarios: modern data center, normal data center and old data center as in Table 5. In the normal data center the percentage of different server classes is fully balanced. In the old data center, the percentage of server classes with less cores is predominant.

Table 5. Resource scenarios

Scenario	Nr. Server type1	Nr. Server type2	Nr. Server type3	Nr. Server type4
Modern –1	50	100	150	200
Normal – 2	100	100	100	100
Old – 3	200	150	100	50

With each resource configuration, we generated a raw set of jobs. The jobs come randomly to the system within the period of 1000 time slots. The parameters for each job are determined by random selection. As the jobs in the traditional computing style scenario may have a long runtime period, we selected the runtime for each job spanning from 1 to 100 time slots. It should be noted that 1 time slot lasts 5 minutes.

We generated 3 federated data centers configurations: federated data centers with many old data centers, federated data centers with balanced types of data center, federated data centers with many modern data centers. The detail of each federated data centers configuration is presented in Table 6.

Table 6. Configuration of three federated data centers

Federated ID	Configuration	Nr. Old centers	Nr. Normal centers	Nr. Modern centers
1	Many Old	6	3	1
2	Balanced	3	4	3
3	Many Modern	1	3	6

We use 3 PUE/CUE configurations as presented in Table 7.

Table 7. PUE/CUE configurations

Type	Energy source	PUE	ESC	CUE
Low	Oil 20%, Hydro 40%, Nuclear 40%	1.3	0.12844	0.166792
Medium	Coal 50%, Nuclear 30%, Hydro 20%	1.5	0.45983	0.689745
High	Coal 80%, Oil 20%	1.8	0.85	1.53

To assign PUE/CUE to data center, we use 3 assigning configurations as presented in Table 8.

Table 8. Assigning configurations

Assign ID	Energy source
1	Old data center high PUE/CUE, normal data center normal PUE/CUE, modern data center low PUE/CUE
2	Old data center normal PUE/CUE, normal data center normal PUE/CUE, modern data center normal PUE/CUE
3	Old data center low PUE/CUE, normal data center normal PUE/CUE, modern data center high PUE/CUE

With each federated data centers configuration, with each PUE/CUE configuration, we run 3 simulation scenarios.

- Scenario 1: Each new VM will be allocated with the single allocation algorithm developed in phase 1 (see D4.1), in its own data center
- Scenario 2: Each new VM will be allocated with single federated allocation algorithm (see Chapter III.1.1.a.).
- Scenario 3: Each new VM will be allocated with single federated allocation algorithm (see Section 3). Every *k* timeslots, call global federated allocation algorithm.

For each scenario, calculate the Energy/CO_2 in 1000 timeslots where 1 timeslot is an arbitrary unit, which simulates 5 minutes of real time. We compare the result of the three aforementioned scenarios.

5.2 Results

The simulation results in terms of energy consumption (in MW*timeslot) are presented in Table 9.

Table 9. Simulation results for energy consumption

Federated ID	Assign ID	Single allocation - Energy	Single federated allocation		Single + global federated allocation		
			Energy	Saving	Energy	Transfer Energy	Saving
1	1	385.79	277.049	28%	265.6591	0.003442	31%
1	2	354.12	329.6796	7%	313.7542	0.003736	11%
1	3	338.59	289.3345	15%	275.3579	0.003646	19%
2	1	395.18	314.6082	20%	301.7559	0.049073	24%
2	2	396.13	369.2494	7%	354.1649	0.033214	11%
2	3	413.55	326.031	21%	312.712	0.004585	24%
3	1	412.09	357.4885	13%	343.6139	0.068402	17%
3	2	445.99	418.3728	6%	402.1352	0.003558	10%
3	3	502.17	367.9638	27%	353.6827	0.004252	30%

The energy saving rate achieved applying federated optimization compared to not applying it is also presented in Fig. 3.

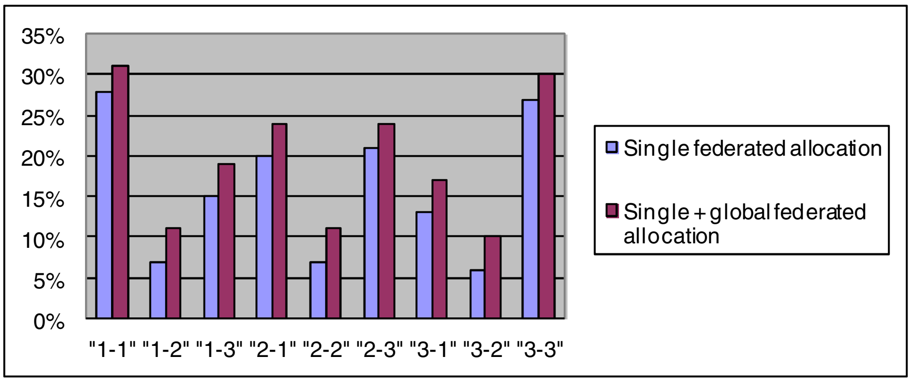

Fig. 3. Energy saving rate of applying federated optimization for traditional data centers

The simulation result in terms of CO_2 emission (in Ton*timeslot/h) is presented in Table 10.

Table 10. Simulation results for CO_2 emission.

Federated ID	Assign ID	Single allocation - CO_2	Single federated allocation		Single + global federated allocation		
			CO_2	Saving	CO_2	Transfer CO_2	Saving
1	1	252.81	34.36192	86%	32.94925	0.00279	87%
1	2	162.83	151.5919	7%	144.2691	0.005537	11%
1	3	124.12	40.41604	67%	38.46371	0.008822	69%
2	1	183.24	37.32418	80%	35.79941	0.005822	80%
2	2	182.15	169.7897	7%	162.8534	0.003441	11%
2	3	233.33	45.11798	81%	43.27482	0.00734	81%
3	1	117.63	41.04069	65%	39.44785	0.007853	66%
3	2	205.08	192.3807	6%	184.9142	0.00496	10%
3	3	363.22	49.277	86%	47.3645	0.006843	87%

The CO_2 emission saving rate achieved applying federated optimization compared to not applying it is also presented in Fig.4.

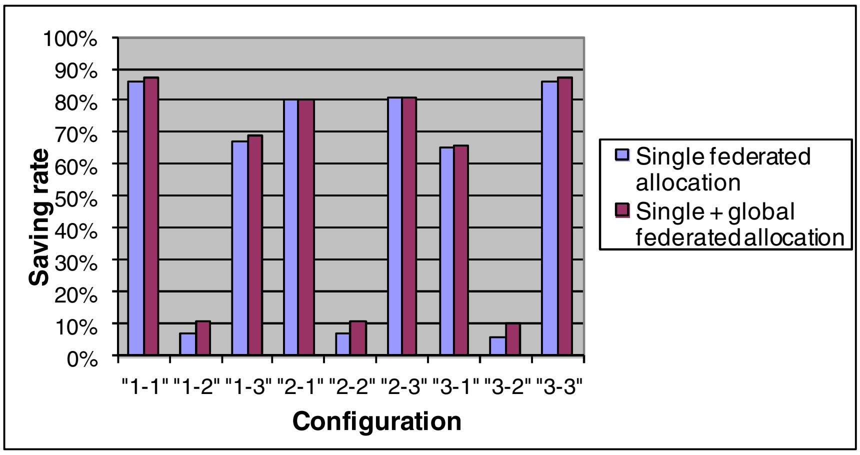

Fig. 4. CO_2 emission saving of applying federated optimization for traditional data centers

The simulation result shows the efficiency of the energy aware algorithms for the federated traditional data centers. Depending on the configuration, the saving rate spans from 10% to 31% in energy consumption and from 10% to 87% in CO_2 emission.

From the distribution of saving rate for both energy and CO_2 emission according to configurations, we can see that the federated optimization algorithm is very effective when the values of PUE/CUE of each data center in the federation is greatly differ from each other. Indeed, with the simulation configurations 1-2, 2-2 and 3-2, where the PUE/CUE values are the same for each data center, the saving rate is much smaller than in other configurations where the PUE/CUE values have a large variance.

6 Conclusions and Future Work

In this work we have proposed the optimization algorithms for resources management in a federated data center. To decrease the power consumption and carbon emission, we needed to find the servers with the lowest power consumption and shift the workload to these facilities. Simulations made with different types and configurations of data centers have shown that the power consumption saving can be up to 31% and carbon emission reduction can be up to 87%.

Our future work aims at applying the devised optimization algorithms in cloud data centers and evaluate the energy and carbon emission saving rates.

Acknowledgements. The authors would like to acknowledge the FIT4Green project (grant agreement no. 249020) sponsored by the European Commission. The authors also thank Mikko Majanen (VTT, Finland) for his valuable comments.

References

1. Dang, M.-Q., Basmadjian, R., De Meer, H., Lent, R., Mahmoodi, T., Sannelli, D., Mezza, F., Dupont, C.: Energy efficient resource allocation strategy for cloud data centres. In: 26th Int. Symposium on Computer and Information Sciences, pp. 133–141. Springer Press (2011)
2. Berl, A., Gelenbe, E., Di Girolamo, M., Giuliani, G., De Meer, H., Dang, M.-Q., Pentikousis, K.: Energy-Efficient Cloud Computing. J. Computer 53(7), 1045–1051 (2010)
3. Bradley, D.J., Harper, R.E., Hunter, S.W.: Workload-based power management for parallel computer systems. IBM J. of Research and Development 47(5-6), 703–718 (2003)
4. Meisner, D., Gold, B.T., Wenisch, T.F.: PowerNap: Eliminating server idle power. In: 14th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 205–216. ACM Press (2009)
5. Carrol, R., Balasubramaniam, S., Donnelly, W.D.: Dynamic optimization solution for green service migration in data centres. In: IEEE International Conference on Communications, pp. 1–6. IEEE Press (2011)
6. Barbagallo, D., Di Nitto, E., Dubois, D.J., Mirandola, R.: A Bio-inspired Algorithm for Energy Optimization in a Self-organizing Data Center. In: Weyns, D., Malek, S., de Lemos, R., Andersson, J. (eds.) SOAR 2009. LNCS, vol. 6090, pp. 127–151. Springer, Heidelberg (2010)
7. Berral, J.L., Goiri, I., Nou, R., Julia, F., Guitart, J., Gavalda, R., Torres, J.: Towards energy-aware scheduling in data centers using machine learning. In: 1st International Conference on Energy-Efficient Computing and Networking, pp. 215–224. ACM Press (2010)
8. Tang, Q., Gupta, S.K.S., Varsamopoulos, G.: Energy-efficient thermal-aware task scheduling for homogemeous high-performance computing data centers: a cyber-physical approach. IEEE Transactions on Parallel and Distributed Systems 19(11), 1458–1472 (2008)