Hierarchical Approach for Green Workload Management in Distributed Data Centers

Agostino Forestiero[1], Carlo Mastroianni[1], Michela Meo[2],
Giuseppe Papuzzo[1], and Mehdi Sheikhalishahi[3]

[1] ICAR-CNR and Eco4Cloud srl, Rende (CS), Italy
[2] Politecnico di Torino, Italy
[3] University of Calabria, Rende (CS), Italy

Abstract. The efficient management of geographically distributed data centers has become an important issue not only for big companies that own several sites, but also due to the emerging of inter-Cloud infrastructures that allow heterogeneous data centers to cooperate. These environments open unprecedented avenues for the support of a huge amount of workload, but they need the definition of novel algorithms and procedures for their management, where scalability is a priority. The complexity derives by the size of the system and by the need for accomplishing several and sometimes conflicting goals, among which: load balancing among multiple sites, prevention of risks, workload consolidation, and reduction of costs, consumed energy and carbon emissions. In this paper a hierarchical approach is presented, which preserves the autonomy of single data centers and at the same time allows for an integrated management of heterogeneous platforms. The framework is purposely generic but can be tailored to the specific requirements of single environments. Performances are analyzed for a specific Cloud infrastructure composed of four data centers.

Keywords: Cloud Computing, Distributed Data Center, Energy Saving.

1 Introduction

The ever increasing demand for computing resources has led companies and resource providers to build private warehouse-sized data centers, or to offload applications to the data centers owned by a Cloud company. Overall, data centers require a significant amount of power to be operated. The total electricity demand of data centers increased by about 56% from 2005 to 2010, and the electricity usage accounted for about 1.5% of the worldwide electricity usage in 2010 [6], which is comparable to the aviation industry. The financial impact for the data center management is also huge, since a data center spends between 30% to 50% of its operational expense toward electricity. The efficient utilization of resources in these data centers is therefore essential to reduce costs, energy consumption, carbon emissions and also to ensure a high quality of service to users.

The virtualization technology allows multiple Virtual Machines (VMs) to be run on the same physical server. Although this helps to increase the efficiency of data centers,

L. Lopes et al. (Eds.): Euro-Par 2014 Workshops, Part I, LNCS 8805, pp. 323–334, 2014.

the optimal distribution of the applications to servers [1] is still an open problem, especially in large and dynamic systems. The problem is even more complex in geographically distributed data centers, whose adoption is rapidly increasing. They are deployed by major cloud service providers, such as Amazon, Google, and Microsoft, to match the increasing demand for resilient and low-latency cloud services, or to interconnect heterogenous data centers owned by different companies, in the so-called "Inter-Cloud" scenario. In such environments, data centers offer different and time-varying energy prices, and workload variability is experienced both within single sites and across the whole infrastructure.

The dynamic migration of workload among data centers has become an opportunity to improve several aspects: better resiliency and failover management, improved load balancing, and exploitation of the the "follow the moon" paradigm, i.e., move the workload where the energy is cheaper/cleaner and/or cooling costs are lower. Inter-site migration is enabled by the availability of a much higher network capacity, thanks to both physical improvements (e.g., through techniques such as wavelength division multiplexing) and logical/functional enhancements (e.g., the adoption of Software Defined Networks). Reliable and low-latency connections can be used to shift significant amount of workload from one site to another through dedicated networks or even via regular Internet connections.

Nonetheless, these advancements do not reduce the complexity of the involved issues, among which: determine whether the benefits of workload migrations may overcome the drawbacks, from which site and to which site to migrate, what specific portion of the workload should be migrated, how to reassign the migrating workload in the target site, etc. Some significant efforts have been done in this area. The electricity price variation, both across time and location, is exploited to reduce overall costs using different strategies, among which: the Stratus approach [2] exploits Voronoi partitions to determine to which data center requests should be routed; the algorithm proposed by Mehta et al. [9] assigns virtual machines to servers using a constraint programming approach; Ren at al. [10] use an online scheduling algorithm based on Lyapunov optimization techniques. The algorithms presented in [7] and [4] tackle the problem considering the user's point of view, and aim to choose the most convenient data center to which the user should consign a service or VM.

However, the cited approaches, as well as many others, aim to solve the optimization problem as a whole, in a centralized fashion, undergoing the risk of originating two main issues: (i) algorithms of this kind may be poorly scalable, both for the number of parameters that they must consider and for the huge size of the problem, as it may involve tens of thousand of servers; (ii) they generally assume that all sites share the same strategy and algorithms, which may hamper their autonomy. The need for autonomous management is self-explanatory in multi-owned data centers, and is crucial even within a single-owner infrastructure, for example in the case that one or several sites are the former asset of an acquired company, or are hosted by co-located multi-tenant facilities.

A self-organizing hierarchical architecture is proposed in [3], but it is limited to the management of a single data center. This paper presents EcoMultiCloud, a hierarchical framework for the distribution and consolidation of the workload on a multi-site platform. The framework allows for an integrated and homogeneous management of heterogeneous platforms but at the same time preserves the autonomy of single sites. It is composed of two layers: at the lower layer, each site adopts its own strategy to distribute and consolidate the workload internally. At the upper layer, a set of algorithms – shared by all the sites – are used to evaluate the behavior of single sites and distribute the workload among them, both at the time that new applications/VMs are assigned and when some workload migration from one site to another is deemed appropriate. At each site one server is elected as point of contact (PoC) and periodically sends to other sites' PoCs a number of parameters that summarize the state of the site, possibly including the overall utilization of resources, the efficiency of computation, the energy costs, the amount of CO_2 emissions, etc. Upon reception of such data from the other sites, the PoC executes the upper layer algorithms to: (i) determine the target data center to which a new application should be assigned; (ii) check if the workload is well balanced among the different sites, and (iii) trigger migration of applications when needed. This strategy resembles the one used to cope with traffic routing in the Internet, where a single protocol – Border Gateway Protocol – is used to interconnect different Autonomous Systems (ASs), while every AS is free to choose its own protocol – e.g., RIP or OSPF – for internal traffic management.

The reminder of the paper is organized as follows: Section 2 describes the EcoMultiCloud architecture and illustrates the roles and objectives of two layers; Section 3 describes the algorithm used by the upper layer for the assignment of Virtual Machines; Section 4 analyzes the performance of the assignment algorithm in terms of carbon emissions and load balancing and compares EcoMultiCloud with a reference algorithm; Finally, Section 5 concludes the paper.

2 Architecture for Inter-DC Workload Distribution

This section describes the hierarchical architecture of EcoMultiCloud for the efficient management of the workload in a multi-site scenario. The architecture is composed of two layers: (i) the *upper layer* is used to exchange information among the different sites and drive the distribution of Virtual Machines among the data centers and (ii) the *lower layer* is used to allocate the workload within single data centers.

EcoMultiCloud extends the decentralized/self-organizing approach, recently presented in [8], for the consolidation of the workload in a single data center. The single data center solution, referred to as EcoCloud, dynamically consolidates Virtual Machines (VMs) to the minimum number of servers, and allows the remaining servers to enter low consuming sleep modes. With EcoCloud key decisions regarding the local data center are delegated to single servers, which autonomously decide whether or not to accommodate a VM or trigger a VM migration. The data center manager has only a coordination role. In a similar fashion, at the upper level of the multi-site EcoMultiCloud architecture, most of the intelligence is left to single data centers which, for example, decide which information is relevant and should be delivered to other data centers, which portion of the local workload should be migrated elsewhere, etc. Coordinating decisions, for example about the necessity of migrating an amount of workload from one site to another, are taken combining the information related to single data centers. Beyond decentralization, a key feature of EcoMultiCloud is its modularity: provided that the interface between the lower and the upper layer is preserved, each layer is free to modify the respective algorithms and their implementation. At the lower layer, each data center is fully autonomous, and can manage the internal workload using either EcoCloud or any other consolidation algorithm. So different data centers can adopt different internal algorithms. On the other hand, the upper layer algorithms may be tuned or modified without causing any impact on the operation of single sites.

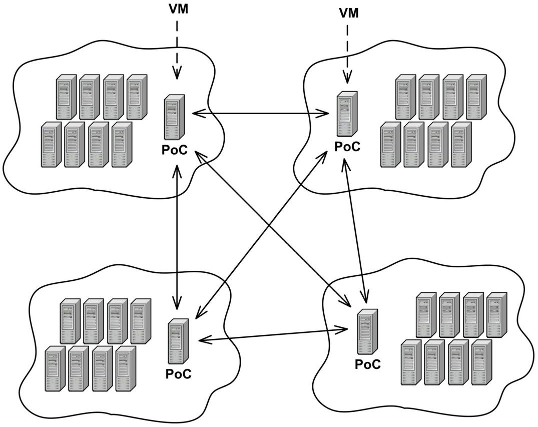


Fig. 1. EcoMultiCloud scenario: the PoCs of the different data centers exchange high level information about the state of local data centers. Such information is used, for example, to decide which site should accommodate a new VM.

The reference scenario is depicted in Figure 1, which shows four interconnected data centers. Each data center elects a single Point of Contact (PoC), a software that in the most typical case may be deployed on the same host as the manager of the local virtualization infrastructure, e.g., the vCenter in the case of VMware. The PoC integrates the information coming from the lower layer and uses it to implement the functionalities of the upper layer. The PoC is required to: (i) communicate with the local data center manager in order to acquire detailed knowledge about the current state of the local data center, for example regarding the usage of host resources and the state of running VMs; (ii) extract relevant high level information about the state of the data center; (iii) transmit/receive such high level information to/from all the other PoCs; (iv) execute the algorithms of the upper layer to combine the collected information and take decisions about the distribution of the workload among the data centers. For example, the assignment algorithm is used to decide to which data center a new VM should be assigned. Once the VM is delivered to the target site, this will use the lower layer algorithms to assign the VM to a specific host.

The framework is designed so that all the PoCs are able to execute the upper layer algorithms and, for example, choose the target DC for a VM originated locally. This requires an all-to-all data transmission among the PoCs, but this is not an issue due to the relatively low number of interconnected sites and the tiny amount of transmitted data. Indeed, in a multi-site scenario the choice of a single coordination point would be unappropriate for both technical and administrative reasons.

Since the single data centers are autonomous regarding the choice of the internal algorithms for workload management, this paper focuses on the algorithms of the upper layer. At least three algorithms must be executed at each PoC: (i) an assignment algorithm that determines the appropriate target data center for each new VM; (ii) a redistribution algorithm that periodically evaluates whether the current load balance is appropriate and, if necessary, decides whether an amount of workload should be migrated to/from another site; (iii) a migration algorithm that determines to which target site or from which source site the workload should be migrated.

The assignment algorithm is the core one: it has the primary role of distributing the workload of new VMs on the basis of a set of objectives decided by the management and pertaining to costs, consumed energy, carbon emissions, load balancing, etc. The other two algorithms are tailored to the dynamic redistribution of the workload. They share the same objectives of the assignment algorithm, but may take into account additional considerations, among which: the tolerance admitted on the achievement of the objectives, the limits on the frequency of migrations and on the amount of migrated data, the balance between benefits and costs related to migrations etc. This work focuses on the assignment algorithm and leaves the analysis of workload migration to future research.

3 Multi-site Assignment Algorithm

As mentioned in the previous section, a key responsibility of the PoC is to analyze detailed data about the local data center and summarize relevant information that is then transmitted to remote PoCs and used for the assignment and redistribution of workload. The nature of the high level information depends on the objectives that must be achieved. Some important goals are:

1. Reduction of costs. The cost associated to the execution of a given workload depends on many factors, among which the cost of power needed for computation, for cooling and for power distribution, the costs related to staff, servers maintenance, etc. An important element to consider is that the cost of electricity is generally different from site to site and also varies with time, even on a hour-to-hour basis, therefore the overall cost may be reduced by shifting portions of the workload to more convenient sites;
2. Reduction of consumed energy. The amount of consumed energy is generally easier to evaluate than the costs, as moderns data centers are equipped with sensors that monitor the power usage in computational resources. The total power may be obtained by multiplying the power consumed for computation by the PUE (Power Usage Efficiency) index;
3. Reduction of carbon emissions. Companies are today strongly encouraged to reduce the amount of carbon emissions, not only to compel to laws and rules, but also to advertise their green effort and attract customers that are increasingly careful about sustainability issues;
4. Quality of service. The workload must be distributed without overloading any single site, as this may affect the quality of the service perceived by users. Moreover,

quality of service may be improved by properly combining and assigning applications having different characteristics, for example, CPU-bound and RAM-bound applications;

5. Load balancing. In a multi-DC environment, especially if managed by the same organization, it may be important to balance the load distributed to the different sites. Among the rationales are: a better balance may help improve the responsiveness of the sites, decrease the impact on physical infrastructure – e.g., in terms of cooling and power distribution – help to prevent overload situations.
6. Inter-DC data transmission. In some cases it is more efficient to assign a VM to the local data center, instead of delivering it to a more convenient remote data center, depending on many factors, among which the amount of data used by the VM, the available inter-DC bandwidth and the type of applications hosted by the VMs. For example, choosing a local data center is more convenient in the case that the VM hosts a database server, much less if it runs a Web service, especially in the frequent case that Web services are replicated on several data centers.

The goals are not independent from each other: for example, overall costs depend on the consumed power, while a good load balance may help improve the quality of service. All the mentioned goals are important, yet different data centers may focus on different aspects, depending on the specific operating conditions and on the priorities prescribed by the management. The assignment algorithm described in the following is specifically devised for the case in which the two primary objectives are the reduction of overall carbon emissions and the load balancing. These two goals are chosen because they are representative of two opposite needs, the need for optimizing the overall efficiency and the need for guaranteeing the fairness among data centers. However, the assignment algorithm can be easily adapted to a different choice of the objectives.

For the described scenario, the PoC of each data center collects two kinds of information: the overall utilization of the data center resources, separately computed for each resource type (CPU, RAM, disk memory etc.) and the carbon footprint rate of the servers. The overall utilization of CPU is computed as the total amount of CPU utilized by servers divided by the CPU capacity of the entire data center. The same type of computation is done for the other hardware resources. The bottleneck resource for a data center is the one with the largest utilization. The carbon footprint rate of a server s, c_s, is measured in Tons/MWh [5]. The contribution of a server to carbon emissions is computed by multiplying the carbon footprint rate by the energy it consumes. The overall carbon footprint of a data center can then be approximated by summing the contributions of the servers and then multiplying the obtained quantity by the data center PUE, which allows the contribution of the physical infrastructure to be considered. When assigning a VM, the target data center should be chosen so as to minimize the incremental increase of the carbon footprint and at the same time keep/improve the load balance among the data centers. To this aim, a PoC does not need to know the carbon footprint rate of all the servers of remote sites: it only needs to know, per each site, the *best available carbon footprint rate*, i.e., the minimum rate among the servers that are available to host the VM. In fact, if the assignment algorithms of local sites share the same goals, the VM will be assigned to a server with that value of the carbon footprint rate.

Following these considerations, the assignment algorithm requires that the PoC of a server i transmits to the others three very simple pieces of data: (i) the utilization of the bottleneck resource – denoted as U_i, (ii) the best available carbon footprint rate of a local server, c_s, and (iii) the data center PUE. The last two parameters may be combined, and the carbon parameter C_i of a data center i is defined as:

$$C_i = PUE_i \cdot min\{c_s | \text{ } server\ s\ is\ available\} \tag{1}$$

In a data center many servers have the same characteristics, for example the servers included in the same rack/cluster. Therefore, the computation of C_i can be simplified by considering the carbon emission rate of each cluster instead of each server, and by evaluating the "*is available*" condition for entire clusters as well. Knowing the values of C_i and U_i for each remote data center, the PoC can now choose the best target data center for a VM. For each data center i, the function f^i_{assign} is computed as follows:

$$f^i_{assign} = \beta \cdot \frac{C_i}{C_{max}} + (1-\beta) \cdot \frac{U_i}{U_{max}} \tag{2}$$

In the expression, values of C_i are normalized with respect to the maximum value communicated by the data centers, and the same is done with U_i. The two mentioned goals – reduction of carbon emissions and load balancing – are weighted through a parameter β, having value between 0 and 1. After computing the values of f_{assign} for each data center, the VM is assigned to the data center having the lowest value. Depending on the value of β this may correspond to giving higher priority to the reduction of carbon emissions (values of β closer to 1) or to the fair balance of load (values of β closer to 0). Expression (2) can be easily generalized to the cases that more or different objectives are chosen.

```
function EcoMultiCloud-AssignmentAlgorithm($\beta$)
    while VM arrives
        for each remote datacenter $DC_i$
            Request $C_i, U_i$ parameters
        end for
        $C_{max} = Max\{C_i | \ i = 1 \cdots N_{DC}\}$
        $U_{max} = Max\{U_i | \ i = 1 \cdots N_{DC}\}$
        for each $DC_i$ : $DC_i$ is not full, that is, $U_i < U_{T_i}$
            $f^i_{assign} = \beta \cdot \frac{C_i}{C_{max}} + (1-\beta) \cdot \frac{U_i}{U_{max}}$
        end for
        $DC_{target} = DC_j$ such that $f^j_{assign} = min\{f^i_{assign} | \ i = 1 \cdots N_{DC}\}$
        Assign VM to $DC_{target}$
    end while
end function
```

Fig. 2. The EcoMultiCloud assignment algorithm

Figure 2 reports the pseudo-code used by a data center PoC to choose the target data center, among the N_{DC} data centers of the system, for a VM originated locally. First, the PoC requests the values of U_i and C_i to all the remote data centers[1]. Then, it computes the maximum values of both parameters, for the normalization, and computes the expression (2) for any data center that has some spare capacity, i.e., for which the utilization of the bottleneck resource does not exceed a given threshold U_{T_i}. Finally, the VM is assigned to the data center that has the lowest value of expression (2). Once consigned to the target data center, the VM will be allocated to a physical host using the local assignment algorithm. This paper does not focus specifically on the inter-DC migration of VMs. However the same algorithm, or a variant, may be used to determine the target data center of a VM that is being migrated.

4 Performance Evaluation of the Assignment Algorithm

To correctly evaluate the performance of the EcoMultiCloud assignment algorithm, it is necessary to prove that the hierarchical approach – according to which the VM is assigned in two steps, first to a target data center, then to a physical host – does not cause a performance degradation with respect to a single layer algorithm that has full visibility about all VMs and servers. Such a confirmation would suggest the choice of the hierarchical approach, which offers notable advantages in terms of scalability, autonomy of sites, overall administration, information management.

To this aim, we take as a reference a single level/centralized assignment algorithm presented and evaluated in [5]. The reference algorithm, called ECE (Energy and Carbon-Efficient VM Placement Algorithm) considers all the clusters of the distributed architecture and sorts them with respect to the value of (PUE x carbon footprint rate). Each VM is assigned to the available cluster with the minimum value and then, within that cluster, is assigned to the most power-efficient server. In [5] it is shown that this algorithm performs better than a number of other common heuristics based on the First Fit approach. Comparison is made to the hierarchical EcoMultiCloud approach, where the upper layer assignment algorithm is the one described in Section 3, while the lower layer algorithm is ECE, applied not the whole system but to single data centers.

For a fair comparison we consider the same scenario as [5], with four interconnected data centers having values of PUE and carbon footprint rate as reported in Table 1. Each data center includes two "rooms", a room A with newer and more efficient servers and a room B with older and less efficient servers. A data center has a single value of PUE, but two different values of carbon footprint rate for the two rooms.

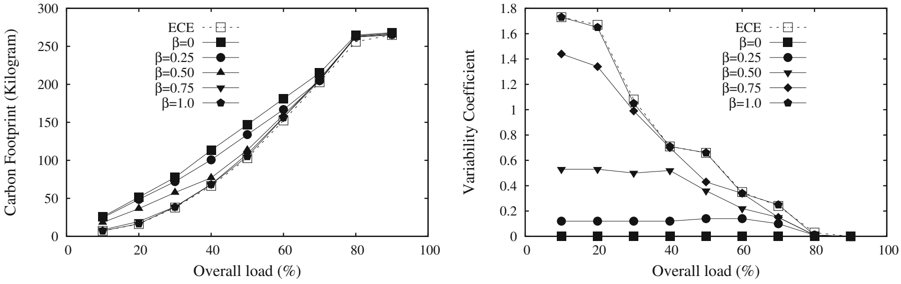
Data about VMs and physical hosts was taken from the logs of a real Proof of Concept performed by the company Eco4Cloud srl (www.eco4cloud.com), spin-off from the National Research Council of Italy, on the data center of a telecommunications operator. The data center contains 56 servers virtualized with the platform VMware vSphere 5.0. Among the servers, 38 are equipped with processor Xeon 24 cores and 100-GB RAM, and 18 with processor Xeon 16 cores and 64-GB RAM. All the servers have network adapters with bandwidth of 10 Gbps. The servers hosted 992 VMs which

[1] As an alternative, values can be transmitted periodically in a push fashion. In both cases the amount of transmitted information is tiny.

Table 1. PUE and carbon footprint rate of the four data centers in the considered scenario

Data center	PUE	Carbon footprint rate (Tons/MWh)	
		Room A	Room B
DC 1	1.56	0.124	0.147
DC 2	1.7	0.350	0.658
DC 3	1.9	0.466	0.782
DC 4	2.1	0.678	0.730

were assigned a number of virtual cores varying between 1 and 8 and an amount of RAM varying between 1 GB and 16 GB. Servers and VMs were replicated for all the rooms of the considered scenario, and only the values of PUE and carbon footprint rate were differentiated as described in Table 1. The most utilized resource in this scenario is the RAM, therefore the RAM utilization of data centers is considered when computing expression (2). A constraint imposed by the data center administrators was that the utilization of server resources cannot exceed 80%. The *overall load* of the entire system, used as a parameter in the evaluation, is defined as ratio between the total amount of RAM utilized by the VMs and the RAM capacity of the system.



(a)

(b)

Fig. 3. Total carbon footprint (a) and variability coefficient (b) vs. the overall load with different values of the parameter β

The performances have been analyzed with an event-based Java simulator that has been previously validated with respect to real data for the case of a single data center [8]. The VMs are assigned one by one executing the described assignment algorithm at the data center where each VM is located. Figure 3 shows the performance of the assignment algorithm versus the overall load of the system, when using values of β equal to 0 (such a value means that the load balance is the only goal), 0.25, 0.50, 0.75 and 1 (the only goal is the reduction of carbon footprint). The two plots report two indices: the total carbon footprint (a) and the variability coefficient (b). The latter is used as an index for the load balance and is computed by considering the RAM utilization of the four data centers and dividing the standard deviation by the average. We prefer the variability coefficient rather then the standard deviation because this helps to highlight relative rather than absolute deviations with respect to the average.

The performance of the single level ECE algorithm is also reported for comparison. As the two objectives are contrasting, a higher value of β allows the carbon footprint to be decreased, but at the expense of a greater load imbalance. It is interesting to notice that the values obtained with β=1 are almost equal to those obtained with the single level algorithm. This is consistent with the fact that the ECE algorithm does not consider the load balance as an objective. More importantly, this means that the hierarchical approach does not cause any performance degradation regarding the two considered metrics, which corresponds to the desired behavior, as said at the beginning of this section.

Depending on the management requirements, a proper value of β can be set accordingly. For example, if a constraint is given either on the admitted degree of load imbalance or on the overall carbon emissions, the value of β can be set so as to respect the constraint while optimizing the value of the other goal. This optimization analysis is left to future work. In the following, we analyze the behavior observed with specific values of β. Figures 4, 5 and 6 show the values of RAM utilization and carbon footprint rate for the single data centers, with values of β equal to 0, 0.5 and 1, respectively. From Figure 4 we can analyze the case where the only goal to be achieved is a fair load balance. Indeed, the servers are all utilized at the same level, whatever is the overall load, as seen in Figure 4(a), while Figure 4(b) shows that the carbon footprint is proportional to the amount of emissions of the single data centers, as detailed in Table 1. When the two objectives must be balanced (case with β=0.5), the data centers are loaded with different rates, as shown in Figure 5(a): for example, the most efficient data center, DC 1, is loaded more rapidly than the others, while the data center DC 4 is fully loaded after the others are already utilized at the maximum allowed level.

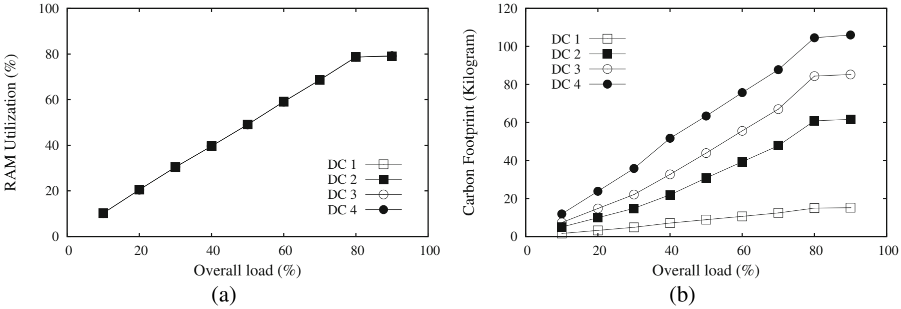
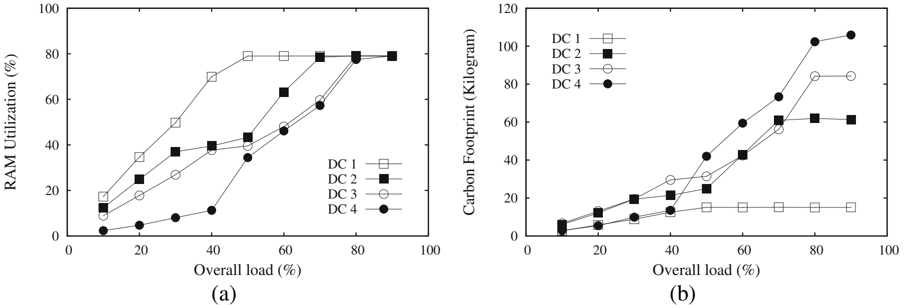


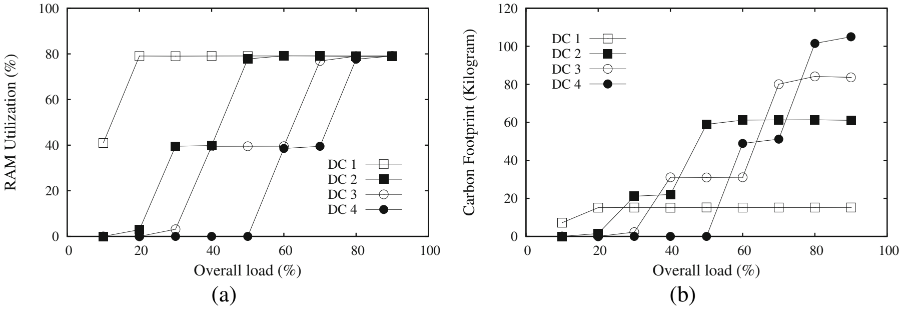
Fig. 4. RAM utilization (a) and carbon footprint (b) of the four data centers vs. the overall load with β=0

Figure 6 focuses on the case in which we are only interested is the reduction of carbon emissions. The two plots show that the data centers are loaded in an order that corresponds to their efficiency. Even more, we can notice that the order is respected also with reference to single data center rooms. For example, after DC 1 is fully loaded, the VMs are first assigned to Room A of DC 2, than to Room A of DC 3, then to Room B of DC 2, and so on. It may be easily verified that this order follows the values of (PUE x carbon footprint rates) of the different rooms, as reported in Table 1. It is noticed



(a) (b)

Fig. 5. RAM utilization (a) and carbon footprint (b) of the four data centers vs. the overall load with β=0.5



(a) (b)

Fig. 6. RAM utilization (a) and carbon footprint (b) of the four data centers vs. the overall load with β=1

that the curves of carbon footprint, in Figure 6(b), intersect among each other: with a low overall load, carbon footprint emissions are larger in more efficient data centers, because these are the first to be loaded; with a high overall load, all the data centers are highly utilized and the least efficient are those that cause the highest carbon emissions.

5 Conclusions

This paper has presented EcoMultiCloud, a hierarchical approach that aims to improve the workload management of a multi-site data center. The related architecture comprises two layers, the upper layer for the assignment/migration of workload among remote sites, and the lower layer that assigns Virtual Machines to physical hosts within every local site. The approach is flexible and can be utilized to achieve and balance different goals, among which reductions of costs/consumed energy/carbon emissions, load balancing, etc. The paper has focused on the analysis of a four-site system in the case that the goals to be achieved are the reduction of carbon emissions and the load balancing among data centers. Performance analysis has proven that the hierarchical approach achieves nearly the same quantitative results as a reference centralized solution, but offers better functionalities in terms of flexibility – it can be adapted to the specific goals specified by the management – and autonomy of single data centers, as they are free to adopt any internal algorithm for workload management.

References

1. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Generation Computer Systems 28(5), 755–768 (2012)
2. Doyle, J., Shorten, R., O'Mahony, D.: Stratus: Load balancing the cloud for carbon emissions control. IEEE Transactions on Cloud Computing 1(1), 116–128 (2013)
3. Feller, E., Rilling, L., Morin, C.: Snooze: A scalable and autonomic virtual machine management framework for private clouds. In: Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012), pp. 482–489 (May 2012)
4. Goiri, I., Le, K., Guitart, J., Torres, J., Bianchini, R.: Intelligent placement of datacenters for internet services. In: 31st International Conference onDistributed Computing Systems (ICDCS), Minneapolis, Minnesota, USA, pp. 131–142 (June 2011)
5. Khosravi, A., Garg, S.K., Buyya, R.: Energy and carbon-efficient placement of virtual machines in distributed cloud data centers. In: Wolf, F., Mohr, B., an Mey, D. (eds.) Euro-Par 2013. LNCS, vol. 8097, pp. 317–328. Springer, Heidelberg (2013)
6. Koomey, J.: Growth in data center electricity use 2005 to 2010. Tech. rep., Analytics Press, Oakland, CA (August 2011)
7. Li, W., Svärd, P., Tordsson, J., Elmroth, E.: Cost-optimal cloud service placement under dynamic pricing schemes. In: 6th IEEE/ACM International Conference on Utility and Cloud Computing, Dresden, Germany, pp. 187–194 (2013)
8. Mastroianni, C., Meo, M., Papuzzo, G.: Probabilistic consolidation of virtual machines in self-organizing cloud data centers. IEEE Transactions on Cloud Computing 1(2), 215–228 (2013)
9. Mehta, D., OSullivan, B., Simonis, H.: Energy cost management for geographically distributed data centres under time-variable demands and energy prices. In: 6th IEEE/ACM International Conference on Utility and Cloud Computing, Dresden, Germany, pp. 26–33 (December 2013)
10. Ren, S., He, Y., Xu, F.: Provably-efficient job scheduling for energy and fairness in geographically distributed data centers. In: 2012 IEEE 32nd International Conference on Distributed Computing Systems (ICDCS), pp. 22–31 (June 2012)