

Green Metascheduler Architecture to Provide QoS in Cloud Computing

Oswaldo Adilson de Carvalho Junior

Federal University of Alfenas, Institute of Science and Technology

Poços de Caldas-MG, Brazil, 37715-400

University of São Paulo, Institute of Mathematics and Computer Sciences

São Carlos-SP, Brazil, 13566-590

osvaldo.carvalho@unifal-mg.edu.br

and

Sarita Mazzini Bruschi

Regina H. Carlucci Santana

Marcos José Santana

University of São Paulo, Institute of Mathematics and Computer Sciences

São Carlos-SP, Brazil, 13566-590

{osvaldo,sarita,rca,mjs}@icmc.usp.br

Abstract

The aim of this paper is to propose and evaluate GreenMACC (Green Metascheduler Architecture to Provide QoS in Cloud Computing), an extension of the MACC architecture (Metascheduler Architecture to provide QoS in Cloud Computing) which uses greenIT techniques to provide Quality of Service. The paper provides an evaluation of the performance of the policies in the four stages of scheduling focused on energy consumption and average response time. The results presented confirm the consistency of the proposal as it controls energy consumption and the quality of services requested by different users of a large-scale private cloud.

Keywords: Green Computing, Cloud Computing, Scheduling, Performance Evaluation, Simulation

1 Introduction

Green Computing is an approach that has been studied in recent years and aims at promoting the use of Information technology (IT) with a fair and legitimate concern regarding environmental issues. The main focus is on saving resources, where in this area, energy is the most relevant factor.

Current climate change concerns have driven people to think *âgreenâ* and it is widely recognized that part of the energy being consumed by Data Centers can be reduced, transforming this into a fundamental issue. This energy saving can be achieved while servers are not in use [1]. Green metascheduling is included in this context as it can help in the choice of Data Centers and also the adequate allocation of Virtual Machines (VMs) to hosts. Even in cases where companies are not concerned about a sustainable planet, nor have a correct environmental view, energy reduction, and consequently CO₂ reduction, offers another advantage: a reduction in operational costs. By extending the MACC [2], the aim of this work is to enable any company that decides to use it to work with green computing in a private cloud so that quality of service can be offered to its users. As the architecture already has all the features to offer adequate QoS for clouding, it can be used and extended for a Green IT panorama, resulting in the GreenMACC proposal.

Current work found in the literature show that specific issues are aimed at efficient green scheduling. These issues, which in this work are called Decision Analysis Points for Green Scheduling (PADEVE - Pontos de Análise para Decisão no Escalonamento Verde in Portuguese) are: Processor, Network, Refrigeration and CO₂ Emission. The literature shows that the most commonly studied PADEVE for Green Meta-scheduling

is the processor, where one of the most used techniques for energy saving is DVFS (Dynamic Voltage and Frequency Scaling) [3] [4].

In 2009, an architecture called Green Cloud [3] was proposed, using scheduling that is also based on energy saving using the DVFS. In this work, first we propose the workload distribution and then turning off those hosts which are not being used. On the other hand, the scheduling proposed and other paper is aimed at turning off the under-used hosts, the migration of VMs and using DVFS [4]. Other work focuses exclusively on the processor, concerned mainly about the distribution of VMs aiming at overloading the processing of some hosts in order to release others and in this way turn them off, or even calculating the average processing time needed for the task to be executed in a host and based on that information carry out the scheduling turning off the hosts not being used [5]. One of the main concerns when the PADEVE processor is being considered for energy saving is to attempt to reduce the maximum number of processors being used. Different techniques can be used to achieve this, ranging from using neural networks [1] to using migration of virtual machines [6] [7] [8] [9].

Another commonly used PADEVE for Green scheduling is the network. There are various studies available in the literature proposing the use of migration in order to reduce energy consumption [8] [4]. Other work is concerned with migration, but avoiding its excess, achieving more energy saving than the others [6] [7] [9].

Some studies that use refrigeration as the PADEVE are concerned about the energy consumed by air-conditioners in Data Centers [10]. Other work focuses on reducing the rotation of the processor coolers [4]. A third concern is aimed at reducing the temperature generated by the processor [6]. All evaluated areas show a reduction in energy consumption.

Other work available in the literature focuses mainly on carbon gas (CO₂) emission. For instance, a work that proposes scheduling to analyze different points such as: processor consumption, air-conditioner consumption, energy cost and CO₂ emission [10]. The main idea behind the analysis of this PADEVE is to provide the meta-scheduler with a coefficient of the Carbon gas emission of Data Centers (made available by the US Environment Agency) and from this data take a decision for scheduling [10] [11]. The GreenMACC offers the possibility to automatically choose a policy setup in the four scheduling stages. The four stages are cited: Data Center choice, creation of the virtual machine to the hosts, allocation of the virtual machine to the hosts and allocation of the service to the virtual machine. These will be discussed in more details in the Section 3.

In addition to the Introduction, this work is organized in four sections as follows: The MACC architecture is defined in Section 2, which was used as a basis for the proposal of the GreenMACC architecture. Section 3 presents the proposed architecture, as well as its dataflow. Moreover, it includes the policies included in the four stages of scheduling. Finally, it presents a qualitative comparison between GreenMACC and other architectures. An evaluation of the architecture's performance is provided in Section 4, as well as the methodology used and the results obtained. In the last section, conclusions of this work are drawn.

2 MACC - Metascheduler Architecture to provide QoS in Cloud Computing

The work where MACC is presented [2] shows a general view of its architecture. The highest layer is used for interacting with the user, where requests are entered and services are offered. The next layer entails controlling the flow of services using two components: admission control and value control. The third layer is where the main operations of the meta-scheduler are managed. In the fourth layer, the hypervisor can be found, for example Xen or VMWare. Finally, the last layer is the Physical Infrastructure one. The interlayer components communicate among themselves by exchanging information aimed at offering the requested service. The execution sequence is initiated when the request is received from the user (client) containing the criteria of QoS defined. The admission control receives the request and verifies the availability of resources in the MDSM (Monitoring and Discovery System Manager). In the affirmative case, the request is sent to the value control in order to negotiate the price for the requested service. If the agreement is sealed, the request is forwarded to the meta-scheduler's core for it to define the policies for the allocation and creation of Virtual Machines (MVs), as well as the way the service will be presented. Finally, when the service is concluded, the results are sent to the client.

The MACC was designed based on Federations; that is, clouds offering the same service and, although they belong to different companies, they communicate by interacting among themselves and when one of them realizes that it does not have enough resources, it sends the client's service request to another cloud for it to process. Another concern of this meta-scheduler is that its negotiation and scheduling are focused on the cost of the service executed. The following section presents the GreenMACC which has the MACC as a basis of its architecture, though with a different focus. Instead of an Intercloud with scheduling concerned with costs, the proposed architecture enables green policies to be implemented in a private cloud.

3 GreenMACC - Green Metascheduler Architecture to provide QoS in Cloud Computing

GreenMacc is an extension of MACC and enables users to make use of the services available in the cloud and, in a transparent way, uses green scheduling policies; that is, it is concerned with energy consumption and CO₂ reduction. Its data flow is different from others developed to public clouds since it was designed aimed at a private cloud where the company's interest in energy reduction is more evident due to the reduction in electricity bills. The three following sub-sections provide more details about the data flow of the proposed architecture, explaining clearly how the four stages of scheduling are carried out and present a qualitative comparison with other architecture proposals found in the literature.

3.1 Architecture and Dataflow

Some differences can be observed between MACC and GreenMACC. The modified architecture can be seen on the left side of Figure 1 and the name of the new components or modified components it's written in red.

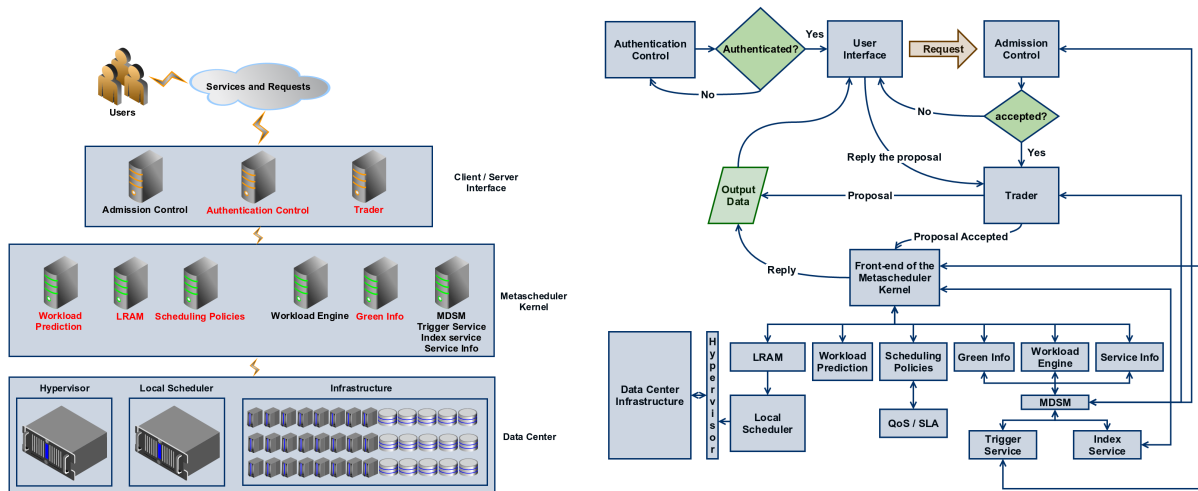


Figure 1: General View of the GreenMACC and GreenMACC Flowchart [12]

No changes were made to the first layer of services and requests, however in the second layer, the value control was removed because the proposed architecture is intended for a private cloud, where no negotiation policies are needed for charging users. Instead, Authentication Control and Trader were introduced. The former is responsible for the user authentication which can be performed by login or another common method for automatic validation. The latter is responsible for determining the performance needs a user has in order to achieve the desired QoS and then, depending on the results, save energy. The decision may be either automatic, depending on the kind of user (manager, analyst, typist, and so on), or by allowing the user to choose among available possibilities where there is an interaction between the performance and energy consumption.

In the meta-scheduler core layer, changes were made to the scheduling policy module in order to implement green policies in addition to those solely concerned with QoS. For this, two new components were included: the first one is responsible for keeping and updating all information needed for the use of green meta-scheduling techniques (Green Info), such as: specific consumption of each existing processor in the Data Center, use of DVFS, status of all hosts (on, off, standby), a processor's temperature, a Data Center's CO₂ emission coefficient, and so on; the second one is the Workload Prediction, which is responsible for the data necessary for foreseeing the workload of each host or processor core, needed for taking decisions in some policies for migrating virtual machines and turning hosts off. Yet, in the GreenMACC core, a change to the LRAM is presented; unlike the MACC, where it only provided the necessary information for decision taking, in the GreenMACC it helps in decision making, choosing which scheduling policy should be used at each of the 4 stages: Data Center choice, creation of the virtual machine, allocation of the virtual machine and allocation of the service to the virtual machine. These will be discussed in more details in the following subsections. This decision making can be done by observing different factors such as the user's profile and history or, alternatively, analysing the current workload imposed to the cloud. In this work, the second option was used. To better understand how a service request is processed and met, Figure 1 shows, on the

right side, the dataflow diagram for the whole architecture.

The process begins with the user's authentication. Then, using the user's interface, the user's service request is created. A request is then sent to the Admission Control, who in turn, checks the MDSM (Monitoring and Discovery System Manager) concerning the availability of resources and services. The MDSM checks all information concerned with the resources in the Service Index, regarding the available services in the Service Information module and the time needed for executing the requested service in the Work Load Manager. At that point, the MSDM sends all this information to the Admission Control, responsible for deciding whether the request should be accepted or refused. If there are enough resources and the requested service is available, the request is sent to the Trader module. The Trader module is responsible for negotiating with the user a way through which the service can be met while, at the same time, making it possible to obtain a reduction in the energy consumption. This process is called Green Negotiation. For this, the Trader module consults the MDSM, who in turn, consults the Green Information module about the PADEVEs. The latter delivers the data to the MSDM which are immediately sent to the Trader module. Once the negotiation with the user is sealed, which can be carried out automatically or not, the information concerning the Service Level Agreement (SLA) and the QoS are stored in a specific place to be used by the scheduling policies. At this point, the request is sent to the meta-scheduler's core that is in charge of managing all information stored in the components. The core sends all information needed to the LRAM, where all these data are analyzed and the decision is taken as to which policies implemented at the meta-scheduler will be used at each of the four scheduling stages. Once the choice is made, the information is passed to the local scheduler where the scheduling stages are managed. Once the service processing is finished, the response is sent to the user.

For the sake of understanding how the whole process works, from the moment when the user requests the service until the response of the service, two sequence diagrams are presented. The first one can be seen on the left side of Figure 2, where the whole process is shown. In this diagram, it is considered that all consultations for availability of resources and services were answered affirmatively and also that the green negotiation with the client was successful. In this diagram, the internal process of the meta-scheduler's core is omitted in order to offer a general view of the response time to the service request. The meta-scheduler's detailed internal process is presented in the diagram on the right side of Figure 2.

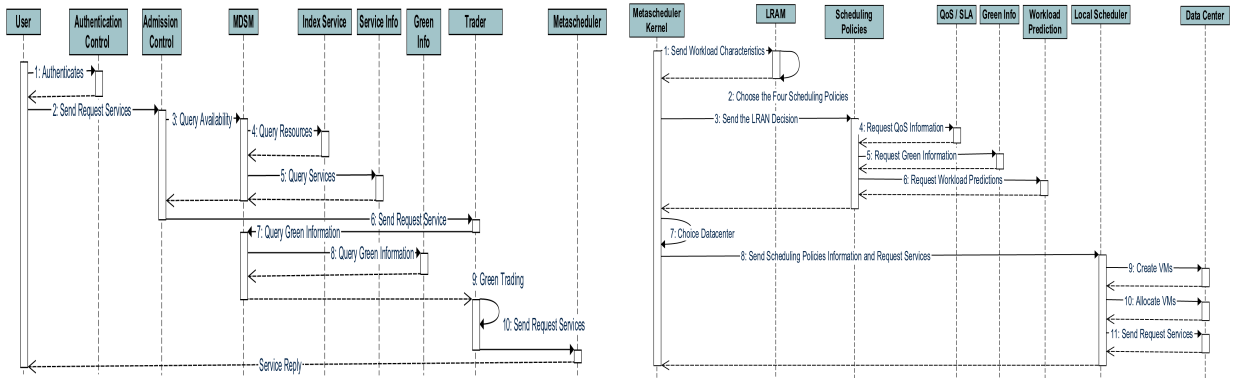


Figure 2: Sequence Diagrams
[12]

The diagram on the left side of Figure 2 shows how transparent the process is to the user. After authentication, the user sends the service request when all verifications are carried out without the user noticing the process. Afterwards, the user receives proposals with possible offers for the service. After choosing the one that better suits his/her needs, the user waits for the service response. This process can become even more transparent if the implementation of the negotiation is automatized according to the user's characteristics or history. Taking this into account, after the authentication, the user sends the request and is not concerned about the negotiation involving energy saving and the SLA. The diagram on the right side of Figure 2 shows the sequence of actions that take place at the meta-scheduler's core. This process is also transparent to the user. When the request reaches the processor's core, the first step is to send the characteristics of the workload and, if necessary, also the characteristics of the user to the LRAM, who in turn, will choose the best policy for each scheduling stage for that specific case. After choosing the policies, the LRAM sends the name of the chosen policies which are triggered immediately. The chosen policies search for all the information needed to execute it in the QoS/SLA modules, Green information and Workload Forecast. The first policy to be carried out is the choice of datacenter. Then, the information and

service requests are sent to the local scheduling which creates the virtual machines. After they are created, the virtual machines are allocated to the hosts of the chosen datacenter and finally the required services are distributed to the VMs to be executed.

As mentioned, the GreenMACC scheduling process has 4 stages: Choice of Data Center, Creation of Virtual Machine, Allocation of Virtual Machine and Allocation of Service to the Virtual Machine. The following subsections provide a detailed explanation for each of these, as well as the policies implemented in each of them.

3.2 Policies for Choosing the Data Center

The policies of this stage are responsible for making decisions in the choice of the Data Center that will receive the user's request and execute the service. In order to evaluate the proposed architecture of this work, two policies for the choice of the Data Center were evaluated: Round Robin and Network Capacity Based (NCB). Although these policies were not initially designed for Green Computing, they were chosen aiming to understand better their behavior in an architecture aimed at this target.

The first policy used during this stage, the Round Robin (RR), carries out the scheduling in a way in which requests are distributed to the Data Centers one by one, following a Singly Circular Linked List. The main advantage offered by this policy is that it avoids overloading the meta-scheduler, since there is no decision making that requires intensive processing. However, the Data Center chosen may not be the ideal one, and a situation may be possible where either the service is forwarded to a Data Center that is already overloaded or the hosts are being under-used, resulting in a negative situation creating a situation that can have a negative effect on the energy consumption. The second policy used at this stage of scheduling is the Network Capacity Based (NCB). This policy uses the network's information, in this case the latency, as the fundamental criteria for taking the decision as to which Data Center the user's service request should be sent to. This policy will always choose the lowest latency value available in the network. The advantage of this policy is that it will always choose the best option, however the metascheduler will have a high overload since all Data Centers will be consulted before taking the decision. After choosing the Data Center, the next step is concerned with choosing how the virtual machines will be created in the chosen Data Center. For this, policies are available for the creation of VMs.

3.3 Virtual Machine Creation Policies

After choosing the Data Center, the process moves on to the second stage for scheduling the Meta-scheduler, the creation of Virtual Machines. This work evaluates two policies for creation, both are from the MACC and are presented in this section.

The two chosen policies are SD2c (Slotted Dynamic 2 vCores) and SD4c (Slotted Dynamic 4 vCores). The main difference between them is the number of vCPUs (virtual CPUs) created at each Virtual Machine (VM). Both policies follow 3 phases, where the first one refers to the number of VMs created and this number depends on the demand presented by the client. The second phase is characterized for being the only one where the policies are different, since it is at this phase that the number of vCPUs for each VM created is defined. For policy SD2, two vCPUs are defined, whereas for the SD4c policy, there are four. For both, the vCPUs' computational potency is fixed. The third and last phase entails deciding the number of cores for the Data Center's physical host that will be allocated to each vCPU of the created Virtual Machine. This number varies between 2, 4 and 8 cores, according to the services demand required. The choice of these two policies had the same purpose of the previous one; that is, understanding the behavior of policies designed with other purposes when used in an architecture aimed at policies focused on Green Computing. After creating the VMs, the next step is to allocate them to the hosts at the chosen Data Center, thus moving on to the next stage.

3.4 Virtual Machine Allocation Policies

The third stage of the metascheduling entails allocating virtual machines created in the previous stage to the physical hosts of the Data Center. Another task of this phase is to manage the migrations of these VMs when the implemented scheduling policy has this feature. In this article, two policies were used. One having static allocation features, i.e. without migrating to virtual machines and the other policy having dynamic features. The latter uses the migration technique. Both policies were presented in studies that aim to evaluate policies concerned with reducing energy consumption [7] [13]. This choice was made to better understand the behavior of GreenMACC when policies are used that employ virtual machine migration techniques and also policies that do not use these techniques. Another objective of this choice was to prove that the proposed architecture could blend policies of very distinct features in all the scheduling stages.

The policy without migration allocates virtual machines in a static way to the hosts following a simple list. Whereas the policy with migration, Static Threshold, has a fixed limit of Service Level Agreements (SLA). This limit of the agreement is a determining factor when taking decisions of migrating one virtual machine. After allocating the virtual machines to the hosts, it can be finally decided how to distribute the required services.

3.5 Service Allocation Policies

The objective of the last stage of scheduling is to choose the way to allocate services to the virtual machines which have already been allocated in the previous stage, i.e. all the tasks concerning the required services are allocated to the virtual machines. For this task, two scheduling techniques were used: Time-share and Space-share. The two policies were already used in work which was concerned about saving energy [6][7] [13], despite not having been developed for this purpose exclusively. It was for this reason that both were chosen for this level of scheduling.

The Time-share policy can allocate more than one service to a virtual machine, not creating queues of services awaiting allocation. Whereas the Space-share policy is characterized as allocating only one service for each VM. In the latter case, a queue of services is created to be allocated to the VMs created in the third stage of scheduling of the GreenMACC.

3.6 Qualitative Comparison of GreenMACC with Other Architectures

In the literature, there are various studies on the topic of this article. The aim of this section is to make a qualitative comparison, using a table where a distinction between the proposal of this article and other related work can be made. In Table 1, the architectures evaluated in relation to six characteristics can be observed: control mechanisms of multiple Data Centers, user controls, energy consumption (kWs), carbon dioxide emission (CO₂), the Quality of Service (QoS) and finally the automatized choice of policies in the four stages of scheduling (Auto).

The differences between the architecture proposals for cloud scheduling can be observed in Table 1. The MACC [2], an architecture that was used as a basis to develop GreenMACC, manages to work with multiple Data Centers aiming at the Quality of Service. However, there are no specific modules which can implement green policies. The MACC does not have authentication and user control, whereas the GreenMACC does, which enables it to make automatic choices of which policies can be executed according to the user profile. This automatized choice can be also made according to the number of services imposed to the GreenMACC.

Table 1: Comparison of related work [14]

Architecture	Multiples DCs	Users	Kws	CO ₂	QoS	Auto
GreenMACC	X	X	X	X	X	X
MACC[2]	X				X	
CAGCA[11]	X		X	X	X	
GreenCloud[3]			X		X	
GCA[9]		X	X		X	
2 Levels[15]	X				X	
HICCAM[16]	X	X			X	
UMATGC2[17]	X	X			X	

As well as the MACC and the GreenMACC, the Carbon Aware Green Cloud Architecture (CAGCA) [11] can also manage multiple Data Centers. The CAGCA presents a green architecture which can allow green scheduling aiming at energy consumption and carbon dioxide emission, as well as being concerned about the QoS. However, it does not have authentication and user control, nor the capacity of choosing multiple policies in the four stages of cloud scheduling.

The GreenCloud [3] was not designed for multiple datacenters and also does not have user control. However, its architecture prioritizes scheduling aiming at saving energy and QoS without being concerned about CO₂ emission. The Green Cloud Architecture (GCO) [9] offers user control and techniques to save energy and provide QoS, although it does not manage multiple datacenters and, consequently, does not offer automatization when choosing policies in the four stages of scheduling. Its policies are also not concerned about carbon dioxide emission.

In 2009, a meta-scheduler that offers two levels of scheduling [15] was proposed. In the same way as MACC, the first level helps in choosing the Data Center, whereas the second one in allocating resources at the chosen Data Center. The meta-scheduler in two levels proposed in the cited work is able to work

with multiple datacenters and can implement policies at this level of scheduling. Similar to the MACC, it is also concerned about the QoS offered to the user, however in the work presented no user control model was cited or one that allows having control over the energy consumption data and carbon dioxide emission. The two-level meta-scheduler does also not offer the possibility of automatically choosing the various scheduling policies for both levels.

The Hybrid Cloud Construction and Management (HICCAM) architecture [16] has a model responsible for managing users which allows its authentication and offers strict control for using the services offered by the cloud. However, there is no concern about green computing, nor an automatized system which can offer an option menu of techniques and scheduling.

The most recent work in this comparison with GreenMACC is the UMATGC2 metascheduling architecture [17]. This meta-scheduler has very similar features to HICCAM as it offers cloud user control, it makes scheduling possible at a level above the datacenters and prioritizes policies that exclusively aim at the QoS offered to its users. Managing various scheduling techniques does not exist and there is no structure in the architecture which offers the possibility of managing energy consumption or carbon dioxide emission.

According to what has been described in this section, it can be concluded that the GreenMACC is a complete metascheduling architecture and it can manage all the users through the Authentication module. The architecture in this article proposes implementing the policies at the four stages. It also proposes that various policies in each one of the stages can be implemented and the best policies for each one of the stages can be chosen automatically with the purpose of negotiating with the user in an optimized way. The GreenMACC can even implement policies which provide QoS and green policies altogether.

4 Evaluation of Architecture

In this section, the methodology used to evaluate the architecture, as well as the results of this work are presented.

4.1 Methodology

To evaluate the proposed architecture, a simulation technique was used in this work. The tool which was used was the CloudSim [18] [19] in version 3.0 which has been used in various other work in the cloud computing area [1] [15] [7].

The model used for the simulations has the following features: 15 Data Centers with 1,000 hosts each. The hosts can have two, four or six cores equally divided into each Data Center. The fixed features of the hosts are the 16GB RAM memory and 1 Gbit/s Bandwidth. The simulations are done considering the execution of 24 hours in a private cloud, with Data Centers all around the world on the five continents. All the Data Centers offer the same service. The stored data, as well as the applications and services are not geographically distant, i.e. the data is reproduced in all the Data Centers. All the services offered are applications with CPU-Bound features. This reproduction data model in all the Data Centers can be used by companies which offer the same services in all of its Data Centers. As an example, a bank where the services offered in the cloud are the same regardless of its geographical location. Implementing GreenMACC in the CloudSim was done by creating classes, whereby each class has all the attributes and methods necessary to function exactly as a specific module of the architecture presented in this work.

After the GreenMACC has been implemented in the CloudSim, the policies of the four stages of scheduling and of the meta-scheduler are carried out and evaluated. In total, there are eight policies, two for each one of the stages. For this evaluation, five factors were used (four scheduling stages and the workload). For each factor, two levels were used: in the scheduling stages, they are the policies themselves and for the workload the light and heavy levels were used. The light workload was modeled on 30 users, having requests of 500 services per user and little processing. The heavy workload had 60 users, having requests of 1,000 services per user and a high processing demand. The service request is done automatically by the abovementioned users. The factors and levels are presented in Table 2.

Table 2: Factors and Levels UsedThe service request is done automatically by the abovementioned users. in the Evaluation

Factors	Levels
Choice of Data Center	RR e BCR
Creation of Virtual Machine	SD2c e SD4c
Allocation of Virtual Machine	Without Migration and With Migration
Allocation of Services	Space-share e Time-share
Workload	Light and Heavy

The metrics considered in the GreenMACC are the Average Response Time (ART) which considers the network time, queuing time and service time, as well as energy consumption in kiloWatts/second (kW/s). To make the proposed evaluation, each scenario was repeated 10 times, each one with a seed of generation of different random numbers so that subsequently the average and confidence interval can be calculated. This is necessary because there is some randomness in the request arrival rate. The complete factorial way using all the factors and levels was applied, i.e. all the possible scenarios were tested. In the next section, the results obtained from the simulations of the scenarios, described previously, are presented.

4.2 Results

In this section, the results of the simulations described in the previous section are presented and discussed. First of all, the main effects on the average response time and on the energy consumption are presented when one of the five factors evaluated in this work is altered. Afterwards, an evaluation is made of the results obtained from a comparison between different scenarios analysing the importance of choosing the correct policy from each one of the scheduling stages in relation to the workload used.

4.2.1 Effects caused by different workloads

The energy consumption with a light workload can be observed in the graphs on the left in Figure 3. The SD2c policy consumed a little more energy compared to the SD4c policy when services that require a little workload are used. This occurred due to the SD2c policy having less vCPUs than the SD4c, which overloaded the system increasing the processor clock and thus increasing the energy consumption. The change in policies for choosing a datacenter did not influence the energy consumption with a light workload. This is because of the large number of datacenters in the infrastructure offered for a workload that does not require a huge effort. The policy of allocating without migration with a light workload, showed not only a lower ART than the one with migration, but also lower consumption. This took place for the same reason described previously, i.e. the cost of VM migration for a light service workload is very high requiring much work from the network and processing, which demands more energy consumption. Concerning the ART, in the fourth stage, the space-share scheduling policy proved to be better than the Time-share, however regarding energy consumption it was less economical. This is due to the fact that the Timeshare policy does not use process queues, i.e. it allocates all the services to the virtual machines at the same time, which have been created and already allocated. Although it requires more response time, the energy consumption remains more linear. Concerning the Space-share policy, there are high peaks of processing which causes more energy consumption.

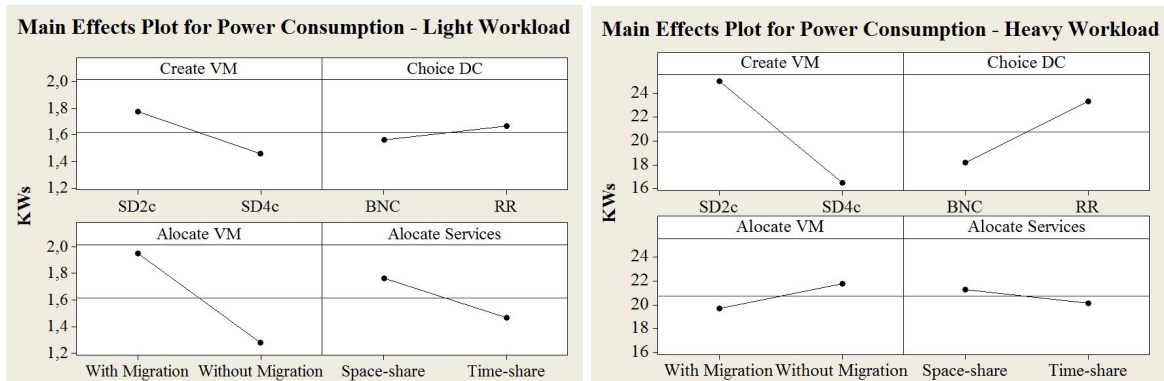


Figure 3: Effects on the energy consumption with light and heavy workloads

Some conclusions can be drawn from the graphs on the right in Figure 3 regarding the influence of scheduling policies in energy consumption with services that have a heavy workload. The first observation can be made concerning the increase in energy consumption in all the cases due to the increase in the workload. The SD2c policy, which had a light workload, already had less consumption than the SD4c. After increasing the workload, the better efficiency of the first was clearer than that of the second policy evaluated. In the stage of choosing the datacenter, the BCR policy, despite having a higher ART, consumed significantly less energy than the Round Robin. The RR policy uses all the datacenters of the cloud which enables it to obtain a lower ART. On the other hand, the energy consumption tends to be higher than the BCR as the second one reduces the number of active datacenters significantly with hosts which are turned on. In the third stage, it can be observed that the policy with migration, apart from having less ART than the one

without migration, also proved to be more efficient in relation to energy consumption. From the moment that the workload was increased the VM migration was worth it, contrary to when a light workload was used in the system. As there was an increase in the workload, the Time-share policy remained as a more economical policy in the fourth stage for the same reason as when a light workload was used.

In Figure 4, it can be observed from the graphs on the left that the policies evaluated in the stages of the Choice of Data Center (Chooses DC) and the creation of Virtual Machines (Creates VM) did not significantly influence the Average Response Time with a light workload. This is due to the fact that the infrastructure offered (15DCs with 1,000 hosts each) is much higher than the necessary to execute the workload required. In the Virtual Machine Allocation (Allocates VM) stage, the policy without migration has a lower ART than the policy with migration when a light workload is used due to the high cost that the migration of VMs requires from the system as a whole. From the four stages evaluated, the changes in the Service Allocation (Allocates Services) policy are those that most influence the ART. The Space-share policy is much more efficient than the Timeshare. This fact is due to the Timeshare policy allocating all the virtual machines at once to the processing cores without having queues. This causes a high occurrence of the context switch in the processor making the system slower. On the other hand, this does not happen in the Space-share policy as VMs are allocated one at a time, and thus does not overload the processing units.

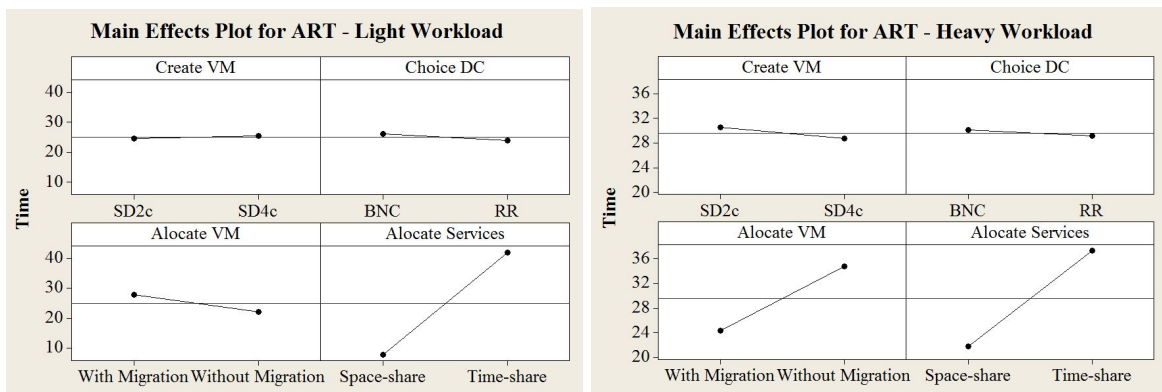


Figure 4: Effects on the ART with light and heavy workloads

In the graphs on the right, Figure 4 shows the results of the ART with a heavy workload. As the workload is increased, the times increase considerably as expected and the efficiency of the policies are clearer. In the VM creation stage, the SD4c policy is a little more efficient than the SD2c and the RR presented slightly less times in relation to BCR. The fact that SD4c has more vCPUS than the SD2c influenced the ART as expected. As the RR policy does not have any pre-processing to take decisions in the scheduling, it obtained less times than the BCR. In the VM allocation stage, an inversion in relation to the light workload could be observed. Concerning the heavy workload, the policy that used the migration technique showed it had less time than the policy with VM static allocation features. The explanation for this fact is the cost of migrating a VM from one host to another within a Data Center. Regarding the light workload, the VM migration is not worth it due to the time spent by the processing migration in relation to the ART. In the last stage of scheduling, the Space-share policy proved to be the most efficient for the same reason already described earlier.

4.2.2 Efficiency of LRAM

Based on the results presented in this work, a comparison could be made of scenarios (Table 3) where the importance of the LRAM module making a good choice of the policies for each one of the scheduling stages could be observed. In order to understand better how the LRAM takes decisions when choosing policies, there are two graphs in Figure 5. The first refers to the results obtained with a light workload and the second presents values obtained using a heavy workload. In both graphs, on the x axis the values concerning the energy consumption of all the cases possible between the policies of the four stages are shown. On the y axis, the values concerning the ART can be found. The graphs are divided into 4 squares. The lines that separate the squares are defined according to two different criteria, one for each line. The parallel line on the x axis specifies the maximum limit of the deadline established in the user's contract. The parallel line on the y axis is defined by the value of the average of energy consumption values obtained using the specific workload.

In the following algorithm, it can be observed how the LRAM is implemented to always obtain the best case possible. First, it verifies if there are any results in square 1, if there are, it should take the least energy

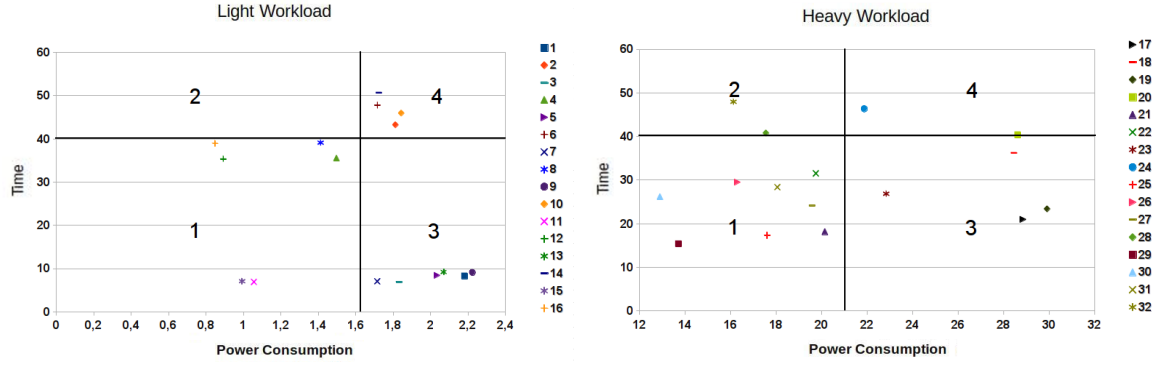


Figure 5: Values analysed by the LRAM for decision making [12]

consumption as the cases in square 1 are below the time contracted. If there is no result in square 1, the algorithm searches for results in square 2 as the main objective is to save energy. If there any values in this square, the priority will be the cases with the least ART. The aim of this decision is to obtain a value of energy consumption lower than the average and meanwhile allow the contract to be broken in the least time possible. If none are found in square 2, the next square is 3. In this case, the least energy consumption is prioritized as its time is already lower than the deadline established in the contract. If none of the cases are found, the last option is square 4, where the algorithm calculates the average between the ART and the energy consumption and chooses what will result in the least value.

```

while There are values do
  Search for values;
  if There are any in square 1 then
    Use the one with the least consumption value;
  else if There are any in square 2 then
    Use the one with the least ART;
  else if There are any in square 3 then
    Use the one with the least consumption;
  else
    use the one with the least average between;
  end if
end while

```

Based on the rules previously specified, the LRAM will choose what the best case is in Table 3. The worst case is defined using the LRAM rules in an inverted way.

Table 3: Chosen scenarios [12]

Stages	Best		Worst	
	Light	Heavy	Light	Heavy
Choice of Data Center	BCR	BCR	BCR	RR
Creation of VM	SD4c	SD4c	SD4c	SD2c
Allocation of VM	Without migration	With migration	With migration	Without migration
Allocation of Services	Space-share	Time-share	Time-Share	Time-share

In the graphs on the left in Figure 6, it can be observed that when the best choice of policies is made in the case of a light workload, a time six times less can be obtained. Using a heavy workload, if the policy is not well chosen, the ART can be more than doubled.

It can be observed from the graphs on the right in Figure 6 that the wrong choice can lead to practically the double of energy consumption in the datacenter for services with a light workload. By analysing the composition of the policies for each scheduling stage with heavy workloads, the energy consumption in the datacenter can worsen by more than double in case it is not done in an optimised way.

In this section, the results obtained from the experiments carried out to evaluate GreenMAcc were presented. In the following section, the final conclusions of this work are drawn and suggestions for future work are made.

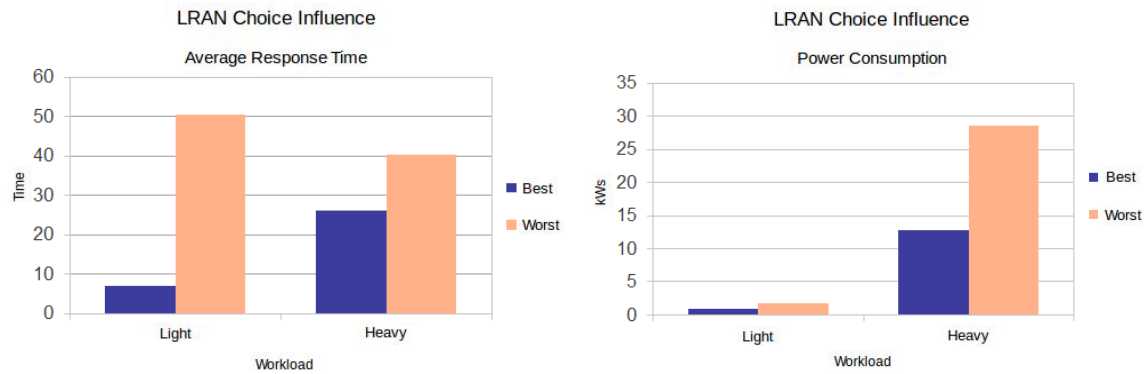


Figure 6: Comparison between the best and worst cases for Average Response Time and Energy Consumption [12]

5 Conclusion

The aim of this work was to present and evaluate a new architecture of green meta-scheduler providing QoS. In this section, the conclusions of the proposal and the results of this article are presented.

GreenMACC is an extension of the MACC architecture [2] for green computing in private clouds. In order for policies to be used which are concerned about green computing, the MACC architecture was remodeled, creating new modules and excluding or modifying others. In order to evaluate the performance of GreenMACC and the scheduling policies implemented for this work, the simulation technique was used. The tool used was CloudSim 3.0 [18] [19]. The proposed architecture proved to be consistent, carrying out the services required using various scheduling techniques in all its stages. Moreover, it showed the capacity of obtaining and controlling the energy consumption and the ART, an important metric to control the QoS. A scheduling policy, regardless of the stage it is in, can offer a service with a lower ART, however it could present a higher energy consumption, as in the cases of the SD4c, Round Robin and Space-share policies. There are cases where there is a lower ART and still lower energy consumption. This is the case of the VM allocation policies. The workload directly influences the ART and energy consumption, and it is also an important factor for choosing the scheduling policy in all the stages, especially in the VM allocation. In the case of having a light workload, the policy without migration proved to be more adequate, however when the workload increased considerably, the policy using the migration technique was more advantageous. The suitable choice of scheduling policies by analyzing the ART and energy consumption made in the LRAM module of GreenMACC showed a strong connection to obtain less response time, a crucial factor to respect the user's contract (SLA) and also to save energy in a private cloud. These facts help companies, who use the proposed architecture, to reduce costs in energy consumption, as well as maintaining the QoS provided to their users. The focus of future work will be to evaluate the flexibility of the LRAM GreenMACC module, to implement and assess other green policies in the GreenMACC and create new policies which can be optimized for the architecture proposed in this work.

References

- [1] T. V. T. Duy, Y. Sato, and Y. Inoguchi, "Performance evaluation of a green scheduling algorithm for energy savings in cloud computing," in *Proceedings of the 2010 IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum, IPDPSW 2010*, 2010. [Online]. Available: www.scopus.com
- [2] M. L. M. Peixoto, "Oferecimento de QoS para computação em nuvens por meio de metaescalonamento," Doutorado em Ciências Matemáticas e de Computação, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP), São Carlos - SP - Brazil, 2012.
- [3] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, and Y. Chen, "GreenCloud: a new architecture for green data center," in *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, ser. ICAC-INDST '09. New York, NY, USA: ACM, 2009, pp. 29–38. [Online]. Available: <http://doi.acm.org/10.1145/1555312.1555319>

- [4] D. Lago, E. Madeira, and L. Bittencourt, "Power-Aware Virtual Machine Scheduling on Clouds Using Active Cooling Control and DVFS," in *9th International Workshop on Middleware for Grids, Clouds and e-Science - MGC 2011*, dec. 2011.
- [5] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *Journal of Supercomputing*, pp. 1–13, 2010, article in Press. [Online]. Available: www.scopus.com
- [6] A. Beloglazov and R. Buyya, "Energy Efficient Allocation of Virtual Machines in Cloud Data Centers," in *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, may 2010, pp. 577–578.
- [7] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *CCGrid 2010 - 10th IEEE/ACM International Conference on Cluster, Cloud, and Grid Computing*, 2010, pp. 826–831.
- [8] A. Younge, G. von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for Cloud computing environments," in *Green Computing Conference, 2010 International*, aug. 2010, pp. 357–364.
- [9] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. In Press, Corrected Proof, pp. –, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167739X11000689>
- [10] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers," *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp. 732–749, 2010. [Online]. Available: www.scopus.com
- [11] S. K. Garg, C. S. Yeo, and R. Buyya, "Green cloud framework for improving carbon efficiency of clouds," in *Proceedings of the 17th international conference on Parallel processing - Volume Part I*, ser. Euro-Par'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 491–502. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2033345.2033396>
- [12] O. A. de Carvalho Junior, S. M. Bruschi, R. H. C. Santana, and M. J. Santana, "Greenmacc: An architecture to green metascheduling with quality of service in private clouds," in *XL Latin American Computing Conference (CLEI 2014)*, Sept 2014, pp. 1–9.
- [13] A. Beloglazov and R. Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers," *Concurr. Comput. : Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012. [Online]. Available: <http://dx.doi.org/10.1002/cpe.1867>
- [14] O. A. de Carvalho Junior, S. M. Bruschi, R. H. C. Santana, and M. J. Santana, "Green cloud meta-scheduling: A flexible and automatic approach," *Journal of Grid Computing*, apr 2015.
- [15] R. Jayarani, R. V. Ram, S. Sadhasivam, and N. Nagaveni, "Design and Implementation of an Efficient Two-level Scheduler for Cloud Computing Environment," *Advances in Recent Technologies in Communication and Computing, International Conference on*, vol. 0, pp. 884–886, 2009.
- [16] R. V. D. Bossche, K. Vanmechelen, and J. Broeckhove, "Cost-optimal scheduling in hybrid IaaS clouds for deadline constrained workloads," in *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010*, 2010, pp. 228–235. [Online]. Available: www.scopus.com
- [17] J. Zhang, C. Gu, X. Wang, and H. Huang, "A unified metascheduler architecture for telecom grade cloud computing," in *Information Science and Technology (ICIST), 2013 International Conference on*, March 2013, pp. 354–360.
- [18] R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya, "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services," *CoRR*, vol. abs/0903.2525, 2009.
- [19] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software - Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011. [Online]. Available: www.scopus.com