2015-04

Green Cloud Meta-Scheduling: a flexible and automatic approach

Journal of Grid Computing,The Netherlands : Springer,In press

J Grid Computing
DOI 10.1007/s10723-015-9333-z

Green Cloud Meta-Scheduling

A Flexible and Automatic Approach

Osvaldo Adilson de Carvalho Junior · Sarita Mazzini Bruschi · Regina Helena Carlucci Santana · Marcos José Santana

Received: 27 October 2014 / Accepted: 25 March 2015

Abstract This article aims to evaluate the flexibility of GreenMACC (Metascheduling Green Architecture to Provide Quality of Service in Cloud Computing). The GreenMACC has a module called LRAM (Local Resource Allocation Manager) to automate the execution of all scheduling policies implemented in the architecture. This module enables the Meta-scheduler automatically adjust for each type of service requested by the user of a private cloud. Due to this function, can be ensure the most appropriate behavior to the principles of GreenIT while worrying about the quality of service. In this paper is shown the importance of the LRAM on GreenMACC. This article is also shown how to include a new policy in GreenMACC in a way that identifies the LRAM and automatically use. Through the performance evaluation of the new policy included it could be concluded that the GreenMACC is a flexible, reliable architecture and the LRAM module enables the automation of choosing the best scheduling mechanism in a private cloud.

Keywords Green cloud computing · Scheduling · Simulation · Quality of service · Energy

O. A. de Carvalho Junior (✉)
Federal University of Alfenas, Poços de Caldas, MG, Brazil
e-mail: osvaldo.carvalho@unifal-mg.edu.br

O. A. de Carvalho Junior · S. M. Bruschi ·
R. H. Carlucci Santana · M. J. Santana
University of São Paulo, São Carlos, SP, Brazil

S. M. Bruschi
e-mail: sarita@icmc.usp.br

R. H. Carlucci Santana
e-mail: rcs@icmc.usp.br

M. J. Santana
e-mail: mjs@icmc.usp.br

1 Introduction

The Green Computing is a vision addressed in recent years aimed at encouraging the use of Information Technology (IT) with a fair and legitimate concern for the environment. The main issue is the economy of resources which, in this area, we have the power as the most important factor. Besides the concern about energy consumption, emissions of CO_2 (Carbon dioxide) in the atmosphere is an important area for GreenIT factor.

Currently, concerns about climate change lead scientists to think increasingly in ways, shapes or alternative pathways that seek to facilitate sustainable consumption of natural resources. It is known that a portion of the energy consumption of data centers can be reduced, which can be achieved with proper management servers that are not in use [1]. Green meta-scheduling is inserted in this context, assisting in the choice of datacenters and adequate allocation of Virtual Machines (VMs) on the hosts.

This work is not intended to present a new architecture for cloud computing as those given in papers like Rimal et al. [2]. In fact, the aim of this paper is to present an architecture of a metascheduler to cloud computing.

The GreenMACC [3] is an extension of MACC (Metascheduler Architecture to provide QoS in Cloud Computing) [4], whose main aim is to schedule virtual machines to hosts in order to achieve the SLA (Service Level Agreement) signed with the client. On the other hand, the GreenMACC, enables not only the use of scheduling policies with QoS in mind, but also the use of policies whose main objective is green computing in a private cloud. As the MACC architecture already offers adequate QoS for cloud, it is possible to use and extend it for a GreenIT panorama resulting in the GreenMACC being proposed. The MACC uses economy models for negotiation with the user. As the MACC is designed to Intercloud, the negotiation of service value with the client becomes necessary. There are very well defined forms of negotiation specified in the literature [5]. However, for this article such trading will not be thorough because of GreenMACC architecture be designed for a private cloud.

In studies that can be currently found in the literature, there is the use of specific points for performing efficient green staggering. These points are called **(PADEVE)**, **P**ontos de **A**nálise para **D**ecisão no **E**scalonamento **VE**rde in portuguese. these points are: processor, Network, Refrigeration and issuance of CO_2 [3].

The point that has been used in the literature for more green is scheduling the processor. Various techniques are used emphasizing the **DVFS** (Dynamic Voltage and Frequency Scaling) [6, 7].

Other works in the literature concerned with the distribution of VMs in order to overload the processing of some *hosts* to turn off those who are idle. The calculation of the average processing time required of the task to be performed on a textit host can be used as a basis for scheduling and so off the *hosts* without using [8].

The reduction in the number of processors in use in a datacenter seeking energy savings is more a form of scheduling used in the literature. For this there are several techniques, such as using neural networks [1], or use live migration of virtual machines [9–12].

The literature paper [13] that proves that transport and switching in a computer network can have a significant percentage of energy consumption in the cloud. As the use of the Next Generation Network, being a flexible network infrastructure [14], it can become an ally for making decisions on scheduling. One way to analyze network usage is observing the amount of migration of virtual machines. There are some works that propose the use of migration for reducing energy consumption [7, 11]. Other articles focus on reducing the amount of migration, which generates greater savings than using the migration without any control [9, 10, 12].

Jobs that have cooling as PADEVE have several concerns such as consumption of air conditioning Datacenters [15], spin down the processor cooler [7] and temperature reduction generated by the processor [9]. In all evaluated areas has been a reduction in energy consumption. The emission of carbon dioxide (CO_2) is another point to be used in scheduling green policies. There are works that analyze proposed possuim as many points as processor consumption, consumption of air conditioning, the cost of energy and the emission of CO_2. The main idea of this analysis PADEVE is send to meta-scheduler a coefficient of carbon emissions of data centers (provided by American Environmental Agency) and through these data make the decision of scheduling [15, 16].

The emission of carbon dioxide (CO_2) is another point to be used in scheduling green policies. There are works that analyze proposed possuim as many points as processor consumption, consumption of air conditioning, the cost of energy and the emission of CO_2. The main idea of this analysis PADEVE is send to meta-scheduler a coefficient of carbon emissions of data centers (provided by American environmental agency) and through these data make the decision of scheduling [15, 16].

The GreenMACC already have on your Scheduler module focusing on various QoS policies inherited from the MACC, or energy saving but does not have a policy which is PADEVE the issuance of CO_2. A major goal of this work is to demonstrate the flexibility of GreenMACC in using various scheduling policies with different visions and goals, and automatically choose the most appropriate to the agreement made with the user. For a better understanding of the mechanism GreenMACC, this work demonstrates the insertion of a new policy variables, functioning, goals and points of analysis other than those already implemented.

This article is organized into six sections. The Section 2 defines the structure of GreenMACC architecture where the LRAM is inserted. The third section presents the methodology for the evaluation of LRAM and GreenMACC. The fourth describes the LRAM in more detail, explaining about its operation. The fifth section demonstrates how it is possible to insert a new policy in GreenMACC and integrate it with the whole architecture enabling use by the LRAM and a performance evaluation of the new policy implemented in the architecture is presented. And finally, ends with the completion of this work.

2 GreenMACC - Green Metascheduler Architecture to provide QoS in the Cloud Computing

The GreenMACC [3] is an extension of the MACC that enables the offering of services provided in the cloud using green scheduling policies in set of policies that can provide QoS transparent to the user. Its data stream is different because it was designed with a private cloud. The following three subsections present in more detail the data flow of the proposed architecture, explains the operation of the four stages of scheduling GreenMACC with policies that will be evaluated and finally make a qualitative comparison with other architectures proposed in the literature.

2.1 Architecture and Data Flow

The architectural overview of GreenMACC can be observed in Fig. 1.

In the first layer are the authentication control and the Trader. The first is responsible for authenticating the user login can be made, or by other known methods of automatic authentication. The second is responsible for negotiating with the user which needs to response time, energy consumption, and other points used by the scheduling policies. This information will be used by LRAM in deciding which policy to use. This negotiation with the user can be automatically, according to user class (president, manager, analyst, etc.), or allowing the user to choose possibilities presented where there is an interaction between QoS provisioning and GreenIT.

In the core layer meta-scheduler there are several modules, each with its assignment to the service to be delivered to the user in accordance with negotiated. One of the modules is responsible for updating and contain all information necessary to the use of green metascheduling (Green Info) techniques, such as specific energy consumption of each processor in the datacenter, using DVFS, the state of all hosts (on, off, standby), temperature of a processor, emission coefficient CO_2 a Datacenter, etc; another module, Workload Prediction is responsible for the data required for load forecasting each host or processor core, necessary for decision making of some policies migrating virtual machines and hosts the shutdown. The remaining modules were already in MACC and have not been modified, with the exception of Scheduler and LRAM. The first is adapted to allow the implementation of green policies. However the most significant change is presented in LRAM. In MACC, this module was only an intermediary between the core and the GreenMACC datacenter, ie, transmitting the required information to the local scheduler datacenter create VMs and allocating them to hosts. In GreenMACC, it helps in decision making, choosing which scheduling policy is used in each of the four stages: first the choice of Datacenter, after the creation of the virtual machines and how these will be allocated on the hosts, and finally in the allocation of tasks of the services required on virtual machines. All stages cited will be detailed in the following subsections. This decision making can be made by observing several factors such as the negotiation made with the user and also analyzing the current burden to the cloud.

For better understanding of how the service request is processed and answered, in Fig. 2 one can observe a flow diagram of the whole architecture.

The process starts with the user authentication through the Authentication Control. Then, through the user interface, is created the service request by the user. The request is sent to the Admission Control which queries the MDSM (Monitoring and Discovery System Manager) about the availability of resources and services. The MDSM query information in the Index Service resources on the services available in the Service Info and also on the time required to perform the requested service the Workload Engine. At that moment the MDSM returns the information to the Admission Control responsible for the decision on admission or discard the request. If there sufficient resources and the requested service is available, the request is passed to the Trader. The Trader aims to try to negotiate with the user so that the service will be attended by analyzing what is most important to

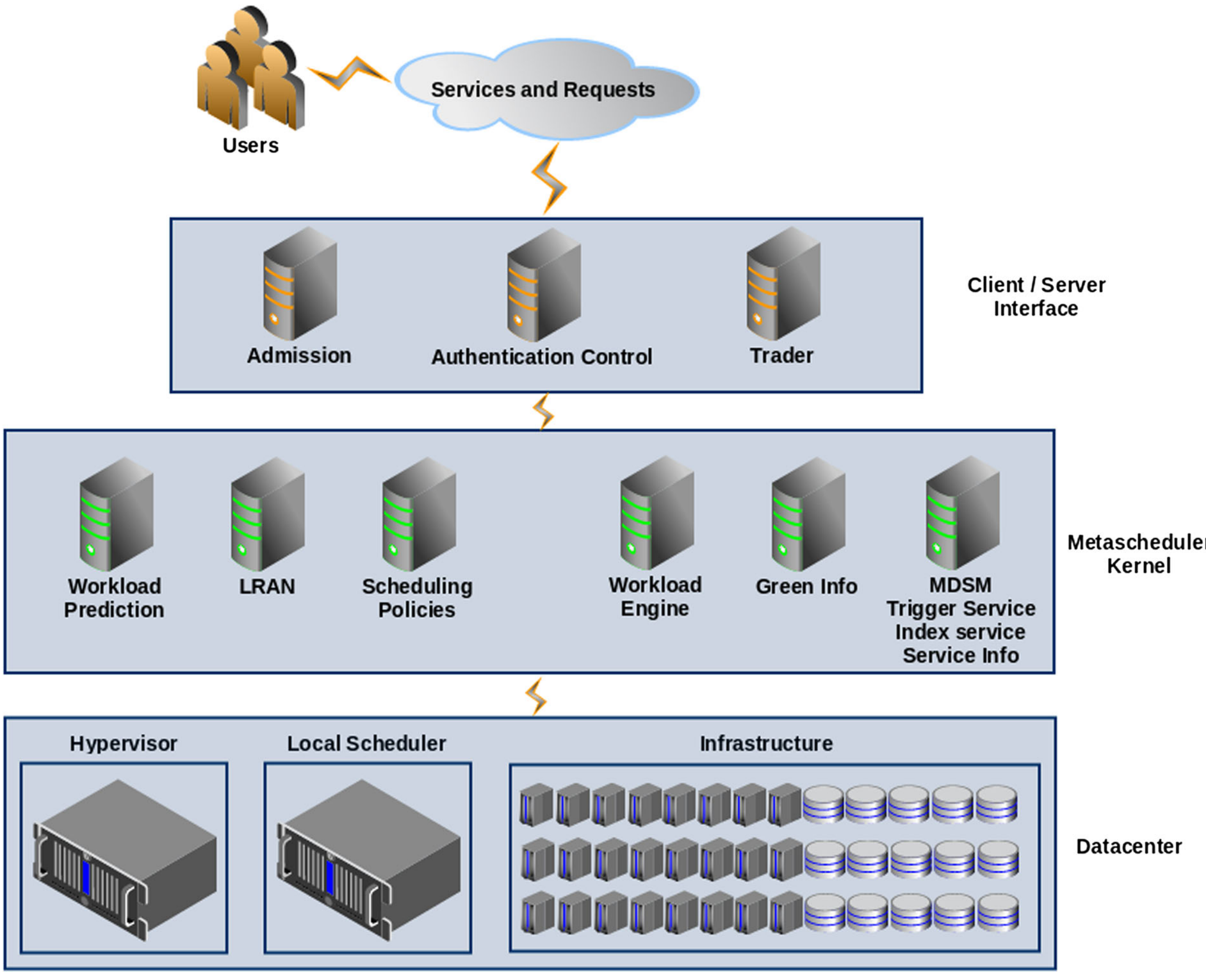


Fig. 1 Architectural Overview of GreenMACC [3]

the user (response time, energy consumption, etc.). This process is called Green Trading. To this query the Trader MDSM which then queries the Green Info about the points of green analysis (PADEVE). The latter returns the data to the MDSM which are then sent to the Trader. Made negotiation with the user, which can be done automatically or not, the contract information (SLA) and QoS are saved in a specific location for use by the scheduling policies. At this point the request is passed to the front-end core meta-scheduler that manages all information contained on its components. The front-end sends all information of negotiation and workload to LRAM, where this data is analyzed and made decision making of policies implemented in meta-scheduler which will be used in each of the four stages of scheduling. After choosing the datacenter for the first stage of scheduling policy information from the other stages are passed to the local scheduler chosen datacenter. Finishing up the processing of the requested service response is sent to the user.

The progress of the whole process from the user requests the service to answer this service is transparent to the user. After authentication the user sends the request of the service and at that time all the necessary checks are made without the user becomes aware of the process. Then the proposal of the possibilities of the service offer is presented to the user appears. After choosing the proposal that best meets your needs, just wait for the response from the service. This process can be even more transparent if the implementation of automated trading is in accordance with the characteristics or history of the user. Thus, after authentication, the user sends the request

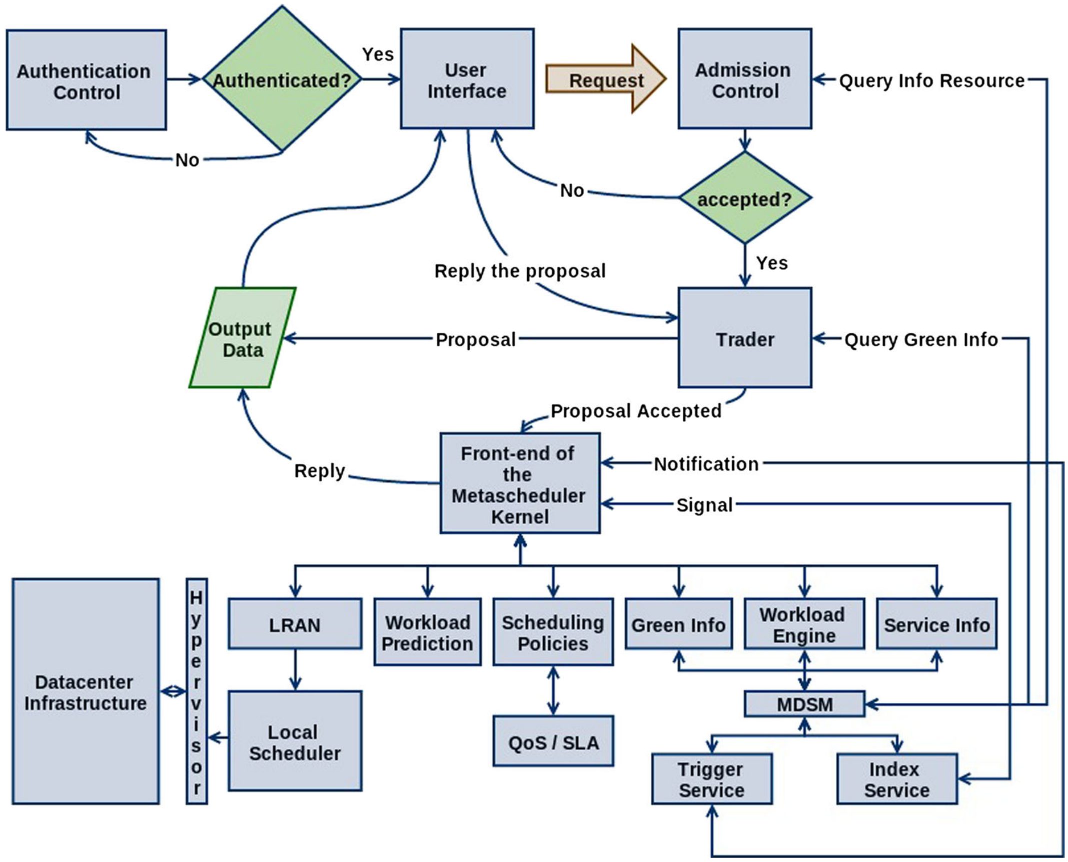


Fig. 2 Dataflow Diagram of GreenMACC [3]

and does not care about the negotiation that involves energy savings, response time, etc. Actions that occur in the core of meta-scheduler are also transparent to the user. When the request reaches the front end of the core meta-scheduler the first step is to send the characteristics of the load and, if applicable, the user also to LRAM. In LRAM, choosing the best policy for each stage of scheduling for the particular case occurs. After the choice of policies, LRAM sends the name of the chosen policies and then these are triggered in Scheduler module. The chosen policies seek all the information necessary for their implementation in QoS / SLA, GreenInfo and Workload Prediction modules. The first policy to be implemented is the choice of the Datacenter. Then the information is sent and the request for services to the local scheduler that creates Virtual Machines. Soon after created the Virtual Machines are allocated on hosts Datacenter chosen and finally the requisite tasks are distributed among VMs to run. In the next subsection are presented each of these stages and the policies implemented in each.

2.2 Scheduling Stages

As already mentioned, the GreenMACC has 4 stages of scheduling: Choice of Datacenter, Virtual Machine Creation, Virtual Machine Allocation and finally Task Allocation in Virtual Machines.

2.2.1 Choice of Datacenter Policies

The policies of this stage are responsible for making decisions in the choice of the Datacenter that will receive the users request and execute the service. In this work, three policies were used for the choice of the Datacenter: Round Robin (RR), Network Capacity Based (NCB) and CEB - CO_2 Emission Based; where the latter is based on the work by Garg et al. [15, 16]. The first two policies already exist in the GreenMACC and were used in this work for demonstrating a case-study whereby a new policy is inserted into the GreenMACC and also for evaluating how the LRAM module adapts itself to these changes. The other reason for choosing these two policies is to demonstrate the ability of the GreenMACC to work with policies not initially designed for GreenIT. The latter policy was implemented exclusively for this work aiming to demonstrate how to include a new policy into the GreenMACC. These presented policies already existed in the other scheduling stages, thus, this is the only stage receiving a new policy.

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 Datacenters one by one, following a Singly Circular Linked List. The main advantage offered by this policy is that it avoids overloading the meta-scheduler, as there is no decision making that requires intensive processing. However, the Datacenter chosen may not be the ideal one, and a situation may be possible where either the service is forwarded to a Datacenter that is already overloaded or the hosts are being under-used, resulting in 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 networks information, in this case the latency, as the fundamental criteria for taking the decision as to which Datacenter the users 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 in relation to the latency, however the meta-scheduler will have a high overload since all Datacenters will be consulted before taking the decision. As for the CEB policy, there is no overload involved as the information concerning CO_2 emission is not very dynamic when compared to the networks latency and remains fixed for a long period in the GreenInfo module. After choosing the Datacenter, the next step is concerned with choosing how the virtual machines will be created in the chosen Datacenter. For this, policies are available for the creation of VMs.

The third policy used during this stage, the CO_2 Emission Based (CEB), is better explained in Section 4, where the automatization of the GreenMACC is discussed.

2.2.2 Virtual Machine Creation Policies

After choosing the Datacenter, 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 of VMs, 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, as it is at this phase that the number of vCPUs for each VM created is defined. For policy SD2c, 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 Datacenter'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 have the same purpose, i.e. to show how the GreenMACC can work with different kinds of policies of widely varying purposes. Moreover, as the meta-scheduler also needs to meet the agreement made with the user, it also offers to the LRAM policies concerned with the quality of service. After creating the VMs, the next step is to allocate them to the hosts at the chosen Datacenter, thus moving on to the next stage.

2.2.3 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 Datacenter. Another task of this phase is to manage the migrations of these MVs when the scheduling policy implemented at this stage has this feature. In this article, two policies were used. One having static allocation features, i.e. without migrating (whithoutM) to virtual machines and the other policy having dynamic features (withM). The latter uses the migration technique. Both policies were presented in studies that aim to evaluate policies concerned with reducing energy consumption

[10, 17]. This choice was made to allow the LRAM to choose between policies with and without migration and also to better understand the behavior of GreenMACC when policies are used that employ virtual machine migration techniques, as well as 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 without losing the focus, either in green computing or in the quality of service.

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

2.2.4 Task Allocation Policies

The objective of the last stage of scheduling is to choose the way to allocate tasks to the virtual machines which have already been allocated in the previous stage. For this, two already existing scheduling techniques were used: Time-share and Space-share. The two policies were already used in work which was concerned with saving energy [9, 10, 17], 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.

2.3 Comparison Between GreenMACC with Other Architectures

There are several works in literature related to the topic of this article. The result of this subsection is to make a qualitative comparison using a table where you can make a distio between GreenMACC with other related work. Table 1 can observe seven columns. The first specifies the architectures evaluated, the remaining columns indicate with an ”X” which architectures possuim control mechanisms multiple datacenters, users, power consumption (kWs), emission of Carbon Dioxide (CO_2), Quality of Service (QoS) and finally automated choice of policies in the four stages of scheduling (Auto) respectively.

Table 1 allows to observe the differences between the proposed architectures for scheduling cloud. The MACC [4] architecture that formed the basis for the development of GreenMACC, can work with multiple datacenters seeking the quality of services. However, no specific modules that allow the implementation of green policies. The MACC also has no authentication and user control, on the other hand has GreenMACC these mechanisms, which allows you to automatically make the choices of what policies can be implemented as user profile. This automated selection may also be made according to the load imposed on the services GreenMACC. Besides the MACC and the GreenMACC the CAGCA (Carbon Aware Green Cloud Architecture) [16] also has the ability to manage multiple datacenters. The CAGCA presents an architecture that allows green green staggering targeting both energy consumption and the emission of carbon dioxide in addition to worrying about QoS. However lacks authentication and user control nor the ability to choose multiple policies in four-stage scheduling of the cloud. The GreenCloud [6] was not designed for multiple datacenters and also has no control of users. However its architecture prioritizes the scheduling aimed at saving energy and QoS without worrying about the issuance of CO_2. The GCA (Green Cloud Architecture) [12] offers and user control techniques for energy saving and respect for QoS, however does not manage multiple datacenters and therefore does not offer the choice of automation policies in the four stages of scheduling. Besides not having in their political concerns with carbon dioxide emissions.

In 2009, we proposed a meta-scheduler that offers two levels of scheduling [18], like the MACC, the first choice in the Datacenter and the second resource allocation Datacenter chosen. The meta-scheduler on two levels proposed in the cited work can work with multiple datacenters and allows policies to implement this level of scheduling. And as the MACC, also has concerns with the quality of service offered to the User, however the work presented is not cited any control module or allow users to have control over

Table 1 Comparison of Related Work

Architecture	Multiple DC	Users	Kws	CO2	QoS	Auto
GreenMACC	X	X	X	X	X	X
MACC[4]	X				X	
CAGCA[16]	X		X	X	X	
GreenCloud[6]			X		X	
GCA[12]		X	X		X	
2Levels[18]	X				X	
HICCAM[19]	X	X			X	
UMATGC2[20]	X	X			X	

data consumption and carbon dioxide emissions. The meta-scheduler 2 levels also does not offer the possibility of choosing automaticar various scheduling policies for both levels.

The HICCAM (Hybrid Cloud Construction and Management) [19] has a responsible for managing the users module that enables their authentication and provides a strict control of the use of services offered in the cloud. His priorities are always their QoS and scheduling techniques allow you to manage the use of multiple data centers. However, there is no concern about green computing or an automated system that allows choose from a menu of techniques for scheduling options.

The most recent work in this present compared with GreenMACC is the architecture of meta-scheduling UMATGC2 [20]. This meta-scheduler has characteristics closely resembling the HICCAM because it offers control of cloud users, enables scheduling up to a level of Datacenters and prioritize policies that are exclusively for the quality of service offered to its user. The management of various scheduling techniques does not exist and there is no structure in the architecture that offers the possibility of managing the energy consumption or the emission of carbon dioxide.

Given the above it can be concluded that the GreenMACC is the most complete among the evaluated architecture because its configuration enables the deployment and management of automated scheduling policies in all four stages without losing a concern with the quality of services offered to users. In GreenMACC is also possible to have full control over the cloud allowing users to implement a customization in choosing the policies to be implemented.

In this section the architecture of the proposed meta-scheduler was presented in addition to the policies implemented at each level of scheduling and ended with a comparison of related work. In the following section we present the LRAM, responsible for the choice of scheduling policies in the four stages of scheduling module.

3 LRAM (Local Resource Allocation Manager)

The LRAM module is able to analyze the information concerning the kind of workload imposed by the user and, after the automatic analysis, choose among the policies implemented in the Scheduling Policies Module, i.e. the ones that offer a better result according to the negotiation. This important role of the LRAM is called automatization of the GreenMACC, as it allows for decisions regarding the four scheduling stages to be taken in an automatic way. In this section, the stage of scheduling that is made choice of datacenter the existing policies in the GreenMACC, BNC and Round Robin, will be used. The new policy will be used in the next section.

Section 4 shows the process followed by the LRAM to decide automatically among the implemented policies at each scheduling stage.

In order to understand better how the LRAM takes decisions when choosing policies, there are two graphs in Figs. 3 and 4. The first refers to the results obtained with a light workload and the second presents values obtained using a heavy workload. In both graphs, the x axis represents the values concerning the energy consumption in KWs, and the y axis represents values referring to the average time response in time units. The graphs are divided into 4 squares. The lines

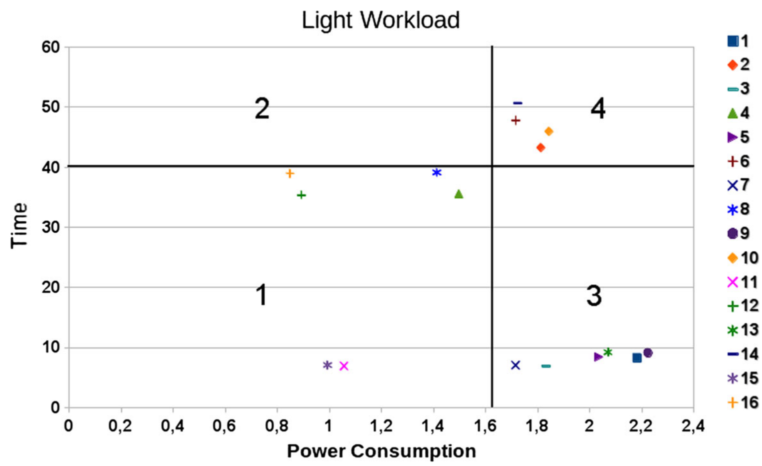


Fig. 3 Values Analyzed by the LRAM for Decision Making [3]

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 users contract. The parallel line on the y axis is defined by the value of the average of energy consumption values obtained with the specific load being used.

With the quadrants defined, the the LRAM can make the decision according to the following rules in order of priority:

Rules

1. Choose the case with the lowest power consumption in quadrant 1
2. Choose the case with the lowest average response time (ART) in quadrant 2
3. Choose the case with the lowest power consumption in quadrant 3
4. Choose the case with smaller value from the average between the ART and energy consumption in quadrant 4

In the following algorithm can be seen as the LRAM is implemented to always get the best possible case. In this example, the user has prioritized energy savings. First there is the existence of a configuration of algorithms in four stages that fits into Quadrant 1 If there is, will be selected configuration with the lowest energy consumption since the cases in quadrant 1 are below the contracted time. If there is no configuration algorithms in quadrant 1, the algorithm searches results in quadrant 2 If values exist in this quadrant priority will be cases of low TMR. This decision aims to obtain a value of energy consumption below average while allowing the contract is broken the shortest possible time. If not also found in the next quadrant

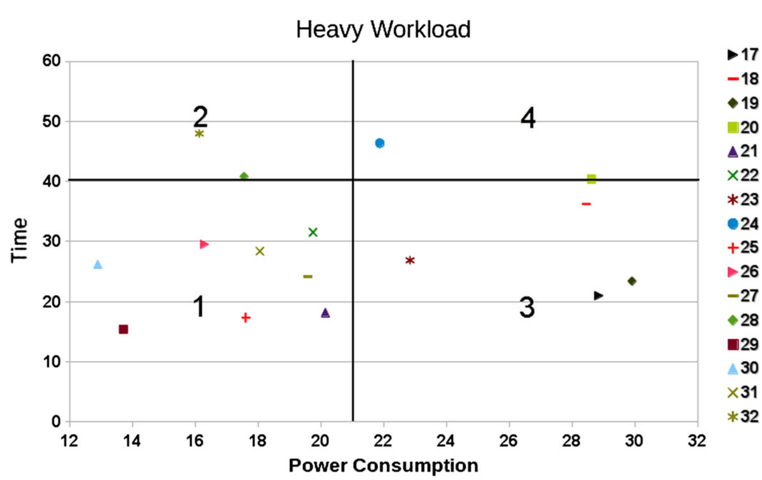


Fig. 4 Values Analyzed by the LRAM for Decision Making [3]

2 quadrant 3 is that if the priority is lower power consumption since your time is below the deadline set in the contract. And if none of this is found the last option is the four quadrant where the algorithm calculates the average between TMR and the consumption of energy and chooses the result that a lower value.

LRAM Algorithmic (Energy Consumption Priority)

While Exists Values
 Search Values;
 if Exists value in quadrant 1
  Use the lowest value of consumption;
 else if Exists value in quadrant 2
  Use the lowest average response time (ART);
 else if Exists value in quadrant 3
  Use the lowest value of consumption;
 else
  Use the lowest average between ART and consumption;
 end if
end while

The LRAM, based on the rules specified above, choose what the Table 2 is called a best case. The worst case is defined using the rules of the LRAM in reverse.

In Fig. 5 can be seen that when making the best choice of policy in the case of a light workload, can obtain six times shorter. With a heavy workload, if the choice of policies is not well made, can the average response time worsens in approximately 60 %.

Observing the graphs of Fig. 6 it turns out that with a wrong choice can have power consumption of datacenter services for nearly folded with light workload. Analyzing the composition of policies for each stage of scheduling with heavy lift services, can the power consumption in the datacenter worsen more than double if it is done so not optimized.

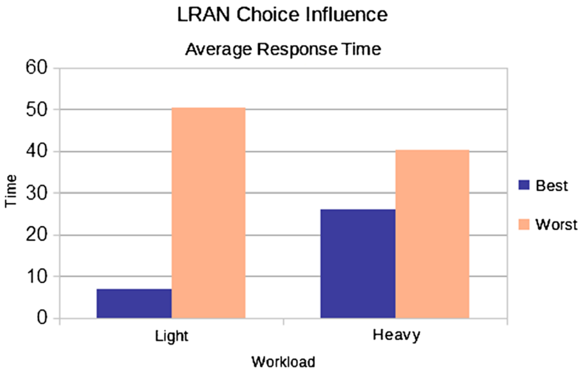


Fig. 5 Comparison between the Best and the Worst Case to Average Response Time [3]

Discussed in the next section is one of the main advantages of GreenMACC, the automation, ie, how GreenMACC allow the inclusion of new policies and new variables and automatically adjusts to them.

4 GreenMACC Automatization

As shown in Fig. 1, the GreenMACC offers an interface with the user. This interface enables interaction with two kinds of users: the regular user, i.e. who uses the system only to request a service, and the advanced user, i.e. the system administrator. The interface for the system administrator enables him/her to insert scheduling policies in any of the four scheduling stages of the GreenMACC. In addition to being able to insert new policies, the administrator can also insert new variables in order to obtain more system information or to use them as input variables for the new implemented policies.

In Fig. 7 can view a GreenMACC interface with your system administrator. This interface allows the administrator to enter new policies GreenMACC addition of new variables. In the combo box to the left,

Table 2 Selected Scenarios

Stages	Best		Worst	
	Light	Heavy	Light	Heavy
DC Choice	BNC	BNC	BNC	RR
MV Creation	SD4c	SD4c	SD4c	SD2c
MV Allocation	whithoutM	withM	withM	whithoutM
Task Allocation	Space-share	Time-share	Time-Share	Time-share

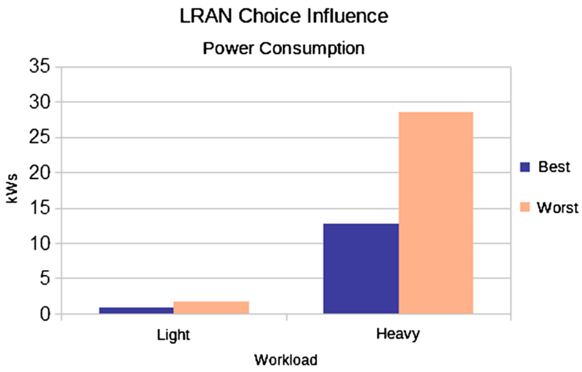


Fig. 6 Comparison between the Best and the Worst Case to Energy Consumption [3]

the administrator chooses which of the four stages of scheduling that the new policy will be inserted. Just below the combo box is the field for the date that the proposed changes take effect, ie, the date that the GreenMACC must be updated. In the center of the browser is a text box where the administrator must implement the new policy. In the lower right text box you must enter the new variables. Below the right lower corner of the text box is the field where you can select the shape that the LRAM must analyze the new output variables. The purpose of this information is to make clear to the LRAM the smallest values of the new variable are better results than those with higher values or vice versa. In case the administrator wants to enter only a new variable without a new policy or just a new policy without a new variable, simply leave the text box related to the non inserted data blank.

To better demonstrate the automation of GreenMACC will be used as example the inclusion of a new policy that also needs a new variable. A policy based on the issuance of CO_2 (CEB) fits this case to be a new policy that uses a new variable (CO_2). The CEB political scales so that the requests are distributed to the datacenters that possess a lower emission of CO_2 per Killowatt / second consumed. Its advantage is that there will always be priority when choosing datacenters that pollute less. However, the choice of datacenter may not be optimal for reducing energy consumption or quality of service.

To insert this new political system administrator must initially indicate the combo box to the left that the new policy should be inserted at the stage of choosing the Datacenter. The system puts in the date field to the current date automatically, ie, it should be changed only if the administrator decides for any technical issue that the date should be different from the date of the day that the new policy is being inserted. In the central text box administrator must implement the code of the new policy CEB. In the lower right text box you must enter the variable responsible for storing the carbon dioxide emissions per Kws of each Datacenter. In the field below the ”smaller” option should be selected, because then the lowest values of carbon dioxide emissions are better than the more than CO_2 emissions in the atmosphere.

When the administrator clicks the OK button a number of actions are performed automatically in the system. The first action will be to insert the new policy in the Scheduler module GreenMACC the first stage of scheduling, ie, the choice of the Datacenter. Then responsible for storing the new emissions CO_2 datacenter each variable will be inserted into an existing table in GreenInfo module. With the inclusion of the variable in GreenInfo, your content will be considered by the module Trader in negotiations with the user.

Fig. 7 GreenMACC System Administrator Interface

The last action is to pass the module LRAM information that the lowest values of carbon dioxide emissions are the best results. These data are important when running the benchmark for updating the LRAM. This update enables the LRAM module insert in your range of options implemented the new policy.

It is important to highlight that the flexibility of the GreenMacc is limited to its architecture. At every new variable inserted, the infrastructure needs to be updated. That is, in the case of the variable of CO_2 emission, the Datacenters administrator of the cloud has to configure the system so that the Datacenters inform the GreenMACC of the necessary data in order to enable the new policy to be used. The information sent by each Datacenter will be inserted automatically by the system into the table of the GreenInfo module.

Another important feature to be highlighted is how the LRAM configures itself. The LRAM run a benchmark at the time a new policy or a new variable is entered. This benchmark can be better visualized in Fig. 8.

The benchmark is in fact a simulation of the GreenMACC using the CloudSim [21, 22] in version 3.0, which already has been used in several other papers in the area of cloud computing [1, 10, 18].

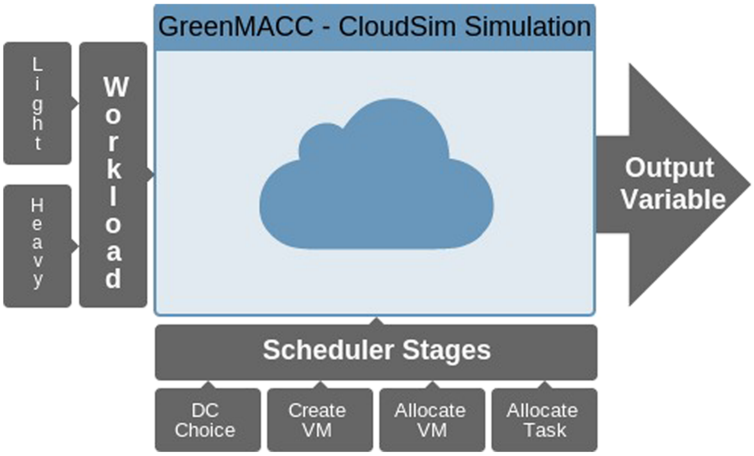
The model used for the simulations has the following characteristics: 15 Datacenter with 1000 hosts each. Hosts can have two, four or six cores divided equally in each datacenter. The hosts are fixed characteristics with 16GB of RAM and 1 Gbit/s of Bandwich. The simulations are made considering implementing a private cloud for 24 hours, with data centers around the world on five continents. All Datacenter offer the same service. Stored data and the applications and services are not geographically distant, ie, the data are replicated in all datacenters. All services offered are applications such CPU-Bound. This model of data replication in all datacenters can be used by companies offering the same services in all its datacenters. One can cite the example of a bank where the company offered services in the cloud are the same regardless of their geographical location.

The implementation of the GreenMACC CloudSim was done by creating classes, where each class has all the attributes and methods needed to function exactly like a specific module of the architecture presented in this work. Figure 9 presents the class diagram that allows better visualization, and consequently, a better understanding of how GreenMACC was implemented in CloudSim.

With GreenMACC implemented in CloudSim, begins the implementation of the policies of the four stages of scheduling meta-scheduler. In total there are 8 policies, two for each of these stages. The benchmark uses five factors (4 stages scheduling and workload). For each factor, we have two levels: the stages of scheduling are the very policies and the workload light and heavy levels are used. The light load was modeled by 30 users, with 500 requests per user tasks with little processing. The heavy load is 60 users, with 1,000 requests for user tasks with high processing demand. The factors and levels are shown in Table 3. The total scenarios considering the variations in the levels when using the full factorial design is 32 (2^5).

The output variables were initially considered by the benchmark **A**verage **R**esponse **T**ime **(ART)** which considers the network time, queuing time and service time, and energy consumption in **k**ilo **W**atts/**s**econds

Fig. 8 LRAM Benchmark



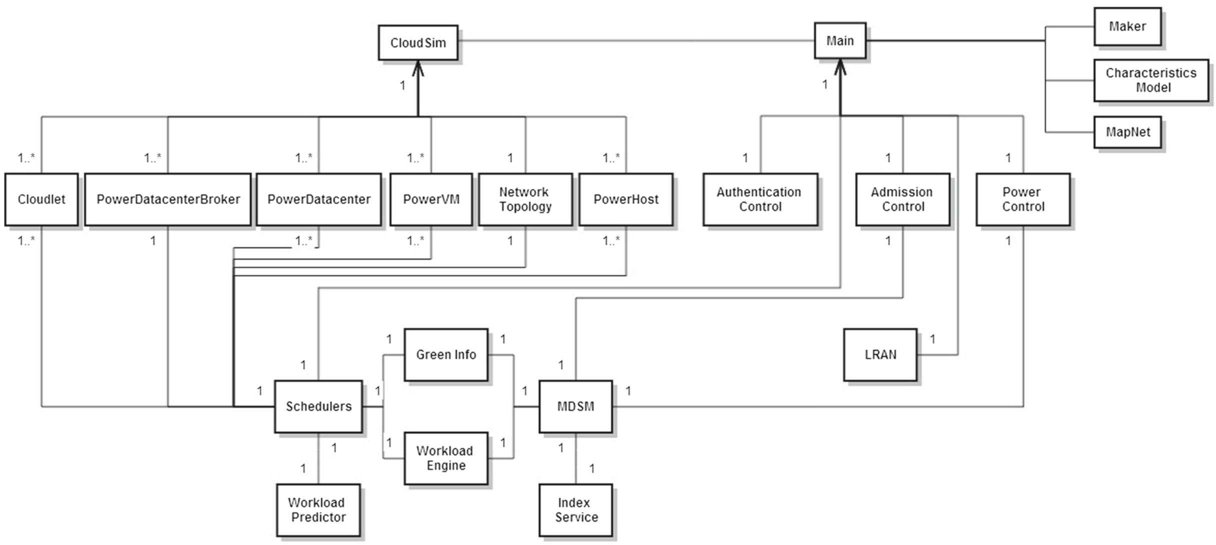


Fig. 9 GreenMACC Class Diagram [3]

(kWs). However, these variables can be changed as the user chooses to include a new variable analysis.

The shape of the benchmark run depends on the information updated by the administrator. Ie, only the administrator can enter a new policy, or only variables, or inserting a new policy set of new variables. In all three cases the benchmark runs 10 times each scenario, each with a different seed for generation can be subsequently calculated and the average confidence interval random numbers. This is necessary because there is a randomness in the arrival rate of requests. Each of the three possibilities is treated in a way:

- Insertion of a new policy: in this case, you increase the number of scenarios, because now one of the factors has a more level. So that the information is current LRAM, the benchmark should be run again, now considering this policy as a fixed factor of the stage where it is being inserted, and should vary all the other four factors (the other three stages of scheduling and the load).
- Inserting a new variable: in this case, you should perform the initial benchmark with 32 scenarios, only now having as response variable to be included variable.
- Insertion of a new policy and a new variable: in this case, the benchmark should be run for all scenarios, including the new political factor in what it refers to, then all scenarios must be run again for the results with the new variable can be obtained.

For this article we chose the third case to be what encompasses the first two cases. After inserting the new policy CEB Carbon Dioxide emission becomes a new variable. In the next section a case study considering the inclusion of a new policy with CEB responsible for the emission of carbon dioxide each

Table 3 Factors and Levels Used by the Benchmark

Factors	Levels
Datacenter Choice	RR e BNC
Virtual Machine Creation	SD2c e SD4c
Virtual Machine Allocation	Without Migration e With Migration
Task Allocation	Space-share e Time-share
Workload	Light and Heavy

Datacenter private cloud variable used in this work will be presented.

5 Case Study Inserting a New Politics and New Variable

The main purpose of this section is to demonstrate how GreenMACC was designed to be a flexible architecture. Initially one needs to define what policy will be inserted. In this work the policy based on the emission coefficient of CO_2 (CEB) was chosen. The CEB has as input the variable emission coefficient CO_2 per kW/h (CO_2/kWh) consumed in the Datacenter. This article is used to evaluate a private cloud with 15 datacenters around the world, more exactly in 7 countries. The criterion for the choice of countries was based on the economic importance of the country to the continent or region. The selected countries were: USA (North America), Brazil (Latin America), Germany (Europe), Australia (Oceania), South Africa (Africa), UAE (Middle East) and Japan (Asia). Based on Appendix F of the official document of the United States Department of Energy [23] was stipulated for each datacenter the amount of CO_2/kWh which is emitted into the atmosphere by each of the selected countries. This value refers to the geographical position of each datacenter, ie, each country has a value for the coefficient of carbon dioxide emissions and it is this value that should be attributed to the Datacenter. This information must be entered in GreenInfo module architecture. Once inserted into the green GreenInfo information to be used in scheduling, the new policy should be implemented in the Scheduler module. Another module that must be updated is the Trader. This module is responsible for negotiating with the user. Before inserting the new policy points were negotiated with the user response time and power consumption. After insertion of new information into GreenInfo, the Trader will also have the coefficient of emission of carbon dioxide (CO_2) as a point of negotiation with the user. This Trader update occur automatically when the user requests a new service and the Trader requesting to MDSM the green information to GreenInfo.

The next step is to run the benchmark to obtain the necessary means for the LRAM is updated results. From the moment that the LRAM is updated, the module can now make decisions, ie, can already choose between the policies already implemented and the new policy.

With the inclusion of the new policy, the Trader can negotiate with the user the emission of carbon dioxide plus Average Response Time (ART) and energy consumption. As an example for such work will be considered that the user negotiated as most important for the provision of service to carbon dioxide emissions and energy consumption factors. Ie the user wants to have the greenest possible service.

Made negotiation, LRAM will use the same algorithm as the previous choice of policies for the 4 levels of scaling. The difference is the factor used, ie, instead of using ART will Issuance of CO_2. This characteristic makes the GreenMACC a flexible architecture.

LRAM Algorithmic (Energy Consumption and CO_2 Priority)

while Exists Values
  Search Values;
   if Exists value in quadrant 1
   Use the lowest value of consumption;
  else if Exists value in quadrant 2

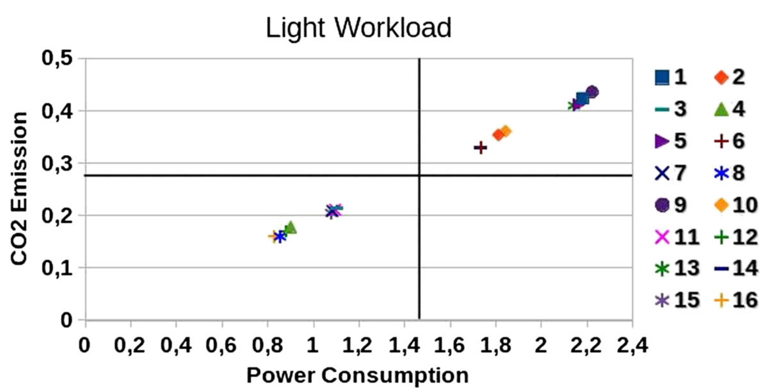
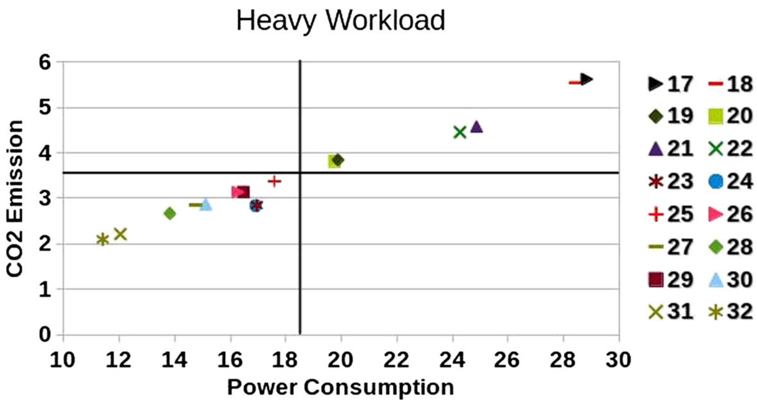


Fig. 10 Values Analyzed by LRAM for Decision Making

Fig. 11 Values Analyzed by LRAM for Decision Making



Use the lowest CO_2 emission;
else if Exists value in quadrant 3
Use the lowest value of consumption;
else
Use the lowest average between CO_2 emission and consumption;
end if
end
while

The LRAM will use graphics data represented in Figs. 10 and 11 for the decision in the choice of four policies used. The final decision will be taken as the LRAM the kind of burden to the system.

If the load is light, ie does not require high processing power, the LRAM use the data of graph of Fig. 10. On the other hand, if the load is heavy LRAM will use the data from the graph in Fig. 11.

In the next subsection it can be observed an evaluation of the results obtained with the new policy.

5.1 Evaluation of the results obtained with the New Policy Inserted

In this section a performance evaluation of the results obtained with the new policy inserted in GreenMACC is presented. With the results presented in this subsection becomes clearer understanding of flexibility and ability to make decisions and consequently the LRAM and the GreenMACC. For this evaluation used the same model used in the previous section results. The difference is the new policy implemented in GreenMACC whose PADEVE the issuance of CO_2.

In Fig. 12 can be observed average response time (ART) at best and in the worst case choice of scheduling policies in the four stages of GreenMACC. Considering these results, it can be seen that with a light load, the better the event delivers the service to the user within less than the worst case time. However, with services that require a higher computational power, the delivery time of the service of the worst case is less than in the best case. This is due to

Fig. 12 Comparison Between the Best and the Worst Case to Average Response Time

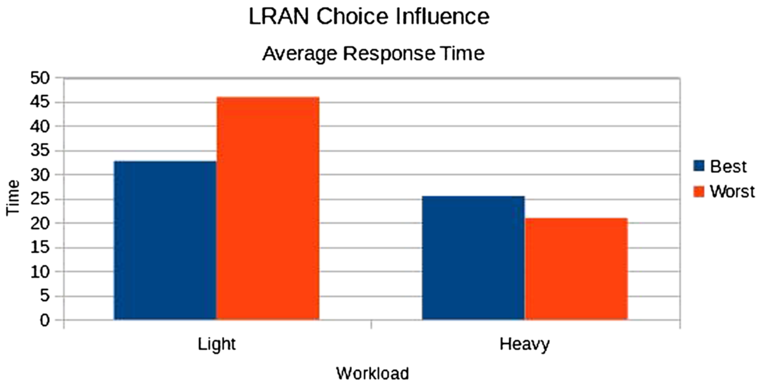
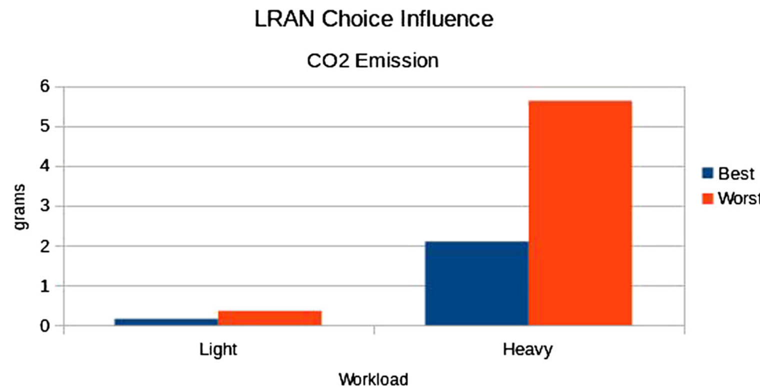


Fig. 13 Comparison Between the Best and the Worst Case for CO_2 Emission



the fact that at the time of negotiation with the user the option was to prioritize the reduction of energy consumption and reducing emissions of carbon dioxide. This option causes the LRAM choice policies for the four stages focusing on negotiating with the user. Ie, as the Average Response Time was not a factor considered for choosing the LRAM, cases where not get a great result for this factor may occur. In contrast, although not the optimal case, the times obtained in the best case did not exceed the stipulated deadline.

Figure 13 displays the graph with the results of CO_2 emission in the atmosphere considering the two types of tested load. When the user requires less computational work, ie, a light load demand services, the best case emits an amount of CO_2 slightly smaller than the worst case. However, when the workload increases, with the user requiring a higher computing power with a heavy load of services the gap widens considerably. Unlike the ART, the emission of carbon dioxide was a relevant choice for the LRAM factors. With the proper choice of scheduling policies taken by LRAM (best case) it can be observed a reduction of more than 50 % in carbon dioxide emissions in relation to a possible case where there is no LRAM (worst case).

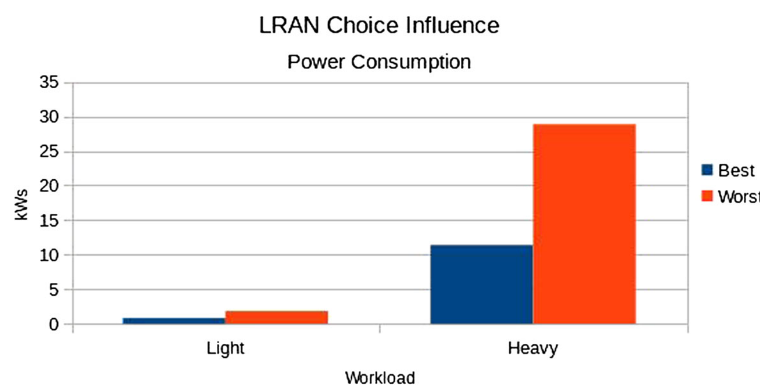
Another factor considered by LRAM the choice of scheduling policies due to negotiation with the user is power consumption. In Fig. 14 is possible to observe the figures for energy consumption in relation to the load. When the load is light, the best case presents more economical than the worst case. Requiring a heavy load of service due to a suitable choice of LRAM, one can reduce the power consumption of the data center to less than half. These results confirm the efficiency of LRAM regarding the negotiation made with the user.

As can be observed in this section, the LRAM is a relevant module in the GreenMACC architecture. The next section draws the final conclusions of this work.

6 Conclusion

The aim of this work was to present and evaluate the flexibility of a new architecture of green metascheduling providing QoS. In this section, the conclusions

Fig. 14 Comparison between the Best and the Worst Case for Energy Consumption



of the proposal and the results of this article are presented.

The LRAM is a module of the GreenMACC, an extension of the MACC architecture [4], for green computing in private clouds.

The proposed architecture proved to be consistent, carrying out the services required using various scheduling techniques in all its stages. Moreover, it showed flexibility allowing for the insertion of a new scheduling policy where the input and output variables are different from the policies previously implemented in the GreenMACC.

The initial negotiation with the user made by the Trader module is a relevant one. The results demonstrate that the factors negotiated with the user, and chosen as priorities, obtained positive results when the LRAM chose the policies. Even in the case when the factor was not considered as priority by the user (ART), the best case did not exceed the deadline.

The workload directly influences the results and it is also an important factor for choosing the scheduling policy in all the stages. For all kinds of workload tested, the results are positive regarding the negotiation with the user. That is, regardless of the workload, the factors chosen by the user as relevant for offering the requested service show values where the choice of policies made by the LRAM respects the contract made with the user.

The suitable choice of scheduling policies made by the LRAM module based on the negotiation with the user, showed a strong ally to obtain reduction in CO_2 emission in the atmosphere and also in energy consumption in a private cloud. These facts help companies, who use the proposed architecture to significantly reduce costs in energy consumption.

Acknowledgments We thank FAPESP, CNPq and CAPES for financial support.

References

1. Duy, T.V.T., Sato, Y., Inoguchi, Y.: 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)
2. Rimal, B.P., Jukan, A., Katsaros, D., Goeleven, Y.: Architectural requirements for cloud computing systems: an enterprise cloud approach. J. Grid Comput. **9**(3) (2011). doi:10.1007/s10723-010-9171-y
3. de Carvalho Junior, O.A., Bruschi, S.M., Santana, R.H.C., Santana, M.J.: GreenMACC: An architecture to green metascheduling with quality of service in private clouds. In: XL Latin American Computing Conference (CLEI), 2014, pp. 1–9 (2014). doi:10.1109/CLEI.2014.6965136
4. Peixoto, M.L.M.: Oferecimento de qos para computação em nuvens por meio de metaescalonamento. Phd thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP), São Carlos - SP - Brazil (2012)
5. Prodan, R., Wieczorek, M., Fard, H.M.: Double auction-based scheduling of scientific applications in distributed grid and cloud environments. J. Grid Comput. **9**(531) (2011). doi:10.1007/s10723-011-9196-x
6. Liu, L., Wang, H., Liu, X., X. Jin, He, W.B., Wang, Q.B., Chen, Y.: GreenCloud: a new architecture for green data center. In: Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session (ACM, New York, NY, USA), ICAC-INDST '09, pp. 29–38 (2009). doi:10.1145/1555312.1555319
7. Lago, D., Madeira, E., Bittencourt, L.: 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 (2011)
8. Lee, Y.C., Zomaya, A.Y.: Energy efficient utilization of resources in cloud computing systems, Journal of Supercomputing, pp. 1–13. Article in Press (2010)
9. Beloglazov, A., Buyya, R.: Energy efficient allocation of virtual machines in cloud data centers. In: Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on, pp. 577–578, (2010). doi:10.1109/CCGRID.2010.45
10. Beloglazov, A., Buyya, R.: Energy efficient resource management in virtualized cloud data centers. In: CCGrid 2010 - 10th IEEE/ACM International Conference on Cluster, Cloud, and Grid Computing, pp. 826–831 (2010)
11. Younge, A., von Laszewski, G., Wang, L., Lopez-Alarcon, S., Carithers, W.: Efficient resource management for cloud computing environments. In: Green Computing Conference, 2010 International, pp. 357–364 (2010). doi:10.1109/GREENCOMP.2010.5598294
12. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing, Future Generation Computer Systems In Press, Corrected Proof (2011). doi:10.1016/j.future.2011.04.017
13. Baliga, J., Ayre, R., Hinton, K., Tucker, R.: Green cloud computing: balancing energy in processing, storage, and transport. Proc. IEEE **99**(149) (2011). doi:10.1109/JPROC.2010.2060451
14. Rings, T., Caryer, G., Gallop, J., Grabowski, J., Kovacikova, T., Schulz, S., Stokes-Rees, I.: Grid and cloud computing: opportunities for integration with the next generation network. J. Grid Comput. **7**(375), 2009. doi:10.1007/s10723-009-9132-5
15. Garg, S.K., Yeo, C.S., Anandasivam, A., Buyya, R.: Environment-conscious scheduling of HPC applications on

distributed Cloud-oriented data centers. J. Parallel Distrib. Comput. **71**(6), 732 (2010)
16. Garg, S.K., Yeo, C.S., Buyya, R.: Green cloud framework for improving carbon efficiency of clouds. In: Proceedings of the 17th international conference on Parallel processing - Volume Part I (Springer-Verlag, Berlin, Heidelberg), Euro-Par'11, pp. 491–502 (2011)
17. Beloglazov, A., Buyya, R.: 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. **24**(13), 1397 (2012). doi:10.1002/cpe.1867
18. Jayarani, R., Ram, R.V., Sadhasivam, S., Nagaveni, N.: Design and implementation of an efficient two-level scheduler for cloud computing environment, advances in recent technologies in communication and computing, international conference on 0, 884 (2009)
19. Bossche, R.V.D., Vanmechelen, K., Broeckhove, J.: Cost-optimal scheduling in hybrid IaaS clouds for deadline constrained workloads. In: Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010, pp. 228–235 (2010)
20. Zhang, J., Gu, C., Wang, X., Huang, H.: A unified MetaScheduler architecture for telecom grade cloud computing. In: Information Science and Technology (ICIST), 2013 International Conference on, pp. 354–360 (2013). doi:10.1109/ICIST.2013.6747567
21. Calheiros, R.N., Ranjan, R., Rose, C.A.F.D., Buyya, R.: CloudSim: a novel framework for modeling and simulation of cloud computing infrastructures and services, CoRR abs/0903.2525 (2009)
22. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R.: CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software - Practice and Experience **41**(1), 23 (2011)
23. USDE: Voluntary reporting of greenhouse gases. Appendix F. Eletricity emission factors. Tech. rep. (2007)