Cloud resource management to improve energy efficiency based on local nodes optimizations

Belen Bermejo*, Carlos Guerrero, Isaac Lera, Carlos Juiz

Computer Science Department, University of the Balearic Islands. Crta. Valldemossa km 7.5, Palma, E07122, Spain.

Abstract

The energy consumption of Cloud Computing systems is one of the current concerns of systems architects. In order to reduce the energy consumption, they have provided techniques which go through the design of locations for data centres, together with techniques for the proper management of resources, taking into account the energy consumption of the system. This paper presents a resource allocation technique that maximizes the system efficiency. This technique is based on taking decisions at two levels: physical machine level and overall system level. Each of the levels ensure its own proper performance. To test that the technique complies with the initial hypothesis, simulations through the CloudSim tool have been developed. Also, we compared the results with a resource allocation technique based on a full knowledge of the system. As a result, we obtained a better solution time with the proposed technique that the other technique. Moreover, we obtained a lesser use of intra cluster network and a more energetically efficient cloud system.

© 2016 The Authors. Published by Elsevier B.V.

Keywords: Cloud computing; Resource allocation; Virtualization; Energy efficiency

1. Introduction

The definition of Cloud Computing, which has grown in acceptance, was created by NIST1: Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and service) that can be rapidly provisioned and released
with minimal management effort or service provider interaction. This paradigm is characterized by continuous changes in the environment and in the requirements to be met. Continuous changes occur autonomously and unpredictably, and they are out of control of the cloud provider.

Virtualization is a necessary technology that enhances Cloud Computing by allowing the creation of multiples Virtual Machines (VM) over the underlying hardware and improves resource utilization. A VM is a software implementation of the Instruction Set Architecture (ISA) at the application layer of the system that abstracts the application layer from the hardware layer. Each of the VM operates as an independent computation unit to execute various applications.

Recent studies have shown that the main challenges that cloud providers have to face are: reduction of energy consumption, performance level improvement and availability and dependability enhancement. Moreover, energy management is becoming a priority in the design and operation of complex service-based systems, as the impact of energy consumption associated with physical infra-structure is steadily increasing.

This paper proposes an allocation-resource technique based on open loop control system. The technique is based on acting at different levels: the lower level makes decisions locally on each Physical Machine (PM), and the highest level acts on the overall system taking into account local decisions. It contributes to the state-of-the art in cloud Computing research by improving the energy efficiency systems and providing new insight into the allocation-resource techniques.

2. Related work

One of the task to reduce the energy consumption in a cloud system is tend consolidating VM in PM, that is, concentrating the workload in the fewest possible PM. Thus, the energy consumption will be reduced. The disadvantage of that is that the performance system could be harmed. For this reason, one of the areas where it has been working is on the allocation resources. That is, distributing the VM through the system as efficiently as possible, in terms of performance.

Resource management is a core function of any man-made system. It affects the three basic criteria for the evaluation of a system: performance, functionality and cost. An inefficient resource management has a direct negative effect on the performance and the cost and an indirect effect on the functionality of the system. Studies about resource-allocation, show techniques in order to monitoring the performance and availability system. On the other hand, there are studies that propose techniques for the control of energy consumption in cloud systems through the resource management.

All these studies, referring to performance and availability system, make use of a centralized model of the system, that is, there is a central entity, which knows the system state, at all time. This allows knowing, among other things, in which PM can be hosted a VM. The disadvantage of these techniques is that the knowing of the system state is necessary to make a decision. On the other hand, this fact implies that the decision-making takes place more slowly, and the internal communication network slows down. The latter is because the central entity needs to know the state of all PM, and for this, it needs to access them. Moreover, it has not taken into account the energy consumption in the decision-making, only the performance system.

Also, studies concerning the energy consumption of these systems make use of a centralized model, making decisions as a whole, taking into account the energy consumption in the decision-making.

3. System model

In this section, we introduce a model of IaaS provider, which consists of actors and allocation techniques. The actors are users and IaaS providers. The system consists of an application layer and the infrastructure layer functions. This system distributes cloud services to users who have a SLA (Service Level Agreement) with the service provider.

The system consists of a number of PM, which allocates VM. In our case, users make a request to allocate a VM in the system. When the user decides to make this request, the system will allocate this VM to the corresponding PM, making the overall system as efficient as possible.
Each of the PM has a local controller, which keeps the PM in its most efficient state. An efficient state refers to a situation with a better relationship between performance and energy consumption. This local controller is the only one that knows the state of the PM, and it proposes changes in the PM to improve it. Thus, information has not to be exchanged between PM’s performance and energy consumption.

On the other hand, there is a general controller, which is responsible for analysing the changes proposed by each of the local controllers located at each of the PMs, and send to the actuator the best action among the possible candidates in that moment. The choice of one action or another will aim to ensure the health of the system as a whole, not of any particular PM.

The advantage of this model over the traditional is that PMs provide the information about their needs, only when a significant change occurs in them (the computing is relocated to the PM). This implies that the number of network messages decreases, and it will not need to monitor the status of PM when a new VM need to be allocated.

Therefore, two levels of actions are defined and we defined algorithms for optimization in both levels. Different controllers may define switches on the state of the PM that is expressed in actions. These actions which can be executed on the level of the PM are as follows: to migrate a VM from one PM to another one; to turn off a PM; to delete a VM from one of PM; to create a VM into a PM; to turn on a VM that was turned off.

The aim of the local controller is to ensure the health of its PM. Therefore, we should define the performance metrics that determine a state of a PM, the efficiency of a given resource ($E_i$). From a PM, we will take into account the usage percentage of the CPU ($CPU$), the usage percentage of the main memory ($MEM$) and the energy consumption of a resource ($CONS$). Moreover, related to VM, we will take into account the internal percentage of CPU and memory usage, but not the percentage consumed by the VM on the PM. Also, we can define the measure of the efficiency of each resource by taking into account the PMs.

$$E_{CPU} = \frac{CPU_{PM}}{CONS_{CPU_{PM}}} \quad E_{MEM} = \frac{MEM_{PM}}{CONS_{MEM_{PM}}}$$

(1)

It is necessary to highlight the conditions on which the system shown above is based. The system is centralized, i.e. it only consists of a cluster. The latency of the network within the cluster is regarded as constant, so it will not be considered when designing the resource allocation algorithm. All PMs consist of a unique processing unit and a unique main memory. The PMs could not have the same features between them, and all users will be of the same type, that is, only one type of SLA is contemplated. This means that there will be only one queue to store the users’ requests.

4. Algorithms and strategies

Two action levels are defined to optimize the energy efficiency of the system. The first and lower level is the PMs, and the higher level, the controller.

4.1. Physical machine level

At the lowest level, each PM will host a number of VM, and each represents a different customer. These VM do not share information among them, they are totally independent. Each PM stores a history of states, where a state is defined as a vector that contains the following parameters:

- ($<CPU(VM_i),MEM(VM_i)>$): a list that contains the use of the CPU and the main memory of each VM that are hosted on the PM.
- $CPU_{PM}$: the CPU usage of the PM.
- $MEM_{PM}$: the main memory usage of the PM.
- $N_{VM}$: the number of VM hosted on the PM.
- $CONS_{PM}$: the power consumption on the PM.

The definition of state from these parameters has an advantage, as it is not necessary to know what the energy consumption of VM is, since the consumption of the PM is associated with a certain configuration of VM.
The goal at the PM level is to reach the higher energy efficiency point, whether the utilization of PM is at the maximum level or not. The energy efficiency point is the level of utilization of the system in which the ratio between performance and energy consumption is the highest. Some systems have the energy efficiency located at 100% utilization and the best strategy, in those cases, is to allocate the maximum number of VM in each PM. But the most common systems are those in which the energy efficiency point is located below 100%. It is in those systems in which our proposal has a better behaviour.

To reach this objective we have designed a process that will be run on each PM. In general, this process will take as input its own current and historical state of the PM, and it will generate recommended actions that would be made on the PM to achieve the most efficient state. Together with the actions, a value representing the importance level of this action will be calculated. The higher value of importance has a greater priority than the other values, and then, the action. First, we need to obtain the possible configurations that the PM could have:

- For each sample of the historical, the vector formed by the efficiency of CPU, the efficiency of the main memory and the number of VMs will be obtained. Each component will have the same importance, and thus the same weight (1/3).
- Euclidean distance between the current status and the optimum settings are calculated. It is said that a configuration is optimal or accepted when it is within the limits of efficiency that PMs accept. We will store those settings that meet the efficiency requirements.
- The distance of the current configuration to the total shutdown of the PM state is also calculated.

Once that it is set the state that needs to be reached, it is necessary to determine what needs to be carried out on the PM to achieve those configurations. If the setting is “turned off”, the actions to be carried out will be to deallocate each of the VM and finally shut down the PM. However, if the number of VM of the current state is less than the optimal configuration, they will have to deallocate as many as VM exceed the optimal setting. Those VM which are less similar in features to the optimal setting VM will be deallocated. Otherwise, if VM are required to achieve optimal configuration, the PM will be to accept VM which make the two settings be as similar as possible to the features of VM.

Once all possible options have been calculated, the PM also informs the global controller of the resources that are available, thus the global controller will know in which PM new VM can be allocated, if it were necessary.

In short, the output will be a series of blocks of actions with the weight of each action, where each block represents a configuration where the weight of the block as a whole is the sum of the weights of the actions. Each PM sends the block of shares in a single message to the global controller to take the right decision.

4.2. Controller level

The higher level of acting is the global controller level. This element is in charge of assessing all requests that it receives from the PM and ordering a series of actions in the system as a whole. The actions will be chosen for the system to improve their status. Moreover, the controller will also receive requests from new customers who demand service.

Customers will be stored in a single queue, which gives access to the controller. They will be transformed into allocation requests VM with a determined level of utilization of the CPU and the main memory. All requests have the same priority.

The queuing model that represents the controller is the M/M/1 type. The customer arrival process is a homogeneous Poisson $\lambda > 0$ rate. In this case, the server (the global controller) is always busy and the output process of customers is a homogeneous Poisson $\mu > 0$. And the system has a unique server and infinite capacity$^{11}$.

The strategy that will be followed by the global controller to resolve the two sets of requests is based on simple operations such as sorting by priority and creating similar partners. Thus, the controller will run very simple actions to perform, so that the time needed to find a solution is lower than the time required by other approaches for VM allocation. The users are not compromised by excessive time to find a solution.

The first group of requests that the global controller will address are those of new users, so that it is ensured that all new users are served. The user request and requests for acceptance of VM from PM are sorted by priority. For each allocation request, the most similar acceptance request which is sufficient to accommodate the new VM will
be checked. In the event that there are not enough acceptance requests, the global controller creates a new VM in the most suitable PM. Thus, the time for allocation is short, not reaching the best possible allocation, but it avoids expanding the user waiting time.

Once new users are served, the rest of acceptance requests and deallocation request will be resolved. Deallocation requests need to be attended by complete blocks, that is, either all VM or none are evicted. Therefore, it prevents PM from remaining underutilized, thus creating a low efficiency of the machine. Hence, for each of the deallocate blocks it will look for the most adequate acceptance request. As with the first group of requests, the request for acceptance has to be similar in order not to underutilize resources and to accommodate the VM. If the acceptance requests are not sufficient, the global controller will create new VM on the most suitable PM.

In terms of events, each time the global controller receives a new request from users, it analyses the information of the PM that have been provided at that time, that is, there is no synchronization between the PM and the global controller.

5. Performance evaluation

We use CloudSim as a cloud environment simulator to implement our algorithm within this environment. The evaluation is based in the performance comparison of the proposed technique with two cases based only on the utilization level of the nodes: (a) maximize the utilization of the nodes to reduce the number of active nodes; (b) minimize the utilization of the nodes to increase the number of active nodes and increase the availability of resources in the system. The comparison is based on the number of message interchanged among the network, the execution time of the allocation algorithm and the overall power consumption and resource utilization (processor and memory).

The machine where simulations have been performed is equipped with an Intel i5 processor (4 cores - 2.53 GHz) and 4GB RAM. Also, note that the cache memory has been released from each execution.

The scenario on which the simulation was performed consists of 10 PM, which have random and different characteristics from each other. Each of the PM presents an initial state, defined as shown in Section 3. Also, each PM has an history of states. Both, the initial state and the historical were generated randomly. The number of allocated VM and new users are generated randomly too, and the features of these VM. The power consumption data for different levels of system utilization has been determined using a G4 Proliant machine.

5.1. Message number

In the utilization-based techniques, the resource allocation algorithm needs to know the state of each PM, and then, the PM has to send the state to the algorithm. Therefore, the number of messages sent will be always twice the number of the PM there. Instead, using the proposed technique, a maximum number of messages to be sent over the network will be equal to the number of the PM, since they are the ones who disclose their status.

In figure 1.a, we can see graphically the behaviour of this parameter. It has varied the number of PM, up to 10. The number of VM to allocate has followed an exponential distribution.

As the simulation results, the dotted line represents the theoretical behaviour of the based-utilization technique, be it low or high. The refractivity with the theoretical result obtained in the simulation carried out. The dotted line represents the theoretical behaviour of the proposed technique. We can see that the maximum curve that the number of message reach, will be maximum the number of PM. On the other hand, the solid line represents the behaviour obtained from the simulation, where we can check that the number of sent messages never exceeds the maximum number of messages of the theoretical model.

In any case, it notes that both the theoretical model and the simulation performed gives better results than the utilization-based techniques, managing to reduce the utilization of the internal cluster network.

5.2. Controller service time

The user response time is the sum of service time of the controller (S), the waiting time in the queue (τq) and the time it takes the system to accommodate the VM (τa). It is assumed that all customers expect a constant time in the queue, and that time hosting the VM is constant among all solutions: \( R = S + \tau_q + \tau_a \)
The service time of the controller can be decomposed into three components: the time to sort and classify requests from PMs \( t_s \), the time to resolve the requests of new users \( t_u \) and the time to resolve PM’s requests \( t_p \): \( S = t_s + t_u + t_p \)

Another important and decisive aspect in the response time of the system is the computational cost of the algorithms that perform the PM controller and global controller. The algorithm used to do the sorting is the Dual-Pivot Quicksort. Therefore, where \( m \) is the number of requests, the upper bound of computational cost is given by: \( O(m) + O(m \cdot \log(m)) = O(2m + m \cdot \log(m)) \). In the case of utilization-based techniques, the relevant transactions are the tour of each of the PMs and the time it takes the algorithm to find a decision, that is, of all possible options, find the suitable. Then, being the number of PM where \( n \), the upper bound of computational cost is \( O(n) + O(n) = O(2n) \).

The number of new system requests is determined by the rate of arrival. It may be the case that this arrival rate is greater than the number of PM system. But you can also give the opposite case, so that the values of \( n \) and \( m \) have no mathematical relationship. Moreover, in the executed simulation, \( n \) has remained constant, at 10, instead \( m \) has far exceeded this value.

In figure 1.b, it is shown how evolves this parameter depending on the number of users, i.e. VM. The results were obtained from the executed simulation. The dotted line shows the time evolution of service-based techniques used, however, the solid line shows the evolution regarding the proposed technique.

It can be seen that the service time is greater in the technique proposed. Although this fact, in the proposed technique, reducing the information transmitted by the network is significant, and the complexity of the same. Therefore, although the calculation time is greater by the controller, the information you receive will be easier to analyse and the network is less collapsed.

5.3. Overall optimization

The number of PM for this experiment was set to 10. First, we can see the parameters for each of the 10 PM in its initial state and then those same values once the execution of the allocation-resource technique, both those the utilization-based as proposed technique.

Figure 2.a shows the number of VMs housed in each of the PM. It can be seen that at the end of the simulation, the technique that allocate more VM in the PM is the based on maximum use. This is because this technique strengthens the greatest possible number of VM in an PM, thus more PM can be turned off. Also, it notes that the proposed technique is not far from the best in this case, so we can say that we are also consolidating VMs.
Regarding the use of resources of the PM, it can be seen in the figures 2.b and 2.c, corresponding to the use of the CPU and main memory that from the consolidation of VM in the PM, a greater utilization is achieved on maximum-utilization-based technique and on the proposed technique.

As the energy consumption of the PM, it has previously said that as utilization grows, more efficient is the PM, on energy consumption is concerned. In figure 2.d, consumption of each of the PM is displayed, and shows that the techniques that have consolidated more VM are those involving increased its consumption in the PM. That is, on the maximum-utilization-based technique and on the proposed technique, the energy consumption is higher. Nevertheless, the PM as a local level is more efficient.

Table 1 shows the final state of the whole system after applying the above techniques. Related to energy consumption, we can say that with the application of proposed technique, it has succeeded in reducing the energy consumption of the whole system. Also, it has been achieved, increasing the utilization of the system as whole, so the system is more energy efficient.

As the distance to the optimum state, the proposed technique has an intermediate performance. This is because the proposed technique is more flexible in selecting the optimal state, as more options are valued than the utilization-based techniques. It has also succeeded in reducing the number of on-state PM. Finally, the number of VM migrated between PM return to have a middle ground. This implies that the utilization of the internal cluster network is reduced. It can be said that doing half of relocate action, very close to the best values are obtained, which is the maximum utilization of the PMs.

6. Conclusions

This paper includes a technique for allocation of resources in Cloud Computing environments, in order to minimize the energy consumption and maximize the performance system. To do this, it has been chosen to design a
technique based on decision-making at local level. This means that it operated at two levels, locally and at the level of the overall system.

The simulation results obtained with CloudSim show that the proposed technique has reduced the number of messages transmitted through the network. With about half of messages, it has achieved a result similar to the most favourable case, the maximum consolidation of the VM. The utilization of resources of CPU and main memory of the proposed technique is greater than utilization-based technique. Regarding to the energy consumption of the system, we can say that the proposed technique enables lower energy consumption. So, it can be said that with a considerable reduction in the network utilization, it has been able to get the system reaches its most efficient point to a reduction in energy consumption.

Table 1. Overall system comparison.

<table>
<thead>
<tr>
<th></th>
<th>Initial state</th>
<th>Minimal approach</th>
<th>Maximum approach</th>
<th>Proposed approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power consumption (W)</td>
<td>102.3</td>
<td>103.4</td>
<td>102.6</td>
<td>101.6</td>
</tr>
<tr>
<td>Number of physical machines</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Number or reallocations</td>
<td>-</td>
<td>41</td>
<td>138</td>
<td>77</td>
</tr>
<tr>
<td>Distance to highest efficiency point</td>
<td>-</td>
<td>42.2</td>
<td>12.7</td>
<td>27.7</td>
</tr>
<tr>
<td>Memory utilization (%)</td>
<td>59.9</td>
<td>60.3</td>
<td>48.2</td>
<td>55.3</td>
</tr>
<tr>
<td>CPU utilization (%)</td>
<td>47.4</td>
<td>49.4</td>
<td>43.5</td>
<td>50.3</td>
</tr>
</tbody>
</table>

References

1. Mell, P., Grance, T. The NIST definition of cloud computing. 2011