International Journal of Web Information Systems

Energy efficient and latency optimized media resource allocation

Masoud Nosrati Ronak Karimi

Received 18 October 2015
Accepted 31 October 2015

Energy efficient and latency optimized media resource allocation

Masoud Nosrati and Ronak Karimi
*Department of Computer Engineering, Collage of Engineering,
Kermanshah Branch, Islamic Azad University, Kermanshah, Iran*

Abstract

Purpose – This paper aims to provide a method for media resource allocation in Cloud systems for supporting green computing policies, as well as attempting to improve the overall performance of system by optimizing the communication latencies.

Design/methodology/approach – A common method for resource allocation is using resource agent that takes the budgets/prices of applicants/resources and creates a probability matrix of allocation according to the policies of system. Two general policies for optimization are latency optimization and green computing. Presented heuristic for latencies is so that the average latencies of communication between applicant and resource are measured, and they will affect the next decision. For gaining green computing, it is attempted to consolidate the allocated resources on smaller number of physical machines. So calculation formula of the price of each resource is modified to decrease the probability of allocating the resources on the machine with least allocated resources.

Findings – Results of proposed method indicates its success in both green computing and improving the performance. Experiments show decreasing 21.4 per cent of response time simultaneously with increasing tasks in the tested range. The maximum and minimum of saved energy is acceptable and reported as 79.2 and 16.8 per cent.

Research limitations/implications – Like other centralized solutions, the proposed method suffers from the limitations of centralized resource agent, like bottle neck. But the implementation of distributed resource agent is postponed to future work.

Originality/value – Proposed method presents heuristics for improving the performance and gaining green computing. The key feature is formulating all the details and considering pitch variables for controlling the policies of system.

Keywords Optimization, Energy efficiency, Resource allocation, Cloud media resource, Communication latency

Paper type Research paper

1. Introduction

The emergence of distributed computers lets users share their requirements across the network through a proper communication channel and reach the resources that are distributed all over the network (Nezarat *et al.*, 2015). It helps to create an integrated vision of system and utilize the capabilities of the whole of system for all users. Emmerich (1997) defines a distributed system as a collection of autonomous computers that are connected through a network and distribution middleware, and it makes computers enable to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility. High-performance computing (HPC) systems are created as a result of this approach.

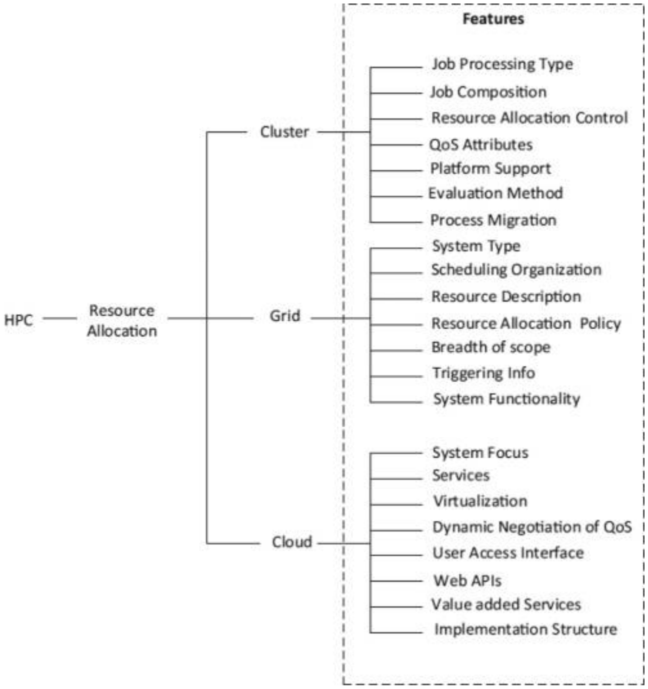
International Journal of Web Information Systems
Vol. 12 No. 1, 2016
pp. 2-17
© Emerald Group Publishing Limited
1744-0084
DOI 10.1108/IJWIS-10-2015-0031

Many applications of HPC systems such as industrial (Yi-wei, 2015; Esmaeilzadeh and Sattari, 2015), educational (Sharma and Kharel, 2015), medical (Edessy *et al.*, 2015) and commercial (Malekakhlagh and Meysamifard, 2015) came to existence after the provision of hardware infrastructures. HPCs are preferred rather than single computers for the following reasons (Hussain *et al.*, 2013):

- network connections which construct the spirit of distributed applications;
- availability of parallelism by executing parallel grains on different machines; and
- better reliability rather than single systems.

Previous studies categorize the distributed systems (DSs) into three types: Cluster, Grid and Cloud. In classic texts, Cluster is known as a distributed system with homogeneous nodes, and Grid is reputed for heterogeneous nodes (Tanenbaum and van Steen, 2007). But, in recent studies, there are Clusters with heterogeneous nodes implemented. It shows that homogeneity cannot be a good metric for classification. Because of it, Hussain *et al.* (2013) categorizes the DSs by their resource allocation features, as shown in Figure 1. In this study, we will get into the media resource allocation in Cloud systems. Cloud computing relies on sharing of resources to achieve coherence through the vision of a single computer for end-users.

Wide range of different factors influencing the Cloud systems causes difficulties for modeling them. For example, number of machines, types of applications, processing



Source: Hussain *et al.* (2013)

Figure 1. High-performance computing systems categories and attributes

load and many other important factors can be pointed out. Also, there are different types of services that can be considered as critical points of modeling, including Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Another important notion is the issue of virtualization that provides the ability of migration. Using this technique, virtual machines are generated and placed in real machines that provide wider range of facilities via VM migration for improving the performance of the system. The concept of migration lets a virtual machine to run on a real machine, then stop running and migrate to another real machine and resume its execution. It can be very helpful for making fault-tolerant systems or optimizing the energy consumption. Also, it is widely utilized in load balancing. The purpose of this technique is to transfer the computation from a physical node to another node which seems to be more appropriate for accomplishing the task. Another challenging issue is Quality of Service (QoS). QoS can be set in run time or statically when designing the system. Run time QoS adaptation is called "Dynamic QoS Negotiation" (Pinel *et al.*, 2011). For example, in a research by Pinel *et al.*, it is mentioned that DQoSN can be implemented by a special entity or through self-algorithms (Pinel *et al.*, 2011).

The challenges that were talked can vividly indicate the delicacy and toughness of resource allocation in Cloud systems. Accordingly, Sharkh *et al.* classify the recent studies in Cloud RA in three categories (Sharkh *et al.*, 2013):

1. studies focusing on processing resources like Maguluri *et al.* (2012) and Alicherry and Lakshan (2012);
2. studies focusing on network resources like Sun *et al.* (2012) and Kantarci and Mouftah (2012); and
3. studies focusing on power and energy resources like Srikantaiah *et al.* (2009) and Chase *et al.* (2001).

Also, Sharkh *et al.* (2013) states that challenges are internal or external. External challenges include regulative and geographical challenges (it is about the issues that distribution of system will cause in geographical location and regulative and security) and charging model issues (it is about the charges for customers to subscribe the Cloud system). On the other hand, internal challenges include the data locality: combining compute and data management; reliability of network resources inside a data center; and software defined networking design challenges inside the data centers. SDN is a networking paradigm that puts the forwarding plane and software control in different layers. Details are mentioned in the study by Sharkh *et al.* (2013).

One of the important challenges in real-world implementation of distributed systems is the power and energy consumption. Different strategies are introduced for optimizing the energy consumption. Most of them try to aggregate the resources on smaller number of servers to shut down the non-busy ones. Resource aggregation has a trade-off with the performance of whole system. Also, it may cause a bottle neck on the I/O of the running servers. It can affect the QoS of system, too. Average response time will be increased. So the studies that aim at improving both performance and optimizing the energy consumption are needed to be conducted.

As the conclusion of this section, it should be restated that in every resource allocation method, all the above-mentioned factors including processing capabilities, network resources and energy consumption must be considered. Regarding the trade-off among these items, the best configuration should be found and set according to the requirements of system. Besides, other issues like the portability and fault-tolerance should be paid attention. Recent studies go through these issues separately, and many approaches for optimization of resource management are offered by them.

In our previous research, we proposed an optimization on a common resource allocation method based on the latencies of communications to improve the performance of the system by minimizing the communication latencies (Nosrati *et al.*, 2015). Because of it, we added a table to resource agent (entity that allocates resources to the applicants) to hold the history of previous allocations. Then, we built up a probability matrix for the allocation of resources partially based on the history of latencies. In current study, we are going to enhance the previous work by involving a strategy for green computing to achieve less energy consumption via consolidation of allocated resources on smaller number of physical machines. The main idea is to decrease the probability of choosing the media resources that are located on physical machines that are used less than the other machines. When no resource is allocated on a machine, it will be turned off to save the energy.

In the rest of this paper, the related studies are abstracted. Then, we will have a brief look at a common media resource allocation method and the strategies that resource agent utilizes to choose the best resource for the best applicant. After pointing out the standard method of resource allocation, we offer our contribution based on the optimization of latencies between the resource and applicant nodes, and the strategy of energy efficiency will be talked. Results of simulation of proposed method show better performance, less execution time and less energy consumption for this method rather than the base approach. At the end of paper, discussion and conclusion on the features of proposed method with the suggestions for future work is presented.

2. Related work

In Kusic *et al.* (2009), authors aim at reducing the power consumption by consolidation of virtual machines on smaller number of machines. Also, minimizing the performance loss is the secondary goal of this research. But the major limitation is with the high execution time that is caused by its complex model of optimization (Gupta, 2015). Another research by Song *et al.* (2009) attempted to maximize the resource utilization and fulfill the performance requirements. Their method had nothing to do with network or media resources and the emphasis was on computational capabilities. Buyya *et al.* (2010) aimed at minimizing the energy consumption by leveraging the heterogeneity of cloud data centers to provide the green computing. But they faced two major limitations:

(1) lower level of performance that was caused by its trade-off with consolidation; and

(2) the probability of SLA violation.

Ye *et al.* (2010) take the live migration into account to consolidate the resources dynamically, as what Beloglazov and Buyya (2010a, 2010b) do. In an eminent research by Xhao *et al.* (2013), authors presented a system that used virtualization to allocate media resources dynamically based on application demands. Their method supported green computing via optimizing the number of servers in use. They introduced the concept of “skewness” for measuring the inequality in the multi-dimensional resource utilization in a server and attempted to minimize the skewness to combine different types of workloads and improve the total utilization of server resources. Beloglazov and Buyya in Dhingra and Paul (2013) attempted to focus on the optimization of resource allocation with maximum utilization, minimum migration and minimum overhead. Their method supported the virtual machines migration for dynamic media resource allocation (Tang *et al.*, 2014).

In the most relevant study to our work, Tang *et al.* (2014) took the credibility of the resource into account for resource allocation. Credibility is considered as a quality attribute for the data and denotes that the performance of the network is affected by information credibility and the timeline for finding the power supplier. Considering the credibility of the resource can help achieve better performance. Because of this, the allocation probability matrix P should be modified so that the probability of allocating the low-credible resources decreases, while the probability of allocating the high-credible resources increases.

3. Common resource allocation method

In this section, a common resource allocation method (that is called standard method in the section of experimental results) is talked that many studies like Zhang *et al.* (2013), Sun *et al.* (2010) and Zhang and Zhu (2013) utilized it for their resource allocation optimization (Nosrati and Karimi, 2016). In this method, a resource agent is considered as the entity that performs the resource allocation. According to Figure 2, both resource owners and resource applicants send their costs to resource agent. General policy of resource agent is to sort the requests and resources and totally allocate the applicants with highest budget to the resource with lowest price (Tang *et al.*, 2014).

Let $U = \{u_1, u_2, u_3, \ldots, u_m\}$ be the set composed of m resource applicants; each task of resource applicant u_i is t_i, so the task set of U can be described as $T = \{t_1, t_2, t_3, \ldots, t_m\}$. And t_i has four attributes $t_i = \{tid_i, l_i, b_i, d_i\}$, where tid_i is the ith task's identify, l_i is the ith task's length, b_i is the ith task's budget and d_i is the deadline of the task.

Let $R = \{r_{1,} r_{2,} r_{3, \ldots,} r_n\}$ be the set composed of n resources, and each r_j has five attributes $r_{j\,=}\ \{rid_{j,} cpu_{j,} st_{j,} lp_{j,} hp_j\}$, where rid_j is the jth media resource's identify, cpu_j is the jth media resource's computing ability of solving the task, st_j is the start time to deal with a new task (i.e. the current workload of resource r_j), lp_j is the jth media resource's lowest price and hp_j is the jth media resource's highest price.

The media resources allocation probability matrix is shown as P, in which each p_{ij} is the probability of resource j to be allocated to applicant i:

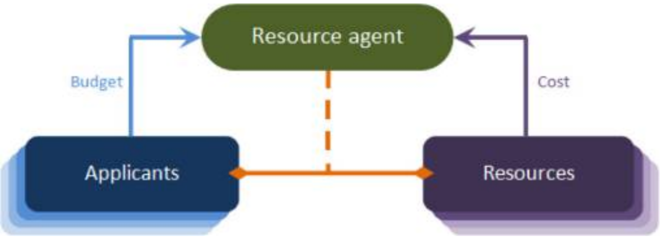


Figure 2.
Cloud resources allocation environment

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & \dots & p_{mn} \end{bmatrix} s.t.\, 0 \le p_{ij} \le 1, \ \sum_{i=1}^{m} p_{ij} = 1, \ \sum_{j=1}^{n} p_{ij} = 1$$

Before making any decision about resource allocation, the budget of applicant and the price of resource should be calculated and submitted to resource agent. It is important to consider the following point all the time:

- Resource must be capable enough to process the request of applicant in the deadline: $d_i - st_j - l_i / cpu_j \ge 0$
- The price of resource must be less than or equal to the applicant: $b_i/l_i \ge lp_j$
- The budget of applicant must be at least equal to the average of the price of remaining resources: $\left(\overline{lp} = \left(1/n\right)\sum_{j=1}^{n} lp_j\right)$

The budget of applicant is calculated from equation (1), where $bid_i^{resource}$ has an inverse relationship with the number of remaining resources. It means when the number of unallocated resources is decreasing, proposed budget of applicant for the resource will be increased and vice versa. Other impressing factor is average remaining time that Anthony and Jennings (2003) calculated it as equation (2). Total budget based on equation (3) is the sum of both equations (1) and (2) with the weights of α' and β'. Weights might be changed as the pitches for controlling and satisfying the policies of the system:

$$bid_i^{resource}(t) = \overline{lp} + \left(\frac{b_i}{l_i} - \overline{lp}\right)\left(1 - \frac{n_i^t}{n_i^{max}}\right)^{\frac{1}{\alpha}} \tag{1}$$

$$\overline{rt}_i(t) = \sum_{j=1}^{n} \frac{(rt_{ij}(t)\omega_{ij})}{n_i^{max}}, \ \omega_{ij} = \begin{cases} 1 & if\ rt_{ij}(t) \ge 0 \\ 0 & otherwise \end{cases}$$

,

$$bid_i^{time}(t) = \overline{lp} + \left(\frac{b_i}{l_i} - \overline{lp}\right)\left(1 - \frac{\overline{rt}_i(t)}{rt_i^{max}}\right)^{\frac{1}{\beta}} \tag{2}$$

$$bid_i(t) = \alpha' bid_i^{resource}(t) + \beta' bid_i^{time}(t) \tag{3}$$

$$0 \le \alpha', \beta' \le 1$$

In these equations, n_i^t is the number of remaining resources in the time t, which can be applied by applicant t_i, and n_i^{max} is the maximum number of resources that might be applied by t_i. Remaining time of t_i which is utilizing r_j is calculated as $rt_{ij}(t) = d_i - st_j - l_i/cpu_j$. Let rt_i^{max} be the maximum time of waiting for t_i. Different applicant's budget curve can be adjusted by changing α and β. In Figure 3, different values of

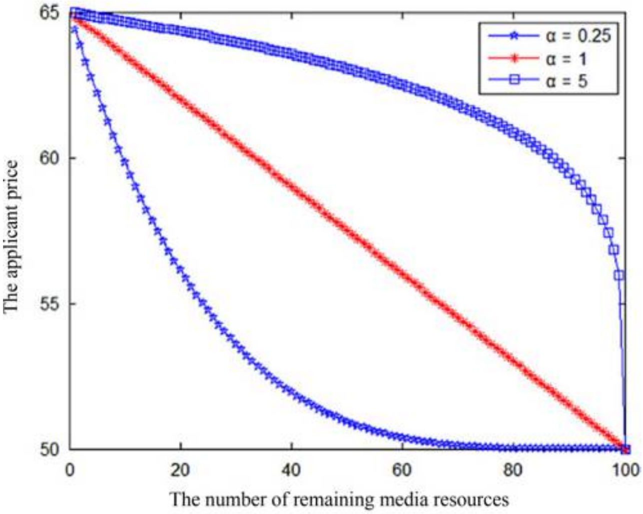


Figure 3.
Resource applicants' price considering remaining resources with different values of α

Source: Tang *et al.* (2014)

α are shown. It has the same shape for β and σ (that will be introduced in next section).

After the calculation of the budget of applicant, it is time to calculate the price of the resource. General policy of the resource owner is to service the applicant with the highest budget to increase the utilization of the resource. In other words:

$$rp_j(t) = lp_j + (hp_j - lp_j)\left(\frac{st_j(t)}{wl_j(t)}\right)^{\frac{1}{\sigma}} \tag{4}$$

where $rp_j(t)$ is the price of resource at time t and $st_j(t)$ is the current workload or the workload at the start time of the task at time t. $wl_j(t)$ is the workload of r_j after the last allocation. In this equation, general trend is to decrease the price of the resource when the allocated resource is going to finish the task to let other applicants to apply for it easily and with less loss of time.

After the calculation of both budget and price, they are submitted to resource agent. It sorts the applicants' budgets in descent order and the resource prices in ascent order. Then, according to equation (5), final price is calculated as the average of the richest applicant (bid_i^{max}) and cheapest resource (rp_j^{min}):

$$fp(t) = \frac{1}{2}(bid_i^{max}(t) + rp_j^{min}(t)) \tag{5}$$

Matrix P is then constructed according to the final prices. Resource allocation methods utilize this P for binding the resource to the applicant. But an important issue is optimizing the values of matrix P to achieve more valuable goals including green computing and higher performance.

This section was mostly adopted from the study by Tang *et al.* (2014), and readers can refer to it for further details about the construction P and strategies of applicant and resource owners.

4. Optimization of matrix *P*

Our contribution includes two parts: first, the energy efficiency issue is considered with the modification of resource price calculation; and second, matrix P is modified with latencies of communication to achieve better performance. Details of the optimization are described in following four sub-sections.

4.1 Energy efficiency issue

So far, many studies aimed at achieving an energy-efficient resource allocation scheme. A common way is the consolidation of resources on smaller number of physical machines. In this way, unused servers could be switched off. For optimizing the energy consumption by consolidating the media resources, two heuristics are considered. Our first heuristic is to increase the price of the resources on physical machines that are already turned off and decrease the price of resources that are placed on turned on machines. For the second heuristic, it must be noted that among those machines that are working, the price of the resources on the machine that hosts minimum number of allocated resources must be soared to spread the next allocations on other machines that host more allocated resources. In this way, the focus is on turning off a physical machine with minimum running resources, all the time. For the first heuristic, the formula of rp_j should be modified as follows:

Let $Phy_k = \{R_d, stt\}$ be a physical machine that hosts $R_d = \{r_1, r_2, \ldots, r_d\}$ resources and the stt shows the state of machine as:

$$stt = \begin{cases} +1 & machine \ \ is \ \ on \\ -1 & machine \ \ is \ \ off \end{cases}$$

The rp_j for resources on machines that their state is $+1$ must be decreased, and the other way round is true for the state of -1. The amount of changes must be specified first. There might be different approaches for this. We utilized the standard deviation as a coefficient of rp_j to change it according to our policy:

$$\delta = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(rp_j - \overline{rp})^2} \ s.t. \ r_j \ is \ not \ allocated$$

where δ is the standard deviation of for all the n resources in the system which are not allocated and $\overline{rp}$ is the average of resource prices.

New value for rp_j can be calculated based on the state of the machine and the δ coefficient as:

$$rp_j(t) = rp_j(t) - \left(stt \times \tau \times \frac{\delta}{g}\right)$$

where τ is a pitch for controlling the effect of coefficient of δ and g is the number of resources that are hosted by the physical machine in which r_j is hosted. Please note that the modest amount of $\tau = 0.5$ is set for our experiments, but it can be increased/decreased for changing the focus of the system on energy consumption issue, especially about the cost of switching on the physical machines.

For the second heuristic, let w be the number of all the resources on machines that are turned on and δ' be the standard deviation of all w resources that are not allocated:

$$\delta' = \sqrt{\frac{1}{w}\sum_{j=1}^{w}(rp_j - \overline{rp})^2 \; s.t. r_j \; is \; not \; allocated}$$

New value for rp_j can be calculated by δ' coefficient as:

$$rp_j(t) = \begin{cases} rp_j' + \vartheta\dfrac{\delta'}{w} & machine \; with \; minimum \; allocated \; resources \\ rp_j' - \vartheta\dfrac{\delta'}{w} & other \; machines \end{cases}$$

where ϑ is a pitch for controlling the effect of coefficient of δ'. The modest amount of $\vartheta = 0.5$ is set for our experiments, but it can be increased/decreased for changing the focus of the system on energy consumption issue, too. By equation (7), the price of the resources on physical machine with least allocated media resources will increase, while the price of other sources will decrease. It helps to consolidate the usage of media resources on other physical machines and tends to turn off the physical machine with least unallocated resources. For this purpose, new value of resource price ($rp_j(t)$) will be sent to resource agent to take the energy efficiency issue into account while allocation.

4.2 Taking the communication latencies into account

Definitely, there might be a geographical distance between the resource and applicant that causes latencies in communication. These latencies affect the whole performance of the system and increase the average response times. These negative results will be emerged while the execution of fine-grained parallel programs. Trade-off between the latency factors that might be forked from the faults in the system of the geographical distance and the performance of the system encourages us for the optimization of matrix P to achieve better performance. The main contribution is to maintain a history of latencies. The history of latencies can be taken into account for constructing P. It will help to have better performance of the system when there are similar resources. Then, resource agent can consider the latencies as a part of the weight of fp.

The construction of the table of latencies is done gradually. After each resource allocation, a record is added to the table of latencies that shows the latency between t_i and r_j or modifies the previous records between them. Let resource r_j is allocated to applicant t_i for the first time. Latencies of messages communication can be measured through sending acknowledge packets. Acknowledge packets might be more than once submitted from applicant to resource (and vice versa) to collect the average of the latencies:

$$\overline{LC}_{ij} = \sum_{q=1}^{p} PL_q$$

where $\overline{LC}_{ij}$ is the average latency of communication between applicant t_i and resource r_j and PL_q is the latency of qth acknowledge message. Note that there are no lower and upper bounds for $\overline{LC}_{ij}$ $(0 \leq \overline{LC}_{ij} \leq \infty)$. Value 0 for $\overline{LC}_{ij}$ is when the resource and applicant are located at the same node and there is no communication latency, and value ∞ is when the resource is faced with failure and it sends no response to the acknowledge message. So it cannot be utilized directly in the P. Here, we need to change it to the scale of 0 to 1 to affect the P with the value of $\overline{LC}_{ij}$. Because of it, the average of all $\overline{LC}_{ij}$ that are stored in the specified table should be calculated as equation (8):

$$ALC(t) = \frac{1}{m \times n} \times \sum_{i=1}^{n} \sum_{j=1}^{m} \overline{LC}_{ij} \mid \textit{if there is a record in table oflatencies for } t_i \textit{and } r_j \tag{8}$$

where $ALC(t)$ is the average of latencies of communications between nodes t_i and r_j, where they were allocated previously and there is a record in the table of latencies for them.

Equation (9) generates a number between 0 and 1 for the modification of matrix P:

$$TLC_{ij}(t) = 1 - \left(\frac{\overline{LC}_{ij}}{\overline{LC}_{ij} + ALC(t)} \right) \tag{9}$$

where $TLC_{ij}(t)$ is the impact of total latency between applicant t_i and resource r_j at the time t. $TLC_{ij}(t) = 0$ means that the resource is unavailable, and it gave no response to the acknowledge packet $(\overline{LC}_{ij} = \infty)$; accordingly, $TLC_{ij}(t) = 1$ means that there is no latency and resource and applicant are located at the same node $(\overline{LC}_{ij} = 0)$.

Now, matrix LC can be constructed as equation (10):

$$LC = \begin{bmatrix} l_{11} & l_{12} & \dots & l_{1n} \\ l_{21} & l_{22} & \dots & l_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ l_{m1} & l_{m2} & \dots & l_{mn} \end{bmatrix} s.t.\, 0 \leq l_{ij} \leq 1, \sum_{i=1}^{m} l_{ij} = 1, \sum_{j=1}^{n} l_{ij} = 1 \tag{10}$$

where l_{ij} is the impact of latency between applicant t_i and resource r_j which is obtained from $TLC_{ij}(t)$.

4.3 *Modification of matrix* P *with LC*

Now, it is time to have a consequent matrix at resource agent to do the resource allocation efficiently. It should be pointed out that in all the systems, policies regulate everything. So some facilities to implement the policies should be considered. So we will put a weight on the P and LC to be able to control them. The consequent is matrix FP as in equation (11):

$$FP(t) = \frac{1}{\theta + \lambda} \times (\theta P + \lambda LC)$$

$$= \frac{1}{\theta + \lambda} \times \begin{bmatrix} \theta p_{11} + \lambda l_{11} & \theta p_{12} + \lambda l_{12} & \dots & \theta p_{1n} + \lambda l_{1n} \\ \theta p_{21} + \lambda l_{21} & \theta p_{22} + \lambda l_{22} & \dots & \theta p_{2n} + \lambda l_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \theta p_{m1} + \lambda l_{m1} & \theta p_{m2} + \lambda l_{m2} & \dots & \theta p_{mn} + \lambda l_{mn} \end{bmatrix}$$

where θ is the weight of P and λ is the weight of LC to implement the policies of system.

4.4 Side issues

Inherently, the proposed method has some features to detect the unavailable resources. In fact, $\overline{LC}_{ij}$ can indicate the availability of resource r_j. It can be a good feature for handling the faults of system by isolating the crashed resources. This issue will also assist the migration strategies to have a better performance. Replication is not out of this subject, too.

The way of calculation of $\overline{LC}_{ij} \rightarrow \infty$ to check their availability from time to time.

5. Simulation and results

The implementation of proposed method and analyzing the results lead to better understanding about its efficiency. To evaluate the performance of the proposed algorithm, we implement it by the CloudSim toolkit (Calheiros *et al.*, 2011). Each task is submitted according to Poisson distribution after its previous tasks; the length of each task is considered as a random number within 100,000-200,000; the number of tasks are considered between 100 and 1,000, while the number of resources is between 50 and 150 on ten physical machines; the deadline d_i of task t_i is set according to equation (12), and the budget b_i of task t_i is set according to equation (13) (Tang *et al.*, 2014):

$$d_i = st_j + random\left(\frac{l_i}{1.1 \times cpu_j}, \frac{l_i}{0.9 \times cpu_j}\right) \tag{12}$$

$$b_i = l_i \times random\ (0.9\overline{lp}, 1.1\overline{hp}) \tag{13}$$

where $\overline{lp}$ and $\overline{hp}$ are the average values of the media resources' lp and hp (Tang *et al.*, 2014).

Figure 4 shows a comparison between the response times of common method (which was talked in Section 3) and the proposed method with optimization based on the communication latencies between the nodes and energy efficiency policies. It is vividly seen that latency optimized method (which we will call LO) has the best response time rather than standard and EELO (energy efficient and latency optimized). In standard resource allocation, the common method (Section 3) is implemented, in which the tasks come into the system and are executed normally. In this way, latencies between the nodes are waivered. For example, when resource agent is allocating a resource to the applicant, it does not consider the factor of latencies. So it might choose a resource with a high level of latency to be allocated to applicant. It will cause longer response times. So the basic strategy is to let the resource agent to know about the latencies among the nodes of resources and applicants. Then, it can include the factor of latencies to make decisions about choosing the resources for allocation. This advantage improves the response times, especially in the case of increasing the tasks. Finally, the EELO is located between the standard and LO. It is because of the energy efficiency strategy. As we saw, the price of resources is changed to lead to smaller number of physical machines to be switched on. This issue has a trade-off with plain latency optimization. According to Figure 5, it is clear that the energy consumption has an impressive abatement in EELO. Results indicate a maximum of 79.2 per cent and minimum of 16.8 per cent energy saving in the workload range of our experiments. On the other hand, the maximum reported improvement is 29.3 per cent for LO in the *task number = 1,000* and 21.4 per cent for EELO at the point *task number = 840*. Figure 6 shows the effect of different values of τ and ϑ on the number of powered on machines. In Figure 6(a), the moving average of machines is depicted. It almost shows that with increasing the values of pitch variables τ and ϑ, the number of powered on machines will be decreased. To clarify this issue, the logarithmic trendline of number of machines is shown in Figure 6(b). Experimental results show the reverse relationship of τ and ϑ with the number of physical machines that are powered on.

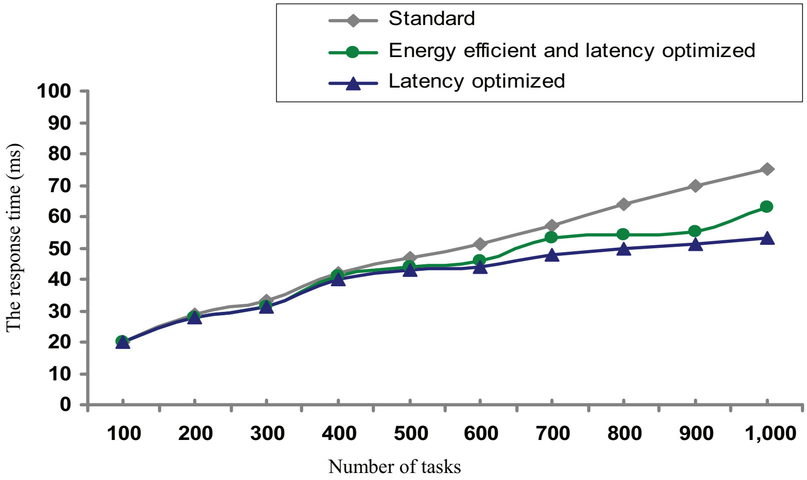


Figure 4.
Comparison of response time between the standard and latency optimized and energy-efficient latency-optimized methods

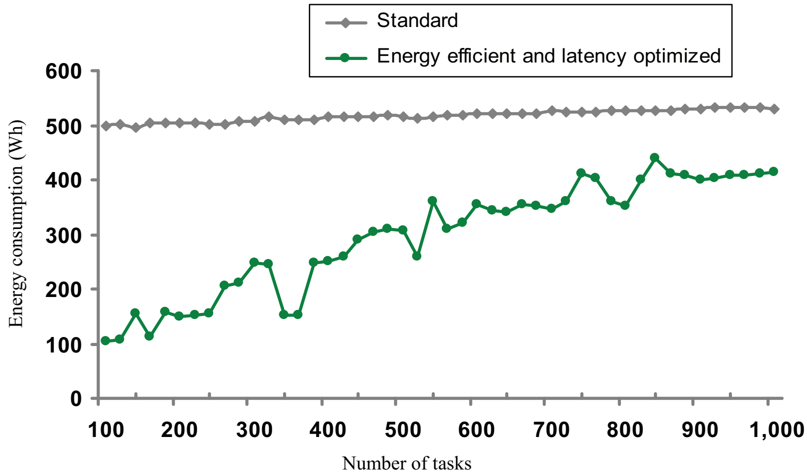
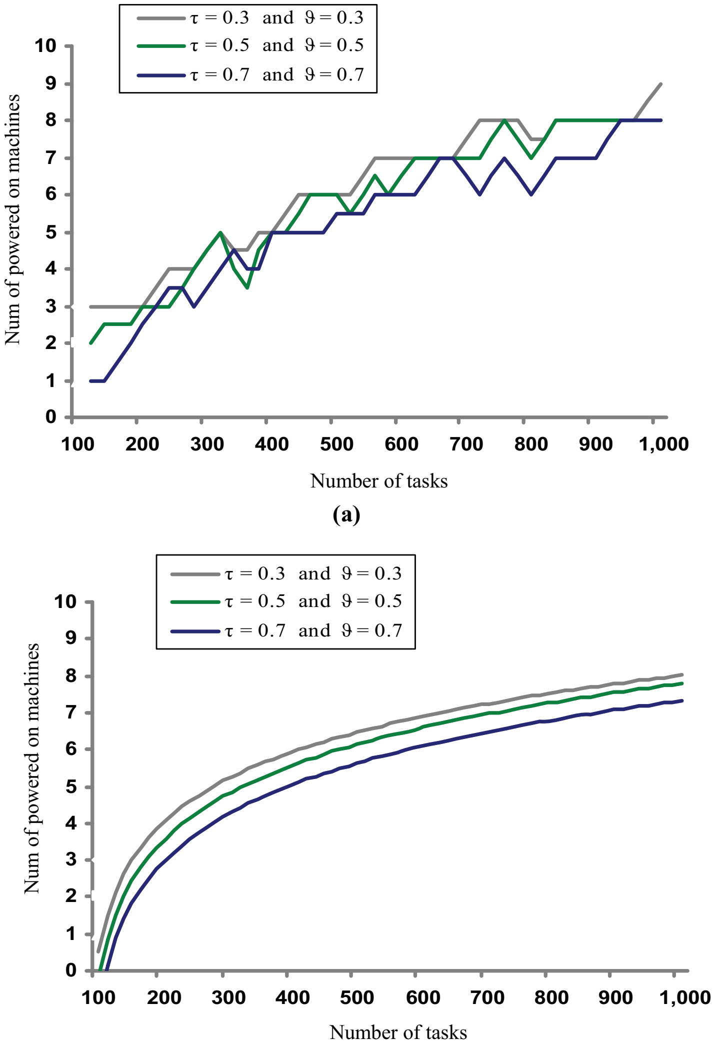


Figure 5.
Comparison of energy consumption between the standard mode and energy-efficient latency-optimized method

6. Discussion and conclusion

The impact of communication latencies on total performance of Cloud systems encouraged us to optimize the media resource allocation for improving the overall performance. Almost in all distributed systems, the common way is to count the resource allocation as a duty of resource agent. Both resources and applicants calculate their prices and budgets and send it to the resource agent. Resource agent then makes the decision to allocate the most



(a)

(b)

Figure 6.
Effect of different values of τ and ϑ on the number of physical machines that are powered on

Notes: (a) Moving average of physical machines that are powered on; (b)logarithmic scheme appropriate resource to the best applicant. Because of it, resource agent constructs the matrix P (as in Section 3) based on the prices and budgets. Optimization of P can lead to better performance. In our method, we attempted to do the optimizations to achieve two general goals. Our first heuristic was for choosing the resources with the least latencies in communication with applicant node. Because of this, we considered a table in resource agent that holds the history of the resource allocation bindings with their average latencies. At first, this table has no record, and after each allocation, a record is added or updated. For next allocations, the values of latency impact will be taken into account for making decision. Accordingly, Matrix FP is constructed where $FP_{ij}(t)$ is the possibility of resource r_j to be allocated to applicant t_i. This value is obtained from matrix P and the average latencies of previous allocations. On the other hand, this method was potent to detect the failure of resources by measuring the latency of communications; which is an important point for the other issues like migration, resource replication and fault-tolerance. The second heuristic was about the energy efficiency, so that we established policies for calculating the price of resources to consolidate the allocated resources on smaller number of physical machines. In this way, some of the machines could be turned off, because they were not in use. For the evaluation of the proposed method, average response time was considered as a metric. Results indicate the better response time rather than standard method, especially by increasing the number of tasks and passing time. Besides, monitoring the energy consumption of the system indicated an acceptable amount of saving energy. Our method presented a centralized resource agent. As it is seen in other centralized solutions, proposed method suffers from the limitations of centralization, like bottle neck. But the implementation of distributed resource agent is postponed to future work. As the last notion, there are still vacant positions for future studies on the methods of resource consolidation, heuristics for distributing the resource agent, taking the live virtual machine migration into account for dynamic resource allocation, etc.

Conflict of interest statement: The authors declare that there is no conflict of interests regarding the publication of this article.

References

Alicherry, M. and Lakshman, T.V. (2012), "Network aware resource allocation in distributed clouds", *Proceeding of IEEE INFOCOM '12*, *Orlando, FL*, 25-30 March, pp. 963-971.

Anthony, P. and Jennings, N.R. (2003), "Developing a bidding agent for multiple heterogeneous auctions", *ACM Transactions on Internet Technology*, Vol. 3 No. 3, pp. 185-217.

Beloglazov, A. and Buyya, R. (2010a), "Energy efficient resource management in virtualized cloud data centers", *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid 2010)*, *Melbourne*.

Beloglazov, A. and Buyya, R. (2010b), "Energy efficient allocation of virtual machines in cloud data centers", *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid 2010)*, *Melbourne*.

Buyya, R., Beloglazov, A. and Abawajy, J. (2010), "Energy-Efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges", *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010)*, *Las Vegas, CA*, 12-15 July.

Chase, J.S., Anderson, D.C., Thaker, P.N., Vahdat, A.M. and Doyle, R.P. (2001), "Managing energy and server resources in hosting centers", 18th ACM Symposium on Operating Systems Principles, Banff, 21-24 October.

Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F. and Buyya, R. (2011), "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.

Dhingra, A. and Paul, S. (2013), "A survey of energy efficient data centers in a cloud computing environment", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2 No. 10.

Edessy, M., EL-Darwish, A.G., Nasr, A.A., Ali, A.A., El-Katatny, H. and Tammam, M. (2015), "Different modalities in first stage enhancement of labor", *General Health and Medical Sciences*, Vol. 2 No. 1, pp. 1-4.

Emmerich, W. (1997), "Distributed system principles, lecture notes", available at: www.0.cs.ucl.ac.uk/staff/ucacwxe/lectures/ds98-99/dsee3.pdf

Esmaeilzadeh, B. and Sattari, S.T. (2015), "Monthly evapotranspiration modeling using intelligent systems in Tabriz, Iran", *Agriculture Science Developments*, Vol. 4 No. 3, pp. 35-40.

Gupta, P.M. (2015), "A review on energy efficient techniques in green cloud computing", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 5 No. 3, pp. 550-554.

Hussain, H., Malikb, S.U.R., Hameedb, A., Khanb, U.S., Bicklerb, G., Min-Allaha, N., Qureshia, M.B., Zhangb, L., Zhangb, W., Ghanid, N., Kolodzieje, J., Zomayaf, A.Y., Xug, C.-Z., Balajih, P., Vishnui, A., Pinelj, F., Peceroj, J.E., Kliazovichj, D., Bouvryj, P., Lik, H., Wang, L., Chenm, D. and Rayesn, A. (2013), "A survey on resource allocation in high performance distributed computing systems", *Parallel Computing*, Vol. 39 No. 11, pp. 709-736, available at: http://dx.doi.org/10.1016/j.parco.2013.09.009.

Kantarci, B. and Mouftah, H.T. (2012), "Scheduling advance reservation requests for wavelength division multiplexed networks with static traffic demands", *IET Communications*, Vol. 2 No. 8, pp. 806-811.

Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N. and Jiang, G. (2009), "Power and performance management of virtualized computing environments via lookahead control", *Cluster Computing*, Vol. 12 No. 1, pp. 1-15.

Maguluri, S., Srikant, R. and Ying, L. (2012), "Stochastic models of load balancing and scheduling in cloud computing clusters", *Proceeding of IEEE INFOCOM '12, Orlando, FL*, 25-30 March, pp. 702-710.

Malekakhlagh, E. and Meysamifard, S. (2015), "Industry pathology to develop global market entry strategies: emphasizing on small and medium-sized enterprises", *International Journal of Economy, Management and Social Sciences*, Vol. 4 No. 2, pp. 188-193.

Nezarat, A., Raja, M. and Dastghaibifard, G. (2015), "A new high performance gpu-based approach to prime numbers generation", *World Applied Programming*, Vol. 5 No. 1, pp. 1-7

Nosrati, M., Chalechale, A. and Karimi, R. (2015), "Latency optimization for resource allocation in cloud computing system", Computational Science and its Applications – ICCSA 2015, Banff, 22-25 June, pp. 355-366, Springer International Publishing.

Nosrati, M. and Karimi, R. (2016), "Investigating a benchmark cloud media resource allocation and optimization", *World Applied Programming*, Vol. 6 No. 1.

Pinel, F., Pecero, J., Bouvry, P. and Khan, S. (2011), "A two-phase heuristic for the scheduling of independent tasks on computational grids", *ACM/IEEE/IFIP International Conference on High Performance Computing and Simulation (HPCS), Bologna*, July, pp. 471-477.

Sharkh, A.M., Jammal, M., Shami, A. and Ouda, A. (2013), "Resource allocation in a network-based cloud computing environment: design challenges", *IEEE Communications Magazine*, Vol. 51 No. 11, pp. 46-52.

Sharma, G. and Kharel, P. (2015), "E-participation concept and web 2.0 in e-government", *General Scientific Researches*, Vol. 3 No. 1, pp. 1-4.

Song, Y., Wang, H., Li, Y., Feng, B. and Sun, Y. (2009), "Multi-tiered on- demand resource scheduling for VM-Based data center", *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid (2009), Shanghai*, pp. 148-155.

Srikantaiah, S., Kansal, A. and Zhao, F. (2009), "Energy aware consolidation for cloud computing", *Cluster Computing*, Vol. 12, pp. 1-15.

Sun, D., Chang, G., Wang, C., Xiong, Y. and Wang, X. (2010), "Efficient Nash equilibrium based cloud resource allocation by using a continuous double auction", *Proceedings of the2010 International Conference on Computer Design and Applications (ICCDA)*, *Qinhuangdao*, 25-27 June, Vol. 1 pp. V1-94-V1-99.

Sun, G., Anand, V., Yu, H.F., Liao, D. and Li, L. (2012), "Optimal provisioning for elastic service oriented virtual network request in cloud computing", *IEEE Global Communications Conference (GLOBECOM), 2012*, *Anaheim, CA*, 3-7 December, pp. 2541-2546.

Tanenbaum, A.S. and van Steen, M. (2007), *Distributed Systems: Principles and Paradigms*, Pearson Prentice Hall, Upper Saddle River, NJ, ISBN0-13-239227-5.

Tang, R., Yue, Y., Ding, X. and Qiu, Y. (2014), "Credibility-based cloud media resource allocation algorithm", *Journal of Network and Computer Applications*, Vol. 46, pp. 315-321, available at: http://dx.doi.org/10.1016/j.jnca.2014.07.018i.

Xiao, Z., Song, W. and Chen, Q. (2013), "Dynamic resource allocation using virtual machines for cloud computing environment", *IEEE Transactions on Parallel And Distributed Systems*, Vol. 24 No. 6.

Ye, K., Huang, D., Jiang, X., Chen, H. and Wu, S. (2010), "Virtual machine based energy efficient data center architecture for cloud computing: a performance perspective", IEEE/ACM International Conference on Green Computing and Communications and IEEE/ACM International Conference on Cyber, Physical and Social Computing, Hangzhou, 18-20 December, pp. 171-178.

Yi-wei, F. (2015), "Limitation on stability and performance of control system over a communication channel", *International Journal of Engineering Sciences*, Vol. 4 No. 3, pp. 19-27.

Zhang, B., Zhao, Y. and Wang, R. (2013), "A resource allocation algorithm based on media task QoS in cloud computing", *Proceedings of the 4th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, *Beijing*, pp. 841-844.

Zhang, M. and Zhu, Y. (2013), "An enhanced greedy resource allocation algorithm for localized SC-FDMA systems", *IEEE Communications Letters*, Vol. 17 No. 7, pp. 1479-1482.

Corresponding author
Masoud Nosrati can be contacted at: minibigs_m@yahoo.co.uk