Group technology-based model and cuckoo optimization algorithm for resource allocation in cloud computing

S. Shahdi-Pashaki*, Ehsan Teymourian**, Vahid Kayvanfar***, GH.M. Komaki****, A. Sajadi****

*Department of Industrial Engineering, Faculty of Engineering, Shomal University, Amol, Iran (email miladshahdi@gmail.com)
**Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran
***Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran
****Department of Electrical Engineering and Computer Science, Case Western Reserve University, USA

Abstract: The control of operational costs is one of the main goals of resource management problem in cloud computing (CC). This paper presents a new mathematical model based on group technology (GT) to map the virtual machines (VMs) to workflows in order to control some costs (e.g. transfer costs, penalty costs and server cost) when the VMs are running. GT is a well-known manufacturing technique in industrial environments which can control some measures (e.g. part movements, resource utilization). In large size problems a cuckoo optimization algorithm (COA) is proposed. To test the effectiveness of our approaches, we first generate a set of problems randomly and then compare the model and COA with a well-known algorithm in literature called Round robin (RR). Analyzing the computational results proves that our approaches give better performance than RR.

© 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Cloud computing; Group technology; Virtual machine; Cuckoo optimization algorithm.

1. INTRODUCTION

Nowadays the availability of computing resources at anytime and in anyplace has become a major challenge for individuals, companies, and organizations. In this context, IT professionals have tried to offer new solutions to address this challenge. Cloud computing (CC) is the latest solution to meet this challenge. CC is result of the convergence of several technologies such as hardware, internet technologies, distributed computing and systems management. The main advantage of cloud computing is providing computing resources similar to traditional public utility services (i.e., water, electricity, gas, and telephony). Users (individuals and organizations) can pay their bills based on pay as you go model (Buyya et al., 2011; Selvarani and Sadhasivam, 2010). Therefore, they can reduce some additional costs such as the purchase of IT infrastructure as well as maintenance. The CC service providers offer three main layers of services to users: (1) infrastructure as a service, (2) platform as a service, and (3) software as a service (Buyya et al., 2011). Fig. 1 presents the main layers of services in CC.

Cost reduction is one of the main goals of distributed computing systems. Users are seeking to reduce costs (i.e., hardware and software) by transferring their operation to the cloud environment. On the other hand, service providers (SPs) want to increase the profits obtained from resources through increasing the income of hosting more number of users and also reducing the costs.

Appropriate resource management is a factor with large impact on reducing the computational cost and increasing the efficiency of CC. Users submit a wide range of tasks to CC systems that some of them are simple (e.g., a comment on the Facebook) and some others complex like banking operations (e.g., electronic funds transfer (EFT) and automated clearing house (ACH)). Fig. 2 indicates an electronic funds transfer (EFT) via payment card as a workflow. The EFT is an electronic transfer of money from one account to another between two or more financial institutions.

A SP is obliged to allocate computing resources to users regarding the service level agreement (SLA) to minimize some operational costs. The following costs may exist in CC:

**Transfer cost:** As mentioned earlier takes can be workflows in CC. Thus, it is possible that a server fails to execute all subtasks of a task due to the limitations of hardware and software. Therefore, the SP may have to be transfer tasks between servers to complete (Figure 3). Communicating and data exchanging between servers are prerequisites for a
Transfer cost. To the system which called "Transfer cost".

Fig. 2. Workflow of electronic funds transfers

Penalty cost: In CC, each task has multiple qualities of service (QoS) that are included in service level agreement (SLA). A penalty cost occurs when the provisions contained in the SLA are not met. Many tasks, particularly in e-banking should be processed into high-volume batches. For example, in ACH networks (i.e., processing of large volumes of transactions in batches at specified times) the penalty cost occurs when the transactions not processed in their due time.

Server costs: A cloud computing environment consists of multiple servers that service providers employ to execute the tasks. Using each server imposes several costs to the system (e.g., hardware, maintenance and energy consumption) that are variable from one server to another. A service provider should utilize existing servers so that the mentioned costs are minimized.

All the above costs are unavoidable, but they can be controlled by with proper management of available resources. One of the well-known manufacturing techniques in industrial environments is named group technology (GT), which has a high potential to reduce production costs. GT can prevent unnecessary movements (transfers) and increases the production efficiency in manufacturing systems. This technique was first proposed by Mitrofanov (1966) and then developed by Burbidge (1971). The main objective of GT is grouping the similar parts and machine to increase the flexibility. Cellular manufacturing system (CMS) is an application of GT for implantation of the modern manufacturing systems (e.g., flexible manufacturing system and just in time). Cell formation (CF) is a component of CMS that organizes independent cells by specifying the machine and part groups. The structural similarities of cloud systems and CMS are the main reason to suggest GT for resource management in CC. The servers, virtual machines, and tasks in CC have similar functions to cells, machine, and component, respectively. This study investigates a resource allocation problem with the aim of reducing the costs of service providers and users. A mathematical model based on GT is presented to minimize three costs including server, penalty, and transfer costs. Also, the proposed model makes a decision to use a proper server and virtual machines in response to user's requests. The paper is further structured as follows. Related work of resource management in CC and group technology is represented in Section 2. Problem modeling consists of its description is demonstrated in Section 3. Section 4 gives the structure of proposed COA.

2. LITERATURE REVIEW

Studies addressing the problem of resource management are quite extensive and rapidly growing. Most of the studies focus on heuristic and meta-heuristic approaches. Ergu et al. (2013) applied an analytic hierarchy process (AHP) to determine the priority of tasks to allocate resources to them. Li (2009) developed a queuing model and design a cost function to meet the requirements of users and maximize the profit of SP. A particle swarm optimization (PSO) developed by Pandey et al. (2010) to minimize the computational and data transferring cost. Their heuristic reduced the costs comparing to the best resource selection (BRS) algorithm. A novel scheduling algorithm that provides accordance between time and cost is presented by Liu et al. (2010). This algorithm controls one factor (i.e., cost) at a fixed level of the other factor (i.e., time). A successive approach using a game theory was presented by Wei et al. (2010). It solved a resource allocation problem in cloud computing through two consecutive steps. A resource allocation problem exclusive of resource sharing is investigated in the first step; after that, a binary integer programming is proposed to update solutions considering resource sharing. A genetic algorithm presented by Hu et al. (2010) to minimize the number of VM migrations. Xu et al. (2011) applied the Berger model to introduce an algorithm that its performance is fair in allocation of resources to tasks. The Berger model is a social psychological theory provided by Joseph Berger to appraise the impact of actor's behavior. The algorithm first classifies tasks based on QoS preferences and then assigns to resources according to obtained prioritization. Tayal (2011) suggested a fuzzy-GA to schedule the resources and balance the load on them. Several heuristics for allocating and scheduling the computing resources regarding QoS requirements are developed by Beloglazov, Abawajy and Buyya (2012). Also, Selvaran and Sadhasivam (2010) suggested a scheduling algorithm in addition to the grouping of tasks to enhance the computation/communication ratio. According to the studies reviewed it is found that some of them are seeking a fair allocation of resources, and others Furthermore, lots of studies have been done on GT including mathematical modeling, heuristics and meta-heuristic techniques. Two mathematical models suggested by Chan et al. (2008) including multi-objective and single objective model to form machine/part groups and control the intra and inter-cell movements. Nair and Narendran et al (2010) presented a machine and part families technique relying on production sequence data. Zhao and Wu (2000) tried to optimize multiple goals including movements cost, cell load variation and exceptional elements (EEs) in a CFP with genetic algorithm. Mahdavi and Mahadevan (2008) addressed a CFP and cell layout problem (CLP) simultaneously. They suggested an integrated mathematical model to control forward against backtracking movements. Mahdavi et al. (2013) perused a biphasic CFP in view of the multifunctional machines. Their goal was to develop a mathematical model to minimize the dissimilarity of machines groups in cells. Tavakkoli et al. (2007) compared the performance of different algorithms (i.e. GA, SA, and tabu search (TS)) for a dynamic CFP and found that the probability to achieve the optimal solution is increased when the genetic operators are steadily improved. According to the studies reviewed the main
misions of GT (i.e. Consolidate Parts to machines and machines to servers, controlling the inter/intra-cell movement costs and enhancing the flexibility of system) and the objectives of resource management in CC (i.e. Consolidate tasks to VMs and VMs to servers, controlling the transfer costs and enhancing the flexibility of system) are one and the same. These are the main reasons for using GT for resource management in CC.

These are the main reasons for using GT for resource and enhancing the flexibility of tasks to VMs and VMs to servers, controlling the transfer costs objectives of resource management in CC (i.e. Consolidate such as workflows that should be further investigated. Also assumed independent and simple while more complex tasks In most studies in the field of cloud computing tasks are In this study, a CC environment is assumed with mathematical model based group technology to manage VMs In order to control these computational cost. 

### 3. PROBLEM DESCRIPTION

In most studies in the field of cloud computing tasks are assumed independent and simple while more complex tasks such as workflows that should be further investigated. Also, there are some unavoidable costs (e.g., penalty and transfer costs) that have been less studied. So, due to the lack of exact solution for resource management problem (most of solution costs) that have been less studied. So, due to the lack of exact mathematical model based group technology to manage VMs in order to control these computational cost.

**Fig. 3. Simple case of resource allocation in cloud computing**

In most studies in the field of cloud computing tasks are assumed independent and simple while more complex tasks such as workflows that should be further investigated. Also, there are some unavoidable costs (e.g., penalty and transfer costs) that have been less studied. So, due to the lack of exact solution for resource management problem (most of solution costs) that have been less studied. So, due to the lack of exact mathematical model based group technology to manage VMs in order to control these computational cost.

### 3. PROBLEM DESCRIPTION

In this study, a CC environment is assumed with $K$ servers and $P$ types of VMs for executing $J$ tasks that must be processed in batch in a specified time period. The capacity of VMs and servers are known and constant during the time and introduced the by some parameters including: central processing unit (CPU), storage capacity and memory. The VMs and tasks must be distributed among servers and VMs according their capacity so that the allocation costs including: cost of using of servers, tasks transfer costs between servers and penalty costs be minimized. Fig. 3 presents the schematic view of resource allocation problem in CC. Thus, the notations used in problem formulation are defined as follows.

**Nomenclatures**

<table>
<thead>
<tr>
<th>Indices</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>Task ($j = 1, \ldots, J$)</td>
</tr>
<tr>
<td>$i$</td>
<td>Subtask ($i = 1, \ldots, I$)</td>
</tr>
<tr>
<td>$p$</td>
<td>Virtual machine type ($p = 1, \ldots, P$)</td>
</tr>
<tr>
<td>$k$</td>
<td>Server ($k = 1, \ldots, K$)</td>
</tr>
<tr>
<td>$M$</td>
<td>Large positive number</td>
</tr>
</tbody>
</table>

**Variables:**

- $\tau_{ijp}$: processing time for subtask $i$ of task $j$ on machine $p$
- $a_{ijp}$: 1 if subtask $i$ of task $j$ will require machine type $p$; 0, otherwise
- $L_p$: Amount of QoS that machine type $p$ can provide
- $C_k$: Amount of QoS that server $k$ can provide
- $DS_j$: Amount of demand of task $j$ stated in the SLA
- $A_p$: The unit cost of VM creation
- $B_i$: The unit cost of task transfer
- $\alpha, \beta, \gamma, \delta$: The user defined preferences

**Decision variables:**

- $X_{ijpk}$: 1 if subtask $i$ of task $j$ is done on machine type $p$ in server $k$; 0, otherwise
- $N_{pk}$: Number of machine type $p$ assigned to server $k$;
- $Y_{ijk}$: 1 if subtask $i$ of task $j$ assigned to server $k$; 0, otherwise
- $H_k$: The number of tasks assigned to server $k$
- $V_i$: 1 if server $k$ be turned on, $H_k > 0$; 0, otherwise
- $Y_j$: 1 if the demand of task $j$ is not met; 0, otherwise
- $D_j$: Amount of the demand of task $j$ that are met by the service provider
- $S_j$: 1 if there is a shortage for task $j$; otherwise 0

**Mathematical formulation.** Using above notations, the proposed model is now written by:

$$
F = \alpha \sum_{i} \sum_{j} \sum_{p} \sum_{k} X_{ijpk} \left[ Y_{ijk} \left(1-Y_{ijk}\right) \right] + \beta \sum_{k} P_i V_i + \gamma \sum_{p} N_{pk} A_p + \delta \sum_{j} \left( DS_j - D_j \right) Y_j P_j
$$

s.t.

- $X_{ijpk} = a_{ijp} Y_{ijk}$ $\forall i, j, p, k$ (2)
- $X_{ijpk} \leq N_{pk}$ $\forall i, j, p, k$ (3)
- $\sum_{p} \sum_{k} X_{ijpk} = 1$ $\forall i, j$ (4)
- $\sum_{i} \sum_{j} D_j \cdot a_{ijp} \cdot X_{ijpk} \leq L_p \cdot N_{pk}$ $\forall p, k$ (5)
- $\sum_{p} L_p \cdot N_{pk} \leq C_k V_k$ $\forall k$ (6)
- $M \cdot (S_j - 1) < DS_j - D_j$ $\forall j$ (7)
- $M \cdot S_j \geq DS_j - D_j$ $\forall j$ (8)
- $\sum_{i} \sum_{j} Y_{ijk} = H_k$ $\forall k$ (9)
The objective function (1) consists of four terms. The first term presents transfer cost which is occurs when sequential subtasks are executed on different servers \((Y_{ik} = 1)\) and \((Y_{i+1,k} = 0)\). The second to fourth term shows the cost of servers, the VM creation cost, and total penalty cost respectively. The phrase \((DS_i - D_j) \geq 0\) shows unmet demands. Constraint (2) shows the relation between two variables \((X_{ijpk} \text{ and } Y_{ijpk})\) and state that task \(i\) of task \(j\) is performed on VM \(p\) in server \(k\) \((X_{ijpk} = 1)\) if \((Y_{ik} = 1)\) and \((a_{ijp} = 1)\) simultaneously. Constraints (3) and (4) ensure that each subtask is assigned just to one VM in one server that required VMs are existent in it. Constraints (5) and (6) guarantees that virtual machines and servers’ capacity are not exceeded. Constraints (7) and (8) represents shortage \((S_i = 1)\) or lack of it \((S_i = 0)\) together. Constraints (9) to (11) show the use \((V_t = 1)\) or non-use \((V_t = 0)\) of servers and state that a server is used only when at least one task be assigned to it \((H_t > 0)\). Ultimately, binary and non-negative integer decision variables are presented in Constraints (12) and (13).

4. CUCKOO OPTIMIZATION ALGORITHM

To evaluate the effectiveness of mathematical model test problems are solved by Lingo 9. After running the model several times we found that the model can solve small problems up to \(7 \times 5 \times 4\) (tasks \(\times\) VM type \(\times\) servers) sizes. So, it is necessary to use approximate solution methods for larger problems. For this purpose, we suggest a cuckoo optimization algorithm (COA). COA is one of the evolutionary techniques proposed by Rajabioun (2011). This algorithm imitates Cuckoo’s habits and lifestyle (i.e., laying their eggs in the nests of other birds and migrating to more suitable habitats for breeding their eggs) to find near-optimal solution. The COA starts with an initial population where for each individual lays its egg in host nest. A number of eggs are identified by the host and the residuals migrate to the best habitat in the area after becoming to mature cuckoos.

4.1. Proposed COA

In this section, the procedure of the designed COA for large size problems is described as following.

**Habitat structure of the proposed COA.** The structure of each habitat (i.e., solution) which composed of three genes (i.e., \(X, Y, \text{and } N\)) is presented in Fig. 4. Matrix \(X\) and \(Y\) are related to assign subtasks to VMs and subtasks to servers, where their elements vary from 1 to \(p\) and 1 to \(k\), respectively. Matrix \(N\) is related to the number of VMs in servers and their elements are positive integer. Worth noting that matrix \(X\) is constant and obtained by \(a_{ijp}\).

**Generating the initial cuckoo habitat.** For generating the initial habitats, the first matrix \(N\) is randomly initialized so that the server capacity does not exceed. Then, matrix \(Y\) is initialized randomly with regard to the possibility of task execution.

**Profit function.** The profit value is criterion for the suitability rate of a solution for raising cuckoo eggs. In this paper, we use the objective function of model as a profit function (PF) according to Eq. (14).

\[
PF = -\alpha \sum_{X} \sum_{Y} \sum_{K} \sum_{I} \sum_{P} \sum_{J} D_{ijpk}B_{ij} \left[ Y_{ij}(1 - Y_{i+1,j}) \right] + \beta \sum_{P} \sum_{J} P_{ij} + \gamma \sum_{K} \sum_{I} \sum_{P} N_{ijpk} A_{ij} + \delta \sum_{S} (DS_{i} - D_{i}) \cdot Y_{ij}
\]

**Immigration operator.** As mentioned previously, mature cuckoos migrate to the best habitat with the maximum similarity between their own eggs and eggs of host birds. But during migration they only fly a part of distance (% \(\lambda\) of all distance) with the amount of deviation (\(\varepsilon\)) as shown in Fig. 5. This means that the cuckoos will be indwelled in habitats that are most similar to the target after the migration. Therefore, due to discrete nature of problem and also matrix nature of habitats, we suggest an immigration procedure according to the problem space (Fig. 6). As is clear, First \(K\%\) (where \(K\) is a random round number) of columns (or rows) of target habitat, i.e., matrix \([N]\) and \([Y]\) are selected and copied into columns (rows) of the current habitat identically. After applying the immigration operator on current habitat, some parts of the new habitat (the second column of matrices \(Y\) and \(N\)) will be quite similar to the target habitat. So, the target habitat can be reached by applying the immigration operator frequent basis.

**Repairing.** An infeasible habitat is arising when the number of VMs in servers and the subtasks on VMs exceed their capacity. In this situation, a procedure modifies the infeasible solutions. The repair procedure is applied on some parts \((X\text{ and } Y\text{ or both})\) of an infeasible habitat that is changed by operators (Fig. 7).

**Proposed COA.** The parameter of COA and their values which are obtained experimentally as well as the overall
structure of the proposed COA are presented as a pseudo-code in Table 1 and Fig. 9 respectively. The algorithm starts with generating the solutions (i.e., cuckoos habitat) randomly and several new solutions are created by laying eggs from each solution. The best solution is selected among the available solutions, and then other solutions become similar to the best solution (i.e., immigration). This process will continue to get the maximum number of generations ($G$).

<table>
<thead>
<tr>
<th>Current habitat</th>
<th>New habitat</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$ $Y$ $Z$ $V$</td>
<td>$X$ $Y$ $Z$ $V$</td>
</tr>
<tr>
<td>1 1 2 3 1 1 1 1 4 2 3</td>
<td>1 1 2 3 1 1 1 1 4 2 3</td>
</tr>
<tr>
<td>3 2 1 2 1 1 1 1 4 2 3</td>
<td>3 2 1 2 1 1 1 1 4 2 3</td>
</tr>
</tbody>
</table>

![Fig. 5. Egg laying](image)

![Fig. 6. Immigration operator](image)

Reparing
1 Begin
2 If the machine section have been changed, then
3 While equipped capacity is less than total capacity of servers
4 Do Rearrange the machines
5 End While
6 Elseif the servers have been changed, then
7 While equipped capacity of machine is less than total capacity in all servers
8 Do Rearrange components based on their possibility of doing then on the machines
9 End While
10 End

![Fig. 7. Pseudo-code of repairing](image)

Table 1. the parameter of COA

<table>
<thead>
<tr>
<th>Parameters of COA</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Initial Cuckoos</td>
<td>5</td>
</tr>
<tr>
<td>Minimum Number of Eggs Laid by Each Cuckoo ($E_{min}$)</td>
<td>2</td>
</tr>
<tr>
<td>Maximum Number of Eggs Laid by Each Cuckoo ($E_{max}$)</td>
<td>4</td>
</tr>
<tr>
<td>Maximum Number of Iterations</td>
<td>1000</td>
</tr>
<tr>
<td>Number of cuckoos groups</td>
<td>4</td>
</tr>
<tr>
<td>Maximum Number of Living Cuckoos ($C_{max}$)</td>
<td>10</td>
</tr>
</tbody>
</table>

4.2. Computational Results
To prove the performance of the proposed model and COA, we tested them on some random problems randomly generated with $i \times k$ (tasks $\times$ servers) demotions. The model and COA were developed using lingo 9 and MATLAB 2010 and on a PC Pentium (R) Dual-core CPU with 2.50 GHz and 2GB RAM. Details of VMs presented in Table 2 collected from the Amazon EC2.

We also measure the performance of our method compared with other methods available in the literature, for this purpose, we selected the round-robin (RR) is one of the algorithms employed by network schedulers in computing systems.

![Fig. 8. Pseudo-code of the proposed COA](image)

![Fig. 9. Comparison of Model, COA and RR for small problems](image)

![Fig. 10. Comparison of Model, COA and RR for large problems](image)

Table 2. Information of virtual machines

<table>
<thead>
<tr>
<th>Capacity / VM</th>
<th>VM1</th>
<th>VM2</th>
<th>VM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>15 GB</td>
<td>17.1 GB</td>
<td>7 GB</td>
</tr>
<tr>
<td>CPU</td>
<td>8 EC2 units</td>
<td>6.5 EC2 units</td>
<td>20 EC2 units</td>
</tr>
<tr>
<td>Storage</td>
<td>1690 GB</td>
<td>420 GB</td>
<td>1690 GB</td>
</tr>
<tr>
<td>Cost</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The average normalized objective values (i.e., transfer cost, server cost, penalty cost and VM cost) obtained by our approaches (i.e., the model which solved by Lingo software and proposed COA which solved by MATLAB 2010...
programing language) and RR for 10 small sizes (e.g., 7 and 5) and 10 large sizes (e.g., 25 and 15) are graphically presented in Fig. 9 and Fig. 10. Our proposed approaches have better performance compared to RR both in small and large problems.

5. CONCLUSION
Reducing the computational costs is one of the main purposes of cloud computing systems. Existence of an appropriate resource management strategy which can greatly select the best virtual machine (VMs) for executing the tasks can greatly meet this goal. So, in this paper a new mathematical model based on group technology (GT) was proposed to minimize some computational costs (e.g., execution, penalty, and VM creating and data transfer costs). Also, a cuckoo optimization algorithm was proposed to solve large scale problems that have no optimum solution. As future direction of research, one can apply Intelligent water drops (e.g., Booyavi et al., 2014) or Harmony search (e.g., Komaki et al., 2014) and compare them to the proposed COA in this study.

REFERENCES
Burbidge, J.L. (1971). Production flow analysis, Production Engineer, 50, 139-152.