Multi-objective resources allocation approaches for workflow applications in Cloud environments

Kahina Bessai, Samir Youcef, Ammar Oulamara, Claude Godart, Selmin Nurcan

Multi-objective resources allocation approaches for workflow applications in Cloud environments

Kahina Bessai[1], Samir Youcef[2], Ammar Oulamara[2], Claude Godart[2] and Selmin Nurcan[1]

[1]University of Paris1-Panthéon- Sorbonne, Centre de Recherche en Informatique
90, rue de Tolbiac 75634 Paris, France
{kahina.bessai,selmin.nurcan}@malix.univ-paris1.fr
[2]University of Lorraine, LORIA-INRIA-UMR 7503
BP 239, Vandoeuvre-les-Nancy, France
{youcef,oulamara,godart}@loria.fr

Abstract. Resources allocation and scheduling has been recognised as an important topic for business process execution. However, despite the proven benefits of using Cloud to run business process, users lack guidance for choosing between multiple offering while taking into account several objectives which are often conflicting. Moreover, when running business processes it is difficult to automate all tasks. In this paper, we propose three complementary approaches for Cloud computing platform taking into account these specifications.

1 Introduction

The Cloud computing has quickly changed the way that compute resources can be used and allow users to access compute on the fly according to the application's need. For example, to run any desired software Amazon's EC2 provide a Web service through which users can boot an Amazon Machine Image. However, despite the proven benefits of using Cloud to execute business processes, users lack guidance for choosing between different offering while taking into account several objectives often conflicting.

Moreover, most existing workflow matching and scheduling algorithms consider only an environment in which the number of resources is assumed to be bounded. However, in distributed systems such as Cloud computing this assumption is in opposition to the usefulness of such systems. Indeed, the "illusion of infinite resources" is the most important feature of Clouds [2][3], which means that users can request, and are likely to obtain, sufficient resources for their need at any time. Additionally to this characteristic, a Cloud computing environment can provide several advantages that are distinct from other computing environments [3]. Moreover, unlike scientific workflows, where generally their processing are fully automated, when executing workflow processes it is difficult to automate all theirs tasks. Indeed, certain tasks require validations that cannot be automated because they are subject to human intervention.

To overcome the limitations of existing approaches for workflow process optimization, we propose an extension of our recent study [1].

2 Problem formulation

Definition 1 (Business process). *A business process application is represented as a directed acyclic graph (DAG) denoted $G = (T, E)$, where:*

1. *$T = \{t_1, ..., t_n\}$ is a finite set of tasks.*
2. *E represents the set of directed edges. An edge (t_i, t_j) of graph G corresponds to the data dependencies between these tasks (the data generated by t_i is consumed by t_j).*
3. *Task t_i is called the immediate parent of t_j which is the immediate child task of t_i.*

Let $Data$ be a $n \times n$ matrix of communication data, where $data[i, j]$ is the amount of data required to be transmitted from task t_i to task t_j.

Definition 2 (Resource graph). *The resources are represented as a directed graph denoted RG. Formally, a resources graph is represented by $RG = (R, V)$, where:*

1. *$R = \{VM_1, ..., VM_m, HR_1, ..., HR_{m'}\}$ is a finite set of virtual machines types and human resources.*
2. *V represents the set of directed edges. Each edge is denoted (VM_i, VM_j) corresponding to the link between these virtual machines.*

Let B be a $m \times m$ matrix, in which $B[i, j]$ is the bandwidth between virtual machine types VM_i and VM_j, where $B[i, i] \longrightarrow \infty$ means that there is no transfer data.

Let $r(t_j)$ denotes the resource (virtual machine or human resource) that executes task t_j. The transfer time $TT(r(t_i), r(t_j))$, which is for transferring data from task t_i (executed by $r(t_i)$) to task t_j (executed by $r(t_j)$) is defined by:

$$TT(r(t_i), r(t_j)) = \frac{data[i, j]}{B[r(t_i), r(t_j)]}$$

Let ET be a $n \times m$ execution time matrix in which $ET(t_i, r_j)$ gives the execution time estimation to complete task t_i by resource r_j.

Let UEC be a $(m + m')-$dimensional unit execution cost vector, where $UEC(r_j)$ represents the cost per time unit incurred by using the resource r_j. Let EC be a $n \times m + m'$ execution cost matrix in which $EC(t_i, r_j)$ gives the execution cost to complete task t_i by resource r_j defined by:

$$EC(t_i, r_j) = ET(t_i, r_j) \times UEC(r_j)$$

The data transfer cost $TC(r(t_i), r(t_j))$, which is the cost incurred due to the transfer of data from task t_i (executed by $r(t_i)$) to task t_j (executed by $r(t_j)$), is defined by: $TC(r(t_i), r(t_j)) = data[i, j] \times (C_{out}(r(t_i)) + C_{in}(r(t_j)))$

where $C_{out}(r(t_i))$ and $C_{in}(r(t_j))$ represent respectively the cost of transferring data from $r(t_i)$ and the cost of receiving data on $r(t_j)$.

3 The objective functions

a. Time objective function Let EST (earliest start time) and EFT (earliest finish time) attributes that characterize the set of resources (virtual machine types and human resources). These attributes are derived from a given partial matching and scheduling (i.e. a task t_i is assigned to virtual machine $VM(t_i)$ or human resource $HM(t_i)$). The partial schedule refers to the fact that for each task the earliest start time and the earliest finish time values are obtained using only the tasks that must be performed before it. $EST(t_i, r_j)$ and $EFT(t_i, r_j)$ are the earliest execution start time and the earliest execution finish time of task t_i on resource r_j, respectively. For the input task t_{input}, the earliest execution start time and the earliest execution finish time are given by Equation 1 and Equation 2, respectively:

$$EST(t_{input}, r_j) = 0 \tag{1}$$

$$EFT(t_{input}, r_j) = ET(t_i, r_j) \tag{2}$$

For the other tasks in the graph, the EST and the EFT values are computed recursively, starting from the initial task, as shown in Equation 3 and Equation 4. In order to compute the EFT of a task t_j, all immediate predecessor tasks of t_j must have been assigned and scheduled with the consideration of the transfer time.

$$EFT(t_i, r_j) = EST(t_i, r_j) + ET(t_i, r_j) \tag{3}$$

$$EST(t_i, r_j) = \max\left\{avail[j], \max_{t_p \in pred(t_j)} [AFT(t_p) + TT(r(t_p), r(t_i))]\right\} \tag{4}$$

where $pred(t_i)$ is the set of immediate predecessors of task t_i. $avail[j]$ is the earliest time at which resource r_j is ready for task execution. As the number of virtual machines is assumed to be infinite, then $avail[j] = 0$ for all the used virtual machines. In other words, if task t_i is performed by resource r_j which is a virtual machine then $EST(t_i, r_j)$ is computed as follows:

$$EST(t_i, r_j) = \max_{t_p \in pred(t_j)} \{AFT(t_p) + TT(r(t_p), r(t_i))\} \tag{5}$$

The $avail[j]$ is the time that resource r_j completed the execution of the task t_i and it is ready to execute another task. The inner max block in the EST equation returns the ready time, i.e. the time when all required data by t_i has arrived at resource r_j.

After a task t_i is scheduled on a resource r_j, the earliest start time and the earliest finish time of t_i on resource r_j is equal to the actual start time, denoted $AST(t_i)$, and the actual finish time, denoted $AFT(t_i)$, of task t_i, respectively.

After all tasks in a graph are scheduled, the schedule length (i.e., the overall completion time) will be the actual finish time of the exit task (i.e. $AFT(t_{exit})$). The schedule length, also called *makespan*, is defined as:

$$makespan = AFT(t_{exit}) \tag{6}$$

Predictive heuristic for human resource availabilities The actual finish time as computed previously does not take into account the fact that human resources can perform other tasks that do not belong to the same process. More precisely, the work list of a given human resource may contain work items of different processes. The realism of this assumption can be disputed as a human resources are "shared" by more than one process. Thus, it is might be desirable to design a procedure in order to predict the human resources availabilities. Concretely, we propose to estimate this availability values taking into account the previous observation. Let $availability[j]_{k+1}$ is the estimation of resource j availability and t_i the next task that can be performed by this resource. Instead of simply adapting to the computed availability using equation 4, one can try to forecast and estimate what $availability[j]_{k+1}$will be using the historical data.

b. Cost objective function The cost function is a structure independent criterion defined as the sum of the costs of executing all workflow tasks, given by:

$$cost = \sum_{j=1}^{n} \left\{ EC(r(t_j)) + \sum_{p \in pred(t_j)} TC(r(t_j), r(t_p)) \right\} \tag{7}$$

Thus, the *cost objective function* is to determinate the assignment of tasks of a given workflow application such that its overall execution cost is minimized.

To take into account these objectives simultaneously, we propose to use the proposed approaches in our study [1] by replacing the execution time and the execution cost by respectively Equations 4 and 7.

4 Conclusion

In this paper, we have proposed a model for business processes execution in Cloud computing environments. The proposed approaches extend our previous algorithms. More precisely, three complementary approaches are designed to deal with the problem of matching and scheduling business process tasks in Cloud context taking into account two objectives (execution time and cost incurred using a set of resources). We plan to extend the proposed work to take into account others criteria like carbon emission and energy cost.

References

1. K. Bessai, S. Youcef, A. Oulamara, C. Godart and S. Nurcan *Bi-criteria workflow tasks allocation and scheduling in Cloud computing environments*, In Cloud Computing 2012, pp. 638-645..
2. *Workflow Management Coalition: Terminology* & *Glossary,* WfMC-TC-1011, 1999.
3. R. Buyya, S. Pandey and C. Vecchiola, *Cloudbus toolkit for market-oriented cloud computing*, In CloudCom'09: Proceedings of the 1st International Conference on Cloud Computing, volume 5931 of LNCS, pp. 24-44, Springer, Germany, December 2009.