Available online at www.sciencedirect.com

ScienceDirect

IERI Procedia 10 (2014) 216 – 223

Procedia IERI

www.elsevier.com/locate/procedia

2014 International Conference on Future Information Engineering

Towards Better Workflow Execution Time Estimation

Artem M. Chirkin[a,b], Sergey V. Kovalchuk[a,]*

[a]*ITMO University, Birzhevaya line, 4, 199034, Saint-Petersburg, Russian Federation*
[b]*University of Amsterdam, Spui 21, 1012 WX,bAmsterdam, Netherlands*

Abstract

Authors highlight the importance of estimating workflow execution time in the scheduling problem and in pricing models in Clouds. Claim that better approaches need to be developed to obtain precise estimates of execution time. Describe straightforward regression for stochastic execution time estimation. Propose a combination of Chebyshev-like distribution-free inequalities and distribution-based approaches to calculate better confidence bounds on runtime based on results of statistical tests. Finally, authors explain the need of developing better algorithm to compute workflow execution time based on stochastic estimates of tasks' execution time. Explain the need of developing better workflow runtime estimation algorithm. Finally, authors present an approach to estimating workflow execution time based on stochastic estimates of tasks' execution time.

Selection and peer review under responsibility of Information Engineering Research Institute

Keywords: runtime, execution time, workflow, scheduling

1. Introduction

Nowadays research and production computational problems require more computing resources. Therefore heterogeneous systems (HSs) based on Grid and Cloud technologies become more popular because they

* Corresponding author. Tel.: +7-962-687-0460; fax: +7-812-232-23-07.
E-mail address: kovalchuk@mail.ifmo.ru.

Selection and peer review under responsibility of Information Engineering Research Institute
doi:10.1016/j.ieri.2014.09.080

provide high flexibility in scaling resources, usage of large data amounts, possibility to run single service on different computers as a part of a complex engine [1].

This work addresses the problem of estimating execution time of complex tasks in HS. Complex tasks executed in HS usually consist of several atomic tasks (stages) and form a workflow. We refer to the workflow as a collection of tasks and informational dependencies between them; these dependencies impose some restrictions on execution order of the tasks. Number of papers are written to describe different implementations of workflow management systems (e.g. [2, 3]). The main question researchers address when developing management systems is how to schedule workflow in the most efficient way: minimizing execution time, cost, meet some deadline or other constraints [4, 5]. We say that HS contains number of packages (applications) which provide services; each task of the workflow is an execution of a single service. Additionally, HS contains number of nodes (computing units). Scheduling problem consists of mapping tasks to nodes optimally. Often used approach to represent workflow is a directed acyclic graph (DAG) [4, 6, 7]. It is well-known that complexity of a general scheduling problem in DAG is NP-Complete [8], thus many heuristics exist that provide near-optimal solution to the scheduling problem.

Significant part of the scheduling problem is estimation of workflow execution time. We suppose the runtime estimation problem is not so well-developed as the scheduling problem because several straightforward techniques exist that give acceptable for many scheduling cases performance. However, we claim that these techniques may fail in some cases, especially in deadline constrained setting. Moreover, when estimating workflow execution time, one should take into account dependence between executed tasks, heterogeneity of the tasks and computing resources, variety of workflow forms. The most important feature of HS is that some execution time components are stochastic in their nature (data transfer time, overheads, etc.). These facets are not considered in modern researches together, therefore the problem needs a further research.

Besides scheduling, runtime estimation is used by the clients of HS. Cloud computing providers create pricing models to give their clients billing information [9]; they have to calculate precisely execution time estimates to present expected price of workflow execution for a user. Given the estimated time bounds, one can use a model described in [10] to provide user with pre-billing information.

The aim of this work is to highlight common problems of estimating workflow execution time and propose a solution that takes into account complexity and stochastic nature of workflow processes.

2. Execution time in workflow scheduling

Workflow scheduling is well-known to be a complex problem; it requires to consider its different facets (e.g. resource model, workflow model, scheduling criteria, etc.) [4]. Therefore many authors describing infrastructure of their scheduling systems do not focus on the runtime estimation. We separate these two questions so that results obtained in a research dedicated to general workflow execution time problem can be used in complex scheduling systems which already exist or are being developed. In such setting an estimating system provide values of target function (execution time) to the scheduling system similar as if the scheduling system have calculated it internally. Additionally, we can use the estimating system to get remaining execution time at any stage of workflow execution to monitor the process or re-schedule remaining tasks. Some examples of scheduling systems that naturally can use any implementations of runtime estimation module are CLAVIRE (ITMO University, St.Petersburg, Russia) [2, 11], framework proposed by Ling Yuan at al.[3], and CLOWDRB by Somasundaram and Govindarajan [12].

Today a large variety of scheduling heuristics have been studied. Some of them do not require to estimate workflow runtime [13, 14], but this decreases flexibility and maximum possible effectiveness of a scheduler [15]. Most schedulers can be classified by the type of assumption regarding time. The first class of assumptions requires only information whether one task needs longer time to execute than other (i.e. ordered