Simulation Modelling Practice and Theory 79 (2017) 37–53

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

# Simulation of a workflow execution as a real Cloud by adding noise

Roland Mathá[a,*], Sasko Ristov[a,b], Radu Prodan[a,c]

[a] Distributed and Parallel Systems Group, Institute for Computer Science, University of Innsbruck, Technikerstr. 21a, A-6020 Innsbruck, Austria
[b] Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, 1000 Skopje, Macedonia
[c] Institute of Information Technology, University of Klagenfurt, Universitätsstr. 65-67, 9020 Klagenfurt, Austria

ARTICLE INFO

*Article history:*
Received 11 July 2016
Revised 2 August 2017
Accepted 20 September 2017

*Keywords:*
Inaccuracy
Makespan
Metric
Modelling
Precision
Simulator

ABSTRACT

Cloud computing provides a cheap and elastic platform for executing large scientific workflow applications, but it rises two challenges in prediction of makespan (total execution time): performance instability of Cloud instances and variant scheduling of dynamic schedulers. Estimating the makespan is necessary for IT managers in order to calculate the cost of execution, for which they can use Cloud simulators. However, the ideal simulated environment produces the same output for the same workflow schedule and input parameters and thus can not reproduce the Cloud variant behavior. In this paper, we define a model and a methodology to add a noise to the simulation in order to equalise its behavior with the Clouds' one. We propose several metrics to model a Cloud fluctuating behavior and then by injecting them within the simulator, it starts to behave as close as the real Cloud. Instead of using a normal distribution naively by using mean value and standard deviation of workflow tasks' runtime, we inject two noises in the tasks' runtime: noisiness of tasks within a workflow (defined as *average runtime deviation*) and noisiness provoked by the environment over the whole workflow (defined as *average environmental deviation*). In order to measure the quality of simulation by quantifying the relative difference between the simulated and measured values, we introduce the parameter *inaccuracy*. A series of experiments with different workflows and Cloud resources were conducted in order to evaluate our model and methodology. The results show that the inaccuracy of the makespan's mean value was reduced up to 59 times compared to naively using the normal distribution. Additionally, we analyse the impact of particular workflow and Cloud parameters, which shows that the Cloud performance instability is simulated more correctly for small instance type (inaccuracy of up to 11.5%), instead of medium (inaccuracy of up to 35%), regardless of the workflow. Since our approach requires collecting data by executing the workflow in the Cloud in order to learn its behavior, we conduct a comprehensive sensitivity analysis. We determine the minimum amount of data that needs to be collected or minimum number of test cases that needs to be repeated for each experiment in order to get less than 12% inaccuracy for our noising parameter. Additionally, in order to reduce the number of experiments and determine the dependency of our model against Cloud resource and workflow parameters, the conducted comprehensive sensitivity analysis shows that the correctness of our model is independent of workflow parallel section size. With our sensitivity anal-

* Corresponding author.
*E-mail addresses:* roland@dps.uibk.ac.at (R. Mathá), sashko@dps.uibk.ac.at (S. Ristov), radu@dps.uibk.ac.at (R. Prodan).

https://doi.org/10.1016/j.simpat.2017.09.003

ysis, we show that we can reduce the inaccuracy of the naive approach with only 40% of total number of executions per experiment in the learning phase. In our case, 20 executions per experiment instead of 50, and only half of all experiments, which means down to 20%, i.e. 120 test cases instead of 600.



## 1. Introduction

Cloud has evolved in a promising platform for many scientific applications and workflows' execution [1]. Still, there are many challenges that could impact on the decision whether to migrate the execution to the Cloud. Due to Cloud's performance instability [2], the cost and makespan (execution time) cannot be predicted correctly as they depend on the amount of leased resources and time period of leasing. This is emphasised even more for dynamic scheduling of workflows [3], since they consist of control and data dependent tasks. On the other hand, the same instance can behave totally unexpected in two different periods of time [4], and even a small fluctuation in the task runtime will change the scheduling, which can result in huge discrepancy of makespan caused by additional network traffic between instances and shifting the starting time of tasks. Utilising more resources can reduce the makespan, but in the same time it will increase the cost; this tradeoff between the cost and makespan is analysed by many authors in the literature [5], which also impacts the Cloud's performance instability.

Instead of executing a workflow schedule (usually in hours) to estimate its makespan and cost, one can use a Cloud simulator, which can simulate the execution within seconds [6]. However, most simulators are static and will always predict the same makespan for a specific workflow schedule and configured input parameters, such as workflow structure and computation and communication requirements, available resources and their capacity or scheduling policy. For the same schedule, each simulator will almost always return the constant result for the makespan. In real Cloud environment, the makespan and cost will be always different, due to its pay-as-you-go pricing model, dynamic starting of instances [7] and performance fluctuation [8] caused by heterogeneity of underlying hardware, multi-tenancy, migrations and relocations, or other issues in order to satisfy the constraints in service level agreements.

Simulators should reproduce the execution in a real Cloud; therefore, a methodology is necessary that will make the simulations accurate and precise [9]. In this paper, we propose a new methodology how a simulator can *learn* from the Cloud's behavior and then to *configure* the model for tasks' runtime instability by introducing a noise in order to reproduce the real Cloud dynamic environment for workflow executions. As a baseline, we consider the *accuracy* of the model, which is represented through the *trueness* (i.e. how close the simulation's mean values are with the true mean values of Cloud executions) and the *precision* (i.e. how close are the corresponding standard deviations) of the simulation, as defined in ISO-5725 standard [10]. This two-phase process consists of learning and configuration phases. In the former, an agent learns the Cloud's behavior by measuring the deviation of each pair of two executions (repetitions) of the same experiment in the real Cloud. Further on, in the latter phase, the agent configures the simulator by injecting the learned behavior as a noise within the simulation. By this process, the simulation becomes "instable" by generating task execution times as a random variable distributed with some probability function, in order to behave more closely as the real Cloud. Instead of using a normal distribution naively by using mean value and standard deviation of workflow tasks' runtime, we inject two noises in the tasks' runtime: noisiness of tasks within a workflow (defined as *average runtime deviation*) and noisiness provoked by the environment over the whole workflow (defined as *average environmental deviation*). Our noised model reduces the workflow makespan simulation inaccuracy up to 59 times compared with the simulation that naively uses the mean value and standard deviation of workflow tasks' runtime. Apart of the improved accuracy of the introduced approach (or model), the sensitivity analysis shows that we need only 15 to 20 "repetitions" of some experiments in the learning phase. That is, we can reuse the learned behavior for other workflow experiments.

The rest of the paper is organised as follows. Section 2 presents the related works in modeling the workflow execution instability and the features of Cloud simulators in this domain. In Section 3, we present a short background related to specific terms of workflows and Cloud. The theoretical analysis of Cloud instability and simulating its behavior is conducted in Section 4. Our model of adding a noise in simulation is described in Section 5, followed by process of noising in Section 6. In Section 7, we evaluate our model with a series of experiments, while in Section 8, we conduct a comprehensive sensitivity analysis in order to determine the minimal number of repetitions of each experiment in real Cloud for determining our noise metric. The strength and application domain of the model, along with additional insights are discussed in Section 9. Finally, we conclude the paper and present our future work in Section 10.

## 2. Related work

In this section we present the related work divided in two parts: the review of the cloud performance instability and the existing simulators and their features to simulate such instability.