Future Generation Computer Systems 57 (2016) 13–23

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Combining analytical modeling, realistic simulation and real experimentation for the optimization of Monte-Carlo applications on the European Grid Infrastructure

Sorina Camarasu-Pop *, Tristan Glatard, Hugues Benoit-Cattin

Université de Lyon, CREATIS, CNRS UMR5220, Inserm U1044, INSA-Lyon, Université Lyon 1, France

HIGHLIGHTS

- We combine production experiments, modeling and simulation of applications deployed on distributed resources.
- We exhaustively evaluate an analytical model of the application makespan.
- We study the influence and calibrate the parameters of the application workflow.
- Our simulator is thoroughly validated and uses real production traces as inputs.

ARTICLE INFO

Article history:
Received 24 July 2014
Received in revised form
4 September 2015
Accepted 4 November 2015
Available online 18 November 2015

Keywords:
Simulation
Distributed computing
Modeling
Optimization
EGI

ABSTRACT

Evaluating the performance of distributed systems through real experimentation is resource-consuming and by essence very difficult to reproduce. Conversely, analytical modeling and simulation facilitate investigation, but their level of realism needs to be evaluated to avoid misinterpretation. In this paper, we combine production experiments and realistic simulation for performance modeling and optimization of application workflows deployed on the European Grid Infrastructure (EGI), one of the largest distributed systems in the world. We use a validated simulator to (i) exhaustively evaluate an analytical model of the application makespan and (ii) study the influence and calibrate the parameters of the application workflow, in particular the checkpointing period. Experimental results show that the model fits the simulated makespan with a relative error of at most 15%, and that simulation allows us to validate analytical models in a more exhaustive manner than what is possible with production experiments. Results also show that, provided that the simulator is correctly validated and instantiated, simulation can be safely used for exhaustive parameter studies, allowing for a quick and fine tuning of sensitive application parameters.

1. Introduction

Scientific and commercial applications require nowadays an ever-increasing number of resources to deliver results for growing problem sizes in a reasonable time lapse. To meet the needs of these applications, various initiatives have built distributed computing systems offering several thousands of processing units. The European Grid Infrastructure (EGI[1]), BOINC platforms[2] or the Amazon Elastic Compute Cloud (EC2)[3] are only a few examples of large computing infrastructures widely adopted for compute-intensive software.

Monte-Carlo applications are a good example of compute-intensive methods. They are used in several scientific domains to produce realistic results from repeated sampling of events generated by pseudo-random algorithms. They are easily parallelizable and thus well suited for distributed computing. Nevertheless, efficient execution on large-scale computing grids is difficult to achieve, partly because of the heterogeneity and high error rates which characterize these infrastructures, where geographically distributed nodes co-operate to execute the multiple tasks of different applications. In order to cope with possible high failure rates, fault tolerance has to be provided at the application level. A commonly utilized technique for this is application-level checkpointing, which consists in saving a snapshot of the application's state, so that it can restart from that point in case of failure.

* Corresponding author. Tel.: +33 4 72 43 72 99.
E-mail address: sorina.pop@creatis.insa-lyon.fr (S. Camarasu-Pop).

1 http://www.egi.eu.

2 http://boinc.berkeley.edu.

3 http://aws.amazon.com/ec2.

http://dx.doi.org/10.1016/j.future.2015.11.002

For example, the GATE [1] Monte-Carlo application, our primary use-case for this study, has a mechanism to periodically pause, save results on disk and then continue its execution.

The optimization of distributed applications depends on the fine tuning of a certain number of parameters (e.g., the checkpointing period) that need to be set according to the application requirements and the infrastructure characteristics. On infrastructures such as EGI, optimizing such parameters is difficult because of the "production" nature of the system, which is uncontrolled and ever evolving, making it difficult to reproduce execution conditions and performance measures. In addition, such production systems are costly and using them with the unique goal of parameter optimization or extensive performance evaluation can be an unacceptable expense and a waste of resources. To tackle this problem, multiple approaches can be used, in particular mathematical modeling and simulation.

Mathematical modeling [2,3] helps to understand the behavior of distributed applications and tune parameters. Modeling remains a challenge for applications running in production conditions because most variables needed to evaluate the model are unknown before the completion of the experiment, for instance the background load of the infrastructure or the characteristics of the resources involved in the execution. The models used in production have to be able to cope with such a lack of information, focusing on variables that are measurable by the applications. They are thus of great help to understand the influence of parameters and the relation between various terms but are often impossible to use for performance prediction or parameter tuning in production.

Discrete-event simulation [4,5] has emerged as a method to facilitate experiments in the field of distributed computing. Simulation not only allows easier prototyping, parameter tuning and testing of new methods, but it also enables their thorough evaluation in situations not easily encountered in real-world scenarios. Simulators are often used for performance comparison. Different scenarios (such as different scheduling algorithms) are implemented and executed with the same simulator under the same assumptions and conditions for comparison purposes. Nevertheless, simulation results are rarely validated against production results. The same problem occurs when it comes to simulation toolkits, very few being extensively validated. Too often, the models used in simulation are either oversimplified to favor simulation speed, or too complex, therefore hardly usable.

In previous works, based on our extensive experience collected on the production Virtual Imaging Platform [6] (VIP[4]), we (i) proposed a mathematical model [7] allowing to understand the relationship between the different parameters and (ii) built and validated a simulator [8] based on the SimGrid toolkit [9] allowing to simulate the hardware platform, the core software services, the deployment, and the application used in VIP.

In this paper, we combine production experiments, modeling and simulation for performance evaluation and parameter tuning of application workflows deployed on EGI. The simulator described in [8] is now used to (i) exhaustively evaluate an extension of the model proposed in [7] and to (ii) study the influence of parameters such as the checkpointing period. The use of these three levels of abstraction (mathematical model, discrete-event simulation and real production experiments) and their mutual comparison allow us to study and optimize the system with a good level of confidence.

The rest of the paper is organized as follows. Section 2 presents related work on simulation, its usage and evaluation in the field of distributed computing, as well as on formal modeling and how analytical models can be used for performance evaluation and parameter tuning. Section 3 presents the real and simulated platforms, while Section 4 proposes an extension of the model presented in [7]. The experiments and results are presented in Section 5 and conclusions are drawn in Section 6.

2. Related work

2.1. Simulation

Simulation toolkits have become increasingly interesting instruments and there currently exists a variety of simulators, among which OptorSim [10], GridSim [11], PeerSim [12], CloudSim [13], MONARC [14], and SimGrid [15]. Recent reviews of these tools are available, e.g., in [16] and in [5]. The related work presented in the following covers literature on the types of applications and services that have been simulated recently, as well as on simulation usage and evaluation.

Different types of applications and grid services have been simulated to evaluate and improve their performance. The work in [17] simulates bag-of-task applications to study their scalability. In [18] simulation is used to study the checkpointing of sequential applications to improve fault-tolerance. In [4], the authors report on a simulation involving a DIRAC pilot-job scheduler, while the work in [19] simulates data management services. A few MapReduce tools have been simulated too; for instance, in [20] the authors simulate a Hadoop cluster and use it to study the impact of data locality, network topology, and failures on applications. In [21], the authors propose a MapReduce simulator using GridSim.

Simulators are often used to assess and compare scheduling algorithms [22,23]. In [22], the authors propose a MONARC-based solution to analyze the performance of grid scheduling algorithms for tasks with dependencies (DAGs). The Alea 2 simulator presented in [23] is a grid and cluster scheduling simulator based on GridSim and designed for the study, testing and evaluation of various job scheduling techniques. Alea 2 contains workload parsers that can read popular trace formats such as the Grid Workloads Format (GWF) and the Standard Workloads Format (SWF). However, none of these simulators validate simulation results against a "ground-truth" such as logs or results obtained in production.

In [24,25], simulations are used for experiments run by CERN teams (ALICE and ATLAS) on the European Grid Infrastructure (EGI) to study the performance of production grid services, e.g. job processing and data management. In [24], the authors describe a usage of the MONARC framework for a simulation of the job processing performance at an ALICE Tier-2 site. They use the MONARC2 simulation tool to study the efficiency of job processing, the storage element performance and the data transfer speed at the ALICE Tier-2 site in Prague. The authors mention that they performed a validation study consisting in reproducing in simulation the performance of a real production storage element. They affirm that "from comparison of the simulated and actual plots we concluded, that the simulation tool works according to our expectations", but unfortunately no actual data supporting this validation are presented. In [25], the authors simulate the ATLAS Distributed Data Management system using the SimGrid toolkit. The proposed simulation framework is evaluated based on the comparison with historical workloads. Results show an overall simulation error of 20%.

To increase simulation accuracy, production logs can be used as input for simulators as reported in [2,26]. In [2], the authors analyze the distribution of job waiting times in EGI and they perform simulations to compare their proposed late-binding (pilot-job) model with the classical job submission method. In [26], the authors mention that to evaluate their Ex-PERT scheduling framework "in a variety of scenarios yet within our budget", they

4 https://vip.creatis.insa-lyon.fr.