ARTICLE IN PRESS
JID: SIMPAT [m3Gsc;December 15, 2017;5:26]
Simulation Modelling Practice and Theory 000 (2017) 1–16
Contents lists available at ScienceDirect
Simulation Modelling Practice and Theory
journal homepage: www.elsevier.com/locate/simpat

Distributed simulation optimization and parameter exploration framework for the cloud[☆]

Michele Carillo[a], Gennaro Cordasco[b], Flavio Serrapica[a], Vittorio Scarano[a], Carmine Spagnuolo[a,*], Przemysław Szufel[c]

[a] *Università degli studi di Salerno, Italy*
[b] *Università degli Studi della Campania "Luigi Vanvitelli", Italy*
[c] *Warsaw School of Economics (WSE - SGH), Poland*

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Agent-based simulation
Parallel computing
Distributed computing
Simulation optimization
Model exploration
Cloud computing

ABSTRACT

Simulation models are becoming an increasingly popular tool for the analysis and optimization of complex real systems in different fields. Finding an optimal system design requires performing a large sweep over the parameter space in an organized way. Hence, the model optimization process is extremely demanding from a computational point of view, as it requires careful, time-consuming, complex orchestration of coordinated executions. In this paper, we present the design of SOF (Simulation Optimization and exploration Framework in the cloud), a framework which exploits the computing power of a cloud computational environment in order to carry out effective and efficient simulation optimization strategies. SOF offers several attractive features. Firstly, SOF requires "zero configuration", as it does not require *any* additional software installed on the remote node; only standard Apache Hadoop and SSH access are sufficient. Secondly, SOF is transparent to the user, since the user is totally unaware that the system operates on a distributed environment. Finally, SOF is highly customizable and programmable, since it enables the running of different simulation optimization scenarios using diverse programming languages – provided that the hosting platform supports them – and different simulation toolkits, as developed by the modeler. The tool has been fully developed and is available on a public repository[1] under the terms of the open source Apache License. It has been tested and validated on several private platforms, such as a dedicated cluster of workstations, as well as on public platforms, including the Hortonworks Data Platform and Amazon Web Services Elastic MapReduce solution.

1. Introduction

Complex system simulation is gaining relevance in business and academic fields as a powerful experimental tool for research and management. Simulations are mainly used to analyze behaviors that are too complex to be studied analytically,

☆ A preliminary version of this paper was presented at the 24th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2016), Heraklion, Crete, Greece, February 17–19, 2016.

* Corresponding author.
E-mail addresses: michele.carillo@gmail.com (M. Carillo), gennaro.cordasco@unicampania.it (G. Cordasco), flavio.serrapica@gmail.com (F. Serrapica), vitsca@dia.unisa.it (V. Scarano), cspagnuolo@unisa.it (C. Spagnuolo), pszufe@gmail.com (P. Szufel).

[1] SOF GitHub public repository, https://github.com/isislab-unisa/sof.

https://doi.org/10.1016/j.simpat.2017.12.005
1569-190X/© 2017 Published by Elsevier B.V.

Table 1
Simulation-optimization algorithms can be divided into four computational categories (a list of example SO algorithms is presented). The goal of this paper is to provide a framework that allows rapid deployment of computational infrastructure supporting all of the proposed categories - with particular focus on the parallel simulation-optimization problem.

Parameter space	Computation type	
	Sequential (single thread)	Parallel (multi-process)
discrete	KG [20], IZ [35], PGS [9] VIP [12]	NHH [37], NSGS [35] OCBA [11] SKG, AKG, AOCBA [27]
continuous	Stochastic Krigging [5] RSM [33]	GP-BUCB [16], $EI^{(\mu,\lambda)}$ [26]

or too costly to be tested experimentally [28]. The representation of such complex systems results in a mathematical model comprising several parameters. Hence, the question arises as to how changes in model parameter values influence model output (simulation meta-modeling, e.g. [6]) and how to find model parameter values that yield minimum (maximum) model output (simulation-optimization e.g. see [19,47]). Considering the multi-dimensionality of the parameter space, exploring parameter values and determining the optimal parameters configuration is by no means an easy undertaking and requires extensive computing power.

In this paper a software framework for Simulations Optimization (SO) is developed. SO is understood as techniques studied for ascertaining the parameters of the model that minimize (or maximize) given criteria (one or many), which can only be computed by performing a simulation run — see [25,47]. There are several approaches to addressing simulation-optimization problems depending on model type. Simulation-optimization model types can be classified either by algorithm type or by computation type.

The following set SO model classes are considered in the literature (see [7,19]): (1) Ranking and Selection, (2) Response Surface Methodology (RSM), (3) Gradient-Based Procedures, (4) Random Search, (5) Sample Path (6) Optimization and Meta-heuristics. Since many methods exist for parallel simulation-optimization there is a need for a general computational approach to manage such a process.

From a computational point of view, the problems considered in the literature can be classified according to decision space (continuous and discrete) and parallelization (single-threaded vs multi-threaded). Most simulation-optimization policies assume sequential data processing and hence do not allow for parallelization. To illustrate the proposed classification approach a list of sample simulation-optimization algorithms and their classes has been presented in Table 1.

Branke et al. [9] note that since several approaches are available for simulation-optimization, a researcher should choose the one which has the best performance for a given problem. However, in the case of parallel, distributed algorithms, implementing the appropriate computational infrastructure is a very sophisticated task. The goal of this paper is to provide a distributed software framework that will streamline this complex process.

There are several frameworks for distributed computing including MPI, task management platforms such as SGE, and Hadoop. Ni et al. [36] compare MapReduce applicability with a message passing interface (MPI). They found out that MapReduce has a much higher overhead – in particular in scenarios where running simulations requires large amount of input data (they give a geographic system simulation as an example). However, they also note that the MapReduce model is much more appropriate for cloud computing environments, like Amazon EC2, where nodes might become unavailable during computations. Indeed, Apache Hadoop [44] offers built-in protection against core failures by having the master core periodically detect the status of worker cores and re-launch failed map or reduce tasks. Moreover, the Hadoop distributed file system (HDFS) [44], which is used by Apache Hadoop to keep MapReduce input and output, also maintains replicates of data blocks to ensure that no data is lost as a result of a single hardware failure. Therefore, the increased overhead of those fault tolerance mechanisms is often offset by the decrease in cost of using cheaper resources.

This raises the need for tools which exploit the computing power of parallel systems to improve the effectiveness and the efficiency of SO strategies. The crucial characteristics of such tools are: *zero configuration, ease of use, programmability* and *efficiency*. Zero configuration and ease of use are required because both the design and the use of SO strategies are performed by domain experts, who are seldom computer scientists and have limited knowledge of managing modern parallel infrastructures. Programmability is mandatory because different models usually require different SO strategies. Finally, the system must be efficient in order to meet the huge demand for computing power.

In the paper the specific problem of introducing or evaluating new ways to explore the parameters space is addressed — a support tool to perform the parameters space exploration and SO on stochastic simulation model is developed. The proposed framework exploits the computing power of a cloud computational environment in order to quickly carry out simulation exploration and/or optimization strategies.

The applicability of the proposed SOF framework will be illustrated by an agent-based simulation model (ABSM). ABSM enables the reproduction of complex and significant aspects of real phenomena by defining a small set of simple rules regulating how agents interact in social structures and how information is spread from agent to agent. Agent-based models have been successfully applied in several fields such as biology, sociology, economics, military and infrastructure – for a