Performance Evaluation 82 (2014) 15–38

Contents lists available at ScienceDirect

Performance Evaluation

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

Blending randomness in closed queueing network models

Giuliano Casale [a,*], Mirco Tribastone [b], Peter G. Harrison [a]

[a] Department of Computing, Imperial College London, SW7 2AZ, London, UK

[b] Electronics and Computer Science, University of Southampton, UK

ARTICLE INFO

Article history:
Received 3 January 2013
Received in revised form 5 June 2014
Accepted 11 September 2014
Available online 6 October 2014

Keywords:
Random environments
Fluid models
Iterative approximation
Transient analysis
Laplace transform

ABSTRACT

Random environments are stochastic models used to describe events occurring in the environment a system operates in. The goal is to describe events that affect performance and reliability such as breakdowns, repairs, or temporary degradations of resource capacities due to exogenous factors. Despite having been studied for decades, models that include both random environments and queueing networks remain difficult to analyse. To cope with this problem, we introduce the *blending algorithm*, a novel approximation for closed queueing network models in random environments. The algorithm seeks to obtain the stationary solution of the model by iteratively evaluating the dynamics of the system in between state changes of the environment. To make the approach scalable, the computation relies on a fluid approximation of the queueing network model. A validation study on 1800 models shows that blending can save a significant amount of time compared to simulation, with an average accuracy that grows with the number of servers in each station. We also give an interpretation of this technique in terms of Laplace transforms and use this approach to determine convergence properties.

1. Introduction

The recent growth of interest in cloud computing has lead researchers to investigate systems that operate in environments shared by multiple tenants. Modelling these environments is challenging, since performance and availability levels for a user may change over time in a complex manner depending on the activity of the other tenants. For example, cloud virtual machines may experience transient CPU contention periods due to the activity of other virtual machines co-located on the same physical host. Similarly, a web farm deployed on the cloud may experience network bandwidth fluctuations due to traffic generated by other tenants. Accurate models of these systems require the ability to characterise exogenous events that arise in the operational environment of a system.

In performance and availability prediction, the problem of environment modelling may be tackled by associating a state, called a *stage*, to each possible condition of the environment. The active stage of the environment is assumed to evolve in time according to a Markov process. The description of the system then becomes parametric in the currently active stage. A model of this kind is said to describe the system as operating in a *random environment*. Queueing systems operating in random environments have been investigated for decades [1,2], however models involving queueing networks are often intractable, calling for the development of customised performance evaluation methodologies.

Here we focus on systems that can be modelled as closed queueing network models, for example multi-tier software systems [3]. There is a limited literature on closed systems operating in random environments, often because they do not

* Corresponding author. Tel.: +1 757 221 2029; fax: +1 757 221 1717.
E-mail addresses: cg025mf@gmail.com, g.casale@imperial.ac.uk (G. Casale), m.tribastone@soton.ac.uk (M. Tribastone), pgh@doc.ic.ac.uk (P.G. Harrison).

http://dx.doi.org/10.1016/j.peva.2014.09.001
0166-5316/

enjoy closed form expressions of the state probabilities except in particular cases [4–6]. In comparison, several works have focused on open systems operating in random environments, which are often tractable by means of conditional PASTA arguments [7]. The lack of results for the closed case limits their tractability to small problems where the underlying Markov model can be evaluated numerically. Alternatively, simulation may be used, but its running time makes it impractical for optimisation studies.

A computationally more viable route is offered by analytical approximations of closed networks based on product-form queueing theory. In the *average-environment* approximation [8], the system is studied with exponential state transitions having rates equal to their average value when the environment reaches equilibrium. This captures well environments where transitions happen *frequently* with respect to the representative timescales of the system. Conversely, in the *decomposition* approximation, an isolated model is evaluated for each stage of the random environment. The isolated solutions are then weighted across stages using the equilibrium distribution of the random environment [9,10]. This approach describes well an environment where transitions happen *rarely* compared to the representative timescales of the system (e.g., component failures). However, in intermediate situations both types of approximations tend to be loose. While numerical iterative methods exist to address this type of issue [11], they are unable to cope with state spaces of the scale considered in large queueing network models. This calls for developing approximation techniques for queueing network analysis in random environments that are more robust than average-environment and decomposition approximations.

To cope with this challenge, we present the *blending* algorithm, an iterative approximation scheme for closed queueing networks operating in random environments. Blending evaluates at each iteration the dynamics of the system in between two successive stage changes using transient analysis. A sequence of these analyses is used to determine a fixed point for the transient trajectories. The fixed point is then used to compute performance measures of the system for all stages of the random environment. From such embedded averages, it is simple to approximate the long-term equilibrium behaviour of the system and thus estimate its performance measures. We also interpret this value as a Laplace transform evaluated at a specific point and use this interpretation to develop a convergence analysis for the method.

The main challenge of the blending approach is to represent the transient dynamics of the system in between two successive stage changes. Since queueing network models often have an intractably large state space, we cannot use direct numerical methods to evaluate the underlying Markov process. To cope with state space explosion, we use a fluid approximation based on ordinary differential equations (ODEs) defined in the sense of Kurtz [12]. Even though Kurtz's theory has found a wealth of applications in a broad range of disciplines, including chemistry [13], ecology [14], randomised algorithms [15], and communication networks [16], the application to closed models in random environments seems novel. Using Kurtz's theory, we analyse the queueing network as a density-dependent process. This process is used to approximate inexpensively the transient evolution of queue-lengths in between successive stage changes. The number of ODEs does not increase with the number of jobs, which is the parameter that most influences the state space size, thus enabling the evaluation of large-scale models.

We evaluate the algorithm by a comprehensive numerical study consisting of 1800 models. We solve these models by the blending algorithm and compare results with simulation. The study is quite extensive, corresponding to a month of computation time on a private cluster. The results for blending are in good agreement with simulation, but are far less expensive computationally.

The rest of this paper is organised as follows. Section 2 provides a motivating example. The class of models we focus on is defined in Section 3. The high-level structure of the proposed algorithm is defined in Section 4 and further developed in Section 5, where we introduce a fluid approximation for the class of models considered. In Section 6 we develop the blending algorithm and discuss its convergence in Section 7. Numerical results on random instances are given in Section 8. Related work is discussed in Section 9, followed by conclusions in Section 10. Appendix provides related material on Markovian random environments.

2. Motivation

To motivate the present work, we present a toy example illustrating the effects of a random environment on the performance measures of a closed queueing network model. The interested reader can find similar examples in previous work, e.g., for open models [7]. The goal of this motivating example is to build intuition on the type of transient behaviours that a closed network can exhibit in the presence of a random environment. As we show in Appendix, the statements in this section hold for a more general class of systems than queueing networks.

Consider a tandem closed network composed of a first-come first-served (FCFS) single-server queue and a delay station (infinite-server queue, $-/GI/\infty$). The network topology is cyclic, with a population of $N = 2$ jobs. Service times at the single-server queue are exponentially distributed with rate μ. The queueing network operates in a Markovian random environment that causes server breakdown at the single-server queue with rate α_{21}, and subsequent repair with rate α_{12}. The random environment is independent of the state of the stations. Before breakdown, the single-server queue processes jobs with rate $\sigma_2 > 0$, but upon a breakdown it switches to a rate $\sigma_1 < \sigma_2$ until repaired. This simple system may be described by a continuous-time Markov chain (CTMC) with infinitesimal generator $\boldsymbol{Q}$ embedding an environment with generator $\boldsymbol{E}$, where

$$\boldsymbol{Q} = \left[\begin{array}{c|c} \boldsymbol{Q}_1^* - \alpha_{12}\boldsymbol{I} & \alpha_{12}\boldsymbol{I} \\ \hline \alpha_{21}\boldsymbol{I} & \boldsymbol{Q}_2^* - \alpha_{21}\boldsymbol{I} \end{array}\right], \qquad \boldsymbol{E} = \begin{bmatrix} -\alpha_{12} & \alpha_{12} \\ \alpha_{21} & -\alpha_{21} \end{bmatrix},$$