Performance models for analysis and control of it systems

Hristov, A.

2018

document version
Publisher's PDF, also known as Version of record

Summary

The increasing complexity of IT infrastructures poses significant challenges in managing computer systems. Ensuring efficient usage of the available resources while preserving the desired Quality of Service (QoS) demands one to go beyond ad-hoc solutions. Motivated by this, we develop tools and methods to evaluate the performance of ICT service chains, and furthermore, to manage their control in an optimal manner. One of the main goals of our research is to bridge the gap between theory and practice. Therefore, we present possible IT applications as a context for each of the studied techniques. We explore a new solution concept by combining the fields of queueing theory and machine learning. We believe that the introduced techniques have a great potential offering remarkably accurate, and at the same time easily scalable, generic solutions to real-world problems.

Motivated by caching in database application, in Chapter 2 we analyze a tandem queueing model consisting of two servers where the output of one of the servers becomes the input of the second node. More specifically, we model the application level sending write requests to the cache as the first server and the cache as the second one. Furthermore, we introduce the possibility to switch on/off any of the two servers at any given moment. Next to that, we assume holding costs at each of the two queues and a start-up cost for the second server, i.e., the cache. This way, we formulate a Markov Decision Process (MDP) with an optimal control policy defined as the one minimizing the long-term average costs of the system. However, the existing numerical techniques to compute this optimal policy are inefficient or even unfeasible for some real-world applications. Therefore, we present a scalable approximation algorithm to obtain a threshold-type decision policy. The researched technique is rather intuitive and the approximated policy results in long-term average costs within a few percentages of the optimal. Therefore, we believe that our method contributes to expanding both the theoretical and the practical knowledge on the matter.

Chapter 3 further explores the performance evaluation and optimal control

of database caching mechanisms. In contrast to the preceding chapter, this chapter focuses on the so-called 'write-behind' cache mechanism. We model the system as a single server that processes jobs in two stages. The first stage corresponds to pre-processing jobs one at a time, i.e., requests being written in the cache. After pre-processing the jobs are accumulated in a batch of given size K and served at once, i.e., the requests are being transferred from the cache to the database. The goal of the study is to identify the batch size K that minimizes the expected waiting time of arriving jobs. To approximate the optimal value of K, we first derive an analytic solution for the stationary distribution of jobs in the system for two special cases: $K = 1$ and $K = 2$. Next, we outline a fluid approach which results in an approximation of the expected waiting time for large batch sizes. Finally, we show how to combine the insights from these techniques to find the optimal batch size in an efficient and scalable way. Extensive numerical experimentation shows that the approximation works extremely well for a wide range of parameter combinations.

In Chapter 4 we introduce a new technique to obtain closed-form approximations of the optimal threshold-based policy for MDPs. Our method uses the Symbolic Regression (SR) algorithm. We present how to significantly improve both the accuracy and the execution time of this evolutionary algorithm by tailoring it to the corresponding MDP framework. Applying our approach on two running examples results in analytic expressions that approximate the optimal control policy with great accuracy. Next to that, we show that the obtained mathematical formulas allow sensitivity analysis of the system parameters. In addition, the opportunity to instantly calculate a new threshold function for any change in the parameters makes our solution particularly appealing for an application in real-time systems. Furthermore, we believe that the introduced technique is highly generic and is applicable to a broad range of other MDPs.

In Chapter 5 we further research the potential of combining queueing theory together with tools from machine learning. We introduce a way to incorporate insights and results derived from queueing theory techniques into SR. The presented method leads to closed-form approximations for the relevant performance metrics of a given system. To illustrate the technique, we apply it to the so-called two-stream blending system. We use our method to obtain closed-form expressions for the mean waiting time and the throughput. The generated analytic formulas are remarkably accurate and at the same time algebraically simple.

Finally, we abstract over the models considered to this point and adopt a high-level view of the ICT service chains. In Chapter 6, we analyze a Layered Queueing Network (LQN) in which a given number of servers are organized in a nested fashion, and therefore influence each others service rate. We present a simple, computationally tractable and nevertheless highly accurate approximation algorithm for obtaining the servers' utilization rates and throughput of the given nested system. We believe that the high accuracy of the approximation opens up opportunities to extend the model to more general LQNs. Furthermore, we show that even when performance metrics such as the saturation throughput and the utilization rates of the servers are known, determining the limiting factor in the network is far from trivial. Therefore, we introduced a server-wise metric for identification of the bottleneck that extends the intuitive 'slowest server rule'. The conducted numerical tests show that the proposed bottleneck identification technique is more accurate than the existing algorithms so far. Next to that, being an extension of the 'slowest server rule', our method is applicable to queueing networks without a layered structure as well.