Evaluating Performance Attributes of Layered Software Architecture

Vibhu Saujanya Sharma¹, Pankaj Jalote¹, Kishor S. Trivedi²

¹ Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur, INDIA, 208016
{Vsharma, Jalote}@cse.iitk.ac.in
² Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA
Kst@ee.duke.edu

Abstract. The architecture of a software system is the highest level of abstraction whereupon useful analysis of system properties is possible. Hence, performance analysis at this level can be useful for assessing whether a proposed architecture can meet the desired performance specifications and can help in making key architectural decisions. In this paper we propose an approach for performance evaluation of software systems following the layered architecture, which is a common architectural style for building software systems. Our approach initially models the system as a Discrete Time Markov Chain, and extracts parameters for constructing a closed Product Form Queueing Network model that is solved using the SHARPE software package. Our approach predicts the throughput and the average response time of the system under varying workloads and also identifies bottlenecks in the system, suggesting possibilities for their removal.

1 Introduction

Software architecture is an important phase in software lifecycle as it allows taking early design decisions about a system. Moreover it is also the earliest point in system development at which the system to be built could be analyzed [7], [9]. Analysis of a system at the architectural level enables the choice of the right architecture for the system under consideration, thus saving major potential modifications later in the development cycle or tuning the system after deployment.

Out of the various attributes that could be assessed, performance attributes are most sought after in any software system. Performance is an umbrella term describing various aspects, such as responsiveness, throughput, etc. of the system. Assessing and optimizing these aspects is essential for the smooth and efficient operation of the software system. There have been many approaches [2],[3],[4],[5],[12] for performance evaluation of software systems, the pioneering work being done by C.U. Smith, [2] which introduced the concept of Software Performance Engineering (SPE).

Layered software architecture is a very prevalent software architectural style that is followed by almost all client-server and web based systems. Layered architecture
helps to structure applications that can be decomposed into n groups of subtasks in which each group is at a particular level of abstraction with well-defined interfaces [8]. The \(i^{th}\) layer could communicate with only the \((i-1)^{th}\) and the \((i+1)^{th}\) layer. Layered architecture is widely used in almost all web-based systems where performance is a critical factor. Hence, performance analysis of layered systems is of much importance to system architects. Moreover as a large number of layered systems already exist, performance predictions with varying number of clients or with the addition or scaling up of components in the system would be beneficial to system administrators, who manage such systems.

In this paper, we present an approach for performance evaluation of software systems following the layered architectural style. In the past SPE has been largely seen as an activity, which requires specialized skills and in-depth knowledge of both software architecture and performance modeling. Thus one of the motivating factors of our effort is to provide an approach that could be used by software engineers for designing new systems and system administrators for tweaking existing systems alike. The aim of the approach is to output the traditional performance parameters as well as suggest to the user bottleneck components that need to be scaled up. Our system thus removes the performance analyst from the loop in that the activities traditionally performed by him/her are automated.

Our approach consists of modeling the layered software system as a closed Product Form Queueing Network (PFQN) [1], and then solving it for finding performance attributes of the system. One of our aims is to ask for specifications that are easy to provide even for someone who is not an expert in this field. After getting the specifications we model the system initially as a Discrete Time Markov Chain (DTMC), with each layer in the system, corresponding to a state in the DTMC [13]. This DTMC is then analyzed to find the total service requirements of the software system over the different hardware nodes or machines. The closed PFQN model is then constructed using this information along with the specifications given by the user. Modeling machines having limited software resources such as threads is also performed at this stage using a hierarchical approach.

This closed PFQN model is then fed to SHARPE [11] which is a versatile software package for analyzing performance, reliability and performability models. The output from SHARPE is then further analyzed, and the results include the classical performance metrics such as the throughput and the average response time along with information about system bottlenecks and suggested scale-ups for them. Along with these, it predicts the improvement in system performance if the suggested scale-ups are done. This is done by reconstructing the model internally, accommodating the scaled-up components and solving it again, using our approach.

The tool, which we have developed as an implementation of this approach, requires minimal knowledge of queueing models or any other performance modeling techniques to use it. The specifications which are needed could be easily procured and hence the tool facilitates modeling new systems as well as helping in scaling existing systems.