High-level Approach to Modeling of Observed System Behavior

Thomas Begin [1] Alexandre Brandwajn [2] Bruno Baynat [3] Bernd E. Wolfinger [4]
Serge Fdida [5]

Abstract

Current computer systems and communication networks tend to be highly complex, and they typically hide their internal structure from their users. Thus, for selected aspects of capacity planning, overload control and related applications, it is useful to have a method allowing one to find good and relatively simple approximations for the observed system behavior. This paper investigates one such approach where we attempt to represent the latter by adequately selecting the parameters of a set of queueing models. We identify a limited number of queueing models that we use as "Building Blocks" (BBs) in our procedure. The selected BBs allow us to accurately approximate the measured behavior of a range of different systems. We propose an approach for selecting and combining suitable BB, as well as for their calibration. Finally, we validate our methodology and discuss the potential and the limitations of the proposed approach.

1 Introduction

A commonly used method for analytic performance modeling of computer and communication systems, which we refer to as the constructive approach, is to attempt to reproduce in the mathematical model essential aspects of the system structure and operation [6]. This constructive approach has its limits. First, important aspects of large and heterogeneous computer or communication systems may be largely unknown. Second, extensive knowledge and expertise might not be available to correctly identify key system components and features lest the resulting models become unrealistic or intractable in their complexity. These difficulties motivate in part our approach. In our high-level modeling, we don't necessarily seek to "mimic" the structure of the system under study. Rather, we focus on the observable behavior as given by measurements of the system, and attempt to infer from it a possible high-level model structure capable of adequately reproducing the observed system. In doing so, we forego the detailed representation of the system in favor of the possibility that a relatively simple model, not necessarily related to the apparent structure of the system, might be able to capture the behavior of the system under consideration. An obvious justification for our approach is that, even in a complex system, it is possible that a small number of components, or a single component, may be the critical bottleneck, effectively driving the system behavior. This idea is by no means novel, and has been frequently employed in the past, e.g. in the case of an Internet path ([9] and [1]), disk arrays ([11]), time-sharing system ([10]) and a Web server ([4]).

Our approach has several advantages. First, it requires a priori little information about the system, and can be easily embedded in an automatic software tool that does not need any special modeling or queueing theory expertise from the end user. Second, our approach may provide the performance analyst with a ready-to-use model to generate reliable estimates for system performance at other workload levels, without the expense and the effort of obtaining additional measurements. Finally, it may help discover properties of the system not immediately apparent from the system structure by delivering a simple model able to adequately represent the system.

2 General framework

Systems considered in our study may represent a whole computer or communication system, or specific components such as processors, a disk array, an Ethernet network or a WLAN, etc. Requests refer to the individual entities that are treated by the system, such as packets or frames in the case of networks, I/O requests in the case of storage systems, HTTP requests in the case of web servers, etc. The workload (offered load) includes all the requests that are submitted to the system for treatment. In our view, the system performance changes in response to the workload, and these changes are reflected in the corresponding measurements.

Our approach relies on the availability of measurements of specific system performance parameters such as throughput, loss probability, average response time and queue length, denoted by $\bar{X}_{mes}$, $\bar{L}_{mes}$, $\bar{R}_{mes}$ and $\bar{Q}_{mes}$ respectively. The throughput $\bar{X}_{mes}$ represents the average number of requests that leave the system per unit time (this quantity may differ from the offered workload if the system is subject to losses). $\bar{L}_{mes}$ gives the probability that an arriving request is rejected. $\bar{R}_{mes}$ defines the average sojourn time (waiting for and receiving

[1] T. Begin is a Phd candidate at the Universite Pierre et Marie Curie, LIP6, France. email: thomas.begin@lip6.fr

[2] A. Brandwajn is a Professor at the University of California Santa Cruz, Baskin School of Engineering, USA. email: alexb@soe.ucsc.edu

[3] B. Baynat is an Assistant Professor at the Universite Pierre et Marie Curie, LIP6, France. email: bruno.baynat@lip6.fr

[4] B.E. Wolfinger is a Professor at the Universitaet Hamburg, Dept. Informatik, Germany. email: wolfinger@informatik.uni-hamburg.de

[5] S. Fdida is a Professor at the Universite Pierre et Marie Curie, LIP6, France. email: serge.fdida@lip6.fr

service) experienced by a request inside the system. Finally, $\bar{Q}_{mes}$ represents the average number of requests in the system. Note that, by Little's law [8], $\bar{Q}_{mes} = \bar{X}_{mes} \times \bar{R}_{mes}$ so that it suffices to measure any two of these three quantities. Each measurement point corresponds to a set of performance parameters that have been measured at a particular state of the load (e.g. $(\bar{X}_{mes}, \bar{R}_{mes})$) and may in general also include input parameters such as the corresponding offered load. A total of n measurement points for the same system constitutes a set of measurements. In computer networks, a point from a set of measurements may contain typical performance parameters such as the throughput at an interface, the time spent by packets inside the network and possibly the packet loss ratio. In disk arrays, a set of measurements may include parameters such as the I/O response time, I/O request throughput, device utilization, etc.

3 High-Level Modeling

3.1 Set of Building Blocks

One of the premises of our high-level approach is that a complex system may exhibit behavior that can be reproduced by a relatively simple queueing model. We use a set of generic models that we call "Building Blocks" (denoted by "BBs" in the rest of the paper). BBs include queues such as the M/M/C, M/M/C/K, M/G/1, M/G/1/K [3], as well as the M/G/C approximation [7], and original queueing systems whose service times are driven by the congestion parameters of an embedded model. Reference [2] discusses in more detail these models with load-dependant service times. To represent the fact that, in some systems, the response time comprises a fixed overhead as an additive load-independent component, we expand our BBs to include a fixed "offset" value (denoted by *Off* in figures). Note that this offset does not affect the congestion at the server, and the response time in our BBs is simply the sum of the offset value and the response time at the server.

3.2 Error criterion

We need a way to measure the goodness of fit of a given model (a Building Block together with a set of values of its parameters) versus the measurement set. This is the role of the error criterion, referred to as θ, that aims at providing a convenient way to compare fairly any models. A simple way to define this function is to consider the sum of the deviations between mean sojourn time obtained from measurements and the one obtained from the model for values of throughput equal to the measured throughput (see Figure 1). θ can be formally expressed by relation 1, where $\bar{R}_{mes,i}$ $(i = 1, \ldots, n)$ are the measured mean response time values, and $\bar{R}_{th,i}$ $(i = 1, \ldots, n)$ are the corresponding mean response times obtained from a given model. Some adjustments to the definition of θ are possible in order to take into account absolute and relative components for deviations, as well as confidence weights associated with measures.

$$\theta = \sum_{i=1}^{n} |\bar{R}_{th,i} - \bar{R}_{mes,i}| \quad (1)$$

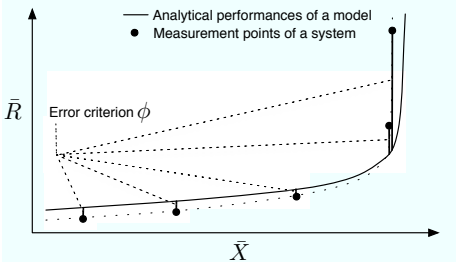

Figure 1: Error criterion

3.3 Search for an adequate model among the Building Blocks

In our high-level approach, we attempt to automatically calibrate each BB separately and select among them the "best" model (the best BB with the best calibration). By calibration of a BB we mean the search for a set of values of model parameters that minimizes the error criterion θ. In general, this leads to a non-linear numerical regression problem. We avoid algorithms based on derivatives of θ. In most cases, differentiating θ, if at all possible, is time consuming and it is specific to each BB. We cast the calibration of a BB as a numeric optimization problem, and we choose to employ Derivative Free Optimization (DFO) methods [5]. These methods have the advantage that no derivatives are invoked or estimated. They are not specific to a particular BB, so that the introduction of a new BB is an easy task. In our specific implementation, we use a local quadratic approximation, which implies a low computational cost while speeding up the convergence. The results of our experiments indicate that the proposed search method tends to be robust and very fast for BBs with a limited number of parameters (say, up to 5 or 6). More details about the technique can be found in a technical report [2].

3.4 Requirements for the methodology

Measurements represent a key component for our approach. To be of use, the sets of measurements must satisfy certain common sense conditions. First, all measurement points from a particular set must come from the same system whose structural properties must not change between measurements. In particular, the background traffic that shares the system resources with the measured traffic should be either negligible, constant, or in a clear relationship to the measured traffic for all measurement points. Second, the available measurement data must adequately capture the salient features of system behavior in the range of interest. Clearly, for instance, if the system response exhibits an inflection point and this point is not present in the measurement data, there is little chance that the model proposed by our approach will correctly reproduce such a behavior.

4 A case study from real-life disk controllers

The model selected by our proposed approach is referred to as the laureate model. In addition to simply matching the data

points in the measurement set, we would want the laureate model to be able to correctly predict the performance of the system within some reasonable domain. Therefore, we deliberately remove one or more data points from the measurement sets. Having found the laureate model for such reduced data set, we then test the ability of this model to predict the system performance at the removed data points.

We present here two sets of measurements obtained for disk controllers in mainframe environments. The measurement points give the expected I/O response time as a function of the attained I/O throughput (in requests per time unit) for measurement Set 1 and Set 2, respectively in Figures 2 and 3.

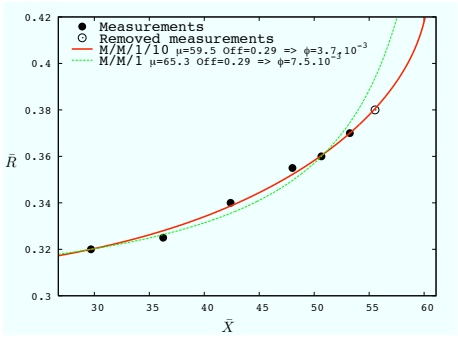

Figure 2: Disk controllers - Set 1

We observe in Figure 2 that the best model - among the BBs considered - in this case is the M/M/C/K queue, while for the measurement set of Figure 3 it is the M/G/1 queue. In both cases, the laureate models not only closely reproduce the data points used in the calibration process, but are also able to adequately predict the performance for the removed data points. The relative prediction errors for the removed data points are below 1% both for the expected throughput and the mean I/O time in Figure 2, and below 5% in Figure 3. We note that removing different points during the calibration process leads to similarly successful predictions by the laureate models.

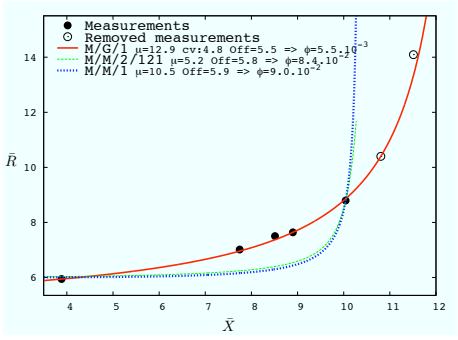

Figure 3: Disk controllers - Set 2

Our approach has been successfully applied to various real-life systems such as networks, as well as a multiprocessor system. These case studies are described in a technical report [2].

5 Conclusions

We have presented a high-level modeling approach based on measurement data. Unlike in constructive modeling, we don't seek to represent "explicitly" the structure of the system being studied. We focus on the measurement results, and attempt to discover a more or less elementary model that might correctly reproduce the observed behavior. Our main contribution lies in the automation of the search for the laureate model among a set of "Building Blocks" (BB). As a result, performance analysts with a minimal queueing network background can use the resulting tool. The laureate models obtained from our approach are useful to predict performance at workload levels for which measurements may not have been obtained. The nature of the best-fitting BB may also be of help for constructive modeling of the system. Indeed, it may provide guidance in the search for simple approximations, by indicating which BB may and which ones may not work. The potential drawback of our approach is that there is in general no clear readily seen relationship between the parameters of the laureate model and the "natural" parameters of the corresponding constructive model. This limits also the predictive application of the laureate model in that it is not typically clear how the parameters of the laureate should be modified to reflect a change in the characteristics of the system being modeled. However, we believe that, packaged as a ready-to-use tool, our approach can be of significant value both to the performance analyst in capacity planning situation, and to the performance modeler in general.

References

[1] S. Alouf, P. Nain, and D. Towlsey. Inferring network characteristics via moment-based estimators. In *INFOCOM*, pages 1045–1054, 2001.

[2] T. Begin, A. Brandwajn, B. Baynat, B. Wolfinger, and S. Fdida. Automatic Modeling for Black Box Systems. Technical report, LIP6, 2007.

[3] A. Brandwajn and H. Wang. A Conditional Probability Approach to M/G/1-like Queues. 2006. Submitted for publication, available as a technical report.

[4] J. Cao, M. Andersson, C. Nyberg, and M. Kihl. Web server performance modeling using an M/G/1/K*PS queue. In *ICT: 10th International Conference on Telecommunications*, volume 2, pages 1501–1506, 2005.

[5] A. Conn, K. Scheinberg, and P. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming*, 79:397–414, 1997.

[6] P. Heidelberger and S. Lavenberg. Computer performance evaluation methodology. *IEEE Trans. Computers*, 33:1195–1220, 1984.

[7] T. Kimura. Approximations for Multi-Server Queues: System Interpolations. *Queueing Systems: Theory and Applications*, pages 347–382, 1994.

[8] L. Kleinrock. *Queueing Systems, Volume 1: Theory.* 1975.

[9] K. Salamatian and S. Fdida. A framework for interpreting measurement over Internet. In *ACM SIGCOMM workshop*.

[10] A. Scherr. An Analysis of Time-Shared Computer Systems. *MIT Press*, 1967.

[11] E. Varki, A. Merchant, J. Xu, and X. Qiu. Issues and challenges in the performance analysis of real disk arrays. *IEEE Trans. Parallel Distrib. Syst.*, 15:559–574, 2004.