Transaction Profile Estimation of Queueing Network Models for IT Systems Using a Search-Based Technique

Shadi Ghaith, Miao Wang, Philip Perry, and John Murphy

School of Computer Science,
University College Dublin, Ireland
shadi.ghaith@ucdconnect.ie

Abstract. The software and hardware systems required to deliver modern Web based services are becoming increasingly complex. Management and evolution of the systems requires periodic analysis of performance and capacity to maintain quality of service and maximise efficient use of resources. In this work we present a method that uses a repeated local search technique to improve the accuracy of modelling such systems while also reducing the complexity and time required to perform this task. The accuracy of the model derived from the search-based approach is validated by extrapolating the performance to multiple load levels which enables system capacity and performance to be planned and managed more efficiently.

1 Introduction

The IT services offered over the internet, and within enterprise premises, increased rapidly over the past few decades. The expectation of a consistent high-quality level of service is growing and becoming more difficult to be met. In the meanwhile, the demand for a cost effective solution to efficiently use computing resources according to workload variations is getting stronger, especially for enterprise service providers.

Predicting enterprise applications service levels under various loads and resources is widely done based on time and cost efficient performance modelling techniques [1]. A Queueing Network Model (QNM) representing the various system resources (such as the CPU and Disk I/O) is built and solved during this process. One input to the QNM solver is the amount of time required to serve each transaction on each resource in all visits to the underlying resource, while excluding the queuing time. This input value is known as the Service Demand.

Obtaining such service demands by measuring the time spent by a single user transaction on each resource faces a major measurement problem due to small service demands. Additionally, some functionality, such as caching, can be only triggered when multiple users access the system, which leads capacity engineers to infer these service demands from multiuser measurements [2] [3] [4]. Such approaches have problems (we will discuss some of them in Section 2)

C. Le Goues and S. Yoo (Eds.): SSBSE 2014, LNCS 8636, pp. 234–239, 2014.

which lead to various approximations and lengthy procedures. This causes a less accurate prediction results which are usually solved by deliberatly increasing the hardware requirments to compensate for this imprecision, resulting in an under-utilized infrastructure. Additionally, the lengthy procedure can easily put an extra pressure on an already busy Capacity Management (CM) projects.

In this paper, we propose a search-based solution to obtain a more accurate service demands from the data obtained from multiuser performance data. This work improves the capacity planning process by enhancing its accuracy and duration. Also, other processes relying on the service demands, such as the performance regression testing technique [5] [6], will also benefit of this work.

2 Queueing Network Models for Capacity Planning

Transactions are created by users of enterprise applications to perform certain functionalities (such as search and buy). These transactions are served by the various system resources (such as the CPU and Disk I/O of each server) to fulfil the user request [1]. The flow of the transaction through the system can be represented by a Queueing Network Model (QNM) [1] such as the one shown in Figure 1. Each node represents a system resource (e.g. CPU, Disk I/O) which consists of a processing unit and a queue for the transaction to wait if the processing unit is busy. Each transaction may visit each resource multiple times and the total time required to serve the transaction on each resource, during all visits and excluding the time in the queue, is called service demand.

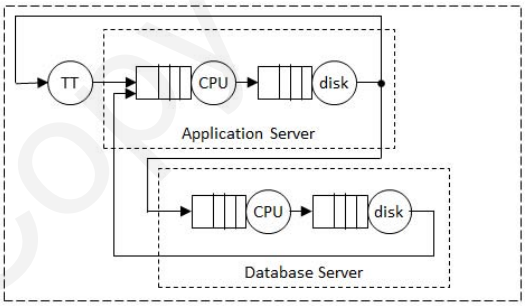

Fig. 1. Queueing Network Model of a Three-Tiered Computer System

The Transaction Response Time (TRT) is the time spent by a certain transaction on all system resources [1] which is the sum of the actual processing time (service demand) along with the time in the queue. The Transaction Profile (TP) is defined as the total series of service demands on all system resources. Thus, the TP is the lower bound of the TRT, or in other words it is equal to the TRT when the transaction is the only one in the system (i.e. no queueing).

Capacity Management (CM) of the system is the process of predicting its performance by solving the QNM (via various known techniques). This provides

predictions of TRTs and Resources Utilizations (RUs) for various load and system configurations as depicted in Figure 2. This typically uses the following three inputs:

1. The QNM of the system, similar to the one shown in Figure 1.
2. The load characterization [1] which includes the number of users issuing each transaction and user Think Time (TT).
3. Service demands (TP) for each transaction at each resource. Our focus in this paper is on enhancing the process to obtain the service demands (TPs).

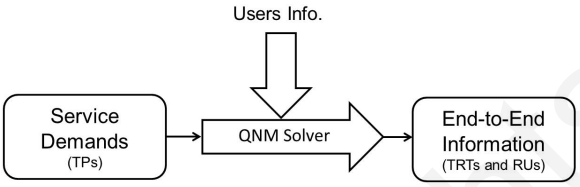

Fig. 2. Solving of Queueing Network Model to Predict System Performance

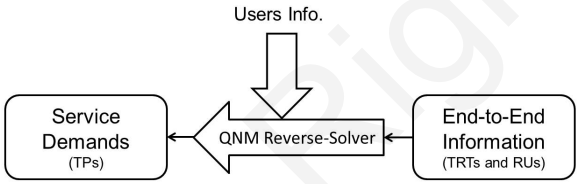

Fig. 3. Presented Approach to Infer TPs From Performance Data (TRTs, RUs)

Estimating the profile of service demands for a given transaction (i.e. the TP) would be most easily achieved by running a single transaction through the system, but this has the following problems:

1. High inaccuracy in measuring service demands due to their small values [3].
2. It is unreliable to measure service demands on virtualised systems [3].
3. Some system behaviours (such as caching) are only triggered under load.

To solve these problems, researchers proposed to focus on measuring the end-to-end parameters, particularly the TRTs and RUs, and to infer the service demands from them [2] [3] [4]. This typically involved running the tests with a number of different load values (time consuming) and simplifying the system to a single node approximation (rarely applicable).

3 Inferring Service Demands

We propose a new method which does not rely on the single node approximation and can use a single multiple user test run at one load level. We propose to

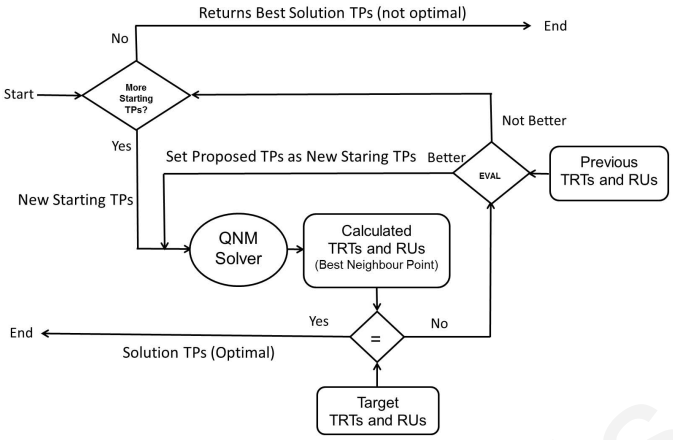

Fig. 4. Reverse-solve QNM by Applying a Search-based Technique

reverse the normal CM process as depicted in Figure 3. The inputs for the QNM reverse-solver are user information, TRTs, RUs while the outputs are the service demands, i.e. the TPs. The QNM reverse-solver is implemented by using the *Repeated Local Search (Continuous Space Hill Climbing)* technique [7] as shown in Figure 4.

The *Local Search* technique is repeated multiple times each with a different starting condition (set of TPs). Given that any TP can have a value ranging from zero to the value of the corresponding TRT, the starting TPs are nominated by evenly dividing the range (0 to TRT) to a predefined number of points. The number is chosen as a compromise between the search time and the accuracy of the result. At the end all the solutions returned by the Local Search (local maximum) are compared, as will be described shortly, and best one is returned.

The quality of the solution (EVal) is defined as the distance between the TRTs (and RUs) and their corresponding target values (ultimate solution) which is calculated as shown below (a smaller value is better):

1. Calculate percentage difference between each proposed and target TRT (RU).
2. For all the TRTs find the average percentage value (Do same for all RUs).
3. Do a weighted summation of the above averages.

The summation of all service demands corresponding to each transaction should equal to the starting TP. Accordingly, the initial service demands are set by dividing the starting TP value among the corresponding service demands either equally or with a predetermined ratio of that transaction. These ratios can be learned by the system and applied to subsequent searches on next releases or similar applications.

The *Local Search* technique loops over all initial service demands. For each one, tests the adjacent points by incrementing (decreminting) that service demand with selected steps. The QNM is solved for all solutions and the outcome is

Table 1. Using the Proposed Technique to Infer TPs from a Multiuser (100 users) Run and Use Them to Predict TRTs at Other Loads and Compare with Measured Values

Transaction Name	**Inferred TP (sec)**	**TRT Predicted and Measured (sec)**			
		Users	**200**	**300**	**400**
Home	0.43	Predicted	0.71	0.91	1.05
		Measured	0.74	0.91	1.11
List Products	0.24	Predicted	0.45	0.79	0.95
		Measured	0.48	0.80	1.01
Search for Products	0.26	Predicted	0.39	0.47	0.65
		Measured	0.42	0.49	0.66

evaluated using EVal and if the best of these neighbour points is better than the current point the search will move to it. The same is done for all points in the loop. If the EVal of the proposed solution (generated by the end of the loop) is lower than an epsilon value then the entire search process concludes with a perfect solution. Otherwise, if the proposed solution is better than the previous one, it is set as a new set of starting TPs point and the loop is restarted over all service demands (Local Search still running). This continues until the proposed solution is not better than the current one (local maximum) so the Local Search iteration concludes and a new one is triggerd with new starting TPs.

4 Experimental Evaluation

In order to evaluate our technique, we performed tests on the reference JEE enterprise application (JPetStore) with 4 performance runs of 100, 200, 300 and 400 users. Each run included 10 transactions applied by a load generator tool. Each run took around one hour and the TRTs and RUs were recorded.

We used the presented technique to infer the TPs using the run with 100 users. The TPs corresponding to a subset of the transactions are shown in the second column of Table 1. Then we used these TPs to predict the TRTs at loads of 200, 300 and 400 users using the process in Figure 2. The last three columns of the Table show the TRTs obtained using the prediction process and direct measurements. We can see that the results are close and the error is within normal limits for the QNM based techniques. The search process to obtain the TPs converged in less than 8 minutes, which yields a significant saving of several hours over the existing inference techniques. Finally, we set the number of Local Search repeats to 10 but all cases converged in maximum of three repeats.

5 Related Work

Inferring the service demands (TPs) from TRTs and RUs has been explored because measuring such counters is much easier than measuring the service demands. Casale et al [2] proposed to do a linear regression between the RUs and the service demands. While Kraft et al [3] proposed a similar approach but to

use TRTs instead of RUs. These two approaches assume the system can be simplified to one node which does not hold true for most IT systems. In addition, multiple runs with different loads are required which is both time and resource consuming. Liu et al [4] proposed a formulation to estimate the parameters of the QNM using a quadratic programming framework. This requires the end-to-end parameters TRTs, RUs and Throughput from a set of runs which is costly.

6 Conclusions and Future Work

Current techniques to obtain service demands used to solve QNMs in CM projects are inaccurate and time consuming. In this paper, we addressed these concerns by proposing a search-based approach to infer the service demands from end-to-end data (TRTs and RUs) collected from only one performance run. We explained the technique using Local Search accompanied with Restarting techniques. In future, we plan to explore other search alternatives in order to improve the performance of the reverse-solver, mainly when the number of transactions becomes large. Also, we believe that many parameters of the search process can be learned and generalized for various enterprise application (and transaction) categories. In addition, we will show the improvement on CM projects and other fields utilizing service demands, such as the regression testing data analysis.

Acknowledgment. Supported, in part, by Science Foundation Ireland grant 10/CE/I1855.

References

1. Grinshpan, L.: Solving Enterprise Applications Performance Puzzles. John Wiley and Sons, Inc., Hoboken (2012)
2. Casale, G., Cremonesi, P., Turrin, R.: Robust workload estimation in queueing network performance models. In: Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing, PDP 2008 (2008)
3. Kraft, S., Pacheco-Sanchez, S., Casale, G., Dawson, S.: Estimating service resource consumption from response time measurements. In: Proceedings of the International ICST Conference on Performance Evaluation Methodologies and Tools (2009)
4. Wynter, L., Liu, Z., Cathy, H.X., Zhang, F.: Parameter inference of queueing models for it systems using end-to-end measurements. Performance Evaluation 63(1), 36–60 (2006)
5. Ghaith, S., Wang, M., Perry, P., Murphy, J.: Automatic, load-independent detection of performance regressions by transaction profiles. In: Proceedings of the 2013 International Workshop on Joining AcadeMiA and Industry Contributions to testing Automation, JAMAICA 2013, pp. 59–64. ACM, New York (2013)
6. Ghaith, S., Wang, M., Perry, P., Murphy, J.: Profile-based, load-independent anomaly detection and analysis in performance regression testing of software systems. In: 17th European Conference on Software Maintenance and Reengineering (CSMR 2013), Genova, Italy (2013)
7. Harman, M., Hierons, R., Jones, B., Lumkin, M., Mitchell, B., Mancoridis, S., Rees, K., Roper, M., Clarke, J., Dolado, J.J., Shepperd, M.: Reformulating software engineering as a search problem. In: Software IEE Proceedings (June 2003)