On the Predictive Performance of Queueing Network Models for Large-Scale Distributed Transaction Processing Systems

Oliver Hühn, Christian Markl, Martin Bichler

Department of Informatics, Boltzmannstraße 3
Technische Universität München
85748 Garching, Germany

Phone: +49 89 289 17537

Fax: +49 89 289 17535

{oliver.huehn, christian.markl, martin.bichler}@mytum.de

http://ibis.in.tum.de

Abstract

Automated business processes running on distributed transaction processing (DTP) systems characterize the IT backbone of services industries. New web services standards such as BPEL have increased the importance of DTP systems in business practice. IT departments are forced to meet pre-defined quality-of-service metrics, therefore performance prediction is essential. Unfortunately, the complexity of multiple interacting services running on multiple hardware resources as well as the volatility in the demand for these services can make performance analysis extremely difficult. While business process automation has been a dominant topic in the recent years, surprisingly little has been published on performance modelling of large-scale DTP systems. In this paper, we will describe these systems with respect to the workloads and technical features, and compare the predictive accuracy of different types of queueing models and discrete event simulations experimentally. The experiments are based on two real-world DTP systems and respective data sets of a telecom company. Overall, we found that while the results for average utilization scenarios are quite similar, the effort to implement and run analytic solutions is much lower. As long as standard distributional assumptions of analytical models hold, they provide a reliable and fast methodology to explore different demand mix scenarios even for large-scale systems. The difficulty to estimate service and arrival time parameters and demand mix for the respective queueing network models can largely be reduced with appropriate tooling. Often, this information is missing in IT departments. Also, complex event conditions and error handling in DTP systems can make the analysis difficult. For many DTP applications, however, performance modelling could provide valuable decision support for service level management.

Keywords: *Performance Modelling, IT Service Management, Transaction Processing, Queueing Network Model, Discrete Event Simulation*

1. Introduction

Automated business processes running on *distributed transaction processing* (DTP) systems are the IT backbone of many businesses these days. DTP systems support the flexible, easily adaptable composition of distributed software services in heterogeneous environments. Such systems can be found in all areas of today's services industries such as the airline, banking, insurance, or telecom sector. Automated business processes describe transactions that need to be executed consistently and reliably across multiple information systems. For example, adding a new customer to a telecommunication company typically requires a credit check, the assignment of a new phone number, entries in the billing and CRM systems, inserts into various databases of the network that the company is operating, etc. In practice, most automated business processes are executed on *transaction processing* (TP) monitors such as BEA Tuxedo®, IBM CISC®, Vitria®, or TIBCO Business Works®.

DTP systems are typically business critical applications and consist of dozens of business processes that are executed on many heterogeneous software systems. They need to be designed for hundreds of thousands of requests a day, often hundreds of them in parallel and need to meet high quality of service standards in terms of response times, throughput, and availability.

Capacity planning and performance modelling for these systems are difficult tasks; in particular since the demand for certain processes can vary considerably over time due to volatility in the demand of consumers and the market in general. Performance problems are a frequent consequence. While investment costs for new hardware have been decreasing, the energy costs in data centres have been increasing during the past years. IDC reports that the cost of power and cooling has increased 400% over the past decade, and these costs are expected to continue to rise [1]. In such an environment, IT service managers need to strike a balance, trying to maintain agreed upon quality of service while minimizing cost of the operations.

Unfortunately, the complexity of multiple interacting processes running on multiple hardware resources makes performance prediction and capacity planning difficult. Distributed systems are already hard to analyze, but the problem becomes even harder when they are composed of black-box components: software from many different (and perhaps competing) vendors, usually with no source code available.

Conceptually, DTP systems can be seen as queueing systems with jobs, waiting lines, and service stations. Service stations can be chained to form queueing networks where jobs departing from one station enter the next one after being served. The difficulty with queueing systems is that their response times develop in a non-linear way, and it is not easy to predict, which station in a network of interleaved processes will first become a bottleneck given a certain demand mix.

There are basically three approaches to support performance prediction of queueing systems in general: *experiments* in the lab, *analytical models*, and *simulation*. Experiments in such an environment are problematic. First, in 24/7 operations experiments cannot be done on a production system. Therefore, companies need to set up a lab infrastructure to perform overload tests, which is very costly. Second, a single experiment can only provide results for a particular demand mix of processes. Since every experiment is costly, it is also expensive to perform sensitivity analyses with a larger set of demand scenarios. In comparison, analytical models and simulation are cheaper to set up provided that there is tool support available including log file analysis, analytic solvers, and a simulation engine. Once a valid model is found, it is easy to do various forms of sensitivity analyses. There is surprisingly little empirical work on the usage of analytical models or simulation for the capacity planning of DTP systems [2-5]. Also, knowledge about and usage of these techniques is limited or not existent at all by many IT service providers.

Queueing theory has been a main area of research in Computer Science and Operations Research in particular in the 70s and 80s. However, most published applications up until now focus on rather small systems, such as single computer configurations or isolated three-tier web applications. While business process

modelling and automation has been an important research thread in Information Systems in the recent years [6, 7], surprisingly little has been published on performance modelling of automated business processes.

Although, the results of analytical queueing network models for smaller applications in lab environments with synthetic workloads are promising [8], it is not clear that robust results can be achieved in large-scale DTP systems in the field. First, arrival distributions do not typically meet the assumptions of analytical models. Second, there are a number of technical features of DTP systems (e.g., rollbacks or additional management overhead) that are difficult to describe in a queueing network model. Finally, the sheer size of these systems can become a problem. Although, some assumptions are not fulfilled, such models might still serve as a useful approximation of the real world. So far, there is little empirical evidence in the literature that queueing network models or simulation can provide reliable results for large-scale DTP systems. Given the importance of DTP systems and automated business processes in practice and in the academic literature, we believe that performance modelling deserves more attention.

In this paper, we evaluate the predictive accuracy of different types of analytical queueing network models. We will compare the results to those of a discrete event simulation and the actual response times in log data. We will also show prerequisites for performance prediction in this field and discuss problems that limit the applicability of analytical models.

Experimental results for these questions need to be based on real-world data. Therefore, we have analyzed log data of two large DTP systems of a European telecom provider. Based on an analysis of log data, we have derived an estimator for arrival rates and service times. As this data is readily available for most productive DTP systems, we did not have to rely on additional measurements. We have parameterized different types of analytical models and simulations to get predictions for response times and throughput, which we could then compare with actual performance indices. The effort for the experimental setup is significant, as it requires not only the implementation of respective queueing network solvers and simulations but also a comprehensive tool to parse, analyze, and filter huge

amounts of log data with different syntax. The daily volume of log data for both systems that we analyzed is between 400 and 900 MByte, and more than 20 GByte per month.

In the next section, we will discuss related literature. Section 3 will briefly describe essential characteristics of DTP systems, and Section 4 will cover appropriate performance modelling techniques. Section 5 will provide an overview of the data and the system configuration of the systems in question. In Section 6 we will describe the experimental setup, and in Section 7 the results, before we conclude in Section 8.

2. Related Literature

Performance modelling has long been an issue in Computer Science and Operations Research [9-11]. Target areas of performance analysis included file and memory systems, databases, computer networks, operating systems, fault-tolerant systems, and real-time systems [12, 13]. In contrast, in our work, we focus on the performance prediction for large-scale DTP systems. Whereas many papers on performance prediction are based on synthetic workloads or lab settings [2, 3], we have analyzed DTP implementations in the field.

Benchmark tests are regularly used in practice for capacity planning and bottleneck detection of computer systems [14]. Several popular benchmarks exist. The ones most suitable for DTP systems are the SAP Standard Application Benchmarks [15], the SPEC JBB 2005 [16], the Oracle Applications Standard Benchmark [17], and TPC-E [18], replacing the popular TPC-C benchmark [19]. While benchmark tests deserve their place, we argue that respective experiments should be complemented by performance models. IT service managers need to be able to analyze the impact of variations in the demand mix. Benchmark tests are typically designed to describe the capacity of hardware configurations. They do not vary the demand mix and will only provide valid results for a particular demand mix scenario.

Published applications of *queueing network models* (QNMs) to distributed systems that we know of are restricted to rather small applications, such as three-

tier web services [4, 5]. Urgaonka et al. have recently applied QNMs for predicting the performance of multi-tier internet services [20]. The systems were smaller and they could apply an exact *mean value analysis* (MVA) algorithm to solve closed QNMs with good predictive accuracy. We solve large performance models analytically by applying several approximate algorithms for open and closed queueing networks. Approximate algorithms are necessary due to the size of the queueing network models and the underlying systems, which cannot be solved exactly any more. In addition, we develop a *discrete event simulation* (DES) engine to compare the results. DES can be of interest, for example, when the stationarity assumption of product-form QNMs regarding arrival rate distributions is violated. The characteristics of real-world demand distributions have been discussed in a number of recent publications [21-23].

3. Distributed Transaction Processing

Typical DTP systems are structured as distributed applications, with services running on different processors or in different processes. For instance, a multi-tiered system might start with requests initiated in a Web portal, forwarded to a TP monitor, which in turn calls several applications or services (databases, ERP modules, name service, credit-card authorization, etc.). In this paper, we will call such automated business processes that are executed without human interaction *workflows*. Typically, several workflows share the same services. For example, many of them might call the order entry or the CRM system. DTP systems ensure transactional integrity by implementing a distributed two-phase commit protocol.

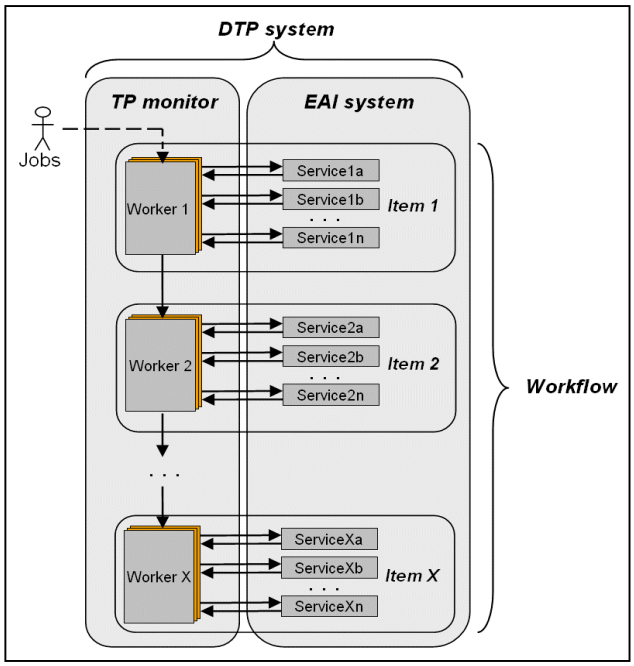

Fig. 1: Typical structure of a DTP system

Figure 1 shows the structure of a large-scale DTP system: a workflow is composed of multiple transactions (aka *items*) calling a number of basic *services* sequentially or in parallel. Multiple concurrent instances of items exist, sharing an item queue. In addition, items can be part of multiple workflows creating a complex network of interwoven business processes.

Services provide basic business functionality of a backend system, or a composite service that call others to implement new service functionality [24]. In most DTP systems one can find internal as well as external services. A problem with the latter ones is that their total workload can often not be observed.

Typical order-entry DTP systems have to deal with data entered directly by the user. Incomplete or erroneous data (e.g., in address data), but also technical failures of the IT infrastructure like hardware defects can cause exceptions. Such exceptions are handled by a TP monitor through *rollbacks* enforcing item level transactional integrity.

4. Performance Modelling Techniques

Performance modelling and prediction is important for capacity planning tasks of IT service providers. The goal of performance modelling is to gain an understanding of a computer system's actual performance and to predict how changes might affect the performance measures of the system in the future [25]. Based on forecasts about future demand, an IT service manager wants to predict response times, and determine bottlenecks, i.e., those servers which need to be upgraded to improve the performance of the overall system.

DTP systems can be modelled as queueing systems with jobs arriving in queues of service stations. The difficulty with queueing systems of this sort is that response times develop in a non-linear way and cannot be predicted by simply extrapolating response times at low demand. When one station in a queueing network becomes overloaded, this can impact all processes and their response times will grow rapidly. The two main approaches to performance modelling of

such queueing systems are *queueing theory* (QT) and *discrete event simulation* (DES).

4.1. Queueing Theory

Queueing theory is an analytical modelling technique for the mathematical analysis of systems with waiting lines and service stations. Queueing network models (QNMs) represent a system as a network of service stations with queues that serve requests of several classes [26]. Applications range from manufacturing system planning and computer processor design to models of multi-tier web services [20, 27-29].

A single service station consists of one or more identical servers with a joint waiting room. Jobs arrive at the queue with an arrival rate λ and have an expected service time $E(S)$. If the servers are all occupied, jobs have to line up in the queue. The so-called Kendall notation [30] is often used to classify different types of service stations: A / B / C (where A stands for the distribution of inter-arrival times of customers, B for the distribution of service times, and C for the number of service stations). A and B typically take the following distributions types: M (Exponential / Markovian Distribution) or G (General / Arbitrary Distribution).

A QNM consists of a number of interconnected service stations. Depending on their characteristics and of the workload (number/type of jobs), several exact and approximate solution techniques exist. A solution consists of response times for jobs, throughput rates, the lengths of waiting lines, and the utilization of service stations. Parameters such as the service time of jobs in a computer system are often not readily available, which is one reason why the technique is rarely used for the capacity planning of IT systems.

Queueing networks can be classified into three categories: open, closed, and mixed queueing networks. Open queueing networks have an external input and an external final destination. In closed queueing networks the customers circulate continually never leaving the network. Mixed queueing networks combine open and closed QNs. If all service stations in the network fulfil certain assumptions concerning the distribution of inter-arrival rates and service times and the queueing discipline, each

single queueing system can be examined on its own, in isolation from the rest of the network. Networks fulfilling these conditions are referred to as separable or product-form networks.

The most famous result concerning product-form queueing networks was presented by Baskett, Chandy, Muntz and Palacios in [26] known as BCMP theorem. It defines the well-known class of BCMP queueing networks with product-form solution for open, closed or mixed models with multiple classes of customers and various service disciplines and service time distributions. The stationary state distribution is expressed as the product of the distributions of the single queues with appropriate parameters and, for closed networks, with normalization constant. Stationarity means that the mean, variance and autocorrelation structure of a stochastic process do not change over time. This assumes flat looking series, without trend, with constant variance over time, a constant autocorrelation structure over time and no periodic fluctuations

We can analyze product-form networks with various computational algorithms to evaluate the performance indices. For the computation of closed QNs we apply either the exact Mean Value Analysis (MVA) [31] or, for networks with a large numbers of users and multiple job classes, the Self Correcting Approximation Technique (SCAT) [32]. These algorithms provide the evaluation of average performance indices with a polynomial space and time computational complexity in the network dimension that is the number of service stations and the population. In our work we use algorithms for open M/M/c queues, as well as algorithms for closed QNMs belonging to the BCMP family. We refer the interested reader to [29] for a more detailed description of various QNM algorithms.

The assumptions of product-form QNMs are restrictive and often not met to the full extent. For example, QNMs produce steady state performance metrics, while demand of most information systems is volatile throughout the day. Nevertheless, experts in other areas regularly use their predictions as approximations [28].

4.2. Discrete Event Simulation

In order to provide insight into problems which do not fall under the mathematical realm of queueing theory, alternative means of analysis have been devised, most notably *discrete event simulations* (DES) [33]. DES deals with the modelling of a stochastic system as it evolves over time in which the system state changes only at discrete points of time [34]. In a DES the operation of a system is represented as a chronological sequence of events. Each event occurs at an instant in time and marks a change of state in the system.

The advantage of simulation is that the performance analyst is not forced to make many assumptions as required for analytical solutions allowing for the consideration of more detailed network models of a wider variety of systems. However, it is typically more time-consuming and costly to implement a simulation of the system under study that is exact enough to allow for significant performance analysis. The closer the simulation should model the characteristics of large-scale IT infrastructures, the more effort it takes to develop a simulation. Thus, one has to carefully determine the scope and the level of detail.

The DES was executed on a custom developed simulation, especially designed to fit the characteristics of DTP systems such as multiple workflows, nested service call structures, and different station types. We developed a simulation engine as part of our open source framework PerMoTo (Performance Modelling Tool) specially designed for the evaluation of DTP systems [35]. Parts of the simulation engine are based on JSIM, an open source simulation engine of the Java Modelling Tools framework (JMT [36]). PerMoTo allows for the computation of several performance measures of multi-class queueing network models including response time, queueing time, queue length, utilization and throughput.

The simulation engine is based on a discrete event list. Each event, like a job entering a station or the departure of a job after service completion, is represented as an entry in this list. The list acts as a message broker, dispatching messages to the related simulation nodes. Each node has three main sections: an input section, a service section and an output section. The input section is responsible for receiving incoming jobs; storing them in a queueing buffer and releasing them

from the queue by realizing a certain queueing discipline such as FCFS (First Come First Served) or LCFS (Last Come First Served). The service section simulates the service execution on the node. The time needed to process the job on this station is specified by the parameters of a service time distribution.

As soon as a simulation is started, a statistical analyzer logs various performance measures. The results are analyzed using transient detection and confidence interval estimation algorithms. These calculations are periodically run during the simulation execution and as soon as the requirements are met, the simulation is stopped.

The transient detection is implemented using the R5 heuristic [37] and the MSER-5 stationarity rule [38]. When a steady-state is assumed, all collected samples are cleared and the statistical analyzer proceeds with the calculation of the confidence interval estimation using spectral methods [39]. For our experiments we imposed the simulations to stop after the half-width of the estimated 95 % confidence interval is no more than 10 % of the non-transient sample mean. After matching these criteria, the calculated performance measures are stored to the result database and the simulation is finished. In the experiments presented in this paper we assumed the same product-form assumptions and model parameters as for the analytic solution techniques.

4.3. Performance Prediction Example

Performance models are used by IT service managers to predict response times and bottlenecks in the system. We have integrated a number of QN algorithms as well as our simulation engine in *PerMoTo* to carry out these tasks [35]. Here, we will provide a few sample predictions to illustrate typical use cases in IT service management.

Figure 2 shows the development of end-to-end workflow response time when increasing the demand on the eight most frequent workflows of System A up to 4.5 times from a baseline workload. Two workflows (*WF A2, WF A7*) are taking significantly more time due to queueing effects on some shared items while the response times of the other workflows do not change significantly.

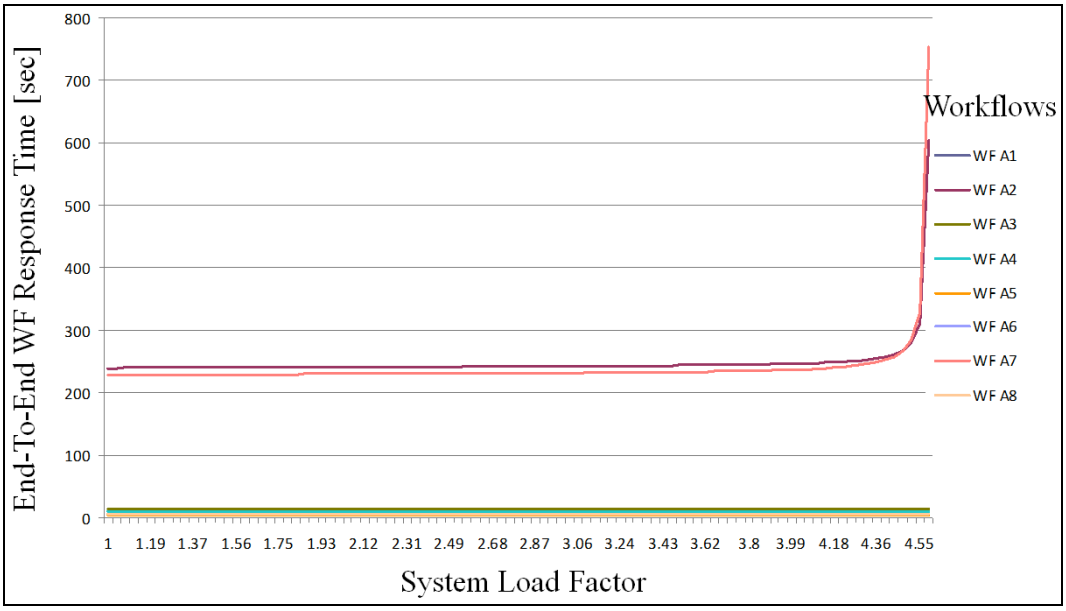

Fig. 2: Capacity planning scenario: Development of end-to-end workflow response times while increasing the overall system load up to 4.58 times of the initial load

Figure 3 depicts the utilization of all 39 items of DTP system A when increasing the demand on each workflow up to 4.58 times. Item 40 shows the highest utilization. Further drill down into the performance measures on the various services would identify the service station that evolves as bottleneck of the DTP system given this specific demand mix.

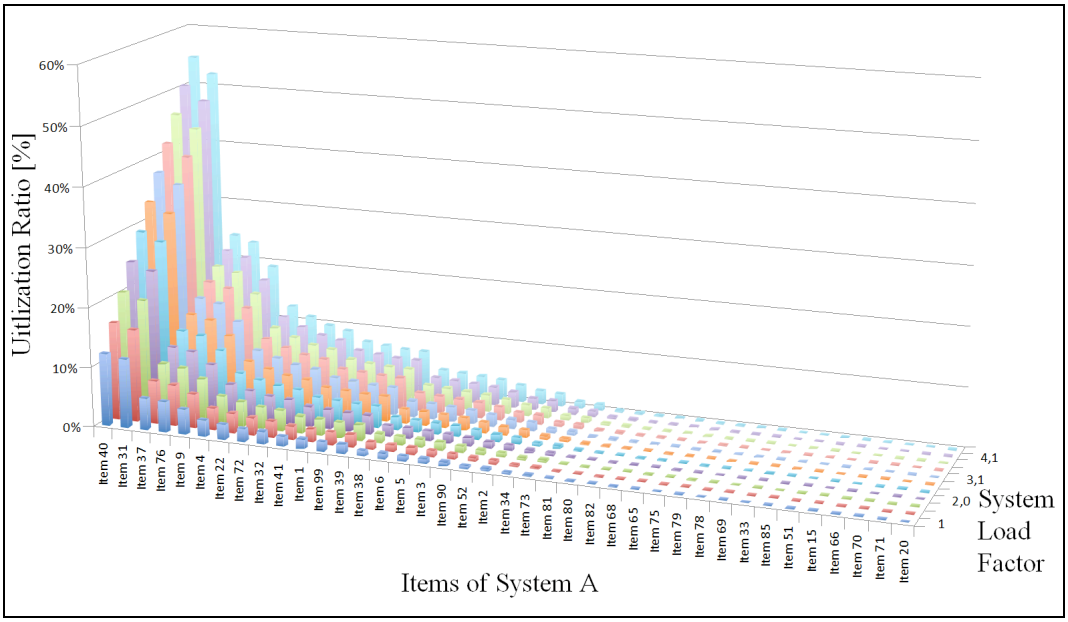

Fig. 3: Bottleneck detection scenario: Development of item utilizations of system A when increasing the overall load up to 4.58 times of the initial value (prediction of Open QNM)

The demand mix of a real-world DTP system typically changes over time. In the scenario below, we ran the same scenario with a different mix, i.e., a higher overall demand for the order entry process. This means, we performed sensitivity analyses with respect to the demand mix. In this scenario, the item with the highest utilization was item 9 (see Figure 4) instead of item 40 (see Figure 3).

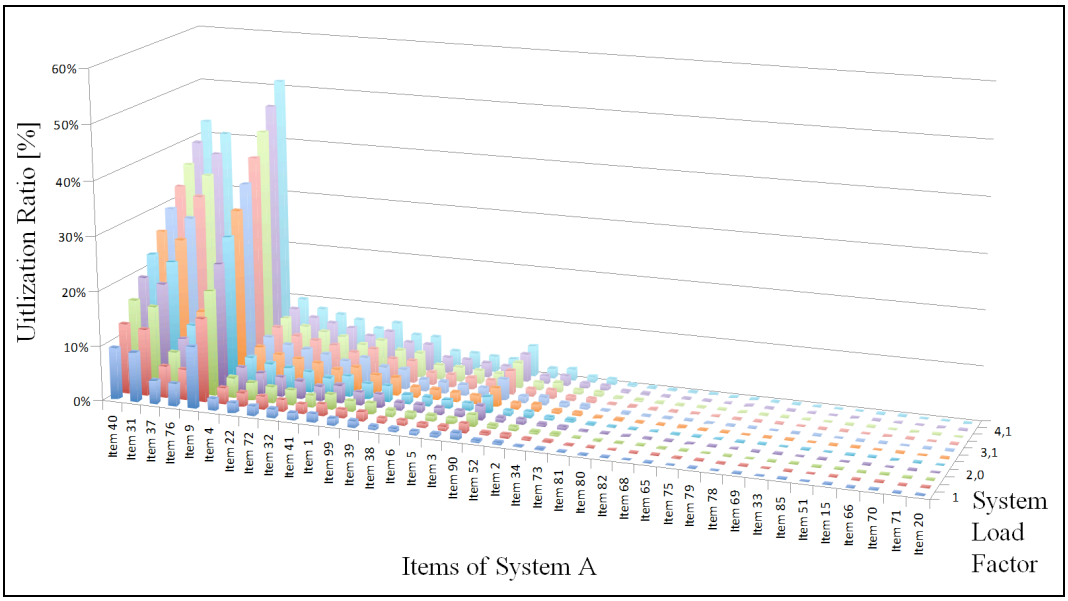

Fig. 4: Bottleneck detection scenario like in Figure 3 but with changed demand mix, assuming an advertisement campaign; new bottleneck candidate is Item 9

These types of sensitivity analyses are important for an IT service manager to understand different demand scenarios and their impact on the utilization of items, services, and response times.

5. Data and DTP System Configuration

In this paper, we evaluate the predictive performance of analytical queueing network models and simulations for DTP systems. We like to explore the problems in the application of QNMs to large-scale DTP systems. We have therefore looked at two productive DTP systems of a European telecom provider.

System A is the central IT backbone for workflows related to the management of the retail customer segment including billing, customer data acquisition, network provisioning, and phone number management. The technical implementation is based on the Transaction Monitor "Bea Tuxedo" ™. Requests on System A are

initiated in the point of sales systems of our industry partner: internet portals, shop-based applications, and call centres. Billing and maintenance workflows are initiated by other internal IT systems.

System B is the integration backbone for workflows related to the management of products hosted by our industry partner but originally sold as prepaid telecommunication or DSL packages by external third-party companies. The supported functionality includes tariff management, customer subscription and deactivation, SIM and phone number management, billing, and age verification. System B serves two main classes of workflows: order-entry workflows initiated by customers over a Voice Portal (*ICR* - Interactive Voice Response System), and workflows used by internal IT systems of the partner companies like billing or tariff administration. Technically, B is based on a customized version of "Tibco Business Works" ™.

Details of the system configurations concerning its size and structure are summarized in Table 1: System A serves 18 business critical workflows. The variable tasks of the workflows are achieved by 39 different item types, calling 51 services. The length of the workflows varies: a single one is very short as it consists of only one item step; the others are more complex and contain up to 17 items. A maximum of 33 different service types are called inside a single workflow, and up to seven service types are called within a single item. The length of the items varies as well – one contains only a single service step, the others call several services, up to a maximum of 12. Single item types are typically called by more than one workflow (up to three); single service types are included in up to 25 different items and a maximum of 14 different workflows.

Unlike in system A, the audit logs of the second DTP system under study contain only information on workflow and item level. Thus, we modelled the queueing networks of B on item level granularity. System B serves 15 workflows containing 35 different item types. Several workflows of system B call just one item type while others are more complex. A maximum of 19 items are called within a single workflow, belonging to a maximum of 17 different types. Most

items are called in various workflows, one item type actually by up to seven different ones.

System Characteristics	System A	System B
	Amount / Min-Max	*Amount / Min-Max*
Overall number of Workflow types in system	18	15
Overall number of Item types in system	39	35
Overall number of Service types in system	51	-
Overall Item calls in system	53	90
Overall Service calls in system	156	-
Number of Item types in single Workflow	1 - 17	1 - 17
Number of Service types in single Item	1 - 7	-
Number of Service types in single Workflow	1 - 33	-
Number of Item steps in single Workflow	1 - 17	1 - 19
Number of Service steps in single Item	1 - 12	-
Number of Service steps in single Workflow	1 - 65	-
Number of Workflow types calling single Item type	1 - 3	1 - 7
Number of Item types calling single Service type	1 - 25	-
Number of Workflow types calling singe Service type	1 - 14	-

Table 1: System characteristics of the real-world DTP systems under study

Both DTP systems have a release cycle of three months allowing the company to react to changes in the business demand. Whereas the introduction of new workflows is relatively rare (e.g., the introduction of a new product), changes in the services are part of every new release, adding functionality or fixing errors. Such changes can require new estimates for service times.

In order to estimate model parameters, we have analyzed a large volume of log data that is described as follows. The log data is written in daily log files containing up to seven million raw text lines (400 to 900 MBytes per day). We have analyzed log data of both systems of nine weeks in summer 2008. Timestamps and request IDs allow for estimating the input parameters that are relevant for QNMs and DES, most notably arrival rates on a workflow level and service times on a service level, as well as end-to-end response times.

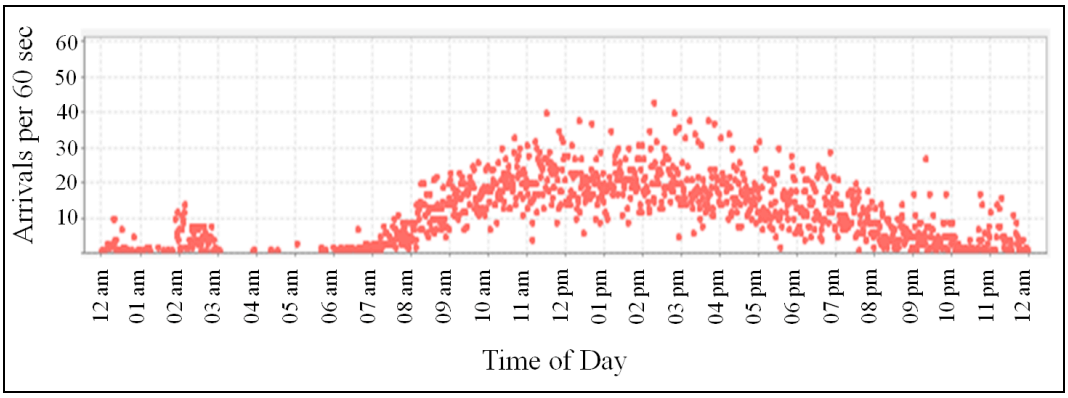

Fig. 5: Sample of the daily demand on a workflow of system A that exhibits a characteristic pattern of both DTP systems on weekdays

The demand mix describes the number of requests for the different workflows. Figure 5 shows a daily workload sample that is typical for both DTP systems on weekdays.

The daytime workloads are generated by shop employees, customers using the web portal, and call centre agents. End-to-end response times of most workflows are the primary metric in SLAs as response time is recognized directly by the customers and other end users. Weekdays exhibit an overall workload that is much higher than the one on weekends (up to 5 times more requests).

As Figure 5 shows, the daytime workload exhibits a certain usage pattern: increasing overall workload in the morning, a plateau phase throughout the day, followed by a decreasing workload in the evening hours. Figure 6 shows the daily demand mix on the workflows of systems A and B. The total demand for the eight most frequent workflows A1 to A8 of System A makes up 98.3% of the overall workload of this system. Similarly, the sum of shares of the workflows B1 to B8 make up already 99.7% of the overall workload of System B.

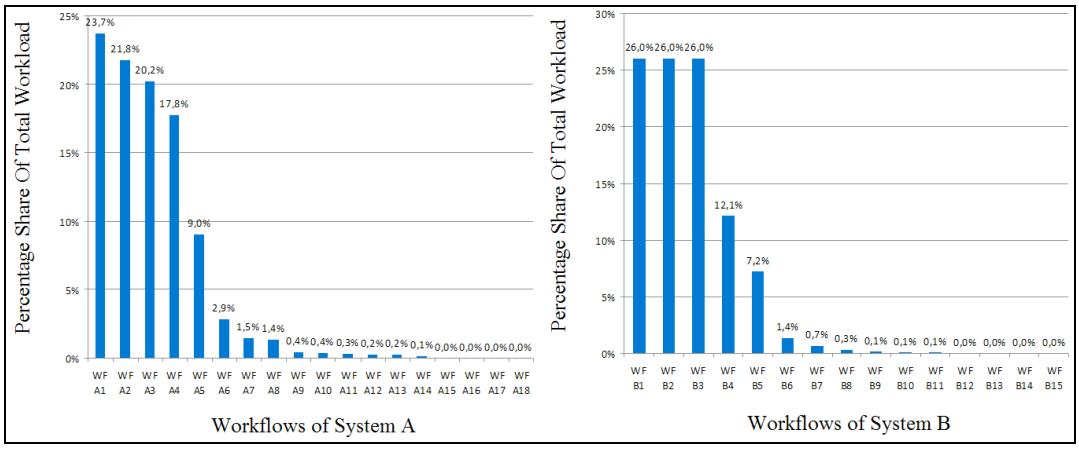

Fig. 6: Typical demand mix of systems A and B in percentage shares per day

There are seasons with significantly higher workloads throughout the day, such as the time before Christmas. These are the top-selling weeks of the year for mobile phone providers. For example, the total average workload per week increases by 25.4% in December 2007 compared to that of September. Similar increases can be a consequence of new campaigns or the introduction of new products. Figure 7 depicts the workload changes from our base period used to parameterize our QNMs (40 weekdays in June and July 2008) to our prediction period (5 weekdays in July and August 2008). The number of requests on the frequently called workflows in September increases even further. Altogether, the demand ratios on System A increased by 9.8%, on System B even by 31.5% when comparing the prediction period to the base period. For an IT manager, it is important to know, what the response times will be if there is a 20%, 50% or even an 80% increase in the demand for some workflows, and to perform respective sensitivity analyses.

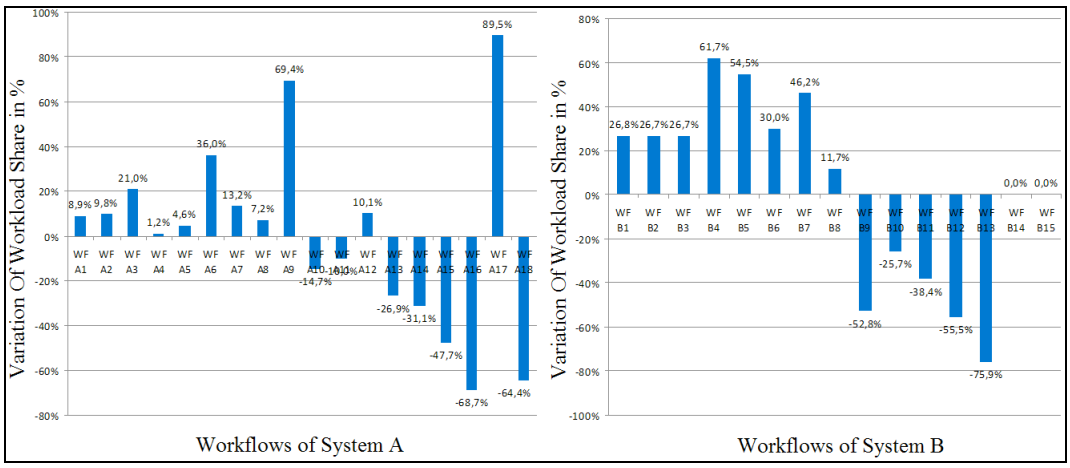

Fig. 7: Changes of the daily demand for workflows of systems A and B between the base and the prediction period in summer 2008

6. Experimental Setup

In our experiments, we predict response times for particular days using different types of queueing network models and a discrete event simulation for the two different DTP systems A and B (using TP monitors of different vendors) described in Table 1.

There are a number of possibilities for modelling multi-class QNMs. First, the level of granularity might differ. We have modelled the DTP System A on a *service level* and on an *item level*, System B only on an item level since service level log data was not available. Second, the DTP systems are modelled as *open* and *closed QNMs* to allow for the analysis of the performance of both types of QNMs for large DTP systems. We apply an algorithm developed by Bolch et al. [40] to directly transform the open into closed QNMs. These different models constitute the main treatments of our experimental setup.

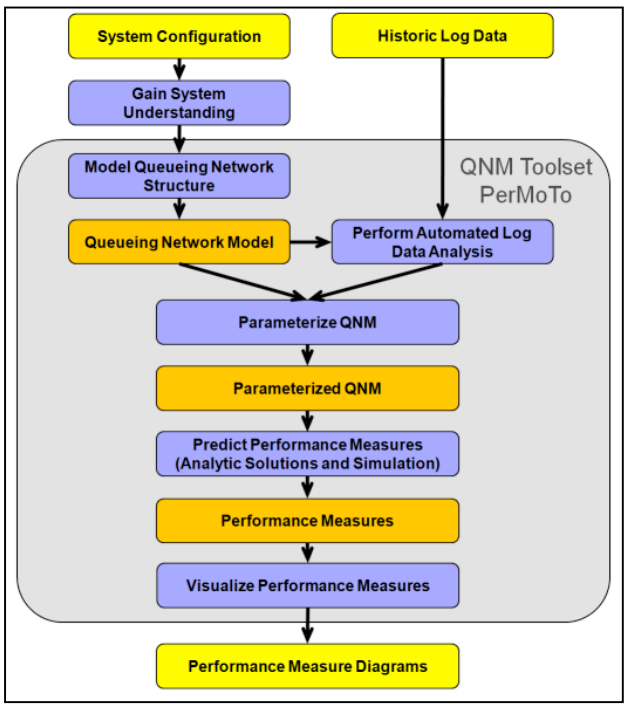

Fig. 8: Performance modelling process

Figure 8 provides an overview of the main steps in the performance prediction experiments: based on information about the system configuration and historic log data, we modelled the structure of the queueing network and estimated the relevant parameters. The models were then solved in the PerMoTo toolset, which allows for the prediction of response times, queueing times, throughput, and utilization.

The PerMoTo tool includes a log analyzer that is used to parameterize the QNM. This includes the determination of appropriate estimators for the workflow arrival rates and service times, i.e., the fraction of time needed to execute a single service

step of a workflow (see Figure 1). With appropriate tooling even the analysis of large log files becomes viable. Actually, the results of such an analysis (demand mix, arrival rates, service times) are of interest in itself to IT service managers, even without a performance prediction, as they reveal arrival rates and demand mix that are often unknown. The use of log data is non-intrusive and leverages data that is already available when using DTP systems. In contrast, some software vendors provide specific tools that monitor resource consumption of individual applications on certain servers.

The performance measures can then be calculated using analytic solvers or a discrete event simulation. Finally, the performance measures can be visualized. The entire PerMoTo toolset can be downloaded as open source at: http://ibis.in.tum.de/research/itsom/itm08/.

In Table 2, we provide details of the models of two business critical DTP systems A and B of our industry partner. They can both be considered large-scale DTP systems, with more than 15 classes and more than 100 stations.

Queueing Network Model Characteristics	**System A**	**System B**
	Amount / Min-Max	*Amount / Min-Max*
Number of classes (i.e., workflows)	18	15
Number of stations (i.e., services)	140	126
Number of parallel workers in station	1 - 8	1 - 6
Mean response time on a service level [sec]	0.17	-
Min - Max response time on a service level [sec]	0.01 - 4.41	-
Mean response time on item level [sec]	0.68	0.23
Min - Max response time on item level [sec]	0.01 - 5.46	0.051 - 2.53
Mean response time on workflow level [sec]	37.81	5.13
Min - Max response time on workflow level [sec]	4.41 - 248.26	0.01 - 36.06

Table 2: Key characteristics of the queueing network models of systems A and B

A full description of all models for these systems in XML format can be downloaded from http://ibis.in.tum.de/research/itsom/itm08/.

For the parameterization of systems A and B, we used log data of 40 weekdays in June and July 2008. We chose the last week of July as the prediction period as this week had a higher than average demand on both systems due to a marketing

campaign. Thus, the average arrival rates of our forecast models are estimated based on the five weekdays of this week. The demand mix of systems A and B vary throughout the day, but reach a plateau between 11 am and 6 pm (see Figure 5). While stationarity of the time series is an idealized assumption, the results of our QNMs can still be used as an estimate for response time and utilization for these time frames during the day.

7. Results

We focus on throughput and response time as primary metrics of interest for end-to-end quality-of-service. Table 3 summarizes the mean throughput (number of workflow calls per second) of the system during the prediction period and the predictions of open and closed QNMs, as well as the DES. The QNMs provide a very high predictive accuracy if compared to the mean throughput of the prediction period, higher than that of DES.

System	Workflow	*Mean [#/sec]*	*Open QNM [#/sec]*	*Closed QNM [#/sec]*	*DES [#/sec]*
A	WF A1	0.340	0.340	0.339	0.342
A	WF A2	0.312	0.312	0.306	0.314
A	WF A3	0.290	0.290	0.287	0.298
A	WF A4	0.255	0.255	0.254	0.262
A	WF A5	0.130	0.130	0.130	0.126
A	WF A6	0.041	0.041	0.041	0.042
A	WF A7	0.021	0.021	0.021	0.021
A	WF A8	0.020	0.020	0.020	0.019
B	WF B1	0.139	0.139	0.139	0.141
B	WF B2	0.139	0.139	0.139	0.141
B	WF B3	0.139	0.139	0.139	0.142
B	WF B4	0.065	0.065	0.065	0.065
B	WF B5	0.039	0.039	0.039	0.039
B	WF B6	0.007	0.007	0.007	0.008
B	WF B7	0.004	0.004	0.004	0.004
B	WF B8	0.002	0.002	0.002	0.002

Table 3: Mean throughputs and predictions of the three models

Response times are more difficult to predict as they exhibit a high variance due to various latencies in DTP systems and in computer systems in general. Table 4

provides an overview of the median, the mean, the coefficient of variation (CV) of the empirical response times, the predictions, and the respective root mean squared error (RMSE).

System	Workflow	Real-World Response Time				Prediction Open QNM		Prediction Closed QNM		Prediction DES	
		Median [sec]	*Mean [sec]*	*CV*	*RMSE [sec]*	*Prediction [sec]*	*RMSE [sec]*	*Prediction [sec]*	*RMSE [sec]*	*Prediction [sec]*	*RMSE [sec]*
A	WF A1	9.688	9.926	0.421	3.994	9.353	4.007	9.329	4.010	9.382	4.004
A	WF A2	237.656	248.276	0.346	76.685	239.853	75.537	238.629	75.583	238.752	75.579
A	WF A3	14.936	14.769	0.233	3.212	14.879	3.208	14.878	3.208	14.880	3.208
A	WF A4	10.552	11.213	0.446	4.621	10.147	4.680	10.147	4.680	10.147	4.680
A	WF A5	4.996	5.091	0.595	3.017	4.945	3.019	4.945	3.019	4.946	3.019
A	WF A6	6.146	6.192	0.522	3.125	5.392	3.162	5.375	3.166	5.380	3.164
A	WF A7	220.078	240.215	0.619	168.781	228.281	169.367	226.639	169.968	226.656	169.961
A	WF A8	7.840	8.278	0.553	3.767	7.728	3.745	7.728	3.745	7.728	3.745
B	WF B1	0.264	0.281	0.197	0.047	0.258	0.050	0.258	0.050	0.257	0.050
B	WF B2	0.256	0.276	0.217	0.051	0.255	0.053	0.255	0.053	0.259	0.052
B	WF B3	0.460	0.482	0.203	0.087	0.439	0.094	0.438	0.094	0.441	0.093
B	WF B4	11.860	12.512	0.225	2.479	13.365	2.719	13.329	2.704	13.412	2.738
B	WF B5	0.016	0.018	0.247	0.003	0.019	0.004	0.019	0.004	0.019	0.004
B	WF B6	5.692	6.186	0.300	1.627	5.278	1.773	5.267	1.778	5.274	1.775
B	WF B7	0.016	0.017	0.244	0.003	0.018	0.004	0.018	0.004	0.018	0.004
B	WF B8	0.016	0.018	0.235	0.003	0.019	0.004	0.019	0.004	0.019	0.004

Table 4: Real-world workflow response times and the predictions of Open QNM, Closed QNM and DES solution methods with corresponding RMSE

The variance in the response time leads to the fact that the RMSE is high and difficult to interpret. As an alternative measure, one might therefore be interested in the difference of the predictions from the median of the response times observed during the prediction period. Table 5 and Figure 9 provide an overview of these deviations in percentage of the median.

System	Workflow	Real-World Resp. Time	Open QNM	Closed QNM	DES
		Median [sec]	*Error [%]*	*Error [%]*	*Error [%]*
A	WF A1	9.688	- 3.457	- 3.709	- 3.157
A	WF A2	237.656	0.924	0.409	0.461

A	WF A3	14.936	- 0.383	- 0.385	- 0.376
A	WF A4	10.552	- 3.841	- 3.841	- 3.842
A	WF A5	4.996	- 1.022	- 1.022	- 0.994
A	WF A6	6.146	- 11.296	- 11.571	- 11.481
A	WF A7	220.078	3.727	2.981	2.989
A	WF A8	7.840	- 1.435	- 1.434	- 1.434
B	WF B1	0.264	- 2.143	- 2.185	- 2.558
B	WF B2	0.256	- 0.484	- 0.526	1.021
B	WF B3	0.460	- 4.655	- 4.683	- 4.053
B	WF B4	11.860	12.691	12.388	13.086
B	WF B5	0.016	17.049	17.049	17.702
B	WF B6	5.692	- 7.268	- 7.465	- 7.343
B	WF B7	0.016	15.371	15.371	14.550
B	WF B8	0.016	16.907	16.907	17.462

Table 5: The medians of the real-world response times and respective errors of Open QNM, Closed QNM and DES

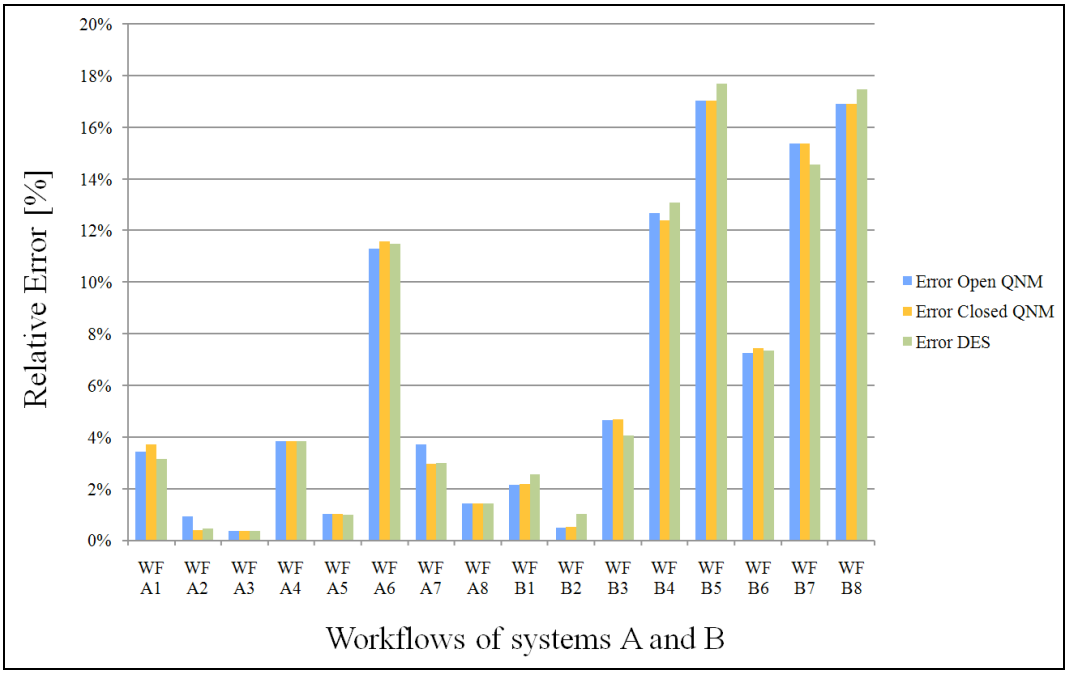

Fig. 9: Relative error of the predictions of Open QNM, Closed QNM and DES to the medians of the real-world workflow response times of systems A and B

While closed QNMs often have a higher predictive accuracy than open QNMs [27], there was no significant difference in our predictions. This might be explained by the fact, that the systems were oversized and not fully utilized during

the prediction period. In general, the differences can partly be explained by the variance in the data, which makes it much harder to predict than throughput.
Those workflows that had more than 10% difference were also the ones accessing external systems that could not be fully observed.

Overall, the predictions of our DES were not significantly better than those of the Open and Closed QNMs. This is due to the fact that the basic QN model and the parameters were the same, and we used the same estimates for arrival rates and service times. In such situations, DES does not have a clear advantage over analytical solutions. Only, when there are systems specifics that cannot be modelled as product-form QN, DES would have advantages. For example, if the arrival rates were very volatile during the day and stationarity assumption was heavily violated, DES would provide a better prediction. It is also possible to analyze state-dependent routing policies in workflows or different priority-based scheduling strategies as they can be used in DTP systems instead of FCFS or LCFS and analyze their impact on the different performance metrics. This was actually one of the main reasons for implementing the DES.

Solution Method	**System A**		**System B**	
	Single Load Factor	*What-If Analysis*	*Single Load Factor*	*What-If Analysis*
Open QNM	0.095 sec	8.911 sec	0.032 sec	8.211 sec
Closed QNM	2.915 sec	274.236 sec	1.077 sec	102.892 sec
DES	around 8 h	-	around 6 h	-

Table 6: Calculation times of the different solution methods of a single load scenario and a what-if analysis containing 100 load scenarios

This flexibility of DES, however, comes at a cost. Table 6 shows how long it takes to generate predictions for the two DTM systems A and B on a PC with an Intel Core 2 Duo processor, 2 x 2.66 GHz, 4 GB RAM, and MS Vista Business. While open QNMs could be solved in a matter of millisconds, and closed QNMs in seconds, the DES typically ran for hours in order to produce the same performance metrics. For more complex what-if analyses including 100 scenarios of increasing load, the DES would take way too much time, while the QNMs provided results within a few minutes, even for this complex scenario.

Apart from the fact that the arrival rates were not fully stationary and response times exhibited a large variance, we encountered a few additional problems, when modelling DTP systems.

- First, errors and rollbacks occurred in a number of transactions. They can be modelled as probabilistic conditions in the QNMs, but depend heavily on the implementation and the type of workflow. The reasons for errors in workflows need to be analyzed separately, as they can significantly impact the overall performance of the system.
- Second, a common problem in large-scale DTP systems is that they access external systems that cannot be fully observed by the analyst. Sometimes, the response times of these workflows dominate the entire workflow and render accurate predictions impossible.
- Third, some workflows have complex event conditions that sometimes lead to the fact that items or services are skipped or alternative services are called. Of course, one could model a single class as multiple and estimate their respective arrival rates. However, this information is often very difficult to extract from the log files, plus it makes the interpretation of performance metrics for many different classes of a particular workflow much more complex for the analyst.

8. Conclusions

Automated business processes running on DTP systems are of fundamental importance for nowadays services industries. IT managers need to meet pre-defined quality-of-service metrics for these systems. Therefore, performance prediction and sizing have become essential.

Queueing Theory and DES have been used in the past to develop performance models for queueing systems. Applications that are discussed in the literature were, however, largely restricted to fairly small IT systems. In this paper, we analyzed the predictive performance of open and closed QNMs, and DES. Overall, the predictions based on estimates of arrival rates and service times were close to the mean throughput and response times that we found in the log data. The relative error of the predictions of Open QNM, Closed QNM and DES to the

medians of the real-world workflow response times of systems A and B was between 0.38 % and a maximum of 17.7 %.

Given the fact that our analysis was based on two productive DTP systems in the field, some caveats apply. As the infrastructure is very expensive, it cannot easily be replicated in a lab environment. While we were fortunate to get access to log data and system specifications of these mission critical systems, we could not run experiments with overload scenarios. We did, however, evaluate such situations in earlier work in smaller laboratory settings and found the predictions of QNMs to be very accurate [8].

Overall, QNMs are very helpful for IT managers to predict throughput and response times. DES are much more expensive from a computational point of view and also very costly to build. They are, however, useful to analyze questions that are beyond the theory of product-form queueing networks, such as special scheduling or routing strategies applied in real-world DTP systems.

Given the importance of end-to-end quality-of-service and the fact that much of the theory and algorithmic solutions to queueing network models have been developed in the 70s and 80s, it is unclear why these tools are hardly being used in practice. One explanation is that IT managers need to have a good understanding of service times and arrival rates in their system. Often, this knowledge is not available at a sufficient level of detail to set the model parameters. In our analysis, we could show that it is not necessary to set up an expensive metering and monitoring infrastructure. Standard log data of DTP systems is typically rich enough to get parameter estimates with sufficient quality that allows for useful predictions and sensitivity analyses.

In our future work we want to investigate performance tuning and software-based capacity adaption based on the prioritization of workflows. Simulations should be used to analyze how well these strategies help to cope with volatile demand. Another set of questions arises in virtualized data centres where capacity of virtual machines can be adapted on the level of the virtual machine manager. Also here, we are interested in the adaptivity of such strategies to changes in the arrival rates.

Acknowledgments

We thank our industry partner for their helpful support and the data sets they provided. This article is based upon work supported by the Bavarian Science Foundation (Bayerische Forschungsstiftung) under grant number AZ-715-06.

References

[1] IDC, *Solutions for the Datacenter's Thermal Challenges*. 2007, whitepaper.

[2] Qin, M., et al., *Automatic generation of performance models for distributed application systems*, in *Proceedings of the 1996 conference of the Centre for Advanced Studies on Collaborative research*. 1996, IBM Press: Toronto, Ontario, Canada.

[3] Rukma Prabhu, V. and A. Varsha, *A methodology and tool for performance analysis of distributed server systems*, in *Proceedings of the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China.

[4] Chen, Y., et al. *SLA Decomposition: Translating Service Level Objectives to System Level Thresholds*. in *ICAC '07: Proceedings of the Fourth International Conference on Autonomic Computing*. 2007. Washington, DC, USA: IEEE Computer Society.

[5] Liu, X., J. Heo, and L. Sha. *Modeling 3-Tiered Web Applications*. in *MASCOTS '05: Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*. 2005. Washington, DC, USA: IEEE Computer Society.

[6] Ouyang, C., et al., *From Business Process Models to Process-oriented Software Systems.* ACM Transactions on Software Engineering and Methodology (TOSEM), 2008.

[7] Dehnert, J. and W.M.P. van der Aalst, *Bridging The Gap Between Business Models And Workflow Specifications.* Int. J. Cooperative Inf. Syst., 2004. **13**(3): p. 289-332.

[8] Brandl, R., M. Bichler, and M. Ströbel, *Cost Accounting for Shared IT Infrastructures: Estimating Resource Utilization in Distributed IT Infrastructures.* Wirtschaftsinformatik, 2007. **49**(2): p. 83-94.

[9] Fitzsimmons, J. and M. Fitzsimmons, *Service Management*. 2004, New York: McGraw-Hill/Irwin.

[10] Neumann, K. and M. Morlock, *Operations Research*. 2002, München: Hanser.

[11] Whinston, W., *Operations Research*. 1994: ITP.

[12] Noel, E. and K.W. Tang. *Performance modeling of multihop network subject to uniform and nonuniform geometric traffic*. in *IEEE/ACM Transactions on Networking (TON)*. 2000.

[13] Thakkar, S.S. and M. Schweiger. *Performance of an OLTP Application on Symmetry Multiprocessor System*. in *17th Annual International Symposium on Computer Architecture*. 1990. Seattle, WA.

[14] Saavedra, R.H. and A.J. Smith, *Analysis of benchmark characteristics and benchmark performance prediction.* ACM Trans. Comput. Syst., 1996. **14**(4): p. 344--384.
[15] SAP. *SAP Standard Applications Benchmarks*. [cited 2009; Benchmark Specification]. Available from: http://www.sap.com/solutions/benchmark/index.epx.
[16] SPEC. *SPECjbb2005*. [cited 2009; Benchmark Specification]. Available from: http://www.spec.org/jbb2005/.
[17] Oracle. *Oracle Applications Standard Benchmark*. [cited 2009; Benchmark Specification]. Available from: http://www.oracle.com/apps_benchmark/.
[18] TPC. *TPC-E OLTP*. [cited 2009; Benchmark Specification]. Available from: http://www.tpc.org/tpce/tpc-e.asp.
[19] TPC. *TPC-C V5*. [cited 2009; Benchmark Specification]. Available from: http://www.tpc.org/tpcc/default.asp.
[20] Urgaonkar, B., et al. *An analytical model for multi-tier internet services and its applications*. in *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. 2005. New York, USA: ACM.
[21] Feitelson, D.G. *Workload Modeling for Performance Evaluation*. in *Performance Evaluation of Complex Systems: Techniques and Tools, Performance 2002, Tutorial Lectures*. 2002. London, UK: Springer-Verlag.
[22] Feitelson, D.G. *The Forgotten Factor: Facts on Performance Evaluation and Its Dependence on Workloads*. in *Euro-Par '02: Proceedings of the 8th International Euro-Par Conference on Parallel Processing*. 2002. London, UK: Springer-Verlag.
[23] Stewart, C., T. Kelly, and A. Zhang. *Exploiting nonstationarity for performance prediction*. in *EuroSys '07: Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*. 2007. New York, NY, USA: ACM.
[24] Josuttis, N., *Soa in Practice: The Art of Distributed System Design*. 2007: O'Reilly Media, Inc.
[25] Bailey, D.H. and A. Snavely. *Performance Modeling: Understanding the Present and Predicting the Future*. in *Euro-Par 2005 Parallel Processing*. 2005. Lisbon, Portugal.
[26] Baskett, F., et al., *Open, Closed, and Mixed Networks of Queues with Different Classes of Customers.* J. ACM, 1975. **22**(2).
[27] Bolch, G., et al., *Queueing Networks and Markov Chains - Modeling and Performance Evaluation with Computer Science Applications*. Second ed. 2006, Hoboken, New Jersey: John Wiley & Sons, Inc.
[28] Kounev, S. and A. Buchmann, *Performance Modeling and Evaluation of Large-Scale J2EE Applications.* Proceedings of the 29th International Conference of the Computer Measurement Group on Resource Management and Performance Evaluation of Enterprise Computing Systems (CMG 2003), Dallas, Texas, USA, December 7-12, 2003, 2003: p. 273-283.
[29] Menasce, D.A., L.W. Dowdy, and V.A.F. Almeida, *Performance by Design: Computer Capacity Planning By Example*. 2004, Upper Saddle River, NJ, USA: Prentice Hall PTR.

[30] Kendall, D.G., *Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain.* The Annals of Mathematical Statistics, 1953. **24**(3): p. 338-354.
[31] Reiser, M. and S.S. Lavenberg, *Mean-Value Analysis of Closed Multichain Queuing Networks.* J. ACM, 1980. **27**(2): p. 313-322.
[32] Neuse, D. and K. Chandy, *SCAT: A heuristic algorithm for queueing network models of computing systems.* SIGMETRICS Perform. Eval. Rev., 1981. **10**(3): p. 59-79.
[33] Fishman, G.S., *Discrete-event simulation*. 2001, London, UK: Springer-Verlag.
[34] Banks, J., et al., *Discrete-Event Simulation*. 2005: Prentice Hall.
[35] Hühn, O. and C. Markl. *PerMoTo - Performance Modelling Tool suite*. in *WITS 07 - Seventeenth Annual Workshop on Information Technologies and Systems*. 2007. Montreal, Canada.
[36] Bertoli, M., G. Casale, and G. Serazzi. *The JMT Simulator for Performance Evaluation of Non-Product-Form Queueing Networks*. in *ANSS '07: Proceedings of the 40th Annual Simulation Symposium*. 2007. Washington, DC, USA: IEEE Computer Society.
[37] Fishman, G.S., *Statistical Analysis for Queueing Simulations.* MANAGEMENT SCIENCE, 1973. **20**(3): p. 363-369.
[38] Spratt, S.C., *Heuristics for the startup problem*. 1998, Department of Systems Engineering, University of Virginia.
[39] Heidelberger, P. and P.D. Welch, *A spectral method for confidence interval generation and run length control in simulations.* Commun. ACM, 1981. **24**(4): p. 233-245.
[40] Bolch, G., H. Jung, and M. Gaebell. *Entwicklung und Validierung der Schließmethode zur Analyse offener Warteschlangennetze*. in *Operations Research Proceedings 1992*. 1993. Heidelberg, Germany: Springer Verlag.