Chapter 17
Resource Demand Estimation

Simon Spinner and Samuel Kounev

As discussed in Chapter 7, resource demands, also referred to as service demands, play a key role in operational analysis and queueing theory. Most generally, the *resource demand* or *service demand* of a unit of work (e.g., request, job, or transaction) at a given resource in a system refers to the average time the respective unit of work spends obtaining service from the resource over all visits to the latter, excluding any waiting times (cf. Chapter 7, Section 7.1.2). Resource demands are normally quantified based on measurements taken on the system under consideration; however, the accurate quantification of resource demands poses many challenges. The resource demand for processing a request in a computing system is influenced by different factors, for example: (1) the application logic, which specifies the sequence of instructions to process a request; (2) the hardware platform, which determines how fast individual instructions are executed; and (3) platform layers (hypervisor, operating system, containers, or middleware systems), which may introduce additional processing overhead. While the direct measurement of resource demands is feasible in some systems, it requires an extensive instrumentation of the application, and it typically introduces significant overheads that may distort measurements. For instance, performance profiling tools (cf. Section 6.3 in Chapter 6) can be used to obtain execution times of individual application functions when processing a request. However, the resulting execution times are not broken down into processing times at individual resources, and profiling tools typically introduce high overheads, influencing the system performance.

In this chapter, we survey, systematize, and evaluate different approaches to the statistical estimation of resource demands based on easy to measure system-level and application-level metrics. We consider resource demands in the context of computing systems; however, the methods we present are also applicable to other types of systems. We focus on generic methods to approximate resource demands without relying on dedicated instrumentation of the application. The goal is to estimate the resource demands based on *indirect measurements* (cf. Section 6.1 in Chapter 6) derived from commonly available metrics (e.g., end-to-end response time or resource utilization).

S. Kounev et al., *Systems Benchmarking*, https://doi.org/10.1007/978-3-030-41705-5_17

The methods we consider face the following challenges:

- The value of a resource demand is platform-specific (i.e., only valid for a specific combination of application, operating system, hardware platform, etc.). The hardware platform determines how fast a piece of code executes in general. Furthermore, each platform layer on top (e.g., hypervisor, operating system, and middleware systems) may add additional overheads, influencing the resource demands of an application.
- Applications often serve a mix of different types of requests (e.g., read or write transactions), which also differ in their resource demands. For resource management purposes, it is beneficial to be able to distinguish between different types of requests. Quantifying resource demands separately for each type of request (i.e., workload class) often poses technical challenges due to the lack of fine-granular monitoring data.
- Modern operating systems can provide only aggregate resource usage statistics on a per-process level. Many applications, especially the ones running in data centers, serve different requests with one or more operating system processes (e.g., HTTP web servers). The operating system is unaware of the requests served by an application and therefore cannot attribute the resource usage to individual requests.
- Many applications allow only the collection of time-aggregated request statistics (e.g., throughput or response time) while they are serving production workloads. A tracing of individual requests is often considered too expensive for a production system, as it may influence the application performance negatively.
- Resource demands may change over time due to platform reconfigurations (e.g., operating system updates) or dynamic changes in the application state (e.g., increasing database size). Therefore, resource demands need to be updated continuously at system run time based on up-to-date measurement data.

In the rest of this chapter, we survey the state of the art in resource demand estimation and provide a systematization of existing estimation methods discussing their pros and cons with respect to how well they deal with the above challenges. The goal of the systematization is to help performance engineers select an estimation method that best fits their specific requirements. We first survey existing estimation methods and describe their modeling assumptions and their underlying statistical techniques. Then, we introduce three dimensions for systematization: (1) input parameters, (2) output metrics, and (3) robustness to anomalies in the input data. For each dimension, we first describe its features and then categorize the estimation methods accordingly. In addition to the systematization, we compare and evaluate the different estimation methods in terms of their accuracy and execution time. The presented systematization and comparison of estimation methods are based on Spinner et al. (2015) and Spinner (2017). Finally, we briefly discuss a recent approach to resource demand estimation that relies on multiple statistical techniques for improved robustness and uses a cross-validation scheme to dynamically select the technique that performs best for the concrete scenario (Spinner, 2017).

In the following, we use a consistent notation for the description of the different approaches to resource demand estimation. We denote resources with the index $i = 1 \ldots I$ and workload classes with the index $c = 1 \ldots C$. The variables used in the description are listed in Table 17.1, which are consistent with the notation we used in Chapter 7 (Section 7.2.2) in the context of queueing networks. As usual, we assume that the considered system is in operational equilibrium (i.e., over a sufficiently long period of time, the number of request completions is approximately equal to the number of request arrivals). As a result, the arrival rate λ_c is assumed to be equal to the throughput X_c. Furthermore, as mentioned earlier, we use the term resource demand as a synonym for service demand, and for simplicity of exposition, we assume $V_{i,c} = 1$; that is, no distinction is made between service demand and service time.

Table 17.1: Notation used in resource demand estimation

Symbol	Meaning
$D_{i,c}$	Average resource demand of requests of workload class c at resource i
$U_{i,c}$	Average utilization of resource i due to requests of workload class c
U_i	Average total utilization of resource i
$\lambda_{i,c}$	Average arrival rate of workload class c at resource i
$X_{i,c}$	Average throughput of workload class c at resource i
$R_{i,c}$	Average response time of workload class c at resource i
R_c	Average end-to-end response time of workload class c
$A_{i,c}$	Average queue length of requests of workload class c seen upon arrival at resource i (excluding the arriving job)
$V_{i,c}$	Average number of visits of a request of workload class c at resource i
I	Total number of resources
C	Total number of workload classes

17.1 Estimation Methods

In this section, we describe the most common methods for resource demand estimation that exist in the literature. Table 17.2 gives an overview of the different methods.

Table 17.2: Overview of estimation methods categorized according to the underlying statistical techniques

Technique	Variant	References
Approximation with response times		Urgaonkar et al. (2007) Nou et al. (2009) Brosig et al. (2009)
Service demand law		Lazowska et al. (1984) Brosig et al. (2009)
Linear regression	Least squares	Bard and Shatzoff (1978) Rolia and Vetland (1995) Pacifici et al. (2008) Kraft et al. (2009); Pérez, Pacheco-Sanchez, et al. (2013)
	Least absolute differences	Stewart et al. (2007); Q. Zhang et al. (2007)
	Least trimmed squares	Casale et al. (2008); Casale et al. (2007)
Kalman filter		Zheng et al. (2008) Kumar, Tantawi, et al. (2009) Wang, Huang, Qin, et al. (2012); Wang, Huang, Song, et al. (2011)
Optimization	Non-linear constrained optimization	L. Zhang et al. (2002) Menascé (2008)
	Quadratic programming	Liu et al. (2006); Wynter et al. (2004) Kumar, L. Zhang, et al. (2009)
Machine learning	Clusterwise linear regression	Cremonesi, Dhyani, et al. (2010)
	Independent component analysis	Sharma et al. (2008)
	Support vector machine	Kalbasi, Krishnamurthy, Rolia, and Richter (2011)
	Pattern matching	Cremonesi and Sansottera (2012, 2014)
Maximum likelihood estimation		Kraft et al. (2009) Pérez, Pacheco-Sanchez, et al. (2013)
Gibbs sampling		Sutton and Jordan (2011) Wang and Casale (2013)
Demand estimation with confidence (DEC)		Kalbasi, Krishnamurthy, Rolia, and Dawson (2012); Rolia, Kalbasi, et al. (2010)

17.1.1 Approximation with Response Times

Assuming a single queue and insignificant queueing delays compared to the resource demands, we can approximate the resource demands with the observed response times. However, this trivial approximation only works with systems under light load where a single resource dominates the observed response time. This approximation is used by Nou et al. (2009), Urgaonkar et al. (2007), and Brosig et al. (2009).

17.1.2 Service Demand Law

The service demand law (cf. Chapter 7, Sections 7.1.2 and 7.2.3) is an operational law that can be used to directly calculate the demand $D_{i,c}$ given the utilization $U_{i,c}$ and the throughput $X_{i,c}$. However, modern operating systems can report the utilization only on a per-process level. Therefore, we usually cannot observe the per-class utilization $U_{i,c}$ directly, given that single processes may serve requests of different workload classes. Given a system serving requests of multiple workload classes, Lazowska et al. (1984) and Menascé et al. (2004) recommend to use additional per-class metrics if available (e.g., in the operating system) to apportion the aggregate utilization U_i of a resource between workload classes. Brosig et al. (2009) use an approximate apportioning scheme based on the assumption that the observed response times are proportional to the resource demands.

17.1.3 Linear Regression

Given a linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta}$ (cf. Chapter 2, Section 2.7.1) is a vector of resource demands $D_{i,r}$ and $\mathbf{Y}$, $\mathbf{X}$ contain observations of performance metrics of a system, we can use linear regression to estimate the resource demands. Two alternative formulations of such a linear model have been proposed in the literature:

- The utilization law (cf. Chapter 7, Sections 7.1.1 and 7.2.3) requires observations of the aggregate utilization U_i and the throughputs $\lambda_{i,c}$. This is a classical model used by different authors (Bard and Shatzoff, 1978; Casale et al., 2007; Kraft et al., 2009; Pacifici et al., 2008; Rolia and Vetland, 1995; Stewart et al., 2007; Q. Zhang et al., 2007). Some of the authors include a constant term $U_{i,0}$ in the model in order to estimate the utilization caused by background work.
- Kraft et al. (2009) and Pérez, Pacheco-Sanchez, et al. (2013) propose a linear model based on a multi-class version of the response time equation $R_i = D_i(1+A_i)$ requiring observations of the queue length A_i seen by a newly arriving job and its response time R_i. In their initial work, Kraft et al. (2009) assume a FCFS scheduling strategy; Pérez, Pacheco-Sanchez, et al. (2013) generalize the model to PS queueing stations.

Bard and Shatzoff (1978), Rolia and Vetland (1995), Pacifici et al. (2008), and Kraft et al. (2009) use nonnegative least squares regression for solving the linear model. Other regression techniques, such as least absolute differences regression (Stewart et al., 2007; Q. Zhang et al., 2007) or least trimmed squares (Casale et al., 2008; Casale et al., 2007), were proposed to increase the robustness of regression-based estimation techniques to multi-collinearities, outliers, or abrupt changes in the demand values.

17.1.4 Kalman Filter

The resource demands of a system may vary over time, for example, due to changing system states or changing user behavior. These variations may be abrupt or continuous. In order to track time-varying resource demands, Zheng et al. (2008), Kumar, Tantawi, et al. (2009), and Wang, Huang, Qin, et al. (2012) use a Kalman filter (cf. Chapter 2, Section 2.7.2). The authors assume a dynamic system where the state vector $\mathbf{x}$ consists of the hidden resource demands $D_{i,c}$ that need to be estimated. Given that no prior knowledge about the dynamic behavior of the system state exists, they assume a constant state model; that is, Equation (2.49) on page 40 is reduced to $\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{w}_k$.

The observation model $\mathbf{z} = h(\mathbf{x})$ requires a functional description of the relationship between the observations $\mathbf{z}$ and the system state $\mathbf{x}$. Wang, Huang, Qin, et al. (2012) use the observed utilization U_i as vector $\mathbf{z}$ and define $h(\mathbf{x})$ based on the utilization law (cf. Equation 7.38 on page 167). Given the linear model, a conventional Kalman filter is sufficient. Zheng et al. (2008) and Kumar, Tantawi, et al. (2009) use an observation vector consisting of the observed response time $R_{i,c}$ of each workload class and the utilization U_i of each resource. The function $h(\mathbf{x})$ is defined based on the solution of a M/M/1 queue (cf. Equation 7.43 on page 168) and the utilization law. Due to the non-linear nature, it requires an extended Kalman filter design—see Equation (2.51) on page 40.

17.1.5 Optimization

Given a general queueing network, we can formulate an optimization problem to search for values of the resource demands so that the differences between performance metrics observed on the real system and the ones calculated using the queueing network are minimized. The main challenge is the solution of the queueing network. Depending on the structure of the queueing network, its solution may be computationally expensive and the optimization algorithm may need to evaluate the queueing network with many different resource demand values in order to find an optimal solution. Existing approaches (Kumar, L. Zhang, et al., 2009; Liu et al., 2006; Menascé, 2008; Wynter et al., 2004) assume a product-form queueing network

with an open workload. Then, the equations in Chapter 7, Section 7.2.4, can be used to calculate the end-to-end response times.

Given N observations of the end-to-end response time $\tilde{R}_c$ and the utilization $\tilde{U}_i$, Liu et al. (2006) propose the following objective function:

$$\min_{\mathbf{D}} \sum_{n=1}^{N} \left(\sum_{c=1}^{C} p_c \left(R_c(\mathbf{D}) - \tilde{R}_c^{(n)} \right)^2 + \sum_{i=1}^{I} \left(U_i(\mathbf{D}) - \tilde{U}_i^{(n)} \right)^2 \right). \tag{17.1}$$

The function $R_c(\mathbf{D})$ is based on the solution of a M/M/1 queue—see Equation (7.43) on page 168—and $U_i(\mathbf{D})$ on the utilization law.

The factor p_c introduces a weighting according to the arrival rates of workload classes $p_c = \lambda_c / \sum_{d=1}^{C} \lambda_d$. The resulting optimization problem can be solved using quadratic programming techniques.

Kumar, L. Zhang, et al. (2009) extend this optimization approach to estimate load-dependent resource demands. Their approach requires prior knowledge of the type of function (e.g., polynomial, exponential, or logarithmic) that best describes the relation between arriving workloads and resource demands.

Menascé (2008) formulates an alternative optimization problem that depends only on response time and arrival rate measurements:

$$\min_{\mathbf{D}} \sum_{c=1}^{C} \left(R_c(\mathbf{D}) - \tilde{R}_c \right)^2 \text{ with } R_c(\mathbf{D}) = \sum_{i=1}^{I} \frac{D_{i,c}}{1 - \sum_{d=1}^{C} \lambda_{i,d} D_{i,d}} \tag{17.2}$$

$$\text{subject to } D_{i,c} \geq 0 \quad \forall i, c \text{ and } \sum_{c=1}^{C} \lambda_{i,c} D_{i,c} < 1 \quad \forall i.$$

In contrast to Liu et al. (2006), this formulation is based on a single sample of the observed response times. Menascé (2008) proposes to repeat the optimization for each new sample using the previous resource demand estimate as the initial point. To solve this optimization problem we depend on a non-linear constrained optimization algorithm.

17.1.6 Machine Learning

Cremonesi, Dhyani, et al. (2010) use clusterwise regression techniques to improve the robustness to discontinuities in the resource demands due to system configuration changes. The observations are clustered into groups where the resource demands can be assumed constant, and the demands are then estimated for each cluster separately. In Cremonesi and Sansottera (2012) and Cremonesi and Sansottera (2014) an algorithm is proposed based on a combination of change-point regression methods and pattern matching to address the same challenge.

Independent Component Analysis (ICA) is a method to solve the *blind source separation* problem (i.e., to estimate the individual signals from a set of aggregate

measurements). Sharma et al. (2008) describe a way to use ICA for resource demand estimation using a linear model based on the utilization law. ICA can provide estimates solely based on utilization measurements when the following constraints hold (Sharma et al., 2008): (1) the number of workload classes is limited by the number of observed resources, (2) the arrival rate measurements are statistically independent, and (3) the inter-arrival times have a non-Gaussian distribution while the measurement noise is assumed to be zero-mean Gaussian. ICA not only provides estimates of resource demands, but also automatically categorizes requests into workload classes.

Kalbasi, Krishnamurthy, Rolia, and Richter (2011) consider the use of Support Vector Machines (SVM) (Smola and Schölkopf, 2004) for estimating resource demands. They compare it with results from LSQ and LAD regression and show that it can provide better resource demand estimates depending on the characteristics of the workload.

17.1.7 Maximum Likelihood Estimation (MLE)

Kraft et al. (2009) and Pérez, Pacheco-Sanchez, et al. (2013) use Maximum Likelihood Estimation (MLE) (cf. Chapter 2, Section 2.7.3) to estimate resource demands based on observed response times and queue lengths seen upon arrival of requests. Given N response time measurements $R_i^1, \ldots, R_i^N$ of *individual* requests, the estimated resource demands $D_{i,1}, \ldots, D_{i,C}$ are the values that maximize the likelihood function $\mathbb{L}(D_{i,1}, \ldots, D_{i,C})$ defined as follows:

$$\max \mathbb{L}(D_{i,1}, \ldots, D_{i,C}) = \sum_{k=1}^{N} \log f(R_i^k \mid D_{i,1}, \ldots, D_{i,C}). \tag{17.3}$$

The density function f is obtained by constructing a phase-type distribution. The phase-type distribution describes the time to absorption in a Markov chain representing the current state of the system. Observations of the queue lengths are necessary in order to be able to construct the corresponding phase-type distribution. Kraft et al. (2009) describe the likelihood function for queueing stations with FCFS scheduling. Pérez, Pacheco-Sanchez, et al. (2013) generalize this approach to PS scheduling.

17.1.8 Gibbs Sampling

Sutton and Jordan (2011) and Wang and Casale (2013) both propose approaches to resource demand estimation based on Bayesian inference techniques (cf. Section 2.7.4). Sutton and Jordan (2011) assume an open, single-class queueing network. They develop a deterministic mathematical model allowing for the calculation of service times and waiting times of individual requests given the arrival times,

departure times, and the path of queues of all requests in a queueing network. They assume that this information can be only observed for a subset of requests. Therefore, they propose a Gibbs sampler to sample the missing departure times of requests that were not observed. Given the posterior distribution of the departure times of all requests, they then derive the expected resource demands at the individual queues.

Wang and Casale (2013) assume a multi-class, closed queueing network that fulfills the BCMP theorem (cf. Chapter 7, Section 7.2.2). Under this assumption, the probability distribution of the queue lengths for given resource demands is well-known (see Equation 7.36 on page 166). They assume the availability of queue-length samples from a real system and construct a Gibbs sampler for the posterior distribution $f(\mathbf{D}|\mathbf{A})$, where $\mathbf{D}$ is a vector of resource demands $D_{i,c}$ and $\mathbf{A}$ is a vector of observed queue lengths $A_{i,c}$. They propose an approximation for the conditionals of the posterior distribution as required by the Gibbs sampling algorithm. A main challenge is the calculation of the normalization constant G for the steady-state probabilities (cf. Equation 7.36 on page 166), which is nontrivial for a closed queueing network. Wang and Casale (2013) propose a Taylor expansion of G and apply an algorithm based on mean-value analysis (MVA) to determine its value.

17.1.9 Other Approaches

Rolia, Kalbasi, et al. (2010) and Kalbasi, Krishnamurthy, Rolia, and Dawson (2012) propose a technique called Demand Estimation with Confidence (DEC) for estimating the aggregate resource demand of a given workload mix. This technique assumes that a set of benchmarks is available for the system under study. Each benchmark utilizes a subset of the different functions of an application. DEC expects the measured demands of the individual benchmarks as input and then derives the aggregate resource demand of a given workload mix as a linear combination of the demands of the individual benchmarks. DEC is able to provide confidence intervals of the aggregate resource demand (Kalbasi, Krishnamurthy, Rolia, and Dawson, 2012; Rolia, Kalbasi, et al., 2010).

17.2 Input Parameters

Methods for resource demand estimation often differ in terms of the set of input data they require. We do not consider parameters of the underlying statistical techniques (e.g., parameters controlling an optimization algorithm) because they normally are specific to the concrete implementation of an estimation method.

Figure 17.1 depicts the main types of input parameters for demand estimation algorithms. The parameters are categorized into *model parameters* and *measurements*. In general, parameters of both types are required. Model parameters capture information about the performance model for which we estimate resource demands.

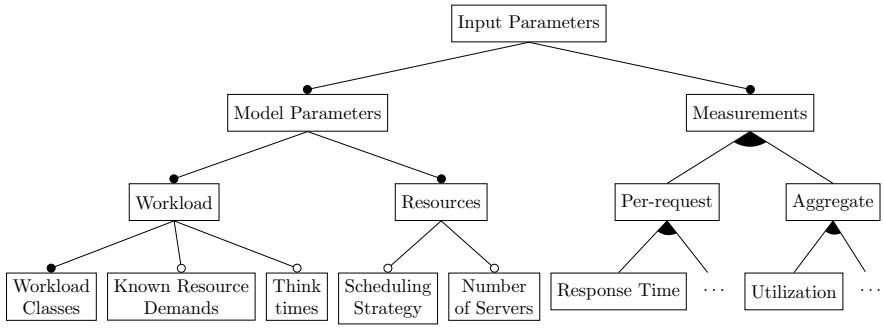

Fig. 17.1: Types of input parameters

Measurements consist of samples of relevant performance metrics obtained from a running system, either a live production system or a test system.

Before estimating resource demands, it is necessary to decide on certain modeling assumptions. As a first step, resources and workload classes need to be identified. This is typically done as part of the workload characterization activity when modeling a system. It is important to note that the observability of performance metrics may influence the selection of resources and workload classes for the system under study. In order to be able to distinguish between individual resources or workload classes, observations of certain per-resource or per-class performance metrics are necessary. At a minimum, information about the number of workload classes and the resources for which the demands should be determined is required as input to the estimation. Depending on the estimation method, more detailed information on resources and workload classes may be expected as input (e.g., *scheduling strategies*, *number of servers*, or *think times*).

Measurements can be further grouped on a *per-request* or *aggregate* basis. Common per-request measurements used in the literature include response times, arrival rates, visit counts, and queue lengths seen upon arrival. Aggregate measurements can be further distinguished in *class-aggregate* and *time-aggregate* measurements. Class-aggregate measurements are collected as totals over all workload classes processed at a resource. For instance, utilization is usually reported as an aggregate value because the operating system is agnostic of the application internal logic and is not aware of different request types in the application. Time-aggregate measurements (e.g., average response times or average throughput) are aggregated over a sampling period. The sampling period can be evenly or unevenly spaced.

Categorization of Existing Methods

We consider the methods for resource demand estimation listed in Table 17.2 and examine their input parameters. Table 17.3 shows an overview of the input param-

Table 17.3: Input parameters of estimation methods

Estimation method	Measurements					Parameters		
	U_i	R_c	X_c/λ_c	$A_{i,c}$	$V_{i,c}$	$D_{i,c}$	Z	P
Approximation with response times								
Urgaonkar et al. (2007)		✗[1]			✗			
Nou et al. (2009)	✗	✗						
Brosig et al. (2009)		✗						
Service demand law								
Lazowska et al. (1984)			✗			✗[2]		
Brosig et al. (2009)	✗	✗			✗			
Linear regression								
Bard and Shatzoff (1978)								
Rolia and Vetland (1995)								
Pacifici et al. (2008)	✗		✗					
Q. Zhang et al. (2007)								
Stewart et al. (2007)	✗		✗					
Kraft et al. (2009); Pérez, Casale, et al. (2015)		✗		✗				✗
Casale et al. (2008); Casale et al. (2007)	✗		✗					
Kalman filter								
Zheng et al. (2008)	✗	✗	✗					
Kumar, Tantawi, et al. (2009)	✗	✗	✗					
Wang, Huang, Qin, et al. (2012)	✗		✗					
Optimization								
L. Zhang et al. (2002)	✗	✗	✗			(✗)[5]		✗
Liu et al. (2006); Wynter et al. (2004)	✗	✗	✗		✗			✗
Menascé (2008)		✗	✗			✗[3]		
Kumar, L. Zhang, et al. (2009)	✗	✗	✗					✗
Machine learning								
Cremonesi, Dhyani, et al. (2010)	✗		✗					
Sharma et al. (2008)	✗							
Kalbasi, Krishnamurthy, Rolia, and Richter (2011)	✗		✗					
Cremonesi and Sansottera (2012, 2014)	✗		✗					
Maximum likelihood estimation								
Kraft et al. (2009)		✗[4]		✗[4]			✗	✗
Pérez, Casale, et al. (2015)		✗[4]		✗[4]			✗	✗
Gibbs sampling								
Sutton and Jordan (2011)		✗[4]	✗[4]					✗
Wang and Casale (2013)				✗[4]			✗	
Kalbasi, Krishnamurthy, Rolia, and Dawson (2012); Rolia, Kalbasi, et al. (2010)			✗		✗			

[1] Response time per resource
[2] Measured with accounting monitor—system overhead not included
[3] A selected set of resource demands is known a priori
[4] Non-aggregated measurements of individual requests
[5] Requires coefficient of variation of resource demands in case of FCFS scheduling

eters of each estimation method (utilization U_i, response time R_c, throughput X_c, arrival rate λ_c, queue length $A_{i,c}$, visit counts $V_{i,c}$, resource demands $D_{i,c}$, think time Z, and scheduling policy P). Parameters common to all estimation methods, such as the number of workload classes and the number of resources, are not included in this table. The required input parameters vary widely between different estimation methods. Depending on the system under study and on the available performance metrics, one can choose a suitable estimation method from Table 17.3. Furthermore, approaches based on optimization can be adapted by incorporating additional constraints into the mathematical model capturing the knowledge about the system under study. For example, the optimization approach by Menascé (2008) allows one to specify additional known resource demand values as input parameters. These a priori resource demands may be obtained from the results of other estimation methods or from direct measurements.

Another approach that requires resource demand data is described by Lazowska et al. (1984, Chapter 12) who assume that the resource demands are approximated based on measurements provided by an accounting monitor; however, such an accounting monitor does not include the system overhead caused by each workload class. The system overhead is defined as the work done by the operating system for processing a request. Lazowska et al. (1984) describe a way to distribute unattributed computing time among the different workload classes, providing more realistic estimates of the actual resource demands.

Approaches based on response time measurements, such as those proposed by L. Zhang et al. (2002), Liu et al. (2006), Wynter et al. (2004), and Kumar, L. Zhang, et al. (2009), require information about the scheduling strategies of the involved resources abstracted as queueing stations. This information is used to construct the correct problem definition for the optimization technique. The estimation methods proposed by Kraft et al. (2009), Pérez, Pacheco-Sanchez, et al. (2013), and Wang and Casale (2013) assume a closed queueing network. Therefore, they also require the average think time and the number of users as input.

In addition to requiring a set of specific input parameters, some approaches also provide a rule of thumb regarding the number of required measurement samples. Approaches based on linear regression (Kraft et al., 2009; Pacifici et al., 2008; Rolia and Vetland, 1995) need at least $K + 1$ linear independent equations to estimate K resource demands. When using robust regression methods, significantly more measurements might be necessary (Casale et al., 2008; Casale et al., 2007). Kumar, L. Zhang, et al. (2009) provide a formula to calculate the number of measurements required by their optimization-based approach. The formula provides only a minimum bound on the number of measurements and more measurements are normally required to obtain good estimates (Stewart et al., 2007).

17.3 Output Metrics

Approaches to resource demand estimation are typically used to determine the mean resource demand of requests of a given workload class at a given resource. However, in many situations, the estimated mean value may not be sufficient. Often, more information about the confidence of estimates and the distribution of the resource demands is required. The set of output metrics an estimation method provides can influence the decision to adopt a specific method.

Generally, resource demands cannot be assumed to be deterministic (Rolia, Kalbasi, et al., 2010); for example, they may depend on the data processed by an application or on the current state of the system (Rolia and Vetland, 1995). Therefore, resource demands are described as random variables. Estimates of the mean resource demand should be provided by every estimation method. If the distribution of the resource demands is not known beforehand, estimates of higher moments of the resource demands may be useful to determine the shape of their distribution.

We distinguish between point and interval estimators of the real resource demands. Generally, confidence intervals would be preferable; however, it is often a challenge to ensure that the statistical assumptions underlying a confidence interval calculation hold for a system under study (e.g., distribution of the regression errors).

In certain scenarios, for example, if DVFS or hyperthreading techniques are used (Kumar, L. Zhang, et al., 2009), the resource demands are load-dependent. In such cases, the resource demands are not constant; they are rather a function that may depend, for example, on the arrival rates of the workload classes (Kumar, L. Zhang, et al., 2009).

Categorization of Existing Methods

Table 17.4 provides an overview of the output metrics of the considered estimation methods. Point estimates of the mean resource demand are provided by all approaches. Confidence intervals can be determined for linear regression using standard statistical techniques as mentioned by Rolia and Vetland (1995) and Kraft et al. (2009). These techniques are based on the Central Limit Theorem (cf. Section 2.5 in Chapter 2), assuming an error term with a Normal distribution. Resource demands are typically not deterministic, violating the assumptions underlying linear regression. The influence of the distribution of the resource demands on the accuracy of the confidence intervals is not evaluated for any of the approaches based on linear regression. DEC (Kalbasi, Krishnamurthy, Rolia, and Dawson, 2012; Rolia, Kalbasi, et al., 2010) is the only approach for which the confidence intervals have been evaluated in the literature. The MLE approach (Kraft et al., 2009) and the optimization approach described by L. Zhang et al. (2002) are capable of providing estimates of higher moments. This additional information comes at the cost of a higher amount of required measurements.

Table 17.4: Output metrics of estimation methods

Estimation method	Resource demands			
	Point estimates	Confidence interval	Higher moments	Load-dependent
Response time approximation				
Urgaonkar et al. (2007)	✗			
Nou et al. (2009)	✗			
Brosig et al. (2009)	✗			
Service demand law				
Lazowska et al. (1984)	✗			
Brosig et al. (2009)	✗			
Linear regression				
Bard and Shatzoff (1978) Rolia and Vetland (1995), Pacifici et al. (2008)	✗	✗[2]		
Q. Zhang et al. (2007)	✗	✗[2]		
Kraft et al. (2009); Pérez, Casale, et al. (2015); Pérez, Pacheco-Sanchez, et al. (2013)	✗	✗[2]		
Casale et al. (2008); Casale et al. (2007)	✗	✗[2]		
Kalman filter				
Zheng et al. (2008)	✗			
Kumar, Tantawi, et al. (2009)	✗			
Wang, Huang, Qin, et al. (2012)	✗			
Optimization				
L. Zhang et al. (2002)	✗		✗[1]	
Liu et al. (2006); Wynter et al. (2004)	✗			
Menascé (2008)	✗			
Kumar, L. Zhang, et al. (2009)	✗			✗
Machine learning				
Cremonesi, Dhyani, et al. (2010)	✗			
Sharma et al. (2008)	✗			
Kalbasi, Krishnamurthy, Rolia, and Richter (2011)	✗			
Cremonesi and Sansottera (2012, 2014)	✗			
Maximum likelihood estimation				
Kraft et al. (2009)	✗		✗	
Pérez, Casale, et al. (2015)	✗		✗	
Gibbs sampling				
Sutton and Jordan (2011)	✗			
Wang and Casale (2013)	✗			
Kalbasi, Krishnamurthy, Rolia, and Dawson (2012); Rolia, Kalbasi, et al. (2010) (DEC)	✗	✗		

[1] Only feasible if a priori knowledge of the resource demand variance is available.
[2] The accuracy of the confidence intervals is not evaluated.

All of the estimation methods in Table 17.2 can estimate load-independent mean resource demands. Additionally, the enhanced inferencing approach (Kumar, L. Zhang, et al., 2009) also supports the estimation of load-dependent resource demands, assuming a given type of function.

17.4 Robustness

Usually, it is not possible to control every aspect of a system while collecting measurements. This can lead to anomalous behavior in the measurements. Casale et al. (2007), Casale et al. (2008), and Pacifici et al. (2008) identified the following issues with real measurement data:

- presence of outliers,
- background noise,
- non-stationary resource demands,
- collinear workload, and
- insignificant flows.

Background activities can have two effects on measurements: the presence of outliers and background noise. Background noise is created by secondary activities that utilize a resource only lightly over a long period of time. Outliers result from secondary activities that stress a resource at high utilization levels for a short period of time. Outliers can have a significant impact on the parameter estimation resulting in biased estimates (Casale et al., 2007). Different strategies are possible to cope with outliers. It is possible to use special filtering techniques in an upstream processing step or to use parameter estimation techniques that are inherently robust to outliers. However, tails in measurement data from real systems might belong to bursts (e.g., resulting from rare but computationally complex requests). The trade-off decision as to when an observation is to be considered an outlier has to be made on a case-by-case basis, taking into account the characteristics of the specific scenario and application.

The resource demands of a system may be non-stationary over time (i.e., not only the arrival process may change over time, but also the resource demands, which, for example, can be described by a $M_t/M_t/1$ queue). Different types of changes are observed in production systems. Discontinuous changes in the resource demands can be caused by software and hardware reconfigurations, for example, the installation of an operating system update (Casale et al., 2007). Continuous changes in the resource demands may happen over different time scales. Short-term variations can often be observed in cloud computing environments where different workloads experience mutual influences due to the underlying shared infrastructure. Changes in the application state (e.g., database size) or the user behavior (e.g., increased number of items in a shopping cart in an online shop during Christmas season) may result in long-term trends and seasonal patterns (over days, weeks, and months). When using the estimated resource demands to forecast the required resources of an application

over a longer time period, these non-stationary effects need to be considered in order to obtain accurate predictions. In order to detect such trends and seasonal patterns, it is possible to apply forecasting techniques on a time series resulting from the repeated execution of one considered estimation method over a certain time period. An overview of such forecasting approaches based on time series analysis can be found in Box et al. (2015).

Another challenge for estimation methods is the existence of collinearities in the arrival rates of different workload classes. There are two possible reasons for collinearities in the workload: low variation in the throughput of a workload class or dependencies between workload classes (Pacifici et al., 2008). For example, if we model *login* and *logout* requests each with a separate workload class, the resulting classes would normally be correlated. The number of logins usually approximately matches the number of logouts. Collinearities in the workload may have negative effects on resource demand estimates. A way to avoid these problems is to detect and combine workload classes that are correlated.

Insignificant flows are caused by workload classes with very small arrival rates in relation to the arrival rates of the other classes. Pacifici et al. (2008) experience numerical stability problems with their linear regression approach when insignificant flows exist. However, it is noteworthy that there might be a dependency between insignificant flows and the length of the sampling time intervals. If the sampling time interval is too short, the variance in arrival rates might be high.

Categorization of Existing Methods

Ordinary least-squares regression is often sensitive to outliers. Stewart et al. (2007) come to the conclusion that least-absolute-differences regression is more robust to outliers. Robust regression techniques, as described in Casale et al. (2007) and Casale et al. (2008), try to detect outliers and ignore measurement samples that cannot be explained by the regression model. Liu et al. (2006) also include an outlier detection mechanism in their estimation method based on optimization.

In general, sliding window or data aging techniques can be applied to the input data to improve the robustness to non-stationary resource demands (Pacifici et al., 2008). In order to detect software and hardware configuration discontinuities, robust and clusterwise regression approaches are proposed by Casale et al. (2007), Casale et al. (2008), and Cremonesi, Dhyani, et al. (2010). If such discontinuities are detected, the resource demands are estimated separately before and after the configuration change. Approaches based on Kalman filters (Kumar, Tantawi, et al., 2009; Zheng et al., 2008) are designed to estimate time-varying parameters. Therefore, they automatically adapt to changes in the resource demands after a software or hardware discontinuity. None of the considered estimation methods is able to learn long-term trends or seasonal patterns (over days, weeks, or months).

Collinearities are one of the major issues when using linear regression (Chatterjee and Price, 1995). A common method to cope with this issue is to check the workload

classes for collinear dependencies before applying linear regression. If collinearities are detected, the involved workload classes are merged into one class. This is proposed by Pacifici et al. (2008) and Casale et al. (2007). The DEC approach (Rolia, Kalbasi, et al., 2010) mitigates collinear dependencies, since it estimates the resource demands only for mixes of workload classes.

Pacifici et al. (2008) also consider insignificant flows. They call a workload class insignificant if the ratio between the throughput of the workload class and the throughput of all workload classes is below a given threshold. They completely exclude insignificant workload classes from the regression in order to avoid numerical instabilities.

17.5 Estimation Accuracy and Execution Time

Depending on the concrete application scenario, the presented methods for resource demand estimation can differ significantly in terms of their accuracy and execution time. Spinner et al. (2015) present a comprehensive experimental comparison, evaluating the different estimation methods in terms of their accuracy and overhead. The aim of the evaluation is to answer the following questions:

- How do the different methods compare in terms of estimation accuracy and execution time?
- Which factors influence the estimation accuracy of the different methods?
- How to automatically decide which set of estimation methods to apply in a given scenario?

To address these questions, the influence of the following factors on the estimation accuracy is evaluated: length of sampling interval, number of samples, number of workload classes, load level, collinearity of workload classes, missing workload classes for background activities, and presence of delays during processing at a resource. Table 17.5 lists the estimation methods considered in the experimental evaluation.

Table 17.5: Estimation methods considered in the experimental evaluation

Abbreviation	Estimation method
SDL	Service demand law (Brosig et al., 2009)
UR	Utilization regression (Rolia and Vetland, 1995)
KF	Kalman filter (Kumar, Tantawi, et al., 2009)
MO	Menascé optimization (Menascé, 2008)
LO	Liu optimization (Liu et al., 2006)
RR	Response time regression (Kraft et al., 2009)
GS	Gibbs sampling (Wang, Huang, Qin, et al., 2012)

In the following, we summarize the results of the experimental comparison by Spinner et al. (2015):

- When using estimation methods based on time-aggregated observations (e.g., UR, KF, MO, or LO), the length of the sampling interval is an important parameter that needs to be adjusted to the system under study. A good sampling interval length depends on the response times of requests and the number of requests observed in one interval. The sampling interval should be significantly larger than the response times of requests to avoid end effects, and it should be long enough to be able to calculate the aggregate value based on the observations of a significant number of requests (more than 60 requests per sampling interval has proven to provide good results).
- Most estimation methods (except MO and LO) are negatively influenced when reducing the experiment length to 10 min (i.e., 10 samples). However, they still yield results with acceptable accuracy (relative demand error below 8%).
- All estimation methods are sensitive to the number of workload classes. The linear regression method UR, which uses only utilization and throughput observations, generally yields a degraded accuracy in scenarios with several workload classes. Observations of the response times of requests can help to improve the estimation accuracy significantly even in situations with a very high number of workload classes. However, it is crucial to ensure that the modeling assumptions of the estimation methods using response times are fulfilled as they are highly sensitive to violated assumptions (e.g., incorrect scheduling strategies). Furthermore, insignificant flows can impair resource demand estimation. Workload classes with a small contribution to the total resource demand of a system should therefore be excluded from resource demand estimation.
- When a system operates at a high utilization level (80% or higher), the estimation methods KF, MO, LO, and GS may yield inaccurate results.
- Collinearities in throughput observations of different workload classes impair the estimation accuracy of UR. While it correctly estimates the total resource demand, the apportioning between workload classes is wrong. The other estimation methods are much less sensitive to collinearities in throughput observations.
- Methods that rely on response time observations (e.g., MO, RR, and GS) are more robust to missing workload classes than methods based on utilization.
- Delays due to non-captured software or hardware resources have a strong influence on the estimation accuracy of estimation methods based on observed response times. While some estimation methods (e.g., L. Zhang et al. (2002), Liu et al. (2006), and Menascé (2008)) consider scenarios where multiple resources contribute to the observed end-to-end response time, only Pérez, Pacheco-Sanchez, et al. (2013) consider contention due to *software* resources.
- There are significant differences in the computational complexity of the different estimation methods. In the considered datasets, the estimation takes between under 1 ms and up to 20 s depending on the estimation method. When using resource demand estimation techniques on a production system (e.g., for online performance and resource management), the computational effort needs to be taken into account (especially in data centers with a large number of systems).

In summary, the evaluation shows that using response times can improve the accuracy of the estimated resource demands significantly compared to the traditional approach based on the utilization law using linear regression, especially in cases with multiple workload classes. However, estimation methods employing response time measurements are very sensitive if assumptions of the underlying mathematical model are violated (e.g., incorrect scheduling strategy).

17.6 Library for Resource Demand Estimation (LibReDE)

While the presented systematization and experimental comparison provide a solid basis for selecting the right resource demand estimation method for a given scenario, the selection is still not trivial and requires expertise on the underlying statistical techniques and their assumptions. Also, in many cases, it may be infeasible to determine the right method in advance, as the respective input data may only be available at system run time and the decision would have to be made on-the-fly. Furthermore, the system and its workload may change over time requiring a dynamic switchover to a different estimation method.

Spinner (2017) presents an approach to resource demand estimation that relies on multiple statistical techniques for improved robustness and uses a cross-validation scheme to dynamically select the technique that performs best for the concrete scenario. This simplifies the usage of resource demand estimation methods for performance engineers. Furthermore, it is a crucial building block for Application Performance Management (APM) techniques that automatically estimate resource demands at system run time and use them for online resource management. The approach has been implemented as an open-source tool called LibReDE.[1] The tool includes a library for resource demand estimation, providing ready-to-use implementations of eight common estimation methods.

The main idea of LibReDE is to leverage *multiple statistical techniques* combined with a *feedback loop* to improve the accuracy of the resource demand estimation by iteratively: (1) adapting the estimation problem, (2) selecting suitable statistical methods to be applied, and (3) optimizing the configuration parameters of each method. LibReDE uses *cross-validation* techniques with an error metric based on the deviation between the observed response times and utilization, on the one hand, and the respective predicted metrics using the resource demand estimates, on the other hand.

LibReDE applies multiple statistical techniques in an online setting, automatically combining, weighting, and iteratively refining their results (in a feedback loop) to produce as accurate estimates as possible. Further details on LibReDE and the respective estimation approach it implements can be found in Spinner (2017).

[1] http://descartes.tools/librede

17.7 Concluding Remarks

In this chapter, we surveyed, systematized, and evaluated different approaches to the statistical estimation of resource demands based on easy to measure system-level and application-level metrics. The goal of the presented systematization is to help performance engineers select an estimation method that best fits their specific requirements. We first surveyed existing estimation methods and described their modeling assumptions and their underlying statistical techniques. Then, we introduced three dimensions for systematization: (1) input parameters, (2) output metrics, and (3) robustness to anomalies in the input data. For each dimension, we first described its features and then categorized the estimation methods accordingly. We considered resource demands in the context of computing systems; however, the methods we presented are also applicable to other types of systems. We focused on generic methods to determine resource demands without relying on dedicated instrumentation of the application. The goal was to estimate the resource demands based on *indirect measurements* derived from commonly available metrics (e.g., end-to-end response time or resource utilization). We summarized the results of a comprehensive experimental comparison evaluating the different estimation methods in terms of their accuracy and overhead. The evaluation revealed that using response times can improve the accuracy of the estimated resource demands significantly compared to the traditional approach based on the utilization law using linear regression, especially in cases with multiple workload classes. However, estimation methods employing response time measurements are very sensitive if assumptions of the underlying mathematical model are violated.

References

Bard, Y. and Shatzoff, M. (1978). "Statistical Methods in Computer Performance Analysis". In: *Current Trends in Programming Methodology Vol. III: Software Modeling*. Ed. by K. M. Chandy and R. T.-Y. Yeh. Prentice-Hall: NJ, Englewood Cliffs, pp. 1–51 (cited on pp. 368–370, 375, 378).

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. Fifth edition. Wiley Series in Probability and Statistics. John Wiley & Sons: Hoboken, New Jersey, USA (cited on p. 380).

Brosig, F., Kounev, S., and Krogmann, K. (2009). "Automated Extraction of Palladio Component Models from Running Enterprise Java Applications". In: *Proceedings of the Fourth International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2009)—ROSSA 2009 Workshop*. (Pisa, Italy). ICST/ACM (cited on pp. 368, 369, 375, 378, 381).

Casale, G., Cremonesi, P., and Turrin, R. (2008). "Robust Workload Estimation in Queueing Network Performance Models". In: *Proceedings of the 16th Euromicro*

Conference on Parallel, Distributed and Network-Based Processing (PDP 2008). (Toulouse, France). IEEE: Piscataway, New Jersey, USA, pp. 183–187 (cited on pp. 368, 370, 375, 376, 378–380).

Casale, G., Cremonesi, P., and Turrin, R. (2007). "How to Select Significant Workloads in Performance Models". In: *Proceedings of the 33rd International Computer Measurement Group Conference (CMG 2007)*. (San Diego, CA, USA) (cited on pp. 368–370, 375, 376, 378–381).

Chatterjee, S. and Price, B. (1995). *Praxis der Regressionsanalyse*. Second Edition. Oldenbourg Wissenschaftsverlag: Munich, Germany (cited on p. 380).

Cremonesi, P., Dhyani, K., and Sansottera, A. (2010). "Service Time Estimation with a Refinement Enhanced Hybrid Clustering Algorithm". In: *Analytical and Stochastic Modeling Techniques and Applications—17th International Conference, ASMTA 2010—Proceedings*. (Cardiff, UK). Ed. by K. Al-Begain, D. Fiems, and W. J. Knottenbelt. Vol. 6148. Lecture Notes in Computer Science. Springer-Verlag: Berlin, Heidelberg, pp. 291–305 (cited on pp. 368, 371, 375, 378, 380).

Cremonesi, P. and Sansottera, A. (2012). "Indirect Estimation of Service Demands in the Presence of Structural Changes". In: *Proceedings of the 2012 Ninth International Conference on Quantitative Evaluation of Systems (QEST 2012)*. IEEE Computer Society: Washington, DC, USA, pp. 249–259 (cited on pp. 368, 371, 375, 378).

– (2014). "Indirect Estimation of Service Demands in the Presence of Structural Changes". *Performance Evaluation*, 73. Elsevier Science: Amsterdam, The Netherlands, pp. 18–40 (cited on pp. 368, 371, 375, 378).

Kalbasi, A., Krishnamurthy, D., Rolia, J., and Dawson, S. (2012). "DEC: Service Demand Estimation with Confidence". *IEEE Transactions on Software Engineering*, 38(3). IEEE Computer Society: Washington, DC, USA, pp. 561–578 (cited on pp. 368, 373, 375, 377, 378).

Kalbasi, A., Krishnamurthy, D., Rolia, J., and Richter, M. (2011). "MODE: Mix Driven On-line Resource Demand Estimation". In: *Proceedings of the 7th International Conference on Network and Service Management (CNSM 2011)*. (Paris, France). IEEE: Piscataway, New Jersey, USA (cited on pp. 368, 372, 375, 378).

Kraft, S., Pacheco-Sanchez, S., Casale, G., and Dawson, S. (2009). "Estimating Service Resource Consumption from Response Time Measurements". In: *Proceedings of the 4th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2009)*. (Pisa, Italy). ICST/ACM, p. 48 (cited on pp. 368–370, 372, 375–378, 381).

Kumar, D., Tantawi, A. N., and Zhang, L. (2009). "Real-Time Performance Modeling for Adaptive Software Systems with Multi-Class Workload". In: *Proceedings of the 17th Annual Meeting of the IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*

(MASCOTS 2009). (London, UK). IEEE Computer Society: Washington, DC, USA, pp. 1–4 (cited on pp. 368, 370, 375, 378, 380, 381).

Kumar, D., Zhang, L., and Tantawi, A. N. (2009). "Enhanced Inferencing: Estimation of a Workload Dependent Performance Model". In: *Proceedings of the 4th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2009)*. (Pisa, Italy). ICST/ACM (cited on pp. 368, 370, 371, 375–379).

Lazowska, E. D., Zahorjan, J., Graham, G. S., and Sevcik, K. C. (1984). *Quantitative System Performance: Computer System Analysis Using Queueing Network Models*. Prentice-Hall: Upper Saddle River, NJ, USA (cited on pp. 368, 369, 375, 376, 378).

Liu, Z., Wynter, L., Xia, C. H., and Zhang, F. (2006). "Parameter Inference of Queueing Models for IT Systems using End-to-End Measurements". *Performance Evaluation*, 63(1). Elsevier Science: Amsterdam, The Netherlands, pp. 36–60 (cited on pp. 368, 370, 371, 375, 376, 378, 380–382).

Menascé, D. A. (2008). "Computing Missing Service Demand Parameters for Performance Models". In: *Proceedings of the 34th International Computer Measurement Group Conference (CMG 2008)*. (Las Vegas, Nevada, USA), pp. 241–248 (cited on pp. 368, 370, 371, 375, 376, 378, 381, 382).

Menascé, D. A., Almeida, V. A., and Dowdy, L. W. (2004). *Performance by Design: Computer Capacity Planning By Example*. Prentice Hall: Upper Saddle River, NJ, USA (cited on p. 369).

Nou, R., Kounev, S., Julià, F., and Torres, J. (2009). "Autonomic QoS Control in Enterprise Grid Environments using Online Simulation". *Journal of Systems and Software*, 82(3). Elsevier Science: Amsterdam, The Netherlands, pp. 486–502 (cited on pp. 368, 369, 375, 378).

Pacifici, G., Segmuller, W., Spreitzer, M., and Tantawi, A. N. (2008). "CPU Demand for Web Serving: Measurement Analysis and Dynamic Estimation". *Performance Evaluation*, 65(6-7). Elsevier Science: Amsterdam, The Netherlands, pp. 531–553 (cited on pp. 368–370, 375, 376, 378–381).

Pérez, J. F., Casale, G., and Pacheco-Sanchez, S. (2015). "Estimating Computational Requirements in Multi-Threaded Applications". *IEEE Transactions on Software Engineering*, 41(3). IEEE Computer Society: Washington, DC, USA, pp. 264–278 (cited on pp. 375, 378).

Pérez, J. F., Pacheco-Sanchez, S., and Casale, G. (2013). "An Offline Demand Estimation Method for Multi-threaded Applications". In: *Proceedings of the 2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2013)*. (San Francisco, CA, USA). IEEE Computer Society: Washington, DC, USA, pp. 21–30 (cited on pp. 368, 369, 372, 376, 378, 382).

Rolia, J., Kalbasi, A., Krishnamurthy, D., and Dawson, S. (2010). "Resource Demand Modeling for Multi-Tier Services". In: *Proceedings of the First Joint WOSP/SIPEW International Conference on Performance Engineering (ICPE 2010)*. (San Jose, CA, USA). ACM: New York, NY, USA, pp. 207–216 (cited on pp. 368, 373, 375, 377, 378, 381).

Rolia, J. and Vetland, V. (1995). "Parameter Estimation for Performance Models of Distributed Application Systems". In: *Proceedings of the 1995 Conference of the Centre for Advanced Studies on Collaborative Research (CASCON 1995)*. (Toronto, Ontario, Canada). IBM Press, p. 54 (cited on pp. 368–370, 375–378, 381).

Sharma, A. B., Bhagwan, R., Choudhury, M., Golubchik, L., Govindan, R., and Voelker, G. M. (2008). "Automatic Request Categorization in Internet Services". *SIGMETRICS Performance Evaluation Review*, 36(2). ACM: New York, NY, USA, pp. 16–25 (cited on pp. 368, 372, 375, 378).

Smola, A. J. and Schölkopf, B. (2004). "A Tutorial on Support Vector Regression". *Statistics and Computing*, 14(3). Kluwer Academic Publishers: Hingham, MA, USA, pp. 199–222 (cited on p. 372).

Spinner, S. (2017). "Self-Aware Resource Management in Virtualized Data Centers". PhD thesis. Würzburg, Germany: University of Würzburg (cited on pp. 366, 383).

Spinner, S., Casale, G., Brosig, F., and Kounev, S. (2015). "Evaluating Approaches to Resource Demand Estimation". *Performance Evaluation*, 92. Elsevier Science: Amsterdam, The Netherlands, pp. 51–71 (cited on pp. 366, 381, 382).

Stewart, C., Kelly, T., and Zhang, A. (2007). "Exploiting Nonstationarity for Performance Prediction". In: *ACM SIGOPS Operating Systems Review - Proceedings of the 2nd ACM SIGOPS EuroSys European Conference on Computer Systems*. (Lisbon, Portugal). Vol. 41. 3. ACM: New York, NY, USA, pp. 31–44 (cited on pp. 368–370, 375, 376, 380).

Sutton, C. and Jordan, M. I. (2011). "Bayesian Inference for Queueing Networks and Modeling of Internet Services". *The Annals of Applied Statistics*, 5(1). The Institute of Mathematical Statistics, pp. 254–282 (cited on pp. 368, 372, 375, 378).

Urgaonkar, B., Pacifici, G., Shenoy, P. J., Spreitzer, M., and Tantawi, A. N. (2007). "Analytic Modeling of Multitier Internet Applications". *ACM Transactions on the Web*, 1(1). ACM: New York, NY, USA (cited on pp. 368, 369, 375, 378).

Wang, W., Huang, X., Qin, X., Zhang, W., Wei, J., and Zhong, H. (2012). "Application-Level CPU Consumption Estimation: Towards Performance Isolation of Multi-tenancy Web Applications". In: *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing (CLOUD 2012)*. (Honolulu, HI, USA). IEEE Computer Society: Washington, DC, USA, pp. 439–446 (cited on pp. 368, 370, 375, 378, 381).

Wang, W., Huang, X., Song, Y., Zhang, W., Wei, J., Zhong, H., and Huang, T. (2011). "A Statistical Approach for Estimating CPU Consumption in Shared Java Middleware Server". In: *Proceedings of the 2011 IEEE 35th Annual Computer Software and Applications Conference (COMPSAC 2011)*. (Munich, Germany). IEEE Computer Society: Washington, DC, USA, pp. 541–546 (cited on p. 368).

Wang, W. and Casale, G. (2013). "Bayesian Service Demand Estimation Using Gibbs Sampling". In: *Proceedings of the 2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2013)*. IEEE Computer Society: Washington, DC, USA, pp. 567–576 (cited on pp. 368, 372, 373, 375, 376, 378).

Wynter, L., Xia, C. H., and Zhang, F. (2004). "Parameter Inference of Queueing Models for IT Systems using End-to-End Measurements". In: *Proceedings of the ACM SIGMETRICS International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS 2004)*. ACM: New York, NY, USA, pp. 408–409 (cited on pp. 368, 370, 375, 376, 378).

Zhang, L., Xia, C. H., Squillante, M. S., and Mills, W. N. (2002). "Workload Service Requirements Analysis: A Queueing Network Optimization Approach". In: *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS 2002)*. (Fort Worth, TX, USA). IEEE Computer Society: Washington, DC, USA, pp. 23–32 (cited on pp. 368, 375–378, 382).

Zhang, Q., Smirni, E., and Cherkasova, L. (2007). "A Regression-Based Analytic Model for Dynamic Resource Provisioning of Multi-Tier Applications". In: *Proceedings of the Fourth International Conference on Autonomic Computing (ICAC 2007)*. IEEE Computer Society: Washington, DC, USA, p. 27 (cited on pp. 368–370, 375, 378).

Zheng, T., Woodside, C. M., and Litoiu, M. (2008). "Performance Model Estimation and Tracking Using Optimal Filters". *IEEE Transactions on Software Engineering*, 34(3). IEEE Computer Society: Washington, DC, USA, pp. 391–406 (cited on pp. 368, 370, 375, 378, 380).