## Title Page

**TITLE**: Quality-of-Service in Cloud Computing: Modeling Techniques and Their Applications

**AUTHORS**:
Danilo Ardagna, Politecnico di Milano, Italy, ardagna@elet.polimi.it
Giuliano Casale, Imperial College London, UK, g.casale@imperial.ac.uk
Michele Ciavotta, Politecnico di Milano, Italy, ciavotta@elet.polimi.it
Juan F. Pérez, Imperial College London, UK, j.perez-bernal@imperial.ac.uk
Weikun Wang, Imperial College London, UK, weikun.wang11@imperial.ac.uk

**SUBMITTING AUTHOR**:
Giuliano Casale
Department of Computing
Imperial College London
180 Queens Gate, London, SW7 2AZ; UK
Office: 432
Phone: +44 20 759 42920 (Fax: 48932)

# Quality-of-Service in Cloud Computing: Modeling Techniques and Their Applications

**Danilo Ardagna, Giuliano Casale, Michele Ciavotta, Juan F. Pérez, Weikun Wang.**

the date of receipt and acceptance should be inserted later

**Abstract** Recent years have seen the massive migration of enterprise applications to the cloud. One of the challenges posed by cloud applications is Quality-of-Service (QoS) management, which is the problem of allocating resources to the application to guarantee a service level along dimensions such as performance, availability and reliability. This paper aims at supporting research in this area by providing a survey of the state of the art of QoS modeling approaches suitable for cloud systems. We also review and classify their early application to some decision-making problems arising in cloud QoS management.

**Keywords** Quality of service · Cloud computing · Modeling · QoS management

## 1 Introduction

Cloud computing has grown in popularity in recent years thanks to technical and economical benefits of the on-demand capacity management model [14]. Many cloud operators are now active on the market, providing a rich offering, including Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) solutions [140]. The cloud technology stack has also become mainstream in enterprise data centers, where private and hybrid cloud architectures are increasingly adopted [2].

Danilo Ardagna and Michele Ciavotta are with DEIB, Politecnico di Milano, Italy. Emails: {ardagna,ciavotta}@elet.polimi.it; Giuliano Casale, Juan F. Pérez and Weikun Wang are with the Dept. Computing, Imperial College London, UK. Emails: {g.casale,j.perez-bernal,weikun.wang11}@imperial.ac.uk

Address(es) of author(s) should be given

Even though the cloud has greatly simplified the capacity provisioning process, it poses several novel challenges in the area of Quality-of-Service (QoS) management. QoS denotes the levels of performance, reliability, and availability offered by an application and by the platform or infrastructure that hosts it[1]. QoS is fundamental for cloud users, who expect providers to deliver the advertised quality characteristics, and for cloud providers, who need to find the right tradeoffs between QoS levels and operational costs. However, finding optimal tradeoff is a difficult decision problem, often exacerbated by the presence of service level agreements (SLAs) specifying QoS targets and economical penalties associated to SLA violations [12].

While QoS properties have received constant attention well before the advent of cloud computing, performance heterogeneity and resource isolation mechanisms of cloud platforms have significantly complicated QoS analysis, prediction, and assurance. This is prompting several researchers to investigate automated QoS management methods that can leverage the high programmability of hardware and software resources in the cloud [100]. This paper aims at supporting these efforts by providing a survey of the state of the art of QoS modeling approaches applicable to cloud computing and by describing their initial application to cloud resource management.

*Scope.* Cloud computing is an operation model that integrates many technological advancements of the last decade such as virtualization, web services, and SLA management for enterprise applications. Characterizing cloud systems thus requires using diverse modeling techniques to cope with such technological heterogeneity. Yet, the QoS modeling literature is extensive, making it difficult to have a comprehensive view of the

available techniques and their current applications to cloud computing problems.

*Methodology.* The aim of this survey is to provide an overview of early research works in the cloud QoS modeling space, categorizing contributions according to relevant areas and methods used. Our methodology attempts to maximize coverage of works, as opposed to reviewing specific technical challenges or introducing readers to modeling techniques. In particular, we focus on recent modeling works published from 2006 onwards focusing on QoS in cloud systems. We also discuss some techniques originally developed for modeling and dynamic management in enterprise data centers that have been successively applied in the cloud context. Furthermore, the survey considers QoS modeling techniques for interactive cloud services, such as multi-tier applications. Works focusing on batch applications, such as those based on the MapReduce paradigm, are therefore not surveyed.

*Survey Organization.* This survey covers research efforts in *workload modeling*, *system modeling*, and their *applications* to QoS management in the cloud.

- *Workload modeling* involves the assessment or prediction of the arrival rates of requests and of the demand for resources (e.g., CPU requirements) placed by applications on an infrastructure or platform, and the QoS observed in response to such workloads. We review in Section 2 cloud measurement studies that help characterize those properties for specific cloud systems, followed by a review of workload characterizations and inference techniques that can be applied to QoS analysis.
- *System modeling* aims at evaluating the performance of a cloud system, either at design time or at runtime. Models are used to predict the value of specific QoS metrics such as response time, reliability or availability. We survey in Section 3 formalisms and tools employed for these analyses and their current applications to assess the performance of cloud systems.
- *Applications* of QoS models often appear in relation to decision-making problems in system management. Techniques to determine optimized decisions range from simple heuristics to nonlinear programming and meta-heuristics. We survey in Section 4 works on decision making for capacity allocation, load balancing, and admission control including research works that provide solutions for the management of a cloud infrastructure (i.e., from the cloud provider perspective) and resource management techniques for the infrastructure user (e.g., an application provider aiming at minimizing operational expenditure, while providing QoS level guarantees to the end users).

Section 5 concludes the paper and summarizes the key findings.

## 2 Cloud Workload Modeling

The definition of accurate workload models is essential to ensure good predictive capabaility for QoS models. Here, we survey workload characterization studies and related modeling techniques.

### 2.1 Workload Characterization

*Deployment environment.* Several studies have attempted to characterize the QoS showed by cloud deployment environments through benchmarking. Statistical characterizations of empirical data are useful in QoS modeling to quantify risks without the need to conduct an ad-hoc measurement study. They are vital to estimate realistic values for QoS model parameters, e.g., network bandwidth variance, virtual machine (VM) startup times, start failure probabilities. Observations of performance variability have been reported for different types of VM instances [44, 95, 106]. Hardware heterogeneity and VM interference are the primary cause for such variability, which is also visible within VMs of the same instance class. Other works characterize the variability in VM startup times [88, 106], which is correlated in particular with operating system image size [88]. Some studies on Amazon EC2 have found high-performance contention in CPU-bound jobs [136] and network performance overheads [123]. A few characterization studies specific to public and private PaaS hosting solutions also appeared in the literature [56, 80], together with comparisons of cloud database and storage services [40, 75, 122, 81]. Also, a comparison of different providers on a broad set of metrics is presented in [79].

*Cloud application workloads.* While the above works focus on describing the properties of the cloud deployment environment, users are often faced with the additional problem of describing the characteristics of the workloads processed by a cloud application.

Blackbox forecasting and trend analysis techniques are commonly used to predict web traffic intensity at different timescales. Time series forecasting has been extensively used for web servers for almost two decades. Autoregressive models in particular are quite common in applications and they are already exploited in cloud application modeling, e.g., for auto-scaling [105]. Other

common techniques include wavelet-based methods, regression analysis, filtering, Fourier analysis, and kernel-based methods [48].

Recent works in workload modeling that are relevant to cloud computing include [69,39,49]. [69] uses Hidden Markov Models to capture and predict temporal correlations between workloads of different compute clusters in the cloud. In this paper, the authors propose a method to characterize and predict workloads in cloud environments in order to efficiently provision cloud resources. The authors develop a co-clustering algorithm to find servers that have a similar workload pattern. The pattern is found by studying the performance correlations for applications on different servers. They use hidden Markov models to identify temporal correlations between different clusters and use this information to predict future workload variations. [39] defines a Bayesian algorithm for long-term workload prediction and pattern analysis, validating results on data from a Google data center. The authors define nine key features of the workload and use a Bayesian classifier to estimate the posterior probability of each feature. The experiments are based on a large dataset collected from a Google data center with thousands of machines. [49] applies pattern recognition techniques to data center and cloud workload data. The authors propose a workload demand prediction algorithm based on trend analysis and pattern recognition. This approach aims at finding a way to efficiently use the resource pool to allocate servers to different workloads. The pattern and trend are first analyzed and then synthetic workloads are created to reflect future behaviors of the workload. [144] uses a Kalman filter to model the interference caused when deploying applications on virtualized resources. The model accounts for time variations in VM resource usage, and it is used as the basis of a VM consolidation algorithm. The consolidation algorithm is tested and shown to be highly competitive. As the problem of workload modeling is far from trivial, [57] proposes a best practice guide to build empirical models. Important issues are treated, such as the selection of the most relevant data, the modeling technique, and variable-selection procedure. The authors also provide a comparative study that highlights the benefits of different forecasting approaches.

## 2.2 Workload Inference

The ability to quantify resource demands is a pre-requisite to parameterize most QoS models for enterprise applications. Inference is often justified by the overheads of deep monitoring and by the difficulty of tracking execution paths of individual requests [9]. Several works have investigated over the last two decades the problem of estimating, using indirect measurements, the resource demand placed by an application on physical resources, for example CPU requirements. From the perspective of cloud providers and users, inference techniques provide a means to estimate the workload profile of individual VMs running on their infrastructures, taking into account hidden variables due to lack of information.

*Regression Techniques.* A common workload inference approach involves estimating only the *mean* demand placed by a given type of requests on the resource [91, 103,104]. In [91] a standard model calibration technique is introduced. The technique is based on comparing the performance metrics (e.g., response time, throughput and resource utilization) predicted by a performance model against measurements collected in a controlled experimental environment. Given the lack of control over the system workload and configuration during operation, techniques of this type may not be applicable to production systems for online model calibration. These methods exploit queueing theory formulas to relate the mean values of a set of performance metrics (e.g., response times, throughputs, or resource utilizations) to a mean demand to be estimated, e.g., CPU demand. Regression techniques can exploit these formulas to obtain demand estimates from system measurements [82, 115,116,141,83].

[141] presents a queueing network model where each queue represents a tier of a web application, which is parameterized by means of a regression-based approximation of the CPU demand of customer transactions. It is shown that such an approximation is effective for modeling different types of workloads whose transaction mix changes over time.

[83] proposes instead service demand estimation from utilization and end-to-end response times: the problem is formulated as quadratic optimization programs based on queueing formulas; results are in good agreement with experimental data. Variants of these regression methods have been developed to cope with problems such as outliers [28], data multi-collinearity [65], online estimation [66], data aging [96], handling of multiple system configurations [35], and automatic definition of request types [34,109].

[65] proposes the Demand Estimation with Confidence (DEC) approach to overcome the problem of multicollinearity in regression methods. DEC can be iteratively applied to improve the estimation accuracy.

[35] proposes an algorithm to estimate the service demands for different system configurations. A time based linear clustering algorithm is used to identify different linear clusters for each service demands. This

approach proves to be robust to noisy data. Extensive validation on generated dataset and real data show the effectiveness of the algorithm.

[34] proposes a method based on clustering to estimate the service time. The authors employ density based clustering to obtain clusters of service times and CPU utilizations, and then use a cluster-wise regression algorithm to estimate the service time. A refinement process is conducted between clustering and regression to get accurate clustering results by removing outliers and merging the clusters that fit the same model. This approach proves to be computationally efficient and robust to outliers.

In [66] an on-line resource demand estimation approach is presented. An evaluation of regression techniques Least Squares (LSQ), Least Absolute Deviations (LAD) and Support Vector Regression (SVR) is presented. Experiments with different workloads show the importance of tuning the parameters, thus the authors proposes an online method to tune the regression parameters.

[28] presents an optimization-based inference technique that is formulated as a robust linear regression problem that can be used with both closed and open queueing network performance models. It uses aggregate measurements (i.e., system throughput and utilization of the servers), commonly retrieved from log files, in order to estimate service times.

[96] considers the problem of dynamically estimating CPU demands of diverse types of requests using CPU utilization and throughput measurements. The problem is formulated as a multivariate linear regression problem and accounts for multiple effects such as data aging. Also, several works have shown how combining the queueing theoretic formulas used by regression methods with the Kalman filter can enable continuous demand tracking [132, 143].

Regression techniques have also been used to correlate the CPU demand placed by a request on multiple servers. For example, linear regression of average utilization measurements against throughput can correctly account for the visit count of requests to each resource [141].

Stepwise linear regression [38] can also be used to identify request flows between application tiers. The knowledge of request flow intensities provides throughputs that can be used in regression techniques.

## 3 System Models

Workload modeling techniques presented in Section 2 are agnostic of the logic that governs a cloud system. Explicit modeling of this logic, or part of it, for QoS prediction can help improving the effectiveness of QoS management.

Several classes of models can be used to model QoS in cloud systems. Here we briefly review queueing models, Petri nets, and other specialized formalisms for reliability evaluation. However, several other classes exist such as stochastic process algebras, stochastic activity networks, stochastic reward nets [84], and models evaluated via probabilistic model checking [24]. A comparison of the pros and cons of some popular stochastic formalisms can be found in [31], where the authors highlight the issue that a given method can perform better on some system model but not on others, making it difficult to make absolute recommendations on the best model to use.

### 3.1 Performance Models

Among the performance models, we survey queueing systems, queueing networks, and layered queueing networks (LQN). While queueing systems are widely used to model single resources subject to contention, queueing networks are able to capture the interaction among a number of resources and/or applications components. LQNs are used to better model key interaction between application mechanisms, such as finite connection pools, admission control mechanisms, or synchronous request calls. Modeling these feature usually require an in-depth knowledge of the application behavior. On the other hand, while closed-form solutions exist for some classes of queueing systems and queueing networks, the solution of other models, including LQNs, rely on numerical methods.

*Queueing Systems.* Queueing theory is commonly used in system modeling to describe hardware or software resource contention. Several analytical formulas exist, for example to characterize request mean waiting times, or waiting buffer occupancy probabilities in single queueing systems. In cloud computing, analytical queueing formulas are often integrated in optimization programs, where they are repeatedly evaluated across what-if scenarios. Common analytical formulas involve queues with exponential service and arrival times, with a single server ($M/M/1$) or with $k$ servers ($M/M/k$), and queues with generally-distributed service times ($M/G/1$). Scheduling is often assumed to be first-come first-served (FCFS) or processor sharing (PS). In particular, the $M/G/1$ PS queue is a common abstraction used to model a CPU and it has been adopted in many cloud studies [11] [135], thanks to its simplicity and the suitability to apply the model to multi-class workloads. For

instance, an SLA-aware capacity allocation mechanism for cloud applications is derived in [11] using an $M/G/1$ PS queue as the QoS model. In [135] the authors propose a resource provisioning approach of N-tier cloud web applications by modeling CPU as an $M/G/1$ PS queue. The $M/M/1$ open queue with FCFS scheduling has been used [7,51,142] to pose constraints on the mean response time of a cloud application. Heterogeneity in customer SLAs is handled in [42] with an $M/M/k/k$ *priority* queue, which is a queue with exponentially distributed inter-arrival times and service times, $k$ servers and no buffer. The authors use this model to investigate rejection probabilities and help dimensioning of cloud data centers. Other works that rely on queueing models to describe cloud resources include [76,77]. The works in [76,77] illustrate the formulation of basic queueing systems in the context of discrete-time control problems for cloud applications, where system properties such as arrival rates can change in time at discrete instants. These works show an example where a non-stationary cloud system is modeled through queueing theory.

*Queueing Networks.* A queueing network can be described as a collection of queues interacting through request arrivals and departures. Each queue represents either a physical resource (e.g., CPU, network bandwidth, etc) or a software buffer (e.g., admission control, or connection pools). Cloud applications are often tiered and queueing networks can capture the interactions between tiers. An example of cloud management solutions exploiting queueing network models is [1], where the cloud service center is modeled as an open queueing network of multiclass single-server queues. PS scheduling is assumed at the resources to model CPU sharing. Each layer of queues represents the collection of applications supporting the execution of requests at each tier of the cloud service center. This model is used to provide performance guarantees when defining resource allocation policies in a cloud platform. Also, [50] uses a queueing network to represent a multi-tier application deployed in a cloud platform, and to derive an SLA-aware resource allocation policy. Each node in the network has exponential processing times and a generalized PS policy to approximate the operating system scheduling.

*Layered Queueing Networks.* Layered queueing networks (LQNs) are an extension of queueing networks to describe layered software architectures. An LQN model of an application can be built automatically from software engineering models expressed using formalisms such as UML or Palladio Component Models (PCM) [16]. Compared to ordinary queueing networks, LQNs provide the ability to describe dependencies arising in a complex workflow of requests and the layering among hardware and software resources that process them. Several evaluation techniques exist for LQNs [47,93,119,99].

LQNs have been applied to cloud systems in [43], where the authors explored the impact of the network latency on the system response time for different system deployments. LQNs are here useful to handle the complexity of geo-distributed applications that include both transactional and streaming workloads.

[64] uses an LQN model to predict the performance of the RuBis benchmark application, which is then used as the basis of an optimization algorithm that aims at determining the best replication levels and placement of the application components. While this work is not specific to the cloud, it illustrates the application of LQNs to multi-tier applications that are commonly deployed in such environments.

[15] investigates a prediction-based cloud resource allocation and management algorithm. LQNs are used to predict the performance of an enterprise application deployed on the cloud with strict SLA requirements based on historical data. The authors also provide a discussion about the pros and cons of LQNs identifying a number of key limitations for their practical use in cloud systems. These include, among others, difficulties in modeling caching, lack of methods to compute percentiles of response times, tradeoff between accuracy and speed. Since then, evaluation techniques for LQNs that allow the computation of response time percentiles have been presented [99].

*Hybrid models.* Queueing models are also used together with machine learning techniques to achieve the benefits of both approaches. Queueing models use the knowledge of the system topology and infrastructure to provide accurate performance predictions. However, a violation of the model assumptions, such as an unforeseen change in the topology, can invalidate the model predictions. Machine learning algorithms, instead, are more robust with respect to dynamic changes of the system. The drawback is that they adopt a black-box approach, ignoring relevant knowledge of the system that could provide valuable insights into its performance.

[38] studies the relations between workload and resource consumption for cloud web applications. Queueing theory is used to model different components of the system and data mining and machine learning approaches ensure dynamic adaptation of the model to work under system fluctuations. The proposed approach is shown to achieve high accuracy for predicting workload and resource usages.

[118] proposes a robust performance model architecture focusing on analyzing performance anomalies and localizing the potential source of the discrepancies. The performance models are based on queueing-network models abstracted from the system and enhanced by machine learning algorithms to correlate system workload attributes with performance attributes.

A queueing network approach is taken in [110] to provision resources for data-center applications. As the workload mix is observed to fluctuate over time, the queueing model is enhanced with a clustering algorithm that determines the workload mix. The approach is shown to reduce SLA violations due to under-provisioning in applications subject to to non-stationary workloads.

### 3.2 Dependability Models

Petri nets, Reliability Block Diagrams (RBD), and Fault Trees are probably the most widely known and used formalisms for dependability analysis. Petri nets are a flexible and expressive modeling approach, which allows a general interactions between system components, including synchronization of event firing times. They also find large application also in performance analysis.

RBDs and Fault Trees aim at obtaining the overall system reliability from the reliability of the system components. The interactions between the components focus on how the faulty state of one or more components results in the possible failure of another components.

*Petri nets.* It has long been recognized the suitability of Petri nets for performance and dependability of computer systems. Petri nets have been extended to consider stochastic transitions, in stochastic Petri nets (SPNs) and generalized SPNs (GSPNs). They have recently enjoyed a resurgence of interest in service-oriented systems to describe service orchestrations [19].

In the context of cloud computing, we have more application examples of Petri nets nets for dependability assessment, than for performance modeling. Applications to cloud QoS modeling include the use of SPNs to evaluate the dependability of a cloud infrastructure [25], considering both reliability and availability. SPNs provide a convenient way in this setting to represent energy flow and cooling in the infrastructure. [127] proposes the use of GSPNs to evaluate the impact of virtualization mechanisms, such as VM consolidation and live migration, on cloud infrastructure dependability. GSPNs are used to provide fine-grained detail on the inner VM behaviors, such as separation of privileged and non-privileged instructions and successive handling by the VM or the VM monitor. Petri nets are here used in combination with other methods, i.e., Reliability Block Diagrams and Fault Trees, for analyzing mean time to failure (MTTF) and mean time between failures (MTBF).

*Reliability Block Diagrams.* Reliability block diagrams (RBDs) are a popular tool for reliability analysis of complex systems. The system is represented by a set of inter-related blocks, connected by series, parallel, and $k$-out-of-$N$ relationships.

In [45], the authors propose a methodology to evaluate data center power infrastructures considering both reliability and cost. RBDs are used to estimate and enforce system reliability. [36] investigates the benefits of a warm-standby replication mechanism in Eucalyptus cloud computing environments. An RBD is used to evaluate the impact of a redundant cloud architecture on its dependability. A case study shows how the redundant system obtains dependability improvements. [89] uses RBDs to design a rejuvenation mechanism based on live migration, to prevent performance degradation, for a cloud application that has high availability requirements.

*Fault Trees.* Fault Trees are another formalism for reliability analysis. The system is represented as a tree of inter-related components. If a component fails, it assumes the logical value *true*, and the failure propagation can be studied via the tree structure. In cloud computing, Fault Trees have been used to evaluate dependencies of cloud services and their effect on application reliability [46]. Fault Trees and Markov models are used to evaluate the reliability and availability of fault tolerance mechanisms. [61] uses Fault Trees and Markov models to evaluate the reliability and availability of a cloud system under different deployment contexts. Based on this evaluation, the authors propose an approach to identify the best mechanisms according to user's requirements. [71] presents a methodology to identify, mitigate, and monitor risks in cloud resource provisioning. Fault Trees are used to assess the probability of SLA violations.

### 3.3 Black-box Service Models

Service models have been used primarily in optimising web service composition [13], but they are now becoming relevant also in the description of SaaS applications, IaaS resource orchestration, and cloud-based business-process execution. The idea behind the methods reviewed in this section is to describe a service in terms of its response time, assuming the lack of any further information concerning its internal characteristics (e.g., contention level from concurrent requests).

Non-parametric blackbox service models include methods based on deterministic or average execution time values [85,8,52,27,62]. Several works instead adopt a description that includes standard deviations [13,138, 113] or finite ranges of variability for the execution times [33,107]. Parametric service models instead assume exponential or Markovian distributions [32,102], Pareto distributions to capture heavy-tailed execution times [53], or general distributions with Laplace transforms [90].

[59] presents a graph-theoretic model for QoS-aware service composition in cloud platforms, explicitly handling network virtualization. Here, the authors explore the QoS-aware service provisioning in cloud platforms by explicitly considering virtual network services. A system model is demonstrated to suitably characterize cloud service provisioning behavior and an exact algorithm is proposed to optimize users' experience under QoS requirements. A comparison with state of the art QoS routing algorithms shows that the proposed algorithm is both cost-effective and lightweight.

[72] considers QoS-aware service composition by handling network latencies. The authors present a network model that allows estimating latencies between locations and propose a genetic algorithm to achieve network-aware and QoS-aware service provisioning.

The work in [137] considers cloud service provisioning from the point of view of an end user. An economic model based on discrete Bayesian Networks is presented to characterize end-users long-term behavior. Then the QoS-aware service composition is solved by Influence Diagrams followed by analytical and simulation experiments.

### 3.4 Simulation Models

Several simulation packages exist for cloud system simulation. Many solutions are based on the CloudSim [23] toolkit that allows the user to set up a simulation model that explicitly considers virtualized cloud resources, potentially located in different data centers, as in the case of hybrid deployments. CloudAnalyst [129] is an extension of CloudSim that allows the modeling of geographically-distributed workloads served by applications deployed on a number of virtualized data centers.

EMUSIM [22] builds on top of CloudSim by adding an emulation step leveraging the Automated Emulation Framework (AEF) [21]. Emulation is used to understand the application behavior, extracting profiling information. This information is then used as input for CloudSim, which provides QoS estimates for a given cloud deployment.

Some other tools have been developed to estimate data center energy consumption. For example, GreenCloud [73], which is an extension of the packet-level simulator NS2[2], aims at evaluating the energy consumption of the data center resources where the application has been deployed, considering servers, links, and switches.

Similarly, DCSim [67] is a data center simulation tool focused on dynamic resource management of IaaS infrastructures. Each host can run several VMs, and has a power model to determine the overall data center power consumption.

GroudSim [94] is a simulator for scientific applications deployed on large-scale clouds and grids. The simulator is based on events rather than on processes, making it a scalable solution for highly parallelized applications.

## 4 Applications

A prominent application of QoS models is optimal decision-making for cloud system management. Problem areas covered in this section include capacity allocation, load balancing, and admission control. Several other relevant decision problems exist in cloud computing, e.g., pricing [124], resource bidding [111], and provider-side energy management [17].

We classify works in three areas using, for comparability, a taxonomy similar to the one appearing in the software engineering survey of Aleti et al. [5], which also covers design-time optimization studies, but does not focus on cloud computing. Our classification dimensions follow from these questions:

*Perspective*: is the study focusing on the perspective of the infrastructure user or or on the perspective of the provider?
*Dimensionality*: is the study optimizing a single or multiple objective functions?
*Solution*: is the presented solution centralized or distributed?
*Strategy*: is the optimization problem tackled by an exact or approximate technique?
*Time-scale*: is the time-scale for the performed adaptations, which can be short (seconds), medium (minutes), or long (hours/days)?
*Discipline*: is the management approach based on control theory, machine learning or operations research (i.e., optimization, game theory, bio-inspired algorithms)?

Table 1 provides a taxonomy of the papers reviewed in the next sections, organized according to the above

criteria. Few remarks are needed to clarify the methodology used to classify the papers:

- In the *Perspective* dimension, a public PaaS or SaaS service built on top of a public IaaS offering is classified as a user-side perspective.
- Under *Dimensionality*, we treat studies that weight multiple criteria into a single objective as Multi-Objective methods.

Finally, the following observations on the *Discipline* dimensions must also be made.

- *Control theory* has the advantage of guaranteeing the stability of the system upon workload changes by modeling the transient behavior and adjusting system configurations within a transitory period [97].
- *Machine learning* techniques, instead, use learning mechanisms to capture the behavior of the system without any explicit performance or traffic model and with little built-in system knowledge. Nevertheless, training sessions tend to extend over several hours [68] and retraining is required for evolving workloads.
- *Operations research* approaches are designed with the aim of optimizing the degree of user satisfaction. The goals, in fact, are expressed in terms of user-level QoS metrics. Typically, this approach consists of a performance model embedded within an optimization program, which is solved either globally, locally, or heuristically.

### 4.1 Capacity Allocation

*Infrastructure-Provider Capacity Allocation*

The capacity allocation problem arising at the provider side involves deciding the optimal placement of running applications on a suitable number of VMs, which in turn has to be executed on appropriate physical servers. The rationale is to assign resource shares trying to minimize management costs (formed mainly by costs associated with energy consumption), while guaranteeing the fulfilment of SLAs stipulated with the customers. Bin packing is a common modeling abstraction [142], but its NP-hardness calls for heuristic solutions. In [54] the capacity allocation problem is solved by means of a dynamic algorithm, since static allocation policies and pricing usually lead to inefficient resource sharing, poor utilization, waste of resources and revenue loss when demands and workloads are time varying. The paper presents a Minimum Cost Maximum Flow (MCMF) algorithm and compares it against a modified Bin-Packing formulation; the MCMF algorithm exhibits very good performance and scalability properties. An autoregressive process is used to predict the fluctuating incoming demand.

In [133], autoscaling is modeled as a modified Class Constrained Bin Packing problem. An auto-scaling algorithm is provided that automatically classifies incoming requests. Moreover, it improves the placement of application instances by putting idle machines into standby mode and reducing the number of running instances in condition of light load.

[117] proposes a fast heuristic solution for VM placement over a very large number of servers in a IaaS data center to equally balance the CPU load among physical machines, taking into account also memory requirements of running applications.

In [30] is proposed a framework for VM deployment and reconfiguration optimization, with the aim at increasing profits of IaaS providers. The authors reduce costs considering the balance of multi-dimensional resources utilization and building up an optimization method for resource allocation; as far as reconfiguration is concerned, they propose a strategy for VM adjustment based on time-division multiplex and on VM live migration.

In [1] a VM placement problem for a PaaS is solved at multiple time-scales through a hierarchical optimization framework. Authors in [92] provide a solution for traffic-aware VM placement minimizing also network latencies among deployed applications. The work presents a two-tier approximate algorithm able to successfully solve very large problem instances. Moreover, a formulation for the considered problem is presented and its hardness is proven.

A capacity allocation problem is also studied in [128], in which a game-theoretic method is used to find approximated solutions for a resource allocation problem in the presence of tasks with multiple dependent subtasks. The initial solution is spawned by a binary integer programming method; then, evolutionary algorithms are designed to achieve a final optimized solution, which minimizes efficiency losses of participants.

[50] provides a solution method for a multi-dimensional capacity allocation problem in a IaaS system, while guaranteeing SLA requirements to customers running multi-tiers applications. An improved solution is obtained starting from an initial configuration based on an upper bound; then, a force-directed search is adopted to increase the total profit. Moreover, a closed-form formula for calculating the average response time of a request and a unified framework to manage different levels of SLAs are provided.

[139] considers an online mechanism for computing resource allocation to VMs subject to limited informa-

*Table 1:* Decision-Making in the Cloud - a Taxonomy

| *Category* | *Value* | Application | | |
|---|---|---|---|---|
| | | *Capacity Allocation* | *Admission Control* | *Load Balancing* |
| Perspective | Infrastructure user | [11] [18] [20] [26] [29] [63] [86] [87] [98] [108] [6] [114] [121] [131] | [78] [121] [130] | [10] [11] [18] [26] [125] [3] [101] |
| | Infrastructure provider | [1] [4] [7] [30] [41] [42] [50] [54] [55] [60] [70] [74] [77] [92] [105] [117] [120] [126] [128] [133] [139] [142] [144] | [7] [74] [70] [37] [134] [4] [42] | [51] [112] [92] [120] |
| Dimensionality | Single-Objective | [1] [11] [20] [92] [7] [18] [29] [41] [50] [54] [55] [63] [70] [86] [87] [98] [108] [6] [114] [120] [121] [126] [128] [139] [142] [144] [60] | [7] [121] [70] [130] [78] [37] [134] | [10] [3] [18] [92] [120] [11] [101] [112] [51] |
| | Multi-Objective | [30] [42] [133] [117] [4] [26] [74] [105] [131] | [42] [4] [74] | [125] [26] |
| Solution | Centralized | [4] [7] [29] [30] [41] [50] [92] [54] [55] [70] [74] [86] [87] [98] [105] [117] [108] [6] [114] [120] [121] [126] [133] [139] [42] [128] [144] [60] [142] [131] | [4] [7] [58] [42] [74] [121] [70] [130] [78] [37] [134] | [10] [3] [51] [92] [125] [112] [120] |
| | Decentralized | [11] [18] [20] [63] [26] [1] [77] | | [11] [18] [26] [101] |
| Strategy | Exact | [42] [74] [121] [29] [7] [108] [6] [114] [142] | [42] [7] [58] [74] [121] | [3] |
| | Approximate | [11] [18] [92] [4] [20] [70] [63] [29] [144] [60] [30] [41] [50] [86] [87] [98] [105] [139] [117] [126] [120] [133] [54] [26] [1] [77] [128] [131] | [4] [70] [130] [78] [37] [134] | [51] [10] [18] [92] [11] [26] [101] [125] [112] [120] |
| Timescale | Short | [11] [18] [42] [55] [77] [108] [114] | [58] [7] [42] [37] [134] | [10] [3] [11] [18] [112] |
| | Medium | [4] [7] [11] [92] [42] [74] [63] [29] [30] [41] [50] [60] [86] [87] [98] [105] [139] [20] [117] [120] [133] [54] [26] [1] [77] [70] [128] [142] [144] [6] [131] | [4] [74] [42] [70] [130] | [11] [92] [26] [101] [120] [125] |
| | Long | [121] [29] [87] [126] [1] | [121] [78] | [51] |
| Discipline | Control Theory | [77] [41] [86] [98] [6] | | |
| | Machine Learning | [86] [120] [144] | [134] | [120] |
| | Operations Research | [4] [7] [11] [42] [55] [70] [74] [121] [63] [29] [92] [30] [50] [60] [86] [87] [105] [20] [128] [139] [117] [126] [133] [54] [26] [1] [114] [108] [131] [142] | [4] [7] [58] [42] [130] [37] [121] [74] [70] [112] | [10] [3] [92] [11] [26] [101] [125] [51] |

tion. The algorithm evaluates allocation and revenues as the users place requests to the system. Furthermore, the authors prove that their approach is incentive compatible; they also report extensive simulation experiments.

[126] considers capacity allocation subject to two pricing models, a pay-as-you-go offering and periodic auctions. An optimal capacity segmentation strategy is formulated as a Markov decision process, which is then used to maximize revenues. The authors propose also a faster near-optimal algorithm, proven to asymptotically approach the optimal solution, and show a significantly lower complexity with respect to the optimal method.

[105] proposes a model-predictive resource allocation algorithm that auto-scales VMs, with the aim of optimizing the utility of the application over a limited prediction horizon. Empirical results demonstrate that the proposed method satisfies application QoS requirements, while minimizing operational costs.

[41] develops a resource manager that uses a combination of horizontal and vertical scaling to optimize both resource usage and the reconfiguration cost. Finally, the solution is tested using real production traces.

[144] builds a VM consolidation algorithm that makes use of an inference model that considers the effect of co-located VMs to predict QoS metrics. In this method, the workload is modeled by means of a Kalman filter, while the resource usage profile is estimated with a Hidden Markov Model. The proposed method is tested against SPECWeb2005.

[60] also considers the VM consolidation problem by modeling the VM resource demands as a set of correlated random variables. The result is a multi-capacity stochastic bin packing problem, which is solved by means of a simple, scalable yet effective heuristic.

[55] uses a multivariate probabilistic model to schedule VMs among physical machines in order to improve resource utilization. This approach also considers migration costs, and the multi-dimensional nature of the VM resource requirements (e.g., CPU, memory, and network).

Finally, in [120] a framework that automatically reconfigures the storage system in response to fluctuations in the workload is presented. The framework makes use of a performance model of the system obtained through statistical machine learning. Such model is embedded into an effective Model-Predictive Control algorithm.

*Infrastructure-User Capacity Allocation*

From the user perspective, capacity allocation arises in IaaS and PaaS scenarios where the user is in charge with the control of the number of VMs or application containers running in the system. In this context the user is generally a SaaS provider, which wants to maximize her revenues providing a service that meets a certain QoS. Then, the problem to be addressed is to determine the minimum number of VMs or containers needed to fulfill the target QoS, pursuing the best trade-off between cost and performance.

From the user side, capacity allocation is often implemented through auto-scaling policies. [87] defines an auto-scaling mechanism to guarantee the execution of all jobs within given deadlines. The solution accounts for workload burstinesses and delayed instance acquisition. This approach is compared against other techniques and it shows cost savings from 9.8% to 40.4%. [86] compares several approaches for decision-making, as part of an autonomic framework that allocates resources to a software application.

[98] proposes a multi-model control-based framework to deal with the highly nonlinear nature of software systems. An extensible meta-model and a class library with an initial set of five models are developed. Finally, the presented approach is endorsed against fixed and adaptive control schemes by means of a campaign of experiments.

In [29] an optimal resource provisioning algorithm is derived to deal with the uncertainty of resource advance-reservation. The algorithm reduces resources under- and over-provisioning by minimizing the total cost for a customer during a certain time horizon. The solution methods are based on the Bender decomposition approach to divide the problem into sub-problems, which can be solved in parallel, and an approximation algorithm to solve problems with a large set of scenarios.

On-demand and reserved resources are considered in the model proposed in [26] to define a bio-inspired self-adapting solution for cloud resource provisioning with the aim of minimizing the number of required virtual machines while meeting SLAs.

A decentralized probabilistic algorithm is also described in [20], which focuses on federated clouds. The proposed solution has the aim to take advantage of a Cloud federation to avoid the dependence on a single provider, while still minimizing the amount of used resources to maintain a good QoS level for customers. The solution provides an effective decentralized algorithm for deploying massively scalable services and it is suitable for all the situations in which a centralized solution is not feasible.

[6] aims at dynamic resource provisioning exploiting horizontal elasticity. Two adaptive hybrid controllers, including both reactive and proactive actions, are employed to decide the number of VMs for a cloud service to meet the SLAs. The future demand is predicted by a queueing-network model.

A key-value store is presented in [114] to meet low-latency Service Level Objectives (SLOs). The proposed middleware achieves high scalability by using replication, providing more predictable response times. An analytical model, based on queueing theory, is presented to describe the relation between the number of replicas and the service level, e.g., the percentage of requests processed according to SLOs.

A capacity allocation problem in presented in [108] that exploits both horizontal and vertical elasticity. An integer linear problem is used to calculate an optimized new configuration able to deal with the current workload. However, reconfiguration is executed if the associated overhead cost calculated on a expected stability duration is lower than a certain minimum benefit defined by a human decision maker.

In [131] two multi-objective customer-driven SLA-based resource provisioning algorithms are proposed. The objectives are the minimization of both resource and penalty costs, as well as minimizing SLA violations. The proposed algorithms consider customer profiles and quality parameters to cope with dynamic workloads and heterogeneous cloud resources.

Finally, a profile-based approach for scalability is described in [63], the authors propose a solution based on the definition of platform-independent profiles, which enable the automation of setup and scaling of application servers in order to achieve a just-in-time scalability of the execution environment, as demonstrated with a case study presented in the paper.

## 4.2 Load Balancing

*Infrastructure-Provider Load Balancing*

Request load-balancing is an increasingly supported feature of cloud offerings. A load balancer dispatches requests from users to servers according to a load dispatching policy. Policies differ for the decision approach and for the amount of information they use. Research work has focused on policies that are either simple to implement, and thus minimize overheads, or that offer some optimality guarantees, typically proven by analytical models.

The research literature has investigated both centralized and decentralized load balancing mechanisms for providers.

Among centralized approaches, [3] introduces an offline optimization problem for geographical load balancing among data centers, explicitly considering SLAs and dynamic electricity prices. This is complemented with an online algorithm to handle the uncertainty in electricity prices. The proposed algorithm is compared against a greedy heuristic method and it shows significant cost savings (around 20-30%).

A load balancer is presented in [51] to assign VMs among geographically-distributed data centers considering predictions on workload, energy prices, and renewable energy generation capacities. Two complementary methods are proposed: an offline deterministic optimization method to be used at design time and an online VM placement, migration and geographical load balancing algorithm for runtime. The authors studied the behavior of both online and offline algorithms by means of a simulation campaign. The results demonstrate that online version of the algorithm performs 8% worse than the offline one because it deals with incomplete information. On the other hand, the analysis also shows that turning on the geographical load balancing has a strong impact on quality of the solutions (between 27% and 40%) of the online algorithm.

[112] proposes an online load balancing policy that considers the inherent VM heterogeneity found in cloud resources. The load balancer uses the number of outstanding requests and the inter-departure times in each VM to dispatch requests to the VM with the shortest expected response time. The authors demonstrate that their solution is able to improve the variance and percentiles of response times with respect to a built-in policy of the Apache web server.

Decentralized methods are considered in [26], which proposes a self-organizing approach to provide robust and scalable solutions for service deployment, resource provisioning, and load balancing in a cloud infrastructure. The algorithm developed has the additional benefit to leverage Cloud elasticity to allocate and deallocate resources to help services to respect contractual SLAs.

Another example is the cost minimization mechanism for data-intensive service provisioning proposed in [125]. Such mechanism uses biological evolution concepts to manage data application services and to produce optimal composition and load balancing solutions. A multi-objective genetic algorithm is described in detail but a systematic experimental campaign is planned as future work.

*Infrastructure-User Load Balancing*

In the studies considered in the previous section, the load balancer is installed and managed transparently by the cloud provider. In some cases, the user can decide to install its own load balancer for a cloud application. This may be helpful, for instance, to jointly tackle capacity allocation and load balancing.

For example, [11] considers a joint optimization problem on multiple IaaS service centers. A non-linear model for the capacity allocation and load redirection of multiple request classes is proposed and solved by decomposition. A comparison against a set of heuristics from the literature and an oracle with perfect knowledge about the future load shows that the proposed algorithm overcomes the heuristic approaches, without penalizing SLAs and it is able to produce results that are close to the global optimum. [10] provides a simple heuristic for user-side load-balancing under connection pooling that is validated against an IaaS cloud dataset. The main result is that the presented approach is able to provide tight guarantees on the optimality gap and experimental results show that it is at the same time accurate and fast.

Hybrid clouds are considered in [121]. The authors formulate an optimization problem faced by a cloud

procurement endpoint (a module responsible for provisioning resources from public cloud providers), where heavy workloads are tackled by relying on public clouds. They present a linear integer program to minimize the resource cost, and evaluate how the solution scales with the different problem parameters.

In [101] a structured peer-to-peer network, based on distributed hash tables, is proposed to support service discovery, self-management, and load-balancing of cloud applications. The effectiveness of the peer-to-peer approach is demonstrated through a set of experiments executed on Amazon EC2.

Finally, [18] proposes an adaptive approach for component replication of cloud applications, aiming at finding a cost-effective placement and load balancing. This is a distributed method based on an economic multi-agent model that achieves high application availability guaranteeing at the same time service availability under failures.

### 4.3 Admission Control

*Infrastructure-Provider Admission Control*

Admission control is an overload protection mechanism that rejects requests under peak workload conditions to prevent QoS degradation. A lot of work has been done in the last decade for optimal admission control in web servers and multi-tier applications. The basic idea is to predict the value of a specific QoS metric and if such value grows above a certain threshold, the admission controller rejects all new sessions favoring the service of requests from already admitted sessions.

In cloud computing, several works on admission control have emerged in IaaS. [70] develops an analytical model for resource provisioning, virtual machine deployment, and pool management. This model predicts service delay, task rejection probability, and steady-state distribution of server pools.

The availability of resources and admission control is also discussed in [74]. The work uses a probabilistic approach to find an optimized allocation of services on virtualized physical resources. The main requirement of this system is the horizontal elasticity. In fact, the probability of requesting more resources for a service is at the basis of the formulated optimization model, that constitutes a probabilistic admission control test.

[7] proposes a joint admission control and capacity allocation algorithm for virtualized IaaS systems minimizing the data center energy costs and the penalty incurred for request rejections and SLA violations. SLAs are expressed in terms of the tail distribution of application response times.

[4] optimizes the allocation and scheduling of VMs in federated clouds using a genetic algorithm. The solution is composed by two parts: First, servers selection in a data-center is performed by using a search based bio-inspired technique; then, data centers are selected within the cloud federation by using a shortest path algorithm, according to the available bandwidth of links connecting the domains. The aim of the paper is to exploit resources in domains with low allocation costs and, at the same time, achieve better network performance among cloud nodes.

[42] allows service providers to reserve a certain amount of resources exclusively for some customers, according to SLAs. The proposed framework helps to stipulate a realistic SLA with customers and supports dynamic load shedding and capacity provisioning by considering a queueing model with multiple priority classes. The main performance metric being optimized is the rejection probability, which has to guarantee the value stipulated in the SLA.

The work in [37] proposes an admission control protocol to prevent over-utilization of system resources, classifying applications based on resource quality requirements. It uses an open multi-class queueing network to support a QoS-aware admission control on heterogeneous resources to increase system throughput.

In order to control overload in Database-as-a-Service (DaaS) environments, [134] proposes a profit-aware admission control policy. It first uses nonlinear regression to predict the probability for a query to meet its requirement, and then decides whether the query should be admitted to the database system or not.

*Infrastructure-User Admission Control*

From the cloud-user perspective, the admission control mechanism is used as an extreme overload mechanism, helpful when additional resources are obtained with some significant delay. For example, during a cloud burst (i.e., when part of the application traffic is redirected from a private to a public data center to cope with a traffic intensity that surpasses the capacity of the private infrastructure), if the public cloud resources are not provided timely, one can decide to drop new incoming request to preserve the QoS for users already in the system (or at least part of them, e.g., *gold customers*), avoiding application performance degradation.

Three different admission control and scheduling algorithms are proposed in [130] to effectively exploiting public cloud resources. The paper takes the perspective of a SaaS provider with the aim of maximizing the profit by minimizing cost and improving customer satisfaction levels.

[78] introduces a client-side admission control method to schedule requests among VMs, looking at minimizing the cost of application, SLA violations and IaaS resources.

## 5 Discussion and Conclusion

In recent years, cloud computing has matured from an early-stage solution to a mainstream operational model for enterprise applications. However, the diversity of technologies used in cloud systems makes it difficult to analyze their QoS and, from the provider perspective, to offer service-level guarantees. We have surveyed current approaches in workload and system modeling and early applications to cloud QoS management.

From this survey, a number of insights arise on the current state of the art:

- The number of works that apply white-box system modeling techniques is quite limited in QoS management, albeit popular in the software performance engineering community. This effectively creates a divide between the knowledge that can be made available for an application by its designers and the techniques used to manage it. A research question is whether the availability of more detailed information about application internals can provide significant advantages in QoS management. Indeed, a trade-off exists between available information, QoS model complexity, computational cost of decision-making, and accuracy of predictions. This trade-off requires further investigation by the research community.
- Gray-box models that emphasize resource consumption modeling are currently prevalent in QoS management studies. However, description of performance is often quite basic and associated with mean resource requirements of the applications. However, the cloud measurement studies in Section 2.1 have identified performance variability as a major issue in today's offerings, calling for more comprehensive models that can describe also the variability in CPU requirements, in addition to mean requirements. Such extension has been explored in black-box system models (e.g., QoS in web services), but it is far less understood in white-box and gray-box modeling.
- Quite surprisingly, we have found a limited amount of work specific to workload analysis and inference techniques in the cloud. Most of the techniques used for traffic forecasting, resource consumption estimation, and anomaly detection have received little or no validation in a cloud environment. As such, it remains to establish the robustness of current techniques to noisy measurements typical of multi-tenant cloud environments.
- Another observation arising from our survey is that the literature is rich in works focusing on IaaS systems, often deployed on Amazon EC2, at present the market leader in this segment.
- If we consider the resource management mechanisms for applications QoS enforcement provided by public clouds, they are quite simplistic if compared to current research proposals. Indeed, such mechanisms are mainly *reactive* and are triggered by thresholds violations (related to response times, as in Google App Engine, or CPU utilization or other low level infrastructure metrics, as in Amazon EC2.) Vice versa, integrating workload characterisation, system models and resource management solutions, *pro-active* systems, may help to prevent QoS degradation. The development of research prototypes that are transferable in commercial solutions seems to remain an open point.
- Finally, in cloud systems an important role is played by resource pricing models. There is a growing interest towards understanding better cloud spot markets, where bidding strategies are developed for procuring computing resources. Approaches are currently being proposed to automate dynamic pricing and cloud resources selection. We expect that, in upcoming years, these models will play a bigger role than today in capacity allocation frameworks.

Summarizing, this survey shows that the literature has already a significant number of works in cloud QoS management, but their focus leaves open several research opportunities in the areas discussed above.

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

All authors read and approved the final manuscript.

### Acknowledgment

The research reported in this article is partially supported by the European Commission grant FP7-ICT-2011-8-318484 (MODAClouds, www.modaclouds.eu).

## Endnote

[1]Throughout this paper, we mainly focus on QoS aspects pertaining to performance, reliability and availability. Broader descriptions of QoS are possible (e.g., to include security) but they are not contemplated in the present survey.

[2]`http://www.isi.edu/nsnam/ns/`

## References


1. B. Addis, D. Ardagna, B. Panicucci, M. S. Squillante, and L. Zhang. A hierarchical approach for the resource management of very large cloud platforms. *IEEE Transactions on Dependable and Secure Computing*, 10(5):253–272, 2013.
2. B. Adler. Designing private and hybrid clouds: Architectural best practices, 2012.
3. M. A. Adnan, R. Sugihara, and R. K. Gupta. Energy efficient geographical load balancing via dynamic deferral of workload. In *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, CLOUD '12, pages 188–195, Honolulu, HI, USA, 2012.
4. L. Agostinho, G. Feliciano, L. Olivi, E. Cardozo, and E. Guimaraes. A bio-inspired approach to provisioning of virtual resources in federated clouds. In *Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, DASC 2011, pages 598–604, Sydney, NSW, Australia, 2011.
5. A. Aleti, B. Buhnova, L. Grunske, A. Koziolek, and I. Meedeniya. Software architecture optimization methods: A systematic literature review. *IEEE Transactions on Software Engineering*, 39(5):658–683, 2013.
6. A. Ali-Eldin, J. Tordsson, and E. Elmroth. An adaptive hybrid elasticity controller for cloud infrastructures. In *IEEE Network Operations and Management Symposium (NOMS)* , pages 204–212, 2012.
7. J. Almeida, V. Almeida, D. Ardagna, I. Cunha, C. Francalanci, and M. Trubian. Joint admission control and resource allocation in virtualized servers. *Journal of Parallel and Distributed Computing*, 70(4):344–362, 2010.
8. M. Alrifai, D. Skoutas, and T. Risse. Selecting skyline services for QoS-based web service composition. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 11–20, Raleigh, NC, USA, 2010.
9. A. Anandkumar, C. Bisdikian, and D. Agrawal. Tracking in a spaghetti bowl: Monitoring transactions using footprints. In *Proceedings of the 2008 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 133–144, Annapolis, Maryland, USA, June 2008. ACM Press.
10. J. Anselmi and G. Casale. Heavy-traffic revenue maximization in parallel multiclass queues. *Performance Evaluation*, 70(10):806–821, 2013.
11. D. Ardagna, S. Casolari, M. Colajanni, and B. Panicucci. Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems. *Journal of Parallel and Distributed Computing*, 72(6):796–808, 2012.
12. D. Ardagna, B. Panicucci, M. Trubian, and L. Zhang. Energy-aware autonomic resource allocation in multitier virtualized environments. *IEEE Transactions on Services Computing*, 5(1):2–19, 2012.
13. D. Ardagna and B. Pernici. Adaptive service composition in flexible processes. *IEEE Transactions on Software Engineering*, 33(6):369–384, 2007.
14. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.
15. D. Bacigalupo, J. van Hemert, X. Chen, A. Usmani, A. Chester, L. He, D. Dillenberger, G. Wills, L. Gilbert, and S. Jarvis. Managing dynamic enterprise and urgent workloads on clouds using layered queuing and historical performance models. *Simulation Modelling Practice and Theory*, 19:1479–1495, 2011.
16. S. Becker, H. Koziolek, and R. Reussner. The Palladio component model for model-driven performance prediction. *Journal of Systems and Software*, 82(1):3–22, 2009.
17. A. Beloglazov, R. Buyya, Y. C. Lee, and A. Y. Zomaya. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in Computers*, 82:47–111, 2011.
18. N. Bonvin, T. Papaioannou, and K. Aberer. An economic approach for scalable and highly-available distributed applications. In *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*, CLOUD '10, pages 498–505, Miami, FL, USA, 2010.
19. A. Brogi, S. Corfini, and S. Iardella. From OWL-S descriptions to Petri nets. In *Service-Oriented Computing - ICSOC 2007 Workshops*, volume 4907 of *Lecture Notes in Computer Science*, pages 427–438. Springer-Verlag, 2009.
20. N. M. Calcavecchia, B. A. Caprarescu, E. Di Nitto, D. J. Dubois, and D. Petcu. DEPAS: A decentralized probabilistic algorithm for auto-scaling. *Computing*, 94(8–10), 2012.
21. R. N. Calheiros, R. Buyya, and C. A. F De Rose. Building an automated and self-configurable emulation testbed for grid applications. *Software–Practice & Experience*, 40:405–429, 2010.
22. R. N. Calheiros, M .A. S. Netto, C. A. F. De Rose, and R. Buyya. Emusim: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications. *Software–Practice & Experience*, 43:595–612, 2013.
23. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software–Practice & Experience*, 41(1):23–50, 2011.
24. R. Calinescu, C. Ghezzi, M. Z. Kwiatkowska, and R. Mirandola. Self-adaptive software needs quantitative verification at runtime. *Communications of the ACM*, 55(9):69–77, 2012.
25. G. Callou, P. Maciel, D. Tutsch, and J. Araujo. Models for dependability and sustainability analysis of data center cooling architectures. In *Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, TMS-DEVS '11, pages 274–281, 2011.
26. B. A. Caprarescu, N. M. Calcavecchia, E. Di Nitto, and D. J. Dubois. Sos cloud: Self-organizing services in the cloud. In *Bio-Inspired Models of Network, Information, and Computing Systems*, volume 87 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 48–55, 2012.
27. V. Cardellini, E. Casalicchio, V. Grassi, and R. Mirandola. A framework for optimal service selection in broker-based architectures with multiple QoS classes. In *Proceedings of the 2006 IEEE Services Computing Workshops*, SCW '06, pages 105–112, Chicago, IL, USA, 2006.

28. G. Casale, P. Cremonesi, and R. Turrin. Robust workload estimation in queueing network performance models. In *Proc. of Euromicro PDP*, pages 183–187, 2008.
29. S. Chaisiri, B.-S Lee, and D. Niyato. Optimization of resource provisioning cost in cloud computing. *IEEE Transactions on Services Computing*, 5(2):164–177, 2012.
30. W. Chen, X. Qiao, J. Wei, and T. Huang. A profit-aware virtual machine deployment optimization framework for cloud platform providers. In *Proceedings of the 2012 IEEE Fift International Conference on Cloud Computing*, CLOUD '12, pages 17–24, Honolulu, HI, USA, 2012.
31. M.-Y. Chung, G. Ciardo, S. Donatelli, N. He, B. Plateau, W. Stewart, E. Sulaiman, and J. Yu. Comparison of structural formalisms for modeling large markov models. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, pages 196–, April 2004.
32. A. Clark, S. Gilmore, and M. Tribastone. Quantitative analysis of web services using srmc. In *Formal Methods for Web Services*, volume 5569 of *Lecture Notes in Computer Science*, pages 296–339. Springer Berlin Heidelberg, 2009.
33. M. Comuzzi and B. Pernici. A framework for qos-based web service contracting. *ACM Transactions on the Web*, 3(3):1–52, 2009.
34. P. Cremonesi, K. Dhyani, and A. Sansottera. Service time estimation with a refinement enhanced hybrid clustering algorithm. In *Analytical and Stochastic Modeling Techniques and Applications*, pages 291–305. Springer, 2010.
35. P. Cremonesi and A. Sansottera. Indirect estimation of service demands in the presence of structural changes. In *Proceedigs of Quantitative Evaluation of Systems (QEST)*, pages 249–259. IEEE, 2012.
36. J. Dantas, R. Matos, J. Araujo, and P. Maciel. An availability model for eucalyptus platform: An analysis of warm-standy replication mechanism. In *Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics*, SMC 2012, pages 1664–1669, Seoul, Korea, 2012.
37. C. Delimitrou, N. Bambos, and C. Kozyrakis. QoS-aware admission control in heterogeneous datacenters. In *Proceedings of the 2013 10th International Conference on Autonomic Computing*, ICAC 13, pages 291–296, San Jose, CA, USA, 2013.
38. P. Desnoyers, T. Wood, P. J. Shenoy, R. Singh, S. Patil, and H. M. Vin. Modellus: Automated modeling of complex internet data center applications. *TWEB*, 6(2):8, 2012.
39. S. Di, D. Kondo, and Walfredo C. Host load prediction in a google compute cloud with a Bayesian model. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC12, pages 1–11, Salt Lake City, Utah, USA, 2012.
40. I. Drago, M. Mellia, M. M Munafo, A. Sperotto, R. Sadre, and A. Pras. Inside Dropbox: understanding personal cloud storage services. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 481–494, Boston, MA, USA, 2012.
41. S. Dutta, S. Gera, A. Verma, and B. Viswanathan. Smartscale: Automatic application scaling in enterprise clouds. In *Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing*, CLOUD '12, pages 221–228, Honolulu, HI, USA, 2012.
42. W. Ellens, M. Zivkovic, J. Akkerboom, R. Litjens, and H. van den Berg. Performance of cloud computing centers with multiple priority classes. In *Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing*, CLOUD '12, pages 245–252, Honolulu, HI, USA, 2012.
43. A. Faisal, D. Petriu, and M. Woodside. Network latency impact on performance of software deployed across multiple clouds. In *Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research*, CASCON '13, pages 216–229, Ontario, Canada, 2013.
44. B. Farley, A. Juels, V. Varadarajan, T. Ristenpart, K. D. Bowers, and M. M. Swift. More for your money: Exploiting performance heterogeneity in public clouds. In *Proceedings of the 2012 Third ACM Symposium on Cloud Computing*, SoCC '12, pages 1–14, San Jose, CA, USA, 2012.
45. J. Figueiredo, P. Maciel, G. Callou, E. Tavares, E. Sousa, and B. Silva. Estimating reliability importance and total cost of acquisition for data center power infrastructures. In *Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics*, SMC 2011, pages 421–426, Anchorage, AK, USA, 2011.
46. B. Ford. Icebergs in the clouds: The other risks of cloud computing. In *Proceedings of the 2012 4th USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'12, pages 2–2, Boston, MA, USA, 2012.
47. G. Franks, T. Al-Omari, C. M. Woodside, O. Das, and S. Derisavi. Enhanced modeling and solution of layered queueing networks. *IEEE Transactions on Software Engineering*, 35(2):148–161, 2009.
48. C. Gasquet and P. Witomski. *Fourier analysis and applications: filtering, numerical computation, wavelets*, volume 30 of *Texts in applied mathematics*. Springer-Verlag, pub-SV:adr, 1999.
49. D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. Workload analysis and demand prediction of enterprise data center applications. In *Proceedings of the 2007 IEEE 10th International Symposium on Workload Characterization*, IISWC '07, pages 171–180, Boston, MA, USA, 2007.
50. H. Goudarzi and M. Pedram. Multi-dimensional sla-based resource allocation for multi-tier cloud computing systems. In *Proceedings of the 2011 IEEE 4th International Conference on Cloud Computing*, CLOUD '11, pages 324–331, Washington, DC, USA, 2011.
51. H. Goudarzi and M. Pedram. Geographical load balancing for online service applications in distributed datacenters. In *Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing*, CLOUD '13, pages 351–358, Santa Clara, CA, USA, 2013.
52. J. El Haddad, M. Manouvrier, and M. Rukoz. TQoS: Transactional and QoS-aware selection algorithm for automatic web service composition. *IEEE Transactions on Services Computing*, 3(1):73–85, 2010.
53. S. Haddad, L. Mokdad, and S. Youcef. Response time of BPEL4WS constructors. In *Proceedings of the 2010 IEEE Symposium on Computers and Communications*, ISCC'10, pages 695–700, Riccione, Italy, 2010.
54. M. Hadji and D. Zeghlache. Minimum cost maximum flow algorithm for dynamic resource allocation in clouds. In *Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing*, CLOUD '12, pages 876–882, Honolulu, HI, USA, 2012.
55. S. He, L. Guo, M. Ghanem, and Y. Guo. Improving resource utilisation in the cloud environment using multivariate probabilistic models. In *Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing*, CLOUD '12, pages 574–581, Honolulu, HI, USA, 2012.

56. Z. Hill, J. Li, M. Mao, A. Ruiz-Alvarez, and M. Humphrey. Early observations on the performance of Windows Azure. *Scientific Programming*, 19(2-3):121–132, 2011.
57. G. A. Hoffmann, K. S. Trivedi, and M. Malek. A best practice guide to resource forecasting for computing systems. *IEEE Transactions on Reliability*, 56(4):615–628, 2007.
58. D. T. Huang, D. Niyato, and P. Wang. Optimal admission control policy for mobile cloud computing hotspot with cloudlet. In *Proceedings of the 2012 IEEE Wireless Communications and Networking Conference*, WCNC 2012, pages 3145–3149, Paris, France, 2012.
59. J. Huang, Y. Liu, and Q. Duan. Service provisioning in virtualization-based cloud computing: Modeling and optimization. In *Proceedings of 2012 IEEE Global Communications Conference*, GLOBECOM 2012, pages 1710–1715, Anaheim, CA, USA, 2012.
60. I. Hwang and M. Pedram. Hierarchical virtual machine consolidation in a cloud computing system. In *Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing*, CLOUD '13, pages 196–203, Santa Clara, CA, USA, 2013.
61. R. Jhawar and V. Piuri. Fault tolerance management in IaaS clouds. In *Proceedings of 2012 IEEE First AESS European Conference on Satellite Telecommunications*, ESTEL 2012, pages 1–6, Rome, Italy, 2012.
62. D. Jiang, G. Pierre, and C.-H. Chi. Autonomous resource provisioning for multi-service web applications. In *Proceedings of the 2010 19th International Conference on World Wide Web*, WWW '10, pages 471–480, Raleigh, NC, USA, 2010.
63. Y. Jie, Q. Jie, and L. Ying. A profile-based approach to just-in-time scalability for cloud applications. In *Proceedings of the 2009 IEEE International Conference on Cloud Computing*, CLOUD '09, pages 9–16, Bangalore, India, 2009.
64. Gueyoung Jung, Kaustubh R Joshi, Matti A Hiltunen, Richard D Schlichting, and Calton Pu. Generating adaptation policies for multi-tier applications in consolidated server environments. In *Autonomic Computing, 2008. ICAC'08. International Conference on*, pages 23–32, Chicago, IL, USA, 2008. IEEE.
65. A. Kalbasi, D. Krishnamurthy, J. Rolia, and S. Dawson. DEC: Service demand estimation with confidence. *IEEE Transactions on Software Engineering*, 38(3):561–578, 2012.
66. A. Kalbasi, D. Krishnamurthy, J. Rolia, and M. Richter. MODE: Mix driven on-line resource demand estimation. In *Proceedings of the 7th International Conference on Network and Services Management*, pages 1–9. International Federation for Information Processing, 2011.
67. G. Keller, M. Tighe, H. Lutfiyya, and M. Bauer. DCSim: A data centre simulation tool. In *Proceedings of 2012 8th international conference on Network and service management, and 2012 workshop on systems virtualiztion management*, CNSM-SVM 2012, pages 385–392, Las Vegas, NV, USA, 2012.
68. J. Kephart, H. Chan, R. Das, D. Levine, G. Tesauro, F. Rawson, and C. Lefurgy. Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs. In *Proceedings of the Fourth International Conference on Autonomic Computing*, ICAC '07, pages 24–24, Jacksonville, FL, USA, 2007.
69. A. Khan, X. Yan, Shu Tao, and N. Anerousis. Workload characterization and prediction in the cloud: A multiple time series approach. In *Proceedings of the 2012 IEEE Network Operations and Management Symposium*, NOMS 2012, pages 1287–1294, Maui, HI, USA, 2012.
70. H. Khazaei, J. Misic, V. Misic, and S. Rashwand. Analysis of a pool management scheme for cloud computing centers. *IEEE Transactions on Parallel and Distributed Systems*, 24(5):849–861, 2013.
71. M. Kiran, M Jiang, D.J. Armstrong, and K. Djemame. Towards a service lifecycle based methodology for risk assessment in cloud computing. In *Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, DASC 2011, pages 449–456, Sydney, NSW, Australia, 2011.
72. A. Klein, F. Ishikawa, and S. Honiden. Towards network-aware service composition in the cloud. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 959–968, Lyon, France, 2012.
73. D. Kliazovich, P. Bouvry, and S. U. Khan. GreenCloud: A packet-level simulator of energy-aware cloud computing data centers. In *Proceedings of the 2010 IEEE Global Telecommunications Conference*, GLOBECOM 2010, pages 1–5, Miami, FL, USA, 2010.
74. K. Konstanteli, T. Cucinotta, K. Psychas, and T. Varvarigou. Admission control for elastic cloud services. In *Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing*, CLOUD '12, pages 41–48, Honolulu, HI, USA, 2012.
75. D. Kossmann, T. Kraska, and S. Loesing. An evaluation of alternative architectures for transaction processing in the cloud. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 579–590, Indianapolis, IN, USA, 2010.
76. D. Kusic and N. Kandasamy. Risk-aware limited lookahead control for dynamic resource provisioning in enterprise computing systems. In *Proceedings of the 2006 IEEE International Conference on Autonomic Computing*, ICAC '06, pages 74–83, Dublin, Ireland, 2006.
77. D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing*, 12(1):1–15, 2009.
78. P. Leitner, W. Hummer, B. Satzger, C. Inzinger, and S. Dustdar. Cost-efficient and application sla-aware client side request scheduling in an infrastructure-as-a-service cloud. In *Preoceedings of the 2012 IEEE 5th International Conference on Cloud Computing*, CLOUD '12, pages 213–220, Honolulu, HI, USA, 2012.
79. A. Li, X. Yang, S. Kandula, and M. Zhang. Cloudcmp: comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM, 2010.
80. Z. Li, L. O'Brien, R. Ranjan, and M. Zhang. Early observations on performance of Google compute engine for scientific computing. In *Proceedings of the 2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, volume 1 of *CloudCom 2013*, pages 1–8, Bristol, United Kingdom, 2013.
81. S. Liu, X. Huang, H. Fu, and G. Yang. Understanding data characteristics and access patterns in a cloud storage system. In *Proceedings of the 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, CCGrid 2013, pages 327–334, Delft, Nederlands, 2013.
82. Y. Liu, I Gorton, and A Fekete. Design-level performance prediction of component-based applications. *IEEE Transactions on Software Engineering*, 31(11):928–941, 2005.

83. Z. Liu, L. Wynter, C. Xia, and F. Zhang. Parameter inference of queueing models for it systems using end-to-end measurements. *Performance Evaluation*, 63(1):36–60, 2006.
84. F. Longo, R. Ghosh, V.K. Naik, and K.S. Trivedi. A scalable availability model for Infrastructure-as-a-Service cloud. In *Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems Networks*, DSN 2011, pages 335–346, Hong Kong, China, 2011.
85. N. Ben Mabrouk, S. Beauche, E. Kuznetsova, N. Georgantas, and V. Issarny. QoS-aware service composition in dynamic service oriented environments. In *Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*, Middleware '09, pages 1–20, Urbanna, Il, USA, 2009.
86. M. Maggio, H. Hoffmann, M. D. Santambrogio, A. Agarwal, and A. Leva. A comparison of autonomic decision making techniques. Technical Report MIT-CSAIL-TR-2011-019, Massachusetts Institute of Technology, 2011.
87. M. Mao and M. Humphrey. Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In *Proceedings of the 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 1–12, Seattle, WA, USA, 2011.
88. M. Mao and M. Humphrey. A performance study on the VM startup time in the cloud. In *Proceedinngs of the 2012 IEEE Fifth International Conference on Cloud Computing*, CLOUD '12, pages 423–430, Honolulu, HI, USA, 2012.
89. M. Melo, P. Maciel, J. Araujo, R. Matos, and C. Araujo. Availability study on cloud computing environments: Live migration as a rejuvenation mechanism. In *Proceedings of the 2013 IEEE/IFIP 43rd International Conference on Dependable Systems and Networks*, DSN 2013, pages 1–6, Hong Kong, China, 2013.
90. D. A. Menascé, E. Casalicchio, and V. K. Dubey. On optimal service selection in service oriented architectures. *Performance Evaluation*, 67(8):659–675, 2010.
91. D. Menascé, V. Almeida, and L. Dowdy. *Capacity planning and performance modeling: from mainframes to client-server systems*. Prentice-Hall, Inc., 1994.
92. X. Meng, V. Pappas, and L. Zhang. Improving the scalability of data center networks with traffic-aware virtual machine placement. In *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, pages 1154–1162, San Diego, CA, USA, 2010.
93. T. Omari, G. Franks, M. Woodside, and A. Pan. Efficient performance models for layered server systems with replicated servers and parallel behaviour. *Journal of Systems and Software*, 80(4):510–527, 2007.
94. S. Ostermann, K. Plankensteiner, R. Prodan, and T. Fahringer. Groudsim: An event-based simulation framework for computational grids and clouds. In *Proceedings of the 2010 Conference on Parallel Processing*, Euro-Par 2010, pages 305–313, Ischia, Italy, 2011.
95. Z. Ou, H. Zhuang, J. K. Nurminen, A. Ylä-Jääski, and P. Hui. Exploiting hardware heterogeneity within the same instance type of amazon ec2. In *Proceedings of the 4th USENIX Conference on Hot Topics in Cloud Ccomputing*, HotCloud'12, pages 4–4, Boston, MA, USA, 2012.
96. G. Pacifici, W. Segmuller, M. Spreitzer, and A. Tantawi. CPU demand for web serving: Measurement analysis and dynamic estimation. *Performance Evaluation*, 65(6):531–553, 2008.
97. P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Salem. Adaptive control of virtualized resources in utility computing environments. In *Proceedings of the 2Nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, EuroSys '07, pages 289–302, Lisbon, Portugal, 2007.
98. T. Patikirikorala, A. Colman, J. Han, and L. Wang. A multi-model framework to implement self-managing control systems for qos management. In *Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '11, pages 218–227, Honolulu, HI, USA, 2011.
99. J. F. Pérez and G. Casale. Assessing sla compliance from palladio component models. In *Proceedings of the 2013 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, SYNASC '13, pages 409–416, Timisoara, Romania, 2013.
100. D. Petcu, G. Macariu, S. Panica, and C. Craciun. Portable cloud applications - from theory to practice. *Future Generation Computer Systems*, 29(6):1417–1430, 2013.
101. R. Ranjan, L. Zhao, X. Wu, A. Liu, A. Quiroz, and M. Parashar. Peer-to-peer cloud provisioning: Service discovery and load-balancing. In *Cloud Computing*, Computer Communications and Networks, pages 195–217. Springer London, 2010.
102. P. Reinecke and K. Wolter. Phase-type approximations for message transmission times in web services reliable messaging. In *Proceedings of the 2008 SPEC International Workshop on Performance Evaluation: Metrics, Models and Benchmarks*, SIPEW '08, pages 191–207, Darmstadt, Germany, 2008.
103. J. Rolia and V. Vetland. Parameter estimation for performance models of distributed application systems. In *In Proc. of CASCON*, page 54. IBM Press, 1995.
104. Jerome Rolia and Vidar Vetland. Correlating resource demand information with ARM data for application services. In *Proceedings of the 1st international workshop on Software and performance*, pages 219–230. ACM, 1998.
105. N. Roy, A. Dubey, and A. Gokhale. Efficient autoscaling in the cloud using predictive models for workload forecasting. In *Proceedings of the 2011 IEEE International Conference on Cloud Computing*, CLOUD '11, pages 500–507, Washington, DC, USA, 2011.
106. J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz. Runtime measurements in the cloud: Observing, analyzing, and reducing variance. *Proceedings of VLDB Endowment*, 3(1–2):460–471, 2010.
107. D. Schuller, A. Polyvyanyy, L. García-Bañuelos, and Stefan Schulte. Optimization of complex qos-aware service compositions. In *Proceedings of the 2011 9th International Conference on Service-Oriented Computing*, ICSOC'11, pages 452–466, Paphos, Cyprus, 2011.
108. M. Sedaghat, F. Hernandez-Rodriguez, and E. Elmroth. A virtual machine re-packing approach to the horizontal vs. vertical elasticity trade-off for cloud autoscaling. In *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference*, CAC '13, pages 6:1–6:10, Miami, FL, USA, 2013.
109. Abhishek B Sharma, Ranjita Bhagwan, Monojit Choudhury, Leana Golubchik, Ramesh Govindan, and Geoffrey M Voelker. Automatic request categorization in internet services. *ACM SIGMETRICS Performance Evaluation Review*, 36(2):16–25, 2008.
110. R. Singh, U. Sharma, E. Cecchet, and P. Shenoy. Autonomic mix-aware provisioning for non-stationary data center workloads. In *Proceedings of the 7th ACM international conference on Autonomic computing*, pages 21–30, 2010.

111. K. Sowmya and R.P. Sundarraj. Strategic bidding for cloud resources under dynamic pricing schemes. In *Proceedings of 2012 International Symposium on Cloud and Services Computing*, ISCOS 2012, pages 25–30, Mangalore, India, 2012.
112. S. Spicuglia, L. Y. Chen, and W. Binder. Join the best queue: Reducing performance variability in heterogeneous systems. In *Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing*, CLOUD '13, pages 139–146, Santa Clara, CA, USA, 2013.
113. S. Stein, T. R. Payne, and N. R. Jennings. Flexible provisioning of web service workflows. *ACM Transactions on Internet Technology*, 9(1):1–45, 2009.
114. C. Stewart, A. Chakrabarti, and R. Griffith. Zoolander: Efficiently meeting very strict, low-latency slos. In *Proceedings of the 10th International Conference on Autonomic Computing (ICAC)*, pages 265–277, 2013.
115. Charles A Sutton and Michael I Jordan. Probabilistic inference in queueing networks. In *SysML*, 2008.
116. Charles A. Sutton and Michael I. Jordan. Inference and learning in networks of queues. In Yee Whye Teh and D. Mike Titterington, editors, *AISTATS*, volume 9 of *JMLR Proceedings*, pages 796–803. JMLR.org, 2010.
117. C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici. A scalable application placement controller for enterprise data centers. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 331–340, Banff, Canada, 2007.
118. E. Thereska and G. R Ganger. IRONmodel: Robust performance models in the wild. *ACM SIGMETRICS Performance Evaluation Review*, 36(1):253–264, 2008.
119. M. Tribastone. A fluid model for layered queueing networks. *IEEE Transactions on Software Engineering*, 39(6):744–756, 2013.
120. B. Trushkowsky, P. Bodík, A. Fox, M. J. Franklin, M. I. Jordan, and D. A. Patterson. The SCADS director: Scaling a distributed storage system under stringent performance requirements. In *Proceedings of the 9th USENIX Conference on File and Stroage Technologies*, FAST'11, pages 12–12, San Jose, CA, USA, 2011.
121. R. Van den Bossche, K. Vanmechelen, and J. Broeckhove. Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads. In *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*, CLOUD'10, pages 228–235, Miami, FL, USA, 2010.
122. H. Wada, A. Fekete, L. Zhao, K. Lee, and A. Liu. Data consistency properties and the trade-offs in commercial cloud storage: the consumers' perspective. In *Proceedings of the 5th Biennial Conference on Innovative Data Systems Research*, CIDR 2011, pages 134–143, Asilomar, CA, USA, 2011.
123. G. Wang and T. S. E. Ng. The impact of virtualization on network performance of amazon ec2 data center. In *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, pages 1163–1171, San Diego, CA, USA, 2010.
124. H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, and L. Zhou. Distributed systems meet economics: pricing in the cloud. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, HotCloud'10, pages 6–6, Boston, MA, USA, 2010.
125. L. Wang and J. Shen. Towards bio-inspired cost minimisation for data-intensive service provision. In *Proceedings of the 2012 IEEE First International Conference on Services Economics*, SE 2012, pages 16–23, Honolulu, HI, USA, 2012.
126. W. Wang, B. Li, and B. Liang. Towards optimal capacity segmentation with hybrid cloud pricing. In *Proccedings of the 2012 IEEE 32nd International Conference on Distributed Computing Systems*, ICDCS 2012, pages 425–434, Macau, China, 2012.
127. B. Wei, C. Lin, and X. Kong. Dependability modeling and analysis for the virtual data center of cloud computing. In *Proceedings of the 2011 IEEE 13th International Conference on High Performance Computing and Communications*, HPCC 2011, pages 784–789, Bamff, Canada, 2011.
128. G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong. A game-theoretic method of fair resource allocation for cloud computing services. *Journal of Supercomputing*, 54(2):252–269, 2010.
129. B. Wickremasinghe, R.N. Calheiros, and R. Buyya. CloudAnalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In *Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications*, AINA 2010, pages 446–452, Perth, Australia, 2010.
130. L. Wu, S. K. Garg, and R. Buyya. SLA-based admission control for a software-as-a-service provider in cloud computing environments. *Journal of Computer and System Sciences*, 78(5):1280–1299, 2012.
131. L. Wu, S. K. Garg, S. Versteeg, and R.Buyya. SLA-based resource provisioning for hosted software as a service applications in cloud computing environments. *IEEE Transactions on Services Computing*, 99:1, 2013.
132. X. Wu and M. Woodside. A calibration framework for capturing and calibrating software performance models. In *Computer Performance Engineering*, pages 32–47. Springer, 2008.
133. Z. Xiao, Q. Chen, and H. Luo. Automatic scaling of internet applications for cloud computing services. *IEEE Transactions on Computers*, 63(5):1111–1123, 2014.
134. P. Xiong, Y. Chi, S. Zhu, J. Tatemura, C. Pu, and H. HacigümüŞ. Activesla: A profit-oriented admission control framework for database-as-a-service providers. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, SOCC '11, pages 1–14, Cascais, Portugal, 2011.
135. P. Xiong, Z. Wang, S. Malkowski, Q. Wang, D. Jayasinghe, and C. Pu. Economical and robust provisioning of n-tier cloud workloads: A multi-level control approach. In *Proceedings of the 31st IEEE International Conference on Distributed Computing Systems (ICDCS)* , pages 571–580, 2011.
136. Y. Xu, Z. Musgrave, B. Noble, and M. Bailey. Bobtail: Avoiding long tails in the cloud. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, NSDI '13, pages 329–342, Lombard, IL, USA, 2013.
137. Z. Ye, A. Bouguettaya, and X. Zhou. QoS-aware cloud service composition based on economic models. In *Proceedings of the 10th International Conference on Service-Oriented Computing*, ICSOC'12, pages 111–126, Shanghai, China, 2012.
138. T. Yu, Y. Zhang, and K.-J. Lin. Efficient algorithms for web services selection with end-to-end QoS constraints. *ACM Transactions on the Web*, 1(1), May 2007.
139. S. Zaman and D. Grosu. An online mechanism for dynamic vm provisioning and allocation in clouds. In *Proceedings of the 2012 IEEE 5th International Conference on Cloud Computing*, CLOUD '12, pages 253–260, Honolulu, HI, USA, 2012.

140. Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, 2010.
141. Q. Zhang, L. Cherkasova, and E. Smirni. A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In *Proc. of the 4th ICAC Conference*, pages 27–27, 2007.
142. Q. Zhang, Q. Zhu, M. F. Zhani, and R. Boutaba. Dynamic service placement in geographically distributed clouds. In *Proceedings of the 2012 IEEE 32Nd International Conference on Distributed Computing Systems*, ICDCS '12, pages 526–535, Macau, China, 2012.
143. T. Zheng, C. M. Woodside, and M. Litoiu. Performance model estimation and tracking using optimal filters. *IEEE Transactions on Software Engineering*, 34(3):391–406, 2008.
144. Q. Zhu and T. Tung. A performance interference model for managing consolidated workloads in QoS-aware clouds. In *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, CLOUD '12, pages 170–179, Honolulu, HI, USA, 2012.