J Netw Syst Manage (2015) 23:567–619
DOI 10.1007/s10922-014-9307-7

# Resource Management in Clouds: Survey and Research Challenges

**Brendan Jennings · Rolf Stadler**

Received: 13 March 2013 / Revised: 11 February 2014 / Accepted: 20 February 2014 /
Published online: 2 March 2014

**Abstract** Resource management in a cloud environment is a hard problem, due to: the scale of modern data centers; the heterogeneity of resource types and their interdependencies; the variability and unpredictability of the load; as well as the range of objectives of the different actors in a cloud ecosystem. Consequently, both academia and industry began significant research efforts in this area. In this paper, we survey the recent literature, covering 250+ publications, and highlighting key results. We outline a conceptual framework for cloud resource management and use it to structure the state-of-the-art review. Based on our analysis, we identify five challenges for future investigation. These relate to: providing predictable performance for cloud-hosted applications; achieving global manageability for cloud systems; engineering scalable resource management systems; understanding economic behavior and cloud pricing; and developing solutions for the mobile cloud paradigm.

**Keywords** Cloud computing · Resource allocation · Resource management · Virtualization · Survey

## 1 Introduction

Over the past decade, advances in commodity computing and virtualization technologies have enabled the cost-effective realization of large-scale data centers

B. Jennings (✉)
TSSG, Waterford Institute of Technology, Waterford, Ireland
e-mail: bjennings@ieee.org

R. Stadler
ACCESS Linnæus Center, KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: stadler@kth.se
URL: http://www.ee.kth.se/~stadler

that run large portion of today's Internet applications and backend processing. The economies of scale that arose allowed data center infrastructure to be leased profitably to third parties. Thus emerged the *Cloud Computing* paradigm, wherein a pool of computing resources is shared between applications that are accessed over the Internet. Cloud computing has become a broad and popular term, used not only by the technology community but the general public as well. It refers to applications delivered over the Internet, as well to hardware and system software residing in data centers that host those applications.

Cloud computing encompasses both (a) the provision of resources to third parties on a leased, usage-based basis, and (b) the private infrastructures maintained and utilized by individual organizations. The former case is referred to as *public clouds* and the latter as *private clouds*; the scenario where an enterprise extends the capacity of its private cloud by leasing public cloud resources is referred to as *cloud bursting*—which creates a *hybrid cloud*. Another emerging catergory is *community clouds* [37, 152], in which the cloud resources are contributed by many individuals/organisations and in which governance is decentralised. Public cloud environments are typically classified as *Infrastructure-as-a-Service (IaaS)*, *Platform-as-a-Service (PaaS)*, and *Software-as-a-Service (SaaS)*, depending on the level of abstraction at which the service is offered (see Sect. 2).

Research related to different aspects of cloud computing has accelerated steadily over the last 3–5 years; workshops and conferences have been established, and an increasing number of publications is being produced. General introductions into the field and its research challenges have been provided by Zhang et al. [267] and Armbrust et al. [17]. Gulati et al. [93] discuss resource management challenges and techniques with the perspective of VMWare's offerings. A topic that is receiving considerable attention is data center networking, which has been surveyed by Abts and Felderman [1] and Bari et al. [21].

Our objective with this paper is to provide a comprehensive survey of recent research into the challenging topic of *resource management in cloud environments*. We want to convey the complexity of the problem, describe the state-of-the-art, and outline the fundamental open challenges—as we see them. For the survey, we reviewed 350+ recent papers, from which we selected about two thirds for this article. We hope that our effort provides a point entry for researchers attracted to this topic and additional information and orientation for those who are already working on its challenges.

In the context of cloud computing, we understand *resource management* as the process of allocating computing, storage, networking and (indirectly) energy resources to a set of applications, in a manner that seeks to jointly meet the performance objectives of the applications, the infrastructure (i.e., data center) providers and the users of the cloud resources. The objectives of the providers center around efficient and effective resource use within the constraints of Service Level Agreements (SLAs) with the Cloud Users. Efficient resource use is typically achieved through virtualization technologies, which facilitate statistical multiplexing of resources across customers and applications. The objectives of the Cloud Users tend to focus on application performance, their availability, as well as the cost-effective scaling of available resources in line with changing application demands. Often, these objectives come with constraints regarding resource dedication to meet non-functional requirements relating to, for example, security or regulatory compliance.

To lay the foundation for the survey, we introduce in Sect. 2 the actors in the cloud computing ecosystem and their roles in resource management, we outline possible management objectives these actors may choose, we detail the types of resources that are managed, and we review the key technologies that enable their management. Further, we introduce a conceptual framework that gives a high-level view of the functional components of a cloud resource management system and their interactions. In Sect. 3 we survey the recent literature in cloud resource management. We classify the field into eight functional areas (each introduced in Sect. 2.5), namely:

- global scheduling of virtualized resources;
- resource demand profiling;
- resource utilization estimation;
- resource pricing and profit maximization;
- local scheduling of cloud resources;
- application scaling and provisioning;
- workload management; and,
- cloud management systems.

In Sect. 4 we identify five research challenges that we believe merit special attention. The first three, concerning achieving predictable performance, enabling global manageability and engineering scaleable resource management systems, relate to known but inherently difficult problems of complex software systems. The last two concern economic behavior/pricing models and mobile cloud computing. They reflect emerging trends in the provisioning and use of cloud-hosted applications. Finally, Sect. 5 summarizes our findings and concludes the paper.

## 2 Scope of Cloud Computing Resource Management

In this section we outline the scope of the resource management problem in cloud environments. We introduce the actors in a cloud ecosystem, focussing on their role(s) in the resource management process and the kinds of management objectives they are likely to pursue. We discuss the resource types that comprise a cloud environment and briefly introduce the key technologies that enable their management. Finally, we introduce our conceptual framework for cloud resource management, outlining the key resource management functions and their inter-relations. The latter categorization is then used in Sect. 3 to structure our literature review.

### 2.1 Actors and Cloud Application Provisioning Models

As we noted above, cloud offerings are generally marketed as IaaS, PaaS or SaaS, but there is a spectrum of interpretations of these terms, which have made clear comparisons and cloud interoperability difficult. However, recent IaaS-focussed standardization efforts, notably the OGF's Open Cloud Computing Interface (OCCI) [63], the DMTF's Cloud Infrastructure Management Interface (CIMI) [87], along with the de-facto standard that is Amazon Web Services [15], are providing marketing terms like PaaS and IaaS with a stronger technical underpinning. In particular, adoption of standardized interfaces should provide a stronger separation of concerns in cloud environments, both public and private. As depicted in Fig. 1, and broadly following the delineation outlined by Armbrust et al. [17], we assume the following roles that can be adopted by actors in the cloud environment: the *Cloud Provider*, the *Cloud User* and the *End User*. Depending on the use of IaaS or PaaS interfaces these roles could be adopted by a single organization (in a private cloud), or by different organizations (in a public cloud). From a cloud resource management perspective these roles can be outlined as follows:

*Cloud Provider* manages a set of data center hardware and system software resources, providing in the public cloud context either IaaS or PaaS abstractions of those resources to Cloud Users, or in the private cloud context, managing all the resources required to provide SaaS to End Users. The Cloud Provider is responsible for allocating these resources so that it meets SLAs agreed with Cloud Users/End Users and/or achieves other management goals;

*Cloud User* uses public clouds to host applications that it offers to its End Users. It is responsible for meeting SLAs agreed with its customers (i.e., End Users) and is typically concerned with doing so in a manner that minimizes its costs and maximizes its profits by ensuring that the level of resources leased from the Cloud Provider scales in line with demands from its End Users;

*End User* generates the workloads (application usage sessions) that are processed using cloud resources. She typically does not play a direct role in resource management,[1] but her behavior can influence, and can be influenced by, the resource management decisions of the Cloud User and Cloud Provider.

## 2.2 Management Objectives

Actors in a cloud computing environment each have objectives they seek to achieve through configuring, or otherwise influencing, resource management processes. Many management objectives are possible. For brevity, we limit the discussion here to the scenario in which the Cloud Provider and Cloud User are different organizations and where, as depicted in Fig. 1a, the Cloud User leases resources from the Cloud Provider via an IaaS interface.

[1] A notable exception is the ability for an End User to configure the behavior of certain non-delay sensitive, computation focussed distributed applications, for example those based on the MapReduce framework.

In the IaaS context, the Cloud Provider will seek to satisfy SLAs it has agreed with Cloud Users regarding the provision of virtual infrastructure via its data center resources. An SLA is a formal agreement between the Cloud Provider and the Cloud User, defining in quantitative terms the functional and non-functional aspects of the service being offered. SLAs vary widely in scope and specificity; typically, they encompass aspects of, *inter alia*, service availability, service performance, security and privacy, data access, problem resolution, change management and dispute mediation. Of these, the quantifiable Service Level Objectives (SLOs) pertaining to availability (e.g., the service is available for 99.99 % of the time within a given year) and performance (e.g., the maximum query response time is 10 ms) are directly related to resource management. An example of a typical SLA for an IaaS service is that of the Amazon EC2 service [12], which defines an availability objective of "monthly uptime percentage of at least 99.95 % for Amazon EC2 and Amazon EBS within a region." A detailed discussion of cloud computing SLAs, focussing on recent research efforts in this area can be found in a recent European Commission report [136].

Depending on the specifics of an SLA the satisfaction of a given SLO may be viewed as a constraint (SLOs *must* be satisfied insofar as possible) or as an objective (SLOs *should* be satisfied, to the degree possible given other constraints and objectives). The Cloud Provider may offer different service levels to its customers and may choose to prioritize access to resources to different customer groups depending on the nature of their SLAs. In such cases it will seek to pursue a *service differentiation* management objective.

In addition to management objectives relating to satisfying customer SLAs, the Cloud Provider may pursue objectives specifically relating to the management of its data center infrastructure. Such objectives could, for example, include: *balanced load*, whereby resources should be allocated in a manner such that utilization is balanced across all resources of a particular type; *fault tolerance*, whereby resources are allocated in a manner such that the impact of a failure on system performance is minimized; or *energy use minimization*, whereby data center resources are allocated in a manner that the amount of energy required to execute a given workload is minimized. The Cloud Provider may seek to jointly optimize a number of metrics, to optimize them in a prioritized manner (for example, ensure a given level of fault tolerance and minimize energy usage otherwise), or to optimize them in arbitrary consistent combinations. Moreover, the Cloud Provider may choose to apply different objectives during different operational conditions; for example, it may seek to minimize energy use during low load conditions, but switch to seeking service differentiation objectives during overload conditions.

The Cloud User has a different set of management objectives. It typically has an SLA in place with its own customers (the End Users). In a manner similar to the Cloud Provider, it is likely to formulate its management objectives in line with these SLAs. In parallel, it may seek to exploit the *elasticity* property of cloud environments, that is, the ability to immediately make available additional resources to accomodate demand surges and release them whenever demand abates (Herbst et al. [102] discuss elasticity in detail and propose an alternative definition). The Cloud User may thus formulate management objectives that reflect its approach

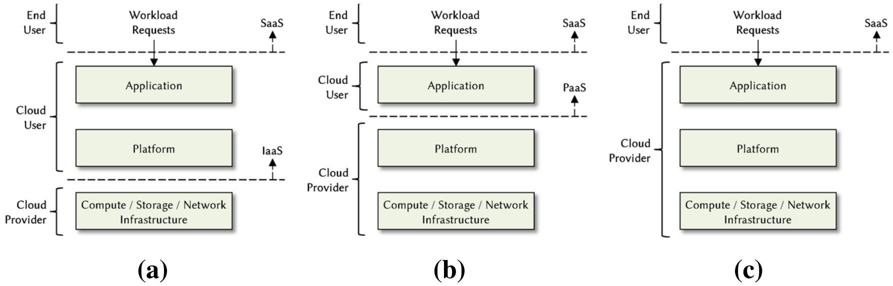

**Fig. 1** Three common application provisioning models. End Users generate workload requests for applications hosted on (typically) virtualized infrastructure based on compute servers, storage devices and networking equipment. Applications are often run on specialized software platforms that provide software abstractions and application provisioning functionality. The roles of End User, Cloud User, and Cloud Provider can be adopted by a single organization (in the case of a private cloud) or by different organizations (in a public cloud). **a** Environment using IaaS interface, **b** environment using PaaS interface, **c** environment with only a SaaS interface

to resource reservation—either conservatively over-provisioning resources (at extra cost) to accommodate demand surges, or aggressively minimizing cost by closely matching the level of leased resources with demand, but running the risk of violating SLAs with End Users.

The exemplar management objectives discussed above represent a small selection of the conceivable set of objectives. However, as is evident from our literature review in Sect. 3 management objectives are rarely discussed in detail, and we lack a principled treatment of the different possible objectives and how they relate to the engineering of cloud systems.

## 2.3 Resource Types

We now briefly outline the main types of resources that comprise the subjects of a cloud resource management system:

### *2.3.1 Compute Resources*

Compute resources are the collection of *Physical Machines (PMs)*, each comprised of one or more processors, memory, network interface and local I/O, which together provide the computational capacity of a cloud environment. Typically PMs have deployed on them virtualization software that allows them host a number of *Virtual Machines (VMs)* that are isolated from each other and that may run different operating systems, platforms and applications. In the literature, most researchers model VMs and PMs as being constrained by their processing capacity and memory availability. However, recent work [6, 86] highlights the impact of contention between VMs caused for shared processor caches and other micro-architectural resources, suggesting that resource management process may benefit from more detailed models of compute resources.

### 2.3.2 Networking Resources

Compute resources (on PMs) within a data center are packaged into racks and are typically organized as clusters of thousands of hosts for resource allocation purposes. Critically, these PMs must be interconnected with a high-bandwidth network, typically built upon Gigabit Ethernet or InfiniBand technologies. For all cloud-hosted applications, particularly those realized via parallel computation, communication overhead imposed by the data center networking technologies and networking protocols constrain overall performance. There are two critical aspects.

The first is *network topology*, the design of which significantly impacts performance and fault tolerance. Current data center network topologies are based on hierarchical, tree-like topologies similar to those used in early telephony networks, although a number of alternative topologies including proposals based on fat trees [8], hyper-cubes [97] and randomized small-world topologies [197] have emerged. In all cases a key goal is engineering a scaleable topology in which "increasing the number of ports in the network should linearly increase the delivered bisection bandwidth" [1].

The second key aspect is more directly tied to resource management: it is how to provide predictable latency and bandwidth in a data center network in the face of varying traffic patterns. Traditionally, the solution has been network over-provisioning, but this is prohibitively expensive in large scale data centers and is inherently difficult due to lack of detailed traffic models. Given this, there has been a move towards implementing service differentiation via Quality-of-Service (QoS) policies that segregate traffic for performance isolation, so permitting high-level traffic engineering. A natural extension of this approach towards technologies enabling the virtualization of data center networks is currently gaining attention [21]. The creation of virtual data center networks offer the opportunity to deploy custom network addressing schemes and networking protocols, the latter being a particular concern given the problems associated with the TCP incast behavior in the data center context [44].

### 2.3.3 Storage Resources

Public Cloud Providers, such as Amazon, offer persistent storage services of different types, ranging from virtual disks and database services to object stores, each service having varying levels of data consistency guarantees and reliability.

A difficult issue for storage services is how to achieve elasticity, so that a service dynamically scales with an increasing number of users, load, or data volume, and, similarly, shrinks if these metrics tend in the opposite direction. Scaling is hard to achieve in traditional database systems, which offer strong data consistency and Atomicity, Consistency, Isolation and Durability (ACID) transactional properties. Fortunately, many cloud-hosted web applications, such as blogs, tolerate a weaker level of consistency, e.g., eventual consistency. This allows designers to exploit the trade-off between performance (response time, availability) and consistency in such systems. This has led to the development of a range of so-called "NoSQL" data storage technologies, optimized for different operational and functional conditions.

Examples of these technologies include document stores, column stores, graph databases, key-value stores and triple stores; an overview of this area is provided by Robinson et al. [182].

Recently, *distributed key-value stores* have attracted much attention as a technology for elastic storage. Key-value stores allow for insertion, retrieval and writing of objects that are identified by keys from a flat name space and are utilized by many cloud-hosted applications. They are typically built on structured overlays that run on commodity hardware; examples include *Voldemort* used by LinkedIn [209], *Cassandra* by Facebook [137], and *Dynamo* by Amazon [61].

### *2.3.4 Power Resources*

Data centers account for a significant portion of worldwide energy usage[2] and energy costs, both direct costs to power utilities and associated power distribution and cooling costs, account for a significant portion of the overall cost base for a data center operator. For example, Hamilton [99] estimates that direct energy consumption costs account for 19 % of the overall operational costs for a large scale data center, whilst power distribution and cooling infrastructure (amortized over 15 years) account for 23 %.

In a data center power is consumed by servers themselves, as well as by networking equipment, power distribution equipment, cooling infrastructure and supporting infrastructure (lighting etc.). Data centers typically source power from one or more power utility providers, with a recent trend towards local power generation/storage, especially from renewable energy sources including wind and solar [263]. Given the costs associated with powering data centers there is significant focus on minimizing energy consumption. There are four main approaches: development of low power components to improve hardware energy efficiency; developing techniques for energy-aware resource management; implementing applications in an inherently energy efficient manner; and development of more efficient cooling systems and the location of data centers in geographical areas with advantageous climatic conditions.

## 2.4 Enabling Technologies

We now outline the main technologies that enable cloud computing environments and that are directly or indirectly harnessed by cloud resource management processes.

### *2.4.1 Infrastructure Scaling*

The main reason why Cloud Providers can offer application hosting at relatively low cost is the economies of scale they achieve through the concentrated deployment of

[2] The New York Times, in an article from September 2012 [79] quotes findings by Koomey [131] estimating that in 2010 data centers accounted for between 1.1 and 1.5 % of worldwide electricity use; for the US the estimate was between 1.7 and 2.2 %.

large numbers of Commercial-off-the-Shelf (COTS) PMs connected via high speed networks, also realized via COTS equipment. The large scale of cloud environments means that, on the one hand, resource management systems have a significant degree of flexibility in allocating resources, but, on the other hand, scaling the resource management processes themselves becomes an important consideration.

#### *2.4.2 Virtualization*

Virtualization refers to the process of creating an emulation of a hardware or software environment that appears to a user as a complete instance of that environment. Virtualization software is commonly used in cloud environments to create virtual machines (emulated computing servers), virtual storage disks and, sometimes, to create virtual networks or virtual data centers (the latter is termed network virtualization [21]). In this paper we refer to these collectively as *Virtual Infrastructure (VI)*. From the resource management perspective, virtualization serves the purpose of supporting the dynamic *sharing* of data center infrastructure between cloud hosted applications.

#### *2.4.3 Virtual Machine Migration*

A consequence of server virtualization is that VM data and execution state can be readily embodied in a set of files. Thus, a VM can have its execution externally suspended, be transferred to another physical server, and restarted with the same state—a process termed *live migration*. Placement of new VM instances, together with VM live migration provides the Cloud Provider with the ability to *dynamically adapt* resource allocations, for example by moving VMs to PMs with available resources when demand increases, or consolidating VMs on a smaller number of PMs in order to minimize energy usage.

#### *2.4.4 Equipment Power State Adjustment*

Energy use minimization has become an important objective for Cloud Providers. Responding to this, processor manufacturers and equipment vendors have developed energy efficient hardware and provide software interfaces that can be used to configure the energy consumption profiles of data center equipment. For example, many processors can now be configured to slow down their clock speeds [a process termed Dynamic Voltage Scaling (DVS)], are optimized for virtualization performance, and have the ability to operate at higher temperatures (reducing the need for costly cooling in the data center). Many of these facilities are explicity configurable, typically via an implementation of the standardized Advanced Configuration and Power Interface (ACPI) [2]. This configurability means that power management capabilities like DVS can be subsumed into the broad cloud resource management process. For a survey of technologies for energy use minimization by processes, as well as by storage devices, cooling systems and data center networks see Jing et al. [118].

## 2.5 Resource Management Functions

In this section we outline the main functionalities embodied in resource management systems for cloud environments. A decomposition into a conceptual framework depicting logical functional elements and their interactions is provided in Fig. 2. These functional elements should coordinate to provide a complete resource management solution that monitors and controls different (physical and virtualized) resource types in line with management objectives.

In Fig. 2 the functional elements are mapped to the Cloud Provider and Cloud User roles in line with an IaaS cloud offering. The Cloud Provider is responsible for monitoring the utilization of compute, networking, storage, and power resources and for controlling this utilization via global and local scheduling processes. The Cloud User monitors and controls the deployment of its application modules on the VI it leases from the Cloud Provider and controls the workload it receives from End Users of its application(s). Both Cloud Provider and Cloud User may have the ability to dynamically alter the prices they charge for leasing of VI and usage of applications, respectively. The responsibilities of the Cloud User are broadly similar to the tasks involved in managing dedicated infrastructure, but with the additional flexibility to request or release VI resources in response to changing application demands, or indeed, to changes in prices offered by the Cloud Provider. Finally, the End User has limited responsibility for resource management; they generate workload requests (possibly influenced by dynamic prices quoted by the Cloud User) and in some cases exert control of where and when workloads are placed.

The framework is depicted from an IaaS perspective. However, it is also applicable to the PaaS and SaaS perspectives—the functional elements remain the same, but responsibility for provision of more of them rests with the Cloud Provider. In the PaaS case, the role of the Cloud User is split into a Platform Provider and an Application Provider, where the degree of resource allocation responsibility falling on each varies depending on the scope of the provided platform. For example, with *Microsoft Azure* [157] Application Providers are responsible for controlling the admission and dispatch of workload requests, whilst with *Google App Engine* [85] the platform automatically handles demand scaling, launching application instances transparently as workload volumes increase. In the SaaS case, the Platform and Application Provider are typically the same organization, and often that organization is also the Cloud Provider; such an organization would then have responsibility for all resource management functionality. We now introduce the scope of the functional entities in more detail; later, in Sect. 3, we survey the state-of-the-art in techniques for their realization.

### *2.5.1 Global Scheduling of Virtualized Resources*

The Cloud Provider must support VI either directly leased by Cloud Users (for IaaS clouds), or utilized to support cloud-hosted applications (for PaaS/SaaS clouds). Global scheduling relates to the *system-wide* monitoring and control of virtualized and underlying physical resources in line with the Cloud Provider’s management objectives. It encompasses admission control of requests from IaaS Cloud Users for

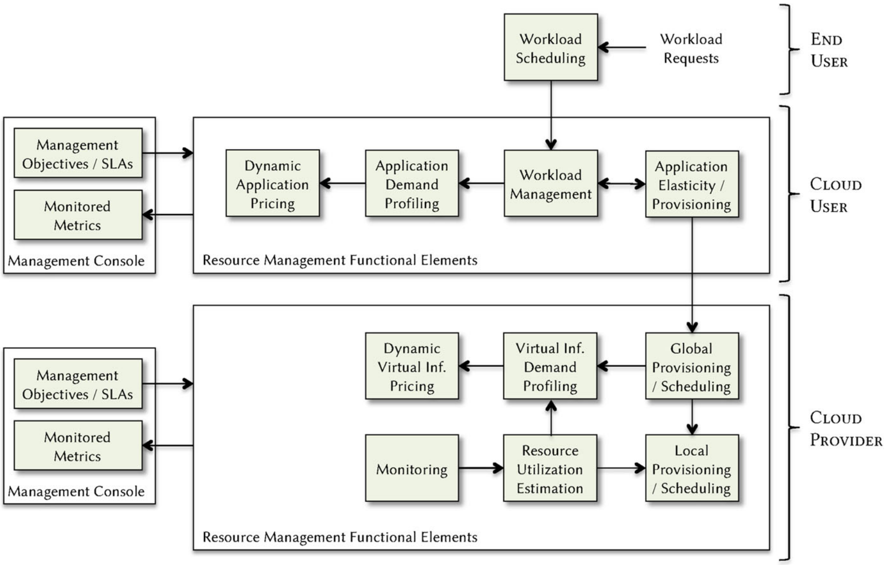
Workload Scheduling
Workload Requests
End User
Management Objectives / SLAs
Monitored Metrics
Management Console
Dynamic Application Pricing
Application Demand Profiling
Workload Management
Application Elasticity / Provisioning
Resource Management Functional Elements
Cloud User
Management Objectives / SLAs
Monitored Metrics
Management Console
Dynamic Virtual Inf. Pricing
Virtual Inf. Demand Profiling
Global Provisioning / Scheduling
Monitoring
Resource Utilization Estimation
Local Provisioning / Scheduling
Resource Management Functional Elements
Cloud Provider

**Fig. 2** Conceptual framework for resource management in a cloud environment. *Arrows* represent the principal information flows between functional elements. The *diagram* depicts the responsibilities of the actors (End User, Cloud User, Cloud Provider) in an IaaS environment. In the case of a PaaS or SaaS environment the partitioning of responsibilities is different

VI deployment, the initial placement of VI on physical infrastructure, and the adaptation of this placement over time.

### 2.5.2 Resource Demand Profiling

To be effective, resource management needs to achieve an appropriate balance between *reactivity*, i.e., adjusting resource allocations in response to detected changes in demand, and *proactivity*, i.e., adjusting resource allocations in response to predicted demand, with predictions typically being based on historical measurements. We term the latter Demand Profiling, and our conceptual framework includes a VI Demand Profiler, which assesses demand patterns for VI mainly as an input to global scheduling and VI pricing, and an Application Demand Profiler, which assess demand patterns for individual applications that, similarly, is mainly used as input to application workload management and pricing.

### 2.5.3 Resource Utilization Estimation

Resource management decisions by the Cloud Provider and the Cloud User require accurate estimations of the state of the physical and virtualized resources required to deliver cloud hosted applications. State estimation for compute, network, storage and power resources are provided by the Resource Utilization Estimation functional element; it provides input into cloud monitoring and resource scheduling processes.

### *2.5.4 Resource Pricing and Profit Maximization*

From a business perspective cloud computing services are commonly viewed as a means of delivering utility computing—resources required to deliver computing services are metered, based on duration of use and/or on usage level. Current IaaS offerings typically charge for VMs or VI on a lease duration basis, with different rates for differently dimensioned resources (see, e.g., *Amazon EC2* pricing [13]). PaaS offerings typically charge based on a combination of duration-based flat rates and usage-based rates for bandwidth, storage and API calls/transactions (see for example *Google AppEngine* pricing [84] and *Microsoft Azure* pricing [176]).

Offering cloud services in a usage-based, static pricing model remains the predominant business model for IaaS and PaaS providers. However, there has been a recent move towards *dynamic* pricing of cloud resources, notably by Amazon via their *Amazon Spot Instances* offering [14]. By lowering prices when their data centers are lightly loaded and *vice versa* Cloud Providers hope to encourage greater levels of usage, thus increasing their resource utilization and, by extension, maximizing their profit. Similarly, Cloud Users seek to obtain the lowest possible prices for the resources they lease, thus minimizing their costs and ultimately maximizing their profits. Assuming that dynamic pricing becomes increasingly prevalent, our framework includes VI Pricing and Application Pricing functional elements, both requiring accurate demand profile input in order to maximize their impact.

### *2.5.5 Local Scheduling of Virtualized Resources*

Physical resources employ local resource management and scheduling functions to decide how to share access between the virtualized resources placed on them. For compute resources a hypervisor deployed on a PM will control the sharing of CPU, local I/O and network interfaces between the VMs it hosts. Depending on hypervisor capabilities, Local Provisioning/Scheduling functional elements may have the ability to dynamically adjust the share of resources allocated to VMs. Similarly, controllers of distributed storage systems will control access to storage capacity, network admission controllers and multi-path forwarding controllers share network resources, and power management systems can control power utilization through device control and server consolidation. In all cases the operation of Local Provisioning/Scheduling functional elements should be coordinated so that they collectively satisfy management objectives.

### *2.5.6 Application Scaling and Provisioning*

Cloud users who deploy applications on VI leased from an IaaS Cloud Provider typically need to control how the software modules that comprise their application(s) are deployed over this VI, whilst in PaaS/SaaS clouds, the Cloud Provider directly assumes this responsibility. For applications comprised of multiple software modules, and for groups of applications that are closely coupled, their initial placement on the VI must be ascertained. In particular, for computation-intensive applications, such as MapReduce implementations, software modules may need to be configured for optimal resource usage given the characteristics of the VI they will execute on. Furthermore, as application demand changes, the placement and configuration of application modules may be dynamically changed—a process whose success is contingent of accurate estimation of future demand. This functionality is realized by the Application Scaling and Provisioning functional element. In some cases, Cloud Users may offer End Users the capability of deploying bespoke software modules on their VI, who will need to make the decision as to whether they have sufficient resources available to deploy such modules. Finally, in response to changing application demand, the amount of VI leased supporting the application will need to be dynamically scaled.

### *2.5.7 Workload Management*

Once application modules have been placed on the VI a Cloud User must exert control over the workload requests generated by End Users. Initially, the Cloud User must decide whether a given workload request can be processed in line with the Cloud User's management objectives. Such management objectives may state, for example, that all accepted workloads should satisfy specified availability and performance SLOs. Once a workload is accepted, a related task is to dispatch it to one of the potentially many instantiations of the software modules that manage its execution, again in line with the Cloud User's management objectives.

### *2.5.8 Cloud Management Systems*

Management systems are used by the Cloud Providers and, sometimes, Cloud Users to provide the reporting and command-and-control interface used by human system administrators. They are used to define *management metrics* that provide feedback regarding the operation of the resource management system. These metrics will use the resource utilization estimates and demand profiles, together with other system information, all of which can be aggregated and fused to provide meaningful monitoring information. In particular, of critical importance is monitoring of the degree to which the system meets its SLAs. For the Cloud Provider these SLAs will be with its customers (the Cloud Users), whereas the Cloud User may choose to monitor the performance of the Cloud Provider in meeting SLAs and/or may monitor its own adherence to the SLAs it has agreed with its own customers (the End Users). As discussed in Sect. 2.2, *management objectives* define the strategies according to which cloud resources are allocated to applications.

## 3 Cloud Computing Resource Management: State-of-the-Art

Many academic and industrial researchers are actively addressing cloud resource management. Given the volume of literature on the topic we do not provide an exhaustive survey; rather, we discuss a representative sample of works, comprising what we believe are those most significant. Our review is structured around the eight resource management activities outlined in Sect. 2.5.

## 3.1 Global Scheduling of Virtualized Resources

Global scheduling involves a system-wide perspective on the allocation of the physical and virtualized resources that comprise a cloud environment. Recent years have seen a proliferation of proposals for frameworks and techniques for global scheduling. Most proposals originate from academic researchers; notable exceptions include the discussion by Gulati et al. [89] of the VMware *Distributed Resource Scheduler* and *Distributed Power Management* systems, which provide centralized control of a cluster of virtualized servers, performing VM placement, as well as load balancing and VM consolidation via live migration. Similarly, Wilkes [119] provides a high-level overview of two generations of Google's cluster management systems, focusing on design goals, management objectives and unsolved challenges, whilst Zhu et al. [269] describe Hewlett Packard's *1000 Islands* architecture, which utilizes three types of controllers that operate over different scopes and timescales. Frameworks from the academic community that incorporate controllers directly or indirectly cooperating to achieve global scheduling include: *vManage* [135], which loosely couples controllers for power and cooling management with controllers for virtualization management; the system proposed by Beloglazov and Buyya [23], which comprises controllers that cooperatively control VM placement and live migration; and *TROPIC* [148], which performs transactional cloud resource orchestrations that enforce safety, provide concurrency and increase system robustness.

The majority of proposed frameworks utilize resource controllers that are *centralized* in the sense that they have full control of the allocation of a given set of resources. Such solutions can reasonably be expected to scale only to systems incorporating on the order of thousands of servers. To meet the scalability challenge some frameworks (e.g., [120, 160]) propose hierarchical organization of controllers. A small number of frameworks incorporate *decentralized* controllers—three examples being those proposed by Wuhib at al. [244, 246], Feller et al. [69] and Schwarzkopf et al. [187]. We now review specific proposals for resource management techniques (Table 1 provides a summary comparison), grouping them into techniques providing initial placement/provisioning of VMs, techniques for dynamic placement of VMs, techniques for VM placement that consider sharing of networking resources, and techniques that consider global energy use minimization:

### *3.1.1 Virtual Machine Placement and Provisioning*

Whenever a request for the provisioning of one or more VMs is made by a Cloud User, the Cloud Provider's resource management system schedules the VMs by placing them onto PMs—the decision should help achieve the Cloud Provider's current management objective(s). Initial work on the VM placement problem typically assumed that VMs are assigned static shares of a PM's managed resources (typically CPU and/or memory). Given this, the placement of VMs onto PMs is

**Table 1** Comparing works in global scheduling for cloud environments

| Publication | | Characteristics | | | | Architecture | | |
|---|---|---|---|---|---|---|---|---|
| Reference | Year | SP | DP | NA | EA | Cent. | Hier. | Dist. |
| Chase et al. [43] | 2001 | | | | × | × | | |
| Bobroff et al. [31] | 2007 | × | × | | × | × | | |
| Gmach et al. [80] | 2008 | × | × | | × | × | | |
| Cardosa et al. [39] | 2009 | × | | | × | × | | |
| Li et al. [140] | 2009 | × | × | | × | × | | |
| Verma et al. [221] | 2009 | × | | | × | × | | |
| Zhu et al. [269] | 2009 | × | × | | | | × | |
| Beloglazov and Buyya [23] | 2010 | × | × | | × | | × | |
| Jung et al. [120] | 2010 | × | × | | × | | × | |
| Meng et al. [153] | 2010 | × | | | | × | | |
| Meng et al. [155] | 2010 | × | | × | | × | | |
| Parolini et al. [171] | 2010 | | | | × | | × | |
| Speitkamp and Bichler [202] | 2010 | × | × | | × | × | | |
| Yazir et al. [259] | 2010 | × | × | | | | | × |
| Bin et al. [29] | 2011 | × | × | | | × | | |
| De and Roy [59] | 2011 | × | | | | × | | |
| Kumar et al. [135] | 2011 | × | × | | × | × | | |
| Jayasinghe et al. [115] | 2011 | × | | × | | | × | |
| Koslovski et al. [132] | 2011 | × | | × | | × | | |
| Moens et al. [160] | 2011 | × | | | | | × | |
| Panigrahy et al. [169] | 2011 | × | | | | × | | |
| Sharma et al. [190, 191] | 2011 | × | × | | | × | | |
| Wang et al. [228] | 2011 | × | | × | | × | | |
| Wilcox et al. [239] | 2011 | × | | | | × | | |
| Xu and Fortes [250] | 2011 | × | × | | × | × | | |
| Xu and Li [252] | 2011 | | × | | × | × | | |
| Alicherry et al. [10] | 2012 | × | | × | | × | | |
| Biran et al. [30] | 2012 | × | | × | | × | | |
| Breitgand and Epstein [34] | 2012 | × | | × | | × | | |
| Feller et al. [69] | 2012 | × | × | | | | | × |
| Ghorbani and Caeser [77] | 2012 | | × | × | | × | | |
| Giurgiu et al. [78] | 2012 | × | | × | | × | | |
| Gulati et al. [89] | 2012 | × | × | | × | × | | |
| Guo et al. [96] | 2012 | | × | × | | × | | |
| Hu et al. [107] | 2012 | × | | × | | × | | |
| Jiang et al. [117] | 2012 | × | | × | | × | | |
| Liu et al. [148] | 2012 | × | × | × | | × | | |
| Konstanteli et al. [130] | 2012 | × | | | × | × | | |
| Viswanathan et al. [223] | 2012 | × | | | | | × | |
| Wen et al. [237] | 2012 | | × | × | | × | | |

**Table 1** continued

| Publication | | Characteristics | | | | Architecture | | |
|---|---|---|---|---|---|---|---|---|
| Reference | Year | SP | DP | NA | EA | Cent. | Hier. | Dist. |
| Wuhib et al. [247] | 2012 | × | × | | | | | × |
| Wuhib at al. [244] | 2012 | × | × | | × | | | × |
| Zhang et al. [262] | 2012 | | × | | | × | | |
| Al-Haj and Al-Shaer [9] | 2013 | | × | × | | × | | |
| Esteves et al. [66] | 2013 | × | | × | | × | | |
| Foster et al. [70] | 2013 | | × | | | × | | |
| Shi et al. [194] | 2013 | × | × | | | × | | |
| Schwarzkopf et al. [187] | 2013 | × | × | | | | | × |
| Rabbani et al. [178] | 2013 | × | | × | × | × | | |
| Roytman et al. [183] | 2013 | × | | | × | × | | |
| Wuhib et al. [246] | 2013 | × | × | × | × | | | × |
| Zhani et al. [268] | 2013 | × | × | × | × | × | | |

Most solutions are based on centralized control, some use hierarchical control schemes, and a few general distributed solutions. Many solutions combine static and dynamic VM placement

*SP* static placement, *DP* dynamic placement, *NA* network aware, *EA* energy aware, *Cent.* centralized, *Hier.* hierarchical, *Dist.* distributed

related to the vector bin packing problem, which can be used to model static resource allocation problems where the resources used by each item are additive. To achieve server consolidation, for instance, the optimal placement is one where the items (i.e., VMs) are packed into a minimum number of bins (i.e., PMs), such that the vector sum of the items received by any bin does not exceed the bin’s (resource) limit. The vector bin packing problem and its variants are NP-hard problems [54]; thus, heuristic algorithms are commonly proposed and evaluated.

Most proposed heuristics are based on greedy algorithms using simple rules, such as First Fit Decreasing (FFD) and Best Fit Decreasing. Panigrahy et al. [169] study variants of the FFD algorithm and propose a new geometric heuristic algorithm which scales to large data centers without a significant decrease in performance. Wilcox et al. [239] propose a new generic algorithm, Reordering Grouping Genetic Algorithm (RGGA), which they apply to VM placement. Other works proposing bin packing heuristics for placement of VMs on PMs include Jung et al. [121], Gupta et al. [98], and Li et al. [140].

Many variants of the VM placement problem have been studied. Cardosa et al. [39] propose a solution for VM placement that accounts for the ability of some hypervisors to dynamically allocate a pool of unreserved CPU and memory resources between a set of contending VMs. Viswanathan et al. [223] provide a solution for private clouds in which VMs are placed on those PMs that already host VMs with complementary workloads. Meng et al. [153] propose a VM placement solution in which multiple VMs are consolidated and provisioned together, in order to exploit the statistical multiplexing amongst the workload patterns of the individual VMs.

More recent works recognize that, increasingly, provisioning requests from Cloud Users involve *sets of VMs*, rather than single VMs. In such cases, placement of the set of VMs is not only constrained by the collated resource requirements, but also by a range of possible placement constraints indicated by the Cloud User. For example: Shi et al. [194] consider a range of constraints relating to full deployment, anti-colocation and security; Jayasinghe et al. [115] consider demand constraints (i.e., lower bounds for VM resource allocations) and placement constraints (i.e., colocation and anti-colocation); whilst Konstanteli et al. [130] consider similar constraints, adding some that indicate whether a set of VMs can be deployed across more than one data center in a geo-distributed cloud environment. Breitgand and Epstein [35] consider elastic services realized by a fixed constellation of VMs, but where the number of VM instances and their size varies in line with demand. They define an "Elastic Services Placement Problem (ESPP)" that extends the Generalized Assignment Problem (GAP), but which is harder to solve directly as it does not admit a constant factor approximation. Instead, they transform ESPP into an analogous multi-unit combinatorial auction, which they solve using a column-generation method that provides good solutions for realistic-sized placement problems in reasonable time. A different perspective is taken by De and Roy [59], who outline a solution that focuses on the resiliency of VMs comprising a multi-tiered application, by seeking to spread the VMs across as many PMs as possible. Bin et al. [29] addressed a similar problem—that of making VMs resiliant to *k* PM failures, so that if up to *k* such failures occur in the cluster there is a guarantee that this VM is relocated to another PM without relocating other VMs.

### *3.1.2 Dynamic Virtual Machine Placement*

Following initial placement, VMs can be rescaled (in a limited way) as demand for the applications they host changes [39, 191]. However, from a global scheduling perspective a more powerful tool is *live migration* of VMs [52], whereby running VMs are paused, serialized and transferred to a different PM, where they are once again scheduled for execution. Live migration offers many benefits, including improving fault management, simplifying system maintenance and, from the resource management perspective, enabling global scheduling objectives, such as balancing load across PMs and consolidating VMs on a minimal number of PMs. Many studies have assessed the overhead of VM live migration (e.g., [7, 52, 126, 164]), and many proposals for optimizing the process have been made (e.g., [22, 105, 146, 151, 242]).

Bobroff et al. [31] propose a first-fit heuristic that dynamically remaps VMs to PMs in a manner that minimizes the number of PMs required to support a workload at a specified allowable rate of SLA violations; they present an extensive evaluation of the approach based on production traces. Sharma et al. [190, 191] propose *Kingfisher*, a set of techniques for VM rescaling, replication and live migration, where the problems are formulated as Integer Linear Programming problems for which greedy heuristic solutions are proposed. Zhang et al. [262] address control of VM migrations in scenarios where the Cloud Provider intentionally overcommits their PM resources to VMs; their technique minimizes the number of VM migrations and migrates VMs with strong utilization correlation to different PMs in order to minimize the risk of future PM overload. Xu and Li [252] propose a novel formulation of the VM live migration problem: they cast it in terms of the Stable Marriages problem [71], which they argue is more appropriate than optimization based formulations that dictate "an arbitrary way of resolving the conflicts of interest between different stakeholders in order to achieve a global notion of performance optimality." They propose a polynomial time algorithm that generates an egalitarian stable matching between VMs and PMs that they show balances VM performance with migration overhead. From a more practical perspective, researchers have observed [62, 186] that in public clouds the same VM instance types can exhibit significantly different performance levels depending on characteristics (processor, I/O speed etc.) of the hosting PM; heterogeneity should thus be taken into account by VM migration controllers. Guo et al. [96] address the use of VM migration in the context of cloud bursting for enterprise applications, focussing in particular on the decision regarding which VMs to migrate to an external cloud environment and when. Shifrin et al. [196] also address cloud bursting; they analyse a more generalised task scheduling problem formulated as a Markov Decision Process and show that a threshold based scheduling policy is optimal. Foster et al. [70] also address VM migration and VM consolidation, but from the perspective of how these are used to satisfy dynamically changing management objectives that are enforced via policies. Their study addresses how to change between the confliciting goals of minimizing SLA violations (by spreading VMs across more PMs) and minimizing power consumption (by consolidating VMs on fewer PMs).

While the above solutions are centralized in nature, other works propose decentralized algorithms to control the live migration process. Yazir et al. [259] propose a scheme wherein autonomous "node agents," each associated with a PM, cooperate to control the VM migration process. The node agents employ Multiple Criteria Decision Analysis (MCDA) using the PROMETHEE method. Whilst the approach is promising, its scalability is only evaluated for a relatively small problem size (a data center with 2,500 PMs supporting 25,000 VMs). Yanggratoke et al. [257] and Wuhib et al. [246, 247] propose a decentralized solution, utilizing a gossip protocol that realizes an efficient heuristic that again minimizes the number of migrations; their approach is shown to scale to problem sizes above 150,000 PMs and 350,000 VMs. Similar to Foster et al. [70] they also consider how different management objectives can be pursued by resource controllers.

These works focus on calculating new VM placement plans that meet changing demand patterns. Whilst some authors take into account the desire to minimize the number of VM migrations required to generate the new placement plan, few have focussed on the challenge of planning the sequence of migrations in a manner that does not lead to violation of performance and security constraints. Ghorbani and Caeser [77] provide a solution for determining the order of VM migrations considering that there is limited bandwidth available for the migration process. Al-Haj and Al-Shaer [9] formulate VM migration planning as a constraints satisfaction problem, which they solve using a Satisfiability Modulo Theory (SMT) solver. This approach allows them to take into account a wide range of constraints relating to the migration process itself and the resulting VM placement plan. However, the computational overhead of using an SMT solver may mean that the approach would only be feasible for relatively small problem sizes.

Zhang et al. [266] investigate dynamic VM placement in geo-distributed clouds. The goal is to minimize hosting cost while achieving SLO objectives under changing demand patterns and resource costs. Their solution is based on control- and game-theoretic models.

### *3.1.3 Network-Aware Virtual Machine Placement*

VMs access a PM's network interface to communicate with other application and system components. Hypervisors typically treat a PM's network interfaces as unmanaged resources—they do not provide a guaranteed allocation to individual VMs, relying on statistical multiplexing and that fact that VMs are unlikely to simultaneously maximize their use of their nominally assigned bandwidth [34]. However, this means that there is potential for VMs to affect each other's performance due to contention for network interface resources [186]. Recent works have addressed how to mitigate this risk by taking this contention into account when placing VMs. Wang et al. [228] apply the Stochastic Bin Packing problem formulated by Kleinberg et al. [127] for bandwidth allocation to bursty network traffic flows to VM placement. They treat a VM's demand as a random variable and propose an online VM placement algorithm where the number of PMs required to place a set of VMs on is within $(1+\epsilon)(1+\sqrt{2})$ of the optimum for any $\epsilon > 0$. Breitgand and Epstein [34] improve the result of Wang et al. by an algorithm with a competitive ratio of $(2+\epsilon)$ for any $\epsilon > 0$.

Intuitively, it is beneficial to place VMs that interact with each other in close proximity in the data center network topology. Meng et al. [155] formulate heuristics that seek to minimize the aggregate traffic rates perceived by switches in the data center network by placing VMs with large mutual bandwidth in close proximity to each other. Evaluation results using production traces show a significant performance improvement. Similar approaches to minimizing data center traffic have been proposed by Jayasinghe et al. [115], Shrivastra et al. [198] and Wen et al. [237]. Biran et al. [30] address network-aware VM placement; their Min Cut Ratio-aware VM Placement (MCRVMP) formulation and heuristics to solve it incorporate constraints evolving from complex network topologies and dynamic routing schemes, but also take into account the time varying nature of traffic demands in a manner that finds placements that have sufficient spare capacity across the network to absorb unpredicted traffic bursts. Jiang et al. [117] address data center traffic minimization via an online algorithm that exploits multi-path routing capabilities and live migration of VMs. They argue that it is important to not only optimize the placement of VMs but also jointly optimize the routing between PMs hosting interacting VMs. Alicherry et al. [10] extend the network-aware VM placement problem to geo-distributed clouds, proposing algorithms that select which data center to place a VM in, and, within that data center, which PM to place it on. Hu et al. [107] approach network-aware placement from the perspective of a

Cloud User who has leased a group of VMs with set inter-VM bandwidth limits; their *vBundle* system facilitates the exchange of bandwidth allocations between VMs in a group as traffic patterns evolve.

Increasingly, Cloud Users request a complete logical network of interacting VMs (which we term VI) from the Cloud Provider. In such cases, the Cloud Provider must place VMs in a manner consistent with the SLAs associated with such a VI request. Ballani et al. [20] discuss abstractions for VI requests by the Cloud User and evaluate an implementation called *Oktopus*, which they show yields significantly improved (and better predictable) performance for the Cloud User. An alternative VI abstraction is discussed in [95]. Benson et al. [25] evaluate *CloudNaaS*, a system that allows Cloud Users deploy VIs that incorporate many network functions, including custom addressing, service differentiation and interposition of middle-boxes; whilst Chen et al. [47] allow Cloud Users to control the routing between communicating VMs, in accordance with the Cloud Provider's general policies. Esteves et al. [66] propose a higher-level abstraction for VI specification in which applications specify objectives realised via a set of VI allocation actions, which may be adjusted over time so that the requirements of the different applications supported by the data centre are balanced. Rabbani et al. [178] look at the placement of VI (which they term a Virtual Data Center (VDC)) in a data centre, taking account not only of placement of VMs, but also the allocation of network resources at swtiches and on data center network links. Giurgiu et al. [78] address the placement of a VI in large data centers considering both compute and network resource requirements, but also taking into account availability requirements. Zhani et al. [268] develop techniques to allow the Cloud Provider dynamically adjust VI resource allocations (through scaling and VM migration) over time so that the Cloud Providers revenue is increased and their energy costs are minimized. Finally, Koslovski et al. [132] propose solutions for allocation of resources in support of VI deployment in geo-distributed clouds.

### *3.1.4 Energy-Aware Virtual Machine Placement*

As outlined in Sect. 2.4, cloud resource management systems can avail of capabilities such as processor DVS and VM live migration to minimize energy consumption in the data center. Jing et al. [118] provide a detailed review of the state-of-the-art in this broad area, so we limit ourselves to a brief review of representative works addressing global scheduling aspects. One of the earliest studies in the area was performed by Chase et al. [43], who addressed "right-sizing" of data centers—dynamically resizing the active PMs as load varies—under the control of a market-based resource allocation algorithm. Work on the data center right sizing problem has continued from both algorithm development and system implementations perspectives; good examples of recent efforts are [48, 73, 106, 142]. Many of these works extend beyond simple power on/off models to consider modulation of PM processor speeds via DVS and similar technologies.

In parallel to efforts relating to data center right-sizing, the introduction of virtualization technologies and support for VM live migration led researchers to consider the *consolidation* of VMs onto a minimal number of PMs. A very large body of work has emerged on this topic. The problem is normally formulated as a bin packing problem, as for example in [23, 140, 204]. Recent work by Roytman et al. [183] take account of the fact that virtualization does not prevent all forms of resource contention on a PM (e.g., for shared caches and memory bandwidth); they propose VM consolidation algorithms that balance the level of consolidation with the performance degradation arising from resource contention. Some works augment the consolidation objective with the goal to minimize reallocations (live migrations) when new VMs are admitted, either by adding a second objective to the problem formulation [80], or by constraining the number of reallocations [202]. Other works, e.g., Verma et al. [221], focus less on the optimal consolidation at a given point in time, considering instead the best candidates for consolidation over a longer period during which VM resource demands can vary significantly. An issue closely linked to the consolidation of VMs on PMs is the effect this process has on the energy expended by data center cooling systems. Some researchers have considered strategies for joint optimization of PM energy use and cooling system energy use; see, e.g., [58, 171, 232, 250]. As noted by Wilkes [119], it is extremely challenging to jointly optimize allocation of server, network, and cooling resources since in data centers networking, power distribution and cooling system topologies are typically deployed in a non-coordinated manner. However, this problem is likely to be mitigated by the move towards "data center in a box" modular architectures, in which data centers are built from smaller modules with coordinated management facilities.

### 3.2 Resource Demand Profiling

Cloud environments predominantly host services accessed over the Internet. Given that it is well known that the workload of web applications varies dynamically over multiple time scales [56] considerable attention has been given to developing techniques that provide demand profiling for resource management purposes. Approaches can be broadly classified as being model-driven, where a pre-identified application model is used to profile demand, or model-free, where statistical forecasting techniques are used to make predictions on future workload intensity. Examples of earlier work focussing on demand profiling for web applications include: Chandra et al. [42], who present a prediction algorithm that estimates the workload parameters of applications based on time series analysis of request arrivals (which are modeled as an AR(1) process [33]); Bennani and Menasce [24], who use multi-class queuing networks to model the workload of dedicated data center hosted application environments; and Urgaonkar et al. [218], who also propose a queuing network model, but do so for multi-tier applications where individual queues represent different tiers, and the effects of session-based workloads, concurrency limits, and caching are modeled. A more recent work by Ali-Eldin et al. [11] uses adaptive proactive controllers that estimate the future load for a service in order to support service elasticity. Similarly, Nguyen et al. [165] address proactive elasticity controllers, but they focus on decisions to start up new application server instances and dynamic VM cloning to minimize application startup times.

Approaches to demand profiling developed in the early 2000s for web applications have been recently tailored to cloud-hosted applications. Many proposed cloud resource management frameworks incorporate demand profiling functions; e.g., the *Mistral* framework [120] incorporates a Workload Predictor functional element that uses an Autoregressive Moving Average (ARMA) filter [33] for time series analysis to predict demand. Singh et al. [200] describe *Predico*, a "What If" analysis tool that uses demand profiles constructed as networks of queues that can be used to analytically model application behaviour; these models are then used to provide behavior predictions in response to system administrator queries formulated in a bespoke query language. A similar approach is taken by Padala et al. [168] who use an ARMA filter in modeling the relationship between an application's resource allocations and its normalized performance, under the assumption that such a linear model provides a reasonable local approximation of the actual non-linear, workload-dependent relationship in a virtualized environment. Gong et al. [83] also apply time series analysis: they use a combination of a Fast Fourier Transform (FFT) to detect repeating resource usage patterns for one subset of applications and a discrete-time Markov chain to build short-term demand predictions for the subset of applications that do not exhibit repeating patterns. This approach is further developed by the authors in Shen et al. [193], where the use of online adaptive padding and reactive error correction to mitigate under-estimation is described. The use of FFT and other signal processing techniques to detect application signatures has also been explored in other works, see, e.g, Gmach et al. [82] and Buneci and Reed [38].

The works cited above typically seek to model the relationship between the resources used by an application and that application's performance. However, demand profiling can extend beyond this, notably to characterizing an application's demand for power. Kansal et al. [123] describe the *Joulemeter* system which facilitates power metering of VMs, which can be used by a Cloud Provider as an input into resource management processes that seek to minimize data center energy usage. Smith et al. [201] argue that power usage models of this form could be used to create pricing models that encourage Cloud Users to create energy-efficient applications. Alternatively, Vasić et al. [219] propose the use of such models to provide applications with a "power budget" under the expectation that they will modify their behavior to stay within this limit. Lim et al. [141] address power budgeing in virtualized data centers; their solution, termed virtualized power shifting (VPS) coordinates the power distribution amongst groups of VMs within given peak power capacity.

### 3.3 Resource Utilization Estimation

Resource Management processes require utilization estimates for two main purposes: firstly, to assess the level of free resources of given types within the system, and, secondly, to profile the resource utilization patterns of individual workloads so that impact of allocating resources to these workloads can be assessed. The latter is especially relevant for workloads exhibiting business in resource utilization, where resource management processes often over-commit resources to workloads relying on statistical multiplexing to smooth out the average resource usage.

The majority of cloud resource management research papers assume that instantaneous and/or historical resource usage measurements are accurate and readily available. Indeed, commonly used tools such as *collectd* [55], *Nagios* [163] and *Ganglia* [74] provide the capability to monitor a wide range of monitored metrics, including compute, network and storage resource utilizations. One notable exception is Gulati et al. [89], who describe the particular CPU and memory related utilization metrics used by VMware's *Distributed Power Management (DPM)* to trigger management actions including VM migration and PM power-on. Recently, researchers have started to address the issue of estimating the utilization of micro-architectural resources such as shared processor caches, contention for which have been shown to negatively impact the performance of consolidated VMs [6, 86]. Profiling individual application or workload's resource utilization profiles is a more complex task, which has been generally addressed as part of resource demand profiling (see Sect. 3.2). Whilst this is primarily done for resource allocation purposes it is also important for assessing the amount of resources applications use as inputs into cost models used for pricing cloud services; to this end Gmach et al. [81] assess a number of approaches for apportioning costs based on resource utilization estimation.

### 3.4 Resource Pricing and Profit Maximization

One of the main benefits to a Cloud User in hosting applications on a public IaaS cloud is that it lowers its capital and operational costs. However, given the sometimes complex pricing models offered by public IaaS Cloud Providers, assessing the monetary gains of migrating to cloud hosting can be complex. Khajeh-Hosseini et al. [124, 125] describe an approach to systematically provide a cost-benefit analysis of migrating enterprise services to hosting on a public cloud. Other researchers address more specifically the problem of minimizing costs when dimensioning VI to support deployment of a cloud hosted application. For example: Sharma et al. [191] describe a technique to identify the set of VMs that minimizes cost whilst satisfying a given workload requirement; Chaisiri et al. [41] address cost minimization through the formulation of a stochastic programming model, proposing a provisioning algorithm that balances longer term, less costly use of reserved resources with shorter-term, more costly, on-demand resource usage; and Wang et al. [227] address the cost-minimizing provisioning of cloud resources for geo-distributed provisioning of media streaming services, taking into account temporal and regional demand patterns. Wang et al. [226] explore strategies that help a Cloud User to decide whether, in the face of changing application behavior, it is better to reserve discounted resources over longer periods or lease resources at normal rates on a shorter term basis. Ishakian et al. [112] explore the potential for Cloud Users to express workload flexibilities (using Directed Acyclic Graphs), which can be used by Cloud Providers to optimise cloud resource usage and provide these Cloud Users with discounted prices. Wang et al. [229] propose a *cloud brokerage service* that acts as an intermediary between the Cloud User and the

Cloud Provider, exploiting the price benefits of long-term resource reservations and multiplexing gains to offer discounts to Cloud Users.

Other researchers have addressed pricing from the opposite perspective—that of the Cloud Provider. Sharma et al. [192] apply financial option theory from economics to guide pricing strategies for cloud resources. Xu and Li [253] highlight that in pricing cloud resources there is a need to balance the trade-off between perishable capacity (initialized resources generate no revenue) and stochastic demand (often influenced by prices). They formulate an infinite horizon stochastic demand program for revenue maximization and present numerical results that demonstrate its efficacy. Whilst pricing of resource in a public IaaS cloud is typically considered only in terms of VM numbers, size and lease terms/period Niu et al. [166] identify the potential for Cloud Providers to price bandwidth guarantees that are being enabled by advances in data center engineering and propose a suitable pricing model.

As discussed in Sect. 2.5.4, in 2009 Amazon introduced dynamically priced *Amazon Spot Instances*; this motivated efforts by researchers to reverse engineer the pricing strategy for these instances [5, 114]. Furthermore, the success of spot instances has encouraged researchers to propose new models for cloud resource pricing and provisioning. Wang et al. [230] present an economic analysis of the long-term impact of spot instance pricing on present and future revenue from the Cloud Providers perspective. Ben-Yehuda et al. [26] have envisioned the evolution of “Resource-as-a-Service (RaaS)” cloud model, wherein resources are continuously rented by Cloud Users at a low level of granularity and in accordance with dynamic, market-driven prices. Other researchers, e.g., Zaman and Grosu [261] and Wang et al. [231], have also explored the potential for using combinatorial auction techniques to regulate prices and allocate resources between Cloud Providers, Cloud Users and, potentially, End Users.

## 3.5 Local Scheduling of Virtualized Resources

In this section we provide an overview of a range of techniques developed to control the localized scheduling of access by VMs to compute, network and storage resources.

### *3.5.1 Local Scheduling of Virtual Machines*

Gulati et al. [89] provide an overview of how compute resources are modeled for the purposes of local scheduling in VMWare’s product suite; similar approaches are adopted in other commercial and in open-source hypervisors such as *Xen* [248]. They outline how users can specify *reservations* (minimal allocations of a resource), *limits* (maximum allocations of a resource) and *shares* (weights governing the actual allocation when the resource is oversubscribed and other VMs are contending for the allocation). They outline how this model is generalized to a pool of resources provided by a cluster of PMs. Solutions of this form seek to provide an appropriate balance between static, user-specified resource allocations and automated allocation of resource amongst contending VMs.

In contrast, academic researchers tended to focus on providing fully automated solutions, in which continuously monitored VM resource demands are inputs to a resource controller, which, cognizant of application SLOs, dynamically allocates resources to contending VMs. For example, Padala et al. [168] propose a control-theoretic approach involving a two-layer multi-input, multi-output (MIMO) controller that facilitates the use of application priorities for service differentiation under overload. Xu et al. [255] address a similar problem, but apply fuzzy-logic approaches (both fuzzy modeling and fuzzy prediction) to provide input to a resource controller. Two more recent works addressing variants of this problem are by Rao et al. [180], who apply reinforcement learning techniques; and Han et al. [100], who propose what they term as a lightweight resource scaling approach.

Other researchers have addressed the allocation of PM resources to VMs from an alternative perspective, namely the role this process can play in minimizing data center energy usage; we describe two notable examples of such works. Cardosa et al. [39] highlight that, at the time of writing, most work on power consolidation did not take into account the ability, as described above, for hypervisors to dynamically allocate resources; using a testbed comprised of *VMWare ESX* servers they show how a utility improvement of up to 47 % can be achieved. Urgaonkar et al. [216] describe a solution involving the use of Lyapunov Optimization to jointly maximize the utility of average application throughput and energy costs.

The works discussed above all address the explicit local *allocation* of resources to VMs. However, recently researchers have started to address other aspects that influence VM performance. Sukwong et al. [207] address the sequence of VM execution, which they show can significantly impact the response time for requests arrive concurrently for the different VMs hosted on a PM. They propose a local scheduler combining both compute resource allocation and control of VM execution sequencing. Another factor influencing the performance of VMs is contention for micro-architectural resources (such as shared caches and memory controllers) on multi-core PMs. Govindan et al. [86] highlight this issue and, focussing on shared processor caches, propose a non-intrusive technique for predicting the performance interference due to their sharing. Ahn et al. [6] propose the use of live migration to dynamically schedule VMs to minimize contention for different micro-architectural resource types. Kocoloski et al. [129] also address interference between VMs co-hosted on a PM, but focus on the co-location of VMs hosting High Performance Computing (HPC) applications and traditional cloud-hosted applications. Novaković et al. [167] describe DeepDive, a system design to transparently identify and mitigate interference between VMs in IaaS environments; its migitation strategies include altering the placement of VMs in order to isolate VMs that negatively impact on their co-hosted VMs; work with similar goals is also reported by Mukherjee et al. [162], who propose a software based probe approach for interference detection.

### *3.5.2 Local Scheduling of Network Access*

As outlined in Sect. 2.3.2 providing predictable latency and bandwidth is a significant challenge in the data center. From a local scheduling perspective TCP operates sub-optimally in virtualized environments. One well known issue is the TCP *Incast* pattern in which high bandwidth, low latency conditions lead TCP implementations to behave in a pathological manner in which “applications see *goodput* that is orders of magnitude lower than the link capacity” [44]. Gamage et al. [72] report on findings that indicate that high levels of VM consolidation on a PM increase CPU scheduling latencies, which in turn greatly increases the round-trip times for TCP connections. Given these, and related TCP shortcomings when used in the data center, a number of alternative transport protocols have been proposed and evaluated, e.g., multi-path TCP [179] and $D^3$, a deadline-aware control protocol [241].

Other researchers have addressed specific issues relating to deployment and migration of VMs. Voorsluys et al. [225] study the effects of VM live migration on the performance of applications in co-hosted VMs, finding that in some cases performance degradation can be appreciable. Kochut and Karve [128] propose optimizations for the transfer of VM image files during the provisioning process, whilst Peng et al. [172] address the same issue by proposing a virtual image distribution network that harnesses the hierarchical nature of the data center network topology. Curtis et al. [57] outline *Mahout*, a solution in which elephant flows are detected at the PM and a central, OpenFlow-like, controller provides traffic management functions to optimize the network to transfer these high-bandwidth flows.

Whilst the issues outlined above illustrate the potential effect of the network on VM performance the central task of apportioning available network resources to co-hosted VMs remains. This issue has received significant attention in the recent literature. As outlined by Popa et al. [174] sharing network resources is challenging because “the network allocation for a VM X depends not only on the VMs running on the machine with X, but also on the other VMs that X communicates with, as well as the cross traffic on each link used by X.” They highlight the trade-off between the degree of multiplexing of network links and the ability to provide minimum bandwidth guarantees to VMs and propose a “Per Endpoint Sharing” mechanism to explicitly manage this trade-off. Jeyakumar et al. [116] also address the network sharing problem, proposing *EyeQ*, a solution in which a portion (on the order of 10 %) of access link bandwidth is not allocated, providing a simple local solution that offers statistical guarantees on allocation of bandwidth to contending VMs. Kumar et al. [134] focus on data parallel applications and argue that network resources should be shared in a manner cognizant of data transfer patterns that can be adopted by parallel applications. Ballani et al. [20], describe *Oktopus*, a system which creates a virtual network with communicating VMs as nodes in which bandwidth between VMs is statically allocated, thus providing strict bandwidth guarantees. Shieh et al. [195] propose *Seawall*, a system in which bandwidth is shared amongst contending VMs to achieve max–min fairness via the tunneling of traffic through congestion controlled, point-to-multipoint tunnels. This work points towards a growing interest in the creation of data center network virtualization [21], wherein SDN technologies could be applied to create virtual data centers offering strong performance isolation for individual Cloud Users.

### *3.5.3 Local Scheduling of Storage Resources*

There is a substantial body of work, published over many years, on sharing and scheduling of storage resources in computing systems. However, the introduction of server virtualization technologies, wherein hypervisors mediate the access of VMs to physical storage resources, creates some important differences relative to the traditional case of an operating system controlling the access of processes to such resources. Notably, hypervisors such as *Xen* [248] implement their own disk I/O schedulers that process batches of requests from VMs, typically in a round-robin manner. Furthermore, VMs' virtual disks are stored as large files on the physical disks, leading to a larger degree of spatial locality of accesses [143]. Given these and other issues, researchers have proposed dynamic solutions for proportionally sharing disk bandwidth between VMs to achieve performance isolation; see, e.g., [88, 91, 94, 189]. More recent works have sought to go beyond the proportional sharing for performance isolation model. One example is the work of El Nably et al. [65], who propose a scheme wherein VMs that behave efficiently in terms of utilizing storage resources are rewarded by receiving a higher proportion of resources in comparison to contending, less efficient VMs. A second example is the work of Tudoran et al. [214], who address VMs hosting data-intensive applications, proposing a system that federates the local virtual disks of a set of VMs into a shared data store. Xu et al. [251] take a different approach to optimising VM I/O processing—they offload this processing to a dedicated core, which runs with a smaller time slice than cores shared by production VMs, thus significantly improving disk (and network) I/O throughput.

There is also a considerable body of work on shared storage systems in the data center context. For example Wang et al. [233] describe Cake, a two-level scheduling approach for shared storage systems, in which users can specify their performance requirements as high-level SLOs on storage system operations. The approach has the advantage that it supports generic SLOs, so that latency-sensitive and batch workloads can be readily consolidated on the same storage system. Cidon at al. [51] address shared storage systems in the data center from a data-loss minimization perspective, presenting Copyset Replication, a technique for data replication that they implemented on both HDFS and RAMcloud and demonstrated that it achieves a near optimal trade-off between the number of nodes on which data is replicated and the probability of data loss during cluster power outage events.

### 3.6 Application Scaling and Provisioning

A Cloud User that leases VMs from an IaaS Cloud Provider must monitor demand for its application(s), decide how many instances of the applications to deploy, and on which VMs, all the while observing constraints relating to which applications can/cannot be placed on given VMs. Given the bursty nature of demand for web applications this task has received considerable attention in recent years. Early work by Appleby et al. [16] allocated applications to full machines, so that applications do not share machines. Urgaonkar et al. [217] was one of the first works to propose a solution in which applications can share machines, although they consider only the dynamic modification of the number of instances deployed of a given application.

Tang et al. [211] address the complete problem—considering the allocation of multiple resource types, dynamic modification of the number of instances and of placement of individual instances, as well as placement constraints. They formulate a class-constrained multiple-knapsack problem in which the objectives are to maximize the satisfied demand for the collection of applications, to minimize the number of application instances starts and stops and to maximize the degree to which resource utilization is balanced across the machines. They propose a heuristic that has been implemented in IBM's *Websphere* product suite [108] and which, since it is $\mathcal{O}(N^{2.5})$ where $N$ is the number of physical machines, can be implemented in a centralized controller that provisions applications in clusters of 1,000s of servers. Other researchers have addressed similar problems; e.g., Gmach et al. [82] propose a fuzzy-logic based controller that dynamically migrates or replicates application instances in response to changing demand.

Some researchers have sought to overcome the scalability limitations inherent in centralized application scaling controllers. Adam and Stadler [3] describe a service middleware that implements decentralized algorithms for application placement and request routing that operate over a self-organizing management overlay. Their application placement algorithm seeks to maximize a utility function that captures the benefit of allocating compute resources to an application given the relative importance of different applications. It operates by performing a local optimization on each server, based on state information from that server's neighborhood. Evaluation results show that the solution scales both in terms of number of servers and in the number of supported applications. Wuhib et al. [245] also propose a decentralized solution for application placement, using a round-based gossip protocol; in each round, a server probabilistically selects another server to exchange state information with—information both servers then use to adjust their allocation of resources to applications.

Moens et al. [160] argue that a hierarchical model in which centralized controllers (placing applications for a number of servers) are organized in a tree-like hierarchy can combine the benefits of close-to-optimal placements by centralized controllers with the scalability of a decentralized solution. In [160] they propose a scheme where the hierarchy is structured in a manner inspired by B-Tree data structures and compare it to a modified version of the centralized algorithm proposed by Tang et al. [211]; however, they do not provide a comparison to a decentralized solution. Moens and De Turck [159] generalise the approach, proposing a generic framework for the construction and maintenance of management hierarchies for large scale management applications. The same authors have also explored the feature placement problem, where so-called application features, which are used by multiple applications, are placed on a computing infrastructure [161].

Recently, researchers have studied variants of the application provisioning and scaling problem described above. Carrera et al. [40] present an application placement algorithm (which is an extension of the algorithm presented in [211]) which addresses fairness between applications. Their formulation seeks to achieve max–min fairness by maximizing the minimum utility value amongst all applications to which resources need to be allocated to. In contrast to previous solutions their formulation is a non-linear optimization; they argue that the increased solution complexity is justified given the improvement in fairness. Ishakian and

Bestavros [111] also address fairness between applications, but argue that the primary goal should be to allocate resources to applications so that the degree to which SLAs are satisfied is maximized. Ghanbari et al. [75] and Simmons et al. [199] address how application scaling can be governed by policy rules, which they view as formal representations of a strategy. They concentrate on PaaS providers hosting applications on behalf of SaaS providers, proposing a scheme in which a strategy-tree is utilized at the SaaS layer to guide the online selection of a set of PaaS policies whose enforcement configures the application scaling process. Villegas et al. [222] study the interplay between policies used for application scaling on the one hand and workload dispatching to application instances on the other hand. They propose a taxonomy of policies of both kinds and provide results of experiments performed in three public IaaS clouds and draw conclusions about the performance–cost trade-offs between the different policies and combinations thereof.

Other researchers are focussing on engineering applications themselves in a manner that best exploits elasticity. For example, Chuang et al. [50] describe *EventWave*—"an event-driven programming model that allows developers to design elastic programs with inelastic semantics while naturally exposing isolated state and computation with programmatic parallelism." A related aspect is how best to capture users' expectations of performance and cost goals when controlling application elasticity. This issue has been explored by Jalaparti et al. [113], who describe *Bazaar*, a system that allows Cloud Users express high-level goals, which are then used to predict the resources needed to achieve them. They argue that this represents from the prevailing resource-centred Cloud User/Cloud Provider interfaces towards more job-centric interfaces that offer significant benefits to both parties.

## 3.7 Workload Management

In cloud environments workloads are the application usage sessions processed using cloud resources. Requests for these sessions must be admitted or blocked, and must be dispatched to one of (typically) a number of application instances that can process them. In this section we discuss works addressing a number of aspects: load balancing; workload control in data analytics frameworks; workload management of storage resources; and workload management in geo-distributed clouds.

### *3.7.1 Load Balancing*

Given that elasticity is one of the main benefits associated with cloud hosting, the focus has been on efficient and scalable *load balancing* (also referred to as *load sharing*), sometimes incorporating *admission control*. Load balancing has been studied in many contexts and over many years; surveys of classical approaches are provided by Wang and Morris [234] and by Kremien and Kramer [133]. In the computing context, load balancing algorithms seek to distribute workloads across a number of servers in such a manner that the average time taken to complete execution of those workloads is minimized—which typically results in server utilization being maximized and balanced. Load balancing schemes can be classified as being *static* or *dynamic*. In static schemes the current state of the servers is not considered when dispatching workloads; examples of such schemes include random selection of servers and Round Robin (where workload is sent to servers in a statically-defined, periodic sequence). On the other hand, dynamic schemes involve direct notification of or indirect inference of server state by the load balancer. A second classification distinguishes between *pre-emptive* and *non pre-emptive* load balancers: in pre-emptive schemes workloads, either queued for processing at a server or being processed, can be passed to another server for completion, whereas in non pre-emptive schemes workloads are assigned to a server upon arrival at the load balancer and cannot be subsequently re-assigned. Because of the dynamic nature of demands for the majority of cloud-hosted applications and the inefficiencies associated with transferring workloads between servers most researchers have focussed on the development and evaluation of dynamic, non pre-emptive load balancing schemes.

One of the well-known load balancing schemes is termed the *power-of-d* algorithm, which is based on the *Supermarket Model* studied by Mitzenmacher [158] and which in turn builds upon theoretical work by Azar et al. [18]. In this scheme *d* servers are sampled, and the workload is dispatched to the server with the lowest utilization. This approach is shown to provide exponentially improved workload execution times (in the number of servers) in comparison to random server selection. One issue with this and similar schemes is the overhead associated with transferring server state to the load balancer. Breitgand et al. [36] address this point and propose an *Extended Supermarket Model*, for which they show that there is an optimal number of servers that should be monitored to obtain minimal average service time at a given cost. Based on this result, they propose self-adaptive load balancing schemes that they show, via simulation and a test-bed implementation, are superior to schemes that are oblivious to the monitoring cost. Lu et al. [149] address the state transfer issue by decoupling the discovery of the more lightly loaded servers from the process of workload assignment: servers indicate to load balancers when their processors are idle. However, this approach raises the question which subset of the load balancers a server should notify. To address this issue their *Join-Idle-Queue* algorithm load balances idle processors across the load balancers. Their results show that this approach out-performs the *power-of-d* algorithm in terms of minimizing workload completion time, and in doing so incurs no communication overhead on the critical path of processing a workload request.

Whilst the works discussed above address the general problem of load balancing in large-scale server clusters, other researchers have focussed on aspects associated with IaaS and PaaS cloud environments. One common concern is how to take account of SLAs when dispatching workloads. For example: Gmach et al. [82] propose an SLA-aware workload scheduling scheme as part of a three-tier resource management system; whilst Zhang et al. [264] describe a scheme in which agents on servers predict available resources for a control interval, which is notified to load balancers who then optimize dispatching of workloads to achieve service differentiation. Liu et al. [144] focus on workload management in the context of data center energy use and the desire to maximize the use of energy from renewable sources. Their framework includes a workload-scheduling scheme that makes use of the fact that many batch workloads are delay tolerant, so they can be delayed in a manner that minimizes overall energy use and maximizes renewable energy use. Tumanov et al. [215] explore expressing workload requirements as either hard or soft constraints, where soft constraints relate to characteristics that provide benefits but are not mandatory; they describe how expressing these constraints as relatively simple utility functions enables the creation of a scheduler that maximises utility of workload placement.

### *3.7.2 Data Analytics Frameworks*

An area of workload management that has generated considerable interest in recent years is the allocation of resources to data-intensive applications such as those based on the MapReduce model [60]. Data analytics frameworks like *Hadoop* [236], which implement MapReduce, are now widely used in cloud-hosted applications, so how to optimize their performance is of significant interest. Zaharia et al. [260] showed that *Hadoop*'s default task scheduler implementation performed badly when deployed on a public IaaS cloud due to its implicit assumption that compute resources available to it are homogeneous and that tasks make progress linearly. They propose a new task scheduling algorithm called "Longest Approximate Time to End (LATE)," which they show is robust to resource heterogeneity and thus leads to significantly improved *Hadoop* performance. Other researchers have also addressed the optimization of MapReduce implementations. For example Sandholm and Lai [184] address MapReduce platforms supporting execution of multiple applications; they propose the use of user-assigned priorities to dynamically adjust the allocation of resources to contenting applications. Tan et al. [210] argue that, since map and reduce phases exhibit different characteristics and have a complex and tight interdependency, a scheduler should couple their progress rather than treat them separately. They use an analytic model and simulations to show that their Coupling Scheduler can outperform the standard Fair Scheduler by up to an order of magnitude. These papers represent a flavor of the significant ongoing work in this area; an overview of this work, pointing out unrealistic assumptions and simplifications made by many researchers, is provided by Schwarzkopf et al. [188]. Recently, a more radical approach to solving this issues has been proposed by Vavilapalli et al. [220], who describe *YARN*, an analytics platform in which resource management functions are separated from the programming model: many scheduling-related functions are delegated to per-job components. This separation provides flexibility in the choice of programming framework, so that the platform can support not only MapReduce, but also other models such as *Dryad* [110] and *Storm* [206].

Besides improving the performance of the schedulers that are central to data analytics frameworks researchers have also addressed how to provision these frameworks with resources during the execution of jobs. Lee et al. [138] address the issue that the heterogeneity of resources within a typical data center cluster environment means that certain jobs will execute more efficiently on certain machines. They propose a new metric they term "resource share," which captures the contribution of each resource to the progress of a job, showing how it can be used to determine the scheduling of resources in an architecture where a number of core machines are augmented by “accelerator” machines when additional computation is required on a shorter-term basis. Hindman et al. [104] recognise that different data analytics frameworks have emerged that are optimized for different applications and that such frameworks are often simultaneously run in the same cluster. They propose *Mesos*, a thin resource sharing layer that facilitates resource sharing across data analytic frameworks, through providing a standard interface to the frameworks to offer them resources. The frameworks themselves decide which resources to accept and how to use them for their computations. In [76] the same authors address the problem of fair resource allocation where resources are heterogeneous, proposing a new resource allocation policy termed “Dominant Resource Fairness,” which generalizes (weighted) max–min fairness to the multiple resource scenario. More recently this approach has been further enhanced to address hierarchical scheduling of multiple resources, in which nodes in a hierarchy receive at least their fair share of resources, regardless of the behavior of other nodes [28].

### *3.7.3 Workload Management of Storage Resources*

Another issue of growing concern in cloud environments is workload management of storage resources. Gulati et al. [92] describe *Pesto*, a framework for storage workload management that has been implemented as part of VMWare’s *vSphere* [224] offering. *Pesto* incorporates a load balancing scheme whereby storage workloads are placed on virtual disks based on latency measurements gathered from the set of available disks (this is an extension of a previous scheme by the same authors [90]). Fan et al. [67] focus on load balancing requests for distributed storage systems like distributed file systems, distributed object caches, and key-value storage systems, proposing an architecture in which a small and fast cache placed at a load balancer can ensure effective load balancing, regardless of the query distribution. Park et al. [170] describe *Romano*, which achieves efficient I/O load balancing by constructing and continuously adapting workload-specific models of storage devices automatically. They take into account the heterogeneous response of storage systems and the interference between workloads, formulating a load balancing algorithm based on simulated annealing, which reduces performance variance by 82 % and maximum latency by 78 % in comparison to previous techniques.

### *3.7.4 Workload Management in Geo-Distributed Clouds*

Recently, researchers started to consider workload management in the context of geo-distributed clouds, i.e. cloud environments realized via resources housed in data centers that are geographically dispersed, but that are managed by a single organization. In particular works including [4, 109, 249, 254, 258, 266] have proposed algorithms designed to reduce overall costs, e.g., by using data centers where electricity prices are currently lowest or by scheduling delay-tolerant workloads for times when the best electricity prices are offered. Wu at al. [243]

specifically address the deployment of large-scale social media streaming applications in geo-distributed clouds, proposing algorithms to control the replication of content across data centers and a request distribution algorithm cognizant of content locations. Qian and Rabinovich [177] address the combined problem of provisioning application instances across a geo-distributed cloud and distributing, via routing policies, End User requests to these instances; they propose a novel demand clustering approach which scales to realistic system sizes.

### 3.8 Cloud Management Systems

As outlined in Sect. 2.5.8, cloud management systems provide the command and control interfaces that allow Cloud Providers and Cloud Users to both codify their management objectives as SLAs and management policies, and to analyze management metrics that provide feedback on how these objectives are being met. Section 3.1.3 reviewed some proposals that have emerged to allow Cloud Users request increasingly complex VI instantiations from Cloud Providers [20, 25, 47, 132]. Regardless of the complexity of the VI they lease, Cloud Users will agree to SLAs specifying, for example, minimal average allocations of given resources within a given time period. Chhetri et al. [49] address the process of establishing such SLAs; they observe that many interaction models (e.g., auctions or one-to-one negotiations) are possible, so they propose a policy-based framework that automates selection of the interaction model based on the nature of each incoming VI deployment request. Ishakian and Bestavros [111] point out that cloud SLAs are typically quite coarse-grained—in that they often specify minimal levels of resources allocated over a relatively long time period. Thus, allocating resources based on these SLAs can be sub-optimal. To mitigate this, they propose *MorphoSys*, which gives Cloud Providers the ability to safely transform Cloud User SLAs into finer-grained SLAs that can be used in the resource allocation process.

Other works focus on the policies Cloud Providers apply to ensure that Cloud User SLAs are satisfied and their own management objectives are met. Liu et al. [147] present the *COPE* platform, which performs automated cloud resource optimization based on declarative policies that capture system-wide constraints and goals. Cloud User SLAs are aggregated and codified as the goals of constraint optimization problems, which are solved by a central controller for a data center (although a distributed scheme for federated clouds is also outlined). Macías and Guitart [150] address situations where not all SLAs can be satisfied, outlining how policies can be applied to achieve a service differentiation in which preferential customers are allocated resources at the expense of other customers. Further examples of works addressing the use of policies to enforce management objectives include Sulistio et al. [208], Weng and Bauer [238], and Chen et al. [46].

Once SLAs have been agreed and management objectives set there is a need to monitor the system to assess its success in achieving its goals. As discussed in Sect. 3.3 software tools like *collectd* [55] and *Ganglia* [74] provide the ability to monitor a large range of system metrics. Most of these do so via a centralized architecture, with *Ganglia* being an exception as it offers the ability to create a (static) distributed monitoring overlay. However, it is accepted that the large scale of cloud environments and the dynamicity of the resource utilization patterns within them necessitate decentralized monitoring solutions that can be configured to balance the trade-off between accuracy and timeliness of monitored metrics. Some examples of distributed cloud monitoring solutions are described in [154, 156]. Stadler et al. [205] identify design principles for such monitoring solutions and describe and compare two approaches for distributed monitoring of aggregates—one based on self-organizing spanning trees and one based on gossip protocols.

### 3.9 Cloud Measurement Studies

Although not part of our conceptual framework for cloud resource management, measurement studies from production cloud environments provide valuable insights into the challenges that need to be addressed when designing and implementing resource management solutions. Google has released measurement traces from a reasonably large cluster environment [240]. Results of a number of analyses of this trace have been published: Liu and Cho [145] focus on machine maintenance events, machine characteristics and workload characteristics, whilst Reiss et al. [181] highlight the heterogeneity of both the resource types in the cluster and the manner in which they are used, pointing out that such heterogeneity reduces the effectiveness of traditional resource scheduling techniques. They also note that the workload is highly dynamic and is comprised of many short jobs and a smaller number of long running jobs with stable resource utilizations. Zhang et al. [265] focus less on cluster characteristics inferred from the trace, instead using it to evaluate an energy use minimization scheme.

Beyond analyses of the Google traces, other researchers have studied aspects of other Cloud Provider systems insofar as is possible without operational traces. Chen et al. [45] provide a study of inter data center traffic based on anonymized NetFlow datasets collected at the border routers of five Yahoo! data centers, inferring characteristics of Yahoo!'s data center deployment schemes and the level of correlation between different Yahoo! services. Schad et al. [186] describe a study of the performance variance of the *Amazon EC2* public IaaS offering, using established micro-benchmarks to measure variance in CPU, I/O and network metrics; their results indicate a significant amount of performance variance, not only over time butalso between different availability zones and locations. Potharaju and Jain [175] focus on the impact of network failures on cloud-hosted services: they analyse 3 years of network event logs collected in a Cloud Provider network across thousands of devices spanning dozens of data centers, finding that network failures have significant adverse impact on cloud-hosted services.

## 4 Research Challenges

Based on our analysis, we identify five challenges we believe specifically merit future investigation. The first three relate to known general problems that are inherently hard to solve in theory and that require additional efforts to produce practical solutions that are effective in the cloud context. The last two refer to significant resource management challenges that arise from the evolution of the cloud computing paradigm.

### 4.1 Achieving Predictable Performance for Cloud-Hosted Applications

Understanding the key performance metrics of an application and how these metrics evolve over time, depending on load and allocated resources, is essential for meeting SLAs and for making effective decisions about when to allocate additional resources or release unused resources. When addressing these issues, one must recognize that modeling and understanding the performance of any complex software system (such as a modern cloud platform or a cloud application) is intrinsically hard. One reason is that computer operating systems that drive cloud platforms today do not provide real-time guarantees. (Local resource allocation is done through setting priority levels of schedulers, for instance.) Further, and more importantly, there is no foundational theory to guide us in building practical tools to predict and control the performance of programs. This is the case for computing systems in general, but becomes even more prominent for cloud environments, due to the additional layer of virtualization on top of which cloud applications execute.

From a Cloud Provider's perspective, efficient use of resources implies statistical multiplexing of resources across customers and applications. As a consequence, the resources that a virtualized infrastructure of an IaaS service provides at any given time are less predictable than those of a traditional, dedicated physical infrastructure. With interactive applications, for example, Cloud Users can experience varying response times for a steady load, which today prevents many businesses from using IaaS services.

While we believe that a general solution to the problem of predictable performance is not feasible in the short-to-medium term, there are particular areas that can be successfully addressed today, with significant potential impact on technology evolution. First, for a virtualized infrastructure, the tradeoff between predictability of available resources and the efficiency of using the underlying hardware infrastructure needs to be thoroughly investigated (Breitgand and Epstein [34] is an important step in this direction.) A potential outcome of this work could be that Cloud Providers can make profitable use of their resources by offering a range of options to Cloud Users, spanning from deterministic guarantees of resource availability all the way to best-effort services with no, or weak, statistical guarantees. Further, a general mechanism could be developed through which a Cloud Provider can pass information about current and expected resource availability to a Cloud User, which then uses the information to optimize its application, as has been demonstrated for MapReduce implementations [19, 103, 184, 235].

Second, the general problem of performance prediction can be more easily addressed for specific use cases and constraints. For instance, performance models can be devised for specific architectures, such as multi-tiered web services (e.g., [139]). Also, it is often important to understand the performance of systems under moderate load, where performance bottlenecks can be more easily identified than under high load, and performance analysis can be concentrated on understanding and controlling those few bottlenecks [256].

Third, the performance of an application can be controlled using a feedback loop, applying techniques from control theory [101]. This approach has been successfully demonstrated, e.g., for dynamically resizing elastic services in response to workload changes, in order to maintain a target performance metric, as detailed in [212, 213].

Fourth, the effectiveness of performance prediction critically depends on the accuracy of forecasting demand, also known as *demand profiling* (see Sect. 2.5.2). Approaches to demand profiling can be broadly classified as either model based, wherein a model of resources, application and user behavior is developed and parameterized through measurements, or model independent, wherein data analytics approaches, including time series forecasting, are applied. As cloud-hosted services proliferate and exhibit greater diversity, effective tools will be needed that classify and characterize demand at different operational scopes and timescales and represent this knowledge in a manner useful to resource controllers.

### 4.2 Achieving Global Manageability of Cloud Environments

Global Manageability refers to the ability to control a system in order to achieve a given set of global, system-wide objectives. Today's technology provides global manageability in a very limited way, and we believe there is a clear need for a principled approach in this direction. In the following, we map out the design space and the main challenges.

First, as pointed out in Sect. 2, resource management in clouds involves resources of different types, including compute, communication, storage, and energy resources. Even focusing on a single type like compute resources—as many researchers have done until now—involves understanding the interaction of different types of controllers (admission controllers, global schedulers, local schedulers, etc.), which has not yet been systematically investigated. Furthermore, resource management of different resources must be performed in a coordinated manner, since they are often dependent. For instance, there is a spatial dependency between compute and communications or between compute and storage resources. Furthermore, there are tradeoffs between the usage of different resource types, for example, between networking and energy resources or between compute and storage resources. Such trade-offs must be analyzed, so that they can be controlled and exploited. As this survey demonstrates, most work on resource management in clouds has focused on compute resources (with networking and and storage being addressed in isolation by a specialized communities). A comprehensive overall approach for joint allocation of compute, networking, storage, and energy resources is missing, and such an approach would form the basis for effective global management.

In multi-tenant clouds, Cloud Provider and Cloud Users have only partial knowledge of the system. All actors take control of decisions independently from one another, following their own objectives, which can lead to unintended consequences and should be investigated. The effect can possibly be mitigated, for

PaaS systems in particular, through sharing system information across provider/tenant APIs.

A specific challenge in global management relates to management across data centers, to achieve a desired level of service availability and robustness, or to meet a specific performance target, e.g., a bound on response times, in support of a global customer base. A related challenge refers to global management of a federated cloud infrastructure, either in the form of managing a hybrid cloud or cooperative management of a shared infrastructure. Federated solutions will increase in importance, for instance, to facilitate big data applications where data, for practical or legal reasons, cannot be moved between clouds.

### 4.3 Engineering Scalable Resource Management Systems

In order to exploit economies of scale, large data centers often include hundreds of thousands of servers. To efficiently use such an infrastructure, resource management systems must scale along several dimensions, in particular, with the number of physical components, such as servers and switches, the number of VI instances, and the number of supported applications and their End Users.

Today, to the best of our knowledge, resources in data centers are statically partitioned into pre-defined clusters of physical machines and associated equipment, which typically include several thousand servers. Within each such cluster, resources are dynamically allocated using centralized controllers. The primary reason for the limited cluster sizes is the fact that centralized resource controllers do not feasibly scale beyond those limits, as the computational overhead and response times become too large. As this survey shows, research is underway towards decentralizing resource allocation, which would allow, in principle, to treat a data center as a single, flat resource pool. Such a solution, however, is not desirable for robustness reasons, as a single failure may impact the operations of an entire center. We believe that a dynamic partitioning scheme may be most beneficial, where clusters are dynamically formed, based on current demand and application mix, and where resources within clusters are managed independently. Dynamic partitioning would enable controlling the tradeoff between robust operation and efficient resource allocation and would probably involve a combination of centralized and distributed control schemes.

### 4.4 Economic Behavior and Pricing Strategies

Dynamic pricing of cloud resources, wherein usage prices are dynamically set by pricing agents, is an emerging trend in IaaS and PaaS cloud offerings. Cloud Users, and potentially End Users, can exploit price fluctuations for a base load of batch tasks that can be delayed in anticipation of lower prices in the future. Cloud Users can also exploit the performance heterogeneity of VI leased from IaaS providers in order to lower their costs—a process termed "placement gaming" by Farley et al. [68]. Moreover, as described in Sect. 2.5.4, Cloud Providers, and potentially Cloud Users, can apply dynamic pricing as an instrument to influence demand.

Pricing of resource usage is considered a hard problem, as the body of prior work from the economics field suggests. (For an introduction see [173] and [64].) In the cloud context, like in other domains, modeling and formulating pricing of resources starts with understanding and modeling consumer behavior (i.e., Cloud Users or End Users) in terms of demand patterns and price sensitivity. As can be seen from Sect. 3.4, research interest in dynamic pricing of cloud resources is growing, although the focus thus far has been on minimizing costs for Cloud Users, e.g., through scheduling their workloads over time or across Cloud Providers. Interestingly, we could not find work into strategies from the perspective of the Cloud Provider, which we feel merits exploration.

Many Cloud Users/End Users will not have the expertise to properly exploit dynamic price fluctuations, which may open the field for *cloud brokers* [229], who accept risk associated with reserving dynamically priced resources in return for charging higher but stable prices. Moreover, Cloud Users may harness nested virtualization capabilities of operating systems [27] to directly re-sell computing capacity they lease from a Cloud Provider [203]. Modeling behavior and devising pricing strategies for such a cloud ecosystem are topics that should be investigated.

### 4.5 Challenges in Mobile Cloud Computing

Advances in web technologies have facilitated the migration of traditional desktop applications to the cloud. In parallel, the proliferation of handheld devices leads us to expect that future end users will, for the most part, use handheld, mobile devices to access cloud-hosted applications. This trend gives rise to the *mobile cloud computing paradigm*, which, at its core, extends the data center to the very edge of the network.

Compared to desktop computers, mobile devices are energy and bandwidth constrained. Furthermore, delays incurred in transferring data to/from cloud data centers present limitations, primarily for realtime interactive applications. The basic challenge of mobile cloud computing is how to overcome or mitigate these constraints through novel engineering. It seems clear that architectural changes to the current network and computing infrastructure will be needed: (some) data and computation must be available close to or at the network edges. Techniques need to be developed to distribute computation and data across the mobile devices and the fixed infrastructure, in order to meet delay, energy and bandwidth–related objectives. Also, software frameworks need to be developed that support those techniques.

Research in mobile cloud computing is still at an early stage. While some interesting architectural proposals have been put forward (e.g., [32, 53, 122, 185]), we believe that the major contributions in this area are still to be made.

## 5 Conclusions

In this paper we surveyed research into resource management for cloud environments. While working on this survey, we were surprised by the amount of recent results that we found, and the paper grew therefore larger than anticipated. To clarify the discussion and better place individual contributions in context, we outlined a framework for cloud resource management, which lays the basis for the core of the paper, the state-of-the-art survey. Based on the available literature, we decided to classify the field into eight subdomains related to global scheduling, local scheduling, demand profiling, utilization estimation, pricing, application scaling, workload management, cloud management, and measurement studies. We concluded the paper with a set of fundamental research challenges, which we hope will encourage new activities in this fascinating and timely field.

**Acknowledgments** We are grateful to colleagues who provided helpful comments and suggestions regarding this work, specifically: David Breitgand (IBM Research), Shane Dempsey (Arc Mediation), Lei Shi (WIT) and Fetahi Wuhib (Ericsson Research). This work was partially funded by Science Foundation Ireland via Grants 08/SRC/I1403 and 08/SRC/I1403-STTF11 and by the European Commission via FP7 Grant No. 612480.

## References

1. Abts, D., Felderman, B.: A guided tour of data-center networking. Commun. ACM **55**(6), 44–51 (2012). doi:10.1145/2184319.2184335
2. ACPI—advanced configuration and power interface. http://www.acpi.info/ (2012)
3. Adam, C., Stadler, R.: Service middleware for self-managing large-scale systems. IEEE Trans. Netw. Serv. Manag. **4**(3), 50–64 (2007). doi:10.1109/TNSM.2007.021103
4. Adnan, M.A., Sugihara, R., Gupta, R.: Energy efficient geographical load balancing via dynamic deferral of workload. In: Proceedings of 5th IEEE International Conference on Cloud Computing (CLOUD 2012), pp. 188–195. IEEE (2012)
5. Agmon Ben-Yehuda, O., Ben-Yehuda, M., Schuster, A., Tsafrir, D.: Deconstructing amazon EC2 spot instance pricing. In: Proceedings of 3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2011), pp. 304–311. IEEE (2011). doi:10.1109/CloudCom.48
6. Ahn, J., Kim, C., Choi, Y.R., Huh, J.: Dynamic virtual machine scheduling in clouds for architectural shared resources. In: Proceedings of 4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2012) (2012)
7. Akoush, S., Sohan, R., Rice, A., Moore, A., Hopper, A.: Predicting the performance of virtual machine migration. In: Proceedings of 2010 IEEE International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS 2010), pp. 37–46. IEEE (2010). doi:10.1109/MASCOTS.2010.13
8. Al-Fares, M., Loukissas, A., Vahdat, A.: A scalable, commodity data center network architecture. In: Proceedings of the ACM SIGCOMM 2008 Conference on Data communication (SIGCOMM 2008), pp. 63–74. ACM (2008). doi:10.1145/1402958.1402967
9. Al-Haj, S., Al-Shaer, E.: A formal approach for virtual machine migration planning. In: Proceedings of 9th International Conference on Network and Service Management (CNSM 2013), pp. 51–58. IFIP (2013)
10. Alicherry, M., Lakshman, T.: Network aware resource allocation in distributed clouds. In: Proceedings of 2012 IEEE International Conference on Computer Communications (Infocom 2012), pp. 963–971. IEEE (2012). doi:10.1109/INFCOM.2012.6195847
11. Ali-Eldin, A., Tordsson, J., Elmroth, E.: An adaptive hybrid elasticity controller for cloud infrastructures. In: Proceedings of 13th IEEE Network Operations and Management Symposium (NOMS 2012), pp. 204–212. IEEE (2012). doi:10.1109/NOMS.2012.6211900
12. Amazon EC2 FAQs. http://aws.amazon.com/ec2/faqs/ (2013)
13. Amazon EC2 pricing. http://aws.amazon.com/ec2/pricing/ (2012)
14. Amazon EC2 spot instances. http://aws.amazon.com/ec2/spot-instances (2012)
15. Amazon Inc.: Amazon web services. http://aws.amazon.com/ (2012)
16. Appleby, K., Fakhouri, S., Fong, L., Goldszmidt, G., Kalantar, M., Krishnakumar, S., Pazel, D., Pershing, J., Rochwerger, B.: Oceano-SLA based management of a computing utility. In:

Proceedings of 7th IEEE/IFIP International Symposium on Integrated Network Management (IM 2001), pp. 855–868. IEEE (2001). doi:10.1109/INM.2001.918085

17. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010). doi:10.1145/1721654.1721672
18. Azar, Y., Broder, A.Z., Karlin, A.R., Upfal, E.: Balanced allocations. SIAM J. Comput. **29**(1), 180–200 (1999). doi:10.1137/S0097539795288490
19. Babu, S.: Towards automatic optimization of MapReduce programs. In: Proceedings of 1st ACM Symposium on Cloud Computing (SoCC 2010), pp. 137–142. ACM (2010). doi:10.1145/1807128.1807150
20. Ballani, H., Costa, P., Karagiannis, T., Rowstron, A.: Towards predictable datacenter networks. In: Proceedings of ACM SIGCOMM 2011 Conference on Data Communication (SIGCOMM 2011), pp. 242–253. ACM (2011). doi:10.1145/2018436.2018465
21. Bari, M.F., Boutaba, R., Esteves, R., Podlesny, M., Rabbani, M.G., Zhang, Q., Zhani, M.F.: Data center network virtualization: a survey. IEEE Commun. Surv. **15**(2), 909–928 (2013). doi:10.1109/SURV.2012.090512.00043
22. Bazarbayev, S., Hiltunen, M., Joshi, K., Sanders, W.H., Schlichting, R.: Content-based scheduling of virtual machines (VMs) in the cloud. In: Proceedings of 33rd IEEE International Conference on Distributed Computing Systems (ICDCS 2013), pp. 93–101. IEEE (2013)
23. Beloglazov, A., Buyya, R.: Energy efficient resource management in virtualized cloud data centers. In: Proceedings of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid 2010), pp. 826–831. IEEE (2010). doi:10.1109/CCGRID.2010.46
24. Bennani, M., Menasce, D.: Resource allocation for autonomic data centers using analytic performance models. In: Proceedings of 2nd International Conference on Autonomic Computing (ICAC 2005), pp. 229–240. IEEE (2005). doi:10.1109/ICAC.2005.50
25. Benson, T., Akella, A., Shaikh, A., Sahu, S.: CloudNaaS: a cloud networking platform for enterprise applications. In: Proceedings of 2nd ACM Symposium on Cloud Computing (SoCC 2011), pp. 8:1–8:13. ACM (2011). doi:10.1145/2038916.2038924
26. Ben-Yehuda, O.A., Ben-Yehuda, M., Schuster, A., Tsafrir, D.: The resource-as-a-service (RaaS) cloud. In: Proceedings of 4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2012). USENIX (2012)
27. Ben-Yehuda, M., Day, M.D., Dubitzky, Z., Factor, M., Har'El, N., Gordon, A., Liguori, A., Wasserman, O., Yassour, B.A.: The turtles project: design and implementation of nested virtualization. In: Proceedings of 9th USENIX Conference on Operating Systems Design and Implementation (OSDI 2009), pp. 1–6. USENIX (2010)
28. Bhattacharya, A.A., Culler, D., Friedman, E., Ghodsi, A., Shenker, S., Stoica, I.: Hierarchical scheduling for diverse datacenter workloads. In: Proceedings of 4th ACM Symposium on Cloud Computing (SoCC 2013). ACM (2013)
29. Bin, E., Biran, O., Boni, O., Hadad, E., Kolodner, E., Moatti, Y., Lorenz, D.: Guaranteeing high availability goals for virtual machine placement. In: Proceedings of 31st IEEE International Conference on Distributed Computing Systems (ICDCS 2011), pp. 700–709. IEEE (2011). doi:10.1109/ICDCS.2011.72
30. Biran, O., Corradi, A., Fanelli, M., Foschini, L., Nus, A., Raz, D., Silvera, E.: A stable network-aware VM placement for cloud systems. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 498–506. IEEE (2012). doi:10.1109/CCGrid.119
31. Bobroff, N., Kochut, A., Beaty, K.: Dynamic placement of virtual machines for managing SLA violations. In: Proceedings of 10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007), pp. 119–128. IEEE (2007). doi:10.1109/INM.2007.374776
32. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of 1st Workshop on Mobile Cloud Computing (MCC 2012), pp. 13–16. ACM (2012). doi:10.1145/2342509.2342513
33. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis: Forecasting and Control, 4th edn. Wiley, London (2008)
34. Breitgand, D., Epstein, A.: Improving consolidation of virtual machines with risk-aware bandwidth oversubscription in compute clouds. In: Proceedings of 2012 IEEE International Conference on Computer Communications (Infocom 2012), pp. 2861–2865 (2012). doi:10.1109/INFCOM.2012.6195716

35. Breitgand, D., Epstein, A.: SLA-aware placement of multi-virtual machine elastic services in compute clouds. In: Proceedings of 12th IFIP/IEEE Symposium on Integrated Network Management (IM 2011), pp. 161–168. IEEE (2011). doi:10.1109/INM.2011.5990687
36. Breitgand, D., Cohen, R., Nahir, A., Raz, D.: On cost-aware monitoring for self-adaptive load sharing. IEEE J. Sel. Areas Commun. **28**(1), 70–83 (2010). doi:10.1109/JSAC.2010.100108
37. Briscoe, G., Marinos, A.: Digital ecosystems in the clouds: Towards community cloud computing. In: Proceedings of 3rd IEEE International Conference on Digital Ecosystems and Technologies (DEST 2009), pp. 103–108. IEEE (2009). doi:10.1109/DEST.2009.5276725
38. Buneci, E.S., Reed, D.A.: Analysis of application heartbeats: learning structural and temporal features in time series data for identification of performance problems. In: Proceedings of 2008 ACM/IEEE Conference on Supercomputing (SC 2008), pp. 52:1–52:12. IEEE (2008)
39. Cardosa, M., Korupolu, M., Singh, A.: Shares and utilities based power consolidation in virtualized server environments. In: Proceedings of 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009), pp. 327–334 (2009). doi:10.1109/INM.2009.5188832
40. Carrera, D., Steinder, M., Whalley, I., Torres, J., Ayguade, E.: Utility-based placement of dynamic web applications with fairness goals. In: Proceedings of 11th IEEE/IFIP Network Operations and Management Symposium (NOMS 2008), pp. 9–16. IEEE (2008). doi:10.1109/NOMS.2008.4575111
41. Chaisiri, S., Lee, B.S., Niyato, D.: Optimization of resource provisioning cost in cloud computing. IEEE Trans. Serv. Comput. **5**(2), 164–177 (2012). doi:10.1109/TSC.2011.7
42. Chandra, A., Gong, W., Shenoy, P.: Dynamic resource allocation for shared data centers using online measurements. In: Jeffay, K., Stoica, I., Wehrle, K. (eds.) Proceedings of 2003 International Workshop on Quality of Service (IWQoS 2003), LNCS, vol. 2707, pp. 381–398 (2003)
43. Chase, J.S., Anderson, D.C., Thakar, P.N., Vahdat, A.M., Doyle, R.P.: Managing energy and server resources in hosting centers. SIGOPS Oper. Syst. Rev. **35**(5), 103–116 (2001). doi:10.1145/502059.502045
44. Chen, Y., Griffith, R., Liu, J., Katz, R.H., Joseph, A.D.: Understanding TCP incast throughput collapse in datacenter networks. In: Proceedings of the 1st ACM workshop on Research on enterprise networking (WREN 2009), pp. 73–82. ACM (2009). doi:10.1145/1592681.1592693
45. Chen, Y., Jain, S., Adhikari, V., Zhang, Z.L., Xu, K.: A first look at inter-data center traffic characteristics via yahoo! datasets. In: Proceedings of 2011 IEEE Conference on Computer Communications Workshops (Infocom 2011), pp. 1620–1628. IEEE (2011). doi:10.1109/INFCOM.2011.5934955
46. Chen, X., Mao, Y., Mao, Z.M., Van der Merwe, J.: Declarative configuration management for complex and dynamic networks. In: Proceedings of 6th International Conference on Emerging Network Experiments and Technologies (Co-NEXT 2010), pp. 6:1–6:12. ACM (2010). doi:10.1145/1921168.1921176
47. Chen, C.C., Yuan, L., Greenberg, A., Chuah, C.N., Mohapatra, P.: Routing-as-a-service (RaaS): a framework for tenant-directed route control in data center. In: 2011 IEEE International Conference on Computer Communications (Infocom 2011), pp. 1386–1394. IEEE (2011). doi:10.1109/INFCOM.2011.5934924
48. Chen, Y., Das, A., Qin, W., Sivasubramaniam, A., Wang, Q., Gautam, N.: Managing server energy and operational costs in hosting centers. SIGMETRICS Perform. Eval. Rev. **33**(1), 303–314 (2005). doi:10.1145/1071690.1064253
49. Chhetri, M., Vo, Q.B., Kowalczyk, R.: Policy-based automation of SLA establishment for cloud computing services. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 164–171. IEEE (2012). doi:10.1109/CCGrid.116
50. Chuang, W.C., Sang, B., Yoo, S., Gu, R., Killian, C., Kulkarni, M.: EventWave: programming model and runtime support for tightly-coupled elastic cloud applications. In: Proceedings of 4th ACM Symposium on Cloud Computing (SoCC 2013). ACM (2013)
51. Cidon, A., Rumble, S., Stutsman, R., Katti, S., Ousterhout, J., Rosenblum, M.: Copysets: reducing the frequency of data loss in cloud storage. In: Proceedings of 2013 USENIX Annual Technical Conference (ATC 2013). USENIX (2013)
52. Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., Warfield, A.: Live migration of virtual machines. In: Proceedings of 2nd Conference on Symposium on Networked Systems Design and Implementation (NSDI 2005), pp. 273–286. USENIX (2005)
53. Clinch, S., Harkes, J., Friday, A., Davies, N., Satyanarayanan, M.: How close is close enough? understanding the role of cloudlets in supporting display appropriation by mobile users. In:

Proceedings of 2012 IEEE International Conference on Pervasive Computing and Communications (PerCom 2012), pp. 122–127. IEEE (2012). doi:10.1109/PerCom.6199858

54. Coffman Jr, E.G., Garey, M.R., Johnson, D.S.: Approximation Algorithms for NP-Hard Problems, pp. 46–93. PWS Publishing Co., Boston (1997)
55. collectd—the system statistics collection daemon. http://collectd.org/ (2012)
56. Crovella, M., Bestavros, A.: Self-similarity in world wide web traffic: evidence and possible causes. IEEE/ACM Trans. Netw. **5**(6), 835–846 (1997). doi:10.1109/90.650143
57. Curtis, A., Kim, W., Yalagandula, P.: Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection. In: Proceedings of 2011 IEEE Conference on Computer Communications Workshops (Infocom 2011), pp. 1629–1637. IEEE (2011). doi:10.1109/INFCOM.2011.5934956
58. Das, R., Yarlanki, S., Hamann, H., Kephart, J.O., Lopez, V.: A unified approach to coordinated energy-management in data centers. In: Proceedings of 7th International Conference on Network and Services Management (CNSM 2011), pp. 504–508. IFIP (2011)
59. De, P., Roy, S.: VMSpreader: multi-tier application resiliency through virtual machine striping. In: Proceedings of 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011), pp. 185–192. IEEE (2011). doi:10.1109/INM.2011.5990690
60. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008). doi:10.1145/1327452.1327492
61. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., Vogels, W.: Dynamo: amazon's highly available key-value store. SIGOPS Oper. Syst. Rev. **41**(6), 205–220 (2007). doi:10.1145/1323293.1294281
62. Dejun, J., Pierre, G., Chi, C.H.: Resource provisioning of web applications in heterogeneous clouds. In: Proceedings of 2nd USENIX Conference on Web Application Development (WebApps 2011), pp. 5–15. USENIX (2011)
63. Edmonds, A., Metsch, T., Papaspyrou, A., Richardson, A.: Toward an open cloud standard. IEEE Internet Comput. **16**(4), 15–25 (2012)
64. Elmaghraby, W., Keskinocak, P.: Dynamic pricing in the presence of inventory considerations: research overview, current practices, and future directions. Manag. Sci. **49**(10), 1287–1309 (2003). doi:10.1287/mnsc.49.10.1287.17315
65. Elnably, A., Du, K., Varman, P.: Reward scheduling for QoS in cloud applications. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 98–106. IEEE (2012). doi:10.1109/CCGrid.120
66. Esteves, R.P., Zambenedetti Granville, L., Bannazadeh, H., Boutaba, R.: Paradigm-based adaptive provisioning in virtualized data centers. In: Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 169–176. IEEE (2013)
67. Fan, B., Lim, H., Andersen, D.G., Kaminsky, M.: Small cache, big effect: provable load balancing for randomly partitioned cluster services. In: Proceedings of 2nd ACM Symposium on Cloud Computing (SoCC 2011), pp. 23:1–23:12. ACM (2011). doi:10.1145/2038916.2038939
68. Farley, B., Juels, A., Varadarajan, V., Ristenpart, T., Bowers, K.D., Swift, M.M.: More for your money: exploiting performance heterogeneity in public clouds. In: Proceedings of the Third ACM Symposium on Cloud Computing (SoCC 2012), pp. 20:1–20:14. ACM, New York, NY (2012). doi:10.1145/2391229.2391249. http://doi.acm.org/10.1145/2391229.2391249
69. Feller, E., Rilling, L., Morin, C.: Snooze: A scalable and autonomic virtual machine management framework for private clouds. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 482–489. IEEE (2012). doi:10.1109/CCGrid.71
70. Foster, G., Keller, G., Tighe, M., Lutfiyya, H., Bauer, M.: The right tool for the job: switching data centre management strategies at runtime. In: Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 151–159. IEEE (2013)
71. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Am. Math. Mon. **69**(1), 9–15 (1962). doi:10.2307/2312726
72. Gamage, S., Kangarlou, A., Kompella, R.R., Xu, D.: Opportunistic flooding to improve TCP transmit performance in virtualized clouds. In: Proceedings of 2nd ACM Symposium on Cloud Computing (SoCC 2011), pp. 24:1–24:14. ACM (2011). doi:10.1145/2038916.2038940
73. Gandhi, A., Harchol-Balter, M., Das, R., Lefurgy, C.: Optimal power allocation in server farms. In: Proceedings of 11th International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2009), pp. 157–168. ACM (2009). doi:10.1145/1555349.1555368

74. Ganglia monitoring system. http://ganglia.sourceforge.net/ (2012)
75. Ghanbari, H., Simmons, B., Litoiu, M., Iszlai, G.: Exploring alternative approaches to implement an elasticity policy. In: Proceedings of 2011 IEEE International Conference on Cloud Computing (CLOUD 2011), pp. 716–723. IEEE (2011). doi:10.1109/CLOUD.2011.101
76. Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., Stoica, I.: Dominant resource fairness: fair allocation of multiple resource types. In: Proceedings of 8th USENIX Conference on Networked Systems Design and Implementation (NSDI 2011), pp. 24–24. USENIX (2011)
77. Ghorbani, S., Caesar, M.: Walk the line: consistent network updates with bandwidth guarantees. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN 2012), pp. 67–72. ACM, New York, NY (2012). doi:10.1145/2342441.2342455
78. Giurgiu, I., Castillo, C., Tantawi, A., Steinder, M.: Enabling efficient placement of virtual infrastructures in the cloud. In: Proceedings of 13th International Middleware Conference (Middleware 2012), pp. 332–353. (2012). http://dl.acm.org/citation.cfm?id=2442626.2442648
79. Glanz, J.: Power, pollution and the internet. The New York Times p. A1. http://www.nytimes.com/2012/09/23/technology/data-centers-waste-vast-amounts-of-energy-belying-industry-image.html (2012)
80. Gmach, D., Rolia, J., Cherkasova, L., Belrose, G., Turicchi, T., Kemper, A.: An integrated approach to resource pool management: policies, efficiency and quality metrics. In: Proceedings of 2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN 2008), pp. 326–335. IEEE (2008). doi:10.1109/DSN.2008.4630101
81. Gmach, D., Rolia, J., Cherkasova, L.: Chargeback model for resource pools in the cloud. In: Proceedings of 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011), pp. 622–625. IEEE (2011). doi:10.1109/INM.2011.5990586
82. Gmach, D., Krompass, S., Scholz, A., Wimmer, M., Kemper, A.: Adaptive quality of service management for enterprise services. ACM Trans. Web **2**(1), 8:1–8:46 (2008). doi:10.1145/1326561.1326569
83. Gong, Z., Gu, X., Wilkes, J.: PRESS: PRedictive elastic ReSource scaling for cloud systems. In: Proceedings of 6th International Conference on Network and Service Management (CNSM 2010), pp. 9–16. IFIP (2010). doi:10.1109/CNSM.2010.5691343
84. Google app engine pricing. http://cloud.google.com/pricing/ (2012)
85. Google app engine. https://developers.google.com/appengine/ (2012)
86. Govindan, S., Liu, J., Kansal, A., Sivasubramaniam, A.: Cuanta: quantifying effects of shared on-chip resource interference for consolidated virtual machines. In: Proceedings of 2nd ACM Symposium on Cloud Computing (SoCC 2011), pp. 22:1–22:14. ACM (2011). doi:10.1145/2038916.2038938
87. Group, D.M.T.F.C.M.W.: Cloud infrastructure management interface (CIMI) model and REST interface over HTTP specification. http://dmtf.org/standards/cmwg (2012)
88. Gulati, A., Ahmad, I., Waldspurger, C.A.: PARDA: proportional allocation of resources for distributed storage access. In: Proceedings of USENIX 7th Conference on File and Storage Technologies (FAST 2009), pp. 85–98. USENIX (2009)
89. Gulati, A., Holler, A., Ji, M., Shanmuganathan, G., Waldspurger, C., Zhu, X.: VMware distributed resource management: design, implementation, and lessons learned. VMware Tech. J. **1**(1), 45–64 (2012). http://labs.vmware.com/publications/gulati-vmtj-spring2012
90. Gulati, A., Kumar, C., Ahmad, I., Kumar, K.: BASIL: automated IO load balancing across storage devices. In: Proceedings of 8th USENIX Conference on File and Storage Technologies (FAST 2010), pp. 169–182. USENIX (2010)
91. Gulati, A., Merchant, A., Varman, P.J.: mClock: handling throughput variability for hypervisor IO scheduling. In: Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI 2010, p. 1–7. USENIX Association, Berkeley, CA (2010). http://dl.acm.org/citation.cfm?id=1924943.1924974
92. Gulati, A., Shanmuganathan, G., Ahmad, I., Waldspurger, C., Uysal, M.: Pesto: online storage performance management in virtualized datacenters. In: Proceedings of 2nd ACM Symposium on Cloud Computing (SoCC 2011), pp. 19:1–19:14. ACM (2011). doi:10.1145/2038916.2038935
93. Gulati, A., Shanmuganathan, G., Holler, A., Irfan, A.: Cloud scale resource management: challenges and techniques. In: Proceedings of 3rd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2011) (2011)
94. Gulati, A., Shanmuganathan, G., Zhang, X., Varman, P.J.: Demand based hierarchical QoS using storage resource pools. In: Proceedings of 2012 USENIX Annual Technical Conference (ATC 2012). USENIX (2012)

95. Guo, C., Lu, G., Wang, H.J., Yang, S., Kong, C., Sun, P., Wu, W., Zhang, Y.: SecondNet: a data center network virtualization architecture with bandwidth guarantees. In: Proceedings of 6th International on emerging Networking EXperiments and Technologies (CoNEXT 2010), pp. 15:1–15:12. ACM (2010). doi:10.1145/1921168.1921188
96. Guo, T., Sharma, U., Wood, T., Sahu, S., Shenoy, P.: Seagull: intelligent cloud bursting for enterprise applications. In: Proceedings of 2012 USENIX Annual Technical Conference (ATC 2012). USENIX (2012)
97. Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., Shi, Y., Tian, C., Zhang, Y., Lu, S.: BCube: a high performance, server-centric network architecture for modular data centers. SIGCOMM Comput. Commun. Rev. **39**(4), 63–74 (2009). doi:10.1145/1594977.1592577
98. Gupta, R., Bose, S., Sundarrajan, S., Chebiyam, M., Chakrabarti, A.: A two stage heuristic algorithm for solving the server consolidation problem with item-item and bin-item incompatibility constraints. In: Proceedings of 2008 IEEE International Conference on Services Computing (SCC 2008), vol. 2, pp. 39–46. IEEE (2008)
99. Hamilton, J.R.: Cooperative expendable micro-slice servers (CEMS): low cost, low power servers for internet-scale services. In: Proceedings of 4th Biennial Conference on Innovative Data Systems (CIDR 2009) (2009)
100. Han, R., Guo, L., Ghanem, M., Guo, Y.: Lightweight resource scaling for cloud applications. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 644–651. IEEE (2012). doi:10.1109/CCGrid.52
101. Hellerstein, J., Parekh, S., Diao, Y., Tilbury, D.M.: Feedback Control of Computing Systems. Wiley, London (2004)
102. Herbst, N.R., Kounev, S., Ruessner, R.: Elasticity in cloud computing: what it is, and what it is not. In: Proceedings of 10th International Conference on Autonomic Computing (ICAC 2013), pp. 23–27. USENIX (2013)
103. Herodotou, H., Dong, F., Babu, S.: No one (cluster) size fits all: automatic cluster sizing for data-intensive analytics. In: Proceedings of 2nd ACM Symposium on Cloud Computing (SoCC 2011), pp. 18:1–18:14. ACM (2011). doi:10.1145/2038916.2038934
104. Hindman, B., Konwinski, A., Zaharia, M., Ghodsi, A., Joseph, A.D., Katz, R., Shenker, S., Stoica, I.: Mesos: a platform for fine-grained resource sharing in the data center. In: Proceedings of 8th USENIX Conference on Networked Systems Design and Implementation (NSDI 2011), pp. 22–22. USENIX (2011)
105. Hines, M.R., Gopalan, K.: Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning. In: Proceedings of 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE 2009), pp. 51–60. ACM (2009). doi:10.1145/1508293.1508301
106. Horvath, T., Skadron, K.: Multi-mode energy management for multi-tier server clusters. In: Proceedings of 17th International Conference on Parallel Architectures and Compilation Techniques (PACT 2008), pp. 270–279. ACM (2008). doi:10.1145/1454115.1454153
107. Hu, L., Ryu, K.D., Silva, M., Schwan, K.: v-bundle: Flexible group resource offerings in clouds. In: Proceedings of 32nd IEEE International Conference on Distributed Computing Systems (ICDCS 2012), pp. 406–415. IEEE (2012). doi:10.1109/ICDCS.2012.61
108. IBM software–WebSphere extended deployment. http://www-01.ibm.com/software/webservers/appserv/extend/ (2012)
109. Ilyas, M., Raza, S., Chen, C.C., Uzmi, Z., Chuah, C.N.: RED-BL: energy solution for loading data centers. In: Proceedings of 2012 IEEE International Conference on Computer Communications (Infocom 2012), pp. 2866–2870. IEEE (2012). doi:10.1109/INFCOM.2012.6195717
110. Isard, M., Budiu, M., Yu, Y., Birrell, A., Fetterly, D.: Dryad: distributed data-parallel programs from sequential building blocks. In: Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007, EuroSys ’07, p. 59–72. ACM, New York, NY (2007). doi:10.1145/1272996.1273005 http://doi.acm.org/10.1145/1272996.1273005
111. Ishakian, V., Bestavros, A.: MORPHOSYS: efficient colocation of QoS-Constrained workloads in the cloud. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 90–97. IEEE (2012). doi:10.1109/CCGrid.44
112. Ishakian, V., Sweha, R., Bestavros, A., Appavoo, J.: CloudPack: exploiting workload flexibilty through rational pricing. In: P. Narasimhan, P. Triantafillou (eds.) Proceedings of 2012 International Middleware Conference (Middleware 2012), pp. 374–393. Springer, Berlin Heidelberg (2012). http://link.springer.com/chapter/10.1007/978-3-642-35170-9_19

113. Jalaparti, V., Ballani, H., Costa, P., Karagiannis, T., Rowstron, A.: Bridging the tenant-provider gap in cloud services. In: Proceedings of the Third ACM Symposium on Cloud Computing (SoCC 2012), p. 10:1–10:14. ACM, New York, NY (2012). doi:10.1145/2391229.2391239 http://doi.acm.org/10.1145/2391229.2391239
114. Javadi, B., Thulasiramy, R., Buyya, R.: Statistical modeling of spot instance prices in public cloud environments. In: Proceedings of 4th IEEE International Conference on Utility and Cloud Computing (UCC 2011), pp. 219–228. IEEE (2011). doi:10.1109/UCC.2011.37
115. Jayasinghe, D., Pu, C., Eilam, T., Steinder, M., Whally, I., Snible, E.: Improving performance and availability of services hosted on IaaS clouds with structural constraint-aware virtual machine placement. In: Proceedings of 2011 IEEE International Conference on Services Computing (SCC 2011), pp. 72–79. IEEE (2011). doi:10.1109/SCC.2011.28
116. Jeyakumar, V., Alizadeh, M., Mazieres, D., Prabhakar, B., Kim, C.: EyeQ: practical network performance isolation for the multi-tenant cloud. In: Proceedings of 4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2012) (2012)
117. Jiang, J.W., Lan, T., Ha, S., Chen, M., Chiang, M.: Joint VM placement and routing for data center traffic engineering. In: Proceedings of 2012 IEEE International Conference on Computer Communications (Infocom 2012), pp. 2876–2880. IEEE (2012)
118. Jing, S.Y., Ali, S., She, K., Zhong, Y.: State-of-the-art research study for green cloud computing. The Journal of Supercomputing pp. 1–24 (2011). doi:10.1007/s11227-011-0722-1
119. John Wilkes, 2011 GAFS Omega. http://youtu.be/0ZFMlO98Jkc (2011)
120. Jung, G., Hiltunen, M., Joshi, K., Schlichting, R., Pu, C.: Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures. In: Proceedings of IEEE 30th International Conference on Distributed Computing Systems (ICDCS 2010), pp. 62–73 (2010). doi:10.1109/ICDCS.2010.88
121. Jung, G., Joshi, K., Hiltunen, M., Schlichting, R., Pu, C.: Generating adaptation policies for multi-tier applications in consolidated server environments. In: Proceedings of 2008 International Conference on Autonomic Computing (ICAC 2008), pp. 23–32. IEEE (2008). doi:10.1109/ICAC.2008.21
122. Kannan, S., Gavrilovska, A., Schwan, K.: Cloud4Home–enhancing data services with @Home clouds. In: Proceedings of 31st IEEE International Conference on Distributed Computing Systems (ICDCS 2011), pp. 539–548. IEEE (2011). doi:10.1109/ICDCS.2011.74
123. Kansal, A., Zhao, F., Liu, J., Kothari, N., Bhattacharya, A.A.: Virtual machine power metering and provisioning. In: Proceedings of 1st ACM Symposium on Cloud Computing (SoCC 2010), pp. 39–50. ACM (2010). doi:10.1145/1807128.1807136
124. Khajeh-Hosseini, A., Greenwood, D., Sommerville, I.: Cloud migration: A case study of migrating an enterprise IT system to IaaS. In: Proceedings of 3rd IEEE International Conference on Cloud Computing (CLOUD 2010), pp. 450–457. IEEE (2010). doi:10.1109/CLOUD.2010.37
125. Khajeh-Hosseini, A., Greenwood, D., Smith, J.W., Sommerville, I.: The cloud adoption toolkit: supporting cloud adoption decisions in the enterprise. Softw. Pract. Exp. **42**(4), 447–465 (2012). doi:10.1002/spe.1072
126. Kikuchi, S., Matsumoto, Y.: What will happen if cloud management operations burst out? In: Proceedings of 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011), pp. 97–104. IEEE (2011). doi:10.1109/INM.2011.5990679
127. Kleinberg, J., Rabani, Y., Tardos, É: Allocating bandwidth for bursty connections. In: Proceedings of 29th Annual ACM Symposium on Theory of Computing (STOC 1997), pp. 664–673. ACM (1997). doi:10.1145/258533.258661
128. Kochut, A., Karve, A.: Leveraging local image redundancy for efficient virtual machine provisioning. In: Proceedings of 13th IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), pp. 179–187. IEEE (2012). doi:10.1109/NOMS.2012.6211897
129. Kocoloski, B., Ouyang, J., Lange, J.: A case for dual stack virtualization: consolidating HPC and commodity applications in the cloud. In: Proceedings of the Third ACM Symposium on Cloud Computing (SoCC 2012, p. 23:1–23:7. ACM, New York, NY (2012). doi:10.1145/2391229.2391252
130. Konstanteli, K., Cucinotta, T., Psychas, K., Varvarigou, T.: Admission control for elastic cloud services. In: Proceedings of 5th IEEE International Conference on Cloud Computing (CLOUD 2012), pp. 41–48. IEEE (2012)
131. Koomey, J.G.: Growth in data center electricity use 2005 to 2010. Technical report, Analytics Press (2011). http://www.analtticspress.com/datacenters.html

132. Koslovski, G., Soudan, S., Goncalves, P., Vicat-Blanc, P.: Locating virtual infrastructures: users and InP perspectives. In: Proceedings of 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011), pp. 153–160. IEEE (2011). doi:10.1109/INM.2011.5990686
133. Kremien, O., Kramer, J.: Methodical analysis of adaptive load sharing algorithms. IEEE Trans. Parallel Distrib. Syst. **3**(6), 747–760 (1992). doi:10.1109/71.180629
134. Kumar, G., Chowdhury, M., Ratnasamy, S., Stoica, I.: A case for performance-centric network allocation. In: Proceedings of 4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2012). USENIX (2012)
135. Kumar, S., Talwar, V., Kumar, V., Ranganathan, P., Schwan, K.: Loosely coupled coordinated management in virtualized data centers. Clust. Comput. **14**(3), 259–274 (2011). doi:10.1007/s10586-010-0124-9
136. Kyriazis, D.: Cloud computing service level agreements–exploitation of research results. Technical report, European Commission, Brussels (2013). http://ec.europa.eu/digital-agenda/en/news/cloud-computing-service-level-agreements-exploitation-research-results
137. Lakshman, A., Malik, P.: Cassandra: a decentralized structured storage system. SIGOPS Oper. Syst. Rev. **44**(2), 35–40 (2010). doi:10.1145/1773912.1773922
138. Lee, G., Chun, B.G., Katz, R.H.: Heterogeneity-aware resource allocation and scheduling in the cloud. In: Proceedings of 3rd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2011). USENIX (2011)
139. Levy, R., Nagarajarao, J., Pacifici, G., Spreitzer, A., Tantawi, A., Youssef, A.: Performance management for cluster based web services. In: Proceedings of 8th IFIP/IEEE International Symposium on Integrated Network Management (IM 2003), pp. 247–261 (2003). doi:10.1109/INM.2003.1194184
140. Li, B., Li, J., Huai, J., Wo, T., Li, Q., Zhong, L.: EnaCloud: an energy-saving application live placement approach for cloud computing environments. In: Proceedings of 2009 IEEE International Conference on Cloud Computing (CLOUD 2009), pp. 17–24. IEEE (2009). doi:10.1109/CLOUD.2009.72
141. Lim, H., Kansal, A., Liu, J.: Power budgeting for virtualized data centers. In: Proceedings of 2011 USENIX Annual Technical Conference (ATC 2011). USENIX (2011)
142. Lin, M., Wierman, A., Andrew, L., Thereska, E.: Dynamic right-sizing for power-proportional data centers. In: Proceedings of 2011 IEEE International Conference on Computer Communicaitons (Infocom 2011), pp. 1098–1106. IEEE (2011). doi:10.1109/INFCOM.2011.5934885
143. Ling, X., Jin, H., Ibrahim, S., Cao, W., Wu, S.: Efficient disk I/O scheduling with QoS guarantee for xen-based hosting platforms. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 81–89. IEEE (2012). doi:10.1109/CCGrid.17
144. Liu, Z., Chen, Y., Bash, C., Wierman, A., Gmach, D., Wang, Z., Marwah, M., Hyser, C.: Renewable and cooling aware workload management for sustainable data centers. In: Proceedings of 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/PERFORMANCE 2012), pp. 175–186. ACM (2012). doi:10.1145/2254756.2254779
145. Liu, Z., Cho, S.: Characterizing machines and workloads on a google cluster. In: Proceedings of 8th International Workshop on Scheduling and Resource Management for Parallel and Distributed Systems (SRMPDS 2012) (2012)
146. Liu, H., Jin, H., Liao, X., Hu, L., Yu, C.: Live migration of virtual machine based on full system trace and replay. In: Proceedings of 18th ACM International Symposium on High Performance Distributed Computing (HPDC 2009), pp. 101–110. ACM (2009). doi:10.1145/1551609.1551630
147. Liu, C., Loo, B.T., Mao, Y.: Declarative automated cloud resource orchestration. In: Proceedings of 2nd ACM Symposium on Cloud Computing (SoCC 2011), pp. 26:1–26:8. ACM (2011). doi:10.1145/2038916.2038942
148. Liu, C., Mao, Y., Chen, X., Fernandez, M.F., Loo, B.T., van der Merwe, J.: TROPIC: transactional resource orchestration platform in the cloud. In: Proceedings of 2012 USENIX Annual Technical Conference (ATC 2012). USENIX (2012)
149. Lu, Y., Xie, Q., Kliot, G., Geller, A., Larus, J.R., Greenberg, A.: Join-idle-queue: a novel load balancing algorithm for dynamically scalable web services. Perform. Eval. **68**(11), 1056–1071 (2011). doi:10.1016/j.peva.2011.07.015
150. Macias, M., Guitart, J.: Client classification policies for SLA enforcement in shared cloud datacenters. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 156–163. IEEE (2012). doi:10.1109/CCGrid.15

151. Mann, V., Vishnoi, A., Kannan, K., Kalyanaraman, S.: CrossRoads: seamless VM mobility across data centers through software defined networking. In: Proceedings of 13th IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), pp. 88–96. IEEE (2012). doi:10.1109/NOMS.2012.6211886
152. Marinos, A., Briscoe, G.: Community cloud computing. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) Cloud Computing, LNSC 5931, pp. 472–484. Springer, Berlin (2009). http://link.springer.com/chapter/10.1007/978-3-642-10665-1_43
153. Meng, X., Isci, C., Kephart, J., Zhang, L., Bouillet, E., Pendarakis, D.: Efficient resource provisioning in compute clouds via VM multiplexing. In: Proceedings of 7th International Conference on Autonomic Computing (ICAC 2010), pp. 11–20. ACM (2010). doi:10.1145/1809049.1809052
154. Meng, S., Iyengar, A.K., Rouvellou, I.M., Liu, L.: Volley: Violation likelihood based state monitoring for datacenters. In: Proceedings of 33rd IEEE International Conference on Distributed Computing Systems (ICDCS 2013), pp. 1–10. IEEE (2013)
155. Meng, X., Pappas, V., Zhang, L.: Improving the scalability of data center networks with traffic-aware virtual machine placement. In: Proceedings of 2010 IEEE International Conference on Computer Communications (Infocom 2010), pp. 1154–1162. IEEE (2010)
156. Meng, S., Liu, L., Wang, T.: State monitoring in cloud datacenters. IEEE Trans. Knowl. Data Eng. **23**(9), 1328–1344 (2011). doi:10.1109/TKDE.2011.70
157. Microsoft azure. http://microsoft.com/azure (2012)
158. Mitzenmacher, M.: The power of two choices in randomized load balancing. IEEE Trans. Parallel Distrib. Syst. **12**(10), 1094–1104 (2001). doi:10.1109/71.963420
159. Moens, H., De Turck, F.: A scalable approach for structuring large-scale hierarchical cloud management systems. In: Proceedings of 9th International Conference on Network and Service Management (CNSM 2013), pp. 1–8. IFIP (2013)
160. Moens, H., Famaey, J., Latré, S., Dhoedt, B., De Turck, F.: Design and evaluation of a hierarchical application placement algorithm in large scale clouds. In: Proceedings of 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011), pp. 137–144. IEEE (2011). doi:10.1109/INM.2011.5990684
161. Moens, H., Truyen, E., Walraven, S., Joosen, W., Dhoedt, B., Turck, F.D.: Cost-effective feature placement of customizable multi-tenant applications in the cloud. J. Netw. Syst. Manag. (2013). doi:10.1007/s10922-013-9265-5
162. Mukherjee, J., Krishnamurthy, D., Rolia, J., Hyser, C.: Resource contention detection and management for consolidated workloads. In: Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 294–302. IEEE (2013)
163. Nagios—the industry standard in IT infrastructure monitoring. http://www.nagios.org/ (2012)
164. Nelson, M., Lim, B.H., Hutchins, G.: Fast transparent migration for virtual machines. In: Proceedings of 2005 USENIX Annual Technical Conference (USENIX 2005), pp. 391–394. USENIX (2005)
165. Nguyen, H., Shen, Z., Gu, X., Subbiah, S., Wilkes, J.: AGILE: elastic distributed resource scaling for infrastructure-as-a-service. In: Proceedings of 10th International Conference on Autonomic Computing (ICAC 2013), pp. 69–82. USENIX (2013)
166. Niu, D., Feng, C., Li, B.: Pricing cloud bandwidth reservations under demand uncertainty. SIGMETRICS Perform. Eval. Rev. **40**(1), 151–162 (2012). doi:10.1145/2318857.2254776
167. Novaković, D., Vasić, N., Novaković, S., Kostić, D., Bianchini, R.: DeepDive: transparently identifying and managing performance interference in virtualized environments. In: Proceedings of 2013 USENIX Annual Technical Conference (ATC 2013). USENIX (2013)
168. Padala, P., Hou, K.Y., Shin, K.G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A.: Automated control of multiple virtualized resources. In: Proceedings of 4th ACM European Conference on Computer Systems (EuroSys 2009), pp. 13–26. ACM (2009). doi:10.1145/1519065.1519068
169. Panigrahy, R., Talwar, K., Uyeda, L., Wieder, U.: Heuristics for vector bin packing. Technical report, Microsoft Research (2011)
170. Park, N., Ahmad, I., Lilja, D.J.: Romano: autonomous storage management using performance prediction in multi-tenant datacenters. In: Proceedings of the Third ACM Symposium on Cloud Computing (SoCC 2012), pp. 21:1–21:14. ACM, New York, NY (2012). doi:10.1145/2391229.2391250
171. Parolini, L., Tolia, N., Sinopoli, B., Krogh, B.H.: A cyber-physical systems approach to energy management in data centers. In: Proceedings of 1st ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS 2010), pp. 168–177. ACM (2010). doi:10.1145/1795194.1795218

172. Peng, C., Kim, M., Zhang, Z., Lei, H.: VDN: virtual machine image distribution network for cloud data centers. In: Proceedings of 2012 IEEE International Conference on Computer Communicatons (Infocom 2012), pp. 181–189. IEEE (2012). doi:10.1109/INFCOM.2012.6195556
173. Phillips, R.: Pricing and Revenue Optimization. Stanford University Press, Stanford (2005)
174. Popa, L., Krishnamurthy, A., Ratnasamy, S., Stoica, I.: FairCloud: sharing the network in cloud computing. In: Proceedings of 10th ACM Workshop on Hot Topics in Networks (HotNets-X), pp. 22:1–22:6. ACM (2011). doi:10.1145/2070562.2070584
175. Potharaju, R., Jain, N.: When the network crumbles: An empirical study of cloud network failures and their impact on services. In: Proceedings of 4th ACM Symposium on Cloud Computing (SoCC 2013). ACM (2013)
176. Pricing details: Windows Azure. http://www.windowsazure.com/en-us/pricing/details/#business-analytics (2012)
177. Qian, H., Rabinovich, M.: Application placement and demand distribution in a global elastic cloud: A unified approach. pp. 1–12. USENIX (2013)
178. Rabbani, M.G., Esteves, R.P., Podlesny, M., Simon, G., Zambenedetti Granville, L., Boutaba, R.: On tackling virtual data center embedding problem. In: Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 177–184. IEEE (2013)
179. Raiciu, C., Barre, S., Pluntke, C., Greenhalgh, A., Wischik, D., Handley, M.: Improving datacenter performance and robustness with multipath TCP. In: Proceedings of ACM SIGCOMM 2011 Conference on Data Communication (SIGCOMM 2011), pp. 266–277. ACM (2011). doi:10.1145/2018436.2018467
180. Rao, J., Bu, X., Wang, K., Xu, C.Z.: Self-adaptive provisioning of virtualized resources in cloud computing. In: Proceedings of 2011 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2011), pp. 129–130. ACM (2011). doi:10.1145/1993744.1993790
181. Reiss, C., Tumanov, A., Ganger, G.R., Katz, R.H., Kozuch, M.A.: Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In: Proceedings of 3rd ACM Symposium on Cloud Computing (SoCC 2012). ACM (2012)
182. Robinson, I., Webber, J., Eifrem, E.: Graph Databases, 1st edn. O'Reilly, Media (2013)
183. Roytman, A., Kansai, A., Govindan, S., Liu, J., Nath, S.: PACMan: performance aware virtual machine consolidation. In: Proceedings of 10th International Conference on Autonomic Computing (ICAC 2013), pp. 83–94. USENIX (2013)
184. Sandholm, T., Lai, K.: MapReduce optimization using regulated dynamic prioritization. In: Proceedings of 11th International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2009), pp. 299–310. ACM (2009). doi:10.1145/1555349.1555384
185. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for VM-Based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 14–23 (2009). doi:10.1109/MPRV.2009.82
186. Schad, J., Dittrich, J., Quian-Ruiz, J.A.: Runtime measurements in the cloud: observing, analyzing, and reducing variance. Proc. VLDB Endow. **3**(1–2), 460–471 (2010)
187. Schwarzkopf, M., Konwinski, A., Abd-El-Malek, M., Wilkes, J.: Omega: flexible, scalable schedulers for large compute clusters. In: Proceedings of 8th ACM European Conference on Computer Systems (EuroSys 2013), pp. 351–364. ACM, New York, NY (2013). doi:10.1145/2465351.2465386
188. Schwarzkopf, M., Murray, D.G., Hand, S.: The seven deadly sins of cloud computing research. In: Proceedings of 4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2012). USENIX (2012)
189. Seelam, S.R., Teller, P.J.: Virtual I/O scheduler: a scheduler of schedulers for performance virtualization. In: Proceedings of 3rd International Conference on Virtual Execution Environments (VEE 2007), pp. 105–115. ACM (2007). doi:10.1145/1254810.1254826
190. Sharma, U., Shenoy, P., Sahu, S., Shaikh, A.: A cost-aware elasticity provisioning system for the cloud. In: Proceedings of 31st IEEE International Conference on Distributed Computing Systems (ICDCS 2011), pp. 559–570. IEEE (2011). doi:10.1109/ICDCS.2011.59
191. Sharma, U., Shenoy, P., Sahu, S., Shaikh, A.: Kingfisher: Cost-aware elasticity in the cloud. In: Proceedings of 2011 IEEE International Conference on Computer Communications (Infocom 2011), pp. 206–210. IEEE (2011). doi:10.1109/INFCOM.2011.5935016
192. Sharma, B., Thulasiram, R., Thulasiraman, P., Garg, S., Buyya, R.: Pricing cloud compute commodities: a novel financial economic model. In: Proceedings of 12th IEEE/ACM International

Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 451–457. IEEE (2012). doi:10.1109/CCGrid.126

193. Shen, Z., Subbiah, S., Gu, X., Wilkes, J.: CloudScale: elastic resource scaling for multi-tenant cloud systems. In: Proceedings of 2nd ACM Symposium on Cloud Computing (SoCC 2011), pp. 5:1–5:14. ACM (2011). doi:10.1145/2038916.2038921
194. Shi, L., Butler, B., Botvich, D., Jennings, B.: Provisioning of requests for virtual machine sets with placement constraints in IaaS clouds. In: Proceedings of 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 499–505. IEEE (2013)
195. Shieh, A., Kandula, S., Greenberg, A., Kim, C., Saha, B.: Sharing the data center network. In: Proceedings of 8th USENIX Conference on Networked Systems Design and Implementation (NSDI 2011), pp. 23–23. USENIX (2011)
196. Shifrin, M., Atar, R., Cidon, I.: Optimal scheduling in the hybrid-cloud. In: Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 51–59. IEEE (2013)
197. Shin, J.Y., Wong, B., Sirer, E.G.: Small-world datacenters. In: Proceedings of the 2nd ACM Symposium on Cloud Computing (SoCC 2011), pp. 2:1–2:13. ACM (2011). doi:10.1145/2038916.2038918
198. Shrivastava, V., Zerfos, P., Lee, K.w., Jamjoom, H., Liu, Y.H., Banerjee, S.: Application-aware virtual machine migration in data centers. In: Proceedings of 2011 IEEE International Conference on Computer Communications (Infocom 2011), pp. 66–70. IEEE (2011). doi:10.1109/INFCOM.2011.5935247
199. Simmons, B., Ghanbari, H., Litoiu, M., Iszlai, G.: Managing a SaaS application in the cloud using PaaS policy sets and a strategy-tree. In: Proceedings of 7th International Conference on Network and Service Management (CNSM 2011), pp. 1–5. IEEE (2011)
200. Singh, R., Shenoy, P., Natu, M., Sadaphal, V., Vin, H.: Predico: a system for what-if analysis in complex data center applications. In: Proceedings of 12th International Middleware Conference (Middleware 2011), pp. 120–139. IFIP (2011). http://dl.acm.org/citation.cfm?id=2414338.2414348
201. Smith, J.W., Khajeh-Hosseini, A., Ward, J.S., Sommerville, I.: CloudMonitor: profiling power usage. In: Proceedings of 5th IEEE International Conference on Cloud Computing (CLOUD 2012), pp. 947–948. IEEE (2012)
202. Speitkamp, B., Bichler, M.: A mathematical programming approach for server consolidation problems in virtualized data centers. IEEE Trans. Serv. Comput. **3**(4), 266–278 (2010). doi:10.1109/TSC.2010.25
203. Spillner, J., Brito, A.: Lifting cloud infrastructure service consumers to providers. Tiny Trans. Comput. Sci. **1**(1) (2012). http://tinytocs.org/vol1/papers/tinytocs-v1-spillner-brito.pdf
204. Srikantaiah, S., Kansal, A., Zhao, F.: Energy aware consolidation for cloud computing. In: Proceedings of 2008 Conference on Power Aware Computing and Systems (HotPower 2008), pp. 10–10. USENIX (2008)
205. Stadler, R., Dam, M., Gonzalez, A., Wuhib, F.: Decentralized real-time monitoring of network-wide aggregates. In: Proceedings of 2nd Workshop on Large-Scale Distributed Systems and Middleware (LADIS 2008), pp. 7:1–7:6. ACM (2008). doi:10.1145/1529974.1529984
206. Storm, distributed and fault-tolerant realtime computation. http://storm-project.net/ (2013)
207. Sukwong, O., Sangpetch, A., Kim, H.: SageShift: managing SLAs for highly consolidated cloud. In: Proceedings of 2012 IEEE International Conference on Computer Communicatons (Infocom 2012), pp. 208–216. IEEE (2012). doi:10.1109/INFCOM.2012.6195591
208. Sulistio, A., Kim, K.H., Buyya, R.: Managing cancellations and no-shows of reservations with overbooking to increase resource revenue. In: Proceedings of 8th IEEE International Symposium on Cluster Computing and the Grid, 2008 (CCGRID 2008), pp. 267–276. IEEE (2008). doi:10.1109/CCGRID.2008.65
209. Sumbaly, R., Kreps, J., Gao, L., Feinberg, A., Soman, C., Shah, S.: Serving large-scale batch computed data with project voldemort. In: Proceedings of 10th USENIX Conference on File and Storage Technologies (FAST 2012), pp. 18–18. USENIX (2012)
210. Tan, J., Meng, X., Zhang, L.: Performance analysis of coupling scheduler for MapReduce/Hadoop. In: Proceedings of 2012 IEEE International Conference on Computer Communicatons (Infocom 2012), pp. 2586–2590. IEEE (2012). doi:10.1109/INFCOM.2012.6195658
211. Tang, C., Steinder, M., Spreitzer, M., Pacifici, G.: A scalable application placement controller for enterprise data centers. In: Proceedings of 16th International Conference on World Wide Web (WWW 2007), pp. 331–340. ACM (2007). doi:10.1145/1242572.1242618

212. Toffetti, G., Gambi, A., Pezz, M., Pautasso, C.: Engineering autonomic controllers for virtualized web applications. In: Benatallah, B., Casati, F., Kappel, G., Rossi, G. (eds.) Web Engineering, No. 6189 in LNCS, pp. 66–80. Springer, Berlin (2010)
213. Trushkowsky, B., Bodík, P., Fox, A., Franklin, M.J., Jordan, M.I., Patterson, D.: The SCADS director: scaling a distributed storage system under stringent performance requirements. In: Proceedings of 9th USENIX Conference on File and Storage Technologies (FAST 2011), pp. 12–12. USENIX (2011)
214. Tudoran, R., Costan, A., Antoniu, G., Soncu, H.: TomusBlobs: towards communication-efficient storage for MapReduce applications in azure. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 427–434. IEEE (2012). doi:10.1109/CCGrid.104
215. Tumanov, A., Cipar, J., Ganger, G.R., Kozuch, M.A.: alsched: algebraic scheduling of mixed workloads in heterogeneous clouds. In: Proceedings of the Third ACM Symposium on Cloud Computing (SoCC 2012), pp. 25:1–25:7. ACM, New York, NY (2012). doi:10.1145/2391229.2391254
216. Urgaonkar, R., Kozat, U., Igarashi, K., Neely, M.: Dynamic resource allocation and power management in virtualized data centers. In: Proceedings of 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010), pp. 479–486. IEEE (2010). doi:10.1109/NOMS.2010.5488484
217. Urgaonkar, B., Shenoy, P., Roscoe, T.: Resource overbooking and application profiling in shared hosting platforms. SIGOPS Oper. Syst. Rev. **36**(SI), 239–254 (2002). doi:10.1145/844128.844151
218. Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M., Tantawi, A.: An analytical model for multi-tier internet services and its applications. SIGMETRICS Perform. Eval. Rev. **33**(1), 291–302 (2005). doi:10.1145/1071690.1064252
219. Vasić, N., Barisits, M., Salzgeber, V., Kostic, D.: Making cluster applications energy-aware. In: Proceedings of 1st workshop on Automated control for datacenters and clouds (ACDC 2009), pp. 37–42. ACM (2009). doi:10.1145/1555271.1555281
220. Vavilapalli, V.K., Murthy, A.C., Douglas, C., Agarwal, S., Konar, M., Evans, R., Graves, T., Lowe, J., Shah, H., Seth, S., Saha, B., Curino, C., O'Malley, O., Radia, S., Reed, B., Baldeschwieler, E.: Apache hadoop YARN: yet another resource negotiator. In: Proceedings of 4th ACM Symposium on Cloud Computing (SoCC 2013). ACM (2013)
221. Verma, A., Dasgupta, G., Nayak, T.K., De, P., Kothari, R.: Server workload analysis for power minimization using consolidation. In: Proceedings of 2009 USENIX Annual Technical Conference (USENIX 2009), p. 28. USENIX (2009)
222. Villegas, D., Antoniou, A., Sadjadi, S., Iosup, A.: An analysis of provisioning and allocation policies for infrastructure-as-a-service clouds. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 612–619. IEEE (2012). doi:10.1109/CCGrid.46
223. Viswanathan, B., Verma, A., Dutta, S.: CloudMap: workload-aware placement in private heterogeneous clouds. In: Proceedings of 13th IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), pp. 9–16. IEEE (2012). doi:10.1109/NOMS.2012.6211877
224. VMware vSphere private cloud computing and virtualization. http://www.vmware.com/products/datacenter-virtualization/vsphere/overview.html (2012)
225. Voorsluys, W., Broberg, J., Venugopal, S., Buyya, R.: Cost of virtual machine live migration in clouds: a performance evaluation. In: Jaatun, M., Zhao, G., Rong, C. (eds.) Cloud Computing, LNCS, Vol. 5931, pp. 254–265. Springer, Berlin (2009)
226. Wang, W., Li, B., Liang, B.: To reserve or not to reserve: Optimal online multi-instance acquisition in IaaS clouds. In: Proceedings of 10th International Conference on Autonomic Computing (ICAC 2013), pp. 13–22. USENIX (2013)
227. Wang, F., Liu, J., Chen, M.: CALMS: cloud-assisted live media streaming for globalized demands with time/region diversities. In: Proceedings of 2012 IEEE International Conference on Computer Communicatons (Infocom 2012), pp. 199–207. IEEE (2012). doi:10.1109/INFCOM.2012.6195578
228. Wang, M., Meng, X., Zhang, L.: Consolidating virtual machines with dynamic bandwidth demand in data centers. In: Proceedings of 2011 IEEE International Conference on Computer Communicaitons (Infocom 2011), pp. 71–75. IEEE (2011). doi:10.1109/INFCOM.2011.5935254
229. Wang, W., Niu, D., Li, B., Liang, B.: Dynamic cloud resource reservation via cloud brokerage. In: Proceedings of 33rd IEEE International Conference on Distributed Computing Systems (ICDCS 2013), pp. 400–409. IEEE (2013)

230. Wang, P., Qi, Y., Hui, D., Rao, L., Liu, X.: Present or future: Optimal pricing for spot instances. In: Proceedings of 33rd IEEE International Conference on Distributed Computing Systems (ICDCS 2013), pp. 410–419. IEEE (2013)
231. Wang, Q., Ren, K., Meng, X.: When cloud meets eBay: towards effective pricing for cloud computing. In: Proceedings of 2012 IEEE International Conference on Computer Communicatons (Infocom 2012), pp. 936–944. IEEE (2012). doi:10.1109/INFCOM.2012.6195844
232. Wang, Z., Tolia, N., Bash, C.: Opportunities and challenges to unify workload, power, and cooling management in data centers. In: Proceedings of 5th International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks (FeBiD 2010), pp. 1–6. ACM (2010). doi:10.1145/1791204.1791205
233. Wang, A., Venkataraman, S., Alspaugh, S., Katz, R., Stoica, I.: Cake: enabling high-level SLOs on shared storage systems. In: Proceedings of the Third ACM Symposium on Cloud Computing (SoCC 2012), pp. 14:1–14:14. ACM, New York, NY (2012). doi:10.1145/2391229.2391243
234. Wang, Y.T., Morris, R.: Load sharing in distributed systems. IEEE Trans. Comput. **C–34**(3), 204–217 (1985). doi:10.1109/TC.1985.1676564
235. Warneke, D., Kao, O.: Exploiting dynamic resource allocation for efficient parallel data processing in the cloud. IEEE Trans. Parallel Distrib. Syst. **22**(6), 985–997 (2011). doi:10.1109/TPDS.2011.65
236. Welcome to Apache Hadoop!. http://hadoop.apache.org/ (2012)
237. Wen, X., Chen, K., Chen, Y., Liu, Y., Xia, Y., Hu, C.: VirtualKnotter: online virtual machine shuffling for congestion resolving in virtualized datacenter. In: Proceedings of 32nd IEEE International Conference on Distributed Computing Systems (ICDCS 2012), pp. 12–21. IEEE (2012). doi:10.1109/ICDCS.2012.25
238. Weng, D., Bauer, M.: Using policies to drive autonomic management of virtual systems. In: Proceedings of 6th International Conference on Network and Service Management (CNSM 2010), pp. 258–261. IFIP (2010). doi:10.1109/CNSM.2010.5691193
239. Wilcox, D., McNabb, A., Seppi, K.: Solving virtual machine packing with a reordering grouping genetic algorithm. In: Proceedings of 2011 IEEE Congress on Evolutionary Computation (CEC), pp. 362–369. IEEE (2011)
240. Wilkes, J., Reiss, C.: Details of the ClusterData-2011-1 trace (2011). https://code.google.com/p/googleclusterdata/wiki/ClusterData2011_1
241. Wilson, C., Ballani, H., Karagiannis, T., Rowtron, A.: Better never than late: meeting deadlines in datacenter networks. In: Proceedings of ACM SIGCOMM 2011 Conference on Data Communication (SIGCOMM 2011), pp. 50–61. ACM (2011). doi:10.1145/2018436.2018443
242. Wood, T., Shenoy, P., Venkataramani, A., Yousif, M.: Black-box and gray-box strategies for virtual machine migration. In: Proceedings of 4th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2007), vol. 7, pp. 229–242. USENIX (2007)
243. Wu, Y., Wu, C., Li, B., Zhang, L., Li, Z., Lau, F.: Scaling social media applications into geo-distributed clouds. In: Proceedings of 2012 IEEE International Conference on Computer Communications (Infocom 2012), pp. 684–692. IEEE (2012). doi:10.1109/INFCOM.2012.6195813
244. Wuhib, F., Stadler, R., Lindgren, H.: Dynamic resource allocation with management objectives–implementation for an OpenStack cloud. In: Proceedings of 8th International Conference on Network and Services Management (CNSM 2012), pp. 309–315. IEEE (2012)
245. Wuhib, F., Stadler, R., Spreitzer, M.: Gossip-based resource management for cloud environments. In: Proceedings of 6th IEEE International Conference on Network and Service Management (CNSM 2010), pp. 1–8. IEEE (2010). doi:10.1109/CNSM.2010.5691347
246. Wuhib, F., Yanggratoke, R., Stadler, R.: Allocating compute and network resources under management objectives in large-scale clouds. J. Netw. Syst. Manag. (2013). doi:10.1007/s10922-013-9280-6
247. Wuhib, F., Stadler, R., Spreitzer, M.: A gossip protocol for dynamic resource management in large cloud environments. IEEE Trans. Netw. Serv. Manag. **9**(2), 213–225 (2012). doi:10.1109/TNSM.2012.031512.110176
248. Xen hypervisor. http://www.xen.org/ (2012)
249. Xu, H., Feng, C., Li, B.: Temperature aware workload management in geo-distributed datacenters. In: Proceedings of 10th International Conference on Autonomic Computing (ICAC 2013), pp. 303–314. USENIX (2013)
250. Xu, J., Fortes, J.: A multi-objective approach to virtual machine management in datacenters. In: Proceedings of 8th ACM International Conference on Autonomic Computing (ICAC 2011), pp. 225–234. ACM (2011). doi:10.1145/1998582.1998636

251. Xu, C., Gamage, S., Lu, H., Kompella, R.R., Xu, D.: vTurbo: accelerating virtual machine I/O processing using designated turbo-sliced core. In: Proceedings of 2013 USENIX Annual Technical Conference (ATC 2013). USENIX (2013)
252. Xu, H., Li, B.: Egalitarian stable matching for VM migration in cloud computing. In: Proceedings of 2011 IEEE Conference on Computer Communications Workshops (Infocom Workshops 2011), pp. 631–636. IEEE (2011). doi:10.1109/INFCOMW.2011.5928889
253. Xu, H., Li, B.: Maximizing revenue with dynamic cloud pricing: the infinite horizon case. pp. 2929–2933. IEEE (2012)
254. Xu, D., Liu, X.: Geographic trough filling for internet datacenters. In: Proceedings of 2012 IEEE International Conference on Computer Communicatons (Infocom 2012), pp. 2881–2885. IEEE (2012). doi:10.1109/INFCOM.2012.6195720
255. Xu, J., Zhao, M., Fortes, J., Carpenter, R., Yousif, M.: Autonomic resource management in virtualized data centers using fuzzy logic-based approaches. Clus. Comput. **11**(3), 213–227 (2008). doi:10.1007/s10586-008-0060-0
256. Yanggratoke, R., Kreitz, G., Goldmann, M., Stadler, R.: Predicting response times for the spotify backend. In: Proceedings of 8th International Conference on Network and Services Management (CNSM 2012), pp. 117–125. IEEE (2012)
257. Yanggratoke, R., Wuhib, F., Stadler, R.: Gossip-based resource allocation for green computing in large clouds. In: Proceedings of 7th International Conference on Network and Service Management (CNSM 2011), pp. 1–9. IEEE (2011)
258. Yao, Y., Huang, L., Sharma, A., Golubchik, L., Neely, M.: Data centers power reduction: A two time scale approach for delay tolerant workloads. In: Proceedings of 2012 IEEE International Conference on Computer Communicatons (Infocom 2012), pp. 1431–1439. IEEE (2012). doi:10.1109/INFCOM.2012.6195508
259. Yazir, Y., Matthews, C., Farahbod, R., Neville, S., Guitouni, A., Ganti, S., Coady, Y.: Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis. In: Proceedings of 3rd IEEE International Conference on Cloud Computing (CLOUD 2010), pp. 91–98. IEEE (2010). doi:10.1109/CLOUD.2010.66
260. Zaharia, M., Konwinski, A., Joseph, A.D., Katz, R., Stoica, I.: Improving MapReduce performance in heterogeneous environments. In: Proceedings of 8th USENIX Conference on Operating Systems Design and Implementation (OSDI 2008), pp. 29–42. USENIX (2008)
261. Zaman, S., Grosu, D.: Combinatorial auction-based mechanisms for VM provisioning and allocation in clouds. In: Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 729–734. IEEE (2012). doi:10.1109/CCGrid.22
262. Zhang, X., Shae, Z.Y., Zheng, S., Jamjoom, H.: Virtual machine migration in an over-committed cloud. In: Proceedings of 13th IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), pp. 196–203. IEEE (2012). doi:10.1109/NOMS.2012.6211899
263. Zhang, Y., Wang, Y., Wang, X.: GreenWare: greening cloud-scale data centers to maximize the use of renewable energy. In: Proceedings of 12th ACM/IFIP/USENIX International Conference on Middleware (Middleware 2011), pp. 143–164. Springer, Berlin (2011). doi:10.1007/978-3-642-25821-3_8
264. Zhang, S., Wu, H., Wang, W., Yang, B., Liu, P., Vasilakos, A.V.: Distributed workload and response time management for web applications. In: Proceedings of 7th International Conference on Network and Services Management (CNSM 2011), pp. 198–206. IFIP (2011)
265. Zhang, Q., Zhani, M.F., Zhu, Q., Zhang, S., Boutaba, R., Hellerstein, J.: Dynamic energy-aware capacity provisioning for cloud computing environments. In: Proceedings of 2012 International Conference on Autonomic Computing (ICAC 2012). IEEE (2012)
266. Zhang, Q., Zhu, Q., Zhani, M., Boutaba, R.: Dynamic service placement in geographically distributed clouds. In: Proceedings of 32nd IEEE International Conference on Distributed Computing Systems (ICDCS 2012), pp. 526–535. IEEE (2012). doi:10.1109/ICDCS.2012.74
267. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. **1**(1), 7–18 (2010). doi:10.1007/s13174-010-0007-6
268. Zhani, M.F., Zhang, Q., Simon, G., Boutaba, R.: VDC planner–dynamic migration-aware virtual data center embedding for clouds. In: Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 18–25. IEEE (2013)
269. Zhu, X., Young, D., Watson, B., Wang, Z., Rolia, J., Singhal, S., McKee, B., Hyser, C., Gmach, D., Gardner, R., Christian, T., Cherkasova, L.: 1000 Islands: an integrated approach to resource

management for virtualized data centers. Clus. Comput. **12**(1), 45–57 (2009). doi:10.1007/s10586-008-0067-6

**Brendan Jennings** is the Director of Research with the Telecommunications Software and Systems Group at Waterford Institute of Technology, Ireland. He was awarded B.Eng. and Ph.D. degrees by Dublin City University in 1993 and 2001, respectively. In 2012 he was a visiting researcher at the Royal Institute of Technology (KTH) in Stockholm, Sweden; currently, he is a visiting researcher at EMC$^2$ Research Europe in Cork, Ireland.

**Rolf Stadler** is a professor at the Royal Institute of Technology (KTH) in Stockholm, Sweden. He holds an M.Sc. degree in mathematics and a Ph.D. in computer science from the University of Zürich. Before joining the faculty at KTH in 2001, he had positions at the IBM Zürich Research Laboratory, Columbia University, and ETH Zürich.