Simulation Modelling Practice and Theory 39 (2013) 92–103

Contents lists available at SciVerse ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing

Georgia Sakellari [a,*], George Loukas [b]

[a] School of Architecture, Computing and Engineering, University of East London, United Kingdom
[b] School of Computing and Mathematical Sciences, University of Greenwich, United Kingdom

ARTICLE INFO

Article history:
Received 11 January 2013
Received in revised form 16 April 2013
Accepted 17 April 2013
Available online 15 May 2013

Keywords:
Cloud computing
Energy efficiency
Survey

ABSTRACT

The first hurdle for carrying out research on cloud computing is the development of a suitable research platform. While cloud computing is primarily commercially-driven and commercial clouds are naturally realistic as research platforms, they do not provide to the scientist enough control for dependable experiments. On the other hand, research carried out using simulation, mathematical modelling or small prototypes may not necessarily be applicable in real clouds of larger scale. Previous surveys on cloud performance and energy-efficiency have focused on the technical mechanisms proposed to address these issues. Researchers of various disciplines and expertise can use them to identify areas where they can contribute with innovative technical solutions. This paper is meant to be complementary to these surveys. By providing the landscape of research platforms for cloud systems, our aim is to help researchers identify a suitable approach for modelling, simulation or prototype implementation on which they can develop and evaluate their technical solutions.

1. Introduction

Excellent surveys of technical solutions towards better performance and energy-efficiency of cloud systems can be found in [1–3], and a review of relevant metrics and benchmarks has been provided in [4]. Our survey complements these works by presenting the mathematical models, software simulation approaches and experimental prototypes used in the literature as cloud computing research platforms. Our aim is to help researchers entering this field determine the research platform that they would need to develop or employ, so as to implement and evaluate their technical solutions.

Research on the energy efficiency and performance of cloud computing primarily focuses on the Information as a Service (IaaS) paradigm, where providers offer computing resources in the form of virtual or actual physical machines, and the interface to access them so that users can install their own operating system and software. Other common cloud computing approaches include Platform as a Service (PaaS), where users are provided with the low-level software and hardware to develop their own applications, and Software as a Service (SaaS), where the application is hosted and maintained by the SaaS provider but its users do not have access to the platform or the hardware infrastructure. Where appropriate, we are specifying what type of service a research platform has been designed for.

We begin in Section 2 with a description of the various mathematical approaches that have been used to model cloud systems. Most are formulations of cloud processes as optimisation problems that aim to analytically identify a cloud's configuration settings that would optimise its quality of service (QoS), performance or energy efficiency under given constraints.

* Corresponding author. Tel.: +44 2082237927.
E-mail addresses: g.sakellari@uel.ac.uk (G. Sakellari), g.loukas@gre.ac.uk (G. Loukas).

1569-190X/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.simpat.2013.04.002

Validation of these models can be done through simulation, possibly using one of the software packages reviewed in Section 3, or on an actual experimental testbed with physical and virtual machines. In Section 4.1.1 we review the general-purpose commercial cloud systems that have been used as research platforms in the literature, and in Section 4.1.2 the research testbeds that have been designed specifically for scientific research and development. Yet, possibly the majority of researchers tend to build their own, usually small-scale, cloud testbeds on the machines that they have available in their laboratories. For this reason, in Section 4.2 we review the relevant software frameworks for setting up and managing a private cloud. It is important to note that the cloud platforms surveyed here are not the only possible ones, and that our emphasis throughout the paper is on those that have been used widely for academic research on the QoS, performance or energy-efficiency of cloud computing.

2. Mathematical modelling of cloud systems

Apart from being less demanding on hardware and software investment, mathematical modelling and analysis may also be attractive for providing an understanding of the interdependencies involved in cloud computing. It is particularly suitable for identifying optimal values and equilibria and predicting behaviour.

Islam et al. [5] have developed an elasticity model for cloud instances. They have taken the assumption that each resource type (CPU, memory, network bandwidth, etc.) can be allocated in units and that the user is aware of the allocated resources and the relevant QoS metrics for their requests, as in the case of Amazon CloudWatch. The model combines the cost of provisioned but unutilised resources and the performance degradation cost due to underprovisioning. The consumer's detriment in overprovisioning is the difference between chargeable supply and demand, while the cost of underprovisioning is quantified through the percentage of rejected requests. They have also taken the assumption that the customers are able to convert the latter into the estimated financial impact to themselves. In order to extract a single elasticity metric they have proposed running different workloads on the cloud under investigation and taking the geometric mean of the combined costs described before. The associated experimental testbed is described in Section 4.1.

Abdelsalam et al. [6] have analysed the mathematical relationship between the service level agreements (SLAs) that govern the applications and the number of servers used to run them, as well as the frequencies the latter should run at to minimise power consumption. They have assumed that the cloud is homogeneous, that each machine can operate at discrete frequencies with different power consumption, and that the SLAs specify each client's needs by aggregating the processing needs of the applications to be run, the expected number of users and the average response time per user request. The problem of assigning a given number of jobs with different processing requirements to servers with limited capacity, such that the number of servers used is minimised, is NP-hard. The authors have simplified it by assuming that a job can be divisible over multiple servers. They have focused on applications that depend heavily on user interaction, such as web applications and web services. In the same area, Van et al. [7] have tackled the optimisation problems of virtual machine provisioning and placement. For the former, they have developed a global utility function representing power saving and SLA satisfaction, while they have approached the latter as a multiple knapsack problem with capacity constraints on the physical hosts. They have also described how their mathematical models can be used with a middleware framework managing the provisioning and placement of virtual machines on an actual cloud system.

The energy-information transmission tradeoff has been studied in [8] on a number of optimisation problems, such as minimising the energy and bandwidth cost or minimising the total carbon footprint subject to quality of service constraints. Based on their analytical solutions, one can determine whether to build a data centre at a given location, how many servers it should contain, how the service requests from users from different places would need to be routed towards each data centre, etc. In [9], the same research group have concentrated on workload distribution among geographically dispersed data centres to benefit from the location diversity of different types of available renewable energy resources. They have argued that by running data centres close to renewable energy sources could reduce not only their energy costs but also their Carbon Footprint. To address quality of service guarantees they have proposed real-time monitoring of queue lengths and effectively stabilising short queue lengths in each data centre.

Gelenbe et al. [10] have addressed the choice between a local or remote cloud service based on energy and quality of service criteria. They have formulated an optimisation problem for load sharing between a local and a remote cloud service as a multi-objective function formulated to optimise response time and energy consumption per job. They have created a composite cost function based on the Pollaczek–Khintchine formula, assuming Poisson arrivals, while parameters for service times and power consumption have been measured experimentally in a single server. The authors have argued that by adjusting the system load between local and remote cloud services they can achieve optimum tradeoffs between energy consumption and service times.

In [11] the energy consumption of cloud computing has been analysed in the cases of both public and private clouds from the aspects of switching and transmission, as well as data processing and data storage. The energy consumption was considered as an integrated supply chain logistics problem involving processing, storage, and transport. The authors claim that choosing a cloud can be more energy-efficient for users, especially when their computing tasks are of low intensity or infrequent.

Garg et al. [12] have modelled various energy characteristics, such as energy cost, carbon emission rate, workload and CPU power efficiency. Based on these models, they have proposed scheduling policies that reduce energy across a cloud

Table 1
Mathematical modelling of cloud systems.

Mathematical model	Energy efficiency	Performance/ QoS	Description
Islam et al. [5]		×	Models the elasticity characteristics of a cloud
Abdelsalam et al. [6]	×		Computes the optimal number of servers and the frequencies at which they should run to minimise energy
Van et al. [7]	×	×	Energy-based utility maximisation formulation for virtual machine provisioning and multiple knapsack for their placement under capacity constraints
Mohsenian-Rad et al. [8,9]	×	×	Energy-information transmission tradeoffs through various optimisation problems covering whether to build a datacentre at given location, etc.
Gelenbe et al. [10]	×	×	Formulates an optimisation problem for load sharing between a local and a remote cloud based on the tradeoffs between energy and QoS
Baliga et al. [11]	×		Analyses the energy a cloud consumes based on processing, storage and transport
Garg et al. [12]	×		Models various energy characteristics, such as energy cost, carbon emission rate, workload and CPU power efficiency, in relation to scheduling
Beloglazov and Buyya [13]	×		Derives CPU utilisation thresholds in a large-scale heterogeneous cloud, based on a statistical analysis of data collected for each virtual machine
Mi et al. [14]	×		Models virtual machine allocation as a multi-constraint optimisation problem of maximising CPU utilisation and minimising energy

infrastructure with multiple sites in various locations. All CPUs within a data centre have been considered to be homogeneous. They have assumed that the energy cost of the cooling system depends on a coefficient of performance (COP), which represents the efficiency of a cooling system as the ratio of the amount of energy consumed by CPUs to the energy consumed by the cooling system. Although COP varies with cooling air temperature, it is assumed to remain constant during a scheduling cycle. Also, the execution time of applications is assumed to be inversely proportional to the CPU operating frequency, which can be adjusted within a set of discrete values. Based on this assumption, they have derived a mathematical expression for the calculation of energy for different CPU frequencies. They have proved that there is always a frequency point that minimises the energy, and should be the frequency at which the CPU is to be adjusted for a specified set of parameters and subject to specified QoS constraints, such as deadlines. In [13], the same research group have modelled the power consumption of an IaaS cloud server as a linear function of the CPU utilisation and the performance degradation cost of migrating a virtual machine as a function of utilisation, memory and bandwidth. They have also defined a metric of the SLA violation to represent the percentage of the CPU performance demanded that has not been allocated. Based on the assumption that the host's utilisation approximately follows a normal distribution and can be modelled as a *t*-distribution, they have derived CPU utilisation thresholds based on a statistical analysis of data collected for each virtual machine.

Mi et al. [14] have formulated the multi-constraint optimisation problem of finding the optimal number of physical machines that maximises their resource utilisation while minimising the power consumption. They have made the assumption that CPU is the only resource of a physical machine. In order to predict the future load of the applications and the number of future requests, they have used Brown's quadratic exponential smoothing formula. To find an optimal reconfiguration policy they have proposed a self-configuration genetic algorithm, the fitness functions of which are composed of both a power consumption function and a punish function so that the CPU utilisation is kept between two threshold values.

Table 1 provides a list and brief description of the existing mathematical models of cloud systems, and identifies whether they are used for modelling energy efficiency or performance/QoS in a cloud environment.

The mathematical approaches presented above focus more on optimisation of the operational characteristics of computational resources and less on communications or complex user behaviour. For these, simulation may be a better approach. We detail the relevant software simulation packages in the next section.

3. Cloud simulation software

While grid computing simulators have existed for a while [15–17], they cannot sufficiently model the cloud infrastructure. Yet, there are still only a few options for simulating a cloud architecture, possibly because virtualisation has enabled the deployment of virtual private clouds on small scale physical testbeds (Section 4.2). Nevertheless, there have been some notable proposals for software simulation of clouds of very large scale. For example, the CloudSim simulation framework [18,19] has been shown to be able to instantiate 100,000 machines in less than 5 min, requiring only 75 MB of RAM. It is based on the SimJava [20] discrete event simulation engine at the lowest layer, while the higher layers implement the GridSim toolkit [15] for the modelling of the cluster, including networks, traffic profiles, resources, etc. CloudSim effectively extends the GridSim core functionalities by modelling storage, application services, resource provisioning between virtual machines, and data centre brokerage, and can even simulate federated clouds.

CloudSim has been modified and extended by several research groups. For example, it has been used as is for the evaluation of energy aware algorithms in [21], and its communication flow has been modified in [22] to evaluate a green scheduling algorithm that uses neural networks to predict the future load demand based on historical data of the demand. They have simulated 32 servers for a small-scale cloud and 500 for a medium-scale one. The loads generated were taken from real traces of historical requests to NASA and ClarkNet webservers. In [23], CloudSim has been enhanced in terms of its ability to represent the user's rather than the provider's perspective. The end result is CDOSim, a simulator that allows the integration of fine-grained models, for example for determining the best trade-off between costs and performance or for comparing run-time reconfiguration plans. The veracity of its enhancements have been confirmed against a 5-node experimental EC2 implementation. CloudSim has also been extended for education purposes in [24]. The end result is TeachCloud, which provides a simple graphical interface through which students can modify a cloud's configuration and perform simple experiments.

Cloudsim has recently also been extended by its own authors. In [25], they have introduced NetworkCloudSim where the focus is on the network flow model for data centres and the topologies, bandwidth sharing and latencies involved. Also, the modelling of applications has become more detailed by being able to represent multiple tasks for each application and tasks that are completed over multiple stages. As a result, it has been shown to be well suited to simulate advanced scheduling and resource allocation mechanisms.

Yet, being network flow-based rather than packet-based, CloudSim's network model cannot be as accurate as GreenCloud's [26], which has been designed on top of the network simulator ns-2 [27]. Also, unlike CloudSim, which is a generalist simulator, GreenCloud focuses specifically on the measurement of energy consumption [26]. The power models used to estimate the energy consumption assume proportionality of the power consumption to the CPU load in servers, and that the power consumption of switches is almost constant and proportional to the transmission rate only at a very small scale. It allows the configuration of different workload arrival rates and patterns, and can implement different power management techniques of putting components to sleep. Although it can support a relatively large number of servers, each server is considered to have a single core and there is no consideration of virtualisation, storage area networks and resource management. Thus, it is unclear whether it can be used to conduct experiments for evaluating the trade-off between performance and energy consumption [28] in a precise manner.

Another cloud simulator is MDCSim [29], which has been designed with an emphasis on multi-tier data centres. It can analyse a cluster-based data centre with detailed implementation of each individual tier. It has been configured into three layers, including a communication layer, a kernel layer and user-level layer, for modelling the different aspects of a cloud, and can estimate the throughput, response times, and power consumption. The latter is approximated using linear functions of the server utilisation, which in turn is calculated based on the number of nodes, number of requests and average execution time of requests.

More recently, Nunez et al. [30,31] have developed the iCanCloud simulator. It is based on SIMCAN [32], which is a software simulation framework for large storage networks. iCanCloud can predict the trade-off between costs and performance of a particular application in a specific hardware in order to inform the users about the costs involved. They have focused in particular on Amazon-like policies which charge users in a pay-as-you-go manner. iCanCloud has a full graphical user interface from which experiments can be designed and run, but existing software systems can only be modelled manually. It also allows parallel execution of one experiment over several machines.

Table 2 lists the most commonly used software for simulating cloud systems, and identifies whether they are used for investigating energy efficiency or performance/QoS in a cloud environment. It also specifies the programming language each simulator is written in, the software availability and the type of license. The only one of these that specifically supports parallel execution by design is iCanCloud.

Table 2
Cloud simulation software.

Simulation software	Energy efficiency	Performance/ QoS	Programming language	Currently available on the web	Open-source	Description
CloudSim [18,19]	×	×	Java	×	×	Modular and extensible open-source simulator, able to model very large scale clouds
Duy et al. [22]	×		Java			Modified CloudSim's [18] communication flow to evaluate neural a network-based scheduling method for reduced power consumption
CDOSim [23]		×	Java			Enhances CloudSim [18] by facilitating integration of more fine-grained models
TeachCloud [24]			Java	×	×	Extends CloudSim [18] with a simple interface for educational purposes
NetworkCloudSim [25]		×	Java			Extends CloudSim [18] with an emphasis on the modelling of the network and of more fine-grained applications
GreenCloud [26]	×		C++	×	×	Simulates a cloud as a packet network and estimates energy consumption at the servers, switches and links level
MDCSim [29]	×	×	Java			Simulates multi-tier data centres in detail
iCanCloud [30,31]		×	Java	×	×	Simulates a cloud and predicts the trade-offs between cost and performance of a set of applications executed on specific hardware

While simulation offers a number of advantages especially in terms of scalability and experiment repeatability, it is still based on assumptions and simplifications that might not fully represent an actual cloud. For this reason, it may be preferable to use real cloud testbeds (Section 4).

4. Cloud testbeds

To set up a cloud environment for experimental purposes, one needs to have access to a hardware infrastructure and a software framework to manage it. Researchers have used both general-purpose commercial cloud services and purpose-built scientific cloud testbeds. The former are often better for quick and small-scale experiments without significant prior investment, while the latter are typically preferred for longer term research and may be private or available to the general scientific community.

4.1. Cloud hardware setup

4.1.1. Commercial cloud services used in research

The commercial cloud service that has been investigated and used the most in academic research is Amazon Elastic Compute Cloud (EC2). It provides Infrastructure as a Service (IaaS) at a scale that can accommodate entire grids and parallel production infrastructures. It enables the users to increase or reduce the number of virtual machines needed and charges them according to the size of the instances and the capacity used. Being historically one of the most significant cloud services, a number of software frameworks for the development of private clouds have ensured to be compatible with EC2. Examples include Eucalyptus, OpenNebula, Nimbus and Xen Cloud Platform, which are discussed in Section 4.2.

Over the years, there have been a number of research publications using EC2 or evaluating it. Schad et al. [33] have experimentally measured its performance in terms of instance startup time, CPU performance, memory speed, hard disk performance and bandwidth of network traffic exchanged between instances. They have used one small and one large standard instance in different locations of the cloud (US and EU) and compared the usage of EC2 with their own local 10-computer physical cluster running a 50-node virtual cluster. Each physical computer ran the Linux Open Suse 11.1 operating system on a 2.66 GHz 64-bit Quad Core Xeon, six 750 GB SATA hard drives and three Gigabit network cards in bonding mode. They measured bandwidth with *iPerf*, CPU and memory speed with the Unix benchmark utility *ubench*, and disk performance with *bonnie++*. They used this implementation in order to demonstrate that the Amazon EC2 cloud might not be ideal as a scientific research platform for scientific research because its performance varies considerably in comparison to a local cluster environment like theirs. They have concluded that results based on Amazon EC2 might not be sufficiently repeatable and reproducible, which is undesirable for scientific measurements.

Ostermann et al. [34] have also evaluated how useful the Amazon EC2 can be for scientific experiments, and in particular for high performance distributed computing. They have conducted experiments on an EC2 environment with 1–128 cores running standard instances of different sizes on Fedora Core 6 virtual machines. They have studied the cases of single instances over a short period, multiple instances over a short period, and single instances over a long period of time and have compared their results with the HPCC single-job benchmarks. The performance metrics that were measured were the time it took an instance to release the resources back to the cloud (release time), the installation time, memory, I/O, reliability and the time taken to perform a variety of numerical operations. The findings of this work have suggested that performance and reliability of EC2 are low for high performance usage and should not be used except in cases where resources are needed quickly and only temporarily. The same conclusion was reached by Hill and Humphrey [35], who have compared memory performance of EC2 and their local two-node cluster running the Message Passing Interface system that is commonly used for parallel computing. While EC2 itself may not be ideal as a research platform, its architecture can be used as the modelling basis for more dependable simulations, such as the gang-scheduling models in [36,37].

Ostermann et al. have extended their work in [38], measuring the performance of further commercial cloud services, including GoGrid, ElasticHosts and Mosson, in particular regarding many-tasks. They have compared their findings with workload traces taken from parallel production infrastructures and grids. To conduct these experiments, they have extended their own large-scale distributed testing framework (GrenchMark [39]) to allow it to generate and submit both real and synthetic workloads to clouds environments and execute and analyse existing benchmarks. They have used the same experimental setup and performance metrics as before [34], plus the operational cost of the workload execution and the slowdown factor. Their findings have, once more, led them to the conclusion that the performance of all the cloud environments they have investigated is low for high performance usage and should only be used in cases where resources are needed instantly and temporarily.

In [5] the authors adopted a client–server e-commerce architecture. The local client consisted of the java workload generator JMeter [40], producing ten workload patterns of different burstiness. The application used as the benchmark was the TPC-W online bookshop implementation [41], run both locally and remotely on an EC2 cloud with instances that are regulated by a single load-balancer, dynamically altering the number of instances based on the workload. To monitor the performance of the system and measure the number of instances Amazon's CloudWatch tool has been used. The authors have proposed a single quantitative metric derived from all ten workload experiments which constitutes a measure of elasticity in clouds. Using this as a benchmark users could compare different cloud platforms and choose the one most appropriate to

Table 3
Commercial cloud testbeds used in research.

Commercial cloud	Service	Example research	Description
Amazon EC2	IaaS		Enables users to modify the number of virtual machines and charges based on capacity and instance size
		Schad et al. [33]	Evaluates EC2 for research use by comparing its performance variance to a local 50-node cluster
		Ostermann et al. [34]	Evaluates EC2 for research use by measuring its performance and reliability for high performance usage
		Hill and Humphrey [35]	Compares EC2 memory performance with a local 2-node cluster
		Iosup et al. [38]	Measures multi-task performance of EC2, GoGrid, ElasticHosts, and Mosson clouds as research platforms
		Chiu and Agrawal [44]	Evaluates Amazon Web Services in terms of cost-performance tradeoffs of caching and storage
		Islam et al. [5]	Evaluates EC2's elasticity of response time and cost structure
Amazon S3	Storage		Provides online storage through web service interfaces
		Wang et al. [42]	Uses S3 and EC2 to investigate performance fluctuations during data transfer
Google App Engine	PaaS		Enables users to run web applications. The data storage can grow according to the user's needs and traffic
		Iosup et al. [43]	Evaluates Amazon Web Services and Google App Engine performance based on real traces
		Bunch et al. [45]	Extends a Google App Engine API to act as a universal interface to disparate cloud databases
Google Apps	SaaS		Offers several web applications for word processing, email, calendar, etc.
		Alemany et al. [46]	Proposes an e-learning environment based on Google Apps
Windows Azure	PaaS/IaaS		Provides a platform for building, deploying and managing applications and virtual machines
		Lu et al. [47,48]	Proposes an add-on service for improving Azure's efficiency and evaluates it on a bioinformatics application

them. They have approached elasticity from the perspective of the time that it takes to make resources available, after a given request, and the unit of time used to charge the customer (per-hour being less elastic than per-minute).

The experiments in [42] have been conducted on the Amazon EC2 and Amazon S3 storage services. A S3 storage space was registered and several small EC2 instances were leased to act as the destination and as intermediate virtual machines. Files of different sizes were downloaded from S3 nodes to the EC2 cloud. To evaluate performance, the download time and the corresponding transfer rate were measured. The authors also implemented their own software-based middleware between S3 and EC2 clients to intercept data requests and reroute them by replicating the data over a number of virtual EC2 machines, so as to improve performance.

The performance of Amazon Web Services (AWS) and Google App Engine (GAE) has been evaluated in [43]. The authors have statistically analysed yearly traces they obtained from online sources, for different services and for various metrics, such as response time and latency during database queries and the time taken to calculate the 27th fibonacci number. AWS has also been evaluated in terms of the cost *Vs.* performance tradeoffs of caching and storage in [44], where application-dependent attributes were found to have far-reaching implications on the cost of sustaining their cache. GAE is a PaaS service for implementing Python and Java web applications with an emphasis on scalability. Developers upload their applications and make them available to users through GAE, which handles the administration, maintenance and hardware. Bunch et al. [45] have extended a GAE API to act as a universal interface to disparate cloud databases. They have concluded that cloud database technologies vary greatly and that those with more entry points for reading and writing data typically have superior performance. Less often, researchers have used Google Apps, a SaaS cloud service provided by Google. It includes tools for web-based email, storage, document management, etc. However, emphasis is usually on e-learning [46] rather than energy efficiency or performance.

Windows Azure is a PaaS service for running Windows applications and storing data in the cloud. It has been evaluated as a platform for large scale scientific experiments in [47], which has also proposed a parameter sweep add-on service to address Azure's weaknesses in terms of failures and load imbalances. In [48], this add-on service was evaluated on AzureBLAST, an Azure-based application that allows researchers to compare DNA or protein sequences against large databases of known sequences to find similar structures.

Table 3 lists existing commercially available clouds that have been used for scientific research purposes, together with a short description, the type of service they provide and examples of research papers that have used them.

4.1.2. Research cloud testbeds

A very large cloud testbed comprising of federated heterogeneous distributed data centres has been described in [49]. OpenCirrus is a joint initiative sponsored by Hewlett–Packard, Intel and Yahoo! in collaboration with more than eight other organisations and universities around the world such as the US National Science Foundation, the Russian Academy of Sciences, and Carnegie Mellon University. The testbed is composed of ten sites in North America, Europe and Asia and consists of several thousand cores and associated storage. It enables researchers with similar interests to exchange data sets and develop standard cloud computing benchmarks. To support users working with very large data sets the Hadoop [50] distributed file system is used to aggregate the storage from all the nodes of a domain. The management of the virtual machines can be done by different services, such as Eucalyptus (Section 4.2), as long as they are compatible with the EC2 interface. Finally, a monitoring service such as Ganglia [51] is used to monitor the clusters' health and collect operational data. Numerous projects are currently running in OpenCirrus involving a range of research areas, such as power-aware workload scheduling, improving cluster performance, data computation overlay for aggregation and analysis, astrophysics, graph data mining, and DNA sequencing [52].

Although OpenCirrus is possibly the largest cloud testbed available for research, it has not been designed to support computations that span more than one data centre. This is addressed by Open Cloud [53], a testbed with over 30 members including Cisco, NASA, and universities from the United States and Japan, running projects primarily in the areas of Big Data cloud computing middleware. Instead of the familiar commodity Internet, they are using wide area high performance networks to connect their four data centres, all of which are located in the United States.

Another cloud infrastructure available to the scientific community is Science Clouds [54] which consists of four sites in the United States. They were configured with the nimbus toolkit to enable remote leasing of resources via virtual machines in the same manner as EC2 leases. The sites communicate with a VPN-based virtualisation network. Users are able to access this network through a web-based social networking interface, which enables them to deploy a VPN by creating and managing a "social" user group. Science Clouds enables users to connect to EC2 clusters through an IaaS gateway, which effectively extends the pool of available resources.

Virtual Computing Lab (VCL) is a private cloud that provides free cloud use to academic institutes for a limited amount of time. It relies on a reservation system where users define their requirements and the amount of time they need the cloud resources for, before they are granted permission to use it. The resources can be used in a variety of ways, including using it as IaaS by loading their own virtual machine image, as PaaS by executing their code on an existing software platform, or as SaaS by using a single application on a cluster of computers. The cloud software framework used is xCAT (Section 4.2), while the reservation system and interface are based on the open-source Apache VCLweb-server software. VCL is currently used mainly for teaching and e-learning reasons, but also for research projects, an example of which is the elastic resource scaling system for multi-tenant cloud computing infrastructures proposed in [55].

More recent testbeds that can be used for large-scale cloud computing research include FutureGrid, a grid testbed distributed over six sites in the United States, and Grid'5000, distributed over nine sites in France. A recent application presented in [56] uses Grid'5000 as a testbed for cloud-based massive data analytics. A new testbed, Helix Nebula, is currently under development by the CERN, EMBL and ESA research centres to provide cloud resources to governments, businesses and citizens, geared towards big-science projects. In addition to these large-scale testbeds, smaller scale scientific clouds are beginning to appear around the world. An example is Okeanos [57], an IaaS cloud service powered by the open-source cloud software platform Synnefo and by Google Ganeti for the backend cluster management. In addition to the usual features for managing virtual machines, running tests and collecting statistics, Okeanos also has an internal firewall system. However, it is still in its alpha testing phase and limited to universities and research institutes located in Greece.

Of course, each research group may also decide to build their own private testbed to evaluate their energy efficiency or performance-enhancing technologies. For example, in [58], a testbed of four physical machines was used to measure power consumption against CPU and disk utilisation. They used the Xperf utility to monitor resource utilisation and the WattsUp Pro ES power meter to measure power consumption. Through their experiments they identified empirically optimal points for CPU, disk and energy values. Then they applied the same configuration to test a consolidation algorithm which allocates incoming workload based on the Euclidean distances that a workload allocation would have if assigned to a specific server with the optimal values that were empirically obtained. Van et al. [7] have built a testbed of three physical machines of four CPU cores each, running one virtual web machine per core or one batch virtual machine per two cores. They have generated a series of applications with different priorities to validate the resource arbitration of heterogeneous applications and show that the balancing of the quality of service and energy can be achieved by prioritising the applications. Heartfield and Loukas [59] have set up a virtual cloud on a single machine to investigate the effect of worm propagation through the Dropbox and

Table 4
Cloud testbeds used in research.

Scientific testbed	Description
OpenCirrus [49,52]	Large scale cloud research testbed distributed over 10 locations worldwide
Open Cloud Testbed [53]	Large scale cloud research testbed distributed over four US locations, supporting computations over more than one data centre
Science Clouds [54]	Cloud research testbed distributed over four US locations, able to connect to Amazon EC2
Virtual Computing Lab [55]	Private cloud providing free usage to academic institutes for limited amount of time
Future Grid	A grid testbed distributed over six sites in the United States
Grid'5000 [56]	Testbed for cloud-based massive data analytics
Okeanos	IaaS cloud still in its alpha testing phase and limited to universities and research institutes located in Greece
Other Individual implementations	Not publicly available small-scale clouds set up by individual research groups for their research. Examples include [58,7,59–61]

SugarSync cloud storage applications. They have demonstrated that social engineering attacks can be automated in such a cloud environment. Basmadjian et al. [60] have created an 8-node private cloud based on their own cloud controller, running on Red Hat Linux and a power and monitoring controller based on collected, an open-source Linux daemon to collect performance and power data. They have used this configuration to develop and evaluate an energy-aware plug-in, that runs on top of its automation and monitoring frameworks and triggers certain optimisation algorithms every time a new virtual machine is created or terminated.

While some research groups choose to develop their own software for developing and controlling their private clouds, others may use existing software frameworks, such as the ones presented in the section that follows. Table 4 lists the range of real testbeds used by the scientific community, with a short description for each.

4.2. Software frameworks for setting up private cloud testbeds

Currently one of the most popular cloud computing frameworks, Eucalyptus [62] implements infrastructure as a service (IaaS) enabling users to run and control virtual machine instances across a variety of physical resources. The framework consists of a node controller on each physical machine that hosts and controls the execution of its virtual machine instances, a cluster controller that manages the node controllers and schedules their instances, a storage controller for storing and accessing virtual machine images and user data, and the cloud controller as the entry point into the cloud for its users and administrators.Eucalyptus can run even as a standalone cloud on a single Linux machine that supports hardware virtualisation and has 100 GB disk space. This makes it particularly attractive to researchers with limited hardware resources. There is, however, a recommendation for a minimum of three 2.4 GHz multi-core machines. An example has been presented in [60], where a 3-node cluster was built to implement and test a distributed web interactive prototype system for sharing health records within a federated private cloud.

Huai et al. [64,65] have developed iVIC, a network software operating system that enables IaaS and SaaS on a pool of virtual machines. It delivers software on demand through presentation streaming mode to computers or mobile phones, and it has been used to minimise energy consumption in [66]. In their experiments, they have used an iVIC cloud pool of 60 Intel core2 duo servers running Linux 2.6.18, a Pentium-D PC as the global controller and a few desktop PCs as the clients. They tried different types of workloads, including an Apache Web Server and MySQL Server running for a long period of time, CPU demanding bioinformatic applications, and a C compiler that requires less computing power, all generated according to a Poisson distribution. In practice, they used this infrastructure to develop and test an energy saving job scheduler that determines where a virtual machine should be initialised or migrate in order to minimise the number of nodes that are on. To start, stop or migrate virtual machines they have used Xen's Python management API. A security enhancement has also been introduced in [67].

OpenNebula [68] is an open-source distributed virtual machine manager, which focuses on data centre virtualisation and enterprise private cloud computing, and has often been used in conjunction with OpenStack and Eucalyptus. OpenNebula requires only 10 MB for the front-end's base installation and a number of hosts connecting over a physical network. Its compartmentalised design makes it attractive for easy integration of custom algorithms on small-scale testbeds, such as the power-based custom scheduler introduced in [61] and tested on an OpenNebula pool of four servers.

The open-source platform with the most active community is OpenStack [69]. It has been developed as a linux-based collaborative project with three strands. OpenStack Compute deploys automatically the provisioned virtual compute instances, OpenStack Object Storage provides redundant storage of static objects, and OpenStack Image Service provides service discovery, registration and delivery for virtual disk images. While there is a recommendation for 32 GB RAM for each compute node, it is possible to build a test-environment one with only 2 GB. OpenStack is increasingly supported by commercial cloud providers, such as Rackspace, Canonical, Dreamhost and HP Cloud, and is used extensively as a research platform. For example, Corradi et al. [70] have used it to develop their virtual machine consolidation algorithms and test them in practice on their private cloud. They have developed their own cloud management platform on top of OpenStack to optimise consolidation along power consumption, host resources and networking. Wuhib et al. have focused on the use of management objectives for dynamic resource allocation on an OpenStack cloud [71] and have shown how load balancing, energy efficiency and service differentiation objectives can be mapped onto the resource allocation controllers. Beloglazov and Buyya have also worked on an extension for optimised consolidation, with their original design of the OpenStack Neat framework presented in [72].

Another open-source IaaS option is Nimbus, which has been designed specifically for scientific collaboration and is deployable on the Grid'5000 and FutureGrid large-scale testbeds. Its performance for scientific applications has been compared with Azure in [73] and was shown to incur less variability and to have better support for data intensive applications, while Azure would deploy faster and have a lower cost. Its configuration toolkit has been extended in [74] to deploy backfill virtual machines on idle cloud nodes for high throughput computing in an opportunistic manner.

Abiquo, formerly known as AbiCloud, is a platform manager for private clouds inside an organisation's firewall, through a user interface. It supports Linux, Windows and Mac operating systems, and has both an open-source and an enterprise version, the latter enabling users to manage multiple datacentres and link multiple clouds together. For testing, Abiquo requires only two 32-bit 1.6 GHz machines with 1 GB RAM and 100 GB hard disk space each, but the requirements increase significantly for production development. Abiquo's open-source version includes a best-fit scheduler to select the machine with the highest number of unused CPU cores [75].

Table 5
Software frameworks for cloud testbeds.

Software framework	Open-source	Proprietary	Service	Description
Eucalyptus [62]	×		IaaS	Generalist open-source cloud computing framework running primarily on Ubuntu and CentOS
iVIC [64–67]	×		IaaS/SaaS	Cloud-based virtual computing environment with emphasis on energy efficiency and security
OpenNebula [68]	×		IaaS	Distributed virtual machine manager with a compartmentalised design for easy integration of custom algorithms
OpenStack [69–71]	×		IaaS	Open-source platform with the most active community and substantial industrial approval
OpenStack Neat [72]	×		IaaS	Framework for optimised virtual machine consolidation in OpenStack [69]
Nimbus [73,74]	×		IaaS	Designed specifically for scientific collaboration and is deployable on the Grid'5000 and FutureGrid testbeds
Abiquo [75]	×	×	IaaS	Platform manager for private clouds. Its open-source version has limited functionality
xCAT [76]	×		IaaS	Toolkit for management and provisioning of distributed computing resources
XCP [77]	×		IaaS	Tool for automatic configuration and maintenance of a cloud platform
Entropy [7]	×		IaaS	Manages the placement of virtual machines based on a constraint programming module
BtrCloud [78,79]		×	IaaS	Extends Entropy [7] with a scripting language for administrators, a graphical interface, etc.
Tplatform [84]	×		PaaS	Designed primarily for web mining applications
ECP [84]		×	IaaS	The free Spotcloud version is limited to small-scale clouds
mOSAIC [86]	×		PaaS/IaaS	Provides interoperability between different clouds and has been extended in terms of resilience to denial of service
OpenQRM [80,81]	×		IaaS	Platform for automation and management of scalable clusters that can be employed in federated cloud environments
WSO2 Stratos [82,85,83]	×		PaaS	Java PaaS platform with an emphasis on the number of available core features, including security and storage

The Extreme Cloud Administration Toolkit (xCAT) is an open-source toolkit for management and provisioning of distributed computing resources. It has been used in [76] as the underlying cloud infrastructure to replicate the operation of botnets for large-scale malware propagation scenarios.

Xen Cloud Platform (XCP) is a cloud infrastructure virtualisation solution that does not provide the overall architecture for cloud services, but a tool for automatic configuration and maintenance of the platform. It has been used in [77] to provide the cloud infrastructure for large scale data processing.

Another open-source virtual machine manager is Entropy [7]. It reconfigures the state and the placement of a client's virtual machines according to high-level constraints expressed on demand by the system administrators and the clients. This work has been recently extended in BtrCloud [78]. The latter includes BtrScript, a scripting language that helps administrators manipulate large sets of elements and set high-level goals through policies, such as a load balancing across CPUs or minimisation of the number of servers needed to operate. As in Entropy, the placement of virtual machines is determined dynamically according to a constraint programming module [79]. There is also a monitoring module and a graphical interface.

Open Qlusters Resource Manager (Open QRM) is an open-source data centre platform for automation and management of scalable clusters, and it can be employed in a federated cloud environment over a wide area network [80]. For this purpose, it has been extended in [81], which has proposed an image cloning methodology for reducing bandwidth and cloud resources when migrating from one cloud infrastructure to another.

WSO2 Stratos is an open-source Java PaaS platform with an emphasis on the number of available core features [82]. Ardagna et al. [83] have used WSO2 to evaluate techniques for automatic scalability of a PaaS cloud with changing environments and loads.

Other frameworks that have been proposed in the literature but have not been used widely for research purposes yet include Tplatform and Enomaly ECP Spotcloud [84]. The former is a PaaS framework designed primarily for web mining applications, while the latter is an IaaS framework that focuses on small scale. A more recent one is mOSAIC, which provides the necessary tools for achieving interoperability between different clouds. Its aim is to allow the development of applications that can run on multiple clouds. Its ontology has been analysed in [85] and it has been extended in [86] with features that improve its resilience against specific types of denial of service attacks.

We do not by any means claim that this is a comprehensive list of open-source software frameworks. Here, we have attempted to briefly describe the ones that have been used in research broadly related to cloud performance, QoS and energy efficiency. Table 5 provides a summary of these options, specifying the type of service they provide, a short description and information on whether they are open-source or proprietary.

5. Conclusions

For a researcher, the choice of testbed may be seen as a tradeoff between realism and scientific practicality. Commercial testbeds are naturally realistic as they are the ones used by the general public in practice, but they cannot provide to the scientist dependable experiments and enough control. Unsurprisingly research on energy efficiency has been limited to private rather than publicly available cloud testbeds as the users need to be able to measure the power consumption of the infrastructure involved.

The scale and topology of a cloud infrastructure affect considerably its networking element, which in turn affects the QoS provided to the user. This possibly renders small-scale testbeds unsuitable for QoS-related research. Yet, despite the increasing use of multi-site clouds that operate on complex topologies, there seems to be a general lack of detail in terms of the networking aspects. For example, most mathematical models address QoS as high-level optimisation problems, making use of the body of knowledge in areas such as operations research but usually not taking into account queuing theory (an exception being [10]) and other fundamental areas of communications. Also, it is worth noting that several cloud testbeds have been developed from previous grid and high performance computing testbeds, and as a result research is currently possibly biased towards the high performance aspect rather than perhaps the human–cloud interaction, where the body of knowledge in human–computer interaction could be applicable.

As is often the case in research, we believe that the usefulness of simulation and experimental research platforms depends heavily on the existence of realistic user traffic traces and workloads. The systematic extraction of such traces would also help immensely in setting realistic assumptions for mathematical modelling. Perhaps an approach that would provide satisfactory realism and scientific usability would be for a large-scale scientific testbed to be available for normal, every-day use in the same way a commercial one is. Of course, this would be with the understanding that experimental technologies may be tried on the real systems in real time and data may be collected for scientific purposes.

Research in cloud computing may be seen as particularly demanding with regards to infrastructure and equipment, possibly being a barrier for entry for sections of the scientific community. It is not possible to provide precise guidelines regarding the optimal platform for each type of research and it would be unwise to do so based on a platform's popularity for a given type or research. For example, Amazon EC2 has been used extensively for research in energy efficiency, but there is significant evidence that it cannot produce sufficiently repeatable and reproducible scientific results [33]. By reviewing the mathematical models, simulation software and cloud testbeds used in academic research, we have aimed to provide the landscape of available options for new researchers entering the field, especially from scientific areas that have not been utilised sufficiently in cloud computing.

References

[1] B. Rimal, E. Choi, I. Lumb, A taxonomy and survey of cloud computing systems, in: Proceedings of the 5th International Joint Conference on INC, IMS and IDC (NCM'09), IEEE, 2009, pp. 44–51.
[2] Q. Zhang, L. Cheng, R. Boutaba, Cloud computing: state-of-the-art and research challenges, Journal of Internet Services and Applications 1 (1) (2010) 7–18.
[3] A. Beloglazov, R. Buyya, Y. Lee, A. Zomaya, et al, A taxonomy and survey of energy-efficient data centers and cloud computing systems, Advances in Computers 82 (2) (2011) 47–111.
[4] Z. Li, L. O'Brien, H. Zhang, R. Cai, On a catalogue of metrics for evaluating commercial cloud services, in: Proceedings of the ACM/IEEE 13th International Conference on Grid Computing (GRID), 2012, pp. 164 –173..
[5] S. Islam, K. Lee, A. Fekete, A. Liu, How a consumer can measure elasticity for cloud platforms, in: Proceedings of the 3rd Joint WOSP/SIPEW International Conference on Performance Engineering, ICPE '12, Boston, Massachusetts, USA, 2012, pp. 85–96.
[6] H. Abdelsalam, K. Maly, D. Kaminsky, Analysis of energy efficiency in clouds, in: Proceedings of Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, IEEE, Athens, Greece, 2009, pp. 416–421.
[7] H. Van, F. Tran, J.-M. Menaud, Performance and power management for cloud infrastructures, in: Proceedings of the 3rd International Conference on Cloud Computing, IEEE, Miami, FL, USA, 2010, pp. 329–336.
[8] A.-H. Mohsenian-Rad, A. Leon-Garcia, Energy-information transmission tradeoff in green cloud computing, in: Proceedings of Globecom, IEEE, Miami, FL, USA, 2010.
[9] M. Ghamkhari, H. Mohsenian-Rad, Optimal integration of renewable energy resources in data centers with behind-the-meter renewable generator, in: Proceedings of the International Conference in Communications (ICC'2012), IEEE, Ottawa, Canada, 2012.
[10] E. Gelenbe, R. Lent, M. Douratsos, Choosing a local or remote cloud, in: Proceedings of the 2nd Symposium on Network Cloud Computing and Applications, London, UK, 2012.
[11] J. Baliga, R. Ayre, K. Hinton, R. Tucker, Green cloud computing: balancing energy in processing, storage, and transport, Proceedings of the IEEE 99 (1) (2011) 149–167.
[12] S. Garg, C. Yeo, A. Anandasivam, R. Buyya, Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers, Journal of Parallel and Distributed Computing 71 (6) (2011) 732–749.
[13] A. Beloglazov, R. Buyya, Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers, in: Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science, MGC '10, Bangalore, India, 2010, pp. 4:1–4:6.
[14] H. Mi, H. Wang, G. Yin, Y. Zhou, D. Shi, L. Yuan, Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers, in: Proceedings of the 7th International Conference on Services Computing (SCC), Miamy, FL, USL, 2010, pp. 514 –521.
[15] R. Buyya, M. Murshed, Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing, Concurrency and Computation: Practice and Experience 14 (13–15) (2003) 1175–1220.
[16] A. Legrand, L. Marchal, H. Casanova, Scheduling distributed applications: the simgrid simulation framework, in: Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2003), 2003, pp. 138–145.
[17] C. Dumitrescu, I. Foster, Gangsim: a simulator for grid scheduling studies, in: Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005), vol. 2, 2005, pp. 1151–1158.
[18] R. Calheiros, R. Ranjan, C. De Rose, R. Buyya, Cloudsim: a novel framework for modeling and simulation of cloud computing infrastructures and services. arXiv:0903.2525.

[19] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, R. Buyya, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software: Practice and Experience 41 (1) (2011) 23–50.
[20] F. Howell, R. McNab, SimJava: a discrete event simulation library for java, Simulation Series 30 (1998) 51–56.
[21] Y. Shi, X. Jiang, K. Ye, An energy-efficient scheme for cloud resource provisioning based on cloudsim, in: Proceedings of the Annual International Conference on Cluster Computing (CLUSTER), Austin, TX, USA, 2011, pp. 595–599.
[22] T. Duy, Y. Sato, Y. Inoguchi, Performance evaluation of a green scheduling algorithm for energy savings in cloud computing, in: Proceedings of International Symposium on Parallel and Distributed Processing, Workshops and PhD Forum (IPDPSW), IEEE, Atlanta, GA, USA, 2010.
[23] F. Fittkau, S. Frey, W. Hasselbring, Cloud user-centric enhancements of the simulator cloudsim to improve cloud deployment option analysis, in: Proceedings of the 1st European conference on Service-Oriented and Cloud Computing, ESOCC'12, 2012.
[24] Y. Jararweh, Z. Alshara, M. Jarrah, M. Kharbutli, M. Alsaleh, Teachcloud: a cloud computing educational toolkit, in: Proceedings of the 1st International IBM Cloud Academy Conference (ICA CON 2012), IBM, Research Triangle Park, NC, USA, 2012.
[25] S. Garg, R. Buyya, Networkcloudsim: modelling parallel applications in cloud simulations, in: Proceedings of the 4th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2011), Melbourne, Australia, 2011, pp. 105–113.
[26] D. Kliazovich, P. Bouvry, Y. Audzevich, S. Khan, Greencloud: a packet-level simulator of energy-aware cloud computing data centers, in: Proceedings of the Global Telecommunications Conference (GLOBECOM 2010), IEEE, 2010, pp. 1–5.
[27] S. McCanne, S. Floyd, Network Simulator ns-2, 1997. <http://www.isi.edu/nsnam/ns/>.
[28] G. Sakellari, C. Morfopoulou, E. Gelenbe, Investigating the Tradeoffs between Power Consumption and Quality of Service in a Backbone Network, Future Internet (2013), MDPI.
[29] S. Lim, B. Sharma, G. Nam, E. Kim, C. Das, Mdcsim: a multi-tier data center simulation, platform, in: Proceedings of IEEE International Conference on Cluster Computing and Workshops, IEEE, New Orleans, Louisiana, USA, 2009.
[30] A. Nunez, J.L. Vazquez-Poletti, A.C. Caminero, G.G. Castane, J. Carretero, I.M. Llorente, iCanCloud: a flexible and scalable cloud infrastructure simulator, Journal of Grid Computing 10 (1) (2012) 185–209.
[31] A. Nunez, J.L. Vazquez-Poletti, A.C. Caminero, J. Carretero, I.M. Llorente, Design of a new cloud computing simulation platform, in: Proceedings of the International conference on Computational science and its applications, ICCSA'11, Santander, Spain, 2011, pp. 582–593.
[32] A. Nunez, J. Fernandez, J.D. Garcia, L. Prada, J. Carretero, Simcan: a simulator framework for computer architectures and storage networks, in: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Simutools '08, Marseille, France, 2008, pp. 73:1–73:8.
[33] J. Schad, J. Dittrich, J.-A. Quiane-Ruiz, Runtime measurements in the cloud: observing, analyzing, and reducing variance, Proceedings of the VLDB Endowment 3 (1–2) (2010) 460–471.
[34] S. Ostermann, R. Iosup, N. Yigitbasi, T. Fahringer, A performance analysis of EC2 cloud computing services for scientific computing, in: Proceedings of the ICST International Conference on Cloud Computing, 2009.
[35] Z. Hill, M. Humphrey, A quantitative analysis of high performance computing with Amazon's EC2 infrastructure: the death of the local cluster?, in: Proceedings of the 10th IEEE/ACM International Conference on Grid Computing, 2009, pp. 26–33.
[36] I.A. Moschakis, H.D. Karatza, Performance and cost evaluation of Gang Scheduling in a Cloud Computing system with job migrations and starvation handling, in: Proceedings of IEEE Symposium on Computers and Communications (ISCC), 2011, pp. 418–423.
[37] I.A. Moschakis H.D. Karatza, Evaluation of gang scheduling performance and cost in a cloud computing system, The Journal of Supercomputing 59 (2) (2012) 975–992.
[38] A. Iosup, S. Ostermann, M. Yigitbasi, R. Prodan, T. Fahringer, D. Epema, Performance analysis of cloud computing services for many-tasks scientific computing, IEEE Transactions on Parallel and Distributed Systems 22 (6) (2011) 931–945.
[39] A. Iosup, D. Epema, Grenchmark: a framework for analyzing, testing, and comparing grids, in: Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid (CCGRID 06), vol. 1, 2006, pp. 313–320.
[40] E. Halili, Apache JMeter: A Practical Beginner's Guide to Automated Testing and Performance Measurement for Your Websites, Packt Pub Limited, 2008.
[41] D. Menasce, Tpc-w: a benchmark for e-commerce, Internet Computing, IEEE 6 (3) (2002) 83–87.
[42] J. Wang, P. Varman, C. Xie, Avoiding performance fluctuation in cloud storage, in: International Conference on High Performance Computing (HiPC), Goa, India, 2010, pp. 1–9.
[43] A. Iosup, N. Yigitbasi, D. Epema, On the performance variability of production cloud services, in: Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2011), 2011, pp. 104 –113.
[44] D. Chiu, G. Agrawal, Evaluating caching and storage options on the amazon web services cloud, in: Proceedings of the 11th IEEE/ACM International Conference on Grid Computing, IEEE, Brussels, Belgium, 2010.
[45] C. Bunch, N. Chohan, C. Krintz, J. Chohan, Y. Nomura, J. Kupferman, P. Lakhina, Y. Li, An evaluation of distributed datastores using the AppScale cloud platform, in: Proceedings of the 3rd International Conference on Cloud Computing, IEEE, Miami, FL, USA, 2010, pp. 305–312.
[46] J. Alemany, X. Perramon, L. Panadès, Improving the quality of the practicum. the use of moodle and google docs in monitoring the practicum process after the ehea, CIDUI-Llibre d'actes 1 (1).
[47] W. Lu, J. Jackson, J. Ekanayake, R. Barga, N. Araujo, Azureblast: a case study of cloud computing for science applications, in: Proceedings of the 2nd International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, Indianapolis, IN, USA, 2010, pp. 209–217.
[48] W. Lu, J. Jackson, R. Barga, Azureblast: A case study of cloud computing for science applications, in: Proceedings of the 1st Workshop on Scientific Cloud Computing, Chicago, IL, USA, 2010.
[49] A. Avetisyan, R. Campbell, I. Gupta, M. Heath, S. Ko, G. Ganger, M. Kozuch, D. O'Hallaron, M. Kunze, T. Kwan, K. Lai, M. Lyons, D. Milojicic, H.Y. Lee, Y.C. Soh, N.K. Ming, J.-Y. Luke, H. Namgoong, Open cirrus: a global cloud computing testbed, Computer 43 (4) (2010) 35–43.
[50] T. White, Hadoop: The Definitive Guide, O'Reilly Media, 2012.
[51] M. Massie, B. Chun, D. Culler, The ganglia distributed monitoring system: design, implementation, and experience, Parallel Computing 30 (7) (2004) 817–840.
[52] C. Baun, M. Kunze, Performance measurement of a private cloud in the opencirrus testbed, in: Proceedings of the International Conference on Parallel Processing, Euro-Par'09, Springer-Verlag, Delft, The Netherlands, 2010, pp. 434–443.
[53] R. Grossman, Y. Gu, M. Sabala, C. Bennet, J. Seidman, J. Mambratti, The open cloud testbed: a wide area testbed for cloud computing utilizing high performance network services, in: Proceedings of GridNets, Athens, Greece, 2009..
[54] K. Keahey, T. Freeman, Science clouds: early experiences in cloud computing for scientific applications, in: Proceedings of Cloud Computing and Its Applications (CCA-08), Chicago, IL, USA, 2008.
[55] Z. Shen, S. Subbiah, X. Gu, J. Wilkes, Cloudscale: elastic resource scaling for multi-tenant cloud systems, in: Proceedings of the 2nd ACM Symposium on Cloud Computing, SOCC '11, ACM, Cascais, Portugal, 2011, pp. 5:1–5:14.
[56] R. Moussa, Massive data analytics in the cloud: TPC-H experience on hadoop clusters, IJWA 4 (3) (2012) 113–133.
[57] E. Koukis, P. Louridas, Okeanos IaaS, in: Proceedings of the EGI Community Forum/EMI Second Technical Conference (EGICF12-EMITC2), Munich, Germany, 2012.
[58] S. Srikantaiah, A. Kansal, F. Zhao, Energy aware consolidation for cloud computing, in: Proceedings of the Conference on Power Aware Computing and Systems, USENIX Association, 2008, p. 10.
[59] R. Heartfield, G. Loukas, On the feasibility of automated semantic attacks in the cloud, in: Proceedings of the 27th International Symposium on Computer and Information Sciences, Springer, Paris, France, 2012, pp. 343–351.
[60] R. Basmadjian, H. De Meer, R. Lent, G. Giuliani, Cloud computing and its interest in saving energy: the use case of a private cloud, Journal of Cloud Computing 1 (5) (2012).

[61] A. Younge, G. von Laszewski, L. Wang, S. Lopez-Alarcon, W. Carithers, Efficient resource management for cloud computing environments, in: Proceedings of the International Green Computing Conference, Chicago, IL, USA, 2010, pp. 357–364.
[62] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, D. Zagorodnov, The eucalyptus open-source cloud-computing system, in: Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID '09, 2009, pp. 124–131.
[64] J. Huai, Q. Li, C. Hu, Civic: a hypervisor based virtual computing environment, in: Proceedings of the International Conference on Parallel Processing Workshops (ICPPW 2007), 2007, p. 51..
[65] Y. Chen, T. Wo, J. Li, An efficient resource management system for on-line virtual cluster provision, in: Proceedings of the International Conference on Cloud Computing (CLOUD '09), 2009, pp. 72–79.
[66] B. Li, J. Li, J. Huai, T. Wo, Q. Li, L. Zhong, Enacloud: an energy-saving application live placement approach for cloud computing environments, in: Proceedings of the International Conference on Cloud Computing (CLOUD'09), IEEE, 2009, pp. 17–24.
[67] J. Li, B. Li, T. Wo, C. Hu, J. Huai, L. Liu, K. Lam, Cyberguarder: a virtualization security assurance architecture for green cloud computing, Future Generation Computer Systems 28 (2) (2012) 379–390.
[68] J. Fontan, T. Vazquez, L. Gonzalez, R. Montero, I. Llorente, Opennebula: The open source virtual machine manager for cluster computing, in: Open Source Grid and Cluster Software Conference, San Francisco, CA, USA, 2008.
[69] X. Wen, G. Gu, Q. Li, Y. Gao, X. Zhang, Comparison of open-source cloud management platforms: openstack and opennebula, in: Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012), IEEE, Chongqing, China, 2012, pp. 2457–2461.
[70] A. Corradi, M. Fanelli, L. Foschini, Vm consolidation: a real case based on openstack cloud, Future Generation Computer Systems, 2012.
[71] F. Wuhib, R. Stadler, H. Lindgren, Dynamic resource allocation with management objectives – implementation for an openstack cloud, in: Proceedings of the 8th International conference on Network and Service Management (CNSM), 2012.
[72] A. Beloglazov, R. Buyya, Openstack Neat: A Framework for Dynamic Consolidation of Virtual Machines in Openstack Clouds – A Blueprint, Tech. Rep. CLOUDS-TR-2012-4, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, August 2012.
[73] R. Tudoran, A. Costan, G. Antoniu, L. Bougé, A performance evaluation of azure and nimbus clouds for scientific applications, in: Proceedings of the 2nd International Workshop on Cloud Computing Platforms, ACM, 2012, p. 4.
[74] A. Krishna, C. Rajendra, Improvising the infrastructure as a service cloud, International Journal of Advanced Research in Computer Engineering and Technology (IJARCET) 1 (5) (2012) 367–370.
[75] S. Jagannathan, Comparison and Evaluation of Open-Source Cloud Management Software, Ph.D. Thesis, KTH, 2012.
[76] J. Calvet, J. Fernandez, P. Bureau, J. Marion, et al., Large-scale malware experiments: why, how, and so what?, in: Proceedings of the Virus Bulletin International Conference, Vancouver, Canada, 2010.
[77] Y. Tabaa, A. Medouri, M. Tetouan, Towards a next generation of scientific computing in the cloud, International Journal of Computer Science Issues (2012).
[78] R. Pottier, J.-M. Menaud, Btrscript: a safe management system for virtualized data center, in: Proceedings of the 8th International Conference on Autonomic and Autonomous Systems (ICAS 2012), St. Maarten, The Netherlands Antilles, 2012, pp. 49–56.
[79] F. Hermenier, A. Lebre, J.-M. Menaud, Cluster-wide context switch of virtualized jobs, in: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, ACM, New York, NY, USA, 2010, pp. 658–666.
[80] A. Celesti, F. Tusa, M. Villari, A. Puliafito, How to enhance cloud architectures to enable cross-federation, in: Proceedings of the 3rd International Conference on Cloud Computing, IEEE, Miami, FL, USA, 2010, pp. 337–345.
[81] A. Celesti, F. Tusa, M. Villari, A. Puliafito, Improving virtual machine migration in federated cloud environments, in: Proceedings of the 2nd International Conference on Evolving Internet (INTERNET), Valencia, Spain, 2010, pp. 61–67.
[82] A. Azeez, S. Perera, S. Weerawarana, P. Fremantle, S. Uthaiyashankar, S. Abesinghe, Wso2 stratos: an application stack to support cloud computing, Information Technology 53 (4) (2011) 180–187.
[83] C. Ardagna, E. Damiani, F. Frati, D. Rebeccani, M. Ughetti, Scalability patterns for platform-as-a-service, in: Proceedings of the 5th International Conference on Cloud Computing, Honolulu, Hawaii, USA, 2012, pp. 718–725.
[84] P. Endo, G. Gonçalves, J. Kelner, D. Sadok, A survey on open-source cloud computing solutions, in: Proceedings of the 8th Workshop on Clouds, Grids and Applications (WCGA), Gramado, Brazil, 2010, pp. 3–16.
[85] F. Moscato, R. Aversa, B. Di Martino, T. Fortis, V. Munteanu, An analysis of mosaic ontology for cloud resources annotation, in: Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), IEEE, Szczecin, Poland, 2011, pp. 973–980.
[86] M. Ficco, M. Rak, Intrusion tolerance in cloud applications: the mosaic approach, in: Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), Palermo, Italy, 2012, pp. 170–176.