Computer Networks 57 (2013) 2093–2115

Contents lists available at SciVerse ScienceDirect

Computer Networks

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

Survey Paper

Cloud monitoring: A survey

Giuseppe Aceto, Alessio Botta, Walter de Donato, Antonio Pescapè *

University of Napoli Federico II, Napoli, Italy

ARTICLE INFO

Article history:
Received 25 October 2012
Received in revised form 15 February 2013
Accepted 3 April 2013
Available online 10 April 2013

Keywords:
Cloud monitoring
Cloud measurements
SLA monitoring
Cloud resource monitoring
Cloud monitoring metrics
Cloud monitoring platforms

ABSTRACT

Nowadays, Cloud Computing is widely used to deliver services over the Internet for both technical and economical reasons. The number of Cloud-based services has increased rapidly and strongly in the last years, and so is increased the complexity of the infrastructures behind these services. To properly operate and manage such complex infrastructures effective and efficient monitoring is constantly needed.

Many works in literature have surveyed Cloud properties, features, underlying technologies (e.g. virtualization), security and privacy. However, to the best of our knowledge, these surveys lack a detailed analysis of monitoring for the Cloud. To fill this gap, in this paper we provide a survey on Cloud monitoring. We start analyzing motivations for Cloud monitoring, providing also definitions and background for the following contributions. Then, we carefully analyze and discuss the properties of a monitoring system for the Cloud, the issues arising from such properties and how such issues have been tackled in literature. We also describe current platforms, both commercial and open source, and services for Cloud monitoring, underlining how they relate with the properties and issues identified before. Finally, we identify open issues, main challenges and future directions in the field of Cloud monitoring.[1]

1. Introduction

Cloud Computing [1] has rapidly become a widely adopted paradigm for delivering services over the Internet. This is due to a number of technical reasons, including: improvement of energy efficiency, optimization of hardware and software resources utilization, elasticity, performance isolation, flexibility, and on-demand service schema [2]. In addition to such technical benefits, the literature has shown how the Cloud Computing model provides several economical benefits including minimal capital and operational expenditures (CAPEX and OPEX). For all these reasons, the number of organizations adopting Cloud solutions and subscribers accessing Cloud services has rapidly increased, exceeding the optimistic initial plans, and so has done the complexity of Cloud systems. Cloud services are on-demand, elastic and scalable, and the following main features are therefore needed by a Cloud system: availability, concurrency, dynamic load balancing, independence of running applications, security, and intensiveness (as defined and analyzed in [3]). To provide these features, advanced virtualization techniques, robust and dynamic scheduling approaches, advanced security measures and disaster recovery mechanisms are implemented and operated in Cloud Computing systems. Data centers for Cloud Computing continue to grow in terms of both hardware resources and traffic volume, thus making Cloud operation and management more and more complex [149].

In this scenario, accurate and fine-grained monitoring activities are required to efficiently operate these platforms and to manage their increasing complexity.

In literature, there is a large number of works proposing surveys and taxonomies of Cloud Computing in general

* Corresponding author. Tel.: +39 0817683856; fax: +39 0817683816.
E-mail addresses: giuseppe.aceto@unina.it (G. Aceto), a.botta@unina.it (A. Botta), walter.dedonato@unina.it (W. de Donato), pescape@unina.it (A. Pescapè).

[1] Preliminary results within the same framework have been published in *G. Aceto, A. Botta, W. de Donato, A. Pescapè, "Cloud monitoring: definitions, issues and future directions", 1st IEEE International Conference on Cloud Networking (IEEE CloudNet'12)", Paris (France), November 28–30, 2012.*

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.04.001

[4–10], of Virtualization technologies [11,12], and of Cloud Security [13–19]. To the best of our knowledge, however, there are no specific surveys on platforms, techniques, and tools for monitoring Cloud infrastructures, services, and applications. This is what we define as *Cloud monitoring*.

In this paper, we provide a survey of Cloud monitoring, analyzing the articulate state of the art in this field. According to the indications reported in [126], we adopt the research methodology depicted in Fig. 1, which is described in the following.

- We select a well-known taxonomy of the terms and roles in the field of Cloud Computing for the contextualization of the contributions we provide in this paper. To this aim we use the work of the National Institute of Standards and Technology (NIST) [1,20].
- After analyzing the literature in the field of Cloud Computing, using the definitions proposed by NIST, we provide a two-axis taxonomy for Cloud monitoring:
  - one axis is for the several motivations for monitoring Cloud Computing (Section 3);
  - the other axis is further expanded along three dimensions: layers; abstractions level; tests and metrics (Section 4).
- Thanks to the results of the previous step, we analyze several research works for deriving the main properties of systems for Cloud monitoring, the issues associated with such properties, and the contributions in literature regarding these properties and issues (Section 5). Moreover, we analyze a number of commercial and open source platforms and a number of services for Cloud monitoring, evidencing also their relation with the properties and issues discussed before (Section 6).
- The previous steps provide us the inputs to derive the open issues and the future directions in the field of Cloud monitoring (Section 7).

We believe that this paper provides contributions of interest for the research community, analyzing the literature and shedding light on the current and future research issues on Cloud monitoring.

2. Cloud Computing: a brief overview

According to the NIST, Cloud Computing is defined as follows [1]:

> "*Model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*"

The NIST and Cloud community have also defined the following important concepts: (i) *Essential characteristics*, (ii) *service models*, (iii) *hosting*, (iv) *deployment models*, and (v) *roles* [20]. In Table 1 we list these concepts because they are useful and required for the topics discussed in this paper. Considering the wide spread of these concepts in the literature related to Cloud Computing, a deep and detailed discussion about them is out of the scope of this paper. We refer the reader to other works [1,4–10,21] to deepen these definitions and terms. For the sake of brevity in the following we refer to a Cloud Service Provider as "Provider" and to a Cloud Service Consumer as "Consumer", whenever the specific kind of service involved is nonessential or the context is clear about it.

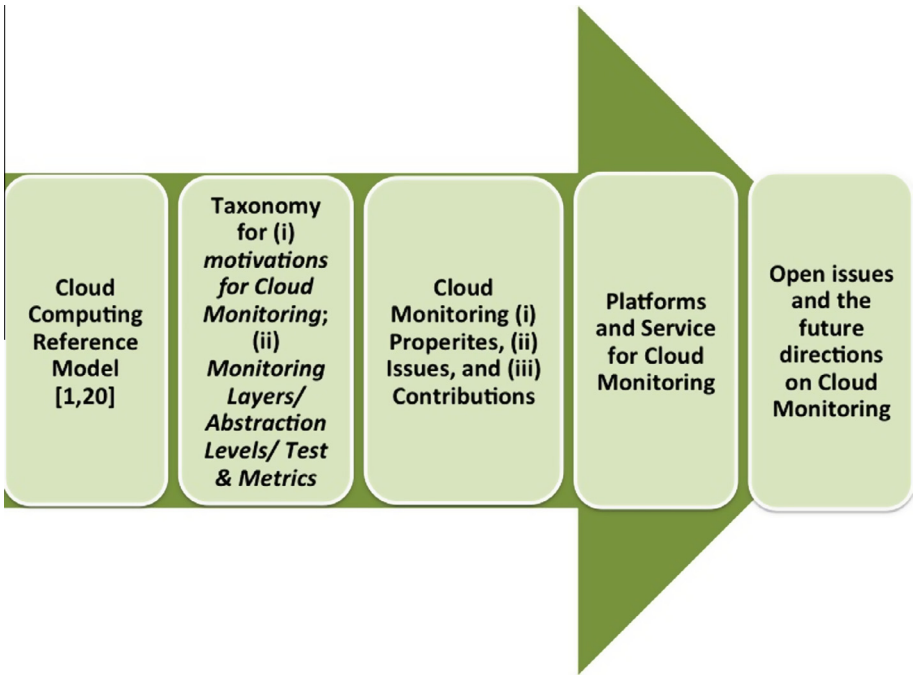

Fig. 1. Research methodology.

Table 1
Cloud Computing: terms and definitions [1,20].

Essential characteristics. Cloud Computing has five essential characteristics as defined by the NIST: (i) *On-demand self-service*; (ii) *broad network access*; (iii) *resource pooling*; (iv) *rapid elasticity*; (v) *measured service*
Service models. According to the type of provided capability, the NIST broadly divided the Cloud Computing services into three categories: (i) *Infrastructure as a service* (*IaaS*); (ii) *platform as a service* (*PaaS*); (iii) *software as a service* (*SaaS*)
Hosting. According to the type of hosting, Cloud Computing can be considered as: (i) *External*; (ii) *internal*
Deployment models. Considering the location of the Cloud, deployment models are typically classified as: (i) *Private cloud*; (ii) *Community Cloud*; (iii) *Public Cloud*; (iv) *Hybrid Cloud*
Roles. Multiple roles can be supported by a Cloud developer, many of which can exist within a single organization: (i) *Cloud Auditor*; (ii) *Cloud Service Provider*; (iii) *Cloud service carrier*; (iv) *Cloud Service Broker*; (v) *Cloud Service Consumer*

Cloud Computing has a number of positive aspects pushing for its rapid adoption, from both economical and technical points of view. As for the former, with respect to other service hosting possibilities, Cloud provides a lower Total Cost of Ownership (TCO), an increased flexibility in terms of both resources and Service Level Agreements (SLAs), and allows to focus on the core business, ignoring the issues related to the infrastructure management. As for the latter, Cloud Computing provides an improved scalability, ubiquitous access to data and resources, and advanced disaster recovery mechanisms.

Together with these positive aspects Cloud Computing has a number of challenges on which the research community and industry are investing a lot of resources: (i) provision of scalability, load balancing, Quality of Service (QoS), service continuity and application performance; (ii) provision and guarantee of SLAs; (iii) management of large scale, complex and federated infrastructures; and (iv) analysis of the root causes of end-to-end performance. In order to cope with such challenges, accurate and fine-grained monitoring and measurement techniques and platforms are required. A careful analysis of the motivations for Cloud monitoring is reported in the following section.

3. Cloud Computing: the need for monitoring

Monitoring of Cloud is a task of paramount importance for both Providers and Consumers. On the one side, it is a key tool for controlling and managing hardware and software infrastructures; on the other side, it provides information and Key Performance Indicators (KPIs) for both platforms and applications. The continuous monitoring of the Cloud and of its SLAs (for example, in terms of availability, delay, etc.) supplies both the Providers and the Consumers with information such as the workload generated by the latter and the performance and QoS offered through the Cloud, also allowing to implement mechanisms to prevent or recover violations (for both the Provider and Consumers). Monitoring is clearly instrumental for all the activities covered by the role of Cloud Auditor. In more general terms, Cloud Computing involves many activities for which monitoring is an essential task. In this Section we carefully analyze such activities, underlining the role of monitoring for each of them. In Fig. 2 these activities are reported in a taxonomy of main aspects of Cloud monitoring considered in this paper.

3.1. Capacity and resource planning

One of the most challenging tasks for application and service developers, before the large scale adoption of Cloud Computing, has always been resource and capacity planning (e.g. Web Services [22,142]). In order to guarantee the performance required by applications and services, developers have to (i) quantify capacity and resources (e.g. CPU, memory, storage, etc.) to be purchased, depending on how such applications and services are designed and implemented, and (ii) determine the estimated workload. However, while an estimation can be obtained through static analysis, testing and monitoring, the real values are unpredictable and highly variable. Cloud Service Providers usually offer guarantees in terms of QoS and thus of resources and capacity for their services as specified in SLAs [23], and they are in charge of their resource and capacity planning so that service and application developers do not have to worry about them [24]. To this end, monitoring becomes essential for Cloud Service Providers to predict and keep track of the evolution of all the parameters involved in the process of QoS assurance [25] in order to properly plan their infrastructure and resources for respecting the SLAs.

3.2. Capacity and resource management

The first step to manage a complex system like a Cloud consists in having a monitoring system able to accurately capture its state [26]. Over the years, virtualization has become a key component to implement Cloud Computing. Hiding the high heterogeneity of resources of the physical infrastructure, virtualization technologies introduced another complexity level for the infrastructure provider, which has to manage both physical and virtualized resources [25,27–29]. Virtualized resources may migrate from a physical machine to another at any time. Hence, in Cloud Computing scenarios (specially in mobile ones [30]) monitoring is necessary to cope with volatility of resources [31] and fast-changing network conditions (which may lead to faults).

In the context of public critical services (e.g., healthcare or other strategic applications), when using IaaS, concerns about QoS and QoP (Quality of Protection) become very critical. Indeed, when adopting Cloud infrastructures, companies and people expect such services to have 100% uptime. Thus, a resilient and trustworthy monitoring of the entire Cloud infrastructures is needed to provide availability [32].

3.3. Data center management

Cloud services are provided through large scale data centers, whose management is a very important activity.

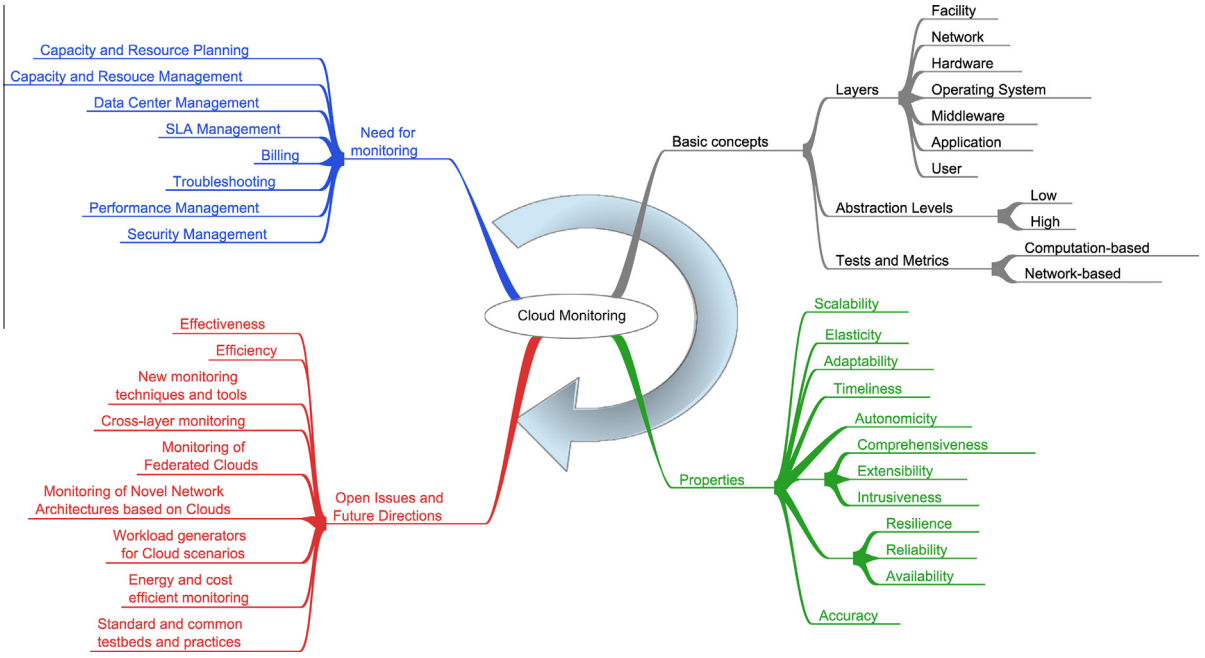

Fig. 2. Cloud monitoring: motivations, properties, basic concepts, open issues and future directions.

Actually, this activity is part of resource management and we reported it here because of its importance and of its peculiar requirements. Data center management activities (e.g. data center control) imply two fundamental tasks: (i) *monitoring*, that keeps track of desired hardware and software metrics; and (ii) *data analysis*, that processes such metrics to infer system or application states for resource provisioning, troubleshooting, or other management actions [33]. In order to properly manage such data centers, both monitoring and data analysis tasks must support real-time operation and scale up to tens of thousands of heterogeneous nodes, dealing with complex network topologies and I/O structures. In this context *energy-efficiency* is a major driver of monitoring data analysis for planning, provisioning and management of resources.

3.4. SLA management

The unprecedented flexibility in terms of resource management provided by Cloud Computing calls for new programming models in which Cloud applications can take advantage of such new feature [34], whose underlying premise is monitoring. Moreover, monitoring is mandatory and instrumental in certifying SLA compliance when auditing activities are performed to respect regulation [35] (e.g. when government data or services are involved). Finally, monitoring may allow Cloud Providers to formulate more realistic and dynamic SLAs and better pricing models by exploiting the knowledge of user-perceived performance [36].

3.5. Billing

One of the essential characteristics of Cloud Computing (see Table 1 and NIST definition [1]) is the offer of "measured services", allowing the Consumer to pay proportionally to the use of the service with different metrics and different granularity, according to the type of service and the price model adopted.

With reference to the service models reported in Section 2, examples of billing criteria are: for SaaS, the number of contemporary users, or the total user base, or application-specific performance levels and functions; in PaaS services, the CPU utilization, or the task completion time; for IaaS, the number of VMs, possibly varying with different CPU/Memory setups [83,117]; we refer the interested reader to [130] for a review of theoretical pricing models.

For each of the reported pricing models and service models, monitoring is necessary both from the Provider side for billing, and from the Consumer side for verifying his own usage and to compare different Providers, a non-trivial process requiring monitoring functionalities and tools [117].

When the billing granularity is coarse – e.g. per VM in IaaS, or up to a maximum database size for a SaaS [80] – the pricing is considered a "flat rate", depending on the subscription duration, and the required monitoring is relatively basic. A significantly more complex scenario is the presence of a Cloud Service Broker (see Section 2 and [20]): in this case advanced monitoring is of paramount importance for the resource provisioning and charge-back strategies at the base of the Cloud Broker's business [66].

3.6. Troubleshooting

The complex infrastructure of a Cloud represents a big challenge for troubleshooting (e.g. root cause analysis), as the cause of the problem has to be searched in several possible components (e.g. network, host, etc.), each of them made of several layers (e.g. real and virtual hardware, host

and guest OS, etc.). A comprehensive, reliable and timely monitoring platform is therefore needed for Providers to understand where to locate the problem inside their complex infrastructure and for Consumers to understand if any occurring performance issue or failure is caused by the Provider, network infrastructure, or by the application itself [37].

3.7. Performance management

Being the hardware infrastructure maintenance delegated to the Providers, the Cloud Computing model is attractive for most Consumers (primarily medium sized enterprises and research groups). However, despite the attention paid by Providers, some Cloud nodes may attain performance orders of magnitude worse than other nodes [38]. If a Consumer adopts a public Cloud to host a mission-critical service or for a scientific application, performance variability and availability become a concern. Therefore, from a Consumer's perspective, monitoring the perceived performance is necessary to adapt to the changes or to apply corrective measures. For instance, a Consumer may decide to host applications at multiple Clouds to ensure high-availability, switching between Clouds depending on the measured performance. Monitoring is then necessary since it may considerably improve the performance of real applications [39] and affect activity planning and repeatability of experiments.

3.8. Security management

Cloud security is very important for a number of reasons. Security is considered as one of the most significant obstacles to the spread of Cloud Computing, especially considering certain kinds of applications (e.g. business-critical ones) and Consumers (e.g. governments) [40]. Different works in literature have provided reviews and recommendations for Cloud security (see e.g. [40] and the references therein, and [41,42]). For managing the security in Cloud infrastructures and services, proper monitoring systems are needed. Moreover, for hosting critical services for public agencies, Clouds have to satisfy strict regulations and prove it. And this can be done through a monitoring system that enables auditing (e.g. to certify the compliance to regulations and obligations, such as keeping data of a user inside country borders) [35,43].

4. Cloud Monitoring: basic concepts

As introduced in Section 3, Cloud monitoring is needed to continuously measure and assess infrastructure or application behaviors in terms of performance, reliability, power usage, ability to meet SLAs, security, etc. [44], to perform business analytics, for improving the operation of systems and applications [45], and for several other activities (see Section 3). In this section we introduce a number of concepts at the base of Cloud monitoring that are used to set the context for the following sections, while in Fig. 2 we report these concepts in a taxonomy we propose for main aspects of Cloud monitoring we consider in this paper.

4.1. Layers

According to the work of the Cloud Security Alliance, a Cloud can be modeled in seven layers: *facility*, *network*, *hardware*, *OS*, *middleware*, *application*, and the *user* [54,41,42]. Considering the roles defined in Section 2, these layers can be controlled by either a Cloud Service Provider or a Cloud Service Consumer. They are detailed in the following:

- *Facility*: at this layer we consider the physical infrastructure comprising the data centers that host the computing and networking equipment.
- *Network*: at this layer we consider the network links and paths both in the Cloud and between the Cloud and the user.
- *Hardware*: at this layer we consider the physical components of the computing and networking equipment.
- *Operating System* (*OS*): at this layer we consider the software components forming the operating system of both the host (the OS running on the physical machine) and the user (the OS running in the virtual machine).
- *Middleware*: at this layer we consider the software layer between the OS and the user application. It is typically present only in the Cloud systems offering SaaS and PaaS service models.
- *Application*: at this layer we consider the application run by the user of the Cloud system.
- *User*: at this layer we consider the final user of the Cloud system and the applications that run outside the Cloud (e.g. a web browser running on a host at the user's premise).

In the context of Cloud monitoring, these layers can be seen as where to put the probes of the monitoring system. In fact, the layer at which the probes are located has direct consequences on the phenomena that can be monitored and observed.

Orthogonally to these layers, *system-wide* and *guest-wide* measurements, as proposed by Du et al. [141] in the context of profiling virtual machines, can be defined when discussing what can be monitored inside and what can be monitored outside a Cloud system.

Besides, due to the very high complexity of Cloud systems, it not possible to be sure that certain phenomena are actually observed or not. For example, if we put a probe into an application that runs into the Cloud, to collect information on the rate at which it exchanges information with other applications running in the same Cloud, we do not necessarily know if this rate comprises also the transfer rate of the network. It depends on if the two applications run on the same physical host or not, and this information is not always exposed by the Provider. Similar issues arise for evaluating the performance of computation: the time required for a task completion can depend on the actual hardware that is executing the instructions (usually exposed only as a CPU model – equivalent) and on the workload due to other virtualized environments

running on the same physical server (which are not exposed at all to the Consumer).

4.2. Abstraction levels

In Cloud Computing, we can have both high- and low-level monitoring, and both are required [46]. High-level monitoring is related to information on the status of the virtual platform. This information is collected at the middleware, application and user layers by Providers or Consumers through platforms and services operated by themselves or by third parties. In the case of SaaS, high-level monitoring information is generally of more interest for the Consumer than for the Provider (being closely related to the QoS experienced by the former). On the other hand, low-level monitoring is related to information collected by the Provider and usually not exposed to the Consumer, and it is more concerned with the status of the physical infrastructure of the whole Cloud (e.g. servers and storage areas, etc.). In the context of IaaS, both levels are of interest for both Consumers and Providers.

More precisely [41], for low-level monitoring specific utilities collect information at the hardware layer (e.g., in terms of CPU, memory, temperature, voltage, workload, etc.), at the operating system layer and at middleware layer (e.g., bug and software vulnerabilities), at the network layer (e.g., on the security of the entire infrastructure through firewall, IDS and IPS), and at the facility layer (e.g. on the physical security of involved facilities through monitoring of data center rooms using video surveillance and authentication systems). Section 6 provides a deep analysis of several platforms (commercial and open source) for both high- and low-level monitoring, while in the following the most common metrics and tests are defined.

4.3. Tests and metrics

Monitoring tests can be divided in two main categories: *Computation-based* and *Network-based* [47]. *Computation-based* tests are related to monitoring activities aimed at gaining knowledge about and at inferring the status of real or virtualized platforms running Cloud applications.

4.3.1. Computation-based

Tests are related to the following metrics: *server throughput*, defined as the number of requests (e.g. web page retrieval) per second; *CPU Speed; CPU time per execution*, defined as the CPU time of a single execution; *CPU utilization*, defined as the CPU occupation of each virtual machine (useful to monitor the concurrent use of a single machine by several VMs); *memory page exchanges per second*, defined as the number of memory pages per second exchanged through the I/O; *memory page exchanges per execution*, defined as the number of memory pages used during an execution; *disk/memory throughput*; *throughput/delay of message passing between processes; duration of specific predefined tasks; response time; VM startup time; VM acquisition/release time; execution/access time, up-time*. All of them can be evaluated in terms of classical statistical indicators (mean, median, etc.) as well as in terms of temporal characterization and therefore stability/variability/predictability. Computation-based tests are operated by the provider or sometimes demanded to third parties. For example, in the case of EC2 and Google App Engine, Hyperic Inc publishes results of these test on CloudStatus [48].

4.3.2. Network-based

Tests are related to the monitoring of network-layer metrics. This set includes *round-trip time* (RTT), *jitter*, *throughput*, *packet/data loss*, *available bandwidth*, *capacity*, *traffic volume*, etc. [49–52]. Using these metrics, several experimental studies in literature compared legacy web-hosting and Cloud-based hosting [53,142].

4.4. A note on Cluster vs Grid vs Cloud monitoring

Similarities and overlapping of properties among Cloud Computing and previous distributed paradigms have led to deep discussion on the definition of Cloud Computing and its peculiar characteristics [1,20,130,131,133]: here we consider the differences from the point of view of monitoring.

Compared with the case of Grid Computing, the monitoring of a Cloud is more complex because of the differences in both the trust model and the view on resource/services presented to the user [131]. In fact the main objective of a Grid is the sharing of resources across multiple organizations [132], implying simpler accounting criteria and limited resource abstraction, which creates a simple relation between monitoring parameters and physical resource status. On the other hand, for the Cloud, the presence of multiple layers and service paradigms (see Section 2) leads to high abstraction of resources, resulting in more opaque relationship between the layer- or service-specific observables and the underlying resources. Moreover we expressly note that in Cloud Computing, even if the abstract interfaces offered to a Consumer could apparently require a reduced necessity for monitoring with respect to Grid, in reality such need is pushed on the Provider of the service, that has to cope with promised or expected performance and with optimization of resources in a highly dynamic and heterogeneous scenario.

This gap of objectives and transparency has to be filled when adopting for a Cloud a monitoring system coming from the Grid Computing field. Finally, as noted in previous sections, the "on demand" service paradigm poses additional challenges to monitoring systems not designed for high *churning* of both users and resources.

Most of the monitoring approaches and platforms proposed for the Grid case [59,60,91–93] have been customized for Cloud systems. Zanikolas et al. [94] surveyed the Grid monitoring research field by introducing the involved concepts, requirements, phases, and related standardization activities (e.g. Global Grid Forum's Grid Monitoring Architecture). Furthermore, they proposed a taxonomy – built by considering scope, scalability, generality and flexibility – of Grid monitoring systems aiming at classifying a wide range of projects and frameworks. In the next section we thoroughly discuss the issues and the proposed solutions regarding the adoption in the Cloud scenario of systems designed for slowly changing fixed infrastructure. These aspects have to be taken into account when consider-

ing Ganglia [59], Nagios [60], MonaLisa [91], R-GMA [92] and GridICE [93] and similar systems to monitor a Cloud. All those differences are even more stressed when comparing Cloud paradigm to Cluster Computing [133]: in this case the relatively rigid architecture, the limited possibilities of service negotiation and the low automation of resource provisioning make Clusters comparable to a base technology for Cloud IaaS Providers, and lead to requirements in terms of monitoring that are a limited subset of the ones of a Cloud. Therefore, most characterizing properties for Cloud monitoring systems either do not apply for Clusters or Grids (namely Elasticity, Adaptability, Autonomicity) or are not vital for their purpose (Comprehensiveness, Extensibility and Intrusiveness).

5. Cloud monitoring: properties and related issues

In order to operate properly, a distributed monitoring system is required to have several properties that, when considered in the Cloud Computing scenario, introduce new issues. In this Section we define and motivate such properties, analyze the issues arising from them, and discuss how these issues have been addressed in literature. In Fig. 2 we report these properties in a taxonomy of main aspects regarding Cloud monitoring considered in this paper.

In Fig. 3 we illustrate the research issues associated with each of the properties considered. This picture shows that, as will be clearer in the following, (i) the research issues to be tackled range in a wide and heterogeneous set, comprising multidisciplinary research areas, and (ii) some of these issues are related with more than one property, i.e. their solution may provide multiple benefits.

5.1. Scalability

A monitoring system is *scalable* if it can cope with a large number of probes [55]. Such property is very important in Cloud Computing scenarios due to the large number of parameters to be monitored about a huge number of resources. This importance is amplified by the adoption of virtualization technologies, which allow to allocate many virtual resources on top of a single physical resource. The measurements required to obtain a comprehensive view on the status of the Cloud lead to the generation of a very large volume of data coming from multiple distributed locations. Hence, a scalable monitoring system should be able to efficiently collect, transfer and analyze such volume of data without impairing the normal operations of the Cloud.

In literature such issue has been mainly addressed by proposing architectures in which monitoring data and events are propagated to the control application after their aggregation and filtering, in order to reduce their volume: *aggregation* combines multiple metrics into a synthetic one that is inferred or not directly monitored; *filtering* avoids useless data to be propagated to the control application.

Most of the proposed architectures, regardless of the specific low-level or high-level monitored parameters, adopt a subsystem to propagate event announcements [23,25,33,56] or rely on agents, which are responsible for performing data collection, filtering and aggregation [23,25,57]. In this context, different aggregation strategies have been proposed: extraction of high-level performance metrics by means of machine learning algorithms [24]; extraction of predicted parameters by combining metrics from different layers (hardware, OS, application and user) and by applying Kalman filters [58]; linear combination of OS-layer metrics [56]. Some architectures further improve scalability by adopting additional optimizations: efficient algorithms for agent deployment and interconnection [57]; Content Based Routing (CBR) and Complex Event Processing (CEP) facilities [37]; lightweight analysis close to the data source, adjustable sampling, time-based filtering, and ad hoc collection and aggregation strategies applied to different partitions of the monitored system [44].

5.2. Elasticity

A monitoring system is *elastic* if it can cope with dynamic changes of monitored entities, so that virtual resources created and destroyed by expansion and contraction are monitored correctly [55]. Such property, also referred to as *dynamism* [23], implies scalability and adds to it the requirement of supporting on-line upsizing or downsizing of the pool of monitored resources.

As opposed to the static nature of previous computing paradigms (e.g. Grid computing), Cloud Computing requires its resources to be dynamic, thus making elasticity an essential property for its monitoring system, as derived

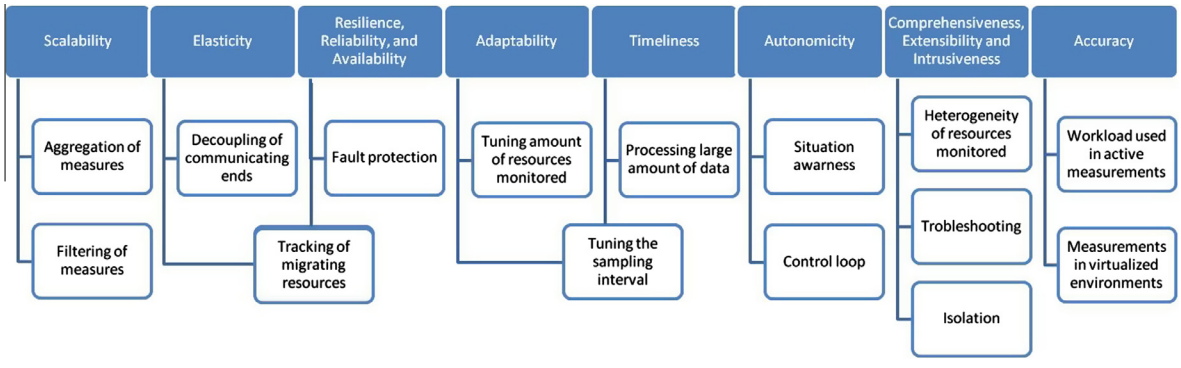

Fig. 3. Properties of systems for Cloud monitoring and related research issues.

from three main drivers: varying assignment of resources to users, varying monitoring requirements for the user, and varying presence of users (multi-tenant scenarios). A challenge in providing elasticity is related with the fact that it is a new fundamental property introduced by Cloud monitoring and not previously considered as a requirement for monitoring generic distributed systems. Therefore, many different monitoring systems proposed for large distributed systems (e.g. Ganglia [59], Nagios [60]) have been designed for a relatively slowly changing physical infrastructure. Thus, they do not assume or support a rapidly changing dynamic infrastructure and they are not suitable for as-is adoption in Cloud scenarios.

In literature, a number of extensions to traditional monitoring systems have been proposed to address this challenge. They basically added the support for monitoring virtualized resources and the condition-triggered reporting in a push fashion, often exploiting a publish-subscribe paradigm to decouple communication ends and to support dynamism. In order to cope with migration of virtual resources, in the Lattice platform [55] an hypervisor controller is responsible of tracking the presence of virtual execution environments (VEEs) by obtaining a list of running VEEs from the hypervisor, on a regular basis. Analyzing such list the controller determines (i) if there is a new VEE, in which case it adds a new probe for that VEE, or (ii) if a VEE has been shut down, in which case it deletes the probe for that VEE. A more comprehensive solution has been provided by Carvalho and Granville [61], which makes Nagios aware of machine virtualization using both *active checks* (a remote execution feature that realizes a pull communication paradigm) and *passive checks* (physical hosts notifying the Nagios server about the VMs that are currently running, implementing a push communication paradigm). An analogous extension to Nagios is provided with a RESTful Event Brokering module [25], which allows the monitoring of both physical and virtual infrastructures; elasticity is obtained by exploiting the design patterns of a traditional service-oriented architecture to realize a twofold push–pull model: monitoring information is pushed by agents towards the management layer and information consumers can pull data from it. More complex solutions are possible when the Cloud is taken into account in the design of the monitoring system. For example, Monalytics [44] has been designed for scalability and effectiveness in heavily dynamic scenarios, providing among the other features: runtime *discovery* of the monitored resources and runtime *configuration* of the monitoring agents. These features are obtained by means of an election-based hierarchy of brokers that collect, process and transmit monitoring information, where the communication topology and the kind of computations are dynamically modified according to the status of the monitored resources.

5.3. Adaptability

A monitoring system is *adaptable* if it can adapt to varying computational and network loads in order not to be *invasive* (i.e. impeding for other activities) [55].

Due to the complexity and the dynamism of the Cloud scenarios, adaptability is fundamental for a monitoring system in order to avoid as much as possible a negative impact of monitoring activities on normal Cloud operations, especially when active measurements are involved. In fact, the workload generated by active measurements, together with the collection, processing, transmission and storage of monitoring data and the management of the monitoring subsystem, require computing and communication resources and therefore constitute a cost for the Cloud infrastructure. Thus, the ability to tune the monitoring activities according to suitable policies is of significant importance to meet Cloud management goals. Providing adaptability is not trivial, because it requires to quickly react to load changes, maintaining the right trade-off between precision (e.g. predictable latencies) and invasivity.

In literature, such issue has been faced by several studies [31,25,57,44,33] by tuning the amount of monitored resources and the monitoring frequency. For instance, Park et al. [31] presented an approach based on Markov Chains, to analyze and predict resource states, in order to adaptively set a suitable time interval to push monitoring information. Another example is the Monalytics system [44], which configures its agents according to the monitoring topologies (i.e. those used to collect, process and transmit monitoring data), modeled as Dynamic Computational communication Graphs (DCG). Depending on the workload, it allows to configure the existing agents in real time – providing them with new monitoring and analysis codes or changing the methods being used – and to dynamically discover and attach to new data sources. The latter approach is further analyzed by Wang et al. [33], who assessed and compared the performance of such dynamic topologies against traditional ones, in terms of time-to-insight (see the following section on Timeliness) and management cost (modeled as capital cost for hardware and associated software management), showing that this approach is both flexible and performance/cost effective.

5.4. Timeliness

A monitoring system is *timely* if detected events are available on time for their intended use [33].

Monitoring is instrumental to activities related with core goals of a Consumer or a Provider, hence failing to get the necessary information on time for the appropriate response (e.g. to raise an alarm, to provision more resources, to migrate services, to enforce a different policy) would void the usefulness of monitoring itself. Timeliness is interdependent with other properties of the monitoring system, such as Elasticity, Autonomicity and Adaptability. Thus, granting it implies the same challenges or trade-offs between opposing requirements. More in detail, the time between the occurrence of an event and its notification can be broken down in different contributions: sampling, analysis and communication delay. Each of them poses some issues. The shorter the sampling interval, the smaller is the delay between the time a monitored condition happens and is captured. Thus, to obtain up-to-date information, a trade-off between Accuracy and sampling frequency is necessary, considering also the resource constraints (e.g. CPU, network bandwidth or memory). The analysis delay poses a similar issue regarding *complex* events (i.e. the result of a computation over multiple

parameters), which requires to consider also the time to get all the necessary information, besides the computing time itself. Finally, being the Cloud a distributed system, the communication delay can be significant because the information may have to travel across multiple links to reach processing nodes and this delay is even more important when considering complex events involving information coming from remote sources.

In literature, the choice of the sampling interval has been considered by Park et al. [31]: in order to cope with highly volatile resources (of mobile devices) a behavioral model of the resource is used to predict the suitable interval duration. The problem of keeping the communication and analysis delays as low as possible has been considered by Wang et al. [33], who proposed a *close-to-the-data* analysis approach realized by performing computations on information gathered by nearby nodes and by adapting the communication and analysis topology to meet low delay goals. In order to evaluate the Timeliness, they defined the *Time to Insight* metric as "the latency between when one monitoring sample (indicating event of interests) is collected on each node and when the analysis on all of those monitoring samples has completed". Such metric is then used to evaluate different communication and analysis topologies and the trade-offs with the related infrastructure costs.

5.5. Autonomicity

An *autonomic* monitoring system is able to self-manage its distributed resources by automatically reacting to unpredictable changes, while hiding intrinsic complexity to Providers and Consumers [62].

As Cloud infrastructures are meant to provide *on-demand self-service* and *rapid elasticity* while operating continuously with minimal service interruptions, it is extremely important for the monitoring system to be able to react to detected changes, faults and performance degradation, without manual intervention. Supporting autonomicity in such a monitoring system is not trivial, since it requires to implement a control loop that receives inputs from a huge number of sensors (i.e. the monitoring data) and propagates control actions to a large number of distributed actuators. This in turn implies elasticity and Timeliness. Moreover the analysis capabilities for situation awareness must be implemented (the complexity and layering of Cloud infrastructure pose obstacles to this) and the definition of suitable policies to drive the behavior of the monitoring system in response to the detected events is necessary.

In literature such issues have been addressed by several studies [62–65,55,33] and applied to different kinds of events. For example, focusing on bottlenecks and over-provisioning for a multi-tier Web application hosted on a Cloud, Iqbal et al. [63] proposed two methodologies for the automatic detection and resolution of bottlenecks and for the identification and the retraction of over-provisioned resources. Such methodologies are driven by maximum response time requirements and are shown to be useful to provide SLAs. About system failures, Ayad and Dippel [65] propose an agent-based monitoring system that continuously checks for the availability of VMs and automatically recovers them in case of failures. In order to automatically cope with SLA violations, the DeSVi architecture [64] allocates computing resources for a requested service based on user requests, and arranges its deployment on virtualized infrastructures. Resources are monitored using a framework capable of mapping low-level resource metrics (e.g. host up- and down-times) to user-defined SLAs (e.g. service availability). The detection of possible SLA violations is performed using predefined service level objectives and knowledge databases for the analysis of the monitored information, thus allowing to automatically propose reactive actions.

5.6. Comprehensiveness, Extensibility and Intrusiveness

A monitoring system is *comprehensive* if it supports different types of resources (both physical and virtualized), several kinds of monitoring data, and multiple tenants [23]; it is *extensible* if such support can easily be extended (e.g., through plug-ins or functional modules); it *intrusive* if its adoption requires significant modification to the Cloud [25].

The first two properties are strictly related: the latter represents the possibility to enhance the former without modifying the monitoring framework. Having a comprehensive monitoring system is useful for both developers (IaaS and PaaS Consumers) and their respective Providers. The advantage for the former is related to the possibility to adopt a single monitoring API, independently of what kind of monitoring information is actually used. For the latter, the advantage consists in deploying and maintaining only one single monitoring infrastructure. By also providing extensibility, such advantages can easily persist to changes or additions of underlying components, and maintaining low intrusiveness allows to minimize the instrumentation costs. Cloud Computing is a relatively new paradigm and no common standards have been widely adopted by deployed systems. Most non-Cloud-specific monitoring systems were already designed to provide extensibility and low intrusiveness, and their extension to Cloud scenarios retained such features [59,60,26,66]. Some issues arise when considering comprehensiveness. A first issue is related to the fact that a holistic monitoring system has to support different underlying architectures, technologies, and resources, while preserving isolation among different tenants. On the other side, a comprehensive monitoring system allows to better perform troubleshooting activities, which raises another issue due to the intrinsic dynamicity of Cloud environments and to the large number and heterogeneity of resources and parameters considered at different layers.

The preservation of isolation has been explicitly addressed for the first time in literature by Hasselmeyer and d'Heureuse [23] in their agent-based architecture. It achieves isolation in terms of tenant visibility by directing monitoring information flows through the same stream management system, which exposes the information only to the intended recipients. Moreover, in order to allow for interoperability of the functional blocks, these are connected with adapters that abstract the data from the specific technologies. Regarding the support for heterogeneous virtualized environments,

the VMDriver [71] monitoring subsystem has been proposed for the interception of events occurring at the VM level (set at the OS layer). This allows to monitor the state of virtual machines hiding the differences of guest OSes. About the issues implied in troubleshooting large numbers of dynamic and heterogeneous components, several studies have been carried out to understand the cause of the performance observed in Cloud environments. Most of them selected Amazon EC2 as a case study [67,68,39]. Hill and Humphrey [67] were unable to identify the cause of the performance observed for scientific applications. For instance, in their experiments, the maximum rate at which two processes were exchanging information could be attributed to different causes (network, L2 caches, etc.), depending on where the processes were running (in different hosts in the same or different datacenters, in the same host, etc.), on the number of concurrent processes (and therefore VMs) on the same host, etc. Wang and Ng [68] found that, even when the data center network is lightly utilized, virtualization can still cause significant throughput instability and abnormal jitter, and identified the processor sharing mechanism as the main responsible. Schad et al. [39] found the performance observed at different layers (application and OS) significantly variable with time and VM instances, thus impacting data-intensive applications predictability and repeatability of wall-clock timed experiments. Working on a small testbed, Mei et al. [47] focused on the impact of co-locating applications in a virtualized Cloud in terms of throughput and resource sharing effectiveness. They found that in presence of idle instances, other VMs result to be scheduled less frequently for less time, which is primarily due to two factors: (i) the execution of timer tick for the idle guest domain and the context switch overhead, and (ii) the processing of network packets, such as address resolution protocol (ARP) packets, which causes I/O processing in guest domain. They observed also that the duration of performance degradation experienced due to the creation of new VMs on demand is typically bounded within 100 s, and it is related with the machine capacity, the workload level in the running domain, and the number of new VM instances to start up. Finally, they found that co-locating two applications on VMs hosted on the same physical machine produces performance degradation when involving CPU intensive tasks and, when multiple guest domains are running, the context switches among them lead to more frequent cache and translation lookaside buffer (TLB) miss, which result in more CPU time consumption in serving the same data.

5.7. Resilience, Reliability, and Availability

A monitoring system is *resilient* when the persistence of service delivery can justifiably be trusted when facing changes [69], that basically means to withstand a number of component failures while continuing to operate normally; it is *reliable* when it can perform a required function under stated conditions for a specified period of time; it is *available* if it provides services according to the system design whenever users request them [70].

As monitoring is functional to critical activities of the Cloud, such as billing, SLA compliance verification and resource management (see. Section 3), the monitoring system has to be resilient, reliable and available in order not to compromise such activities. With the heavy usage of virtualization technologies by Cloud platforms, monitored hosts and services can migrate from a physical computer to another, invalidating the classical monitoring logics and mining the reliability of the monitoring system. Hence, the necessity to provide such properties for Cloud monitoring poses several issues, such as tracking and managing heterogeneous monitored and monitoring resources, characterizing possible faults of the monitoring system itself and protecting against them.

Several research works considered different aspects regarding resilience. Some works [30,31] tackled the resilience to faults in mobile Cloud Computing scenarios, where mobile devices – which now have significant computational power and storage space – are considered as a high volatile resource, and such volatility influences the choice of the monitoring frequency. Ayad and Dippel [65] considered resilience as affected by the adopted virtualization technologies. As for reliability, some researchers aimed at determining the performance of particular Clouds, for example when analyzing the performance of Amazon Web Services, experiencing difficulties related to the fact that the probes used for monitoring were not available in some periods [72]. Romano et al. [37], proposed a Cloud monitoring facility suitable for QoS, called QoS-MONaaS, which stands for "Quality of Service MONitoring as a Service", that is specifically designed to be reliable and offers monitoring facilities "as a Service", allowing its user (Provider or Consumer) to describe in a formal SLA the Key Performance Indicators (KPIs) of interest and the alerts to be raised when an SLA breach is detected. The basis for its reliability is drawn from high-level facilities provided by an underlying platform (SRT-15 [73]), where anonymization is applied before processing monitoring data. Finally, focusing on availability, Padhy et al. [32] considered byzantine faults affecting a Cloud monitoring system and proposed a publish-subscribe paradigm for communication and event handling with redundant brokers, and leveraged Byzantine Fault Tolerance algorithms [74] to ensure tolerance to attacks and failures.

5.8. Accuracy

We consider a monitoring system to be *accurate* when the measures it provides are accurate, i.e. they are as close as possible to the real value to be measured.

The accuracy is important for any distributed monitoring system because it can heavily impact the activities that make use of the monitoring information. For instance, when the monitoring system is used for troubleshooting, inaccuracy in measure may lead to incorrect identification of the cause of the problem. In the context of Cloud Computing, accuracy becomes even more important. Firstly, since Cloud services are subject to well-defined SLAs, and Providers have to pay penalties to their customers in case of SLAs violations, inaccurate monitoring can lead to money loss. Secondly, being the monitoring system used for important activities of the Cloud (see Section 3), accurate monitoring is necessary to effectively and efficiently perform them. The analysis of the literature reveals two main issues related to the accuracy of monitoring systems in Cloud Computing scenarios. The first one is related to

the workload used to perform the measurements: in order to monitor the Cloud, especially when using active monitoring approaches, it is necessary to apply a suitable stress (e.g. the HTTP GET to a WEB server in the Cloud must arrive with a certain statistical distribution in order to accurately compute the average response time of the WEB server). The second issue is related to the virtualization techniques used in the Cloud: performed measurements may be affected by errors imputable to the virtualization systems that add additional layers between applications and physical resources (e.g. time-related measurements are impaired by the sharing of physical resources such as CPUs, interface cards, and buffers).

Several contributions have been provided in literature with regard to these two issues. As for the workload, research efforts in this area comprise the characterization of real workloads, the reproduction of such workload in the Cloud, which tests to perform and how, which parameters to measure, etc. A number of research groups carried out experimental campaigns on different Clouds to understand their performance, both in general and for specific applications. Several studies [62,72,75,76,67,77–79] investigated the performance of specific Clouds in order to understand if and how they can support scientific and high performance applications. Most of these works can be located at the application layer [62,72,75,76,67] because they used custom applications running in the Cloud. Ostermann et al. [72] performed also an analysis at user layer, thanks to emulated web browsers issuing requests to servers running in the Cloud. The range of metrics considered is very wide and includes the money cost for using the Cloud services [62], execution time of specific jobs [62,75,76], VM acquisition/release times [72], disk throughput [72,67], access time and throughput of memory [72,67], CPU speed [72], and throughput and/or latency of messages exchanged by the applications [72,76,67]. Moreover, most of these research works have been conducted on Amazon EC2 [62,72,75,76,67] while a few others used also other kinds of testbeds, typically located at the researchers' premises [75,67]. On the other hand, different works in literature studied the possibility to use Cloud for supporting database [80,39] and service-oriented [81,82,56] applications. Typically these studies are conducted at application [81,39] or user layer [81,80,39]. The metrics considered by these works include CPU speed [81,39], disk throughput [81,39], VM startup time [39], throughput, jitter and loss of the network [39], memory throughput [39], server throughput [80] and money cost [80]. These research works have been conducted on several different commercial Clouds, including Amazon EC2 [81,80,39], Google App Engine [80], Microsoft Azure [80], as well as on local testbeds [39]. Finally, Binnig et al. [83] evidenced a number of limitations of the benchmarks used by many of the previously cited works. In particular, they suggest that aspects such as scalability, peak loads, and fault tolerance are not considered by current state-of-the-art benchmarks such as TPC-H for OLAP [84], TPC-C for OLTP [85], or TPC-W [86] for e-commerce applications. The authors also propose a number of other tests and parameters to evaluate these important aspects of modern Clouds.

As for the impact of virtualization on measurement accuracy, the works in literature analyzed the accuracy of RTT [87–89], jitter, capacity and available bandwidth [87], topology [87], and also the performance of auto-tuning applications [90]. Regarding the delay, jitter, capacity and available bandwidth, the problem consists in having accurate time stamping at the measuring nodes. Implementing VMs at the end nodes requires a timely scheduling and switching mechanism between the different VMs. As a consequence, packets belonging to a specific VM may be queued until the physical system switches back to that VM, which leads to inaccurate time stamping [87]. Some works [88,89] reported that accurate RTT measurements are possible only under low network and computing loads, and that most delay is introduced while sending packets (as opposed to receiving packets). They conclude that kernel-space timestamps are not enough accurate under heavy network load, and access to timestamps as seen by physical network interfaces would be necessary to overcome this issue [89]. Regarding topology measurements, Abujoda [87] proved that network virtualization generates several virtual topologies on top of a single physical topology, and common active measurement tools like *traceroute* are unable to discover the real physical topology. Moreover, their accuracy is affected by the migration of nodes, which dynamically modifies the placement of virtual nodes and the distance among them. Finally, regarding the performance of auto-tuning applications, Youseff et al. [90] showed that the combination of ATLAS auto-tuning and Xen paravirtualization delivers native execution performance and nearly identical memory hierarchy performance profiles. Moreover, they showed that paravirtualization has no significant impact on the performance of memory-intensive applications, even when memory becomes a scarce resource.

6. Cloud monitoring platforms and services

In this section we review the most spread commercial and open source platforms for Cloud monitoring as well as services that can help Consumers to assess the performance and the reliability of Cloud services (see Table 2). We describe both Cloud management solutions that contain a module specifically targeted to the monitoring and solutions whose only target is Cloud monitoring.

6.1. Commercial platforms

According to the definitions reported in Section 4, commercial platforms implement both high- and low-level monitoring.

6.1.1. CloudWatch

In line with other commercial Providers, Amazon does not provide information on the low-level monitoring system used, and the way monitoring data are gathered, collected and analyzed is secret. At high level, Amazon provides users with a service called CloudWatch. CloudWatch is able to monitor services like EC2, in which collected information are mainly related to the virtual

Table 2
Cloud monitoring platforms and services.

Commercial platforms	Open source platforms	Services
CloudWatch [95]	Nagios [104]	CloudSleuth [112]
AzureWatch [137]	OpenNebula [105]	CloudHarmony [113]
CloudKick [96]	CloudStack ZenPack [108]	Cloudstone [114]
CloudStatus [48]	Nimbus [110]	Cloud CMP [116]
Nimsoft [97]	PCMONS [111]	CloudClimate [118]
Monitis [99]	DARGOS [128]	Cloudyn [119]
LogicMonitor [100]	Hyperic-HQ [138]	Up.time [120]
Aneka [101]	Sensu [139]	Cloudfloor [121]
GroundWork [129]		CloudCruiser [122]
		Boundary [136]
		New Relic [140]

platforms. CloudWatch gathers several kinds of monitoring information and it stores them for two weeks. On these data, users can build plots, statistics, indicators, temporal behaviors, thresholds, alarms, etc. Alarms can trigger specific actions like event notification, through the Amazon SNS service, or Autoscaling [95]. The billing of this monitoring service is managed separately and it is independent of the monitored resources. Recently, Amazon is changing the billing plans for the monitoring service, making it free of charge with basic features and a sampling rate of five minutes, and charging the advanced features and the sampling rate of one minute [46]. CloudWatch mainly focuses on Timeliness, Extensibility, and Elasticity, while it results to be limited in terms of cross-layer monitoring (see Section 7).

6.1.2. AzureWatch

Although the Windows Azure SDK offers to developers a specific software library to monitor their applications, some third-party monitoring services have been developed around it. Among them we considered AzureWatch [137], which monitors and aggregates key performance metrics from the following Azure resources: instances, databases, database federations, storage, websites and web applications. It further supports user-defined performance counters related to quantifiable application metrics. According to the information available on the website, it explicitly addresses Scalability, Adaptability, Autonomicity, and Extensibility.

6.1.3. CloudKick

RackSpace, through *Cloud Sites*, provides its users with monitoring data like CPU utilization and traffic volume. In addition, RackSpace provides tools, called *Cloud tools*, able to build a complete monitoring solution with specific focus on virtual machines and alerting mechanisms. RackSpace has recently acquired CloudKick [96], a multi-Cloud management platform with a wide range of both high- and low-level monitoring features and metrics, and the possibility to develop custom plugins. Monitoring data can be visualized in real time and alert systems can be configured to inform users in real time (e.g. through email or SMS) [46]. The platform mainly addresses Scalability and Adaptability.

6.1.4. CloudStatus

CloudStatus [48], built on top of Hyperic-HQ, is one of the first independent Cloud monitoring services supporting Amazon Web Services and Google App Engine. It provides monitoring of user application performance, a methodology for evaluating the root cause analysis of performance changes and degradations, and both real time and weekly trends of monitored metrics. The main feature advertised of such platform is Timeliness.

6.1.5. Nimsoft

Nimsoft Monitoring Solution (NMS) [97] is able to monitor data centers of both private and public Clouds. It provides a single view on the IT infrastructures and services provided by Google Apps, RackSpace Cloud, Amazon, Salesforce.com and others through a "unified monitoring dashboard". It has been used for monitoring SLAs [98] and it provides Scalability and Comprehensiveness as main features.

6.1.6. Monitis

Monitis [99] adopts agents installed on the resources to be monitored to inform users about the service performance and to send alerts when resource are considered scarce. It is mainly focused on Amazon services and provides an open API, based on the HTTP REST protocol, for extending and customizing the platform. Its main feature is Comprehensiveness.

6.1.7. LogicMonitor

LogicMonitor [100] allows to monitor virtualized infrastructures by adopting an elastic multi-layer approach. Ranging from web servers or databases running on VMs to the underlying hypervisors, it automatically discovers and monitors newly added or deleted resources as they are provisioned, by properly grouping them and sending related notifications. It natively supports several virtualized environments (e.g. Citrix XenServer, VMware vSphere [127], and ESX) and Cloud platforms (e.g. Amazon EC2, and Eucalyptus). All the information is visible through flexible dashboards, which allow to correlate performance and resolve issues. It provides Scalability, Elasticity, and Comprehensiveness as main features.

6.1.8. Aneka

Aneka [101,66,102] is a framework for the development, deployment, and management of Cloud applications. Aneka consists of a scalable Cloud middleware that is deployed on top of heterogeneous computing resources, and an extensible collection of services, coordinating the execution of applications, monitoring the status of the Cloud, and providing integration with existing Cloud technologies. Aneka provides an extensible API for the development of distributed applications, integration of new capabilities into the Cloud, and support of different types of Clouds: public, private, and hybrid. Aneka implements a service-oriented architecture (SOA), and services are the fundamental components of an Aneka Cloud. The framework includes the basic services for infrastructure and node management, application execution, accounting, and system monitoring. The middleware represents the distributed infrastructure constituting Aneka Clouds and

provides a collection of services for interaction with the Cloud, which include monitoring, execution, management, and all the other functions implemented in the framework. Its monitoring component mainly focuses on Scalability and Elasticity.

6.1.9. GroundWork

GroundWork [129] can monitor any class of device or virtual entity in a data center, from servers to security devices. With its open platform, new devices are easy to add using plugins and connectors and, thanks to the use of Nagios, it can integrate the thousands of available Nagios plugins for expanded monitoring coverage. As for the Cloud and the virtualization Monitoring, GroundWork monitors infrastructure and applications virtualized or physical in the Cloud or at the users' premises. GroundWork supports virtualization Providers such as VMware and Cloud Providers like Amazon: using monitoring from someone different than the provider, it is easier to get common metrics, verify service levels, and pursue a multi-vendor strategy for cost avoidance. It mainly focuses on Comprehensiveness.

6.2. Open source platforms

6.2.1. Nagios

Nagios [104] is a well-known enterprise-class open source monitoring platform that has been extended to support the monitoring of Cloud infrastructures. It has been extended with monitoring capabilities for both virtual instances and storage services [46]. Thanks to such extensions it has been adopted for monitoring Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs To Useful System) [103], a well-known open source platform for Cloud Computing, compatible with both EC2 and S3 Amazon services. It is also used for monitoring OpenStack [107], an open source Cloud platform for IaaS (Ubuntu adopts it as standard private Cloud solution since release 11.10) composed of three main projects: *Compute, Object Store,* and *Image Service.* The main feature offered by Nagios is Extensibility.

6.2.2. OpenNebula

OpenNebula [105][106] is an open source toolkit for the management of distributed and heterogeneous public, private, and hybrid Cloud infrastructures. Through a module called Information Manager, it monitors Cloud physical infrastructures and provides information to Cloud Providers. Monitoring data are collected through probes installed on the nodes, queried through SSH connections, and they are related to information concerning the status of physical nodes. It provides Scalability and Adaptability as main features.

6.2.3. CloudStack ZenPack

CloudStack [108] is an open source software written in Java, designed to deploy and manage large networks of virtual machines, as a highly available and scalable Cloud platform. It currently supports the most popular hypervisors (e.g. VMware, Oracle VM, KVM, XenServer, and Xen Cloud Platform), and offers three ways to manage Cloud Computing environments: an easy-to-use web interface, a command line tool, and a full-featured RESTful API. In order to monitor CloudStack virtual and physical devices, a Zenoss extension called ZenPack [109] can be used. It manages both alerts and events and provides the parameters (aggregated from all zones, pods, Clusters and hosts) related to the memory, CPU, and storage, as well as to the network. The main feature offered by the CloudStack ZenPack is Timeliness.

6.2.4. Nimbus

The Nimbus [110] platform is an integrated set of tools (application instantiation, configuration, monitoring, repair, etc.) to implement infrastructure Clouds for scientific users supporting the combination of OpenStack, Amazon, and other Clouds. Its Infrastructure is an open source EC2/S3-compatible IaaS implementation, specifically targeting features of interest for the scientific community, such as support for proxy credentials, batch schedulers, best-effort allocations, etc. As for monitoring, a set of tools provide a structure and APIs for launching, configuring, and monitoring Cloud applications, the most important of which are Context Broker and *cloudinit. d*. The Context Broker, by adopting a "pull" model, allows to coordinate large virtual Cluster launches automatically and repeatably. A launch can consist of many VMs and can span multiple IaaS Providers, including offerings from commercial and academic space. *cloudinit. d*, by adopting a "push" model, allows to launch, configure, monitor, and repair a set of interdependent virtual machines in an IaaS Cloud or over a set of IaaS Clouds. Nimbus monitoring components mainly focus on Autonomicity.

6.2.5. PCMONS

The Private Cloud MONitoring System [111] is composed of the following seven modules:

- *Node Information Gatherer*. It is responsible of gathering and collecting information on local nodes (e.g. information on VMs) and sending them to the Cluster Data Integrator.
- *Cluster Data Integrator*. It is responsible of organizing the nodes in Clusters and, through an agent, of collecting data for the other modules.
- *Monitoring Data Integrator*. It is responsible of collecting and storing data in a database and provides information to Configuration Generator.
- *VM Monitor*. It is responsible of installing and executing scripts over VMs to gather data of interest.
- *Configuration Generator*. It retrieves data from the database and generates the configuration files for other tools (e.g., visualization of monitoring data).
- *Monitoring Tool Server*. It is responsible of receiving monitoring data from different resources and updating the database. The current version adopts the Nagios format.
- *User Interface*. The current version uses the Nagios interfaces.

The first release of PCMONS is compatible with Eucalyptus for the monitoring of the infrastructure and with Nagios for the visualization of monitoring data. The architecture of PCMONS is reported in Fig. 4. Its main feature is Extensibility.

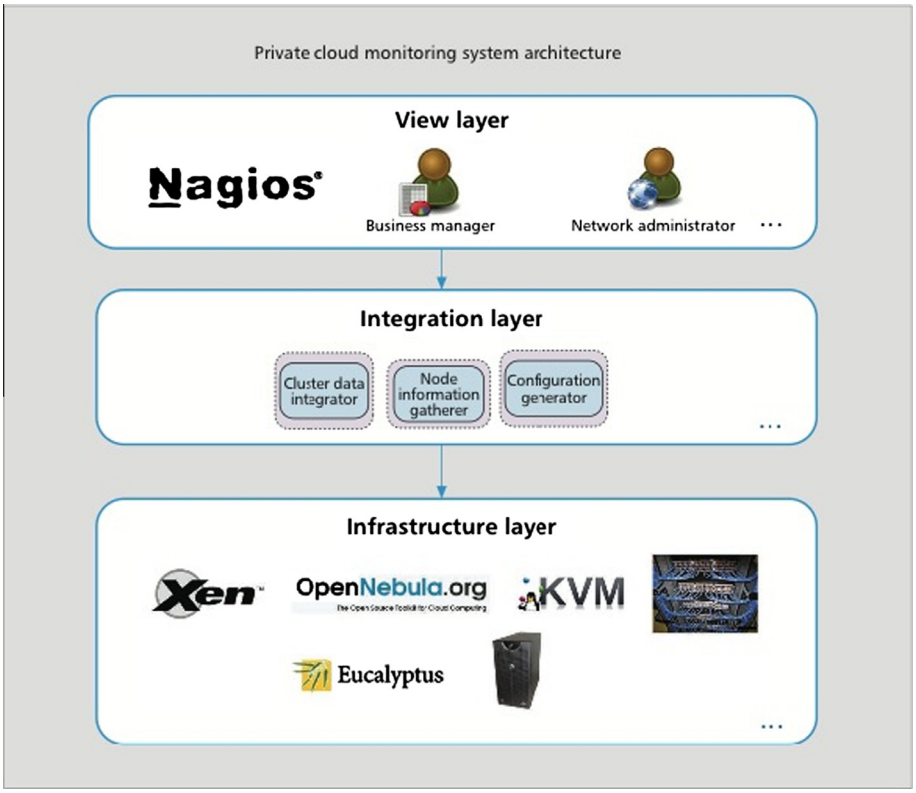

Fig. 4. PCMONS architecture. (source [111])

6.2.6. DARGOS

DARGOS [128] is a distributed Cloud monitoring architecture using a hybrid push/pull approach to disseminate resource monitoring information. DARGOS provides measures of the physical and virtual resources in the Cloud while maintaining a low overhead. In addition, it has been designed to be flexible and extensible with new metrics easily. DARGOS architecture (see Fig. 5) is composed of two main components:

- *Node Monitoring Agent* (*NMA*): The NMAs are responsible of collecting the statistics of resource usage (CPU, Memory, Hypervisor...) in a certain node and deliver them to the NSAs. Each NMA is associated with a certain zone in the Cloud. Often, it is installed in nodes that are the resource pool for the Cloud.
- *Node Supervisor Agent* (*NSA*): The NSA subscribes to monitoring information being published by NMA. It caches locally resource information received. NSAs can monitor multiple zones at the same time (specified by a regex).

DARGOS mainly copes with Extensibility, Adaptability, and Intrusiveness.

6.2.7. Hyperic-HQ

Hyperic-HQ [138] is the open source core of the CloudStatus platform and supports the management and monitoring of Cloud infrastructures performance, including both virtual and physical resources. Its Java-based agents support any platform, including Unix, Linux, Windows, Solaris, AIX, HPUX, VMware, and Amazon Web Services. It further provides detailed reporting and analysis on critical data that analyze IT and web operations service levels, resource utilization efficiency, exception reports and operations strategies. It mainly focuses on Scalability and Comprehensiveness.

6.2.8. Sensu

Designed to overcome the limits of traditional monitoring platforms in Cloud environments, Sensu [139] is based on RabbitMQ, a message-oriented middleware that includes a monitoring server, platform independent agents and a web-based dashboard. It leverages Advanced Message Queing Protocol (AMQP) for scalable processing and secured communication, and implements a REST-based JSON API for data retrieval. The platform is mainly focused on Extensibility and Elasticity.

6.3. Services for assessing cloud performance and dependability

6.3.1. CloudSleuth

CloudSleuth [112] is a web-based Cloud performance visualization tool. Its main objective is the analysis of a notable number of public IaaS and PaaS Providers by monitoring two user-layer properties: Reliability and Timeliness. The test is performed accessing from geographically

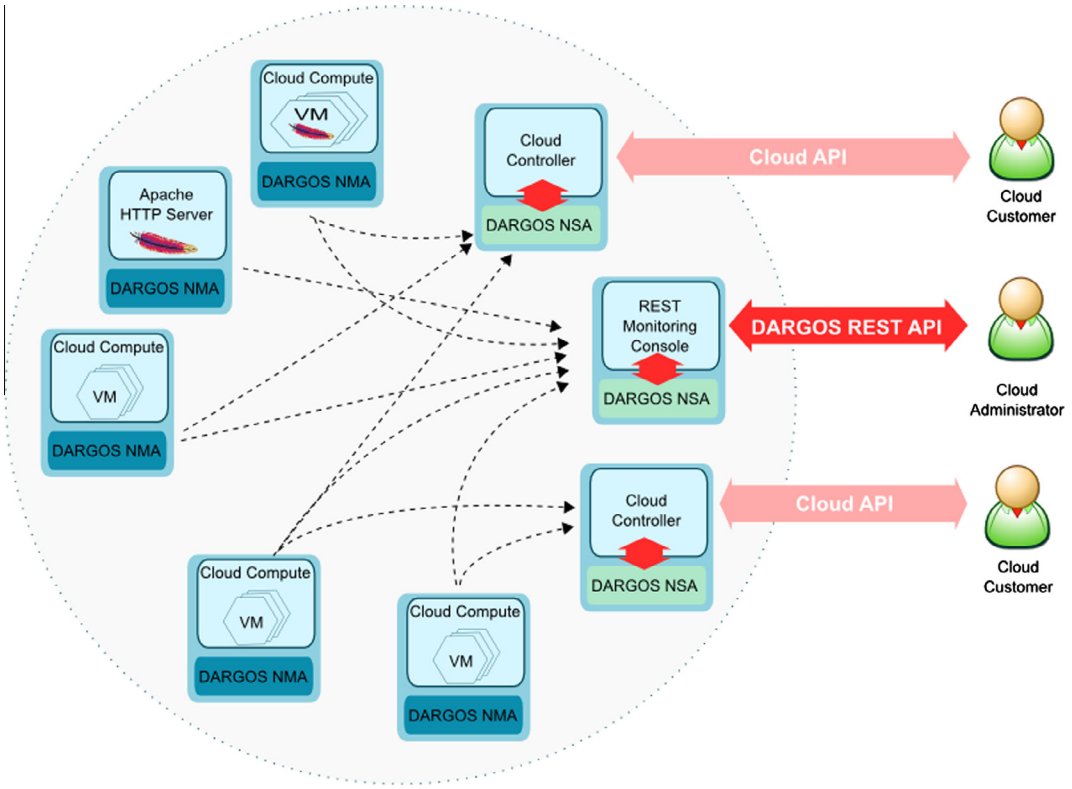

Fig. 5. DARGOS architecture. (source [128])

distributed locations (Gomez Performance Network) a simple application deployed on the monitored Clouds. This application is implemented with two main objectives: to mimic a website with dynamic content, and to be equally deployable on different kinds of IaaS or PaaS. The experienced performance is plotted over different time intervals, showing the evolution over time of the Cloud response times.

6.3.2. CloudHarmony

CloudHarmony [113] provides a wide set of performance benchmarks of public Clouds. The tests consider the common OS-layer metrics (related to CPU, disk and memory I/O), a wide set of application-layer benchmarks, such as Unixbench and IOzone (considering completion times for tasks like integer and floating point operations, file system access, system call overhead, etc.), and user-layer tests (RTT and throughput experienced by a web application). Furthermore, Cloud-to-Cloud network performance is assessed in terms of RTT and throughput. Finally, Cloud uptime monitoring on large time scales is performed through a geographically distributed network able to check connectivity in different ways, the basic ones being ping and TCP port checks. Such service mainly provides Comprehensiveness and Timeliness.

6.3.3. Cloudstone

Cloudstone [114,115] is a UC Berkeley project aimed at providing a benchmark for reproducible and fair performance assessment on Clouds. All of its components are open-source and are chosen to implement a model of realistic usage of a Web 2.0 application. A whole application is provided to be deployed on a IaaS, and to be tested with *Faban*, a Markov-chain-based workload generator. Tools to manage the deployment, testing and results report are also available. A notable characteristic is the overall performance index considered: "dollars per user per month", i.e. the cost for serving a given number of users with given QoS (expressed in terms of percentile of requests served below a given time threshold). The project mainly focuses on Accuracy and Availability.

6.3.4. Cloud CMP

Cloud CMP [116,117] is a tool developed by Duke University and Microsoft Research aimed at comparing the cost effectiveness of different Cloud Providers. This is done by extensively assessing the performance of a common core set of offered facilities, including computing instances, storage, intra-Cloud networks and Cloud-to-user network. For each facility, a number of metrics are evaluated. The benchmark suite is publicly available and consists of a web service to be deployed on a Cloud instance, commanded by clients to execute the requested benchmark tasks and report results. The tool mainly addresses Accuracy and Availability.

6.3.5. CloudClimate

CloudClimate [118] is a website displaying graphs of monitoring tests run on different Clouds (from different Providers and at different locations). Monitored Clouds

host IaaS services used as probes: each application runs workload tests (assessing CPU, memory and disk performance) and sends HTTP and ping requests to the others to assess user-layer performance in terms of latency and availability. Content Delivery Network (CDN) instances are also tested in a similar way, evaluating download times of a 65 kB file. The results are plotted on a 1-month timescale reporting monitored metrics as measured from different vantage points. This allows to infer by comparison where the root cause for possible anomalies is located: the virtual server, the Cloud infrastructure, or which end-to-end network path. The project is mainly focused on Availability, Timeliness and Resilience.

Other interesting and powerful Cloud monitoring services are Cloudyn [119], Up.time [120], Cloudfloor [121], CloudCruiser [122], Boundary [136], and New Relic [140]. They are all agent-based and mostly give insights on Availability, Timeliness, Accuracy and Resilience.

6.4. Current overall picture of Cloud monitoring solutions

As highlighted above, there is a large number of solutions for monitoring public and private Cloud platforms, having different properties and each mainly focusing on a subset of the features enumerated in Section 5. We underline how some features, namely Intrusiveness, Resilience, Reliability, Availability and Accuracy, are not explicitly considered or advertised by most commercial or open source solutions for Cloud monitoring (see Tables 3 and 4). Most interesting, this set of seemingly marginal properties, if referred to the monitored Cloud instead of the monitoring platform, are specifically evaluated by most of the services that assess Cloud performance and dependability (see Table 5). This shows that properties highly valued for Cloud services are currently not central for most of the analyzed Cloud monitoring platforms themselves. We also notice how several issues (see Section 7) are not yet deemed crucial by the considered platforms and services and foresee space for future research in such direction.

The platforms and services surveyed above allow to collect a number of different kinds of metrics. The type and the number of such metrics can indeed be very high, and most of the platforms and services also allow to define new ones. Describing all the metrics for all the services is out of the scope of this paper both because of the detail level necessary for this description and of the space required. As an example, CloudKick by RackSpace [96] allows to collect a very high number of metrics from inside (through agents) and outside (called remote check) the monitored hosts. The type of metrics span from memory and CPU to disk and network. For each type, there are a number of metrics that can be collected (e.g., for memory usage there is the actual amount of free and used memory, the swap pages in and out, the total memory available and used, etc. plus user-defines ones).

7. Cloud monitoring: open issues and future directions

The infrastructure of a Cloud is very complex. This complexity translates into more effort needed for management and monitoring. The greater scalability and larger size of Clouds compared to traditional service hosting infrastructures, involve more complex monitoring systems, which have therefore to be more scalable, robust and fast. Such systems must be able to manage and verify a large number of resources and must do it effectively and efficiently. This has to be achieved through short measurement times and fast warning systems, able to quickly spot and report performance impairments or other issues, to ensure timely interventions such as the allocation of new resources.

Table 3
Key properties of commercial cloud monitoring platforms.

Platform	Scalability	Elasticity	Adaptability	Timeliness	Autonomicity	Comprehensiveness	Extensibility	Intrusiveness	Resilience	Reliability	Availability	Accuracy
CloudWatch [95]		✓		✓			✓					
AzureWatch [137]	✓		✓		✓		✓					
CloudKick [96]	✓		✓									
CloudStatus [48]				✓								
Nimsoft [97]	✓					✓						
Monitis [99]						✓						
LogicMonitor [100]	✓	✓				✓						
Aneka [101]	✓	✓										
GroundWork [129]						✓						

Table 4
Key properties of open source cloud monitoring platforms.

Platform	Scalability	Elasticiy	Adaptabilitiy	Timeliness	Autonomicity	Comprehensiveness	Extensibility	Intrusiveness	Resilience	Reliability	Availability	Accuracy
Nagios [104]							✓					
OpenNebula [105]	✓		✓									
CloudStack ZenPack [109]				✓								
Nimbus [110]					✓							
PCMONS [111]							✓					
DARGOS [128]			✓				✓	✓				
Hyperic-HQ [138]	✓					✓						
Sensu [139]		✓					✓					

Table 5
Properties assessed by cloud performance and dependability monitoring services.

Platform	Scalability	Elasticiy	Adaptabilitiy	Timeliness	Autonomicity	Comprehensiveness	Extensibility	Intrusiveness	Resilience	Reliability	Availability	Accuracy
CloudSleuth[112]				✓						✓		
CloudHarmony [113]				✓		✓						
Cloudstone [114]											✓	✓
Cloud CMP [116]											✓	✓
CloudClimate [118]				✓					✓		✓	
Cloudyn [119]				✓					✓		✓	✓
Up.time [120]				✓					✓		✓	✓
Cloudfloor [121]				✓					✓		✓	✓
CloudCruiser [122]				✓					✓		✓	✓
Boundary [136]				✓					✓		✓	✓
New Relic [140]				✓					✓		✓	✓

Therefore, monitoring systems must be refined and adapted to different situations in environments of large scale and highly dynamic like Clouds.

In Section 5 we analyzed in detail the main properties and the related issues that monitoring systems have to face in order to be deployed on a Cloud. As shown, most of these issues have received attention from the research community and important results have been reached. However, some of them still require considerable effort to achieve the maturity level necessary for their seamless integration in such a complex infrastructure. In the following, we firstly discuss these properties and issues, grouping them in two macro-categories: effectiveness and efficiency. Then, we put forward a set of challenges that, in our view, Cloud monitoring systems will

have to face in the next future, indicating possible future research directions in Cloud monitoring.

7.1. Effectiveness

Main open issues reside in the possibility to have a clear view of the Cloud and to pinpoint the original causes of the observed phenomena. To achieve this, improvements are needed in terms of: (i) custom algorithms and techniques that provide effective summaries, filtering and correlating information coming from different probes; (ii) root cause analysis techniques able to derive the causes of the observed phenomena, spotting the right thread in the complex fabric of the Cloud infrastructure; and (iii) very importantly, accurate measures in an environment dominated by virtualized resources. In Section 5 we described different contributions on this topic (e.g., [34,67,89]). However, we believe that the Cloud complexity requires more effort in each of these three research areas (see e.g., [123] for similar issues on 3G network monitoring).

As the monitoring system has become a strategic subsystem for Cloud environments, its resilience is to be considered a fundamental property. On this topic, the analysis of the literature highlighted important contributions focused on resilience to faults and to VM migration and reconfiguration (e.g., [35,71]). Building on this, we believe that more effort is required in order for current Cloud monitoring systems to be also reliable.

Even if implicitly addressed in Scalability and Adaptability issues, Timeliness in itself is explicitly considered and evaluated only in [33]. This is a fundamental property that can be effectively used to quantitatively evaluate a Cloud monitoring system and objectively compare it with alternatives (e.g., by defining a specific kind of monitored event and measuring the time needed for the information to reach the managing application). Future proposals and comparisons of Cloud monitoring systems should include the use of the related metric, *Time to Insight*, and further research is needed in this field to model the relations among the parameters involved in Timeliness.

Similar considerations can be made about the property of Availability of a monitoring system: though it is closely related with Scalability and Reliability, to the best of our knowledge there are no evaluations in terms of percentage of missed events, unanswered queries and similar failures in employing the monitoring subsystem and no explicit design constraints in ensuring a given level of availability (possibly 100%, as monitoring is a critical feature). The implications in terms of costs of obtaining less than 100% availability should be considered and assessed as well.

7.2. Efficiency

Referring to the issues reported in Section 5, main improvements in terms of efficiency are expected for data management. In particular, algorithms and techniques more and more efficient are needed to manage the large volume of monitoring data necessary to have a comprehensive view of the Cloud, quickly and continuously, and without putting too much burden on the Cloud and monitoring infrastructures both in terms of computing and communication resources. The monitoring system should be therefore able to do several operations on data (collect, filter, aggregate, correlate, dissect, store, etc.) respecting strict requirements in terms of time, computational power, and communication overhead. These requirements become more and more strict with the increasing spread of Cloud Computing and therefore, the increasing number of users and resources.

Besides the improvements reported above, in the next future we foresee different possible research directions for Cloud monitoring. They are detailed in the following.

7.3. New monitoring techniques and tools

Effective monitoring techniques should be able to provide, on the one hand, very fine grained measures, and, on the other hand, a synthetic outlook of the Cloud, involving all the variables affecting the QoS and other requirements. At the same time, the techniques should not add performance burden to the system (think, for example, to mobile Cloud). Finally, they should be integrated with a control methodology that manages performance of the enterprise system. For all these reasons, new monitoring techniques and tools specifically designed for Cloud Computing are needed.

7.4. Cross-layer monitoring

The complex structure of Cloud is made of several layers to allow for functional separation, modularity and thus manageability. However, such strong layering poses several limits on the monitoring system, in terms of kinds of analysis and consequent actions that can be performed. These limits include the inability for Consumers to access lower-layer metrics and for Providers to access upper-layer ones. As a consequence, Consumers and Providers make their decisions based on a limited horizon. Overcoming this limitation is very challenging, technology-, privacy- and administration-wise.

7.5. Cross-domain monitoring: Federated Clouds, Hybrid Clouds, multi-tenancy services

Cloud Service Providers offer different types of resources and levels of QoS that can be exploited by cross-domain solutions to improve resource utilization, end-to-end performance, and resiliency. When standardized, the collaboration across multiple Cloud infrastructures is referred to as *resource federation*; however, such standardization process is still at an early stage [134,135]. Among different Cloud monitoring infrastructures there is a high heterogeneity of systems, tools, and exchanged information, and monitoring of Federated Clouds is part of ongoing research [147]. Security considerations add to these standardization issues: whenever domain boundaries are crossed, monitoring activities are challenged with security limits that can be enforced between different Cloud infrastructures (Federated Clouds), between private and public Cloud (Hybrid Clouds), or among different tenants (multi-tenant services). Research in the field of security has focused on cross-domain data leakage and its prevention, where the ability to

monitor services performance has been considered as a security risk and monitoring is an attack tool [150], and not for its potential value in evaluating and predicting the performance of a given service. As a consequence, obtaining a comprehensive monitoring solution for cross-domain solutions still represents a challenging task and it has not been properly addressed in literature yet.

7.6. Monitoring of novel network architectures based on Cloud

As reported at [143], *Cloud-based networking* is a new way to roll out distributed enterprise networks, via highly resilient, multi-tenant applications that require no capital investment in networking equipment. Unlike traditional hardware-based legacy solutions, Cloud-based networking is extremely simple, enabling enterprises to deploy remote locations in a short time and operate their distributed networks via a Cloud-based application, while providing high levels of centralized control (thanks to protocols like OpenFlow [144]) and network visibility. One of the most used platforms for Cloud-based networks is OpenStack [107] with one of its related projects called Quantum [145]. Several big players, like Cisco and Juniper, are interested and are working for integrating Cloud-based networks in their legacy networks. They are planning to use Software Defined Networks [146], based on Open Flows, to implement and integrate Cloud-based networks. Other networking approaches, such as Information-Centric Networking have been also proposed as enabling technology for Cloud management [148]. As a consequence, monitoring solutions should be adapted and improved to measure and control these new network scenarios.

7.7. Workload generators for Cloud scenarios

In Section 5 we discussed the issues and the literature related to test configuration and, in particular, to workload modeling and generation. This analysis evidenced that while different contributions have been provided in terms of studies of real and synthetic workloads, an important remaining challenge is that of workload generators specifically designed for Cloud scenarios (see e.g. [124] for emerging networking scenarios).

7.8. Energy and cost efficient monitoring

Monitoring activities can be highly demanding in terms of computing and communication resources, and therefore in terms of energy and cost. Another important challenge for next generation Cloud monitoring systems is that of performing monitoring activities satisfying their basic requirements (Accuracy, Completeness, Reliability, etc.), but minimizing the related energy consumption and cost.

7.9. Standard and common testbeds and practices

In literature, it is very difficult to find standards on procedure, format, and metrics for Cloud monitoring. In our opinion, an effort should be made in this direction. For example, Open Cirrus [125] is an open Cloud Computing research testbed to support research into the design, provisioning, and management of services at a global, multi-datacenter scale. The open nature of the testbed is designed to encourage research into all aspects of service and datacenter management. The collaborative use of research facilities provides ways to share tools, lessons learned and best practices, and ways to benchmark and compare alternative approaches for Cloud monitoring. To foster the progress of the state of the art, open platforms for fair comparison and experimentations of Cloud monitoring tools and techniques are needed.

8. Conclusion

In this paper we have provided a careful analysis of the state of the art in the field of Cloud monitoring. Fig. 2 shows a taxonomy containing a quick snapshot of the main aspects we have considered in this paper. In more detail, we have discussed the main activities in Cloud environment that have strong benefit from or actual need of monitoring. To contextualize and study Cloud monitoring, we have provided background and definitions for key concepts. We have also derived the main properties that Cloud monitoring systems should have, the issues arising from these properties, and the related contributions provided in literature so far. We have then described the main platforms (both commercial and open source) and services for Cloud monitoring, indicating how they relate with such properties and issues. Finally, we have discussed the open issues, challenges and future directions in the field of Cloud monitoring.

Acknowledgements

This work has been partially funded by PLATINO (PON01_01007) by MIUR, by "SMART HEALTH CLUSTER OSDH - SMART FSE - STAYWELL" (PON04a2_C) project by MIUR, by "Un sistema elettronico di elaborazione in tempo reale per l'estrazione di informazioni da video ad alta risoluzione, alto frame rate e basso rapporto segnale rumore" Project of the F.A.R.O. Programme, and Google Faculty Award for the UBICA project. We thank the Editor and the anonymous reviewers for the valuable comments that helped improving the manuscript.

References

[1] P. Mell, T. Grance, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, 2011. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, A view of cloud computing, Commun. ACM 53 (4) (2010) 50–58.
[3] Gurdev Singh, Shanu Sood, Amit Sharma, CM-measurement facets for cloud performance, International Journal of Computer Applications 23 (3) (2011).
[4] A. Ben Letaifa, A. Haji, M. Jebalia, S. Tabbane, State of the art and research challenges of new services architecture technologies: virtualization, SOA and cloud computing, International Journal of Grid and Distributed Computing 3 (4) (2010).
[5] C. Gong, J. Liu, Q. Zhang, H. Chen, Z. Gong, The characteristics of cloud computing, in: 39th International Conference on Parallel Processing Workshops (ICPPW), 2010, pp. 275–279.
[6] S. Zhang, S. Zhang, X. Chen, X. Huo, Cloud computing research and development trend, ICFN '10 Second International Conference on Future Networks, 93–97 (2010) 22–24.

[7] M. Ahmed, A. Sina, Md.R. Chowdhury, M. Ahmed, Md.M.H. Rafee, An advanced survey on cloud computing and state-of-the-art research issues, IJCSI 9 (1) (2012).

[8] B.P. Rimal, E. Choi, I. Lumb, A taxonomy and survey of cloud computing systems, in: NCM'09. Fifth International Joint Conference on INC, IMS and IDC, 2009, pp. 44–51.

[9] I. Foster, Y. Zhao, I. Raicu, S. Lu, Cloud computing and grid computing 360-degree compared, in: Grid Computing Environments, Workshop, 2008. GCE'08, 2008, pp. 1–10.

[10] L. Atzori, F. Granelli, A. Pescapè, A network-oriented survey and open issues in cloud computing, in: Cloud Computing: Methodology, System, and Applications, CRC, Taylor & Francis Group, 2011.

[11] N.M. Chowdhury, R. Boutaba, A survey of network virtualization, Computer Networks 54 (5) (2010) 862–876.

[12] S.N.T. Chiueh, A Survey on Virtualization Technologies, RPE Report, 2005, pp. 1–42.

[13] Dimitrios Zissis, Dimitrios Lekkas, Addressing cloud computing security issues, Future Generation Computer Systems 28 (3) (2012) 583–592.

[14] Md. Tanzim Khorshed, A.B.M. Shawkat Ali, Saleh A. Wasimi, A survey on gaps threat remediation challenges and some thoughts for proactive attack detection in cloud computing, Future Generation Computer Systems 28 (6) (2012) 833–851.

[15] J. Yang, Z. Chen, Cloud computing research and security issues, in: International Conference on Computational Intelligence and Software Engineering (CiSE), 2010, pp. 1–3.

[16] R. Choubey, R. Dubey, J. Bhattacharjee, A survey on cloud computing security, challenges and threats, International Journal 3 (2011).

[17] K. Ren, C. Wang, Q. Wang, Security challenges for the public cloud, IEEE Internet Computing 16 (1) (2012) 69–73.

[18] A. Goyal, S. Dadizadeh, A survey on cloud computing, University of British Columbia Technical Report for CS 508, 2009, pp. 55–58.

[19] S. Srinivasamurthy, D.Q. Liu, Survey on Cloud Computing Security, 2010.

[20] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, D. Leaf, NIST Cloud Computing Reference Architecture NIST Special Publication 500-292, 2011.

[21] S. Woodward, Cloud Computing Measurement, IFP Users Group, 2012.

[22] D.A. Menascé, V.A.F. Almeida, Capacity Planning for Web Services: Metrics, Models, and Methods, Prentice Hall, 2002.

[23] P. Hasselmeyer, N. d'Heureuse, Towards holistic multi-tenant monitoring for virtual data centers, in: Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP, 2010, pp. 350–356.

[24] J. Shao, Q. Wang, A performance guarantee approach for cloud applications based on monitoring, in: Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual, 2011, pp. 25–30.

[25] G. Katsaros, R. Kübert, G. Gallizo, Building a service-oriented monitoring framework with REST and nagios, in: 2011 IEEE International Conference on Services Computing (SCC), 2011, pp. 426–431.

[26] A. Viratanapanu, A.K.A. Hamid, Y. Kawahara, T. Asami, On demand fine grain resource monitoring system for server consolidation, in: Kaleidoscope: Beyond the Internet? – Innovations for Future Networks and Services, 2010 ITU-T, IEEE, 2010, pp. 1–8.

[27] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini, E. Turrini, QoS–Aware clouds, in: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010, pp. 321–328.

[28] Jin Shao, Hao Wei, Qianxiang Wang, Hong Mei, A runtime model based monitoring approach for cloud, in: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010, pp. 313–320.

[29] G.T. Lakshmanan, P. Keyser, A. Slominski, F. Curbera, R. Khalaf, A business centric end-to-end monitoring approach for service composites, in: 2010 IEEE International Conference on Services Computing (SCC), 2010, pp. 409–416.

[30] H.T. Dinh, C. Lee, D. Niyato, P. Wang, A survey of mobile cloud computing: Architecture,applications and approaches, Wireless Communications and Mobile Computing (2011). p. n/a.

[31] J.S. Park, H.C. Yu, K.S. Chung, E.Y. Lee, Markov chain based monitoring service for fault tolerance in mobile cloud computing, in: 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA), 2011, pp. 520–525.

[32] S. Padhy, D. Kreutz, A. Casimiro, M. Pasin, Trustworthy and resilient monitoring system for cloud infrastructures, in: Proceedings of the Workshop on Posters and Demos Track, Ser. PDT '11, ACM, New York, NY, USA, 2011.

[33] C. Wang, K. Schwan, V. Talwar, G. Eisenhauer, L. Hu, M. Wolf, A flexible architecture integrating monitoring and analytics for managing large-scale data centers, in: Proceedings of ICAC, 2011.

[34] M. Rak, S. Venticinque, T. Mahr, G. Echevarria, G. Esnal, Cloud application monitoring: the mOSAIC approach, in: 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, 2011, pp. 758–763.

[35] P. Massonet, S. Naqvi, C. Ponsard, J. Latanicki, B. Rochwerger, M. Villari, A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures, in: 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011, pp. 1510–1517.

[36] A. Khurshid, A. Al-Nayeem, I. Gupta, Performance evaluation of the illinois cloud computing testbed, Unpublished, Tech. Rep., 2009.

[37] L. Romano, D.D. Mari, Z. Jerzak, C. Fetzer, A novel approach to QoS monitoring in the cloud, in: International Conference on Data Compression, Communications and Processing, vol. 0, 2011, pp. 45–51.

[38] M. Armbrust et al., Above the clouds: a berkeley view of cloud computing. EECS Department, UCB, Tech. Rep. UCB/EECS-2009-28, 2009.

[39] J. Schad, J. Dittrich, J.A.Q. Ruiz, Runtime measurements in the cloud: observing, analyzing, and reducing variance, in: Proc. VLDB Endow., vol. 3(1–2), 2010, pp. 460–471.

[40] Y. Chen, V. Paxson, R. H. Katz, What's New about Cloud Computing Security? Technical Report No. UCB/EECS-2010-5, 2010.

[41] J. Spring, Monitoring cloud computing by layer, Part 1, IEEE Security & Privacy 9 (2) (2011) 66–68.

[42] J. Spring, Monitoring cloud computing by layer, Part 2, IEEE Security & Privacy 9 (3) (2011) 52–55.

[43] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, et al., The RESERVOIR model and architecture for open federated cloud computing, IBM Journal of Research and Development 53 (4) (2009).

[44] M. Kutare, G. Eisenhauer, C. Wang, K. Schwan, V. Talwar, M. Wolf, Monalytics: online monitoring and analytics for managing large scale data centers, in: Proceedings of the 7th International Conference on Autonomic Computing, Ser. ICAC '10, ACM, New York, NY, USA, 2010, pp. 141–150.

[45] V. Kumar, Z. Cai, B.F. Cooper, G. Eisenhauer, K. Schwan, M. Mansour, B. Seshasayee, P. Widener, Implementing diverse messaging models with self-managing properties using IFLOW, in: ICAC, 2006.

[46] Frédéric Desprez, Eddy Caron, Luis Rodero-Merino, Adrian Muresan, Auto-scaling, load balancing and monitoring in commercial and open-source clouds, in: Cloud Computing: Methodology, System and Applications, CRC Press, 2011.

[47] Y. Mei, L. Liu, X. Pu, S. Sivathanu, Performance measurements and analysis of network I/O applications in virtualized cloud, in: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010, pp. 59–66.

[48] http://www.hyperic.com/products/cloud-status-monitoring.

[49] R.P. Karrer, I. Matyasovszki, A. Botta, A. Pescapè, MagNets – experiences from deploying a joint research-operational next-generation wireless access network testbed, in: Proceedings of International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM), May 2007, Orlando, Florida (USA), to appear.

[50] M. Bernaschi, F. Cacace, A. Pescapè, S. Za, Analysis and experimentation over heterogeneous wireless networks, in: First IEEE International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (TRIDENTCOM'05) – ISBN 0-7695-2219-X/05, February 2005, Trento (Italy), 2005, pp. 182–191.

[51] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, A. Pescapè, Broadband internet performance: a view from the gateway, in: ACM SIGCOMM 2011 Proceedings, Toronto, ON, Canada, August 15–19, 2011.

[52] A. Botta, A. Pescapè, G. Ventre, Quality of service statistics over heterogeneous networks: analysis and applications, in: Special Issue of Elesevier EJOR on 'Performance Evaluation of QoS-aware Heterogeneous Systems', vol 191(3), 2008, pp. 1075–1088.

[53] Vinod Venkataraman, Ankit Shah, Yin Zhang, Network-Based Measurements on Cloud Computing Services. <http://www.cs.utexas.edu/~vinodv/files/cc-measure.pdf>.

[54] Security Guidance for Critical Areas of Focus in Cloud Computing v2.1, Cloud Security Alliance, 2009. <www.cloudsecurityalliance.org/csaguide.pdf>.
[55] S. Clayman, A. Galis, L. Mamatas, Monitoring virtual networks with lattice, in: Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP, 2010, pp. 239–246.
[56] F. Azmandian, M. Moffie, J.G. Dy, J.A. Aslam, D.R. Kaeli, Workload characterization at the virtualization layer, in: 2011 IEEE 19th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 25–27 July 2011, pp. 63–72.
[57] S. Clayman, R. Clegg, L. Mamatas, G. Pavlou, A. Galis, Monitoring, aggregation and filtering for efficient management of virtual networks, in: Proceedings of the 7th International Conference on Network and Services Management, 2011, pp. 234–240.
[58] R. Mehrotra, A. Dubey, S. Abdelwahed, W. Monceaux, Large scale monitoring and online analysis in a distributed virtualized environment, in: 2011 Eighth IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems, 2011, pp. 1–9.
[59] M.L. Massie, B.N. Chun, D.E. Culler, The ganglia distributed monitoring system: design, implementation and experience, Parallel Computing 30 (2004).
[60] Nagios. <http://www.nagios.org/>.
[61] M.B. de Carvalho, L.Z. Granville, Incorporating Virtualization Awareness in Service Monitoring Systems, 2011.
[62] Rizwan Mian, Patrick Martin, Jose Luis Vazquez-Poletti, Provisioning data analytic workloads in a cloud, Future Generation Computer Systems (2012).
[63] Waheed Iqbal, Matthew N. Dailey, David Carrera, Paul Janecek, Adaptive resource provisioning for read intensive multi-tier applications in the cloud, Future Generation Computer Systems 27 (6) (2011) 871–879. ISSN 0167-739X, 10.1016/j.future.2010.10.016.
[64] Vincent C. Emeakaroha, Marco A.S. Netto, Rodrigo N. Calheiros, Ivona Brandic, Rajkumar Buyya, César A.F. De Rose, Towards autonomic detection of SLA violations in cloud infrastructures, Future Generation Computer Systems (2012).
[65] A. Ayad, U. Dippel, Agent-based monitoring of virtual machines, in: 2010 International Symposium in Information Technology (ITSim), vol. 1, 2010, pp. 1–6.
[66] C. Vecchiola, X. Chu, R. Buyya, Aneka: a software platform for NET-based cloud computing, in: W. Gentzsch, L. Grandinetti, G. Joubert (Eds.), High Speed and Large Scale Scientific Computing, IOS, 2009, pp. 267–295.
[67] Z. Hill, M. Humphrey, A quantitative analysis of high performance computing with Amazon's EC2 infrastructure: the death of the local cluster? in: 2009 10th IEEE/ACM International Conference on Grid Computing, 13–15 October 2009, pp. 26–33.
[68] G. Wang, T.S.E. Ng, The impact of virtualization on network performance of amazon ec2 data center, in: 2010 Proceedings of IEEE INFOCOM, 2010, pp. 1–9.
[69] Jean-Claude Laprie, From dependability to resilience, in: IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2008.
[70] Robert W. Shirey, Internet security glossary, Internet Engineering Task Force RFC 4949 Informational, 2007.
[71] G. Xiang, H. Jin, D. Zou, X. Zhang, S. Wen, F. Zhao, VMDriver: a driver-based monitoring mechanism for virtualization, in: 2010 29th IEEE Symposium on Reliable Distributed Systems, 2010, pp. 72–81.
[72] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, D.H.J. Epema, A performance analysis of EC2 cloud computing services for scientific computing, in: Cloud Computing, Springer, 2010, pp. 115–131.
[73] The SRT-15 Project. <www.73.eu>.
[74] A. Bessani, From byzantine fault tolerance to intrusion tolerance (A Position Paper), in: 5th Workshop on Recent Advances in Intrusion-Tolerant Systems (WRAITS), DSN, 2011.
[75] C. Vecchiola, S. Candey, R. Buyya, High-performance cloud computing: a view of scientific applications, in: 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN), 14–16 December 2009, pp. 4–16.
[76] Edward Walker, Benchmarking amazon EC2 for high-performance scientific computing, in: LOGIN, vol. 33(5), 2008, pp. 18–23.
[77] A. Ganapathi, Yanpei Chen, A. Fox, R. Katz, D. Patterson, Statistics-driven workload modeling for the Cloud, in: 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW), 1–6 March 2010, pp. 87–92.
[78] Y. Chen, A.S. Ganapathi, R. Griffith, R.H. Katz, Towards understanding cloud performance tradeoffs using statistical workload analysis and replay, University of California at Berkeley, Technical, Report No. UCB/EECS-2010-81, 2010.
[79] Hongzhang Shan, Katie Antypas, John Shalf, Characterizing and predicting the I/O performance of HPC applications using a parameterized synthetic benchmark, in: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing (SC '08).
[80] Donald Kossmann, Tim Kraska, Simon Loesing, An evaluation of alternative architectures for transaction processing in the cloud, in: Proceedings of the 2010 International Conference on Management of Data (SIGMOD '10), ACM, New York, NY, USA, 2010, pp. 579–590.
[81] Jiang Dejun, Guillaume Pierre, Chi-Hung Chi, EC2 performance analysis for resource provisioning of service-oriented applications, in: Proceedings of the 2009 International Conference on Service-Oriented, Computing (ICSOC/ServiceWave'09).
[82] Devarshi Ghoshal, Richard Shane Canon, Lavanya Ramakrishnan, I/O performance of virtualized cloud environments, in: Proceedings of the Second International Workshop on Data Intensive Computing in the Clouds (DataCloud-SC '11).
[83] Carsten Binnig, Donald Kossmann, Tim Kraska, Simon Loesing, How is the weather tomorrow? Towards a benchmark for the cloud, in: Proceedings of the Second International Workshop on Testing Database Systems (DBTest '09), ACM, New York, NY, USA, 2009.
[84] TPC. TPC-H Benchmark 2.8, 2008. <http://www.tpc.org/tpch/>
[85] TPC. TPC-C Benchmark 5.10.1, 2009. <http://www.tpc.org/tpch/>
[86] TPC. TPC-W Benchmark 1.8, 2002. <http://www.tpc.org/tpcw/>
[87] Ahmed Abujoda, Network measurements in virtualized networks and its challenges, in: 6th GI/ITG KuVS Workshop on Future Internet, 22 November 2010 in Hannover, Germany.
[88] I.M. Rafika, N. Sadeque, JA Andersson, A. Johnsson, Time-stamping accuracy in virtualized environments, in: 2011 13th International Conference on Advanced Communication Technology (ICACT), 2011, pp. 475–480.
[89] J. Whiteaker, F. Schneider, R. Teixeira, Explaining packet delays under virtualization, ACM SIGCOMM Computer Communication Review 41 (1) (2011).
[90] L. Youseff, K. Seymour, H. You, J. Dongarra, R. Wolski, The impact of paravirtualized memory hierarchy on linear algebra computational kernels and software, in: Proceedings of the 17th International Symposium on High Performance, Distributed Computing, 2008, pp. 141–152.
[91] H. Newman, I. Legrand, P. Galvez, R. Voicu, C. Cirstoiu, MonALISA: a distributed monitoring service architecture, CHEP03, La Jolla, California, 2003.
[92] A. Cooke, A.J.G. Gray, L. Ma, W. Nuttetal, R-GMA: an information integration system for grid monitoring, in: Proceedings of the 11th International Conference on Cooperative, Information Systems, 2003, pp. 462–481.
[93] S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G.L. Rubini, G. Tortone, M.C. Vistoli, GridICE: a monitoring service for grid systems, Future Generation Computer Systems 21 (4) (2005) 559–571.
[94] S. Zanikolas, R. Sakellariou, A taxonomy of grid monitoring systems, Future Generation Computer Systems 21 (1) (2005) 163–188.
[95] http://awsdocs.s3.amazonaws.com/AmazonCloudWatch/latest/acw-dg.pdf.
[96] http://www.cloudkick.com/home.
[97] http://www.nimsoft.com/solutions/nimsoft-monitor/cloud.
[98] A. V. Dastjerdi, S. G. H. Tabatabaei, R. Buyya, A dependency-aware ontology-based approach for deploying service level agreement monitoring services in cloud, Software: Practice and Experience 42 (2012) 501–518.
[99] http://portal.monitis.com/.
[100] http://www.logicmonitor.com/monitoring/storage/netapp-filers/.
[101] http://www.manjrasoft.com/.
[102] Rodrigo N. Calheiros, Christian Vecchiola, Dileban Karunamoorthy, Rajkumar Buyya, The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid clouds, Future Generation Computer Systems 28 (6) (2012) 861–870.
[103] http://www.infolinux.it/utility/eucalyptus-il-cloud-open-source/.
[104] http://nagios.sourceforge.net/docs/nagioscore-3-en.pdf.
[105] http://opennebula.org/documentation:archives:rel2.0:img.
[106] http://en.wikipedia.org/wiki/OpenNebula.
[107] http://docs.openstack.org/diablo/openstack-compute/admin/os-compute-adminguide-trunk.pdf.
[108] http://www.cloudstack.org/.
[109] https://github.com/zenoss/ZenPacks.zenoss.CloudStack.
[110] http://www.nimbusproject.org/.

[111] S.A. Chaves, R.B. Uriarte, C.B. Westphall, Toward an architecture for monitoring private clouds, IEEE Communications Magazine 49 (2011) 130–137.
[112] https://cloudsleuth.net/.
[113] http://cloudharmony.com/.
[114] http://radlab.cs.berkeley.edu/wiki/Projects/Cloudstone.
[115] W. Sobel et al., Cloudstone: multi-platform multi-language benchmark and measurement tools for Web 2.0, in: Proceedings of the Cloud Computing and Its Applications, 2008.
[116] http://cloudcmp.net/.
[117] A. Li, X. Yang, S. Kandula, M. Zhang, CloudCmp: comparing public cloud providers, in: Proceedings of the 10th Annual Conference on Internet Measurement, IMC '10, ACM, New York, NY, USA, 2010, pp. 1–14.
[118] http://www.cloudclimate.com.
[119] http://www.cloudyn.com/.
[120] http://www.uptimesoftware.com/cloud-monitoring.php.
[121] http://cloudcruiser.com/.
[122] http://cloudfloor.com/.
[123] A. Botta, A. Pescapè, C. Guerrini, M. Mangri, A customer service assurance platform for mobile broadband networks, IEEE Communications Magazine 49 (10) (2011) 101–109.
[124] A. Dainotti, A. Botta, A. Pescapè, A tool for the generation of realistic network workload for emerging networking scenarios, Computer Networks (2012).
[125] https://opencirrus.org/.
[126] B. Kitchenham, Procedures for Performing Systematic Reviews, Keele University Technical, Report TR/SE-0401, 2004.
[127] http://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-pubs.html.
[128] A. Corradi, L. Foschini, J. Povedano-Molina, J.M. Lopez-Soler, "DDS-enabled Cloud management support for fast task offloading," Computers and Communications (ISCC), 2012 IEEE Symposium on, 1–4 July 2012, pp. 67–74.
[129] http://www.gwos.com/features/.
[130] P. Samimi, A. Patel, Review of pricing models for grid & cloud computing, in: 2011 IEEE Symposium on Computers & Informatics (ISCI), IEEE, 2011, pp. 634–639.
[131] I. Foster, Y. Zhao, I. Raicu, S. Lu, Cloud computing and grid computing 360-degree compared, in: Grid Computing Environments Workshop, 2008. GCE039;08, IEEE, 2008, pp. 1–10.
[132] I. Foster, C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2004.
[133] R. VBuyya, Chee Shin Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems 25 (6) (2009) 99–616.
[134] http://www.opencloudware.org/.
[135] http://www.compatibleone.org/.
[136] https://boundary.com/.
[137] http://www.paraleap.com/azurewatch.
[138] http://sourceforge.net/projects/hyperic-hq/.
[139] http://www.sonian.com/cloud-monitoring-sensu/.
[140] http://newrelic.com/enterprise/visibility.
[141] J. Du, N. Sehrawat, W. Zwaenepoel, Performance profiling of virtual machines, in: 7th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE '11), ACM, New York, NY, USA, 2011, pp. 3–14.
[142] A. M. Aljohani, D.R.W. Holton, I.-U. Awan, J.S. Alanazi, Performance evaluation of local and cloud deployment of web clusters, in: NBiS 2011, pp. 274–278.
[143] http://en.wikipedia.org/wiki/Cloud-based_networking.
[144] http://www.openflow.org/.
[145] http://wiki.openstack.org/Quantum.
[146] http://en.wikipedia.org/wiki/Software_Defined_Networking.
[147] M. Villari, I. Brandic, and F. Tusa (Eds.), Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice, (FSCI11) IGi Global, 2012, pp. 1–17.
[148] B. Ohlman, A. Eriksson, R. Rembarz, What networking of information can do for cloud computing, in: IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'09), 2009, pp. 78–83.
[149] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Muñoz, G. Tofetti, Reservoir – when one cloud is not enough, Computer 44 (3) (2011) 44–51.
[150] T. Ristenpart, E. Tromer, H. Shacham, S. Savage, Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds, in: Proceedings of the 16th ACM Conference on Computer and Communications, Security, 2009, pp. 199–212.

Giuseppe Aceto is a Ph.D. student in Electronic and Telecommunications Engineering at the Department of Electrical Engineering and information Technologies of the University of Napoli Federico II (Italy), where he received his M.S. degree in Telecommunications Engineering in 2008, defending a thesis about a unified platform for available bandwidth estimation in heterogeneous IP networks. He is a junior researcher for the same Department. His research interests are focused on networking, more specifically on network measurements and traffic analysis. Giuseppe Aceto is coauthor of papers on international journals (ACM Performance Evaluation Review, Elsevier Journal of Network and Computer Applications) and international conferences (IEEE International Workshop on Measurements & Networking, IEEE INFOCOM 2010, IEEE Symposium on Computer and Communications). In 2010 he was awarded with the best local paper award at IEEE ISCC 2010.

Alessio Botta is a postdoc at the Department of Electrical Engineering and Information Technologies of the University of Napoli Federico II (Italy). He graduated in Telecommunications Engineering (M.S.) and obtained the Ph.D. in Computer Engineering and Systems, both at University of Napoli Federico II. His research interests are in the area of networking and, in particular, in the area of network performance measurement and improvement, with a specific focus on wireless and heterogeneous systems. Alessio Botta has coauthored more than 40 international journal (IEEE Communications Magazine, IEEE Transactions on Parallel and Distributed Systems, Elsevier Computer Networks, etc.) and conference (IEEE Globecom, IEEE ICC, IEEE ISCC, etc.) publications. He has served and serves several technical program committees of several international conferences (IEEE Globecom, IEEE ICC, etc.) and he acts as reviewer for different international conferences (IEEE Infocom, etc.) and journals (IEEE Transactions on Mobile Computing, IEEE Network, IEEE Transactions on Vehicular Technology, etc.) in the area of networking. In 2010 he was awarded with the best local paper award at IEEE ISCC 2010.

Walter de Donato is a postdoc at the Department of Electrical Engineering and Information Technologies of the University of Napoli Federico II (Italy). He received a M.S. degree in Computer Engineering and a Ph.D. in Computer Engineering from the same University. His research activity mainly concerns methodologies, techniques, and distributed architectures for measuring, analyzing, classifying, and monitoring network traffic and also covers the following topics: network topology discovery and mapping, Linux-based embedded systems, network processor architectures, and content distribution networks. Walter de Donato has coauthored several international conference (ACM Sigcomm, PAM, IEEE Globecom, etc.) publications. He served as reviewer for several international conferences (IEEE Globecom, IEEE ICC, etc.) and journals (Computer Networks, etc.) in the area of networking. In 2011 he was awarded with the TEA (Technologybiz Endorsement Award).

Antonio Pescapè is an Assistant Professor at the Department of Electrical Engineering and Information Technologies of the University of Napoli Federico II (Italy) and Honorary Visiting Senior Research Fellow at the School of Computing, Informatics and Media of the University of Bradford (UK). He received the M.S. Laurea Degree in Computer Engineering and the Ph.D. in Computer Engineering and Systems, both at University of Napoli Federico II. Antonio Pescapè teaches courses in Computer Networks, Computer Architectures, Programming, and Multimedia and he has also supervised and graduated more than 130 among B.S., M.S., and Ph.D. students. His research interests are in the networking field with focus on Internet Monitoring, Measurements and Management and on Network Security. Antonio Pescapè has coauthored over 130 journal (IEEE Communications Magazine, JSAC, IEEE Wireless Communications Magazine, IEEE Networks, etc.) and conference (SIGCOMM, IMC, PAM, Globecom, ICC, etc.) publications and he is co-author of several patents pending. He has served and 23 serves on more than 150 technical program committees of IEEE and ACM conferences. He has served as Editorial Board Member of IEEE Survey and Tutorials (2007–2010) and was guest editor for the special issue of Computer Networks on "Traffic classification and its applications to modern networks". For his research activities he has received several awards. In 2009 he was awarded the IET Communications Premium Award 2009; in 2010 he was awarded the best local paper award at IEEE ISCC 2010; in november 2011 he was awarded the TEA (Technologybiz Endorsement Award); he was awarded by Open Source Software World Challenge for the D-ITG platform in 2011 and for the TIE platform in 2012; in 2012 two of his papers have been awarded the IRTF ANRP (Applied Networking Research Prize) and he was awarded the Best Poster award at SIGCOMM 2012; he was awarded the Google Faculty Award in 2013. He is a Senior Member of the IEEE. Finally, Antonio Pescapè has served and serves as independent reviewer/evaluator of research and implementation projects and project proposals co-funded by the Swedish government, several Italian local governments, Italian Ministry for University and Research (MIUR) and Italian Ministry of Economic Development (MISE).