A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives

ARTICLE *in* JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING · OCTOBER 2014

Impact Factor: 1.01 · DOI: 10.1016/j.jpdc.2014.06.007

5 AUTHORS, INCLUDING:

Vincent C. Emeakaroha
University College Cork
26 PUBLICATIONS **214** CITATIONS
SEE PROFILE

Philip Healy
University College Cork
32 PUBLICATIONS **68** CITATIONS
SEE PROFILE

John P. Morrison
University College Cork
109 PUBLICATIONS **421** CITATIONS
SEE PROFILE

Theodore Gerard Lynn
Dublin City University
55 PUBLICATIONS **19** CITATIONS
SEE PROFILE

J. Parallel Distrib. Comput. 74 (2014) 2918–2933

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives

Kaniz Fatema [a,*], Vincent C. Emeakaroha [a], Philip D. Healy [a], John P. Morrison [a], Theo Lynn [b]

[a] Irish Centre for Cloud Computing & Commerce, University College Cork, Ireland
[b] Irish Centre for Cloud Computing & Commerce, Dublin City University, Ireland

HIGHLIGHTS

- Surveyed monitoring tools revealing common characteristics and distinctions.
- Identified practical capabilities of monitoring tools.
- Presented taxonomy of monitoring capabilities.
- Analysed strengths and weakness of monitoring tools based on taxonomy.
- Discussed challenges and identified future research trends in Cloud monitoring.

ARTICLE INFO

Article history:
Received 17 September 2013
Received in revised form
2 May 2014
Accepted 19 June 2014
Available online 5 July 2014

Keywords:
Cloud management
Monitoring tools
Cloud operational areas
Capabilities
Taxonomy
Survey

ABSTRACT

The efficient management of Cloud infrastructure and deployments is a topic that is currently attracting significant interest. Complex Cloud deployments can result in an intricate layered structure. Understanding the behaviour of these hierarchical systems and how to manage them optimally are challenging tasks that can be facilitated by pervasive monitoring. Monitoring tools and techniques have an important role to play in this area by gathering the information required to make informed decisions. A broad variety of monitoring tools are available, from general-purpose infrastructure monitoring tools that predate Cloud computing, to high-level application monitoring services that are themselves hosted in the Cloud. Surveying the capabilities of monitoring tools can identify the fitness of these tools in serving certain objectives. Monitoring tools are essential components to deal with various objectives of both Cloud providers and consumers in different Cloud operational areas. We have identified the practical capabilities that an ideal monitoring tool should possess to serve the objectives in these operational areas. Based on these identified capabilities, we present a taxonomy and analyse the monitoring tools to determine their strength and weaknesses. In conclusion, we present our reflections on the analysis, discuss challenges and identify future research trends in the area of Cloud monitoring.

1. Introduction

The emergence of Cloud Computing has ushered in a new era of Internet-based service provisioning opportunities. Cloud Computing is characterised by the provision of resources as general utilities that can be leased and released in an on-demand manner. Consequently, IT resources represent an operational rather than a capital expenditure. A broad variety of pricing models can be applied to Cloud resources, from simple fixed rental schemes to pay-as-you-go models. Monitoring techniques are indispensable in order to manage large-scale Cloud resources and enforce quality of service for consumers.

Given the multi-tenant nature of Cloud environments, efficient management in the face of quality of service and performance constraints can be a challenge. Monitoring tools have an important role to play in these areas by allowing informed decisions to be made regarding resource utilisation. Automated monitoring of physical and virtual IT resources allows for the identification and resolution of issues with availability, capacity, and other quality requirements. The benefits of automated monitoring have long been recognised, even in non-Cloud environments. The importance of monitoring has been widely addressed in the literature in

* Corresponding author.
E-mail address: k.fatema@cs.ucc.ie (K. Fatema).

http://dx.doi.org/10.1016/j.jpdc.2014.06.007
0743-7315/

various contexts, such as: system/network [13,35,71], distributed systems/Grid [93,4,95], application [47,10] and Cloud [1,33]. For Cloud environments, appropriate monitoring is crucial as usage-based billing and elastic scaling are impossible to implement in the absence of relevant metrics. Currently, a variety of Cloud monitoring tools is applied in an *ad-hoc* and non-systematic way, everywhere from low-level, general-purpose infrastructure monitoring to high-level application and service monitoring. The purpose of this paper is to comprehensively review these tools to assess whether they are adequate in satisfying the essential objectives for measuring intrinsic Cloud behaviours.

The focus of our work is to capture the evolutionary adaptation of monitoring tools' capabilities from general purpose to Cloud monitoring and to present a full capability analysis with respect to practical Cloud operational areas that would help Cloud providers and customers in making an informed choice of an appropriate monitoring tool. The monitoring platforms considered in this paper have been chosen based on literature reviews and perceived industrial acceptance.

The main contributions of this paper can be summarised as follows: (i) it surveys the range of monitoring tools currently in use to gain their technical insights, (ii) it identifies the desired capabilities of monitoring tools to serve different Cloud operational areas from both providers' and consumers' perspectives, (iii) it then presents a taxonomy of the identified capabilities, (iv) it analyses the available tools based on the identified capabilities and unveils the capabilities that are under-represented, Cloud operational areas that are currently strongly supported by those monitoring tools and the areas that need further development and, (v) it discusses future research challenges and trends in Cloud monitoring and management.

Our paper flows as follows: First, we assign the tools into categories, Cloud specific and non-Cloud specific. After studying the tools and extracting technical capabilities, we summarise these in Tables 1 and 2. We then focus on Cloud-specific monitoring capabilities and derive a taxonomy, which we present in Section 4. We then re-examine all of the tools again in light of the taxonomy in Section 5 in order to identify their strengths and weakness when applied to particular operational areas.

The remainder of this paper is organised as follows: Section 2 provides information on various computing environments, ranging from single machine to various distributed systems, and the usage of monitoring in those environments. In Section 3, available monitoring tools are identified and described. Section 4 describes taxonomy of desirable monitoring capabilities that forms the basis for the analysis of the identified monitoring tools. In Section 5, we analyse the identified monitoring tools. Section 6 presents analysis of the related work. Section 7 discusses the identified challenges while Section 8 concludes the paper and focuses on the future research trends in Cloud monitoring.

2. Background

Monitoring tools have long been used for tracking resource utilisation and the performance of systems and networks. These tools have traditionally been administrated by a single administrator in a single domain. Subsequently, the development of distributed systems like clusters forced the evolution of monitoring tools to meet the demand of these new environments. As more sophisticated distributed environments emerged, including Grids and Clouds, monitoring tools had to be developed to capture their salient characteristics. In the case of Grid these included computations that span multiple administrative domains and in the case of Clouds on-demand multi-tenant services. According to National Institute of Standards and Technology (NIST) definition [58]: *"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".*

Cloud computing has shifted computation from local machines to services accessed via the network. Services in Cloud are typically offered via three different service models which can be viewed as a layered model of services: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).

IaaS providers, such as Amazon EC2, offer virtual resources, such as machine instances, storage, and networking devices, as a service to the end user and/or the layers above, enabling self-service for virtualized resources. A virtual machine manager, or hypervisor, is required in order to make the physical resources available to virtual machine instances. Through the application of clustering techniques, multiple physical servers can be aggregated into a resource pool from which storage, CPU, memory and networking resources can be dynamically allocated to a set of VMs. Clustering of the resource pool ensures high availability in the presence of hardware failures.

PaaS providers utilise the virtual machines' environment (e.g. operating systems and tools) to provide a scalable abstractions of the underlying resources onto which applications can be deployed. For example, Google AppEngine provides developers with scalable application back end functionality for dynamic web serving, automatic scaling and load balancing [86]. The resulting abstraction is, from the developer's point of view, completely divorced from the underlying infrastructure. Similar offerings exist for Java EE (*e.g.*, Red Hat OpenShift [61]) and Ruby on Rails (*e.g.*, Engine Yard [73]), among a host of others. Some Cloud service providers, such as Microsoft Azure, offer tools and features as part of their offerings that simplify the development of PaaS services [56]. Open-source tools such as Nimbus [62] and Cloudify [32] are available that simplify PaaS integration in a provider-agnostic fashion.

SaaS (Software-as a-Service) providers offer software as a service which may hide the service implementation, thus the SaaS customer is not necessarily aware of the underlying platforms, infrastructure, or hardware.

Given the rich architecture of Cloud, effective monitoring requires an appropriate suite of tools capable of monitoring in the IaaS, PaaS and SaaS layers.

3. Review of existing monitoring systems

In the light of the above discussion, we examined the available monitoring tools and divided them into two broad categories: (i) general-purpose monitoring tools; and (ii) Cloud-specific monitoring tools. We examine each of these categories in turn in subsequent subsections to gain an understanding of their common characteristics and functionalities. The technical details, as well as its reported limitations and usage experiences are summarised in Tables 1 and 2. These tabular presentations give readers the opportunity to gain technical insights into the tools. Furthermore, the presented information also helps to identify the features of the tools, which are later used for capability based analysis and the development of a taxonomy.

3.1. General-purpose monitoring tools

Before the advent of Cloud Computing, a number of tools were already available for the purpose of monitoring diverse IT infrastructure resources such as networks and compute nodes. Some specialise in particular domains such as HPC clusters and Grids. Many of these tools continue to be developed, and could be adopted in Clouds for monitoring at various abstraction levels. In

Table 1
Feature matrix for general purpose monitoring tools.

Tool	Monitored resources	Agent language	Agent OS	Licence	Alerts	Enterprise messaging support	Reported limitations	Usage experience
Nagios [7]	System resources, network, sensors, applications and services	C	Linux/Unix (Windows via proxy agent)	Open source (GPL)	E-mail, SMS, Custom	No	Difficulty of configuration [43,70] inability to work at high sampling frequency [48, 79], inability to bind service due to VM migration [22]	Grid Infrastructure monitoring [41], VM monitoring [48,79,22]
Collectd [21]	System resources, network, sensors, databases, applications and services	C	Linux/Unix and Mac OS	Open source (GPLv2)	N/A	AMQP (above V5.1)	No visualisation platform	Gathering metrics from Cloud deployments for forensic purpose [63]
Opsview Core [52]	System resources, network, database, applications and services	Perl, C	Linux/Unix, Windows	Open source (GPL)	E-mail, SMS, Custom	No	N/A	N/A
Cacti [55]	Mainly network	PHP, C	Linux based OS and Windows	Open source (GPL)	Audible alerts, E-mail	No	N/A	Integrated development of various third-party monitoring tools [96]
Zabbix [74]	System resources, network, sensors, databases, applications and services	C, PHP, Java	Linux/Unix, Mac, Windows	Open source (GPL)	E-mail, Custom	XMPP	Auto-discovery feature of Zabbix can be inefficient. [72]	In a Cloud broker engine [8] to dynamically scale Cloud resources, in private Cloud monitoring [14].
Open NMS [77]	Mainly network	Java	Linux/Unix, Windows, Mac	Open source (GPLv3)	E-mail, SMS	JMS	Auto discovery service has limited capability and its service is not generalised for all network services [57]	N/A
Ganglia [64]	System resources	C, Perl, PHP, Python	Linux/Unix, Solaris, Windows, Mac	Open Source (BSD)	N/A	No	Difficult to customise, introduces overhead both at hosts and networks because of the multicast updates, and XML event encoding [93]	Server side data collection for a Cloud based monitoring solution [50,89] and resource provisioning in Rocks Clusters [49]
Hyperic HQ [39]	System resources, network, database, applications and services	Java	Linux/Unix, Windows, Mac	Open source (GPLv2)	E-mail, SMS	No	High memory requirement [13] and difficulty of customising graphical interface [52]	For collecting software inventory data in CloudAlloc, a monitoring and reservation system for compute clusters [42].
IBM Tivoli [40]	System resources, network, database, applications and services	Java	Linux/Unix, Windows, Mac	Commer-cial	E-mail, SMS	No	N/A	In monitoring IBM Cloud [11], in enterprise service management [16].
Kiwi Application Monitor [30]	Application processes and user activity	N/A	Windows	Open source	Custom	No	N/A	In measuring application's peak memory usage and CPU time in the simulation of a space efficient quantum computer [30].
R-OSGi [83]	Distributed applications	Java	N/A	Open source	E-mail, SMS	No	Problem with service registration and static configuration [83]	Getting dependency information for a distributed application [27]
DAMS [46]	Distributed applications	Java	N/A	Open source	N/A	No	System's performance is affected [46]	Open Education Management System of Central Radio and TV University [46].

Table 2
Cloud specific monitoring tools review.

Name of tool	Monitored resource	Port-able	Scal-able	Open source	Operating system	Alerts	Messaging system	Implementation language	Reported limitations
Amazon Cloud Watch [3]	AWS resources and applications and services running in AWS	No	Yes	No	Linux, Windows	E-mail	Amazon Simple Queue Service	Scripts in Windows PowerShell for Windows, Perl for Linux	Works for Amazon resources only, works in centralised models, does not ensure availability, potential security threat due to non-efficient use [75]
Azure Watch [5]	Azure based resources	No	Yes	No	Windows	E-mail, report to browser, smartphone, RSS feed	No	.Net	Works for Azure based resources only
Nimsoft [65]	Various Cloud resources, OS, Network and applications	Yes	Yes	No	Linux, Netware, Unix, Windows, Mac	E-mail, RSS, text messages, instant messenger, Twitter	Nimsoft Message Bus	C/C++, Java, Perl, VB, and .Net	Does not cap resource consumption by the monitor, not fault-tolerant [66], does not support SLA compliance checking for data durability or location [80]
Monitis [69]	Network, Server, Cloud resources and applications	Yes	Yes	No	Linux/Unix, Windows, Mac	E-mail, IM, SMS, Twitter, live voice	No	C++	N/A
CloudKick [38]	Cloud resources, application and protocols	Yes	No	No	Linux, Windows	E-mail, webhook, SMS	No	C#, .NET, Python, PHP, Java and Ruby	Works in centralised models and does not ensure availability of services [75], can monitor only via CloudKick Agent [54], only suitable for infrastructure monitoring [38]
Boundary Application Monitor [12]	Cloud applications	Yes	Yes	No	Linux, Windows, Mac	E-mail, SMS	No	N/A	N/A
mOSAIC [82]	Cloud applications	Yes	No	Yes	Linux based OS	SLA violation warning	AMQP	Java	It monitors the applications developed with mOSAIC API only [26]
CASViD [26]	Cloud applications	Yes	Yes	Yes	Unix, Linux	SLA violation warning	JMS	Python, Java	Does not support multi-tier application monitoring
PCMONS [23]	Cloud resources	No	No	Yes	Linux	Tool dependent	No	Python, Perl, Bash script	Works only for infrastructure monitoring in private Clouds [23]

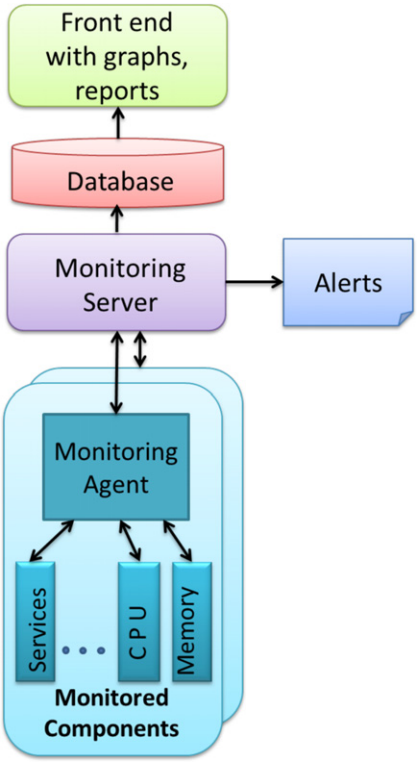


Fig. 1. Infrastructure monitoring tool architecture.

this section, we discuss the features of general purpose infrastructure and application monitoring tools.

General purpose infrastructure monitoring tools typically utilise a client–server model by installing an agent in every system to be monitored. Fig. 1 shows this general architecture. Monitoring agents measure the metric values from monitored components and send them to the monitoring server. The server stores the collected metrics into a database, analyses them, and sends alerts. It may generate graphs, trending reports and SLA reports based on the monitored metrics retrieved from the database.

As illustrated in Fig. 2, primitive metric measurements can be initiated either by the monitoring server (depending on system configuration) or it can be initiated by an external program (or script) residing on the monitored resource. The latter arrangement is useful when monitoring is performed behind a firewall. For example, active mode monitoring in Nagios [7] is initiated at regular intervals by executing the Nagios Remote Plugin Executor (NRPE) as shown in Fig. 2(a). Passive monitoring in Nagios uses the Nagios Service Check Acceptor (NSCA) and is initiated by external applications. Fig. 2(b) depicts the passive mode.

When a monitoring agent is initiated for collecting metrics, it measures the relevant metric values from the monitored components and passes them on to the monitoring server. Depending on the configuration of the system, the server may send alerts to interested parties on occurrence of an event. Most of the monitoring systems use e-mail and SMS as alerting mechanisms (*e.g.* Nagios, Opsview [52], Zabbix [74] and Open NMS [77]). Some monitoring tools, such as Cacti [55], use audible alerts. Others, such as Collectd [21] and Ganglia [64], have no alerting mechanisms.

Monitoring servers may be arranged hierarchically to provide distributed monitoring. Zabbix, IBM Tivoli [40] and Opsview adopted this approach. In this arrangement, a child server passes its monitored data to a parent server which stores them into a database, allowing them to be displayed through a web front-end interface. Fig. 3 illustrates the hierarchical architecture of the Zabbix monitor. Ganglia, on the other hand, aggregates data from all the Ganglia Monitoring Daemons (Gmond) via Ganglia Meta Daemon (Gmetad) while utilising a multicasting technique amongst the nodes in a cluster.

Few tools are available for general purpose application monitoring. The most appropriate application monitoring technique depends on the nature of the application. Kiwi Application Monitor [30] monitors centralised applications running in a single machine by inspecting OS processes. Other tools, such as Remoting-Open Services Gateway initiative (ROSGi) Development Tool (RDT) [83] and Distributed Application Monitoring System (DAMS) [46], aim to monitor distributed applications involving network communications. RDT [83] helps to develop, deploy and monitor distributed Java applications with the Eclipse IDE while DAMS [46] enhances the byte code of distributed Java applications and monitors the application module and class methods at runtime.

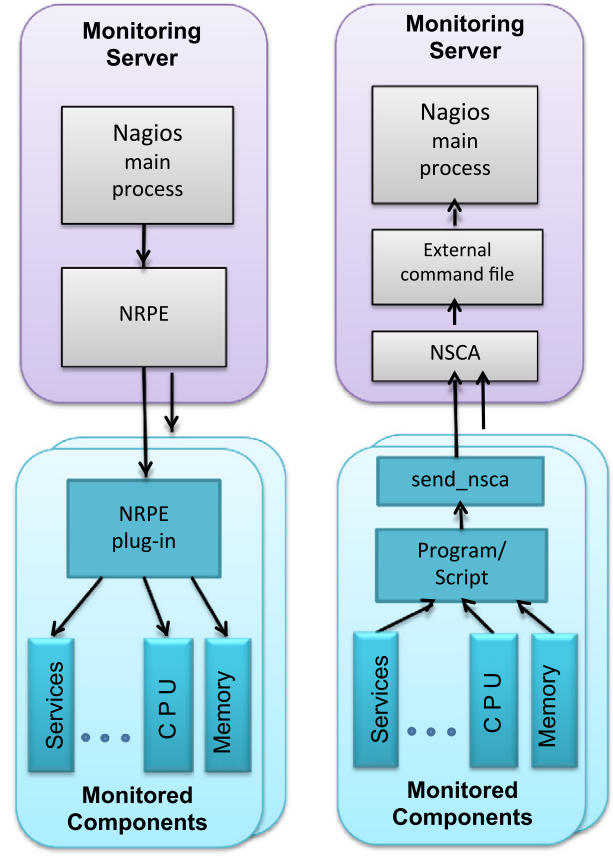
Table 1 summarises the general-purpose monitoring tools considered above and provides an overview of the technical details of the each of the tools, such as: what resources the tool aims to monitor; what programming language is used to implement the tool agent; operating systems the agent works on; whether the tool is under open source licence; the alerting mechanism used if any; and, whether the tool has any enterprise messaging support, *i.e* whether the tool uses a standardised messaging middleware. Reported limitations and documented usage experience in the scientific literature are also included which offers an opportunity to learn about the tools from the experience of others. Table 1 provides a brief comparison of the technology across monitoring tools. The extraction of technical details and usage experience of the tools also helps to identify the capabilities of the tools which are discussed in Section 5.

The tools described above are general-purpose, and are not designed specifically for monitoring Cloud deployments. However, they can be adapted for this purpose by changing/extending their design with Cloud-specific monitoring features.

3.2. Cloud specific monitoring tools

The advent of Cloud Computing has given rise to the development of Cloud-specific monitoring tools. Currently, Cloud providers offer diverse services using proprietary software and management techniques. In addition, many of these providers use provider-dependent monitoring tools which complement their offerings. For instance, Amazon Cloud Watch [3] monitors Amazon Web Services (AWS) such as Amazon EC2, Amazon RDS DB instances, and applications running on AWS. Azure Watch [5], on the other hand, monitors Azure-based resources, Windows Azure instances, SQL Azure Databases, websites, web applications, and Windows Azure storage. Both of these tools allow for user defined metrics monitored. In contrast, provider-independent Cloud monitoring tools exist that can be used to monitor a multiplicity of Cloud platforms. For example, Nimsoft [65] can monitor Amazon EC2, S3 Web Services, Rackspace Cloud, Microsoft Azure, Google App Engine, Google Apps and Salesforce CRM; Monitis [69] can monitor Amazon EC2/AWS and Rackspace Cloud; CloudKick [38] can monitor Amazon EC2, GoGrid and Rackspace Cloud. Finally, some monitoring tools, such as Private Cloud Monitoring Systems (PCMONS) [23] only monitor private Clouds. Table 2 summarises the review of Cloud specific monitoring tools.

Many Cloud based monitoring tools were initially designed to monitor IT infrastructure and was later extended to monitor Cloud deployments. For instance, the Nimsoft monitor originally provided monitoring for infrastructure, network, servers, databases, applications and virtualised environments but later it evolved to offer its services as a Unified Cloud Monitor for monitoring externally hosted systems and services. Like general purpose monitoring tools, provider independent Cloud monitoring tools have



(a) Active mode. (b) Passive mode.

Fig. 2. Nagios active and passive monitoring modes.

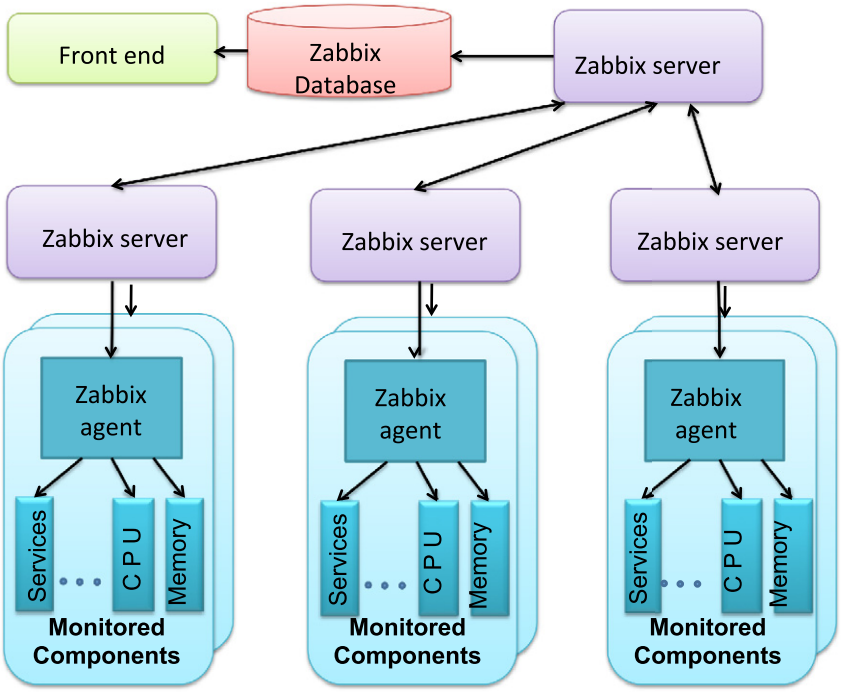


Fig. 3. Zabbix monitoring architecture.

agents installed in the monitored systems, which measure and send monitored data to the server for processing. The server issues alerts in cases needing attention. Most of the Cloud monitoring tools (*e.g.*, Nimsoft, CloudKick, Monitis, Boundary application monitor [12]) offer their services as SaaS that can be used to monitor third party Cloud installation. To realise this, the third

party Clouds must support the installation and execution of SaaS agents. To efficiently determine the response time and communication latency, some monitoring tools, such as Boundary only inspect packet header information in contrast to other tools, which periodically sample the entire monitored dataset.

Many of the Cloud monitoring tools, including Amazon Cloud Watch, Azure Watch, Nimsoft and Monitis, are capable of monitoring at both the infrastructure and application levels. However, some tools are capable of monitoring only at the infrastructure (CloudKick, PCMONS) or application levels (Boundary application monitor, mOSAIC [82] and Cloud Application SLA Violation Detection (CASViD) [26]). In some cases, Cloud providers publish the source code of their monitoring agents under open source licences. For instance, CloudKick offers their client code in C#, .Net, Python, Java and Ruby and Nimsoft's client code is available in C, Java, Perl, VB, and .Net.

Similar to Table 1, Table 2 specifies the technical details of the Cloud specific monitoring tools which includes resources monitored by the tool, whether the tool is open source, operating systems supported by the tool, alerting mechanisms supported by the tool, whether the tool uses any standardised messaging middleware, the programming language used to implement the tool and any reported limitations found in the literature.

4. Importance of monitoring: A taxonomy

Monitoring is a term currently used in several fields of study to represent various processes. In the context of computing, there are some definitions for this term relating it to specialised areas such as Grid, Service Oriented-Architecture (SOA) and Distributed Systems [2,93,60,51]. However, these definitions are not complete and do not reflect the full characteristics of a modern monitoring system in our opinion.

Therefore, we propose a precise and concise definition of the term *monitoring* so that we can contextualise what we believe to be the important capabilities associated with the monitoring process. These capabilities can then be used as the basis to assess the monitoring tools described previously. Thus, informed by Kornaros et al. [51] and Mansouri et al. [60], we define monitoring as:
"*A process that fully and precisely identifies the root cause of an event by capturing the correct information at the right time and at the lowest cost in order to determine the state of a system and to surface the status in a timely and meaningful manner*".

This definition views monitoring from a capability perspective. Terms like *lowest cost*, *system's status* and *timely*, indicate how the monitoring process should be exploited operationally in the service of managing complex systems like Clouds. The key characteristics of Cloud, such as, agility, low cost, device and location independence, multi-tenancy, high reliability, high scalability, security and sustainability [34], also identify some capabilities a monitoring tool should possess to adapt to Cloud environments. There are many Cloud operational areas, such as SLA and configuration management and security, within which monitoring plays an important role in servicing Cloud provider and Cloud consumer objectives. In this section, we identify the monitoring capabilities that are essential for facilitating complex management activities in Cloud environments. From this we construct and present a taxonomy of monitoring capabilities in the context of specific Cloud operational areas.

4.1. Desirable Cloud monitoring capabilities

This section presents some important capabilities of an efficient Cloud monitoring tool. Interpreting our definition of monitoring in the context of Cloud, and from the key Cloud characteristics, we identify the following list of prevalent capabilities:

- **Scalability:** Cloud deployment can be of very large scale, consisting of thousands of nodes. In order to manage these resources, a monitoring tool needs to be scalable to deliver the monitored information in a timely and flexible manner. The importance of this capability has been discussed in the literature [33,2,93,37]. Developers are currently striving to achieve high scalability in terms of resource and application management in Clouds. Therefore, a scalable means of supervising the service delivery processes for adequate management is fundamental.
- **Portability:** Cloud environments incorporate heterogeneous platforms and services. Therefore, the portability of monitoring tools, *i.e.*, the ability to move the tool from one platform to another, is indispensable to facilitate efficient management of such environments and to achieve wide penetration across multiple Clouds [93,33].
- **Non-intrusiveness:** In Clouds, there are large numbers of resources to be monitored, hence the computational power consumed by the monitoring tools to monitor all of these resources might have a big impact on the performance of the overall system. To cater for such environments, a monitoring tool should consume as little resource capacity as possible on the monitored systems so as not to hamper the overall performance of the monitored systems [33,2].
- **Robustness:** Clouds represent a frequent and dynamically changing environment. It is important that the monitoring tool detects changes in circumstances, such as the addition or removal of tenants and resources [37,2]. A monitoring tool needs the ability to adapt to a new situation by continuing its operation in the changed environment, which helps to mitigate faults and to provide accurate monitored information [33].
- **Multi-tenancy:** Clouds may offer a multi-tenant environment where multiple tenants share the same physical resources and application instances. A number of works in the literature have discussed the necessity of this functional requirement, especially in guaranteeing service level agreements and virtual machine monitoring [18,91,37]. To support multi-tenancy provisioning, the Cloud monitoring tool should maintain concurrency, *i.e.*, multiple customers being able to get common monitored information and isolation, *i.e.*, tenants only being able to access the information that is addressed to them. Efficient monitoring tools should embody this capability.
- **Interoperability:** Currently, Cloud environments may include dozens of independent, heterogeneous data centres operating mostly as stand-alone resources. Many business analysts have predicted the need for interoperable federated Clouds [15]. Interoperability is a prerequisite for Cloud bursting and for the creation of federated offerings from multiple providers [37,91]. A modern monitoring tool should be capable of sharing monitoring information between heterogeneous Cloud components for managing collaborative operations.
- **Customizability:** There are presently numerous Cloud service offerings and many providers are seeking ways to deliver unique services to their customers by allowing them high customisation flexibility. Considering the large number of customers, providers must be able to manage the service customisation of each customer, for example, by granting customers the ability to choose the metrics to be monitored for their service. Thus, to realise this goal, efficient monitoring tools should possess this capacity.
- **Extensibility:** With the rapid growth of Cloud computing, there are continuous changes and extensions to technologies especially in the area of management. Since monitoring techniques are fundamental to Cloud management, the monitoring tools need to be extensible and be able to adapt to new environments, such as being able to incorporate new monitoring metrics [93]. This is typically achieved through a modular design.

- **Shared resource monitoring:** Cloud technology uses virtualization of physical machine resources to achieve usage isolation in the form of virtual machines. The virtual machines share the underlying resources while multiple applications share the resources of virtual machine. Thus, to avoid resource contention among the virtual machines or manage resources shared by applications in a virtual machine, efficient monitoring is needed. A Cloud monitoring tool needs the capability of supervising shared resources to manage such an environment [37].
- **Usability:** Usability is one of the critical issues facing the adoption of Cloud computing. Fitness for purpose is an important factor when evaluating usability since the intended goal of a monitoring tool determines the usability judgement. As a consequence, any monitoring tool that is designed to support Cloud management needs to be easily useable [37]. To be highly useable a monitoring tool should facilitate deployment, maintenance and human interaction.
- **Affordability:** One of the reasons behind the popularity of Cloud adaptation is the reduction of cost. Cost effectiveness of a monitoring tool (*e.g.,* being open source) impacts on its wide spread acceptance [19]. A Gartner Survey [31] shows that more than 85% of enterprises were using open source software as of 2008 to drive down cost and increase flexibility. This indicates that being open source would also have a positive impact on the prominence of a monitoring tool. We rate affordability by considering both the cost of monitoring agent and the back end server component.
- **Archivability:** The availability of historical data can be useful for analysing and identifying the root cause of a problem in the long term [13,71]. In order to serve this purpose, a monitoring tool should possess a means of storing historical data.

4.2. Cloud operational areas facilitated by monitoring

Cloud stakeholders, such as providers and consumers, have varying motivations for gaining insight into Cloud operations. In this section, we present some Cloud operational areas that can be supported by monitoring. Later on, we present a taxonomy of the corresponding capabilities that a monitoring tool needs to possess in order to support these Cloud operational areas.

Cloud computing offers a new style of computing that allows consumers to pay only for the services used and frees them from the management overhead of the underlying infrastructure. This enables low initial set-up cost for business owners. In spite of the pricing advantage, consumers are still sceptical about Cloud offerings [59]. For assurance, they may require insight into areas such as (i) Cloud usage information to confirm the correctness of their bills, (ii) SLA enforcement mechanisms ensuring their QoS objectives and (iii) security and privacy policies guiding the storage and transfer of their data. Surveys show that the possible lack of security and loss of control over data in Clouds are among the major concerns of Cloud consumers [90,17]. Monitoring plays a role in detecting security breaches and hence can provide assurance of security maintenance.

Whilst Cloud consumers are free from the worry of maintenance overhead, providers on the other hand have the responsibility of maintaining and managing the underlying infrastructure. Monitoring is an essential part of Cloud management and serves various objectives of Cloud providers, such as (i) provisioning resources/services, (ii) optimal capacity planning, (iii) assuring SLAs, (iv) configuration management, (v) billing and (vi) security/privacy assurance.

The above discussion highlights the Cloud operational areas that are facilitated by monitoring. Table 3 summarises these areas and indicates the areas where providers and consumers have different monitoring perspectives.

In the following, we describe the Cloud operational areas that can benefit from monitoring. In the process, we reflect on the desirable capabilities of a monitoring tool that would make it fit for the purpose in question.

Accounting and billing: The notion of providing computing as a utility service relies heavily on the ability to record and account for the Cloud usage information on which billing schemes are based. Accurate accounting and billing relies on the ability to capture the consumption and allocation information of virtual resources as well as that of applications (*e.g.* compute hour used, bandwidth used) [24]. This is a capability that monitoring can provide. Furthermore, the provision of a transparent billing system that is able to record data in a verifiable and trustworthy manner, to ensure protection against forgery and false modifications, requires robust and secure monitoring capabilities [78,87].

SLA management: A Service Level Agreement (SLA) represents a contract signed between a service provider and a customer specifying the terms of a service offering including quality of service (QoS), pricing and penalties in case of violating the agreed terms [25,36]. SLA management is an area of great importance for Cloud providers since the assurance of SLA enforcement is inevitable for customer satisfaction and hence is a driving force for the continuity and growth of a Cloud business. The providers are expected to meet the QoS requirements as well as the Key Performance Indicators (KPI) for services in order to enforce their agreed SLA terms. Monitoring is essential to achieve these goals. The monitoring capabilities required to support operations in this area include the ability to measure QoS parameters, storing and analysing data, resource consumption measuring and SLA parameter assessment. These capabilities are expected from a monitoring tool for the purpose of SLA management [36,20,76,25].

Service/resource provisioning: Service/resource provisioning involves the allocation of resources optimally in order to match the workload [28]. It is an essential requirement for providing Cloud elasticity. Provisioning can be implemented in two ways: (1) static provisioning where VMs are created with a specified size and then consolidated onto a set of physical servers. The VM capacity does not change; and (2) dynamic provisioning: VM capacity is dynamically adjusted to match workload fluctuations [68]. The ability to measure the overall resource consumption of a system, along with the ability to measure per service resource consumption (which identifies the amount of resources each individual service needs), is essential for efficient provisioning. Furthermore the ability to assess risk and QoS is needed for effective provisioning decisions, such as whether to allocate/release resources to ensure that the quality is not compromised or resources are not wasted [28,94].

Capacity planning: Capacity planning is an important domain in Cloud computing, especially for the provider. It ensures adequate resource availability to meet the capacity demand necessary for securing a level of service quality and for serving various Cloud operational management activities, *e.g.,* disaster recovery and maintaining backups [81]. The ability to measure capacity usage enables operations such as predicting the need for more resources or determining resource wastage. Furthermore, the ability to detect Cloud node availability is necessary to maintain a required level of resource limits [67]. Monitoring capabilities such as component status identification play an important role in facilitating these goals.

Configuration management: Configuration is a set of parameters and values that determine the behaviour of devices and software [88]. While a Cloud provider may operate a multi-tenant environment it needs to manage customer-specific configuration. The initial configurations may contain the minimal set of resources required for a certain service. Resources may be added or released depending on varying load resulting into reconfiguration at run

Table 3
Cloud operational areas: provider and consumer perspectives.

Cloud operational area	Provider perspective	Consumer perspective
Accounting and billing	yes	yes
SLA management	yes	yes
Service and resource provisioning	yes	no
Capacity planning	yes	no
Configuration management	yes	no
Security and privacy assurance	yes	yes
Fault management	yes	no

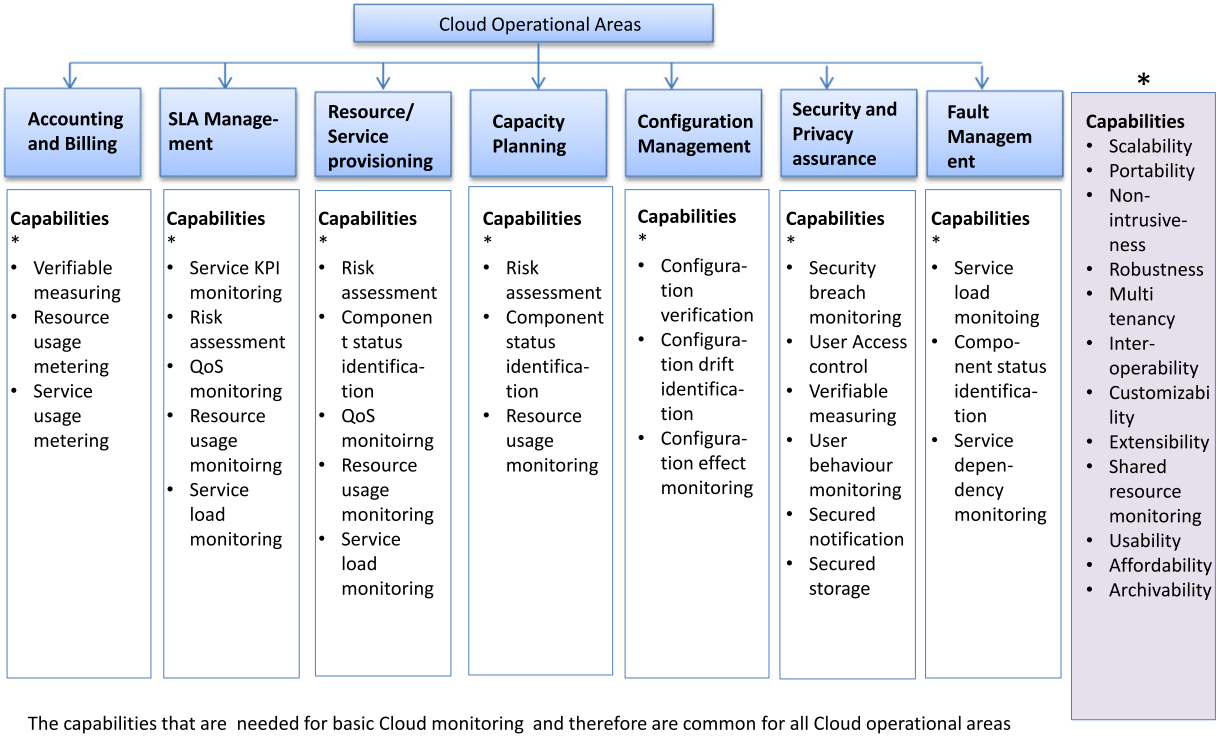


Fig. 4. Taxonomy of monitoring capabilities based on Cloud objectives.

time. Configuration management system needs to be able to verify specified configurations and to identify possible changes [29,88]. In this area, monitoring supports: configuration verification by identifying a configuration and verifying that the instances have the desired configuration; configuration drift identification by determining whether the configuration of an instance has changed; and Configuration effect monitoring by auditing the performance effects due to configuration changes.

Security and privacy assurance: The risk of compromising security and privacy is one of the major hindrances against the widespread use of Cloud computing [85]. The capability of detecting security breaches or attacks [53] is essential for this purpose and monitoring can help in this regard, for example, by identifying a malicious process consuming disappropriate system resources. To ensure Cloud services' security, it is important to assure that the tool being used for monitoring should not introduce any vulnerabilities. This is important especially in a multi-tenant Cloud environment. Monitoring capabilities such as user based access control, secure notification and storage are essential to support this operational area [92,90,91].

Fault management: Fault management is one of the core operational areas of Cloud that enables reliable and resilient Cloud services [44]. Continuous monitoring enables timely prediction and detection of failures, which can be pro-actively handled by replacing the suspected components [6]. Due to the complex constitution of Cloud components, faults can occur in many different ways, *e.g.*, server overload or network/hardware/service failure [45]. Therefore, the capability of identifying the load of a Cloud system and detecting availability of components is necessary to support the operations in this area.

In Fig. 4, we present the taxonomy of the identified monitoring capabilities categorised on the basis of Cloud operational areas. There are 12 common capabilities which are essential for all Cloud operational areas (as described in Section 4.1). It can be seen in the taxonomy that some of the capabilities are common in a number of operational areas but not in all operational areas; that results into 29 distinguished capabilities. Based on this taxonomy, we analyse the described monitoring tools to identify their strengths, challenges and to predict future trends in this area.

5. Analysis of tools based on taxonomy

This section presents our analysis of the monitoring tools described in Sections 3.1 and 3.2. This analysis is based on the monitoring capabilities taxonomy described previously. In line with our approach in Section 3, we partition the set of tools into two groups: those that are general purpose and those that are Cloud specific.

Table 4 presents the analysis of the general purpose monitoring tools and Table 5 presents the analysis for Cloud specific monitoring tools. The 2nd columns of the tables show the weighted average percentage of the implemented capabilities by the tools. In the

Table 4
General purpose monitoring tool analysis.

Capability/features	Percentage implemented	Nagios	Collectd	Opsview	Cacti	Zabbix	Open NMS	Ganglia	Hyperic	IBM Tivoli	Kiwi Monitor	DAMS	RDT
Scalability	46%	no	no	limited	no	yes	no	yes	yes	yes	no	yes	no
Portability	79%	limited	limited	yes	limited	yes	yes	yes	yes	yes	no	yes	yes
Non-intrusiveness[a]	50%	limited	limited	yes	no	yes	yes	limited	limited	yes	no	no	no
Robustness	33%	no	no	no	no	no	no	yes	yes	yes	no	no	yes
Multi-tenancy	33%	yes	no	yes	no	no	no	no	yes	yes	no	no	no
Interoperability	25%	no	yes	no	no	yes	yes	no	no	no	no	no	no
Customizability	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Extensibility	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Shared resource monitoring	42%	yes	yes	yes	no	yes	no	no	yes	no	no	no	no
Usability[a]	92%	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Affordability	79%	limited	yes	limited	yes	yes	yes	yes	limited	no	yes	yes	yes
Archivability	67%	yes	no	yes	yes	yes	yes	yes	yes	yes	no	no	no
Verifiable measuring	0%	no	no	no	no	no	no	no	no	no	no	no	no
Resource usage metering	75%	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	no
Service usage metering	50%	yes	yes	yes	no	yes	no	no	yes	no	yes	no	no
Service KPI monitoring	0%	no	no	no	no	no	no	no	no	no	no	no	no
QoS monitoring	50%	yes	no	yes	no	yes	no	no	yes	yes	yes	no	no
Risk assessment	58%	yes	no	yes	yes	yes	yes	no	yes	yes	no	no	no
Component status identification	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Service load monitoring	50%	yes	yes	yes	no	yes	no	no	yes	no	yes	no	no
Configuration verification	75%	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	no
Configuration drift identification	75%	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	no
Configuration effect monitoring	75%	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no	no
Security breach monitoring	33%	yes	no	no	no	yes	no	no	yes	yes	no	no	no
User access control	50%	no	no	yes	yes	yes	yes	no	yes	yes	no	no	no
User activity	17%	no	no	no	no	no	no	no	no	yes	yes	no	no
Secured notification	17%	no	no	no	no	no	yes	no	no	yes	no	no	no
Secured storage	17%	no	no	no	no	no	no	no	yes	yes	no	no	no
Service dependency	21%	no	no	no	no	no	no	no	yes	no	no	yes	yes
Percentage covered by tools		61%	52%	70%	46%	78%	59%	50%	**85%**	78%	33%	30%	30%

[a] We note that some of these capabilities are somewhat subjective. With that in mind the evaluation presented here is based on the weight of opinion as reflected in the reviewed literature.

calculations, we assign "1" if a tool has a particular capability and "0" if it does not. There is the assignment of "0.5" if the tool partly implements such a capability. The sum of these values is used to calculate the assessment percentage. According to Tables 4 and 5, a "1" is equivalent to "yes", a "0" implies "no" and "0.5" represents "limited". The capabilities that have scored "0" for all the tools are excluded from the calculation of weighted average percentage of capabilities covered by each tool which are presented in the last rows of the tables. For tools with multiple versions—for example, Nagios, Opsview, Hyperic, CloudKick, Nimsoft and Monitis, the capabilities are identified based on the superset of features of all versions.

The determinations of the capability implementation by the tools are based on literature reviews and evaluations of the tools. Some of the capabilities such as non-intrusiveness and usability are subjective, hence they are evaluated based on the weight of opinion found in the reviewed literature.

5.1. Analysis of general purpose monitoring tools

The Cloud monitoring capabilities taxonomy is applied to analyse the general purpose monitoring tools to determine how these tools could be extended/adapted to facilitate efficient Cloud monitoring.

As shown in Table 4, this group of tools has broad implementation of the portability, customizability, extensibility, usability, affordability, resource usage metering, component status identification ability, configuration verification ability, configuration drift identification ability and configuration effect monitoring ability.

These tools are weaker on scalability, non-intrusiveness, robustness, multi-tenancy, per service resource consumption ability, QoS monitoring, risk assessment and, service load monitoring capabilities. Interoperability and other security and privacy assurance related capabilities are least implemented by these tools as shown by their calculated percentages. Some desirable capabilities such as verifiable metering, service KPI monitoring are not implemented by any of the tools in this group. The lack of support for these capabilities illustrates the fact that the tools were not designed with Cloud in mind and must be extended or redesigned to be useable in a Cloud environment. The identification of these capabilities emphasises the challenges in adapting these tools for monitoring Clouds, since these capabilities are essential for efficient Cloud operational management.

The last row of Table 4 shows the weighted average percentage of the capabilities covered by each monitoring tool. This row provides a relative comparison of the number of capabilities implemented by each tool. As shown in the table, the Hyperic monitoring tools possess the highest percentage, implementing 85% of all capabilities. All of the general purpose monitoring tools surveyed scored in excess of 50% except for Kiwi Application Monitor, DAMS and RDT each of which only implemented around 30% of the listed capabilities. These three tools were mainly designed to monitoring general applications and the scoring shows the fact that they are not fit for full Cloud operational management, which requires resource and application monitoring.

5.2. Analysis of Cloud specific monitoring tools

In this section, we analyse the Cloud specific monitoring tools using the previously described taxonomy. The goal of the analysis is to determine the strengths, drawbacks and challenges facing these tools. Table 5 presents the analysis. As shown in the table, all of the tools implement the customizability, extensibility, resource usage metering, service usage metering, component

Table 5
Cloud based monitoring tool analysis.

Capability/features	Percentage implemented	CloudKick	Nimsoft	Monitis	Amazon Cloud Watch	Azure Watch	PCMONS	Boundary app. monitor	mOSAIC	CASViD
Scalability	78%	yes	yes	yes	yes	yes	no	yes	no	yes
Portability	56%	limited	yes	yes	no	no	no	yes	yes	yes
Non-intrusiveness[a]	94%	yes	yes	yes	yes	yes	limited	yes	yes	yes
Robustness	67%	yes	yes	yes	yes	yes	no	yes	no	no
Multi-tenancy	44%	yes	yes	no	yes	yes	no	no	no	no
Interoperability	33%	no	no	no	no	no	yes	no	yes	yes
Customizability	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes
Extensibility	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes
Shared Resource monitoring	78%	yes	yes	yes	yes	yes	no	yes	no	yes
Usability[a]	94%	yes	yes	yes	yes	yes	limited	yes	yes	yes
Affordability	50%	limited	limited	limited	no	no	yes	no	yes	yes
Archivability	78%	yes	yes	yes	yes	yes	yes	yes	no	no
Verifiable measuring	0%	no	no	no	no	no	no	no	no	no
Resource usage metering	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes
Service usage metering	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes
Service KPI monitoring	0%	no	no	no	no	no	no	no	no	no
QoS	89%	yes	yes	yes	yes	yes	no	yes	yes	yes
Risk assessment	92%	yes	yes	limited	yes	yes	yes	yes	yes	yes
Component status identification	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes
Service load monitoring	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes
Configuration verification	78%	yes	yes	yes	yes	yes	yes	no	no	yes
Configuration drift identification	78%	yes	yes	yes	yes	yes	yes	no	no	yes
Configuration effect monitoring	100%	yes	yes	yes	yes	yes	yes	yes	yes	yes
Security breaches monitoring	56%	yes	yes	no	yes	yes	no	yes	no	no
User access control	67%	yes	yes	yes	yes	yes	no	yes	no	no
User activity monitoring	0%	no	no	no	no	no	no	no	no	no
Secured notification	33%	no	yes	no	yes	yes	no	no	no	no
Secured storage	33%	no	yes	no	yes	yes	no	no	no	no
Service dependency	11%	no	no	no	no	no	no	yes	no	no
Percentage covered by tools		78%	**87%**	70%	85%	85%	52%	73%	54%	69%

[a] We note that some of these capabilities are somewhat subjective. With that in mind the evaluation presented here is based on the weight of opinion as reflected in the reviewed literature.

status identification, service load monitoring and configuration effect monitoring capabilities.

This group of tools lacks on the implementation of portability, multi-tenancy, interoperability, secured notification, secured storage and service dependency capabilities. These tools are generally designed for monitoring in Clouds and many of them are commercial and proprietary, *i.e.*, provider and platform dependent. This accounts for the low levels of interoperability (33%) and portability (56%) observed. The implementation of multi-tenancy of these tools is 44% compared to 33% for general purpose monitoring tools. Since this capability is indispensable for multi-tenant Cloud environment, thus, we argue that this area is a challenge for future research. The need for securing the monitoring tool itself is important for ensuring that the tool does not create any security holes in the Cloud. As can be observed, capabilities associated with secure monitoring are lacking implementations. Therefore, future research efforts are required in this area.

Compared to the general purpose monitoring tools, the scalability (78%), non-intrusiveness (94%), robustness (67%), multi-tenancy (44%), shared resource monitoring (78%), resource usage metering (100%), per service resource consumption metering (100%) and service load monitoring (100%) capabilities are better addressed in the Cloud specific monitoring tools. This is reasonable as the demand for these capabilities is high in Clouds.

In line with our descriptions, the last row of Table 5 shows the weighted average percentage of the capabilities implemented by each of the tools. The Nimsoft monitor emerges as the best with 87% coverage and the PCMONS tool is the least with 52%.

One interesting observation is that most of the capabilities are improved with Cloud based monitoring tool except for portability and affordability. This reinforces the fact that many of the Cloud specific monitoring tools are commercial and vendor dependent, which make the tools less portable and less affordable as most of the general purpose monitoring tools are open-source.

It is interesting to note that none of the tools analysed implements verifiable measuring, service KPI monitoring and user activity monitoring capabilities. This shows that the realisation of these features is still a challenge, which indicates that further research efforts are needed in this direction.

5.3. Capability implementation coverage

In this section, we present a graphical representation of the capability implementation percentages of the monitoring tools. We take the Cloud operational areas associated with the capabilities into consideration. This representation clearly shows the Cloud operational areas that are currently better supported by the monitoring tools in terms of implemented capabilities and the ones that are poorly supported, which exposes the need for further research efforts in those areas.

Fig. 5 shows this graphical representation. It includes the capabilities of the general purpose as well as the Cloud based monitoring tools. From the figure, it can be observed that all the operational areas are better supported by the Cloud based monitoring tools compared to the general purpose monitoring tools. This can be attributed to the fact that Cloud based monitoring tools are better specialised in terms of supporting Cloud operational management.

With the exception of capabilities relating to security and privacy assurance operational area, other areas are supported by at least 57% implementation coverage. This implies that the current monitoring tools are least supportive of security and privacy assurance management in Clouds.

Table 6
Cloud operational area based analysis of general purpose monitoring tools.

Cloud operational areas	Nagios	Collectd	Opsview	Cacti	Zabbix	Open NMS	Ganglia	Hyperic	IBM Tivoli	Kiwi Monitor	DAMS	RDT
Accounting and billing	61%	64%	79%	46%	**86%**	64%	68%	**86%**	71%	36%	43%	43%
SLA management	66%	56%	81%	47%	**88%**	63%	59%	**88%**	75%	38%	38%	38%
Service and resource provisioning	68%	59%	82%	50%	**88%**	65%	62%	**88%**	76%	41%	41%	41%
Capacity planning	60%	60%	80%	56%	**87%**	73%	70%	**87%**	80%	33%	46%	46%
Configuration management	63%	66%	80%	56%	**86%**	73%	73%	**86%**	80%	26%	40%	40%
Security and privacy assurance	44%	41%	59%	38%	70%	59%	50%	76%	**82%**	29%	41%	41%
Fault management	57%	60%	73%	43%	80%	60%	63%	**87%**	67%	40%	53%	53%

Table 7
Cloud operational area based analysis of Cloud based monitoring tools.

Cloud operational areas	CloudKick	NimSoft	Monitis	Amazon Cloud Watch	Azure Watch	PCMONS	Boundary app. monitor	mOSAIC	CASViD
Accountability and Billing	86%	**89%**	82%	79%	79%	57%	79%	64%	79%
SLA management	88%	**90%**	81%	81%	81%	56%	81%	69%	81%
Service and resource provisioning	88%	**91%**	82%	82%	82%	59%	82%	70%	82%
Capacity planning	87%	**90%**	80%	80%	80%	60%	80%	66%	80%
Configuration management	87%	**90%**	83%	80%	80%	60%	73%	60%	80%
Security and privacy assurance	75%	**90%**	66%	81%	81%	38%	69%	44%	56%
Fault management	80%	**83%**	77%	73%	73%	53%	80%	60%	73%

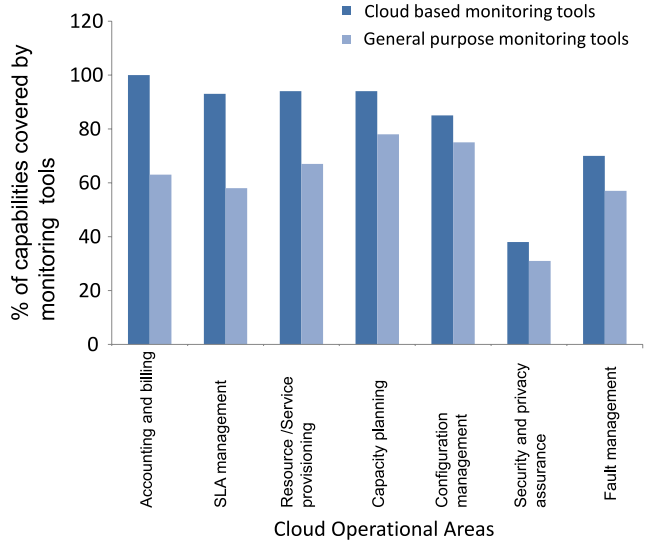


Fig. 5. Capacity implementation coverage by monitoring tools.

5.4. Cloud operational area coverage

In this section, we present the coverage for each Cloud operational area by the monitoring tools in terms of capability implementation percentages relevant for an operational area. This analysis reveals the fitness of each monitoring tool for various Cloud operational areas. Table 6 shows the analysis of general purpose monitoring tools in terms of coverage for each Cloud operational area whereas Table 7 shows the same for Cloud specific monitoring tools. The percentages shown in the tables are calculated from Tables 4 and 5 by separating the relevant capabilities of each Cloud operational areas as presented in the taxonomy.

From the comparison of Tables 4 and 6, it is interesting to observe that although Hyperic performs the best among the general purpose monitoring tools in terms of implementation percentage of the considered capabilities, it is outperformed by IBM Tivoli in terms of security and privacy assurance. Zabbix is also found to be as good as Hyperic for supporting almost all of the operational areas except for security and privacy assurance and fault management. However, among the Cloud specific monitoring tools Nimsoft is found to perform the best not only in terms of implementation percentage of the considered capability but also for the support of each individual Cloud operational areas. In the next section, we differentiate our contributions to the previous work.

6. Related work

This section analyses the previous work on this topic and highlights our differences. Buytaert et al. [13] provide a detailed review of some monitoring tools: Nagios, OpenNMS, Zabbix, and Hyperic while identifying the expected features of a monitoring system from the users', system administrators', and developers' perspectives. However, their analysis is limited to the suitability of the tools for monitoring computing infrastructures and networks only. Krizanic et al. [52] review performance monitoring tools and categorise them according to the operating systems and notification/alerting facilities that they support. Nevertheless, their analysis does not consider the application of the tools to Cloud monitoring. Zanikolas et al. [93] categorise monitoring tools applicable to Grid Computing based on their usefulness for generating, processing, distributing and consuming monitoring data. The authors identify the overlapping functionality, lack of agreement among the protocols and semantics used in monitoring projects while pointing out the need for more coordination and interoperability among those projects. But, their approach does not consider Cloud environments. Benedict [9] surveys performance monitoring tools with the goal of helping developers choose the best monitoring tool for High Performance Computing (HPC) applications. His review does not consider the applicability of these tools to high-level Cloud monitoring scenarios.

Rimal et al. [84] survey a number of Cloud service providers and place their offerings into categories such as architecture, virtualization technology, load balancing, fault tolerance, interoperability, and storage systems. Nevertheless, the authors do not consider the monitoring techniques used by these Cloud providers. Aceto et al. [2,1] present a taxonomy of Cloud monitoring which includes (i) the need for monitoring, revealing the various Cloud activities where monitoring is an essential task (ii) basic concepts, such as layers, abstraction levels and metrics, (iii) properties, which exhibit various characteristics a distributed monitoring system

should have and (iv) open issues and future directions. Furthermore, their survey categorises the tools into open-source, commercial and services. Our work addresses the same area, *i.e.* Cloud monitoring but, from a different perspective. We surveyed the monitoring tools and categorised them into general purpose and Cloud specific monitoring tools. Such a categorisation realises a historical view of these tools evolution from the pre-Cloud era to the Cloud era. We identified the prevalent Cloud monitoring capabilities, which ensue from our definition of monitoring and key Cloud characteristics. We provided distinguished provider and consumer views of various Cloud operational areas where monitoring plays an inevitable role. The identified operational areas are deeply explored to surface the required capabilities from a monitoring tool in order to serve the objectives of those areas. Additionally, we presented a taxonomy of these capabilities. Moreover, our work provides a clear association of the monitoring tool capabilities to the Cloud operational areas, which is not the case in [2,1].

To the best of our knowledge none of the previous work gave a historical evolution of monitoring tools and clearly associates Cloud operational areas with the required monitoring capabilities to effect informed decision and thereby achieve efficient management. In the next section, we discuss the identified research challenges.

7. Identified challenges

In this section, we discuss the challenges and future trends derived from our analysis.

7.1. Trends and challenges

Cloud Computing is a promising technology that is currently gaining wide acceptance in industry. Efficient management of this technology still remains a challenge and will be an important research area for the coming years. The work presented in this paper has analysed the state of the art of monitoring tools using a capability based approach. This novel approach afforded us the opportunity to compare and contrast the strengths and weaknesses of these tools in terms of their implemented functionalities.

General purpose monitoring tools, as we observed, are mainly focused on infrastructure resource monitoring. Many of these tools are currently being adapted for monitoring in Clouds, and we foresee that this adaptation process will continue into the future. In our analysis of general purpose monitoring tools, we observed that the realisation of some desirable capabilities such as scalability, robustness and interoperability, is still a challenge. Our analysis showed that none of the tools surveyed have capabilities for verifiable metering and service KPI monitoring. Verifiable metering ensures the integrity of monitored data and is important for trustworthiness of the Cloud provider. Monitoring service KPI is important for SaaS provider to measure the performance of the software service. Furthermore, the tools lack capabilities implementing user activity monitoring, secured notification, secured storage and service dependency monitoring. In a multi-tenant Cloud environment, it is important to monitor user activity to detect, for example, a malicious user. We consider these capabilities to be important for advanced Clouds operational management. We predict, therefore, that future research in this area will be focused on addressing these issues.

As evident from our review, Cloud specific monitoring tools are mostly being developed by commercial Cloud providers to help in managing their services and enforcing agreed Service Level Objectives (SLOs). This makes these tools platform dependent and proprietary. As can be seen from our analysis in Table 5, these tools are currently lacking in the implementations of portability and affordability capabilities. Some Cloud specific monitoring tools have also been developed in academia. However, these are largely prototypes and may not be production ready. We predict that most of the research effort on Cloud monitoring in the future will focus on security capabilities of monitoring tools to make Cloud more dependable and trustworthy.

Lack of support for these capabilities may say more about the speed at which the field is evolving rather than the importance with which these capabilities are viewed. We argue that this is an important area for research for the future.

8. Conclusion and future work

Monitoring in Clouds is an area that is yet to be fully realised. It plays an important role in supporting efficient management of various Cloud operational areas, including SLA management, service and resource provisioning. In addition we have postulated that it can be used perhaps in the future to provide quantitative support for trust assurance.

In this paper, we have reviewed the technical details of the monitoring tools classifying them into general purpose and Cloud specific categories, which offers us an opportunity to gain insights of the tools and historically analyse their evolution. From our monitoring definition, we derived a list of capabilities that are considered relevant to facilitate efficient Cloud operational management. Since Cloud environments consist of various areas with unique functions that can be managed separately, we have identified some of these areas in order to investigate the role of monitoring in supporting them from both providers' and consumers' perspectives. To systematically achieve this goal, we defined a taxonomy grouping the capabilities of the different Cloud operational areas. This taxonomy provides a useful benchmark for analysing the strengths and weakness of the monitoring tools. Tables 4 and 5 highlight the extent to which the tools considered address the list of capabilities. More importantly it draws attention to those capabilities that are under-represented.

We noted that general purpose monitoring tools and Cloud specific monitoring tools as a whole exposed different capabilities in accordance with their deployment domain. In general, this unbalanced emphasis on certain capabilities was broadly in line with expectations. General purpose monitoring tools are commonly designed with a client–server architecture where the client resides on the monitored object and communicates information to the server. These tools were designed for monitoring fixed-resource environments where there was no dynamic scaling of resources. This is reflected in the lack of many capabilities like scalability as shown by our analysis in Table 4. We argue that in designing future monitoring tools especially for Clouds, these challenges must be addressed since issues such as scalability are important for Cloud monitoring.

The capability based analysis of the monitoring tools has helped to identify the Cloud operational areas that are better addressed by these tools and more importantly the areas that are lacking support so that future research can take them into consideration. The review and analysis in this paper have provided a comprehensive overview of the described monitoring tools and their ability to support Cloud operational management. We consider this piece of work as a reference and a basis for further research work in this area as described in the next section.

8.1. Future work

Proper and extensive Cloud monitoring has applications far beyond Cloud operational management. Applied intelligently, Cloud monitoring can form the foundation for trust assurance and analytic techniques for visualising and analysing monitored data. From the trust assurance perspective, surveys of consumers and enterprises have consistently highlighted concerns about

entrustment of data to Cloud service providers. Cloud trustmarks have been proposed as a means of addressing these concerns [59]. By utilising modern web technologies, such as HTML 5, trustmarks could be presented as active dynamic entities that succinctly communicate up-to-date values for a number of high-level dependability measures. These dependability measures would be based on "live" analytics of aspects of the underlying service supported by appropriate active monitoring.

From the perspective of data analysis and visualisation, the management of diverse Cloud metrics is an on-going challenge. To address this challenge the construction of a metric ontology which would categorise and structure in a useful manner the thousands of available Cloud metrics would be extremely beneficial. Such an ontology would afford us the chance to identify and study the important factors for making decisions, for instance, moving business applications to the Cloud or selecting particular providers and service offerings. This work would also include the design of analytic tools for analysing data and generating trend information through a dashboard, for example.

Acknowledgments

The research work described in this paper was supported by the Irish Centre for Cloud Computing and Commerce, an Irish national Technology Centre funded by Enterprise Ireland and the Irish Industrial Development Authority.

References

[1] G. Aceto, A. Botta, W. de Donato, A. Pescapè, Cloud monitoring: definitions, issues and future directions, in: Proceedings of IEEE 1st International Conference on Cloud Networking (CLOUDNET), IEEE, 2012, pp. 63–67.
[2] G. Aceto, A. Botta, W. de Donato, A. Pescapè, Cloud monitoring: a survey, Comput. Netw. 57 (9) (2013) 2093–2115.
[3] Amazon, Amazon CloudWatch: Developer guide, 2009.
[4] 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 Gener. Comput. Syst. 21 (4) (2005) 559–571.
[5] AzureWatch (Jan. 2014). URL https://www.paraleap.com/AzureWatch.
[6] A. Bala, I. Chana, Fault tolerance-challenges, techniques and implementation in cloud computing, Computer Science and Engineering Department, Thapar University Patiala, Punjab, India.
[7] W. Barth, Nagios: system and network monitoring, No Starch Pr (2008).
[8] P. Bellavista, G. Carella, L. Foschini, T. Magedanz, F. Schreiner, K. Campowsky, QoS-aware elastic cloud brokering for IMS infrastructures, in: Computers and Communications (ISCC), 2012 IEEE Symposium on, IEEE, 2012, pp. 157–160.
[9] S. Benedict, Performance issues and performance analysis tools for HPC cloud applications: a survey, Computing (2012) 1–20.
[10] B. Bonev, S. Ilieva, Monitoring Java based SOA applications, in: Proceedings of the 13th International Conference on Computer Systems and Technologies, ACM, 2012, pp. 155–162.
[11] G. Boss, P. Malladi, D. Quan, L. Legregni, H. Hall, Cloud computing, IBM white paper 1369.
[12] Boundary application monitor (Jan. 2014). URL http://boundary.com/blog/tag/application-monitoring/.
[13] K. Buytaert, T. De Cooman, F. Descamps, B. Verwilst, Systems monitoring shootout, in: Linux Symposium, 2008, p. 53.
[14] V. Cardellini, S. Iannucci, Designing a flexible and modular architecture for a private cloud: a case study, in: Proceedings of the 6th International Workshop on Virtualization Technologies in Distributed Computing Date, in: VTDC'12, ACM, New York, NY, USA, 2012, pp. 37–44. http://dx.doi.org/10.1145/2287056.2287067.
[15] A. Celesti, F. Tusa, M. Villari, A. Puliafito, How to enhance cloud architectures to enable cross-federation, in: Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing, in: CLOUD'10, 2010, pp. 337–345.
[16] J. Chen, R. Childress, I. Mcintosh, G. Africa, A. Sitaramayya, A service management architecture component model, in: Network and Service Management (CNSM), 2011 7th International Conference on, IEEE, 2011, pp. 1–4.
[17] D. Chen, H. Zhao, Data security and privacy protection issues in cloud computing, in: Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on, vol. 1, IEEE, 2012, pp. 647–651.
[18] X. Cheng, Y. Shi, Q. Li, A multi-tenant oriented performance monitoring, detecting and scheduling architecture based on sla, in: 2009 Joint Conferences on Pervasive Computing (JCPC), 2009, pp. 599–604. http://dx.doi.org/10.1109/JCPC.2009.5420114.
[19] C.C. Cirstoiu, C.C. Grigoras, L.L. Betev, A.A. Costan, I.C. Legrand, Monitoring, accounting and automated decision support for the alice experiment based on the MonALISA framework, in: Proceedings of the 2007 Workshop on Grid Monitoring, ACM, 2007, pp. 39–44.
[20] M. Comuzzi, C. Kotsokalis, G. Spanoudakis, R. Yahyapour, Establishing and monitoring SLAs in complex service based systems, in: Web Services, 2009. ICWS 2009. IEEE International Conference on, IEEE, 2009, pp. 783–790.
[21] B. Cowie, Building A Better Network Monitoring System, 2012.
[22] M. de Carvalho, L. Granville, Incorporating virtualization awareness in service monitoring systems, in: Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on, 2011, pp. 297–304. http://dx.doi.org/10.1109/INM.2011.5990704.
[23] S. De Chaves, R. Uriarte, C. Westphall, Toward an architecture for monitoring private clouds, IEEE Commun. Mag. 49 (12) (2011) 130–137.
[24] E. Elmroth, F.G. Marquez, D. Henriksson, D.P. Ferrera, Accounting and billing for federated cloud infrastructures, in: Grid and Cooperative Computing, 2009. GCC'09. Eighth International Conference on, IEEE, 2009, pp. 268–275.
[25] V.C. Emeakaroha, I. Brandic, M. Maurer, S. Dustdar, Low level metrics to high level SLAs-LoM2HiS framework: bridging the gap between monitored metrics and SLA parameters in cloud environments, in: High Performance Computing and Simulation (HPCS), 2010 International Conference on, IEEE, 2010, pp. 48–54.
[26] V. Emeakaroha, T. Ferreto, M. Netto, I. Brandic, C. De Rose, CASViD: application level monitoring for SLA violation detection in clouds, in: Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual, 2012, pp. 499–508. http://dx.doi.org/10.1109/COMPSAC.2012.68.
[27] Z. Feng, L. Huang, R-binder: application of using service binder in R-OSGi, in: Computer Science and Computational Technology, 2008, in: ISCSCT'08. International Symposium on, Vol. 2, IEEE, 2008, pp. 34–39.
[28] A.J. Ferrer, F. Hernández, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R.M. Badia, K. Djemame, et al., Optimis: a holistic approach to cloud service provisioning, Future Gener. Comput. Syst. 28 (1) (2012) 66–77.
[29] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini, E. Turrini, QoS aware clouds, in: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, IEEE, 2010, pp. 321–328.
[30] M. Frank, L. Oniciuc, U. Meyer-Baese, I. Chiorescu, A space-efficient quantum computer simulator suitable for high-speed FPGA implementation, Proc. SPIE 7342, Quantum Information and Computation VII, 734203 (April 27, 2009); http://dx.doi.org/10.1117/12.817924.
[31] Gartner, Open source survey 2008. URL http://www.gartner.com/technology/home.jsp.
[32] GigaSpaces, Easy deployment of mission-critical applications to the cloud, 2011.
[33] S.V. Gogouvitis, V. Alexandrou, N. Mavrogeorgi, S. Koutsoutos, D. Kyriazis, T. Varvarigou, A monitoring mechanism for storage clouds, in: Cloud and Green Computing (CGC), 2012 Second International Conference on, IEEE, 2012, pp. 153–159.
[34] C. Gong, J. Liu, Q. Zhang, H. Chen, Z. Gong, The characteristics of cloud computing, in: Parallel Processing Workshops (ICPPW), 2010 39th International Conference on, IEEE, 2010, pp. 275–279.
[35] D. Gupta, P. Mohapatra, C.-N. Chuah, Efficient monitoring in wireless mesh networks: overheads and accuracy trade-offs, in: Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on, IEEE, 2008, pp. 13–23.
[36] Z. Haiteng, S. Zhiqing, Z. Hong, Z. Jie, Establishing service level agreement requirement based on monitoring, in: Cloud and Green Computing (CGC), 2012 Second International Conference on, IEEE, 2012, pp. 472–476.
[37] 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, IEEE, 2010, pp. 350–356.
[38] B. Hoßbach, Reaktives cloud monitoring mit complex event processing (Master's thesis), Philipps Universität Marburg, Department of Mathematics and Informatics, Germany, 2011.
[39] Hyperic (Jan. 2014). URL http://www.hyperic.com.
[40] IBM Tivoli monitoring (Jan. 2014). URL https://publib.boulder.ibm.com/infocenter/ieduasst/tivv1r0/index.jsp?topic=/com.ibm.iea.itm/plugin_coverpage.html.
[41] E. Imamagic, D. Dobrenic, Grid infrastructure monitoring system based on Nagios, in: Proceedings of the 2007 workshop on Grid monitoring, in: GMW'07, ACM, New York, NY, USA, 2007, pp. 23–28. http://dx.doi.org/10.1145/1272680.1272685.
[42] E. Iori, A. Simitsis, T. Palpanas, K. Wilkinson, S. Harizopoulos, CloudAlloc: a monitoring and reservation system for compute clusters, in: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, in: SIGMOD'12, ACM, New York, NY, USA, 2012, pp. 721–724. http://dx.doi.org/10.1145/2213836.2213942.
[43] C. Issariyapat, P. Pongpaibool, S. Mongkolluksame, K. Meesublak, Using Nagios as a groundwork for developing a better network monitoring system, in: Technology Management for Emerging Technologies (PICMET), in: 2012 Proceedings of PICMET'12, IEEE, 2012, pp. 2771–2777.
[44] R. Jhawar, V. Piuri, M. Santambrogio, A comprehensive conceptual system-level approach to fault tolerance in cloud computing, in: Systems Conference (SysCon), 2012 IEEE International, IEEE, 2012, pp. 1–5.
[45] R. Jhawar, V. Piuri, M. Santambrogio, Fault tolerance management in cloud computing: A system-level perspective, Syst. J. IEEE 7 (2) (2013) 288–297.

[46] H. Jiang, H. Lv, N. Wang, R. Di, A performance monitoring solution for distributed application system based on JMX, in: Grid and Cooperative Computing (GCC), 2010 9th International Conference on, 2010, pp. 124 –127. http://dx.doi.org/10.1109/GCC.2010.35.
[47] W.M. Jones, J.T. Daly, N. DeBardeleben, Application monitoring and checkpointing in HPC: looking towards exascale systems, in: Proceedings of the 50th Annual Southeast Regional Conference, ACM, 2012, pp. 262–267.
[48] G. Katsaros, R. Kübert, G. Gallizo, Building a service-oriented monitoring framework with REST and Nagios, in: Services Computing (SCC), 2011 IEEE International Conference on, IEEE, 2011, pp. 426–431. http://dx.doi.org/10.1109/SCC.2011.53.
[49] E. Kijsipongse, S. Vannarat, Autonomic resource provisioning in Rocks clusters using Eucalyptus cloud computing, in: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, in: MEDES'10, ACM, New York, NY, USA, 2010, pp. 61–66. http://dx.doi.org/10.1145/1936254.1936265.
[50] I. Konstantinou, E. Angelou, D. Tsoumakos, C. Boumpouka, N. Koziris, S. Sioutas, Tiramola: elastic nosql provisioning through a cloud management platform, in: SIGMOD Conference'12, ACM, 2012, pp. 725–728.
[51] G. Kornaros, D. Pnevmatikatos, A survey and taxonomy of on-chip monitoring of multicore systems-on-chip, ACM Trans. Des. Autom. Electron. Syst. 18 (2) (2013) 17.
[52] J. Krizanic, A. Grguric, M. Mosmondor, P. Lazarevski, Load testing and performance monitoring tools in use with AJAX based web applications, in: MIPRO, 2010 Proceedings of the 33rd International Convention, IEEE, 2010, pp. 428–434.
[53] R.L. Krutz, R.D. Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Wiley, 2010.
[54] B.S. Lee, S. Yan, D. Ma, G. Zhao, Aggregating IaaS service, in: SRII Global Conference (SRII), 2011 Annual, IEEE, 2011, pp. 335–338.
[55] D. liang Lee, S.-Y. Yang, Y.-J. Chang, Developing an active mode of network management system with intelligent multi-agent techniques, in: Pervasive Computing (JCPC), 2009 Joint Conferences on, 2009, pp. 77–82.
[56] A. Lenk, M. Klems, J. Nimis, S. Tai, T. Sandholm, What's inside the cloud? An architectural map of the cloud landscape, in: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, IEEE Computer Society, 2009, pp. 23–31.
[57] T. Lindsey, Apparatus and method for modeling, and storing within a database, services on a telecommunications network, US Patent App. 10/127, 315 (April 2002).
[58] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, D. Leaf, NIST Cloud Computing Reference Architecture, vol. 500, NIST Special Publication, 2011, p. 292.
[59] T. Lynn, P. Healy, R. McClatchey, J. Morrison, C. Pahl, B. Lee, The case for cloud service trustmarks and assurance-as-a-service, in: Proceedings of the 3rd International Conference on Cloud Computing and Services Science (CLOSER), 2013.
[60] M. Mansouri-Samani, M. Sloman, Monitoring distributed systems, IEEE Netw. 7 (6) (1993) 20–30.
[61] F. Marchioni, JBoss AS 7 Configuration, Deployment and Administration, Packt Publishing Ltd., 2012.
[62] P. Marshall, H. Tufo, K. Keahey, D. LaBissoniere, M. Woitaszek, Architecting a large-scale elastic environment: recontextualization and adaptive cloud services for scientific computing, in: 7th International Conference on Software Paradigm Trends (ICSOFT), Rome, Italy, 2012.
[63] R. Marty, Cloud application logging for forensics, in: Proceedings of the 2011 ACM Symposium on Applied Computing, ACM, 2011, pp. 178–184.
[64] M. Massie, B. Chun, D. Culler, The Ganglia distributed monitoring system: design, implementation, and experience, Parallel Comput. 30 (7) (2004) 817–840.
[65] A. Meddeb, Internet QoS: Pieces of the Puzzle, IEEE Commun. Mag. 48 (1) (2010) 86–94.
[66] R. Mehrotra, A. Dubey, J. Kwalkowski, M. Paterno, A. Singh, R. Herber, S. Abdelwahed, RFDMon: a real-time and fault-tolerant distributed system monitoring approach (10/2011 2011).
[67] D.A. Menascé, P. Ngo, Understanding cloud computing: Experimentation and capacity planning, in: Computer Measurement Group Conference, 2009.
[68] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, D. Pendarakis, Efficient resource provisioning in compute clouds via VM multiplexing, in: Proceedings of the 7th International Conference on Autonomic Computing, ACM, 2010, pp. 11–20.
[69] Monitis (Jan. 2014). URL https://www.monitis.com/.
[70] S. Mongkolluksamee, Strengths and limitations of Nagios as a network monitoring solution, in: Proceedings of the 7th International Joint Conference on Computer Science and Software Engineering (JCSSE 2010), 2009, pp. 96–101.
[71] D.E. Morgan, W. Banks, D.P. Goodspeed, R. Kolanko, A computer network monitoring system, IEEE Trans. Softw. Eng. SE-1 (3) (1975) 299–311.
[72] J. Murphy, Snoscan: an iterative functionality service scanner for large scale networks (Ph.D. thesis), Iowa State University, 2008.
[73] Y. Natis, E. Knipp, R. Valdes, D. Cearley, D. Sholler, Who is who in application platforms for cloud computing: the cloud specialists, 2009.
[74] R. Olups, Zabbix 1.8 Network Monitoring, Packt Publishing, 2010.
[75] 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, in: PDT'11, ACM, 2011, pp. 3:1–3:2.
[76] M. Palacios, J. Garcia-Fanjul, J. Tuya, G. Spanoudakis, Identifying test requirements by analyzing SLA guarantee terms, in: Web Services (ICWS), 2012 IEEE 19th International Conference on, IEEE, 2012, pp. 351–358.
[77] C. Pape, R. Trommer, Monitoring VMware-based virtual infrastructures with OpenNMS, 2012.
[78] K. Park, J. Han, J. Chung, Themis: a mutually verifiable billing system for the cloud computing environment, in: IEEE Transaction on Service Computing http://dx.doi.org/10.1109/TSC.2012.1.
[79] M. Paterson, Evaluation of Nagios for real-time cloud virtual machine monitoring, 2009.
[80] Z.N. Peterson, M. Gondree, R. Beverly, A position paper on data sovereignty: the importance of geolocating data in the cloud, in: Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, 2011.
[81] T. Pueschel, D. Neumann, Management of cloud infrastructures: Policy-based revenue optimization, in: Thirtieth International Conference on Information Systems (ICIS 2009), 2009, pp. 1–16.
[82] M. Rak, S. Venticinque, T. Máhr, G. Echevarria, G. Esnal, Cloud application monitoring: the mOSAIC approach, in: Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on, IEEE, 2011, pp. 758–763.
[83] J.S. Rellermeyer, G. Alonso, T. Roscoe, Building, deploying, and monitoring distributed applications with Eclipse and R-OSGi, in: Proceedings of the 2007 OOPSLA Workshop on Eclipse Technology eXchange, in: eclipse '07, ACM, New York, NY, USA, 2007, pp. 50–54. http://dx.doi.org/10.1145/1328279.1328290.
[84] B. Rimal, E. Choi, I. Lumb, A taxonomy and survey of cloud computing systems, in: INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on, IEEE, 2009, pp. 44–51.
[85] F. Sabahi, Cloud computing security threats and responses, in: Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, 2011, pp. 245–249.
[86] D. Sanderson, Programming Google App Engine, O'Reilly Media, 2009.
[87] V. Sekar, P. Maniatis, Verifiable resource accounting for cloud computing services, in: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, ACM, 2011, pp. 21–26.
[88] A. Sekiguchi, K. Shimada, Y. Wada, A. Ooba, R. Yoshimi, A. Matsumoto, Configuration management technology using tree structures of ICT systems, in: Proceedings of the 15th Communications and Networking Simulation Symposium, Society for Computer Simulation International, 2012, p. 4.
[89] T. Somasundaram, K. Govindarajan, Cloud monitoring and discovery service (CMDS) for IaaS resources, in: Advanced Computing (ICoAC), 2011 Third International Conference on, 2011, pp. 340–345. http://dx.doi.org/10.1109/ICoAC.2011.6165199.
[90] S. Subashini, V. Kavitha, A survey on security issues in service delivery models of cloud computing, J. Netw. Comput. Appl. 34 (1) (2011) 1–11.
[91] D. Tovarnak, T. Pitner, Towards multi-tenant and interoperable monitoring of virtual machines in cloud, in: Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2012 14th International Symposium on, 2012, pp. 436–442.
[92] L.M. Vaquero, L. Rodero-Merino, D. Morán, Locking the sky: a survey on IaaS cloud security, Computing 91 (1) (2011) 93–118.
[93] S. Zanikolas, R. Sakellariou, A taxonomy of grid monitoring systems, Future Gener. Comput. Syst. 21 (1) (2005) 163–188.
[94] Q. Zhang, L. Cherkasova, E. Smirni, A regression-based analytic model for dynamic resource provisioning of multi-tier applications, in: Autonomic Computing, 2007. ICAC '07. Fourth International Conference on, 2007, http://dx.doi.org/10.1109/ICAC.2007.1.
[95] X. Zhang, J.L. Freschl, J.M. Schopf, A performance study of monitoring and information services for distributed systems, in: High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on, IEEE, 2003, pp. 270–281.
[96] W. Zhou, C. Chen, S. Li, Monitoring system for the campus card based on cacti and nagios, Shiyan Jishu yu Guanli 28 (4) (2011) 246–249.

Kaniz Fatema is a postdoctoral researcher at University College Cork focusing on cloud monitoring techniques, gathering and combining metrics from various layers of cloud and other topics related to cloud computing. Her Ph.D. research was conducted under an EC FP7 project (Trusted Architecture for Securely Shared Services) and focused on designing an authorisation system that provides privacy of personal data through access control policies and conflict resolution policies from multiple authorities dynamically, obtaining access control rules from the legislation (EU Data Protection Directive). She has multidisciplinary research experiences (in Information Security and Privacy, Privacy Law, Cloud computing, Speech Processing, Machine Learning, and Digital Signal Processing). Kaniz Completed her Ph.D. studies at the University of Kent, UK. She obtained her M.Sc. (Eng) in Data Communications from the University of Sheffield, UK with distinction. She did M.Sc. and B.Sc. (honours) in Computer Science and Engineering from the University of Dhaka, Bangladesh. She worked as a Lecturer at the Stamford University Bangladesh for 2 years and as an Assistant Lecturer at the University of Kent, UK for 3.5 years.

Vincent C. Emeakaroha is currently a postdoctoral researcher at the Irish Centre for Cloud Computing and Commerce with affiliation to University College Cork Ireland. Previously, he was a research assistant at the Distributed Systems Group, Information Systems Institute, Vienna University of Technology (TU Wien). He received his bachelor degree in Computer Engineering in 2006 and gained double masters in Software Engineering & Internet Computing in 2008, and in Informatics Management in 2009 all from Vienna University of Technology. In 2012, he received his Ph.D. in Computer Science with excellence from the same University. He has worked in different research projects over the years including the Foundations of Self governing ICT Infrastructures (FoSII) project funded by the Vienna Science and Technology Fund (WWTF) and Holistic Energy Efficient Approach for the Management of Hybrid Clouds (HALEY) funded by the Vienna University of Technology Science Award. He participated in the European Commissions COST Action on Energy Efficient Large Scale Distributed System. In October 2010, he was a visiting researcher at the Technical University of Catalunya Barcelona Spain. His research areas of interest include resource management in large scale distributed systems, Cloud monitoring, service provisioning management, autonomic computing, energy efficiency in Clouds, SLA and QoS management.

Philip Healy has over a decade's experience in parallel and distributed software development, in both academic and industrial settings. He completed his Ph.D. studies in 2006 in the area of cluster computing with FPGAs. His recent academic work includes the research and development of a remote monitoring system for physiological signals acquired in a critical care setting. His recent industry experience includes the design and development of large-scale distributed data processing applications using cutting-edge technologies such as Hadoop and HBase. He is currently a Research Fellow at The Irish Centre for Cloud Computing & Commerce.

John Morrison is the founder and director of the Centre for Unified Computing. He is a co-founder and co-director of the Boole Centre for Research in Informatics and a co-founder and co-director of Grid-Ireland. Prof. Morrison is a Science Foundation of Ireland Investigator award holder and has published widely in the field of Parallel Distributed and Grid Computing. He has been the guest editor on many journals including the Journal of Super Computing and the Journal of Scientific Computing. He is on the Editorial Board of Multi-Agent and Grid Systems: An International Journal, published by ISO Press, and the International Journal of Computational Intelligence: Theory and Practice (IJCITP). He is a member of the ACM and a senior member of the IEEE. Prof Morrison is a member of the I2Lab Advisory Board in the University of Central Florida. He has served on dozens of international conference programme committees and is a co-founder of the International Symposium on Parallel and Distributed Computing.

Theo Lynn is a Senior Lecturer at DCU Business School and is Principal Investigator of the Irish Centre for Cloud Computing and Commerce. Theo leads the Techspectations initiative, part of DCU Business Schools MarketingLab and teaches strategy and digital marketing at DCU Business School. He is a partner in The 30/60 Partnership, a seed capital investment fund for early stage technology startups and advises a number of domestic and international companies.