An Overview of User-level Usage Monitoring in Cloud Environment

Manoj Kesavulu, Duc-Tien Dang-Nguyen, Markus Helfert, Marija Bezbradica

Lero-The Irish Software Research Organization
School of Computing, Dublin City University, Dublin, Ireland
manoj.kesavulu2@mail.dcu.ie, {duc-tien.dang-nguyen, markus.helfert, marija.bezbradica}@dcu.ie

Abstract

Cloud computing monitors applications, virtual and physical resources to ensure performance capacity, workload management, optimize future application updates and so on. Current state-of-the-art monitoring solutions in the cloud focus on monitoring in application/service level, virtual and physical (infrastructure) level. While some of the researchers have identified the importance of monitoring users, there is still need for developing solutions, implementation and evaluation in this domain. In this paper, we propose a novel approach to extract end-user usage of cloud services from their interactions with the interfaces provided to access the services called User-level Usage Monitoring. We provide the principles necessary for the usage data extraction process and analyse existing cloud monitoring techniques from the identified principles. Understanding end-user usage patterns and behaviour can help developers and architects to assess how applications work and how users satisfy with the provided services.

Keywords: Cloud, Application, End-user, Usage Monitoring, Principles, Usage data

1 Introduction

In the last decade, we have witnessed the major change in software and applications in which cloud computing is becoming widely used, providing users with the possibility of using different devices to use (access) the cloud-based services seamlessly (Mell & Grance, 2011). The number of Cloud-based services has increased rapidly and strongly, offering various advantages over traditional software including reducing time to benefit, scalability, accessing through various interfaces and so on. However, it is also increased the complexity of the management of infrastructures behind these services. To properly operate and manage such complex infrastructures effective and efficient monitoring is constantly needed (Aceto, Botta, De Donato, & Pescapè, 2013).

Traditionally, the cloud provider (vendor) provides application performance management (APM) tools (for example, CloudWatch[1] in Amazon Web Services) to

[1] http://aws.amazon.com/cloudwatch/

monitor the status of the deployed applications, the amount of resources used by the applications based on the agreement between cloud vendor and the application provider called Service Level Agreement (SLA). These APM tools work at the infrastructure and service levels, providing mainly a vast amount of usage data of the resources used which can be turn into some knowledge for resource provisioning. However, it is nontrivial to obtain user-related information, for example how users satisfy with the given services or applications, from such kind of data. The application developers can also use various third-party monitoring tools like New Relic[2], Binado[3] and so on. But these tools mainly focus on monitoring application oriented usage including measuring the number of users logged-in to the application, identifying rare logins, cloud resource usage, idle times, license types etc. We firmly believe that exploiting usage data at an user level could give much more insights for the application development. Understanding usage data of an application has various uses such as to personalise the application according to the end-user's preferences (Yang et al., 2017), profiling users for security (Al-Bayati, Clarke, & Dowland, 2016), improvement in marketing of software products (Bucklin & Sismeiro, 2009) and to analyse the performance of the application in the deployed environment for maintenance purposes (Bezemer, Zaidman, Platzbeecker, Hurkmans, & Hart, 2010; Petruch et al., 2012).

In this paper, we provide principles for a cloud monitoring tool, extracted from cloud standards such as ISO and TOG. Furthermore, we analyse the existing state-of-the-art monitoring solutions with respect to the monitoring level in cloud. As a result of the analysis, we have identified user-level usage monitoring as the research gap. With the improving of the data mining tools, these usage data can be gathered from online services by collecting all traces of user activity to produce clickstreams, sequences of timestamped events generated by user actions. For example, in web-based services, these might include detailed HTTP requests. For mobile applications, clickstreams can include everything from button clicks, to finger swipes and text or voice input (Wang, Zhang, Tang, Zheng, & Zhao, 2016). By using user-level usage monitoring, we believe the following challenges can be addressed:

---

[2] https://newrelic.com/

[3] https://www.binadox.com/salesforce-saas-monitoring/

- *Usage based metering/billing:* user-level usage data helps cloud provider to design the billing policy to reflect the actual usage of the application by the end-user.
- *Resource provisioning:* based on the usage data, predict the resources that may be allocated to an application.
- *Focused application updates:* developers can determine features of the application that are critical to end-user. Hence, focus the development costs and time on such features.
- *Understanding user satisfaction:* instead of surveying and asking feedback, how users satisfy with an application can be revealed via their usage data.
- *Discovering user behaviour patterns:* Every user has their own pattern when using an application or a service. Understanding these patterns could help improve the service or discover the trends in advance. These patterns, can be discovered from the usage data.

Analysing and understanding the usage data from the user's perspective can be used by the software developers and software architects to determine how much development time, development cost to allocate and spend for which features of the cloud application before rolling out new updates. As a part of our work, we aim to build the usage data extraction artefact and follow the evaluation approach using Design Science Research (Helfert, Donnellan, & Ostrowski, 2012). The terms *cloud service consumer, customer* and *end-user* mean the same and are used interchangeably in this paper.

The remainder of the paper is structured as follows: in Section 2, we discuss cloud standards, specifically the monitoring aspect of the cloud and propose the principles for a cloud monitoring tool/solution. In Section 3, we review the state-of-the-art cloud monitoring solutions. In Section 4, we present a comparative analysis of the cloud monitoring tools and as a result identify the research gap, usage monitoring. In Section 5, we provide conclusions and directions for future work.

2 Monitoring principles from cloud standards

Since the advent of cloud computing, various monitoring solutions emerged. However, existing cloud systems and enterprises incorporating them normally follow different architectures and standards bringing a vast amount of challenges in communications as well as organisations for the applications and services in the cloud environment. As a result, the new services, applications and the monitoring solutions has the need to follow the principles set forward by the cloud standards (International Organization for standardization (ISO) defines *standard* as “specifications for products, services and systems, to ensure quality, safety and efficiency” (ISO, n.d.)). Monitoring at user-level, consequently, should follow criteria and requirements as in other levels. For this purpose, in this section, we discuss and review widespread cloud standards and provide the principles a cloud monitoring tool should follow.

2.1 International Organization for Standardization

ISO in collaboration with International Electrotechnical Commission (IEC)[4] drafted the “Information technology — Cloud computing — Reference architecture” document known as ISO/IEC 17789:2014 (ISO/IEC, 2014). This International Standard specifies the cloud computing reference architecture (CCRA). The reference architecture includes the cloud computing roles, activities, functional components and their relationships. The standard describes the activities of various components of the cloud. In this section, we focus our discussion on the activities of the monitoring component of the cloud. The monitor service activity monitors the delivered service quality with respect to service levels as defined in the service level agreement (SLA) between cloud service customer and cloud service provider. This activity uses the built-in monitoring functions of the cloud system. The ISO standard describes the following responsibilities of the monitoring activity:

- keeping track of how much use is being made of each cloud service, and by which users. This includes assurance that the use is appropriate;
- monitoring the integration of the cloud services with customer's existing ICT systems to ensure that business goals are being met;
- defining measurement points and performance indicators related to the service in question (e.g., service availability, service outage frequency, mean time to repair, responsiveness of the provider's help desk, etc.);
- monitoring, analysing and archiving of these indicator data;
- comparing the actual service quality delivered with the agreed service quality

[4] http://www.iec.ch/

The standard also specifies integration of existing Information and Communications Technology (ICT) components and application with the target cloud services and its implications on the monitoring component which involves creating and monitoring specific user accounts and identities use of management interfaces for cloud services and integrating logging and security incident management between cloud services and user monitoring and management infrastructure. The user interface through which an end-user interacts with cloud service provider and with cloud services, performs customer related administrative activities, and monitors cloud services is described as user layer in the standard. A user interface is typically a thin client interface such as a web-browser, smartphone app or a command-line interface, can be collectively called “front-end interfaces” (Kesavulu, Helfert, & Bezbradica, 2017). A monitoring functional component in a cloud environment should provide the following capabilities:

- monitor the activities of functional components throughout the cloud service provider's system. This includes the components that are involved in the direct use of cloud services by the end-user: cloud service users including the service access and service implementation (e.g., the invocation of a cloud service operation by a specific user);
- report time-sensitive critical events based on monitoring cloud provider’s system behaviour (e.g., the occurrence of a fault, the completion of a task), or log system execution in the form of historical data (e.g., service usage data);
- storage and retrieval of data obtained from monitoring activity as logging records. The monitoring component is also responsible to guarantee the availability, confidentiality and integrity of the logging records.

2.2 The Open Group Standard

The Open Group (TOG)[5] is a global consortium that enables the achievement of business objectives through IT standards. They provide a standard for cloud computing called “The Open Group Cloud Ecosystem Reference Model” which defines the cloud reference model and provides guidance on how to apply it with The Open Group Architecture Framework (TOGAF) and ArchiMate (open and

[5] http://www.opengroup.org/

independent modelling language for Enterprise Architecture) standards to develop an Enterprise Architecture (The Open Group, 2014).

TOG standard specifies the following three activities of a monitoring component:

- Monitoring subscription (SLA monitoring): Service Providers design and utilise multiple subscription models for charging users based on resource usage by the end-users. Some examples of subscription models may include fixed, tier-based (e.g., Gold, Silver, and Platinum), pay-as-you-go payment terms (monthly, quarterly, annually). Monitor allocation and consumption of Cloud Services to enable cloud service providers to facilitate charge-back to their subscribed consumers based on subscription models.
- Resource Health Monitoring: provides a broad view of issues that impact cloud resources with the aim to improve performance, accountability, and business results. This includes identifying, diagnosing, reporting of the issues affecting the virtual and physical cloud resources.
- Service Health Monitoring: is similar to Resource health monitoring but the focus here is on the services provided by the cloud provider. In addition to identification, diagnosis and reporting, this activity is also responsible for providing tools to monitor defined SLAs.

#	Principle for monitoring in the cloud	Standard	
		ISO	TOG
P1	Monitor delivered service quality as defined in SLA	✓	✓
P2	Monitor usage of services by user	✓	✓
P3	Monitor the integration of the cloud services with customer's existing ICT systems	✓	
P4	Monitoring component should ensure analysing and archiving of monitored data	✓	
P5	User interface should be provided to the cloud provider and user to manage the monitoring tasks and visualise the results	✓	
P6	Monitoring component should guarantee availability, confidentiality and integrity of the logging records	✓	
P7	Consider different subscription models to define monitoring metrics		✓

Table 1: Cloud Monitoring Principles

3 Existing cloud monitoring solutions

Cloud providers offer diverse services to the cloud users using proprietary software and management techniques. Many of these providers use provider-dependent monitoring tools which complement their offerings. In addition, many monitoring solutions are being developed by researchers, enterprises. In this section, we review the state-of-the-art cloud monitoring solutions inspired by recent cloud monitoring survey paper by (Syed, Gani, Ahmad, Khan, & Ahmed, 2017).

3.1 PCMONS

The authors in De Chaves, Uriarte, & Westphall (2011) proposed an open-source architecture for cloud monitoring. The authors propose a three-layered architecture: (i) infrastructure layer; (ii) integration layer and (iii) view Layer. Infrastructure layer consists of basic hardware, software, network and operating system. Integration layer is responsible for visualisation environment and hypervisors to acquire infrastructural related information. The view layer is responsible for presenting the monitoring data appropriately to the type of user (here, a user represents actors such as developer, administrator or a manager, not an end-user). The authors also demonstrate the PCMONS tool in this paper using agent insertion based monitoring methodology (for every new VM). This method creates additional overhead affecting VMs performance. The monitoring component in this paper is called as VM monitor, which injects scripts into the VMs that send useful data (for example, processor load and memory usage) from the VM to the monitoring system.

3.2 GmonE

The authors in Montes, Sánchez, Memishi, Pérez, & Antoniu (2013) present a method to categorise the monitoring solutions according to monitoring level and vision, where monitoring level deals with layers of cloud computing as defined in Mell & Grance (2011) and cloud vision defines how to distinguish monitoring data to analyse and present to different actors (such as end-users, developers, architects, managers and so on). The authors define client-side monitoring vision as the client’s view of the cloud and deals with presenting the monitoring data to clients in terms of SLA agreement and contracts. The authors have proposed layered cloud monitoring architecture called GMonE, which is composed of four key components including GMonEMon, monitoring Plug-ins, GMonEDB and GMonEAccess. The authors claim GMonEMon can run in any component of cloud that needs to be monitored to collect and send metric data to the GMonEDB and are implemented in the form of plugins. Monitoring data include status of the VMs, simultaneous network connections, application usage patterns. GMonEDB is responsible for receiving monitoring data and manages it for GMonE as a database. The GMonEAccess is a user interface which provides visualisation of monitored data.

3.3 NFM

In Suneja, Isci, Koller, & de Lara (2016), authors have proposed a novel cloud monitoring technique called Near Field Monitoring (NFM). The monitoring process is instantiated without inserting any agent into the user space. The operational logic of NFM include VM introspection using kernel data to extract system state and container namespace mapping, which enables the monitoring component to run irrespective of health of VM/containers. In NFM, a user/host can subscribe and unsubscribe the monitoring service as it runs independent of VM/container. Monitoring user interaction and user behaviour is not considered in the context of this paper.

3.4 MonSLAR

In Al-Shammari & Al-Yasiri (2015), the authors have presented a monitoring architecture called MonSLAR. The proposed architecture comprises of two versions of middleware, one for the user (client) side and one for the provider (server) side. Both the versions use REST (Representational State Transfer) protocols to dispatch requests and receive responses between client and server sides, thus enabling monitoring of components. MonSLAR provides information to the users about SLA if the services used by the user meet the agreed upon metrics. For the service provider, MonSLAR provides a method to measure the cloud user’s satisfaction using a combination of network Quality of Service (QoS) and SLA parameters and term this as Quality of Experience (QoE). However, this work does not consider user’s interaction with the cloud applications and the related implications on the cloud resources.

3.5 MonPaaS

In Alcaraz Calero & Aguado (2015) the authors present an open-source adaptive monitoring platform as a service (MonPaaS) tool. The proposed tool has two different monitoring modes including cloud provider monitoring and user monitoring (capability provided to user for monitoring the deployed cloud resources). MonPaas is implemented by integrating Nagios[6] and OpenStack. This system intercepts the message queue of OpenStack, and use these messages to provide information about VMs. Both cloud providers and users can access the MonPaaS module in the form of API. The monitoring logic includes creation of separate VM for each new user, this creates additional performance overhead on the system. MonPaas uses Nagios for distributed monitoring, and DNX and Nconf to provide graphical management interface. MonPaas monitors physical and virtual resources and updates any change in physical or virtual infrastructure. However, user-level monitoring is not considered.

3.6 Monitoring-As-a-Service OCCI API

In Ciuffoletti (2016) the author proposes an on-demand monitoring as-a-service model as an extension to Open Cloud Computing Interface (OCCI) API[7], an open source IaaS service API that provides some standards and protocols for the cloud systems. The monitoring logic introduces a monitoring agent called "Sensor". Users can define the monitoring metric data through mixins, the sensor collects these user-defined metric data. Mixins have three different features including *metric* which defines the functionality of the requested entity, *aggregator* that defines how raw measurements should be processed, and *publisher* that defines how the metrics are used. The author also presents the monitoring extension as a prototype based on Docker. Although the focus of this work is to provide capability to user to define the monitoring metrics, only physical and virtual resources are monitored. User-level monitoring is not considered in the context of this paper.

3.7 DB Management Framework

In Zhao, Sakr, & Liu (2015), authors have presented a framework for the management of cloud-based database; with the aim to identify the consumer requirement to meet the terms defined in the SLA. The architecture of the proposed framework consists of

[6] https://www.nagios.org/
[7] http://occi-wg.org/

three modules: (i) the monitor module; (ii) the control module and (iii) the action module. The monitor module is responsible to gather information based on two metrics: (i) data freshness and (ii) transaction response time. Proposed model monitors database services and performs adaptive actions to avoid any violation of SLA defined by specific application. User behaviour or interactions are not considered in the context of this work.

3.8 SLA Monitoring

Service Level Agreement specifies terms and conditions of cloud services agreed between a cloud service provider and cloud service consumer. The SLA parameters need to be monitored to avoid SLA violation, which can result in the form of financial penalty. In Anithakumari & Chandrasekaran (2017), the authors have used monitoring techniques to analyse the parameters of SLA with the aim to predict any possible violation. The monitoring component monitors the Service Level Objective (SLO) values such as response time and job execution time from all the running instances, which forms the basis for determining SLA violations. In case, if SLA is not met, the penalty imposed is presented in SLO. Authors have also proposed an adaptive resource management. In this approach, additional resources (more VMs) are deployed to run when an SLA violation is predicted or occurred, with the aim to execute the current job and mitigate future SLA violations. Authors have also presented a prototype using GMOND module provided by Ganglia (Massie, 2004) for runtime monitoring and Java messaging service (JMS) and MySQL is used as a database. The emphasis of this work is mainly on monitoring SLA parameters on the server side. However, the implication of user interaction and user behaviour on SLA are not considered.

3.9 Dynamic Pricing Policy

In Anwar et al., (2015), a dynamic pay-per-usage charging solution for the cloud service providers is presented. By utilising monitoring agent, they have proposed a solution for charging with less overhead. The authors used OpenStack's Ceilometer to collect metering data. The advantage of this approach is that instead of using separate VMs for management (i.e. metering/monitoring etc.) they have utilised resources of the same VM for monitoring purpose. Additionally, the system automatically allocate new VM if the existing running VMs reaches maximum load. A down-side of this approach is that the additional overhead on the performance of VMs. The main focus of this work is metering and the authors have only considered monitoring physical and virtual resources. User-level usage based metering is not considered.

3.10 Power and Performance Management

Users and user-side applications typically do not have access to information on cloud software and hardware resource utilisation and power consumption. Alternatively, public cloud offers little access to the information about user application requirement. Turk et al., (2016) aim to address this issue by proposing an architecture for monitoring by providing detailed information about the different layers of cloud for users and providers. This work utilises the Massachusetts Open Cloud (MOC), a public cloud established for research purpose. In the proposed work, authors focus on using cloud monitoring for power and performance management in the cloud data-centres. The proposed architecture is divided into four layers including Data collection layer, Data retention & consolidation layer, Services layer, and Advanced monitoring applications layer. The authors have used a combination of Sensu (open-source monitoring software), Ceilometer, LogStach (data acquisition and transport tool) and RabbitMQ (open-source message queue tool) for acquisition and collection of Data, and integrated InfluxDB, Elastic-Search, and MonoDB for database purpose in their proposed architecture. The monitoring component in this architecture monitors the cloud hardware resources and user-level usage and its implication and effects on power consumption is not considered.

4 Comparative analysis of cloud monitoring solutions

This section presents a comparative analysis of the cloud monitoring solutions discussed in Section 3. The focus of the analysis here is to identify the solutions based on the monitoring level (User, Application/Service, Infrastructure/Resource level) in cloud, techniques followed and implementation status of user-level usage monitoring.

User-level usage monitoring represents the usage data generated in the cloud system due to the user’s interaction with the cloud application. The authors in Montes et al., (2013) have considered user-level usage monitoring in their taxonomy but have not implemented in the GmonE tool. The author in Ciuffoletti (2016) introduces a monitoring agent named as the “Sensor”. The sensor collects metric data, defined in

Monitoring Tools	**Monitoring level**			**User-level Monitoring Status**	**Monitoring technique**
	User	**Application / Service**	**Infrastructure / Resource**		
PCMONS	X	✓	✓	Not Considered	Scripts in VM
GmonE	✓	X	✓	Identified, Not Implemented	Plugins in VM
NFM	X	X	✓	Not Considered	Kernel data in VM
MonSLAR	✓	X	✓	Identified (User satisfaction for SLA)	REST Protocol
MonPaaS	X	X	✓	Not Considered	Special Monitor VM
Monitoring-as-a-Service OCCI API	X	X	✓	Not Considered	Server application
DB Management Framework	X	✓	X	Not Considered	Database proxy software
SLA Monitoring	X	✓	✓	Not Considered	GMOND tool by Ganglia
Dynamic Pricing Policy	X	X	✓	Not Considered	OpenStack Ceilometer[8]
Power and Performance Management	X	X	✓	Not Considered	Sensu, Ceilometer, LogStach

Table 2: Comparative Analysis of Cloud Monitoring Solutions

[8] Ceilometer is an OpenStack service, used for metrics collection for billing

Mixins by users. But the important thing to note here is that a user defines the metrics of the monitoring agent, which is different from user-level usage monitoring as the user interaction is not monitored. In Alcaraz Calero & Aguado (2015), the authors use the term user monitoring, where a separate VM called Monitor VM (MVM) is created for each new customer. Each MVM monitors physical and virtual resources but not the user interaction. Similarly, majority of the monitoring solutions consider users in the cloud monitoring domain but user-level usage monitoring and its implications on the service and infrastructural resource usage in the cloud have not been considered.

Different cloud monitoring tools contribute to different characteristics of the cloud including metering, billing, SLA management, error and fault fixing, resource provisioning, workload management, and so on. In Kesavulu et al., (2017), the authors have defined criteria for the user-level usage data and proposed a usage data extraction framework adhering to the defined criteria. The idea of monitoring user behaviour is to understand how users interact with the application and this is mainly done through analysing the clickstreams (Banerjee & Ghosh, 2001; Bucklin & Sismeiro, 2009; Pachidi et al., 2014; Wang, Zhang, Tang, Zheng, & Zhao, 2016). The authors Cito et al. (2015) provide a high-level taxonomy of types of operation data that can be treated as user-level usage data:

- Monitoring data (Operational application metadata)
  - Performance data – service response times, database query times
  - Load data – incoming request rate, server utilisation
  - Costs data – hourly cloud virtual machine costs, data transfer costs per 10,000 page views
  - User behaviour data – clickstreams, page views,
- Production data
  - Data produced by SaaS application itself-placed orders, customer information.

Summarising in Table 2 are the current tools and applications for usage monitoring in the cloud domain. These tools and applications are suggested in Syed et al., (2017) showing that major of them are working on the monitoring data at service and infrastructure level while only GmonE and MonSLAR have identified the potential of user level monitoring (not implemented). This confirms our motivations to exploit and explore researches on this novel type of data to cope with the challenges pointed out in Section 1.

5 Conclusion and future work

In this paper we reviewed state-of-the-art cloud monitoring solutions that have considered the user's perspective as a part of their tools. Furthermore, we have analysed the monitoring solutions according to their level of monitoring (user, application/service, infrastructure/resources) in the cloud and techniques used or adopted for the actual monitoring task. As a result of the analysis, we see that existing cloud monitoring solutions consider users in the cloud monitoring domain but user-level usage monitoring and its implications on the service and infrastructural resource usage in the cloud has not been considered. Consequently, we presented the related standards in ISO and TOG based on those, proposed the principles for cloud monitoring solutions to follow. We firmly believe that understanding the usage patterns of the end-users and usage behaviour can overcome the challenges mentioned in Section 1. The future work of this research includes (1) design and development of a novel approach to monitor cloud application usage by end-users, namely User-level Usage Monitoring, which is the process of identification, extraction and analysis of the data that represent users' interaction with the cloud application (2) extending the review by considering other cloud standards and any principles that may reveal; (3) evaluation of the user-level usage monitoring tool.

Acknowledgement

This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

References

Aceto, G., Botta, A., De Donato, W., & Pescapè, A. (2013). Cloud monitoring: A survey. *Computer Networks*, *57*(9), 2093–2115. https://doi.org/10.1016/j.comnet.2013.04.001

Al-Bayati, B., Clarke, N., & Dowland, P. (2016). Adaptive Behavioral Profiling for Identity Verification in Cloud Computing: A Model and Preliminary Analysis. *GSTF Journal on Computing*, *5*(1), 21–28. https://doi.org/10.5176/2251-3043_4.4.348

Al-Shammari, S., & Al-Yasiri, A. (2015). MonSLAR: A middleware for monitoring SLA for RESTFUL services in cloud computing. *2015 IEEE 9th International Symposium on the Maintenance and Evolution of Service-Oriented Systems and Cloud-Based Environments, MESOCA 2015 - Proceedings*, 46–50.

https://doi.org/10.1109/MESOCA.2015.7328126
Alcaraz Calero, J. M., & Aguado, J. G. (2015). MonPaaS: An adaptive monitoring platformas a service for cloud computing infrastructures and services. *IEEE Transactions on Services Computing*, *8*(1), 65–78. https://doi.org/10.1109/TSC.2014.2302810
Anithakumari, S., & Chandrasekaran, K. (2017). Negotiation and Monitoring of Service Level Agreements in Cloud Computing Services. *Proceedings of the International Conference on Data Engineering and Communication Technology: ICDECT 2016, Volume 2*, 651–659. https://doi.org/10.1007/978-981-10-1678-3_62
Anwar, A., Sailer, A., Kochut, A., Schulz, C. O., Segal, A., & Butt, A. R. (2015). Cost-Aware Cloud Metering with Scalable Service Management Infrastructure. *Proceedings - 2015 IEEE 8th International Conference on Cloud Computing, CLOUD 2015*, 285–292. https://doi.org/10.1109/CLOUD.2015.46
Bezemer, C. P., Zaidman, A., Platzbeecker, B., Hurkmans, T., & Hart, A. (2010). Enabling multi-tenancy: An industrial experience report. *IEEE International Conference on Software Maintenance, ICSM*. https://doi.org/10.1109/ICSM.2010.5609735
Bucklin, R. E., & Sismeiro, C. (2009). Click Here for Interact Insight: Advances in Clickstream Data Analysis in Marketing. *Journal of Interactive Marketing (Mergent, Inc.)*, *23*(1), 35–48. https://doi.org/10.1016/j.intmar.2008.10.004
Ciuffoletti, A. (2016). Application level interface for a cloud monitoring service. *Computer Standards and Interfaces*, *46*, 15–22. https://doi.org/10.1016/j.csi.2016.01.001
De Chaves, S. A., Uriarte, R. B., & Westphall, C. B. (2011). Toward an architecture for monitoring private clouds. *IEEE Communications Magazine*, *49*(12), 130–137. https://doi.org/10.1109/MCOM.2011.6094017
Helfert, M., Donnellan, B., & Ostrowski, L. (2012). The case for design science utility and quality-Evaluation of design science artifact within the sustainable ICT capability maturity framework. *Systems, Signs & Actions*, *6*(1), 46–66. Retrieved from http://www.sysiac.org/uploads/sysiac2012-helfert.pdf
ISO/IEC. (2014). *ISO/IEC 17789 - Information technology - Cloud computing - Reference architecture*. *ISO* (Vol. 2014).
Kesavulu, M., Helfert, M., & Bezbradica, M. (2017). A Usage-based Data Extraction Framework for Cloud-Based Application - An Human-Computer Interaction approach. In *International Conference on Computer-Human Interaction Research and Applications (CHIRA)*. Madeira, Portugal.
Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. *NIST Special Publication*, *145*, 7. https://doi.org/10.1136/emj.2010.096966
Montes, J., Sánchez, A., Memishi, B., Pérez, M. S., & Antoniu, G. (2013). GMonE: A complete approach to cloud monitoring. *Future Generation Computer Systems*, *29*(8), 2026–2040. https://doi.org/10.1016/j.future.2013.02.011
Petruch, K., Tamm, G., & Stantchev, V. (2012). Deriving In-Depth Knowledge from IT-Performance Data Simulations. *International Journal of Knowledge Society Research*, *3*(2), 13–29. https://doi.org/10.4018/jksr.2012040102
Suneja, S., Isci, C., Koller, R., & de Lara, E. (2016). Touchless and always-on cloud analytics as a service. *IBM Journal of Research and Development*, *60*(2–3), 11:1-11:10. https://doi.org/10.1147/JRD.2016.2518438
Syed, H. J., Gani, A., Ahmad, R. W., Khan, M. K., & Ahmed, A. I. A. (2017). Cloud monitoring: A review, taxonomy, and open research issues. *Journal of Network*

and Computer Applications, *98*(March), 11–26. https://doi.org/10.1016/j.jnca.2017.08.021
The Open Group. (2014). *The Open Group Cloud Ecosystem Reference Model*.
Turk, A., Chen, H., Tuncer, O., Li, H., Li, Q., Krieger, O., & Coskun, A. K. (2016). Seeing into a Public Cloud: Monitoring the Massachusetts Open Cloud. *USENIX Workshop on Cool Topics on Sustainable Data Centers (CoolDC 16)*, 2–7.
Wang, G., Zhang, X., Tang, S., Zheng, H., & Zhao, B. Y. (2016). Unsupervised Clickstream Clustering for User Behavior Analysis. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, 225–236. https://doi.org/10.1145/2858036.2858107
Yang, J., Wang, H., Lv, Z., Wei, W., Song, H., Erol-Kantarci, M., … He, S. (2017). Multimedia recommendation and transmission system based on cloud platform. *Future Generation Computer Systems*, *70*, 94–103. https://doi.org/10.1016/j.future.2016.06.015
Zhao, L., Sakr, S., & Liu, A. (2015). A Framework for Consumer-Centric SLA Management of Cloud-Hosted Databases. *IEEE Transactions on Services Computing*, *8*(4), 534–549. https://doi.org/10.1109/TSC.2013.5