Towards Independent Run-time Cloud Monitoring

Luuk Klaver
Monitoring & Control Services
TNO
The Hague, Netherlands
luuk.klaver@tno.nl

Thijs van der Knaap
Monitoring & Control Services
TNO
Groningen, Netherlands
thijs.vanderknaap@tno.nl

Johan van der Geest
Monitoring & Control Services
TNO
Groningen, Netherlands
johan.vandergeest@tno.nl

Edwin Harmsma
Monitoring & Control Services
TNO
Groningen, Netherlands
edwin.harmsma@tno.nl

Bram van der Waaij
Monitoring & Control Services
TNO
Groningen, Netherlands
bram.vanderwaaij@tno.nl

Paolo Pileggi
Monitoring & Control Services
TNO
The Hague, Netherlands
paolo.pileggi@tno.nl

ABSTRACT

Cloud computing services are integral to the digital transformation. They deliver greater connectivity, tremendous savings, and lower total cost of ownership. Despite such benefits and benchmarking advances, costs are still quite unpredictable, performance is unclear, security is inconsistent, and there is minimal control over aspects like data and service locality. Estimating performance of cloud environments is very hard for cloud consumers. They would like to make informed decisions about which provider better suits their needs using specialized evaluation mechanisms. Providers have their own tools reporting specific metrics, but they are potentially biased and often incomparable across providers. Current benchmarking tools allow comparison but consumers need more flexibility to evaluate environments under actual operating conditions for specialized applications. Ours is early stage work and a step towards a monitoring solution that enables independent evaluation of clouds for very specific application needs. In this paper, we present our initial architecture of the Cloud Monitor that aims to integrate existing and new benchmarks in a flexible and extensible way. By way of a simplistic demonstrator, we illustrate the concept. We report some preliminary monitoring results after a brief time of monitoring and are able to observe unexpected anomalies. The results suggest an independent monitoring solution is a powerful enabler of next generation cloud computing, not only for the consumer but potentially the whole ecosystem.

CCS CONCEPTS

• **Information systems** → **Information system applications** → **Decision support systems** → Online analytical processing

ICPE '21 Companion, April 19–23, 2021, Virtual Event, France

ACM ISBN 978-1-4503-8331-8/21/04…$15.00. https://doi.org/10.1145/3447545.3451180

KEYWORDS

Cloud computing, Run-time monitoring, Performance evaluation, Benchmarking

ACM Reference format:

Luuk Klaver, Thijs van der Knaap, Johan van der Geest, Edwin Harmsma, Bram van der Waaij and Paolo Pileggi. 2021. Towards Independent Run-time Cloud Monitoring. In *Proceedings of 2021 ACM/SPEC International Conference on Performance Engineering (ICPE'21), 19 – 23 April, Virtual Event, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3447545.3451180

1 INTRODUCTION

Cloud environments, henceforth called *cloud*[1], are vital to the digital transformation of many businesses. It connects customers better and lowers the total cost of ownership due to its flexibility in resource rental, bringing greater value and savings. Key aspects of a cloud strategy are service offering, pricing and provider presence.

Despite unpredictable costs, unclear performance, minimal data or assets control, and inconsistent security, the public cloud proliferates [1, 2]. Demand for accurate and independent cost and performance information is increasing due to increasing popularity of multi-cloud setups or even cloud federation approaches.

Cloud ecosystem participants, like application owners or developers, i.e., *consumer*, need flexible and effective ways to access run-time cloud monitoring easily. Such enabling technology will provide a mechanism for accurate and custom assessments to suit more complex application needs and promote strong antitrust.

Paradoxically, the market is both rather mature yet quite radical. Defined a decade ago, the cloud standards roadmap of the National Institute of Standards (NIST) reference architecture [3] was recently elaborated to have over five times as many roles [4].

---

[1] For the sake of readability of the manuscript, *cloud* refers to cloud computing, as well as the computing solutions offered by cloud providers.

The coveted ecosystem lock-in is controlled by the largest cloud providers but their position is being challenged by, for example, the GAIA-X project [5]. This initiative, still in its nascent stage, challenges the openness, interoperability and transparency of these behemoths to regain and advance socio-economic and digital sovereignty and independence of cloud for Europe.

Currently, the largest providers are Amazon, Microsoft, Alibaba, Google and Tencent [6]. With the significant power this position gives them, there needs to be a mechanism by which they can be held accountable to what they promise.

In this paper, we present our initial architecture of an independent cloud monitoring tool, called the Cloud Monitor, that aims to integrate existing and new cloud benchmarks in a flexible and extensible way. It leverages container and orchestration applications, *Docker* and *Kubernetes*, respectively, to build an extensive monitor for the cloud. This work is in an early stage and is a step towards a solution that enables independent evaluation of clouds with specialized application needs and whose applications operate differently under specific conditions.

By way of a simplistic demonstrator, we illustrate the concept. Two different cloud providers, one well-known and the other not, are monitored and some preliminary results are discussed from consumer and other ecosystem participant perspectives. After a brief time of monitoring, we were able to observe unexpected anomalies.

In the following section, we give background about the cloud ecosystem and benchmarking technologies, followed by related work in Sect. 3. In Sect. 4, we present our approach from concept to implementation. In Sect. 5, we present our real-world demonstrator. Finally, we conclude with future work.

2 BACKGROUND

Cloud involves the delivery of public or private services, like storage, compute and networking, delivered over the internet. This makes it possible to access computing on demand and remotely.

The NIST cloud computing reference architecture [3] comprises five major cloud actors, i.e., the Consumer, Provider, Broker, Auditor, and Carrier, each having a specific role in the ecosystem. Böhm et al. [7] and Floerecker et al. [4] elaborate this with role-based models. Passau Cloud Computing Ecosystem (PaCE) Model by Floerecker et al. comprises twenty-six roles for the market, grouped into five categories, namely Client, Vendor, Hybrid, Support and Environment.

The ecosystem and value chain, e.g., the Cloud Value Chain Reference Model [8], are quite complex with only key participants having significant influence. World citizens are all subject to cloud-based solutions supported by this ecosystem.

From the EU perspective, it is imperative to have a secure federated ecosystem with the high standards of data sovereignty and compliance. GAIA-X [5], promotes open, transparent and trusted digital ecosystems with available, accessible and shared data and services. With already more than three-hundred organizations, they aim to promote a more economically sustainable and diverse ecosystem for a safer and fairer digital Europe.

2.1 Benchmarking clouds

Benchmarking is the practice of comparing performance and business practice to a typically standard, or standardized, reference. Cloud benchmarking [9, 10] is well-studied and quite mature standards are readily available. Microbenchmarks focus on a specific aspect of the system. It might evaluate a single metric in a system, which can form part of a key performance indicator (KPI) that comprises a set of metrics. Laaber et al. [11] show just how insightful and powerful a single cloud microbenchmark can be.

The Standard Performance Evaluation Corporation (SPEC) develops a benchmark suite for cloud performance. The SPEC Cloud ® 2018 benchmark [12] builds on the original 2016 release with a variety of enhancements, like new key metrics. Specifically, it addresses the performance of IaaS platforms.

Market research gives insight about the largest providers and the challenge of cloud performance information: Cisco makes some interesting discoveries [13], e.g., not all clouds rely heavily on public Internet for data transport. More importantly, significant performance anomalies exist that depend on provider, consumer location, and hosting region.

Understanding how a cloud performs, for applications that behave differently under different conditions, is only one of many benchmarking challenges. Consider the different actors, roles and stakeholders in the model, ecosystem, and value chain. Next generation clouds need to perform for different kinds of KPIs. Varghese and Buyya [14] identified research directions for next generation clouds to include sustainability, security, expressivity, marketplace, management and reliability. Benchmarks need to be flexible enough to address different metrics without being an obstacle for the users of these tools, or the interpreter of its results.

3 RELATED WORK

The suitability of a performance metric depends on the computing model or layer and the application characteristics [15]. While being hard to select metrics based on a specific application, careful consideration avoids misleading information about actual performance. Typical metrics reported include resource utilization, response time, energy consumption and operating cost [16, 17, 18].

SLA violation and cloud cost [15] have been measured, also for hybrid environments like cloud edge. Jha et al. [18] use simulation as an alternative to monitoring cloud. Despite promising results, measuring costs accurately is still a challenge.

Cockroach Labs developed a freely-available tool that assesses and compares Extract, Transform, Load systems using popular cloud provider instances [19]. They manually schedule microbenchmarks that give insight into the CPU, network and disk access performances of e-commerce applications. Designed to stress provisioning and run-time aspects of a cloud using disk— and CPU-intensive workloads, their benchmark runs until it reaches specific quality of service (QoS) conditions. KPIs include a Performance Score and Relative Scalability.

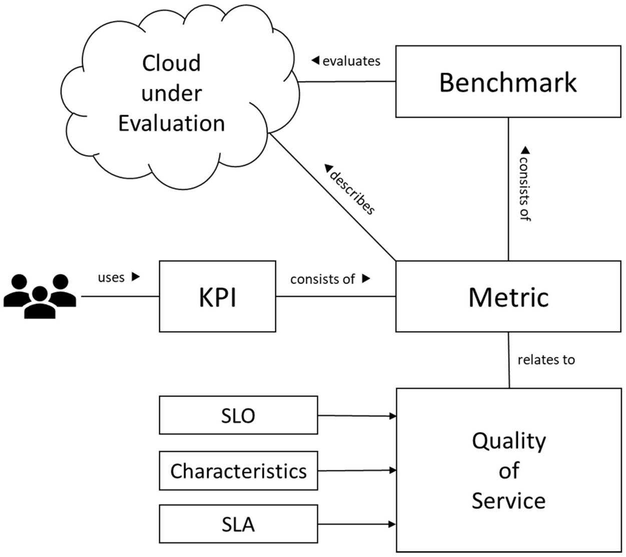

Figure 1: Cloud Monitor concepts and associations

Google's PerfKit Benchmarker[2] is another open-source tool that makes benchmarking faster and easier by automating networking setup, Virtual Machine (VM) provisioning, and test runs [20]. It supports a broad range of network performance tests like VM-to-VM latency, throughput, and packets-per-second for multiple clouds. Results can be viewed the results in the free Google Data Studio[3]. Moreover, a benchmarking methodology using PerfKit Benchmarker continuously and consistently for performance measurement and benchmarking was co-developed to help identify trends and patterns.

Monitoring is also an essential part of service mesh technology, like *Istio*. However compared to our envisioned external monitoring approach, services meshes are applicable for internal application monitoring purposes, monitoring signals related to latency, traffic, errors and saturation, for example [21].

The Cloud Monitor extends the state of the art by aiming for greater ease-of-use for custom monitoring tasks. The main focus is greater flexibility and extensibility for the user of the monitor.

4 CLOUD MONITOR APPROACH

4.1 Principal Concepts

Executing a standard benchmark gives meaningful insight into certain performances of the cloud being evaluated, called the *Cloud under Evaluation* (*CuE*). However, the results of these benchmarks are correlated to a specific use-case defined for a particular consumer. Highlighting principal concepts, we detail the benchmarking task to give a clear understanding of the parts: In Fig. 1, we show the principal concepts.

A benchmark is an independent part of a metric but can also be a part of one or more metrics used in different environments. A metric relates to QoS that comprises cloud characteristics, i.e., service level agreements and objectives (SLA, SLO), and workload characteristics. The SLO are the objectives fulfilling consumer requirements, while the SLA are on these objectives and serves as the official contract between consumers and providers.

Basing the QoS only on the SLA and SLO is insufficient because it only considers platform-related rules. Including characteristics, based on the nature of the workload, can lead to improved insight. From these metrics, a KPI is defined. It contains either all or some of the defined metrics based on consumer priorities. E.g., GPU speed may be important but might have an insignificant impact on the overall workload.

4.2 Operational concepts

Beyond flexibility and extensibility, we also consider stability and maintainability paramount to the Cloud Monitor. The system should be able to deploy, collect, and process benchmark results under different and variable actual conditions. Concurrently, uptime of the CuE must be insignificantly, or not at all, affected by the monitor itself.

Moreover, monitoring and metering tools included in the CuE must not affect the assessment by the Cloud Monitor in any way to guarantee its independence.

Run-time monitor deployment in the CuE should be timely and current without the hassle of implementing new monitoring infrastructure for each provider.

Scalability, the possibility to execute potentially many microbenchmarks simultaneously on multiple, and possibly, different clouds, while still collecting and processing results in real-time, is another important feature.

Finally, extensibility of the Cloud Monitor is essential to cope with future asset types in the portfolio of a specific CuE. Existing and new benchmarks should integrate with these with little effort.

4.2 Architecture

The Cloud Monitor architecture, shown in Fig. 2, has the following main components: Core, Executor, QoS evaluator, and KPI generator.

When a consumer requests to monitor a cloud, they provide workload-specific criteria via the QoS evaluator that, in turn, provides a set of QoS metrics. The KPI generator then creates or associates microbenchmarks to a consumer-specific KPI.

Benchmark jobs are submitted to the core, which is a log-based event-streaming microservice that schedules benchmarks and performs real-time analysis on the results of the CuE.

The executor is responsible for maintaining and fetching environments based on the benchmark requests from the core. Besides this, the executor is responsible for deploying benchmarks on the chosen cloud. Differentiating the executor from the core, better scalability and stability are achieved, i.e., the executor can rapidly scale to initiate run-time monitors, while the core manages the requests and data received from the KPI generator and CuE, respectively.

---

[2] https://github.com/GoogleCloudPlatform/PerfKitBenchmarker, accessed: 28/12/20
[3] https://datastudio.google.com/, accessed: 28/12/20

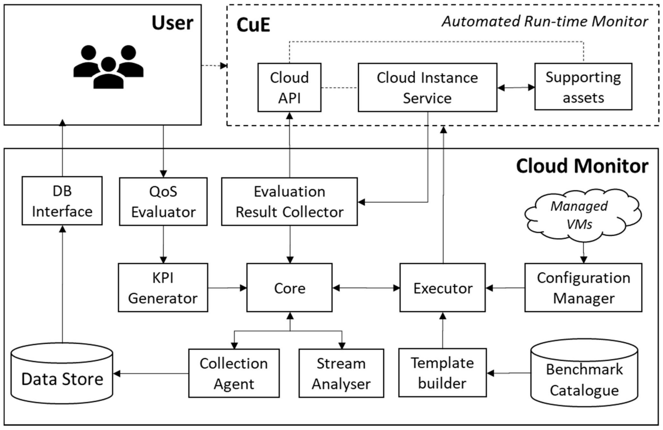

Figure 2: Cloud Monitor architecture and run-time monitor

To promote consistency of the CuE environment, VM images are prebuilt and stored. Managed VMs is a repository that contains cloud vendor-specific machine images, like an Azure image, with prerequisite software, like installed programs, region and permission level settings, but no benchmark itself. This is stored in a separate benchmark catalogue repository.

The configuration manager combines the benchmark and the prebuilt VM image, based on the KPI generator requirements.

The evaluation result collector receives performance data and sends it to the core, where real-time data formatting, analytics and temporary storage occur. The executor retrieves these results and verifies the deployment, thus promoting trustworthiness and traceability of the result. The latter gives confidence that the data was for a specific benchmark version or for a specific CuE.

4.3 Implementation Choices

The Cloud Monitor deployment is managed by *Terraform v0.13.3*. Each microservice is scaled and managed by *Kubernetes (K8S)* using *Node v.1.17.14-gke.400*. Nodes all execute in *Docker* containers with *Python 3.9.0* or *Nodejs v.12.0*.

The K8S cluster runs the *Google Kubernetes Engine* (GKE) having three *e2-standard-4* instances in the *europe-west4* region. This region was chosen because of its cost-efficient instances.

InfluxData suite is used to process and store results. It contains a readily-available open-source time-series database, called *InfluxDB,* and has a strong user community and integrates easily with powerful ecosystem tools, like *Telegraph* and *Google Data Studio*. In fact, *Telegraf v.1.16* is used as the collection agent with an *InfluxDBv.1.8* database. The core uses *Kafka v.2.7. KSQL .v.16 v* as stream analyser.

For image management we used two open-source tools: *Packer v.1.6* is retains consistency in the machine images, while machine image configurations are managed by *Ansible v.2.10*.

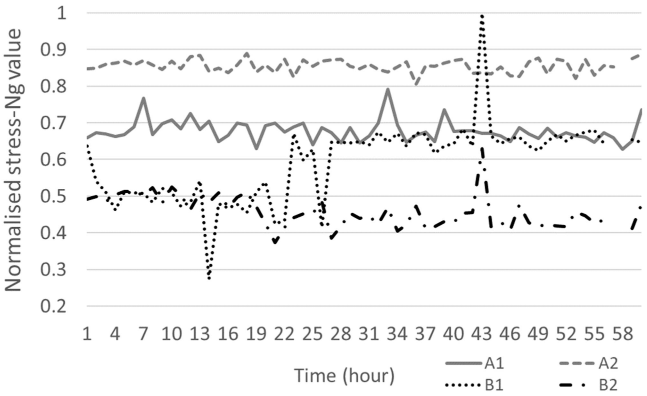

Figure 3: Normalised *stress-ng* values

To deploy and activate cloud resources, we either used the *OpenStack API v.2.1* or the cloud provider's own API.

5 DEMONSTRATOR

We monitor CPU performance using the *stress-ng* microbenchmark. We selected two seemingly comparable VM instances as potential services of each of two different cloud providers: Provider A is a more well-known provider, while Provider B is a smaller lesser-known alternative.

We selected instances such that one option is expected to be more performant than the other. The first is less performant than the second, resulting in Instances A1, A2, B1 and B2. Table 1 give some CPU specification details.

Instance	A1	A2	B1	B2
Virtual CPUs	2	2	2	2
CPU type	Intel/AMD EPYC Rome 2.25-3.4 GHz	Xeon 2.8-3.4 GHz	Xeon 2.4-3.4 GHz	Xeon 2.9-3.8 GHz
RAM (Gb)	8	8	4	4

Table 1: Instance specification details

Every hour, one performance test was submitted to the Cloud Monitor for each instance automatically. This was done for a duration of sixty hours.

stress-ng bogus operations per second results are normalized [22]. Moreover, we report the progressive change ratio: This is a simple metric reporting the absolute relative difference between successive *stress-ng* results. Calculated as the ratio of the absolute difference between two successive values with respect to the first value, we use it as a demonstrative indication of instance stability, i.e., the extent to which current measurements change with respect to the most recent one.

5.1 Real-time Monitoring Results

The normalised *stress-ng* value is shown over the reported period for all monitored instances in Fig. 3. Provider B instances appear to offer lower CPU performance than those of Provider A.

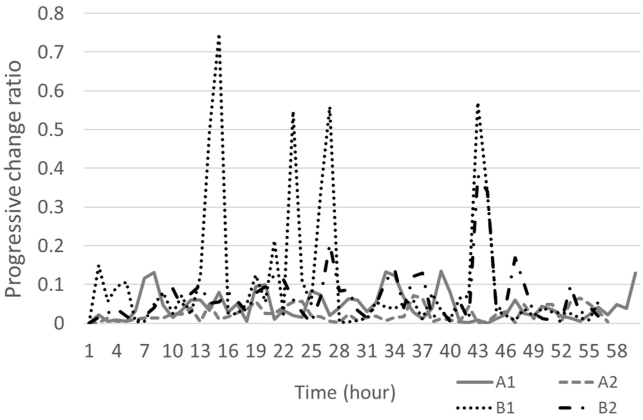

Figure 4: Progressive change ratio of *stress-ng* values

Instance B1 appears to transition to another profile between hours 22 and 28. Moreover, significant high and low spikes occur around hours 13 and 43, the latter coinciding with an increased performance spike of Instance B2.

There is a small yet noticeable gap between hours 55 and 58 for Instances A2, B1 and B2. This was due to international network interruption made public afterwards.

Fig. 4 shows the progressive change ratio for the instances. In addition to exhibiting lower performance, Provider B also seems to provide a less stable performance. Instance B1 shows particularly high subsequent changes of over 50 percent, with a peak of 75 percent around hour 15. Instance B2 seems to behave similarly to B1, especially around hour 45.

The mean versus the standard deviation of the progressive change ratio is shown in Fig. 5. It is clear that Provider B instances have a higher change ratio mean and greater standard deviation than Provider A, suggesting that the CPU performance of their services are more volatile, according to the *stress-ng* microbenchmark. Instance A2 appears to be the superior performant instance of all in this regard.

However, considering the performance reported by Fig. 3, Instance A1 appeared to go unaffected by the network interruption, while the service of all others stopped.

5.2 Discussion

The demonstrator is intentionally simplistic for the sake of clarity and shows how the monitoring results are useful to different actors in the ecosystem:

It helps the consumer decide which instance is (better-)suited for their performance needs. In our demonstrator, CPU performance, together with the availability and volatility of the network, can be observed. Brokers can also use the monitor to support the consumer.

For multi-cloud options, brokers and consumers can assess whether and when to switch providers or instances. This becomes particularly valuable when the switching of providers can occur while the application is in production, leading to real-time optimization of KPIs.

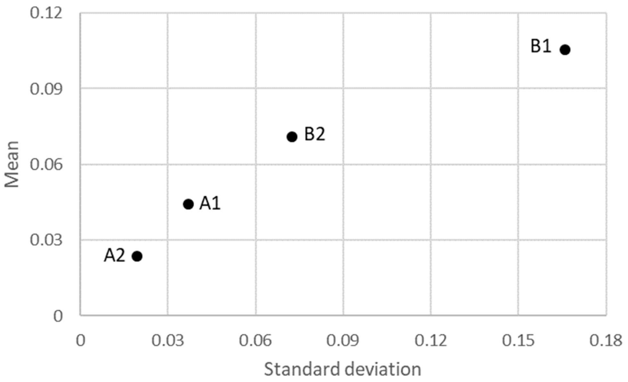

Figure 5: Mean versus standard deviation of the progressive change ratio

Providers can use the monitor to observe the performance of their own or competitor’s offerings. They identify which offerings are similar, and gain insights into their own performance. Moreover, by observing performance of known instances, current workload state and its seasonality can be observed and assessed.

Auditors, the official actors, or consumers themselves can, by way of the Cloud Monitor device, hold providers more accountable with greater transparency. As clearly observed in Fig. 4, unknown or unclear performance characteristics can be observed, as we see for Instance B1.

Taking special note of the network failure towards the end of the observation period, the discrete-time nature of the frequency of performance sampling does not exclude other outage period. This requires careful consideration of, not only which metrics are relevant and important, but the experimental setup relevant to the purpose of the use of the Cloud Monitor.

6 CONCLUSION AND FUTURE WORK

In this paper, we describe our first steps towards an independent Cloud Monitor, presenting the initial architecture and implementation. By way of a simplistic demonstrator, we show how it is a valuable tool that can help ecosystem actors, like consumers, clarify cloud performance.

Depending on the specific actors’ needs, its flexibility and extendibility should allow new or existing benchmarks to be quickly integrated. Automatic deployment on the cloud of choice, called the Cloud under Evaluation (CuE), then makes cloud monitoring more accessible, resulting in greater transparency in the ecosystem. Furthermore, providing stability and maintainability is promoted by decoupling the CuE from the Cloud Monitor.

However, although the Cloud Monitor we are developing is a powerful enabling technological solution for next generation cloud, there are still quite a few obstacles to address:

Understanding which metrics or microbenchmarks are appropriate for a specific application is not trivial. How to adapt KPIs and QoS metrics that are usually application-specific is not an easy problem to solve. The relationship between the actual application and which benchmarks accurately represent them is future work.

Running a monitor and, specifically, doing so undetected is another challenge. If a cloud can identify a monitoring activity, it could potentially provide ideal performance so as to feign good performance.

Then there is the experiment paradox: when the current workload effects the benchmarking result, the benchmarking process needs to be sufficiently lightweight or computationally intense to determine how well the cloud instance would perform under the expected application's impact.

Finally, we highlight the issue of cost. Executing a run-time monitor is not free. The benchmark application, the frequency with which it is executed, and the data storage required to execute and collect the result are all factors that determine the cost of the monitoring task.

Having a Cloud Monitor that deploys, execute and collects run-time monitors can be automated, made flexible and relatively easy to use. We see high accessibility as the overarching goal. However, just as with any tool used in a relatively complicated activity, the Cloud Monitor still requires some thought about how to apply it and how to interpret the eventual results, ultimately making the cloud more transparent.

ACKNOWLEDGEMENTS

We would like to acknowledge the *ITEA3 TESTOMATproject* and the *Dutch Ministry of Economic Affairs and Climate Policy* for supporting the groundwork. We thank Sebastiaan Tesink for their careful reading and discussion, improving manuscript quality.

REFERENCES

[1] Philipp Leitner and Jurgen Cito. 2016. Patterns in the Chaos – A Study of Performance Variation and Predictability in Public IaaS Clouds. ACM Transactions on Internet Technology 16(3).

[2] Duha Alsmadi and Victor Prybutok. 2018. Sharing storage behavior via cloud computing; Security and privacy in research and practice. Computers in Human Behavior 85. 218-226.

[3] Michael Hogan, Fang Liu, Annie Sokol and Jin Tong. 2011. NIST Cloud Computing Standards Roadmap. Cloud Computing Standards Roadmap Working Group. Special Publication 500-291. NIST.

[4] Sebastian Floerecke, Franz Lehner and Sebastian Schweikl. 2020. Cloud computing ecosystem model: evaluation and role clusters. Electronic Markets, Springer.

[5] Public Relations. 2020. GAIA-X: A Pitch Towards Europe. Status Report on User Ecosystems and Requirements. Federal ministry for Economic Affairs and Energy. Germany.

[6] Colleen Graham, Fabrizio Biscotti, Bindi Bhullar, Terilyn Palanca, Craig Roth, Neha Gupta, Julian Poulter, Alys Woodward and Jim Hare. 2020. Market Share: Enterprise Public Cloud Services, Worldwide, 2019. Gartner Research. Gartner ID G00717082.

[7] Markus Böhm, Galina Koleva, Stefanie Leimeister, Christoph Riedl and Helmut Krcmar. 2010. Towards a Generic Value Network for Cloud Computing. Economics of Grids, Clouds, Systems, and Services. Grid Economics and Business Models. LNCS 6296. Springer. 129-140.

[8] Ashraf Mohammed, Jörn Altmann and Junseok Hwang. 2009. Cloud Computing Value Chains: Understanding Businesses and Value Creation in the Cloud. In: Economic Models and Algorithms for Distributed Systems. Autonomic Systems. Basel. 187-208.

[9] David Bermbach, Erik Wittern and Stefan Tai. 2017. Cloud Service Benchmarking: Measuring Quality of Cloud Services from a Client Perspective. Springer International Publishing.

[10] Brendan Gregg. 2021. Systems Performance: Enterprise and the Cloud. 2nd Edition. Addison-Wesley. ISBN: 0-13-682015-8

[11] Christoph Laaber, Joel Scheuner and Philipp Leitner. 2019. Software microbenchmarking in the cloud. How bad is it really? In: Empirical Software Engineering 24 (4). Springer. 2469–2508.

[12] Standard Performance Evaluation Corporation. 2018. SPEC Cloud® IaaS 2018. Standard. Version 1.1. URL: https://www.spec.org/cloud_iaas2018, accessed: 11/2/2021.

[13] Cisco Systems, Inc. 2019. Cloud Performance Benchmark: 2019-2020 Edition. Research Report. Thousand Eyes.

[14] Blesson Varhese and Raijkumar Buyya. 2018. Next generation cloud computing: New trends and research directions. Future Generation Computer Systems 79 (3). 849-861.

[15] Mohammad Aslanpour, Sukhpal Gill and Adel Toosi. 2020. Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research. Internet of Things 12. Elsevier.

[16] Behnam Dezfouli, Immanuel Amirtharaj and Chia-Chi Li. 2018. EMPIOT: An Energy Measurement Platform for Wireless IoT Devices. Network and Computer Applications 121. 135-148.

[17] Sukhpal Singh, Inderveer Chana, Maninder Singh and Rajkumar Buyya. 2016. SOCCER: Self-Optimization of Energy-efficient Cloud Resources. Cluster Computing 19. 1787-1800.

[18] Devki Jha, Khaled Alwasel, Areeb Alshoshan, Xianhua Huang, Ranesh Naha, Sudheer Battula, Saurabh Garg, Deepak Puthal, Philip James, Albert Zomaya, Schahram Dustdar and Rajiv Ranjan. 2020. IoTSim-Edge: A simulation framework for modeling the behavior of Internet of Things and edge computing environments. Software. Wiley.

[19] Paul Bardea, Charlotte Dillon, Nathan VanBenschoten and Andy Woods. 2019. 2020 Cloud Report. Report. Cockroach Labs.

[20] Derek Phanekham, Matthew Zaber, Suku Nair, Steve Deitz, Rick Jones, Manasa Chalasani and Mike Truty. 2020. Measuring Cloud Network Performance with PerfKit Benchmarker. Technical White Paper. Google Cloud.

[21] Wubin Li, Yves Lemieux, Jing Gao, Zhuofeng Zhao and Yanbo Han. 2019. Service Mesh: Challenges, State of the Art, and Future Research Opportunities. In proceedings: Service-Oriented System Engineering, International Conference on. IEEE.

[22] Yadolah Dodge and Daniel Commenges. 2003. The Oxford Dictionary of Statistical Terms. Oxford University Press. ISBN 0-19-920613-9 (*entry for normalization of scores*)