# A Research-backed Extended Taxonomy for Cloud Computing Elasticity

Raouia Bouabdallah[a]
*Higher Management Institute of Tunis, Tunis University, Bou Choucha Street, Tunisia*

Keywords: Elasticity, Taxonomy, Cloud Computing, Provisioning, Monitoring.

Abstract: Elasticity is an important feature to characterize cloud computing from traditional Information Technology (IT) infrastructure. It refers to the ability of the cloud provider to provision and release cloud resources, with demand, appearing to be infinite in any quantity at any time.In this paper, we propose a taxonomy which is an extended elasticity classifications compared to existing ones. Then, we discuss industrial and academic research on cloud elasticity to identify the main issues and drawbacks. Finally, we propose a synthesis of the studied works based on elasticity solutions' characteristics provided from our taxonomy.

## 1 INTRODUCTION

Cloud computing (Fontana de Nardin et al., 2021) is a new paradigm in the IT evolution. In the literature, there is no,universal definition of the cloud computing (Chen et al., 2022). It is commonly accepted that the cloud is characterized by a certain degree of elasticity and on demand access capacity. A definition given by the National Institute of Standards and Technologies (NIST) (Mell and Grance, 2011) defines cloud Computing as a model for enabling on-demand network access to a shared pool of configurable computing resources that can be quickly put into service through the interaction with the provider. This model is composed of five essential characteristics (Karuna Pande Joshi and Yesha, 2010) which are: on-demand self-service, broad network access, resource pooling, measured service (Sambit et al., 2020) and rapid elasticity. On-demand self-service refers to the ability of a cloud provider to provision cloud resources whenever the clients need them. Broad network access describes the network used to access resources hosted in the cloud. Resource pooling describes a situation in which the resources of the provider are pooled among several clients. Measured service refers to the ability of the cloud provider to monitor, control and report resources. Elasticity is a unique feature of the cloud technology(BAR, 2020). It refers to the ability of cloud provider to provision cloud resources, with demand, appearing to be infinite in any quantity at any time (Beltran, 2016). Efficient management is a challenging task with cloud elasticity. To get a wide view of the main cloud elasticity issues and gaps, a profound survey of research efforts is required. In this paper, we will propose an extended elasticity classifications taxonomy and we will discuss industrial and academic research on cloud elasticity based on elasticity solutions' characteristics.

The structure of this paper is as follows. In Section 2, we present related studies accordance with the elasticity. Afterwards, we present, in Section 3, a synthesis of the mentioned works within a classification table. In Section 4, we provide an extended classification in accordance with the scope, purpose, policy, method, monitoring metric, etc.

## 2 OVERVIEW OF ELASTICITY SOLUTIONS

To get a broad view of the elasticity problem, a review of related work is required to identify the main issues and drawbacks. Doing so, we present in this section an overview of state-of-the-art solutions. We classify these solutions regarding the elasticity policies used for executing of monitoring actions. These policies are classified into three models which are: reactive, proactive and hybrid (e.g. proactive and reactive models).

### 2.1 Reactive Elasticity

Many of the research is interested with the reactive elasticity model. For instance, the authors in

[a] https://orcid.org/0000-0002-6386-4223



Bouabdallah, R.
A Research-backed Extended Taxonomy for Cloud Computing Elasticity.
DOI: 10.5220/0011070500003200
In *Proceedings of the 12th International Conference on Cloud Computing and Services Science (CLOSER 2022)*, pages 231-237
ISBN: 978-989-758-570-8; ISSN: 2184-5042

(P. Fawaz and Lionel, 2016) and (Paraiso et al., 2013) propose a multi-cloud platform as a service (PaaS) approach based on a reactive model for monitoring the distributed application through a *Monitoring* component which is deployed on any cloud resource (e,g. VM) hosting the application. This component is responsible for capturing any change in the state of the application as well as collect, aggregate and report information about an application deployed on multiple cloud environments. This approach adopts a reactive model based on a set of elasticity rules to increase or decrease the number of resources running the application via a load balancer used to distribute the application's workloads. However, these rules are not dedicated to specify the predication expression.

In this work (d. Alfonso et al., 2013), the authors propose a system called CLUES (Cluster Energy Saving System) which is a general power management tool based on Dynamic Power Management (DPM) approach to optimize the energy consumption of the cluster . This system uses an automated mechanism to power on or power off the resources (e.g., virtual machines) according to the current workload and integrates different resource managers by using *Resource Manager Connectors*. However, these connectors are basically used to the batch systems which provide control over batch jobs, than the cloud systems (e.g., opennebula and openstack). The cloud systems can not provide monitoring information about the deployed resources such as the number of handled jobs regarding the processing units (cores). The authors, in this work, propose also a reactive model to handle the energy consumption. This model executes an action when the measured value is over or under a given threshold defined by the client. However, the proposed model does not take into account a time interval that the scale out (down) condition should satisfy before executing an action. This may result in oscillation of the system.

Another reactive model of elasticity was proposed by (Mohamed et al., 2016) to optimize the provision of resources in the cloud. This work proposes an extension of the Open Cloud Computing Interface (OCCI) for monitoring and reconfiguring resources. To do so, It defines a list of OCCI Entities and Mixins to enable cloud resources elasticity. This elasticity was defined by adding elasticity rules based on an event-condition-action approach for monitoring the metric data obtained from cloud infrastructure. But, this work does not provide a predictive elasticity mechanism to dynamically minimize the resource consumption over time.

Most of these previous researchers ( such as (P. Fawaz and Lionel, 2016), (d. Alfonso et al., 2013) and (Mohamed et al., 2016)) apply threshold-based rules to perform the reactive model. Most of them allow clients to set rules for the provisioning of resources based on two threshold values per performance metric, which are: the upper threshold (ThrU) and the lower threshold (ThrL). However, (Koperek and Funika, 2012) is based on four threshold values including: the upper threshold (ThrU), slightly below the upper threshold (ThrbU), lower threshold (ThrL) and slightly above the lower threshold (ThroL).

RESERVOIR (*Resources and Services Virtualization without Barriers*) is a FP7 project (Rochwerger et al., 2009) that aims to provide to all clients services oriented computing without requiring a large capital investment in the infrastructure. To do so, the RESERVOIR refers to the peer-to-peer federated clouds to distribute data centers owned by separate providers. When a cloud provider does not have the needed computational resources to serve its clients, it rents these ones from another cloud provider. This project has two main actors which are: the *Service Provider* and *Infrastructure Provider* (Galán et al., 2009). The *Service Provider* represents the entity that needs to lease an IT capacity (e.g., hardware) from a cloud infrastructure provider, instead, it uses an in-house one. Doing so, it defines its requirements in a *Service Definition Manifest* (Chapman et al., 2012). This is used to specify one or more virtual machine images in a single file named OVF package which is based on the DMTF's Open Virtualization Format (OVF) standard. The *Service Definition Manifest* may also define elasticity rules using Key Performance Indicators (KPIs). The *Infrastructure Provider* is the essence of the RESERVOIR service cloud. It represents a cloud vendor that provides resources as virtual machines as a service in a pay-as-you-go financial model.

## 2.2 Proactive Elasticity

Some researchers are interested with the proactive elasticity to adjust the provisioning of cloud resources. The time series analysis approach is a proactive model in nature. This approach is used to predict a sequence of metric data measured over a time. It includes a set of forecast methods such as: the moving average, auto-regression, exponential smoothing and polynomial regression. For example, the authors in (Gong et al., 2010) use the moving average as a forecast method to automatically predict the resources required to the application. Unfortunately, this work provides poor results. Others (Roy et al., 2011) use a second order auto-regressive moving average method (ARMA) to identify the number of resources used by

the application. It is worth to note that the proactive model is not only used to identify the workload of the application as explained previously in (Gong et al., 2010) and (Roy et al., 2011), but also is used to optimize the number of VM migrations in the cloud data center. For example, Subhadra and Anil (Shaw and Singh, 2015) hope, by this migration, to minimize the number of physical machines with reduce the performance degradation comes by unneeded movement of VMs. The optimization of the number of VM migrations depends on the current as well as the future CPU utilization of physical hosts using the exponential smoothing as a forecast method.

### 2.3 Hybrid Elasticity

A few pieces of research involved both the proactive and reactive models. For instance, the authors in (Shaw and Singh, 2015) propose reactive and proactive models for resolving bottlenecks of the web application in order to satisfy the response time requirement. The reactive elasticity method is applied to resolve the bottleneck by scaling up the resources required by the web application. The proactive elasticity method is applied when the allocated resources are not required during a period of time to scale down theses resources whenever possible. This model is based on the time series analysis approach and is developed using the polynomial regression method.

In (Bauer et al., 2019), the authors propose a reactive and proactive models to scale up and down cloud resources meeting the SLA. The authors used the queering theory method as a proactive model to scale down the resources and the thresholds method as reactive model to scale them up. This work does not propose a novel reactive model method. Moreover, the proposed proactive model uses the number of requests handled per time unit as a control metric instead of the hardware metrics such as memory and CPU usage.

In (Urgaonkar et al., 2008), the authors propose propose a reactive and proactive models. The predictive method is used over long time scales( hours and days). While, the reactive model is used over short time scales ( seconds and minutes).

## 3 SYNTHESIS OF RELATED WORK

We present an overview of state-of-the-art solutions ((P. Fawaz and Lionel, 2016), (d. Alfonso et al., 2013), (Rochwerger et al., 2009), AgentRW11 , etc ) that focus on the monitoring. It is worth noting that these solutions handle partially the problem of the bottleneck situations, especially the CPU bottleneck which is due to the high traffic web application. Moreover, these situations attempt to provide monitoring for applications in cloud. But almost all the proposed situations give tooling solutions to monitor cloud applications behavior. It is worth noting that almost all of these cited works either do not target monitoring at different models which are reactive and proactive, or do not provide an efficient solution to do that. In Figure 1, we propose a synthesis of the studied works based on elasticity solutions' characteristics provided from our taxonomy that we will detail in the next section.

## 4 ELASTICITY TAXONOMY

Through the analysis of the studied solutions, we note that these solutions follow different methods, strategies and techniques so as to build their mechanisms. A review of related studies ((Coutinho et al., 2015), (Najjar et al., 2014), (BAR, 2020)) in necessary to identify the different classifications that have been suggested in accordance with the elasticity solutions' characteristics. We push further these studies ((Coutinho et al., 2015), (Najjar et al., 2014)) and we propose a taxonomy which is an extended elasticity classifications compared to existing ones. This taxonomy is summarized in Figure 2 and is based on the following characteristics: the scope, purpose, policy, method, provider, monitoring metric and standard.

### 4.1 Scope

This characteristic defines in which cloud layers the elasticity actions are applied (Al-Dhuraibi et al., 2018). The decisions made by the cloud provider or by the user of the cloud technology can be executed either in the infrastructure or the application-platform levels. We assume that these decisions are represented by the actions of the provisioning of new resources and releasing the unused ones. When these decisions are performed at the infrastructure level, the provider, which has an elasticity controller, converts the client's requirement to actions based on the virtualization technology like VMs or containers. In case the decisions of elasticity actions are performed on the application or platform level, the elasticity controller is embedded in the application which can be either one tier or multi-tiered (Herbst et al., 2013). This controller interacts with the cloud infrastructure so as to add or release resources.

## 4.2 Purpose

The elasticity solutions have different purposes which are: increase (or decrease) the infrastructure capacity, energy, performance and cost. Increase or decrease the infrastructure capacity refers to the strategy applied by the cloud provider to re-size its infrastructure capacity according to the amount of clients using this one. The energy purpose aims at optimizing the total energy consumption of IT equipments by reducing the amount of servers used to run virtual machines. The cost purpose is closely related to the energy one but also covers the operational cost of such a resource, hosted in the cloud, with maintaining the required performance. The performance refers to the ability of the cloud provider to retain Quality of Service (QoS) metrics so as to respond to the client's requirement. By getting a broader view to the elasticity objectives, we notice that there are a plenty of perspectives. The cloud IaaS providers look to maximize their profit in accordance with minimizing its infrastructure. The cloud PaaS providers search to reduce their payments. The clients (end-users) seek to minimize the cost they pay to cloud provider, while still receiving a good Quality of Service (QoS).

## 4.3 Policy

The policy refers to the required interactions for the execution of elasticity actions. Regarding related studies (Pham, 2016), there are two types of policies : the manual and automatic. The manual policy is performed by a manual intervention from the client which is responsible for provisioning and monitoring his virtual environment. The provision of a service in the cloud has five phases executed in a sequential order (Karuna Pande Joshi and Yesha, 2010), which are: the requirements, discovery, negotiation, composition and consumption. In the requirement phase, the client details his requirements. Then, the provider searches the cloud services corresponding to the client's need in the discovery phase. Afterwards, the negotiation phase occurs when there is conflicts between the cloud provider and client about a service. In the composition phase, the provider combines the services required by the client to provide a single service. Finally, this service is delivered to the client in the consumer phase. It is worth to note that the manual policy is used in several cloud providers such as Rackspace and Microsoft Azure. However with the automatic policy, all the elasticity actions are performed automatically, which could be classified into the reactive and proactive models.The reactive model

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Standard** | OVF | | | | | | | ↘ |
| | OCCI | | | | ↘ | | | |
| **Monitoring Metric** | OS | | ↘ | | ↘ | | | ↘ |
| | Application | | ↘ | ↘ | ↘ | ↘ | ↘ | |
| **Provider** | Federated | Peer-to-peer | | | | | | ↘ |
| | | Centralized | ↘ | | | | | |
| | Single | | | ↘ | ↘ | ↘ | ↘ | |
| **Method** | Fine-grained | Vertical | | | | | | |
| | | Horizontal | | ↘ | ↘ | ↘ | ↘ | ↘ |
| | Coarse-grained | | | | | | | ↘ |
| **Policy** | Automatic | Reactive | ↘ | ↘ | ↘ | ↘ | ↘ | ↘ |
| | | Proactive | ↘ | | | ↘ | ↘ | |
| | Manual provisioning | consumption | | | | | | |
| | | Composition | ↘ | | | | | |
| | | Negotiation | | | | | | |
| | | Discovery | | | | | | |
| | | Requirements | ↘ | | | | | |
| **Purpose** | Performance | | | | ↘ | | | |
| | Energy | | | ↘ | | | | |
| | capacity | | | | | | | |
| | Dec/Infrastructure | | ↘ | | | ↘ | ↘ | ↘ |
| **Scope** | Applicaton | | ↘ | | | | | |
| | Infrastructure | | | ↘ | ↘ | ↘ | ↘ | ↘ |
| **Studied Solutions** | | | SoCloud (P. Fawaz and Lionel, 2016) (Paraiso et al., 2013) | CLUES (d. Alfonso et al., 2013) | (Mohamed et al., 2016) | (Shaw and Singh, 2015) | (Bauer et al., 2019) | RESERVOIR (Rochwerger et al., 2009) |

Figure 1: Synthesis of related work.

reacts based on thresholds or rules, due to the observed workload changes. Several industrial cloud providers use this model such as: Amazon EC2 and Rightscale, as well as academic researches such as: (Han et al., 2014) and (Liu et al., 2013). Whereas, the proactive model reacts to the predicted workload changes. Most of the reviewed works ( such as (Dawoud et al., 2011), (Liu et al., 2013) and (Sharma et al., 2011)) that used this model could fit in one of these three groups of technique: the time series analysis, queuing theory and control theory.

### 4.4 Method

It is related to the applied methods to increase (or decrease) the size of cloud resource. With regard to the underlining studies (Pham, 2016), there are two kinds of methods : the coarse-grained or fine-grained. The coarse-grained scaling refers to the ability of the cloud provider to provision resources from an external one. By the way, the fine-grained refers to the ability of the cloud provider to scale resources in the frontier of its infrastructure capacity. Generally within the same provider, the resources can be scaled in two different manners: vertical and horizontal. The vertical elasticity consists in increasing or decreasing characteristics of resources (e.g virtual machine instances) such as the CPU, memory, etc. However, the horizontal elasticity refers to the ability to increase or decrease the number of allocated resources (e.g virtual machine instances) needed to run an application in the cloud provider.

### 4.5 Provider

It refers to the number of cloud providers that the elastic solutions support simultaneously, which could be a single cloud or a federation of clouds. A single provider means that the elasticity control is applied on only one cloud provider. While in case of federated clouds, the elasticity control is executed across a set of cloud providers. The examination of review works show us that there are two cloud federation models: vertical federation and horizontal federation (Celesti et al., 2010). The vertical federation reaches all levels of cloud service models such as IaaS, PaaS and SaaS. For instance, a SaaS cloud provider puts his services on the top of a PaaS cloud provider like Microsoft Azure Service Platform and Google App Engine. However, the horizontal federation of clouds deals with one cloud service model level such as IaaS.

### 4.6 Monitoring Metric

The monitoring metrics describe the types of values the elasticity control use to monitor the behavior of the underling cloud resources. The monitoring metrics are available at these levels: the operating system and application metrics. The operating system metrics are applied in evaluating the performance of a virtual hardware with overlooking the applications running on it. These metrics are: the CPU, RAM, Disk space and network. However, the application metrics refer to the units of work that involve on the virtual hardware as well as applications. These metrics can be the response time and the number of requests.

### 4.7 Standard

It refers the standards that the elastic solutions support, which can be the OCCI [1] and OVF [2]. The Open Cloud Computing Interface (OCCI) is one of the first open extensible standards for managing any kind of resources provided by cloud providers. This standard is supported by a large community of cloud providers including commercial ones such as Amazon EC2, as well as cloud platforms such as CloudStack, Eucalyptus, OpenNebula. While, the Open Virtualization Format (OVF) (DMTF, 2015) specification offers a portable and an extensible format for packaging and distributing cloud resources (e.g virtual machines) in a standard format.

## 5 CONCLUSIONS

In this paper, we discussed industrial and academic researches on the elasticity in the cloud computing. Then, we presented a synthesis of the mentioned works within a classification table. Afterword, we proposed an extended elasticity classifications taxonomy which is an extended elasticity classifications compared to existing ones. This taxonomy is based on the following characteristics: the scope, purpose, policy, method, provider, monitoring metric and standard.

In our future work, we will limit our research to the coarse grained elasticity and we will propose an extension for the Open Cloud Computing Interface (OCCI) to support the automatic negotiation between different cloud providers.

[1] OCCI: Open Cloud Computing Interface
[2] OVF: Open Virtualization Format

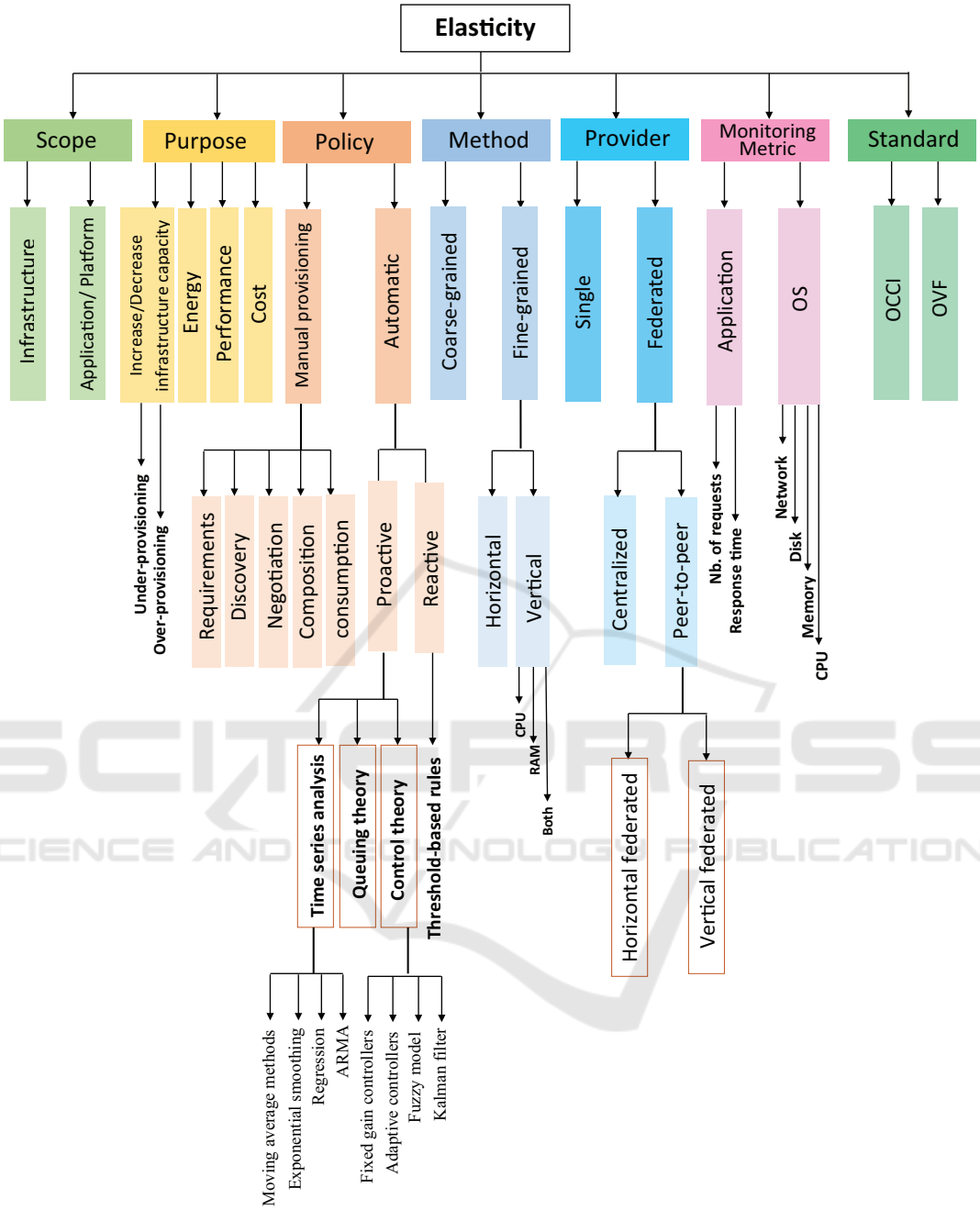


Figure 2: Classification of elasticity solutions.



## ACKNOWLEDGEMENTS

This research was enabled in part by support provided by SSOIE-COSMOS laboratory from university of Tunis (Tunisia). We would also like to thank Dso Services company by providing us the required resources to release this work.

## REFERENCES

(2020). The views, measurements and challenges of elasticity in the cloud: A review. *Computer Communications*, 154:111–117.

Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., and Merle, P. (2018). Elasticity in cloud computing: State of the art and research challenges. 11(2):430–447.

Bauer, A., Herbst, N., Spinner, S., Ali-Eldin, A., and Kounev, S. (2019). Chameleon: A hybrid, proactive auto-scaling mechanism on a level-playing field. volume 30, pages 800–813.

Beltran, M. (2016). Becloud: A new approach to analyse elasticity enablers of cloud services. *Future Generation Computer Systems*, 64:39–49.

Celesti, A., Tusa, F., Villari, M., and Puliafito, A. (2010). How to enhance cloud architectures to enable cross-federation. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 337–345.

Chapman, C., Emmerich, W., Márquez, F. G., Clayman, S., and Galis, A. (2012). Software architecture definition for on-demand cloud provisioning. In *Cluster Computing*, volume 15, pages 79–100.

Chen, Y., Huo, J., Li, X., Bi, K., Ma, N., Jing, Y., and Ma, X. (2022). Classification and characteristic analysis of the clouds and dust in a dust-carrying precipitation process based on multi-source remote sensing observations. *Atmospheric Pollution Research*, 13(1):101267.

Coutinho, E. F., de Carvalho-Sousa, F. R., Rego, P. A. L., Gomes, D. G., and de Souza, J. N. (2015). Elasticity in cloud computing: a survey. In *Annals of Telecommunications*, page 1–21.

d. Alfonso, C., Caballer, M., Alvarruiz, F., and Hernandez, V. (2013). An energy management system for cluster infrastructures. In *Computers & Electrical Engineering*, volume 39, pages 2579 – 2590.

Dawoud, W., Takouna, I., and Meinel, C. (2011). Elastic vm for cloud resources provisioning optimizatio. In *In Advances in Computing and Communications. Springer*, page 431–445.

DMTF (2015). Open virtualization format specification.

Fontana de Nardin, I., da Rosa Righi, R., Lima Lopes, T. R., André da Costa, C., Yeom, H. Y., and Köstler, H. (2021). On revisiting energy and performance in microservices applications: A cloud elasticity-driven approach. *Parallel Computing*, 108:102858.

Galán, F., Sampaio, A., Rodero-Merino, L., Gil, V., and Vaquero, L. M. (2009). Service specification in cloud environments based on extensions to open standards. In *Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE*, COMSWARE '09, pages 19:1–19:12, New York, NY, USA. ACM.

Gong, Z., Gu, X., and Wilkes, J. (2010). Press: Predictive elastic resource scaling for cloud systems. In *2010 International Conference on Network and Service Management*, pages 9–16.

Han, R., Ghanem, M. M., Guo, L., Guo, Y., and Osmond, M. (2014). Enabling cost-aware and adaptive elasticity of multi-tier cloud applications. In *Future Generation Computer Systems*, volume 32, page 82–98.

Herbst, N., Kounev, S., and Reussner, R. (2013). Elasticity in cloud computing: What it is, and what it is not. pages 23–27.

Karuna Pande Joshi, T. F. and Yesha, Y. (2010). Integrated lifecycle of it services in a cloud environment (detailed paper). In *In proceedings of The Third International Conference on the Virtual Computing Initiative (ICVCI 2009)*, Research Triangle Park, NC. IBM.

Koperek, P. and Funika, W. (2012). Dynamic business metrics-driven resource provisioning in cloud environments. In *Parallel Processing and Applied Mathematics*, pages 171–180, Berlin, Heidelberg. Springer Berlin Heidelberg.

Liu, Z., Wang, S., Sun, Q., Zou, H., and Yang, F. (2013). Cost-aware cloud service request scheduling for saas providers. In *The Computer Journal*, page 291–301.

Mell, P. and Grance, T. (2011). *The NIST Definition of Cloud Computing*. NIST, national institute of standards and technology special publication 800-145 edition.

Mohamed, M., Belaid, D., and Tata, S. (2016). Extending occi for autonomic management in the cloud. In *Journal of Systems and Software*, volume 122, pages 416 – 429.

Najjar, A., Serpaggi, X., Gravier, C., and Boissier, O. (2014). Survey of elasticity management solutions in cloud computing. In *in Continued Rise of the Cloud*, page 235–263.

P. Fawaz, M. P. and Lionel, S. (2016). socloud: A service-oriented component-based paas for managing portability, provisioning, elasticity, and high availability across multiple clouds. In *Computing*, volume 98, pages 539–565.

Paraiso, F., Merle, P., and Seinturier, L. (2013). Managing elasticity across multiple cloud providers. In *Proceedings of the 2013 International Workshop on Multi-cloud Applications and Federated Clouds*, pages 53–60, New York, NY, USA. ACM.

Pham, M. L. (2016). Roboconf : an Autonomic Platform Supporting Multi-level Fine-grained Elasticity of Complex Applications on the Cloud.

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W., and Galan, F. (2009). The reservoir model and architecture for open federated cloud computing. In *IBM Journal of Research and Development*, volume 53(4), page 4–1.

Roy, N., Dubey, A., and Gokhale, A. (2011). Efficient autoscaling in the cloud using predictive models for workload forecasting. In *2011 IEEE 4th International Conference on Cloud Computing*, pages 500–507.

Sambit, K.-M., Bibhudatta, S.-P., and Paramita, P. (2020). Load balancing in cloud computing: A big picture. *Journal of King Saud University - Computer and Information Sciences*, 32(149-158). https://doi.org/10.1016/j.procs.2015.03.168.

Sharma, U., Shenoy, P., Sahu, S., and Shaikh, A. (2011). A cost-aware elasticity provisioning system for the cloud. In *in 2011 31st International Conference on Distributed Computing Systems (ICDCS)*, page 559–570.

Shaw, S. B. and Singh, A. K. (2015). Use of proactive and reactive hotspot detection technique to reduce the number of virtual machine migration and energy consumption in cloud data center. In *Computers & Electrical Engineering*, volume 47, pages 241 – 254.

Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P., and Wood, T. (2008). Agile dynamic provisioning of multi-tier internet applications. In *ACM Trans. Auton. Adapt. Syst.*, volume 3, pages 1:1–1:39, New York, NY, USA. ACM.