Elasticity in Cloud Computing: What It Is, and What It Is Not

Nikolas Roman Herbst, Samuel Kounev, Ralf Reussner
Institute for Program Structures and Data Organisation
Karlsruhe Institute of Technology
Karlsruhe, Germany
{herbst, kounev, reussner}@kit.edu

Abstract

Originating from the field of physics and economics, the term elasticity is nowadays heavily used in the context of cloud computing. In this context, elasticity is commonly understood as the ability of a system to automatically provision and deprovision computing resources on demand as workloads change. However, elasticity still lacks a precise definition as well as representative metrics coupled with a benchmarking methodology to enable comparability of systems. Existing definitions of elasticity are largely inconsistent and unspecific, which leads to confusion in the use of the term and its differentiation from related terms such as scalability and efficiency; the proposed measurement methodologies do not provide means to quantify elasticity without mixing it with efficiency or scalability aspects. In this short paper, we propose a precise definition of elasticity and analyze its core properties and requirements explicitly distinguishing from related terms such as scalability and efficiency. Furthermore, we present a set of appropriate elasticity metrics and sketch a new elasticity tailored benchmarking methodology addressing the special requirements on workload design and calibration.

1 Introduction

Elasticity has originally been defined in physics as a material property capturing the capability of returning to its original state after a deformation. In economical theory, informally, elasticity denotes the sensitivity of a dependent variable to changes in one or more other variables [1]. In both cases, elasticity is an intuitive concept and can be precisely described using mathematical formulas.

The concept of elasticity has been transferred to the context of cloud computing and is commonly considered as one of the central attributes of the cloud paradigm [10]. For marketing purposes, the term elasticity is heavily used in cloud providers' advertisements and even in the naming of specific products or services. Even though tremendous efforts are invested to enable cloud systems to behave in an elastic manner, no common and precise understanding of this term in the context of cloud computing has been established so far, and no ways have been proposed to quantify and compare elastic behavior. To underline this observation, we cite five definitions of elasticity demonstrating the inconsistent use and understanding of the term:

1. *ODCA, Compute Infrastructure-as-a-Service* [9] "[...] defines elasticity as the configurability and expandability of the solution [...] Centrally, it is the ability to scale up and scale down capacity based on subscriber workload."
2. *NIST Definition of Cloud Computing* [8] "Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time."
3. *IBM, Thoughts on Cloud, Edwin Schouten, 2012* [11] "Elasticity is basically a 'rename' of scalability [...]" and "removes any manual labor needed to increase or reduce capacity."
4. *Rich Wolski, CTO, Eucalyptus, 2011* [12] "Elasticity measures the ability of the cloud to map a single user request to different resources."
5. *Reuven Cohen, 2009* [2] Elasticity is "the quantifiable ability to manage, measure, predict and adapt responsiveness of an application based on real time demands placed on an infrastructure using a combination of local and remote computing resources."

Definitions (1), (2), and (3) in common describe elasticity as the scaling of system resources to increase or decrease capacity, whereby definitions (1), (2) and (5) specifically state that the amount of provisioned re-

sources is somehow connected to the recent demand or workload. In these two points there appears to be some consent. Definitions (4) and (5) try to capture elasticity in a generic way as a 'quantifiable' system ability to handle requests using different resources. Both of these definitions, however, neither give concrete details on the core aspects of elasticity, nor provide any hints on how elasticity can be measured. Definition (3) assumes that no manual work at all is needed, whereas in the NIST definition (2), the processes enabling elasticity do not need to be fully automatic. In addition, the NIST definition adds the adjective 'rapid' to elasticity and draws the idealistic picture of 'perfect' elasticity where endless resources are available with an appropriate provisioning at any point in time, in a way that the end-user does not experience any performance variability.

We argue that existing definitions of elasticity fail to capture the core aspects of this term in a clear and unambiguous manner and are even contradictory in some parts. To address this issue, in this short paper, we propose a new refined definition of elasticity considering in detail its core aspects and the prerequisites of elastic system behavior (Section 2). Thereby, we clearly differentiate elasticity from its related terms scalability and efficiency. In Section 4, we present metrics that are able to capture elasticity, followed by Section 5, in which we outline a benchmarking methodology for quantifying elasticity discussing the issues of representativeness, reproducibility and fairness of the measurement approach.

2 Elasticity

In this section, we first describe some important prerequisites in order to be able to speak of elasticity, present a new refined and comprehensive definition, and then analyse its core aspects and dimensions. Finally, we differentiate between elasticity and its related terms scalability and efficiency.

2.1 Prerequisites

The scalability of a system including all hardware, virtualization, and software layers within its boundaries is a prerequisite in order to be able to speak of elasticity. Scalability is the ability of a system to sustain increasing workloads with adequate performance provided that hardware resources are added. Scalability in the context of distributed systems has been defined in [6], as well as more recently in [3, 4], where also a measurement methodology is proposed.

Given that elasticity is related to the ability of a system to adapt to changes in workloads and resource demands, the existence of at least one specific adaptation process is assumed. The latter is normally automated, however, in a broader sense, it could also contain manual steps. Without a defined adaptation process, a scalable system cannot behave in an elastic manner, as scalability on its own does not include temporal aspects.

When evaluating elasticity, the following points need to be checked beforehand:

- *Autonomic Scaling:*
  What adaptation process is used for autonomic scaling?
- *Elasticity Dimensions:*
  What is the set of resource types scaled as part of the adaptation process?
- *Resource Scaling Units:*
  For each resource type, in what unit is the amount of allocated resources varied?
- *Scalability Bounds:*
  For each resource type, what is the upper bound on the amount of resources that can be allocated?

2.2 Definition

Elasticity is the degree to which a system is able to adapt to workload changes by provisioning and deprovisioning resources in an autonomic manner, such that at each point in time the available resources *match* the current demand as closely as possible.

2.3 Dimensions and Core Aspects

Any given adaptation process is defined in the context of at least one or possibly multiple types of resources that can be scaled up or down as part of the adaptation. Each resource type can be seen as a separate dimension of the adaptation process with its own elasticity properties. If a resource type is a container of other resources types, like in the case of a virtual machine having assigned CPU cores and RAM, elasticity can be considered at multiple levels. Normally, resources of a given resource type can only be provisioned in discrete units like CPU cores, virtual machines (VMs), or physical nodes. For each dimension of the adaptation process with respect to a specific resource type, elasticity captures the following core aspects of the adaptation:

Speed The speed of scaling up is defined as the time it takes to switch from an underprovisioned state to an optimal or overprovisioned state. The speed of scaling down is defined as the time it takes to switch from an overprovisioned state to an optimal or underprovisioned state. The speed of scaling up/down does not correspond directly to the technical resource provisioning/deprovisioning time.

Precision The precision of scaling is defined as the absolute deviation of the current amount of allocated resources from the actual resource demand.

As discussed above, elasticity is always considered with respect to one or more resource types. Thus, a direct comparison between two systems in terms of elasticity is only possible if the same resource types (measured in identical units) are scaled.

To evaluate the actual observable elasticity in a given scenario, as a first step, one must define the criterion based on which the amount of provisioned resources is considered to *match* the actual current demand needed to satisfy the system's given performance requirements. Based on such a matching criterion, specific metrics that quantify the above mentioned core aspects, as discussed in more detail in Section 4, can be defined to quantify the practically achieved elasticity in comparison to the hypothetical *optimal elasticity*. The latter corresponds to the hypothetical case where the system is scalable with respect to all considered elasticity dimensions without any upper bounds on the amount of resources that can be provisioned and where resources are provisioned and deprovisioned immediately as they are needed exactly matching the actual demand at any point in time. *Optimal elasticity*, as defined here, would only be limited by the resource scaling units.

2.4 Differentiation

In this section, we highlight the conceptual differences between elasticity and the related terms scalability and efficiency.

Scalability is a prerequisite for elasticity, but it does not consider temporal aspects of how fast, how often, and at what granularity scaling actions can be performed. Scalability is the ability of the system to sustain increasing workloads by making use of additional resources, and therefore, in contrast to elasticity, it is not directly related to how well the actual resource demands are matched by the provisioned resources at any point in time.

Efficiency expresses the amount of resources consumed for processing a given amount of work. In contrast to elasticity, efficiency is not limited to resource types that are scaled as part of the system's adaptation mechanisms. Normally, better elasticity results in higher efficiency. The other way round, this implication is not given, as efficiency can be influenced by other factors independent of the system's elasticity mechanisms (e.g., different implementations of the same operation).

3 Derivation of the Matching Function

To capture the criterion based on which the amount of provisioned resources is considered to match the actual current demand, we define a matching function $m(w) = r$ as a system specific function that returns the minimal amount of resources r for a given resource type needed to satisfy the system's performance requirements at a specified workload intensity. The workload intensity w can be specified either as the number of workload units (e.g., user requests) present at the system at the same time (concurrency level), or as the number of workload units that arrive per unit of time (arrival rate). A matching function is needed for both directions of scaling (up/down), as it cannot be assumed that the optimal resource allocation level when transitioning from an underprovisioned state (upwards) are the same as when transitioning from an overprovisioned state (downwards).

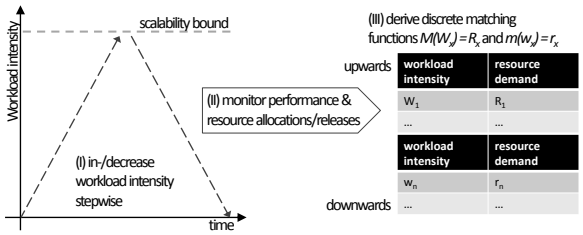

Figure 1: Illustration of a Measurement-based Derivation of Matching Functions

The matching functions can be derived based on measurements, as illustrated in Figure 1, by increasing the workload intensity w stepwise, while measuring the resource consumption r, and tracking resource allocation changes. The process is then repeated for decreasing w. After each change in the workload intensity, the system should be given enough time to adapt its resource allocations reaching a stable state for the respective workload intensity. As a rule of thumb, at least two times the technical resource provisioning time is recommended to use as a minimum. As a result of this step, a system specific table is derived that maps workload intensity levels to resource demands, and the other way round, for both scaling directions within the scaling bounds.

4 Elasticity Metrics

To capture the core elasticity aspects *speed* and *precision*, we propose the following definitions and metrics as illustrated in Figure 2:

- $\overline{A}$ is the average time to switch from an underprovisioned state to an optimal or overprovisioned state and corresponds to the average *speed* of scaling up.

- $\sum A$ is the accumulated time in underprovisioned state.
- $\overline{U}$ is the average amount of underprovisioned resources during an underprovisioned period.
- $\sum U$ is the accumulated amount of underprovisioned resources.
- $\overline{B}$, $\sum B$, $\overline{O}$, and $\sum O$ are defined similarly for overprovisioned states.

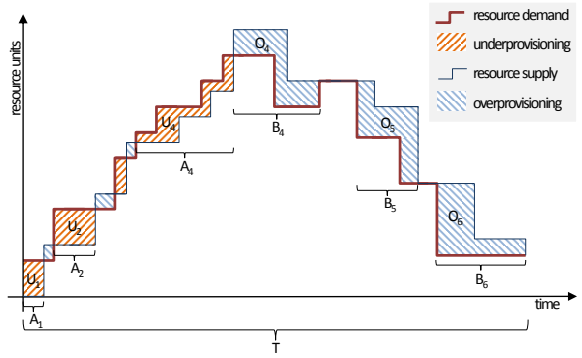

Figure 2: Capturing Core Elasticity Metrics

We define the average *precision* of scaling up P_u as $P_u = \frac{\sum U}{T}$ where T is the total duration of the evaluation period, and accordingly $P_d = \frac{\sum O}{T}$ is defined as the average precision of scaling down. Based on the above defined quantities, one could define an *elasticity* metric for scaling up E_u as inversely proportional to $\overline{A}$ and $\overline{U}$, e.g. $E_u = \frac{1}{\overline{A} \times \overline{U}}$, and accordingly *elasticity* for scaling down $E_d = \frac{1}{\overline{B} \times \overline{O}}$. The elasticity of a system under test (SUT) s can then be captured in a matrix M_s where each vector v_d represents an elasticity dimension d and contains the values of the elasticity core metrics E_u, $\overline{A}$, P_u for scaling up and E_d, $\overline{B}$, P_d for scaling down.

As an alternative to these metrics, the dynamic time warping (DTW) distance [7] can be used as a robust distance metric to capture the similarity between the demand and supply curves as well as to approximate the technical reaction time of the adaptation mechanism. A case study demonstrating this approach can be found in [5].

5 Towards Benchmarking Elasticity

Characterizing the elasticity of a single system is not a simple task on its own and it becomes even more complicated when comparing different systems. An elasticity benchmark is expected to deliver reproducible results and generate a consistent order of the different systems under test (SUTs) reflecting their potential and observed elasticity, while not mixing this with general system efficiency and scalability aspects. Traditional benchmarking approaches induce identical workloads on different SUTs to provide a basis for fair comparisons, whereas an elasticity benchmark is required to induce identical demand curves. If two elastic systems exhibit significant differences in efficiency (the amount of resources required for meeting performance requirements at a given workload intensity level), it might well be that when processing an identical workload, their adaptation mechanisms are exercised in a significantly different manner. As illustrated in Figure 3, in that case, deriving the elasticity metrics for the same workload would result in unfair comparison since the more efficient system would appear to exhibit better elasticity given that its adaptation mechanisms were not stressed to the same extent.

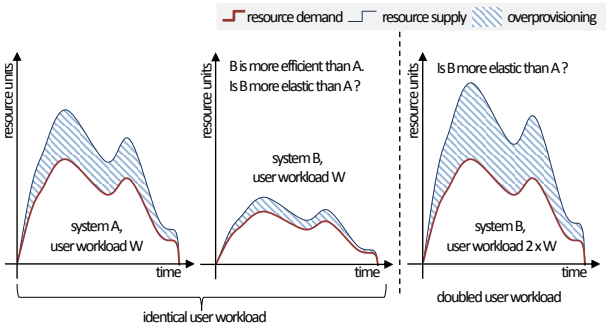

Figure 3: Elasticity vs. Efficiency

Therefore, the first step towards portability of an elasticity benchmark and comparability of its results would be the specification of a representative set of demand curves and common performance goals in terms of responsiveness, throughput or utilisation for the considered resource types. The demand curves themselves should contain bursts of different intensity, upward and downward scaling trends and seasonal patterns of different shapes, concerning amplitude, duration and base level capturing the most representative real-life scenarios. Further challenges include the automated derivation of the mapping functions as well as the generation of a workload that induces the targeted demand curves as accurately as possible on the evaluated SUTs.

6 Conclusion

In this short paper, we proposed a refined definition of elasticity to contribute in establishing a common understanding of this term in the context of cloud computing. Furthermore, we examined the core aspects of elasticity explicitly differentiating it conceptually from the classical notions of scalability and efficiency. Finally, we propose metrics to capture the core elasticity aspects as well as an elasticity benchmarking approach focusing on the special requirements on workload design and its implementation.

References

[1] CHIANG, A. C., AND WAINWRIGHT, K. *Fundamental methods of mathematical economics*, 4. ed., internat. ed., [repr.] ed. McGraw-Hill [u.a.], Boston, Mass. [u.a.], 2009.

[2] COHEN, R. Defining Elastic Computing, September 2009. http://www.elasticvapor.com/2009/09/defining-elastic-computing.html, last consulted Feb. 2013.

[3] DUBOC, L. *A Framework for the Characterization and Analysis of Software Systems Scalability*. PhD thesis, Department of Computer Science, University College London, 2009. http://discovery.ucl.ac.uk/19413/1/19413.pdf.

[4] DUBOC, L., ROSENBLUM, D., AND WICKS, T. A Framework for Characterization and Analysis of Software System Scalability. In *Proceedings of the 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering (ESEC-FSE '07)* (2007), ACM, pp. 375–384.

[5] HERBST, N. R. Quantifying the Impact of Configuration Space for Elasticity Benchmarking. Study thesis, Faculty of Computer Science, Karlsruhe Institute of Technology (KIT), Germany, 2011. http://sdqweb.ipd.kit.edu/publications/pdfs/Herbst2011a.pdf.

[6] JOGALEKAR, P., AND WOODSIDE, M. Evaluating the scalability of distributed systems. *IEEE Transactions on Parallel and Distributed Systems 11* (2000), 589–603.

[7] KEOGH, E., AND RATANAMAHATANA, C. A. Exact indexing of dynamic time warping. *Knowl. Inf. Syst. 7*, 3 (Mar. 2005), 358–386.

[8] MELL, P., AND GRANCE, T. The NIST Definition of Cloud Computing. Tech. rep., U.S. National Institute of Standards and Technology (NIST), 2011. Special Publication 800-145, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[9] OCDA. Master Usage Model: Compute Infratructure as a Service. Tech. rep., Open Data Center Alliance (OCDA), 2012. http://www.opendatacenteralliance.org/docs/ODCA_Compute_IaaS_MasterUM_v1.0_Nov2012.pdf.

[10] PLUMMER, D. C., SMITH, D. M., BITTMAN, T. J., CEARLEY, D. W., CAPPUCCIO, D. J., SCOTT, D., KUMAR, R., AND ROBERTSON, B. Study: Five Refining Attributes of Public and Private Cloud Computing. Tech. rep., Gartner, 2009. http://www.gartner.com/DisplayDocument?doc_cd=167182, last consulted Feb. 2013.

[11] SCHOUTEN, E. Rapid Elasticity and the Cloud, September 2012. http://thoughtsoncloud.com/index.php/2012/09/rapid-elasticity-and-the-cloud/, last consulted Feb. 2013.

[12] WOLSKI, R. Cloud Computing and Open Source: Watching Hype meet Reality, May 2011. http://www.ics.uci.edu/~ccgrid11/files/ccgrid-11_Rich_Wolsky.pdf, last consulted Feb. 2013.