Elasticity and Scalability Centric Quality Model for the Cloud

ALFATH Abdeladim
ENSIAS, Mohamed V University
abdeladim.alfath@um5s.net.ma

Salah BAINA
ENSIAS, Mohamed V University
salah.baina@gmail.com

Karim BAINA
ENSIAS, Mohamed V University
karim.baina@gmail.com

***Abstract*—Cloud computing seems to be the most logical shift in terms of Information Technology after Internet, Social Networking... Despite the potential benefits that cloud computing offers, the model brings new issues, challenges, and needs in term of SLA formalization, Quality of Service (QoS) evaluation due to the heterogeneous resources and to the special features it implies, such as Elasticity and Scalability. In the scope of this paper we focus on the Elasticity and Scalability attributes to assess their impact on the QoS. The paper provides a multi-lenses overview that can help both cloud consumers and potential business application's owners to understand, analyze, and evaluate important aspects related to Scalability and Elasticity capabilities. We determine and analyze the key features of these characteristics and derive metrics that evaluate the cloud elasticity-centric capabilities. We present a specific quality model for those two characteristics derived from their sub-attributes.**

Keywords—Cloud Computing, Quality of Service, Elasticity, Scalability, Quality Metrics

I. INTRODUCTION

Designing applications for large communities is more challenging than for internal small group of users [5]. Thus, the applications hosted in the cloud have to handle dynamically the workload changes in order to deliver the service with a constant level of quality. On the one hand, the final users should see the same level of performance even when the number of requests increases so that the under- provisioning is avoided. On the other hand, the allocation of the resources should be optimized so that over-provisioning could be avoided. Elasticity and Scalability are the most suitable features of the cloud computing that enable to match these requirements of the cloud and then meet the economic goals of the cloud paradigm [9].

Otherwise, the formalization of QoS expected when it comes to Elasticity and Scalability is not "mature" yet. The definition of new metrics may result in better QoS formalization in the SLA. In the scope of this paper, we pinpoint some sub- attributes of the elasticity and scalability and then derive the metrics that impact the quality the two mentioned characteristics. These metrics will help the providers of the cloud services to maintain the quality of their service and help the final clients to express their needs on the SLA in terms of QoS.

The difference between elasticity and scalability should be highlighted at this stage. Scalability can be seen as the ability of a system to handle a larger workload requirement by providing a proportional amount of resources [4]. This is part of what elasticity needs, but perfect scalability does not ensure perfect elasticity. We underline the fact that elasticity depends on the speed of response to changed workload and then we can confirm that time plays a major role when it comes to elasticity [4]. By contrast, scalability allows the system as long as it needs to meet the changed load. Also, good elasticity requires that the system reduce costs when workload decreases, while scalability just considers growth [13].

Elasticity completes scalability and adds the temporal aspect to it. Scalability attributes are common with elasticity. Under-provisioning and overprovisioning for instance reflects the ability of the cloud service to allocate the required resources to fulfill a request which is relevant for both Scalability and Elasticity. The speed of the resources to be allocated / released is more significant when we talk about Elasticity. The figure bellow sums up these differences in terms of sub-attributes of elasticity and scalability.

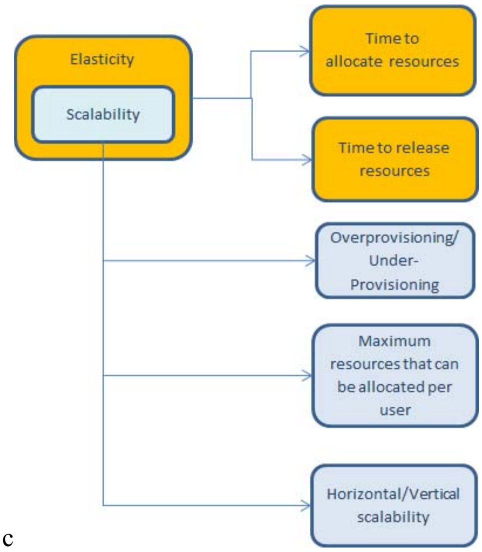
Elasticity
Scalability
Time to allocate resources
Time to release resources
Overprovisioning/ Under-Provisioning
Maximum resources that can be allocated per user
Horizontal/Vertical scalability
c

Fig. 1. Attributes of the Elasticity and Scalability

The metrics presented later in this document are based on these attributes. As shown in the figure, and mentioned earlier, Elasticity covers the Scalability since it takes into consideration the time aspect besides all the other attributes. For instance, we found that the time for allocating resources and the time for releasing resources have direct impact on the quality of the elasticity.

978-1-4799-5979-2/14/$31.00 ©2014 IEEE

Our paper is structured as the following: first, we will focus on the similar works that have been presented in term of quality of the Cloud. Then, we will highlight the concepts of the Elasticity and Scalability. In the next section, we will present four metrics that reflects and describe the quality of the Elasticity/Scalability of a given cloud service. The last sections will be dedicated to the scoring of a cloud service and to the model assessment.

II. SIMILAR WORK:

Jae Yoo Lee, and his team developed a quality model for evaluating Software as a service. This model sets up some metrics to evaluate the quality of software as a service in the cloud. Their approach consisted of defining key features of SaaS and then, deriving quality attributes from the key features. They finally defined metrics for the quality attributes [1].

A group of researchers from Melbourn University proposed a Measurement Index Cloud framework—SMICloud—which enables final users to select a Cloud service. The SMICloud framework uses QoS final user requirements and previous user experiences and performance of services to offer features such as Cloud service selection and ranking of Cloud services. This decision making tool aims at evaluating Cloud services in terms of KPIs and user requirements. Numerous metrics are established to evaluate several quantifiable aspects of the cloud such as performance, availability... The framework doesn't evaluate qualitative aspects such as security and elasticity [8].

A Group of researchers from Australia have presented an approach to measuring imperfections in elasticity for a given workload in terms of monetary units. The model studies a system that involves a variety of resource types. For example, the capacity of an EC2 instance can be measured by looking at its CPU, memory, network band width, etc. The model assumes that each resource type can be allocated in units. The elasticity model is composed of two parts: penalty for over-provisioning and penalty for under-provisioning. The former captures the cost of provisioned but unutilized resources, while the latter measures opportunity cost from the performance degradation that arises with under-provisioning [4].

Our approach focuses on two features of the cloud which are elasticity and scalability. We define the key attributes of each characteristic and then derive some metrics that reflect the quality of the elasticity/scalability in a given cloud. Our metrics can help both service providers and final clients to score and evaluate cloud services. Our approach leads to an explicit measurement, and enables taking concrete decisions on the SLA aspects that are evaluated, Provisioning rates, and the particular Rate of Resources Allocation/Release.

A. Elastic Approach Vs. Static Approach

Traditionally, business organizations plan their computing infrastructure based on maximum expected computing resource capacity (i.e., fixed computing capacity as depicted in Fig. 2). Given today's dynamic and agile changes in business needs and growth, such capacity planning has to be more flexible and economical for two main reasons. First, traditional infrastructure capacity planning involves very large upfront capital investment which could reduce organization's cash flow considerably and it has a very long payback period. Second, such large computing capacity cannot be efficiently utilized [2]. As illustrated in Fig.2, there are time periods where resources are under-utilized. In addition to such resource waste, there are additional on-going costs which are important to maintain computing infrastructure operational and healthy (e.g., physical space, electricity power, management services, maintenance, etc.).

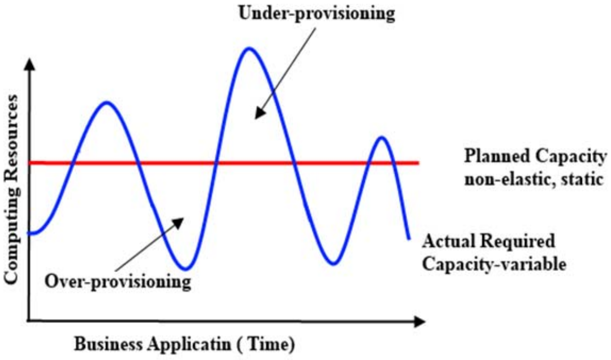

Fig. 2. Traditional provisioning Vs the Elasticity.

Figure2 illustrates elasticity and economics concepts in terms of computing infrastructure resources and also highlights some time periods where required application's computing capacity exceeds its planned capacity. Such scenario could occur because of unexpected workload spikes and/or business growth. Inability to meet such unexpected and dynamic computing capacity often leads to customer frustration and negative impact on organization's reputation and as a result potential loss of profit and customers [10].Otherwise, Cloud computing model aims to optimize the use of the resources. A highly-elastic cloud can ensure that that the resources provisioning follows exactly and dynamically the resources demand. This can be illustrated by the figure 3.

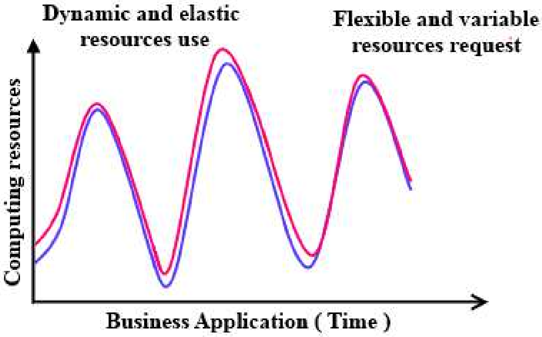

Fig. 3. Perfect Elasticity

B. Insight over Elasticity:

The term of elasticity comes from the world of the cloud. It is an important criterion of the quality of a virtual infrastructure: the ability to add or remove, in short to move resources (CPU, memory, space and hard disk performance mainly) in order to meet capacity needs. We can make an analogy between the virtual infrastructure and a balloon: to inflate or deflate a balloon more or less easily or quickly depends on the elasticity of it. Generally, several aspects reflect the quality of the elasticity in the cloud[13]:

- The speed to move the necessary resources – to inflate / deflate the balloon.
- The amount of resources displaced. The ability to move only certain resources – memory, CPU, disk space.

The elasticity is needed to meet peaks of workload, and answer the needs of scalability. However, these two notions should not be confused.

C. *Insight over Scalability:*

Scalability is the ability to meet the increased workload by adding resources. It can be measured as a ratio between loads vs. resources. If an application delivers a good performance with N resources for X users, and if its performance remains the same by multiplying resources by 2 when the number of users is multiplied by 2, then this application has a good scalability. If this ratio remains constant as the two factors increase, the scalability will be linear. This is optimal. But it is more likely that some components will not be up to the desired scalability, and therefore that performance stabilizes at a certain threshold, despite the addition of additional resources [13]. Cost reduction remains the main motivation when it comes to go into the cloud and virtualized infrastructures. Then the second main reason is Capacity management.

III. OUR METRICS

In this section we will define some metrics that describe the elastic behavior of the cloud. These metrics are derived from the attributes of the cloud. We first identify the key attributes of the elastic cloud. The obtained metrics return values that range from 0 as the lowest value to 1 as the highest value for all the sub-attributes we measure.

A. *Under provisioning average:*

Under-provisioning can be illustrated through the graphic (figure 4). It occurs when the requested resources are beyond the available resources or the allocated resources [6]. Associated with the factor of time this attribute can reflect the average of the unsatisfied requests during a certain period of time. This attribute is more relevant for the final client who pays a service provider when he wants to check that he received exactly the service level he pays for.

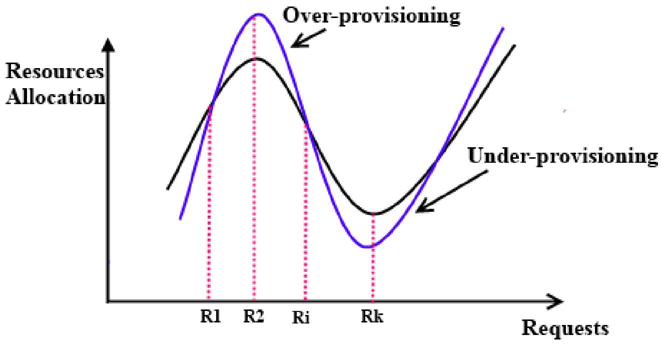

Fig. 4. Representation of the Resources provisioning

The graphic represents the requested resources (black curve) and the allocated resources (blue curve). For some requests the allocated resources are beyond/under the requested resources.

We define ‘n’ as a number of the requests received during a period of time DT and ‘i’ the number of the requests where the requested resources are beyond the available resources. Under provisioning average will be defined as the ratio $\frac{\mathrm{i}}{\mathrm{n}}$. This ratio increases when the number of the unsatisfied requests increases. However, from the client perspective, the metric should describe the cloud with the lowest under provisioning average as the best cloud. We then suggest:

$$UDPR_{DT} = 1 - \frac{\mathrm{i}}{\mathrm{n}}$$

The range is 0…1. The value 1 implies that no request in under-provisioned. The value 0 means that all the requests are under-provisioned.

B. *Over-provisioning:*

Over-provisioning can be illustrated through the graphic (figure 4). It occurs when the allocated resources are beyond the requested resources. From a service provider perspective; associated with the factor of time, this attribute can reflect the average of the unused resources [6] (loss in term of resources use).

The approach for quantifying over-provisioning is similar to what we did for under provisioning. We define n as a number of the requests received and ‘i’ as the number of the requests where the requested resources are beyond the available resources during a period of time DT. Under provisioning average will be defined as the ratio$\frac{\mathrm{i}}{\mathrm{n}}$. This ratio increases when the number of the unsatisfied requests increases. However, from the service provider perspective, the metric should describe the cloud with the lowest over-provisioning average as the best cloud. We then suggest:

$$OVPR_{DT} = 1 - \frac{\mathrm{i}}{\mathrm{n}}$$

The range is 0…1. The value 1 implies that no request in Over-provisioned. The Value 0 means that all the requests are Over-provisioned.

C. *Allocation reactivity:*

Another aspect that deserves a particular examination when we evaluate elasticity/scalability is the time required for a request to be processed. This aspect reflects the rapidity of the cloud to deal with requests and workloads increase. The formalization of such a determinant characteristic in SLA may be very helpful for both the service provider and the client. For every request received, we adopt the following

representation in which we distinguish three intervals of time [0,Ta],[Ta,Tp] and [Tp,Tr].
(fig5):

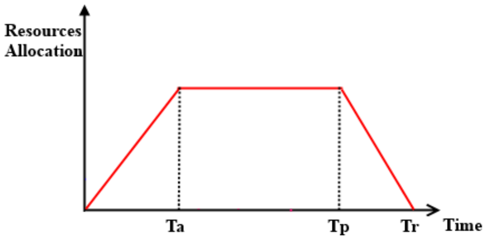

Fig. 5 Representation of the resources allocation

-The processing time, which is the time interval when the resource is fully dedicated to the instance related to the request. The end of the processing marks the end of the processing time [Ta,Tp].
-The Release time, which is related to the instant Tr when the resource has been effectively released and then can be affected to another request. The release Time can be seen as the interval [Tp,Tr].

In a perfectly elastic clouds, the allocation time Ta is close to 0 which means that the resources are instantly assigned to the corresponding requests once they are received. Otherwise, the resources can be released immediately once the processing has been finished which means that Tr may be close to Tp. This can be illustrated through the graphic in figure 6:

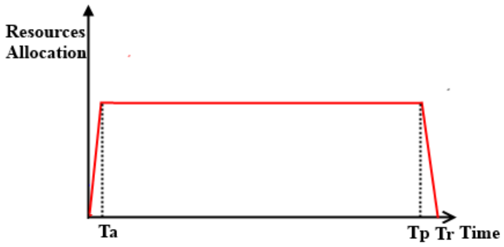

Fig. 6. Resources allocation with optimized Elasticity

To formalize the importance of this aspect we consider k as the number of requests received in a time interval. We associate to each request R_i where I ∈ [1, k] a corresponding allocation time Tai and processing timeTp_i. Cloud services with a good elasticity should be able to allocate the resources in a very short time. In a perfectly elastic model, e can assume that the allocation time $[0, Ta_i]$ should be negligible when compared with the processing time $[Tp_i, Ta_i]$. A metric that reflects the level of the cloud elasticity through several requests can be given by:

$$Al = 1 - \frac{\sum_{i=1}^{K} Min\left(1, \frac{Ta_i}{Tp_i - Ta_i}\right)}{\mathrm{K}}$$

The metric returns a value that ranges from 0 to 1 and reflects the reactivity of the cloud against the workload. It takes 0 when $Ta_i > Tp_i - Ta_i$ for every request R_i as the lowest possible value and takes 1 if $Ta_i = 0$ for every request R_i which means immediate responding to every request.

D. Release reactivity:

Similarly to the allocation time, the time required to release resources is another aspect that is worth a particular examination when we evaluate elasticity. In other words, it can be associated with the time elapsed from the instant the processing of a given request ends to the time the related resources are released effectively. The importance of this aspect comes from the fact that when a huge workload is received, the required resources are not released by other instances while the processing of the related requests has been finished. This situation results in delays in terms of processing and waste in terms of resources use.

The approach to measure this aspect is similar to the previous one adopted for the allocation reactivity. We consider K as the number of requests received in a time interval. We associate to each request $\mathrm{R_i}$ where I ∈ [1, k] a corresponding allocation time Tai, a processing time Tpi, and moreover a Release time Tri. If the cloud is perfectly elastic, the release time – which is the time difference between the two instants Tr_i and Tp_i $(Tr_i - Tp_i)$ – should be negligible in comparison with the processing time $[Tp_i, Ta_i]$. To illustrate this, we suggest the following metric which reflects the behavior of the resources allocation for several requests:

$$Re = 1 - \frac{\sum_{i=1}^{K} Min\left(1, \frac{Tr_i - Tp_i}{Tp_i - Ta_i}\right)}{\mathrm{K}}$$

The metric returns a value that ranges from 0 to 1 and reflects the reactivity of the cloud against the workload. It takes 0 when $Tr_i - Tp_i > Tp_i - Ta_i$ for every request R_i as the lowest possible value and takes 1 if $Tr_i = Tp_i$ for every request R_i which means immediate release for resources.

IV. Scalability Centric Metrics

Scalability reflects the ability of the system to deliver the same level of the performance when the workload/number of the requests increases. Measuring this characteristic means determining the limit to which users can request additional resources and see the same level of performance. Quantifying this aspect will result in defining new scalability metric. Otherwise, as the first figure shows, Elasticity and Scalability share several characteristics which means that the previously defined metrics remain valid but not sufficient to describe the full power of the scalability. Practically, we distinguish two types of scalability:

A. Vertical scalability:

Scaling vertically or scaling up means to add more resources to a given platform node, like CPU cores or memory in a way that the platform node can handle a larger workload. By scaling up a single platform node, physical limits that impact bandwidth, computational power etc. are often reached quite fast [3, 12] proofreading spelling and grammar:

B. *Horizontal scalability:*

Scaling horizontally or scaling out means adding new nodes (e.g. virtual machine instances or physical machines) to a cluster or distributed system in a way that the entire system can handle bigger workloads. Depending on the type of the application, the high I/O performance demands of the single instances that work on shared data often increase communication overheads and prevent the emergence of substantial performance gains, especially when adding nodes at bigger cluster sizes. In some scenarios, scaling horizontally may even result in performance degradation [3].

C. *Scalability metric:*

Jae Yoo Lee, and his team have presented a metric for Scalability: Coverage of Scalability. This metric measures the average amount of allocated resources among the amount of requested resources. This can be computed as [1]:

$$\frac{\sum_{i=1}^{K}\left(\frac{Amount\ of\ requested\ resources\ of\ ist\ request}{Amount\ of\ allocated\ resources\ of\ ist\ request}\right)}{\mathrm{K}}$$

Although this formula reflects the ability of the cloud to fit the increase of the workload; it doesn't describe the level to which the cloud can deliver the same level of performance when the number of requests increases. In our future work we will highlight this feature and try to define more specific metrics for the two mentioned types of Scalability.

V. Final Score Fomalization and Selection Algorithm:

From SLA perspective, a final elasticity score can be calculated and associated to a given cloud service. This score may help both the client and the service provider to evaluate the quality of the service. The score is derived from the previously calculated sub-scores of the corresponding attributes. In this section we will define our utility function which will enable us to calculate the final $QoS_{Elasticity/Scalability}$score for a cloud service.

We suppose that the selection of the cloud service is based on several parameters. Let $A = \{X_1, X_2, .., X_n\}$ be namely the vector of QoS attributes considered in the selection of the cloud service provider. Let $\mathrm{S} = \{SP_1, SP_2, .., SP_k\}$ be the set of the cloud service providers that can fulfill the needs of a given client. If we assume that the quality attributes are independent, the linear aggregate utility function can be defined by:

$$\mathrm{QoS} = w_1A_1 + w_2A_2 + .. + w_nA_n \qquad (1)$$

$$\text{With } \sum_{i=1}^{n} w_i = 1$$

A_i represents the individual utility function associated with the QoS attribute X_i, and W_i is the weight that the service consumer assigns to that attribute. Various functions may be used to express the service consumer utility of an attribute X_i. We use the aggregate function defined by Alvin AuYoung and Laura Grit [14] to express an utility function:

$$A_i = {x_i}^{\beta_i}$$

β_i is a measure of the service consumer sensitivity to the QoS attribute x_i. When $\beta_i = 0$, the service consumer is indifferent to QoS attribute x_i. When $\beta_i = 1$, the service consumer is moderately sensitive to QoS attribute x_i (the relationship is linear). When $\beta_i > 1$, the service consumer is increasingly sensitive to QoS attributex_i. As β_i increases, the service consumer is expressing increasing concern about x_i. For $\beta_i < 1$, as β_i decreases to approach 0, the service consumer is expressing increasing indifference to having x_i. The function (1) becomes:

$$\mathrm{QoS} = w_1{x_1}^{\beta_1} + w_2{x_2}^{\beta_2} + .. + w_n{x_n}^{\beta_n}$$

In our model we limit the set of parameters to four elements $A = \{X_1, X_2, X_3, X_4\}$ in which X_1, X_2, X_3, X_4 are associated respectively to the four aforementioned attributes: under-provisioning average, over-provisioning average, Allocation reactivity and release reactivity. In the rest of the paper we consider that β_i= 1 for every parameter. Then the expression (1) becomes:

$$\mathrm{QoS}_{Elasticity} = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$

The range of the final score is 0...1 and the higher value indicates that the Cloud is more elastic. In the scope of this paper, we assume that the client specifies the weights depending on his requirements.

VI. Proof of Concept

As a proof of concept, we consider a scenario where a client has to make a choice among four cloud service providers: $\mathrm{S} = \{SP_1, SP_2, SP_3, SP_4\}$based on their elasticity performances. Table 1 shows the weight and the sensitivity value associated to each Elasticity parameter.

TABLE I. Weight and Sensitivity associated to each parameter

	under-provisioning	over-provisioning	Allocation reactivity	release reactivity
ωi	0.15	0.15	0.35	0.35
βi	1	1	1	1

Let $\mathrm{QoS}_{Minimum}$ be the minimum score accepted in term of Elasticity score. We assume that the client would like to have $\mathrm{QoS}_{Minimum}$ = 0, 85. We make the assumption that the cloud service providers have found the following values in the elasticity attributes after calculating them through the metrics presented in this paper:

TABLE II. Final Scores of the service providers

	under-provisioning	over-provisioning	Allocation reactivity	release reactivity	Final score
SP1	0,8	0,9	0,8	0,75	0,7975
SP2	0,9	0,85	0,8	0,92	0,8645
SP3	0,95	0,9	0,9	0,92	0,9145
SP4	0,86	0,92	0,75	0,7	0,7745

The final scores of the service providers SP1 and SP4 are under the minimum required $\text{QoS}_{Minimum}$ = 0, 85. Service Providers SP2 and SP3 have an acceptable final score from the client perspective. SP3 seems to be the most elastic service.

VII. Quality model assessment:

In this section, we assess the validity of the final score formula presented in the section VI of this paper. For this purpose we consider the three criteria of the IEEE standard 1061 [15]:

- Correlation: The aforementioned metrics are obtained from quality attributes. There is an obvious linear association between these quality attributes metrics and the final score. For instance, the final score is a weighted sum of the presented metrics UDPV, OVPR, Re and All. According to our formula, each quality attribute can affect a calculated result of Elasticity/Scalability.
- Consistency: we consider three cloud services S1, S2, Sn. For each $\text{QoS}_{Elasticity}$ attribute (Over-provisioning, under-provisioning, allocation reactivity, release reactivity), if the values A1, A2, An, have relationship A1 > A2 > An, then the corresponding $\text{QoS}_{Elasticity}$ values have the relationship $\text{QoS}_{Elasticity1}$> $\text{QoS}_{Elasticity2}$ >$\text{QoS}_{ElasticityN}$. The final score of $\text{QoS}_{Elasticity}$ is calculated based on numerical values of various attributes. In other words, consistency of the formula is self-evident since it's a linear function of the values of the four attributes.
- Discriminative power: The formula of $\text{QoS}_{Elasticity}$ enables to rank cloud services and has a discriminative capability. If we consider three cloud service providers S1, S2, S2 (for which we know the corresponding values of the Elasticity attributes), the formula can rank the three cloud service providers based on a numerical calculation.

VIII. Conclusion:

In this paper, we presented a comprehensive quality model for evaluating Elasticity and Scalability in the cloud. First, we extracted several key-features of Elasticity/Scalability. We derived quality attributes from the key features. Then, we defined four metrics for evaluating these quality attributes. We believe that these metrics are meaningful for service providers. They cover essential features of Elasticity/Scalability and provide a way to evaluate the Cloud in quantitative manner. Through our quality model, service providers can evaluate the elasticity of their cloud services. Moreover, the metrics developed in this paper may result in better SLA formalization and may help a client to choose a service provider as we presented.

IX. Future work:

In this article we have focused on the elasticity aspect of the cloud. Our future work will emphasize more on the scalability metrics. Elasticity and Scalability are two different but correlated concepts that share several key-features. The metrics defined in the scope of this paper cover both Scalability and Elasticity but they remain not sufficient to describe the full power of the Scalability.

Otherwise, we would like to project our model on a real case. We will provide measures obtained from several cloud providers and then apply our Elasticity/Scalability model to compare them.

References

[1] Jae Yoo Lee, Jung Woo Lee, Du Wan Cheun, and Soo Dong Kim A Quality Model for Evaluating Software-as-a-Service in Cloud Computing .

[2] Basem Suleiman· Sherif Sakr· Ross Jeffery· Anna Liu, On understanding the economics and elasticity challenges of deploying business applications on public cloud infrastructure

[3] Rajkumar Buyya Dynamically Scaling Applications inthe Cloud

[4] Sadeka Islam, Kevin Lee, Alan Fekete, Anna Liu, How a consumer can measure elasticity for cloud platforms

[5] Deepal Jayasinghe, Simon Malkowski, Qingyang Wang, Jack Li, Pengcheng Xiong, and Calton Pu Variations in Performance and Scalability when Migrating n-Tier Applications to Different Clouds

[6] Michael Kuperberg Nikolas Herbst Joakim von Kistowski Ralf Reussner . Defining and Quantifying Elasticity of Resources in Cloud Computing and Scalable Platforms

[7] Thibault Dory, Boris Mej´ ıas Peter Van Roy Measuring Elasticity for Cloud Databases

[8] Saurabh Kumar Garg , Steve Versteegb, Rajkumar Buyya a A framework for ranking of cloud computing services

[9] Gao, J. SaaS performance and scalability evaluation in clouds

[10] Ahmed Ali-Eldin, Efficient Provisioning of Bursty Scientific Workloads on the Cloud Using Adaptive Elasticity Control

[11] Prasad Jogalekar, Murray Woodside Evaluating the Scalability of Distributed Systems

[12] Greg Barish, Scalable and High-Performance Web Applications

[13] Ricky Ho, Between Elasticity and Scalability

[14] AuYoung, Laura Grit, Service contracts and aggregate utility functions, Alvin

[15] Software Engineering StandardsCommittee, IEEE Standard for a Software Quality Metrics Methodology, page 11

978-1-4799-5979-2/14/$31.00 ©2014 IEEE