International Journal of Advanced Scientific Technologies, Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) Volume.3,Special Issue.1,March.2017

PRICE AND BENEFITS SLA MAPPING TECHNIQUE FOR OUTLINING STANDARDIZED ITEMS IN CLOUD COMPUTING MARKETS

R. Padmini
Academic Consultant,SoET
SPMVV,Tirupathi,India

E. Ramesh Babu
Academic Consultant, SoET
SPMVV,Tirupathi,India

Abstract— Due to the big variety in computing assets and, therefore, the big variety of various sorts of service Level Agreements (SLAs), any marketplace for computing resources faces the capacity trouble of a low marketplace liquidity. To counteract this hassle, supplying a set of standardized computing sources is suitable. each of these standardized computing resources is described via an SLA template. An SLA template defines the shape of an SLA, the attributes, the names of the attributes, and the attribute values. When you consider that those SLA templates are presently static, they can't replicate adjustments in users' needs. To address this shortcoming, we present the radical method of adaptive SLA matching. This approach adapts SLA templates based on SLA mappings by means of allowing Cloud customers to outline mappings between public SLA templates, that are to be had in the Cloud market, and their private SLA templates, which are used for diverse in-residence business procedures of the Cloud consumer. Except displaying how public SLA templates adapt to the call for of users, we additionally examine the benefits and costs of this technique. Charges are incurred each time a consumer has to define a new SLA mapping to a public SLA template because of its variation. In particular, within this paper, we look into the cost relying on the usage of exclusive public SLA template edition methods. The simulation results show that the usage of heuristics within model strategies enables balancing the price and gain of the SLA mapping method.

Index Terms—Cloud Computing, Service Level Agreements,

I.INTRODUCTION

Computing useful resource allocations in Clouds are based not most effective on practical requirements however also on distinct non-useful necessities. those non-practical requirements (e.g., software execution time, reliability, and availability) are termed nice of service (QoS) necessities and are expressed and negotiated with the aid of carrier level Agreements (SLAs). so as to facilitate the introduction and management of SLAs, SLA templates were added. SLA templates, which constitute popular SLA codecs, include factors including names of buying and selling parties, names of SLA attributes, measurement metrics, and attribute values [1]. In Cloud computing markets, customers and sellers of computing sources face the hassle of varying definitions of computing assets. Computing sources are defined via special non-standardized attributes (e.g., CPU cores, execution time, inbound bandwidth, outbound bandwidth, and processor kind). [4]. dealers use them to describe their deliver of resources and consumers use them to describe their call for for resources. as a result, a big range of various SLAs exists within the marketplace. The achievement of matching asks (i.e, gives of sellers) and bids (i.e., gives of buyers) becomes not possible [1]. processes tackling this plethora of SLA attributes encompass using standardized SLA templates for a selected client base[5,6], downloadable predefined providerspecific SLA templates [7], and the use of ontologies [8,9]. these methods absolutely outline SLA templates and require users to agree a priori on predefined requirements. The SLA templates are static. However, the call for of customers modifications over the years. For instance, the emergence of multi-core architectures in computing sources required the inclusion of the brand new characteristic "range of cores", which turned into now not present in an SLA template more than one years ago. however, the prevailing methods for the specification of SLA templates can't without problems cope with call for changes. these tactics contain heavy person-interactions to adapt existing SLA templates to changing marketplace conditions.

In this paper, observe adaptive SLA mapping, a new technique that may react to converting market situations [1]. This technique adapts public SLA templates, that are used within the Cloud marketplace, based totally on SLA mappings. SLA mappings, which have been described by using customers based on their desires, bridge the

differences between existing public SLA templates and the personal SLA template (i.e., the SLA template of the user). considering the fact that a consumer can't easily change the personal SLA template due to internal or legal organizational requirements, an SLA mapping is a handy workaround.

The benefits of SLA mappings for marketplace members are threefold. first off, traders can keep their private templates, which are required for different business techniques. Secondly, based totally on their submitted mappings of private SLA templates to public SLA templates, they make contributions to the evolution of the market's public SLA templates, reflecting all traders' desires. Thirdly, if a set of recent products is brought to the market, our method can be carried out to discover a set of new public SLA templates. a lot of these blessings result inglad users, who hold to use the market, consequently growing liquidity in the Cloud market. however, these advantages come with some cost for the person. each time a public SLA template has been adapted, the users of this template have to re-define their SLA mappings. The 5 contributions of this paper are: (1) the definition of the suitable use case to exemplify the adaptive SLA mapping approach; (2) the definition of three variation techniques for adapting public SLA templates to the needs of the consumer; (3) the investigation of situations below which SLA templates should be tailored; (four) the formalization of measures (i.e., utility and cost) to assess SLA adaptations and SLA model methods; and (five) the creation of an emulation approach for the use cases. The the rest of the paper is prepared as follows: segment 2 describes related paintings. section three introduces the adaptive SLA mapping approach and the application and value model. The simulation setup, the three edition methods, and the simulation infrastructure are described in phase four. Phase 5 offers the simulation outcomes and a dialogue. Phase 6 concludes the paper.

II. RELATED WORK

For putting this paintings in context of the today's, we in short describe Cloud marketplaces and the existing paintings on SLAs. currently, a massive quantity of business Cloud vendors have entered the utility computing marketplace, presenting a number of distinctive styles of services. We distinguish between computing infrastructure offerings, which can be pure computing assets on a pay-per-use basis [11, 12, 13], software program services, that are computing sources in mixture with a software solution [6, 14], and platform services, which permit customers to create their personal offerings in mixture with the assist of assisting services of the platform company. the first kind of offerings consists of a virtual machine, as in the case of Amazon's EC2 service, or inside the form of a computing cluster, as finished with the aid of Tsunamic technologies. The variety of sources presented through a issuer is low. as an example, Amazon and EMC added only three derivations in their simple useful resource type [5]. Examples for the second one type of services are offerings supplied with the aid of Google (Google Apps [6]) and Salesforce.com [14]. those corporations provide get entry to to software program on pay-according to-use basis. these software program-as-a-provider (SaaS) solutions can hardly be integrated with other solutions, because of their huge range. Examples for the third type of Cloud offerings are solar N1 Grid [15], pressure.com [14], and Microsoft Azure [16]. on this category, the focus lies on provisioning critical simple services which might be wished via a large variety of programs. those fundamental offerings can be ordered on a pay-in keeping with-use basis. even though the intention of these services is a unbroken integration with the users packages, standardization of interfaces is largely absent. Concluding, we are able to kingdom that, apart from first tries in the provider type infrastructure as a provider, standardization tries do nearly now not exist. The principle SLA matching mechanisms are based on DAML-S, or comparable semantic technologies. [8] describe a framework for semantic matching of SLAs based totally on WSDL-S and OWL. [9] gift a unified QoS ontology relevant to precise scenarios such as QoS-based internet services selection, QoS tracking, and QoS edition. [17] present an autonomic Grid architecture with mechanisms for dynamically reconfiguring service middle infrastructures. it's far exploited to satisfy varying QoS necessities. Besides those mechanisms, [10] speak self sustaining QoS management, the usage of a proxy-like method for defining QoS parameters that a carrier has to keep throughout its interaction with a selected consumer. The implementation is based on WS-agreement, the usage of predefined SLA templates. however, they can't don't forget adjustments in consumer wishes, that is critical for developing a hit markets, as proven in our in advance paintings [1]. several works on modern-day SLA management are supplied in [2]. besides, no matter the sort of method used, those strategies do not evaluate and provide an explanation for the advantage and fees through the advent of SLA matching mechanisms.

II ADAPTIVE SLA REPRESENTING

In this section, we present a use case for Representing SLA mapping. Besides, we discuss the SLA life cycle and introduce the utility and cost model for assessing SLA matching approaches.

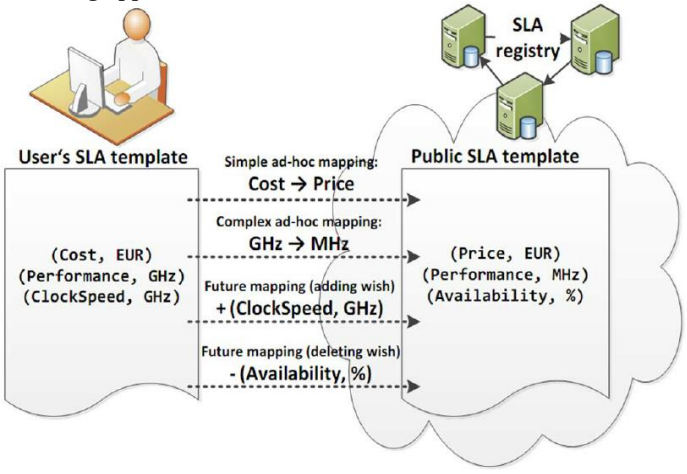

Fig. 3.1 Use Case

Since resources can be exposed as services using typical Cloud deployment technologies (i.e.,SaaS/PaaS/IaaS), we assume that the service provider of Figure 1 registers its resources (e.g., infrastructure, software, platforms) to particular public templates (step 1, Figure 1). If some differences between its resources (private SLA template) and the public templates exist, the provider defines SLA mappings, transforming the private template into the public template (step 2, Figure 1). The management of SLA mappings, which is performed with VieSLAF, is explained in detail in [3]. In step 3, Cloud users can look up Cloud services that they want to use in their workflow. In Figure 1, we exemplified a business process (i.e, workflow) for medical treatments [18]. It includes various interactions with human beings (e.g., the task of getting a second opinion on a diagnosis) as well as interaction with different infrastructure services. Some of these tasks (e.g., the reconstruction of 2-dimensional SPECT images to 3-dimensional SPECT images) can be outsourced to the Cloud [18]. Thereby, we assume that the private SLA template (representing the task) cannot be changed, since it is also part of some other local business processes and has to comply with different legal guidelines for electronic processing of medical data. Therefore, in case the user decides to outsource a task and discovers differences between the private SLA template and the public SLA template, the user defines an SLA mapping. The mapping describes the differences between the two SLA templates (step 4). A typical mapping is the mapping of an attribute name to another attribute name (e.g., number of CPUs to cores) or the inclusion of a new SLA attribute (e.g., parallel programming models) into the SLA template. The public SLA templates are stored in searchable repositories using SQL and non-SQL-based databases (e.g., HadoopDB). The SLA mappings, which have been provided by users and providers to the entity managing the public SLA templates, are evaluated after certain time periods, in order to adapt the public SLA templates to the needs of the users. The adapted public SLA templates replace the existing public SLA templates in the repository, constituting our novel approach of adaptive SLA mapping. The adaptation method, which adapts the public SLA templates, performs it such that the new public SLA templates represent user needs better than the old SLA templates (step 5). Besides the adaptation of attribute names and attribute values, the adaptations can also include definitions of new branches of templates (e.g., a medical SLA template can be substituted by more specialized templates on medical imaging and surgery support). The definition of different versions of a particular template is also possible as shown for the templates in the bioinformatics domain (step 6).

3.2 Threshold Method

As a way to boom the requirements for deciding on the most candidate, this method introduces a threshold value. If an attribute name is used extra than this threshold (which can be tailored) and has the very best count number, then this attribute name can be decided on. If multiple is above the brink and they have the equal rely, the method proceeds as defined for the maximum technique. If none is above the required threshold, then the method sticks to the presently used attribute name. observe, in the course of the examples on this paper, we restore the edge to 60%. instance: Assuming an instance in which none of the attribute names has a mapping percent above 60% and all counts are identical, the edge technique sticks to the characteristic call that is presently used inside the public SLA template.

3.3 Maximum-Percentage-Change Method

This technique is divided into steps. inside the first step, the attribute call is selected in keeping with the maximum technique. In the 2nd step, which contains τ iterations, attribute names might be changed, most effective if the proportion difference among the very best count attribute name and the currently selected characteristic call exceeds a threshold. the brink σT is set to fifteen%. A low threshold ends in greater mappings, whereas a excessive threshold leads in average to fewer mappings. After τ iterations (e.g., $\tau = 10$), the approach re-starts with executing step one. It lets in even slighter adjustments to take impact. Instance: allow's assume the mapping be counted resulted in characteristic call A ′ having the best matter. by means of making use of the most technique, A′ is chosen. within the subsequent iteration, the variety of mappings for every attribute name has modified. characteristic name A accounted for 10%, A′ for 28%, A ″ for 32%, and A‴ for 30% of all mappings. Assuming a threshold of 15%, the chosen attribute does not alternate. the share difference between attribute call A′ and the attribute name A″ with the highest count number is simplest thirteen.three%.

3.4 Utility and Cost Model

For the reason that intention of this paper is to assess the advantage and the value of using the adaptive SLA mapping approach for finding the most suitable standardized items in a Cloud marketplace, we outline a utility model and a value version. The software function and the value function, which take attributes of the patron's SLA template and the attributes of the general public SLA template as input variables, allows to quantify the benefit and price. For our application version, we expect an boom in advantage, if an attribute of each templates is identical. this is stimulated via the truth that the Cloud useful resource traded is identical to the want of the client (or the provisioned resource of the company) and, consequently, no inefficiency via useful resource over-provisioning takes place. The cost version captures the attempt of converting an SLA mapping. A fee to the consumer is handiest incurred, if the user wishes to change its SLA mapping due to a trade inside the public SLA template. To officially introduce those fashions, we introduce some definitions. The set of SLA attributes is defined as Tvar. For instance, we set Tvar = α, β, wherein α represents wide variety of Cores in a single CPU and β represents amount of CPU Time (word, α and β may also constitute characteristic values). All feasible attribute names that a user can map to a $\pi \in$ Tvar are denoted as Var(π). within our instance, we set Var(α) = A,A′,A″,A‴, representing Var("range of Cores in one CPU") = CPU Cores, Cores of CPU, range of CPUCores, Cores, and

Var(β) = B, B′, B″, B‴. Assuming a set of consumers' private templates C = c1,c2,...,cn, we will now outline the connection of a particular SLA characteristic to a selected name of this SLA attribute at the new release i ∈ N for every non-public and public template p, p ∈ C∪ T as

SLAp, I : Tvar →[π∈ Tvar Var(π). (1)

With recognize to our example, we count on SLAT,0(α) = A and SLAT,0(β) = B as our preliminary public template T at generation zero. primarily based on those definitions, we outline the application characteristic u+c, I and the value feature u−c,I for consumer c, attribute π ∈Tvar, and new release

i ≥ 1 as u+c,i(π) = (1, SLAc,i(π) = SLAT, i(π) 0, SLAc, i(π) 6= SLAT, i(π)

(2) SLAc, i(π) = SLAT, i(π)
SLAc,i(π) 6= SLAT, i(π)∧
SLAT,i−1(π) = SLAT,i(π)
1/2, SLAc, i(π) 6= SLAT ,i(π)∧
SLAT,i−1(π) 6= SLAT, i(π)

(3) We pick out our application feature as exemplified in [20]. The software feature states that a client c receives a software of 1, if the name of the attribute of the private SLA template matches the name of the public SLA template characteristic, and a software of zero otherwise. The cost characteristic states that a customer has a price of half of, if the attribute names do not suit and the general public template characteristic of the ultimate new release modified to a brand new one. In this example, the customer has to define a new attribute mapping, as he can not use the antique one anymore. in the different two instances, the consumer has no price, seeing that either the characteristic names in shape or the general public template characteristic call did not alternate since the ultimate generation. which means he does no longer need any new mapping. for that reason, for attribute π, the purchaser c at generation i gets the internet utility

uoc,i, π = u + c, I (π)−u−c, I (π).

(4)The net utility for all attributes at generation i for customer c is defined as the sum of the net utilities

uoc, I, π: uoc, i = ∑π∈ Tvar ,uoc, i,π.

(5) The overall software and overall price (i.e., the application and value of all users C and attributes π at generation i) are described as:

U + i = ∑c ∈C,∑π ∈ Tvaru + c, i(π)

(6) U−i = ∑c ∈C∑π∈ Tvaru − c, I(π)

(7) Consequently, the overall internet utility at new release i is defined as the distinction among the general utilities minus the overall price:

Uoi = U + i –U − i.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Net Utilities of Adaptation Methods

Using the SLA mapping method, the user gets the advantage of getting access to public SLA templates that replicate the general market demand (i.e., the common consumer's call for). This benefit of some consumer is expressed with equation 2. however, this comes with the value for defining new SLA mappings every time the public SLA template modified (equation three). within this phase, we inspect the cost of all users (equation 7), the software of all users (equation 6), and the net application of all users (equation 8) for exceptional version methods. The internet application metric is used to decide which of the 3 model techniques is superior. the primary adaption approach that we investigate is the maximum approach. it's miles our reference approach, since it does not use any heuristics. The simulation consequences, which are shown on this segment, have been acquired from running the simulation with parameter settings as defined in segment . The simulation results shown are averages over all eventualities. The advantage of this method is that the general public SLA template generated with this method minimizes the variations to all private SLA templates of all users. This approach calls for, however, many modifications of SLA mappings.

V .CONCLUSION AND OUTLOOK

In this paper, we've got investigated value, application, and internet utility of the adaptive SLA mapping technique, in which marketplace participants may additionally define SLA mappings for translating their non-public SLA templates to public SLA templates. opposite to all other available SLA matching methods, the Representative SLA mapping approach facilitates continuous variation of public SLA templates based totally on marketplace developments. but, the model of SLA mappings comes with a value for users within the form of effort for generating new SLA mappings to the adapted public SLA template. To calculate the value and benefits of the SLA mapping technique, we utilized the SLA control framework VieSLAF and simulated distinct marketplace conditions. Our findings show that the fee for SLA mappings may be decreased with the aid of introducing heuristics into the variation techniques for producing adapted public SLA templates. The methods show fee reduction and growth in common average net utility.

REFERNCES

[1] M. Risch, I. Brandic, J. Altmann. Using SLA Mapping to Increase Market Liquidity. NFPSLAM-SOC 2009. In njunction with The 7th International Joint Conference on Service Oriented Computing, Stockholm, Sweden, November 2009.

[2] R. Buyya, K. Bubendorfer. Market Oriented Grid and Utility Computing. John Wiley & Sons, Inc., New Jersey, USA, 2008

[3] I. Brandic, D. Music, P. Leitner, S. Dustdar. VieSLAF Framework: Enabling Adaptive and Versatile SLA-Management. GECON2009. In conjunction with Euro- Par 2009, 25- 28 August 2009, Delft, The Netherlands.

[4] M. Risch, J. Altmann. Enabling Open Cloud Markets Through WS-Agreement Extensions. Service Level Agreements in Grids Workshop, in conjunction with GRID 2009, CoreGRID Springer Series, Banff, Canada, October 2009.
[5] Amazon Elastic Compute Cloud (Amazon EC2), http://aws.amazon.com/ec2/, 2010.
[6] Google Apps, http://www.google.com/apps/, March 2010.
[7] Business Objective Driven Reliable and Intelligent Grids for Real Business (BREIN), http://www.eu-brein.com/, February 2010.
[8] N. Oldham, K. Verma, A. P. Sheth, and F. Hakimpour. Semantic WS-agreement partner selection. 15th International Conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 2006. [9] G. Dobson, A. Sanchez-Macian. Towards Unified QoS/SLA Ontologies. IEEE Services Computing orkshops (SCW), Chicago, Illinois, USA, pp.18-22, September 2006.
[10] B. Koller, L. Schubert. Towards Autonomous SLA Management Using a ProxyLike Approach. ltiagent Grid Systems. vol.3, no.3, IOS Press, Amsterdam, The Netherlands, 2007.
[11] M. Risch, J. Altmann, L. Guo, A. Fleming, C. Courcoubetis. The GridEcon Platform: A Business Scenario Testbed for Commercial Cloud Services. 6th international Workshop on Grid Economics and Business Models, Delft, The Netherlands, August 2009.
[12] TsunamicTech.Inc., http://www.clusterondemand.com/, 2010.
[13] EMC Atmos Online, https://mgmt.atmosonline.com/, 2010.
[14] Salesforce.com, http://www.salesforce.com, March 2010.
[15]SunGrid, http://www.sun.com/service/sungrid/index.jsp, 2010.
[16]MicrosoftAzure, http://www.microsoft.com/windowsazure/, 2010.
[17] D. Ardagna, G. Giunta, N. Ingraa, R. Mirandola, and B.Pernici. QoS-Driven Web Services Selection in Autonomic Grid Environments. International Conference on Grid Computing, High Performance and Distributed Applications (GADA), Montpellier, France, November 2006.