Hybrid Cloud Selection Approach to Automate the Cloud Service Selection based on Decision Support System

G. UdhayaKumar[1] N. Rajganesh[2] S. Sindhuja[3]

[1]*Final Year Student*

[2]Assistant Professor

[3]Assistant Professor

Department of IT

A. V. C College of Engineering

Udhayatweety88@gmail.com

Rajganesh@avccengg.net

Abstract— *Cloud computing is the latest computing paradigm that de-livers hardware and software resources as virtualized services in which users are free from the burden of worrying about the low-level system administration details. Migrating Web applications to Cloud services and integrating Cloud services into existing computing infrastructures is non-trivial. It leads to new challenges that often require innovation of paradigms and practices at all levels: technical, cultural, legal, regulatory, and social. The key problem in mapping Web applications to virtualized Cloud services is selecting the best and compatible mix of software images. Clear and consistent assessment of the various capabilities of cloud service providers (CSPs will become an essential factor in deciding on which CSPs to use in the future, particularly as cloud service provision expands further into more sensitive and regulated areas. This paper describes an approach that is useful in this regard. Specifically, we describe a mechanism in which context is gathered relating to CSPs this is inputted to a rule-based system and decisions are output about the suitability of each CSP, including an analysis of privacy and security risk and recommended stipulations to be taken into account when negotiating contracts and SLAs.*

Keywords: Cloud Computing, Migration Process, Decision Support, Selection Algorithm, Factors, Criteria, Service Selection, Automation.

1. INTRODUCTION

Migrating legacy applications to the cloud is a non-trivial task as it leads to technical challenges. So present the generic framework Cloud-Genius. And an implementation that leverage well known multi- criteria decision making technique Analytic Hierarchy Process. The emergence of Cloud computing over the past five years is potentially breakthrough .

Cloud providers including Amazon Web Services (AWS), Salesforce.com, or Google App Engine give users the option to deploy their application over network of infinite resource pool with practically no capital investment and with modest operating cost proportional to the actual use. By leveraging Cloud services to host Web applications organizations can benefit from advantages such as elasticity, pay-per-use, and abundance of resources.

Cloud computing being a disruptive technology an adoption brings along risks and obstacles [2]. Risks can turn into effective problems or disadvantages for organizations that may decide to move Web applications to the Cloud.

Considering this increases the complexity of the decision associated with migrating a Web application to the Cloud. Such a decision depends on many factors, from risks and costs to security issues and service level expectations. Another critical hurdle is the complexity of migrating a Web application to the Cloud on a technical level while incorporating economical aspects. A migration from an organization-owned data center to a Cloud infrastructure service implies more than few trivial steps.

The following steps outline a migration of an organization's Web application to an equivalent ona Cloud infrastructure service. Steps of a migration to a Platform-as-a-Service (PaaS) ordering would delilver in several Regards. First, an appropriate Cloud infrastructure service, or Infrastructure as-a-Service (IaaS) ordering, is selected. This demands a well-thought decision to be made that considers all relevant factors, like e.g., price, Service Level Agreement (SLA) level, or support quality.

The basis of a selection are data and measurements regarding each factor that describe the quality and make service options comparable. Secondly, the existing Web application and its platform, a Web server, are transferred from the local data center to the selected Cloud infrastructure service. Therefore, the Web application and server must be converted into a format expected by a Cloud infrastructure service. Typically, in this step the whole Web application is bundled as a virtual machine (VM) image that consists of a software stack from operating system and software platforms to the software containing the business logic. Since it is often unachievable to convert an existing Web application and its server directly to a Cloud infrastructure service compatible VM image format, an adequate existing VM image ordered by the Cloud provider is chosen and customized. Existing images vary in many ways, such as underlying operating system, software inside the software stack or software versions. Hence, selecting a functionally correct VM image becomes a complex task. Besides, choosing a comprehensive VM image helps to minimize the effort of installing a software stack on a basic image.

In more complex settings multiple components and databases must be migrated in parallel, what requires to apply the steps described above component-wise. Additionally, interconnections and relations between the components must be considered.

In this paper we introduce the Cloud Genius framework that lowers hurdles introduced by the complexity of the Cloud migration process. Cloud Genius offers a detailed process and comprehensive decision support that reduces a Web engineer's effort of finding a proper infrastructure service and VM image when migrating a Web application to the Cloud. The paper is structured as follows. First, we reflect related work in Section 2 and then present the Cloud Genius framework in Section 3. Further, we give an example application of Cloud Genius in Section 4 and present Cumulus Genius, a prototypical implementation, in Section 5. In Section 6 we present the results of experiments on Cumulus Genius' time complexity. Propose an approach that translates both selection steps into multi-criteria decision-making problems

To determine the most valuable combination of a Cloud VM image and a Cloud infrastructure service.

2. SYSTEM ARCHITECTURE:

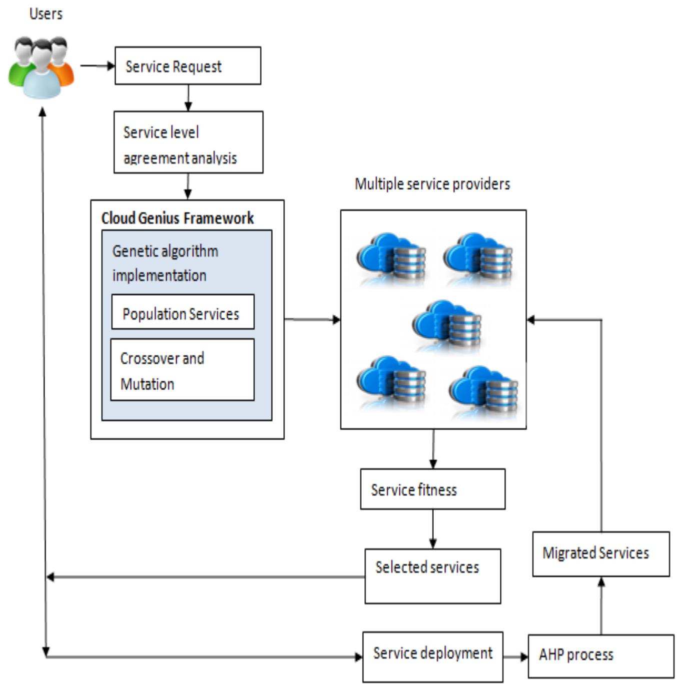

Fig. 1 System architecture Proposed system

In automate the cloud service selection are classified as five modules. They are

- Service Request
- Cloud Framework
- Service Selection
- Migration of web services
- Evaluation criteria

2.1 Service Request

These keywords are set as service request .
The service request also includes service level agreements. This module SLA includes cost, validity and service name.

2.2 loud Framework

Owning to its complexness, the traditional approach that user still need to essentially compose construct service oriented architecture that cloud framework for selecting services are delivered to the consumer through a third party entity the support of the user in selecting the provider that better meets his SLA requirements.

2.3 Service selection

Services Can implement PSO to select the best services. PSO a population of strings (called chromosomes or the genotype of the genome)The new population is then used in the next iteration of the algorithm.

2.4 Migration of web service

The service migration is done by multi-criteria decision analysis. MCDA, one of the most important branches of operations research. It can be done by Analytic Hierarchy Process (AHP).Attributes structured into a hierarchal relationship, which is very useful.

2.5 Evaluation criteria

Proposed the AHP based decision-making model to select a suitable cloud service provider focused on the IaaS provider for companies' users. The three criteria such as provider, service and support perspectives for decision making were identified and the hierarchy model was also constructed. The Evaluation Framework focuses on key criteria to help a company assess cloud implementation strategies.

Migrating an application to the Cloud entails a loss of control over the QoS characteristics of the application due to the reliance on the QoS levels offered by the service provider. As a result, the QoS characteristics offered by a Cloud service provider appear have a greater importance to application stakeholders than hosting the application in a non-Cloud environment. The most obvious case of this dependence is the effect of service provider outages to businesses hosting their applications with them. As discussed in (Badger et al., 2012), availability is usually calculated in practice on the basis of the billing cycle, which may vary from minutes to hours, and not on the total up-time of the service. Furthermore, failure to provide this availability is typically compensated with service credit in future use of the services. In any case, the overall availability of a migrated application can actually vary significantly with respect to the availability of the used services, depending on the actual behavior of the application. Safely predicting the availability of such an application is largely an open issue.

service selection are there are various alternative mechanisms for providing the information about what the CSP will do: using the DSS directly with the customer, or else using the tool to produce output that is then accessible publicly, e.g, via a website which contains the CSP's answers to common question the DSS would ask. These CSP policies can then be taken as input by the DSS in order to make an evaluation for the current organization's context. The motivation for CSPs to provide input for the tool is market forces – otherwise, only the 'lowest' default policy within the system would be allocated to their offering as this process In addition relevant customer requirements are explicitly displayed to the SP admin so he can clearly understand what the minimal requirements of the customer are used to encrypted.

3.1 DECISION SUPPORT FOR CLOUD SERVICE SELECTION:

What becomes obvious from the previous discussions that migrating an application to the Cloud requires making a number of decisions related to the how the application is to be refractor for the Cloud, how it is supposed to scale, an so on. In this sense, cost analysis and performance prediction, which has been the focus of the Sot A as discussed in Section 2 provide only partial support for these purposes. Addressing these concerns, It provides an overview of the concepts required to implement a decision support system to support the migration of applications to the Cloud. This conceptual view defines our vision for what constitutes a complete solution for application developers and stakeholders alike that are considering whether and how to migrate their application to the Cloud.

More specifically, two types of concepts are identified decisions that need to be made (and therefore are the focus of the system), and tasks that need to be performed in order to support these decisions and that in turn affect their outcome.

Each of these decisions has a direct or unintentional influence on the others, illustrated with transparent arrows in Fig. 1. Deciding on the appropriate elasticity strategy for example is meaningless without the Cloud service provider supporting it. The selection of service provider includes the decision which service delivery model to use. As a result, both the options in distributing the application and how multitenancy can be implemented is directly influenced by it.

In addition to the decisions and their relationships, the following tasks are also identified.

3.2 Work Load Profiling.

Defining or estimating the expected work load profile of the application is prerequisite for both other tasks to be performed (performance calculation and cost analysis), as well as input for any decision on how to distribute the application, and which elasticity strategy fits this profile.

3.3 Compliance Assurance.

Ensuring the compliance to regulations regarding, e.g., privacy of personal data, affects directly both the selection of the provider (especially in terms of location of the data service), as well as how the application is to be distributed so that e.g. personal data may need to be retained on premises.

3.4 Identification of Security Concerns.

Defining which data and communications are critical to be protected drives the selection of an appropriate service provider that fulfills these security constraints. Delivering the application in a multitenant manner imposes further constraints with respect to data isolation that need to be considered for this task.

3.5 Identification of Acceptable QoS Levels.

Based on existing and planned SLAs, acceptable levels for QoS characteristics like availability of the service provider can be inferred. Beyond service provider selection, this task also drives the definition of an appropriate elasticity strategy to ensure the cloud selection process.

3.6 Services.

Based on this information the CSP will be asked questions such as the location of the servers to be used, access control mechanisms used, encryption mechanisms used, purposes for which data will be used, whether there is further outsourcing sharing of information with other organizations, and the mechanisms in place that the CSP has for enforcing the organization's requirements along the chain, backup provisions, etc.

USECASE for Proposed system:

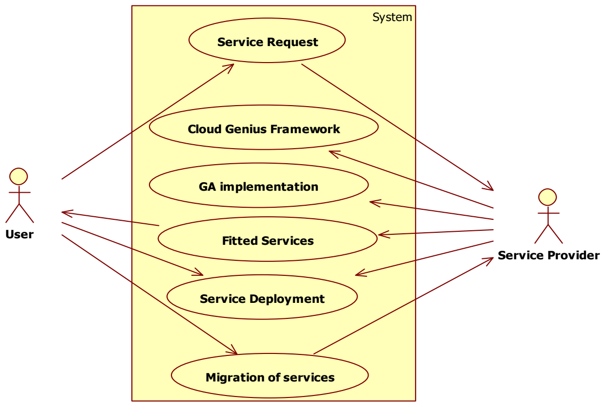

Fig. 2 Use case for proposed system

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations.

Use cases have been submitted from various TC members, but for ease of consumption and comparison, each has been presented using an agreed upon "Use Case Template. The cloud infrastructure uses identities. If there is a configuration that is an accepted standard, then it is easier to migrate the configuration across cloud infrastructures. This type of migration is desirable to permit subscribers the ability easily move applications between cloud deployment types and between cloud providers without loss of the identities associated with the applications. This category contains a listing of one or more the cloud employment or service models that are featured in the use cases.

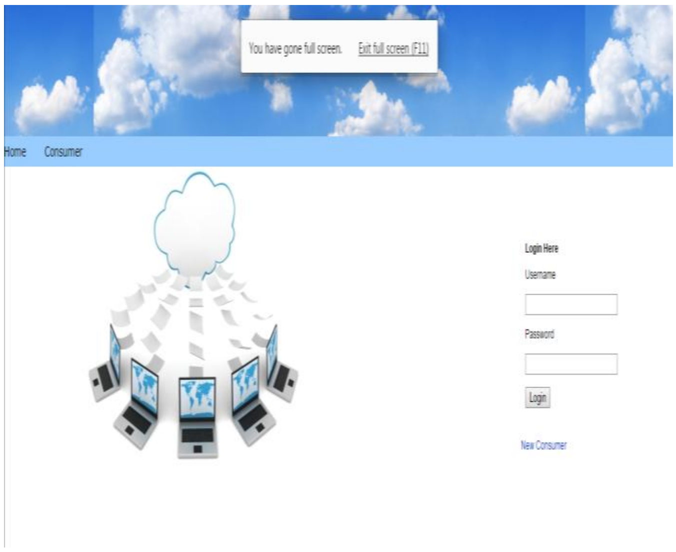

Fig. 3 Login System

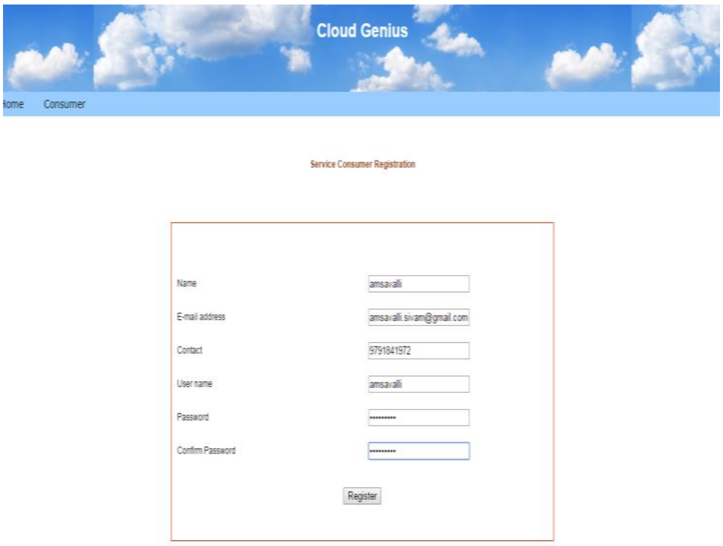

Fig. 4 User Details

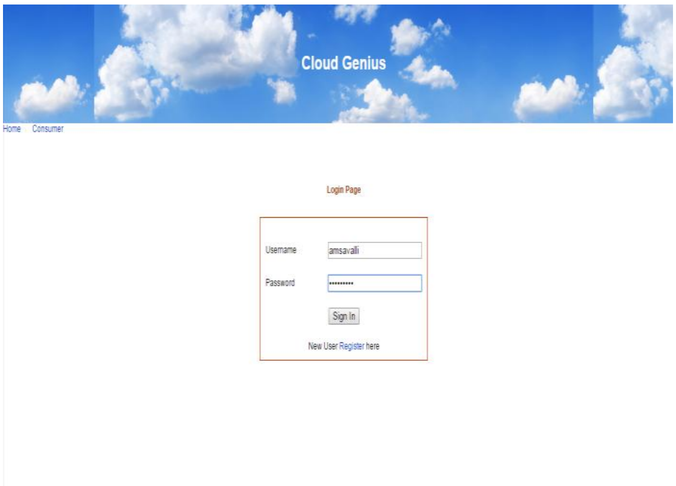

Fig. 5 User Login After service selection

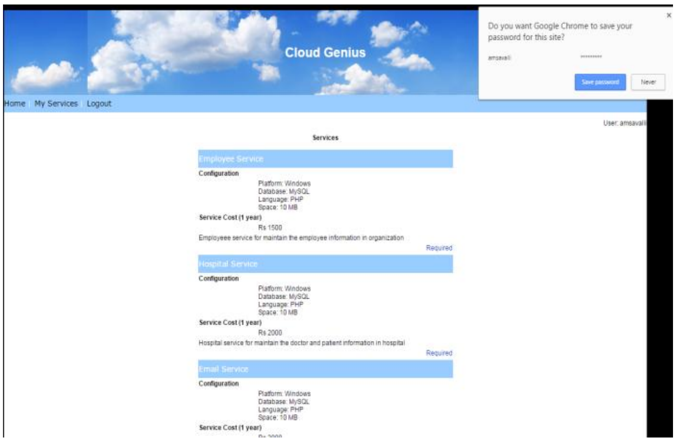

Fig. 6 User Service entry details

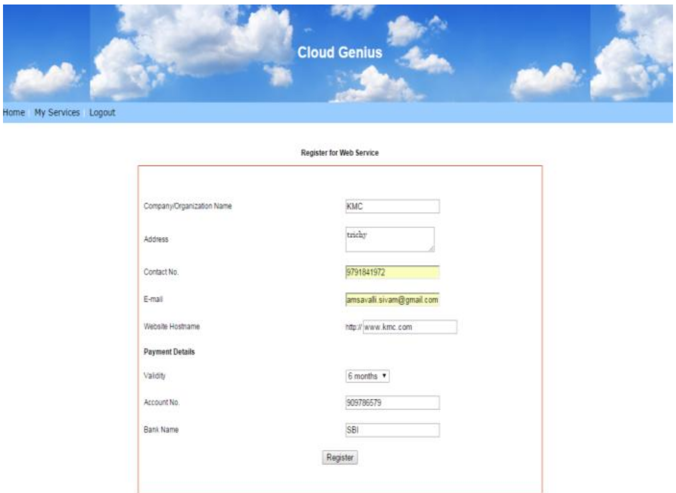

Fig. 7 Service Registration Form

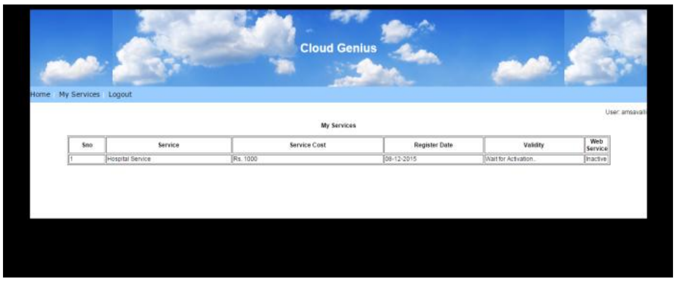

Fig. 8 Cloud Services

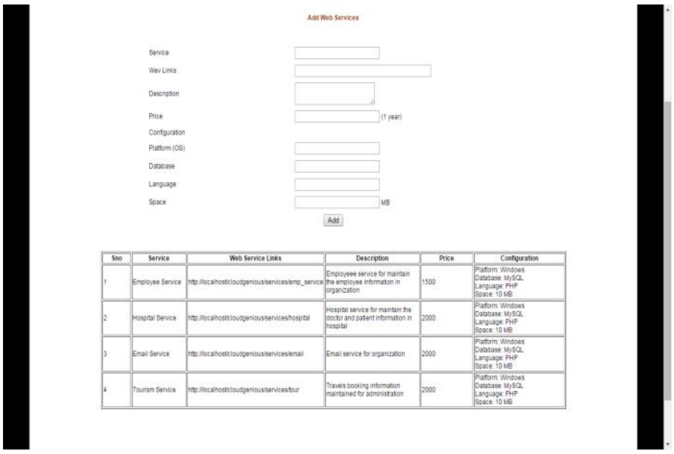

Fig. 9 User Acceptance cloud region

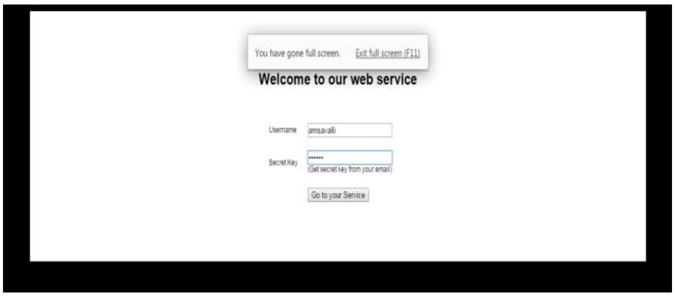

Fig. 10 Invitation Form

In web service welcome page are above, Depending on how the application is distributed across different service

providers, which providers where selected for this purpose, and which elasticity strategy was defined, this task creates projections about the nonfunctional behavior of the application after it is migrated to the Cloud. These projections can in turn be used to change any of the previous decisions, starting a new feedback loop.

It providing multiple points of entry to the decision making process of migrating an application to the Cloud. A Cloud migration decision support system should therefore support this process not in a structured hierarchical manner, but rather in a networked fashion, allowing application designers to provide different answers for each decision that needs to be made, ask them for the appropriate input required to perform the identified tasks, and demonstrate the effects of their decisions to concrete goals like the cost and performance of the application. Such an approach is missing from the current State of the Art and constitutes our future work in the area of service selection process

Conclusion:

In this paper, we presented the extended CloudGenius

framework, which provides a hybrid approach that combines multi-criteria decision making technique with evolutionary optimization technique for: (i) helping application engineers with the selection of the best service mix at IaaS layer and (ii) enabling migration of web application clusters distributed across clouds.

The application to the Cloud is a multi-dimensional problem with multiple decisions to be taken: how to segment and distribute the application, which Cloud service provider and offering to choose, which elasticity strategy fits better the application, and what are the requirements in implementing application multitenancy. These decisions influence each other, and depend on a number of tasks like cost analysis and performance prediction that need to be performed in order to support them. Related to these decisions and tasks are a series of research challenges stemming from the need to adapt the application to operate in the Cloud environment that also need to be addressed. Our vision for a decision support system for migration of applications to the Cloud service selection.

Reference

1) M. Menzel and R. Ranjan, “CloudGenius: Decision Supportfor Web Server Cloud Migration,” in Proceedings of the 21st International Conference on World Wide Web, ser. WWW ’12. New York, NY, USA: ACM, 2012.

2) R. Mietzner, T. Unger, and F. Leymann, “Cafe : A Generic ConfigurableCustomizable Composite Cloud

Application Framework," Framework, pp. 357–364, 2009.

3) M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski,G. Lee, D. Patterson, A. Rabkin, I. Stoica et al., "Above the Clouds: A Berkeley View of Cloud Computing," EECS Department,University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.

4) C. Zeng, X. Guo, W. Ou, and D. Han, "Cloud computing service composition and search based on semantic," in

Cloud Computing, ser. Lecture Notes in Computer Science,

M. Jaatun, G. Zhao, and C. Rong, Eds. Springer Berlin

Heidelberg, 2009

5)M.Menzel, M. Klems, H. A. Lˆe, and S. Tai, "A configuration

crawler for virtual appliances in compute clouds," in International Conference on Cloud Engineering (IC2E) 2013.