HYBRID CLOUD SELECTION APPROACH TO AUTOMATE THE CLOUD SERVICE SELECTION BASED ON DECISION SUPPORT SYSTEM

HEMACHANDIRAN.S [1],GEETHA.N [2]

[1]*PG Student, Dept of CSE, Bharathiyar College of Engineering and Technology, Karaikal.*

[2]*Assistant Professor, Dept. of CSE, Bharathiyar College of Engineering and Technology, Karaikal.*

Abstract— Cloud computing is the delivery of computing resources on demand with reduced management effort. One of the key problems in migrating multi-component enterprise applications to Clouds is selecting the best mix of VM images and Cloud infrastructure services. A migration process has to ensure that Quality of Service (QoS) requirements are met, while satisfying conflicting selection criteria, e.g. throughput and cost. The main problem in mapping software applications to cloud services is selecting the best and most compatible software components to ensure a cost- effective model. When selecting components to migrate the cloud, software engineers must consider many criteria and complex dependencies among other systems' components. Thus, a technique for locating components to be migrated without actually moving them is needed. To overcome these challenges, we propose an approach which can be used in the hybrid decision-making process based on a set of measurable factors in the pricing models of cloud providers. Then, a proposed cost measuring function is used to choose the optimal migration scenarios. And implement Genetic algorithm to find the fitness values to select optimal cloud service system to overcome complexities in web application systems.

Keywords- Cloud Computing, Multi criteria Decision Analysis, Software as a Service, Virtual machine.

1. INTRODCUTION

Cloud computing systems fall into one of five layers; applications, software environments, software infrastructure, software kernel, and hardware. At the bottom of the cloud stack is the hardware layer which is the actual physical components of the system. At the top of the layers is the cloud application layer, which is the interface of the cloud to the users through web browsers and computing terminals. The ability of clouds to add or remove resources within few minutes allows matching resources to workload much more efficiently. Employing elasticity allows

reducing this resources waste. Moreover, it makes use of the economic benefits of the cloud by adopting the "pay-as-you go" concept since hours purchased via cloud computing is usually distributed non-uniformly in time. Thus, framework is used to measure the placement of each service. Factors which are included in most cloud providers framework are the transferred data between the cloud's components and the on-premise components (the outgoing and incoming bandwidths), the processing power utilization (PPU) or the compute utility, and the storage allocated by the client in GB. Other factors include the recipients' emails, additional public IP addresses, and RAM usage, etc. We think that, in migration scenarios of software components, the amount of the transferred data and the PPU are among the most critical and measurable factors. This is because they contribute to the running cost when cloud solutions are adopted. Thus, if a hybrid method is to be followed, communication between migrated parts and other on-premise parts in addition to processing power utilization must be minimized.

1.1Cloud Computing Models

Cloud Providers offer services that can be grouped into three categories.

1. Software as a Service (SaaS): In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced. On the customers‟ side, there is no need for upfront investment in servers or software licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted & maintained. Today SaaS is offered by companies such as Google, Salesforce, Microsoft, Zoho, etc.

2. Platform as a Service (Paas): Here, a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider's infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySql and PHP), restricted J2EE, Ruby etc. Google‟s App Engine, Force.com, etc are some of the popular PaaS examples.

3. Infrastructure as a Service (Iaas): IaaS provides basic storage and computing capabilities as standardized services over the

network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, GoGrid, 3 Tera, etc

1.2 MULTICRITERIA DECISION ANALYSIS

One of the major challenges facing an entrepreneur in business prioritization entails coming-up with a reliable model that will rank the available business opportunities (where should we invest). A useful class of models that rank the opportunities is called multi-criteria decision analysis (MCDA) that deals with decisions involving the choice of a best alternative from several potential candidates, subject to several criteria such as those faced by a potential investor. Although MCDA technique seems to offer a natural mechanism to tackle problems of this nature, there is no evidence of their use. Partly this is because the problem is an inter-disciplinary in nature (i.e. Entrepreneurship and Management Science) but mainly because MCDA requires intervention of Management Science that lacks in an entrepreneur. We therefore propose to build the first MCDA business selection tool with inherent family of models to solve the problem. In essence, one way of establishing this relevance is through a study involving statistical correlation analysis. Second, as individual investors have boundaries on the investment capability and preferences; wish to develop a framework that would account these limitations. This will only be possible if, for each question asked, we enumerate the possible responses from the investor. As the responses are directly linked to the model, it is essential that the validity of such responses is supported via a triangulation research technique. The third challenge entails weighting the responses from the potential investors and linking them to the MCDA model. In this case, weight normalization heuristic needs to be developed and embedded in the MCDA model. While numerous methods exist for weights normalization, we propose a popular rating method that requires the responses to be expressed on a numerical scale. Finally, for the model to work we requires a database hooked into the MCDA model that enumerates possible business opportunities available within a region. Initially, a sample size based on stratified random sampling technique will be selected. The sample will then be analyzed and generalized for the

remaining regions (population) through a fall-back principle that uses statistical induction. MCDA is a structured approach to decision-making that quantitatively evaluates alternatives, in this case, metrics, based on defined project criteria, expert opinions, and stakeholder preferences. It integrates a wide variety of information to evaluate project alternatives and rank them based on their aggregated value with respect to a set of criteria. It usually consists of four stages. The project team, incorporating expert and stakeholder opinions, must define: (1) the set of possible decision alternatives (in this case, metric alternatives) to be evaluated and ranked; (2) the criteria of the value tree that will influence the decision that these alternatives will be evaluated against; (3) the importance of each criterion relative to the others or their "weight" followed by a normalization of weights performed separately for each order of criteria (criteria of order one, criteria of order two (or sub-criteria), etc.); and (4) the value of each alternative with respect to each criterion. Depending on the specific MCDA method, (3) and (4) may also include uncertainty estimates. The hypothetical monitoring goal of the optimal set of metrics is twofold: (i) to select the best restoration alternative; and, (ii) to evaluate restoration project success by measuring the degree to which the intended objectives have been achieved following the project implementation period

2. RELATED WORK

CloudGenius: Decision Support for Web Server Cloud Migration

In this paper, introduce the CloudGenius framework that lowers hurdles introduced by the complexity of the Cloud migration process. CloudGenius 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 order reflects the fact that an image can be chosen for a certain Cloud infrastructure service only. Alternatively, selecting a Cloud VM image first restrains the number of eligible Cloud infrastructure services, typically to one. 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. With CloudGenius we 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. The CloudGenius framework defines a Cloud migration process.

A Petri Net-based Model for Web Service Composition

In this paper, we propose a Petri net-based algebra for modeling Web services control flows. The model is expressive enough to capture the semantics of complex service combinations and their respective specificities. The obtained framework enables declarative composition of Web services. We show that the defined algebra caters for the creation of dynamic and transient relationships among services. A Web service behavior is basically a partially ordered set of operations. Therefore, it is straight-forward to map it into a Petri net. Operations are modeled by transitions and the state of the service is modeled by places. The arrows between places and transitions are used to specify causal relations. We assume that a Petri net, which represents the behavior of a service, contains one input place (i.e., a place with no incoming arcs) and one output place (i.e., a place with no outgoing arcs). A Petri net with one input place, for absorbing information, and one output place, for emitting information, will facilitate the definition of the composition operators and the analysis as well as the verification of certain properties (e.g, reach ability, deadlock, and liveness).

Portable Cloud Services Using TOSCA

In the life cycle's production phase, the cloud management platform uses management plans to manage the service instance for compliance with the service-level agreements (SLAs) negotiated at subscription time. For example, the management platform assigns additional resources to the instance when the number of users increases, and removes them when users are no longer using the service. The cloud service provider or consumer can also trigger management plans manually — for example, to back up or upgrade the service. Finally, when the cloud service consumer decides to get rid of the service or the subscription expires, the service instance terminates, and all the resources go back into the resource pool. TOSCA describes composite applications and their management in a modular and portable fashion. It thus defines service templates that contain a cloud service's topology (for instance, an application is hosted on an

application server, which is in turn hosted on an operating system) and its operational aspects (such as how to deploy, terminate, and manage this service). Service templates are interpreted by a TOSCA-compliant environment, which operates the cloud services and manages their instances. The creator of a cloud service captures its structure in a service topology — a graph with nodes and relationships. Nodes represent the service's components, and relationships connect and structure nodes into the topology

Cloud Model for Service Selection

Firstly, previous selection approaches do not consider the uncertainty of QoS seriously. QoS values of the composite applications are usually computed by aggregated QoS values of the remote Web services, which may come from different organizations, implemented by different programming languages, and run on different platforms. Secondly, with the increase of web services in the Internet, computation time of the web service selection approaches becomes larger. Real-time optimal web service selection becomes more and more difficult. Existing approaches focused too much on the optimization of selection algorithms themselves to reduce computation time and neglected the basic impact factor (i.e., the exponential increase in the number of web services). To address these challenges, we present a cloud model based web service selection approach. The main contributions of this paper can be summarized as follows. (1) We address the problem of web service selection and demonstrate the influence of uncertainty of QoS on the service selection process. (2) We propose a novel concept, called QoS uncertainty computing, to model the inherently uncertain of Web service QoS. We adopt cloud model to compute the uncertainty of QoS. According to the three numerical characteristics of cloud model, the web services with a large variance on their QoS can be pruned. To the best of our knowledge, this is the first work that computes the uncertainty of QoS for web service selection. (3) Based on cloud model, we propose a fast and reliable QoS-aware service selection approach.

3. PROPOSED SYSTEM

By leveraging Cloud services to host web services can benefit from advantages such as elasticity, pay-per-use, and abundance of resources. However, users tend to avoid or delay migrations of service deployment to the Cloud due to multiple hurdles. With

Cloud computing being a disorderly technology an implementation brings along risks and obstacles. The proposed system handles hurdle is the complexity of migrating services to the Cloud on a technical level while incorporating economical aspects. A migration from an organization-owned data center to a Cloud infrastructure service imply more than few unimportant steps. The following steps outline a migration of an organization's Web application to an equivalent on a Cloud infrastructure service. Steps of a migration to a Platform-as-a-Service (PaaS) contribution would differ in numerous regards. There are two major approaches for migration to the cloud. The first approach is to progress the whole application to the cloud. On the other hand, adopt hybrid migration. The former approach is likely to provide privileged response times. In hybrid migration, some parts of the application are stimulated to the cloud, while other parts are kept on basis based on migrated process. This hybrid approach done using Genetic algorithm and Analytical Hierarchy Process (AHP).

3.1 GENETIC ALGORITHM

In a genetic algorithm, a population of services (called chromosomes or the genotype of the genome), which encode candidate services (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better services. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of services has been produced, or a satisfactory fitness level has been reached for the population.

3.2 ANALYTICAL PROCESS (AHP):

The AHP is based on a couple wise comparisons, with the attributes prepared into a hierarchal relationship, which is very useful. The hierarchy starts from the top level towards the goal; the lower levels correspond to criterion, sub-criteria, and so on. In this hierarchy tree, the process starts from leaf nodes and progress up to the top level. Each output level represents the hierarchy equivalent to the weight or influence of different kindling originating for that level. Finally, after making the

comparisons, the best option with respect to each attribute is usually selected.

3.3 MODULES

Service Request

Web service provides communication between two electronic devices over a network. It is a software function provided at a network address over the Web with the service always on as in the concept of utility computing. The term *Web services* describes a standardized way of integrating web based applications using XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall. In this module, user search services using keywords. These keywords are set as service request. The service request also includes service level agreements.

Cloud Framework

With the development of the cloud and related technology, more and more persons and enterprises start to implement their business process by using some resources, which belong to other organizations. Owning to its complexness, the traditional approach that user still need to essentially compose detailed workflow descriptions by hand can't satisfy the requirements the cloud application in reality. So we can construct service oriented architecture that cloud framework for selecting services. It can be done by service brokering process. Service brokering is a business model where services are delivered to the consumer through a third party entity or company called a broker, who acts as mediator between the two parties. With the emergence of Cloud computing, service brokering has been adopted to add new business values to Cloud services. Among them is the support of the user in selecting the provider that better meets his SLA requirements.

Service selection

With the growing number of alternative services in the cloud environment, users have put forward new requirements to solve

the service dynamic selection problem quickly and efficiently. In this paper, an evaluation model of service process which considers concurrent requests and service association is proposed. So in this module, we can implement genetic algorithm to select the best services. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population

Migration of web services

The service migration is done by multi-criteria decision analysis (MCDA). MCDA enables the user to choose from among a number of available choices. MCDA, one of the most important branches of operations research, aims to design mathematical and computational tools for selecting the best alternative among several choices, with respect to specific criteria, either by a single decision maker or by a group. It can be done by Analytic Hierarchy Process (AHP). The AHP is based on a pair wise comparison, with the attributes structured into a hierarchal relationship, which is very useful. The hierarchy starts from the top level towards the goal; the lower levels correspond to criteria, sub-criteria, and so on. In this hierarchy tree, the process starts from leaf nodes and progresses up to the top level. Each output level represents the hierarchy corresponding to the weight or influence of different branches originating for that level.

Evaluation criteria

In this module, 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. Our proposed approach provides improved accuracy rate and efficiency for service selection.

CONCLUSION AND FUTURE WORK

For solving the problem of discovering a user's optimal parameter portfolio for service level and evaluating the properties of any kind of candidate cloud services, we have proposed the cloud service selection model, Cloud Genius to evaluate the properties and select the optimal service which satisfies both user-specified service level and goals most. In hybrid migration, software engineers face the problem of locating the optimal set of components to be migrated statically before actually moving them and calculating the benefit associated with each available migration scenario. This is even more difficult for large systems such as enterprise applications. We conclude that migration scenarios can be guided statically by measuring the degree of coupling among migration candidates. Experimental results emphasize the opinion that less computation and more generic components are more suitable for migration.

6.2 FUTURE WORK:

In future work, we can extend our approach to improve service selection using particle swarm optimization algorithm. And service can be selected quickly and accurately. PSO is one of the important evolutionary algorithms in service selection.

REFERENCES

[1] M. Menzel and R. Ranjan, "CloudGenius: Decision Support for 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. Hamadi and B. Benatallah, "A Petri Net-based Model for Web Service Composition," in Proceedings of the 14th Australasian database conference-Volume 17. Australian Computer Society, Inc., 2003, pp. 191–200.

[3] T. Binz, G. Breiter, F. Leyman, and T. Spatzier, "Portable cloud services using tosca," Internet Computing, IEEE, vol. 16, no. 3, pp. 80–85, 2012.

[4] S. Wang, Z. Zheng, and Q. Sun, "Cloud model for service selection," . . . WKSHPS),

2011 IEEE . . . , no. 60821001, pp. 666–671, 2011.

[5] R. Buyya, R. Ranjan, and R. Calheiros, "Intercloud: Utilityoriented federation of cloud computing environments for scaling of application services," Algorithms and Architectures for Parallel Processing, pp. 13–31, 2010.

[6] "Web Application Hosting in the AWS Cloud Best Practices," http://media.amazonwebservices.com/AWS Web Hosting Best Practices.pdf, accessed 2013-07-19., Amazon Web Services.

[7] 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.

[8] P. Mell and T. Grance, "The nist definition of cloud computing, recommendations of the national institute of standards and technolog," NIST Special Publication, 2011.

[9] "BitNami Cloud Images," http://bitnami.org/learn more/ cloud images, accessed 2011-12-02, BitNami.

[10] "The Cloud Market," http://cloudmarket.com, accessed 2011- 10-19.