

Dynamic Resource Selection in Cloud Service Broker

Ganis Zulfa Santoso¹, Young-Woo Jung¹, Seong-Woo Seok¹, Emanuele Carlini², Patrizio Dazzi², Jörn Altmann³, John Violos⁴, and Jamie Marshall⁵.

¹ Cloud Computing Research Department, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea. Email: {ganis,jungyw,swsok}@etri.re.kr

² Institute of Information Science and Technologies, National Research Council of Italy (CNR) Pisa, Italy. Email: {emanuele.carlini,patrizio.dazzi}@isti.cnr.it

³ College of Engineering, SNU, Rep. of Korea. Email: altmann@snu.ac.kr

⁴ ICCS-NTUA, Greece. Email: john@mail.ntua.gr

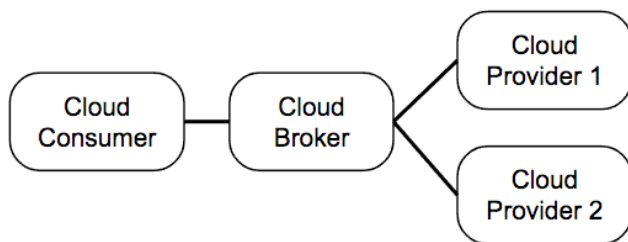
⁵ Amenesik SARL, St Pierre Les Nemours, France, e-mail: ijm@amenesik.com

Abstract—Cloud Service Broker federates multiple Cloud Service Providers into a single entity to customers. The benefits that can be enjoyed by Cloud Service Consumer are flexibility, ease of use, and reduced cost. However, because of the unique properties and configurations of each cloud provider, sometime it's not easy to migrate between one cloud provider to another. Furthermore, the advantage of using broker should be obtained by consumers in any life cycle of consumer's software not only during the deployment of the software. This paper outlines the main idea and design of the dynamic resource selection in BASMATI Cloud Federation.

Keywords—Cloud Federations; Resource Selection; Cloud Service Broker; Cloud Service Provider; Cloud Resources; Cloud Service Sonsumer;

I. INTRODUCTION

Cloud Service Broker acts as an intermediary between Cloud Service Consumer and Cloud Service Provider [1] as described by Picture 1 below. Broker selects the best provider for consumer according to consumer's requirements and provider's performance. Among others, the benefits of using a broker for a consumer are increased flexibility, ease of service, and reduced cost.



Picture 1. Usage Scenario for Cloud Service Brokerage

In terms of flexibility, by not contacting multiple cloud providers directly, consumer should only create an account in broker and the respective consumer will be granted access to multiple providers. By using broker, consumer can avoid to be locked in a single provider and get the freedom to move between cloud providers.

Cloud providers have their own terms and definitions for features in their products that may confuse potential customers. To make it even worse, the specifications of the hardware, software, and network are hard to be compared. Broker tries to provide the easy use of the service by simplifying the resource selection to the customers.

Broker can help cut down the cost of renting servers in cloud providers by assisting consumer in selecting the most suitable infrastructure of platform for consumer's software.

The mentioned benefits that are provided by broker for consumer are mostly enjoyed only during the deployment of the software. In other words, it's a static resource selection. However, the cloud computing industry is changing insistently. A new feature, a new cloud provider, a new datacenter, or a new pricing scheme is being introduced rapidly by the market. The already deployed software should take the latest offers from the market into consideration as they probably can improve performance and reduce cost. For example, the recent launch of Singapore datacenter by Google Cloud Platform (GCP) [2] should be considered by nearby customers in that area. Therefore, it is needed for a feature in broker to actively looking for better available resources for the customers. Such feature is called 'dynamic resource selection'.

It's not really a clear cut process on software or application migration between cloud providers. Each provider has their own quirks and configurations on how the software can run on their IaaS (Infrastructure as a Service) and PaaS (Platform as a Service). This is a major disadvantage for consumers since they are locked in certain vendors [7]. Therefore, the initial benefit, flexibility, is nullified by provider's inflexibility. For a migration between a software or an application between cloud

providers to be conducted smoothly, certain requirements need to be integrated into broker.

II. REQUIREMENTS

In order for broker to meet the goals described in previous section, there are many requirements which need to be fulfilled. First, the broker should actively and periodically measure the computing and network power of available resources in providers with a standardized benchmark method, especially for the newly available resources. The benchmark should cover all the metrics that are commonly considered by customers. Furthermore, the benchmark should consist of synthetic benchmark and real world benchmark. This is needed so the power of the underlying infrastructure of a cloud provider can be objectively compared and broker can convincingly recommend certain broker for a consumer. The problem that can arise from this method is probably the cost and the time to run and test each available resources in the industry. Fortunately, in cloud computing, it's common to bill by hour or even minute. For example, to run one of the highest specification of virtual machines in Amazon Web Service such as *m3.2xlarge* will cost less than half dollar for an hour [3]. A broker can run hundreds instances from different types for few hours in a month and will not break a bank. Furthermore, provider are providing API to access their platform therefore the benchmarking can be done automatically and periodically without human intervention. It is also important to do it across datacenters in the same provider to gather the most representative and the most accurate performance of the infrastructure.

Another requirement is provider should know the requirements of consumer's application. It can be gathered manually by consumer with a format that is understandable by broker, or automatically gathered by a monitoring system integrated in broker. The monitoring system should be able to measure the load and predict the required specification based, if exists, on the recent usage of the software. With the detailed information on both provider and consumer, broker can accurately give a recommendation on the best resources that suit the application. As mentioned before, the benchmarking and monitoring of should be done periodically in any stage of application to achieve the dynamicity of resource selections in broker.

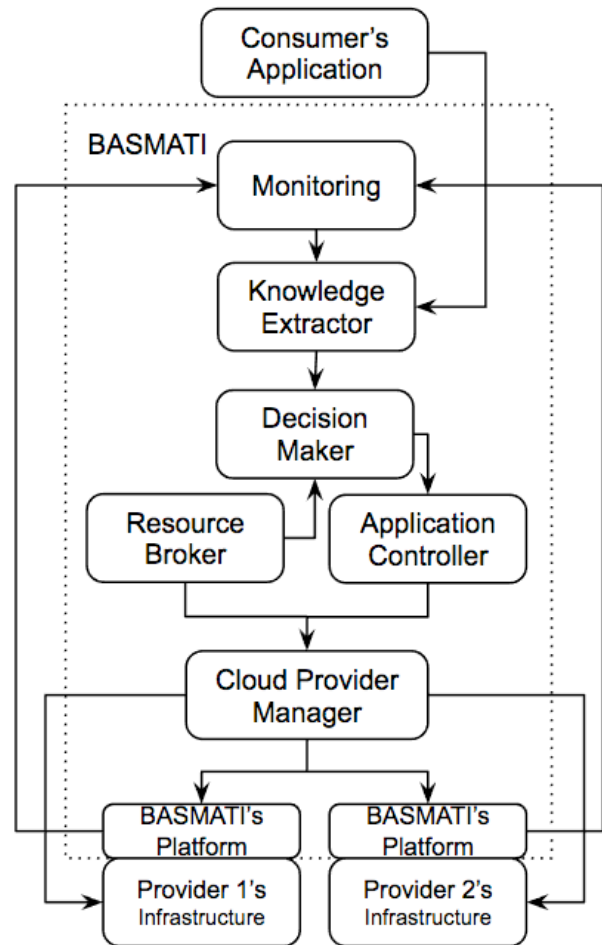
Once the previous requirement is accomplished, another requirement is the adaptability of consumer's application into different provider. Because if broker is recommending consumer to migrate to another provider but it's not possible to do so, then it's not a smart thing to recommend it in the first place. For the adaptability to be possible, a common platform needs to be available in all providers. It needs to be an open source or a free platform to ease the propagation to all provider. The application of consumer is running on top of this platform thus during migration, consumer doesn't know that the underlying provider is actually changing. Before migration, broker should install the platform to target provider.

III. DESIGN

BASMATI – *Cloud Brokerage Across Borders for Mobile Users and Applications* is a joint research project between South Korea and EU Horizon 2020 in developing a federated cloud platform. One of the goals of the project is to support the dynamic resource selection. The architecture and diagram of BASMATI to realize that feature is described by Picture 2. It is need to be strongly noted that the components and the flows that are described here are the minimum in achieving the dynamic resource selection. The whole architecture of BASMATI is not discussed.

A. Consumer's Application

The application is feeding the Knowledge Extractor (KE) about its requirements. Furthermore, consumer can also inform the KE about the detailed usage of users of its application to further improve the accuracy of the requirements.



Picture 2. The minimum components in BASMATI to realize the dynamic resource selection feature.

B. Monitoring

Monitoring monitors the usage and load of running applications in the infrastructure. The information is feed by

BASMATI's platform. The collected information will be gathered and aggregated in a form that's understandable and easy to be analyzed.

C. Knowledge Extractor

KE is analyzing information from both Monitoring and consumer's application. KE should be able to give a recommendation on what's the needed infrastructure to run the applications without any hiccup at all. Since it's very possible for a consumer to either overestimate or underestimate the requirements. The recommendation should also consider the time and date because usage load of an application is highly influenced by those metrics.

D. Decision Maker

Decision Maker (DM) is gathering information from both KE and Resource Broker (RB) on what is the best plan for the application. It needs to carefully calculate whether the migrations to the new provider is beneficial and profitable. When a decision is already made and execution of a plan is needed, DM is calling Application Controller (AC) to execute it.

E. Resource Broker

The function of RB is to give recommendations on what are the best resources available for the running applications. As mentioned before, the benchmark should be conducted in provider's servers. Therefore, RB periodically contacting provider Manager to run a virtual machine or a container in providers and conducting benchmarks. The benchmark result is gathered in RB and analyzed to give reasoning behind RB's recommendation.

F. Application Controller

AC is responsible to control the life cycle of the application since its deployment until its removal. During migration to a new provider, AC should precisely conduct it in a manner that no downtime is occurring.

G. CSP Manager

Cloud Service Provider Manager has the capability to contact multiple cloud providers through an interface. The manager also responsible to install BASMATI's Platform (BP) in each server of cloud providers. The examples of cloud service provider manager in the industry are Compatible One [4], Scalr [5], RightScale [6], and so on.

H. BASMATI's Platform

BP is a platform that is actually a hypervisor, or a container depending on the application, which encapsulates the

consumer's applications. As mentioned before, it is required that the platform is an open source or a free software. At the moment the most suitable candidates are KVM and Docker.

I. Cloud Provider's Infrastructure

Cloud provider's infrastructure is the physical hardware that are running the consumer's applications. The hardware can be cloud or edge resources.

IV. OUTLOOK

This paper presents the main principles and design decision of the BASMATI Cloud Federation project in meeting the dynamic resource selection in Cloud Service Brokerage. Currently, the many components that are displayed here are in the process of development. After the development is finished, the integrations between components need to be conducted.

ACKNOWLEDGMENT

This research has been carried on with support of the BASMATI project. BASMATI (<http://basmati.cloud>) has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement no. 723131 and from ICT R&D program of Korean Ministry of Science, ICT and Future Planning no. R0115-16-0001

REFERENCES

- [1] Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L. and Leaf, D., 2011. NIST cloud computing reference architecture. *NIST special publication*, 500(2011), p.292.
- [2] Lardinois, Frederic. "Google Launches Its Cloud Platform Region In Tokyo." *TechCrunch*. TechCrunch, 08 Nov. 2016. Web. 27 Apr. 2017.
- [3] "EC2 Instance Pricing – Amazon Web Services (AWS)." *Amazon Web Services, Inc.* N.p., n.d. Web. 27 Apr. 2017.
- [4] S. Yangui, I.-J. Marshall, J.-P. Laisne, and S. Tata, "Com- patibleone: The open source cloud broker," *Journal of Grid Computing*, vol. 12, no. 1, pp. 93–109, 2014.
- [5] "Enterprise-Grade Cloud Management Platform." *Scalr*. N.p., n.d. Web. 27 Apr. 2017.
- [6] "RightScale Cloud Management Platform." *RightScale Cloud Management*. N.p., n.d. Web. 27 Apr. 2017.
- [7] Carlini, Emanuele, Massimo Coppola, Patrizio Dazzi, Laura Ricci, and Giacomo Righetti. "Cloud federations in contrail." In *European Conference on Parallel Processing*, pp. 159-168. Springer Berlin Heidelberg, 2011.