Dynamic Resource Selection in Cloud Service Broker

Ganis Zulfa Santoso[1], Young-Woo Jung[1], Seong-Woo Seok[1], Emanuele Carlini[2], Patrizio Dazzi[2], Jörn Altmann[3], John Violos[4], and Jamie Marshall[5].

[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

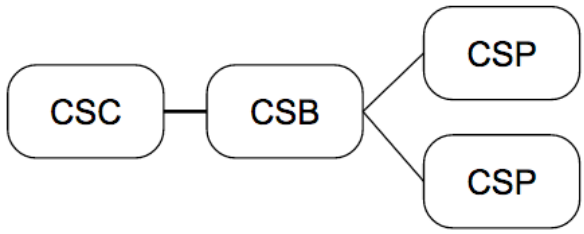
[5] Amenesik SARL, St Pierre Les Nemours, France, e-mail: ijm@amenesik.com

***Abstract*—Cloud Service Broker (CSB) federates multiple Cloud Service Providers (CSP) into a single entity to customers. The benefits that can be felt by Cloud Service Costumer (CSC) are flexibility, ease of use, and reduced cost. However, because of the unique properties and configurations of each CSP, sometime it's not easy to transfer between one CSP to another. Furthermore, the advantage of CSB should be obtained by CSCs in any life cycle of CSC'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 Selecition; Cloud Service Broker; Cloud Service Provider; Cloud Resources;

I. INTRODUCTION

Cloud Service Broker (CSB) act as an intermediary between Cloud Service Costumer (CSC) and Cloud Service Provider (CSP) [1] as described by Picture 1 below. CSB selects the best CSP for CSC according to CSC's requirements and CSP's performance. Among others, the benefits of using a CSB for a CSC are increased flexibility, ease of service, and reduced cost.



Picture 1. Usage Scenario for Cloud Service Brokerage

In term of flexibility, by not contacting multiple CSPs directly, CSC should only create an account in CSP and the respective CSC will be granted an access to multiple CSPs. By using CSB, CSC can avoid to be locked in a single CSP and get the freedom to move between CSP.

CSPs have their own jargons for features in their products that may confuse the potential costumers. Furthermore, the jargons may be a complete opposite with what other providers market. To make it even worse, the specifications of the hardware, software, and network are hard to be compared. CSB tries to provide the easy use of the service by simplifying the resource selection to the costumers.

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

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

It's not really a clear cut process on software or application transfer between CSPs. Each CSPs 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). Therefore, the initial benefit, flexibility, is nullified by CSPs's inflexibility. For a transfer between a software or an application between CSPs to be conducted smoothly, certain requirements are need to be integrated into CSB.

II. REQUIREMENTS

In order for CSB to meet the goals described in previous chapter, there are many requirements which need to be

fulfilled. First, the CSB should actively and periodically measure the computing and network power of available resources in CSPs with a standardized benchmark method, especially for the newly available resources. The benchmark should cover all the metrics that commonly considered by costumers. Furthermore, the benchmark should consist of synthetic benchmark and real world benchmark. This is needed so CSP can objectively compared the power of underlying infrastructure and convincingly recommend certain CSP for a CSC. 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 CSB can run it hundreds instances from different types for few hours in a month and will not break a bank. Furthermore, CSP 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 CSP to gather the most representative and the most accurate performance of the infrastructure.

Another requirement is CSP should know the requirements of CSC's hardware. It can be gathered manually by CSC with a format that is understandable by CSB, or automatically gathered by a monitoring system integrated in CSB. 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 CSP and CSC, CSB 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 CSB.

Once the previous requirement accomplished, another requirement is the adaptability of CSC's application into different CSP. Because if CSB is recommending CSC to migrate to another CSP 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 CSPs. It needs to be an open source or a free platform to ease the propagation to all CSPs. The application of CSC is running on top of this platform thus during migration, CSC doesn't know that the underlying CSP is actually changing. Before migration, CSB should install the platform to target CSP.

III. Design

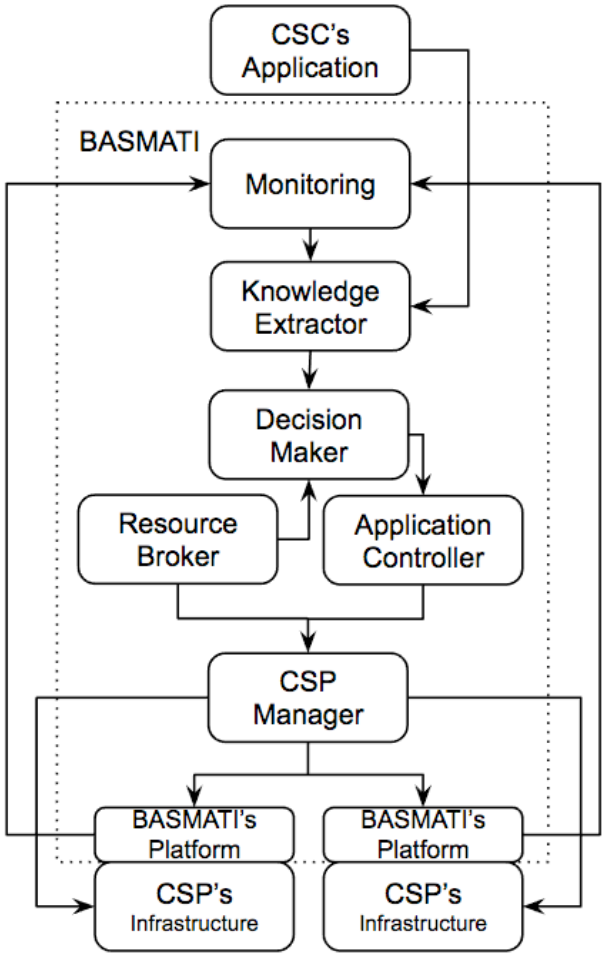
BASMATI – *Cloud Brokerage Across Borders for Mobile Users and Applications* is a joint research project between South Korea and EU Horizon 2020 in developing a federated cloud platform. One of the goals of the project is to support the dynamic resource selection. The architecture and diagram of BASMATI 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. CSC's Application

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

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.



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

C. Knowledge Extractor

KE is analyzing information from both Monitoring and CSC'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 CSC 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 CSP 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 CSP's servers. Therefore, RB periodically contacting CSP Manager to run a virtual machine or a container in CSPs 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 CSP, AC should precisely conduct it in a manner that no downtime is occurring.

G. CSP Manager

CSP Manager has the capability to contact multiple cloud providers through an interface. CSP Manager also responsible to install BASMATI's Platform (BP) in each server of CSPs. The examples of CSP 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 depends on the application, which encapsulates the CSC'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. CSP's Infrastructure

The physical hardware that are running the CSC's applications that belong to CSP. The hardware can be a cloud or edge resource.

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.