BASMATI: An Architecture for Managing Cloud and Edge Resources for Mobile Users

Jörn Altmann[1], Baseem Al-Athwari[1], Emanuele Carlini[2], Massimo Coppola[2], Patrizio Dazzi[2], Ana Juan Ferrer[6], Netsanet Haile[1], Young-Woo Jung[5], Jamie Marshall[4], Enric Pages[6], Evangelos Psomakelis[3], Ganis Zulfa Santoso[5], Konstantinos Tserpes[3 7], and John Violos[3]

[1] College of Engineering, SNU, Rep. of Korea e-mail: jorn.altmann@acm.org, {baseem, netsaneth}@snu.ac.kr
[2] ISTI-CNR, Pisa, Italy, e-mail: {e.carlini, m.coppola, p.dazzi}@isti.cnr.it
[3] ICCS-NTUA, Greece e-mail: {tserpes, psomakelis, violos}@mail.ntua.gr
[4] Amenesik SARL, St Pierre Les Nemours, France, e-mail: ijm@amenesik.com
[5] Cloud Computing Research Department, ETRI, Daejeon, Rep. of Korea, e-mail: {jungyw, ganis}@etri.re.kr
[6] Atos Origin, Barcelona, Spain, e-mail: {ana.juanf, enric.pages}@atos.net
[7] HUA, Athens, Greece, e-mail: tserpes@hua.gr

Abstract. The BASMATI architecture is designed to improve the service quality perceived by end-users. In particular, it focuses on the support of applications that offer services to mobile end-users, ranging from those crossing national borders to those roaming around locally and who both need access to widely dispersed cloud resources. To achieve this, the architecture of BASMATI is built around the concepts of cloud federation and offloading, embodying both heterogeneous resources of different cloud providers and various computational devices located at the edge of the network with different access policies. The BASMATI architecture leverages intelligent decision support for brokering resources, user mobility modeling, a highly reactive management of applications, and a business-oriented cloud federation logic, to drive the efficient and proactive allocation of services onto proper cloud resources. Within this paper, we describe the architectural requirements and the architecture, overview.

Keywords: Cloud Computing, Cloud Federation, Distributed Computing, Quality of Service, Service Level Agreement, Resource Brokering, Mobile Computing, Cloud Architecture Design.

1 Introduction

The cloud computing paradigm is rapidly changing the landscape of information technology. It offers to most end-users and application providers a close approximation of unlimited scalability with virtually zero fixed costs for infrastructure by allowing to rent computing resources and services as pay-per-use [1].

Furthermore, the adoption of the service oriented architecture (SOA) approach for exploiting cloud resources allows easy use of externalized resources and services. In order to safeguard the integrity of the business, cloud service rentals need to be coupled with contracts (i.e., Service Level Agreements), specifying the expected quality of service (QoS). Elastic and dynamic service provisioning also mandates that service level agreements (SLA) are supported by automatic contract re-negotiation, enforcement, and monitoring of the QoS [2].

With the rising of cloud computing technology, the mobile computing paradigm has also grown widespread. Customized apps on end-user's mobile devices are more and more the primary front-end to a plethora of back-end application services hosted on clouds (e.g., storage services, calendar services, email services, social networking services).

While a single marketplace for cloud services can be highly beneficial to both end-users and application service providers, the lack of universal cloud standards and technologies results in poor interoperability between data-centers. It also increases the user lock-in to specific service providers. This collides with the core requirement of mobile cloud applications, i.e. the swift resource relocation within heterogeneous infrastructures.

To address this lack of interoperability, academia and industry have given shape to cloud federations, in which various cloud service providers join their resources to collaboratively increase their market share [3]. Nowadays, the original cloud federation concept is further evolving into more complex, heterogeneous and functionality-rich paradigms [4].

However, existing cloud federations do not realize inter-cloud interoperability. They do not allow heterogeneous services to be represented in a directory of services. They also do not provide a single access point for application service providers to control their application services for supporting the needs of mobile users, who require cross-border access to geographically spread cloud services.

To solve this shortcoming, within the framework of the South-Korea and EU Horizon 2020 joint research project BASMATI – *Cloud Brokerage Across Borders for Mobile Users and Applications*, a federated cloud platform with intelligent decision support for brokering resources has been developed [5]. To this point, the BASMATI federated cloud platform comprises already several optimization algorithms and a cloud federation architecture. The basis for the global design of the BASMATI architecture builds on a wide range of recent projects and technologies, including CompatibleOne [6], OPTIMIS [7], PaaSport, and Broker@Cloud, Easiclouds [8], Contrail [9], and AnyBroker.

The contributions with respect to the cloud federation architecture, which are presented in this article, are: first, a detailed description of the way that the different architecture layers and modules interact with each other; second, intelligent decision support for exploiting cloud services, edge resources, knowledge about users, and application behavior. The contributions allow the platform to forecast the expected load and latencies imposed on specific service instances and allow foreseeing the utilization of allocated resources across multiple clouds.

The remainder of this article is organized as follows: Chapter 2 gives a quick overview about the state-of-the-art in cloud federation architectures. Chapter 3 discusses the assumptions and sets out the requirements for the cloud federation architecture. The BASMATI architecture is introduced in chapter 4. Chapter 5 concludes the paper with brief summary and a discussion.

2 Related Work

2.1 Cloud Federation

Various research works have defined cloud federation. Haile and Altmann (2015) described cloud federation as a strategic alliance between cloud providers, in which cloud providers have reached a cross-site agreement for cooperating regarding the deployment of service components and the use of capacity from each other to cope with demand variations of clients [3]. Another definition is offered by Altmann and Kashief (2014) as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [8].

As part of a cloud federation, even a small service provider can offer a truly global service without spending a dime building new infrastructure. For companies with spare capacity in the data center, the federation also provides a simple way to monetize that capacity by submitting it to the marketplace for other providers to buy, creating an additional source of revenue.

There are immediate benefits for end users, too. The federated cloud means that end users can host apps with their federated cloud provider of choice, instead of choosing from a handful of global cloud providers on the market today and making do with whatever pricing, app support and SLAs they happen to impose. Cloud users can choose a local host with the exact pricing, expertise and support package that fits their need, while still receiving instant access to as much local or global IT resources as theyd like. They get global scalability without restricted choice, and without having to manage multiple providers and invoices. This is an extension of the concept of federation as defined previously (e.g. [10, 11]) and which assumed that the federation should be comprised solely from cloud providers rather than cloud service providers who manage a limited number of cloud resources.

2.2 Existing Cloud Architectures

Among the many cloud architectures that exists, this overview on the state-of-the-art focuses on the ad-hoc mobile cloud platforms. These platforms follow design patterns that are of interesting to BASMATI.

The ad-hoc mobile cloud approach organizes the collective resources of the various mobile devices in the local vicinity, in order to create a virtual-cloud. In such context, a mobile device will use the resources of other close devices (the virtual cloud) instead of its own, in the same way, it would do with a remote datacenter. In principle, this approach can support high user mobility and create virtual clouds on demand according to the necessity. Several ad-hoc mobile cloud platforms try to recreate typical cluster computation (such as MapReduce) in a virtual cloud composed of mobile devices. The approach presented by Huerta-Canepa and Lee realize a Hadoop ([12]) computation on top of a virtual cloud [13]. Hyrax supports a distributed computation based on Hadoop on a virtual cloud as well [14], also including the Hadoop Distributed File System (HDFS) for the storage. Ghasemi-Falavarjani et al. developed a context-aware offloading middleware for mobile cloud (OMMC) to collect contextual information of mobile devices [15]. By considering neighboring mobile devices as service providers, they investigated the resource allocation problem to select service providers that minimizes the completion time of the offloading along maximizing lifetime of mobile devices satisfying deadline constraint [15]. Pu et al. proposed device-to-device (D2D) Fogging framework for mobile task offloading based on network-assisted D2D collaboration, where mobile users can dynamically and beneficially share the computation and communication resources among each other via the control assistance by the network operators. The purpose of their D2D Fogging is to achieve energy efficient task executions for network wide users [16].

3 Basic Assumptions and Challenges

The architecture of the BASMATI platform makes a few assumptions with respect to the type of the applications, the off-loading, and quality of service management. Based on those assumptions, the requirements for BASMATI architecture are derived.

A mobile application is typically split into two parts, the front-end (FE), which is executed on a mobile device, and the back-end (BE), which is a composition of services deployed in the Cloud. It is assumed that it is mainly the BE that poses the hardest limitations on the mobile app in terms of performance. A basic tool to tackle those limitations and ensure that the application will perform according to its promised levels is offloading, i.e. the delegation of some computational task from a resource to another with the purpose to optimize the overall system operation. Commonly, this delegation is directed from the core of the computing infrastructure (i.e. cloud) to the *edge* of the network hierarchy and is supported by dedicated, low power devices such as Raspberry Pi and Banana Pi. This sort of service shuffling across clouds, is in some cases is also called *Cloudbursting*.

The key to performing efficient offloading of this sort, is to carefully monitor the resource utilization against its promised thresholds and apply optimization techniques that consider the state of the whole system rather than a part of it, including the application context. As such, apart from SLA objectives, violation history [17, 18] and resource availability BASMATI seeks for correlations between user characteristics -including mobility patterns- and resource utilization.

By ingesting those factors (resource requirements, availability and application context) in a single complex, performance analysis module, BASMATI achieves to intelligently offload tasks by leveraging on the edge infrastructure's characteristics.

The BASMATI architecture provides specific support for offloading by providing a combined brokering support that encompasses Clouds and Edges. The optimization function of the cloud brokering support for the offloading can deal with services and resources from very distinct Clouds with different cost, federation memberships, SLA metrics, and types of network constraints (bandwidth, latency, access policies). The optimization function of the brokering support can consider the proximity of appropriate edge and Cloudlet resources for offloading.

In order to support a seamless mobile experience, the BASMATI architecture not only needs to address SLAs but also targets Quality of Experience (QoE) for the end-user. QoE perception is linked to parameters akin to both latencies (network, back-end server startup) and bandwidths (network and compute bandwidth, and access to Cloud-stored content). Based on a suitable metrics of user QoE, BASMATI can provide resources that match the user needs in term of QoE (e.g., allocate additional servers to services when needed). It can be performed by anticipating users' behavior in the short term (e.g., by tracking and analyzing aggregated user activities). This task of the BASMATI architecture is performed by modules for analyzing the patterns of user mobility and by exploit models for analyzing the application behavior with respect to its environment.

4 The BASMATI Architecture

The platform governing the BASMATI ecosystem is organized according to a layered architecture composed of three layers (Figure 1). The lowest layer is the *Providers Management Layer* for managing cloud providers and for edge providers. The middle layer is the *Federation Management Layer*, which provides to the federation of cloud providers. The upper layer is the *Application Management Layer* for managing application back-end and application front-end. Each of the these three layers is comprised of a number of modules.

4.1 Providers Management Layer

The providers management (PM) layer deals with the resources that are offered by providers in the context of the federation and can be exploited by the application. It provides an abstraction for the different kinds of resources involved in BASMATI. For example, the two main resource types (i.e., cloud resources and edge resources) do not require different types of mechanisms for resource accounting, user identity management, and for the actual deployment of applications onto the provider resources. However, the restrictions that come with the low-capacity of edge resources.

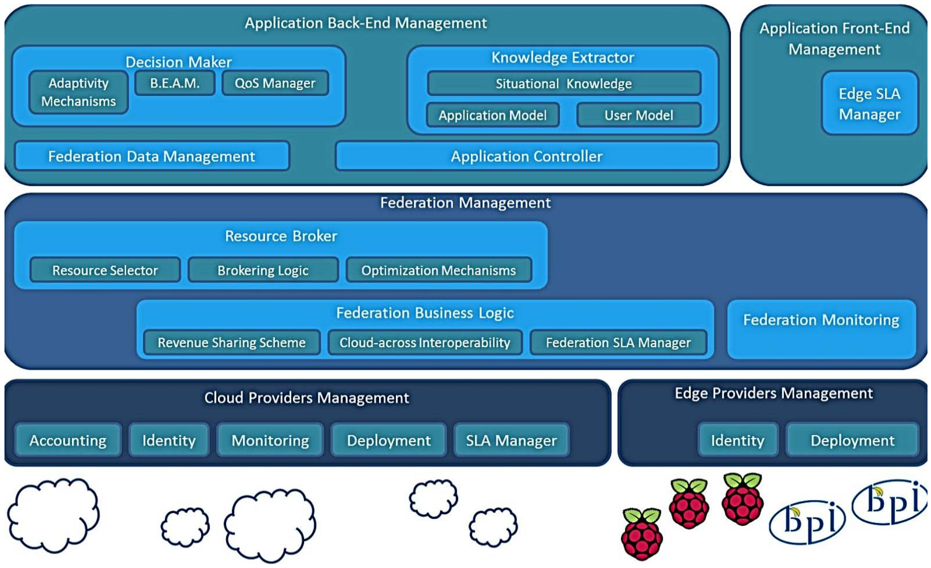

Fig. 1. BASMATI Architecture

Cloud Providers Management The cloud providers management (CPM) includes all those mechanisms and components aimed at easing the exploitation of distinct cloud providers of the cloud federations. Adapters, stubs or partial re-implementation of common services depend on the API provided by the underlying cloud providers.

Identity. Each partner member of the federation exposes the identity API, which allows other members of the federation to make use of the offered services so long as the terms of the federation agreement governing the offer is respected. This is an essential part of the federation, as it is the backbone for the resource and service pricing schemes. It will be used for security, the identification of the financial transactions required for invoicing, federation member management, and for revenue and cost sharing across the cloud federation.

Accounting. The accounting module records all transaction using the information from the identity module and the monitoring module. It provides the raw resource usage data needed for charging according to the service level agreement.

Deployment. After joining a cloud federation, cloud providers expose deployment APIs for the use by customers and other members of the federation. Cloud providers also expose APIs for monitoring the deployed services. From an operative point of view, the deployment module accesses the services of the cloud provider.

Monitoring. The monitoring module collects monitoring data from the cloud provider and aggregates and filters this information before forwarding the information to modules of the federation management layer, including the federated SLA manager and accounting.

SLA Manager. The SLA manager organizes the SLA templates offered by cloud providers and maintains a list of SLAs that have been established between customers (or cloud federation) with a cloud provider. It also checks whether the quality of service satisfy the established SLAs and reports this violation to the application controller.

Edge Providers Management (EPM) The edge providers management (EPM) comprises modules that address the restrictions that come with the use of edge resources. Those restrictions comprise many different ownership of edge resources, large number of providers of edge resources, and restricted functionality of resources. Besides the aspects that are considered for cloud resources, some additional aspects of edge resources require attention. These additional aspects are related to identity and deployment.

Identity. With respect to identity, the large number of providers of edge resources mandates for a possibly decentralized implementation of identity services. It would allow not only to deal with the large number of providers of edge resources but also with the mobility of users as different locations offer different edge resources.

Deployment. The deployment service will support front-end-offloading, which is a variation of computation offloading and requires different protocols than those of the CPM deployment.

4.2 Federation Management (FM) Layer

FM layer is aimed at providing the mechanisms for achieving an efficient mapping of applications onto the clouds of providers participating in the BASMATI federation. This layer comprises three modules: *federation business logic, federation monitoring, and resource broker.*

Federation Business Logic This module embodies the logic and mechanisms to face three key duties: *(i)* enabling the effective deployment and execution of applications across different cloud providers; a feature of paramount importance in the context of BASMATI, in which final users are expected to be highly nomadic, thus accessing cloud resources from many different locations; *(ii)* orchestrating and managing the SLAs of applications for the federation; *(iii)* setting proper revenue sharing schema, defined and implemented within the federation and implemented.

Federation Monitoring The federation monitoring acts as the collector of data sources from various resources belonging to the federation and applications. To realize that, the federation monitoring (1) has a tight relationship with the monitoring components at the cloud providers management level and (2) needs to coordinate and integrate seamlessly with applications running in the BASMATI environment.

Resource Broker The resource broker role in the BASMATI architecture is to provide tools and mechanisms for selecting resources fulfilling the minimum performance requirements of applications (e.g., capacity constraints, availability of resources). To this end, it exploits mechanisms for heuristically provide a range of resources with different values of attributes. For this, the brokering logic organizes the indexing and retrieval of the list of cloud resources available for placement and offloading.

4.3 Application Management (AM) Layer

The AM layer is the architectural layer devoted to the management of BASMATI applications. It is composed of two parts: application back-end management and application front-end management. The former dealing with the governance of the cloud-side subset of applications, whereas the latter focuses on the front-end of applications (e.g., smart-phone apps), which run on mobile devices. The AM layer comprises 7 modules.

Federation Data Management This modules provides the mechanisms enabling the efficient management of application-related data. In addition, the mechanisms are able to work both with structured and unstructured data sources, while providing access to the information to the application as well to the decision maker.

Knowledge Extractor One of the key features of the BASMATI approach are the advanced solutions for the characterization of the behavior of applications, the mobility of users, and the combination of such analyses to generate a more complex situational knowledge (i.e., the information associated to a certain kind of user when interacting with a given type of application). The models built by Knowledge Extractor are exploited by the decision maker to drive the mapping of application onto resources.

Decision Maker In order to perform a proper allocation of BASMATI applications into the resources belonging to the cloud federation, BASMATI relies on the interplay of three modules: decision maker, application controller, and resource broker. The decision maker is the component analyzing the requirements of applications, represented by means of the BASMATI Enhanced Application Model (BEAM), a TOSCA[8] dialect. BEAM describes the structure of applications along with (non) functional requirements. The information provided by the

[8] https://www.oasis-open.org/committees/tosca/

knowledge extractor can be used to drive the eventual composition and replication of the modules of the applications. By interacting with the resource broker about the available resources (requiring the minimum requirements), a ranked list of deployment plans are derived and eventually passed to the Application Controller. To this end, the decision maker exploits optimization mechanisms that consider feedback on past allocations to perform resource classification and prediction on the behavior of applications.

Application Controller The application controller, which performs the actual deployment, takes the deployment plans provided by the decision maker and uses one of those for the actual deployment and keeps the remaining one for alternative deployments that might be needed in case of a SLA violation.

Edge SLA Manager The Edge SLA Manager is the module, running on mobile devices, aimed at identifying potential edge devices, with which service level agreements could be established for supporting the application. This is the module driving workload offloading from smart-phones and tablets to the devices located at the edge of federation.

5 Conclusion

Our paper reflects the architecture design of the BASMATI multi-cloud brokering platform and a descriptions of its modules. It highlights the emphasis on the challenges infused by the nature of mobile cloud services but also by the cloud marketplace as it is shaped in EU and Korea, which is dominated by small cloud service providers who manage a limited amount of cloud resources to support applications. The same market is also characterized by the lack of cloud vendors.

The paper addresses all the issues that stem from the requirements analysis based on the abovementioned characteristics. The emphasis is put on the support of cloud service providers' federations and task offloading towards the edge of the computing and network infrastructure. It remains to validate the platform using real use cases and a reference implementation in order to check the platform interplay with advanced mechanisms on (1) user and application behavior and on (2) offloading.

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 programme 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. Rohitratana, J., Altmann, J.: Impact of pricing schemes on a market for software-as-a-service and perpetual software. Future Generation Computer Systems **28**(8) (2012) 1328–1339
2. Breskovic, I., Altmann, J., Brandic, I.: Cost model based service placement in federated hybrid clouds. Future Generation Computer Systems **29**(4) (2014) 1000–1011
3. Haile, N., Altmann, J.: Risk-benefit-mediated impact of determinants on the adoption of cloud federation. In: 19th Pacific Asia Conference on Information Systems (PACIS 2015), AIS (2015)
4. Carlini, E., Dazzi, P., Mordacchini, M.: A holistic approach for high-level programming of next-generation data-intensive applications targeting distributed heterogeneous computing environment. Procedia Computer Science **97** (2016) 131–134
5. Altmann, J., Carlini, E., Coppola, M., Dazzi, P., Ferrer, A.J., Haile, N., Jung, Y.W., Kang, D.J., Marshall, I.J., Tserpes, K., Varvarigou, T.: A Brokerage Architecture on Federated Clouds for Mobile Applications. In M. Bubak, M. Turaa, K.W., ed.: CGW16 Proceedings, ACC CYFRONET AGH, Krakw (2016) 59–60
6. Yangui, S., Marshall, I.J., Laisne, J.P., Tata, S.: Compatibleone: The open source cloud broker. Journal of Grid Computing **12**(1) (2014) 93–109
7. Ferrer, A.J., Hernndez, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., Sirvent, R., Guitart, J., Badia, R.M., Djemame, K., Ziegler, W., Dimitrakos, T., Nair, S.K., Kousiouris, G., Konstanteli, K., Varvarigou, T., Hudzia, B., Kipp, A., Wesner, S., Corrales, M., Forg, N., Sharif, T., Sheridan, C.: Optimis: A holistic approach to cloud service provisioning. Future Generation Computer Systems **28**(1) (2012) 66 – 77
8. Altmann, J., Kashef, M.M.: Cost model based service placement in federated hybrid clouds. Future Generation Computer Systems **41** (2014) 79–90
9. Carlini, E., Coppola, M., Dazzi, P., Ricci, L., Righetti, G.: Cloud federations in contrail. In: European Conference on Parallel Processing, Springer Berlin Heidelberg (2011) 159–168
10. Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I.M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., et al.: The reservoir model and architecture for open federated cloud computing. IBM Journal of Research and Development **53**(4) (2009) 4–1
11. Celesti, A., Tusa, F., Villari, M., Puliafito, A.: How to enhance cloud architectures to enable cross-federation. In: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, IEEE (2010) 337–345
12. Shvachko, K., Kuang, H., Radia, S., Chansler, R.: The hadoop distributed file system. In: 26th symposium on mass storage systems and technologies (MSST), IEEE (2010) 1–10
13. Huerta-Canepa, G., Lee, D.: A virtual cloud computing provider for mobile devices. In: 1st ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond, ACM (2010)
14. Marinelli, E.E.: Hyrax: cloud computing on mobile devices using mapreduce. In: No. CMU-CS-09-164, Carnegie-Mellon University Pittsburgh PA School of Computer Science (2009) 1–10
15. Ghasemi-Falavarjani, S., Nematbakhsh, M., Ghahfarokhi, B.S.: Context-aware multi-objective resource allocation in mobile cloud. Computers and Electrical Engineering **44** (2015) 218–240

16. Pu, L., Chen, X., Xu, J., Fu, X.: 2d fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted d2d collaboration. Journal on Selected Areas in Communications **34**(12) (2016) 3887–3901
17. Tserpes, K., Aisopos, F., Kyriazis, D., Varvarigou, T.: A recommender mechanism for service selection in service-oriented environments. Future Generation Computer Systems **28**(8) (2012) 1285–1294
18. Tserpes, K., Aisopos, F., Kyriazis, D., Varvarigou, T.: Service selection decision support in the internet of services. In: Proceedings of the 7th International Conference on Economics of Grids, Clouds, Systems, and Services. GECON'10, Berlin, Heidelberg, Springer-Verlag (2010) 16–33