POLAr: Geographic Placement Optimization for Latency Sensitive Applications

Vinicius Monteiro de Lira
Institute of Information Science and Technologies (ISTI)
National Research Council (CNR)
Pisa, Italy
vinicius.monteirodelira@isti.cnr.it

Emanuele Carlini
Institute of Information Science and Technologies (ISTI)
National Research Council (CNR)
Pisa, Italy
emanuele.carlini@isti.cnr.it

Patrizio Dazzi
Institute of Information Science and Technologies (ISTI)
National Research Council (CNR)
Pisa, Italy
patrizio.dazzi@isti.cnr.it

***Abstract*—To assure a timely fruition of media and interactive applications to end users is a complex challenge, especially when potentially spread worldwide, at home or in mobility. It in fact requires a careful placement of the software services on the right computational resources, such that those services are placed as close as possible to end users to mitigate the effect of network on the user experience. In this demo paper, we present a tool that aims to facilitate the placement of latency sensitive applications on computational resources, by considering the geographical positioning of the user demand, the user experience, and the budget limitation of application owners.**

I. Introduction

By means of the recent advancements in service computing, we assisted to a paradigm switch for a large number of services and applications, leading to a innovative rethink of many existing commercial products, including latency-aware and interactive services. Nowadays, many companies are moving their interactive and media products from airwaves and satellite connections to the Internet. It is now possible for people to consume interactive contents and watch events (music, sport, exhibitions, etc.) live from home or in mobility on their smartphone, even when traditional broadcast channels of their own country are not interested in covering such entertainment products.

As matter of facts, to ensure a proper Quality of Experience (QoE) to largely attended events can be challenging from the perspective of the computational infrastructure delivering the content to end users. An insufficient amount of resources can lead to bottleneck, but also a sufficient - but badly placed - amount of resources can lead to unacceptable latency. In fact, it is of paramount importance to place content delivery services as close as possible to the end users, and therefore it is necessary a careful design of the deployed infrastructure, which should take into account the location of the end-users to properly estimate the computational and network footprint generated by such users. This demo paper presents POLAr, a tool that facilitates the placement of content streaming application services over public and private cloud computing resources. POLAr focuses on matching the user demands as much as possible, while ensuring an acceptable QoE within a given budget.

The authors acknowledge the support of project H2020- 723131 BASMATI: Cloud Brokerage Across Borders for Mobile Users and Applications.

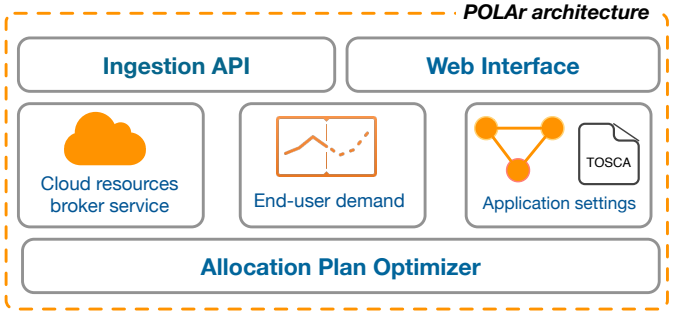

Fig. 1. The POLAr architecture

II. System Architecture

The modular architecture of POLAr is presented in Figure 1. The architecture loosely represents the one that has been designed and developed during the BASMATI H2020 project [1], in which authors were involved. POLAr is composed by the following components: (i) A *cloud resources broker*, which represents a broker service that interfaces to multiple differing cloud providers. This component provides a list of available cloud resources. For each cloud resource, POLAr keeps the static properties (i.e. the instance types specifying RAM, CPUs, storage capacity, etc), the cost (per hour) and the geographic location of the resource. (ii) An *end-user demand database*, which stores the spatial distribution of the end users interested in the specific content delivered by the application. The geographical scale can go from local (e.g. a city) to global (e.g. worldwide). This spatial distribution can come from multiple sources, such as: market analysis or prior knowledge of the application usage through monitoring tools; (iii) An *application setting*, which is a set of TOSCA[1] files that describes the application in terms of services and connection among them. Thus, in POLAr, for each service composing the application, there is a corresponding specification of the non

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

functional requirements, the expected QoE[2], and the amount of end users that can be supported concurrently with a given computational resource. For finding a more refined approximation of these two latter settings might involve the use of extensive benchmarking of cloud resources [2] and the application of machine learning techniques [3]. (iv) An *ingestion API*, which is a REST Full API that can be used by the application owner or digital agents to ingest content into the latter two components; (v) An *allocation plan optimizer*, that exploits a genetic meta heuristic (similar to the one presented in [4], with the addition of the geographically distributed user demand) to select the cloud resources, among the available ones, where to place the application service. It also computes, for each selected resource, the number of instances needed to support the expected QoE. This component solves a multi-objective optimization problem, that maximizes the number of customer with the latency under a given threshold while respecting the given budget. For the optimization task, this components takes as input the estimated spatial distribution of the end users, the list of available service providers, the application settings and a given budget. The budget is informed through the *web interface* which is better discussed in Section III.

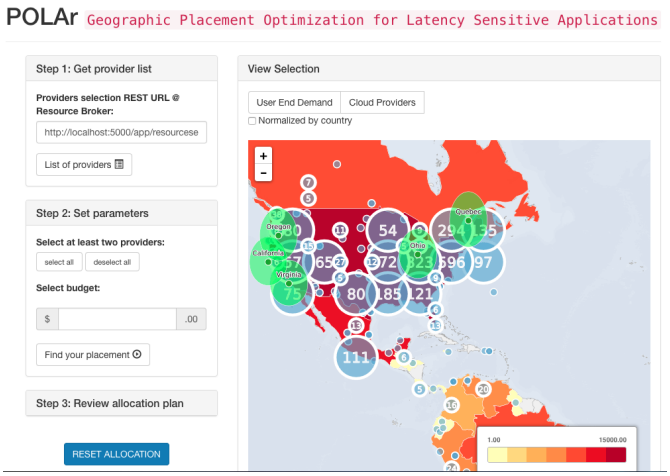

Fig. 2. POLAr web interface.

III. Demonstration

POLAr offers a web interface with an interactive map from which application owners can visualize the end user demand and the locations of the available cloud providers to host the application. The application owner also uses the web interface to request an optimized geographic placement plan. For this task, a budget needs to be informed. Once an optimization is requested, the list of cloud providers selected to compose the optimized plan is shown on the map and highlighted in the web interface. For each selected cloud provider, POLAr details the cost, the specification of the virtual machines and the number of instances to be allocated. POLAr also provides an estimation of the coverage considering the resources allocated to support the application requirements and the end user demand.

End User demand visulization. The application demand is represented by the number of users that are interested in the application. Through the interactive map, the application owner can see the location of them by clicking on the button "User demand". As illustrated Figure 2, the blue circles indicate the location and the number of users. For a global perspective, the application owner can also visualize such information normalized by country clicking on the checkbox "Normalize by countries". A REST API can be used to ingest the user demand of application into POLAr. The only requirement is that the input data must contain a geospatial value indicating the location of the users.

Cloud providers retrieval. The list of cloud providers available in POLAr can be visualized on the map. As shown in the Figure 2, using the interactive map, the application owner can also see more detailed information by clicking over the green points that representing cloud providers. The current list of providers is retrieved from the Amenesik Enterprise Cloud[3] through an OCCI [4] interface that allows direct and authenticated access to the Amenesik Cloud Engine. By using standard interfaces, other cloud providers directories can be easily integrated in POLAr.

Geographic Placement Optimization. The application owner can request an optimized placement plan by informing a maximum budget and clicking on the button "Find your placement". POLAr returns the list of cloud providers selected to allocate the application. More information can be visualized by navigating through the list of selected providers. Furthermore, POLAr shows a summarized information specifying the total cost of the plan (always lower than the budget specified), the number of virtual machines to be allocated and an estimated coverage considering the user demand. To execute this operation, it is necessary that both the application settings and the user demand have been previously configured. If the application owner is not satisfied with the generated placement, it is possible to request a different one by tweaking the budget and/or manually narrow down the list of available providers.

[2]In POLAr the QoE is the maximum acceptable latency for the end users

[3]http://www.amenesik.com/

[4]Open Cloud Computing Interface. More details in http://occi-wg.org/

References

[1] E. Carlini, M. Coppola, P. Dazzi, K. Tserpes, J. Violos, Y.-W. Jung, G. Z. Santoso, J. Altmann, J. Marshall, E. Pages, et al., Basmati: Cloud brokerage across borders for mobile users and applications, in: European Conference on Service-Oriented and Cloud Computing, Springer, 2017, pp. 181–186.

[2] M. B. Chhetri, S. Chichin, Q. B. Vo, R. Kowalczyk, Smart cloudbencha framework for evaluating cloud infrastructure performance, Information Systems Frontiers 18 (3) (2016) 413–428.

[3] J. Violos, V. M. de Lira, P. Dazzi, J. Altmann, B. Al-Athwari, A. Schwichtenberg, Y.-W. Jung, T. Varvarigou, K. Tserpes, User behavior and application modeling in decentralized edge cloud infrastructures, in: International Conference on the Economics of Grids, Clouds, Systems, and Services, Springer, 2017, pp. 193–203.

[4] G. F. Anastasi, E. Carlini, M. Coppola, P. Dazzi, Qos-aware genetic cloud brokering, Future Generation Computer Systems 75 (2017) 1–13.