Towards Cloud-Based Architectures for Robotic Applications Provisioning

Fatima Zahra Errounda[1], Fatna Belqasmi[2], Roch Glitho[3]
Concordia Institute of Information System Engineering (CIISE)
Concordia University, Montreal, Canada
[1]f_errou@encs.concordia.ca, [2]fbelqasmi@alumni.concordia.ca, [3]glitho@ciise.concordia.ca

***Abstract*—Robotic applications are widely used in various domains (e.g. healthcare, agriculture). However, provisioning them in a cost-efficient manner remains an uphill task. Cloud computing is a new paradigm with three key facets: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Rapid application development/deployment, pay-per-use and efficient use of resources are among the expected benefits. Cloud computing is a promising technology for application provisioning and can bring robotic application provisioning to the next level. This paper identifies the shortcomings of the state of the art in cloud-based architectures for robotic applications provisioning. It sketches an overall business model to tackle the identified shortcomings. It proposes an overlay-based architecture to handle the cloud interactions aspects of the proposed business model. The implementation aspects of the overlay-based architecture are discussed. Research directions are also identified.**

Keywords—cloud computing, robotic applications, overlay network, machine-to-machine communication

I. Introduction

Robots are present in a plethora of application domains (e.g. healthcare, agriculture). Unfortunately, provisioning (i.e. developing, deploying and managing) robotic applications in a cost efficient manner remains very challenging. Due to the lack of third party application re-use, the robotic application development process remains slow, as it is usually done from scratch.

Cloud computing offers three types of provision capability: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [1]. In the IaaS, the providers offer computing resources such as servers, routers and power management. The PaaS offers a framework for the development, management and deployment of applications that execute on the IaaS. The SaaS contains the applications that are hosted in the clouds.

Cloud computing enables rapid application development, deployment, and efficient use of resources. These benefits make cloud computing an ideal candidate for application provisioning in robotics. The state of the art in cloud-based robotic application provisioning can be grouped under two categories: robotic applications that are offered as cloud services (e.g., [2] and [3]); and robotic applications that re-use third party applications (e.g. [4] and [5]). However, issues such as the provisioning of a framework that allows third party applications' publication, discovery and execution, and the inter-cloud interactions are still open.

In this paper, we sketch a business model for cloud-based robotic application provisioning then we describe a new overlay-based architecture to solve the associated cloud interaction issues. The rest of the paper is organized as follows. In Section II, we present an illustrative scenario and the requirements for the cloud-based architecture. In Section III, we sketch the overall business model we envision. Section IV presents the architecture we propose to tackle the cloud interaction issues of the sketched business model. We conclude with research directions.

II. Scenario and Requirements

A. *Wild fire Suppression Scenario*

Let us consider a wildfire suppression robotic application that detects and suppresses wildfires using a fleet of heterogeneous robots that can be aerial or ground. Aerial robots apply retardants to reduce fire intensity and spread rate. The ground robots suppress fires using water and foam. The robotic application can be accessed from anywhere, anytime, using a computer, a smartphone or a tablet. If a fire is detected, the application evaluates its intensity and spread rate, and deploys the appropriate robots to extinguish it.

The wildfire suppression application re-uses a fire detection application and a supply management application. The fire detection application utilizes a network of sensors that are deployed in the forest to gather information about wind speed, gas concentration, moisture and temperature. The supply management application provides the firefighter robots with the adequate water and foam supply stations depending on the distance, the type of supply and its accessibility.

B. *Requirements*

We divide the requirements into three groups:

1) General requirements

The first requirement is the possibility for robotic applications to discover, activate and execute third party applications. The second requirement is the possibility for third party applications to publish their services for robotic applications to use them. The third requirement is the possibility to expose the third party applications in a way that novice and experienced robotic application developers can easily re-use them while composing applications.

2) Requirements on the robotic applications

The first requirement is the need of a lightweight standard technology at the interfaces that are offered to end-users and other applications. The second requirement is the possibility to access the robotic applications from anywhere and anytime. The third requirement is the efficient use of the resources on which the applications rely.

3) Requirements on the robotic application provisioning

The first requirement is the necessity of loose coupling between the robotic applications and the third party applications. The second requirement is the possibility for the third party application providers to offer their applications through a standard unified interface technology. Another requirement is rapid application development and deployment by novice application developers.

III. A New Business Model

In this section we propose a business model that can be used as a starting point for further research. We start with the assumptions and principles then the business model.

A. Assumptions and Principles

The key assumption is that robotic applications as well as third party applications are offered in clouds as SaaS. Third party application clouds may provide the three layers of provisioning (i.e. IaaS, PaaS, and SaaS) or only the SaaS layer.

The first principle is that robotic applications are offered as SaaS. The second principle is that these applications are developed using a PaaS that takes into account the peculiarities of the robotic applications (e.g., the availability of commonly used building blocks). The third principle is that robotic applications execute over an IaaS that uses robot virtualization to optimize resource usage. Virtualization is a key technology of cloud computing. It refers to resources abstraction thereby enabling efficiency by sharing these resources. A distinctive aspect of robotic cloud is the need of robots' locations to be factored in when sharing robots among different applications. This is fundamentally different from sharing other types of resources (e.g., computing and storage resources). For the sake of simplicity we assume that the three layers are deployed at each robotic application cloud.

B. The Business Model

We induce from the assumptions and principles a business model with four actors: the end-user, two cloud providers (i.e. the robotic application provider and the third party application provider), and the interaction framework provider. The robotic provider offers robotic applications as SaaS to end-users. The third party application provider offers third party applications as SaaS. The robotic provider may re-use these applications during the provisioning process. The interaction framework provider offers a platform that enables the interactions between the cloud providers for discovery, activation and execution.

We will illustrate these interactions using the wildfire suppression scenario. We have one robotic application provider (i.e. the wildfire suppression robotic provider), two third party application providers (i.e. the fire detection provider and the supply management provider), and the interaction framework provider. The fire detection and the supply management applications' providers connect to the interaction framework to publish their applications. The wildfire suppression provider also connects to the interaction framework. The order in which the three application providers connect to the framework is not interdependent. Then, the wildfire suppression robotic provider discovers the two third party applications and composes them to develop the wildfire suppression application. When this application is executed, it activates and executes the two third party applications.

IV. An Overlay-Based Approach for Tackling the Interaction Issues of the Overall Architecture

Several studies have been undertaken in the area of clouds' interconnection (e.g. [6] and [7]), but none of the proposed solutions meet all of our requirements. None of them offers a publication and discovery mechanism where SaaS-residing applications can be the users. In this section, we start with the architectural assumptions, then we describe the proposed overlay architecture. This is followed by the overlay messages, the overlay nodes architecture, and the procedures.

A. Architectural assumptions

We use peer-to-peer overlays as the cornerstone of the proposed architecture. For the proposed architecture, we assume that each third party application cloud provider owns a repository in which it stores the third party applications it offers. We also assume that each robotic cloud provider owns a repository that holds the robotic applications it offers. The third assumption is that all third party application providers use a common ontology to describe their applications. The goal of this high-level application description is to make application discovery easy for both novice and experienced robotic application providers.

B. The overlay architecture

Fig. 1 shows the interaction overlay network architecture. There are three types of nodes; the ThirdPartyAppsAgent, the RoboticAppsAgent, and the OntologyStore. Each ThirdPartyAppsAgent represents one and only one third party application cloud in the overlay. It activates and executes the third party applications that pertain to the cloud it represents. It communicates with the repository of its corresponding cloud, and with the other nodes in the overlay network.

There is only one RoboticAppsAgent per robotic cloud in the overlay. The RoboticAppsAgent communicates with the robotic applications that reside in its corresponding cloud to

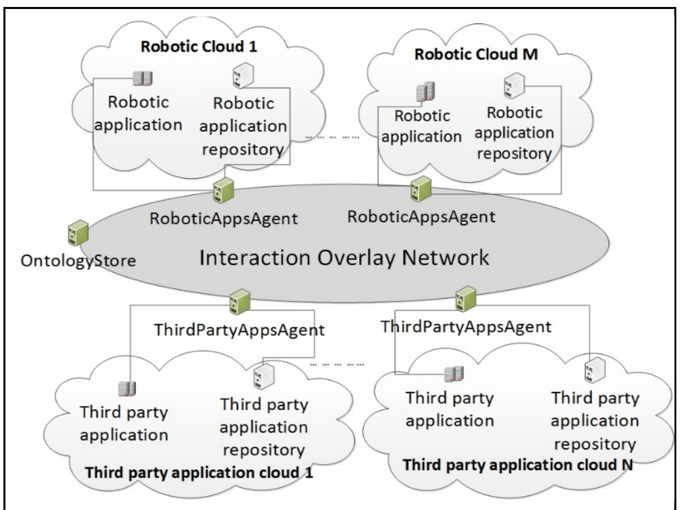

Fig. 1. the overlay structure

manage their third party applications' discovery, activation and execution requests. The RoboticAppsAgent finds and contacts the ThirdPartyAppsAgent which represents the third party applications' cloud that are available in the overlay.

The OntologyStore maintains and provides the ontology, it is managed by the Interaction Overlay Network provider. It communicates with the other nodes in the overlay network to offer the ontology when requested. It may be implemented as a distributed store, but this is out of the scope of this paper.

C. The overlay messages

Table I presents the messages exchanged between the overlay nodes.

TABLE I. Overlay Node Messages

Message type	***Message description***	***Message address***
Publish	Publishes the third party applications' description to the overlay network. Sent by the ThirdPartyAppsAgent.	Multicast
Discover	Finds the third party application requested by the robotic application. Sent by the RoboticAppsAgent to the overlay network.	Multicast
Activate	Activates the specific third party application requested by the robotic application. Sent by the RobotcAppsAgent to the ThirdPartyAppsAgent that represents the specified third party application.	Unicast
Execute	Executes the specific third party application requested by the robotic application. Sent by the RoboticAppsAgent to the ThirdPartyAppsAgent that represents the third party application.	Unicast

D. The overlay nodes' architecture

There are three layers in the overlay nodes' architecture; the service layer, the abstraction layer, and the communication layer. The service layer handles the robotic application requests. It identifies the robotic application request (i.e., third party application's discovery, activation or execution). It communicates with the abstraction layer to look up for the requested third party application. And, it performs the requested action. For the OntologyStore, the service layer handles the requests to get a copy of the ontology.

The ThirdPartyAppsAgent's abstraction layer builds an abstract representation of the third party applications residing in its corresponding cloud using the common ontology. The ThirdPartyAppsAgent's uses the abstraction layer to look up for third party applications. The communication layer enables communication among the different overlay nodes, i.e. between the overlay nodes and the cloud applications, as well as between the overlay nodes and the cloud repositories.

E. Procedures

General procedures: The general procedures include the publication, the discovery, the activation, and the execution procedures. For the sake of simplicity we will only present the discovery procedure.

First, the robotic application sends a third party application's discovery request to the overlay via the RoboticAppsAgent's communication layer. The RoboticAppsAgent extracts the discovery criteria from the received request (e.g. find a third party application of supply-management type). The abstraction layer maps the request to the ontology using the common application ontology and instructs the communication layer module to send a "Discover" message to the other nodes in the overlay. When a ThirdPartyAppsAgent node receives the discovery request, it looks up for the requested third party application in the abstraction layer. If it finds it, it sends feedback to the RoboticAppsAgent that informs the robotic application.

In the wildfire scenario, the ontology contains concepts related to the fire detection and the supply management, such as gas concentration, temperature levels, water and foam supply station location. An example of a discovery request is to look for a fire detection application that relies on gas concentration method.

Overlay management procedures: We can identify the following procedures for the overlay nodes: joining, voluntary leaving, and node failure. Because of space limitations, we will only present the ThirdPartyAppsAgent joining procedure.

To join the overlay network, the ThirdPartyAppsAgent requests the ontology from the OntologyStore. Then, it gets the third party applications' categorization information from the third party application repository. We assume that the third party application cloud provider categorizes the applications offline using the common ontology. After the ThirdPartyAppsAgent receives the categorization information, it stores it at the abstraction layer. Then it publishes the set of third party applications that it represents.

V. Research Directions

Several research items need to be addressed. Such as the application of cloud paradigm to robotic applications at the IaaS, PaaS and SaaS level. The interface technology at the SaaS level must be lightweight enough to support small footprint device access. The PaaS must take into account the distinctive characteristics of robotic applications. Robot virtualization is a key research area in its very early stage.

Other research items are: the design of a well-defined ontology, the integration of different ontologies, and the distributed provisioning of the ontology. In the wildfire scenario, defining an ontology that combines concepts related to fire detection, fire suppression and supply management is a challenge.

References

[1] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, "A break in the clouds " *ACM SIGCOMM Computer Communication Review,* vol. 39, pp. 50, 2008

[2] Y. Chen, Z. Du and M. García-Acosta, "2010 fifth IEEE international symposium on service oriented system engineering; robot as a service in cloud computing" in 2010, pp. 151-158.

[3] R. Doriya, P. Chakraborty and G. C. Nandi, "Robotic services in cloud computing paradigm," in *Cloud and Services Computing (ISCOS), 2012 International Symposium on,* 2012, pp. 80-83.

[4] Lujia Wang, Ming Liu and M. Q. -. Meng, "Towards cloud robotic system: A case study of online co-localization for fair resource competence," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on,* 2012, pp. 2132-2137.

[5] Haiyan Wu, Lei Lou, Chih-Chung Chen, S. Hirche and K. Kuhnlenz, "Cloud-Based Networked Visual Servo Control," *Industrial Electronics, IEEE Transactions on,* vol. 60, pp. 554-566, 2013.

[6] Chen Shan, Chang Heng and Zou Xianjun, "Inter-cloud operations via NGSON," *Communications Magazine, IEEE,* vol. 50, pp. 82-89, 2012.

[7] M. Z. Hasan, M. Morrow, L. Tucker, S. L. D. Gudreddi and S. Figueira, "Seamless cloud abstraction, model and interfaces," in *Kaleidoscope 2011: The Fully Networked Human? - Innovations for Future Networks and Services (K-2011), Proceedings of ITU,* 2011, pp. 1-8.