AN OCCI COMPLIANT INTERFACE FOR IAAS PROVISIONING AND MONITORING

Salvatore Venticinque, Alba Amato and Beniamino Di Martino
Dep. of Information Engineering, Second University of Naples, Aversa, Italy

Keywords: Cloud Agency, Negotiation, Monitoring.

Abstract: In the Cloud scenario provisioning and monitoring play an important role giving the possibility to maintain always the best resources configuration that satisfies the application requirements. Dynamic Cloud provisioning and monitoring allow for the possibility of getting resources in a way that is suited especially well to the business model of IT companies, which can adapt their costs to the current needs continuously and easily. A solution for Cloud resource provisioning and monitoring should also be vendor independent, platform neutral to choose the best proposal among a collection of business offers the widest it is possible. One of the first proposals of standard in Cloud is represented by OCCI (Open Cloud Computing Interface) that is a protocol and API for all kinds of management tasks. In this paper we describe a proposal of extension of OCCI to support provisioning, monitoring and reconfiguration. Furthermore we introduce Cloud Agency, a software platform that complements the common IAAS management facilities with a set of advanced services for dynamic provisioning and monitoring of Cloud resources.

1 INTRODUCTION

Cloud computing represents an opportunity for IT users to reduce costs and increase efficiency providing an alternative way of using IT services. For big companies, which own large physical infrastructures, it allows for high utilization. On the other hand it gives to small and medium enterprises the possibility of using services and technologies that were prerogative of large ones, by paying only for the resources needed and avoiding upfront investment. In fact both costs estimation and provisioning can be delayed . But if there are many advantages for the users, there are also problems that, if not resolved, can vanish the benefits. A well known problem is the *vendor lock-in*, that is the risk for the user of being locked to a provider finding very difficult to migrate its application to other solutions when is not satisfied about the service level or about its cost. This is due to the absence of standards for Cloud solutions that ensure portability among different vendors' technologies. To overcome this problems it is fundamental to have a general purpose set of specifications and facilities for cloud that are vendor independent, platform neutral and can be extended to solve a variety of problems in Cloud computing. One of the first proposal of standard in Cloud is represented by OCCI[1] (Open Cloud Computing Interface) that is a protocol and API for all kinds of management tasks. This solutions is aimed at the fulfillment of three requirements: integration, portability and interoperability for common tasks including deployment, autonomic scaling and monitoring still offering a high degree of extensibility. In this paper we describe a proposal of extension of OCCI to support provisioning, monitoring and reconfiguration. Furthermore we introduce Cloud Agency, a software platform that complements the common IAAS management facilities with a set of advanced services for dynamic provisioning and monitoring of Cloud resources.

2 RELATED WORK

According to (Buyya et al., 2009) a market-oriented resource management is needed in order to regulate the supply and demand of Cloud resources. The current Cloud computing technologies offer a limited support for dynamic negotiation of SLAs between participants. A Cloud multi-agent management architecture is proposed in (Cao et al., 2009; You et al., 2009). Preliminary investigations by the authors on

[1] http://occi-wg.org/

Venticinque S., Amato A. and Di Martino B..
AN OCCI COMPLIANT INTERFACE FOR IAAS PROVISIONING AND MONITORING.
DOI: 10.5220/0003932101630166
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 163-166
ISBN: 978-989-8565-05-1

related topics have been presented in (Aversa et al., 2010). SLA@SOI is the main project that aims (together with other relevant goals) at offering an open source based SLA management framework. It will provide benefits of predictability, transparency and automation in an arbitrary service-oriented infrastructure, being compliant with the OCCI standard. In order to check or guarantee an agreed SLA at IAAS level, it is necessary to monitor performance and quality indexes, to enforce the agreed service terms. Traditional monitoring technologies for single machines or Clusters are restricted to locality and homogeneity of monitored objects and, therefore, cannot be applied in the Cloud in an appropriate manner (Emeakaroha et al., 2011). In (Andrew et al., 2011) an extension of OCCI for IAAS provisioning is described, but it doesn't add new functionalities.

3 OCCI EXTENDED

Obtaining a common interface is the focus of the OCCI group that defined a model for Cloud management at IAAS. OCCI defines entities, API and protocol. The specification of Cloud API is a Resource Oriented Architecture (ROA) that uses REpresentational State Transfer (REST)protocol. The OCCI core meta-model (Metsch et al., 2011) is shown in the left part of the Figure 1. It provides means of handling abstract Cloud resources. Any resource exposed through OCCI is a Resource or a sub-type thereof. A resource can be e.g. a virtual machine, a job in a job submission system, a user, etc. The Resource type is complemented by the Link type which associates one Resource instance with another. Entity is an abstract type, which both Resource and Link inherit. Each sub-type of Entity is identified by a unique Kind instance. The Kind type is the core of the type classification system built into the OCCI Core Model. Kind is a specialization of Category and introduces additional resource capabilities in terms of Actions. An Action represents an invocable operation applicable to a resource instance. The last type defined by the OCCI Core Model is the Mixin type. An instance of Mixin can be associated with a resource instance, i.e. a sub-type of Entity, to ”mixin” additional resource capabilities at run-time. Resources can be managed using a set of operations create, retrieve, update and delete. Currently three types of resources are considered: storage, network and compute resources. It is also possible to manage state (start, stop, restart) and create new resources using actions. In order to extend the OCCI model by autonomic and dynamic services, it is necessary that new functionalities can be deployed in user's Cloud. To be compliant with the standard it is necessary to extend the core model using inheritance of *MIXIN* to provide IAAS resources with monitoring and provisioning capabilities, and *ENTITY*, to represent those new concepts, which are introduced by the services we are going to describe. In Figure 1 the new entities and mixin, an their relationship with the OCCI core model are shown. Among the entities, a Call for Proposal (CFP) describes the list of resources which are necessary to run a Cloud application. A Cloud customer defines the values of those attributes of OCCI Compute, Storage and Links which are significant for his application requirements. A CFP will include also the negotiation rules to select the best offer among the ones proposed by providers. A Proposal is a candidate for a Service Level Agreement(SLA). It includes an instance for each resource in the CFP with all the attribute specified, including other information such as the cost and service levels. It can be accepted or refused by the customer. An SLA is an accepted proposal, it is agreed by the customer with one provider. Among the new mixin, a Vendor is committed to get an offer for resource provisioning, to accept or refuse that offer, to get information about a resource or to perform an action on it against a specific Cloud provider or technology. A Broker is an intermediary that provides information and assists in finding the right cloud-based solution. It receives a CFP, ask to vendors for available offers, brokers the proposal that best serves the user and allows to close the transaction. The Meter is a mixin that monitors some parameters chosen by user and gives the measure of performance indexes for those parameters. It can be deployed into the Cloud, within a Virtual Machines, to dynamically complement the user's Cloud resource with this new capability. We need to run Meter instances on any resource to perform locally specialized algorithm to take measures. An Archiver collects measures from Meters and stores them in a

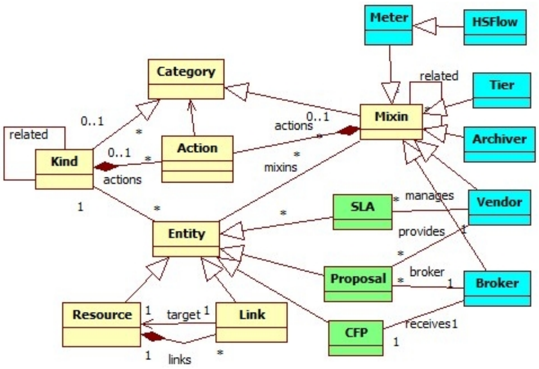

Figure 1: OCCI Extended model.

knowledge base. It provides metrics and performance statistics computed on the knowledge base. A Tier is a mixin that use statistics provided by the Archiver, it detects critical conditions such as SLA violations, resource saturation or underutilization and then informs the user or autonomically performs required reaction as new provisioning or management.

4 CLOUD AGENCY DESIGN

Cloud Agency uses the agents technology to implements the OCCI extended model. Each mixin presented before will be implemented by a software agent. According to our extended OCCI model Cloud Agency is both a mixin itself and a collection of those mixin presented before. It can run anywhere, also in an user's Resource, if it has been implemented as a deployable software component. Cloud Agency can access, on behalf of the user, the utility market of Cloud computing to maintain always the best resources configuration that satisfies the application requirements. This system is being designed and developed within the research activities of the FP7 mOSAIC project. It is in charge to provide the collection of Cloud resources, from different vendors, that continuously meets the requirements of users applications. According to the available offers, it generates a service level agreement that represents the result of resource brokering and booking with available providers. The user is able to delegate to the Agency the monitoring of resource utilization, the necessary checks of the agreement fulfillment and eventually re-negotiations. Cloud Agency will supplement the common management functionalities which are currently provided by IAAS Private and Public infrastructure with new advanced services, by implementing transparent layer to IAAS Private and Public Cloud services. Cloud Agency will support the Cloud user in two different scenarios: **Deployment**, to discover and buy the available resources needed to run Cloud applications; and **Execution**, to monitor and eventually to reconfigure Cloud resources according the changed requirements of the Cloud Application. The following concepts of CA appear in the General Overview of Cloud Agency. A **Call for proposals (CFP)** is a document that define requirements of resource to be acquired, prices and other information relevant for the negotiation. A **Resource List** is the part of the CFP where necessary resources and their attributes are specified. **PolicyLists** are different rules to be used for defining resource brokering strategies that take in input users requirements and providers offers. **Cloud Resources** are the result of the CA Provisioning. They are described, together service levels and other information in a Service Level Agreement (SLA).

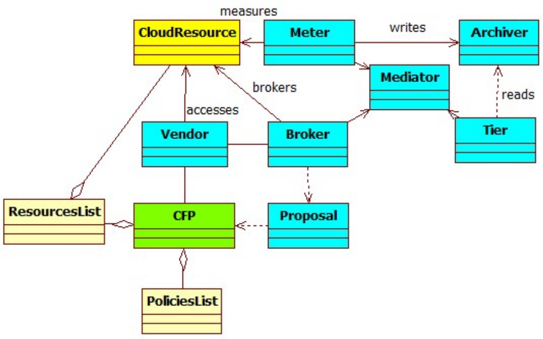

Figure 2: Cloud Agency Conceptual Model.

The blue components of Figure 2 represent agents and provide the necessary facilities to expose Provisioning, Management, Monitoring and Reconfiguration Services. In particular the **Vendor** implements a wrapper for a specific Cloud. It is used to get an offer for resource provisioning, to accept or refuse that offer, to get information about a resource or to perform an action on it (start, stop, resume). **Brokers** receive a CFP, ask to vendors for available offers, broke the best one and allow to close the transaction. **Meters** are deployed on Cloud resources to measure performance indexes. The **Archiver** collects measures from Meters and store them in a knowledge base. It provides metrics and performance figures computed on the knowledge base. **Tiers** use reconfiguration policies to: detect critical events from performance monitoring of Resource; evaluate critical conditions, such as SLA violations, saturation or underutilization; perform required reaction as new provisioning or management. The **Mediator** receives information and configuration requests directly from the user. It is in charge of starting new transaction for provisioning by creating broker agents. It starts also new tiers and meters, and returns information about CA configuration and services. In Figure 3 the Cloud user prepares a CFP to be sent to the Cloud Agency. The Cloud Agency returns a proposal from a Cloud Provider that satisfies those requirements. If the mOSAIC User accepts that Proposal an SLA is agreed and the offered resources have been allocated. After that the User can use the Platform for the application Deployment. The mOSAIC user exploits an Application Tool for resource provisioning to define his requirements at IAAS and to start a transaction. The Mediator receives the CFP, creates a Broker instance that is responsible to handle that CFP. The Broker search for available Vendors and asks for a proposal to each one. It collects the proposals and performs a brokering to choice the best one. In an asynchronous way the best

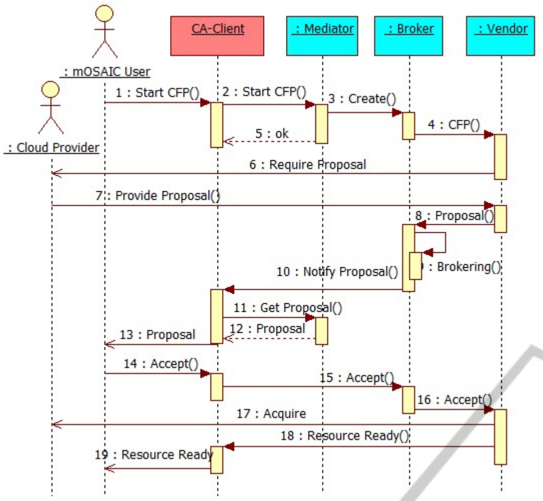

Figure 3: Interaction Model for Provisioning.

proposal is notified to the Application Tool that retrieves the proposal details and waits for the Users agreement. If the user agrees the proposal is amended and the resources are allocated.

Cloud Agency has been implemented as a Multi Agent systems. Execution environment for agents and communication facilities are provided by the Jade agent platform(Bellifemine et al., 2003). Jade has been chosen as a platform to provide an execution environment of software agents, an Agent Communication Channel (ACC) and some protocol implementation to support communication. AMS and DF provide standard services of FIPA compliant agent platforms: a name server for agents and a yellow pages registry for publication and discovery of agent base services. Agents implement the new OCCI MIX-INs defined in section 3. They embed the interaction protocols of the deployment and execution scenarios. Agents will communicate among them via standard ACL (Agent Communication Language) over HTTP, or over other transport protocols if it is necessary.

5 CONCLUSIONS

In this paper we presented a proposal of extension of OCCI to support provisioning, monitoring and reconfiguration. Standardization will allow user to choose in cloud without vendor lock-in and to realize portability giving the possibility to use the cloud services provided by multiple vendors so enjoying the full benefits of cloud computing. In particular here we designed new runtime capability which can be added to the Cloud at IAAS for for autonomic provisioning and monitoring. Finally we introduced the design of Cloud Agency, an agents based software platform the implements the proposed model.

ACKNOWLEDGEMENTS

This work has been supported by the FP7-ICT-2009-5-256910 (mOSAIC) EU project and by the MIUR-PRIN 2008 Cloud@Home: a New Enhanced Computing Paradigm.

REFERENCES

Andrew, E., Thijs, M., and Alexander, P. (2011). Open cloud computing interface in data management-related setups. In Fiore, S. and Aloisio, G., editors, *Grid and Cloud Database Management*, pages 23–48. Springer.

Aversa, R., Di Martino, B., Rak, M., and Venticinque, S. (2010). Cloud agency: A mobile agent based cloud system. In *Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems*, CISIS '10, pages 132–137, Washington, DC, USA. IEEE Computer Society.

Bellifemine, F., Caire, G., Poggi, A., and Rimassa, G. (2003). Jade - a white paper, in exp in search of innovation. *Special Issue on JADE TILAB Journal*.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.*, 25:599–616.

Cao, B.-Q., Li, B., and Xia, Q.-M. (2009). A service-oriented qos-assured and multi-agent cloud computing architecture. In *Proceedings of the 1st International Conference on Cloud Computing*, CloudCom '09, pages 644–649, Berlin, Heidelberg. Springer-Verlag.

Emeakaroha, V. C., Netto, M. A. S., Calheiros, R. N., Brandic, I., Buyya, R., and De Rose, C. A. F. (2011). Towards autonomic detection of SLA violations in cloud infrastructures. *Future Generation Computer Systems*.

Metsch, T., Edmonds, A., and al. (2011). Open cloud computing interface core and models. *Standards Track, no. GFD-R in The Open Grid Forum Document Series*.

You, X., Xu, X., Wan, J., and Yu, D. (2009). Ras-m: Resource allocation strategy based on market mechanism in cloud computing. In *Proceedings of the 2009 Fourth ChinaGrid Annual Conference*, CHINAGRID '09, pages 256–263, Washington, DC, USA. IEEE Computer Society.