NEGOTIATION POLICIES FOR PROVISIONING OF CLOUD RESOURCES

Salvatore Venticinque[1], Viorel Negru[2,3], Victor Ion Munteanu[2,3], Calin Sandru[2,3], Rocco Aversa[1] and Massimiliano Rak[1]

[1]*Second University of Naples, Naples, Italy*

[2]*Research Institute e-Austria (IeAT), Timisoara, Romania*

[3]*Computer Science Department, West University of Timisoara, Timisoara, Romania*

Keywords: Negotiation, Sky computing, Multi-agent system, Negotiation policies, Cloud resources, Quality of services, Service-level agreement, Resource provisioning.

Abstract: Cloud represents today a computing market where users can buy resources according to a *pay-per-use* business model. Cloud providers offer different kind of services which can be characterized by different service levels. Negotiation of the best resource can be very difficult because there is a semantic gap between the different provider SLAs and the requirements of an user's application. We address the negotiation issue within the research activity of the mOSAIC project by designing and developing an agents based service at platform level for provisioning of cloud resources. In order to allow the execution of the application and fulfill the developer's requirements, the service uses a policy based approach that is able to choose, at infrastructure level, the best solution, in terms of a collection of cloud resources from different providers.

1 INTRODUCTION

Cloud Computing has emerged as the reference paradigm for selling or renting computing resources. Let's imagine the Internet as a huge computer market where customers look for, and get any kinds of computer resources. We mean hardware, operating systems, development platforms, application servers or complete applications.

In this context, the problems that this paper addresses are on how to describe one's own requirements, how to buy, pay and get cloud resources, how to check if the resources are compliant with the provider's service level agreement.

The business model of cloud computing is *pay per use*. It means that the business model does not tie customers to one provider and that they could change it when the application requirements change or when a more convenient offer is available.

However this opportunity is limited by the fact that, currently, providers use proprietary technologies which make it impossible for a transparent migration of users' applications and platforms from one cloud solution to another. This is known as *vendor-lock-in* problem. It represents a relevant issue that is adressed by the scientific community and founding calls. In this context, the mOSAIC project (mOSAIC, 2010) proposes a new, enhanced, programming paradigm that is able to exploit the cloud computing features, building applications which are able to adapt themselves as much as possible to the available resources and to acquire new ones when needed. mOSAIC offers together an API for flexible, scalable, fault tolerant, provider independent cloud applications and a framework for enriching them with all needed features by programming.

When interoperability and portability of applications across heterogeneous clouds are supported, autonomous services, which can automatically manage discovery, negotiation and monitoring of cloud resources, represent an added value for users within the sky computing paradigm (Aversa et al., 2011).

This article presents the state of art of research activities on negotiation and complementary services in Cloud. Furthermore it focuses on the agent based solution proposed within the mOSAIC project and provides details about the last design choices and developments. Section 3 covers negotiation as a service. The negotiation module, policies, models and algorithms are discussed in section 4. Related work is dis-

Venticinque S., Negru V., Ion Munteanu V., Sandru C., Aversa R. and Rak M..
NEGOTIATION POLICIES FOR PROVISIONING OF CLOUD RESOURCES.
DOI: 10.5220/0003747003470350
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 347-350
ISBN: 978-989-8425-96-6

cussed in section 5. Finally, we come to conclusions in section 6.

2 CLOUD AGENCY

In the current context of the Cloud market, mOSAIC is expected to offer the freedom of choice at programming level as well as at the resource level. It intends to create and promote an open-source Cloud application programming interface and a platform targeted for developing multi-Cloud oriented applications.

The Cloud Agency represents the central point for resource management in the mOSAIC project. As a multi-agent system, the Cloud Agency has been designed to offer resource negotiation, monitoring and reconfiguration services. These services can be accessed by the user's application either through mOSAIC's runtime (using mOSAIC's SDK) or independently (using API calls). For the purpose of this paper, the focus will be on the negotiation service. The process of resource provisioning involves specification of parameters, gathering of offers and selecting the best ones that suit the application needs. Once the resources have been chosen, the SLA can be finally generated. A long study has been made in order to identify SLAs and QoS requirements that are the inputs of a negotiation module. These requirements are the basis for defining resources and for defining both negotiation policies and for defining monitoring and reconfiguration policies.

3 NEGOTIATION AS A SERVICE

The Cloud Agency's negotiation service is provided as an asynchronous API that is designed around an event driven model. The service calls are divided in Service Requests, Callbacks and Queries. Access is enabled through a HTTP REST interface.

- **Service Requests** are used execute something and generate events within the agency (ex. to start a negotiation, to accept to refuse a SLA, to change a policy, etc). They are asynchronous and the result comes through a CallBack.
- **CallBacks** are used to receive and handle events which follow previous Service Requests.
- **Queries** are used to get information (ex. get the list of vendors or resources). They are synchronous and return immediately the response if it is available, an exception otherwise.

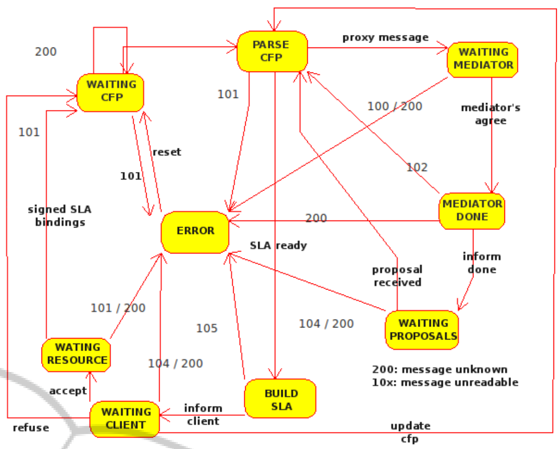

Figure 1: Asynchronous behavior of the Negotiator.

This service is implemented by distributed and specialized agents, which interact using standard protocols. A *Client Agent* acts as access point (proxy between the HTTP REST interface and the internal Agent Communication Language). It receives the user's requests and cooperates with the *Negotiator Agent* in order to broker cloud services and resources. The *Negotiator Agent* handles SLA Template (Call For Proposal) analysis and proposal evaluation. It works with *Mediator Agents* in order to broker each needed resource. The *Mediator Agent* identifies available *Vendor Agents* and contacts each of them for bids on the needed resources. The *Vendor Agents* are wrappers (drivers) that translate the specific provider protocols into the uniform protocol used inside the mOSAIC framework.

The cooperation among agents has been designed adopting well defined Agents Interaction Protocols of the FIPA standard (Grama et al., 2000). The *Client Agent* implements a standard FIPA *Contract Net Interaction Protocol* with the *Negotiator Agent*. It submits a call for proposal using an SLA Template with the application requirements and gets the SLA Proposal. The *Mediator Agent* implements the standard FIPA *Brokering Interaction Protocol*. The subprotocol used to talk with vendors is again the standard FIPA *Contract Net Interaction Protocol*.

The behavior of each agent has been designed as an automata. In Figure 1 the behavior of the *Negotiator Agent* is shown. All the states can be followed (from receiving a Call For Proposal to building a SLA Proposal and, once accepted, booking the resources).

4 NEGOTIATION MODULE: POLICIES, MODELS AND ALGORITHMS

The negotiation model was designed around the cooperation of the agents involved (*Client Agent*, *Negotiator Agent*, *Mediator Agents* and *Vendor Agents*. These agents work together to achieve a first-price sealed-bid auction. The *Vendor Agents* submit bids and the *Negotiator Agent* awards the contract to the lowest price bidder.

Communication needs (with the client and within the agency) led to the adoption of WS-Agreement as a protocol for establishing agreements. The protocol has been used to describe documents ranging from Call for Proposal to SLA Proposal. Each WS-Agreement document has it's terms defined using OCCI resource specification. Once such a resource has been negotiated it will contain additional information offered by the provider (ex. price, QoS, access information).

Having all this information enables possible reasoning about it. It thus makes the creation of rules for filtering for the best candidates or for specific criteria easy.

Negotiation policies are a set of high level governing rules which define assertions or actions to be taken when certain conditions are met. The language chosen for describing these policies is WS-Policy which is an XML-based standard that is integrated within the WS-Agreement. We have extended this language in order to accommodate for our specific goals.

When designing the policies, there were several features sought like the ability to:

- define the interaction with resource parameters and provider parameters;
- control the way matching is performed (exact, approximate etc);
- identify common elements that are relevant to the negotiation process;
- have flexibility when defining policies: in terms of resources, providers, negotiation mechanism, etc;
- be able to combine policies easily (as long as they are not in conflict);

All the policies are defined within a $< wsp : Policy >$ XML tag that is within the WS-Agreement. They are defined as blocks of $< neg : Constraint >$. Each such block is composed of $< neg : Targets >$ and $< neg : Conditions >$. The $< neg : Targets >$ block describes a list of targets that that policy applies to. The $< neg : Conditions >$ block describes the rule conditions. Conditions can apply to any of the resource attributes both defined through the OCCI specifications and those added by the provider for QoS and pricing. If the targets satisfy the conditions then the whole $< neg : Constraint >$ block is validated. If all $< neg : Constraint >$ blocks are validated then the chosen resources are valid for building the SLA Proposal.

```
<neg:Constraint>
  <neg:Targets>
    <wsp:All>
      <neg:Target>
        <neg:Resource>mod:Compute</neg:Resource>
      </neg:Target>
    </wsp:All>
  </neg:Targets>
  <neg:Conditions>
    <wsp:ExactlyOne>
      <neg:Condition>
        <neg:Issue>vendor</neg:Issue>
        <neg:Operator>=</neg:Operator>
        <neg:Value>Google</neg:Value>
      </neg:Condition>
      <neg:Condition>
        <neg:Issue>vendor</neg:Issue>
        <neg:Operator>=</neg:Operator>
        <neg:Value>Microsoft</neg:Value>
      </neg:Condition>
    </wsp:ExactlyOne>
  </neg:Conditions>
</neg:Constraint>
```

Figure 2: Policy example.

Figure 2 expresses the flexibility of the policy system. The figure describes a policy that targets compute elements and forces their vendor to be either Google or Microsoft.

5 RELATED WORK

Research into cloud market-oriented resource management and QoS-based resource allocation mechanisms has been done by (Buyya et al., 2009). An attempt to define several QoS metrics is presented in (Cao et al., 2009): response time, availability, reliability, cost and reputation are considered. A reference of SLA model is provided in (Sim, 2010) where SLA objectives (SLOs) are used to compose an SLA. A number of service levels and performance metrics for each resource results in multiple SLOs for every service. The work presented in (Kertesz et al., 2009) represents a first proposal to combine SLA-based resource negotiations with virtualized resources in terms of on-demand service provision.

Cloud multi-agent management architectures have been proposed in (Cao et al., 2009) and (You et al., 2011).The implementation that was presented in (Sim, 2010) is very interesting because it supports dynamic negotiations.

Approaches to policy based automatic SLA negotiation have been made by (Zulkernine and Martin, 2011) for Web Services and by (Xiao and Cao, 2010) for Internet-based Virtual Computing Environments.

Preliminary investigations by the authors on related topics have been presented in (Venticinque et al., 2011) and (Aversa et al., 2010). Here we describe last advances within the scheduled activities of the mOSAIC project.

6 CONCLUSIONS

In this paper we described the requirements, the design and the prototype implementation of Cloud Agency, an agent based software that provides provisioning capability to Cloud, the focus of the paper being on the policy based negotiation module. The Cloud-Agency has been conceived to be used both as a stand-alone software and as a component of the mOSAIC framework. The agency exposes negotiation, monitoring and autonomic reconfiguration services. Asynchronous APIs have been designed and provided by a REST interface, which follows the OCCI reference model. To execute services, agents communicate using standard FIPA messages and according to FIPA interaction protocols. A demonstrator has been implemented using the Jade platform. Detection of critical situations and autonomic reconfiguration of cloud resources are based on reconfiguration policies. Reconfiguration policies are event-condition-action rules that allow to design autonomic behavior in a flexible and dynamic way. Future works aim at engineering the complete Cloud Agency also by implementing and testing effective algorithms for negotiation, and real profiles for monitoring and reconfiguration in real case studies. Components of the mOSAIC framework that allows to automatically integrate the agent services into the Cloud applications will be also developed.

ACKNOWLEDGEMENTS

This research is supported by the European grant FP7-ICT-2009-5-256910 (mOSAIC) and partially supported by the grants POSDRU 21/1.5/G/13798, POSDRU/88/1.5/S/49516 and by the Romanian Government PNII grant nr. 12118/2008 (SCIPA).

REFERENCES

Aversa, R., Di Martino, B., Rak, M., and Venticinque, S. (2010). Cloud agency: A mobile agent based cloud system. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*, pages 132 –137.

Aversa, R., Venticinque, S., Martino, B. D., and Petcu, D. (2011). Agent based cloud provisioning and management, design and prototypal implementation. *1st International Conference on Cloud Computing and Services Science (CLOSER2011)*, pages 184–191.

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.

Grama, A., Kumar, V., and Sameh, A. (2000). Foundation for intelligent physical agents. www.http://www.fipa.org.

Kertesz, A., Kecskemeti, G., and Brandic, I. (2009). An sla-based resource virtualization approach for on-demand service provision. In *Proceedings of the 3rd international workshop on Virtualization technologies in distributed computing*, VTDC '09, pages 27–34, New York, NY, USA. ACM.

mOSAIC (2010). The mosaic project. http://mosaic-cloud.eu/.

Sim, K. (2010). Towards complex negotiation for cloud economy. In Bellavista, P., Chang, R.-S., Chao, H.-C., Lin, S.-F., and Sloot, P., editors, *Advances in Grid and Pervasive Computing*, volume 6104 of *Lecture Notes in Computer Science*, pages 395–406. Springer Berlin / Heidelberg.

Venticinque, S., Aversa, R., Di Martino, B., Rak, M., and Petcu, D. (2011). A cloud agency for sla negotiation and management. In Guarracino, M., Vivien, F., Trff, J., Cannatoro, M., Danelutto, M., Hast, A., Perla, F., Knpfer, A., Di Martino, B., and Alexander, M., editors, *Euro-Par 2010 Parallel Processing Workshops*, volume 6586 of *Lecture Notes in Computer Science*, pages 587–594. Springer Berlin / Heidelberg.

Xiao, Z. and Cao, D. (2010). A policy-based framework for automated sla negotiation for internet-based virtual computing environment. In *Proceedings of the 2010 IEEE 16th International Conference on Parallel and Distributed Systems*, ICPADS '10, pages 694–699, Washington, DC, USA. IEEE Computer Society.

You, X., Wan, J., Xu, X., Jiang, C., Zhang, W., and Zhang, J. (2011). Aras-m: Automatic resource allocation strategy based on market mechanism in cloud computing. *Journal of Computers*, 6(7).

Zulkernine, F. and Martin, P. (2011). An adaptive and intelligent sla negotiation system for web services. *Services Computing, IEEE Transactions on*, 4(1):31 –43.