Model based Cloud Programming

Stefan Wesner, Lutz Schubert

Höchstleistungsrechenzentrum Stuttgart,
Universität Stuttgart, Nobelstrasse 19, 70569 Stuttgart, Germany
emails: {wesner,schubert}@hlrs.de

Keywords: future clouds, cloud computing, programming model, execution environment

1. Introduction

Cloud computing has gained significant popularity not only for research and industrial use but similarly for private users. However the concept is already over-hyped. The promise of infinitely scalable elastic resources changing without complex systems administration and paying only for resources used cannot be realised immediately without coping with a certain level of complexity. Key question is how (and if at all) within organisations the obvious benefits of the cloud approach are realisable at considerable cost. For example IaaS (Infrastructure as a Service) public clouds have different interfaces and conditions of use thus for an organisation to 'scale out' requires considerable investment using skilled technical staff.

Allowing organisations to 'scale out' from their private cloud to public clouds is still a considerable technical challenge in particular if the solution should not be bound to a certain vendor or solution. Consequently there is a need for an open and integrated platform, to support both *deployment* and *design* of Cloud applications, together with an accompanying methodology that allows model-based development, configuration, optimisation, and deployment of existing and new applications independently of the existing underlying Cloud infrastructures.

2. Description of a problem solution

Within the recently started large scale European research project PaaSage such an open platform consisting of three main parts (1) An integrated development environment (IDE) based on Eclipse extended to allow expression of Cloud applications using a Cloud Modelling Language (CloudML) (2) the "executionware" as the intermediary layer exploiting the hints and semantics within the CloudML enabled application for mapping them on the execution infrastructure of different cloud providers. Part of this matching process is also the selection of the appropriate execution resources as not all execution layers come with all the necessary capabilities or monitoring hooks to serve all type of functions. The application development as well as the execution/mapping is supported by the third part, the "upperware". The "upperware" is a collection of tools and components supporting application designer and developers as well as cloud operators in their respective roles within the PaaSage platform.

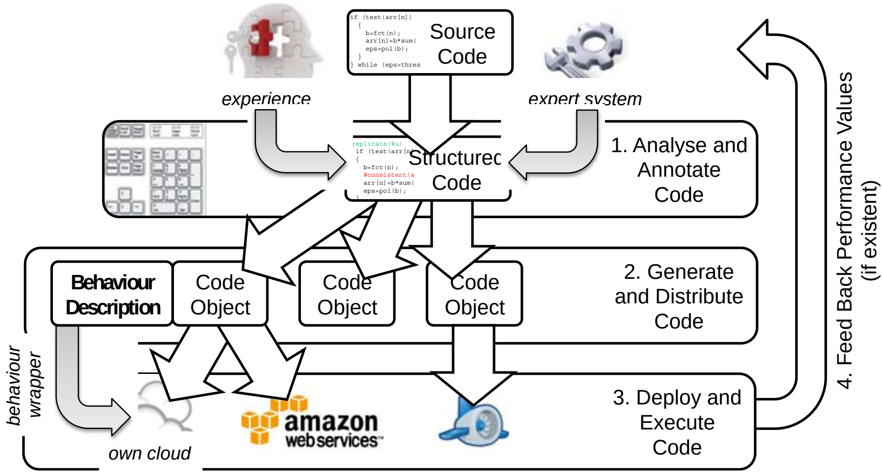

Fig. 1. Anticipated Cloud Programming Principle.

As shown in Fig. 1 above, the overall idea is to decouple the design and programming of the cloud application from the concrete underlying technological infrastructure or technology. This is achieved by allowing the application developer to express in a meta-model the architecture of the application and annotate the source code with hints and semantics using a Cloud modelling language (Cloud ML). The modelling language must not be understood as a semantically enriched workflow language but aims to provide enough information for a speculative profiler to derive at development time data dependencies, communication patterns, computing complexities and other characteristics of the functional units of the application and their interrelations. As the matching process has potentially a large number of feasible stable solutions there must be a possibility to work with uncertainty. Within PaaSage an intelligent stochastic reasoned deriving the options and provide a rating/ranking of them is needed. As the quality of such a process heavily depends on the collected historical data the reasoned will have to learn from each performance assessment for a chosen configuration and must apply stepwise optimizations.

While such an approach would allow already a post-execution optimization for future execution of similar applications it is not appropriate for long running applications. In order to allow also optimization steps during the execution a functional adaptation engine maintaining a causal link between behaviour of the underlying infrastructure and key performance indicators of the application is part of the anticipated platform.

In order to realise such functionality a highly scalable and effective information collection framework bringing together current information from the infrastructure as well as capabilities to store and analyse historical data is essential.

3. Results

The proposed platform in PaaSage is partially based on the Future and Emerging Technology Project S(o)OS where initial programming model for heterogeneous computing devices, speculative mapping and execution runtime on local and rather homogeneous resources could be already validated.

References

1. M. Assel, L. Schubert, D. Rubio Bonilla, and S. Wesner. Beyond Clouds: Towards Real Utility Computing. Scalable Computing: Practice and Experience, vol. 12, no. 2, pp. 179–191. SCPE, 2011.
2. L. Schubert, A. Kipp, and S. Wesner. Above the Clouds: From Grids to Service- oriented Operating Systems. Towards the Future Internet - A European Research Perspective, pp. 238 – 249. Amsterdam: IOS Press, 2009, pp. 238 – 249.