Microsoft Azure Research Award

Project title: *DEtecting and Solving Performance Antipatterns in Cloud Enviroments – DESPACE*

Abstract: *DESPACE* will develop a prototype of a performance analyser for the interoperability of online auctions with private owners of goods. The performance analyser will exploit an IoT infrastructure to gather data, and Microsoft Azure as cloud infrastructure for the back-end data processing and for planning the actions to support fast and secure communications during the auctions. The system will be self-adaptive and, in particular, will detect the potential performance flaws and/or security issues for the used resources and take suitable refactoring actions while ensuring data confidentiality and synchronisation.

Problem Statement

Performance management is a hot topic for the Information Technology (IT) industry, since performance is a serious problem in a significant fraction of projects. Performance failures occur when a software product is unable to meet its overall objectives due to inadequate performance. A survey [Compuware06] has been conducted on the performance management practice inviting 150 senior IT managers (i.e. responsible for testing and performance working at large organizations) to fill a questionnaire about their experience in the field. The survey demonstrates that performance failures are still occurring in many organizations, as more than half of organizations said they experience unexpected performance issues in 20% or more of their newly deployed applications.

The primary cause of performance failures is a reactive approach to performance during the development process. Cost and schedule pressures encourage project managers to adopt a “fix-it-late” approach in which performance is ignored until there is a problem. When a problem is discovered, more hardware is needed, developers must try to tune the software to meet performance objectives or both, but in some cases it is simply not possible to meet performance objectives by tuning. The fundamental lesson learned from the documented projects is that it is better to prevent performance failures and the resulting project crises. The problem addressed by this project is to manage software performance with a proactive approach thus to anticipate probable performance flaws, by developing a techniques for identifying and removing those problems early in the software development process.

To this end, we adopt a model-based approach that creates performance models early in the development cycle and uses quantitative results from these models to adjust the architecture and design with the purpose of meeting performance requirements. Figure 1 schematically represents the typical steps that are executed at a generic phase of the software lifecycle to conduct a software performance analysis process. Rounded boxes in the Figure represent operational steps whereas square boxes represent input/output data. The software performance engineering (SPE) process consists of three different phases: in the *modeling* phase an (annotated) software model is built; in the performance *analysis* phase a performance model is obtained through model transformation, and such model is solved to obtain the performance results of interest; in the *refactoring* phase the performance results are interpreted and, if necessary, feedback is generated as refactoring actions on the original software and/or performance model. Our model-based approach is aimed at building a software product that meets performance objectives, thus to avoid project crises due to the late discovery of performance issues.

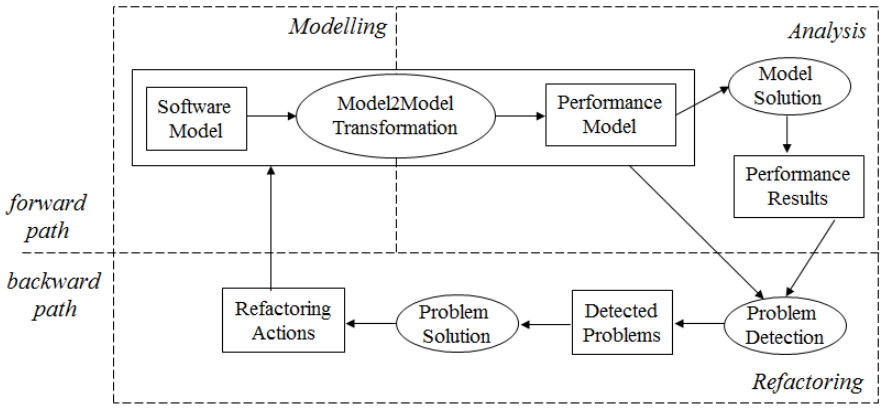

Figure 1 – Software performance engineering (SPE) process.

In this context we developed *PANDA* [Catia-PhDThesis11], i.e. a model-based framework addressing the refactoring phase by means of performance antipatterns [Smith03], that are recurring solutions to common mistakes (i.e. bad practices) in the software development. Such antipatterns can play a key role in the software performance domain, since they can be used in the search of performance problems as well as in the formulation of their solutions in terms of design alternatives. The key idea is to devise a methodology to keep track of the performance knowledge by integrating different forms of data (e.g. software model elements, performance indices), in order to support relationships between such data and to manage them over time, while the development advances.

However, the management of electronic purchases, bank transfers or even stock exchanges (required by auctions) gives rise to additional issues, since some bad human behaviors may occur: spam or junk mails, viruses, trojan horses or other attacks are commonly suffered. The main drawback in this domain is that security attacks clearly have an impact on the functionality of systems. The occurrence of attacks in software systems leads software designers to introduce different fault-tolerant techniques, such as recovery procedures, and/or security mechanisms, such as encryption of data, in order to react to intrusions. Therefore, the necessity of balancing security and performance in these systems becomes clear, and it is necessary to find appropriate security strategies while minimising performance penalties. To this end, it is necessary to anticipate, respond to and isolate communication issues among the participants of the auctions thus to mitigate the impact and to speed up the recovery.

This project is aimed at developing a performance analyser prototype, i.e., *DESPACE,* that exploit an IoT infrastructure to gather up to date performance analysis results, and Microsoft Azure seems very promising for the back-end data processing and for planning reconfiguration actions. The system will be self-adaptive and, in particular, will detect the potential performance flaws and/or security issues, thus to guarantee confidentiality among auction's participants.

Moving the model-based framework in cloud environments could be a challenging task and its feasibility mostly depends on the availability of the performance knowledge we need for reasoning about probable performance issues. In particular, our model-based framework needs the information coming from software models (e.g. software and hardware resources, the behavioral description of the provided services, etc.) as well as from performance models (e.g. utilization of hardware resources). In particular: (i) in the forward path, appropriate model annotations shall be specified for cloud-related performance parameters, and existing model transformations shall be adapted to deal with these additional parameters that must find their counterparts in the generated performance models; (ii) in the backward path, a combination of classical bottleneck analysis and (more recently studied) performance antipattern detection and solution techniques will be adopted, where it is expected that cloud-specific performance antipatterns will be identified and embedded in the SPE process.

The expected contributions are: (i) analysis of the domain, i.e. a repository of the performance knowledge we need to manage cloud environments; (ii) with the support of Microsoft Azure, detecting and solving software performance antipatterns in cloud environments thus to identify performance issues and solve them through suitable reconfiguration actions that provide performance improvements. The tangible assets of this project would be to implement a prototype tool able to detect and solve performance antipatterns in cloud environments (see Figure 2) by means of Microsoft Azure.

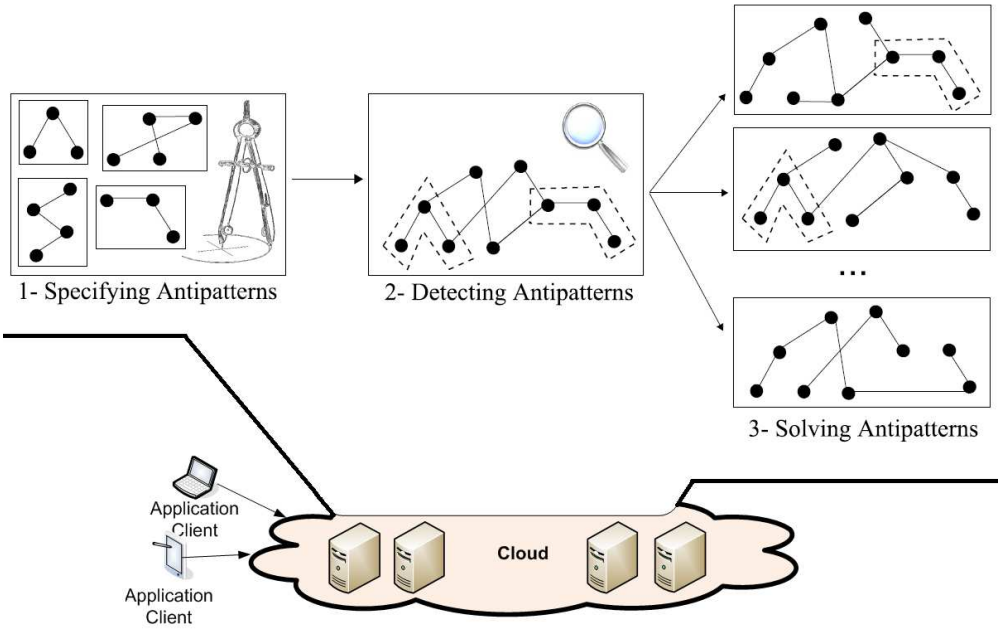

Figure 2 – Software performance antipatterns applied to cloud environments.

REFERENCES:

[Compuware06] Compuware. Applied performance management survey, Online: http://www.cnetdirectintl.com/direct/compuware/Ovum APM/APM Survey Report.pdf, 2006.

[Smith03] Connie U. Smith, Lloyd G. Williams: More New Software Antipatterns: Even More Ways to Shoot Yourself in the Foot. International CMG Conference: pp. 717-725, 2003.

[Catia-PhDThesis11] Catia Trubiani, "Automated generation of architectural feedback from software performance analysis results", Ph.D. Thesis in Computer Science, University of L'Aquila (AQ), L'Aquila, Italy, 2011. Online: http://www.di.univaq.it/catia.trubiani/phDthesis/PhDThesis-CatiaTrubiani.pdf

Microsoft Research offers to make available one Microsoft Azure Platform 12 month Research Pass to Gran Sasso Science Institute (GSSI), Italy at no cost for research purposes. Microsoft's intent is to provide the authorized users of the institution (including its researchers, instructors and students taking the instructors' courses) with access to Microsoft Azure.

The estimated total market value of this offer is USD $ 40,000.

Responsible of the project: Catia Trubiani, Post-doc Researcher at Gran Sasso Science Institute (GSSI), viale Francesco Crispi n.7, 67100, L'Aquila (AQ), Italy (IT), E-mail: catia.trubiani@gssi.infn.it