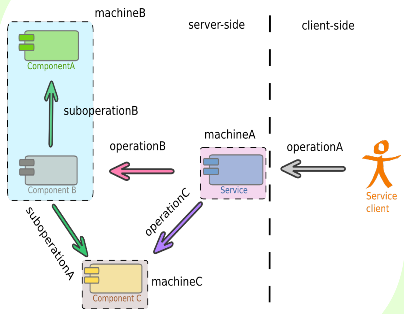
Semantic-based SLA-oriented performance monitoring in the ProActive environment

Dariusz Król <dkrol3@gmail.com>, Włodzimierz Funika <funika@agh.edu.pl>
Institute of Computer Science AGH, al. Mickiewicza 30, 30-059 Krakow, Poland

Intro

Nowadays, most of the science disciplines as well as most of the business markets are aided by computer systems. From some time now, commercial companies are getting more and more interested in one of the existing contract types, called Service Level Agreement which is signed between the software provider and the software consumer.

Example

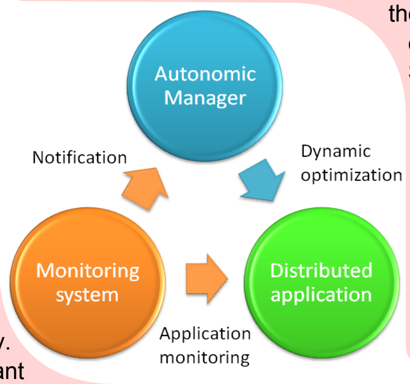


A typical example of an internally distributed service is depicted in the figure on the left. The client has a signed SLA contract with a service provider about maximum response time. The client sends a request to a server which uses other machines for parallel computation. Bad things may happen if these machines will run some other jobs at the same time.

Problem

In the last several years, the need for supporting long lasting and more complex applications with the computer systems was emerging. It`s a great issue when combining with a need of maintaining these applications on a level defined by SLA. Especially, when considering distributed business applications, the problem of enforcement of SLA contracts by adapting the system to the current state of the runtime environment in an automatic way is highly important.

Solution

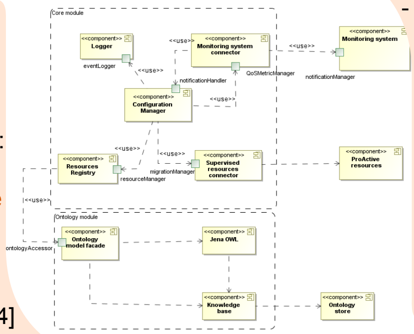


Our solution to the problem [1] is based on applying mechanisms known from the autonomic computing [3] for managing the application in an automatic way. The most important feature of the designed solution is optimizing an execution of the supervised application at runtime. Semantic information about the runtime environment along with on-line monitoring are utilized to precisely locate bottlenecks of the application and perform a necessary optimization according to a defined SLA.

Details

A prototype implementation of the presented approach is realized in form of a distributed system whose component diagram is depicted in the figure below. Two most important components are:

- Configuration manager which handles notifications from the monitoring system,
- Ontology model facade which responses to questions about information stored in a knowledge base.



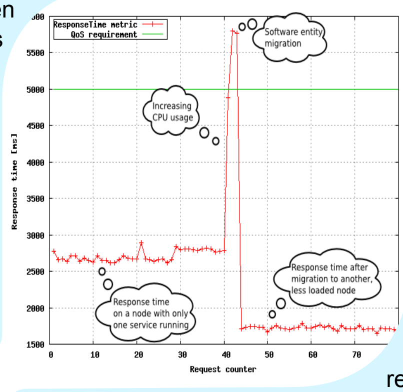
Tools

During implementation work the following tools were used:

- ProActive parallel suite [4] for creating distributed services,
- the IC2D tool [4] for monitoring a ProActive-based application,
- SemMon semantic-oriented monitoring system [2],
- Jena semantic framework,
- Java-related technologies (e.g. JMX, RMI).

Results

The prototype implementation was tested in a typical use case with one worker thread and two nodes. An SLA contract between service provider and consumer contained a requirement about the maximum response time of a service. The formulated strategy was applied to migrate the worker thread between available nodes at runtime according to current situation. The response time metric is depicted in the figure on the right.



Future work

Although the existing prototype can be applied in a number of use cases, it still requires work in the following aspects:

- scalability and reliability : in order to optimize high available application (e.g. 99.99% availability) the presented system has to work with an even higher value
- self-healing : such features as restarting or restoring the supervised services when the system (server) fails or goes down.

References:

1. Krol, D: Semantic-based SLA-oriented proactive performance monitoring, Msc. Thesis, AGH, Krakow, 2009.
2. W. Funika, P. Godowski, and P. Pegiel: A Semantic-Oriented Platform for Performance Monitoring of Distributed Java Applications, in: M. Bubak, G. D. van Albada, J. Dongarra and P.M.A. Sloot (Eds.), Proc. of ICCS 2008, Krakow, Poland, June 2008, volume III, LNCS 5103, Springer, 2008, pp. 233-242
3. Autonomic computing web site IBM, 2001, http://www.research.ibm.com/autonomic
4. Francoise Baude, Denis Caromel and Matthieu Morel. From Distributed Objects to Hierarchical Grid Components. LNCS 2888, pp. 1226-1242. Springer Berlin / Heidelberg, 2003

AGH

CYFRONET