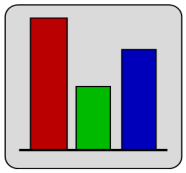

BEEN

Development Documentation

ObjectWeb
Open Source Middleware

Table of Contents

1 Project Overview

Project name:	BEEN (Benchmarking Environment)
Project description:	A generic tool for automated benchmarking in a heterogeneous distributed environment
Development time:	Autumn 2004–winter 2006
Advisor:	Tomáš Kalibera
Team members:	Jakub Lehotský, David Majda, Branislav Repček, Michal Tomčányi, Antonín Tomeček, Jaroslav Urban
Web page:	http://been.objectweb.org/
Support:	been@dsrg.mff.cuni.cz

BEEN is a generic tool for regression benchmarking in a heterogeneous distributed environment. While current tools for regression benchmarking are usually tied to a specific benchmark, BEEN presents a common execution environment suitable for running many kinds of different benchmarks (i.e. it is benchmark-independent).

For detailed description of the project and its relation to other similar tools, see *Part I* of the user documentation.

BEEN was developed in close cooperation with Distributed Systems Research Group[1] at the Faculty of Mathematics and Physics of the Charles University in Prague. The project was also accepted as an ObjectWeb[2] project and is hosted on ObjectWeb Forge[3].

1 http://dsrg.mff.cuni.cz/
2 http://www.objectweb.org/
3 http://forge.objectweb.org/

2 Team Member Responsibilities

Jakub Lehotský:

- Results Repository

David Majda:

- Software Repository
- web interface
- RSL interpreter
- Host Runtime
- project web
- documentation

Branislav Repček:

- Host Manager
- configuration detection
- load monitoring
- documentation

Michal Tomčányi:

- Benchmark Manager
- Xampler
- core architecture

Antonín Tomeček:

- Host Runtime
- Task Manager

Jaroslav Urban:

- RUBiS
- core architecture
- documentation

3 Project History

- **Summer 2004** – Michal Tomčányi and Antonín Tomeček work on a prototype implementation, testing the ideas of task execution in a distributed environment. Together with Tomáš Kalibera they try to find more developers interested in the project.
- **Autumn 2004** – Jakub Lehotský, David Majda, Branislav Repček and Jaroslav Urban join the project team. The project is officially started. The idea is to work fast and finish the project in spring 2005.

  First draft of the project architecture is created and the responsibilities of the team members are set: Jakub will write the Results Repository, David will implement the Software Repository and web interface to all components, Branislav will work on the detectors and Host Manager, Antonín will implement the Host Runtime and the Task Manager, Jaroslav and Michal will deal with the benchmarking core and benchmarks that BEEN needs to support (Xampler and RUBiS).
- **Winter 2004–2005** – David writes the first version of the Software Repository – the only component, which will not undergo serious architectural changes later. First sketch of the user interface with a module for the Software Repository and overall architecture of the Results Repository are created. But it becomes clear that the benchmarking parts of the system need serious rethinking. The spring finish date is abandoned.
- **Spring–summer 2005** – Michal and Jaroslav meet regularly and create the project architecture and design the plugin-based extension system, while Branislav writes first version of the detectors and the Host Manager. The rest of the team mostly waits for the benchmarking core to be in usable state.
- **Autumn 2005** – the Host Manager's database engine is rewritten by Branislav. David rewrites the web interface to be more modular, writes parts interacting with the Host Manager and implements the RSL query language parser and interpreter.
- **Winter 2005–2006** – work on the benchmarking core and plugins resulted in first successful run of the Xampler comparison experiment. Jakub implements the core of the Results Repository and David writes a web interface for it. We make progress but we see that the project will require much more work than anticipated.
- **Spring 2006** – our advisor Tomáš Kalibera approaches us with an idea to write a paper about BEEN for a VALUETOOLS 2006 conference. All members of the team agree and write the article in March and April. The paper becomes useful for better definition of the project architecture and later when writing the project documentation, but delays other work on the project. Nevertheless, intensive work on the Benchmark Manager and benchmarking plugins continues.

  We decide to finish BEEN in winter 2006, as summer and autumn dates seem too unrealistic.
- **Summer 2006** – work on all parts of BEEN continues with increasing intensity. Jaroslav successfully executes first comparison analysis using RUBiS plugin.

  The team decides that David will take over the Host Runtime and Antonín will continue only with development of the Task Manager.

  We begin to think about the documentation and decide that we will write it

collectively (everybody will document the part he implemented) and Branislav will be responsible for overall documentation structure and merging the parts together. We decide that we will use OpenOffice.org for writing the documentation, as we don't want to learn TeX or Docbook and the other good office suite – Microsoft Office – cannot be run on Linux, which most team members use as their primary system.

- **Autumn 2006** – David writes the web interface for the rest of the components, intensive work continues mainly on the Benchmark Manager, Host Manager, Results Repository and regression analysis support.

  The school provides us with a VMware server for testing. Jaroslav set-ups the virtual machines and begins to test BEEN on them. The testing server proves as very useful tool.

  The responsibility for documentation is transferred from Branislav to David and Jaroslav, because they have more time for it.

- **Winter 2006** – we finalize the work on the Benchmark Manager, Results Repository and regression analysis support. David and Jaroslav write most of the documentation.

4 Critical Decisions

The basics of the BEEN architecture were originally designed by our advisor, Tomáš Kalibera. The benchmarking core details were later designed mainly by Michal Tomčányi and Jaroslav Urban.

We consider splitting BEEN into several more-or-less independent components a good decision – it makes the system very modular and makes the development much easier than in a monolithic system since everybody works on his component and all the interaction between the components is routed through a few well-defined interfaces.

In retrospect, we should have probably merged the Task Manager and the Host Manager components, because they both manage information about hosts in the system and currently need to cooperate closely.

One of the decisions that influenced the development a lot was executing the tasks in separate virtual machines. Although it increased the reliability of the system, it also complicated debugging (remote debugging facilities of Java have to be used to attach a debugger to the external process), thus slowing down the development significantly.

When developing the web user interface, the use of Java language was a big drag. Its static nature, need of compilation and poor capabilities for working with strings made the development quite time-consuming and painful. We should have probably used some of the lightweight dynamic languages built on the Java platform, such as Nice, BeanShell, Groovy, JRuby or Jython.

On the other hand, the choice of the Java language and platform for development of BEEN components was probably the right one. The static typing helped to define interfaces and the compilation into bytecode allowed easy transporting of the code over the network and execution in heterogeneous environment. However it would be interesting to compare the speed of the development and bug rates with a similar system built in more dynamic language (such as Ruby or Python).

We have decided against the use of relational database for results storage since the data structure is hierarchical with varying table widths. This format is difficult to store in the relational database thus we decided to store data in custom format employing facilities already provided by the file system and NetCDF libraries. This provided for better performance and higher flexibility of the Results Repository and easier data access for the statistical processing by the R scripting language.

We have used automatic unit tests for several components. The decision proved as a good one, as it significantly decreased number of bugs in the tested components. We have also used the Selenium Core[4] testing suite for functional testing of the web interface – traditionally a difficult area to test. The functional tests proved even more useful than the unit tests, as they also tested the interaction between BEEN components.

From the management point of view, there was one mistake: We did not know what features will be really important for the end user and which will not. We didn't have previous experience with benchmarking, and the use cases of BEEN are quite unconventional. Our knowledge about benchmarking improved over time, but the overall uncertainty lead to implementation of several features which proved unnecessary, and also complicated prioritization of the development tasks. In general, implementation of BEEN was much more difficult and time consuming than anticipated.

4 http://www.openqa.org/selenium-core/

5 Further Development

BEEN was developed in close cooperation with Distributed Systems Research Group[5] at the Faculty of Mathematics and Physics of the Charles University in Prague. The research group is interested in using the project for purposes of benchmarking the Mono implementation of the .NET framework and possibly in other projects.

We have also received indications that several companies are interested in the project, and one seriously considers using BEEN for testing of its developed software. Currently, we do not want to disclose the names of the companies, to protect their interests.

The interest of several parties in BEEN will probably lead in its further development. Most members of the current BEEN team are not interested in continuing its development, however BEEN is an open source product (licensed under the terms of the LGPL license), so it can be easily extended and further developed by anyone. Project team members are committed to support further developers with their advice and knowledge about the project.

BEEN is already registered as a project at the ObjectWeb consortium[6]. We anticipate that the development will continue there, under the lead of Distributed Systems Research Group.

5 http://dsrg.mff.cuni.cz/
6 http://www.objectweb.org/