Evolving Applications: Object-migration with Ruby and GemStone

Bachelor Project Proposal, WS 2012/2013 – SS 2013

Software Architecture Group, Prof. Dr. Robert Hirschfeld

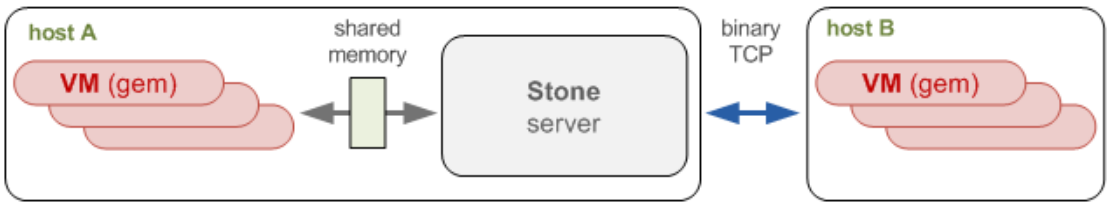
MagLev – a Ruby on GemStone

MagLev (http://maglev.gemstone.com) is a fast, 64-bit open source implementation of the Ruby programming language and its libraries. It is built on top of VMware's GemStone virtual machine and database, a Smalltalk proven in production for 30 years. It provides an object-oriented database system with persistence by reachability. GemStone/S has been used in finance, logistics, and telecommunication for years.

MagLev takes full advantage of GemStone features, with JIT compilation and distributed shared cache, fully ACID transactions, and enterprise-class NoSQL data-management capabilities. It can transparently manage terabytes of objects. There are no restrictions on what types of objects, classes, blocks, threads or continuations can be stored and executed in the database.

Object-migration with Ruby

The GemStone persistence method is similar to image-based persistence typical for other Smalltalks. Maintenance tasks like removal, update, and migration of data happen naturally at development time. However, Ruby developers are used to thinking about software in terms of files and expect to start afresh every time they launch an application. Persistent state is explicitly managed in frameworks such as Ruby on Rails with object-relational mapping or separate NoSQL data stores.



The GemStone VM and Database Architecture

In typical Web development, the structure and content of an application's data evolves over time. This has caused frameworks such as Rails to include means to anticipate such changes. *Migration-DSLs* are the typical way to express changes in databases there. GemStone/S, however, has

advanced, inherently object-oriented means of evolving objects, which are not yet available to MagLev applications.

The goal of this project is to propose and implement the advanced object-migration features of GemStone/S for MagLev, for easy use by Ruby developers using Rails and other current Ruby frameworks. In the course of the project, participants will be required to study, implement, and evaluate appropriate object-migration techniques for evolving Ruby applications with GemStone/S persistence.

Implementation

Existing approaches for Rails and Seaside will be implemented and evaluated. The students will analyze the workflow of both the typical Rails and Seaside developer to identify idioms and devise appropriate ways to expose GemStone features such as class history and class initializers to Ruby.

The implementation of the project will involve programming in C, Smalltalk, and Ruby for each layer of the MagLev stack, starting with the GemStone VM, over Smalltalk primitives to the Ruby core. Extreme Programming (XP) and other agile methodologies will be employed for software development.

Organization

A group of about six to eight (6-8) students may participate in the project. Organization and tasks will be determined by the project participants. The project will be carried out at the Hasso-Plattner-Institut in Potsdam. Project participants are expected to communicate with our partner via email, chat, or voice on a regular basis. In WS 2012/2013, participants will work on Seaside and Rails prototypes, possibly extending the VM with any missing features for either Seaside or Rails. Main steps in design and implementation of a solution are to be executed in SS 2013. Expected results include a software library accompanied by appropriate documentation.

Partner & Contact

Monty Williams, Peter McLain
VMware R&D, Gemstone Systems, Beaverton, OR, USA
http://www.gemstone.com/, monty@vmware.com, pmclain@vmware.com

Prof. Dr. Robert Hirschfeld, Tobias Pape, Tim Felgentreff
Software Architecture Group, Hasso-Plattner-Institut, Potsdam
http://www.hpi.uni-potsdam.de/swa, hirschfeld@hpi.uni-potsdam.de