Project Proposals 2008

Please remember to use the application template when applying for a project.

Garbage Collection

Separate Heap For VM objects

In most JVMs there is no confusion between memory allocated for the application and memory allocated by the running of the VM itself (for example a call to malloc() within the JIT). However, in Jikes RVM, the VM and the application are both written in Java. Moreover, they currently share the same heap. It would be very desirable to improve this situation and separately allocate VM and application objects. Aside from cleaner accounting and behaving more like a production JVM, there may be opportunities for performance optimizations since the lifetimes of objects created by the JIT will typically be bounded by the invocation of a single compilation, as an example.

This project would start by identifying all transitions from the application into the VM proper and channeling all such transitions through a zero-cost "trap", which simply serves as a marker. The trap can be viewed as analogous to a kernel trap in the OS setting. The project would also involve writing a simple checking routine which would walk the stack and determine whether execution was currently within the VM or application context. The combination of these mechanisms could then be used to identify and verify all application<->VM transitions.

RVM-399

Interested mentors: Steve Blackburn

Garbage Collection Visualization

the Jikes RVM has had GC visualization previously through the GCspy framework. This has become slightly out-of-date and it would be interesting to improve it and possibly look to integrate the visualization side of the work with a sister project of the RVM's tuning fork.

RVM-388
It may be interesting to run this project with Harmony-GC-3.

Interested mentors: Richard Jones

New Garbage Collection Implementations for MMTk

We would be interested in hearing about adding new GC implementations to MMTk. Our highest priorities are on-the-fly garbage collectors (RVM-400), the Compressor(RVM-401), and a classic Baker collector(RVM-403).

Implementing and debugging a garbage collector is difficult. This project is probably only suitable for someone with extensive implementation experience with either or both of MMTk and the algorithm in question.

Interested mentors: Steve Blackburn

Compilers

Copy eliminating baseline compiler

A new framework for baseline code generation using a copy eliminating baseline compiler.

RVM-161

Interested mentors: Eliot Moss, Ian Rogers

x86 64 compiler support

Related to the previous project, continue efforts to port to x86 64 the Jikes RVM.

RVM-169

Interested mentors: Ian Rogers

strictfp support

This project will look to extend existing compilers to have strictfp support to enable the execution of strict math code.

RVM-235

Interested mentors: Ian Rogers

Reduce cost of inlining runtime services

This project will look to reduce the cost of optimizing services repeatedly within the optimizing compiler.

RVM-148

Interested mentors: Ian Rogers

Loop unrolling

Implement a better loop unrolling phase taking into account various considerations about when and how to unroll.

RVM-404

Interested mentors: Ian Rogers

Java Operating Systems

Work on the continued integration of the RVM with JNODE

This project will look to continue to harmonize development effort of the Jikes RVM with the JNode operating system.

RVM-382

Interested mentors: Ian Rogers, Peter Barth, Fabien Duminy

Runtime services

Class Unloading

Currently all classes share a "Java table-of-contents" or the JTOC which has no support for unloading classes. We need to restructure the JTOC interface to support class unloading by making it per class loader, or per class. This is likely to also involve looking into modifying the current calling conventions.

RVM-324

Interested mentors: Robin Garner, Ian Rogers

Lock Reservation/Biasing

Lock reservation and biasing are ways of giving threads ownership of objects so that they can avoid or use cheaper tests to confirm their ownership of an object. Handling contention and releasing ownership of the objects is complicated. This project will look to implement these schemes in the RVM.

RVM-290

Interested mentors: Tony Hosking, Ian Rogers

Native Threading

The Jikes RVM's threading model was refactored and made more modular last year. A native thread stub exists without any implementation. We should implement this stub for different underlying threading systems. Some of the underlying threading systems we'd hope to support are pthreads, Windows threads, glib threads and Harmony's hythreads.

RVM-91

Interested mentors: Tony Hosking, Ian Rogers

Harmony class library support

The Apache Harmony project provides open source class libraries and virtual machines that are aiming to be certified as Java compatible. The Harmony libraries are already used by a number of products and we would like to integrate them into the Jikes RVM.

RVM-358
Harmony-VM-2

Interested mentors: Ian Rogers, Tim Ellison

OpenJDK class library support

The OpenJDK class libraries are the standard class libraries in use in Sun and other production VMs. They are now available under the GPL but with a native interface implemented typically in a file called jvm.cpp (see Cacao). We would like to integrate with OpenJDK and fix likely issues in wanting to have Java-in-Java implementations of a lot of their calls, as well as support for different threading models.

RVM-373

Interested mentors: Andrew Hughes

Debugger support

Following last years SoC we have partial JDWP support. It would be good to increase this to include things such as break points. It would also be good to implement the JVMTI native interface for debugging.

RVM-33

Interested mentors: Kathryn McKinley, Steve Blackburn, Ian Rogers

Modularity

Currently there is a connection between different parts of the RVM that is less than desirable. For example, the on-stack-replacement algorithm is implemented in parts in the class loader. We would like to have better isolation of components to more easily allow plug in replacements to be created. Such changes are likely to have an effect across the code base, so interacting with the development community will be important.

RVM-405

Interested mentors: Ian Rogers