

Ideas for code contributions

This page highlights some ideas for code contributions. Some of these selections also appear in our [issue tracker](#). Please volunteer to pluck these fruit, or suggest high-priority, but low-effort tasks by creating new tracker entries. If you want to help out and don't know what to do, ask us on the [mailing lists](#) and mention what kind of tasks you're interested in.

Low-hanging fruit (high priority, low-effort)

- **Complete missing pieces of library interface:** There are a handful of methods in Jikes RVM's implementation of a few core library classes (`java.lang.Class`, `java.lang.Runtime`, `java.lang.Throwable`) that throw `VM_UnimplementedError`. Provide an implementation for one of these methods.
- **Improve efficiency of some optimizing compiler phase:** Many of the optimizing compiler passes use sub-optimal data structures, and compile-time could easily be improved. Some simple phases to start with include the local optimizations and flow-insensitive optimizations.
- **Implement missing JNI functionality:** See tracker items: [CreateJavaVM](#), [Attach/DetachCurrentThread](#). [OnUnload](#) is also missing but not a high priority because the Jikes RVM currently does [not implement class unloading](#).
- **Strip mining:** This optimization would be profitable, by reducing the overhead of yield point checks in tight loops.
- **Global Array Bounds Check Elimination:** While the literature presents several algorithms for array bounds check elimination, the optimizing compiler does not have a complete implementation. Note that there are some non-trivial technical difficulties with [ABCD](#), regarding integer overflow. A simple range propagation dataflow solver would be an improvement over what the system has now (namely nothing).

High priority, large effort

- **Support for OpenJDK as a class library:** We have received community contributions for OpenJDK support. However, the patch set is large and not split into commits. Some parts of the patch set probably need to be changed or rewritten. You can help by grouping changes into commits and testing those. Reviews would also be helpful.
- **Support for x64:** The situation is basically the same as for OpenJDK.
- **Better testing infrastructure:** We currently using Cattrack, a custom Ruby-on-Rails application for our regression tests. Cattrack needs to be updated to a new version of Ruby-on-Rails. An alternative would be to use a CI server like Jenkins. In that case it would probably be necessary to write custom plugins so that we can keep Cattrack's functionality.
- **Improve situation for the optimizing compiler so that we can enable previously disabled optimizations:** There are a number of optimizations in the opt compiler that are currently disabled by default because they are considered too buggy. We currently don't have proper tests for the optimizing compilers so we need to write a lot of tests to ensure that the compiler works correctly. The compiler-internal IR verification (and its paranoid variant) could also be improved. There are also opportunities in the refactoring of the [phase organisation](#).

A number of other possible projects can be found on the Project proposals / ideas pages that are linked from for our [Google Summer of Code](#) pages.