inside the compilers. Java's inherent threading has meant all garbage collectors are parallel and integrate with the runtime model. Java's libraries provide inspiration for the interfaces of the collector, which means that writing new garbage collectors is significantly simplified in MMTk.

Lessons Learned

Jikes RVM is a successful research virtual machine, providing performance close to the state-of-the-art in a flexible and easy-to-extend manner. Being written in the language supported by the runtime allows close integration and reuse of components. The use of Java allows a simple-to-understand codebase, good modularity, and the use of high-quality tools, such as IDEs.

The Java language and the implementations of JVMs are developing, and Jikes RVM has and will continue to develop along with these changes. One interesting development is the X10 programming language, which tackles the issue of how programmers will develop applications for many core systems by providing thread-safety guarantees, just as garbage collection provides memory-safety guarantees in Java. Jikes RVM's codebase already provides an excellent test bed for development ideas with X10. As JVM implementations create new optimizations, such as new garbage-collection techniques or object inlining, the ability to slot these optimizations into a framework such as Jikes RVM means the runtime, compilers, and codebase also improve and demonstrate the strength of metacircularity.

The extensibility of a metacircular environment make Jikes RVM an excellent platform for multilanguage virtual machine research. This extension would also allow aspects of Jikes RVM to be written in different programming languages.

There are many other exciting extensions to Jikes RVM and related projects, including support for languages such as C and C++ through binary translation, extensions to provide aspect-oriented programming within the virtual machine, and making the entire virtual machine into an operating system to remove barriers to runtime optimization. Although initially controversial, garbage collection and adaptive and link-time optimization are now desired by developers. Jikes RVM shows how this can be achieved through one common metacircular environment that will help develop, and adapt to, these features. Jikes RVM supports a large research community and achieves high-performance. By having a beautiful architecture, Jikes RVM can continue to provide a platform for future runtime environments.

References

Abelson, Harold, Gerald Jay Sussman, and Julie Sussman. 1985. *Structure and Interpretation of Computer Programs*. Cambridge, MA: MIT Press.

Aho, Alfred, Ravi Sethi, and Jeffrey Ullman. 1986. *Compilers, Principles, Techniques, and Tools*. Boston, MA: Addison-Wesley.