

Kaffe OS: A Java Operating System

Single-language runtime systems, in the form of Java virtual machines, have become widely deployed platforms for executing untrusted mobile code. For example, Java virtual machines are used to execute downloaded applets inside Web browsers, uploaded servlets within Web servers, and uploaded queries within databases. Type-safe languages such as Java provide much more safety than languages like C or C++: they provide inter-application memory protection, so that one program cannot directly crash another program.

result, one program could attempt to deny service to other programs on the same virtual machine. The ability to isolate programs from each other (both in terms of their safety and resource usage) is necessary to allow Java virtual machines to execute untrusted code, which may be buggy or outright malicious.

We have built a prototype Java virtual machine that we call KaffeOS, which provides these features for a Java runtime. KaffeOS supports the OS abstraction of a process in a Java virtual machine. Each process executes as if it were run in its own virtual machine, and each process' heap is separately garbage-collected. KaffeOS's ability to isolate processes from each other enables it to thwart resource-based denial-of-service attacks. That is, programs cannot acquire arbitrary amounts of memory or CPU time and prevent other programs from executing. Although KaffeOS's relatively poor performance on benchmarks (it is based on the public-domain Kaffe JVM), it can deliver better performance than IBM's commercial JVM in the presence of malicious code. Our performance results demonstrate that resource-based denial-of-service attacks can be stopped effectively on KaffeOS.

KaffeOS demonstrates that Java can become a robust platform for executing untrusted mobile code. More broadly, the KaffeOS prototype demonstrates that there is value in applying operating systems principles to building language runtimes. In addition, it demonstrates that software mechanisms can be used to support operating systems functionality in language runtimes at relatively low cost. We plan to make a public release of KaffeOS in the near future. Our work on KaffeOS has been supported by DARPA.

KaffeOS Features

- Isolation of untrusted code
- Safe and complete process termination
- Precise memory and CPU accounting
- Inter-process object sharing

Unfortunately, type-safe language runtimes are not yet good enough at isolating programs from each other. They do not provide the ability to fully isolate applications from each other. Even though one Java program cannot directly crash another one, it could crash the Java virtual machine on which they both run. They also do not provide any mechanisms to limit a program's resource consumption. As a

For more information about KaffeOS:

Wilson Hsieh • wilson@cs.utah.edu • Tel:(801) 585-5047
School of Computing at the University of Utah
50 S. Central Campus Dr Rm 3190 • Salt Lake City, UT 84112-9205
<http://www.cs.utah.edu/~wilson>

