



Tail Calls for the Java HotSpot™ VM

Diplomaufgabe für Arnold Schwaighofer
Matr.-Nr.: 9920231

When a method call is the last instruction of a method, it is in certain cases possible to remove the stack frame of the caller method before the stack frame of the callee is created. Such method calls are called *tail calls*. They are important for functional programming languages where recursion is commonly used for many algorithms. Without tail calls, the stack would overflow. Implement this optimization for the Java HotSpot™ VM of Sun Microsystems.

The Java HotSpot™ uses a mixed-mode approach for the execution of Java bytecodes. A method is at first interpreted and later compiled by a just-in-time compiler if its invocation counter crosses a certain threshold. Two different just-in-time compilers are available: the client compiler and the server compiler. Start your implementation with tail calls for the client compiler. It is necessary to detect if a method call is a suitable candidate for a tail call, and to modify the method prologue and epilogue to build the correct stack frames. Test and evaluate your implementation with both object-oriented and functional programs.

Several dynamic features of Java such as security checks and dumping of stack traces are affected by tail calls. Investigate the implications of tail calls and check if your changes comply with the Java virtual machine specification.

Also investigate the possibility for hard tail calls, i.e. possibilities to guarantee tail calls for certain call sites. This requires tail calls to be marked in the bytecodes. The class loader, bytecode verifier, interpreter and just-in-time compiler must handle these marked calls.

Der Fortgang der Arbeit ist in 14-tägigem Abstand mit dem Betreuer zu besprechen. Für die Ausarbeitung der schriftlichen Arbeit sind die Richtlinien des Instituts für Systemsoftware zu beachten.

Programmiersprache: C++

Nähere Auskünfte: Dipl.-Ing. Christian Wimmer