

UNIVERSITY OF SUSSEX
COURTESY



The Effect of Java Exceptions on Code Optimizations

Andrew St. Johns

Deshaun Watson

{andrewst, desw}@cogs.susx.ac.uk

Report 1

Part 1

Computer Science
Department of Computer Science
University of Sussex
Brighton BN1 9QJ
UK
N1 - 1

The Effect of Java Exceptions on Code Optimizations

Abstract. This paper discusses the effect of Java exceptions on code optimizations. It shows that the presence of exceptions makes control flow prediction highly unpredictable and thus hinders aggressive optimization techniques.

Abstract. The Java language supports exceptions by providing language constructs that allow the programmer to manage any exception that may occur during execution. The Java Virtual Machine (JVM) supports this management by maintaining tables that provide information on the exception handler's location along with the type of exception that it handles. The resulting program's control flow during execution is highly unpredictable and is not amenable to static analysis. Such control flow prediction is a valuable asset to an aggressively optimized program.

▶ E to Java Ex ptions on Co pt at ons

in a predefined intermediate language. Clearly this is not an ideal solution since the behaviour of the program will change and the benefits of exception-based control flow will be lost.

▶ R at R s ar ▶

The importance of predicting the dynamics of program execution was examined by Larus [3]. He shows that successful prediction of the control flow leads to the exposure of optimization opportunities. He also notes the problems caused by the cost of running the tools that identify interprocedural paths. His paper presents another factor that increases the number of dynamic procedural context changes. The introduction of exceptions therefore increases the cost of exposing the executed control flow and therefore decreases the opportunity for optimizations.

An' r t ns an' D s atson

3 Analysis

4 Conclusions

Exceptions in Java add many benefits to the design and implementation of a program from enforcing caller handling of errors to recovering from system resource limits. As a strongly typed object-oriented language, Java adds to these benefits by enabling the compiler to validate that explicit exceptions are being handled in the surrounding scope. With these benefits come disadvantages. The fragmen-