Abstract

This paper describes our experience in today’s working of JVM which we use is platform dependent, it create an issue regarding Java being platform independent. The different platform uses different JVM architecture like RISC and CISC use different JVM. This paper defines porting Compaq’s Fast VM from the Alpha processor architecture to the Intel x86 processor architecture. We encountered several opportunities and pitfalls along the way in porting a JVM designed for RISC architecture to CISC architecture. Our goal was to preserve most of the FastVM’s performance benefits already available on the Alpha platform, and modify or discover new optimizations as they were required for x86. We found that by porting a
Cross Architectural Portability of a JVM Implementation

fast RISC JVM to x86, we could generate a JVM with performance competitive to state-of-the-art production JVM implementations. This step can provide more efficient and more compatible programming and implementation environment.

References

- M. Poletto and V. Sarkar. Linear scan register allocation. ACM Transactions on
- D. J. Scales, K. H. Randall, and S. G. J. Dean. The Swift Java Compiler: Design
- R. Sethi and J. D. Ullman. The generation of optimal code for arithmetic expressions.
12Crosby Drive, Bedford, MA 01730, USA, second edition, 1995.
- T. Suganuma, T. Ogasawara, M. Takeuchi, T. Yasue, M. Kawaihito, K. Ishizaki, H.
Komatsu, and T. Nakatani. Overview of the IBM Java Just-in-Time Compiler. IBM Systems
- B. -S. Yang, S. -M. Moon, S. Park, J. Lee, S. Lee, J. Park, Y. C. Chung, S. Kim, K.
Ebioglu, and E. Altman. LaTTe: A Java VM just-in-time compiler with fast and efficient register
allocation. In Proceedings of the 1999 International Conference on Parallel Architectures and
Compilation Techniques (PACT &apos;99), pages 128-138, Newport Beach, California, Oct.

Index Terms
Computer Science
Mobile Computing And
Software Engineering

Keywords