Characterization and Optimization of Java Applications

Abstract:

Over the past few years the Java programming language has come into widespread use. It offers a variety of features that allow applications to be developed with relative ease. It is also the first language in widespread use to offer true platform independence, allowing applications developed on one computer architecture to run on many others. Unfortunately platform independence comes with a significant loss of runtime performance by necessitating the use of a platform independent representation of the program known as Java bytecodes. As a result, many new optimization techniques have been developed in an attempt to make the performance of Java applications competitive with applications developed in other languages. Unfortunately, current optimization techniques are unable to completely bridge the performance gap. In order to facilitate the development of further optimization techniques, a detailed analysis of the Java bytecodes used during the execution of a Java program has been performed. This was accomplished by modifying an existing Java virtual machine to record information about each bytecode as the application was executing. Custom software was also developed to analyze the data once it was collected. A number of interesting patterns were found in the data collected from several commonly used Java benchmarks including those contained within the industry standard SPEC JVM98 benchmark suite. One particularly interesting pattern is that certain sequences of Java bytecodes occur frequently during the execution of many different programs. Furthermore, when these sequences of bytecodes are considered as a group, rather than individually, a number of potential optimizations become apparent. A set of optimization rules are presented and applied to some of the most frequently occurring sequences. Theoretical performance improvements are also presented for each of the sequences considered.