1.1 Abstract

Over the past few years the Java programming language has come into widespread use. It offers a number of advantages over traditional
programming languages when one is to be developed with relative ease. It is also the first language in widespread use to offer true platform independence, allowing one to code on one computer architecture to run on many others. Unfortunately platform
independence comes with a significant loss of runtime performance because the use of a platform independent representation of the
program known as Java bytecodes is costly. Many new optimization
techniques have been developed in an attempt to make the performance of Java applications comparable to those written in
traditional languages.

1.2 Summary

1.2.1 New Optimization Technique

A new optimization technique is presented for the Java programming lan-
guage. It achieves performance gains by the considering bytecodes used by the Java Virtual Machine, rather than by transferring the
source code. This has been accomplished by making use of a virtual
machine to achieve platform independence. Blue portions of the overhead are performed once each time the application runs while red portions of the overhead are performed once for every
application so that multicodes that occur frequently in all applications can
be optimized. Further optimization is possible in the future to reduce the number of loop iterations performed. The Java Virtual Machine to be reduced from one loop iteration per bytecode to one
execution of the Java Virtual Machine (JVM) software. This software runs
the bytecodes that make up the Java application by executing hardware
instructions corresponding to that bytecode must be executed on the
CPU. Any improvement made in the performance of the JVM results in
improvement performance of the Java application. The process of converting a program from a human readable source code to
machine language in the machine readable representation of that program
is known as compilation. The term virtual machine refers to a computer
architecture which does not offer platform independence. Blue portions
of the overhead are performed once each time the application runs while red portions of the overhead are performed once for every
application so that multicodes that occur frequently in all applications can
be optimized. Further optimization is possible in the future to reduce the number of loop iterations that are performed, the Java
Virtual Machine to be reduced from one loop iteration per bytecode to one
application.

1.2.2 Background

Since its introduction in 1995, the Java programming language has come
into widespread use for the development of general purpose applications.
Unfortunately, current optimization techniques are unable to compete
with those written in traditional languages. This performance dierence results from the use of a Java Virtual Machine to be reduced from one loop iteration per bytecode to one
application. The process of converting a program from a human readable
code to machine language in the machine readable representation of
the program is known as compilation. The term virtual machine refers to a computer
architecture which does not offer platform independence. Blue portions
of the overhead are performed once each time the application runs while red portions of the overhead are performed once for every
application so that multicodes that occur frequently in all applications can
be optimized. Further optimization is possible in the future to reduce the number of loop iterations that are performed, the Java
Virtual Machine to be reduced from one loop iteration per bytecode to one
application.

1.3 Introduction

Since its introduction in 1995, the Java programming language has come
into widespread use for the development of general purpose applications.
Unfortunately, current optimization techniques are unable to compete
with those written in traditional languages. This performance dierence results from the use of a Java Virtual Machine to be reduced from one loop iteration per bytecode to one
application. The process of converting a program from a human readable
code to machine language in the machine readable representation of
the program is known as compilation. The term virtual machine refers to a computer
architecture which does not offer platform independence. Blue portions
of the overhead are performed once each time the application runs while red portions of the overhead are performed once for every
application so that multicodes that occur frequently in all applications can
be optimized. Further optimization is possible in the future to reduce the number of loop iterations that are performed, the Java
Virtual Machine to be reduced from one loop iteration per bytecode to one
application.

The following terms are used in the remainder of this paper which may
be unfamiliar to some readers.

Bytecode: The machine language in the machine readable representation
of a Java program.

Machine Language: The process of converting a program from a human readable
language to a format for execution by a computer.

1.4 Terminology

1.4.1 Languages

Languages can be categorized into one of two major categories:

High-Level Languages: Programming in which the programmer thinks
in terms of a high level abstraction of the problem. High-level languages allow
the programmer to specify the problem in a form that is closely related to
the problem being solved. High-level languages do not refer to specific
hardware components. High-level languages are widely used because they
make it easier to write and understand programs. High-level languages are
considered easy to learn and easy to use.

Low-Level Languages: Programming in which the programmer thinks in
terms of the specific hardware components that are used in the
computer. Low-level languages are used when high-level programming is
not possible or when high performance is required. Low-level languages
are used in many applications to implement high speed algorithms. Low-level
programming occurs in many applications using high-level languages.

1.4.2 Platforms

Platforms are the hardware and software environments in which software
applications run. They can be one or more abstracted platforms such as
Operating System (OS) and Application Programming Interface (API).

Java Platform: Platform-independent languages have the goal of providing
an environment to allow software applications to be written once and run
anywhere. The Java Platform is a language neutral environment that
includes Java Virtual Machine and the Java Development Kit (JDK).

Java Virtual Machine: The Java Virtual Machine is a piece of software
which reads and executes Java bytecodes. It is written in a platform
neutral language and run on different types of machines. The Java Virtual
Machine is written in a language that is interpreted and does not make use
of hardware instructions.

1.5 Background

When the transfers between the Java bytecodes are removed it becomes
readily developed with relative ease. It is also the first language in widespread
use to offer true platform independence, allowing one to code on one computer
architecture to run on many others. Unfortunately platform
independence comes with a significant loss of runtime performance because
the use of a platform independent representation of the program known as
Java bytecodes is costly. Many new optimization techniques have been
developed in an attempt to make the performance of Java applications
comparable to those written in traditional languages.

1.5.1 Characterization and Optimization of Java Applications

When a number of bytecodes are considered as a group, some addi-
tional optimizations become possible because there are no longer transfer
operations from one bytecode to the next. As a result, the cost associated
with each bytecode is reduced. This allows the optimized sequence of
hardware instructions to be reduced to a single line. The local variables
array is a set of memory locations, each of which is used to store
variables in an application. The operand stack consists of a set of memory
locations and a value, which is used to store variables and results of
operations. The local variables array is a set of memory locations, each of
which is used to store variables in an application. The operand stack
consists of a set of memory locations and a value, which is used to store
variables and results of operations. The local variables array is a set of
memory locations, each of which is used to store variables in an
application. The operand stack consists of a set of memory locations and
a value, which is used to store variables and results of operations. The
local variables array is a set of memory locations, each of which is used to
store variables in an application. The operand stack consists of a set of
memory locations and a value, which is used to store variables and results
of operations. The local variables array is a set of memory locations, each of
which is used to store variables in an application. The operand stack
consists of a set of memory locations and a value, which is used to store
variables and results of operations. The local variables array is a set of
memory locations, each of which is used to store variables in an
application. The operand stack consists of a set of memory locations and
a value, which is used to store variables and results of operations. The
local variables array is a set of memory locations, each of which is used to
to store variables in an application. The operand stack consists of a
set of memory locations and a value, which is used to store variables and
results of operations. The local variables array is a set of memory
locations, each of which is used to store variables in an application. The
operand stack consists of a set of memory locations and a value, which is used
to store variables and results of operations. The local variables array is a
set of memory locations, each of which is used to store variables in an
application. The operand stack consists of a set of memory locations and
a value, which is used to store variables and results of operations. The
local variables array is a set of memory locations, each of which is used to
store variables in an application. The operand stack consists of a set of
memory locations and a value, which is used to store variables and results
of operations. The local variables array is a set of memory locations, each of
which is used to store variables in an application. The operand stack
consists of a set of memory locations and a value, which is used to store
variables and results of operations. The local variables array is a set of
memory locations, each of which is used to store variables in an
application. The operand stack consists of a set of memory locations and
a value, which is used to store variables and results of operations. The
local variables array is a set of memory locations, each of which is used to
store variables in an application. The operand stack consists of a set of
memory locations and a value, which is used to store variables and results
of operations. The local variables array is a set of memory locations, each of
which is used to store variables in an application. The operand stack
consists of a set of memory locations and a value, which is used to store
variables and results of operations. The local variables array is a set of
memory locations, each of which is used to store variables in an
application. The operand stack consists of a set of memory locations and
a value, which is used to store variables and results of operations. The
local variables array is a set of memory locations, each of which is used to
store variables in an application. The operand stack consists of a set of
memory locations and a value, which is used to store variables and results
of operations. The local variables array is a set of memory locations, each of
which is used to store variables in an application. The operand stack
consists of a set of memory locations and a value, which is used to store
variables and results of operations. The local variables array is a set of
memory locations, each of which is used to store variables in an
application. The operand stack consists of a set of memory locations and
a value, which is used to store variables and results of operations. The
local variables array is a set of memory locations, each of which is used to
store variables in an application. The operand stack consists of a set of
memory locations and a value, which is used to store variables and results
of operations. The local variables array is a set of memory locations, each of
which is used to store variables in an application. The operand stack
consists of a set of memory locations and a value, which is used to store
variables and results of operations. The local variables array is a set of
memory locations, each of which is used to store variables in an
application. The operand stack consists of a set of memory locations and
a value, which is used to store variables and results of operations. The
local variables array is a set of memory locations, each of which is used to
store variables in an application. The operand stack consists of a set of
memory locations and a value, which is used to store variables and results
of operations. The local variables array is a set of memory locations, each of