Abstract

Matlab is a popular dynamic array-based language commonly used by students, scientists and engineers who appreciate the interactive development style, the rich set of array operators, the extensive builtin library, and the fact that they do not have to declare static types. Even though these users like to program in Matlab, their computations are often very compute-intensive and are better suited for emerging high performance computing systems. This paper reports on MiX10, a source-to-source compiler that automatically translates Matlab programs to X10, a language designed for "Performance and Productivity at Scale"; thus, helping scientific programmers make better use of high performance computing systems.

There is a large semantic gap between the array-based dynamically-typed nature of Matlab and the object-oriented, statically-typed, and high-level array abstractions of X10. This paper addresses the major challenges that must be overcome to produce sequential X10 code that is competitive with state-of-the-art static compilers for Matlab which target more conventional imperative languages such as C and Fortran. Given that efficient basis, the paper then provides a translation for the Matlab `parfor` construct that leverages the powerful concurrency constructs in X10.

The MiX10 compiler has been implemented using the McLab compiler tools, is open source, and is available both for compiler researchers and end-user Matlab programmers. We have used the implementation to perform many empirical measurements on a set of 17 Matlab benchmarks. We show that our best MiX10-generated code is significantly faster than the de facto Mathworks' Matlab system, and that our results are competitive with state-of-the-art static compilers that target C and Fortran. We also show the importance of finding the correct approach to representing the arrays in X10, and the necessity of an *IntegerOkay* analysis that determines which double variables can be safely represented as integers. Finally, we show that our X10-based handling of the Matlab `parfor` greatly outperforms the de facto Matlab implementation.

1 Getting started guide

The artifact is provided as a zipped tar archive(.tar.gz) of a directory named `MiX10`. The `README.txt` file provided in the MiX10 directory gives a list of the various components of our artifact.

The main component of our artifact is the MiX10 compiler itself, provided as an executable jar file. It is a command-line only tool. It takes an input Matlab file and compiles it to X10. In this section we first list the prerequisites required to run the jar file and then give step-by-step instructions to run the compiler. We also give a description of various command-line switches available with the compiler.

1.1 Prerequisites

The following list gives the various prerequisites in order to successfully run the MiX10 compiler:

Operating System: In its current version MiX10 is available only for Unix and Unix-like operating systems only. In particular, we have tested the MiX10 compiler on Ubuntu Linux 12.04 (kernel version: 3.8.0-35-generic #52 precise1-Ubuntu SMP Thu Jan 30 17:24:40 UTC 2014) and Mac OS X 10.9 (Darwin Kernel Version 13.2.0: Thu Apr 17 23:03:13 PDT 2014).

Java: The MiX10 compiler requires the Java Runtime Environment 1.8.0 (Java 8). It is incompatible with the previous versions of the Java Runtime Environment.