Geophysical Research Abstracts
Vol. 16, EGU2014-12254, 2014
EGU General Assembly 2014

MIST: An Open Source Environmental Modelling Programming Language Incorporating Easy to Use Data Parallelism.

Tim Bellerby

University of Hull, Department of Geography, Environment and Earth Sciences Hull, United Kingdom
(t.j.bellerby@hull.ac.uk)

Model Integration System (MIST) is open-source environmental modelling programming language that directly incorporates data parallelism. The language is designed to enable straightforward programming structures, such as nested loops and conditional statements to be directly translated into sequences of whole-array (or more generally whole data-structure) operations. MIST thus enables the programmer to use well-understood constructs, directly relating to the mathematical structure of the model, without having to explicitly vectorize code or worry about details of parallelization. A range of common modelling operations are supported by dedicated language structures operating on cell neighbourhoods rather than individual cells (e.g.: the 3x3 local neighbourhood needed to implement an averaging image filter can be simply accessed from within a simple loop traversing all image pixels). This facility hides details of inter-process communication behind more mathematically relevant descriptions of model dynamics.

The MIST automatic vectorization/parallelization process serves both to distribute work among available nodes and separately to control storage requirements for intermediate expressions – enabling operations on very large domains for which memory availability may be an issue.

MIST is designed to facilitate efficient interpreter based implementations. A prototype open source interpreter is available, coded in standard FORTRAN 95, with tools to rapidly integrate existing FORTRAN 77 or 95 code libraries. The language is formally specified and thus not limited to FORTRAN implementation or to an interpreter-based approach. A MIST to FORTRAN compiler is under development and volunteers are sought to create an ANSI-C implementation. Parallel processing is currently implemented using OpenMP. However, parallelization code is fully modularised and could be replaced with implementations using other libraries. GPU implementation is potentially possible.