MAFE: An Environment for MATLAB-to-C Compilation Supporting Static and Dynamic Memory Allocation and Multi-Level User Interactive Code Optimization

Christakis Lezos[1] Grigoris Dimitroulakos[1] Ioannis Latifis[1]
Konstantinos Masselos[1]

[1] University of Peloponnese, Department of Informatics and Telecommunications, Terma Karaiskaki,
22100 Tripoli, Greece, {lezos,dhmhgre,latifis,kmas}@uop.gr

MATLAB compilation to lower/implementation level languages is performed for application development (e.g. embedded C generation, high-level synthesis to VHDL) and for performance optimization. In this work MAFE, an environment for MATLAB-to-C compilation is proposed. The C code generated by MAFE allocates memory for arrays both statically and dynamically. MAFE's approach for dynamic memory allocation preallocates arrays and a maximum and an imaginary size are assigned to them. The imaginary size changes when a new value is assigned to the array. This way the array size can change dynamically without any reallocation cost. Furthermore, MAFE can optionally generate exception functions that implement runtime checks on the arrays' sizes which is an advantage over Mathworks' MATLAB Coder that infers all array sizes at compile time and does not generate code for execution time size checks. MAFE environment also includes a source code optimizer applying loop and data reuse exploitation transformations. The optimizer supports developers in efficiently applying transformations interactively both at low level (C code) and at high level (MATLAB code). Experimental results prove that the optimizer can: 1) improve execution time of a MATLAB algorithm up to 30% for the C code generated by MAFE and up to 22% for the C code generated by MATLAB Coder, and 2) reduce cache misses up to 31% for the C code generated by MAFE and up to 40% for the C code generated by MATLAB Coder.