BASTION - Armines DEMONSTRATOR

Broadcast And Surveillance TechnologIes Over Networks

General description

Rapid prototyping provides a fast way for engineers to verify the design of their applications especially in signal and image processing. MATLAB is widely used for this purpose, but it requires further processing to achieve good performance, since it is an interpreted programming language rather than a compiled language like C or FORTRAN. Therefore, especially when real-time processing is expected, a methodology along with associated tools should be provided for the main purpose of code acceleration.

Goals / Objectives

In the Bastion project, the main objective of Armines was to design a methodology that details the necessary steps to achieve good performance of Matlab programs : code documentation, profiling, identification of bottlenecks and performance optimisation.

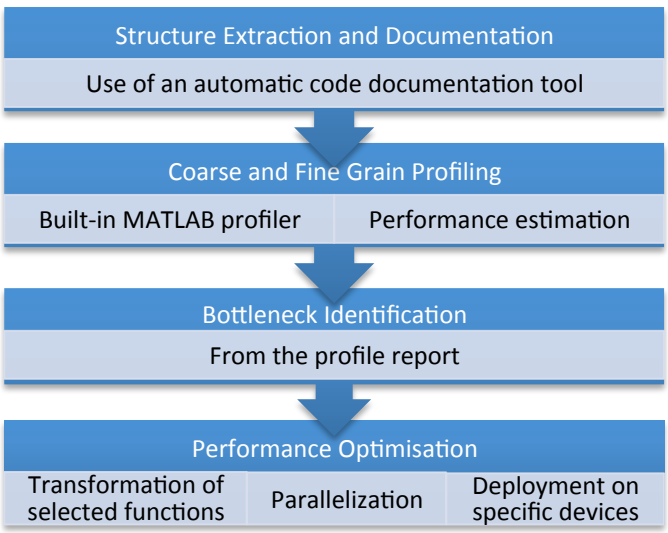
Thus Armines has developed the associated support tools and makes them avalaible to the computer science community, as free licensed softwares.

Finally Armines has investigated and developed a seamless parallel programming features within Matlab and automatic deployment mechanisms on hardware accelerators (GPU and FPGA).

The main use cases have been illustrated on image processing codes from SAGEM.

Results I - Methodology

Given a legacy MATLAB code, we suggest and consider the following steps in order to reach an accelerated version.



Results II - Tools

Regarding the support tools, Armines has implemented the three following tools to assist code optimization activities :

UNIVERSAL REPORT is a high-quality code analysis and documentation software. Its goal is to analyze and generate a structured and well-formatted documentation of a given program.

MatlabMex is a tool that can automatically generate a wrapper for C,C++ or Fortran routines that could then be dynamically linked and used from the MATLAB program.

Matlabcoder is a framework to automatically extract from a given MATLAB code the required information for its efficient conversion using MATLAB Coder.

The MATLAB built-in profiler has been used to instrument the code in order to collect the performance metrics at runtime. These timing information serve during the optimisation steps.

Results III – Optimisation et Parallelisation

Several identified source-to-source transformations can improve MATLAB code such as loop vectorisation, memory pre-allocation, inplace computation, function inlining, data type optimization, change to appropriate array accesses,…

A Pthread based parallelization mechanism for multicore machines has been developed. It gives comparable results as the *parfor* construct provided and managed by MATLAB.

A generic solution through OpenCL functions, compiled into a mex file for direct use within MATLAB, has been proposed for deployment onto GPU too.

Conclusion

Armines has investigated systematic and automatic transformations to accelerate MATLAB prototype codes, considering all kind of devices that have the potential of providing a good level of performance. Regarding the support tools, we have implemented a number of tools to assist code optimization activities. Our tools address each category of computing units: CPU, GPU and FPGA. For the latter, more investigations are expected and the work is ongoing.

Project leader: Mark Verhoeven
Company: Axon Digital Design BV
Email: mark.verhoeven@axon.tv
Tel.: +31 6 51 71 06 82

Project Participants for Armines
Lab : CRI - MINES ParisTech – PSL Research University
Corinne Ancourt, François Irigoin, Claude Tadonki

www.projectbastion.eu
www.eniac.eu