Universidade do Minho

Proposal for Lab Projects on
Evaluation of Tools to Develop Efficient Parallel Applications

1. Introduction / Scientific Background / Related Work

Current development of computationally intensive scientific code must always target parallel computing resources, which range from simple homogeneous multicore devices to multi-server systems (clusters) where each server may be an heterogeneous system with multicore device(s) and accelerator(s) that can be vector oriented (e.g., GPUs), matrix/AI oriented (e.g., NNPs) or spatial oriented (e.g., FPGAs).

Efficient use of these systems often requires some knowledge of the architecture of the computing devices, but also skills to explore alternative approaches to parallel languages and the expertise to adequately take advantage of software tools and guides to aid the development of efficient applications.

These development tools can be supplied by independent providers but they are often available at the device manufacturer website (Intel, AMD, ARM, NVidia, ...). However, most performance engineers trained at UMinho are not taking advantage of these tools and in most cases, this is due to the required time-effort to capture and synthetize the key components in these tools.

This proposal for Lab Projects aims to overcome this limitation and to analyze the rich availability of

- new **Intel** tools for parallel and heterogeneous systems (most of them in the Intel Parallel Studio XE 2020, already installed in the SeARCH cluster); some of these HPC systems are freely available at an Intel cloud, for 120 days;
- a new **Intel** programming model and language for heterogeneous systems to replace CUDA and OpenCL (oneAPI, SYLC, DPC++);
- **ARM** tools for Marvell Thunder X HPC systems (including the Thunder X server in SeARCH cluster).

2. Problem Statement / Goals of the Projects

Projects will be individually defined, customized for the interest of the student group working in the project. Some URL links are provided here to help the groups to understand the scope and opportunities the projects can offer:

- set of slides of an Intel presentation at UPorto in 17-jan-20 (*http://gec.di.uminho.pt/miei/cpd/LEI/IPS-XE2020_Introd.pdf*);
- second set of slides at the same seminar at UPorto (*http://gec.di.uminho.pt/miei/cpd/LEI/oneAPI_EJ_2020.pdf*)
- Intel training videos and other resources for oneAPI (*https://techdecoded.intel.io/topics/oneapi/#gs.vkrfx9*), plus a PDF guide to oneAPI programming (*http://gec.di.uminho.pt/miei/cpd/LEI/oneAPIProgrammingGuide_6.pdf*)
- Intel introduction to Data Parallel C++, DPC++ (*https://techdecoded.intel.io/essentials/dpc-part-1-an-introduction-to-the-new-programming-model/#gs.vkot5i*) and a PDF document, chapters 1 to 4 of an yet unpublished book, (*http://gec.di.uminho.pt/miei/cpd/LEI/DataParallelCppCh1-4.pdf*)
- the SYCL specification, integrating OpenCL devices with modern C++ (*http://gec.di.uminho.pt/miei/cpd/LEI/sycl-1.2.1.pdf*)
- the ARM developer website (*https://developer.arm.com/tools-and-software/server-and-hpc*)
- a guide for tools in ARM64 (*http://gec.di.uminho.pt/miei/cpd/LEI/Best-Practice-Guide-ARM64.pdf*)
- a list of Marvell software partners for ARM Thunder X (*https://www.marvell.com/server-processors/partners.jsp*)

3. The case study

There is currently a scientific application (and associated working code) to address the 2D convection-diffusion problem that often uses a direct method (Gaussian elimination) to solve a system of N linear equations, requiring N^4 operations. This sequential code requires tuning to improve its performance in several platforms, including on servers based on Xeon devices, on a server based on an ARM Thunder X device and on GPUs. Some evaluation work on more updated devices might be able taking advantage of an Intel cloud with heterogenous servers.

This computationally intensive problem can be solved using a more efficient computational method that solves the system of N linear equations in 2N times with N^2 operations each, implemented in two steps, one to solve row by row, the other column by column. Each N^2 operation is fully independent in each step, opening an opportunity to an embarrassingly parallel solution.

This method also explores the way matrices are stored in a computer memory, either in row major (e.g., in C) or column major (e.g., Fortran), by splitting each iteration in two, aka the alternating direction implicit method (ADI). Current vector extensions in processing units require the vector elements to be in contiguous memory locations to avoid performance penalties.

To overcome these limitations several approaches must be proposed and evaluated following the guidelines of the available tools.

A report must be produced with a simple to use guide to develop efficient parallel applications, to be followed by future students in the Master degree in Informatics Engineering (and current performance engineers)

UMinho, 30-jan-20