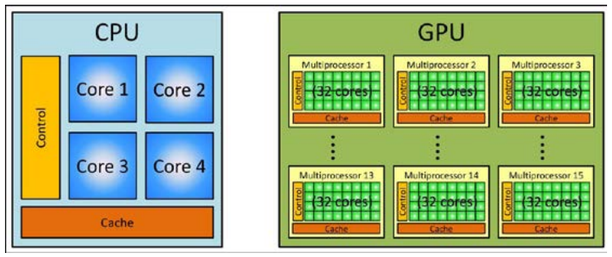
Efficient Pattern-Matching Image Compression on Many-Core Systems

This joint project with Polytechnic Institute of Leiria studied the implementation of high performance image coding algorithms, using many-core platforms, in Graphic Processing Units (GPU). GPU are a common and low-cost solution, with massive parallel computational capabilities (thousands of cores). The use of programming techniques that exploit the full parallelization potential of GPU enable high speed ups in execution times.



PROJECT WEBPAGE URL
https://www.it.pt/Projects/Index/1604

Main Project Team

Nuno Rodrigues (PhD)	**MSP-LR**
Patrício Domingues (PhD)	MSP-LR
Sérgio Faria (PhD)	MSP-LR
Murilo Carvalho (PhD)	DET/UFN Brasil
João Silva (BSc.)	MSP-LR
Tiago Ribeiro (BSc.)	MSP-LR
Rita Silva (BSc.)	MSP-LR
Telmo Marques (BSc.)	MSP-LR

Funding Agencies

FCT - Fundação para a Ciência e Tecnologia PTDC/EIA-EIA/122774/2010	**53,334€**
Start Date	01/01/2012
Ending Date	30/06/2014

Indicators

Journal Papers	1
Conference Papers	2
Conference proceedings (as editor)	2

Two Main Publications

L. Lucas, K.W. Wegner, N. M. M. Rodrigues, C. L. P. Pagliari, E. Silva, S. M. M. Faria, **Intra Predictive Depth Map Coding using Flexible Block Partitioning**, IEEE Trans. on Image Processing, Vol. 24, No. 11, pp. 4055 - 4068, November, 2015

P. Domingues, J. S. Silva, T. R. Ribeiro, N. M. M. Rodrigues, M. Carvalho, S. M. M. Faria, **Optimizing Memory Usage and Accesses on CUDA-Based Recurrent Pattern Matching Image Compression**, Computational Science and Its Applications – ICCSA 2014, Guimarães, Portugal, Vol. 8582, pp. 560 - 575, July, 2014

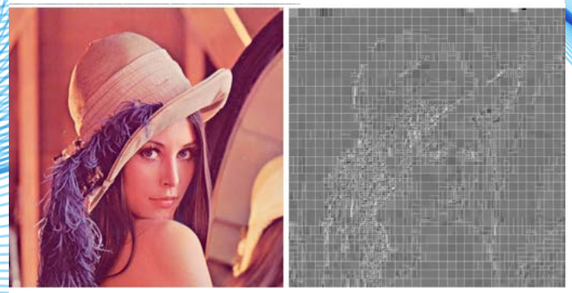


Fig. 1 MMP Segmentation of the Lena image.

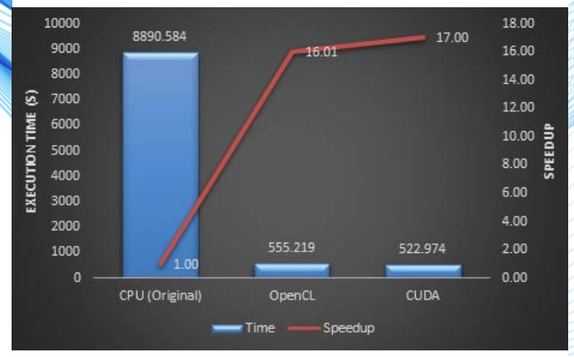


Fig. 2 Execution times and speedup of the sequential and CUDA-based MMP versions.

GENERAL MOTIVATION AND OBJECTIVES

In recent years, the members of the IT research team worked on a new compression paradigm for multidimensional (image and video) signals. The result is the coding method Multiscale Multidimensional Parser (MMP). MMP is an extension of approximate pattern matching since it uses adaptive multiscale dictionaries that contain concatenations of scaled versions of previously encoded blocks. From the compression point of view, MMP presents an excellent performance level. The rate-distortion results for both image and video compression are consistently better than those of the state-of-the-art transform-based methods, like JPEG2000 and H.264/AVC. In spite of their high compression performance, MMP-based methods share the computational complexity issues common to all approximate pattern matching schemes. This results in lengthy execution times, yielding implementations that cannot deliver the speed needed for a daily usage. The objective of this project is to harness the power of GPUs to provide for faster MMP's implementation.

Many-core processors, namely GPU, are revolutionizing computing, bringing high performance computing to the regular personal computer user. Indeed, modern GPU have hundreds of processing cores that can simultaneously execute thousands of threads delivering single floating-point performance above 1 TeraFLOPS for the state-of-the-art models. In average, GPU outperform CPU in floating-point operations by an approximate factor of 20x. Additionally, GPU performance is growing much faster than CPU, and thus the performance gap is rapidly widening towards GPU.

CHALLENGE

The exploitation of MMP's intrinsic parallelism by general purpose GPU is a promising alternative to the previous attempts to reduce the computational complexity associated with CPU-based implementations of MMP. In spite of the relevant gains, these methods have failed to produce an implementation fast enough to be adequate to common use of MMP.

The main challenge of this project is to speed up the MMP coding algorithm by adapting it to GPU-based platforms, assessing at the same time, the strengths and weaknesses of the CUDA and OpenCL platforms. This can be achieved by exploiting several features of the MMP algorithms combined with the development of new techniques that better suit the implementation of MMP in a many-core GPU platform.

WORK DESCRIPTION AND ACHIEVEMENTS

Following the study of the original CPU-based implementation of the MMP algorithm, a study on the computational complexity of each of function was performed by a profiling software, in order to determine the most relevant computational bottlenecks.

The second stage of the project focused on adapting MMP-based algorithms to the specificities of many-core platforms provided by GPU. The objective was to create two versions: a CUDA-based and an OpenCL-based one. The best approach for efficiently offloading the most complex tasks to the GPU, by way of appropriate kernels, was investigated. Another important task was to properly adjust the algorithms to the memory hierarchy of GPUs in order to cope with the hardware features and the differences between CUDA and OpenCL that can impact performance.

During the project, new compression models were developed, that were able to reduce the computational complexity of the algorithm and to have a more suited approach for making efficient use of GPU and hybrid CPU-GPU programming models and to take into account the limitations of the programming models, namely memory accesses.

The optimized versions of MMP encoding the 512 x 512 8-bit gray Lenna image run in 522.974 seconds, for the CUDA version, and 555.219 seconds, for the OpenCL implementation. These values compare against the 8990.584 seconds needed by the sequential version of MMP, which means that the GPU implementation achieved a speedup of 17.191. This demonstrates the ability to achieve meaningful speedup with GPUs even for challenging algorithms such as MMP.