Caravela: A Distributed Stream-Based Computing Platform

Leonel Sousa and Shinichi Yamagiwa
Electrical and Computer Eng. Dep., INESC-ID/IST, TULisbon
R. Alves Redol, 9, 1000-029, Portugal
Email: {yama, las}@ inesc-id.pt
Phone: +351 21 3100210 Fax: +351 21 3145843

Stream-based programming style gathers data into a stream, operates on it and then scatters it back to memory. Recently the computational power growth of Graphical Processing Units (GPUs) exceeds the one established by the Moore's law, so that high level languages and interfaces, such as the Cg and Brook for GPUs (www.gpgpu.org), have been developed to ease the programming of stream processors for general purpose applications. The application of these pieces of work has been limited to compute small-scale problems locally, on a single stream processor. An exception is the case of a cluster of GPUs supported with MPI [1], that directly applies stream computing to the distributed programming model with a very limited set of experiments.

In this workshop we propose a new *flow-model* that extends the basic model for stream-based computing by considering random access to multiple input streams and by supporting recursive computation. The *Caravela* platform presented in this paper defines an execution unit, based on the proposed *flow-model*, that is composed by input data streams, output data streams and a program to process those I/O data streams. This *flow-model* fits particularly well into the GPUs because the GPU supports stream-based computation using texture image inputs. The Caravela package available to download from http://www.caravela-gpu.org allows to apply the *flow-model* unit concept to distributed computing based on GPUs available in recent computers. *Caravela* encapsulates the program in a *flow-model* unit and assigns it to a distributed computing environment, by allowing to execute it in any available computer and by directly collecting the data through the memory or the network [2]. These tools provide a uniform programming interface that hides the differences between graphics runtimes and achieves high performance (DirectX9 and the OpenGL 2.0 are actually supported). Data buffering optimization methods are applied to efficiently implement recursive computation [3].

This paper illustrates the usage of the *Caravela* platform for different applications, such as recursive linear filtering for audio and image processing, 2-D DWT for JPEG2000 and decoding LDPC codes for DVB-S2. Experimental results show that a significant improvement can be achieved with GPUs against general purpose processors and that the implemented swap frame data buffering method allows to improve even more this efficiency, up to 60%, for recursive computation.

Future research directions are pointed to achieve distributed computing based on the Caravel platform, namely the extension of *Caravela* tools to implement the concept of meta-pipelining and its integration in the MPI programming interface.

[1] Z. Fan, F. Qiu, A. Kaufman, S.Y-Stover, “GPU Cluster for High Performance Computing”, *Proc. ACM/IEEE Supercomputing Conference*, 2004, pp. 47-58.

[2] S. Yamagiwa and L. Sousa, “Caravela: A Novel Environment for data-flow-based distributed processing”, accepted for publication in IEEE Computer in January 2007.

[3] S. Yamagiwa and L. Sousa, “Data Buffering Optimization Methods Toward a Uniform programming Interface for GPU-based Applications”, accepted for publication on the ACM International Conference on Computing Frontiers, May 2007.