32nd URSI GASS, Montreal, 19–26 August 2017

GPGPU FDTD: Output of data

Fumie Costen[(1)(2)], Loukas Xanthos*[(1)], Ryutaro Himeno[(3)] and Hideo Yokota[(2)]
(1) School of Electrical and Electronic Engineering, The University of Manchester, United Kingdom
(2) Image Processing Research Team, Center for Advanced Photonics, RIKEN, Saitama, Japan
(3) Advanced Center for Computing and Communication, RIKEN, Saitama, Japan

1 Extended Abstract

General Purpose GPU (GPGPU) devices, also known as accelerators, are gaining popularity in the scientific community for use in High Performance Computing (HPC) environments for the accelerated execution of highly parallelised and vectorised software applications. This is due to the Single Instruction, Multiple Threads (SIMT) architecture of accelerator devices, which makes them more suitable than CPU devices for use with highly parallelised tasks, where the same operation is applied to a large amount of data. As a result of the highly parallelised operation, such software achieve faster execution speeds on accelerators than equivalent software on CPU devices.

Despite the powerful abilities of accelerator devices for parallel processing, the PCI-Express (PCIe) interface constitutes a bottleneck when heavy device-to-host data transfers are performed. In addition, the developers are responsible for correctly utilising the architecture of the accelerator device. Hence, developers have to issue data transfers in their code, in a way that not only optimally exploits the device cache architecture, but also that will not result in a much slower execution time than the case when they transfer a bigger amount of data by correct exploitation of the device architecture.

The Finite Difference Time Domain (FDTD) algorithm can be highly parallelised and vectorised to benefit from execution on accelerator devices. However, its performance can be significantly affected by the large amount of data output it performs at every time step, virtually cancelling out any performance boost provided by the GPGPU acceleration of its computation. To overcome the PCIe throughput limitations, the output of solely the points of interest must by carried out, instead of the whole FDTD space, in an optimal way, with the device architecture taken in consideration, for the avoidance of further latencies. Numerous studies of the development of GPGPU versions of the computation part of the FDTD algorithm have been published. Nevertheless, to the authors' best knowledge, no study has been carried out for the examination of the possible speed up of the data output part of the FDTD simulation on accelerators, when only a subset of the FDTD space is of interest. We developed various approaches to speed up the data output generated on an NVIDIA Kepler K20X accelerator.

In the conference we will illustrate the comparative performance of three methods for the transfer of a single line in the FDTD space from the GPU memory into the system's main memory and three methods for the transfer of a single plane in the FDTD space from the GPU memory into the system's main memory. Moreover we will assess the performance of these methods when used for the device-to-host transfer of several, not necessarily consecutive, lines or planes. Furthermore we will present three techniques for the transfer of points at random locations in the FDTD space and examine their comparative performance.