Technology Strategy Board
Driving Innovation

Knowledge Transfer Network
Industrial Mathematics

A step change in speed

REINSURANCE STRATEGY OPTIMISATION

Willis Re / King's College London

The need

Reinsurance provides protection for insurance companies against the risk of losses including from catastrophic events such as earthquakes and floods. A reinsurance contract is drawn by dividing the range of losses into a number of layers, and each layer has a placement associated with it. This is the level of protection offered if the loss were to fall in that particular layer. A key question the client must ask is, "which set of placements is the optimal choice for me?"

To answer such a question requires calculating the contract premium for all possible variations of placement values in each layer and calculating some key statistics on the result. The objective of this project is to optimise this calculation using Graphical Processing Units (GPUs).

The outcomes

The intern created three prototypes during the project. The first, is a multi-threaded C++ application using only the processing cores available in the Central Processing Unit (CPU) of a single computer. The second, solved the problem using standard libraries available using the Graphics Processing Unit (GPU). The third is an optimised version of the second application.

		Execution Time (in seconds)		
m	$\|S\|$	RROptimiser1 (Quad Core AMD 3.2GHz Sempron)	RROptimiser2 (CUDA GTX280)	RROptimiser3 (CUDA GTX280)
3	1331	3.2	10	0.39
4	14641	13.8	30	1.16
5	161051	118	360	9.55
6	1771561	1359	4120	105.58

In a previous internship project, an intern created a grid computing solution which solved an example problem in 2 minutes using 36 computers without the use of a GPU. The same problem took 50 minutes on one computer.

As a result of the current project, it now takes 9 seconds to compute on a single computer with the use of a GPU which costs as little as £100. A fast tool was required since timetables for certain decisions about reinsurance placements for their clients can be measured in days and this is now available for deployment in Willis Re.

"…to our surprise, [at first] there was very little speed improvement when using available library algorithms. Asad analysed the problem and implemented his own solution, coded in CUDA, and achieved a 10-fold increase in execution speed. Great result."

Jürgen Gaiser-Porter, Willis Re

Technical summary

The very nature of the problem exhibits a large amount of data parallelism; that is it may be divided into several sub problems, which can be solved independently, making GPUs ideal candidates to solve the problem.

In 2007 NVIDIA released CUDA (Compute Unified Device Architecture). This is a low level programming language similar to C used to program general purpose graphics processors (GPGPUs), much like C/C++ are used to program CPUs. GPGPUs however have a completely different design philosophy. They were designed to compute thousands of similar calculations for each pixel on the screen. Through CUDA GPU's can be used to solve more generic problems.

To solve the problem at hand three prototypes were created:

- **RROptimiser1:** In order to monitor the possible performance benefits of CUDA it was natural to start with a benchmark. This was created as a multithreaded C++ application utilising the BOOST and ATLAS libraries.
- **RROptimiser2:** Although CUDA is a low-level language used to program GPGPUs, several high-level libraries such as THRUST and CUBLAS have been written to access the functionality of the GPGPU without the need to write low-level code. From a maintenance point of view it seemed logical to exploit the power of these libraries next. Unfortunately, although these libraries are good at solving one large problem in parallel, they are not designed to solve several small problems in parallel, which was the motivation of the next prototype.
- **RROptimiser3:** An application written in CUDA and C++. This program computes several (approximately 4000) reinsurance result vectors in parallel, making it superior to the previous approaches. The high-level libraries mentioned above do not accommodate this approach.

The fastest of the three was RROptimiser3 which was 20 times faster than the multi-threaded, CPU optimised code.

"This project proved to be a great CUDA exercise, a tool that I no doubt will need in my further research. CUDA has opened a whole new world of scientific computation for me."

Asad Munir
King's College London

"This project highlighted the strong potential of GPUs for accelerating compute-intensive search methods for optimization. It has given me the confidence to explore GPU implementations of related problems in asset allocation, and was a good example of King's College collaborating with its neighbours in the financial industry. Asad Munir did a great job."

William Shaw
King's College London

This project was part of the programme of industrial mathematics internships managed by the Knowledge Transfer Network (KTN) for Industrial Mathematics. The KTN works to exploit mathematics as an engine for innovation. It is supported by the Technology Strategy Board, in its role as the UK's national innovation agency, and the Engineering and Physical Sciences Research Council, in its role as the main UK government agency for funding research and training in engineering and the physical sciences.

EPSRC
Engineering and Physical Sciences Research Council

Project Details

Partners
Willis Re
King's College London

Project investment
£10,000

Intern
Asad Munir

For further details on the technology:
Jürgen Gaiser-Porter
Willis Re
gaiserporterj@willis.com

For further information on internships and other collaborations:
Lorcán Mac Manus
Industrial Mathematics KTN
lbmm@industrialmaths.net
+44 (0) 1483 579108