Thalesian workshop on GPU computing

Suggested laboratory exercises

Claudio Albanese

Download OPLib from

http://www.albanese.co.uk/oplib/OPLib.zip

and unzip the tarball.

If you have Visual Studio 2005 or 2008, just open the project and compile.

If you are on a Linux workstation, you need to compile two libraries: libopc.so and libopcuda.so. In the directories OPLib/oplib/opc and OPLib/oplib/opcuda there are sample make files, one for a 32 bit Linux system and another for a 64 bit Linux system. Choose one and edit it if needed.

Once the native libraries are compiled, they are deposited in the directory OPLib/bin which contains precompiled assemblies for the benchmarks. Assemblies include:

betest.exe: a simple Black Scholes test

cdgemm.ese: a benchmark for the double precision matrix multiplication routine dgemm CPU side

clv.exe: a benchmark for a local volatility Monte Carlo generator CPU side

cmt.exe: a benchmark for the Mersenne Twister algorithm CPU side

csgemm.exe: a benchmark for the single precision matrix multiplication routine sgemm CPU side

csv.exe: a benchmark for a stochastic volatility Monte Carlo generator CPU side

glv.exe: a benchmark for a local volatility Monte Carlo generator CPU side

gsgemm3.exe: a benchmark for ordinary (third level) sgemm GPU side

gsgemm4.exe: a benchmark for the fourth level extension sgemm4 of the matrix multiplication routine on GPUs

gsgemv2: a benchmark for ordinary (second level) sgemv GPU side

gsgemv4: a benchmark for sgemv4 GPU side, the fourth level extension of sgemv

gsv.exe:a benchmark for a stochastic volatility Monte Carlo generator GPU side

To run the benchmarks and tests on Linux, use the mono JITter by entering the command mono clv.exe

Notice that this will work only if you correctly set the environment variable LD_LIBRARY_PATH to point to the directories with the Intel MKL libraries and the CUDA libraries, as all linkages in shared objects are dynamic.

More advanced tasks would be the following:

*) Open the visual studio solution project with monodevelop

*) Edit the C# code for OPPricer.cs to run the test using GPUs in single precision as opposed to CPUs.

*) Help fix a bug! All benchmarks work well under Windows 32 and 64 bit and under Linux 32 bit. However, under Linux 64 bit the benchmarks glv.exe and gsv.exe fail as mono cannot find all the dependencies of the library libopcuda.so.

Take home assignments:

*) Build benchmark programmes hlv.exe and hsv.exe which are hybrid in the sense that kernels are obtained GPU side while Monte Carlo scenario generation is carried out CPU side.

*) Reproduce the results on the likelihood ratio method in the paper “Monte Carlo Pricing using Operator Methods and Measure Changes” I co-authored with Hongyun Li.