Banking on Monte Carlo ... and beyond
Agenda

- Introduction
- What’s the problem?
- GPUs – an opportunity?
- NAG’s research/experience/feedback
- Real-world use: Monte-Carlo and beyond
- Next steps
- Summary
NAG Background

- Founded 1970
  - Co-operative software project
  - Not-for-profit organisation
  - Surpluses fund on-going research
- ~$12m financial turnover
- ~100 employees
  - ~65% developers/technical consultants
  - Oxford (HQ), Manchester, UK; Chicago, USA; Tokyo, Japan; Taipei, Taiwan
NAG Products & Services

- Numerical and Statistical Libraries
  - Over 1600 user-callable components

- Consulting Services
  - Code development, tuning, tailoring

- HPC Services
  - Procurement advice, market watch, benchmarking
  - Computational Science and Engineering (CSE) support

- Experts in Numerical Engineering
What Happened to my Escalator?

- Escalator?
  - Want a quicker solution? Buy a new processor
- Multi-core/Many-core are a major challenge for many existing codes
- The escalator has stopped... or gone into reverse!
  - Existing codes may well run slower
What Can We Do?

- There is no “silver bullet”
  - (In most cases)
  - We’ve passed the end of this escalator
- It’s the software stupid!
  - Need to re-write/re-tune the software for new hardware
  - But which hardware?
- GPUs offer an interesting solution for some key applications
  - NVIDIA clearly lead the way with CUDA
  - OpenCL?/AMD?/Intel?
GPUs – An Opportunity?

- Large-scale SIMD/SIMT
  - simplified logic so more of the chip for calculations
- Excellent bandwidth to the GPU memory
- $O(10)$ power savings [BNP Paribas]
- Good programming environment with CUDA
  - And hopefully OpenCL for portability
- Can work well for embarrassingly parallel applications
GPUs in Computational Finance?

- Ovum report (August 2010)
- Lots of POCs – almost all with NVIDIA
- Monte Carlo, Finite Differences, Differential Equations
- Adopt CUDA or wait for open standard?
- Serious competition in 2012 (AMD/Intel)
Monte Carlo Methods

- Often used when infeasible/impractical to use a deterministic method
  - Take random samples of the input domain
  - Perform deterministic calculations based on the random inputs
  - Aggregate the results
- The more samples and the more ‘random’ the better
- Embarrassingly parallel (except RNGs!)
- Speed matters
Early Market Pull

- NAG closely monitors the HPC marketplace
  - Enforced change painful
  - Many technologies being evaluated

- NAG’s product implementation teams
  - Finance sector showing particular interest (POCs)
  - Monte Carlo methods particularly important ... but other areas now under investigation (e.g. PDEs, optimisation)

- NAG GPU Library (beta)
  - Worked closely with Prof Mike Giles, Oxford University
  - RNGs and distributions
  - PDEs ... very soon
Early Successes (last year)

- BNP Paribas
  - NAG mrg32k3a works well in BNP Paribas CUDA “Local Vol Monte-Carlo”

![Graph showing speed-up of NAG MRG32k3a/GX260 versus BNPP CPU version.](image-url)
Latest Successes

- (Almost) all tier 1’s have POCs running
- Some close to going live on early projects
- E.g. Barclays Capital ...
  - “Thank you for the GPU code, we have achieved speed ups of x120”
Focus of this presentation is a credit risk loss simulation

Why is a simulated approach taken?

Complex portfolio dynamics
- small probabilities of default (PD)
- large portfolios $O(10^6)$
- inter-dependence through default correlation
- highly non-linear behaviour

Analytical approach
- restrictive assumptions
- limited application

How many simulations are required?

Estimate $O(10^9)$ simulations required
BENCHMARKING: GPU VS. CPU ARCHITECTURE

- GPU Tesla C1060 vs. single core CPU

- Speed-up:
  - GPU vs. Hi-Performance CPU $108 \times$
  - GPU vs. Standard CPU $787 \times$

- Time to compute $10^9$ simulations
  - Standard CPU would take around 2 months
  - Hi-Performance CPU would take over a week
  - GPU would take 2.5 hours
  - GPU (4 $\times$ ) server rack less than 40 minutes

- CPU optimisation can offer significant gains
Is Monte Carlo the Answer?

- Not ‘the’ answer, but...
- ...given these speed-ups perhaps it can be used much more?
- Good list of application areas on Wikipedia
  - en.wikipedia.org/wiki/Monte_Carlo_method#Applications
- In general, we need to be re-thinking:
  - How we solve problems - new (or old!) algorithms
  - Which techniques work best on which architectures
- Acid test
  - How well can it work for my application?
Next Steps

- **NAG GPU Library**
  - Currently in beta, but pressure to productise
  - RNGs/distributions/ Brownian bridge; PDEs – very soon
  - Which other algorithms do we need to implement?

- **NAG Libraries (1600+ components)**
  - Should we implement on CPU calling out to GPU?
  - ‘Automatic’ cross-compilation
  - SMP implementations on multi-core CPU also works well

- **Algorithms**
  - Collaborating widely to look at new algorithms for new architectures
NAG GPU Lib: Improvements and Issues

- **Updated RNGs**
  - Mersenne Twister (with skip-ahead)
  - Scrambled sequencing for Sobol (Hickernell)
  - Tuned for Fermi (next slide)

- **Implementing PDEs**
  - ADI/FD with Crank-Nicolson, Craig-Sneyd
  - Challenges because of lack of cache ...
  - Fermi implementation 15-20x CPU version

- **Main issue for mainline product**
  - Need to be able to allow GPU only (device level) functions but NOT have to supply source!
RNG Performance Numbers

- From GEMS report (to be published soon)
  - Intel figures tuned by Intel

<table>
<thead>
<tr>
<th></th>
<th>Fermi GPU (pts/ms)</th>
<th>Intel MKL on Xeon E5410</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Thread</td>
<td>2 Threads</td>
</tr>
<tr>
<td>MRG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unif</td>
<td>dp</td>
<td>7.71E+06</td>
</tr>
<tr>
<td></td>
<td>sp</td>
<td>7.45E+06</td>
</tr>
<tr>
<td>Exp</td>
<td>dp</td>
<td>5.44E+06</td>
</tr>
<tr>
<td></td>
<td>sp</td>
<td>2.67E+06</td>
</tr>
<tr>
<td>Norm</td>
<td>dp</td>
<td>4.61E+06</td>
</tr>
<tr>
<td></td>
<td>sp</td>
<td>2.44E+06</td>
</tr>
<tr>
<td>Sobol</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unif</td>
<td>dp</td>
<td>1.74E+07</td>
</tr>
<tr>
<td></td>
<td>sp</td>
<td>1.35E+07</td>
</tr>
<tr>
<td>Exp</td>
<td>dp</td>
<td>7.94E+06</td>
</tr>
<tr>
<td></td>
<td>sp</td>
<td>3.21E+06</td>
</tr>
<tr>
<td>Norm</td>
<td>dp</td>
<td>8.60E+06</td>
</tr>
<tr>
<td></td>
<td>sp</td>
<td>1.62E+06</td>
</tr>
</tbody>
</table>
Summary

- Difficult/exciting times for all
- Exciting developments on NVIDIA GPUs – getting better all the time
- NAG is actively involved in R&D in this area and has beta software available
- NAG is seeking feedback on further areas of interest from the community

Thank You

ian.Reid@nag.co.uk  www.nag.co.uk/numeric/gpus
Acknowledgements

- Professor Mike Giles, Oxford University
  - CUDA code
  - Expertise
  - Unbounded enthusiasm

- Technology Strategy Board (TSB), Smith Institute
  - Project funding

- NVIDIA
  - Hardware support and general interaction