Computational Finance on GPUs

Mike Giles

mike.giles@maths.ox.ac.uk

Oxford-Man Institute of Quantitative Finance

Oxford University Mathematical Institute

Oxford e-Research Centre

NVIDIA CUDA Fellow for Computational Finance

Intel Finance Forum, October 8th, 2009

Opportunity

- CPUs have up to 6 cores (each with a SSE vector unit) and 10-30 GB/s bandwidth to main system memory
- NVIDIA GPUs have up to 30×8 cores on a single chip and 100+ GB/s bandwidth to graphics memory
- offer 50–100$\times$ speedup relative to a single CPU core
- roughly 10$\times$ speedup relative to two quad-core Xeons (if not using SSE instructions)
- also 10$\times$ improvement in price/performance and energy efficiency

How is this possible? Logically simpler cores (SIMD units, no out-of-order execution or branch prediction, minimal caching) for vector computing, not general purpose

Opportunity

Is this GPU advantage sustainable? Yes!

- IBM, AMD and Intel all producing GPUs too
- NVIDIA has a good headstart on software side with CUDA environment
- new OpenCL software standard (based on CUDA and pushed by Apple) will probably run on all platforms
- driving applications are:
  - computer games “physics”
  - video (e.g. HD video decoding)
  - computational science
  - computational finance
  - oil and gas

Why GPUs will stay ahead

Technical reasons:

- SIMD units means larger proportion of chip devoted to floating point computation (but CPUs will respond with longer vector units – AVX)
- tightly-coupled fast graphics memory means much higher bandwidth

Commercial reasons:

- CPUs driven by price-sensitive office/home computing; not clear these need vastly more speed
- CPU direction may be towards low cost, low power chips for mobile and embedded applications
- GPUs driven by high-end applications – prepared to pay a premium for high performance

Use in computational finance

- Bloomberg has a large cluster:
  - 48 NVIDIA Tesla units, each with 4 GPUs
  - alternative to buying 2000 CPUs
- BNP Paribas has a small cluster:
  - 2 NVIDIA Tesla units
  - replacing 250 dual-core CPUs
  - factor 10x savings in power (2kW vs. 25kW)
- lots of other banks doing proof-of-concept studies
  - my impression is that IT groups are keen, but quants are concerned about effort involved
- Several ISV's now offer software based on CUDA

Programming

Big breakthrough in GPU computing has been NVIDIA's development of CUDA programming environment

- C plus some extensions and some C++ features
- host code runs on CPU, CUDA code runs on GPU
- explicit movement of data across the PCIe connection
- very straightforward for Monte Carlo applications, once you have a random number generator
- significantly harder for finite difference applications (but will be much easier with next-generation GPU)
- see example codes on my website

My experience

- Random number generation (mrg32k3a/Normal):
  - 2500M values/sec on GTX 280
  - 70M values/sec/core on Xeon using Intel's VSL
- LIBOR Monte Carlo testcase:
  - 180x speedup on GTX 280 compared to single thread on Xeon
- 3D PDE application:
  - factor 50x speedup on GTX 280 compared to single thread on Xeon
  - factor 10x speedup compared to two quad-core Xeons

GPU results are all single precision – double precision is currently 2-4 times slower, no more than factor 2 in future

Programming

Software alternatives:

- OpenCL
  - no personal experience
  - looks similar to the lower-level CUDA device API
  - I’m waiting for simpler higher-level layer, and to hear from others on pros/cons versus CUDA
  - will probably start using it within a year or so

Programming

Software alternatives:

- Microsoft's DX Compute
  - unlikely to be used for scientific computing, but maybe for games and multimedia applications
- Intel: Ct, TBB, SSE/AVX vectors, `icc`, OpenCL
  - I find range of alternatives confusing – look to Intel for clear guidance on pros and cons
  - I think SSE/AVX vectors may offer best performance but programming is tedious (worse than CUDA?)
  - I hope OpenCL support is good (should map very naturally to SSE/AVX vectors)

Current developments

NVIDIA: new GPUs just announced, with OpenCL support

AMD: new GPUs out now – OpenCL support coming soon

IBM: Cell hard-to-use – terminating future development?

Intel:

- Larrabee GPU out soon, 2nd-gen in another year for computational applications?
- Ct and TBB preferred software approach, but will support OpenCL?
- Also watch AVX vectors for mainstream CPUs, but performance limited by available bandwidth?

Current developments

Supermicro, HP: 1U / 2U servers with built-in GPUs

IBM: planning a GPU blade solution (in addition to Cell)

Dell: “personal supercomputer” with up to 3 GPUs

Portland Group: developing additional compiler support for CUDA – may extend it to OpenCL and target other back-ends in the future?

What is needed now?

Skilled manpower, training:

- 50+ on Oxford CUDA mailing list: students and post-docs in almost all science departments
- 1-week CUDA course this summer
- in 3 years time, many PhDs in computational science will have these skills

More development of libraries, high-level packages:

- Random number generation (NAG)
- Monte Carlo simulation
- PDE solvers
- …

Further information

LIBOR and finite difference test codes
`www.maths.ox.ac.uk/~gilesm/hpc/`

NAG Numerical Routines for GPUs
`www.nag.co.uk/numeric/GPUs/`

Intel's Larrabee and AVX
`software.intel.com/en-us/articles/larrabee/`
`software.intel.com/en-us/avx/`

NVIDIA's computational finance page
`www.nvidia.com/object/computational_finance.html`

Supermicro's GPU servers
`www.supermicro.com/GPU/`