Accelerating Quantitative Financial Computing with CUDA and GPUs

NVIDIA GPU Technology Conference
San Jose, California
March 20, 2013

Gerald A. Hanweck, Jr., PhD
CEO, Hanweck Associates, LLC
Agenda

- Why GPUs in Quant Finance?
- Overview of GPU Technology
- GPU Quant Finance Case Studies
  1: Real-Time Option Analytics
  2A: Stochastic Volatility Modeling
  2B: Stochastic Volatility + Jumps Modeling
  3A: Large-Scale Interest-Rate Swaps Value-at-Risk (VaR)
  3B: Large-Scale Monte Carlo VaR
  3C: Large-Scale Parametric VaR
  4: Pricing a Basket Barrier Option
  5: Random Number Generation

- Concluding Remarks

March 20, 2013
Q: What Is the Biggest Problem Facing the Capital Markets Today?

A: *Intraday and Real-Time Risk Management*

- Increasingly complex, global structured products
- Higher correlation risk and systemic risk
- Greater regulatory requirements
- Massive grid-computing infrastructure costs
- Exploding market-data message rates
GPU Acceleration in Quant Finance

- **10x faster** “dollar-for-dollar” than conventional CPU computing.

- **10x faster** means:
  - overnight → over lunch
  - over lunch → get a cup of coffee
  - get a cup of coffee → don’t blink!

- Better risk management.

- Reduce total cost of ownership and infrastructure cost explosion.
GPUs in Quant Finance

Source: NVIDIA

March 20, 2013
Hanweck Associates’ Volera™ GPU-Accelerated Product Line

VoleraFEED™
Real-time, low-latency datafeed of options implied volatilities and greeks covering global markets, powered by Hanweck Associates’ Volera™ GPU-accelerated engine. VoleraFEED powers:

ISE Implied Volatility & Greeks Feed™

NYSE Technologies. Options Analytics™

VoleraRISK™

Premium Hosted Database™
Hosted historical and real-time “tick-level” database service of equity and options prices and analytics, with 300+ TB of data stored in an enterprise-scale cloud. “Data-and-Analytics-as-a-Service” paradigm.

In partnership with

Options Volatility Service™
Historical, end-of-day options analytics database covering more than 6,000 U.S. companies over the past 12 years.

In partnership with

March 20, 2013
NVIDIA GPU Performance

Source: NVIDIA
NVIDIA GPU Architecture

2,688 CUDA cores*
3.95 Tflops single-precision
1.31 Tflops double-precision
6 GB ECC DRAM
250 GB/sec DRAM bandwidth
64 KB of RAM for shared memory and L1 cache (configurable)

* Tesla K20X series GPU
Case Study #1: Real-Time Options Analytics

Real-time, low-latency implied volatilities and Greeks (binomial tree)

Hanweck Associates’ VoleraFEED™ Real-Time Options Analytics Engine

- real-time, low-latency implied volatilities and Greeks
- U.S. OPRA universe:
  - 530,000 options on 3,800 stocks
  - 2012: 4.6 million msg/sec peak
  - 2014: 15.1 million msg/sec peak*
- 128-step CRR binomial tree
  - discrete dividends (escrowed)
  - discount and borrow curves
  - bid/ask/mid implieds & mid Greeks

Average Chain Calculation

20 milliseconds**

* OPRA Jan 2014 projection
** 4 NVIDIA Kepler K10s
Case Study #2A: Stochastic Volatility Modeling

Price options under the Heston* stochastic volatility model:

- European-style call and put options
- Solution involves numerical integration of complex-valued integrands for each distinct strike and expiry
- Simpson’s rule with dynamic integration ranges
- Hardware: NVIDIA C2070 GPU vs. Intel Xeon E5640 (2.67 GHz)

Option Pricing under Stochastic Volatility:

2,000 option pricings per second
(70x faster than a single CPU core)

Price options under the Bates* stochastic volatility+jumps model

- European-style call and put options
- Solution involves FFT integration of the characteristic function for each expiry across range of strikes**
- NVIDIA C2090 GPU, cuFFT 4.0

Option Pricing under Stochastic Volatility + Jumps:
1,200 expiry pricings per second (double precision, $2^{15}$ nodes)


Case Study #3A: Large-Scale Interest-Rate Swaps Risk

Calculate Value-at-Risk (VaR) for a large-scale portfolio of interest-rates swaps (IRS):

- 30,000 distinct IRS positions.
- 1,300 Monte Carlo paths representing yield-curve shocks.
- Full cash-flow and day-count revaluation in each path.
- Calculation of VaR and expected shortfall.
- Hardware: 1 NVIDIA C2090 GPU w/ 8-core Xeon host server.

Large-Scale, IRS VaR:

10 seconds
(vs. 45 minutes on a CPU-based compute grid)

March 20, 2013
Case Study #3B: Large-Scale Monte Carlo Risk

System for real-time risk monitoring of large portfolios of listed options:

- 350,000 distinct options representing the listed universe.
- 10,000 Monte Carlo paths generated from factor shocks (2,500 factors) on 3,500 underlying stocks and indices.
- Hundreds of large portfolios.
- Full binomial-tree revaluation of each option in each path.
- Calculation of VaR and expected shortfall under multiple correlation scenarios.
- Hardware: 24 NVIDIA C2050 GPUs w/ 8-core Xeon host servers.

Large-Scale, Full-Revaluation Monte Carlo VaR:
< 1 minute
(hundreds of times faster than a single CPU core)
Case Study #3C: Large-Scale Parametric VaR

System developed for a large investment bank to evaluate parametric factor VaR on millions of private client portfolios, with aggregation across accounts, advisors, offices and regions:

- 1.25 million portfolios
- 2,000 factors covering 400,000 global assets
- Hardware: 12 NVIDIA C2050 GPUs w/ 8-core Xeon host server

Large-Scale Parametric Factor VaR:
2 minutes
(hundreds of times faster than a single CPU core)
Case Study #4: Basket Barrier-Option (Monte Carlo)

Valuing a basket barrier-option

Monte Carlo simulation of a multi-factor, local-volatility model for pricing lookback structures:
- 4 underlying assets
- 100,000 MC paths
- 750 steps per path

<table>
<thead>
<tr>
<th>Stage</th>
<th>CPU Time (sec)</th>
<th>GPU Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 1: RNG</td>
<td>20.2</td>
<td>0.139</td>
</tr>
<tr>
<td>Stage 2: Path Gen</td>
<td>23.5</td>
<td>0.181</td>
</tr>
<tr>
<td>Stage 3: Payoffs</td>
<td>8.5</td>
<td>0.223</td>
</tr>
<tr>
<td>Stage 4: Stats</td>
<td>9.3</td>
<td>0.061</td>
</tr>
<tr>
<td>Total</td>
<td>61.9</td>
<td>0.604</td>
</tr>
<tr>
<td>Performance Gain</td>
<td></td>
<td><strong>102x faster</strong></td>
</tr>
</tbody>
</table>

Realistic “dollar-for-dollar” performance gain: **12x faster**

1. One core of Intel Xeon L5640 @ 2.26GHz
2. One NVIDIA Fermi C2070 GPU
3. Performance adjusted for: core/GPU density, amortized hardware costs, power/cooling costs, etc.
Case Study #5:
Random Number Generation

Implementation of a GPU-parallel Monte Carlo simulation and random-number generator for a major investment bank:

- Monte Carlo simulation of a multi-factor, local-volatility model for pricing lookback structures
- Implementation of an efficient GPU-parallel random-number generator*
- Hardware: 1 NVIDIA C2070 GPU w/ 8-core Xeon host server

GPU-Parallel Monte Carlo:
2.5 billion normal random numbers per second
(200x faster than a single CPU core)

Random-Number Generation

Large base of existing GPU code and resources:

NVIDIA’s cuRAND RNG library
http://developer.nvidia.com/curand
  • L’Ecuyer (MRG32k3a), MTGP Mersenne Twister, XORWOW PRNG and Sobol QRNG

NVIDIA’s CUDA SDK sample code:
  • Niederreiter, Sobol QRNGs, Mersenne Twister
  • Monte Carlo examples

GPU Gems 3 and GPU Computing Gems (Emerald Edition)
GPU Gems 3 is available online: http://developer.nvidia.com/object/gpu-gems-3.html
  • Tausworth, Sobol and L’Ecuyer (MRG32k3a)
  • Monte Carlo examples (GPU Gems 3)
Concluding Remarks

Real-time and intra-day risk management is a major problem facing the financial industry today... but it is pushing conventional computing to its limits.

GPUs are the way forward. Major financial institutions are using them for quant finance.

Performance gains of more than 10x – dollar for dollar – are achievable in practice in many common use cases, which is generally sufficient to offset the costs of new development.

GPU programming in general – and CUDA in particular – push developers to parallelize their code.

Parallelizing quant finance is critical if quant finance software is to take advantage of the advances in many-core hardware.