SciFinance®

Parallel Computing with SciFinance®

By harnessing the power of NVIDIA GPUs or multi-CPU workstations, SciFinance parallel codes for Monte Carlo pricing models run blazingly fast. SciFinance GPU-enabled codes achieve astounding acceleration of up to 30X-220x. SciFinance OpenMP-compliant codes yield near linear acceleration on multi-CPU workstations.

SciFinance is a code synthesis technology for building derivatives pricing and risk models. SciFinance automatically produces C/C++ pricing model source code from concise, keyword-rich pricing model specifications. SciFinance supports the modeling of any derivative instrument that can be valued using PDE or Monte Carlo techniques. At a keystroke, quantitative developers at financial institutions around the world enjoy easy access to state of the art pricing model techniques and methodologies with the comfort that the pricing model source code is open to inspection and editing. Now, SciFinance provides the same easy access to parallel computing code synthesis for Monte Carlo models, including complex processes, path dependencies and Bermudan exercise/callable structures.

Manual programming for parallel systems is arcane. The code may be difficult to debug and full performance hard to achieve. But now, SciFinance users need only add a single keyword to a pricing model specification to generate parallel pricing model C/C++ source code. Code is compiler ready with an identical pricing function argument list to serial version and numerical test results within the MC variance. No programming and blazingly fast performance.

GPU

NVIDIA GPU-enabled parallel code executes critical sections of the pricing model source code on the graphics card, taking advantage of its highly parallel architecture. By simply adding the keyword "CUDA" to a model specification, SciFinance synthesizes GPU-enabled codes that demonstrate astounding accelerations: 30X-220X faster than serial codes on a standard PC. One PC equipped with several inexpensive NVIDIA GPU cards can replace many racks of blades with a single box, with commensurate reduction in both footprint and energy consumption.

The SciFinance-NVIDIA GPU solution offers marked efficiency improvements for both trading and risk management functions.

- SciFinance significantly reduces the time and cost of developing derivative pricing and risk models
- SciFinance automatically generates GPU-enabled Monte Carlo pricing and risk models run 30X-220X faster than serial code. No CUDA or parallel computing expertise is required.
- NVIDIA GPU provides at least an order of magnitude greater application performance per dollar than traditional solutions based solely on CPUs.

OpenMP

Nearly all modern desktop computers have multiple CPUs, usually from two to eight, while workstations may have many more. SciFinance users can take advantage of this power by simply adding the keyword "OpenMP" to a model specification. The synthesized parallel code is compliant with the OpenMP standard and with existing Windows and Unix compilers. It executes in the multi-processor environment with nearly linear speed-up, e.g. a factor of 3.9X on a quad-core PC or 22X on a 24 CPU workstation.

SciComp Inc.
www.scicomp.com

SciFinance GPU/OpenMP Case Studies

Case Study I: Equity Linked Note

The Note is linked to a basket of N indices. If, on semi-annual observation dates, the performance of all indices is above the knock out barrier for that date, the Note redeems early at par plus a bonus coupon. If early redemption does not occur, then at maturity either i) the Note redeems at par plus a maturity coupon, or ii) if the performance of at least n of the indices have even been below the knock in barrier during the tenor, on a continuously observed basis, then the Note redeems at the performance percentage of the worst index.

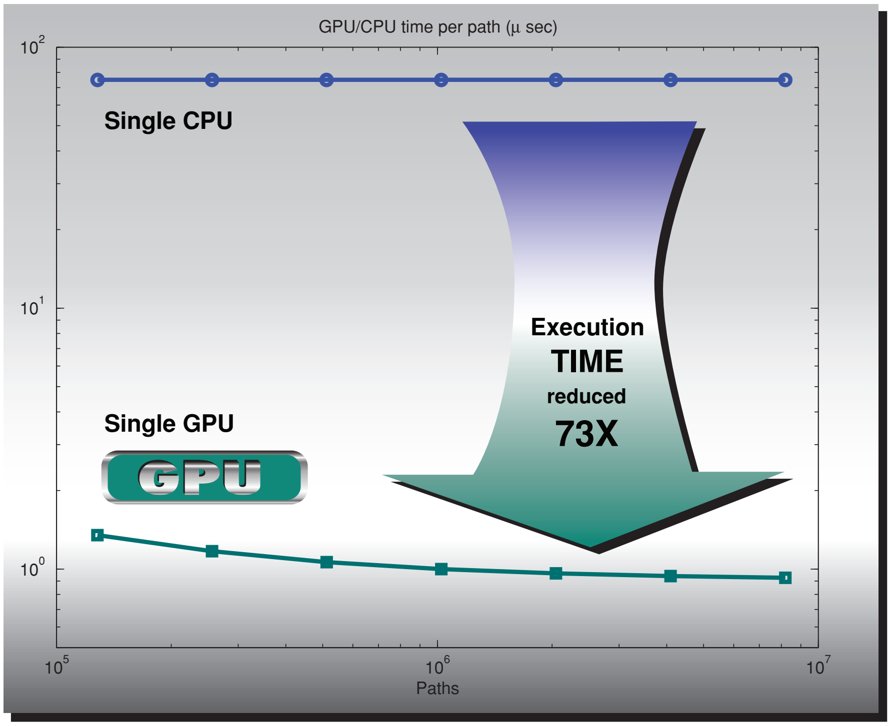
Pricing Model

The Heston stochastic volatility model is used for each index, including cross correlation of index levels and volatility of variance among indices. The paths are constructed with quasi-random (Sobol sequence) numbers and Brownian bridge scrambling. Continuity corrections (again via a Brownian bridge) convert discretely monitored knock-in barrier observations to continuous ones.

Timing

N=6 indices, n=3 required for knock-in, one million quasi-random paths, each of 16 time steps per year over a maturity of 3 years, requiring a 864-dimensional Sobol sequence. Standard deviation of computed Note PV is is 0.01%. (All timings on an Intel Xeon E5405 2.0GHz CPU running Windows XP with a NVIDIA® Tesla C1060.)

	OpenMP	GPU	GPU
Serial	**OpenMP (quad-core PC)**	**Single GPU**	**Dual GPU**
75.0 seconds	**19.0 seconds**	**1.03 seconds**	**0.69 seconds**
	(x 3.94)	**(x 73)**	**(x 109)**



SciFinance GPU/OpenMP Case Studies

Case Study II: FX Accumulator

This simple accumulator contract pays at expiration the forward FX rate X multiplied by the discretely monitored occupation fraction:

$$X * \text{fraction of monitor dates on which } XMin \le X \le XMax$$

Where XMin and XMax can be different for every Monitor Date.

Pricing Model

We price the contract under the double Heston model described by Kainth and Saravanamuttu, ‘Modeling the FX Skew’ (available online).

$$dX/X = (rd-rf)\,dt + \mathrm{sqrt}(v1)\,dW1 + \mathrm{sqrt}(v2)\,dW2$$

$$dv1 = kappa1\,(theta1-v1)\,dt + sigma1\,\mathrm{sqrt}(v1)\,dZ1$$

$$dv2 = kappa2\,(theta2-v2)\,dt + sigma2\,\mathrm{sqrt}(v2)\,dZ2$$

$$\langle dW1\ dZ1\rangle = rho1\,dt$$

$$\langle dW2\ dZ2\rangle = rho2\,dt$$

$$\langle dW1\ dW2\rangle = \langle dZ1\ dZ2\rangle = \langle dW1\ dZ2\rangle = \langle dW2\ dZ1\rangle = 0$$

where X is the FX process, rd is the domestic short rate, rf the foreign short rate, v1 and v2 the instantaneous FX variances, kappa1 and kappa2 the FX mean reversion speeds, theta1 and theta2 the long term mean FX variances, v01 and v02 the spot FX variances, and rho1 and rho2 the correlations of the variance processes with the FX process drivers.

The paths are constructed with quasi-random (Sobol sequence) numbers and Brownian bridge scrambling.

Timings

A six month USD/JPY accumulator with daily monitoring. Range barriers set at 90% and 110% of initial spot FX. One quarter million paths. Standard deviation of computed PV is less than 0.01%.

Serial	OpenMP (quad-core PC)	Single GPU	Dual GPU
52.1 seconds	13.2 seconds	0.75 seconds	0.50 seconds
	(x 3.94)	(x 68)	(x 104)

SciFinance GPU/OpenMP Case Studies

Case Study III: Quanto CMS Spread Range Accrual Swap

The floating leg of the swap pays in domestic currency a coupon based on the domestic LIBOR rate:

Coupon = Gear Libor + Spread

while the "fixed" leg pays in domestic currency a fixed coupon with a range accrual multiplier based on foreign CMS spread and FX rate:

Coupon = FixedCoupon RangeFraction

where RangeFraction is the fraction of Monitor Dates in any period for which both of these conditions are true:

FXMin <= FX <= FXMax   AND   SMin <= CMSSpread <= SMax

And where CMSSpread is the spread between foreign CMS rates.

Pricing Model

Both domestic and foreign interest rates are modeled by a two-factor Gaussian (G2) short-rate models, while the exchange rate follows a lognormal process. The full 5x5 correlation matrix is reduced to five free correlation parameters: an inter-factor correlation for each G2 model, and three macro correlations: rd-rf, rd-FX and rd-FX. Brigo & Mercurio provide details of the correlation reduction.

Timing

A ten year USD/GBP Quanto CMS swap with semi-annual coupons for both legs. The CMSSpread is between 2 and 5 year foreign rates. The CMS spread range limits are 0-2%, while the FX range limits are 83% and 117% of initial FX spot. One quarter million paths.

	OpenMP	GPU	GPU
Serial	OpenMP (quad-core PC)	Single GPU	Dual GPU
234 seconds	59.8 seconds	.72 seconds	0.48 seconds
	(x 3.91)	(x 325)	(x 487)

SciComp Inc.

5806 Mesa Drive, Suite 250, Austin TX 78731, 512.451.1050, fax 512.451.1622

www.scicomp.com