GPU-enabled Real-time Risk Pricing in Option Market Making

Cris Doloc, Ph.D.

21st March 2013

Disclaimer

- *The statements, remarks and conclusions of this presentation are my own and they do not represent necessarily the view of CTC*

- *The results presented in this paper are not necessarily related to the work I am currently doing for CTC, nor to the technology or infrastructure employed by CTC*

OUTLINE

1. *Introduction to Market Making in exchange traded Options*
2. *Defining the problem: Real-time Pricing of RISK*
3. *Survey of numerical methods for Option Risk valuation*
4. *CUDA implementation of accelerated lattice models*
   → *Preliminary results & comparative study*
5. *Potential new developments*
6. *Conclusions*

GPU users Ecosystem in finance

- **Investment Banks– OTC**
- **Insurance companies**
- **Solution Vendors / ISV**
- **Academia**

→ ***Proprietary Trading Firms***

→ ***Hedge Funds***

1. Intro to Options Market Making

Def. Market Maker:

- Providing liquidity continuously by capturing Bid-Ask spread
- Managing Risk and Inventory

Types of Risk:

- Market - Sharpe ratio
- Tail - Spans / shocks
- Liquidity - Liq. horizon
- Execution - market share/open interest
- Overhead - personnel
- Operational - technology infrastructure

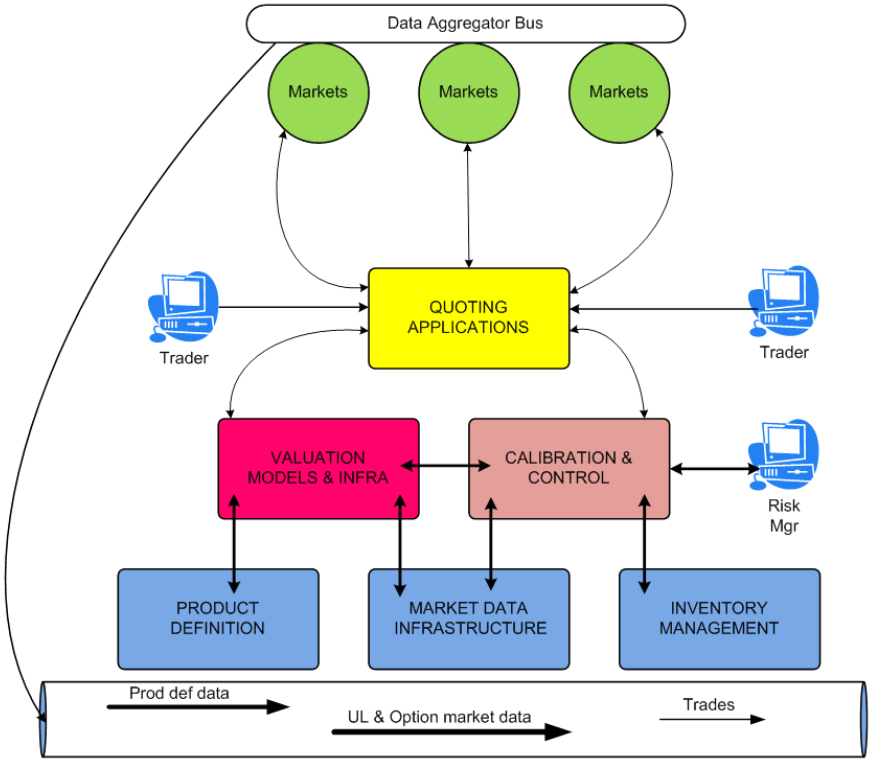
Def. Market Maker:

- Great risk adjusted returns (SR: 5-10)
- High burn rate – expensive technology

BID QTY	PRICE	ASK QTY
	100.50	3,500
	100.40	5,000
	100.30	7,000
	100.20	12,000
	100.10	15,000
	100.00	10,000
15,000	99.90	
25,000	99.80	
13,000	99.70	
7,000	99.60	
6,500	99.50	
5,000	99.40	
1,500	99.30	

Risk
&
Inventory

TECHNOLOGY INFRASTRUCTURE



- MM challenges:
  1. a high burn rate – very complex technology infrastructure
  2. limited scale (too risky > 20% market share)

High burn rate + limited scale → ***efficiency***

Innovation - part of the survival toolkit
- ➢Trading sheets 5-10 years ago
- ➢Now they are chasing microseconds

Precise Real-time Risk control is a key to success and survival

2. Defining the Problem

Options on a wide variety of Asset Classes and Geographies

- **Index** : Cash / ETF / Futures
- **Commodities**: Energy / Metals / Agricultural
- **Fixed Income**: entire Yield curve (US and foreign)
- **Equities**: US and foreign

Dimensionality

- Large portfolios: +100K products
- Complex scenarios: Underlying Prices / Volatility / Events: dividends, credit, political
- Real-time requirements - time-scale → ***sub-second***

Pricing Risk IN "REAL-TIME"

Drivers for Risk system requirements:

REGULATIONS & CORRELATION Risk

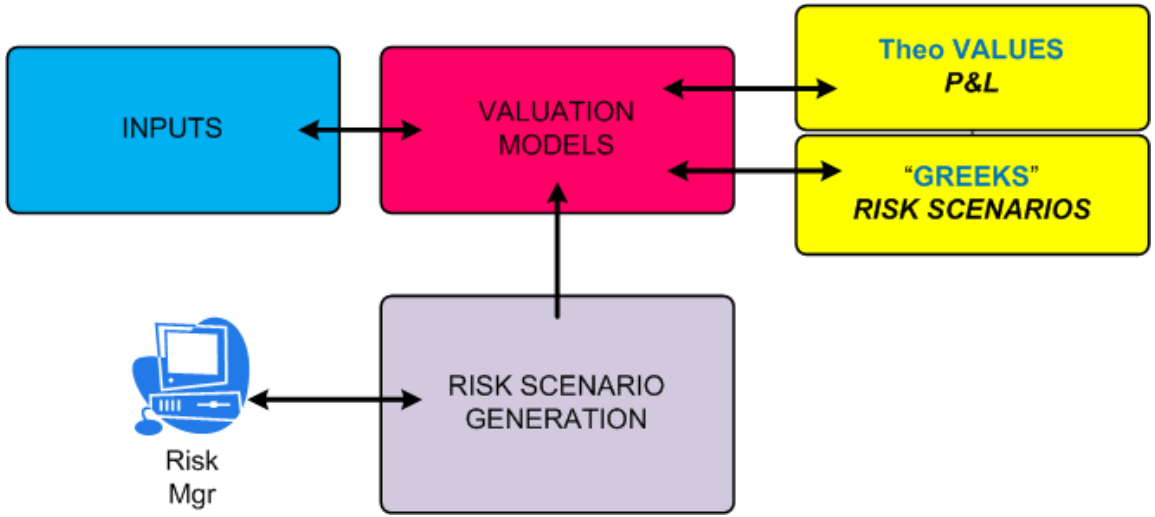
- ❑ High **AVAILABILITY** through **ACURACY/SPEED** → **TIGHTER RISK CONTROL**
- ❑ **SCALABILITY** across users/prod/geographies → **COST REDUCTION**

TIME SCALE: Real-time vs. Batch

- ➢ Real Time → P&L and basic Greeks (model parameters sensitivities)
- ➢ Scenarios → traditionally in batch mode, but currently RT

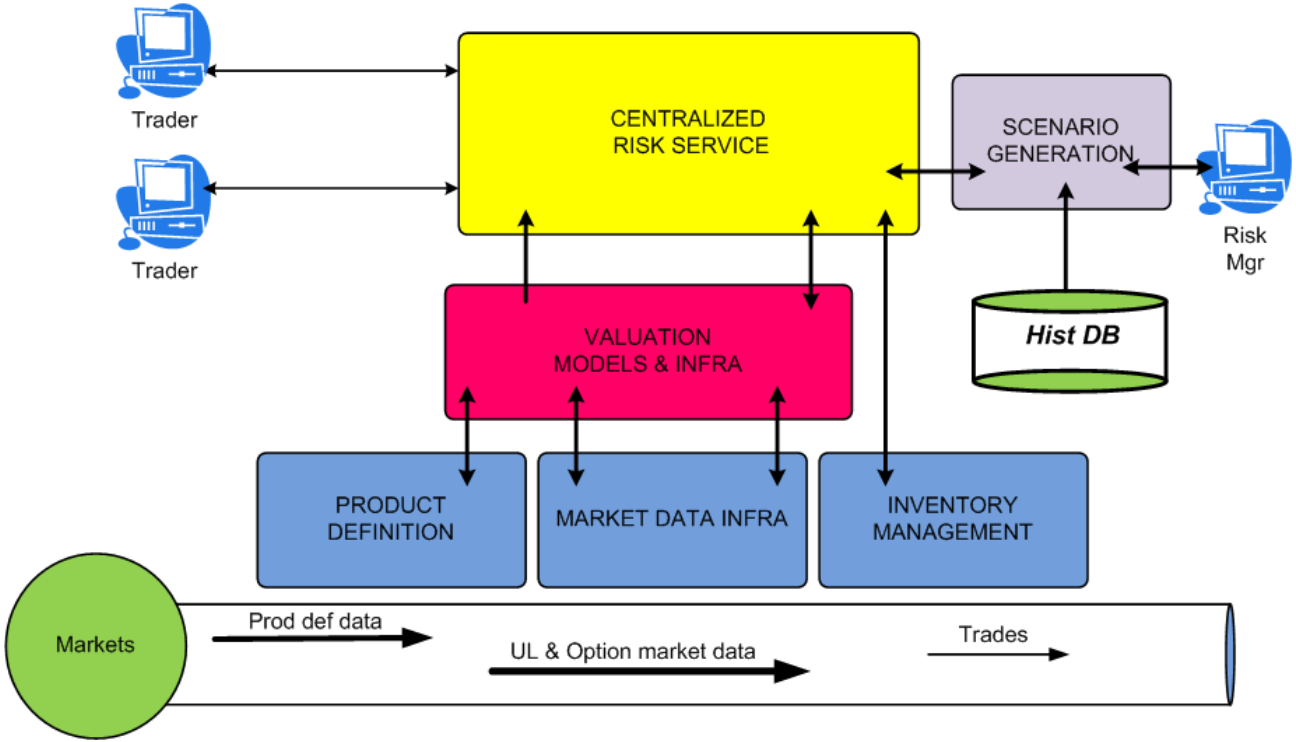
RISK VALUATION WORKFLOW

- ***DATA***
- ***COMPUTATION***



- Normalization
- Aggregation
- Stress testing

Real-time RISK Infrastructure



3. Survey of numerical methods for Pricing

Pricing Options with Early Exercise features:

1. PDE : *Finite Differences - Crank-Nicholson*
2. Analytical approximations: *Barone-Adesi-Whaley* [1]
3. ***Trees*** (explicit PDEs with backwardation)

- Trees have a natural financial interpretation, simple to build, and they converge to the Black–Scholes value
- Trying to approximate a probability measure rather than a PDE gives rise to different ***ideas for acceleration*** and parameter choices
- Numerous studies → find the most effective binomial tree by examining many acceleration techniques - increase ***Order of Convergence***
- Tree is specified by: **p**(up move) - **u**(up mult.) – **d**(down mult.)
- Self-similarity: p(N), u(N), d(N) but not of the step number
- Risk-neutrality: $$p = \frac{e^{r\Delta T} - d}{u - d},$$

VARIETY OF BINOMIAL TREE MODELS

- Cox-Ross-Rubinstein [2] : matches first 2 moments
  - *log-returns are binomially distributed*
  - *OC = 1*

$$u_n = e^{\sigma\sqrt{\Delta T}} \qquad d_n = e^{-\sigma\sqrt{\Delta T}} \qquad \Delta T = \frac{T}{n}$$

- Tian: matches first 3 moments

  OC = 1

$$u_n = \frac{1}{2} r_n v_n \left(v_n + 1 + (v_n^2 + 2v_n - 3)^{\frac{1}{2}} \right) \qquad r_n = e^{r\Delta T}$$

$$d_n = \frac{1}{2} r_n v_n \left(v_n + 1 - (v_n^2 + 2v_n - 3)^{\frac{1}{2}} \right) \qquad v_n = e^{\sigma^2 \Delta T}$$

- Jarrow-Rudd:
  - *not risk-neutral*
  - *OC = 1*

$$u_n = e^{\mu\Delta T + \sigma\sqrt{\Delta T}} \qquad \mu = r - \frac{1}{2}\sigma^2$$

$$d_n = e^{\mu\Delta T - \sigma\sqrt{\Delta T}} \qquad p_n = \frac{1}{2}$$

- "Accelerated" trees:
  - *A. Leisen-Reimer*
  - *B. Adaptive Mesh*

A. Leisen-Reimer TREE

Modifying the parameters of the binomial tree to minimize the oscillating behavior of the value function [3,4]

- Convergence of the CRR binomial trees is oscillatory
- Goal: maximize precision by minimizing # steps *N (odd #)*
- Leisen and Reimer (1996) developed a method where the choice of *p*, *u, d* was such to increase the order of convergence: **OC = 2**
- Leisen-Reimer suggest to use inversion formulae reverting the standard method—they use normal approximations to determine the binomial distribution *B(n, p):*
- *Model parameters:*

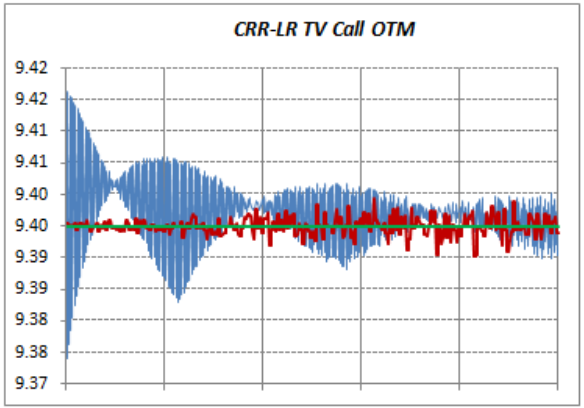
$$p(z) = \frac{1}{2} + sign(z)\frac{1}{2}\sqrt{1 - exp\left[-\left(\frac{z}{n+\frac{1}{3}}\right)^2\left(n+\frac{1}{6}\right)\right]}$$

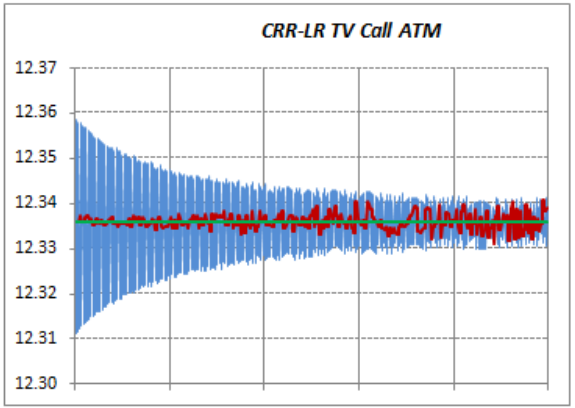
$$u = e^{(r_d - r_f)\Delta t}\frac{p(d_+)}{p(d_-)}$$

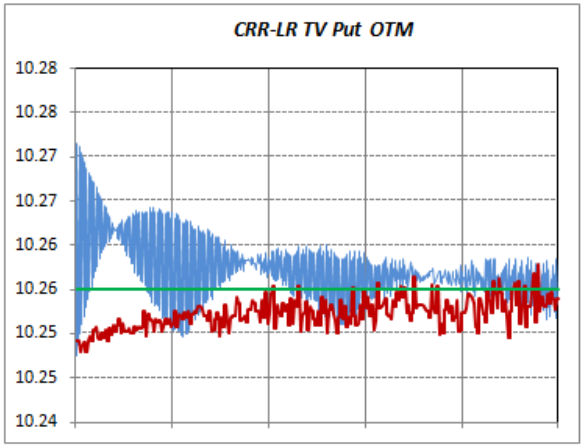
$$d = \frac{e^{(r_d - r_f)\Delta t} - p(d_-)u}{1 - p(d_-)}$$

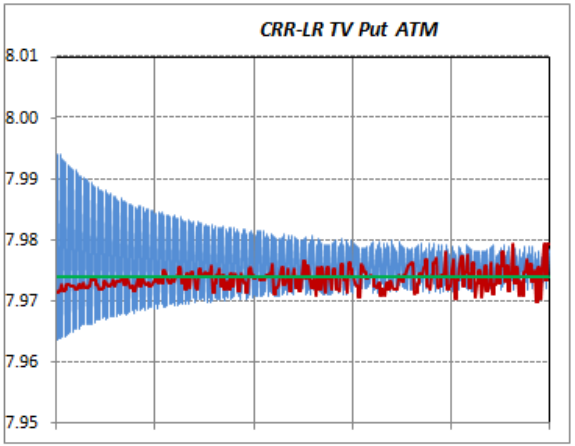
$$d_\pm = \frac{ln\left(\frac{S_0}{K}\right) + \left(r_d - r_f \pm \frac{1}{2}\sigma\right)T}{\sigma\sqrt{T}}$$

CRR versus L-R









S = 95
K = 100
σ = 25%
r = 5%
1 yr

Ways to FURTHER improve convergence

1. Truncation

- Construct the tree as far as 6 σ from the mean (log-space). Edge continuation value →Black–Scholes
- Has minimal effect on the price: typical effects are around 10^{-12}
- For large numbers of steps it can have large effects on speed of implementation since the number of nodes no longer grows quadratically (for small N, slightly slower b/c of BS eval.)

2. Control variates

- Given a binomial tree, one prices both the American put and the European put. If P_A is the tree price of the American put, P_E that of the European and P_{BS} that given by the Black–Scholes formula, we take the error controlled price to be:

$$\hat{P}_A = P_A + P_{BS} - P_E$$

3. Smoothing

- No exercise opportunities within the final step, so the derivative is effectively European
- More accurate price can be obtained by using the Black–Scholes formula for the final step
- With this technique we therefore replace the value at each node in the second final layer with the maximum of the intrinsic and the Black–Scholes values

Richardson Extrapolation

- If after n steps the price is: $X_n = TruePrice + \frac{E}{n} + o(1/n)$
- Then taking a linear combination of two calculations for n and 2n+1 steps:
  $Y_n = A_n X_n + B_n X_{2n+1}$
- With the additional constraints of: $A_n + B_n = 1.0$
  $\frac{A_n}{n} + \frac{B_n}{2n+1} = 0.0$
- Will lead to: $Y_n = TruePrice + o(1/n)$
- In reality for an American Option, the error will have also an *oscillation term*, but Richardson extrapolation will still reduce dramatically the size of the error:

$$Err = \frac{TreePrice - TruePrice}{0.5 + TruePrice - IntrinsicValue}$$

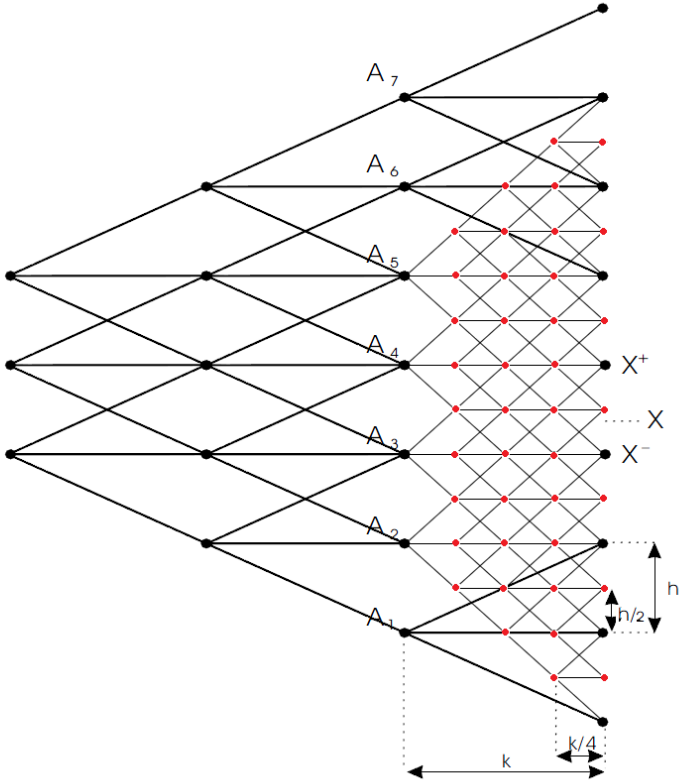
(Broadie & Detemple [5])

B. ADAPTIVE MESH TREES

- *For many Option Valuation problems (dividends, jumps, barrier options) convergence may be still slow and erratic*
- *Figlewski & Gao have introduced the adaptive mesh model (AMM)* [6]
- *Flexible approach that sharply reduces nonlinearity error by **grafting** one or more small sections of **fine high-resolution lattice** onto a tree with coarser time and price steps.*
- *Using a discrete-time/state lattice for an asset whose price is actually generated by a logarithmic diffusion introduces two different types of approximation errors:*
  - ***Distribution error** stems from the use of a discrete bi/tri probability distribution to approximate the continuous lognormal distribution produced by a diffusion process*
  - ***Non-linearity error** arises when the option TV is highly nonlinear or discontinuous in some region (at the strike price, at the expiration date, or at the barrier edge). TV errors could be quite large (long time to die out as # of time steps in the tree is increased)*

- *It is important for the fine mesh structure to be **isomorphic** so that additional finer sections of mesh can be added using the same procedure. This permits increasing the resolution in a given section of the lattice as much as one wishes without requiring the step size to change elsewhere.*

AMM tree example



- Model parameters:
  - *h – price step*
  - *k – time step*

$$h = \sigma\sqrt{3k}$$

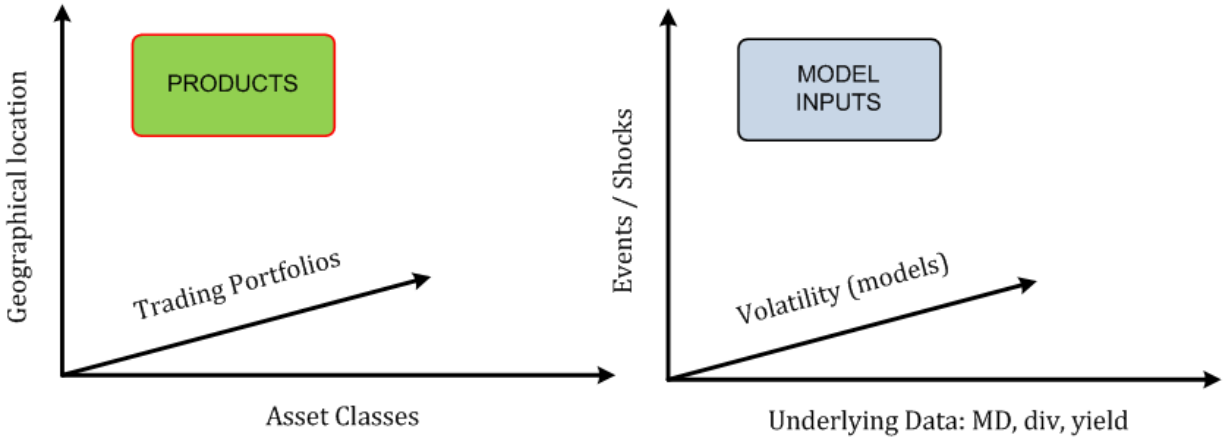
- The discrete trinomial process matches ***the first five moments*** of the continuous lognormal diffusion that it is designed to approximate
- In traditional Trinomial tree reducing ½ the price step to alleviate the non-linearity errors quadruples the number of nodes
- Grafting a ½ step mesh adds just 40 additional nodes of price computation in critical region

- ***AMM tree with 100 time steps and 10 dividend dates (critical regions with ¼ mesh), is more accurate than a 2000 step Trinomial and runs 500 times faster!***

4. Implementing ACCELERATED LATTICE Models in CUDA

- After the 2008 crisis the main focus in financial technology → **Real-Time Risk control**
- Pricing engines used by the trading desks needed to be **faster**, more **scalable** & more **cost effective**
- Emergence of the GPU technology provides **T-FPS performance** for pricing engines adding great **scalability** and **cost reduction** factors
- GPU is well suited for highly dimensional, computational intensive, parallel problems:

  - ***Real Time Risk valuation in Option Market Making***



REAL-Time RISK Valuation in CUDA

- GPU massive parallelization potential meets the requirements of the RT-R problem:
  - Millions of simultaneous calculations easily available within one process
  - Increased computing speed (x10-50 factors)
  - Dramatic reduction of the hardware footprint required to host the calculations
  - Better availability and scalability across products, desk and geographies
- Prototyping a new generation of Pricing Engines on latest GPU devices:
  - 2011: **Quadro 4000** CUDA 2.0 → 256 cores / 0.95 GHz / 2 GB
  - 2012: **GTX 680** CUDA 3.0 → 1536 cores / 1 GHz / 6 GB

- A comparison study

		Theo. Gain Factor
CPU	: 12 core **Intel Xeon X5680** / 3.57 GHz / 32 GB	1
GPU_1	: 4 x **Quadro 4000** - 1024 cores / 1 GHz / 2 GB	24
GPU_2	: 2 x **GTX 680/690** – 3072 cores / 1 GHz / 6 GB	72

- Minimizing to the Root Mean Square error (Broadie & Detemple)

Comparative study: models & devices

Speed / Accuracy **vs.** **Scalability / Efficiency**

Model	#Options	#Steps	Time-CPU (ms)	T-GPU_1 (ms)	Ratio_1	RMS_B/D (10^{-6})	T-GPU_2 (ms)	Ratio_2	RMS B/D (10^{-6})
CRR	100,000	100	3,282	553	5.93	379.30	173	18.99	363.30
L-R	100,000	**100**	3,995	643	6.22	**14.36**	201	19.90	**10.36**
CRR-RES	100,000	100	6,975	1,177	5.93	141.65	368	18.97	129.64
LR-RES	100,000	100	8,055	1,320	6.10	10.97	412	19.54	10.69

Model	#Options	#Steps	Time-CPU (ms)	T-GPU_1 (ms)	Ratio_1	RMS_B/D (10^{-6})	T-GPU_2 (ms)	Ratio_2	RMS B/D (10^{-6})
CRR	100,000	**500**	75,533	2,258	33.45	**17.30**	836	90.31	**15.70**
L-R	100,000	500	115,495	3,655	31.60	2.36	1,354	85.32	2.25
CRR-RES	100,000	500	160,522	4,805	33.41	12.46	1,780	90.20	13.36
LR-RES	100,000	500	232,874	7,506	31.03	1.97	2,780	83.77	1.58

Model	#Options	#Steps	Time-CPU (ms)	T-GPU_1 (ms)	Ratio_1	RMS_B/D (10^{-6})	T-GPU_2 (ms)	Ratio_2	RMS B/D (10^{-6})
Trinomial	10,000	**2,000**	**211,669**	6,517	32.48	**39.30**	2,256	93.82	**37.25**
AMM	10,000	**100**	**433**	14	30.71	**24.72**	7	64.63	**25.69**

Multiprocessor occupancy

- The ***multiprocessor occupancy*** is the ratio of active warps to the maximum number of warps supported on a multiprocessor of the GPU:
  - CUDA 2.0 → 48 x 32 = 1536 threads / MP
  - CUDA 3.0 → 64 x 32 = 2048 threads / MP

- ***Maximizing the occupancy*** can help to ***cover latency during global memory loads*** that are followed by a __syncthreads()

- The occupancy is determined by the amount of shared memory and registers used by each thread block → optimize the size of thread blocks in order to maximize occupancy

- Higher occupancy <> higher performance. If a kernel is not bandwidth-limited or latency-limited, then increasing occupancy will not necessarily increase performance

- If a kernel grid is bottlenecked by computation and not by global memory accesses, then increasing occupancy may have no effect. In fact it could create adverse effects: additional instructions, more register spills to local memory (which is off-chip), more divergent branches, etc.

Resource allocation mapping

Compute Capability	Quadro 4000 (2.0)	GTX 680 (3.0)
#cores / SMX	96	192
Warp schedulers	2	4
SMX	8	16

- Data parallelism: 1 option per block
- Task parallelism: 1 tree step per thread

CUDA 3.0

- For 128 steps LR option valuation on could calculate 16 options per MP by reusing input data for same expiration (2048 threads per MP – see Occupancy XL)

It takes ~ 2.5 microseconds to value (LR) an option at 128 steps → 1.2 MM options / second

- For 500 steps → 50,000 options / second (N^2 degradation)

5. FUTURE DEVELOPMENTS

- ***Implementing/reusing Tri-Diagonal solvers in CUDA for Finite Differences***
  - Parallel Cyclic Reduction algorithm: a specialized version of Cyclic Reduction designed to maintain uniform parallelism $\mathbf{O(\log_2 n)}$
  - "Near Real-Time" VAR calculations
  - Portfolio optimization via PCA and co-integration

- ***Financial application of Topological Data Analysis***
  - A recently introduced mathematical method (using algebraic topology) to analyze very large sets of data by capturing the shape of a point-cloud that "persists" in a dynamical setting → heavy parallel problem

Bibliography

[1] Barone-Adesi & R.E. Whaley. “Efficient Analytic Approximation of American Options Values.” *Journal of finance*, 42 (1987), pp. 301–320

[2] Cox, J. C., Ross, S. A., and Rubinstein, M. “Option Pricing: a Simplified Approach.” *Journal of Financial Economics* , 7 (1979), pp. 229–264

[3] Leisen, D. P. J. “Pricing the American Put Option: a Detailed Convergence Analysis for Binomial Models.” *Journal of Economic Dynamics and Control*, 22(1998), pp. 1419–1444

[4] Leisen, D., and Reimer, M. “Binomial Models for Option Valuation - Examining and Improving Convergence.” *Applied Mathematical Finance*, 3(1996), pp. 319–346

[5] Broadie, M., and J.B. Detemple. “American Options Valuation: New Bounds. Approximations and a Comparison of Existing Methods.” *Review of Financial Studies*, 9, No. 4 (1996), pp. 1211–1250

[6] Figlewski, S., and Gao, B. “The Adaptive Mesh Model: a New Approach to Efficient Option Pricing.” *Journal of Financial Economics* , 53 (1999), pp. 313–351

6. CONCLUDING REMARKS

- GPU Technology is in use currently by Proprietary Trading firms b/c :
  - Deal with “high dimensionality problems”: derivatives pricing & risk and portfolio optimization → GPU is the right tool for this problems
  - A “post-crisis” industry mandate for tighter Risk control & cost reduction
- Impressive improvements → orders of magnitude
  - Speed / Accuracy
  - Scalability / Efficiency
- Great potential for new advances:
  - BIG data processing
  - New acceleration techniques

Q & A