CATEGORY: **FINANCE**
POSTER **FI01**
CONTACT NAME
Christos Delivorias: christos@delivorias.me

GPU TECHNOLOGY CONFERENCE

ACCELERATING FINANCIAL MODELS

Christos Delivorias[1], Erik Vynckier[1], Peter Richtárik[2], Martin Takáč[2]

[1]Scottish Widows Investment Partnership, 60 Morrison Street, Edinburgh EH3 8BE, UK
[2]School of Mathematics, University of Edinburgh, James Clerk Maxwell Building, Edinburgh EH9 3JZ, UK
Christos.Delivorias@swip.com, Erik.Vynckier@swip.com, Peter.Richtarik@ed.ac.uk, M.Takac@sms.ed.ac.uk

THE UNIVERSITY of EDINBURGH

SCOTTISH WIDOWS INVESTMENT PARTNERSHIP
PERFORM

CONTRIBUTION:

Implement Heston's stochastic volatility model with Andersen's QE discretisation using Quasi-random numbers for variance reduction.

Accelerate this model using:

CUDA GPGPU AWS HPC	Matlab PCT GPGPU	Techila Azure Cloud	FPGA
2 Tesla M2050 GPU	2 Tesla M2090 GPU	356 cores @1.6GHz	4 × Max3 cards

Evaluate Improvement in acceleration.

QE-Scheme DISCRETISATION

Leif Andersen [1] proposed a new scheme to discretise the stochastic volatility and the price of an underlying asset.

Require: The present value for the variance, $\hat{V}(t)$
Ensure: The value for the variance in the subsequent time-step, $\hat{V}(t+\Delta t)$

Compute $m \leftarrow \theta + (\hat{V}(t)-\theta)e^{-\kappa\Delta t}$
Compute $s^2 \leftarrow \frac{\hat{V}(t)\xi^2 e^{-\kappa\Delta t}}{\kappa}(1-e^{-\kappa\Delta t}) + \frac{\theta\xi^2}{2\kappa}(1-e^{-\kappa\Delta t})^2$
Compute $\psi \leftarrow \frac{m^2}{s^2}$
if $\psi \le \psi_c$ **then**
  Compute $b \leftarrow 2\psi^{-1} - 1 + \sqrt{2\psi^{-1}}\sqrt{2\psi^{-1}-1}$
  Compute $a \leftarrow \frac{m}{1+b^2}$
  Generate Normal random variate Z_V
  return $\hat{V}(t+\Delta t) \leftarrow a(b+Z_V)^2$
else
  Compute $p \leftarrow \frac{\psi-1}{\psi+1} \in [0,1)$
  Compute $\beta \leftarrow \frac{1-p}{m}$
  Generate Uniform random variate U_V
  if $0 \le U_V \le p$ **then**
    return $\hat{V}(t+\Delta t) \leftarrow 0$
  else
    return $\hat{V}(t+\Delta t) \leftarrow \beta^{-1}\ln\left(\frac{1-p}{1-U_V}\right)$
  end if
end if

BIBLIOGRAPHY

[1] L. Andersen, Efficient Simulation of the Heston Stochastic Volatility Model. Available at SSRN: http://ssrn.com/abstract=946405, 2007.

[2] G. Levy, An introduction to quasi-random numbers. http://bit.ly/13AUKVj, 2002.

[3] H. Niederreiter, Random number generation and Monte Carlo method. In SIAM, 1992.

[4] C. Delivorias, https://github.com/cmdel/mc-sim, 2012.

Quasi-Random VARIATES

Such numbers can be sampled from the so called "low discrepancy" sequences. A sequence's discrepancy is a measure of its uniformity and is defined by the following definition [see Levy[2]].

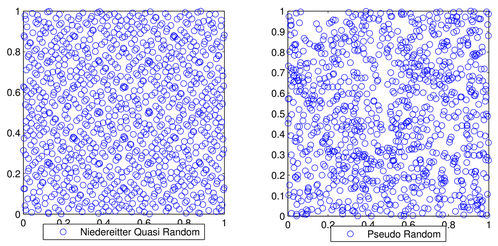
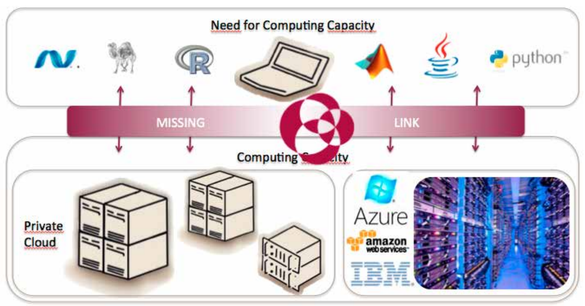


Figure 1: The Niederreiter [3] sequence is generated in MATLAB with the NAG functions g05yl and g05ym

TECHILA



Techila Technologies provided a middleware layer for parallel distribution. The grid comprised of 354 cores clocked at 1.6GHz on Microsoft's Azure cloud. The only change required in the code was the substitution of the *for* command with the *cloudfor* one.

```
% Main Monte Carlo loop
cloudfor pth = 1:paths
%cf:force:largedata
  (...)
    Payoff(pth) = max(S(pth,end)-K,0);
                  % Call option
  (...)
cloudend
```

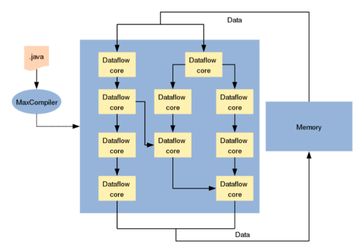
MATLAB GPGPU

The Heston model was ported to the GPU architecture by use of Matlab's Parallel Computing Toolbox. All functions that are executed on the GPGPU card are within the *gpuArray* namespace.

```
(...)
% Assign each worker to a different GPU
spmd
gpuDevice(labindex);
end
(...)
% Main Monte Carlo loop
for step = 1:steps
    Zn1 = parallel.gpu.GPUArray.randn(
        paths,1);
  Zn2 = parallel.gpu.GPUArray.randn(
      paths,1);
  Uv  = parallel.gpu.GPUArray.rand(paths
      ,1);
    [S,V]=arrayfun(@myGPUFun, Zn1,Zn2,Uv
      ,S,V,theta, kappa, dt, xi,r,K0,
      K1,K2,K3,K4);
end
(...)
```

DATAFLOW ON FPGA

What Maxeler provide is a way to create workers on an FPGA that are highly specialised, and extremely quick at conducting a specific operation. The kernels are large synchronous dataflow pipelines that implement the mathematics and the control of the problem. They are asynchronously coupled to other kernels and I/O sources and sinks (DRAM, PCIe, inter-chip links,) by the manager. The overall process is as quick as the flow.



PROJECT CONTRIBUTORS

NVIDIA. TECHILA MAXELER Technologies MAXIMUM PERFORMANCE COMPUTING BOSTON Servers · Storage · Solutions nag MathWorks

PRECISION *and* ACCURACY

The QE-scheme has high accuracy, compared to the analytic price calculated with the NAG function, and converges stably and quickly to a low standard error. The asset's variance is strictly positive and strongly repelant from zero.

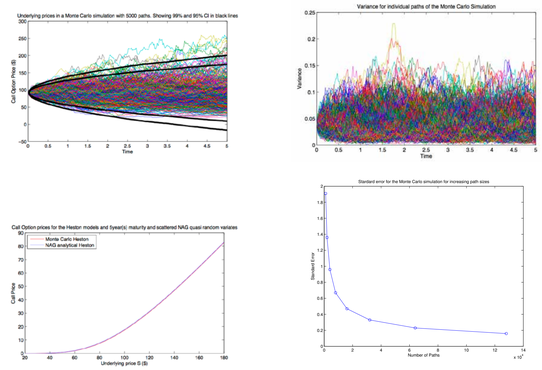
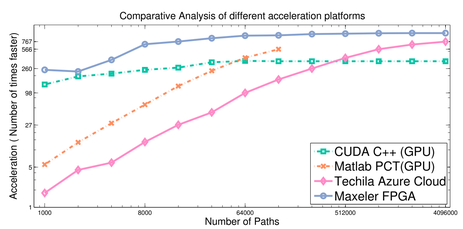


Figure 2: Matlab figures for the Monte Carlo simulation of the Heston model. The parameters for this simulation are: $S_0 = 100, V_0 = 0.04, K = 100, \rho = -0.2, \xi = 0.2, \theta = 0.04, \kappa = 1.5, T = 5.0$, NoSteps $= T * 250$, NoPaths $= 1000, \lambda = 0, r = 0.0$, and $q = 0.0$. For the bottom two plots the S price is perturbed in the prices $S_0 = [20 \quad 30 \quad \cdots \quad 100 \quad \cdots \quad 170 \quad 180]$.

RESULTS

All platforms seem to converge to the same acceleration boundary. This is however with a different cost for each one of them. Total cost of ownership and ease of development should be taken into account. For code samples see [4]



CONCLUSIONS

1. FPGAs fill the pipe early on and plateau at a certain acceleration; needs to scale horizontally with additional cards
2. FPGAs had the best acceleration; high development friction
3. Easiest implementation was with Techila, followed by Matlab PCT
4. Performance via middle-ware suffers at small path number sizes, but catches up eventually