Monte Carlo for Financial Computations on GPU

Who? A. Lokman Abbas-Turki

From? ENPC

When? Presentation
20 October 2008

Plan

Plan

Introduction

Parallelizing LCG

Method
Perspectives

Path-dependent Contracts

Ratchet Product
Target Redemption Risk Neutral
Speedup Survey

American Options

Conclusion

GeForce 8800 GTX

- Shadering frequency: 1.35 GHz
- Contains 128 processing units: we hope 128x
- Memory access 2x speeder (900 MHz): maximum speedup 256x
- Handles floats and integers

OBJECTIVES

1°) Reducing the communication between CPU and GPU

2°) Reducing the communication between the processing units on the GPU

Solutions

1°) Generating random numbers directly on the GPU

2°) Using inter-independent random generators

Plan

General Form

- Using 3 constants a, c et m. Choosing the seed randomly, the recurrence of LCG is:

$$x_n = a x_{n-1} + c \quad mod(m) \tag{1}$$

- Periodic, maximal period T_{max} takes the value :

CASE 1 $T_{max} = m$ if $c \neq 0$.
CASE 2 $T_{max} = m - 1$ if $c = 0$.

- m is a prime number in order to obtain a similar random behavior for all bits, typically $m = 2^{31} - 1$.
- $c = 0$ because it only translates the sample of random numbers

Parallel Dependence

$$x_n = ax_{n-1} \mod m$$
et
$$y_n = a^r y_{n-1} \mod m$$

r prime with $m-1$.
Without loss of generality: $x_0 = y_0 = z$, the inter-correlation exponential sum of $x_n - y_n$:

$$\frac{1}{m-1} \sum_{0 \le n \le m-1} \chi(x_n - y_n) = \frac{1}{m-1} \sum_{n=0}^{m-2} e^{\frac{2\pi i}{m} f(n)} \qquad (2)$$

with $f(n) = z(a^n - a^{rn})$.

But: $1/(m-1) \times |\sum_{n=0}^{m-2} e^{\frac{2\pi i}{m} f(n)}| \le 1/(m-1) \times (r-1)\sqrt{m}$

If: $1/(m-1) \times (r-1)\sqrt{m} = O(\sqrt{1/m})$

Then: $r - 1 << \sqrt{m}$

Application

- $m = 2^{31} - 1$
- $\sqrt{m} \approx 46341$
- Taking $r \in \{1,\ldots,1000\}$
- One generator for each value of r

Sequential Aspect ?

What about the random behavior of the generator associated with r?

Choosing the multiplier

Keep only the multipliers that minimize the value of:

$$\left(\left|-\frac{1}{2}\sum_{1\leq j\leq t}^{j\ odd} e_j - \sum_{1\leq j\leq t}^{j\ even} e_j + \frac{1}{2}\right|, \quad \left|\sum_{1\leq j\leq t}^{j\ odd} e_j + \frac{1}{2}\sum_{1\leq j\leq t}^{j\ even} e_j - \frac{1}{2}\right|\right)$$

e_j = partial coefficients of Euclide table associated to $a^r\ mod(m)$ and m.

This choice:

- reduces the correlation between the random numbers of the same generator.
- insures a good distribution in terms of discrepancy.

Final Results

- $\sim 512 = 2^8$ generators with good parallel and sequential behavior.
- The total period is $\sim 2^8 * 2^{31} = 2^{39}$: sufficient for most financial applications.

The new Tesla generation

(a) (b)

Fig.: *Tesla 10 series : (a) C1060, (b) S1070*

Specifications

- The number of processing units doubled
- Memory space quadrupled
- Double precision

Application

- $m = 2^{61} - 1$
- $\sqrt{m} \approx 1.5e6$
- Taking $r \in \{1,...,1e4\}$

Results for this Tesla

- $\sim 4096 = 2^{12}$ generators with good parallel and sequential behavior.
- The total period is $\sim 2^{12} * 2^{61} = 2^{73}$

Plan

Monte Carlo

$$\sum_{i=0}^{N} \mathbb{E}\left(\prod_{j=1}^{i} B(T_{j-1},T_j)F_i\right) \qquad (3)$$

with:

$$F_i = f(L_{T_i},F_{i-1}) = \Delta T(K + F_{i-1} - L(T_i,T_i,T_i+\Delta T))_+ \qquad (4)$$

and:

$$L(t,T,T+\Delta T) = \frac{1}{\Delta T}\left(\frac{B(t,T)}{B(t,T+\Delta T)} - 1\right) \qquad (5)$$

Algo for date k

- Bond simulation: $B(T_{k-1},T_k)$
- Actualizing the product: $\prod_{j=1}^{k} B(T_{j-1},T_j) = \prod_{j=1}^{k-1} B(T_{j-1},T_j) * B(T_{k-1},T_k)$
- Payoff computation: $F_k = f(L_{T_k},F_{k-1})$ according to: (4) and (5)
- Actualizing the result: $\sum_{i=1}^{k}\left(\prod_{j=1}^{i} B(T_{j-1},T_j)F_i\right)$

Monte Carlo

$$\sum_{i=1}^{N}\mathbb{E}\left(\prod_{j=1}^{i}B(T_{j-1},T_j)1_{\sum_{k=0}^{i-1}F_k<M}F_i\right) \tag{6}$$

with:

$$F_i=\Delta T(K-L(T_i,T_i,T_i+\Delta T))_+ \tag{7}$$

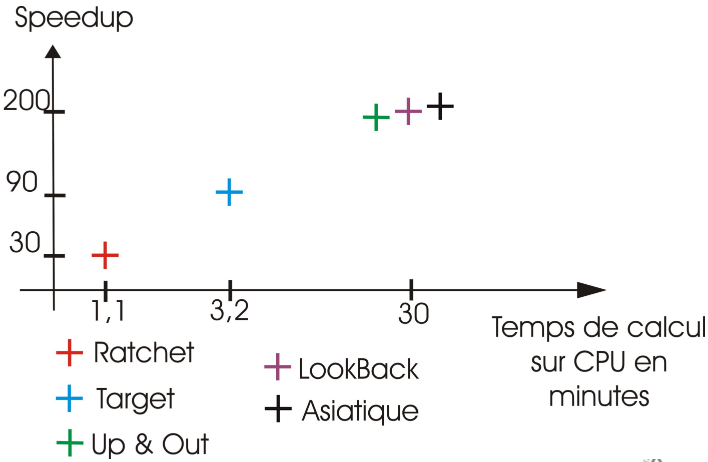
and:

$$L(t,T,T+\Delta T)=\frac{1}{\Delta T}\left(\frac{B(t,T)}{B(t,T+\Delta T)}-1\right) \tag{8}$$

Algo for date k

- Bond simulation: $B(T_{k-1},T_k)$
- Actualizing the product: $\prod_{j=1}^{k}B(T_{j-1},T_j)=\prod_{j=1}^{k-1}B(T_{j-1},T_j)*B(T_{k-1},T_k)$
- Payoff computation: F_k according to: (7) and (8)
- Actualizing the result: $\sum_{i=1}^{k}\left(\prod_{j=1}^{k}B(T_{j-1},T_j)1_{\sum_{k=0}^{i-1}F_k<M}f(X_{T_i})\right)$
- Actualizing the sum: $\sum_{k=0}^{i}F_k=\sum_{k=0}^{i-1}F_k+F_i$

Speedup of five exotic European contracts



Plan

The problem to solve

$$P_t(x) = \sup_{\theta \in \mathcal{T}_{t,T}} \mathbb{E}_{t,x}\left(e^{-r(\theta-t)}\Phi(X_\theta)\right) \tag{9}$$

Approximation

$$\mathcal{T}_{0,T} = \{0 = t_0 < t_1 < \ldots < t_n = T\} \tag{10}$$

Dynamic Programmation

$$\begin{cases} P_T(X_T) = P_{t_n}(X_{t_n}) = \Phi(X_T) \\ \forall k \leq n-1,\ P_{t_k}(X_{t_k}) = \max(\Phi(X_{t_k}), e^{-r(t_{k+1}-t_k)}\mathbb{E}\left(P_{t_{k+1}}(X_{t_{k+1}}) \,\middle|\, X_{t_k}\right)) \end{cases} \tag{11}$$

Dynamic Programmation

$$\begin{cases} P_T(X_T) = P_{t_n}(X_{t_n}) = \Phi(X_T) \\ \forall k \leq n-1,\ P_{t_k}(X_{t_k}) = \max(\Phi(X_{t_k}), e^{-r(t_{k+1}-t_k)}\mathbb{E}\left(P_{t_{k+1}}(X_{t_{k+1}}) \middle| X_{t_k}\right)) \end{cases} \tag{12}$$

Malliavin with Black & Scholes

$$\forall i \leq N,\ \mathbb{E}(P_{t_{k+1}}(X_{t_{k+1}})|X^i_{t_k}) = \frac{\mathbb{E}\left(P_{t_{k+1}}(X_{t_{k+1}})\frac{H(X_{t_k}-X^i_{t_k})}{X_{t_k}}\Delta W_k\right)}{\mathbb{E}\left(\frac{H(X_{t_k}-X^i_{t_k})}{X_{t_k}}\Delta W_k\right)} \tag{13}$$

avec : $\Delta W_k = (t_{k+1} - t_k)W_{t_k} - t_k(W_{t_{k+1}} - W_{t_k}) + t_k(t_{k+1} - t_k)\sigma$

Some execution times for 512x512 sample size

nb of factors	10 time steps	100 time steps
1	1m 50s	22m 00s
5	2m 10s	23m 00s
20	3m 15s	45m 00s
40	5m 00s	53m 00s

Plan

Computation Performance

- Path-dépendants European contrats $\sim$ 100x speeder than CPU.
- American Contracts simulation: very fast.

Perspectives

- Studying the convergence of American prices using Malliavin Calculus.
- GPU: Technology in continual development.
- Cluster implementation: Increasing the speedup.

Conclusion

Promising future for GPUs

Thank you !