Vectorizing Black-Scholes in the GPU

Vini and Diogo

Grinnell College
Computer Science Department

December 15, 2023

Overview

Introduction

Goals: Parallelize option pricing with Black-Scholes model using the GPU

- Investigate the intricacies of option pricing with the Black-Scholes Model.
- Implement parallelization techniques to improve computational efficiency.
- Synchronize threads to avoid repeated and/or wrong calculations.
- Implement a file handling system that smoothly process data for the Black Scholes computation
- Test the performance impact of GPU parallelization.

Class concepts: Parallelism with GPUs, Thread Synchronization, and Files and File Systems

First, what are options?

- Financial contract that gives the holder the right (but not the obligation) to buy or sell an underlying asset at a predetermined price within a specified timeframe.
- **Call Option**:A call option gives the holder the right to buy the underlying asset at a predetermined price (strike price) before or at the expiration date.
- **Put Option**: A put option gives the holder the right to sell the underlying asset at a predetermined price (strike price) before or at the expiration date.

Figure: Call option

Figure: Put option

Black-Scholes Model: determining option premium

Model

$$rV = \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S}$$

European Option Solution

$$\begin{aligned}
\mathrm{Call}(S_t, t) &= \Phi(d_1)S_t - \Phi(d_2)Ke^{-rt} \\
\mathrm{Put}(S_t, t) &= Ke^{-rt} - S_t + \mathrm{Call}(S_t, t) \\
d_1 &= \frac{\ln\frac{S_t}{K} + (r + \frac{\sigma^2}{2})t}{\sigma\sqrt{t}} \\
d_2 &= d_1 - \sigma\sqrt{t}
\end{aligned}$$

Note: Returns on stocks are assumed to be normally distributed.

Black-Scholes Model: calculating d_1 and d_2

- We can use the error function to calculate the normal distribution cdf:

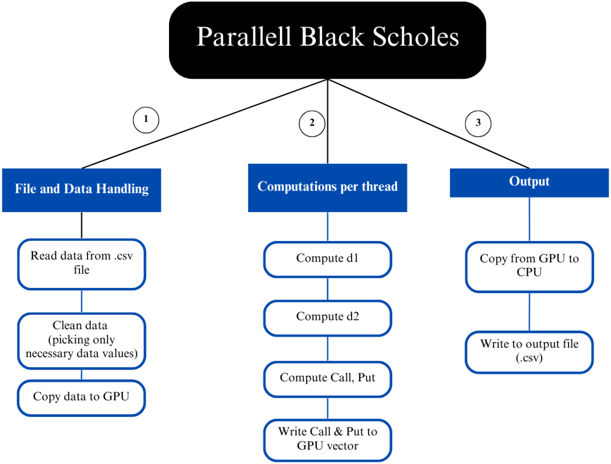
$$\Phi(x) = \frac{1}{2}(1 + \mathrm{erf}(x))$$

- Where

$$\mathrm{erf}(x) = \frac{2}{\pi} \int_0^x e^{-t^2} \mathrm{d}t$$

- Then, we can calculate d_1 and d_2 using the equations from the last slide, and use them to find the call and put prices

Implementation Details



DEMO

Implementation Roadmap and Challenges

- Roadmap
  1. Understanding black scholes
  2. Naïve implementation of the model
  3. Parallelize model for large scale data
  4. Develop script to parse csv
- Challenges
  - Linker errors for handling multiple .cu and .c files
  - Double computation avoidance
  - Implementation of the normal cdf using the error function
  - Math
- Next Steps
  - Implement the computation of the Greeks for each option provided
  - Check for a way to avoid putting our code in a .h

Thank you! Questions?