Parallelism

David Lu T14902116
蔡琦皇 P13922006
楊翊廷 R14944018
詹淯翔 B13201026

Problem description

Motivation: Amongst the various option pricing techniques, Monte Carlo is one of the most flexible and uses simulation to generate prices of an underlying asset. It is often used to model Asian and European options. However, Monte Carlo methods are computationally expensive. Simulating millions of price paths, each with hundreds of steps, requires lots of computational power. On a CPU, simulations can take seconds to minutes, which is too slow for the real world. GPUs are well suited for this type of programming. Each simulation path is independent, which means it's pretty natural to parallelize them. The goal of this project is to maximize simulation throughout and explore the use of CUDA in finance.

Problem description

Asset Price Dynamics: We model the underlying asset price St using the Geometric Brownian Motion (GBM)
$公式$

Monte Carlo Method: Estimating the expected payoff of financial derivatives by simulating thousands of possible price paths and discounting them back to the present.

Computational Bottleneck: Real-time pricing of complex options (e.g., Asian or Multi-Asset) requires a massive number of simulations, leading to high latency on serial CPU implementations.

Objective

Goal: Identify the most efficient way to accelerate Monte Carlo simulations.

- **Performance Testing:** Benchmarking throughput and latency.
- **Scaling Strategy:** Transitioning from CPU to GPU, and finally to Multi-GPU.
- **Validation:** Ensuring accuracy remains consistent across all hardware platforms.

Implementation

We implemented three levels of parallelism:

- **OpenMP:** Multi-core CPU parallelization.
- **CUDA (Single GPU):** Massive threading for SIMT (Single Instruction, Multiple Threads).
- **Multi-GPU (4x CUDA):** Distributed workload across 4 GPUs to maximize throughput.

Experiments

Testing 4 option types with increasing complexity:

- European: Standard benchmark.
- Asian: Path-dependent (Arithmetic Average).
- Bucket (Basket): Multi-asset simulation.
- Multi-Asset: High-dimensional simulation using Cholesky decomposition.

European

Asian

Basket

Multi-Asset

Environment

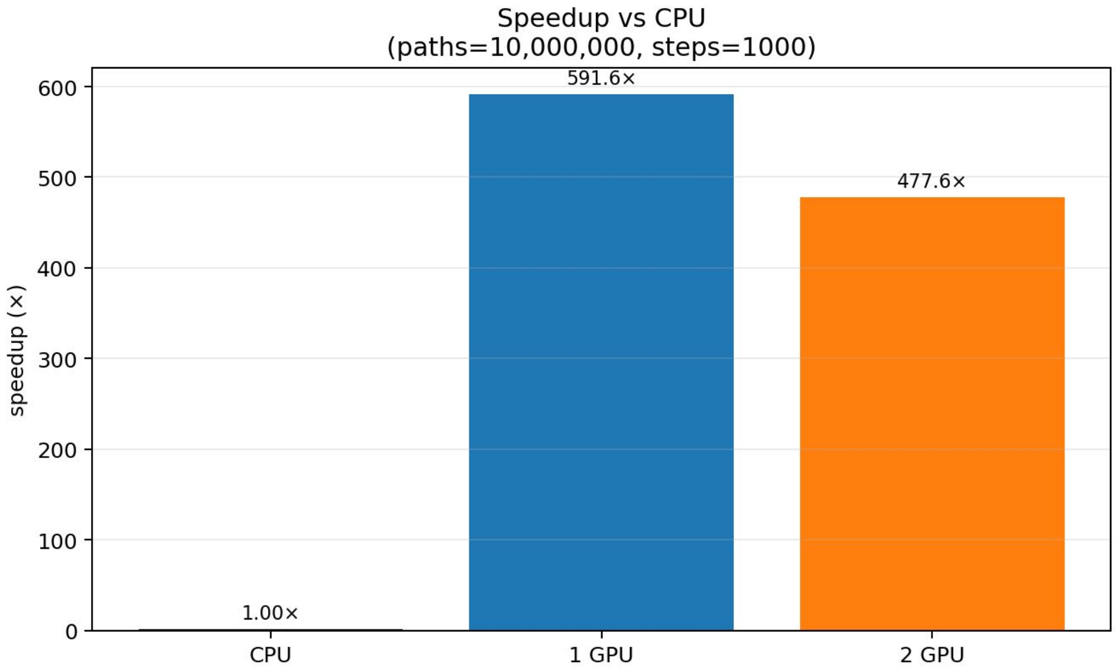
CPU

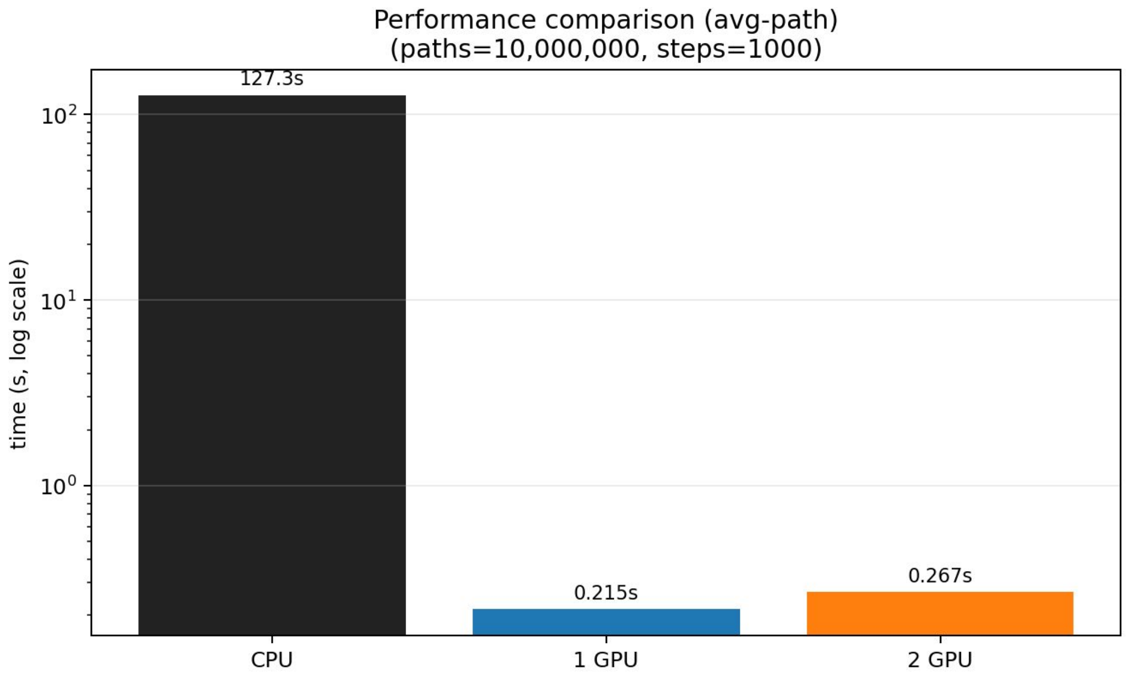
項目	描述	數量
CPU	Intel Xeon Gold 6154 18 Cores 3.0GHz	2
Memory	32GB DDR4-2666 RDIMM	24
系統硬碟	2.5" 240GB SATAIII	2 (RAID 1)
資料暫存硬碟	4TB NVMe	1
GPU	NVIDIA® Tesla® V100 SXM2	8
網路卡	Mellanox InfiniBand EDR 100Gb	4

GPU

```
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.161.08             Driver Version: 535.161.08   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla V100-SXM2-32GB           On  | 00000000:1B:00.0 Off |                    0 |
| N/A   28C    P0              42W / 300W |      0MiB / 32768MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2-32GB           On  | 00000000:1C:00.0 Off |                    0 |
| N/A   26C    P0              40W / 300W |      0MiB / 32768MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+
```

Result


Speedup vs CPU
(paths=10,000,000, steps=1000)
speedup (×)
600
500
400
300
200
100
0
591.6×
477.6×
1.00×
CPU
1 GPU
2 GPU



Performance comparison (avg-path)
(paths=10,000,000, steps=1000)
time (s, log scale)
10^2
10^1
10^0
127.3s
0.215s
0.267s
CPU
1 GPU
2 GPU

limitation