Problem Title: Parallelizing Monte Carlo Simulation of Financial Derivatives using SYCL

Problem Statement:
Monte Carlo simulation is a popular technique used in finance to simulate the behavior of financial derivatives, such as options, in order to estimate their fair value and risk exposure. However, Monte Carlo simulation can be computationally expensive and time-consuming, especially for large portfolios or complex models, which limits its practical use. To overcome this challenge, parallel computing techniques can be used to accelerate Monte Carlo simulations and make them more efficient.

Task:
The aim of this hackathon is to parallelize Monte Carlo simulation of financial derivatives using SYCL/DPC++ Libraries to design and code a simulation model that accurately predicts stock prices using Monte Carlo simulation within a defined timeframe. The project will involve the following tasks:

1. Implementing a Monte Carlo simulation model for financial derivatives using SYCL/DPC++ Libraries
2. Developing a parallel algorithm that can distribute Monte Carlo simulations across multiple devices, such as CPUs and GPUs, using SYCL.
3. Benchmarking the performance of the parallel Monte Carlo simulation against the serial version and assessing the speedup achieved.
4. Optimizing the parallel Monte Carlo simulation for performance and scalability by experimenting with different parameters, such as batch size and number of iterations.
5. Validating the results of the Monte Carlo simulation against analytical solutions and real-world data to ensure accuracy and reliability.

Note: It is mandatory to use the SYCL/DPC++ Libraries for the solution. Leverage the libraries and optimizations for the maximum efficiency and better models.

You can explore the Intel® Optimized Frameworks, SYCL/DPC++ Libraries, ICX to find the related libraries and optimizations for improving performance which will make your solution stand-out. Some of the useful links to explore more about oneAPI are mentioned below:

1. oneAPIDeep Neural Network Library
2. Intel oneAPI Math Kernel Library (oneMKL)
3. Intel® oneAPI Data Analytics Library
4. Intel® oneAPI DPC++ Library
5. Intel oneAPI DPC++ Library (oneDPL)
6. Intel® Optimization for TensorFlow*
7. Intel® Optimization for PyTorch*
8. Intel® Distribution for Python*
9. Intel® Extension for Scikit-learn
10. SYCL/DPC++ Essentials

Youtube Links:
oneDPL | oneAPI DPC++ Library
Direct Programming with SYCL

Expected outcomes:
The project is expected to deliver a parallel Monte Carlo simulation model for financial derivatives using SYCL that can be used for fast and accurate pricing and risk analysis. The project will also provide insights into the performance and scalability of SYCL for parallel computing in finance and demonstrate its potential for accelerating other financial simulations and applications.

Development of the algorithm in SYCL
Developing a parallel algorithm that can distribute Monte Carlo simulations across multiple devices, such as CPUs and GPUs, using SYCL.

Benchmarking results
Benchmarking the performance of the parallel Monte Carlo simulation against the serial version and assessing the speedup achieved.

Optimization results
Optimizing the parallel Monte Carlo simulation for performance and scalability by experimenting with different parameters, such as batch size and number of iterations

Dataset Description:
The historical data contains records about the stock price of various stocks like Apple, Microsoft, Facebook. The dataset also contains a date-wise price of stock with open, close, high, and low prices along with volume traded on that day.

Attributes	Values
Date	DateTime
Open	Decimal
High	Decimal
Low	Decimal
Close	Decimal
Volume	Numeric
Stock	String

Dataset:

1. Training_Data
2. Test_Data
3. Sample_Submission
4. Data_Dictionary

Judging Criteria:

1. Usage of SYCL/DPC++ Libraries (40%)
2. Completeness 10%
3. Correctness & Performance 10%
4. Scalability 5%
5. Ease of implementation 5%
6. Benchmarking 15%
7. Optimization 15%

Deliverables: (Ideation Phase)

1. Presentation.pptx: This should contain a description of what you have tried to build, what problem you are solving. It should clearly mention the usage of SYCL/DPC++ Libraries/Intel®
2. Participants are required to submit a comprehensive write-up that details their chosen theme, approach to the problem, and the code used to build their solution. This write up should be in the form of a technical article posted on Medium. This write-up will be evaluated by the judges for the functionality and creativity of the submitted solution. Therefore, it is crucial to submit a complete and clear write-up to increase your chances of winning the competition
3. readme.txt: This should contain clear step-by-step instructions on how to build, deployment and usages of the Web Applications.
4. Usage of Intel DevCloud. **Link:** DevCloud
5. Exploring the oneAPI toolkits/Libraries and its usage in the product is an essential aspect.
6. Products/Projects without oneAPI as the core component will not qualify for the hackathon

Note: The source code that's been submitted should be aligned with the project structure that you find on SYCL/DPC++ Libraries

Deliverables: (POC Round)

1. Solution file containing the predicted readmissions (Output file).
2. Code file for producing the solution file (Source Code file or .ipynb file).
3. Share your project on Github with source codes.
4. Video showing the demo functionalities of the Web application over YouTube/Drive.

5. Presentation.pptx: This should contain a description of what you have tried to build, what problem you are solving, and why your Web Application/solution should be considered for the final round. It should clearly mention the usage of SYCL/DPC++ Libraries
6. Implementation using oneAPI is mandatory for the POC completion.
7. Exploring the oneAPI toolkits/Libraries and its usage in the product is an essential aspect.
8. Products/Projects without oneAPI as the core component will not qualify for the hackathon

Submission Guidelines:

1. We expect a proof of concept/prototype to be built. Architecture and the usage of SYCL/DPC++ Libraries has to be clearly presented in the presentation.
2. The code must be made available through GitHub and the same should be presented in the submission page. Ensure access permissions are proper. You can share the solution, challenges faced, learnings, tech stack used over the GitHub. You can check out this link for reference:
    - https://www.intel.com/content/www/us/en/developer/articles/code-sample/oneapi-dpcpp-compiler-example-particle-diffusion.html
    - https://www.intel.com/content/www/us/en/develop/documentation/explore-dpcpp-samples-from-intel/top.html
3. The demonstration video can be presented through YouTube/Drive. Ensure you give access to view.
4. We prefer the projects to be submitted in the DevMesh portal.
   Link: Devmesh Portal

Resources:

Intel® oneAPI DPC++ Library

oneDPL | oneAPI DPC++ Library

Direct Programming with SYCL