

Relatório - Aula 11: Iteradores e Funções Customizadas

Aluno: Alexandre Santarossa

Disciplina: Supercomputação

Data de Entrega: 20/09/2024

1. Introdução

Nesta aula, exploramos técnicas avançadas para o processamento eficiente de grandes volumes de dados financeiros utilizando GPUs. O foco principal foi implementar algoritmos para calcular a variância e analisar variações diárias nos preços das ações, aproveitando iteradores dinâmicos e funções customizadas da biblioteca Thrust.

2. Cálculo da Variância usando Iteradores Dinâmicos

Desenvolvemos um programa em CUDA que calcula a variância das diferenças nos preços das ações. Utilizamos o `thrust::transform_reduce` junto com um iterador constante para gerar dinamicamente o vetor de médias, otimizando o uso de memória na GPU. Essa abordagem permite realizar cálculos complexos sem a necessidade de alocação adicional de memória, aproveitando a paralelização oferecida pelas GPUs.

3. Análise de Variação Diária dos Preços de Ações

Implementamos um código para calcular as variações diárias dos preços das ações a partir do arquivo `stocks-google.txt`. Utilizamos a função `thrust::transform` para gerar um vetor `ganho_diario`, que representa a diferença entre o preço de um dia e o anterior. Essa operação foi cuidadosamente projetada para garantir que o tamanho do vetor resultante fosse consistente com os dados de entrada.

4. Contagem de Dias com Aumento no Preço das Ações

Para identificar quantos dias apresentaram aumento no preço das ações, empregamos a função `thrust::count_if` juntamente com uma função customizada `is_positive`. Essa abordagem permitiu contar de forma eficiente apenas os elementos positivos no vetor `ganho_diario`, aproveitando a capacidade de processamento paralelo da GPU.

5. Cálculo do Aumento Médio nos Dias com Aumento

Desenvolvemos um método para calcular a média dos aumentos nos dias em que o preço das ações subiu. Inicialmente, utilizamos `thrust::replace_if` para substituir valores negativos por zero no vetor `ganho_diario`. Em seguida, calculamos a soma desses valores positivos e dividimos pelo número de dias com aumento, obtendo assim a média desejada de forma eficiente.

6. Discussão de Metodologia

A utilização da biblioteca Thrust facilitou a implementação de operações paralelas complexas, como transformações, reduções e contagens condicionais. O emprego de iteradores dinâmicos permitiu uma manipulação eficiente dos dados, minimizando o uso de memória e aproveitando ao máximo a capacidade de processamento da GPU. As funções customizadas foram essenciais para adaptar as operações às necessidades específicas da análise financeira, demonstrando a flexibilidade e o poder da programação paralela em aplicações de supercomputação.

Conclusão: A aula proporcionou uma compreensão aprofundada sobre como utilizar GPUs e a biblioteca Thrust para otimizar o processamento de grandes volumes de dados financeiros. As técnicas abordadas demonstraram ser eficazes na realização de cálculos estatísticos e análises de variação, evidenciando a importância da programação paralela na supercomputação.