14º SIEPE SALÃO INTERNACIONAL DE ENSINO, PESQUISA E EXTENSÃO
#EDUCAÇÃO + CIÊNCIA INDEPENDÊNCIA
30 nov. a 02 dez. 2022
REALIZAÇÃO: unipampa Universidade Federal do Pampa
APOIO: CNPq CAPES FAPERGS

AVALIAÇÃO DE DESEMPENHO DE UMA APLICAÇÃO DE MEIOS POROSOS

Lucas Antônio Toscan Sartori, discente de graduação, Universidade Federal do Pampa, Campus Alegrete
Claudio Schepke, docente, Universidade Federal do Pampa

lucassartori.aluno@unipampa.edu.br

O método de secagem de grãos consiste em armazenar os corpos porosos em um silo de secagem, a arquitetura do silo é composta de uma estrutura metálica e fechada com um formato cilíndrico, seu interior possui um piso perfurado e, na parte superior, um espalhador de grãos. O papel do silo de secagem é elevar a temperatura do ar que, por sua vez, passará por entre os orifícios dos grãos, eliminando uma certa quantidade de umidade (taxa de umidade ideal: 12% a 14%), esta técnica traz benefícios como menor perda de grãos e conservação do produto por um tempo maior.

A modelagem do método é representada por um fluido que escoa por entre um recipiente com partículas, cuja as quais possuem propriedades físicas bem definidas. Caso as partículas sólidas estejam em movimento, o problema pode ser definido como meio poroso não estacionário, caso contrário, será definida como meio poroso.

O presente artigo propõe-se a avaliar o desempenho de diferentes interfaces de programação paralela em uma aplicação de meios porosos, simulando a secagem de grãos.

Para que a aplicação tenha um comportamento parecido com o método de secagem é necessário simular a ação das moléculas que se dá através do deslocamento em função do tempo, contudo isto requer uma computação extremamente eficiente dos dados. Fazendo uso de computação heterogênea e diferentes diretivas de compilação para gerar códigos concorrentes através de threads, reduzindo assim o tempo de execução. A computação heterogênea requer o uso de interfaces de programação paralela, as escolhidas para a realização destes testes foram: OpenACC e OpenMP.

OpenACC é uma interface de programação paralela voltada para arquitetura heterogênea que disponibiliza ao programador diretivas de compilação, estas são chamadas em tempo de execução para gerar códigos para a GPU, permitindo que as variáveis sejam transferidas da memória da CPU para a GPU, executando os cálculos em um modelo vetorial, isto viabiliza uma maior eficiência na hora de apurar os resultados. Já o OpenMP é voltado ao paralelismo para aplicações de memória compartilhada, onde vários processadores compartilham da mesma memória, podendo operar de forma independente ou conjunta, também é possível definir o número de threads que serão usadas em cada paralelização.

O modelo de paralelismo usado, também aceito pela comunidade como modelo geral, é o “Fork/Join”, onde a aplicação inicia a execução no laço principal (master thread), em seguida é ramificado em sub-threads, e, até que em um dado momento da execução, estas sub-threads retornam seus respectivos resultados para a thread principal. Um dos fatores mais impactantes no desempenho de algoritmos é a computação heterogênea, para isto, as

interfaces armazenam um espaço na memória da GPU (co-processador), que executa as instruções seguindo um modelo vetorial, empiricamente a GPU demonstra um desempenho 20 vezes maior que a CPU.

A aplicação consiste de etapas iterativas de pré e pós-processamento, esta são chamadas em uma rotina principal, na etapa de pré-processamento ocorre a leitura de arquivos e alocação dos dados, a etapa iterativa é responsável por desenvolver o cálculo em função do tempo discreto (contabilizado em 0.1 segundos a cada iteração), ao final da iteração ocorre a atualização de algumas variáveis, por fim, a etapa de pós-processamento é encarregada de gerar saídas em formato de arquivo escritos por algoritmos em python.

O conjunto de equações escolhido foi o de Navier Stokes, trata sobre a mecânica dos fluidos, garante que o princípio fundamental da dinâmica seja aplicado em todas as partículas do escoamento, para calcular a velocidade de uma molécula é levado em consideração a variação da posição em função do tempo em um campo vetorial. Para a realização dos cálculos, o domínio foi decomposto por volumes finitos.

A máquina utilizada nos testes realizados possuía a seguinte configuração: SO: Ubuntu focal 20.04, GPU: Nvidia Quadro M5000, CUDA: 10.1.243, compilador: pgf90. A GPU mencionada, trata-se de um GPGPU, uma classe de GPU utilizada para propósito geral, não utilizada somente para renderização gráfica, mas sim para processamento de inteligências artificiais, cálculos numéricos e aplicações do gênero.

Foi executado um total de 30 testes para obter a média de tempo, para cada teste foram utilizadas 20,000 iterações. 3 testes de caso foram definidos para avaliar o desempenho dos algoritmos paralelos, nestes, foram utilizadas malhas bidimensionais com os seguintes tamanhos: 51 x 63, 100 x 24 e 200 x 249. Os resultados obtidos, em média, nas execuções sequenciais para estas malhas foram: 310,833, 1258,613 e 837,340 segundos.

Foi desenvolvida uma aplicação de meios porosos abertos, tendo como aplicação a secagem de grãos, testes foram realizados para atestar a precisão do código. O trabalho explora a computação heterogênea fazendo uso de interfaces paralelas (OpenMP e OpenACC) com diferentes números de threads.

Agradecimentos: PROBITI e PROBIC FAPERGS/UNIPAMPA.

Palavras-chave: Processamento de Alto Desempenho; Simulação de Secagem de Grãos; Programação Paralela.