



UM MODELO DE PARALELISMO EFICIENTE PARA O PROBLEMA DO MEIO ABERTO-POROSO ACOPLADO APLICADO AO PROCESSO DE SECAGEM DE GRÃOS

Hígor Uélinton da Silva, discente de graduação, Universidade Federal do Pampa, Campus Alegrete
Claudio Schepke, docente, Universidade Federal do Pampa

higorsilva.aluno@unipampa.edu.br

Estima-se que entre 10% a 25% da safra de grãos seja perdida na pós-colheita. A secagem, que é uma das etapas mais críticas na sequência do processamento do grão para sua correta conservação após a colheita, é um processo feito para retirar a umidade de um material, comumente utilizado na produção de alimentos. Esse processo consiste na remoção da umidade do núcleo até que o teor de umidade seguro seja geralmente 12-14 % em base úmida. Considerando que a massa do grão é uma quantidade de espaços sólidos e vazios (orifícios) através dos quais um fluido pode passar, a secagem do grão pode ser considerada um problema de meio aberto-poroso acoplado. Um modelo matemático e de simulação computacional foi previamente proposto para descrever a convecção em um fluxo livre com um obstáculo poroso aplicado à secagem de grãos. A partir dele, um esquema de dinâmica de fluidos computacional foi implementado em FORTRAN usando Volume Finito para simular e calcular as soluções numéricas. A aplicação consiste essencialmente de um grande laço de repetição, que vai iterando os passos discretos de tempo, onde para cada passo de tempo são calculadas as propriedades físicas simuladas. Porém, esse tipo de aplicação demanda muito tempo de execução para qualquer simulação simples. Para reduzir a execução para um tempo aceitável, exploramos a simultaneidade das instruções em arquiteturas multicore e GPU. Essas arquiteturas podem ser adotadas para aplicações numéricas. Neste trabalho, o foco foi acelerar o tempo de processamento da aplicação. Inicialmente, para determinar um modelo de paralelização eficiente para esse problema, decidiu-se pelo uso de duas interfaces já conhecidas, sendo elas OpenMP e OpenACC. OpenMP é baseado no modelo de execução *fork-join*, onde uma *thread* mestre começa a ser executada e gera *threads* de trabalho para executar em paralelo conforme necessário ou especificado. Essa interface utiliza diretivas para especificar os trechos de código onde se busca uma execução paralela. Desse modo consegue-se preservar a estrutura do código original, fazendo leves alterações tais como acréscimo de linhas ou alterações de estruturas. OpenACC também é uma API baseada em diretivas, assim como OpenMP, para o desenvolvimento de aplicações paralelas em arquiteturas heterogêneas, disponível para as linguagens C, C++ e FORTRAN. Essas diretivas especificam *loops* e blocos de código que podem ser descarregados da CPU para um acelerador conectado. No trabalho, *threads* paralelas foram criadas para dividir o cálculo dos laços das rotinas chamadas na etapa iterativa do código, tanto com OpenMP como OpenACC. O ambiente computacional utilizado nesse trabalho para execução dos testes é composto por dois processadores Intel Xeon E5-2650 e uma GPU Nvidia Quadro M5000. O processador conta com frequência de 2 GHz, com 8 núcleos e 16 *threads*, caches L1, L2 e L3 de, respectivamente, 32 KB, 256 KB e 20 MB e memória RAM de 128 GB. A GPU conta com frequência de 1,04 GHz, 2048 núcleos CUDA, caches L1 e L2 de, respectivamente, 64 KB e 2 MB e memória global de 8 GB. Em ambas as interfaces de programação, o tempo de processamento foi reduzido significativamente. O tempo total de simulação foi 8 vezes menor para uma arquitetura multicore (16 núcleos físicos) e 17,3 vezes menor usando uma única GPU. Dados esses resultados, a proposta do trabalho conseguiu reduzir significativamente o tempo de simulação, sendo possível realizá-las em poucos minutos. Como trabalhos futuros, pretende-se paralelizar a aplicação usando outra interface de programação paralela para uma possível comparação entre os resultados e avaliação de qual modelo seria o mais eficiente.

Agradecimentos: à FAPERGS pela concessão de bolsa de iniciação científica nos anos 2020 (PROBIC) e 2021 (PROBITI).

Palavras-chave: StarPU; OpenACC; OpenMP; Paralelismo de Tarefas; Secagem de Grãos.