XXXI Seminário de INICIAÇÃO CIENTÍFICA DA UFERSA

Núcleo de Avaliação: Núcleo II
Área temática: Ciências Exatas e da Terra
Área do Conhecimento: Ciência da Computação

Integração de Java e OpenACC via JNI para Aceleração em GPU: Estudo de Caso com Multiplicação de Matrizes

Samuel Rógenes Carvalho Freire, Paulo Henrique Lopes Silva

A crescente demanda por aplicações de alto desempenho impulsiona o uso de aceleradores, como as GPUs, que são essenciais para lidar com grandes volumes de dados e operações matemáticas intensivas. No entanto, linguagens de alto nível, como Java (Oracle and OpenJDK Community, OpenJDK 21 Documentation, 2025), amplamente utilizada em aplicações corporativas e em algumas áreas científicas, não oferecem suporte nativo à programação paralela em GPUs. Para contornar essa limitação e combinar a portabilidade e a simplicidade de Java com a eficiência do paralelismo em GPU, este trabalho teve como objetivo desenvolver e avaliar uma biblioteca que integra Java à programação paralela OpenACC (OpenACC, The OpenACC Application Programming Interface, Version 3.3, 2022) por meio da *Java Native Interface* (JNI) (Oracle, Java Native Interface Specification, 2025). Como material e método, foi utilizado o clássico problema de multiplicação de matrizes quadradas, selecionado devido ao seu alto custo computacional e elevado paralelismo intrínseco. A versão paralela foi implementada em C++ usando diretivas OpenACC, compiladas com o NVIDIA HPC SDK (NVIDIA Corporation, NVIDIA HPC SDK Documentation, Version 25.3, 2025), enquanto a interface e a chamada de execução foram controladas pela aplicação principal em Java. A comunicação entre as linguagens foi estabelecida pela JNI. O algoritmo empregou a definição padrão de multiplicação (Cij = Soma de A_ik * B_kj) com complexidade temporal $O(n^3)$ para o cálculo de cada elemento da matriz resultante. Para os testes, foram utilizadas matrizes quadradas com tamanhos variando de 100 até 5000 elementos. O desempenho da versão paralela em GPU foi comparado com o de uma versão sequencial equivalente em CPU, compilada com g++ (Free Software Foundation, GNU Compiler Collection, 2025). Os resultados obtidos revelam um importante *trade-off* de desempenho. Para matrizes pequenas (até 1000 x 1000), a GPU não apresentou ganhos significativos ou foi mais lenta que a CPU, pois o tempo foi dominado pela sobrecarga da JNI e pelo custo da conversão de estruturas Java para C++ contíguas, além da transferência de dados entre CPU e GPU (*Host-to-Device*) (Joshi, Best Practices for Using the Java Native Interface (JNI), 2020). Entretanto, para matrizes maiores, a GPU superou claramente a CPU, atingindo *speedups* expressivos de aproximadamente 4,23 vezes para matrizes de 2000 x 2000 e 4,45 vezes para matrizes de 5000 x 5000, evidenciando a eficiência da paralelização em larga escala. Em conclusão, o desenvolvimento da biblioteca demonstrou que a integração Java-OpenACC via JNI é uma alternativa viável e eficaz para aplicações de alto desempenho, permitindo conciliar a portabilidade de Java com a aceleração

por GPU, embora exija a consideração do *overhead* da JNI, que torna a paralelização vantajosa apenas para problemas de grande porte.

Palavras-chave: Programação Paralela, OpenACC, Java Native Interface, Computação de Alto Desempenho.

Agência financiadora: PIVIC (IC)

Campus: Mossoró