

# Proposta para um RTM Portátil: Explorando Kokkos e RAJA em Múltiplas GPUs

Rodrigo C. Machado<sup>1</sup>, Arthur F. Lorenzon<sup>1</sup>, Philippe O. A. Navaux<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{rcmachado, aflorenzon, navaux}@inf.ufrgs.br

**Resumo.** *Este trabalho propõe portar uma implementação CUDA de RTM multi-GPU para os frameworks Kokkos e RAJA, com o objetivo de avaliar seu desempenho e compatibilidade em diferentes arquiteturas de GPUs. A pesquisa envolve o estudo dos frameworks, a adaptação do código CUDA para C++ utilizando Kokkos e RAJA, e uma análise experimental comparando as versões portadas com a implementação original em CUDA e com a versão gerada pela ferramenta Hipify para GPUs AMD. A avaliação será feita com base em tempo de execução, uso de memória e escalabilidade.*

## 1. Introdução

A modelagem reversa no tempo (RTM) é uma técnica amplamente utilizada na exploração sísmica para reconstrução da propagação de ondas e obtenção de imagens do subsolo com alta precisão. *Software* científico de alto desempenho, como a RTM, frequentemente opera em grande escala e exige hardware especializado. Implementações modernas desse método utilizam *GPUs* para acelerar os cálculos, sendo a arquitetura CUDA [Cook 2012] da *NVIDIA* uma escolha comum devido ao seu alto desempenho. No entanto, a dependência de uma tecnologia específica pode dificultar a migração para outras plataformas de hardware, tornando necessário um grande esforço para adequação do código.

Empresas como *NVIDIA* e *AMD* dominam o mercado de *GPUs*, enquanto a *Intel* busca consolidar sua posição nesse segmento. A dependência de uma única plataforma pode impactar as decisões de aquisição de *hardware*, restringindo escolhas que poderiam ser feitas com base apenas no desempenho e custo do equipamento, sem considerar o esforço necessário para portar código existente.

Para atender à crescente demanda por códigos portáteis, que possibilitem a geração de binários compatíveis com múltiplas arquiteturas de *CPUs* e *GPUs* de diferentes fabricantes a partir do mesmo código-fonte, algumas soluções têm sido desenvolvidas e adotadas na comunidade de computação científica. Dentre essas soluções, destacam-se os *frameworks* *Kokkos* [Trott et al. 2021] e *RAJA* [Beckingsale et al. 2019], que oferecem abstrações da execução paralela e facilitam a adaptação do código para diferentes plataformas.

Neste contexto, este trabalho propõe portar uma versão *CUDA* de um simulador de RTM *TTI multi-GPU* para os *frameworks* *Kokkos* e *RAJA*, explorando suas capacidades para tornar a aplicação compatível com múltiplas arquiteturas de *GPUs*, reduzindo a dependência de um único fornecedor de *hardware*.

## 2. Proposta de pesquisa

Este trabalho propõe a portabilidade de uma implementação *CUDA* da modelagem reversa no tempo (*RTM*) para os *frameworks Kokkos* e *RAJA*, com o objetivo de avaliar os impactos dessa adaptação em termos de desempenho e compatibilidade com diferentes arquiteturas.

A pesquisa inicia-se com o estudo detalhado dos *frameworks*, buscando compreender como cada um gerencia memória e executa código paralelo. O foco está em analisar as diferenças em relação ao modelo de programação *CUDA*, além de entender as abstrações oferecidas e as possíveis limitações impostas pela portabilidade.

Com esse conhecimento, o próximo passo é portar o código original do *RTM*, escrito em *CUDA*, para versões em *C++* utilizando *Kokkos* e *RAJA*. O objetivo é permitir sua execução em múltiplas plataformas sem grandes modificações. Durante esse processo, será avaliado o nível de controle sobre recursos específicos das arquiteturas e os impactos das abstrações utilizadas.

Por fim, será realizada uma análise experimental para comparar o desempenho das versões portadas com a implementação original em *CUDA* e com uma versão gerada automaticamente pela ferramenta *Hipify* [AMD 2025] para execução em *GPUs AMD*. A comparação será feita considerando métricas como tempo de execução, uso de memória e escalabilidade, permitindo avaliar a viabilidade do uso desses *frameworks* em aplicações científicas de alto desempenho.

## 3. Cronograma de atividades

O estado atual da pesquisa envolve a compreensão do software-alvo, *RTM TTI CUDA*, e sua reescrita utilizando *Kokkos*. Inicialmente, o código está sendo portado para execução em uma única *GPU*. Resultados iniciais da comparação entre as versões *CUDA* e *Kokkos* em *GPU* única devem estar disponíveis até o fim de março. A próxima etapa consiste na divisão da carga de trabalho entre múltiplas *GPUs* em *Kokkos*, seguida da reescrita do código original para *RAJA*. Resultados de todas as versões devem estar prontos até o fim de maio.

## References

- AMD (2025). *HIP: C++ Heterogeneous-Compute Interface for Portability*. <https://github.com/ROCm/HIP>.
- Beckingsale, D. A., Burmark, J., Hornung, R., Jones, H., Killian, W., Kunen, A. J., Pearce, O., Robinson, P., Ryujin, B. S., and Scogland, T. R. W. (2019). *RAJA: Portable Performance for Large-Scale Scientific Applications*. In *2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*, pages 71–81.
- Cook, S. (2012). *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition.
- Trott, C., Berger-Vergiat, L., Poliakoff, D., Rajamanickam, S., Lebrun-Grandie, D., Madsen, J., Al Awar, N., Gligoric, M., Shipman, G., and Womeldorff, G. (2021). The kokkos ecosystem: Comprehensive performance portability for high performance computing. *Computing in Science Engineering*, 23(5):10–18.