

# Um Framework para Criar Benchmarks de Aplicações Paralelas de Stream

Adriano Marques Garcia<sup>1</sup>, Dalvan Griebler<sup>1</sup>, Claudio Schepke<sup>2</sup>, Luiz G. Fernandes<sup>1</sup>

<sup>1</sup> Escola Politécnica, Grupo de Modelagem de Aplicações Paralelas (GMAP), Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, Brasil.

adriano.garcia@edu.pucrs.br

<sup>2</sup> Universidade Federal do Pampa (UNIPAMPA), Alegrete, RS, Brasil

**Resumo.** *Este trabalho apresenta o SPBENCH, um framework para o desenvolvimento de benchmarks de processamento de stream em C++. O SPBENCH fornece um conjunto de aplicações realísticas através de abstrações de alto nível e permite customizações nos dados de entrada e métricas de desempenho.*

## 1. Introdução

Aplicações de processamento de *stream* recebem fluxos contínuos de dados e, idealmente, os processam em aproximadamente tempo real. O paralelismo nesse paradigma é explorado através da implementação de operadores ou estágios, que formam um grafo acíclico dirigido. Diferentes padrões de paralelismo podem ser expressados nesses grafos, como paralelismo de dados, tarefas e *pipeline* [Andrade et al. 2014]. Implementar paralelismo em processamento de *stream* pode ser uma tarefa desafiadora, mesmo utilizando Interfaces de Programação Paralela (IPPs) que provêm padrões paralelos estruturados ou abstrações de alto-nível, como Intel TBB, FastFlow [Aldinucci et al. 2017] e SPar [Griebler et al. 2017]. Aplicações desse domínio podem ter milhares de linhas de código, o que dificulta reconhecer todos os operadores/estágios e identificar todas as dependências de dados entre eles. O comportamento dos dados de entrada também pode variar muito durante a execução, impactando no desempenho. A avaliação de desempenho dessas aplicações na literatura também não costuma seguir uma metodologia clara e padronizada.

A crescente demanda aliada ao aumento no poder dos processadores multinúcleo traz consigo o desenvolvimento de novas tecnologias focadas nessas arquiteturas e em linguagens como o C++, como novas bibliotecas, *frameworks*, e diferentes técnicas de paralelismo e otimização. Existe uma lacuna em relação *benchmarks*. Estes são importantes, pois fornecem uma plataforma para pesquisadores testarem novas tecnologias e também servem como um parâmetro de avaliação e qualidade. Portanto, visando ajudar a preencher essa lacuna foi proposto SPBENCH (*stream processing benchmark*), que tem como objetivo facilitar a criação de *benchmarks* para processamento de *stream* em C++.

## 2. SPBENCH

SPBENCH permite que usuários possam facilmente utilizar diferentes IPPs para implementar paralelismo em aplicações de processamento de *stream*, criando *benchmarks* personalizados. O *framework* provê uma API em C++ que possibilita aos usuários acessarem uma versão simplificada das aplicações suportadas pelo SPBENCH. A API automaticamente adiciona aos *benchmarks* diferentes métricas de desempenho, como latência, *throughput* (vazão de dados), e consumo de recursos, em diferentes níveis de profundidade e granularidade. Além disso, são disponibilizados métodos para o usuário controlar o comportamento dos dados, como a origem deles (disco, memória, rede), o tamanho (usando

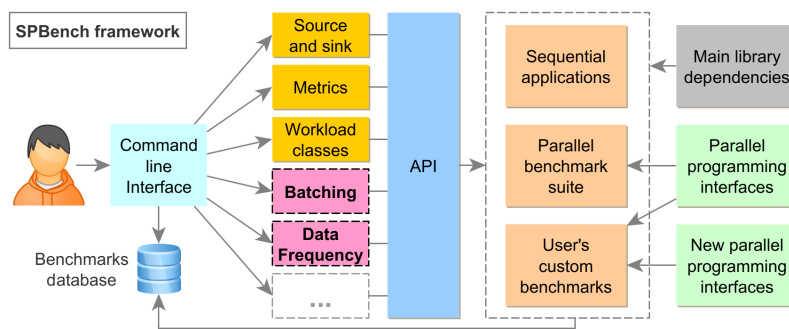


Figura 1. Framework SPBENCH.

*micro-batching*), a frequência dos dados, e também instanciar múltiplas fontes. Também são incluídas diferentes classes de *workload* para cada aplicação.

O *framework* SPBENCH [Garcia et al. 2022] é apresentado na Figura 1 e apresenta duas novas funcionalidades da API (*micro-batching* e frequência de dados) que foram implementadas recentemente e estão sob estudo e análise. *Micro-batching* é uma técnica de otimização que permite aumentar *throughput* em troca de latência e variar a frequência de dados permite representar cenários realísticos e mais complexos. Diferentemente dos *benchmarks* tradicionais, no SPBENCH a interação com o usuário ocorre totalmente através de uma Interface de Linha de Comando (CLI). O SPBENCH mantém uma base de dados com *benchmarks* previamente criados. Através da CLI o usuário pode acessar, compilar e executar esses *benchmarks*, podendo também modificá-los para testar diferentes técnicas ou estratégias de paralelismo. A CLI também permite que usuários criem novos *benchmarks* e adicionem dependências de compilação. Portanto, o SPBENCH é totalmente modular e o código paralelo pode ser rapidamente replicado para criar diferentes *benchmarks*.

Dessa forma, uma das grandes vantagens do SPBENCH é permitir que os usuários possam focar em escrever e otimizar o código paralelo, sem perder tempo com aspectos menos relevantes, complexos e de mais baixo nível de cada aplicação. Devido à forma simples como as aplicações são apresentadas, o SPBENCH também tem potencial para ser utilizado como ferramenta de ensino/aprendizagem de programação paralela para processamento de *stream*. Atualmente o SPBENCH disponibiliza quatro aplicações, incluindo Ferret da suite PARSEC. Também são disponibilizados *benchmarks* pra todas as aplicações utilizando Intel TBB, FastFlow e SPar. Diferentes experimentos já foram conduzidos e apresentados em [Garcia et al. 2022]. O *framework* segue em constante desenvolvimento e no futuro próximo adicionaremos novas aplicações, novas IPPs e outras funcionalidades.

## Referências

- Aldinucci, M., Danelutto, M., Kilpatrick, P., and Torquati, M. (2017). *Fastflow: High-Level and Efficient Streaming on Multicore*, chapter 13, pages 261–280. John Wiley & Sons.
- Andrade, H. C., Gedik, B., and Turaga, D. S. (2014). *Fundamentals of stream processing: application design, systems, and analytics*. Cambridge University Press.
- Garcia, A. M., Griebler, D., Schepke, C., and Fernandes, L. G. (2022). SPBench: a framework for creating benchmarks of stream processing applications. *Computing*, 104(1):1–23. Springer.
- Griebler, D., Danelutto, M., Torquati, M., and Fernandes, L. G. (2017). SPar: A DSL for High-Level and Productive Stream Parallelism. *Parallel Processing Letters*, 27(01):1740005.