Uma Suíte de *Benchmarks* Parametrizáveis para o Domínio de Processamento de Stream em Sistemas Multi-Core

Carlos A. F. Maron[1], **Luiz Gustavo Fernandes**[1]

[1] Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Grupo de Modelagem de Aplicações Paralelas (GMAP), Porto Alegre – RS – Brasil

carlos.maron@acad.pucrs.br,luiz.fernandes@pucrs.br

Resumo. *Avaliar o desempenho é importante para computação. Porém, assim como o hardware, o software também deve ser avaliado quando características podem influenciar no seu comportamento. Nestes casos, a suíte de benchmarks parametrizáveis para o processamento de stream serve como uma ferramenta de apoio ao usuário e até programadores.*

1. Introdução

Aplicações do domínio de *stream* estão se destacando nos últimos tempos. Tanto na indústria como na academia, estas aplicações se tornaram cada vez mais recorrentes no processamento de informações. Alguns exemplos destas aplicações estão presentes em nosso cotidiano, como aplicações que processam imagem, vídeo e áudio, redes sociais, casas inteligentes, *Big Data*, entre outras. As aplicações deste domínio processam um *stream*, formado por conjuntos infinitos de elementos (Ex, imagens, *frame*, estrutura, etc.). Tais aplicações têm um comportamento semelhante ao de uma linha de produção automobilística, onde existe uma sequência de postos de trabalhos que realizam uma tarefa sobre cada um dos carros que entram nesta linha. No código de uma aplicação, os postos de trabalhos são os estágios que realizam as operações sobre cada um dos elementos.

Paralelizar estas aplicações pode ser um desafio para encontrar a melhor configuração de desempenho. Cada aplicação tem particularidades ao lidar com o seu fluxo contínuo, pois muitas propriedades são variáveis. Por exemplo, os estágios podem realizar computações diversificadas, com ou sem estado interno, elementos do *stream* ter tamanhos e tipos diferentes. Além disso, algumas aplicações podem exigir diferentes propriedades no desempenho, como vazão ou latência. Desse modo, avaliar o desempenho neste cenário pode ser complicado, já que paralelizar estas aplicações é um desafio [Griebler et al. 2017, Griebler et al. 2018] e algumas propriedades podem impactar no desempenho.

No entanto, ao deparar-se com as clássicas suítes de *benchmarks*, como NAS [Bailey et al. 1991] e PARSEC [Bienia 2011], que possuem aplicações do domínio de *stream*, percebe-se que essas aplicações não refletem totalmente o comportamento e a paralelização de aplicações deste tipo. Além disso, são *benchmarks* usados para mostrar o desempenho da infraestrutura e não abordam as propriedades das aplicações de *stream*, além de que a parametrização nestas suítes é limitada. Basicamente, resumem-se na definição do número de *threads*, no tamanho da entrada ou complexidade do processamento.

2. Proposta

Este trabalho propõem a suíte BenSP - *Benchmark for Stream Parallelism*. Uma suíte de *benchmarks* para o domínio de processamento de *stream* em sistemas *multi-core*.

Com o BenSP as propriedades de *stream* podem ser testadas em uma aplicação por meio de parâmetros. Essa característica da suíte foi inspirada no *microbenchmark* Eigenbench [Hong et al. 2010], que deriva características e comportamentos de sistemas de memória transacional por meio de parâmetros.

A Figura 1 mostra a possibilidade de parametrização da aplicação Dedup na suíte BenSP. Em aplicações como o Dedup, utilizadas para remoção de dados duplicados e compressão de arquivos [Bienia 2011], o processo de fragmentação define o tamanho médio do elemento de *stream* (*chunk*), resultando em diferentes comportamentos como o aumento do uso de memória, tempo de execução, taxa de deduplicação e compressão. Testes realizados com a parametrização mostraram um ganho no tempo de execução de 67%.

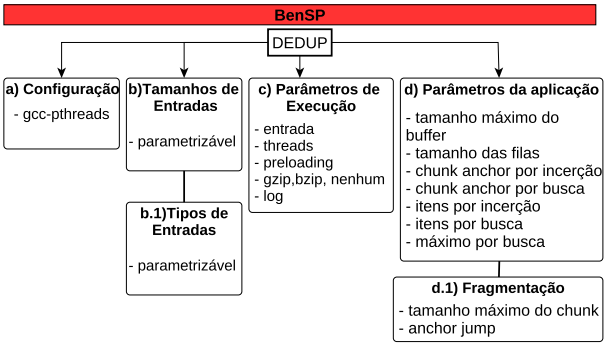

Figura 1. Parametrização Suíte BenSP.

Além do Dedup, aplicações como Ferret e DPI (*Deep Packet Inspection*) também fazem parte da suíte BenSP. Por exemplo, o elemento de *stream* (*frame*) no Ferret é um dos parâmetros que pode ser ajustado. Neste caso, como a aplicação processa a busca de similaridade em imagens, vídeo e áudio, isso influenciará na precisão e no tempo de execução da aplicação. No DPI, o tamanho do elemento de *stream* (pacote de rede) também pode ser ajustado, inferindo no reconhecimento do tráfego de rede, por exemplo.

Referências

[Bailey et al. 1991] Bailey, D. H. et al. (1991). The NAS Parallel Benchmarks. *International Journal of High Performance Computing Applications*, 5(3):63–73.

[Bienia 2011] Bienia, C. (2011). *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University.

[Griebler et al. 2017] 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.

[Griebler et al. 2018] Griebler, D., Filho, R. B. H., Danelutto, M., and Fernandes, L. G. (2018). High-Level and Productive Stream Parallelism for Dedup, Ferret, and Bzip2. *International Journal of Parallel Programming*, pages 1–19.

[Hong et al. 2010] Hong, S., Oguntebi, T., Casper, J., Bronson, N., Kozyrakis, C., and Olukotun, K. (2010). Eigenbench: A simple exploration tool for orthogonal tm characteristics. In *Workload Characterization (IISWC), 2010 IEEE International Symposium on*, pages 1–11.