

Proposta de uma Suíte de Benchmarks para Processamento de Stream em Sistemas Multi-Core

Adriano Marques Garcia¹, Dalvan Griebler¹, Luiz Gustavo Fernandes¹

¹ 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

Resumo. *O aumento no volume de dados gerados por sistemas computacionais e a necessidade por processamento rápido desses dados vem alavancando a área de processamento de stream. Entretanto, ainda não existe um benchmark para auxiliar desenvolvedores e pesquisadores. Este trabalho visa propor uma suíte de benchmarks para processamento de stream em arquiteturas multi-core e discute as características necessárias no desenvolvimento dessa suíte.*

1. Introdução

Com os conjuntos de dados atingindo escalas de terabytes e petabytes e um aumento na demanda por análises em tempo real, o processamento tradicional de dados orientado a lotes (*batch processing*) não é suficiente. Para acompanhar o fluxo de dados recebidos, as organizações estão cada vez mais adotando o processamento de *stream*. No processamento de *stream* os dados são processados continuamente à medida que novos dados ficam disponíveis para análise, aplicando uma série de filtros (estágios em um *pipeline*) e mantendo o resultado final atualizado de acordo com cada novo dado processado [Andrade et al. 2014]. Portanto, cada novo resultado está disponível em tempo real, geralmente com uma latência de alguns milissegundos. Embora o processamento de *stream* não precise ser em tempo real, ele pode permitir o processamento e a análise de dados usados por muitos sistemas de tempo real. Esses, podem exigir reações rápidas aos dados de entrada, como monitoramento de cliques em páginas na internet, análise financeira em pregão da bolsa de valores, ou dados de sensores de um dispositivo de Internet das Coisas (IoT), por exemplo [Andrade et al. 2014].

Para avaliar a capacidade de processamento de diferentes tipos de dispositivos, que executam vários tipos de aplicações, usando técnicas e linguagens de implementação distintas, normalmente se usam *benchmarks*. Na computação, um *benchmark* pode ser descrito como um ou mais programas que representam determinados cenários e que servem como uma métrica ou um parâmetro de comparação [Obaidat and Boudriga 2010]. No entanto, essa é uma área em crescimento e muitas das aplicações recentes carecem de *benchmarks* representativos. Dessa forma, avaliar o comportamento de uma nova técnica, linguagem, aplicação etc. em relação ao estado da arte sem o uso de um *benchmark* pode ser uma tarefa difícil e que abre espaço para questionamentos. *Benchmarks* resolvem parte desse problema, fornecendo métricas, ferramentas e aplicações que são reconhecidas pelos pesquisadores e desenvolvedores da área. Neste trabalho é proposta a criação de uma suíte de *benchmarks* para processamento paralelo de *stream* em arquiteturas multi-core.

2. Benchmarks para Processamento de Stream

Existem muitas limitações em termos de *benchmarks* para processamento de *stream*, como a falta de aplicações representativas e exploração de paralelismo. Poucos *benchmarks* paralelos incluem algumas aplicações do domínio de *stream*, como o NPB e o PARSEC, por exemplo. Aliado à isso, também existem diversos paradigmas de programação que são utilizados e que aumentam a complexidade de se desenvolver um *benchmark* abrangente. Reactive, dataFlow e paralelismo de *stream* são alguns desses paradigmas. O paralelismo de *stream* (*stream parallelism*) é um paradigma que herda as características dos demais e permite que se tenha estágios de um *pipeline* executando em paralelo. Isso pode melhorar o desempenho de uma aplicação, aumentando o *throughput* e diminuindo a latência, que são fatores importantes nesta era de informação em tempo real.

Existem alguns trabalhos relacionados com propostas similares, mas esses focam em sistemas distribuídos ou propõem um *microbenchmark* altamente parametrizável para simular o comportamento de outras aplicações. Portanto, ainda não há uma suíte abrangente de *benchmarks* para processamento paralelo de *stream* em sistemas multi-core.

3. Proposta de uma Suíte de Benchmarks

Um *benchmark* precisa ser bem aceito pela comunidade para se tornar um ponto de referência. Para que isso ocorra, é importante que o *benchmark* seja representativo para o estado da arte, ou seja, deve possuir características que representem diferentes tipos de aplicações, linguagens, paradigmas, etc. do mundo real. Na próxima etapa desta pesquisa faremos um mapeamento para identificar essas características. Buscaremos trabalhos que utilizaram aplicações do domínio de Processamento de Stream em seus experimentos para elaborar um mapa com as principais características encontradas. A Figura 1 apresenta um possível modelo desse mapa e exemplos de características que podem ser encontradas.

Aplicações Ex.: Processamento de vídeo, Monitoramento, Deduplicação etc.	Paradigmas Ex.: Stream parallelism, data stream, DataFlow etc.	Workloads Ex.: Diferentes balanceamentos, frequência de dados, complexidade etc.
Linguagens Ex.: C/C++, Scala, SPar etc.	Bibliotecas Ex.: TBB, FastFlow, PThreads etc.	Padrões Paralelos Ex.: map, farm, janela deslizante etc.

Figura 1. Exemplos de possíveis características do *benchmark* proposto.

Nossa pesquisa já possui algumas aplicações desenvolvidas por colaboradores para suprir a falta de um *benchmark* e que poderão fazer parte da suíte (as aplicações Dedup, Ferret e Bzip2, por exemplo). Com a definição das características mais representativas identificadas na próxima etapa do estudo, poderemos filtrar ou modificar essas aplicações, ou ainda fazer novas implementações de acordo com os resultados obtidos.

Referências

- Andrade, H. C., Gedik, B., and Turaga, D. S. (2014). *Fundamentals of stream processing: application design, systems, and analytics*. Cambridge University Press.
- Obaidat, M. S. and Boudriga, N. A. (2010). *Fundamentals of performance evaluation of computer and telecommunication systems*. John Wiley & Sons.