

# Adicionando Elasticidade a Padrões Paralelos

Cristiane de Andrade<sup>1</sup>, Guilherme Galante<sup>1</sup>

<sup>1</sup>Ciência da Computação – Universidade Estadual do Oeste do Paraná (UNIOESTE)  
Caixa Postal – 85.819-110 – Cascavel – PR – Brazil

candradesph@gmail.com, guilherme.galante@unioeste.br

***Resumo.** Padrões paralelos têm sido propostos para esconder algumas complexidades da construção de aplicações paralelas. Entretanto, não há padrões abordando o desenvolvimento de aplicações paralelas elásticas. Este trabalho visa estender os padrões apresentados na literatura propondo um modelo para o desenvolvimento de aplicações com suporte à elasticidade.*

## 1. Introdução

Um padrão paralelo é um princípio orientador que ajuda os desenvolvedores a paralelizar uma aplicação sugerindo como implementar computação concorrente. Usando essa abordagem, os desenvolvedores podem selecionar facilmente um padrão paralelo apropriado e seguir um padrão bem estabelecido método para dividir um programa em blocos para aproveitar de paralelismo.

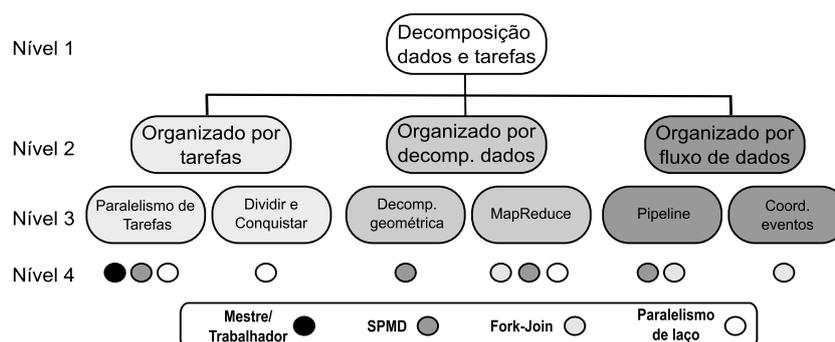
Nesse contexto, destacam-se as iniciativas de [Mattson et al. 2004] e [McCool et al. 2012] que propuseram padrões paralelos para estruturar o desenvolvimento de programas paralelos. O primeiro é voltado principalmente para o projeto arquitetônico de aplicativos, enquanto o segundo apresenta uma abordagem de que se concentra na estratégia de algoritmo (ou esqueletos). No entanto, atualmente não há na literatura padrões de programação paralelas que abordem a implementação da elasticidade. Nesse sentido, este trabalho unifica os trabalhos citados e adiciona padrões para a exploração da elasticidade em aplicações paralelas.

## 2. Unificando padrões paralelos

Baseado nos trabalhos de [Mattson et al. 2004] e [McCool et al. 2012], propõem-se uma unificação, juntando elementos de ambos os modelos. Como resultado dessa unificação, obteve-se a árvore de decisão apresentada Figura 1, tem como objetivo orientar a escolha dos padrões a serem implementados. No primeiro nível temos a forma como o algoritmo é estruturado, no segundo nível temos os padrões de estruturas de algoritmos propostos, e no terceiro nível temos os padrões de implementação destes algoritmos. No Nível 4 encontram-se quatro padrões de estrutura de apoio: Mestre/Trabalhados, SPMD, Fork-Join e Paralisismo de laço, os quais descrevem abordagens de estruturação do código-fonte da aplicação, pensando já nos mecanismos que serão utilizados para implementação efetiva do algoritmo.

## 3. Definindo padrões de elasticidade

A elasticidade é considerada uma propriedade muito importante da computação em nuvem porque permite que os recursos sejam escalonados dinamicamente conforme sua carga de trabalho aumentar ou diminuir em tempo de execução.



**Figura 1. Padrões paralelos unificados**

Baseados nos quatro padrões de estrutura de apoio, é possível definir padrões para a exploração da elasticidade em cada um deles. No padrão *Mestre/Trabalhador*, é possível adicionar novos mestres e trabalhadores à aplicação de modo a tratar diferentes aspectos. Ao se adicionar novos mestres pode-se solucionar possíveis gargalos de processamento. Por sua vez, a alocação dinâmica de trabalhadores garante que a aplicação consiga ter um desempenho de acordo com sua carga de trabalho. A elasticidade pode colaborar na melhoria do padrão *SPMD*, permitindo a escalabilidade, o balanceamento de carga, a eficiência na utilização de recursos e a adaptação a variações de carga. A capacidade de ajustar dinamicamente os recursos segundo as demandas da aplicação, ajuda a otimizar o desempenho e a eficiência do sistema *SPMD*. A estrutura de um programa *Fork/Join* possui um comportamento dinâmico natural, uma vez que as tarefas são bifurcadas (geralmente em forma de árvore) e unidas em tempo de execução. Assim, ao se utilizar uma abordagem elástica, pode-se começar a alocação pelos recursos para a raiz da árvore e, a cada divisão da carga de trabalho, criam-se recursos adicionais para mapear as tarefas. À medida que os resultados se fundem, os recursos que estiverem ociosos, podem ser desalocados. No padrão de *Paralelismo de Laço*, a elasticidade pode ser explorada na alocação de recursos apropriados para a execução laço. Assim, pode-se adequar os recursos utilizados de acordo com o número de iterações que serão executadas.

É importante destacar que é necessário usar frameworks que dêem suporte a cada um dos modelos de elasticidade, de modo a interagir com os controladores das máquinas subjacentes, por exemplo, gerenciadores de recursos de um cluster ou cloud [Galante and da Rosa Righi 2022].

## Referências

- Galante, G. and da Rosa Righi, R. (2022). Adaptive parallel applications: from shared memory architectures to fog computing (2002–2022). *Cluster Computing*, 25(6):4439–4461.
- Mattson, T. G., Sanders, B., and Massingill, B. (2004). *Patterns for parallel programming*. Pearson Education.
- McCool, M., Reinders, J., and Robison, A. (2012). *Structured parallel programming: patterns for efficient computation*. Elsevier.