

Padrões de programação paralela em nuvens computacionais

Cristiane de Andrade¹, Guilherme Galante¹

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

{cristiane.andrade, guilherme.galante}@unioeste.br

Resumo. *Padrões de computação paralela tem sido propostos com o objetivo de auxiliar desenvolvedores a tirar proveito do paralelismo em arquiteturas modernas. Este trabalho propõem apresentar padrões de projeto para o desenvolvimento de aplicações paralelas e uma classificação e definição desses padrões. E, além disso, estender este padrões propostos para que eles contemplem a elasticidade, realizando uma revisão dos trabalhos que abordem elasticidade e aplicações paralelas, com o objetivo de propor padrões de programação paralela para aplicações elásticas.*

1. Introdução

Padrões de programação paralela tem por objetivo especificar e definir a estrutura algorítmica, tratando as questões de acesso aos dados e os detalhes do ambiente arquitetural a ser paralelizado, provendo maior eficiência na utilização dos recursos disponíveis, sem que a aplicação fique dependente de um hardware específico. Desenvolver aplicações paralelas não é uma tarefa trivial, e esses padrões buscam trazer um direcionamento para o desenvolvedor em relação a análise de requisitos e nos cuidados em relação a programabilidade dos algoritmos paralelos.

Nesse contexto, dois trabalhos se destacam na proposição de soluções, um deles apresentado por [Mattson et al. 2004] e outro apresentado por [McCool et al. 2012]. [Mattson et al. 2004] organizam o projeto de desenvolvimento de aplicações paralelas em quatro fases, facilitando que se encontre o tipo concorrência existente na aplicação, as sessões críticas e trazem uma orientação sobre o melhor padrão a aplicar. Já [McCool et al. 2012] trazem uma definição de diversos padrões de programação paralelas.

Nesses padrões uma característica que não é abordada, é a elasticidade. A elasticidade pode ser definida como a capacidade de gerenciar recursos computacionais em tempo de execução, alocando-os ou liberando-os rapidamente de acordo com a demanda [Galante and Bona 2014]. A utilização da elasticidade se tornou possível com advento da computação em nuvem, que proporciona um modelo de computação *pay-per-use*, o que a torna uma ótima opção para execução de aplicações paralelas, pois o custo é somente dos recursos realmente utilizados. Neste contexto, o objetivo deste trabalho é propor a unificação e adaptação destes dois padrões de projeto para o desenvolvimento de aplicações paralelas, e ainda estender este padrão unificado para abordar a questão da elasticidade.

2. Padrões de projeto para aplicações paralelas

[Mattson et al. 2004] propõem padrões para programação paralela, organizando o projeto de desenvolvimento de aplicações paralelas em quatro fases: encontrando a concorrência,

estrutura do algoritmo, estrutura de suporte e mecanismos de implementação.

[McCool et al. 2012] trazem em seu livro a proposta de padrões de computação paralela, com foco para desenvolvedores de C e C++, mas afirmam que os padrões podem ser aplicados para qualquer tipo de aplicação, independente da linguagem e do tipo de aplicação.

Apresentam ainda a classificação e exemplos de implementação de cada padrão. O conceito utilizado para a implementação dos padrões, é o conceito de aninhamento. [McCool et al. 2012] propõem que, onde há “caixas de tarefas” é possível substituir por qualquer outro padrão de programação sequencial, e isso permite a construção de programas paralelos determinísticos. São apresentados diversos padrões de programação, que estão organizados da seguinte forma: padrões de fluxo de controle serial estruturado; padrões de controle paralelo; padrões de gerenciamento de dados seriais e padrões de gerenciamento de dados paralelos.

3. Incluindo Elasticidade no Contexto de Padrões de Projeto

Algumas características intrínsecas da computação em nuvem, tornam o desenvolvimento de aplicações paralelas, para este modelo de computação, um desafio. Dentre estas características estão: heterogeneidade do *hardware*, que não é controlada pelo usuário ou desenvolvedor, a utilização da elasticidade, em tempo de execução e o balanceamento de carga.

A criação de padrões de programação tem por objetivo definir abstrações no desenvolvimento das aplicações, para que sejam portáteis, não dependam de uma estrutura de *hardware* específica, e que seja de fácil manutenção e flexíveis para possam se beneficiar do uso da elasticidade.

Após a realização de revisão das duas abordagens, avaliando os pontos em comum e os pontos distintos entre elas, objetiva-se unificar a proposta de ambos os trabalhos. Num segundo momento, para estender estes padrões unificados, com o objetivo que eles contemplem a elasticidade, será feita uma revisão dos trabalhos que abordem elasticidade, e então, baseado nisso, propor padrões de programação paralela para aplicações elásticas. Para fazer a validação desses padrões, a proposta é analisar alguns trabalhos presentes no estado da arte, para analisar como esses padrões vem sendo utilizados nessas aplicações paralelas elásticas. Faremos uma avaliação do resultado obtido nesses trabalhos com esses padrões.

Espera-se com este trabalho apresentar padrões de projeto para o desenvolvimento de aplicações paralelas e uma classificação e definição desses padrões, para que as aplicações possam se beneficiar da elasticidade provisionada pela computação em nuvem.

Referências

- Galante, G. and Bona, L. C. (2014). Supporting elasticity in openmp applications.
- 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.