Refinando e Balanceando o Particionamento de Aplicações Científicas baseadas em Tarefas em Plataformas Heterogêneas

Lucas Leandro Nesi[1]*, **Arnaud Legrand**[2], **Lucas Mello Schnorr**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970, Porto Alegre – RS – Brasil

[2]Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG
F-38000, Grenoble – France

{lucas.nesi, schnorr}@inf.ufrgs.br

arnaud.legrand@imag.fr

Resumo. *Aplicações científicas podem sofrer com o desbalanceamento de carga causado pelo seu próprio algoritmo, ou pelas plataformas computacionais heterogêneas. Abordagens com escalonamento dinâmico, como o paradigma orientado a tarefas, podem auxiliar na programação e na redução deste desbalanceamento. Este trabalho visa estudar estratégias, para* runtimes *baseados em tarefas, para melhor balancear o particionamento destas aplicações.*

1. Introdução

As plataformas de computação de Alto Desempenho proveem os recursos computacionais necessários para diversas aplicações científicas. Para aproveitar tais plataformas, é necessário realizar um particionamento do domínio do problema dessas aplicações, fazendo com que cada recurso trabalhe sobre um fragmento do problema. Nestes cenários, é fundamental que a execução das aplicações permaneça balanceada entre os recursos, evitando tempos ociosos. Assim, todos os recursos devem contribuir para a solução proporcionalmente ao seu poder computacional. O particionamento e o balanceamento são tarefas complexas pois exigem uma divisão proporcional dos fragmentos sem desconsiderar as comunicações obrigatórias causadas por essa divisão. A divisão se torna mais complexa ao empregar plataformas com recursos de processamento diversos e poder computacional diferentes [Beaumont et al. 2019]. Tanto o algoritmo da aplicação ou a plataforma computacional podem ser a origem das causas para este problema. O **desbalanceamento devido ao algoritmo da aplicação** pode ser encontrado em projetos recentes, tais como da fatoração LU/Cholesky [Pinto et al. 2018] ou Ondes3D [Tesser et al. 2017]. No caso de Cholesky, a diferente quantidade de tarefas aplicadas a cada bloco do particionamento causa o problema. No Ondes3D, a aplicação divide igualmente os dados entre os recursos e apresenta um grande desbalanceamento temporal e espacial. Outra fonte de **desbalanceamento tem origem na própria plataforma computacional**. As plataformas HPC estão cada vez mais heterogêneas, adicionando mais complexidade na programação das aplicações paralelas [Dongarra et al. 2017]. Esta heterogeneidade pode ser intra nó, com aceleradores associados às CPUs, ou entre nós, onde os nós computacionais são diferentes. Além disso, estudos recentes mostram que nós homogêneos apresentam variações de

*Bolsa da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES); Finance Code 001.

desempenho por causa do processo de manufatura [Inadomi et al. 2015]. Uma solução possível para esses desequilíbrios é gerenciar dinamicamente a carga das aplicações.

Em aplicações MPI tradicionais, normalmente tanto o mapeamento de dados e o escalonamento de tarefas é estático. A vantagem desta abordagem é remover sobrecargas associadas ao gerenciamento dinâmico. Entretanto, instabilidades no algoritmo ou problemas na plataforma podem levar a diferenças do modelo estático utilizado. Uma alternativa com abordagem dinâmica é o paradigma de programação baseado em tarefas. Neste paradigma, todas as decisões são tomadas por um *runtime*. O programador utiliza uma forma declarativa de programação e divide a computação em tarefas com dependências entre elas, por meio de um Grafo Acíclico Dirigido (DAG). Além do escalonamento de tarefas dinâmico, o *runtime* é responsável pelo mapeamento de dados, e pode ser feito de forma estática ou dinâmica. Um exemplo de *runtime* baseado em tarefas é o StarPU [Augonnet et al. 2011]. Em um único nó, o StarPU pode utilizar múltiplos escalonadores para mapear computação em CPUs e aceleradores automaticamente movendo os dados, onde tanto o escalonamento de tarefas quanto o mapeamento de dados são dinâmicos. Entretanto, para múltiplos nós, utilizando o módulo StarPU-MPI, é necessário que os dados sejam mapeados estaticamente entre os recursos e as tarefas sejam escalonadas sobre eles. Este comportamento justifica um bom particionamento estático inicial.

Proposta de Trabalho: Este trabalho visa estudar estratégias dinâmicas para o refinamento do particionamento de aplicações científicas para os *runtimes* baseados em tarefas em plataformas heterogêneas. O StarPU-MPI será utilizado como *runtime* baseado em tarefas. Inicialmente será realizado o estudo com aplicações de álgebra linear densa. A aplicação começa a execução com um particionamento estático previamente calculado, e o *runtime* se torna responsável por verificações e balanceamento dinâmico. Esta verificação pode ser realizada de maneira local, distribuída, ou centralizada. Identificando os desbalanceamentos, este trabalho propõe investigar métodos em como o *runtime* pode balancear a computação, movendo dados, considerando informação do DAG, carga atual, ou estados dos recursos próximos.

Referências

Augonnet, C. et al. (2011). StarPU: A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures. *Conc. Comp.: Pract. Exp., SI:EuroPar 2009*, 23.

Beaumont, O. et al. (2019). Recent advances in matrix partitioning for parallel computing on heterogeneous platforms. *IEEE Transactions on Parallel and Distributed Systems*.

Dongarra, J. et al. (2017). With extreme computing, the rules have changed. *Computing in Science Engineering*, 19(3):52–62.

Inadomi, Y. et al. (2015). Analyzing and mitigating the impact of manufacturing variability in power-constrained supercomputing. In *SC '15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*.

Pinto, V. G. et al. (2018). A visual performance analysis framework for task based parallel applications running on hybrid clusters. *Conc. Comp.: Pract. Exp.*

Tesser et al. (2017). Performance modeling of a geophysics application to accelerate over-decomposition parameter tuning through simulation. *Conc. Comp.: Pract. Exp.*