

Utilização de aprendizado por reforço para adaptar a quantidade de recursos e acelerar aplicações

Lucas Leandro Nesi^{1,2*}, Arnaud Legrand², Lucas Mello Schnorr¹

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

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

{lucas.nesi, schnorr}@inf.ufrgs.br, arnaud.legrand@imag.fr

Resumo. *O desempenho de aplicações paralelas depende da quantidade de recursos utilizados. Entretanto, recursos excessivos são frequentemente prejudiciais, uma vez que pode acontecer comunicação excessiva, sincronizações, e comportamentos difíceis de antecipar. Neste artigo, estudamos estratégias de aprendizado por reforço para que no contexto de uma aplicação baseada em tarefas, esta se adapte dinamicamente ao melhor conjunto de nós heterogêneos.*

1. Introdução

A heterogeneidade é uma realidade em recursos HPC. Ela está presente intra-nó, com a combinação de CPU e aceleradores, e também ao nível de sistema, onde existem diferentes nós computacionais [HPE et al. 2022]. Ainda, aplicações HPC podem ter várias fases computacionais com requisitos computacionais diferentes [Nesi et al. 2021]. A complexidade de portabilidade e de corretamente utilizar todos os recursos intra-nó motivaram a adoção do paradigma de programação baseado em tarefas. Neste, a aplicação é estruturada como um Gráfico Acíclico Dirigido (DAG) de tarefas, escalonados dinamicamente por um *runtime* para os recursos [Thibault 2018].

Embora o paradigma baseado em tarefas atenua a sobrecarga de comunicação, efeitos imprevisíveis (contenção, sincronizações) continuam a ser possíveis e particularmente difíceis de modelar *a priori*, especialmente quando se explora conjuntos heterogêneos de nós [Nesi et al. 2022]. Como a quantidade de nós é diretamente relacionada ao desempenho e a estes efeitos, métodos que aprendem e se adaptam são desejáveis.

Utilizando a aplicação iterativa ExaGeoStat [Abdulah et al. 2018], que atua sobre dados geo-estatísticos e possui duas fases predominantes, foi estudado estratégias para guiar a escolha da quantidade de nós a ser utilizada em umas das fases em cada iteração. O método de aprendizado por reforço do Processo Gaussiano (GP) com o *Upper Confidence Bound* (UCB) assume dados correlacionados e modela possíveis realizações de funções sobre eles [Gramacy 2020]. Ele serve como um *surrogate*, onde a aplicação pode escolher a melhor opção e explorá-la. Estendemos ele para o **GP-Discont**, adicionando conhecimentos de HPC, como a adição de limites de busca, definição de uma tendência, e modelagem de discontinuidades do comportamento entre partições de nós. Assim, a cada iteração, a aplicação escolhe uma quantidade de nós, recalculando e melhorando o GP.

*Bolsista do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

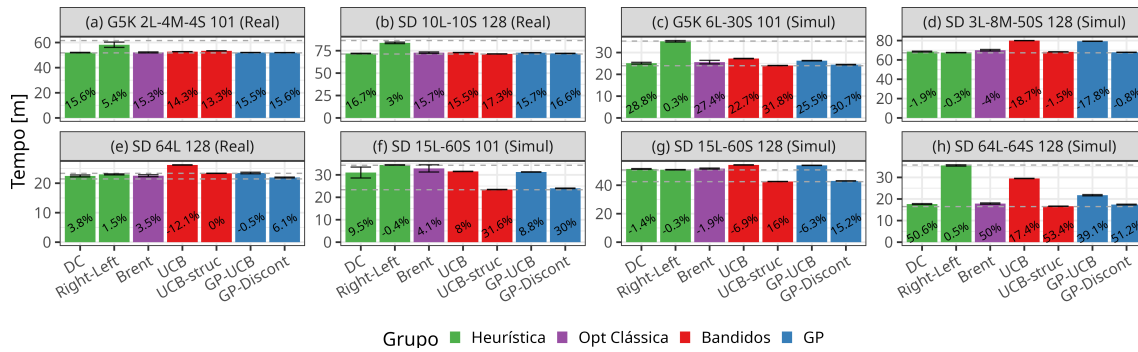


Figura 1. Resultados das estratégias em diferentes cenários de heterogeneidade.

A Figura 1 apresenta oito casos estudados (facetas), onde foram utilizadas oito estratégias candidatas para aprender a melhor quantidade de nós para uma das fases. Cada faceta apresenta uma quantidade de nós de três possíveis categorias (L, M ou S) no supercomputador Santos Dumont ou no Grid5000. Foram utilizadas execuções reais (Real) ou simulações (Simul). O eixo Y é o tempo de execução de toda a aplicação utilizando a estratégia para variar a quantidade de nós dinamicamente durante as iterações. O eixo X mostra as diferentes estratégias de exploração. **DC** é uma estratégia de *divide and conquer*, **Right-Left** utiliza todos os nós e vai diminuindo até a situação piorar, **Brent** é uma otimização para dados contínuos, **UCB** e **UCB-struct** utilizam o método de bandidos e uma versão com espaço restrito, **GP-UCB** é o processo gaussiano tradicional e **GP-Discont** é a nossa versão utilizando conhecimentos de HPC na modelagem. A linha horizontal pontilhada superior mostra o caso onde é usado todos os nós em todas as iterações, e a inferior a melhor opção durante toda a aplicação. As percentagens internas mostram o ganho comparada com a linha superior (utilizando todos os nós nas iterações).

Os resultados mostram que a estratégia **GP-Discont** é a única que, em todos os casos, atinge os melhores resultados e encontra rapidamente a melhor configuração. O ganho é de até 51%. Isto demonstra a importância de corretamente adequar a aplicação à quantidade de recursos. Trabalhos futuros incluem a utilização da estratégia para outros problemas adaptativos, como o uso de compressão (*low-rank* ou *mixed precision*).

Referências

- Abdulah, S. et al. (2018). Exageostat: A high performance unified software for geostatistics on manycore systems. *IEEE Transactions on Parallel and Distributed Systems*, 29(12):2771–2784.
- Gramacy, R. (2020). *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press.
- HPE, U.-U. H., Laure, E., Narasimhamurthy, S., and Suarez, E. (2022). Heterogeneous high performance computing. *ETP4HPC White Paper*.
- Nesi, L. L., Legrand, A., and Schnorr, L. M. (2021). Exploiting system level heterogeneity to improve the performance of a geostatistics multi-phase task-based application. In *50th International Conference on Parallel Processing, ICPP*, New York, NY, USA. ACM.
- Nesi, L. L., Schnorr, L. M., and Legrand, A. (2022). Multi-Phase Task-Based HPC Applications: Quickly Learning how to Run Fast. In *2022 IEEE Int. Parallel and Dist. Proc. Symp. (IPDPS)*.
- Thibault, S. (2018). *On Runtime Systems for Task-based Programming on Heterogeneous Platforms*. Habilitation à diriger des recherches, Université de Bordeaux.