

Uso de Aprendizado por Reforço para estratégia de amostragem econômicas em ferramenta de análise de escalabilidade

Pedro Rici¹, Samuel Xavier-de-Sousa¹

¹Departamento de Engenharia da Computação e Automação
Universidade Federal do Rio Grande do Norte (UFRN)

pedro.rici.064@ufrn.edu.br, samuel@dca.ufrn.br

Resumo. *A crescente demanda por poder computacional em áreas como simulações científicas e inteligência artificial torna a análise de escalabilidade essencial em ambientes de Computação de Alto Desempenho (HPC). Este trabalho propõe uma estratégia baseada em Aprendizado por Reforço para reduzir o tempo dessa análise, mantendo sua acurácia. A abordagem será aplicada à ferramenta PaScal Suite, com um agente treinado para selecionar amostras mais econômicas.*

1. Introdução

A computação de alto desempenho (HPC, do inglês *High Performance Computing*) vem ganhando cada vez mais destaque frente à crescente demanda por poder computacional. Um dos principais desafios ao se utilizar aplicações em HPC é compreender como cada aplicação responde à ampliação dos recursos computacionais e ao aumento do tamanho do problema [Pacheco 2011], em outras palavras, entender como uma aplicação escala.

A análise de escalabilidade é fundamental para garantir o uso eficiente de recursos computacionais, permitindo identificar gargalos e prever o comportamento da aplicação em arquiteturas maiores. Para isso, já existem ferramentas que automatizam esse processo, como o Scalasca [Geimer et al. 2010] e o PaScal Suite [da Silva et al. 2017]. Entretanto, esse tipo de análise pode ser extremamente custosa e demorada. Isso ocorre porque, para obter dados confiáveis, é necessário realizar simulações repetidas com diferentes configurações, variando, por exemplo, o número de processadores e o tamanho do problema [Ritter et al. 2020].

Dessa forma, torna-se relevante buscar estratégias que otimizem esse processo sem comprometer a precisão dos resultados, reduzindo o tempo necessário para a análise com a mesma eficácia. Na literatura, alguns autores já exploraram soluções nessa direção, como [Ritter et al. 2020], que empregou técnicas como o Aprendizado por Reforço e modelos empíricos para prever comportamentos de escalabilidade com menos experimentos.

2. Proposta de Trabalho

O objetivo deste trabalho é propor uma estratégia para reduzir o tempo de análise de escalabilidade utilizando Aprendizado por Reforço, aplicada à ferramenta *PaScal Suite*. A metodologia utilizada será inspirada no trabalho de [Ritter et al. 2020], que consiste

no desenvolvimento de um agente de Aprendizado por Reforço para determinar quais pontos de amostragem proporcionam uma análise da escalabilidade da aplicação com menor tempo, mas mantendo a acurácia.

Para isso é necessário uma ambiente de avaliação, um agente de Aprendizado por reforço, que será treinado e validado nesse ambiente. Após isso, poderá testar e avaliar o desempenho através de casos de estudos em aplicações escolhidas. A Figura 1 exemplifica a proposta através de um fluxograma.

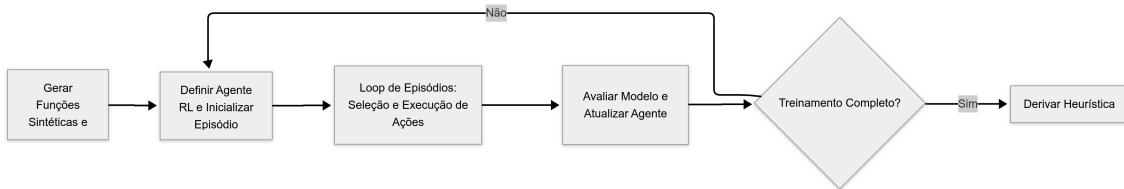


Figure 1. Fluxograma da proposta apresentada

3. Resultados Esperados

Com esta pesquisa, espera-se obter uma metodologia que proporcione a redução do tempo na análise de escalabilidade de aplicações utilizando a ferramenta *PaScal Suite* mantendo a acurácia das suas medições. Além disso, outras contribuições esperadas são: a demonstração da eficácia do agente de Aprendizado por Reforço na seleção de pontos de medição, reduzindo o número total de experimentos necessários; e geração de heurísticas reutilizáveis para futuras análises de escalabilidade, otimizando o processo em diferentes tipos de aplicações. Por fim, deseja-se mostrar que técnicas de inteligência artificial podem ser integradas com sucesso a ferramentas de análise de desempenho.

References

- da Silva, A. B., Cunha, D. A., Silva, V. R., de A. Furtunato, A. F., and Xavier-de Souza, S. (2017). Pascal viewer: A tool for the visualization of parallel scalability trends. In *International Workshop on Extreme-Scale Programming Tools*, pages 250–264. Springer.
- Geimer, M., Wolf, F., Wylie, B. J., Ábrahám, E., Becker, D., and Mohr, B. (2010). The scalasca performance toolset architecture. *Concurrency and computation: Practice and experience*, 22(6):702–719.
- Pacheco, P. (2011). *An introduction to parallel programming*. Elsevier.
- Ritter, M., Calotoiu, A., Rinke, S., Reimann, T., Hoefler, T., and Wolf, F. (2020). Learning cost-effective sampling strategies for empirical performance modeling. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 884–895.