# GERENCIAMENTO DE FILAS E ESCALONAMENTO COM AUXÍLIO DE ML

Lucas Oliveira Macedo, Guilherme Piêgas Koslovski

## INTRODUÇÃO

A execução de aplicações científicas complexas em ambientes de Computação de Alto Desempenho (HPC - *High Performance Computing*) é frequentemente modelada através de fluxos de trabalho, representados por Grafos Acíclicos Direcionados (DAGs). O escalonamento eficiente das tarefas desses *workflows* nos recursos distribuídos é um problema de otimização NP-difícil. Tradicionalmente, utilizam-se heurísticas de escalonamento em tempo de compilação, como *Heterogeneous Earliest Finish Time* (HEFT), *Critical-Path-on-a-Processor* (CPOP) e Min-Min (TOPCUOGLU; HARIRI; WU, 2002), para alocar as tarefas buscando minimizar o tempo total de execução (*Makespan*). Contudo, a eficácia de cada algoritmo varia a depender das características estruturais do grafo, do volume de dados trafegados e do estado dinâmico da infraestrutura, como o congestionamento da rede.

Diante da ausência de uma heurística universalmente superior para todos os cenários, torna-se necessário avaliar múltiplos objetivos, como *Bounded Slowdown* e o custo de alocação, para definir o algoritmo ideal. No entanto, simular cada cenário consome tempo e recursos. Assim, o problema central reside na dificuldade de selecionar dinamicamente a melhor política de escalonamento sem recorrer a simulações prévias.

Neste contexto, o objetivo deste trabalho é desenvolver um modelo preditivo baseado em Redes Neurais em Grafos (GNN - *Graph Neural Networks*) capaz de atuar como um meta-escalonador. A partir de uma análise da topologia do *workflow* e das exigências de rede e CPU, a GNN visa prever a heurística de escalonamento vencedora sob uma perspectiva multiobjetivo. Adicionalmente, busca-se integrar os ambientes WfCommons (FERREIRA DA SILVA et al., 2022) e WRENCH (CASANOVA et al., 2018) para garantir a geração de um *dataset* confiável e reprodutível, consolidando o treinamento e a avaliação do modelo.

## DESENVOLVIMENTO

A metodologia proposta foi estruturada em duas etapas principais: a geração da base de dados por meio de simulação de escalonamento e a modelagem do meta-escalonador preditivo via Redes Neurais em Grafos. Na primeira etapa, instâncias de *workflows* científicos representados como DAGs foram processadas para extração de atributos das tarefas (*runtime*) e custos de comunicação nas dependências (*data cost*). Para cada grafo, foi simulada uma execução sob seis heurísticas tradicionais de escalonamento estático: HEFT, *Earliest Finish Time* (EFT), Min-Min, Max-Min, CPOP e *Predict Earliest Finish Time* (PEFT). O desempenho de cada algoritmo foi avaliado a partir de quatro métricas: *Makespan* (normalizado pelo caminho crítico), *Bounded Slowdown*, *Footprint* de recursos e balanceamento de carga. Uma função multiobjetivo com pesos equalitários (25% por métrica) foi aplicada para eleger o escalonador vencedor de cada cenário sob diferentes taxas de degradação da rede, compondo o gabarito supervisionado do *dataset*.

Na segunda etapa, os atributos contínuos foram normalizados por meio de padronização estatística (*StandardScaler*) para estabilização numérica dos gradientes. A arquitetura da IA foi implementada em Python com a biblioteca *PyTorch Geometric* (FEY; LENSSEN, 2019), utilizando uma Rede Convolucional em Grafos Residual (ResGCN) (LI et al., 2019). A estrutura do grafo foi mapeada

através de tensores de adjacência (*edge_index*), viabilizando operações de propagação e agregação de mensagens (*Message Passing*) ponderadas pelo volume de dados trafegado. Conexões residuais (*skip connections*) foram incorporadas entre as camadas convolucionais para mitigar o problema do gradiente evanescente. As representações latentes dos vértices foram sintetizadas por camadas de *pooling* global e concatenadas a variáveis globais do sistema antes da classificação final. O treinamento utilizou a função de perda *Cross-Entropy Loss* combinada a pesos de classe balanceados (*balanced class weights*), assegurando a correta penalização do modelo diante do desbalanceamento amostral dos algoritmos.

**RESULTADOS**

Os experimentos demonstraram a viabilidade da utilização da rede convolucional residual (ResGCN) como meta-escalonador preditivo. Durante a fase de treinamento, a GNN processou com êxito os tensores de topologia e os atributos de cada vértice. Observou-se no *dataset* um forte desbalanceamento empírico, no qual heurísticas focadas em alto paralelismo (como o Min-Min) venciam a avaliação multiobjetivo com frequência expressivamente superior a algoritmos de nicho (como o CPOP). Para mitigar esse viés, foi aplicado um balanceamento dinâmico de pesos na função *Cross-Entropy Loss*. Essa abordagem técnica permitiu penalizar o modelo severamente ao classificar incorretamente as classes minoritárias, forçando a convergência equilibrada do gradiente.

O processo de aprendizado resultou em uma curva de otimização estável, validando a capacidade da arquitetura em agregar as características dos nós vizinhos (*Message Passing*) para compreender o caminho crítico e os gargalos de comunicação do grafo. A eficácia preditiva da rede consolida a relevância do meta-escalonador na otimização de recursos de HPC, conforme a evolução das épocas de treinamento ilustrada no gráfico da Figura 1 (Anexos). Como principal limitação da atual etapa, o simulador utilizado para construir a base de dados operou de forma determinística em *scripts* isolados, o que motivou a etapa seguinte da pesquisa: a padronização do ambiente físico virtual através da integração com o *framework* WRENCH, visando elevar a fidelidade física das próximas avaliações.

**CONSIDERAÇÕES FINAIS**

A pesquisa evidenciou que a utilização de Redes Neurais em Grafos é uma abordagem viável e altamente promissora para atuar como um meta-escalonador inteligente em ambientes de Computação de Alto Desempenho. A arquitetura demonstrou capacidade de extrair e aprender características estruturais complexas dos fluxos de trabalho científicos, conseguindo prever heurísticas de escalonamento adequadas sob uma perspectiva multiobjetivo. A aplicação de técnicas de balanceamento de pesos se provou essencial para a estabilidade do treinamento, impedindo a rede de favorecer tendenciosamente algoritmos majoritários. Como desdobramento natural e trabalho futuro, a migração do motor de simulação analítico para o *framework* WRENCH, aliado à padronização das instâncias de grafos pelo WfCommons, elevará a pesquisa aos padrões globais da comunidade científica, garantindo a escalabilidade, a precisão física da simulação de rede e a total reprodutibilidade do *dataset* gerado.

**Palavras-chave:** redes neurais em grafos; escalonamento multiobjetivo; aprendizado de máquina; fluxos de trabalho científicos; WRENCH; PyTorch Geometric.

**ANEXOS**

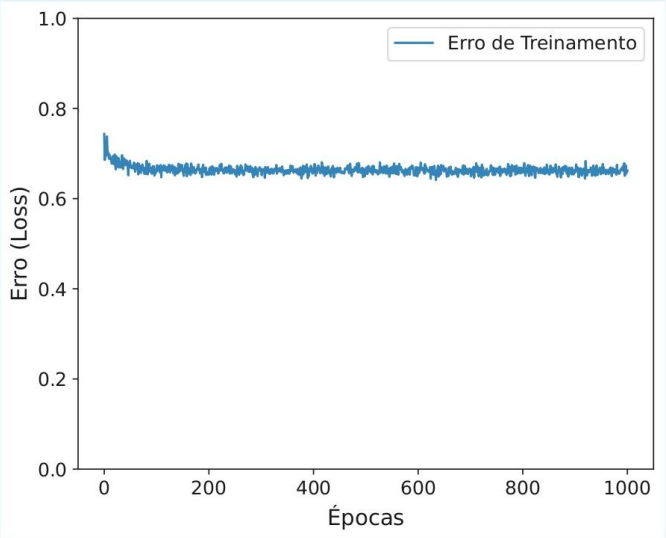



**Figura 1.** *Curva de convergência da função de perda (Cross-Entropy Loss) durante as épocas de treinamento da ResGCN.*

**REFERÊNCIAS BIBLIOGRÁFICAS**

CASANOVA, H. et al. WRENCH: A framework for simulating workflow management systems. Em: **2018 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)**. IEEE, 2018. p. 74-85.

FERREIRA DA SILVA, R. et al. WfCommons: A framework for enabling scientific workflow research and development. **Future Generation Computer Systems**, v. 128, p. 16-27, 2022.

FEY, M.; LENSSEN, J. E. Fast graph representation learning with PyTorch Geometric. Em: **ICLR Workshop on Representation Learning on Graphs and Manifolds**, 2019.

TOPCUOGLU, H.; HARIRI, S.; WU, M. Performance-effective and low-complexity task scheduling for heterogeneous computing. **IEEE Transactions on Parallel and Distributed Systems**, v. 13, n. 3, p. 260-274, 2002.

LI, G. et al. DeepGCNs: Can GCNs go as deep as CNNs? Em: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. 2019. p. 9267-9276.