

## COMPARAÇÃO E MELHORIA DE POLÍTICAS DE ESCALONAMENTO GUIADAS POR ML

Claudinei Cabral Júnior, Guilherme Piêgas Koslovski

### INTRODUÇÃO

O escalonamento de tarefas em infraestruturas de Computação de Alto Desempenho representa um desafio crítico que impacta diretamente a experiência dos usuários e a eficiência dos recursos computacionais. Embora técnicas de Aprendizado de Máquina tenham demonstrado potencial significativo para otimizar políticas de escalonamento (Yen, 2023; Abbasl, 2020; Diel, 2025), administradores de tais infraestruturas frequentemente optam por heurísticas simples devido à falta de explicabilidade e confiança nos indicadores de desempenho de longo prazo das políticas baseadas em RL (Santos, 2019). O objetivo deste trabalho foi realizar uma análise comparativa aprofundada entre dois escalonadores baseados em aprendizado de máquina: *Actor-Critic Reinforcement Learning* (ACRL) (Koslovski, 2024) e Decima (Mao, 2019), fornecendo indicadores quantitativos e qualitativos essenciais para auxiliar administradores na tomada de decisões informadas sobre a adoção dessas tecnologias.

### DESENVOLVIMENTO

A metodologia consistiu em uma análise comparativa estruturada utilizando um protocolo de simulação padronizado para garantir uma avaliação justa entre os modelos. Foram definidos dois cenários experimentais baseados em grafos acíclicos direcionados (DAG, do inglês *Directed Acyclic Graph*): (i) DAGs com tarefas agrupadas sequencialmente e (ii) DAGs baseados em tarefas com dependências auxiliares. Os experimentos utilizaram cargas de trabalho representativas do benchmark TPC-H (TPCH-H BENCHMARK, 2025), executando sobre uma infraestrutura com 50 servidores. As métricas quantitativas incluíram *footprint* (percentual de servidores em uso), *bounded slowdown* (tempo acumulado de tarefas em fila), número de trabalhos e tarefas em fila e número de trabalhos e tarefas escalonados. A análise qualitativa abrangeu tempo de resposta, complexidade de parametrização, tempo de treinamento e função de recompensa e objetivo. Ambos os modelos foram treinados com configurações padronizadas, sendo o Decima testado com 100, 1000 e 10000 episódios, enquanto o ACRL utilizou 1000 episódios.

### RESULTADOS

Os resultados revelaram comportamentos distintos entre os escalonadores conforme a estrutura dos DAGs. No cenário de tarefas agrupadas, o Decima manteve filas de trabalhos reduzidas em suas três configurações: 99% com fila  $\leq 1$  na configuração de 100 episódios, 77% com fila  $\leq 1$  na configuração de 1000 episódios e por fim, 70% com fila  $\leq 1$  na configuração de 10000 episódios. Apesar disso, o modelo apresentou fila de tarefas mais longas (até 530 tarefas) em todas suas configurações. Já o modelo ACRL demonstrou filas de tarefas menores (63% abaixo de 61 tarefas) mas fila de trabalhos com um crescimento linear devido às limitações de sua janela temporal. Em termos de uso de recursos, o Decima alcançou utilização consistentemente alta ( $> 94\%$ ), enquanto o ACRL manteve utilização mais baixa (85% dos valores  $< 60\%$ ). No cenário de tarefas auxiliares, o ACRL demonstrou uma fila de trabalho constante em 1 e um rápido escalonamento de tarefas com uma utilização de recursos mais distribuída (57% dos valores  $< 50\%$ ). As Figuras 1 e 2 apresentam o *bounded slowdown* (*bsld*), métrica importante para experiência dos usuários (Feitelson, 1998), em ambos os cenários. Na Figura 1, o modelo

ACRL apresenta uma média de *bsld* baixa em grande parte de sua distribuição, com aproximadamente 40% dos trabalhos experienciando um *bsld* de 1650. Na Figura 2, o ACRL exibiu um comportamento em dois estágios, com uma transição abrupta de 1687 para 2348, enquanto Decima mostrou crescimento linear e gradual. Qualitativamente, o ACRL requer tempo de treinamento significativamente maior (18h vs 5m-70h do Decima) e apresenta maior tempo de resposta (0.195s vs 0.018s), além de maior complexidade de parametrização para janelas temporais.

### CONSIDERAÇÕES FINAIS

A análise demonstrou que não existe uma solução única ideal para todos os cenários, confirmando a ausência de uma política "bala de prata". Os resultados evidenciam que a escolha entre escalonadores baseados em RL deve considerar não apenas métricas de desempenho, mas também fatores qualitativos como tempo de treinamento, complexidade de configuração e explicabilidade das decisões. A metodologia desenvolvida pode ser aplicada para avaliar outros escalonadores baseados em RL, fornecendo base sólida para decisões administrativas em ambientes HPC reais. Como trabalho futuro, cita-se aplicação de técnicas de otimização

**Palavras-chave:** escalonamento; aprendizado de máquina; computação de alto desempenho; ACRL; Decima;

### ILUSTRAÇÕES

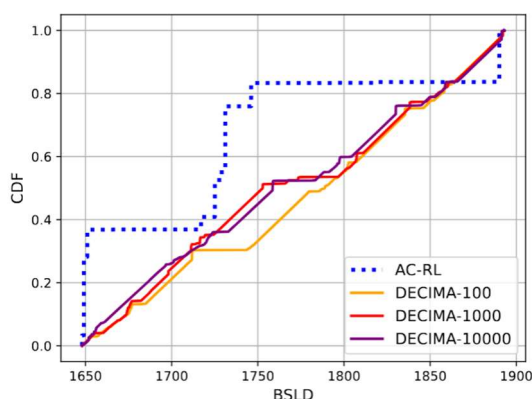


Figura 1. Bounded Slowdown em DAG com tarefas agrupadas

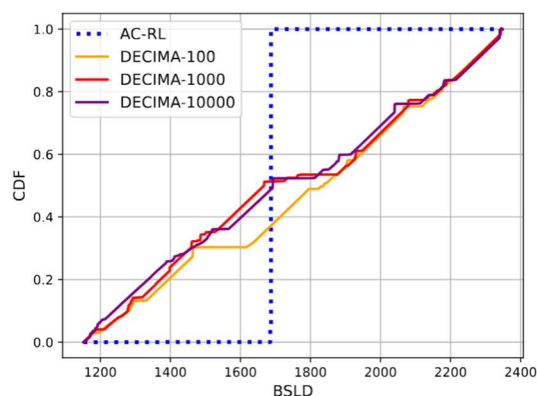


Figura 2. Bounded Slowdown em DAGs com tarefas auxiliares

**REFERÊNCIAS BIBLIOGRÁFICAS**

ABBASLOO, S.; YEN, C.-Y.; CHAO, H. J. Classic meets modern: a pragmatic learning-based congestion control for the internet. In: PROCEEDINGS OF THE ANNUAL CONFERENCE OF THE ACM SPECIAL INTEREST GROUP ON DATA COMMUNICATION ON THE APPLICATIONS, TECHNOLOGIES, ARCHITECTURES, AND PROTOCOLS FOR COMPUTER COMMUNICATION, 2020, New York. Anais... New York: ACM, 2020. p. 632-647. (SIGCOMM '20). Disponível em: <https://doi.org/10.1145/3387514.3405892>. Acesso em: 20 ago. 2025.

CARASTAN-SANTOS, D. et al. One can only gain by replacing easy backfilling: a simple scheduling policies case study. In: IEEE/ACM INTERNATIONAL SYMPOSIUM ON CLUSTER, CLOUD AND GRID COMPUTING, 19., 2019, [s.l.]. Proceedings... [S.l.]: IEEE, 2019. p. 1-10. Acesso em: 20 ago. 2025.

DIEL, G.; KRAUS, A. E. N.; KOSLOVSKI, G. P. Knowledge-based job scheduling for HPC. Cluster Computing, [s.l.], v. 28, n. 3, p. 153, 2025. Acesso em: 20 ago. 2025.

FEITELSON, D. G.; RUDOLPH, L. Metrics and benchmarking for parallel job scheduling. In: FEITELSON, D. G.; RUDOLPH, L. (Ed.). *Job Scheduling Strategies for Parallel Processing*. Berlin: Springer Berlin Heidelberg, 1998. p. 1–24. Disponível em: <https://doi.org/10.1007/BFb0053978>. Acesso em: 20 ago. 2025.

HONGZI, M. et al. Learning scheduling algorithms for data processing clusters. In: PROCEEDINGS OF THE ACM SPECIAL INTEREST GROUP ON DATA COMMUNICATION, 2019, [s.l.]. Anais... [S.l.]: ACM, 2019. Acesso em: 20 ago. 2025.

KOSLOVSKI, G. P. et al. Actor-critic reinforcement learning for resource management and task scheduling in cloud computing environments. Journal of Parallel and Distributed Computing, [s.l.], v. 124, p. 67-82, 2024. Acesso em: 20 ago. 2025.

TPC-H BENCHMARK. [S.l.]: [2025]. Disponível em: <https://www.tpc.org/tpch/>. Acesso em: 20 ago. 2025.

YEN, C.-Y.; ABBASLOO, S.; CHAO, H. J. Computers can learn from the heuristic designs and master internet congestion control. In: PROCEEDINGS OF THE ACM SIGCOMM CONFERENCE, 2023, New York. Anais... New York: ACM, 2023. p. 255-274. (ACM SIGCOMM '23). Disponível em: <https://doi.org/10.1145/3603269.3604838>. Acesso em: 20 ago. 2025. Acesso em: 20 ago. 2025.

**BOLSISTA:** Claudinei Cabral Júnior

**MODALIDADE DE BOLSA:** PROBIC/UDESC

**VIGÊNCIA:** 08/2024 a 08/2025– Total: 12 meses

**ORIENTADOR(A):** Guilherme Piêgas Koslovski

**CENTRO DE ENSINO:** CCT

**DEPARTAMENTO:** Departamento de Ciência da Computação

**ÁREAS DE CONHECIMENTO:** Ciências Exatas e da Terra / Ciência da Computação

**TÍTULO DO PROJETO DE PESQUISA:** Mecanismos para Alocação de Infraestruturas Virtuais baseados em Aprendizado de Máquina e Acelerados por GPUs Parte 3

**Nº PROTOCOLO DO PROJETO DE PESQUISA:** NPP3275-2023