Auto-Adaptive Multi-Objective Scheduling for Academic HPC Grids: Simulation and Execution with OAR and SimGrid

Xavier P. Sebastião[1,2], Gerson Geraldo H. Cavalheiro[1]

[1] Graduate Program in Computing
Technological Development Center – Federal University of Pelotas (UFPel)
Anglo Campus - Gomes Carneiro Street, 01 – Downtown – Pelotas, RS – Brazil

[2]Zambeze University (UniZambeze) – Ulóngué-Angónia – Mozambique

{xpsebastiao, gerson.cavalheiro}@inf.ufpel.edu.br

Abstract. *This paper proposes an auto-adaptive scheduling framework integrating OAR, a production-grade scheduler deployed in Grid'5000, with SimGrid-based simulation to enable continuous learning and multi-objective optimization. Unlike existing approaches that treat prediction, simulation, and scheduling separately, this framework establishes a closed feedback loop where runtime data from simulated Grid'5000 environments informs policy evaluation, and validated policies are designed for deployment back into OAR via its plugin architecture. The proposed framework intents to balance makespan, energy consumption, and fairness while supporting containerized scientific workflows through simulation that models operational conditions.*

1. Introduction

High-performance computing (HPC) infrastructures are essential for large-scale scientific simulations and data-intensive experiments in academic research [Shan et al. 2023]. These workloads typically execute on shared infrastructures organized as institutional HPC grids, which aggregate heterogeneous clusters under multi-user policies [Hilman et al. 2020]. Grid'5000 exemplifies such infrastructure, which provides approximately 30,918 cores, 935 compute nodes, and 760 GPUs across 11 sites in Europe for experiment-driven research [Grid'5000 2026]. This type of environments faces significant workload variability, resource heterogeneity, and multi-stage workflow dependencies that makes efficient scheduling a persistent challenge [Bader et al. 2024].

Academic HPC grids must balance competing objectives such as minimizing job turnaround time, maximizing resource utilization, ensuring fairness across users, and reducing energy consumption [Fard et al. 2014]. Traditional schedulers including SLURM, PBS, and OAR, provide robust queue management and prioritization but rely on predefined static policies with limited adaptivity. OAR, widely adopted in Grid'5000, offers advanced features like conservative backfilling and advance reservations, yet lacks native support for learning-based or self-adaptive scheduling mechanisms [Bolze et al. 2006].

Simulation frameworks enable reproducible evaluation of scheduling strategies. SimGrid has become a reference platform for modeling large-scale distributed systems under controlled conditions [Casanova et al. 2025], with Batsim allowing batch scheduler research, processing job traces and producing detailed performance metrics including waiting times, response times, utilization, and energy consumption [Casanova et al. 2025]. Studies in HPC demonstrate SimGrid-based simulations with significant calibration accuracy improvements using production workloads [Horzela et al. 2024]. However, integration between production schedulers like OAR and simulators like SimGrid remains limited. This prevents the transfer of simulation insights into operational policies.

Multi-criteria optimization in HPC has evolved from single-objective formulations to approaches balancing competing goals [Fard et al. 2014]. Research works demonstrate that RL-based schedulers are effective and visualize the potential of deep reinforcement learning (DRL) for adaptive scheduling. Hierarchical approaches that combine graph neural networks show 12.54% makespan reduction and 5.83% load balancing improvements in simulated heterogeneous environments [Zhang et al. 2024]. Transformer-enhanced deep Q-networks achieve 13.66% makespan reduction, 16.65% energy improvement, and 44.72% cost reduction in dynamic workflow scheduling [Ding et al. 2024].

Multi-agent reinforcement learning approaches show improved stability and generalizability for scheduling problems [Pu et al. 2024]. Self-attention mechanisms with gated transformers demonstrate significant performance improvements under high-load conditions [Gao et al. 2025]. Yet these approaches remain evaluated in isolated simulation without connection to production schedulers.

This fragmentation represents a critical gap. Integrated frameworks that continuously refine scheduling decisions based on simulated feedback modeling real operational conditions remain limited. OAR's role in Grid'5000 presents an opportunity to develop adaptive scheduling through simulation that can accurately recreate production behavior before any deployment. This paper proposes an auto-adaptive framework bridging OAR and SimGrid through closed-loop architecture where all components are developed and validated within a unified simulation environment to model Grid'5000 platform.

2. Proposed Solution Architecture

The framework establishes continuous feedback in a unified simulation environment that models a real Grid'5000. Four integrated components operate in SimGrid, to allow comprehensive validation under controlled but realistic conditions. See Figure 1 for the detailed proposed solution architecture.

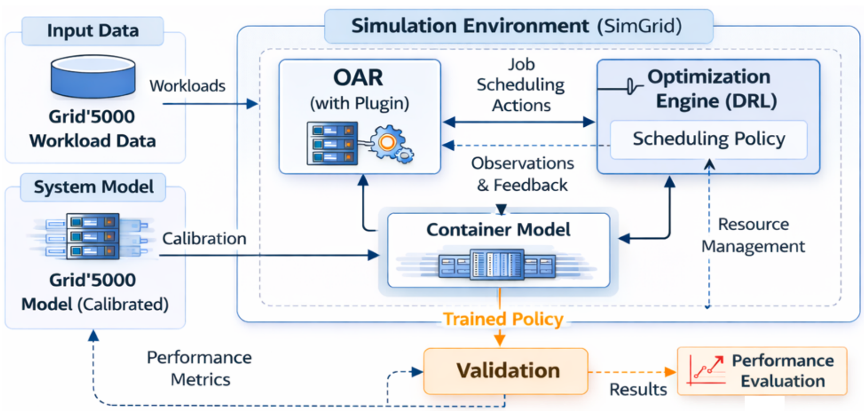

Figure 1. High-level OAR–SimGrid Integrated Scheduling Architecture

The OAR monitoring layer extends instrumentation to capture job characteristics (submission time, requested resources, runtime), resource utilization metrics (CPU, memory, I/O), energy consumption, and scheduling decisions on simulated Grid'5000 clusters using real workload traces [Bolze et al. 2006]. The plugin exercises the extension role for interfacing with the optimization engine to receive updated parameters and with monitoring section to observe effects and provide real-time state for adaptive decisions.

The SimGrid simulation environment, with Batsim as an extension, models academic grid infrastructure with validated accuracy [Casanova et al. 2025], incorporating network topology, node heterogeneity, job dependencies, and energy patterns calibrated from real Grid'5000 measurements [Horzela et al. 2024]. The optimization engine employs deep reinforcement learning agents that continuously refine decisions based on simulated real-time state, with state including queue characteristics, resource availability, recent performance, and energy patterns; actions corresponding to scheduling decisions; rewards combining objectives. Following multi-agent approaches [Pu et al. 2024], the framework enables distributed learning across multiple scheduler instances.

The framework models containerized jobs using Docker or Singularity technologies to capture image size and transfer costs, resource requirements from manifests, isolation overhead, co-location constraints, and image caching [Shan et al. 2023]. Scheduling is formulated as multi-objective optimization with three primary objectives [Fard et al. 2014], balancing performance, energy consumption, and fairness subject to resource capacities, job dependencies, policy limits, and data locality. Fairness is controlled by reputation tracking based on historical completion rates, and policy compliance influences scheduling priority.

To incentivize cooperative behavior in multi-institutional grids, we incorporate reputation tracking based on historical job completion rates, policy compliance, contributions to grid maintenance, and workload reliability. Reputation scores influence scheduling priority and resource access within the simulated environment.

3. Evaluation and Expected Results

Training, evaluation, and validation will use Grid'5000 workload traces during simulation [Bolze et al. 2006]. In addition, the framework will compare with FIFO, EASY backfilling, and recent RL-based schedulers [Zhang et al. 2024, Ding et al. 2024, Gao et al. 2025] implemented inside SimGrid. Metrics will include waiting time, resource utilization, energy, and fairness.

Expected contributions include a validated Grid'5000 simulation model that approximates real operational behavior, to be released as open source; a complete implementation of the framework with OAR and SimGrid that validates closed-loop adaptation; quantitative optimization and analysis of performance-energy-fairness tradeoffs under realistic conditions; and an open-source simulation environment that enables reproducible research. In future, the framework may be extended to support other tools and metrics.

4. Conclusion

This paper proposes an auto-adaptive framework for integrating OAR and SimGrid into a simulation that reflects real operational behavior. By having continuous feedback that links monitoring, simulation, optimization, and deployment interfaces within a unified simulation environment, the framework is expected to achieve a comprehensive validation of adaptive scheduling under realistic conditions while reducing production risks (such as hardware damage, system downtime, data loss, and energy inefficiency). This approach addresses the fragmentation between prediction, simulation, and execution, and creates a pathway to transition from learning-based scheduling to future production deployments.

References

Bader, J., Lehmann, F., Thamsen, L., Leser, U., and Kao, O. (2024). Lotaru: Locally predicting workflow task runtimes for resource management on heterogeneous infrastructures. *Future Generation Computer Systems*, 150:171–185.

Bolze, R., Cappello, F., Caron, E., Daydé, M., Desprez, F., Jeannot, E., Jégou, Y., Lanteri, S., Leduc, J., Melab, N., et al. (2006). Grid'5000: a large scale and highly reconfigurable experimental grid testbed. *The International Journal of High Performance Computing Applications*, 20(4):481–494.

Casanova, H., Giersch, A., Legrand, A., Quinson, M., and Suter, F. (2025). Lowering entry barriers to developing custom simulators of distributed applications and platforms with simgrid. *Parallel Computing*, 123:103125.

Ding, F., Yuan, Y., Lv, L., Zhang, R., and Zhou, W. (2024). Transformer-enhanced dqn approach for energy and cost-efficient large-scale dynamic workflow scheduling in heterogeneous environment. *IEEE Internet of Things Journal*, 11(22):37351–37367.

Fard, H. M., Prodan, R., and Fahringer, T. (2014). Multi-objective list scheduling of workflow applications in distributed computing infrastructures. *Journal of Parallel and Distributed Computing*, 74(3):2152–2165.

Gao, X., Dong, H., Zhang, L., Wang, Y., Yang, X., and Li, Z. (2025). Self-attention mechanisms in hpc job scheduling: A novel framework combining gated transformers and enhanced ppo. *Applied Sciences*, 15(16):8928.

Grid'5000 (2026). Grid'5000 - hardware overview. `https://www.grid5000.fr/w/Hardware`. Accessed: March 4, 2026.

Hilman, M. H., Rodriguez, M. A., and Buyya, R. (2020). Multiple workflows scheduling in multi-tenant distributed systems: A taxonomy and future directions. *ACM Computing Surveys (CSUR)*, 53(1):1–39.

Horzela, M., Casanova, H., Giffels, M., Gottmann, A., Hofsaess, R., Quast, G., Tisbeni, S. R., Streit, A., and Suter, F. (2024). Modeling distributed computing infrastructures for hep applications. In *EPJ Web of Conferences*, volume 295, page 04032. EDP Sciences.

Pu, Y., Li, F., and Rahimifard, S. (2024). Multi-agent reinforcement learning for job shop scheduling in dynamic environments. *Sustainability*, 16(8):3234.

Shan, C., Wu, C., Xia, Y., Guo, Z., Liu, D., and Zhang, J. (2023). Adaptive resource allocation for workflow containerization on kubernetes. *Journal of Systems Engineering and Electronics*, 34(3):723–743.

Zhang, Z., Xu, C., Liu, K., Xu, S., and Huang, L. (2024). A resource optimization scheduling model and algorithm for heterogeneous computing clusters based on gnn and rl: Z. zhang et al. *The Journal of Supercomputing*, 80(16):24138–24172.