ParslBox: Computational Materials Workflow Executor for AI-Agents on High-Performance Computing Systems

1st Fakhrul Hasan Bhuiyan
Computational Science Division
Argonne National Laboratory
Lemont, IL 60439, USA
Email: fbhuiyan@anl.gov

2nd Álvaro Vázquez-Mayagoitia
Computational Science Division
Argonne National Laboratory
Lemont, IL 60439, USA
Email: vama@alcf.anl.gov

***Abstract*—Artificial intelligence (AI) agents can autonomously construct large computational materials workflows, but executing these adaptive, fine-grained workloads efficiently across diverse high-performance computing (HPC) systems remains challenging. We present ParslBox (PBX), a lightweight workflow execution toolkit for computational materials applications on HPC systems. PBX provides a local job database, fine-grained CPU/GPU resource management, portable execution across heterogeneous HPC systems, and command-line, Python, and Model Context Protocol interfaces for both users and AI agents. Scalability was demonstrated on up to 2,000 nodes and 24,000 concurrent workers while maintaining low orchestration overhead. Strong-scaling studies using LAMMPS with MACE models and VASP demonstrated transparent execution of production scientific applications. Agentic benchmarks showed reliable workflow construction across multiple closed-weight and open-weight language models. End-to-end agent-driven generation of machine-learned force fields for molten LiCl-NaCl and LiCl-KCl systems further demonstrated autonomous orchestration of complex multi-stage HPC workflows.**

***Index Terms*—Computational materials, Workflow, AI agents, High-performance computing, MLIP.**

I. Poster Summary

Artificial intelligence (AI) agents are beginning to transform computational science by converting high-level scientific goals into executable workflows. In computational materials research, such workflows can include thousands of interdependent molecular dynamics (MD) simulations, density functional theory (DFT) calculations, and analysis tasks [1], [2]. Recent agentic workflow systems have shown strong promise for autonomous workflow construction [3]–[5]. However, efficient execution of agent-generated workflows on high-performance computing (HPC) systems remains difficult. HPC sites differ in schedulers, queue policies, software environments, hardware architectures, and resource allocation mechanisms. Large agent-generated campaigns therefore require an execution layer that is portable across sites, aware of sub-node resources, and simple to deploy on production supercomputers.

ParslBox, built on top of Parsl, is a lightweight workflow management and execution toolkit for computational materials workflows on HPC systems [6]. The toolkit combines a local SQLite job database, fine-grained CPU and GPU resource management, portable Message Passing Interface (MPI) launch generation, application plugins, and ready-to-use system configurations. Job state is stored in a local database rather than an external service, so ParslBox does not require a user-managed database daemon or a hosted control plane. Jobs can be added, queried, filtered, updated, and executed through a command-line interface, a Python application programming interface (API), or a Model Context Protocol (MCP) server. The aligned interfaces allow human users and AI agents to interact with the same execution engine.

ParslBox executes heterogeneous workloads within large HPC allocations. Each job contains a resource specification that includes nodes, GPUs, CPU occupancy, and MPI ranks per node. The resource manager classifies jobs as sub-node CPU, sub-node GPU, full-node CPU, or full-node GPU tasks. Sub-node GPU jobs receive explicit GPU identifiers and associated CPU cores, with CPU-GPU affinity maps used when available. Sub-node CPU jobs receive core slices sized to the requested occupancy. Multi-node MPI jobs acquire exclusive nodes to avoid interference. Jobs that cannot be launched immediately are either held in a backlog (static mode) or released back to the job database (dynamic mode) and re-evaluated as resources become available. Portable launch commands are generated for site-specific launchers, including mpirun, mpiexec, and srun.

Scalability was evaluated on the Aurora supercomputer at the Argonne Leadership Computing Facility (ALCF). Stress tests used up to 24,000 concurrent GPU-tile jobs across 2,000 nodes. The test workloads contained an equal mix of LAMMPS, VASP, Python, and Julia placeholder jobs so that orchestration overhead could be isolated from application runtime. ParslBox maintained a nearly constant head-node process count and a sub-gigabyte orchestrator memory footprint. At the largest tested scale, a single `pbx run` process managed 24,000 concurrently tracked GPU jobs with only about 410 megabytes of resident memory. The results showed that large campaigns could be executed with low and predictable overhead, without auxiliary database or control services.

Production application benchmarks were also performed on

Aurora. Strong-scaling studies used LAMMPS with MACE models for MD simulations and VASP for spin-polarized DFT calculations [2], [7]–[11]. LAMMPS simulations of 3C-SiC systems containing 500 to 100,000 atoms scaled to thousands of GPU tiles. VASP calculations also scaled across increasing GPU counts before application-level diminishing returns were reached. Across both workloads, ParslBox configured resource allocation, CPU-GPU affinity, MPI launch, and job submission without application-specific changes. The measured scaling behavior was governed by the underlying scientific applications rather than by the workflow executor.

Agentic workflow construction was evaluated with closed-weight and open-weight language models. Each model received a single prompt and was asked to register workflows containing 10 to 20,000 jobs through either the MCP interface or a command-line skill. Closed-weight models achieved 100 percent success across all tested scales with the MCP interface. The command-line skill produced near-perfect success across all models up to 10,000 jobs. The benchmarks showed that ParslBox can support reliable workflow construction by AI agents with different levels of tool-use capability.

End-to-end agent-driven execution was demonstrated for machine-learned interatomic potential (MLIP) development for molten salt systems. Coding agents, including Claude Code, OpenCode, and ChemGraph, were provided with a scientific workflow skill and the ParslBox interface [3], [12]. From a single user instruction, the agents generated complete directed acyclic workflow graphs for fine-tuning MACE models for LiCl-NaCl and LiCl-KCl molten salts. The workflows included sampling MD simulations, farthest-point sampling, DFT labeling of energies, forces, and stresses, dataset assembly, MLIP training, validation simulations, and post-processing [13]. The generated workflows were executed through ParslBox on ALCF Aurora. Radial distribution functions from the resulting models agreed with prior computational and experimental results [14], [15], demonstrating successful autonomous workflow construction and execution.

ParslBox provides a practical execution substrate for traditional computational materials campaigns and emerging AI-driven scientific computing. Lightweight deployment, persistent job tracking, fine-grained resource management, portable MPI execution, scalable performance, and agent-friendly interfaces allow ParslBox to lower the barrier to autonomous and reproducible HPC workflow execution on leadership-class systems.

Data and Code Availability

The ParslBox source code is publicly available at https://github.com/fbhuiyan2/parslbox. The training datasets, provided as atomic structures in XYZ files, and the fine-tuned models for the LiCl-NaCl and LiCl-KCl binary molten-salt systems are available through Zenodo at https://doi.org/10.5281/zenodo.22848611.

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Science Energy Earthshot Initiative as part of the Center for Steel Electrification by Electrosynthesis (C-STEEL) at Argonne National Laboratory under Contract Number DE-AC02-06CH11357. We acknowledge the computing resources provided by the "Improv" and "Swing" computing clusters operated by the Laboratory Computing Resource Center at Argonne National Laboratory. This research used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357. A.V.M. and F.H.B. were supported by the Office of Science, U.S. Department of Energy, under contract DE-AC02-06CH11357.

References

[1] E. Deelman, T. Peterka, I. Altintas, C. D. Carothers, K. K. Van Dam, K. Moreland, M. Parashar, L. Ramakrishnan, M. Taufer, and J. Vetter, "The future of scientific workflows," *The International Journal of High Performance Computing Applications*, vol. 32, no. 1, pp. 159–175, 2018.

[2] I. Batatia, P. Benner, Y. Chiang, A. M. Elena, D. P. Kovács, J. Riebesell, X. R. Advincula, M. Asta, M. Avaylon, W. J. Baldwin, F. Berger, N. Bernstein, A. Bhowmik, F. Bigi, S. M. Blau, V. Cărare, M. Ceriotti, S. Chong, J. P. Darby, S. De, F. Della, V. L. Deringer, R. Elijošius, Z. El-Machachi, E. Fako, F. Falcioni, A. C. Ferrari, J. L. A. Gardner, M. J. Gawkowski, A. Genreith-Schriever, J. George, R. E. A. Goodall, J. Grandel, C. P. Grey, P. Grigorev, S. Han, W. Handley, H. H. Heenen *et al.*, "A foundation model for atomistic materials chemistry," *The Journal of Chemical Physics*, vol. 163, no. 18, p. 184110, Nov 2025. [Online]. Available: https://pubs.aip.org/aip/jcp/article/163/18/184110/3372267/A-foundation-model-for-atomistic-materials

[3] T. D. Pham, A. Tanikanti, and M. Keçeli, "ChemGraph as an agentic framework for computational chemistry workflows," *Commun Chem*, vol. 9, no. 1, p. 33, Jan 2026.

[4] Q. Campbell, S. Cox, J. Medina, B. Watterson, and A. D. White, "MDCrow: automating molecular dynamics workflows with large language models," *Mach. Learn.: Sci. Technol.*, vol. 7, no. 2, p. 025037, Mar 2026.

[5] G. Liu, S. Yang, and Y. Zhong, "Masgent: an AI-assisted materials simulation agent," *Digital Discovery*, vol. 5, no. 5, pp. 2151–2171, May 2026.

[6] Y. Babuji, A. Woodard, Z. Li, D. S. Katz, B. Clifford, R. Kumar, L. Lacinski, R. Chard, J. M. Wozniak, I. Foster *et al.*, "Parsl: Pervasive parallel programming in python," in *Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing*, 2019, pp. 25–36.

[7] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Comp. Phys. Comm.*, vol. 271, p. 108171, 2022.

[8] D. P. Kovács, J. H. Moore, N. J. Browning, I. Batatia, J. T. Horton, Y. Pu, V. Kapil, W. C. Witt, I.-B. Magdau, D. J. Cole *et al.*, "Mace-off: Short-range transferable machine learning force fields for organic molecules," *Journal of the American Chemical Society*, vol. 147, no. 21, pp. 17 598–17 611, 2025.

[9] G. Kresse and J. Furthmüller, "Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set," *Computational materials science*, vol. 6, no. 1, pp. 15–50, 1996.

[10] ——, "Software vasp, vienna (1999)," *Phys. Rev. B*, vol. 54, no. 11, p. 169, 1996.

[11] J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew, and J. Sun, "Accurate and numerically efficient r2scan meta-generalized gradient approximation," *The journal of physical chemistry letters*, vol. 11, no. 19, pp. 8208–8215, 2020.

[12] OpenCode AI Team, “Opencode.” [Online]. Available: https://opencode.ai

[13] F. H. Bhuiyan, J. Guo, C. J. Benmore, A. L. Blockmon, D. Johnson, and A. Vazquez-Mayagoitia, “Structure of molten fecl2 and fecl3,” *ACS Materials Au*, 09 2026. [Online]. Available: https://doi.org/10.1021/acsmaterialsau.6c00155

[14] J. Wang, J. Wu, Z. Sun, G. Lu, and J. Yu, “Molecular dynamics study of the transport properties and local structures of molten binary systems (li, na) cl,(li, k) cl and (na, k) cl,” *Journal of Molecular Liquids*, vol. 209, pp. 498–507, 2015.

[15] J. Guo, L. Ward, Y. Babuji, N. Hoyt, M. Williamson, I. Foster, N. Jackson, C. Benmore, and G. Sivaraman, “Composition-transferable machine learning potential for licl-kcl molten salts validated by high-energy x-ray diffraction,” *Physical Review B*, vol. 106, no. 1, p. 014209, 2022.