The Journal of Open Source Software

ParaMonte II: An Ecosystem of Computational Algorithms for Parallel Monte Carlo Simulations and Machine Learning in C, C++, Fortran.

Amir Shahmoradi[1, 2], **Fatemeh Bagheri**[1], **and Joshua Alexander Osborne**[1]

1 Department of Physics, The University of Texas, Arlington, TX **2** Division of Data Science, The University of Texas, Arlington, TX

DOI:

Software

- Review
- Repository
- Archive

Editor:

Submitted: August 1, 2024
Published:

Summary

ParaMonte (standing for Parallel Monte Carlo) is a serial and MPI/Coarray-parallelized library of Monte Carlo routines and Machine Learning algorithms. Originally designed for optimizing, sampling, and integrating mathematical density functions (Shahmoradi & Bagheri, 2021), in particular, the posterior distributions of Bayesian models in data science, Machine Learning, and scientific inference, the new library release targets a much broader community of data scientists and ML/AI users within fields of science which traditionally require High-Performance Computing, such as Space Physics and Meteorology. The ParaMonte library has been developed to unify the **automation**, **accessibility**, **high-performance**, **scalability**, and **reproducibility** of Monte Carlo (MC) simulations and machine learning (ML) algorithms. The current major library release retains the functionalities of the original MC algorithms (e.g., **ParaDRAM**, a **Para**llel **D**elyaed-**R**ejection **A**daptive **M**etropolis Markov Chain Monte Carlo sampler (Shahmoradi & Bagheri, 2020), (Shahmoradi & Bagheri, 2020a), (Shahmoradi & Bagheri, 2020c), (Kumbhare & Shahmoradi, 2020), accessible from a wide range of programming languages including C, C++, Fortran, with a similar Application Programming Interface and simulation environment across all supported programming languages. But more importantly, the library has morphed into a 100% generic multi-precision collection of over 1000 fully documented, benchmarked, and exemplified new functionalities and algorithms for common machine learning tasks. The ParaMonte library is MIT-licensed as before and is permanently located and maintained at https://github.com/cdslaborg/paramonte.

Statement of need

ParaMonte II library release addresses the need for generic multi-precision algorithms for machine learning tasks within the HPC community. Numerical algorithms within the C and Fortran communities are frequently designed and optimized for double-precision floating-point format calculations, often corresponding to the IEEE 754 double-precision binary floating-point format. However, higher precisions than double may be essential in many modern use cases, or lower-precision calculations may suffice. Comparable libraries to ParaMonte's functionalities may be the Fortran standard library and the GNU C Scientific library. However, there is a significant discrepancy in the scope and functionalities of these libraries. The lack of multi-precision machine learning algorithms is particularly felt within the Fortran community (e.g., (Kedward et al., 2022), (Pribec et al., 2023)).

The origins of ParaMonte

The newest major release of the ParaMonte library grew out of the need for a free, user-friendly generic multi-precision high-performance parallel machine learning software with minimal external dependencies to reduce complexities of the build process and software version incompatibilities, a phenomenon that has become known as the *dependency-hell* among software developers. The project was originally meant to aid the library's primary authors and developers in their research in the fields of Astrophysics, Biophysics, Data Science, and Space Physics (Shahmoradi, 2013), (Shahmoradi, 2013), (Shahmoradi & Nemiroff, 2014), (Shahmoradi & Nemiroff, 2015), (Shahmoradi & Nemiroff, 2019), (Osborne, Shahmoradi, & Nemiroff, 2020), (Osborne, Shahmoradi, & Nemiroff, 2020). Over the past years, it remained in private usage until the spring of 2024, when we released version 2.0 of the library as an open-source project for public usage and contributions.

The build process

The ParaMonte library is permanently on GitHub and can be viewed at https://github.com/cdslaborg/paramonte. The library's building process is fully automated. Extensive detailed instructions are also available on the documentation website of the library and in the root directory of the project's repository on GitHub. The library has been meticulously carefully designed to preserve its portability while minimally requiring the CMake build generator software and a C/Fortran compiler suite for a successful build on any platform. Specifically, the latest GNU and Intel compiler suites are guaranteed to compile the library by design without failure.

The ParaMonte library modules

In addition to enhancements to originally existing routines of the C/C++/Fortran interfaces to the library (e.g., the Monte Carlo samplers (Shahmoradi & Bagheri, 2021), (Shahmoradi & Bagheri, 2020), (Shahmoradi & Bagheri, 2020b), (Shahmoradi & Bagheri, 2020c), (Kumbhare & Shahmoradi, 2020)), the new library release contains a comprehensive list of generic multi-precision interfaces that can be used for the following major tasks: 1) type coercion, 2) string manipulation, 3) polynomial calculations, 4) common filesystem tasks, 5) calendrical calculations, 6) common operating system tasks, 7) computing properties of statistical samples, 8) common numerical algebraic and mathematical tasks, 9) manipulating arrays of arbitrary intrinsic types and kind parameters, 10) computing properties of famous statistical distributions with arbitrary precision, 11) generating deterministic pseudo-random numbers from various statistical distributions, 12) optimization and sampling of mathematical density functions with arbitrary precision, and 13) many other functionalities that are further discussed in the library documentations.

The library also contains a modernization and significant extension of the venerable **FFTPACK** library to support multi-precision Fast-Fourier Transforms and a modernization and significant extension of the venerable **QuadPack** library to support multi-precision computations of Cauchy Principal Values and one-dimensional finite, semi-finite, and infinite integrations with potential singularities.

Parallelism

All stochastic sampling and integration generic library interfaces in all supported programming languages support parallelization of the simulations on shared and distributed memory architectures using OpenMP and Message Passing Interface (MPI) communication protocols.

The ParaMonte Application Programming Interface

Special care has been made to develop highly-similar (if not the same) Application Programming Interface (API) to the ParaMonte library samplers across all supported programming languages. However, the new interfaces are incompatible with the previous major library release. The project uses semantic versioning to readily inform the users about the backward-compatibility of each release.

Tests and code coverage

ParaMonte II comprises nearly 1,000,000 lines of code, primarily in modern Fortran, C, and C++. We have developed tens of performance benchmarks, tens of thousands of examples, and nearly 2.5 million individual tests that verify the library's functionalities and performance. The code coverage report analyses of the ParaMonte kernel library can be generated as outlined in the library documentation. The tests currently cover over 70% of the library.

Documentation and Repository

Extensive documentation and examples in C, C++, Fortran (as well as other programming languages) are available on the documentation website of the library at https://www.cdslab.org/paramonte/c/latest, https://www.cdslab.org/paramonte/cpp/latest, https://www.cdslab.org/paramonte/fortran/latest. Extensive generic documentation https://www.cdslab.org/paramonte/ also covers topics shared between all programming language enviroments (e.g., the installation procedure). The ParaMonte library is MIT-licensed and is permanently located and maintained at https://github.com/cdslaborg/paramonte.

A high number of comments in a codebase frequently indicates an organized, well-documented project and is considered a sign of a helpful and disciplined development team. As of August 2024, comments approximately make up 22% of the entire ParaMonte codebase. Currently, the Open Hub indexer service ranks the ParaMonte project's level of documentation as "impressive" and among the top 10% of all projects in the same category on Open Hub.

Funding and Acknowledgements

Open Hub estimates the ParaMonte library's development cost at 237 years of effort. Considering the redundancy associated with generic multi-precision interfaces (roughly inflating the library size 5-fold) and the presence of a large number of example scripts in the library (comprising roughly 10% of the library), an estimate of $30 - 40$ years of developer effort seems more realist. The current state of the library has resulted from more than a decade of continuous development, predominantly by the core developers of the library as graduate students, postdocs, and faculty in their free time. We thank the Texas Advanced Computing Center for providing the supercomputer time for testing and developing this library.

References

Kedward, L. J., Aradi, B., Certik, O., Curcic, M., Ehlert, S., Engel, P., Goswami, R., et al. (2022). The state of fortran. *Computing in Science & Engineering*, *24*(2), 63–72.

Kumbhare, S., & Shahmoradi, A. (2020). Parallel adapative monte carlo optimization, sampling, and integration in c/c++, fortran, matlab, and python. *Bulletin of the American Physical Society*.

Osborne, J. A., Shahmoradi, A., & Nemiroff, R. J. (2020). A multilevel empirical bayesian approach to estimating the unknown redshifts of 1366 batse catalog long-duration gamma-ray bursts. Retrieved from http://arxiv.org/abs/2006.01157

Osborne, J. A., Shahmoradi, A., & Nemiroff, R. J. (2020). A Multilevel Empirical Bayesian Approach to Estimating the Unknown Redshifts of 1366 BATSE Catalog Long-duration Gamma-Ray Bursts, *903*(1), 33. doi:10.3847/1538-4357/abb9b7

Pribec, I., Magnin, V., Lozada-Blanco, A., Certik, O., Vandenplas, J., Curcic, M., Goswami, R., et al. (2023). Introducing the fortran-lang community. In *DeRSE23-conference for research software engineering in germany*.

Shahmoradi, A. (2013). A multivariate fit luminosity function and world model for long gamma-ray bursts. *The Astrophysical Journal*, *766*(2), 111.

Shahmoradi, A. (2013). Gamma-Ray bursts: Energetics and Prompt Correlations. *arXiv e-prints*, arXiv:1308.1097. Retrieved from http://arxiv.org/abs/1308.1097

Shahmoradi, A., & Bagheri, F. (2020). ParaDRAM: A cross-language toolbox for parallel high-performance delayed-rejection adaptive metropolis markov chain monte carlo simulations. *arXiv preprint arXiv:2008.09589*.

Shahmoradi, A., & Bagheri, F. (2020a). ParaDRAM: A Cross-Language Toolbox for Parallel High-Performance Delayed-Rejection Adaptive Metropolis Markov Chain Monte Carlo Simulations. *arXiv e-prints*, arXiv:2008.09589. Retrieved from http://arxiv.org/abs/2008.09589

Shahmoradi, A., & Bagheri, F. (2020b). ParaDRAM: A Cross-Language Toolbox for Parallel High-Performance Delayed-Rejection Adaptive Metropolis Markov Chain Monte Carlo Simulations. *arXiv e-prints*, arXiv:2008.09589. Retrieved from http://arxiv.org/abs/2008.09589

Shahmoradi, A., & Bagheri, F. (2021). ParaMonte: A high-performance serial/parallel Monte Carlo simulation library for C, C++, Fortran. *The Journal of Open Source Software*, *6*(61), 2741. doi:10.21105/joss.02741

Shahmoradi, A., & Bagheri, F. (2020c, August). ParaMonte: Parallel Monte Carlo library.

Shahmoradi, A., & Nemiroff, R. (2014). Classification and energetics of cosmological gamma-ray bursts. In *American astronomical society meeting abstracts# 223* (Vol. 223).

Shahmoradi, A., & Nemiroff, R. J. (2015). Short versus long gamma-ray bursts: A comprehensive study of energetics and prompt gamma-ray correlations. *Monthly Notices of the Royal Astronomical Society*, *451*(1), 126–143.

Shahmoradi, A., & Nemiroff, R. J. (2019). A Catalog of Redshift Estimates for 1366 BATSE Long-Duration Gamma-Ray Bursts: Evidence for Strong Selection Effects on the Phenomenological Prompt Gamma-Ray Correlations. *arXiv e-prints*, arXiv:1903.06989. Retrieved from http://arxiv.org/abs/1903.06989