# STORM: Scalable One-Sided Communication for Distributed Monte Carlo Particle Transport

Maor Mizrachi
maormiz@cs.huji.ac.il
The Hebrew University of Jerusalem
Jerusalem, Israel

## Abstract

Domain-decomposed Monte Carlo transport transfers particles between ranks when they cross subdomain boundaries. Irregular, bursty traffic makes communication and completion detection difficult. STORM combines one-sided remote direct memory access (RDMA) through the OpenFabrics Interfaces (OFI/libfabric) with batched queues, ordered publication, dynamic growth, and distributed termination. On 2,240–13,440 ranks of Leonardo, it achieves 88.3% constant-global-workload scaling efficiency and over 97% weak-scaling efficiency. Against STORM's optimized nonblocking point-to-point MPI backend, OFI is up to 1.27× faster in the synthetic benchmark and 1.41× faster in one 4,480-rank Implicit Monte Carlo Hohlraum run.

## Keywords

Monte Carlo transport, RDMA, OFI, distributed memory, particle migration

## 1 Problem and Contribution

Domain decomposition assigns mesh regions to ranks; particles that cross a boundary must reach the next owner. Prior transport algorithms use asynchronous messages and global completion coordination [1, 2, 5]. A remote write alone does not solve queue publication, overflow, and termination while particles remain in flight. Remote-write queues and termination detection are established techniques; STORM's contribution is their integration into a reusable particle-migration protocol with dynamic queue growth and an application-facing transport interface [8]. This study evaluates its OFI backend against its own optimized MPI point-to-point backend. STORM also has an MPI remote-memory-access (RMA) path, but neither that path nor NVSHMEM was benchmarked here.

## 2 Protocol and Matched Comparison

The application advances particles and identifies destinations; STORM transports application-defined records and detects distributed completion. Each communicating rank pair uses a single-producer, single-consumer ring in registered receiver memory. The sender batches records, checks capacity, writes payloads, and publishes a new tail only after those payloads are remotely visible. The receiver consumes entries below the published tail. When the ring fills, the receiver preserves unread records, registers larger storage, and communicates new metadata; a pairwise lock protects replacement. Distributed accounting includes local and in-flight work.

Both compared implementations are STORM backends: the baseline uses nonblocking MPI point-to-point messages, and the evaluated backend uses OFI one-sided writes. They share particle transport, mesh decomposition, generated workload, aggregation threshold, and termination framework. Their queue implementations follow their respective communication semantics. OFI avoids matched receives on the particle-transfer path; the receiver still checks published work.

## 3 Evaluation and Scope

The synthetic test uniformly emits particles into a transparent three-dimensional Voronoi mesh with reflecting boundaries, inducing frequent crossings. Leonardo runs use 2,240–13,440 MPI ranks, one per physical core. Reported points are medians of ten runs, timing the final three transport cycles.

For *constant-global-workload* scaling, cells per rank remain fixed while the mesh grows; the timestep shrinks so the expected global particle-step count stays approximately constant. This is not fixed-mesh strong scaling. Weak scaling instead fixes cells, emissions, and timestep per rank. Efficiency compares elapsed time at each rank count with the 2,240-rank time for the corresponding experiment. OFI retains 88.3% constant-global-workload efficiency and over 97% weak-scaling efficiency at 13,440 ranks; across tested points it is 1.14–1.27× faster than the MPI baseline.

These comparisons do not establish an advantage over other one-sided systems. Per-transfer latency, provider-dependent ordering behavior, and any additional core usage by progress threads were not separately characterized in these results.

## 4 Application, Validation, and Student Work

STORM communicates boundary-crossing photon packets for RICH, which handles the radiation physics and mesh. Its repository includes Cartesian and Voronoi validation cases based on nonlinear Marshak waves [4, 6], a multigroup step-opacity problem [3], and a frequency-dependent moving-slab benchmark [7]. In one three-dimensional Hohlraum run at 4,480 ranks, OFI is 1.41× faster than STORM's MPI backend (Fig. 1).

The student designed and implemented the communication protocol, queue-growth and termination mechanisms, optimized MPI baseline, benchmarks, RICH integration, and large-scale evaluation. Code and regression tests are public.[1]

[1] https://github.com/maormizrachi/STORM

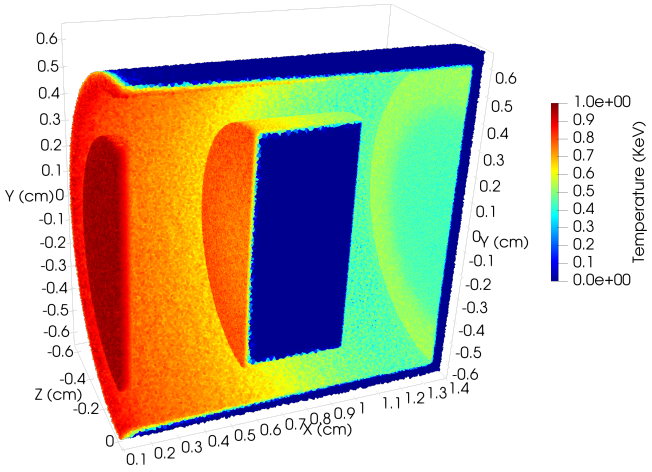



**Figure 1: Hohlraum on an unstructured Voronoi mesh. The rendered material regions are colored by temperature; the image illustrates the application workload, not a communication measurement.**

## 5 Conclusion

STORM integrates a receiver-passive OFI transfer path with the queue management and completion logic needed by irregular particle transport. Its measured advantage is over the matched STORM MPI backend on the reported workloads and systems.

## Acknowledgments

This work used EuroHPC Joint Undertaking resources on Leonardo at CINECA (EHPC-DEV-2026D04-204). Portions of the prose were generated, rephrased, or revised using OpenAI ChatGPT (GPT-6 Astra); the author reviewed and assumes responsibility for the content [9].

## References

[1] Thomas A. Brunner and Patrick S. Brantley. 2009. An Efficient, Robust, Domain-Decomposition Algorithm for Particle Monte Carlo. *J. Comput. Phys.* 228, 10 (2009), 3882–3890. doi:10.1016/j.jcp.2009.02.013

[2] Thomas A. Brunner, Todd J. Urbatsch, Thomas M. Evans, and Nicholas A. Gentile. 2006. Comparison of Four Parallel Algorithms for Domain Decomposed Implicit Monte Carlo. *J. Comput. Phys.* 212, 2 (2006), 527–539. doi:10.1016/j.jcp.2005.07.009

[3] Jeffery D. Densmore, Kelly G. Thompson, and Todd J. Urbatsch. 2012. A Hybrid Transport-Diffusion Monte Carlo Method for Frequency-Dependent Radiative-Transfer Simulations. *J. Comput. Phys.* 231, 20 (2012), 6924–6934. doi:10.1016/j.jcp.2012.06.020

[4] Nitay Derei, Shmuel Balberg, Shay I. Heizler, Elad Steinberg, Ryan G. McClarren, and Menahem Krief. 2024. The Non-Equilibrium Marshak Wave Problem in Non-Homogeneous Media. *Physics of Fluids* 36, 12 (2024), 127149. doi:10.1063/5.0244247

[5] Tim J. Harries. 2015. Radiation-Hydrodynamical Simulations of Massive Star Formation Using Monte Carlo Radiative Transfer—I. Algorithms and Numerical Methods. *Monthly Notices of the Royal Astronomical Society* 448, 4 (2015), 3156–3166. doi:10.1093/mnras/stv158

[6] Menahem Krief and Ryan G. McClarren. 2024. Self-Similar Solutions for the Non-Equilibrium Nonlinear Supersonic Marshak Wave Problem. *Physics of Fluids* 36, 1 (2024), 017108. doi:10.1063/5.0186666

[7] Ryan G. McClarren and N. A. Gentile. 2021. Frequency-Dependent Material Motion Benchmarks for Radiative Transfer. In *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. Raleigh, North Carolina. arXiv:2009.01905 [cs.CE].

[8] Maor Mizrachi, Barak Raveh, and Elad Steinberg. 2026. STORM: RDMA-Based Monte Carlo Transport Scheme for Distributed-Memory Particle Simulations. *The Astrophysical Journal Supplement Series* (2026). arXiv:2607.20639 [cs.DC] doi:10.3847/1538-4365/aea3f8

[9] OpenAI. 2026. ChatGPT (GPT-6 Astra). Large language model used for writing assistance. Accessed September 2026.