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


Maor Mizrachi

The Hebrew University of Jerusalem, Israel


## Motivation & Challenge†

Monte Carlo transport tracks representative photon or neutrino packets as they move through and interact with matter. Distributed simulations divide the domain among MPI ranks. When a particle crosses a subdomain boundary, it moves to the rank responsible for the neighboring region.

These crossings occur unpredictably between irregular rank pairs. Traditional two-sided MPI therefore requires receiver participation, polling, and synchronization that can become a major scalability bottleneck.

## Research Gap & Contributions†

Prior domain-decomposed Monte Carlo implementations use matched MPI messages for particle migration [2], requiring receiver-side progress and completion handling.

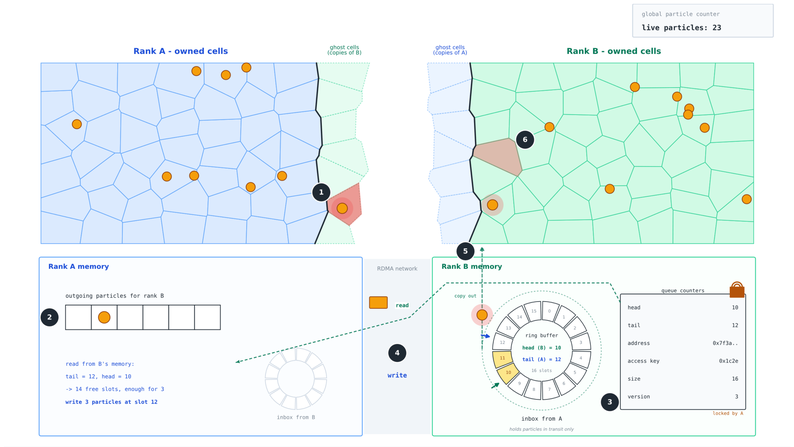
STORM [1] supplies reusable communication infrastructure independent of the physics model and mesh. The application advances particles and selects destinations.

STORM provides batching, receiver-owned queues, ordered publication, dynamic buffer growth, and distributed termination.

The open-source protocol combines receiver-passive transfers, safe publication, queue growth, and distributed completion. Its evaluated OFI (libfabric) backend scales to 13,440 ranks.

## Core Idea†

The sender accumulates particles into batches to amortize transfer and publication overhead. When a batch is ready, it writes into a peer-owned queue and publishes the new tail only after the payload is remotely visible. The receiver consumes entries below that tail when it services the queue; it posts no matching receive for the transfer.




Particle data is deposited directly into remote memory, and the receiver does not post a matching receive.

## Methods†

- Dedicated single-producer/single-consumer queues isolate traffic between geometrically neighboring ranks.
- A capacity check precedes each write; the sender publishes the remote tail only after the payload is visible.
- Bursts trigger queue growth. A pairwise lock protects replacement without blocking unrelated rank pairs.
- Distributed accounting tracks particles that remain in flight.

## Benchmark†

**Uniform-emission stress test on a 3D unstructured Voronoi mesh.**

The material is transparent (particles move easily) and boundaries are reflecting, forcing photons to cross many rank boundaries and therefore creating a communication-dominated workload.

Voronoi cells were generated randomly and uniformly.

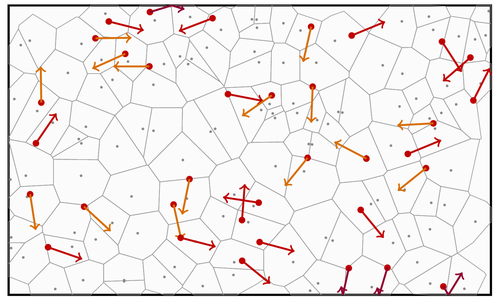


Illustration of the communication stress test: particles migrate between processor-owned Voronoi subdomains.

**Setup:**

- Tested on 2,240–13,440 ranks on Leonardo, hosted by CINECA (Italy).
- One physical core per MPI rank.
- For work-preserving scaling, cells per rank remain fixed while the global mesh grows. The timestep is reduced as $\Delta t \propto P^{-4/3}$ to keep the expected global particle-step count approximately constant and avoid cache-driven superlinear scaling. Due to the fact the timestep changes, this is not a traditional strong scaling.

## Portability

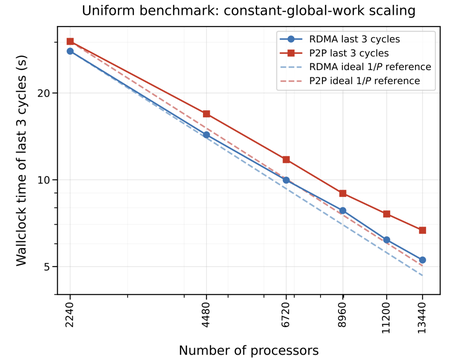
STORM's OFI backend has been also validated on Frontier (ORNL), Snellius (SURF), and ICPL (HUJI). The full regression suite passes on all systems, and allocation-limited runs preserve the expected scaling behavior within the tested ranges. Detailed large-scale results are shown for Leonardo, where the available allocation enabled testing to 13,440 ranks.

## Results†

We compare two STORM backends: optimized point-to-point (P2P) MPI and OFI/libfabric remote direct memory access (RDMA) over InfiniBand Verbs.

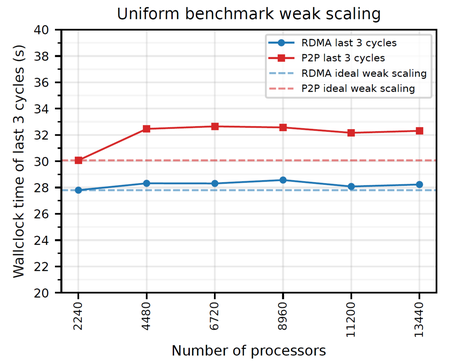
Matched transport, decomposition, workload, and batching threshold; queue structures follow each backend's semantics. Blue curves: STORM OFI/RDMA. Red curves: STORM MPI/P2P.

**Work-Preserving Scaling:**




STORM OFI/RDMA: 88.3% work-preserving efficiency relative to 2,240 ranks. Leonardo; median of 10 runs.

**Weak Scaling (Constant Workload Per Rank):**



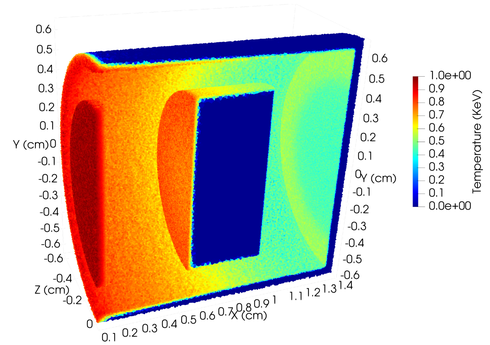

STORM OFI/RDMA: $>$ 97% weak-scaling efficiency relative to 2,240 ranks. Leonardo; median of 10 runs.

## Integration†

STORM was developed to provide the communication infrastructure for the Monte Carlo radiation-transport module in RICH, a three-dimensional astrophysical hydrodynamics code. RICH simulates radiation by tracking representative photon packets as they move through and exchange energy with matter.

For benchmarking, STORM also includes a radiation transport implementation, and its modular design supports different physics models and mesh types.

The IMC implementation was validated against established benchmarks included in the public repository (as a test suite). The public validation suite includes analytically solvable problems, all of which pass their prescribed numerical tolerances.




A three-dimensional hohlraum benchmark simulated using STORM's new Monte Carlo radiation-transport module. Color indicates the material temperature.

STORM's RDMA backend was 1.41× faster than optimized P2P MPI in the 3D Hohlraum benchmark at 4,480 ranks.

## Conclusion†

| | | |
|---|---|---|
| **13,440** ranks tested | **88.3%** Work-preserving scaling efficiency(*) | $>$ 97% weak scaling efficiency(*) |
| **Up to** 1.27× **faster** than optimized P2P MPI in benchmark | | 1.41× **faster** than optimized P2P MPI in the 3D Hohlraum application |

Across these runs, STORM OFI/RDMA outperforms STORM's optimized MPI/P2P backend. Its advantage grows under work-preserving scaling as communication pressure increases.

* Efficiencies compare 2,240 and 13,440 ranks. MPI RMA and NVSHMEM were not benchmarked.

## References


[1] M. Mizrachi et al., "STORM: RDMA-Based Monte Carlo Transport Scheme for Distributed-Memory Particle Simulations", *ApJS* (2026), doi:10.3847/1538-4365/aea3f8.

[2] T. J. Harries, MNRAS 448, 3156–3166 (2015).


† Sections marked were generated or revised with OpenAI ChatGPT (GPT-6 Astra), accessed September 2026. The author reviewed and takes responsibility for the content.

**Visit Us on GitHub!**

## Contact


maormiz@cs.huji.ac.il


## Acknowledgements


This work used computational resources awarded by the EuroHPC Joint Undertaking (project EHPC-DEV-2026D04-204), on Leonardo hosted by CINECA.
AI-assisted prose used OpenAI ChatGPT (GPT-6 Astra), September 2026. The author reviewed all content.