Bigger Networks, More Trajectories: Scaling Exact Stochastic Simulation on GPUs through Sparse Storage and a Warp-ary Propensity Tree

Alfia Parvez[1,3], Zaida(Zan) Luthey-Schulten[1,2,3]

NSF STC for Quantitative Cell Biology[1], Department of Chemistry and Physics, University of Illinois Urbana-Champain (UIUC)[2], Beckman Institute[3]

MOTIVATION: the two walls

Low-copy-number biochemistry is intrinsically stochastic. Deterministic rate equations report only mean behavior, whereas the **Chemical Master Equation** (CME) must be sampled with **Gillespie's Direct Stochastic Simulation Algorithm** (SSA) to recover fluctuations, distributions, and rare-event statistics. Two facts collide on modern hardware:

Models are now huge: Rule-based networks routinely reach tens of thousands of reactions (M).
Statistics demand breadth: A single SSA trajectory is one noisy sample; reliable means, distributions, and rare-event estimates require **thousands of independent replicates**.

Conventional exact-SSA implementations hit **two walls:**

- **Memory wall:** Species(N) * Reactions(M) and dense stoichiometry tables reach gigabytes;
- **Selection wall:** naive Direct-Method reaction selection is O(M) per step, and every step is a tightly-coupled microseconds-scale dependency.

Take-home: Exact Stochastic Simulation scales with number of reactions for **large reaction network** models and number of trajectories which breaks dense SSA.

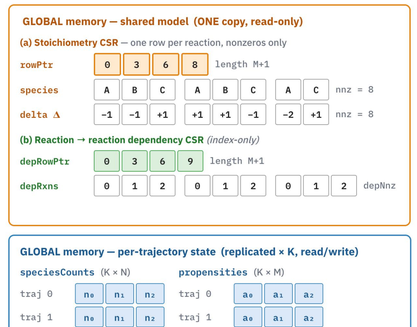
IDEA

We introduce a GPU based Gillespie Direct Solver in Lattice Microbes (LM) software :

LM GPU Gillespie Direct Solver: Hybrid parallelism both fine grained and coarse grained using a reaction selection method inspired by Komarov & D'Souza's two-level search for generating thousands of trajectories and storing data in a **Compressed Sparse Row (CSR)** format.

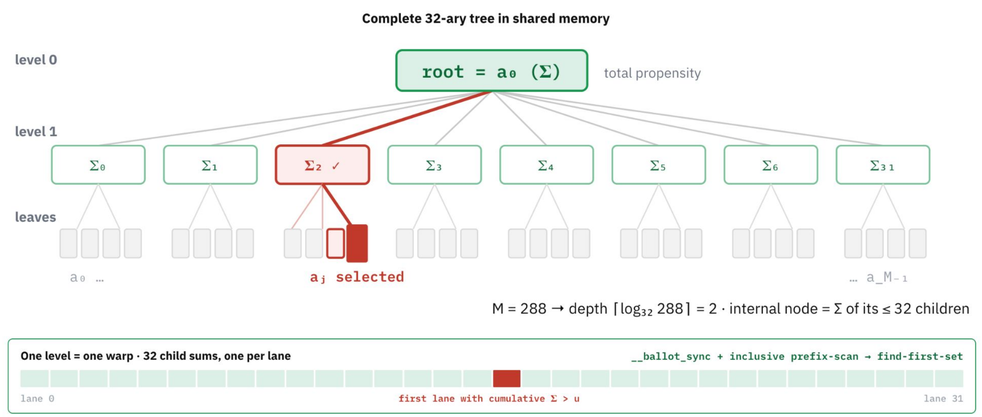
Sparse storage of data

one CUDA block = one warp (32 lanes) = one trajectory • example: N = 3 species (A, B, C), M = 3 reactions, nnz = 8



Total device memory = O(M + nnz) + O(K·(N + M)) never the dense O(M·N) that walls large models on coarse GPU tools.

Warp-ary (32-ary) propensity tree — $O(\log_{32} M)$ exact selection



GPU Solver Algorithm Summary

One warp (32 threads) per trajectory: thousands of trajectories concurrently on one GPU.

- **CSR storage format:** stoichiometry + dependency graph in O(nnz); large models stay resident on-device
- **Reaction selection** : We build a complete 32-ary propensity tree in shared memory: leaves are per-reaction propensities; each internal node sums its ≤32 children; the root is the total propensity.
- **Selection:** Optimized Komarov & D'Souza's two-level search -> $O(\log_{32}M)$, replacing the O(M/32) linear within-block scan.
- **Update:** a changed leaf's delta is scattered into distributed level-1 bins (atomics), then the tiny upper levels are rebuilt.

RESULTS SUMMARY

- **Breaking the memory wall:** O(M+nnz) CSR footprint, ~800× reduction on FcεRI. Our solver ran for all 5 networks without running out of memory (OOM) compared to its counterparts Köster (infeasible for M ≥ 3,749) and crn-jax (~0.5 TB dense store OOM).
- **Breaking the Selection wall:** Per-step cost stays sublinear shown in Plot 2. with near-constant T_up ($4.3\times10^{-8} \rightarrow 4.0\times10^{-7}$ s over a 3,000 × M increase) vs the ~O(M) coarse Direct Method, resulting in 2–3× and ~12× speedups for different model sizes.
- **Ensemble throughput scales with K:** Plot 1 shows saturation behavior and sustained steps/s.
- **Head-to-head and correctness:** The correctness of the solver was validated against the well-validated CPU reference StochKit2. (shown in Plot 3)

BACKGROUND

Exact SSA on the CPU is mature:

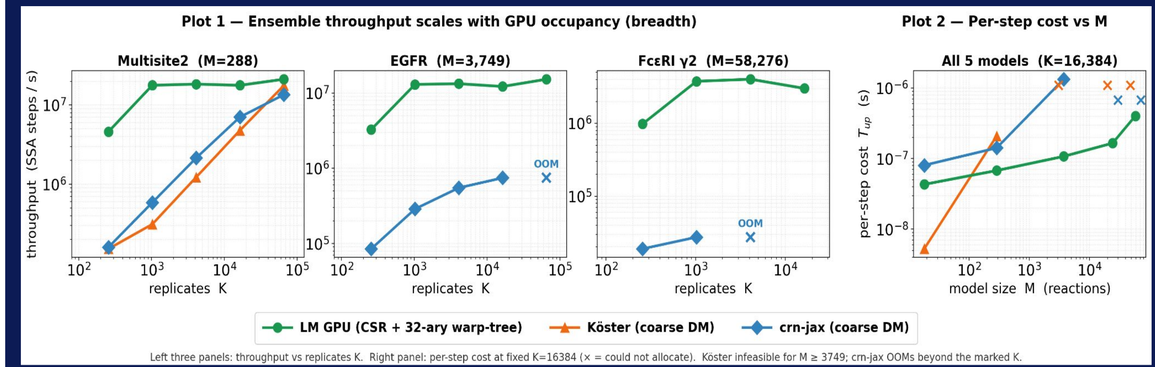
- **Ordering-based (ODM, SDM):** O(M) worst-case selection, but frequently firing reactions are moved earlier to reduce average search cost.
- **Logarithmic (LDM, NRM):** O(logM) selection using partial-sum trees or indexed priority queues.
- **Constant-time (CR, PSSA-CR):** expected O(1) selection when the number of propensity groups remains bounded.
- **Sparse updates:** dependency graphs update only the D affected reactions, giving O(D) work with $D \ll M$, rather than recomputing all M propensities.

GPU based solvers scale mainly by coarse ensembles (one trajectory/thread) and developing:

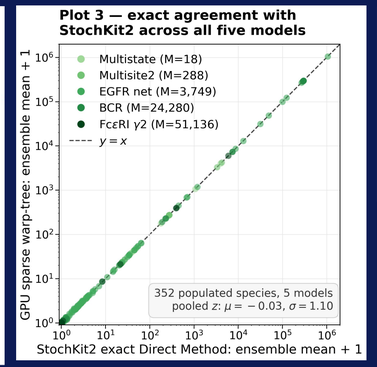
- **Coarse** [1,2]: 10^3–10^6 trajectories can be realized for small models, but each thread runs the serial step → **cost + memory** grow with dense model size.
- **Fine-grained** [3,4]: parallelize updates/selection across a warp but use redundant or limited sparse representations and lack publicly available source code.

RESULTS

We benchmarked the solver against publicly available GPU-based SSA tools on five reaction models spanning three orders of magnitude in size: multistate **(Multistate, N=9, M=18)**, multi-site phosphorylation **(Multisite2, N=66, M=288)**, epidermal growth factor receptor signalling **(Egfr_net, N=356, M=3749)**, B-cell receptor signalling **(BCR, N=1122, M=24388)**, and high-affinity human IgE receptor signalling **(Fceri_gamma2, N=3744, M=58276)**. The results were benchmarked on NVIDIA B200 GPUs.



Left three panels: throughput vs replicates K. Right panel: per-step cost at fixed K=16384 (× = could not allocate). Köster infeasible for M ≥ 3749; crn-jax OOMs beyond the marked K.



FUTURE WORK

Open-source release in Lattice Microbes v3.0: The solver is implemented and validated. We plan an open-source release in Lattice Microbes v3.0, with the 32-ary warp-tree as the default GPU selection path, direct SBML/BioNetGen import, and the reproducibility benchmark suite.

This work is funded by NSF STC Quantitative Cell Biology which aims to develop 4D (space plus time) whole-cell models and experiments to transform our understanding of how cells function, forging a new scientific understanding of the cell that will have implications for human health, photosynthesis, evolution, and beyond.

REFERENCES

[1] T. Köster, L. Herrmann, P. Andelfinger, and A. M. Uhrmacher, "GPU-accelerated simulation ensembles of stochastic reaction networks," in Proc. Winter Simulation Conf. (WSC), 2022, pp. 2570–2581.

[2] R. Henry, "crn-jax: Chemical reaction networks in JAX—a GPU-optimized Gillespie/SSA library," v0.3.1, 2026. [Online]. Available: https://github.com/robinhenry/crn-jax

[3] I. Komarov and R. M. D'Souza, "Accelerating the Gillespie exact stochastic simulation algorithm using hybrid parallel execution on graphics processing units," PLoS ONE, vol. 7, no. 11, p. e46693, 2012.

[4] K. Sumiyoshi, K. Hirata, N. Hiroi, and A. Funahashi, "Acceleration of discrete stochastic biochemical simulation using GPGPU," Front. Physiol., vol. 6, art. 42, 2015.

[5] E. Roberts, J. E. Stone, and Z. Luthey-Schulten, "Lattice Microbes: High-performance stochastic simulation method for the reaction-diffusion master equation," J. Comput. Chem., vol. 34, no. 3, pp. 245–255, 2013.