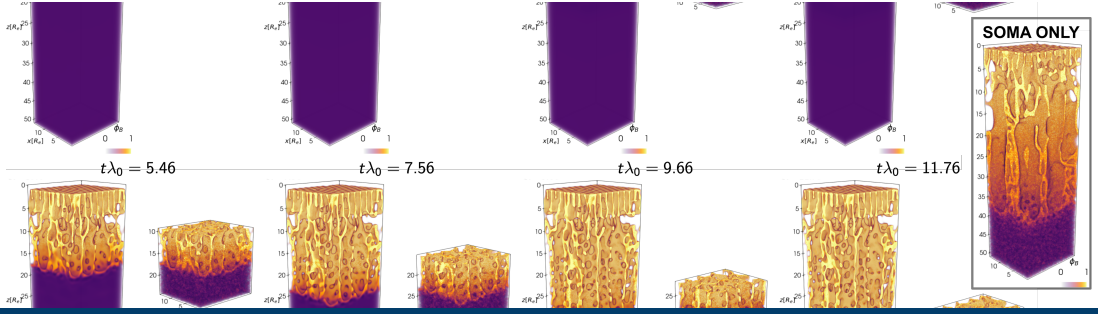


COUPLING COMPLEMENTARY SIMULATIONS FOR ENERGY OPTIMIZATION

26 June 2026 | **Adel Dabah**, G. Häfner, S. Happ, S. Pickartz, M. Müller, A. Herten | JSC, ParTec, U. Göttingen

MOTIVATION

Goal: digital twin for the fabrication and optimization of soft matter physics and materials science.

- Lab experiments are time-consuming, expensive, and not scalable.
- Simulations are a cost-effective solution → ***efficiency and scalability***.
- HPC tracks thousands to millions of interacting particles over long timescales → allowing **real-time digital twin interaction**.
  - Faster computation and better resolution of material properties.
  - Parameter space exploration → **impossible** otherwise.

The Problem

- Polymer simulations take days and large power consumption.
- GPU scaling increases energy per step (paradox!)

SOMA AND THE ENERGY SCALING

SOMA (SOft coarse-grained Monte-carlo Acceleration) is a particle-based Monte Carlo simulation for polymer and soft matter systems.

- Used at computational materials science at mesoscopic scales.
- Monte Carlo chains
- Slow but accurate (Thermal fluctuations)
- Small domains i.e. not scalable

SOMA GPU Scaling

	4 GPU	8 GPU	12 GPU	16 GPU
Speedup	1.0×	1.67×	2.05×	2.33×
Efficiency	100%	84%	68%	58%
Energy/step	1.0	1.40	1.58	1.72
Steps/kJ	1.0×	0.71×	0.63×	0.58×

Energy per step INCREASES 72% despite 2.33× speedup!

Why This Happens

$$E(N) \approx \frac{P_{\text{GPU}} \cdot T(1)}{\eta(N)}$$

As parallel efficiency η drops:

- Communication overhead grows
- Processors idle more and consumes power
- Total energy increases

More GPUs != Less Energy

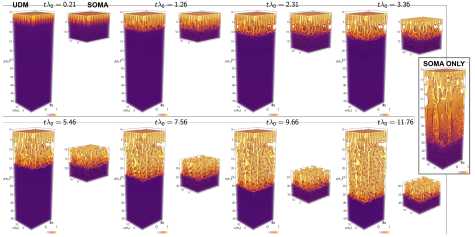
SOLUTION

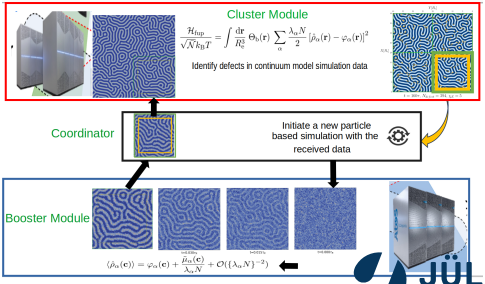
Our Solution: Coupling two complementary models

- Couple continuum (UDM) + particle (SOMA) models
- Run cheap UDM everywhere
- Spawn expensive SOMA only where needed

Results

Speedup: **13×**
Energy: **707.5 J** → **28.9 J**
Reduction: **96%**
Fidelity: **maintained**
Doing more with less:
4 A100 GPUs (coupled)
8 A100 GPUs (baseline)





UNEYAMA–DOI MODEL (UDM)

UDM is a continuum model that propagates local concentration fields using PDEs, capturing the diffusive dynamics of monomer species driven by solvent–nonsolvent.

- PDEs for concentration field evolution
- Fast and cheap in terms resources utilization
- Scalable for large domains
- Lacks chain details (not accurate)

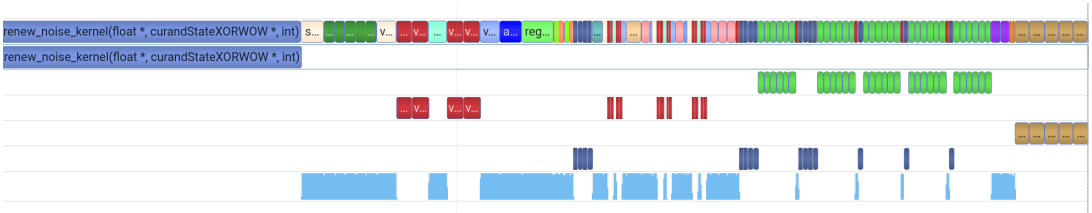


Figure: Nsight Systems timeline illustrating the execution of one UDM time step

UDM OPTIMIZATION

Bottlenecks Found

- 112 kernels per step
- `renew_noise`: 26% time
- Fourier transforms: 16%

Optimization 1: Kernel Fusion

- 112 $\rightarrow$ 69 kernels
- Global writes: -87% and +20% Gain

Optimization 2: Async RNG

- Pre-generate on GPU
- Overlap with compute
- `renew_noise`: 26% $\rightarrow$ 2.9%
- Gain: +30% i.e. Total of +70% speedup

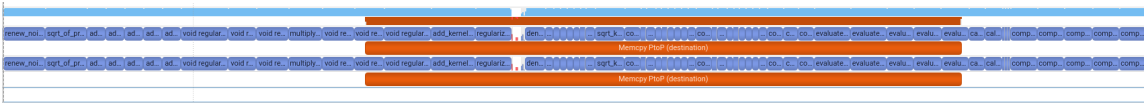


Figure: UDM random number generation overlapped with the simulation

SOMA OPTIMIZATION

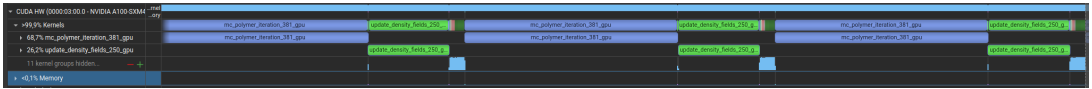


Figure: Two kernels dominate (95%) of SOMA execution

Bottlenecks Found: `mc_polymer`: 65%; `update_density`: 35%; Initialization: 235 seconds

Optimization 1: GPU Init

- Move to GPU
- 235s $\rightarrow$ 1s (Critical for coupling)

Optimization 2: Memory

- Coalesced access
- Reduce registers
- Runtime: -40%

Results: A100: +56% and H100: +119%

OPTIMIZATION RESULTS

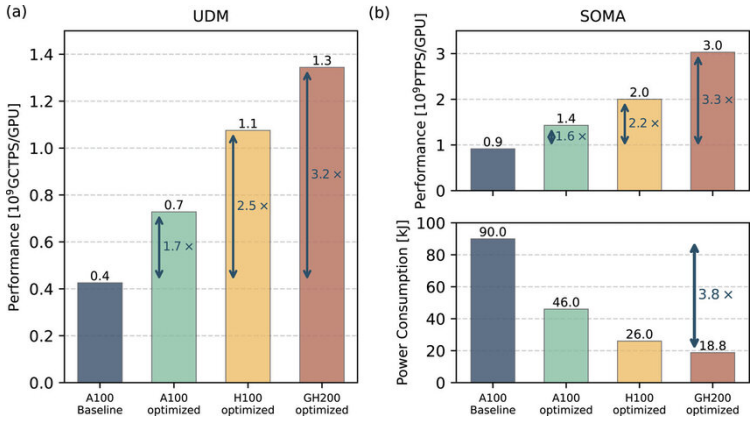


Figure: Performance improvements of (a) the UDM program and (b) the SOMA code achieved by individual optimizations using a single GPU. It is given for the UDM in units of Grid-Cell Time steps Per Second (GCTPS) per GPU and for SOMA in units of PTPS per GPU.

COORDINATOR LIBRARY

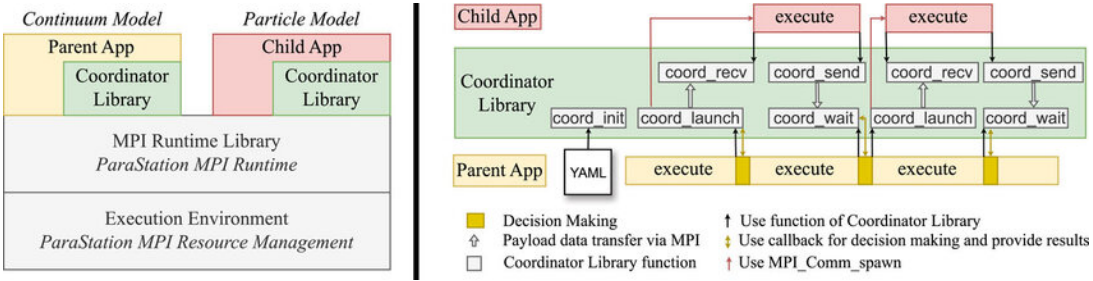


Figure: Overview of coordinator library integration into MPI applications and the ParaStation MPI runtime environment (left) and overall workflow, including coordinator API used by parent and child applications (right).

DECISION MAKING

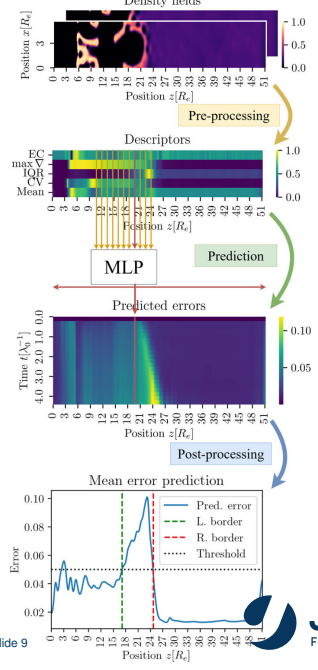
Architecture

- **Parent:** UDM (full domain)
- **Child:** SOMA (spawned)
- **Coordinator:** MPI-based
- **Decision:** ML-driven

Spawn Decision

- Input: UDM fields
- Model: MLP
- Output: error prediction
- Action: spawn if threshold
- Cost: negligible

Dynamic resource allocation based on simulation state → **Energy gain**



JÜLICH Forschungszentrum

COUPLING PROTOCOL

Synchronization every $t_{\text{sync}} = 0.16\,\lambda_0^{-1}$

1. UDM sends concentration fields to ML
2. ML identifies subdomain boundaries
3. Spawn SOMA with parameters (HDF5)
4. SOMA equilibrates particles
5. Boundary zones: particle-continuum conversion
6. UDM integrates SOMA results
7. Check flux; respawn if needed

COUPLED-VERSION PROFILING

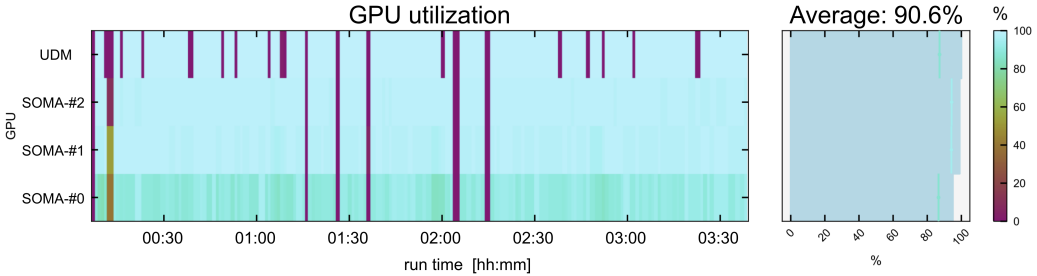


Figure: GPU utilization for the multi-scale coupled application.

Figure: Nvidia Nsight Systems profiling for the coupled UDM-SOMA multi-scale application.

LOAD BALANCING

The Trade-off

Subdomain size L_{sd} affects idle time:

- Too small: SOMA finishes early → UDM waits
- Too large: UDM finishes early → waits
- Optimal: near-zero idle = max speedup

Sweet Spots

- $\{L_{\text{sd}} = 12R_e, L_z = 78R_e\} \rightarrow 13\times$ speedup
- $\{L_{\text{sd}} = 18R_e, L_z = 100R_e\} \rightarrow 11.5\times$ speedup

Recommended: $L_{\text{sd}} = 18R_e$, $L_z = 78R_e$

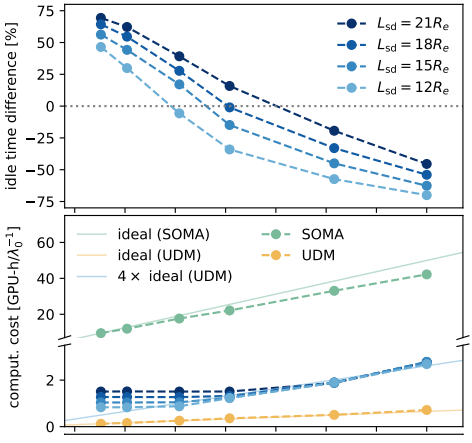


Figure: Load balancing SOMA and UDM subdomains

JÜLICH Forschungszentrum

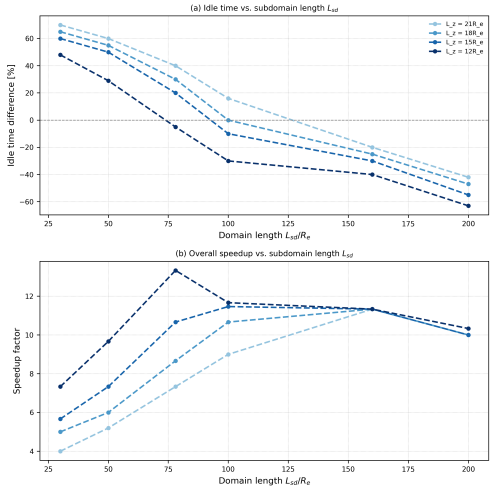
PERFORMANCE RESULTS

Coupled (4 A100) vs Baseline (8 A100)

Metric	Baseline	Coupled
Speedup	1×	13×
Energy/step	707.5 J	28.9 J
Reduction	-	-96%
Steps/kJ	1×	24.5×

Why It Works

- UDM intrinsically cheap
- SOMA only on subdomain
- Optimal sizing $\rightarrow$ zero idle
- Fewer GPUs, each productive



JÜLICH Forschungszentrum

SCIENTIFIC FIDELITY: TIME EVOLUTION

NIPS Simulation

Domain:
$19.4R_e \times 22.4R_e \times 96R_e$
Grid: $194 \times 224 \times 960$
Time: $T = 96\,\lambda_0^{-1}$

Observations

- Phase separation
- Structure formation
- Dynamics evolution

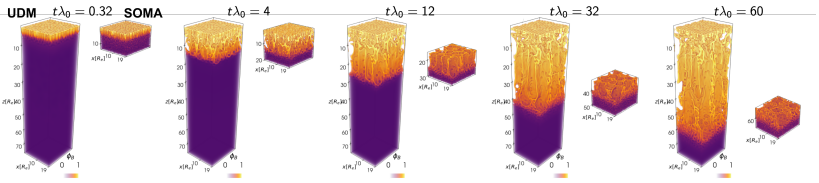


Figure: Evolution of coupled multi-scale solution

FIDELITY: QUANTITATIVE VALIDATION

Three Simulations

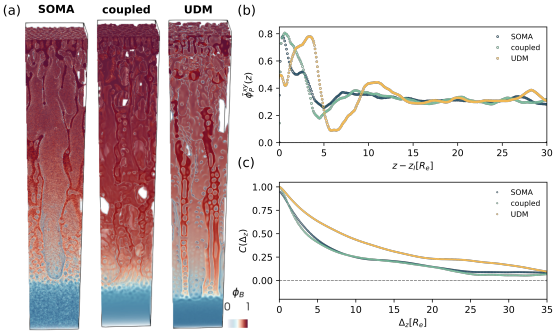
- SOMA ($L_z = 200R_e$): reference
- Coupled ($L_z = 96R_e$): hybrid
- UDM ($L_z = 96R_e$): continuum

Metrics Analyzed

- Laterally-averaged density
- Vertical correlation
- Macro-pore persistence

Results

- Coupled: matches SOMA
- UDM: significant deviations



(a) 3D · (b) Density · (c) Correlation

LESSONS 1 & 2:

Speedup != Less Energy From 8 GPU $\rightarrow$ 16 GPU
Time: -43% (better) while Energy/step: +72% (worse) as efficiency drops
Energy reduction requires reducing total work, not just parallelizing it.

Accuracy-Energy Trade-off: toward Cost-benefit model
Invoke SOMA only if accuracy gain justifies energy cost $\rightarrow$ Energy-aware multi-fidelity modeling

LESSON 3: REPORT THREE METRICS

Why three metrics are essential:
Speedup alone: hides energy cost
Energy alone: obscures productivity

The Three Metrics

1. Speedup (wall-clock time)
2. Energy/step (absolute cost)
3. Steps/kJ (efficiency)

	Speedup	Energy/step	Steps/kJ
SOMA scaling	+45%	+72%	-42%
Coupled	13×	-96%	+24.5×

SOMA scaling: $\rightarrow$ inefficient
Coupling: all improve $\rightarrow$ superior

LESSON 4: GENERALIZATION TO OTHER WORKFLOWS

The coordinator pattern applies beyond polymer physics

- **Climate:** coarse global + fine storm
- **Chemistry:** MD + QM/MM around bond
- **CFD:** RANS + LES near component

Requirements

- Well-defined APIs
- Clear data contracts
- MPI inter-communicator
- Resource control

CONCLUSIONS

- Coupled simulation delivers 13× speedup and 96% energy reduction
- GPU kernel optimization yields significant but architecture-dependent gains
- A coordinator is essential for efficient coupling and fine-grained resource control
- ML-driven decisions enables error prediction in our case.

Two Takeaways

1. **GPU Optimization** — Software optimization to fully exploit hardware is essential, but it is a never-ending loop.
2. **Less Is More** — Doing *less* work in the first place, i.e., Sustainable exascale via algorithmic co-design

Thank You

Dabah, Häfner, Happ, Pickartz, Müller, Herten

ISC-HPC 2026

Questions?