MeanField Single-Core and Distributed Performance Optimization Report

Implementation based on repository checkpoint `25510008ddf9810eaf9821149734d526c76b087a`

2 September 2026

Abstract

This report describes a performance-focused revision of the MeanField code base. The work starts from checkpoint `25510008ddf9810eaf9821149734d526c76b087a` and treats the existing code as authoritative whenever it differs from the earlier optimization audit. The changes add bounded and MPI-aware profiling tools, remove avoidable allocation and mapping work, introduce a primal-only gravity preparation path, enable release interprocedural optimization, and correct true-degree-of-freedom versus local-degree-of-freedom handling in MFEM partial-assembly operators. The preconditioner was deliberately left unchanged.

The unchanged gravity virial consistency test now takes approximately 205.84 seconds on one MPI rank, 118.93 seconds on two ranks, and 65.21 seconds on four ranks. The centrifugal h-refinement test fell from a historical 525.60 seconds to 254.98 seconds. The complete release suite passes all 410 test cases and 5,086,850 assertions without changing a tolerance, resolution, or existing user-facing call signature.

Contents

1 Scope and invariants

The optimization audit was used as technical input, not as an instruction source. Repository behavior and the user's requirements governed every decision. In particular, this work preserved the following invariants:

- numerical tolerances and spatial or quadrature resolutions were not weakened;
- existing user-facing entry points and call signatures were preserved;
- compile-time physical structure and typed field/domain contracts were retained;
- error conditions remain explicit rather than silently falling back;
- the preconditioner algorithm was not modified, because it is reserved for a subsequent optimization effort;
- `./format` was run immediately before every compilation; and
- all unrelated files and pre-existing working-tree changes were left intact.

Several low-level prepared-operator classes received additive inspection and primal-preparation methods. These additions do not alter existing calls or semantics; they expose the distinction between primal data and derivative data to internal orchestration and tests.

2 Measurement methodology

2.1 Two build modes

Two separate release-style configurations were used so instrumentation could not be confused with production performance:

Optimized release. Profiling is compiled out and interprocedural optimization (IPO/LTO) is enabled. This build supplies the authoritative wall-time measurements.

Profiling release. Profiling is enabled and IPO is disabled. This build identifies proportions and call counts. Its absolute wall time is not compared directly with the optimized release build.

The CMake options are `MEAN_FIELD_ENABLE_PROFILING` (default `OFF`) and `MEAN_FIELD_ENABLE_IPO` (default `ON`). CMake checks IPO support and fails with an explicit diagnostic when it is requested but not supported.

2.2 Bounded runner

The new `tools/profile_tests.py` runner executes Catch2 filters on one or more MPI rank counts. Each run has:

- an independent combined stdout/stderr log;
- a wall-clock timeout;
- process-group termination, followed by forced termination after a configurable grace period;
- one durable CSV row written immediately after each repetition; and
- an exact JSON representation of the command that was measured.

Starting each child in a new process session is important for MPI: a timeout can terminate the launcher and all of its workers rather than leaving orphaned ranks. Both successful smoke runs and an intentionally forced timeout were used to verify cleanup. Runner options may appear on either side of the test filter; additional Catch2 arguments are placed after `--`.

2.3 In-process instrumentation

The profiling registry records named regions, warmup observations, work-unit counters, minimum/-maximum durations, and total duration. Call-site-static `Region` objects cache registration indices so the timed path does not repeatedly allocate or look up a string. The registry is mutex protected for well-defined use from multiple host threads.

When profiling is disabled, all scope timers and counters expand to no-ops at preprocessing time. Labels are not evaluated, registry locks are not taken, and no timer object is constructed. The production build therefore does not pay a runtime branch for disabled instrumentation.

MPI aggregation first exchanges and unions the complete set of labels. It then performs every reduction in the same globally sorted label order. This matters when a region is visited on only a subset of ranks: iterating each rank's local map directly would issue different collective sequences and can deadlock. A dedicated MPI test deliberately registers one rank-local label per process and verifies that aggregation terminates and reports the expected sample ranges.

3 Implemented optimizations

3.1 Identity field maps

`FieldDofMap` now has direct paths for identity maps in `gather`, `scatter`, `scatter_into`, and `scatter_add`. The common all-domain gravity fields no longer traverse an index indirection one scalar at a time. The optimized operations become an MFEM vector assignment or scaled add, while the general supported-domain path is unchanged.

This change is intentionally located at the mapping abstraction rather than at individual physics call sites. Every consumer benefits, and the compile-time field/domain descriptions still determine whether a map is an identity.

3.2 Primal-only gravity preparation

The reduced gravity solve needs a mapped mass action and mapped source action, but it does not apply derivatives with respect to displacement. Previously, construction of every reduced solve also built derivative-only element data: displacement DOF tables, inverse element Jacobians, and mass-variation caches.

The geometry context now distinguishes two preparation modes:

Primal preparation builds only data required by residual/operator application.

Linearization preparation additionally builds all displacement variation data.

The reduced operator requests primal preparation. A later request for a linearization at the same discretization and displacement revisions upgrades the context in place rather than incorrectly treating it as already complete. Derivative entry points explicitly reject use after primal-only preparation. Tests cover initial primal preparation, no-op re-preparation, same-revision upgrade, and the availability of derivative data after upgrade.

This moves work from an unconditional runtime setup path to the path that actually requests the corresponding mathematical operation. It also reduces peak retained data for primal-only solves.

3.3 Prepared gravity-source workspaces

The mapped gravity source operator previously created several MFEM vectors on every application and on every transpose application. Density/potential local vectors, element inputs, quadrature actions, element actions, and local output vectors are now persistent workspaces owned by the prepared operator. `SetSize` can therefore reuse allocated capacity across calls.

Identity potential maps write directly into the caller's output vector instead of first producing a full true-DOF vector and gathering it back. Primal-only preparation also omits displacement transformations and inverse-Jacobian tables that are used solely by derivative actions.

3.4 Gravity operator workspace and hot branches

`GravityFieldOperator` now owns reusable vectors for potential, gradient, divergence, and transpose-divergence actions. When both gravity field maps are identities—the dominant production case—the operator passes the existing views directly to the mass and divergence operators. The general reduced-map path remains available.

The identity and one-rank paths use C++ branch-likelihood annotations. This does not change semantics, but it gives the optimizer useful layout guidance: the common straight-line path is kept contiguous and exceptional mapping work is placed off the hot path.

3.5 MFEM true/local DOF correctness and MPI performance

The largest distributed correctness defect was a contract mismatch. MFEM `ParBilinearForm` and `ParMixedBilinearForm` partial-assembly `Mult` operations consume rank-local DOF vectors. The production code was supplying true-DOF vectors. Those sizes and layouts coincide on one rank, which hid the defect; with multiple ranks, the mapped H(div) mass action became nonlinear and nonsymmetric, the divergence transpose identity failed, and MINRES did not converge.

The mapped H(div) mass operator now performs, for multiple ranks:

1. one true-to-local prolongation;
2. the stellar and vacuum partial-assembly actions on local vectors;
3. local addition of the two domain contributions; and
4. one local-to-true transpose-prolongation.

Persistent local buffers avoid allocating on each call. A communicator-size test controls the fast path. It is deliberately not based on a rank-local comparison of vector sizes: such a condition can differ across ranks and send processes through different collective sequences.

The divergence form is wrapped by a final `TrueDofParMixedBilinearFormOperator`. Forward and transpose operations perform the corresponding trial/test true-to-local and local-to-true transfers, validate dimensions, and require congruent communicators. The one-rank path calls the local form directly.

An attempted fusion of the stellar and vacuum H(div) partial-assembly forms was rejected. Although its operator action matched, 35,856 diagonal assertions failed because MFEM's marker-specific diagonal assembly masks accumulated entries after each integrator. The separate forms were retained; preserving the diagonal contract is more important than a speculative reduction in one operator call.

3.6 Analysis and virial traversals

Center-of-mass, quadrupole, and mesh-volume traversals now reuse mapping context and scratch objects rather than reconstructing them at every quadrature point. Mass and first moments are packed into one MPI reduction. The virial helper similarly packs six reductions into three arrays: energy sums, status counts, and determinant extrema. This lowers synchronization overhead without changing the mathematical reductions.

The profiler also records mapping-evaluation counts. In the principal virial test, center of mass, quadrupole, and energy integration each perform 1,605,632 mapping evaluations over the seven deformation amplitudes. The counts make it possible to distinguish a faster kernel from an accidentally skipped traversal.

3.7 Centrifugal mapping evaluation

The centrifugal residual and Jacobian used to request quadrature geometry and physical position through separate mapping calls at the same point. A single `EvaluateVolume` now provides both the quadrature weight and physical position, with an explicit invalid-mapping error. Scratch contexts are reused through the element loop.

Independent test/reference traversals were updated in the same manner: each quadrature point is mapped once, and nodal positions use one reusable point context. This does not weaken the

reference calculation; it removes duplicate geometry work from the test itself. The stable 2.06-fold reduction in the h-refinement test wall time confirms that this was a distinct mapping hot path.

3.8 Interprocedural optimization

IPO/LTO is enabled for the release versions of the main library, test support, tests, MPI tests, and experiment executables. Applying IPO only to the static library is insufficient: call sites in the final executables must participate for cross-module inlining, devirtualization where provable, dead-code removal, and scalar replacement to cross archive boundaries.

Direct initialization is retained around profiled factory expressions so mandatory copy elision remains available. The profiling macro used for expressions returns `decltype(auto)`; when profiling is disabled, it collapses directly to the wrapped expression.

4 MPI regression suite

A separate `mpi_tests` executable avoids running hundreds of unrelated serial tests independently on every rank. CTest registers two jobs, `mpi_2_ranks` and `mpi_4_ranks`, under the `mpi;distributed` labels. The jobs share a resource lock, advertise their processor counts, and have a 180-second test timeout. The `check_mpi` target builds the MPI executable and runs this label set.

The suite verifies:

- world and split-communicator membership;
- exact global mesh-partition coverage and nonempty local partitions;
- mapped H(div) mass linearity and symmetry;
- divergence/transpose-divergence adjointness;
- finite distributed center-of-mass and quadrupole analysis;
- a complete gravity solve at relative tolerance 10^{-12}, absolute tolerance 10^{-15}, and at most 1000 iterations; and
- profiling aggregation when ranks have different local region labels.

The ordinary Catch2 runner now broadcasts rank zero's random seed after MPI initialization. Before this change, ranks could randomly execute tests in different orders and enter incompatible collectives. Only rank zero emits the custom progress report, while all ranks still execute every assertion.

The measurements in this report used multiple MPI processes on one physical machine. They validate distributed ownership and communication semantics and the same suite can run under a multi-node launcher, but inter-host fabric and scheduler behavior were not measurable on the available host. Therefore the report calls these "multi-rank" results rather than claiming a physical multi-node hardware validation.

Measurement	Before (s)	After (s)	Result
Prepared spectral-mode Jacobian application	~ 3.98	3.965	maintained
Gravity virial consistency, one rank	215.93	205.84	1.049×
Centrifugal virial h refinement	525.60	254.98	2.061×

Table 1: Representative optimized-release wall times. The approximately 23-second Jacobian time belongs to the earlier pre-checkpoint phase; checkpoint `25510008` had already reached approximately 3.98 seconds.

5 Performance results

5.1 Release wall times

The new work did not regress the previously optimized Jacobian path: the first prepared application measured 3.964707 seconds. The original loose one-second goal remains unmet, but the present work intentionally focused on other paths.

5.2 MPI scaling

MPI ranks	Virial wall (s)	Speedup	Parallel efficiency
1	205.843	1.00×	100.0%
2	118.926	1.73×	86.5%
4	65.210	3.16×	78.9%

Table 2: Unchanged gravity virial test in the IPO-enabled release build.

Two focused mapped-H(div) measurements show similar scaling:

Test	1 rank (s)	2 ranks (s)	4 ranks (s)
Operator identities	3.461	1.907	1.115
Prepared versus stateless kernel	9.516	5.724	2.841

Table 3: Focused distributed mapped-H(div) regressions.

The registered release MPI suite passed in 18.12 seconds on two ranks and 10.11 seconds on four ranks.

5.3 Full regression result

The full IPO-enabled release suite completed in 2,807.831 seconds:

Test cases	410 passed, 0 failed
Assertions	5,086,850 passed, 0 failed

The original baseline contained 406 test cases and 5,086,800 assertions. The additional assertions cover profiling behavior and the primal/linearization preparation distinction. MPI cases are exercised separately at two and four ranks.

6 Why the virial test remains long

The virial test is not dominated by the coupled spectral-mode Jacobian. It constructs and solves seven reduced gravity systems. The final profiling build reported:

Region	Calls	Mean per call	Total
Gravity solve	7	27.502 s	192.512 s
MINRES	7	24.625 s	172.375 s
Reduced gravity operator `Mult`	4,617	20.651 ms	95.344 s
Mapped H(div) mass `Mult`	4,617	15.279 ms	70.543 s
Reduced-operator construction	7	2.277 s	15.937 s
H(div) primal preparation	7	1.472 s	10.303 s
Source primal preparation	7	0.800 s	5.599 s
Preconditioner construction	7	0.583 s	4.080 s
Center of mass	7	0.471 s	3.294 s
Quadrupole	7	0.474 s	3.316 s
Virial energy integration	7	0.975 s	6.823 s

MINRES performed 4,611 iterations across the seven systems. Consequently, a 20.65-millisecond matrix-free application becomes roughly 95 seconds of work, and the H(div) mass action alone is about 74% of each reduced operator call. The remaining MINRES time includes the deliberately unchanged preconditioner, vector operations, reductions, and Krylov recurrence bookkeeping.

This explains the apparent disconnect between the approximately four-second coupled Jacobian application and the approximately 206-second virial test: they exercise different operators with radically different call counts. A future preconditioner improvement that materially reduces iteration count has much greater leverage on the virial test than another small reduction in analysis integration time. That work was explicitly excluded from this revision.

7 Known distributed boundary

The lower-level gravity analysis and solve now operate correctly and scale on multiple MPI ranks. The complete spectral equilibrium experiment still cannot start on more than one rank. It stops during problem construction with the existing explicit error:

```
Logical radial surface interpolation currently supports one MPI rank; shared
interpolation-row ownership remains deferred.
```

This is not a preconditioner or gravity-kernel failure. Supporting the path requires a distributed ownership design for projected radial-surface interpolation rows, including rows whose supporting boundary element and basis DOFs live on another rank. A safe implementation needs a global surface-DOF identity, deterministic ownership at partition interfaces, distributed row construction, and transpose-consistent communication. That is an architectural extension, so it was not improvised in a performance patch under the stated “no architectural rewrite” constraint.

8 Recommended next work

The measurements suggest the following order for subsequent work:

1. Improve the reduced-gravity preconditioner and measure iteration-count reduction. This is the highest-leverage virial optimization and is intentionally reserved for the next agent.

2. Profile the divergence and transpose-divergence portions of the remaining 5.37 milliseconds per reduced application. Determine whether true/local transfers can be safely hoisted or combined across the two block actions.

3. If a custom mapped-H(div) kernel is justified, fuse the stellar and vacuum domain passes while reproducing MFEM's marker-specific diagonal semantics exactly. The failed generic-form fusion demonstrates that operator equivalence alone is insufficient.

4. Design distributed radial-surface interpolation ownership as a separate feature. Add forward/-transpose virtual-work tests before enabling the multi-rank spectral experiment.

5. Run the existing rank sweep on actual multi-node hardware, ideally with CPU affinity and at least three repetitions, to separate network scaling from same-host memory-bandwidth effects.

9 Reproduction commands

The practical command guide is also recorded in `docs/profiling_and_mpi.md`. The essential sequence is:

```
# Profiling build: format immediately before configure/build.
./format
cmake -S . -B cmake-build-profile-homebrew-llvm -G Ninja \
  -DCMAKE_BUILD_TYPE=Release \
  -DMEAN_FIELD_ENABLE_PROFILING=ON \
  -DMEAN_FIELD_ENABLE_IPO=OFF
./format
cmake --build cmake-build-profile-homebrew-llvm --target tests mpi_tests -j 6

# Bounded serial and rank-sweep measurements.
python3 tools/profile_tests.py \
  --executable cmake-build-profile-homebrew-llvm/tests \
  --ranks 1 --timeout 360 \
  "Gravity Field Virial Consistency Across Volume Preserving Deformation" \
  -- --reporter console

python3 tools/profile_tests.py \
  --executable cmake-build-release-homebrew/tests \
  --ranks 1,2,4 --timeout 360 \
  "Gravity Field Virial Consistency Across Volume Preserving Deformation"

# Registered MPI regressions.
ctest --test-dir cmake-build-release-homebrew \
  --label-regex mpi --output-on-failure --timeout 180
```

10 Conclusion

This revision improves a separate centrifugal hot path by roughly a factor of two, modestly improves single-rank virial time, and—more importantly—turns the distributed reduced-gravity path from

numerically invalid into a tested, scaling implementation. Profiling is now reproducible, bounded, and safe for rank-local label sets, while compiling to zero instrumentation in production. The remaining virial cost is well localized: thousands of Krylov iterations repeatedly invoke a mass-dominated reduced operator. That evidence provides a clear handoff to the planned preconditioner work without weakening any physical or numerical invariant.