Accelerating wind energy simulations by reducing communication costs

Mukul Dave
mhdave@lbl.gov

NESAP for Simulation Postdoc,
National Energy Research Scientific Computing Center (NERSC),
Lawrence Berkeley National Laboratory

NUG Community Call – June 20, 2024
Wind farm simulations need atmospheric data as initial and boundary conditions

The physics important for the turbines have to be captured correctly in the atmospheric models while factoring in terrain/offshore conditions.

Energy Research and Forecasting (ERF)
• is a modern C++ based, GPU-enabled alternative to the Weather Research and Forecasting (WRF) model.
• prioritizes accurate prediction of low-level winds.
• provides models for offshore and complex terrain environments.

The **AMReX** framework provides
• block-structured adaptive mesh refinement.
• advanced data structures and memory management.
• interface for parallelism over CPUs and GPUs with efficient data transfer and load balancing.
How can we run the simulations faster?

- Profiling to find bottleneck ↔ detailed optimization.
  What if communication cost >> computational bottleneck?

- Systemic changes to data structures or memory management.
- Determining optimal run time settings: number of procs, affinity…

Run time settings or systemic changes can provide a significant boost when communication costs become the main bottleneck.
Reducing communication costs: a preview

➢ OpenMP reduces communication cost within a node.

➢ GPU-aware MPI reduces communication cost between GPUs.

➢ A separate memory pool for communication buffers improves performance.

Directed and guided by:
Ann Almgren, Don Willcox, Weiqun Zhang, Aaron Lattanzi
Center for Computational Sciences and Engineering (CCSE), AMCR Division, Berkeley Lab
AMReX employs OpenMP multithreading to reduce intra-node communication costs

**MPI+X strategy for parallelism**

- Message passing interface (MPI) explicitly transfers data among distributed processes that provide coarse-grained parallelism over **grids**.
- **X = OpenMP** implements multiple threads on CPUs/cores within a node that compute **tiles** in parallel.
- Multiple threads can access the same memory space on a node (shared memory parallelism).
Running with 4-8 threads per process provides up to a 33% reduction in wall time

Atmospheric boundary layer (ABL) simulations
- 100 time steps, I/O and diagnostic calculations are turned off.
- Tested over **2 nodes (256 physical cores)** on NERSC’s Perlmutter system. The problem size and number of cores are kept constant while varying the balance of MPI processes and OMP threads per process.

![Graph showing Elapsed Time (s) vs. OMP threads per process (256 / no. of MPI procs). The graph indicates that more threads per process incur high synchronization costs.](image)
GPUs provide a 30x speed up for ERF by running thousands of threads in parallel

**MPI+X strategy for parallelism**

- **X** = CUDA/HIP/SYCL for GPUs based on the vendor.
- Each MPI process on a CPU “host” assigns work to a single GPU.

\[ \log_{\text{gpu}}(i) = \text{hi}_{\text{gpu}}(i) \]
GPU-aware MPI can transfer data directly between GPUs, bypassing the host. This requires setting a specific process-GPU-NIC affinity.

No GPUDirect P2P

GPUDirect P2P

Enabling GPUdirect results in a reduction of more than 20% in the wall times

Weak scaling of the ABL application on Perlmutter

- The domain size is $128 \times 128 \times 512$ for a single GPU;
- this is progressively scaled up to $2048 \times 1024 \times 512$ for 128 GPUs (over 32 nodes).
Communication buffers are used for efficient data transfer

Data aggregation reduces communication latency.

Slice of 3D data to be communicated is aggregated into a contiguous 1D buffer.
A distinct memory pool for comm buffers improves communication costs by 20% - 200%

Communication buffers sharing the same memory pool as data buffers degrades performance for a specific application.

Hypothesis: Comm buffers get assigned the same pointer in subsequent transfers, preventing the overhead of re-registering the address.
Summary: strategies to reduce communication costs

➢ Use of 4 - 8 OpenMP threads for shared memory parallelism over CPUs.

➢ Enabling direct data transfers between GPU - requires specific run time affinity settings.

➢ Implemented a distinct memory pool (arena) for communication buffers on the GPU.

These have a more significant impact than detailed profiling of the subroutines or kernels.
ERF is ready to be coupled with the wind turbine simulations

DOE Energy Earthshot Research Center

FLOWMAS: Floating Offshore Wind Modeling and Simulation