High-Performance Computing Benchmarking for Coastal Hydrodynamic Modelling using Delft3D Flexible Mesh

Parallel performance, scalability, and efficiency of D-Flow FM across MPI libraries (MPICH vs. Intel MPI), CPU core counts, and partition combinations — for 2D and 3D, barotropic and baroclinic configurations on an HPC.

Sharing is encouraged

HPC DETAILS

ABSTRACT

Abdulaziz Alabduljalil [1], Nada Alsulaiman [1], Yousef Alosairi [1], Tahani Hussain [2]

1 Environment & Life Sciences Research Centre, Kuwait Institute for Scientific Research, P.O. Box 24885, Safat 13109, Kuwait · 2 System & Software Development Department, Kuwait Institute for Scientific Research, P.O. Box 24885, Safat 13109, Kuwait

01 Abstract & Motivation

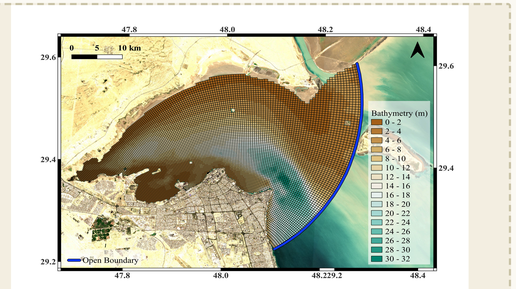
High-resolution coastal hydrodynamic models are constrained by available computing resources, forcing a trade-off between resolution and runtime.

High-Performance Computing (HPC) lifts this constraint: increasing simulation speed while preserving resolution. We benchmark resource configurations for a coastal model built with **Delft3D Flexible Mesh (D-Flow FM)**, focusing on parallel performance, scalability, and efficiency.

Experiments compare two MPI libraries — **MPICH and Intel MPI** — across multiple CPU core counts and partition combinations on 2D and 3D, barotropic and baroclinic configurations. Runtime depends strongly on configuration, MPI implementation, and partition layout.

Goal: Practical guidance on optimal HPC configurations, resource allocation, and MPI library selection for high-resolution coastal hydrodynamic models in operational and research settings.

02 Domain & Scale



LOCATION	**Kuwait Bay, Kuwait**
MODELLED AREA	**1,127 km²**
2D COMPUTATIONAL CELLS	**8,277**
3D COMPUTATIONAL CELLS	**83,700**
3D VERTICAL LAYERS	**10 sigma layers**
MEAN EQUIVALENT CELL SIZE	**369 m × 369 m**
SIMULATION DURATION	**14 days**
SOFTWARE	**Delft3D-FM+Apptainer** v2026.01
CLUSTER	**IKARUS HPC** Kuwait Institute for Scientific Research (KISR)

03 Experimental Setup

We benchmark four model configurations spanning the **2D / 3D** and **barotropic / baroclinic** axes, each run under both MPI libraries across a sweep of CPU core counts and partition layouts. Colour-codes are consistent across the poster.

MODEL	FORCING & ACTIVE FIELDS
2D Barotropic	**Tides · wind forcing**
2D Baroclinic	**Tides · temperature · salinity · wind · surface heat exhange**
3D Barotropic	**Tides · wind forcing**
3D Baroclinic	**Tides · temperature · salinity · wind · surface heat exhange**

MPI LIBRARIES COMPARED

MPICH
Reference open-source MPI implementation.

Intel MPI
Vendor-tuned MPI with Intel-fabric optimisations.

CPU	1, 10, 20, and 40 CPU cores / Node
PARTITIONS	1 to 160 partitions **(Nodes multiplied by CPU cores)**
RAM	50 GB / Node
HPC NODES	1 to 4 nodes

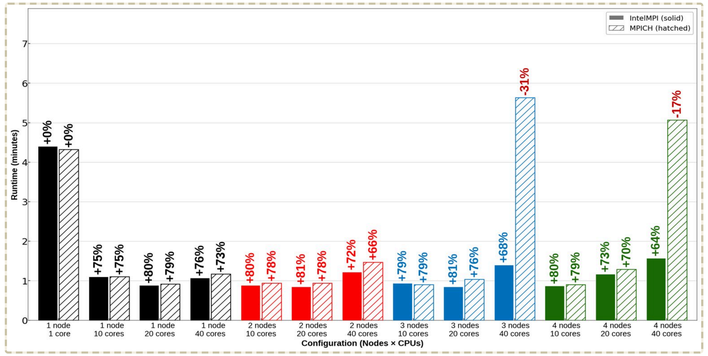
MPICH uses a Hydra launcher with a flat (non-hierarchical) collective algorithm. The algorithm uses an $O(N)^2$ message pattern, struggling with multi-node communication where additional nodes multiply communication costs, but shining in single-node configurations with shared memory.

IntelMPI is coupled with TCP communication fabrics that allows and forces inter-node communication over Infiniband. The communication overhead is thus reduced twice: first within node via shared memory, and second via TCP across node representatives.

04 Results

2D Barotropic

INTEL MPI MPICH

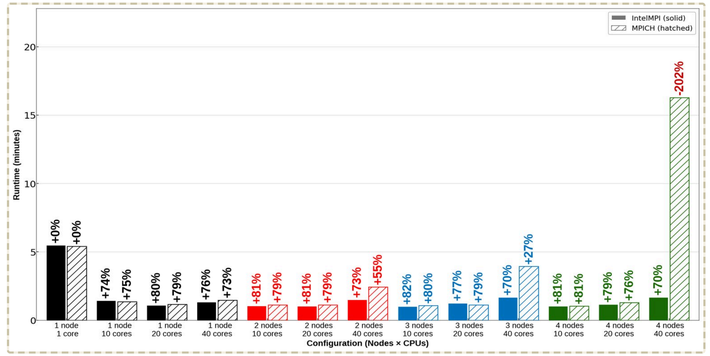


IntelMPI Best: 51 sec at **2 & 3 nodes × 20 CPU cores** · **81%** faster than serial (04:24 min).

MPICH Best: 54 sec at **3 & 4 nodes × 10 CPU cores** · **79%** faster than serial (04:19 min).

2D Baroclinic

INTEL MPI MPICH

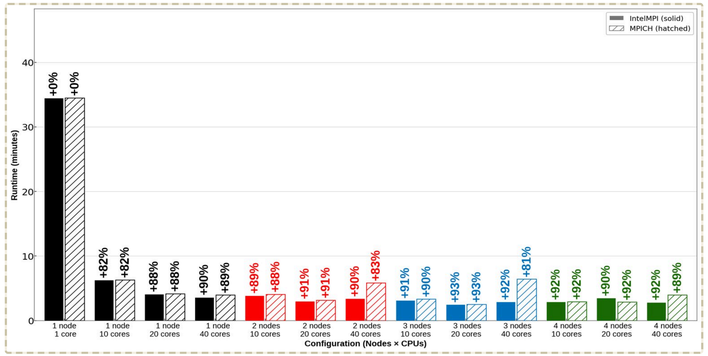


IntelMPI Best: 01:00 min at **3 nodes × 10 CPU cores** · **82%** faster than serial (05:29 min).

MPICH Best: 01:01 min at **4 nodes × 10 CPU cores** · **81%** faster than serial (05:24 min).

3D Barotropic

INTEL MPI MPICH

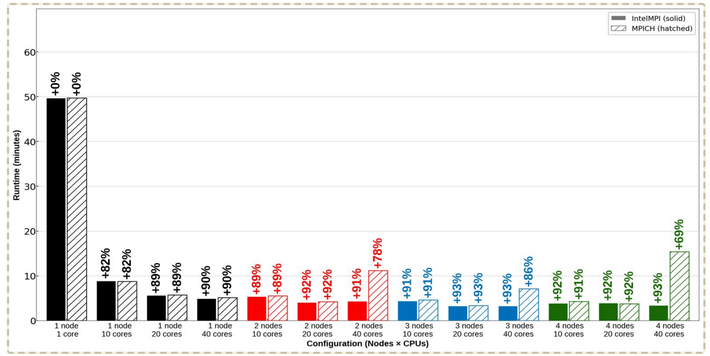


IntelMPI Best: 02:30 min at **2 nodes × 20 CPU cores** · **93%** faster than serial (34:29 min).

MPICH Best: 02:29 min at **3 nodes × 20 CPU cores** · **93%** faster than serial (34:30 min).

3D Baroclinic

INTEL MPI MPICH



IntelMPI Best: 03:14 min at **3 nodes × 40 CPU cores** · **93%** faster than serial (49:43 min).

MPICH Best: 03:21 min at **3 nodes × 20 CPU cores** · **93%** faster than serial (49:40 min).

05 Conclusions

01 Optimal configuration is model-dependent

Runtime depends jointly on dimensionality (2D vs. 3D), physics (barotropic vs. baroclinic), MPI library, and partition layout. **Intel MPI and MPICH share equal performance until 80 partitions. Intel MPI then is quicker.**

02 MPI library choice matters

MPICH and Intel MPI do not scale identically on the same hardware. The best-performing library shifts with model configuration, with MPICH favoring low node counts and Intel MPI performing better due to TCP layer (two-level tree reduction).

03 Partition count has a sweet spot

Beyond a configuration-specific threshold, additional partitions degrade wall-clock time as communication overhead dominates. The less nodes and more resources allocated, the better the outcome.

06 Recommendations · HPC ADMIN PERSPECTIVE

A Benchmark per workload, not once per cluster

Default partition counts and MPI defaults are not transferable across model setups. Provision a short benchmarking job before allocating a long campaign.

B Expose MPI-library choice to users

Create modules of both MPICH and Intel MPI; document the configurations where each wins so users can pick deliberately rather than by habit.

C Cap partitions to the measured sweet spot

Encourage right-sizing over over-allocation. In all node configurations, going over 20 CPU cores per node adds more communicaiton overhead than can be handled.

EMAIL aabduljalil@kisr.edu.kw

LINKEDIN linkedin.com/in/abdulaziz-alabduljalil

KISR EST. 1967

EGU '26 VIENNA