1st OpenFOAM HPC Challenge: I/O Format

Sergey Lesnik

sergey.lesnik@wikki-gmbh.de

20th OpenFOAM Workshop, Vienna, 01/07/2025

Current Parallel I/O

- Uncollated format (standard)
    - The number of files written, scales with number of cores N
    - Once the case is decomposed,
        - it is necessary to change N files to make a change (e.g. a BC)
        - it can only be restarted on the same number of cores
    - Substantial amount of time-to-solution and I/O load is consumed by mesh/field decomposition, reconstruction, redistribution
- Collated format
    - Reduces number of files but no solution for other issues mentioned above
- adiosFoam: a function object by Mark Olesen (ESI)
    - basically a wrapper for the current uncollated format, not fully integrated

Coherent Format

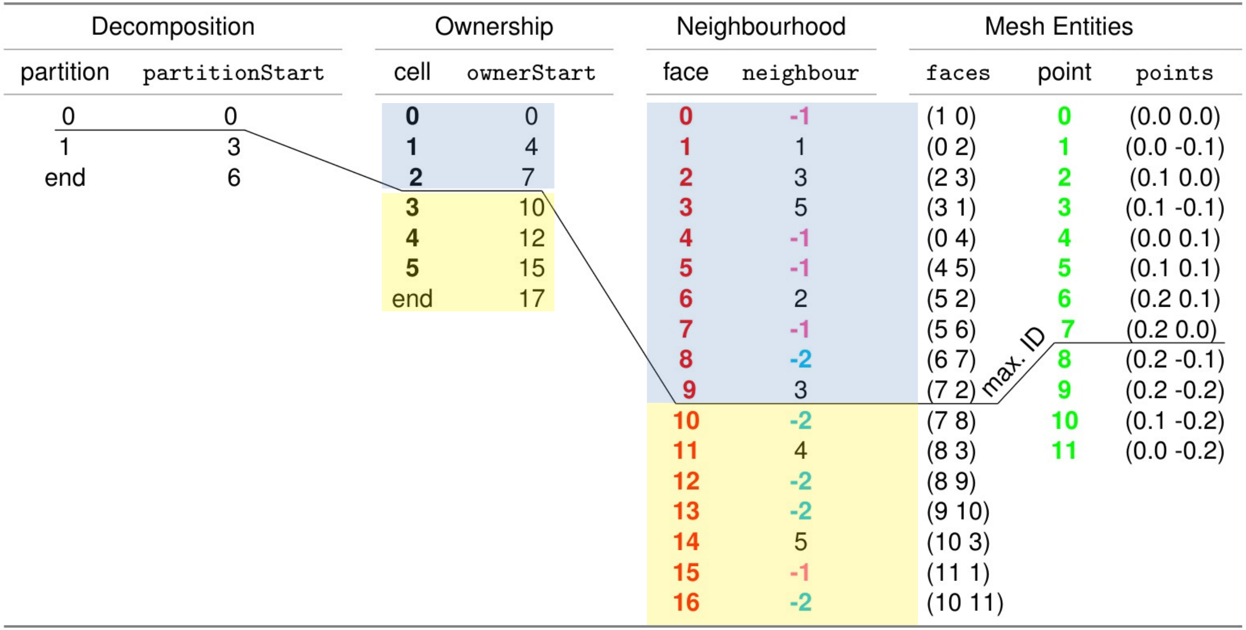
- Decrease time-to-solution
  - Start a parallel run without pre-processing on a desired number of cores (no decomposePar)
  - No reconstruction for post-processing (no reconstructPar)
  - Edit a single ASCII file to alter a BC on all processors
- Reduce number of files (i.e. inodes) substantially
- When viewing a slice in paraview, load only the data associated with this slice and do it fast
- Support lossy and lossless compression

Coherent Format Development

- Idea and architecture: Henrik Rusche
- Project: exaFOAM
- Original implementation in **foam-extend 4.1**
  - Done by
    - HLRS: Gregor Weiß, Andreas Ruopp, Flavio Galeazzo
    - Wikki: Sergey Lesnik, Henrik Rusche
  - training session at OFW 18 (Genoa) available online
- Integration into OpenFOAM **v2406, v2412**
  - done by HLRS, Wikki, ESI (Mark Olesen)
  - training session at OFW 19 (Beijing) available online
  - implementation available at https://gitlab.com/openfoam/community/exafoam/io

Coherent Format

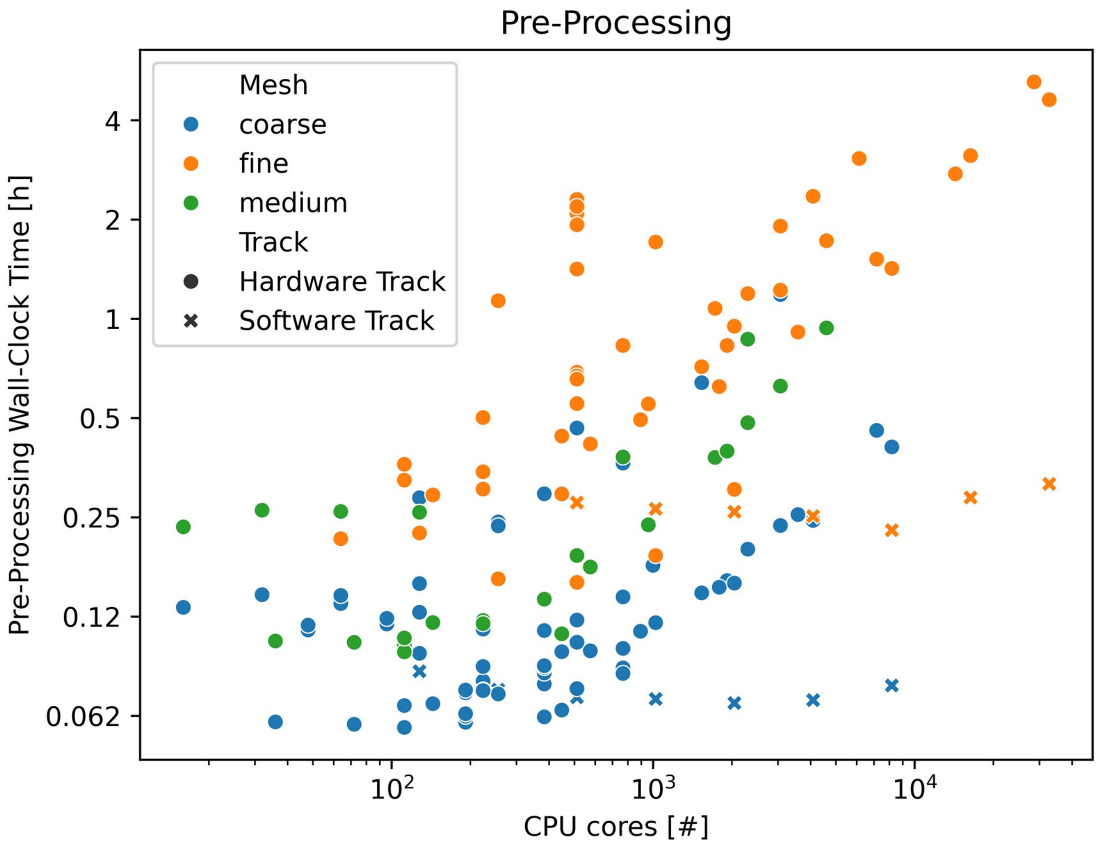
- Serial and parallel meshes are indistinguishable
- No duplication of data
- Sliceable structure
- Transport layer: ADIOS2



Decomposition		Ownership		Neighbourhood		Mesh Entities		
partition	partitionStart	cell	ownerStart	face	neighbour	faces	point	points
0	0	0	0	0	-1	(1 0)	0	(0.0 0.0)
1	3	1	4	1	1	(0 2)	1	(0.0 -0.1)
end	6	2	7	2	3	(2 3)	2	(0.1 0.0)
		3	10	3	5	(3 1)	3	(0.1 -0.1)
		4	12	4	-1	(0 4)	4	(0.0 0.1)
		5	15	5	-1	(4 5)	5	(0.1 0.1)
		end	17	6	2	(5 2)	6	(0.2 0.1)
				7	-1	(5 6)	7	(0.2 0.0)
				8	-2	(6 7)	8	(0.2 -0.1)
				9	3	(7 2)	9	(0.2 -0.2)
				10	-2	(7 8)	10	(0.1 -0.2)
				11	4	(8 3)	11	(0.0 -0.2)
				12	-2	(8 9)		
				13	-2	(9 10)		
				14	5	(10 3)		
				15	-1	(11 1)		
				16	-2	(10 11)		

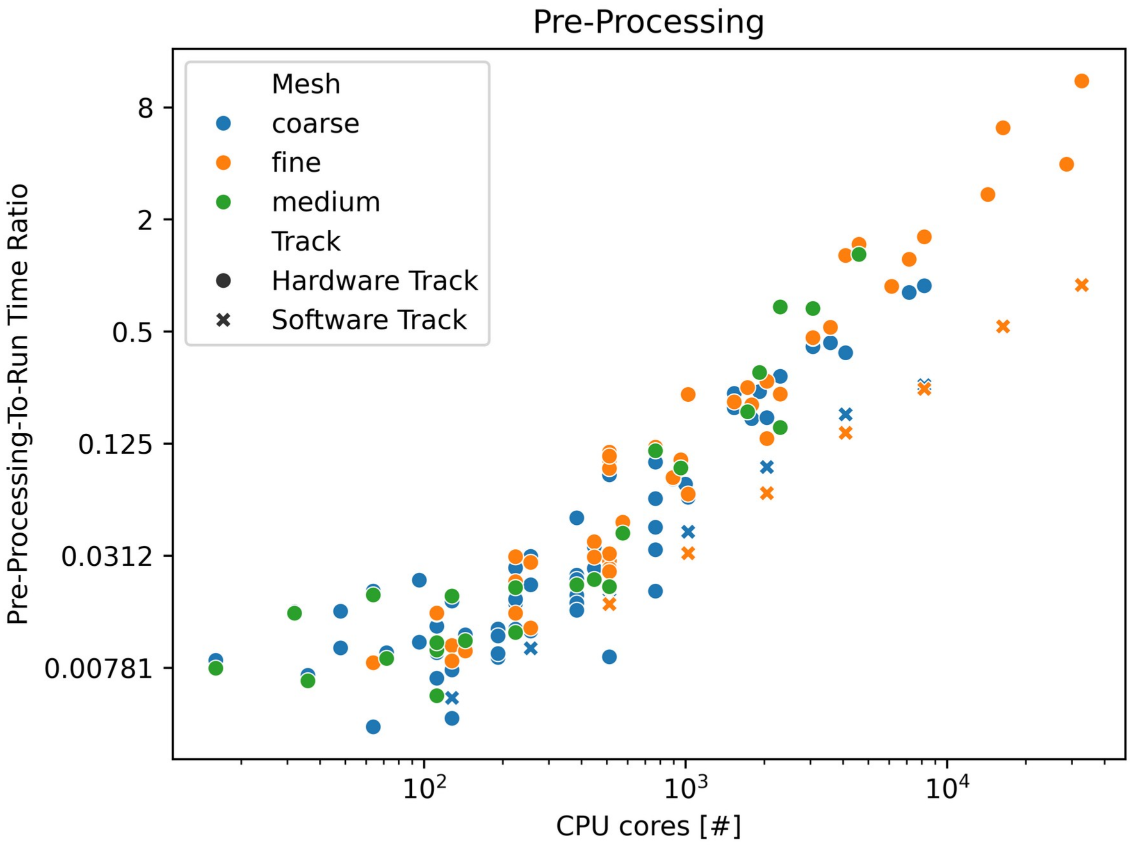
Pre-Processing

- Writing overhead during the solver run is low
- I/O is a large bottleneck during pre-processing
- Pre-processing includes
  - decomposePar (serial)
  - renumberMesh
  - potentialFoam
  - applyBoundaryLayer



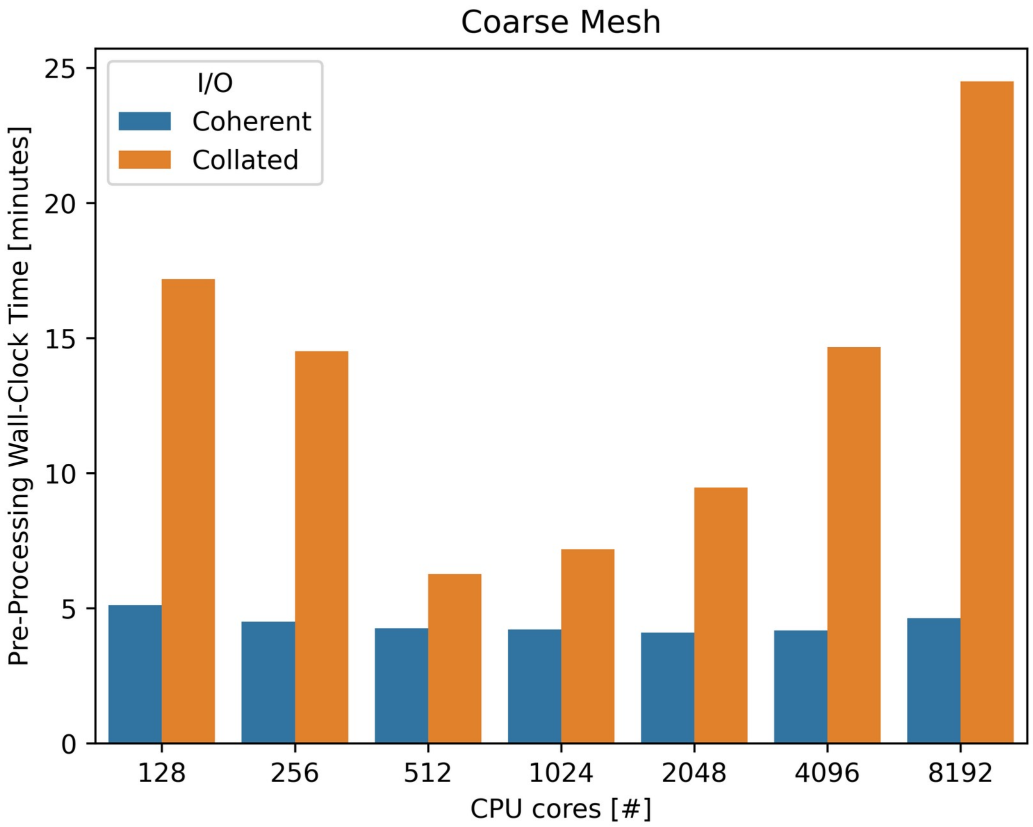
Pre-Processing vs Run

- What is the portion of pre-processing relative to the solver run?
- Ratio = pre-processing WCT / solver WCT
- Up to 8 times more time spent on pre-processing than solving the problem

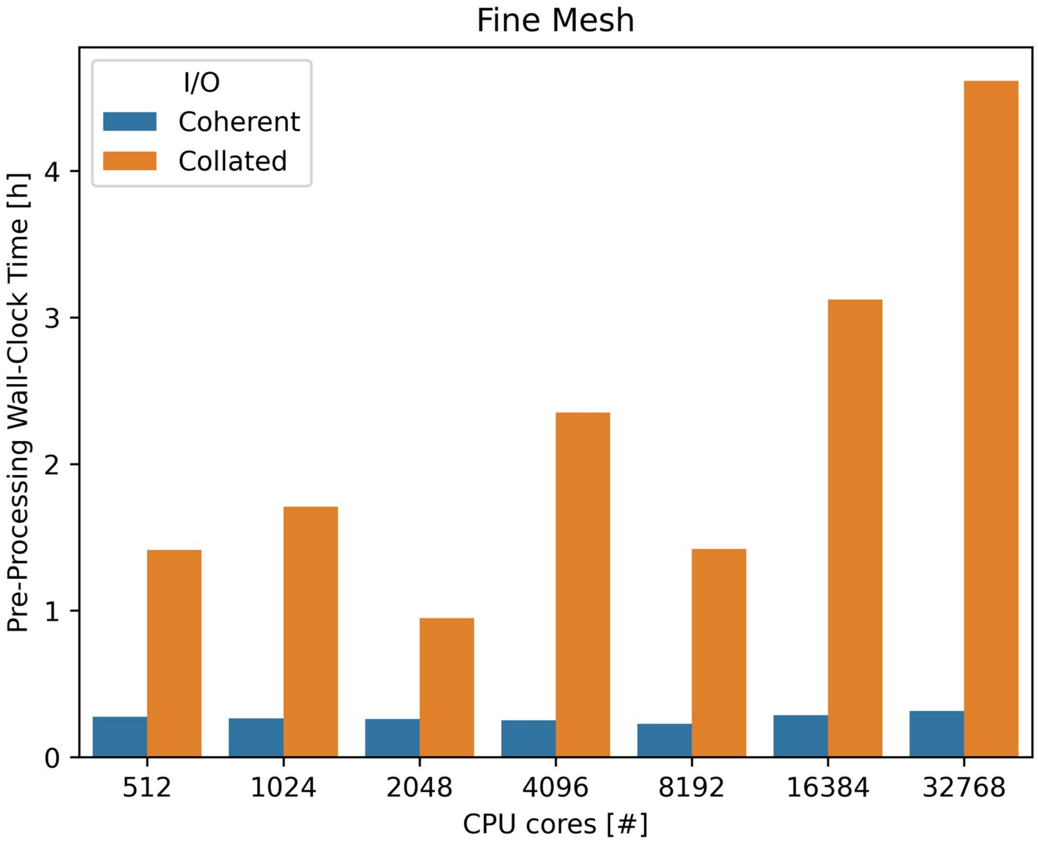


Coherent vs Collated Format

- LUMI Supercomputer
- File system: Lustre
- Collated is used with ioRanks option producing up to 32 processor directories

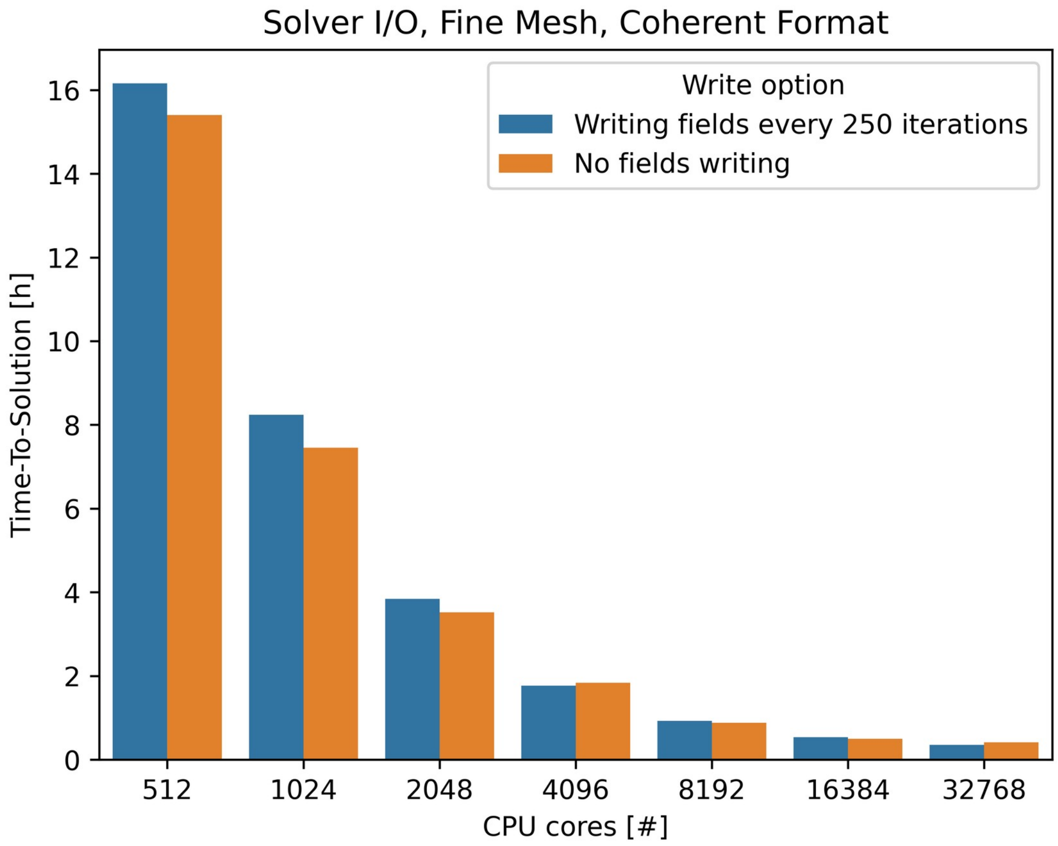


Coherent vs Collated Format



Solver Run

- Writing overhead during the solver run (simpleFoam) is lower than for the pre-processing
- Results for
  - Fine mesh
  - Coherent format
- Lower overhead for higher number of cores
  - Probably due to larger number of aggregators (one per node)



Conclusion

- The coherent format offers many benefits
  - No reconstruction needed
  - Substantially smaller number of files
  - Setup is easy to edit through ASCII files
- The coherent format is up to 20 times faster for the pre-processing than the collated format
- I/O overhead for the solver is lower than for the pre-processing
  - Prominent for lower number of cores
  - Not significant for higher number of cores

Thank you!

exaFOAM

EuroHPC
Joint Undertaking

SPONSORED BY THE

Federal Ministry
of Education
and Research