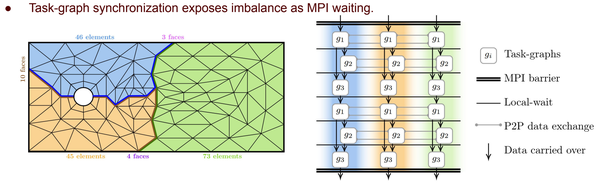
Online Load Balancing of High-Order Mixed-Element Simulations on Heterogeneous Architectures

Sambit Mishra, Semih Akkurt, Freddie Witherden

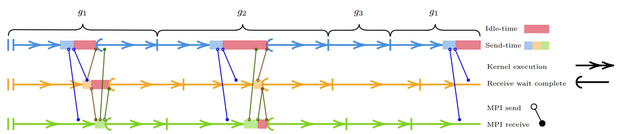
Concept

Why balance online?

- Scale resolving simulations performed on clusters can be expensive, spanning from hours to months.
- Efficient use of shared HPC resources involves allocations combining devices with different throughputs; this requires optimal load partitioning in heterogeneous architecture settings so no available resource stays idle.
- Rank performance depends on device, element type, memory behaviour, and communication.
- Static partitioners require these costs to be prescribed before execution.
- Task-graph synchronization exposes imbalance as MPI waiting.



A mesh partition and its task-graph execution between solution snapshots



Rank time is divided among computation, communication, and idle waiting

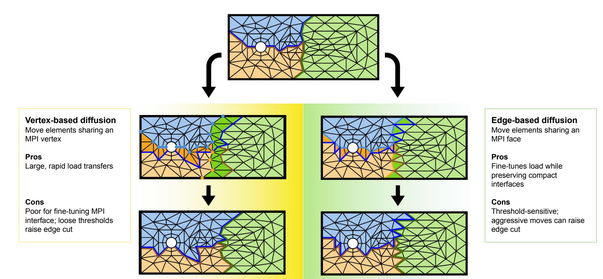


Effective rank cost is rank-local non-MPI work with the waiting that rank induces on its neighbours

Objective: Equalise effective rank cost across all ranks

Converge online toward a partitioning $\pi^{(m)}$ whose per-rank loads reflect their newly observed effective costs. At every load-balancing step, recompute each target load $\tilde{N}_i^{(m)}$ from fresh measurements of $C_i^{(m)}$, integer-round to obtain $N_i^{*,(m)}$, diffuse elements, and repeat until the partition and its performance stabilise.

$$\tilde{N}_i^{(m)} = N_{\text{tot}} \frac{N_i^{(m)}/C_i^{(m)}}{\sum_k N_k^{(m)}/C_k^{(m)}} \qquad \boxed{\pi^{(m)} \longrightarrow C_i^{(m)} \longrightarrow \tilde{N}_i^{(m)} \longrightarrow N_i^{*,(m)} \longrightarrow \pi^{(m+1)}} \quad \circlearrowleft \quad \pi^{best}$$

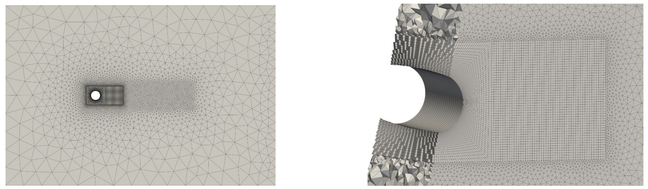


Load-balancing across ranks as a combination of vertex-based (left) and edge-based diffusion (right)

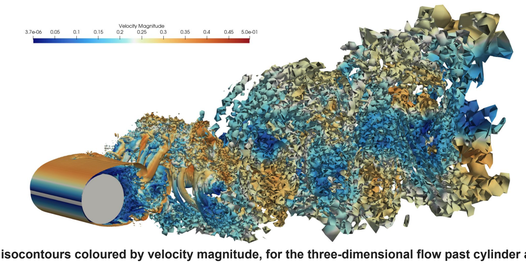
Test case setups and baselines

Setting up a scale-resolved 3D simulation across three clusters

- The C3900 case models turbulent flow past a circular cylinder at Re=3900, polynomial order 4.
- The simulation is performed on a mixed-element mesh with 352,594 elements (26M solution points).
- Each mesh partition is mapped to one MPI rank, and each rank is bound to an allocated CPU or GPU resource.



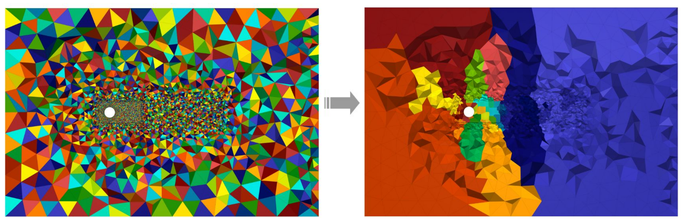
Projected (left) and a crinkle-cut (right) views of the 3D mixed unstructured mesh



Q-criterion isocontours coloured by velocity magnitude, for the three-dimensional flow past cylinder at Re = 3900

Setup	Cluster	Brief description of resources	Devices in the heterogeneous architecture	Isolated homogeneous architecture performance	Resource allocation	Heterogeneous run ideal target (sum-of-parts)
G4C44	Launch	Each node has 2 GPUs and a CPU	2× AMD Genoa 9654 (across 2 nodes)	4.5 GDoF/s (Best partitioning)	**48 ranks across 2 nodes** 4 GPU ranks and 44 CPU ranks Each GPU rank uses one GPU, is bound to 8 CPU cores Each CPU rank uses 8 CPU cores	**10.4 GDoF/s**
			4× NVIDIA A30 (across 2 nodes)	5.9 GDoF/s (Best partitioning)		
A2G3	Grace	First node has 2 A100 GPUs, second node has 3 A40 GPUs	2× NVIDIA A100 (in 1 node)	4.8 GDoF/s (Sum-of-parts)	**5 ranks across 2 nodes** 2 A100 GPU ranks and 3 A40 GPU ranks Each GPU rank uses one GPU, is bound to 1 CPU core	**6.4 GDoF/s**
			3× NVIDIA A40 (in 1 node)	1.6 GDoF/s (Sum-of-parts)		
G2H2	Vista	Each GH200 node has a GPU and a CPU	2× Hopper GPUs (across 2 GH200 nodes)	9.8 GDoF/s (Best partitioning)	**4 ranks across 2 nodes** 2 GPU ranks and 2 70-core CPU ranks Each GPU rank uses one GPU, is bound to 2 CPU cores Each CPU rank uses 70 CPU cores	**11.1 GDoF/s**
			2× 72-core Grace CPUs (across 2 GH200 nodes)	1.3 GDoF/s (Best partitioning)		

Hardware, MPI-rank allocation, and homogeneous reference performance for the three heterogeneous configurations

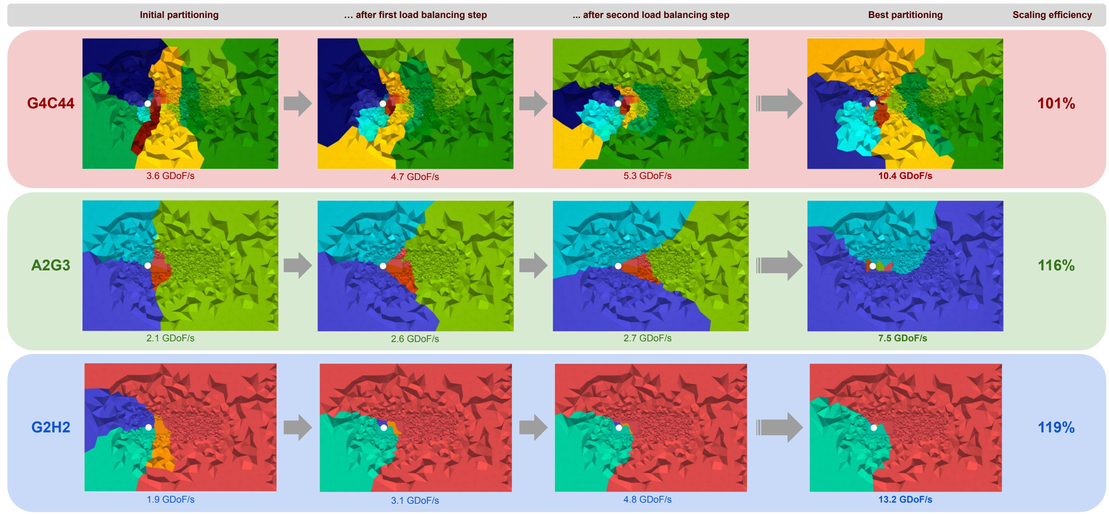


One island-removal step repartitions a 48-parts random partitioning to a contiguous partitioning

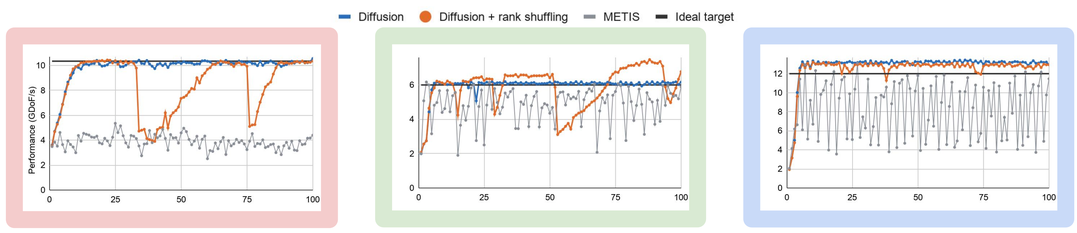
Results

Super-scaling observed across three diverse cluster architectures!

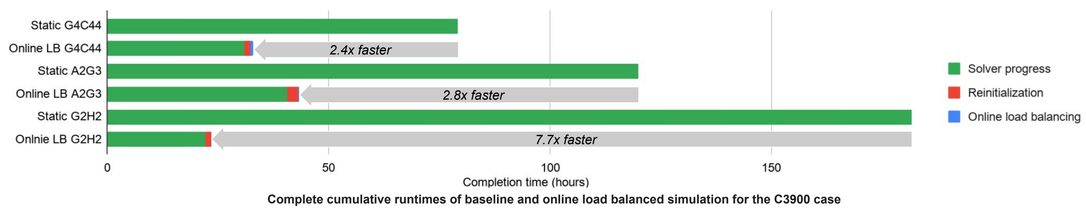
- Online load balancing was performed alongside simulation progress.
- To escape local optima, the partition of the worst-performing rank was intermittently removed and reintroduced.
- Supports adding and removing mesh partitions towards elastic rank counts; however the evaluated clusters use fixed-size allocations, so dynamic acquisition/release of MPI ranks was not exercised.



Online diffusion-based load balancing with rank shuffling enabled from an initial METIS partition, to the best observed partition



Partition performance over 100 online load-balancing steps using diffusion, diffusion with rank shuffling, and METIS



Complete cumulative runtimes of baseline and online load balanced simulation for the C3900 case

References

[1] Witherden et al., “PyFR v2.0.3,” Comput. Phys. Commun. 311, 109567 (2025); [2] Witherden, Vermeire & Vincent, “Heterogeneous computing with PyFR,” Comput. Fluids 120, 173–186 (2015); [3] Karypis & Kumar, “Multilevel partitioning of irregular graphs,” SIAM J. Sci. Comput. 20, 359–392 (1998); [4] Dzanic, Girimaji & Witherden, “PANS within a high-order flux-reconstruction framework,” J. Comput. Phys. 456, 110992 (2022).

Acknowledgements

Computations used the Launch cluster (NSF award 2232895), the Grace cluster at HPRC (Texas A&M University), and the Vista system at the TACC (UT Austin) under allocation OCE26006. SM and FDW were partially supported by AFOSR grant FA9550-23-1-0232, “Enabling next-generation heterogeneous computing for massively parallel high-order compressible CFD,” under the direction of Dr. Fariba Fahroo. SM was also partially supported by NSF award 2112356. SA was supported by EPSRC grant EP/W026686/1.