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

Sambit Mishra
Department of Ocean Engineering
Texas A&M University
College Station, TX, USA
sambit98@tamu.edu

Semih Akkurt
NVIDIA
London, UK
sakkurt@nvidia.com

Freddie Witherden
Department of Ocean Engineering
Texas A&M University
College Station, TX, USA
fdw@tamu.edu

***Abstract*—We present an online load balancing method for heterogeneous high-order flux reconstruction simulations on mixed unstructured meshes. The method infers cost imbalance from MPI wait-time statistics, derives target element counts, and relocates elements between neighbouring ranks using diffusion. Integrated into the open-source compressible Navier–Stokes solver PyFR, it is evaluated for flow past a circular cylinder at Reynolds number 3900. Online load balancing reaches 101%, 116%, and 119% of homogeneous sum-of-parts targets. Complete simulations achieve 2.4x, 2.8x, and 7.7x time-to-solution speedups on the three heterogeneous configurations.**

***Index Terms*—high-order methods, flux reconstruction, mixed unstructured meshes, heterogeneous computing, online load balancing**

I. Introduction

Scale-resolving simulations performed on clusters can span hours to months. Efficient use of shared HPC allocations combining devices with different throughputs requires optimal load partitioning so no available resource remains idle. For mixed-element discretizations, performance also depends on element type, memory behaviour, interfaces, and communication overlap. Static graph partitioners such as METIS require these costs before execution [1]. We instead infer cost during execution and couple it to incremental repartitioning.

II. Simulation test case

The C3900 test case is turbulent flow past a cylinder at $Re = 3900$, following the high-order setup of Dzanic et al. [2]. Its fourth-order mesh contains 196,594 tetrahedra, 6,000 pyramids, and 150,000 hexahedra: 352,594 elements and approximately 26 million solution points.

III. Diffusion-based repartitioning

An integer flow plan specifies element transfers between adjacent ranks. Vertex-based diffusion considers elements connected across an MPI vertex and enables large transfers. Edge-based diffusion uses face affinity to fine-tune the load while maintaining compact interfaces. Relocating interface-adjacent elements retains the existing partition structure (Fig. 1). We use π to denote a mesh partitioning. In particular, π_o^M is the METIS partitioning obtained with equal device and element-type weights, while π^{best} is the highest-performing partitioning observed during online load balancing.

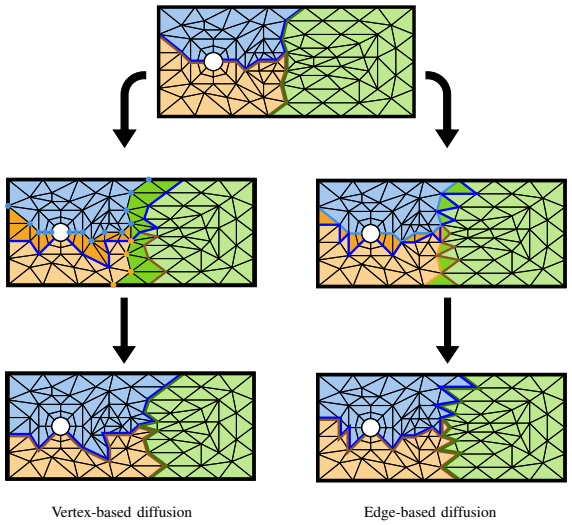


Fig. 1: Vertex- and edge-based diffusion across an MPI interface.

IV. Online load balancing

For an observed right-hand-side evaluation, t_i^{all} is rank i's total graph duration. MPI request-completion measurements provide outgoing-request wait s_i^{out}, incoming-data wait r_i^{in}, and neighbour waiting induced by rank i, denoted r_i^{out}. The non-MPI work and effective cost are

$$t_i^c = t_i^{\text{all}} - (r_i^{\text{in}} + s_i^{\text{out}}), \qquad C_i = t_i^c + r_i^{\text{out}}. \tag{1}$$

Local waiting is insufficient: a rank may wait little yet delay its neighbours.

Let N_i be the current element count and $N_{\text{tot}} = \sum_i N_i$. The real-valued target is

$$\widehat{N}_i = N_{\text{tot}} \frac{N_i/C_i}{\sum_k N_k/C_k}. \tag{2}$$

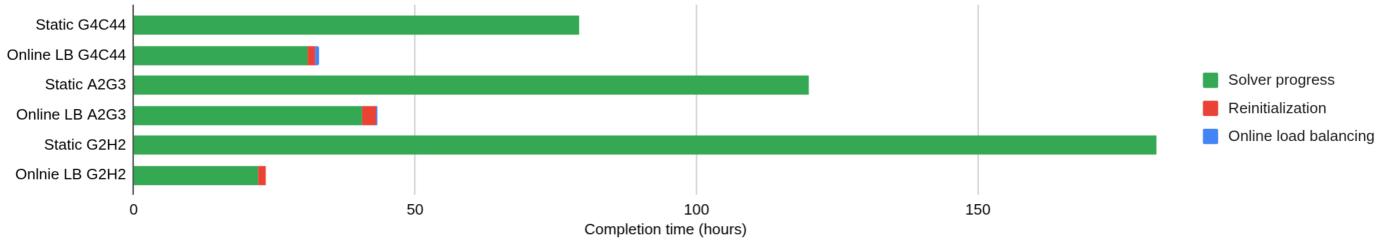


Fig. 3: Complete cumulative runtime of baseline and online load-balanced simulations for the C3900 case.

Targets are mapped to integers while preserving N_{tot}. Each step remeasures costs, recomputes $\widehat{N}_i$, and diffuses elements toward the new targets. The objective is to equalise effective costs so no rank systematically delays the others.

TABLE I: Cluster setups for the C3900 case.

Property	G4C44	A2G3	G2H2
Cluster	Launch	Grace	Vista
	2× AMD Genoa 9654	2× NVIDIA A100	2× NVIDIA Grace CPU
Devices	4× NVIDIA A30	3× NVIDIA A40	2× NVIDIA Hopper GPU
Ranks	48	5	4
Nodes	2	2	2

Rank shuffling combines partition removal, online balancing, and partition addition to escape local optima. Addition/removal is implemented, but dynamic MPI-rank acquisition/release was not exercised on these fixed-size allocations.

V. Results and Discussion

Table I summarizes the configurations. Online balancing improves different initial partitionings without prescribed weights. Against homogeneous sum-of-parts baselines, the best observed online performance reaches 101% on G4C44, 116% on A2G3, and 119% on G2H2. These super-scaling results exceed the measured targets. METIS is more variable, demonstrating the value of preserving interface structure. Fig. 2 shows the progression from π_o^M to π^{best} on G4C44.

Complete RK45 simulations achieve time-to-solution speedups of 2.4×, 2.8×, and 7.7× on G4C44, A2G3, and G2H2, respectively (Fig. 3). The largest improvement is on G2H2, where online load balancing reduces wall-time from 181.7 h to 23.6 h.

VI. Conclusions

Robust heterogeneous load balancing arises from interaction between runtime cost inference and incremental repartitioning in PyFR [3]. The signal captures element type, interface topology, device performance, and communication overlap, while diffusion preserves useful partition structure.

Acknowledgment

Supported by AFOSR FA9550-23-1-0232 under Dr. Fariba Fahroo (SM and FDW), NSF 2112356 (SM), and EPSRC EP/W026686/1 (SA). Computations used Launch (NSF 2232895), Grace, and TACC Vista (OCE26006).

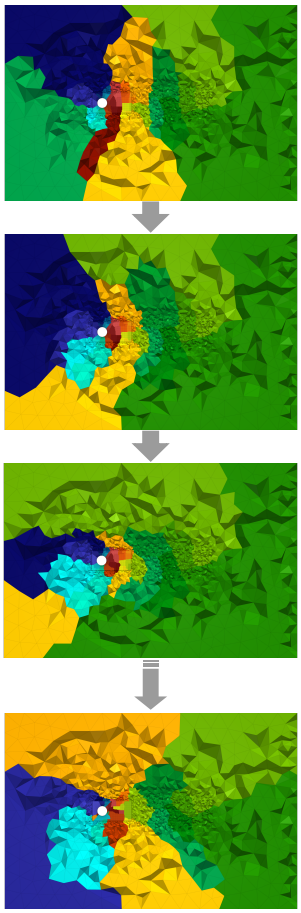


Fig. 2: Online diffusion from π_o^M to π^{best} on the 48-partition G4C44 setup.

References

[1] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, pp. 359–392, 1998.
[2] T. Dzanic, S. Girimaji, and F. Witherden, "Partially-averaged Navier–Stokes simulations of turbulence within a high-order flux reconstruction framework," *J. Comput. Phys.*, vol. 456, 110992, 2022.
[3] F. D. Witherden et al., "PyFR v2.0.3: Towards industrial adoption of scale-resolving simulations," *Comput. Phys. Commun.*, vol. 311, 109567, 2025.
[4] F. D. Witherden, B. C. Vermeire, and P. E. Vincent, "Heterogeneous computing on mixed unstructured grids with PyFR," *Comput. Fluids*, vol. 120, pp. 173–186, 2015.