STATIC-LOAD BALANCED AND SPATIAL PARTITIONING PARALLEL OVERSET GRID ASSEMBLER WITH IMPLICIT HOLE CUTTER

Orhan Shibliyev, Ibrahim Sezai

Eastern Mediterranean University (EMU)
Department of Mechanical Engineering
Gazimagusa, Mersin 10, Turkey
e-mail: orhan.shibliyev@emu.edu.tr,
e-mail: ibrahim.sezai@emu.edu.tr

Key words: Overset methods, Computational Fluid Dynamics, Parallel CFD applications

Summary. Overset mesh approach is useful for unsteady fluid flow problems which involve multiple moving components with irregular geometries and where generation of single grid around all components with satisfactory topology and resolution is time consuming. Original partitioning layout which is provided by unweighted graph partitioner is remapped to spatial layout to gather overlapping cells together. To achieve load balance, remapping is performed with an adaptive Cartesian or load map and at each refinement weighted graph partitioner is used to obtain balanced partition. To avoid inter-process communication and expensive intersection tests, bounding-box-based inverse map is used as a compromise between Exact Inverse Map and Approximate Inverse Map. Load balancing is static and performed only in the first time step or whenever load distribution is imbalanced due to relative motion of components. Several load estimation types are tested to determine correlation between estimated and temporal load differences among processsors.

1 INTRODUCTION

Overset grid assembly consists of several tasks such as partitioning and distribution of meshes to processors, cell type identification which includes donor search, hole cutting and filtration of cells prior to cell identifications, transfer of cells for load balancing and/or relative mesh motion. In serial computing environment, cell type identification is the sole set of algorithms which should be optimized as much as possible. In parallel environment, however, all tasks contributes to run-time performance and load imbalance. Weights of tasks depend on many conditions, for example, in this work, the code is designed in such a way that in initial time step all tasks take approximately same amount of time to perform while in subsequent time steps cell type identification becomes the dominant operation.

Before emergence of overset mesh approach, for flow solvers, work load used to be partitioned with graph partitioner which is usually obtained with METIS/ParMETIS [1] by giving equal weights to graph vertices. On the other hand, equal-weight partitioning does not result in balanced load distribution for OGA as definition of load for OGA is strongly related to number of overlapping mesh-blocks in unit region. Since load is related mainly to degree of overlap, it is reasonable, for OGA, to use spatial partitioning which is similar to "rendezvous" approach of Plimpton et al. [2] and also used by SUGGAR++ [3].

Equal-weight partitioning results in balanced partitions when as for flow solvers, each cell takes the same amount of time to complete tasks. Additional load balance is required to follow either equal-weight or spatial partitioning.

Instead of performing dynamic load balance in each time step [4], static load balance is applied in only the first time step or whenever load distribution is deprecated due to mesh motion. Static load balancer is especially effective for problems with small time steps because of low rate of load deprecation.

An adaptive Cartesian mesh or load map is used to determine load distribution. Mesh cells are registered to bins of load map spatially therefore, partitioning approach is spatial partitioning. Cells in bins are partitioned with graph partitioner of METIS with weights set to bin loads to optimize partitioning for OGA. If load distribution is still imbalanced, the most overloaded bin is refined resulting in adaptive load map. After several adaptive refinements, when load balance is achieved, partitions are distributed to processors.

Static load balancer requires load estimation as no temporal information is available before simulation. Success of static balancer is dependent on accuracy of load estimation. Several load definitions are tested to determine correlation between estimated and temporal load differences.

2 RESULTS and DISCUSSION

Test cases are run on TUBITAK ULAKBIM TRUBA's [5] Barbun cluster [6]. Test case is composed of a background mesh and overset meshes as shown in Figure 1. Background mesh is a regular Cartesian mesh while overset mesh is a rectangular radial mesh with a single hole. Overset meshes rotate at constant angle of 10^{-4} radians in counter-clockwise direction in subsequent assemblies.

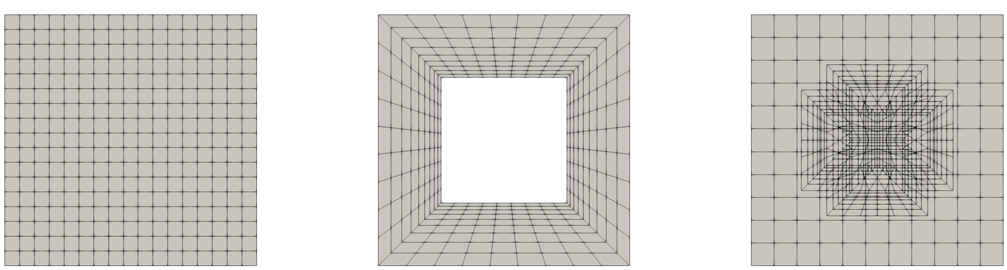

Figure 1: Background and overset meshes in different configurations. Background and overset mesh in 2-mesh configuration and 6-mesh configuration.

There are four input parameters as shown in Table 1. Only the influence of load estimation types are presented due to space constraints.

Table 1: Parameters

Number of processors (np)	8, 16, 24, 32
Total number of meshes (nmesh)	2, 6, 10
Total number of cells (ncell) (M)	1, 3, 7, 11
Hole cutting method (hc)	Implicit, direct
Load estimation method (le)	Solver, improved, area
Number of assemblies	3

Table 2: Load estimation methods.

Estimation type	Definition	Accuracy	Conservative
Solver	$L_s = \sum_{i=1}^{N} n_i$	Low	Yes
Improved	$L = \begin{cases} L_s, & N > 1 \\ 0, & N \leq 1 \end{cases}$	Medium	No
Area	$L = \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} A_{ij}(n_i/A_i + n_j/A_j)$	High	No

Three different load estimations as shown in Table 2 are tested to determine correlation between estimated and temporal deviations. Deviation is defined as normalized difference of maximum and minimum loads. Solver-based estimation is popular with flow solvers while area based estimation is optimized for OGA by involving intersection area in estimation. Improved estimation is improved version of solver based estimation and it nullifies load if there is no more than one mesh in unit region. In Figure 2a, estimated deviations decrease from area to solver based estimations and also only solver based estimation results in estimated load lower than the threshold of 10%. Although all estimated load deviations were reduced below 10% threshold in load mapping, estimated loads before and after adaptive refinement do not match except for solver based estimation. This is due to non-conservative nature of load estimations except solver based estimation.

Figure 2b shows temporal deviation of donor search. With increasing number of processors, solver based estimation tends to result in lower deviations. However, absolute run time differences decrease as well with number of processors as shown in Figure 3, that is, percent differences become less significant.

Figure 2c shows difference of estimated and temporal deviation of donor search. From clustering of data points around zero for area based estimation, it is clear that area based estimation is successful in predicting temporal deviations while solver based estimation is not.

3 CONCLUSIONS

As a result, area based estimation is found to be superior over other estimation types in terms of predicting temporal deviation. Improved version is compromise between solver

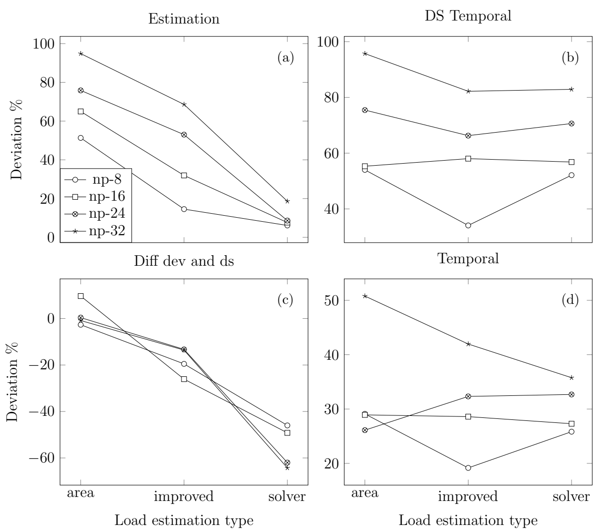

Figure 2: Deviations for different load estimations. hc-direct, ncell-11M, assembly-0

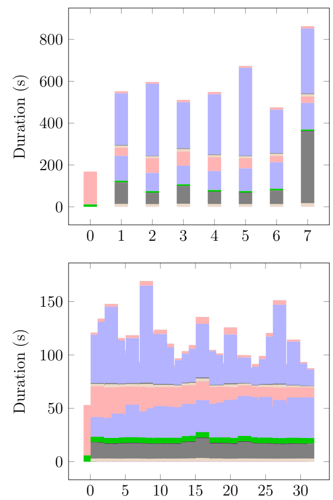

Figure 3: Total task durations for each processor. nmesh-10, ncell-11M, hc-implicit, assembly-0. Top: np-8, bottom: np-32

and area based estimations and intended to be used if usage of area based estimation becomes costly for higher number of processors.

References

[1] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[2] Steven J Plimpton, Bruce Hendrickson, and James R Stewart. A parallel rendezvous algorithm for interpolation between multiple grids. *Journal of Parallel and Distributed Computing*, 64(2):266–276, 2004.

[3] Ralph W Noack, David A Boger, Rober F Kunz, and Pablo M Carrica. Suggar++: An improved general overset grid assembly capability. *Proceedings of the 47th AIAA Aerospace Science and Exhibit*, pages 22–25, 2009.

[4] Beatrice Roget and Jayanarayanan Sitaraman. Robust and efficient overset grid assembly for partitioned unstructured meshes. *Journal of Computational Physics*, 260: 1–24, 2014.

[5] Truba. `https://www.truba.gov.tr/index.php/en/main-page/`.

[6] Truba-barbun. `http://wiki.truba.gov.tr/index.php/TRUBA-barbun`.