High Performance Optimizations and Dynamic Load Balancing for Computational Aerodynamics

Robin R, Blessy Boby, Shilpa S, Gmon K

Under the guidance of Prof. **Dr. Ajeesh Ramanujan**
Project done under VSSC, by the guidance of **Harichand M. V**, Scientist Engineer, VSSC

Abstract

Load Balancing is a challenging and interesting problem that appear in all High Performance Computing(HPC) environments. Most of the applications running in HPC environments use MPI based parallel computing. This approach needs load assignment, before starting the com- putations, which is difficult in many problems where computational load changes dynamically. In VSSC Aeronautics Entity, parallel computing is used to solve Computational Fluid Dynamics problems. Computational fluid dynamics is a branch of fluid mechanics that uses numerical analysis and data structures to solve and analyze problems that involve fluid flows. Memory bandwidth utilization of CFD problems are very low, making them computationally inefficient.

Problem Statement

The main objectives of the project are to

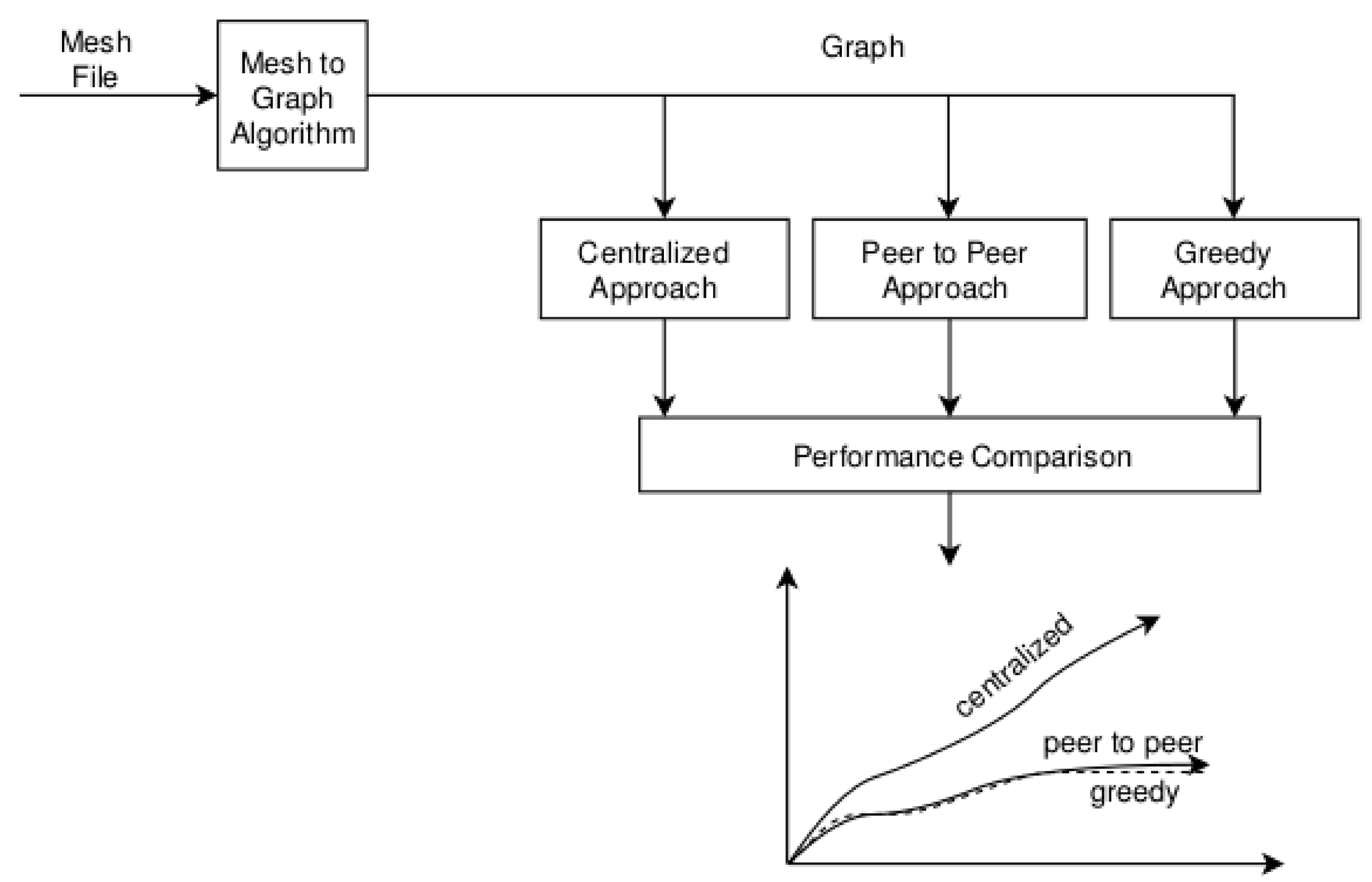
- Improve the dynamic load balancing capability in an MPI environment for solving CFD problems.
- Improve data locality of CFD problems.

Concepts

- High-performance computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly. It is occasionally used as a synonym for supercomputing.
- Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical analysis and data structures to solve and analyze problems that involve fluid flows are used to perform the calculations required to simulate the interaction of liquids and gases with surfaces defined by boundary conditions.
- Mesh File is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics and solid modeling.The faces usually consist of triangles (triangle mesh), quadrilaterals, or other simple convex polygons.
- MPI is a specification for the developers and users of message passing libraries. It is a system that aims to provide a portable and efficient standard for message passing.

Method

- The input is provided in the form of mesh vtk format. The SU2 mesh format carries an extension of .su2, and the files are in a readable ASCII format.
- This mesh file is given as input into the algorithm which convert it into graph file.
- Then the graph file is partitioned using METIS into different subgraphs.
- For the load balancing, different communication paradigms are applied to the graphs.
  - First, Centralized approach is applied, the result is analysed, the drawbacks are identified.
  - Second, Peer to peer approach is applied, the result is analysed and the drawbacks are identified.
  - Third, Greedy approach is applied, the result analysed.
- Performances of the three algorithms are analysed and comparison graph is plotted.



Result

- For the centralized approach, the amount of communication can be represented as 2n + 2(m-1)*(n-1), where m is no of cores and n is no of vertices. This is the worst case scenario. This is because one of the cores act a master and it requires a lot of mpi calls.
- For the peer to peer approach, the amount of communication can be represented as 2(m-1)*m. When compared to the centralized approach, peer to peer is faster and more efficient as it involves less no of mpi calls. Also the communication order and dependency are not considered.
- For the greedy approach, the amount of communication is same as the peer to peer approach, but the difference lies in the delay. Here the dependency is considered and the communication takes place slot wise in which each slot consists of list pair of nodes which are independent to each other and whose communication data size is largest at that point of time.

References

- Adaptive MPI ***http://charm.cs.illinois.edu/manuals/html/ampi/manual.html***
- Mesh File ***https://su2code.github.io/docs/Mesh-File/***
- Metis ***http: //glaros.dtc.umn.edu/gkhome/metis/metis/overview***
- MPI ***https://wiki.mpich.org/armci-mpi/index.php/MainPage***
- VTK Format ***https://vtk.org/wp-content/uploads/2015/04/file-formats.pdf***
- MPI Advanced ***http://mpitutorial.com/tutorials/mpi-scatter-gather-and-allgather/.***
- Charm++ ***http://charm.cs.illinois.edu/research/charm***

Department of Computer Science and Engineering
College of Engineering, Trivandrum