Scalable Mesh Generation for HPC Applications

Rajeev Jain
Mathematics and Computer Science Division
Argonne National Lab
jain@mcs.anl.gov

Navamita Ray
Mathematics and Computer Science Division
Argonne National Lab
nray@mcs.anl.gov

Iulian Grindeanu
Mathematics and Computer Science Division
Argonne National Lab
iulian@mcs.anl.gov

Danqing Wu
Mathematics and Computer Science Division
Argonne National Lab
wuda@mcs.anl.gov

Vijay Mahadevan
Mathematics and Computer Science Division
Argonne National Lab
mahadevan@anl.gov

ABSTRACT

Computational solvers simulating physical phenomenon on complex domain geometries need well resolved, high-quality meshes to tackle the discrete problem efficiently. Mesh generation for HPC applications is a complex process requiring access to geometry data and efficient mesh data-structures in a parallel setting as well as optimization techniques for quality. The performance of the mesh generation algorithms are affected by the scalability of such components.

MeshKit, developed as a component of the open-source SIGMA tool chain, supports variety of meshing algorithms leveraging the scalable interfaces in SIGMA to geometry data (CGM) and unified data-structures(MOAB) that can be used for HPC applications. In this poster we present performance results of two key aspects of a parallel mesh generation algorithm (1) resolution of entities on partition boundaries and (2) parallel IO under SIGMA.

Keywords

Mesh generation, Software framework, HPC Applications, Performance

1. SIGMA: SCALABLE INTERFACES FOR GEOMETRY AND MESH BASED APPLICATIONS

SIGMA[1] is an open-source tool chain that provides interfaces and tools to understand geometry models, create high quality unstructured meshes along with unified data-structures to load and manipulate parallel computational meshes for various HPC applications. Optimal quality mesh generation is a complex process for complicated curved problem geometries, with physics imposed spatial resolution requirements. These tools simplify the process of generation and handling of discrete meshes with scalable algorithms to leverage efficient usage from desktop to petascale architectures.

Mesh Generation Toolkit(MeshKit)

MeshKit[4, 5, 2] is an open-source mesh generation environment that enables both users and developers to create high quality CAD and mesh models for simulations. It provides a robust, parallel and flexible interface, where various state-of-art open-source mesh generation algorithms and non open-source meshing libraries can be accessed uniformly. It uses the Common Geometry Module (CGM) for accessing geometries and the data structures in the parallel infrastructure Mesh Oriented datABase(MOAB) for mesh representation.

Common Geometry Module(CGM)

The Common Geometry Module(CGM)[3] is a geometry library that provides modeler-independent means for accessing and modifying geometry through a generic interface, while maintaining a reference to the original model in its native format avoiding translation between different geometry formats. In addition, it provides several unique capabilities among geometry engines, like support for non-manifold topology, support for virtual geometry, support for accessing the geometry in parallel, or mesh-based geometry representation.

Mesh Oriented datABase(MOAB)

The Mesh Oriented datABase(MOAB)[6] is a parallel mesh framework for representing both unstructured and structured meshes and provides an array- based data-structure to efficiently represent associated data, with abstractions to handling several different file formats, and enabling in-memory modifications. It provides a number of parallel mesh capabilities such as a parallel IO based on the parallel HDF5 library, interfaces to existing state of the art mesh partitioners, algorithms to resolve entities on partition boundaries, exhanging ghost layers and field data exchanges. It can also support hierarchical mesh generation through uniform refinement and quality metrics.

Performance

We present the performance(weak scalability studies) of the parallel IO and resolution of shared entities for MOAB and

MeshKit. In MOAB, a rectangular domain was meshed as an unstructured hexahedral mesh stored in memory and saved in file. The partitioned mesh was generated on each task locally and after mesh generation all the tasks wrote to the same file which was subsequently read in memory (which also resolves the entities on the partition boundary) to explore the capability of the parallel IO file system. It was found that the parallel IO didn't scale and lost efficiency significantly upto thousand of cores. This behaviour seems to be due to the amount of indirect referencing involved to access entities on each partition while writing to or reading from a single file. However, it still further investigation and optimization. The shared entities at the interfaces between partitions were resolved using a vertex global ids based algorithm with crystal router, an efficient gather-scatter algorithm for sparse communication.The interface resolution is highly scalable and maintains efficiency to thousands of processors.

One of the HPC applications using MeshKit is the automatic mesh generation of complex nuclear reactor assemblies and core geometries(templates). In our weak scalability studies, assemblies were generated with approximately 7500 hexes and subsequently refined two time times resulting in 60K and 480K hexes at each level. The shared entity resolution algorithm used was a geometric proximity based vertex-merge algorithm for both core assemblies and after refinement. We obtained good scaling for this algorithm for upto 1K cores. On the other hand, the parallel IO deteriorated similar to the previous test.

2. REFERENCES

[1] SIGMA website. `http://sigma.mcs.anl.gov`, 2015. Accessed: 2015-06-12.

[2] R. Jain and T. J. Tautges. Generating unstructured nuclear reactor core meshes in parallel. In *23rd International Meshing Roundtable*, 10/2014 2014.

[3] T. J. Tautges. CGM: A geometry interface for mesh generation, analysis and other applications. *Engineering with Computers*, 17(3):299–314, 2001.

[4] T. J. Tautges and R. Jain. Creating geometry and mesh models for nuclear reactor core geometries using a lattice hierarchy-based approach. *Engineering with Computers*, Sept. 2011.

[5] T. J. Tautges, J. Kraftcheck, J. Porter, A. Caceres, I. Grindeanu, D. Karpeev, R. Jain, H.-J. Kim, S. Cai, S. Jackson, J. Hu, B. Smith, C. Verma, S. Slattery, and P. Wilson. MeshKit: a Open-Source library for mesh generation. In *Proceedings, SIAM Conference on Computational Science & Engineering*, Reno, NV, Mar. 2011. SIAM.

[6] T. J. Tautges, R. Meyers, K. Merkley, C. Stimpson, and C. Ernst. MOAB: a Mesh-Oriented database. SAND2004-1592, Sandia National Laboratories, Apr. 2004.