# Scalable Mesh Generation for HPC Applications

Rajeev Jain, Navamita Ray, Iulian Grindeanu, Danqing Wu, Vijay Mahadevan

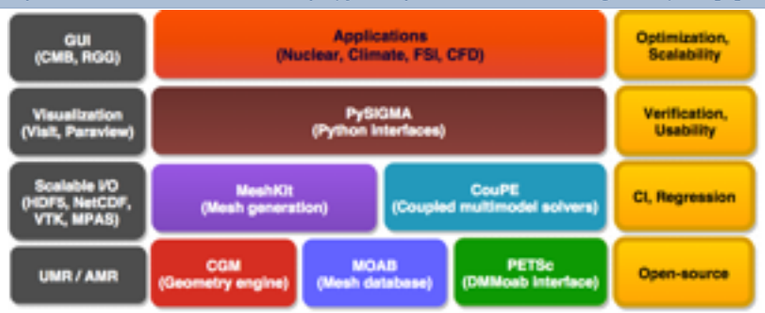
Argonne NATIONAL LABORATORY

## Problem Description

- 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 preservation.
- A key component of any such parallel mesh infrastructure is the parallel I/O file system whose performance is vital to any complex application workflow.
- **MeshKit,** developed as a component of ***SIGMA*** tool chain, supports variety of meshing algorithms leveraging the scalable interfaces in SIGMA to geometry data (**CGM**) and unified parallel data-structures (**MOAB**) that can be used for HPC applications.

## Scalable Interfaces for Geometric and Mesh based Applications(SIGMA)

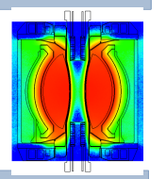
*An open-source toolchain to simplify computational modeling workflow[4]*




## Common Geometry Module(CGM)

*Libraries for querying and modifying solid geometry model[1]*

- Common API and topological model to access geometry representations for a variety of solid modeling engines such as ACIS, Open-CASCADE, Facet-based surfaces, etc.
- Provides parallel geometry model querying and handling; virtual geometry, non-manifold topology.



## Mesh Oriented datABase(MOAB)

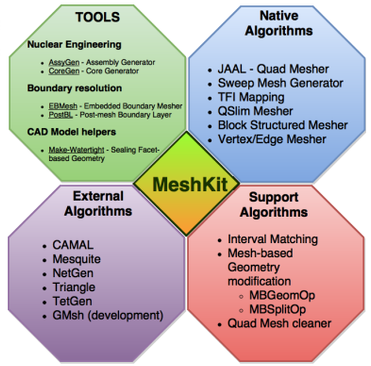
*Efficient array-based data structures to handle mesh queries[2]*

- Represent unstructured and structured mesh and field data on a mesh efficiently.
- Parallel mesh capabilities:
  - IO(parallel HDF5 library based)
  - link to existing state of the art mesh partitioners(ParMetis, Zoltan)
  - algorithms for resolving entities on shared processor interfaces(geometric proximity based, global id based)
  - exchange ghost layers, mesh migration, field data exchange
- Hierarchical mesh generation for unstructured meshes through uniform refinement and quality metrics.
- Multimesh intersection and transfer algorithms.

## Mesh Generation Toolkit(MeshKit)

*Flexible algorithms to generate high-quality meshes[3]*

- Provides a collection of meshing algorithms to support mesh generation; a platform to develop mesh generation algorithms.
- Coordination of BREP-based meshing process, mesh smoothing, optimization
- Notable meshing algorithms include embedded boundary meshing, watertight models, mesh based geometry generator
- Exposes mesh manipulation and generation features such as **Copy**, **Move**, **Rotate** and **Extrude Mesh**.




- Uses SIGMA tools CGM and MOAB for accessing geometry model and mesh representation in a parallel setting.

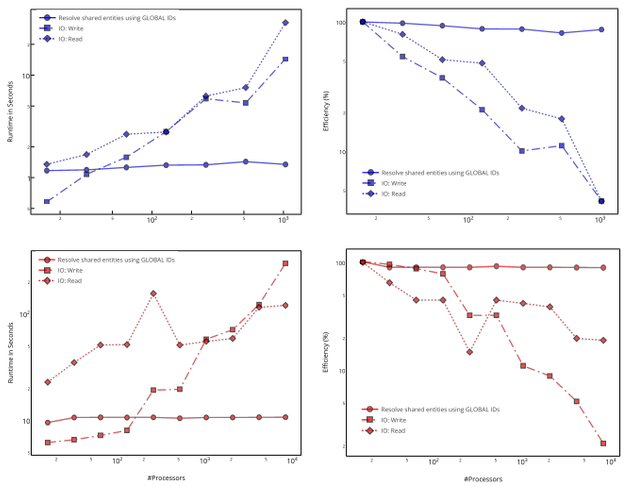
## MOAB Weak Scaling Studies

- A mini-app **GenLargeMesh** for generating large meshes to explore the capability of MOAB's parallel infrastructure and of the parallel IO file system.
- It creates 3D hexahedral meshes for a rectangular domain in-memory and writes out the mesh to a file in parallel.
- The partitioned mesh is generated on each task locally, and all tasks write to the same file. Approximately 21.6K hexes per task; largest mesh sizes: 221M (Blues) and 1.8B (Vesta).
- Parallel I/O (based on hdf5 library) reads from or writes to a single file which involves indirect referencing to access entities on each partition. This effects the weak scalability of the I/O negatively as seen below.
- Shared entities at the interfaces between partitions is resolved using vertex global ids and crystal router, an efficient gather-scatter algorithm for sparse communication. The interface resolution is highly scalable and maintains efficiency to thousands of processors.

**Machine Details:**

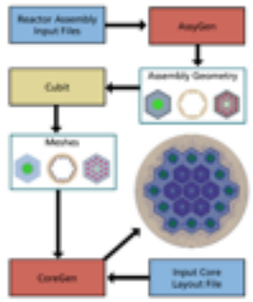
**Blues**: 310 nodes, 16 cores/node (Intel Sandy Bridge), 64 GB of RAM per node

**Vesta**: 2,048 nodes, 16 cores/node(1600 MHz PowerPC A2), 16 GB RAM per node

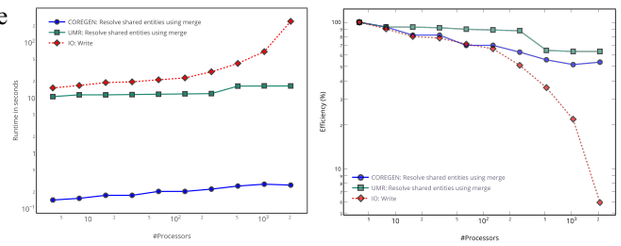



## MeshKit for Nuclear Engineering

- Automatic mesh generation describing complex nuclear reactor assemblies and core geometries(templates) using ***MeshKit***
- ***Reactor Geometry Generator a.k.a.,*** **RGG GUI,** developed in collaboration with *Kitware Inc*.
- Weak scalability studies on **Blues** for assembly generation along with two levels of degree 2 uniform refinement. Each core 7500 hexes; 60K and 480K after 2 refinement levels; largest mesh size 1.1B hexes.
- Entities on the shared processor interface are resolved using a geometric proximity based vertex-merge algorithm for both core assemblies and after refinement; maintains scalability upto 1K cores.




- Parallel write deteriorates significantly around 1K cores.




## Conclusion

- SIGMA tools provide necessary components and interfaces for developing advanced mesh generation capabilities.
- The parallel infrastructure in MOAB for resolving shared entities using geometrical proximity and global ids for vertices are highly scalable onto thousands of processors.
- The parallel IO deteriorates in performance and needs further investigation and optimization.

## References

[1] T. J. Tautges, R. Meyers, K. Merkley, C. Stimpson, and C. Ernst, "MOAB: A Mesh-Oriented Database," Sandia National Laboratories, SAND2004-1592, Apr. 2004.

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

[3] Rajeev Jain, T.J. Tautges, "Generating Unstructured Nuclear Reactor Core Meshes in Parallel", In Proceedings of 23rd International Meshing Roundtable, Oct 2014.

[4] http://sigma.mcs.anl.gov/

## Acknowledgements

This work was supported by UChicago Argonne, LLC, a U.S. Department of Energy Office of Science laboratory, operated under Contract No. DE-AC02-06CH11357.

U.S. DEPARTMENT OF ENERGY