Unstructured Meshing Techniques

FASTMath Team Members (only listing lead from each group): K. Devine, L. Diachin, K.E. Jansen, M.S. Shephard and T. Tautges

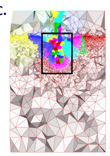
Unstructured meshes, often adaptively defined, can yield required levels of accuracy using many fewer degrees of freedom at the cost of more complex parallel data structures and algorithms. FASTMath is providing the parallel unstructured mesh data structures and services needed by developers of PDE solution procedures targeted for exascale computers

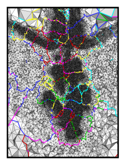
Unstructured Meshes for Exascale Computers

Key unstructured mesh technologies needed

- Effective parallel mesh representations meeting the needs of applications
  - MOAB - Compact representation that takes full advantage of mesh sets
  - PUMI/FMDB - Representation designed for evolving (adapted) meshes
- Base parallel mesh funtionalities
  - Control of partitioned mesh
  - Movement of mesh entities between processes
  - Associated data, ghosting, grouping, etc.
- Key services
  - Load balancing (static and dynamic)
  - Mesh-to-mesh solution transfer
  - Mesh optimization
  - Mesh adaptation

Poster highlights some recent developments



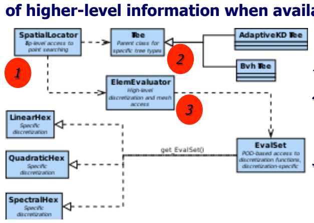


Parallel Point Location for Solution Transfer

Goal: Simplify combination of geometric searching and discretization

- Geometric searching: parallel point-in-element query for 2D/3D, for various element topologies (tri/quad/polygon, tet/prism/pyramid/polygon/hex)
- Discretization: linear, quadratic, spectral FE, FV

Challenge: (local) discretization interacts with both mesh and solver sides of codes; support reuse on both sides, while taking advantage of higher-level information when available



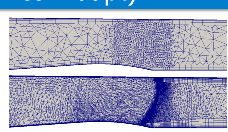
1. *SpatialLocator top-level access*
2. *Tree defines abstract interface, each tree type better at specific tasks*
3. *ElemEvaluator caches entity-specific data, then calls POD (Plain Old Data)-based functions*

Mesh Adaptation (MeshAdapt)

A general parallel mesh adaptation tool

Recent Developments

- Boundary layer thickness adaptation
- Adapted meshes to 92B elements
- Improved mixed element mesh adaptation



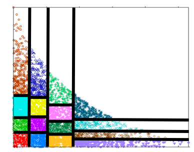
Hybrid MPI+threads Geometric Partitioning

Goal: Effective, efficient partitioning on parallel multicore computers

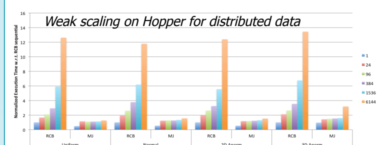
- Geometric partitioning: Assign data to parts based on physical proximity
- Effective for dynamic applications and those requiring geometric locality (e.g., adaptive mesh refinement, particle methods, contact detection)

New Parallel Multijagged method (MJ):

- Generalization of Recursive Coordinate Bisection (RCB) that uses multiple cuts in each dimension
- Available in Zoltan2



A 4x4 MJ partition



MJ scales better than RCB due to fewer recursion levels & lower data movement.

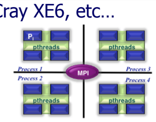
MJ's MPI+OpenMP implementation enables faster partitioning for applications using MPI+threads.

# MPI ranks	Partitioning time (s); 51M coords	
	MJ; 6 threads per rank	RCB
4	11.3	26.2
8	8.9	13.1
16	3.0	7.1

Two-Level Partitioning – MPI and Threading

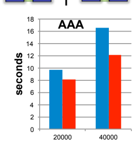
Need to account for hybrid architecture of BG/Q, Cray XE6, etc…

- Reduce data movement
- Reduce memory usage



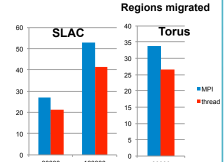
Two level partition approach

- Partition to process, then partition on process
- MPI between process, threads on process
- Applications only aware of two level partition – Parallel Control Utility (PCU) manages messages and threads at the mesh infrastructure (PUMI) level
- Shared memory optimizations
  - Inter-process entity movement
  - Inter-thread movement



Test results on Blue Gene/Q

- AAA mesh: 2M tets, 32 parts, 2 nodes
- SLAC mesh: 17M tets, 64 parts, 4 nodes
- Torus mesh: 610M tets, 4096 parts, 256 nodes
- Test: local migration, all MPI vs. 1 MPI rank/16 threads per node
- Speedup: up to 27%



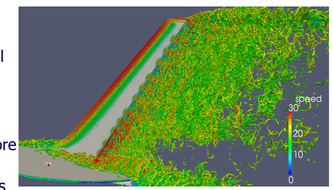
Massively Parallel PDE Solver (PHASTA)

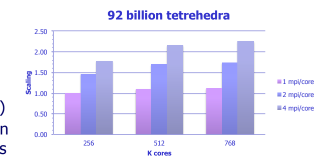
Implicit, Adaptive Grid CFD

- Extreme Scale Applications:
  - Aerodynamics flow control
  - Multiphase flow
- Early Science Project (MIRA)
- Full Machine Strong scaling
  - Variable MPI processes/core
  - 92 Billion Tetrahedra
  - 262144 to 3,145,728 parts
  - 1/core 100% scaling
  - 2/core 146-155% scaling
  - 4/core 178-226% scaling

Partitioning pipeline

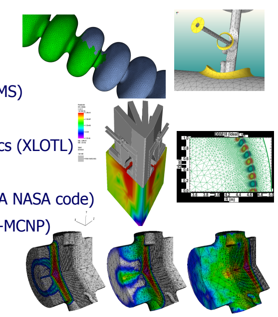
- Zoltan ParMetis of 180M elements to 16k parts (global)
- Refinement to 92B then Zoltan HyperGraph up to 3072k parts





Application List

- Accelerator Modeling (ACE3P)
- Climate data analysis (Par-NCL)
- FE-based neutron transport (UNIC)
- Fluid/Structure interaction (AthenaVMS)
- Fusion Edge Physics (XGCI)
- Fusion first wall chemistry & dynamics (XLOTL)
- Fusion Plasmas (M3DC1)
- High-speed viscous flows (FUN3D – A NASA code)
- Monte Carlo neutron transport (DAG-MCNP)
- Multiphase reactor flows (PHASTA)
- SEM-based CFD (Nek5000)
- Solid Mechanics (Albany)



Future Plans

Continue to provide unstructured mesh data structures and services

- Algorithms needed to operate on exascale systems
- Effective implementation on exascale systems
- Extend and expand services to meet the needs of applications
- Demonstrate use on complex geometry applications

More Information: http://www.fastmath-scidac.org or contact Mark Shephard, shephard@rpi.edu, 518-276-8044

Argonne National Laboratory · Berkeley Lab · Lawrence Livermore National Laboratory · Sandia National Laboratories · Berkeley University of California · Columbia University · Rensselaer · SMU · The University of British Columbia