The Data Transfer Kit: A Geometry Rendezvous-Based Tool for Multiphysics Data Transfer

Stuart R. Slattery and Paul P.H. Wilson, University of Wisconsin - Madison

Roger P. Pawlowski, Sandia National Laboratories

May 1, 2013

Outline

- Overview
- Concepts and Geometric Rendezvous
- DTK Algorithms
- Data Transfer Example
- Scaling Studies

What is DTK?

- Collection of geometry-based data mapping algorithms for shared domain problems
- Data maps allow for efficient movement of data in parallel (e.g. between meshes of a different parallel decomposition)
- Ideally maps are generated at a desirable time complexity (logarithmic)
- Input mesh and geometry data drive the map generation
- Should be viewed as a service providing suite of concrete algorithm implementations

Software Overview

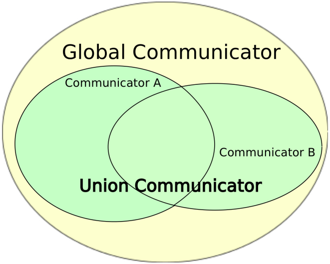
- Preliminary development of mesh-based capabilities during summer 2012 CASL internship at ORNL
- Additional development of geometry-based capabilities during fall 2012
- Implemented in C++
- Heavy use of the Trilinos scientific computing libraries
- Open-source BSD 3-clause license
- https://github.com/CNERG/DataTransferKit

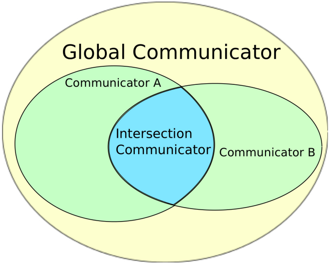
Concepts and Geometric Rendezvous

- Communicators
- Shared Domain Problems
- Parallel Topology Maps
- The Rendezvous Algorithm

Communicators

- DTK handles source and target communicators of arbitrary relation
- Any amount of overlap or lack thereof supported
- A global communicator required (doesn't have to be MPI_COMM_WORLD)





Shared Domain Problems

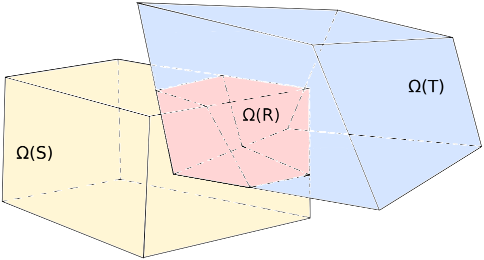


Figure: *Shared domain example.*

$\Omega(S)$ (yellow) is the source geometry, $\Omega(T)$ (blue) is the target geometry, and $\Omega(R)$ (red) is the shared domain.

- Defined over a communicator that encapsulates the union of the source and target communicators
- Source and target must be of same geometric dimension
- The rendezvous algorithm leveraged to provide parallel topology maps for shared domains

Parallel Topology Maps

- An operator, $\mathbf{M}$, that defines the translation of a field, $\mathbf{F}(s)$, from a source spatial domain, Ω_S, to a field, $\mathbf{G}(t)$, in the target spatial domain Ω_T, such that $\mathbf{G}(t) \leftarrow \mathbf{M}(\mathbf{F}(s))$ and $\mathbf{M} : \mathbb{R}^D \rightarrow \mathbb{R}^D, \forall r \in \Omega_R$, where Ω_R is the geometric rendezvous of the source and target.
- $\mathbf{M}$ is in general expensive to generate but cheap to apply
- For static Ω_S and Ω_T, building $\mathbf{M}$ is a one-time, upfront cost

The Rendezvous Algorithm

- Initially developed by the SIERRA team in the mid-2000's for parallel mesh-based data transfer [1]
- Creates a parallel topology map that can be used repeatedly for data transfer
- Map execution uses asynchronous strategy (posts and waits) with minimal messages
- Effectively $N * log(N)$ time complexity for parallel topology map generation
- Relies on the generation of a secondary decomposition of the source and target meshes with a geometric-based partitioning (RCB)

---

[1] S. Plimpton, B. Hendrickson, and J. Stewart, A parallel rendezvous algorithm for interpolation between multiple grids, Journal of Parallel and Distributed Computing, vol. 64, pp. 266276, 2004

The Rendezvous Decomposition

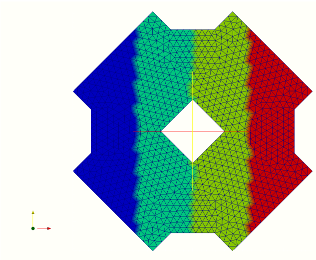


Figure: *Source mesh for 2D shared domain example.*

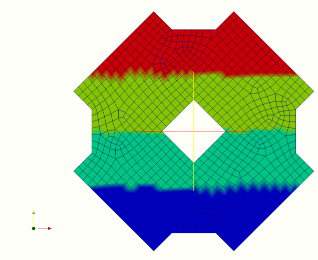


Figure: *Target mesh for 2D shared domain example.*

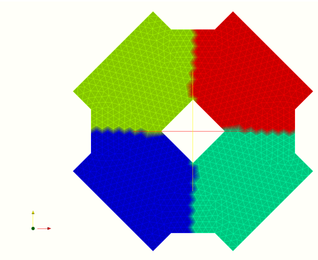
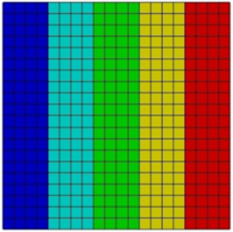


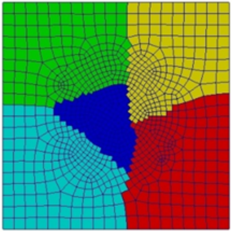
Figure: *Rendezvous decomposition for 2D shared domain example.*

Searching the Rendezvous Decomposition

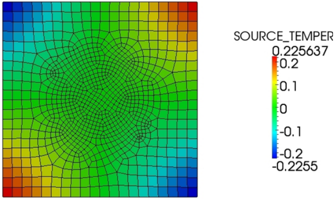
- Hierarchical parallel search tree
- Rendezvous decomposition provides parallel search
- kD-tree provides on-process proximity search
- Newton iterations provide final point location
- Results in reasonable scalability

Standard Mesh-Based Rendezvous Map



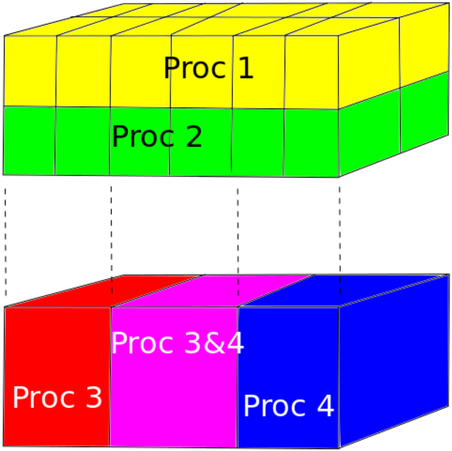


- Mesh-to-Mesh transfer
- Used to move $\mathbf{F}(\hat{r})$ between meshes of arbitrary distribution
- Requires user code for evaluations in mesh elements

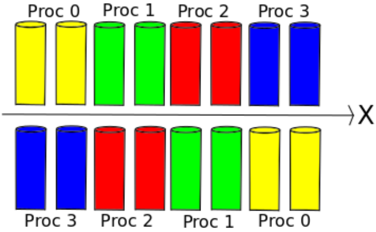


Other Rendezvous-Based Maps: Integral Assembly

- Mesh-to-geometry transfer
- Used to assemble f_Ω with mesh and geometry of arbitrary distribution into measure-weighted integral
- The mesh is assumed conformal
- Requires user code for integrations in mesh elements
- See example/IntegralAssembly

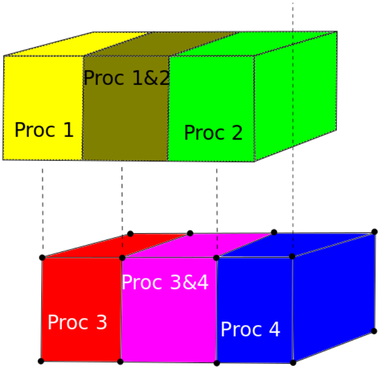


Other Rendezvous-Based Maps: Geometry to Geometry



- Simple geometry-to-geometry transfer capability
- Geometries are assumed conformal
- Requires user code for evaluations in geometry
- See example/GeometryToGeometry

Other Rendezvous-Based Maps: Geometry to Mesh



- Similar to mesh-based rendezvous
- Does not require a mesh, conceptual in this case
- Requires user code for evaluations in geometry
- See example/GeometryToMesh

DTK Implementation Scaling Results

- Mesh-to-mesh transfer
- Worst case scenario study (all-to-all) with random points
- Qualitatively similar to the SIERRA results
- Largest test problems so far over 1.0E9 elements and 1.0E5 cores

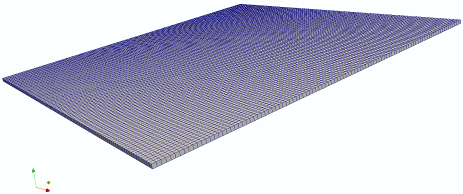


Figure: *Local mesh partition for scaling studies. This particular mesh partition has 1.0E4 tri-linear hexahedrons.*

Strong Scaling

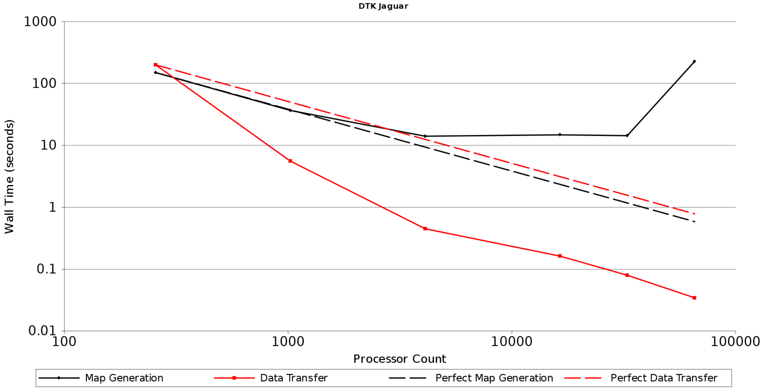


Figure: Strong scaling study results. The solid black curve reports the wall time to generate the mapping vs. number of processors while the solid red curve reports the wall time to transfer the data vs. number of processors. The dashed lines give perfect strong scaling the map generation (black) and the data transfer (red).

Weak Scaling

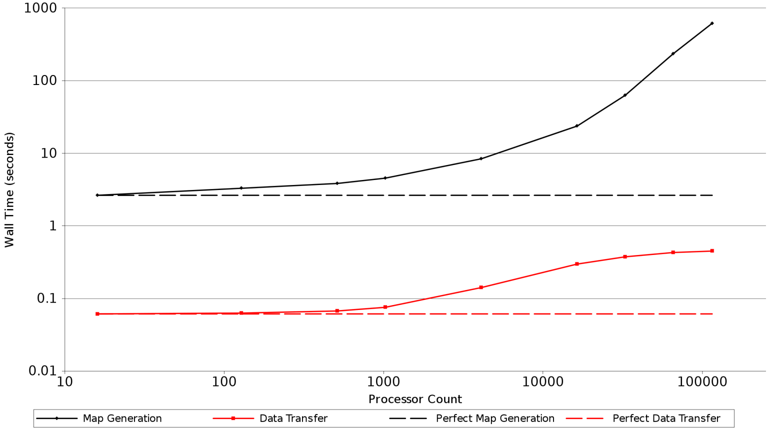


Figure: Weak scaling study results. The solid black curve reports the wall time to generate the mapping vs. number of processors while the solid red curve reports the wall time to transfer the data vs. number of processors. The dashed lines give perfect weak scaling the map generation (black) and the data transfer (red).

Acknowledgements