PORTAGE - A Flexible Conservative Remapping Framework for Modern HPC Architectures

R.V. GARIMELLA, P.D. CROSSMAN*, G.A. DILTS, R.S. ERRAMILLI†, C.R. FERENBAUGH, A.M. HERRING, E. KIKINZON, C.M. MALONE, N. RAY, and M.L. ROGERS, Los Alamos National Laboratory

Portage is a massively parallel remapping framework to transfer fields between general polyhedral meshes while conserving integral quantities of interest. The framework also has the capability to remap data between two point clouds. Portage is templated on the component classes required in conservative remapping - search, intersection and interpolation - as well as on the mesh and field managers. Applications supply Portage with custom components while the framework takes care of distributed parallelism using MPI and threaded parallelism using NVIDIA Thrust. This enables Portage to scale to many thousands of cores. Moreover, the imposition of a functional design on the components used by Portage makes it very amenable to achieve task parallelism with runtime systems such as Legion. Portage has been tested in 2D/3D for remapping between general polygonal and polyhedral meshes and between point clouds. We present scaling results for distributed (MPI) and on-node parallelism (OpenMP) on LANL's HPC machines.

ACM Reference Format:
R.V. Garimella, P.D. Crossman, G.A. Dilts, R.S. Erramilli, C.R. Ferenbaugh, A.M. Herring, E. Kikinzon, C.M. Malone, N. Ray, and M.L. Rogers. 2017. PORTAGE - A Flexible Conservative Remapping Framework for Modern HPC Architectures. 1, 1, Article 1 (October 2017), **??** pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Remapping is the transfer of data fields from one computational domain to another. *Conservative remapping* is the transfer of *intensive* data fields from one computational domain to another while conserving an *extensive* quantity [**?**]. For example, we may require transfer of cell-based densities from one computational mesh to another while conserving total mass. Remapping is a necessary step in many numerical simulations in order to transfer data between:

- poor and improved quality meshes in an Arbitrary-Lagrangian-Eulerian (ALE) simulation,
- meshes used by two different physics packages in a multi-physics code, or
- between two meshes used by different simulation codes.

Portage is a massively parallel remapping framework to transfer fields between general polyhedral meshes while conserving integral quantities of interest. The framework also has the capability to remap data between two point clouds and is being extended to be able to remap between meshes and point clouds. Portage is conservative, accurate, bounds-preserving, fast and scalable to many thousands of cores.

*Also with Gordon College.
†Also with Princeton University.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2017 Association for Computing Machinery.
Manuscript submitted to ACM

Portage performs the conservative remapping between meshes by computing the exact intersection of source and target meshes. First, Portage identifies the source cells that may potentially overlap each target cell (*Search*). Then it computes the moments of intersection (area and centroid) between each target cell and overlapping source cells (*Intersect*). Finally, the value on each target cell is computed as the interpolation of source cell values weighted by the moments of intersection (*Interpolation*). Remapping between point clouds follows a similar algorithm with three main steps. Point-cloud based remapping is based on high-fidelity reconstruction of source fields using Local Regression Estimation [**?**].

Portage is templated on all the component classes performing the above steps in remapping. Additionally, it is templated on the mesh and field data managers. This makes Portage extremely customizable at minimal cost. Some classes supplied with Portage are tree or grid based search for meshes and point clouds, 2D/3D intersection (`https://github.com/devonmpowell/r3d`), up to 3rd-order-accurate interpolation, and, mesh wrappers for the Jali (`https://github.com/lanl/jali`) and FleCSI (`https://github.com/laristra/flecsi`) mesh frameworks.

While applications furnish Portage with custom components (or use default ones), Portage takes care of distributed parallelism using MPI and threaded parallelism using NVIDIA Thrust. If the source mesh cells required by each target cell are available on the same rank, the whole process would be embarassingly parallel. However, in the absence of such guarantees on partitioning, the source mesh and field data may have to be transmitted to ranks that need them. In spite of this initial MPI communication, Portage scales very well to thousands of cores. Moreover, the imposition of a functional design on the components used by Portage makes it very amenable to achieve task parallelism with runtime systems such as Legion [**?**].

Portage has been tested in 2D/3D for remapping between general polygonal and polyhedral meshes and between point clouds. We present scaling results for distributed (MPI) and on-node parallelism (OpenMP) on some newer HPC machines at LANL. In particular, we present results for scaling on ASC Snow, an Intel Haswell based machine with 2 sockets per node and 18 threads per socket. We show excellent scaling of Portage on one node using an OpenMP backend to thrust and on thousands of nodes using MPI+OpenMP (strong and weak). Work is on-going to port a recent version of the code to use CUDA.

Future work will include multi-material remapping with interface reconstruction, remapping of vector and tensor fields and comparing the cost of direct mesh-mesh remapping with mesh-particle-mesh remapping. We also plan to specialize Portage for specific applications such as Cartesian meshes or meshes whose topology has not changed but the geometry has changed a little.

Portage is open-sourced under a 3-part BSD licence and is available at `https://github.com/laristra/portage`.

1 ACKNOWLEDGEMENTS

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396, and was supported by the DOE Advanced Simulation and Computing (ASC) program. LA-UR-26-862

REFERENCES

[1] M.J. Shashkov, L.G. Margolin "Second-order sign-preserving conservative interpolation (remapping) on general grids," Journal of Computational Physics, v 184, n 1, 2003, pp. 266-298.

[2] M. Bauer, S. Treichler, E. Slaughter, A. Aiken, "Legion: Expressing Locality and Independence with Logical Regions," International Conference on Supercomputing, Salt Lake City, UT, Nov. 2012.

[3] G.A. Dilts, A. Haque, J. Wallin "Tuned Local Regression Estimators for the Numerical Solution of Differential Equations," Meshfree Methods for Partial Differential Equations, Springer, Berlin, Heidelberg, 2003, pp. 87-104.