Accelerated Libraries for Exascale (ALExa) and ForTrilinos

ECP-ST 2.3.3.11

Summary: The ALExa project focuses on preparing DTK and TASMANIAN for exascale platforms and integrating these libraries into ECP applications. These libraries deliver capabilities identified as needs of ECP applications: (1) the ability to transfer computed solutions between grids with differing layouts on parallel accelerated architectures (DTK), enabling simulation projects such as ExaAM to seamlessly combine results from different computational grids to perform their required simulations; and (2) the ability to construct fast and memory efficient surrogates to large scale engineering models with multiple inputs and a large number of outputs, enabling uncertainty quantification (both and forward and inverse) as well as optimization and efficient multi-physics simulations in projects such as ExaStar (Tasmanian).

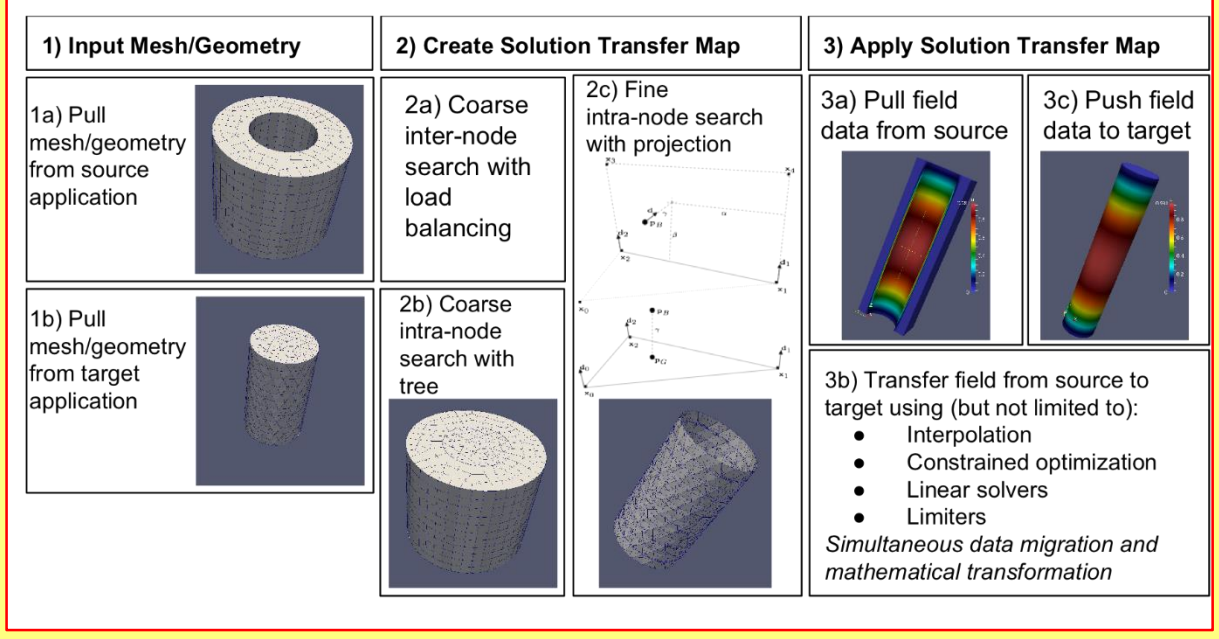
Personnel: John Turner (*PI*), Judy Hill, Wayne Joubert (Co-PIs), Stuart Slattery (*DTK*), Damien Lebrun-Grandie (*DTK/Tasmanian*), Andrey Prokopenko (*DTK/ForTrilinos*), Bruno Turcksin (*DTK*), Miroslav Stoyanov (*Tasmanian*), Seth Johnson (*ForTrilinos)*

DataTransferKit (DTK)

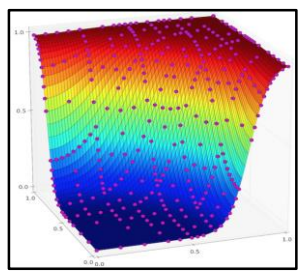
Purpose: Transfers computed solutions between grids with differing layouts on parallel accelerated architectures

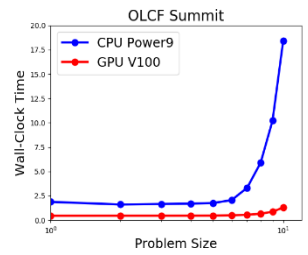
Importance: Coupled applications frequently have different grids with different parallel distributions; DTK is able to transfer solution values between these grids efficiently and accurately

Methodology: Initial setup phase to create optimal data transfer schedule between grids; transfer step to map data from source to target in a way that satisfies required mathematical properties (see diagram below)



Tasmanian





Purpose: Constructs efficient surrogate models for high dimensional problems and performs parameter calibration and optimization geared towards applications in uncertainty quantification (UQ).

Importance: UQ pertains to the statistical properties of the output of a complex model with respect to variability in multiple model inputs; large number of simulations are required to compute reliable statistics which is prohibitive for most large scale applications. A surrogate model is constructed from a moderate set of simulations using carefully chosen input values; analysis is performed on the fast and cheap surrogate.

Methodology: Toolkit for sparse grid calculations, with interfaces in C++, Python, Fortran, command line and MATLAB.

Sparse Grids capability:

- Surrogate modeling from black-box models using automated asynchronous sampling strategies in a distributed MPI environment
- GPU accelerated surrogates for coupled multi-physics simulations
- Reduced (lossy) representation of tabulated scientific data
- Data mining and manifold learning from random data

DiffeRential Evolution Adaptive Metropolis (DREAM):

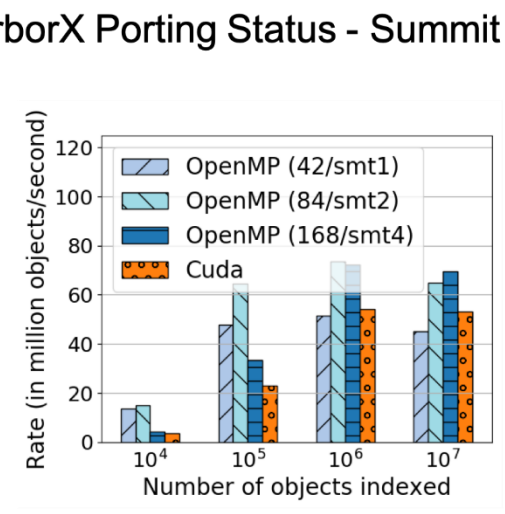
- Bayesian inference and parallel Markov Chain Monte Carlo (MCMC)
- Parameter estimation/calibration
- Global optimization and optimization under uncertainty

ArborX

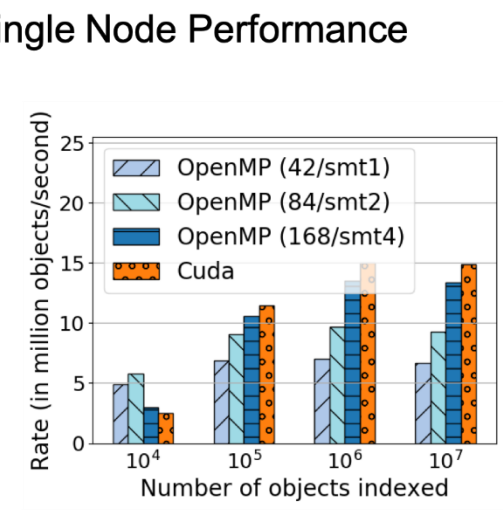
Purpose: Library designed to provide performance portable algorithms for geometric search.

Importance: Many applications require geometric search, including DTK, to define a neighborhood of objects, objects in contact, etc. Applications include multiphysics coupling, molecular dynamics and N-body simulations, and wind energy.

Methodology: Generate a multidimensional tree structure in parallel from an input set of geometric objects, perform a batched set of queries in parallel on the tree data structure to determine outcomes such as nearest neighbors or intersections.



BVH construction on one node of Summit (1 GPU)



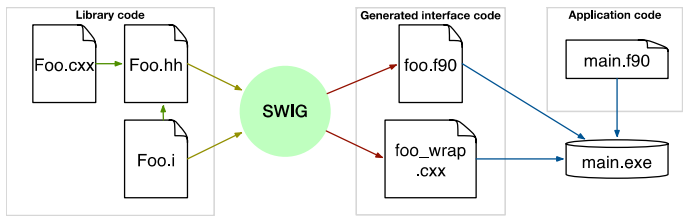
Spatial search on one node of Summit (1 GPU)

ForTrilinos

Purpose: Expose exascale-enabling C++ libraries to Fortran application codes.

Importance: Advanced GPU-and MPI-enabled numerical solvers are necessary for many apps to perform well on pre-exascale hardware and beyond. ForTrilinos exposes the Trilinos numerical solvers to Fortran applications, and its underlying technology enables other scientific and numeric libraries to be accessible to Fortran users as well.

Methodology: The SWIG-Fortran code parses Trilinos C++ header files and generates C-linkage wrapper code and Fortran-2003 interface code, mapping C++ features such as classes to their Fortran equivalents.



Trilinos features

- Define operators in Fortran as type-bound methods for inversion-of-control
- Nonlinear solvers through NOX, including Jacobian-Free Newton–Krylov (JFNK)
- Linear and eigenvalue solvers through Tpetra, including support for preconditioners

SWIG features

- Automatic conversion to native types
- Classes and inheritance
- Template instantiation
- Overloading, exceptions, and more

Flibcpp features

- Fortran interfaces to C++ standard library
- Algorithms, RNGs, vectors, sets, strings
- Native Fortran type conversion

U.S. DEPARTMENT OF ENERGY
Office of Science
EXASCALE COMPUTING PROJECT
OAK RIDGE National Laboratory