SAND2022-14591C

Sandia National Laboratories

Exceptional service in the national interest

Chapel/C++ Interoperability

Enabling Large-Scale Data Science without Costly I/O

S. Isaac Geronimo Anderson, *Daniel M. Dunlavy*

CLSAC, October 26, 2022

SAND2022-####

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. ...ation under contract DE-NA0003525.

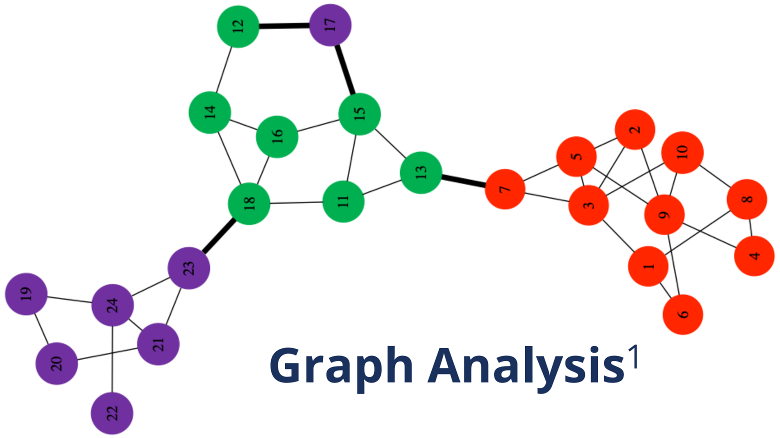
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

U.S. DEPARTMENT OF ENERGY

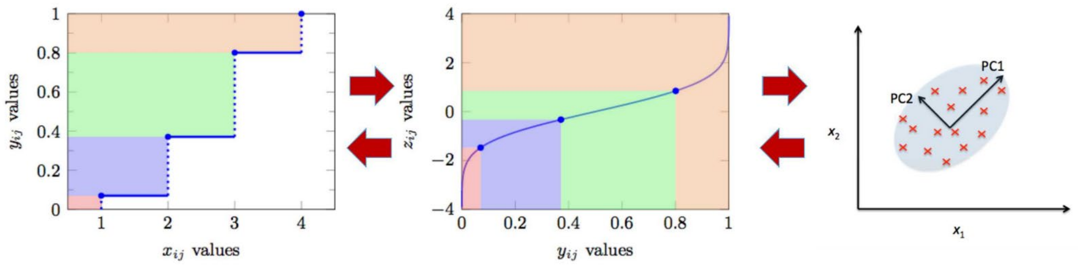
NNSA

Large-Scale Data Science Needs High Performance Computing

Motivating Problems



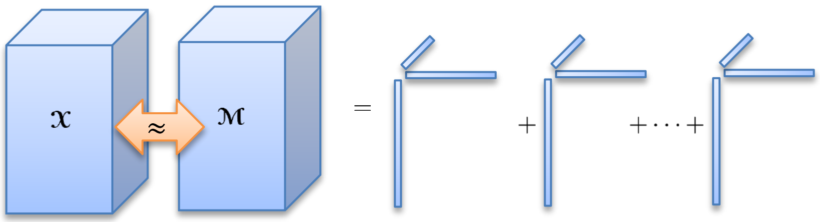
Graph Analysis[1]

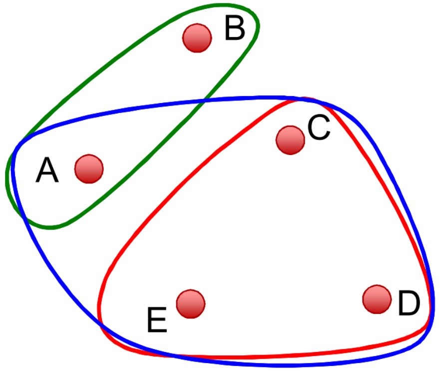


$$\ell(\Theta = UV^T\sigma, F_j|X) = \sum_{(i,j)\in\Omega} \log\left(\phi(\Phi^{-1}(F_j(X_{i,j})), \mu = \theta_{ij}, \sigma = \sigma)\right)$$

Statistical Modeling of Data with Missing Values[2]

Low-Rank Matrix and Tensor Decompositions[3]





Hypergraph Analysis[4]

[1]McCrary, *et al.* 2022; [2]Anderson-Bergman, *et al.* 2018; [3]Teranishi, *et al.* 2020; [4]Wolf, *et al.* 2016

Bridging the Gap Between Usability and Performance

Python[1] ⟷ ctypes, CFFI, PyBind11, cython, … ⟷ **C++/OpenMP/MPI**[2]

arkouda[3] ⟷ usability + parallel execution ⟷ **Chapel**[4]

[1]python.org; [2]isocpp.org, openmp.org, mpi.org; [3]bears-r-us.github.io/arkouda; [4]chapel-lang.org

Leveraging Decades of Advances in High Performance Computing

Our Motivation

arkouda | Chapel | C++/OpenMP/MPI

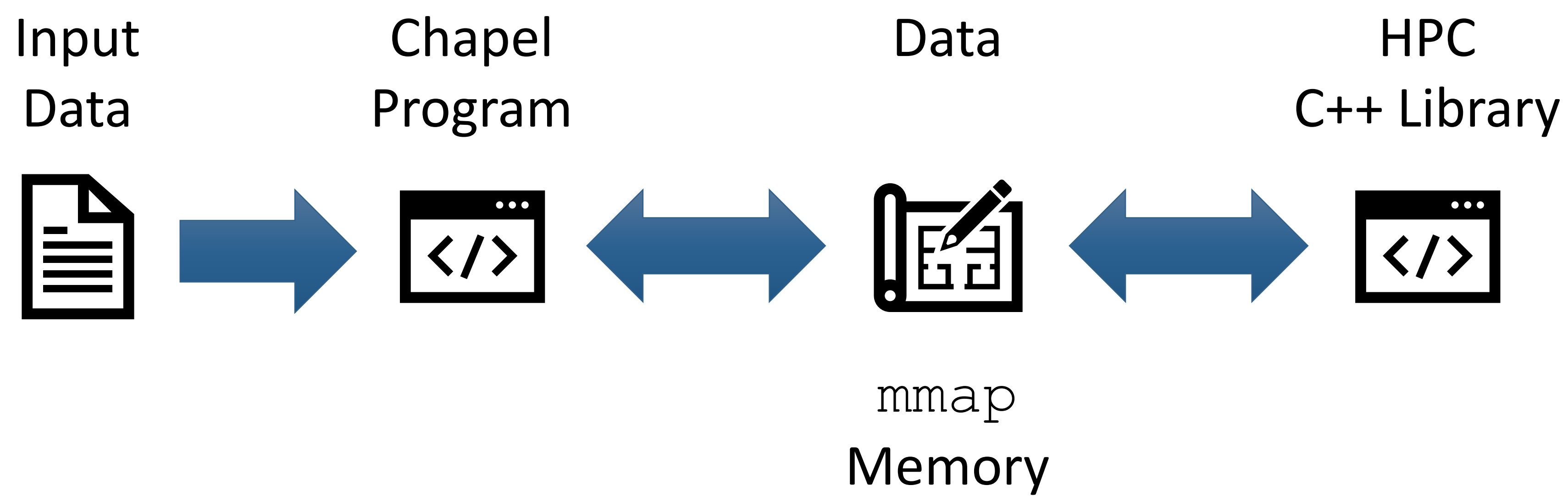
Leverage Existing High Performance Software Libraries Written in C++

Avoid Costly I/O Operations While Sharing Data Between Processes

[Others focus on direct implementation in Chapel: e.g., earlier Bader talk, CHIUW proceedings]

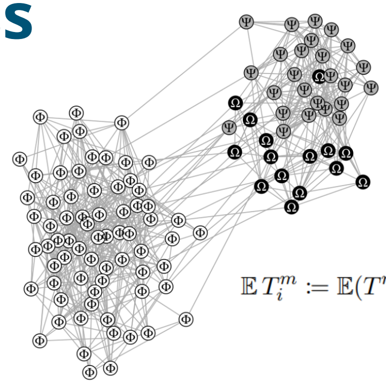
Chapel/C++ Interoperability: Avoiding Costly I/O

Chapel/C++ Interoperability using `mmap`



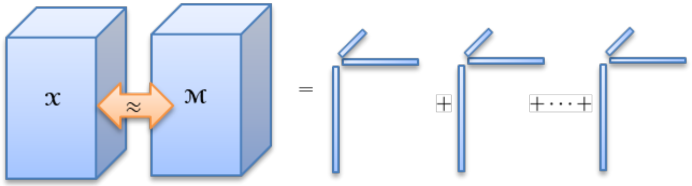
Avoids Passing Data Between Chapel and C++ Programs Via Files

Chapel/C++ Interoperability: Current Progress



$$\mathbb{E}\,T_i^m := \mathbb{E}(T^m = t \mid X_0 = i) = \sum_{t=1}^{\infty} T^m\,\mathbb{P}(T_i^m = t)$$

- **Computing Graph Hitting Time Moments**[1,2]
  - **Data:** Edge lists stored as two 1D arrays
  - **Results:** Comparable strong scaling, 64 locales (Chapel)/MPI ranks (C++)
  - **Costs:** Chapel code changes, C++ interface code, negligible `mmap` overhead



- **Computing Low-Rank Tensor Decompositions**[3, 4]
  - **Data:** sparse tensor data in coordinate format (2D indices array, 1D values array)
  - **Results:** Comparable strong scaling, 64 locales (Chapel)/MPI ranks (C++)
  - **Costs:** ~~Chapel code changes~~, C++ interface code, negligible `mmap` overhead, Chapel alignment to C++ data layout and distribution constraints

C++ HPC Codes Often Highly Optimized for Specific Data Layouts

[1]McCrary, *et al.* 2022; [2]Foss, *et al.* 2020; [3]Devine, *et al.* 2020; [4]current work

Chapel/C++ Interoperability: Next Steps

- **Distributed File Readers**
- **Broader Interoperability Support in Chapel**
- **Alternatives to `mmap`**
- **Arkouda ↔ Chapel ↔ C++ Integration**

Contact:

dmdunla@sandia.gov

References

- Anderson-Bergman, Kolda & Kincher-Winoto (2018). XPCA: Extending PCA for a Combination of Discrete and Continuous Variables, *arXiv:1808.07510.*
- Devine & Ballard (2020), GentenMPI: Distributed Memory Sparse Tensor Decomposition, Sandia National Laboratories Technical Report, SAND2020-8515.
- Foss, Lehoucq, Stuart, Tucker & Berry (2020). A Deterministic Hitting-Time Moment Approach to Seed-set Expansion over a Graph, *arXiv:2011.09544.*
- McCrary, Devine & Younge (2022). Integrating Chapel programs and MPI-Based Libraries for High-performance Graph Analysis, *Chapel Implementers and Users Workshop (CHIUW)*.
- Teranishi, Dunlavy, Myers & Barrett (2020). SparTen: Leveraging Kokkos for On-node Parallelism in a Second Order Method for Fitting Canonical Polyadic Tensor Models to Poisson Data, *IEEE High Performance Extreme Computing Conference (HPEC)*.
- Wolf, Klinvex & Dunlavy (2016). Advantages to Modeling Relational Data using Hypergraphs versus Graphs, *IEEE High Performance Extreme Computing Conference (HPEC).*