Portable Programming of High-performance Data Transformation

Marziyeh Nourian
Mentor: Dr. Andrew A. Chien

Introduction and Objectives

- Explosive rise of "big data" and data-intensive computing calls for efficient data representations. Efficient data representations allow for storage, movement, and computational efficiency.
- Conventional sequential processors and programming models are not designed for efficient data transformation. Transformation of different data representations is a key performance challenge, limiting the use of representations with expensive transformations in practice.
- We propose a computational model called extended Deterministic finite-state Transducer (DFST+) and a high-level programming model called Transducer Form (TFORM) that allows for a compact, portable and efficient implementation of data transformation.

Data Transformation Computational and Programming Model

- Extended Deterministic Finite-State Transducer (DFST+) extends DFST (traditional computational model for data transformation) with variables, actions on the variables, and transitions conditional to the variables to enable compact and efficient representations of data transformation.
- TFORM programming model enables portable expression of DFST+

Applications

- Data analytics systems: Enhancing performance of transformations exploited in Parquet (data format commonly used in data analytics systems) library
- Sparse matrix computations: Enhancing performance of transformations for different sparse representations and encodings
- Graph processing: Enhancing performance of using compressed functional tree (CTree) representation of graphs

Experimental Evaluation

- We compare performance of Parquet and SciPy (for sparse matrix transformation) library on CPU (64 cores) with CPU and UDP implementation of TFORM-based transformations

Conclusion

- We propose DFST+ (a new data transformation model), TFORM (a data transformation programming model that expresses DFST+), and TFORM VM to efficiently implement TFORM programs, enabling superior performance gains on UDP accelerator and competitive performance on CPU compared to the hand-tuned libraries.
- We exploit portable and efficient TFORM programs to unlock the power of existing and future data representations and hardware accelerators for efficient data transformation.

References