How to achieve efficient and lightweight data migration between database systems supporting diverse data models?

Background
- A part of the polystore system, BigDAWG, that tightly couples diverse databases and provides data model transparency.
- Polystore requires data migration in two ways:
  1) Transfer partial results of query executions on different database engines
  2) Migrate data when the workload changes in order to achieve improved performance

Motivation
- Need to transfer data between many database systems for optimized data placement
- Efficient data migration to take advantage of superior data processing in specialized database systems

Current issues
- CSV based data migration is expensive because of parsing and deserialization
- Binary format is not always compact
- Parallelism is hard to exploit (file division)

Keys to Performance
- Parallel migration (if possible)
- Migration in binary format (if supported)
- Adaptive compression for data transfer via network
- Advancement in hardware: SIMD and RDMA

Main goals
- Efficient and adaptive data migration framework for the BigDAWG system
- Flexible and highly performant tool for data migration between many databases

System overview

3 Approaches to Data Migration

Data Migration Framework

Results

Breakdown migration from PostgreSQL to SciDB

Usage of resources on the SciDB end for CSV loading

Adaptive migration from PostgreSQL to Accumulo (TPC-H data)