List of Project Topics (Proposals)

Last update: Apr 14, 2025

LDE Projects in the Context of Our Two Open-source Systems

- **DAPHNE EU-project**
  https://github.com/daphne-eu/daphne
  - Focus on integrated data analysis pipelines
  - Project implementation mainly in C++

- **Apache SystemDS**
  https://github.com/apache/systemds
  - Focus on the end-to-end data science lifecycle
  - Project implementation mainly in Java and DML

Topics in DAPHNE

Implementation mainly in C++ and DaphneDSL, some also in Python

https://github.com/daphne-eu/daphne/issues?q=is%3Aissue%20state%3Aopen%20label%3A%22LDE%20summer%202025%22

#955 Locality Sensitive Hashing (LSH)

- **Motivation**
  - LSH is a powerful technique for efficient approximate nearest neighbor search in high-dimensional spaces.
  - Unlike traditional hash functions that aim to minimize collisions, LSH is designed so that similar items have a higher probability of colliding.
  - This property makes LSH invaluable for many data-intensive applications including recommendation systems, duplicate detection, and clustering large datasets.
- **Task** (in C++ and DaphneDSL)
  - Design and implement a LSH algorithm in DaphneDSL.
  - Explore different variants of LSH (e.g., MinHash, SimHash, random projections)
  - Create user-friendly functions that integrate well with DAPHNE's existing APIs.
  - Demonstrate the effectiveness of LSH through example applications.
- **More Information & Hints**
  - https://github.com/daphne-eu/daphne/issues/955
  - Contact: Philipp Ortner

#956 Large Language Model (GPT-2) Inference

- **Motivation**
  - Large Language Models (LLMs) are enabling hitherto unknown applications.
  - GPT-2 is a transformer-based language model that can generate coherent text based on a given prompt.
- **Task** (in C++, DaphneDSL/Python)
  - Implement GPT-2 inference (forward pass) in DAPHNE, to this end:
    - Understand the architecture and operation of transformer-based language models.
    - Implement required and missing operators in the DAPHNE system (C++).
    - Implement required and missing library functions (either Python+DaphneLib or DaphneDSL).
    - Performance experiments may investigate, e.g., the generated tokens/second.
  - Groups may implement additional optimizations beyond the basic functionality and add more models.
- **More Information & Hints**
  - https://github.com/daphne-eu/daphne/issues/956
  - Contact: Philipp Ortner

#957 Efficient Sparse Data Transfer between DAPHNE and Python Libraries

- **Motivation**
  - Sparse matrices containing mostly zeros is commonplace in many applications (e.g., graph processing).
  - Various efficient physical representations for sparse data (e.g., CSR, CSC, COO)
  - DAPHNE needs to integrate with the existing Python-based data science ecosystem to achieve user adoption.
  - Efficient data transfer is key and DAPHNE partly supports it, but sparse representations are still missing.
- **Task** (in C++ and Python)
  - Implement efficient (ideally zero-copy) bidirectional data transfer of sparse data between DAPHNE and famous Python libraries like SciPy, TensorFlow, and PyTorch.
  - Extend the existing data transfer for dense numpy arrays by support sparse data structures.
  - Efficiently handle data and value type conversions between different physical matrix representations.
  - Showcase the efficient data transfer by offloading computations from Python to DAPHNE
- **More Information & Hints**
  - https://github.com/daphne-eu/daphne/issues/957
  - Contact: Patrick Damme

#512 Simplification Rewrites for Linear and Relational Algebra

- **Motivation**
  - User programs/queries are typically parsed into an initial, unoptimized internal representation (IR).
  - This initial IR usually offers large potential for performance optimization through simplification rewrites, which reorder, remove, or replace operations when certain patterns are detected (e.g., `t(t(X)) -> X`).
  - The smart application of simplification rewrites can increase performance and reduce memory consumption by orders of magnitude; but so far, DAPHNE has only limited support for simplification rewrites.
- **Task** (in C++)
  - Design a simplification rewrite system as a part of DAPHNE's MLIR-based optimizing compiler.
  - Implement a good set of useful static and dynamic rewrites for linear and relational algebra.
- **More Information & Hints**
  - https://github.com/daphne-eu/daphne/issues/512
  - Contact: Patrick Damme

#511 Efficient Parallel Hash-Join Operator for Speeding up the SSB

- **Motivation**
  - The Star Schema Benchmark (SSB) is a well-known benchmark for analytical query processing.
  - The runtime of most SSB queries is dominated by PK-FK joins and semi-joins.
  - An efficient join implementation is crucial for achieving good results in this benchmark.
- **Task** (in C++)
  - Implement an efficient parallel hash-join (separate build and probe) operator on columnar data.
  - Devise an efficient hash table implementation and support it for intermediate results in DaphneIR.
  - Parallelism should be achieved through multiple threads (MIMD), and optionally also through SIMD operations.
  - In a first step, multi-threading could be applied inside the operator; based upon that, an integration with DAPHNE's vectorized engine could be tackled.
- **More Information & Hints**
  - https://github.com/daphne-eu/daphne/issues/511
  - Contact: Patrick Damme

#521 Efficient Matrix Multiplication for Generic Value Types

- **Motivation**
  - Matrix multiplication is a central operation in many machine learning and data analysis algorithms.
  - Various libraries (e.g., BLAS) provide highly optimized implementations, but are typically limited to certain data and value types, especially dense matrices of single/double-precision floating point values.
  - DAPHNE strives to be extensible w.r.t. to data and value types, thus it needs efficient matrix multiplications for other types, too.
- **Task** (in C++)
  - Implement efficient matrix multiplication for various combinations of dense/sparse inputs/output, different values types (e.g., integers, bool), and input shapes (e.g., matrix-matrix and matrix-vector).
  - On the one hand, write hand-tuned kernels for a handful of cases
  - On the other hand, devise a generic implementation that comes as close as possible to these specialized ones.
  - Showcase the benefit of your kernels on ML algorithms dominated by matrix multiplications.
- **More Information & Hints**
  - https://github.com/daphne-eu/daphne/issues/521
  - Contact: Patrick Damme

#690 IDE/Tooling Support for DaphneDSL and DaphneIR

- **Motivation**
  - DaphneDSL is DAPHNE's domain-specific language for integrated data analysis pipelines.
  - Its syntax is inspired by languages like Python, R, and C.
  - DaphneDSL can be written in any text editor, but support in an integrated development environment would increase user productivity.
- **Task**
  - Implement support for DaphneDSL in a widely-used IDE (preferably VS Code), including a LSP and TreeSitter.
  - The tool should be connected to the DAPHNE compiler, especially to its features for type/shape/property inference in order to augment the DaphneDSL code with additional information
- **More information & hints**
  - https://github.com/daphne-eu/daphne/issues/690
  - Contact: Philipp Ortner

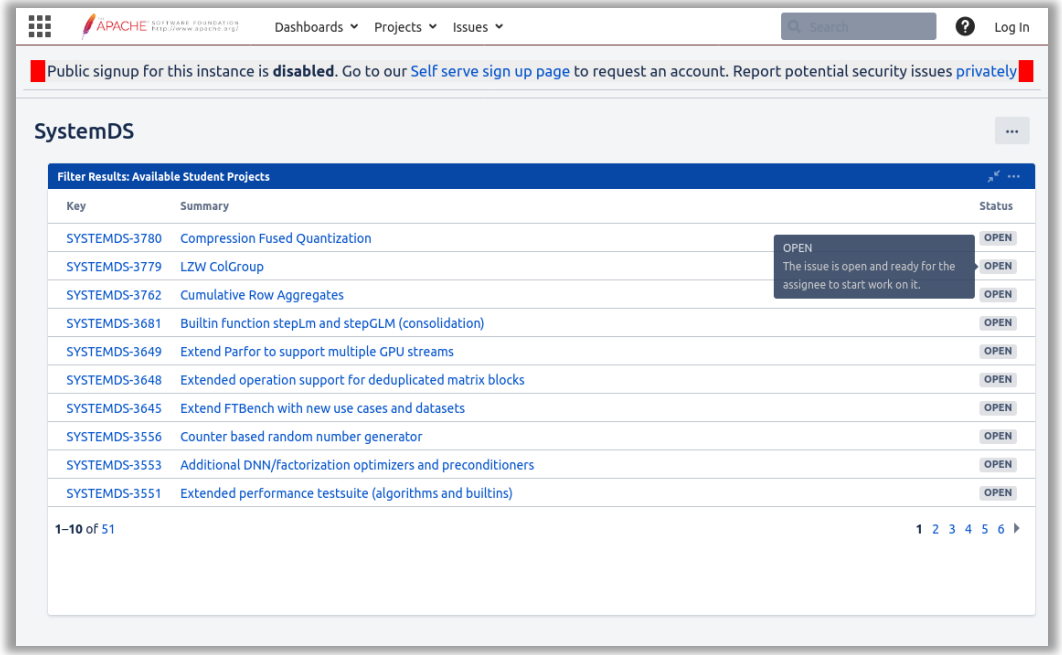
Topics in Apache SystemDS

Implementation mainly in Java and DML

https://issues.apache.org/jira/secure/Dashboard.jspa?selectPageId=12335852#Filter-Results/12365413

Topics on Apache SystemDS

- **See the Full List of Available Student Projects:**
  - https://issues.apache.org/jira/secure/Dashboard.jspa?selectPageId=12335852#Filter-Results/12365413



- **Topics in the Context of, e.g.**
  - Compression
  - Feature Transformation
  - New operations, new algorithms
  - Compiler rewrites
  - Tooling (e.g., performance tests)
  - Python API
  - …
- **More Information & Hints**
  - Contact: Matthias Boehm

Alternative: Propose Your Own Topic Idea

- **We are open to additional topic proposals**
  - In the context of data engineering, data management, and machine learning systems
  - If you are passionate about your idea
  - More topics in SystemsDS and DAPHNE or other open-source systems possible,
    but contributions might be more difficult to get accepted
  - **If you would like to propose your own topic**, **approach me by email by May 02, 23:59 CEST**;
    in any case, **also fill in the poll regarding the topic selection with your preferred topics from the list above**