D1.3 First Annual Project Report

DAPHNE

Integrated Data Analysis Pipelines for Large-Scale Data Management, HPC, and Machine Learning

Version 1.1

PUBLIC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957407.

Document Description

This report summarizes the work carried out throughout the first year of the DAPHNE project, as well as the progress and initial results obtained so far. In detail, we share the overall work and progress towards the objectives, status of groups of work packages, and impact. Additionally, this report also includes an update of the exploitation and dissemination plan, as well as the research data management plan.

D1.3 First Annual Project Report			
WP1 – Project Management			
Type of document	R	Version	1.1
Dissemination level	PU		
Lead partner	KNOW		
Author(s)	Matthias Boehm (KNOW)		
Reviewer(s)	All project partners		

Revision History

Version	Revisions and Comments	Author / Reviewer
V1.0	Summary of work and progress of the first project year.	Matthias Boehm
V1.1	Incorporated comments and suggestions by Florina, Pınar, Patrick, Tilmann, and Eva	Matthias Boehm

1 Overview of Work Performed and Progress

Summarizing a discussion of the advisory board during the first general assembly meeting at the end of October 2021, the overall progress of the DAPHNE project is **good** and in line with the grant agreement, but of course there is room for improvement. The key targets and results are accomplished, the work packages are on track, the paper on the overall DAPHNE system architecture has been accepted, and the minor issues (e.g., meeting frequency, prototyping effort, better integration of use cases and benchmarking) can be addressed in the second year. In this section, we describe in more detail the progress with regard to the objectives, the individual work packages, and how it relates to the expected impact.

1.1 Objectives

The overall objective of the DAPHNE project is to define and build an open and extensible system infrastructure for integrated data analysis pipelines of data management and query processing, high-performance computing (HPC), and ML training and scoring in order to address productivity, utilization, and hardware challenges.

Strategic Objectives: The original work plan of the grant agreement defined the following three strategic objectives.

- ***Objective 1 System Architecture, APIs and DSL:*** Improve the productivity for developing integrated data analysis pipelines via appropriate APIs and a domain-specific language, an overall system architecture for seamless integration with existing data processing frameworks, HPC libraries, and ML systems. A major goal is an open, extensible reference implementation of the necessary compiler and runtime infrastructure to simplify the integration of current and future state-of-the-art methods.
- ***Objective 2 Hierarchical Scheduling and Task Planning:*** Improve the utilization of existing computing clusters, multiple heterogeneous hardware devices, and capabilities of modern storage and memory technologies through improved scheduling as well as static (compile time) task planning. In this context, we also aim to automatically leverage interesting data characteristics such as sorting order, degree of redundancy, and matrix/tensor sparsity.
- ***Objective 3 Use Cases and Benchmarking:*** The technological results will be evaluated on a variety of real-world use cases and datasets as well as a new benchmark developed as part of the DAPHNE project. We aim to improve the accuracy and runtime of real-world use cases combining data management, machine learning, and HPC – this exploratory analysis serves as a qualitative study on productivity improvements (Objective 1). The variety of real-world use cases will further be generalized to a benchmark for integrated data analysis pipelines quantifying the progress compared to state-of-the-art (Objective 2).

Progress on Strategic Objectives: After an orientation phase including initial discussions and feasibility studies, the project consortium started already in February 2021 building an MLIR-based prototype of the DAPHNE system architecture, DSL, and related APIs. Partners in WP 2 and 3 defined the overall architecture, language abstractions, and optimizing compiler; Work packages 3, 4, and 5 designed and implemented initial runtime kernels for local and distributed computation, as well as a tiled (vectorized) execution engine, which allows operator fusion and different task scheduling strategies; and WP 6 and 7 contributed GPU and FPGA kernels and their integration, as well as supporting surveys and research on related aspects of HW accelerators and computational storage. Furthermore, WP 8 and 9 conducted a broad survey of DM, HPC, and ML benchmarks, and described initial reference implementations of the different use cases. To summarize, we made good progress on all three strategic objectives and set the foundation (prototype and use case pipelines) for building out the infrastructure and use case implementation on this infrastructure in the following project years.

Technical Objectives: Besides the strategic objectives, the work plan also identified the following technical areas that need technological advancements beyond state of the art.

- ***Seamless Integration of DM, HPC, ML Systems***
- ***Intermediate Representation and Systematic Lowering***
- ***Holistic Optimization of Integrated Data Analysis Pipelines***
- ***Code Generation for Sparsity Exploitation***
- ***Hierarchical Scheduling and HW Exploitation***
- ***Managed Storage Tiers and HW Acceleration***
- ***Systematic Exploitation of Data Characteristics***

Progress on Technical Objectives: So far, the detailed research efforts on the mentioned technical objectives are rather limited, but we built the initial system infrastructure around DaphneIR as intermediate representation with systematic lowering and the prototype can already execute runtime plans that comprise query processing (DM) and numerical computation (HPC, ML). At the same time, this infrastructure was defined in a way that allows simple integration of new optimization passes (including the propagation and exploitation of data characteristics), code generation for fused operator pipelines, and hierarchical scheduling. In the next years, we will address the individual technical objectives in much more detail.

1.2 Work Packages

In order to allow effective communication and reduce redundancy, we organized the regular work-package-related meetings and activities in groups of two work packages that are closely related in terms of technical content. Accordingly, we report on the work carried out in individual work packages and tasks in relation to these groups as well.

1.2.1 Work Packages 1 and 10 (Management)

Project Management: Work package 1 and partially 10 (both led by KNOW) covered all project management and external communication activities. Besides the organization, preparation, and running of weekly (and later biweekly) all-hands meetings, in the first year, these activities included especially the setup of the entire project infrastructure (mailing lists, Nextcloud instance, data sharing, Gitlab repositories, initial website, twitter account, deliverable templates, slide templates, video conferencing). These aspects have been documented in project and risk management plan [D1.1] and the research data management plan [D1.2]. Furthermore, we organized the kickoff meeting in 12/2020 and the first general assembly meeting in 10/2021, which where both held virtually due to the COVID pandemic. The initial website was a placeholder and supposed to be replaced by a new website with more information until Milestone MS1 05/2021. Due to an extended sick leave of the project coordinator and delayed replacement in 09/2021, the new website got delayed as well. We are currently working on finalizing this website by 11/2021 and will upload all public talks, deliverables, and papers there. Work package 1 also conducted project monitoring and reporting based on internal partner reports every three months.

Dissemination and Exploitation: Besides papers and talks related to the individual technical work packages, we conducted various dissemination and exploitation activities representing the DAPHNE overall project. The following list summarizes papers and talks that presented the initial system architecture and experiments, as well as use cases and project idea.

- Patrick Damme et al: DAPHNE: An Open and Extensible System Infrastructure for Integrated Data Analysis Pipelines, CIDR 2022 [D+22].
- Matthias Boehm: DAPHNE: Integrated Data Analysis Pipelines for Large-Scale Data Management, HPC, and Machine Learning, BDVA DataWeek, 05/2021.

1.2.2 Work Packages 2 and 3 (System Architecture and DSL)

Requirements and System Architecture: After many discussions with the different partners and analysis of the use cases, and broader trends towards integrated data analysis pipelines, in work packages 2 and 3, we then summarized the requirements on an open and extensible system infrastructure, defined the overall system architecture in terms of supported data types, operations, and system internal components. This system architecture was documented in Deliverable D2.1 [D2.1] and formed the basis, together with initial experiments for selected pipelines, for a joint CIDR 2022 paper [D+22].

Language Abstractions, DSL, APIs: In more detail, we also defined the DAPHNE language abstractions in the form of DaphneDSL (a domain-specific language for integrated data analysis pipelines), and DaphneAPI (a Python API with lazy evaluation as an additional entry point to simplify adoption). These language abstractions have been described with sufficient detail of underlying design principles, supported data types and operations, scoping and polymorphism, control flow, and means of configuration/extensibility, in Deliverable D3.1 [D3.1]. In this context, we also described already the initial design of the MLIR-based optimizing compiler, DaphneIR as the central intermediate representation of the compiler, and future extensions by higher-level built-in operations. For the camera-ready version of the CIDR 2022 paper, we intend to incorporate these abstractions for a holistic description of DAPHNE.

DAPHNE Prototype: Since February 2021, we actively develop an initial prototype of the DAPHNE system including the parser, compiler, and runtime as well as build scripts and continuously growing test framework. The source code and artifacts as well as development tasks are managed in a private Gitlab repository hosted on premises of KNOW. Towards the end of end of the first project year, we see increasing contributions by many project partners (317 commits from 13 unique contributors over the first year), which go through a pull-request/ review/merge process in order to ensure high quality. Meanwhile, the prototype has all basic infrastructure in place and thus, allows for use and extension in individual research activities and student programming projects.

1.2.3 Work Packages 4 and 5 (Runtime and Scheduling)

The joint discussions on work packages 4 and 5 (which closely relate to work package 2 and 3) combined both, (1) knowledge sharing of selected aspects of runtime and scheduling techniques in query processing, HPC, and ML systems by the involved partners, as well as (2) in-depth design discussions of runtime aspects of the initial prototype and future extensions.

Local Runtime: Initial efforts on building the DAPHNE runtime system centered around the discussion and implementation of core data structures such as frames, dense and sparse matrices, as well as a variety of kernels, in terms of concrete operator implementations for instantiated inputs. These kernels include I/O operations for specific data formats, variants of linear algebra operations, variants of relational algebra operations, as well as various aggregations, reorganizations and statistical functions. In order to allow for operator fusion, flexible multi-device utilization, and a central scheduling decisions, we introduced a tiled (vectorized) execution engine that processes in a task-based manner operator pipelines on tiles of inputs. The detailed design is covered in deliverables on the system architecture [D2.1], language abstractions and compilation [D3.1], and the initial DSL runtime design [D4.1].

Distributed Runtime: Beyond the local runtime design, we also created an initial design and implementation of the distributed runtime system. The two distributed matrix/frame representations are (1) federated data objects (where the coordinator keeps detailed metadata where which piece of a matrix or frame resides), and (2) distributed collections of fixed-size tiles (where map/reduce-style data-parallel computation strategies apply). Our initial prototype of the distributed runtime uses federated data objects and applies local, multi-threaded operations (via the tiled execution engine) at the distributed workers. Additionally, we internally provide a set of distribution primitives such as `distribute(X)` for creating a distributed matrix from a local matrix, `broadcast(X)` to send a local matrix to all workers, and `collect()` to obtain partial results from the different workers. Handles to such distributed data objects are kept in metadata objects of the different data structures at the coordinator allowing a creation on demand before spawning distributed jobs. The overall design is described in deliverables on the system architecture [D2.1], and the initial DSL runtime design [D4.1]. Additional work outside the prototype infrastructure also investigates additional distribution primitives, collective operations, parameter servers, and similar distribution strategies.

Scheduling: Hierarchical scheduling and task planning is a strategic objective of the DAPHNE project. Besides initial discussions on terminology, deployment environments, and workloads, efforts in work package 5 focused on a requirements analysis from use case ML pipelines, over task-parallel loops and operator pipelines, to data-parallel processing of fused operator pipelines and kernels. In this context, we already explored and integrated various task scheduling (partitioning) strategies such as static and dynamic self-scheduling schemes in the tiled execution engine. The details are described in the deliverables on system architecture [D2.1], initial DSL runtime design [D4.1], and scheduler design for pipelines and tasks [D5.1].

1.2.4 Work Packages 6 and 7 (Computational Storage & HW Accelerators)

Similar to the other groups of work packages, also work package 6 (on computational storage) and 7 (HW accelerators) form a natural group with partial overlap and potential synergies. Besides, knowledge sharing between the involved partners (ITU, TUD, KNOW, Intel and DLR) and previous projects, initial work of the first year covered basic I/O support for selected data formats, an analysis of the design space and current technology trends, as well as an initial integration of GPU and FPGA operations, related data placement primitives, and even custom, tailor-made device kernels for selected operations.

GPU and FPGA Integration: The integration of GPU and FPGA accelerators is important for performance of various end-to-end integrated data analysis pipelines, and at the same time, serve as examples of integrating other types of HW accelerators through means of extensibility (see work packages 2 and 3). In the first year, we designed the general integration of HW accelerators by example of GPUs. This integration includes kernels for context creation and device initialization, data representations and data handles through meta data objects, and common GPU operations for linear algebra, aggregations, and selected deep neural network operations. Furthermore, we devised specialized kernels for FPGAs (e.g., quantization) and are currently in progress of integrating a broader set of FPGA kernels. Both GPUs and FPGAs will also become part of the tiled execution engine in order to simultaneously exploit CPUs, GPUs, and FPGAs for fused operator pipelines.

Other Work: Related to work package 6 and 7, more specialized work is currently conducted outside the scope of the DAPHNE prototype with the intend of later integration. Examples include virtual vector abstractions for SIMD exploitation, assembling computational storage hardware platforms and initial experimentation, exploration of abstractions for complex storage hierarchies, and performance models for heterogeneous hardware devices.

1.2.5 Work Packages 8 and 9 (Use Cases and Benchmarks)

Use Cases: The work packages 8 and 9 created common terminology and a joint understanding of requirements via weekly and later biweekly meetings for in-depth discussions of the individual use cases (including knowledge sharing between partners), the use case descriptions, and ML pipeline implementations. A major outcome of these discussions are the use case pipelines documented in Deliverable D8.1 [D8.1] which serve as example use cases for the DAPHNE system infrastructure and real-world benchmarks. During these discussions, we already identified future work for improvements of the individual pipelines and relevant measurements to quantify the use case improvements achieved through DAPHNE.

Benchmarks: Complementary to the real-world use cases, which act as grounding and benchmarks for work packages 2-7, work package 9 broadly surveyed existing benchmarks in databases, large-scale data-parallel computation, high-performance computation, and ML systems. This survey was documented in Deliverable D9.1 [D9.1]. Additionally, the partner HPI made major contributions to the development of the TPCx-AI benchmark, which was released with reference implementation kits in 09/2021 [TPC21]. Related to benchmarking ML pipelines HPI also conducted projects with a team of bachelor students towards a benchmark framework for integrated data analysis pipelines, tested with a subset of datasets from DAPHNE use cases.

1.3 Impact

During the first year, we developed – through extensive discussions and prototyping – the initial design and system architecture of the open and extensible DAPHNE system infrastructure. This prototype sets the foundation for broader impact in the next years. In the following, we additionally comment on the expected impacts.

Impact 1: *"Increased productivity and quality of system design and software development thanks to better methods, architectures and tools for complex federated/distributed systems handling extremely large volumes and streams of data" (Horizon 2020 Work Programme 2018-2020, Part 5.i - Page 49)*

The strategic objective 1 on system architecture, APIs, and DSL directly addresses the challenge of increasing the productivity of developing and deploying integrated data analysis pipelines. Besides the initial prototypes of DaphneLib and DaphneDSL, we work towards implementing the use case pipelines in this DSL and its higher-level primitives (e.g., SQL query processing, data engineering, ML algorithms, and numerical computation). So far, the key performance indicators (KPIs) K1-K3 have not been addressed directly, but for K2 (8 developed IDA pipelines) we have 5 use case pipelines implemented in state-of-the-art baseline systems, and 4 micro IDA pipelines (including query processing and linear regression, CNN classification, K-Means clustering, and graph processing) for continuous experiments.

Impact 2: *"Demonstrated, significant increase of speed of data throughput and access, as measured against relevant, industry-validated benchmarks" (Horizon 2020 Work Programme 2018-2020, Part 5.i - Page 49)*

Effective utilization of heterogeneous hardware, elimination of unnecessary overheads (e.g., data transfer), and holistic optimization of IDA pipelines is a central theme of building the DAPHNE system infrastructure. Initial experiments with our current prototype already showed moderate improvements of up to 2x compared to state-of-the-art systems such as TensorFlow as an ML system, and MonetDB/DuckDB as analytical and embedded database systems. KPIs K4 and K5 aim at robust average improvements of 4x and increased utilization by 1.5x, which we aim to achieve through better utilization of heterogeneous hardware, and joint optimization of complex IDA pipelines. This goal is the core of research in the technical work packages throughout the next years.

Impact 3: *"Demonstrated adoption of results of the extreme-scale analysis and prediction in decision-making, including AI (in industry and/or society)" (Horizon 2020 Work Programme 2018-2020, Part 5.i - Page 49)*

The DAPHNE prototype includes an initial distributed runtime backend and we already conducted initial experiments on different clusters and the Vega Supercomputer. However, the development of the distributed backend is still in an early stage. Accordingly, there is limited impact regarding K6 and K7 that aim to enable 6 out of 8 use case IDA pipelines to exploit data-parallel computation, HPC, and distributed ML training and scoring, as well as reductions of prediction errors due to full-scale data processing and low-effort IDA pipeline refinements. These improvements have additional potential for positive impact on industry and society.

Impact 4: *Open and Extensible System Infrastructure (Additional Substantial Impacts Not Mentioned in the Work Program)*

Developing DAPHNE as an open and extensible system infrastructure, with extensibility in mind from the beginning, is a chance for academia and industry for reusing this infrastructure in their own research, experiments, and products without significant upfront investment. In the technical work packages, we aim to make baseline comparisons (e.g., scheduling algorithms and kernel implementations) also available through these extensibility mechanisms.

1.4 Access to Research Infrastructure

During the first project year, no formal access provisioning activities to research infrastructures were conducted under the grant. However, there was a healthy reuse of pre-existing Research & Development infrastructure and sharing among the partners:

- Reuse of pre-existing R&D infrastructure (compute clusters and HW accelerators) by the individual partners
- Loan agreement between INTP and KNOW for an Intel FPGA PAC D5005 with Stratix 10SX FPGA and 32GB device memory for integration experiments with CPUs, GPUs, and FPGAs at KNOW (coordinator)
- Setup of specialized hardware (funded under the grant agreement) for experiments with computational storage (e.g., DAISY OpenSSD with Xilinx Zynq Ultrascale+ZU17EG FPGA and ARM Cortex-A53 CPU at ITU)
- Access to the Vega Supercomputer at UMAR through the EuroHPC JU Benchmark and Development Access Calls[1]

2 Update of Exploitation and Dissemination Plan

Overall Plan: The original dissemination and exploitation plan is still valid and we will materialize a refined dissemination plan in Deliverable D10.1 by M18, and report on community building in Deliverable D10.2 by M36. Our current efforts focus on dissemination and exploitation via publications and talks, as well as contributions to benchmarks for facilitating exchange and interaction among stakeholders of the DM, HPC, and ML systems communities. A description of the DAPHNE system architecture has been accepted at CIDR 2022 (DM) and we also submitted a contribution to HiPEACinfo 65 (HPC).

Open Source Release: Our dissemination efforts focus on system and data engineers as well as application users via a strategy of (1) creating value by improved productivity, better resource utilization, and less overhead, (2) open sourcing the DAPHNE system infrastructure for system researchers and early users, (3) sharing benchmarks and artifacts for reproducibility and focus on extensibility for early adoption by researchers, (4) reaching out to potential users, and (5) carefully considering feedback, and feature requests to ground our research efforts. Originally, the open source release of the DAPHNE reference implementation was planned for Deliverable D10.3 by M42. Having built the basic DAPHNE prototype as a foundation, we

[1] https://prace-ri.eu/hpc-access/eurohpc-access/eurohpc-ju-benchmark-and-development-access-calls/

currently plan of having the initial open source release already in early 2022 in order to support open science, early adoption and feedback, and broader impact.

3 Update of Data Management Plan

The initial data management plan has been described in detail in Deliverable 1.2 [D1.2], which is a living document and updated project-internally on demand. In general, the project consortium is committed to open science and reproducible research, making publications, reports, datasets, artifacts, and experimental results, publicly available by default.

4 References

[D1.1] DAPHNE: D1.1 Project and Risk Management Plan, EU Project Deliverable, 03/2021

[D1.2] DAPHNE: D1.2 Research Data Management Plan, EU Project Deliverable, 05/2021

[D2.1] DAPHNE: D2.1 Initial System Architecture, EU Project Deliverable, 08/2021

[D3.1] DAPHNE: D3.1 Language Design Specification, EU Project Deliverable, 11/2021

[D4.1] DAPHNE: D4.1 DSL Runtime Design, EU Project Deliverable, 11/2021

[D5.1] DAPHNE: D5.1 Scheduler Design for Pipelines and Tasks, EU Project Deliverable, 11/2021

[D6.1] DAPHNE: D6.1 Report on Search Space Analysis, Automatic Capability Configuration, EU Project Deliverable, 11/2021

[D8.1] DAPHNE: D8.1 Initial Pipeline Definition all Use Cases, EU Project Deliverable, 08/2021

[D9.1] DAPHNE: D9.1 SotA Survey of Benchmarks from DM, HPC, and ML Sys, EU Project Deliverable, 11/2021

[D+22] Patrick Damme, Marius Birkenbach, Constantinos Bitsakos, Matthias Boehm, Philippe Bonnet, Florina Ciorba, Mark Dokter, Pawel Dowgiallo, Ahmed Eleliemy, Christian Faerber, Georgios Goumas, Dirk Habich, Niclas Hedam, Marlies Hofer, Wenjun Huang, Kevin Innerebner, Vasileios Karakostas, Roman Kern, Tomaž Kosar, Alexander Krause, Daniel Krems, Andreas Laber, Wolfgang Lehner, Eric Mier, Marcus Paradies, Bernhard Peischl, Gabrielle Poerwawinata, Stratos Psomadakis, Tilmann Rabl, Piotr Ratuszniak, Pedro Silva, Nikolai Skuppin, Andreas Starzacher, Benjamin Steinwender, Ilin Tolovski, Pınar Tözün, Wojciech Ulatowski, Aristotelis Vontzalidis, Yuanyuan Wang, Izajasz Wrosz, Aleš Zamuda, Ce Zhang, Xiao Xiang Zhu: DAPHNE: An Open and Extensible System Infrastructure for Integrated Data Analysis Pipelines. **CIDR 2022**.

[TPC21] TPC: TPC Express AI (TPCx-AI) Standard Specification Revision 1.0.0 http://tpc.org/TPC_Documents_Current_Versions/pdf/TPCX-AI_v1.0.0.pdf