D1.5

3rd Annual Project Report

DAPHNE

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

Version 1.4

PUBLIC

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

Document Description

In D1.5 DAPHNE project team describes the progress made until project month 36 and here particularly the work done in project year 3 (M25/Dec 2021 – M36/Dec 2022). This report presents an overview of the type and purpose of the document, its revision history, the strategic objectives of DAPHNE project and the work carried out in project year 3 to reach these objectives. Then, a more detailed description concerning work done in the 3rd project year across all work packages (WPs) is provided. To round off, this report presents a brief outlook addressing the next steps.

D1.5 3rd Annual Project Report			
WP1 – Project Management			
Type of document	R	Version	1.4
Dissemination level	PU	Project month	36
Lead partner	KNOW		
Authors	Mark Dokter (KNOW), Eva Paulusberger (KNOW), Patrick Damme (TUB), Nils Strassenburg (HPI), Ilin Tolovski (HPI), Tilmann Rabl (HPI), Andreas Laber (IFAT), Philippe Bonnet (ITU), Benjamin Steinwender (KAI), Dirk Habich (TUD), Florina M. Ciorba (UNIBAS), Jonas Korndorfer (UNIBAS), Aleš Zamuda (UM), Dimitrios Tsoumakos (ICCS)		
Reviewers	Bernhard Peischl (AVL), Gustavo Alonso (ETH)		

Revision History

Version	Revisions and Comments	Author / Reviewer
V1.0	Initial draft and structure	Eva Paulusberger (KNOW)
V1.1	Text sections by responsible partners	Mark Dokter (KNOW), Eva Paulusberger (KNOW), Patrick Damme (TUB), Nils Strassenburg (HPI), Ilin Tolovski (HPI), Tilmann Rabl (HPI), Andreas Laber (IFAT), Philippe Bonnet (ITU), Benjamin Steinwender (KAI), Dirk Habich (TUD), Florina M. Ciorba (UNIBAS), Jonas Korndorfer (UNIBAS), Aleš Zamuda (UM), Dimitrios Tsoumakos (ICCS)
V1.2	Semantic adaptations	Eva Paulusberger (KNOW)
V1.3	Reviews 1 + 2	Bernhard Peischl (AVL), Gustavo Alonso (ETH)
V1.4	Finalization	Eva Paulusberger (KNOW)

Abbreviations

Abbreviated Term	Full Term
API	Application Programming Interface
BoD	Board of Directors
CUDA	Compute Unified Device Architecture
D	Deliverable
DaphneLib	Python API of DAPHNE
DoA	Description of Actions
DSL	Domain-specific Language
(e)BPF	Berkeley Packet Filter
FPGA	Field Programmable Gate Array
GPU	Graphics Processing Unit
HPC	High Performance Computing
IR	Intermediate Representation
M	Month
ML	Machine Learning
MLIR	Multi-Level Intermediate Representation for Compiler Infrastructure
MPI	Message Passing Interface
psutil	Python System and Process Utilities
OSS	Open-Source Software
T	Task
UC	Use Case
RPC	Remote Procedure Call
SIMD	Single Instruction/Multiple Data
WP	Work Package

1 Introduction and Purpose of this Document

In D1.5 DAPHNE project team describes the progress made from project month 25 to project month 36, respectively the work done in project year 3 (M25/Dec 2022 – M36/Nov 2023).

First, this report refers to the structure and purpose of the document. Second, D1.5 outlines the main objectives in DAPHNE and what DAPHNE consortium has done, particularly in project year 3, to reach those targets. Third, this annual report presents an overview of achievements throughout all the work packages 1 to 10, also here specifically addressing the work in the 3rd project year. Finally, an outlook is provided, and next steps are being addressed. The purpose of this document is therefore to provide an overview of work carried out in DAPHNE project until M36, with an emphasis on updates of project year 3.

2 Strategic Objectives

This section shows the strategic objectives and the work of DAPHNE consortium towards these objectives in the 3rd project year.

2.1. Objective 1 System Architecture, APIs and DSL (WP2-4)

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.*[1]

The most significant work towards this first objective was the further progressing within DaphneLib, a Python API with lazy evaluation as an additional entry point to simplify adoption, the advancement of algorithms and profiling/logging, as well as internal improvements such as the vectorized engine advancement. During the first two years of the DAPHNE project, the design of the system architecture had been completed and initial developments of the API and DSL implemented. Now during the third year of the project, the consortium focused on refinement and further implementation of planned features in these areas. The system architecture stabilized with minor additions to the code infrastructure like profiling and logging facilities. The DaphneDSL improved in many ways due to user feedback following increased use throughout the work packages. Consequently, more algorithms are implemented, leading to more operations being developed and tested. The DaphneLib, our python API, saw many

[1] The objectives, here outlined in italics, adhere to DAPHNE Grant Agreement.

improvements in supported operations higher level functionality. In that regard, DAPHNE can now interface with data structures of other python packages directly through shared memory, fostering efficient interoperation with external libraries. Other minor improvements and bug fixes that, in total, have a major impact on stability and therefore user experience, have found their way into the prototype.

2.2. Objective 2 Hierarchical Scheduling and Task Planning (WP5-WP7)

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.*

The work towards the second objective of DAPHNE has been spread across several work packages to jointly improve the capabilities of our prototype, not only regarding state-of-the-art performance, but also versatility and innovation. Catering to the latter, DAPHNE can now be deployed in distributed environments using MPI. The initial implementation aims for feature parity with the existing distributed backend that uses gRPC and needs to spawn separate worker processes. The range of supported operations has been extended to cater to the needs of algorithms that have been used in showcasing DAPHNE. This includes CUDA kernels as well as relational algebra or SQL operators to name only a few. The latter is one of several opportunities for code generation and operator fusion that DAPHNE can leverage. Initial work towards generating optimized execution paths at run-time has also been introduced for directly compiling MLIR based operations to executable code without the need of preexisting kernels as well as generation of CUDA code for efficient execution on GPU devices. The ongoing developments using FPGA accelerators will lead to automated kernel selection capabilities to leverage these devices in suitable situations. In conclusion, the most promising steps towards meeting the expectations of objective 2 in project year 3 were the extension of kernels, the MPI backend as well as the initial code generation.

2.3. Objective 3 Use Case Studies and Benchmarking (WP8-WP9)

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).*

The Use Case and benchmarking work packages have made progress according to DoA during the third year of the DAPHNE project. Besides the solutions developed in these areas improving over time due to active development, they also directly benefit from DAPHNE becoming more mature in its overall state. The close collaboration of the pipeline developers of the Use Case partners with the partners focusing on the DAPHNE prototype development has led to a better mutual understanding of both sides, which enables not only a better progress towards first results from real world applications but also a better focus on the pain points of practitioners and where to further improve the DAPHNE infrastructure. Alongside these positive developments, the work package for benchmarking has completed the initial implementation of the benchmarking framework UMLAUT, focusing on interaction with the DAPHNE prototype through our DaphneLib python interface. Using and accessing DAPHNE's features has become a more pleasant experience to DAPHNE target users in general with the introduction of precompiled container images and increased effort in writing documentation and presenting it well with a searchable interface.

3 Status and Progress Update of All Work Packages

This section provides an overview of the progress made in year 3 throughout the work packages.

3.1 WP1 Project Management (KNOW) [M1-M48]

WP 1 Project Management seeks to provide a high-quality work environment for research activities to thrive, to coordinate across WPs, to keep track of reporting and deliverables and to improve communication throughout the consortium, the European Commission and beyond. In this section WP1 Lead KNOW reports about the objectives of WP1 and the work towards those objectives in project year 3.

Regarding the main objectives of this WP (1) to act as the communication interface with the European Commission, DAPHNE project management has sought to share information effectively via the EU portal, reporting on all relevant continuous reporting items (deliverables, milestones, risks, publications, dissemination and communication, patents/IPR, innovation, open data, gender, ABS regulation) and communicate with the project officer (PO) on relevant topics such as organizing meetings (i.e. organization of Use-Case Workshop, General Assembly and Review meeting), aligning on reporting (i.e. workflow for Periodic Report), asking for changes to the Grant Agreement and support (1st project amendment, Non-Disclosure Agreement for Marcus Paradies of former partner Deutsches Zentrum für Luft- und Raumfahrt (DLR) and now University of Technology Ilmenau (TUI)) or reaching out for exploitation purposes (sharing invitations and aligning on events, i.e. BDVA and ICT-51 PPP cluster events).

Concerning WP1 objective (2) to establish means of effective communication and collaboration within the consortium, DAPHNE project management has reported on these means in D1.1 Project and Risk Management Plan [1]. In line with this plan, we have kept our basic 4-level project structure, used the, in D1.1 [1] described, mailing lists and tools, e.g., DAPHNE cloud and GitLab, as well as file storages for communication and collaboration purposes. We have migrated the development from a private GitLab instance to GitHub and maintained our DAPHNE registration procedure to ensure a smooth transition into (and out of) the project communication platforms and channels. Complementary to these channels and platforms we use regular WP-specific meetings, bilateral meetings, as well as All-hands meetings (monthly for 1,5 hours) across the entire consortium to ensure accurate and high-quality communication.

Resulting in WP1 objective (3) the organization of calls and meetings of the consortium, we hold our All-hands meetings monthly. In these consortium meetings, we discuss administrative and team updates, WP/technical updates, reporting and deliverable tracking such as news on for example publications and conferences; every month there are updates from all WP leads. The meeting serves the purpose of bringing together all consortium members, providing relevant information in a structured way, asking for support and alignment if required and giving everyone the chance to clarify questions that are preferred to be discussed orally within the whole consortium.

In addition to these regular general meetings, project consortium has met for the general assembly meeting once a year. The purpose and general outline of this meeting format have been elaborated in various documents such as the Proposal, the Grant Agreement or the Project Plan (D1.1) [1]. Updates are that now, that the end of the third project year is approaching, multiple demo presentations complement the WPs and UCs presentations and give more variety to the meeting. Moreover, in project year 3 we have started to meet personally again for a Technical Meeting, the Use Case Workshop, the Advisory Board and the General Assembly Meeting. We experienced a boost in motivation based on these personal encounters.

WP1 project management reinforces the communication structure devised in D1.1 Project and risk management plan [1] and reminds the entire consortium to be aware of the objectives and the related deliverables. Specific internal project management tracking tools are the deliverable and reviewer tracker, the exploitation tracker, and the financial tracking; their results are filtered and – depending on confidentiality-level - reported further within the EU portal and/or our project website [2]. Moreover, this objective seeks to ensure strategic realignment in case of unforeseen circumstances. Strategic realignment in project year 3 was necessary in terms of changes to DAPHNE project team (countermeasures to risks 1 and 2) with a shift of Marcus Paradies to Technische Universität Ilmenau (TUI), and the involvement of the consortium partners' legal departments and consultants.

Objective (5) addresses the coordination and quality assurance of reporting efforts. The submitted reports and deliverables that were submitted until the end of the first project period have been accepted and we have received the (reviewer) feedback that they are of rather high quality. With the constructive feedback received, we have started to include more reporting details, complementing our cross-referencing, and improve our prototype documentation efforts in the second project period and particularly in project year 3. Moreover, we should highlight that all consortium partners are collaborating on the DAPHNE system and that this intense collaboration is the essence of our project. Financial controlling, budget and effort reporting have been carried out, depicting deviations from the original budget plan, and giving the consortium partners the chance to compensate for these deviations in the upcoming project periods. The current DAPHNE website https://DAPHNE-eu.eu [2] shows those reports that are available to the public in the section *Publications*.

The central project management endeavors to maintain a general project overview across all WPs and the budget, to relate the actual work being done to the original project plan and to ensure effective communication have been carried out in this third project year - with many lessons learned, such as the importance of clear and motivating communication, including the setting of boundaries, decent conflict management skills, a professional team, as well as high responsiveness. Moreover, we found that for project management to thrive administrative/organizational and classic project management, communication, as well as technical skills need to be comprised in the consortium. These complementing skills result in a focus on the right priorities.

Eventually, a lesson learned is that as manager you need to observe and reflect on the project, the players/organizations, and the processes, but then not get lost in detail, but gain an overview, re-focusing on the higher European objectives, which are well thought-through. Here, patience and acting ethical, helpful, and social are key. In project management, we look forward to learning and growing even more as the project is starting to enter its last phase next year 2024, when it comes to truly exploiting the lessons learned from DAPHNE. For now, as the end of the 3rd project year, it is important to demonstrate that we have met the promises made in the Grant Agreement, our project plan.

As such, we have implemented DAPHNE language abstractions and the extended compiler prototype, shaping the scheduling structures available in DAPHNE, implemented relational filtering operators within the prototype and overview of data path optimizations and placement, enhanced the overview of the code generation framework as well as the DAPHNE pipelines, and implemented an internal benchmarking toolkit. Eventually, we are meeting to discuss this progress and further steps in our General Assembly Meeting 2023 coming up at Datahouse Graz on December 6. This progress is demonstrated in more detail in the following progress updates from the technical work packages (WPs 2-9).

3.2 WP2 System Architecture (KNOW) [M1-M21]

The open system architecture that fosters extensibility and caters to the needs of data analysis and data processing in the fields of machine learning (ML) and high-performance computing (HPC), which was documented in deliverable D2.1 [3]. In addition, D2.2 [4] refers to the refined overall architecture and key design decisions of the DAPHNE system infrastructure as an open and extensible system for IDA pipelines, comprising query processing, ML, and HPC. WP2 System Architecture was active until M21. The reporting period for this report D1.5 does therefore not cover WP2 activities.

3.3 WP3 Abstractions and Compilation (TUB) [M1-M48]

DSL Abstractions. In project year 3, we continued the implementation of DAPHNE's language abstractions, whose design we have described in detail before in D3.1 [5], focusing on two main contributions.

(1) We significantly extended the existing initial implementation of DaphneLib. DaphneLib enables using DAPHNE from Python scripts, thereby fostering the integration of DAPHNE into existing Python-based data science workflows. DaphneLib owes its efficiency to lazy evaluation, i.e., primitives are not eagerly executed, but instead an internal graph representation is built, which is transparently converted into a DaphneDSL script to leverage DAPHNE's entire optimizing compiler and runtime stack. We considerably extended DaphneLib's feature set by supporting virtually all DaphneDSL built-in functions. Furthermore, DaphneLib now also supports lazily evaluated complex control flow such as if-then-else, loops, and user-defined functions. We also designed and implemented efficient bi-directional data transfer between DAPHNE and widely used Python libraries, such as numpy, pandas, TensorFlow, and PyTorch.

(2) We started the creation of a large corpus of common high-level data science primitives implemented in DaphneDSL. In a first step, we manually translated complex scripts from Apache SystemDS to DaphneDSL, including linear regression, decisions trees, and random forests. In a second step, we designed and implemented a prototype of an automatic translation tool converting SystemDS's DML scripts to DaphneDSL. The goal is to provide users a large standard library of data science primitives at a conveniently high abstraction level to boost their productivity. Besides that, we further extended DAPHNE's SQL parser to support more expressive query processing.

Compilation. Based upon an early description of the compiler design in D3.1 [5] and the initial compiler prototype D3.2 [6], we focused on two main contributions in project year 3. Parts of these have already been included in the extended compiler prototype D3.3 [7].

(1) We improved the support for representing, propagating, and exploiting interesting data properties as the basis for complex compile-time and run-time decisions. In addition to the already existing intra-procedural analysis for data/value type, shape, and sparsity of intermediates, we added inter-procedural analyses to propagate interesting properties into complex user-defined functions. Moreover, we started working on a framework for supporting a rich and easily extensible set of interesting data properties. We expect the exploitation of interesting properties to become an important component for global pipeline optimizations across primitives for query processing, machine learning, and high-performance computing.

(2) We added code generation for different backends. So far, the DAPHNE compiler was limited to lowering DaphneIR's domain-specific operations from linear and relational algebra to calls

to pre-compiled kernels written manually in C++. Meanwhile, we added an alternative route to the DAPHNE compiler, which lowers DaphneIR operations to specialized code generated on-the-fly. We currently support two backends. First, an MLIR-based backend generating code in lower-level MLIR dialects like linalg, affine, math, and memref. Our proof-of-concept implementation for operations on dense matrices already allows interesting instruction-level optimizations, such as inlining, operator fusion, and special simplification rewrites. We also made sure that C++ kernels and generated MLIR-kernels are interoperable in terms of their input/output data, such that we can combine them within a single pipeline. Second, a CUDA-based backend generating CUDA C++ code, with a focus on sparse data processing on GPUs.

Besides that, further work on the DAPHNE compiler concerned update-in-place optimizations for increased cache efficiency and an extension of DaphneIR with operations for columnar data to facilitate analytical query processing. Finally, we worked on various little feature improvements and bug fixes in the DAPHNE prototype, many of which were initiated through the increased use of the system by both the other technical work packages and our use-case partners.

3.4 WP4 DSL Runtime and Integration (ICCS) [M1-M48]

In WP4 we have outlined the updates and enhancements in the design and implementation of the DAPHNE Runtime system. We can categorize progress made in this period along the following axes: a) advances in I/O; b) updates in runtime communication and c) updates inside the execution engine itself.

Relative to I/O in DAPHNE Runtime: We have updated the DAPHNE serialization library to support DenseMatrices and CSRMatrices. Three serialization methods are implemented, offering flexibility in preserving and reconstructing DAPHNE objects. This not only improves I/O performance but also broadens support for distributed execution to various data types.

During this period we have also commenced an integration of the runtime with HDFS. We are leveraging the Libhdfs3 library for efficient interfacing with HDFS from C++ code. The workflow involves file upload, serialization, data distribution, information gathering, deserialization, and inter-node write and read. This integration is planned to greatly enhance data locality and establish a scalable distributed computing environment.

Relative to Runtime Communication advances: DAPHNE's communication capabilities are enhanced with the full incorporation of MPI and the introduction of synchronous gRPC alongside asynchronous gRPC. To overcome the limitation of message size in both frameworks (as well other possible future integrations), we have introduced message chunking. This enables the system to handle larger data volumes by breaking them into manageable segments, seamlessly facilitated by the serialization library.

Execution Engine Updates: The execution engine undergoes several updates, including metadata support for both gRPC and MPI, integration with the Eigen library for eigenvalue and eigenvector calculations, CSR support for PageRank, and CUDA-specific kernels. Moreover, basic monitoring support for the runtime is added through the PAPI library, allowing fine-grained event information extraction.

Finally, we provide access to the DAPHNE runtime v2 prototype, publicly available in the DAPHNE development repository. Guidelines for building and running DAPHNE are included in this period's deliverable, along with examples of using different communication frameworks for executing DSL scripts.

3.5 WP5 Scheduling and Resource Sharing (UNIBAS) [M1-M48]

The WP5 team has made substantial progress in shaping the scheduling structures available in DAPHNE. Our primary focus in the DAPHNE project year 3 was ensuring that DAPHNE supports efficient scheduling strategies. We implemented such efficient scheduling strategies, improved the usability of the scheduling mechanisms, and developed a strategy to unburden the user of the process of selecting a high-performing scheduling strategy.

We implemented task scheduling mechanisms for data analysis pipelines inside the DAPHNE scheduling framework namely DAPHNEsched. For shared-memory environments, this includes the implementation of the scheduling infrastructure, two static, and ten efficient dynamic scheduling techniques. The static scheduling strategies are STATIC straightforward parallelization strategy and MSTATIC (a modified version of STATIC) that considers a static work partitioning with a smaller grain size than default STATIC. The dynamic strategies include the following from the class of self-scheduling algorithms: Guided (GSS), Trapezoid (TSS), Practical implementation of Factoring (FAC2), Trapezoid Factoring (TFSS), Fixed-increase (FISS), Variable-increase (VISS), Performance loop-based (PLS), Practical implementation of Fixed-Size Chunk (MFSC) and Probabilistic Self-Scheduling (PSS). These scheduling mechanisms are implemented in DAPHNEsched and improve DAPHNE efficiency and cater to the dynamic nature of shared memory environments, providing adaptability in the face of changing workloads and systems.

Another achievement is the incorporation of the concept of task granularity into DAPHNEsched. The concept is defined to limit the smallest granularity of the task granularity to use with a given scheduling strategy. This concept is already applied in widely used parallel runtimes, such as OpenMP, and can be used by the user to minimize scheduling overhead and/or improve data locality. We also implemented in DAPHNEsched an automated task grain size selection mechanism, allowing the user to achieve improved performance without extensive experimentation. This significantly improves the usability of the DAPHNEsched scheduling strategies while preserving improved performance at a low cost or user effort.

DAPHNEsched also supports three strategies for organizing work queues. The default work queue is centralized across all devices and device groups, while the other work queue strategies are one queue per CPU group, and one queue per CPU core. To achieve load balancing in the presence of multiple work queues, we implemented four work-stealing victim selection strategies: SEQ is the default selection strategy in which the victim is the next adjacent worker; the other selection strategies are the following: SEQPRI where adjacent workers on the same NUMA domain are prioritized as victims, RANDOM where victims are randomly selected, and RANDOMPRI where victims are randomly selected among workers on the same NUMA domain.

Communicating using the MPI is pivotal not only in HPC but also in data science. DAPHNEsched also supports scheduling of data analysis pipelines at the MPI process level. Currently, it statically distributes the workload and data while delegating the dynamic scheduling and load balancing process to the local (shared memory) DAPHNE runtime. Based on this support, we are currently developing dynamic scheduling solutions at the level of MPI processes.

In summary, DAPHNEsched offers numerous efficient scheduling strategies to lower resource waste and increase performance. We also improved user experience by studying and developing automated scheduling strategies that eliminate the need for user expertise to use and achieve high performance with DAPHNE. Finally, to ensure DAPHNE's scalability, we developed the structure for process-level scheduling with MPI. This will allow the development of efficient dynamic scheduling strategies for process level with MPI.

3.6 WP6 Computational Storage (ITU) [M1-M48]

The integration of computational storage in DAPHNE requires the availability of devices supporting code offload, from the hosts where DAPHNE is run. We completed the design and implementation of the Delilah prototype. This was marked by the publication of a paper at Damon 2023 titled "Delilah: eBPF Offload on Computational Storage" [9]. This groundwork now makes it possible to work towards the integration of Delilah in DAPHNE, which will be the main focus next year. The demonstrator D6.3 [10] illustrates the progress TUD and ITU have made implementing relational filtering operators with various forms of eBPF offload. Delilah provides an asynchronous interface, which is different from the synchronous I/O interface currently used in the DAPHNE runtime. DLR has focused on efficiently leveraging asynchronous I/Os. This work is also demonstrated in D6.3 [10].

Another line of work in WP6 is closely related to WP7. Delilah now incorporates hardware accelerated functions. Intel is exploring off-path computational storage, i.e., the combination of FPGA and SSDs connected via a PCIe switch. This diversity will enable us to compare different approaches to computational storage, using DAPHNE.

3.7 WP7 HW Accelerator Integration (TUD) [M1-M48]

The integration of hardware accelerators into DAPHNE, as originally discussed in D7.1 [11], was further advanced and the design refined as well as initial operations were accelerated using GPU and FPGA as described in D7.2. [12] In the domain of GPU processors, the supported operations range from common unary and binary arithmetic over linear algebra to machine learning specific kernels commonly applied in neural networks. Additionally, supporting a certain type of hardware accelerator brings along the need for kernels that deal with context creation and device initialization, to keep track of the handles to their API and other specifics.

The memory management aspects of these devices are wrapped inside their respective classes and data handles and details of its representation is organized in so-called meta data object. This structure enables fine grained data placement decisions. For GPU computing, we rely on the CUDA API and the hardware that supports it to a large extent. However, an initial integration of OneAPI from Intel has been done to further extend the list of supported devices in this category and not solely rely on the products of a single vendor. This initial integration also drives the documentation efforts of extensibility to guide potential third parties in the endeavor of adding functionality to DAPHNE.

The second major family of hardware accelerators, namely FPGA, has also seen the successful development of important operations like quantization or general matrix multiply (GEMM) and an initial integration of an Intel Stratix based accelerator. Further efforts of more specialized work dealing with SIMD exploitation through a SIMD abstraction library, performance models and code generation has been conducted and documented in D7.2 [12] and D7.3. [13] In particular, the further development of the SIMD abstraction library (virtual vector library), which not only maps to various SIMD extensions of general-purpose CPUs but also to Intel FPGA cards using OneAPI, should be emphasized. More details are reported in D7.3 [13], which mainly summarizes our achieved results for code generation.

To simultaneously exploit these heterogeneous devices, we started to add the functionality of running fused pipelines through our vectorized execution engine on them. This integration not only enables us to further exploit available resources but also gives us the opportunity to add tuning knobs for scheduling and load balancing where extra care needs to be taken to cater to every device's needs. Regarding input and output, as the utilization of accelerators usually implies a certain cost of pushing the task to the compute units and pulling the result back to main memory.

3.8 WP8 Use Case Studies (KAI) [M1-M48]

Since the report on initial Use Case pipelines (deliverable D8.1) [14], work has progressed towards enhancing the pipelines to utilize the DAPHNE system infrastructure. Naturally, this goes together with providing feedback to the technical work packages and/or by requesting certain features. Such are low-level primitives that operate on matrix data types like `idxMin` and `idxMax` but also high-level functions for data processing and machine learning like `randomForest`.

The technical details of the individual Use Cases are specified in D8.2 [15], which is submitted at the same time as this annual report (end of November 2023/M36). Regarding feature requests & bug reporting, upon working with the DAPHNE system, some parts of compiling the DSL script are still in an early stage. To overcome our workarounds, we regularly request for new features and sometimes also find bugs. Overall, we have created 15 issues on GitHub, whereby 7 of them are already solved and implemented in the source code. Of the remaining open issues, some are already being implemented.

In context to open-source contributions, the people working in WP8 on the Use Cases are the first users of the DAPHNE system infrastructure outside of the core development team. We thereby provide not only valuable feedback, but even contribute directly on the open-source GitHub repository. Through these activities, together with the ongoing discussion and exchange rounds, we actively take part in the development process of the DAPHNE system infrastructure.

3.9 WP9 Benchmarking and Analysis (HPI) [M1-M48]

In the third project year (M25-M36) the DAPHNEBench Benchmark definition (T9.2) that was presented at the end of M30 in deliverable D9.2 [16] was finalized. The definition and development of the first version of an internal Benchmarking and Profiling Toolkit (T9.3) was started. Next to the regular Work Package meetings and the All-hands meetings we meet for an in-person Use Case meeting in Graz to present the status of our work and discuss it with other DAPHNE Use Case partners. In the following, we first give details on our work in T9.2 before we continue with T 9.3 and D 9.3. Below we describe our work by task.

In our survey on Big Data, HPC, and ML benchmarking frameworks [17], we have identified a need for a benchmarking framework that will encompass and measure the performance of Integrated Data Analysis (IDA) pipelines. The findings in deliverable 9.1 were based on the lack of convergence in DM, HPC, and ML benchmarking frameworks. We proposed a framework that would combine metrics and measurement aspects from the three domains.

In Task 9.2, we outlined a set of requirements necessary to define the benchmarking framework. The focus of the proposed framework is to capture the complete pipeline lifecycle, covering multiple benchmarking aspects and abstraction levels of IDA pipelines. To evaluate the complete pipeline lifecycle, we defined metrics for tracking the end-to-end performance, as well as metrics that cover different runtime aspects. We defined the system under test to provide different levels of abstraction.

In the benchmarking framework definition [17], we cover scenarios in which the SuT can be a single method call, a single pipeline stage, as well as a complete pipeline. We capture the diversity in IDA pipelines by implementing diverse workloads based on DAPHNE Use Cases, as well as open-source workloads. Specifically, we used the workloads developed in DAPHNE Use Case 1 - Earth Observation, and Anomaly Analysis Use Case provided by hardware manufacturer Backblaze [18].

We have defined a data model that supports evaluating the heterogeneity of IDA pipelines. The data model allows us to capture performance and runtime characteristics in the several pipeline dimensions. We have summarized the definition of the benchmarking toolkit in deliverable 9.2 [16]. As part of task 9.3, we started defining and implementing an internal benchmarking toolkit that is able to cover a wide variety of metrics. To easily integrate the measurement of any metric without significant change to the original pipeline used for benchmarking we follow a decorator-based design.

To measure all metrics of interest we integrated widely used libraries and tooling such as Pythons time library [19] to use the Pythons performance counter for time-based metrics, psutil (python system and process utilities) [20] for memory and CPU metrics, and the pyRAPL library to measure energy and power [21]. All measurements are logged into a database backend and are interactively accessible via an easy-to-use command line interface that automatically provides plots for easy analysis of measurements.

To automate benchmarking results for some real workloads as easily as possible, we started implementing first versions of scripts that automatically run real-life Use Cases provided by our project partners. For now, we focused on the Earth Observation Use Case and the Back-blaze Anomaly Analysis Use Cases. We described the current state of our work in deliverable D9.3

[22]. Furthermore, we demonstrated and discussed our benchmarking toolkit with work package partners at the aforementioned DAPHNE Use Case meeting in Graz where we also showed some initial results of bench-marking the DAPHNE workloads written in DAPHNE's domain-specific language as well as in DAPHNE's proprietary version of its Python interface DaphneLib.

3.10. Dissemination and Exploitation (KNOW) [M10-M48]

The primary objective of dissemination and exploitation is a broad and open sharing of the project results via publications and talks, dedicated networking efforts, and an open-source reference implementation. In project year 3 these activities have been re-defined in D10.1 Refined Dissemination and Exploitation plan [23] and have been carried out accordingly.

In the first task of this WP T10.1 - which is the only active task until M24 - Continuous Dissemination via Publications and Talks [M10-M48, Lead: KNOW, Participants: all, Effort: 13PM] - complementary to the work in the technical WPs that perform dissemination via scientific publications, which is interleaved with the actual research, 51 publications and 75 talks have been organized or co-organized are being listed and further communicated via the EU portal and conveyed to the public via our DAPHNE website [2].

The dissemination tracking document is updated every 3 months asking project partners to provide details concerning their DAPHNE-related activities and update the listed items. The publications and talks are inter-linked according to the open-source principle. The organization of the dissemination tracker is time-consuming, but necessary as all relevant dissemination and communication activities must be recorded.

Thus, the current overview of our publications and talks throughout the project runtime till now (M1-M36) shows 51 publications and 75 talks. They directly link to the source material. We have achieved to promote DAPHNE in the central venues of data management, such as VLDB, HiPEAC and Euro-Par as well as workshops organized by the European Big Data Value Association (BDVA). KNOW are an active member of the interplay between EU research policy and research organizations through our leading role in BDVA. KNOW former CEO has been on the board of directors (BoD) of BDVA since 2016, granting KNOW access to the policy making level. Her mandate was taken over by our current CSO Roman Kern. KNOW also actively participate in BDVA taskforces.

In addition to the listed publications and talks we have also connected to similar projects such as MARVEL, EVEREST, or eFlows4HPC synergizing within ML/DL systems and system support. In detail, in a reciprocal process we have been exploring architectures for gathering heterogenous data, language abstractions, intermediate representation, methods for extreme-scale analytics, e.g. combination of ML models, simulations and subsequent data analysis in different Use Cases, standardized interconnection methods, e.g., runtime integration, HPC libraries as well as data fusion and data integration technologies.

As complementary dissemination and exploitation measures to maximize impact - in addition to publications and talks and exploring synergies by reaching out to similar projects - we have contributed to benchmarks and standards and have implemented a release timeline to the open-sourced DAPHNE reference implementation. We have started to facilitate real-world data or datasets with similar characteristics to simplify reproducing our experimental results.

Eventually, as planned we have regularly updated our DAPHNE website https://DAPHNE-eu.eu [2] which communicates the project idea and objectives clearly. The website gives a general overview of the project, showing latest news, listing all consortium partners, displaying a team photo of our personal gathering for the TEC-Use Case Workshop in Graz in September 2023, presenting portrait photos of the consortium partners, referring to the EU's Horizon 2020 research and innovation program, displaying a regularly updated list of publications and talks including the public deliverables, presenting the DAPHNE Use Cases, and showing a contact form for visitors to get in touch. The website also invites to visit the open-source community on GitHub and follow DAPHNE project on social media. DAPHNE is active on Twitter/X and LinkedIn. We consider these social media engagements not as core but still as must-haves to promote our project and easily link to our target users and project partners.

Within the 3rd project year, we have been able to increase the quantity of our social media posts and our efforts towards project exploitation. Therefore, we have created a product portfolio element for DAPHNE to make it more accessible to a broader audience. In this portfolio we describe the advantages of DAPHNE to potential customers. Regarding visuals, we have worked on a DAPHNE poster that can be adapted easily.

Regarding DAPHNE major target groups, our dissemination efforts address system or data engineers that are designing and operating robust infrastructures, and application users that are mostly concerned with productivity and accurate predictions. The open-source strategy has been mobilized to attract target users and invite them to give feedback, which is considered as vital function to keep customers and users interested. Moreover, these loops and enable us to improve DAPHNE. Inconvenient or not directly applicable user requests are retained for potential joint follow-up research or industry projects which we are starting now at the end of the 2nd project period/end of project year 3.

4 Future Outlook

The next steps for DAPHNE consortium are to continue the high efforts of professionalizing DAPHNE system and focusing on the integration of DAPHNE on Use Case level. On administrative level, the 2nd periodic report that is to be developed with the beginning of December 2023 (to be finished by end of December for our reviewers and at the end of January 2024 at the very latest) needs our conjoined attention. Moreover, the contents of the 2nd Review Meeting that is to take place on January 17, 2024, need to be prepared.

As we are more focused on the work itself than reporting or communicating it, we need to improve our self-presentation regarding the conjoined work on DAPHNE. Eventually, as the bridge between HPC and ML aspects is unique in DAPHNE and is worth of deeper investigation, we are considering a potential future collaboration of DAPHNE consortium within future projects. Finding a joint direction and strategy in which we'd like to develop and thus, bringing together the different strengths and interests of our consortium within these potential future projects, is another high-priority task that is initiated currently in November 2023.

In conclusion, referring to what is promised in the Grant Agreement for project year 3, we are on track as we have deepened the inter-connection of the Use Cases and DAPHNE system in this year with promising results. Eventually, the personal encounters in project year 3 have motivated us to continue exploiting all our efforts and deliver all aspects of DAPHNE system in high quality.

References

[1] D1.1 Project and risk management plan [confidential]

[2] DAPHNE EU project website https://DAPHNE-eu.eu

[3] D2.1 Initial System Architecture [public] https://daphne-eu.eu/wp-content/uploads/2021/11/Deliverable-2.1-fin.pdf

[4] D2.2 Refined System Architecture [public] https://daphne-eu.eu/wp-content/uploads/2022/08/D2.2-Refined-System-Architecture.pdf

[5] D3.1 Language Design Specification [public] https://daphne-eu.eu/wp-content/uploads/2022/06/DAPHNE_D3.1_LanguageDesign_v1.2.pdf

[6] D3.2 Compiler Prototype [public] https://daphne-eu.eu/wp-content/uploads/2022/03/DAPHNE_D3.2_CompilerPrototype_v1.1-for-Website.pdf

[7] D3.3 Extended Compiler Prototype https://daphne-eu.eu/wp-content/uploads/2023/05/DAPHNE_D3.3_ExtendedCompilerPrototype_v1.1.pdf

[8] GitHub repository https://github.com/DAPHNE-eu/DAPHNE

[9] Niclas Hedam, Morten Tychsen Clausen, Philippe Bonnet, Sangjin Lee, Ken Friis Larsen: "Delilah: eBPF-offload on computational storage" (19th International Workshop on Data Management on New Hardware – DaMoN) https://dl.acm.org/conference/damon

[10] D6.3 Prototype and overview of data path optimizations and placement [public]; to be submitted at the same time as this report in project month 36/November 2023

[11] D7.1 Design of integration HW accelerators [public] https://daphne-eu.eu/wp-content/uploads/2022/05/DAPHNE_Deliverable_7.1.pdf

[12] D7.2 Prototype and Overview HW Accelerator Support and Performance Models [public] https://daphne-eu.eu/wp-content/uploads/2022/12/D7.2-Prototype-and-Overview-HW-Accelerator-Support-and-Performance-Models.pdf

[13] D7.3 Prototype and overview code generation framework [public]; to be submitted at the same time as this report in project month 36/November 2023

[14] D8.1 Initial pipeline definition all use cases [public] https://daphne-eu.eu/wp-content/uploads/2021/11/Deliverable-8.1-fin.pdf

[15] D8.2 Improved pipelines all use case studies [public]; to be submitted at the same time as this report in project month 26/November 2023

[16] D9.2 Initial benchmark concept and definition [public] https://daphne-eu.eu/wp-content/uploads/2022/12/D9.2-Initial-Benchmark-Concept-and-Definition.pdf

[17] Nina Ihde, Paula Marten, Ahmed Eleliemy, Gabrielle Poerwawinata, Pedro Silva, Ilin Tolovski, Florina M. Ciorba, and Tilmann Rabl: "A Survey of Big Data, High Performance Computing, and Machine Learning Benchmarks" (Technology Conference on Performance Evaluation & Benchmarking, TPC, published by Springer, 2021) https://hpi.de/fileadmin/user_upload/fachgebiete/rabl/publications/2021/A_Survey_of_Big_Data_High_Performance_Computing_and_Machine_Learning_Benchmarks.pdf

[18] Backblaze Hard Drive Data and Stats https://www.backblaze.com/b2/hard-drive-test-data.html

[19] Python Standard Library Documentation – time https://docs.python.org/3/library/time.html#time.perf_counter

[20] psutil Documentation https://psutil.readthedocs.io/en/latest/

[21] pyRAPL Documentation https://pyrapl.readthedocs.io/en/latest/quickstart.html

[22] D9.3 Initial prototype of benchmarking toolkit [public]; to be submitted at the same time as this report in project month 36/November 2023

[23] D10.1 Refined dissemination and exploitation plan [confidential]