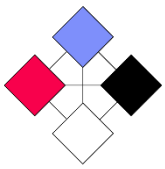
D9.2 Initial Benchmark Concept and Definition



DAPHNE

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

Version 1.2

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 D9.2, HPI presents the initial concept and definition of the benchmark for the DAPHNE system.

D1.10 Refined Dissemination and Exploitation Plan			
WP9 – Initial benchmark concept and definition			
Type of document	R	Version	1.2
Dissemination level	PU	Project month	24
Lead partner	HPI		
Author(s)	Paula Marten, Ilin Tolovski, Nils Strassenburg, Tilmann Rabl		
Reviewer(s)	Philippe Bonnet, Ahmed Eleliemy		

Revision History

Version	Revisions and Comments	Author / Reviewer
V1.0	Initial draft	Ilin Tolovski
V1.1	Draft before review	Paula Marten, Ilin Tolovski, Tilmann Rabl
V1.2	Draft after review	Paula Marten, Ilin Tolovski, Tilmann Rabl

1 Introduction

Benchmarking tools are essential to evaluate systems and their capabilities. The current landscape of data processing systems consists of multi-faceted architectures that incorporate methods from big data management (BD), high-performance computing (HPC), and machine learning (ML). While the systems in these domains are converging in the ways they handle data and perform computation, the tools currently available for their evaluation remain specialized for measuring individual aspects of the workloads. Such limited measurements cannot provide wider scope insights into the systems' functionality and their end-to-end performance.

In a survey on big data, high performance computing, and machine learning benchmarking frameworks, we have observed that the state of the art only partially reflects the convergence of the systems in the respective domains [10]. In order to evaluate the performance of such data processing systems fairly, there is a need for a benchmarking framework that can altogether measure the data management, computation, and statistical aspects of a system.

There exist several benchmarks covering the BD, HPC and ML domains. Most BD benchmarks focus on the system performance on data analysis tasks. Some benchmarks also evaluate the performance on either data collection or storage. However, only few benchmarks, such as BigBench [8] cover all tasks of a BD system. Convergence towards other domains can be observed in Big Bench and Big Data Bench [7], allowing to measure the system performance on statistical computations.

A similar situation can be observed when evaluating HPC Benchmarks. Some benchmarks, for example, SPEC MPI [15] evaluate different performance categories of HPC systems. Other benchmarks such as HPL-AI [9] and ML-Perf-HPC Benchmark [6] exhibit some convergence toward covering multiple aspects of an IDA pipeline, as they include high- and low-level precision calculations.

There exist several ML benchmarks, most of them focusing on the training aspect of ML systems, as this is usually the bottleneck for such systems. MLPerf [12] also includes model inference and time consumption metrics offering the possibility to evaluate a system's inference performance, while still maintaining a focus on training performance. There are benchmarks also aimed at the preprocessing stage of ML, such as CleanML [11], converging towards big data benchmarking frameworks.

Furthermore, there are multiple levels of an ML system that can be targeted by a benchmark. Metrics like hyperparameter influence or number of epochs needed to reach a certain loss or accuracy target the application and middleware layer of the system. Metrics concerning the consumption of resources like energy and CPU can be used to also take the hardware layer into account. Some benchmarks, for example, DeepBench [13] and MLPerf [12] include metrics concerning the hardware layers' performance during training like TeraFLOPS per millisecond and throughput, respectively.

To conclude, while most benchmarks focus on one aspect of their corresponding application domain, some benchmarks combine multiple tasks of the domain. However, there are no benchmarks showing convergence towards covering multiple tasks of the IDA pipeline as well as covering multiple levels of the system. This might be because not only is the purpose of the

stages disjoint, but especially because each task and layer require different metrics [10]. This lack of a benchmark able to provide metrics for all tasks and layers of an IDA pipeline prevents the end-to-end evaluation and comparison of IDA systems, such as DAPHNE.

In this report, we present our definition of an end-to-end benchmarking framework that measures the end-to-end performance of a complex data processing system.

We focus on the complete pipeline lifecycle, covering several benchmarking aspects and abstraction levels of IDA-pipelines, collecting a set of metrics reflecting the system's performance and resource utilization through supervised metrics. To this end, we also cover metrics related to domain-specific tasks such as simulation and computation for HPC, data cleaning for BD or preprocessing, model training, validation, and inference for ML. To this end, we also collect a set of metrics reflecting the system's performance and resource utilization through supervised metrics, as well as the performance of the trained model through valued metrics.

This report is structured as follows. In Section 2, we define the benchmarking framework by describing the IDA pipelines, the benchmarking aspects, the systems under test, as well as the workloads covered by the framework. In Section 3, we provide the outline for implementing such a framework, focusing on the design of the benchmarking methods, framework, data model, as well as the metrics covered by the benchmarking framework. In Section 4, we showcase two use cases, earth observation, and hard drive anomaly analysis, as examples of IDA pipelines covering all four benchmarking aspects. Finally, in Section 5, we conclude the report.

2 Benchmark Definition

This section focuses on defining the core characteristics of the benchmarking framework. We introduce IDA pipelines, the benchmarking aspects, systems under test and workloads covered by the proposed framework.

2.1 IDA Pipelines

IDA pipelines execute workflows combining multiple tasks, such as generating data with the use of simulations, data sorting or encoding and training, and using an ML model. The pipelines can be divided into three stages, computation, data processing, and training, and include components from BD, HPC and ML applications. In Figure 1 [10], we show the complete IDA pipeline ecosystem. In this report, we focus on the application and middleware layers.

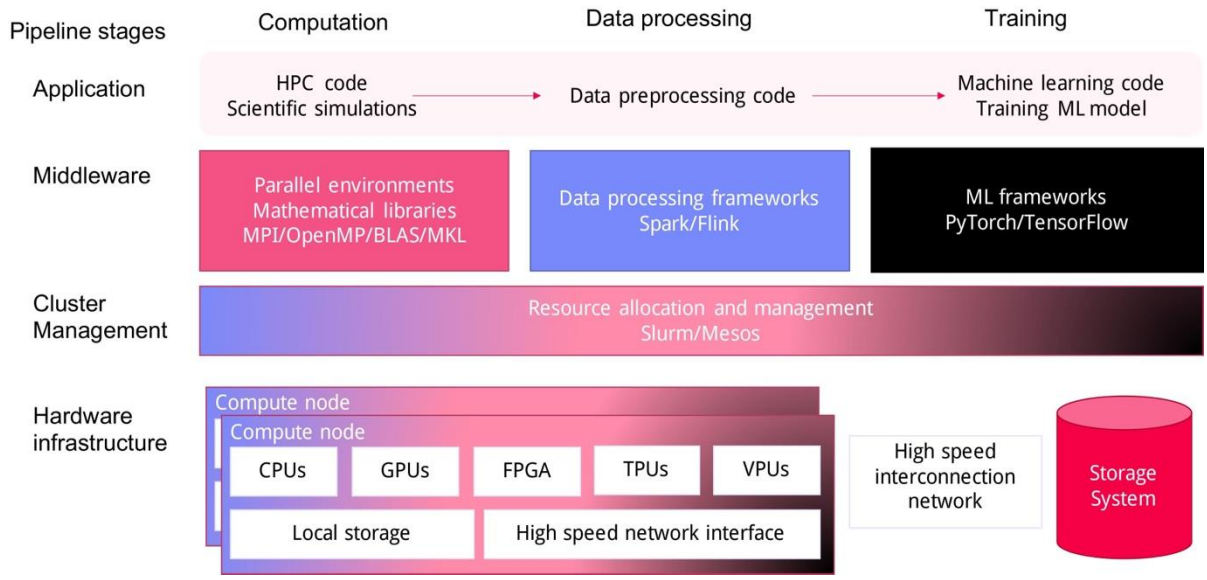


Figure 1: IDA Pipeline and its ecosystem [10]

Each of the domains, BD, ML, and HPC, presents its own challenges, computation methods, and performance metrics that are challenging to evaluate. Some benchmarks from each of these domains show convergence towards covering multiple stages and layers of the IDA pipeline. Thus, a benchmarking framework needs to cover these as well, including distinct metrics for each stage.

2.2 Benchmarking Aspects

Big data, high performance computing, and machine learning systems are often represented as pipelines composed of several stages or aspects, such as data preprocessing or cleaning, simulation, computation or training, and specifically for ML systems, inference. Current benchmarks are specialized to evaluate only individual aspects of the pipeline, having a low or no overlap with the other stages.

We propose a benchmarking framework that includes all aspects of an IDA pipeline. To accurately measure the performance of each aspect, we collect domain specific metrics related to each aspect as well as end-to-end runtime metrics, allowing for a more general evaluation of the whole pipeline. We also evaluate the training aspect in more detail by collecting training specific metrics.

With the end-to-end metrics, we evaluate the runtime characteristics of each benchmarking aspect. These include the elapsed time, memory consumption, CPU consumption, energy consumption, as well as the throughput and latency of the system. For the training-specific metrics, we focus on the model performance with respect to time, as in epochs-to-accuracy, epochs-to-loss, as well as collecting configuration parameters of the pipeline to measure the hyperparameter influence, and the confusion matrix of the generated predictions.

The proposed framework enables tracking of all end-to-end metrics in each aspect of an IDA pipeline. In this way, we evaluate the share of each stage in the end-to-end pipeline performance. Accordingly, the framework supports isolated evaluation of individual aspects for more focused pipeline analysis.

The benchmarking framework is developed to evaluate pipelines executed by systems that cover more than one aspect of the IDA stack. Our system of focus is the DAPHNE system for integrated data analysis. However, the proposed framework can also be used to evaluate IDA pipelines consisting of one or more data processing systems that are integrated in an IDA. The benchmarking framework can be used to evaluate end-to-end IDA pipelines, as well as individual parts of the pipeline, related to the BD, HPC and ML domain.

The benchmarking framework also allows for easy comparisons between multiple pipeline-runs and systems and features the possibility to visualize metrics obtained during a pipeline run.

2.3 System under Test

As a part of the WP 9 in the DAPHNE EU project, we focus on evaluating the end-to-end system capabilities of the DAPHNE system for integrated analysis. However, the benchmarking framework design is lightweight and modular, which can be integrated into IDA pipelines that include one or more data processing frameworks across different stages.

The proposed framework can be used as a wrapper around data processing methods. Runtime metrics are wrapped around individual method calls, as well as several methods comprising one or more stages of the pipeline. To evaluate each system integrated into an IDA pipeline, the benchmarking framework is to be used for a set of methods run by the system at hand. This allows us to determine per-system bottlenecks in the pipeline, as well as focus on the runtime performance of individual system operators. In Figure 2, we show an example of a benchmark run for a given system under test.

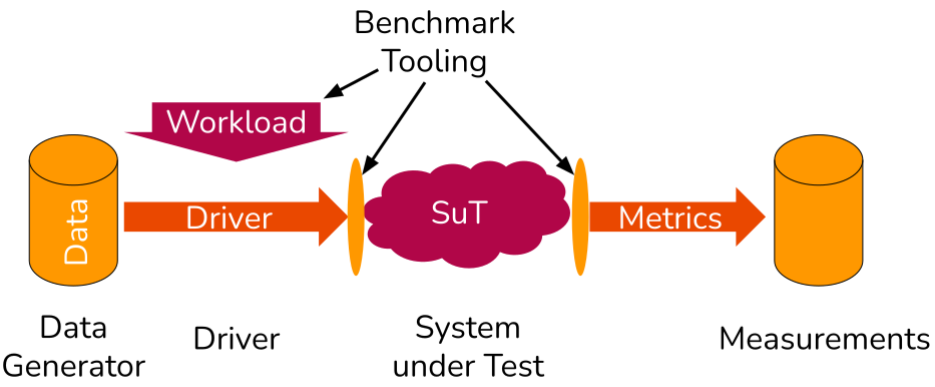


Figure 2: Execution scenario of a given benchmark

At the definition stage of the framework, we are covering generic data processing pipelines, that can include Python libraries, as well as Java and C++ executables.

2.4 Workloads

The workloads currently supported by the proposed framework include:

- Data cleaning and preparation workloads – part of the data preprocessing stage of an IDA pipeline
- Machine learning model training – part of the training stage of an IDA pipeline
- Inference – part of the computation stage of an IDA pipeline

Such workloads are included as elements of benchmarking use cases that we considered during the design process and are described in more detail in Section 4. Our benchmarking framework will be extended to support workloads that reflect all stages of an IDA pipeline. The new workloads are part of the future work and implementation of the benchmarking framework.

3 Benchmark Specification

In this section, we present the specification of the new benchmarking framework. We outline the implementation plan for the benchmarking methods and framework. Additionally, we specify the metrics and the data model in more detail.

3.1 Benchmarking Methods

The developed benchmarking framework captures *supervised* and *valued* metrics. The supervised metrics monitor the resource consumption of the system. For some supervised metrics, namely latency and throughput, these measurements are also combined with pipeline internal data. Supervised metrics are collected using decorator functions which annotate the parts of the pipeline that are being measured and are responsible for collecting the desired data. Further, valued metrics collect metric values which are the result of the different stages of an IDA pipeline, such as epochs to accuracy and confusion matrices.

3.2 Framework

The benchmarking framework is implemented in Python and provides a package that measures the metrics of the system and model at hand. Furthermore, it also provides a command line interface to evaluate the results and compare them across multiple pipeline runs. The metrics are stored in a database, whose entries are managed by a central Benchmark class. To capture the valued metrics tracker classes, each responsible for capturing one metric and reporting it to the benchmark instance, are used. For supervised metrics the BenchmarkSupervisor class, a decorator for the pipeline, collects all specified metrics and passes them to the benchmark instance.

3.3 Data Model

Metrics of multiple pipeline runs are stored in one database, allowing for easier comparison and visualization of multiple pipeline runs together. The database consists of two tables, as shown in Figure 3.

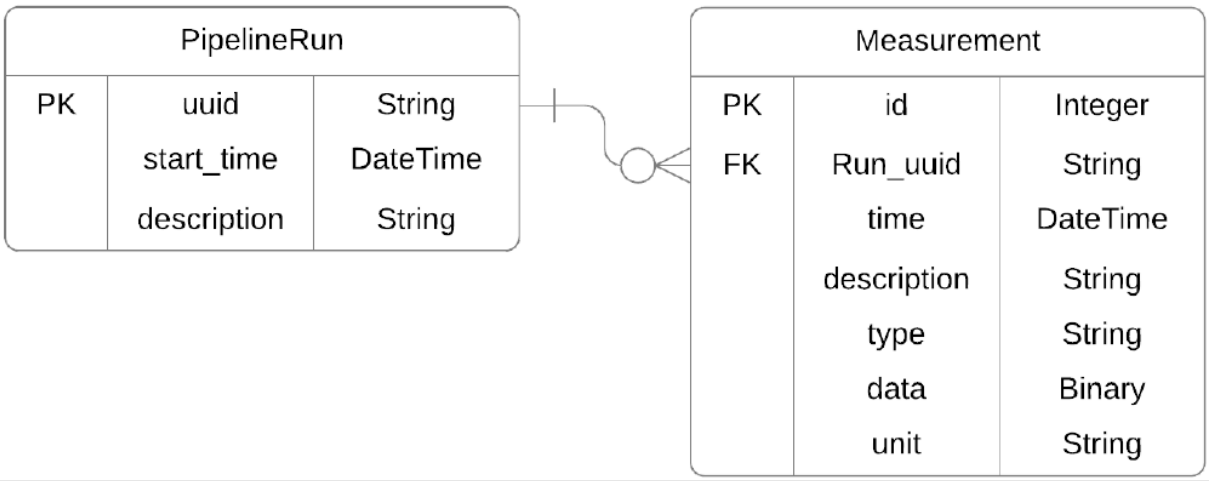


Figure 3: ERD for the proposed data model

One table saves entries for each pipeline run, including an ID, the start time of the run and a description. The other table consists of entries for all metrics collected. A measurement entry includes an id, the id of the corresponding pipeline run, a time, description, the type of metric saved and the data including its unit.

3.4 Metrics

The available metrics are listed in Table 1 and are divided into valued and supervised metrics. They are further specified in the following sections.

Valued metrics	**Supervised metrics**
Confusion matrix	Time consumption
Hyperparameter influence	Memory consumption
Epochs to accuracy	Energy consumption
Epochs to loss	CPU consumption
	Latency
	Throughput

Table 1: Available metrics, valued metrics are marked blue, supervised metrics are marked pink

3.4.1 Valued Metrics

The confusion matrix is measured during the training step, as well as during testing and inference. For each predicted datapoint the confusing matrix saves the predicted and actual value. Epochs to accuracy and epochs to loss both require a list of integers as values. These are the target accuracies or losses, respectively. For each value the time, where the accuracy or loss of the model exceeds or undercuts this value is collected.

The benchmarking framework further tracks certain hyperparameters, the influence of the value of single hyperparameters can be calculated from these values.

3.4.2 Supervised Metrics

The time consumption metric measures the time taken for execution of the function, which is evaluated. The memory consumption metric measured the memory used in the execution of a function at given intervals. The energy and power usage of the execution of a function is measured by the corresponding metrics using the Running Average Power Limit [5] and pyRAPL[14]. Power consumption is measured at given intervals. To collect the CPU consumption metric the CPU usage of the running Python instance is measured in percentage at given intervals.

The latency metric of the benchmarking framework measures the time the pipeline needs to process a specified number of data points and dividing the number of data points used by this time. Similarly, the throughput metric is collected by dividing the time needed to process a specified number of data points by this number.

4 Use Cases

In this section, we present two use cases for an end-to-end ML system, including the tasks for each use case and the available data, outlining how a ML system evaluated by the benchmarking framework could be implemented.

Use Case	Data	Workload	SuT
Earth observation	Sentilel-1 & Sentinel-2 [3]	Climate zone classification	BD & ML System
Backblaze Anomaly Analysis	Backblaze Dataset [1]	Hardware Failure Prediction	ML System

Table 2: Specification of the two use cases

4.1 Earth Observation (DAPHNE UC-1)

The earth observation case study focuses on classifying local climate zones from satellite images. The classes are based on the surface structure of the area as well as anthropogenic parameters. This information can be gained from the Sentinel-1 and Sentinel-2 datasets, which include high resolution satellite imagery as well as auxiliary datasets [3]. Labeled training datasets are available. Considering the large size of the datasets and the foreseeable growth in data as well as data size, this use case presents the opportunity to evaluate the ability of the systems to complete data preparation, classification and analysis tasks at a large scale. For the model training stage, we use deep learning approaches which have proven to be an effective solution for image recognition tasks. Here, the benchmarking framework can measure the time and memory consumed by the system under test during training. The same measurements can be taken during the testing stage.

4.2 Backblaze Anomaly Analysis

The Backblaze anomaly analysis is based on publicly available and free data on the operability of hard drives provided by Backblaze [2]. For each operational hard drive there is daily information on the model, serial number, model, capacity, and possible failures available as well as SMART stats. The task at hand is to predict future hardware failures. To this end, we use both traditional machine learning methods, such as decision trees, support vector machines, and deep learning approaches. The usage of different training methods for the same task requires different data preparation techniques which can be evaluated separately. This allows us to not only collect ML-specific metrics related to the training process, but to also measure system performance during data parsing and transferring, such as the used memory and time.

5 Conclusions

The current benchmarking landscape has not adopted the convergence between systems developed in the big data, high performance computing, and machine learning domains. Integrated data analysis pipelines, as the centerpiece in these systems, cannot be comprehensively evaluated. In this report, we present an initial concept and definition of a benchmarking framework that can track and collect metrics from all stages in the IDA pipeline lifecycle. We selected a set of runtime metrics that measure the time and general resource consumption of each benchmarking stage. Additionally, for the training aspect of the IDA pipeline, we measure the model performance with respect to training time, and epochs executed.

The implementation of the framework is centered around decorator methods that are not interfering with the general runtime and can be integrated easily into existing pipelines. Finally, we include two use cases that we considered for the design of the benchmarking framework, one DAPHNE project IDA pipeline for landscape classification, and an open-source use-case on anomaly analysis. Our next steps include incorporating the framework into a working prototype that can be generally used to evaluate the performance of the DAPHNE integrated analysis system.

6 References

[1] Adolf, Robert, et al. "Fathom: Reference workloads for modern deep learning methods." 2016 IEEE International Symposium on Workload Characterization (IISWC). IEEE, 2016.

[2] Backblaze Hard Drive Data and Stats. https://www.backblaze.com/b2/hard-drive-test-data.html, accessed 2022-10-27

[3] Corpernicus Open Access Hub. https://scihub.copernicus.eu/, accessed 2022-10-27

[4] Caldas, Sebastian, et al. "Leaf: A benchmark for federated settings." arXiv preprint arXiv:1812.01097, 2018.

[5] David, Howard, et al. "RAPL: Memory power estimation and capping." 2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED). IEEE, 2010.

[6] Farrell, Steven, et al. "MLPerf™ HPC: A Holistic Benchmark Suite for Scientific Machine Learning on HPC Systems." 2021 IEEE/ACM Workshop on Machine Learning in High Performance Computing Environments (MLHPC). IEEE, 2021.

[7] Gao, Wanling, et al. "Bigdatabench: A scalable and unified big data and ai benchmark suite." arXiv preprint arXiv:1802.08254, 2018.

[8] Ghazal, Ahmad, et al. "Bigbench: Towards an industry standard benchmark for big data analytics." Proceedings of the 2013 ACM SIGMOD international conference on Management of data, 2013.

[9] Hpcg benchmark. https://icl.bitbucket.io/hpl-ai/, accessed 2022-10-27

[10] Ihde, Nina, et al. "A Survey of Big Data, High Performance Computing, and Machine Learning Benchmarks." Technology Conference on Performance Evaluation and Benchmarking. Springer, Cham, 2021.

[11] Li, Peng, et al. "Cleanml: A benchmark for joint data cleaning and machine learning [experiments and analysis]." arXiv preprint arXiv:1904.09483, 2019: 75.

[11] Mattson, Peter, et al. "Mlperf training benchmark." Proceedings of Machine Learning and Systems 2, 2020: 336-349.

[13] Narang, S. "Deepbench." https://svail.github.io/DeepBench/, accessed 2022-10-27

[14] PyRAPL. https://pypi.org/project/pyRAPL/, accessed 2022-10-27

[15] Specmpi. https://www.spec.org/mpi2007/, accessed: 2022-11-19