http://www.everest-h2020.eu

dEsign enVironmEnt foR Extreme-Scale big data analyTics on heterogeneous platforms

EVEREST

D2.6 — Definition of Data Requirements

The EVEREST project has received funding from the European Union's Horizon 2020 Research & Innovation programme under grant agreement No 957269

Project Summary Information

Project Title	dEsign enVironmEnt foR Extreme-Scale big data analyTics on heterogeneous platforms
Project Acronym	EVEREST
Project No.	957269
Start Date	01/10/2020
Project Duration	42 months
Project Website	http://www.everest-h2020.eu

Copyright

Num.	Partner Name	Short Name	Country
1 (Coord.)	IBM RESEARCH GMBH	IBM	CH
2	POLITECNICO DI MILANO	PDM	IT
3	UNIVERSITÀ DELLA SVIZZERA ITALIANA	USI	CH
4	TECHNISCHE UNIVERSITAET DRESDEN	TUD	DE
5	Centro Internazionale in Monitoraggio Ambientale - Fondazione CIMA	CIMA	IT
6	IT4Innovations, VSB – Technical University of Ostrava	IT4I	CZ
7	VIRTUAL OPEN SYSTEMS SAS	VOS	FR
8	DUFERCO ENERGIA SPA	DUF	IT
9	NUMTECH	NUM	FR
10	SYGIC AS	SYG	SK

Project Coordinator: Christoph Hagleitner – IBM Research – Zurich Research Laboratory

Scientific Coordinator: Christian Pilato – Politecnico di Milano

Disclaimer

Deliverable Information

Work-package	WP2
Deliverable No.	D2.6
Deliverable Title	Definition of Data Requirements
Lead Beneficiary	USI
Type of Deliverable	Report
Dissemination Level	Public
Due Date	31/01/2023

Document Information

Delivery Date	31/03/2023
No. pages	39
Version | Status	0.4 | Final
Responsible Person	Francesco Regazzoni (USI)
Authors	Francesco Regazzoni (USI), Tom Slooff (USI), Subhadeep Banik (USI), Martin Šurkovský (IT4I), Dionysios Diamantopoulos (IBM), Burkhard Ringlein (IBM), Antonella Galizia (CIMA – CNR), Antonio Parodi (CIMA), Riccardo Cevasco (DUF), Fabien Brocheton (NUM), Radim Cmar (SYG), Serena Curzel (PDM), Christian Pilato (PDM)
Internal Reviewer	Jeronimo Castrillon (TUD)

The list of authors reflects the major contributors to the activity described in the document. All EVEREST partners have agreed to the full publication of this document. The list of authors does not imply any claim of ownership on the Intellectual Properties described in this document.

Revision History

Date	Ver.	Author(s)	Summary of main changes
09.09.2022	0.1	Tom Slooff (USI)	Ported D2.3 from word to latex.
30.01.2023	0.2	Francesco Regazzoni (USI)	Included contributions from partners.
18.02.2023	0.3	Francesco Regazzoni (USI)	Included contributions from partners.
22.02.2023	0.4	Francesco Regazzoni (USI)	Included contributions from partners and finalization.

Quality Control

Approved by Internal Reviewer	23/03/2023
Approved by WP Leader	31/03/2023
Approved by Scientific Coordinator	31/03/2023
Approved by Project Coordinator	31/03/2023

Table of Contents

1 Executive Summary

This document updates the first version of the requirements on data from an application and environment point of view. The whole EVEREST environment is largely data driven. The analysis on the properties of the data, the way in which they have to be propagated through the EVEREST environment, and the requirements that have to be fulfilled during the computation are fundamental for designing and tuning all the blocks that will compose the EVEREST environment. The first version of this deliverable, D2.3, detailed the data requirements extracted in the first part of the project, and represented the starting point for this analysis. This deliverable updated D2.3 in various aspects. Firstly, the data requirements of the use cases have been revised and, were needed, updated to include changes and modifications that happened during the development of the project. These changes involve a more precise definition of applications and addition or removal of data-sets. Secondly, in addition to the application layer, the data requirements are specified also at a granularity of kernel or at the granularity of component. Finally, the data management policy have been updated to reflect the currently used infrastructure (whose description has been also modified and completed accordingly). General considerations done for Deliverable D2.3 are still valid here: data are only a part of the whole EVEREST environment. As such, they affect and are affected by other components of the environment. Requirements on languages, described in Deliverable D2.5 are largely influenced by the data and their requirements. Data requirements, that are extensively discussed here, are extracted from applications, thus intersect with Deliverable 2.4 that deals with the definition of the use cases. To have self-containing deliverables, we report also in Deliverable D2.4 the requirements related to data (e.g., the tables summarizing the type of data, their short description, and their size). Nevertheless, D2.4, D2.5, and D2.6 are strongly intertwined and together are the base for the development of the EVEREST environment (WP3, WP4, and WP5) and its validation in WP6. Finally, to have a self contained deliverable, we decided to write this deliverable starting directly with the content of D2.3. The main updates from that version are listed below. List of main changes between this deliverable and the first version of the “ Definition of Data Requirements"

- Section 3. Since this deliverable includes the data requirements also at a granularity of kernel and at a granularity of component, this section now include a descripton of the following kernels: RRTMG Module, Map-Matching, Traffic Prediction CNN Training, Traffic Predition Inference, PTDR, and of the following components: WRF, Aggregations, ADMS.
- Section 5. The description of the use cases and the related figures has been update according to the evolution of the project. In particular, the air quality monitor is now using only IFS data, and the IA and ADMS components will be executed on NUM servers. In the traffic modeling use case the description of the intermediate inputs and outputs have been updated and improved. For all the use case, where needed, the tables of inputs and outputs data have been updated, inserted or removed. Section 5.5 discussing the data input and output at the granularity of kernel or at the granularity of component has been addede.
- Section 6 The table of data requirement and their priority has been updated according to the project evolution.
- Section 7. This secton has been update reflecting the final computing infrastructure of the project. In particular, a detailed description of the IT4I infrastructure and of the IBB NFS support have been added.

In addition to these major changes, Introduction and Conclusions are also update reflecting the changes of the whole deliverable.

1.1 Structure of the Document

Section 2 recalls the goals of the EVEREST project. Section 3 summarizes the use cases, the workflows and the kernels from the data requirements point of view. Section 4 introduces the concept of data requirements. Section 5 discusses the data requirements in each use case, and in the related workflows and kernels. Section 6 abstracts the requirements from the use cases and derives the requirements, imposed by data, that have to be supported by the EVEREST environment. Section 7 discusses the data management policies in datacenters.

1.2 Related Documents

Deliverable D1.2 – data management plan for making the data available outside the consortium
Deliverable D2.1 – more details on the application use cases (initial version)
Deliverable D2.2 – more details on the language requirements (initial version)
Deliverable D2.3 – first version of this deliverable describing the data requirements
Deliverable D2.4 – more details on the application use cases
Deliverable D2.5 – more details on the language requirements

2 Introduction

The main goal of the EVEREST project is to propose a design environment for High Performance Big Data Analytics applications on FPGA-based systems. The EVEREST environment will be composed of a **target system**, which will seamlessly combine high-end CPUs, FPGA accelerators, and edge devices, and a **design environment**, with higher-level domain-specific abstractions that map to state-of-the-art programming models such as OpenCL, SYCL, or OpenMP. The environment handles the complexity of the underlined architecture and perform local and global optimizations. Integration and optimization on such a large and heterogeneous environment should be necessarily data driven. Data format and data location, when strictly enforced, will impose the constraints and will indirectly authorize or forbid certain optimizations. Requirements in terms of real time and minimum throughput will directly affect the selection of the target platforms and will force the use of certain communication links. Other non-functional requirements on data, such as authentication and confidentiality, will be fundamental to enable the safe and privacy-preserving execution of big data analytics. This deliverable reports the final requirements on data, focusing on data allocation, data communication, and data protection. We will begin summarizing the use cases of the project, then we will provide a detailed analysis of the data requirements within each use case and also at the granularity of kernel and at the granularity of component. Finally, we will abstract these use case specific requirements to define more general data requirements that should be supported by the EVEREST environment.

3 Summary of EVEREST Use Cases

Design and optimizations within the EVEREST will be driven by data. To abstract the constraints and the requirements on the EVEREST environment, we begin by analyzing the requirements on the data used in each of the use cases of the project. The detailed description of each use case is provided in Deliverable D2.1. To be able to understand and abstract the data requirements, in this section, we quickly summarize each use case. All use cases combine heterogeneous data sources to achieve their respective goals.

3.1 Renewable Energy Prediction

The goal of this use case is to reduce the cost of imbalance production of energy from renewable sources. Imbalances are often caused by severe meteorological events, which can be mitigated and even avoided if these events are correctly predicted. Currently, forecast of renewable energy production is based on models having a coarse grain spacing grid (between 15 km and 25 km). This use case uses **meteorological data** and **weather models** to improve the forecast of the energy produced by a wind farm within the next 24 hours, with an hourly prediction granularity. This resolution will be achieved by analyzing real-time weather data, to obtain high-localized meteorological predictions, and combining these predictions with advanced artificial intelligence algorithms to forecast the energy production.

3.1.1 Component: WRF

The Weather Research and Forecasting (WRF) Model is a state-of-the-art mesoscale numerical weather prediction system designed for both atmospheric research and operational forecasting applications. It features two dynamical cores, a data assimilation system, and a software architecture supporting parallel computation and system extensibility. In the EVEREST project, the Advanced Research WRF – ARW dynamical core is exploited. The model serves a wide range of meteorological applications across scales from tens of meters to thousands of kilometers. Due to the high computational and memory requirements, HPC resources are mandatory to run the simulations.

3.1.2 Kernel: RRTMG module

Radiative processes are among the most complex and computationally intensive parts of all model physics. As an essential component of modeling the atmosphere, radiation, directly and indirectly, connects all physics processes with model dynamics, and it regulates the overall earth-atmosphere energy exchanges and transformations. This is true also for the WRF model since it accounts for about 20-30% of the execution time. The RRTMG kernel provides an implementation of such a physics process utilizing the correlated-k approach to determine the total radiative flux at any given location.

3.2 Air Quality Monitoring

Concentration and spreading of the pollution produced by industry and productive sites is largely influenced by weather conditions. This use case aims at forecasting the environmental impacts of chemical pollutants of industrial sites, and to use such a forecast for regulating the emissions during production and for adapting the production accordingly. To achieve the high precision needed, we will combine **high-resolution weather ensembles** with **local data** collected in real-time. Weather data will be collected with a radius of few kilometers surrounding the industrial plant and feed to artificial intelligence algorithms to support decisions on the productions schedule.

3.2.1 Component: Aggregation

Each day, this component receives a stream of two kinds of data: (i) the first one contains weather location observations from a measurement station at an industrial site, and (ii) the second one consists of numerical weather forecasts (twice per day, 48 hours of forecast) for the same meteorological parameters for this location. It is an ensemble of numerical forecast from various origins: WRF performed by CIMA on EVEREST infrastructure for France domain (output of WRF kernel for France), WRF performed by NUM on NUM servers for France and Europe. This component performs an ensemble aggregation forced by the local observation (the mathematical method is a ridge regression with discounted loss method, with objective to minimize the root mean square error between the aggregated prediction and the observation over a training period). The output is a more robust weather forecast for the next 48 hours.

3.2.2 Component: ADMS

Each day, this component receives as input a set of (i) emission forecast for various pollutants for the next 48 hours for an industrial site and (ii) weather numerical forecast for the next 48 hours (output of aggregation component) and performs air dispersion calculation (using last-generation gaussian modelling) over a grid of 10×10 km^2 in order to get map of pollution for the next 48 hours.

3.3 Traffic Modelling

The goal of traffic modeling and prediction system is to minimize the congestion and to optimize the flow of traffic, ultimately reducing the travel time and, as direct consequence, the pollution. In this use case, we concentrate on traffic modeling and prediction within cities. The model will be built combining **real-time sensory data** with **long history records**. Traffic simulator will be used to boost the raw sensory data dataset into rich training sequences that will be then used to train traffic prediction models. The results of this prediction model will be used to provide an advanced route calculation, that will be offered as a service to the customers. Challenges related with this use case are related to the number of vehicles monitored in parallel (vehicle positions are collected approximately every 5 seconds) resulted in a cumulated large data set further processed with the combination of both macro and microscopic approaches.

3.3.1 Kernel: Map-Matching

The map-matching kernel is the part of the FCD (floating car data) workflow and it is the vector-in vector-out processing, which converts the sequence of time-stamped GPS positions of a vehicle into road speeds linked to road segments. Using HMM (Hidden Markov Model), Viterbi decoder and Dijkstra routing components it calculates the most probable vehicle path on the road network from noisy GPS coordinates. The goal is to execute this conversion for millions of vectors daily with energy efficiency.

3.3.2 Kernel: Traffic Prediction CNN Training

The traffic prediction training kernel is a typical deep neural network component, which learns the prediction of road speeds for a coming hour based on the current and short term history speeds information on a surrounding roads. The supervised learning uses 17 data input and 4 data outputs. The training needs to be executed on each major road in a city to cover a complete traffic picture. For a medium size city it is around 10,000 roads, meaning 10,000 prediction models.

3.3.3 Kernel: Traffic Prediction Inference

The traffic prediction kernel is a part of the Routing workflow and it is a many vector-in many vector-out processing, which calculates the predicted traffic view for the complete city in one shot. It uses the outcome of training workflow returning the coefficients of a neural network for each road. Considering a medium-size city it calculates 10,000 speed prediction vectors for 4 future time points (15min, 30min, 45min, 60min) for each major road.

3.3.4 Kernel: Probable delay on route at departure time (PTDR)

The PTDR kernel is a part of the Routing workflow. It has the same interface as the kernel 3.3.3 (Traffic Prediction Inference) so they can be used interchangeably, more precisely the PTDR kernel can be exchanged for the traffic predictor. Nevertheless, before having a trained traffic predictor the PTDR kernel is used within the simulator to determine a probable delay on a route at departure time. Within the traffic simulator we can cover the situation not observed in real traffic, hence boost the data for training. The kernel is called in every re-routing request; for example in simulator there are thousands of vehicles simulated at once, each vehicle is allowed to continue for 20s and then ask for re-routing.

4 The Concept of Data Requirements

In this section, we introduce the concept of data requirements, the idea that is behind it, and the way in which we collected and organized the data requirements. Data requirement is a well-known concept in business analysis and management. In that context, data are collected and organized mostly for supporting the decision-making process and to support the management process. In the context of EVEREST, we define "data requirements" as the collection of 1) all the **characteristics** of the data handled by the EVEREST environment and 2) the **properties** that should be guaranteed when handling the data within the EVEREST environment. As we can see, data requirements are not only "requirements" in the strict sense, i.e., they are not only direct requirements for specific components of the EVEREST platform. For us, data requirements are more a collection of facts (such as "data entering in this computation are in the range of giga bytes") or properties (such as "data confidentiality should be guaranteed"). However, these facts and properties impose, also indirectly, requirements on the EVEREST environments (e.g., in terms of bandwidth for transferring giga bytes of data, or in terms of encryption functionalities that must be present in the system).

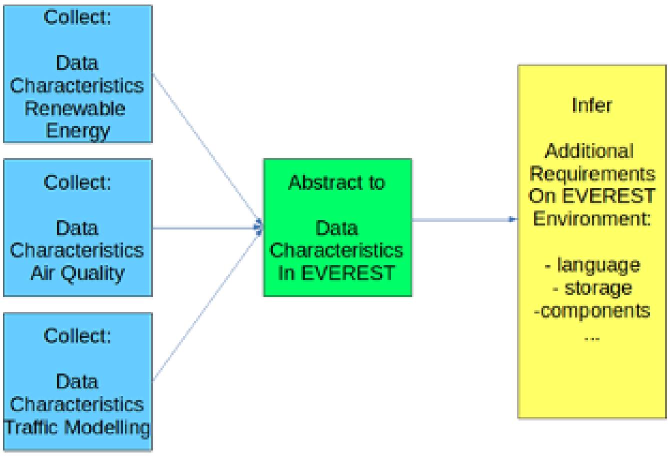

Figure 1 – Process of data requirements

Figure 1 depicts the flow used in EVEREST to collect, abstract, and process the data requirements. Initially, we collect the characteristics of data belonging to each application. This process is similar to the one followed to define other requirements and begins with the requirements elicitation. In our case, the elicitation phase has been carried out with dedicated meetings, where the use cases have been extensively discussed from the data point of view. Among other properties, we collected information about the size of the data, the frequency of the updates, the amount needed to start the computation, the amount of data produced by each computation, the need of confidentiality, integrity, and authentication of the data, the reliability of the source and the format of the data, etc.

This information has been collected and analyzed, and from this, we abstracted the characteristics of the data within EVEREST. The final step was to derive, from these abstracted characteristics, additional requirements for the EVEREST environment. All these steps are detailed in the next sections.

5 Data Properties within the Use Cases

In this section, we present the properties of the data of each use case. For each use case, we discuss the amount of data that are needed to carry out the computation, the type of the computation, where the data are located, and if there are any security requirements to them. We do so from the application point of view, considering the generic application, not the specific data that are used in this project. We will start this section discussing the way in which data requirements are presented. We will then enter into the details of each use case, reporting in descriptive text the properties collected during the dedicated phone conferences. We will conclude summing up the collected information in a tabular form.

5.1 Data Analysis Approach

For each use case, we split the analysis of the data properties in eight axes. In this section we introduce and detail each of them, presenting how each voice would affect the decisions in the EVEREST environment.

Type of data: in general, the type of data is not expected to affect significantly the EVEREST environment. Nevertheless, specific type of data, stored or loaded in a specific format, could require a conversion or a long loading procedure. Also, the use of different types (heterogeneous data sources) will require to understand how to combine them also from the semantic viewpoint.

Size of the data: the size of the data is fundamental since it directly impacts the time needed to process them. Furthermore, it would affect the memory and storage units that will constitute the EVEREST system. Here we report the order of magnitude of the input and output data.

Data transfer within the application: here we discuss the expected speed of data transfer within the use case. The information on data transfer collected here are necessary to drive the optimization.

Loading data in the system: here we discuss how the data are loaded into the system and how the results of the computation is accessed from the external world. This will affect the interface (e.g., the protocols) between the EVEREST environment and the external world.

Location of the data and of computation: here we discuss where the data are located on the scenario envisioned for the use case and where the computation is carried out. The location could be, for instance, on the EVEREST environment or outside it, for instance on a server of a project partner that does not belong to the EVEREST environment. Current location of data is important to identify where data relocation is possible, and to enable optimizations aimed at limiting the data transfer.

Timing requirements on data and computation: here we present which ones are the timing requirements on data and computation, meaning the frequency when the data should be read/processed by the next step in the computation, but also the frequency in which the data are produced and need to be stored and the approximate time needed by each computation. This would affect the allocation of the computational elements.

Reliability properties associated with data: here we present, in case they are present, properties related to reliability of the data, such as the need to ensure the correctness of the data, the fact that the data could not be reliable and needs to be checked or verified, the need of error correction codes to ensure the transfer. This would affect the way in which internal communication is designed and implemented.

Security properties associated with data: here we report the security feature associated with the data, which will be used to infer mechanism for data protection. We consider here the need, when relevant, to fulfill requirements in terms of authenticity of the data (parties producing the data should be identified), confidentiality of the data (only authorized parties should access them), availability of the data (data should be available when needed). We consider these aspects both in the computation and in the transfer of the data, since these are the aspects of the EVEREST environment that will be affected by this requirement.

5.2 Renewable Energy Prediction

Figure 2 depicts the case of study of energy prediction. In the figure we report where the input data are entering in the computation and where a result is produced, and we discuss the data flow within this use case. In the rest of this section, we will present the requirements on data in this use case.

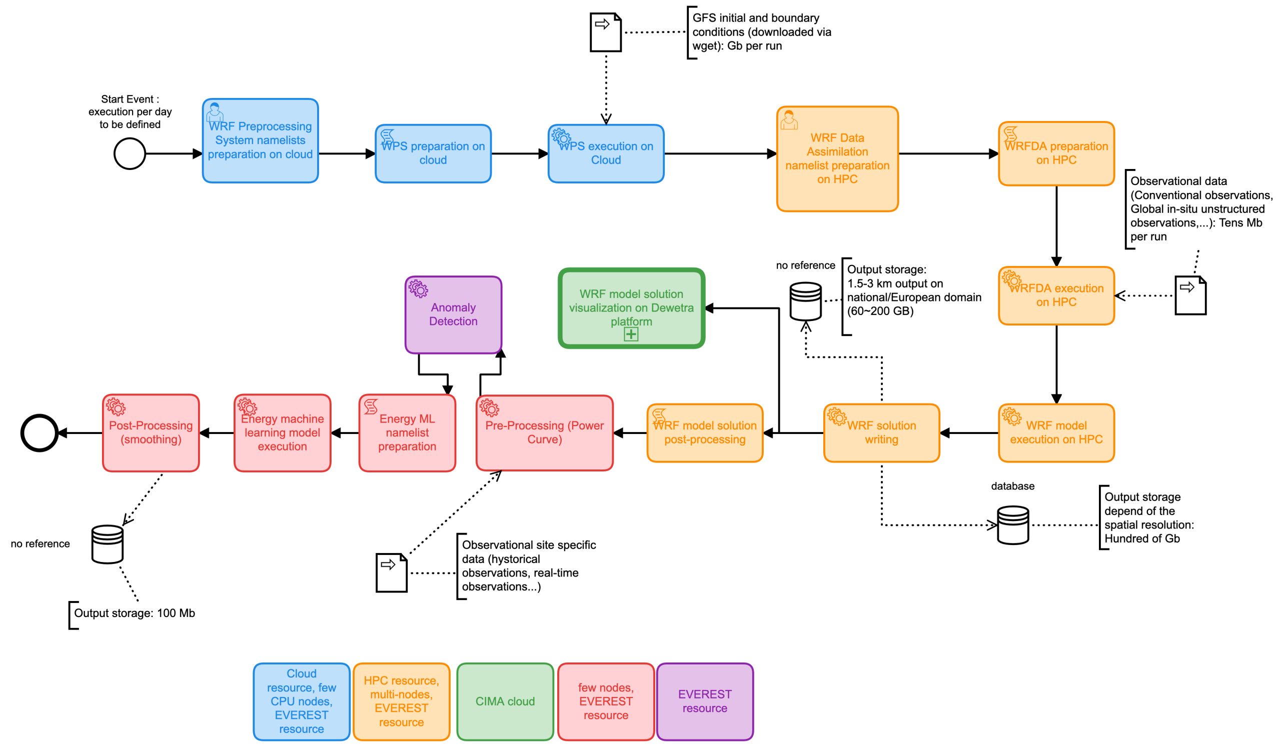

Figure 2 – Renewable Energy Prediction Use Case.

Type of data: we load three types of data in this use case. Initial and boundary conditions, observational data, and observational site-specific data. The initial and boundary conditions include GFS (Global Forecasting System) and IFS (Integrated Forecasting System) data. They are stored and passed to the system in the format Grib2. Observational data are unstructured data that store observational data collected from different providers and sources. The last type of data are observational data specific to the target location. The use of these data type does not impose particular constraints to the system. The data produced by the computation of this use case are of two types: the WRF update, and the energy prediction data. To allow the correct loading of the data into the system, it is however necessary to ensure that all the needed parsing features are provided.

Size of the data: the size of data storing initial and boundary conditions is in the range of Gb per run. The observational data that are used during the HPC computation are in the range of tens of Mb per run. The site-specific observation used to forecast the energy production are in the range of few Mb per computation. The computation produces hundreds of Gb per execution, while the data related to the energy forecast are approximately few Mb per execution. Data that will be used for weather forecast and for all the related training need to be stored in a database. The database should be sized sufficiently to store data of several months.

Data transfer within the application: The data transfer within the use case varies, depending on the task. From the right part of Figure 2, we can see that the computation related to the weather forecast prediction, requires transfer of quite a large amount of data, in the order of tens of Gb per computation. This amount goes up to 100 Gb in the phase where initial and boundary conditions are prepared and transferred. After the WRF data selection, in the final phases of the computation, where the forecast data are combined to the site-specific data to estimate the energy generated, the amount of data that is transferred drops to Mb per computation for one wind farm.

Loading data in the system: the initial conditions are loaded either using dedicated APIs or using classical network communication protocols, such as wget. The local weather forecast data and the local observation data are loaded into the system by FTP or other state-of-the-art transfer protocols.

Location of the data and of computation: input data are loaded from external sources. The data produced during the computation are stored on databases. Generally, there are no strict requirements on the location of the databases storing the output of the computation. As a result, when convenient, the databases could even be integrated within the EVEREST environment. It is however required that the updated WRF data are stored in a location reachable within European or national domains. The colors of the different blocks of Figure 2 describe the location of the computation. The green color identifies the portion of the use case that is located on CIMA servers. This is mostly related with visualization aspects. The red, yellow, blue and purple blocks identify the computations that are carried out (or that can be potentially carried out) within the EVEREST environment. The red blocks require cloud resources with few CPUs and can be executed either on the EVEREST servers or on the servers of DUFERCO. The blue blocks require cloud resources with few CPUs and are executed on the EVEREST servers, while the yellow blocks require HPC resources and need to be executed on the EVEREST environment.

Timing requirements on data and computation: input data to the computation are produced with a time frame that ranges between seconds and tens of minutes. Data related with in situ observation and data related with local weather forecast are generated with the frequency of seconds. New observational data are produced within minutes, while new initial and boundary condition data are loaded in the computation within tens of minutes. The time required to carry out the computation is indicated in red on top of each block in Figure 2, and varies between minutes and seconds, with the only exception of the WPS execution on the cloud that takes between tens of minutes to hours.

Reliability properties associated with data: data that are loaded from external sources are assumed to be correct and no verification on them is required in addition to the one required by the used transfer protocols.

Security properties associated with data: loading of the data is assumed to always happen from trusted parties, and there is no possibility for an adversary to inject malicious data with the goal of altering the AI algorithms used during the computation. Because of this, adversarial attacks or similar attacks aiming at altering the behavior of the computation are not considered and we do not need to protect from them. The computation also happens on trusted premises. However, we need to ensure security and privacy of the data in all the steps of the internal communication. The communication between each block of computation should thus be encrypted and authenticated, to ensure confidentiality of the data and to authenticate them, using the appropriated algorithms and protocols. The flow of information within the system should also be checked and monitored, to ensure that the data, loaded as trusted, are not corrupted by the interaction (even unwanted) with untrusted data.

The main characteristics of input and output data are summarized in the following tables.

INPUT DATA

ID	ITEM	DESCRIPTION
D1	Dataset name and reference	Global Forecasting System (GFS)
	Dataset size	1 Gb (analysis and 48 hours boundary conditions at hourly temporal resolution and 25 km grid spacing)
	Dataset description	Initial and boundary conditions data
	Standards, format and metadata	Grib2 format
	Archiving and preservation (including storage and backup)	During the project: input for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for model initialization. After the project: data will be stored on the CIMA storage system.

Table 3 – Renewable Energy Prediction use case – dataset GFS

ID	ITEM	DESCRIPTION
D3	Dataset name and reference	Wunderground stations
	Dataset size	Kbytes for one year for one station
	Dataset description	Personal weather stations data (2 m temperature, 10 m wind, 2 m relative humidity, surface pressure, rainfall intensity and rainfall depth) at hourly temporal resolution
	Standards, format and metadata	ASCII format
	Archiving and preservation (including storage and backup)	During the project: input for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for model data assimilation. After the project: data will be stored on the CIMA storage system.

Table 4 – Renewable Energy Prediction use case – dataset Weather Underground stations

ID	ITEM	DESCRIPTION
D4	Dataset name and reference	Italy authoritative weather stations
	Dataset size	Kbytes for one year for one station
	Dataset description	Personal weather stations data (2 m temperature, 10 m wind, 2 m relative humidity, surface pressure, rainfall intensity and rainfall depth) at hourly temporal resolution
	Standards, format and metadata	ASCII format
	Archiving and preservation (including storage and backup)	During the project: input for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for model data assimilation. After the project: data will be stored on the CIMA storage system.

Table 5 – Renewable Energy Prediction use case – dataset Authoritative Weather Stations

ID	ITEM	DESCRIPTION
D5	Dataset name and reference	Italian Civil Protection radar mosaic
	Dataset size	12 Mb (24 hours data at hourly temporal resolution, 4 CAPPI levels)
	Dataset description	Radar reflectivity cappi at 2000, 3000, 4000 and 5000 m
	Standards, format and metadata	GeoTIFF format
	Archiving and preservation (including storage and backup)	During the project: input for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for model data assimilation. After the project: data will be stored on the CIMA storage system.

Table 6 – Renewable Energy Prediction use case – dataset Radar mosaic

ID	ITEM	DESCRIPTION
D7	Dataset name and reference	Historical power generation (reference will be updated)
	Dataset size	About 200 KB per wind farm (1 year)
	Dataset description	Wind farm specific on hourly basis [MWh] for models training and calibration; minimum 1 year, possibly 2 years or more
	Standards, format and metadata	CSV format
	Archiving and preservation (including storage and backup)	During the project: inputs for the selected periods will be stored for the duration of the EVEREST project. After the project: inputs will be stored on DUF storage system. Sample data may be archived for future research.

Table 7 – Renewable Energy Prediction use case – dataset historical power generation

ID	ITEM	DESCRIPTION
D8	Dataset name and reference	Historical power availability (reference will be updated)
	Dataset size	About 2 MB per wind farm (1 year)
	Dataset description	Wind farm specific on hourly basis unavailability [MW or %] due to maintenance or failure, for models training and calibration; minimum 1 year, possibly 2 years or more
	Standards, format and metadata	CSV format
	Archiving and preservation (including storage and backup)	During the project: inputs for the selected periods will be stored for the duration of the EVEREST project. After the project: inputs will be stored on DUF storage system. Sample data may be archived for future research.

Table 8 – Renewable Energy Prediction use case – dataset historical power availability

ID	ITEM	DESCRIPTION
D9	Dataset name and reference	Historical TSO curtailments (reference will be updated)
	Dataset size	About 100-200 KB per wind farm (1 year)
	Dataset description	Wind farm specific capacity limitations by TERNA (Italian Transmission System Operator); on hourly basis data [MW] for models training and calibration; minimum 1 year, possibly 2 years or more
	Standards, format and metadata	CSV format
	Archiving and preservation (including storage and backup)	During the project: inputs for the selected periods will be stored for the duration of the EVEREST project. After the project: inputs will be stored on DUF storage system. Sample data may be archived for future research.

Table 9 – Renewable Energy Prediction use case – dataset historical curtailments

OUTPUT DATA

ID	ITEM	DESCRIPTION
D11	Dataset name	WRF output
	Dataset size	7 Gb (cloud permitting grid spacing, 24 hours forecast at hourly time resolution, 500x500x50 grid points)
	Dataset description	2D and 3D meteorological fields, over different locations at hourly temporal resolution, for a selected period.
	Standards, format and metadata	NetCDF format
	Archiving and preservation (including storage and backup)	During the project: Outputs for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for renewable energy application. After the project: Outputs will be stored on the CIMA storage system. Sample data may be archived for future research.

Table 10 – Renewable Energy Prediction use case – dataset WRF model

ID	ITEM	DESCRIPTION
D13	Dataset name	Wind power forecast predicted with Machine Learning
	Dataset size	About 10 KB per wind farm (72 hr)
	Dataset description	Wind farm specific on hourly basis [MWh] for models training and calibration; minimum 1 year, possibly 2 years or more
	Standards, format and metadata	CSV format
	Archiving and preservation (including storage and backup)	During the project: outputs for the selected periods will be stored for the duration of the EVEREST project. After the project: outputs will be stored on DUF storage system. Sample data may be archived for future research.

Table 11 – Renewable Energy Prediction use case – dataset energy prediction (ML)

5.3 Air Quality Monitoring

Figure 3 depicts the whole air quality use case and reports all the requirements on data as initially forecasted. The core of the computation of the air quality use case is similar to the one of the renewable energy prediction use case, but, the data used (and so their size and the main properties) are different. We will discuss the data requirements related to the air quality monitoring use case in this sub-section. From this figure, there are two main updates with respect to the initial planning: **Initialisation and boundary conditions of WRF simulation:** Due to uncertainties on ECMWF agreement to use IFS forecast for the EVEREST project, we had anticipated an alternative way to use the GFS forecast. Fortunately, an ECWMF agreement has been obtained, so only IFS will be used, which is known to have a better quality over Europe.

IA and ADMS execution: The two executions will be deployed on NUM servers. This is to more easily manage in a secure way the transfer of full WRF data from EVEREST infrastructure to externals servers such as the NUM ones.

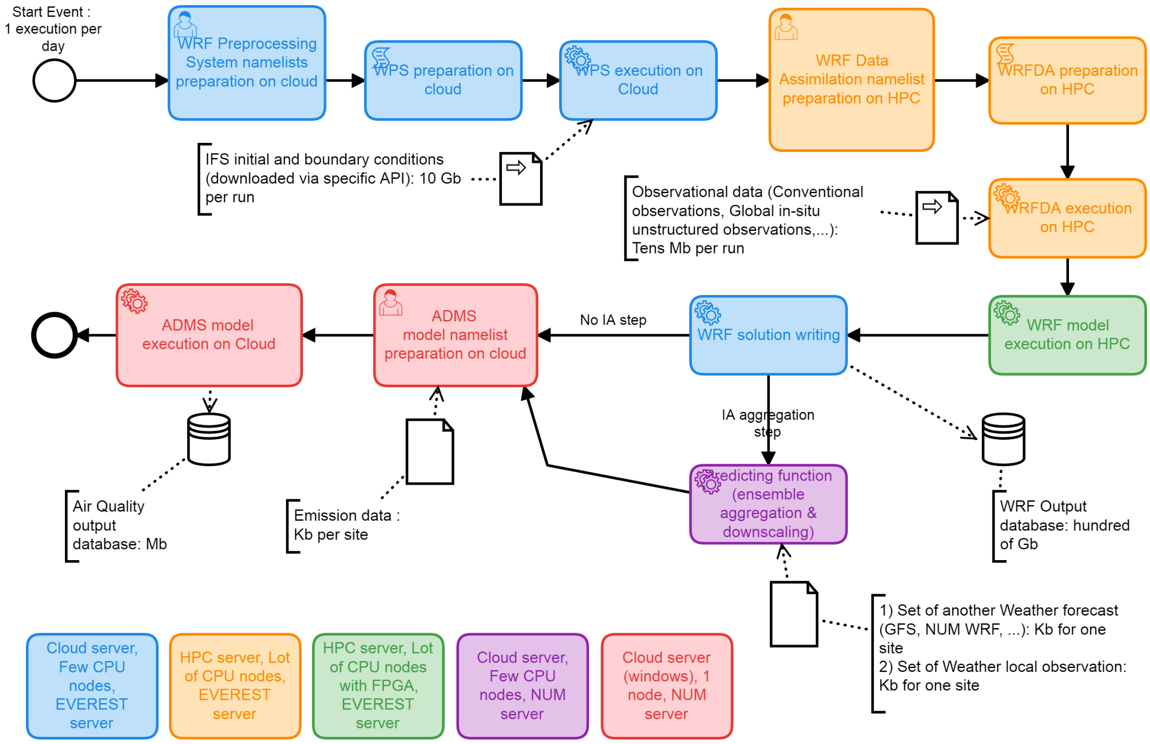

Figure 3 – Air Quality Monitor Use Case.

Type of data: the data that are loaded in this use case belong to four groups: initial and boundary conditions, observational data, weather forecast data and local weather observational data, and emission data. The initial and boundary conditions use outputs from IFS forecasts. Observational data are unstructured data that store observational data collected in situ. The weather forecast data are two different type of prediction data that are needed to predict weather conditions close to the monitoring location. The last type of data input are data storing information about the atmospheric emissions on the industrial site under monitoring. The data that are produced by the system are of two types: an updated set of weather forecast data and the data related to the air quality around the location under monitoring. Except for the correct parser for loading the data in the proper format (and, correspondingly, to store them) the data type of this use case does not impose any specific requirement.

Size of the data: the size of data storing initial and boundary conditions is in the range of 10 Gb per run (IFS). The observational data needed to carry out the computation are in the range of tens of Mb per run. The weather forecast data have a size is in the range of Kb per run, for each site on which the computation is focused. The data related to the emission of the site under monitor are also in the range of Mb per monitored location. The WRF data produced are in the range of tens of Gb per execution, while the data related to the air quality at the location are in the range of Mb per execution. As in the previous use case, also here data that will be used for weather forecast and for all the related training need to be stored in a database. Also in this use case, the database should be sized sufficiently to store data of few months.

Data transfer within the application: The data transfer within the use case varies, depending on the task. From the right part of Figure 3, we can see that the computation related to the weather forecast prediction, requires transfer of quite a large amount of data, in the order of tens of Gb per computation. This amount goes up to 100 Gb in the phase where initial and boundary conditions are prepared and transferred. In the final phases of the computation, where the forecast data are combined to the emission data to estimate the air quality, the amount of data that is transferred drops to Kb per computation for one industrial site.

Loading data in the system: as in the case of the renewable energy forecasts, the initial conditions are loaded either using dedicated APIs or using classical network communication protocols, such as wget. The local weather forecast data, the local observation data and emission data are loaded into the system by FTP.

Location of the data and of computation: input data are loaded from external sources. The data produced during the computation are stored on databases. From the use case point of view, there are no strict requirements on the location of these database, that can thus be within the EVEREST environment or outside at a location of a project partner. The location of the computation is identified in Figure 3 by the color key. The green blocks, that perform the external visualization of the results of the computation, are located on CIMA or NUM systems. The full WRF execution (WPS, WRFDA and WRF) are located on the EVEREST infrastructure. As indicated above, we decided to move the IA and ADMS execution from the the EVEREST environment to NUM environment, since the computation load is small. This also allow testing the interfaces to exchange large amount of data between two different geographical locations. From the figure, it is also possible to differentiate the type of the service that is needed to carry out the computation. They include cloud servers with one CPU (red), cloud servers with few CPUs (blue and purple), HPC servers with some CPUs (yellow) and HPC servers with a significant number of CPUs and FPGA (white).

Timing requirements on data and computation: Input data to the computation are produced with a time frame that ranges between seconds and tens of minutes. Data related with in situ monitoring of emissions and data related with local weather forecast are generated with the frequency of seconds. New observational data are produced within minutes, while new initial and boundary condition data are loaded in the computation within tens of minutes. The time required to carry out the computation is indicated in red on top of each block in Figure 3, and varies between minutes and seconds, with the only exception of the WPS execution on the cloud that takes between tens of minutes to hours.

Reliability properties associated with data: as in the previous case, also all the data used in this use case are loaded from external sources that are assumed to be correct and no verification on data is required in addition to the one already present in the used transfer protocols.

Security properties associated with data: also in this case the situation is similar to the renewable energy forecast use case. The loading of the data is assumed to always happen from trusted parties with no possibility of injecting malicious data. The computation happens on trusted premises, but we need to ensure confidentiality and authentication of the data in all the steps of the internal communication, and we need to ensure the monitoring of the flow of information. Since IA and ADMS execution is happening outside the EVEREST environment, it is necessary to secure the external communication between EVEREST environment and NUM environment.

The main characteristics of input and output data are summarized in the following tables. Dataset previously named D14 and which refers to GFS data for initialisation of WRF forecast is no more exploited in the use case.

INPUT DATA

ID	ITEM	DESCRIPTION
D15	Dataset name and reference	Integrated Forecasting System (IFS)
	Dataset size	3 Gb (analysis and 48 hours boundary conditions at hourly temporal resolution and 9 km grid spacing)
	Dataset description	Initial and boundary conditions data
	Standards, format and metadata	Grib2 format
	Archiving and preservation (including storage and backup)	During the project: input for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for model initialization. After the project: data will be stored on the CIMA storage system.

Table 12 – Air quality use case – dataset IFS

ID	ITEM	DESCRIPTION
D16	Dataset name and reference	Wunderground stations
	Dataset size	Kbytes for one year for one station
	Dataset description	Personal weather stations data (2 m temperature, 10 m wind, 2 m relative humidity, surface pressure, rainfall intensity and rainfall depth) at hourly temporal resolution
	Standards, format and metadata	Ascii format
	Archiving and preservation (including storage and backup)	During the project: input for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for model data assimilation. After the project: data will be stored on the CIMA storage system.

Table 13 – Air quality use case – dataset Weather Underground stations

ID	ITEM	DESCRIPTION
D17	Dataset name and reference	Local weather observation from industrial site
	Dataset size	Kbytes for one year
	Dataset description	Weather parameters data (2m temperature, 10 m wind,...) at hourly temporal resolution measured at local site of each industrial site
	Standards, format and metadata	Binary format
	Archiving and preservation (including storage and backup)	During the project: input for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for machine learning aggregation.

Table 14 – Air quality use case – dataset local weather observation from industrial site

ID	ITEM	DESCRIPTION
D18	Dataset name and reference	Emission data from industrial site
	Dataset size	Kbytes for one year
	Dataset description	Emission parameters data (pollutants concentrations at emission, emission temperature, velocity of release, ...) at hourly temporal resolution measured for the different sources of each industrial site
	Standards, format and metadata	Ascii format
	Archiving and preservation (including storage and backup)	During the project: input for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for air-quality model forecast

Table 15 – Air quality use case – Emission data from industrial site

ID	ITEM	DESCRIPTION
D19	Dataset name and reference	Deterministic NUM weather forecast for each industrial site
	Dataset size	Mbytes
	Dataset description	Weather parameters forecast data (2m temperature, 10 m wind,...) for each industrial site at hourly temporal resolution measured selected period
	Standards, format and metadata	Binary format
	Archiving and preservation (including storage and backup)	During the project: input for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for machine learning aggregation.

Table 16 – Air quality use case – NUMTECH weather forecast at industrial site

OUTPUT DATA

ID	ITEM	DESCRIPTION
D20a	Dataset name and reference	WRF output (reference will be updated)
	Dataset size	4 Gb (cloud permitting grid spacing, 48 hours forecast at hourly time resolution, 500x500x50 grid points) - Kbytes for the extraction for one site
	Dataset description	2D and 3D meteorological fields, over different locations at hourly temporal resolution, for a selected period.
	Standards, format and metadata	NetCDF format for full dataset, and Binary for extraction for one site
	Archiving and preservation (including storage and backup)	During the project: Outputs for the selected periods will be stored for the duration of the EVEREST project, with a focus on parameters used for air quality application. After the project: Outputs will be stored on the CIMA storage system. Sample data may be archived for future research.

Table 17 – Air quality use case –WRF model output

ID	ITEM	DESCRIPTION
D20b	Dataset name and reference	Aggregated Weather forecast (reference will be updated)
	Dataset size	Kbytes
	Dataset description	1D meteorological field, over one location, at hourly temporal resolution for forecast of 48 hours.
	Standards, format and metadata	Ascii format
	Archiving and preservation (including storage and backup)	During the project: Stored on NUM server. After the project: Sample data may be archived for future research.

Table 18 – Air quality use case – Weather Aggregation model output

ID	ITEM	DESCRIPTION
D21	Dataset name and reference	ADMS output (reference will be updated)
	Dataset size	2 Mb (48 hours forecast for a 10km*10km grid and various pollutants)
	Dataset description	2D air-quality concentrations over different locations (France and Italy) at hourly temporal resolution, for a selected period.
	Standards, format and metadata	ASCII format
	Archiving and preservation (including storage and backup)	During the project: Outputs for the selected periods will be stored for the duration of the EVEREST project, with a limited set of pollutants. After the project: Sample data may be archived for future research.

Table 19 – Air quality use case - ADMS model output

5.4 Traffic Modeling

Figure 4 depicts the whole traffic modeling use case and summarizes the properties and the requirements on the data used in it. In this sub-section we will present them in detail.

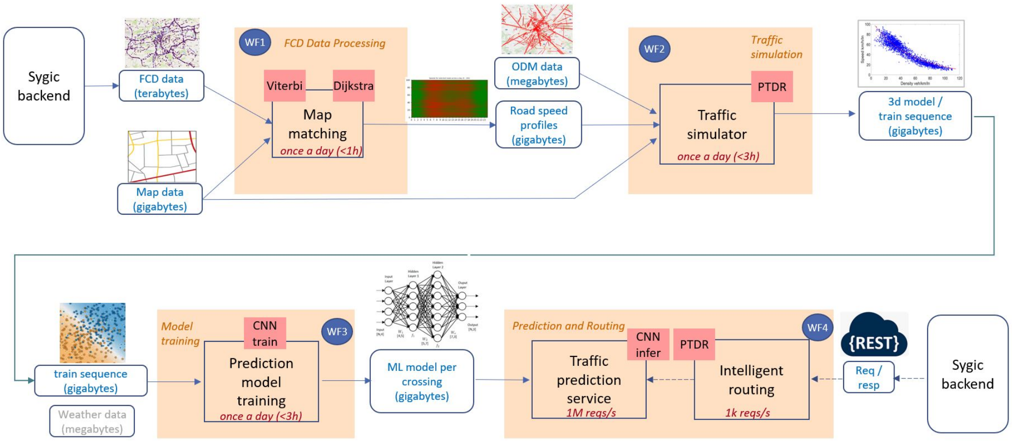

Figure 4 – Traffic Modeling Use Case.

Type of data: in this use case we load three input data sources: 1) FCD data, that are data storing the GPS position of the vehicle and its speed, 2) ODM data, that are describing the mobility flow in a target city, and 3) the map of a target city. The intermediate outputs are 1) map-matched road speed data, 2) sequences for prediction model training and 3) traffic model profile. The request/response operation of the system is to provide the routes to be followed by vehicles, which are calculated based on the traffic model and the prediction service. Also in this case, as in the previous two, the type of data used requires only to provide the right parsing capabilities.

Size of the data: FCD data are intermediate long-term storage subject to daily updates. They enter into the system in groups of gigabytes, and then they are accumulated in the EVEREST environment. After accumulation, data size is counted in terabytes. The OMD data are in the range of megabytes, while the map information is in the range of gigabytes. These data require long-term storage unit that allows updates. The request/response routing operation has the size of few kilobytes. This use case also stores the following three main intermediate data: the training sequences, traffic profile and the ML prediction models. They are in the range of gigabytes up to terabytes]. Concerning the type of storage, the training sequences require a short-term storage, while the models require a long term storage with a possibility of updating.

Data transfer within the application: The size of the data transfer within the use case is exactly as the size of the data themselves, thus ranges from terabytes (the accumulated FCD data) to megabytes for ODM data and ML models.

Loading data in the system: the data are loaded in the system and offloaded from the system via the **SYG** servers. Loading and offloading of the data is carried out using state of the art internet protocols (including, for instance, https).

Location of the data and of computation: data are stored in dedicated databases. There are no specific requirements about the location of these databases, it is however preferred that intermediate data of the computation are stored inside the EVEREST environment. The computation of the traffic model and the route calculation will happen inside the EVEREST environment. Each of these computations is envisioned to be carried out on HPC or in cloud, while FGPA will be used for significant acceleration of algorithmic kernels.

Timing requirements on data and computation: input data are partially updated every day. The size of these partial updates is in the range of gigabytes. Furthermore, the input data are fully updated on a monthly basis. The traffic model and prediction are recomputed once a day, and the computation requires several hours. We expect to handle approximately 1000 routing requests per second.

Reliability properties associated with data: as in the other two use cases, all the data used in the traffic prediction use case are assumed to be correct and no verification on data is required in addition to the one already present in the used transfer protocols.

Security properties associated with data: in this case, the situation is similar to the renewable energy forecast and air quality use case. The loading of the data always happens from **SYG** backend and the queries to request routing information are also carried out via the **SYG** backend. Because of this we consider the data always trusted. The computation runs on the EVEREST environment that, as in the previous cases, we also consider trusted. We however need to ensure confidentiality and authentication of the data in all the steps of the internal communication, and we need to ensure the monitoring of the flow of information also in this case.

The main characteristics of input and output data are summarized in the following tables.

INPUT DATA

ID	ITEM	DESCRIPTION
D22	Dataset name and reference	FCD data input
	Dataset size	Gigabytes
	Dataset description	Raw data of vehicle speeds and positions
	Standards, format and metadata	CSV
	Archiving and preservation (including storage and backup)	No Data are processed and converted into more suitable form. However, an intermediate storage will be needed.

Table 20 – Traffic monitoring use case – FCD data input

ID	ITEM	DESCRIPTION
D23	Dataset name and reference	ODM data input
	Dataset size	Megabytes
	Dataset description	Traffic flow characterization on a set of origin-destination points
	Standards, format and metadata	CSV
	Archiving and preservation (including storage and backup)	Yes, Data should be preserved till next update comes

Table 21 – Traffic monitoring use case – ODM data input

ID	ITEM	DESCRIPTION
D24	Dataset name and reference	Map data input
	Dataset size	Gigabytes
	Dataset description	Road network topology
	Standards, format and metadata	Annotated graph
	Archiving and preservation (including storage and backup)	Data should be preserved till next update comes

Table 22 – Traffic monitoring use case – Map data input

OUTPUT DATA

ID	ITEM	DESCRIPTION
D25	Dataset name and reference	Road speed data
	Dataset size	Gigabytes up to Terabytes
	Dataset description	Map matched FCD data into road speeds organized in a suitable structure
	Standards, format and metadata	Nosql database
	Archiving and preservation (including storage and backup)	Data should be persistently stored, and are managed in 100-day FIFO-like window. Some extracts can be open to public

Table 23 – Traffic monitoring use case – organized Road speed data

ID	ITEM	DESCRIPTION
D26	Dataset name and reference	Traffic model training sequence
	Dataset size	Gigabytes [up to Terabytes
	Dataset description	Training vectors for ML processing
	Standards, format and metadata	CSV
	Archiving and preservation (including storage and backup)	Data should be persistently stored, while being subject to updates. Some extracts can be open to public

Table 24 – Traffic monitoring use case – Traffic model training sequence

ID	ITEM	DESCRIPTION
D27	Dataset name and reference	Traffic 3D model profiles
	Dataset size	Gigabytes
	Dataset description	Speed, flow, density characterisation of roads
	Standards, format and metadata	CSV
	Archiving and preservation (including storage and backup)	Data should be persistently stored, while being subject to updates. Some extracts can be open to public

Table 25 – Traffic monitoring use case – Traffic model 3D profiles

ID	ITEM	DESCRIPTION
D28	Dataset name and reference	ML traffic model coefficients
	Dataset size	Gigabytes
	Dataset description	Coefficients of multiple ML components
	Standards, format and metadata	CSC
	Archiving and preservation (including storage and backup)	Data should be persistently stored, while being subject to updates. Some extracts can be open to public

Table 26 – Traffic monitoring use case – ML traffic model coefficients

5.5 Profiling Information Associated with the Data

In this section we analyze the data at a lower granularity (kernel or component level), reporting the information related to a single instance of each component or kernel. The following tables summarize how the data are used within the components or kernels. Tables report in details which data are used, which data are produced, and the exact data format used in the computation.

Kernel Workflow reference Dataset correspondance	Pre-processing Wind Energy Prediction D7, D8, D9, D11
Data input 1	Name: Historical power generation Format: CSV Size: 200 KB Data access type: stream in
Data input 2	Name: Historical power availability Format: CSV Size: 2 MB Data access type: stream in
Data input 3	Name: Historical TSO curtailments Format: CSV Size: 100 KB Data access type: stream in
Data output 1	Name: Pre-processing (power curve) Format: CSV Size: 1 MB Data access type: stream out
Data parallelisation aspect	Not needed

Table 27 – Renewable Energy Prediction use case - Pre-processing Kernel

Kernel Workflow reference Dataset correspondance	ML Run Wind Energy Prediction D7, D8, D9, D11
Data input 1	Name: Historical power generation Format: CSV Size: 200 KB Data access type: stream in
Data input 2	Name: Historical power availability Format: CSV Size: 2 MB Data access type: stream in
Data input 3	Name: Historical TSO curtailments Format: CSV Size: 100 KB Data access type: stream in
Data output 1	Name: Wind power forecast predicted with Machine Learning Format: CSV Size: 100 KB Data access type: stream out
Data parallelisation aspect	Not needed

Table 28 – Renewable Energy Prediction use .case - ML- Run Kernel

Kernel Workflow reference Dataset correspondance	Forecast smoothing Wind Energy Prediction D13
Data input 1	Name: Wind power forecast predicted with Machine Learning Format: CSV Size: 100 KB Data access type: stream in
Data output 1	Name: Wind power forecast smoothed with Kalman filter Format: CSV Size: 100 KB Data access type: stream out
Data parallelisation aspect	Not needed

Table 29 – Renewable Energy Prediction use case - Forecast smoothing Kernel

Component Workflow reference Dataset correspondance	WRF Meteo prediction D1, D3, D4, D5, D11
Data input 1	Name: GFS Format: Grib2 Size: 1 Gb per 48H dataset, at hourly timestep and 25 Km grid space Data access type: stream in
Data input 2	Name: Wunderground stations Format: ASCII Size: Kbytes for one year for one station Data access type: stream in
Data input 3	Name: Italy authoritative weather stations Format: ASCII Size: Kbytes for one year for one station Data access type: stream in
Data input 4	Name: Italian Civil Protection radar mosaic Format: GeoTIFF Size: 12 Mb per 24H dataset, hourly timestep, 4 CAPPI levels Data access type: stream in
Data ouput 1	Name: WRF output Format: NetCDF Size: 7Gb per 24H dataset, hourly timestep, 2.5KM grid space Data access type: stream out
Data parallelisation aspect	128 to 384 for stream in/out

Table 30 – Renewable Energy Prediction use case - WRF Componenet

Component Workflow reference Dataset correspondance	WRF Meteo prediction for Air quality D15, D16, D20a
Data input 1	Name: IFS Format: Grib2 Size: 3 Gb per 48H dataset, at hourly timestep and 9 Km grid space Data access type: stream in
Data input 2	Name: Wunderground stations Format: ASCII Size: Kbytes for one year for one station Data access type: stream in
Data ouput 1	Name: WRF output Format: NetCDF Size: 4Gb per 48H dataset, hourly timestep, 2.5KM grid space Data access type: stream out
Data parallelisation aspect	128 to 384 for stream in/out

Table 31 – Air quality use case - WRF Component

Component Workflow reference Dataset correspondance	Aggregation weather aggregation for Air quality D17, D19, D20a, D20b
Data input 1	Name: Local weather data Format: Binary Size: kbytes for one year at hourly timestep for one site Data access type: stream in
Data input 2	Name: WRF output forecast (extraction at one site) Format: Binary Size: Kbytes for 48 hours of forecast at hourly resolution for one site Data access type: stream in
Data input 3	Name: NUM output weather forecast Format: Binary Size: Kbytes for 48 hours of forecast at hourly resolution for one site Data access type: stream in
Data ouput 1	Name: Aggregated weather forecast Format: ASCII Size: Kbytes for 48 hours of forecast at hourly resolution for one site Data access type: stream out
Data parallelisation aspect	1 for stream in/out

Table 32 – Air quality use case - Aggregation Component

Component Workflow reference Dataset correspondance	ADMS Air prediction for Air quality D18, D20b, D21
Data input 1	Name: Emission data Format: Ascii Size: kbytes for one year at hourly timestep for one site Data access type: stream in
Data input 2	Name: Aggregated weather forecast Format: ASCII Size: Kbytes for 48 hours of forecast at hourly resolution for one site Data access type: stream in
Data ouput 1	Name: ADMS output Format: ASCII Size: 2Mb per 48H dataset, hourly timestep, 10km*10km grid Data access type: stream out
Data parallelisation aspect	1 for stream in/out

Table 33 – Air quality use case - ADMS Component

Kernel Workflow reference Dataset correspondance	RTE-RRTMG Meteo prediction D1, D3, D4, D5, D11
Data input 1 Data output 1	Name: Temperature, pressure, and gas concentrations profiles; surface parameters Format: NetCDF Size: ≤ 5072 Bytes per column Data access type: stream in Name: Radiative fluxes and their derivatives Format: NetCDF Size: 1200 Bytes per column Data access type: stream out
Data parallelisation aspect	in the order of 10^5 parallel columns

Table 34 – Air quality use case - RTE-RRTMG Kernel

Kernel Workflow reference Dataset correspondance	Map-matching FCD Data processing D22, D24, D25
Data input 1	Name: GPS vector Format: 20 x GPS, where GPS = float, float + int16, int16 Size: 250 bytes Data access type: stream in
Data input 2	Name: Map graph Format: List of <edge=int16,vertex=int16> Size: 100 Mbytes Data access type: static input / preloaded
Data output 1	Name: Roadspeed vector Format: 20 x Int16, Int16 Size: 100 bytes Data access type: stream out
Data parallelisation aspect	1 to 100 for stream in/out

Table 35 – Traffic monitoring use case - Map Matching Kernel

Kernel Workflow reference Dataset correspondance	Traffic prediction CNN training Model training D26, D28
Data input 1	Name: Traffic model training sequence Format: 10,000,000 x int16 + 21 x int16 Size: 1GB Data access type: random multiple access input
Data output 1	Name: ML traffic model coefficients Format: 10,000 x NNcoeff, where NNcoeff = 500 x float Size: 20 MB Data access type: output export
Data parallelisation aspect	1 to 10,000 for stream-in/out

Table 36 – Traffic monitoring use case - Traffic prediction CNN tranining Kernel

Kernel Workflow reference Dataset correspondance	Trafic Prediction Inference Prediction and Routing D28
Data input 1	Name: ML traffic model coefficients Format: 10,000 x NNcoeff, where NNcoeff = 500 x float Size: 20 MB Data access type: static input / preloaded
Data input 2	Name: Prediction request Format: 17 x fixedpoint16 Size: 40 bytes Data access type: stream in
Data output 1	Name: Prediction response Format: 4 x fixedpoint16 Size: 10 bytes Data access type: stream out
Data parallelisation aspect	1 to 10,000 for stream-in/out

Table 37 – Traffic monitoring use case - Traffic Prediction Inference Kernel

Kernel Workflow reference Dataset correspondance	Probable delay on route at departure time (PTDR) Traffic simulator, Intelligent routing The kernel is working on preprocessed data based on D22, D23, D24, D25
Data input 1	Name: List of route segments with *length*, *speed*, and *level of service probability profile* Format: List<EnrichedSegment>; **EnrichedSegment** { Segment, SegmentTimeProfile[672] }; **Segment** { double, double }; **SegmentTimeProfile** { double[4], double[4] } Size: route length $\times$ 43024 bytes Data access type: stream in
Data input 2	Name: departure time Format: long Size: 8 bytes Data access type: parameter
Data input 3	Name: number of samples Format: int Size: 4 bytes Data access type: parameter
Data output 1	Name: Probable delay on route at departure time Format: long Size: 8 bytes Data access type: return value
Data parallelisation aspect	Parallel over the number of samples.

Table 38 – Traffic monitoring use case - PTDR Kernel

6 Extracting the Data Requirements

Starting from the analysis of the data properties, from the applications' point of view, in this section we extract the constraints and the requirements that the data properties impose on the application. These, ultimately, have to be considered, managed and supported by the EVEREST environment. We will conclude this section summarizing requirements and constraints imposed by the data on the EVEREST environment in a tabular format.

In all the use cases, the data type does not impose specific constraints to the EVEREST environment, except for the obvious need of having the right tools to parse and load the data from the desired. Scripts and routines to carry out such a parsing are currently part of the data preparation phase, that is performed at the boundary of the EVEREST platform. As a result, we do not enforce any specific requirement related to the data type in this phase. In all the use cases, loading of the data happens via standard network protocols. Interfaces to these protocols should be present in EVEREST. Finally, some data are loaded using dedicated APIs. We treat routines that use dedicated APIs as parsing scripts, that have to be provided at the boundary of the EVEREST platform.

The size of the input and output data varies quite significantly, also within a single use case. Data sets can have a size of few kilobytes up to terabytes. Very likely, large data sets should be stored close to where the computation is performed. However, we leave this decision to the optimization phase, that will co-optimize computation and memory allocation. To do so, is it necessary to be able to specify, within the EVEREST environment, the need of a storage element, its size, and its relationship with the computation (are the data input or output of a specific computation?).

Some input data are globally updated, while some others feature a partial update. Furthermore, intermediate data could need to be stored in the long run, while some others can be removed immediately after the computation is completed. To optimize the management of the storage elements, in the EVEREST environment, we would need to support the specification of the type of storage required (persistent, temporal).

As for the storage, the sizing of the data communication link is imposed by the data. Data transfers in the three use cases are in the range of tens of gigabytes. Communication links between the different blocks in the EVEREST environment should thus be able to handle the transfer of such amount of data within few minutes.

Concerning the communication, we do not have specific requirements regarding the reliability of the data in addition to the mechanism already used in state-of-the-art network protocols. Security wise, we need to ensure that the data of all the applications are kept private, and we must ensure that the flow of information within the EVEREST platform is correct and does not introduce security hazards. To do so, EVEREST should provide libraries and primitives to enforce the confidentiality of the data, their integrity and their authentication (for instance using the appropriated cryptographic algorithm). To guarantee the flow of information, EVEREST should provide the adequate support for information flow tracking, embedding mechanism such as taint propagation in a transparent way.

We summarize these requirements in a tabular form in the remaining part of this section.

Id	Requirement	Description	Priority	Explanation/Possible implementation
RD1	Data Parsing	Capability of parsing the data with all the formats requires by the use cases	Could have	This operation is in the data preparation phase at boundary of the EVEREST environment: for some of the EVEREST use case, data are prepared externally and then are sent in the EVEREST environment for processing, for some others, data preparation is part of the workflow, so parser, although executed on the EVEREST environment, are not directly provided by EVEREST. When data are prepared outside the environment (or executed in the environment but as the rest of the computation) as in the use cases considered in the project, having parser capabilities provided by the environment is not required. However, embedded parsers for common data formats in the EVEREST environment could be useful.
RD2	Data loading with standard interfaces	Data are loaded using standard network protocol	Must have	The EVEREST environment should support the loading of the data using a suitable network protocols, such as https.
RD3	Data loading with APIs	Data are loaded using APIs	Must have	Certain part of the use cases require to load the data using specific APIs. The EVEREST environment should provide the support for that where needed.
RD4	Data storage availability	Data should/could be stored close to where the computation is needed	Must have	The EVEREST environment will include storage elements of various types and sizes.
RD5	Data storage specification	Type and size of storage varies, there is the need to be able to specify size, the type of storage to use, and the relation with the computation during optimization	Should have	EVEREST languages will allow to specify the size and the type of storage required and the relation between the storage and the computation.
RD6	Data storage lifetime	There is the need to be able to specify the lifetime of the storage element (temporary/permanent)	Should have	EVEREST languages will allow to specify the lifetime of the storage.
RD7	Data Transfer bandwidth	Communications requires to move up to terabytes of data. Data communication link should support it	Must have	The EVEREST platform will include high speed link among the components.
RD8	Data Reliability	There is no specific need for data reliability in addition to the network protocols	Must have	The network protocols used for communication should include support for reliability. When this is not present it should be added.
RD9	Data confidentiality	Only authorized parties can access the data	Must have	When needed, communication between the nodes in EVEREST will be encrypted using an appropriate encryption algorithm.

(To be continued)

Id	Requirement	Description	Priority	Explanation/Possible implementation
RD10	Data flow control	The flow of information does not cause security hazards	Could have	For the security of the EVEREST use cases, there is currently no need to monitor the flow of information. However, when applications belonging to several distributed computations are executed in a shared environment, having the support for information flow tracking is very important to provide security guarantees. Since flow tracking is the most relevant security feature among the ones not implemented in the project, we marked it could have to underline that this is the next security extension for the EVEREST platform.
RD11	Data integrity	Data should not be tampered during communication	Must have	Protocol and constructions ensuring integrity will be implemented in the EVEREST environment.
RD12	Data authenticity	Data are authenticated and the authentication is verified	Must have	The algorithm for encryption will be selected to provide also authentication.

7 Data management policies on Data centres

7.1 Computing Infrastructures

At this date, the EVEREST architecture can include the following components:

- Servers shared between partners such as potentially **IBM** and **IT4I** servers and partner's servers for their own usage (**NUM**, **CIMA**, **IT4I**, **SYG**).
- Roles of each servers between pre- and post-treatment, execution, visualizations, data storage (temporary, permanent, ...).
- **IT4I** operates three supercomputers: Karolina (15.7 PFlop/s, installed in the summer of 2021), Barbora (849 TFlop/s, installed in the autumn of 2019), and the specialised NVIDIA DGX-2 system for artificial intelligence computation, (130 TFlop/s and 2 PFlop/s in AI, installed in the spring of 2019). User access to **IT4I** supercomputing services is based on projects — membership in a project provides the access to the granted computing resources. Computational resources may be allocated via several allocation mechanisms. For the EVEREST project, there are relevant Open Access Grant Competitions announced 3 times a year (February, June, October) for employees of research, scientific and educational organizations[1], or EuroHPC JU grant competitions for Europe petascale systems including Karolina supercomputer (for example, continuously opened Benchmark and Development Access call)[2]. **IT4I** is also a member of LUMI consorium. EuroHPC LUMI supercomputer, implemented in Kajaani, Finland, is one of the world's fastest computing systems with a performance of over 550 PFlop/s. Also these resources can be available through open calls.
- **IBM** will offer an FPGA-based prototype architecture with both bus-attached and network-attached FPGA devices. **IBM** operates a research infrastructure for external collaborations at the IBM Research Europe – Zurich lab called Zurich Yellowzone Compute Cluster (ZYC2). ZYC2 offers of FPGA-accelerated x86- and POWER-based nodes that can be accessed directly or via private VMs controlled by an OpenStack-based cloud control. Additionally, the cloudFPGA platform is available in ZYC2 providing the power of network-attached FPGAs.

IT4I has established and continually improves an internationally recognized information security management system, manages risks, and has established processes and regulations to secure information against misuse, unauthorized changes, and loss. In December 2018, IT4I was awarded ISO 27001 certification (ISO/IEC 27001:2013, ČSN ISO/IEC 27001:2014). The certificate was obtained for provision of national supercomputing infrastructure services, solution of computationally intensive problems, performance of advanced data analysis and simulations, and processing of large data sets.

IT4I can also offer an experimental cloud infrastructure with OpenStack and VMware deployments, backed by CEPH storage. The infrastructure has 100G Ethernet network, virtualisation nodes, two additional data nodes with NVMe and NVDIMM storage and two dedicated gateway servers for HPC clusters access. An NVIDIA GPU and Intel Stratix 10 FPGA card is also available. The infrastructure is operated by ADAS Lab directly as a research infrastructure.

Data is stored on local volumes on the nodes at IBM. As ZYC2 is a research infrastructure there is no guaranteed service level, and each user is responsible to create backups of his/her data. To acquire access to ZYC2 an email must be sent to the responsible indicating a reason why access is required. After IBM internal approval, the requestor must agree to the *ZYC2 Access Agreement* and the *ZYC2 Data Usage Agreement*. After consent has been received a user account is created and documentation is sent out to describe how to connect to ZYC2 via VPN.

The data used during the project are not confidential or sensitive. In all cases, information is anonymized before entering into the project environment, as discussed in previous sections. Nevertheless, some data can

[1] https://www.it4i.cz/en/for-users/open-access-competition
[2] https://prace-ri.eu/hpc-access/eurohpc-access/

be shared only within the consortium members. These data will be communicated only by means of institutional channels, and the access will be protected by username and password (or similar access control systems). The security of some internal of data will be guaranteed using state-of-the-art open and largely supported cryptography solution, which will be defined during the project.

IBM's ongoing efforts to implement effective data management policies for our project, has successfully installed a Network File System (NFS) setup across the four nodes of our cluster. The NFS setup is backed by a RAID0 Non-Volatile Memory Express (NVMe) solid-state drive (SSD) configuration, consisting of four drives each with a storage capacity of 512GB. The NFS share has been mounted at the $/scratch$ location of the cluster and is accessible to all EVEREST users with both read and write permissions. The configuration of the NFS share has been optimized to allow access by SLURM jobs, thereby providing seamless integration with our cluster computing environment. This NFS setup represents a significant milestone in our efforts to improve data management and storage within our data centers, and we are confident that it will greatly enhance the efficiency and reliability of our data storage and retrieval processes. We will continue to monitor and maintain the system to ensure it operates optimally, meeting the demands of our growing data storage needs.

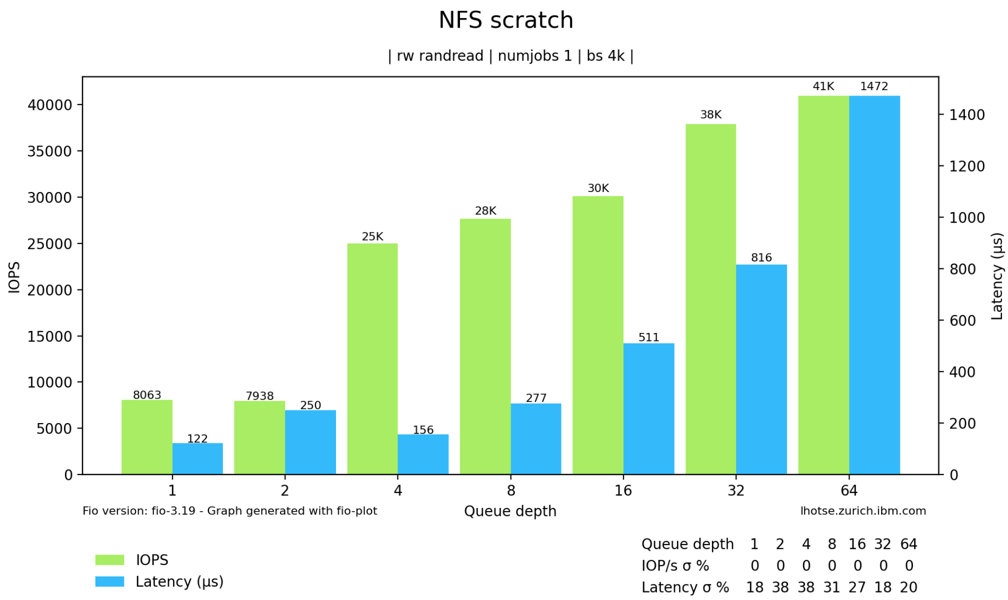

Figure 5 – Benchmarking of NFS share on EVEREST HPC cluster: Rand read with 1-job.

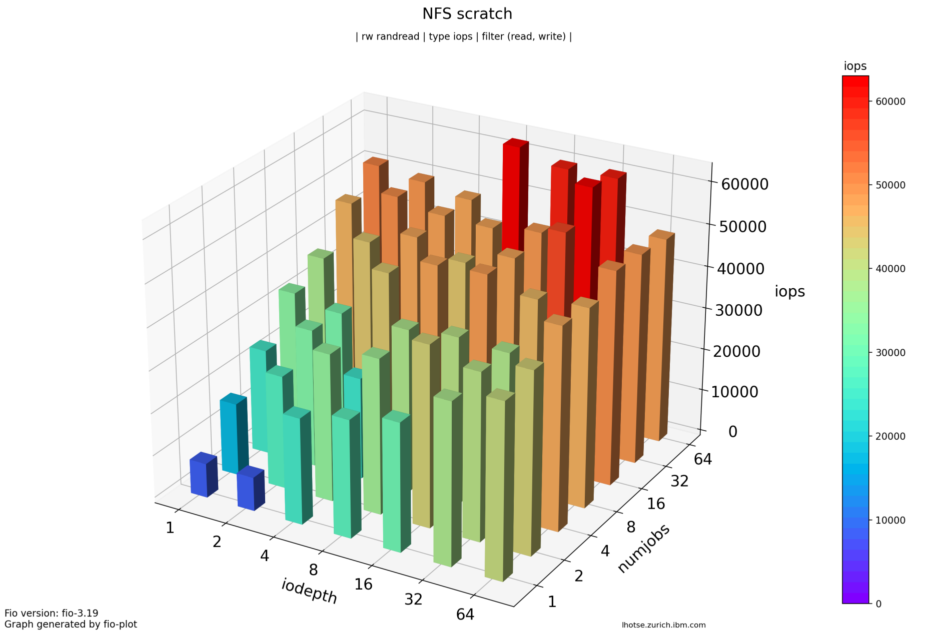

Figure 6 – Benchmarking of NFS share on EVEREST HPC cluster: Rand read with [1-64]-jobs and [1-64]-threads.

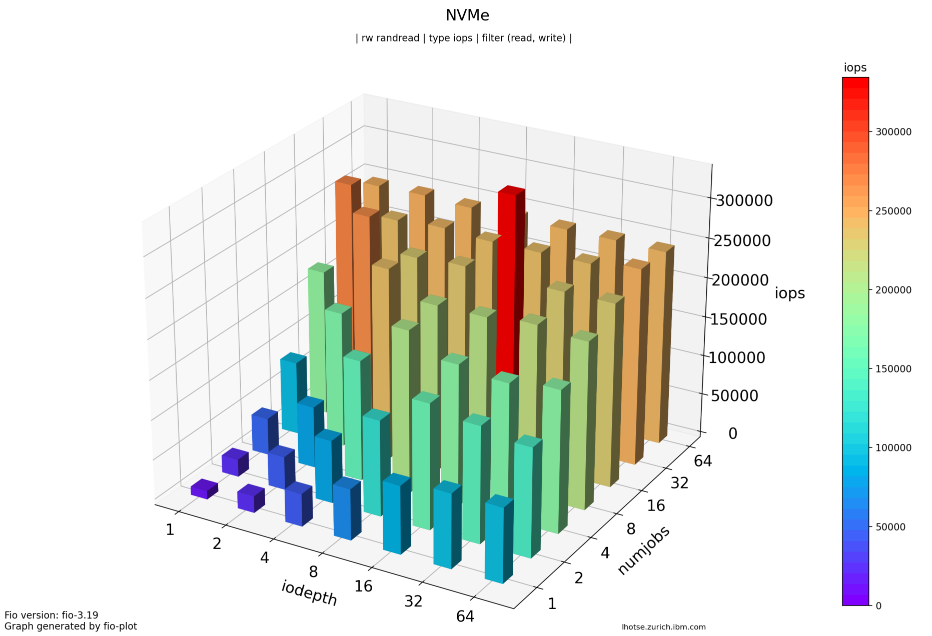

Figure 7 – Benchmarking of a standalone NVMe disk of the NFS share of EVEREST HPC cluster: Rand read with [1-64]-jobs and [1-64]-threads.

Additionally, this NFS setup serves as a prototype for our data management policies and has undergone benchmarking using the FIO tool. The results of these benchmarks are depicted in Figures 5,6,7, and demonstrate the performance capabilities of the system. Specifically, Figure 5 shows the IOPs and latency results using only one rand-read job on fio tool. Figures 6 and Figure 7 show an extensive benchmarking of the storage medium, using 1-64 jobs and 1-64 I/O depth, by either accessing the disk through the NFS share from a remote node or by accessing it as a standalone device (i.e. local mount) respectively. Our plan moving forward is to assess if the performance of this prototype meets the criteria required for our high-performance computing (HPC) workflows within the project. Based on these results, we will make informed decisions regarding the extension of this data management setup to our Alveo cluster. This NFS setup, combined with our ongoing efforts to optimize data management and storage, will help ensure that our data centers remain efficient and effective in meeting the evolving needs of our project.

8 Conclusions

This deliverable reports the final version of the data requirements and it is intended to be self-contained. Because of this, we started from the text of Deliverable 2.3, that was updated were needed. We started summarizing the three use cases of the project, namely the renewable energy production, the air quality monitoring, and the traffic modeling and with a short description of the kernels and components of each of the EVEREST use case. Then we presented the process followed during the data requirement phase that, starting from the collection of the characteristics of the data used in each use case, allowed us to infer the additional requirements on the EVEREST platform imposed by the data. The analysis is now carried out also at the granularity of the use case, but also at the granularity of kernel or at the granularity of components. This allowed us to crystallize the size of the input/output of the computational modules considered in the EVEREST used case and to identify, where applicable, the level of parallelisation allowed by the data (which ranges from none to a complete parallelization over the number of samples.)

From the analysis at the use case granularity, we have divided the data properties in eight axes: type of data, size of data, data transfer within the application, loading data in the system, location of the data and of computation, timing requirements on data and computation, reliability properties associated with data, and security properties associated with data. We have then classified the data requirements based on their priority and we identified that several requirements are classified as "must have" (these are the most important requirements), while few others are classified as “should have”. From our analysis it emerged that despite the heterogeneity of the application, data types do not impose specific constraints on the application. The only exceptions are the obvious parsing capabilities and the support for APIs loading the respective data types, which are indeed classified as “must have”. The size of the data within the individual use cases varies between a fews kilobytes and several terabytes. Furthermore, there could be the need to store large amounts of intermediate data. The EVEREST SDK must provide the possibility to specify the characteristics of a specific storage element, including its size, and the location where the computation should be carried out. This information is needed to optimize the storage allocation.

The size of the data transfers is in the range of tens of gigabytes, thus the communication links internal to the EVEREST environment should support such a size. Requirements on the storage size and bandwidth for transfer are classified as “must have” since they impose the size and type of storage elements.

Related to security and reliability, the “must have” requirements include mechanisms to ensure that the data used throughout the workflows is correct, authenticated, and that its integrity and availability is preserved as well as sufficient to address the need of the application. In addition to the security functionalities imposed by the use case and already included in the EVEREST environment, for future applications of the EVEREST environment could be also useful to ensure that the data flow within the EVEREST platform does not introduce new security hazards. Because of this, R10 has been classified with “could have” priority