SPECTRUM

D5.1 Representative use cases: analysis and alignment

Status: Approved by the EC

Dissemination Level: Public

Funded by the European Union

Disclaimer: Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. SPECTRUM is funded by the European Union - Grant Agreement Number 101131550 - www.spectrumproject.eu

Abstract	
Key Words	Artificial Intelligence, Batch, Big Data, CERN, Data, Data-Driven, e-infrastructures, EuroHPC, Exascale, Federation, High Energy Physics, HPC, HTC, LOFAR, Radio Astronomy, Representative Use Cases, Research Computing, SKA, SKAO

This document describes the collection and analysis of requirements from representative use cases for next-generation big data sciences. The work presented here was executed by Work Package 5 of SPECTRUM, and details prominent use cases from fundamental research in areas of High Energy Physics and Radio Astronomy, as well as other related fields. Cross-cutting analysis of requirements is presented along with gap analysis.

Document Description			
D5.1 Representative use cases: analysis and alignment			
Work Package Number 5			
Document Type	Report		
Document status	Approved by the EC	**Version**	1.0
Dissemination Level	Public		
Copyright status	 This material by Contributing Parties of the SPECTRUM Consortium is licensed under a Creative Commons Attribution 4.0 International License.		
Lead Partner	CERN		
Document link	https://documents.egi.eu/document/4071		
Digital Object Identifier	https://zenodo.org/records/15310223		
Author(s)	• David Southwick (CERN) • Eric Wulff (CERN) • John Swinbank (ASTRON)		
Reviewers	• Jeff Wagg (CNRS) • Andrea Manzi (EGI)		
Moderated by	• Patricia Ruiz (EGI)		
Approved by	• Xavier Salazar (EGI) - on behalf of AMB		

Revision History			
Version	**Date**	**Description**	**Contributors**
V0.1	21.11.2024	First draft	Eric Wulff (CERN)
V0.2	06.12.2024	Wrote the introduction and methodology sections. Revised the outline of the deliverable. Wrote a first draft of the conclusion section. Added intro to the cross-cutting requirements section and a comprehensive table summarizing main technical needs for each use case.	Eric Wulff (CERN)
v0.3	07.03.2025	Internal Review	David Southwick (CERN)
v0.4	21.03.2025	External Review	Jeff Wagg (OCA) Andrea Manzi (EGI)
V0.5	29.04.2025	Incorporated feedback	David Southwick (CERN)
V0.6	30.04.2025	AMB Approval	Xavie Salazar (EGI)
V1.0	30.04.2025	**Final**	

Terminology / Acronyms	
Terminology / Acronym	**Definition**
AAI	Authentication and Authorisation Infrastructure
AI	Artificial Intelligence
ALICE	*A Large Ion Collider Experiment*, a specialized detector on the LHC at CERN
ATLAS	*A Toroidal LHC Apparatus,* a general purpose detector on the LHC at CERN
CERN	European Organization for Nuclear Research
CMS	*Compact Muon Solenoid,* a general purpose experiment on the LHC at CERN
CVMFS	CERN Virtual Machine File System
EAB	External Advisory Board
GDPR	General Data Protection Regulation
GID	Group ID
HEP	High Energy Physics
HTC	High Throughput Computing
HPC	High Performance Computing
LHC	Large Hadron Collider, a circular particle accelerator at CERN
LHCb	*Large Hadron Collider beauty,* a specialized detector on the LHC at CERN
LIGATE	*Ligand Generator and portable drug discovery platform AT Exascale*, a EuroHPC project
LLM	Large Language Model
LOFAR	the LOw Frequency ARray
MD	Molecular Dynamics
MISTRAL	Meteo Italian Supercomputing portal
NRENs	National Research and Education Network Organizations
RA	Radio Astronomy
SKAO	Square Kilometer Array Observatory
SLURM	*Simple Linux Utility for Resource Management*, a HPC job scheduler
SRCNet	*SKA Regional Center Network*

SRIDA	Strategic Research, Innovation and Deployment Agenda
SSO	Single Sign On
WLCG	Worldwide LHC Computing Grid

Table of Contents

List of Figures

List of Tables

Executive summary

This deliverable presents the analysis of requirements for representative next-generation use cases in big data sciences, as well as presents a cross-cutting analysis of common themes and trends. Included is a description of the methodology for identifying, selecting, and analysing the use cases, as well as a short reflection on applications to other fields of research.

The use cases are primarily selected from the fields of High Energy Physics (HEP) and Radio Astronomy (RA), which in the coming years will launch new instruments around the same time that will produce in excess of one Exabyte of data per year each, several orders of magnitude larger than current compute capabilities. Several other complementary use cases are presented that highlight related challenges.

The main findings in this deliverable are reported in a cross-cutting analysis highlighting common requirements and challenges, developed from the technical analysis of individual use case requirements. The key findings identify a significant need to greatly expand the compute capabilities of the fields to process unprecedented volumes of data. It is clear this must include heterogeneous compute models with GPUs and specialized accelerators, an area in which all use cases are investing heavily. Deployment of advanced workflow management systems capable of orchestrating large multi-step workflows across distributed resources is critical to Exascale computing. The geographical distribution of the research collaborations and the sheer scale of data involved necessitate robust data federation mechanisms. These must support efficient data discovery, access, and transfer across multiple sites and infrastructures. Finally, significant technical challenges remain, particularly in I/O performance, and energy efficiency.

This deliverable was executed by the SPECTRUM Work Package 5 on Landscape, use cases, challenges and gaps, along with significant contributions from subject matter experts from the selected use cases.

This deliverable will serve as one of the inputs to the second phase of SPECTRUM for the production of a Strategic Research, Innovation, and Deployment Agenda (SRIDA) and a Technical Blueprint for Compute and Data Continuum.

1. Introduction

The SPECTRUM project aims to address the unprecedented data processing, simulation, and analysis needs of frontier research communities, particularly in High Energy Physics (HEP) and Radio Astronomy (RA). As these fields prepare to launch new instruments that will generate data at an unprecedented scale, entering the Exascale era, it becomes crucial to design and develop innovative compute solutions for data-intensive architectures, resource federation models, and IT frameworks. This technical report, titled "Representative use cases: analysis and alignment," focuses on a critical aspect of the SPECTRUM project's approach to meeting these challenges.

The primary objective of this report is to present a comprehensive analysis of representative science use cases with impact on the compute and data continuum, with particular relevance from the HEP and RA communities. By examining these use cases in detail, we aim to:

1. Identify current and future requirements for compute resources, data storage/processing capabilities, and essential services.
2. Uncover cross-cutting needs that span multiple scientific domains.
3. Reflect on how these findings align with the known needs of other scientific communities.

Our approach to this analysis consists of three main steps.

1. Selection of carefully chosen representative use cases from the HEP and RA communities that exemplify the data-intensive challenges faced by these fields. These use cases serve as a foundation for our analysis and provide concrete examples of the scale and complexity of future research needs.
2. Detailed analysis where each selected use case undergoes a thorough examination to determine its specific requirements in terms of computational resources, data storage and processing capabilities, and supporting services and infrastructure. This step includes deep technical discussions with use case experts.
3. Performing cross-cutting analysis by comparing the individual use case analyses, we identify common themes and shared requirements across different scientific applications. This step is crucial for developing solutions that can benefit multiple research domains and maximize the impact of future investments in research infrastructure.

The findings presented in this report will contribute significantly to the SPECTRUM project's overarching goal of delivering a Strategic Research, Innovation and Deployment Agenda (SRIDA) and a Technical Blueprint for a European compute and data continuum. By grounding our recommendations in real-world scientific use cases, we ensure that the proposed solutions are directly aligned with the needs of the research community.

Furthermore, this analysis will play a vital role in:

- Informing the design of future Exabyte-scale research data federations and compute continua.
- Guiding the integration of emerging technologies such as Exascale HPC, AI, and Quantum computing systems into existing research infrastructures.
- Addressing key challenges related to scalability, performance, energy efficiency, portability, interoperability, and cybersecurity.

As we proceed with this analysis, we remain committed to the vision of enabling data-intensive scientific collaborations across Europe, supported by a robust and flexible research infrastructure capable of meeting the demands of frontier science in the coming decades.

2. Methodology

The methodology for this report is designed to systematically analyze representative use cases from the High Energy Physics (HEP) and Radio Astronomy (RA) communities, identify their specific requirements, and derive cross-cutting needs that can inform the development of a federated European compute and data continuum. The following steps outline the approach taken.

The first step consisted of selecting representative use cases. To ensure relevance and coverage, representative use cases were selected based on their ability to highlight key scientific and technical challenges faced by the HEP and RA communities. The selection criteria included:

- Scientific significance and alignment with the SPECTRUM project's objectives.
- Data-intensive nature of the use case, particularly in terms of compute, storage, and data transfer requirements.
- Representation of both current challenges and future visions for Exascale-era research.

Input from domain experts within the SPECTRUM consortium and the broader Community of Practice was solicited to identify use cases that exemplify diverse workflows, data lifecycles, and computational needs. In addition the SPECTRUM External Advisory Board (EAB) was consulted and the final list of use cases to be analysed was shared with them.

The second step was to create a structured framework for analyzing the use cases. Each selected use case was analyzed using this framework to address key aspects critical to understanding its requirements. The framework is based on the template produced by SPECTRUM T5.1 and includes the following sections:

- Scientific Challenge: Overview of the scientific objectives, community served, technical challenges, timescale, and open science principles.
- Storage: Analysis of data volumes, lifecycles, safety measures, confidentiality needs, and storage patterns.
- Data Transport: Examination of data throughput, geographical transfer requirements, streaming needs, and transfer technologies.
- Compute: Description of compute platforms, unit characteristics (e.g., runtime, operations), scaling models, and software dependencies.
- Workflow Management: Exploration of multi-step workflows, orchestration tools, time-critical steps, and execution environments.
- Access and Analysis: Assessment of interaction modes with resources (e.g., interactive vs. batch), in-situ vs. remote analysis needs, and required capabilities for analysis.
- Non-technical Challenges: Consideration of access mechanisms, authorization systems, training/support needs, and regulatory compliance.
- Gap Analysis: Identification of missing services or capabilities that hinder productivity or scalability.

A copy of the template is provided in **Annex 1: Use Case Documentation Template**.

Step three concerned the identification of cross-cutting needs. After analyzing individual use cases, a comparative analysis was conducted to identify common themes and shared requirements across the HEP and RA communities. This step involved aggregating findings from all use cases to pinpoint recurring challenges in various areas. It also included highlighting opportunities for shared solutions that could address multiple scientific domains.

Throughout the process outlined above, feedback was sought from domain experts within the SPECTRUM consortium as well as external stakeholders in the Community of Practice and from use case experts. This iterative engagement helped refine the analysis framework and validate findings.

The results from this methodology are documented in this report to directly support the development of the Strategic Research, Innovation and Deployment Agenda (SRIDA) and Technical Blueprint for a European compute and data continuum. The structured approach ensures that findings are actionable and aligned with SPECTRUM's overarching goals.

3. Representative use cases

3.1. Representative use cases from High Energy Physics

The selected use cases for High Energy Physics (HEP) represent experimental physics via the four major experiments operating in the Large Hadron Collider (LHC) beamline (ATLAS, ALICE, CMS, LHCb), theoretical physics (QCD), and two modern AI methodologies (MLPF, LLM). The use cases are grouped thematically and by their computing models.

LHC Experiments have traditionally relied on batch CPU processing provided by the Worldwide LHC Computing Grid (WLCG) to compute and process the experimental data, simulated data, and analysis workloads required for operation. From 2030, these experiments together will generate more than 1 Exabyte of data (experimental+simulated) per year, projected to far exceed the computation capabilities of traditional grid computing. While all avenues to reduce this gap are being pursued, including development of AI methodologies later described, integration of HPC sites is being actively pursued by all experiments. Owing to the common computing model, the technical requirements for automated execution of the full pipeline of workloads from HEP experiments shares three common features necessary for deployment on HPC. The first is deployment of "edge" service(s) for job orchestration, monitoring, data orchestration, and integration with federated batch queues. There are several flavors of these services developed by the use cases, and all have been extended to support workload management using SLURM[1]. Supporting this, the second feature is sufficient external connectivity and infrastructure support for large-scale federated data transfers. The workload management service interacts with this file transfer endpoint to coordinate pre-placement of datasets when available, or to stream files as required by jobs. Finally, the jobs expect a common compute environment with a minimum compute capability and mounted network file system(s), such as the CERN Virtual Machine File System (CVMFS)[2].

Subsets of a particular experiment's full workload pipeline may have vastly reduced requirements, or may satisfy elements via containerization or other access methods.

Theoretical modeling and simulation with Quantum Chromodynamics (QCD, Lattice-QCD) has always been a major global consumer of compute resources, and will continue to drive innovation and performance of HPC centers worldwide. New AI methodologies will complement and in some cases replace workflows across all domains of HEP, driven by disruptive changes in computing architecture. Currently, there are no mature AI workloads in production for HEP, thus the majority of computing needs in the coming years will be in the form of training campaigns, but this will mature into production inference workloads across a variety of heterogeneous hardware.

3.1.1. CMS

The CMS community comprises over 4,000 physicists, engineers, and computer scientists from 240 institutes across more than 50 countries. Data generated by the CMS detector is distributed through the WLCG and analyzed by institutions worldwide. To manage the increasing data rates and ensure efficient analysis, CMS requires scalable and federated access to HPC sites, enhanced submission infrastructure, and standardized transfer tools.

The CERN CMS Production 2029 use case outlines the projected computational and storage needs for the Compact Muon Solenoid (CMS) experiment during the High Luminosity Large Hadron Collider (HL-LHC) era, beginning in 2029. As one of CERN's primary particle detectors, CMS relies heavily on computing resources to process, analyze, and store vast amounts of data generated from high-energy collisions. With increasing beam collision frequency and intensity yielding exponential data complexity, this use case highlights the growing computational demands and the pivotal role of HPC in supporting the experiment's goals.

CMS currently uses HPC resources for workflows such as event generation, simulation, and reconstruction, contributing 5-10% of total computing power. Looking ahead to Phase-II of the HL-LHC, computing needs are expected to surge, driven by the superlinear growth in reconstruction time per event due to increased

[1] https://slurm.schedmd.com/documentation.html
[2] https://cernvm.cern.ch/fs

collisions per bunch crossing (Pile Up). In 2023, pileup levels were at 52, but this will increase to 140 in Run 4 and 200 in Run 5, amplifying the complexity of particle tracking algorithms and necessitating significant computational upgrades.

CMS storage needs encompass software, scratch space, and input/output data. Calibration data and software are typically accessed through CERN's CVMFS. The scratch space for running jobs is minimal (less than 2 GB/core), while input/output storage varies based on workload. The event reconstruction (RECO) is particularly demanding, with RAW input datasets reaching petabyte scales and derived output datasets in the gigabyte range.

Projected data volumes for 2029 include:

- RAW data: 34 billion events, totaling 146 PB.
- Monte Carlo (MC) simulations: 85 billion events, requiring 6.12 PB.

Data is retained for weeks to months until processing is complete, after which it is transferred offsite. Long-term retention is not required at HPC sites, as source data is archived within the WLCG.

Data throughput is expected to double by 2029, reaching 5 MB/s per core. Large LHC dataset transfers occur via dedicated private networks between WLCG Tier0 and Tier1 sites, and the remaining Tier2 sites via network overlays to ensure low latency and high bandwidth on the national Research and Education networks[3]. Tools such as Rucio[4] orchestrate data transfers across sites, leveraging protocols like XRootD[5] for on-demand job data streaming.

CMS workloads operate through batch submission and parallel processing. Compute tasks include collision event processing, simulation, and reconstruction, with a focus on CPU-heavy operations. AI and GPU-based methods are increasingly employed to optimize resource use. By 2029, reconstruction workflows will handle billions of events annually, scaling linearly with pileup and event rates.

A typical reconstruction job processes 4.3 MB per event, with memory usage of 2 GB per core. Multi-threaded jobs dominate, with thousands of simultaneous jobs running 24/7. Scaling compute resources in line with pileup levels and leveraging AI-based tracking solutions are critical to meeting future demands.

CMS workflows rely on HTCondor[6] for federated job scheduling, with pilots dynamically pulling jobs from a global pool. Automated workflows manage up to 500,000 CPUs across WLCG and HPC sites. Output data is transferred to external collection points post-processing.

User interaction is limited to initial setup and development, with most tasks automated. Interactive analysis is conducted through Jupyter notebooks and web interfaces, with data analyzed in situ or transferred to remote sites for further study.

Key challenges include:

1. HPC Integration: Improved federated access to HPC resources for long-term allocations.
2. Storage Solutions: Aligning HPC storage with CMS data management software to create federated storage endpoints capable of XRootD-based transfers to streamline data movement.
3. Authentication: Addressing security policies that hinder automation by expanding acceptable authentication methods.

By addressing these challenges, CMS aims to harness HPC resources more effectively, ensuring the success of its physics program during the HL-LHC era and beyond.

[3] https://connect.geant.org/2023/12/20/lhcone-a-model-of-network-evolution-in-the-21st-century
[4] https://rucio.cern.ch
[5] https://xrootd.org
[6] https://htcondor.org

3.1.2. ATLAS

The ATLAS (A Toroidal LHC Apparatus) Experiment use case represents the computational and storage requirements for the operation of the experiment during the High-Luminosity period of the LHC, from 2029-2041. The ATLAS Experiment is a collaboration of over 6,000 physicists and engineers that operates on the same beamline and distributed computing infrastructure, and with a similar purpose as the CMS use case described in section 3.1.1. The ATLAS experiment has developed similar computing pipeline steps, and thus shares many requirements and challenges. The increased LHC operating parameters during the high-luminosity runs of the LHC means ATLAS will generate data that is significantly more complex, and at higher frequencies, resulting in 7-10 times higher compute requirement than current levels.

ATLAS is projected to collect around 400 PB/yr of data starting from 2029 consisting of both detector and simulation datasets. Physics events are typically batched into 1-10GB files of ROOT[7] or HDF5[8] format, and mostly read sequentially. Parallel jobs may read the same datasets, but output unique files. Tiered storage is not built into the data model, but is used for specific high I/O or high iops jobs. Input and output datasets have individual group IDs for tracking and transfer systems. Datasets are foreseen to be stored across federated storage systems at CERN and participating WLCG sites. ATLAS has developed and deployed various grid connector edge services for HPC such as the Advanced Resource Connector Computing Element (ARC CE)[9] which manages dataset transfers to and from HPC sites corresponding to job needs.

Typical job requirements are 8 cores, 1GB RAM/core, 20GB scratch, 10GB/s link, no latency requirements, with an average of 12H slots (adjustable 3-24H based on number of events/job). Datasets are normally transferred at the beginning and end of job slots, so the average over many asynchronous jobs is not significantly higher than for a single job. In 2024, ATLAS consumed >600k cores continuously, this annual need is expected to increase to several million in 2030. The software binaries target x86, aarch64, SSE, AVX512, with upcoming support for NVIDIA, TPUs and FPGAs. The ATLAS software stack is around 5.5M lines of code mostly written in C++ and is generally accessed pre-compiled via mounted CVMFS volumes during runtime. Conditions (workload configuration tunables) are delivered from remote OracleDB via edge services at HPC sites, or can be exported as SQL files.

Workload management is centrally automated using the Production and Distributed Analysis (PanDA[10]) system which requires an edge service (Harvester) at HPC centers to submit, monitor, and collect jobs. Conditions data needed by jobs is likewise delivered via FroNTier[11] edge service.

3.1.3. LHCb

The LHCb (LHC beauty) Monte Carlo Simulation use case concerns a subset of the complete compute pipeline for the LHCb detector named Monte Carlo Simulation. This subset consists of the creation of simulated particles (event generation), detector simulation using the generated events in digital models of the detector, and finally event reconstruction and selection of the simulated collisions. The LHCb collaboration has chosen this subset for HPC due to the simplified requirements for running (no input datasets), and due to the fact that this subset represents 90% of the total projected CPU use until the end of LHC run 5 in 2041.

During run 4 (2030-2034) the LHCb collaboration expects to produce several PB of simulation data, and during run 5 (2036-2041) this will grow by approximately 7.5X. As with the other LHC experiments, this is foreseen to be stored in federated WLCG storage clusters in 10-100GB files in ROOT format. As the Monte Carlo simulation use case does not require any input datasets, the only input for HPC jobs is access to the simulation software, either via CVMFS or Apptainer[12], and some relational databases for metadata. Jobs are batched within one site and require <10MB/s local connectivity and <1MB remote connectivity (DB, conditions data). Job outputs are bulk transferred to WLCG grid storage sites via fast data transfer tooling.

[7] https://root.cern
[8] https://www.hdfgroup.org/solutions/hdf5
[9] https://www.nordugrid.org/arc/ce
[10] https://panda-wms.readthedocs.io
[11] https://frontier.cern.ch
[12] https://apptainer.org

Simulation data production is expected to scale linearly, with a 7.5X increase starting at the beginning of run 5 due to detector upgrades.

MC-sim jobs are single node batch submission compiled for X86, with a small subset of applications available for aarch64. The collaboration is actively exploiting GPUs and other accelerators where possible and expects this to be adopted in the future. Of the described components of the use case, the dominant consumer of CPU work is detector simulation. Typical jobs are scheduled <24H and require 2GB memory/core, <20GB local scratch space per core. Jobs are scheduled from a central federated management system (DIRAC WMS[13]) and for automated HPC execution this entails an edge service to coordinate scheduling, monitoring, and transfer of job output.

3.1.4. ALICE

ALICE (A Large Ion Collider Experiment) use case represents the current detector HPC use with projections for future needs during the High Luminosity era of the LHC. ALICE is one of the four major experiments operated at the CERN LHC and its detector is targeted towards the study of heavy-ion physics and the resulting Quark Gluon Plasma[14].

The total ALICE storage current content is 14 billion files of various sizes from few KB to 10GB with a growth rate of 10% per year. The read to write data ratio is approximately 8/1, with random access from disk storage and all data is written and read using the XrootD protocol and tools. The current total available CPU resources for ALICE are approximately 250k cores, with about 10% provided by HPC. This scales linearly with computational resources and is expected to grow about 15% annually through 2041. The main computational workflows are as follows:

- Monte Carlo simulation - CPU-intensive payload with minimal (up to few MB) input data per unit workflow and medium (up to 5GB) output data. The payload duration is of the order of a few hours and runs on 8 CPU cores. The fraction of the MC simulation is ~40% of the total distributed computing resources of ALICE and runs on any Grid resources, including HPCs.
- RAW data reconstruction - CPU and I/O intensive payload with large (order of 100GB) of input data per unit workflow and medium (up to 10GB) output data. The payload duration is of the order of a few hours and runs on a various number of CPU cores (8-64), combined with accelerators (GPU). The fraction of the RAW data processing is ~30% of the total distributed computing resources of ALICE and it targets specific computing centres (T0 and T1s). One of the HPCs accessible to ALICE (Nurion@KISTI T1) is used for this workflow.
- Organized analysis - I/O intensive payload with large (up to 100GB) input data per unit workflow and small (up to 1GB) output data. The payload duration depends strongly on the input data size and varies between 20 min to several hours and typically runs on 2 CPU cores. The fraction of the Organized analysis is ~30% of the total distributed computing resources of ALICE and runs on any Grid resources, including HPCs.

ALICE software is compiled for x86, ARM, and GPUs and executed as OCI-standard containers. Access to HPC compute resources typically occurs through a point-of-presence host at the site, the VO-box. This host runs ALICE-specific persistent services, including a monitoring module and a JobAgent submission module to the CE gateway or batch system.

The ALICE Grid middleware, JAliEn[15], employs late-binding techniques for all payload types through pilot jobs. Data locality, meaning payloads execute where the data resides, is a crucial matching parameter for sites and data-intensive workloads such as Raw reconstruction and Organized analysis. While payloads primarily access data locally, they are permitted to reach over the wide area network (WAN) to read configuration files and conditions data, which are stored in a limited number of geographically distributed locations to optimize access times and network load. Additionally, payloads can access data over the WAN from remote storage containing secondary copies in cases where locally stored data is temporarily inaccessible due to missing files or storage overload. The total WAN traffic resulting from these exceptions is approximately 5% of the overall data traffic generated by ALICE payloads.

[13] https://dirac.readthedocs.io
[14] https://home.cern/science/experiments/alice
[15] https://jalien.docs.cern.ch

Beyond payload access, ALICE utilizes JAliEn's built-in tools to execute data transfers between storage elements. These tools also rely on XrootD, specifically the xrd3cp third-party copy implementation, which facilitates server-to-server data transfer.

ALICE disk storage elements exclusively utilize three storage technologies: EOS[16], XrootD[17], and dCache[18]. The software delivery to all computing centers, including HPC facilities, is exclusively handled through CVMFS. CVMFS is fully integrated into the three HPC facilities accessible to the ALICE experiment.

Since payloads directly access data from storage in a point-to-point manner, all ALICE storage elements require two ports (1094/TCP, 1095/TCP) to be open to the world for incoming access, as dictated by the XrootD protocol and the XrootD-enabled storage solutions. Similarly, all computing center WNs, including HPCs must have these ports open for outgoing access. This constitutes a strong requirement for site infrastructure, including HPC facilities. Given that the majority of Worldwide LHC Computing Grid (WLCG) sites are interconnected through the LHCONE overlay network, this traffic is deemed secure and does not necessitate passing through site firewalls.

3.1.5. AI-based particle flow reconstruction for LHC particle detectors

The AI-based Particle Flow (PF) reconstruction use case for Large Hadron Collider (LHC) particle detectors focuses on enhancing data reconstruction for HEP experiments using ML. This initiative builds upon the established PF algorithm introduced in CMS Run 1, aiming to improve particle reconstruction by integrating information from multiple sub-detectors, thereby increasing the resolution of jets and missing transverse energy measurements.

At the core of this use case is the Machine Learning-based Particle Flow (MLPF) approach, which reformulates PF reconstruction as a supervised learning problem. By leveraging ML models, MLPF can match or exceed the performance of heuristic algorithms, offering scalability, efficiency, and adaptability to new detector configurations. This adaptability is critical for current and future detectors, such as those envisioned for the Future Circular Collider (FCC). The MLPF approach is particularly attractive due to its ability to retrain for varying detector conditions and its suitability for parallelization on modern heterogeneous computing architectures, including GPUs and neural network accelerators.

The storage demands for this project are driven by large-scale simulations and data processing. Simulated datasets can reach tens of terabytes, with processed machine-learning-ready datasets occupying up to a terabyte. Each ML training session typically involves between 100 and 1000 GB of data, depending on the complexity of the events and detector configurations. The datasets are stored long-term (5+ years) for reproducibility, with open access for specific datasets like those from CLIC[19], while CMS data remains restricted until formally released.

Data transport involves transferring TB-scale datasets from Tier-2 computing sites to global HPC facilities for training and processing. These transfers are facilitated by protocols like FTS[20] and Globus[21], with storage at CERN's EOS infrastructure. In the future, distributed data production and caching at HPC sites during training are anticipated to streamline workflows.

Compute requirements are substantial, with ML training typically distributed across multiple GPUs or AI accelerators. A single training session may require 4 to 8 NVIDIA A100 GPUs, and distributed training can span up to 24 compute nodes, consuming thousands of GPU hours. The project also heavily relies on hyperparameter optimization (HPO), further amplifying computational demands. Full-scale HPO runs, distributed across 96 GPUs, can consume around 12,000 GPU hours. Annual compute estimates are projected between 50,000 and 100,000 GPU hours, with growth expected as model complexity increases.

[16] https://eos-web.web.cern.ch/eos-web
[17] https://xrootd.org
[18] https://www.dcache.org
[19] https://clicdp.web.cern.ch
[20] https://fts.web.cern.ch/fts
[21] https://www.globus.org/data-transfer

The project's workflows span multiple steps, including physics simulations, data preprocessing, model training, and event reconstruction. These workflows are managed through site-specific configurations and batch systems like SLURM, with interactive analysis facilitated by Jupyter notebooks. Data analysis typically occurs in situ at HPC sites, but smaller datasets (1–100 GB) may be transferred for local analysis.

Access to computational resources is currently distributed among collaborators at various institutions, with researchers utilizing local clusters or EuroHPC facilities like LUMI-G[22]. The project team consists of 1 to 10 researchers, with GPU access being a key bottleneck. Expanding GPU availability and access to AI accelerators would significantly enhance productivity, enabling larger datasets, more complex models, and deeper hyperparameter optimization.

Overall, the AI-based PF use case exemplifies the transformative potential of ML in HEP, with far-reaching implications for detector development and future collider experiments. Addressing gaps in GPU resources and unifying workflow environments will be crucial for sustaining and expanding this innovative approach.

3.1.6. LLMs for the CERN accelerator complex

The "LLMs for the CERN Accelerator Complex" use case outlines the deployment of Large Language Models (LLMs) to enhance knowledge retrieval, user support, and operational efficiency at CERN. This initiative, led by CERN's Beams and IT departments, focuses on developing AccGPT, a chatbot leveraging LLMs to interact with CERN's extensive internal knowledge base. The project aims to streamline technical support, improve productivity, and assist new users in navigating CERN's complex tools and frameworks.

AccGPT is designed to serve the CERN community by providing instant responses to technical and operational queries, reducing the burden on experts and accelerating routine workflows. The long-term vision includes expanding the chatbot's capabilities to provide coding assistance and even integrating LLMs into CERN's control rooms to facilitate real-time interaction between operators and machinery.

Key scientific and technical challenges include ensuring the accuracy and contextual relevance of LLM-generated responses, safeguarding sensitive internal data, and scaling the system to handle diverse queries. AccGPT is currently in active development, with small-scale deployments underway. Future iterations will extend functionality and expand the model's reach across CERN's various operational domains.

From a storage perspective, the current dataset supporting AccGPT is under 10 GB, primarily consisting of structured text. While this is expected to grow, the increase will be manageable in the near term. However, future multi-modal models incorporating images, plots, or audio could drive storage requirements significantly higher. Retaining multiple versions of datasets and storing LLM model checkpoints may lead to terabytes of storage needs. Confidentiality measures are paramount, with secure handling of sensitive CERN data being a critical priority.

Data transport requirements are modest, with most data residing within CERN's internal infrastructure. The primary need involves efficient, low-latency transfers within CERN sites. However, if inference were to leverage external HPC resources, mechanisms to securely transmit user queries to HPC sites and return results in real-time would be necessary. Latency-sensitive operations, such as those in control rooms, may require on-site deployment to ensure rapid response times.

Compute resources are central to this use case, particularly for inference and deployment of LLMs ranging from 8 billion to 405 billion parameters. GPUs, specifically NVIDIA A100 and H100 models, are essential to handle inference workloads. Smaller models can run on a single A100, while larger models may require multi-node deployments using clusters of H100 GPUs. Training, while not currently conducted due to resource constraints, could benefit from fine-tuning existing models with additional compute capacity.

Inference workflows are interactive, responding to user queries through a web-based interface. Developers interact with the system via Jupyter notebooks, shell access, or CERN's internal portals. While the workflow is straightforward, operational use in control rooms may introduce stricter time constraints, necessitating

[22] https://docs.lumi-supercomputer.eu/hardware/lumig

optimized deployment solutions. Kubernetes could be employed to manage model instances, providing scalability and load balancing.

Access to computational resources is governed by CERN's internal authentication protocols, with potential expansion to EuroHPC supercomputers. The primary non-technical challenges involve securing sufficient GPU resources for large-scale inference and fine-tuning, as well as maintaining data confidentiality.

The primary gaps identified in this use case include:

1. **Increased GPU Resources** – Expanded GPU availability for inference and fine-tuning.
2. **Scalability** – Dynamic scaling solutions for compute and storage resources.
3. **Data Security** – Enhanced confidentiality measures for sensitive internal datasets.
4. **Automation** – Automated workflow tools for streamlined model updates and deployments.

By addressing these challenges, CERN aims to fully integrate LLMs into its accelerator operations, enhancing productivity, facilitating faster learning, and streamlining complex processes across its infrastructure.

3.2. Representative use cases from Radio Astronomy

The Radio Astronomy use cases have been chosen to reflect current and future needs for a selection of major data-intensive radio observatories with substantial European user communities. In particular, we focus on:

- LOFAR, the LOw Frequency ARray, the world's largest and most sensitive telescope operating at low radio frequencies. LOFAR's physical infrastructure spans Europe, and it already operates a distributed data processing and archiving system. LOFAR is currently in the midst of a major mid-life upgrade (the "LOFAR2.0" effort), which will result in substantially higher data rates. Upgrading the data processing, dissemination, and analysis capabilities of the observatory are seen as critical for the success of LOFAR2.0.
- The SKA Observatory (SKAO), which is building two new telescopes in South Africa (SKA-Mid) and Australia (SKA-Low). These are expected to enter into operation by the end of this decade, and will constitute the most data-intensive astronomical observatory to date, generating of order an exabyte annually. Data from SKAO will be managed through a globally-distributed network of "regional centres", some of which will be based in Europe.

It is a common feature of each of these facilities that they act as reconfigurable and reusable observatories, each of which is designed to address a range of different science cases. For example, the SKA Observatory has convened 14 separate Science Working Groups (SWGs),[23] each of which represents the needs of a specific science community to the Observatory — these range from (for example) the physics of our Sun to the nature of the Big Bang and the early universe. Each SWG is planning to propose multiple scientific experiments, and it is beyond the scope of this work to produce use cases for all of these. In selecting use cases for analysis, we have focused on a subset of the possible science cases that represent a cross-section of the workloads that the observatories will be called upon to address. It is, however, important to bear in mind that the flexibility the instruments provide must be reflected in the associated data and compute continuum, which will have to continuously grow and adapt to new science goals.

We selected a range of science use cases that seek to illuminate the key technical challenges faced by each of these observatories. In addition, we chose one use case that illustrates the need for combining data from multiple observatories to drive scientific discovery. The remainder of this section provides a brief summary of each use case.

[23] https://www.skao.int/en/science-users/science-working-groups

3.2.1. The Power Spectrum of 21 cm HI Fluctuations (SKAO)

SKA Low will conduct surveys of the sky at frequencies between 50 and 220 MHz to observe the redshifted 21 cm neutral Hydrogen line in emission associated with the intergalactic medium as it is ionized by the first galaxies and active galactic nuclei. In this way, astronomers can probe the formation of structure in the early universe and track the imprint of the early galaxies on the surrounding gas.

The SKAO is not yet operational, and the survey strategy is still under development. However, the use case assumes a hierarchical series of surveys, starting with a wide (large area of the sky) but shallow (short integration time) survey, and gradually moving to smaller areas but with longer integration times (and hence higher sensitivity) per unit area.

Within the Observatory, data will be processed to a first level of scientific readiness (referred to as "Observatory Data Products", or ODPs). These ODPs are then supplied to a global network of SKA Regional Centres — the "SRCNet" — for archiving, dissemination, and analysis. The generation of ODPs within the Observatory is out of scope for this use case, which focuses purely on the SRCNet. The SKA Low telescope will support a total data rate of 350 PB/year to the SRCNet; the rate required to satisfy this science case will be less than this, and will vary depending on the detailed survey strategy. No individual SRC network node is expected to accept the total data volume; rather, it will likely be distributed across a global network of order 10 data centres. Facilitating effective analysis means that the data distribution must be intelligent: data which will be analysed together should be stored together, and it must be possible to transfer data between network nodes. The technologies and algorithms required to facilitate this are currently under development within the SRCNet construction effort; it is expected that they will be based on Rucio.

When data has been transferred to the SRCNet, the processing previously performed within the Observatory is repeated on a subset of the data to check calibration. A further stage of processing generates advanced data products: power spectra and images with foreground sources subtracted. Current codes for this work primarily make use of CPUs, but some or all of it will likely be migrated to GPUs in the future.

The interpretation of the data will benefit from large-scale cosmological simulations. These simulations may either be run directly on the SRCNet nodes themselves, or more likely on external systems: the eventual execution model will likely depend upon resource availability in the operational era. Preliminary analysis will be performed using semi-numerical simulations used to train an emulator (the training requiring access to a GPU); the final analysis will require high-resolution simulations, requiring tens of millions of CPU hours to generate.

All data products (both Observatory and Advanced) will be archived. The total volume required for products derived from both SKA Low and SKA Mid is expected to amount to around 700 PB/year; the fraction of that that is dedicated to this science project will be an operational decision.

Data products will ultimately be made available to end users to access either by download (although data volumes will often be prohibitive) or through online interactive analysis tools like Jupyter notebooks. Access to these systems will be moderated through an SKAO-administered federated authentication & authorization system likely based on INDIGO IAM.[24]

3.2.2. The Star Formation History of the Universe (SKAO)

While this use case also relies on the SKA Observatory, in contrast to §3.2.1 it makes use of the SKA Mid telescope in South Africa. The goal is to measure the average star formation rate in galaxies as a function of cosmic time by observing radio synchrotron continuum emission from galaxies around frequencies of 1 GHz.

Despite the use of a different telescope, there are many similarities with the previous use case. In particular, while the survey strategy is still under development, we anticipate a three-level hierarchical survey moving from wide and shallow to narrow and deep. Again, the first level of science products (ODPs) are generated

[24] https://indigo-iam.github.io

within the Observatory itself and supplied to the Regional Centre Network for archiving, dissemination, and analysis. As with SKA Low, SKA Mid supports a total data rate of 350 PB/year to the SRCNet; the rate that will be dedicated to this particular science case will vary depending on the detailed survey strategy and the decisions of the telescope's time allocation committee.

To satisfy this science case, data ("gridded visibilities") from multiple observing runs must be combined before images are made. This imposes additional constraints on the way that data is distributed across the SRCNet: it is necessary to ensure that data that will be combined is stored together to minimize data transport requirements. The combination and processing is then performed by large-scale batch workflows. The workflow definition language is yet to be selected. Current prototype codes are primarily based on CPUs, but it is likely that many of the key algorithms will be ported to GPUs before the survey commences. However, note that each of the SRCNet nodes may provide a different range of hardware resources: care is therefore needed to ensure the data is processed at nodes which have the appropriate capabilities. In total, it is expected that 35 PFLOPS of computing will be required to process the total data volume generated by both SKA telescopes combined; again, the fraction of that total which is allocated to this project will depend on survey strategy and telescope time allocation.

After the data has been processed, users may simply want to access either their own data or data from the archive, e.g. by obtaining previews of images or catalogues or downloading reduced volume datasets directly to their own computer. However, we do not expect that users will be routinely downloading large volumes of data to external compute resources. Instead, SRCNet will provide a "science gateway" that provides simple data-querying and discovery and the ability to "drill down" to analyze specific datasets with dedicated visualization tools or analysis environments like Jupyter notebooks. It is also expected that users will be able to define and run complex, multi-stage workflows on the data products, using the same frameworks as for the standard data processing described above.

3.2.3. Extragalactic Surveys (LOFAR)

Deep, wide-field extragalactic surveys provide an important resource for astronomers: they enable statistical studies of the properties of whole classes of astronomical objects, provide the opportunity to compare objects observed at one wavelength with the same objects observed at other wavelengths by cross-matching with other surveys or targeted observations, and open the prospect of serendipitous discovery of new source classes or astronomical phenomena.

LOFAR's combination of extremely wide field of view and exceptional sensitivity at low radio frequencies make it a uniquely capable survey instrument. Over its operational lifetime to date, the LOFAR community has conducted, analyzed and published a series of surveys with a wide range of scientific applications, in particular advancing our understanding of the formation and evolution of galaxies, clusters, and active galactic nuclei. Over the next several years, the LOFAR2.0 era will enable a new and even more scientifically productive range of surveys — with a concomitant increase in data management and processing requirements.

Data processing for surveys of this type is extremely challenging: the data is large and complex, the algorithms are computationally expensive, and research is ongoing into the most reliable and efficient approaches. Furthermore, the instrument must contend with interference from both terrestrial sources and extremely luminous celestial sources, even outside the nominal field of view of the telescope.

When operating in survey mode, LOFAR's antenna stations collect data at a rate of tens of Tb per second. The instrument itself performs a series of "online" (near-real-time) processing steps which are designed to reduce the data through combining, averaging, and correlating the data collected across the array, ultimately delivering tens of Gb per second to the central processing cluster (CEP), located in Groningen.

After arriving on CEP, the current mode of operation processes data through a further series of batch processing jobs. First, a preprocessing pipeline on CEP identifies and eliminates interference (from both terrestrial and celestial sources) and then compresses the data for onward transmission. It is then transferred to the LOFAR Long Term Archive (LTA) for storage, further processing, and dissemination. The LTA is currently distributed over three sites: SURF (Netherlands), Forschungszentrum Jülich (Germany) and Poznań Supercomputing & Networking Centre (Poland).

In the *current* LOFAR system, data is downloaded from the LTA by science teams and processed through the next stages of data reduction — direction-independent calibration, direction-dependent calibration, and imaging — using their own resources (e.g. university clusters, national facilities). In the *future* (LOFAR2.0) system, data processing will be carried out within the LTA itself.

Data processing is carried out using pipelines composed of software components (usually written in C++ and/or Python), with the overall workflow being described using the Common Workflow Language[25]. Current pipelines are all CPU-based, but work is underway to implement key algorithms using GPU accelerators. Data processing from a single observation can be parallelised and distributed along both time and frequency axes, but there are synchronization and combination points where data must be collected.

The current state-of-the-art in terms of LOFAR surveys, LoTSS Data Release 2, consists of around 170 PB of data products. However, this is expected to grow substantially in the LOFAR2.0 regime. Ultimately, it is expected that LOFAR2.0 will archive around 20 PB of data products and require 200 million core hours per year of data processing, although the detailed allocation of these resources to particular science projects has not yet been carried out.

After the calibrated and imaged data products have been stored in the LTA, they are made available to the astronomical community to access. Typically, products have a one-year proprietary period, during which they are only available to the science team that originally requested them; after that, they are made freely available.[26] Currently, data is made available for download to end user systems, but over the next several years the telescope expects to provision a "science platform" that will enable scientists to access and analyse data using tools like Jupyter notebooks running directly in the data centres. The observatory is in the process of rolling out a federated authentication and authorization system based on SURF Research Access Management[27] which will be used for both access to the LTA and to the future science platform.

3.2.4. Fast Radio Bursts and Transients (Multiple Facilities)

Transient and variable celestial events provide astronomers with an important tool for probing the high-energy universe. For example, over the last decade, the detection of gravitational waves has given us a direct view of the last moments in which neutron stars and black holes spiral into each other, while multi-wavelength observations have let us watch in near real-time bursts of radiation from accreting compact objects and the implosions of massive stars.

One class of astrophysical transient that remains an enigma are *fast radio bursts* (FRBs). First observed in 2007, these very short duration pulses of radio emission are much more luminous than any other similar phenomena. These events are also frequent: thousands are seen per day, isotropically distributed across the whole sky. Some FRBs are seen to repeat at the same place, while others are one-time-only events.

Our understanding of FRBs is limited by the relatively small fraction of the parameter space that has been searched so far. Although we know that many events are happening, in practice we are only able to observe perhaps 1% of the sky at any given time, and that using a limited set of frequency bands and timescales. The *EuroFlash*[28] project aims to address this issue by combining:

- Targeted, high-resolution studies of known FRBs and their local environments to properly explore their possible origins and nature;
- Studies of as-yet unexplored parameter space to discover new FRBs and other similar phenomena;
- Rapid, multi-wavelength follow-up observations with a range of telescopes to capture as much information as possible about new FRBs as they are discovered.

The most computationally-challenging and data-intensive part of this experiment will be the commensal — ie, in parallel with and relying on data taken for — search for new low-frequency FRBs and FRB-like signals

[25] https://www.commonwl.org
[26] LOFAR ERIC has committed to FAIR principles, but more work remains to make the current archive entirely FAIR.
[27] https://sram.surf.nl
[28] https://cordis.europa.eu/project/id/101098079

with LOFAR. The currently-envisaged system relies on a new cluster to be installed in parallel with existing LOFAR central processing systems. It will receive a duplicate copy of the data being sent to central processing; a total of 228 Gb/s during survey operations. This cluster will provide capabilities for searching both beamformed (timeseries) and image data from the telescope for new transients in real-time. This search will rely on versions of existing (open source) LOFAR pipelines (as employed, for example, in use case §3.2.3), modified to operate at high time resolution (10 ms), in addition to new, bespoke codes. The expected resource requirements during normal LOFAR2.0 operations will be:

- 1,024 CPU cores;
- 32 GB RAM per core;
- 64 GPUs with tensor cores;
- 8 PB storage.

The in-cluster storage provides a data buffer of at least one day, and means that long-period searches can be carried out. Ultimately, however, scientific data products will be flushed to the multi-site LOFAR Long Term Archive (described in §3.2.3) for preservation. Only data which directly contains transients of interest will be stored; the data volume for long-term archiving is not significant.

In addition to the blind-search system described above, modified back-end systems will be deployed at a range of other, higher-frequency, radio telescopes across Europe. These are intended not as search systems, but as monitors: they will track the behaviour of a set of FRBs over time, providing both high-cadence information about their temporal behaviour, and milliarcsecond-precision spatial localization. These systems will be based on a similar processing system to that installed at LOFAR, but are much less computationally demanding since the blind search is unnecessary.

When a new transient is detected, low-latency multi-wavelength follow-up observations (that is, observations of the transient at wavelengths and with telescopes other than the detection instrument) are vital to understand the nature of the physical processes at work. For this reason, the EuroFlash system will issue machine-readable alerts, using the standard *VOEvent* format,[29] when behaviour which merits follow-up is observed. This should be rapidly disseminated and (where appropriate) automatically acted upon by other facilities.

3.3. Other use cases

3.3.1. MISTRAL (Meteo Italian Supercomputing Portal)

MISTRAL is an acronym for Meteo Italian SupercompuTing poRtAL – was born as a project coordinated by Cineca, carried out in collaboration with key national stakeholders in the weather sector, such as National Civil Protection Department, Arpae, Arpa Piemonte and Dedagroup.

Funded in the context of the Connecting European Facility call for proposals in 2018, the project aimed to build a national open weather data platform to provide national and international citizens, public administrations, and private organizations with weather data from observational networks, historical and real-time analysis and forecasts. This goal has been realized with the bringing online of the Meteo-Hub platform[30] in 2020.

The Mistral use case within the SPECTRUM project exemplifies a robust, data-driven initiative designed to address significant scientific challenges in meteorology. Central to this effort is the MeteoHub platform, which serves as a comprehensive, harmonized repository for weather data, responding to the increasing demands for high-quality, real-time meteorological information. This initiative underscores the importance of accessible, standardized data in supporting researchers, public agencies, and private sector entities engaged in climatology, disaster management, and related fields.

[29] https://www.ivoa.net/documents/VOEvent
[30] https://meteohub.mistralportal.it

The primary goal of MeteoHub is to aggregate disparate datasets from numerous providers, ensuring rapid access to large volumes of weather data. This task necessitates addressing technical hurdles such as near real-time data availability, seamless integration across diverse sources, and enabling post-processing operations to tailor outputs for user-specific needs. By consolidating weather data, MeteoHub eliminates the inefficiencies of fragmented data repositories, thereby fostering enhanced research and operational applications.

From a data perspective, MeteoHub handles vast quantities of information, including approximately 2 TB of observational data and 280 TB of forecast data. Daily operations involve ingesting and processing high-frequency datasets, with retention policies ensuring lifelong data availability. Observational data flows into the platform via Apache Nifi[31], stored initially in a Postgres database (DBAIIE), and subsequently archived in Arkimet, a meteorological database. This architecture facilitates efficient data management and accessibility through APIs, which accommodate approximately 500 monthly data extraction requests from 100 active users.

On the computational side, MeteoHub relies heavily on CPUs and cloud-based virtual machines for data ingestion and extraction. HPC resources generate forecast data, with workflows focused on post-processing tasks such as computing derived meteorological variables, performing temporal and spatial operations, and converting data formats. Although parallelism is not currently utilized, potential scalability exists to manage future data growth and increased user demand. In the future, AI-based tools or GPUs could be integrated to enhance advanced post-processing capabilities and manage the anticipated growth in data volume.

Workflow management within MeteoHub is automated, leveraging Apache Nifi and containerized environments orchestrated by the Rapydo framework[32]. Observational data ingestion follows structured workflows, while user requests for data extraction are processed through Celery[33] containers, highlighting the platform's emphasis on efficiency and modular design. As part of its future evolution, MeteoHub plans to adopt Kubernetes[34] to improve scalability, addressing the growing need for dynamic resource allocation.

In terms of user interaction, MeteoHub offers access via Single Sign-On (SSO) for streamlined engagement with its web portal and APIs. Users can initiate and manage data extraction tasks through intuitive interfaces, with optional post-processing capabilities performed either on the platform or locally. Comprehensive user manuals, tutorials, and dedicated support channels further enhance user experience and accessibility.

Despite its robust infrastructure, MeteoHub faces challenges such as limited computational capacity and scalability constraints. Addressing these gaps through expanded resources and advanced orchestration techniques will ensure the platform's continued growth and relevance in the evolving landscape of meteorological research and operational forecasting.

3.3.2. LIGATE Molecular Dynamics

The LIGATE molecular dynamics (MD) use case highlights the application of exascale computing to genomics and precision medicine, with a focus on advancing molecular docking simulations for drug discovery. As part of the European LIGATE project[35], launched in 2021, this initiative leverages HPC to accelerate virtual screening processes, improve computational efficiency, and enhance the accuracy of drug candidate identification. By integrating innovative algorithms and scalable software, LIGATE aims to transform drug discovery and precision medicine by reducing computational costs and processing times.

Key objectives of the project include adapting molecular docking workflows for exascale environments, expediting drug discovery through advanced computational tools, and fostering collaboration among European researchers and institutions. LIGATE developed open-access resources, ensuring that data and simulation scripts remain available for future use. The project concluded in mid-2024, but the resulting datasets and methodologies continue to be accessible, reinforcing the project's long-term impact.

[31] https://nifi.apache.org
[32] https://rapydo.github.io/docs
[33] https://docs.celeryq.dev
[34] https://kubernetes.io
[35] https://www.ligateproject.eu

The computational demands of LIGATE primarily revolve around large-scale molecular dynamics (MD) simulations. Over the course of the project, approximately 16,000 simulations were conducted, with each producing 1 to 1.5 GB of data. In total, the simulation outputs required around 40 TB of storage. Post-processing operations further expanded this data volume. Stored at CINECA, the data archive is slated for transfer to the Molecular Dynamics Database (MDDB) portal by mid-2025, ensuring continued accessibility to the broader scientific community.

In terms of data transport, LIGATE's workflows involve minimal data movement during active simulations. Most data remains stored locally until post-simulation processing, at which point it is transferred for archiving or broader distribution. This workflow emphasizes efficiency, as the computational intensity of simulations far exceeds the data transfer requirements. Streaming data is not necessary, simplifying the overall infrastructure.

The compute architecture employed by LIGATE relies heavily on GPUs, as molecular dynamics simulations benefit significantly from GPU acceleration. The majority of simulation workflows can operate on a single GPU, making the process cost-effective and scalable. Despite ongoing interest in incorporating AI and quantum computing, current simulations remain grounded in traditional physics-based models, reflecting the limitations of AI in handling large biomolecular structures. GPU power advancements will incrementally enhance performance, but scaling beyond a single GPU remains challenging due to the fixed size of molecular systems.

Workflow management for LIGATE integrates various tools and frameworks, including Python scripts for SLURM job management, HyperQueue[36] for smaller simulations, and LEXIS[37] for cross-HPC workflows. While LEXIS demonstrated interoperability across HPC sites like LUMI and Leonardo, it was not deployed in production for long-duration MD simulations. SLURM-based workflows dominated due to their simplicity and reliability for large-scale tasks.

User access to LIGATE resources is facilitated through HPC-installed tools, custom software, and Jupyter notebooks. Future data accessibility will be supported by the MDDB portal, streamlining the retrieval and analysis of MD data through a user-friendly web interface.

Despite its technical successes, LIGATE faced non-technical challenges, notably the need for coordinated team efforts in simulation management and data analysis. The reliance on UNIX group permissions for resource access and occasional support from HPC infrastructure staff highlights the collaborative nature of the project.

Gap analysis reveals opportunities to simplify access to HPC resources across multiple sites and streamline workflow software to reduce operational overhead. Implementing fast, customizable data transfer mechanisms and unified access policies could further enhance the scalability and accessibility of LIGATE's computational framework.

3.3.3. Simulation of plastic neural networks in the brain

This use case focuses on simulating how neural networks in the brain change over time, a process known as structural and synaptic plasticity. As neuroscience becomes more data-driven, researchers are using tools like NEST[38] and Arbor[39] to build realistic brain models based on high-resolution imaging and experimental data. These models are computationally expensive, especially when simulating plasticity, which can make simulations run up to 100 times slower than real-time. This adds pressure on memory usage, communication between computing nodes, and storage performance.

This use case supports a wide range of researchers, including computational neuroscientists, experimental biologists, clinicians working on brain disorders, and engineers building brain-inspired technologies. The main simulation tools, NEST and Arbor, are open-source and part of the EBRAINS research infrastructure.

[36] https://it4innovations.github.io/hyperqueue/stable
[37] https://docs.lexis.tech
[38] https://www.ebrains.eu/tools/nest
[39] https://www.ebrains.eu/tools/arbor

NEST uses a C++ backend with MPI and OpenMP parallel programming and offers a Python interface. It performs well on large computing systems and is being extended to work on GPUs. Arbor is also written in C++ and optimized for GPU use. It supports detailed neuron shapes and complex plasticity mechanisms.

Simulations generate large amounts of data. For example, building a model of a brain region like the hippocampus or cerebellum can require 10 to 40 GB just for the connectivity data. Simulating activity and changes in the network adds another 2 to 5 GB per second of simulated biological time. Currently, for the most developed use case in NEST, a typical 6,000-second (simulation time, not biological time) run creates about 12.5 GB of raw data in around 55,000 files, plus a 54 MB summary after analysis. These results are written in parallel by the simulation processes. After processing, the data is stored for long term analysis. Although data privacy is not a major concern yet, future use of personalized models will need to meet data protection rules like GDPR.

The large input datasets are usually transferred once per experiment. After that, simulations access the data locally. The simulations are mostly run in batch mode. For example, NEST typically uses 10 computing nodes on the JUWELS cluster for 8 hours per job. Researchers needed to run about 2,000 such jobs to get enough data, averaging about 100 jobs per week. Arbor models are currently tested on 8-node GPU clusters for 4 hours at a time, but future runs on the new JUPITER supercomputer will involve up to 6,000 nodes for 24 hours. These large-scale simulations are scheduled between May and October 2025.

To make analysis faster and reduce wasted computing time, researchers are working on in-situ data analysis and interactive tools that let them check results and change parameters during a simulation. Right now, the workflow is mostly simulate-then-analyze, using job scheduling systems like Slurm or UNICORE. However, Common Workflow Language (CWL) is gaining popularity in EBRAINS, and new tools already allow users to make adjustments in real time and visualize ongoing simulations. These developments are laying the foundation for more integrated and responsive simulation pipelines that include machine learning and optimization.

There are also organizational challenges. The work is carried out by 8 to 12 research groups across Europe. We heavily rely on software and services by the EBRAINS RI, so it would also be ideal if we could have the EBRAINS AAI as authorization system, however this is not necessary. For the moment we use the HPC site-specific authentication and authorization services to gain access to specific projects. The SDL Neuroscience team helps with training and support by offering workshops, documentation, and tutorials. In the future, as simulations become more personalized, compliance with laws like GDPR and the EU AI Act will become more important.

4. Cross-cutting requirements

The analysis of representative use cases from HEP and RA reveals several cross-cutting requirements that are essential for the development of a robust European compute and data continuum for research. These requirements span multiple domains and highlight the need for common integrated solutions to address the challenges of data-intensive science in the Exascale era.

4.1. Storage

Total data volumes vary substantially by project, ranging from tens of GB for some projects (e.g. AccGPT) to an exabyte and beyond (total output from each of CERN and SKAO around 2030). While only a small fraction of this is foreseen to be transferred to and from HPC storage systems as necessary for compute, for larger allocations this may amount to many petabytes. A European data continuum should therefore be capable of managing data at an exabyte-per-project scale, but should gracefully scale to smaller projects as required.

By contrast, the typical data volume required for an individual unit of work is comparatively small: an individual collision in HEP or observation in RA may be only a few Megabytes, thus it is custom to combine many thousands into a chunked dataset for ease of storage and transport (1-10GB).

The data lifecycle and retention periods described in the use cases distinguish two general cases:

1. Raw data from a detector or telescope which requires some form of prompt processing, and secure redundant long-term (multi-decade) storage. This has generally remained under the custody of the experiment, as this requirement has been extremely challenging in the current project/proposal driven landscape of HPC access and the comparatively short machine lifespan.
2. Derived or Simulated datasets which are generally much smaller and have much shorter retention periods, if any. These datasets are generally reproducible, and thus rarely have requirements for redundancy or data safety.

Requirements for data safety, including redundancy, duplication, or archival storage technologies are similarly split into the two above cases. Raw datasets are committed to long-term tape storage after initial analysis datasets are derived from the processed raw, and both the raw and derived datasets are duplicated geographically for redundancy and to enable collaborative analysis with distant researchers.

Finally, all of the use cases describe a common need to make data available to the general public or scientific community. Many projects mention a requirement for proprietary or embargo periods on new datasets. Since not all projects have the same data policies, this implies that a flexible system of controlling access to data is necessary, with project administrators able to schedule public releases of their data on demand.

Data confidentiality requirements expressed are commonly to facilitate the scientific process (e.g. give the team that collected the data the chance to publish it first), rather than to protect sensitive information (e.g. personally-identifiable, commercially sensitive, security related). This implies that complex encryption schemes may not generally be necessary.

4.2. Data transport

Effective data access is crucial for most use case applications running on HPC sites. Data handling efficiency directly impacts use cases' ability to fully utilize HPC resources in a HTC manner, historically not designed for data intensive applications. Optimizing data access, alongside efficient software and applications, is key to maximizing HPC site utilization. Data transport is divided into three aspects in the use cases: during job execution (inter-site transport), data ingress/egress (intra-site transport), as well as mechanisms to automate transport.

During job execution, the use case's data transport needs are diverse. On one end, the vast majority of HEP experiment workloads are "embarrassingly parallel" data-driven jobs that do not require inter-node communication or performant fabrics, and instead put a collective strain on network file systems that scales linearly by job type and count (for example up to 10MB/s/core throughput projected from some CMS workloads). At the other end, use cases like LatticeQCD and AI/ML projects continue to drive the development of cutting-edge transport fabrics.

Due to the rate and volume of projected data production in all HEP and RA use cases, data transport in and out of HPC centers will become a severe bottleneck to utilization without high-speed data transfer services, sufficiently provisioned WAN connectivity, and capable data transfer nodes. HEP experiments have deployed high-speed automated, accountable, policy-based transfers between grid computing sites, and RA is likely to adopt these tools and strategies.

Finally, while advanced automation of inter- and intra-node transport has been available for some time with message passing standards (MPI and OpenMP), use cases mention many HPC sites today lack similar automation for WAN transport between remote storage, other HPC sites, or large data centers necessary to achieve automated data-driven job execution.

4.3. Compute

The increasing complexity of data analysis and simulation tasks in both HEP and RA has led to a growing demand for more powerful and efficient computing resources. Today's dominant use of CPU-based systems, while effective for certain workloads, struggles to keep pace with the parallel processing needs of

modern scientific applications. This challenge has prompted a significant and rapid shift towards GPU-enabled systems. GPUs, with their massively parallel architectures, offer substantial performance improvements for data-intensive computations, enabling faster processing of large datasets and more complex simulations and show promise to fulfill future compute projection requirement gaps.

In HEP, GPU acceleration is increasingly utilized for Monte Carlo simulations, event reconstruction, and real-time data filtering, significantly reducing computational time. These tasks involve repetitive, parallelizable calculations, making them ideal candidates for GPU processing. Similarly, in RA, GPUs facilitate the processing of massive datasets for imaging, signal detection, and calibration tasks, which require handling vast amounts of data and performing intricate calculations in real-time. Both HEP and RA describe development strategies for optimizing common frameworks and toolkits for GPU execution for cost and environmental reasons.

The surge in machine learning (ML) and deep learning (DL) applications further underscores the necessity for GPU resources. ML techniques are revolutionizing data analysis by enabling automated feature extraction, anomaly detection, and predictive modeling. In HEP, ML algorithms are employed to enhance particle identification, optimize detector performance, and accelerate data analysis pipelines. Similarly, in RA, ML methods improve source detection, classification of celestial objects, and noise reduction in observational data. These applications often require training complex neural networks on large datasets, a task that is computationally intensive and benefits significantly from the parallel processing capabilities of GPUs.

The growing integration of ML techniques necessitates scalable and accessible GPU infrastructures. This shift highlights the need for investment in high-performance computing (HPC) systems equipped with advanced GPU architectures. Furthermore, the development of specialized software frameworks and optimized algorithms is essential to fully harness GPU potential. Collaborative efforts between scientific communities and technology providers can accelerate the deployment of tailored solutions that address the specific computational needs of HEP and RA.

The escalating data demands in HEP and RA, compounded by the increasing adoption of machine learning methodologies, underscore the critical need for expanded and efficient GPU resources. The transition towards GPU-enabled systems is driven by the necessity to handle complex, large-scale data processing tasks and to support advanced analytical techniques. Addressing these computational challenges is vital to unlocking new scientific insights and sustaining the pace of discovery in these data-intensive disciplines.

4.4. Workflow management

All use cases described rely on automated workflow management for launching, monitoring, accounting, and data management. HEP experiments have well developed workflow management systems to manage the complex tasks involved in data processing, reprocessing, Monte Carlo event generation, detector simulation, and analysis. Integrating these management systems into HPC centers has required careful tailoring to match the computational resources, configuration, and policies. To maximize HPC adoption HEP experiments and LOFAR require a uniform mechanism(s) for integrating the HEP workflow management systems into the HPC centers. The SKAO is not yet operational and is still under development, however the requirements are expected to be similar.

4.5. Data access & analysis

Scientific workflows in radio astronomy and high energy physics require a hybrid computational model that supports both batch and interactive processing. Batch submission systems are well-suited for large-scale simulations and data processing tasks that can run unattended for extended periods. However, interactive data analysis is equally crucial, allowing researchers to perform exploratory analyses, visualize data, and refine computational models in real-time.

Balancing batch and interactive workloads requires flexible and adaptive resource management systems. Cloud computing paradigms, containerization technologies (e.g., Docker, Singularity), and on-demand

resource provisioning can offer the necessary flexibility. Interactive platforms like Jupyter Notebooks and integrated data analysis environments must be optimized to run efficiently on high-performance computing systems.

Additionally, supporting a diverse user base with varying computational needs necessitates user-friendly interfaces and tools that lower the barrier to entry for non-expert users while providing advanced capabilities for power users. This dual approach fosters collaboration and accelerates scientific discovery by enabling researchers to focus on analysis rather than infrastructure management

4.6. Non-technical challenges

Policy, Allocation periods, Strategic access

Large allocations of resources at HPC sites are generally awarded through peer-reviewed proposals and must be consumed within a year. In contrast, LHC experiments, with their multi-decade[40] physics programs and very-long-term computing commitments, favor multi-year allocations and guaranteed continuous site involvement via explicit MoUs with WLCG. Such allocations facilitate better resource planning and enable the experiments to secure resources that align with their long-term physics goals. Although LHC experiments invest in developing portable software that can be executed with minimal changes across various platforms, multi-year allocations (or at least annual "guaranteed" allocations) would allow them to capitalize on additional investments needed for specific HPC centers. Similar conditions exist for RA with long-term planning and operation of telescopes.

Currently, each HPC center operates with its own unique set of hardware, software, configurations, and policies. This diversity, while reflecting the specific needs and capabilities of each center and increasing the heterogeneity of the offer, poses a significant challenge for researchers and experiments that must configure and adapt their workflows to accommodate these differences manually, a process that can be time-consuming, error-prone, inefficient, in many cases need substantial R&D. A common set of standard interfaces, protocols, and policies would reduce the overhead and streamline the process of deploying workflows to multiple compute centers.

4.7. Gap analysis

Considering all the aspects of the use cases above, there are several common missing pieces that would yield the greatest success in HPC adoption.

- Developing a plan providing multi-year allocations for storage and compute, enabling HPC resources to more easily be incorporated into use case planning horizons.
- Rucio filetransfer endpoints setup on HPC storage partitions in a manner similar to HEP and RA storage sites would enable transparent data transfers and enable autonomous transfers via integration with use case's workload management systems.
- Develop standard facility API. Significant R&D to develop, but would ultimately reduce costs of future integrations.

HPC facility security models are increasingly moving toward a security posture, where multi-factor authentication (MFA) is the only acceptable method of user authentication to a site. While this model works well for individual (or small groups of) researchers submitting tasks to an HPC by hand via interactive shell, the process by which workloads are submitted to computing resources would ideally be entirely automated, modulo initial setup. Sites with restrictive MFA policies that require a human interaction component make automation extremely difficult. As such there is a general, demonstrable need to broaden acceptable authentication factors in HPC site policies to include some additional authentication element that does not preclude automation, while still meeting the security needs of these facilities.

As the SKA telescope time allocation process has not yet occurred, and the prototyping of SRCNet has just begun, many of the assumptions presented here are likely to evolve.

[40] https://lhc-commissioning.web.cern.ch/schedule/LHC-long-term.htm

Table 1: Requirements extracted from the information provided in the use case documents based on the SPECTRUM use case template. Note that some fields are marked as "Not specified" where the information was not clearly provided in the use case description. The requirements may evolve over time, especially for future projects and upgrades.

	MLPF reconstruction (LHC)	ATLAS Experiment	CMS Experiment	Extragalactic Surveys with LOFAR	LHCb Experiment	SKA Regional Centre Network Africa	SKA Regional Center Network Australia	AccGPT for CERN accelerator complex	Lattice QCD	ALICE Experiment	LOFAR Fast Radio Bursts	LIGATE Molecular Dynamics	EBrains	MISTRAL Meteo Hub	SRCNet AI for Hi-rich galaxy search
Community Size	25-12,000 members HEP community	6,000 members 43 countries	5,500 members 54 countries	10 countries	1,800 members 22 countries	SKA community 16 countries	SKA community 16 countries	4,000 members CERN community	2000 members 20 countries	2,000 members 38 countries	20 institutions 10 countries	11 institutions 5 countries	50 members 5 countries	Italian Meteo society, EU open data users	SKA community 16 countries
Annual Data Volume	Tens of TBs of raw data, up to 1 TB processed per detector	400 PB/year in 2029	146.20 PB RAW + 6.12 PB MC	~20 PB/year	100 PB to tape and 50 PB to disk per year (Run 3)	~700 PB/yr	~700 PB/yr	<10 GB of text data currently	1-10 PB per year	~10% annual growth from current 400 PB total storage	~8 PB/day buffered; small subsets retained long-term	40 TB/ yr	Not specified	~280TB	~700 PB/year
Storage Type	Distributed across multiple sites	Distributed across multiple sites (disk, tape)	Distributed across multiple sites (disk, tape)	Distributed across multiple sites	Distributed storage (disk, tape)	Distributed across SRCNet nodes	Distributed across SRCNet nodes	Local CERN infrastructure	Parallel file systems (e.g., Lustre, GPFS)	Disk (150 PB) and custodial tape storage (250 PB), XrootD protocol	Distributed across multiple sites	Central (CINECA, mddbr.eu)	Distributed across multiple sites	Central (CINECA, EU Open Data)	Distributed across SRCNet nodes with "hot" and "cold" tiers
Data Transfer Rate	Not specified	100 GB/s (grid transfers)	5 MB/s/core local, <100 Gbps bulk transfer	Few Gb/s to archive, <1 Gbps user access	<10 MB/s/core locally, <1 MB/s remote	>1 Gbps, TBD	>1 Gbps, TBD	Not specified	Several GB/s per job	1GB/s/core locally, <100GB/s bulk transfers	LOFAR: 130 Gbps during survey; EVN: 1–2 Gbps/telescope	<1Gbps	Several GB/s per job	<10Gbps	>1 Gbps, TBD
Compute Resources	4-8 HPC GPUs per training. 50k–100k GPU-hours/year	~600k cores continuously	Hundreds of thousands of simultaneous jobs	Varies, up to hundreds of thousands of CPU hours per pointing	Not specified	Up to 35 PFLOPS peak	Up to 35 PFLOPS peak	3-15 high-end GPUs for inference	Petascale to exascale computing	~250k cores, 10% from HPC	~1000 cores, ~60 GPUs during data taking	1 GPU/ workload	Petascale to exascale computing	<100 nodes per partition, ~10M corehours/ year/ partition	Up to 35 PFLOPS peak

	MLPF reconstruction (LHC)	ATLAS Experiment	CMS Experiment	Extragalactic Surveys with LOFAR	LHCb Experiment	SKA Regional Centre Network Africa	SKA Regional Center Network Australia	AccGPT for CERN accelerator complex	Lattice QCD	ALICE Experiment	LOFAR Fast Radio Bursts	LIGATE Molecular Dynamics	EBrains	MISTRAL Meteo Hub	SRCNet AI for Hi-rich galaxy search
CPU Architecture	x86	Primarily x86, support for ARM	Primarily x86, support for ARM, IBM POWER	Primarily CPU-based, moving towards GPU	x86, aarch64 for subset of applications	Not specified / TBD	Not specified/ TBD	x86	Primarily x86, support for ARM	Primarily x86, support for ARM	x86	x86	x86	x86	Not specified/ TBD
GPU Usage	Heavily used for ML training. Nvidia & AMD.	Limited, mainly for ML training and analysis	Increasing use, especially for ML algorithms	Moving towards GPU usage in next ~2 years	Exploring ways to speed up simulation with GPUs	Not specified	For AI training of emulators	Intensive for LLM inference and training. Nvidia.	Increasingly important	Critical for reconstruction jobs, AI workloads	Critical for data taking,	Constrained to 1 GPU per workload	Increasing use, especially for ML algorithms	In development	Planned for AI astronomical analysis
Memory Requirement	80GB GPU memory per GPU	<1 GB/core for 8-core jobs	2 GB/core (may increase)	Not specified	<2 GB/core	Not specified / TBD	No specified / TBD	25-410 GB GPU vRAM per instance	Not specified	2GB/core	30GB/core	40GB GPU	Not specified	<500GB/ node	No specified / TBD
Workflow Management	Not specified	PanDA, HTCondor, k8s, openstack	HTCondor, GlideinWMS	Moving towards CWL	DIRAC WMS	TBD	TBD	Not specified	Not specified		CWL defined for slurm	LEXIS, HyperQueue	Simulation followed by post-processing step	Apache NiFi, Rapydo framework	TBD
Access	Batch jobs, interactive analysis via JupyterLab	Batch jobs, interactive analysis via JupyterLab	Batch jobs, interactive analysis via JupyterLab	Moving towards online analysis with Jupyter notebooks	Batch jobs, interactive analysis via JupyterLab	Batch jobs, interactive analysis via JupyterLab	Batch jobs, interactive analysis via JupyterLab	Web interface, Jupyter notebooks	CLI	Batch jobs, interactive analysis via JupyterLab	Batch jobs, interactive analysis via JupyterLab	CLI, Interactive analysis via JupyterLab	Batch jobs, interactive analysis via JupyterLab	Web API calls	Batch jobs, interactive analysis via JupyterLab
Authentication	CERN SSO (OAuth2.0) with federated WLCG IAM	CERN SSO (OAuth2.0) with federated WLCG IAM	CERN SSO (OAuth2.0) with federated WLCG IAM	Moving towards federated authentication system	CERN SSO (OAuth2.0) with federated WLCG IAM	Federated AAI	Federated AAI	CERN SSO (OAuth2.0) with federated WLCG IAM	CERN SSO (OAuth2.0) with federated WLCG IAM	CERN SSO (OAuth2.0) with federated WLCG IAM	SURF SRAM	HPC site-specific authentication	HPC site-specific authentication	Meteo Hub AAI	Federated AAI

5. Reflections on other communities

The High Energy Physics community has benefited immensely from the creation of the distributed computing and storage platform today known as the WLCG. This computing model allows more than 10,000 physicists from around the world to collaborate and do analysis where the data is. Innovations in distributed computing, data management, and collaboration have influenced many fields: in astronomy, SKA and the Large Synoptic Survey Telescope (LSST) will use a similar grid model for handling massive datasets, as well as many tools and developments in federated access and data management. Life sciences have adopted distributed computing for genomics, such as in the ELIXIR[41] and Galaxy Project[42], as well as distributed research for COVID-19. Neuroscience and brain simulation communities have established the EBRAINS research infrastructure[43], which supports, amongst others, the EBRAINS use case discussed in this document. Collaborations with EGI continue to support climate sciences and earth observation in Copernicus[44] and IPCC[45] simulations.

Democratizing scientific research by breaking down traditional barriers to computing power, data access, and global collaboration becomes even more important as fields begin to confront exascale data and compute challenges. The ability to seamlessly integrate and move compute and data between different tier EuroHPC centers and scientific research infrastructures would facilitate collaboration in research as well as increase accessibility for smaller research groups and fields. Federation will also reduce the operational costs of HPC centers as it eliminates duplicate tasks of isolated sites and allows optimization or offload workloads across tiered resources and exposes a common environment for members of domain-specific communities.

Research areas and communities do differ in the scale and granularity of data access/storage/analysis and compute requirements, and in the best suited compute paradigms (such as HPC for closely coupled, potentially large-scale simulations, high-throughput computing (HTC) for ensembles of small-scale parallel simulation/analysis tasks, or AI inference, which will gain importance in the future). An effective, common federation solution must be able to accommodate such different requirements.

Finally, a common federated environment simplifies training and education, increasing HPC skills in Europe.

6. Conclusions

The comprehensive analysis of representative use cases from High Energy Physics (HEP) and Radio Astronomy (RA) conducted in this report provides crucial insights into the future of data-intensive scientific research and the requirements for a European compute and data continuum. This section summarizes the key findings, implications, and recommendations derived from our in-depth examination.

The following key findings have been identified:

1. **Unprecedented Data Volumes**: Both HEP and RA are entering an era of steep growth in data generation. The upcoming generation of instruments and experiments will produce data volumes several orders of magnitude larger than current capabilities, necessitating a paradigm shift in data management and processing strategies.
2. **Heterogeneous Computing Needs**: The analyzed use cases demonstrate a clear trend towards heterogeneous computing environments. While traditional CPU-based high-throughput computing remains essential, there is a growing demand for GPU acceleration and some interest in specialized AI hardware. Quantum computing resources are on the radar but, although there is some interest

[41] https://elixir-europe.org/platforms/compute
[42] https://galaxyproject.org
[43] https://www.ebrains.eu/
[44] https://www.copernicus.eu
[45] https://www.ipcc.ch

for R&D in this area, there are no plans considering quantum resources for production jobs in the foreseeable future.

3. **Complex Workflow Management**: As scientific workflows become increasingly sophisticated, there is a pressing need for advanced workflow management systems capable of orchestrating multi-step processes across distributed resources, ensuring data locality, and optimizing resource utilization.
4. **Data Federation and Accessibility**: The geographical distribution of research collaborations and the sheer scale of data involved necessitate robust data federation mechanisms. These must support efficient data discovery, access, and transfer across multiple sites and infrastructures..
5. **Scalability Challenges**: Scaling current infrastructure and workflows to Exascale levels presents significant technical challenges, particularly in areas such as I/O performance, network bandwidth, and energy efficiency.

The findings from this analysis provide concrete use cases and requirements that should directly inform the Strategic Research, Innovation and Deployment Agenda. Particular emphasis should be placed on addressing the identified scalability challenges and supporting the transition to heterogeneous computing environments. In addition, the Technical Blueprint for a European compute and data continuum must incorporate flexible architectures capable of supporting diverse workflow patterns, data management needs, and computing paradigms identified across the use cases. Furthermore, the identification of cross-cutting needs highlights opportunities for fostering collaboration between HEP, RA, and other scientific domains. This collaboration can lead to more efficient resource utilization and accelerated innovation in shared tools and methodologies.

The path towards a federated European Exabyte-scale research data and compute continuum is challenging but essential for maintaining Europe's position at the forefront of scientific discovery. As we move forward, it is crucial to maintain flexibility in our approach, continuously reassessing and adapting to evolving scientific needs and technological advancements. The SPECTRUM project, through its comprehensive analysis and strategic planning, is well-positioned to guide this evolution, fostering an ecosystem of innovation that will enable groundbreaking scientific discoveries in the Exascale era.

7. Annexes

- Annex 1: Use Case Documentation Template
- Annex 2: CMS Experiment Production Workloads 2029
- Annex 3: A Measurement of the Power Spectrum of 21cm HI Fluctuations during the Epoch of Reionisation and Cosmic Dawn
- Annex 4: Extragalactic Surveys with LOFAR
- Annex 5: Measuring the star formation history of the Universe
- Annex 6: Fast Radio Bursts & Astronomical Transients
- Annex 7: ATLAS Experiment
- Annex 8: Monte Carlo Simulation at LHCb Experiment
- Annex 9: ALICE Experiment
- Annex 10: AI-based particle flow reconstruction for LHC particle detectors
- Annex 11: LLMs for the CERN accelerator complex
- Annex 12: LIGATE MD
- Annex 13: Simulation of plastic neural networks in the brain
- Annex 14: Mistral Use Case
- Annex 15: Artificial Intelligence to find and characterize HI-rich galaxies

Annex 1: Use Case Documentation Template

Introduction

The SPECTRUM Project[46] will develop a blueprint and a research & development agenda for a data and compute continuum with the ultimate goal of supporting the full range of European scientific research communities. We start by taking the high-energy physics and radio astronomy communities as exemplars of the needs of the wider scientific ecosystem.

An important part of our analysis is to understand the use cases these communities have for data- and compute-intensive research. Since the project goals are to develop an agenda for future development, we are interested in current use cases and their expected evolution as well as predictions of significant or potentially disruptive use cases expected over the next decade. Please clearly lay out the relevant timescales as part of your use case.

This document provides a template that will be used to solicit use cases. It suggests a structure that breaks each use case down into major areas, and the types of information that we hope to gather for each of those areas. It is understood that detailed descriptions of all of these areas may not be possible, in particular for speculative future use cases; here, we encourage the use case authors to provide as much information (or informed estimates) as they have available.

Where appropriate, we encourage use case authors to include illustrations and diagrams to help explain their goals to a general audience. References to documents describing details of the use case are also welcome; please feel free to provide links to material publicly available elsewhere rather than duplicating it here.

This template will be accompanied by an example use case, prepared by the SPECTRUM project team, to demonstrate the sort of input we are hoping for.

Provide a Use Case Name

Scientific challenge

This section should address the following points:

- An introduction and description of the scientific background & objectives
- A short description of the scientific community served by the use case
- Scientific and technical challenges to achieve the objectives
- The timescale and scope of the use case — is this describing current work, or a vision for the future? If the latter, when?
- Open science and open data, including commitment to FAIR[47] data or other relevant principles and policies.

Storage

This section should address the following points:

- The total data volume required to address the scientific goals.
- The typical data volume required for an individual unit of work (e.g. job, service, or workflow).
- The data lifecycle, including retention period(s).
- Requirements for data safety, including redundancy, duplication, or archival storage technologies.

[46] https://www.spectrumproject.eu/

[47] Findable, Accessible, Interoperable, Reusable; https://www.go-fair.org/fair-principles/

- Needs for data confidentiality, including both motivation (e.g. privacy, commercial value, embargo periods) and expected technical measures (e.g. encryption).
- Typical storage patterns, including (for example):
    - Size and number of files or records, directory structures, read vs. write access.
    - File/DB access (contiguous, striped, random, ...).

Data transport

This section should address the following points:

- Typical compute job/service data throughput and access latencies, including (as applicable) both local and remote data access or transfer.
- Geographical extent of data transfers — within one site, across multiple sites, multiple infrastructures, ...
- How the data transfer is expected to evolve with time — e.g. constant transfer rates, linear increase, occasional bursts.
- Is streaming access to data required? If so, provide information on the expected latencies, data rates, and the stream topology (e.g. centralized, point-to-point)
- Data transfer technologies, protocols and tools (e.g. FTS, Globus, Unicore FTP)

Compute

This section should address the following points:

- Describe the general form of computing required by this use case (e.g. batch).
- What sort of compute platforms are currently used, or could be used in future? Consider e.g. CPUs, GPUs, AI and other accelerators, Quantum or Neuromorphic computing.
- Provide as much information as possible about the characteristics for a single unit of work (batch job, service invocation, etc). This could include:
    - Complexity and estimated runtime.
    - Types of operation (e.g. integer, floating point).
    - Other specific compute requirements (e.g. small data type support, tensor computing)
- At what rate, and in what number, are those units of work executed?
- Will the compute volume scale with time? If so, why (e.g. more data collected, desire to take advantage of more compute) and share thoughts on the scaling model (e.g. to multiple cores, multiple nodes, ...)
- Do you require specific frameworks, software stacks, or applications?

Workflow management

This section should address the following points:

- Does your use case involve automated, multi-step workflows? If so, please describe them in terms of number of steps, transfer of data between steps, and execution location (e.g. within one node, within one site, across multiple sites)
- Do your workflows contain time-critical steps?
- How is the workflow described (e.g. Common Workflow Language, Nextflow, ...)?
- What is the execution environment for the workflow? Consider orchestration (e.g. Kubernetes), monitoring, accounting, etc.

Access and analysis

This section should address the following points:

- How do you interact with the resources (interactive or offline/batch mode, access via ssh/scp, explicit service invocation, web portal, ...)?
- Do you need access to interactive facilities for computational steering?

- Will data be analyzed in-situ, or transferred to another site (e.g. local workstation) for analysis? In the latter case, how much data will be transferred?
- If analysing data in situ, which capabilities are required (e.g. remote visualization, Jupyter notebook, shell access).

Non-technical challenges

This section should address the following points:

- How do you or your collaboration obtain access to computational and data storage resources?
- How many individuals need to access the resources? What sort of authorization system do you rely on?
- Do you provide or make use of training, assistance, or support facilities provided by the e-infrastructure?
- Is your data covered by data protection or other regulatory frameworks (e.g. GDPR, AI Act).

Gap analysis

- Considering all of the aspects of your use case described above, what are the biggest missing pieces? What additional services or capabilities could make you most productive?

Annex 2: CMS Experiment Production Workloads 2029

The below example describes the total expected compute needs for 2029 based on the most recent estimates. In 2023 around 10 percent of the annual computing requirement was fulfilled by HPC centers. As such, this example does not aim to have the entirety of these needs fulfilled entirely by HPC, but rather to document the future expected technical requirements. This example document is a distillation of the original publications listed below and in the References. It is for example purposes and is not exhaustive.

The source and references for this summary can be found in https://cds.cern.ch/record/2815292 as well as elements from

1. https://doi.org/10.1088/1742-6596/898/9/092050,
2. https://doi.org/10.1051/epjconf/202024509012,
3. https://doi.org/10.1051/epjconf/202429501035,
4. https://doi.org/10.48550/arXiv.2312.00772 and
5. https://doi.org/10.48550/arXiv.2304.07376.

Scientific challenge

Background & objectives

The success of the physics program of the Compact Muon Solenoid (CMS) experiment at the Large Hadron Collider (LHC) critically depends on having sufficient computing resources to process, store, and analyze collision and simulated data samples in a timely manner. This will be no less true in the HL-LHC era starting in 2029. Early estimates from 2017 and 2018 [1] projected the resource needs to be several times greater than the project's budget, assuming a flat funding profile.

Today, HPCs are used in production by CMS for all kinds of central workflows, including event generation, simulation, digitization, pileup mixing, reconstruction, and creation of analysis data formats MiniAOD and NanoAOD. During the past years, HPC machines contributed significantly to the processing of the Run 2 data as well as the generation of the related Monte Carlo samples. Between five and ten percent of the total used computing power dedicated to that activity came from HPCs.

CMS extrapolations to Phase-II show a very steep dependency of computing needs on the number of simultaneous collisions per bunch crossing (pileup, PU).

The reconstruction task executes all of the algorithms needed to interpret signals as being due to the interaction of identifiable particles with the detector. The reconstruction time per event increases superlinearly with the number of pileup events per collision due to the combinatorial nature of the most resource-intensive algorithms. (PU of 52 in 2023, PU of 140 for Run 4, 200 for Run 5).

Scientific user community

The CMS experiment is a general purpose particle physics detector at the LHC at CERN. CMS has over 4000 particle physicists, engineers, computer scientists, technicians and students from around 240 institutes and universities from more than 50 countries. The collaboration operates and collects data from the CMS detector, one of the general-purpose particle detectors at CERN's LHC. Collaborators from all over the world helped design and fabricate components of the detector, which were brought to CERN for final assembly. Data collected by CMS are shared with several computing centres via the Worldwide LHC Computing Grid (WLCG). From there, they are distributed to CMS institutions in over forty countries for physics analysis.

Scientific and technical challenges

Looking ahead to the HL-LHC (High Luminosity Large Hadron Collider), the data rates and data complexity caused by increased luminosity will increase dramatically. All of this results in excess pressure on the current CMS computing infrastructure and an outlook for the HL-LHC where our existing computing infrastructure will simply not be sufficient to meet the demands of the experiment.

The ability to fully utilize large allocations on HPCs depends on having a submission infrastructure capable of scaling up to a sufficient number of tasks and execution slots from HTCondor. This implies work on federated access to HPC sites, standardized transfer tooling and policies, as well as common service and access methods.

Timescale and scope

The description that follows describes projections for run 4 and 5 of the LHC, expected to begin in 2029.

Open science and open data

The "CMS data preservation, re-use and open access policy" can be found here: https://opendata.cern.ch/record/415/files/CMS-Data-Policy-1.3.pdf

The CMS Collaboration recognizes the unique nature of CMS data and is committed to preserve its data and to allow their re-use by a wide community of collaboration members long after the data are taken, experimental and theoretical HEP scientists who were not members of the collaboration, educational and outreach initiatives, and citizen scientists in the general public.

Data produced by the LHC experiments are usually categorised in four different levels (DPHEP Study Group, 2009). The CERN Open Data portal focuses on the release of event data from levels 2 and 3. The LHC collaborations may also provide small samples of level 4 data.

CMS will provide open access to its data at different points in time with appropriate delays, which will allow CMS collaborators to fully exploit the scientific potential of the data before open access is triggered.

- At level 1, the additional data is made available at the moment of the publication, such as extra figures and tables.
- At level 2, simplified data format samples are released promptly as determined by the Collaboration Board.
- At level 3, public data releases, accompanied by stable, open source, software and suitable documentation, will take place regularly. CMS will normally make 50% of its data available 6 years after they have been taken. The proportion will rise to 100% within 10 years, or when the main analysis work on these data in CMS has ended. However, the amount of open data will be limited to 20% of data with the similar centre-of-mass energy and collision type while such data are still planned to be taken. The Collaboration Board can, in exceptional circumstances, decide to release some particular data sets either earlier or later.
- At level 4, small samples of raw data potentially useful for studies in the machine learning domain and beyond can be released together with level 3 formats. If storage space will be available, raw data can be made public after the end of all data taking and analysis.

For the widest possible re-use of the data, while protecting the Collaboration's liability and reputation, data will be released under the Creative Commons CC0 waiver[48]. Data will also be identified with persistent data identifiers, and it is expected that the third parties cite the public CMS data through these identifiers.

Storage

Data volume

Storage use consists of software, scratch, and input/output data. Access to calibration data and CMS software is normally provided via the Cern Virtual Machine File System (CVMFS), deployed as a service at a compute site and mounted to each node (or alternatively provided from the local network file system). Nodes generally require 20-50 GB of cache.

[48] https://creativecommons.org/publicdomain/zero/1.0/

Low-latency scratch space for actively running jobs is very useful for HEP workflows that often deal with many small configuration files. Requirements vary by job type, but typically are less than 2GB/core scratch.

Input/output data storage is highly dependent on workload (for example event generation has no input data). Event reconstruction (RECO) jobs, which process RAW datasets, are the most demanding. Typical RAW input datasets are a small fraction of the total produced, O(PB). Derived output datasets (results) are highly reduced, O(MB-GB). Dataset files typically contain many thousands of events per file, and are binned to a common size (eg 1 TB dataset split at 200GB segments)

Availability of outbound internet access from worker nodes to experiment owned storage at other sites (either directly or through a proxy edge service) allows input data reads and output data writes from jobs to bypass the local storage available at the HPC, at the cost of larger loads on networks and on the storage at CMS sites.

Table A2.1: Annual data volume by data type.

Event Type	Volume	Size per event	Total
RAW	34×10^9 events	4.30 MB	146.20 PB
MC	85×10^9 events	0.18 MB	6.12 PB

Data lifecycle & retention

Input datasets are ideally staged on shared storage for a period of weeks-months until jobs corresponding to that source are fulfilled. They are then evicted and a new dataset is fetched. If there is insufficient quota for long-term rotating staging, a smaller cache space may be used and data streamed from external (WAN) storage pools, although this impacts throughput and scalability. There is no long term retention expectation for input datasets.

Output data is transferred as soon as reasonable to external (WAN). It may be temporarily staged on local/shared storage awaiting transfer services.

There is no foreseen retention period once results are verified as successfully transferred.

Data safety

None foreseen at HPC sites, all source data is already duplicated and archived via WLCG resources.

Data confidentiality

Data is embargoed for physics analysis for a period of years, after which all data is publicly available. This is described in the section on open data policies above.

Typical storage patterns

Input data will dominantly be read sequentially and Analysis Object Data (AOD) formats permit the use of object storage technologies. Jobs are independent, and thus may have high impact during startup when loading software and conditions required. There is no restriction on how many jobs may read from the same input data concurrently, but output is independent.

Table A2.2: CMS event storage format is RAW/AOD/miniAOD/nanoAOD, ranked by projected size.

Tier	Event size [MB]	
	200 PU	140 PU
RAW	5.9	4.3
AOD	2	1.4
MiniAOD	0.25	0.18
NanoAOD	0.004	0.004

Table A2.3: CMS projected annual total events by type for runs 4 and 5.

Parameter	Run 4 ('29-'32)	Run 5 ('35-'38)
Common		
LHC Energy [TeV]	14	
Average PU	140 (70 in '29)	200 (100 in '35)
Integrated luminosity / year [fb^{-1}]	270 (135 in '29)	340 (170 in '35)
Livetime pp / year [s/10^6]	6	6
Livetime HI / year [s/10^6]	1.2	1.2
Yearly capacity evolution under flat budget for disk, CPU, and tape (hardware replacement included)	$+15 \pm 5\%$	
CMS-Specific		
Prompt HLT Rate [kHz]	5	7.5
Collected events / year (10^9)	34	51
MC events / year (10^9)	85	104
CPU-GPU cost ratio per unit computation	2.8x	

Data transport

Throughput and access latencies

Transfer throughput is expected to increase from 2.5MB/s/core in 2023 to an upper bound of 5MB/s/core in 2029. CMS does not currently require low latency storage, but depending on the size of the allocation at the HPC center, i.e. the number of cores, throughput from storage can become a bottleneck.

Geographical extent of data transfers

Large LHC data transfers should ideally be routed through private networks (such as GEANT) and avoid the public internet to guarantee acceptable latency, bandwidth and overall Quality of Service. This happens through the various national research networks which are interconnected worldwide into the primary WLCG grid LHCONE and LHCOPN networks.

How the data transfer is expected to evolve with time

Transfer is approximately linear with job scaling, transfer size will increase in 2030, and 2035 according to the LHC schedule due to event size, complexity, and frequency increase..

Streaming data

For running jobs with pre-placed datasets and availability of edge services and caches such as CVMFS, remote streaming data can be minimized to small conditions data, and job management communication (few Mbps/node). If data is not local, it can be streamed via transfer protocols and entirely avoid the local

storage system, at a much higher transfer requirement (5MB/s/core). On heavily restricted sites with no node connectivity, this has been avoided to varying degrees via message passing using the shared file system.

Data transfer technologies, protocols and tools

For the volume of datasets required, SSH-based tools will be insufficient. RUCIO is used by CMS for orchestrating transfers between Data Transfer Nodes (DTN) over a variety of protocols including FTS, S3, and Globus (although Globus is not preferred). XrootD is the preferred protocol for streaming job data on-demand. CMS is following new projects such as SENSE/AutoGOLE and NOTED, as well as leveraging modern advancements in SDN to further reduce transfer requirements and increase throughput.

Compute

General form of computing

Batch submission, data-driven parallel computing.

Compute platforms

Primarily x86, with support for ARM, IBM POWER; RISC-V possible when production hardware matures. GPU primarily CUDA, AMD/Intel being ported. CMS is heavily invested in replacing the most compute-expensive CPU functions with AI for several operations.

Job description

CMS workloads consist of several chain-able steps that may begin with collision event data (real or simulated), and result in analysis datasets. All steps may be performed in a single job with no intermediate files, or only a subset, as the software stack is common for all stages.

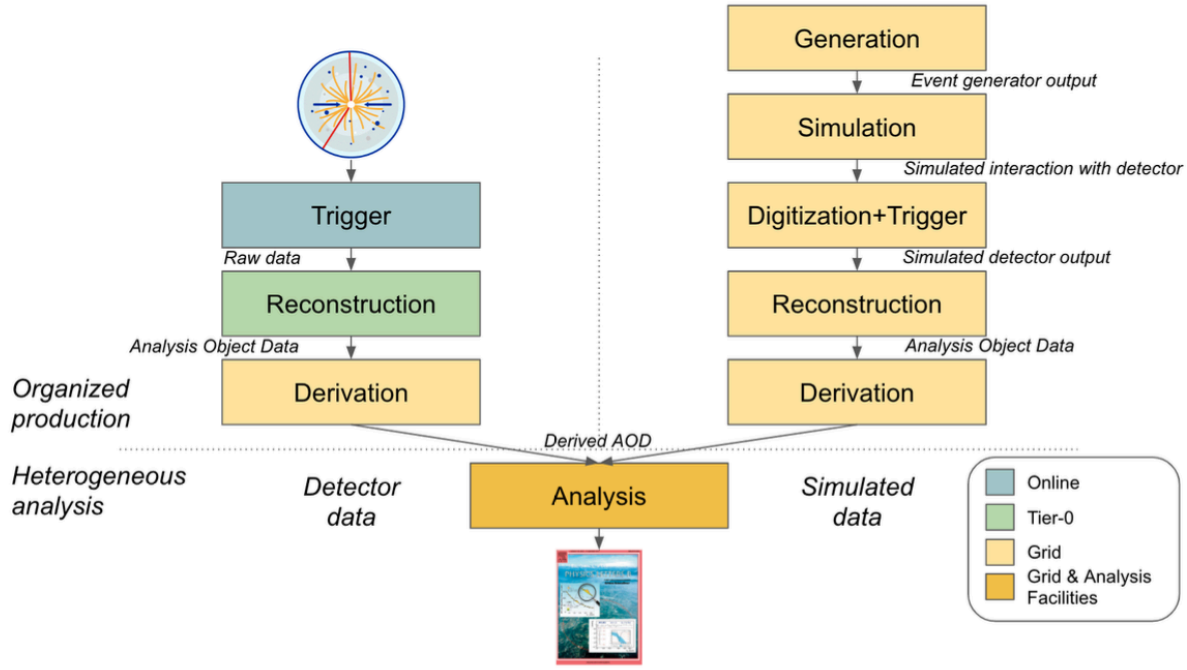


Figure A2.1: General CERN experiment compute steps.

The total compute need is dominated by Reconstruction (RECO/RECOsim) in which large 'raw' event data is transformed into comparatively smaller analysis data formats. **For the purposes of this example use case, we will detail only the reconstruction step.**

Reconstruction jobs will process RAW event types of the size 4.3MB/event. Each core currently requires 2GB memory, but this may increase with event complexity. Each event reconstruction further depends on *conditions* data describing the calibration of individual detector readout electronics and geometries, as well as information about the event collision pile-up (PU libraries/mix). These are provided via a squid proxy infrastructure. For efficiency reasons local squid proxies are preferred, but can also use the squid proxies at remote CMS sites if necessary (with some efficiency loss). Reconstruction time per event is driven by tracking, (particle track trajectory calculation) which represents about 45% of the total reconstruction time. A promising set of innovative ML driven approaches to reconstruction is being investigated in CMS, even if it is too early to assess their potential impact on the reduction of projected CPU needs.

Table A2.4: Processing times by type in HL pileup schemes.

Processing Step	Time/evt [HS06s]	
	200 PU	140 PU
Gen+Sim	1900	
Digi+PU mix+Reco	5100	3200

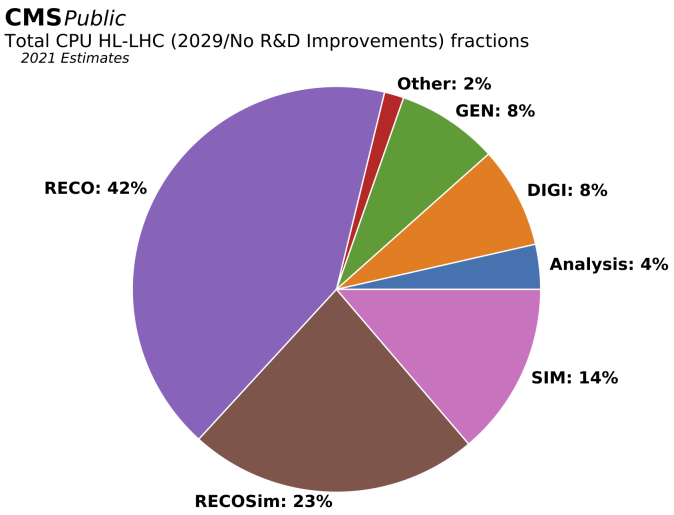


Figure A2.2: Total annual CPU processing fractions by processing step.

At what rate, and in what number, are those units of work executed?

Jobs consist of a batch of events to be processed. In 2023, Computing is benchmarked on the basis of:

- CMS GEN-SIM (MC prod.) 4 Threads: 1G/thread, 100evt/thread
- CMS DIGI (simulation digitization) 4 Threads: 2G/thread, 100evt/thread
- CMS RECO 4 Threads: 2G/thread, 100evt/thread

Multiple jobs are started until all CPU cores are consumed, for example 64 jobs would fit on a modern 256 core node.

From the CMS detector, RAW events will be recorded at around 5,000/second. For Reconstructions jobs, 34 Billion RAW events and another 85 Billion Simulated events are expected for processing in 2029, with several hundreds of thousands of simultaneous jobs running 24/7 to meet this need.

HS06 (HEPspec06) units depend on CPU efficiency. 2023 CPUs are up to 16 HS06 per core hour, 2029 CPUs may be closer to 20, depending on R&D.

Scaling of compute requirement with time

Within the same event pileup (PU) schedule delivered by the LHC, compute requirements will scale linearly with event production. Large step increases will occur when the pileup increases from 52 in 2023, to 70, 140, then 200, as the complexity of track reconstruction algorithms in RECO increases more than linearly with pileup, making tracking one of the main consumers of compute resources in the HL-LHC environment.

Software stack

The majority of the CMS offline software (CMSSW) is written in C++, with components in Fortran, and Python. The CMS software framework has the capability today to run production workflows on GPUs, offloading work on anything external to a CPU thread, whether an accelerator or another process. Current GPU algorithms under development for 2029 are written in a mixture of python, CUDA, and GPU-portable frameworks.

Software is generally read from CVMFS, or provided via container or local filesystem.

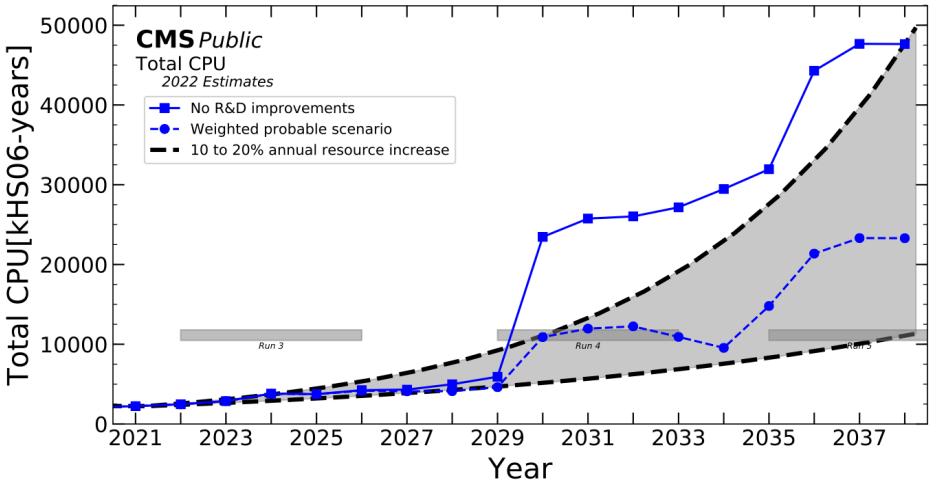


Figure A2.3: CMS Annual compute projections through 2040.

Workflow management

Workflow description

For the purposes of this example use case, the below describes central production workflows, and not user (physicist) submitted workflows, although the general principle is similar.

CMS employs heavily automated workflow management to coordinate roughly 500k CPUs across a global batch system. CMS computing jobs execute on independent, federated, compute nodes via *pull* mechanism (in contrast with SLURM *push* paradigm), where a (partially) empty node first runs a *pilot* step (via GlideInWMS) that reports back the available resources to the federated job pool (HTCondor). A job is then *pulled* by the node, according to matching resources, and execution begins. Remote job monitoring and reporting is facilitated by HTcondor, as well as dynamic resizing of resources.

CMS compute steps are capable of running in a chained pipeline within the same node, potentially eliminating the need for intermediate storage transfers between stages. In all cases, resulting job output is transferred to the collector

Do your workflows contain time-critical steps?

In general, CMS workflows have not been latency dependent, nor time-critical, however in high pileup runs there is a risk of data production exceeding the computational requirement to process it, resulting in a several year delay for physics analysis (or reduced precision of measurements).

How is the workflow described (e.g. Common Workflow Language, Nextflow, …)?

Job definitions are simple bash scripts which contain the necessary CMSSW commands and input data references.

What is the execution environment for the workflow? Consider orchestration (e.g. Kubernetes), monitoring, accounting, etc.

Workflows need access to WAN, ideally from the compute node, to communicate monitoring and scheduling information of jobs, and to pull external datasets. This may entail a dedicated or virtualized DTN service, both for File Transfer (via FTS/S3 etc), and XrootD-based DTN for on-demand streams. Squid proxy caching services further reduce the cost of these transfers.

CVMFS, ideally deployed as an edge service (either dedicated machine, or via Kubernetes), and mounted to each node is expected by CMS jobs. User namespaces provided by the kernel allows portable CVMFS access on sites where it has not been installed, at a higher cost to efficiency and networking. Workloads can execute in Apptainer by this same mechanism.

Access and analysis

This section should address the following points:

- How do you interact with the resources (interactive or offline/batch mode, access via ssh/scp, explicit service invocation, web portal, …)?
- Do you need access to interactive facilities for computational steering?
- Will data be analyzed in-situ, or transferred to another site (e.g. local workstation) for analysis? In the latter case, how much data will be transferred?
- If analysing data in situ, which capabilities are required (e.g. remote visualization, Jupyter notebook, shell access)?

How do you interact with the resources?

Compute sites are foreseen to be fully automated, outside of initial setup interaction. A small amount of compute time may be used interactively, likely via web interface (Jupyter notebooks) for analysis and development tasks from users.

Do you need access to interactive facilities for computational steering?

Interactive facilities are not needed for computational steering but physicists often use interactive jobs with e.g. Jupyter Notebooks for analysis.

Will data be analyzed in-situ, or transferred to another site (e.g. local workstation) for analysis? In the latter case, how much data will be transferred?

All batch job results will be transferred off-site to CMS collection points. Interactive jobs normally are not downloaded if the size is large, as efficient transfer mechanisms are already in place.

If analysing data in situ, which capabilities are required (e.g. remote visualization, Jupyter notebook, shell access)?

The largest request is for Jupyter notebooks via web interface.

Non-technical challenges

How do you or your collaboration obtain access to computational and data storage resources?

Access to CMS computational and data storage resources is typically obtained through institutional affiliations, collaborations, or project-specific agreements. User certificates and tokens are used for authentication.

How many individuals need to access the resources? What sort of authorization system do you rely on?

The CMS batch system is fully automated. Jobs are placed by the batch system on behalf of an authenticated individual or service, authenticated via token or certificate provided by the Interoperable Global Trust Federation (IGTF) framework.

Do you provide or make use of training, assistance, or support facilities provided by the e-infrastructure?

Any resources that individuals have access to is supported by both a CMS and CERN training services and support ticket systems.

Is your data covered by data protection or other regulatory frameworks (e.g. GDPR, AI Act)?

Event data contains no GDPR or AI-act coverage.

Gap analysis

- Considering all of the aspects of your use case described above, what are the biggest missing pieces? What additional services or capabilities could make you most productive?

Work with HPC providers to:

1. develop a plan for providing multi-year allocations for storage and compute such that HPC resources can be incorporated into CMS planning horizons. [2]
2. develop a plan for providing large storage allocations that can federate with existing CMS data management software, such that HPC storage can be used in a manner similar to how WLCG site storage is used today (i.e., as Rucio Storage Endpoints). [2]
3. standardize on a technology for providing a facility API that permits both interactive and automated access to manage job workflows and data, in order to integrate better with CMS and other large-scale multi-institutional scientific collaborations. If such an API were standardized, it would require some significant R&D up-front to integrate with our respective workflow management systems but would ultimately reduce the costs of initial integration and ongoing operations in the long run. [2]

Furthermore, HPC facility security models are increasingly moving toward a security posture, where multi-factor authentication (MFA) is the only acceptable method of user authentication to a site. While this model works well for individual (or small groups of) researchers submitting tasks to an HPC by hand via interactive shell, the process by which workloads are submitted to computing resources would ideally be entirely automated, modulo initial setup. Sites with restrictive MFA policies that require a human interaction

component make automation extremely difficult. As such there is a general, demonstrable need to broaden acceptable authentication factors in HPC site policies to include some additional authentication element that does not preclude automation, while still meeting the security needs of these facilities. [2]

While the largest HPC sites generally provide dedicated Data Transfer Nodes (DTNs), typically only the proprietary Globus Online service is supported for large-scale transfers. While Globus is appropriate for many of the users that HPC facilities service, LHC experiments instead rely on the XRootD protocol and SciToken-based authentication for large scale transfer of data between and within WLCG computing sites. Working together with HPC facilities to provide an XRootD-based DTN for use by LHC experiments would be of great help. If successful, this approach will significantly simplify integrating HPCs into the data federations for CMS, reducing unnecessary proxy transfers via the WAN while making the facility appear more similar to other WLCG sites. [2]

Due to queue times in the HPC batch system, the pilot jobs often stay pending for some time before they run, meaning that CMS jobs that originally triggered them will have run somewhere else in the meantime. As long as CMS can maintain a continuous job queue for HPC resources, this does not matter, the pilots will just execute other CMS jobs targeted at HPC. This doesn't necessarily mean utilizing HPC resources will require extra effort, but it does mean that in this model a good HPC utilization depends on a large enough fraction of the workflow mix that is in the system at any given time being able to run on HPC. [2]

References

1. CMS Phase-2 Computing Model: Update Document, 2022, https://cds.cern.ch/record/2815292
2. Barreiro-Megino, Fernando & Bryant, Lincoln & Hufnagel, Dirk & Anampa, Kenyi. (2023). US ATLAS and US CMS HPC and Cloud Blueprint.

Annex 3: A Measurement of the Power Spectrum of 21cm HI Fluctuations during the Epoch of Reionisation and Cosmic Dawn

Scientific challenge

The SKA Low frequency interferometer (SKA-Low) in Western Australia has been designed as an observatory, with a primary science goal of revealing the formation of structure and the imprint of galaxies on their surrounding gas in the Early Universe. In order to achieve this, SKA-Low will conduct surveys at frequencies between 50 and ~220 MHz in order to observe the redshifted 21cm neutral Hydrogen line in emission associated with the intergalactic medium as it is ionized by the first galaxies and active galactic nuclei. The survey could be conducted in three, successively smaller and deeper (longer integration time) tiers [6]. Data processing begins at the Science Data Processor (SDP) located at the HPC Science Processing centre in Perth, Western Australia. The SDP generates Observatory Data Products (ODPs) in the form of radio continuum images ("model", "residual" and "clean" images) as well as calibrated visibility data, averaged in time and frequency, and a local sky model catalogue. These are then transported over 100 Gps links to the SKA Regional Centres for further processing. Note that calibrated and uncompressed visibilities are not a standard SKA data product deliverable, and so special justification may be required. This scientific experiment aims to measure the power spectrum of the 21cm HI line, which is cleaner to extract from the visibility data than from images.

The SKA Regional Centre Network (SRCNet) will provide the user interface, analysis tools and long term data preservation of data products for the SKA Observatory (SKAO) [1]. The SKA scientific community will be globally distributed. These external scientific users and observatory staff should have the ability to search the archive and retrieve either ODPs, or Advanced Data Products (ADPs). For the purpose of this use case, we describe the user processing and handling of the extragalactic radio continuum images. The SRCNet architecture and operations plans to abide by the FAIR data principles.

For this use case, the survey could be conducted by dividing each station into smaller "substations" in order to sample more short baselines with a larger field of view. One proposed survey strategy would be to observe three tiers of different sizes and depths for 2500 hours each, with the widest tier planned to cover 10,000 square degrees, while the medium and deep tiers would cover 1000 and 100 square degrees, respectively. The wide and deep tiers could use two 150 MHz wide beams covering the sky frequency range of 50 to 200 MHz to improve the observing efficiency. The medium tier could be observed with a single beam of 300 MHz covering 50 to 350 MHz.

As the SRCNet hardware and software are scheduled for initial prototyping in late 2024 [4], and the full deployment is expected toward the end of the decade, this use case should be viewed as forward looking.

Storage

The data storage requirements for the SRCNet anticipate receiving ~700 PB total of ODPs from the two telescopes each year. These datasets will initially be used to make Project-Level Data Products (PLDPs) and ADPs that will be archived. The typical size of an ODP will depend on the length of the observing block, and for this experiment we anticipate a data rate of at least 6.5 Gbps at the beginning of SKA-LOW operations.

The survey project team users (scientific data analysts) will be provided with a personal file system for a limited duration (for example, Portable Operating System Interface (POSIX)), to which they can upload and download small files and submit new ADPs, including from the archive, to within the per-project resource allocation [5]. These files are available for further processing and visualisation, as well as to hold code and additional uploaded data for analysis. Processing logs will be stored along with the data for all processing

operations and will record information on software and resources used and any other parameters to ensure reproducibility of new data product generation.

In addition to the file system, users can generate their own databases which can be created, accessed, updated and queried using tools such as Structured Query Language (SQL), in order to store structured data relevant to their science cases [5]. Database rights will be shareable between groups of users.

ODPs will typically have a proprietary period of one year, giving the project teams the chance to analyse and publish their data before they are made public to the broader scientific community. All datasets will exist in multiple locations for security. It is possible that after one year, the data will be moved from a "hot" storage buffer to "cold" storage [3].

Data transport

Data will be stored across the SRCNet, throughout nodes located in Canada, Europe (including UK), South Africa (Cape Town), Australia, India (Pune) and China (Shanghai). There must be at least 1 Gbps upload and download speed between all nodes in the SRCNet. Users will be distributed globally, and not necessarily collocated with the data they are retrieving and viewing. Figure A3.1 shows the data flow and expected volumes and rates from the SKA-Low telescope to the end user for this use case - the Epoch of Reionization/Cosmic Dawn experiment.

It is anticipated that the data transfers will be continuous as new projects are completed at the telescope sites and data are moved to the SRCNet. The typical telescope observing block might be in the range of 4 to 8 hours, primarily during night time when ionospheric fluctuations are lower. These datasets will then be processed in Perth at the HPC Science Processing Centres before distribution to the SRCNet nodes. Real time data streaming to the SRCNet will not be required for this use case.

The user authentication system should use federated protocols that all the SRCs could use and, also, would allow different identity providers. The SRCNet will also make use of Virtual Observatory (VO) protocols to allow users to compare their SKA data products to images from other observatories.

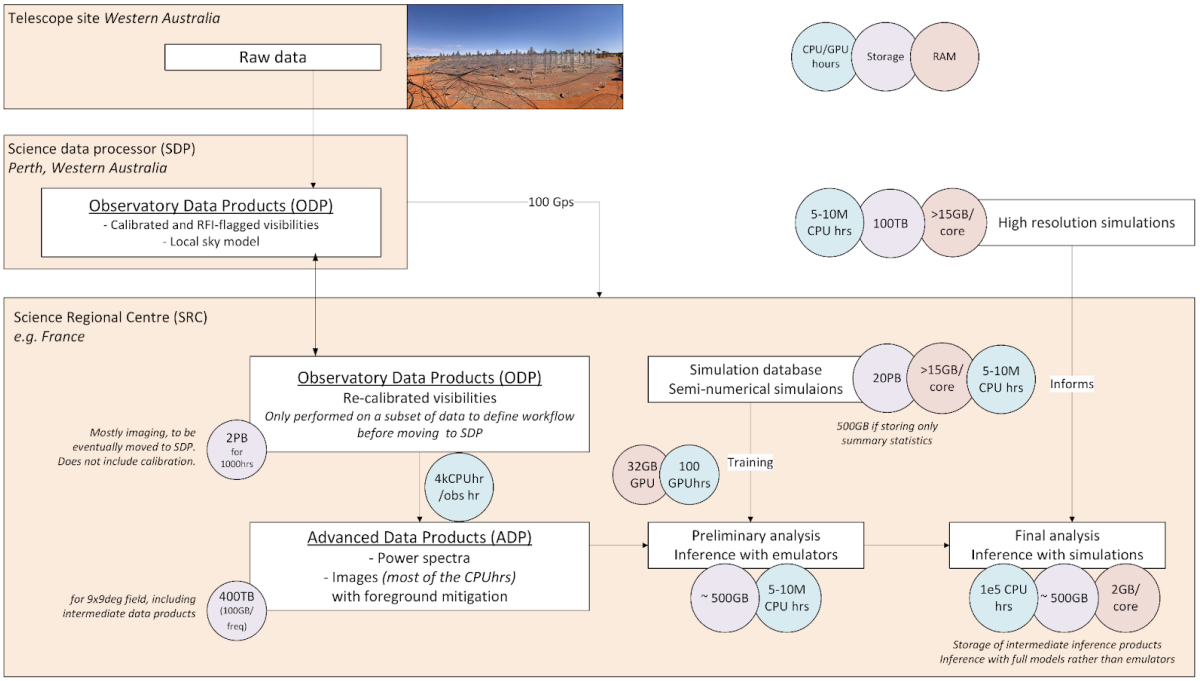


Figure A3.1: - Data flow and estimated file sizes for the various stages of the SKA-Low Epoch of Reionization/Cosmic Dawn experiment (A. Gorski & F. Mertens).

Compute

We assume that data products are processed multiple times in the SRCs (whilst the arrays are small the data rates are so low that this is not a driver) but less frequently once the arrays are fully built and the pipelines are more robust. Using a parametric model for the Science Data Processor (the HPC based in Perth and Cape Town), we estimate the total computing requirements of the SRCNet to be up to a peak of 35 PFLOPs. For good performance, high-performance storage close to the computing units could be needed ("hot" buffer) [4].

For the data analysis performed at the SDP stage, radio frequency interference (RFI) is removed before time averaging and smoothing the channels to a lower resolution. A sky foreground model is typically required for calibration of the images, and the SKAO maintains ownership of this model while updates may be provided by the survey team. The power spectrum calculated from the calibrated visibilities is the most computationally demanding and time consuming step, and will be conducted within the SRCNet. It is expected that reprocessing of data will also be required in order to improve the sky model, adjust the calibration strategy, and possibly excise low-level RFI. In addition to the power spectrum, spectral line data cubes and continuum images will be created by deconvolving the calibrated visibilities from all observing runs.

The data processing steps within the SRCNet will also require the extraction of advanced data products (ADPs) in the form of power spectra calculated both parallel and perpendicular (frequency, or redshift) to the plane of the sky. Typically, CPUs are used to run inference and GPUs are used to train emulators. This particular science case differs from many other SKA use cases in that it requires computationally expensive large-scale cosmological simulations (including hydrodynamics) in order to interpret the observations. It is anticipated that some groups may propose to run their simulations on the SRCNet nodes, which may require days of computing time as well as additional storage. Additional tools for the extraction of cosmological information will also include artificial intelligence models trained on these simulations.

More details on the computing and processing requirements will become available as the SRCNet prototyping activities evolve beginning at the end of 2024 and throughout the construction phase.

Workflow management

The SRCNet science analysis platform will facilitate workflow management with a tool that will allow users to specify individual workflow steps and combine them to form a larger workflow which can be stored in the software repository and re-used by others. It will be possible to combine workflow steps sequentially as well as using simple programming constructs (and, or, if), and each workflow step will draw on tools from the software repository, such as code within a notebook, a call to one of the pre-defined APIs, a call to a piece of software that is pre-installed within an existing defined software environment, or a call to another workflow [5].

After a workflow is defined, there will be options to run the workflow either in real-time or as a background process that can be scheduled to maximise efficiency and prioritisation, with the user being notified upon completion/failure.

The exact details of the workflows will become clearer as the SRCNet prototyping activities evolve.

Access and analysis

The SRCNet will implement a federated Authentication and Authorisation Infrastructure (AAI). This will integrate national federations through an international inter-federation service (e.g. eduGain) to enable the use of existing institutional accounts, and to allow the use of existing institutional credentials to authenticate with the SRCNet Infrastructure. The AAI will link these credentials to a centrally coordinated unique SRCNet Identity (SKA-ID?). A network of coordinated services will manage group membership and other relevant attributes, facilitating authorisation decisions for access to SRCNet data, computing, and other resources [3].

The main UI for the platform will be a web page (the 'Gateway', which will be hosted by the SRCNet) providing access to the functionality of the SRC node through a range of services. The user can sign on to the portal on the front page using single sign-on criteria and will then be given access to the full UI; without signing on there will likely only be limited public data access to, for example, image previews and catalogues. The UI will be consistent across different SRC nodes.

Once a dataset has been selected, users will need to be able to visualise the data either by running tools built into the platform, or by spawning a tool that runs within a software environment or notebook. The visualisation tools provided by the platform will include tools suitable for large datasets. Other panels will provide access to a notebook (possibly Jupyter) for interactive analysis and the ability to run containers, Virtual Machines (VMs), or distributed jobs, as well as constructing more complex workflows [5].

Non-technical challenges

The SKA Observatory data access policy gives exclusive rights to scientific project team members for a specified duration of time (currently assumed to be one year). After this proprietary period expires, anyone in the broader scientific community will be permitted to access the data. Survey teams may consist of ten, or more members, however it is likely that four, or five people will be actively working on a project data set.

Scientific and technical staff based at the SRCNet nodes will be expected to run regular training sessions for the community. One of the objectives of the SKAO and SRCNet operating model is to ensure that even non radio astronomers should be able to extract scientific data from the observatory.

GDPR will apply to the user account data, and possibly the AI Act will apply to machine learning models developed for astronomical data analysis.

Gap analysis

As the telescope time allocation process has not yet occurred, and the prototyping of SRCNet has just begun, many of the assumptions presented here are likely to evolve.

References

1. Bolton, R., et al., 2023, SRCNet Vision and Principles
2. Franzen, T. et al., 2023, SRCNet Use Cases
3. Salgado, J., et al., 2023a, SRCNet Software Architecture Document
4. Salgado, J., et al,, 2023b, SRCNet Top-Level Roadmap
5. Skipper, C. et al., 2023, SRCNet Science Analysis Platform Vision
6. Wagg, J., et al., 2021, SKA1 Scientific Use Cases

Annex 4: Extragalactic Surveys with LOFAR

Scientific challenge

Deep, wide-field surveys provide an important resource for the astronomical community. They enable statistical studies of the properties of whole classes of astronomical objects, provide the opportunity to compare objects observed at one wavelength with the same objects observed at other wavelengths by cross-matching with other surveys or targeted observations, and open the prospect of serendipitous discovery of new source classes or astronomical phenomena. Typically, such surveys are both immediately analyzed and drive the publication of new results, but also archived and serve as a legacy resource for the astronomical community for many years or decades to come. Prominent examples of such surveys include 2MASS,[49] DSS,[50] and NVSS.[51]

LOFAR[52] is the largest and most sensitive radio telescope operating at frequencies between 10 and 240 MHz. It consists of antennas that are individually sensitive to the whole visible sky, grouped into stations geographically distributed across Europe, and combined in software to produce a highly flexible and agile observing system. LOFAR provides unprecedented sensitivity, high angular resolution, and an extremely wide field of view (up to hundreds of square degrees, depending on configuration — compared with ten square degrees for the state-of-the-art in optical survey telescopes[53]).

These properties make LOFAR a uniquely valuable survey instrument, and this has been capitalized on by the LOFAR Surveys Key Science Project.[54] Over the ~15 years of LOFAR's operational lifetime to date, the Surveys KSP have conducted, analyzed, and published a series of surveys, with a wide range of scientific applications but primarily targeted to advance our understanding of the formation and evolution of galaxies, clusters, and active galactic nuclei.

Conducting surveys with LOFAR is challenging for a number of reasons. These include:

- Calibrating and making images from the telescope itself is fundamentally algorithmically challenging. In the early years of LOFAR operations, making science-grade images was a research problem in its own right. Contemporary processing pipelines now operate reliably when observing with only the central stations in the array and at higher frequencies; using the full European array (necessary to achieve the highest angular resolutions) or observing at lower frequencies is still extremely challenging.
- Calibration and imaging is compute intensive.
- Data rates generated by the telescope, and the resulting data volume of reduced data for analysis and archiving, are extremely high.
- Interference from terrestrial radio sources must be accounted for.

The most recent release of data from a large-scale LOFAR imaging survey is LoTSS[55] Data Release 2. This use case is most immediately based on the processing and analysis performed as part of that work. However, LOFAR is currently undergoing a major mid-life upgrade,[56] with a concomitant increase in data rates and computational requirements, and we are planning for future generations of LOFAR survey. We note in the text where future evolution is expected.

[49] The Two Micron All Sky Survey; Skrutskie et al., 2006, AJ, 131, 1163.
[50] The Digitized Sky Survey; https://archive.stsci.edu/dss/acknowledging.html
[51] The NRAO VLA Sky Survey; Condon et al., 1998, AJ, 115, 1693.
[52] https://www.lofar.eu/
[53] https://www.lsst.org/about/tel-site/optical_design
[54] https://lofar-surveys.org/
[55] LOFAR Two-meter Sky Survey; Tasse et al, 2021, A&A, 648, A1.
[56] LOFAR2.0; Van Cappellan et al, 2023, URSI GASS, 106.

All LOFAR data is governed by the LOFAR ERIC Data Policy,[57] which affirms that "LOFAR ERIC therefore undertakes to adhere to the principles of Open Science, Open Access, and FAIR data". In general, LOFAR data is subject to a predefined proprietary period during which it is available for the exclusive use of the team that proposed for the observation, and then becomes available to the community at large. To date, the bulk of data collected has been stored indefinitely, but increasing data volumes and the concomitant expense means that in future a "retirement" policy will be applied to some data products, selected by and subject to both technical and scientific review.

Storage

Radio astronomy data processing is, in some senses, a continuous set of data reduction steps. Broadly, the total data rate being collected by the antenna stations is on the order of 10s of Tb per second. This data is combined at the station level and shipped to the central "correlator" in Groningen, which has an input data rate on the order of 100s of Gb/s. The correlator provides an output rate of 10s of Gb/s to a co-located processing cluster. The cluster performs pre-processing of the data, which includes both averaging and compression,[58] and delivers preprocessed "visibility" data to the archive at a rate of a few Gb/s.

During LOFAR operations to date, the preprocessed data has accumulated in the archive, which now stores a total of around 60 PB. Science projects, including LoTSS, are responsible for downloading data from the archive, processing it to form science-ready products such as images (using their own resources) and disseminating the results.

In the future (LOFAR2.0) model, it is seen as unsustainable for the archive to store preprocessed data indefinitely. Instead, we anticipate a limited retention period (of perhaps 18 months) for this data, after which it will be removed. During that 18 months, the Observatory and the community will collaborate to process it to form science ready products, which will then be stored and disseminated through the archive. Over a five-year lifetime, around 100 PB of storage will be required for the LOFAR2.0 archive, depending on the details of the observing programme. Note that despite the data retention policy, the LOFAR2.0 instrument will generate more data per unit time than LOFAR.

During operations, the telescope will observe a given direction on the sky (a "pointing") for a specified period (typically 8 hours). Typically, two pointings can take place simultaneously. Data from each pointing is divided by frequency into 244 separate "subbands", which can be preprocessed and stored separately.

Science processing takes place in two phases. The first phase (direction-independent processing) combines the input subbands into groups of 10. All of these groups are then required for direction-dependent processing in the second phase. This results in science-ready products including images, calibrated visibilities, and assorted ancillary data.

The total scientific data volume distributed per LoTSS DR2 pointing is around 200 GB; the second LoTSS data release consists of 841 pointings and requires around 170 TB of storage in total. This is intended to be a legacy resource for the community, as therefore to be retained indefinitely.

Note that the future (LOFAR2.0) model will retain less visibility data, but the image data will be much larger due to the enhanced high-resolution capability of the telescope. The inability to archive the visibility data for the long term is seen as a problem, as history has shown that archival visibilities are frequently scientifically useful, but this is driven by practical concerns. One could imagine a future European storage landscape where it is possible to store visibilities bringing with it substantial benefits.

The data is not intrinsically sensitive or of high value. A proprietary period is applied to facilitate the scientific process, and enforced through the archive's authentication and authorization system.

The data can (in general) be reproduced by carrying out the observation a second time and reprocessing the resulting data products. This process has a cost associated with it. The need for replication or other

[57] https://www.lofar.eu/wp-content/uploads/2023/06/Data-policy_LOFAR-ERIC.pdf
[58] Offringa, 2016, A&A, 595, A99.

safety measures should be driven by an economic analysis including the cost of storage, cost of re-observing, and an evaluation of the future scientific impact of the data.

Data transport

Major data transport steps are:

- Within the telescope (from antenna to station electronics, from station to correlator, from hardware to central processing cluster); assumed to be out of scope for this analysis.
- From the central processing cluster to the archive. As above, the total data rate is in the range of few Gb/s (sustained). Latency is not critical, but throughput is: the central processing cluster does not have capacity to buffer data indefinitely before it is cleared to the archive. Note that the archive is distributed over multiple sites (currently SURF, NL; Forschungszentrum Jülich, Germany; Poznan Supercomputing and Networking Centre, Poland; expansion to additional sites is likely in future). A given observation will likely be sent only to a single site, but sustained transfer to any site should be possible when required. Data is transferred using FTS.
- Data is withdrawn from the archive by HTTP transfer. Within each archive site, data is managed using dCache, being written to tape for long term storage. When a user wishes to access the data, they submit a staging request to have the data transferred to disk. This process may take multiple hours. When the data is ready for download, the user is emailed HTTP URLs.
- Processed data products are currently made available for download by HTTP from the SURF Data Repository.[59] Access latency is not critical.

Over the next five years:

- The data volume transferred to the archive will increase significantly.
- More data processing will take place within the archive, so fewer bulk downloads of data will be necessary.
- Data may need to be migrated between data centres for processing.
- Users will both up and download science-ready data from the archive.

Over the next 10 years:

- Latency of data transfer to the archive will become critical, as data volumes become increasingly large compared to the buffers available within central processing.

Compute

The main LOFAR data reduction steps (preprocessing, direction-independent calibration, direction-dependent calibration and imaging) are currently executed as batch jobs. At present, all of them are executed as pure CPU codes on a series of SLURM clusters (preprocessing is carried out on the telescope's central processing cluster in Groningen; other steps in the various external data centres described above).

Some aspects of radio astronomy processing are well suited to the use of GPUs. Various aspects of the algorithms have been prototyped and demonstrated to show improved performance on GPU hardware. However, as of now, no GPU-based codes are used in production processing. This is expected to change over the next ~2 years.

The applications of quantum computing in radio astronomy have been explored in the literature,[60] but the results are not currently encouraging ("with the hardware currently available, no tangible quantum advantage can be identified"). The authors identify possible future benefits with improved hardware; we will continue to monitor the field, but believe any concrete planning or investment would be premature.

[59] https://repository.surfsara.nl/collection/lotss-dr2

[60] Brunet et al, 2024, A&C, 47, 100796; Renaud et al, 2024, A&C, 47, 100803

The units of work (pointings, subbands) are described under storage, above. In the case of LoTSS DR2 — which includes only stations located in the Netherlands — processing each pointing requires on the order of 5,000 CPU hours. Including the full range of international stations, as is the plan for LOFAR2.0, increases this number dramatically to perhaps hundreds of thousands of CPU hours per pointing. However, this is an area of active research; algorithmic development has been progressing rapidly, and this number has fallen by a factor of several in just a few years.

Future upgrades are likely to cause substantial increases in data volumes, due to both increased flexibility of the observing modes and increased (network and observing) bandwidth. In a regime where latency is not the primary consideration, work can be conveniently distributed along the time axis (that is, each pointing is sent to a different node for processing). However, latency reduction requires that work also be distributed along the frequency axis.

Workflow management

Workflows consist primarily of executable code written in C++ and Python. To date, a heterogeneous mixture of workflow frameworks have been used, including e.g. Stimela;[61] the current effort is mainly standardized around CWL. Workflows are primarily executed on SLURM clusters, using a variety of bespoke solutions for monitoring.

Typical radio astronomy survey workflows include LINC,[62] Rapthor,[63] and DDF-Pipeline.[64] Each unit of work (pointing) is executed within one site. Data is transferred between steps using the filesystem.

Access and analysis

Current situation

The bulk data products generated during processing (images, calibrated visibilities) are made available through the SURF Data Repository.[65] Users can select and download individual pointings for offline analysis.

During processing, a source catalogue (ie, list of all astronomical objects detected in the images) is also generated, and is available for download separately. Many analyses can be based either entirely on the catalogue, or can use the catalogue to select only the relevant pointings for download, thereby massively reducing the data volumes needed to perform analysis.

In addition, cross-matching between the radio catalogue and optical counterparts, as well as spectroscopic classification of radio sources, has been performed by the LOFAR Surveys team. These are vital inputs for future scientific analysis.

All of these data products and the means of accessing them are listed on the LOFAR Surveys website.[66] Where possible, data is also made available through International Virtual Observatory Alliance (IVOA) interfaces,[67] which provide standard mechanisms for data interoperability and transfer across astronomical data centres.

Future situation

In the future, we expect:

[61] https://stimela.readthedocs.io/en/latest/
[62] https://linc.readthedocs.io/en/latest/
[63] https://rapthor.readthedocs.io/en/latest/
[64] https://github.com/mhardcastle/ddf-pipeline/tree/master
[65] https://repository.surfsara.nl/collection/lotss-dr2
[66] https://lofar-surveys.org/dr2_release.html
[67] https://www.ivoa.net/

- Advanced / scientific / reduced data products will be hosted on the same infrastructure (the LOFAR LTA) as instrumental data, rather than in the SURF Data Repository.
- Science teams will be required to contribute advanced products back to the LTA when they are published.
- Although data will continue to be available for download, many smaller-scale jobs will be performed online, within the LTA. Astronomers are likely to use Jupyter notebooks as the primary interface, with to be determined systems for bulk processing (much current work focuses on Dask, but no decisions have been taken). This will be subject to quota / resource allocation.
- Visualization tools such as CARTA[68] will provide progressive download and visualization of massive data volumes.

Non-technical challenges

Current situation

Preprocessing and LTA data storage is performed on infrastructure provided by the national partners in LOFAR ERIC at SURF, FZJ and PSNC. Contributions from each nation to the LOFAR infrastructure are counted as part of their subscription costs to the ERIC. How each nation procures the infrastructure internally varies from case to case. ASTRON manages an authentication & authorization system for the LTA.

Science processing after the data has been downloaded from the LTA is coordinated by the LOFAR Surveys project on a heterogenous variety of clusters across Europe. These include university infrastructure and data centre access provided through proposal calls.

End user analysis is performed by downloading data onto their own system, be it a laptop or a cluster (obtained in the same ways as above) depending on the type of processing.

Future situation

The bulk of science processing is expected to transition to centralized infrastructure, provided by the ERIC partners. Advanced data products will be stored in the LOFAR LTA after they have been generated.

Interactive and bulk processing analysis facilities (see previous section) will also be provisioned in the LTA.

Access control will be through a federation authentication & authorization system (likely SURF Research Access Management[69]).

Management of user data is subject to the GDPR.

Gap analysis

The most immediate and practical need is for structural support of bulk storage and computing allocations. Ideally, this would happen at the European scale, rather than relying on individual nations to make their own arrangements.

It would be ideal to have a single control panel that provides a unified view of resources, data, and activity across all of our LTA sites. Currently, ASTRON manages this for itself, but this is made more complex by the heterogeneity across the various data centres.

Since we are reliant on distributed infrastructure, the more homogeneous that infrastructure is, the easier it is to address. This includes everything from the underlying hardware (what systems do we optimize for?) to storage mechanisms, authentication & authorization, etc.

[68] https://cartavis.org/
[69] https://sram.surf.nl/

In the current system, when data is sent to an LTA site, that is effectively its permanent home: it will be stored and processed there. Joint processing of that data with data located at a different site is not provided for, and moving data between sites would be logistically complex. A major upgrade would be to move to a more dynamic system, where data and processing jobs can migrate relatively seamlessly between the sites.

There is currently no framework for low-latency data processing.

The “science platform” data analysis functionality described above does not yet exist. Various development efforts are underway both inside and outside radio astronomy (it seems that almost every project at the moment wants to develop a platform); in particular, we closely track the work being carried out in the SKA Regional Centre Network. However, at the moment, this landscape seems fragmented and confused. A truly flexible, scalable off-the-shelf science platform would be a major asset.

Annex 5: Measuring the star formation history of the Universe

Scientific challenge

The Square Kilometre Array (SKA) Mid frequency interferometer (SKA-Mid) is capable of measuring the average star-formation rate in galaxies over cosmic time. This is achieved through observations of radio synchrotron continuum emission from galaxies around frequencies of ~1 GHz (band 2 for SKA-Mid). The survey would be conducted in three, successively smaller and deeper (longer integration time) tiers [6]. Data processing begins at the Science Data Processor (SDP) located at the HPC Science Processing centre in Cape Town, South Africa. The SDP generates Observatory Data Products (ODPs) in the form of radio continuum images ("model", "residual" and "clean" images). These are then transported over 100 Gps links to the SKA Regional Centres for further processing.

The SKA Regional Centre Network (SRCNet) will provide the user interface, analysis tools and long term data preservation of data products for the SKA Observatory (SKAO) [1]. The SKA scientific community will be globally distributed. These external scientific users and observatory staff should have the ability to search the archive and retrieve either ODPs, or Advanced Data Products (ADPs). For the purpose of this use case, we describe the user processing and handling of the extragalactic radio continuum images. The SRCNet architecture and operations plans to abide by the FAIR data principles.

As the SRCNet hardware and software are scheduled for initial prototyping in late 2024 [4], and the full deployment is expected toward the end of the decade, this use case should be viewed as forward looking.

Storage

The data storage requirements for the SRCNet anticipate receiving ~700 PB total of ODPs from the two telescopes each year. These datasets will initially be used to make Project Level Data Products (PLDPs) and ADPs that will be archived. The typical size of an ODP for this use case will vary depending on the size of the field imaged during an observing session.

The users (scientific data analysts) will be provided with a personal file system for a limited duration (for example, Portable Operating System Interface (POSIX)), to which they can upload and download small files and submit new ADPs, including from the archive, to within the per-project resource allocation [5]. These files are available for further processing and visualisation, as well as to hold code and additional uploaded data for analysis. Processing logs will be stored along with the data for all processing operations and will record information on software and resources used and any other parameters to ensure reproducibility of new data product generation.

In addition to the file system, users can generate their own databases which can be created, accessed, updated and queried using tools such as Structured Query Language (SQL), in order to store structured data relevant to their science cases [5]. Database rights will be shareable between groups of users.

ODPs will typically have a proprietary period of one year, giving the project teams the chance to analyse and publish their data before they are made public to the broader scientific community. All datasets will exist in multiple locations for security. It is possible that after one year, the data will be moved from a "hot" storage buffer to "cold" storage [3].

Data transport

Data will be stored across the SRCNet, throughout nodes located in Canada, Europe (including UK), South Africa (Cape Town), Australia, India (Pune) and China (Shanghai). There must be at least 1 Gbps upload and download speed between all nodes in the SRCNet. Users will be distributed globally, and not necessarily collocated with the data they are retrieving and viewing.

It is anticipated that the data transfers will come in continuous waves as new projects are completed at the telescope sites and data are moved to the SRCNet. The typical telescope observing block might be in the range of 4 to 12 hours. These datasets will then be processed in Perth and Cape Town at the HPC Science Processing Centres before distribution to the SRCNet nodes. We do not anticipate that real time data streaming to the SRCNet will be required for this use case.

The user authentication system should use federated protocols that all the SRCs could use and, also, would allow different identity providers. The SRCNet will also make use of Virtual Observatory (VO) protocols to allow users to compare their SKA data products to images from other observatories.

Compute

We assume that data products are processed multiple times in the SRCs (whilst the arrays are small the data rates are so low that this is not a driver) but less frequently once the arrays are fully built and the pipelines are more robust. Using a parametric model for the Science Data Processor (the HPC based in Perth and Cape Town), we estimate the total computing of the SRCNet to be up to a peak of 35 PFLOPS. For good performance, high-performance storage close to the computing units could be needed ("hot" buffer) [4].

For this use case, continuum images are created by combining gridded visibilities from multiple observing runs and then deconvolving. The pixel resolution of the image should be 0.125" in order to fully sample the synthetic beam size of 0.5" which requires imaging with baselines extending out to 90km.

More details on the computing and processing requirements will become available as the SRCNet prototyping activities evolve beginning at the end of 2024 and throughout the construction phase.

Workflow management

The SRCNet science analysis platform will facilitate workflow management with a tool that will allow users to specify individual workflow steps and combine them to form a larger workflow which can be stored in the software repository and re-used by others. It will be possible to combine workflow steps sequentially as well as using simple programming constructs (and, or, if), and each workflow step will draw on tools from the software repository, such as code within a notebook, a call to one of the pre-defined APIs, a call to a piece of software that is pre-installed within an existing defined software environment, or a call to another workflow [5].

After a workflow is defined, there will be options to run the workflow either in real-time or as a background process that can be scheduled to maximise efficiency and prioritisation, with the user being notified upon completion/failure.

The exact details of the workflows will become clearer as the SRCNet prototyping activities evolve, beginning at the end of 2024.

Access and analysis

The SRCNet will implement a federated Authentication and Authorisation Infrastructure (AAI). This will integrate national federations through an international inter-federation service (e.g. eduGain) to enable the use of existing institutional accounts, and to allow the use of existing institutional credentials to authenticate with the SRCNet Infrastructure. The AAI will link these credentials to a centrally coordinated unique SRCNet Identity (SKA-ID?). A network of coordinated services will manage group membership and other relevant attributes, facilitating authorisation decisions for access to SRCNet data, computing, and other resources [3].

The main UI for the platform will be a web page (the 'Gateway', which will be hosted by the SRCNet) providing access to the functionality of the SRC node through a range of services. The user can sign on to the portal on the front page using single sign-on criteria and will then be given access to the full UI; without signing on there will likely only be limited public data access to, for example, image previews and catalogues. The UI will be consistent across different SRC nodes.

Many users will simply want to access either their own data or data from the archive, e.g. by obtaining previews of images or catalogues or downloading reduced volume datasets directly to their own computer; due to the amount of data expected, and the nature of a server-side oriented science platform for analysis, we do not expect that users will be routinely downloading large volumes of data to external compute resources. Simple data-querying and discovery will be the default panel, allowing users who do not wish to carry out more sophisticated analysis to go straight to the data. Once a dataset has been selected, users will need to be able to visualise the data (images, spectra, cubes, catalogues, light curves etc) either by running tools built in to the platform, or by spawning a tool that runs within a software environment or notebook. The visualisation tools provided by the platform will include tools suitable for large datasets. Other panels will provide access to a notebook (possibly Jupyter) for interactive analysis and the ability to run containers, Virtual Machines (VMs), or distributed jobs, as well as constructing more complex workflows [5].

Non-technical challenges

The SKA Observatory data access policy gives exclusive rights to scientific project team members for a specified duration of time (currently assumed to be one year). After this proprietary period expires, anyone in the broader scientific community will be permitted to access the data. Survey teams may consist of ten, or more members, however it is likely that only a small number of people will be actively working on a project data set.

Scientific and technical staff based at the SRCNet nodes will be expected to run regular training sessions for the community. One of the objectives of the SKAO and SRCNet operating model is to ensure that even non radio astronomers should be able to extract scientific data from the observatory.

GDPR will apply to the user account data, and possibly the AI Act will apply to machine learning models developed for astronomical data analysis.

Gap analysis

As the telescope time allocation process has not yet occurred, and the prototyping of SRCNet has just begun, many of the assumptions presented here are likely to evolve.

References

1. Bolton, R., et al., 2023, SRCNet Vision and Principles
2. Franzen, T. et al., 2023, SRCNet Use Cases
3. Salgado, J., et al., 2023a, SRCNet Software Architecture Document
4. Salgado, J., et al,, 2023b, SRCNet Top-Level Roadmap
5. Skipper, C. et al., 2023, SRCNet Science Analysis Platform Vision
6. Wagg, J., et al., 2021, SKA1 Scientific Use Cases

Annex 6: Fast Radio Bursts & Astronomical Transients

Scientific challenge

Transient and variable celestial events provide astronomers with an important tool for probing the high-energy universe. For example, over the last decade, the detection of gravitational waves has given us a direct view of the last moments in which neutron stars and black holes spiral into each other, while multi-wavelength observations have let us watch in near real-time bursts of radiation from accreting compact objects and the implosions of massive stars.

One class of astrophysical transient that remains an enigma are fast radio bursts (FRBs). First observed in 2007, these very short duration pulses of radio emission are much more luminous than any other similar phenomena. These events are also frequent: thousands are seen per day, isotropically distributed across the whole sky. Some FRBs are seen to repeat at the same place, while others are one-time-only events.

FRBs are symptomatic of the most extreme astrophysical environments. As such, they provide a unique way to study the extremes of the Universe and to probe the material between them and the human observer. However, they are also mysterious: even after more than a decade, and despite a rapidly-growing catalogue of observed FRBs, the details of their origin is still unclear. To address this, the EuroFlash project — an ERC-funded programme upon which this use case is based[70] — will adopt a two-pronged approach to help us better understand their cause (or causes):

1. Probe the origin(s) of repeating FRBs and link them to other extreme astrophysical transients like magnetars, accreting black holes, and interacting binaries.
2. Discover FRB-like signals from astrophysical sources that are physically distinct from the sources creating the currently-known FRB population.

These goals will be addressed by a coordinated network of European radio telescopes that together function as a world-leading system for detecting and monitoring FRBs. This will include a wide-field survey capability based around the LOFAR telescope, and specialized high-resolution follow-up and monitoring systems at the Nançay telescope and the European VLBI Network (EVN).

Storage

The data storage requirements during survey operations are substantial. As discussed further below, low-latency processing is essential to ensure prompt response to transient events; however, even despite this, the combination of a high input data rate (see the Data transport section) and a need to buffer multiple hours of data to:

1. Compensate for the large dispersive delays of FRBs;[71]
2. Create integrated sky maps containing multiple hours of data;
3. Search for multi-hour periodicities

means that it is necessary to buffer 1 day's worth of data on the processing system. As described under Compute below, this amounts to a total of 8 PB storage. This storage must be capable of supporting I/O at rates of at least 500 Gbit/second.

However, it is recognized that long-term retention of the total data stream is impractical. Instead, only relatively small subsets of the data that contain scientifically significant events — perhaps a few percent of the total — will be selected for long-term storage. This data will be stored in community-standard file-based formats in the LOFAR Long Term Archive (LTA), a distributed data archive managed for and on

[70] Many thanks to EuroFlash PI Prof. Jason Hessels for permission and assistance in assembling this use case.

[71] Low-frequency radiation reaches the Earth substantially later than its high-frequency counterpart, even when emitted by the same event at the same time.

behalf of LOFAR ERIC based at SURF, Forschungszentrum Jülich and Poznan Supercomputer and Networking Centre.

Data transport

The project's most extreme data transport requirement is from the LOFAR central processor to the cluster used for EuroFlash's search capability. This will reach a data rate of 130 Gbps while survey operations are underway. Within the scope of the EuroFlash project, this processing cluster will be co-located with LOFAR central processing, so the geographical extent of data processing is limited. However, this suggests future use cases in which specialist data processing backends for telescopes could be located in other locations (taking advantage of e.g. hardware availability or local expertise) if high-bandwidth data transport is available.

In addition, data from the EVN telescopes distributed across Europe is transported to JIVE — the Joint Institute for VLBI ERIC, in the northern Netherlands — for correlation and processing. Most EVN telescopes are currently capable of sending 2 Gbit/s to JIVE, although some telescopes are limited to 1 Gbit/s. Further bandwidth is likely to be required in future.

When a new FRB or some other significant celestial event is detected, "follow-up" observations may be requested. This involves automatically triggering other telescopes (providing e.g. additional wavelength coverage or higher angular resolution) to observe the location of the event. These triggers are sent using established international standards.[72] The alert packets are relatively small (likely tens of kilobytes), but prompt and reliable delivery is critical.

Compute

The most computationally intensive part of the EuroFlash workflow is the search cluster that processes data from the LOFAR telescope. This system will comprise 32 compute nodes, each providing:

- 32 compute cores
- 2 tensor-core GPUs
- 1 TB RAM
- 256 TB storage.

This system will run an open-source software stack building on existing tools used in the LOFAR community[73] as well as new software specifically developed for the project, including a bespoke GPU-based code for image-plane de-dispersion.

Similar, but smaller, systems will be provisioned at various EVN telescopes and at Nançay. In addition, and as described above, the EVN data will be transported to JIVE for correlation on dedicated systems there using SFXC, the open-source EVN software correlator.[74]

Workflow management

As described under Compute, above, the workflows developed for this project are generally based on workflows already developed for the LOFAR telescope such as the Rapthor and LINC pipelines. These are complex, multi-step workflows for imaging radio astronomical data, described using CWL, and composed of underlying algorithmic components primarily developed in C++ and Python. These are generally distributed across multiple nodes, and typically executed in a SLURM environment.

As described under Data transport, above, prompt response to transient events is often critical to the science case. This means that low-latency processing is essential.

[72] https://www.voevent.org/; https://arxiv.org/abs/1110.0523
[73] E.g., https://rapthor.readthedocs.io, https://linc.readthedocs.io & https://wsclean.readthedocs.io/
[74] https://arxiv.org/abs/1502.00467

Access and analysis

The output from the workflows described above consist of a range of standard data products, including time series, images, and transient alerts. While alerts are distributed rapidly, other data is stored to the LOFAR Long Term Archive for access & download on demand. Analysis is then typically performed offline using standard tools. Online analysis options including Jupyter notebooks may be considered in future.

Non-technical challenges

The EuroFlash project was funded through an ERC grant which pays for staffing and core computational infrastructure. Long term storage is available through the LOFAR ERIC-provided Long Term Archive, which will soon be based on a federated authentication and authorization system (using SURF SRAM). The data has no particular commercial or personal sensitivity, but is covered by the standard LOFAR ERIC data policy.[75]

Gap analysis

The EuroFlash system is a special-purpose infrastructure funded by an ERC grant. It is illustrative of the sort of experiment that needs to be possible to enable cutting-edge radio astronomy. However, a model in which (relatively) small experiments are required to provision their infrastructure in this way may be inefficient and unsustainable: effective centralized support for systems like this could both lower the burden on science teams and provide a clearer route to sustainability beyond the term limits of project funding.

[75] https://www.lofar.eu/wp-content/uploads/2023/06/Data-policy_LOFAR-ERIC.pdf

Annex 7: ATLAS Experiment

Scientific challenge

This section should address the following points:

- **An introduction and description of the scientific background & objectives**

The primary scientific objective is to leverage data from the ATLAS detector to enhance understanding of fundamental forces, focusing on precision measurements of particles like the Higgs boson and top quarks, and exploring New Physics such as supersymmetric particles and exotic models

The High Luminosity phase of the Large Hadron Collider (HL-LHC) will significantly upgrade the capabilities of the LHC, and the ATLAS experiment will generate data at rates 7-10 times higher than current levels, targeting an average of 200 collisions per bunch crossing.

- **A short description of the scientific community served by the use case**

The ATLAS experiment serves the global high-energy physics community, specifically 6,000 physicists, engineers and software developers involved in the ATLAS experiment.

- **Scientific and technical challenges to achieve the objectives**

With a largely flat computing budget, the ATLAS experiment must meet a projected computing gap of 7-10x due to more complex data recorded at a higher frequency. This will require improvements in storage, simulation, and reconstruction, all while adapting to increasingly heterogeneous computing environments. ATLAS will continue to integrate new HPCs, some of which came online in 2022, to build upon the success of the last few years and to supplement the pledged resources. If and when the production workflows are adapted to utilise significant numbers of GPUs, suitable R&D programs will be launched in order to fully exploit large scale HPC resources that feature such technologies.

- **The timescale and scope of the use case — is this describing current work, or a vision for the future? If the latter, when?**

Projections are based on conditions for HL-LHC 2030-2045

- **Open science and open data, including commitment to FAIR[76] data or other relevant principles and policies.**

  Fully committed to FAIR data.

Storage

This section should address the following points:

- **The total data volume required to address the scientific goals.**

  1EB collected so far (15yrs), will increase to 400PB/yr in 2029.

- **The typical data volume required for an individual unit of work (e.g. job, service, or workflow).**

[76] Findable, Accessible, Interoperable, Reusable; https://www.go-fair.org/fair-principles/

1-10GB input file size, software delivery via CVMFS O(100MB), conditions via oracle. Possibility to containerize part/all in "fat" images for certain workloads.

https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PAPERS/SOFT-2022-02/

- **The data lifecycle, including retention period(s).**

  At least two copies of non-reproducible data are stored on federated storage sites at all times for the lifetime of the experiment. All formats & code used for paper analyses are to be archived. Derived data is, in-principal, fully reproducible, thus not subject to duplication policies. Datasets no longer under active analysis are transferred to tape storage.

  https://cds.cern.ch/record/2012333/files/ATL-CB-PUB-2015-002.pdf

- **Requirements for data safety, including redundancy, duplication, or archival storage technologies.**

  LTS is handled by WLCG grid sites through various tape replicas/disk RAID

- **Needs for data confidentiality, including both motivation (e.g. privacy, commercial value, embargo periods) and expected technical measures (e.g. encryption).**

  Recent experiment data is typically embargoed for a period of time, but all will eventually be publicly available. Open data is offered separately. Data needs an AAI mechanism to access.

- **Typical storage patterns, including (for example):**
  - **Size and number of files or records, directory structures, read vs. write access.**
  - **File/DB access (contiguous, striped, random, …).**

1-10GB typical file size, mostly ROOT, with HDF5 (mainly for ML applications) and a lot of small other types (txt logs, csv, etc). Mostly sequential reads, jobs may process many source files and output many result files. Parallel jobs may read the same files, but output unique files (multi-access). Tiered storage is not built into the data model, but is used for specific workflows that require high I/O or iops. Files have individual GID for each file (Rucio), metadata tracking via Atlas Metadata Interface (AMI) and other various stores. Significant Oracle database infrastructure for job conditions, as well as Hadoop and other technologies.

Data transport

This section should address the following points:

- **Typical compute job/service data throughput and access latencies, including (as applicable) both local and remote data access or transfer.**

  Latency agnostic, grid transfers typically at 100GB/s.

- **Geographical extent of data transfers — within one site, across multiple sites, multiple infrastructures, …**

  Storage & redundancy globally distributed via WLCG sites, job data normally staged at compute site (or streamed), results transferred back to grid sites

- **How the data transfer is expected to evolve with time — e.g. constant transfer rates, linear increase, occasional bursts.**

  Data growth rate will increase compared to relatively constant compute needs

- **Is streaming access to data required? If so, provide information on the expected latencies, data rates, and the stream topology (e.g. centralized, point-to-point)**
- **Data transfer technologies, protocols and tools (e.g. FTS, Globus, Unicore FTP)**

  ATLAS Distributed Computing has moved away from gFTP and dominantly uses FTS via Rucio (S3). Some jobs require XrootD or POSIX mounts. A small number may use https.

  The ARC-CE, a Grid front-end, long established in Nordic countries to enable access to HPC resources pledged as part of NDGF-T1 (a WLCG Tier-1 facility), is an edge service which has since been deployed in many HPC centres as a solution to access the more restricted centres. Specifically, the delegation of data staging from jobs to the ARC-CE works around the problem of access to data on remote grid storage. A similar edge service, Harvester, was recently developed as a common interface to US HPC sites and is now deployed at all currently-used HPCs there. Both these services are actively maintained and adapted to new environments as necessary, and thus we expect to rely on them for HPC access for the foreseeable future.

Compute

This section should address the following points:

- **Describe the general form of computing required by this use case (e.g. batch).**

Single node batch, with a small percentage of interactive analysis jobs (plotting, etc)

- **What sort of compute platforms are currently used, or could be used in future? Consider e.g. CPUs, GPUs, AI and other accelerators, Quantum or Neuromorphic computing.**

Largely x86 CPU, with ongoing development for aarch64 & heavily GPU, with portability frameworks (SYCL, etc) to avoid vendor lock-in with CUDA. Majority of the codebase is C++/python, with segments in Fortran & Java.

Simulation of the detector response to particles emerging from LHC collisions, and the subsequent digitization of this into data that mimics the detector output, is and will be the single largest consumer of CPU for ATLAS

Simulating the effect of the pile-up of multiple collisions on detector readout at the HL-LHC is a data-intensive workflow with significant I/O (~GB/s) and memory (~100GB) requirements

GPUs are not expected to play a significant role in production for ATLAS for some years. In the next few years, GPUs will mainly be employed in ML training and analysis workflows rather than large-scale production campaigns, given the timescales for core software migration described above. Later, ATLAS needs to support and integrate GPU hardware in Athena, for example by offloading computation from multiple CPU processes and threads into a shared GPU resource.

- **Provide as much information as possible about the characteristics for a single unit of work (batch job, service invocation, etc). This could include:**
  - **Complexity and estimated runtime.**
  - **Types of operation (e.g. integer, floating point).**
  - **Other specific compute requirements (e.g. small data type support, tensor computing)**

https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PAPERS/SOFT-2022-02/

The most common submission is a large number of simple 8-core jobs managed by our submission infrastructure. Full Simulation jobs are about half our resources. It is relatively low I/O (particularly low input), reasonably low memory (usually <1 GB/core for 8-core jobs), CPU-intensive, and relatively low communication. Embarrassingly parallel. It is constructed to consume about 12h per job, but we can tune

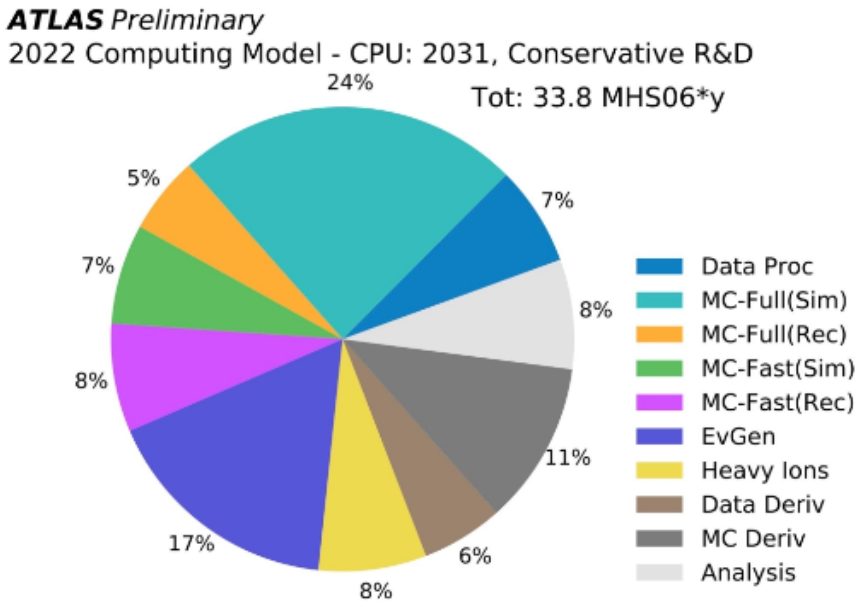
the number of events in a job to adjust the timing. The transfers do not require significant networking, but they usually come at the beginning or end of a job, and we want them to execute quickly. So the average over many jobs running asynchronously is not much higher than the requirement for a single job.

3-24H, <1GB/core, <20GB scratch, 10GB/s+ link, no latency requirements, WAN 100GB+ ideal, 2-16 nodes in multi-node Target X86, aarch64, SSE, AVX512, testing on nvidia, AMD, TPUs, FPGAs.

The fast and full simulation, trigger, and reconstruction workloads all currently run multithreaded, with the last component (pile-up digitization) currently being developed for MT. HEP data processing algorithms on GPUs can be a complicated task, as the inherent branching and memory access patterns of these types of algorithms are not well suited to GPU architectures. Individual tasks that are inherently parallel in nature, such as track seeding, event generation, or calorimeter clustering may also function well on a GPU.

- **At what rate, and in what number, are those units of work executed?**

Currently 7500 MHS23, or about ~600k cores continuously, +10-15% if local analysis jobs are included.



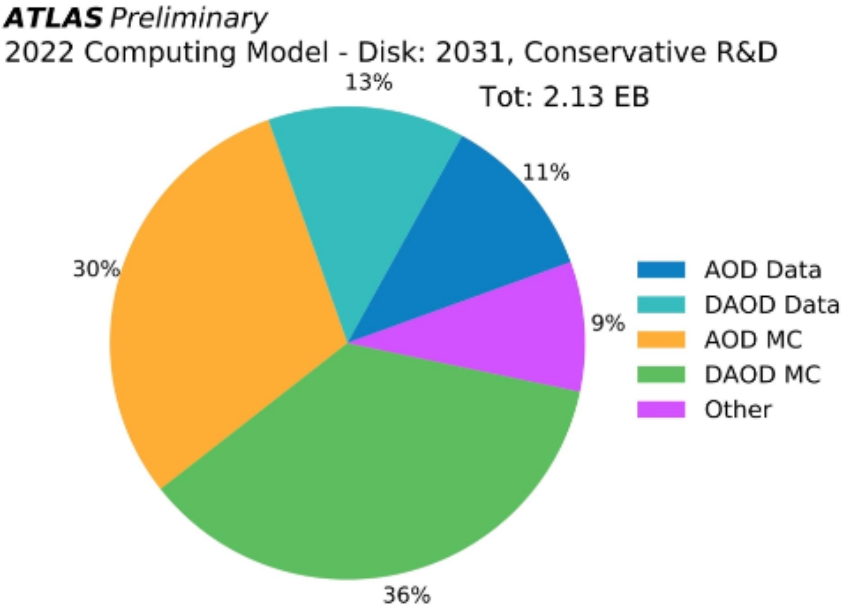


Figure A7.1: ATLAS projections for annual use of compute and storage by workload.

Table A7.1: Comparison of collision complexity, rate, event size, and quantity by run.

	Run 4 (2030)	Run 5 (2034)
Pile-up (collisions per bunch crossing)	≤140	≤200
HLT (trigger rate)	10kHz	10kHz
event size		4.6 MB/event (RAW) 50 KB/event (DAOD_PHYS) 10 KB/event (DAOD_PHYSLITE)
Recorded Events	70x10^9 /yr	70x10^9 /yr
MC events	1.4-1.8x10^10 /yr	1.4-1.8x10^10 /yr

- **Will the compute volume scale with time? If so, why (e.g. more data collected, desire to take advantage of more compute) and share thoughts on the scaling model (e.g. to multiple cores, multiple nodes, …)**

CPU/Compute needs will remain relatively constant throughout HL-LHC, but storage will continue to grow

- **Do you require specific frameworks, software stacks, or applications?**

Most workload software is delivered by mounted CVMFS volumes and conditions databases. For restricted HPC sites "fat containers" have been deployed for MC sim jobs that do not otherwise require input data. Running more complex workflows with additional payload complications or add-ons on this type of HPC, such as analysis or Monte Carlo reconstruction, would require significantly more effort and is not currently foreseen

Workloads are distributed using the Production and Distributed Analysis (PanDA) which requires an edge service (Harvester) to submit, monitor, and collect jobs on HPC centers. Conditions (workload configuration tunables) are delivered from remote OracleDB via FroNTier service (edge), or can be exported as SQL files and packaged/distributed on connectivity restricted sites.

Workflow management

Mixed, batch jobs primarily orchestrated by PanDA, but openstack, k8s, HTCondor all used at different points in the production system

https://doi.org/10.1007/s41781-024-00114-3

Annex 8: Monte Carlo Simulation at LHCb Experiment

Scientific challenge

This section should address the following points:

- An introduction and description of the scientific background & objectives

The LHCb experiment at CERN's Large Hadron Collider is a specialized detector focused on studying heavy flavor physics, particularly the decays of beauty and charm hadrons. Its primary objective is to investigate CP violation and rare decays, which could provide insights into the matter-antimatter asymmetry observed in the universe.

While b-physics remains its core mission, LHCb's research program has expanded to encompass a broad range of topics, including exotic hadron spectroscopy, electroweak physics, and quantum chromodynamics in the forward region. The experiment's unique forward geometry allows it to probe particle interactions in a kinematic range complementary to other LHC experiments.

- **A short description of the scientific community served by the use case**

The LHCb collaboration comprises over 1800 participants from 107 institutions across 26 countries (as of November 2024). This international team is responsible for the detector's construction, operation, and data analysis.

- **Scientific and technical challenges to achieve the objectives**

Processing and selecting events from proton-proton collisions at very high rates (30 MHz). Handling and analyzing large volumes of data (100s of PB per year in the future).

- **The timescale and scope of the use case — is this describing current work, or a vision for the future? If the latter, when?**

Now (LHC Run 3) until the end of LHC Run 5, ~ end of 2041. Requirements will evolve during this time period.

- **Open science and open data, including commitment to FAIR[77] data or other relevant principles and policies.**

LHCb, like all of the four large experiments at the LHC has embraced CERN's open data policy, with the aim to start releasing data from a run within five years of the conclusion of that run. Data may be withheld by an experiment if there are active analyses ongoing. Full datasets will be made available at the close of the collaboration.

Partially - Software released under GPL3.

Storage

This section should address the following points:

- **The total data volume required to address the scientific goals.**

[77] Findable, Accessible, Interoperable, Reusable; https://www.go-fair.org/fair-principles/

In run 3: data flow of 60 PB/year to tape, 15 PB/year to disk, ~10 TB/year for user analysis. [LHCB-TDR-018]. Considering backups, this becomes O(100) PB per year to tape and O(50) PB per year to disk [LHCB-TDR-023]. Data is stored at 1 Tier 0, 7 Tier 1 and 14 Tier 2-D WLCG sites.

Total 1200 PB for run4 (~2033). raw data: 100PB per data-taking year; physics analysis data: 35PB per data-taking year; simulation: a few PB per year. The number does not include the Run5 upgrade requirements (~7.5X).

- **The typical data volume required for an individual unit of work (e.g. job, service, or workflow).**

datasets generally 10GB-100GB, jobs may read one or many files sequentially.

- **The data lifecycle, including retention period(s).**

Data is replicated via WLCG storage sites, LTS replication via tape.

- **Requirements for data safety, including redundancy, duplication, or archival storage technologies.**

Multiple replicas on disk/tape, needs AAI mechanism to access (federated).

- **Needs for data confidentiality, including both motivation (e.g. privacy, commercial value, embargo periods) and expected technical measures (e.g. encryption).**

embargo period, some sets released via opendata.cern.ch

- **Typical storage patterns, including (for example):**
    - **Size and number of files or records, directory structures, read vs. write access.**
    - **File/DB access (contiguous, striped, random, …).**

Mainly cluster filesystems (CVMFS/EOS/etc) with some relational DB for metadata, catalogs, etc. Files are ROOT format + custom LHCb RAW format. Datasets not self-describing; require

Data transport

This section should address the following points:

- **Typical compute job/service data throughput and access latencies, including (as applicable) both local and remote data access or transfer.**

<10MB/s network/core locally, <1MB/s remote connectivity. Preferred outgoing connectivity to limited subnets otherwise edge service (ARC CE) for job pilots

- **Geographical extent of data transfers — within one site, across multiple sites, multiple infrastructures, …**

Batching typically within one site, bulk transfers of datasets & job output to grid storage sites (WLCG)

- **How the data transfer is expected to evolve with time — e.g. constant transfer rates, linear increase, occasional bursts.**

Data volumes will scale linearly until the mid-2030s. An upgrade is expected in Run5 (mid-2030s), with an increase of the data volumes of roughly a factor 7.5.

- Is streaming access to data required? If so, provide information on the expected latencies, data rates, and the stream topology (e.g. centralized, point-to-point)
- **Data transfer technologies, protocols and tools (e.g. FTS, Globus, Unicore FTP)**

FTS3, DIP, SRM(gfal, gfal2), DFC (DIRAC File Catalog), WebDAV, XrootD, POSIX mounts

Compute

This section should address the following points:

- **Describe the general form of computing required by this use case (e.g. batch).**

Dominantly batch, made up of MC Simulations (90% of total CPU use) and offline processing (calibration, record, etc). Small percentage analysis. For HPC we plan to only run MC simulations (no input datasets).

- **What sort of compute platforms are currently used, or could be used in future? Consider e.g. CPUs, GPUs, AI and other accelerators, Quantum or Neuromorphic computing.**

Currently dominated by single-node CPU batch jobs. aarch64 is only available for a small subset of applications.

We are exploiting ways to speed-up simulation with e.g. ML techniques or by using accelerators (GPUs). We foresee an increased usage of ML/AI algorithms and methods for physics analysis, but we cannot quantify the need at this time.

- **Provide as much information as possible about the characteristics for a single unit of work (batch job, service invocation, etc). This could include:**
  - **Complexity and estimated runtime.**
  - **Types of operation (e.g. integer, floating point).**
  - **Other specific compute requirements (e.g. small data type support, tensor computing)**

The workflow using most processing resources (nearly 90% of CPU work) is MonteCarlo simulation, consisting of several steps: event generation with e.g. Pythia and EvtGen or other generators; detector simulation through Geant4 or fast simulation techniques; event reconstruction and selection, through a specific application. The detector simulation is the dominant contribution in terms of CPU usage.

During average job: Duration <1Day, <2GB mem/core, <20GB scratch/core C++/Python

aarch64 (NEON) ,x86 (AVX/2/512)

- **At what rate, and in what number, are those units of work executed?**

1e9

- **Will the compute volume scale with time? If so, why (e.g. more data collected, desire to take advantage of more compute) and share thoughts on the scaling model (e.g. to multiple cores, multiple nodes, …)**

- **Do you require specific frameworks, software stacks, or applications?**

DIRAC WMS (github.com/DIRACGrid), software preferred delivery from CVMFS, Apptainer otherwise. Dependent on LCG releases (https://ep-dep-sft.web.cern.ch/document/lcg-releases)

References:

1. https://cds.cern.ch/record/2776420 LHCb Phase II TDR, 2023
2. https://cds.cern.ch/record/2886764 LHCb DAQ TDR, 2024
3. https://cds.cern.ch/record/2319756 LHCb Computing Model, 2023
4. https://cds.cern.ch/record/2310827 LHCb Computing TDR, 2023
5. https://cds.cern.ch/record/2717938/ LHCb GPU HLT TDR, 2023

Annex 9: ALICE Experiment

Scientific challenge

ALICE (A Large Ion Collider Experiment) is one of the four major experiments operated at the CERN LHC. Its detector is targeted towards the study of heavy-ion physics. At a temperature around 2000 billion degrees (about one hundred thousand times the temperature in the core of the Sun), the protons and neutrons, the building blocks of the atomic nuclei, melt into a plasma of elementary particles which has been dubbed the Quark-Gluon Plasma (gluons being the carriers of strong interaction, that holds the nuclei together). According to the Big Bang cosmological model, such temperatures correspond to the state of the early Universe during the first few millionths of a second: the Quark Gluon Plasma thus represents the primordial state from which all matter evolved in our present Universe. Similar conditions are created, albeit for a fleeting instant, in collisions of heavy-ions at the unprecedented energies offered by the LHC. This opens the unique opportunity to create and study in the laboratory tiny droplets of this primordial matter.

The ALICE experiment is a large international collaboration with over 2000 scientists and engineers from around the world. It has been collecting data since 2010 and will continue to do so during the LHC's upcoming runs.

ALICE is committed to open data and follows the FAIR principles for scientific data management, making its findings available to the public for further research.

Storage

The ALICE experiment currently operates approximately 150PB of disk storage and 250 PB of custodial storage, predominantly tapes. The main computational workflows are as follows:

- Monte Carlo simulation - CPU-intensive payload with minimal (up to few MB) input data per unit workflow and medium (up to 5GB) output data. The payload duration is of the order of a few hours and runs on 8 CPU cores. The fraction of the MC simulation is ~40% of the total distributed computing resources of ALICE and runs on any Grid resources, including HPCs.
- RAW data reconstruction - CPU and I/O intensive payload with large (order of 100GB) of input data per unit workflow and medium (up to 10GB) output data. The payload duration is of the order of a few hours and runs on various numbers of CPU cores (8-64), combined with accelerators (GPU). The fraction of the RAW data processing is ~30% of the total distributed computing resources of ALICE and it targets specific computing centres (T0 and T1s). One of the HPCs accessible to ALICE (Nurion@KISTI T1) is used for this workflow.
- Organized analysis - I/O intensive payload with large (up to 100GB) input data per unit workflow and small (up to 1GB) output data. The payload duration depends strongly on the input data size and varies between 20 min to several hours and typically runs on 2 CPU cores. The fraction of the Organized analysis is ~30% of the total distributed computing resources of ALICE and runs on any Grid resources, including HPCs.

The data retention periods are many years to decades for the RAW and majority of the secondary outputs from the MC simulations and RAW data reconstruction. The output of the Organized analysis is kept for the duration of the analysis cycle, up to the publication of the physics results. Some of the data is moved from disk to custodial storage after the active processing period is completed.

The data safety is assured by keeping 2 copies on two independent and non-collocated storage elements and the ALICE Grid middleware periodically sweeps all disk-located data for integrity checks and replicates lost or corrupted data. The data stored on custodial storage is considered safe and is not subject to these checks.

The data security is assured by the use of tokens created for each individual operation - read/write/delete - and controlled by ACLs stored for every file in the ALICE Grid catalogue. To prevent accidents, the custodial and the disk storage elements containing RAW data are additionally protected and removal of the data is not possible, except through special tools and after multiple validations, executed by multiple experts.

The current content of the ALICE storage is 14 billion files of various sizes from few KB to 10GB with a growth rate of 10% per year. The read to write data ratio is approximately 8/1, with random access from disk storage and all data is written and read using the XrootD protocol and tools. The custodial storage access is strictly regulated and centrally managed. It also uses XrootD for data access.

Data transport

The ALICE Grid middleware, JAliEn, employs late-binding techniques for all payload types through pilot jobs. Data locality, meaning payloads execute where the data resides, is a crucial matching parameter for sites and data-intensive workloads such as Raw reconstruction and Organized analysis. This approach minimizes access latency by predominantly utilizing local area network (LAN) connections within the computing center for data access. While payloads primarily access data locally, they are permitted to reach over the wide area network (WAN) to read configuration files and conditions data, which are stored in a limited number of geographically distributed locations to optimize access times and network load. Additionally, payloads can access data over the WAN from remote storage containing secondary copies in cases where locally stored data is temporarily inaccessible due to missing files or storage overload. The total WAN traffic resulting from these exceptions is approximately 5% of the overall data traffic generated by ALICE payloads.

Since payloads directly access data from storage in a point-to-point manner, all ALICE storage elements require two ports (1094/TCP, 1095/TCP) to be open to the world for incoming access, as dictated by the XrootD protocol and the XrootD-enabled storage solutions. Similarly, all computing center worker nodes, including HPCs must have these ports open for outgoing access. This constitutes a strong requirement for site infrastructure, including HPC facilities. Given that the majority of Worldwide LHC Computing Grid (WLCG) sites are interconnected through the LHCONE overlay network, this traffic is deemed secure and does not necessitate passing through site firewalls.

Data access from payloads can occur through both file download followed by processing or through streaming during processing. The choice between these methods is determined by the input data size and the availability of local cache. Streaming is generally preferred as the majority of data volume is accessed over the LAN, minimizing latency concerns.

Beyond payload access, ALICE utilizes JAliEn's built-in tools to execute data transfers between storage elements. These tools also rely on XrootD, specifically the xrd3cp third-party copy implementation, which facilitates server-to-server data transfer. Typical data copy patterns for this use case are infrequent, with the largest data volumes associated with the replication of raw data from the T0 center to T1 centers on custodial storage.

ALICE disk storage elements exclusively utilize three storage technologies: EOS, XrootD, and dCache, while custodial storage elements employ EOS, CTA, dCache. All these storage systems are XrootD-enabled.

Software delivery to all computing centers, including HPC facilities, is exclusively handled through CVMFS. CVMFS is fully integrated into the three HPC facilities accessible to the ALICE experiment.

Compute

The ALICE distributed computing system, JAliEn, is adaptable to various payload submission methods to local resources, such as through prevalent WLCG Compute Element gateways like HTCondor and ARC, or directly to local batch systems, common in HPC environments. JAliEn includes modules for PBSpro and SLURM, catering to the HPCs accessible to ALICE. Other modules can be incorporated as needed.

Access to local resources typically occurs through a point-of-presence host at the site, the VO-box. This host runs ALICE-specific persistent services, including a monitoring module and a JobAgent submission module to the CE gateway or batch system. The VO-box, usually within the computing center's network segment, receives monitoring and message traffic directly from the worker nodes, aggregates it, and transmits it to the ALICE Grid central services at CERN.

The majority of resources available to ALICE are equipped with x86 architecture CPUs, while a fraction also includes ARM CPUs and GPU-equipped clusters, all fully integrated into the Grid system. The ALICE CI system builds software for all existing platforms and extensions, enabling transparent resource utilization based on payload requirements. One of ALICE's HPCs is equipped with NVIDIA GPUs, actively utilized by ML-enabled workflows. The total available CPU resources for ALICE are approximately 250k cores, with about 10% provided by HPCs, currently considered opportunistic due to lack of government funding agency pledges.

The three major workflows executed on the distributed computing resources are described in the 'Storage' section. The ALICE Grid executes approximately 1 million payloads per day, translating to a frequency of 11.5Hz. This number scales linearly with the growth of CPU and storage resources, averaging 15% annually.

A significant development in recent years has focused on payload isolation and control. Currently, all ALICE payloads execute in a containerized environment using standard tools like Apptainer/Singularity. The ability to start and run containers is a crucial requirement for computing sites, including HPCs. For the latter, container execution methods are adapted to each specific case.

Workflow management

The ALICE workflows are usually constructed to be self-containing, i.e. a major unit of work is executed within each workflow resulting in output suitable for another independent workflow. If there is a workflow dependency, these are steered by the JAliEn Production Management system and are dependent on the data location of the output of the preceding workflow, using the standard JAliEn payload matching algorithms.

The workflow uses JobDescriptionLanguage (JDL) with extensions when submitted to JAliEn, however the sites do not have to interpret these. The local workflow is steered by the JobAgent, already present on the WN and submitted usually through a shell script, containing the necessary environment variables to adapt to the local software environment. The executable code of the payload, as well as the additional libraries are taken from CVMFS and the JobAgent spawns the container for the payload and executes the necessary script to launch it.

The JobAgent is equipped with the necessary probes and messaging tools to transmit a number of parameters, for example CPU, memory and disk utilization of the workflow, to the ALICE monitoring system MonALISA, which in turn provides these to JAliEn for accounting and decision taking purposes. There is no difference in the workflow execution and monitoring mechanisms between standard and HPC facilities.

Access and analysis

The JAliEn Grid middleware manages all computational and storage resources within ALICE. Individual users or production system management tools submit payloads of any type through the JAliEn interface, written in Python and Java. JAliEn analyzes the payload requirements, stores them in a shared task queue, and then schedules them for execution based on integrated global quotas, priorities, and available resources. Users never directly interact with the underlying computing resources at the sites.

In addition to the command-line interface, a web interface (alimonitor.cern.ch) offers a unified view of the ALICE Grid alongside a dedicated portal for user job monitoring, submission, and control.

The JAliEn Python interface, along with the TGrid class and TJALiEn plugin, are all available within ROOT. Users on their workstations leverage these tools to interact with JAliEn and analyze the results of their executed payloads. The amount of data transmitted from the Grid storage to the user typically ranges from 100MB to a few GB.

Non-technical challenges

The ALICE Grid leverages WLCG computing resources provided by computing centers worldwide. These centers, supported by local funding agencies, pledge resources to ALICE annually through a WLCG-established requirement and approval process.

All members of the ALICE collaboration can access these resources. However, for efficient utilization, several centralized systems manage payload submission and control on behalf of users. While individual users have access with specific quotas, they often prefer to utilize centralized systems, particularly organized data analysis platforms, due to the complexity of data processing chains and the large amount of resources required to analyze the data samples.

User access to the Grid is based on the x509 standard for authorization and authentication, integrated into JAliEn.

Centralized training for Grid use is done at thematic tutorial sessions conducted every few months. Comprehensive documentation on software and procedures is also available on the ALICE offline web pages.

The ALICE data and analysis results are non-proprietary and do not contain any personal information, therefore not subject to GDPR regulations.

Gap analysis

The success of the ALICE experiment hinges on the size and efficient utilization of the available distributed computing resources. Integrating new resources, particularly HPCs, is a top priority due to their substantial processing power. However, the unique characteristics of each HPC system pose significant challenges, requiring individual approaches and considerable development and support efforts. This high adoption threshold must be balanced against the potential benefits for the experiment.

Facilitating HPC adoption requires several key elements:

1. A common and standardized access mechanism, analogous to existing Grid gateways.
2. Standardized authorization and authentication tools and protocols.
3. Sufficient external network capabilities to enable access to remote storage and services.
4. Common methods for software delivery, such as CVMFS.
5. The ability to utilize modern techniques for software isolation and control, such as containers.

While some work is always necessary for full integration, most accessible ALICE HPC facilities provide some or all of these elements.

References:

1. https://cds.cern.ch/record/2011297/files/ALICE-TDR-019.pdf ALICE O2 TDR 2019
2. https://arxiv.org/pdf/2412.13755 ALICE HPC O2 production for HPC

Annex 10: AI-based particle flow reconstruction for LHC particle detectors

Scientific challenge

The Particle Flow (PF) algorithm in CMS was introduced in Run 1 and lies at the core of reconstruction, significantly improving the resolution for jets and missing transverse energy. Fundamentally, the algorithm combines information from all sub-detectors such as the tracker and the different calorimeter layers to increase the resolution of the reconstructed particle candidates.

The MLPF approach formulates PF reconstruction for the full event as a supervised learning problem, and can recover, and in some cases exceed, the performance of the heuristic algorithms used in the offline software. First tests at integrating the prototype algorithm to the CMS offline reconstruction software have been carried out, validating the viability of integrating this approach in a realistic reconstruction setup.

A key aspect of the MLPF approach has been to focus on scalable and computationally efficient machine learning methods, demonstrating the possibilities of avoiding quadratic scaling in ML-based full event reconstruction.

An important advantage of ML-based reconstruction approaches such as MLPF is that the algorithm can easily be retrained for new detector conditions or configurations, or even completely new detectors. This could be particularly valuable for quickly mapping out the physics reach of new detectors under study for the Future Circular Collider (FCC), or for other potential future high energy physics (HEP) experiments.

Furthermore, deep learning (DL)-based approaches are typically implemented through operations that are highly suitable for parallelization on heterogeneous computing architectures. Recently, there has been an extensive development of specialised computing platforms (neural network accelerators) for both deep learning training and inference, with the aim of increasing throughput while reducing the energy footprint of the calculations. This makes efficient DL-based methods a potentially powerful alternative to hand-written heuristic algorithms, which may need to be significantly rewritten for each new computing paradigm and detector configuration.

Storage

The data pipeline consists of full simulation of the detector, followed by the extraction of relevant quantities for supervised ML. The raw datasets consist of tens of TBs of simulated low-level detector events, while the processed events suitable for supervised machine learning are currently up to a TB. The dataset sizes are expected to increase in the future, as we have observed scaling laws that result in improved model performance with additional data. Each training typically involves 100-1000 GB of training data, depending on event complexity, detector configuration and the number of collision events. The processed data suitable for supervised machine learning is often split into files of a size that is chosen by the user creating the dataset, most commonly on the order of hundreds of MBs. Data should be retained long-term (5+ years) for reproducibility and comparative analyses. Frequent access is needed for current data, with archival storage for historical datasets.

The data confidentiality rules are different depending on the source of the datasets. The CLIC-based datasets used in [cite] are completely open and can be freely shared. The CMS-based datasets, however, follow the rules of closed data from the CMS experiment until the collaboration has deemed the data open.

Data transport

Currently, the datasets are generated and processed to the ML-suitable format at a single Tier2 computing site. From there, the TB-scale ML datasets are transferred to HPC sites around the world for ML training, and to CERN EOS for medium-term archival. Transfers are required across multiple sites, both within CERN and other international facilities. Essentially, the dataset needs to be stored at the compute site where the

training is taking place, or be streamed there with sufficient bandwidth to avoid data loading into compute nodes becoming a bottleneck. We expect that the data processing model will evolve to encompass distributed production of simulated data, with ML datasets stored centrally at two separate storage sites and cached locally at HPC sites during training time.

Compute

Training

This use case relies on batch processing across multiple GPUs and, where possible, AI accelerators. The workload is well-suited for heterogeneous computing architectures, taking advantage of CPUs, GPUs, and AI accelerators.

Training to the point of convergence and to reach state-of-the-art results in MLPF in a reasonable amount of time requires access to multiple GPUs, preferably 4 to 8 modern HPC GPUs in a single machine. Distributed training is typically done in a data-parallel manner. While the NN model weights and gradients fit comfortably in GPU memory on a single modern GPU, multi-node training is also possible. The latest public results [cite] consumed roughly 30 node-hours per training using compute nodes equipped with 4 NVIDIA A100 GPUs for the GNN-based model. The resources required for training are highly dependent on the model architecture, training parameters and dataset size and complexity. The dataset in [cite] used a CLIC-based detector model and simulated electron-positron collision events. These events are inherently less complex, and therefore easier to reconstruct, than proton-proton collision events that are recorded by detectors such as ATLAS and CMS. Training models to convergence on datasets based on simulated proton-proton collisions and complex highly granular detector geometries may therefore require significantly more computational resources. The number of model trainings that need to be carried out in a given time unit varies across the development cycle. During periods of intense model and/or training data development, dozens of training sessions per week may be required. These do not however necessarily have to be trained until convergence.

Furthermore, hyperparameter optimization (HPO) is an essential part of the optimization workflow to achieve the best possible model performance in terms of event reconstruction quality. HPO requires the training of large numbers of model trials and although there are many techniques to reduce the compute needs of this highly demanding task, it is inevitably a compute resource intensive effort. In [cite], the authors used Bayesian Optimization in combination with the ASHA algorithm to optimize hyperparameters. The optimization was distributed across 96 NVIDIA A100 GPUs spread over 24 compute nodes, consuming roughly 12 '000 GPU hours. A full-scale HPO run show ideally be carried out after each significant development in model architecture or training dataset. Annual compute needs are difficult to predict since they depend on the number of developers that allocate time for this use case in the future but a rough estimate would be around 50k-100k GPU hours per year in the near future. In the mid to longer term future, it is easy to imagine the compute needs growing by orders of magnitude.

Standard ML frameworks (TensorFlow, PyTorch) are critical, along with the appropriate drivers and libraries to run these frameworks on accelerated hardware.

Workflow management

The workflow involves data generation (including physics and detector simulation), data preprocessing, model training, validation, and event reconstruction. Each step may span multiple compute nodes within a single site. As different steps of the workflow require potentially different software stacks and storage access patterns, the workflow is currently not set up in an end-to-end fashion using a single description language or a single execution environment, but rather relies on job-specific and site-specific configuration. Porting the workflow to a common description and a unified execution environment would be of high interest and impact to ensure the sustainability and portability of the project.

Access and analysis

Most jobs are submitted via a batch processing system such as SLURM. JupyterLab or Jupyter notebooks are used for interactive analysis and visualization, either directly through a JupyterLab instance hosted at the HPC site or locally after having transferred results via ssh. If transferred, somewhere between 1 and 100 GB might need to be transferred, depending on the size of the testing datasets and whether the validation plots were already created remotely or not. If validation plots were already created remotely, only the plots would need to be transferred. All validation plots normally sum to less than 100 MB per model.

Non-technical challenges

Currently, each contributor has been using whatever resources available to them individually either at a local cluster to their home institute or remotely via project collaborations. Some have accessed HPC centres such as LUMI-G via EuroHPC. There are between 1 and 10 researchers working on the project that need access to GPUs for development, training and inference.

Gap analysis

Expanded access to GPU and AI accelerator resources would significantly accelerate training and inference, and allow studies of larger datasets, larger models and more extensive hyperparameter optimization.

Annex 11: LLMs for the CERN accelerator complex

Scientific challenge

AccGPT is an innovative pilot project utilizing Large Language Models (LLMs) to create a chatbot for interacting with CERN's extensive internal knowledge base. This initiative is primarily led by the CERN Beams and IT departments, while the objective is to make this chatbot available to the entire CERN community. AccGPT is designed to provide quick and straightforward answers to queries about CERN specific and internal knowledge, similar to ChatGPT, thereby enhancing productivity and decreasing the time experts spend on support tasks. Currently, efforts are focused on enhancing the accuracy of the chatbot and expanding its knowledge base. Looking ahead, there are plans to expand AccGPT's functionalities, for example the introduction of coding assistance features. LLM agents can also be used as personalised and specialised teachers, helping newcomers learn the many complex tools and software frameworks used at CERN. LLM agents will be able to improve knowledge finding, user support, streamline development processes, and enhance onboarding experiences. Furthermore, LLM agents could be deployed in the CERN machine control rooms to allow easier interaction between operators and the machines.

AccGPT is part of a broader vision to embed LLMs within CERN's workflows, improving productivity and reducing the time required for routine queries. AccGPT's objectives include:

- **Knowledge Retrieval**: Enabling easy access to CERN's complex information architecture.
- **User Support**: Assisting users with technical and operational questions related to the CERN accelerator complex.
- **Educational Aid**: Serving as an interactive tutor for newcomers, guiding them through CERN-specific tools and frameworks.

The CERN community includes scientists, technical staff, accelerator operators, and software developers who rely on immediate, precise information to operate and maintain the accelerator complex. By reducing the demand for expert intervention on routine queries, AccGPT also helps experts focus on high-value tasks.

Challenges:

1. **Accuracy and Contextual Relevance**: Developing an LLM capable of understanding and generating responses specific to CERN's specialized knowledge domains.
2. **Data Privacy and Security**: Ensuring that sensitive information within CERN's knowledge base is handled securely and meets data protection regulations.
3. **Scalability**: Expanding the LLM's capabilities to handle diverse queries, increasing accuracy as its user base grows.

This use case is currently in active development, with initial small-scale deployment ongoing. Mid- to long-term plans include extending functionality to more dynamic applications, such as live coding assistance and operational support in control rooms.

Storage

The data contains sensitive information. The size of the dataset is currently not a significant factor, being less than 10GB of text. This will increase in the future, but will most likely not explode in the near to mid-term future. However, storing multiple dataset versions could multiply this number by one or two orders of magnitude. If multi-modal models are explored in the future to enable image or audio capabilities, including for example the creation or interpretation of plots, images and audio recordings, might increase the required data storage significantly since data modes such as images, video and audio require more storage space than text.

Ongoing access to and periodic updates of the datasets are required as the knowledge base evolves. Given the sensitive nature of CERN's internal information, strict confidentiality measures need to be in place.

During inference, most data access is read-heavy, with structured, text-based datasets supporting rapid retrieval of information via RAG (Retrieval Augmented Generation). As the LLM is refined, additional versions may increase storage load, though files remain compact.

Apart from storing datasets, the models used for deployment and fine-tuning have to be stored in some kind of models registry. Depending on the model size, this can occupy hundreds of GB storage per model. During fine-tuning, multiple model checkpoints might be saved per training, thus resulting in the storage needs for model checkpoints quickly reaching the level of TBs.

Data transport

Currently, data access primarily occurs locally within CERN's infrastructure. Expected throughput remains modest, given the text-based dataset and efficient LLM compression. Transfers are mostly within CERN sites, with no significant cross-site transfer needs anticipated. As the LLM is scaled, data access patterns may experience occasional bursts but are unlikely to require extensive, continuous transfer capabilities. Current data transport relies on CERN's internal network, which supports fast, secure transfers and minimizes latency for interactive user queries.

To enable the use of HPC site for model deployment (inference), there needs to be a secure way of transferring user prompts received from e.g. a web interface outside the HPC center, execute the generation step on accelerated hardware inside the HPC center, and then safely transfer the response back to the user. For RAG to work, the RAG database needs to either be stored at the HPC site, or the relevant parts of it needs to be streamed there in real time when a user submits a prompt. In the case of streaming, the latency needs to be sufficiently fast not to degrade the user experience too much. Ideally the user should not have to wait more than a few seconds before receiving a response.

Compute

Inference/Deployment

The current focus is on GPU-intensive inference, leveraging LLMs ranging from 8B to 405B parameters, primarily using LLaMA models. While compute needs will grow as models and usage scale, the emphasis is on deploying multiple specialized small-scale models rather than a single, large, general-purpose model. However, "small-scale" remains loosely defined. Scaling involves multi-node clusters to meet increasing user demand and computational load.

Minimum reqs to start:

Table A11.1: Minimum Requirements by model type.

Model	Inference GPU vRAM per Instance[78]	GPUs for 3 Parallel Instances
LLama 8B (F16)	~25GB (1x A100)	3x A100 40GB
LLama 70B (F16)	~145GB (4x A100/ 2x H100)	6xH100 80GB
LLama 405B (Int8)	~410GB (5x H100)	15xH100 80GB

[78] Additional vRAM might be required for longer context lengths for which some is already taken into account.

User Handling per Instance:

Assuming an 8-hour workday and an average of 20 requests per user:

- **Input Tokens (including system/user prompts & RAG knowledge):** ~3,000 tokens
- **Output Tokens:** ~300 tokens

Approximate inference times for generating 300 tokens:

Table A11.2: Inference time by model type.

Model - GPU	Inference Time per Request
LLama 8B F16 - A100 (PCIe)	~2,5s
LLama 70B F16 - 4xH100	~2,5s

$$Requests\ per\ Instance = \frac{Seconds\ per\ Workday}{Inference\ Time\ per\ Request} = \frac{28{,}000\ seconds}{2.5\ seconds/request} = 11,520\ requests$$

$$Number\ of\ Users = \frac{Requests\ per\ Instance}{Average\ Request\ per\ User} = \frac{11{,}520\ requests}{20\ requests/user} = 576\ users$$

→ **Users per 3 Instances:** $3 * 576\ users = 1,728\ users$

Peak GPU usage occurs during work hours, with near-idle periods outside those hours.

Development Requirements:
A development cluster with 4x H100 GPUs may suffice for a single developer. However, additional resources could be needed depending on use-case complexity (influenced by model size, inference/fine-tuning/HPO) and developer count.

Training

Currently, no training for AccGPT occurs due to hardware limitations. While full LLM training might not be necessary if performant open-source models (e.g., LLaMA) remain accessible, fine-tuning of pre-trained models is desirable. Hardware needs depend on model size and training strategy (e.g., full fine-tuning, LoRA, qLoRA).

Fine-Tuning Requirements:

- **Minimum Cluster for Small-Scale Fine-Tuning:** 4x H100 GPUs.
- **Scaling:** Greater parallelization would accelerate training.

Idle deployment hardware during non-working hours could be repurposed for fine-tuning, optimizing hardware utilization.

Workflow management

AccGPT does not involve complex, multi-step workflows. Primary tasks are single-step inference requests, with straightforward data retrieval. While responses should be prompt, the workflow is not highly time-sensitive. Kubernetes may be considered for orchestration to manage multiple instances of the LLM, ensuring load balancing and scaling.

For certain use cases such as control room operations, time sensitivity is of greater importance. Responses need to be fast and might require access to real-time machine parameters. Such a use case would most likely require on-site compute for deployment of the model.

Access and analysis

User access mode is primarily interactive, with user queries handled in real-time via a web interface. Currently, data is processed locally on CERN infrastructure; no external data transfer is required for analysis. Interactive access via Jupyter notebooks, shell access, and web portals is required for developers.

Non-technical challenges

Access to compute and storage resources has so far been managed through CERN's internal IT systems but applications to EuroHPC supercomputers are in the pipeline. To use the chatbot, authorized users access the system via CERN's authentication protocols.

Gap analysis

- Considering all of the aspects of your use case described above, what are the biggest missing pieces? What additional services or capabilities could make you most productive?

Key gaps and improvement areas for this use case include:

1. **Increased Compute Resources for deployment**
2. **Increased Compute Resources for fine-tuning:** More high-performance GPUs would facilitate fine-tuning and support additional use cases.
3. **Data Confidentiality**: Given the sensitive nature of CERN data, controlled access is essential.
4. **Scalability Solutions**: As LLM use expands, dynamic scaling options for compute and storage will be essential.
5. **Automation in Workflow Management**: Integrating workflow automation tools would streamline model updates and deployments, enhancing overall efficiency and reducing developer workload.

References:

1. https://indico.cern.ch/event/1423858

Annex 12: LIGATE MD

Scientific challenge

The European **LIGATE project** (Leveraging Innovative Technologies for Exascale Computing in Genomics), launched in 2021, aimed to advance precision medicine and genomics through high-performance computing (HPC) and exascale technologies. Specifically, the project sought to improve and scale molecular docking simulations for drug discovery, utilizing cutting-edge computing resources to achieve faster, more efficient, and more accurate results.

Key Objectives:

1. **Exascale Computing Integration**:
    - Adapt molecular docking workflows for exascale systems, which are capable of performing at least 10^{18} calculations per second.
2. **Faster Drug Discovery**:
    - Accelerate the virtual screening of potential drug compounds by utilizing highly efficient computational techniques.
3. **High-Performance Software Development**:
    - Develop scalable and optimized software for HPC infrastructures, integrating innovative algorithms and frameworks.
4. **Open Collaboration**:
    - Foster collaboration among researchers, institutions, and industries across Europe to pool resources and expertise.
5. **Precision Medicine**:
    - Provide tools and methodologies to improve the design and personalization of therapeutic approaches.

Real-World Impacts:

The project had the potential to revolutionize drug discovery by significantly reducing the time and computational cost of identifying viable drug candidates. Its integration with exascale systems aimed to provide researchers with tools capable of handling the increasing complexity of biomedical challenges.

Funded by the European Union's Horizon 2020 program, LIGATE combined expertise in life sciences, computer science, and engineering to bridge the gap between computational biology and practical medical applications.

The project finished in mid-2024 but resources from LIGATE, such as the data from the molecular dynamics simulations including simulation scripts are freely available.

Storage

The molecular dynamics data formed the bulk of the output from the project, comprising nearly 40Tb of disk space. Each simulation job generated on average 1.0-1.5Gb of space but the post-processing workflow would generate another amount with similar dimensions. The number of inputs simulated amounted to close to 4000, but for redundancy and improved statistics, 4 replicas of each input were simulated to give a total

dataset of about 16000 simulations. Since the project guaranteed open data, no attempts were made to encrypt or restrict the data in any way. The data are stored in a data archive at CINECA and will be held until mid-2025. Before then it is hoped to re-process and transfer all the data to the MDDB portal for molecular dynamics trajectories (https://mddbr.eu/).

In the data archive the data are stored in directories containing only a few files: the molecular dynamics trajectory which is the largest, and other files such as the topologies which describe the molecular data and how they are represented in the trajectory. The latter is in compressed binary format and stores data points (the atomic configuration at each time interval) in a contiguous way.

Data transport

Given that the molecular dynamics (MD) simulations can take many days of constant execution but generate only a few Gb of data, the IO throughput is very small. During the simulation the data are stored locally and not usually moved to remote locations (except for the MDDB transfer as described above). Data streaming is not usually required.

Compute

The computing jobs are done in batch, often within chains of batch jobs linked by dependencies in so far as the simulations may require many days of computer time. Currently, the most efficient architecture for biomolecular simulation is GPUs since virtually the entire calculation engine can reside on the accelerator, with minimal interaction with the CPU. In fact, for most workflows only 1 GPU is required for the simulation, regardless of the simulation input. There is considerable interest in using AI for replacing the physics-based calculations at the heart of the MD engine, but so far only relatively small molecules can be studied – much smaller than proteins or membranes which typically form the use cases for biomolecular simulation. The use of quantum computers for studying biomolecular systems is also being studied by many researchers but so far the studies are at an early stage.

The calculations at the heart of an MD are fairly standard floating point type operations, but to ensure stability, particularly in the finite difference algorithm, the operations need to be single precision (32 bits).

The scaling model uses the domain decomposition method which explains why biomolecular simulations do not scale beyond 1 of the latest GPU models – the input system (e.g. a protein) has a fixed size and cannot be arbitrarily scaled to exploit additional resources. Increases in GPU power will only cause incremental changes in performance.

No specific stacks or frameworks are required.

Workflow management

The project implemented several workflows based on the following technologies:

- Simple python scripts to manage SLURM jobs for the MD simulations
- HyperQueue for simplifying SLURM resources
- LEXIS for workflows over multiple HPC resources

Only the first method was used for the long time-scale MD simulations, while HyperQueue was used for smaller simulations in a different part of the project. LEXIS was not used in production but was demonstrated to work over multiple HPC sites (LUMI, Leonardo and Karolina). In LEXIS the workflow is described via the web interface.

Access and analysis

The resources are accessed by tools installed on the HPC system, either custom or by programs provided by the authors of the simulation program. It is also possible to access them via Jupyter. As mentioned above, the MD simulation data is being transferred to the MDDB service where it will be possible to access them via a web portal.

Non-technical challenges

The LIGATE project relied very heavily on teamwork and the computational resources were requested by teams, rather than single PIs. In the case of the MD simulations a team of 5 people shared the workload of running and analysing the simulations. Authorizations were performed using the standard UNIX group access permissions. Support from the HPC infrastructure was required on a small number of occasions. The data were not covered by GDPR or any other regulatory frameworks.

Gap analysis

Additional services could include:

- Simpler access to multiple HPC resources, rather than having to negotiate multiple access policies
- Simple, customizable workflow software which include fast data transfer but which can be used without advanced privileges

Annex 13: Simulation of plastic neural networks in the brain

Scientific challenge

During the past decade, brain research has become more and more data-intensive, and HPC-based simulation at different levels of brain organisation has been supplemented by methods of AI, statistical analysis, and neuroimaging, to name a few of many methods. These methods enable researchers to address problems at multiple scales, e.g., to decode the cellular organisation of the brain, to decipher the rules of axonal connectivity, understand brain dynamics in health and disease or to predict features of brain regions based on existing experimental data from other regions. Such research goals have in common that they require most efficient access to data, as well as computational methods to analyse and reproduce information processing and transmission between different structures in the brain. Each of them alone poses significant challenges to HPC and data management at scale, and this is even more true for their integration into efficient workflows.

Brain plasticity

Models of the dynamics of neural networks in the brain are created based on experimental information that has been collected for years by neuroscientists. The connectivity of the neural networks in the brain plays a critical role in defining its function. Connections in the brain get a base structure during development and are continuously refined during the lifespan of a living being. We know that structural and synaptic plasticity, the phenomena which describe the changes in synaptic connections in the brain, are an essential component of learning, memory, adaptation to stimuli, and healing after lesions. Studying the mechanisms that regulate plasticity is fundamental for neuroscience to reach a better understanding of how brain structure and higher function relate.

Although models of biological neural networks have become an important tool to understand brain dynamics, adding plasticity to these models usually leads to larger computational costs which hinders our capacity to test hypotheses in-silico. This is even more visible when adding structural plasticity, as this modifies the structure of the biological neural network model during simulation and critically affects the underlying computational workload.

Given the magnitude of computational resources offered by systems like JUPITER, it is imperative to adapt our methods and model construction algorithms to benefit from them. It is our hope that Exascale modular supercomputing combined with scalable modelling workflows, simulation tools, such as Arbor and NEST, and ML/parameter optimization tools will enable scientists to address questions in this direction.

A wide range of users of neuroscience codes and other stakeholders like clinical practitioners, patients and technology developers can benefit from improvements of computational methods addressing structural and synaptic plasticity at the different spatial and temporal scales in the brain.

The main target scientific community served by the use case are experimental, computational and clinical neuroscientists aiming at understanding the impact of structural changes in brain networks on higher level function.

Direct beneficiaries from the theoretical and computational neuroscience side are users from the NEST and Arbor communities interested in understanding the variations in communication and computational capacity that this type of plasticity introduces. In particular, this use case is relevant to scientists studying brain development, learning and healing after lesions.

The scientific communities served by the use case also include clinical scientists involved in the design of treatment protocols and procedures for a variety of neurological diseases.

On the other hand, this use case is also relevant for scientists and engineers developing innovative computing technologies involving neuroinspired AI and neuromorphic hardware, who will directly benefit from enhanced knowledge regarding learning, brain plasticity and adaptation.

The study of dynamic structural changes in neural networks is still at the beginning. This is mostly because of the complexity of modelling changing neural morphologies, the lack of detailed experimental data, the complex interaction between different types of plasticity rules, and the large time span that needs to be considered – modelled and simulated – to understand the impact of plasticity at different functional scales, including the molecular one.

Studying slow processes such as dynamically changing brain network structures requires simulating long stretches of biological time. While a cortical microcircuit model can currently be simulated faster than real-time with fixed connections[79], when structural plasticity is enabled the simulations of the same model are about 100 times slower than real time[80] , a situation that will be considerably exacerbated by increasing the biological neural network size to the proposed brain-scale models. This can be partially mitigated by increased computational resources, but more efficient strategies are also required and being developed.

Our vision for the near future is that the scalability and computing speed we aim to achieve on systems like JUPITER will enable the consideration of plastic processes to arrive at individualized brain models and understand inter-individual variability. Simulation codes like NEST and Arbor support increased model robustness, reproducibility and open science.

Beyond modeling unstructured biological neural networks, creating digital twins of brain regions like hippocampus and cerebellum with plasticity features is computationally extremely challenging. A human hippocampus model without plasticity is an already published example[81]. Running functionally meaningful simulations of the cortex and human hippocampus is also highly desirable as these structures are deeply involved in the formation of new memories in cooperation. Cerebellar models are published for the mouse[82] and in preparation for the human brain by the community. For these two example models, one of the most time-consuming steps is the initial network building and connectivity generation from data.

Integrating different types of plasticity into large data-based models poses technical challenges for the I/O, memory management and communication at scale, infrastructure and accelerator variability and visualization, robustness, reliability and observability of large workloads running for long periods of time.

There is currently a large effort in progress by 8-10 groups around Europe to explore simulations of the brain at different scales including structural and other types of plasticity. For this reason, the use case has different workflows that are being explored in parallel by the groups. However, a point of cohesion is the usage of simulation tools within the EBRAINS RI[83]. As there is no dedicated funding for this effort, there is also a developing vision for the future, which we hope can materialize in the next 2-4 years.

- Open science and open data, including commitment to FAIR[84] data or other relevant principles and policies.

Every group working on this topic is committed to enhancing the quality and impact of the achieved results by broad adoption of open science practices. This implies accessibility and open licensing of the developed codes and workflows as well as the accessibility of datasets and reproducible digital model descriptions. Most of the work is connected to the EBRAINS RI, which provides services to register code, data and models

[79] Kurth, Anno C., Johanna Senk, Dennis Terhorst, Justin Finnerty, and Markus Diesmann. "Sub-realtime simulation of a neuronal network of natural density." Neuromorphic computing and engineering 2, no. 2 (2022): 021001.
[80] Diaz-Pier, S., Naveau, M., Butz-Ostendorf, M., Morrison, A. (2016). Automatic Generation of Connectivity for Large-Scale Neuronal Network Models through Structural Plasticity. Front. Neuroanat. https://doi.org/10.3389/fnana.2016.00057
[81] Gandolfi, et al. (2023). Full-scale scaffold model of the human hippocampus CA1 area. Nat. Comp. Neurosci. Doi: 10.1038/s43588-023-00417-2
[82] De Schepper, Robin, et al. "Model simulations unveil the structure-function-dynamics relationship of the cerebellar cortical microcircuit." Communications Biology 5.1 (2022): 1240.
[83] EBRAINS is a digital research infrastructure, created by the HBP, that gathers an extensive range of data and tools for brain-related research. EBRAINS capitalizes on the work performed by the Human Brain Project teams in digital neuroscience, brain medicine, and brain-inspired technology. https://ebrains.eu
[84] Findable, Accessible, Interoperable, Reusable; https://www.go-fair.org/fair-principles/

with structured metadata through the EBRAINS Knowledge Graph. Both NEST and Arbor are open source community codes, part of the EBRAINS infrastructure.

Storage

Data-based models of different regions of the brain can have a large span of data requirements. The network generation phase is currently a bottleneck for these models. Other models, where connectivity can be defined by mathematical and statistical rules, are far more parallelizable and tractable regarding I/O. The hippocampus and cerebellum models require connectivity matrices ranging between 10-40 GBs, which need to be processed and used to generate connections between specific neurons in the network.

To study the effects of plasticity on the brain dynamics, it is necessary to record the spiking activity and connectivity changes or a large portion of the network. This means that moving towards models with millions of neurons, data outputs can easily reach 2-5 GB per biological second of simulation.

Connectivity data is usually not so large, ranging a few hundred MBs per run.

Datasets can be analyzed and post-processed to produce relevant information for the structural analysis. Once this step is completed, data can be shifted to long term storage in case further analysis is required.

This use case only requires archival storage for future analysis on the produced synthetic neural network activity.

At the moment, this use case does not have special data confidentiality or security measures. If personalization will be targeted in the future for e.g. the development of special treatment protocols using patient data, special assurances regarding GDPR compliance of the computing and storage sites will be required.

For our most developed use case in NEST, we have the following storage patterns for 6000s of biological time:

- Post processed data is 54MB per simulation distributed in 12 files.

- Raw data comprises about 12.5GB distributed in ~55,000 individual files which are generated in parallel by each MPI process in the simulation..

- Raw data is written down at the end of the simulation in files of .npy format, inside a single folder per simulation run. File access is contiguous.

Data transport

Compute jobs of data-based models need direct access to the connectivity matrices defining the model structure. No other data access or transfer is required.

Once the connectivity matrices are uploaded to the computing site, there is no need to do further data transfers. Computing jobs just need direct access to these large data files containing the connections between neural structures.

Depending on availability of computing resources, we have had to transfer these datasets among different HPC centers, but this is a one-time staging process.

For the output data, it is expected to have constant transfer rates as simulations are executed, analyzed and processed, especially when different parameters for plasticity rules are being explored in a grid fashion.

No streaming access is required.

Unicore FTP is an option we have used in the past, specially for the connectomes required for data-based models of hippocampus and cerebellum.

Compute

We are currently using CPUs with NEST and GPUs with Arbor in a Batch computing model. We hope to be able to use Neuromorphic computing in the near future, as algorithms for structural plasticity are also available in systems like SpiNNaker2 and BrainScales.

We can provide example information for currently run studies.

1. NEST: These are simulations of two population networks with structural plasticity and synaptic scaling where different types of stimulation and deprivation protocols are introduced. The results of this work are published in Lu et al. 2024[85].

For these runs, our standard job uses 10 nodes for 8 hours on the JUWELS cluster using NEST. Our initial results required 2,000 runs of such a job to get statistically meaningful information on two different plasticity parameters. For the post processing jobs we can fit 4 analysis in parallel on 1 node in JUWELS.

2. Arbor: Base simulations with plasticity run with 8 nodes (4 GPUs per node) for 4 hours. Our expected large scale simulations on JUPITER with structural plasticity using cortex and hippocampus are expected to range between 2048 nodes and 6000 nodes (4 GPUs per node) for 24 hours, testing the scalability to the full JUPITER Booster.

Our simulations run with floating point operations.

1. NEST: Due to the large amount of data produced by these jobs, it was only possible to launch about 100 of these jobs per week. It took us more than 6 months to finalize the whole batch of jobs with their proper post-processing.

2. Arbor: We don't yet have information about this, but we expect to distribute about 300 runs of the base setup in the 6 months between May and October 2025.

For the NEST and simple Arbor setup, these are just initial models of simple networks. We expect the time and computational resources to increase substantially in the next months with the more complex examples and data-based networks. We will probably need to move towards in-situ post-processing and analysis of the data generated by the larger models, as storing all the data will be unfeasible.

Our main two software tools are NEST and Arbor, a short general and technical description of both can be found below. Besides these two main simulation tools, we also profit from the EBRAINS RI as a software environment providing a variety of visualization, analysis and machine learning tools.

NEST

NEST is a simulator for large-scale spiking neural networks. It has been developed with a focus on accuracy, efficiency, and HPC-scalability. The development is driven by requirements from neuroscience research and focuses on complex networks of simplified neuron models. The NEST simulator features a C++ core which is scalable from laptops up to the largest HPC resources currently available to neuroscience (using hybrid parallelization via MPI and OpenMP), and a Python interface (PyNEST) for conveniently defining network models. NEST supports several platforms, with continuous developments to improve on-boarding with simplified installation procedures using virtual machines, Conda, and Docker. The GPU component of NEST is a library written in CUDA-C++, which currently covers a core subset of the models and functions offered by NEST and is being actively developed to increase its coverage. The NEST ecosystem additionally comprises

[85] Han Lu, Sandra Diaz, Maximilian Lenz, Andreas Vlachos (2023). The interplay between homeostatic synaptic scaling and homeostatic structural plasticity maintains the robust firing rate of neural networks. eLife 12:RP88376. https://doi.org/10.7554/eLife.88376.2

NEST Desktop, a graphical user interface to the NEST simulator, and NESTML, a domain-specific language for creating code for neuron and synapse models for different backends, including CPU and GPU. NEST is a community code, inviting contributions to the open-source code base.

Performance on HPC systems: The CPU component of NEST has been continuously tuned and improved over several hardware generations and on different supercomputers. NEST showed good parallel efficiency in a weak-scaling study performed on the Petascale systems JUQUEEN and K computer on up to 82,944 compute nodes with MPI parallelization[86]. Today's systems tend to be composed of fewer but more powerful nodes, e.g., JURECA-DC, JUWELS Booster, or JUSUF with each node using two AMD EPYC Rome CPUs. NEST demonstrates excellent strong scaling across all cores of these CPUs using OpenMP threads[87] . NEST can leverage both the existing inter-node, multi-process parallelization and intra-node shared memory parallelization to combine them into a scheme that adapts well to different system architectures, and we expect this investment in software infrastructure to pay off also on upcoming machines.

The main challenge for NEST moving forward to larger systems will be porting it efficiently to GPUs, an activity that is in progress by the community. Structural plasticity is also being actively worked on for this version of the software.

Arbor

Simulations of biological neural networks with thousands of detailed neurons and hundreds of thousands of synapses are computationally extremely demanding, especially if complex plasticity rules and spatial structure of neurons are being considered. Studies investigating memory formation and consolidation in biologically realistic neural networks (with spiking point neurons and calcium-based, multi-phase plasticity, simulated across hours of biological time[88]) require a large investment of computational resources. Moving from point neurons to modelling the spatial structure of neurons requires further resources, depending on the desired spatial discretization and the number of considered cellular extensions.

To cope with these issues, first, one can use code written in machine-near programming languages such as C++. Second, one may employ available high-performance hardware. However, the entry barrier and the time needed to get involved with these topics are high. By providing an easily accessible frontend, Arbor enables users to take both approaches: it uses fast C++ code, it can run simulations on a variety of backends, including GPUs, and it enables to distribute computation across the compute nodes of high-performance systems in an optimised fashion. Thereby Arbor makes optimal use of the available hardware to implement complex plasticity rules and spatial complexity of neurons Arbor was conceived under the auspices of the HBP with the goal of unlocking modern HPC for neuroscientific research on networks of detailed neurons.

Arbor employs modern and mature software development practices and maintains a culture of quality throughout its procedures and code. It uses Git for version control and GitHub to host its code repository and nearly all of its infrastructure, including CI. We use Github's PR workflow: new requests are hashed out in the Issue tracker, after which a PR may be submitted addressing the request. Code review is a requirement for merging of PRs, and constructive yet rigorous reviews are the norm. Passing of our extensive test-suite (CI), defined as Github Actions, is the other major requirement.

HPC-readiness/scaling, motivation, and challenges: To underline the claim that Arbor scales well, we show a set of benchmark results going up to almost the full JUWELS booster system. The model system comprises continuously spiking rings of neurons interconnected with zero-weight synapses. This setup leads to a stable behaviour of the local rings while placing load on the interconnect. The computationally intensive cell

[86] Jordan, J., Ippen, T., Helias, M., Kitayam,a, I., Sato, M., Igarashi, J., Diesmann, M. and Kunkel, S. (2018). Extremely Scalable Spiking Neuronal Network Simulation Code: From Laptops to Exascale Computers. Front. Neuroinform. 12:2. doi: 10.3389/fninf.2018.00002

[87] Kurth, AC., Senk, J., Terhorst, D., Finnerty, J. and Diesmann, M. (2022). Sub-realtime simulation of a neuronal network of natural density. Neuromorph. Comput. Eng. 2 021001. doi: 10.1088/2634-4386/ac55fc

[88] Luboeinski, J., & Tetzlaff, C. (2021). Memory consolidation and improvement by synaptic tagging and capture in recurrent neural networks. Communications Biology, https://doi.org/10.1038/s42003-021-01778-y

model is based on the Allen database of the mouse primary visual cortex[89]. We show weak scaling for three different fill rates of the accelerator memory. At 100% we place 192,000 cells on the device. For node counts 40 to 200 we find perfect scalability, at 400 efficiency drops to 80% and at 800 to 60%. This is an issue the community is seeking to address going forward in the project. For perfect weak scaling the time spent should be constant, yet at 400 nodes and beyond we find a positive scaling; i.e., a loss in efficiency. Scaling performance is in large part enabled by developing scalable modelling primitives.

Arbor is written from the ground up with optimised performance for a wide variety of hardware configurations in mind. Its design principles, software development culture and goals are firmly focussed towards enabling the largest neuronal network simulations of detailed cells possible. The ability of neuroscientists to create ever larger models, and approach the scales of the human brain, goes hand in hand with the ability to scale the execution of such models across ever larger HPC systems. In that sense, Arbor is a key ingredient in the future of computational neuroscience involving detailed cells.

Workflow management

Usually a simulation step is followed by a post-processing step doing analysis and/or visualization of the network dynamics and structural changes. This is usually done on a few nodes, within one site. In the future, we foresee that in-situ analysis and interactive visualization will be required to: 1) identify at early stages when simulations are not generating the desired results and reduce resource wasting, 2) understand the results of the simulation and 3) optimize the data production to retain only the essential information. None of these stages are time-critical

At the moment, workflows are not too complex and can be defined using Unicore or multiple steps with dependencies in SLURM batch scripts. However, NEST and Arbor are both part of the EBRAINS RI and CWL is a common tool to define workflows in this environment. We can expect more complex workflows to emerge for this use case and CWL or Unicore would be good alternatives to formalize and automate the execution.

We do not explicitly use a workflow orchestrator for the moment. For some instances of the use case we have used the L2L framework [67], which is an implementation of the concept of learning-to-learn or meta-learning. Learning-to-learn is a machine learning approach to improve (learning) performance by generalisation. The framework is implemented in Python, is open-source, and follows an open development approach. This tool allows us to optimize parameter space explorations and reduce resource utilization.

Access and analysis

At the moment we mostly work offline on batch mode with these simulations, access via ssh and scp. However, we also use the JupyterLab at JSC, especially when we want to share the results of the developers, collaborators and students getting started with individual questions within this use case.

A few years ago we developed the first interactive steering and visualization tool for structural plasticity for HPC, which worked extremely well to understand the results from the simulations and also change the simulation parameters during simulations. We would like to continue developments in this direction in the near future.

Most of the analysis for the different workflows in this use case are analyzed in-situ.

In-silico experimentation within this use case would highly benefit from a combination of fast, interactive access to the computing infrastructure and a scalable system which can simulate and analyze data efficiently. Integrated workflows combining data exploration and querying, model generation, simulation, analysis, visualization, and machine learning will provide new ways to achieve the future goals of the use case.

[89] Billeh, Y. N., Cai, B., Gratiy, S. L., Dai, K., Iyer, R., Gouwens, N. W., ... & Arkhipov, A. (2020). Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex. Neuron, https://doi.org/10.1016/j.neuron.2020.01.040

Non-technical challenges

As this use case is carried forward by a variety of groups working in parallel in a distributed manner, most groups request computing time and storage resources for specific workflows independently. However, the SDL Neuroscience supports most of these groups in the formulation of computing time proposals and the optimization of workflows to be executed on the supercomputer.

Today, the effort spans around 8-12 groups Europe-wide. We heavily rely on software and services by the EBRAINS RI, so it would also be ideal if we could have the EBRAINS AAI as authorization system, however this is not necessary. For the moment we use the HPC site-specific authentication and authorization services to gain access to specific projects.

We are heavily involved in training and education activities on how to use HPC for neuroscience research. The SDL Neuroscience generates tutorials and provides dedicated workshops to collaborators, interested PIs and early career researchers who would also like to use simulations with structural plasticity for their own research. As all our code and models are made public and available through the EBRAINS RI, any user of the NEST, Arbor or EBRAINS community can directly benefit from the code, examples and tutorials made available.

Data is not covered by any regulatory frameworks, but in the near future both GDPR and the AI Act could become relevant. As mentioned before, if personalized simulations would be developed in the future, they would have to be executed under GDPR compliance. As this use case can also have an impact on the development of new methods for neuro AI, the AI Act could also become relevant under specific circumstances.

Gap analysis

We are currently moving forward with the implementation of the scientific and technical goals in this use case. We do not have specific funding to work on this, but each group is contributing with in-kind effort and we continuously aim at developing the community and finding ways to further collaborate.

In technical terms, the optimization of the algorithm to perform structural plasticity simulations in NEST and Arbor needs to keep evolving to reach good performance at Exascale. We have no doubt that larger amounts of computational resources will be required to fulfill the objectives of this use case, together with better workflows with in-situ visualization and analysis. Moreover, we need to solve issues related to the network instantiation for the data-based models of specific brain regions, and we need to optimize the communication strategies if we want to be able to execute larger simulations for long biological times.

Annex 14: Mistral Use Case

Scientific challenge

The MeteoHub platform addresses critical scientific challenges within the field of meteorology by providing a centralized and harmonized repository for weather data. The platform is rooted in a long-standing tradition of supercomputing applications in atmospheric modeling, aiming to meet the scientific community's increasing demand for accurate, timely, and accessible weather data.

The primary users of this use case include researchers, public administrations, and private organizations working in meteorology, climatology, and disaster management. Achieving the platform's objectives entails significant technical challenges, such as ensuring the rapid availability of high-quality data, often in near real-time, integrating disparate datasets from numerous providers, and enabling specific post-processing operations during data extraction to tailor the outputs to user needs. Historically, weather information was scattered across various sources, some of which were not easily accessible. MeteoHub, through the Mistral project, resolves this by aggregating, standardizing, and distributing data while enabling the development of value-added services.

This initiative describes both current capabilities and a vision for the future, with plans to incorporate new datasets, including satellite observations and advanced forecasting models. Aligned with open data policies, MeteoHub ensures the accessibility of most datasets, supported by CKAN cataloging to enhance data discoverability and reuse. Data is provided under either the CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/) or CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0/) licenses. This ensures both open access and proper attribution for reuse.

Storage

The MeteoHub platform requires substantial storage capacity to support its scientific and operational goals, with an estimated volume of approximately 2 TB for observational data, user personal space data, and Apache Nifi data, supplemented by around 280 TB of forecast data, including historical weather data. Daily forecast generation significantly contributes to storage demands.

For individual units of work, the data throughput is approximately 300 MB when extracting daily data from all observed datasets, while extracting forecast data for the next 72 hours from a single dataset typically involves 27 GB. Data retention for MeteoHub is lifelong, ensuring continuous availability of critical information, with data quality control measures inherently provided by the data sources. Data safety is a priority, with backup solutions currently being implemented on other storage systems. No specific confidentiality requirements are imposed at this time.

MeteoHub currently manages 27 observational datasets and 19 forecast datasets. Observational data is ingested via Apache Nifi into a Postgres database (DBAIIE) before being transferred to Arkimet, a meteorological database developed by ArpaE, where forecast data is also stored following HPC processing. Both databases support read and write operations, primarily accessed via APIs. Reads are significantly more frequent than writes, with approximately two writes per day for each forecast dataset and at least three writes per hour for observed data during the ingestion process. User traffic primarily dictates the read access patterns, with an average of around 100 active users and 500 data extraction requests per month. Given the recent trends, these numbers are expected to continue growing over time.

Data transport

Data transport within the MeteoHub platform consists of two primary flows: the ingestion of observational data from various providers and the extraction of data by users. Observational data is acquired via Apache NiFi workflows and initially stored in the DBAIIE database. After a retention period of ten days, the data is transferred to the Arkimet database using APIs provided by both systems. Forecast data, in contrast, is generated by meteorological models running on HPC infrastructure, with MeteoHub accessing the folders

where this data is stored. Users can extract data through the web interface, APIs, or the Mistral Meteo-Hub-CLI, a command-line tool for downloading data prepared interactively or scheduled on the platform. The CLI also supports wait-and-download operations via AMQP data-ready queues.

The platform operates on a cloud infrastructure hosting virtual machines for services, alongside HPC systems for running models and generating forecast data. Streaming access is available to selected project partners through AMQP queues, enabling simultaneous data distribution during ingestion. As the project evolves, data transfers are expected to grow linearly, driven by the acquisition of new datasets and forecasting models. Data transfer technologies include FTP, AMQP queues, and HTTP for ingesting observational and radar data.

An analysis of users' requests involving post-processing over the past year indicates an average duration of approximately 1.5 minutes, weighted by the size of the generated files. While this demonstrates the system's ability to handle computationally intensive workflows, further improvements in computation time could be achieved with additional resources. The current architecture adequately supports data flows and provides a degree of scalability, though there is room for enhancement.

Compute

The computational demands of MeteoHub focus primarily on data ingestion and user-requested data extraction, including optional post-processing. Observational data ingestion involves retrieving datasets from providers, processing them, and loading them into the DBAIIE database with custom scripts. Given the high granularity of some data, such as minute-level pluviometric measurements, ingestion workflows require significant CPU resources to ensure efficient processing and organization.

For data extraction, users submit batch requests through APIs, with options for post-processing to tailor the data to their needs. Post-processing capabilities include computing derived variables (e.g., dew-point temperature, wind components, air density), temporal computations (e.g., averages, maxima, minima, accumulations), format conversions (e.g., BUFR to JSON), and spatial manipulations for forecast data (e.g., area cropping, grid interpolation, or applying user-provided templates such as shapefiles). These operations culminate in the creation of downloadable data packages that meet specific user requirements.

The current computational infrastructure relies on CPUs within a cloud environment, dedicated to supporting data ingestion and data extraction workflows. HPC resources are used exclusively for generating forecast data, which is then made accessible through MeteoHub for further processing and distribution. In the future, AI-based tools or GPUs could be integrated to enhance advanced post-processing capabilities and manage the anticipated growth in data volume.

Data extraction tasks are executed sequentially, with each batch request processed independently. While the system does not currently leverage parallelism, scaling opportunities may emerge as data volume increases and post-processing workflows are refined. This growth highlights the potential need for expanded computational resources to support MeteoHub's evolving role in the data computing continuum.

Workflow management

Workflow management within MeteoHub relies on automated, multi-step processes designed to handle both data ingestion and user-requested data extraction. Observational data ingestion is managed by Apache NiFi, which orchestrates workflows tailored to various data retrieval protocols, including AMQP queues (managed by RabbitMQ), FTP, HTTP, and specialized workflows for specific providers. A diagram illustrating the ingestion process is provided in **Figure A14.1**. These workflows are executed entirely within the Docker container hosting NiFi.

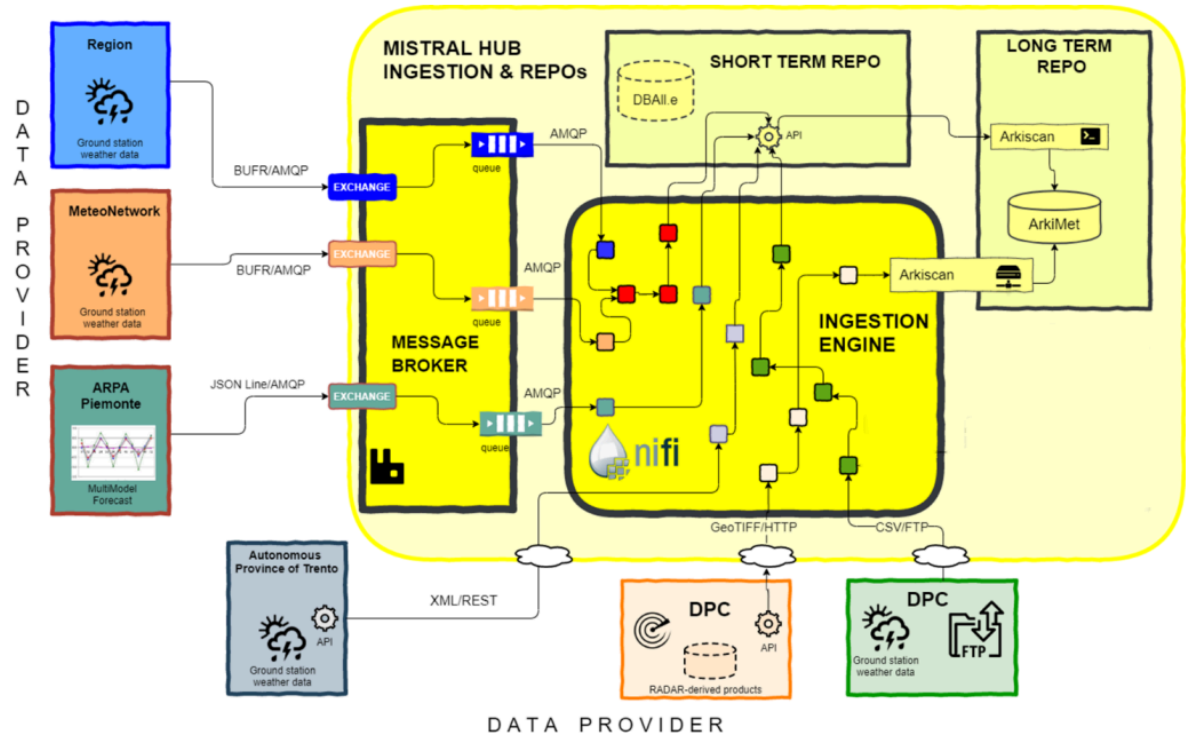


Figure A14.1: Mistral Hub data ingestion pipeline.

Data extraction tasks are managed by backend and Celery containers, which handle database queries, optional post-processing, and the generation of downloadable data packages.

Certain steps in these workflows are time-sensitive. For instance, during ingestion, handling high-granularity datasets such as minute-level precipitation data can result in bottlenecks due to the large volume and frequency of incoming data. Similarly, post-processing tasks, such as precipitation data accumulation, can be computationally intensive and may slow down data extraction workflows because of their high level of granularity.

The ingestion workflows are configured using NiFi Flow Configuration Language (NFCL), an XML-based language. MeteoHub's execution environment is fully containerized and managed through the Rapydo framework. All containers run on virtual machines that also support frontend services and map visualization. The entire infrastructure is hosted on a cloud environment and monitored using CheckMK. Future plans may include exploring the adoption of Kubernetes to enhance orchestration and scalability.

Access and analysis

Users authenticate to the platform via Single Sign-On (SSO), using a token that grants access to both the web portal and APIs. They primarily interact with the platform through these interfaces, which enable data extraction and optional post-processing workflows.

Post-processing computations might be performed in situ within the MeteoHub infrastructure. However, users have the flexibility to download data and perform further analysis on their local machines if desired.

Interactive facilities for computational steering are not required. All workflows are designed to be initiated and managed through the web portal or API calls. Data extraction requests, including post-processing, can be submitted through either interface, providing a streamlined user experience.

Non-technical challenges

Access to computational and data storage resources is granted through SSH for administrators, a limited number of individuals who connect to virtual machines for maintenance tasks. HPC data storage is accessed via the Lustre file system, which integrates HPC-generated data into the cloud environment, making it accessible for processing and analysis.

To support users, MeteoHub provides a comprehensive user manual for interacting with the platform's APIs and web portal, which can be found here: https://www.mistralportal.it/user-guide/. Additionally, tutorials and training materials are available at https://www.mistralportal.it/tutorial-training/ , offering users guidance on effectively utilizing the platform. For any issues or support requests, users can contact the support team via email at mistral-support@cineca.it. A ticketing system is planned for future implementation to streamline support services.

Currently, the data made available through MeteoHub is not subject to data protection or other regulatory frameworks such as GDPR or the AI Act.

Gap analysis

Despite the robust infrastructure and services provided by MeteoHub, there are some key areas where additional capabilities could enhance productivity. The most notable gaps include the need for more computational capacity to handle growing data volumes and complex post-processing tasks. Additionally, scalability of the worker nodes is a critical factor to support an increasing number of concurrent data extraction requests. To address these challenges, the adoption of Kubernetes for container orchestration could provide greater flexibility and scalability in managing workloads, improving overall performance and efficiency.

Annex 15: Artificial Intelligence to find and characterize HI-rich galaxies

Scientific challenge

The SKA Mid frequency interferometer (SKA-Mid) in the Karoo of South Africa has been designed as an observatory, with a broad array of scientific objectives – from probing the formation and evolution of galaxies to studying gravity through surveys of pulsars. One of the earliest scientific motivations for building the SKA has been to study the atomic gas content and kinematics of galaxies by measuring the redshifted 21cm neutral hydrogen line in emission and absorption. Current HI surveys with SKA pathfinders and precursors have detected HI in emission in less than 20k galaxies out to a maximum redshift of z~0.4 when the Universe was 9.4 Gyr years old. The expectation is that the SKA-Mid surveys should increase the number of detections to ~105 and back in time almost 2 Gyr toward the Cosmic peak of star-formation.

The SKA science teams have indicated that surveys could be conducted in three, successively smaller and deeper (longer integration time) tiers [6]. Data processing begins at the Science Data Processor (SDP) located at the HPC Science Processing centre in Cape Town, South Africa. The SDP generates Observatory Data Products (ODPs) in the form of radio spectral line cubes ("model", "residual" and "clean" images) and gridded visibility data. These are then transported over 100 Gbps links to the SKA Regional Centres for further processing.

The SKA Regional Centre Network (SRCNet) will provide the user interface, analysis tools and long-term data preservation of data products for the SKA Observatory (SKAO) user community [1]. The SKA scientific community will be globally distributed. These external scientific users and observatory staff will have the ability to search the archive and retrieve either ODPs, or Advanced Data Products (ADPs) such as HI source candidate lists. For the purpose of this use case, we describe the user processing and handling of the extragalactic radio continuum images. The SRCNet architecture and operations plans to abide by the FAIR data principles.

For this use case, the survey would be conducted using the lowest frequency Bands 1 (350 to 1050 MHz) and 2 (950 to 1760 MHz) receivers on SKA-Mid, following a "wedding cake" observing strategy. The three tiers of the cake would be successively deeper (longer integration time) and narrower (smaller area), with the widest tier planned to cover 20,000 square degrees, while the medium and deep tiers would cover 400 and ~10 square degrees, respectively. The total observing time spent on each tier would be in the range of 2000 to 3000 hours. In the case of the wider area Band 2 surveys, the required spectral resolution would be 10.7 kHz over a bandwidth of 470 MHz, while the Band 1 survey tier would cover 450 MHz between 600 and 1050 MHz, also with 10.7 kHz resolution.

As the SRCNet hardware and software are scheduled for initial prototyping in early 2025 [4], and the full deployment is expected toward the end of the decade, this use case should be viewed as forward looking.

Storage

The data storage requirements for the SRCNet anticipate receiving ~700 PB total of ODPs from the two telescopes each year. These datasets will initially be used to make Project-Level Data Products (PLDPs) and ADPs that will be archived. The typical size of an ODP will depend on the length of the observing block, and for this experiment the expected data rate is uncertain at the present.

The survey project team users (scientific data analysts) will be provided with a personal file system for a limited duration to which they can upload and download small files and submit new ADPs, including from the archive, to within the per-project resource allocation [5]. These files are available for further processing and visualisation, as well as to hold code and additional uploaded data for analysis. Processing logs will be stored along with the data for all processing operations and will record information on software and resources used and any other parameters to ensure reproducibility of new data product generation.

In addition to the file system, users can generate their own databases which can be created, accessed, updated and queried using tools such as Structured Query Language (SQL), in order to store structured data relevant to their science cases [5]. Database rights will be shareable between groups of users.

ODPs will typically have a proprietary period of one year, giving the project teams the chance to analyse and publish their data before they are made public to the broader scientific community. All datasets will exist in multiple locations for security. It is possible that after one year, the data will be moved from a "hot" storage buffer to "cold" storage [3].

Data transport

Data will be stored across the SRCNet, throughout nodes located in Canada, Europe (including UK), South Africa (Cape Town), Australia, India (Pune) and China (Shanghai). There must be at least 1 Gbps upload and download speed between all nodes in the SRCNet. Users will be distributed globally, and not necessarily collocated with the data they are retrieving and viewing.

It is anticipated that the data transfers will be continuous as new projects are completed at the telescope sites and data are moved to the SRCNet. The typical telescope observing block might be in the range of 4 to 8 hours, primarily during night time when ionospheric fluctuations are lower. These datasets will then be processed in Cape Town at the HPC Science Processing Centre before distribution to the SRCNet nodes. Real time data streaming to the SRCNet will not be required for this use case.

The user authentication system should use federated protocols that all the SRCs could use and, also, would allow different identity providers. The SRCNet will also make use of Virtual Observatory (VO) protocols to allow users to compare their SKA data products to images from other observatories.

Compute

We assume that data products are processed multiple times in the SRCs (whilst the arrays are small the data rates are so low that this is not a driver) but less frequently once the arrays are fully built and the pipelines are more robust. Using a parametric model for the Science Data Processor (the HPC based in Perth and Cape Town), we estimate the total computing requirements of the SRCNet to be up to a peak of 35 PFLOPs. For good performance, high-performance storage close to the computing units could be needed ("hot" buffer) [4].

For the data analysis performed at the SDP stage, a barycentric frequency correction is applied before radio frequency interference (RFI) is removed. The visibility data are calibrated before peeling of strong continuum sources and a polynomial is fit in frequency space to model and then subtract the continuum. Spectral line cubes, beams, and gridded visibility data are generated for each day before these are transported to the SRCNet for further processing. At the SRCNet nodes, these cubes (and beams), are combined in the image and visibility domain before a multiscale deconvolution is performed.

The data processing steps within the SRCNet will also require the extraction of advanced data products (ADPs) in the form of HI source candidate lists. There are several tools that have been used for source extraction within the SKA HI science community (several methods are presented in Hartley et al. 2023). Owing to the large aerial coverage of SKA images, Machine Learning (ML) algorithms are expected to be used more frequently for finding and characterising galaxies. One such method that could be deployed within the SRCNet would use a modified YOLO (you only look once) network, a regression-based classifier with a convolutional neural network architecture. This network is first trained on a simulated dataset of HI galaxies before being applied to the actual images. Such a method has already been demonstrated to achieve over 80% accuracy (Hartley et al. 2023).

More details on the computing and processing requirements will become available as the SRCNet prototyping activities evolve beginning at the end of 2024 and throughout the construction phase.

Workflow management

The SRCNet science analysis platform will facilitate workflow management with a tool that will allow users to specify individual workflow steps and combine them to form a larger workflow which can be stored in the software repository and re-used by others. It will be possible to combine workflow steps sequentially as well as using simple programming constructs (and, or, if), and each workflow step will draw on tools from the software repository, such as code within a notebook, a call to one of the pre-defined APIs, a call to a piece of software that is pre-installed within an existing defined software environment, or a call to another workflow [5].

After a workflow is defined, there will be options to run the workflow either in real-time or as a background process that can be scheduled to maximise efficiency and prioritisation, with the user being notified upon completion/failure.

The exact details of the workflows will become clearer as the SRCNet prototyping activities evolve.

Access and analysis

The SRCNet will implement a federated Authentication and Authorisation Infrastructure (AAI). This will integrate national federations through an international inter-federation service (e.g. eduGain) to enable the use of existing institutional accounts, and to allow the use of existing institutional credentials to authenticate with the SRCNet Infrastructure. The AAI will link these credentials to a centrally coordinated unique SRCNet Identity (SKA-ID?). A network of coordinated services will manage group membership and other relevant attributes, facilitating authorisation decisions for access to SRCNet data, computing, and other resources [3].

The main UI for the platform will be a web page (the 'Gateway', which will be hosted by the SRCNet) providing access to the functionality of the SRC node through a range of services. The user can sign on to the portal on the front page using single sign-on criteria and will then be given access to the full UI; without signing on there will likely only be limited public data access to, for example, image previews and catalogues. The UI will be consistent across different SRC nodes.

Once a dataset has been selected, users will need to be able to visualise the data either by running tools built into the platform, or by spawning a tool that runs within a software environment or notebook. The visualisation tools provided by the platform will include tools suitable for large datasets. Other panels will provide access to a notebook (possibly Jupyter) for interactive analysis and the ability to run containers, Virtual Machines (VMs), or distributed jobs, as well as constructing more complex workflows [5].

Non-technical challenges

The SKA Observatory data access policy gives exclusive rights to scientific project team members for a specified duration of time (currently assumed to be one year). After this proprietary period expires, anyone in the broader scientific community will be permitted to access the data. Survey teams may consist of ten, or more members, however it is likely that four, or five people will be actively working on a project data set.

Scientific and technical staff based at the SRCNet nodes will be expected to run regular training sessions for the community. One of the objectives of the SKAO and SRCNet operating model is to ensure that even non radio astronomers should be able to extract scientific data from the observatory.

GDPR will apply to the user account data, and possibly the AI Act will apply to machine learning models developed for astronomical data analysis.

Gap analysis

As the telescope time allocation process has not yet occurred, and the prototyping of SRCNet has just begun, many of the assumptions presented here are likely to evolve.

References

1. Bolton, R., et al., 2023, SRCNet Vision and Principles
2. Franzen, T. et al., 2023, SRCNet Use Cases
3. Hartley, P. et al., 2023, MNRAS, 523, 2
4. Salgado, J., et al., 2023a, SRCNet Software Architecture Document
5. Salgado, J., et al,, 2023b, SRCNet Top-Level Roadmap
6. Skipper, C. et al., 2023, SRCNet Science Analysis Platform Vision
7. Wagg, J., et al., 2021, SKA1 Scientific Use Cases