A Roadmap for HEP Software and Computing R&D for the 2020s

The HEP Software Foundation

Table of Contents

1 Introduction

Particle physics has an ambitious programme of experiments for the coming decades. The programme supports the strategic goals of the particle physics community that have been laid out by the European Strategy for Particle Physics [ESPP2013] and by the Particle Physics Project Prioritization Panel (P5) in the United States [P5-2014]. Broadly summarised these programmes aim to:

- exploit the discovery of the Higgs boson in 2012 as a precision tool for investigating Standard Model (SM) and Beyond the Standard Model (BSM) physics,
- explore matter-antimatter asymmetry, particularly via the properties of B mesons,
- investigate the properties of dark matter,
- probe neutrino oscillations and masses.

The High-Luminosity Large Hadron Collider (HL-LHC) will be a major upgrade of the current LHC supporting the aim of an in-depth investigation of the properties of the Higgs boson and its couplings to other particles. The ATLAS and CMS experiments will measure this, alongside searching for new physics Beyond the Standard Model (BSM), or exploration of that physics, should an earlier discovery be made. Such BSM physics may help shed light on the nature of dark matter, which we know makes up the majority of gravitational matter in the universe, but which does not interact via the electromagnetic or strong nuclear forces [Mangano2016].

The LHCb experimental programme at the LHC and the Belle II experiment at KEK study heavy flavor physics, or B physics, where quantum influences of very high mass particles are manifest in lower energy phenomena. Their primary goal is to look for BSM physics in charge parity (CP) violation (that is, asymmetries in the decays of particles and their corresponding antiparticles) and in rare decays of beauty and charm hadrons. Current observations of these asymmetries do not explain why our universe is so matter dominated. These flavour physics programmes can be related to BSM searches through effective field theory and powerful constraints on new physics can come from such studies.

The study of neutrinos, their oscillations and mass, can also shed light on matter-antimatter asymmetry. The DUNE detector will provide a huge improvement in our ability to probe this physics, detecting neutrinos from the Long Baseline Neutrino Facility at Fermilab, as well as linking to astro-particle physics programmes through the potential detection of neutrinos from supernovas.

In the study of the early universe, immediately after the Big Bang, it is critical to understand the phase transition between the highly compressed quark-gluon plasma and the nuclear matter in the universe today. The ALICE experiment at the LHC and the CMB and PANDA experiments at the Facility for Antiproton and Ion Research (FAIR) at Darmstadt are specifically designed to probe this aspect of nuclear and particle physics.

These experimental programmes already require large investments in upgraded hardware. Similarly, they require commensurate investment in the research and development necessary to deploy software to acquire, manage, process, and analyse the data.

For the HL-LHC, which is scheduled to begin data taking in 2026 and to run into the 2030s, some 30 times more data than the LHC has currently produced will be collected by ATLAS and CMS. As the total LHC data magnitude is already close to an exabyte, it is clear that the problems to be solved require approaches beyond simply scaling current solutions from today's technologies, assuming Moore's Law and more or less constant operational budgets. The nature of computing hardware (processors, storage, networks) is evolving, the quantity of data to be processed is increasing dramatically, its complexity is increasing, and more sophisticated analyses will be required to maximise physics yield. Developing and deploying sustainable software for the future and upgraded experiments, given these constraints, is both a technical and a social challenge. Thus a “software upgrade” is needed to run in parallel with the hardware upgrades planned for the HL-LHC.

In planning for the HL-LHC in particular, it is critical that all of the collaborating stakeholders agree on the software goals and priorities, and that the efforts complement each other. In this spirit, the HEP Software Foundation (HSF) began a planning exercise in late 2016 to prepare a Community White Paper (CWP) at the behest of the Worldwide LHC Computing Grid [WLCG2016]. The goal of the CWP is to provide a roadmap for software R&D in preparation for the HL-LHC era and for other HEP experiments on a similar timescale, which would identify and prioritise the software research and development investments required:

- to achieve improvements in software efficiency, scalability and performance and to make use of the advances in CPU, storage and network technologies,
- to enable new approaches to computing and software that can radically extend the physics reach of the detectors,
- to ensure the long term sustainability of the software through the lifetime of the HL- LHC.

The CWP process, organized by the HSF with the participation of the LHC experiments and the wider HEP software and computing community, began with a kick-off workshop at UCSD/SDSC, USA, in January, 2017 and concluded with a final workshop in June, 2017 in Annecy, France, with a large number of intermediate topical workshops and meetings. The entire CWP process involved an estimated 250 participants.

To reach more widely than the LHC experiments, specific contact was made with individuals with software and computing responsibilities in the FNAL muon and neutrino experiments, Belle II, the Linear Collider community as well as various national computing organisations. The CWP process was able to build on all the links established since the inception of the HSF in 2014.

Working groups were established on various topics which were expected to be important parts of the HL-LHC roadmap: *Careers, Staffing and Training; Conditions Database; Data Organisation, Management and Access; Data Analysis and Interpretation; Data and Software Preservation; Detector Simulation; Event Processing Frameworks; Facilities and Distributed*

Computing; Machine Learning; Physics Generators; Software Development, Deployment and Validation/Verification; Software Trigger and Event Reconstruction; and visualisation. The work of each working group is summarized in this document, with links to the more detailed topical documents when they exist.

This document is the result of the CWP process. We firmly believe that investing in the roadmap outlined here will be fruitful for the whole of the HEP programme.

2 Software and Computing Challenges

By the end of LHC Run 2 it is expected that about 150 fb^{-1} of physics data will have been collected by ATLAS and CMS. Together with LHCb and ALICE the total size of LHC data will be around 1 exabyte, as shown in the table below from the LHC's Computing Resource Scrutiny Group [CSRG2016]. The CPU allocation from the CSRG for 2017 to each experiment is also shown.

Experiment	**Disk Usage (PB)**	**Tape Usage (PB)**	**Total (PB)**	**CSRG CPU 2017 (kHS06)**
ALICE	98	86	184	751
ATLAS	164	324	488	1828
CMS	141	247	285	1678
LHCb	41	79	120	376
Total	**444**	**633**	**1077**	**4633**

Using a conversion from HS06 to CPU cores of around 10 means that LHC computing in 2017 is supported by almost 500k CPU cores.

These resources are deployed everywhere from close to the experiment themselves at CERN to a worldwide distributed computing computing infrastructure, the WLCG. Each experiment has developed its own workload and data management software to manage their share of WLCG resources.

In order to process this data, the 4 LHC experiments have written more than 12 million lines of code over the last 15 years. This has involved contributions from thousands of physicists, encompassing a huge range of skill levels. The majority of this code was written for a single architecture (x86) and with a serial processing model in mind. There is considerable anxiety in the experiments that much of this software is poorly maintained, with the original authors no longer in the field and much of the code itself in a poorly maintained state, ill documented and lacking tests. This code, which is mostly experiment-specific, manages the entire experiment data flow, including data acquisition, high-level triggering, calibration and alignment, reconstruction (of both real and simulated data) and final data analysis.

The HEP community also has a wide range of software that is shared. This includes ROOT [Brun1996] as a data analysis toolkit (though also playing a critical role in the implementation of experiment's data models) and GEANT4 [Agostinelli2003] as the simulation framework through which most detector simulation is achieved. Physics simulation is supported by a wide range of event generators from the theory community ([SHERPA], [ALPGEN]). There is also code developed to support the computing infrastructure itself, such as the CVMFS distributed caching filesystem [CVMFS], the Frontier database caching mechanism [Frontier], the XRootD file access protocol [XRootD] and a number of storage systems (dCache, DPM, EOS).

The list above is by no means exhaustive, but illustrates the huge range of software employed by the HEP community and its critical role in almost every aspect of the programme.

When considering the challenges ahead, even in Run 3 LHCb will process, in software, more than 40 times the number of collisions that it does today and ALICE will readout Pb-Pb collisions continuously at 50kHz. The upgrade to the HL-LHC then produces a step change for ATLAS and CMS. The beam intensity will rise substantially giving bunch crossings where pile-up (the number of discrete pp interactions) will rise, from about 60 today, to about 200. The two experiments will upgrade their trigger systems to record about 10 times as many events as they do today. It is anticipated that HL-LHC will eventually deliver about 300 fb^{-1} of data each year.

The steep rise in resources that are then required to manage this data are estimated in Figures 1 and 2.

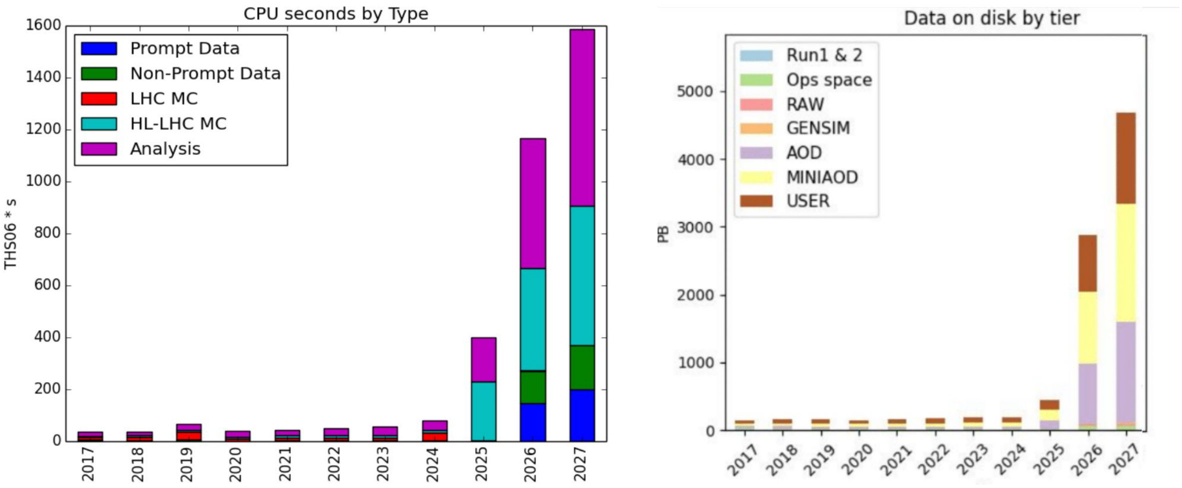

Figure 1. CMS CPU and disk requirement evolution into the first two years of HL-LHC [Sexton-Kennedy2017]

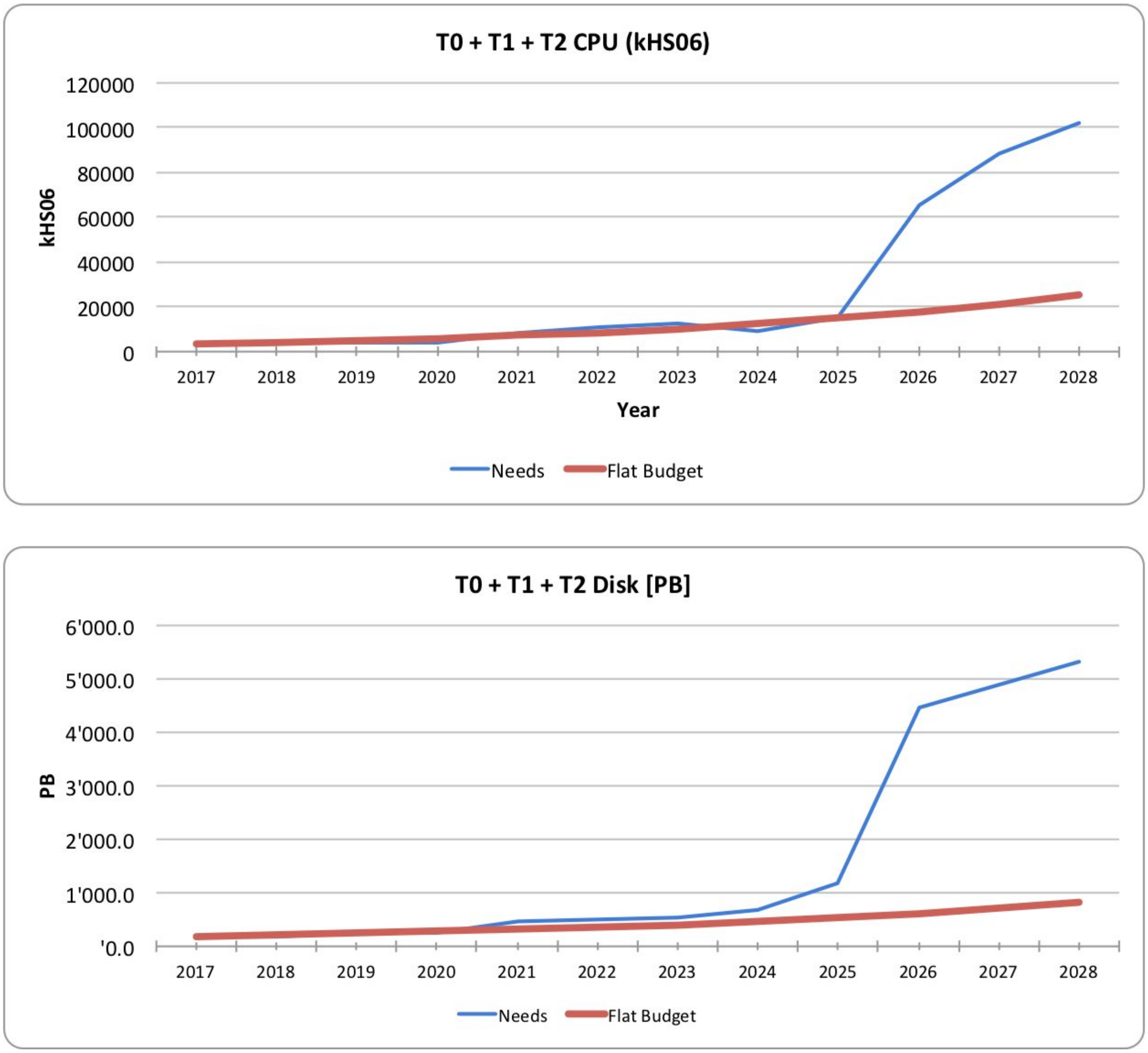

Figure 2. ATLAS CPU and disk requirement evolution into the first three years of HL-LHC [Campana2017]

In general it can be said that the amount of data that experiments can collect and process in the future will be limited by affordable software and computing, not by physics.

The ATLAS numbers, in Figure 2, are particularly interesting as they estimate the resources that will be available to the experiment if a flat funding profile is maintained, taking into account the expected technology improvements given current trends [Panzer2017]. As can be seen, the shortfall between needs and bare technology gains is considerable: x3.7 in CPU and x7.1 in disk in 2027.

While the density of transistors on silicon continues to increase following Moore's Law (albeit more slowly than in the past), power density limitations have limited the clock speed of processors for more than a decade. This has effectively stalled any progress in the processing capacity of a single CPU core. Instead, increases in potential processing capacity come from increases in the core count of CPUs and wide CPU registers. Exploiting this potential requires a shift in programming model to one based on *concurrency*. As a response to this problem in providing effective use of transistors on a die, alternative architectures have become more commonplace. These range from the many core architecture of the Xeon Phi, which combines around 64 modest, but standard, x86_64 cores, to alternatives such as GPGPUs, where the processing model is very different, allowing a much greater fraction of the die to be dedicated to arithmetic calculations, but at a price in programming difficulty and memory handling for the developer that tends to be specific to each processor generation. Further developments may even see use of FPGAs for more general purpose tasks.

Even with the throttling of clock speed to limit power consumption, power remains a major issue. Low power architectures are in huge demand. At one level this simply challenges the dominance of x86 with, for example, Aarch64 devices. More extreme is an architecture that would see specialized processing units dedicated to particular tasks, but with possibly large parts of the device switched off most of the time, so-called dark silicon.

Limitations in affordable storage also pose a major challenge, as does the I/O capacity of ever larger hard disks. In addition, network capacity will probably continue to increase at the required level, but the ability to use it efficiently will need a closer integration with applications. This will require developments in the areas of software to support distributed computing (data and workload management, software distribution and data access) and an increasing awareness of the extremely hierarchical view of data, from long latency tape access and medium-latency network access through to the CPU memory hierarchy.

Taking advantage of these new architectures and programming paradigms will be critical for HEP to increase the capacity of our code to do physics efficiently and to meet the processing challenges of the future. Some of this work will be focused on re-optimised implementations of existing algorithms. This will be complicated by the fact that much of our code is written for the much simpler model of serial processing and without the software engineering needed for sustainability. Proper support for taking advantage of concurrent programming techniques (such as task or thread based programming, as well as vectorised SIMD instructions) through frameworks and libraries, will be essential, as the majority of code will still be written by physicists. Other approaches should examine new algorithms and techniques, including highly parallelised code that can run on GPGPUs or the use of machine learning techniques to replace computationally expensive pieces of simulation or pattern recognition. The ensemble of computing work that is needed by the experiments must remain sufficiently flexible to take advantage of different architectures that will provide computing to HEP in the future. In particular, use of high performance computing sites, which may run with very particular constraints, will very likely be a requirement for the community.

These technical challenges are accompanied by significant human challenges. Software is written by many people in the collaborations, with varying levels of expertise, from a few experts with precious skills to novice coders. Effective mechanisms for incorporating contributions, particularly from novices, will be needed. This implies organising training in effective coding techniques and providing excellent documentation, examples and support. Although it is inevitable that some developments will remain within the scope of a single experiment, tackling the software problems coherently as a community will be critical for achieving success in the future. This will range from sharing knowledge of techniques and best practice to establishing common libraries and projects that will provide generic solutions to the community. Writing code that supports a wider subset of the community than just a single experiment presents a greater challenge, but the potential benefits are huge.

Particle physics is no longer alone in facing these massive data challenges. Experiments in other fields, from astronomy to genomics, will produce huge amounts of data in the future and will need to overcome the same challenges that we face: massive data handling and efficient scientific programming. Establishing links with these fields has already started. Additionally, interest from the computing science community in solving these data challenges exists and mutually beneficial relationships would be possible where there are genuine research problems that are of academic interest to that community and provide practical solutions to ours. The efficient processing of massive data volumes is also a challenge faced by industry, in particular the internet economy, which developed novel and major new technologies, under the banner of *Big Data*, that may be applicable to our use cases.

Establishing a programme of investment in software for the HEP community, with a view to ensuring effective and sustainable software for the coming decades, will be essential to allow us to reap the physics benefits of the multi-exabyte data to come. It was in recognition of this fact that the HSF itself was set up and already works to promote these common projects and community developments [HSF2015].

3 Programme of Work

In the following we describe the programme of work being proposed for the range of topics covered by the CWP working groups. We summarise the main specific challenges each topic will face, describe current practices and propose a number of R&D tasks that should be undertaken in order to meet the challenges. R&D tasks are grouped in two different timescales: short term (by 2020, in time for HL-LHC Computing TDRs of ATLAS and CMS) and longer term actions (by 2022, to be ready for testing or deployment during LHC Run 3).

3.1 Conditions Databases

Scope and Challenges

Conditions data is defined as the non-event data required by data-processing software to correctly simulate, digitize or reconstruct the raw detector event data. The non-event data discussed here consists mainly of detector calibration and alignment data, with some additional data describing the detector configuration, the machine parameters and from the detector control system.

Conditions data is different from event data in many respects, but one of the important differences is that its volume scales with time rather than with the luminosity. As a consequence its growth is limited, as compared to event data: conditions data volume is expected to be at the terabyte scale and the update rate is modest (1Hz). However, conditions data can be used by offline jobs running on a very large distributed computing infrastructure, with tens of thousands of jobs that may try to access the conditions data at the same time, leading to a very significant rate of reading (typically O(10) kHz).

To successfully serve such rates, some form of caching is needed, either by using services such as web proxies (CMS and ATLAS use Frontier) or by delivering the conditions data as files distributed to the jobs. For the latter approach, CVMFS is an attractive solution due to its embedded caching features and its advanced snapshotting and branching features. ALICE have made some promising tests and will move forward in this direction; NA62 have also decided to adopt this solution. However, one particular challenge to be overcome with the filesystem approach is to design an efficient mapping of conditions data and metadata to files in order to use the CVMFS caching layers efficiently; CVMFS caches file objects, thus one big file containing all the conditions will defeat caching.

Efficient caching is especially important in order to support the high reading rates that will be necessary for ATLAS and CMS experiments starting with Run 3. For these experiments, a subset of the conditions data is linked to the continuously decreasing luminosity, leading to an interval granularity of the order of a minute. In ATLAS, the current COOL-based conditions data infrastructure cannot handle this properly, impacting the quality of the reconstruction.

Another important challenge is ensuring the long-term maintainability of the conditions database infrastructure. The initial approach taken by several LHC experiments was based on COOL, a rather fat layer that provides a high-level database-like API hiding the back-end details and enforcing a schema. However, this led to the development of a complex product, suffering from inflated requirements, making optimisation difficult and preventing efficient client-side caching. COOL has been difficult to maintain, with too many tradeoffs to satisfy properly all use cases. Based on this experience, there is now a consensus that there should be less enforcement of a common schema, but rather that the client API should support efficient serialisation/deserialisation of objects coupled with effective caching, and let the object internal structure be managed by clients. In this way, it is possible to adopt well-established open-source products rather than rely on a number of HEP-specific tools. CMS has already started to explore this path by adopting a Boost serialisation of C++ objects. A specific challenge that must be addressed with this solution is the long-term preservation of the data, as the ability to interpret the data may be bound to one particular version of the software. The approach taken to implementing serialisation may also have drawbacks if there is a need to support multiple programming languages, at they tend to be language-specific.

With such an approach for the client, it should be possible to leverage technologies like REST interfaces to simplify insertion and read operations and make them very efficient to reach the rate levels foreseen. Also to provide a resilient service to jobs who depend on it, the client should be able to use multiples proxies or servers to access the data.

One conditions data challenge may be linked to the use of an event service, as ATLAS is doing currently to use efficiently HPC facilities for event simulation or processing. The event service allows to better use resources that may be volatile by allocating and bookkeeping the work done not at the job granularity, but at the event granularity. This reduces the possibility for optimising the conditions data access at the job level and may lead to an increase pressure on the conditions data infrastructure. This approach is still at an early stage and more experience is needed to better appreciate the exact impact on the conditions data.

Current Practices

The data model for conditions data management is an area where the experiments have converged on something like a best common practice. A global tag is the top-level configuration of all conditions data. For a given detector subsystem and a given interval of validity, a global tag will resolve to one, and only one, conditions data payload. The global tag resolves to a particular system tag via the global tag map table. A system tag consists of many intervals of validity or entries in the IOV table. Finally, each entry in the IOV table maps to a payload via its unique hash key in the payload table. A relational database is a good choice for implementing this design. One advantage of this approach is that a payload has a unique identifier, its hash key, and this identifier is the only way to access it. All other information, such as tags and IOV, is metadata used to select a particular payload. This allows a clear separation of the payload data from the metadata and may allow use of a different backend technology to store the data and the metadata. This has potentially several advantages:

- Payload objects can be cached independently of their metadata, using the appropriate technology, without the constraints linked to metadata queries.
- Conditions data metadata are typically small compared to the conditions data themselves, which makes it easy to export them as a single file using technologies like SQLite. This may help in particular for long-term data preservation.
- IOVs, being independent of the payload, can also be cached on their own.

A recent evolution is to move to an online full reconstruction, where the calibrations and alignment are applied in the HLT. This is currently being tested by ALICE and LHCb who will adopt it as their base design in Run 3. This will reduce the need to access conditions data from offline workloads, so an high performance caching infrastructure is not required.

Research and Development programme

R&D actions related to Conditions databases are already in progress and all the activities described below should be completed by 2020. This will provide valuable input for the future HL-LHC TDRs and allow these services to be deployed during Run 3 to overcome the limitations seen in today's solutions.

- File-system view of conditions data for analysis jobs: study how to leverage advanced snapshotting/branching features of CVMFS for efficiently distributing conditions data as well as ways to optimise data/metadata layout in order to benefit from CVMFS caching. Prototype production of the file-system view from the conditions database.
- Identify and evaluate industry technologies that could replace HEP-specific components.
- Migrate current implementations based on COOL to the proposed REST-based approach; study how to avoid moving too much complexity on the client side, in particular for easier adoption by subsystems, e.g. possibility of common modules/libraries.

3.2 Data Analysis and Interpretation

Scope and Challenges

HEP answers scientific questions by studying the observations of experiment detectors and their simulations. The final stages of analysis are usually undertaken by small groups, or even individual researchers. The baseline analysis model utilises successive stages of data reduction, finally analysing a compact dataset with quick real-time iterations. This approach aims at exploiting the maximum possible scientific potential of the data whilst minimising the “time to insight” for a large number of different analyses performed in parallel. It is a complicated product of diverse criteria ranging from the need to make efficient use of computing resources to the management styles of the experiment collaborations. Any analysis system also has to be elastic enough to cope with, e.g., deadlines imposed by conference schedules. Future analysis models must adapt to the massive increases in data taken by the experiments, while retaining this essential “time to insight” optimisation. This problem needs tackled in the context of the hardware evolution discussed elsewhere in this document, but also taking into account the need for an analysis system to be elastic enough to cope with deadlines

Over the past 20 years the HEP community has developed and gravitated around a single analysis ecosystem based on ROOT [Brun1996]. This software ecosystem currently dominates HEP analysis and impacts the full event processing chain, providing foundation libraries, I/O services, etc. It gives an advantage to the HEP community, as compared to other science disciplines, in that it provides an integrated and validated toolkit. This lowers the hurdle to start an analysis, enabling the community to talk a common analysis language, as well as making common improvements as additions to the toolkit quickly become available to the whole community.

However, the emergence and abundance of alternative and new analysis components and techniques coming from industry and open source projects is a challenge for the HEP analysis software ecosystem. The HEP community is clearly very interested in using these tools together with established components in an interchangeable way. The main challenge will be to enable new open source tools to be plugged in dynamically to the existing ecosystem and to provide mechanisms to allow the existing and new components to interact and exchange data efficiently. In the longer term the challenge will be to develop a comprehensive set of “bridges” and “ferries” between the HEP analysis ecosystem and the industry analysis tool landscape, where a “bridge” enables the ecosystem to use an open source analysis tool and a “ferry” allows use of data from the ecosystem in the tool, and vice versa.

The maintenance and sustainability of the current analysis ecosystem also presents a major challenge. It already supports a large number of use cases and integrates and maintains a wide variety of components. Development of new components and maintenance of existing ones have to be prioritised to fit into the available effort envelope, which is provided by a few institutions and is less distributed across the community. Legacy and less used parts of the ecosystem are hard to retire and their continued support strains the available effort. New policies are needed to minimise this effort by retiring little used components from integration and validation efforts, where individuals wishing to continue using retired components will have to take over their maintenance .

Current Practices

Methods for analysing the data at the LHC experiments have been developed over the years and successfully applied to produce physics results during Run 1 and Run 2. Analysis at the LHC experiments typically starts with users running code over centrally-managed data that is of O(100kB/event) and contains all of the information required to perform a typical analysis leading to publication. The most common approach to analysing data is through a campaign of *data reduction* and *refinement*, ultimately producing flat ntuples and histograms used to make plots and tables from which physics inference can be made. The current centrally-managed data is typically too large (e.g., hundreds of TB for LHC Run 2 data) to be delivered locally to the user. An oft stated aim of the data reduction steps is to arrive at a dataset that "can fit on one's laptop", in order to facilitate low-latency, high-rate access to a manageable amount of data during the final stages of an analysis. Creating and retaining intermediate datasets produced by data reduction campaigns, bringing and keeping them "close" to the analysers, is designed to minimise latencies and risks related to resource contention.

There has been a huge investment in using C++ for performance-critical code, in particular in event reconstruction and simulation, and this will continue in the future. However, for analysis applications, Python has emerged as the language of choice in the data science community, and its use continues to grow within the HEP community. Python is highly appreciated for its ability to support fast development cycles and for its ease-of-use, and it offers an abundance of well-maintained and advanced software packages. Experience shows that the simpler interfaces and code constructs of Python could reduce the complexity of analysis code and therefore contribute to decreasing the "time to insight" for HEP analyses. The ROOT team have a highly novel way of binding ROOT, C++ and Python through PyROOT, however, this does not quite reach the ease of use of native Python modules. Increased HEP investment is needed to allow Python become a first class supported language.

Reproducibility is the cornerstone of scientific results and while HEP does not face a reproducibility crisis, it is currently difficult to repeat most HEP analyses after they have been completed. This difficulty mainly arises due to the number of scientists involved, the number of steps in a typical HEP analysis workflow, and the complex ecosystem of software that HEP analyses are based on. Analysis preservation and reproducibility strategies are described in the Data and Software Preservation section, and reproducibility needs to be considered in all new approaches under investigation to become a fundamental component of the system as a whole.

One weak area is infrastructure that can represent the many-to-many mapping between all the different aspects of an analysis. These can include publications, logical labels for the event selection defining signal and control regions, data products associated with the application of those event selections to specific datasets, the theoretical models associated to simulated datasets, the multiple implementations of those analyses from the experiments and theoretical community created for the purpose of analysis interpretation, and the results of those interpretations. Some experiment-specific services, like ATLAS AMI, address part of this problem. As the protocol for (re)interpretation can be clear and narrowly scoped, it is possible to offer it as a experiment-agnostic service. This type of activity lends itself to the Science Gateway [SciGateway] concept, which allow science and engineering communities to access shared data, software, computing services, instruments, educational materials, and other resources specific to their disciplines. Such reinterpretation services have been foreseen for several years, and now most of the necessary infrastructure is in place to create it [RECAST]. Such an interpretation service would greatly enhance the physics impact of the LHC and also enhance the legacy of the LHC well into the future.

Research and Development Programme

In the following we describe initiatives that focus on studying new analysis models that build on the experience of the past. One new model of data analysis, developed outside of HEP, maintains the concept of sequential ntuple reduction but mixes interactivity with batch processing. Apache Spark itself is the leading contender for this type of analysis, as it has a well developed ecosystem with many third-party tools developed by industry, however, it is the *style* of analysis workflow that we are distinguishing here rather than the specific technology present today. Other emerging products are TensorFlow, Dask, Pachyderm, and Thrill. A Spark-like analysis facility would be a shared resource for exploratory data analysis and batch submission. The primary advantage that these software products introduce is in simplifying the user's access to data, lowering the cognitive overhead of setting up and running parallel jobs. Spark itself is hard to interface with C++, but this might be alleviated by projects such as ROOT's TDataFrame, which presents a Spark-like interface in ROOT, and may allow for more streamlined interoperability.

An alternative approach would be to perform *fast querying* of centrally-managed data and compute remotely on the queried data to produce the analysis products of interest. The analysis workflow would be accomplished without focus on persistence of data traditionally associated with data reduction, although transient data may be generated in order to efficiently accomplish this workflow and optionally could be retained to facilitate an analysis "checkpoint" for subsequent execution. In this approach, the focus is on obtaining the analysis end-products in a way that does not necessitate a data reduction campaign and associated provisioning of resources.

Further optimisation of analysis code could be gained by switching to a functional or declarative programming model. This would allow scientists to express the intended data transformation as a query on data. Instead of having to define and control the "how", the analyst would declare the "what" of their analysis, essentially removing the need to define the event loop in an analysis and leave it to underlying services and systems to optimally iterate over events. Analogously to how programming in C++ abstracts implementation features compared to programming in assembler, it appears that these high-level approaches will allow abstraction from the underlying implementations, allowing the computing systems more freedom in optimising the utilisation of diverse forms of computing resources. R&D is already under way (e.g. TDataFrame in ROOT) and this needs to be continued with the ultimate goal of establishing a prototype functional or declarative programming language model.

The I/O performance for iterating over events becomes one of the driving factors in minimising the “time to insight” during data analysis. In fact there are many file format standards used by a wealth of data analytics tools in other science fields and industry, but the HEP community feels that currently ROOT is best suited to fulfil the community’s needs. Disk space requirements are usually a key feature of the experiment computing models, as disk is the most expensive hardware component. The community uses data compression techniques extensively to minimise these storage costs. This reduces the performance of purely iterating over events because of the necessary decompression. To improve our ability to analyse much larger datasets than today, R&D will be needed to investigate file formats, compression algorithms, and new ways of storing and accessing data for analysis.

Towards HL-LHC we envisage dedicated data analysis facilities for experimenters, offering an extendable environment that can provide fully functional analysis capabilities. A “primitive” version of such analysis facilities is currently provided at CERN, Fermilab and elsewhere. However, for HL-LHC, such dedicated Analysis Facilities would provide a complete system engineered for latency-optimisation and stability. Prototyping work is needed to investigate optimisation of the storage systems and to facilitate the utilisation of new additional storage layers, such as SSD storage and NVRAM-like storage. This should include a fresh look at the concept of “virtual data”, optimising the choice between storing versus re-computing data products. Another area that needs attention is access to non-event data for analysis, including cross section values, scale factors, tagging efficiencies, etc.

The following R&D programme lists the tasks that need to be accomplished in order to realise the objectives described above.

By 2020:

- Enable new open source tools to be plugged in dynamically to the existing ecosystem and provide mechanisms to dynamically exchange parts of the ecosystem with new components.
- Prototype a low-latency response high-capacity analysis facility incorporating fast caching technologies to explore a query-based analysis approach.
- Finalize full support of Python in our ecosystem and evolve a policy for ensuring long term maintenance and sustainability.
- Establish a schema for the analysis database.
- Develop a functional prototype of an Interpretation Gateway, integrating the analysis facility, analysis preservation infrastructure, data repositories, and recasting tools.
- Prototype a comprehensive set of “bridges and ferries” (as defined above).

- Develop a prototype functional or declarative programming language model.

By 2022:

- Analysis facility: evaluate chosen architectures and verify design or provide input for corrective actions (Run 3 data should be used); a blueprint for remaining developments and system design should become available, in time for deployment.
- Interpretation gateway: evaluate design or provide input for corrective actions to enable an LHC legacy (re)interpretation gateway.

3.3 Data and Software Preservation to Enable Reuse

Scope and Challenges

Given the very large investment in particle physics experiments, it is incumbent upon physicists to preserve the data and the knowledge that leads to scientific results in a manner such that this investment is not lost to future generations of scientists. For preserving “data”, at what ever stage of production, many of the aspects of the low level bit-wise preservation have been covered by the Data Preservation for HEP group [DPHEP]. The word “knowledge” encompasses the more challenging aspects of recording processing and analysis software, documentation, and other components necessary for reusing a given dataset. Preservation of this type can enable new analyses on older data, as well as a way to revisit the details of a result after publication. The latter can be especially important in resolving conflicts between published results, applying new theoretical assumptions, or evaluating different theoretical models.

Preservation enabling reuse can offer tangible benefits within a given experiment. The preservation of software and workflows such that they can be shared enhances collaborative work between analysts and analysis groups, provides a way of capturing the knowledge behind a given analysis during the review process, enables easy transfer of knowledge to new students or analysis teams, and could establish a manner by which results can be generated automatically for submission to central repositories, such as HEPData. Preservation within an experiment can provide ways of re-processing and re-analysing data that could have been collected more than a decade earlier. Providing such immediate benefits greatly incentivises the adoption of data preservation in experiment workflows, which makes it particularly desirable.

A final series of motivations comes from the potential re-use by others outside of the HEP experimental community. Significant outreach efforts bringing the excitement of analysis and discovery to younger students has been enabled by the preservation of experimental data and software in an accessible format. Many examples also exist of phenomenology papers reinterpreting the results of a particular analysis in a new context. This has been extended further with published results based on the re-analysis of processed data by scientists outside of the collaborations. Engagement of external communities, such as machine learning specialists, can be enhanced by providing the capability to process and understand low-level HEP data in portable and relatively platform-independent packages. This allows external users direct access to the same tools and data as the experimentalists working in the collaborations. Connections with industrial partners, such as those fostered by CERN OpenLab, can be facilitated in a similar manner.

Preserving the knowledge of analysis, given the extremely wide scope of how analysts do their work and experiments manage their workflows, is far from easy. The level of reuse that is applicable needs to be identified and so a variety of preservation systems will probably be appropriate given the different preservation needs between large central experiment workflows and the work of an individual analyst. The larger question is to what extent common low-level tools can be provided that address similar needs across a wide scale of preservation problems. These would range from capture tools, that preserve the details of an analysis and its requirements, to ensuring that software and services needed for a workflow would continue to function as required.

Current Practices

Each of the LHC experiments has adopted a data access and/or data preservation policy, all of which can be found on the CERN Open Data Portal [ODP]. All of the LHC experiments support public access to some subset of the data in a highly-reduced data format for the purposes of outreach and education. CMS has gone one step further, releasing substantial datasets in an AOD format that can be used for new analyses. The data release includes simulated data, virtual machines that can instantiate the added analysis examples, and extensive documentation [CMS-OpenData]. ALICE has promised to release 10% of their processed data after a five-year embargo and has released 2010 data at this time [ALICE-OpenData]. LHCb has promised to release 50% of the processed data and associated software after five years, contingent on having sufficient manpower to produce this. So far, that has not been the case. ATLAS has chosen a different direction for data release: data associated with journal publications is made available and ATLAS also strives to make additional material related to the paper available that allows a reinterpretation of the data in the context of new theoretical models [ATLAS2015a]. ATLAS is also exploring how to provide the capability for reinterpretation of searches in the future via a service such as RECAST, allowing theorists to evaluate the sensitivity of a published analysis to a new model they have developed.

None of the LHC experiments have made recent public statements addressing the new capabilities of the CERN Analysis Portal and whether or not some use of it will be required (or strongly encouraged). All of them support some mechanisms for internal preservation of the knowledge surrounding a physics publication [Shears2017].

Research and Development Programme

There is a significant programme of work already happening in the data preservation area. The goals presented here should be orchestrated in conjunction with projects conducted by the R&D programmes of other working groups, since the questions addressed are common. Goals to address to provide input for Computing TDRs are:

- Develop prototype analysis ecosystem(s), including embedded elements for the capture of preservation information and metadata and tools for the archiving of this information. This should include an early demonstration of an analysis preservation portal with a working UI.
- Demonstrate the capability to provision and execute production workflows for LHC experiments that are composed of multiple independent containers.

- Collection of analysis use cases and elements that are necessary to preserve in order to enable re-use and to ensure these analyses can be captured in developing systems. This should track analysis evolution towards possible “big data” environments and determine any elements that are difficult to capture, spawning further R&D.
- Evaluate any limits of container technologies in the preservation arena.
- Develop prototypes for the preservation and validation of large-scale production executables and workflows.

This would then lead naturally to deployed solutions that support data preservation at some point during LHC Run 3, in particular an analysis ecosystem that enables reuse for any analysis that can be conducted in the ecosystem and a system for the preservation and validation of large-scale production workflows.

3.4 Data Organisation, Management and Access

The reach of data-intensive experiments is limited by how fast data can be accessed and digested by computational resources and both technology and large increases in data volume require new computational models [Butler2013]. Extending current data handling methods and methodologies is expected to be intractable in the HL-LHC era. The development and adoption of new data analysis paradigms gives the field, as a whole, a window in which to adapt our data access and data management schemes to ones that are more suited and optimally matched to a wide range of advanced computing models and analysis applications. This type of shift has the potential for enabling new analysis methods and allowing for an increase in scientific output.

Scope and Challenges

The LHC experiments currently provision and manage about an Exabyte of storage, approximately half of which is archival, and half is traditional disk storage. The storage requirements per year are expected to jump by a factor close to 10 for the HL-LHC. This growth rate is faster than projected Moore's Law gains and will present major challenges. Storage will remain one of the visible cost drivers for HEP computing and the projected increase in the cost of the computational resources will also be huge. The combination of storage and analysis computing costs may restrict scientific output and the potential physics reach of the experiments. Thus new techniques and algorithms are likely to be required.

In devising experimental computing models for this era, many factors have to be taken into account. In particular, the increasing availability of very high-speed networks, which may reduce the need for CPU and data co-location, need to be examined. Such networks may allow for more extensive use of data access over the wide-area network (WAN), which may provide failover capabilities, global and federated data namespaces, and will have an impact on data caching. Shifts in data presentation and analysis models, such as a potential move to event-based data streaming from the more traditional dataset-based or file-based data access, will be particularly important for optimising the utilisation of opportunistic computing cycles on HPC facilities, commercial cloud resources, and campus clusters, and can potentially resolve currently limiting factors such as job eviction.

The three main challenges for data in the HL-LHC era can be summarized as follows:

- The HL-LHC era will significantly increase both the date rate and the data volume. The computing systems will need to handle this without significant cost increases and within evolving storage technology limitations.
- The significantly increased computational requirements for the HL-LHC era will also place new requirements on data. Specifically, the use of new types of compute resources (cloud, HPC) with different dynamic availability and characteristics are used will require more dynamic data management and access systems.
- Applications employing new techniques, such as machine learning training or high rate

> data query systems, will likely be employed to meet the computational constraints and to extend the physics reach of the HL-LHC. These new applications will place new requirements on how and where data is accessed and produced. Specific applications, such as training for machine learning, may require use of specialized processor resources such as GPUs, placing further requirements on data.

The projected event complexity of data from future LHC runs with high pileup and from high resolution liquid argon detectors at DUNE will require advanced reconstruction algorithms and analysis tools to understand. The precursors of these tools, in the form of new machine learning paradigms and pattern recognition algorithms, already are proving to be drivers for the CPU needs of the HEP community. As these techniques continue to grow and blossom, they will place new requirements on the computational resources that need to be leveraged by all of HEP. The storage systems that are developed, and the data management techniques that are employed will need to directly support this wide range of computational facilities, and will need to be matched to the changes in the computational work, so as not to impede the improvements that they are bringing.

As with CPU, the landscape of storage protocols accessible to us is trending towards heterogeneity. The ability to leverage new storage technologies as they become available into existing data delivery models is a challenge that we must be prepared for. This also implies that HEP experiments should be prepared to leverage “tactical storage”, i.e., storage that becomes most cost-effective as it becomes available (e.g., from a cloud provider) and have a data management and provisioning system that can exploit such resources at short notice. Volatile data sources would impact many aspects of the system: catalogs, job brokering, monitoring and alerting, accounting, the applications themselves.

On the hardware side, R&D is needed in alternative approaches to data archiving to determine the possible cost/performance tradeoffs. Currently, tape is extensively used to hold data that cannot be economically made available online. While the data is still accessible, it comes with a high latency penalty, limiting possible analysis. We suggest investigating either separate direct access-based archives (e.g. disk or optical) or new models that overlay online direct access volumes with archive space. This is especially relevant when access latency is proportional to storage density. Either approach would need to also evaluate reliability risks and the effort needed to provide data stability.

Cost reductions in the maintenance and operation of storage infrastructure can be realised through convergence of the major experiments and resource providers on shared solutions. This does not necessarily mean promoting a monoculture, as different solutions will be adapted to certain major classes of use-case, type of site or funding environment. Indeed, there will always be a judgement to make on the desirability of using a variety of specialised systems, or abstracting the commonalities through a more limited but common interface. Reduced costs and improved sustainability will be further promoted by extending these concepts of convergence beyond HEP and into the other large-scale scientific endeavours that will share the infrastructure in the coming decade. Efforts must be made as early as possible, during the formative design phases of such projects, to create the necessary links.

Finally, any and all changes undertaken must not make the ease of access to data any worse than it is under current computing models. We must also be prepared to accept the fact that the best possible solution may require significant changes in the way data is handled and analysed. What is clear is that what is being done today will not scale to the needs of HL-LHC.

Current Practices

The original LHC computing models were based on simpler models used before distributed computing was a central part of HEP computing. This allowed for a reasonably clean separation between three different aspects of interacting with data, namely data organisation, data management and data access.

- *Data organisation* is essentially how data is structured as it is written. Most data is written in flat files, in ROOT format, typically with a column-wise organisation of the data. The records corresponding to these columns are compressed. The internal details of this organisation are visible only to individual software applications.
- The key challenge for *data management* was the transition to the use of distributed computing in the form of the grid. The experiments developed dedicated data transfer and placement systems, along with catalogs, to move data between computing centers. To first order the computing models were rather static: data was placed at sites and the relevant compute jobs were sent to the right locations. Applications might interact with catalogs or, at times, the workflow management systems does this on behalf of the applications.
- Concerning *data access*, various protocols are used for direct reads (rfio, dcap, xrootd, etc.) with a given computer center and/or explicit local stage-in and caching for read by jobs. Application access may use different protocols than those used by data transfers between sites.

Before the LHC turn-on and in the first years of the LHC, these three areas were to first order optimised independently. Many of the challenges were in the area of “Data Management” (DM) as the Worldwide LHC Computing Grid was commissioned. As the LHC computing matured through Run 1 and Run 2, the interest has turned to optimisations spanning these three areas. For example, the recent use of “Data Federations” mixes up Data Management and Access. As we will see below, some of the foreseen opportunities towards HL-LHC may require global optimisations.

Thus in this document we take a broader view than traditional “DM”, and consider the combination of “Data organisation, Management and Access” (DOMA) together. We believe that this full picture of data needs in HEP will provide important opportunities for efficiency and scalability as we enter the many-exabyte era.

Research and Development Programme

In the following we describe tasks that will need to be carried out in order to demonstrate that the increased volume and complexity of data expected over the coming decade can be stored, accessed and analysed at an affordable cost.

1. Event-based granularity will be studied to see whether it can be implemented efficiently, scalably and in a cost-effective manner for all applications making use of event selection, to see whether it offers an advantage over current file-based granularity. The following tasks should be completed by 2020:
    a. Quantify the impact on performances and resource utilisation (storage, network) for the main type of access patterns (simulation, reconstruction, analysis).
    b. Assess impact on catalogs and data distribution.
    c. Assess whether event-granularity makes sense in object stores that tend to require large chunks of data for efficiency.
    d. Test for improvement in recoverability from preemption, in particular when using cloud spot resources and/or dynamic HPC resources.
2. We will seek to derive benefits from data organisation and analysis technologies adopted by the big-data world. A proof-of-concept that involves the following tasks needs to be established by 2020 to allow full implementations to be made in the years that follow.
    a. Study the impact of column-wise vs. row-wise organisation of data on the performance of each kind of access.
    b. See whether map-reduce, Spark-like analysis, and their functional or declarative interfaces, can be adapted to HEP analysis needs.
    c. Evaluate just-in-time decompression schemes and mappings onto hardware architectures considering the flow of data, from spinning disk to memory and application
3. Discover the role data caching can play in order to use compute resources effectively and the technologies that can be used. The following tasks should be completed by 2020:
    a. Quantify the benefit of caching for the main use cases i.e. reconstruction, analysis, and simulation.
    b. Assess the benefit of caching for Machine Learning-based applications, in particular for the learning phase.

In the longer term it is aimed to also study the benefits that can be derived from using different approaches to the way HEP is currently managing its data delivery systems. Two different content delivery methods will be studied, namely Named Data Networking (NDN) and Content Delivery Networks (CDN).

4. Study how to minimise HEP infrastructure costs by exploiting varied quality of service from different storage technologies. In particular, study the role that opportunistic/tactical storage can play as well as different archival storage solutions. A proof-of-concept should be made by 2020, with a full implementation to follow in the following years.
5. Establish how to globally optimise data access latency, with respect to efficiency of using CPU, at a sustainable cost. This involves studying the impact of concentrating data in fewer, larger locations (“data-lake” approach) and making increased use of opportunistic compute resources located further from the data. Again, a proof-of-concept should be made by 2020, with a full implementation in the following years if successful.

3.5 Data-Flow Processing Framework

Scope and Challenges

Frameworks in High Energy Physics are used for the collaboration-wide data processing tasks of reconstruction, simulation and triggering, as well as other tasks that subgroups of the collaboration are responsible for, such as detector alignments. Providing framework services and libraries that will satisfy the compute and data needs for HL-LHC experiments and the Intensity Frontier experiments, while maintaining our efficient exploitation of increasingly heterogeneous resources, is a huge challenge.

To fully exploit the potential of the modern processors, HEP data processing frameworks need to allow for the parallel execution of reconstruction or simulation algorithms on multiple events simultaneously. Frameworks face the challenge of handling the massive parallelism and heterogeneity that will be present in future compute facilities, including multi and many-core systems, GPGPUs, Tensor Processing Units (TPU), tiered memory systems, each integrated with storage and high-speed network interconnects. Efficient running on heterogeneous resources will require a tighter integration with the computing model's higher-level systems of workflow and data management. Common developments, which are in principle desirable, are hampered by many decades of legacy work and, to be successful, must include excellent integration with the wider ecosystem of development, deployment, and runtime components. Experiment frameworks must also successfully integrate and marshall other HEP software that may have its own parallelisation model, such as event generators and Geant simulation.

Developing and evolving our frameworks has to be done recognising the needs of the different stakeholders in the system. This includes physicists who are writing processing algorithms for triggering, reconstruction or analysis; production managers who need to define processing workflows over massive datasets; facility managers, who require their infrastructures to be used effectively; and funding agencies, who may mandate security requirements or common projects.

Current Practices

Although most frameworks used in HEP share common concepts, there are a number of different implementations, some of which are shared between experiments. The Gaudi framework was originally developed by LHCb, but is also used by ATLAS. CMS use their own CMSSW framework, which was forked to provide the art framework for the Intensity Frontier experiments. BELLE II use basf2. The FAIR experiments use FairROOT, closely related with ALICE's AliROOT. The FAIR and ALICE teams are commonly developing a new framework, O2. Almost all of these frameworks have evolved, and continue to evolve, to incorporate concurrency.

Each framework has a processing model, which provides the means to execute and apportion work. Mechanisms for this are threads, tasks, processes and interprocess communication. The different strategies used reflect different tradeoffs between constraints in the programing model, efficiency of execution, and ease of adapting to inhomogeneous resources. These concerns also reflect two different behaviours: maximising throughput, where it is most important to maximise the number of events that are processed by a given resource; and minimising latency, where the primary constraint is on how long it takes to calculate an answer for a particular datum.

Research and Development programme

Initial R&D into frameworks should firstly properly review the existing technology that is used and establish a set of well defined architectural concepts that can be used as a basis for future design studies. Some study of libraries used for concurrency, and their likely evolution, will provide valuable input as well as improving understanding of how to integrate simulation and generator frameworks. Functional programming, as a key design idea, is worthy of particular investigation, along with the benefits offered by a domain specific language that could describe how physics data processing has to be undertaken. Community meetings and workshops, along the lines of the original Concurrency Forum, are envisaged to help foster collaboration in this work [ConcurrencyForum].

After these baselines are established, focus should turn to prototype and demonstrator projects. They will need to be completed by 2020 to be able to inform the HL-LHC Computing TDRs and should demonstrate advances over what is currently deployed. Some common work should be done on how frameworks ought to evolve to incorporate features of functional programming, scheduling across heterogeneous resources, and addressing both necessary data model changes and I/O handling. This must also include strategies for integration with workload management and incorporation of the evolution of processing, storage and networks at facilities. Understanding what, if any, framework consolidation between experiments can be achieved, at what cost and benefits, and how common components might look, is a major goal. A fairly continuous process of technology watch needs to happen to steer efforts towards adopting the most effective solutions. Sharing of ideas and developments through meetings and workshops remains a vital activity.

By 2022, based on the common tools and components identified earlier in the process, work on production quality framework libraries for use by multiple experiments should be advanced. Good integration with the workload management systems and facilities should be achieved, including running in the cloud and at HPCs. There will be independent progress on parallelization strategies and implementations. On the time scale of 5 years we anticipate at least one major paradigm shift will take place, which can not be incorporated by continuous adjustment alone. A future planning workshop at this time should transform the results of the R&D activities into a full development and deployment project plan for HL-LHC running.

3.6 Detector Simulation

Scope and Challenges

The experimental programmes planned in the next decade are driving developments in the simulation domain; they include the High Luminosity LHC project (HL-LHC), neutrino and muon experiments, and studies towards future colliders such as the Future Circular Collider (FCC) and the Compact Linear Collider (CLIC). The requirement of improving precision in the simulation implies production of larger Monte Carlo samples, which scale with the number of real events recorded in future experiments, and this places an additional burden on the computing resources that will be needed to generate them. The diversification of the physics programmes also requires new and improved physics models.

To match these new requirements with foreseen available resources, the speed of simulation codes need to be improved by an order of magnitude. This is a huge challenge. The gains that can be made by speeding up critical elements of the common simulation toolkit (Geant4) can be leveraged for all applications that use it and therefore it is well worth the investment of effort needed to achieve it. The main R&D challenges to be addressed if the required physics and software performance goals are to be achieved are:

- reviewing the physics models’ assumptions, approximations and limitations in order to achieve higher precision, and to extend the validity of models up to FCC energies on the order of 100 TeV;
- redesigning, developing, commissioning detector simulation toolkits to be more efficient when executed on emerging computing architectures like Intel Xeon Phi, and GPGPUs where use of SIMD (vectorisation) is vital; this includes porting and optimising the experiments’ simulation applications to allow exploitation of large HPC facilities;
- exploring different fast simulation options, including common frameworks for fast tuning and validation;
- developing, improving and optimising geometry tools that can be shared among experiments to make the modeling of complex detectors computationally more efficient, modular, and transparent;
- developing techniques for background modeling, including contributions of multiple hard interactions overlapping the event of interest in collider experiments (pile-up);
- revisiting digitization algorithms to improve performance and exploring opportunities for code sharing among experiments;
- recruiting, training, retaining human resources in all areas of expertise pertaining to the simulation domain, including software and physics.

Current Practices

The Geant4 detector simulation toolkit is at the core of simulation in almost every HEP experiment. Its continuous development, maintenance, and support to the experiments is of vital importance and needs to be strengthened. New or refined functionality continues to be delivered in the on-going development programme both in physics coverage and accuracy, whilst introducing software performance improvements whenever possible. In addition the Geant4 collaboration is working closely with user communities to enrich the physics models' validation system with data acquired during physics runs and test beam campaigns. The Geant4 simulation toolkit will continue to evolve over the next decade. This evolution may include contributions from various R&D projects as described in the following section.

Physics models are another critical part of the detector simulation and are continuously being reviewed, and in some cases reimplemented, in order to improve accuracy and software performance. Discrepancies between measurements and simulation have been observed in detector response linearity and energy resolution for some particles, as well as in electromagnetic and hadronic shower shapes. Investigations need to be made, in collaboration with experiments, to understand whether their cause can be attributed to a problem in the underlying physics model or to some other cause, such as an issue with the modelling of detector materials.

It is obviously of critical importance that the whole community of scientists working in the simulation domain continue to work together in as efficient a way possible in order to deliver the required improvements. Very specific expertise is required across all simulation domains, such as physics modeling, tracking through complex geometries and magnetic fields, and building realistic applications that accurately simulate highly complex detectors. Continuous support is needed to recruit, train, and retain people with the unique set of skills needed to guarantee the development, maintenance, and support of simulation codes over the long timeframes foreseen in the HEP experimental programme.

Research and Development programme

To meet the challenge of improving the performance by an order of magnitude, an ambitious R&D programme is underway to investigate ways of improving the performance of all components of the simulation software for the longer term. One of the most ambitious elements of this programme is a new approach to managing particle transport, which has been introduced by the GeantV project. The aim is to deliver a multi-threaded vectorised transport engine that has the potential to deliver large performance benefits. Its main feature is track-level parallelisation, bundling particles with similar properties from different events to process them in a single thread. This approach, combined with SIMD vectorisation coding techniques and use of data locality, is expected to yield significant speed-ups, which are being measured in a realistic prototype currently under development. The GeantV *alpha* release, planned end of 2017, will serve as a preview of the new particle transport engine and demonstrate many of its features.

At the same time as this new transport engine is being developed, work is also on-going to exploit parallelisation techniques to improve the performance of the accompanying modules, including geometry, navigation, and the physics models. They are developed as independent modules in such a way that they can also be used together with the current Geant4 transport engine. Of course, when used with Geant4 they will not expose their full performance potential, since transport in Geant4 is sequential, but this allows their full validation and comparison with the existing implementations. The benefit of this approach is that new developments can be delivered as soon as they are available. The new vectorised geometry package (VecGeom), developed as part of GeantV R&D and successfully integrated into Geant4, is an example that demonstrated the benefit of this approach.

The R&D programme summarized here is organized by topics. More details about each activity can be found in the full CWP document.

Simulation Frameworks

- 2019: GeantV *beta* release availability; it is expected to contain enough functionality to build the first real applications. This will allow performance to be measured and give sufficient time to prepare for HL-LHC running. It should include the use of vectorisation in the various components, complete physics modelling for electrons, gammas and positrons, high performance hadronic interactions, still maintaining simulation reproducibility and demonstrating efficient concurrent I/O and multi-event user data management.

Physics Models

- 2020: new implementation of one full set of hadronic physics models for the full LHC energy range and improved physics for liquid Argon detectors. To address the needs of cosmic frontier experiments optical photon transport must be improved and made faster.
- 2022: improved implementation of hadronic cascade and string models with a modular design.

Experiment Applications

- 2020: LHC, Neutrino and Muon experiments to demonstrate an ability to run their detector simulation in multi-threaded mode, using the improved navigation and electromagnetic physics packages. This should bring experiments more accurate physics and an improved performance.
- 2020: early integration of GeantV beta release in the experiments' simulation to measure the benefits. It may include a vectorised version of readout and digitization. CPU intensive applications may be run on HPC systems, benefitting from an increased event throughput.
- 2022: the availability of a production version of the new track-level parallelisation and fully vectorised geometry, navigation, and physics libraries will offer the experiments the option to finalise integration into their frameworks; intensive work will be needed on physics validation and computing performance tests. If successful, the new engine could be in production on the timescale of the start of the HL-LHC run.

Pileup

Backgrounds to hard-scatter events have many components including in-time pileup, out-of-time-pileup, cavern background and beam-gas collisions. All of these components can be simulated, but they present storage and I/O challenges related to the handling of the large simulated min-bias samples used to model the extra interactions. An R&D programme is needed to study different approaches to managing these backgrounds within the next 3 years:

- Real zero-bias events can be collected, bypassing any zero suppression, and overlaid on the fully simulated hard scatters. This approach faces challenges related to the collection of non-zero-suppressed samples or the use of suppressed events, non-linear effects when adding electronic signals from different samples, and sub-detector misalignment consistency between the simulation and the real experiment.

- Another option is to “pre-mix” together the minimum bias collisions into individual events that have the full background expected for a single collision of interest. Experiments will invest effort on improving their pre-mixing techniques, which allow the mixing to be performed at the digitisation level reducing the disk and network usage for a single event. They are also expected to invest in the development of the zero-bias overlay approach by 2020.

Fast Simulation

The work on Fast Simulation is also accelerating with the objective of producing a flexible framework that permits Full and Fast simulation to be combined for different particles in the same event. Various approaches to Fast Simulation are being tried all with the same goal of saving computing time, under the assumption that it is possible to improve time performance without an unacceptable loss of physics accuracy. Machine Learning is one of the techniques being explored in this context.

- 2018: assessment of the benefit of machine learning approach for Fast Simulation.
- 2019: ML-based Fast Simulation for some physics observables.
- 2022: clarify the possible extent of a common Fast Simulation infrastructure applicable to the variety of detector configurations.

Digitization

- 2020: deliver advanced high-performance, SIMD-friendly generic digitization examples that experiments can use as a basis to develop their own code.
- 2022: fully tested and validated optimised digitization code that can be used by the HL-LHC and DUNE experiments.

Pseudorandom Number Generation

The selection of pseudorandom number generators (PRNGs) presents challenges when running on infrastructures with a large degree of parallelism, as reproducibility is a key requirement. HEP will collaborate with researchers in the development of PRNGs, seeking to obtain generators that address better our challenging requirements. Specific milestones are:

- 2020: develop a single library containing sequential and vectorised implementations of the set of state-of-the-art PRNGs, to replace the existing Root and CLHEP implementations.
- 2022: promote a transition to the use of this library to replace existing implementations in ROOT, Geant4 and GeantV.

3.7 Facilities and Distributed Computing

Scope and Challenges

As was outlined in the section on Computing Challenges, huge resource requirements are anticipated for HL-LHC running. These need to be deployed and managed across the Worldwide LHC Computing Grid (WLCG) infrastructure, which has evolved from the original ideas on deployment before LHC data-taking started [MONARC], to be a mature and effective infrastructure that is now exploited by LHC experiments. Currently hardware costs are dominated by disk storage, closely followed by CPU, followed by tape and networking. Naive estimates of scaling to meet HL-LHC needs would indicate that the current system would need almost an order of magnitude more resources than will be available from technology evolution. Even anticipating substantial software improvements, the major challenge in this area is to find the best configuration for facilities and computing sites that makes HL-LHC computing feasible. This challenge is complicated by substantial regional differences in funding models, meaning that any solutions must be sensitive to these local considerations to be effective.

There are a number of changes that can be anticipated in the timescale of the next decade that must be taken into account. There is the increasing need to use highly heterogenous resources. These include the use of high performance computing infrastructures (HPC), which can often have very particular setups and policies that make their exploitation challenging; volunteer computing, which is restricted in scope and unreliable, but can be a significant resource; and cloud computing, both commercial and research, which offer different resource provisioning interfaces and can be significantly more dynamic than directly funded HEP computing sites. In addition, diversity of computing architectures is expected to become the norm, with different CPU architectures as well as more specialized GPUs and FPGAs.

This increasingly dynamic environment for resources, particularly CPU, must be coupled to a highly reliable system for data storage and a suitable network infrastructure for delivering this data to where it will be processed.

In the network domain there are new technology developments, like Software Defined Networks (SDN), that enable user-defined high capacity network paths to be controlled via experiment software and which could help manage these data flows. These new technologies require considerable R&D to prove their utility and practicality. In addition, the networks used by HEP are likely to see large increases in traffic from other science domains that may reduce our ability to dominate the deployed networks to move data in the way that is done today.

Underlying storage system technology will continue to evolve, for example towards object stores, and R&D is also necessary to understand their usability and their role in the HEP infrastructures. There is also the continual problem of assembling inhomogeneous systems and sites into an effective widely distributed worldwide data management infrastructure that is usable by experiments. This is particularly compounded by the scale increases for HL-LHC where multiple replicas of data (for redundancy and availability) become extremely expensive.

Evolutionary change towards HL-LHC is required, as the experiments will continually use the current system. Mapping out a path for migration then requires a fuller understanding of the costs and benefits of proposed changes. A model is needed in which the benefits of such changes can be evaluated, taking into account hardware and human costs, as well as the impact on software and workload performance that in turn leads to physics impact.

Current Practices

While there are many particular exceptions, most resources incorporated into the current WLCG are done so in independently managed sites, usually with some regional organisation structure and mostly offering both CPU and storage. The sites are usually funded directly to provide computing to WLCG, and are in some sense then "owned" by HEP, albeit often shared with others. Frequently substantial cost contributions are made indirectly, for example through funding of energy costs or additional staff effort, particularly at smaller centers. Tape is found only at CERN and the Tier-1s.

Interfaces to these computing resources are defined by technical operations in WLCG. Frequently there are choices that sites can make amongst some limited set of approved options of interfaces. These can overlap in functionality. Some are very HEP specific and recognised as over-complex: work is in progress to get rid of them. The acceptable architectures and operating systems are also defined at the WLCG level (currently x86_64, running Scientific Linux 6 and compatible) and sites can deploy these either directly onto "bare metal" or use an abstraction layer, such as virtual machines or containers.

There are different logical networks being used to connect sites: LHCOPN connects CERN with the Tier-1 centers and a mixture of LHCONE and generic academic networks connect other sites.

Almost every experiment layers its own customised workload and data management system on top of the base WLCG provision, with a few higher level components in common. The pilot job model for workloads is ubiquitous, where a real workload is dispatched only once a job slot is secured. Data management layers aggregate files in the storage systems into datasets and manage experiment-specific metadata. In contrast to the Monarc model, sites are generally used more flexibly and homogeneously by experiments, both in workloads and in data stored. Considerable network resources are consumed in pre-placing data at sites.

In total, WLCG currently provides the experiments with resources distributed at about 170 sites, in 42 countries, which pledge every year the amount of CPU and disk resources they are committed to delivering. The pledge process is overseen by the Resource Scrutiny Group (C-RSG), mandated by the funding agencies to validate the experiment requests and to identify mismatches with site pledges. These sites are connected by 10-100Gb links and deliver approximately 750k CPU cores and 1EB of storage, of which 300PB is disk. More than 200M jobs are executed each day. [Bird2017].

Research and Development programme

The following areas for study are ongoing and will involve technology evaluations, prototyping and scale tests. They will need to be structured to meet the common milestones of informing the HL-LHC Computing TDRs and deploying advanced prototypes during LHC Run 3.

- Understand better the relationship between the performance and costs of the WLCG system and how it delivers the necessary functionality to support LHC physics. This will be an ongoing process, started by the Performance and Costs Working Group, and aims to provide a quantitative assessment for any proposed changes.
- Define the functionalities needed to implement a federated data center concept ("data lake") that aims to reduce the operational cost of storage for HL-LHC and better manage network capacity. This would include necessary qualities of service and options for regionally distributed implementations, including the ability to flexibly respond to model changes in the balance between disk and tape. This work should be done in conjunction with the Data Organisation, Management and Access WG to evaluate the impact for the different access patterns and data organisations envisaged.
- Establish an agreement on the common data management functionality that is required by experiments, targeting a consolidation and a lower maintenance burden. The intimate relationship between the management of elements in storage systems and metadata must be recognised. This needs to address at least the following use cases:
    - processing sites that may have some small disk cache, but do not manage primary data;
    - fine grained processing strategies that may enable processing of small chunks of data, with appropriate bookkeeping support;
    - integration of heterogeneous processing resources, such as HPCs and clouds.
- Investigate more scalable and uniform means of workload scheduling, which incorporate dynamic heterogenous resources and the capabilities of finer grained processing that increases overall efficiency. The optimal scheduling of specialist workloads that require particular resources is clearly required.
- Contribute to the prototyping and evaluation of a quasi-interactive analysis facility that would offer a different model for physics analysis, but would also need to be integrated into the data and workload management of the experiments. This is work to be done in collaboration with the Data Analysis and Interpretation WG.

3.8 Machine Learning

Machine Learning (ML) is a rapidly evolving approach to characterising and describing data with the potential to radically change how data is reduced and analysed. Some applications will qualitatively improve the physics reach of data sets. Others will allow much more efficient use of processing and storage resources, effectively extending the physics reach of the HL-LHC experiments. Many of the activities in this focus area will explicitly overlap with those in the other focus areas, whereas others will be more generic. As a first approximation, the HEP community will build domain-specific applications on top of existing toolkits and ML algorithms developed by computer scientists, data scientists, and scientific software developers from outside the HEP world. Work will also be done to understand where problems do not map onto existing paradigms well and how these problems can be recast into abstract formulations of more general interest.

Scope and Challenges

The world of data science has developed a variety of very powerful ML approaches for classification (using pre-defined categories), clustering (where categories are discovered), regression (to produce continuous outputs), density estimation, dimensionality reduction, etc. Some have been used productively in HEP for more than 20 years, others have been introduced relatively recently. More are on their way and Deep Learning (DL) techniques look very promising for our field. A key feature of these algorithms is that most have open source software implementations that are reasonably well documented. ML has already become ubiquitous in some types of HEP applications: for example, particle identification algorithms that require combining information from multiple detectors to provide a single figure of merit use a variety of Boosted Decision Trees (BDTs) and neural networks.

The abundance of ML algorithms and implementations presents both opportunities and challenges for HEP. Which are most appropriate for our use? What are the trade-offs of one compared to another? What are the tradeoffs of using ML algorithms compared to using more traditional software? These issues are not necessarily factorisable, and a key goal of a study will be to make sure that the lessons learned by one research team are usefully disseminated to the wider community. In general, each team will serve as a repository of expertise. Beyond the R&D projects it sponsors directly, the team will help others develop and deploy experiment-specific ML-based algorithms in their software stacks. It will provide training to those developing new ML-based algorithms as well as those planning to use established ML tools.

With the advent of more powerful hardware and more performant ML algorithms, these tools will be used to develop application software that will:

- replace the most computationally expensive parts of pattern recognition algorithms and algorithms that extract parameters characterising reconstructed objects;
- compress data significantly with negligible loss of fidelity in terms of physics utility;

- extend the physics reach of experiments by qualitatively changing the types of analyses that can be done.

For example, charged track and vertex reconstruction is one of the most CPU intensive elements of the software stack. The algorithms are typically iterative, alternating between selecting hits associated with tracks and characterising the trajectory of a track (a collection of hits). Similarly, vertices are built from collections of tracks, and then characterised quantitatively. ML algorithms have been used extensively outside HEP to recognize, classify, and quantitatively describe objects. We wish to investigate how to replace the most compute expensive parts of the pattern recognition algorithms and the fitting algorithms that extract parameters characterising the reconstructed objects. As existing algorithms already produce high quality physics, the primary goal of this activity will be developing replacement algorithms that execute much more quickly while maintaining sufficient fidelity.

All HEP detectors produce much more data than can be moved to permanent storage. The process of reducing the size of the data sets is managed by the trigger. Electronics sparsify the data stream using zero suppression and they do some basic data compression. While this reduces the data rate by a factor of 100 or more, to about 1 terabyte per second, another factor of order 1500 is required before the data can be written to tape. ML algorithms have already been used very successfully to rapidly characterise which events should be selected for additional consideration and eventually persisted to long-term storage. The challenge will increase both quantitatively and qualitatively as the number of proton-proton collisions per bunch crossing increases.

Current Practices

The use of ML in HEP analyses has become commonplace over the past two decades. Many analyses use the HEP-specific software package TMVA included in ROOT. Recently, many HEP analysts have begun migrating to non-HEP ML packages like SciKit-Learn and Keras. Data scientists at Yandex created a Python package that provides a consistent API to most ML packages used in HEP, and another that provides some HEP-specific ML algorithms. Packages like Spearmint perform Bayesian optimisation and can improve HEP Monte Carlo work. The keys to successfully using ML for any problem are:

- creating/identifying the optimal training, validation, and testing data samples;
- designing and selecting feature sets;
- defining appropriate problem-specific loss functions.

While each experiment is likely to have different specific use cases, we expect that many of these will be sufficiently similar to each other that research and development can be done commonly. Even when this is not possible, experience with one type of problem will provide insights into how to approach other types of problem. This is why the Inter-experiment Machine Learning forum (IML) has been created in 2016 and already demonstrated the benefits of the collaboration between experiments around Machine Learning.

ML algorithms can often discover patterns and correlations more powerfully than human analysts. This allows qualitatively better analysis of recorded data sets. For example, ML/DL algorithms can be used to characterise the substructure of “jets" observed in terms of underlying physics processes. ATLAS, CMS, and LHCb already use ML algorithms to separate jets into those associated with b-quark, c-quarks, or lighter quarks. ATLAS and CMS have begun to investigate whether sub-jets can be reliably associated with quarks or gluons using ML. If this can be done with both good efficiency and accurate understanding of efficiency, the physics reach of the experiments will be radically extended.

The ATLAS, CMS, and LHCb detectors all produce much more data than can be moved to permanent storage. The process of reducing the size of the data sets is referred to as the trigger. Electronics sparsify the data stream using zero suppression and they do some basic data compression. While this reduces the data rate by a factor of 100 (or more, depending on the experiment) to about 1 terabyte per second, another factor of order 1500 is required before the data can be written to tape (or other long-term storage). ML algorithms have already been used very successfully to rapidly characterise which events should be selected for additional consideration and eventually persisted to long-term storage. The challenge will increase both quantitatively and qualitatively as the number of proton-proton collisions per bunch crossing increases.

Research and Development Roadmap and Goals

The R&D roadmap presented here are based on the preliminary work done in the past years, coordinated by the HSF IMF which will remain the main place to coordinate actions about ML in HEP and ensure the proper links with the data science communities.

By 2020:

- Particle identification and particle properties: in calorimeters or time projection chambers (TPCs), where the data can be represented as a 2D or 3D image, the problems can be cast as a computer vision tasks. DL, in which neural networks are used to reconstruct images from pixel intensities, is a good candidate to identify particles and extract many parameters. Promising DL architectures for these tasks include convolutional, recurrent and adversarial neural networks. A particularly important application is to Liquid Argon TPCs (LArTPCs), which is the chosen detection technology for the flagship neutrino programme.
- Computing resource optimisations: data volume in data transfers is one of the challenges facing the current computing systems. Resource utilisation optimisation based on the enormous amount of data collected can improve overall operations. Networks in particular are going to play a crucial role in data exchange in HL-LHC era. A network-aware application layer may significantly improve experiment’s operations. ML is a promising technology to identify anomalies in network traffic, to predict and prevent

network congestion, to detect bugs via analysis of self-learning networks, and for WAN path optimisation based on user access patterns.

- ML middleware and data formats: HEP is currently mainly relying on ROOT format for its data when the ML community has developed several more specialized formats, often associated with some ML tools. A desirable data format for ML applications should have the following attributes: high read-write speed for efficient training, sparse readability without loading entire dataset into RAM, compression and common use by the ML community. A thorough evaluation of the different data formats and their impact on ML performances in the HEP context is needed and it is necessary to define a strategy for bridging or migrating HEP formats to the chosen ML format(s).
- ML as a Service (MLaS): current cloud providers rely on MLaS model allowing for efficient use of common resources and use interactive machine learning tools. MLaS is not yet widely used in HEP, despite a few successful publications which used it. HEP services for interactive analysis, such as CERN's Service for Web-based Analysis (SWAN), may play an important role in adoption of machine learning tools in HEP workflows. In order to use these tools more efficiently, sufficient and appropriately tailored hardware and instances other than CERN SWAN are needed.

By 2022:

- Detector anomaly detection: data-taking of current complex HEP detectors is continuously monitored by physicists taking shifts to monitor the quality of the incoming data, using reference histograms produced by experts. This makes difficult to anticipate new problems. A whole class of ML algorithms called anomaly detection can be useful for such problems. They are able to learn from data and produce an alert when deviation is seen. By monitoring many variables at the same time such algorithms are sensitive to subtle signs forewarning of imminent failure, so that preemptive maintenance can be scheduled. Such techniques are already used in the industry.
- Simulation: recent progress in high fidelity fast generative models, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), which are able to sample high dimensional feature distributions by learning from existing data samples, offer a promising alternative for simulation. A simplified first attempt at using such techniques saw orders of magnitude increase in simulation over existing fast simulation techniques, but has not yet reached the required accuracy.
- Triggering and real-time analysis: one of the challenges is the trade-off in algorithm complexity and performance under strict inference time constraints. It is necessary to extend currently existing prototype to use DL fast inference in online systems and in particular how to do efficiently the training phase, for example on a large resource platforms. To deal with the increasing event complexity at HL-LHC, we will also explore the use of sophisticated ML algorithms at all trigger levels
- Sustainable Matrix Element Methods (MEM): The MEM is a powerful technique which can be utilised for measurements of physical model parameters and direct searches for new phenomena but it is very computationally intensive and it has limited its applicability

in HEP so far. Using neural networks for numerical integrations is not new. The technical challenge is to design a network which is sufficiently rich to encode the complexity the complexity of the ME calculation for a given process over the phase space relevant to the signal process. Deep Neural Networks (DNNs) are strong candidates.

- Tracking: pattern recognition is the most computationally challenging step. It becomes a computationally huge problem for the HL-LHC. The hope is that machine learning will provide a solution that scales linearly with LHC intensity. A current effort called HEP.TrkX has started to investigate deep learning algorithms such as long-term short-term (LSTM) networks for track pattern recognition on many-core processors.

3.9 Physics Generators

This section is omitted from the current document as the generator community have asked for more time to reach consensus.

3.10 Software Development, Deployment, Validation and Verification

Scope and Challenges

Modern High Energy Physics experiments are large distributed collaborations comprising up to a few hundred people actively writing software. It is therefore vital that the processes and tools used for development are streamlined to ease the process of contributing code and to facilitate collaboration between geographically separated peers. At the same time we must properly manage the whole project, ensuring code quality, reproducibility and maintainability with the least effort possible. Making sure this happens is largely a continuous process, and shares a lot with non-HEP specific software industries.

Work is ongoing to track and promote solutions in the following areas:

- Distributed development of software components, including the tools and processes required to do so (code organisation, documentation, issue tracking, artifact building) and the best practices in terms of code and people management.
- Software quality, including aspects such as modularity and reusability of the developed components, sustainability of the development effort, architectural and performance best practices.
- Deployment of software and interaction with operations teams.
- Validation of the software both at small scales (e.g. best practices on how to write a unit test) and larger ones (large scale validation of data produced by an experiment).
- Software licensing and distribution, including their impact on software interoperability.
- Recognition of the significant contribution that software makes to High Energy Physics as a field.

HEP-specific challenges derive from the fact that HEP is a large, inhomogeneous community with multiple sources of funding, mostly formed of people belonging to small university groups and some larger laboratories. Software development effort within an experiment usually encompasses a huge range of experience and skills, from a few more or less full time experts to many physicist programmers with little formal software training. In addition, the community is split between different experiments that often diverge in timescales, size and resources. Experiment software is usually divided in two separate use cases, production (being it data acquisition, data reconstruction or simulation) and user analysis, whose requirements and life-cycles are completely different. The former is very carefully managed in a centralised and slow moving manner, following the schedule of the experiment itself. The latter is much more dynamic and strongly coupled with conferences or article publication timelines. Finding solutions which adapt well to both cases is not always obvious or even possible.

Current Best Practices

Due to significant variations between experiments at various stages of their lifecycles, there is a huge variation in practice across the community. Thus here we describe *best practice*, with the understanding that this ideal may be far from the reality for some developers.

It is important that developers can focus on the design and implementation of the code and do not have to spend a lot of time on technical issues. Clear procedures and policies must exist to perform administrative tasks in an easy and quick way. This starts with the setup of the development environment. Supporting different platforms not only allows the developers to use their machines directly for the development, it also provides a check of code portability.

To maximise productivity, it is very beneficial to use development tools that are not HEP-specific. There are many open source projects and tools that are of similar scale to large experiment software stacks and standard tools are usually well documented. For source control HEP community has generally chosen to move to *git*, which is very welcome, as it also brings an alignment with many open source projects and commercial organisations. Likewise, CMake is widely used for the builds of software packages, both within HEP and outside. Packaging many build products together into a software stack is an area that still requires close attention with respect to active developments (the HSF has an active working group here).

Proper testing of changes to code should always be done in advance of a change request being accepted. Continuous integration, where merge or pull requests are built and tested in advance is now standard practice in the open source community and in industry. Continuous integration can run unit and integration tests and it can also incorporate code quality checks and policy checks that will help improve the consistency and quality of code at low human cost.

Training and documentation is key to efficient use of developer effort (see also the later chapter on Training). For documentation that has to be specific, favoured solutions would have a low barrier of entry for contributors but also allow and encourage review of material. Consequently it is very useful to host documentation sources in a repository with a similar workflow to code and to use an engine that translates the sources into modern web pages.

Recognition of software work as a key part of science has resulted in number of journals where developers can publish their work. Journals also disseminate information to the wider community in a permanent way and is the most established mechanism for academic recognition. Publication in such journals provides proper peer review, beyond that provided in conference papers, so is valuable for recognition as well as dissemination.

Research and Development Programme

HEP must endeavor to be as responsive as possible to developments outside of our field. In terms of hardware and software tools there remains great uncertainty as to what the platforms offering the best value for money will be on the timescales of a decade. It therefore behooves us to be as generic as possible in our technology choices, retaining the necessary agility to adapt to this uncertain future.

Our vision is characterised by HEP being current with technologies and paradigms that are dominant in the wider software development community, especially for open source software, which we believe to be the right model for our community. In order to achieve that aim we propose that the community establishes a development forum that allows for technology tracking and discussion of new opportunities. The HSF can play a key role in marshalling this group and in ensuring its findings are widely disseminated. In addition, having wider and more accessible training for developers in the field, that will teach the core skills needed for effective software development, would be of great benefit.

Given our agile focus, it is better to propose here projects and objectives to be investigated in the short to medium term, alongside establishing the means to continually review and refocus the community on the most promising areas. The main idea is to investigate new tools as demonstrator projects where clear metrics for success in reasonable time should be established to avoid wasting community effort on initially promising products that fail to live up to expectations.

Ongoing activities, and short-term projects to be completed by 2020, include the following:

- Establish a common forum for the discussion of HEP software problems. This should be modeled along the lines of the Concurrency Forum [ConcurrencyForum], which was very successful in establishing demonstrators and prototypes that were used as experiments started to develop multi-threading frameworks.
- Continue the HSF working group on *Packaging*, with more prototype implementations based on the stronger candidates identified so far.
- Provide practical advice on how to best set up new software packages, developing on the current project template work and working to advertise this within the community.
- Work with HEP experiments and other training projects to provide accessible core skills training to the community. This training should be experiment neutral, but could be usefully combined with the current experiment specific training. Specifically, this work can build on, and collaborate with, recent highly successful initiatives such as the LHCb *StarterKit* and ALICE *Juniors*, and with established generic training initiatives such as *Software Carpentry*.

- Strengthen links with software communities and conferences outside of the HEP domain, presenting papers on the HEP experience and problem domain. SciPy, Supercomputing, RSE Conference and Workshop on Sustainable Software for Science would all be useful conferences to consider.
- Write a paper that looks at case studies of successful and unsuccessful HEP software developments and draws specific conclusions and advice for future projects.

Projects required by 2022 include the following:

- Prototype C++ refactoring tools, with specific use cases in migrating HEP code.
- Prototyping of portable solutions for exploiting modern vector hardware on heterogenous platforms.
- Develop tooling and instrumentation to measure software performance, especially in the domain of concurrency. This should primarily aim to further the developments of existing tools, such as *igprof*, rather than to develop a new one.
- Develop a common infrastructure to gather and analyse data about experiments' software, including profiling information and code metrics, and to ease sharing across different user communities.
- Undertake a feasibility study of a common toolkit for statistical analysis that would be of use in regression testing for experiment's simulation and reconstruction software.

3.11 Software Trigger and Event Reconstruction

Scope and Challenges

The reconstruction of raw detector data and simulated data and its processing in real time represent a major component of today's computing requirements in HEP. Recent work has involved evaluation of the most important components of next generation algorithms, data structures, and code development and management paradigms needed to cope with highly complex environments expected in HEP detector operations in the next decade. New approaches to data processing were also considered, including the use of novel, or at least, novel to HEP, algorithms, and the movement of data analysis into real-time environments.

Software trigger and event reconstruction techniques in HEP face a number of new challenges in the next decade. Advances in facilities and future experiments bring a dramatic increase in physics reach, as well as increased event complexity and rates. At the HL-LHC, the central challenge for object reconstruction is to maintain excellent efficiency and resolution in the face of high pileup values, especially at low object p_T. Detector upgrades such as increases in channel density, high precision timing and improved detector geometric layouts are essential to overcome these problems. In many cases these new technologies bring novel requirements to software trigger and event reconstruction algorithms or require new algorithms to be developed. Ones of particular importance at the HL-LHC include high-granularity calorimetry, precision timing detectors, and hardware triggers based on tracking information which may seed later software trigger and reconstruction algorithms.

Trigger systems for next-generation experiments are evolving to be more capable, both in their ability to select a wider range of events of interest for the physics programme, and their ability to stream a larger rate of events for further processing. ATLAS and CMS both target systems where the output of the hardware trigger system is increased by 10x over the current capability, up to 1 MHz [ATLAS2015, CMS2015]. In LHCb [LHCb2014] and ALICE [ALICE2015], the full collision rate (between 30 to 40 MHz for typical LHC pp operations) will be streamed to real-time or quasi-realtime software trigger systems. The increase in event complexity also brings a "problem" of overabundance of signal to the experiments, and specifically the software trigger algorithms. The evolution towards a genuine real-time analysis of data has been driven by the need to analyse more signal than can be written out for traditional processing, and technological developments which make it possible to do this without reducing the analysis sensitivity or introducing biases.

Evolutions in computing technologies are both opportunities to move beyond commodity x86 technologies, which HEP has used very effectively over the past 20 years, and significant challenges to derive sufficient event processing throughput per cost to reasonably enable our physics programmes [Bird2014]. Specific items identified include the increase of SIMD capabilities (processors capable of running a single instruction set simultaneously over multiple data), the evolution towards multi- or many-core architectures, the slow increase in memory bandwidth relative to CPU capabilities, the rise of heterogeneous hardware, and the possible evolution in facilities available to HEP production systems.

The move towards open source software development and continuous integration systems brings opportunities to assist developers of software trigger and event reconstruction algorithms. Continuous integration systems have already allowed automated code quality and performance checks, both for algorithm developers and code integration teams. Scaling these up to allow for sufficiently high statistics checks is among the still outstanding challenges. As the timescale for experimental data taking and analysis increases, the issues of legacy code support increase. Code quality demands increase as traditional offline analysis components migrate into trigger systems, or more generically into algorithms that can only be run once.

Substantial computing facilities are in use for both online and offline event processing across all experiments surveyed. Online facilities are dedicated to the operation of the software trigger, while offline facilities are shared for operational needs including event reconstruction, simulation (often the dominant component) and analysis. CPU in use by experiments is typically at the scale of tens or hundreds of thousands of x86 processing cores. Projections to future needs, such as for the HL-LHC, show the need for a substantial increase in scale of facilities without significant changes in approach or algorithms.

Current Practices

Currently, the CPU needed for event reconstruction tends to be dominated by charged particle reconstruction (tracking), especially as the need for efficiently reconstructing low p_T particles is considered. Calorimetric reconstruction, particle flow reconstruction, particle identification algorithms also make up significant parts of the CPU budget in some experiments. Disk storage is typically 10s to 100s of PB per experiment. It is dominantly used to make the output of the event reconstruction, both for real data and simulation, available for analysis. Current generation experiments have moved towards smaller, but still flexible, data tiers for analysis. These tiers are typically based on the ROOT [Brun1996] file format and constructed to facilitate both skimming of interesting events and the selection of interesting pieces of events by individual analysis groups or through centralized analysis processing systems. Initial implementations of real-time analysis systems are in use within several experiments. These approaches remove the detector data that typically makes up the raw data tier kept for offline reconstruction, and keep only final analysis objects [Aaij2016, Abreu2014, CMS2016]. In the case of detector calibration and alignment systems, generally a high level of automation is in place across experiments, both for very frequently updated measurements and more rarely updated measurements. Often automated procedures are integrated as part of the data taking and data reconstruction processing chain. Some longer term measurements, requiring significant data samples to be analysed together remain as critical pieces of calibration and alignment work. These techniques are often most critical for a subset of precision measurements rather than for the entire physics programme of an experiment.

The next decade will see the volume and complexity of data being processed by HEP experiments increase by at least one order of magnitude. While much of this increase is driven by the planned upgrades to the four major LHC detectors, new experiments such as DUNE will also make significant demands on the HEP data processing infrastructure. It is therefore essential that event reconstruction algorithms and software triggers continue to evolve so that they are able to efficiently exploit future computing architectures and deal with this increase in data rates without loss of physics capability.

Research and Development Programme

Seven key areas, which are itemised below, have been identified where research and development is necessary to enable the community to exploit the full power of the enormous datasets that we will be collecting. Three of these areas concern the increasingly parallel and heterogeneous computing architectures which we will have to write our code for. In addition to a general effort to vectorise our codebases, we must understand what kinds of algorithms are best suited to what kinds of hardware architectures, develop benchmarks that allow us to compare the physics-per-dollar-per-watt performance of different algorithms across a range of potential architectures, and find ways to optimally utilise heterogeneous processing centres. The consequent increase in the complexity and diversity of our codebase will necessitate both a determined push to educate tomorrow's physicists in modern coding practices, and a development of more sophisticated and automated quality assurance and control for our codebases. The increasing granularity of our detectors, and the addition of timing information which seems mandatory to cope with the extreme pileup conditions at the HL-LHC, will require us to both develop new kinds of reconstruction algorithms and to make them fast enough for use in real-time. Finally, the increased signal rates will mandate a push towards real-time analysis in many areas of HEP, in particular those with low-p_T signatures.

The proposed R&D programme focuses on the following:

- HEP developed toolkits and algorithms typically make poor use of vector units on commodity computing systems. Improving this will bring speedups to applications running on both current computing systems and most future architectures. The goal for work in this area is to evolve current toolkit and algorithm implementations, and best programming techniques, to better use SIMD capabilities of current and future computing architectures.

- Computing platforms are generally evolving towards having more cores in order to increase processing capability. This evolution has resulted in multi-threaded frameworks in use, or in development, across HEP. Algorithm developers can improve throughput by being thread-safe and enabling the use of fine-grained parallelism. The goal is to evolve current event models, toolkits and algorithm implementations, and best programming techniques to improve the throughput of multithreaded software trigger and event reconstruction applications.

- Computing architectures using technologies beyond CPUs offer an interesting alternative for increasing throughput of the most time consuming trigger or reconstruction algorithms. Such architectures (e.g., GPUs, FPGAs) could be easily integrated into dedicated trigger or specialized reconstruction processing facilities (e.g., online computing farms). The goal is to demonstrate how the throughput of toolkits or algorithms can be improved through the use of new computing architectures in a production environment.

- HEP experiments have extensive continuous integration systems, including varying code regression checks that have enhanced the quality assurance (QA) and quality control (QC) procedures for software development in recent years. These are typically maintained by individual experiments and have not yet reached the scale where statistical regression, technical, and physics performance checks can be performed for each proposed software change. The goal is to enable the development, automation, and deployment of extended QA and QC tools and facilities for software trigger and event reconstruction algorithms.

- Real-time analysis techniques are being adopted to enable a wider range of physics signals to be saved by the trigger for final analysis. As rates increase, these techniques can become more important and widespread by enabling only the parts of an event associated with the signal candidates to be saved, reducing the required disk space. The goal is to evaluate and demonstrate the tools needed to facilitate real-time analysis techniques. Research topics include compression and custom data formats; toolkits for real-time detector calibration and validation which will enable full offline analysis chains to be ported into real-time; and frameworks which will enable non-expert offline analysts to design and deploy real-time analyses without compromising data taking quality.

- The central challenge for object reconstruction at HL-LHC is to maintain excellent efficiency and resolution in the face of high pileup values, especially at low object p_T. Both trigger and reconstruction approaches need to exploit new techniques and higher granularity detectors to maintain or even improve physics measurements in the future. It is also becoming increasingly clear that reconstruction in very high pileup environments, such as the HL-LHC or FCC-hh, will not be possible without adding some timing information to our detectors, in order to exploit the finite time during which the beams cross and the interactions are produced. The goal is to develop and demonstrate efficient techniques for physics object reconstruction and identification in complex environments.

- Future experimental facilities will bring a large increase in event complexity. The scaling of current-generation algorithms with this complexity must be improved to avoid a large increase in resource needs. In addition, it may be desirable or indeed necessary to deploy new algorithms, including advanced machine learning techniques developed in other fields, in order to solve these problems. The goal is to evolve or rewrite existing toolkits and algorithms focused on their physics and technical performance at high event complexity (e.g. high pileup at HL-LHC). Most important targets are those which limit expected throughput performance at future facilities (e.g., charged-particle tracking). A number of such efforts are already in progress across the community.

The success of this research and development programme will be intimately linked to challenges confronted in other areas of HEP computing, most notably the development of software frameworks that are able to support heterogeneous parallel architectures, including the associated data structures and I/O, the development of lightweight detector models that maintain physics precision with minimal timing and memory consequences for the reconstruction, enabling the use of offline analysis toolkits and methods within real-time analysis, and an awareness of advances in machine learning reconstruction algorithms being developed outside HEP and the ability to apply them to our problems. For this reason perhaps the most important task ahead of us is to maintain the community, which has coalesced together in this CWP process, so that the work done in these sometimes disparate areas of HEP fuses coherently together into a solution to the problems facing us over the next decade.

3.12 Visualisation

Scope and Challenges

In modern High Energy Physics (HEP) experiments, visualisation of data has a key role in many activities and tasks across the whole data chain: detector development, monitoring, event generation, reconstruction, detector simulation, data analysis, and outreach and education.

Applications which let the user explore event-based data are usually called *event displays*. They are the main tool to explore experimental data at the event level and to visualise the detector itself. There are two main types of event displays: those integrated in the experiments' frameworks, which are able to access and visualise all an experiment's data, but at the cost in complexity and portability; and those designed as cross-platform applications, lightweight and fast, delivering only a simplified version or a subset of the event data. In the first case, access to data is tied intimately to an experiment's data model (for both event and geometry data) and this inhibits portability; in the second, processing the experiment data into a generic format usually loses some details and is an extra processing step. There are then various graphical backends that can be used to visualise the final product, either standalone or within a browser, and these can have a substantial impact on the types of devices supported.

Beyond event displays, HEP also has statistical visualisations, such as histograms, which close the loop between data analyst and data, allowing the analyst to quickly, and with minimal effort, characterise the data (a so called exploratory data analysis). Unlike event displays, these visualisations are not strongly visually linked to the detector geometry, and often aggregate data from multiple events.

Other types of visualisations are used in HEP to display non-spatial data, like the graphs used to visually describe the structure of the detector description or the dependency graph between the data products of different algorithms during reconstruction.

The main challenges in this area are in the sustainability of the many experiment-specific visualisation tools, when common projects could reduce duplication and increase quality and long term viability. The ingestion of event and other data could be eased by common formats, which would need to be defined and satisfy all users. Changes to support a client-server architecture would help broaden the ability to support new devices, like mobile phones. Making a good choice for the libraries used to rendering 3D shapes is also key, impacting on the range of output devices that can be supported and the level of interaction with the user that is offered. Reacting to a fast changing technology landscape is very important - HEP's effort is limited and generic solutions can often be used with modest effort. This applies strongly to the non-event visualisation area, where many open source and industry standard tools exist.

Current Practices

Three key features characterise almost all HEP event displays:

- Event-based workflow: applications access experimental data on an event-by-event basis, visualising the data collections belonging to a particular event. Data can be related to the actual physics events (e.g. physics objects, like jets, tracks) or to the experimental conditions (e.g. detector description versions, calibrations).
- Geometry visualisation: applications provide a representation of the detector's geometry. The application can display the real geometry of the detector, as retrieved from the experiments' software frameworks, or a simplified description, usually for the sake of speed, computing efficiency or portability.
- Interactivity: applications offer different interfaces and tools to users, in order to interact with the visualisation itself, select event data and set cuts on objects' properties.

Experiments have often developed multiple event displays that either take the full integration approach explained above or are standalone and rely on extracted and simplified data.

For the actual visualisation of data, this can be achieved in standalone applications through the low level OpenGL API or within a web browser, using WebGL. Using OpenGL directly is robust and avoids other dependencies, but implies a significant effort. Instead of using the API directly, a library layer on top of OpenGL (e.g., Coin3D) can more closely match the underlying data, like geometry, and offer a higher level API, which simplifies development. This carries risk, however, that if the library itself becomes deprecated, as has happened with Coin3D, the experiment needs to migrate to a different solution or to take on the maintenance burden itself. The alternative, embedding the display in a browser, offers many portability advantages (e.g., easier support for mobile or virtual reality devices), but at some cost of not supporting the most complex visualisations or all useful interactivity.

For statistical data, ROOT [Brun1996] has been the tool of choice in HEP for many years and satisfies most use cases the community have. However, increasing use of generic tools and data formats mean Matplotlib (Python) or JavaScript based solutions (used for example in Juypiter notebooks) have made the landscape more diverse. For visualising trees or graphs, there are many generic offerings.

Research and Development Roadmap

The workshop that was held as part of the CWP process was felt to be extremely useful for exchanging knowledge between developers in different experiments and in bringing in ideas from outside the community. These will now be held as annual events and will facilitate work on the common R&D plan.

The main goal of R&D projects in this area will be to develop techniques and tools which let visualisation applications and event displays be less dependent on specific experiments' software frameworks, leveraging common packages and common data formats. Exporters and interface packages will be designed as bridges between the experiments' frameworks, needed to access data at a high level of detail, and the common packages based on the community standards that this group will develop.

As part of this development work, demonstrators will be designed to show the usability of our community solutions and tools. The goal will be to get a final design of those tools so that the experiments can depend on them in their future developments.

The WG will also work towards a more convenient access to geometry and event data, through a client-server interface. In collaboration with the Data Access and Management WGs, an API or a service to deliver streamed event data would be designed.

The work above should be completed by 2020.

Beyond that point, the focus will be on developing the actual community-driven tools, to be used by the experiments for their visualisation needs in production, potentially taking advantage of new data access services.

4 Training and Careers

Training Challenges

HEP is facing major challenges with its software and computing that require innovative solutions based on the proper adoption of new technologies. More and more technologies emerge from outside HEP as scientific communities and industry face challenges similar to ours and produce solutions relevant to us. The integration of such technologies in our frameworks and computing infrastructure requires skilled people with expertise on the various aspects of software and computing and it is important that a large fraction of the community is able to adopt, or at least use, these new tools and paradigms.

One characteristic quite specific to HEP is that there is an overlap between users (physicists) and computing experts. Instead of the traditional situation in which users express their requirements and computer specialists implement solutions, there is a close collaboration between them which is essential for success. This does not come from an organisational problem that needs to be solved, but is strongly linked to the nature of the science being done where the challenging needs require solutions that have to evolve continuously based on what has been observed and on the experience gained. Many details of the experiment data cannot be known before the data taking has started and each evolution of the detector, or improvement of the machine performance, can have important consequences for the software and the computing infrastructure.

This reinforces the need to spread best software engineering practices and software technologies to a very large number of people, including physicists involved in the design and developments, through the whole spectrum of data processing application from triggering to analysis. This results in a very diverse audience for training: from novice programmers to more advanced or expert users.

Because of the complexity of HEP experiments, reflected in their software, good training which maximises the potential impact on the community is a training done by community experts. At the same time, teaching requires a significant time and this is not always very compatible with the time constraints of these experts. There should be more incentives in our community for training efforts. For young people at the beginning of their career, one possibility would be to take these efforts in account in their career path when today it is often against their short term interest to spend time in training activities, compared to other more visible activities. Possible incentives are highly dependent on policies and boundary conditions of the organisation or country the person is affiliated with.

HEP is a challenging field and it has the potential to attract skilled young people who are looking for experiences in diverse, demanding contexts. Nevertheless, many, if not the vast majority, of these persons will not have their career in HEP. This is one more reason for adopting technologies that can be used outside the field, as much as possible. At the same time, to be valuable for these people, the training provided in the community must not be too specific to HEP use cases, or to one experiment, and should promote practices that can be used outside HEP.

On the other hand, experiments have a scientific programme to accomplish and often tend to focus on the training required to accomplish their short term goals. The right balance should be found between these two requirements. It is necessary to find the appropriate incentives to favor training activities that bring more benefits in the medium to long term, both for the experience, the community and the career of the trainees, possibly outside academic research.

Possible Directions for Training

To increase the training activities in the community, whilst taking into account the constraints of both the attendees and the trainers, it is necessary to explore new approaches to training. The current “school” model (e.g. Bertinoro school of computing, GridKa school of computing) is well established, but is not extensible as it requires a significant dedicated time of all the participants at the same time and location. In spite of this, it remains a very valuable component of the training activities and, as with hands-on tutorials organized during conferences and workshops, the resulting networking is an important feature of these events where people build relationships with other experts.

There are, however, opportunities to work with HEP experiments and other training projects to provide accessible core skills training to the community, by basing them at labs where students can easily travel. This training should be experiment neutral, but could be usefully combined with the current experiment specific training. This work can build on, and collaborate with, recent highly successful initiatives such as the LHCb *StarterKit* and ALICE *Juniors*, and with established generic training initiatives such as *Software Carpentry*.

Several R&D projects in the last years have had training as one of their activities, like DIANA-HEP or MVA4NewPhysics. This has proved to be an efficient incentive to organize training events and has contributed to spread the expertise on advanced topics. We think that training should become an integral part of future major R&D projects in the community.

New pedagogical methods, like active training or peer training, have emerged as interesting approaches, complementary to the schools or topical tutorials. One of the basic ideas is an online material shared by a student and a teacher, possibly with notebooks to provide real examples or practical exercises. Building such material is a time-consuming activity that also requires experts effort. An interesting approach that started to emerge is the ability of students, or other experts, to enrich the initial material by comments or examples. The HSF experimented this approach with WikiToLearn, a platform developed in Italy outside HEP, to promote this kind of training and collaborative elaboration/enrichment of the training materials.

HEP is not the only community with increased needs of training and there is a lot of initiatives and materials available, in the form of online tutorials, active training or Massive open online courses (MOOCs). It would be a waste of effort for HEP to reinvent the wheel and produce its own materials for things that are not specific to our scientific field. HEP should spend some effort to evaluate some of the existing courses and build a repository of the good ones, appropriate to HEP needs. This is not a negligible effort and would require some dedicated effort.

A service that emerged in the last years as a very valuable means of sharing expertise is Question and Answer (Q&A) systems. A few such systems are run by some experiments for their own needs, but it is not necessarily optimal, as the value of these services, as exemplified by StackOverflow, is the large number of contributors with diverse backgrounds. Running a cross-experiment Q&A system has been discussed but it has not yet been possible to converge on a viable approach, both technically and for the effort required to run and support such a service (in particular moderators).

Career Recognition

Computer specialists in our field are often physicists who specialized into computing. This has always been the case and tend to continue. For young people, this leads to a career recognition problem, as this is not a well recognized role, physicists recognition being based generally on participation in data analysis. There is no easy solution to this problem and possible paths for improvements are highly dependent on organisations and countries. Nevertheless, we believe that improving the career recognition of physicists who specialized in computing, like others who specialized in detector hardware, is important for the future to ensure the continued successful collaboration between physicists and computer specialists or computer scientists that is one of the core ingredient for HEP software and computing success.

5 Conclusions

Future challenges for High Energy Physics in the domain of software and computing are not simply an extrapolation of the challenges faced today. The needs of ATLAS and CMS in the high luminosity era far exceed those that can be met by simply by making incremental changes to today's code and and scaling up computing facilities within the foreseen budget. At the same time, the limitation in single core CPU performance is making the landscape of computing hardware far more diverse and challenging to exploit, whilst offering huge performance boosts for suitable code. Exploiting parallelism and other new techniques, such as modern machine learning, offer great promise, but will require substantial work from the community to adapt to our problems. If there was any lingering notion that software or computing could be done cheaply by a few junior people for modern experimental programmes, that should now be thoroughly dispelled.

HEP Software and Computing requires a step change in its profile and effort to match the challenges ahead. We need investment in people who can understand the problems we face, the solutions employed today and have the correct skills to provide innovative solutions for the future. There needs to be recognition from the whole community for the work done in this area, with a recognised career path for these experts. In addition, we will need to invest heavily in training for the whole software community as the contributions of the bulk of non-expert physicists are also vital for our success.

We have presented programmes of work that the community have identified as being part of the roadmap for the future. While there is always some scope to reorient current effort in the field, we would highlight the following work programmes as being of the highest priority for investment to address the goals which were set in the introduction.

Improvements in software efficiency, scalability and performance

> The bulk of CPU cycles consumed by experiments relate to the fundamental challenges of simulation and reconstruction. Thus the work programmes in these areas, together with the frameworks that support them, are of critical importance. Further, as the provisioning of resources in WLCG is the mechanism by which this work actually gets done, optimisation of our distributed computing systems, including data and workload management, is paramount.

Enable new approaches that can radically extend physics reach

> Again, new techniques in simulation and reconstruction will be vital here. Physics analysis is an area where new ideas can be particularly fruitful. Exploring the full potential of machine learning is one common theme that underpins many new approaches and the community should endeavor to share knowledge widely across subdomains. New data analysis paradigms coming from the Big Data industry, based on

innovative parallelised data processing on a large computing farms, could transform data analysis.

Ensure the long term sustainability of the software

Applying modern software development techniques to our codes has, and will continue to, increase developer productivity and code quality. There is ample scope for more common tools and common training to equip the community with the correct skills. Data Preservation makes sustainability an immediate goal of development and analysis and helps reap the benefits of our experiments for decades to come.

When considering a specific proposal from any of the working groups in this document, their impact, measured against these criteria, should be evaluated. Moreover, establishing links outside of our community to other academic disciplines or industry facing similar challenges, as well as with the computer science community who explore innovative paths, has the potential to bring significant benefits. On the decade timescale there will almost certainly be disruptive changes that cannot be planned for and our community must remain agile enough to adapt to these.

The HEP community has many natural subdivisions, between different regional funding agencies, between universities and laboratories and between different experiments. It was in an attempt to overcome these obstacles and to encourage the community to work together in an efficient and effective way that the HEP Software Foundation was established in 2014. This Community White Paper process has been possible only because of the success of that effort in bringing the community together. The need for more common developments in the future, as underlined here, reinforces the importance of the HSF as a common point of contact between all the parties involved, strengthening our community spirit and continuing to help share expertise and identify priorities. For these reasons, we believe that the HSF must be strongly supported as part of our roadmap to success.

Appendix A - List of Workshops

HEP Software Foundation Workshop

Date: 23-26 Jan, 2017

Location: UCSD/SDSC (La Jolla, CA, USA)

URL: http://indico.cern.ch/event/570249/

Description: This HSF workshop at SDSC/UCSD was the first workshop supporting the CWP process. There were plenary sessions covering topics of general interest as well as parallel sessions for the many topical working groups in progress for the CWP.

Software Triggers and Event Reconstruction WG meeting

Date: 9 Mar, 2017

Location: LAL-Orsay (Orsay, France)

URL: https://indico.cern.ch/event/614111/

Description: This was a meeting of the Software Triggers and Event Reconstruction CWP working group. It was held as a parallel session at the "Connecting the Dots" workshop, which focuses on forward-looking pattern recognition and machine learning algorithms for use in HEP.

IML Topical Machine Learning Workshop
Date: 20-22 Mar, 2017
Location: CERN (Geneva, Switzerland)
URL: https://indico.cern.ch/event/595059
Description: This was a meeting of the Machine Learning CWP working group. It was held as a parallel session at the "Inter-experimental Machine Learning (IML)" workshop, an organisation formed in 2016 to facilitate communication regarding R&D on ML applications in the LHC experiments.

Community White Paper Follow-up at FNAL
Date: 23 Mar, 2017
Location: FNAL (Batavia, IL, USA)

URL: https://indico.fnal.gov/conferenceDisplay.py?confId=14032
Description: This one-day workshop was organized to engage with the experimental HEP community involved in computing and software for Intensity Frontier experiments at FNAL. Plans for the CWP were described, with discussion about commonalities between the HL-LHC challenges and the challenges of the FNAL neutrino and muon experiments

CWP Visualisation Workshop
Date: 28-30 Mar, 2017
Location: CERN (Geneva, Switzerland)
URL: https://indico.cern.ch/event/617054/
Description: This workshop was organized by the Visualisation CWP working group. It explored the current landscape of HEP visualisation tools as well as visions for how these could evolve. There was participation both from HEP developers and industry.

DS@HEP 2017 (Data Science in High Energy Physics)
Date: 8-12 May, 2017
Location: FNAL (Batava, IL, USA)
URL: https://indico.fnal.gov/conferenceDisplay.py?confId=13497
Description: This was a meeting of the Machine Learning CWP working group. It was held as a parallel session at the "Data Science in High Energy Physics (DS@HEP)" workshop, a workshop series begun in 2015 to facilitate communication regarding R&D on ML applications in HEP.

HEP Analysis Ecosystem Retreat

Date: 22-24 May, 2017

Location: Amsterdam, the Netherlands

URL: http://indico.cern.ch/event/613842/

Summary report:
http://hepsoftwarefoundation.org/assets/AnalysisEcosystemReport20170804.pdf

Description: This was a general workshop, organized about the HSF, about the ecosystem of analysis tools used in HEP and the ROOT software framework. The workshop focused both on the current status and the 5-10 year time scale covered by the CWP.

CWP Event Processing Frameworks Workshop

Date: 5-6 Jun, 2017

Location: FNAL (Batavia, IL, USA)

URL: https://indico.fnal.gov/conferenceDisplay.py?confId=14186

Description: This was a workshop held by the Event Processing Frameworks CWP working group.

HEP Software Foundation Workshop

Date: 26-30 Jun, 2017

Location: LAPP (Annecy, France)

URL: https://indico.cern.ch/event/613093/

Description: This was the final general workshop for the CWP process. The CWP working groups came together to present their status and plans, and develop consensus on the organisation and context for the community roadmap. Plans were also made for the CWP writing phase that followed in the few months following this last workshop.

References

[Aaij2016] R. Aaij, et. al., Tesla : an application for real-time data analysis in High Energy Physics, Comput. Phys. Commun. 208 (2016) 35-42 (https://cds.cern.ch/record/2147693).

[Abreu2014] R. Abreu, The upgrade of the ATLAS High Level Trigger and Data Acquisition systems and their integration, Proceedings of 19th IEEE-NPSS Real-Time conference 2014, pp.7097414 (2014) (https://cds.cern.ch/record/1702466/).

[Agostinelli2003] Agostinelli et al, Geant4—a simulation toolkit, Nuclear Instruments and Methods in Physics Research A, 506 (2003).

[ALICE2015] ALICE Collaboration, Technical Design Report for the Upgrade of the Online-Offline Computing System, (https://cds.cern.ch/record/2011297).

[ALICE-OpenData] http://opendata.cern.ch/education/ALICE.

[ALPGEN] Mangano *et al*, ALPGEN, a generator for hard multiparton processes in hadronic collisions, arXiv:hep-ph/0206293.

[ATLAS2015] ATLAS Collaboration, ATLAS Phase-II Upgrade Scoping Document, CERN-LHCC-2015-020, LHCC-G-166 (2015) (https://cds.cern.ch/record/2055248)

[ATLAS2015a] ATLAS Collaboration, ATLAS Data Access Policy, ATL-CB-PUB-2015-001, (2015), https://cds.cern.ch/record/2002139/.

[Bird2014] I. Bird, et. al., Update of the Computing Models of the WLCG and the LHC Experiments, CERN-LHCC-2014-014, (2014), (http://cds.cern.ch/record/../LCG-TDR-002.pdf).

[Bird2017] I. Bird. The Challenges of Big (Science) Data, EPS-ECFA Meeting, Venice (2017), (http://eps-hep2017.eu/)

[Brun1996] R. Brun and F. Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/.

[Butler2013] M. Butler, R. Mount, and M. Hildreth. Snowmass 2013 Computing Frontier Storage and Data Management. ArXiv e-prints, November 2013.

[Campana2016] S. Campana, presentation to the 2016 Aix-les-Bains ECFA HL-LHC workshop, Oct 3 2016 https://indico.cern.ch/event/524795/.../ECFA2016.pdf

[Campana2017] ATLAS Collaboration presentation to the LHC RRB, September 2017.

[CMS2015] CMS Collaboration, Technical Proposal for the Phase-II Upgrade of the CMS

Detector, CERN-LHCC-2015-010, LHCC-P-008, CMS-TDR-15-02 (2015) https://cds.cern.ch/record/2020886.

[CMS2016] CMS Collaboration, Search for narrow resonances in dijet final states at s√= 8 TeV with the novel CMS technique of data scouting, Phys. Rev. Lett. 117 (2016) 031802 (https://cds.cern.ch/record/2149625).

[CMS-OpenData] http://opendata.cern.ch/research/CMS.

[CSRG2016] Computing Resources Scrutiny Group Report, CERN-RRB-2016-049, April 2016, https://cds.cern.ch/record/2133213/.

[ConcurrencyForum] http://concurrency.web.cern.ch/.

[CVMFS] J Blomer et al.; 2011 J. Phys.: Conf. Ser. 331 042003, Distributing LHC application software and conditions databases using the CernVM file system http://iopscience.iop.org/article/10.1088/1742-6596/331/4/042003/meta.

[DPHEP] https://hep-project-dphep-portal.web.cern.ch/.

[ESPP2013] The European Strategy for Particle Physics Update 2013, https://cds.cern.ch/record/1567258.

[Frontier] Frontier Distributed Database Caching System, http://frontier.cern.ch

[HSF2015] HEP Software Foundation (HSF) White Paper Analysis and Proposed Startup Plan 1.1, 2015 http://hepsoftwarefoundation.org/assets/HSFwhitepaperanalysisandstartupplanV1.1.pdf.

[HSF2017] http://hepsoftwarefoundation.org/activities/cwp.html

[LHCb2014] LHCb Collaboration, LHCb Trigger and Online Upgrade Technical Design Report, CERN-LHCC-2014-016 (2014) (https://cds.cern.ch/record/1701361).

[MONARC] The MONARC project: http://monarc.web.cern.ch/MONARC.

[Mangano2016] The Physics Landscape of the High Luminosity LHC, http://cern.ch/go/7QCc.

[ODP] CERN Open Data Portal, LHC Experiment Policies, retrieved 2017-08-18, http://cern.ch/go/lVT6.

[P5-2014] Particle Physics Project Prioritization Panel. Building for Discovery: Strategic Plan for U.S. Particle Physics in the Global Context. http://science.energy.gov/~/media/hep/hepap/pdf/May%202014/FINAL_DRAFT2_P5Report_WEB_052114.pdf.

[Panzer2017] Computing Evolution: Technology and Markets, Presented at the HSF CWP Workshop in San Diego, January 2017.

https://indico.cern.ch/event/570249/contributions/2404412/attachments/1400426/2137004/2017-01-23-HSFWorkshop-TechnologyEvolution.pdf

[RECAST] a reinterpretation service https://arxiv.org/abs/1010.2506.

[SciGateway] https://sciencegateways.org.

[Sexton-Kennedy2017] CMS Collaboration presentation to the LHC RRB, September 2017.

[Shears2017] DPHEP Update, presented in the Grid Deployment Board, October 2017, https://indico.cern.ch/event/578991/.

[SHERPA] Gliesberg *et al*, Event generation with SHERPA 1.1, JHEP 0902 (2009).

[SSI2017] https://www.software.ac.uk/which-journals-should-i-publish-my-software, retrieved Octover 2017.

[WLCG2016] Charge for Producing a HSF Community White Paper, July 2016, http://hepsoftwarefoundation.org/assets/CWP-Charge-HSF.pdf.

[XRootD] XRootD file access protocol : XRootD: Home Page.