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

The HEP Software Foundation

v0.2 Released 2017-11-17

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 the scientific goals are:

- 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,
- study matter-antimatter asymmetry manifestations, particularly via the decays of heavy flavoured particles such as D and B mesons, and tau leptons,
- search for signatures of dark matter,
- probe neutrino oscillations and masses,
- study the Quark Gluon Plasma state of matter in heavy-ion collisions,
- explore the unknown.

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 (Figure LHC-Schedule). The ATLAS and CMS collaborations will continue to make measurements in the Higgs sector, while searching for new physics Beyond the Standard Model. Should a BSM discovery be made, a full exploration of that physics will be pursued. 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].

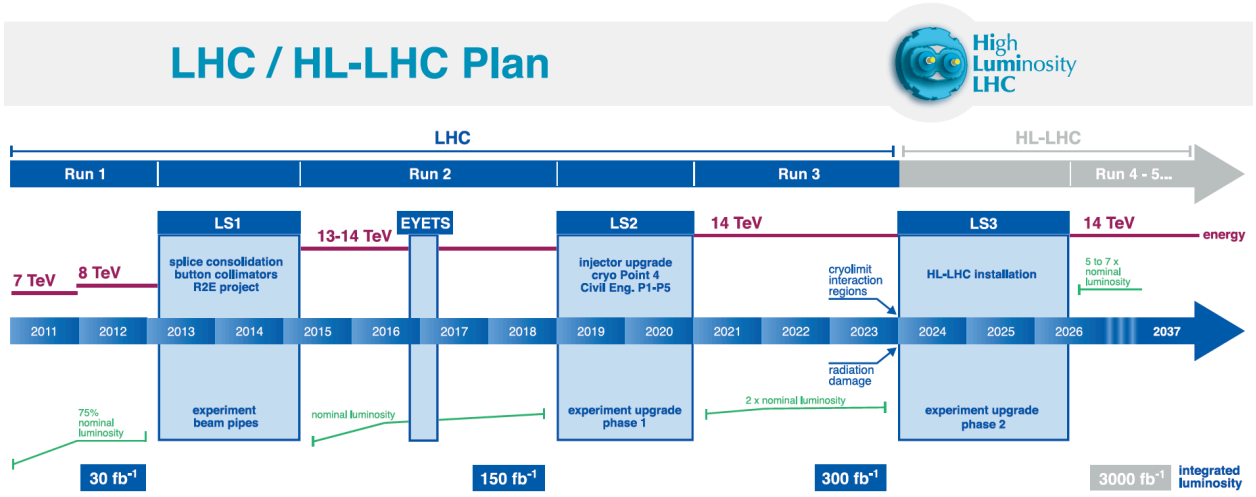

Figure LHC-Schedule: The current schedule for the LHC and HL-LHC upgrade and run. Currently, the start of the HL-LHC run is foreseen for mid 2026. The long shutdowns, LS2 and LS3, will be used to upgrade both the accelerator and the detector hardware. [DetectorUpgrade]

The LHCb experiment at the LHC [LHCb] and the Belle II experiment at KEK [BelleII] study various aspects of heavy flavour physics (B, Charm, D and tau-lepton physics), where quantum influences of very high mass particles are manifest in lower energy phenomena. Their primary goal is to look for BSM physics responsible for charge parity (CP) violation (that is, asymmetries in the decays of particles and their corresponding antiparticles) in rare heavy flavour decays. Current observations of such asymmetries do not explain why our universe is so matter dominated. These flavour physics programmes are related to BSM searches through effective field theory and powerful constraints on new physics keep coming from such studies.

The study of neutrinos, their mass and oscillations, can also shed light on matter-antimatter asymmetry. The DUNE experiment will provide a huge improvement in our ability to probe neutrino physics, detecting neutrinos from the Long Baseline Neutrino Facility at Fermilab, as well as linking to astro-particle physics programmes, in particular through the potential detection of supernovas and relic neutrinos. An overview of the experimental programme scheduled at the Fermilab facility is given in Figure (Figure FNALSchedule).

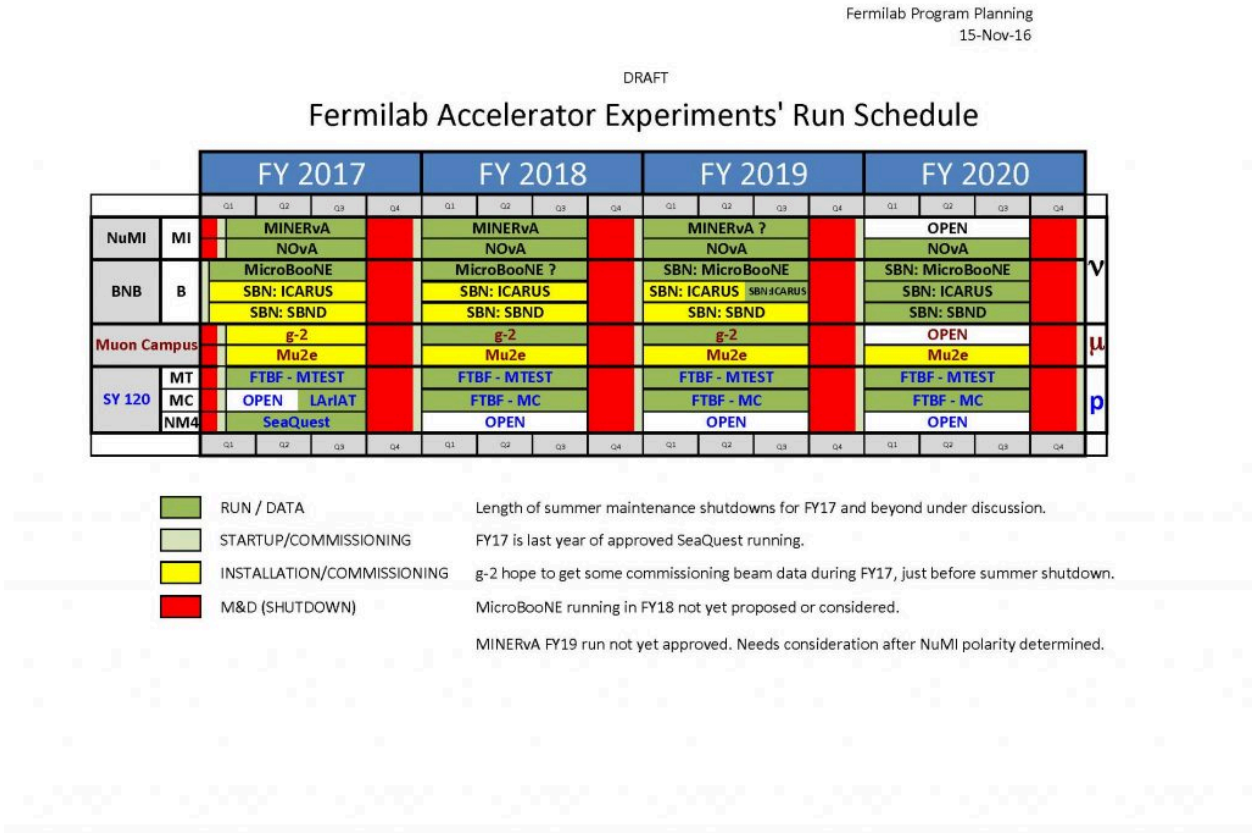

Figure FNALSchedule: **TEMPORARY FIGURE**. Run schedule for the Fermilab facility in the coming years. The installation of new neutrino detectors will be completed in 2019 and data taking will start.

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 [ALICE] and the CMB and PANDA experiments at the Facility for Antiproton and Ion Research (FAIR) are specifically designed to probe this aspect of nuclear and particle physics.

These experimental programmes require large investments in detector hardware, either to build new experiments (e.g. DUNE, FAIR) or to upgrade existing ones (Belle II, HL-LHC). Similarly, they require commensurate investment in the research and development necessary to deploy software to acquire, manage, process, and analyse the data recorded.

For the HL-LHC, which is scheduled to begin taking data in 2026 (see Figure LHC-Schedule) 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 with radically new paradigms, 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, as detailed in this paper. An important message of this report is that 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 (WLCG) [WLCG2016]. The role of the HSF is to facilitate coordination and common efforts in HEP software and computing internationally and to provide a structure for the community to set goals and priorities for future work. The objective of the CWP is to provide a roadmap for software R&D in preparation for the HL-LHC 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,
- to ensure data and knowledge preservation beyond the lifetime of individual experiments,
- to attract the required new expertise by offering appropriate career recognition to physicists specialising in software development and by an effective training effort targeting all contributors in the community

The CWP process, organised 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 at LAP, 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 summarised 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 and may also benefit other projects with similar technical challenges, particularly in astrophysics, e.g., the Square Kilometer Array (SKA), and the Cherenkov Telescope Array (CTA).

2 Software and Computing Challenges

2015 saw the start of Run 2 for the LHC and the LHC reach a proton-proton collision energy of 13 TeV. By the end of LHC Run 2, it is expected that about 150 fb^{-1} of physics data will have been collected by both ATLAS and CMS. Together with LHCb and ALICE the total size of LHC data is around 1 exabyte, as shown in the table below from the LHC's Computing Resource Scrutiny Group [CRSG2017]. The CPU allocation from the CRSG for 2017 to each experiment is also shown.

Experiment	**2017 Disk Pledges (PB)**	**2017 Tape Pledges (PB)**	**Total Disk & Tape Pledges (PB)**	**CRSG CPU 2017 (kHS06)**
ALICE	67	68	138	807
ATLAS	172	251	423	2194
CMS	123	204	327	1729
LHCb	35	67	102	413
Total	**400**	**591**	**990**	**5143**

Table 1 : Resource requests submitted by the 4 LHC experiments to the September 2017 session of the Computing Resources Scrutiny Group.

Using an approximate conversion from HS06 [HS06] to CPU cores of 10 means that LHC computing in 2017 is supported by about 500k CPU cores. These resources are deployed everywhere from close to the experiments themselves at CERN to a worldwide distributed computing infrastructure, the WLCG. Each experiment has developed its own workload and data management software to manage its share of WLCG resources.

In order to process the data, the 4 large LHC experiments have written more than 12 million lines of program 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_64) and with a serial processing model in mind. There is considerable anxiety in the experiments that much of this software is not sustainable, 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 experiments' data storage 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 ([PYTHIA], [SHERPA], [ALPGEN], [MADGRAPH], [HERWIG], amongst many others). 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.

Even in Run 3 LHCb will process, in software, more than 40 times the number of collisions that it does today, and ALICE will read out Pb-Pb collisions continuously at 50 kHz. The upgrade to the HL-LHC for Run 4 then produces a step change for ATLAS and CMS. The beam intensity will rise substantially, giving bunch crossings where pileup (the number of discrete proton interactions) will rise, from about 60 today, to about 200. The two experiments will upgrade their trigger systems to record 5-10 times as many events as they do today. It is anticipated that HL-LHC will deliver about 300 fb^{-1} of data each year.

The steep rise in resources that are then required to manage this data are estimated from an extrapolation of the Run 2 computing model and are shown in Figures 1 and 2.

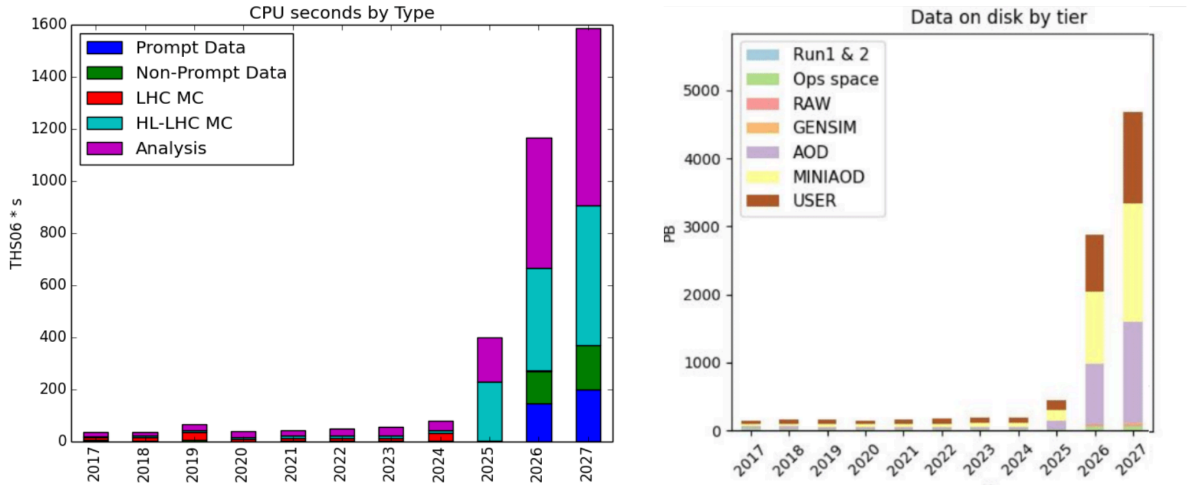

Figure 1. CMS CPU and disk requirement evolution into the first two years of HL-LHC [Sexton-Kennedy2017]

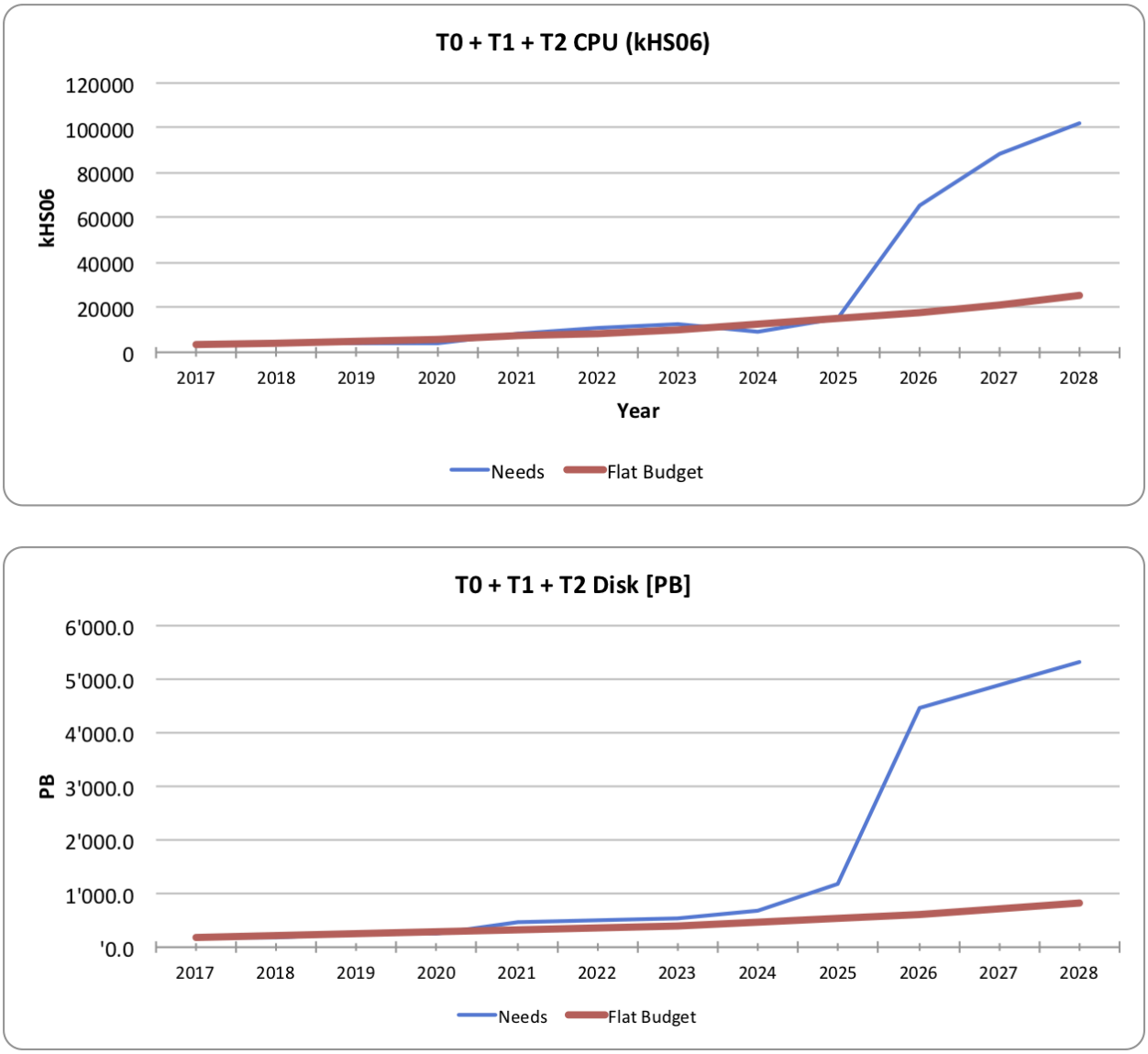

Figure 2. ATLAS CPU and disk requirement evolution into the first three years of HL-LHC [Campana2017]

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, and therefore the physics reach during HL-LHC will be limited by how efficiently these resources can be used.

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: x4 in CPU and x7 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 constraints 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_64 with, for example, Aarch64 devices that may lower the power costs for compute resources better than Intel has achieved with its Xeon architecture. 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. Attracting people with the required skills who can provide leadership is another significant challenge, since it impacts on the need to give adequate recognition to physicists who specialise in software development. This is an important issue that is treated in more detail later in the report.

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 Physics Generators

Scope and Challenges

Monte-Carlo event generators are a vital part of modern particle physics, providing a key component of the understanding and interpretation of experiment data. Collider experiments have a need for theoretical QCD predictions at very high precision. Already in LHC Run 2 experimental uncertainties for many analyses are at the same level, or lower, as those from theory. Many analyses have irreducible QCD-induced backgrounds where statistical extrapolation into the signal region can only come from theory calculations. With future experiment and machine upgrades the uncertainties from data will shrink even further and this will increase the need to reduce the corresponding errors from theory.

Increasing accuracy will compel the use of higher order generators with challenging computational demands. Leading order generators (LO) are only a small part of the overall computing requirements for HEP experiments, but next-to-leading order (NLO) event generation, used more during LHC Run 2, are already using significant resources. By HL-LHC the use of next-to-next-to-leading order (NNLO) event generation will almost certainly be required, which brings formidable computational challenges. Increasing the order of the generators increases greatly the complexity of the phase space integration required to calculate the appropriate QCD matrix elements. The difficulty of this integration arises from the need to have sufficient coverage in a high dimensional space (10-15 dimensions, with numerous peaks) and the fact that many terms in the integration cancel, so that a very high degree of accuracy of each term is required. Memory demands for generators have generally been low and initialisation times have been fast, but the increase in order means that memory consumption becomes important and initialisation times can become very long.

For HEP experiments, matching to final states with a high multiplicity is also needed, as these states become more interesting at higher luminosities and event rates. The (N)NLO matrix elements need to be connected to parton shower algorithms that generate the final states. This process, particularly for high multiplicity, can have a very low efficiency and increases further the computational load needed to generate the necessary number of final state events.

Developments in generator software are mainly done by the HEP theory community. Theorists derive career recognition and advancement from making contributions to theory itself, but not for making improvements in the computational efficiency of generators *per se*. So improving the computational efficiency of event generators, and allowing them to run effectively on resources such as HPCs, will mean engaging with experts in computational optimisation who can work with the theorists who develop generators.

Many event generators in use today were coded decades ago before concurrency was an issue for HEP software. It is a major challenge to modernise much of this software so that it can run

efficiently on modern hardware, in particular Photos, Tauola, EvtGen and Pythia are used by the whole community.

The challenge in the next decade is then to advance the theory and practice of event generators to support the needs of future experiments, reaching a new level of theory precision and recognising the demands for computation and computational efficiency that this will bring.

Current Practice

Extensive use of leading order generators and parton shower algorithms are still made by most HEP experiments. Each experiment has its own simulation needs, but for the LHC experiments 10's of billions of generated events are now used for each Monte-Carlo campaign. During LHC Run 2 more and more NLO generators were used, because of their increased theoretical precision and stability. The raw computational complexity of NLO amplitudes, combined with many-body phase-space evaluations and the inefficiencies of the matching process led to a much increased CPU budget for physics event simulation for ATLAS.

The use of NLO generators by the experiments today is limited because of the way the generators are implemented, producing negative event weights. This means that the total number of events the experiments need to generate, simulate and reconstruct is between x3 to x25 larger for NLO than LO. At the same time, the experiments budget only x2 to x3 more events from simulation than from HLT triggers. Having large NLO samples is thus not consistent with existing computing budgets until a different scheme is developed that does not depend on negative event weights.

While most event generation is run on 'standard' grid resources, effort is ongoing to run more demanding tasks on HPC resources (e.g., W + 5 jet events at the Argonne Mira HPC), however scaling for efficient running on some of the existing HPC resources is not trivial.

Interfaces between event generators and the rest of the experimental apparatus is achieved by standard libraries such as LHAPDF, HepMC and Rivet. These require extensions and sustained maintenance that should be considered a shared responsibility of the theoretical and experimental community in the context of large-scale experiments. In practice, however, it has been difficult to achieve the level of support that is really needed as there has been a lack of recognition for this work.

To help improve the capabilities and performance of generators as used by the experimental HEP programme, and to foster interaction between the communities the MCnet short-term studentship programme has been very useful. Interested experimental PhD students can join a generator group for several months to work on improving a physics aspect of the simulation that is relevant for their work, or to improve the integration of the generator into the experimental framework.

Research and Development Programme

As the MC projects are funded mainly to develop theoretical improvements, and not mainly as “suppliers” to the experimental HEP programme, any strong requests towards efficiency improvements from the experimental community would need to be backed up by plausible avenues of support that can fund contributions from software engineers with the correct technical skills in software optimisation to work within the generator author teams.

In a similar way to the MCnet studentships, a matchmaking scheme could focus on the software engineering side, and transfer some of the expertise available in the experiments or facilities teams to the generator projects. Sustainable improvements are unlikely to be delivered by graduate students “learning on the job” and then leaving after a few months, so to meet the requirement of transferring technical expertise and effort will likely require placements for experienced optimisation specialists and a medium/long-term connection to the generator project.

HEP experiments, which are now very large collaborations including many technical experts, can also play a key role in sustaining a healthy relationship between theory and experiment software. Effort to work on common tools, that benefit both the experiment itself and the wider community, would provide shared value that justifies direct investment from the stakeholders in the community. This model would also be beneficial for core HEP tools like LHAPDF, HepMC and Rivet, where future improvements have no theoretical physics interest anymore, putting them in a similar situation to generator performance improvements. One structural issue blocking such a mode of operation is that experiments do not currently recognise contributions to external projects as experiment service work — a situation deserving of review in areas where external software tools are critical to experiment success.

Specific areas of R&D for event generation on the next 5-10 years are:

- The development of new and improved theoretical algorithms holds out perhaps the largest potential for improving event generators. While it is not guaranteed that simply increasing the effort dedicated to this task will bring about the desired result, the long-term support of event generator development and the creation of career opportunities in this research area are critical given the commitment to experiments on multi-decade scales.
- Expand development in is reweighting of event samples, where new physics signatures can be explored by updating the partonic weights according to new matrix elements. It is necessary that the phase space for the updated model be a subset of the original one, which is an important limitation. Overcoming the technical deficiency of utilising negative event weights is crucial. The procedure is more complex at NLO and can require additional information to be stored in the event files to properly reweight for different cases. Nevertheless, the method can be powerful in many cases and would hugely reduce the time needed for BSM samples.
- At a more technical level concurrency is an avenue that has yet to be explored in depth for event generation. As the calculation of matrix elements requires VEGAS-style integration, this work would be helped by the development of a new Monte-Carlo

integrator. For multi-particle interactions factorising the full phase space integration into lower dimensional integrals would be a powerful method of parallelising, while the interference between different Feynman graphs can be handled with known techniques.

- For widely used generators like Photos, Tauola, EvtGen and Pythia, basic problems of concurrency and thread hostility need to be tackled, to make these packages large scale use on modern processors.
- In most generators, parallelism was added post-facto which leads to scaling problems when the level of parallelism becomes very large, e.g., on HPC machines. These HPC machines will be part of the computing resource pool used by HEP, so solving scaling issues on these resources for event generation is important, particularly as the smaller generator code bases can make porting to non-x86_64 architectures more tractable. The problem of long and inefficient initialisation when a job utilises hundreds or thousands of cores on an HPC needs to be tackled. While the memory consumption of event generators is generally modest, tree level contributions to high multiplicity final states can use significant memory and gains would be expected from optimising here.
- Another underexplored avenue is the efficiency of event generation as used by the experiments. An increasingly common usage is to generate very large inclusive event samples, which are filtered on event final-state criteria to decide which events are to be retained and passed on to detector simulation and reconstruction. This naturally introduces a large wastage fraction of very CPU-expensive event generation, which could be reduced by an emphasis on filtering tools within the generators themselves, designed for compatibility with the experiment requirements. A particularly wasteful example is where events are separated into orthogonal sub-samples by filtering, in which case the same large inclusive sample is generated many times, with each stream filtering the events into a different group: allowing a single inclusive event generation to be filtered into several orthogonal output streams would improve efficiency.

3.2 Detector Simulation

Scope and Challenges

For all its success so far, the challenges faced by the HEP field in the simulation domain are daunting. During the first two runs, the LHC experiments produced, reconstructed, stored, transferred, and analyzed tens of billions of simulated events. This effort required more than half of the total computing resources allocated to the experiments. As part of the high-luminosity LHC physics program (HL-LHC) and through the end of the 2030's, the upgraded experiments expect to collect 150 times more data than in Run 1. The 50 PB of raw data produced in 2016 will grow to approximately 600 PB in 2026 while the CPU needs will increase by a factor of about 60. Demand for larger simulation samples to satisfy analysis needs will grow accordingly. In addition, simulation tools have to serve diverse communities, including accelerator-based particle physics research utilizing proton-proton colliders, neutrino and muon experiments, as well as the cosmic frontier. The complex detectors of the future, with different module- or cell-level shapes, finer segmentation, and novel materials and detection techniques, require additional features in geometry tools and bring new demands on physics coverage and accuracy within the constraints of the available computing budget. The diversification of the physics programmes also requires new and improved physics models. More extensive use of fast simulation is a potential solution, under the assumption that it is possible to improve time performance without an unacceptable loss of physics accuracy.

The gains that can be made by speeding up critical elements of the Geant4 simulation toolkit can be leveraged for all applications that use it and therefore it is well worth the investment in effort needed to achieve it. The main 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 of the order of 100 TeV;
- redesigning, developing and commissioning detector simulation toolkits to be more efficient when executed on emerging computing architectures, such as 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, where the full detector simulation is replaced, in whole or in part, by computationally efficient techniques. Areas of investigation include 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 (pileup);

- 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.

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.

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. 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.

Physics models are a critical part of the detector simulation and are continuously being reviewed, and in some cases reimplemented, in order to improve accuracy and software performance. Electromagnetic (EM) transport simulation is challenging as it occupies a significant part of the computing resources used in full detector simulation. Significant efforts have been made in the recent past to better describe the simulation of electromagnetic shower shapes, in particular to model the H -> γγ signal accurately at the LHC. This effort is being continued with emphasis on reviewing the models assumptions, approximations and limitations, especially at very high energy, and with a view to improving their respective software implementations. In addition, a new “theory-based” model for describing the *multiple scattering* of electrons and positrons has been developed (Goudsmit-Saunderson) that has been demonstrated to outperform, in terms of physics accuracy and speed, the current models in Geant4. The models used to describe the *bremsstrahlung* process have also been reviewed and recently an improved theoretical description of the Landau-Pomeranchuk-Migdal (LPM) effect was introduced which plays a significant role at high energies. Theoretical review of all electromagnetic models, including those of hadrons and ions is therefore of high priority both for HL-LHC and for FCC studies.

Hadronic physics simulation covers purely hadronic interactions. It is not possible for a single model to describe all the physics encountered in a simulation due to the large energy range that needs to be covered and the simplified approximations that are used to overcome the difficulty of solving the full theory (QCD). Currently the most-used reference physics list for high energy and space applications is FTFP_BERT. It uses the Geant4 Bertini cascade for hadron–nucleus

interactions from 0 to 5 GeV incident hadron energy, and the FTF parton string model for hadron–nucleus interactions from 4 GeV upwards. QGSP_BERT is a popular alternative which replaces the FTF model with the QGS model over the high energy range. The existence of more than one model (for each energy range) is very valuable in order to be able to determine the systematics effects related to the approximations used. The use of highly granular calorimeters such as the ones being designed by the CALICE collaboration for future linear colliders, allows a detailed validation of the development of hadronic showers with test-beam data. Preliminary results suggest that the lateral profiles of Geant4 hadronic showers are too narrow. Comparisons with LHC test-beam data have shown that a fundamental ingredient for improving the description of the lateral development of showers is the use of intermediate and low energy models that can describe the cascading of hadrons in nuclear matter. Additional work is currently being invested in the further improvement of the QGS model, which is a more theory-based approach than the phenomenological FTF model, and therefore offers better confidence at high energies, up to a few TeV. This again is a large endeavour and requires continuous effort over a long time.

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. In producing new models of physics interactions and improving the fidelity of the models that exist, it is absolutely imperative that high-quality data are available. Simulation model tuning often relies on test beam data, and a program to improve the library of available data could be invaluable to the community. Such data would ideally include both thin-target test beams for improving interaction models and calorimeter targets for improving shower models. These data could potentially be used for directly tuning Fast Simulation models, as well.

There are specific challenges associated with the Intensity Frontier experimental programme, in particular simulation of the beamline and the neutrino flux. Neutrino experiments rely heavily on detector simulations to reconstruct neutrino energy, which requires accurate modelling of energy deposition by a variety of particles across a range of energies. Muon experiments such as Muon g-2 and Mu2e also face large simulation challenges; since they are searching for extremely rare effects, they must grapple with very low signal to background ratios, and the modeling of low cross-section background processes. Additionally, the size of the computational problem is a serious challenge, as large simulation runs are required to adequately sample all relevant areas of experimental phase space, even when techniques to minimize the required computations are used. There is also a need to simulate the effects of low energy neutrons, which requires large computational resources. Geant4 is the primary simulation toolkit for all of these experiments.

Simulation toolkits do not include effects like charge drift in an electric field or models of the readout electronics of the experiments. Instead, these effects are normally taken into account in a separate step called digitisation. Digitisation is inherently local to a given sub-detector, and often even to a given readout element, so that there are many opportunities for parallelism in terms of vectorisation and multiprocessing or multi-threading, if the code and the data objects are designed optimally. Recently, both hardware and software projects have benefitted from an increased level of sharing among experiments. The LArSoft Collaboration develops and

supports a shared base of physics software across Liquid Argon (LAr) Time Projection Chamber (TPC) experiments, which includes to provide common digitisation code. Similarly, an effort exists among the LHC experiments to share code for modeling of radiation damage effects in silicon. As CMS and ATLAS expect to use similar readout chips in their future trackers, further code sharing might be possible.

The Geant4 simulation toolkit will continue to evolve over the next decade. This evolution will include contributions from various R&D projects as described in the following section. The overriding requirement is to ensure the support of experiments through continuous maintenance, support and improvement of the Geant4 simulation toolkit with minimal API changes visible to these experiments at least until production versions of potentially alternative engines, such as those resulting from ongoing R&D work, become available, integrated, and validated by experiments. The agreed ongoing strategy to meet this goal is to ensure that new developments resulting from the R&D programme can be tested with realistic prototypes and then be integrated, validated and deployed in a timely fashion in Geant4.

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 each component of the simulation software for the longer term. The R&D programme summarised here is organised by topic. More details about each activity can be found in the full Simulation CWP document.

Particle Transport and Vectorisation

One of the most ambitious elements of the simulation R&D 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 improved data locality, is expected to yield significant speed-ups, which are to be measured in a realistic prototype currently under development.

For the GeantV transport engine to display its best computing performance, it is necessary to vectorise and optimise the accompanying modules, including geometry, navigation, and the physics models. They are developed as independent libraries so that they can also be used together with the current Geant4 transport engine. Of course, when used with the current Geant4 they will not expose their full performance potential, since transport in Geant4 is currently sequential, but this allows for a preliminary 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 *alpha* release of the GeantV transport engine, expected at the

end of 2017, will serve as a preview of the new particle propagation approach and will be used to demonstrate many of its features.

- 2019: the *beta* release of the GeantV transport engine will 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 most of the components, including physics modelling for electrons, gammas and positrons, and high performance hadronic interactions, whilst still maintaining simulation reproducibility and demonstrating efficient concurrent I/O and multi-event user data management.

Modularisation

Starting from next release a modularization of Geant4 is being pursued that will allow an easier integration in experimental frameworks, with the possibility to include only the Geant4 modules that are actually used. A further use case is the possibility to use one of the Geant4 components in isolation, e.g. to use hadronic interaction modeling without kernel components from a fast simulation framework. As a first step a preliminary review of libraries granularity is being pursued, which will be followed by a review of intra-library dependencies with the final goal of lose their dependencies.

- 2019: Redesign of some Geant4 kernel components to improve the efficiency of the simulation on HPC systems, starting from improved handling of Geant4 *database*s on large core-count systems. A review will be made of the multi-threading design to be closer to the task-based frameworks, such as TBB.

Physics Models

Developing new and extended physics models to cover extended energy and physics processing of present and future colliders, Intensity Frontier experiments and direct dark matter search experiments. The goal is to extend the missing models (e.g. neutrino interactions), improve models' physics accuracy and, at the same time, improve CPU and memory efficiency. The deliverables of these R&D efforts include physics modules that produce equivalent quality physics and will therefore require extensive validation in realistic applications.

- 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

The experiment applications are essential for validating the software and physics performance of new versions of the simulation toolkit. CMS and ATLAS have already started to integrate

Geant4 multi-threading capability in their simulation applications; in the case of CMS first Full Simulation production in multi-threaded mode was delivered in the Fall of 2017. Specific milestones are as follows:

- 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 the beta release of the GeantV transport engine in the experiments' simulation, including the implementation of the new user interfaces, will allow to make the first performance measurements and physics validation.
- 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 in 2026.

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 minimum 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. Collecting calibration and alignment data at the start of a new Run would necessarily incur delays such that this approach is mainly of use in the final analyses. The experiments are expected to invest in the development of the zero-bias overlay approach by 2020.
- The baseline 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.

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 extent of a common Fast Simulation infrastructure applicable to the variety of detector configurations.

Digitization

It is expected that, within the next 3 years, common digitisation efforts are well-established among experiments and advanced high-performance generic digitisation examples, which experiments could use as a basis to develop their own code, become available.

- 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. Potential use of C++11 PRNG interfaces and implementations, and their extension for our further requirements (output of multiple values, vectorization) will be investigated.
- 2022: promote a transition to the use of this library to replace existing implementations in ROOT and Geant4.

3.3 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, amongst other topics, the evaluation of the most important components of next generation algorithms and data structures 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 the potential for a dramatic increase in physics reach, at the price of an 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 transverse momentum (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.

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 the increase in data rates without loss of physics capability. Projections to future needs, such as for the HL-LHC, show the need for a substantial increase of resources, without significant changes in approach or algorithms, up to a scale not compatible with the foreseen budget constraints.

For this reason, 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 proton-proton 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 to 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 what 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 an opportunity to move beyond commodity x86_64 technologies, which HEP has used very effectively over the past 20 years, and a significant challenge to derive sufficient event processing throughput per cost to reasonably enable our physics programmes [Bird2014]. Among these challenges, important 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. Also, code quality demands increase as traditional offline analysis components migrate into trigger systems, or more generically into algorithms that can only be run once.

Current Practices

Substantial computing facilities are in use for both online and offline event processing across all experiments surveyed. In most experiments, online facilities are dedicated to the operation of the software trigger, but a recent evolution has been to use them opportunistically for offline processing too, when the software trigger does not make them 100% busy, On the other hand, offline facilities are shared for operational needs including event reconstruction, simulation (often the dominant component in several experiments) and analysis. CPU in use by experiments is typically at the scale of tens or hundreds of thousands of x86_64 processing cores.

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 centralised 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].

Critical for reconstruction, calibration and alignment systems generally implement a high level of automation in all experiments, both for very frequently updated measurements and more rarely updated measurements. Automated procedures are often 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. The automatisation techniques are often most critical for a subset of precision measurements rather than for the entire physics programme of an experiment.

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 specialised 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. In addition, for the most likely scenario, it is necessary to assess and minimize possible additional costs coming from the maintenance of multiple implementations of the same algorithm on different architectures.

- 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 the 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 R&D 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.

3.4 Data Analysis and Interpretation

Scope and Challenges

HEP answers scientific questions by analysing the data obtained from detectors and sensors of suitably designed experiments, comparing measurements with predictions from models and theories. Such comparisons, which generally involve the use of simulated data to correct for inefficiencies largely due to experimental effects, are performed typically long after the data taking but they can sometimes be executed also in quasi-real time on selected samples of reduced size. The challenges of such an analysis flow, intrinsically different from the ones addressed here, are discussed in the chapter on “Software Trigger and Event Reconstruction”.

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 reaching a compact dataset for 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, for example, 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.

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 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: a “bridge” enables the ecosystem to use an open source analysis tool; a “ferry” allows the use of data from the ecosystem in the tool, and vice versa. 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.

Reproducibility is the cornerstone of scientific results. 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. A challenge specific to data analysis and interpretation is tracking the evolution of and relationships between all the different aspects of an analysis.

Robust methods for data reinterpretation are also critical. Collaborations typically interpret results in the context of specific models for new physics searches and sometimes reinterpret those same searches in the context of alternative theories. However, understanding the full implications of these searches requires the interpretation of the experimental results in the context of many more theoretical models than are currently explored at the time of publication.

Analysis reproducibility and reinterpretation strategies, need to be considered in all new approaches under investigation so that they become a fundamental component of the system as a whole.

The rapidly evolving landscape of software tools, methodological approaches to data analysis and available infrastructures, requires effort in continuous training, both for novices as well as for experienced and mature researchers, as detailed in the Careers and Training section. The maintenance and sustainability of the current analysis ecosystem also presents a major challenge. Currently, this effort is provided by just a few institutions. Legacy and less-used parts of the ecosystem need to be managed appropriately. New policies are needed to retire little used or obsolete components and free up effort for the development of new components. These new tools should be made attractive and useful to a significant part of the community to attract new contributors.

Current Practices

Methods for analysing HEP data have been developed over many years and successfully applied to produce physics results, including more than 1000 publications, during LHC Runs 1 and 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 for an analysis) to be delivered locally to the user. An often stated aim of the data reduction steps is to arrive at a dataset that “can fit on a 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. At the same time, disk space requirements are usually a key constraint of the experiment computing models, as disk is the most expensive hardware component. The LHC experiments have made a continuous effort to produce optimized data analysis-oriented formats with enough information to avoid the need to use intermediate formats.

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 open source 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 as well as increasing their sustainability. Increased HEP investment is needed to allow Python to become a first class supported language.

One new model of data analysis, developed outside of HEP, maintains the concept of sequential ntuple reduction but mixes interactivity with batch processing. Today, Apache Spark is the leading contender for this type of analysis, as it has a well developed ecosystem with many open-source tools contributed both by the industry and the data-science community. Other products implementing the same analysis concepts and workflows are emerging, such as TensorFlow, Dask, Pachyderm, Blaze, Parsl and Thrill. 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.

An alternative approach, which emerged in the Big Data world and is now widely used in several contexts, is 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 is 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 can 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. It is of interest for the HEP community to understand the role that such an approach could have in the global analysis infrastructure and if it can bring an optimisation of the the global storage and computing resources required for the processing of raw data to analysis.

Another active area regarding analysis in the world outside HEP is the switch to a functional or declarative programming model, as for example provided by Scala in the Spark environment. This allows scientists to express the intended data transformation as a query on data. Instead of having to define and control the “how”, the analyst declares 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. 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 paradigm.

Research and Development Programme

Towards HL-LHC, we envisage dedicated data analysis facilities for experimenters, offering an extendable environment that can provide fully functional analysis capabilities, integrating all these technologies relevant for HEP. Initial prototypes of such analysis facilities are currently underdevelopment. On the time scale of HL-LHC, such dedicated Analysis Facilities would provide a complete system engineered for latency-optimisation and stability.

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. In particular, prototype a Spark-like analysis facility that could be a shared resource for exploratory data analysis and batch submission.
- Complete advanced prototype of a low-latency response, high-capacity analysis facility incorporating fast caching technologies to explore a query-based analysis approach. It should in particular include an evaluation of additional storage layers, such as SSD storage and NVRAM-like storage.
- Expand support of Python in our ecosystem with a strategy for ensuring long term maintenance and sustainability. In particular in ROOT, the current C++ and Python binding through PyROOT, should evolve to reach the ease of use of native Python modules.
- Prototype a comprehensive set of “bridges” and “ferries” (as defined above).
- Develop a prototype model based on functional or declarative programming model for data analysis.
- Conceptualize and prototype analysis “Interpretation Gateway”, including data repositories, and recasting tools.

By 2022:

- Evaluate chosen architectures for analysis facility, verify their design and provide input for corrective actions to test them at a larger scale during Run 3.
- Develop a blueprint for remaining analysis facility developments, system design and support model.

3.5 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 datasets. 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 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 well onto existing paradigms 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 of these have been used productively in HEP for more than 20 years, others have been introduced relatively recently. The portfolio of ML techniques and tools is in constant evolution and, in particular, Deep Learning (DL) techniques look very promising for our field. A key feature of ML algorithms is that most have well documented open source software implementations. 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 using ML algorithms compared to using more traditional software? What are the trade-offs of one approach compared to another? These issues are not necessarily “factorisable”, and a key goal will be to ensure that, as HEP research teams investigate the numerous approaches at hand, the expertise acquired, and lessons learned, get adequately disseminated to the wider community. In general, each *team* - typically a small group of scientists from a collaboration - will serve as a repository of expertise. Beyond the R&D projects it sponsors directly, each team should help others develop and deploy experiment-specific ML-based algorithms in their software stacks. It should 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, the ML toolset will be used to develop application software that could, amongst other things:

- 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 a collider experiment reconstruction software stack. The algorithms employed in these tasks 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 recognise, classify, and quantitatively describe objects. We wish to investigate how to replace the most computationally 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.

As already discussed, all HEP detectors produce much more data than can be moved to permanent storage. The process of reducing the size of the datasets is managed by the trigger system. Electronics usually sparsify the data stream using zero suppression and apply basic data compression. While this typically reduces the data rate by a factor of 100 or more, to about 1 terabyte per second for ATLAS and CMS, 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. At the LHC the challenge will increase both quantitatively and qualitatively as the number of proton-proton collisions per bunch crossing increases. A more detailed discussion on the specifics and types of trigger system is provided in section 3.11, where the so-far unique trigger system of LHCb is further compared to more traditional systems such as the ones briefly outlined here.

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 [TMVA] included in ROOT. Recently, however, many HEP analysts have begun migrating to non-HEP ML packages such as scikit-learn [scikit-learn] and Keras [Keras]. Data scientists at Yandex created a Python package that provides a consistent API to most ML packages used in HEP [REP]. Packages like Spearmint perform Bayesian optimisation and can improve HEP Monte Carlo work. This shift in the set of ML techniques and packages utilised is especially strong in the neutrino physics community, where new experiments such as DUNE place ML at the very heart of their reconstruction algorithms and event selection.

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.

ML algorithms can often discover patterns and correlations more powerfully than human analysts. This allows qualitatively better analyses of recorded datasets. 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 quarks, 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. On the other hand, LHCb’s physics goals led the collaboration investigate and exploit ML algorithms that are minimally biased with respect to the physical observables of interest. As the programme moves further into the study of increasingly precise measurements, work of this kind will become more important.

While each experiment has, or is likely to have, different specific use cases, we expect that many of these will be sufficiently similar to each other that R&D 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 problems. This is why the Inter-experiment Machine Learning forum (IML [IML]) has been created at CERN in 2016. It already demonstrated the benefits of the collaboration between (LHC and non-LHC) experiments around Machine Learning.

Research and Development Roadmap and Goals

The R&D roadmap presented here is based on the preliminary work done in recent years, coordinated by the HSF IML, which will remain the main forum to coordinate actions about ML in HEP and ensure the proper links with the data science communities. The following programme of work is foreseen.

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 task. 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 new flagship neutrino programme DUNE. A proof of concept and comparison of DL architectures should be finalised by 2020.

- ML middleware and data formats: HEP is currently mainly relying on the ROOT format for its data when the ML community has developed several other 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 the entire dataset into RAM, compression and widespread adoption 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).
- 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 as a Service (MLaaS): current cloud providers rely on a MLaaS model allowing for efficient use of common resources and use of interactive machine learning tools. MLaaS is not yet widely used in HEP. 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's SWAN will be identified.

By 2022:

- Detector anomaly detection: data taking in complex HEP experiments is continuously monitored by physicists taking shifts to assess the quality of the incoming data, largely using reference histograms produced by experts. This makes it difficult to anticipate new problems. A whole class of ML algorithms called anomaly detection can be useful for such problems. Such unsupervised algorithms are able to learn from data and produce an alert when deviations are observed. 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. These 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 in simulation saw orders of magnitude increase 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 prototypes to use DL fast inference in online systems. To deal with the increasing event complexity at HL-LHC, we will also explore the use of sophisticated ML algorithms at all trigger levels, building on the pioneering work by the LHCb collaboration, see section 3.11.

- 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. The fact of being very computationally intensive has limited its applicability in HEP so far. Using neural networks for numerical integrations is not new. The technical challenge lies in the design of a network sufficiently rich to encode 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 always is a computationally challenging step. It becomes an overwhelming challenge in the HL-LHC environment. Adequate ML techniques may provide a solution that scales linearly with LHC intensity. An effort called HEP.TrkX [HEP.TrkX] has started to investigate DL algorithms such as long-term short-term (LSTM) networks for track pattern recognition on many-core processors.

3.6 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. Both technology and large increases in data volume require new computational models [Butler2013], compatible with budget constraints (typically a flat budget scenario), which need to be proactively investigated. The integration of newly emerging data analysis paradigms into a new computational model 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 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. Other experiments close to data taking have similar needs, e.g., Belle II has the same data volumes as ATLAS. The storage requirements per year are then expected to jump by a factor close to 10 for the HL-LHC. This growth rate is faster than projected technology gains and will present major challenges. Storage will remain one of the major cost drivers for HEP computing, at a level roughly similar to the cost of the computational resources. 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 experiment 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, provide new possibilities. 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 summarised as follows:

- The HEP experiments of the HL-LHC era will significantly increase both the data rate and the data volume. The computing systems will need to handle this with as small a cost increase as possible 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 computing

resources (cloud, HPC) with different dynamic availability and characteristics 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 future experiments. 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 specialised processor resources such as GPUs, placing further requirements on data.

In particular, the projected event complexity of data from future HL-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 data. The precursors of these tools, in the form of new pattern recognition and tracking algorithms based on machine-learning techniques, are already proving to be drivers for the compute needs of the HEP community. 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 compute resources, the landscape of storage solutions 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 effective data access . We suggest investigating either separate direct access-based archives (e.g. disk or optical) or new models that hierarchically 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 current practices will not scale to the needs of HL-LHC and other major HEP experiments of the HL-LHC era.

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 four different aspects of interacting with data, namely data organisation, data management, data access and data granularity.

- *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 centres. Originally, computing models were rather static: data was placed at sites and the relevant compute jobs were sent to the right locations. Since LHC startup, this model has been made more flexible to limit the non-optimal pre-placement and to take into account data popularity. In addition, applications might interact with catalogs or, at times, the workflow management systems does this on behalf of the applications.

- D*ata access*: historically various protocols have been used for direct reads (rfio, dcap, xrootd, etc.) where jobs are reading data explicitly staged-in or cached by the compute resource used or the site it belongs to. A recent move has been the convergence towards xrootd as the main protocol for direct access. With direct-access, applications may use different protocols than those used by data transfers between sites. In addition, LHC experiments have been increasingly using remote access to the data, without any stage-in operations, using the possibilities offered by protocols like rfio or http.

- D*ata granularity*: the data is split into datasets, defined by physics selections and use-cases, consisting of a set of individual files. While individual files in datasets can be processed in parallel, the files themselves are usually processed as a whole.

Before the LHC turn-on and in the first years of the LHC, these four areas were to first order

optimised independently. As LHC computing matured interest has turned to optimisations spanning multiple 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 section we take a broader view than traditional data management, 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.

- Sub-file, e.g., 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.
- We will seek to derive benefits from data organisation and analysis technologies adopted by other big data users. 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 versus row-wise organisation of data on the performance of each kind of access.
    b. See how an efficient data storage and access solution may look like that supports the use of map-reduce or Spark-like analysis services
    c. Evaluate just-in-time decompression schemes and mappings onto hardware architectures considering the flow of data, from spinning disk to memory and application
- Discover the role data placement optimisation, for example caching, can play in order to use computing resources effectively and the technologies that can be used. The

following tasks should be completed by 2020:

a. Quantify the benefit of placement optimisation 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 and follow-up with the technology evolution outside HEP itself.

In the longer term, it is planned 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 Content Delivery Networks (CDN) and Named Data Networking (NDN).

- 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.

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. This R&D will be done in common with the related actions planned as part of Facilities and Distributed Computing.

3.7 Facilities and Distributed Computing

Scope and Challenges

As 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 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 indicate that the current system would need almost an order of magnitude more resources than will be available from technology evolution. In addition, other initiatives such as Belle2 and Dune in HEP but also SKA will require a comparable amount of resources on the same infrastructure. 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, in particular for simulation; 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 specialised 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. While CPU and disk capacity is expected to increase by respectively 20% and 15% per year for the same cost, the trends of research network capacity increases show a much steeper growth such as two orders of magnitude from now to HL-LHC. Therefore, the evolution of the computing models would need to be more network centric.

In the network domain there are new technology developments, like Software Defined Networks (SDNs), 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 use.

Underlying storage system technology will continue to evolve, for example towards object stores, and, as proposed in the DOMA section, R&D is also necessary to understand their usability and their role in the HEP infrastructures. There is also the continual challenge 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) will 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. Even if HL-LHC is the use case used to build this cost and performance model, because the ten years of experience running large-scale experiments helped to define the needs, it is believed that this work and the resulting model will be valuable for other upcoming data intensive scientific initiatives. This includes future HEP projects such as Belle II, DUNE and possibly ILC experiments but also non HEP ones, such as SKA.

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 at large national facilities, the WLCG Tier-1s [Bird2014]

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 several concepts and a few lower 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.

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 (CRSG), 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 450PB 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. Several of the items below require some coordination with other topical areas discussed in this document and some work is still needed to finalize the detailed action plan. These actions 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 work requires a coordination with the Data Processing Frameworks WG. It 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.
- Explore 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 Data-Flow Processing Framework

Scope and Challenges

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

To fully exploit the potential of 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 computing facilities, including multi-core 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. 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.

Common developments across experiments are desirable in this area but are hampered by many decades of legacy work. Evolving our frameworks has also 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. These frameworks can also be constrained by security requirements, mandated by the groups and instances in charge of it.

Current Practices

Although most frameworks used in HEP share common concepts, there are for mainly historical reasons 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 and various non-LHC experiments. CMS uses its own CMSSW framework, which was forked to provide the art framework for the Intensity Frontier experiments. Belle II uses basf2. The linear collider community developed and uses Marlin. The FAIR experiments use FairROOT, closely related with ALICE's AliROOT. Both experiments are now developing together a new framework, which is called O2. At the time of writing, all major frameworks support basic parallelization, both within and across events, based on a task-based model.

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 trade-offs between constraints in the programing model, efficiency of execution, and ease of adapting to inhomogeneous resources. These concerns also reflect two different behaviours: firstly maximising throughput, where it is most important to maximise the number of events that are processed by a given resource; secondly minimising latency, where the primary constraint is on how long it takes to calculate an answer for a particular datum.

Current practice for throughput-maximising system architectures have constrained the scope of framework designs. Framework applications have largely been viewed by the system as a batch job with complex configuration, consuming resources according to rules dictated by the computing model: one process using one core on one node, operating independently with a fixed size memory space on a fixed set of files (streamed or read directly). Only recently has CMS broken this tradition starting at the beginning of Run 2, by utilising all cores on one virtual node in one process space using threading. ATLAS is currently using a multi-process fork-and-copy-on-write solution to remove the constraint of one core/process. Both experiments were driven to solve this problem by the ever growing needs for more memory per process brought on by the increasing complexity of LHC events. Current practice manages system-wide (or facility-wide) scaling by dividing up datasets, generating a framework application configuration, and scheduling jobs on nodes/cores to consume all available resources. Given anticipated changes in hardware (heterogeneity, connectivity, memory, storage) available at large computing facilities, the interplay between workflow/workload management systems and framework applications need to be carefully examined. It may be advantageous to permit framework applications (or systems) to span resources, permitting them to be first-class participants in the business of scaling within a facility.

Research and Development programme

2018: review of existing technologies that are the important building blocks for the data processing frameworks and agreement on the main architectural concepts for the next generation of frameworks. Community meetings and workshops, along the lines of the original Concurrency Forum, are envisaged to help foster collaboration in this work [ConcurrencyForum]. This includes in particular:

- Libraries used for concurrency, their likely evolution and the issues in integrating the models used by detector simulation and physics generators into the frameworks.
- Functional programming as well as domain specific languages as a way to describe the physics data processing that has to be undertaken rather than how it has to be implemented. This approach is based on the same concepts as the idea for functional approaches for (statistical) analysis as described in the respective section.
- Analysis of the functional differences between the existing frameworks and the different experiment use-cases.

By 2020: prototype and demonstrator projects on the agreed architectural concepts and baseline to inform the HL-LHC Computing TDRs and to demonstrate advances over what is currently deployed. The following specific items will have to be taken into account:

- These prototypes should be as common as possible between existing frameworks or at least several of them as a proof-of-concept of effort and component sharing between frameworks for their future evolution. Possible migration paths to more common implementations will be part of this activity.
- In addition to covering the items mentioned for the review phase, they should particularly demonstrate possible approaches for scheduling the work across heterogeneous resources and using them efficiently, with a particular focus on the efficient use of co-processors, e.g. GPGPUs.
- They need to identify data model changes that are required for an efficient use of new processor architectures (.e.g. vectorisation) and for scaling I/O performances in the context of concurrency.
- Prototypes of a more advanced integration with workload management, taking advantages in particular of the advanced features available at facilities for a finer control of the interactions with storage and network and dealing efficiently with the specificities of HPC resources.

By 2022: production-quality framework libraries usable by several experiment frameworks, covering the main areas successfully demonstrated in the previous phase. During these activities, we expect at least one major paradigm shift to take place on this 5-year time scale. It will be important to continue discussing their impact within the community. This will be ensured through appropriate cross-experiment workshops dedicated to the data processing frameworks.

3.9 Conditions Data

Scope and Challenges

Conditions data is defined as the non-event data required by data-processing software to correctly simulate, digitise or reconstruct the raw detector event data. The non-event data discussed here consists mainly of detector calibration and alignment information, with some additional data describing the detector configuration, the machine parameters, as well as 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 and its advanced snapshotting and branching features. ALICE have made some promising tests and started to use this approach in Run 2; Belle II already took the same approach [WoodACAT2017] and 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.

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 potentially continuously decreasing luminosity, leading to an interval granularity down to the order of a minute. Insufficient or inefficient caching may impact the efficiency of the reconstruction processing.

Another important challenge is ensuring the long-term maintainability of the conditions data storage infrastructure. Shortcomings in the initial approach used in LHC Run 1 and Run 2, leading to complex implementations, helped to identify the key requirements for an efficient and sustainable condition data handling infrastructure. There is now a consensus among experiments on these requirements: ATLAS and CMS are working on a common next-generation conditions database and Belle II, about to start its data-taking, has developed a

solution based on the same concepts and architecture. One key point in this new design is to have a server mostly agnostic to the data content with most of the intelligence on the client side. This new approach should make easier to rely on well established open-source products (e.g. Boost) or software components developed for the processing of event data (e.g. CVMFS). With such an approach, 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 will be able to use multiple 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 increased 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 a full online reconstruction, where the calibrations and alignment are computed and applied in the HLT. This is currently being tested by ALICE and LHCb who will adopt it as their base design in Run 3. If it will offer an opportunity to a separate

distribution of conditions data to reconstruction jobs and analysis jobs, it would put an increased pressure on the access efficiency to the conditions data in the context of the HLT.

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.
- ATLAS and LHCb: 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. ALICE is also planning to explore this approach for the future, as an alternative or to complement the current CVMFS-based implementation.

3.10 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 processing chain: detector development, monitoring, event generation, reconstruction, detector simulation, data analysis, as well as outreach and education.

Event displays are the main tool to explore experimental data at the event level and to visualise the detector itself. There are two main types of applications: firstly those integrated in the experiments' frameworks, which are able to access and visualise all the experiment's data, but at a cost in terms of complexity and portability; secondly 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 detail and is an extra processing step. In addition there are 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 uses visualisation of statistical information, typically histograms, which allow the analyst to quickly characterise the data. Unlike event displays, these visualisations are not strongly linked to the detector geometry, and often aggregate data from multiple events. Other types of visualisations are used to display non-spatial data, such as graphs for describing the logical structure of the detector or graphs that illustrate dependencies between the data products of different reconstruction algorithms.

The main challenges in this domain are in the sustainability of the many experiment-specific visualisation tools, when common projects could reduce duplication and increase quality and long term maintenance. 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, such as mobile phones. Making a good choice for the libraries used to render 3D shapes is also key, impacting on the range of output devices that can be supported and the level of interaction with the user. 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 non-event visualisation, where many open source and industry standard tools can be exploited.

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 such as jets, tracks) or to the experimental conditions (e.g. detector descriptions, calibrations).
- Geometry visualisation: 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.

The visualisation of data 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, such as geometry, and offers a higher level API that simplifies development. However, this carries the risk 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 has been the tool of choice in HEP for many years and satisfies most use cases. However, increasing use of generic tools and data formats mean Matplotlib (Python) or JavaScript based solutions (used for example in Jupyter notebooks) have made the landscape more diverse. For visualising trees or graphs, there are many generic offerings.

Research and Development Roadmap

The main goal of R&D projects in this area will be to develop techniques and tools that 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.

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.

3.11 Software Development, Deployment, Validation and Verification

Scope and Challenges

Modern HEP experiments are often 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, architectural and performance best practices.
- Software sustainability, including both development and maintenance efforts, as well as best practices given long timescales of HEP experiments.
- 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 HEP 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 university groups and HEP-focused 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. Clear guidance and support for good design must be available in advance of actual coding.

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 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. Further validation on different platforms and at large scales must be as automated as possible, including the deployment of build artefacts for production.

Training and documentation is key to efficient use of developer effort (see also the later chapter on Training). Documentation must cover best practices and conventions as well as technical issues. 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 [Ref:SSI2017]. 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. However, this practice is not wide-spread enough in the community and needs further encouragement.

Research and Development Programme

HEP must endeavour 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 timescale 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, 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* [LHCbStarterkit] and ALICE *Juniors*, and with established generic training initiatives such as *Software Carpentry* [SoftwareCarpentry].

- 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.
- Strengthen the publication record for important HEP software packages. Both peer-reviewed journals [SSI2017] and citeable software version records (such as DOIs obtained via Zenodo [Zenodo]).

Longer term projects 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.
- Support the adoption of industry standards and solutions over HEP-specific implementations whenever possible.
- Develop tooling and instrumentation to measure software performance where tools with sufficient capabilities are not available from industry, 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 ones.
- 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.

3.12 Data and Software Preservation

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 whatever 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 [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, as happened with the Kaggle ML challenges. 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 Analysis Object Data (AOD) format that can be used for new analyses. The current 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 is willing to make access to reconstructed data available but is unable to commit to a specific timescale due to resource limitations. A release of ntuple-level data for one high profile analysis, aimed primarily at educational activities, is currently in preparation. 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.

The LHC experiments have not yet set a formal policy addressing the new capabilities of the CERN Analysis Portal and whether or not some use of it will be required or 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 feasibility and cost of common base services has been studied for the bit preservation, the preservation of executable software environments, and the structured capturing of analysis meta-data (Berghaus2016).

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 on the timescale of 2020 are:

- Include embedded elements for the capture of preservation information and metadata and tools for the archiving of this information in developing prototype analysis ecosystem(s). 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 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, in the preservation area, the full potential and limitations of sandbox and “freezing” technologies, possibly coupled with version and history control software distribution systems.
- Develop prototypes for the preservation and validation of large-scale production executables and workflows.
- Integrate preservation capabilities into newly developed computing tools and workflows.

This would then lead naturally to deployed solutions that support data preservation in the 2020-2022 time frame for the HEP experimental programmes, 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.

3.13 Security

Scope and Challenges

Security is a cross-cutting area that impacts our, projects, collaborative work, users and software infrastructure fundamentally. It crucially shapes our reputation, our collaboration, the trust between participants and the users' perception of the quality and ease of use of our services.

There are three key areas:

- Trust & policies, including trust models, policies, compliance, data protection issues.
- Operational security, including threat intelligence, security operations, incident response.
- Authentication & Authorisation, including identity management, identity federation, access control.

Trust and policies

Data Protection defines the boundaries that enable HEP work to be conducted, in particular regarding data sharing aspects, for example between the EU and the US. It is essential to establish a trusted personal data exchange framework, minimising the amount of personal data to be processed and ensuring legal compliance.

Beyond legal compliance and best practice, offering open access to scientific resources and achieving shared goals requires prioritising the protection of people and science, including the mitigation of the effects of surveillance programs on scientific collaborations.

On the technical side, it is necessary to adapt the current, aging trust model and security architecture relying solely on X.509 (which is not the direction the industry is taking), in order to include modern data exchange design, for example involving commercial providers or hybrid clouds. The future of our infrastructure involves increasingly diverse resource providers connected through cloud gateways. For example, HEPCloud [HEPCloud] at FNAL aims to connect Amazon, Google Clouds and HPC centers with our traditional grid computing resources. The HNSciCloud European Project [HNSciCloud] aims to support the enhancement of commercial cloud providers in order to be leveraged by the scientific community. These are just two of a number of endeavours. As part of this modernisation, a transition is needed from a model in which all participating organisations are bound by custom HEP security policies to a more flexible approach where some partners are not in a position to adopt such policies.

Operational security and threat intelligence

As attacks have become extremely sophisticated and costly to defend against, the only cost-effective strategy is to address security threats together, as a community. This involves

constantly striving to liaise with external organisations, including security vendors and law enforcement entities, to enable the sharing of indicators of compromise and threat intelligence between all actors. For organisations from all sectors, including private companies, governments and academia, threat intelligence has become the main means by which to detect and manage security breaches.

In addition, a global forum for HEP and the larger Research & Education community needs to be built, where security experts feel confident enough to share threat intelligence and security expertise. A key to success is to ensure a closer collaboration between HEP security contacts and campus security. The current gap at many HEP organisations is both undermining the community's security posture and reducing the effectiveness of the HEP security strategy.

There are several very active trust groups in the HEP community where HEP participants share threat intelligence and organise coordinated incident response, including but not limited to:

- Chinese Security Federation.
- The European Grid Infrastructure Computer Security Incident Response Team (EGI-CSIRT) [EGI-CSIRT].
- REN-ISAC [REN-ISAC].
- XSEDE [XSEDE] Security Team.

There is unfortunately still no global Research and Education forum for incident response, operational security and threat intelligence sharing. With its mature security operations and dense, global network of HEP organisations, both of which are quite unique in the research sector, the HEP community is ideally positioned to contribute to such a forum and to benefit from the resulting threat intelligence, as it has the exposure, sufficient expertise and connections to lead such an initiative. It may play a key role in protecting multiple scientific domains at a very limited cost.

There will be many technology evolutions as we start to take a serious look at the next generation internet. For example, IPv6 is one upcoming change that has yet to be fully understood from the security perspective. Another high impact area is the internet of things, connected devices on our networks that create new vectors of attack.

It will become necessary to evaluate and maintain operational security in connected environments spanning public, private and hybrid clouds.The trust relationship between our community and such providers has yet to be determined, including the allocation of responsibility for coordinating and performing vulnerability management and incident response. Incompatibilities between the e-Infrastructure approach to community based incident response, and the "pay-for-what-you-break" model of certain commercial companies, may come to light and must be resolved.

Authentication & Authorisation Infrastructure

It is now largely acknowledged that end-user certificates are challenging to manage and create a certain entrance barrier to our infrastructure for early career researchers. Integrating our access control management system with new, user-friendly technologies and removing our dependency on X.509 certificates is a key area of interest for the HEP Community.

An initial step is to identify other technologies that can satisfy traceability, isolation, privilege management and other requirements necessary for HEP workflows. The chosen solution should prioritise limiting the amount of change required at our services, and follow accepted standards to ease integration with external entities such as commercial clouds and HPC centres.

Trust federations and inter-federations, such as the R&E standard eduGAIN [eduGAIN], provide a needed functionality for Authentication. They can remove the burden of identity provisioning from our community and allow users to leverage their home organisation credentials to access distributed computing resources. Although certain web based services have enabled authentication via such federations, uptake is not yet widespread. The challenge remains to have the necessary attributes published by each federation to provide robust authentication.

The existing technologies leveraged by identity federations, e.g. SAML, have not supported non-web applications historically. There is momentum within the wider community to develop next generation identity federations that natively support a wider range of clients. In the meantime there are several viable interim solutions that are able to provision users with the token required to access a service (such as X.509) transparently, translated from their home organisation identity.

Although federated identity provides a potential solution for our challenges in Authentication, Authorisation should continue to be tightly controlled by the HEP community. Enabling Virtual Organisation (VO) membership for federated credentials and integrating such a workflow with existing identity vetting processes is a major topic currently being worked on, in particular within the WLCG community. Commercial clouds and HPC centers have fundamentally different access control models and technologies from our grid environment. We shall need to enhance our access control model to ensure compatibility and translate our grid-based identity attributes into those consumable by such services.

Current Activities

Multiple groups are working on policies and establishing a common trust framework, including the EGI Security Policy Group [EGISecurityPolicyGroup] and the Security for Collaboration among Infrastructures working group [SCI_WG].

Operational security for the HEP community is being followed up in the WLCG working group on Security Operations Centres [WLCG-SOC-WG]. The HEP Community is actively involved in multiple operational security groups and trust groups, facilitating the exchange of threat intelligence and incident response communication. WISE [WISE] provides the forum for e-Infrastructures to share and develop security best practices and offers the opportunity to build

relationships between security representatives at multiple e-Infrastructures of interest to the HEP community.

The evolution of Authentication and Authorisation is being evaluated in the recently created WLCG Working Group on Authorisation [WLCG-AUTH-WG]. In parallel, HEP is contributing to a wider effort to document requirements for multiple Research Communities through the work of FIM4R [FIM4R]. CERN's participation in the European Authentication and Authorisation for Research and Collaboration (AARC) project [AARC] provides the opportunity to ensure that any directions chosen are consistent with those taken by the great community of research collaborations. The flow of attributes between federated entities continues to be problematic, disrupting the authentication flow. Trust between service providers and identity providers is still evolving and efforts within the Research and Education Federations Group (REFEDS) [REFEDS] and the AARC Project aim to address the visibility of both the level of assurance of identities and the security capability of federation participants (through Sirtfi [Sirtfi]).

R&D Roadmap

Over the next decade, it is expected that considerable changes will be made to address security in the domains highlighted above. The individual groups, in particular those mentioned above, working in the areas of trust and policies, operational security, authentication and authorisation, and technology evolutions, are driving the R&D activities. The list below summarises the most important actions:

Trust and Policies

By 2020:

- Define and adopt policies in line with new EU Data Protection requirements.
- Develop frameworks to ensure trustworthy interoperability of infrastructures and communities.

By 2022:

- Create and promote community driven incident response policies and procedures.

Operational Security and threat intelligence

By 2020:

- Offer a reference implementation, or at least specific guidance, for a Security Operation Center deployment at HEP sites, enabling them to take action based on threat intelligence shared within the HEP community.

By 2022:

- Participate in the founding of a global Research and Education Forum for incident response, as responding as a global community is the only effective solution against global security threats.

- Build the capabilities to accommodate more participating organisations and streamline communication workflows, within and outside HEP, including maintaining list of security contacts, secure communications channels, and security incident response mechanisms.

- Reinforce the integration of HEP security capabilities with their respective home organisation, to ensure adequate integration of HEP security teams and site security teams.

By 2025:

- Prepare adequately as a community, in order to enable HEP organisations to operate defendable services against more sophisticated threats, stemming both from global cyber criminal gangs targeting HEP resources (finance systems, intellectual property, ransomware), and from state actors targeting the energy and research sectors with advanced malware.

Authentication and Authorisation

By 2020:

- Ensure that ongoing efforts in trust frameworks are sufficient to raise the level of confidence in federated identities to the equivalent of X.509, at which stage they could be a viable alternative to both grid certificates and CERN accounts.

- Participate in setting directions for the future of identity federations, through the FIM4R [FIM4R] community.

By 2022:

- Overhaul the current Authentication and Authorisation infrastructure, including Token Translation, integration with Community IdP-SP Proxies, and Membership Management tools. Enhancements in this area are needed to support a wider range of user identities for WLCG services.

4 Training and Careers

Supporting software developers to acquire skills and to obtain a successful career is considered an essential goal of the HSF, which has the following specific objectives in its mission, namely:

- to provide training opportunities for developers; this should include the organization of software schools for young developers, such as the CERN School of Computing (CSC) and the INFN International School held in Bertinoro, and of a permanent training infrastructure for accomplished developers;
- to provide career support for developers, for instance by listing job opportunities and by helping to shape well defined career paths that should provide advancement opportunities on par with those in, for example, detector construction;
- to increase the visibility and recognise the value for HEP of software developers, for instance by raising the profile of this career and recognising it of equal value to scientific research, as well as by acknowledging and promoting specific “champions” in the field.

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 software 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 physicists and computing experts. Instead of the more traditional situation in which users express their requirements and computer specialists implement solutions, there is a close collaboration between them that is essential for success. This does not come from an organisational problem that needs to be solved; instead it is strongly linked to the nature of the science being done, as well as the scale and cost-of-ownership of HEP data. These challenging needs require solutions that have to evolve continuously based on what has been observed, the experience gained, and new insights from theorists and experimentalists. Many details of the experiment data cannot be known before 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 model and infrastructure. As for detectors, which require engineers and physicists to have a sufficient understanding of each other’s field of expertise, it is necessary not only that physicists understand software and computing challenges but also that computing experts are able to understand enough of the complex physics problems. Only this guarantees solutions that take into account all necessary details and special cases. Fertilising the complementarity of the two

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

Software training must be carried out in the context of highly complex experimental environment and therefore must be done by people who have a sound knowledge of the scientific and technical details. Preparing training material also takes significant time and needs proper recognition.

Because HEP is a challenging field, it has the potential to attract skilled young people who are looking for experience in diverse, demanding contexts. The skills acquired from HEP can also be very relevant for working in other fields, so the use of generic technologies, where possible, improves people's career prospects generally.

For these reasons, 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. At the same time, 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 must be found between these two requirements. It is necessary to find the appropriate incentives to favour 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. However, it is not extensible as it requires a significant dedicated time of all the participants at the same time and location. It is also subject to the funding constraints of the participant institutions. In spite of this, it remains a very valuable component of the training activities, and we should identify opportunities to work with HEP experiments and other training projects to provide accessible core skills training to the community, by basing them at laboratories 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 recent highly successful initiatives such as the LHCb *StarterKit* and ALICE *Juniors*, and collaborate with established generic training initiatives such as *Software Carpentry*. As with hands-on tutorials organised during conferences and workshops, the resulting networking is an important and distinctive feature of these events where people build relationships with other experts.

In recent years, several R&D projects have had training as one of their core activities, two such projects being DIANA-HEP and MVA4NewPhysics. This has proved to be an efficient incentive

to organise 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, that have emerged as interesting approaches, complementary to the schools or topical tutorials, deserve more investment from the community. One core idea is online material shared by a student and a teacher, possibly with computational notebooks that embed runable code and comments into the same document (such as Jupyter notebooks) to provide real examples or practical exercises. Building such material is a time-consuming activity that also requires expert effort. An interesting approach that has started to emerge is the ability of students, or other experts, to enrich the initial material, in particular by adding comments or examples through a collaborative effort. The HSF started to experiment with this approach with WikiToLearn [WikiToLearn], a platform developed in Italy outside HEP, that promotes this kind of training and collaborative elaboration/enrichment of the training materials. Another experiment has been performed by projects like ROOT that have already started to provide some training material based on notebooks.

HEP is not the only community with increased needs for training and there are a lot of initiatives and materials available, in the form of online tutorials, active training and Massive Open Online Courses (MOOCs). It would be a waste of effort for HEP to reinvent the wheel and produce its own materials for topics that are not specific to our scientific field. As an alternative, HEP should spend some effort to evaluate some of the existing courses and build a repository of selected ones, appropriate to HEP needs. This is not a negligible effort and would require some dedicated support to reach the appropriate level. It should help to increase training efficiency by making easier to identify the appropriate courses or initiatives.

A service that emerged in the last years as a very valuable means of sharing expertise are Question and Answer (Q&A) systems, such as Stack Overflow. A few such systems are run by 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 Support and Recognition

Computer specialists in our field are often physicists who specialise in computing. This has always been the case and needs to continue. Nevertheless, for young people, this leads to a career recognition problem, as software and computing activities are not well recognised roles in the various institutions supporting HEP research and recruiting the people working in the field. The exact situation is highly dependent on policies and boundary conditions of the organisation

or country, but recognition of physicists tends to be based generally on participation in data analysis. This is even a bigger problem if the person is spending time to contribute to training efforts. This impacts negatively on the future of these people and reduces the possibility for HEP to engage them in the training effort of the community, when the community needs to involve more people to participate in this training effort. Recognition of training efforts, either by direct participation to training activities or by providing materials, is an important issue to address, complementary to the incentives mentioned above.

There is no easy solution to this problem. Part of the difficulty is that organisations, and in particular the people inside them in charge of the candidate selections for new positions and promotions, need to adapt their expectations to these needs and to the importance of having computing experts with a strong physics background as permanent members of the community. The actual path for improvements in career recognition, as the possible incentives for participating to the training efforts, depends on the local conditions. Nevertheless, we believe that improving the career recognition of physicists who specialise in computing, like others who specialise in detector hardware, is important for the future to ensure the continued successful collaboration between physicists, computer specialists and 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 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. The sheer volumes of data involved make research into appropriate data formats and event content to reduce storage requirements vital. 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

> 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 increased, 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. Support for common software used across the community needs to be recognised and accepted as a common task, borne by labs, institutes, experiments and funding agencies.

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. Even though this evolution will also require projects and experiments to define clear priorities about these common developments, we believe that the HSF, as a community effort, 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, organised 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.

Appendix B : Glossary

AOD: Analysis Object Data is a summary of the reconstructed event and contains sufficient information for common physics analyses.

BSM : Physics beyond the Standard Model (BSM) refers to the theoretical developments needed to explain the deficiencies of the Standard Model (SM), such as the origin of mass, the strong CP problem, neutrino oscillations, matter–antimatter asymmetry, and the nature of dark matter and dark energy.

Coin3D: Coin3D is a C++ object oriented retained mode 3D graphics API used to provide a higher layer of programming for OpenGL.

COOL: LHC Conditions Database Project, a subproject of the POOL persistency framework.

Concurrency Forum : Software engineering is moving towards a paradigm shift in order to accommodate new CPU architectures with many cores, in which concurrency will play a more fundamental role in programming languages and libraries. The forum on concurrent programming models and frameworks aims to share knowledge among interested parties that work together to develop 'demonstrators' and agree on technology so that they can share code and compare results.

CRSG: Computing Resources Scrutiny Group, a WLCG committee in charge of scrutinizing and assessing LHC experiment yearly resource requests to prepare funding agency decisions.

CSIRT: Computer Security Incident Response Team. A CSIRT provides a reliable and trusted single point of contact for reporting computer security incidents and taking the appropriate measures in response to them.

CVMFS: The CERN Virtual Machine File System (CVMFS) is a network file system based on HTTP and optimized to deliver experiment software in a fast, scalable, and reliable way through sophisticated caching strategies.

CWP: The Community White Paper (this document) is the result of an organised effort to describe the community strategy and a roadmap for software and computing R&D in HEP for the 2020s. This activity is organised under the umbrella of the HSF.

Deep Learning (DL): one class of Machine Learning algorithms, based on a high number of neural network layers.

DPHEP: The Data Preservation in HEP project (DPHEP) is a collaboration for data preservation and long term analysis.

EGI: European Grid Initiative. A European organisation in charge of delivering advanced computing services to support scientists, multinational projects and research infrastructures, partially funded by the European Union. It is operating both a grid infrastructure (many WLCG sites in Europe are also EGI sites) and a federated cloud infrastructure. It is also responsible for security incident response for these infrastructures (CSIRT).

FAIR: The Facility for Antiproton and Ion Research (FAIR) is located at GSI Darmstadt. It is an international accelerator facility for research with antiprotons and ions.

GAN: Generative Adversarial Networks are a class of artificial intelligence algorithms used in unsupervised machine learning, implemented by a system of two neural networks contesting with each other in a zero-sum game framework.

GEANT4 : a toolkit for the simulation of the passage of particles through matter.

GeantV: This is an R&D project that aims to fully exploit the parallelism, which is increasingly offered by the new generations of CPUs, in the field of detector simulation.

GPGPU: General-Purpose computing on Graphics Processing Units is the use of a Graphics Processing Unit (GPU), which typically handles computation only for computer graphics, to perform computation in applications traditionally handled by the Central Processing Unit (CPU). Programming for GPGPUs is typically more challenging, but can offer significant gains in arithmetic throughput.

HEPData: The Durham High Energy Physics Database is an open-access repository for scattering data from experimental particle physics.

HL-LHC: The High Luminosity Large Hadron Collider (HL-LHC) is a proposed upgrade to the Large Hadron Collider to be made in 2026. The upgrade aims at increasing the luminosity of the machine by a factor of 10, up to 10^{35} $cm^{-2}s^{-1}$, providing a better chance to see rare processes and improving statistically marginal measurements.

HLT: High Level Trigger. The computing resources, generally a large farm, close to the detector which process the events in real-time and select those who must be stored for further analysis.

HPC: High Performance Computing.

HS06: HEP-wide benchmark for measuring CPU performance based on the SPEC2006 benchmark (https://www.spec.org).

HSF: The HEP Software Foundation facilitates coordination and common efforts in high energy physics (HEP) software and computing internationally.

IML: The Inter-experimental LHC Machine Learning (IML) Working Group is focused on the development of modern state-of-the art machine learning methods, techniques and practices for high-energy physics problems.

IOV: Interval Of Validity, the period of time for which a specific piece of conditions data is valid.

JavaScript: This is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production.

Jupyter Notebook: This is a server-client application that allows editing and running notebook documents via a web browser. Notebooks are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc...). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc..) as well as executable documents which can be run to perform data analysis.

LHC: Large Hadron Collider, the main particle accelerator at CERN.

LHCONE: a set of network circuits, managed worldwide by the National Research and Education Networks, to provide dedicated transfer paths for LHC T1/T2/T3 sites on the standard academic and research physical network infrastructure.

LHCOPN: LHC Optical Private Network. It is the private physical and IP network that connects the Tier0 and the Tier1 sites of the WLCG.

Matplotlib: This is a Python 2D plotting library that produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

ML: Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed. It focuses on prediction making through the use of computers and emcompasses a lot of algorithm classes (boosted decision trees, neural networks…).

MONARC: MONARC is a model of large scale distributed computing based on many regional centers, with a focus on LHC experiments at CERN. As part of the MONARC project,a simulation framework was developed that provides a design and optimisation tool. The MONARC model has been the initial reference for building the WLCG infrastructure and to organize the data transfers around it.

OpenGL: Open Graphics Library (OpenGL) is a cross-language, cross-platform application programming interface(API) for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit(GPU), to achieve hardware-accelerated rendering.

Openlab: CERN openlab is a public-private partnership that accelerates the development of cutting-edge solutions for the worldwide LHC community and wider scientific research.

P5 : The Particle Physics Project Prioritization Panel is a scientific advisory panel tasked with recommending plans for U.S. investment in particle physics research over the next ten years.

PRNG: A PseudoRandom Number Generator is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers.

PyROOT: a Python extension module that allows the user to interact with any ROOT class from the Python interpreter.

QCD - Quantum Chromodynamics, the theory describing the strong interaction between quarks and gluons.

REST: Representational State Transfer web services are a way of providing interoperability between computer systems on the Internet. One of its main features is stateless interactions between clients and servers (every interaction is totally independent of the others), allowing for very efficient caching.

ROOT: a modular scientific software framework widely used in HEP data processing applications.

SAML: Security Assertion Markup Language. It is an open, XML-based, standard for exchanging authentication and authorisation data between parties, in particular, between an identity provider and a service provider.

SDN: Software-defined networking is an umbrella term encompassing several kinds of network technology aimed at making the network as agile and flexible as the virtualized server and storage infrastructure of the modern data center.

SM : The Standard Model is the name given in the 1970s to a theory of fundamental particles and how they interact. It is the currently dominant theory explaining the elementary particles and their dynamics.

SWAN: Service for Web based ANalysis is a platform for interactive data mining in the CERN cloud using the Jupyter notebook interface.

TBB: Intel Threading Building Blocks is a widely used C++ template library for task parallelism. It lets you easily write parallel C++ programs that take full advantage of multicore performance.

TMVA: The Toolkit for Multivariate Data Analysis with ROOT is a standalone project that provides a ROOT-integrated machine learning environment for the processing and parallel evaluation of sophisticated multivariate classification techniques.

VecGeom: This is the vectorized geometry library for particle-detector simulation.

VO: Virtual Organisation. A group of users sharing a common interest (for example, each LHC experiment is a VO), centrally managed, and used in particular as the basis for authorisations in the WLCG infrastructure.

WebGL: The Web Graphics Library is a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins.

WLCG: The Worldwide LHC Computing Grid project is a global collaboration of more than 170 computing centres in 42 countries, linking up national and international grid infrastructures. The mission of the WLCG project is to provide global computing resources to store, distribute and analyse data generated by the Large Hadron Collider (LHC) at CERN.

X.509: a cryptographic standard which defines how to implement service security using electronic certificates, based on the use of a private and public key combination. It is widely used on web servers accessed using the https protocol and is the main authentication mechanism on the WLCG infrastructure.

x86_64: 64-bit version of the x86 instruction set.

XRootD: software framework that is a fully generic suite for fast, low latency and scalable data access.

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).

[AARC] Authentication and Authorisation for Research and Collaboration project, https://aarc-project.eu.

[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).

[ALICE] A Large Ion Collider Experiment at CERN, http://aliceinfo.cern.ch/Public/Welcome.html

[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/.

[BelleII] The B factory experiment at the SuperKEKB accelerator, https://www.belle2.org

[Berghaus2016] @inproceedings{Berghaus2016, title={CERN Services for Long Term Data Preservation},author={Frank Berghaus and Tibor Simko and Jamie Shiers and Gerardo Ganis and Suenje Dallmeier-Tiessen and Germ{\'a}n Cancio Meli{\'a} and Jakob Blomer}, booktitle={iPRES}, year={2016}}

[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.

[CRSG2017] Computing Resources Scrutiny Group Report, CERN-RRB-2017-125, September 2017, https://indico.cern.ch/event/666079/contributions/2721453/attachments/1544297/2423149/CERN-RRB-2017-125.pdf

[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.

[DetectorUpgrade] http://iopscience.iop.org/article/10.1088/1742-6596/515/1/012012/pdf

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

[eduGAIN] https://www.geant.org/Services/Trust_identity_and_security/eduGAIN

[EGI-CSIRT] European Grid Infrastructure Computer Security Incident Response Team, https://csirt.egi.eu/.

[EGISecurityPolicyGroup] https://wiki.egi.eu/wiki/Security_Policy_Group.

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

[FIM4R] Federated Identity Management for Research, https://fim4r.org.

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

[HEP.TrkX] https://heptrkx.github.io/.

[HEPCloud] http://hepcloud.fnal.gov/.

[HERWIG] The HERWIG Event Generator, https://herwig.hepforge.org

[HNSciCloud] The Helix Nebula Science Cloud European Project, http://www.hnscicloud.eu/

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

[HS06] HEPiX Benchmarking Working Group : http://w3.hepix.org/benchmarking.html.

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

[IML] Inter-Experimental LHC Machine Learning Working Group, https://iml.web.cern.ch/.

[Keras] F. Chollet et al., Keras, https://github.com/fchollet/keras.

[LHCb] The Large Hadron Collider Beauty Experiment, http://lhcb-public.web.cern.ch/lhcb-public/.

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

[LHCbStarterkit] LHCb Starterkit, https://lhcb.github.io/starterkit/.

[MADGRAPH] The MadGraph event generator, http://madgraph.physics.illinois.edu

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

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

[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

[PYTHIA] Pythia 8.2 : http://home.thep.lu.se/~torbjorn/Pythia.html

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

[REFEDS]: The Research and Education Federations Group, https://refeds.org

[REN-ISAC] Research & Education Network Information Sharing and Analysis Center, https://www.ren-isac.net[SciGateway] https://sciencegateways.org.

[REP] Reproducible Experiment Platform, http://github.com/yandex/rep.

[scikit-learn] F. Pedregosa et al., Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825-2830.

[Sexton-Kennedy2017] CMS Collaboration presentation to the LHC LHCC, 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).

[Sirtfi] The Security Incident Response Trust Framework for Federated Identity, https://refeds.org/sirtfi.

[SoftwareCarpentry] https://software-carpentry.org/.

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

[SWAN] Danilo Piparo *et al*., SWAN: A service for interactive analysis in the cloud, Future Generation Computer Systems 78 (2018) 1071 - 1078, https://doi.org/10.1016/j.future.2016.11.035.

[TMVA] @Article{TMVA, author = "Hoecker, Andreas and Speckmayer, Peter and Stelzer, Joerg and Therhaag, Jan and von Toerne, Eckhard and Voss, Helge", title = "{TMVA: Toolkit for Multivariate Data Analysis}", journal = "PoS", volume = "ACAT", year = "2007", pages = "040", eprint = "physics/0703039", archivePrefix = "arXiv", SLACcitation = "%%CITATION = PHYSICS/0703039;%%"}

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

[WLCG-AUTH-WG] WLCG Working Group on Authorisation, project-lcg-authz@cern.ch.

[WLCG-SOC-WG] WLCG Working Group on Security Operations Centres, wlcg-soc-wg@cern.ch.

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

[XSEDE] The Extreme Science and Engineering Discovery Environment, https://www.xsede.org[WISE] https://wise-community.org.

[WikiToLearn] Learn with the best. Create books. Share knowledge. https://en.wikitolearn.org/Main_Page.

[WoodACAT2017] https://indico.cern.ch/event/567550/contributions/2686391/

[Zenodo] https://zenodo.org