The Importance of Software and Computing to Particle Physics

A contribution from the High-Energy Physics Software Foundation to the European Particle Physics Strategy Update 2018-2020

ABSTRACT: In 2017 the experimental High-Energy Physics community wrote a *Roadmap for HEP Software and Computing R&D for the 2020s*[1]. This effort was organised by the HEP Software Foundation[2] (HSF) and was supported by more than 300 physicists from more than 100 institutes worldwide. It delivered a strategy outlining the most important areas in which investment is needed to ensure the success of our experimental programme. This contribution to the ESPP is an executive summary of the most critical and relevant points raised in that white paper.

Contact: Graeme A Stewart, *graeme.andrew.stewart@cern.ch*

[1] *arXiv:1712.06982*
[2] *https://hepsoftwarefoundation.org/*

The Importance of Software to the HEP Experimental Programme

The current success of the experimental High-Energy Physics programme relies hugely on software and computing. We have developed many innovative and unique pieces of software as a necessity, as no off-the-shelf solutions were available. The HEP software landscape consists of tens of millions of lines of code, from widely used community supported packages, like ROOT and Geant4, to experiment and detector specific codes. Thousands of physicists spend time writing this code and the lifetimes of HEP experiments require maintaining and evolving this software over decades. These codes are run at a massive scale. For the LHC experiments, in particular, about 750k CPU cores are constantly employed on the Worldwide LHC Grid running event generation, simulation, reconstruction and analysis jobs that are critical to our physics output. The cost of this part of our programme is very high - to purchase just this CPU resource in a commercial cloud would cost hundreds of millions of Euros every year. It is incumbent on us to use these resources as efficiently as possible for physics.

Future Challenges in an Evolving Technology Landscape

A programme of new and upgraded accelerators and experiments will result in far higher data rates, for both trigger and offline. We need to adapt our data processing code and data handling infrastructure to cope with these rates, but we do so in a landscape where the core technologies and the scientific computing infrastructure will not be the same as before and violate many of the assumptions made in the past.

HEP benefited for many years from the rapid evolution of microprocessor technology and a homogeneous data centre model (x86 and Linux). While shrinking feature sizes still lead to increases in transistor density (Moore's Law), improvements in clock speeds have stopped and only internal processor parallelism is improving. This fundamental shift means that software now has to exploit more and more parallelism to reduce processing time. Controlling memory usage and bandwidth becomes critical to maintaining throughput. This paradigm shift from serial to parallel execution is far complete for HEP and requires more software developers with both specific technical expertise and physics domain knowledge to work together.

Modern complex CPU architectures are now no longer seen as optimal for many workloads and the performance gains on different platforms, such as GPUs or FPGAs, have been much greater than on CPUs. Adapting our large code bases to this new hardware landscape often requires a fundamental redesign of algorithms from traditional serial implementations. The current set of C++ abstractions used for two decades in both the algorithmic and framework codes needs to be substantially rethought.

Currently HEP handles data across many geographically dispersed, but rather homogeneous, computing centres. Our grid infrastructure is now evolving to incorporate other resources, such as HPC centres with GPUs. Other resources may be ephemeral and our new data handling strategies must be able to accommodate these resources easily. Production systems must also adapt to new workflows, e.g. related to large scale machine learning

training. Much of this new infrastructure will be shared with other science communities as well, so assumptions of dedicated network and storage resources only for HEP must be relaxed.

Software for Future Detectors

Efficient working software is required years in advance in order to properly optimise detector design for efficient reconstruction. This is true across all the frontiers of particle physics (energy, intensity and cosmic). One example is future hadron colliders, which present a huge challenge. Such accelerators would produce collisions at a pile-up of up to 1000, requiring new detector technologies and significant advances in reconstruction software to extract signals in a reasonable time. Efficient reconstruction and simulation is needed and must be extremely accurate to support a precision physics search programme. We must anticipate analyses which deal with orders of magnitude more events, requiring more efficient approaches to data storage and processing, while not burdening analysts with complex interfaces. We note that such challenges are faced now in many sectors of 'big data' society and HEP can both learn from and contribute to these domains.

Necessary Investments in People

To face the computing challenges of the future HEP programme substantial effort is required to improve our software and computing. Ongoing investment in R&D, with increased funding, will be critical to extract as much physics as possible from the data delivered by the upgraded LHC and other detectors.

Areas for investment must focus on developing and evolving our software so that it can adapt to the changing and uncertain hardware landscape, that it can be sustained for decades, and that it can incorporate advances from other fields, such as machine learning and data science. This will require cooperation with software engineers and computer scientists who are not traditionally from our field. The necessary skills we need require proper, well-funded training for students and postdocs; this training will make the contribution from HEP even more valuable when people move outside our field. We have long lead times and lifetimes for experiments, meaning we need long lifetimes for software and long-term support for those who develop it. It is thus crucial that we offer well defined career paths, including new roles such as Research Software Engineers, for those who chose to work in these software areas, so as to retain the necessary talent for our discipline. This is especially true for those who chose to focus on the software engineering of scientific code, where long-term career prospects must be viable. This requires endorsing new initiatives to attract funding for a sustainable and cooperative scientific software ecosystem at European universities and labs.