The openlab Platform Competence Center thoughts on a future HEP Software Collaboration – DRAFT v1

Andrzej Nowak, CERN openlab CTO office
June 13th 2014, Geneva

Introduction

This document introduces a somewhat fuzzy view on what the future HEP Software Foundation might look like, and the challenges it might face. The content is derived both from the experience of openlab, which is based much more in the industry and the art of computing than in HEP, as well as from the ideas presented by the various contributors of the April meeting. Therefore, there is a bias towards more generic issues, and possibly the view could be considered too superficial for experienced HEP contributors. Nevertheless, the hope is that some ideas might be considered useful.

Generic challenges ahead of legacy software

Any legacy software today, the HEP variant being no exception, faces a number of crucial challenges. A large code base is difficult to manage and run efficiently on newer hardware. Programs developed with single thread performance in mind, adhering to decades old programming models, do not scale sufficiently on newer systems. Existing benchmarks don't even cover the majority of the performance a computing platform has to offer, and often no longer represent the workloads seen "in the wild". Further, it is now clear that those who wish to make use of accelerators must be willing to completely rethink the implementation and possibly the algorithm of the solution to their scientific problem.

At the same time, with respect to many of these points, and for various objective and subjective reasons, HEP software fell into neglect. Already in the late 90's it was apparent that multi-core processors would be the standard, and by the mid 2000's clear future leaders emerged: many x86-64 cores with vector units. It took more than a decade to take these facts in and produce the first – promising, one should add – prototypes of next-generation HEP programs.

For HEP, the scale of the shift required at this point of time could be compared to that of the move to PCs. Not only that, the multi-core and wide-vector revolution is only a part of the equation, and one that is becoming an out-of-fashion topic. Just around the corner are next generations of accelerators, low power computing cores, disposable systems and most importantly – heterogeneous architectures.

With respect to the last point, the era of fairly homogeneous computing with x86-64 on multi-core will not carry on with the same force as it has until recently. While no one today can say for sure if technologies such as those from ARM or NVIDIA will take over (it would seem unlikely, at least in the short term), the current trends are only the symptoms of a larger shift – that of accelerated change and innovation, to which any software must adapt, and with which any software will have to scale.

Already some time ago, this constraint ceased to become an optional one, and is becoming a real, financially binding threat.

HEP-specific challenges

One of recurring challenges for HEP computing is its employment of the uniquely HEP perspective. Unfortunately, despite many efforts such as our participation in standard committees (e.g. C++), the days where HEP proudly led computing in general, or its large parts, have diminished. It is only a fact of scale that the world rushed on, and it might be necessary to humbly accept this fact. This conclusion leads to a number of secondary observations:

- HEP will benefit from **learning how to take from others**. Software has become an enormous industry, with a supply of professionals that one could only dream of 20 years ago. The said professionals churn out incredible amounts of solutions to problems that are not just our own any longer – reusing and integrating such solutions can bring savings.

- HEP will benefit from **learning from others**. Computing professionals, in particular those in the industry, represent a vast reservoir of knowledge that is out there and can be tapped into at virtually any time. CERN openlab is an example of a format in which such exchanges might take place, but is not the only one. A good link to computing experts not only delivers top-notch expertise where it is needed, but can also deliver expertise *on demand*.

- HEP will benefit from daily **collaborations with computing specialists and software developers**, educated primarily in technical domains. In particular, software engineering related best practices, technical solutions and expertise will help build up a credible and scalable model for software development.

- HEP will benefit from **retaining expertise** which lies between physics and computing. It takes a long time to build up and combines our unique domain with the one of a democratic appeal and reach.

In summary, HEP’s interaction with the external world should intensify, and should cover both “push” and “pull” directions. Components developed for scalability and reuse, satisfying open standards, considerably increase their chance of success and obtaining contributions, whether inside or outside of HEP. A good impact on external ecosystems also increases the chance of favourable decisions from funding agencies.

HEP has enormous expertise and tradition in collaboration. The current challenge is just to take it one level further, outside of the current ecosystem. In face of the shifting computing technology, HEP too can learn to adapt to the accelerating pace of innovation.

The tasks of the HEP Software Foundation

As many have pointed out, there is a wide range of roles the Foundation could play. The following are some loose suggestions which could be thrown into the mix at will:

- **Lay out a framework for innovation** – one of the strengths of HEP development is that in many projects innovations come unhindered in the bottom-up direction. However, a weakness is that there rarely is a systematic process for innovation in place (nb: which could also imply the need for a systematic process for production). One could imagine a scenario where either existing models for innovation are adopted, or there is experimentation with such models. In terms of existing models there is an exciting range of choice, starting with the Xerox legacy and ending with DARPA's ambitious breakthrough projects executed on a relatively short timescale.

- In face of **joint objectives of the stakeholders**: unless the Foundation increases cohesion in the community and introduces coordination, thus reducing the replication of effort, it will face the prospect of having less success and less added value. Sharing expertise and tools – or even knowing where they are – would then be an important part of the activities. Shared technical resources could mean hardware, software or experts, while shared work requires the identification of shared needs.

- For the Foundation to have **high impact**, governors might need to have influence over the members. For instance, standardized technical management could be considered one of the points under this condition. Overall, governors might have to arbitrate between physics goals and the ability to deliver – definitely not an easy task in itself, but one that can only be accomplished with a 10'000 ft view in hand.

- More philosophically, the governors will need to decide on the **proportion of promoting, enabling and limiting roles** in their activities. In less sophisticated language, the Foundation would have carrots at its disposal, but probably also some sticks. On one hand, industry experience demonstrates that projects should compete for the right to be promoted. Such an approach might take what might seem as an extreme angle: many companies drive two independent teams with slightly different points of view and skillsets to the same goal. Such a tactic does not necessarily need to be the norm (that could be wasteful), but on topics of high innovation value it might not be a bad option. At the same time, a delicate balance of efficient resource use will have to be kept – it will be up to the Foundation to maintain a conscious record of the global costs and benefits of an activity, as well as of the overlaps between projects.