

The end of HEP-specific computing as we know it?

O Smirnova^{1,2}, R Brun³, R Cailliau³, F Carminati³, P Elmer⁴

¹ Lund University, Particle Physics, Department of Physics, Box 118, SE 22100 Lund, Sweden

² NeIC, NordForsk, Stensberggata 25, 0170 Oslo, Norway

³ CERN PH, 1211 Geneva, Switzerland

⁴ Princeton University, Princeton, NJ, USA

E-mail: oxana.smirnova@hep.lu.se

Abstract. CHEP Conferences are dedicated to a quite specific scientific computing domain, and deal with rather specialised software, developed for the needs of the High Energy Physics community. The sheer size of this community created an environment which until recently has been to a large extent isolated from the mainstream computing. There is however an emerging trend for the computing solutions to spill outside the traditional laboratory boundaries, benefiting from becoming less domain-specific. This paper summarises the panel discussion held at the CHEP'13 conference with the goal to answer the questions, why the mainstream software approaches are not always suitable for the High Energy Physics community, and why its own solutions so far enjoyed little popularity in other domains?

1. Introduction

The discussion panel “The end of HEP-specific computing as we know it?” took place during the concluding part of the CHEP'13 conference, and was prompted by the observation that the HEP computing community is now less isolated, and starts exploring possibilities of using mainstream approaches. This process, however, is rather slow, as the HEP community neither appears to be ready to accept generic solutions, nor does it seem to be ready to make own solutions more generic. To discuss whether these observations are true, and if so, what are the reasons, a panel of experts was invited, consisting of: René Brun, a key developer of ROOT [1, 2]; Robert Cailliau, a key developer of the World Wide Web; Federico Carminati, a key developer of AliEn [3, 4]; and Peter Elmer, a co-convenor of the “Software Engineering, Parallelism & Multi-Core” track at CHEP'13. The panel was moderated by Oxana Smirnova, a co-developer of ARC [5]. This article briefly summarises the views expressed during the panel discussions.

2. High-level overview

Accelerator-based physics experiments have very specific ways of handling data, due to a number of reasons. One distinct feature of such experiments is that they produce data in copious amounts, being essentially data generators. This is different from many other sciences, where data can not be produced on-demand, and definitely not in such large volumes. In addition, each experimental installation is a unique device, and this uniqueness usually mandates recording data in its own specific manner. Combination of these factors makes it very difficult to use solutions developed for other kinds of data. One could still imagine that different experiments can make use of same solutions, but even this proves to be a challenge. Despite similar scientific objectives,



there are large differences among data sets collected by the accelerator experiments. This in turn prompts development of quite diverse computing tools and methods, as can be seen from the contributions submitted to CHEP Conferences. Even though core technologies change over the years, the bulk of challenges addressed by CHEP remains largely unchanged, as every new experiment tends to re-invent computing solutions, with inevitable customisations.

There are, however, certain similarities and common tools. For example, most experiments nowadays rely on the Scientific Linux operating system [6], use ROOT as the data persistency framework, and GEANT4 [7, 8] for detector simulation. Such solutions were developed by the community and for the community, and as such were not initially intended to be used for other applications. Neither were they always created using industry-standard software development approaches: the development was driven by the people motivated to achieve the best result, and freedom of creativity was considered by the community to be more important than the formal procedures. Creative people often build own tools, not just in HEP: most Open Source software started as a grassroots development.

Even though the HEP community develops or customises almost everything, from protocols to operating systems to end-user tools, there is no way to avoid using commercial or mainstream Open Source software. Scientific Linux is based on Red Hat, many solutions rely on Oracle databases, encryption makes use of OpenSSL, and so on. However, experience shows that straightforward usage of such products in HEP leads to disasters, unless customisation is applied. Such customisations are usually rooted in the specifics of the HEP data handling, and may not be beneficial for other scientific communities, which rely on more traditional data formats.

3. Open Source benefits and challenges

HEP community was among the first to embrace the Open Source software concept. The large size of the community allows to exploit all the strengths of this approach, guaranteeing a large human resource full of ideas, and benefiting from the existing decision-making mechanisms. However, even Open Source projects rely on certain procedures and formalities, which are necessary to maintain coherent software repositories. HEP community is still rather slow in catching up with such aspects.

Choice of an appropriate Open Source licence proves to be a challenge in the HEP environment, where intellectual property is customary shared with thousands of experiment members, and where the concepts of code re-usage or software branching are poorly understood.

The HepForge [9] initiative offers a very convenient Open Source development environment, hosting a large number of HEP software projects. Still, it lacks a repository of binary packages, and many notable softwares are not accessible via HepForge. In general, although Scientific Linux is the reference operating system, especially at CERN, not all HEP software packages are available as *rpm*. One of the reasons is often the unmanageable complexity of dependencies. The tools that are packaged as *rpm* are typically available only from the Scientific Linux repositories, thus the users of other operating systems, especially not Red Hat based ones, have difficulties installing HEP software tools and all the dependencies.

User support is one of the differences between Open Source and commercial softwares. Like most Open Source solutions, those developed for HEP rely on community support via mailing lists and other fora. While this helps solving many trivial issues, hands-on involvement of experts is usually needed when the software is attempted to be used in a new environment. General lack of comprehensive documentation and shortage of experts makes it rather difficult to promote usage of HEP solutions outside the domain. One may argue that a larger user community should bring more contributors, but so far HEP software developers were rather wary of unsolicited input, and had little or no mechanisms of reviewing external contributions.

4. Reaching outside

Despite the overarching drive to create customised software, HEP community has famously produced the World Wide Web. This lead to probably exaggerated expectations that it is capable of delivering other solutions that may change the world the way the Web did. In reality, WWW appearance was an exception rather than the rule. Though the creative environment at CERN is a fertile ground for such ideas, the HEP community itself did not perceive this technology as something particularly useful. Ironically, *HTTP* is still not considered as a usable data and information transfer protocol for HEP computing (protocols like *gridftp*, *xroot* and *LDAP* are preferred), neither are Web service based solutions popular.

Still, developers of HEP software are reasonably proud when their products are used outside the community. In these Proceedings one can find examples of non-HEP applicability of GEANT4, DIRAC [10, 11], PanDA [12, 13] and other tools that were originally developed for HEP. The impact of these can hardly be compared to that of the WWW, but it certainly helps in attracting public funding and hence further improving the software.

One can point out that some of these products, like e.g. DIRAC and PanDA, solve very similar tasks, and HEP solutions may have had a larger impact if they had more commonalities. When CERN started investing into Grid technologies, a substantial effort was dedicated to identify commonalities and develop common approaches. Such common solutions are conceptually possible, as one can find from several CHEP contributions. However, they often turn into a sort of a forced marriage, where neither party is fully satisfied with the result. It is often said, jokingly, that there are five Grids at CERN: one per experiment, and one common. The joke has a grain of truth: diversity of Grid solutions developed at CERN may have contributed to the slow take-up of the technology elsewhere.

5. Conclusion and outlook

It is clear that HEP software can benefit from external contributions, and non-HEP communities may find HEP software useful. In fact, the two aspects are strongly linked: once the software starts to be used for other applications, modifications and extensions are inevitable, and this is best done in co-operation with all the involved communities.

HEP software development communities are traditionally conservative, and would rather see applications tailored to fit the software architecture, rather than making the software more modular and flexible. It however may be beneficial in future to study other non-HEP use cases through close communication with other communities, such as e.g. the astroparticle physics, which already makes use of some HEP products. If HEP software will become more open to external contributions, and better documented, it has a good chance to benefit from extra qualified manpower and possibly from more mainstream development, maintenance and distribution approaches.

References

- [1] ROOT – An Object Oriented Data Analysis Framework Web site URL <http://root.cern.ch>
- [2] Antcheva I, Ballintijn M, Bellenot B *et al.* 2009 *Computer Physics Communications* **180** 2499–2512
- [3] AliEn – ALICE Environment Web site URL <http://alien2.cern.ch>
- [4] Bagnasco S, Betev L, Buncic P *et al.* 2008 *Journal of Physics: Conference Series* **119**
- [5] Ellert M, Grønager M, Konstantinov A *et al.* 2007 *Future Gener. Comput. Syst.* **23** 219–240 ISSN 0167-739X
- [6] Scientific Linux Web site URL <http://www.scientificlinux.org/>
- [7] GEANT4 simulation toolkit Web site URL <http://cern.ch/geant4>
- [8] Agostinelli S, Allison J, Amako K *et al.* 2003 *Nuclear Instruments and Methods A* **506** 250–303
- [9] HepForge Web site URL <http://www.hepforge.org>
- [10] DIRAC – Distributed Infrastructure with Remote Agent Control Web site URL <http://diracgrid.org/>
- [11] Tsaregorodtsev A, Bargiotti M, Brook N *et al.* 2008 *Journal of Physics: Conference Series* **119**
- [12] Big PanDA project Web site URL <http://atlascloud.org:8080/pandawms>
- [13] Maeno T 2008 *Journal of Physics: Conference Series* **119**