Key4hep

Letter of Interest

Authors:

Placido Fernandez Declara[b], Frank Gaede[a], Gerardo Ganis[b], Paolo Giacomelli[f], Benedikt Hegner[b], Clement Helsens[b], Thomas Madlener[a], Andre Sailer[b], Graeme A Stewart[b], Valentin Volkl[b], Joseph Wang[c,1], Wenxin Fang[d], Weidong Li[d], Tao Lin[d], Jiaheng Zou[d], Xingtao Huang[e], Alexey Zhemchugov[g]

[a]DESY; [b]CERN; [c]Twofish Enterprises (Asia) Limited; [d]IHEP; [e]Shandong University; [f]INFN Bologna; [g]JINR Dubna

Introduction

Future HEP experiments require detailed simulation and advanced reconstruction algorithms to explore and maximise the physics reach of their proposed machines and to design, optimise, and study detector geometry and performance. To synergize such developments the CEPC [1], CLIC [2], FCC [3], ILC [4], and SCT [5], communities have started the creation of a "Turnkey Software Stack" (Key4hep), which would provide all the necessary ingredients, from simulation to analysis, for future experiments. This approach is based on the positive experience of the linear collider projects ILC and CLIC, that have developed and used a common software stack iLCSoft [6] over the last decade. This would cover most if not all future linear and circular machines colliding electrons, muons and hadrons. The software stack will facilitate writing specific components for experiments ensuring coherency and maximising the re-use of established packages to benefit from existing solutions and community developments, for example, ROOT, Geant4, DD4hep, Gaudi and podio.

The interplay between reconstruction algorithms and detector geometry, for example in particle flow clustering, means that the detector hardware cannot be developed and designed independently of the software. At the same time, developing and validating sophisticated algorithms, including accounting for a large number of edge cases, requires a significant amount of resources.

Key4hep project

The turnkey software stack should encompass all the libraries needed for simulation, reconstruction, and analysis. The base of the common stack is formed by standard libraries, for example Boost, Python, CMake, compilers, and the operating system. These products are typically developed outside of HEP. Building on top of these libraries are the HEP libraries that provide generic functionality – ROOT [7, 8], Geant4 [9–11], CLHEP [12]. Combining and extending these libraries are tools that address more specific needs but are still used by multiple experiments, for example detector geometry solutions like DD4hep [13–15], pattern recognition for particle flow clustering or neutrino experiments like PandoraPFA [16], or Monte Carlo event generators, like Pythia [17]. The main ingredient is the core *framework* (Gaudi [18]) providing the orchestration layer which controls everything else. Such frameworks usually require an event data model (EDM) for transient and persistent data, interfaces to databases, and many algorithms and tools that implement the simulation

[1] This author will be funded in part by a postdoctoral research grant from the Innovation Technology Fund of the Hong Kong Special Administrative Region

and reconstruction logic, or wrappers to other generic packages that provide the desired functionality, e.g., PandoraPFA, ACTS [19], or FastJet [20]. In addition the turnkey stack will use off-the-shelf packing software, such as Spack [21], and virtualization technology, such as docker [22] and podman [23]. This maximises the benefits of technical solutions that can be shared and will allow the software to be quickly and easily installed in different environments. In addition the common software is distributed through cvmfs[2], with nightly builds and releases. To develop the turnkey stack we created a working group to coordinate development, and our work has been made available on a public software repository [24].

The turnkey software stack, Key4hep, aims to create a complete data processing framework for the benefit of future collider experiments. Where it is found necessary or beneficial, new solutions are adopted, for example a new event data model EDM4hep [25] based on podio, or the Spack packaging tool. This approach does not require one to completely abandon existing solutions. The development and application of the prototype Marlin [26] processor wrapper for the CLIC and ILC reconstruction shows that the most valuable parts, the reconstruction algorithms, can be ported to the new framework with minimal effort. The existing processors can evolve into the Gaudi framework in parallel to continuous validation. Further developments of the event data model and adaptations to the new framework are currently in progress. The first milestone is evolving the processor wrapper beyond its prototype state and to validate the results against the existing CLIC software. Then the individual processors will be adapted to Gaudi and made available for other users of the Key4hep stack. The software for the FCC experiments will also be adapted to Key4hep, but here only the event data model has to be adapted to EDM4hep [25]. The new CEPC software prototype CEPCSW [27] is fully integrated with Key4hep including K4FWCore and EDM4hep. It can be used for validation purpose and provide quick feedback to developers when a new Key4hep version is released.

Invitation to collaborate

The developments for the Key4hep software stack are not closed, and contributions and use by other experiments are welcome. Our software development process uses open source development methodologies and we welcome users and contributors to visit our software repository [24] and join the Key4hep mailing list [3].

One of the goals of the turnkey software stack is to reduce the need to develop core software tools for every experiment individually. To accomplish this we will need a large community of users, testers, and documentation writers to provide feedback on our efforts. Furthermore, we hope that a standard set of software libraries will make it easier to expand software functionality by incorporating new computational technology and pull in new technologies from the general open-source software ecosystem. In addition, a standard set of software tools will allow the HEP community to concentrate resources in areas such as training, documentation, and user support which will facilitate scientific research. We are also working to ensure that Key4hep works on a large set of common linux distributions such as CentOS, Ubuntu, Fedora and Mageia or macOS and to ensure compatibility with the latest tool chains, and are looking for other such collaborations with the open source community to maintain and distribute HEP related software.

[2] See https://key4hep.github.io/key4hep-doc/

[3] See https://e-groups.cern.ch/e-groups/EgroupsSubscription.do?egroupName=key4hep-sw

References

[1] CEPC Study Group Collaboration, M. Dong et al., *CEPC Conceptual Design Report: Volume 2 - Physics & Detector*, arXiv:1811.10545 [hep-ex]. 1

[2] CLICdp Collaboration, CLIC Collaboration, *The Compact Linear Collider (CLIC) - 2018 Summary Report*, https://doi.org/10.23731/CYRM-2018-002. 1

[3] M. Mangano et al., *FCC Physics Opportunities: Future Circular Collider Conceptual Design Report Volume 1. Future Circular Collider*, Dec, 2018. https://doi.org/10.1140/epjc/s10052-019-6904-3. 1

[4] P. Bambade et al., *The International Linear Collider: A Global Project*, 2019. arXiv:1903.01629 [hep-ex]. 1

[5] *Super Charm–Tau Factory*, https://ctd.inp.nsk.su/c-tau/. 1

[6] *iLCSoft project repository*, . https://github.com/iLCSoft. 1

[7] R. Brun and F. Rademakers, *ROOT – An object oriented data analysis framework*, Nucl. Instrum. Meth. **A389** (Apr., 1997) 81–86. http://www.sciencedirect.com/science/article/B6TJM-3SPKX96-1F/2/3aa2b2cb72c9a4316a842802541bf317. 1

[8] F. Rademakers et al., *root*, June, 2018. https://doi.org/10.5281/zenodo.848818. 1

[9] J. Allison et al., *Geant4 developments and applications*, IEEE T. Nucl. Sci. **53** (Feb., 2006) 270–278. 1

[10] S. Agostinelli et al., *GEANT4 - A Simulation Toolkit*, Nucl. Instrum. Meth. **A506** (2003) 250–303. 1

[11] J. Allison et al., *Recent developments in Geant4*, Nucl. Instrum. Meth. **A835** (2016) 186–225. http://www.sciencedirect.com/science/article/pii/S0168900216306957. 1

[12] https://gitlab.cern.ch/CLHEP/CLHEP. 1

[13] M. Frank, F. Gaede, M. Petric, and A. Sailer, *DD4hep*, July, 2018. https://doi.org/10.5281/zenodo.592244. 1

[14] M. Frank, F. Gaede, C. Grefe, and P. Mato, *DD4hep: A Detector Description Toolkit for High Energy Physics Experiments*, J. Phys. Conf. Ser. **513** (Oct, 2013) 022010. 1

[15] A. Sailer, M. Frank, F. Gaede, D. Hynds, S. Lu, N. Nikiforou, M. Petric, R. Simoniello, and G. G. Voutsinas, *DD4Hep based event reconstruction*, J. Phys. Conf. Ser. **898** (Feb, 2017) 042017. https://cds.cern.ch/record/2244623. 1

[16] J. Marshall and M. Thomson, *The Pandora Software Development Kit for Pattern Recognition*, Eur. Phys. J. **C75** (2015) no. 9, 439. 1

[17] T. Sjostrand, S. Mrenna, and P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, JHEP **05** (2006) 026. 1

[18] LHCb Collaboration and ATLAS Collaboration, *Gaudi v33r0*, Dec., 2019. https://doi.org/10.5281/zenodo.3660964. 1

[19] A. Salzburger, B. Schlag, C. Gumpert, F. Klimpel, H. Grasland, J. Hrdinka, M. Kiehn, N. Calace, P. Gessinger, R. Langenberg, and X. Ai, *Acts Project: v0.15.00*, Jan., 2020. https://doi.org/10.5281/zenodo.3626878. 2

[20] M. Cacciari, G. P. Salam, and G. Soyez, *FastJet User Manual*, Eur. Phys. J. **C72** (2012) 1896. 2

[21] B. Morgan, G. A. Stewart, J. C. Villanueva, and H. A. Willett, *Modern Software Stack Building for HEP*, Nov., 2019. https://doi.org/10.5281/zenodo.3598985. 2

[22] D. Merkel, *Docker: Lightweight Linux Containers for Consistent Development and Deployment*, Linux J. **2014** (Mar., 2014) . 2

[23] H. Gantikow et al., *Rootless Containers with Podman for HPC*, . https://vhpc.org/static/PapersPresentations2020/iscworkshops2020_paper_12.pdf. 2

[24] Key4hep Team, *Key4hep repository*, https://github.com/key4hep, 2020. GitHub repository (2020) . 2

[25] C. Helsens, V. Volkl, C. Neubuser, G. Ganis, and J. C. Villanueva, *A software framework for FCC studies: status and plans*, Nov., 2019. https://doi.org/10.5281/zenodo.3599139. 2

[26] F. Gaede, *Marlin and LCCD—Software tools for the ILC*, Nucl. Instrum. Meth. **A559** (2006) no. 1, 177–80. 2

[27] CEPCSW Team, *CEPCSW prototype repository*, https://github.com/cepc/CEPCSW, 2020. GitHub repository (2020) . 2