PROCEEDINGS OF SCIENCE

Software and computing challenges for a Muon Collider Detector

Paolo Andreetto,[b,*] **Nazar Bartosik,**[c] **Massimo Casarsa,**[d] **Alessio Gianelle,**[b] **Karol Krizka,**[a] **Donatella Lucchesi,**[b,e] **Simone Pagan Griso**[a] **and Lorenzo Sestini**[b]

[a] *Lawrence Berkeley National Laboratory, Berkeley, California, USA*
[b] *INFN Sezione di Padova, Padova, Italy*
[c] *INFN Sezione di Torino, Torino, Italy*
[d] *INFN Sezione di Trieste, Trieste, Italy*
[e] *University of Padova, Padova, Italy*

E-mail: paolo.andreetto@pd.infn.it

Studies of physics and detector performance of a possible experiment at a Muon Collider are attracting a lot of interest in the High Energy Physics community. Projections show that high precision measurements are possible as well as very competitive searches for new physics. However, the presence of the intensive beam-induced background, generated by the muon beams decay, poses new computing and software challenges ranging from event simulation to reconstruction algorithms. Moreover, an increasing number of collaborators around the world demands as well an easy to maintain and flexible infrastructure distributed across several countries. This contribution will present the strategies adopted so far to cope with all the difficulties arising from such a complex working environment.

41st International Conference on High Energy physics - ICHEP2022
6-13 July, 2022
Bologna, Italy

*Speaker

PoS(ICHEP2022)226

1. Introduction

The Muon Collider project is an international collaboration of several research centers (BerkeleyLab, CERN, Fermilab and INFN) aiming to design a detector suitable for the next generation of high-energy colliders. One of the key aspects of the collaboration is the creation of the software support required for all the activities of the project. In the following, a complete overview of that kind of support is outlined, starting from the core part of any development, the framework, and moving on, in the next step, to the description of all the resources available for the project. Since many tasks related to the software and resource management are still work in progress, a thorough list of ongoing activities is proposed in the last part.

2. The framework

The Muon Collider framework is an evolution of the ILC Software [1] suite. The ILC Software distribution is a collection of tools and libraries for the simulation and reconstruction of events for the CLIC project. DD4Hep [2] is the toolkit of ILC Software dedicated to the detector simulation; it is built on top of GEANT4 and its modular architecture has been designed in order to simplify the definition of a detector geometry. Reconstruction tasks are handled by Marlin [3]; it implements a processing workflow of plug-ins, or processors, working on datasets whose format is compatible with LCIO specification [4]. A rich set of processors is contained in the software distribution, covering many aspects of the activity, starting from the detector digitization up to higher-level reconstruction algorithms. Several processors integrate third-party libraries, like ACTS [5], Pandora Particle Flow Analysis and FastJet.

The Muon Collider framework enhances the functionalities of the original software suite with extensions specific for the Muon Collider detector. For what concerns the simulation stage, the Muon Collider framework provides:

- the geometry description, in XML format, of the Muon Collider detector [6];
- an extension for DD4Hep that deals with a fine-grained description of the sensors which compose the modules of the vertex detector.

For the reconstruction stage, the framework increases the original set of Marlin processors with:

- a processor that leverages the algorithms contained in the ACTS library to improve the performances of the tracker reconstruction;
- a processor that implements a real time-dependent digitization for silicon sensors.

The framework also enhances the functionalities of original processors with:

- a new mechanism for an optimized overlay of large datasets of beam-induced background (BIB) samples;
- new strategies for dealing with BIB samples in the conformal tracking algorithm, like regional and cone tracking and several new filtering techniques for the tracking system, like double-layer, cone and time filters;

- an optimized general digitization process, that introduces the concept of "time window";
- the support for a new type of calorimeter (Crilin [7]) and new strategies for dealing with BIB in the calorimeter cells, like BIB subtraction.

In order to guarantee the appropriate level of support to developers and final users the choice of one or more reference platforms is crucial. For the Muon Collider framework the reference platform is CentOS 8 Stream. The selected option must be considered a temporary solution since the stream brand of the dismissed CentOS 8 distribution does not consider the publication of any release; it's just a continuous flow of package updates. Despite this, the selected solution is a good compromise: it guarantees enough stability for a good level of support and the availability of updated tools and libraries. For a final solution it is necessary to adapt to the guidelines from the major research organizations like CERN and FermiLab.

The software management of the Muon Collider framework makes use of the development tools supplied by the ILC software distribution. The tools guarantee a high level of portability across different architectures since they implement a sand-box for the building process with few dependencies from the system. The source code for all the components specific for the Muon Collider detector is publicly distributed by the github service [8].

An essential step of the software management process is the selection of the technologies for the final artifacts. For the Muon Collider software the container-ization with Docker [9] has been considered the most versatile solution. A container encapsulates executables, libraries and any other resources, with no dependency from the host system that operates the container itself. Any Docker container is compatible with many high-level orchestration technologies, like docker-compose or Kubernetes. The Docker container [10] produced by the building process is a complete CentOS 8 Stream installation with ILC Software and Muon Collider Software loaded inside and allows a user to:

- run the simulation and reconstruction tools directly from the host system; with this option the container can be considered as a package of executables;
- access the container and perform any action on the CentOS 8 Stream installation; with this option the container can be seen as a virtual machine.

In both cases the data required as input of the simulation or reconstruction and the data produced as output can be shared between the host system and the container through the functionalities of the Docker storage system. The container-ization of the artifacts, with the structure describe above, shows at least a major drawback: it's not suitable for the realization of new components for the framework. Even if it is possible, in principle, to install and setup a complete environment for code development, it ends up to be a cumbersome solution in many cases.

The Docker container for the Muon Collider framework is also the input element for a continuous integration process setup in the github service with workflows. A workflow is able to retrieve the version of the container to certify and to apply a suite of functional tests, for both the simulation and the reconstruction stage. The released Docker container is published by the DockerHub service, with

PoS(ICHEP2022)226

a mirror at CERN. The container is also converted in a format compatible with Apptainer/Singularity [12], so that it is possible to deliver through different channels and simplify the adoption of the artifact in many research contexts. The Singularity container is published by INFN-CNAF storage system (see later) and by the LBNL in a public repository. One of the preferred channel distribution for the HEP community is the CVMFS, a geographically distributed file system. At the current date the Muon Collider Team is working on the publication of the Singularity container by the CERN CVMFS.

3. Resources

The Muon Collider community can rely on a wide set of computational resources, spread across many sites in different countries. As the time goes by, the number of members of the community increases and each new member contributes with new computational efforts. Since the project does not impose any constraint on the contributions, the Muon Collider group must face the challenge of dealing with different technologies, like grid or cloud computing, HPC solutions, and integrate them in a way that the final user does not feel the burden of accessing different technologies.

The first infrastructure available is CloudVeneto [13]. CloudVeneto is a cloud infrastructure operated by the INFN site in Padova and the University of Padova. It's built on top of Openstack®, with the storage system based on Ceph, and makes available a bunch of Infrastructure-As-A-Service solutions like virtual machines, virtual networks and volumes on demand. The infrastructure provides also some Platform-As-A-Service solutions, such that container-ization (Docker, Kubernetes), cluster on demand (Spark, HTCondor), and Object Storage built on top of Ceph. The authorized members of the community can access up to 200 virtual CPUs, with a limit of 100 virtual machines and 740 GB RAM, 90 Tbytes with volumes on demand and 75 Tbytes on the object storage. CloudVeneto requires the registration at the INFN identity provider.

Muon Collider members have at their disposal two HTC solutions, both based on HTCondor batch system. The first one is located at INFN-CNAF in Bologna. It makes available 6 computing elements and a storage element, the StoRM service, with a grid-oriented standard interface (SRM) and up to 150 Tbytes. Both the HTCondor cluster and the storage system requires the registration in the Virtual Organization (VO) of Muon Collider. The security architecture of the VO is bound to grid extensions of the X509 Public Key Infrastructure, the affiliation to the VO is possible only if using accepted digital certificates. The second HPC solutions available is located at CERN. At the current date, the batch system is still in pre-production state, the total amount of resources is not yet completely defined. The CERN site supplies up to 100 Tbytes on the CERN EOS storage system.

The resources provided by the Muon Collider community are not restricted to hardware infrastructure, but include also scientific data. The community is actively working on the creation of a central repository of datasets [14], containing samples from simulated physical processes, obtained with different tools applied to different contexts. The datasets are published via the storage element at INFN-CNAF site, the only requirement for accessing them is the user registration to the Muon Collider VO. The user is also strongly encouraged to contact the Muon Collider group and clarify the usage scope of the required datasets, so the group can suggest the correct way to use the dataset and avoid any possible misunderstanding. No other restrictions are applied at the moment.

Finally, the Muon Collider project provides the community with an aggregation service [15] for any type of documentation. The service is based on the Confluence platform and provides all the features expected for a collaboration site like web publishing, blog and document repositories. The site is protected with a fine grained access control mechanism; the registration of a member is required.

4. Future works

The entire software management process for the Muon Collider framework meets all the requirements for a production-ready software suite. Nevertheless, several aspects can be improved and the Muon Collider working group is constantly investigating new solutions starting from the build system up to the software distribution.

As mentioned before, a continuous integration system is running on the GitHub site and is able to certify the quality of a release thanks to a powerful functional test suite. However, it is based on the final artifact of the building process, it is necessary to extend the integration to the previous steps and the intermediate artifacts. In this way, it is possible to test any component of the framework separately, with both functional and unit tests. A new integration approach can simplify the creation of new type of artifacts, like packages for each component (see later). The Free and Open Source world offers many solutions with high levels of quality and support, like Jenkins or Travis. They are all highly customizable and suitable for any kind of software management workflows. Some of them are already available in many member sites of the Muon Collider collaboration, for example Jenkins is supplied as a service by INFN or as a platform on demand at CERN. Unfortunately, the build system inherited from the ILC Software toolkit is not completely adequate to be integrated in one of the solution proposed; it encapsulates its own specific workflow so minor changes are required in order to "open" the sandbox and inspect any step of the workflow.

Another activity, strongly related to the continuous integration task and still under investigation, is the creation of new type of artifacts. The Muon Collider Team is preparing a distribution based on RPM packages, specific for CentOS 8 Stream. At the current date, the distribution is not yet complete, it covers the core engines for simulation and reconstruction, together the commonly used processors and extensions. Any detail concerning the service dedicated to the publication of the packages is still under discussion.

A major task scheduled for the future evolution of the Muon Collider framework is the integration with the project Key4HEP [16]. The integration is critical as it could require, not only a re-design of several components of the framework, but also a re-definition of the software management process. The first point is out of the scope of this article, but for what concerns the second one the Muon Collider team is evaluating the solutions proposed by the Key4HEP community. The community provides a powerful toolkit, spack [17], that deals with any aspect of the software management process. Since most part of the ILC Software distribution is already managed by Key4HEP with

PoS(ICHEP2022)226

spack, the first step towards the integration consists on inserting the new components into the Key4HEP build system. The adoption of the Key4HEP tools is mutually exclusive with any solution based on ILC Software suite or applications like Jenkins or Travis.

References

[1] ILC Software, https://github.com/iLCSoft/ilcsoftDoc

[2] DD4Hep, https://dd4hep.web.cern.ch/dd4hep/

[3] Marlin, https://github.com/iLCSoft/Marlin

[4] F.Gaede et al., *LCIO - A persistency framework for linear collider simulation studies*, arXiv:physics/0306114

[5] ACTS, https://acts.readthedocs.io/

[6] The Muon Collider detector https://github.com/MuonColliderSoft/detector-simulation

[7] S.Ceravolo et al, *Crilin: A CRystal calorImeter with Longitudinal InformatioN for a future Muon Collider*, arXiv:2206.05838

[8] Muon Collider Sofware, https://github.com/MuonColliderSoft

[9] Docker, https://www.docker.com/

[10] Muon Collider artifacts at DockerHub, https://hub.docker.com/repository/docker/infnpd/mucoll-ilc-framework

[11] Muon Collider Continuous Integration https://github.com/kkrizka/MuonCollider-docker/actions

[12] Apptainer, previously Singularity, https://apptainer.org/

[13] CloudVeneto, http://userguide.cloudveneto.it/

[14] Muon Collider, Monte Carlo Simulated Samples, https://confluence.infn.it/display/muoncollider/Monte+Carlo+Simulated+Samples

[15] The Muon Collider site https://confluence.infn.it/display/muoncollider

[16] Key4HEP, https://key4hep.github.io/key4hep-doc/

[17] Spack, a package management tool for HEP, https://spack.readthedocs.io/

Acknowledgments

AIDAinnova This project has received support (funding) from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No 101004761.