

Large-scale HPC deployment of Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference (SCAILFIN)

Michael Hildreth^{1,*}, Kenyi Paolo Hurtado Anampa², Cody Kankel², Scott Hampton², Paul Brenner², Irena Johnson², and Tibor Simko³

¹Department of Physics, University of Notre Dame, Notre Dame, IN, USA

²Center for Research Computing, University of Notre Dame, Notre Dame, IN, USA

³IT/CDA Division, CERN, 1211 Meyrin, Switzerland

Abstract. The NSF-funded Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference (SCAILFIN) project aims to develop and deploy artificial intelligence (AI) and likelihood-free inference (LFI) techniques and software using scalable cyberinfrastructure (CI) built on top of existing CI elements. Specifically, the project has extended the CERN-based REANA framework, a cloud-based data analysis platform deployed on top of Kubernetes clusters that was originally designed to enable analysis reusability and reproducibility. REANA is capable of orchestrating extremely complicated multi-step workflows, and uses Kubernetes clusters both for scheduling and distributing container-based workloads across a cluster of available machines, as well as instantiating and monitoring the concrete workloads themselves. This work describes the challenges and development efforts involved in extending REANA and the components that were developed in order to enable large scale deployment on High Performance Computing (HPC) resources. Using the Virtual Clusters for Community Computation (VC3) infrastructure as a starting point, we implemented REANA to work with a number of differing workload managers, including both high performance and high throughput, while simultaneously removing REANA's dependence on Kubernetes support at the workers level.

1 Introduction and Motivation

The National Science Foundation (NSF) has made significant investments in major multi-user research facilities (MMURFs), which are the foundation for a robust data-intensive science program. Extracting scientific results from these facilities involves the comparison of “real” data collected from the experiments with “synthetic” data produced from computationally-intensive simulations. This is the *modus operandi* of MMURFs such as the Large Hadron Collider (LHC), IceCube Neutrino Observatory, and the Laser Interferometer Gravitational Wave Observatory (LIGO).

Comparisons of experimental data and predictions from simulations are abstractions of the specific data analysis techniques developed by the respective communities over several decades. In recent years there has been a tremendous amount of interest in leveraging machine learning (ML) and artificial intelligence (AI) techniques to enhance the analysis of data from these facilities.

* Corresponding author: mhildret@nd.edu

While these facilities have highly-engineered systems for data acquisition and there are corresponding systems for generating simulated data, several data analysis tasks are often shepherded manually or through ad hoc scripts that are not well maintained. Many of these tasks involve well-defined workflows amenable to automation that would free up time which could be directed at data analysis innovations and insights. The scientific potential of these facilities can be significantly enhanced by developing cyberinfrastructure (CI) that streamlines specific high-impact use cases.

Through NSF funding to DASPOS [1] and DIANA [2] and in collaboration with CERN, we have developed CI components such as REANA [3] that can run large scale data-intensive analysis workflows in high energy physics (HEP) and bioinformatics on cloud computing resources. REANA is designed to enable reproducibility, reusability, and optimization of the analysis pipeline.

In parallel, NSF has supported the development of a new class of data analysis techniques that leverage ML to improve the discovery potential of MMURFs. Particularly significant is the emergence of a class of likelihood-free inference (LFI) techniques that are needed when the predictions for the data are implicitly defined by simulation, which often leads to an intractable likelihood function. This formulation applies to analysis of data from the LHC, LIGO, radio astronomy, cosmology, phylogeny, epidemiology, and condensed matter physics. It has also become a hot topic in ML and has emerged as a *lingua franca* for data analysis that accelerates convergent research.

Thus far, likelihood-free algorithms have been implemented only on individual machines and in ad hoc scripts. Thus, a scalable implementation for data-intensive applications where the simulation is computationally intensive (e.g. LHC) is needed. The Scalable Cyberinfrastructure for Artificial Intelligence and Likelihood Free Inference (SCAILFIN) project will implement the recently developed algorithms on top of the REANA system with specific LHC science topics driving the initial effort. The resulting applications will run on large (e.g. cloud/HPC) resources to analyze LHC data.

1.1 Likelihood-Free Inference

Recent work in particle physics has focused on extracting maximal information from the large data sets that have been and will be accumulated at the LHC. Many simple inference techniques cannot be easily extended to higher-dimensional realizations due to the need to populate all regions of parameter space with sufficient simulated data in order to estimate the behaviour of, for example, signatures of signal and background processes in each “bin”. One effort to address these shortcomings uses Machine Learning (ML) classifiers to estimate likelihood ratios for the purposes of inference. These Likelihood-Free Inference techniques can “teach” algorithms the behaviour of a multi-scale physics simulation where an analytical form of the likelihood is impossible to obtain [4-6]. More recent versions of this work have incorporated additional information (such as gradient densities) that are available in the simulation to improve the accuracy of the estimators and to dramatically lower the quantity of training data needed for good results [7-9].

1.2 The REANA System

The REANA[3] system was developed by the DASPOS and DIANA-HEP projects in conjunction with CERN IT. Built from commodity elements, REANA encapsulates the technology for preserving, composing, and orchestrating the processing of the different workflow steps. Within REANA, the preserved executables can be reinstated according to the instructions in the workflow scripts by a job scheduler and workflow system that can be attached to any generic computing resource that supports container execution. A

diagram of the REANA system is shown in Figure 1. In this framework, an analyst can preserve individual workflow steps in containers and string them together with a series of scripts that enable extremely complex workflows. Workflow segments and individual steps can be shared and re-composed in different workflows. The characteristics of each workflow step, including inputs and outputs, execution environment, etc., are specified in YAML/JSON format. Linux containers that execute the individual steps of a workflow are orchestrated by the yadage workflow language[10] and execution engine with Kubernetes as the job execution controller. Common Workflow Language (CWL)[11] is also supported.

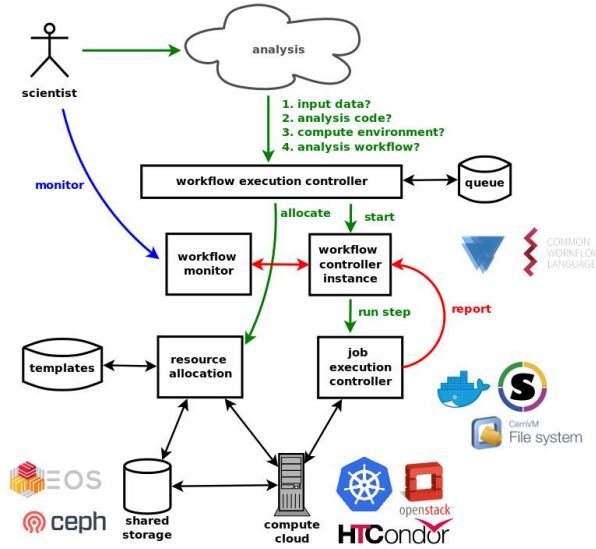


Figure 1. The REANA system architecture

2 SCALFIN

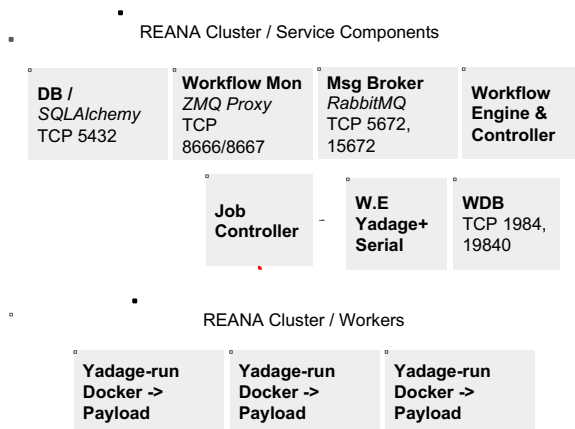
The SCALFIN project [12] aims to create the CI necessary to deploy workflow systems like REANA on large-scale HPC centers in order to facilitate the ML necessary to use Likelihood-Free Inference techniques or other methods needing large-scale simulated samples for training. The subject of this paper is a description of how REANA has been modified for deployment on these large-scale systems and tested with complex workflows.

2.1 Initial Development

A critical design decision was to run REANA as an edge service. All of the elements of the “reana-cluster” component of REANA that handles workflow orchestration, job submission, etc., can be provisioned using infrastructure developed by the Virtual Clusters for Community Computation (VC3) project[13]. This infrastructure allows a user to provision and distribute a pre-defined “cluster” environment over heterogeneous resources with all configuration accomplished in “user” space; no root access is needed. The “reana-cluster” has been established as a template cluster configuration option for VC3, enabling a template VC3 headnode that contains all of the necessary functional pieces of REANA to be deployed at HPC sites. A symbolic representation of the different components of the

reana-cluster as originally developed by the CERN team is shown in Figure 2. As mentioned above, note that almost all of the pieces are commodity components, readily available from outside developers. This makes the REANA infrastructure readily portable.

One component of the VC3 infrastructure that has made the deployment of this system is the existence of a job-submission translation layer that already exists in VC3. VC3 is capable of submitting tasks to a wide variety of schedulers: Torque (Blue Waters), SLURM (NERSC, PSC-Bridges, Stampede2), HTCondor, LSF, SGE, PBS. This functionality has provided a great deal of flexibility as to where VC3 can be used.



12

Figure 2. Reana-cluster components. Each component runs in an individual container, with orchestration provided by Kubernetes.

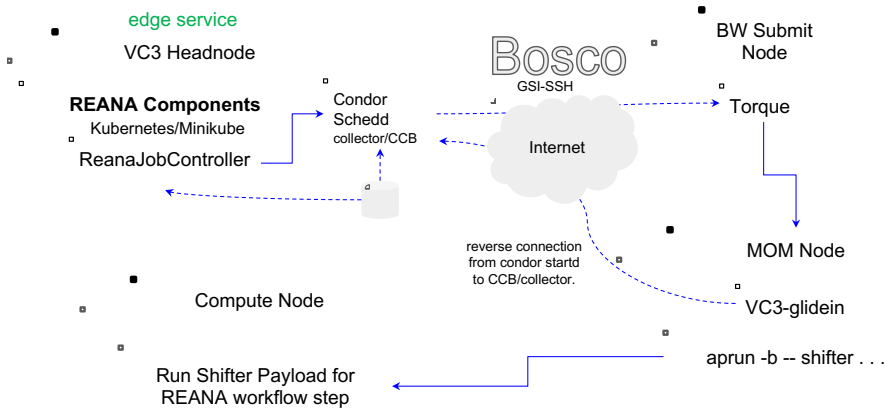
As seen in Figure 1, the original REANA system uses yadage to assemble sequence of job steps based on containerized executables which are instantiated by a Kubernetes cluster for job orchestration. In order to facilitate the conversion to running on large HPC systems, the SCAILFIN team developed an HTCondor[14] backend for REANA. This required modifications to the reana-job-controller and providing a job wrapper for each workflow step so that it could be submitted to HTCondor. The modified reana-job-controller submits each workflow step to a local Condor scheduler. Kubernetes, set up via Minikube in the VC3 headnode, is still used to coordinate the containers running the other services that comprise the reana-cluster.

2.2 Deployment

The VC3/REANA system has now been deployed and tested at a variety of sites with a wide range of operational configurations. Once the REANA VC3 cluster template was implemented, deployment on local HPC infrastructure at the University of Notre Dame, and on, for example the Bridges machine at the Pittsburgh Super Computing Center was quite straightforward; the system worked essentially “out of the box”. The REANA system has also been deployed on Blue Waters, before it was decommissioned, and CORI at NERSC. That required some additional modifications to both the REANA and the VC3 architectures. In particular, support for GSI-SSH authentication had to be added to the VC3 infrastructure and the capability to renew proxies through the VC3 website was enabled. Other patches to support Cray linux environments were also made. In addition, automatic local configuration detection and support for Shifter and Singularity were added so that

linux containers would run on the appropriate local systems[15]. Figure 3 shows the configuration that was deployed on Blue Waters. The translation layer for job submission used in this case is BOSCO[16]. Performance and scale-out tests are ongoing.

SCAILFIN on Blue Waters



20

Figure 3. VC3/REANA configuration as deployed on Blue Waters. The MOM nodes host the local PBS batch system that distributes tasks to the compute nodes.

2.3 Workflows

In order to test the implementation, a complex workflow was run to completion using this infrastructure on Blue Waters. This particular workflow, whose description can be found at <https://github.com/reanahub/reana-demo-bsm-search>, performed a complete analysis searching for physics beyond the Standard Model, comparing simulated samples to data, doing normalizing fits, and extracting limits. The workflow contains thousands of individual steps and involves data, signal, simulation, merging, fitting and plotting steps, all of which are executed in linux containers. Complex workflows such as these are expected to be necessary for training LFI models at scale and thus represent an appropriate test algorithm that exercises all features of the REANA platform on an HPC system.

3. Future Evolution

SCAILFIN is a two-year project now entering its last six months of funding. In the remaining time, scale-out studies of complex workflows will be performed on a variety of HPC machines. Integration of SCAILFIN modifications to the REANA platform will be complete, rendering REANA capable of deployment on a much wider variety of systems. This work is supported by the Office of Advanced Cyberinfrastructure of the US National Science Foundation with grant OAC-1841448. We would like to thank our colleagues in CERN IT and the rest of the SCAILFIN team for their support and collaboration.

References

1. Data and Software Preservation for Open Science (DASPOS); <https://daspos.crc.nd.edu>.
2. DIANA/HEP website. <http://diana-hep.org>.

3. REANA github: <http://reanahub.io/>.
4. See, for example, K. Cranmer, J. Brehmer, G. Louppe, arXiv:1911.01429 (2019); or K. Cranmer, J. Pavez, and G. Louppe, arXiv:1506.02169 (2015).
5. G. Louppe, K. Cranmer, and J. Pavez, “carl: a likelihood-free inference toolbox”, J. Open Source Software (2016).
6. P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, Eur. Phys. J., **C76** (5):235 (2016).
7. J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer, PNAS March 10, (2020) **117** (10) 5242-5249.
8. J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, Phys. Rev. D **98**, 052004 (2018).
9. J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, Phys. Rev. Lett. **121**, 111801 (2018).
10. K. Cranmer and L. Heinrich, J. Phys. Conf. Ser., **898**(10):102019 (2017).
11. www.commonwl.org
12. <https://github.com/scailfin>
13. <https://www.virtualclusters.org>
14. HTCondor: <https://research.cs.wisc.edu/htcondor/>
15. See K. Anampa, et al., “Abstracting container technologies and transfer mechanisms in the Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference (SCAILFIN) project”, these Proceedings.
16. BOSCO: <https://osg-bosco.github.io/docs/>