

Efficient Iterative Calibration on the Grid using iLCDirac

Oleksandr Viazlo^{1,2,*} and André Sailer^{1,**}

¹CERN, 1211 Geneva 23, Switzerland

²Florida State University, Tallahassee, Florida

Abstract. Software tools for detector optimization studies for future experiments need to be efficient and reliable. One important ingredient of the detector design optimization concerns the calorimeter system. Every change of the calorimeter configuration requires a new set of overall calibration parameters which in turn requires a new calorimeter calibration to be done. An efficient way to perform calorimeter calibration is therefore essential in any detector optimization tool set. In this contribution, we present the implementation of a calibration system in iLCDirac, which is an extension of the DIRAC grid interware. Our approach provides more direct control over the grid resources to reduce overhead of file download and job initialisation, and provides more flexibility during the calibration process. The service controls the whole chain of a calibration procedure, collects results from finished iterations and redistributes new input parameters among worker nodes. A dedicated agent monitors the health of running jobs and resubmits them if needed. Each calibration has an up-to-date backup which can be used for recovery in case of any disruption in the operation of the service. As a use case, we will present a study of optimization of the calorimetry system of the CLD detector concept for FCC-ee, which has been adopted from the CLICdet detector model. The detector has been simulated with the DD4hep package and calorimetry performance has been studied with the particle flow package PandoraPFA.

1 Introduction

Future lepton and hadron collider experiments require new detector concepts which will utilize both experience of the current experiments and novel detector technologies. The design of a new detector includes the optimization of the detector geometry, taking into account technological feasibility for material budgets, single-hit or timing resolutions, and cost constraints.

The optimization of the detector design is a long and time-consuming process. The simplified procedure can be described by the following steps: 1) prepare a new geometry and configuration of the detector; 2) run reconstruction of a set of physics and background processes; 3) estimate the performance of each sub-detector system and the detector overall. Repeat the process until the optimal configuration of the detector is found. For promising detector designs, physics analyses with a complete set of signal and background processes

*e-mail: oleksandr.viazlo@cern.ch

**e-mail: andre.sailer@cern.ch

are done to estimate detector capabilities on precision measurements and searches for new physics.

One important ingredient of the detector design optimization concerns the calorimeter system. One of the most promising options for precise jet-energy reconstruction is particle flow clustering with high-granularity calorimeters.

Because the particle flow clustering uses information from reconstructed tracks to separate nearby calorimeter clusters from charged and neutral particles, the tuning of the reconstruction is particularly involved. If the energy estimate in the calorimeters is not precise enough real neutral particles could be lost, or fake clusters could be reconstructed. Every change of the calorimeter configuration requires a new set of overall calibration parameters which in turn requires a new calorimeter calibration to be done.

This is why an efficient way to perform the calorimeter calibration is an important piece for detector development and optimization studies. A new calorimeter calibration can also be required after the calorimeter optimization phase is finished in cases such as when a new Geant4 version is used, or a different Geant4 physics list is used or if there were significant changes in the tracking or particle flow algorithm performances.

In this article we are discussing an improvement of the calorimeter calibration procedure for detector models CLICdet [1] and CLD [2] which are proposed detectors for the future e^+e^- colliders CLIC and FCC-ee respectively. The simulation and reconstruction for these detectors are done with the iLCSoft framework, particularly, the geometry description is done with DD4hep [3, 4] and the event reconstruction with Marlin [5] and the PandoraPFA particle-flow clustering package [6].

We describe the porting of the existing calibration procedure to the iLCDirac [7] extension of the DIRAC interware [8]. This allows the use of the GRID resources for parallel execution of several calibrations. The implementation as an iLCDirac system also benefits from the available functionality in the DIRAC software: interfaces for monitoring, bookkeeping, data management, workload management, authentication, and configuration as described in Section 4.

2 Calorimeter Calibration for Particle Flow Clustering

The calorimeter calibrations provide coefficients to translate the energy seen by the calorimeter system to the total energy of particles stopped in the calorimeter. The calibration procedure provides two sets of constants: one set to scale the raw energy deposited in the ECAL and HCAL sensors to the energy deposited by the particles, a second set to correct the reconstructed energy of particles during the clustering step of the particle flow algorithms. In addition, the likelihood tables for the photon reconstruction using photon shower shapes are recomputed for the new set of constants.

To reach the usual precision of 2% on the raw energy measurement and 0.5% on the energy for PandoraPFA clusters [9], the calibration requires 20–30 iterations in total. Each iteration corresponds to the running of the complete reconstruction of a number of large datasets. The original calibration workflow [10] consists of the following steps: 1) initialize the starting calibration constants, 2) submit jobs to a batch system to perform reconstruction with the current set of calibration constants, 3) after all jobs are finished, retrieve the output from each job (usually a set of histograms), merge them and perform calculations of the new calibration constants set by comparing the truth information with the values obtained from the reconstruction. Steps 2) and 3) are repeated until the required accuracy for the reconstruction is achieved.

Besides running of the reconstruction, step 2 consists of matching and allocating compute resources by the batch scheduler, initialization of the environment, software installation and

downloading of input data to each node. All of these actions have to be repeated for each iteration which leads to a significant delay, overhead, and unnecessary transfer of the same data to the worker node. The reconstruction of the events is trivially parallelizable. However, to benefit from splitting the workload, a sufficient number of resources have to be available otherwise the jobs will wait in a queue.

To keep the reconstruction jobs from waiting in queues too long, a dedicated cluster was used for the original calibration procedure. This dedicated cluster, however, only contains a small number of worker nodes.

One of the key ideas of the new calibration procedure is to optimize the calorimeter calibration workflow by removing the need to submit jobs to the batch system on each iteration by distributing new sets of calibration constants to the worker nodes on demand. For this purpose a new iLCDirac service was developed which supervises calibrations and takes care of managing compute resources and data transfer. By submitting jobs only once, the potential waiting time in batch queues is reduced which allows the use of shared grid resources instead of a dedicated cluster.

3 Goals of the Calibration Service

The calibration service has to fulfill the following requirements:

- maximal automation of the calibration procedure to increase calibration performance and throughput
- easy configuration of calibration parameters and detector settings
- support for multiple detector models which use the Marlin reconstruction and PandoraPFA particle-flow packages (CLICdet, CLD, ILD [11])
- bookkeeping of all executed calibrations; storing of calibration meta-data to easily re-run calibrations with identical configurations if needed
- fault-tolerance: automatically restart in case of any disruption in the operation of the service, monitor the health of running jobs

To achieve the desired throughput and performance, access to a large set of computing resources is needed. The access to heterogeneous resources is provided through the DIRAC interware and includes GRID resources, but can be anything the DIRAC instance has access to. DIRAC offers the functionality to restart services and automatically monitor and resubmit individual jobs to the computing resources on behalf of the user that created the calibration. The interfaces for the linear collider software provided with iLCDirac allow running of all detectors implemented in the respective packages. What is required for the new service are the bookkeeping, the configuration for the calibrations, and the implementation of the calibration workflow, as well as the communication between the individual reconstruction jobs and a central instance to calculate the constants needed for the next iteration.

4 Implementation of the Calibration Service

The new calibration service implements the workflow shown in Figure 1.

A user creates a request to the calibration service to start a new calibration. The user has to provide a list of input files and the calibration settings to be used in the calibration. The calibration settings are inherited from a template which contains default detector settings for each specific detector (e.g., CLICdet, CLD) and calibration settings which the user has to set up (e.g., the number of worker nodes, version of packages to use). The list of all settings is

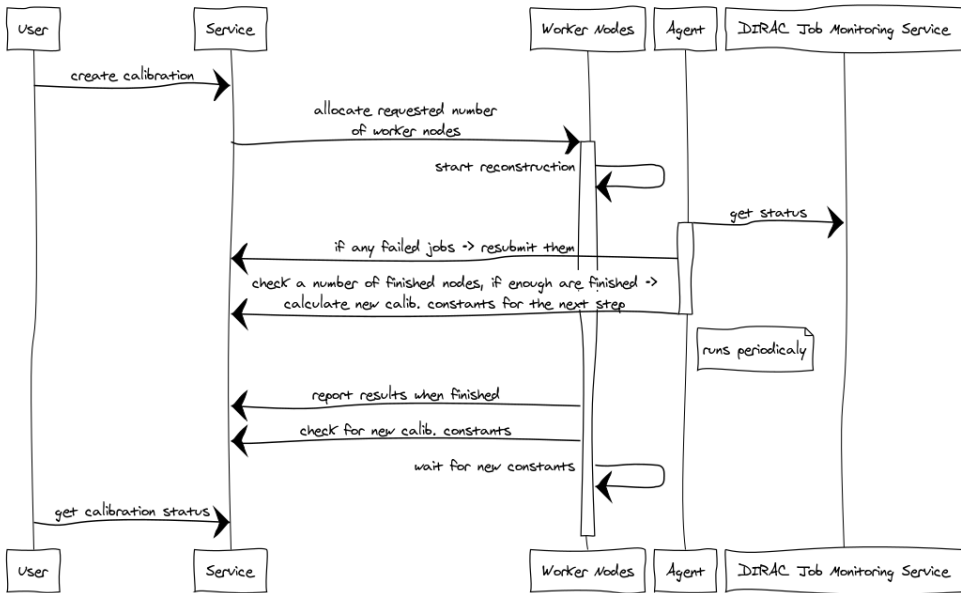


Figure 1: Calibration workflow of the iLCDirac calibration service

documented in the user guide [12]. After receiving a user request, the service allocates the requested number of worker nodes for the entire duration of the calibration by submitting jobs to the DIRAC job manager service. Once a node is allocated, the iLCDirac client is installed, the performance of the worker node is estimated and the environment initialised, all of which takes about 5 minutes. After that, the service copies the input data for all iterations to the node. When everything is set up and the input data is copied event reconstructions starts. The Calibration agent monitors the status of each worker node by periodically sending a requests to the DIRAC job monitoring service with a list of the worker IDs belonging to the current calibration. If reconstruction fails, the agent requests the Calibration service to resubmit reconstruction jobs with the latest set of the calibration constants. When a reconstruction is finished, the worker node sends its results of the reconstruction (e.g., a set of histograms used to evaluate the correctness of reconstructed energy) to the service and waits for the new set of calibration constants, by repeatedly querying the service. The service either responds with a wait instruction or the new set of constants for the next iteration. The service response can also contain instructions to move to a new phase of the calibration. The Calibration agent periodically checks how many results are collected by the calibration service. If the number of results is larger than a configurable fraction the Calibration service merges all results and runs the calibration script which compares the truth information with the values obtained from the reconstruction and calculates the updated set of calibration constants and then returns them to the worker nodes on their next request. The default value of the configurable fraction is 90%. This fraction prevents the slow down of the calibration in cases where one of the allocated worker nodes is significantly slower than the others and to avoid the calibration getting bogged down when the reconstruction is stuck on one of the machines. When the new calibration constants are available, the worker nodes pick them up, update their reconstruction configuration and start a new reconstruction. These steps are repeated until the required

precision of the calibration constants is achieved. Then all computing resources are released and the calibration is bookmarked as successfully finished.

A user can send a request to the calibration service to check the status of any calibration which has been started by him. Also, a user can request to stop the calibration or delete the record of the calibration from the database. Online monitoring of the calibration status is possible with DIRAC job monitor web application.

It is worth to highlight that this implementation satisfies all the goals described in Section 3. The worker nodes are allocated only once at the beginning of the calibration and used for all iterations. All the required data are copied to the nodes only once and then reused for each iteration to run the reconstruction with a new set of calibration constants. This implementation significantly increases the efficiency of the calibration compared to the method described in Section 2.

The Calibration service creates a backup of each calibration after each iteration. It allows the service to recover and continue execution of all active calibrations, in case of a restart of the service by administrator or automatic restart of the service after some disruption in the operation.

5 Usage example

One of the usage examples of this newly developed calibration procedure is a study of optimization of the Silicon–tungsten ECAL detector for the CLD concept for FCC-ee presented below. The original design of the ECAL consists of 40 layers which allow reaching an excellent photon energy resolution. However, the large silicon surface makes the ECAL the most expensive system of CLD. This is why a study has been done to understand the degradation of detector performance by reducing the number of layers in ECAL.

Four options of ECAL with different numbers of layers were considered. Three options consist of 40, 30 or 20 uniformly distributed layers. The last option consists of a combination of 20 thin and 10 thick layers, similarly to the ILD detector concept. Each of the options has a different thickness of the absorber layers (tungsten plates) in order to keep the same total ECAL thickness of $\approx 22X_0$. All other parameters of the ECAL segmentation remain unchanged.

The main parameter of interest was photon energy resolution which almost only depends on ECAL performance. Energy resolutions have been studied for photons with energies of 5, 10, 20, 50 and 100 GeV and are shown in Figure 2. As expected, the best photon resolution over the whole energy range is achieved with the 40 layer configuration, while the worst resolution is achieved with the 20 layer configuration. For the 20+10 mixed layer configuration, the resolution is better than the 30 layer option for low photon energies, while it is worse for higher energies. The ECAL configuration with only 20 layers results in significant degradation of the photon energy resolution, while the 20+10 and the 30 layer options are good alternatives to the default 40 layers – both would reduce the cost of the detector by about 25% for the silicon alone, while not compromising too much on the energy resolution.

With the new calorimeter calibration procedure, it was possible to run the calibration for all four detector configurations simultaneously. The calibrations were performed with default precision (2% on the raw energy measurement and 0.5% on the energy for PandoraPFA clusters) using 200 worker nodes per calibration.

The total time was approximately 3 hours, including the time to submit the jobs to the grid sites. However, the time from submission to job matching depends on the user priority and availability of computing resources. If necessary, these priorities could be tuned. An individual reconstruction iteration on a single worker node took about 10 minutes, which is

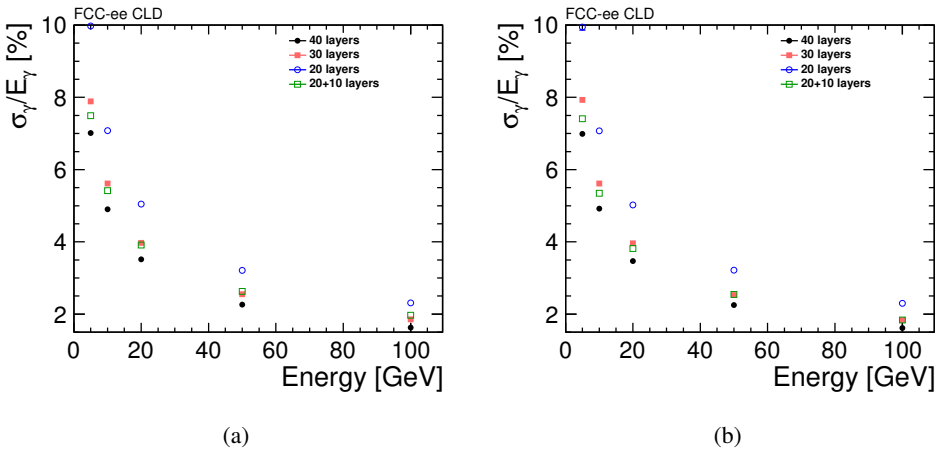


Figure 2: Photon energy resolution as a function of energy in the (a) barrel and (b) endcap regions. The results are shown for different ECAL layer configurations.

of similar magnitude than the initialisation time for a job slot. This illustrates the benefit of the calibration service over repeated job submission.

6 Summary and Outlook

The new Calibration service provides automation and increased speed of the calibration procedure, simplicity of usage, possibility to run many calibrations simultaneously and web-based calibration monitoring. It provides an efficient use of Grid resources and exploits the high-performant tools and services of DIRAC. And it has been successfully used for FCC-ee ECAL optimization studies.

There are still a few potential improvements of the service which can even further enhance its functionality: 1) provide the possibility for the user to configure the order of calibration steps (an order in which calibration constants are derived) to allow to do partial calibration of ECAL or HCAL only or to add some custom steps; 2) add functionality to produce required Geant4 simulation input files automatically.

Acknowledgement

This work benefited from services provided by the ILC Virtual Organisation, supported by the national resource providers of the EGI Federation. This research was done using resources provided by the Open Science Grid, which is supported by the National Science Foundation and the U.S. Department of Energy's Office of Science.

References

- [1] D. Arominski et al., *A detector for CLIC: main parameters and performance* (2018), 1812.07337, <https://cds.cern.ch/record/2649437>
- [2] N. Bacchetta et al., *CLD – A Detector Concept for the FCC-ee* (2019), 1911.12230
- [3] M. Frank, F. Gaede, M. Petric, A. Sailer, *DD4hep* (2018), <https://doi.org/10.5281/zenodo.592244>

- [4] A. Sailer, M. Frank, F. Gaede, D. Hynds, S. Lu, N. Nikiforou, M. Petric, R. Simoniello, G.G. Voutsinas, *J. Phys.: Conf. Ser.* **898**, 042017 (2017)
- [5] F. Gaede, *Nucl. Instrum. Meth.* **A559**, 177 (2006)
- [6] J. Marshall, M. Thomson, *Eur. Phys. J.* **C75**, 439 (2015)
- [7] C. Grefe, S. Poss, A. Sailer, A. Tsaregorodtsev, *J. Phys. Conf. Ser.* **513**, 032077 (2013), CLICdp-Conf-2013-003, <https://doi.org/10.1088/1742-6596/513/3/032077>
- [8] A. Tsaregorodtsev, M. Bargiotti, N. Brook, A.C. Ramo, G. Castellani, P. Charpentier, C. Cioffi, J. Closier, R.G. Díaz, G. Kuznetsov et al., *J. Phys.: Conf. Ser.* **119**, 062048 (2008)
- [9] S. Green, Ph.D. thesis (2017), presented 11 Oct 2017, <https://cds.cern.ch/record/2293158>
- [10] *Pandora calibration and analysis tools in iLCSoft / Marlin framework*, accessed 1 Mar 2020, <https://github.com/PandoraPFA/LCPandoraAnalysis>
- [11] H. Abramowicz et al. (ILD), *The ILD detector at the ILC* (2019), 1912.04601
- [12] O. Viazlo, *Running Pandora Calorimeter Calibration*, accessed 1 Mar 2020, <http://ilcdirac-doc.web.cern.ch/ilcdirac-doc/DOC/Files/UserGuide/calibration.html>