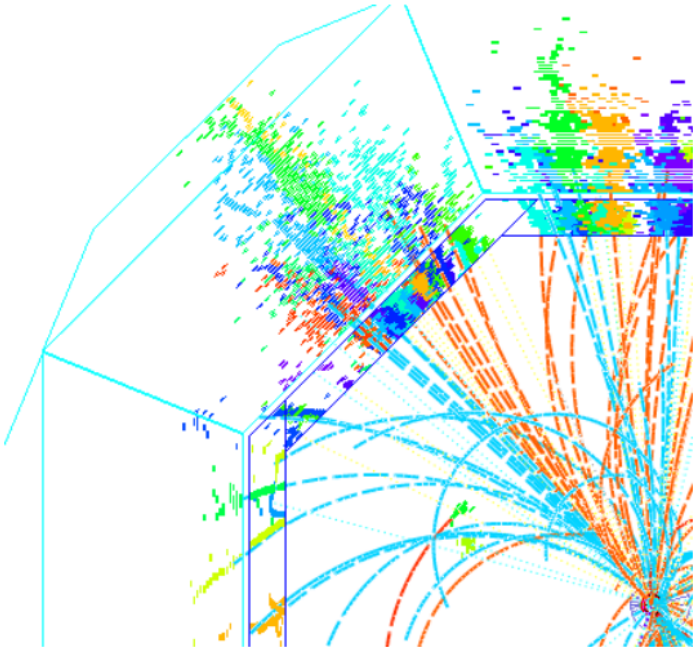
Efficient Iterative Calorimeter Calibration on the Grid using iLCDirac

Andre Sailer, Oleksandr Viazlo

CLICdp collaboration meeting 2019, CERN

28 August 2019

Introduction



- Calorimeter calibrations provide coefficients, which 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:
  - first set to scale the raw energy read out by the ECAL and HCAL electronics to the energy deposited by the particles.
  - second set to correct the reconstructed energy of particles during the clustering step of Pandora.

When is a new calibration needed?

- change in the calorimeter geometry
- new Geant version
- new physics list
- changes in the tracking or particle flow algorithm performances

An efficient way to perform calorimeter calibration is an important piece for detector development and optimization studies.

Old Pandora calorimeter calibration procedure

Calorimeter calibration script

- Originally, the calibration script was developed for ILD detector optimization studies (by S. Green)
  - it was developed to be run on the Cambridge computing cluster
  - it was tied to the ILD detector geometry
- For CLIC studies the calibration script has been heavily modified (by N. Nikiforou and M. Weber) in order:
  - to be run at one of CERN clusters
  - to support the newest reconstruction software (DD4Hep and Marlin)
  - to be applicable for CLIC detector model

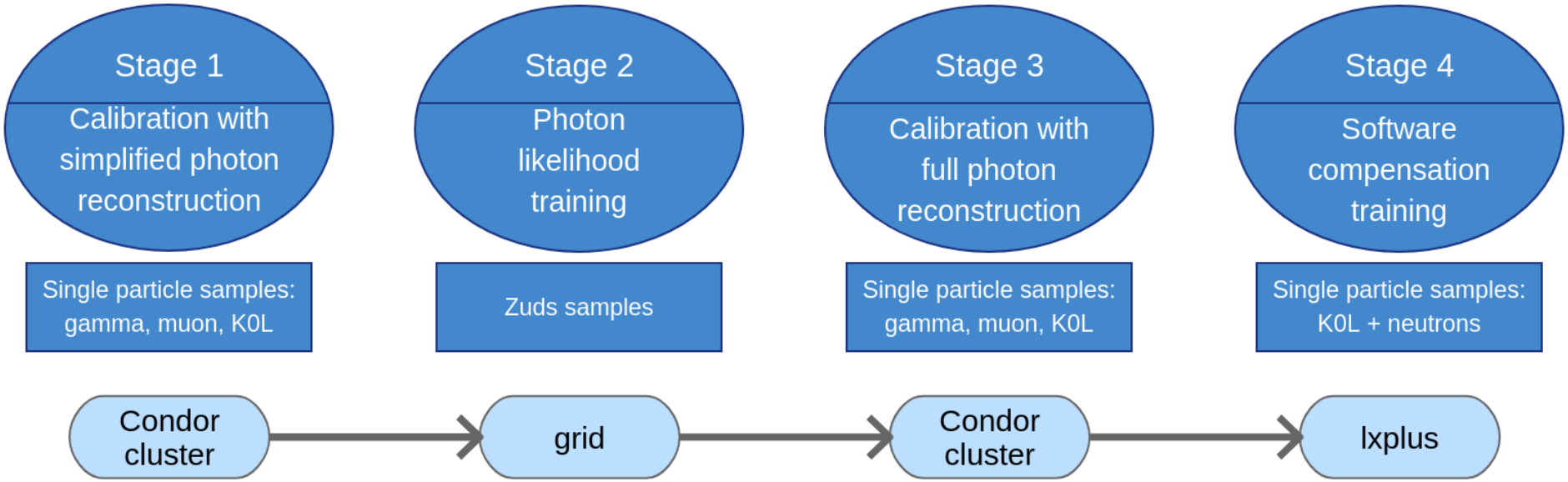
Existing issues

- the main calibration script consists of thousand lines of code written in BASH with many additional calls of python modules
- to add support for the CLD detector model a second instance of the scripts was created since the calibration procedure depends on the detector geometry and configurations
- calibration runs on a standalone condor cluster with a max limit of 100 workers
- data has to be copied locally to the cluster
- calorimeter calibration takes days to be performed

Goals of the new calibration procedure

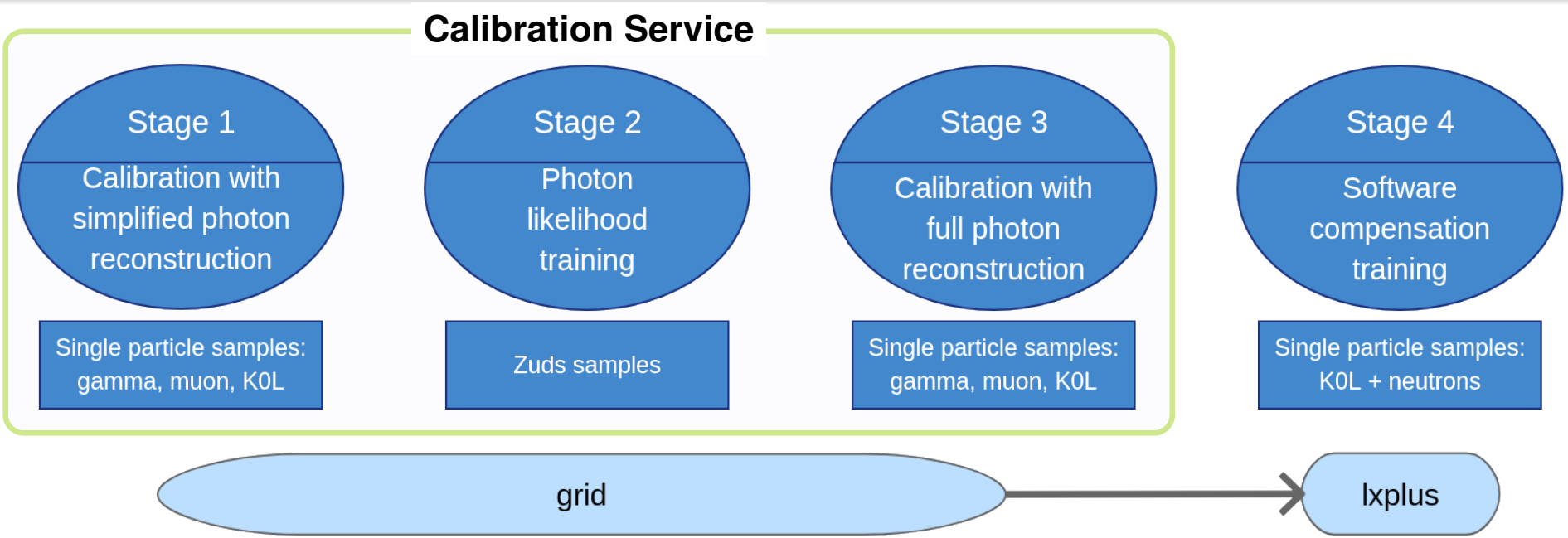
- maximum automization of the calibration procedure
- use the grid instead of a local cluster (more resources)
- to be implemented as a native iLCDirac service which allows exploiting its tools and interfaces for monitoring and bookkeeping
- support for multiple detector models (CLICdet, CLD, other models which use Pandora)

Calibration procedure



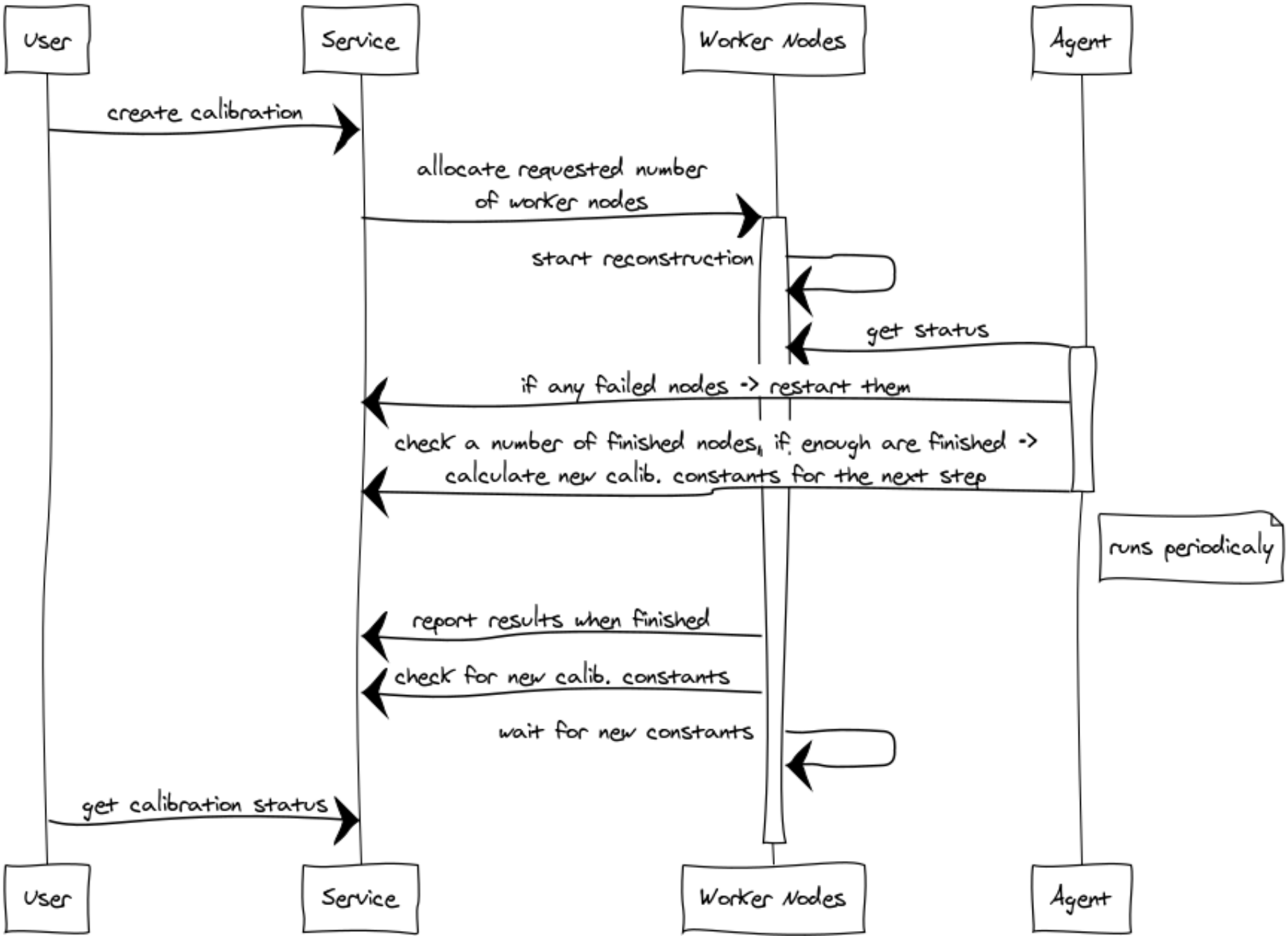
- Calibration consists of four different stages:
  - Stage 1 and 3 are run on the cluster, while stage 2 is run on grid and stage 4 on lxplus/local machine
  - output from one stage has to be plugged in for next stage manually
- Calorimeter optimization study requires to perform calibration for each separate configuration of the detector which can be very time consuming
- New “calibration service” speeds up and simplifies the calibration process by running Stages 1-3 on the grid. It takes cares of bookkeeping of intermediate results (Stage 4 cannot be parallelized)

Dirac calibration service



- Calibration service is a native iLCDirac service
  - python modules with a set of classes with large part of code covered by unit tests
  - grid job monitoring and bookeeping (online calibration status)
- Controls the whole chain of a calibration procedure
- Reserve grid resources for the entire calibration procedure
- Download all required input files for all steps only once per worker node
- Possibility to run multiple calibrations simultaneously
- Significantly faster calibration:
  - no hard limit on number of jobs
- Multiple detector support (CLIC, CLD, other models which use Pandora)
- Simple configuration of calibration parameters and detector settings

Calibration workflow



Reliability of the Calibration Service operation

Calibration Service

- Automatically restarts in case of any disruption in the operation.
- Contains up-to-date backup of each calibration which can be used for recovery.
- Configurable threshold on the fraction of finished jobs to start next step:
  - “stuck” jobs will not affect calibration
  - allows avoiding significant slow down of the calibration by one slow machine
- Stores results of the calibration at the server for configurable amount of time
- User can monitor online calibration status with iLCDirac web-interface

Calibration Agent

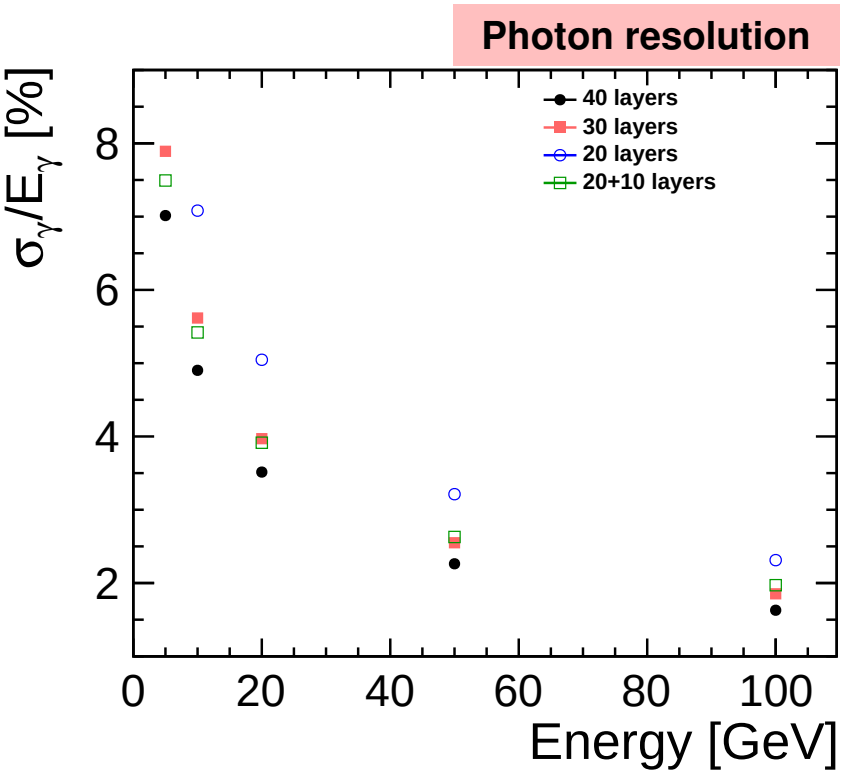
- Monitors the health of running jobs
- In case of job failure - resubmits job with input parameters from the latest step

Example of user input for calibration

```
*** Calibration parameters ***
numberOfJobs                                    100
fractionOfFinishedJobsNeededToStartNextStep     0.9
digitisationAccuracy                            0.02
pandoraPFAAccuracy                              0.005
startPhase                                      0
startStage                                      1
stopPhase                                       99
stopStage                                       99
disableSoftwareCompensation                     True

*** Detector model ***
detectorModel                                   FCCee_o1_v04_ecal20_10.tgz
ecalBarrelCosThetaRange                         [0.0, 0.643]
ecalEndcapCosThetaRange                         [0.766, 0.94]
hcalBarrelCosThetaRange                         [0.15, 0.485]
hcalEndcapCosThetaRange                         [0.72, 0.94]

*** ECAL parameters ***
nEcalThickLayers                                10
nEcalThinLayers                                 20
ecalResponseCorrectionForThickLayers            2.0

*** SW version ***
marlinVersion                                   ILCSoft-2019-07-09_gcc62
DDCaloDigiName                                  MyDDCaloDigi_10ns
DDPandoraPFANewProcessorName                    MyDDMarlinPandora_10ns

*** Input/Output ***
outputPath                                      /ilc/user/o/oviazlo/fccee_caloCalib/
outputSE                                        CERN-DST-EOS
steeringFile                                    fcceeReconstruction_noSWC.xml
gammaFiles                                      fccee_10gev_gamma_FCCee_o1_v04_ecal20_10.txt
kaonFiles                                       fccee_50gev_K0L_FCCee_o1_v04_ecal20_10.txt
muonFiles                                       fccee_10gev_mu_FCCee_o1_v04_ecal20_10.txt
zudsFiles                                       fccee_380gev_Z_uds_FCCee_o1_v04_ecal20_10.txt
nEvtsPerFile                                    {'gamma': 500, 'kaon': 500, 'muon': 500, 'zuds': 50
```

Optimization of ECAL of CLD detector model for FCC-ee

- CLD ECAL performance for different sampling options
  - all options have the same total thickness of $\approx 22\ X_0$
  - FCC-ee centre-of-mass energies: 91.2 - 365 GeV



Layer structure	Thickness tungsten alloy [mm]	Total thickness per layer [mm]
40 uniform	1.9	5.05
30 uniform	2.62	5.77
20 uniform	3.15	7.19
20 thin + 10 thick	1.9 + 3.8	5.05 + 6.95

- 40 layer configuration provides the best photon performance
- 20+10 layer configuration provides better performance at low energies compared to 30 layers which better fits needs of FCC-ee
- 20 layer option leads to significant degradation of photon resolution

Summary and Outlook

New calibration service provides:

- significant automization and increased speed of calibration procedure
- simplicity of usage
- possibility to use all grid resources for calibration
  (and run many calibrations simultaneously)
- no need for “babysitting” your calibrations
- web-based calibration monitoring (by means of iLCDirac)

Next steps

- Deploy into production
- Documentation
- Execution sequence: allow a user to specify in which sequence calibration steps have to be executed
  - allows adding extra calibration steps required in specific cases (e.g. calorimeter with different layer thicknes)
- Simulate required input files automatically

Thank you for your attention!