



Tools for FCC Productions The iLCDirac instance

Lorenzo Valentini, Andre Sailer

CERN-EP-SFT

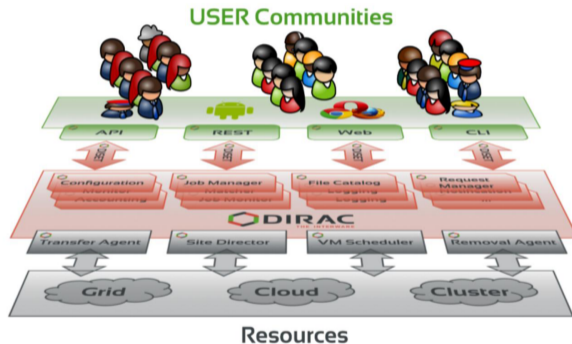
7th FCC Physics Workshop
Jan 31st 2024



Dirac in a nutshell

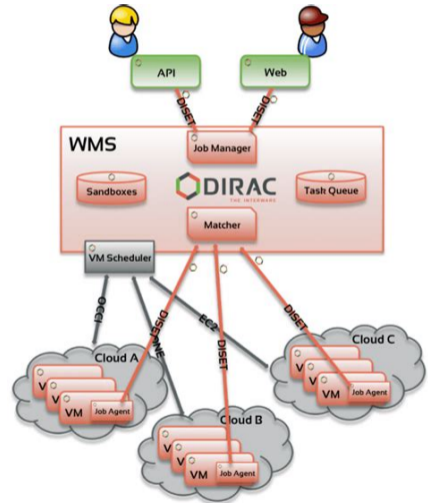
Why the Dirac interware?

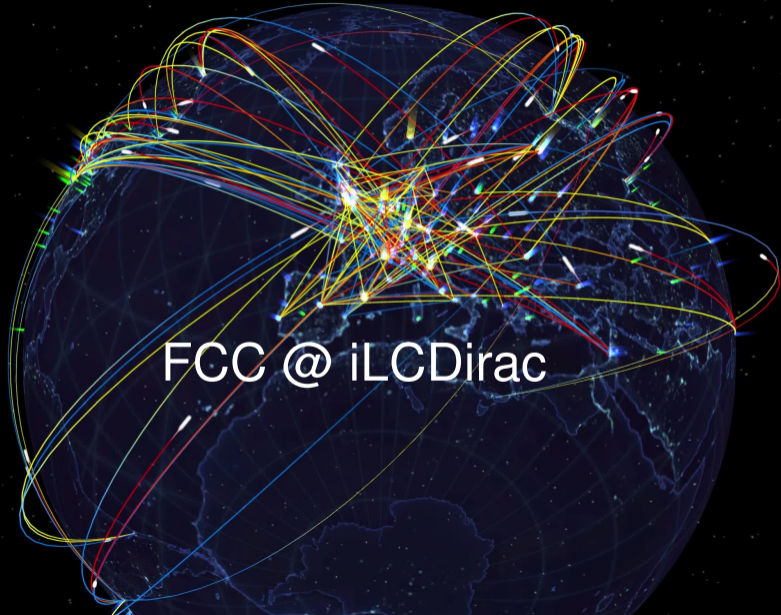
- ▶ Layer between users and resources: homogeneous interface to heterogeneous resources
- ▶ Automated "routine" tasks
- ▶ Monitoring/Controlling/Bookkeeping made easy with the Web Portal and command line interfaces



Dirac in a nutshell

- ▶ Data replication across the grid
- ▶ Accessibility of data by anyone from anywhere
- ▶ Great uptime and failover recovery
- ▶ Workload management over the available resources





FCC @ iLCDirac

Tutorial: Whizard and Delphes

- ▶ We show an example for a transformation using Whizard for event generation and Delphes for fast simulation.
- ▶ We use Whizard 2 to generate 100 events for the $ee \rightarrow ZH$ process, at an energy of 240 GeV
- ▶ We then want to do a fast simulation for the IDEA detector using Delphes, in particular the delphes executable for processing data in STDHEP format (since that is the output of Whizard), and some specific IDEA configuration cards.

Whizard and Delphes: User Job

- ▶ Obtain a certificate and register on iLCDirac
- ▶ Set up Dirac environment and iLCDirac client
- ▶ Create and run your job:

```
12 def run():
13     """Run The Job."""
14     job = UserJob()
15     job.setConfigPackage("fccConfig",
16     ↪ 'key4hep-devel-2')
17
18     whiz = Whizard2()
19     whiz.setVersion('2.8.3')
20     whiz.setNumberOfEvents(100)
21     whiz.setSinFile('ee_ZH_240gev_polp80.sin')
22     whiz.setEnergy(240)
23     whiz.setEvtType('ZH')
24     whiz.setRandomSeed(1234567824579)
25     whiz.setOutputFile('events.stdhep')
26     job.append(whiz)
27
28     delphes = DelphesApp()
29     delphes.setVersion('key4hep-latest')
30
31     ↪ delphes.setExecutableName('DelphesSTDHEP_EDM4HEP')
32     # delphes.setInputFile('./events.stdhep')
33     delphes.getInputFromApp(whiz)
34     delphes.setDetectorCard('card_IDEA.tcl')
35     delphes.setOutputCard('edm4hep_IDEA.tcl')
36     delphes.setEnergy(240)
37     delphes.setOutputFile('output.root')
38     job.append(delphes)
39
40     job.submit(DiracILC(), mode='local')
```

- ▶ Examples of some possible User Jobs can be found on the [iLCDirac User Guide](#).



Production jobs

Whizard and Delphes: Production Job

```
1  [whizard2]
2  Version = 2.8.3
3  EvtType = ZH
4
5  [delphesapp]
6  ExecutableName = DelphesSTDHEP_EDM4HEP
7  DetectorCard = card_IDEA.tcl
8  OutputCard = edm4hep_IDEA.tcl
9  Version = key4hep_230408
10
11 [Production Parameters]
12 machine = ee
13 prodGroup = several
14
15 softwareVersion = key4hep_230408
16 generatorApplication = whizard2
17 generatorSteeringFile = ee_ZH_240gev_polp80.sin
18 processingAfterGen = delphesapp
19
20 configVersion = key4hep-devel-2
21 configPackage = fccConfig
22 eventsPerJobs = 100
23
24 numberOfTasks = 1
25
26 campaign = winter2023
27 energies = 240
28 processes = ZH
29 detectorModel = idea
30
31 productionLogLevel = VERBOSE
32 outputSE = CERN-DST-EOS
33
34 finalOutputSE = CERN-SRM
35 MoveStatus = Stopped
36 MoveGroupSize = 10
37
38 ProdTypes = Gen
```

How to create the transformation

- ▶ Obtain a certificate and register on iLCDirac
- ▶ Set up Dirac environment and iLCDirac client
- ▶ Creating the transformation is as simple as:
 - ▶ `source /cvmfs/clicdp.cern.ch/DIRAC/bashrc # pre-prod instance`
 - ▶ `dirac-proxy-init -g fcc_prod # for the privileged`
 - ▶ `dirac-fcc-make-productions -p > configFile # modify`
 - ▶ `dirac-fcc-make-productions -f configFile # dry run - launching the production`
 - ▶ `dirac-fcc-make-productions -f -x configFile # actual production on the grid`
- ▶ Script reads 'conf' file, sets workflow module parameters ("physics and software"), creates chain of given *ProdTypes*, for each *process* or *prodID*
- ▶ Metadata used to define input data (*EventType*, *ProdID*) and to aggregate information about output folders

Outputs, Metadata

Output path:

```
/fcc/ee/<season-year>/<energy>/<evt-type>/<detector-name&version>/<datatype>/<transf-ID>
```

Metadata (in iLCDirac):

```
'Energy', 'EvtType', 'DetectorType', 'Datatype', 'NumberOfEvents', 'ProdID', 'SWPackages'
```

Output Structure, Metadata

Metadata:

Registered metadata:

```
/fcc/ee/winter2023/240gev/: {'Energy': '240'}  
/fcc/ee/winter2023/240gev/ZH/: {'EvtType': 'ZH'}  
/fcc/ee/winter2023/240gev/ZH/idea: {'DetectorType': 'idea'}  
/fcc/ee/winter2023/240gev/ZH/idea/delphes: {'Datatype': 'delphes'}  
/fcc/ee/winter2023/240gev/ZH/idea/delphes/00012345:  
  {'ProdID': 12345, 'NumberOfEvents': 1000}  
/fcc/ee/winter2023/240gev/ZH/stdhep: {'Datatype': 'stdhep'}  
/fcc/ee/winter2023/240gev/ZH/stdhep/00012345:  
  {'ProdID': 12345, 'NumberOfEvents': 1000}
```

Registered non searchable metadata:

```
/fcc/ee/winter2023/240gev/ZH/idea/delphes/00012345 =  
  {'SWPackages': 'whizard2.2.8.3;delphesapp.key4hep_230408'}  
/fcc/ee/winter2023/240gev/ZH/stdhep/00012345 =  
  {'SWPackages': 'whizard2.2.8.3;delphesapp.key4hep_230408'}
```

Outputs:

ZH_stdhep.stdhep, ZH_delphes.root

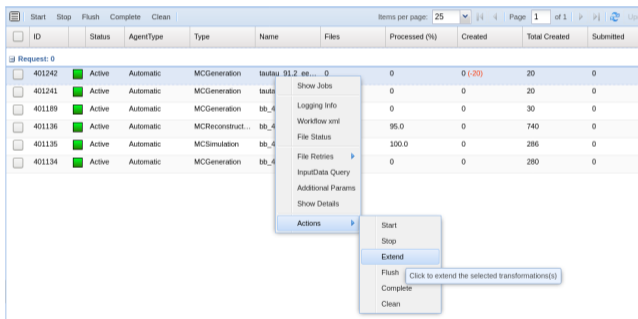
Detailed metadata

- ▶ New agent for generation of downloadable metadata in json format
- ▶ More detailed information about the physics of the process, and the outputs of the production
- ▶ Updated periodically for all the active productions

```
"12345": {  
  "Status": "Active",  
  "Version": 0,  
  "cross-section": "...",  
  "cross-section-error": "...",  
  ...  
  "production-manager": "Lorenzo Valentini",  
  "path": {  
    "1124663": "/fcc/ee/winter2023/240gev/  
              /ZH/idea/delphes/00012345",  
    "1234663": "/fcc/ee/winter2023/240gev/  
              /ZH/stdhep/00012345"  
  }  
}
```

Increasing the size of a production

`https://voilcdiracwebapp3.cern.ch/DIRAC/s:ILC-Development/g:
fcc_prod/?theme=Crisp&url_state=1|*DIRAC.TransformationMonitor.classes.
TransformationMonitor:`



ID	Status	AgentType	Type	Name	Files	Processed (%)	Created	Total Created	Submitted
401242	Active	Automatic	MCGeneration	tautau_91.2_ee..._0		0	0 (:20)	20	0
401241	Active	Automatic	MCGeneration	tautau_91.2_ee..._0		0	0	20	0
401189	Active	Automatic	MCGeneration	bb_4		0	0	30	0
401136	Active	Automatic	MCRReconstruct...	bb_4		95.0	0	740	0
401135	Active	Automatic	MCSimulation	bb_4		100.0	0	286	0
401134	Active	Automatic	MCGeneration	bb_4		0	0	280	0

or

`dirac-ilc-add-tasks-to-prod.py 12345 500`

Find the files, production info

```
dirac-dms-find-lfns Path=/ ProdID=12345 Datatype=delphes
dirac-dms-find-lfns Path=/ EvtType=ZH Energy=240 Datatype=delphes \
    DetectorType=idea
dirac-ilc-get-info -p 12345
dirac-ilc-get-prod-log -P 12345
```

DataManipulation Transformations

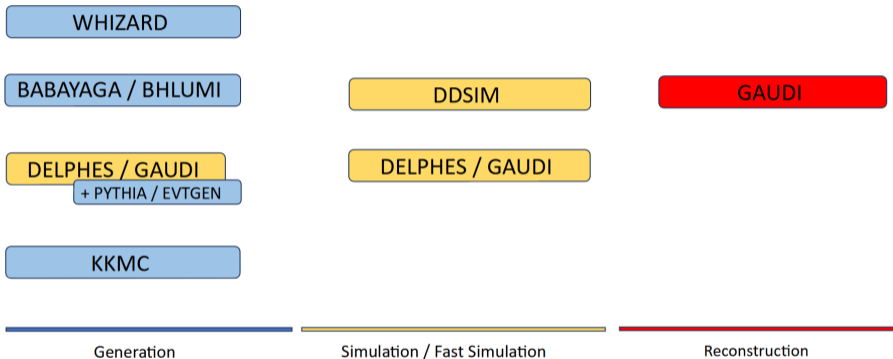
- ▶ scripts to define replication (staging) or moving transformations
 - ▶ Replicate files from A to B, optionally remove from A
 - ▶ Uses FTS3

Set ProdID, SourceSE, TargetSE, Datatype; optional: GroupSize

[dirac-ilc-replication-transformation](#)

[dirac-ilc-moving-transformation](#)

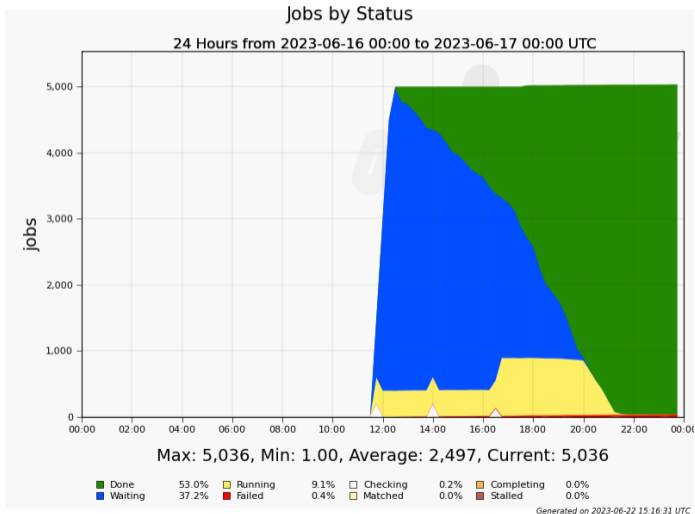
New FCC workflows in iLCDirac



- Examples of all the possible workflows can be found on the [iLCDirac Production Manager Guide](#).

Large Scale Fast Simulation with Delphes and Pythia

- ▶ 5000 jobs for a total of 500M events
- ▶ All the (few) failed jobs successfully resubmitted until the required number of events was reached
- ▶ Entire process completed in less than half a day





Documentation

Documentation

- ▶ <http://lcd-data.web.cern.ch/lcd-data/doc/ilcdircdoc/>
- ▶ Information about commands (scripts) including options
- ▶ API, examples for all applications

ILCDIRAC v25r0p7 documentation » next | modules | index

ILCDIRAC Documentation

Welcome to the ILCDIRAC Documentation.

Interfaces for User Jobs

If you are looking for how to submit jobs for Linear Collider Software please look at the `UserJob` class and the `Applications` modules and finally at the `DiracILC` class

- Applications
- UserJob
- DiracILC

Scripts

Scripts of interest to the casual user are part of the `Interfaces` module

- Interfaces Scripts
 - `dirac-ilc-find-in-FC`
 - `dirac-ilc-show-software`
 - `dirac-repo-create-lfn-list`
 - `dirac-repo-retrieve-jobs-output`
 - `dirac-repo-retrieve-jobs-output-data`
 - `ilcdirc-version`

Support

► In case of fire:

1. Consult documentation:

<http://lcd-data.web.cern.ch/lcd-data/doc/ilcdiracdoc/>

2. Before submitting a ticket, see: <http://lcd-data.web.cern.ch/lcd-data/doc/ilcdiracdoc/DOC/Files/UserGuide/support.html>

3. Submit a ticket to the issue tracker

<https://its.cern.ch/jira/browse/ILCDIRAC>

- See also “Report a Problem” buttons in web portal and documentation

4. Email: ilcdirac-support@cern.ch



The logo for Eurizon, featuring the word "eurizon" in a bold, blue, lowercase sans-serif font. A small yellow dot is positioned above the letter 'i'. Below the main text, the tagline "European network for developing new horizons for RIs" is written in a smaller, blue, sans-serif font.

eurizon
European network
for developing new horizons for RIs



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871072



Thanks for the attention!