



iLCDirac: Status and Plans

André Sailer

CERN-EP-SFT

Linear Collider Workshop
March 15, 2021

Table of Contents



1 Introduction

2 Usage

3 Future Plans

4 Documentation and Support

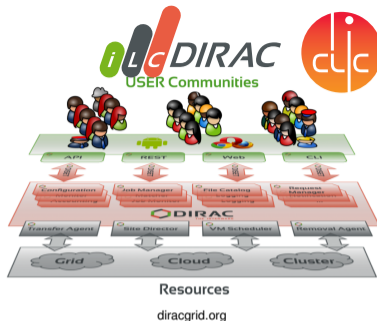
5 Summary

iLCDirac in a Nutshell



iLCDirac is based on the DIRAC interware originally developed for LHCb

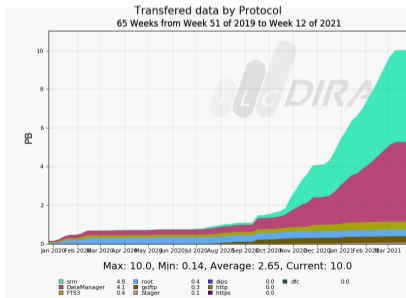
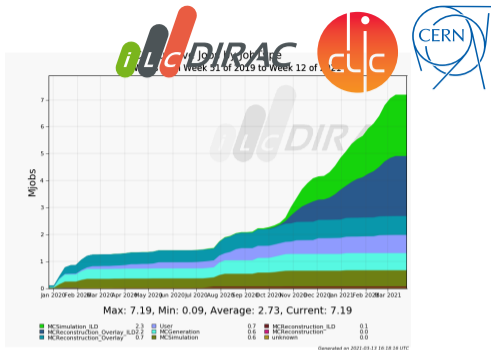
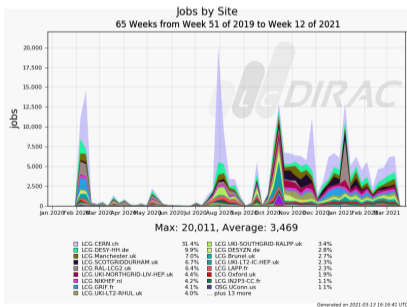
- Dirac (Distributed Infrastructure with Remote Agent Control): High level interface between users and distributed resources
- iLCDirac: Additional functionality to provide simple interface for the users to the LC Software (Whizard, Whizard2, Marlin, Mokka, org.lcsim, SLIC, ROOT, ddsim)
- Central system for large scale productions



```
1 from DIRAC.Core.Base import Script
2 Script.parseCommandLine()
3 import UserJob, Marlin, DiracILC
4 d = DiracILC(); j = UserJob()
5 j.setInputData("/ilc/user/i/initial/simEvents.slcio")
6 j.setOutputData("recEvents.slcio", SE='FOO-BAR')
7 m = Marlin()
8 m.setVersion("ILCSOft-01-17-09")
9 m.setSteeringFile("Steering.xml")
10 m.setNumberOfEvents(100)
11 m.setInputFile("SimEvents.slcio")
12 m.setOutputFile("recEvents.slcio")
13 j.append(m); j.submit(d)
```

Usage

- In last months sustained about 5k jobs, 13k peak
- More than 7 Million jobs in last 5 quarters
- 10 PB of files copied around



Developments in iLCDirac



- Overlay: selection of closest StorageElement (SE), calculated from site coordinates
 - Better efficiency for reconstruction jobs with background Overlay
- More options for job splitting (see examples in next slides)
- More options for configuring Whizard2: basically allowing raw Sindarin files
- New application interface KKMC (thanks to A. Stano, G. Ganis (CERN))
 - cf. [KKMC presentation at this Workshop](#)
- Keeping up with changing grid resources
- Including the FCC Virtual Organization: separate users, resources, common software use
 - iLCDirac servers have a lot of load to spare, or can scale up services
- iLCDirac servers updated to CentOS7

Job Splitting Example I



```
1 from DIRAC.Resources.Catalog.FileCatalog import FileCatalog
2 import DiracILC, UserJob, Marlin
3
4 dIlc = DiracILC()
5
6 inputData = FileCatalog().findFilesByMetadata({'ProdID': 9275, 'DataType': 'SIM'})
7 inputData = inputData['Value'] # assuming success
8
9 job = UserJob()
10 job.setName("SplitTest_%n") # %n will be replaced by the task number
11 job.setCLICConfig('ILCSoft-2020-02-07')
12 job.setOutputData("RecoTest.slcio", OutputPath="RecoTest")
13 job.setSplitInputData(inputData, numberOfFilesPerJob=10)
14
15 marl = Marlin
16 marl.setVersion('ILCSoft-2020-02-07')
17 marl.setSteeringFile('cllcReconstruction.xml')
18 marl.setOutputFile('RecoTest.slcio')
19
20 job.append(ga); job.submit(dIlc)
```

Line 13 is the key, *inputFile* will be automatically filled for the application. Faster job submission.

Job Splitting Example II



```
1  outputFiles = ['electron.slcio', 'muon.slcio', 'pion.slcio']
2  particles = ['e-', 'mu-', 'pi-']
3  job = UserJob()
4  job.setName('DDSimTest_%n')
5  job.setSplitDoNotAlterOutputFilename()
6  job.setSplitParameter('particle', particles)
7  job.setSplitParameter('outputFile', outputFiles)
8  job.setSplitOutputData(outputFiles, 'test/ddsim', 'CERN-DST-EOS')
9
10 ddsim = DDSim()
11 ddsim.setVersion('ILCSoft-2018-08-10_gcc62')
12 ddsim.setDetectorModel('CLIC_o3_v14')
13
14 # the named placeholder '%(particle)s' has the same name as the first argument of setSplitParameter
15 ddsim.setExtraCLIArguments('--gun.particle=%(particle)s')
16 ddsim.setOutputFile('%(outputFile)s.slcio')
```

Should work for any string based application parameter, let me know your use case.

Future Plans



Update to DIRAC v7r1 next week

- New way of distributing external libraries python, necessary libraries for CE and SE access: DiracOS
- Support for new 'CERN Tape Archive' CTA, replacing CERN Castor in April
- Pilot 3: possibility for more flexible job installations

iLCDirac developments:

- Integrate Key4hep Application(s)
 - ▶ Synergy with FCC
 - ▶ Possible synergy for Workflow Modules in Key4hep



- Python 3 compatibility of DIRAC is greatly advancing.
 - Simple compatibility changes in iLCDirac already done
 - Existing tests will greatly help validation
- Together with a new way of installation the DIRAC client for Python 3
 - `pip install ILCDIRAC`
- Will have compatibility for Python 3 server with Python 2 client (or more likely the other way around) for a while, but eventually all user code has to move to Python 3
- Towards the end of the year

Documentation

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

iLCDIRAC CERN

ILCDIRAC v2of0p/ documentation » modules | index

Table Of Contents

- ILCDIRAC Documentation
 - Interfaces for User Jobs
 - Scripts
 - Production Scripts
 - Contents
 - Release Notes
 - Indices and tables

Next topic

Applications

This Page

Show Source

Quick search

Enter search terms or a module, class or function name.

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-llc-find-in-FC
 - dirac-llc-show-software
 - dirac-repo-create-lfn-list
 - dirac-repo-retrieve-jobs-output
 - dirac-repo-retrieve-jobs-output-data
 - ilcdirac-version



■ 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



Summary



- iLcDirac continues to be highly used in the Linear Collider Community
- Small improvements continue to be made in iLcDirac, integrating Key4hep applications
- Work in the background to keep resources available and servers up and running and users happy