

ILCDirac Production

C. Calancha (KEK)
Tokusui Workshop 2015

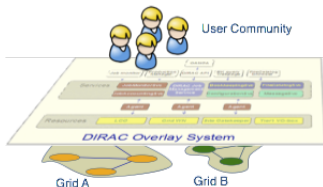
December 9th, 2015

DIRAC & ILCDIRAC

- DIRAC project is a complete Grid solution.
 - DIRAC forms a layer between a particular community and various compute resources.
- ILCDIRAC forms a layer on top of DIRAC providing the computing needs of linear collider experiments.

ILD Mass productions with ILCDIRAC

- ILD has adopted `ilcdircac` for its mass productions.
 - Reliable, scalable, easy to use, good experts support.
 - Other experiments already using it: CLIC, BelleII, Sid.



DIRAC: A software framework for distributed computing

- High level interface between users and distributed resources.
- Adopted by more than 20 virtual organisations (VO).
- Developed mostly by LHCb plus support from a wider community.
- Written in python 2 (225k lines of code).
- Has multiple extensions, e.g. Web(App)DIRAC,...
- Rapid release cycle, 2 or 3 releases per year.



(*) Distributed Infrastructure with Remote Agent Control

- ILCDIRAC is an extension of the dirac system for the ILC VO.
- ILC VO: virtual organisation for linear collider detectors (SiD, ILD and CLICdp).
- Code comprised of 33k lines.
- Centralized MC Production (Event Generation, Geant4 Simulation, Reconstruction).
- User jobs (Generation, Simulation, Reconstruction, Analysis).
- Each release uniquely dependent from a DIRAC release.
- Mostly following DIRAC release cycle.



One Year of Work in a Nutshell

- 0 Validation of the samples produced w/ `ilcdirac` ✓
 - Compared with samples produced as in the DBD.
- 1 Extensive test of the ILD modules running the ILD production chain. ✓
 - These modules were not used before: tested at all ILC energies.
 - Improved readability and submission speed.
- 2 Successfully **run first official ILD mass production** requests using `ilcdirac` ! ✓
- 3 Extending/improving the tools for monitoring the production jobs. ✓
 - Often, I get new ideas for possible improvements(*).

(*) Good response from DIRAC developers to improvement requests.

Testing ILD modules

- Output file name follows ILD name conventions.
- Clean up the code: improved legibility; more compact.
- Improved speed submission significantly.

```
class ILDProductionJob(ProductionJob):
    """ILD Production Jobs definition"""
    def __init__( self ):
        super(ILDProductionJob, self).__init__()
        self.machine = 'ilc'
        self.experiment = 'ILC_ILD'
        self.basepath = self.ops.getValue( '/Production/%s/BasePath' % self.experiment, '/ilc/prod/ilc/mc-dbd/ild/' )
        self.polarization = ""
        self.machineparams = ""
        self.detector = ""
        self.compatmeta = {}
        self.processID = 0
        self.evtclass = ""
        self.evttype = ""
        self.genprocname = ""
        self.usesofttag = False

overlay.setBkgEvType( 'e+e-1loop' )
if energy==500.: #here you chose the overlay parameters as this determines how many files you need
#it does NOT affect the content of the marlin steering file whatsoever, you need to make sure the values
#there are correct. Only the file names are handled properly so that you don't need to care
overlay.setBXOverlay(BXOverlay)
overlay.setGGToHadInt(GGToHadInt500)
elif energy == 1000.:
    overlay.setBXOverlay(BXOverlay)
    overlay.setGGToHadInt(GGToHadInt1000)
elif energy == 350.:
    overlay.setBXOverlay(BXOverlay)
    overlay.setGGToHadInt(GGToHadInt350)
elif energy == 250.:
    overlay.setBXOverlay(BXOverlay)
    overlay.setGGToHadInt(GGToHadInt250)
else:
    print "Overlay ILD: No overlay parameters defined for this energy"

##Reconstruction ILD with overlay
mso = Marlin()
mso.setDebug()
mso.setVersion(MarlinVer) ##PUT HERE YOUR MARLIN VERSION
if ild_rec_ov:
    if energy in [250.0, 350.0, 500.0, 1000.0]:
        mso.setSteeringFile("bbudsc_3evt_stdreco.xml")
        mso.setGearFile("GearOutput.xml")
    else:
        print "Marlin: No reconstruction suitable for this energy"
```

Run First official Requests

- **ILD is now** running its mass productions **using ILCDIRAC**.
 - Currently samples as DBD: ilcsoft. v01-16 simulated with with ILD_o1_v05
 - Newer ilcsoft versions/ILD detector models will be added when needed by users.
 - Module to simulate/reconstruct based on DD4hep under development.

The screenshot displays the ILCDIRAC Production Monitor interface. On the left, there are several filter and control panels: 'Status' (Active, Stopped), 'Agent Type' (All), 'Type' (All), 'Group' (ILD-DBD_higgs_fm_250.0), 'Plugin' (All), 'Date' (YYYY-mm-dd), 'ProductionID', and 'RequestID'. Below these are 'Submit' and 'Reset' buttons. The main area is a table with columns: ID, Status, Type, Files, Processed (%), Total Created, Done, and Failed. The table is divided into two groups: 'Group: ILD-DBD_higgs_fm_250.0' and 'Group: ILD-DBD_higgs_fm_350.0'. Each row represents a production request with its ID, status (Active), type (MCReconstruct... or MCSimulation), file count, and progress. At the bottom, there are navigation controls (Page 1 of 1), a Refresh button, and system information including the user 'calanchac@ ilc_prod' and the timestamp 'Updated: 2015-11-01 03:50 [UTC]'.

ID	Status	Type	Files	Processed (%)	Total Created	Done	Failed
Group: ILD-DBD_higgs_fm_250.0							
6045	Active	MCReconstruct...	100	100.0	120	100	20
6044	Active	MCReconstruct...	100	100.0	112	100	12
6041	Active	MCReconstruct...	100	100.0	116	100	16
6039	Active	MCReconstruct...	100	100.0	101	100	1
6037	Active	MCReconstruct...	100	100.0	117	100	17
6035	Active	MCReconstruct...	100	100.0	102	100	2
6043	Active	MCSimulation	100	100.0	104	100	4
6042	Active	MCSimulation	100	100.0	134	100	34
6040	Active	MCSimulation	100	100.0	109	100	9
6038	Active	MCSimulation	100	100.0	105	100	5
6036	Active	MCSimulation	100	100.0	114	100	14
6034	Active	MCSimulation	100	100.0	104	100	4
Group: ILD-DBD_higgs_fm_350.0							
6055	Active	MCReconstruct...	100	100.0	103	100	3
6054	Active	MCReconstruct...	100	100.0	101	100	1
6052	Active	MCReconstruct...	100	100.0	118	100	18
6050	Active	MCReconstruct...	100	100.0	129	100	29
6047	Active	MCReconstruct...	100	100.0	123	100	23
6045	Active	MCReconstruct...	100	100.0	114	100	14

How to do a Production Request

Send me an e-mail with following information calancha@post.kek.jp (*):

- Physics process.
- Do you need simulate? Reconstruct? Both things?
- ilcsoft version, detector model.
- Location of the generated *stdhep* files.
- How many events from those *stdhep* files you need to simulate?

Request accepted or rejected based on

- Current ILD production needs.
- Current Disk space resources.
- Waiting list of production requests.
- If the request is rejected users still can produce samples on their own using ILCDIRAC.

- Once the request is accepted the production typically start in that same day.
- When the ilcsoft/detector model are already within ILCDIRAC configuration system the request may be ready \approx 1 week(**).

(*) Until end of present japanese fiscal year

(**) Depending of the process, #events and #jobs running in the system at that time.

Significantly improving production monitoring

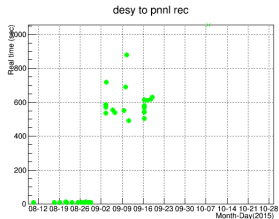
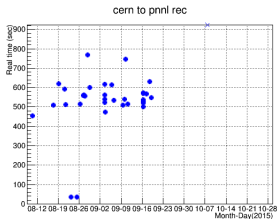
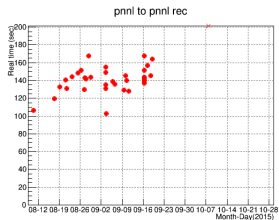
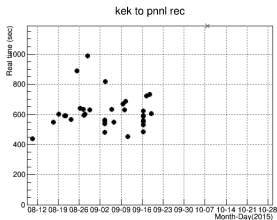
- Extended command line interfaces: transformations, dirac file catalog.
 - Added more functions retrieving real time information.
 - List productions by *real user*(*) now available.
 - Faster responses: Replaced several 1 string argument functions with equivalent accepting lists(**).

```
(Cmd) getALLbyuser calancha Stopped
TransformationID      Status      AgentType      TransformationName      Creat\
ionDate              AuthorDN
5893                 Stopped    Manual         37588_1000.0_split_test-1000gev_20151007_1      2015-\
10-07 12:46:54      /C=JP/O=KEK/OU=CRC/CN=CALANCHA Constantino
5894                 Stopped    Manual         37588_1000.0_ild_sim_test-1000gev_20151007_1      2015-\
10-07 12:47:44      /C=JP/O=KEK/OU=CRC/CN=CALANCHA Constantino
5895                 Stopped    Manual         37588_1000.0_ild_rec_overlay_test-1000gev_20151007_1      2015-\
10-07 12:48:32      /C=JP/O=KEK/OU=CRC/CN=CALANCHA Constantino
6032                 Stopped    Manual         108709_350.0_ild_sim_calancha_20151021_2      2015-\
10-21 00:37:38      /C=JP/O=KEK/OU=CRC/CN=CALANCHA Constantino
6033                 Stopped    Manual         108709_350.0_ild_rec_overlay_calancha_20151021_2      2015-\
10-21 00:38:28      /C=JP/O=KEK/OU=CRC/CN=CALANCHA Constantino
6034                 Stopped    Manual         108714_250.0_ild_sim_calancha_" + genprocessname + "__20151021_2      2015-\
10-21 02:14:25      /C=JP/O=KEK/OU=CRC/CN=CALANCHA Constantino
6035                 Stopped    Manual         108714_250.0_ild_rec_overlay_calancha_" + genprocessname + "__20151021_2      2015-\
10-21 02:15:14      /C=JP/O=KEK/OU=CRC/CN=CALANCHA Constantino
6036                 Stopped    Manual         108716_250.0_ild_sim_calancha_" + genprocessname + "__20151021_2      2015-\
10-21 02:19:31      /C=JP/O=KEK/OU=CRC/CN=CALANCHA Constantino
6037                 Stopped    Manual         108716_250.0_ild_rec_overlay_calancha_" + genprocessname + "__20151021_2      2015-\
10-21 02:20:19      /C=JP/O=KEK/OU=CRC/CN=CALANCHA Constantino
(Cmd)
```

(*) User creating the production. That is typically different than the *owner*, which correspond to the *ilcdircac* leader.

(**) `def myfunc(strgArg):` → `def myfunc(listArg):`

Other Group Activities



- Miyamoto san has developed scripts to monitoring the transfer speed between GRID sites.
- PNNL plan to collaborate with KEK to implement such tools within its framework.
- For details see Miyamoto san talk.

Summary/Prospects

ILCDIRAC

- ILCDIRAC is offering an easy interface for users to run jobs on the GRID.
- Enables centralised production of MC for the LC community (many available resources!).

ILD is Using ILCDIRAC for Mass Productions

- ILD has adopted ILCDIRAC for its mass productions
 - Several official request completed successfully using this new tool.
- Improving existing tools to monitoring the productions more effectively.

Future

- Keep producing samples under request.
- Keep improving/extending the monitoring tools.
- DD4hep interfaz currently under development.
- ILCDIRAC will play a crucial role in a future mass production campaign
 - For detector optimisation studies or to write a TDR.