



iLCDirac: Current Status

Jan Ebbing, jan.hendrik.ebbing@cern.ch

CERN

Karlsruhe Institute of Technology

CLIC Workshop 2017

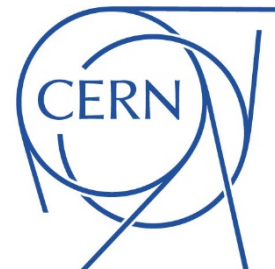
7th March 2017

Outline



- What we use iLCDirac for
- Current Status
- Recent developments:
 - Continuous Integration & Unit testing
 - Calibration system for Pandora
- Outlook
- Conclusions

What we use iLCDirac for



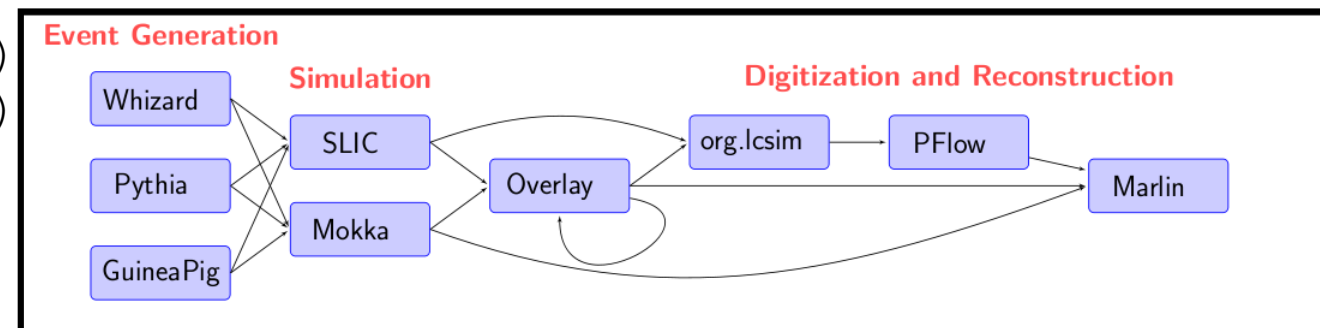
- ILC VO: virtual organisation for linear colliders, ILC & CLIC
- Extension for ILC VO, based on DIRAC by LHCb
 - Workflow modules for LC system to simplify job submission
 - Overlay System
- Entire MC production in one place: Event generation, Geant4 simulation, reconstruction
- User job submission: Generation, simulation, reconstruction, analysis

- Using WLCG and OSG resources, can serve up to 15k-20k jobs at peak

Sample job submission



```
from DIRAC.Core.Base import Script
Script.parseCommandLine()
import UserJob
import Marlin
import DiracILC
d = DiracILC()
j = UserJob()
j.setOutputSandbox("recEvents.slcio")
m = Marlin()
m.setVersion("0116")
m.setSteeringFile("Steering.xml")
m.setInputFile("SimEvents.slcio")
j.append(m)
j.submit(d)
```



A. Sailer

Support channels



- Run commands with the `-ddd` flag
- FAQ: <https://twiki.cern.ch/twiki/bin/view/CLIC/DiracForUsers>
- Collect the information described here: <http://cern.ch/go/6Lb9>
- Submit a ticket to the issue tracker
 1. <https://its.cern.ch/jira/browse/ILCDIRAC>, also has search function. (see the TWIKI for access)
 2. Email: ilcdirc-support@cern.ch
 3. forum.linearcollider.org
- Registration: ilcdirc-register@cern.ch
- Code: <https://gitlab.cern.ch/CLICdp/ILCDIRAC>
- Documentation: <http://lcd-data.web.cern.ch/lcd-data/doc/ilcdircdoc>

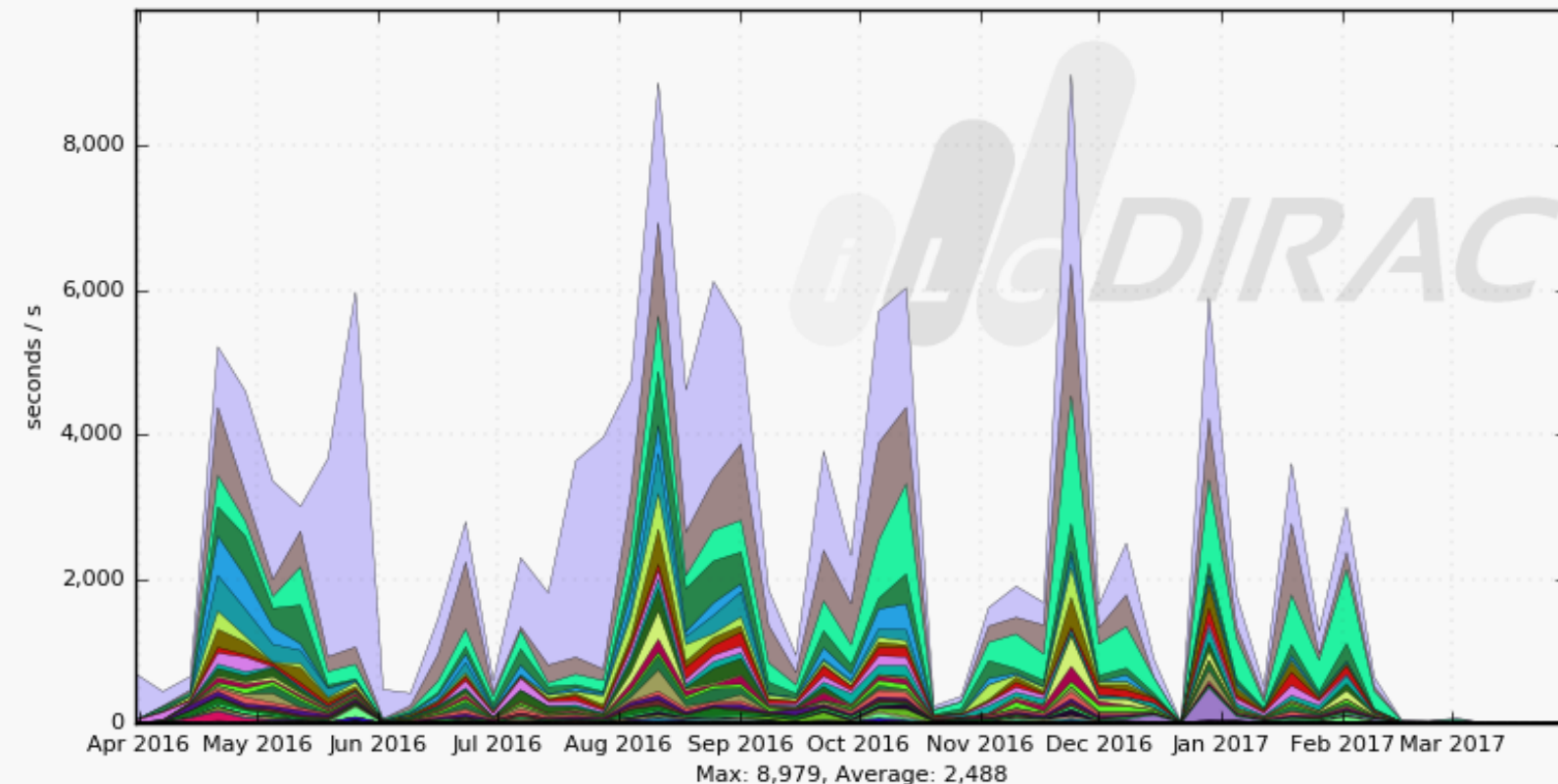
Current Status



- DIRAC v6r15 (→ v6r17) used in production, iLCDirac v26r0
- SLC6 Virtual Machines, total: 100 Cores and 200 GB of RAM
- New since 2016: Edinburgh computing element (CentOS 7)
- 2x3 Servers running Agents and Services (one set for redundancy)
- 3 DIRAC Storage elements (DIP-SE, Log-SE, SB-SE)
- Dedicated web server, DBs hosted on CERN DB on demand
- Testing and CI on 8 dedicated single-core machines

CPU usage by Site

52 Weeks from Week 13 of 2016 to Week 13 of 2017



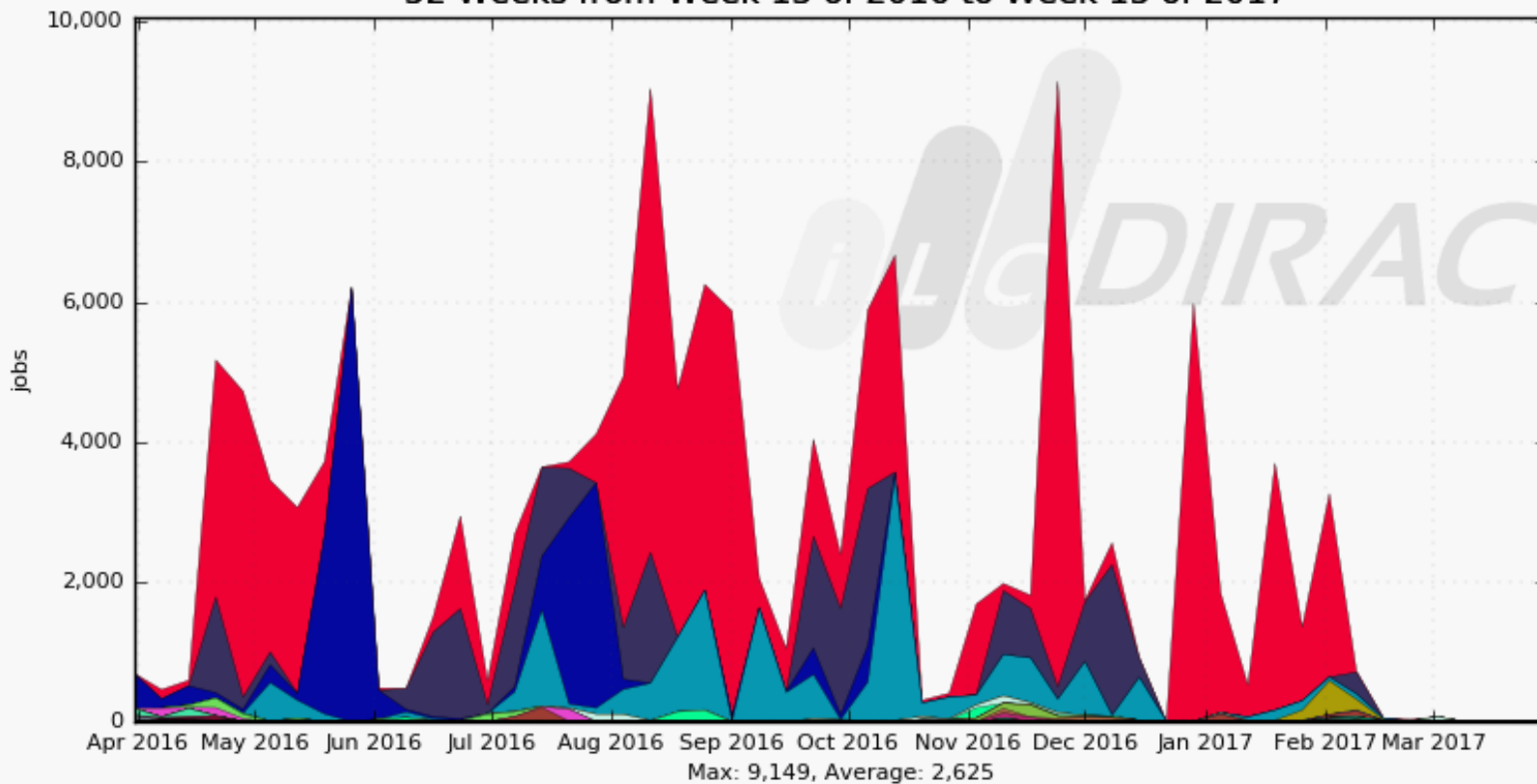
LCG.CERN.ch	36.1%	OSG.PNNL.us	1.9%
LCG.RAL-LCG2.uk	14.6%	LCG.Brunel.uk	1.7%
LCG.DESY-HH.de	12.7%	LCG.IFCA-LCG2.es	1.6%
OSG.FNAL_FERMIGRID.us	6.2%	LCG.DESYZN.de	1.4%
LCG.GRIF.fr	3.0%	LCG.UKI-NORTHGRID-LIV-HEP.uk	1.3%
LCG.UKI-LT2-IC-HEP.uk	2.8%	LCG.Cracow.pl	1.1%
LCG.UKI-SOUTHGRID-RALPP.uk	2.4%	LCG.IN2P3-CC.fr	1.0%
LCG.Manchester.uk	2.4%	LCG.UKI-LT2-RHUL.uk	1.0%
LCG.Oxford.uk	2.1%	... plus 24 more	

Generated on 2017-03-03 10:37:39 UTC



Running jobs by User

52 Weeks from Week 13 of 2016 to Week 13 of 2017



sailer	54.9%	proloff	0.3%	jstrube	0.0%	djeans	0.0%
bxu	15.6%	rstrom	0.2%	ajoffe	0.0%	hono	0.0%
estel	13.4%	twojton	0.2%	nikiforo	0.0%	milap	0.0%
sgreen	12.3%	hutran	0.2%	sopicki	0.0%	shaojun	0.0%
kacarevic	0.6%	jtingey	0.2%	amiyamot	0.0%	mschram	0.0%
webermat	0.5%	gomis	0.1%	jebbing	0.0%	blaising	0.0%
simoniel	0.5%	mperello	0.1%	igarcia	0.0%	sfernana	0.0%
feandria	0.4%	eleogran	0.1%	nurnberg	0.0%	mccoy	0.0%
tjunping	0.4%	zarnecki	0.1%	petric	0.0%	...	plus 3 more

Generated on 2017-03-03 10:47:34 UTC



Continuous Integration



- Procedure in software development
 1. Developer checks code in
 2. Software is automatically built and deployed on test servers
 3. Test suite is run
 4. Failure → E-mail (or similar)
- Ensures code base is always working

Continuous Integration



- Suite of unit, integration and system tests for iLCDirac and the Storage elements (runs sample jobs, store and retrieve files)
- Plethora of new unit tests implemented, coverage from 32,2% to 59%
- Immediate feedback if new bugs introduced → clean codebase, less bugs reach production
- Prevents regression of bugs
- Implemented in GitlabCI

Pandora Calibration System

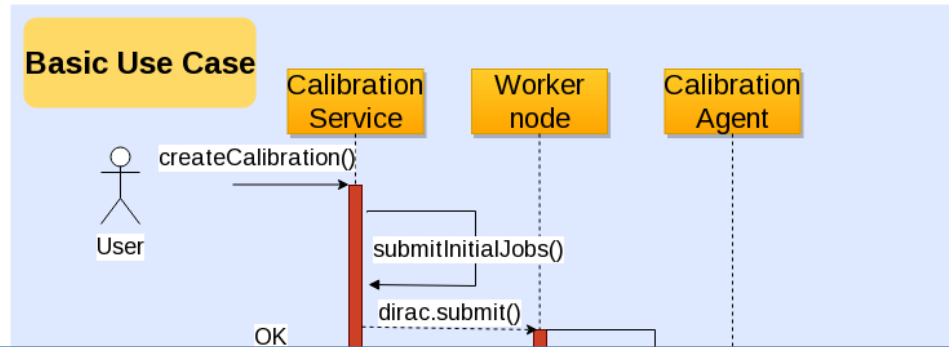


- Allows to efficiently run pandora calibration in iLCDirac
- Current solution: via HTCondor
- Includes one (small) Service and one Agent
- Sample usage:

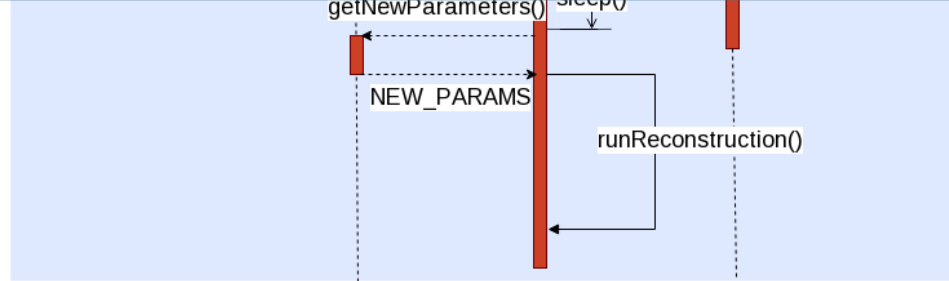
```
numberOfJobs = 100
calibrationService = RPCClient( 'Calibration/Calibration' )
result = calibrationService.createCalibration( 'steering.file', 'sw.ver',
      [ 'inputfile.1', 'if.2' ], numberOfJobs )
```

- Fully documented & under test

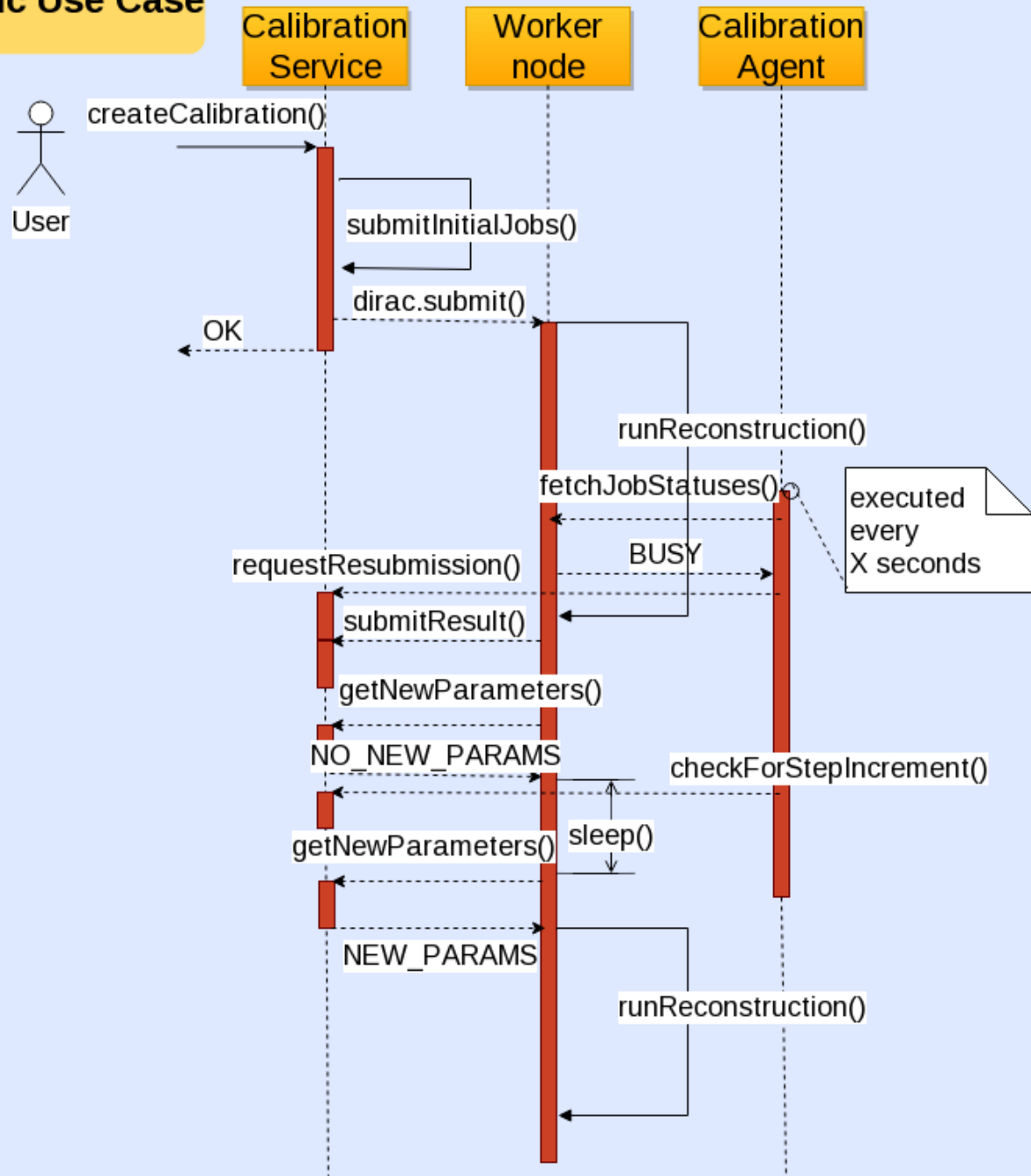
Typical use case



Note: This is simplified and not a 1:1 UML sequence diagram



Basic Use Case



Outlook



- Pandora subsystem implemented until April 2017
- Slow improvement of the iLCDirac codebase (higher test coverage, more documentation) as part of new developments
- More supported LC specific applications
- Prepare for productions in 2017

Conclusions



- Users generally happy with current setup
- Some internal developments to make maintenance easier
- Slowly integrate remaining workflows into iLCDirac