



DIRAC FOR THE LINEAR COLLIDER

8ème Rencontres de utilisateurs de DIRAC

Marko Petrič



On behalf of the **CLICdp** collaboration

Lyon, 23 May 2017

Contents

iLCDirac Use Case

Testing and Documentation for iLCDirac

Developments

Room for Improvements

Conclusions

iLCDirac Use Case

- ▶ ILC VO: virtual organization for linear colliders (CLIC and ILC)
- ▶ iLCDirac is an extension of the DIRAC system for the ILC VO
 - ▶ Workflow Modules for LC Software, Overlay System
 - ▶ JPCS. ILCDirac, a DIRAC extension for the Linear Collider community. Proceedings of CHEP2013. 513 [CLICdp-Conf-2013-003](#)
 - ▶ JPCS. Using OSG Computing Resources with (iLC)DIRAC. Proceedings of CHEP2016. [CLICdp-Conf-2017-003](#)
- ▶ Centralized MC Production (Event Generation, Sim and Rec)
- ▶ User jobs (Generation, Simulation, Reconstruction, Analyses)

Capacity:

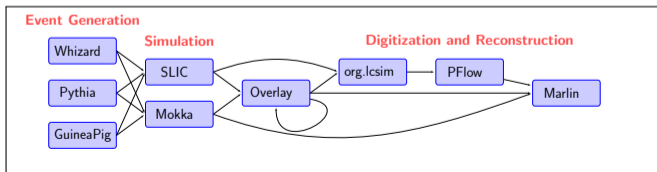
- ▶ Using WLCG and OSG resources (CREAM, Arc, HTCondorCE)
 - ▶ Mostly opportunistic, some dedicated
 - ▶ Around 15k to 20k job slots available at best of times

Code: <https://gitlab.cern.ch/CLICdp/ILCDIRAC>

API, Workflow

- ▶ Define application payload via interfaces
- ▶ Chain applications (append one after the other)

```
from DIRAC.Core.Base import Script
Script.parseCommandLine()
import UserJob
import Marlin
import DiracILC
d = DiracILC()
j = UserJob()
j.setOutputData("recEvents.slcio")
m = Marlin()
m.setVersion("ILCSoft-01-17-09")
m.setSteeringFile("Steering.xml")
m.setInputFile("SimEvents.slcio")
j.append(m)
j.submit(d)
```



Dirac Version

- ▶ DIRAC Version: v6r19p5 (servers), v6r19p7 (pilots)
- ▶ Fairly smooth transition r17→r19, testing paid off
- ▶ Started preparing for r20, already tested/using FTS3

Server Setup

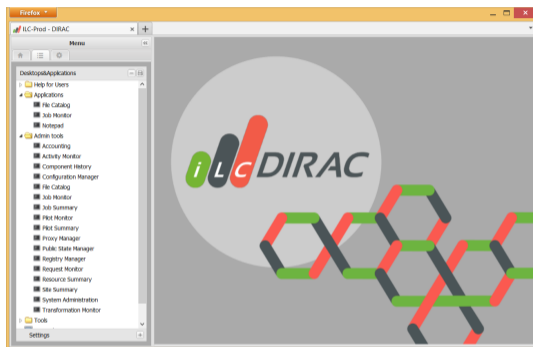
Using set of redundant servers for backup, in case primary servers go down

Total of 100 Cores and 200 GB of Ram, SLC6 Virtual Machines

- ▶ **2 × 3 Servers running Agents and Services: 8 Cores, 16 GB RAM; Split by DIRAC-System**
 1. Framework, Transformation, DataManagement, Configuration
 2. StorageManagement, WorkloadManagement
 3. RequestManagement, Accounting, ResourceStatus
- ▶ **3 DIRAC DIP-Storage SEs: 4 Cores, 8 GB Ram, 1 TB Volume**
 - ▶ DIP-SE, Log-SE, SB-SE
- ▶ **Web Server 4 Cores, 8 GB RAM**
- ▶ **DBs hosted on CERN DB on Demand (iLCDirac, ilcacdb (accounting DB), ilcdtest)**
- ▶ **Development, Testing, continuous integration (8 × 1 core), and spare VMs**

WebApp

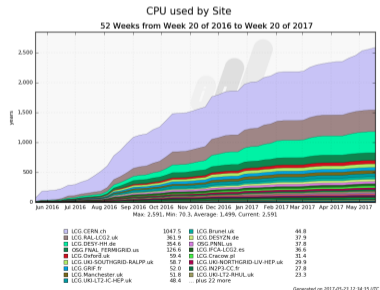
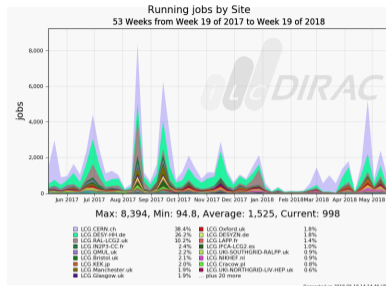
Customized version, direct forwarding to https by nginx (autologin)



- ▶ Possibility to easily add Javascript snippets (e.g., Jira Issue collector) would be appreciated
- ▶ Happy about “soon” move to Extjs 6 and new deployment
- ▶ A lot of open issues for WebApp on GitHub will be closed

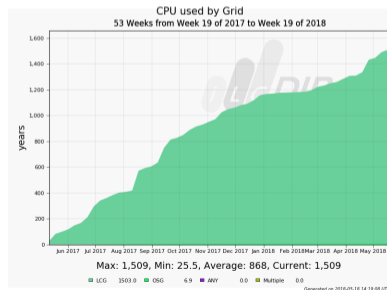
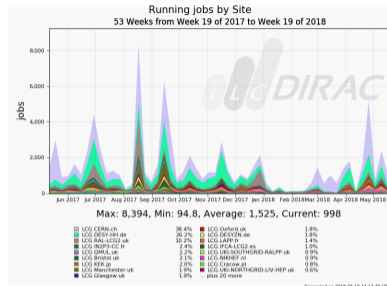
CPU Usage

- ▶ Activity in bursts
- ▶ Maximum > 10k jobs
- ▶ Integrated all OSG resources allowing ILC-VO
- ▶ Slightly less resources used in the last year compared to previous
- ▶ Expect increase for coming year due to European Strategy Update studies



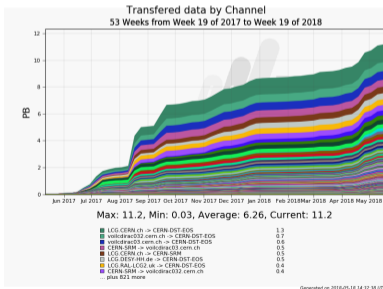
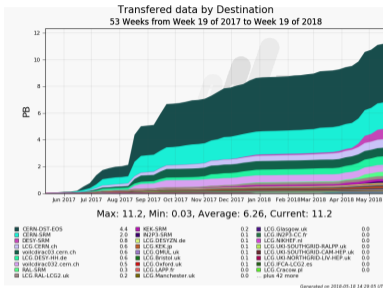
CPU Usage

- ▶ Activity in bursts
- ▶ Maximum > 10k jobs
- ▶ Integrated all OSG resources allowing ILC-VO
- ▶ Slightly less resources used in the last year compared to previous
- ▶ Expect increase for coming year due to European Strategy Update studies



DataManagement

- ▶ Using the DiracFileCatalog
- ▶ 34 Million files (41 million replicas), 7.9 PB (8.4 PB total), 7 Million files, 2 PB since the last workshop
- ▶ Metadata used to define input files for transformations
- ▶ Heavy usage of EOS → more transferred data
- ▶ All based on XRoot



Testing iLCDirac

- ▶ Running Continuous Integration for iLCDirac via GitlabCI+Travis
- ▶ Using 7 categories of tests on SLC6 and CC7
 1. **Workload tests**: try execution of job tests cases
 2. **SE tests**: try copy/add/remove between SEs (XRoot,SRM,DIP)
 3. **unit tests**: code fragments to test individual functions
 4. **pylint**: require no pylint errors (**Catches DIRAC API changes!**)
 5. **format**: require flake8 complaint commits
 6. **docs**: require sphinx successful compilation
 7. **codeclimate**: review new codeclimate warnings/errors
- ▶ Using the HEAD of DIRAC release branch (rel-v6r19)
- ▶ Installation of iLCDirac on SL6 and CC7
- ▶ Aiming for as complete coverage as possible in iLCDirac

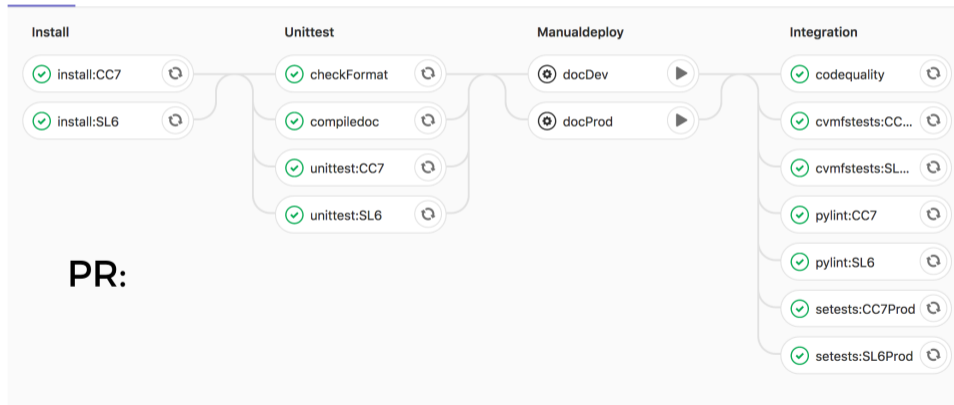
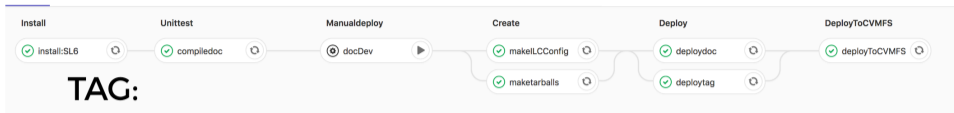
Testing iLCDirac evolution

- ▶ Extended tests since last workshop

test type	May 2017	May 2018
Workload	9	12
SE	3	5
unit	1322	1620

- ▶ Constant coverage 61% to 62%
- ▶ From 9435/16390 relevant lines to 11090/18401 relevant lines covered
- ▶ Direct deployment of client to cvmfs and tarballs and documentation to webserver via GitLab-CI

Testing iLCDirac pipelines



iLCDDirac Documentation

- ▶ Using sphinx based documentation
- ▶ Linked with DIRAC code documentation for base classes, functions,...
- ▶ Documenting the iLCDDirac API for application configuration
- ▶ Use sphinx to publish release notes
- ▶ <http://lcd-data.web.cern.ch/lcd-data/doc/ilcdiracdoc/>

iLCDDirac v28r0p5 documentation

Table Of Contents

- iLCDDirac Documentation
 - User Guide
 - Production Manager Guide
 - Code Documentation
 - Release Notes
 - Acknowledgements and References
 - Developer Guide
 - Indices and tables

Next topic

User Guide

This Page

Show Source

Quick search

Go

iLCDDirac Documentation

Welcome to the iLCDDirac Documentation.

- User Guide
- Production Manager Guide
- Code Documentation
- Release Notes
- Acknowledgements and References

User Guide

Documentation about registration, job submission, file handling can be found in the User Guide

- Registration
- Application Interfaces
- UserJob Interface
- Complete Example Submission Scripts
- Frequently Asked Questions
- Support Requests

Version v28r0p4

WorkloadManagementSystem

Feature

- (1187) JobResetAgent: automatically resets jobs that somehow got stuck in staging or completed status

FrameworkSystem

Feature

- (1181) new MonitorAgents to restart stuck agents

Workflow

Change

- (1182) UploadLogFile and UploadOutputData: Change logic to determine the experiment. ILD is using new paths: /iklprod/iklomo-opt and /iklprod/ikloms (Testing only)

Report a Problem

Important developments

- ▶ **Some developments for iLCDirac**
 - ▶ FileStatusTransformationAgent
 - ▶ Job reset agent
 - ▶ Restart Stuck Agents and Executors
 - Executors are restarted when log file is old and there are jobs in checking state
 - ▶ Automation of the production system
 - ▶ Work on reduction of operational workload
- ▶ **Some contributions to DIRAC**
 - ▶ debugging FTS3
 - ▶ GLUE2 support
 - ▶ Improvements to CI
 - ▶ Several bugfixes

Issues

- ▶ Executors are slow
 - ▶ Failed to configure multiple chains of executors
 - ▶ Installation worked, but there was always just one that treated jobs
 - ▶ Try again at the hackathon?
- ▶ Auto building of release notes?
 - ▶ Template used



andresailer commented 28 days ago

Member



```
BEGINRELEASENOTES
```

```
*Configuration
```

```
FIX: Make the GLUE2 information gathering less verbose; Silently ignore StorageShares
```

```
ENDRELEASENOTES
```

- ▶ but Release Notes don't point to PR
 - *Configuration
 - FIX: (#3671) Make the GLUE2 information gathering less...
- ▶ No correspondence between Release Note and code change

Documentation and Tests

- ▶ Improvement in documentation since last year, don't stop now
- ▶ A Roadmap spanning several future releases would be appreciated
 - ▶ Know when which feature will come, influence on roll-out...
- ▶ Testing is great, there should be more of it

Conclusions

- ▶ We are making good use of our resources via DIRAC
- ▶ Our users are generally happy, and the system is easy to use
- ▶ Life for us admins and developers could be a bit easier
 - ▶ More automation to reduce operational workload