ILCDIRAC– A grid solution for the ILC VO

Christian Grefe[1], Stéphane Poss[1], André Sailer[1], Andrei Tsaregorodtsev[2]

[1]CERN, PH-LCD, [2]CNRS

Motivation

- Scalable mass production tool for:
  - Physics performance of the detectors for the CLIC Conceptual Design report[1],
  - ILC Detailed Baseline Design report[2]
- High level interface to run linear collider applications on the GRID: make it simple for the users
- Software management: users need not to care how an application runs nor where
- Overlay of pile-up: needed for realistic physics performance studies, user interface must remain simple
- File cataloging of output files with support for meta data (not using the LCG File Catalog)

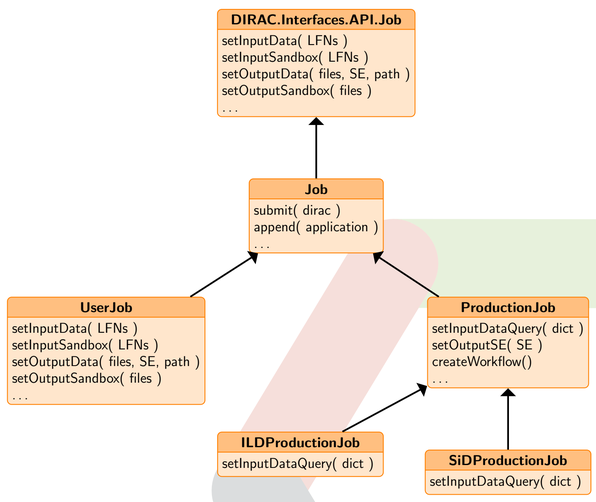
DIRAC?

- DIRAC[3] was designed for the LHCb experiment and allows extensions.
  - Proven to be scalable
  - Provides full GRID solution: Workload Management, File Catalog, Production System, etc.
- Active support in developing ILCDIRAC by the DIRAC developers
- ILCDIRAC is the second large scale extension of the DIRAC system
- The DIRAC File Catalog[4] was developed with ILCDIRAC as the first user

Job-Application Framework

Job Framework

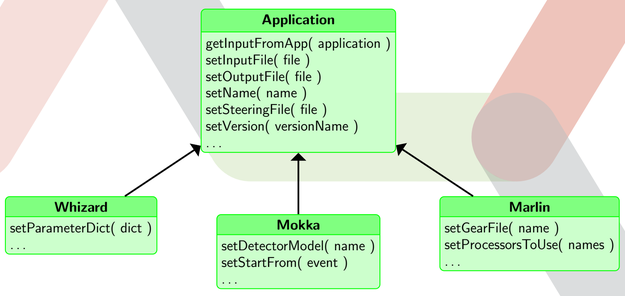
- Many different applications need to run and chained for linear collider detector studies (See Figure right)
- Software packages are interfaced through common IO format and event data model LCIO[5]
- `Job Framework` implementation based on the DIRAC `Workflow` system
- Decouple the `Job` definition from the `Application` definition
- Allow for easy extension
  - User `Jobs`: input and output provided directly by the users
  - Production `Jobs`: submission delegated to the `Transformation System`, so no input sandbox allowed
  - Derived classes take care of specialized meta data and path conventions



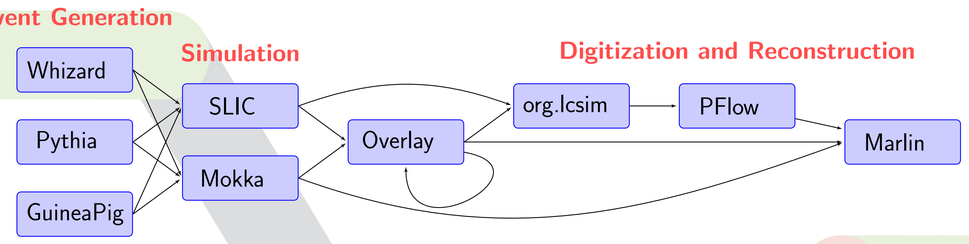
What is an `Application`?

- It converts something into something, or produces something out of nothing
- It is defined by a name and a version
- It needs instructions on what to do
- It produces a log file
- It can produce something that would be used by another application in the same job:→ linking between applications necessary
  - Output of any application can be linked to the input of any application

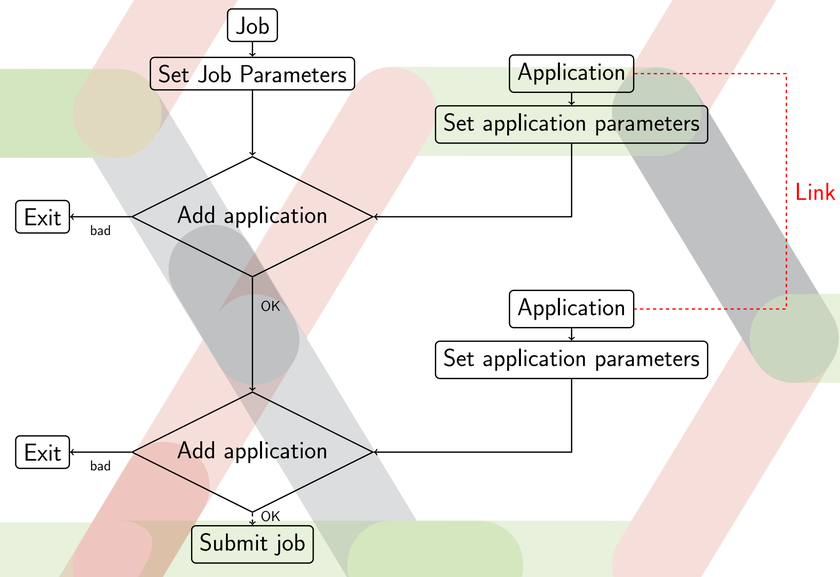
ILC use case: 14 different applications, they all need to be connectable with each other



Example of Full Simulation Software Chain

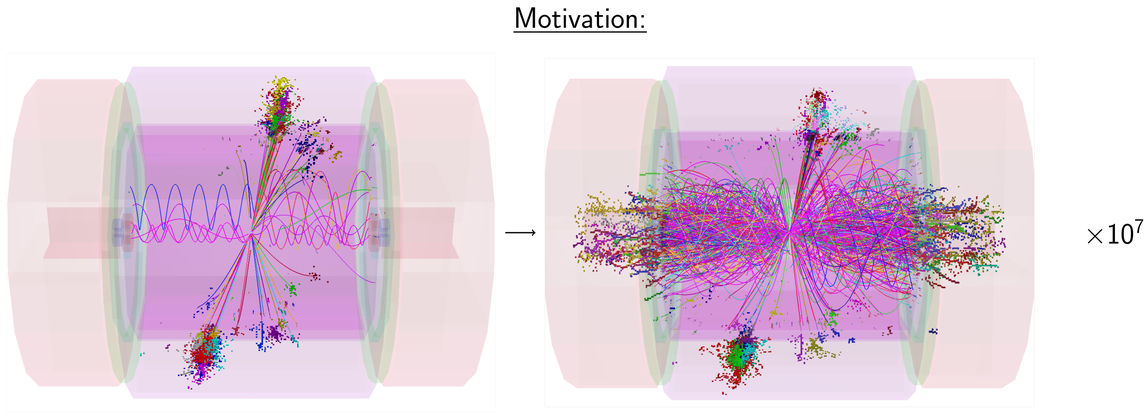


Defining a Job



Overlay System

Motivation:



$\times 10^7$

- CLIC machine induced background and physics background[6]: $\gamma\gamma \rightarrow$ hadrons
  - at CLIC 3TeV, 60×3.2 events need to be overlaid per signal event
- Too time consuming to simulate those directly
- Chosen to add them during digitization: reuse of background events
- Needed procedure to obtain the background files without blocking the storage services:
  - Background files contain 2000 events
  - Can do only 10 signal event with one file, need 20 files per signal file
  - Not many files available, so many reuses of the same files
- Fewer Storage Elements compared to Computing Elements, SEs were under heavy stress

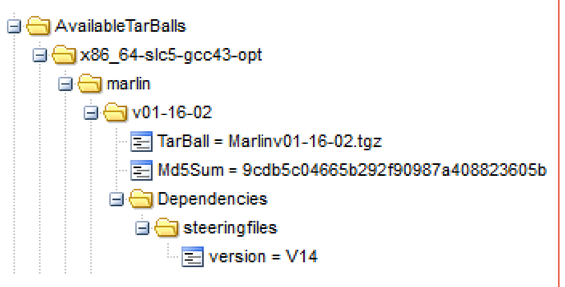
Solution:

- Local data access when possible: direct SE access for CERN, IN2P3, RAL-LCG2 using `xrdcp` or `dccp` when possible, but no SRM access.
  - All other sites use SRM access
  - Ideally: replicate the background samples to ALL SEs and use local access only
- Fine scheduling of jobs:
  - Limit the number of running jobs requiring overlay files
  - Dedicated service to count the number of jobs
- Limit the number of files to copy
  - Background events may be reused for physics events, as they are put in a random bunch crossing around the signal event

Software Management

Current status:

- Applications deployed by each job if necessary
- Use of Shared Software Area where possible
- Installation by pilot jobs in local area if needed
- Dependency relations handled
- Production steering files treated like software
- Locking of shared area to avoid conflicting installations



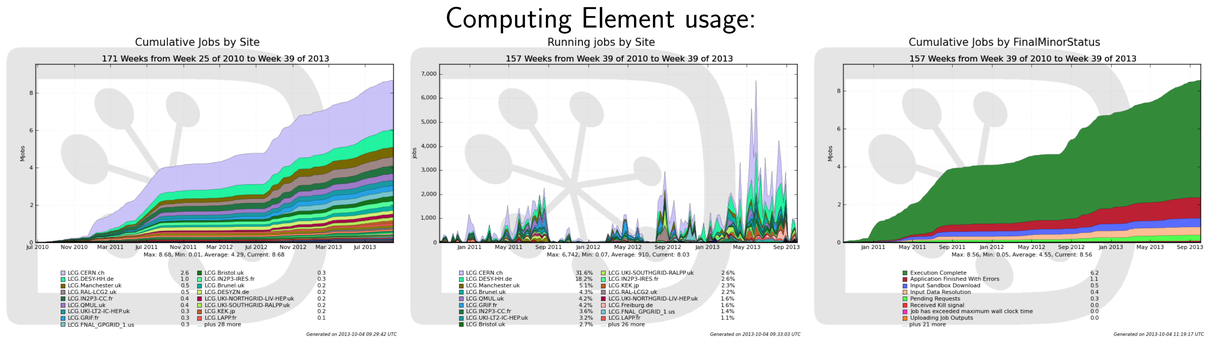
Prospects:

- Use CVMFS to have centrally managed software
- Keep possibility to install local copy if cache not up to date, or CVMFS not available

System Status

- ILCDIRAC is stable, only bug fixes and adaptation to developments of the core DIRAC packages
- Successful completion of large mass productions for the CLIC CDR and ILC DBD
- More than 100 users with new users joining every week

Computing Element usage:

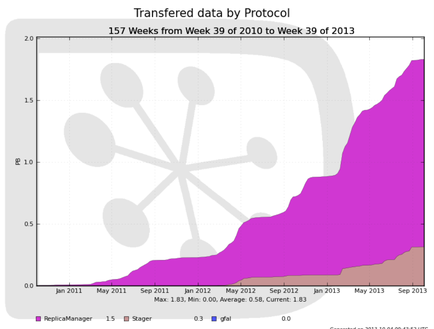


Thanks

- Site administrators for fast interactions in case of issues
- DIRAC developers for the positive discussions and rapid fixes,
- The users for the encouraging feedback

Storage Element Usage

Storage	Total	CLIC	ILC	# Files
CERN	1 PB	961 TB	40 TB	5 425 094
DESY	170 TB	0	160 TB	752 949
KEK	150 TB	76 GB	149 TB	644 365
RAL	147 TB	4 TB	89 TB	1 444 776
PNNL	25 TB	0	26 TB	739 509



Bibliography

[1] L. Linssen, A. Miyamoto, M. Stanitzki, and H. Weerts, eds.
Physics and Detectors at CLIC: CLIC Conceptual Design Report.
CERN, 2012.
ANL-HEP-TR-12-01, CERN-2012-003, DESY 12-008, KEK Report 2011-7, arXiv:1202.5940.

[2] T. Behnke, et al.
The International Linear Collider Technical Design Report - Volume 1: Executive Summary.
2013.

[3] A. Casajus, et al.
Status of the dirac project.
Journal of Physics: Conference Series, vol. 396(3) p. 032107, 2012.

[4] A. Tsaregorodtsev and S. Poss.
Dirac file replica and metadata catalog.
Journal of Physics: Conference Series, vol. 396(3) p. 032108, 2012.

[5] F. Gaede, T. Behnke, N. Graf, and T. Johnson.
LCIO: A Persistency framework for linear collider simulation studies.
In *Conference for Computing in High-Energy and Nuclear Physics*. 2003.

[6] P. Schade and A. Lucaci-Timoce.
Description of the signal and background event mixing as implemented in the Marlin processor OverlayTiming.
CERN LCD-Note-2011-006, 2011.

Mail: christian.grefe@cern.ch, spconsulting@gmail.com, andre.sailer@cern.ch, atsareg@in2p3.fr

WWW: http://ilcdirac.cern.ch