ILC DIRAC, a grid solution for the LC community

S. Poss[1] and P. Majewski[1,2]

[1]CERN, Switzerland
[2]Gdansk University of Technology, Poland

Linear Collider Software Meeting, 5 July 2010

Outline

Introduction

- DIRAC

ILCDIRAC

- Our developments

Performances

- Achievements
- Number of sites
- Storage of output data

File Catalog

Production system

Conclusion

Introduction

In 2011, CLIC community releases the Conceptual Design Report. Volume 3 describes the physics and detector studies.

This needs:

- Generation of MC events for the benchmark channels and background events
- Simulation of detector
- Reconstruction and analysis
- For both ILD and SiD geometries

Need to heavily use the GRID, not much time to start from scratch

$\Rightarrow$ DIRAC

Outline

Introduction
- DIRAC

ILCDIRAC
- Our developments

Performances
- Achievements
- Number of sites
- Storage of output data

File Catalog

Production system

Conclusion

DIRAC:
Distributed Infrastructure with Remote Agent Control

Developed as a full GRID solution for the LHCb experiment.

- System designed to manage large amount of data
- Comply with VO specific problems: heterogeneous resources, applications, etc.
- Overcome deficiencies of standard GRID middleware
- Alleviate the excessive burden from sites in supporting multiple VO
- Uses PYTHON for flexibility and cross platform

Not the only solution to solve those problems, other LHC experiments also developed their tools

More on DIRAC

- Pilot jobs: higher job efficiency than with standard jobs
  - Jobs are pulled from the central task queue
  - Multiple jobs can run in the same CPU slot (Filling mode)
- Apply VO policy directly in DIRAC, not by the site
- Security follows GRID standards

Outline

Introduction

- DIRAC

ILCDIRAC

- Our developments

Performances

- Achievements
- Number of sites
- Storage of output data

File Catalog

Production system

Conclusion

Interfacing with DIRAC core

DIRAC project already made such that adding a new “client” is easy

- LHCbDIRAC is an extension

Our work up to now:

- Use the principles of LHCbDIRAC to build ILCDIRAC
- Wrap around all LC applications to run them safely on the GRID sites
- Prepare documentation
- Set up the production system

Running LC software

Started with ILD software:

- Mokka: needs to run MySQL server
- Marlin: Environment variable resolution

Then implemented SiD software in DIRAC:

- SLIC: packaging done in SLAC, only env variables to resolve
- LCSIM: JAVA based, so no particular difficulties, apart from detector model resolution

Installing applications in local area: space in Shared Area not enough to holde several releases, and LOTS of known problems with NFS. All software packages stored on a Storage Element, accessible with grid tools: fast download, and multiple access in parallel without stability issues.

Documentation

Currently in the process of writing it:

- Extensive tutorial slides available
- Web documentation for the DIRAC API

Plans to write note with details on each available script. See https://twiki.cern.ch/twiki/bin/view/CLIC/DiracUsage for existing doc.

Outline

Introduction
- DIRAC

ILCDIRAC
- Our developments

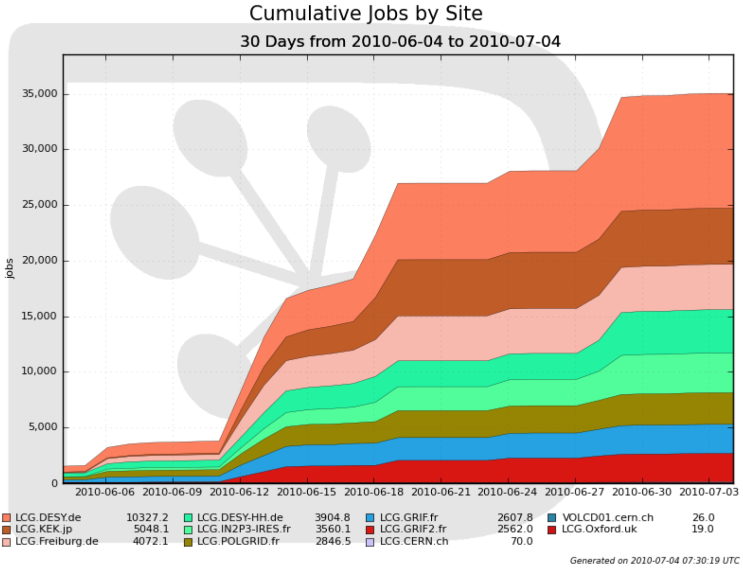
Performances
- Achievements
- Number of sites
- Storage of output data

File Catalog

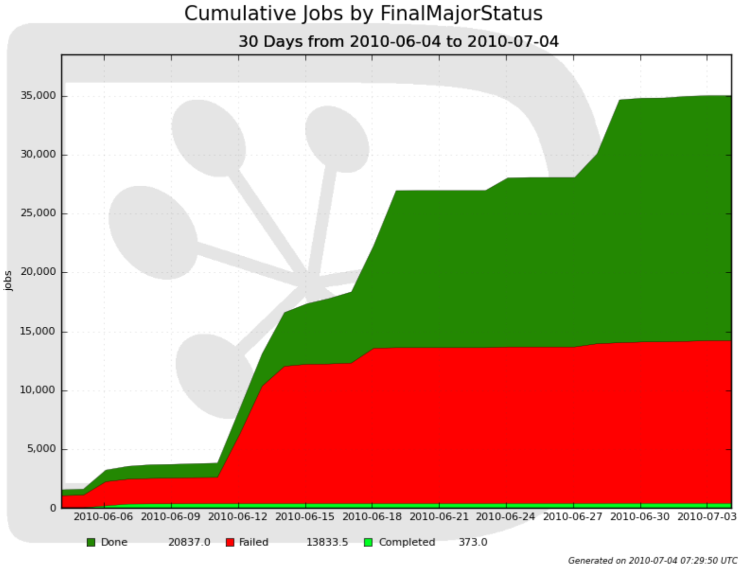
Production system

Conclusion

Number of jobs in the system



Efficiency of the system



Reasons of failed

- Mokka: had an error in folder name containing socket, same folder was used simultaneously by several jobs. Also jobs are killed by site (memory problem?))
- Marlin: code 134 and 9 ?
- SLIC: no problem seen since tar ball was put on Storage Element
- LCSIM: issues with detector model, when not standard
- Uploading data to storage: KEK is far (now have access to KEK SE as buffer); registration of files in catalog sometimes not working, being investigated

Outline

Introduction
- DIRAC

ILCDIRAC
- Our developments

Performances
- Achievements
- Number of sites
- Storage of output data

File Catalog

Production system

Conclusion

Sites used

Using only SL5, 64 bit software, 23 sites available

There used to be only 8, but was a configuration issue in DIRAC

Outline

Introduction
- DIRAC

ILCDIRAC
- Our developments

Performances
- Achievements
- Number of sites
- Storage of output data

File Catalog

Production system

Conclusion

Storage

- Output sandbox automatically copied to Storage Element (SE) if larger than 10Mb
- Output data is systematically copied to SE, by default the country associated storage, e.g. jobs run at KEK store the data at KEK.
- Files are added to the catalog

Users don’t need to worry about data management, they only need to specify their output

File Catalog

Use of the default DIRAC FC for the moment:

- Easy to use as is “file system” like
- Contains only CLIC data for the time being
- Can (will) implement an interface with the ILC catalog
    - DIRAC people are working on something similar so we’ll benefit from developments: synchronization between LFC and Dirac FC, reading mechanisms
- Web interface integrated to DIRAC portal will be implemented

Production system

Idea: provide a system that submits and monitors automatically jobs corresponding to a task applied on a set of files and producing another precise set of files.

Working since Friday, currently in use for PandoraNew tests.

Still some developments to do: storing and availability of log files, better User Interface.

Conclusion

- System proves to be very user friendly
- Should be fully ready for production of massive data samples in September
- Still some developments to be done, especially for the File Catalogs

Prospects:

- Implement in prod system handling of log files
- Perform metadata registration of created data automatically
- Work on generators to add them in DIRAC, so full analysis chain would be possible
- Work on DESY file catalog integration in DIRAC

See tomorrow for “live” presentation of the system.