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

March 2010 / LCWS10, Beijing

Outline

Outline

Aim of this work

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

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

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

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 Mokka and Marlin applications to run them safely on the GRID sites

Mokka

Installation procedure:

- Download tar ball from server and untar it
- Set up MySQL server

User input needed:

- Steering file
- Generator file
- Number of events to run
- Optional: events number to start from, DB slice, output file name (when running Marlin after)

One does not need to make changes to the steering file previously used for interactive tests $\Rightarrow$ fewer errors introduced

Marlin

Installation procedure:

- As Mokka: download and untar tar ball
- Setup the environment variables

Input needed:

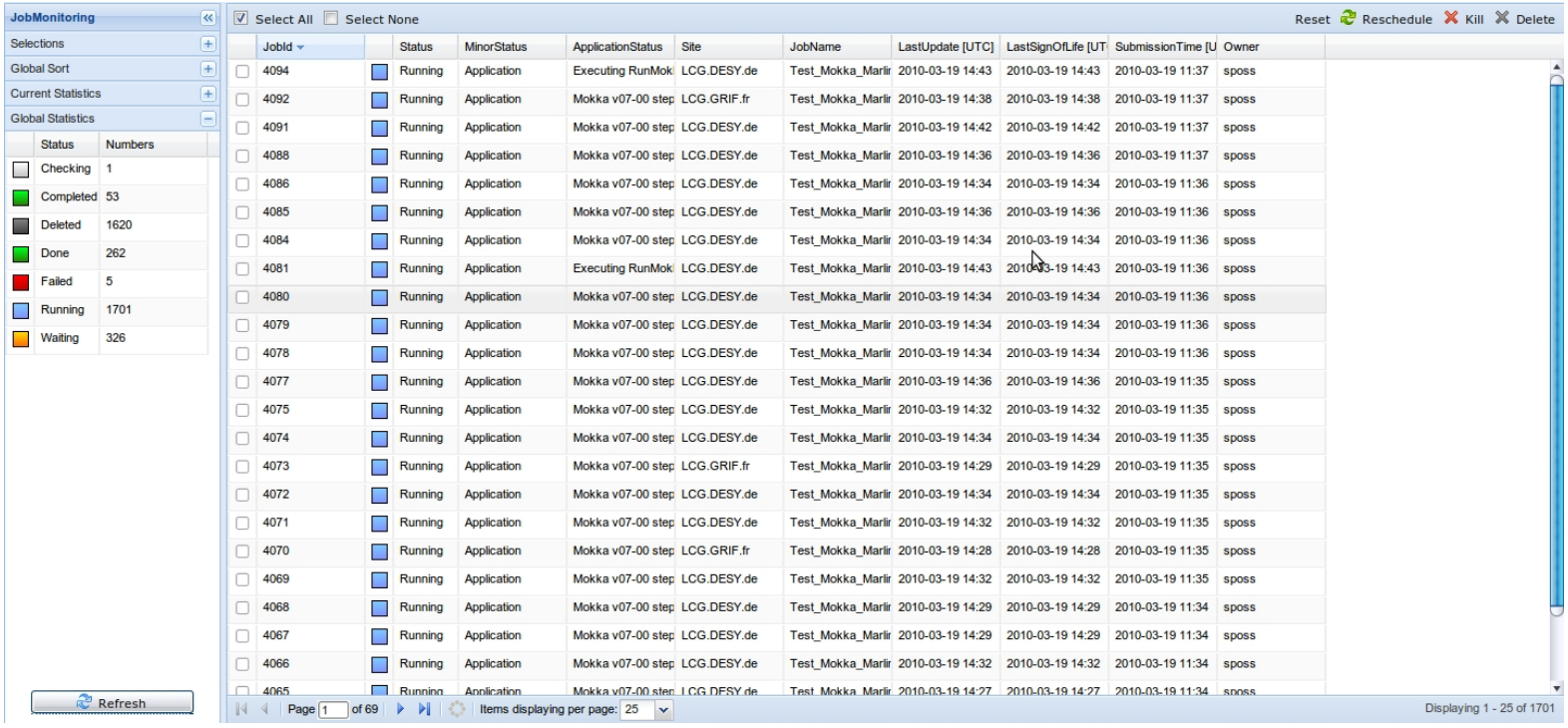
- XML file for steering
- Optional if Mokka was ran before: Gear file, slcio file list, number of events to process

Again, no modification needed in input XML, all are done automatically, more user friendly

Outline

Number of simultaneously running jobs

2000 jobs each using 1000 3TeV $e^+e^- \rightarrow \mu\mu\nu\nu$ events



1700 running in parallel

Outline

Sites used

Using only SL5, 64 bit software, only 6 sites usable:

- DESY (DE)
- KEK (JP)
- GRIF (FR)
- CEA (FR)
- POLGRID (FR)
- IRES (FR)

25 are available for the VO

Outline

Storage

Currently using only IN2P3 storage, but CERN resources should be available soon

- Output files automatically copied to Storage Element (SE) if larger than 10Mb
- Output data is systematically copied to SE
- Files are added to the catalog

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

Outline

MySQL

Problems encountered:

- Threading: MySQLd server must run in background
- Have to wait for socket creation, depends on the site
- Socket path length: limited to 108 characters, while standard GRID directories have > 200
- Using /tmp/ on worker node: sites don't really like that

Outline

FileCatalog and Bookkeeping

Need to:

- Determine common data path structure
- Populate the catalog(s)

In DIRAC, using multiple catalogs in parallel is possible

- DIRAC provides one, in development
- Possibility to implement an interface to the DESY catalog (Jan Engels)

Outline

Other

Software specific:

- Software compatibility with different OS and architecture to run on more sites
- Some channels ($t\bar{t}$) take a long time to process in Mokka, need to split input generator file to avoid hitting the CPU time limit.

VO specific:

- Another VO with same name exists! The resource discovery agent got confused.

Outlook and conclusions

Plans for future:

- File catalog & bookkeeping
- File splitter
- Production system
- Add SiD software (SLIC and LCSIM)

Conclusions:

- Simple to use for any user
- Reliable
- Efficient

Acknowledgments

Many thanks to the DIRAC developers: A. Smith, S. Paterson, A. Tsaregorodstev, J. Closier, V. Roma

Thanks to A. Gellrich for sorting out the VO problems

Thanks to the CERN IT for the support

Thanks to André Sailer for all the help understanding Mokka/Marlin

Backup slides

VO Box requirements

- Local account dirac
- /opt/dirac partition owned by dirac user. Must be large as contains DB and sandbox
- Ports 80, 443 opened for web server
- Ports range 9130:9200 opened for DIRAC services
- Host certificate

MySQL

Installation procedure:

- MySQL inside Mokka tar ball
- Setup the environment variables

Input needed:

- Paths to Local socket, data directory and binary directory: determined at run time
- Database dump file: either specified by user or default from tar ball used

After Mokka run:

- Shutdown MySQLd server through socket
- Remove MySQL from disk

$\Rightarrow$ No user intervention needed.