CERN@school

CERN & Society

The CERN@school Programme: MoEDAL Grid User Guide

T Whyntie[1,2]

[1]Particle Physics Research Centre, School of Physics and Astronomy, Queen Mary University of London, Mile End Road, London, E1 4NS, United Kingdom

[2]The Institute for Research in Schools, Canterbury, CT4 7AU, United Kingdom

E-mail: `t.whyntie@qmul.ac.uk`

Abstract. This document provides a guide to getting on the Worldwide LHC Computing Grid (WLCG) with the Monopole and Exotics Detector at the LHC (MoEDAL) experiment. It covers what is required to get started on the Grid, how MoEDAL is integrated with the WLCG infrastructure, and how to run MoEDAL simulations using the GridPP DIRAC system and Ganga User Interface.

Contents

1. Introduction

1.1. The MoEDAL experiment

The Monopole and Exotics Detector at the LHC (MoEDAL) [1] is the latest addition to a long line of experiments that have searched for Dirac's hypothesised magnetic monopole [2]. Based at the Large Hadron Collider's Interaction Point 8 (IP8), it is housed in the same experimental cavern as the LHCb experiment [3]. Like all experiments at the LHC, meeting the demands of its physics programme [4] requires significant computing power. Simulations of the detector need to be run, data from the subdetectors needs to be stored and processed, and analyses combining the two need to be performed. The Worldwide LHC Computing Grid (WLCG) [5] was designed to meet these needs (and more) of the Large Hadron Collider (LHC) experiments, and thanks to the United Kingdom's GridPP Collaboration [6, 7] MoEDAL has access to the resources the Grid provides. This document aims to provide a guide for users involved with the MoEDAL experiment to harness the computing resources offered by the Grid for the benefit of the MoEDAL Collaboration.

1.2. For whom is this guide written?

This guide is primarily aimed at people working directly on the MoEDAL experiment who wish to use Grid computing resources for their work. You could be:

- a researcher from a MoEDAL member institution;
- a postgraduate student completing a PhD on MoEDAL;
- an undergraduate student working on a project supervised by a MoEDAL researcher;
- a MoEDAL/CERN summer student;
- a school student working with the Institute for Research in Schools (IRIS).

This user guide isn't really aimed at:

- users with no association with the MoEDAL experiment, or;
- readers with little or no computing experience.

Computing knowledge

While every effort has been made to cover as many bases as possible, some computing knowledge is assumed. You can read more about what you might need to know in Section 2.1.

1.3. Overview of the guide

Section 2 covers everything you need to know before attempting to get started on the Grid with MoEDAL, including prerequisites (Section 2.1), conventions used in this guide (Section 2.2), how to get help (Section 2.3), and how this document relates to the GridPP UserGuide (Section 2.4). Section 3 describes how MoEDAL integrates with the Grid's infrastructure, looking at the MoEDAL Virtual Organisation (Section 3.1), how MoEDAL data is stored on the Grid (Section 3.2), and how MoEDAL software is distributed on the Grid with the MoEDAL CernVM-FS repository (Section 3.3). You'll find out how to run MoEDAL simulations in Section 4, which covers setup (Section 4.1), local running (Section 4.2), and grid running (Section 4.3). After a brief look at the next steps to take (Section 5), references, acronymns, and acknowledgements are provided in Sections 6, 7, and 8 respectively.

2. Before we begin

2.1. Prerequisites

If you're working on the MoEDAL experiment, it's very likely that you will have most of the things listed below. But there's no harm in checking!

- **A valid email address**: This sounds kind of obvious - and who *doesn't* have an email address these days? - but you'll need a valid email address from which you can send and receive emails.

> **Institutional email accounts**
>
> *You should use the email account given to you by the school or university associated with the MoEDAL Collaboration, as this will make life a little easier when it comes to granting you access to Grid resources. Only use your CERN email address (if you have one) if it is associated with your Grid certificate.*

- **A GitHub account**: Much of the software used by the GridPP Collaboration and associated partner organisations is hosted on their public GitHub repositories - see, for example, the the GridPP GitHub repository[†] You can sign up for a free GitHub account on the GitHub website.
- **Experience with the command line**: The command line allows you to type instructions into your computer in order to get it to do things for you, rather than relying on clicking on icons, buttons, and other graphical elements of a software package. In his guide, Learn Enough Command Line to be Dangerous[‡], Michael Hartl uses a nice analogy with magic. While it is *technically* possible to use the Grid without using the command line (using, for example, a web browser to access specific Grid systems), using the command line is infinitely easier and gives you much, much more flexibility. Hartl's tutorial, is well worth following if you've not used it before (or even if you have!).
- **A text editor**: We'll be writing scripts - series of commands to be executed one after the other - and for this you'll need a text editor of some description. Emacs, Vim, Vi - whatever you feel most comfortable with. Vim, for example, allows you to edit text from the command line.
- **Programming with Python**: Once we start getting fancy with the MoEDAL software and the Grid, we're going to use the Python programming language to do a lot of the work for us. As such, some familiarity with Python will be handy. There are plenty of (free!) online tutorials available that can get you started. We'll provide plenty of examples too, so don't panic.

† See http://github.com/gridpp
‡ See http://www.learnenough.com/command-line-tutorial

- **A Scientific Linux 6 command line with CVMFS access**: This will either be provided by your institution or via a GridPP CernVM, a Virtual Machine (VM) made by CERN that you can run yourself. The CernVM-File System, a.k.a. CernVM-FS or CVMFS, gives you (and any grid node, for that matter) instant access to all sorts of software *without having to install anything*. So it's worth sorting out! You can find instructions for creating a GridPP CernVM the GridPP UserGuide.
- **Access to a CERN GitLab account** (CERN users only): CERN has its own Git repository service, and every CERN user automatically has an account on it. You'll need to log on and add an SSH key to use it though - see [this page](https://gitlab.cern.ch/help/ssh/README.md) for more information.
- **An SSH key**: Secure Shell (SSH) keys are a way to identify trusted computers without involving passwords. You can generate an SSH key and add the public key to your GitHub account by following the procedures outlined in this this guide from GitHub[†].

Got/done all of that? Good. Now let's look at the conventions used in this guide.

[†] See https://help.github.com/articles/generating-an-ssh-key/

2.2. Conventions used in this guide

The conventions used in this guide are, by and large, self-explanatory. Here we'll look at a few that might not be.

2.2.1. The command line

Following Hartl, we'll present command line examples using a Unix-style command line prompt (a dollar sign), as follows:

```
$ echo "Hello, MoEDAL!"
Hello, MoEDAL!
```

i.e. you type what follows the dollar sign, and hopefully see the same (dollar sign-less) output in your terminal.

Comparing the output

Computer systems are always going to vary from machine to machine, so you may not see exactly the same output from a given command. If something doesn't look right, more often than not a combination of common sense and Googling the output should confirm if you're on the right track.

Where possible we'll use bash environment variables to account for differences in working directories. However, if a particularly user-specific input is needed we'll use square brackets to denote parts of the command that require input specific to your circumstances. For example, when setting your working directory environment variable `WORKING_DIR`, we'd write this:

```
$ export WORKING_DIR=[Your working directory.]
$ echo $WORKING_DIR
[The value of $WORKING_DIR, hopefully your working directory.]
```

which would actually be completed using:

```
$ export WORKING_DIR=/home/lmeitner/moedal-stuff/
$ echo $WORKING_DIR
/home/lmeitner/moedal-stuff/
```

2.2.2. Code listings

Generally speaking, we have tried to avoid listing large swathes of code in this guide - that's what GitHub is for. From time-to-time it may be useful to include a code snippet like the following:

```
#!/usr/bin/env python
print("* This works!")
```

Following Hartl, we will use vertical dots to represent code omitted for the sake of brevity:

```
#!/usr/bin/env python

class GridJob:
    .
    .
    .
    def submit(self, id):
        self.id = id
        .
        .
        .
```

These dots should not be copied into your code.

2.2.3. Hints, warnings, and information

This guide, like many instructional handbooks, uses little pop-out boxes to highlight important points throughout the text.

Hint

This is a hint. Hint boxes are used for pointing out things that might be useful while carrying out the task being described (particularly where we have received user feedback on a given step!).

Warning!

This is a warning. These are used to flag up potential pitfalls or issues you may need to be aware of to avoid making mistakes or doing Something Bad.

Point of information

This is a point of information. These boxes will generally present things that may not directly relate to the topic being discussed but are nonetheless interesting.

2.3. Getting help

There are many ways of getting help and support if you run into problems while working with MoEDAL on the Grid. If you don't happen to have a Grid expert in the office down the corridor, you can try the methods described below.

2.3.1. Googling the error

We can't possibly account for every error a user might encounter when working through this guide, so on encountering a problem your first port of call should be sticking the error message into your Search Engine of Choice.

Let Me Google That For You

This is actually a pretty good approach to software development in general. Thanks to vibrant, enthusastic communities like those at StackExchange many common computing gotchas have been documented and solved on the World Wide Web - so it's always worth checking!

2.3.2. The MoEDAL mailing lists

The mailing lists listed in Table 1 can provide support from both the MoEDAL and GridPP Collaborations.

Table 1: MoEDAL-related support mailing lists. To join them, access the corresponding CERN e-Group or JISC page and submit a request to join.

List	Description	Provider
`MoEDAL-Software@cern.ch`	For MoEDAL software support.	CERN
`MoEDAL-gridsupport@cern.ch`	For MoEDAL-specific Grid support.	CERN
`GRIDPP-SUPPORT@jiscmail.ac.uk`	For support from GridPP.	JISCMail

2.3.3. GridPP-specific issues

If you have a problem relating specifically to the software or infrastructure supported by GridPP (e.g. DIRAC, Ganga, CVMFS, etc.) the easiest way to report problems is by raising an **issue** on the *GridPP UserGuide*'s GitHub repository. Simply log in to GitHub, visit the *UserGuide* issues page and click on the New issue button.

Provide as much information as you can when raising an issue. You can also use the MarkDown format to create hyperlinks and add formatting to your issue.

Don't forget to **Watch** the repository too. You can do this by going to the repository, signing in with your GitHub account, and clicking on the Watch button at the top-right of the page. You'll then be kept up-to-date with issues and new versions as the GridPP infrastructure evolves over time.

2.4. The GridPP UserGuide

Finally, you are strongly advised to read and work through the user guide provided by the GridPP Collaboration [6, 7] on their website:

http://www.gridpp.ac.uk/userguide

This will get you up and running with the Grid software and infrastructure supported by GridPP. You will also join the `gridpp` Virtual Organisation as a "new user".

> Déjà vu?
>
> *If some of the material in the GridPP UserGuide looks familiar, don't be too surprised – this guide was prepared by the same author. Indeed, the integration of the MoEDAL Virtual Organisation has served as a useful case study for GridPP's New User Engagement Programme.*

Right. Let's begin!

3. MoEDAL on the Grid

3.1. The MoEDAL Virtual Organisation

The MoEDAL Virtual Organisation (VO) is a European Grid Initiative (EGI) approved VO that represents MoEDAL on the Worldwide LHC Computing Grid (WLCG). Following EGI naming conventions, it is called:

`vo.moedal.org`

Full details about the VO can be found in the EGI Operations Portal here:

https://operations-portal.egi.eu/vo/view/voname/vo.moedal.org

To join the MoEDAL VO, one should follow the same procedure described in the GridPP UserGuide for joining the `gridpp`: simply access the VO's joining page with a web browser in which you have installed your Grid certificate:

https://voms2.cern.ch:8443/voms/vo.moedal.org/register/start.action

Once you have received an email confirming that your membership has been approved, you'll be able to access the GridPP DIRAC web portal as a **`vo.moedal.org_user`** :

https://dirac.gridpp.ac.uk

by selecting the MoEDAL VO from the drop-down menu in the bottom-right corner of the web page. You should also join the appropriate MoEDAL Grid e-Groups/mailing lists listed in Table 2 (i.e. all of them except the managers' list - unless you *are* a VO Manager, of course).

Table 2: The MoEDAL Grid mailing lists. To join them, access the corresponding CERN e-Group and submit a request to join.

List	Email address
VO Managers	**`MoEDAL-gridmanagers@cern.ch`**
Operations	**`MoEDAL-gridops@cern.ch`**
Users support	**`MoEDAL-gridsupport@cern.ch`**
Users	**`MoEDAL-gridusers@cern.ch`**
Security	**`MoEDAL-gridsecurity@cern.ch`**

Once you've done all that you'll be a proper member of the MoEDAL Virtual Organisation. You're now ready to work with MoEDAL on the Grid.

3.2. MoEDAL data storage in the GridPP DFC

3.2.1. Accessing data on the DFC

If you've followed the GridPP UserGuide, you'll know how to access data stored on the Grid via the DIRAC File Catalog (DFC). The only thing that's different for MoEDAL work is the VO you'll use when generating your Grid proxy. So for example, to access the DFC via the DFC Command Line Interface (DFCCLI):

```
$ source /cvmfs/ganga.cern.ch/dirac_ui/bashrc
$ dirac-proxy-init -g vo.moedal.org_user -M
Generating proxy...
Enter Certificate password: # Enter your grid certificate password...
.
. [Proxy information-based output.]
.
$ dirac-dms-filecatalog-cli
Starting FileCatalog client

File Catalog Client $Revision: 1.17 $Date:

FC:/>
```

As one would expect, MoEDAL data can be found in MoEDAL's space in the DFC.

```
FC:/> ls /vo.moedal.org/
sim
data
user
```

3.2.2. Adding data to the MoEDAL DFC

From here it is straightforward to add data to the DFC by following the instructions in the GridPP UserGuide. Your own data should be added to your space in the MoEDAL user section:

```
FC:/> cd /vo.moedal.org/user/l/lise.meitner
FC:/> pwd
/vo.moedal.org/user/l/lise.meitner
FC:/> mkdir my_moedal_data
FC:/> ls
my_moedal_data
```

The DIRAC command line tools

Don't forget you can use the DIRAC command lines tools to automate uploads and downloads once you have created your user space in the MoEDAL DFC.

3.2.3. An example: the 13 TeV LHE files

Experiment-wide data (i.e. data used by everyone in the Collaboration) can and is also hosted by GridPP and listed in the DFC. For example, the Les Houches Event (LHE) files used in the 13 TeV GEANT4 simulations of the MoEDAL experiment may be found here:

```
FC:> cd /vo.moedal.org/sim/MMT13TeVa/LHE/13TeV/SpinHalf/Events
FC:> ls
run_01
run_02
run_03
run_04
run_05
run_06
run_07
run_08
run_09
run_10
FC:> cd run_01
FC:> ls
events.lhe.gz
run_01_tag_1_banner.txt
unweighted_events.lhe.gz
unweighted_events.root
```

These files can then be used by jobs running on the Grid with a MoEDAL VO proxy, as we'll see in Section 4.3.

3.3. The MoEDAL CVMFS repository

The CernVM File System – a.k.a. CernVM-FS or CVMFS – is used to distribute software on the Grid. You can read more about it in [8]. It is part of the CernVM project [9]. You'll have read about it in the GridPP UserGuide too, but in essence it means MoEDAL's software will be instantly available on any CernVM File System (CVMFS)-enabled computing node you use.

3.3.1. Finding the MoEDAL CVMFS repository

MoEDAL's CVMFS repository can be found here:

`/cvmfs/moedal.cern.ch/`

So, for example, you can find MoEDAL's Gauss software by logging on to any computing node with the MoEDAL CVMFS repository enabled (e.g. a GridPP CernVM) and typing:

```
$ ls /cvmfs/moedal.cern.ch/Gauss
Gauss_v48r1   Geometry
```

Simulations of the MoEDAL experiment, based on LHCb's Gauss software, are currently running with `v48r1`. The custom MoEDAL Gauss modules used may be found here:

```
$ ls /cvmfs/moedal.cern.ch/Gauss/Gauss_v48r1/Sim
Gauss   GaussMoEDAL   GaussMonopoles
```

CVMFS and the LHCb software framework

As you'll see from the next section, we actually use a lot of Large Hadron Collider Beauty (LHCb)'s software in MoEDAL simulations – and it too is available via CVMFS!

We also use CVMFS to distribute the latest MoEDAL geometry files. The latest version of the MoEDAL geometry at the time of writing, `2.0.0`, may be found here:

```
ls /cvmfs/moedal.cern.ch/Gauss/Geometry/2-0-0/
geometry_default.db   geometry_maximal.db   geometry_minimal.db
```

As you'll see in the next section, these are used in the simulations – something that is made trivial by publishing them on the CVMFS repository.

Do not use CVMFS for data!

Note that we do not use CVMFS for data. This is what the DFC is for. The only exception to this might be data used for software testing and validation, but even then using Grid data as part of your tests might not be a bad thing!

3.3.2. Managing the MoEDAL CVMFS repository

To add and remove software from the MoEDAL CVMFS repository, you will need to be a member of the `LxCvmfs-moedal` CERN e-Group†. To access the repository, once you are confirmed as a member of the `LxCvmfs-moedal` e-Group, log onto CERN's lxplus system with your CERN computing account:

```
$ ssh -Y meitner@lxplus.cern.ch
Password:
* *******************************************************************
* Welcome to lxplus0048.cern.ch, SLC, 6.8
* Archive of news is available in /etc/motd-archive
* Reminder: You have agreed to comply with the CERN computing rules
* https://cern.ch/ComputingRules
* Puppet environment: production, Roger state: production
* Foreman hostgroup: lxplus/nodes/login
* LXPLUS Public Login Service
* *******************************************************************
```

Managing CVMFS repositories

As you may have guessed, you'll need a CERN computing account to manage the MoEDAL CVMFS repository. We'll assume that if you've got as far as needed to manage the repository, this shouldn't be a problem.

From there you'll be able to log into the MoEDAL CVMFS repository:

```
$ ssh -Y meitner@cvmfs-moedal.cern.ch
```

After which you'll be presented with most of the instructions you need to start a CVMFS **transaction**, which you'll need to do to make changes to what's in the repository. You should refer to the full instructions for managing a CVMFS repository on the CernVM TWiki page:

https://twiki.cern.ch/twiki/bin/view/CvmFS/MaintainRepositories

Which also includes useful hints and tips on directory structure, managing your software, and how to get the best out of the CVMFS system.

† Full title: Writers of `/cvmfs/moedal.cern.ch`

4. Running MoEDAL simulations

We'll now look at running a simulation of the MoEDAL experiment. Specifically, we'll be running the simulation of monopole-antimonopole production in 13 TeV proton-proton collisions at Interaction Point 8, i.e. the LHCb cavern. We will use the LHCb and MoEDAL software frameworks to simulate the passage of the resulting (anti)monopole through the MoEDAL detector material to see how the various subdetectors perform.

> **Analysing the results**
>
> *Note that we will not cover the analysis of the simulation results here – this is beyond the scope of this guide, as (generally speaking) once you have the output files from the Grid you don't need the Grid to do the analysis.*

We'll actually run the simulation on your local machine before running it on the Grid. This is actually a good thing to do generally, as it will let you quickly debug any problems before running your code on a remote system. Fortunately, switching between local running and Grid running is simple thanks to the Ganga User Interface (UI)[†].

4.1. Setting up

Firstly, you'll need to clone the repository from the Organisation Européen pour la Recherche Nucláire (CERN) GitLab[‡] system. Log on from your CVMFS-enabled machine, create a working directory in your user space (which we will call `$WORKING_DIR`) and check out the code as follows:

```
$ cd $WORKING_DIR # wherever you want to run your simulations from.
$ git clone ssh://git@gitlab.cern.ch:7999/moedal/moedal-run-simulations.git
$ cd moedal-run-simulations
```

> **SSH keys**
>
> *This repository cloning method assumes you have your SSH key in your machine's* `/.ssh` *directory, and that you have uploaded the public key to the CERN GitLab instance, as described in Section 2.1.*

> **Port unblocking**
>
> *Note that the CERN GitLab instance is accessed via port 7999 – make sure this is unblocked in any of your system's firewalls.*

[†] See http://ganga.web.cern.ch
[‡] See http://gitlab.cern.ch

4.2. Local running with Ganga

First, go to the **ganga** directory within the MoEDAL simulation script repository, and then start Ganga:

```
$ cd $WORKING_DIR/moedal-run-simulations/ganga
$ source /cvmfs/ganga.cern.ch/runGanga.sh
```

You'll then see lots of output telling you about how Ganga has started, including the location of your Ganga configuration file `.gangarc`. Make a note of this. The prompt will now change to an interactive iPython prompt. It is from here that you can submit commands to Ganga. We have prepared a script that submits a MoEDAL simulation job to your local system. The beauty of Ganga is that it treats local, batch and grid jobs in the same way, so all it takes is a small tweak of your config file to change from local to grid running. The script creates, configures, and submits a small (four events) LHE simulation to your local system. You can run it and submit the job with:

```
Ganga In [x]: execfile‘(make_local_job.’py)
Ganga In [x]: j.submit()
```

The Ganga command prompt

Note: the small x will actually be the number of the command you've typed into the Ganga prompt, and will keep incrementing by one with every command...

You should then be informed that your job has been submitted via the Ganga output. You can check the status of your jobs with the following Ganga command:

```
Ganga In [x]: jobs
```

Once your job has finished running (showing as `Completed` in the jobs table), you can get the output directory using:

```
Ganga In [x]: j = jobs(X) # where X is the job number in the table.
Ganga In [x]: j.outputdir
```

You can then inspect the contents of this directory (including `stdout` and `stderr`) as usual.

You can find more information about using Ganga here:

http://ganga.readthedocs.io

Aliasing Ganga

Add an alias to your `.bashrc` file to save typing the whole command every time:
`alias run_ganga_cvmfs='source /cvmfs/ganga.cern.ch/runGanga.sh'`

4.3. Grid running with Ganga

DIRAC [10][†] is a software framework for distributed computing that the GridPP Collaboration uses to manage jobs and data on the Grid using their own GridPP Distributed Infrastructure with Remote Agent Control (DIRAC) instance [11, 12] – so we're going to use it too. To get started, we firstly need to change a few settings in our Ganga configuration file to reflect the fact we're running with the MoEDAL VO. Specifically, in your `.gangarc` file (found in your home directory), find and set the following variables:

```
[defaults_DiracProxy]
...
group = vo.moedal.org_user
...
[DIRAC]
...
DiracLFNBase = /vo.moedal.org/user/l/lise.meitner
...
```

(Except use your own DFC user space, of course!)

Then we initialise DIRAC and generate a Grid proxy as usual:

```
$ source /cvmfs/ganga.cern.ch/dirac_ui/bashrc
$ dirac-proxy-init -g vo.moedal.org_user -M
Generating proxy...
Enter Certificate password:
[...output...]
```

Remember how we said switching to Grid running with Ganga was trivial? Well, we just have to use a slightly script to create the Grid-ready job and then submit it as before:

```
Ganga In [x]: execfile('make_dirac_job.py')
Ganga In [x]: j.submit()
```

And that's it! You can check on the status of your job in the GridPP DIRAC system by accessing the GridPP DIRAC page here:

https://dirac.gridpp.ac.uk

With a browser that has your grid certificate installed. You should select **vo.moedal.org_user** from the bottom-right drop down menu, and then **Job Monitor** from the **Jobs** drop-down menu in the top-left corner.

> Submitting jobs to the Grid
>
> *Submitting jobs to the grid can take some time. You might want to make a cup of tea or do something else while waiting!*

[†] See http://diracgrid.org

5. What's next?

Now you can submit jobs successfully, you can start to play around with the parameters in `make_dirac_job.py` to produce different simulation runs. For example, you could use loops with the following variables to submit many jobs at once:

```
## The spin of the magnetic monopole.
monopole_spins = ['SpinZero', 'SpinHalf']

## A dictionary of monopole masses and corresponding LHE run name.
monopole_masses = { \
   200:'run_01', \
   500:'run_02', \
  1000:'run_03', \
  1500:'run_04', \
  2000:'run_05', \
  2500:'run_06', \
  3000:'run_07', \
  4000:'run_08', \
  5000:'run_09', \
  6000:'run_10'  \
}

## A dictionary of magnetic charges to corresponding LHE filenames.
monopole_magnetic_charges = { \
  1:'q10', \
  2:'q20', \
  3:'q30', \
  4:'q40', \
  5:'q50', \
  6:'q60'
}

## A dictionary of geometry names to geometry files.
geometries = { \
  'default':'geometry_default.db', \
  'maximal':'geometry_maximal.db', \
  'minimal':'geometry_minimal.db'
}
```

So there we go – having followed this guide, you should be able to submit MoEDAL simulation jobs using Ganga and GridPP DIRAC. You should also be able to develop new job configurations for new physics scenarios. These should be added to the GitLab repository here:

https://gitlab.cern.ch/moedal/moedal-run-simulations

Good luck!

6. References

[1] The MoEDAL Collaboration. Technical Design Report of the Moedal Experiment. Technical report, June 2009.

[2] P.A.M. Dirac. Quantised Singularities in the Electromagnetic Field. *Proc. Roy. Soc. A*, 133:60, September 1931.

[3] The LHCb Collaboration. The LHCb Detector at the LHC. *JINST*, 3:S08005, 2008.

[4] The MoEDAL Collaboration. The Physics Programme Of The MoEDAL Experiment At The LHC. *Int. J. Mod. Phys. A*, 29:1430050, 2014.

[5] I. Bird, K. Bos, N. Brook, D. Duellmann, C. Eck, I. Fisk, D. Foster, B. Gibbard, C. Grandi, F. Grey, et al. LHC Computing Grid: Technical Design Report. 2005.

[6] The GridPP Collaboration. GridPP: Development of the UK Computing Grid for Particle Physics. *J. Phys. G*, 32:N1–N20, 2006.

[7] D. Britton et al. GridPP: the UK grid for particle physics. *Phil. Trans. R. Soc. A*, 367:2447–2457, 2009.

[8] J. Blomer et al. The Evolution of Global Scale Filesystems for Scientific Software Distribution. *Computing in Science and Engineering*, 17:61–71, 2015.

[9] G. Ganis et al. Status and Roadmap of CernVM. *J. Phys.: Conf. Ser.*, 664:022018, 2015.

[10] A. Tsaregorodtsev et al. DIRAC pilot framework and the DIRAC Workload Management System. *J. Phys.: Conf. Ser.*, 219:062049, 2010.

[11] D. Bauer et al. The GridPP DIRAC project: Implementation of a multi-VO DIRAC service. *J. Phys.: Conf. Ser.*, 664:062009, 2015.

[12] D. Bauer et al. The GridPP DIRAC project - DIRAC for non-LHC communities. *J. Phys.: Conf. Ser.*, 664:062036, 2015.

7. Acronyms

CVMFS CernVM File System

DFC DIRAC File Catalog

DFCCLI DFC Command Line Interface

DIRAC Distributed Infrastructure with Remote Agent Control

CERN Organisation Européen pour la Recherche Nucláire

EGI European Grid Initiative

IRIS The Institute for Research in Schools

LHC Large Hadron Collider

LHCb The Large Hadron Collider Beauty experiment

LHE Les Houches Event (file format)

MoEDAL Monopole and Exotics Detector at the LHC

SSH Secure Shell

UI User Interface

VM Virtual Machine

VO Virtual Organisation

WLCG Worldwide LHC Computing Grid

8. Acknowledgements

The author would like to thank MoEDAL Collaboration members D. Felea (ISS/Bucharest), S. Baines (Uni. Manchester) and J. Flores (Stony Brook Uni.) for support and documentation efforts, and J. Pinfold (Uni. Alberta) and P. Mermod (Uni. Geneva) for many fruitful discussions. We are indebted to D. Bauer, S. Fayer (Imperial College London), and J. Martyniak (now at RAL) for their tremendous work developing and supporting GridPP DIRAC; as well as the Ganga development team[†] for their work on and support for the Ganga UI. Their combined efforts have transformed how small groups can use the Grid and without them the work described here would not have been possible.

This work was supported by the UK Science and Technology Facilities Council (STFC) via grant numbers ST/J000256/1 and ST/N00101X/1, as well as a Special Award from the Royal Commission for the Exhibition of 1851. Computing resource and expertise was kindly provided by the UK's GridPP Collaboration [6, 7][‡].

Version History

Table 3: Version history.

Version	DOI	Description	Author
1.0	10.5281/zenodo.255649	Initial version.	TW

† See https://github.com/ganga-devs/
‡ See http://www.gridpp.ac.uk