Grid and data management (and saving yourself time)

Robert W. Lambert

Introduction

- Prerequisites
  - Good knowledge of Python
  - Grid certificate
  - Have done the Ganga tutorials
  - Have a LHCb Ganga job which makes some output files
- Learning outcomes
  - Achieve 100% success rate on the grid
  - Reduce the mails to the distributed analysis list
  - You will know how to manage your files on the grid

Outline

- The Grid
  - When to use the grid
  - Expectations
  - Getting the most out of the grid
- Data management
  - Where are my data?
  - What can I do with my data?
  - Cleaning up after yourself

Common misconceptions

- The grid is slow
  - Oh it really isn't
- The grid never works
  - It works just fine 80% of the time
- It's hard to find the data
  - The BK is very easy (Tutorial)
- The grid is only for experts
  - That's why we have Ganga (Tutorial)
- Ganga can't do…
  - Yes, but it can do a zillion easier things
  - Why do you want to do *that* anyway?

Common (Bad) Model

- Are you guilty of:

Options from someone else or never tested

Script from someone else or "the same as last time"

Submit straight to Grid

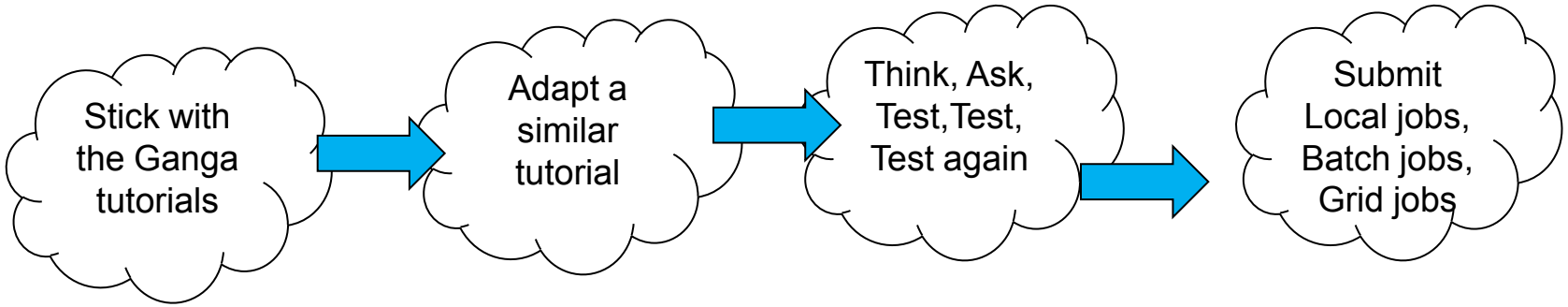
Failure

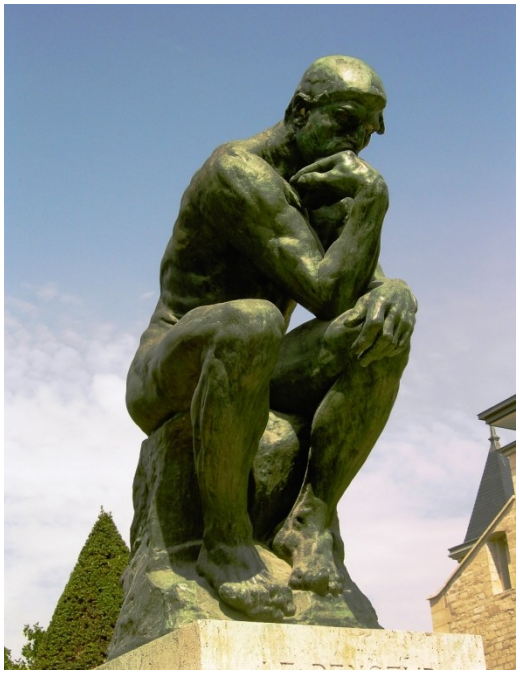
Computer says no.

mailing list or Give up! Go to batch.

Better Model

- Test and understand your code!





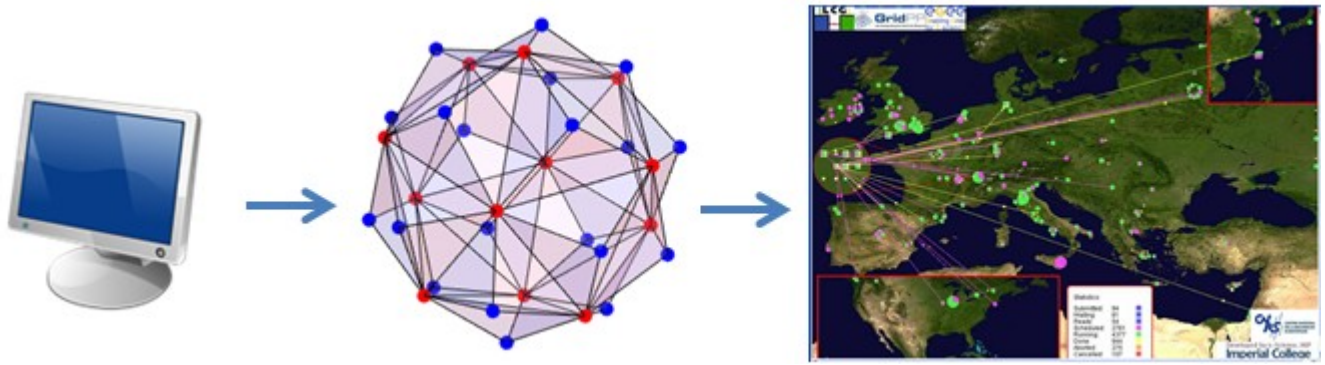
Success!

The Ganga Mantra

- Configure once, run anywhere!

The Ganga Mantra

- Configure once, run anywhere!
- There's a reason Ganga is programmed this way:



- Test locally and with small sets of jobs before submitting 10k!
- This will solve 99% of your "Grid" problems

Success or Failure?

- Failure means:
  - Something went physically wrong with your job on the grid
  - Maybe, though, it was after the important step
- Completed means:
  - The job has stopped
  - It doesn't mean it reached the end, or that it did anything at all
  - Many segfaulting jobs still 'complete'
- Success is defined by you:
  - That ntuple was produced and is readable
  - That line of stdout was printed
  - All input files were seen and processed
  - The integrated lumi calculated is >0

Failures

- Success rate
  - You should expect a finite success rate <100%
  - With adequate testing this is usually 80% on the first pass
  - Most emails to the list are ‘one out of my 100 jobs fails’
    - Wow! That’s 99% success!! Congratulations!!! ZOMG
- Remaining failures
  - Glitch at the site
  - Ran out of CPU
  - Data were not available at that exact moment
- **If 0% succeed, you’ve done something wrong**
- **If 0% succeed at a given site, something else is wrong**

Reaching 100%

- Follow a simple procedure

Check one of the failed jobs

- Weird failure
- Stopped in the middle of the job
- Skipped a lot of files
  Some error on reading a file

80%

Try just resubmitting it

96%

Increase CPU and resubmit

99.2%

Resubmit with the previous site banned

99.8%

Gather up failures into a new job, resplit, resubmit

99.97%

```
In [1]: job('7.99').resubmit()
In [2]: job('7.99').backend.settings={'CPUTime':…*2}
In [3]: job('7.99').backend.settings={'BannedSites':['LCG.CERN.CH']}
In [4]: j=jobs(7).copy()
In [5]: j.inputdata=jobs('7.99').inputdata
In [6]: j.splitter.filesPerJob=1+len(j.inputdata.files)/10
In [7]: j.submit()
```

CPUTime

- Typical queues are in wall-clock time:
  - 1 hour, 8 hours, 24 hours, 48 hours
  - If you use part of a slot, don't worry, DIRAC will fill it up again
- CPU on the grid is in HEPSPEC06 units
  - Performance of the machine * time spent
  - Roughly ½ the speed of an lxplus node
- CPU depends on the input
  - One subjob can take 50% of the total CPU

- CPU reported depends on the machine
  - Not perfectly normalized, sometimes multiplied by #cores!
- Rule of thumb: **(4-8) times the estimate from your local job**

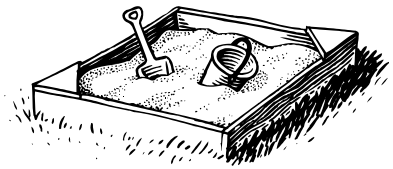
Part II

- Yeah! 100% of my jobs succeeded

… where are my data?

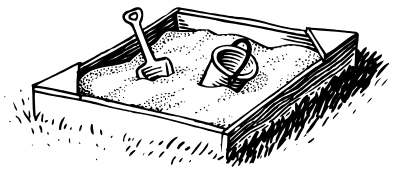
Input

- There are two types of input

1	2
	
Input Sandbox (j.inputsandbox)	Input Data (j.inputdata)
Small files. Need to be copied from your local machine, with the job, to the input directory of the job. e.g. options, scripts, code	Large files. Not copied from your local machine. May exist only where the job is supposed to run. e.g. data from the experiment!

Output

- There are two types of output

1	2
	
Ouput Sandbox (j.outputsandbox)	Ouput Data (j.outputdata)
Small files. Need to be copied once the job has completed from the host machine, to your local machine. e.g. stdout, stderr, Histograms	Large files. Not copied to your local machine. Uploaded to the GRID SE or to CERN CASTOR e.g. Massive ntuples, DSTs to share

Datasets

- inputdata, and outputdata, are Datasets in Ganga
- A dataset is a collection of data files, of one of two types:

Physical Files

You know where the file is, and how to access it. It is located on a disk, this is the actual file you want your job to run over.

```
pf = PhysicalFile('/disk/some/pfn.file')
```

Logical Files

The file may be anywhere on the GRID. The access may be through some obscure protocol, and might be authenticated. The Logical File Name (LFN) is just the name of the file on the GRID, not where it is!

```
lf = LogicalFile('/lhcb/some/lfn.file')
```

Let's move some data

- Ganga makes this easy

```
In [1]: j=jobs(7).copy()
In [2]: j.outputsandbox
Out[1]: ['DVHistos.root', 'DVnTuples.root']
In [3]: j.outputsandbox= ['DVHistos.root']
In [4]: j.outputdata=['DVnTuples.root']
In [5]: j.outputdata.location='GridTutorial'
In [6]: j.submit()
```

- Copy data around

```
In [1]: ds=j.backend.getOutputDataLFNs()
In [2]: ds.replicate('CERN-USER')
In [3]: ds[0].download('/tmp/')
In [4]: afile=PhysicalFile('/tmp/DVnTuples.root')
In [5]: dscp=afile.upload('/lhcb/user/<u>/<uname>/GridTutorial/DVnTuples.root')
In [6]: j.backend.getOutputData()
```

Where are my data?

- In ganga

```
In [1]: j.application.outputdir
In [2]: j.peek()
In [3]: ds=j.backend.getOutputDataLFNs()
In [4]: reps=ds.getReplicas()
In [5]: reps[ds[0].name]
Out[9]: {'CERN-USER': 'srm://srm-
lhcb.cern.ch/castor/cern.ch/grid/lhcb/user/r/rlambert/2010_10/1
1645/11645816/2010.RedoStripping_B0q2DplusMuXTuned.dst',
'IN2P3-USER':
'srm://ccsrm.in2p3.fr/pnfs/in2p3.fr/data/lhcb/user/r/rlambert/2
010_10/11645/11645816/2010.RedoStripping_B0q2DplusMuXTuned.dst}
```

- How much space is that using? (out of ganga)

```
$ SetupProject LHCbDirac
$ dirac-dms-storage-usage-summary --Dir /lhcb/user/<u>/<username>
DIRAC SE              Size (TB)   Files
-----------------------------------------------
CERN-USER             1.3         2230
CERN-tape             0.0         215
…
```

How much data can I keep?

- *Depends where it is*:
  - Your afs home directory: almost nothing
  - The disk on your computer: as much as you like
  - The disk at your institute: as much as they let you
  - Your CERN CASTOR_HOME directory:
    - Well, lots, eventually it will be sent to tape
  - Your GRID storage:
    - Keep <2 TB, depending on activity
- *So please don't*:
  - Merge central files (nor run on unmerged files)
  - Run your own private MC generation
  - Run your own private reconstruction/stripping
  - Forget to clean up after yourself

Cleaning up

- In ganga:

```
In [1]: ds=j.backend.getOutputDataLFNs()
In [2]: for d in ds
  ....:     d.remove()
```

- I don't want to keep the jobs, but want to keep the output

```
In [1]: ds=j.backend.getOutputDataLFNs()
In [2]: box.add(ds, j.id+' '+j.name+' Output LFNs')
In [3]: j.remove()
```

- I don't have the jobs any more (out of ganga)

```
$ SetupProject LHCbDirac
$ dirac-dms-user-lfns
$ dirac-dms-remove-files <a-list-of-lfns>
```

- https://twiki.cern.ch/twiki/bin/view/LHCb/GridStorageQuota

Running on Existing Data

- Your own DSTs
  - Too large to store locally? Need to share with others?
  - No need to download the DSTs
  - You may want to replicate them to more than 1 site
  - Run directly from the Grid with the LFNs
- Your own nTuples
  - Hopefully small enough for InputSandbox
  - No? Download, merge, upload or copy to Castor, **clean up**

```
In [1]: help(j.backend.getOutputData)
In [2]: help(RootMerger)
```

  - Access directly from castor (only a little slower than local)

```
[1] Tfile* f=Tfile::Open('castor:/castor/...')
[2] TBrowser b
```

Hands-on

1. Ganga tutorials
   - First exposure to Ganga
2. DaVinci tutorials
   - Do something useful, make some output files
3. Ganga, grid and data-management (this talk)
   - Submit some things to the grid
   - Juggle output locations
   - Manage your data
   - Remove your data

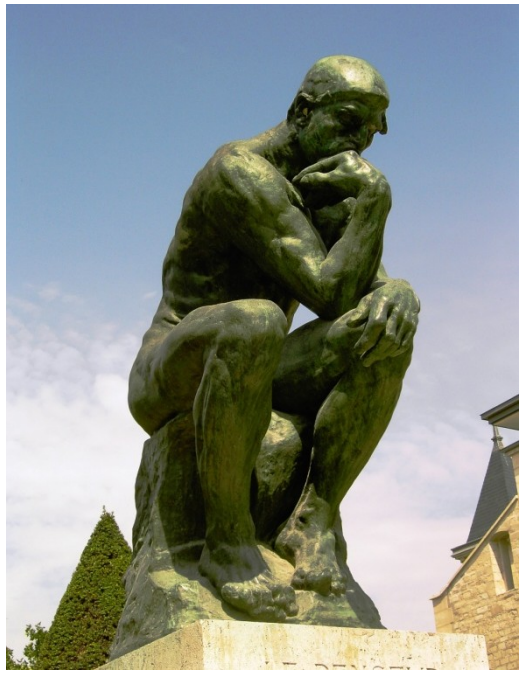
➢ https://twiki.cern.ch/twiki/bin/view/LHCb/GridAndDataManagement

Final Tips

- Use the interactive prompt (as in the tutorials)
  - Once you have one job which works, you're ready for anything
- Use the summary.xml:
  - Like stdout only much much smaller
  - (you can remove stdout most of the time)
  - https://twiki.cern.ch/twiki/bin/view/LHCb/DaVinciTutorial0
- Ganga utilities
  - A user-driven set of python functions for Ganga
  - Really saves time, since we all do the same things
  - (examples in this tutorial are mostly one line)
  - https://twiki.cern.ch/twiki/bin/view/Main/LHCbEdinburghGroupGangaUtils

What you've learnt

1. Ganga makes your life easier
2. Ganga is not a replacement for your brain
3. How to be effective on the GRID
4. How to manage your grid data



End

- Backups are often required