







User Analysis of LHCb Data with Ganga

CHEP 2009

Prague March 23-27

Andrew Maier (CERN/IT) for the Ganga Team

Overview

-  Ganga Introduction
-  Ganga user statistics in LHCb
-  Performing an analysis in LHCb
 -  Data selection
 -  Application configuration
 -  Job submission

Ganga Talks at CHEP 2009

- 👁️ This is a part of a series of talks and posters on Ganga
 - 👁️ General Ganga presentation
 - 👁️ D. van der Ster: [Ganga: User-friendly Grid job submission and management tool for LHC and beyond](#)
 - 👁️ Ganga in ATLAS
 - 👁️ J. Elmsheuser: [Distributed Analysis in ATLAS using GANGA](#)
 - 👁️ **Ganga in LHCb**
 - 👁️ [312] [User analysis of LHCb data with Ganga](#)
 - 👁️ Functional testing using Ganga
 - 👁️ D. van der Ster: [Functional and Large-Scale Testing of the ATLAS Distributed Analysis Facilities with Ganga](#)
 - 👁️ Poster
 - 👁️ D. van der Ster: A PanDA Backend for the Ganga Analysis Interface

Ganga Introduction



What is Ganga?

- Started as an ATLAS/LHCb project
- Ganga is an application that enables a user to

Configure – Prepare – Submit – Monitor

jobs and applications using a variety of resources

The Ganga Mantra:

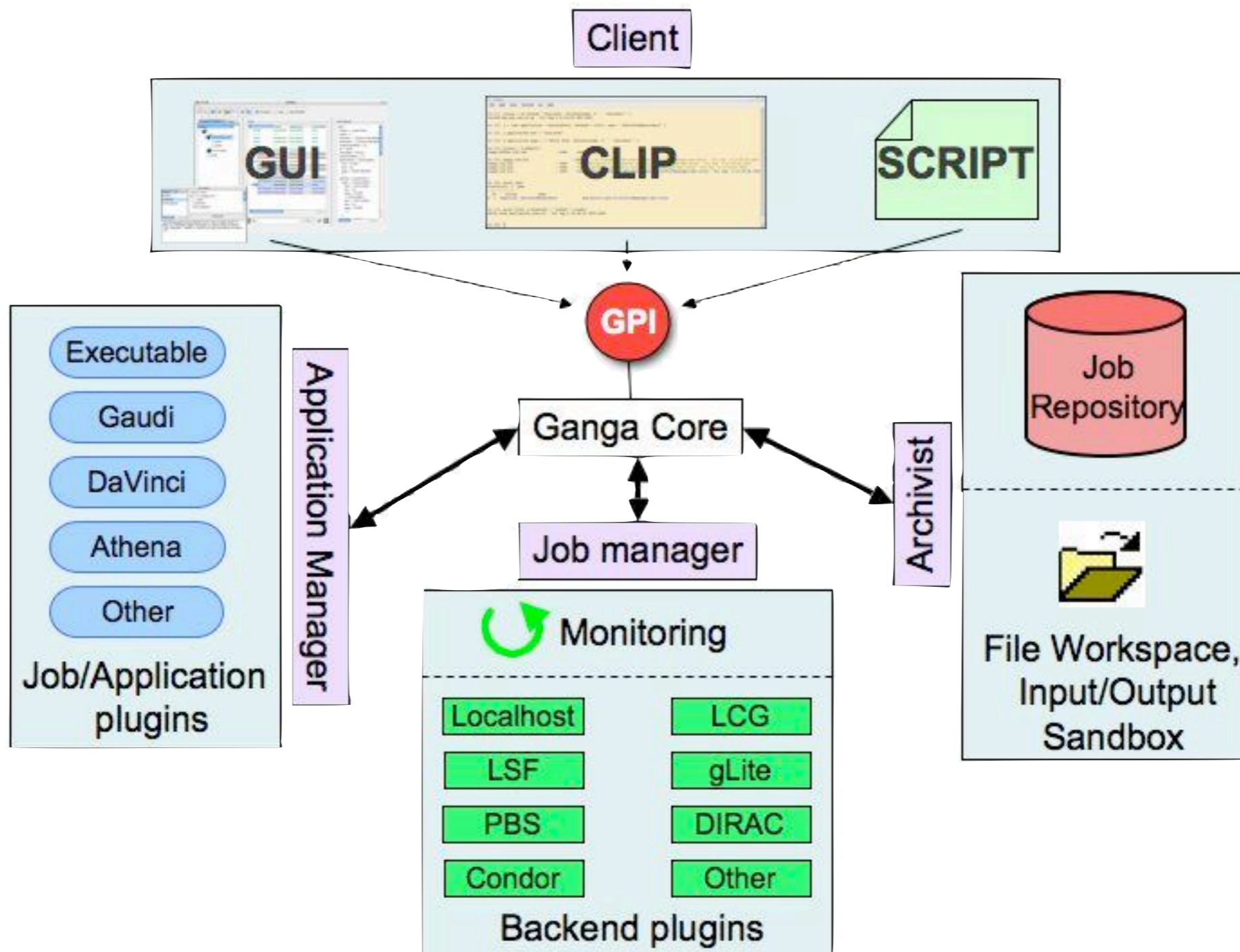


Configure once, run anywhere

Possible resources

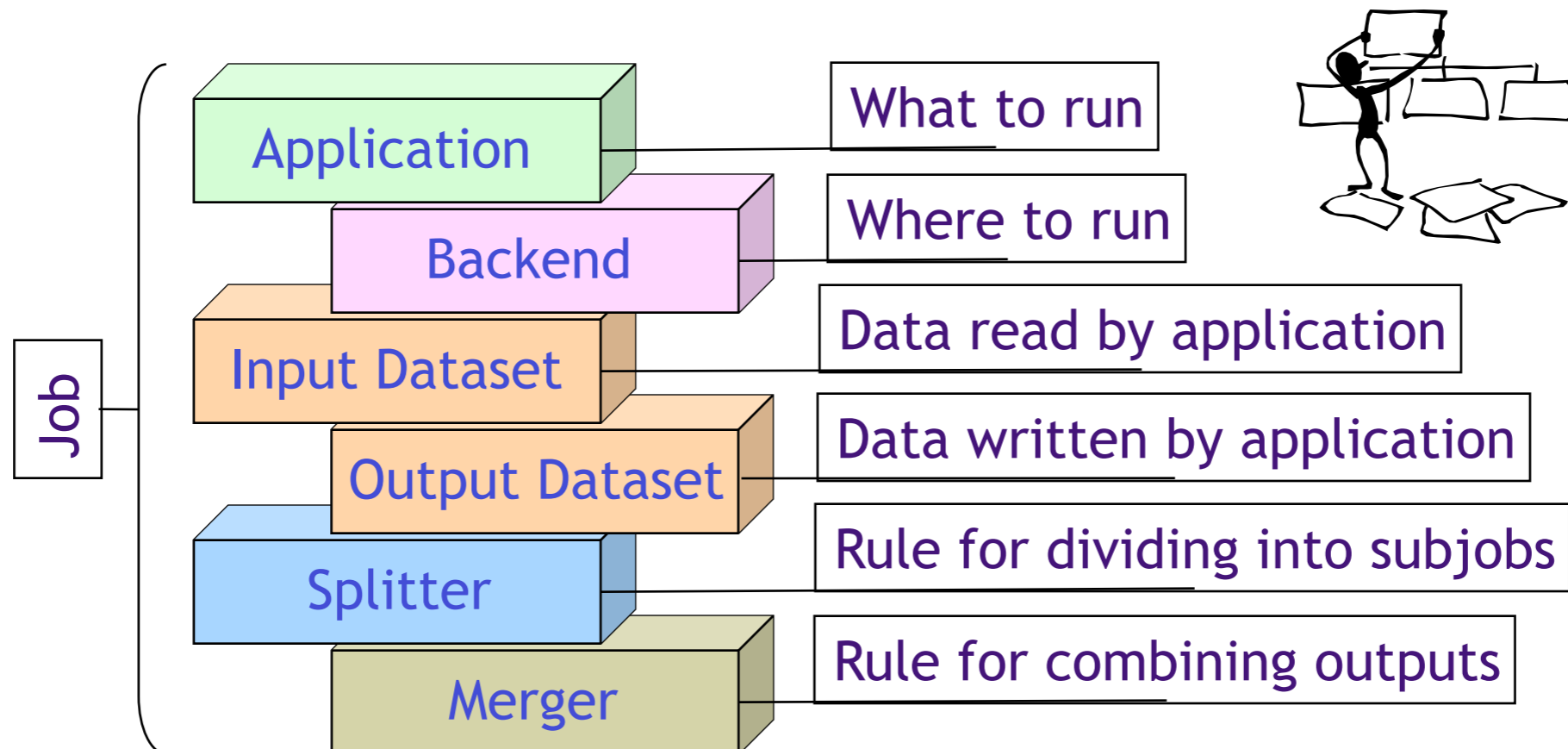
- ☺ The **local machine** (interactive or in background)
- ☺ **Batch** systems (LSF, PBS, SGE, Condor)
- ☺ **Grid** systems (LCG, gLite, NorduGrid)
- ☺ **Workload management** systems (DIRAC, PanDA)
- ☺ Jobs look the same whether they run locally or on the Grid

Ganga Architecture



Ganga Job Object

- 👤 A job in Ganga is constructed from a set of building blocks, not all required for every job



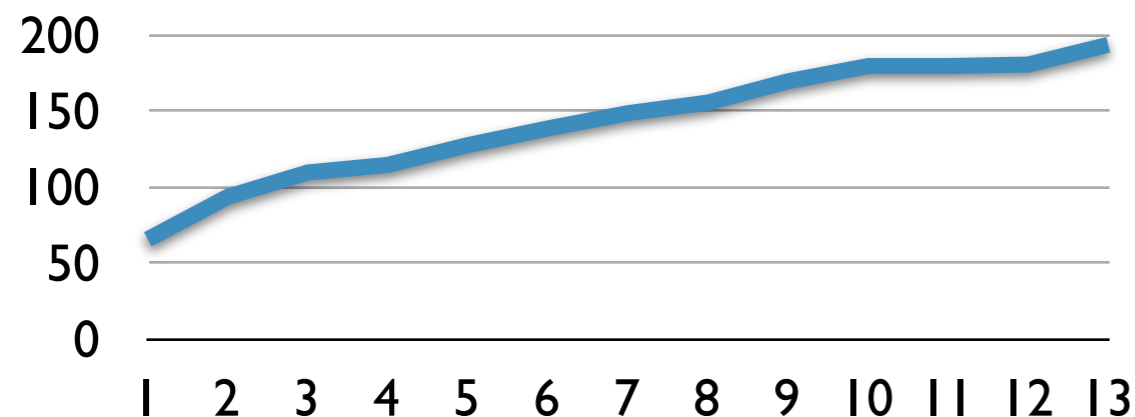
Ganga Usage in LHCb



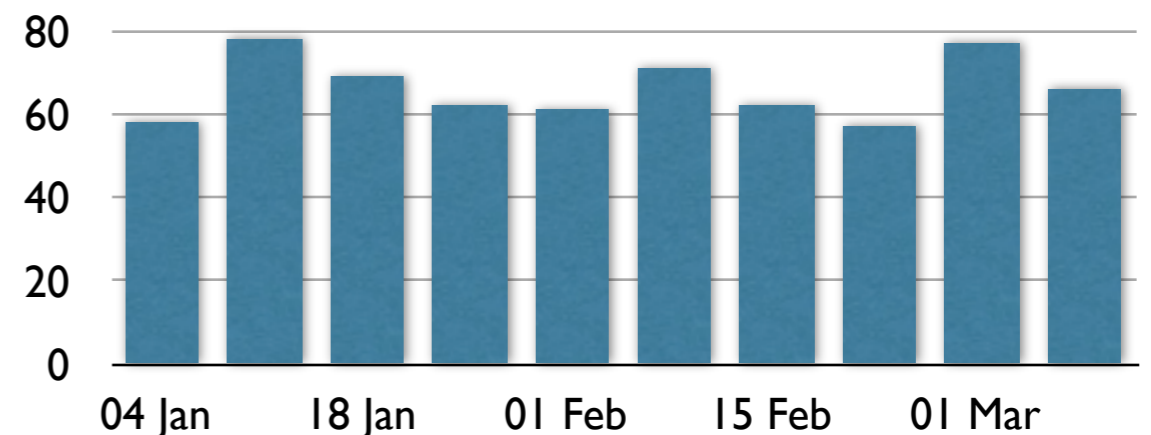
Ganga Usage in LHCb

- ☺ In the last 3 months ~200 unique LHCb users
 - ☺ Stably increasing trend of users over the last 3 years
 - ☺ more users than originally expected physicists from the TDR
 - ☺ TDR considers 140 users
- ☺ >60 physicists use Ganga on average per week

Accumulated unique users vs time in weeks



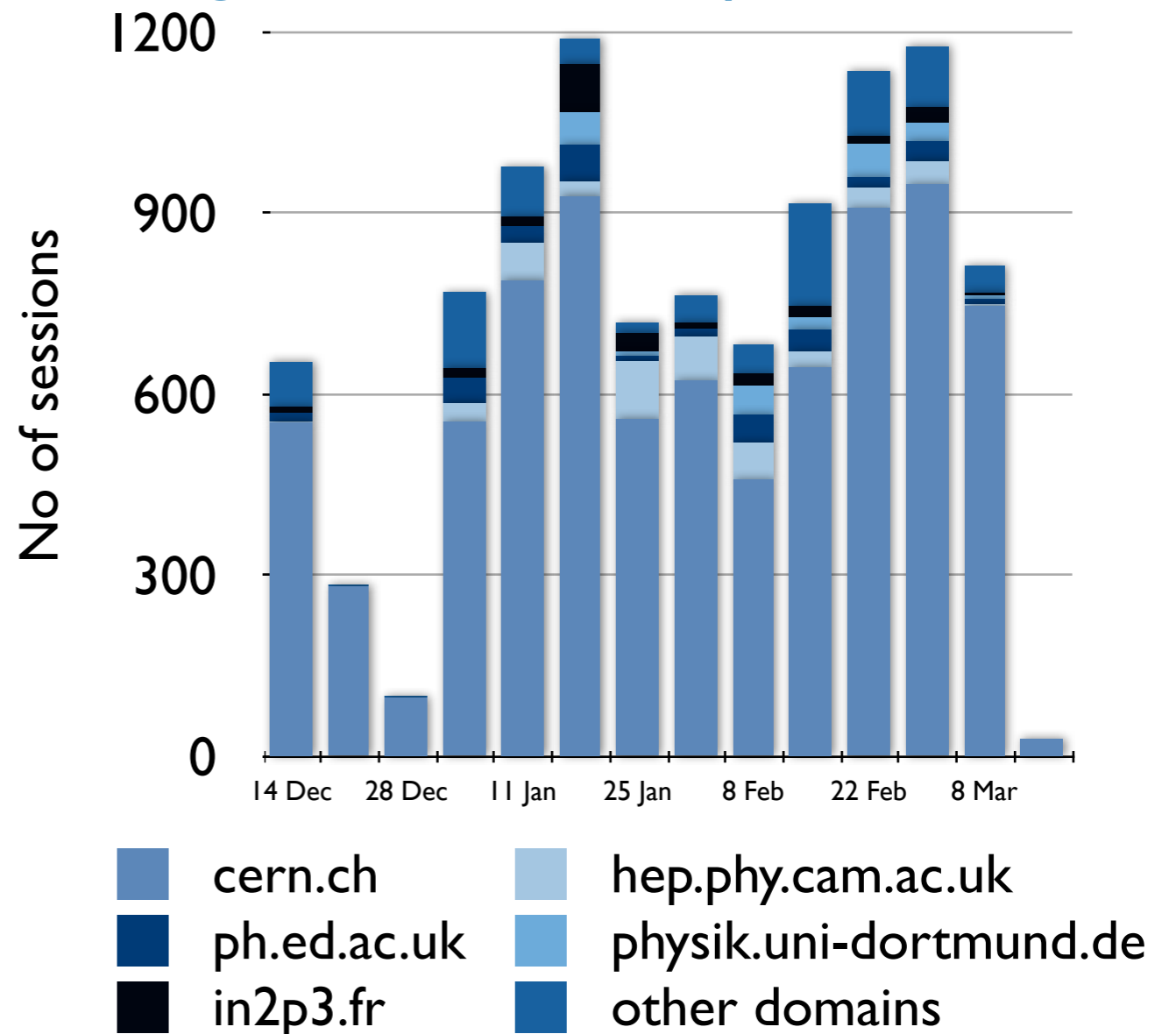
Unique users in LHCb per week in 2009



Ganga installations

- 🌀 Ganga reference installation at CERN
- 🌀 Hence ~80% use the tool at CERN
- 🌀 Several installations are at LHCb institutes
- 🌀 ~20% of LHCb usage

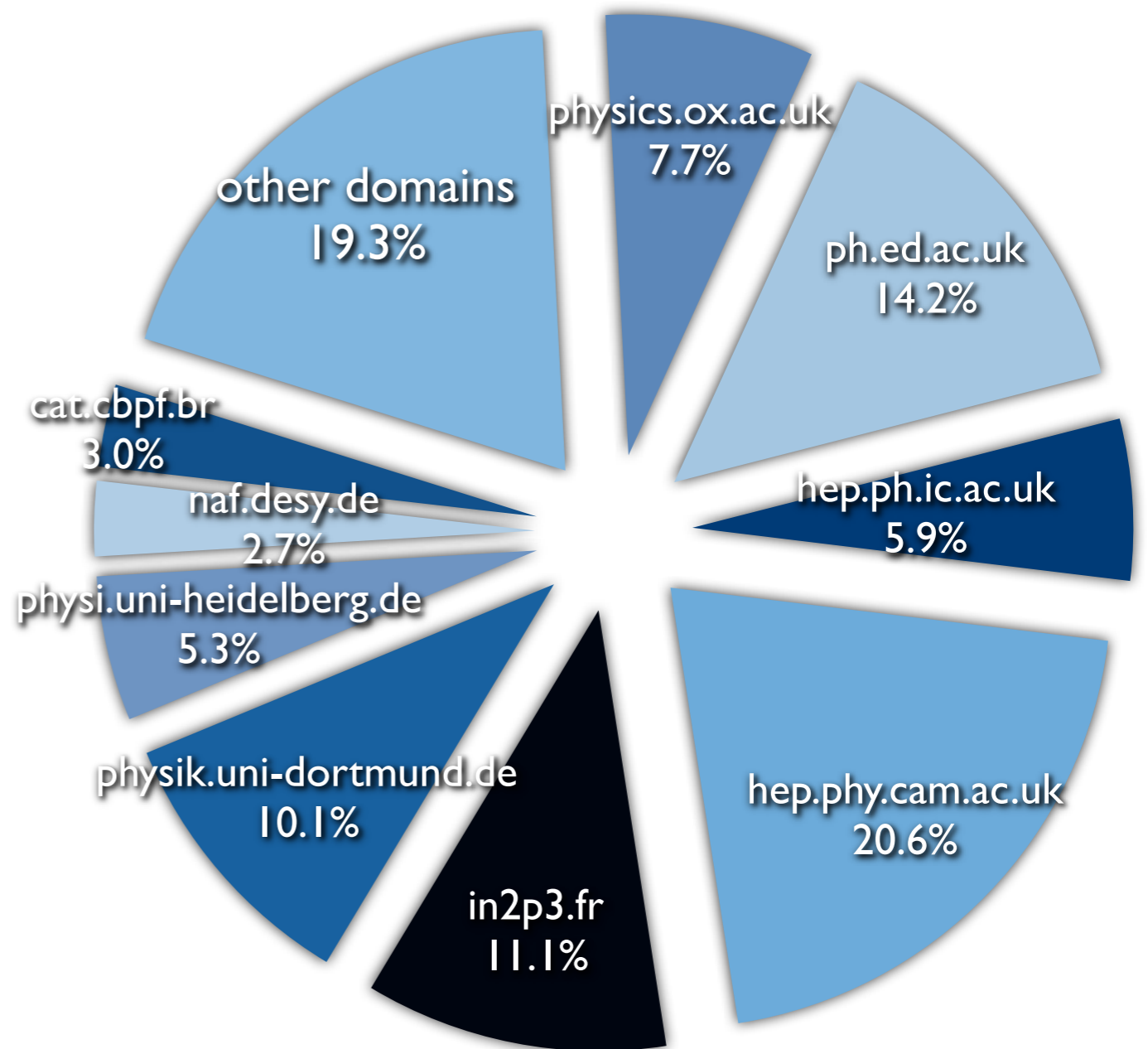
Ganga LHCb sessions by week and domain



Non CERN Ganga installations

- Several non-CERN installations used in LHCb
- Active UK installations (~50%)
- Followed by German (~20%) and French institutes (~11%)

Ganga Sessions of Non-CERN Ganga sites in LHCb





LHCb Plug-ins



LHCb plugins

LHCb specific application handlers

-  All LHCb applications are based on the Gaudi framework
-  Includes specialised splitters

General purpose ROOT handler

-  used for Toy Monte-Carlos (ROOFit)

LHCb specific submission backend

-  The only supported Grid backend is the DIRAC 3 WMS
-  Direct LCG submission of Gaudi jobs not supported



Data Selection

- Graphical bookkeeping browser in Ganga
- Allows selection of data sets from the official LHCb bookkeeping system
- Results in a list of input files, entered into an LHCbDataset object

The screenshot shows the 'Feicim - LHCb Bookkeeping browser' window. It features a 'File' menu, a 'Standard' view mode, and an 'Advanced Queries' checkbox. The main area is divided into a 'Tree' view on the left and a 'Description' table on the right. The tree view shows a hierarchy starting with 'MC', followed by 'Configuration versions', '2007', '2008', '2008-HT', 'Simulation Conditions/DataTaking', 'Beam7TeV-VeloClosed-MagDown', 'Processing Pass', 'GeneratorRequests', 'Event types', and 'File types'. The 'DIGI' folder is selected, showing a 'Nb of Files/Events' of 10/100000. Below the tree, a list of files is shown with their descriptions, including 'SIM' files and '51000001' through '51210005'. The 'Queries' section at the bottom has two radio buttons: 'SimCond/ProcessingPass/Eventtype/Production/FileType/Program/Files' (selected) and 'Event type/SimCond/ProcessingPass/Production/FileType/Program/Files'.

Tree	Description
MC	
Configuration versions	
2007	
2008	
2008-HT	
Simulation Conditions/DataTaking	
Beam7TeV-VeloClosed-MagDown	
Processing Pass	
GeneratorRequests	
Event types	
51000001	e-,fixP=CaloAcc 1*GeV
51000005	e-,fixP=CaloAcc 5.0*GeV
51000010	e-,fixP=CaloAcc 10.0*GeV
File types	
DIGI	
Nb of Files/Events	10/100000
SIM	
51000017	e-,fixP=CaloAcc 16.8*GeV
51000034	e-,fixP=CaloAcc 33.8*GeV
51000050	e-,fixP=CaloAcc 50.0*GeV
51000100	e-,fixP=CaloAcc 100.0*GeV
51000168	e-,fixP=CaloAcc 168.0*GeV
51210001	e+e-,fixP=TrkAcc 1.0*GeV
51210005	e+e-,fixP=TrkAcc 5.0*GeV

LHCbDatasets

-  A list of files (PFNs or LFNs)
-  Can cache replica information for faster resubmission of jobs with the same input data

```
LHCbDataset (  
  cache_date = 'Wed Aug 29 23:49:04 2007' ,  
  files = [ LHCbDataFile (  
    name = 'LFN:/lhcb/production/DC06/phys-v2-  
lumi2/00001889/DST/0000/00001889_00000003_5.dst' ,  
    replicas = ['IN2P3-disk', 'CERN-disk']  
  ) ,  
    ...]  
)
```

LHCb Analysis Job

Gaudi based applications:

Example DaVinci (User analysis)

```
In [2]: app=DaVinci()
```

```
In [3]: print app
```

```
Out[3]: DaVinci (
  extraopts ← None ,
  package = 'Phys' ,
  configured = None ,
  masterpackage = None ,
  platform = 'slc4_ia32_gcc34' ,
  version ← 'v22r1' ,
  setupProjectOptions = '' ,
  user_release_area = '/afs/cern.ch/user/a/andrew/cmtuser' ,
  optsfile ← [File(name='/afs/cern.ch/user/a/andrew/cmtuser/
DaVinci_v22r1/myopts2/myKstarMuMu_writeNtuple.opts', subdir='.')]
)
```

Specify extra configuration properties appended to the options

Specify the application version

Specify the options files to be used

Ganga Benefits

Configures environment

-  sets up the environment in a separated area

Parses options files to extract

-  input data • output data • histograms

Determines Input Sandbox

-  user libraries • preprocessed options files

Splitting and Merging

Easy splitting of jobs

```
j.splitter=DiracSplitter(filePerJob=3)
print j.splitter
DiracSplitter (
  filesPerJob = 3 ,
  maxFiles = -1 ,
  ignoremissing = False
)
```

DiracSplitter takes replica information into account (Guarantees subjobs with all data at one site)

Automatic merging of results

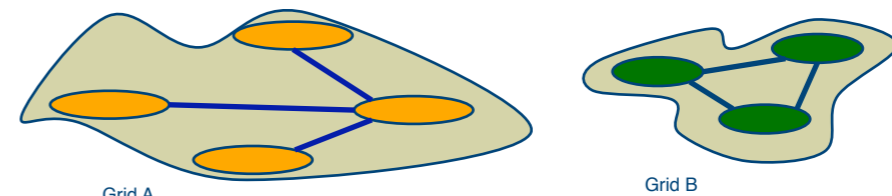
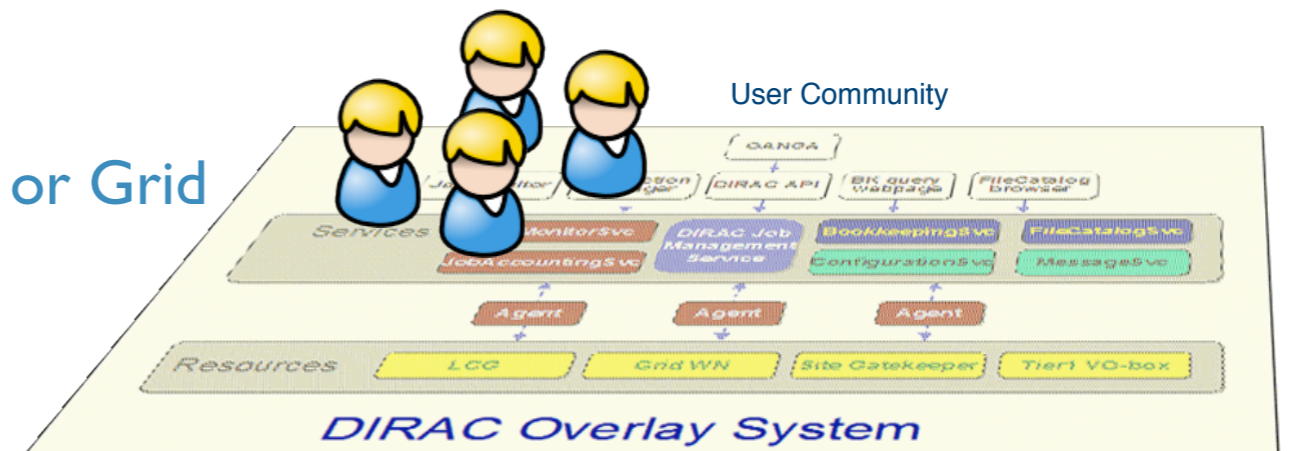
```
j.merger=RootMerger(files=['myhisto.root'])
print j.merger
RootMerger (
  files = ['histo.root'] ,
  ignorefailed = False ,
  overwrite = False
)
```

LHCb Supported Backends

- Submission to all local batch systems available
- Internal look-up for LFN to PFN resolution available
- Same job can run on local batch or Grid

Dirac Interface

- Specific to LHCb
- Only entry to the Grid for LHCb users
- Adds reliability and consistency



DIRAC WMS

- 👁️ DIRAC uses the concept of Pilot Agents.
- 👁️ Allows to build a robust and efficient Workload Management Systems (WMS)
- 👁️ Job prioritisation in a central Task Queue
- 👁️ Implementation of the community policies

DIRAC plugin

- DIRAC was originally designed as the LHCb production system.
 - So originally the backend allowed only submission of Gaudi Jobs
- DIRAC now supports all applications supported by Ganga, e.g. Gaudi, Root and the generic executable

```
backend = Dirac (  
    status = 'Completed' ,  
    actualCE = 'LCG.Barcelona.es' ,  
    statusInfo = 'Execution Complete' ,  
    id = 904010 ,  
    CPUtime = 86400  
)
```

Read-only
status
information

Set CPU time
needed for job

Output handling

- ☺ Outputs are automatically discovered for Gaudi applications
- ☺ Ganga distinguishes between
 - ☺ Sandbox data (returned with the job)
 - ☺ Output data (returned to some large storage area, e.g. CASTOR, or Grid SE)
- ☺ DIRAC returns large sandbox data to Grid storage automatically

Conclusions

- 👁️ Ganga is a well established and accepted user interface for job submission to batch and grid systems in LHCb
- 👁️ Ganga now reaches all analysis users in LHCb
- 👁️ >60 users per week in LHCb use Ganga to perform their analysis
- 👁️ A job repository helps to organise the workflow of the physicist-user
- 👁️ Specialised LHCb plug-ins simplify the submission of complex jobs with support for easy set-up of LHCb applications, splitting and merging
- 👁️ The LHCb specific DIRAC plug-in allows submission of jobs to the DIRAC WMS