

DISTRIBUTED ANALYSIS TOOLS IN ATLAS

Johannes Elmsheuser

Ludwig-Maximilians-Universität München, Germany

30 Nov 2006/D-Grid HEP-CG Workshop



- ① DISTRIBUTED ANALYSIS IN ATLAS
- ② INTRODUCTION TO GANGA
- ③ DEMO: GANGA JOB IN A DISTRIBUTED ENVIRONMENT
- ④ CONCLUSIONS

- ① DISTRIBUTED ANALYSIS IN ATLAS
- ② INTRODUCTION TO GANGA
- ③ DEMO: GANGA JOB IN A DISTRIBUTED ENVIRONMENT
- ④ CONCLUSIONS

ATLAS GRID INFRASTRUCTURE

- Heterogeneous grid environment based on 3 grids:
LCG/EGEE, OSG and Nordugrid



- Grids have different middle-ware, catalogs to store data, software to submit jobs

→ Hide differences from the ATLAS user

DISTRIBUTED ANALYSIS MODEL

The distributed analysis model is based the LHC computing model

- Data is distributed in Tier1/Tier-2 facilities by default
available 24/7
- user jobs are sent to the data
large input datasets (100 GB up to several TB)
- Results must be made available to the user
potentially already during processing
- Data is added with meta-data and bookkeeping in catalogs

→ Distributed Data Management

- Automated file management, distribution and archiving throughout the whole grid
- Random access needs a pre-filtering of data of interest
e.g. Trigger or ID streams or TAGs

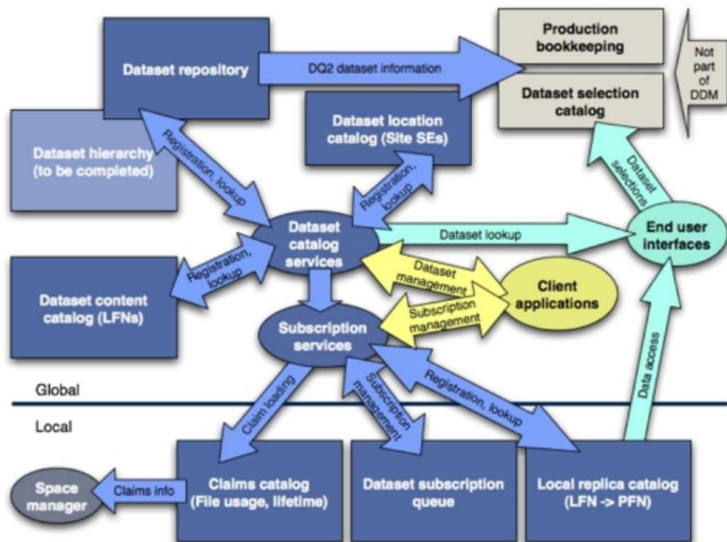
DISTRIBUTED DATA MANAGEMENT IN ATLAS (DDM) I



Don Quijote 2 (DQ2):

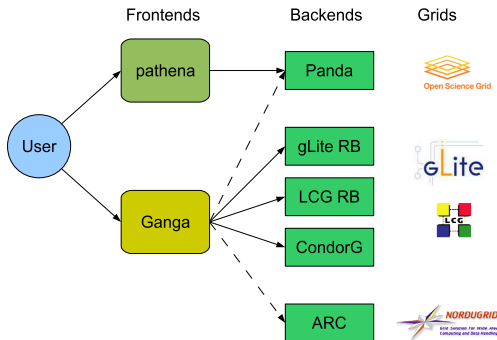
- Management of data flows and single entry point to all distributed ATLAS data
- Move data between grid sites using fts (preferred) or srm-cp
- Query and Retrieval of data
- Data is grouped into datasets, based on meta-data, like run period etc.
- Three components: Central Dataset Catalog, Site Subscription Service and Client Tools

DISTRIBUTED DATA MANAGEMENT II



DISTRIBUTED ANALYSIS STRATEGY

ATLAS offers several ways to do distributed analysis:



- Data from Production System is currently consolidated by DDM-operations team on a few sites: CERN, Lyon, FZK, BNL
- Analysis model implemented by PAT team foresees Athena analysis of AODs/ESDs and interactive use of Athena-aware-ROOT tuples

MIDDLE-WARE FEATURES

LCG/gLite (EGEE):

- Job submission via LCG Resource broker
- Fast bulk submission with new gLite RB, which is still to get in production...
- LFC File catalog

OSG/Panda:

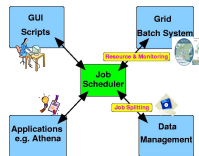
- Panda is an integrated production and distributed analysis system
- Pilot job based and similar to Dirac & Alien
- Simple file catalogs at sites

Nordugrid:

- ARC middle-ware for job submission
- RLS file catalog
- Only production, no distributed analysis yet

JOB SCHEDULER SPECIFICATION

Analysis with long response times
and low level of user interaction
→ Job scheduler (GANGA)



Results from the Gap Analysis:

- Interface for job configuration
- Job submission interface for Grid and Batch systems
- Integration of data management
- Resource estimation
- Job monitoring
- Job error checking
- Collecting and merging of the results
- Job archive

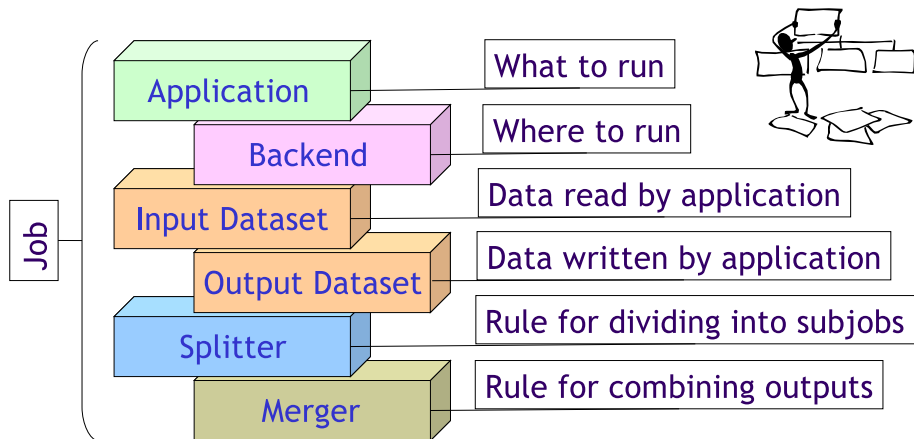
- ① DISTRIBUTED ANALYSIS IN ATLAS
- ② INTRODUCTION TO GANGA
- ③ DEMO: GANGA JOB IN A DISTRIBUTED ENVIRONMENT
- ④ CONCLUSIONS

FRONT-END CLIENT: GANGA

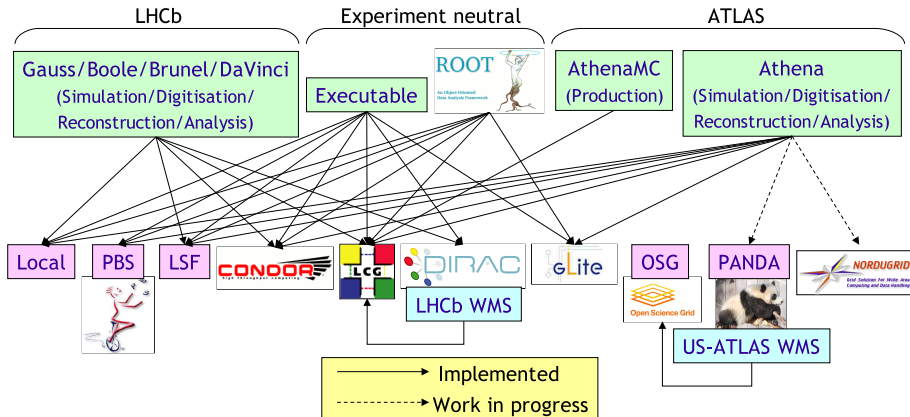
- A **user-friendly** job definition and management tool.
- Allows simple switching between testing on a **local batch system** and large-scale data processing on distributed resources (**Grid**)
- Developed in the context of **ATLAS** and **LHCb** :
 - For ATLAS, have built-in support for applications based on Athena framework, for JobTransforms, and for DQ2 data-management system
- **Component** architecture readily allows extension
- Python framework
- **Development team**: F.Brochu (Cambridge), U.Egede (Imperial), J.Elmsheuser (München), K.Harrison (Cambridge), H.C.Lee (ASCC), D.Liko (CERN), A.Maier (CERN), J.T.Moscicki (CERN), A.Muraru (Bucharest), V.Romanovsky (IHEP), A.Soroko (Oxford), C.L.Tan (Birmingham) and contributions past and present from many others

GANGA II

- Ganga is based on a simple, but flexible, job abstraction
- A job is constructed from a set of building blocks, not all required for every job



- Ganga simplifies running of ATLAS (and LHCb) applications on a variety of Grid and non-Grid back-ends



GANGA IV: MUNICH CONTRIBUTIONS

- Gap analysis within D-Grid: Identify job scheduler candidates and review needs and existing features
- Currently maintaining and devolping the Athena plug-in for analysis and initiated the AthenaMC plug-in for MC generation with ProdSys JobTransforms
- Athena job splitting and DQ2/DDM Integration for input and output
- Direct access to data files via Posix I/O
- Numerous additional features for Athena: shared inbox, local datasets, output download and merging, better error reporting, ..., Condor-G, ...
- Wiki documentation, user support and several tutorials

- Improve DQ2 integration: dataset matching, dataset output, bookkeeping
- Improve job monitoring
- Error recovery
- DIANE (<http://cern.ch/diane>) integration ?
Parallel job execution in master/worker mode
Offers immediate job results and response

- ① DISTRIBUTED ANALYSIS IN ATLAS
- ② INTRODUCTION TO GANGA
- ③ DEMO: GANGA JOB IN A DISTRIBUTED ENVIRONMENT
- ④ CONCLUSIONS

JOB DEFINITION USING THE ROOT PLUGIN

A generic ROOT macro (`$ROOTSYS/tutorials`) can be configured with Ganga.

Job definition within GANGA IPython shell:

```
j = Job()
j.name='ROOT mlpHiggs'
j.application = Root()
j.application.script = '$HOME/demo/mlpHiggs.C'
j.splitter = ArgSplitter()
j.splitter.args = [[10],[50],[100]]
j.inputsandbox = ['$HOME/demo/mlpHiggs.root']
j.outputsandbox = ['mlpOutput.root']
j.backend=LCG()
j.submit()
```

GANGA WORK FLOW FOR ATLAS JOBS I

Ganga **job configuration** and **monitoring** :

- bash shell, IPython command line, GUI (or dashboard)

Data:

- Data is organized in **datasets** with sw version, physics process (MC) as list of files

Job configuration:

- User specifies as **job parameters**: source code area, sw release version, dataset, jobOptions

Ganga Job submission:

- Query DQ2 database, wrap-up source code, submit to LCG/gLite RB
- Job is **sent to the data** with list of filename and guids and recompiled (optional not) against ATLAS SW distribution kit

GANGA WORK FLOW FOR ATLAS JOBS II

Data access:

- Transform GUID→SURL→TURL with help of local LFC catalog
- Preferred: direct access from worker node to SE via dcap/rfio
- Optional: download input files to temp area of worker node

Job output:

- Output is registered on local SE or e.g. CERN castor
- User receives log files back
- User can download output after job termination

Job monitoring:

- Via Resource broker in IPython and GUI (little details)
- Additional job progress report in ARDA dashboard with RB info (still in the integration phase)

ATLAS JOB DEFINITION I

- Job definition from command line on local desktop:

```
athena AnalysisSkeleton_topOptions.py
```

- Job definition from command line to the GRID:

```
ganga athena \  
  --inDS csc11.005320.PythiaH170wll.recon.A0D.v11004107 \  
  --outputdata AnalysisSkeleton.aan.root \  
  --split 3 \  
  --lcg \  
  --ce ce-fzk.gridka.de:2119/jobmanager-pbspro-atlasS \  
  AnalysisSkeleton_topOptions.py
```

ATLAS JOB DEFINITION II

Job definition within GANGA IPython shell

```
j = Job()
j.name='5145'
j.application=Athena()
j.application.prepare()
j.application.option_file='$HOME/athena/12.0.31-ana/InstallArea/job0
j.splitter=AthenaSplitterJob()
j.splitter.numsubjobs = 3
j.merger=AthenaOutputMerger()
j.inputdata=DQ2Dataset()
j.inputdata.type='DQ2_LOCAL'
j.inputdata.dataset='csc11.005145.PythiaZmumu.recon.A0D.v11004103'
j.outputdata=DQ2OutputDataset()
j.outputdata.outputdata=['AnalysisSkeleton.aan.root']
j.backend=LCG()
j.backend.CE='ce-fzk.gridka.de:2119/jobmanager-pbspro-atlasS'
j.submit()
```

MONITORING WITH THE IPYTHON SHELL

```

elmsheus@lxplus014:~ - Befehlsfenster - Konsole
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe

Gange.GPDev.Lib.Job : INFO killing job 34
Gange.GPDev.Lib.Job : INFO killing job 34.0
Gange.GPDev.Lib.Job : INFO job 34.0 status changed to "killed"
Gange.GPDev.Lib.Job : INFO killing job 34.1
Gange.GPDev.Lib.Job : INFO job 34.1 status changed to "killed"
Gange.GPDev.Lib.Job : INFO killing job 34.2
Gange.GPDev.Lib.Job : INFO job 34.2 status changed to "killed"
Gange.GPDev.Lib.Job : INFO job 34 status changed to "killed"
Out [2]: 1

In [3]:

In [3]:execfile('/afs/cern.ch/user/e/elmsheus/split/5320_cern.py')
Gange.GPDev.Lib.Job : INFO submitting job 35
Gange.GPDev.Adapters : INFO submitting job 35.0 to LOG backend
Gange.GPDev.Adapters : INFO submitting job 35.1 to LOG backend
Gange.GPDev.Adapters : INFO submitting job 35.2 to LOG backend

In [4]:

In [4]:jobs
Out [4]: Statistics: 25 jobs

-----
# id status name subjobs application backend backend.actualCE
# 6 completed Executable LCG ce-fzk.gridka.de:2119/jobmanager-pbspro-atlas
# 7 new Executable LCG
# 8 new Executable LCG
# 9 completed Executable LCG grid10.lal.in2p3.fr:2119/jobmanager-pbs-atlas
# 10 completing Executable LCG ce-fzk.gridka.de:2119/jobmanager-pbspro-atlas
# 12 completed Athena LSF
# 13 new Executable Local
# 14 completed Athena LSF
# 15 completed 2 Athena LCG
# 16 completed Athena LSF
# 17 completed athena_log_dq2_test1, GRIDKA, DQ2_LOCAL, 7233 Athena LOG ce-fzk.gridka.de:2119/jobmanager-pbspro-atlas
# 18 new athena_log_dq2_test2, LYON, DQ2_LOCAL, 5300 Athena LCG
# 19 completed athena_log_dq2_test2, LYON, DQ2_LOCAL, 5300 Athena LCG cclegoel102.in2p3.fr:2119/jobmanager-bqs-shor
# 20 completed Athena LSF
# 22 submitted 5320 3 Athena LCG
# 24 submitted 5320 10 Athena LCG
# 26 completed 5145_short 2 Athena LCG
# 27 completed 5300 3 Athena LCG
# 28 new 5300 Athena Local
# 29 completed 5300 3 Athena LCG
# 30 killed 5145 10 Athena LCG
# 31 completed 5145_short 5 Athena LCG
# 32 running 5300 3 Athena LCG
# 34 killed 5320_cern 3 Athena LCG
# 35 submitted 5320_cern 3 Athena LCG

In [5]:

```

MONITORING WITH THE GUI

The screenshot shows the GANGA GUI window titled "GANGA <@xplus014.cern.ch>". The interface includes a menu bar (Job, Folders, View, Help), a toolbar with icons for file operations, and a status bar at the bottom with a "Find" button.

The main window is divided into two panes. The left pane, titled "Jobs", displays a table of job information. The right pane, titled "Job Details", shows the configuration for the selected job (ID 34).

Jobs Table:

id	status	name	application	exe filename	backend
7	completed		Executable	close.sh	LCG
8	new		Executable	close.sh	LCG
9	completed		Executable	close.sh	LCG
10	completing		Executable	close.sh	LCG
12	completed		Athena	*N/A*	LSF
13	new		Executable	echo	Local
14	completed		Athena	*N/A*	LSF
15	completed		Athena	*N/A*	LCG
16	completed		Athena	*N/A*	LSF
17	completed	athena_lcg_dq2_test1, GRIDKA, DQ2_LOCAL, 7233	Athena	*N/A*	LCG
18	new	athena_lcg_dq2_test2, LYON, DQ2_LOCAL, 5300	Athena	*N/A*	LCG
19	completed	athena_lcg_dq2_test2, LYON, DQ2_LOCAL, 5300	Athena	*N/A*	LCG
20	completed		Athena	*N/A*	LSF
22	submitted	5320	Athena	*N/A*	LCG
24	submitted	5320	Athena	*N/A*	LCG
26	completed	5145_short	Athena	*N/A*	LCG
27	completed	5300	Athena	*N/A*	LCG
28	new	5300	Athena	*N/A*	Local
29	completed	5300	Athena	*N/A*	LCG
30	new	5145	Athena	*N/A*	LCG
31	completed	5145_short	Athena	*N/A*	LCG
32	running	5320	Athena	*N/A*	LCG
34	submitted	5320_cern	Athena	*N/A*	LCG

Job Details (Job 34):

```

Job {
  status = 'submitted',
  name = '5320_cern',
  inputdir = '/afs/cern.ch/user/e/elsmsheus/gangadir/workspace/Local/34/input',
  outputdir = '/afs/cern.ch/user/e/elsmsheus/gangadir/workspace/Local/34/output',
  outputsandbox = [],
  id = 34,
  inputdata = DQ2Dataset (
    tagdataset = '',
    type = 'DQ2_LOCAL',
    names = [],
    match_ce = False,
    dataset = 'csc11.005320.PythiaH170wvllrecon.AOD.v11004107'
  ),
  merger = None,
  inputsandbox = [],
  application = Athena (
    alias_release = '11.0.5',
    max_events = None,
    options = None,
    user_setupfile = File (
      name = '',
      subdir = ''
    ),
    option_file = File (
      name = '/afs/cern.ch/user/e/elsmsheus/athena/testarea/11.0.5/PhysicsAna',
      subdir = ''
    ),
    user_area = File (
      name = '/afs/cern.ch/user/e/elsmsheus/gangadir/workspace/Local/UserAr',
      subdir = ''
    )
  ),
  outputdata = DQ2OutputDataset (
    output = [],
    datasetname = ''
  )
}
    
```


- ① DISTRIBUTED ANALYSIS IN ATLAS
- ② INTRODUCTION TO GANGA
- ③ DEMO: GANGA JOB IN A DISTRIBUTED ENVIRONMENT
- ④ CONCLUSIONS

DISTRIBUTED ANALYSIS NEEDS AND CONCLUSIONS

The generic grid job submission framework Ganga can be used with DQ2/DDM to perform Distributed Analysis in ATLAS

Presented Overview about Distributed Analysis in ATLAS and the front-end Ganga:

<http://cern.ch/ganga/>

<https://twiki.cern.ch/twiki/bin/view/Atlas/DistributedAnalysisUsingGanga>

For the distributed analysis it is vital to have:

- Easy interface that does not scare off physicists
- A reliable and robust service of:
 - ATLAS SW Distribution Kit installation
 - Dataset replication
 - Local LFC catalog access and TURL resolving
 - Local Posix access to SE
 - Output storage
 - Job monitoring