Distributed Analysis using GANGA on the EGEE/LCG infrastructure

Johannes Elmsheuser

Ludwig-Maximilians-Universität München, Germany

03 September 2007/CHEP 2007
1 Distributed Analysis Model in ATLAS

2 Distributed Analysis with GANGA

3 Conclusions
Heterogeneous grid environment based on 3 grid infrastructures:

- egee
- Open Science Grid
- NORDUGRID

Grids have different middle-ware, replica catalogs and tools to submit jobs

⇒ Hide differences from the ATLAS user
The distributed analysis model is based the ATLAS 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
• Event Filter Farm at CERN
  • Located near the Experiment, assembles data into a stream to the Tier0

• Tier0 at CERN
  • Raw data → Mass storage at CERN and to Tier 1s
  • Swift production of Event Summary Data (ESD) and Analysis Object Data (AOD)
  • Ship ESD, AOD to Tier1s → Mass storage at CERN

• Tier1s distributed worldwide (10 centers)
  • Re-reconstruction of raw data, producing new ESD, AOD
  • Scheduled, group access to full ESD and AOD

• Tier2s distributed worldwide (∼ 30 centers)
  • MC Simulation, producing ESD, AOD → Tier 1s
  • On demand user physics analysis

• CERN Analysis Facility
  • Analysis
  • Heightened access to ESD and RAW/calibration data on demand

• Tier3s distributed worldwide
Distributed Analysis Model II

Need for: Distributed Data Management (DDM)

- Managed by DDM system DQ2 (Don-Quijote 2)
- Automated file management, distribution and archiving throughout the whole grid using a Central Catalog, FTS, LFCs
- Random access needs a pre-filtering of data of interest e.g. Trigger or ID streams or TAGs (event-level meta data)

Current situation and implementation

- Data from MC Production System is currently consolidated by DDM-operations team on all Tier1 and then all Tier2 sites
- Analysis model foresees Athena analysis of AODs/ESDs and interactive use of Athena-aware-ROOT tuples
- Analysis tuple format(s) in enhancement
Naive assumption: Grid $\approx$ large batch system

- Provide complicated job configuration jdl file (Job Description Language)
- Find suitable Athena software, installed as distribution kits in the Grid
- Locate the data on different storage elements
- Job splitting, monitoring and book-keeping
- etc.

$\implies$ Need for automation and integration of various different components
How to combine all these: **Job scheduler/manager: GANGA**
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 Production System JobTransforms, and for DQ2 data-management system.
- Component architecture readily allows extension.
- Python framework.
Who is GANZA?

- Ganga is an ATLAS - LHCb joint project

- Support for development work from UK (PPARC/GridPP), Germany (D-Grid) and EU (EGEE/ARDA)

- Core team
  F.Brochu (Cambridge), U.Egede (Imperial), J. Elmsheuser (Munich), K.Harrison (Cambridge), H.C.Lee (ASGC Taipei), D.Liko (CERN), A.Maier (CERN), J.T.Moscicki (CERN), A.Muraru (Bucharest), W.Reece (Imperial), A.Soroko (Oxford), CL.Tan (Birmingham)

Ganga is now under GPL
GANGA Job

- 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

![Job Diagram]

- Application
- Backend
- Input Dataset
- Output Dataset
- Splitter
- Merger

- What to run
- Where to run
- Data read by application
- Data written by application
- Rule for dividing into subjobs
- Rule for combining outputs
GANGA Backends and Applications

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

LHCb
- Gauss/Boole/Brunel/DaVinci (Simulation/Digitisation/Reconstruction/Analysis)

Experiment neutral
- ROOT (An Object-Oriented Data Analysis Framework)
- AthenaMC (Production)

ATLAS
- Athena (Simulation/Digitisation/Reconstruction/Analysis)

Available in Ganga 4.3
- Work in progress
GANGA offers three ways of user interaction:

- Shell command line
- Interactive IPython shell
- Graphical User Interface

Job definition at command line for GRID submission:

```
ganga athena
   --inDS csc11.005320.PythiaH170wwll.recon.AOD.v11004107
   --outputdata AnalysisSkeleton.aan.root
   --split 3
   --lcg
AnalysisSkeleton_topOptions.py
```
Job definition within GANGLA IPython shell:

```python
j = Job()
j.application=Athena()
j.application.prepare(athena_compile=False)
j.application.option_file='$HOME/athena/12.0.5/InstallArea/jobOptions/UserAnalysis'
j.splitter=AthenaSplitterJob()
j.splitter.numsubjobs = 3
j.merger=AthenaOutputMerger()
j.inputdata=DQ2Dataset()
j.inputdata.dataset='csc11.005145.PythiaZmumu.recon.AOD.v11004103'
j.inputdata.match_ce=True
j.outputdata=DQ2OutputDataset()
j.outputdata.outputdata=['AnalysisSkeleton.aan.root']
j.backend=LCG()
j.submit()
```
**Job Workflow**

**User**
- User code
- Input Dataset
- Output files
- Some Grid specifics

**Ganga Client**
1. Athena environment parsing
2. DQ2 Database query
3. User Area tar ball creation
4. JDL file, InputSandBox creation
5. Job(s) submission to glite WMS
6. InputSandBox upload to remote SE if too large

**Grid Worker Node**
- Setup Athena Environment
- Inputfile List generation
- LFC Query
- Run Athena with direct POSIX I/O to dCache, Castor or DPM storage element
- Stage-out outputfiles to storage element

**User Actions**
- Monitor Jobs in Ganga Job Repository or ARDA Dashboard
- Output files download
- Output files merging
- Jobs resubmission

**Completion**
- After job completion:
  - Output Sandbox is retrieved
GANGA jobs monitored by the Dashboard

http://dashb-atlas-job.cern.ch/dashboard/request.py/jobsummary

JOB SUMMARY

<table>
<thead>
<tr>
<th>any user</th>
<th>any site</th>
<th>any vosite</th>
<th>any cloud</th>
<th>any tier</th>
<th>any ce</th>
<th>ganga</th>
<th>any dataset</th>
<th>any application</th>
<th>any rb</th>
<th>any activity</th>
<th>any grid</th>
<th>grid</th>
<th>any job</th>
<th>any user</th>
<th>any dataset</th>
<th>any application</th>
<th>any site</th>
<th>any vosite</th>
<th>any cloud</th>
<th>any tier</th>
<th>any ce</th>
<th>ganga</th>
<th>any dataset</th>
<th>any activity</th>
<th>any grid</th>
</tr>
</thead>
</table>

http://dashb-atlas-job.cern.ch/dashboard/request.py/jobsummary
GANGA usage on the Grid

- GANGA has been used in the last 4 month on LCG at:

<table>
<thead>
<tr>
<th>Tier</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage [%]</td>
<td>8</td>
<td>37</td>
<td>40</td>
<td>15</td>
</tr>
</tbody>
</table>

where at Tier1: 48 % Lyon and 36% FZK

⇒ After successful dataset replication, following the ATLAS user analysis computing model

- NorduGrid: backend for ARC submission is used, integrated with Athena and DQ2

- OSG/Panda: recently integrated a backend for OSG/Panda

⇒ Now all ATLAS Grid flavours are supported
Distributed Analysis Tutorials and Support

https://twiki.cern.ch/twiki/bin/view/Atlas/GangaTutorial43

- Edinburgh (February 1st-2nd)
- Milan (February 5th-6th)
- Lyon (March 5th-7th)
- Munich (March 29th-30th)
- Toronto (April 18th)
- Bergen (April 27th)
- Valencia (May 3rd-4th)
- DESY Zeuthen (Sep 18-19th)

- Ganga User support and Feedback via Hypernews
  hn-atlas-GANGAUserDeveloper@cern.ch
Usage Statistics

- Over 820 unique users since beginning of the year
- About 500 ATLAS users have tried Ganga at least once
- About 60 ATLAS Ganga users per week
• GANGA has been used at over 50 different sites in the last month
Recent Aspects of Distributed Analysis using GANGA

- Many groups are able to use GANGA for day to day work
- User feedback drives the development
- Increasing stability and functionality of the middle-ware, but still room for improvements

- Data distribution with complete datasets is a key issue
- Problems with setup during SLC4 migration
- Site specific problems need to be chased up
Distributed analysis needs and Conclusions

For the distributed analysis it is vital to have:

- Easy interface that does not scare off physicists
- A reliable and robust service of many components

Conclusions

- Growing number of users are using the Grid for analysis,
  Still some room to grow
- Data Management is a central issue for Distributed Analysis

Homepage:

- http://cern.ch/ganga

References:

- GANGA depends on many components - for details see talks this week