An introduction to the ATLAS software

James Catmore
Lancaster University, UK
Based on talks by Simon George and David Rousseau
The absolute basics

- **LXPLUS**: the CERN Linux cluster on which the ATLAS software is often run
- **AFS**: Andrews Filesystem - remote filesystem to which LXPLUS is connected
- **ONLINE software**: runs during the data taking (trigger, control, DAQ etc)
- **OFFLINE software**: processes data once it is committed to storage.
  - This tutorial is solely concerned with the offline software
- **ATHENA**: C++ control framework in which data processing and analysis is performed
  - ATHENA is an implementation of **Gaudi**
  - *Athena*: Greek Goddess of wisdom, war, the Arts, industry, justice and skill; sprang fully grown out of her father’s head (Zeus)
  - *Gaudi*: Antoni Gaudi, Barcelona architect 1852 - 1926, famed for the design of Barcelona's La Sagrada Familia, an immense Basilica which has been under construction since the beginning of the 20th century and is not scheduled for completion until 2026
Computer science stuff: what is a framework?

- A skeleton of an application into which developers plug in their code
- Provides most of the common functionality and communications between different components
- Embodies the underlying design and philosophy of the software
- Encourages a common approach
- Factors out common functionality for re-use
...which in practice means...

• The framework ensures your code
  ‣ runs at the right time
  ‣ with the right input data
  ‣ and correctly writes results to disk

• All of the offline processing is done in the Athena framework
  ‣ This flexibility comes at the expense of user-friendliness (unfortunately)
• **Algorithm**: an application - a piece of code that “does something”
  ‣ All algorithms inherit from the Algorithm class, which contains three methods:
    • `Initialize()` - run once at the start
    • `Execute()` - run n times
    • `Finalize()` - run once at the end
  ‣ Algorithms are invoked centrally by the framework
  ‣ Many algorithms can be run in a single job - one after the other
• **Data object**: result of an algorithm, or the input to it
  ‣ E.g. Track, Cluster, Muon, Electron, McEvent
• **Service**: globally available software entity which performs some common task
  ‣ Message printing
  ‣ Histogram drawing
• **Event**: a single pass of the `execute()` method, roughly corresponding to a physics event
• **JobOptions**: Python script which passes user instructions to Athena
  ‣ Which algorithms to run, what order, configuration
  ‣ Control of number of cycles, input/output files, runtime variables etc
Athena components (ii)

- **Tool**: piece of code that is shared between algorithms - it can be executed as many times as you need in the `execute()` method of your algorithms

- **Auditors**: software which monitors the other components of the framework

- **Sequence**: execution order of the algorithms

- **Filters**: software which allows or forbids an event from passing to the next algorithm in the sequence or being written to disk

- **Transient Store (StoreGate)**: service which stores results of algorithms (data objects) and passes them to the next algorithm.
  - The data is held in the computer memory

- **Persistent Store (POOL)**: format in which the data objects are written to disk

- **Converter**: software which enables the data objects used in the code to be written to and read from POOL without the details of the persistency being included in the objects themselves
Athena scheme (a bit less simplified)

- Tracker digits
- Tracks
- CaloCells
- Calorimeter Cells
- Calorimeter clusters
- Clusters
- Electron/photon
- Electrons/photons

StoreGate

Transient Event Data Store

Tracking

Calorimeter Clustering

Electron/photon Identification

Apparent dataflow
Real dataflow
Athena scheme (even less simplified)

- Reconstruction data flow (14.0.0)
Don’t panic!

- You don’t have to worry about most of the complications
- Physics analysis is the simplest part of the framework
- If you’re going to be working on a certain area of the software you’ll just concentrate on a few pieces of code, not the whole framework!
- It is still useful to hear about the full picture, so you have some idea of what the software is doing “under the bonnet”
Code management (i)

• All ATLAS offline code is stored in **SVN** which manages the evolution of the source files

• The basic unit is a **package**:
  
  ‣ each package is entirely defined by
    
    • its **name** and **path** - e.g. Reconstruction/RecExample/RecExCommon
    
    • its **tag** - e.g. RecExCommon-00-09-00

• The build of the binaries (compilation options, include files, libraries…) as well as the run-time environment are managed by a configuration management tool - **CMT**

  ‣ In my opinion this is the most complex bit of the ATLAS software

  ‣ Once you’ve got to grips with this, you’re in business!
Code management (ii)

- A new **release** of the code is entirely rebuilt (very) approximately every month (**developer** release)
- A **major** release is built roughly every six months
- Numbering system:
  - Major release
  - Developer release
  - Bug-fix release
- A release is completely defined by a list of package tags

**15.6.4**
Code management (iii)

• To ease maintenance, a release is split into projects e.g:
  ‣ *AtlasSimulation*: Event generators, detector simulation etc
  ‣ *AtlasAnalysis*: Analysis code and tools
  ‣ *AtlasCore*: core services and structures

• From a release a kit is built (with everything except source code) to export the software to other computing centres

• To allow quick patches/bug fixing, additional caches can be built on top of a release, e.g.:
  ‣ 14.5.0.2 *AtlasTier0*: for prompt reconstruction of data at Tier0 and for reprocessing at Tier1
  ‣ 14.4.0.1 *AtlasProduction*: for grid production

• Every night the release under construction is built (“nightlies”) and content kept for a week (useful only for developers of code)
More on packages

- A package = one library which is automatically loaded at run-time
  - The Athena executable itself is very small
- Packages contain algorithms that do related tasks
- Example: BPhysExamples
  - Example algorithms for B-physics studies
  - Structure (common to most packages):
    - PhysicsAnalysis/BPhys/BPhysExamples/src: C++ files
    - PhysicsAnalysis/BPhys/BPhysExamples/BPhysExamples: Header files
    - PhysicsAnalysis/BPhys/BPhysExamples/python: python configuration files
    - PhysicsAnalysis/BPhys/BPhysExamples/cmt: requirements and dependencies
    - PhysicsAnalysis/BPhys/BPhysExamples/share: job options
- Packages may depend on other packages
  - BPhysExamples depends on BPhysAnalysisTools and BPhysAnalysisObjects, etc
  - Depends = “uses an object defined in another package” = “uses header file found in other package”
  - “Depends” does not mean “needs other package to run first”
Main types of ATLAS data

- **Monte Carlo**
  - Event generator output

- **Digits/RAW**
  - Simulation/detector output

- **Event Summary Data (ESD)**
  - Output of reconstruction

- **Analysis Object Data (AOD)**
  - Summary of reconstruction - primary analysis data

- **Tag**
  - Thumbnail of each event used for identifying interesting events at the analysis stage

- **dESD, dAOD**
  - Data derived from ESD or AOD (more tomorrow)
The data processing chain

Detector & trigger → Reconstruction → AOD & TAG building

Monte Carlo

Digits → Digits → ESD

Physics analysis → dAOD/dESD building → Physics analysis

James Catmore
ATLAS offline software tutorial, CERN, 21st - 23rd October 2009
Physics analysis

Physics analysis can be done

- directly with Athena to produce n-tuples readable by ROOT
- directly with ROOT using Athena-ROOT-Access - then your AOD, ESD etc is your n-tuple

Analysis can be done in either framework

- on every event
- steered by the tag selection mechanism, thereby only selecting certain events
How should physics analyses proceed?

- Setup your environment using configuration manager, CMT
- Look for datasets to analyze using a metadata browser, AMI
- Download a few files locally using a data management tool, DQ2
- Inspect the files using Athena-ROOT-Access (ARA) or look at the events visually with ATLANTIS
- Build your analysis code in Athena or ARA using the local data to check that it is doing what you expect
- Send your jobs to the Grid to process large datasets
- Download the results
- Make histograms using ROOT

Today

Friday
## Where can I run ATLAS software?

<table>
<thead>
<tr>
<th>Where?</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>On LXPLUS</td>
<td>All set up for you; quickest way to start; ideal for testing</td>
<td>Very little quota and 20-minute limit interactive, very long delays with batch</td>
</tr>
<tr>
<td>At my institute</td>
<td>Lots of disk space; no limitations on execution time</td>
<td>You/your institute has to maintain it and pay for it</td>
</tr>
<tr>
<td>On the Grid</td>
<td>The <strong>only</strong> way of running over huge amounts of ATLAS data</td>
<td>A bit difficult at first; the machine you submit from must have ATLAS software</td>
</tr>
<tr>
<td>Laptop (installation)</td>
<td>Lots of disk space; no limitations on execution time</td>
<td>You have to set it all up - this can be a nightmare; you need Linux</td>
</tr>
<tr>
<td>Laptop (CernVM)</td>
<td>Lots of disk space; no limitations on execution time; runs on any platform; software managed centrally</td>
<td>Have to get virtual machine software first; quite complicated to set up; can’t run on nightlies</td>
</tr>
<tr>
<td>AFS-mounted desktop/laptop</td>
<td>Lots of disk space; no limitations on execution time; get the advantage of using LXPLUS set-up</td>
<td>AFS connection can be quite slow</td>
</tr>
</tbody>
</table>
Personal view - how do I work?

- Developing code in latest builds
  - LXPLUS or my AFS-mounted desktop
- Running over a few thousand events
  - AFS mounted local machine
- Running over huge samples
  - Grid, submitting my jobs from local institute, AFS-mounted desktop or CernVM
- Committing code to ATLAS repository and running quick tests
  - LXPLUS or AFS-mounted desktop
- Making histograms, doing statistical analysis
  - On my laptop (ROOT runs on all platforms)
Documentation and help

- Main computing page (including set-up information for non-LXPLUS platforms):

- Code browsing (password needed):
  - https://svnweb.cern.ch/cern/wsvn/atlasoff/

- Documentation for beginners:
  - WorkBook: https://twiki.cern.ch/twiki/bin/view/Atlas/WorkBook
  - Physics analysis WorkBook: https://twiki.cern.ch/twiki/bin/view/Atlas/PhysicsAnalysisWorkBook

- Setting up a desktop at CERN:
  - http://linux.web.cern.ch/linux/install/

- Help forums:
  - This one in particular: https://groups.cern.ch/group/hn-atlas-offlineSWHelp/default.aspx
• The absolute basics
  ▸ Setting up your account
  ▸ Running Athena
  ▸ Checking out and modifying an Athena package
  ▸ Making a new algorithm/package
  ▸ Running event generation
  ▸ Getting run information (run query tool)
  ▸ Finding and obtaining official ATLAS data with AMI and DQ2
  ▸ What’s in the ATLAS data?
Tutorial plan - tomorrow and Thursday

• Physics analysis with ATLAS software
  ▶ Analysis model in the first year of data taking
  ▶ Simple analysis with Athena
  ▶ Using data quality information in your analysis
  ▶ Calculating luminosity in your analysis
  ▶ Trigger-aware analysis
  ▶ Making dAOD
  ▶ Using Athena-ROOT-Access
  ▶ The ATLANTIS event display

• Using the tag system to select events for analysis
Tutorial plan - Friday morning

• Using the Grid via Ganga and pAthena
• Bringing it all together - physics analysis on the Grid
Other points

• ATLAS software isn’t perfect - there may be problems along the way - so please be patient
  ‣ Parts of the tutorial will be on REAL DATA so there might be some rough edges

• There are often many ways of doing the same thing, so if you see a colleague doing something differently to how you’ve been shown here, don’t be surprised

• These tutorials are meant to be informal
  ‣ Ask questions at any time
  ‣ Work through the exercises at your own pace
  ‣ Go for a coffee when you need it!
• This afternoon
  ▶ 60-6-015

• Wednesday, Thursday
  ▶ 60-6-015

• Friday morning
  ▶ 160-1-009
Let’s get started!

- Log into LXPLUS
  - `ssh -l username lxplus.cern.ch`

- Do you have directories called scratch0 and cmthome in your home area?
  - If not, make them:
    - `mkdir $HOME/scratch0`
    - `mkdir $HOME/cmthome`

- Now go to
  - and start the exercises