

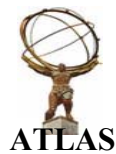
ATLAS-ARDA strategy and priorities

ARDA Workshop

David Adams

BNL

October 21, 2004



David Adams
BROOKHAVEN
NATIONAL LABORATORY



Contents

Introduction

- Key concepts
- Components
- ARDA prototype

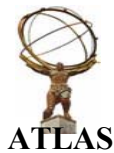
Integration

Strategy

- Workload management
- Data management
- Package management
- Catalog services
- Service discovery
- Service infrastructure

Priorities

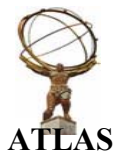
Final comments



Introduction

ATLAS Distributed Analysis (ADA)

- Distributed data, processing and users
- Supports all distributed activities
 - Classical analysis: event data → histograms
 - User-level production
 - Easy to add new activity (transformation)
- Strong emphasis on provenance tracking
 - Automatic, complete and accurate
- Easy to use
 - ROOT, python, GUI and command line submission and monitoring
- Analysis service insulates users from processing systems
 - Same look and feel for local or grid processing



Key concepts

Dataset

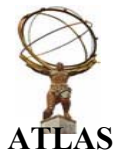
- Describes a collection of data
 - E.g. a collection of reconstructed events,
 - A collection of histograms, or ...

Transformation

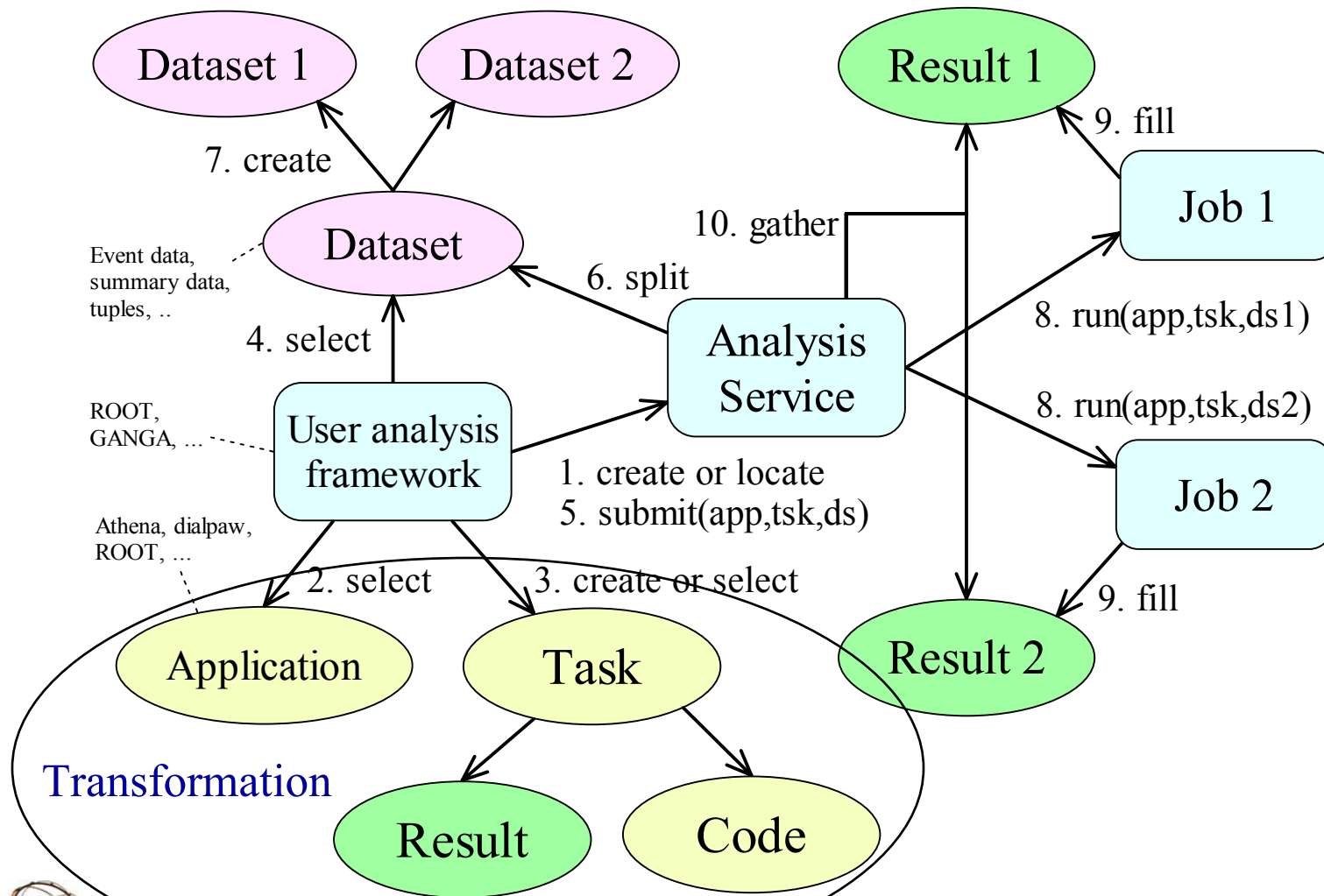
- Defines an operation to be performed on the data
 - Dataset \rightarrow Dataset
- Application + task (user configuration of application)

Job

- Instance of a transformation
- Typical user request processed as a collection of sub-jobs
 - Same transformation acting on sub-datasets
 - Master job includes splitting of input dataset and merging of output
- State and partial results available during processing



Key concepts (cont)



Components

Analysis service

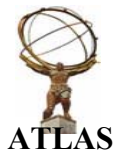
- Web service interface for submitting and monitoring jobs
- Multiple implementations to handle various processing systems
 - Including gLite
- Can be used hierarchically to scale to large number of jobs

Catalog services

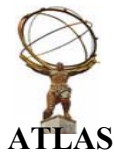
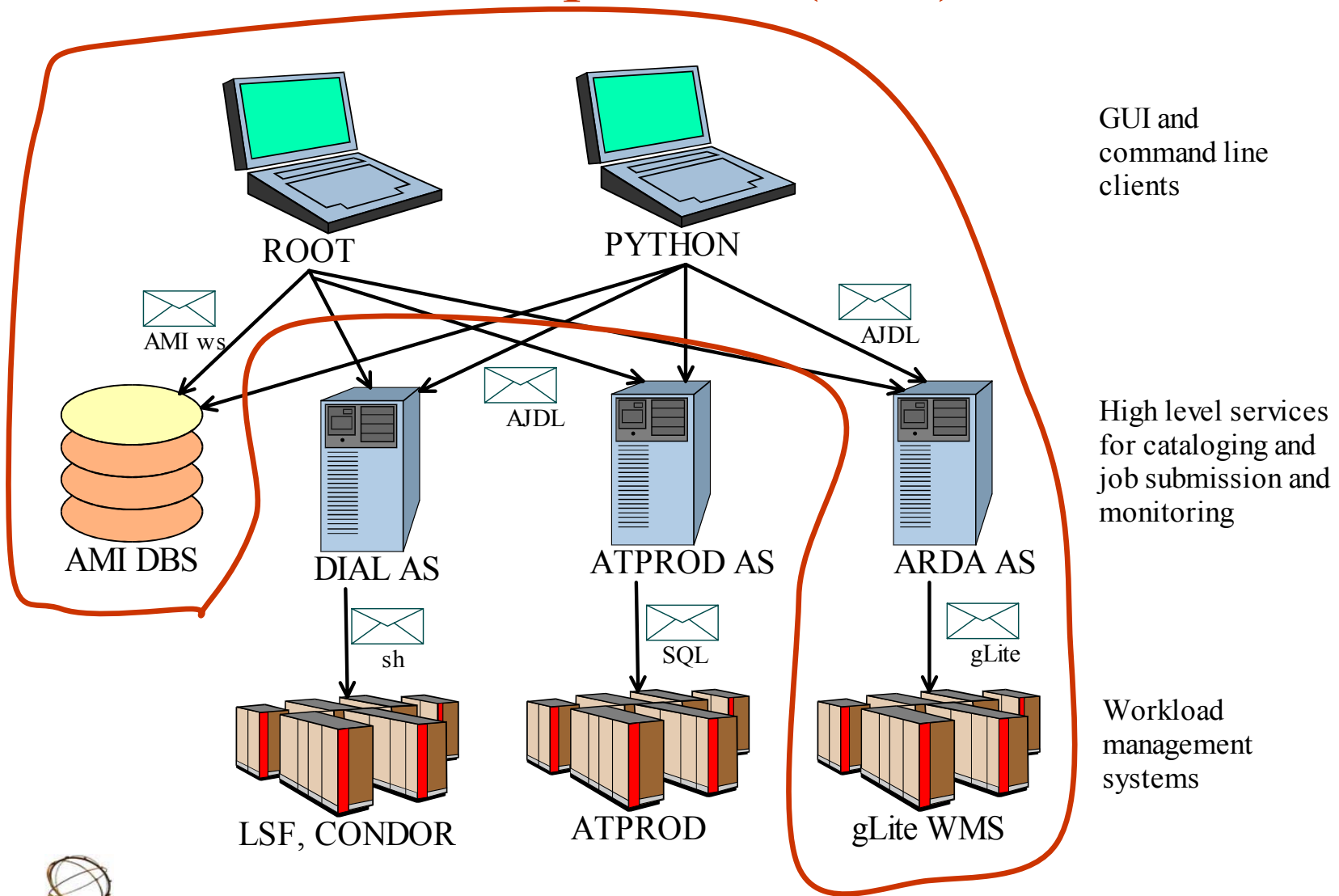
- Interfaces for recording datasets, transformations and jobs
- And metadata associated with these
- Provide provenance tracking

Clients

- Make use of analysis and catalog services
- ROOT and python with more possible
- Common interfaces enable mix and match of clients and services



Components (cont)



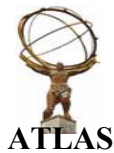
ARDA prototype

ARDA prototype

- End-to-end system
- Based on the gLite middleware

High-level strategy

- Keep the common ATLAS structure
 - Clients
 - Catalog services
- Add gLite analysis service based on the gLite WMS
 - Same user view for submission and monitoring
 - Easy to compare with other options (LCG, LSF, OSG, ...)
- Make use of the other GLite services or interfaces needed for effective use of the WMS
 - E.g. data management, package management, monitoring, ...
- Consider adoption of other useful gLite services and interfaces



Integration

Important to understand integration with other grids

- Nordugrid, OSG, vanilla globus, ...
- Including sites not associated with ATLAS or LHC

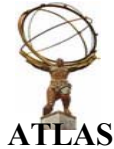
Options

1. Everyone deploys the gLite services
2. GLite services only deployed at EGEE sites
3. Some combination of 1 and 2

In any case, data must be shared

- Encourage adoption of common standards here
- SRM may become a common standard
- RLS appears to be failure
 - ATLAS manages this by adding Don Quijote

Maintaining a common security infrastructure is *very* helpful



Strategy

The following sections provide specifics on the following

- Workload management
- Data management
- Package management
- Catalog services
- Service discovery
- Service infrastructure



David Adams
BROOKHAVEN
NATIONAL LABORATORY



ARDA Workshop Atlas-ARDA strategy/priorities

Oct 21, 2004 10

Workload management

ATLAS jobs are first handled by an analysis service

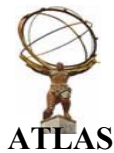
- Job specified by transformation and input dataset

Processing includes the following steps

- Split input dataset
- Define a job for each sub-dataset
- Submit jobs to WMS
- Monitor them
- As each finishes, append to the current result

Make use of the gLite WMS

- To process sub-jobs and possibly to carry out merging
- We must construct appropriate JDL to apply an ADA transformation
 - Where is JDL defined?



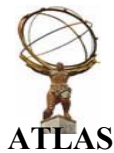
Workload management (cont)

Transformation does the following

- Locate and setup required software
 - Need package management
- Fetch required input data
 - Specified by dataset
 - LFN or GUID used to locate physical files
- Run software
- Save output data
 - Move to SE
 - Record in replica catalog
 - Construct output dataset

All done using generic interfaces

- Same transformation used for local processing and other grids
- User or AS may provide preferences, e.g. which SE or RC



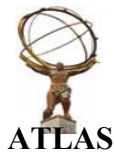
Data management

Local file management

- GLite has adopted SRM 1.1 for local file management (SE)
- ATLAS endorses this decision and is starting in the same way
- High priority (and possible ARDA activity) to integrate SRM into Don Quijote

Replica catalogs

- GLite interfaces for file and replica management are complex
- ATLAS will keep the Don Quijote as the interface over gLite and other useful catalogs
 - At present NG, LCG, Grid3 and gLite
- Need to understand what interface is be presented to gLite WMS
 - Avoid copying all data to gLite or
 - Implementing the full gLite DM interface
 - Unless we adopt Fireman as *the* ATLAS catalog



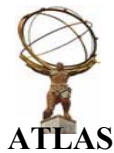
Data management (cont)

Bulk file transfers

- ATLAS is developing reliable transfer software for DC2
- Not yet clear how this connects to the gLite transfer service or the transfer capabilities in the present SRM implementations

Posix-like I/O for logical files

- Do not require this
 - To support as many sites as possible
- Might use it where available
 - Is there a performance gain over copy and run?



Package management

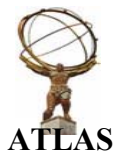
Transformations need to locate software

ATLAS is defining a simple command line interface
pkgmanager for accessing and installing software packages

- `ATLASDIR=`pkg_locate atlas 10.02.00``
- There are two implementations:
 - One based on pacman and
 - One enabling users to register pre-installed software
- Soon add one to combine any number of managers
- Need means to deploy at all sites
 - Then arbitrary transformation can run at any site willing to install all required software

GLite provide PackMan service interface

- Requires a job ID to find package location
- Might be able to provide a wrapper exposing the above interface



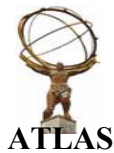
Catalog services

ADA has identified three catalog interfaces:

- XML Repository
 - Stores XML strings indexed by ID
 - ID is a string
- Selection catalog
 - Stores ID's indexed by key-value pairs
- Replica catalog
 - Stores replica ID list indexed by logical ID's
 - Not restricted to files

Now these are C++ class interfaces

- Considering adding web service interface
 - Support clients in other languages
 - Add server-side functionality



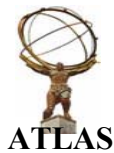
Catalog services (cont)

GLite Fireman is both too complex and too restrictive

- Includes metadata, ACL's, directories,...
- Requires ID be a GUID, replica is a SURL

GLite metadata catalog is close to ADA selection catalog

- Like to add query for schema,
- Support for basic data types (string, int, float), and
- Extensible schema (depending on entry)



Service discovery

We need a mechanism for service discovery

UDDI looks pretty complex

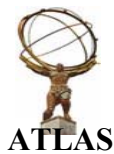
Clarens has put up a discovery service

- ATLAS is evaluating it

This is missing entirely in gLite

We would like to see support for site-based services

- Perhaps a discovery service running at each site
 - Still have to discover the site discovery service
- Convention to specify the site with which a machine is associated
 - Machines away from sites might choose an association
 - E.g. my laptop prefers CERN for storage and job submission



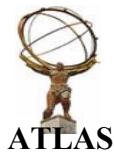
Service infrastructure

There are many service infrastructure issues

- Starting and stopping services
- Monitoring
- Discovery
- Security
 - Gridmap or authorization service
 - Delegation
 - Transport vs. message level security

ATLAS would like to understand how gLite is addressing these issues

- To understand how to deploy our services
- And integrate them with the gLite services



Priorities

1. Workload management service

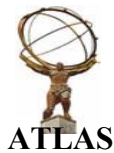
- Web service interface released and deployed (gLite)
 - And all supporting services
- With significant compute resources
- Corresponding analysis service (ARDA)

2. SRM

- Integrated into Don Quijote (ARDA)
- DQ integration into DIAL (DIAL)

3. Fireman

- Web service released and deployed (gLite)
 - Catalog gLite data
 - Candidate for single ATLAS catalog
- Integrated into DQ (ARDA)
 - So data is available to clients and other grids



Priorities (cont)

4. DM interface to gLite WMS for data on other grids

5. Package management

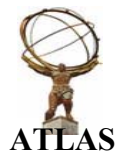
- Command line interface (gLite-NG-OSG or ATLAS)
 - For jobs on worker nodes
- Common WSDL for ADA and gLite
 - To perform or check installations before job submission

6. Service Infrastructure

- Security
- Discovery
- Monitoring

7. Catalog services

- Should be able to agree on metadata catalog



Final comments

Optimistic that ATLAS-ARDA will build a useful analysis system on gLite

- Need release and deployment of high priority gLite services
 - Couple months in advance of delivering end-to-end system
- Focus on delivering baseline interface and implementation
 - Extras and performance enhancements come later
 - GLite, ARDA and ATLAS
- Still much needed
 - From gLite, ARDA and ATLAS

GLite design has improved

- Little direct feedback to comments but
- Documents reflect our comments

Need to coordinate plans for non-EGEE sites

