



Enabling Grids for E-scienceE

Experiences and plans using gLite in HEP

D. Liko / CERN

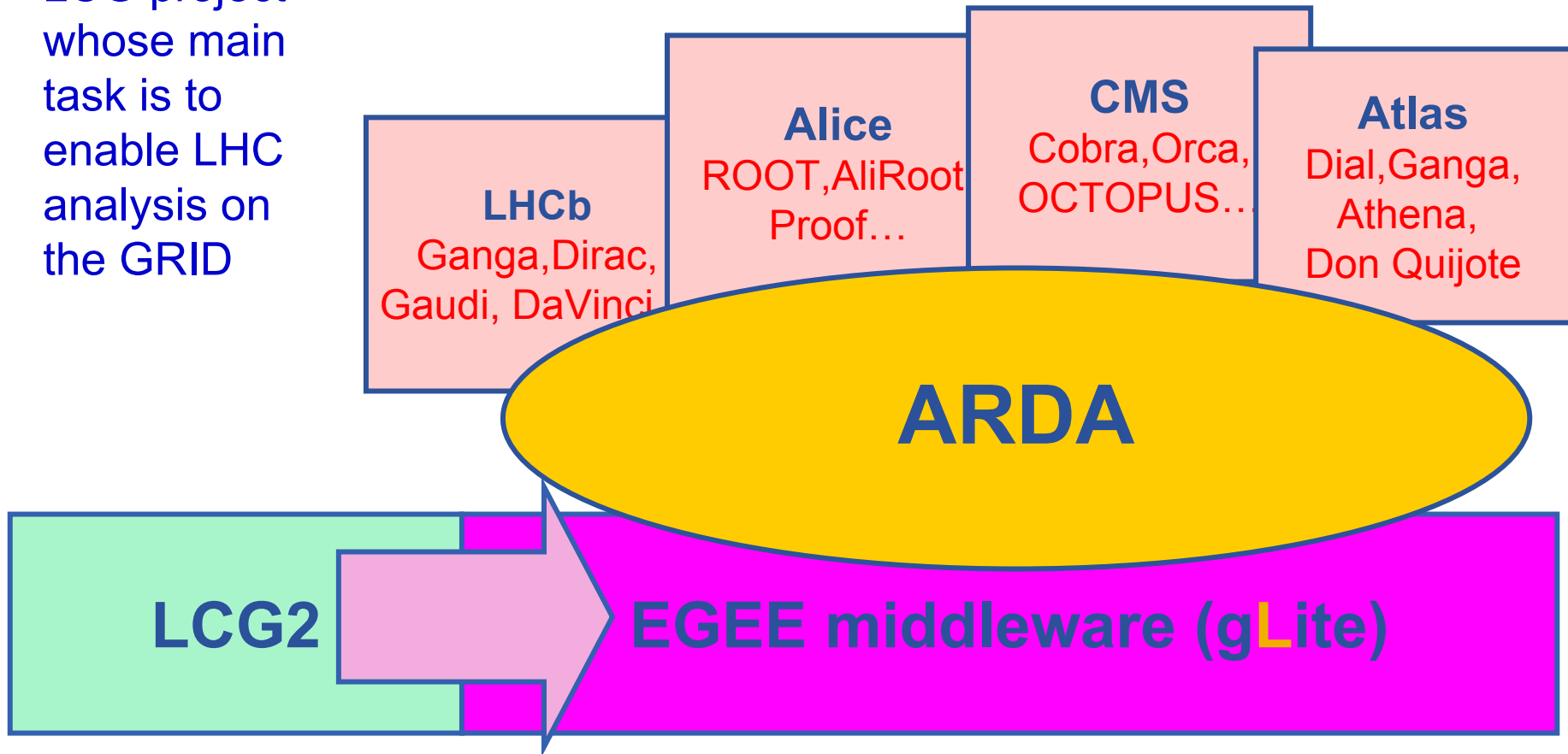
www.eu-egee.org



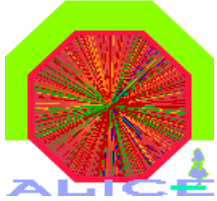
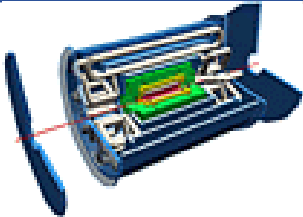



- ARDA in a nutshell
- Experiment prototypes
 - **CMS, ATLAS, LHCb and ALICE**
- Experience with gLite prototype
 - **Since May 18th**
 - **Several new components deployed since october**
- Conclusions

- **ARDA is an LCG project**
 - main activity is to enable LHC analysis on the grid
- **ARDA is contributing to EGEE NA4**
 - uses the entire CERN NA4-HEP resource
- **Work is based on last years experience/components**
 - Grid projects (LCG, VDT, EDG ...)
 - Experiments tools (Alien, Dirac, GAE, Octopus, Ganga, Dial,...)
- **Interface with the new EGEE middleware (gLite)**
 - Use the grid software as it matures
 - Key player in the evolution from LCG2 to the EGEE infrastructure
 - Verify the components in an analysis environments
 - Provide **early and continuous** feedback

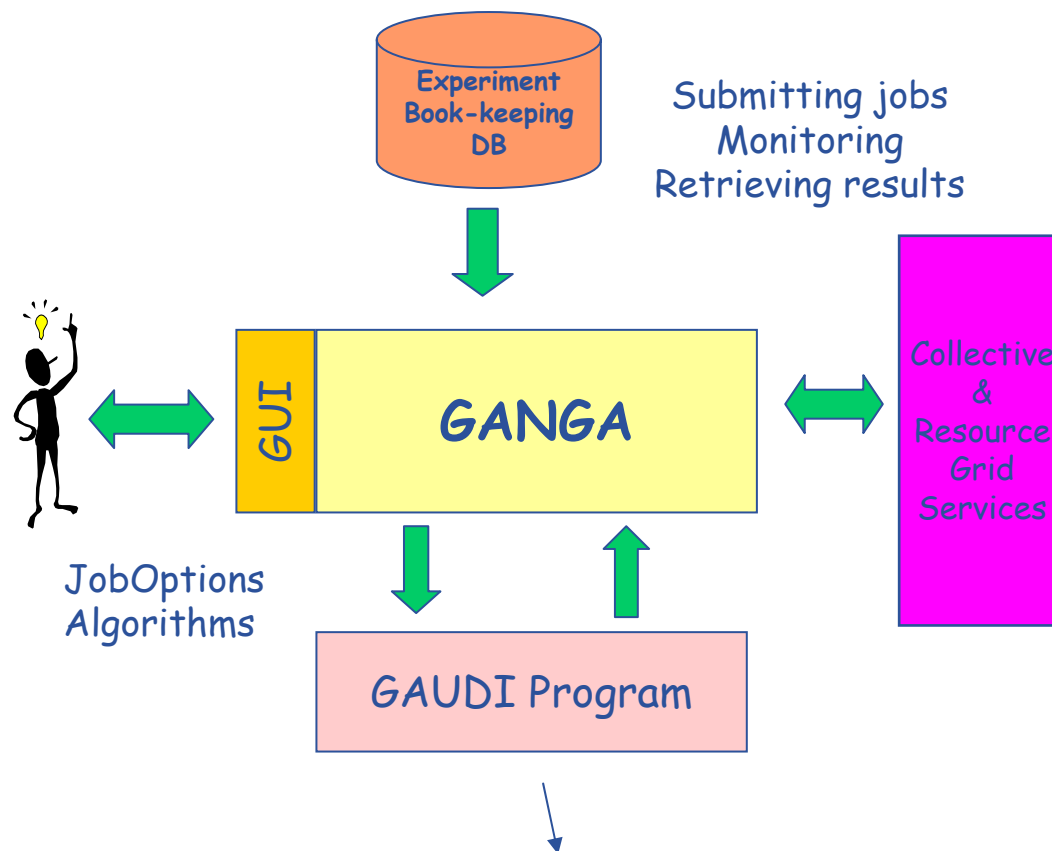
ARDA is an LCG project whose main task is to enable LHC analysis on the GRID



LHC Experiment	Main focus	Basic prototype component	Experiment analysis application framework	Middleware
	GUI to Grid	GANGA	DaVinci	
	Interactive analysis	PROOF ROOT	AliROOT	
	High level service	DIAL	Athena	
	Exploit native gLite functionality	Aligned with APROM	ORCA	

Gaudi/Athena aNd Grid Alliance

- Framework for job creating-submitting-monitoring
- Can be used with different physical applications and different submission back-ends
- Was also chosen as a prototype component by the Atlas experiment



GAUDI – LHCb analysis framework

- Integrating with gLite
 - Enabling job submission through GANGA to gLite
 - Job splitting and merging
 - Result retrieval
- Enabling real analysis jobs to run on gLite
 - Running DaVinci jobs on gLite
 - Installation of LHCb software using gLite package manager
- Release management and software process
 - CVS, Savannah,...
- Participating in the overall development
 - driven by GANGA team

- **GANGA-DIRAC (LHCb production system)**
 - Convergence with GANGA/components/experience
 - Submitting jobs to DIRAC using GANGA

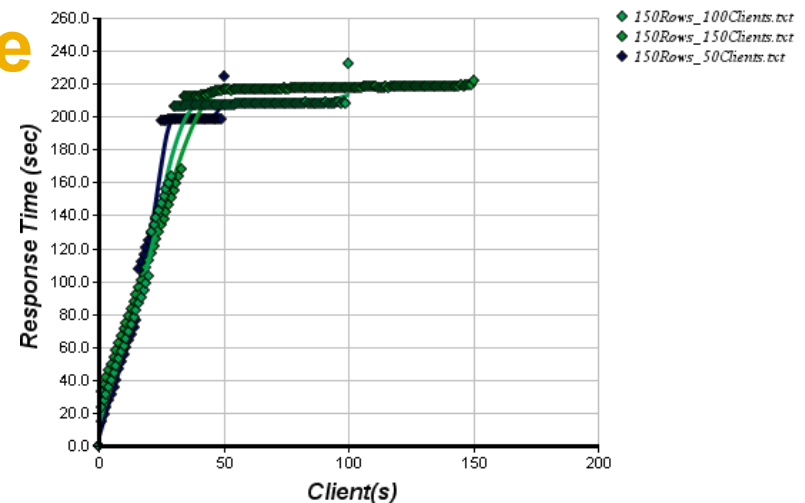
- **GANGA-Condor**

- Enabling submission of jobs through GANGA to Condor

- **LHCb Metadata catalogue performance tests**

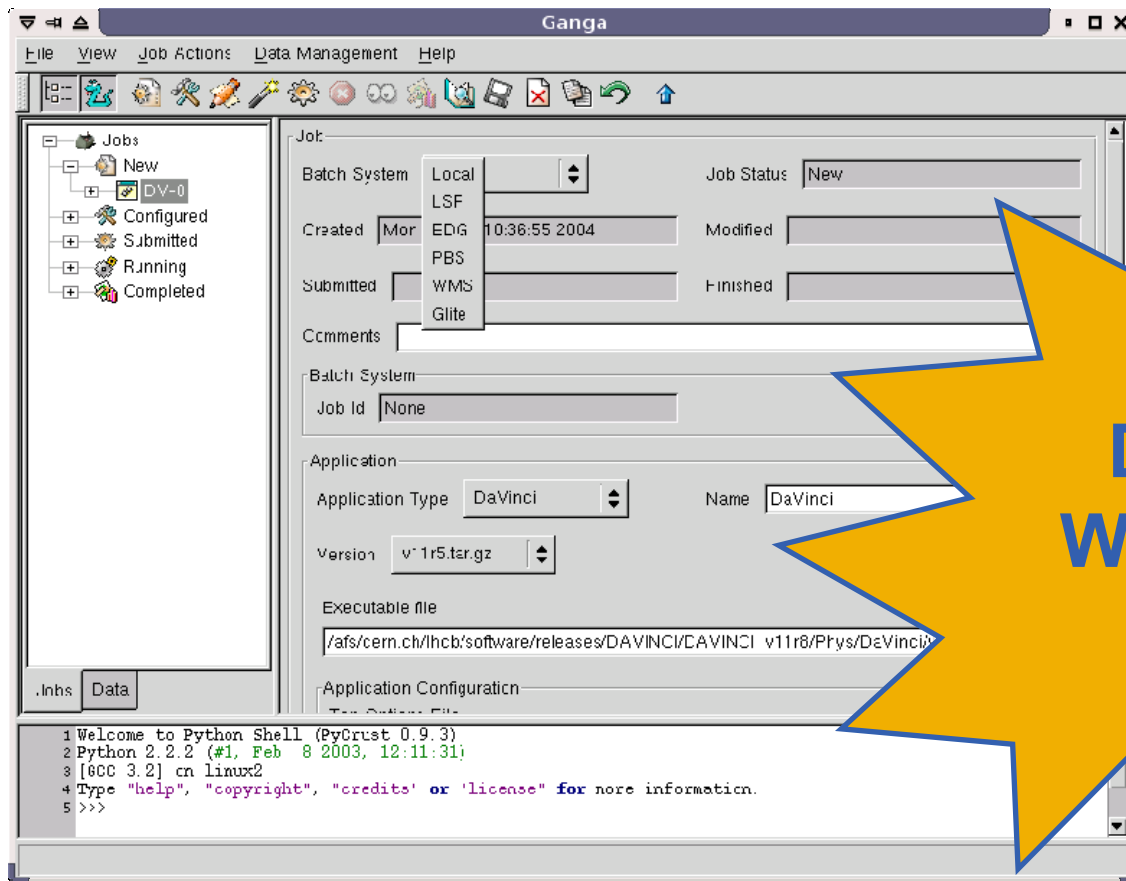
- In collaboration with colleagues from Taiwan

LHCb Bookkeeping Testing Result

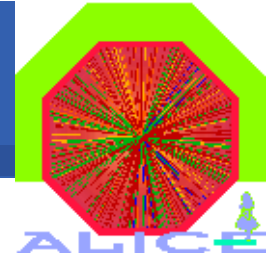


created with ChartDirector from www.advsofteng.com

- GANGA job submission handler for gLite is developed
- DaVinci job runs on gLite submitted through GANGA

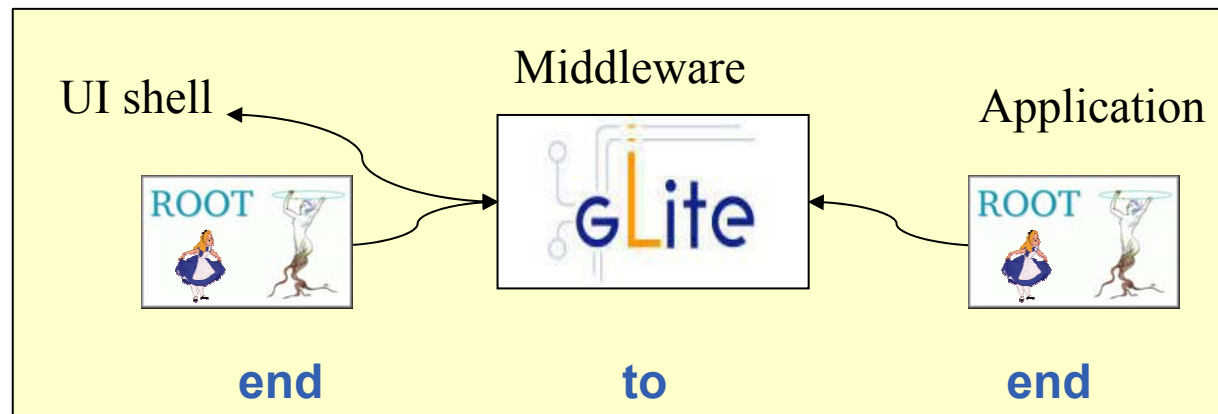


- Enabling of job splitting and output merging functionality
 - **Integrated to the GUI**
- Using of gLite package manager for LHCb software
- Involve members of the LHCb physics community
 - **more “naïve” user feedback**

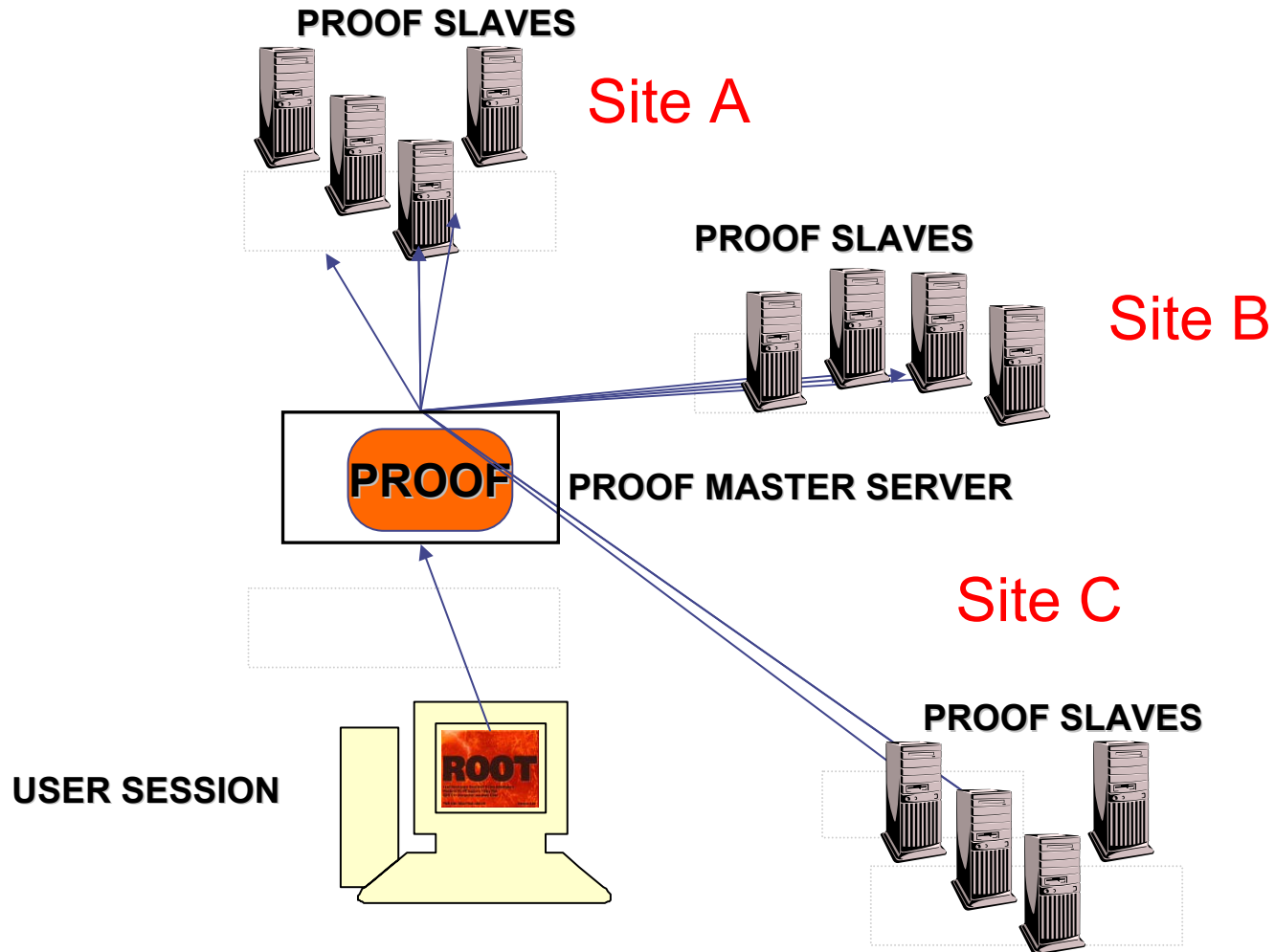
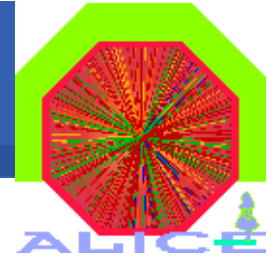


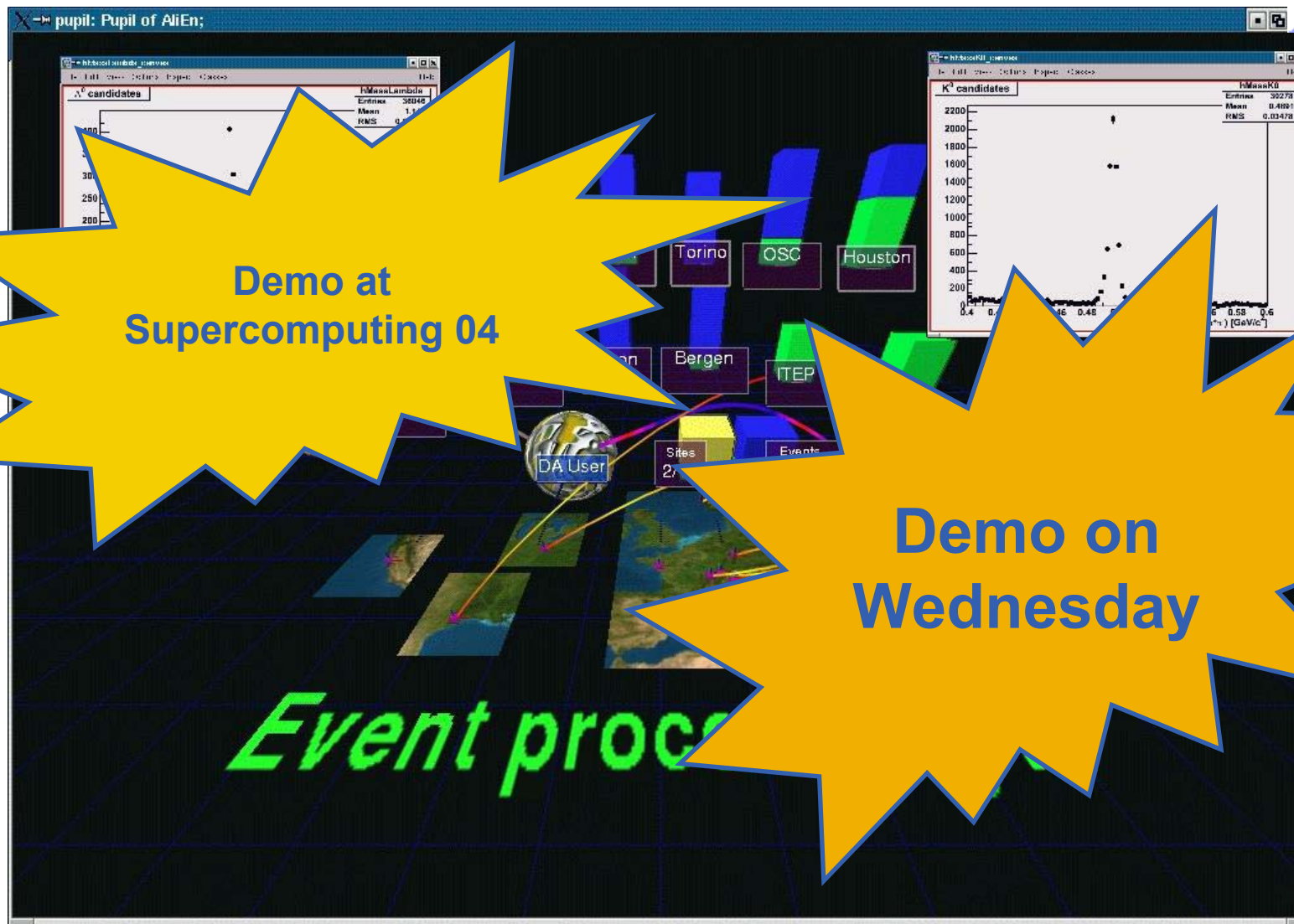
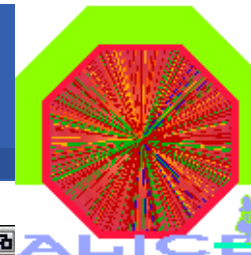
ROOT and PROOF

- ALICE provides
 - the UI
 - the analysis application (AliROOT)
- GRID middleware gLite provides all the rest



- ARDA/ALICE is evolving the ALICE analysis system

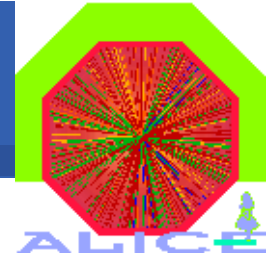


A screenshot of a software interface titled 'pupil: Pupil of AliEn;'. The interface displays two windows: 'Lambda candidates' on the left and 'K candidates' on the right. Both windows show scatter plots with data points and a fitted curve. The 'Lambda candidates' window includes a table with columns for 'Epsilon', 'Mean', and 'RMS'. The 'K candidates' window includes a table with columns for 'Epsilon', 'Mean', and 'RMS'. The background of the interface shows a 3D visualization of particle tracks and event processing components, including labels for 'Torino', 'OSC', 'Houston', 'Bergen', 'ITEP', 'Sites', and 'Events'. A globe labeled 'DA User' is also visible. The text 'Event processing' is written in green at the bottom of the interface.

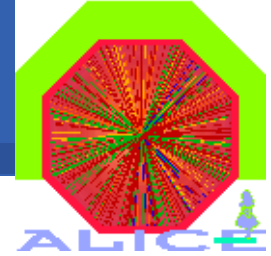
**Demo at
Supercomputing 04**

**Demo on
Wednesday**

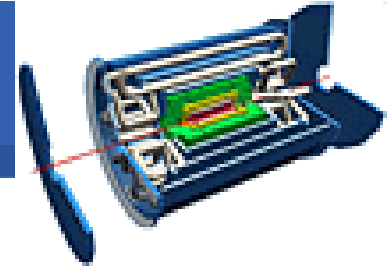
Event processing



- **Developed gLite C++ API and API Service**
 - providing generic interface to any GRID service
- **C++ API is integrated into ROOT**
 - will be added to the next ROOT release
 - job submission and job status query for batch analysis can be done from inside ROOT
- **Bash interface for gLite commands with catalogue expansion is developed**
- **First version of the interactive analysis prototype is ready**
- **Batch analysis model is improved**
 - submission and status query are integrated into ROOT
 - job splitting based on XML query files
 - application (Aliroot) reads file using xrootd without prestaging



- **Create generic API service accessible to all Alice users for batch analysis using bash CLI for the Alice data challenge phase III**
- **Make interactive prototype available to Alice users**
- **Create default XML datasets and default JDLs for analysis**

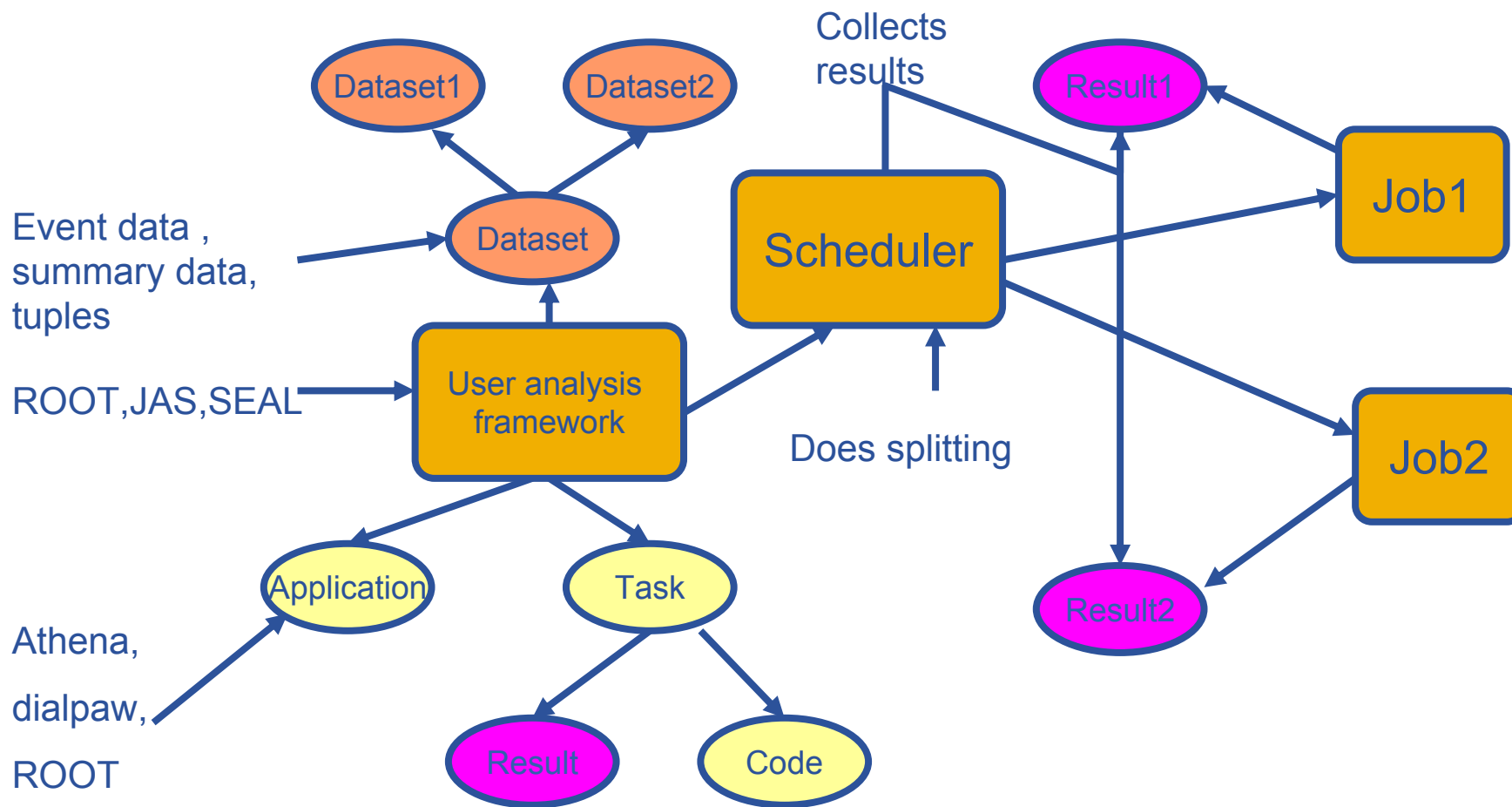
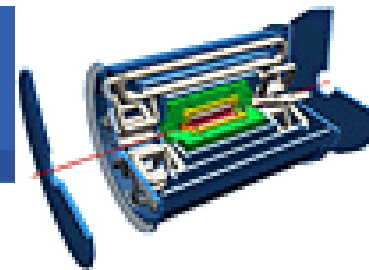


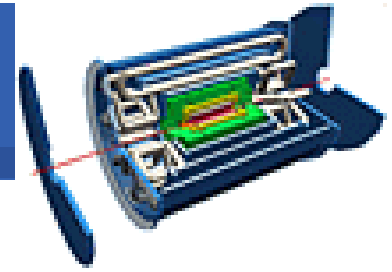
DIAL

Distributed Analysis of Large datasets

High Level Service

- Support the physicist to do their analysis on the GRID
- **Job description by AJDL**
 - General purpose high level job description
- **Functionality**
 - Job splitting
 - Supervision of job execution
 - Result merging
- **The service has been deployed**
 - Legacy batch systems
 - gLite

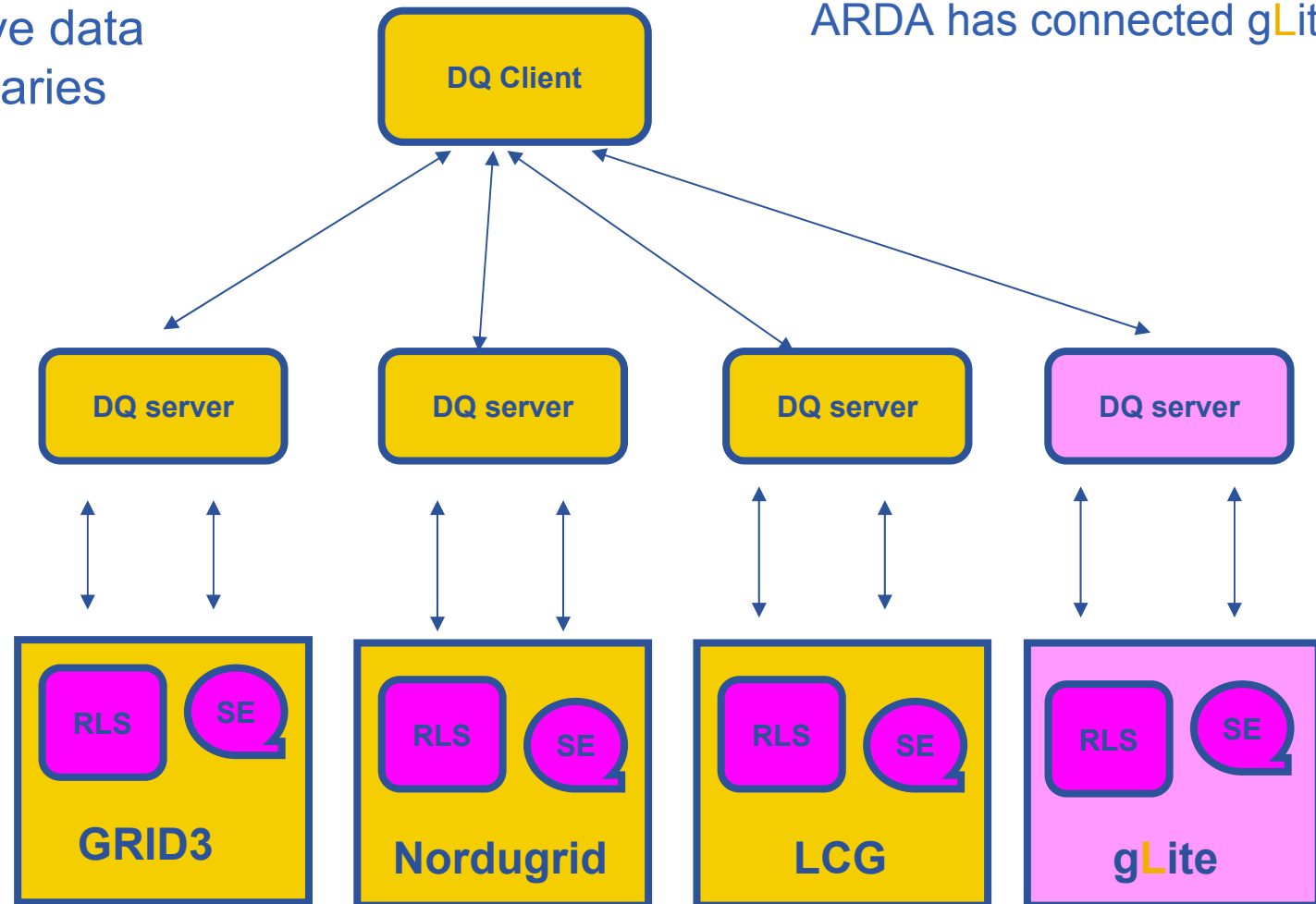


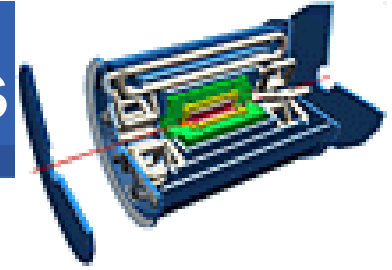


Don Quijote

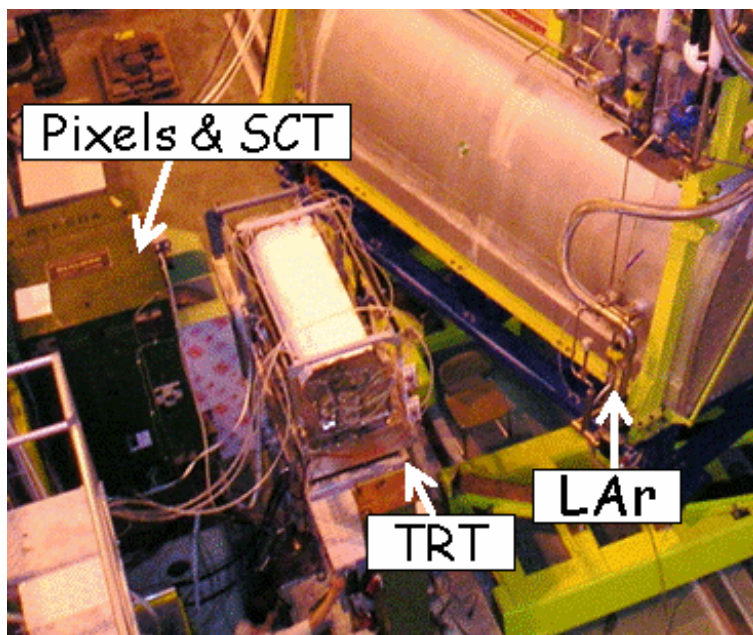
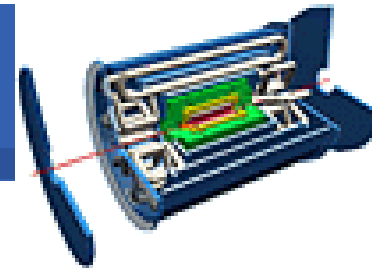
Locate and move data over grid boundaries

ARDA has connected gLite





- **Interface DIAL to gLite**
 - Evolves with test bed
- **Interface Don Quijote to gLite**
 - Evolves with test bed
- **Graphical User Interface ATCOM**
 - Work is ongoing
- **Combined test beam analysis application on gLite**
 - CTB is at the focus of the attention of ATLAS
 - Together with PNPI St. Petersburg



Real data processed at gLite

Standard Athena for testbeam

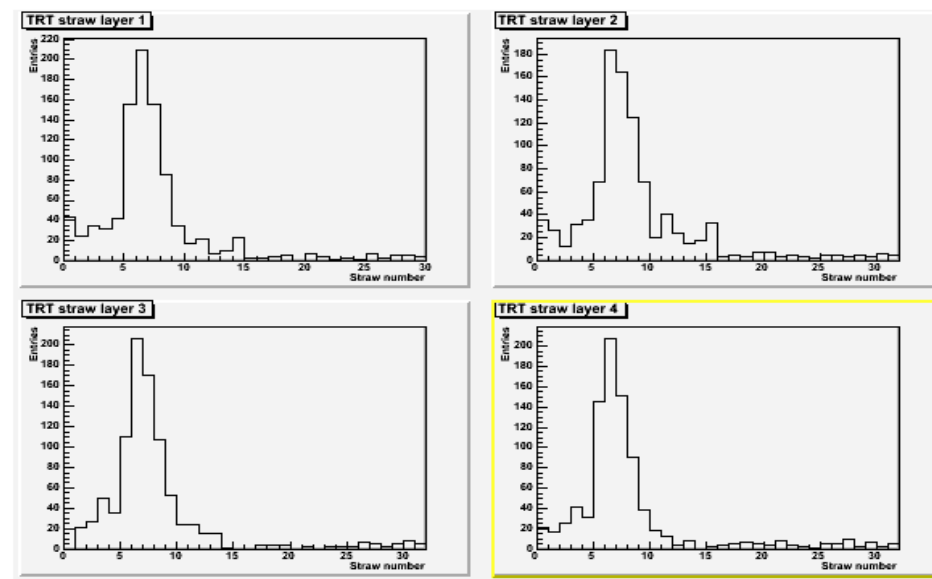
Data from CASTOR

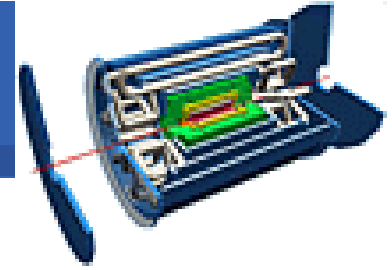
Processed on gLite worker node

Example:

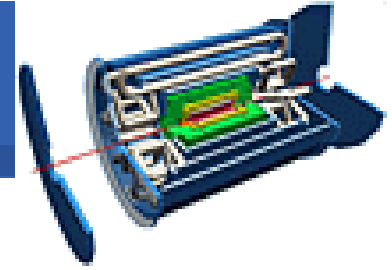
ATLAS TRT data analysis done by PNPI St Petersburg

Number of straw hits per layer

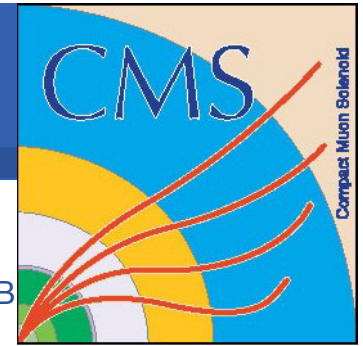




- DIAL server has been adapted to CERN environment and installed at CERN
 - First implementation of gLite scheduler for DIAL available
 - Still depending on a shared file system for inter-job communication
 - ATHENA jobs submitted through DIAL are run on gLite middleware
- Integration of gLite with Atlas file management based on Don Quijote is in progress, first prototype is ready
- Realistic ATHENA jobs executed on the gLite prototype by non-ARDA users (physicists).



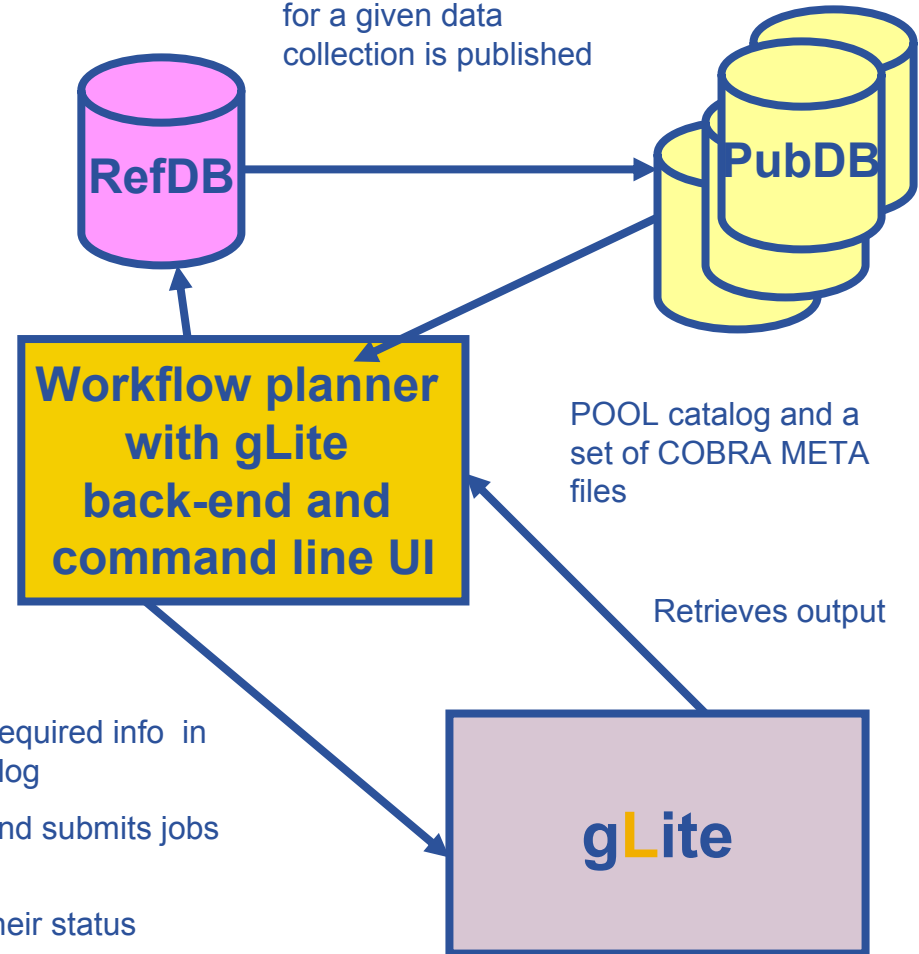
- Evolve DIAL gLite scheduler implementation
 - **directly interaction with the underlying middleware services (gLite WMS)**
 - **Integration of DQ**
- Evolve gLite DQ server
 - **SRM**
 - **Fireman**
- Increase the number of naïve users
 - **Who are not so naïve anyway ...**
 - **We have to provide working solutions that help the physicist to do their work!**



- Aims to end-to-end prototype for CMS analysis jobs on gLite
- **Native middleware functionality of gLite**
- Only for few CMS specific tasks on top of the middleware

Dataset and owner name
defining CMS data collection

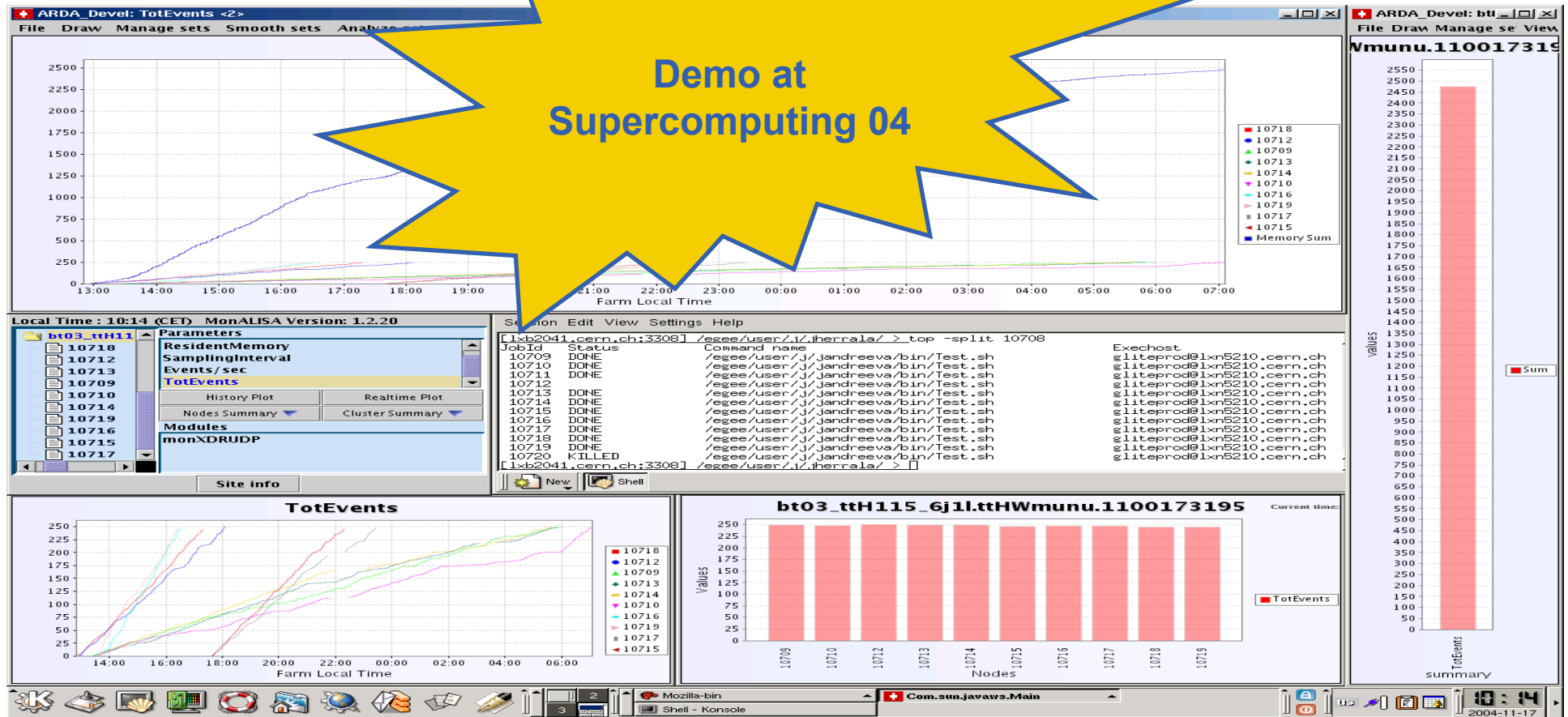
Points to the corresponding PubDB where POOL catalog for a given data collection is published

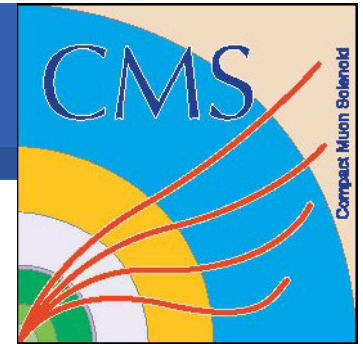


Register required info in gLite catalog
Creates and submits jobs to gLite,
Queries their status

User Job Monitoring

Demo at
Supercomputing 04



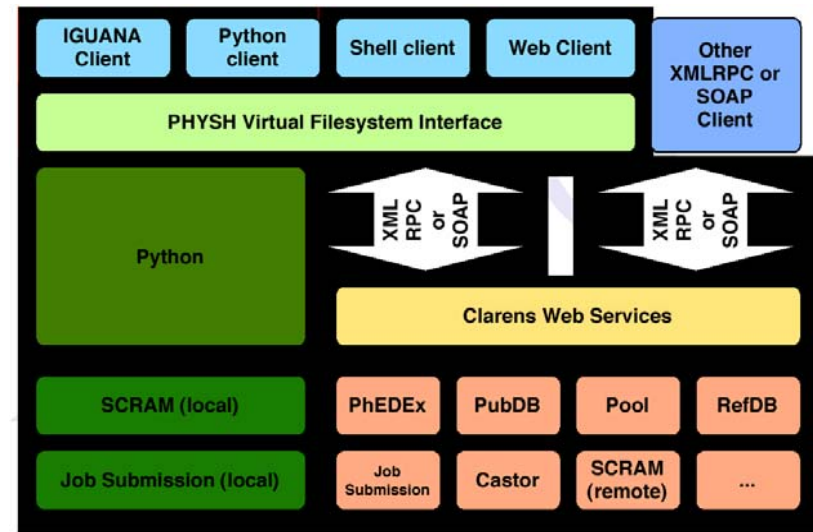


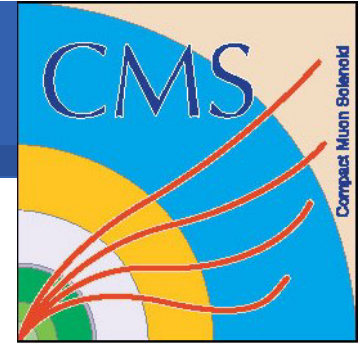
- **Job submission to gLite by PhySH**

- Physicist Shell
- Integrates Grid Tools
- Collaboration with CLARENS

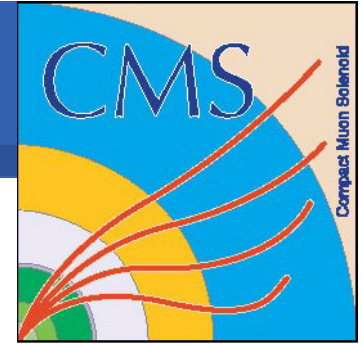
- **ARDA participates also in ...**

- Evolvement of PubDB
 - Effective access to data
- Redesign of RefDB
 - Metadata catalog





- ORCA analysis jobs (real user code) generated by CMS end-to-end prototype using gLite job-splitting functionality and instrumented for MonAlisa monitoring successfully ran on the gLite prototype testbed
- Work focused to enable merging of the output files produced by the child sub-jobs belonging to the same parent master job



- Demo of the prototype at the next CMS week (December)
- Involve naïve CMS users (limited number)
- gLite package manager for full CMS software distributions

- Available to us since May 18th
 - In the first month, many problems connected with the stability of the service and procedures
 - At that point just a few worker nodes available
 - Most important services are available:
file catalog, authentication module, job queue, meta-data catalog, package manager, Grid access service
 - A second site (Madison) available since end of June
 - CASTOR access to the actual data store

Currently 34 worker nodes are available at CERN

10 nodes (RH7.3, PBS)

20 nodes (low end, SLC, LSF)

4 nodes (high end, SLC, LSF)

1 node is available in Wisconsin

- Number of CPUs will increase
- Number of sites will increase
- FZK Karlsruhe is preparing to connect another site
 - **Basic middleware components already installed**
- Further extensions are under discussion right now

- **gLite uses Globus Grid-Certificates(X.509) to authenticate + authorize, session not encrypted**
- **VOMS is used for VO Management**
- **Getting access to gLite for a new user is often painful due to registration problems**
- **It takes minimum one day ... for some it can take up to two weeks!**

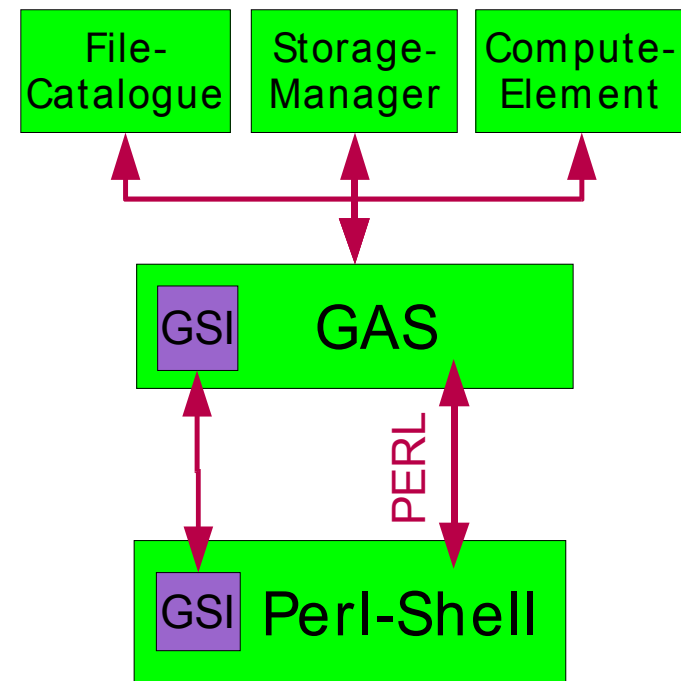
Easy access to gLite considered very important

Three shells available

- **Alien shell**
- **ARDA shell**
- **gLiteIO shell**

- Access through gLite-Alien shell
 - User-friendly Shell implemented in Perl
 - Shell provides a set of Unix-like commands and a set of gLite specific commands

- Perl API
 - no API to compile against, but Perl-API sufficient for tests, though it is poorly documented



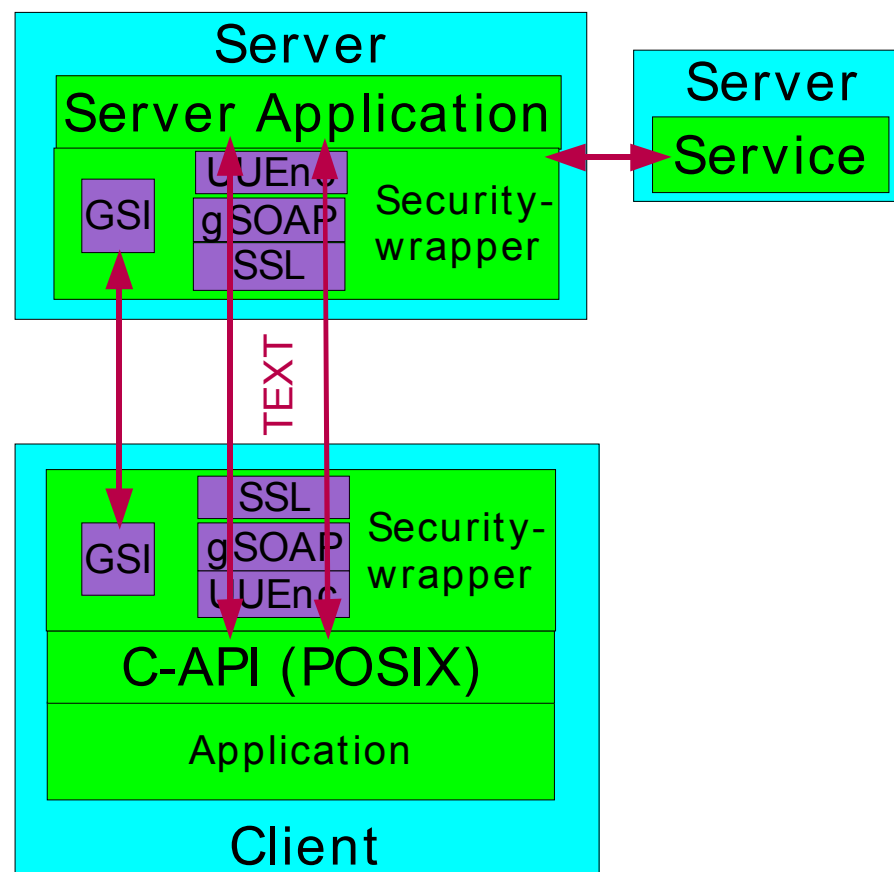
C++ access library for gLite has been developed by ARDA

- High performance
- Protocol quite proprietary...

Essential for the ALICE prototype

Generic enough for general use

Using this API grid commands have been added seamlessly to the standard shell

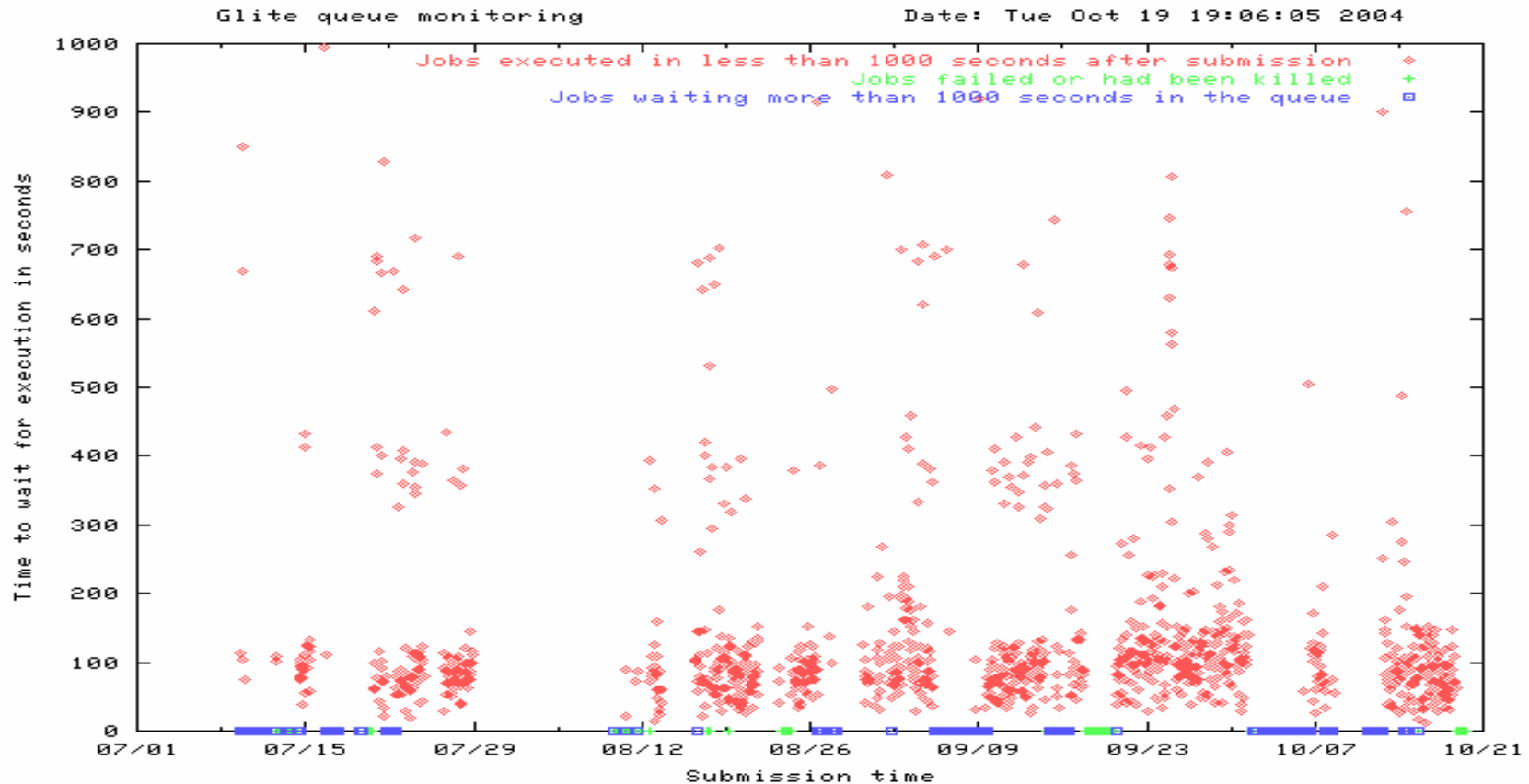


- **Integrate gLite IO as virtual file system**
 - Traps POSIX IO function calls and redirects them
 - No root access necessary
 - No recompilation of programs
- **Not obvious which programs will work**
 - Basic file IO works
 - Some standard program work
 - Editors don't work
 - Postscript viewers don't work
- **Only data access**
 - No job submission
 - No data management per se

- **Lightweight shell is important**
 - Ease of installation
 - No root access
 - Behind NAT routers
- **Shell goes together with the GAS**
 - Should presents the user a simplified picture of the grid
 - Strong aspect of the architecture
 - Not everybody liked it when it was presented
 - But “not everybody” implies that the rest liked the idea
 - Role of GAS should be clarified

ARDA has been evaluating two WMSs

- WMS derived from Alien – Task Queue
 - **available since April**
 - **pull model**
 - **integrated with gLite shell, file catalog and package manager**
- WMS derived from EDG
 - **available since middle of October**
 - **currently push model (pull model not yet possible but foreseen)**
 - **not yet integrated with other gLite components (file catalogue, package manager, gLite shell)**



- **Job queues monitored at CERN every hour by ARDA**
 - 80% Success rate (Jobs don't do anything real)
 - **component support should not depend on single key persons**

Submitting of a user job to gLite

- Register executable in the user bin directory
- Create JDL file with requirements
- Submit JDL

Straight forward, did not experience any problems
except system stability

Advanced features tested by ARDA

- Job splitting based on the gLite file catalogue LFN hierarchy
- Collection of outputs of split jobs in a master job directory

This functionality is widely used in the ARDA prototypes

- **Usage of WMS should to be transparent for the user**
 - same JDL syntax
 - worker nodes should be accessible through both systems
 - same functionality
 - Integration to other gLite services
- **JDL should be standardized on the design level**
 - An API with `submitJob(string)` leaves place for a lot of interpretation
 - There is clearly the place for obligatory and optional parameters
- **Debugging features are essential for the user**
 - Access to stdout/stderr for running jobs
 - Access to system logging information

ARDA has been evaluating two DMS

- **gLite File Catalog**

(deployed in April)

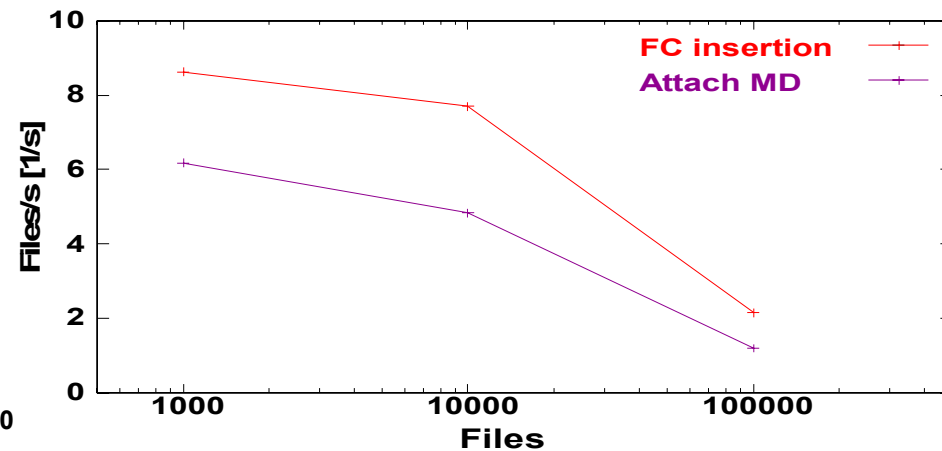
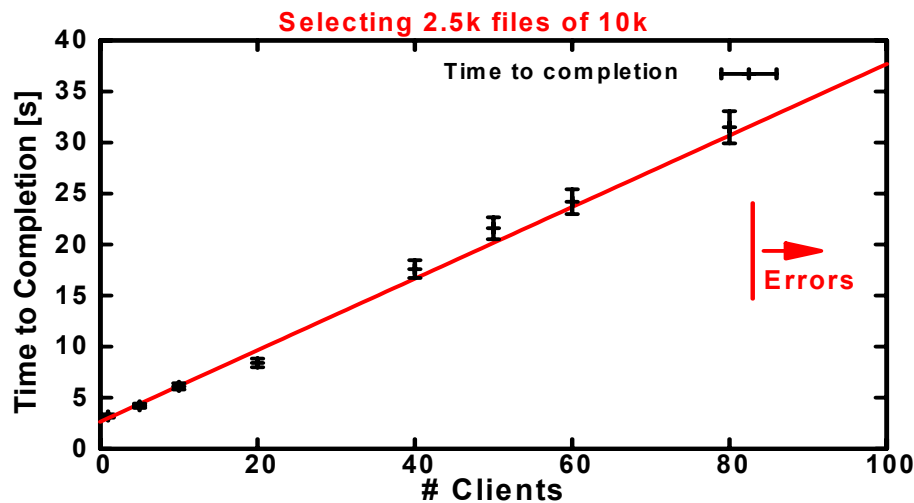
- Allowed to access experiments data from CERN CASTOR and – with low efficiency– from the Wisconsin installation
- LFN name space is organized as a very intuitive hierarchical structure
- MySQL backend

- **Local File Catalogue (Fireman)**

(deployed in November)

- Just delivered to us
- gliteO
- Oracle backend

- **gLite File catalog**
 - Good performance due to streaming
 - 80 concurrent queries, 0.35 s/query, 2.6s startup time

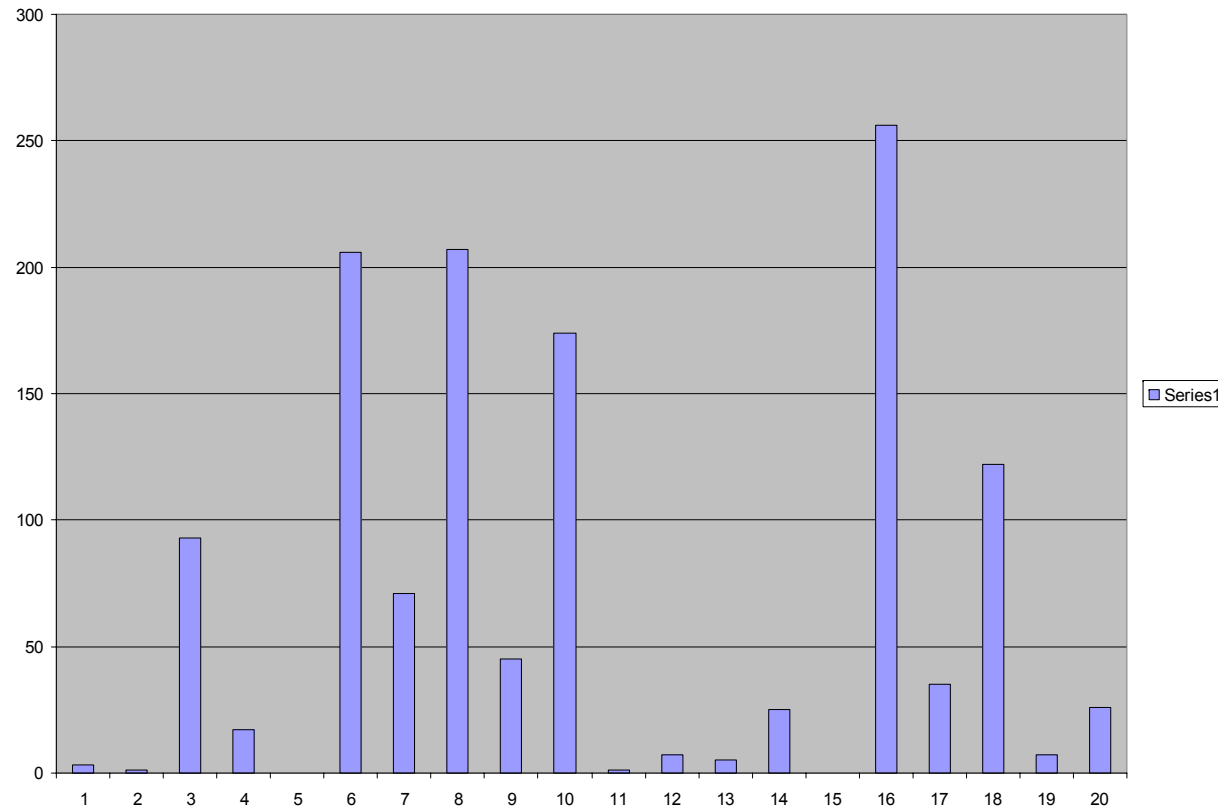


- **Fireman catalog**
 - First attempt to use the catalog: quite high entrance fee
 - Good performance
 - Not yet stable results due to unexpected crashes
 - We are interacting with the developers

- **Single entries up to 100000**
 - Successful, but no stable performance numbers yet
- **Bulk registration**
 - After some crashes, it seems to work more stable
 - No statistic yet
- **Bulk registration as a transaction**
 - In case of error, no file is registered

- **Simple test procedure**
 - Create small random file
 - copy to SE and read it back
 - Check if it still ok
 - Repeat that until one observes a problem
 -
- **A number of crashes observed**
 - From the client side the problem cannot be understood
 - We are interacting with the developers

- Try to copy 1000 files of 0 to 10KB



- On average an error occurred after 64 Files
- About 10 different error messages observed

- **We keep on testing the catalogs**
 - We are in contact with the developers
- **Consider a “clean” C++ API for the catalogs**
 - Hide the SOAP toolkit
 - Probably handcrafted
 - Or is there a better toolkit ????
- **gLiteO has to be rock stable**

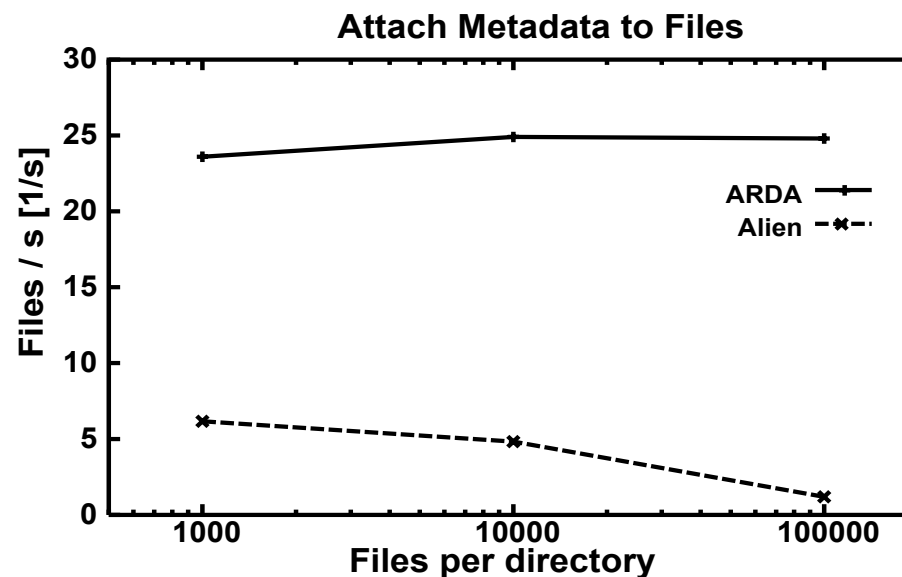
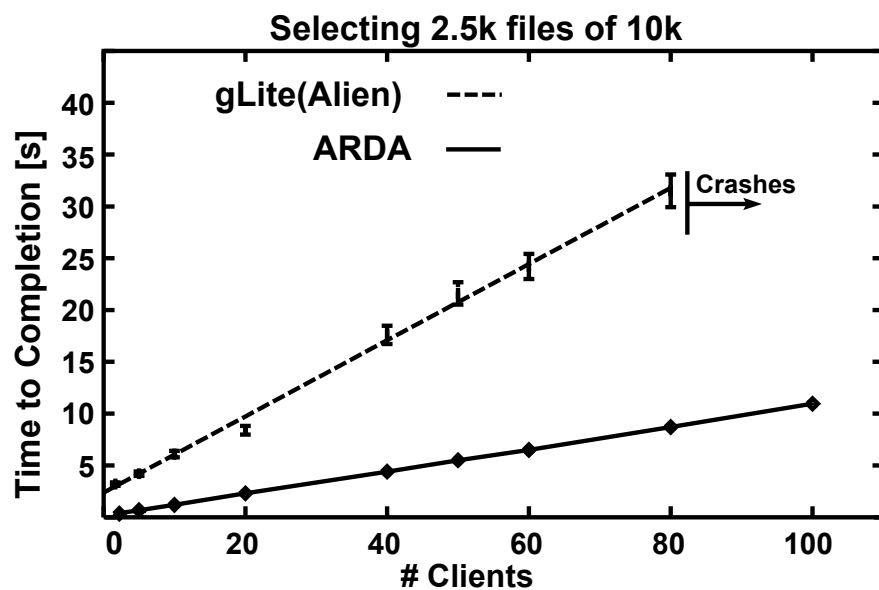
- **Multiple approaches exist for handling of the experiment software and user private packages on the Grid**
 - Pre-installation of the experiment software is implemented by a site manager with further publishing of the installed software. Job can run only on a site where required package is preinstalled.
 - Installation on demand at the worker node. Installation can be removed as soon as job execution is over.
- **Current gLite package management implementation can handle **light-weight** installations, close to the second approach**
- **Clearly more work has to be done to satisfy different use cases**



- gLite has provided a prototype interface and implementation mainly for the Biomed community
- The gLite file catalog has some metadata functionality and has been tested by ARDA
 - **Information containing file properties (file metadata attributes) can be defined in a tag attached to a directory in the file catalog.**
 - **Access to the metadata attributes is via gLite shell**
 - **Knowledge of schema is required**
 - **No schema evolution**
- Can these limitations be overcome?

- ARDA preparatory work
 - Stress testing of the existing experiment metadata catalogues was performed
 - Existing implementations showed to share similar problems
- ARDA technology investigation
 - On the other hand usage of extended file attributes in modern systems (NTFS, NFS, EXT2/3, SCL3, ReiserFS, JFS, XFS) was analyzed:
a sound POSIX standard exists!
 - Presentation in LCG-GAG and discussion with gLite
 - As a result of metadata studies a prototype for a metadata catalogue was developed

- **Tested operations:**
 - query catalogue by meta attributes
 - attaching meta attributes to the files



- **ARDA has been set up to**
 - enable distributed HEP analysis on gLite
 - Contact have been established
 - With the experiments
 - With the middleware
- **Experiment activities are progressing rapidly**
 - Prototypes for LHCb, ALICE, ATLAS & CMS are on the way
 - Complementary aspects are studied
- **ARDA is providing early feedback to the development team**
 - First use of components
 - Try to run real life HEP applications
 - Follow the development on the prototype
- **Some of the experiment related ARDA activities could be of general use**
 - Shell access
 - Metadata catalog

- **The LHC experiments for**
 - Help us understanding their requirements
 - Dedicate resources to us to prepare for the future
 - Try to understand how to use gLite when it will be released
- **The middleware developers**
 - Who guide us in the use of gLite
 - Who interact with us rapidly to solve the problems on the prototype test bed