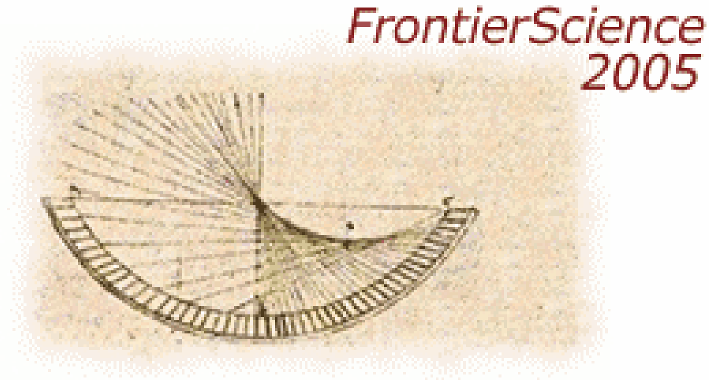
FrontierScience 2005



The ARDA Project
Prototypes for User Analysis on the GRID

Dietrich Liko/CERN IT

Enabling Grids for E-science in Europe

Overview

- ARDA in a nutshell
- ARDA prototypes
  - 4 experiments
- ARDA feedback on middleware
  - Middleware components on the development test bed
  - ARDA Metadata Catalog
- Outlook and conclusions

The ARDA project

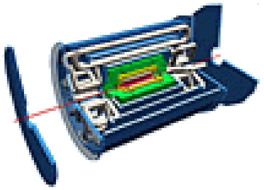
- ARDA is an LCG project
  - Main activity is to enable LHC analysis on the grid
  - ARDA is contributing to EGEE (NA4)

- Interface with the new EGEE middleware (gLite)
  - By construction, ARDA uses the new middleware
  - Verify the components in an analysis environments
    - Contribution in the experiments framework (discussion, direct contribution, benchmarking,…)
    - Users needed here. Namely physicists needing distributed computing to perform their analyses
  - Provide **early and continuous** feedback

- Activity extends naturally also to LCG
  - LCG is the production grid
  - Some gLite components are already part of LCG

See the presentation later

ARDA prototype overview

LHC Experiment	Main focus	Basic prototype component /framework	Middleware
LHCb	GUI to Grid	GANGA/DaVinci	gLite Lightweight Middleware for Grid Computing
ALICE	Interactive analysis	PROOF/AliROOT	
	High-level services	DIAL/Athena	
CMS Compact Muon Solenoid	Explore/exploit native gLite functionality	ORCA	

CMS

- ASAP = Arda Support for cms Analysis Processing
  - First version of the CMS analysis prototype capable of creating-submitting-monitoring of the CMS analysis jobs on the gLite middleware had been developed by the end of the year 2004
  - Prototype was evolved to support both RB versions deployed at the CERN testbed (prototype task queue and gLite 1.0 WMS)
  - Currently submission to both RBs is available and completely transparent for the users (same configuration file, same functionality)
  - Supports also current LCG

Starting point for users

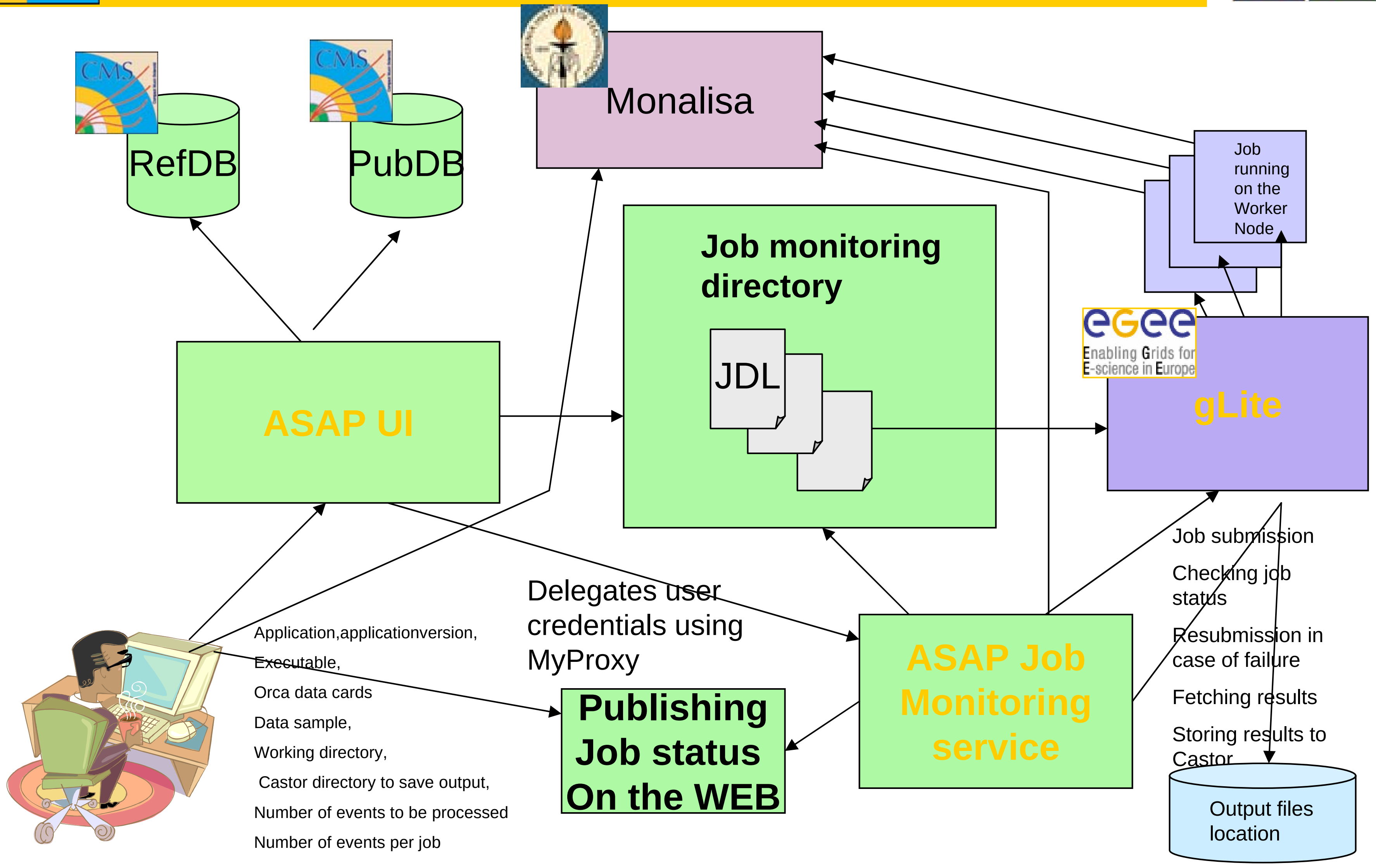
- The user is familiar with the experiment application needed to perform the analysis (ORCA application for CMS)

- The user debugged the executable on small data samples, on a local computer or computing services (e.g. lxplus at CERN)

- How to go for larger samples , which can be located at any regional center CMS-wide?

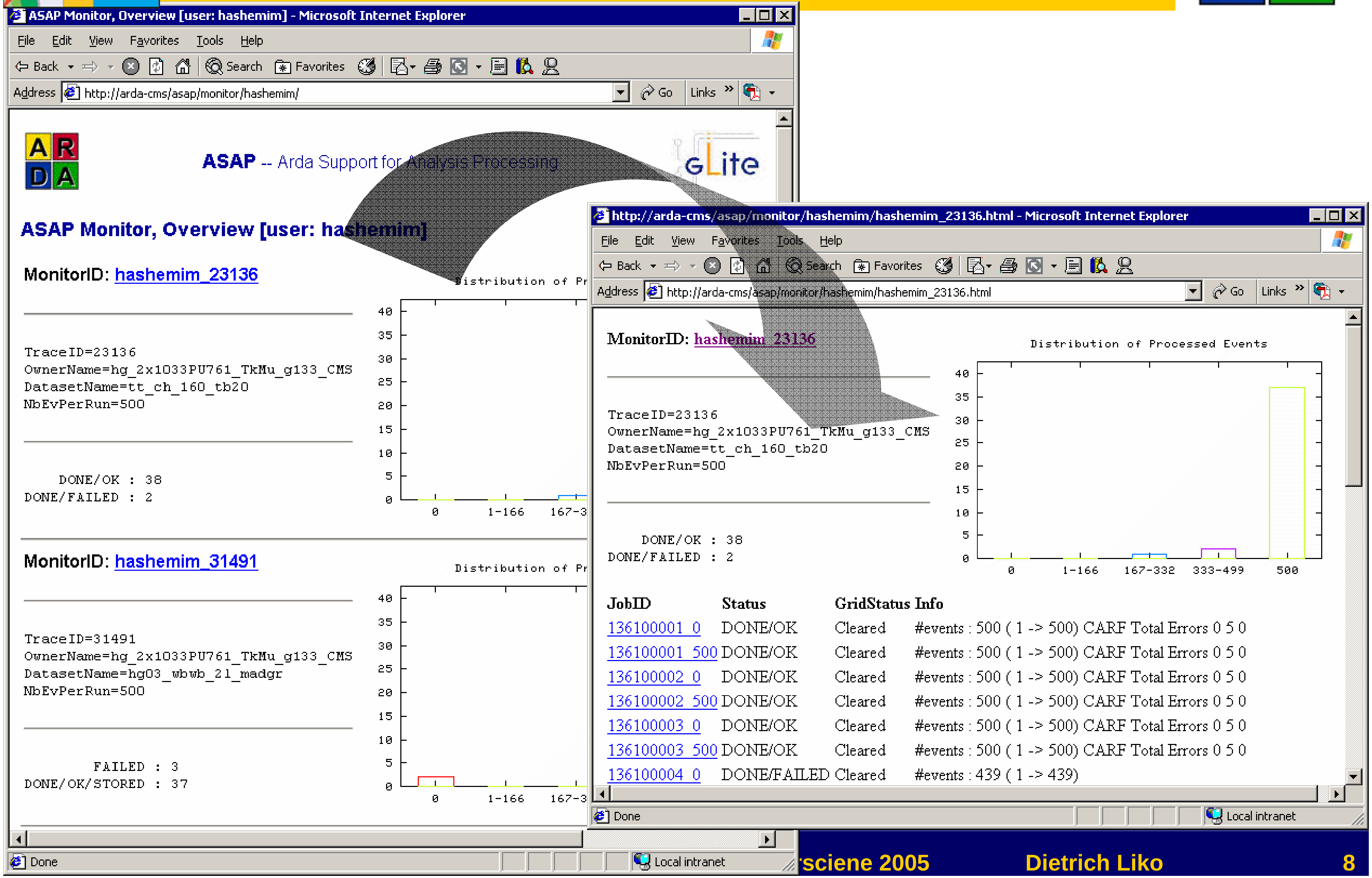
- The users should not be forced :
  - to change anything in the compiled code
  - to change anything in the configuration file for ORCA
  - to know where the data samples are located

ASAP work and information flow



RefDB
PubDB
Monalisa
Job running on the Worker Node
Job monitoring directory
JDL
ASAP UI
gLite
egee Enabling Grids for E-science in Europe
Delegates user credentials using MyProxy
Application,applicationversion,
Executable,
Orca data cards
Data sample,
Working directory,
Castor directory to save output,
Number of events to be processed
Number of events per job
Publishing Job status On the WEB
ASAP Job Monitoring service
Job submission
Checking job status
Resubmission in case of failure
Fetching results
Storing results to Castor
Output files location

Job Monitoring



ASAP Monitor, Overview [user: hashemim] - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://arda-cms/asap/monitor/hashemim/
Go Links
Done
Local intranet
http://arda-cms/asap/monitor/hashemim/hashemim_23136.html - Microsoft Internet Explorer
Address http://arda-cms/asap/monitor/hashemim/hashemim_23136.html
gLite
Distribution of Processed Events
40
35
30
25
20
15
10
5
0
0
1-166
167-332
333-499
500

ASAP -- Arda Support for Analysis Processing

ASAP Monitor, Overview [user: hashemim]

MonitorID: hashemim_23136

```
TraceID=23136
OwnerName=hg_2x1033PU761_TkMu_g133_CMS
DatasetName=tt_ch_160_tb20
NbEvPerRun=500
```

```
    DONE/OK : 38
DONE/FAILED : 2
```

MonitorID: hashemim_31491

```
TraceID=31491
OwnerName=hg_2x1033PU761_TkMu_g133_CMS
DatasetName=hg03_wbwb_21_madgr
NbEvPerRun=500
```

```
        FAILED : 3
DONE/OK/STORED : 37
```

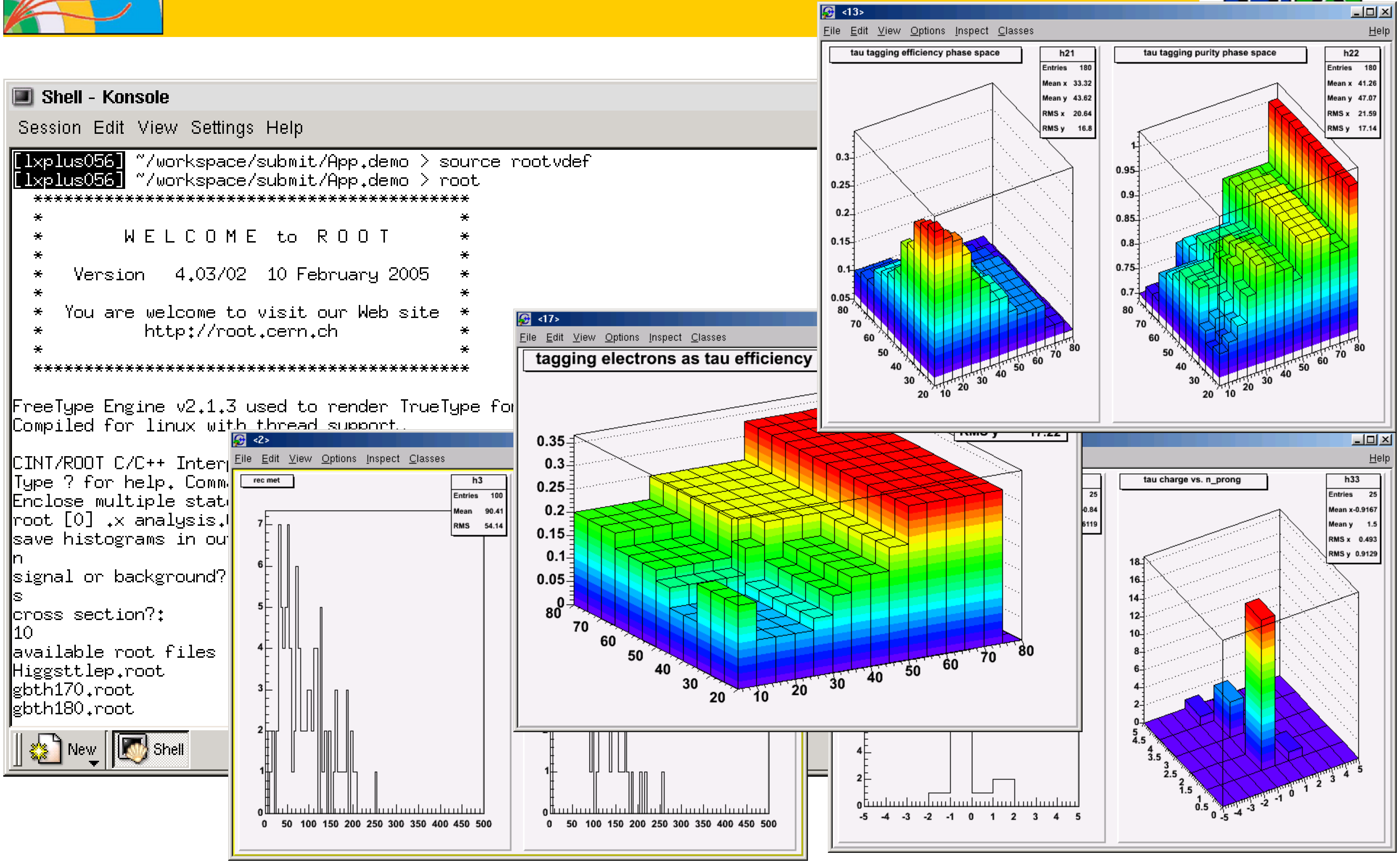
MonitorID: hashemim_23136

```
TraceID=23136
OwnerName=hg_2x1033PU761_TkMu_g133_CMS
DatasetName=tt_ch_160_tb20
NbEvPerRun=500
```

```
    DONE/OK : 38
DONE/FAILED : 2
```

JobID	Status	GridStatus	Info
136100001_0	DONE/OK	Cleared	#events : 500 (1 -> 500) CARF Total Errors 0 5 0
136100001_500	DONE/OK	Cleared	#events : 500 (1 -> 500) CARF Total Errors 0 5 0
136100002_0	DONE/OK	Cleared	#events : 500 (1 -> 500) CARF Total Errors 0 5 0
136100002_500	DONE/OK	Cleared	#events : 500 (1 -> 500) CARF Total Errors 0 5 0
136100003_0	DONE/OK	Cleared	#events : 500 (1 -> 500) CARF Total Errors 0 5 0
136100003_500	DONE/OK	Cleared	#events : 500 (1 -> 500) CARF Total Errors 0 5 0
136100004_0	DONE/FAILED	Cleared	#events : 439 (1 -> 439)

Merging the results



```
[lxplus056] ~/workspace/submit/App.demo > source rootvdef
[lxplus056] ~/workspace/submit/App.demo > root
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   4.03/02   10 February 2005  *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************

FreeType Engine v2.1.3 used to render TrueType fo
Compiled with thread support.

CINT/ROOT C/C++ Inter
Type ? for help. Comm
Enclose multiple stat
root [0] .x analysis.
save histograms in ou
n
signal or background?
s
cross section?:
10
available root files
Higgsttlep.root
gbth170.root
gbth180.root
```

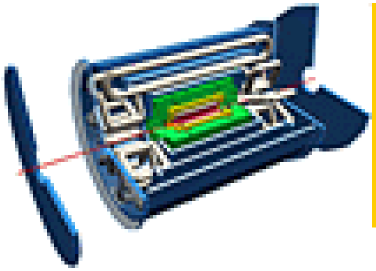
Integration

- Development is now coordinated with the EGEE/LCG Taskforce

- Key ASAP components will be merged and migrated with the CMS mainstream tools as BOSS and CRAB.

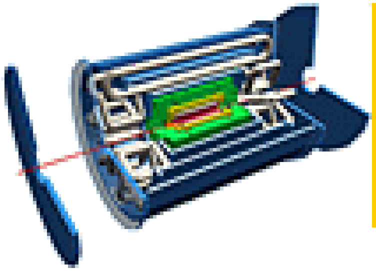
- Selected features of ASAP will be implemented separated
  - Task monitor: correlation/presentation of information from different sources
  - Task manager: control level to provide disconnected operation (submission, resubmission,…)

- Further contribtions
  - Dashboard
  - MonAlisa Monitoring

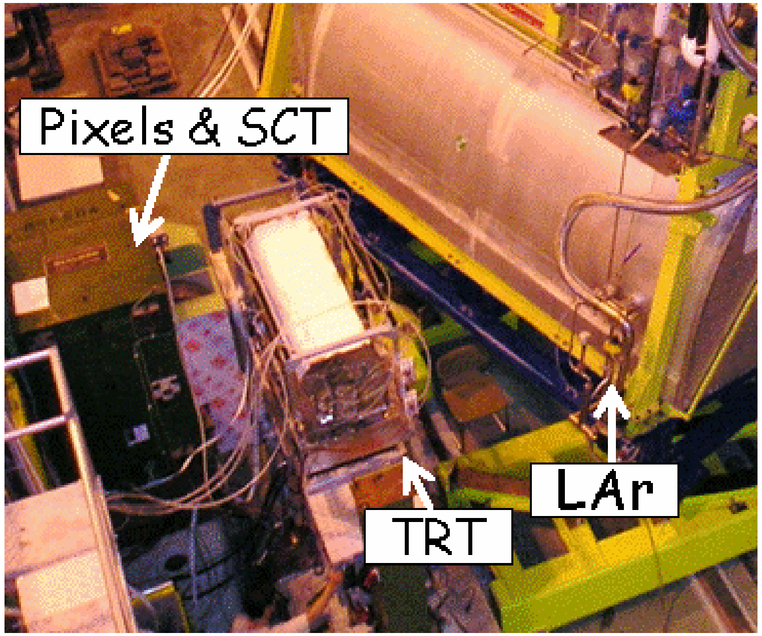


ATLAS

- Main Activities during last year
  - DIAL to gLite scheduler
  - Analysis Jobs with the ATLAS production system
  - GANGA (Principal component of the LHCb prototype, but also part of ATLAS DA)
- Other issues addressed
  - AMI tests and interaction
  - ATCom Production and CTB tools
  - Job submission (ATHENA jobs)
  - Integration of the gLite Data Management within Don Quijote
  - Active participation in several ATLAS reviews
  - First look on interactivity/resiliency issues (DIANE)
- Currently working on redefining the ATLAS Distributed Analysis strategy
  - On the basis of the ATLAS Production system



Combined Test Beam

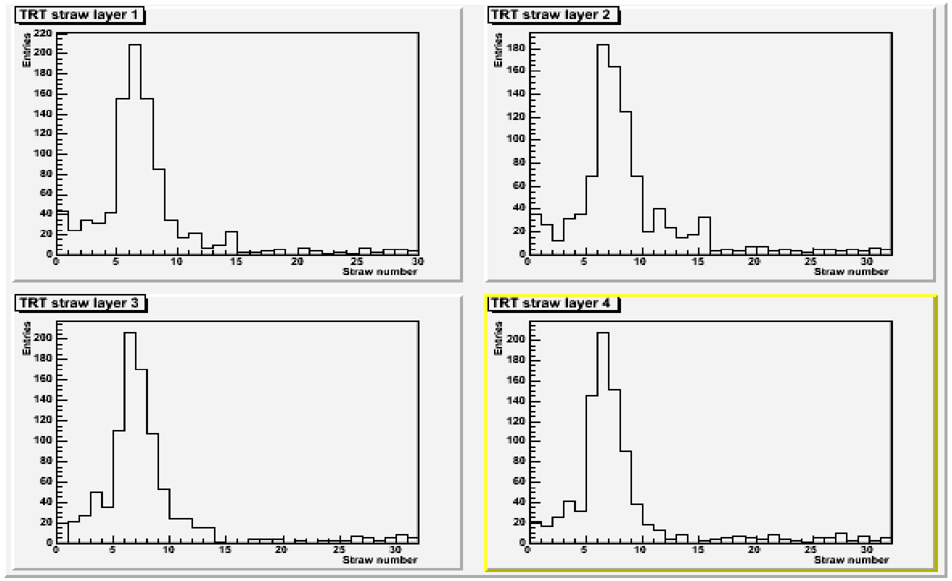


Real data processed at gLite

Standard Athena for testbeam

Data from CASTOR

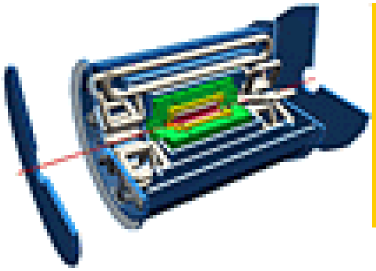
Processed on gLite worker node



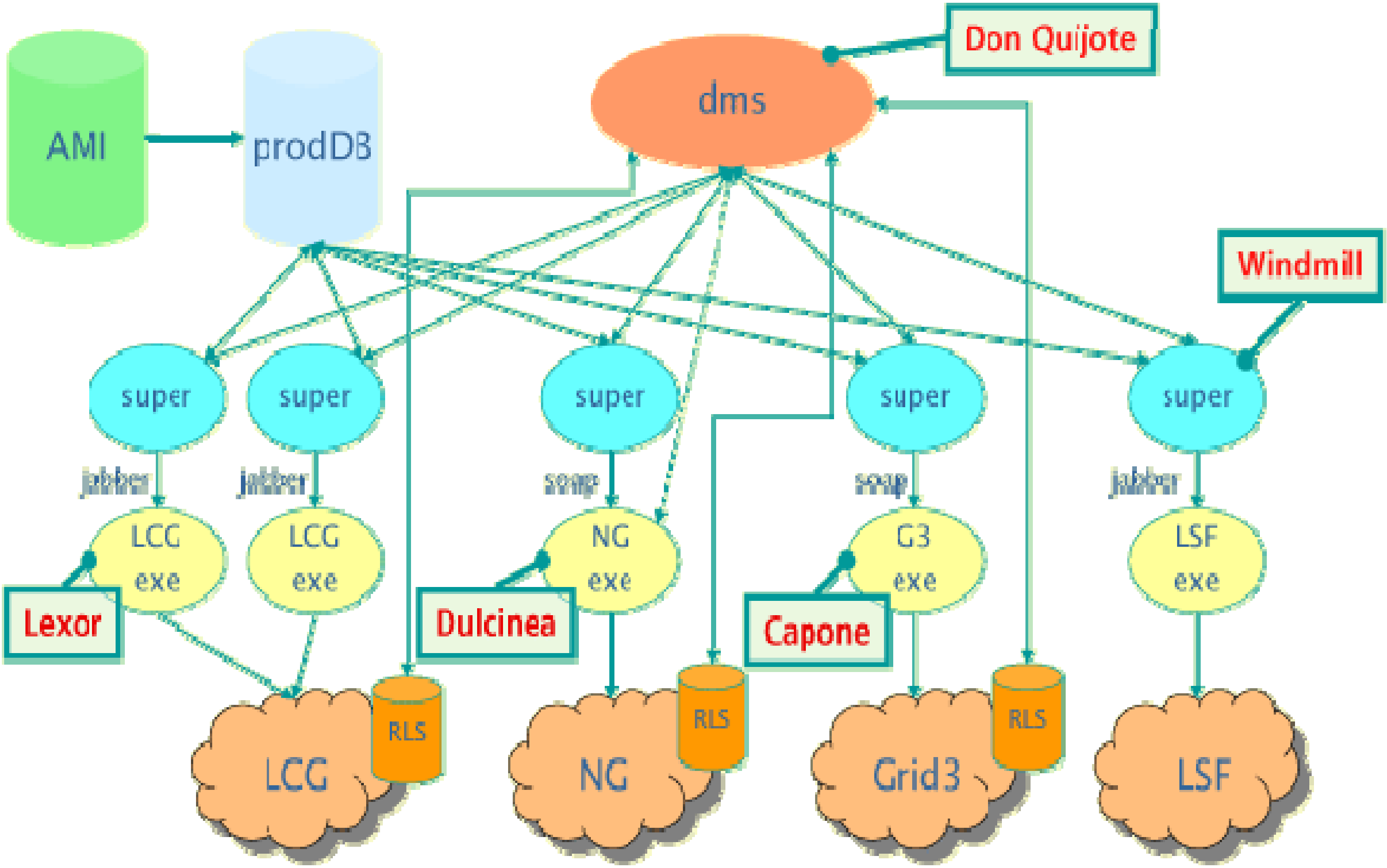
Example:

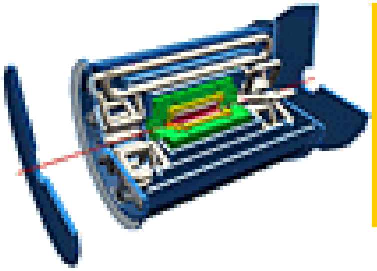
ATLAS TRT data analysis done by PNPI St Petersburg

Number of straw hits per layer



Production system



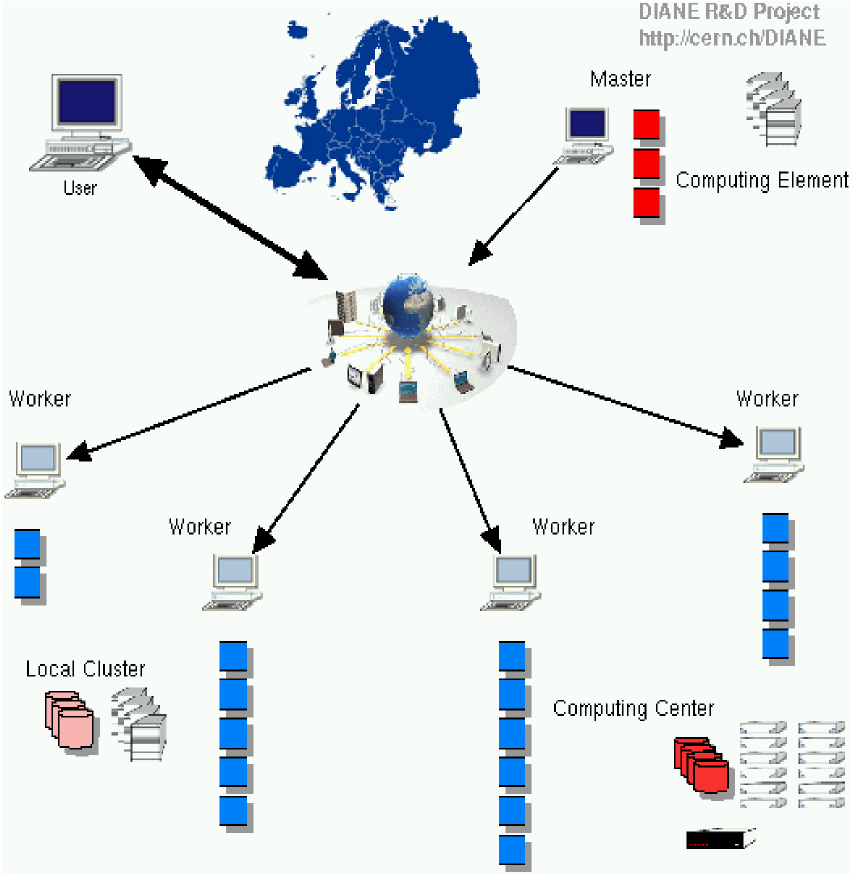


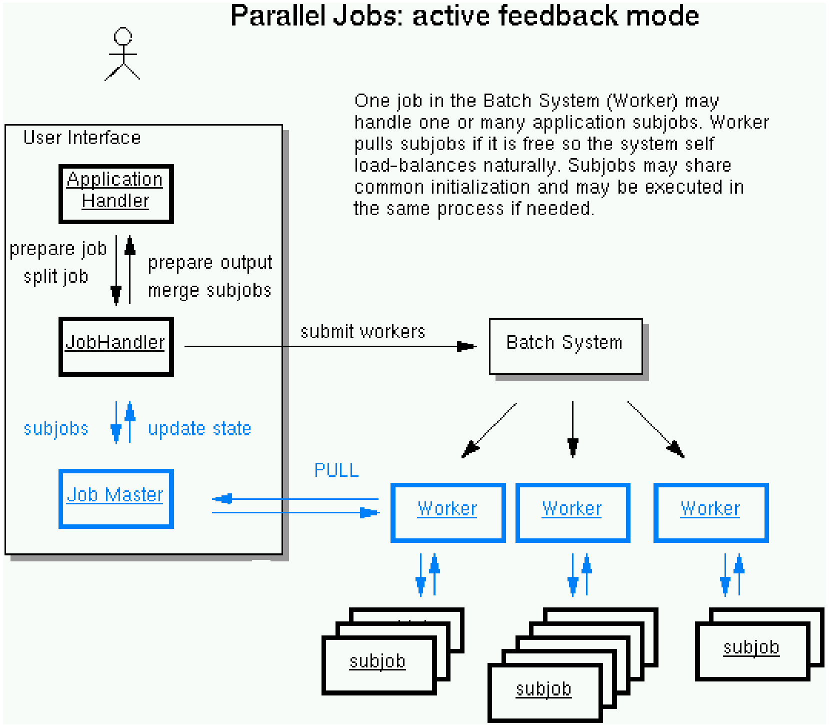
Analysis jobs

- Characteristics
  - Central database
  - Don Quijote Data mangement
  - Connects to several grid infrastructures
    - LCG
    - OSG
    - Nordugrid
- Analysis jobs have been demonstrated together with our colleagues from the production system

Check out the poster

DIANE

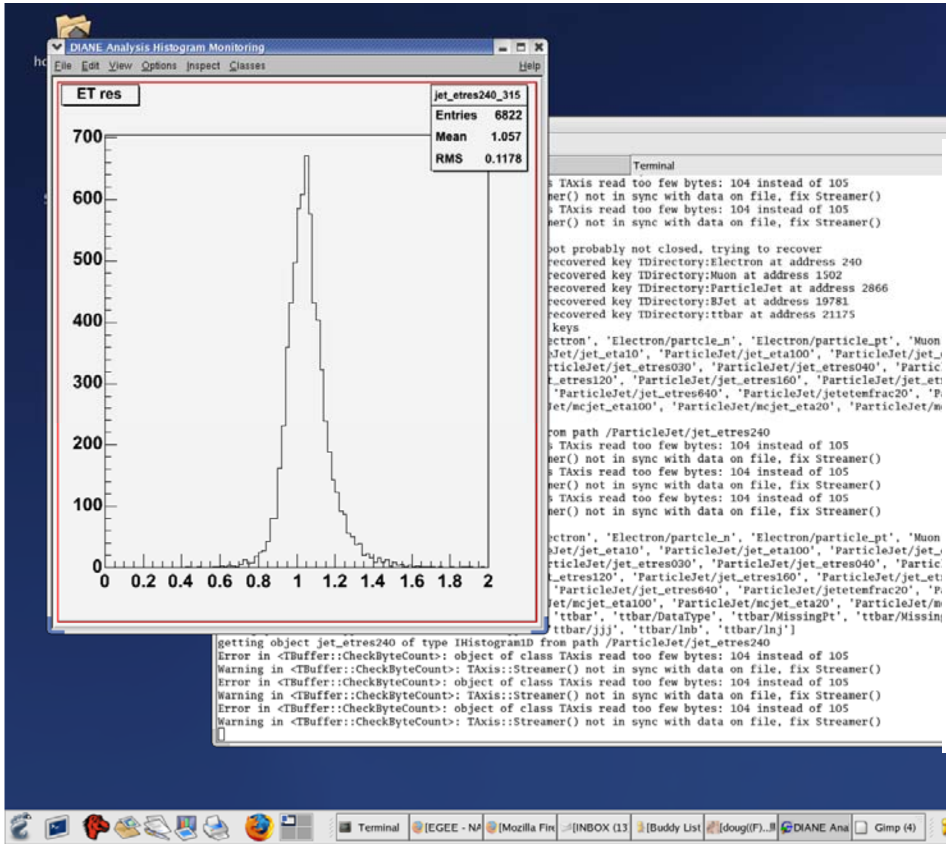



Parallel Jobs: active feedback mode

One job in the Batch System (Worker) may handle one or many application subjobs. Worker pulls subjobs if it is free so the system self load-balances naturally. Subjobs may share common initialization and may be executed in the same process if needed.

Was already mentioned today. Being integrated with GANGA

DIANE on gLite running Athena

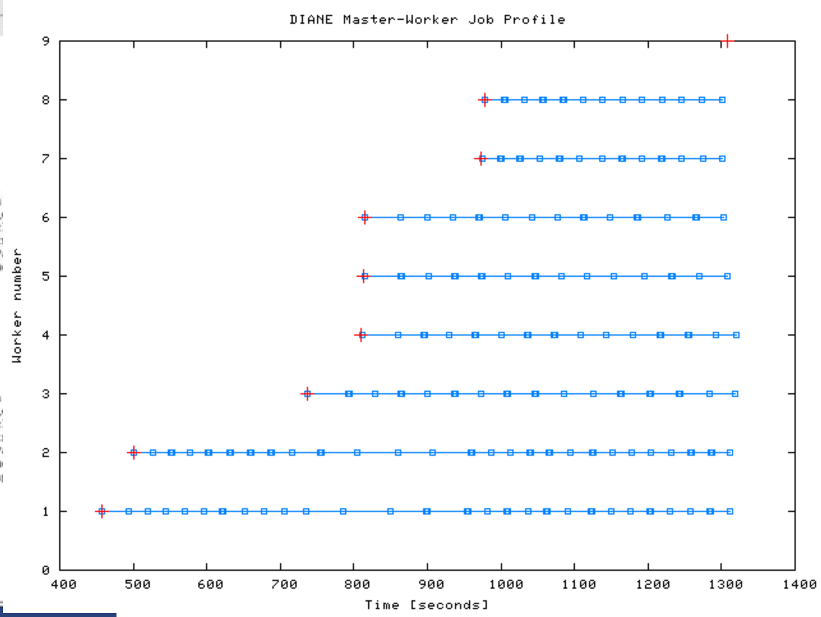


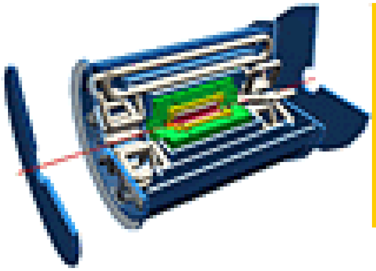
```
s TAxis read too few bytes: 104 instead of 105
ner() not in sync with data on file, fix Streamer()
s TAxis read too few bytes: 104 instead of 105
ner() not in sync with data on file, fix Streamer()

oot probably not closed, trying to recover
recovered key TDirectory:Electron at address 240
recovered key TDirectory:Muon at address 1502
recovered key TDirectory:ParticleJet at address 2866
recovered key TDirectory:BJet at address 19781
recovered key TDirectory:ttbar at address 21175
 keys
ectron', 'Electron/partcle_n', 'Electron/particle_pt', 'Muon
eJet/jet_eta10', 'ParticleJet/jet_eta100', 'ParticleJet/jet_
rticleJet/jet_etres030', 'ParticleJet/jet_etres040', 'Partic
t_etres120', 'ParticleJet/jet_etres160', 'ParticleJet/jet_et
 'ParticleJet/jet_etres640', 'ParticleJet/jetetemfrac20', 'P
Jet/mcjet_eta100', 'ParticleJet/mcjet_eta20', 'ParticleJet/m

rom path /ParticleJet/jet_etres240
s TAxis read too few bytes: 104 instead of 105
ner() not in sync with data on file, fix Streamer()
s TAxis read too few bytes: 104 instead of 105
ner() not in sync with data on file, fix Streamer()
s TAxis read too few bytes: 104 instead of 105
ner() not in sync with data on file, fix Streamer()

ectron', 'Electron/partcle_n', 'Electron/particle_pt', 'Muon
eJet/jet_eta10', 'ParticleJet/jet_eta100', 'ParticleJet/jet_
rticleJet/jet_etres030', 'ParticleJet/jet_etres040', 'Partic
t_etres120', 'ParticleJet/jet_etres160', 'ParticleJet/jet_et
 'ParticleJet/jet_etres640', 'ParticleJet/jetetemfrac20', 'P
Jet/mcjet_eta100', 'ParticleJet/mcjet_eta20', 'ParticleJet/m
 'ttbar', 'ttbar/DataType', 'ttbar/MissingPt', 'ttbar/Missin
'ttbar/jjj', 'ttbar/lnb', 'ttbar/lnj']
getting object jet_etres240 of type IHistogram1D from path /ParticleJet/jet_etres240
Error in <TBuffer::CheckByteCount>: object of class TAxis read too few bytes: 104 instead of 105
Warning in <TBuffer::CheckByteCount>: TAxis::Streamer() not in sync with data on file, fix Streamer()
Error in <TBuffer::CheckByteCount>: object of class TAxis read too few bytes: 104 instead of 105
Warning in <TBuffer::CheckByteCount>: TAxis::Streamer() not in sync with data on file, fix Streamer()
Error in <TBuffer::CheckByteCount>: object of class TAxis read too few bytes: 104 instead of 105
Warning in <TBuffer::CheckByteCount>: TAxis::Streamer() not in sync with data on file, fix Streamer()
```





Further plans

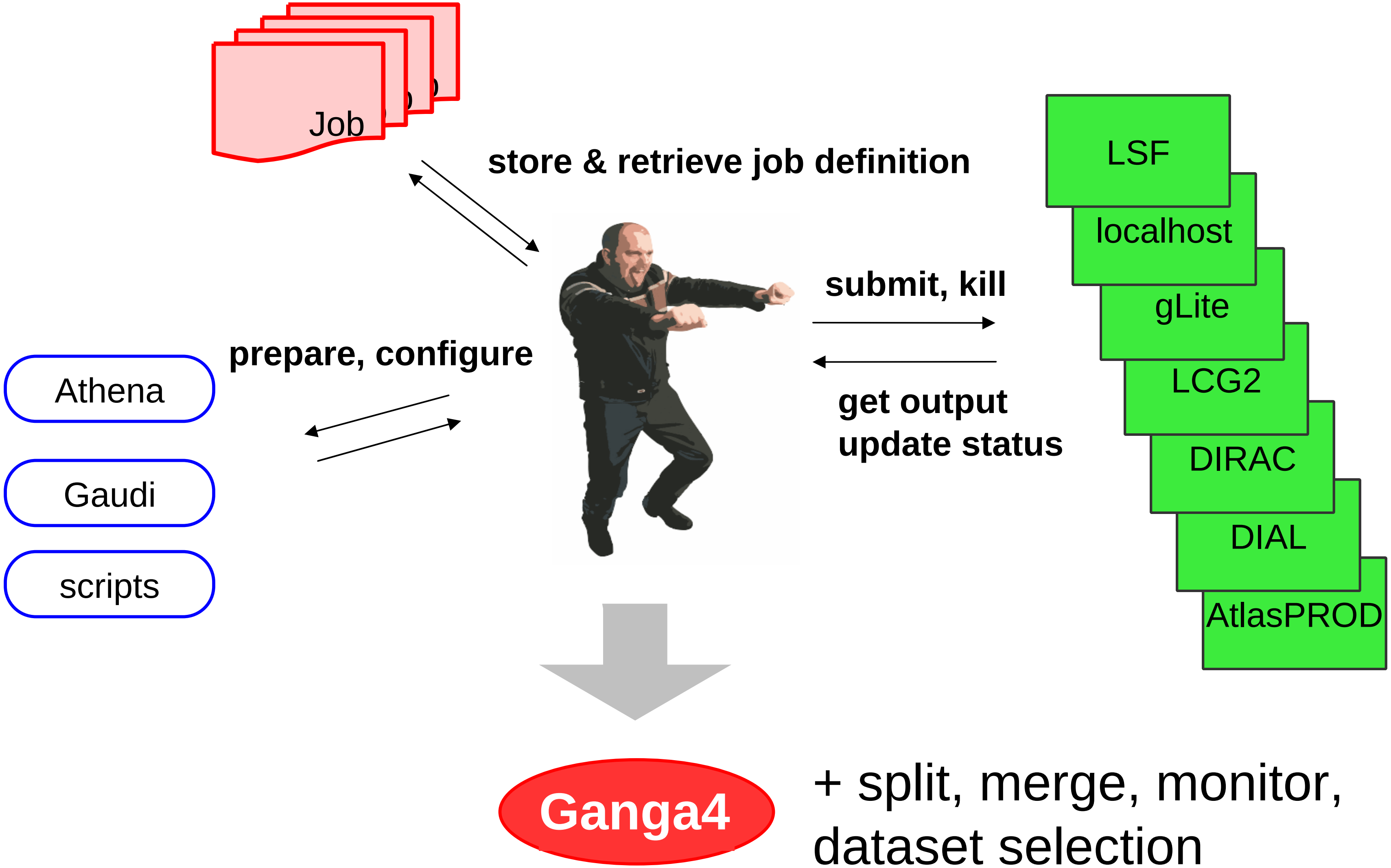
- New assessment of ATLAS Distributed Analysis after the review
  - ARDA has now a coordinating role for ATLAS Distributed Analysis
- Close collaboration with ATLAS production system and LCG/EGEE taskforce
- Close collaboration with GANGA and GridPP
- New players: Panda
  - OSG effort for Production and Distributed Analysis

LHCb

- Prototype is GANGA – A GUI for the GRID
- GANGA by itself is a joint project between ATLAS and LHCb
- In LHCb DIRAC, the LHCb production system, is used as a backend to run analysis jobs

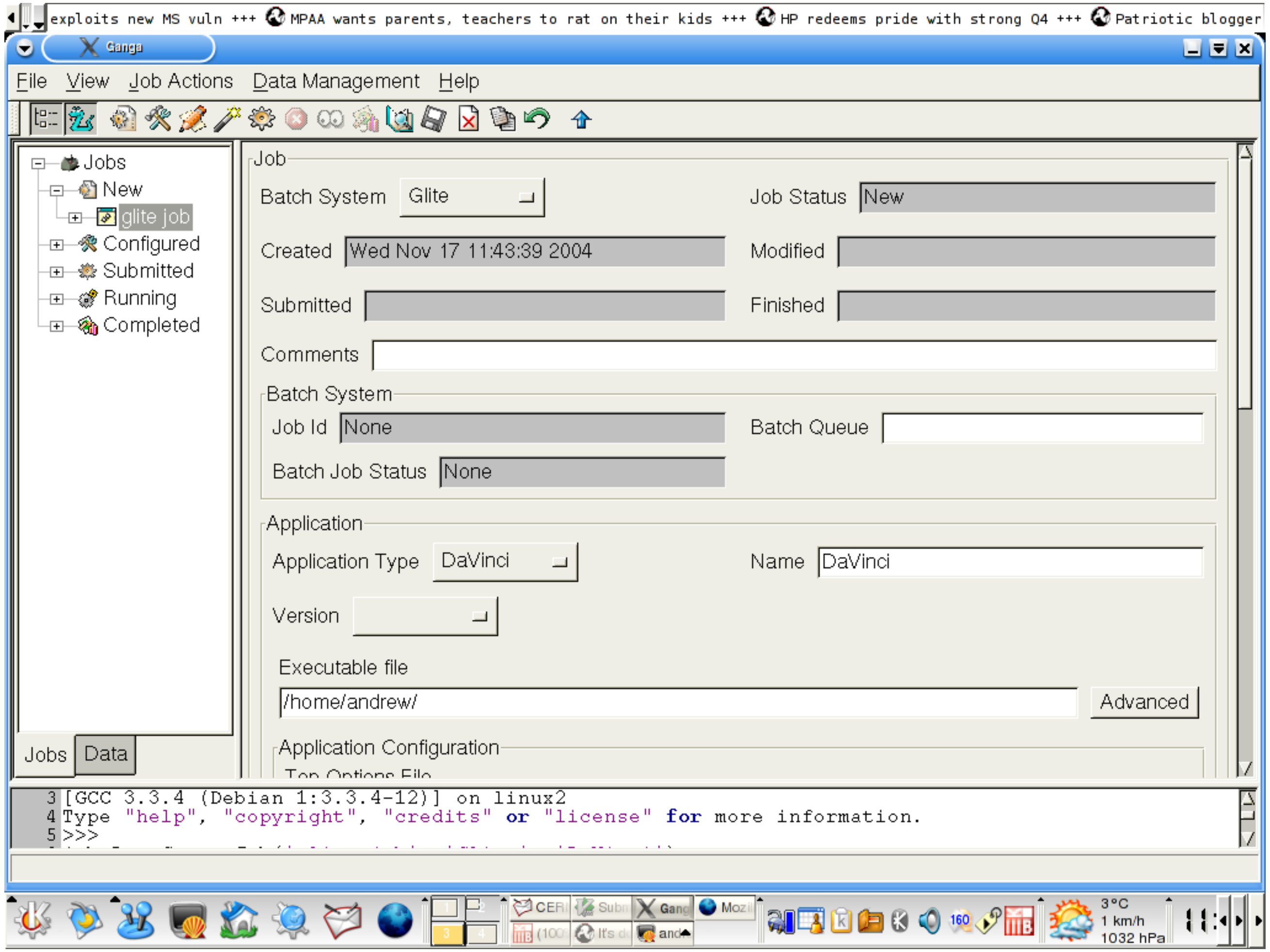
More details on the Poster

What is GANGA ?



+ split, merge, monitor, dataset selection

GANGA 3 - The current release



exploits new MS vuln +++ MPAA wants parents, teachers to rat on their kids +++ HP redeems pride with strong Q4 +++ Patriotic blogger

Job

Batch System: Glite

Job Status: New

Created: Wed Nov 17 11:43:39 2004

Modified:

Submitted:

Finished:

Comments:

Batch System

Job Id: None

Batch Queue:

Batch Job Status: None

Application

Application Type: DaVinci

Name: DaVinci

Version:

Executable file: /home/andrew/

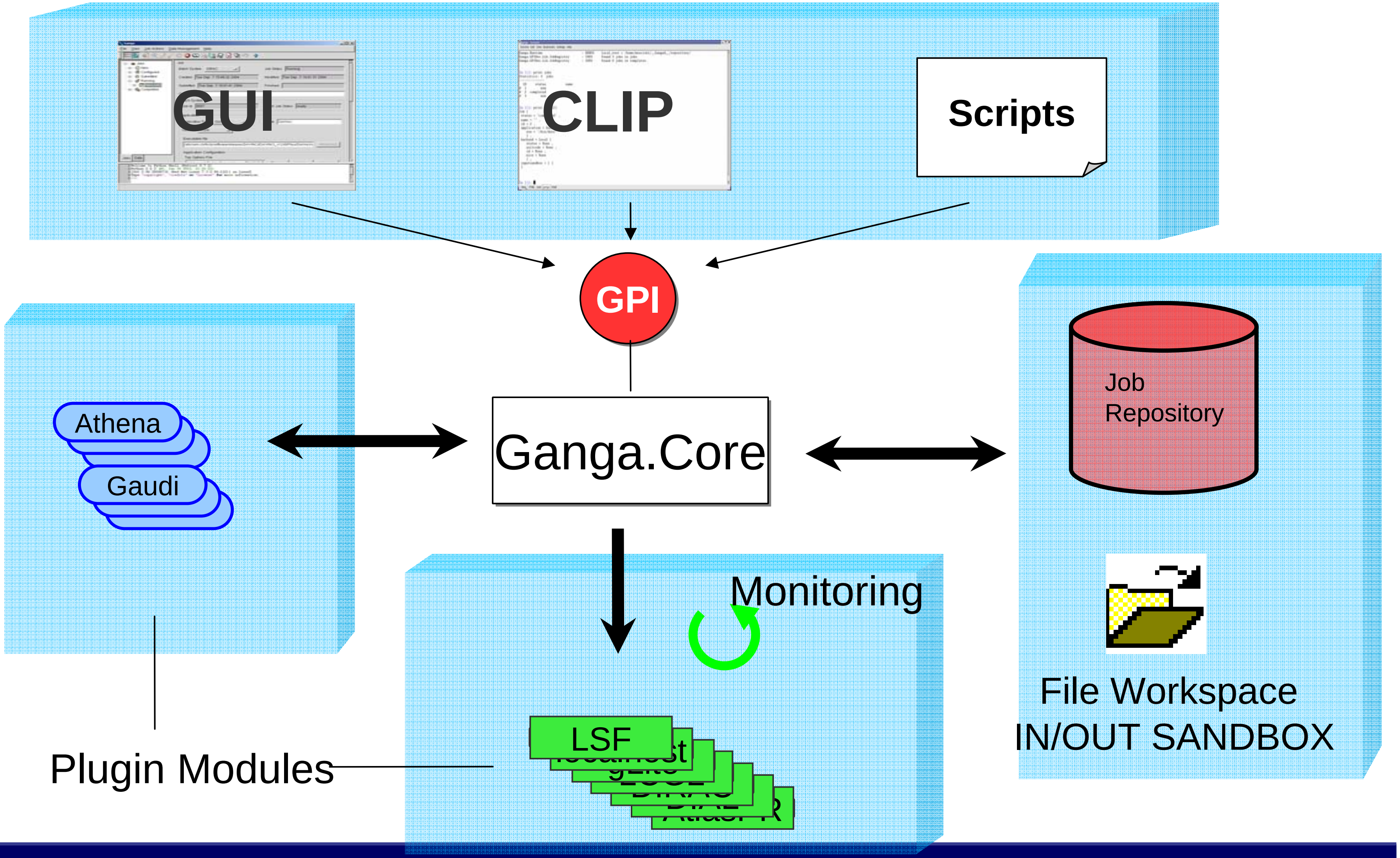
Application Configuration

Top Options File

```
[GCC 3.3.4 (Debian 1:3.3.4-12)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

The current release (version 3) is a GUI Application

Architecture for Version 4

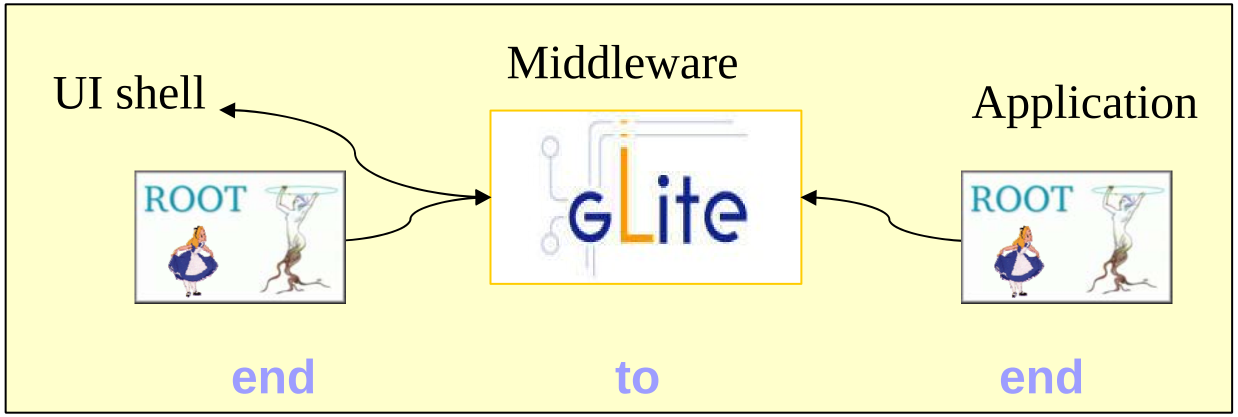


ALICE prototype

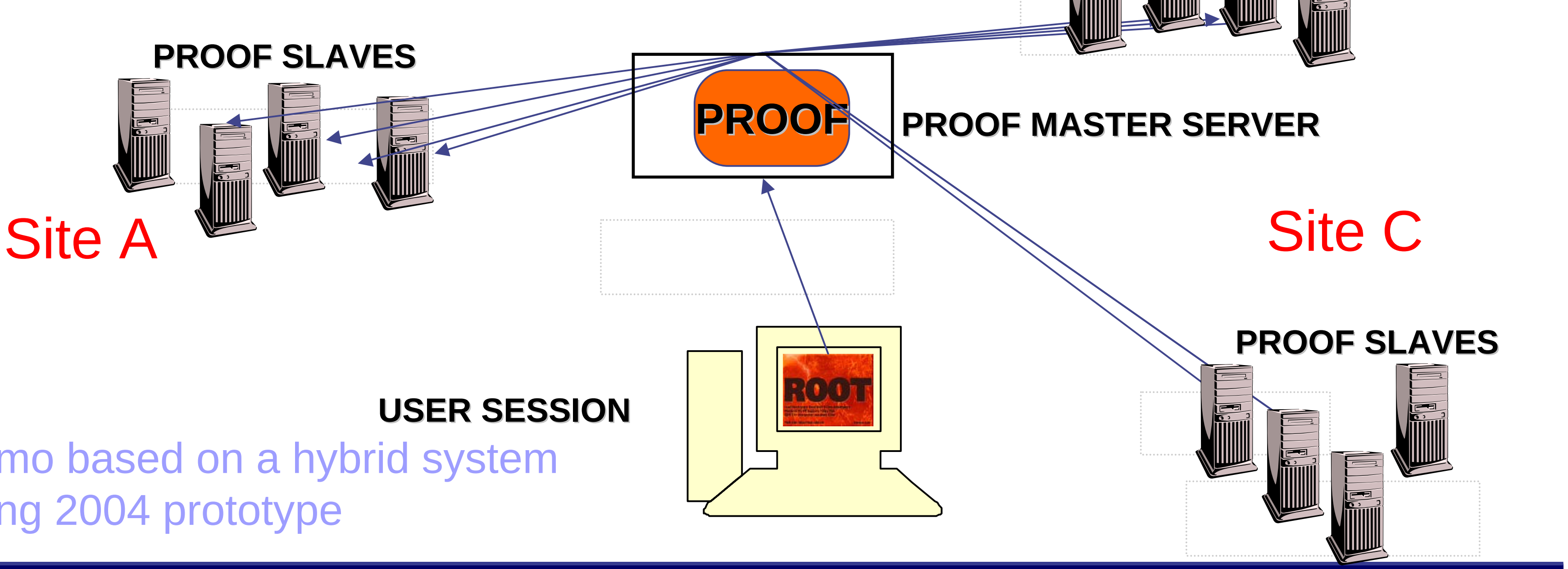
ROOT and PROOF

- **ALICE provides**
  - **the UI**
  - **the analysis application (AliROOT)**

- **GRID middleware gLite provides all the rest**



- **ARDA/ALICE is evolving the ALICE analysis system**



Demo based on a hybrid system using 2004 prototype

ARDA shell + C/C++ API

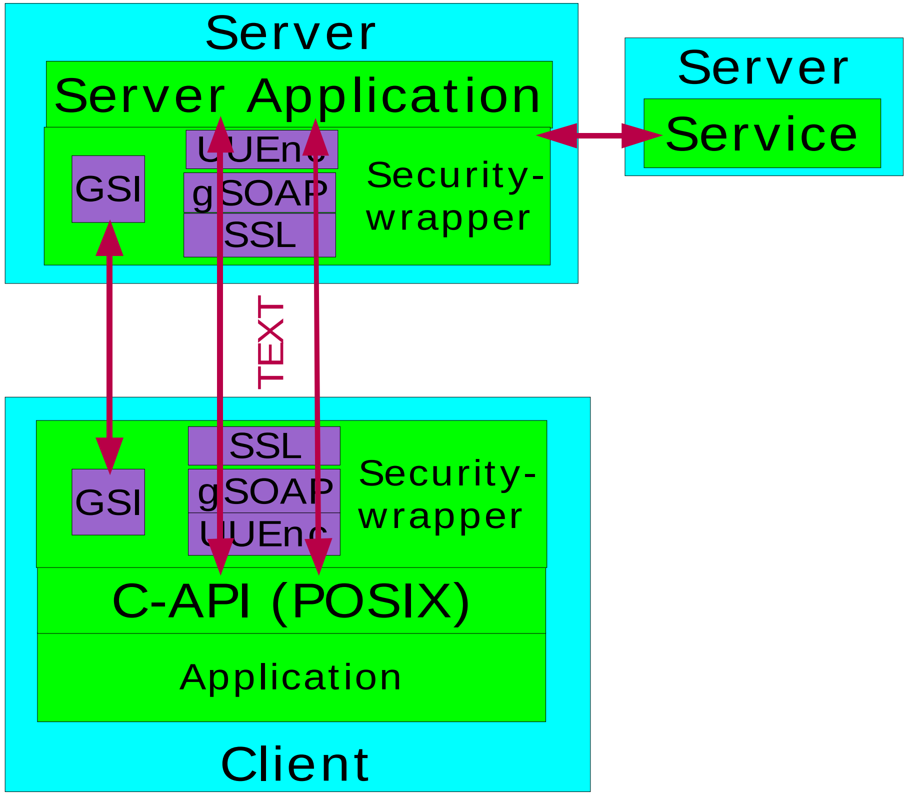
C++ access library for g**L**ite has been developed by ARDA

- 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



Current Status

- Developed gLite C++ API and API Service
  - providing generic interface to any GRID service
- C++ API is integrated into ROOT
  - In the ROOT CVS
  - 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
  - More powerful than the original shell
  - In use in ALICE
  - Considered a “generic” mw contribution (essential for ALICE, interesting in general)
- First version of the interactive analysis prototype 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

Feedback to gLite

- **2004:**
  - **Prototype available (CERN + Madison Wisconsin)**
  - **A lot of activity (4 experiments prototypes)**
  - **Main limitation: size**
    - **Experiments data available! ☺**
    - **Just an handful of worker nodes ☹**

Access granted on May 18th 2004! ☺

- **2005:**
  - **Coherent move to prepare a gLite package to be deployed on the pre-production service**
    - **ARDA contribution:**
    - **Mentoring and tutorial**
    - **Actual tests!**
  - **Lot of testing during 05Q1**
  - **PreProduction Service is about to start!**

Data Management

- Central component
  - Early tests started in 2004

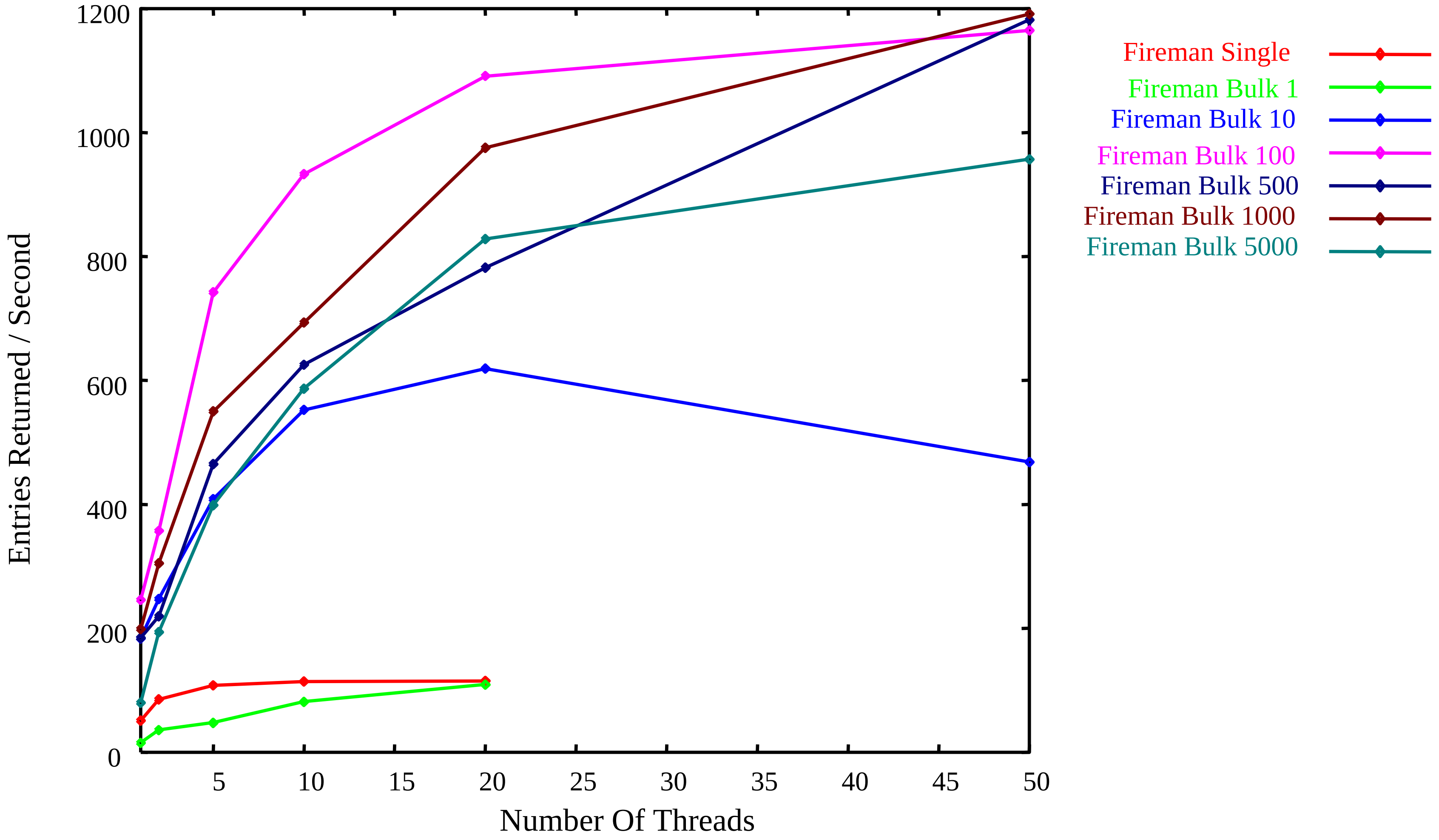
- Two main components:
  - gLiteIO (protocol + server to access the data)
  - FiReMan (file catalogue)

- Both LFC and FiReMan offer large improvements over RLS
  - LFC is the most recent LCG2 catalogue

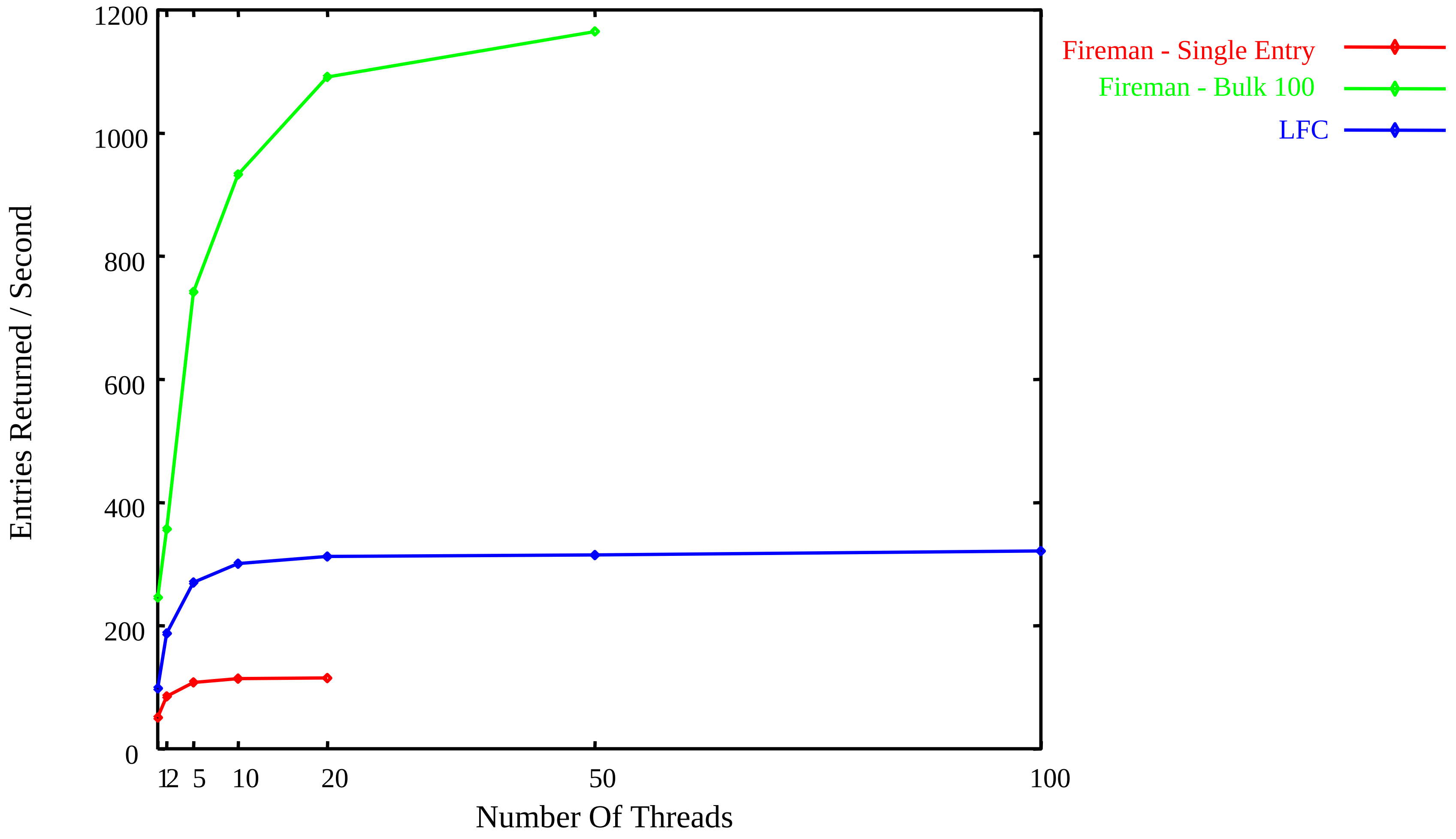
- Still some issues remaining:
  - Scalability of FiReMan
  - Bulk Entry for LFC missing
  - More work needed to understand performance and bottlenecks
  - Need to test some real Use Cases
  - In general, the validation of DM tools takes time!

- Reference – Presentation at ACAT 05, DESY Zeuthen, Germany http://cern.ch/munro/papers/acat_05_proceedings.pdf

Query Rate for an LFN



Comparison with LFC



Workload Management

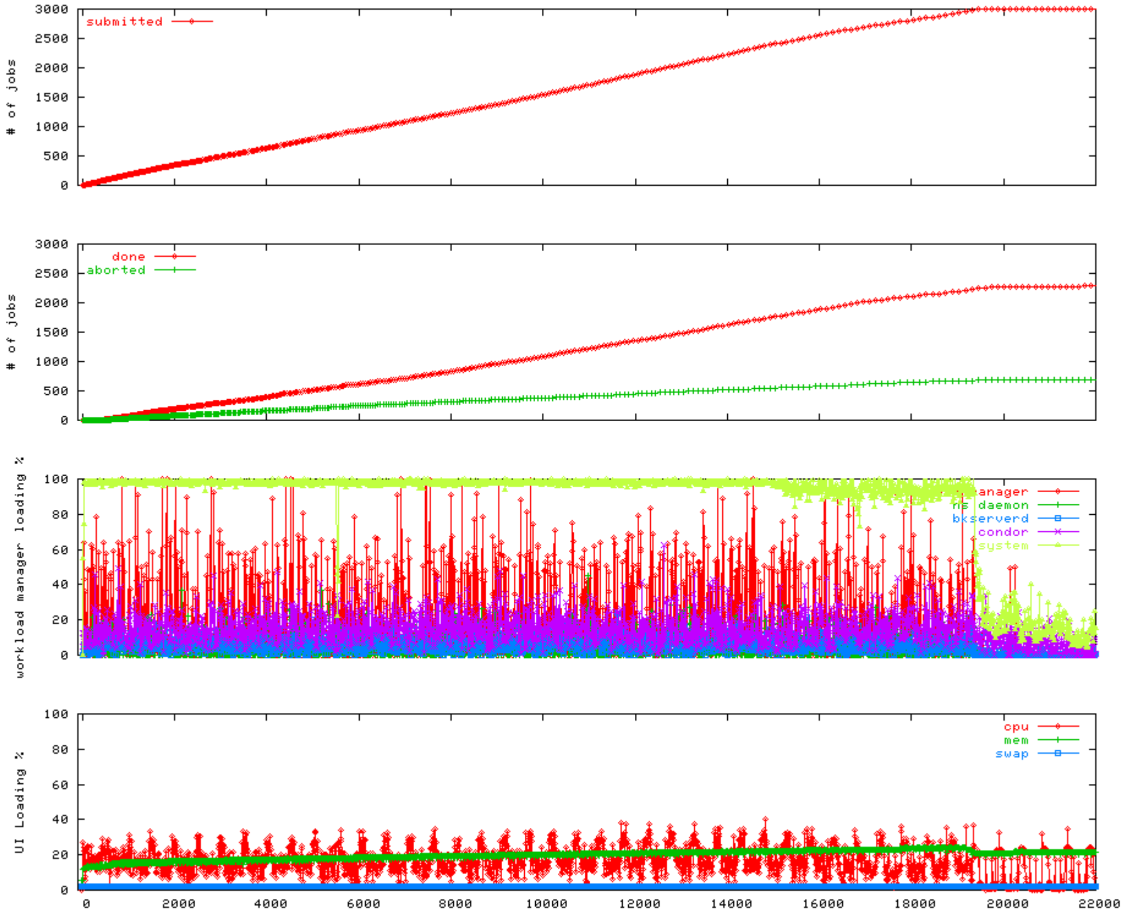
- A systematic evaluation of the WMS performance in terms of the
  - job submission rate (UI - RB)
  - job dispatching rate (RB - CE)

- The first measurement has been done on both gLite prototype and LCG2 in the context of ATLAS; however, the test scenario is generic to all experiments
  - Simple helloWorld job without any InputSandbox
  - Single client, multi-thread job submission
  - Monitoring the overall Resource Broker (RB) loading as well as the CPU/memory usages of each individual service on RB.

- Continuing the evaluations on the effects of
  - Logging and Bookkeeping (L&B) loading
  - InputSandbox
  - gLite bulk submission feature

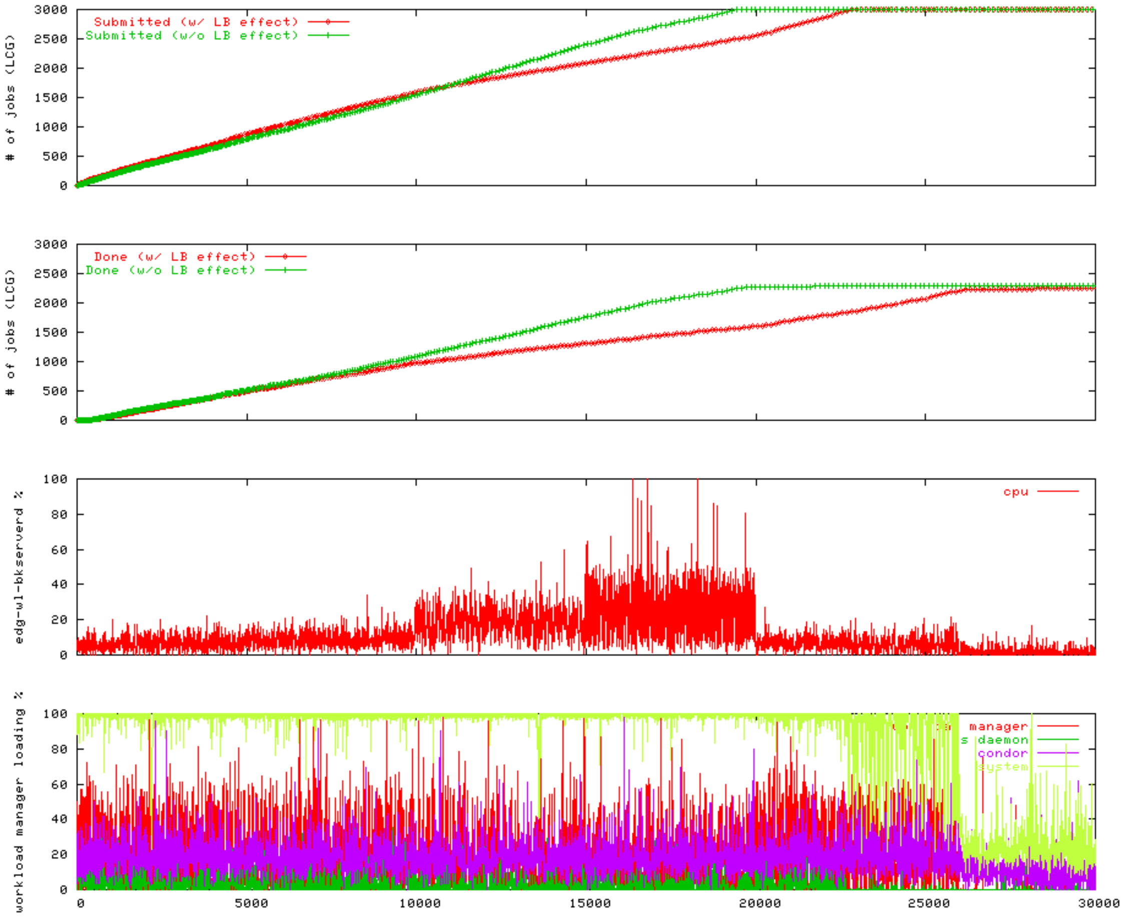
- **Reference:** http://cern.ch/LCG/activities/arda/public_docs/2005/Q3/WMS Performance Test Plan.doc

WMS Performance Test



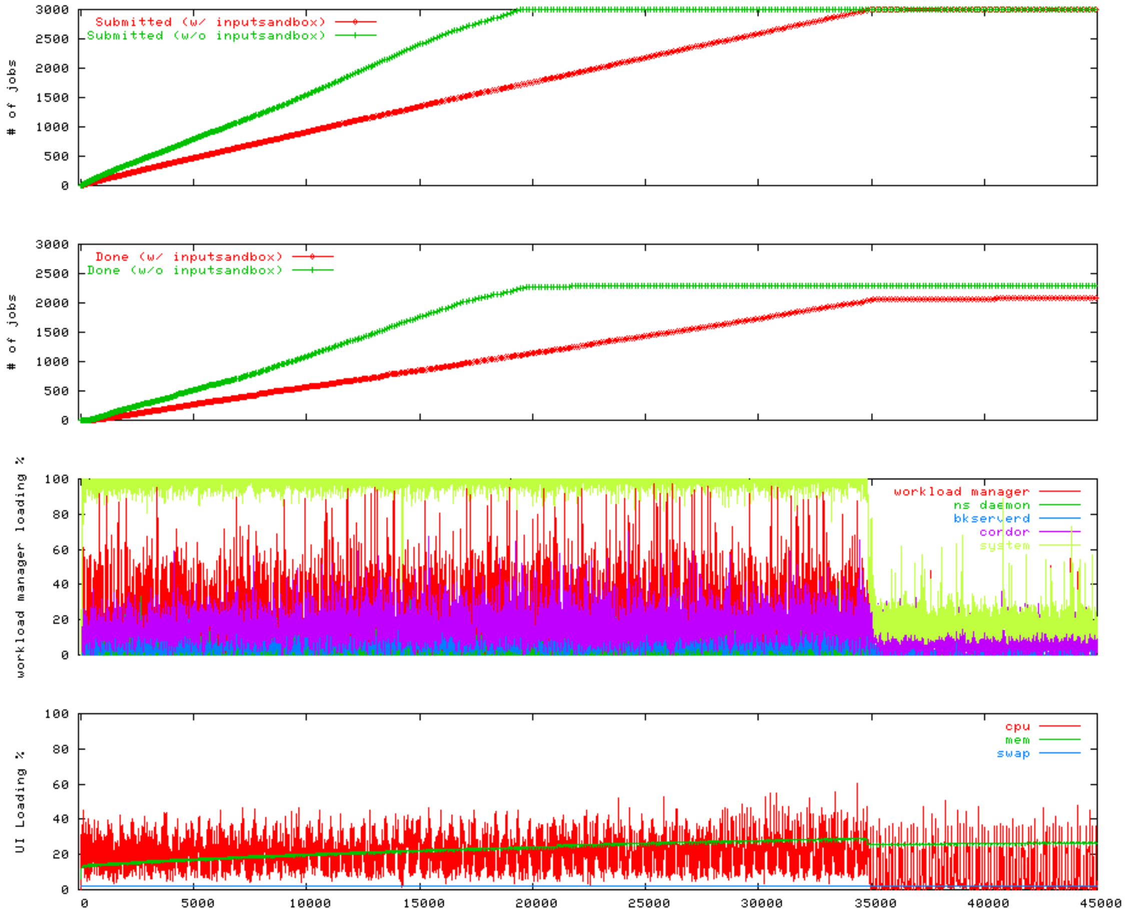
- 3000 helloWorld jobs are submitted by 3 threads from the LCG UI in Taiwan
- Submission rate ~ 0.15 jobs/sec (6.6 sec/job)
- After about 100 sec, the first job reaches the done status
- Failure rate ~ 20 % (RetryCount = 0)

Effects of loading the Logging and Bookkeeping



- 3000 helloWorld jobs are submitted by 3 threads from the LCG UI in Taiwan
- In parallel with job submission, the L&B is also loaded up to 50 % CPU usage in 3 stages by multi-thread L&B queries from another UI
- Slowing down the job submission rate (from 0.15 jobs/sec to 0.093 jobs/sec)
- Failure rate is stable to ~ 20 % (RetryCount = 0)

Effects of Input Sandbox

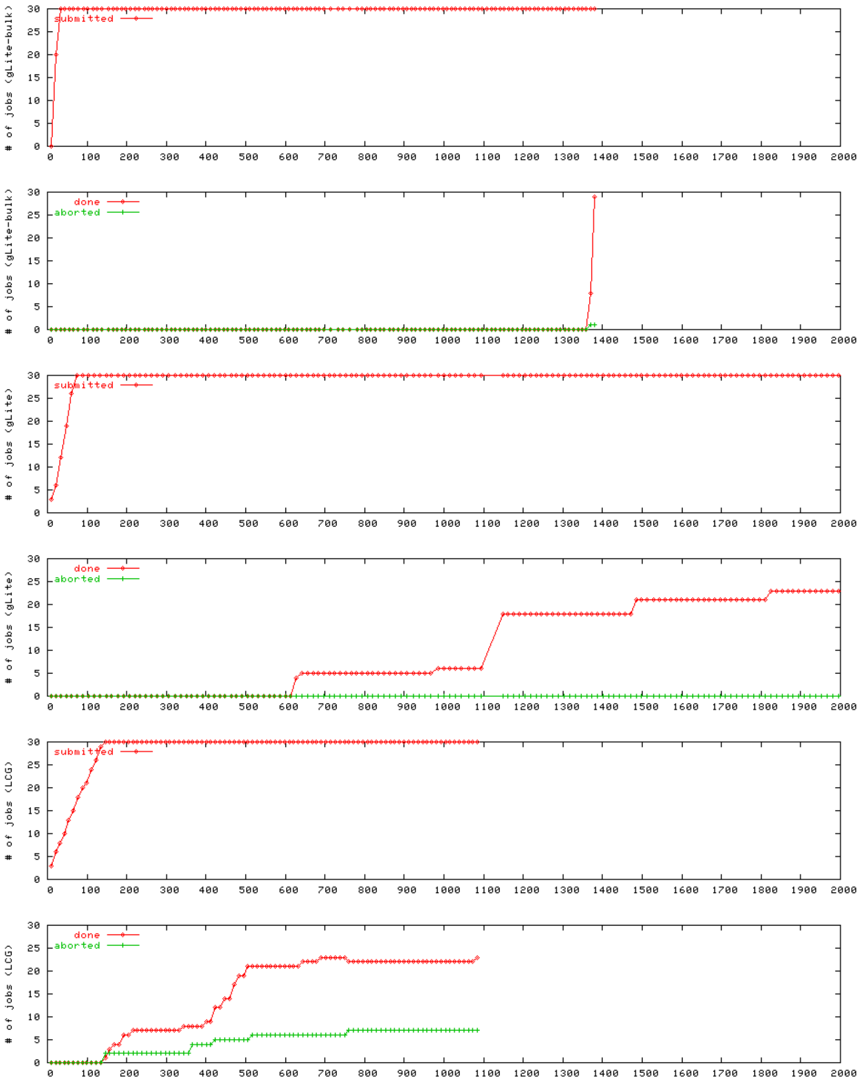


- 3000 jobs with InputSandbox are submitted by 3 threads from the LCG UI in Taiwan

- InputSandbox is taken from the ATLAS production job (~ 36 KBytes per job)

- Slowing down the job submission rate (from 0.15 jobs/sec to 0.08 jobs/sec)

gLite (v1.3) Bulk Submission



- 30 helloWorld jobs are submitted by 3 threads on LCG2 and gLite prototype.
- The comparison between LCG2 and gLite is unfair due to the hardware differences between the RBs.
- On gLite, the bulk submission rate is about 3 times faster than the non-bulk submission.

AMGA - Metadata services on the Grid

- Simple database for use on the GRID
  - Key value pairs
  - GSI security

- gLite has provided a prototype for the EGEE Biomed communit
  - ARDA (HEP) Requirements were not all satisfied by that early version

- Discussion in LCG and EGEE and UK GridPP Metadata group
- Testing of existing implementations in experiments
- Technology investigation

- ARDA Prototype
  - AMGA is now part of gLite Release

- Reference:

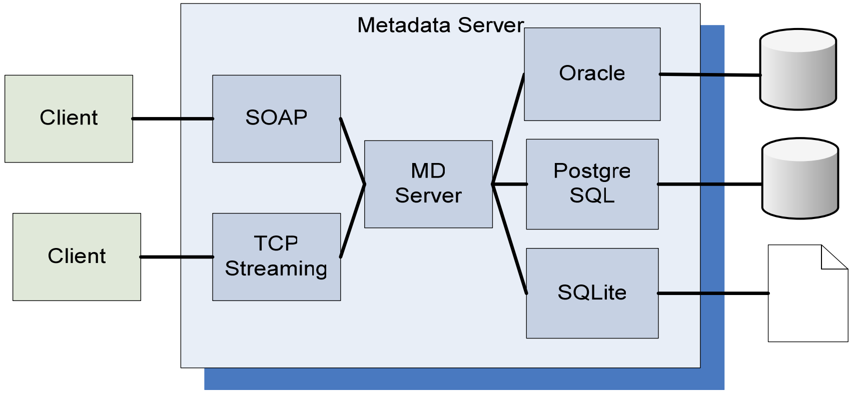
ARDA Implementation

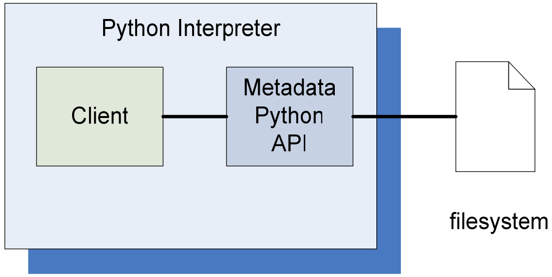
- Prototype
  - Validate our ideas and expose a concrete example to interested parties

- Multiple back ends
  - Currently: Oracle, PostgreSQL, MySQL, SQLite

- Dual front ends
  - TCP Streaming
    - Chosen for performance
  - SOAP
    - Formal requirement of EGEE
    - Compare SOAP with TCP Streaming

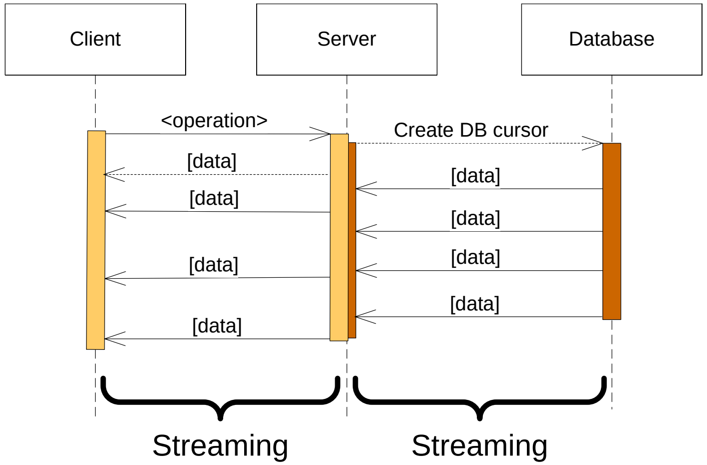
- Also implemented as standalone Python library
  - Data stored on the file system





Dual Front End

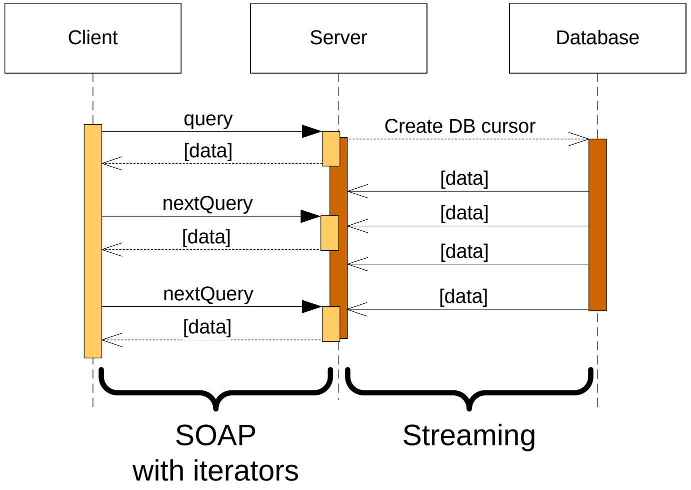
- Text based protocol



- Data streamed to client in single connection
- Impl
  - Perl, Ruby

clean way to study performance implications of protocols...

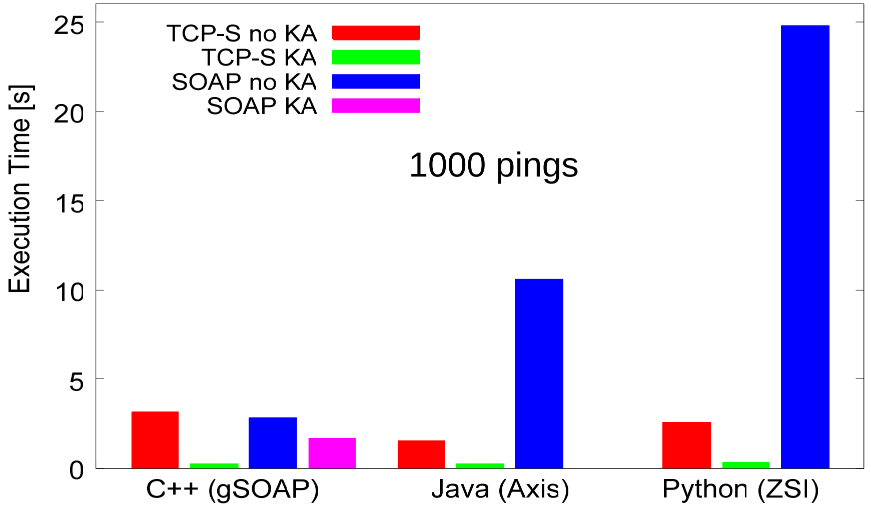
- Most operations are SOAP calls



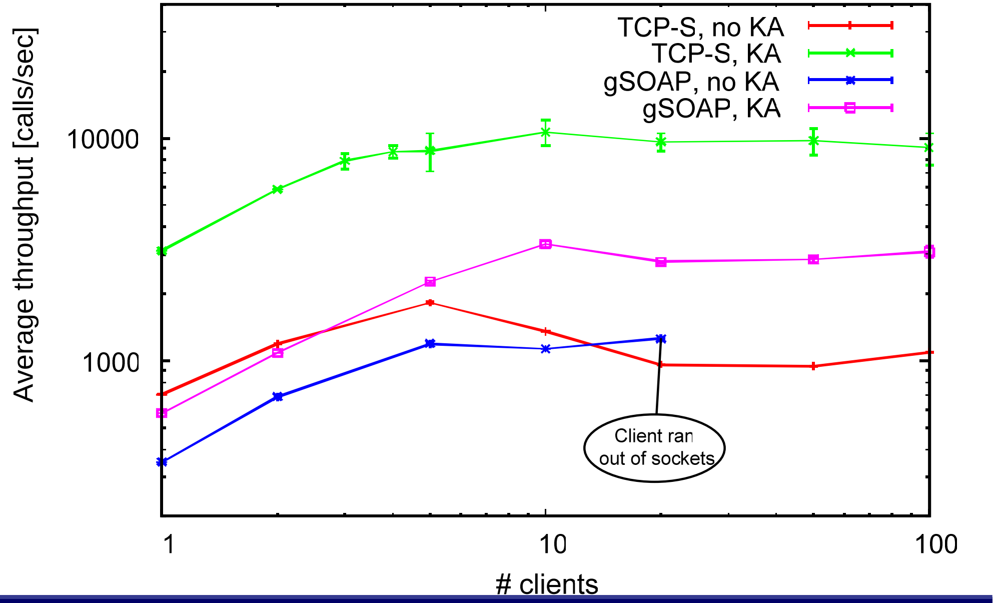
- Based on iterators
  - Session created
  - Return initial chunk of data and session token
  - Subsequent request: client calls nextQuery() using session token
  - Session closed when:
    - End of data
    - Client calls endQuery()
    - Client timeout
- Implementations
  - Server – gSOAP (C++).
  - Clients – Tested WSDL with gSOAP, ZSI (Python), AXIS (Java)

More data coming…

- Test protocol performance
  - No work done on the backend
  - Switched 100Mbits LAN
- Language comparison
  - TCP-S with similar performance in all languages
  - SOAP performance varies strongly with toolkit
- Protocols comparison
  - Keep alive improves performance significantly
  - On Java and Python, SOAP is several times slower than TCP-S



- Measure scalability of protocols
  - Switched 100Mbits LAN
- TCP-S 3x faster than gSoap (with keepalive)
- Poor performance without keepalive
  - Around 1.000 ops/sec (both gSOAP and TCP-S)



Current Uses of AMGA

- Evaluated by LHCb bookkeeping
  - Migrated bookkeeping metadata to ARDA prototype
    - 20M entries, 15 GB
  - Interface found to be complete
  - ARDA prototype showing good scalability
- Ganga (LHCb, ATLAS)
  - User analysis job management system
  - Stores job status on ARDA prototype
  - Highly dynamic metadata
- AMGA is now part of gLite Release
- Integrated with LFC (works side by side)

Summary

- Experiment prototypes
  - CMS: ASAP – now being integrated
  - ATLAS: DIAL - move now to Production System
  - LHCb: GANGA
  - ALICE: PROOF

- Feedback to the Middleware
  - Data management
  - Workload Management

- AMGA Metadata catalog now part of gLite