# Testing of Distributed User Analysis on the Grid

*A. Maier/CERN/IT*

*EGEE Conference 2009, Barcelona*

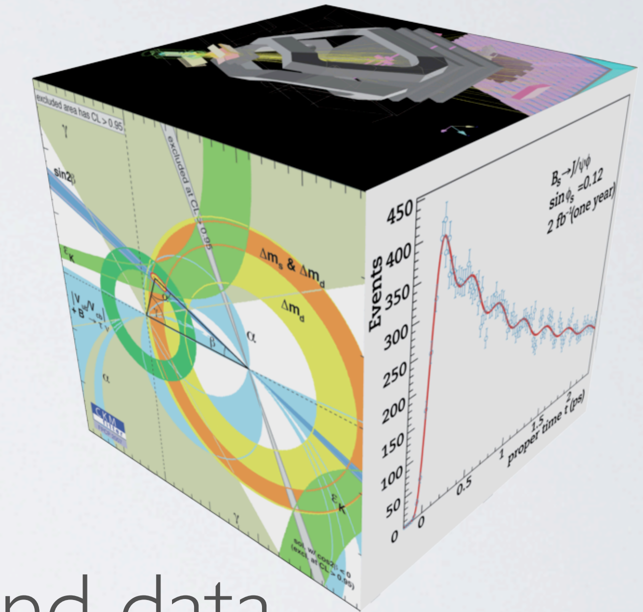
www.eu-egee.org

# OVERVIEW

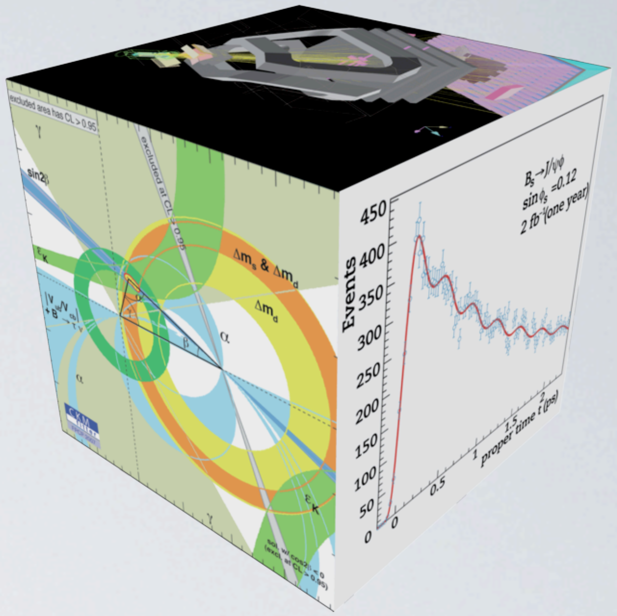
- Testing of user analysis
- Site availability testing using the GangaRobot
- Individual site stress testing using HammerCloud

# TESTING OF USER ANALYSIS

- LHC data analysis will be carried out on the Grid
- Users depend on the availability of Grid resources
- Key questions:
  - WHY test of distributed user analysis?
  - WHAT is the difference between  user analysis and data production?
  - HOW to test distributed user analysis?

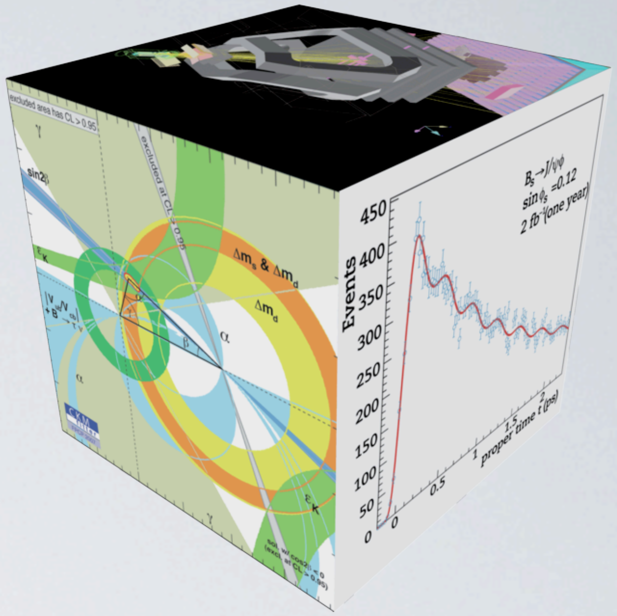


# WHY?



- Data productions (e.g. MC, reprocessing) are well established and are working well
  - e.g. CCRC 09
  - Data productions are orderly, well controlled by a few experts and production managers
- But user analysis is different ...

# WHY?

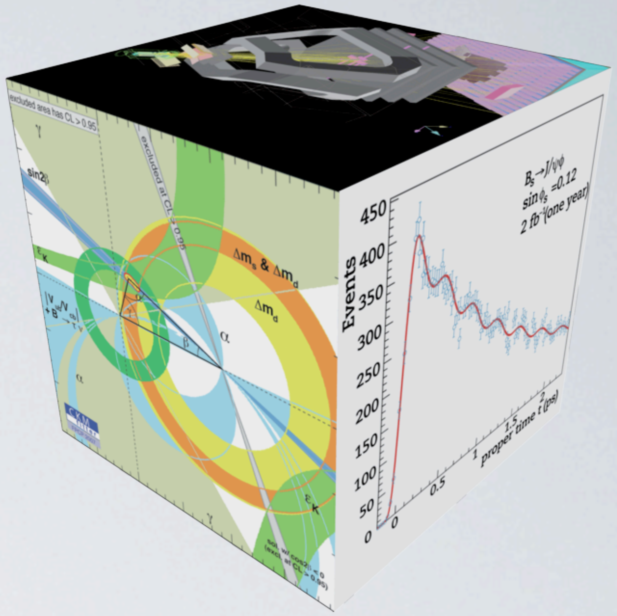


User analysis is usually:

- unscheduled,
- burst like (e.g. before conferences),
- accesses similar files (e.g Higgs analysis files)
- heterogeneous (e.g. reads large amount of data, creates various amount of output)

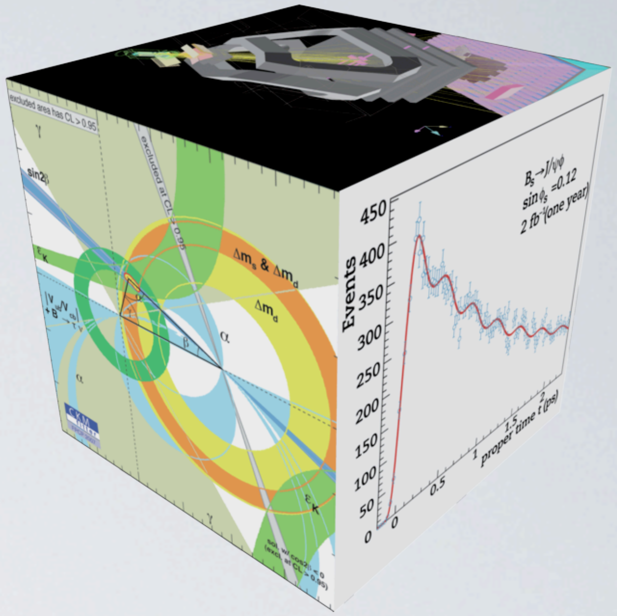
AND Users expect fast turnaround, high reliability, ease of use, want all jobs to end at the same time

# WHAT?



- User analysis tests should use the same tools as employed by the users
  - ➡ no special setups or tools
- Tests should be automated
- Use ‘real’ analysis examples (in close collaboration with the analysis teams):
  - ➡ typical data files
  - ➡ running for a ‘realistic’ amount of time

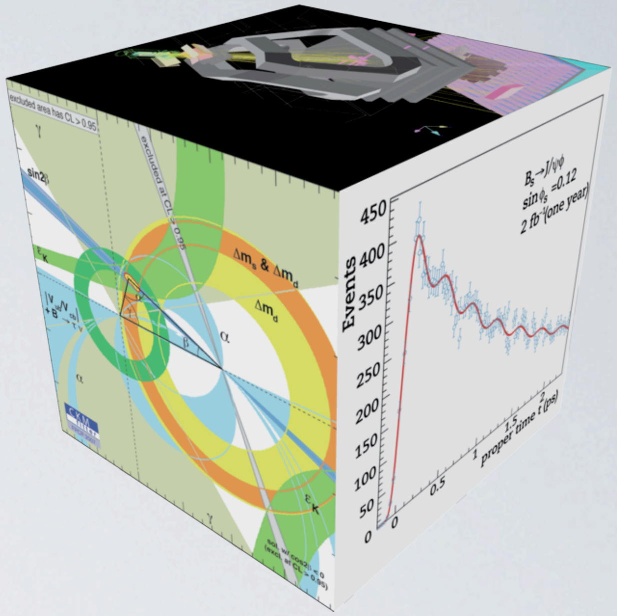
# WHAT?



Test should be done from the users perspective:

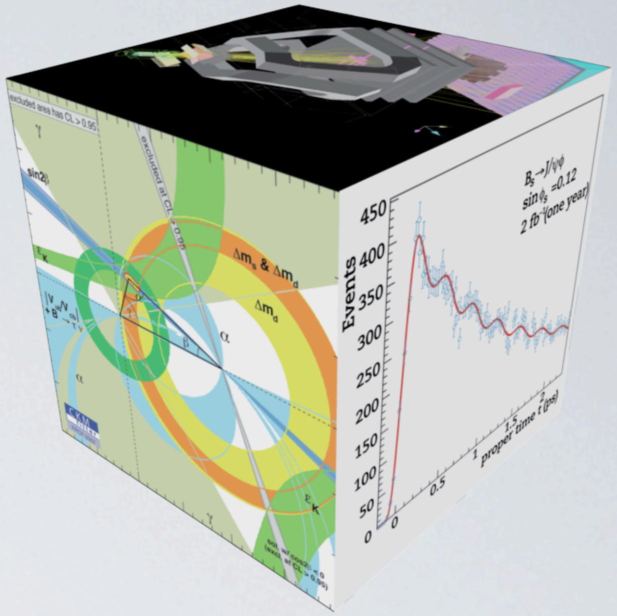
- Did the job run over all the data (are files missing?)?
- Did the outputs return properly?
- Did all jobs run, which ones crashed and why?
- Was the required time in agreement with normal analysis turn-around times?

# HOW?



- All LHC experiments have systems for testing user analysis jobs
- Systems are specific to the experiments
- Talk will be restricted to examples from ATLAS and LHCb
  - GangaRobot and HammerCloud

# TYPES OF TESTS



Automated end-to-end testing of user analysis:

- Test if the entire chain works
- Submission, running, retrieving of results
- Example: GangaRobot

Automated stress testing of individual sites:

- Targets individual sites or clouds
- Example: HammerCloud

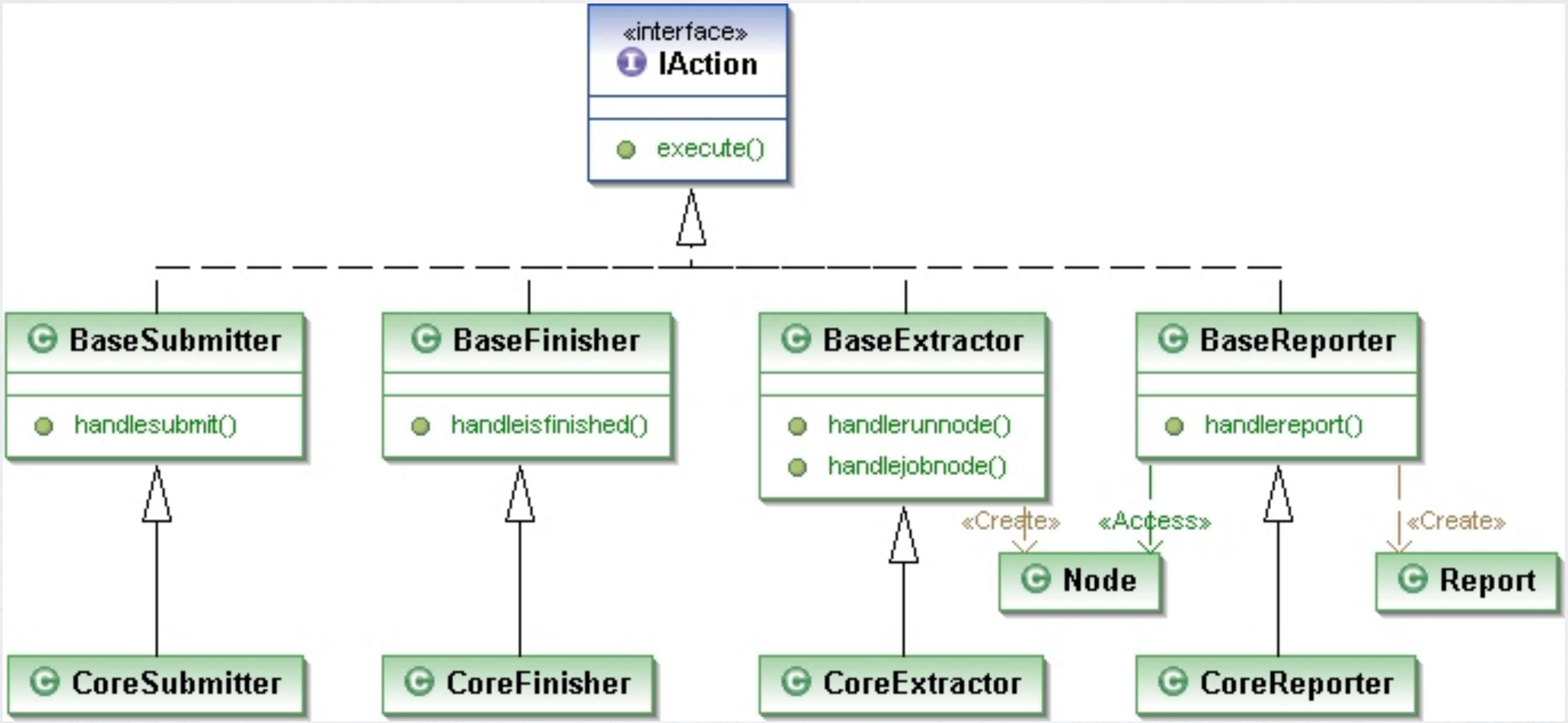
# GANGAROBOT

- GangaRobot is a tool for running a list of actions within a Ganga session
- Actions are instructions to be made in order to perform complex repetitive use-cases
- Actions are defined by implementations of an action interface, e.g.:

→ Submit → Wait → Extract information → Report

# GANGAROBOT

Abstract base action implementations provide a basis for implementing submit / extract / report actions



# GANGAROBOT

Set-up a configuration file with the action classes:

```
[Robot]
Driver_Run = ['submit', 20, 'finish', 'extract', 'report']
FileEmailer_From = sender@domain.org
FileEmailer_Recipients = recipient@domain.org
FileEmailer_TextFile = ~/gangadir/robot/report/${runid}.txt
FileEmailer_HtmlFile = ~/gangadir/robot/report/${runid}.html
Driver_Action_submit = GangaRobot.Lib.Core.CoreSubmitter.CoreSubmitter
Driver_Action_finish = GangaRobot.Lib.Core.CoreFinisher.CoreFinisher
Driver_Action_extract = GangaRobot.Lib.Core.CoreExtractor.CoreExtractor
Driver_Action_report = GangaRobot.Lib.Core.CoreReporter.CoreReporter
Driver_Action_email = GangaRobot.Lib.Ext.FileEmailer.FileEmailer
```

# GANGAROBOT

Suited to perform tasks involving:

- Submitting complex analysis jobs to the grid
- Extracting data about the jobs and the grid environment
- Extract data on the status, availability of result files
- Reporting statistics on the extracted data.

# GANGAROBOT

- Automatic testing by submitting analysis jobs to a large number of sites
- Submit regularly (3-4x /day) a small number of jobs/site (<10)
- Submit jobs trying to access different queues on the CE
- Submit jobs accessing the sandbox via the network or via download
- (…)

# GANGAROBOT

Will find problems in the entire chain:

- Data and software availability (e.g. missing files, missing software)
- Output retrieval and storing (e.g. missing histograms)

Submitted jobs are checked for:

- Exit codes, expected no. of events read, retrieval of output files (and file consistency check)

# GANGAROBOT SUMMARY

- Part of the standard Ganga distribution
- Automated end-to-end testing of user analysis
- Suited to perform complex tasks
- Easy to implement in user defined tasks as actions within a Ganga session

# HAMMERCLOUD

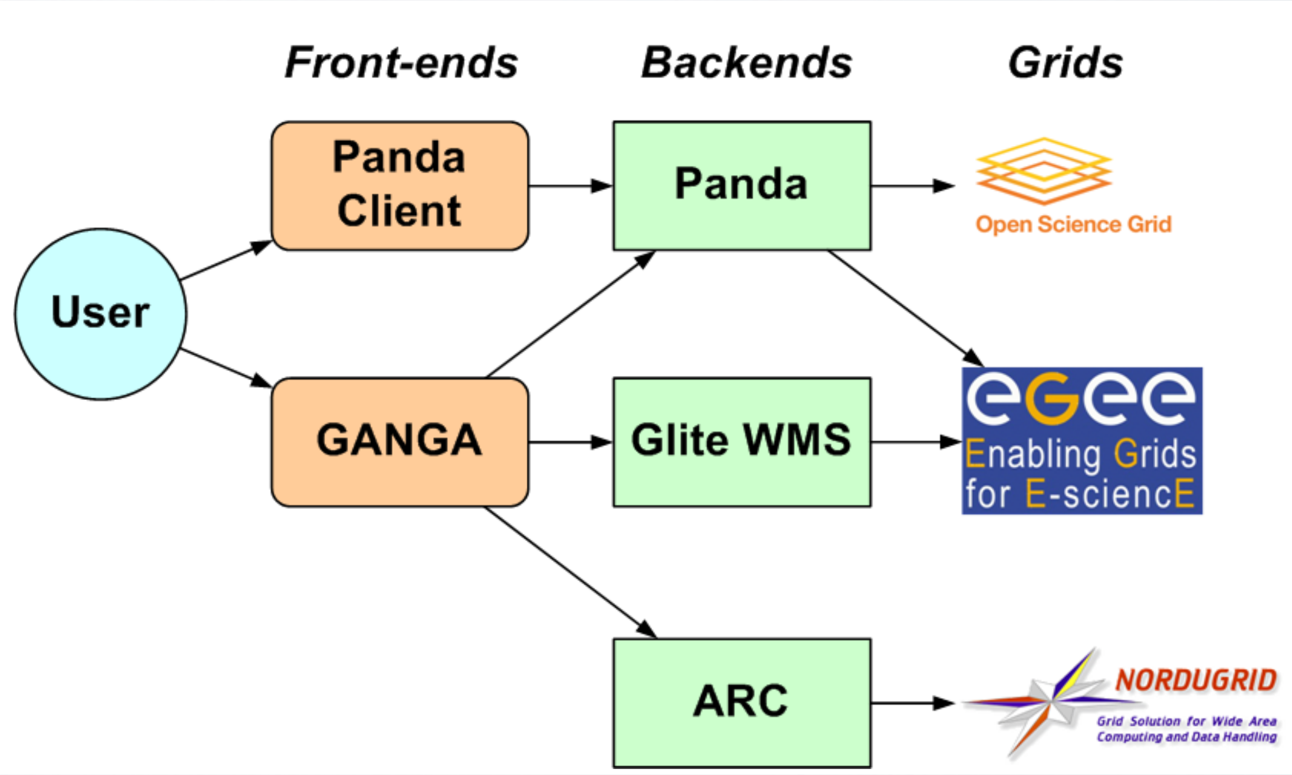
HammerCloud is a tool to stress test individual sites:

- Saturate site with large number of user jobs
- Tests if sites can cope with the large amount of jobs
- Tests configuration problems, e.g., bandwidth to deliver 'hot' files to a large amount of jobs

# HAMMERCLOUD

Uses real analysis jobs:

- submitted via Ganga to
  - Panda
  - gLite WMS
  - NorduGrid (ARC)
- Via the PanDA client to PanDA


Front-ends
Backends
Grids
Panda Client
Panda
Open Science Grid
User
GANGA
Glite WMS
eGee Enabling Grids for E-sciencE
ARC
NORDUGRID Grid Solution for Wide Area Computing and Data Handling

# HAMMERCLOUD TESTS

HammerCloud tests real analyses:

- AOD analysis, based on Athena UserAnalysis pkg, analyzing mainly muons:
  - Input data: muon AOD datasets, or other AODs if muons are not available
  - In principal, the results would be similar to any analysis where the file I/O is the bottleneck
- Reprocessed DPD analysis:
  - Intended to test the remote conditions database (at local Tier 1)

# HAMMERCLOUD TESTS

What metrics does HammerCloud measure?

- Exit status and log files
- CPU/Wallclock ratio, events per second
- Job timing: queue, input sandbox stage-in, Athena/CMT setup, LFC lookup, Athena exec, output storage
- Number of events and files processed (versus what was expected)
- Some local statistics (e.g. network and storage rates) are only available at site level monitoring (site contacts very important!)

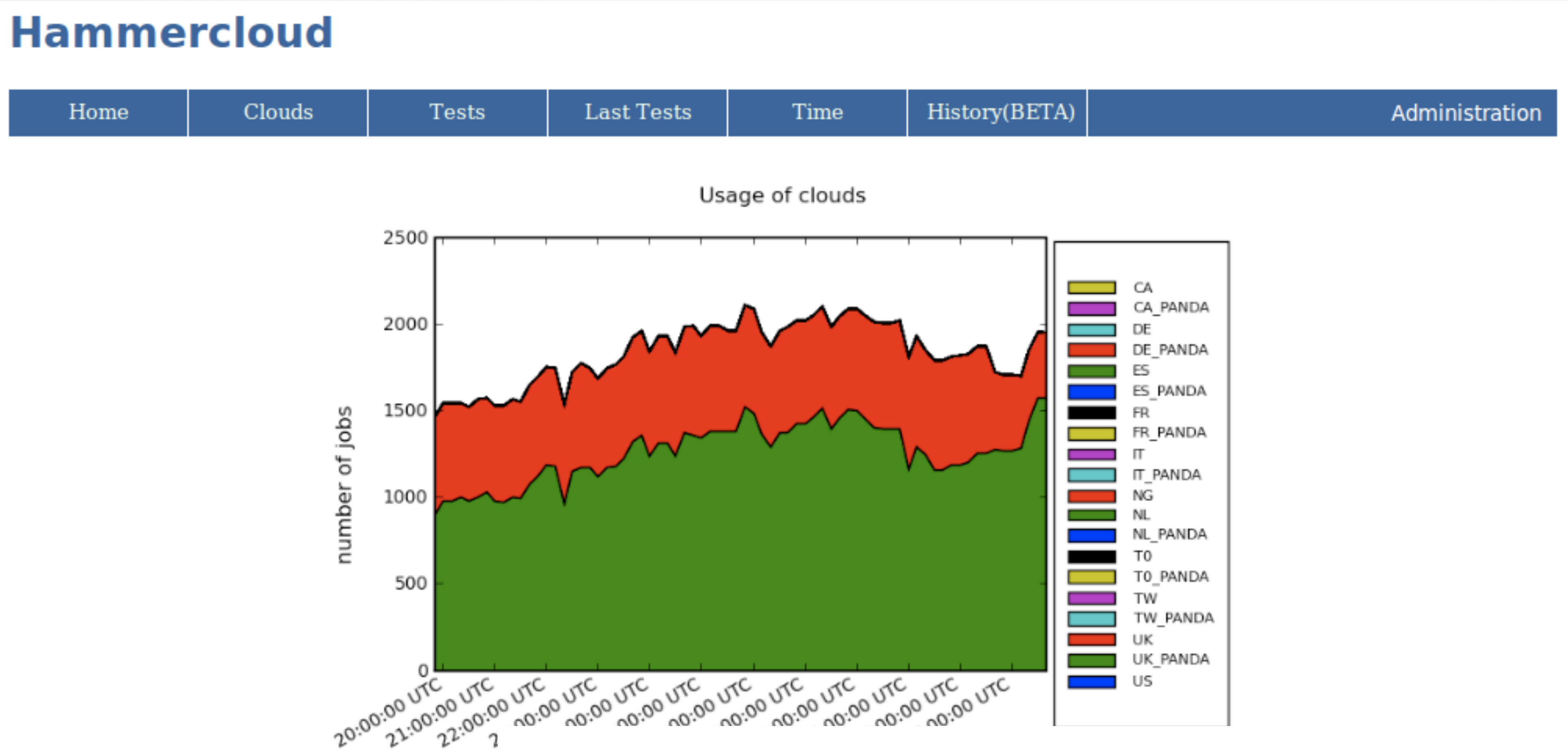
# HAMMERCLOUD TESTS

Up until now, the key variable that HammerCloud is evaluating is the data access method:

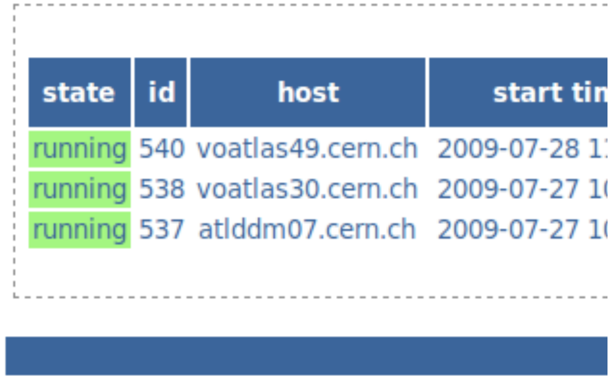
- Posix I/O with local protocol:
  - To tune rfio, dcap, gsidcap, storm, lustre, etc…
  - Testing with read-ahead buffers on or off; large, small or tweaked.
- Copy the files locally before running
  - But disk space is limited, and restarting Athena causes overhead
- Athena FileStager plugin:
  - Uses a background thread to JIT copy the input files from storage
  - Startup – Copy f1 – Process f1 & copy f2 – Process f2 & copy f3 – etc…

# HAMMERCLOUD

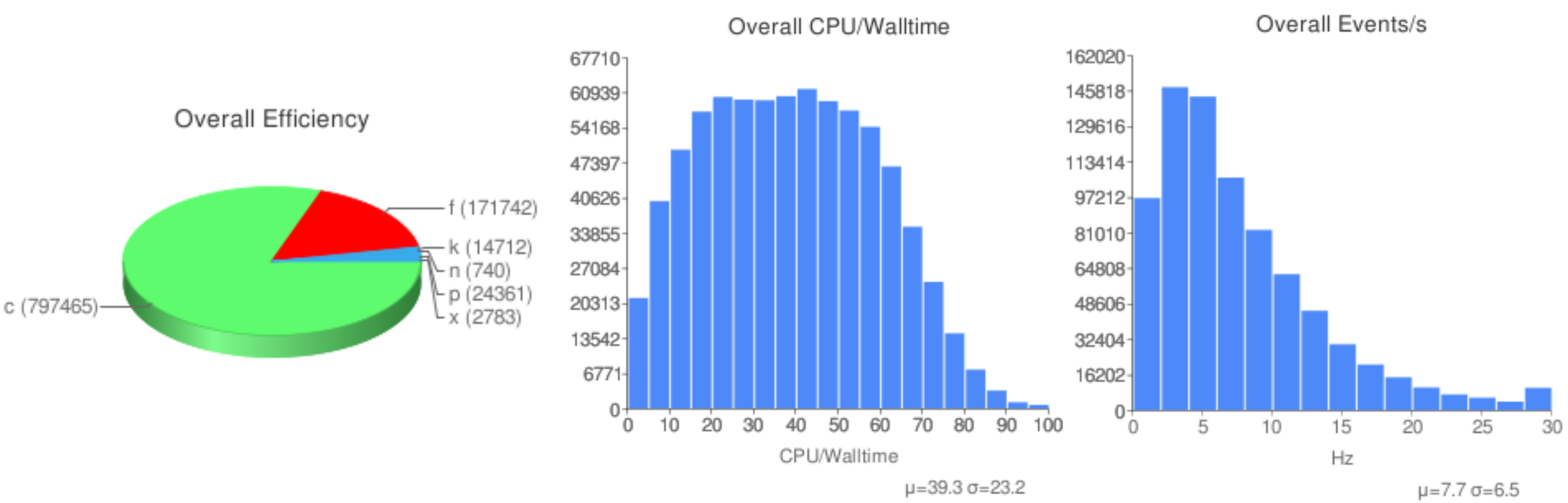
HammerCloud



## Hammercloud



| state | id | host | start tin |
|---|---|---|---|
| running | 540 | voatlas49.cern.ch | 2009-07-28 1: |
| running | 538 | voatlas30.cern.ch | 2009-07-27 1( |
| running | 537 | atlddm07.cern.ch | 2009-07-27 1( |



# HAMMERCLOUD SUMMARY

- Developed for ATLAS (will be adopted by LHCb)
- Automated stress testing of individual sites
- Uses real analysis jobs submitted via Ganga to find site related issues

# SUMMARY

- Testing of user analysis is essential to provide reliable and continuous support for LHC data analysis

- Tests should be reflecting ‘real’ cases without requiring special tools

- The systems for testing user analysis are specific to the different experiments

- Examples presented: GangaRobot and HammerCloud