Reinforcing User Data Analysis with Ganga in the LHC Era: Scalability, Monitoring and User-support

Johannes Elmsheuser

Ludwig-Maximilians-Universität München

on behalf of the GangaDevelopers team and the ATLAS collaboration

19 October 2010/CHEP 2010, Taipei

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Challenges in a LHC Data Analysis

Data volumes

- LHC experiments produce and store several PetaBytes/year
- e.g. ATLAS recorded ∼700M events at $\sqrt{s}$ =7 TeV in 2010 so far

CPUs

- Event complexity (large number of channels) and number of users demands: at least 100000 fast CPUs based on computing model

Software

- The experiments have complex software environment and framework

Connectivity

- Data should be available 24/7 at a high bandwidth

Distributed analysis tools must/should be

- Easy to configure and fast to work with
- Reliable and jobs should have 100% success rate at 1st attempt

Example: ATLAS Analysis in a nutshell

Data

- Centrally organized data distribution by data management system (DQ2) according to computing model

Experiment software (Athena) distribution kits

- Centrally organized installation on EGEE, OSG and NG

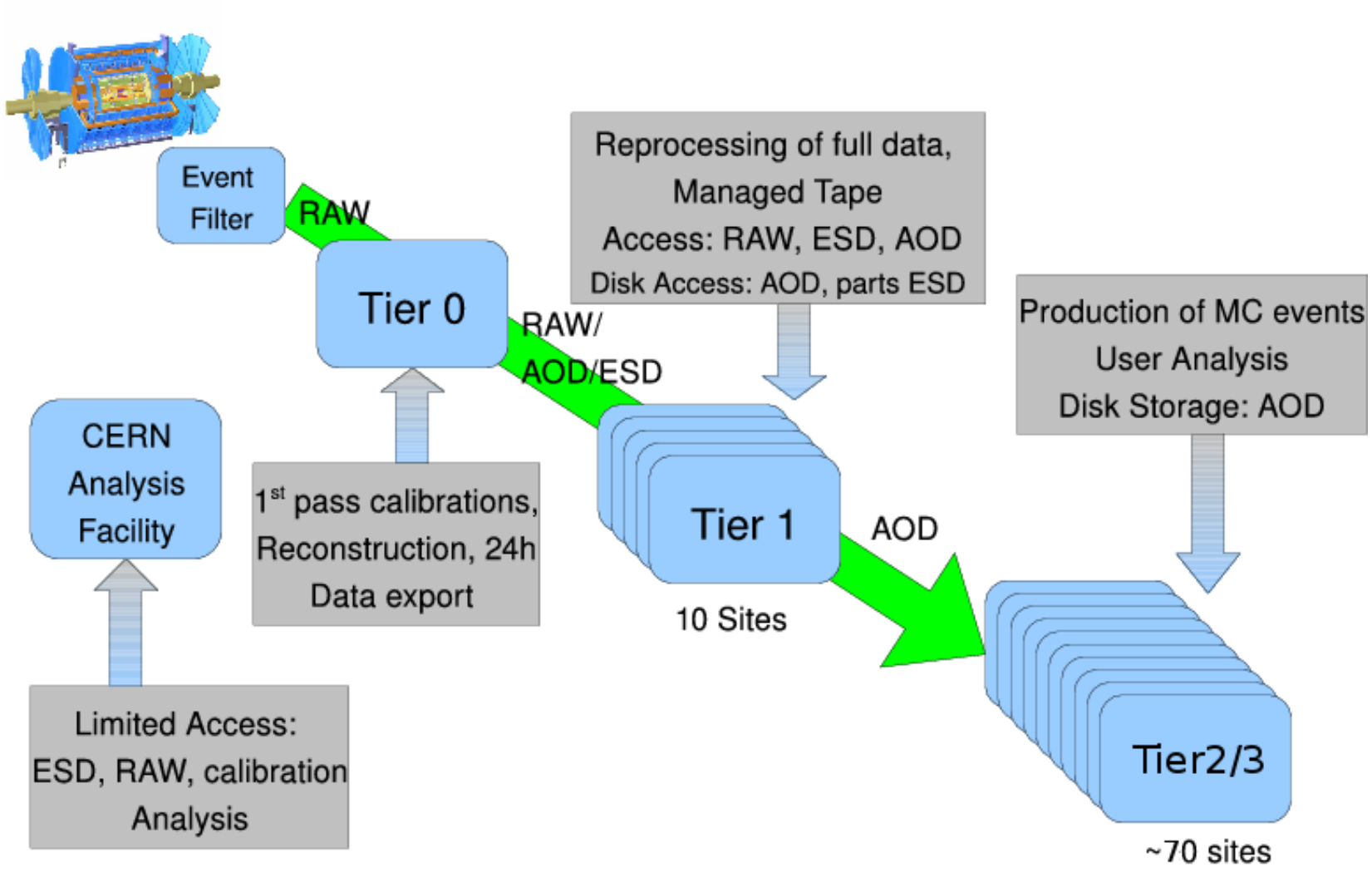
User jobs

- Model: „Job to Data"
- Tools for user job management: Ganga and Panda clients

User Output

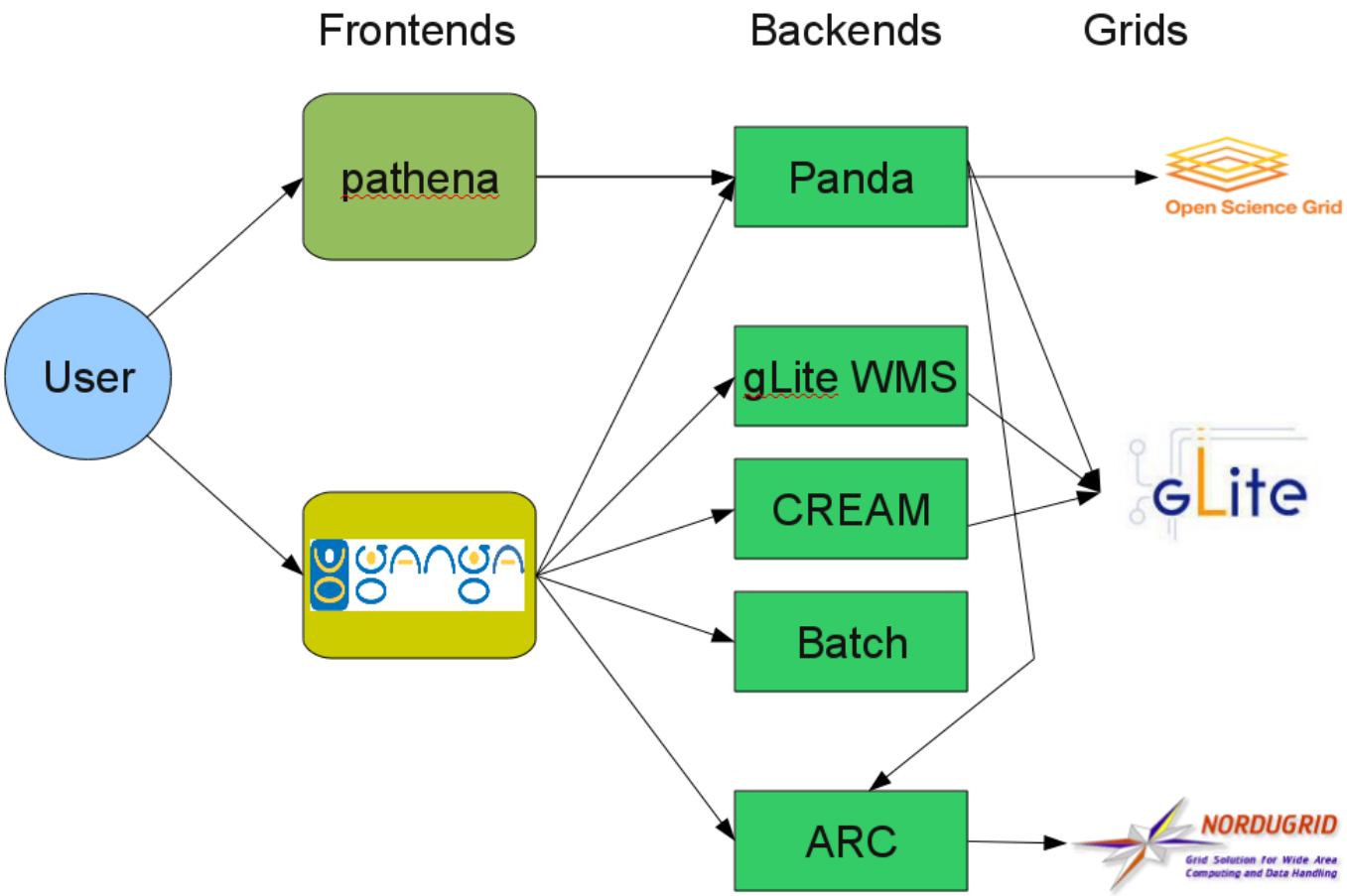
- Store on site scratchdisk or transfer on demand to remote site
- Retrieve output with DQ2 command line tools to local computer

Example: ATLAS Data Distribution



≈ 80 Tier1/2/3 sites managed by DQ2 right now

Example: ATLAS Distributed Analysis Layers



Data is centrally being distributed by DQ2 - Jobs go to data

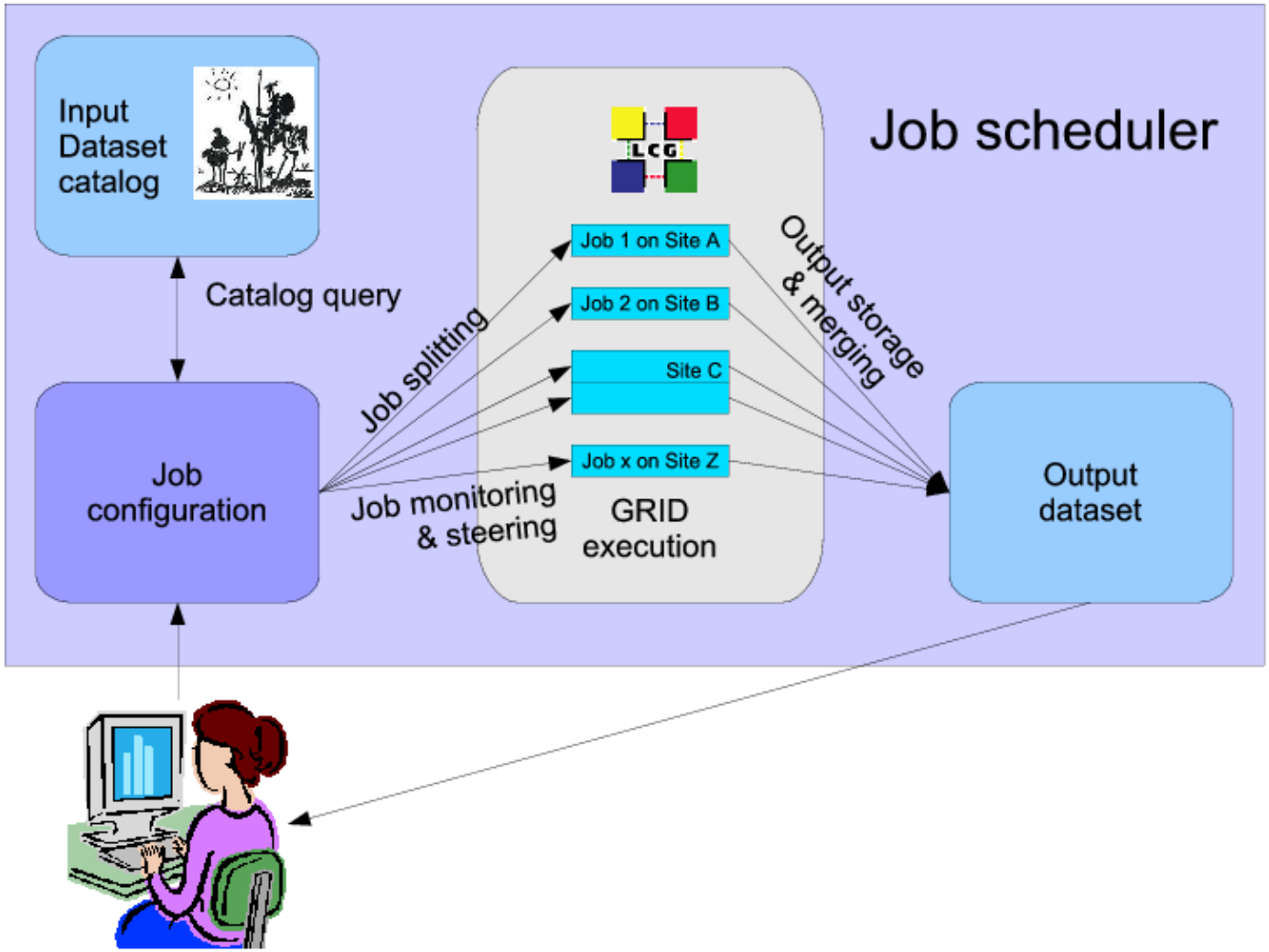
Front-end Client: GANGA

http://cern.ch/ganga

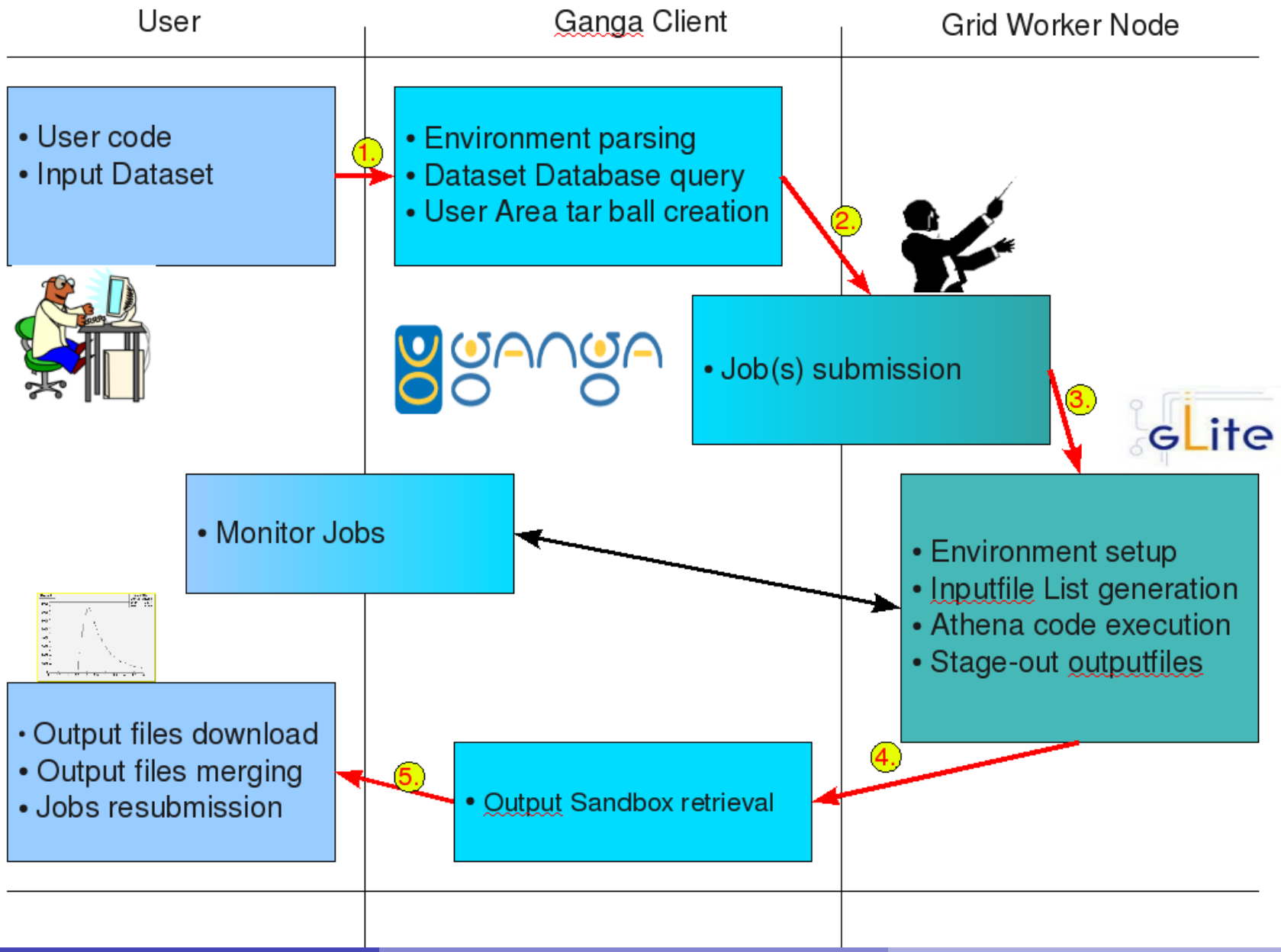
- A user-friendly job definition and management tool.
- Allows simple switching between testing on a local batch system and large-scale data processing on distributed resources (Grid)
- Developed in the context of ATLAS and LHCb similar to open source community-driven project
- Core development is joint between LHCb and ATLAS
- Component architecture readily allows extension
- Mature and stable, with an organized development process - release management shifts and extensive release testing
- Python framework and GPL License

Distributed Analysis: Ganga

How to manage the workflows: Job scheduler/manager: GANGA



Example job workflow



Recently added features I

New job repository

- Job repository is core Ganga component for job bookkeeping and monitoring
- New XML based architecture - more reliable and faster (LHCb start-up time: 1-2s)
- Easily holds several 10k jobs and very fast start-up through lazy loading

Error reporting

- Upload environment to server with single command for user support

Task Monitoring Dashboard

- New ATLAS application monitoring plug-ins
- Job statistics displayed central dashboard webpage

CREAM backend

- Direct submission to CREAM CE

WebGUI monitoring

- New WebGUI based monitoring

Recently added features II

GangaAtlas:

- GangaTasks: Automatic job configuration, steering, throttling, resubmission
- Plugin for ATLAS Metadata Interface (AMI) for event based job splitting and luminosity information
- Plugin to Event-Level Database (ELSSI) for event picking
- Many new features for Panda backend
  - Different ROOT based workflows
  - Cross site job brokering
- Job performance and statistics collection for HammerCloud

GangaLHCb:

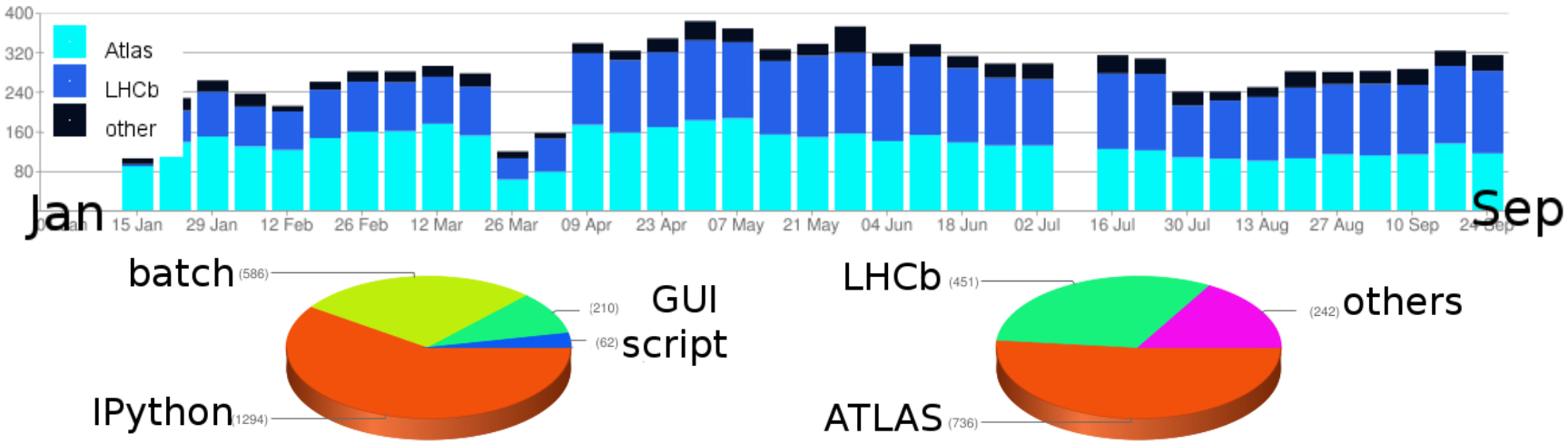
- See talk from M. Williams in this session

Further interesting plugins:

- GangaSAGA: Plugin for SAGA-API
- GangaJEM: Job level monitoring plugin

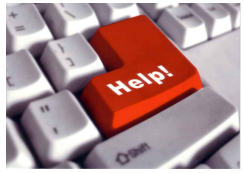
Number of Ganga Users

Unique users by week and experiment (2010):



- Total number sessions: 370413, Number unique users: 1319, Number of sites: 130
- Since start of 7 TeV collisions large increase of distributed analysis jobs

Current user problems and Support



User support is very important but time consuming

Central ticketing system for site or grid middleware probleme: GGUS

- Site or experiment experts try to solve problems
- Often ,,one-way" communication

Support mailing list for analysis tools

- Central discussion board for ,,all" problems
- Discussion of several people
- E.g. in ATLAS and LHCb:
  - Before: only developers as experts - very time consuming
  - Now: experiment shift teams with shift credits
  - Very busy mailing list
  - Hope: user-to-user support similar to open-source projects
- Sites are more stable but still day to day glitches

Infrastructure Tests - Analysis stress tests

ATLAS is/has been testing sites with very high automatic generated analysis load and functional tests: HammerCloud `http://hammercloud.cern.ch/`

(See presentation by D. van der Ster and Poster by F. Legger)

Now also available for CMS and soon for LHCb

Ingredients and Highlights:

- Ganga is central ,,engine" for job configuration, submission, performance collection
- User Analysis puts much higher load on SE compared to CPU dominated simulation
- Analysis tools generally stable and reliable
- Some weak spots detected in site infrastructures, especially in input file access mode lots of tuning potential
- From functional tests: usually only a handful of sites have temporary problems

Next steps

- Consolidate current workflows
- Add more MSG based monitoring to more applications and backends
- Extend WebGUI features
- Make GangaTask plugin accessible for all Ganga applications and backend
- More release testing and nightly builds
- Outreach: Development blog

Prospects and Evolutions

- Infrastructure demonstrated to be able to support LHC data processing and analysis
- A reliable and robust service of many components neccessary
- Significant operational infrastructure behind it
- Adapt to future technologies:
  - Improve data storage and data access
- Network is much better than initially anticipated
  - Rethink data access models
- Experiments have truly distributed models

http://cern.ch/ganga