Evolution of User Analysis on the Grid in ATLAS

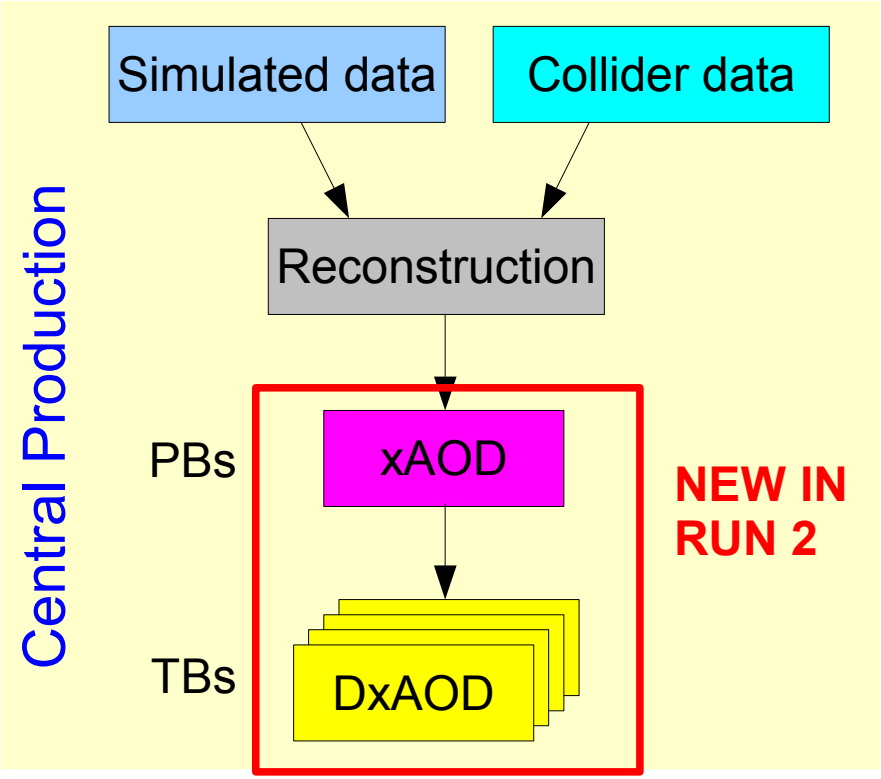
A. Dewhurst (RAL), F. Legger (LMU)

on behalf of the ATLAS collaboration

October 12th, 2016
22nd International Conference on Computing in High Energy and Nuclear Physics (CHEP), San Francisco

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

ATLAS computing model – LHC Run 2

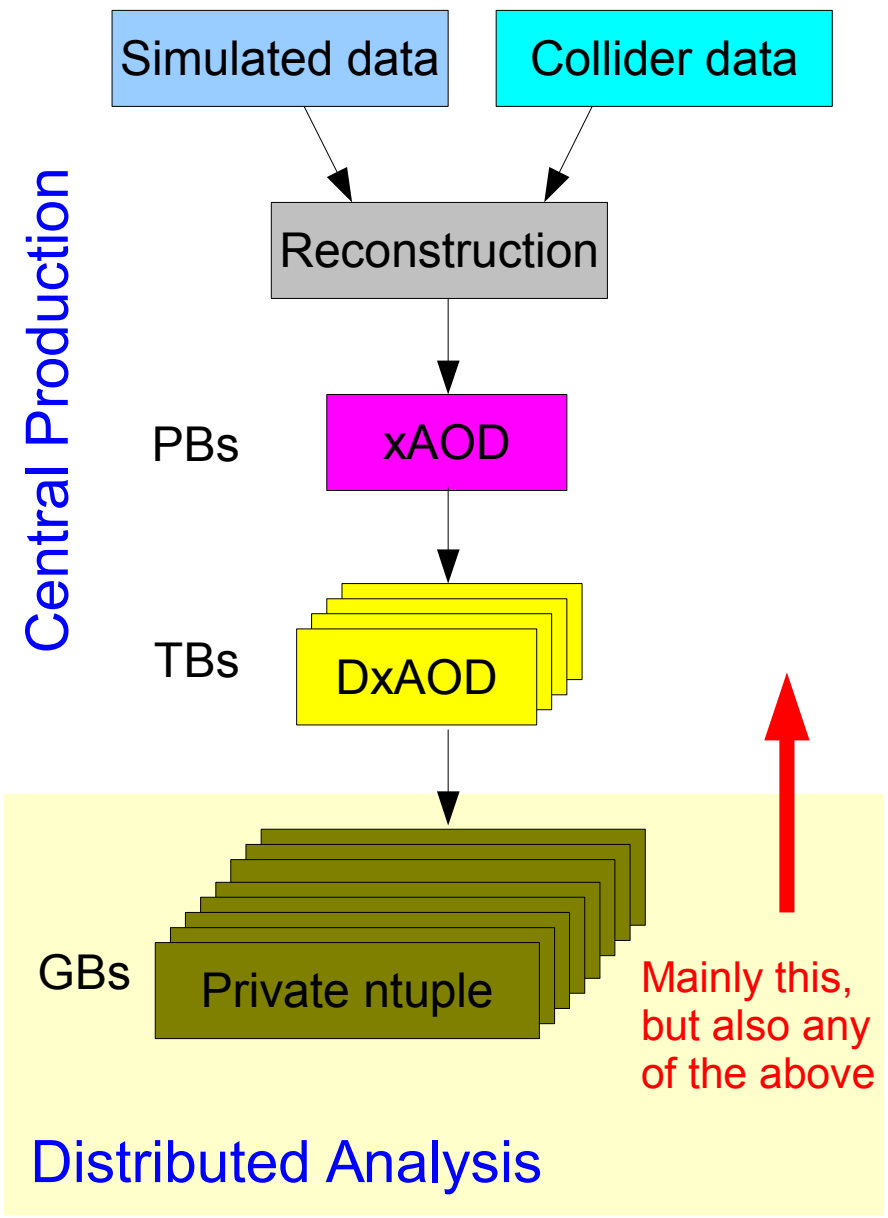


- New analysis data format: xAOD
  - Created on the basis of Run 1 experience with the AOD (Analysis Object Data)
  - Readable in both ROOT and Athena (ATLAS Software framework)
- Central production of Derived xAOD (DxAOD) for physics and performance groups
  - almost 100 formats
  - **Data**: *train model*, i.e. tasks with a single run as input are creating several output formats *(carriages)*
  - **Monte Carlo**: on request

See also *ATLAS Distributed Computing experience and performance during the LHC Run-2* [1]

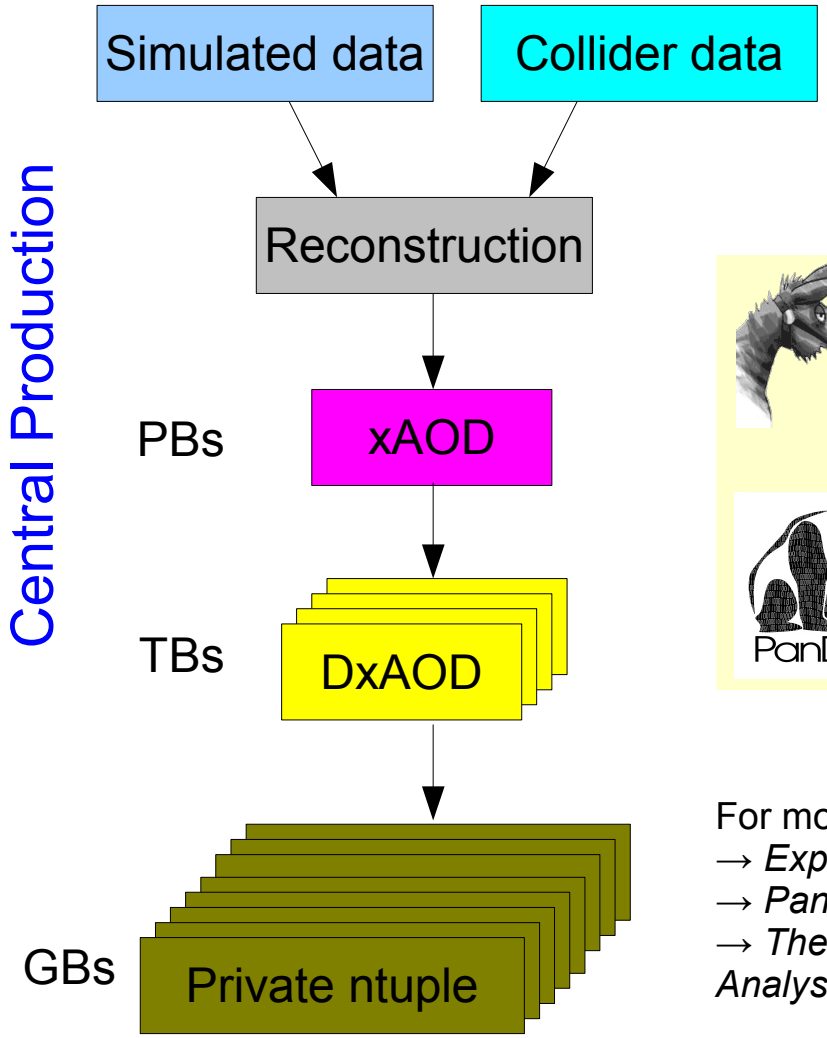
[1] *https://indico.cern.ch/event/505613/contributions/2230713/*

ATLAS computing model – LHC Run 2



- New analysis data format: xAOD
  - Created on the basis of Run 1 experience with the AOD (Analysis Object Data)
  - Readable in both ROOT and Athena (ATLAS Software framework)
- Central production of Derived xAOD (DxAOD) for physics and performance groups
  - almost 100 formats
  - **Data**: *train model*, i.e. tasks with a single run as input are creating several output formats *(carriages)*
  - **Monte Carlo**: on request
- ATLAS users can run ***any*** step on the grid
  - However privately produced data/MC cannot be used in published results
  - Still useful for development, testing, validation of new workflows

ATLAS computing model – LHC Run 2



Thanks to our *working animals*

- Both completely redesigned for Run 2

- Distributed Data Management (DDM): **Rucio**

- Job Workload Management: **PanDA**
  - Database Engine for Tasks (**DEFT**)
  - Job Execution and Definition Interface (**JEDI**)

For more details, see
→ *Experiences with the new ATLAS Distributed Data Management System* [1]
→ *PanDA for ATLAS distributed computing in the next decade* [2],
→ *The ATLAS Production System Evolution. New Data Processing and Analysis Paradigm for the LHC Run 2 and High-Luminosity* [3]

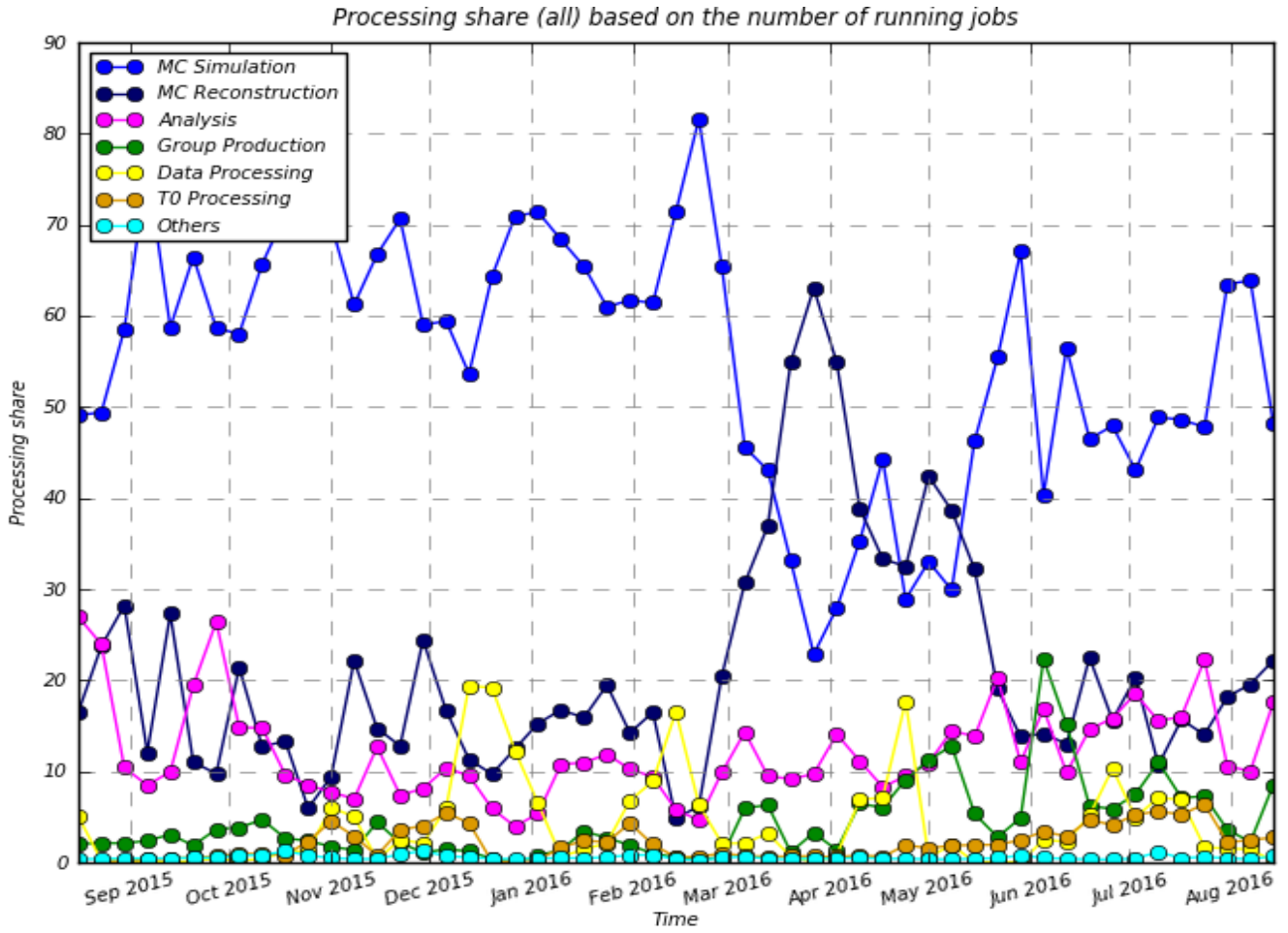
Distributed Analysis

[2] *https://indico.cern.ch/event/505613/contributions/2230910/*
[3] *https://indico.cern.ch/event/505613/contributions/2230704/*
[4] *https://indico.cern.ch/event/505613/contributions/2230440/*

Distributed computing activities

Plot from *http://dashb-atlas-job.cern.ch/dashboard/request.py/dailysummary*

- Analysis share ranges from **10% to 20%** of running jobs
  - Driven by interplay of user and central production activities
- Stable load, except for big MC production or data reprocessing campaigns



Interlude

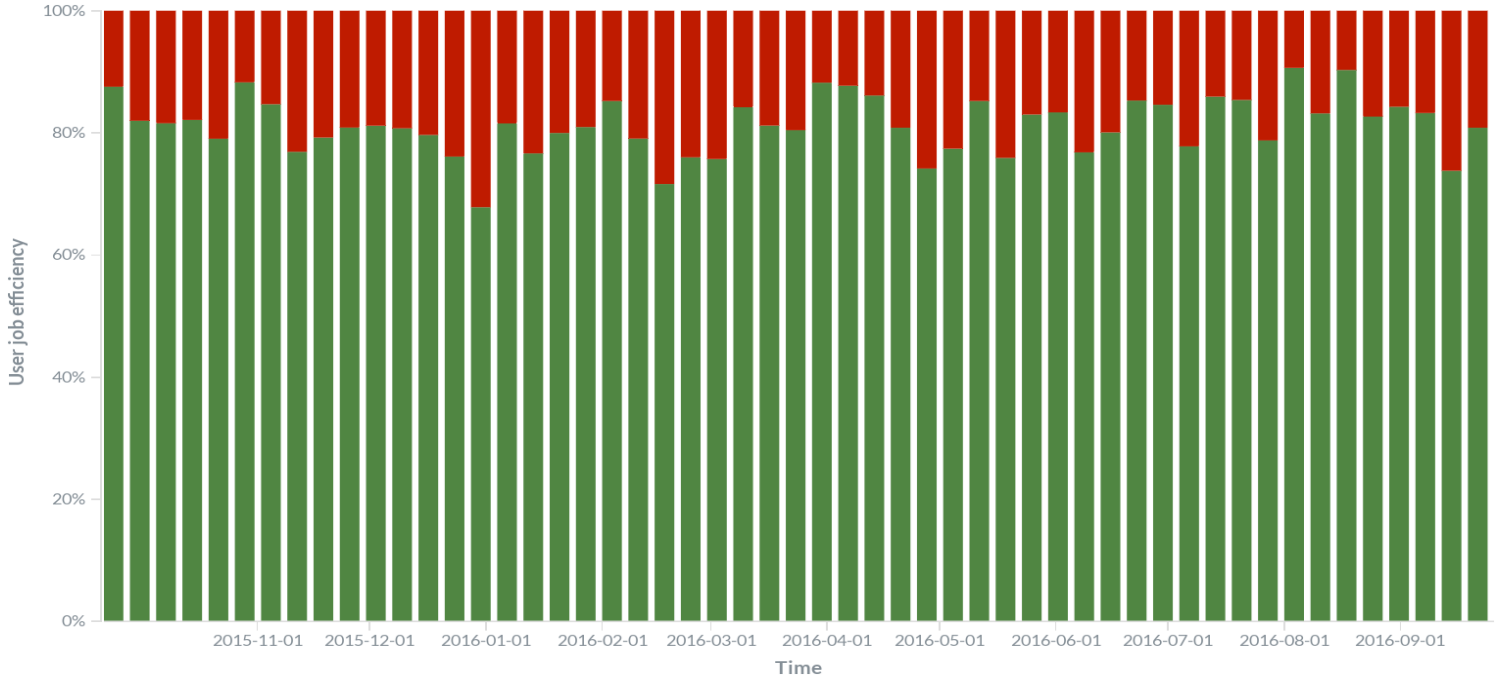
- The *ATLAS analytics* infrastructure based on **Kibana/ElasticSearch** has been used for the plots/numbers presented in the rest of the talk

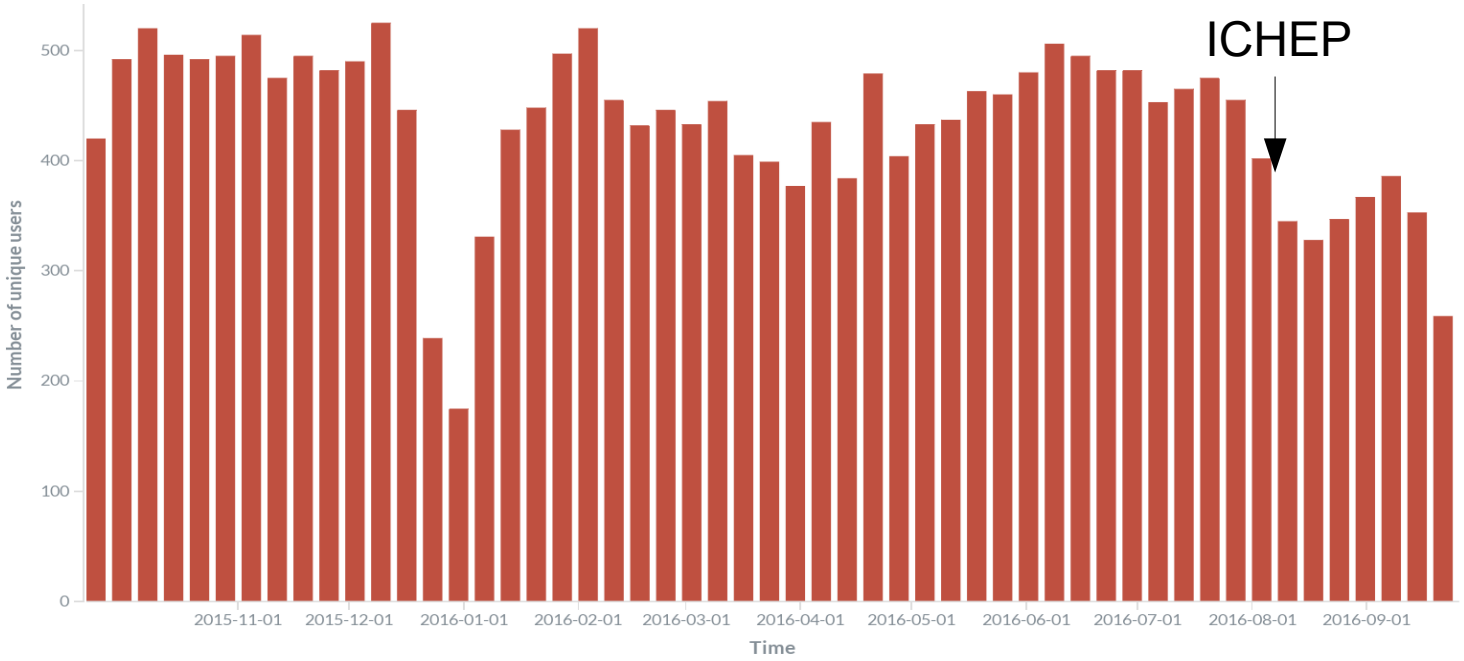
  - Analytics cluster at MWT2 (OpenCloud)
    - See *Big Data Analytics Tools as Applied to ATLAS Event Data* [5] for more details

  - Combining data from two sources:
    - **PanDA** job archive
      - more than 100 fields available for each job: *site, host, user, priority, input and output data, status, errors, memory, …*
      - Available for all jobs from July 2014 to 30 minutes ago
    - **Rucio** storage element dumps
      - information of all datasets managed by Rucio, including *storage element, size, data type, creation date, date of last access, owner…*
      - Daily updated, available since Fall 2015

[5] *https://indico.cern.ch/event/563611/*

User jobs





Flat efficiency*: >80%

** Efficiency is defined as the ratio between successful and total (successful+failed) jobs*

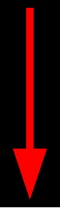
Up to 500 unique users per week

User job metrics

Plots/numbers from *http://atlas-kibana.mwt2.org:5601/app/kibana#/dashboard/FL-Analysis*

June-August 2016

```
> Number of unique users: ~1000
> Average wall time/job: ~1 hour
> Average queue time/job:~2 hours
> Average memory/job: 600 MB
> Average number of jobs/user: 10k/month
> Average number of events/user: 150M/month
> Average input/job: 15 GB / 4 files / 15k events
> Average output/job: 100 MB
```

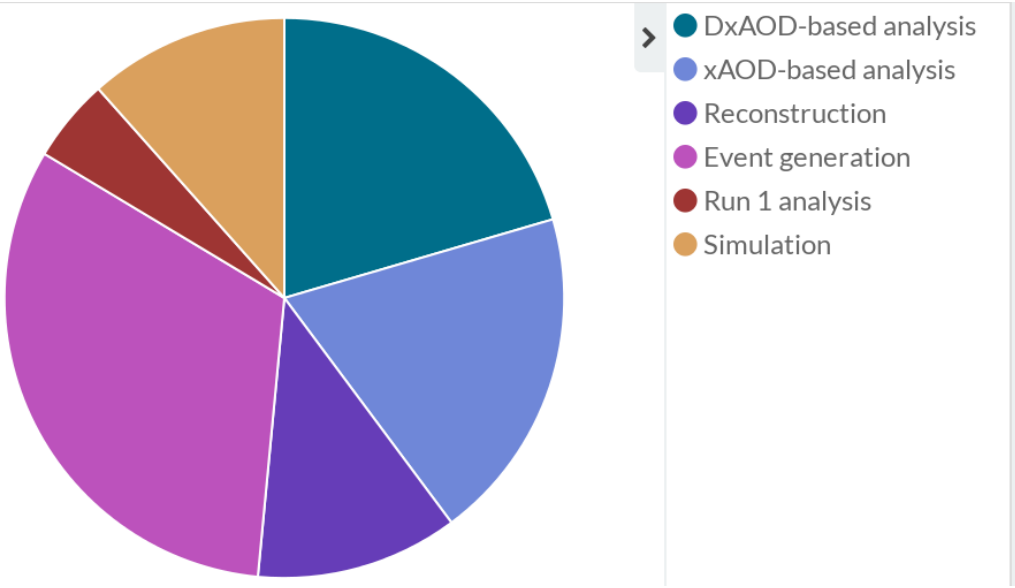


Data reduction of a factor of 150!

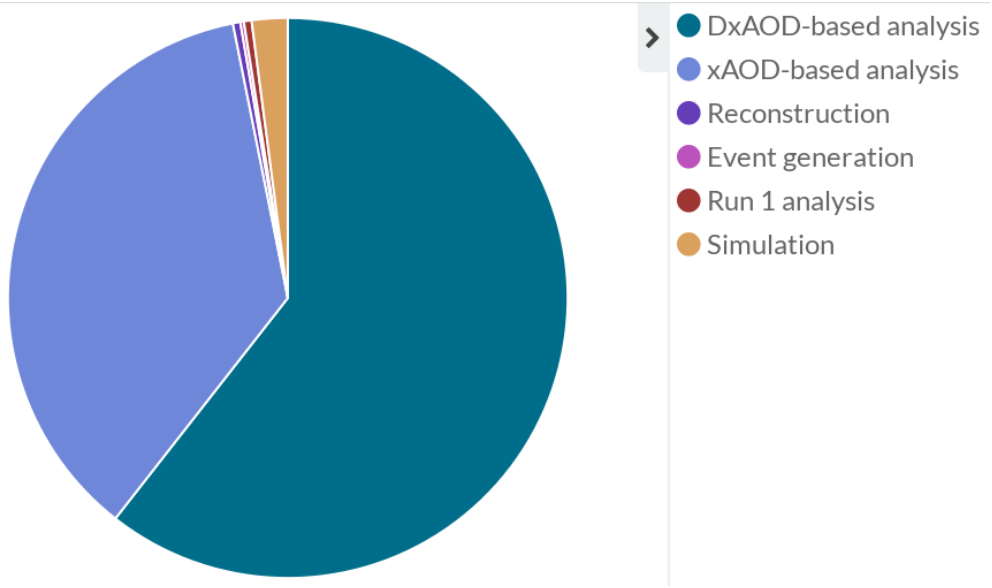
Typical user workflows

Plots from *http://atlas-kibana.mwt2.org:5601/app/kibana#/dashboard/FL-Analysis-Inputs-(for-CHEP)*

June-August 2016



Fraction of total walltime

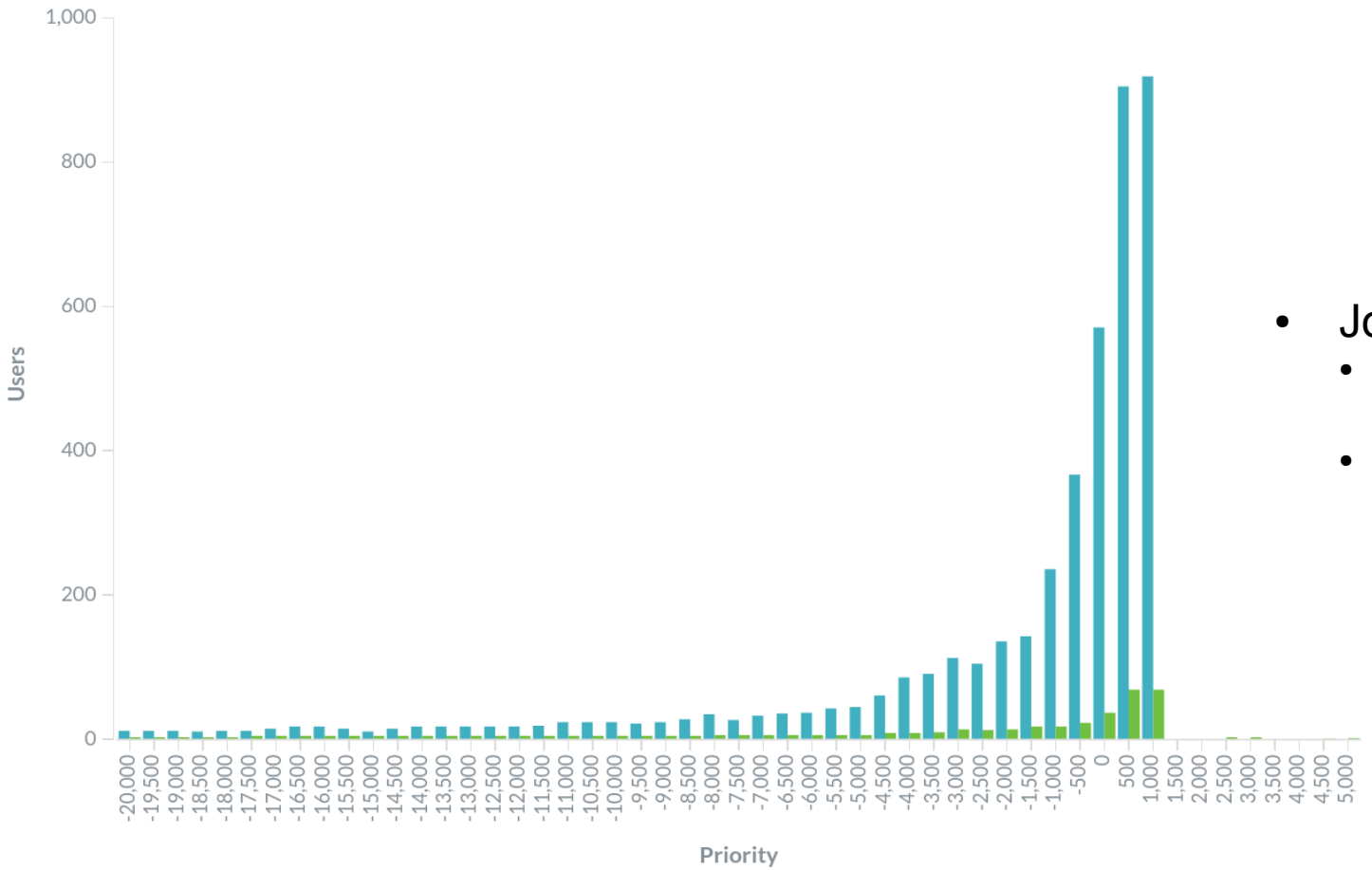


Fraction of processed events

- DxAOD-based analysis most popular workflow
- Production-like workflows (event generation, simulation, reconstruction) more expensive in terms of resources

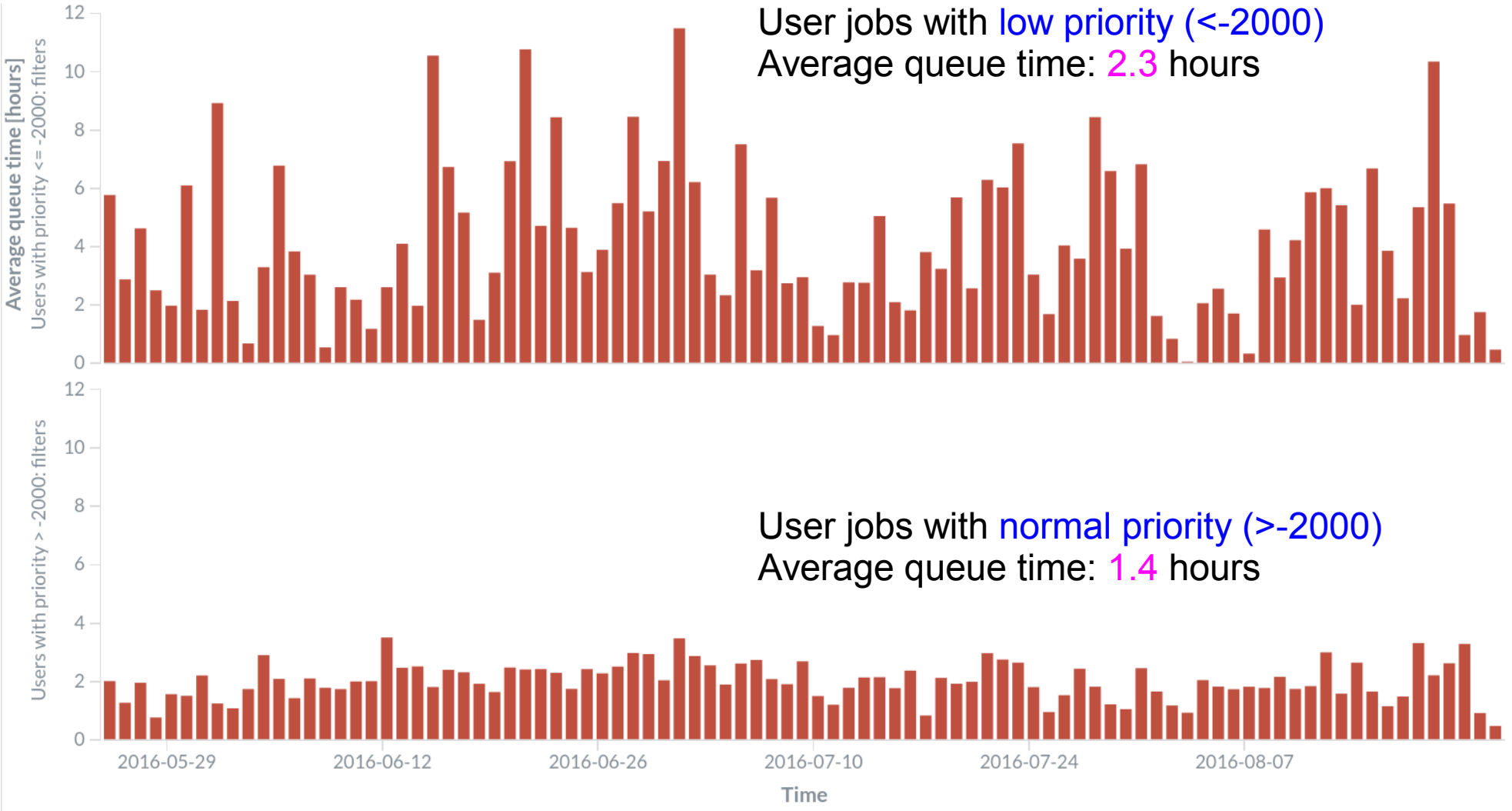
User job priority

- **Priority** system among user jobs
  - User priority starts at 1000 and fairly rapidly decreases
  - *Authorized* users may have priority > 1000 for a limited period of time (urgent production)
  - Jobs submitted with working group privileges do not lower the personal user priority, but still compete with all other user jobs
  - Most users have priority > –2000



- Job concurrency:
  - Number of cores per user limited to 10k
  - *Separate fair share* at sites from production jobs

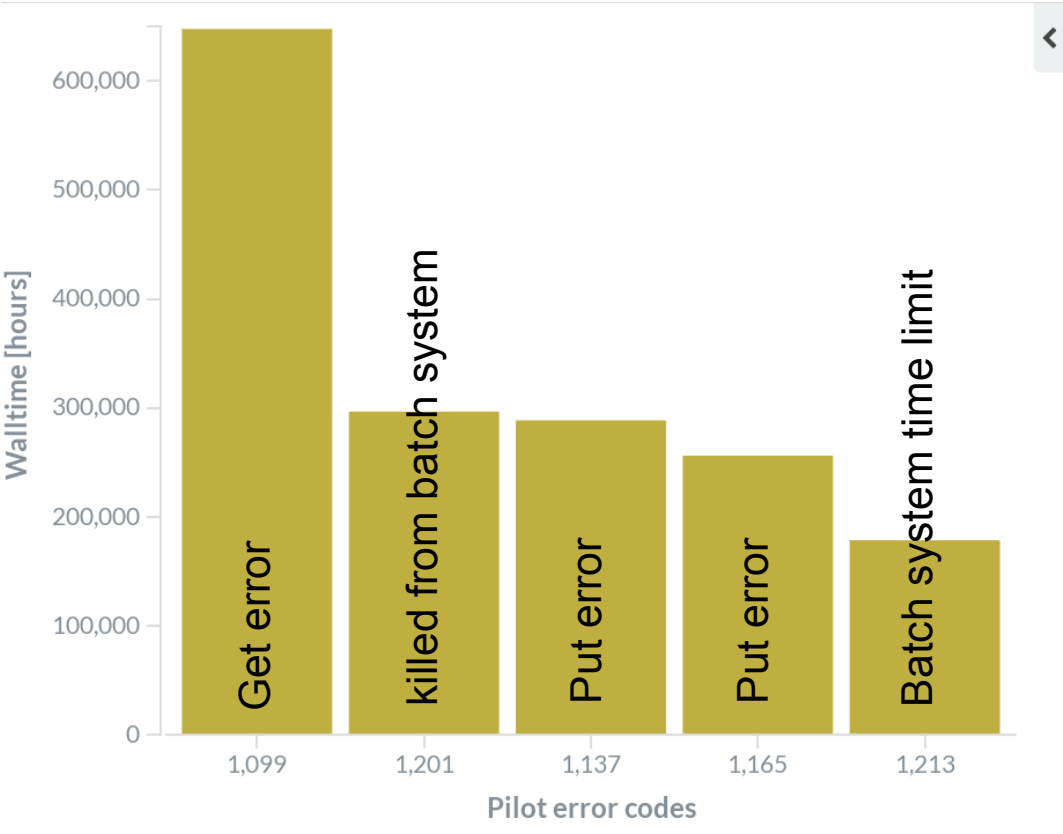
Waiting time



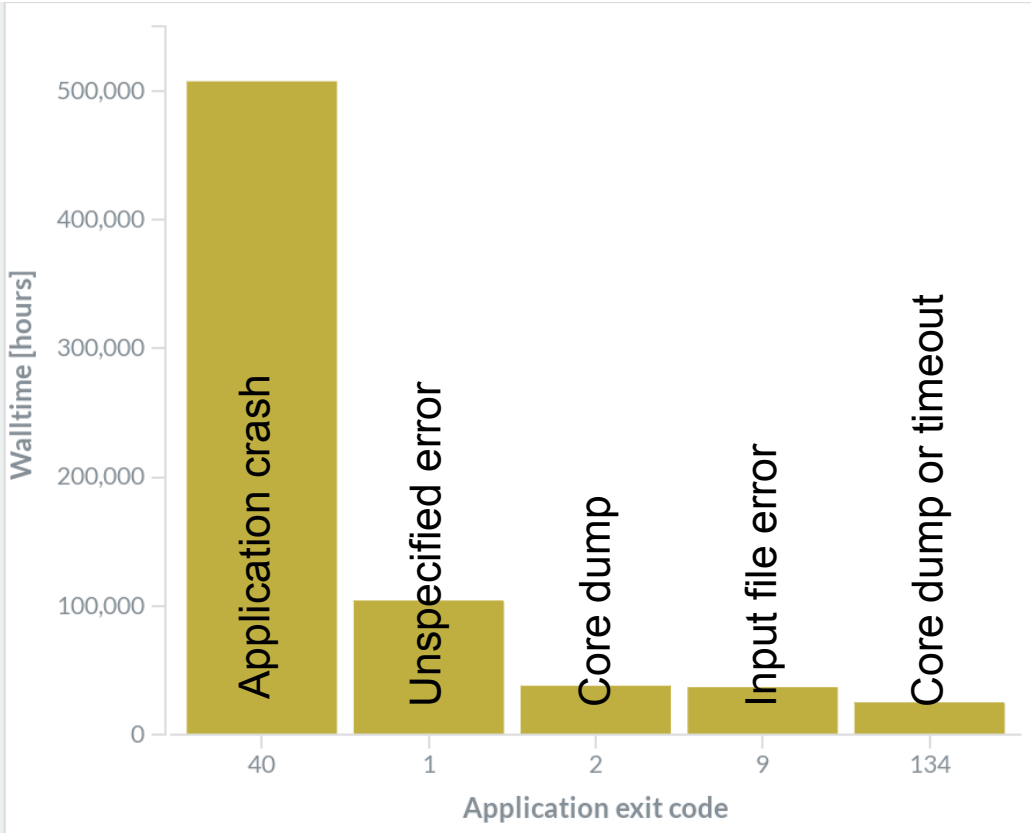
Low priority jobs will run quickly if:
→ grid is empty
→ input datasets at empty sites or event generation (can run anywhere)

Failed user jobs

Top 5 **pilot** exit codes



Top 5 **application** exit codes



- *Walltime* consumption of **failed** jobs:
  - ~50% application errors
  - ~50% infrastructure: storage errors and batch system kills

Future developments

- *Global* ***shares*** for all distributed computing activities
  - Optimize resource usage
  - Eases central operations
- Differentiate ***analysis queues*** to better cope with different workflows
  - *Standard* queues with uniform limits on job duration, memory usage, etc
  - *Special* queues with specific requirements: high memory, multi-core, long jobs
- Dynamic **data placement**
  - Distribute popular formats to offload busy sites

For more details, see also:

→ *How to keep the Grid full and working with ATLAS production and physics jobs* [6]

→ *C3PO - A Dynamic Data Placement Agent for ATLAS Distributed Data Management* [7]

[6] *http://indico.cern.ch/event/505613/contributions/2230712/*
[7] *http://indico.cern.ch/event/505613/contributions/2230956/*

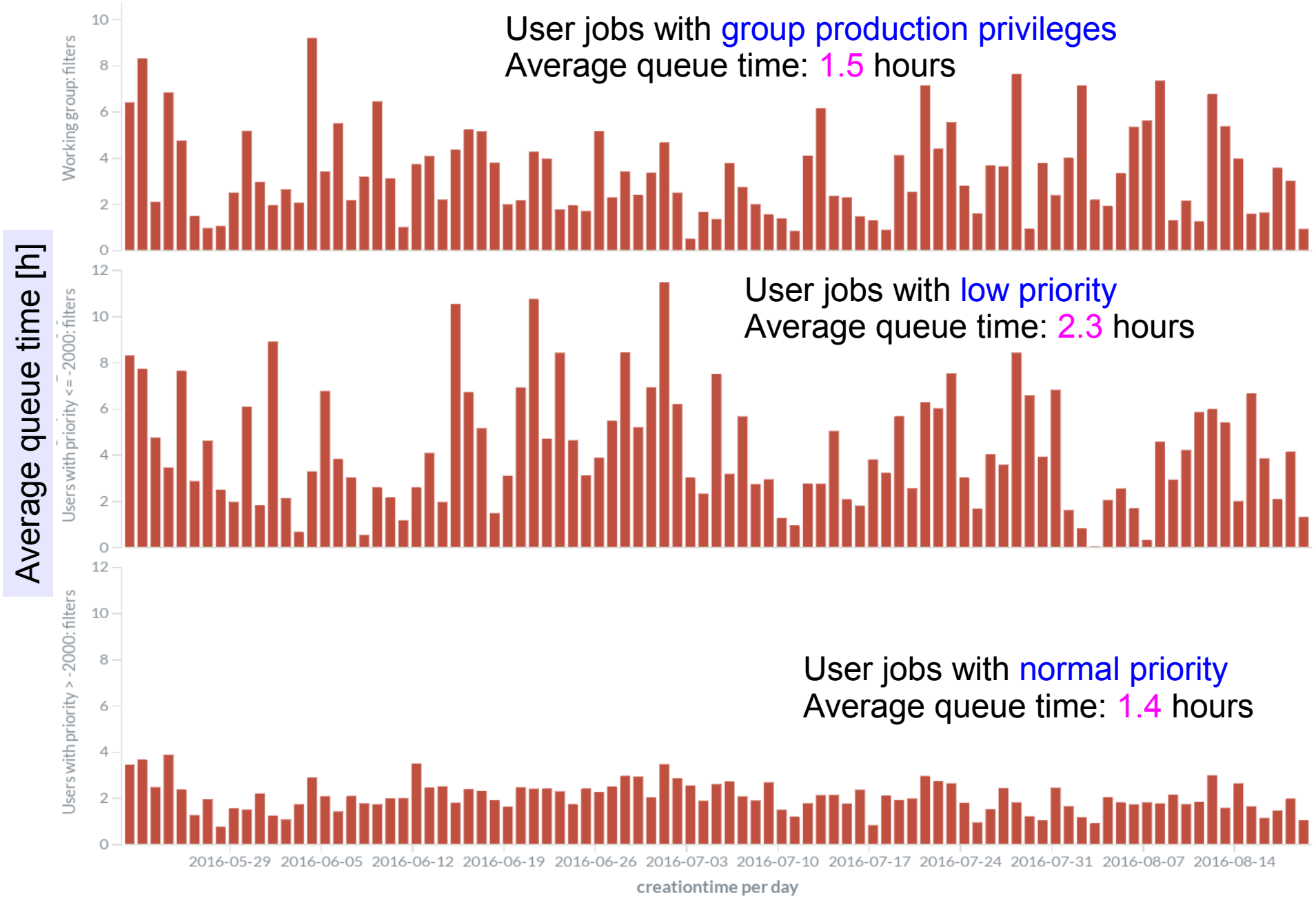
Conclusions

- Several changes to **ATLAS** computing for **LHC Run 2**:
  - New *production* and *data management* systems
  - New *data format* for analysis

- Move was a SUCCESS!
  - Distributed analysis performances stable, 80% job efficiency
  - User workflows are heterogeneous:
    - Creation of final ntuples before publication-ready results
      - New **DxAOD** most popular analysis format
    - Production-like workflows also popular, mainly for testing/validation:
      - event generation, simulation, reconstruction
  - Load balancing achieved through priority system
    - Most user jobs waiting time ~1 hour

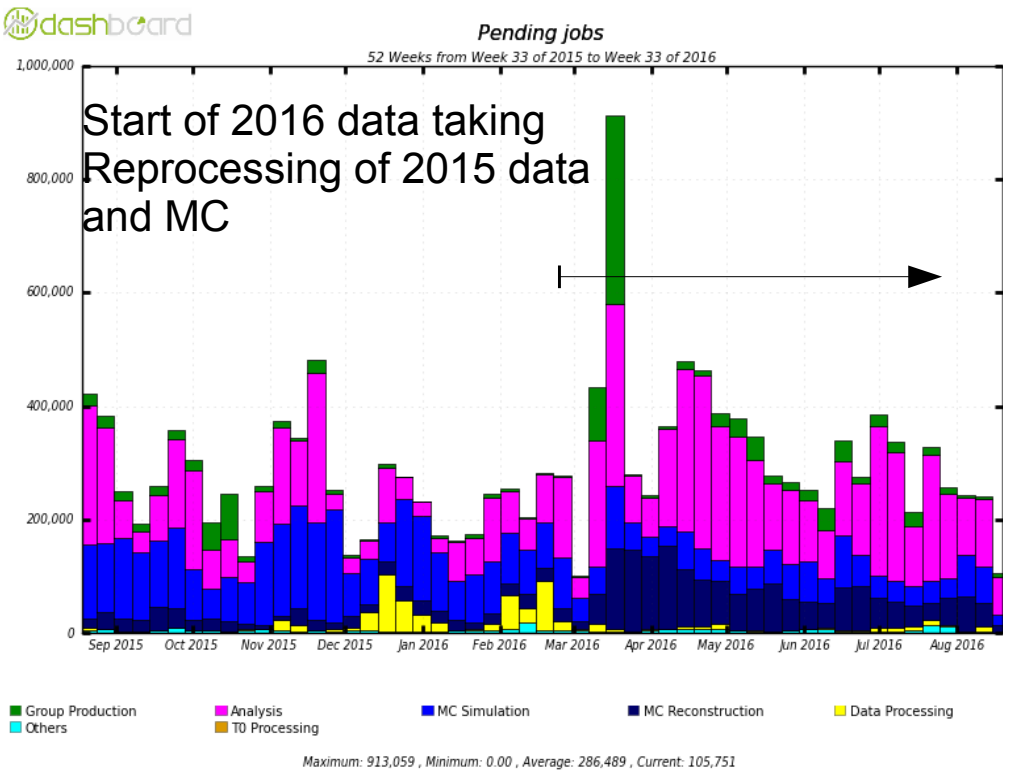
Backup

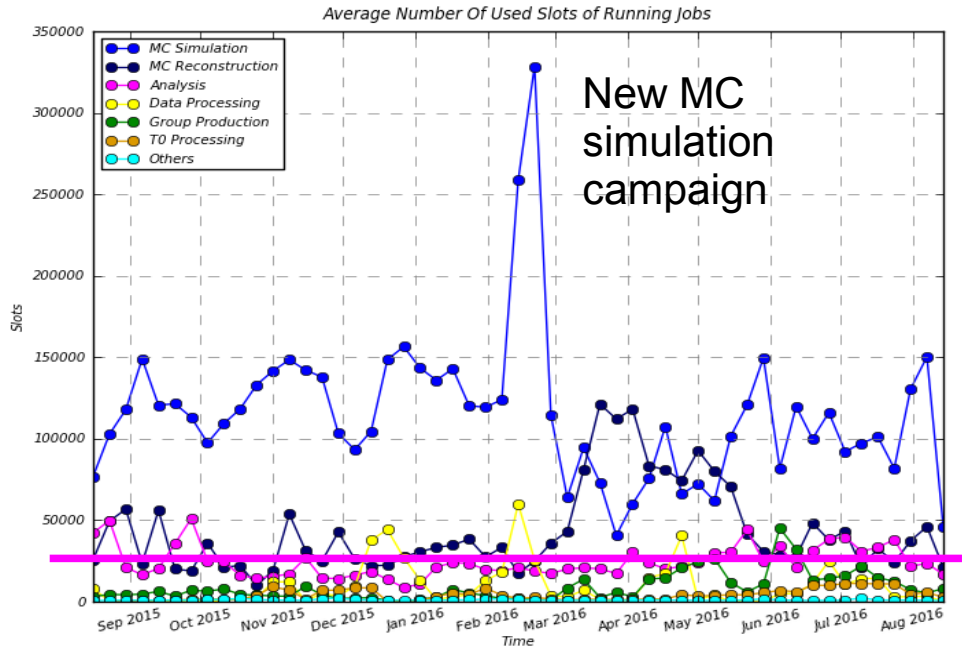
Waiting time



Distributed computing activities

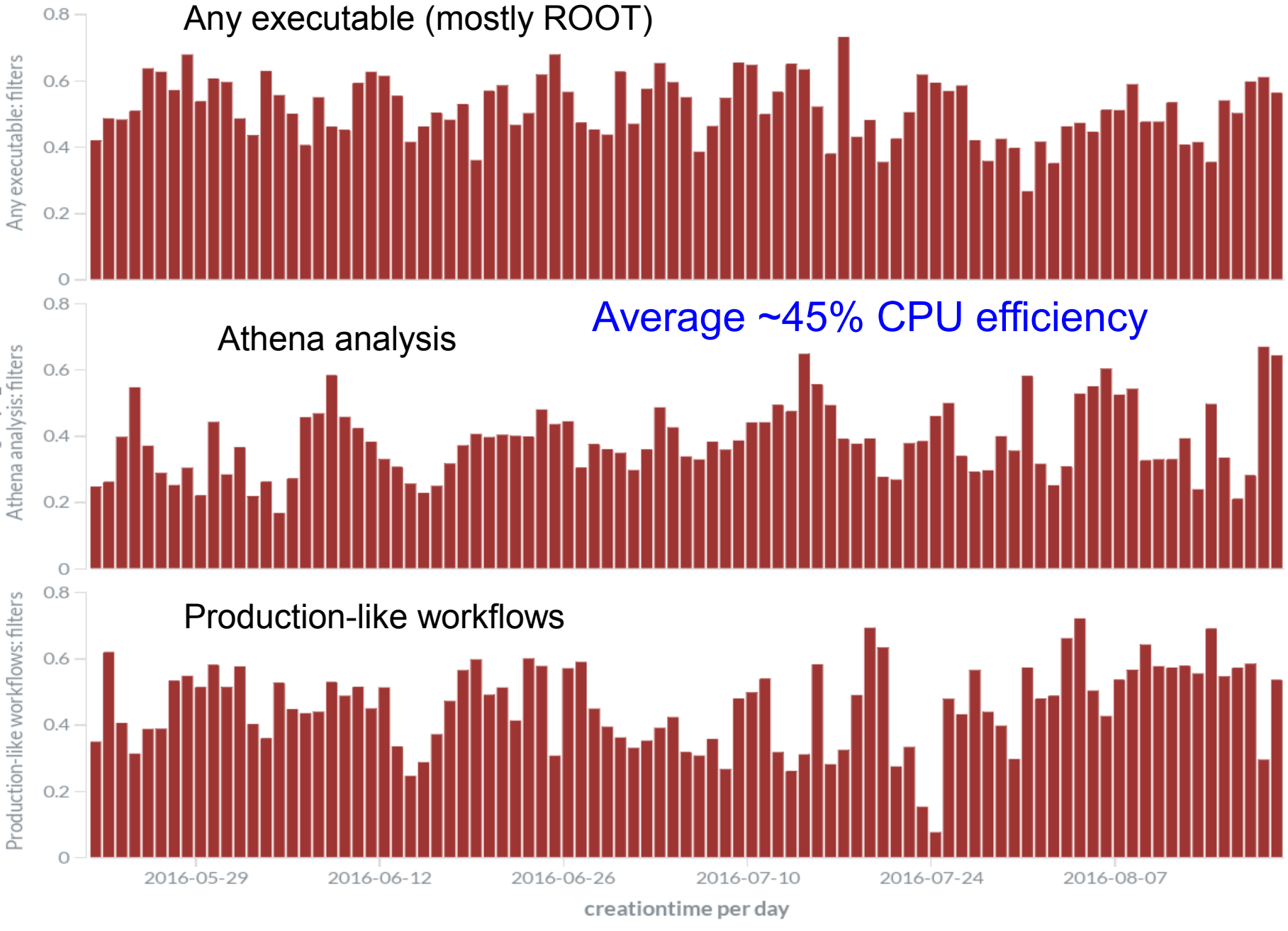
Plots from *http://dashb-atlas-job.cern.ch/dashboard/request.py/dailysummary*





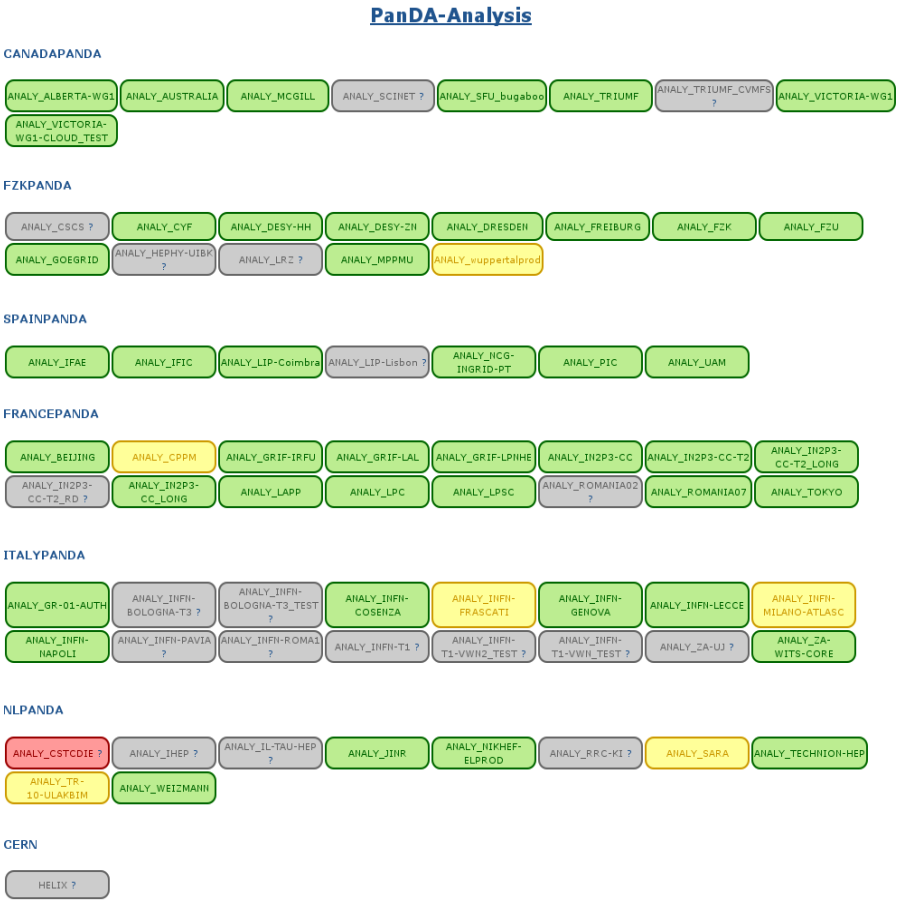
CPU efficiency

Plots from *http://atlas-kibana.mwt2.org:5601/app/kibana#/dashboard/FL-Analysis*



HammerCloud

- Tool for automatic site testing
  - both functional and stress tests
  - Used by ATLAS, CMS, LHCb
  - Crucial tool to test **new deployments** (JEDI, Rucio) before going to production/exposing changes to users
  - Used also for **R&D** of new workflows
  - Fully integrated in ATLAS Grid Information System (AGIS)



- Suite of 3 AFTs, Analysis Functional Tests, mimicking typical user analysis are used for **automatic exclusion** of sites failing the tests from brokerage
  - Typical efficiency of analysis functional tests: 95% → constant over time, clouds, ...

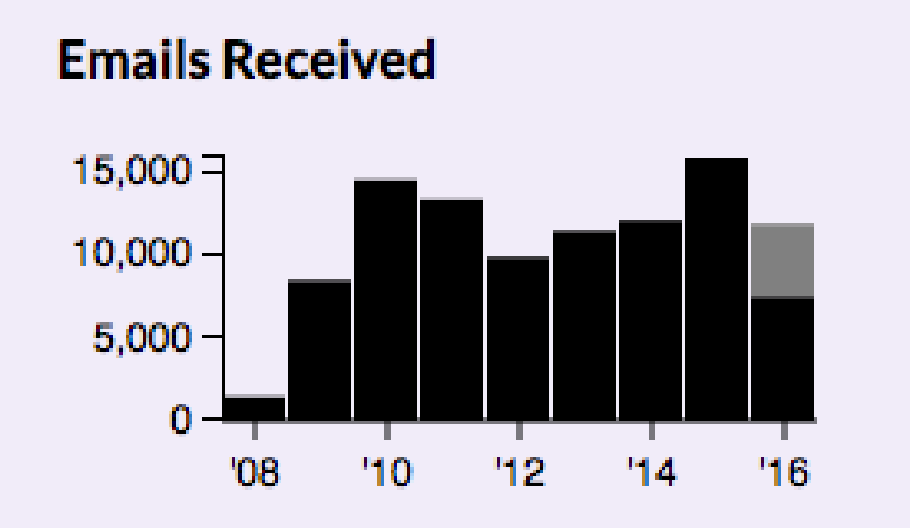
http://hammercloud.cern.ch/hc/app/atlas/

Distributed Analysis Support Team DAST

- User support with dedicated mailing list
  - expert shifters covering 16 hours/day (American and European time-zones)
  - Critical to help users to solve grid issues fast

- Established in October 2008, more than 1000 users

- Covered by DAST:
  - Rucio and Jedi clients
  - Site services/issues
  - Physics analysis tools
  - Monitoring systems



User job priority in PanDA

Job priority

Job priorities are calculated for each user by using the following formula. When a user submits a job which is composed of M subJobs,

$$Priority(n) = 1000 - \frac{T+n}{5} - W*(U-Q)*H(x=(U-Q))$$

where

- Priority(n) … Priority for n-th subJob (0≤n<M)
- T … The total number of the user's subJobs existing in the whole queue. (existing = job status is one of defined,activated,running)
- W … Weight = 100, can be changed on a per-user basis. Needs approval
- U … CPU usage for last 24 hours (in kSI2kday)
- Q … CPU quota = 500, can be changed on a per-user basis. Needs approval
- H(x) … Heaviside step function (0: x≤0, 1: x>0)

T is set to 0 for jobs submitted with group production privileges