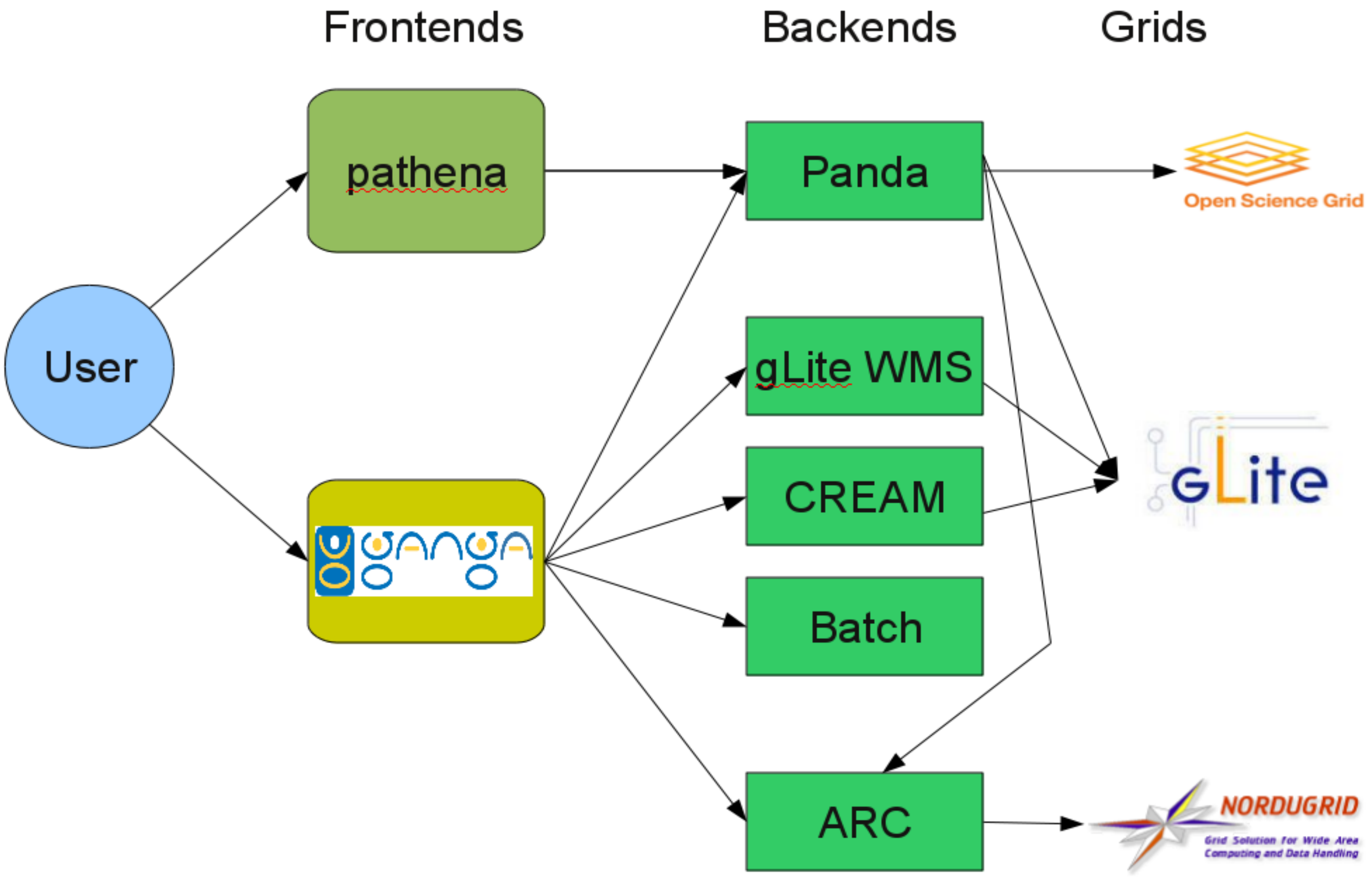
ATLAS Distributed Analysis

3 Feb 2011 - ADC Retreat

Dan van der Ster
Johannes Elmsheuser

ATLAS Distributed Analysis Layers

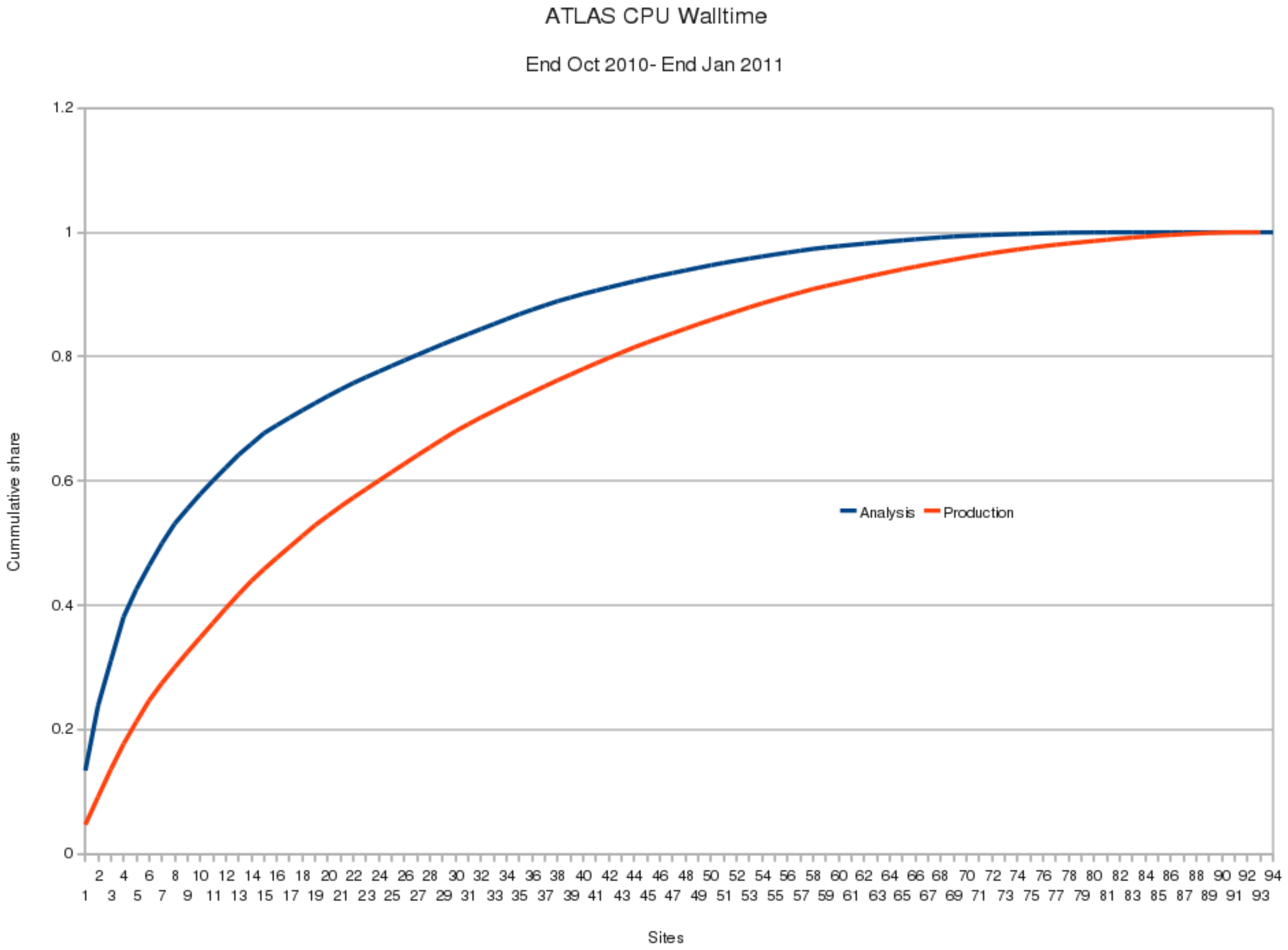


Data is centrally distributed by DQ2 - Jobs go to data

DA Situation

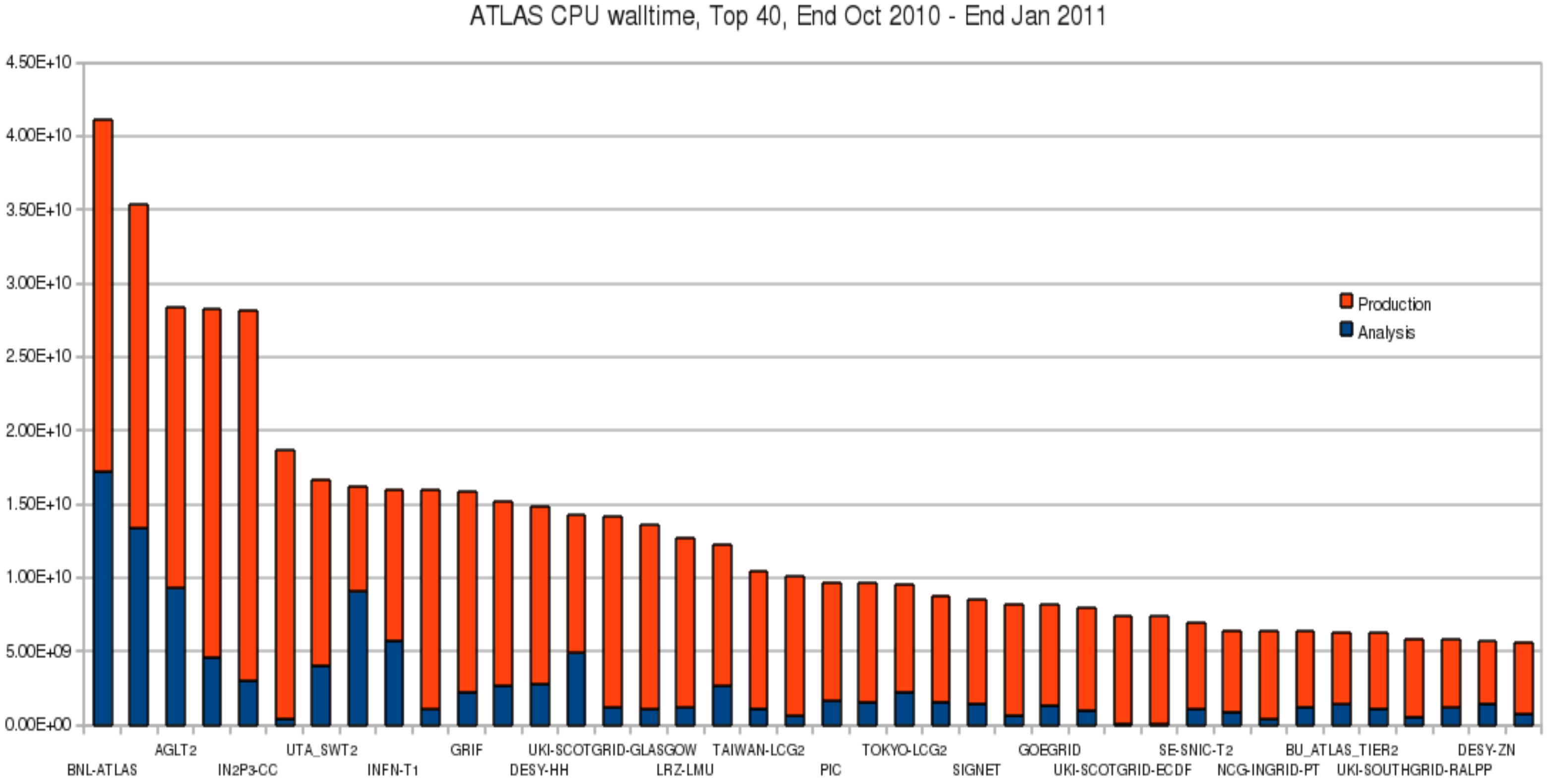
- Steep rise and constant use of user analysis since 7 TeV collision started
- Some power users with many many jobs - is the job output actually looked at ?
- No overall resource saturation so far - still some stronger and weaker clouds and unbalanced job distribution (see later plots)
- ROOT analysis of group D3PDs has developed as second largest workflow
- Panda has developed as user favoured backend
- pathena/prun have majority in user job submission
- Ganga actively advertized in all clouds ?
    - Remember: Ganga can submit to all possible resources

Analysis vs. Production (I)



- Numbers from Historic Dashboard
- Analysis is not so well balanced among the sites as production

Analysis vs. Production (II)



- Analysis share in the cloud:
  - US 33%, DE 15%, FR 17%
- Clear bias of analysis towards US cloud

1. BNL-ATLAS
2. CERN-PROD
3. AGLT2
4. FZK-LCG2
5. IN2P3-CC
6. RAL-LCG2
7. UTA_SWT2
8. WT2
9. INFN-T1
10. UKI-LT2-QMUL
11. GRIF
12. NIKHEF-ELPROD
13. DESY-HH
14. MWT2_UC
15. UKI-SCOTGRID-GLASGOW
16. UKI-NORTHGRID-MAN-HEP
17. LRZ-LMU
18. SARA-MATRIX
19. TAIWAN-LCG2
20. NDGF-T1
21. PIC
22. IFIC-LCG2
23. TOKYO-LCG2
24. TRIUMF-LCG2
25. SIGNET
26. CA-SCINET-T2
27. GOEGRID
28. CYFRONET-LCG2
29. UKI-SCOTGRID-ECDF
30. SWT2_CPB
31. SE-SNIC-T2
32. PRAGUELCG2
33. NCG-INGRID-PT
34. IN2P3-LAPP
35. BU_ATLAS_TIER2
36. WUPPERTALPROD
37. UKI-SOUTHGRID-RALPP
38. IN2P3-LPC

Backends

- Panda has developed as the main user favoured DA backend
- Direct ARC submission has faded out (any known users?).
- glite WMS still has some users.
  - Do we need a date to stop support for ARC/WMS analysis
  - I.e. Should there be a date after which DAST will stop supporting?
- CREAM CE submission for site testing and power users
- Tier3, Batch system backends
  - LSF, PBS, SGE, Condor with Ganga submission
  - Panda @ T3

Frontends

- Should have a single suite of DA frontend tools.
    - Need to clarify the use cases
        - command line submission, scripted submission, monitoring, job management, task bookkeeping, multi-stage jobs.
    - One tool cannot do all, but we should reduce overlapping use cases.
        - Wherever possible code should be reused -- i.e. extend the AthenaUtils and job mgmt libraries.
    - The DA suite should be distributed as one pkg.
        - Centralized Docs/Tutorials for this pkg.

- Should have one single WN code library:
    - No matter how jobs run (Tier0/1/2/3) via batch or Panda, the WN code should be the same.
    - This is needed to unify data access patterns, job error handling, and job monitoring/accounting.
    - dedicated ROOT wrapper necessary

Bookkeeping

- Task Bookkeeping should be more automated.
  - Today bookkeeping can be complicated by jobsets, retries and rebrokerage.
    - I.e. currently the easiest way for users to know if all data has been processed is to look at the output container and check if all expected output files are there.
  - Need for a new tool or modification of an existing tool with more intelligent capabilities.
    - i.e. users should not see jobs that have been killed/retried.
  - Tool could provide a data-centric view on the jobs. I.e. here is a list of the input data, and click to see the user's jobs that processed that data.
  - Task bookkeeping should be achievable via the web or CLI.
- Metadata handling
  - User output metadata in AMI ?
  - Read metadata from AMI (e.g. inputfilepeeker from AMI ?)

DA Front-End Use-Cases

We need to agree on the required use-cases. Proposal:

Submission:

- CLI submit athena/root/generic jobs. Simple & fast.
- API submit athena/root/generic jobs. Essential for power users, "multi-stage" analyses (e.g. pseq, gangatasks, other scripts). Also essential for apps like HC.
- The CLI interface must be a simple wrapper around the API.

Bookkeeping / Job Mgmt:

- Interactive Job bookkeeping. kill/retry. Copy, tweak config, submit would be nice to have! "Task" management is essential (transparent retry/rebro).
- API job management. Needed for the same reasons as the API submit.
- The interactive job bookkeeping must be wrap around the API job manager.

Monitoring

- Should be one single DA monitoring tool, well integrated with the DA clients. (this is already in hand with ADC Monitoring)

- Simple job/task management tasks should be accessible via the web monitoring.
  - I.e. Killing/retrying a job or task
  - Single job peeking (at what scale ?)

Output Merging/Multi-Stage Jobs

- Small outputs for DA jobs should be merged by default.
    - Need to implement a post-job merge for all DA jobs in some cases -- i.e. many small outputs.

- This is related to more generic multi-stage tasks, currently possible via psequencer/prun and GangaTasks -- rationalization of these tools should be investigated

Optimization

- The Grid should have a fast response time.
    - Users report that response time from local batch access to a site can be much faster than Panda access. Need to optimize job response time.
    - Optimize input access: copy-to-scratch vs. directIO vs. FileStager
    - Optimize input data distribution vs. free CPUs (data caching vs. pre-placement)
    - We obverse some fraction of looping jobs,(e.g. crashed ROOT session) - need to kill these earlier
- Grid access competes with local PROOF access
    - e.g. D3PD/NTUP processing much faster on dCache SE for small/medium sized datasets

DA infrastucture and related tools

DaTri:

- Improvements in user friendlyness ?
- Bulk operations ?

DQ2 tools:

- Frequent complaints about output download failure - improvements?

HammerCloud:

- In routine operations
- Actively blacklisting sites - conservative whitelisting planned
- New tests ?
- Now also adding some prototype prodsys tests
- Since beginning only 2-3 persons involved - more support from other ADC areas ?

User Support

- Provide a better user support tool:
  - It has been reported many times by users that it is a common practise is to filter the DA-help mailing list into a separate folder, which they never read. If this is prevalent, it makes user2user support almost impossible.
    - Goal should be to have predominantly user2user support, with an expert shifter available for new/difficult problems.
  - Investigate an integrated FAQ/Forum tool.
    - Shifters/users should be able to easily turn a thread/discussion into an FAQ. When a user reports a problem, the system should check for other similar threads or FAQs before creating a new thread. DAST should have moderation access to this forum with ability to highlight "the solution" posts in a thread, and to close threads.
  - Need a more effective way to send announcements/service notices to users.

Tier 3 issues

Some questions from Doug (overlap with previous points):

- What types of analysis will people be doing in their local clusters?
- Do have the sufficient monitoring hooks in analysis jobs to measure performance
- What about Proof cluster? What about Proof on demand?
- Are analysis codes sufficiently optimized and robust? Is this something where PAT can help?
- What is the overall analysis chain? How does the entire system fit together?
- Are there places for optimization and support labor reduction?

Manpower

- DA tools cover a very wide range of ATLAS software and grid activities
- All kind of (sometimes very technical) aspects: athena, DQ2, Panda, grid middleware etc.
- Only a handful of people involved in the development since a long time
- Have we reached the consolidation phase ?
- New developments may require new manpower ?
- Long term support of the current team ?