

Discussion and Common Analysis suite developments

DA user survey summary slide

Summary and Conclusions

- People are getting their work done, but improvements are necessary
- Main points:
 - Grid reliability is the most important issue. Still many resubmissions (10-20%). We need to improve reliability, or more likely we should make the tools more helpful with failed jobs. Need to investigate better job bookkeeping and automatic retries.
 - Need to speed up job submission and improve auto-configuration.
 - Almost all users want to dq2-get their output. It should better handle resuming downloads and the general process should be more friendly (see text comments for suggestions).
 - The main user support issues are related to the high list volume and archive searchability. (introduce new webboard ?)
- AOD and D3PD most popular input
- D3PD and ntuple/histos most popular output
- Surprisingly few people seem to use PROOF

Summary slides of ADC retreat Napoli

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.

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.**