

Automation: DPA

[ATLAS Distributed Computing TIM \(Stony Brook\)](#)

22 January 2025

Rod Walker (LMU) & Timo Wilken (UTA)



- all ATLAS requests tracked in a [ticket](#)
- change management, notifications?
 - DPA try not to change things out from under sites, but automatic e-mails might help
 - more user-friendly, human-readable changelogs
- even more fine-grained permissions?
 - vs. cloud-based / central teams
- automatic updates for corepower (etc)

- Speed-up the task tails
 - Reduce target walltime at 90% done: already done, pending impact evaluation.
 - job cloning where 1st to run kills the clone: so far used for HLT&Validation. Works for users too.
 - if trust the bookkeeping, and don't move inputs, then it is cheap - just a DB record
- VP-lite: enable remoteio overflow for busy sites
 - Why Lite? Remove dependency on every storage in the world
 - PQ has directio access to **all** data on 1 or more **regional** sites
 - Brokerage can choose the PQ when PQ with local data is busy.
 - or clone jobs to both
 - use ddm endpoint [status](#), for DT or exclusion
 - Read via Xcache to throttle access and cache

- Big advantages
 - Distributed data access, federated identity, shares and priority
 - No issue that collaborators have no local resources
- Users do like Ixplus, NAF, ...
 - User home area persistent & visible
 - OTOH Panda sandboxing is very powerful
 - Interactive access and fast turnaround
 - will not leave idle resources waiting for users (run production instead)
 - oversubscription of interactive resources, is de facto batch access
 - suspicious of demonstrations on dedicated resources
 - Faster push-out of production: suspend and resume G4
 - ARC push not great for turnaround
 - Posix access to human-predictable paths (not .../a0/b9/..)
 - Users like /eos but root turls, or directory of links, equally good if hidden with tools.
 - Multi node or GPU: ML, Columnar analysis?
- Propose to include Panda in AF benchmarking tests

- Motivation: enable quick start for new users
 - Who don't read manuals. Concentrate on physics and code.
- Functionality
 - Standard patterns & frameworks built-in, e.g. PHYSLITE
 - Info boxes, recommendations, avoid conflicts
 - automatic advice and critique from DAST experts
 - pchain workflows, e.g. final merge to 1 file(per run)
 - Seamlessly add Analysis Facility workflows
- Starting from very mature command line clients(pathena/prun)
 - Output could be prun command to paste, including pre-steps (setup and build)
 - Post-x509, maybe tokens allow for central submission of git code.