



<http://atlas.ch>

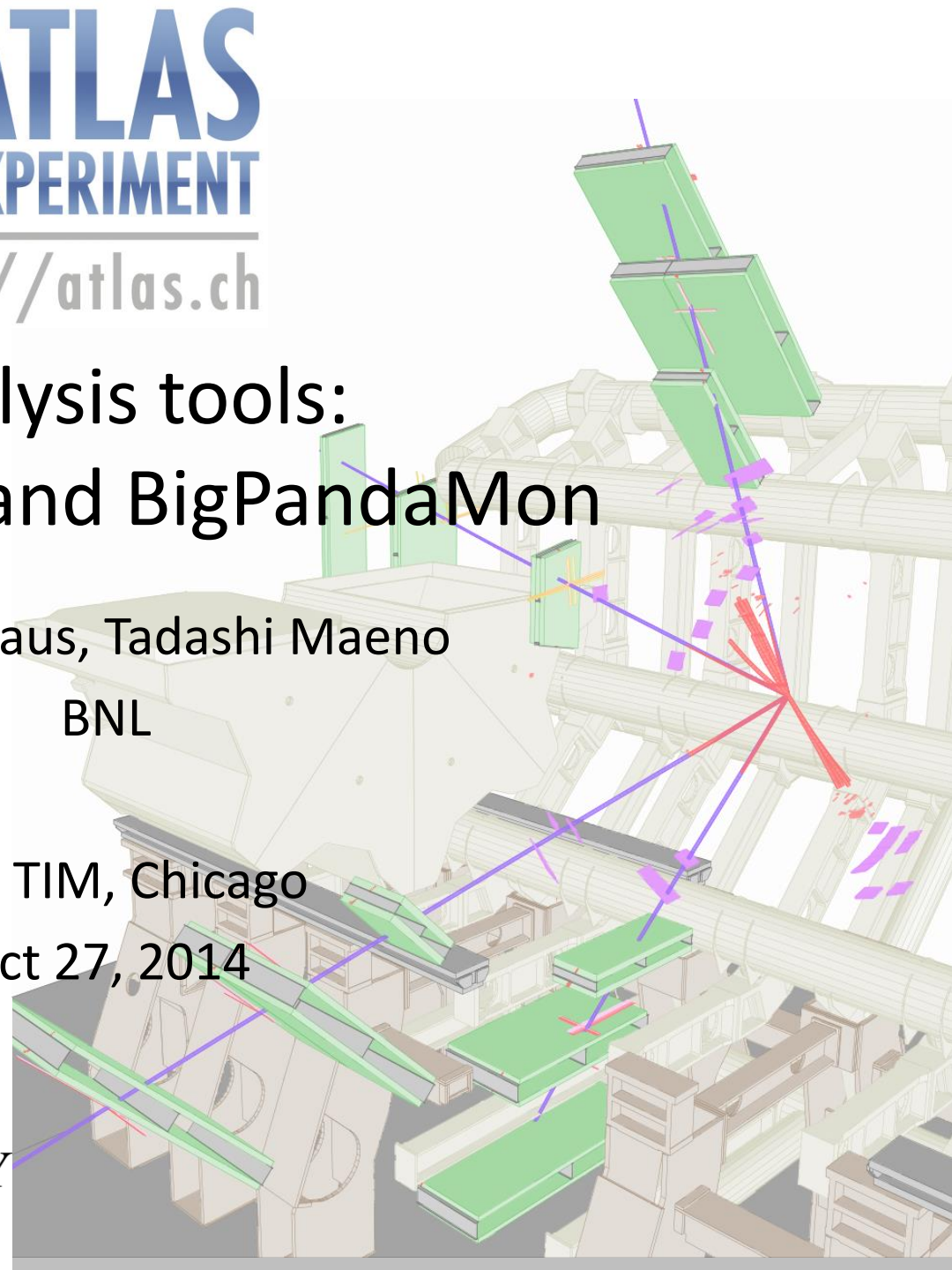
# Analysis tools: JEDI client and BigPandaMon

Torre Wenaus, Tadashi Maeno  
BNL

ADC TIM, Chicago

Oct 27, 2014

**BROOKHAVEN**  
NATIONAL LABORATORY

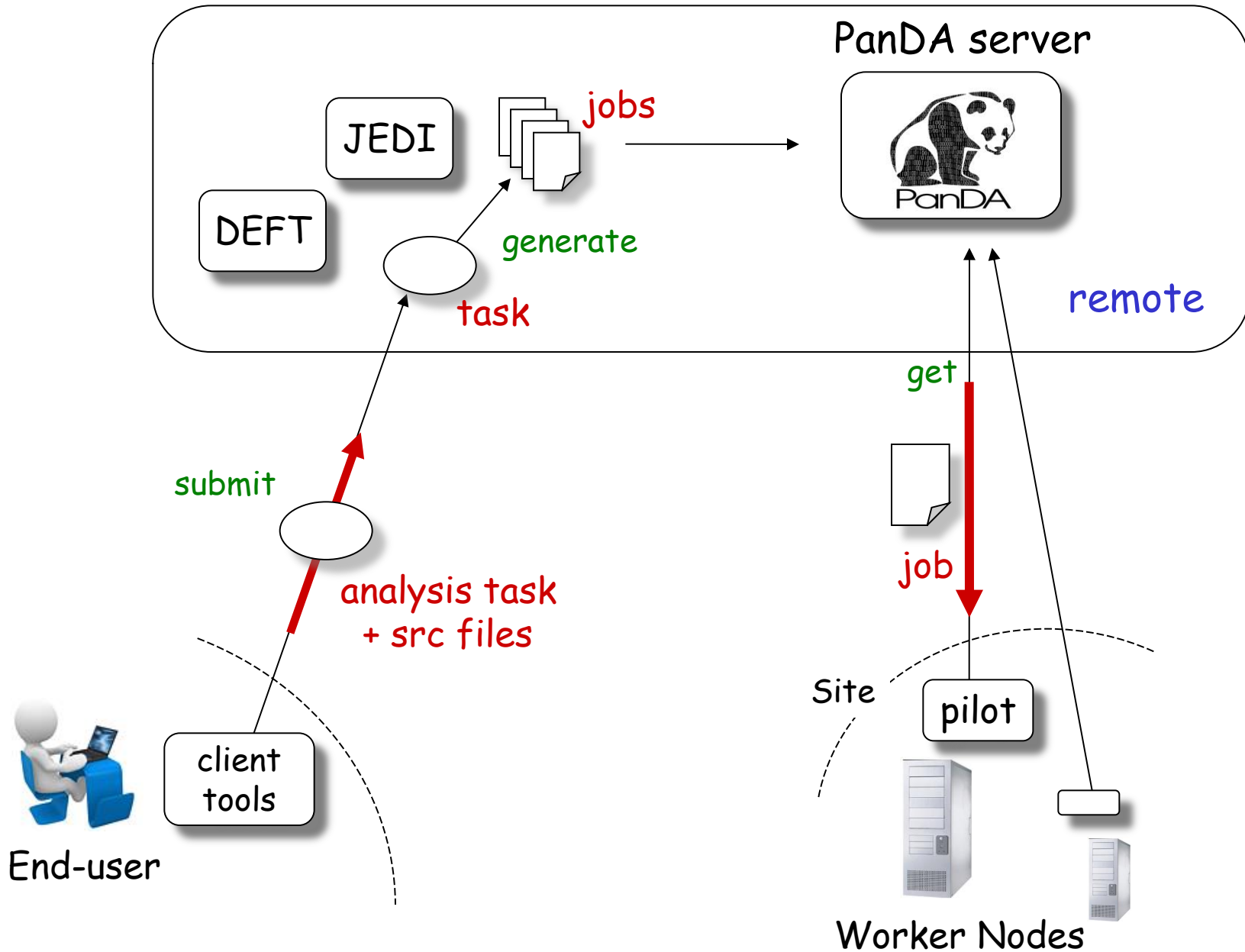




# Analysis tool user experience/feedback

- Tried to get user experience/feedback here, didn't succeed
- So some accumulated impressions & observations
- Thanks to Tadashi, Alden for input

# New Analysis Workflow



# Main changes to analysis with JEDI

- Tasks are submitted to the system instead of jobs
- Many client functions (job submission, retry, kill) are moved to the server-side
- Built-in retry mechanism at the same site and/or other sites
- Capability for task chaining
- Optimization based on job profile measured by scouts
- Tasks are more exposed to users rather than jobs  
→ task-oriented monitoring (JEDI monitor)

# Benefits



- The system works more coherently with user's perspective
  - Users are interested in tasks rather than jobs
- Simplification of client tools and centralization of user functions
  - Better maintainability
- Optimal usage of computing resources without detailed knowledge on the grid
  - Lower hurdle for end users
- Optimization of database access to get/provide task information
  - Performance improvements

# Differences from the old system (from user's point of view)

- Command names and command-line options same as before
- *“Don’t pay any attention to the jobs behind the curtain” is the big conceptual/behavior change*
  - *And can’t be a firm rule – jobs are how the work gets done*
- Some changes (*original list from Tadashi with current comments*)
  - [https://twiki.cern.ch/twiki/bin/view/PanDA/PandaJEDI#Major\\_changes\\_from\\_old\\_system](https://twiki.cern.ch/twiki/bin/view/PanDA/PandaJEDI#Major_changes_from_old_system)
  - Output file names are not sequential
  - One output dataset container is created per output type – *some unhappiness over this from some, I think he put in a bypass?*
  - Tasks with the same outDS cannot be submitted/executed in parallel – *was very unpopular, an ‘at your own risk’ option to by pass this constraint was put in*
  - Scout jobs run before jobs are generated in bulk – *a lot of “why this?” comments; increases end-to-end latency. But didn’t make Alden’s Top Five (coming)*
  - Subject of email notification is changed – *email was tuned as input came in, e.g. to accommodate users’ automated email handling*

# BigPanDA monitor for analysis

Bigpandamon was in reasonable shape for JEDI analysis at rollout but many monitor improvements after the rollout driven by analysis, ops, shifters

- 20140912: On task page highlight input containers, hide datasets by default
- 20140911: Task name search supports wildcarding
- 20140911: Show wait time, duration for jobs not yet running, completed
- 20140818: Job attempt# off for user page, JEDI jobs. Not meaningful in JEDI.
- 20140818: Task attribute summary added to user page
- 20140817: Output container list added to task detail page
- 20140817: Support clarified. Use DAST list, as ever, for dist analysis support
- 20140817: curl retrieval of json task data. Search on curl in [the help](#)
- 20140817: Merge (and other non-retry) jobs supported as a relationship type
- 20140816: Cache lifetime 3min for all pages, documented in footer. Report issues
- 20140816: [Restructured help with consolidated help page](#)
- 20140815: Show prodsys1 tasknames in error summary
- 20140814: Task summaries (huge and slow) removed from non-task dashboards
- 20140814: Task state names in task list, detail pages link to state doc
- 20140814: Task list in user pages brought up to date, with input completion
- 20140814: Wildcarding implemented for taskname in task lists
- 20140813: Works on mobile. Still tuning to do. Report bad issues please.
- 20140813: Add task searches for campaign, project, stream, tag
- 20140813: Add project, stream and tag summary info to task lists
- 20140813: Auto redirect pre-JEDI task detail pages to old monitor
- 20140813: Show in, out datasets on user pages and job lists
- 20140813: Fill state colors in dashboard task, job summary tables
- 20140813: Filled color in the status cells for job and task pages
- 20140812: Campaign added to task list and detail pages
- 20140812: Task pages use new DEFT-friendly 'superstatus'. See task page help.
- 20140812: Stalled task search added
- 20140812: Improved job listings on user pages (matching job lists)

# BigPanDA analysis monitoring feedback

- Biggest monitor issue as seen by users probably “what does it all mean”, at least it was for a while
  - Perhaps overtaken now, as task etc. counts have grown, to “needs to be faster”
- Users grokking task orientation of JEDI, with its implications (e.g. look at input file processing throughput more than job throughput to gauge progress)
- Changed orientation of user pages to emphasize tasks and de-emphasize jobs
- Added more task input based progress info on user pages that analysis users most look at
- Tried an approach of monitor-internal, context dependent doc, to give in-place help... not clear often it is seen or how much it is useful
  - Doc is modular and easily extendable, reusable; have doc appearing in multiple places, next to where it’s useful, without writing it N times
  - Consolidated doc page gathers all modules on one page
  - Doc modules used on multiple pages without duplicating sources; e.g. task and job doc included on user pages
- Please suggest improvements! We rely on DAST => developers as a major channel for suggestions to improve things for analysis users





# Alden's Top Five

*I posed him an unfair question today – what are the top 5 user problems/complaints off the top of your head. But he knows users well so they should be indicative. They are probably too narrow/detailed/in-flux to be usefully probed in a TIM session (ie this one) though...*

- Job problems often show up after “successful” submission as “broken” jobs, or “registering” failures.
- Job splitting behaves differently in new versions, and not all of the previous mechanisms do the previously "expected" thing.
- Event picking/GRL ambiguities — it’s in flux, and users are confused on occasion.
- Visible rebrokerage confuses users, who largely worked in ignorance before.
- Specifying a smaller subset of events in one file for testing or limited runs is difficult.

*There is a “what is going on here?” theme to them – need better visibility, understandability to what the system is doing*

*Some things I thought would be in it, aren’t: scout jobs adding to end-to-end latency, difficulties in getting (all) your outputs back*

# What should we do next

- Light at the end of the tunnel on large consumers of core developer time, e.g.
  - Getting pandajediprodsys2rucio ready for Run 2 and commissioned
  - Integrating opportunistic resources: HPC support, event service (still a fair bit of tunnel left)
- Plus we are actually getting some new manpower
  - 2 FTEs in the (Big)PanDA effort. ‘Paul replacement’ (UTA, can be CERN based) and ‘Jarka replacement (BNL based)
- We will surely be absorbed by existing todo lists feedback from the new tools in strenuous use, but can start to think about where to make the next pushes
  - This TIM’s horizon is up to ~6 months and the question is relevant on that timescale

# What should we do next?

- Off-the-cuff for discussion, possible big push or focus areas
  - Information: gathering it, communicating it, using it. More intelligent automation, feeding more/better info to operators/users.
    - Error conditions, analytics on jobs
    - Metadata (right now the stall in bigpandamon dev is complexities in concocting flexible efficient metadata searches when our metadata is in a biglongstring (dataset, task name). This is stupid.
  - Widely roll out remote data access: understand it, optimize it, render it efficient and debuggable, support it (and the users in using it), at the level needed for wide rollout
  - Data accessibility, integrating lightweight user friendly data access mechanisms (ie http)
  - Dynamic resource-based queues we've already decided we're doing; may be big enough that it effectively is on the 'next big thing' list