Panda at Tier3(g)

More info at https://twiki.cern.ch/twiki/bin/view/Atlas/PandaTier3

Torre Wenaus

U.S. ATLAS Tier3 Meeting

ANL

June 9, 2010

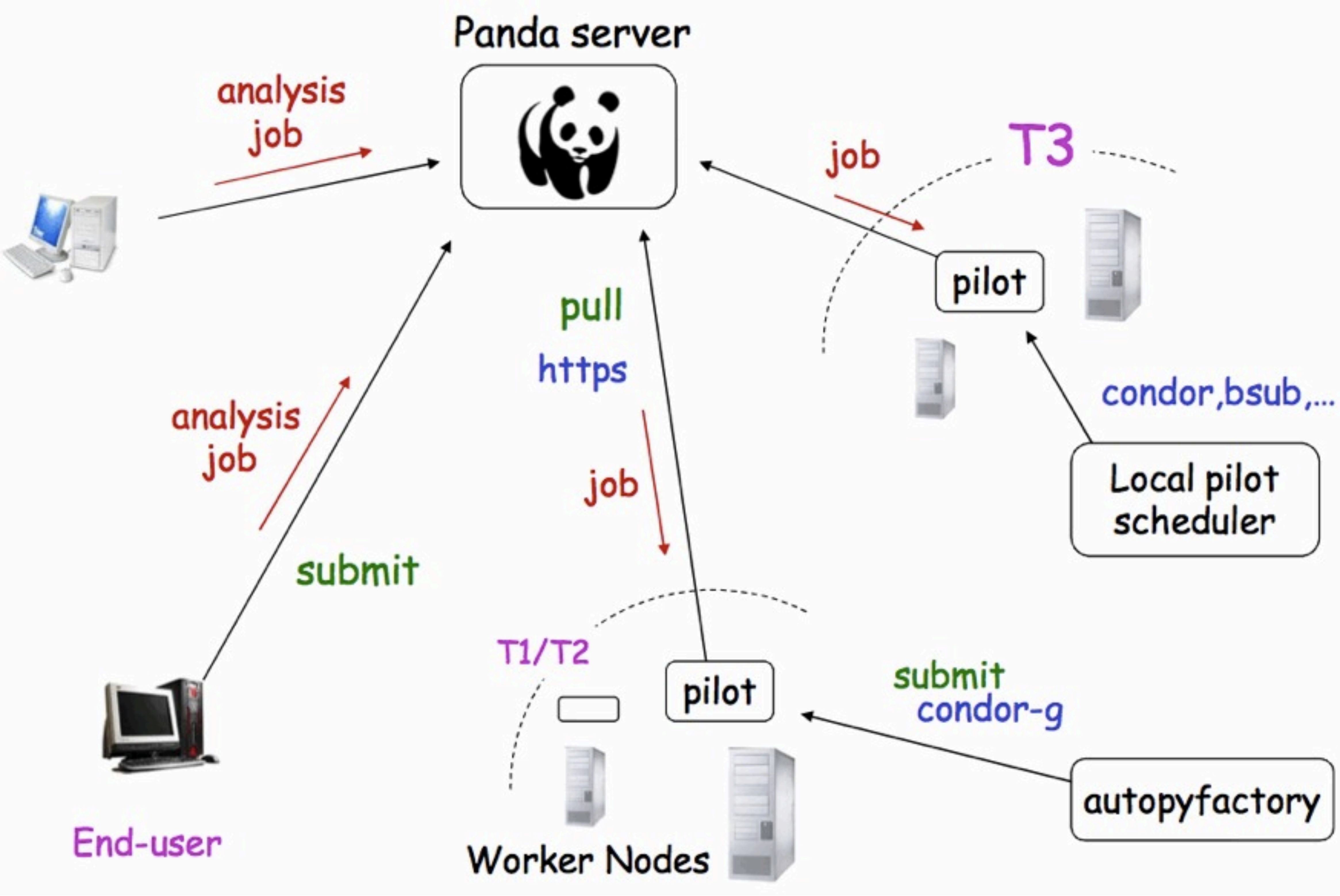
Panda Support at Tier 3

- The objective: support Panda usage at any Tier 3 that (as much as possible) works just like Panda anywhere else
- For on-grid Tier 3s this is trivial; they *do* work like Panda anywhere else
  - Hence this talk doesn't address on-grid Tier 3s any further
- Tier3g is something else; its off-grid aspects require changes in Panda's operation and support
  - But not big changes
  - To the user, using pathena/prun/GangaPanda looks much the same, we have hidden the differences as much as possible
    - The difference we can't hide is 'no datasets at T3g'
  - For the Tier3g site operator, they do need to take on a burden that on-grid sites don't: pilot submission

Panda at T3g

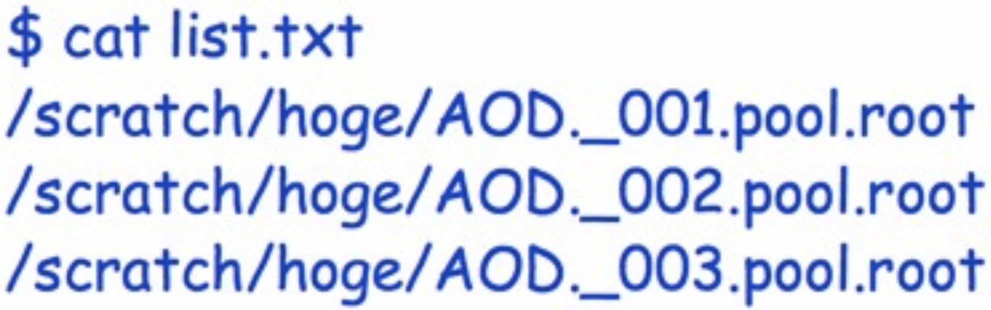
- Key aspects of ‘off-grid’ nature of T3g for Panda:
    - No DQ2, no gatekeeper/CondorG
- Hence pilots are submitted local to the site, direct to condor
    - Only condor local batch supported; others could be added
- User input file specification is via file list, not dataset
- Output ‘dataset’ maps to directory
- No file registrations or catalog dependence in pilot
- Panda monitoring available, same as for on-grid sites
- Can use Panda’s access controls to limit usage to defined list of locals+friends (site managed)

Panda at T3g Schematic



Usage and File Handling

- The ‘no DQ2’ nature of a T3g is flagged to Panda via a site configuration DB parameter (schedconfig.ddm=local)
  - Indicates no DQ2, no file catalog (LFC)
  - Dataset and file lookup by Panda is suppressed
  - Instead, Panda expects to receive a file list
- Users need to provide list of physical filenames of input files
- Conventional pathena:
  - pathena jobOpt.py --inDS dsname --outDS outname
- pathena to T3g:
  - pathena jobOpt.py --site ANLASC --pfnList list.txt --outDS outname



```
$ cat list.txt
/scratch/hoge/AOD._001.pool.root
/scratch/hoge/AOD._002.pool.root
/scratch/hoge/AOD._003.pool.root
```

File Handling (2)

- The input file list is given to transformations on worker nodes and athena reads the inputs directly
  - EventSelector.InputCollections used in job options to pass the file list to athena
  - From pilot's point of view, they are seen as no-input jobs
  - The file specs can be in any form that athena can understand (thus remote I/O specs like xrootd://... are OK as well)
- Output files are written to a subdirectory under a top level that is part of the site configuration (schedconfig.se)
  - schedconfig.se = /scratch/outputs
  - pathena --outDS user10.username.aaa
  - output files found at
    - /scratch/outputs/2010/username/user10.username.aaa/fileN
- Output files owned by the pilot submission account

Local Pilot Scheduler

- Conventional pilot schedulers run ‘centrally’ and use CondorG to submit pilots remotely
- T3g with no Compute Element and no CondorG support doesn’t have this option
- Pilots have to be submitted locally
- The pilot factory used in the US, AutoPilot, supports local pilot submission to the Condor batch system (only)
- This functionality will be added to autopyfactory as well
    - autopyfactory is the soon-to-be ATLAS-wide standard that will replace AutoPilot (timescale several months??)
- If support for other batch systems is absolutely needed, this needs to be put on the autopyfactory todo list ASAP

Pilot Authentication

- In CondorG based pilot submission, grid proxy of job (pilot) submitter is sent by CondorG along with the job
- The proxy and its VOMS attributes is used by the pilot to authenticate itself to the Panda server
- Condor doesn't forward the proxy (except see below) so another approach was needed
- For T3g, pilot authentication to server is based on regular expression match to WN name (schedconfig.allowednode)
- *But w*ith Condor 7.4, proxy sending is supported for local Condor
- Implemented and tested with AutoPilot; works
- Will be integrated into autopyfactory as well
- So we can move to more secure proxy-based authentication when we wish

Access Control

- Site level access control available in Panda for some time, ready for use by Tier3s to control who can use the resource
  - https://twiki.cern.ch/twiki/bin/view/Atlas/PandaUserInfoTool
- Designed for easy use by end-users and easy management by site responsibles
- Various configurations are possible
  - Limit access to a specified group of users
  - Allocate a fraction of CPUs for general ATLAS, and the rest for the local group only
  - Boost the priority for a group of users
- Use the **puserinfo** package, part of the pathena client interface
- Users can add themselves to the access group for a site
  - An email goes to the site responsible to approve or reject
  - And/or the site responsible can add/remove users
- Use the pathena --workingGroup option to use a group that has special rights on the site
  - `$ pathena --workingGroup poweruser --site ANALY_XYZ ...`

Site Requirements

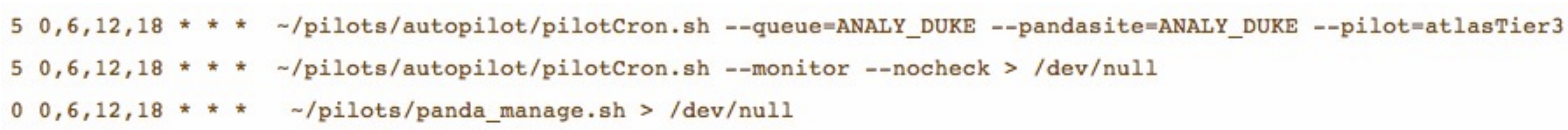
- Outbound http, https support (non-standard ports; may be proxied). Worker nodes and pilot submit host
- Apache web server configured to provide browsing of a directory tree accessible to the pilot submit host, and job output files
  - Used to support pilot and job monitoring in the Panda monitor
- Condor based batch queue (unless/until more batch systems are supported)
- Functional crons. Used for pilot submission, monitoring, management.
- Subversion support; access to CERN subversion server
- Disk storage areas for input data, job outputs

Site Preparations

- The following information must be provided to the Panda team:
  - Pilot submit host configuration
    - e.g. http://panda.cern.ch?tp=host&host=atl009.phy.duke.edu
    - machine name of the submit host
    - directory path where pilot logs are to be written
    - web URL that maps to this directory
    - name pattern for worker nodes (eg. atl[0-9]+\.phy\.duke\.edu)
  - Panda site and file management configuration for the site
    - e.g. http://panda.cern.ch?tp=queue&id=ANALY_DUKE
    - Panda site name (eg. ANALY_DUKE)
    - Condor job def for submission to the right queue
    - top level directory for job outputs
    - web URL mapping to the job output directory

Installation and setup

- Details are at https://twiki.cern.ch/twiki/bin/view/Atlas/PandaTier3
- In brief:
  - Establish the account you'll use for pilot submission
  - Check out the code from CERN svn
  - Set up panda_setup.sh setting env variables
  - Set up crons for pilot submission and monitoring
  - Set up panda_manage.sh cron to do log cleanup etc.
  - Check that pilots are running: http://panda.cern.ch?tp=pilots&accepts=ANALY_DUKE
  - Submit some jobs!



```
5 0,6,12,18 * * *  ~/pilots/autopilot/pilotCron.sh --queue=ANALY_DUKE --pandasite=ANALY_DUKE --pilot=atlasTier3
5 0,6,12,18 * * *  ~/pilots/autopilot/pilotCron.sh --monitor --nocheck > /dev/null
0 0,6,12,18 * * *   ~/pilots/panda_manage.sh > /dev/null
```

Future Developments

- More dataset-like file handling
  - Common directory structure conventions with DDM, cf. Alessandro's talk. Avoid need for file list spec
  - Compatibility with pq2? (PROOF-based DQ2-like cataloging)
- Worker node-level file brokerage
  - Based on the work Charles Waldman and Tadashi Maeno did to implement this capability for pcache
    - But not limited to pcache
  - Pieces are there, needs someone to take it in hand and be a guinea pig for trying this out with help from Charles and the Panda team
- Jobs under a single uid
  - Is this a big issue? There is no attractive solution, so I hope not
    - In principle users can submit their own pilots, but adds complexity, reduces usability, increases latency
    - You certainly don't want to use glexec
- Files under a single uid
  - Could be addressed relatively easily at site level with a privileged cron resetting ownership

Panda @ T3g Status

- Queues set up at Duke, ANL; pilots flowing
  - Tested successfully with 15.6.3 pathena jobs at ANALY_ANLASC
  - I personally haven't tested real jobs at ANALY_DUKE
- Data source and sink functionality in DQ2-free mode validated
  - Outputs to directory structure: Year/Username/Dataset/File
- PandaTier3 wiki gives setup information:
  - https://twiki.cern.ch/twiki/bin/view/Atlas/PandaTier3
- Ready to add more Tier 3s; get in touch!