
Software issues

Nicolas Berger

2010-12-02

2010 Fall Reprocessing Post-Mortem session

Reprocessing software timeline

- Aug 30th : Software freeze
- Sep. 2nd : 16.0.0 built
- Sep. 24th : Conditions freeze
- Oct 9th : 16.0.1.5 built → express stream reprocessing
 - 242 changed packages wrt 16.0.0.
- Oct 29th : 16.0.2.3 built → bulk reprocessing phase 1
 - 91 more changes
- Nov 15th : 16.0.2.5 built → phase 2
 - 45 more changes
- Nov 23rd : 16.0.2.7 built → job recovery
 - 29 more changes
- Dec. 1st : Reprocessing done

Release Tools: What went well

- TagCollector is a mature and extremely useful tool
 - Open Approval procedure is very convenient:
 - Less prone to errors/delays in communicating the requests
 - Developers can give more descriptive (and first-hand) justifications
 - Support/Tests Passed mechanisms very useful to assess what is working as expected.
- Savannah allows good communication with developers on problems
 - Dedicated tracker for reprocessing.
 - 282 bugs matching 16.0.X.Y releases in savannah.
 - 250 other bugs in reproc. tracker, mostly related to failed tasks.

Release Tools : What went less well

- Open approval still not well known, many requests for tag inclusions from developers.
- Many tags never get supported. Usually still collected, to avoid delays...
 - Also some cases of “supported” tags that shouldn’t have been, but a small minority.
- No mechanism for developers to specify which tests have been run to validate a given tag
 - For reprocessing we required Reco_trf AMI=q12x.
 - Most developers ran them
 - Some also specified it in the justification, but not generally done
 - Usually led to asking developers for each new tag, and/or rerunning the tests.
- Many tags never get the “tests passed” label after inclusion in a VAL nightly.
 - A mechanism for developers to specify which tests were run would also be useful.

Releases and caches

- 3 full releases, 15 prod caches built over 3 months
 - More than 1 build/week
 - Needed to run larger test jobs on the grid.
- Release/DBRelease building timely and efficient, good communication with Atlas Release team.
- Would be nice to be able to run production jobs from nightlies.
 - to spot problems faster (e.g. slimmetadata problem in 16.0.1.4, #73945, which required another cache the next day).
 - might reduce the need for some caches ?
- Timing:
 - Strong pressure to build before the weekend so test jobs can run during Sat/Sun. =>many (most?) builds occurred on Fridays
 - Some slipped into the weekend: found that due to manpower issues, no big gains compared to waiting for Monday.
 - Not sure if anything can be done ?

Software Changes

- Most updates went in on time (before Aug. 30th) with no issues.
- Problematic cases
 - Updates that came late
 - Major updates that came on time but needed a lot of tweaking afterwards (jets, muons).
 - Some updates only became needed after updated conditions available (e.g. MuonBoy changes due to new CSC alignment)
 - Should maybe set conditions and software deadlines closer together to converge faster...
- For jets, problem compounded by the fact that 2 disruptive changes (EDM migration and change of collection names) happened simultaneously, with many other packages affected.
 - ERRORS from Collection name issues still being fixed in mid-October (#73127)

Tests

- ATN tests
 - Some tests usually not up to date (if quickly changing "normal" behavior, e.g. trigger menu changes). Usually O(5) failed tests even for good builds.
- RTT/TCT
 - Web interface
 - Lots of useful information (job status, color-coding log files)
 - Not always reliable in flagging problems
 - Takes a long time to go through all jobs
 - Using grep (for "ERROR", "FATAL" and "successful run") commands on log files in AFS location quicker and more reliable.
- What tests to check irreproducibility ? Important for frozen Tier0 policy. Need to deal with AMD vs. Intel, etc.

Physics Validation

- Faster turnaround for PhysVal is desirable
=> would need samples produced faster
- Not enough stats available before ES reproc.
=> some problems only detected then (dead OTX correction). How to spot earlier ?
- DQ histograms:
 - ES reprocessing did not produce DQ histograms for CosmicCalo (because no ESD produced ?)
=> could not perform some checks (noisy LAr Cells). Should be enabled in next reprocessing.
 - Do we need DQ histograms for bulk processing ?
- Some problems caught (too) late (STACO extrapolator issue, #75645).

Bugs

- Reprocessing tracker
 - Posting directly to a subsystem tracker (or the reco tracker) usually preferred (more visibility, faster responses)
- Response time
 - Sometimes long, but biased by retrospect (only remember the bad cases)... Serious problems almost always fixed quickly.
 - Large part of delays due to difficulty with grid jobs
- Crashing grid jobs
 - Not always trivial for developers to recover input files or even log file.
 - Basically relied on Sasha extracting RAW events by hand. Need a better working model...
 - bus errors : caused a lot of problems in production, under investigation.
 - Problems with undetected memory corruption issues: need to improve reporting of information through pilot job.

Conclusions

- Things went well overall – working and validated software delivered (mostly) on time, results seem good (with a couple of exceptions).
- What could be improved
 - Converging to a working cache could have been much quicker
 - Ensuring that updates included on Aug. 30th were mature.
 - Make conditions deadline same as software deadline
 - Problems could be spotted earlier
 - Running prod jobs from nightlies
- More written procedure would be good.
 - Especially since turnover is fairly rapid.
 - Current process relies on handover from previous person and briefing by experts.