Stripping status

Michael Alexander[1], Steffi Reichert[2]

[1]Glasgow, [2]TU Dortmund

Computing workshop 2017/05/16

Outline

S24r0p1, 2015 incremental restripping

- Software & options in AppConfig deployed yesterday.
- Tested & working.
- Production steps ready but currently issues running local tests of the steps, as detailed here, as ancestor files aren't handled correctly. Under investigation.
- Validations to be started today.
- Bandwidth found to be ∼31.1 MB/s. Estimate ∼46 TB per copy of full output.

Stripping29, 2017 concurrent stripping

- Builds on Stripping28. Hopefully most issues already resolved.
- Test on 2016 data, with samples for both tight and loose HLT1 configs - possible as no new lines going to HLT2 full stream, TBC on new TCK later this week.
- Deadline for commits set to Friday 26th May. MRs submitted thereafter unlikely to be included as schedule is tight.
- Will review which lines go to full DST and why, and usage of raw banks vs RelatedInfo tools.
- New tool to store Calo data without needing full raw bank being developed by Olivier.
- Hope to include stream optimisations, to be discussed in the next talk by Radoslav.
- Commissioning JIRA task here.
- Few commits so far.

Stripping30, 2016 pPb and Pbp stripping

- Line commissioning almost complete.
- Retentions $\sim$7 % for pPb and $\sim$14 % for Pbp.
- Waiting on final details so coordinators' tests can be run (we're busy with S24r0p1 and S29 just now anyway).
- Need to backport to stack of Brunel v51r1.
- Commissioning JIRA task here.
- Stripping for 2015 PbPb and PbAr data (Stripping31?) soon to follow.

Outline

Software stack

- New campaign & deadline is announced - whether first stripping of new data, full restripping, or incremental restripping, procedure is much the same.
- Stripping commissioning requires DaVinci release to use.
- Full software stack is normally released well in advance, with preliminary Stripping version - thanks to developers & maintainers of the rest of the stack (Eduardo, Rosen, Vanya, Alex P., et al) & deployment shifters.

Users & liaisons

- Users retune existing lines or (more often) add new lines.
- Retentions for each WG are checked by liaisons & OK'd by coordinators - required to be < 0.05 % for full DST, < 0.5 % for uDST. Timing < 1 ms/event.
- Liaisons collect config dicts for stripping modules to be run & commit to the StrippingSettings archive (like a TCK in Moore).

Coordinators

- Run full tests of all WGs, check global retention & bandwidth. Fix bugs as necessary.
- Create "stripping statistics" page for the campaign, get users to check that lines are running as expected. Pester line authors about high retentions & bandwidth.
- Create StrippingArchive of line builders & CommonParticlesArchive of CommonParticles.
- Add options for StrippingCache to DaVinci.
- Deploy software, create production steps, test & run a validation.
- Users check validation output & give OK or raise issues, in which case go back to software commissioning.
- Once validations are OK'd full productions start.
- Finally, generate web documentation.

Outline

Continuous integration

- Add to gitlab to check for simple bugs.
- For any MR, test that code builds & all lines & streams can be instantiated using their default config.

Consolidation of options

- Tests at all levels (users, liaisons & coordinators, even production) generally involve copying existing scripts of few 100 lines and editing only a few lines.
- Could be consolidated into a single configurable with default config for production & options to enable various tests.
- More robust & flexible. Less effort to perform standard tests & make options for production jobs.
- Stripping configurable would probably have to live in DaVinci (unless we can decouple Stripping & DaVinci - would be nice, but past discussions have pointed out difficulties).

Scripted creation of stripping archives

- Fairly trivial for StrippingArchive & CommonParticlesArchive - can just use rsync, plus some functions to handle package structure.
- For StrippingSettings, each line builder module has a default config which is normally just copied by liaisons to the settings archive.
- Instead, liaisons could simply maintain a list of active configs for each WG for each campaign, then the settings archive can be automatically compiled from the default configs.
- Relieves some load on liaisons, though default configs need to be kept up to date.
- Corner case when two campaigns are being developed simultaneously (as now) and require different configs could be handled by branching.
- Mostly done already.

Automated coordinators' tests

- Use LHCbPR2 to run tests of retentions & bandwidth.
- Can be part of full integration tests of trigger, reconstruction & stripping.
- Facilitated by automatic creation of settings archive, etc.
- Avoid nasty surprises.

Python readable doc for lines

- Create Phys/StrippingDoc, like the current online doc, but contained in a python package.
- Contains a module per campaign with details of each line that ran.
- Details cover output location, output stream, whether it's DST or uDST stream, related info tools & their output locations.
- Ideally would also contain decay descriptors for the output of the lines - would require piecing together descriptors of all particle combiners in the line's selection sequence, not trivial.
- Can then be imported into analysis script to configure, eg DecayTreeTuple - just need to know stripping version and line name.
- Should make user analysis on stripping output (or validation output) much easier.

Generic scripts for analysis on stripping output

- Provide scripts to produce a tuple, or just plots of mass or other variables, from stripping output.
- Also scripts to check/plot data stored by related info tools.
- Put forward as tasks for last impact kit, lead by Iwan Smith.
- Should allow users to check that their lines & related infos work as expected during line development, & to check validation output easily.

MC productions without PID cuts

- Requested several times by users.
- Current implementation requires users to duplicate lines and remove PID cuts, and set a flag so they're only run in MC productions - a bit clumsy.
- Would be better to disable PID cuts globally somehow - hack LoKi so all PID functors always return True?

Conclusions

- S24r0p1 ready for validations.
- S29 in commissioning, should be very similar to S28. Hopefully to contain stream optimisations.
- Long wish list of potential improvements with the aim of easing commissioning process, checks of validation data, & user analysis on stripping output.