pROOT Applications

Zach Marshall (LBL)

With thanks to Mike Hance and Steve Farrell

US ATLAS S&C Meeting

20 Aug 2014

Work Flows

- Generally, in run 1 the procedure has been to work on the grid until the dataset is small enough to be handled locally, then DaTRI/dq2-get it and work locally for the last steps
  - Really, those last steps require fast turn-around, so this is reasonable
- What point grid work stopped is more *site dependent* than analysis dependent!
  - From what I've seen, groups with more disk space do less on the grid
  - Doesn't have to be this way, but it may be unchangeable
- We have two things to do from xAOD derivation to final product
  - Assuming the xAODs, derivations, etc all conform to expectation
- One step will be running all the systematics etc
  - Running this on the grid is a possibility. If EventLoop has nice functionality for shallow copies built in, then it will be a useful solution.
  - These are ROOT jobs, and so in principle could be pROOT (or pRun, or even pAthena, since it's xAOD)
- The final step is making histograms etc – that's ROOT, but expect it to mostly be a local operation

Openings for pROOT

- It should be clear what this is winning us exactly – if it's just for the S&C crowd, or if there is a win for users that is obvious
  - I'm wary of introducing a tool because we can without a clear benefit – we have a lot to get done right now, and having a tool that breaks at the beginning of Run 2 would be… bad.
- Could be used for the systematics step on the grid
  - Usually near-zero initialization time, so don't expect big wins there
  - If this is done with something like EventLoop, then it's just a matter of using a different back end, which is transparent for users
- Unlikely to be used / helpful for the final step
  - If file usage monitoring is the important part, then set a DaTRI switch for this (e.g. "replication to local resources" == do not touch)
- If the xAOD derivations are huge, and we have to re-derive smaller xAODs, then this is also a spot for pROOT to slot in
  - I really hope this isn't the case, but let's see
  - Could also have file usage monitoring implications

Branch Monitoring

- Beware of automatic branch monitoring
- While you can tell how often a branch has been accessed, there are two important factors you are missing
  - The look of horror when you realize that you don't have a branch that you want for debugging a problem and you are going to have to re-run all your data in order to get it included
  - The look of relief when you realize that you included the branch even though it wasn't part of your nominal plots, because it's going to save you weeks of pain and suffering
- Webcam monitoring aside, I don't see a good non-human way to deal with this
  - If it's just to construct a list of commonly copied but rarely used branches that humans can then go through, it might be useful – but be ready for the answer to be "tough, we're keeping them."

DISCUSS.