Running MC & User Experience

- Template for running general MC here:

  http://confluence.slac.stanford.edu/display/Gino/Template+for+MC+Generation+in+Gino

- I have run 3 generic MC sets with Gino so far:
  - 2 sets of 250k allGamma to look at effects of Tower A dead channel list
  - 1 1M allGamma run to test new TkrRecon beta release
  - All sets were single step (Source+Sim+Recon) producing MC, Digi, Recon and Merit root files
- Tools needed/used:
  - GlastSvc v9r12p1 needed to specify run id as string, not int
  - Web front end to upload task xml config file
  - submitTasks.pl to meter runs into the pipeline
  - Web front end to monitor progress
  - deleteRuns.pl, createRuns.pl to rerun failed jobs (if due to resources)
  - pipelineDatasets.cxx, pruneTuple.cxx to create concatenated merit tuple with pruning cuts

Operational Experience

- First two MC runs entirely smooth
  - Batch farm, Gino otherwise idle
  - Had to update config db manually to select batch allocation group to use (MC vs Data)
    - Matt says this has been added to next version of xml parser
  - Submitted 50 runs at a go; no problems; all succeeded in xlong queue with optimized build. Very little watching needed.
- “1M” allGamma run a little more eventful
  - SCS was shutting down, so xlong queues were stopped
    - 20% of jobs failed in long queue (non-opt GR HEAD1.403 – opt segfaulted consistently; Tracy alerted)
  - Coincidentally an old Oracle config had returned limiting db connections to 200 (was set to 500)
    - Jobs failed due to failing to get database connection
  - Hence much more resubmission of jobs
    - Wait until fewer connections in use
    - After a while, it seemed that slower barb machines taken out of service, so nomas ran the jobs in long queue ok!

Config, Monitoring and Cleaning Up

- Config
  - Batch allocation
    - Had to remember to hand edit the db to select correct group
  - Used emacs to clone tasks; had to know where to find source
    - No web editing nor xml download available (yet)
- Web front end showed status of runs, but
  - Could not filter by status (ie just show me failed runs)
    - A little harder to spot all the failed runs; and a little error prone. I killed a good run by reading the wrong line in the display.
    - More filter options would be nice (eg run or date range; I&T has a list)
  - Might be nice if the front end remembered the last task I looked at and come up with it
  - Could not use front end to see log files
    - emacs did the trick
- Cleanup
  - Manual use of deleteRun and createRun
  - Wrote perl script to delete ranges of runs by task
  - Manual erasure from disk of dead runs files (erased wrong one by mistake once)

Post Processing

- Needed concatenated, pruned files for Bill et al
  - Wrote Root class to query the db and return TChain of files for input run request [all runs; run range; list of runs]
    - Should be able to use this class in GlastRelease RootIo to select runs too; ditto for Data Server.
  - Modified DC1's PruneTuple to take a TChain as input
  - Started with Root 4
    - So far does not seem to work with Root3. Not sure why yet. May be moot if we move to Root4 soon.
  - Would be nice if pipeline could do this
    - But would have to know when to do it (MC's can be open ended after all)
    - Takes all runs from the task as input

Quibbles/Issues

- **GlastRelease version recording**
  - Gino records the version of the script it runs, not the underlying application
  - This is hard-coded into a shell script that sets up and runs Gleam
- **Can't prioritize tasks**
  - MCs will be serial (well, FIFO)
  - In next life, would be good to be able to juggle priorities between tasks

Summary

- Gino worked fine
  - DB Connection limit was a bit of a pain, but probably moot for limit of 500 (and probably could be raised again if need be)
  - Web interface is being improved – looking forward to it!