Monitoring Reprocessing

Notes relating to monitoring the Spring 2012 Full Pass7 Fermi Reprocessing project.

- Detailed configuration of project: https://confluence.slac.stanford.edu/display/~dragon/P202+-+Full+Pass7+Reprocessing

- CPU-intensive ROOT generation (full reconstruction) task
    - Pipeline task: http://glast-ground.slac.stanford.edu/Pipeline-II/task.jsp?task=74630144&dataSourceMode=Prod (One top-level stream = One Fermi data run)
    - Task directory: /nfs/farm/g/glast/u38/Reprocess-tasks/P202-ROOT

- Task control
    - trickleStream.py - long-running script to monitor current batch usage, and submit new top-level streams. Run only one instance of this script! I run it on my desktop machine, comet.slac.stanford.edu (RHEL6-64 machine), piping the output into a log file,
        - Starting trickleStream

          ```
          $ cd /nfs/farm/g/glast/u38/Reprocess-tasks/P202-ROOT/config
          $ ./trickleStream.py > tslogNN.txt &
          ```

          (Select "NN" to be next number in sequence.)
        - Stopping trickleStream - this can, of course, be done by killing the process explicitly. Or, one can halt gracefully by creating a file called 'stop' in the config directory, e.g.,

          ```
          $ touch stop
          ```

        - Monitoring trickleStream - watch the log file, e.g.,

          ```
          $ tail -f tslogNN.txt
          ```

- Job Step Failures
    - Monitor failures from the main pipeline page for this task
    - Typically there have been ~10-30 failures per day, mostly transient disk-access problems (NFS, xroot) solved by a single rollback.
    - All rollbacks are **manual** (no automatic retries internal to task) – to prevent serious problems from creating run-away rollbacks.
- Cleaning up xroot scratch space.
    - Script run manually every day or so. This typically liberates about 300+ scratch directories (one scratch directory per Fermi run/top-level stream), or about 10 TB of disk space.
    - Script lives in /nfs/farm/g/glast/u38/Reprocess-tasks/P202-ROOT/cleanup

      ```
      ./runCleanup.sh | tee xlogNN.txt
      ```

      where "NN" is the next number in sequence.
    - Note that this script utilizes 'xrd.pl rmtree' which is a non-synchronous command and can take upwards of 5 minutes to complete.
    - Monitor xroot space usage on this web page updated hourly: http://www.slac.stanford.edu/~wilko/glastmon/xrddisk_stat.html