

L0 Data Processing Sequence

Overview

This page describes the sequencing of disk and database usage during level-0 data delivery and processing, through the following stages:

- FASTCopy receipt
- Raw-archive ingest
- HalfPipe dispatch
- HalfPipe processing
- L1Proc dispatch

The attached [spreadsheet](#) records timing information for this process obtained from 15 level-0 transfer-package injections performed during ISOC Operations Simulation 1 in October, 2007.

FASTCopy Receipt

1. The FASTCopy daemon on glastlnx11 writes the incoming tarball to a staging directory on u23 as the file contents are received over the network.
2. The FASTCopy post-transfer script on glastlnx11 creates a time-based "archive" directory for the tarball on u23, mv's the file there, and unpacks it. Records are posted to Oracle identifying each level-0 file found in the tarball. A downlink_id is assigned and written to the fcopy_incoming table.

Raw-archive ingest

A cron job on glastlnx11 notices the new level-0 files, and does the following for each one sequentially:

1. gunzip's the file from u23 to glastlnx11:/tmp
2. reads the file and builds an in-memory per-packet index by scid, apid, and UTC hour. Also assemble lists of packets forming datagrams (for datagram-oriented APID's).
3. for each (scid,apid,utc_hour):
 - a. open the corresponding raw-packet archive file on u23 and merge it's contents into the index list, eliminating duplicate packets.
 - b. open a new archive file on u23 and use the merged index to write out the merged, ordered set of packets. This involves a sequence of seek(sourcefile), read(srcfile, packet), write(destfile, packet) operations.
 - c. mv the old archive file to archfilename.old, mv the new archive file to archfilename, and unlink archfilename.old (all on u23)
4. Retrieve "orphan" packets from the fcopy_packet table, and identify and post new, complete datagram records to the fcopy_datagram table.
5. Post apid, timespan, and quality information records to the fcopy_rawarchive table.
6. unlink the uncompressed file on glastlnx11:/tmp

Halfpipe Dispatch

1. A cron job on glastlnx11 notices that all level-0 files in the tarball are INGESTDONE, and invokes the ProcessSCI.py application with the incoming_pk of the tarball as an argument.
2. ProcessSCI.py creates a working directory for the HalfPipe on u42, and to it writes a set of XML files defining the datagram segments to be decoded and merged by the halfPipe.
3. ProcessSCI.py calls createStream for the halfPipe.

Halfpipe processing.

1. HalfPipe launches one substream for each datagram segment. These substreams read packet data from the rawarchive on u23, and write .evt files and corresponding .idx files to the working directory on u42.
2. after decoding, the .idx files are segregated by acquisition and merged, then used to write merged, chunked .evt files for each acquisition to the working directory on u42.
3. After merging/chunking, the magic-7 data is retrieved from the raw archive and written to a txt file in the working directory on u42. Acquisition-summary data is posted to the database, retired runs are noted to .txt file in the working directory on u42.
4. All files forming inputs to L1Proc are copied to the AFS staging directory
5. createStream is done for L1Proc.
6. Any segment evt/idx files that are not needed for future downlinks are cleaned up, as are the segregated and merged idx files for this downlink.