Rawspec

Testing Plan

Richard Elkins
2021-11-03 (v3)

https://github.com/texadactyl/rawspec_test

Rawspec Current State

- First stage in multiple production work flows that generate science products for downstream processing and analysis.
- Rawspec github repository: https://github.com/UCBerkeleySETI/rawspec
- 8 open Pull Requests as of 2021-11-03:
  - Some are drafts.
  - All are proposed feature enhancements.
  - Unknown side effects (if any) on the current source base.

√ Testing Development Plan

- Testing scope defined:
  - File-to-file only (no IP streaming).
  - 32-bit, 1 polarization input to blimpy/turbo_seti.
- Identify the .raw files to be used as testing input with assistance from Matt Lebofsky.
- Generate baseline comparators associated with each 0000.raw:
  - Generate turbo_seti top hit table: 0000.raw → .fil → .h5 → .dat → .tbl_dat
  - Generate Filterbank header field table: .h5 → .tbl_hdr
  - Discard .fil, .h5, .dat, and .log artifacts.
  - Remaining baseline file types for each test case:
    - .raw :: input file
    - .tbl_hdr :: a table containing the header field (key, value) pairs
    - .tbl_dat :: a table containing selected turbo_seti top hit records
- The testing baseline environment is now complete and ready for use.

√ Testing High-level Design

- For each test case (0000.raw file), follow this process:
  - Generate turbo_seti top hit table:

    0000.raw → .h5 → .dat → .tbl_dat
  - Generate Filterbank header information:

    .h5 → .tbl_hdr
  - Discard intermediate files (.fil, .h5, .dat, and .log)
  - Compare .tbl_hdr file to its counterpart in the baseline.
  - Compare .tbl_data file to its counterpart in the baseline.
  - Report success and failure instances.

√ Testing Software Development

- Technical Approach aka Detailed Design
- Bash & Python Code
- Installation Testing
- Functional Unit Testing
- End-to-end testing

√ Pull Request Testing Prerequisites

- Login to blpc0.
- If the rawspec_test repository has not yet been installed,

  `git clone https://github.com/texadactyl/rawspec_test`.
- Make sure that blimpy and turbo_seti are installed under $HOME:

  `pip -U --user blimpy`

  `pip -U --user turbo_seti`
- Not dependent on a particular PR.

√ Pull Request Testing Procedures

- Evaluate the PR. Is this a suitable candidate for merging (Y/N) ?
- Testing execution:
  - Login to blpc0.
  - Go to $HOME/rawspec_test/exec
  - Edit bash script xprep.sh to supply URL and BRANCH values for the PR being evaluated.
  - Run bash script xprep.sh to set up the testing trial directory.
  - Run bash script xtest.sh:
    - Generate trial results
    - Compare trial results to their counterparts in the baseline results.

Next Steps

- Evaluate the 8 outstanding PRs.
- Fork rawspec for integrating the FBH5 amendments.