

Part I: Test bed - Basic tests

1. TriDAS Startup: from OFF to running.

Elements: *all* (DBI, FCMSimu, TSC, TSV, HM, TCPU, EM, Monitoring, TriDAS-GUI)

a) Coherence wrt expected:

a.1) Runsetup vs. Datacard

a.2) Dataformats (reproduction of the exponential law of one PMT subsequent hit-time differences in FCMSimu, HM, TCPU and EM-minimum-bias)

a.2) Basic monitored observables: incoming-outgoing throughputs from servers, hit-rates, events selected, size of written data, queues occupancies.

2. TriDAS stability and failover.

Elements: *all* (FCMSimu, DBI, TSC, TSV, HM, TCPU, EM, Monitoring, TriDAS-GUI)

a) Missing items at startup (wrt expected):

a.1) not all Floors

a.2) not all HMs

a.3) not all TCPUs

a.4) DBI connection during configuration

a.5) TSV

a.6) EM

a.7) TSC

b) transient loss of items during the Run:

b.1) some Floor

b.2) some HM

b.3) some TCPU

b.4) DBI connection

b.5) TSV

b.6) EM

b.7) TSC

c) High or variable/not homogeneous throughputs (stress test)

c.1) stability vs. n. of Floors at 50 kHz of hit-rate

c.2) stability vs. homogeneous ramping of the hit-rate/Floors (defining a limit of stability vs. n. of Floors vs. hit-rate)

c.3) stability vs. disomogeneous hit-rate/Floors

(how is the pain for implementing random/controlled variability of hit-rate? Small, with script which stop/config/go)

c.4) test when big delays are introduced among hits ($O(10\text{ s})$).

Part II: Test bed - Triggers and written data

3. Triggers.

Elements: DBI, TCPU, EM, Monitoring

a) Runsetup expectation vs. effective loading:

- a.1) LI and plugins parameter settings
- a.2) n. of loaded plugins

c) Plugins failover:

- c.1) plugin not present
- c.2) crashing plugin
- c.3) stalled plugin

b) comparison of LIs with teoretical predictions:

- b.1) LI trigger rates
- b.2) dimension of the selected events

d) Monitoring:

- d.1) LI monitoring
- d.2) L2 monitoring
- d.3) Event rates monitoring

4. Written data.

Elements: DBI, EM, Monitoring

a) Communication with the DBI

- a.1) final release of a post-trigger file (ready to be sent to storage)
- a.2) summary of the Run
- a.3) possible logging (?)

b) Integrity of the data

- b.1) readout of the post-trigger file (=> and templates for reading)
- b.2) check over minimum bias for poissonian distributions
- b.3) test for event-rate congruity
- b.4) Summary analysis: livetime, n. of events, possible faults (missing sectors or timeslices)

Part III: Test bench - With the FCMServer

5. Fundamental tests.

- a. Dataformats compatibility
- b. Variable throughputs tests (for studying the stability of the connection w.r.t. the FCMServer)
- c. temporary losses of PMTs/Floors.

Then, the idea is to repeat the most relevant tests of Part I and Part II.

For the trigger tests, of course, we must consider how the data are “pumped in” via the Pulsers/Function generators, and the limited number of available FEMs.

Part IV: Test bed/bench - Continuous test

Leaving the system always running in the Bologna’s Farm, with the monitoring setup we expect to install in the Shore station.

Work in progress....