

How to test the online alignment procedure on a single node

Due to the necessity of having DIM, the online alignment tests can only run in the online environment, e.g. on a plus or HLT node.

Setup the build environment

Since subversion is only available on plus nodes, perform these steps there. They are only needed once.

```
lb-dev AlignmentOnline v10r5
cd AlignmentOnlineDev_v10r5
getpack PRConfig head
cd PRConfig
ln -s . v1r999
```

If you want to get the most up-to-date configuration, please also checkout the package with modifications in the LHCb tag collector for the Alignment and AlignmentOnline projects. For example:

```
getpack Alignment/TAlignment head
getpack AlignmentOnline/AlignOnline head
```

Build the packages:

```
make configure
make -j 5 install
```

Make sure to `svn update` and recompile before any tests, as improvements are still being made.

Find a suitable machine to run on

Since plus nodes are quite limited in terms of CPU capacity, one of the HLT performance testing nodes could be used. The farm itself should not be used to avoid interference with datataking (also after the tests have completed).

The HLT performance testing nodes are:

- hltperf-action-x5650
- hltperf-dell-x5650
- hltperf-asus-amd6272

Beware that you need to load the LHCb environment on these machines:

```
source /cvmfs/lhcb.cern.ch/group_login.sh
```

Before running your test is good to check that there are no processes hanging from previous failed jobs.

```
ps -Alf
```

And look for jobs of the kind **GaudiOnlineExe.exe**. If any is present check that the owner is not working on the machine and ask to remove them.

Make sure the data is available

The online alignment test will start a number of worker processes that will read data from file(s). These file(s) must be available on the node that is used, so either copy the files to the node or put them in a folder on NFS. A small sample of test data files is located on hlte0902 at /localdisk/hlt1/test_align.

The worker processes can either each read a different file, or all read the same file. If they all read the same file, the same events will be "sent" to the iterator multiple times, which might not be desirable.

When separate files are required, make sure they are all from the same run and a number of files at least as large as the number of worker processes is present in the data directory.

A script to split a raw file into a larger number of chunks that all contain N different events is attached to the page.

Setup the running environment and run the test

Once you have found a node, log in to it, setup the running environment and run.

```
cd $User_release_area/AlignmentOnlineDev_v10r5
./run bash
export PYTHONPATH=/home/raaij/pydim/lib/python2.7/site-packages:/scratch/jenkins/benchmark/python
export HLTTCROOT=/group/hlt/sattelite/MooreOnlinePit_v24r0p1/TCK/HltTCK/
cd $User_release_area/AlignmentOnlineDev_v10r5/PRConfig/scripts
python AlignOnlineTest.py --numa --nodes=2 --workers=10 --task-log=log/align.log --directory=/loc
```

For an explanation of the arguments to AlignOnlineTest.py, run

```
python AlignOnlineTest.py --help
```

The final argument should be the name of a python module in PRConfig/python/AlignmentTests/TestModules that contains the required information to start the iterator and analyzer processes.

Looking at the (intermediate) results

A new temporary directory is created for each test and printed in the log that goes to the test log file and stdout. This directory should contain all intermediate files, results and conditions.

Cleaning up

If the test completes successfully, all processes will be terminated, except the LogViewer, which can be exited using Ctrl+C. If the test is interrupted, not all processes might be terminated, clean them up using:

```
killall GaudiOnlineExe.exe
```

The temporary directory is not removed and should be manually removed once the files it contains are no longer needed for debugging or analysis.

Developing new tests

To develop a new test, two tasks have to be implemented, an iterator and an analyzer. Both should be contained in some python module and should be startable with a python call to a function.

Both tasks must use the correct components to ensure that the state transitions are correct. The code for the tracker alignment tasks is located in AlignmentOnline /AlignOnline - both python and C++ - and can be used as a template.

Once the iterator and analyzer tasks are ready, a python module needs to be written to run the tests. The module AlignTrackerTest.py located in PRConfig/python/AlignmentTests implements this for the tracker alignment and can be used as an example.

Requirements for iterator and analyzer configuration modules

To create a new test, create a new module in AlignmentTets /TestModules that contains two classes, Iterator and Analyzer.

The Iterator class must be constructable without arguments and implement the *call* function that takes no arguments. This function should return the one-line python statement required to start the iterator process.

The Analyzer class must be constructable with a single argument: the list of available input files. It must also implement the *call* function that takes a single argument: the number of the worker process. This function should return the one-line python statement required to start the analyzer process.

Look at PRConfig/python/AlignmentTests/TestModules/TrackerAlignment.py for an example.

Example pure python based Iterator and Analyzer

A new package named AlignmentOnline /PyAlignOnline has been added to SVN, which contains an example implementation of a purely python based Iterator and Analyzer. To test them, use **PythonAlignment** as the last argument to the call to AlignOnlineTest.py. Note that the PythonAlignment is a skeleton example and thus does not analyze the data.

-- RoelAaij - 2015-01-12

This topic: LHCb > HowToRunAlignmentTestOnline

Topic revision: r23 - 2015-08-13 - MaurizioMartinelli



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback