Table of Contents

Chapter 3.6: Analysis of the collision data

Conceptual overview

Running over large amounts of data can be rather tricky, but good practices can make your life a lot easier. A judicious usage of the computing resources and software tools at your disposal can make your analysis run a lot more smoothly and efficiently, while at the same time causing less of a strain on the limited computing resources.

Before we get started on this WorkBook, it is useful to overview how data are acquired and distributed, in the context of accessing it for your analysis.

The data are collected by the detector and processed through the HLT. From there, the HLT paths are designated to live inside a specific "Primary dataset" (PD). This is the "quantum" of the computing infrastructure. PD's are distributed in entirety to T1's and T2's, so accessing them is the primary mode that you will be using to access the data. The Primary Dataset Working Group (PDWG) is a good resource for you to keep up to speed with the PD's and their deployment.

There are quite a lot of random triggers that occur when the detector is not taking data, and so to account for this, the best practice is to only run on luminosity sections where the detector was "on". This webpage is constantly updated with "good run lists" in a JSON format that correspond to the "DCS bit" being on, and the detector taking data. By using this "good run list" in your CRAB jobs, you will alleviate strain on the resources and run only on the data that is interesting for you for physics analysis.

There are also many triggers within a given primary dataset. It is fast and efficient to request a single trigger (or group of triggers) at the beginning of your path, such that the rest of the sequence will only process if that trigger fired. This also leads to an alleviated strain on the computing resources, and lets your jobs finish much more quickly.

Much of the machine background that comes from, for instance, beam+gas interactions, can be removed by prescriptions from various DPG's. This wisdom is best followed, and hence is provided as a standard cleaning recipe below.

The following recipes will help you to get started with performing effective analysis of the collision data while minimizing impact on the computing environment.

Recipes to get started

In terms of software the user should always follow the instructions from WorkBookWhichRelease.

Physics Data And Monte Carlo Validation (PdmV) group maintains some analysis recipe pages here. Especiale for run-II, next pages 2015, 2016, 2017 and 2018 collects useful information to be aware of when performing analysis or producing ntuples for analysis. Guidelines are collected on which release, detector conditions, and datasets. They also keep track of special filters or tools useful to remove atypical / problematic events

There are several recipes for you to use to clean up the event sample as recommended by the PVT group. For 2010 and 2011 check here.

Selection of good runs

The PVT group maintains centralized good run lists in two different formats here. The files are stored as either JSON format or directly as a CMSSW configuration snippet.

The JSON format files should be used in conjunction with CRAB. Either the JSON or CMSSW formats should be used interactively in FWLite or the full CMSSW framework.

- Using the JSON format with CRAB3 is described here.
- Using the JSON format with CMSSW or FWLite is described in these links:
  - Using JSON for selecting good lumi sections.
  - Detailed description of usage in various contexts.

JSON file updates are announced on this HyperNews forum link.There you can found discussions regarding physics validation of MC and Data production for physics analyses, including good/bad run list based on DQM certification tools.
If you want to contact the experts the email gateway for this forum is:
hn-cms-physics-validation@cernNOSPAMPLEASE.ch

Usage of computing resources

The best policy for users to use CRAB3 in order to process collision data is :

- Use the good run lists (in the JSON format) in CRAB3 as described here.
  - The good run lists are available here.
- From there, publish the dataset if you intend to use grid resources to access it later. Instructions for CRAB3 publication can be found here.
- The final good run lists should be applied at the analysis level. The latest good run lists are available here.

Trigger Selection

Oftentimes, it is extremely useful to apply your trigger selection in your skim or PAT-tuple creation step directly. This reduces the load on the computing and gives you a smaller output. To do so, as an illustration, here is how to select `HLT_Dimuon25_Jpsi` :

```
process.triggerSelection = cms.EDFilter("TriggerResultsFilter",
                                        triggerConditions = cms.vstring('HLT_Dimuon25_Jpsi_v*'),
                                        hltResults = cms.InputTag( "TriggerResults", "", "HLT" ),
                                        l1tResults = cms.InputTag( "" ),
                                        throw = cms.bool(False)
                                        )

...


process.mySequence = cms.Sequence(
    process.triggerSelection*
    process.myOtherStuff
)
```

where `myOtherStuff` is whatever other modules you want to run.

More information on trigger access in analysis can be found at WorkBookHLTTutorial and WorkBookMiniAOD2017#Trigger.

Analysis of the processed data

There are several choices for the user to analyze collision data. There are several examples to help get you started: example in FWLite , TrackAnalysis and MuonAnalysis.

Luminosity information

Luminosity should be calculated with the official **brilcalc** tool, recommended tool for both Run 1 and Run 2 data. For more information, please see the Lumi POG page and the official brilcalc documentation.

About Datasets

The data are collected into "primary datasets", "secondary datasets", and "central skims" for distribution to the computing resources. The definitions are such to keep roughly equal rates for each PD.

The PDWG is a group that defines and monitors these datasets. The PDWG TWiki describes details in how this process is done.

Particulary in runn-II, for producing ntuples for analysis the data are collected into different formats, such as, "AOD", "MiniAOD" and "NanoAOD" . The definitions are such that an attempt is made to maintain the necessary information according to the needs of each analysis.

Overlapping run ranges.

In general, there are many different primary datasets and/or run ranges that you will have to process for your analysis. As things evolve, older data is re-reconstructed, and primary datasets are split into smaller pieces. Because of the fact that oftentimes runs can appear in different datasets in re-reco's, it is often necessary to **first** define a run range for your dataset while running your grid job. This will ease your own accounting later on in the analysis chain.

To do this, it is often advantageous to split up the good run lists into exclusive run ranges, and then pass the split sections to the various grid jobs you are running for the various primary or secondary datasets. See the guide on good lumi sections with details of how to do this.

For instance, given two run ranges "1-20" and "21-50", you could split up your good run list in the JSON format as:

```
filterJSON.py --min 1 --max 20 old.json --output first.json
filterJSON.py --min 21 --max 50 old.json --output second.json
```

This will create disjoint run ranges for all of your datasets, simplifying your accounting.

Review status

Reviewer/Editor and Date (copy from screen)	Comments
JhovannyMejia - 28 Aug 2018	Update to RunII
SalvatoreRoccoRappoccio - 28 Sep 2010	Author

-- SalvatoreRoccoRappoccio - 28-Sep-2010

This topic: CMSPublic > WorkBookCollisionsDataAnalysis
Topic revision: r15 - 2018-09-03 - JhovannyMejia