Measurement of offline efficiencies

Top Trigger Tutorial
October 2nd 2013

Kelly Beernaert

Outline

- Introduction
- Lepton triggers
  - Tag and Probe
  - Uncertainties
- Hadronic triggers
- Presenting the efficiencies
- New trigger strategies
- Conclusions

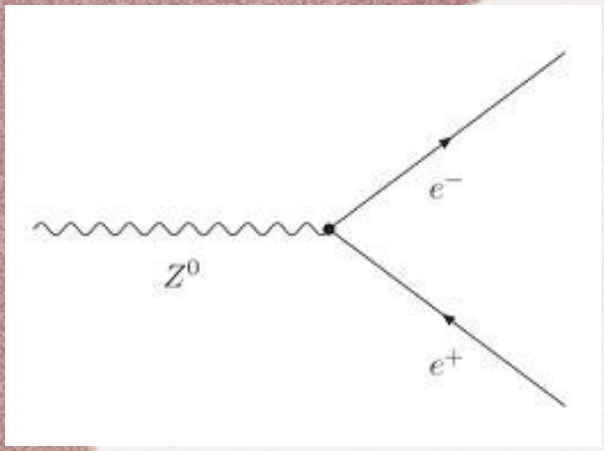
Introduction

- Trigger menu applied in MC is often different than the online trigger menu.
- Correct the MC by applying Scale Factors or absolute efficiency corrections.
- SF are always determined for a specific event selection and for a specific trigger in Data and MC.
- Dependencies on kinematic variables need to be taken into account, to obtain an accurate correction.
- In the lepton+jets cross trigger paths, we make the assumption that the lepton and hadron leg are independent, since we remove one lepton from the jet collection.

Lepton triggers

- Single lepton triggers are usually measured using a Tag and Probe method.
- Tag and Probe method offers many benefits:
  - Well established
  - Unbiased results
  - Low background
- However, you restrict your measurement to the kinematic range of the Z-decay which is not the same as a Ttbar decay.
  → high pt-bins are limited in statistics

Tag and Probe



- Use leptonic decay of the Z to measure electron or muon efficiencies.
- ID and isolation requirements on the Tag lepton are very tight, ensuring a good lepton, usually matched to a trigger object.
- Probe lepton collection consists of the other lepton in the event, satisfying looser ID and forming a Z decay pair with the Tag.
- This construction allows you to have a pure lepton collection on which efficiencies can be studied.

Tag and Probe

- For trigger efficiencies, your Probe passes the Top reference selection (or any selection your particular analysis requires).
- Efficiency is then $\epsilon = \frac{N_{Probe\ passing trigger\ bit}}{N_{Probe}}$
- As long as Tag selection is tighter than the Probe, your efficiency is independent of the Tag selection.
- Tag selection can be made tighter to reduce background levels.

Tag and Probe: obtaining Tag and Probe tree

- CMSSW package: TagProbeFitTreeProducer
- Allows you to make a Tag collection, a Probe collection and combine these in Tag/Probe pairs

```
process.tagMuons = cms.EDFilter("PATMuonRefSelector",
  src = cms.InputTag("patMuons"),
  cut = cms.string("isGlobalMuon && muonID('GlobalMuonPromptTight') && !triggerObjectMatchesByFilter('hltSingleMu3L3Filtered3').empty()"),
)
```

or access the trigger objects manually (see slide 18)

Tag and Probe: obtaining Tag and Probe tree

- Tag collection and Probe collection are combined into pairs by *CandViewShallowCloneCombiner*
- ```
  process.TagProbe = cms.EDProducer("CandViewShallowCloneCombiner",
                                    decay = cms.string("Tag Probe"),
                                    checkCharge = cms.bool(False),
                                    cut   = cms.string("50 < mass < 120"),
                                    )
  ```

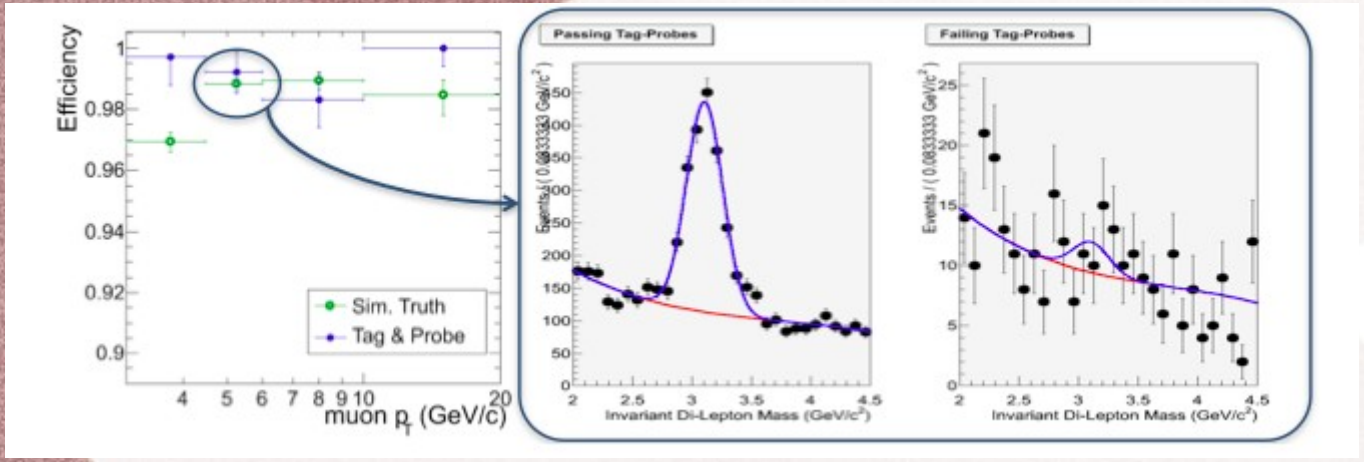
  Charge conjugate states can be imposed, mass window on invariant mass of pair is imposed
- ```
  recoCommonStuff = cms.PSet(
         variables = cms.PSet(
             eta = cms.string("eta()"),
                    pt  = cms.string("pt()"),
                    phi  = cms.string("phi()"),
                    et  = cms.string("et()"),
                    e  = cms.string("energy()"),
                    p  = cms.string("p()"),
                    px  = cms.string("px()"),
                    py  = cms.string("py()"),
                    pz  = cms.string("pz()"),
                    theta  = cms.string("theta()"),
                ),
            ignoreExceptions =  cms.bool (True),
            fillTagTree      =  cms.bool (True),
        )
  ```

Tag and Probe: obtaining Tag and Probe tree

```
process.TrigEff = cms.EDAnalyzer("TagProbeFitTreeProducer",
                                 recoCommonStuff,
                                 isMC = cms.bool(False),
                                 tagProbePairs = cms.InputTag("TagProbe"),
                                 arbitration   = cms.string("None"),
                                 flags = cms.PSet(
                                    passing = cms.InputTag("ProbePassing"),
                                 ),
                                 allProbes     = cms.InputTag("Probe")
                                 )
```

- With the Tag and Probe tree available, the efficiency can be extracted.
- There are various options:
  - Fitting: most precise, most work
  - Counting: adequate with low background contamination
  - Same sign/opposite sign: low background contamination, measure charge misid

Tag and Probe: Fitting the efficiency



- A binned fit to the Z-peak in the invariant mass spectrum to extract the number of passing probes and all probes
- Signal fit function:
  - Fit a template shape from MC
  - A parametrised PDF (eg. Breit-Wigner convoluted with a Crystal Ball)
- Background fit function: exponential multiplied by an error function

Tag and Probe: Uncertainties

- When using Counting: check difference with fitting method
- Check the dependence on the Tag selection
- Vary the fit functions
- Effect of residual background contamination

See AN2012_116

Hadron triggers

- For hadronic triggers Tag and Probe is not possible, fall back to a Cut and Count approach.
- Principle is simple: take all your events which pass your event selection and count which fraction passes the full trigger path.
- However, some thought and consideration is needed to obtain an unbiased measurement.

Hadronic triggers: strategy

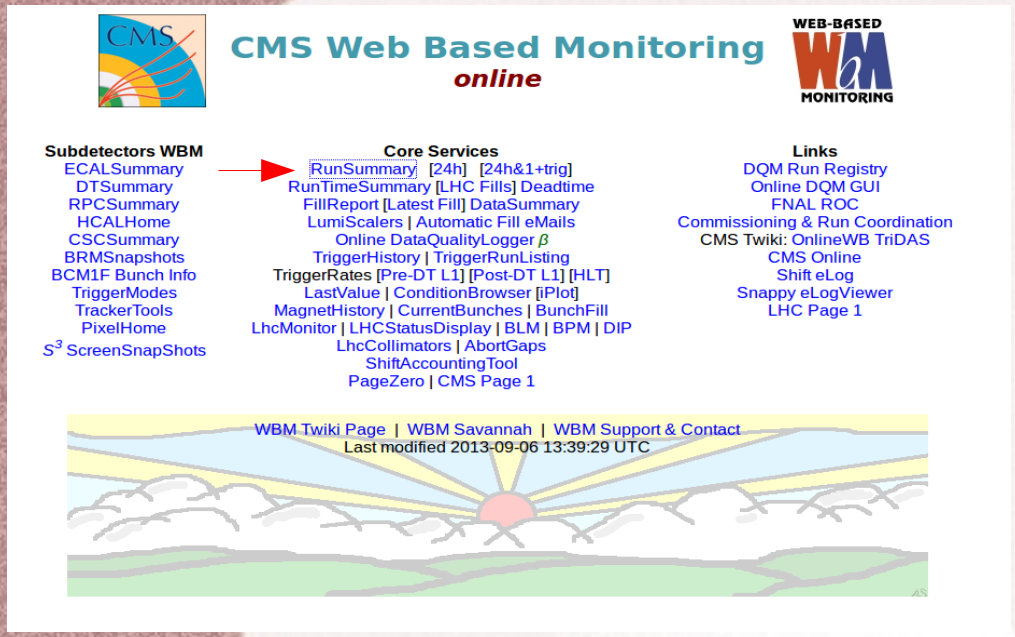
- When measuring the hadron leg in a cross trigger (which comes last in the trigger path), the leptonic leg needs to be fully efficient.
- Therefore our selected events consist of events passing the top reference selection + *passing a single lepton trigger* + passing the leptonic leg of the cross trigger.
- Making the efficiency:

$$\epsilon = \frac{selected\ events\ passing\ full\ cross\ trigger}{selected\ events}$$

- The requirement for the single lepton trigger is necessary since single lepton triggers and LepHad triggers share a Primary Dataset now.

Hadronic triggers: how to get your trigger information?

- https://cmswbm.web.cern.ch/cmswbm/

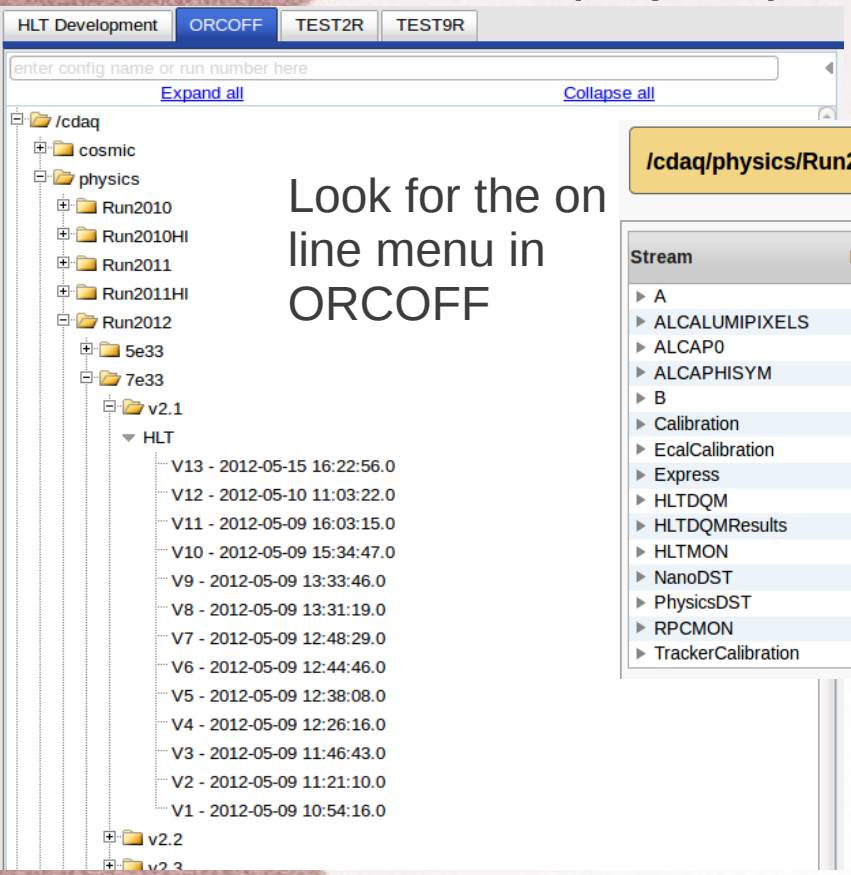


BField	3.799 Tesla
InitialPrescaleIndex	5
Tier0Transfer	true
TriggerMode	l1_hlt_collisions_2012/v91
TTC MI Key	beam1-manual-20110413
L1 Key	TSC_20120511_002881_collisions_BASE
HLT Key	/cdaq/physics/Run2012/7e33/v2.1/HLT/V12
HLT Version	CMSSW_5_2_5_HLT5
L1 Rate	39841.100 Hz
HLT Rate Stream A	166.864 Hz
L1 Triggers	161389224
HLT Triggers Stream A	672913
HLT Size Stream A	151.811 Gb
HLT Rate Stream A	0.038 Gb/sec
LHC Fill	2621
LHC Energy	4000 GeV
Initial Lumi	1,809.8004150 $\times 10^{30} cm^{-2} sec^{-1}$
Ending Lumi	1,713.4112549 $\times 10^{30} cm^{-2} sec^{-1}$
Run Lumi	7.007508 pb^{-1}
Run Live Lumi	6.896945 pb^{-1}
Efficiency By Lumi	98.422%

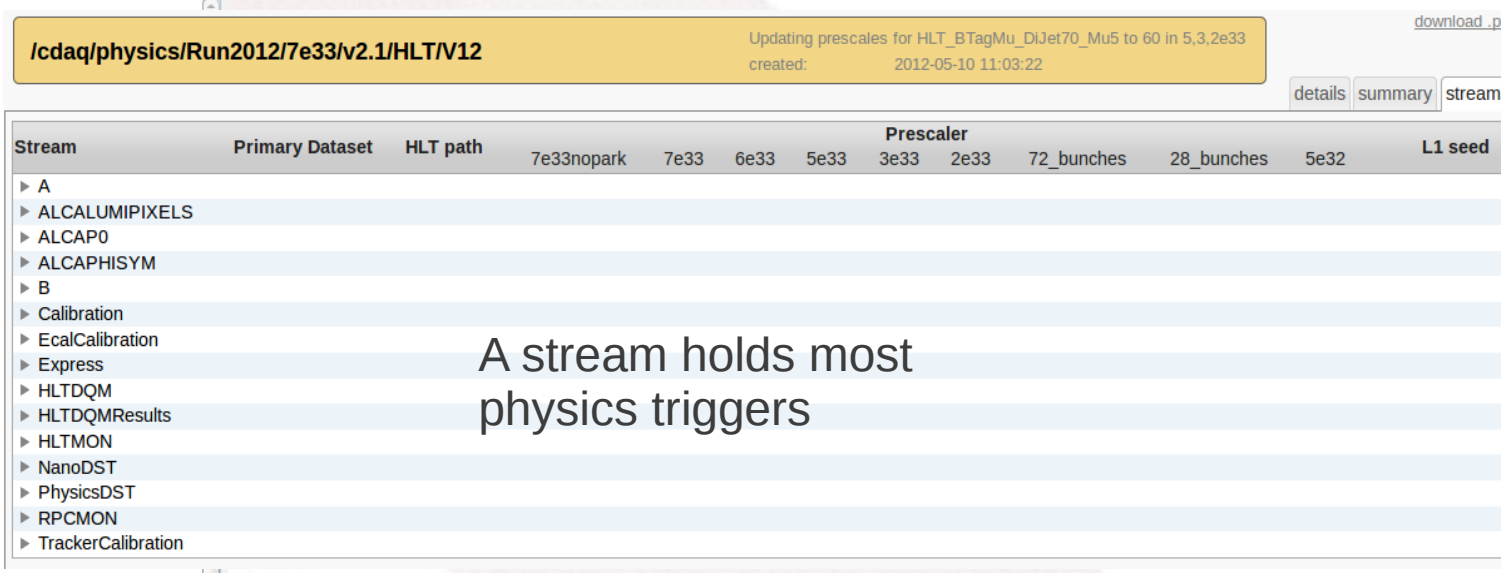
- RunSummary gives you the HLT menu used for a particular run

Hadronic triggers: how to get your trigger information?

- http://j2eeps.cern.ch/cms-project-confdb-hltdev/browser/



Look for the on line menu in ORCOFF



A stream holds most physics triggers

Hadronic triggers: how to get your trigger information?

- Select your primary dataset

Primary Dataset	HLT path	Prescaler 7e33nopark	7e33	6e33	5e33	3e33	2e33	72_bunches	28_bunches	5e
▸ BJetPlusX										
▸ BTag										
▸ Commissioning										
▸ Cosmics										
▸ DoubleElectron										
▸ DoubleMu										
▸ DoubleMuParked										
▸ DoublePhoton										
▸ DoublePhotonHighPt										
▸ ElectronHad										
▸ FEDMonitor										
▸ HTMHT										
▸ HTMHTParked										
▸ HcalHPDNoise										
▸ HcalNZS										
▸ JetHT										
▸ JetMon										
▸ LogMonitor										
▸ MET										
▸ MinimumBias										
▸ MuEG										
▸ MuHad										
▸ MuOnia										
▸ MuOniaParked										
▸ MultiJet										
▸ MultiJet1Parked										
▸ NoBPTX										
▸ PhotonHad										
▸ SingleElectron										
▾ SingleMu										
	HLT_IsoMu17_eta2p1_CentralPFNoPUJet30_BTagIPIter_v1	1	1	1	1	1	1	1	1	
	HLT_IsoMu17_eta2p1_CentralPFNoPUJet30_v1	20	20	20	20	20	20	1	1	
	HLT_IsoMu17_eta2p1_DiCentralPFNoPUJet30_v1	5	5	7	5	5	5	1	1	
	HLT_IsoMu17_eta2p1_TriCentralPFNoPUJet30_v1	1	1	1	1	1	1	1	1	
	HLT_IsoMu17_eta2p1_TriCentralPFNoPUJet50_40_30_v1	1	1	1	1	1	1	1	1	
	HLT_IsoMu20_eta2p1_CentralPFJet80_v6	1	1	1	1	1	1	1	1	
	HLT_IsoMu20_eta2p1_WCandPt80_v6	1	1	1	1	1	1	1	1	

unprescaled trigger + scrolling to right shows L1 Trigger Bit

Hadronic triggers: how to get your trigger information?

- For the HLT_IsoMu17_eta2p1_TriCentralPFNoPUJet30

- Shows all the modules in a trigger, but not all of these write out trigger objects

```
module hltL1sMu14Eta2p1 = HLTLevel1GTSeed {
  bool saveTags = true
  string L1SeedsLogicalExpression = "L1_SingleMu14er"
  InputTag L1MuonCollectionTag = hltL1extraParticles
  bool L1UseL1TriggerObjectMaps = true
  bool L1UseAliasesForSeeding = true
  InputTag L1GtReadoutRecordTag = hltGtDigis
  InputTag L1CollectionsTag = hltL1extraParticles
  int32 L1NrBxInEvent = 3
  InputTag L1GtObjectMapTag = hltL1GtObjectMap
  bool L1TechTriggerSeeding = false
}
```

- Module writing out trigger objects

Hadronic triggers: accessing the trigger objects

```
edm::Handle<edm::TriggerResults> HLTR;
iEvent.getByLabel("HLTriggerResults",HLTR);

edm::Handle<trigger::TriggerEvent> aodTriggerEvent
iEvent.getByLabel("HLTAodSummary", aodTriggerEvent); //global var

std::string module_name = "hltL1sMu14Eta2p1"; //example of a L1 module
edm::InputTag filterTag = edm::InputTag(module_name, "", "HLT");

int filterIndex = aodTriggerEvent->filterIndex(filterTag);
const trigger::Keys &keys = aodTriggerEvent->filterKeys( filterIndex );

const trigger::TriggerObjectCollection objects = aodTriggerEvent->getObjects();

trigger::TriggerObject foundObject = objects[keys[0] ]; //first trigger object for that module
```

This gives you back the online object. Can be matched with ΔR matching to offline object (ΔR < 0.1).

Hadronic triggers

- HLT_IsoMu17_eta2p1_TriCentralPFNoPUJet30
- Check if event (SingleMu PD) passes the single muon trigger (IsoMu24)
- Check if >= 1 trigger object for IsoMu17_eta2p1 module

  → events passing these requirements constitute your “probe event” pool, store the kinematics of jets in your TTree
- Check if >= 3 trigger objects for TriCentralPFNoPUJet30 module

  → store this decision in a leaf of your tree

Hadronic triggers

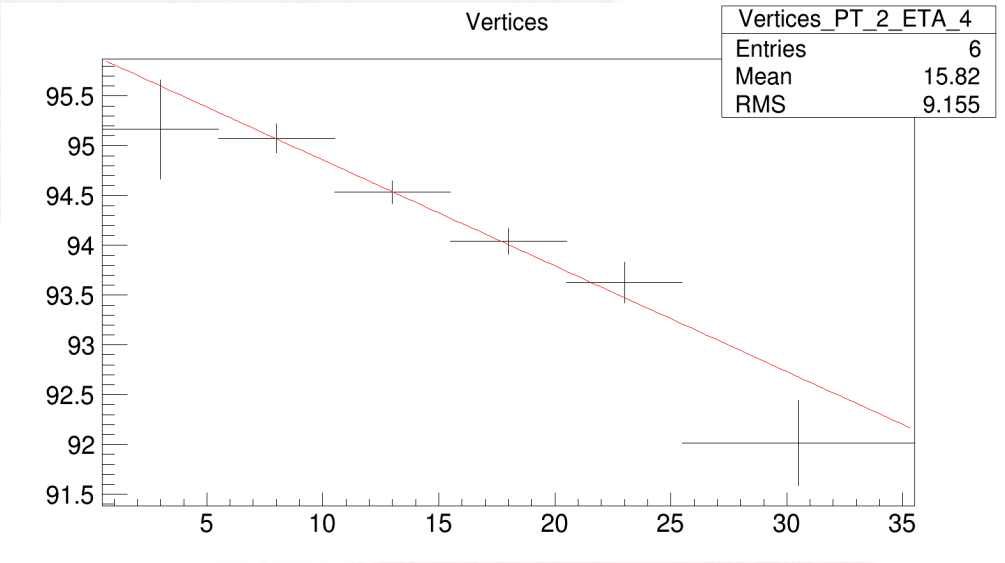
- Once you have your tree, you can easily plot your passing events and all events as a function of any variable you stored and use TgraphAsymmErrors to get the efficiencies.
- For an N-jet trigger, the efficiencies are shown as a function of the N+1st leading offline jet
  → to approximation independent of the trigger objects
- Check the effect of the JES on the trigger efficiency
- Check the dependence on the 1st, 2nd, … Nth offline jet pt, to decide where to put the offline jet pt-cuts

Presenting the efficiencies

- Check the dependence of the trigger efficiency on the kinematics of your offline object:
  - pt: turn-on curve, can help you decide on offline cuts
  - eta: large dependence for electrons, not for jets
  - phi: should be flat
- Event variables:
  - pile-up
  - jet multiplicity

Presenting the efficiencies

- Offer parametrised efficiencies if necessary
- Try to fit the dependence if possible (eg. pile up dependence)
- Small fluctuations can be covered by a systematic uncertainty.



New trigger strategies

- New strategies for measuring trigger efficiencies will become necessary as triggers become more convoluted
- When designing a trigger, a clear strategy on how to measure the efficiency is a necessity. **Plan wisely!**

New trigger strategies

Some considerations to make:

- Single lepton triggers can always be measured with Tag and Probe?
  → not necessarily, with increasing pt-thresholds, limiting yourself to the pt-spectrum of Z decay products may cause statistics limits (pt > 100)
- Cross trigger legs are independent of each other?
  → must be checked by comparing the efficiencies of the lepton leg with the corresponding single lepton trigger (but with much higher pt-threshold), this single lepton trigger needs to be included in the trigger menu as monitoring path

New trigger strategies

- Some considerations to make:
  - There will always be a certain jet-electron ambiguity
    → measure hadron trigger efficiencies in the muon+jets path if a cross trigger is considered
  - Trigger modules are written out consecutively
    → when designing a new type of cross trigger: plan the efficiency strategy beforehand, since the ordering of modules may not be trivial

Conclusions

- Plan the efficiency measurements, along with the design of the trigger!
  - Monitoring paths
  - Module ordering
- Pre-set strategies (Tag and Probe), may no longer be the best option
  → be prepared to think of new strategies