Graphical Tools for Detector Operations and Validation: Should we be involved?

Sam Harper (RAL)

SwiftHEP Meeting

March 27th 2024

Introduction

- was CMS trigger coordinator, during my term oversaw
  - Phase II HLT TDR with first realistic compute estimates
  - CMS's transition to GPUs
  - Run3 prep
- During my term, the biggest problem I identified was not making our code faster
  - this is under control, its challenging but there is a well-thought out plan which SWIFTHEP is a key part of
- It was the lack of tools to operate and run CMS
  - there is laughably little effort on this and the skill set to do so does not overlap with typical skill set we have available

Introduction (II)

More succinctly

- I'm not worried about CMS not being able to meet the computing of the HL-LHC
  - GPU efforts are advanced, lots of code development on going, this is going to converge
- I am worried about not able to turn on the detector or configure it due to broken tools to do so
  - CMS came worryingly close to not being able to change trigger keys when the CERN SSO upgrade happened
    - simply because almost nobody knows how our webbased tooling worked

What I am talking about

- I mean tools like
  - displaying DQM histograms
  - developing trigger paths
  - configuring detector settings
  - displaying detector information
  - presenting information to analyses

typically all need some sort of GUI

- I don’t mean analysis tools
  - RDataframe, coffea, other popular tools
  - already well served by others and in general this is in good shape

Efficient Tools are key to unlocking Physics Potential

- physicists try to do too much
- goal for maximising physics is allow them to achieve more
- efficient tools are key here
  - CMS’s physics potential is absolutely limited by the quality of its tools
    - there are some good ones (eg OMS our online monitoring) but the majority are bad
  - trigger development barrier is too high, folks spend their time battling with the substandard tools rather than making performant triggers
- the problem (at least on CMS) is that we have very little ability to build efficient (non-analysis) tools
  - maybe its better on other experiments?

Graphical Tools

- trend is all GUIs moving to running in the browser
  - sometimes it looks native but often can actually be a webapp running on a bundled version of chromium
    - electron is common framework
    - vscode is probably the most popular one
- GUIs in a browser means written in javascript*
  - there have been attempts with python, so far I’ve not been impressed
- physicists in general do not know javascript
  - nor do they have much experience with REST APIs and other key pardims here
- therefore its very hard to for collaborations to develop such tools as there is a very limited talent pool with the correct skill set
  - difficult to import the skill set: hard to retain full stack devs and hard to communicate the requirements to them

*IMHO any serious development should use typescript which is a superset of javascript
in this talk whenever I say javascript, I really mean typescript

Commonality Between Experiments

- many tools needed by experiments will be bespoke
  - say something unique to configurating a CMS run
- but many will be common
  - every experiment needs a way to visualise validation plots efficiently
- even the bespoke tools have common elements
  - usually its just the "business logic" or exactly what information to display
  - having a good example or libraries will help
  - eg if you want to interact with the CERN SSO can use a common package like tsgauth

Designing Tools

- a tool must provide the correct information
- a tool needs to enable the correct workflows

Typically the experts (aka physicists) know what they need do and an external programmer doesn't

There are two approaches

- teach a physicist how to write the tool
- have the physicist write a detailed specification and requirements document
  - this is harder than it sounds!

Case Studies

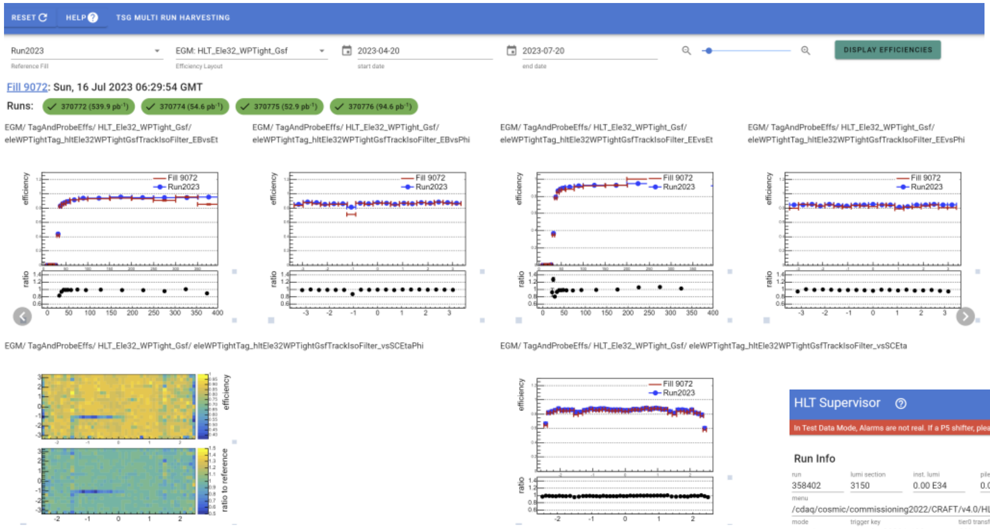
- I learnt typescript, make simple websites hosted on CERN PAAS
- I make tools which I think are most needed
- all tools are somewhat similar architecture
  - frontend : typescript + vue
    - vue is frontend framework, one of the big three (react, angular, vue)
    - still js ecosystem is fast moving, frameworks do get deprecated
  - backend : python using flask (moving to fastapi)
    - try and do as much as possible on the backend as python is very physicist friendly
  - usually some sort of db such as mongo
    - again very physicist friendly, its just json docs
    - note I'm using this to store root histograms now
- once you know how, its very easy to do this
  - the trick is knowing how
- it has been a game changer, I've been busy solving many of the things that irate me on CMS
  - and the tools work exactly how we want it to work

Case Studies: examples

examples:

- hltsupervisor: assists shift crew & experts handle trigger issues at P5
  - analyses the trigger rates and offers clear advice on what to do
  - improves operational efficiency at p5
- hltinfo: provides HLT information for analyses
  - before this, very difficult in CMS to find out basic trigger info
- multi run DQM:
  - CMS DQM plots are run based
  - this agreates key plots over a fill and displays a quick summary on triggr health intended for a busy expert
- timing measurement service
  - allows users to run the standard benchmarks to estimate CPU resources of their path with a single script
  - displays the results easily

Examples of tools



Run2023 — Reference Fill

EGM: HLT_Ele32_WPTight_Gsf — Efficiency Layout

2023-04-20 — start date

2023-07-20 — end date

Fill 9072: Sun, 16 Jul 2023 06:29:54 GMT

Runs: ✓ 370772 (539.9 pb^{-1}) ✓ 370774 (54.6 pb^{-1}) ✓ 370775 (52.9 pb^{-1}) ✓ 370776 (94.6 pb^{-1})

EGM/ TagAndProbeEffs/ HLT_Ele32_WPTight_Gsf/ eleWPTightTag_hltEle32WPTightGsfTrackIsoFilter_EBvsEt

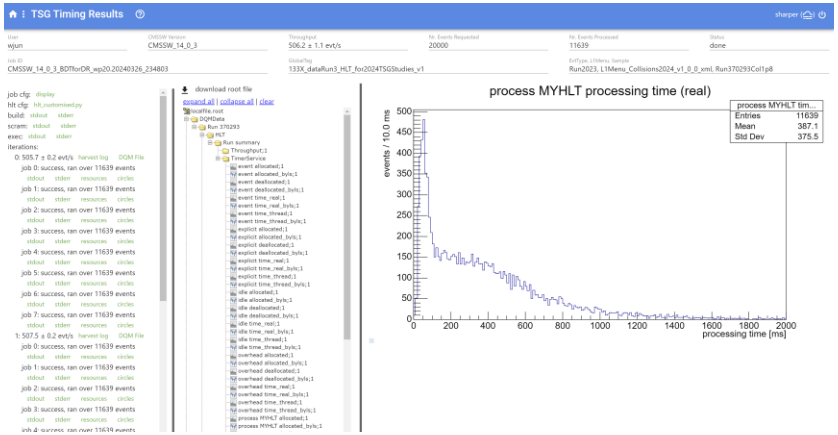
EGM/ TagAndProbeEffs/ HLT_Ele32_WPTight_Gsf/ eleWPTightTag_hltEle32WPTightGsfTrackIsoFilter_EBvsPhi

EGM/ TagAndProbeEffs/ HLT_Ele32_WPTight_Gsf/ eleWPTightTag_hltEle32WPTightGsfTrackIsoFilter_EEvsEt

EGM/ TagAndProbeEffs/ HLT_Ele32_WPTight_Gsf/ eleWPTightTag_hltEle32WPTightGsfTrackIsoFilter_EEvsPhi

EGM/ TagAndProbeEffs/ HLT_Ele32_WPTight_Gsf/ eleWPTightTag_hltEle32WPTightGsfTrackIsoFilter_vsSCEtaPhi

EGM/ TagAndProbeEffs/ HLT_Ele32_WPTight_Gsf/ eleWPTightTag_hltEle32WPTightGsfTrackIsoFilter_vsSCEta



User	CMSSW Version	Throughput	Nr. Events Requested	Nr. Events Processed	Status
wjun	CMSSW_14_0_3	506.2 ± 1.1 evt/s	20000	11639	done

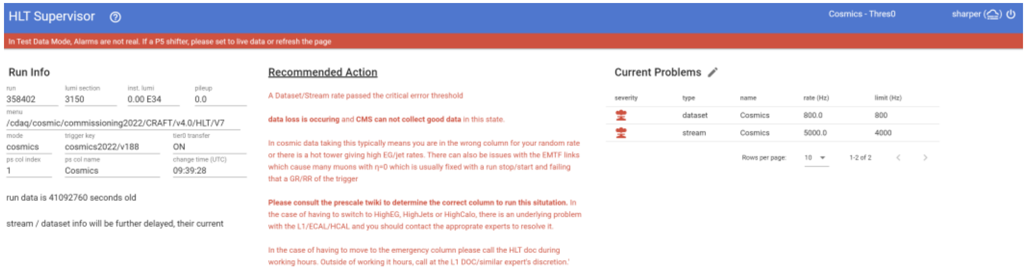
Job ID: CMSSW_14_0_3_BDTforDR_wp20.20240326_234803

GlobalTag: 133X_dataRun3_HLT_for2024TSGStudies_v1

EvtType, L1Menu, Sample: Run2023, L1Menu_Collisions2024_v1_0_0_xml, Run370293Col1p8

job cfg: display
hlt cfg: hlt_customised.py
build: stdout stderr
scram: stdout stderr
exec: stdout stderr
iterations:
0: 505.7 ± 0.2 evt/s harvest log DQM File
job 0: success, ran over 11639 events
stdout stderr resources circles
job 1: success, ran over 11639 events
stdout stderr resources circles
job 2: success, ran over 11639 events
stdout stderr resources circles
job 3: success, ran over 11639 events
stdout stderr resources circles
job 4: success, ran over 11639 events
stdout stderr resources circles
job 5: success, ran over 11639 events
stdout stderr resources circles
job 6: success, ran over 11639 events
stdout stderr resources circles
job 7: success, ran over 11639 events
stdout stderr resources circles
1: 507.5 ± 0.2 evt/s harvest log DQM File
job 0: success, ran over 11639 events
stdout stderr resources circles
job 1: success, ran over 11639 events
stdout stderr resources circles
job 2: success, ran over 11639 events
stdout stderr resources circles
job 3: success, ran over 11639 events
stdout stderr resources circles
job 4: success, ran over 11639 events

process MYHLT tim...	
Entries	11639
Mean	387.1
Std Dev	375.5



In Test Data Mode, Alarms are not real. If a P5 shifter, please set to live data or refresh the page

Run Info

run	lumi section	inst. lumi	pileup
358402	3150	0.00 E34	0.0

menu: /cdaq/cosmic/commissioning2022/CRAFT/v4.0/HLT/V7

mode	trigger key	tier0 transfer
cosmics	cosmics2022/v188	ON

ps col index	ps col name	change time (UTC)
1	Cosmics	09:39:28

run data is 41092760 seconds old

stream / dataset info will be further delayed, their current

Recommended Action

A Dataset/Stream rate passed the critical errror threshold

data loss is occuring and **CMS can not collect good data** in this state.

In cosmic data taking this typically means you are in the wrong column for your random rate or there is a hot tower giving high EG/jet rates. There can also be issues with the EMTF links which cause many muons with η=0 which is usually fixed with a run stop/start and failing that a GR/RR of the trigger

Please consult the prescale twiki to determine the correct column to run this situtation. In the case of having to switch to HighEG, HighJets or HighCalo, there is an underlying problem with the L1/ECAL/HCAL and you should contact the approprate experts to resolve it.

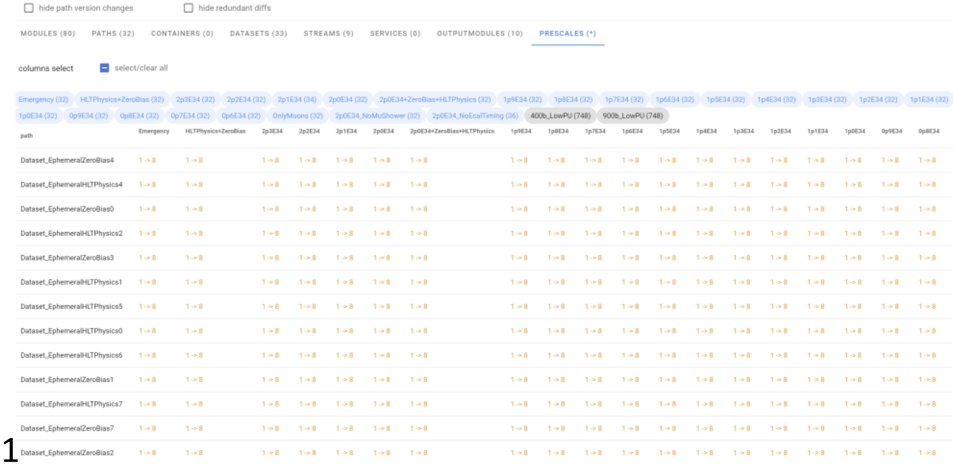
In the case of having to move to the emergency column please call the HLT doc during working hours. Outside of working it hours, call at the L1 DOC/similar expert's discretion.'

Current Problems

severity	type	name	rate (Hz)	limit (Hz)
⚠	dataset	Cosmics	800.0	800
⚠	stream	Cosmics	5000.0	4000

Rows per page: 10 1-2 of 2

Reference: online:/cdaq/physics/Run2023/2e34/v1.2.1/HLT/V1 ⇄ Target: online:/cdaq/physics/Run2023/2e34/v1.2.3/HLT/V4 DIFF



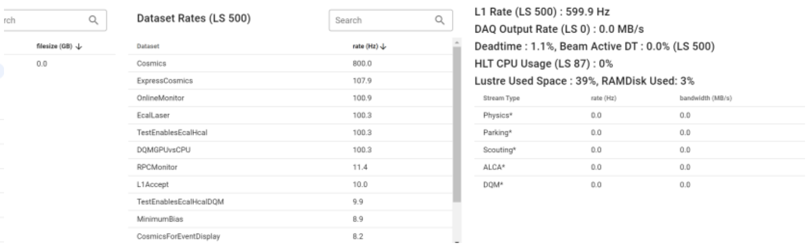
☐ hide path version changes ☐ hide redundant diffs

MODULES (80) PATHS (32) CONTAINERS (0) DATASETS (33) STREAMS (9) SERVICES (0) OUTPUTMODULES (10) PRESCALES (*)

columns select ☐ select/clear all

Emergency (32) HLTPhysics+ZeroBias (32) 2p3E34 (32) 2p2E34 (32) 2p1E34 (34) 2p0E34 (32) 2p0E34+ZeroBias+HLTPhysics (32) 1p9E34 (32) 1p8E34 (32) 1p7E34 (32) 1p6E34 (32) 1p5E34 (32) 1p4E34 (32) 1p3E34 (32) 1p2E34 (32) 1p1E34 (32) 1p0E34 (32) 0p9E34 (32) 0p8E34 (32) 0p7E34 (32) 0p6E34 (32) OnlyMuons (32) 2p0E34_NoMuShower (32) 2p0E34_NoEcalTiming (36) 400b_LowPU (748) 900b_LowPU (748)

path	Emergency	HLTPhysics+ZeroBias	2p3E34	2p2E34	2p1E34	2p0E34	2p0E34+ZeroBias+HLTPhysics	1p9E34	1p8E34	1p7E34	1p6E34	1p5E34	1p4E34	1p3E34	1p2E34	1p1E34	1p0E34	0p9E34	0p8E34
Dataset_EphemeralZeroBias4	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralHLTPhysics4	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralZeroBias0	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralHLTPhysics2	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralZeroBias3	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralHLTPhysics1	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralHLTPhysics5	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralHLTPhysics0	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralHLTPhysics6	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralZeroBias1	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralHLTPhysics7	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralZeroBias7	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8
Dataset_EphemeralZeroBias2	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8	1 -> 8



filesize (GB) ↓
0.0

Dataset Rates (LS 500)

Dataset	rate (Hz) ↓
Cosmics	800.0
ExpressCosmics	107.9
OnlineMonitor	100.9
EcalLaser	100.3
TestEnablesEcalHcal	100.3
DQMGPUvsCPU	100.3
RPCMonitor	11.4
L1Accept	10.0
TestEnablesEcalHcalDQM	9.9
MinimumBias	8.9
CosmicsForEventDisplay	8.2

L1 Rate (LS 500) : 599.9 Hz
DAQ Output Rate (LS 0) : 0.0 MB/s
Deadtime : 1.1%, Beam Active DT : 0.0% (LS 500)
HLT CPU Usage (LS 87) : 0%
Lustre Used Space : 39%, RAMDisk Used: 3%

Stream Type	rate (Hz)	bandwidth (MB/s)
Physics*	0.0	0.0
Parking*	0.0	0.0
Scouting*	0.0	0.0
ALCA*	0.0	0.0
DQM*	0.0	0.0

Summary

- I see a big problem limiting our physics by wasting our personpower due to substandard tools
  - its absolutely limiting our physics on CMS
- I have no idea how to properly solve it
- should swifthep be involved here, is this something we should think about?
  - eg provide off the shelf solutions /examples
  - it would have been a huge help for me when starting out
- or is this something we consider more appropriate for others to solve ?