Prediction of event processing time at the CMS experiment of the Large Hadron Collider

Samir Cury¹, Oliverutsche², Dorian Kcira¹
¹California Institute of Technology, ²Fermi National Accelerator Laboratory

One of the biggest challenges of the CMS experiment is the precise reconstruction of charged particle tracks in the detectors as well as the combination of information from the different sub-detectors. This is done through carefully designed, elaborate algorithms, which translate into CPU-intensive tasks. At the Large Hadron Collider, understanding the details of the algorithm performance and its relation to event complexity is one of the key factors to facilitate the processing of workflows in a more uniform and efficient way. The analysis presented here aims at estimating the event reconstruction time for the future LHC data based on observations on data already acquired.

The following is a measurement of the CMS software (CMSSW) performance for a given software release and type of events (primary dataset). The performance varies significantly according to the type of physics. Different physics signatures naturally produce more, or less hits used for reconstructing tracks. These measurements on existing processed data are used to estimate processing time of future LHC data. One important factor to consider in the estimate is systematic shifts in these measurements caused by the heterogeneity of the processing farms. Different CPU models will result in different processing times for the same collision type. Our measurements have been done over different CPU models so we believe that the resulting average is a representative value that will be the most useful as an estimate for the CMS central operations.

The complexity of track reconstruction is due to the large number of charged particle tracks from the collisions as well as the overlap among them. Iterations become thus necessary in order not only to fit hits in the tracking detectors but also to distinguish the possible contributions coming from combinatorics. The number of hits used for reconstructing tracks depends strongly on instantaneous luminosity and the number of collisions that happen simultaneously per beam bunch crossing (pick-up interactions). The job-splitting algorithm uses performance information collected at the CMS Tier-0 farm to estimate processing times. The observed effect is demonstrated in the figures below. Both figures are based on real log messages to demonstrate how the algorithm works. The observed effects on real data are presented in the paper.

The CMS Fill Report provides plots of instantaneous luminosity and pile-up over time. Here we can compare those with the observed reconstruction time per event of these data observed at the CMS Tier-0. The comparison shows that per-event processing times have a direct relation with pile-up.

Due to the wide range of luminosity values, a wide distribution of job lengths in a multi-run reconstruction workflow is observed. As a consequence, the total time taken to process the data is more uniform when processing high-luminosity data, where a single job can take up to 4 days to finish and more in the case of remiss needed due to job failures.

This study motivated a solution to diminish the long tails in CMS data processing. As the relation between instantaneous luminosity and reconstruction time is well known, we are able to predict the time-per-event by using the luminosity value from the data. Different CMS web services exist that provide access to this kind of information. A job-splitting algorithm was developed for the Workload Management Agent that uses this information to estimate a processing time per event. In addition, the number of events per processing job is dynamically such that the processing times become more uniform. The ideal processing time per job is approximately 6 hours.

The job-splitting algorithm uses performance information collected at the end of each workflow by the WMAs. This information is reported to a specific service maintained by the CMS Tier-0. The information is not only used from the data service for automated systems but also for web-based interfaces to be used by CMS members to visualize performance curves in average processing times per release and dataset.

The observed effect is demonstrated in the figures below. Both figures are based on real log messages to demonstrate how the algorithm works. The improvements shown here are expected to be even larger in future production systems as the performance information gathered is expected to increase.

Acknowledgements


References

[1] D Giordano and G Sguazzoni, CMS reconstruction improvements for the tracking in 2013, doi:10.1007/978-3-642-40599-0_27

[2] The CMS Collaboration, Performance of CMS muon reconstruction in pp collision events at sqrt(s) = 8 TeV, CERN-2012-047


[4] Poster presenter: gutsche@fnal.gov

1. 2013-05-03 14:07:50.088 DELPHI.LuminosityBased This file has average instantaneous luminosity 254373.9384 average time per event 13.937964 and is getting 1178 events per job.

2. 2013-05-03 14:07:50.088 DELPHI.LuminosityBased This file has average instantaneous luminosity 254373.9384 average time per event 13.937964 and is getting 1178 events per job.

3. Example of real log messages to demonstrate the algorithm works.