FERMILAB-CONF-13-494-CD

Available on CMS information server

CMS CR -2013/375

The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

29 October 2013

Event processing time prediction at the CMS Experiment of the Large Hadron Collider

Samir Cury Siqueira for the CMS Collaboration

Abstract

The physics event reconstruction in LHC/CMS is one of the biggest challenges for computing.Among the different tasks that computing systems perform, the reconstruction takes most of the CPUresources that are available. The reconstruction time of a single event varies according to the event complexity.Measurements were done in order to find precisely this correlation, creating means to predict it-based on the physics conditions of the input data.Currently the data processing system do not account that whensplitting a task in chunks(jobs), this can cause a considerablevariation in the job length, thus a considerable increase into theworkflow Estimated Time of Arrival.The goal is to use this estimate on processing time to more efficiently split the work inchunks, considering the CPU time needed for each chunk and due to this,lowering the standard deviation of the job length distribution in aworkflow.

Presented at *CHEP2013 Computing in High Energy Physics 2013*

Event processing time prediction at the CMS experiment of the Large Hadron Collider

Samir Cury[1], **Oliver Gutsche**[2], **Dorian Kcira**[3]

[1,3] California Institute of Technology

[2] Fermi National Accelerator Laboratory

E-mail: [1] samir@hep.caltech.edu, [2] gutsche@fnal.gov, [3] dkcira@caltech.edu

Abstract. The physics event reconstruction is one of the biggest challenges for the computing of the LHC experiments. Among the different tasks that computing systems of the CMS experiment performs, the reconstruction takes most of the available CPU resources. The reconstruction time of single collisions varies according to event complexity. Measurements were done in order to determine this correlation quantitatively, creating means to predict it based on the data-taking conditions of the input samples. Currently the data processing system splits tasks in groups with the same number of collisions and does not account for variations in the processing time. These variations can be large and can lead to a considerable increase in the time it takes for CMS workflows to finish. The goal of this study was to use estimates on processing time to more efficiently split the workflow into jobs. By considering the CPU time needed for each job the spread of the job-length distribution in a workflow is reduced.

1. Introduction

The main goal of this study is to find the factors from data-taking with most influence the processing time of the resulting saved collision data (also called "events"). After quantifying the relation between these factor and the event processing times, systematic deviations are studied as a result of the data-taking parameters and the reduction of those systematic effects is addressed.

It is expected that one of the main factors that influences event processing time is the charged particle track multiplicity of those events. Measurements are presented below of the relation between the two and a method is developed for predicting the processing time of future data in the same luminosity range.

2. Accelerator and collisions

2.1. Charged particle track multiplicity

The complexity of track reconstruction is to a large extent due to the large number of charged particle tracks from the collisions as well as the overlap among these collisions. Iterations become thus necessary in order not only to fit hits in the tracking detectors but also to disentangle the different possibilities resulting from the large combinatorics (see Fig. 1). The number of hits used for track reconstruction depends strongly on instantaneous luminosity and the number of collisions that happen simultaneously per beam bunch crossing (pile-up interactions). Pile-up itself is a function of the accelerator running conditions and instantaneous luminosity. This study focuses, therefore, on instantaneous luminosity.

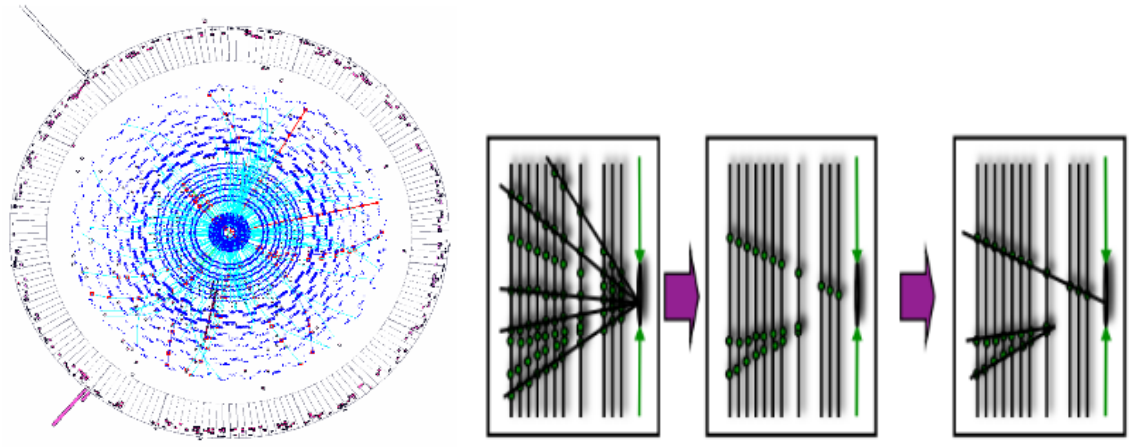


Figure 1. *(left) Visualization of a collision with several charged tracks in the CMS detector. (right) Schematic representation of the steps for the reconstruction and disambiguation of tracks out of many different hits in the CMS tracking system.*

2.2. Instantaneous luminosity

2012 was the most challenging year for the LHC Run1 data-taking (2010 to the first quarter of 2013). The LHC operational energy was 8 TeV. Record instantaneous luminosity values of the collisions were achieved from the LHC and the resulting collisons measured in the CMS detector. This, on its turn, translates into large track multiplicities and increasing the number of pile-up (PU) interactions per bunch-crossing.

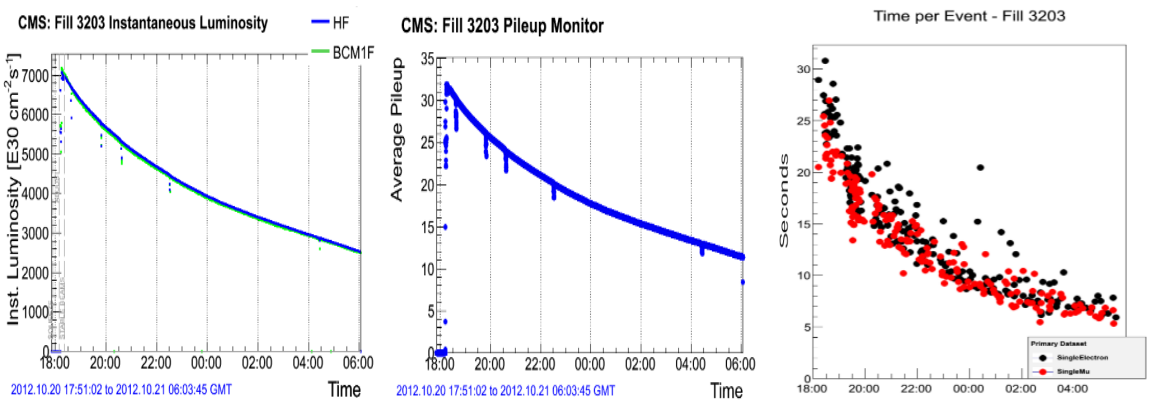


Figure 2. *Instantaneous luminosity (left), pile-up curve (middle), and event processing time (right) as a function of the passing time within a data-taking run at the CMS experiment.*

In a typical data-taking run, the instantaneous luminosity ranges from $7200{\cdot}10^{30}\mathrm{cm}^{-2}\mathrm{s}^{-1}$ at the beginning of the fill to $2500{\cdot}10^{30}\mathrm{cm}^{-2}\mathrm{s}^{-1}$ at the end of the same fill. This results in a number of PU interactions of about 34 at the beginning of the fill and about 12 at its end. Fig. 2 shows the instantaneous luminosity (left), the pile-up curve (middle) as well as the processing time of events (right) as a function of time during a data-taking run. The proportionality between the three quantities can be observed.

Fig 3 shows event displays of the charged particle tracks in the CMS detector coming out of LHC collisions. Two cases are distinguished, the sparse low-luminosity case (left), with lower number of tracks (16 PU events/bunch crossing), and the dense high-luminosity case (right), with large number of tracks (27 PU/bunch crossing).

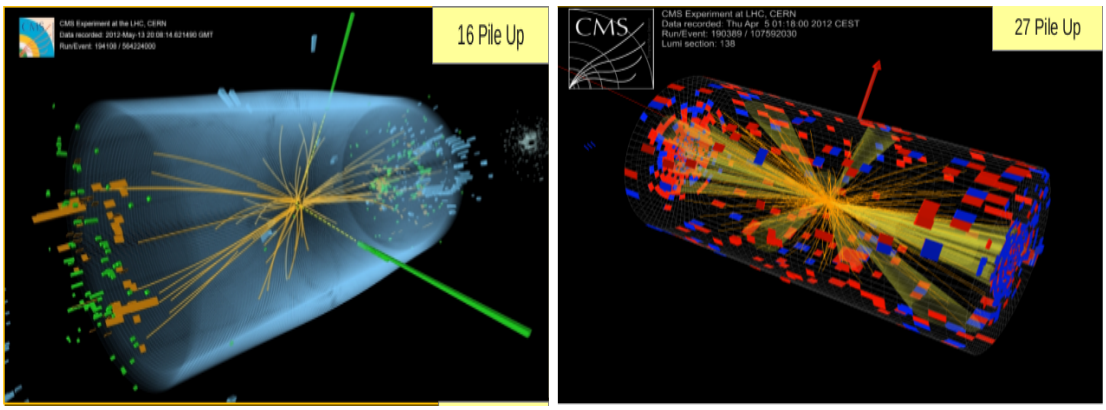


Figure 3. *Visualization of events with charged tracks in the CMS detector for the low-luminosity case (left) and the high-luminosity one (right).*

3. Measurements of event processing times

3.1. Performance curves

The computer performance varies significantly according to the type of physics of the events being processed. Different physics signatures naturally produce more, or less tracks. In this study, measurements on existing processed data are used to estimate processing time of future LHC data. One important factor to consider in the estimate are systematic shifts in the measurements caused by the heterogeneity of the processing farms. Different CPU models will result in different processing time 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.

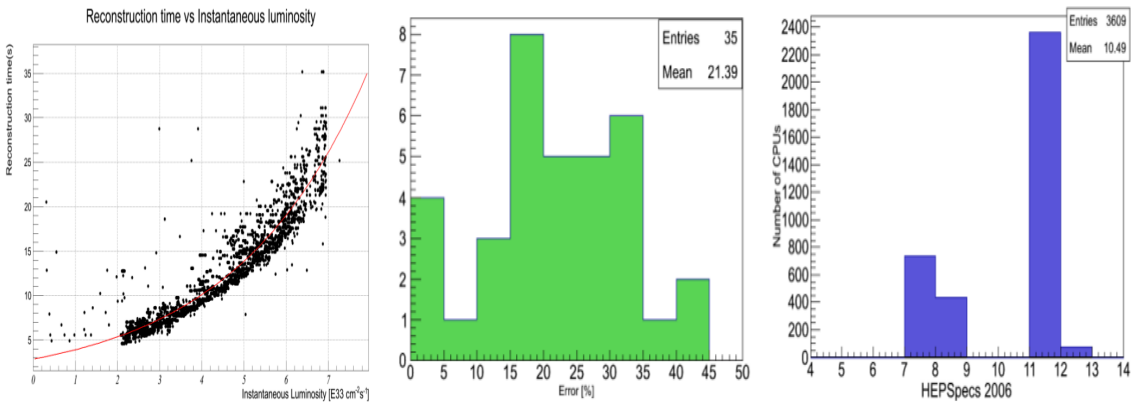


Figure 4. *(left) Measurement of the CMSSW performance with respect to event reconstruction time. (middle) Spread introduced in the reconstruction time by CPU speed fluctuations. (right) Histogram of number of cores in a reconstruction farm in bins of HEPSpec06 values.*

Fig. 4 (left) shows a measurement of the CMS software (CMSSW) performance for a given software release and type of events (primary dataset), in this particular case events with at least one isolated muon (so-called Single Muon).

3.2. Systematic errors

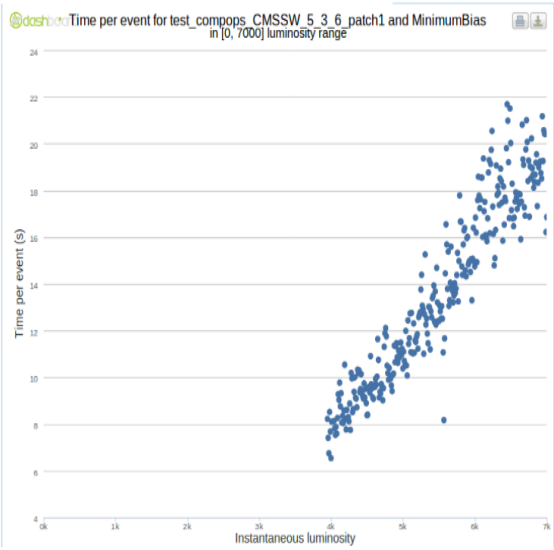
Measurements were done on 35 PromptReco workflows to observe how close to the real value our estimation gets. The error introduced by the CPU speed fluctuation in the Tier 0 farm can be up to 37.75%. This comes from the difference of HEPSpecs 2006 (benchmark unit) between

the fastest and slowest CPU models used. The results are shown in Fig. 4 (middle). The green histogram shows the distribution of error values for all workflows. In Fig. 4 (right) the blue is a histogram of the number of cores in the farm per HS06 values, showing how they contribute to the error.

4. Improvements to monitoring and job splitting

4.1. Monitoring

The job-splitting algorithm uses performance information collected at the end of each workflow run by the CMS workflow-management agents (WMAgents). This information is reported to a specific service maintained by the CMS Dashboard and is used from the data service for automated systems. It is also used by CMS members through a web interface in order to visualize performance curves or average processing times per release and dataset. A performance curve from the CMS Dashboard is shown in Fig. 5 (left). An example of real log messages that demonstrate how the algorithm works is shown in Fig. 5 (right).



2013-10-06 09:47:05,045:DEBUG:LuminosityBased:
This file has average instantaneous luminosity
5202.817391 average time per event 11.358000
and is getting 1901 events per job

2013-10-06 09:47:05,060:DEBUG:LuminosityBased:
This file has average instantaneous luminosity
7228.521739 average time per event 27.652917
and is getting 781 events per job

Figure 5. *CMS DashBoard: (left) Performance curve from reported data; (right) An example of real log messages that demonstrate how the algorithm works.*

4.2. Job splitting

This study motivated a solution to diminish the long tail effects in CMS data processing. As the relation between instantaneous luminosity and reconstruction time is now well determined, 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 chosen dynamically such that the processing times become more uniform. The ideal processing time per job is approximately 8 hours.

4.3. Results

The observed improvements are shown in Fig. 6. Both the left and right figure are result of the same Reconstruction Workflow. The left figure shows the effect for jobs splitted by the common EventBased algorithm, where the number of events is fixed for all jobs. The second figure shows the case where the splitting is done dynamically by the algorithm LuminosityBased algorithm,

described above. A considerably narrower distribution is observed, as expected. This is a result of the job execution time becoming more uniform in the workflows. The algorithm will fall back to the EventBased method for the case where the performance information is not yet available. The improvements shown here are expected to be even larger in future production systems as the performance information gathered is expected to increase.

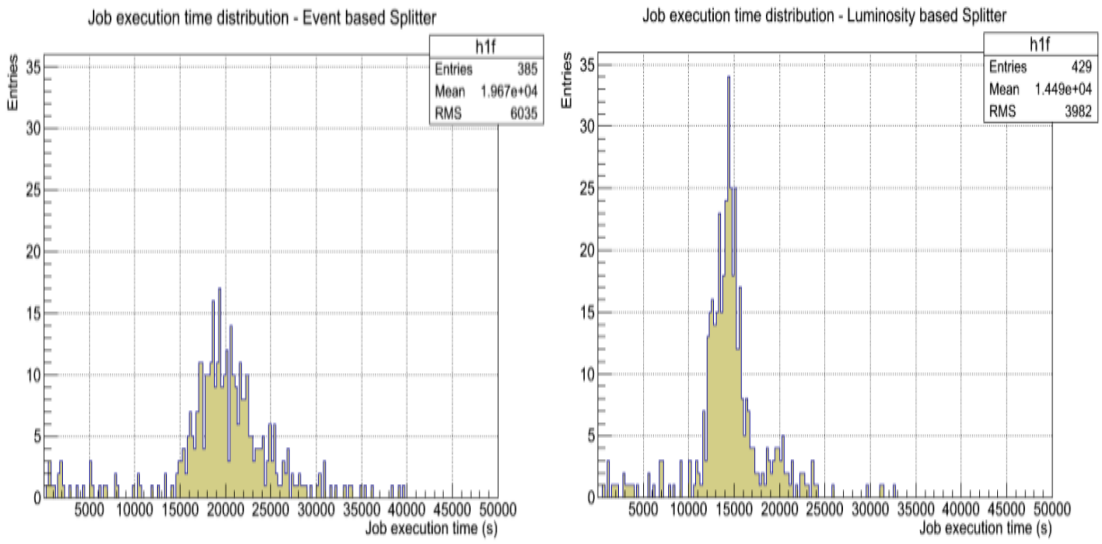


Figure 6. *CMS DashBoard: (left) Performance curve from reported data; (right) An example of real log messages that demonstrate how the algorithm works.*

5. Conclusions

This initial study shows that it is feasible to estimate the time-per-event behavior for reconstruction workflows of CMS. It was observed that heterogeneous computing farms introduce considerable systematic variations into the workflows. This behavior can be taken into account and corrected for. We demonstrated how this information can be used in order to reduce the data processing tails, which have been until now a problem for CMS central production, impacting time-critical prompt-reconstruction workflows in the CMS Tier 0. Furthermore, a job splitting algorithm has been developed that uses performance data dynamically according to the data-taking conditions for the input samples.

6. Acknowledgements

CMS Tier 0 Team, CMS Workload Management Development Team, WLCG DashBoard Team, UERJ Department of High Energy Physics, US-CMS group at the California Institute of Technology, US-CMS group at the Fermi National Accelerator Laboratory.

7. References

[1] D. Giordano and G. Sguazzoni, "CMS reconstruction improvements for the tracking in large pile-up events", doi:10.1088/1742-6596/396/2/022044

[2] The CMS Collaboration, "Performance of CMS muon reconstruction in pp collision events at $\sqrt{(s)} = 7$ TeV", arXiv:1206.4071

[3] T. Hauth, V. Innocente and D. Piparo, "Development and Evaluation of Vectorised and Multi-Core Event Reconstruction Algorithms within the CMS Software Framework", doi:10.1088/1742-6596/396/5/052065

[4] http://www.hep.caltech.edu/cms