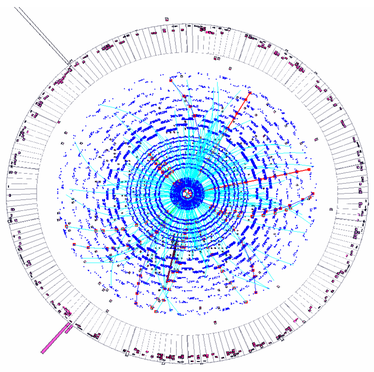
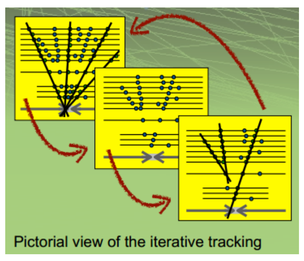
Prediction of event processing time at the CMS experiment of the Large Hadron Collider

Samir Cury Siqueira (UERJ)

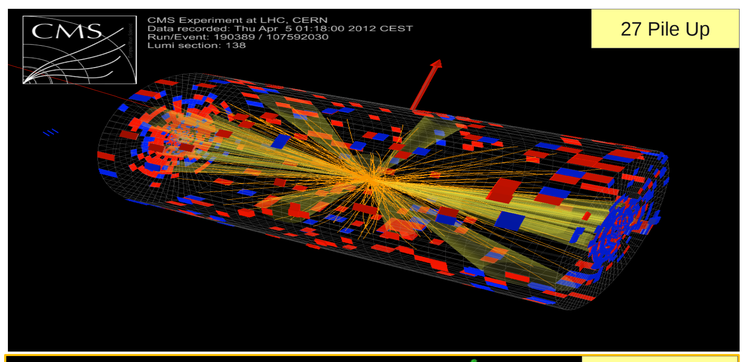
One of the biggest challenges in the CMS detector is the precise reconstruction of particle tracks. This is done by very complex algorithms, which translates into a CPU intensive task. At the scale of the LHC, understanding how the algorithm performance behaves according to event complexity is one of the key factors to process workflows in a more uniform and efficient way. This analysis makes possible to, based on previous observation, estimate how the event reconstruction time will behave for incoming data.

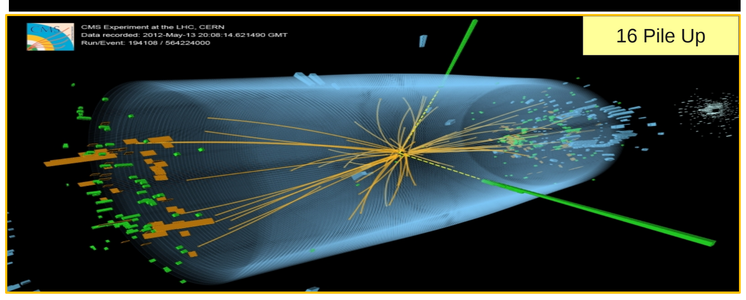




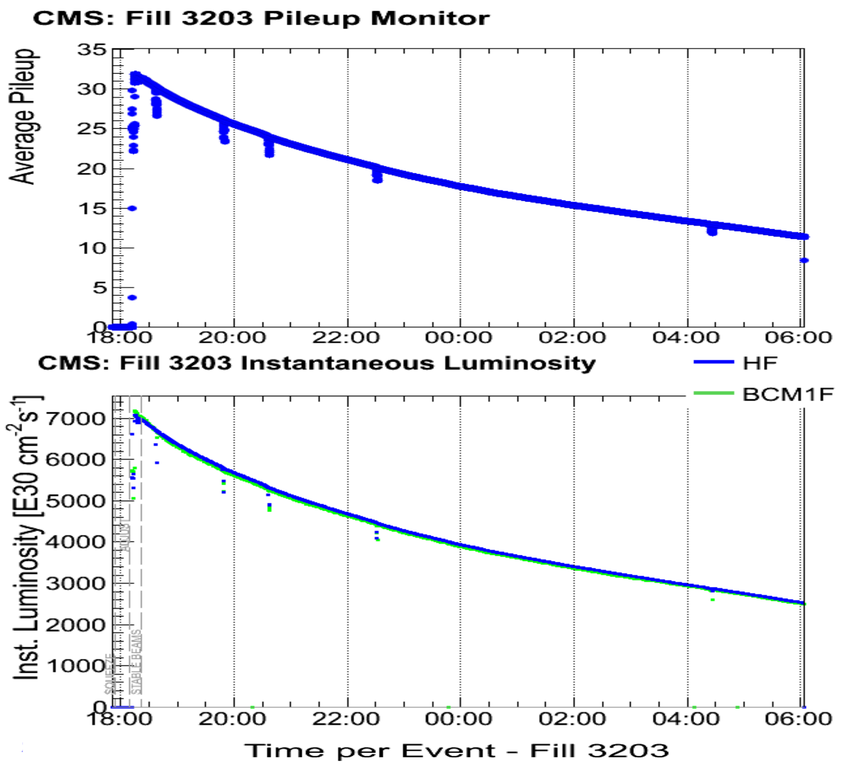
CMS Tracker – event visualization (ORCA)

The complexity of track reconstruction comes from the number of tracks and how much they overlap, causing the algorithm to iterate more before it distinguishes the tracks. This has a direct relation with the instantaneous luminosity, subsequently with the "Number of Pile Up interactions per bunch crossing". The latter is not measured, but a function of the accelerator running conditions and instantaneous luminosity. For this reason we are focusing in instantaneous luminosity on this study, although Pile Up is a more intuitive value.
The Event Display shows visually how the track population looks like in a high pile up event, and in a low pile up one.

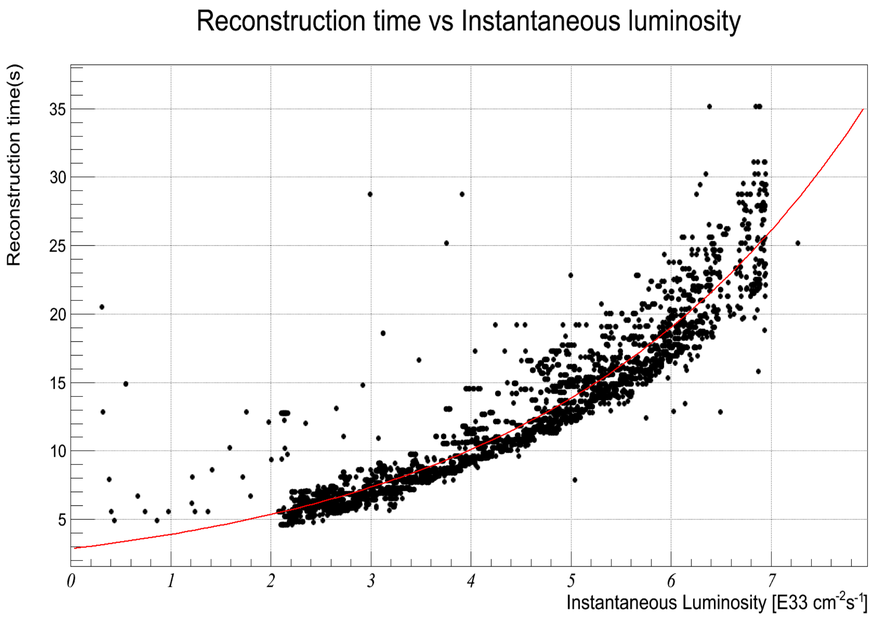




The CMS Fill Report provides plots of instantaneous luminosity and pile up over time of a given fill, here we can compare those with the reconstruction time per event of this data observed in the Tier-0. This comparison shows that time per event has a direct relation with pileup.



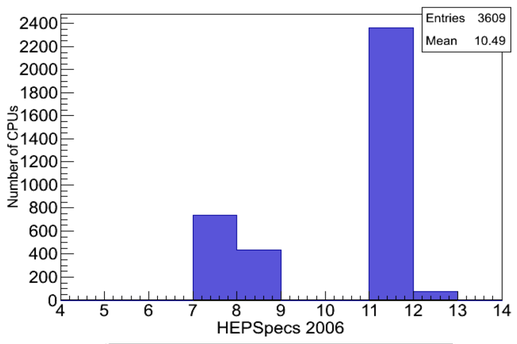
The following is a curve of the CMSSW performance, for a given Release and Primary Dataset (type of event). It varies significantly according to the type of physics, different physics might naturally produce more, or less tracks.
This is a way to estimate what the time per event should be according to what was observed in already processed data. There is an important systematic error in this measurement, caused by the fact that the workflows run in heterogeneous farms. different CPU models will result in different processing times for the same kind of event.The advantage is that, as a general curve, it averages among most of the CPU models that we have in the farms that we utilize. It is at the end the most useful value for central operations.



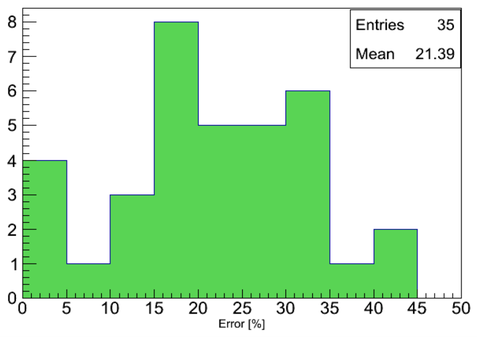
Generic Single Muon performance curve for CMS Reconstruction

Measurements were done on 35 PromptReco workflows to observe how close to the real value the estimation can get. The error introduced by the CPU speed fluctuation, In the Tier-0 farm, is up to 37.75%, which is, the difference of HEPSpecs 2006 (Benchmark unit) between the fastest and slowest CPU model.
The green histogram shows the distribution of error values for all workflows, While the blue is a histogram of number of cores in the farm per HS06 values, showing how they contribute to the error. In the table below, some specific measurements, from smaller to bigger error.

Run #	Estimated job length	Observed job length	Error [%]
202075	6:15:00	6:14:35	0.1
202088	3:12:44	2:59:39	7.2
202014	6:50:15	5:39:11	20.9
201707	5:20:26	7:53:23	32.3

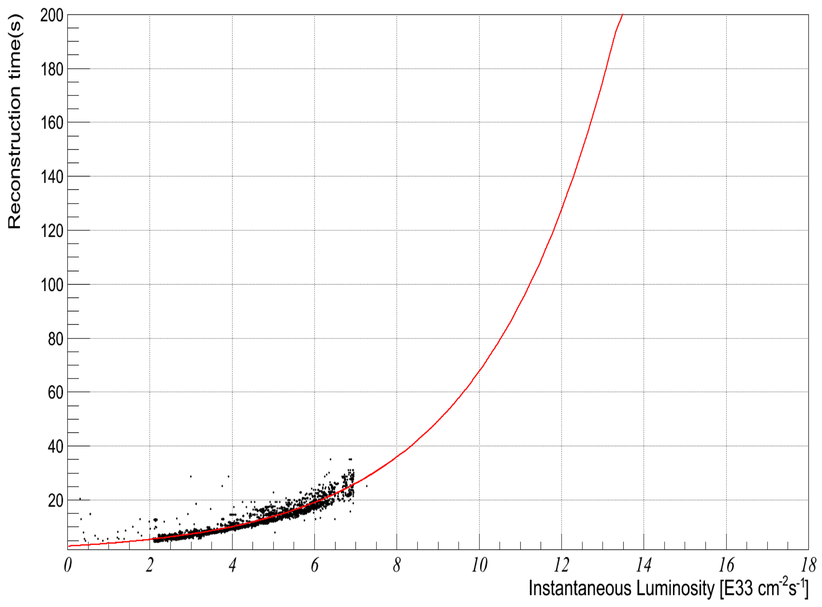


Tier-0 Farm CPU speed distribution



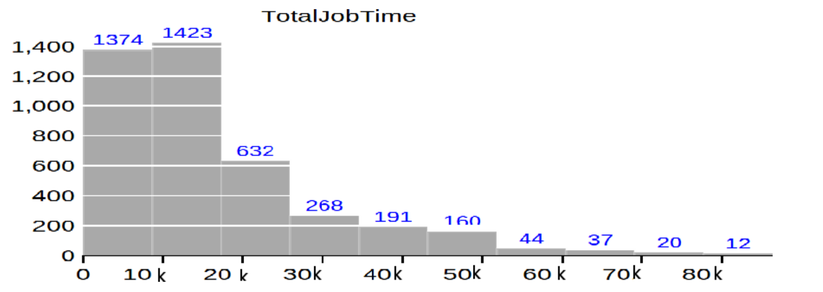
Error distribution – expected x observed average job length

One of the uses of this curve, is to have an idea on how the reconstruction time will look like in higher luminosities, for example, extrapolating until the Run2 (2015) luminosity. Obviously some factors will change and improve the curve parameter, so this is just a guideline to what kind of challenge lies ahead if nothing changes, not a precise report. The Monte-carlo performance agrees until 10e33.

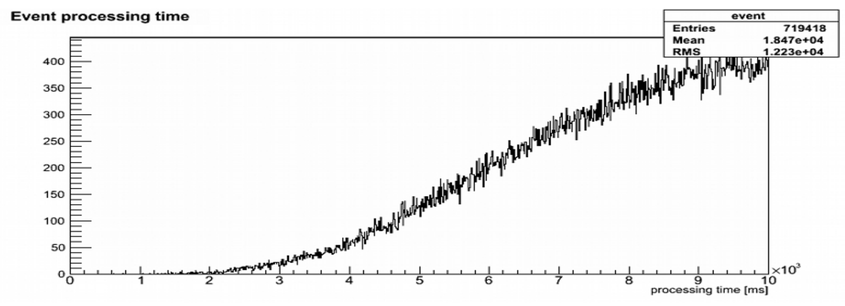


Performance curve extrapolated until Run2 luminosity

Due to the wide range of luminosity, and its effect on time per event, it causes a wide distribution of job length in a multi-run reconstruction workflow. the consequence, is the famous tails effect, where the workflow takes in average one week more to finish after the 99% of the original request has finished most of the processing, this delay is caused by jobs processing high luminosity data, that can take up to 48h to finish. If they retry, even longer.

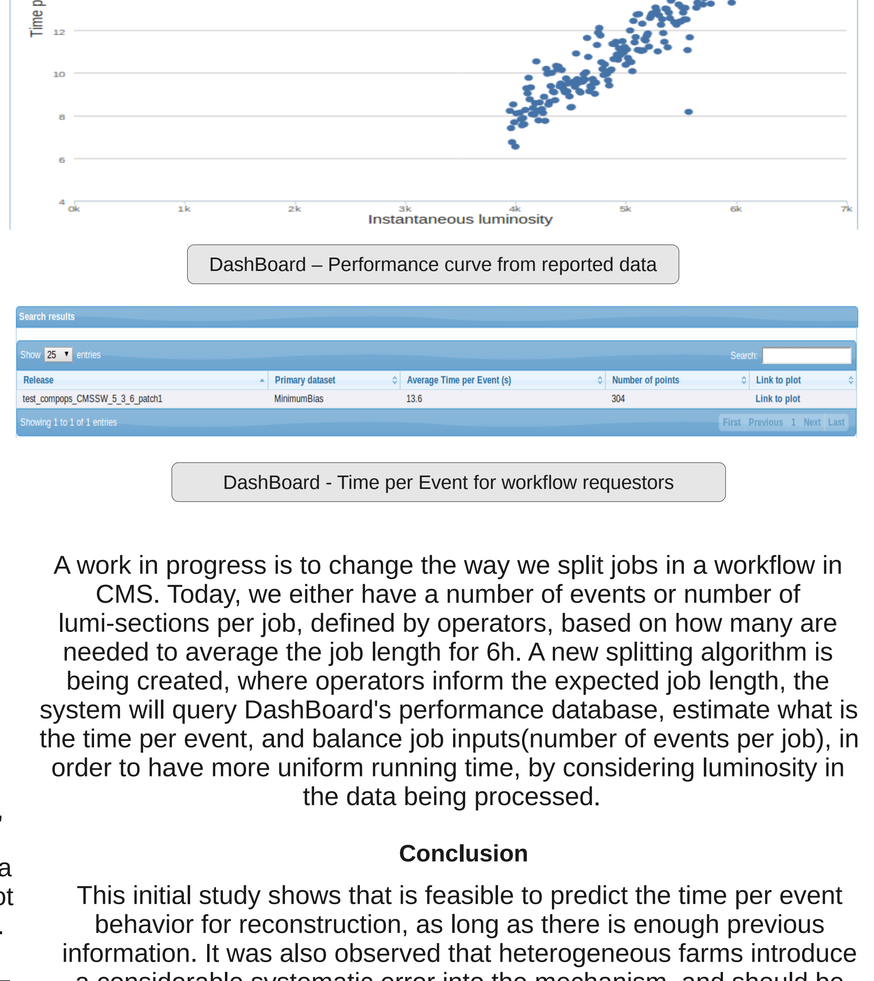


Wide job length distribution – consequence of wide event processing time range

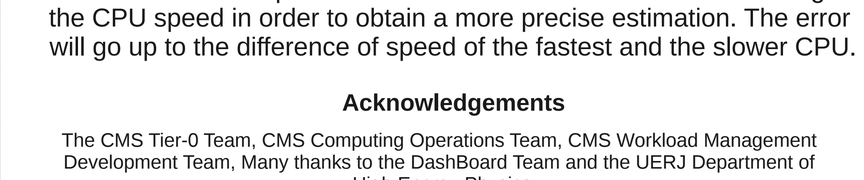


Wide processing time range - consequence of wide instantaneous luminosity range

In order to have automatic ways to monitor the processing time behavior, there were developed automated ways to generate this plot. At the end of a reconstruction workflow, the Workload Management Agent, harvests the performance information and uploads to a central database, in CMS DashBoard. This information is used in monitoring interfaces, and can also be queried by automated systems and scripts, through a DataService.



DashBoard – Performance curve from reported data



DashBoard - Time per Event for workflow requestors

A work in progress is to change the way we split jobs in a workflow in CMS. Today, we either have a number of events or number of lumi-sections per job, defined by operators, based on how many are needed to average the job length for 6h. A new splitting algorithm is being created, where operators inform the expected job length, the system will query DashBoard's performance database, estimate what is the time per event, and balance job inputs(number of events per job), in order to have more uniform running time, by considering luminosity in the data being processed.

Conclusion

This initial study shows that is feasible to predict the time per event behavior for reconstruction, as long as there is enough previous information. It was also observed that heterogeneous farms introduce a considerable systematic error into the mechanism, and should be corrected. The time per event value should be normalized according to the CPU speed in order to obtain a more precise estimation. The error will go up to the difference of speed of the fastest and the slower CPU.

Acknowledgements

The CMS Tier-0 Team, CMS Computing Operations Team, CMS Workload Management Development Team, Many thanks to the DashBoard Team and the UERJ Department of High Energy Physics

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

LISHEP 2013 Conference
Poster presenter : samir@cern.ch