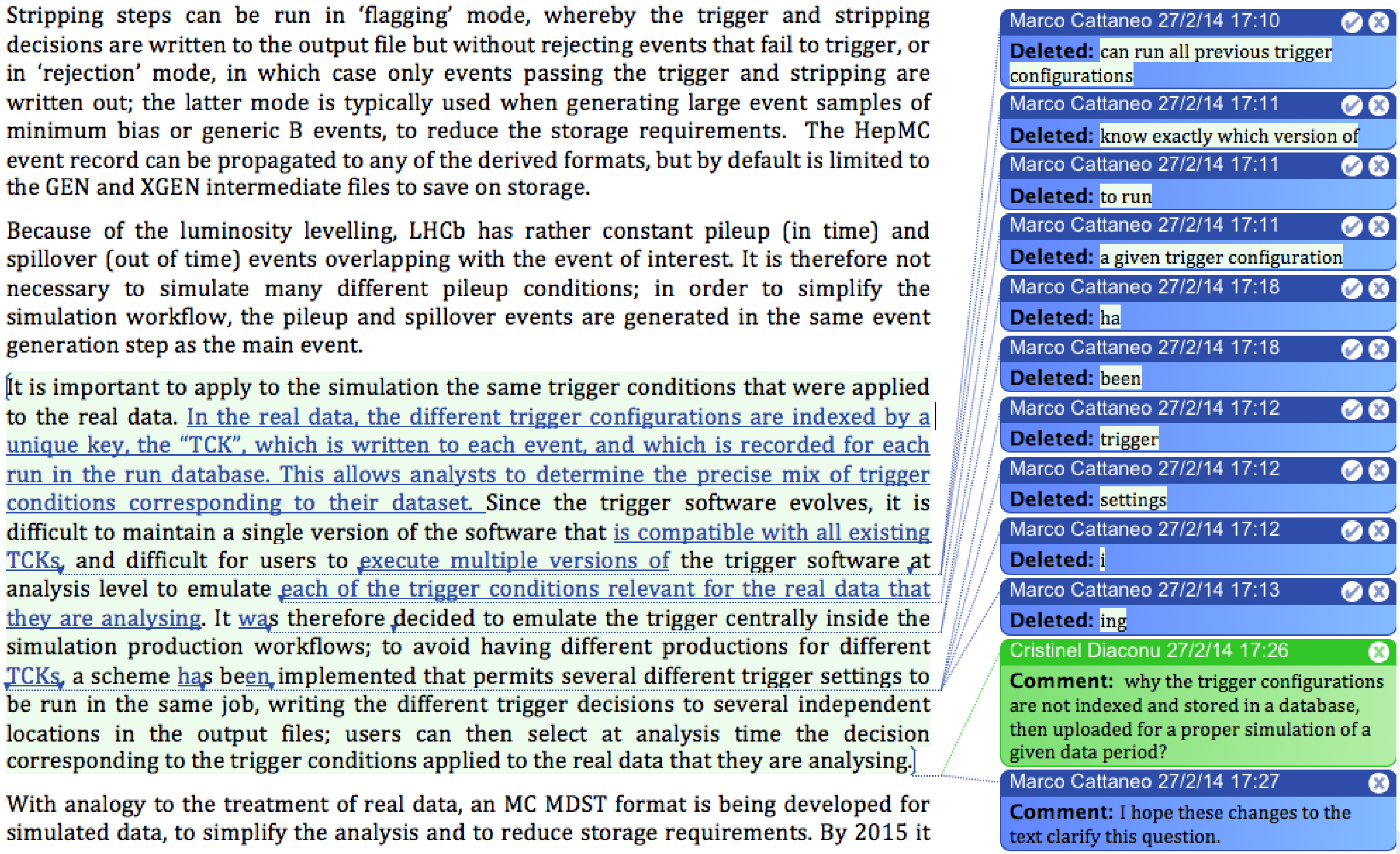
LHCb Computing

CM document discussion

Clarifications, e.g. on trigger simulation



Stripping steps can be run in 'flagging' mode, whereby the trigger and stripping decisions are written to the output file but without rejecting events that fail to trigger, or in 'rejection' mode, in which case only events passing the trigger and stripping are written out; the latter mode is typically used when generating large event samples of minimum bias or generic B events, to reduce the storage requirements. The HepMC event record can be propagated to any of the derived formats, but by default is limited to the GEN and XGEN intermediate files to save on storage.

Because of the luminosity levelling, LHCb has rather constant pileup (in time) and spillover (out of time) events overlapping with the event of interest. It is therefore not necessary to simulate many different pileup conditions; in order to simplify the simulation workflow, the pileup and spillover events are generated in the same event generation step as the main event.

It is important to apply to the simulation the same trigger conditions that were applied to the real data. In the real data, the different trigger configurations are indexed by a unique key, the "TCK", which is written to each event, and which is recorded for each run in the run database. This allows analysts to determine the precise mix of trigger conditions corresponding to their dataset. Since the trigger software evolves, it is difficult to maintain a single version of the software that is compatible with all existing TCKs, and difficult for users to execute multiple versions of the trigger software at analysis level to emulate each of the trigger conditions relevant for the real data that they are analysing. It was therefore decided to emulate the trigger centrally inside the simulation production workflows; to avoid having different productions for different TCKs, a scheme has been implemented that permits several different trigger settings to be run in the same job, writing the different trigger decisions to several independent locations in the output files; users can then select at analysis time the decision corresponding to the trigger conditions applied to the real data that they are analysing.

With analogy to the treatment of real data, an MC MDST format is being developed for simulated data, to simplify the analysis and to reduce storage requirements. By 2015 it

Marco Cattaneo 27/2/14 17:10
Deleted: can run all previous trigger configurations

Marco Cattaneo 27/2/14 17:11
Deleted: know exactly which version of

Marco Cattaneo 27/2/14 17:11
Deleted: to run

Marco Cattaneo 27/2/14 17:11
Deleted: a given trigger configuration

Marco Cattaneo 27/2/14 17:18
Deleted: ha

Marco Cattaneo 27/2/14 17:18
Deleted: been

Marco Cattaneo 27/2/14 17:12
Deleted: trigger

Marco Cattaneo 27/2/14 17:12
Deleted: settings

Marco Cattaneo 27/2/14 17:12
Deleted: i

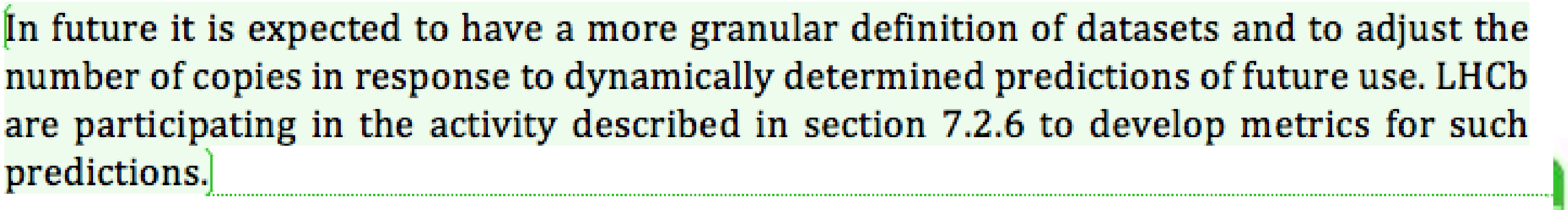
Marco Cattaneo 27/2/14 17:13
Deleted: ing

Cristinel Diaconu 27/2/14 17:26
Comment: why the trigger configurations are not indexed and stored in a database, then uploaded for a proper simulation of a given data period?

Marco Cattaneo 27/2/14 17:27
Comment: I hope these changes to the text clarify this question.

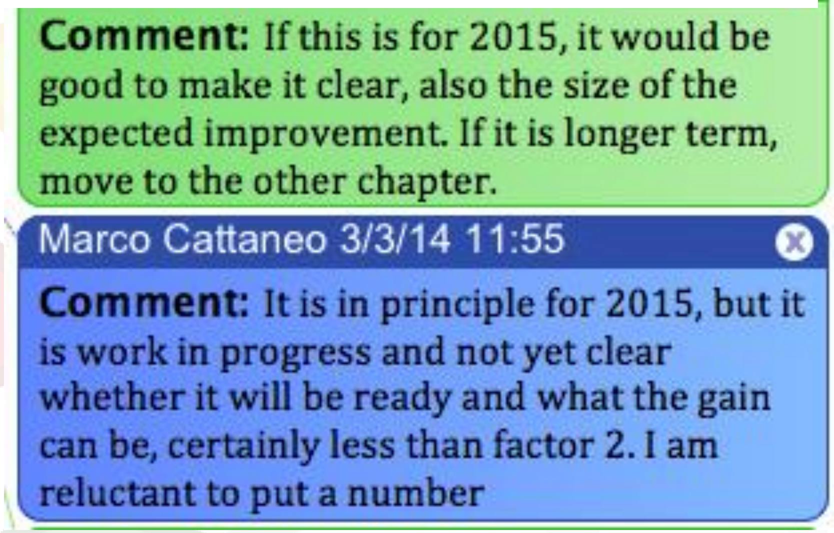
Made changes to text to hopefully clarify some of the questions
How to iterate?

Work in Progress – e.g. Data Placement



In future it is expected to have a more granular definition of datasets and to adjust the number of copies in response to dynamically determined predictions of future use. LHCb are participating in the activity described in section 7.2.6 to develop metrics for such predictions.

Difficult to be precise about work in progress, prefer to stay vague than make false promises

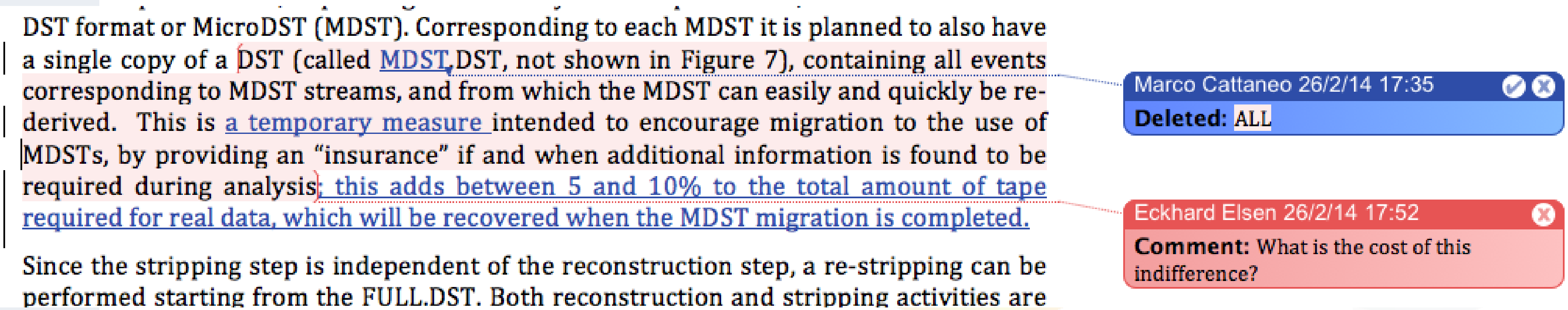


Comment: If this is for 2015, it would be good to make it clear, also the size of the expected improvement. If it is longer term, move to the other chapter.

Marco Cattaneo 3/3/14 11:55

Comment: It is in principle for 2015, but it is work in progress and not yet clear whether it will be ready and what the gain can be, certainly less than factor 2. I am reluctant to put a number

MDST.DST



DST format or MicroDST (MDST). Corresponding to each MDST it is planned to also have a single copy of a DST (called MDST.DST, not shown in Figure 7), containing all events corresponding to MDST streams, and from which the MDST can easily and quickly be re-derived. This is a temporary measure intended to encourage migration to the use of MDSTs, by providing an "insurance" if and when additional information is found to be required during analysis; this adds between 5 and 10% to the total amount of tape required for real data, which will be recovered when the MDST migration is completed.

Since the stripping step is independent of the reconstruction step, a re-stripping can be performed starting from the FULL.DST. Both reconstruction and stripping activities are

Marco Cattaneo 26/2/14 17:35

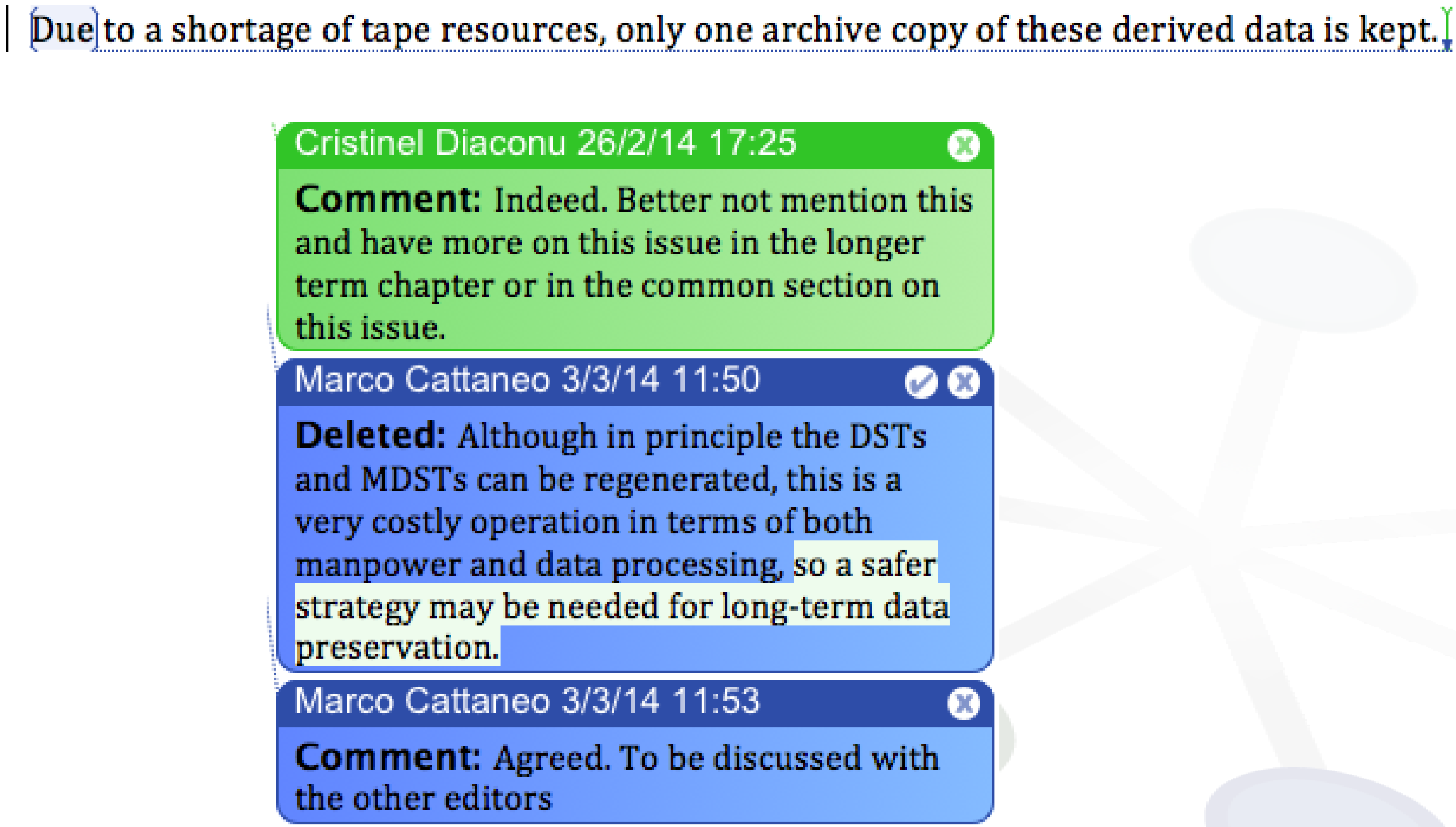
Deleted: ALL

Eckhard Elsen 26/2/14 17:52

Comment: What is the cost of this indifference?

- MDST.DST (was ALL.DST) intended to facilitate migration to MDST
  - This is work in progress
  - Cost is temporary, will be removed when migration is complete
    - Calling it "indifference" seems harsh

Cost of data preservation



Due to a shortage of tape resources, only one archive copy of these derived data is kept.

Cristinel Diaconu 26/2/14 17:25

Comment: Indeed. Better not mention this and have more on this issue in the longer term chapter or in the common section on this issue.

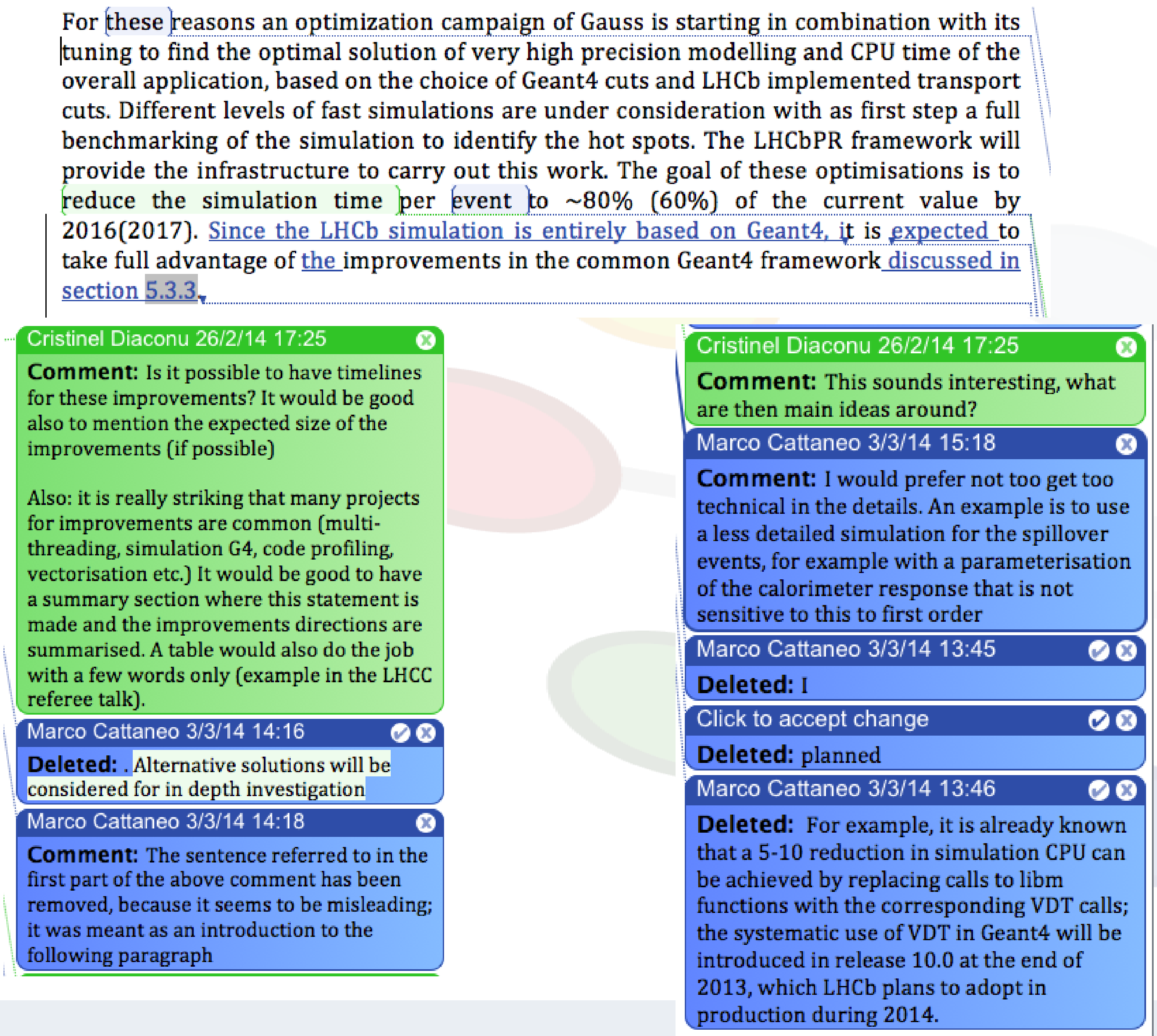
Marco Cattaneo 3/3/14 11:50

Deleted: Although in principle the DSTs and MDSTs can be regenerated, this is a very costly operation in terms of both manpower and data processing, so a safer strategy may be needed for long-term data preservation.

Marco Cattaneo 3/3/14 11:53

Comment: Agreed. To be discussed with the other editors

Simulation



For these reasons an optimization campaign of Gauss is starting in combination with its tuning to find the optimal solution of very high precision modelling and CPU time of the overall application, based on the choice of Geant4 cuts and LHCb implemented transport cuts. Different levels of fast simulations are under consideration with as first step a full benchmarking of the simulation to identify the hot spots. The LHCbPR framework will provide the infrastructure to carry out this work. The goal of these optimisations is to reduce the simulation time per event to ~80% (60%) of the current value by 2016(2017). Since the LHCb simulation is entirely based on Geant4, it is expected to take full advantage of the improvements in the common Geant4 framework discussed in section 5.3.3.

Cristinel Diaconu 26/2/14 17:25

Comment: Is it possible to have timelines for these improvements? It would be good also to mention the expected size of the improvements (if possible)

Also: it is really striking that many projects for improvements are common (multi-threading, simulation G4, code profiling, vectorisation etc.) It would be good to have a summary section where this statement is made and the improvements directions are summarised. A table would also do the job with a few words only (example in the LHCC referee talk).

Marco Cattaneo 3/3/14 14:16

Deleted: . Alternative solutions will be considered for in depth investigation

Marco Cattaneo 3/3/14 14:18

Comment: The sentence referred to in the first part of the above comment has been removed, because it seems to be misleading; it was meant as an introduction to the following paragraph

Cristinel Diaconu 26/2/14 17:25

Comment: This sounds interesting, what are then main ideas around?

Marco Cattaneo 3/3/14 15:18

Comment: I would prefer not too get too technical in the details. An example is to use a less detailed simulation for the spillover events, for example with a parameterisation of the calorimeter response that is not sensitive to this to first order

Marco Cattaneo 3/3/14 13:45

Deleted: I

Click to accept change

Deleted: planned

Marco Cattaneo 3/3/14 13:46

Deleted: For example, it is already known that a 5-10 reduction in simulation CPU can be achieved by replacing calls to libm functions with the corresponding VDT calls; the systematic use of VDT in Geant4 will be introduced in release 10.0 at the end of 2013, which LHCb plans to adopt in production during 2014.

Simulation

- Two questions:
  - What are ideas for reducing LHCb simulation time
    - Answer is improved use of transport cuts, adapt granularity of detector to precision required, do not simulate spillover events in detectors that are insensitive to it (Calo, Muon)
    - Work in progress, and details too technical to be described briefly
  - Common improvements in G4, vectorisation etc.
    - Indeed. Already described in common chapter. Does it need to be expanded?

Resources estimates

- Estimates foe 2015 are now different from request just made to C-RSG
    - □ How do we ensure consistency?