Feedback from ATLAS

Andrea Di Simone CERN and INFN/CNAF

- Comments for the review
- Performance issues

Comments to the reviewers

- Release notes
  - Like to see more reasons given for the changes in the release. The current comments tend to be very low level, making it difficult for users to understand the impact, positive or negative. Also like to see the internal validation test results. Some examples:
    - "Addresses problem report #847" is good. Even better to have a link.
    - "Changed default value of 'facgeom' (4 -> 3.5)" need clarification. What problem is addressed by this? What effect should the user see? Under what circumstances?
    - "Fixed problem for low energy hadrons" is unclear. What problem? What is low energy in this context? Am I likely to see an effect when working at the Tevatron or in radiation therapy? What is the nature of the fix? Please link to some relevant distributions of the before and after situations.
    - "Added first test-prototype classes of elastic Coulomb scattering". Why was this introduced? What is the intended use?

Comments to the reviewers

- Before a release
  - Like to see the set of patches made available for user testing.
- Features and tools
  - finding overlapping volumes: perhaps G4 should have tolerances for minor overlaps.
  - Improved visualization. For example, cut away 1/4 detector to show innard of the detector with event overlay.

Comments to the reviewers

- Quality control
  - 10% of events fail with G4.8.2. Need more thorough testing before general release.
- Performance
  - It is degraded vs time, this trend is maintained since too much and it seems impossible to stop.
- Any new feature should be discussed before the implementation among the LHC experiments first and if a general agreement is reached then it should be implemented.
- In a huge experiment as ATLAS any untested feature could generate after implementation possible impredictable delays in the whole programme.

Performance issues

- Latest official release tested was G4.8.2.p01
  - abort rate still high (10%)
- Development tag kindly provided for us to test it, and it seems to completely solve the problem
- Also tested FTFP (upon request by G4 developers)

CPU time per event (kSI2K)

channel	FTFP	QGSP	QGSP_EMV
susy	1726.92	1706.6	992.36
zee	1722.96	1643.07	962.83
zmumu	1237.64	1372.53	806.53
ztautau	1491.83	1322.46	802.98
h4l	1532.86	1563.48	1122.81
MB	468.54	493.98	298.42
jets	1471.21	1331.88	935.51

- Tests done on special 12.5.0 build on 64 bit machine
- Performance similar to QGSP (slightly slower)