Atlas experience with Geant4 performance

Andrea Di Simone
CERN and INFN/CNAF

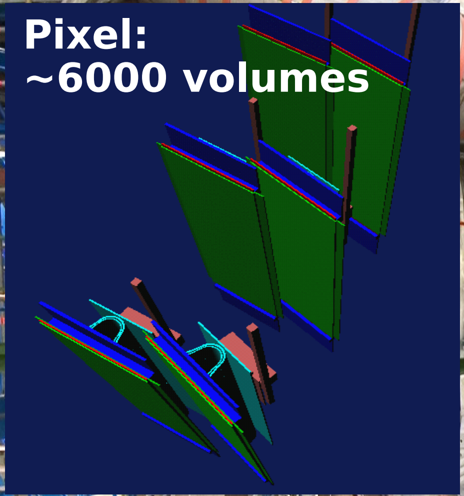
Outline

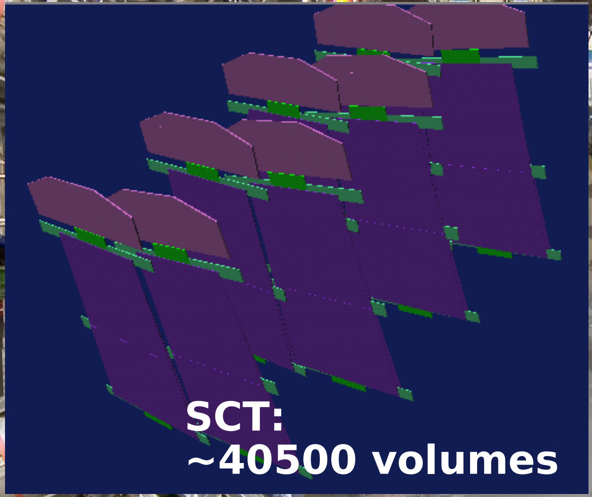
- The ATLAS simulation software
- Computing performance
    - CPU time per event
    - Memory usage @ runtime
    - Eta dependence
    - G4.8 tests

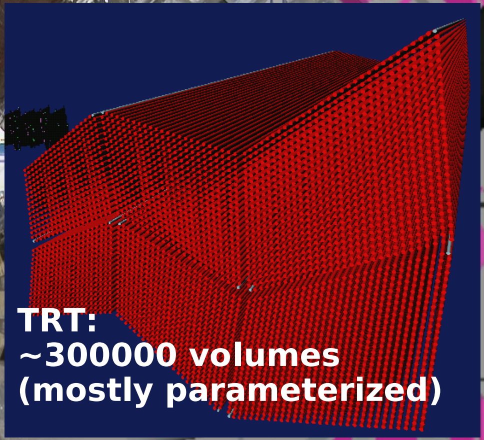
The ATLAS simulation software

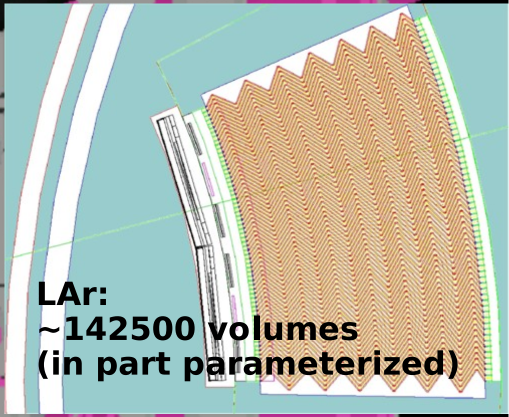
- Since 2002, the old G3 simulation has been replaced by a new G4-based framework as the official ATLAS simulation software
- Fully integrated in the Gaudi-based ATLAS offline framework (Athena)
- Used for the simulation of many different setups:
    - Full ATLAS
    - 2004 Combined test beam
    - Stand alone test beams
    - Cosmic commissioning
    - ...

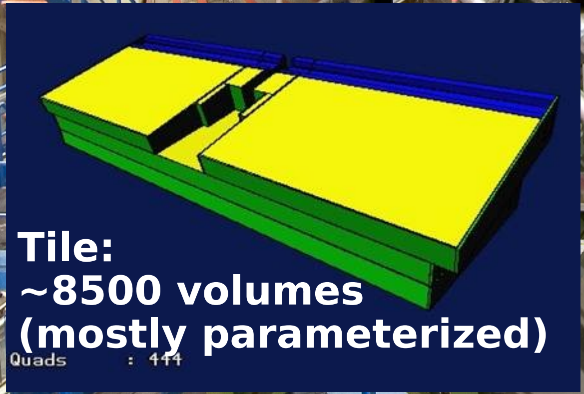
Geometry description

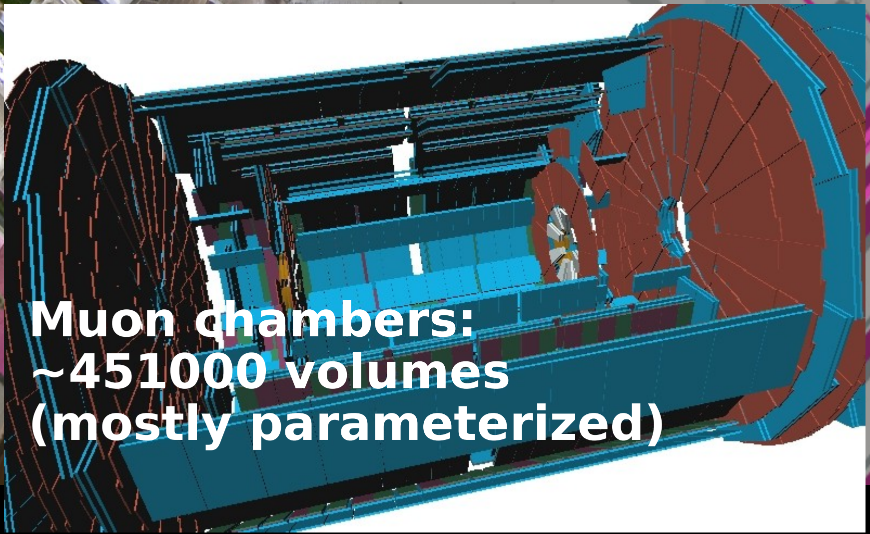


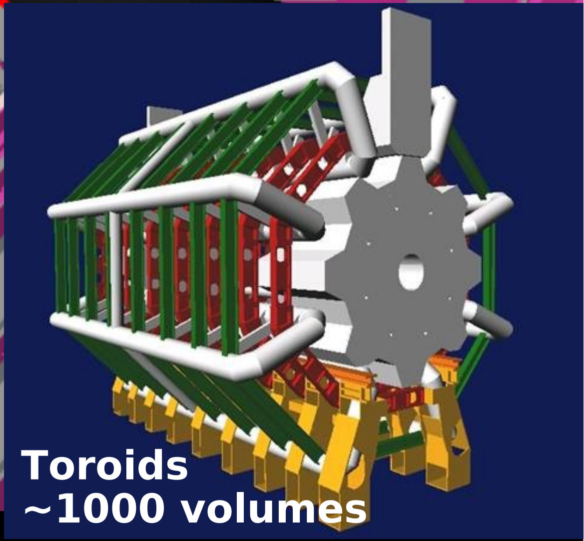












Performance measurements

- ATLAS is a very complex setup:
  - ~10^6 volumes
  - ~200 material/cut couples
- It is therefore a powerful benchmark for G4 robustness/functionality
  - very sensitive to memory issues
  - massive production on the grid allows to spot rare bugs

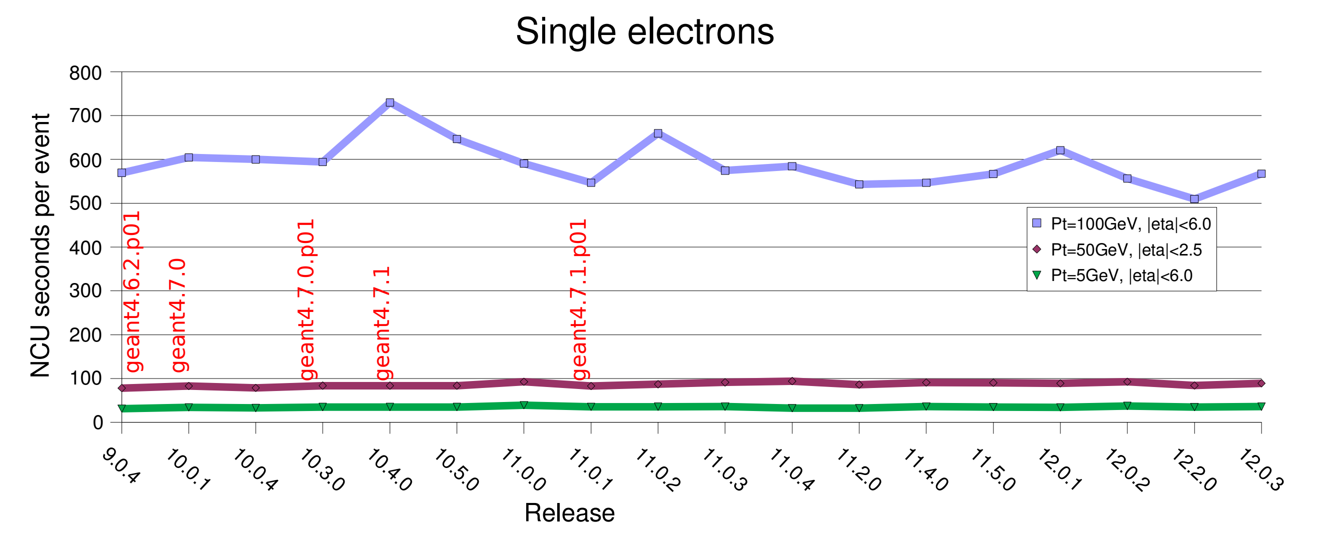
Performance measurements

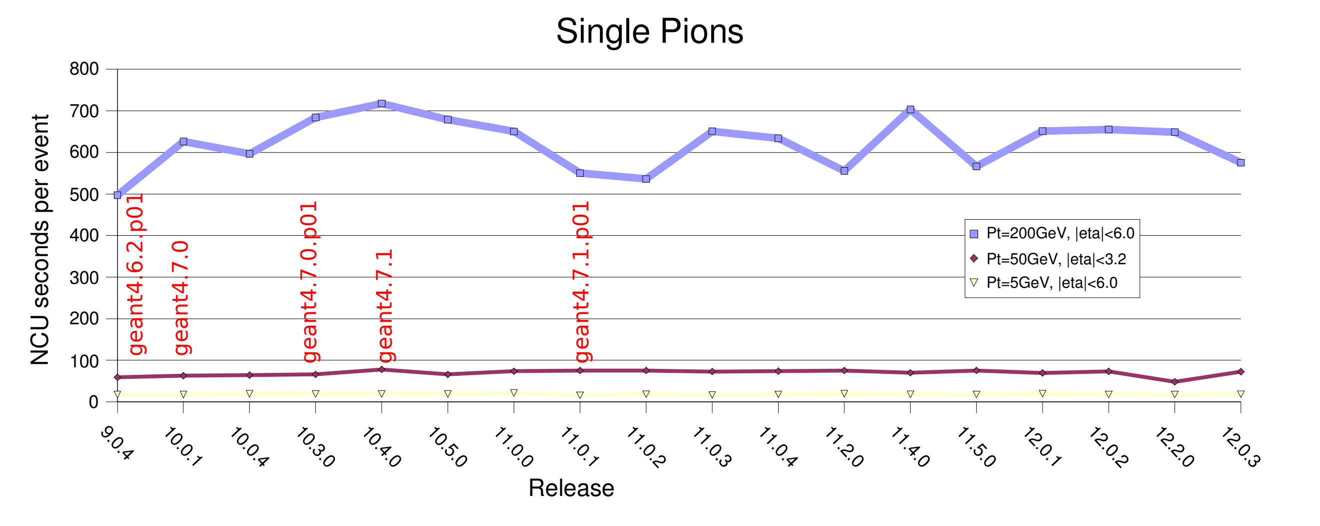
- Two kinds of feedback are given by ATLAS to G4
  - Post-release validation: comes from the tests done at each new release of the AtlasSimulation project, plus feedback from grid production
  - Pre-release validation: after the experience with g4.8 cycle, we volunteered to do some basic functionality tests on g4 release candidates, to help spotting major problems as soon as possible

Computing performance

- Computing performance is kept under continuous monitoring
  - CPU time per event
    - Measured using different samples, both single particles and full physics events
  - Memory at beginning of first event
    - Contributions from each initialization step are measured
  - Memory at end of run
    - Check the absence of leaks

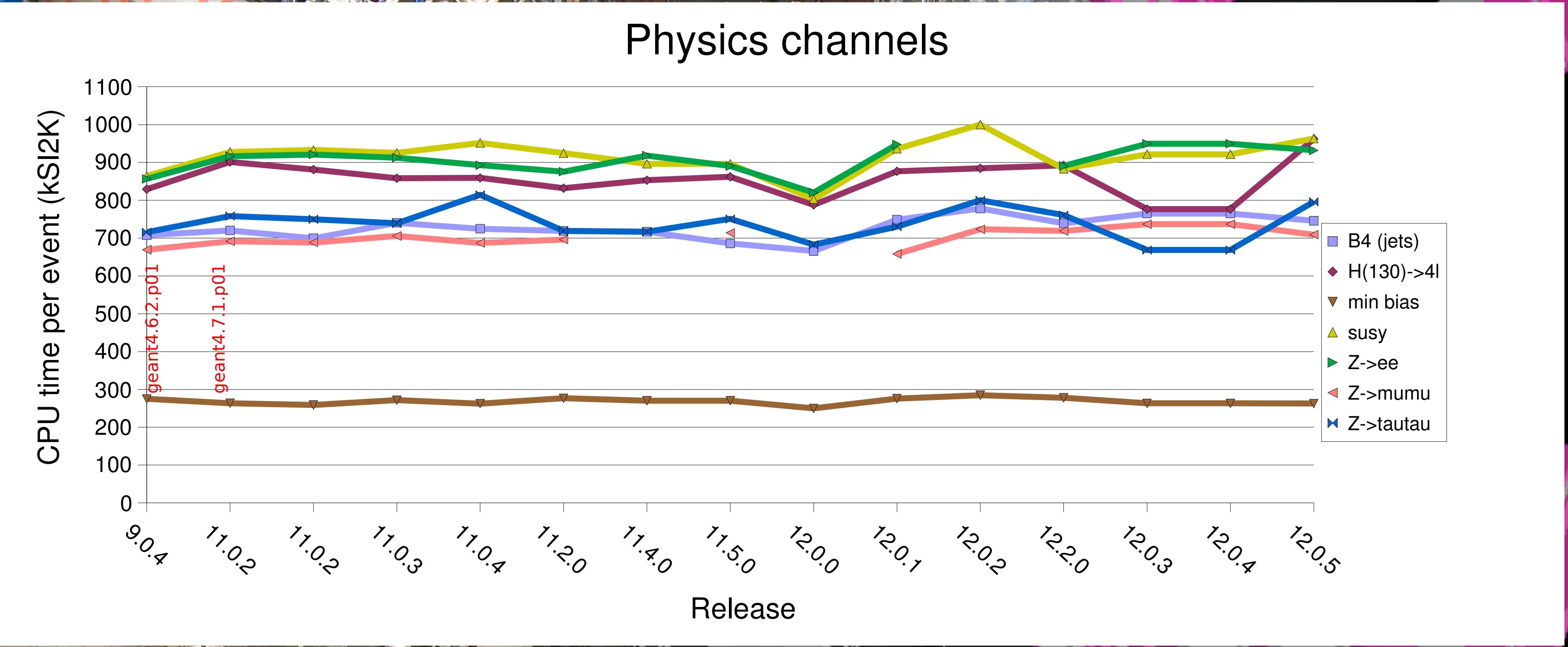
CPU time per event: single particles





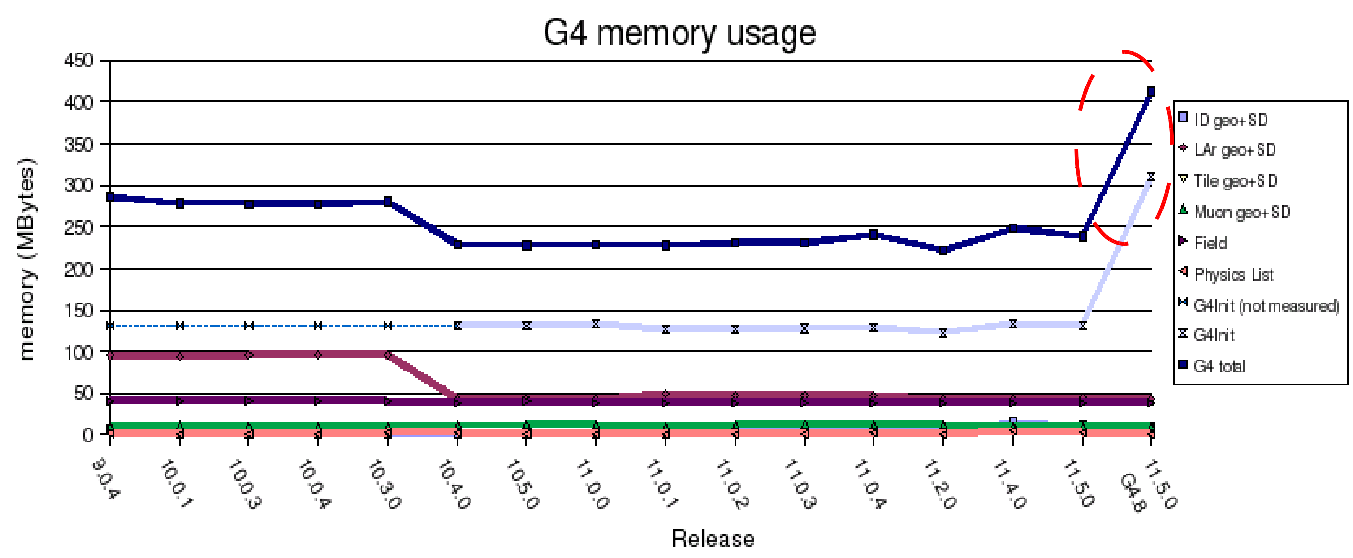
- Single particle performance well under control
  - plots cover the last 2.5 years
- Similar plot available for single muons

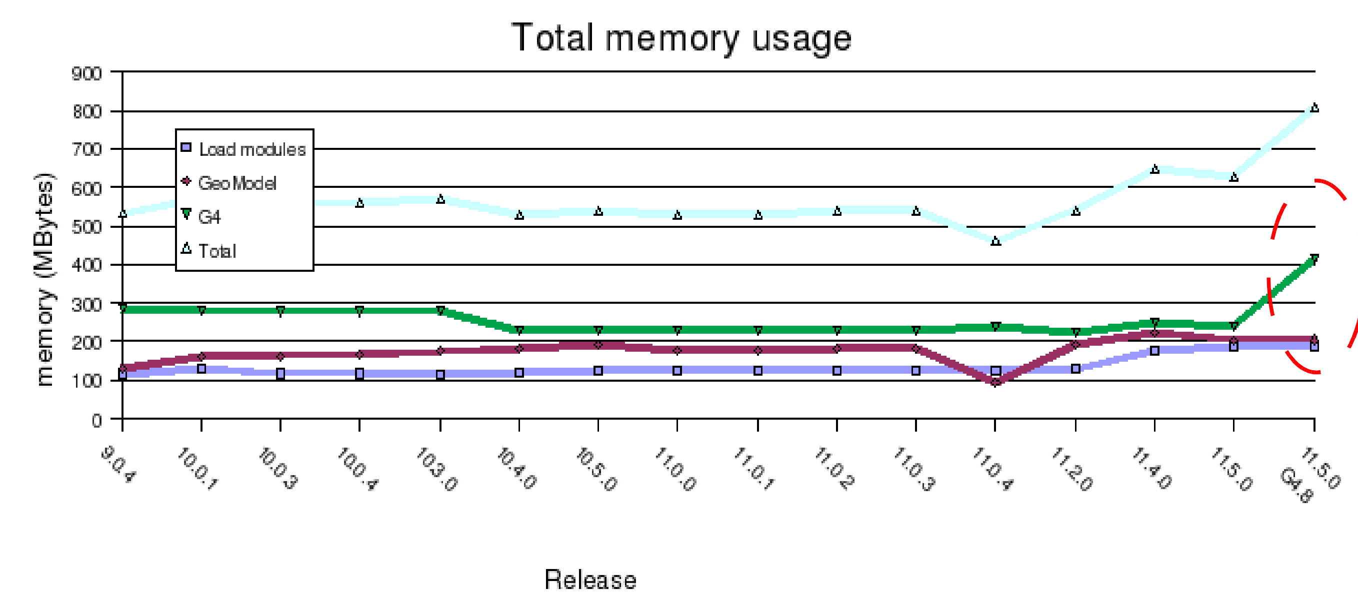
CPU time per event: physics events



- Performance of full physics events is also compatible with the one of release 9.0.4

Memory usage



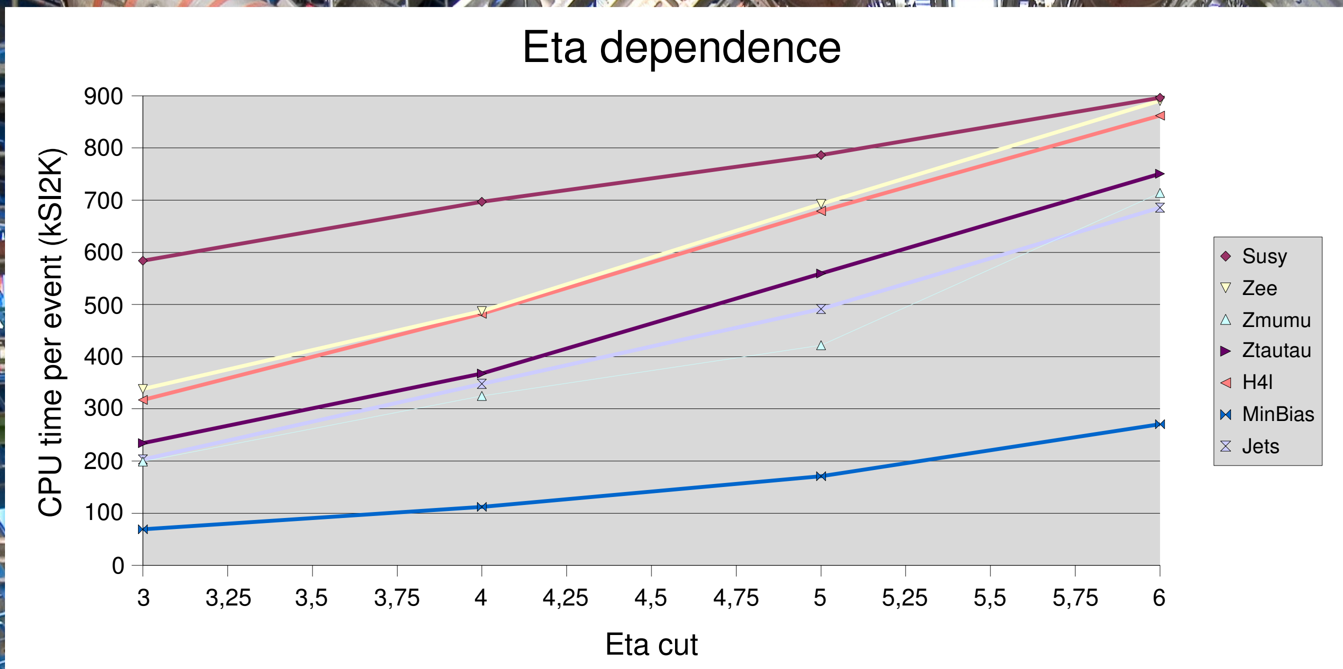


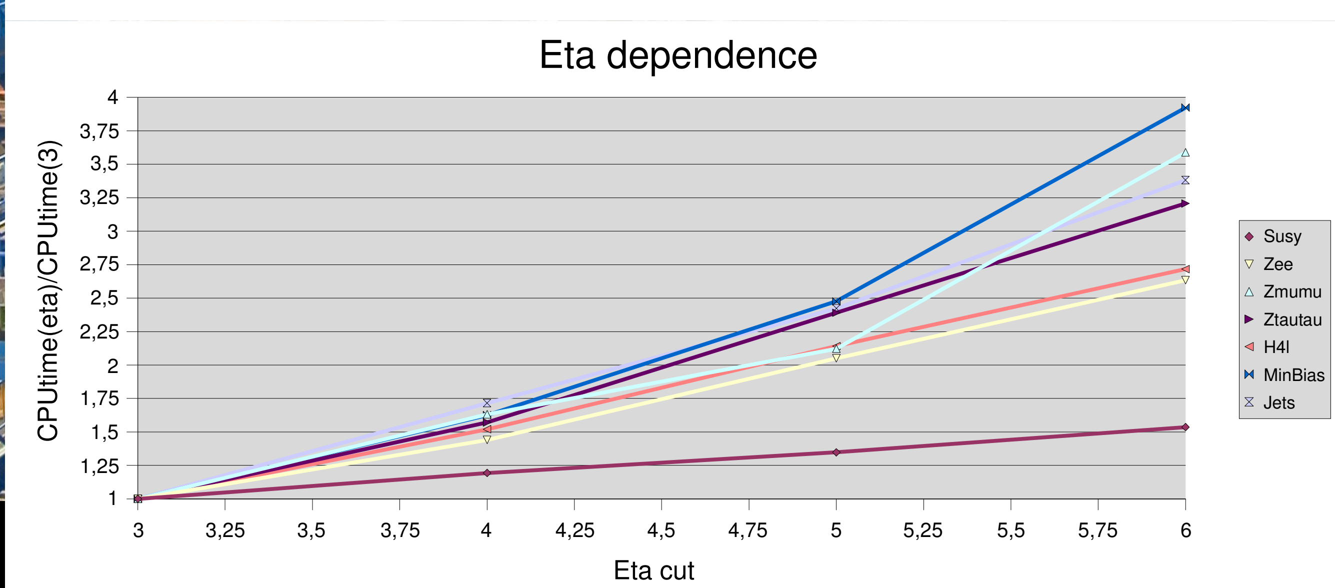
- Memory usage variations observed up to now are not worrying, and are however fully understood
- Small problem with g4.8.0.p01 (now fixed)

Eta dependence

- Users can decide at runtime to limit the simulation only to a certain eta interval
- This has a strong impact on performance
- G4 ATLAS simulation done by default in the eta range (-6,6)
- Older simulation (G3) used to work with a different eta range (-3,3)
- A clear understanding of the eta dependence of the simulation time allows to:
  - Identify the regions where most of the CPU time is being spent
  - Better compare performance with the one by G3

Eta dependence





- Average CPU time per event is measured for different eta intervals using full physics samples.
- As expected, the effect is bigger in minimum bias events. This is clearly visible in the lower plot, where the CPU time is normalized to the time needed in -3<eta<3.

G4.8 tests

- Several tests done in order to understand the impact on computing performance of the new msc implementation
- Basic strategy:
  - build the same ATLAS simulation software twice, using G4.7.1.p01 and G4.8
- G4.8.0 was tested with several different configurations:
  - Default: with the new msc and ATLAS standard cuts
    - 30um in LAr, 1mm elsewhere
  - Special cuts: new msc and 1mm cut for all volumes
  - Msc71: plugging in g4.8.0 the msc implementation from g4.7.1
  - Nsl: same as “Default” but inhibiting the step limitation by the msc

G4.8.0 tests

CPU time per event (kSI2K)

	G4.7	G4.8	G4.8 1mm	G4.8 msc71	G4.8 nsl
Susy	896,46	2019,66	1690,29		849,62
Zee	890,47	1916,37	1573,31	850,41	760,2
Zmumu	713,76	1369,27	1201,99	642,02	671,32
Ztautau	750,73	1427,59	1253,83	743,69	677,34
H4l	862,15	1788,29	1429,86	884,07	783,73
Jets	685,8	1442,15	1364,75	701,05	753,6

Susy	2,25	1,89		0,95
Zee	2,15	1,77	0,96	0,85
Zmumu	1,92	1,68	0,9	0,94
Ztautau	1,9	1,67	0,99	0,9
H4l	2,07	1,66	1,03	0,91
Jets	2,1	1,99	1,02	1,1

- Timing results for full physical events are shown, as obtained in the different configurations. Ratios wrt G4.7.1 timing results are reported as well.

G4.8.0 tests

- The increase in time was really due only to the new msc implementation, and it was connected with the step limitation
- Setting all the production cuts to 1mm did not help in reducing the processing time
- In order to have timing results compatible with the ones we used to have with g4.7, choices were:
  - use the old msc implementation
  - use the new msc implementation, switching off the step limitation

More on G4.8

- Tests repeated systematically at each G4 release
  - several run time problems were spotted:

release	QGSP	QGSP_EMV
G4.8.0	ok	ok
G4.8.1	~5% events aborted	ok
G4.8.2	~76% events aborted	ok
G4.8.3	ok	exception (8/28 jobs)
G4.8.3.p01	ok	ok

  - problems due to different modifications of the G4 code, which gave unexpected results when applied to the ATLAS setup:
    - clashes in our geometry description
    - very strict settings for the tracking in magnetic field

G4.8.3.p01 results

CPUtime per event (kSI2K)

physics channels	G4.7 QGSP_GN	G4.8 QGSP_EMV	G4.8 QGSP	G4.8 QGSP 1mm	G4.8 QGSP_BERT
susy	921,64	1123,82	1956,42	1560,52	2594,16
Zee	949,58	1107,58	1944,05	1546,41	2432,79
Ztautau	668,64	831,19	1429,71	1361,49	2129,3
H(130)4l	776,72	1067,55	1793,55	1468,79	2334,59
MB	263,35	332,66	584,2	509,29	805,98
jets	765,06	920,77	1480,34	1328,76	1957,11

Ratios

physics channels	QGSP_EMV/ QGSP_GN	QGSP/ QGSP_EMV	QGSP1mm/ QGSP_GN	QGSP_BERT/ QGSP	QGSP_BERT/ QGSP_EMV
susy	1,22	1,74	1,69	1,33	2,31
Zee	1,17	1,76	1,63	1,25	2,2
Ztautau	1,24	1,72	2,04	1,49	2,56
H(130)4l	1,37	1,68	1,89	1,3	2,19
MB	1,26	1,76	1,93	1,38	2,42
jets	1,2	1,61	1,74	1,32	2,13

- No particular runtime problems found: both QGSP and QGSP_EMV ran fine
- Performance comparison with g4.7.1
- First look at performance of QGSP_BERT
- Increase of QGSP_EMV wrt g4.7.1

G4.8 - comments

- Computing performance of QGSP_EMV slowly but constantly deteriorating during g4.8 release cycle
  - QGSP_EMV on g4.8.3 is about 20% slower than QGSP_GN on g4.7.1.p01
    - this is a major problem for production
    - G4.8.0 + old msc was performing exactly like g4.7.1, so this must have been introduced in later releases
    - effect seems to be related to hadronic physics (more evident in single pions than single electrons)

Conclusions

- The computing performance of the ATLAS simulation software is continuously monitored:
  - Since more than 2.5 years, both CPUtime per event and memory usage remained constant, in spite of the addition of new features
- Tests with G4.8 show that, unfortunately, we will have a significant time increase, even with QGSP_EMV
- First G4.9 tests did not show any major run time problem, and computing performances similar to g4.8.3.p01
- During G4.8 release cycle, many lessons learnt for what concerns validation of a new G4 release, both from our side and from G4's
- Start to apply what we have learnt to the new G4.9 series