

ATLAS Offline Computing Plans

Rolf Seuster (TRIUMF)
5th Sept. 2014

Trigger, Online and Offline
Computing Workshop

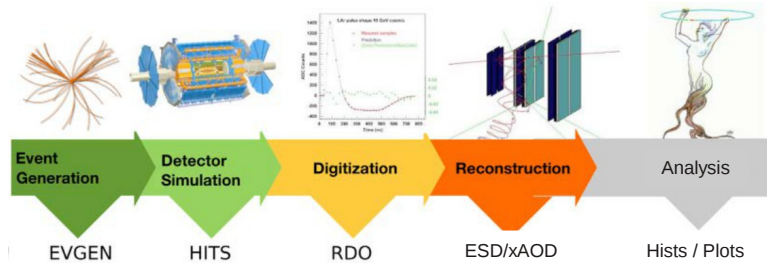
Introduction

- some of the work during LS1 included some preparation for further improvements of our software
- in the following I will explain briefly what has been done during LS1 and what is still missing and will be addressed during run-2 and LS2
- required changes for LS3 are less clear as we don't know how the computing evolves
- presented here are plans and ideas – which should be tested and modified or disgarded if needed !

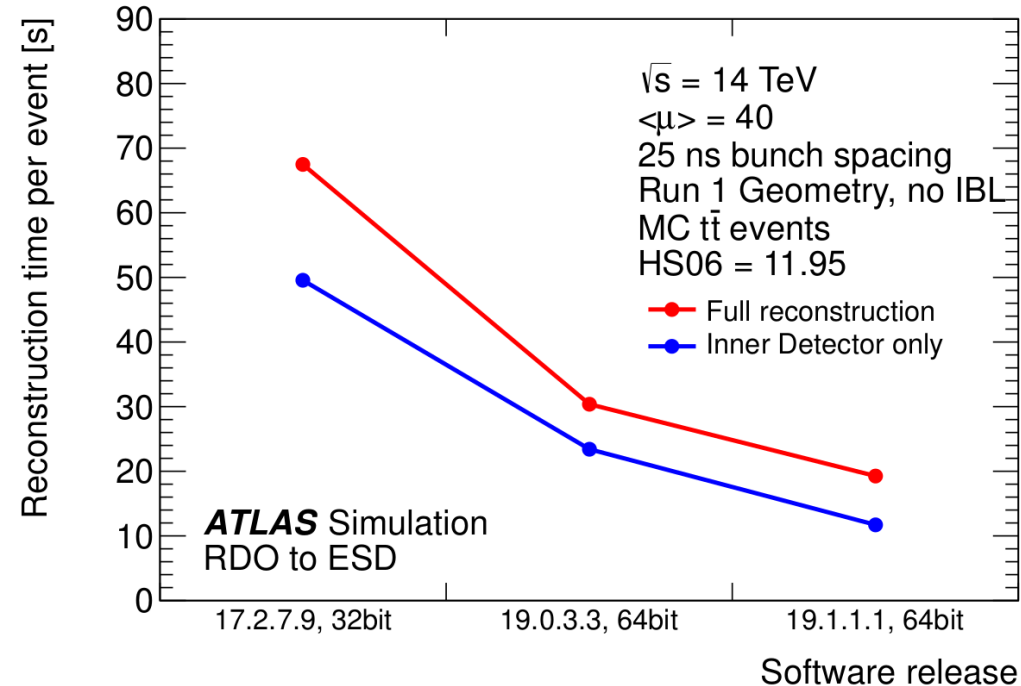
Current Status of ATLAS Software

work done during LS1 – Reconstruction

- Software Speedup by factor of 3x achieved
 - allows to safely operate at 1kHz EF output rate and reconstruct in timely matter at Tier0
- without compromising quality of reco output !
- to save memory will run athenaMP on multi core for sim up to reco and primary xAODs



- analysis is typically much too IO bound



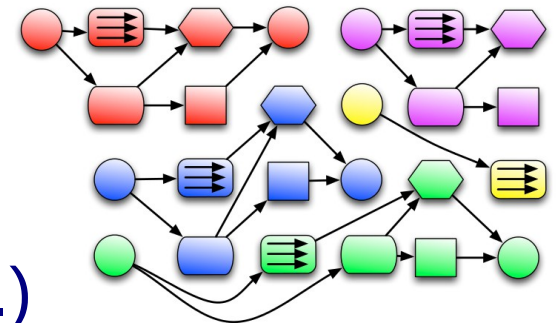
Current Status of ATLAS Software

work done during LS1 – Reconstruction, cont'd

- Speedup achieved by
 - algorithmic improvements + cleanups
 - better seeding strategies for high pileup
 - SW improvements (Eigen Matrix library)
 - 'external' technical improvements
 - different math library (from Intel)
 - switch to 64 bit
 - at cost of more memory
 - new compilers
 - will also allow for new compiler flags
 - SLC6 instead of SLC5

Reconstruction tasks for post-LS1

- athenaMP won't be enough to survive ManyCore area
- need threadsafe framework, e.g. GaudiHive
 - probably need to rework some algorithms to decouple unnecessary dependencies (e.g. split up some CP algos to allow for better concurrency)
 - might be beneficial to combine athenaMP and GaudiHive for intermediate term, until all of ATLAS software is threadsafe or removed from codebase -> to be evaluated
- need also to develop thread safe and threaded versions of long running algorithms and its tools
 - > profiling at high mu will show us which (and likely confirm our questimates)
- -> more details in Markus' talk (RoI, etc.)

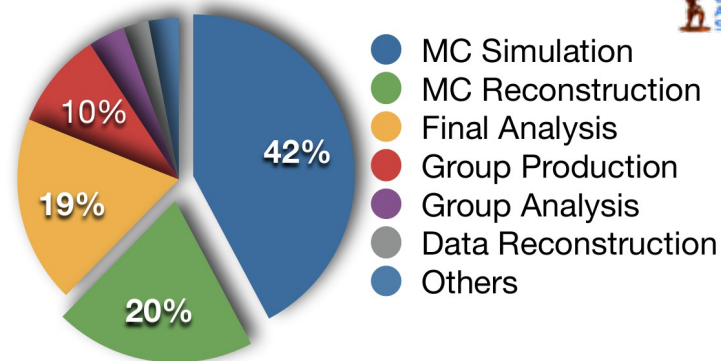


Current Status of ATLAS Software

work done during LS1 – Simulation

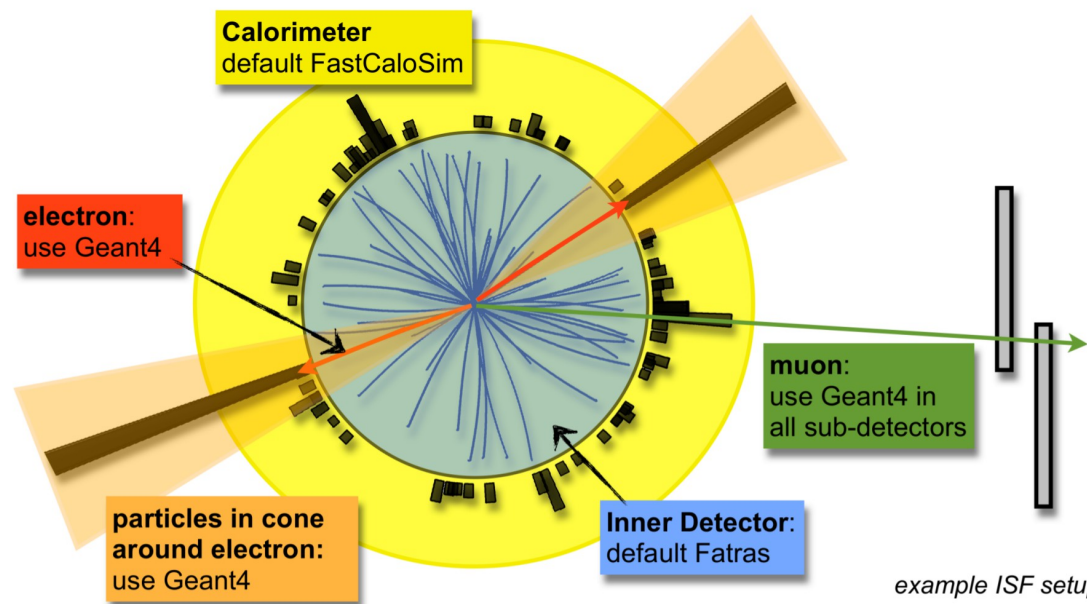
- Simulation (+MC reco) biggest CPU consumer on Grid
- ISF has great potential for huge savings of CPU time !
- Lots of effort went into tuning FastSim for Calorimeter and Fatras for Traching

GRID CPU Consumption



Tracker	Calo	Muons	speedup
full	fast	full	~20
fast	fast	fast/full	>100
RoI guided fast/full			~100

'just' need to establish its usage with physics groups !



example ISF setup

Simulation tasks for post-LS1

- threading of simulation code
 - simulation code has only few interaction points with rest of reco, trigger code -> threading can be implemented almost independently once framework is threadsafe
- merge ISF with GeantMT
 - possibly very first ATLAS application using multi threading in large scale in production !
- test offloading G4 workload to accelerators (e.g. Xeon Phi)
 - optimal usage of opportunistic resources like HPC Tianhe-2 already has, SuperMuc will have XeonPhi, etc... 4 out of Top10 of Top500 have such accelerators ...
 - how will e.g. Knights Landing change the picture ?

Current Status of ATLAS Software

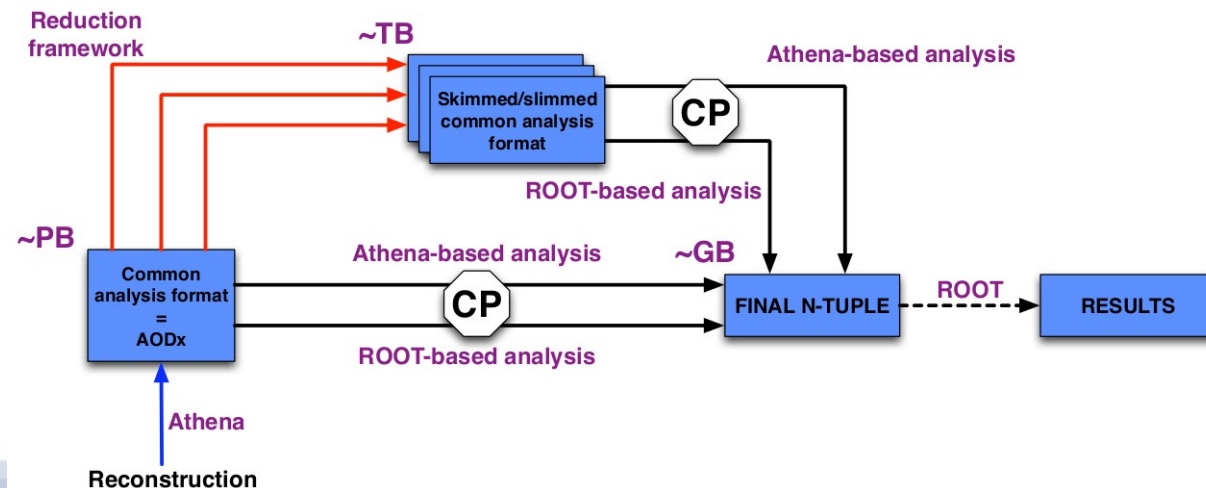
work done during LS1 – Simulation -> Reconstruction

- Integrated Simulation Framework (ISF)
requires equally fast digitisation and reconstruction – naturally at cost of precision
 - both being worked on and steady progress
 - first versions should be ready by 2015
 - ideal for e.g. SUSY grid searches where intermediate files are disgarded

Current Status of ATLAS Software

work done during LS1 - Analysis

- New Analysis Framework:
 - new root and athena readable EDM: xAOD
 - huge, intermediate D^nPD ntuples obsolete
 - reduction framework reduces primary xAOD into manageable chunks used in analyses
 - optimizes disk sizes and time data reaches user



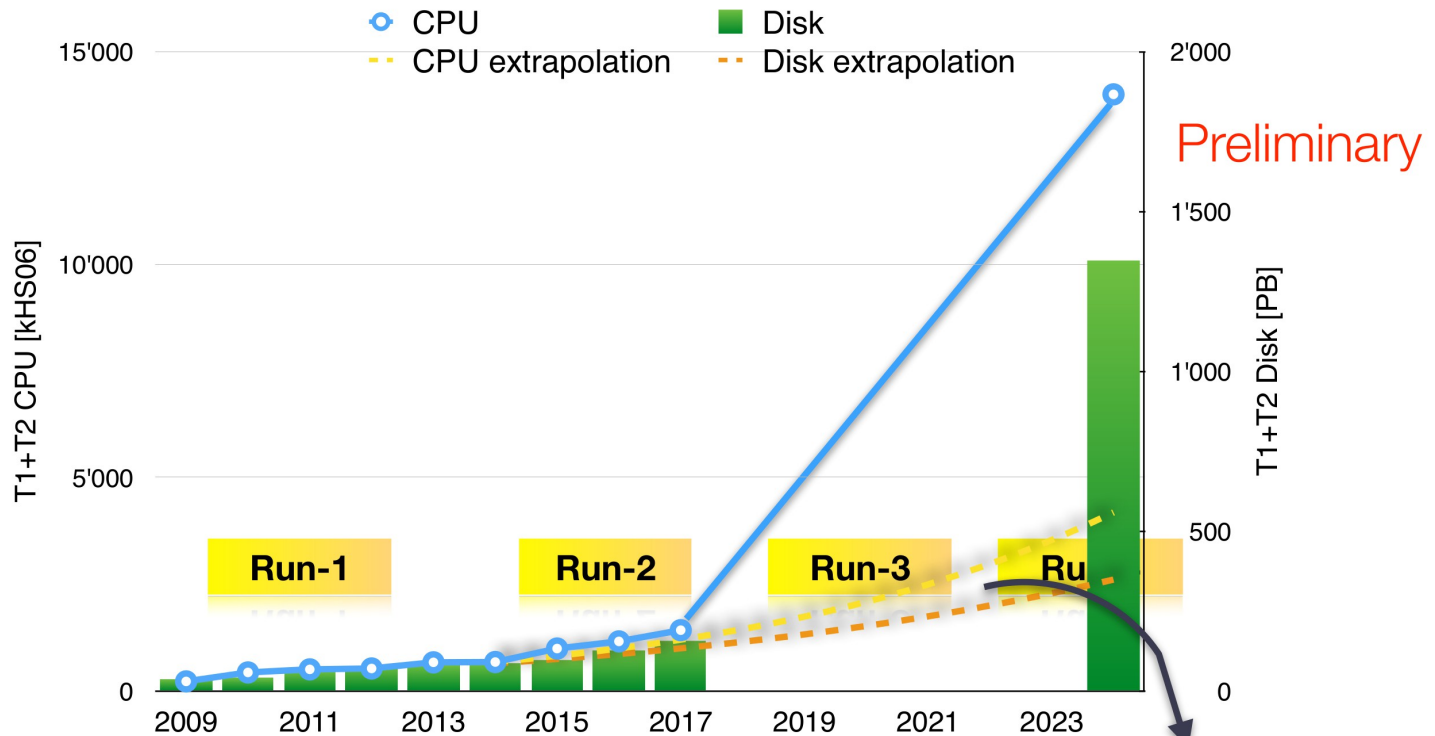
Disk Usage / Sizes

- problem in Run-1: ntuples were often 1:1 copy of AODs, but root- readable; removing duplicated copies frees disk space for important new data -> one of main reasons for xAOD and new analysis framework
- resources will be even tighter with higher lumi/EF output rate -> need much more MC (2/5 billion planned for run-2 for full/fast sim – how much is needed for run-3 /4 ?)
 - need to rethink what to store in xAOD files, and take hit on what can be done with it ... (in run-1 we were able to apply some important fixes after AOD production due to 'redundant' information in AODs ...)
- another application for fast reco/fast sim:
 - evt directly to user ntuple to avoid storing large intermediate files never being looked at again

Projection to Run-4

Run-4 (with 2014 performances)

ATLAS resource needs at T1s & T2s



Extrapolation from 2014 (Moore law)



Eric Lançon on behalf of the ATLAS collaboration

-> need to worry about disk and CPU usage for HL-LHC as well as access to disk (IO and capacity) !

Current Status of ATLAS Computing

work done during LS1 – Grid Software

- currently commissioning new data placement and production system
 - typical lifetime > 5 years or so
 - expect new systems not before run-4 or so
 - run-2 & run-3 similar in requirements for both only HL-LHC changes picture dramatically
 - need to learn from new system as well as need to know new requirements
 - e.g. how to deal with accelerators whole nodes scheduling should help
 - future HW/SW technologies changes might offer completely new solutions

further Tasks for post-LS1

in all domains towards run-3 (and run-4)

- better utilize Auto Vectorization beyond Eigen via algorithmic and EDM changes, incl. using special libraries/packages/tools like vdt where beneficial
- threading was already mentioned
- modularize software such that offloading of specialized tasks becomes easier
 - dedicated EventServer for I/O running on same/different machine (enforcing all IO goes through the framework ...)
 - more speculative:
offload CPU intensive tasks to accelerators including parts of reconstruction (mostly tracking), file (de-) compression (on smallish GPUs/FPGAs !), Geant4 simulation, ideally these accelerators run on same or some other machine (incl. additional CPU cores for e.g. 'big.LITTLE' architectures)

Processor Landscape

- certainly the CPU landscape now more colourful again:
 - ARM 32bit + SIMD
 - + smallish GPUs (e.g. NVidia Jetson)
 - ARM 64bit + SIMD + ??
 - Xeon Phi (a.k.a. Knights Corner / Landing)
 - Knights Landing due in 2015
 - ~70 Atom processors, 4HT, 16 GB fast memory, 384 GB 'slow' DDR4
 - desktop CPUs with GPUs on die
 - Xeon with FPGAs on die (for 'big customers')



<https://communities.intel.com/community/itpeernetwork/datastack/blog/2014/06/18/disrupting-the-data-center-to-create-the-digital-services-economy>

- are we 'big' customers ?
- more for TDAQ, not so much for Grid

Summary and Conclusion

- LS1 work successful: 3x speedup in Reco
 - layed fundaments for further code improvments
- post-LS1 work will continue here
 - Vectorizing: new EDM and Eigen help, still further code changes required
 - Threading: little work done here so far, due to required work on new EDM and Eigen
- And: keep your eyes open for the unexpected !
 - e.g. 'The Machine' from HP in ~2019 it 'revolutionizes' Computing :)

