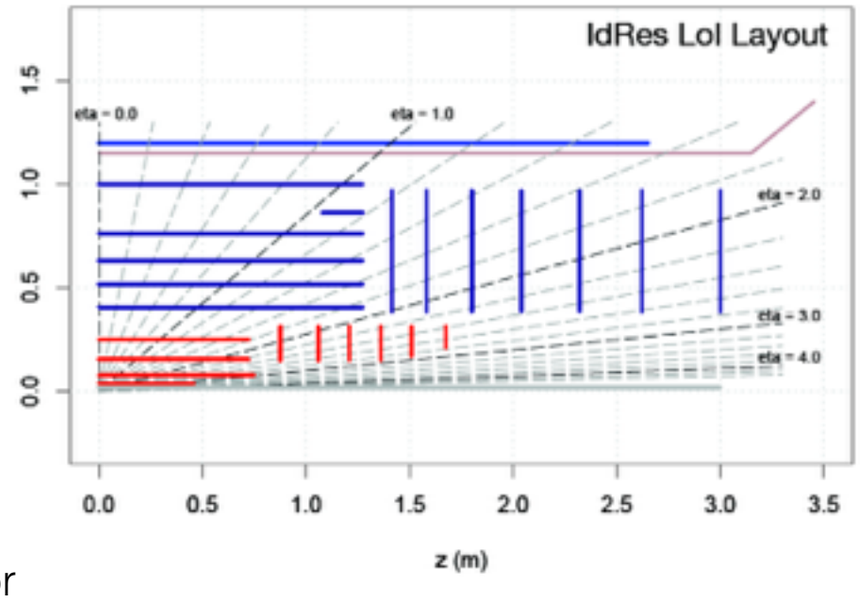
Upgrade Software and Computing for ATLAS

Graeme Stewart

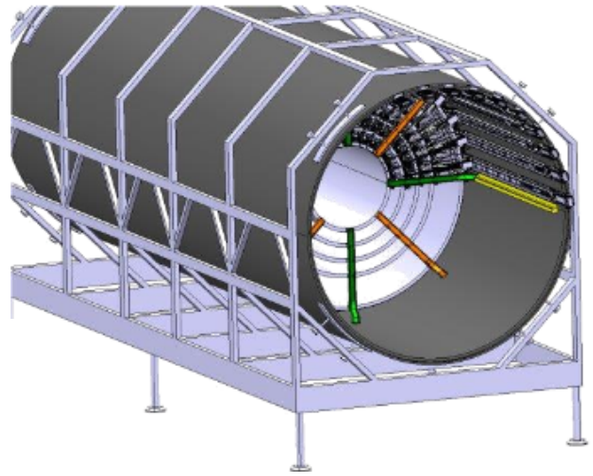
University of Glasgow | School of Physics & Astronomy

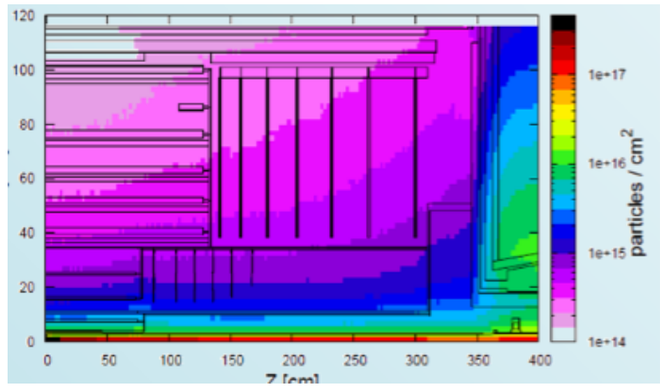
2015-04-12

ATLAS Upgrade Timeline

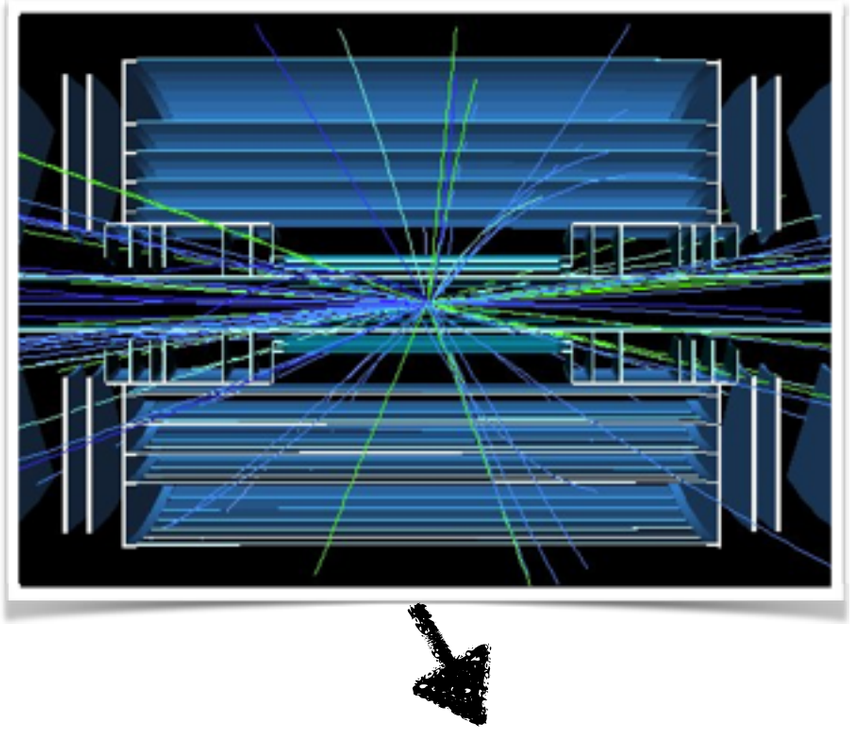


- The major upgrade to ATLAS will come in Phase II (2023-24) and be installed for *High Luminosity LHC* (2025)
    - Also known as Long Shutdown 3, followed by Run4
- Major detector upgrade component is the replacement of the current inner detector with the *ITk*
    - All silicon detector with inner pixel barrels and outer strip detectors
        - Much finer grained detector to keep occupancy down, even at high pileup
        - Radiation hard to survive 10 years of high luminosity
- Readout from L0 trigger into the ATLAS *track trigger*
    - Make tracking information available even at Level 1 (1MHz)
    - Greatly improves the discrimination power of the trigger at μ=140 (levelled luminosity) but anticipate good performance for peaks of μ=200
        - Essential to maintain physics performance of ATLAS
    - Level 1 feeds into the High Level Trigger (HLT)
        - Output rate from the HLT in 5-10kHz range (x5-10 what we have today)

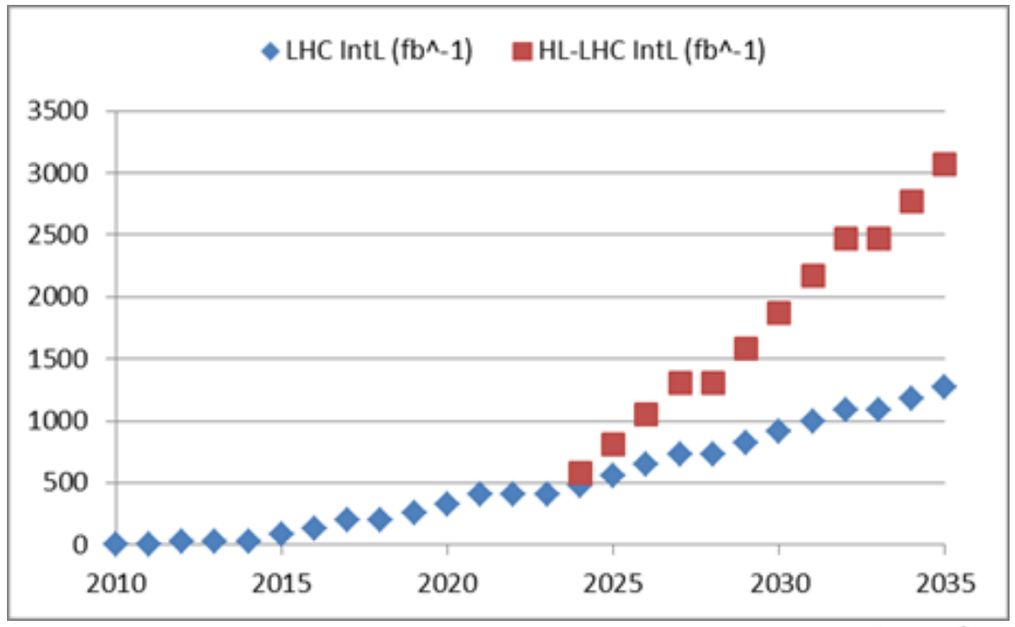




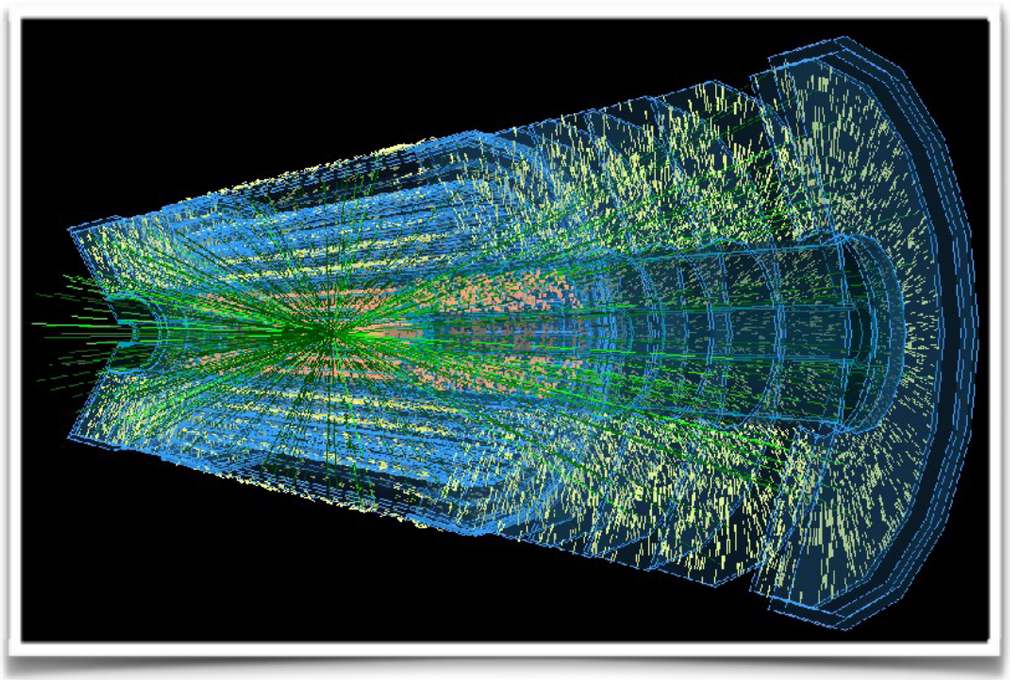
The Computing Challenge



x



Rende Steerenberg, CERN



Event Complexity x Rate = Computing Challenge

- Reconstruction event complexity is naively μ! (factorial)
- Rate increases from 1kHz to 5-10kHz

Scale Numbers

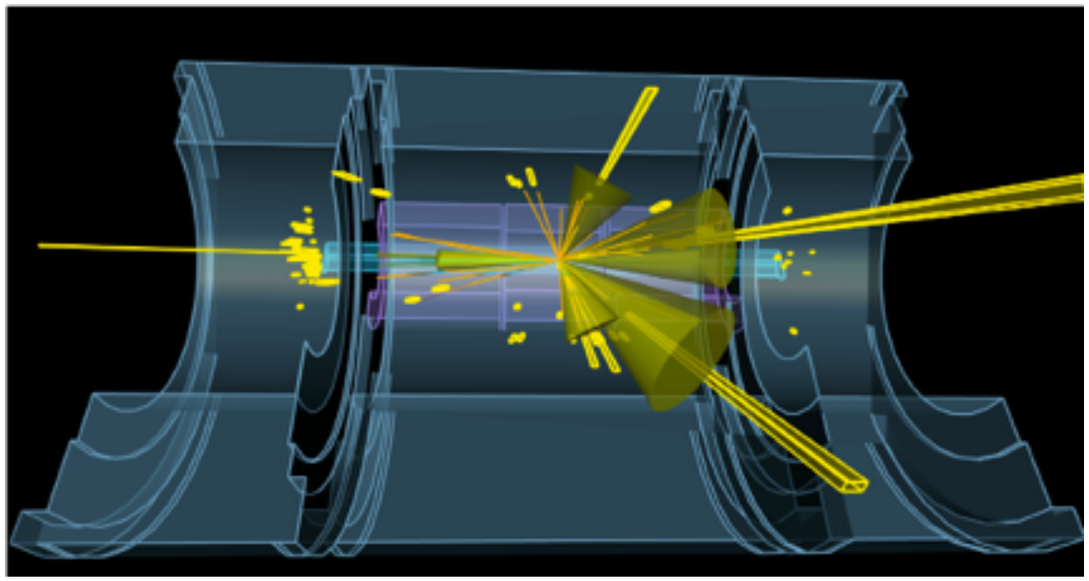
	HLT Output	Events per year	RAW per Event	RAW data per year
Run1	600Hz	3.6B	0.7MB	2.5PB
Run2	1kHz	5B	1.0MB	6PB
Run3	1kHz	5B	1.2MB	7.2PB
Run4	5kHz	25B	2.5MB	75PB

- Assuming an (optimistic) physics beam time of 6M seconds per year
  - However, this is the target for HL-LHC to collect 300fb^{-1} per year
- What will the relationship between RAW data and derived data be?

How hard does it get?

- Event Generation
  - Not intrinsically harder at high luminosities, however better generators and studying rare processes will mean using more cycles; volume increase to scale with events
- Simulation
  - Main scaling of simulation per event is with energy (so ~constant in Runs 2, 3, 4, …); however, more data needs more simulation to accompany it, so volume increases
- Digitisation
  - More or less linear with pile-up as background minimum bias events are layer on top of signal events; more simulation → more digitisation
- Reconstruction
  - Definitely *very hard* at high pile up; scaling is naively $\mu!$ (factorial) for tracking; certainly the biggest challenge faced in software; combines with volume increases
- Analysis
  - Most likely linear with data volumes, but analysis can already be i/o bound; thus i/o becomes a serious problem; need to optimise across huge range of workloads

Event Generation



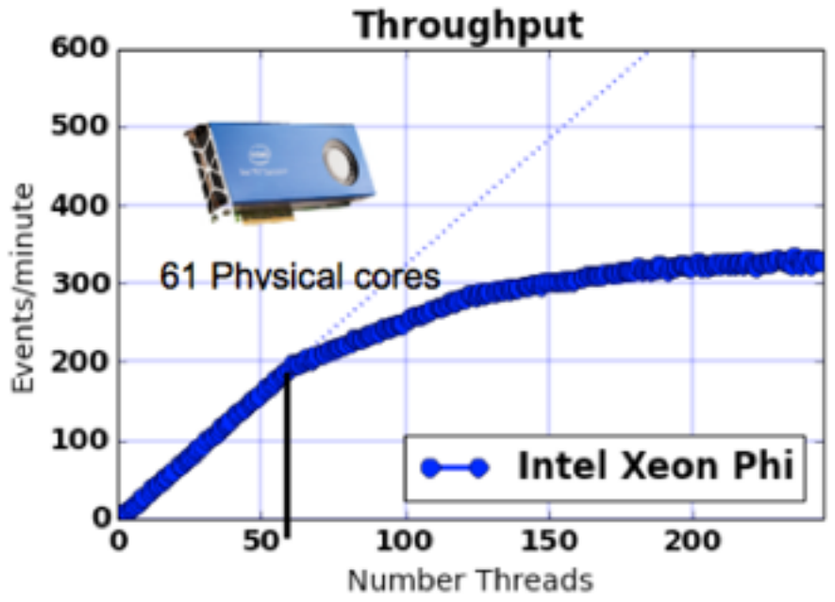
Alpgen+Pythia8 Z+5jet event produced on Mira

Taylor Childers, Tom LeCompte, Tom Uram, Argonne

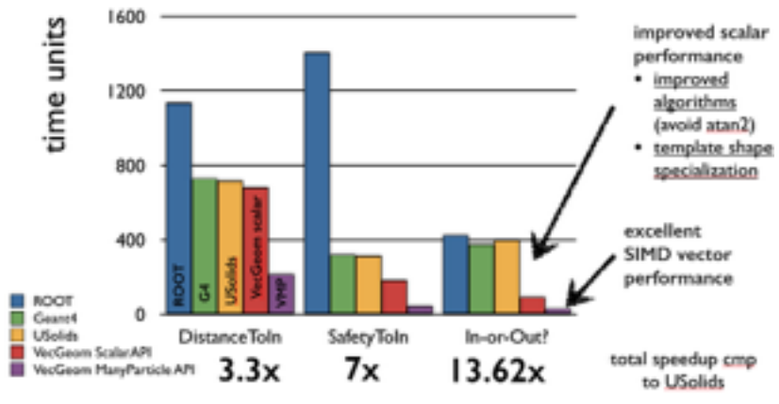
- Event generators are stand-alone pieces of code
- CPU intensive
- Good fit for novel architectures and HPC facilities
  - Relatively easy code to run massively parallel (MPI)
- Optimisations certainly possible and feedback into the ‘vanilla’ x86_86 version
  - Got a x6 improvement in Alpgen speed on Mira (PowerPC HPC at Argonne)
  - Generator authors often keen to work with us to help here

Simulation

- Full Geant 4 simulation of a detector as complex as ATLAS is CPU intensive — can be 1000s per event
- Clearly important to optimise here
  - Ongoing work with the Geant team to improve efficiency
- Simulation is a good target application for many core systems
  - Memory footprint nowhere near that of reconstruction
  - Many concurrency opportunities from independence of particles
- Geant V targets vectoriseable parts of the process — improvements should feed back into G4



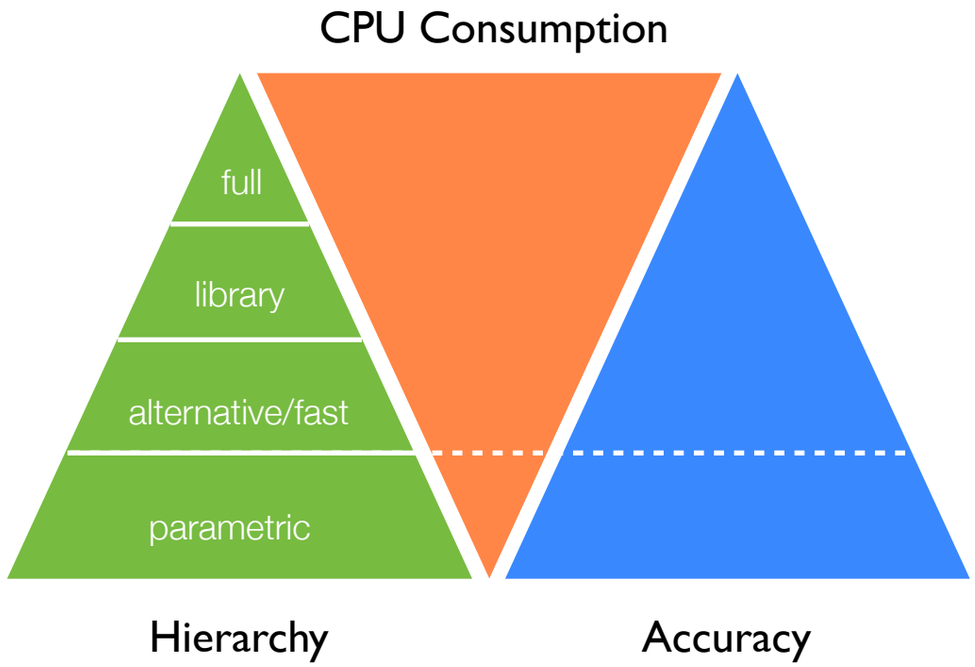
Andrea Dotti, SLAC



Sandro Wenzel, CERN

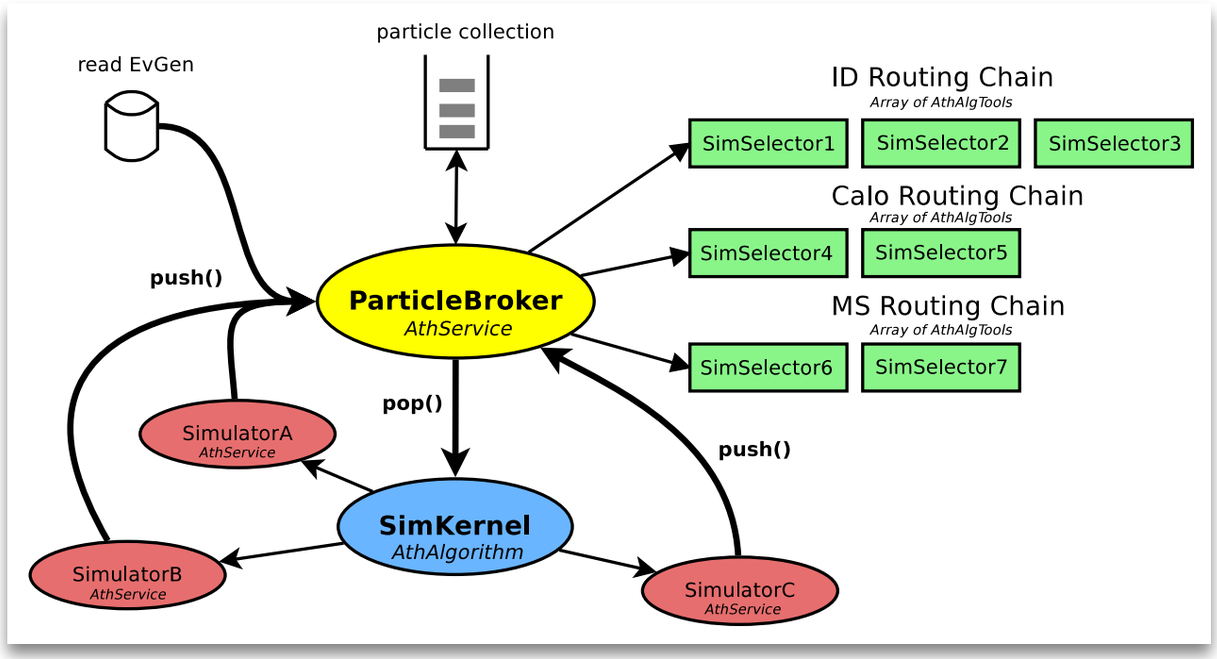
Fast Simulation

- Various flavours of fast simulation available
  - Frozen showers, AtlFast-2, parametric ...
  - Fast track/muon simulation Fatras
- Question is what is the best compromise between CPU consumption and accuracy ?
- So far fast simulation used for
  - Very forward showers in otherwise full sim for large productions of specific samples
    - e.g. SUSY parameter scans
  - Phase-2 upgrade studies
- *Physics validation of fast simulation really takes time and effort though*

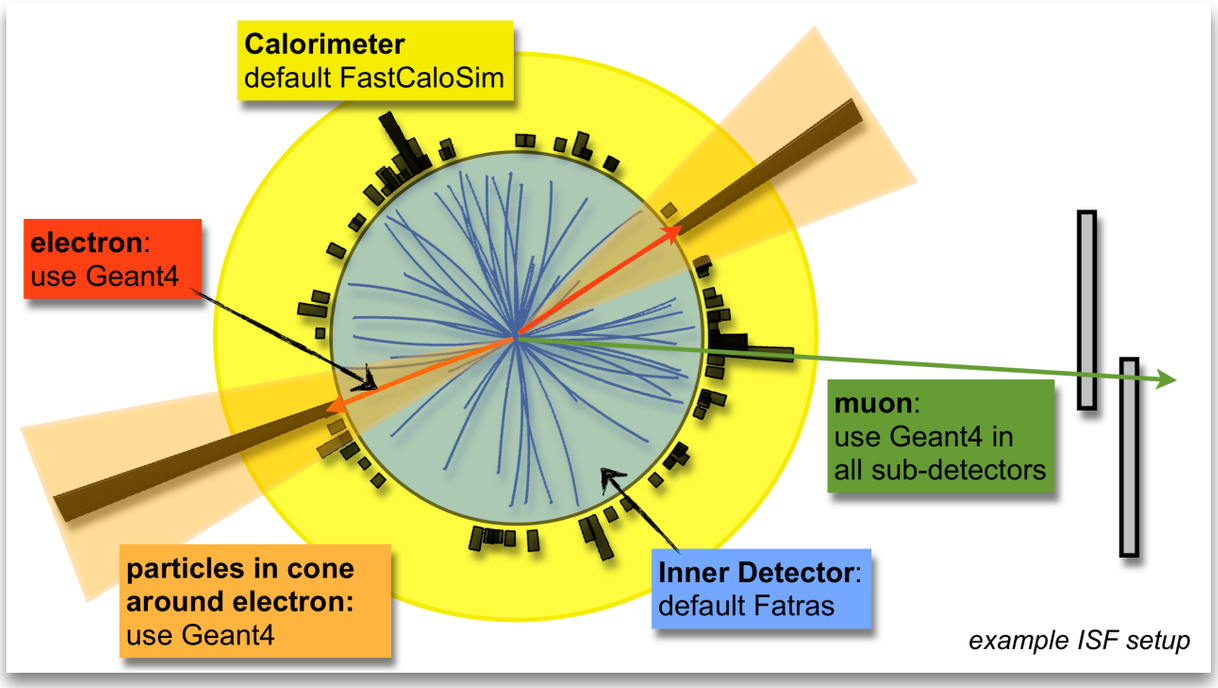


Integrated Simulation Framework

- Single framework for simulation
- Simulation engines act like services
- Choose engine based on particle type and region of interest
- Mix simulation types within a single event
- Full potential realised when combined with fast digitisation and reconstruction



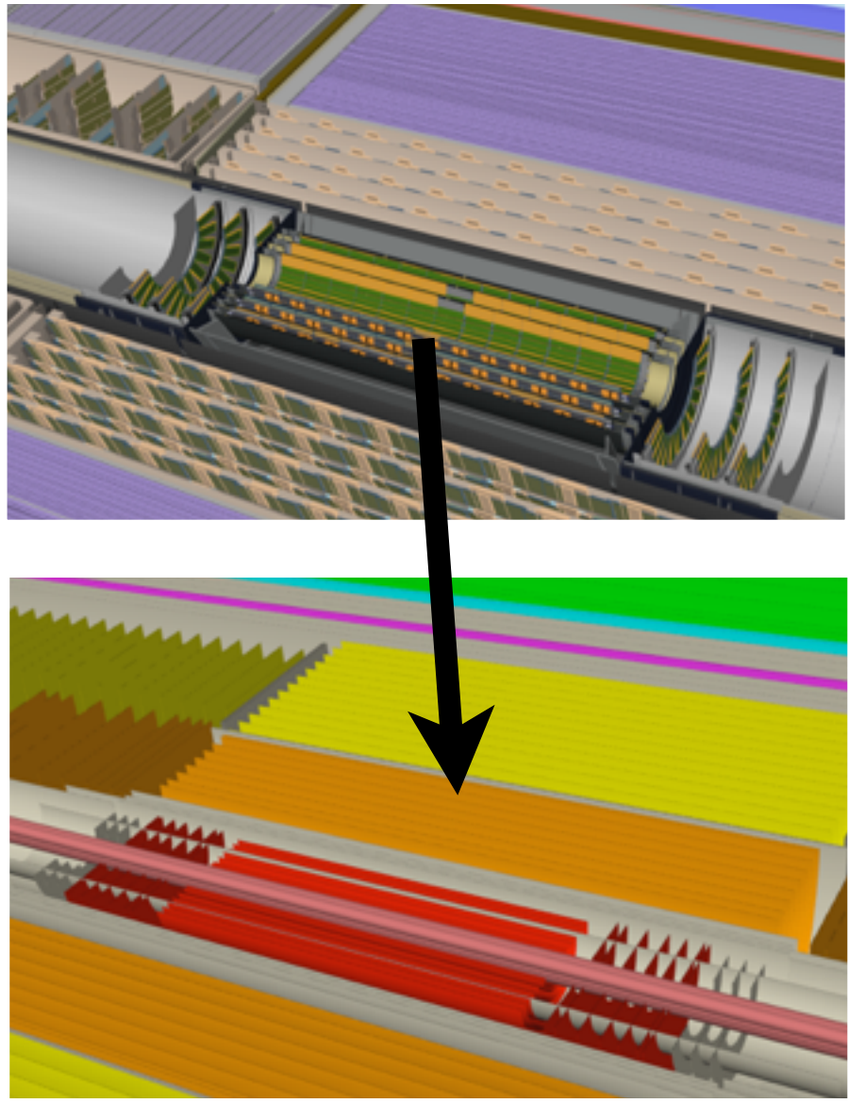
Tracker	Calo.	Muons	speedup
full	fast	full	~20
fast	fast	fast/full	>100
RoI guided fast/full			~100



Fast Simulation: FATRAS

- ATLAS has 2 geometry systems (not special)
  - Full model used in Geant4 with 4.8M placed volumes
- Reconstruction model for fast tracking
  - reduced complexity
  - material projected onto surfaces
- Fast extrapolation engine
  - embedded navigation replaces voxialization
- Fatras simulation engine
  - re-uses track reconstruction infrastructure
  - combined with particle stack and fast physics processes
  - optionally: fast digitisation codes

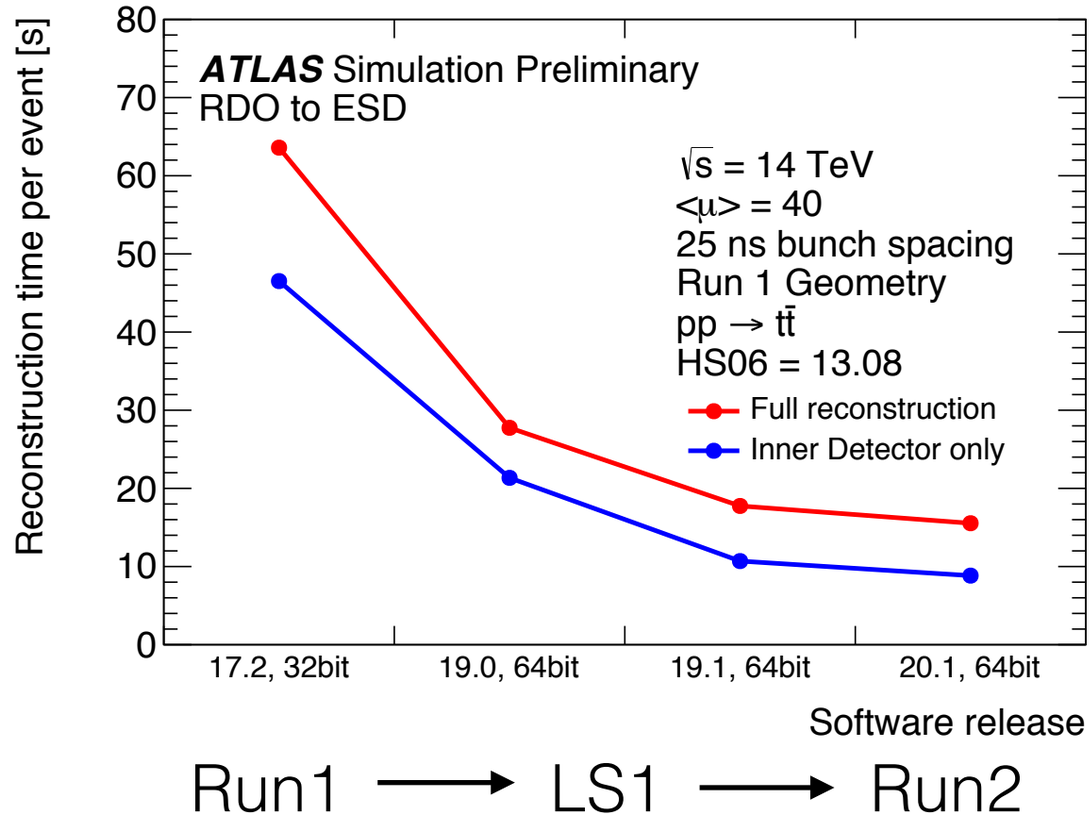
ATLAS	G4	tracking	ratio
crossed volumes in tracker	474	95	**x5**
time in SI2K sec	19.1	2.3	**x8.4**

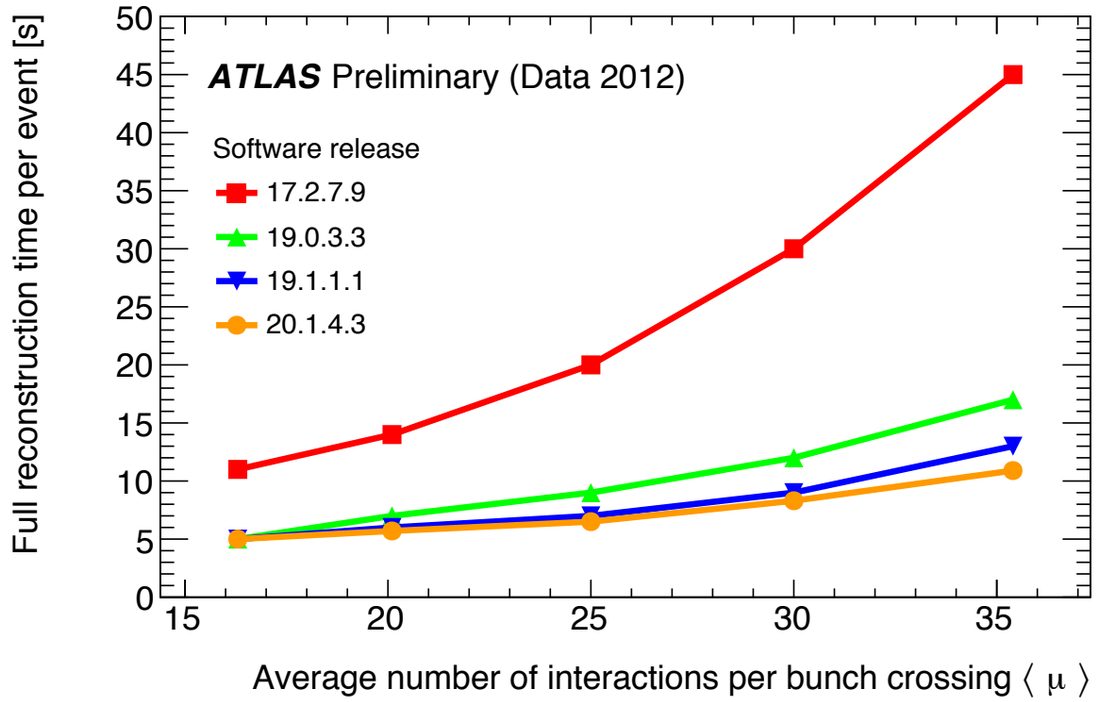


Andy Salzburger, Markus Elsing

Tracking

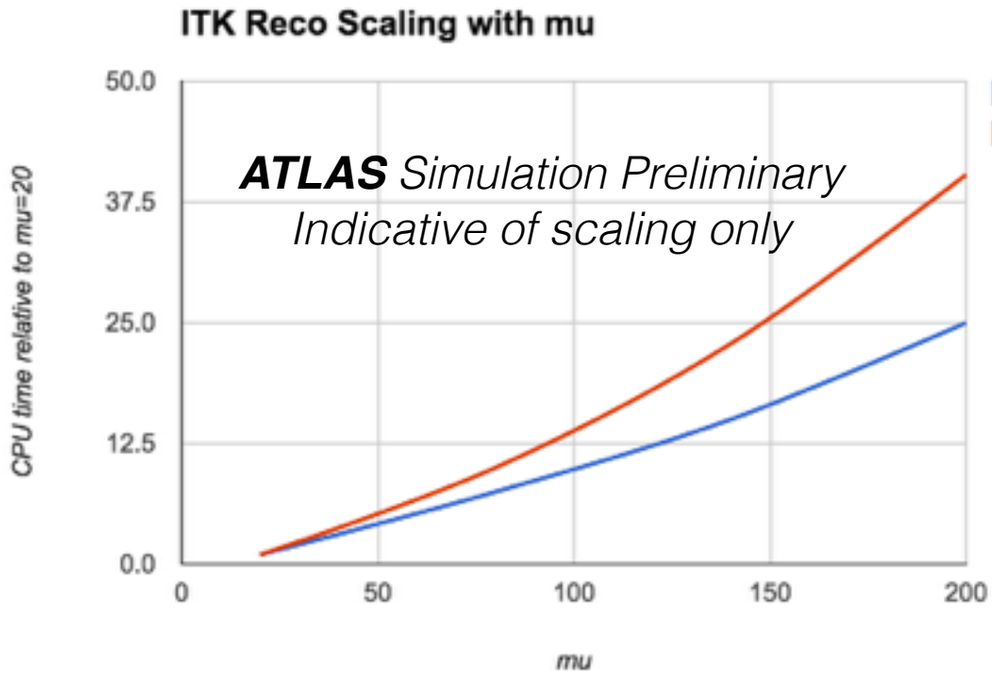
- Efficient tracking is a battle against combinatorics (death to μ!)
- Highly *serial* implementation to reject poor track candidates early and minimise wasted cycles
- Great improvements in Run2 already
  - x4 improvement in overall reconstruction speed, mainly from tracking
  - Greatly improved track seeding strategy has also improved physics quality!
- Note performance on reconstructing high pileup Run1 data is improved





High Luminosity Tracking

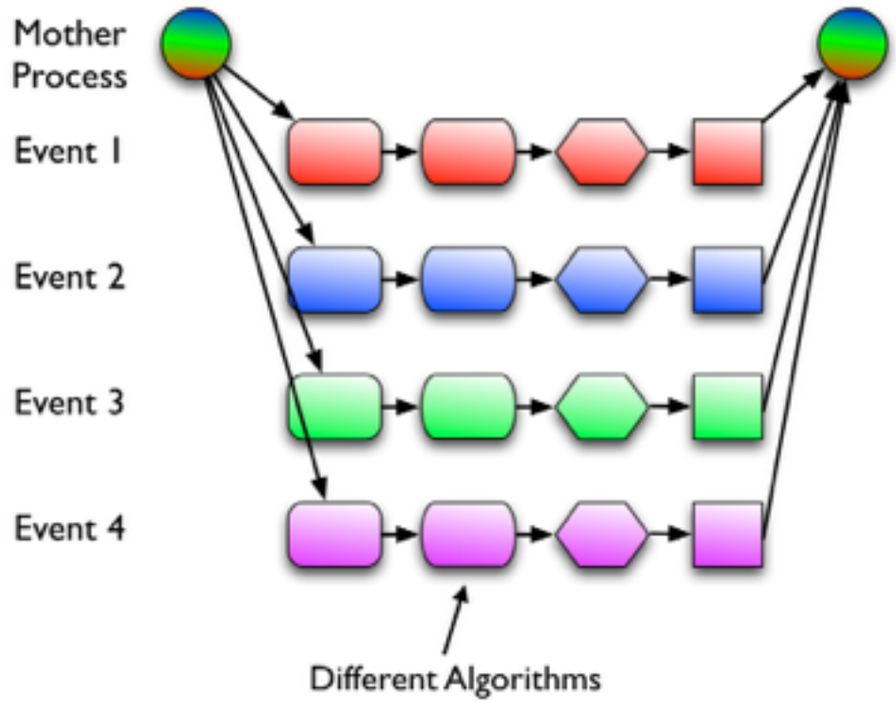
- Current tracking performance at about μ^2
  - x30 at high luminosity
  - Overall x150 when rate is considered
    - *Even the wildest optimist could not foresee this much improvement in CPUs in the next 10 years*
- Serial nature of current strategy limits concurrency
  - Especially disadvantageous on many core systems
  - Can throw more events in, but memory limitations start to kick in (even in a threaded framework)
  - May need to sacrifice some serial efficiencies to being more cores into play
    - Maximise throughput is the goal



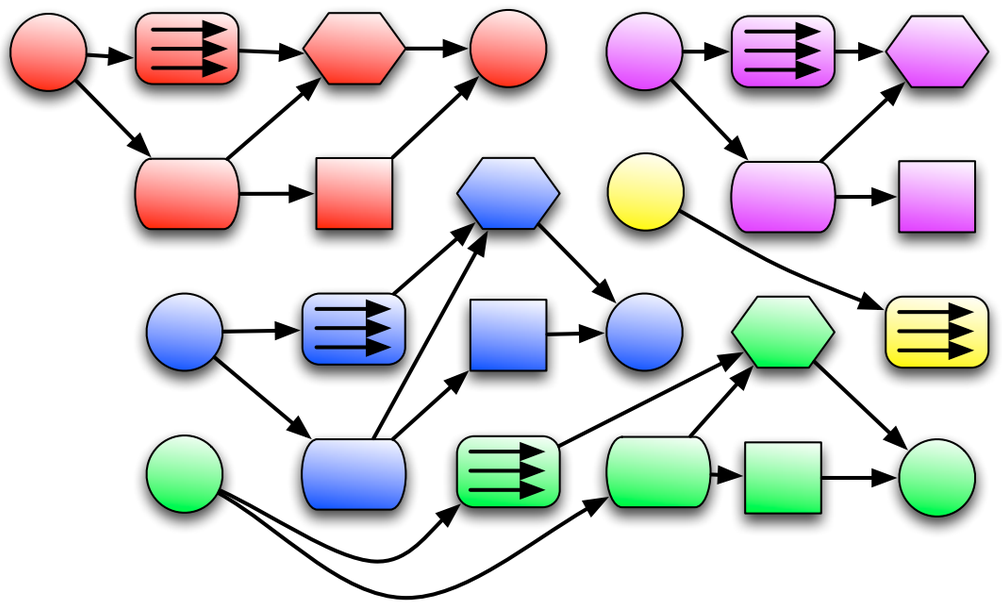
- For monte-calo truth tracking helps a lot
- For data seeding with track trigger/online information might help
- Many interesting new ideas to improve further
  - Deep machine learning and pattern recognition

New Framework: GaudiHive

- Memory constraints, especially on non-Xeon server architectures make reducing memory footprint imperative
    - High luminosity and hard tracking conditions only increase this pressure
- Need to move to a multi-threading framework (beyond AthenaMP)
    - Memory savings can be huge as all heap memory is shared
    - However, a more difficult programming model as threads can interfere with each other: data races and deadlocks
- Development to introduce parallelism into the Gaudi framework used by ATLAS and LHCb
- Take advantage of parallelism between algorithms and across multiple events
- Scheduler is data flow driver, but control flows can also be given (important for online)

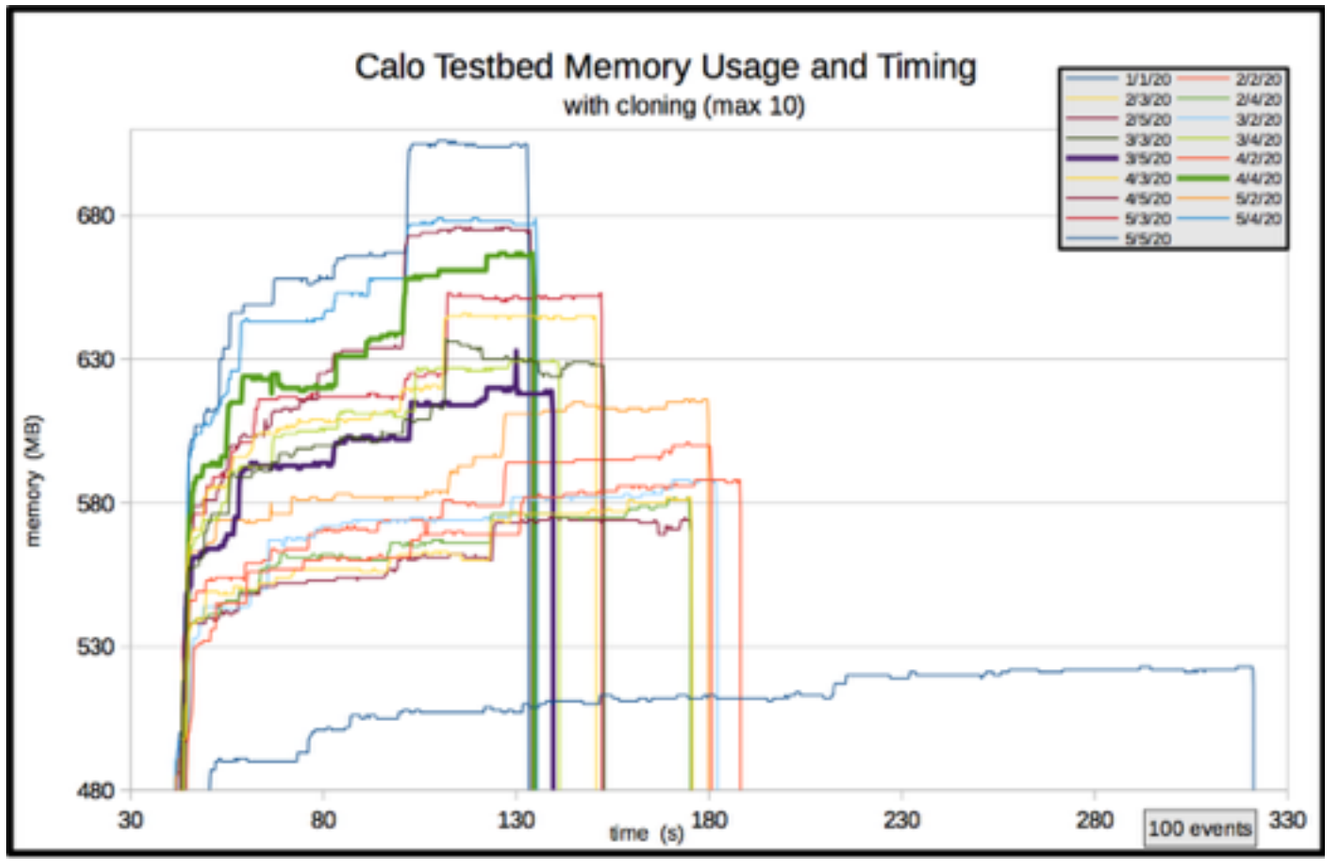


Run2 AthenaMP multi-processing: Each worker uses a separate process, but read-only memory pages are shared



Run3 multi-threaded reconstruction: Colours represent different events, shapes different algorithms; all one process running multiple threads

CaloHive Scaling Tests



Speedup of x3.3 (limited concurrency in this example) with 4 events in flight

Memory increase of only 28%

- ATLAS went though a requirements exercise for the future framework recently
- Much better understanding now of what the design should be and what services the framework should offer
  - HLT support built in
- Practical demonstrators have provided considerable insight into good design patterns and highlighted many anti-patterns in our code
  - It is not easy to back port concurrency into a framework that has run serial for more than a decade
  - Migration plan will provide an evolutionary path forwards towards Run3
- Developer (re)training is a big issue

Analysis

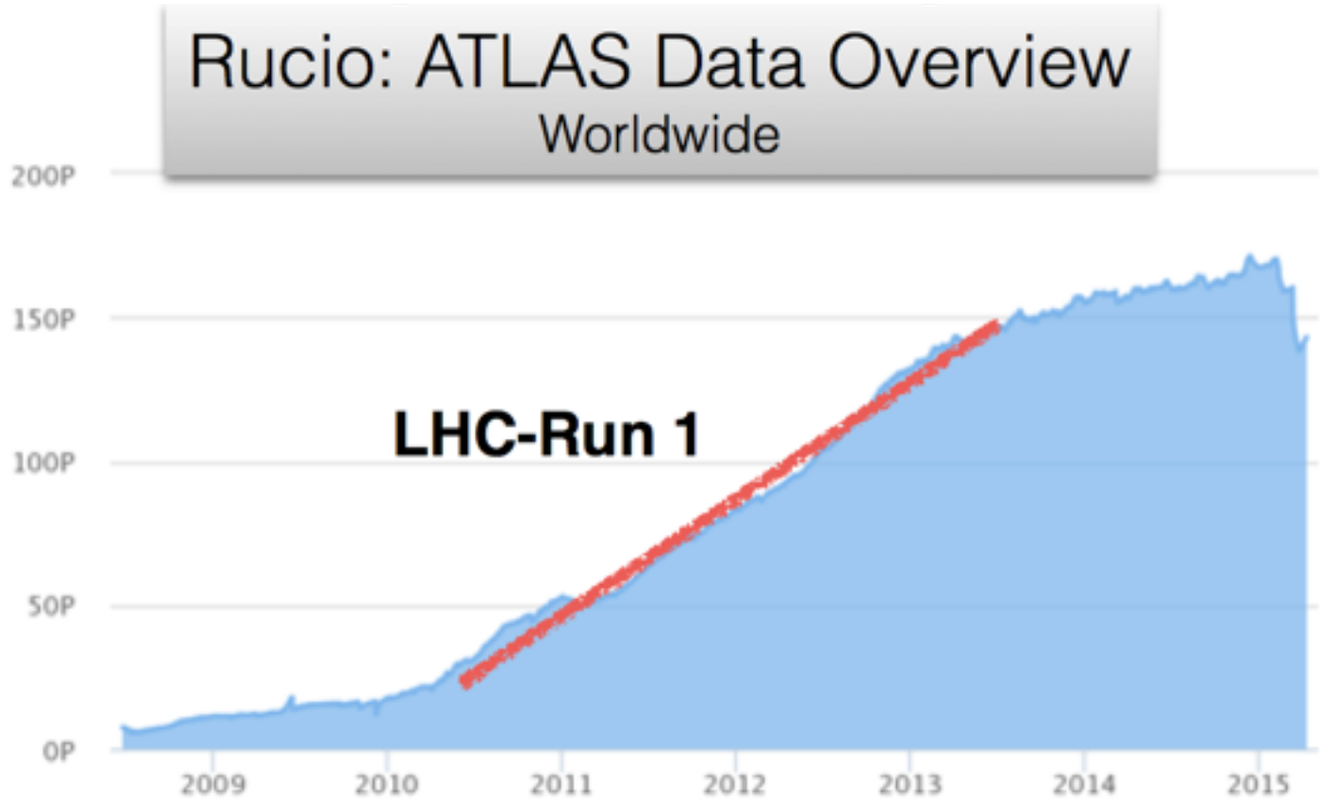
- Smart slimming and skimming frameworks used to bring data volumes under control
  - At the expense of some data duplication (though also augmentation used)
  - Must keep data volume and cpu costs under control
- Limited i/o capacity pushes us towards *train models*
  - One job reads an input AOD file, writes multiple analysis output formats
  - Maximise use of staged data
    - Staged can mean staged from **tape**, moved from **mass storage disk** to local **SSD**, data that has undergone persistent to **transient conversion**, data that has been moved from **main memory** into the **CPU cache** hierarchy…
- Internally analysis may use multi-threading to have multiple events in flight
  - Remains to be seen how useful/possible this model is re. programming difficulties (sandbox each event?)
- Object stores as a disruptive technology here…?

Computing in 10 years…

- This is very hard to predict, but
  - Certainly need custodial storage for RAW data
  - Large quantities of disk for online data
    - Fronted by smart caches of fast storage?
    - (The trick is not to cache what we just used, but what we are just going to use — hinted pre-caching via PanDA, ARC)
    - Will need to manage carefully volumes of derived and simulated data
      - Archive to tape more aggressively than in the past
- Storage services and compute services could increasingly decouple
  - Fast, smart networks funnelling data where it's needed
- Allows for easier use of heterogeneous resources
  - HPC, spot priced clouds, BOINC, …
- Classic WLCG sites will probably get bigger and more efficient
  - Evolution towards wider scientific remit (HPC/HTC convergence) as well as reducing costs and maintaining expertise
  - Smaller resources migrate to lightweight stacks — BOINC clients?

ATLAS Data Management

- Today in Rucio
  - 1,000 ATLAS users
  - 140PB
  - 700M files
  - 130 storage sites
- Run4
  - 1,000 ATLAS users
  - 500PB/year with 100 PB of RAW
  - 4B files/year
  - 130 storage sites (+ volatile storage: Cloud ?)



- DDM will have to scale with the (cumulative) number of files, bytes and data operations, e.g., 2.5M transferred files/day in 2015
- Most of DDM implementations for the LHC experiments are based on dataset/file catalogs, e.g., Oracle, and will follow the advances in data bases, middleware and open & standard technologies
  - Flexible design with no dependence on particular middleware
  - Horizontal scalability as a strong requirement

DDM Future

- Integration of new storage types might be needed
  - Object stores: Opportunity to handle data at the object level, e.g., event, and to store (physics) metadata with data
  - Volatile storage resources: cloud storage, HPC to increase the total storage capacity
- The biggest predictable gain will come from network (x100?) and will influence the experiments computing models
- Custodial data management with 2 copies on tape would stay unchanged but not necessary for the derived and secondary data, e.g., intelligent and content delivery (CDN) networks
  - Store vs. cache vs. recompute
  - Intelligent and maximal use of data when requested

Workload Management

- Major restructuring already for Run2: DEFT and JEDI to handle tasks and job splitting
  - Dynamic job definition to fit available resources
    - HPC, Volunteer, Grid, Cloud, …
  - Fine grained event service jobs for different workloads, beyond simulation
    - Relies on smart caches for storage an fine grained access to events
      - Retrieve, buffer locally and deliver events to workers
    - Meshes very nicely with new framework design
- Easier installation of PanDA as a local resource manager
- Utilise network knowledge to smartly marry data sources and sinks to CPU resources
  - Initially passively, but eventually actively using smart networks?

Conclusions

- Ten years is a long time away in many respects
  - Hard to make *concrete* predictions about hardware and software technologies
  - But we will have to live within our means (tape, disk, cpu and network) — adaptive process between *physics goals* and practical *affordable computing*
- But it's also not so long to see directions that we should be moving in
  - Multi-threading, data oriented designs, parallel algorithms, architectural flexibility, simplicity
  - Smart use of i/o, from tape through to online and local buffers
    - Dynamic popularity and smart workload management
  - Make use of network capacity in matching data capacity to processing capacity
- Many ideas should be advanced in an evolutionary way for Run3 (2020)
  - Testbed for Run4 challenges
  - This is *not so far away*, so need to start now