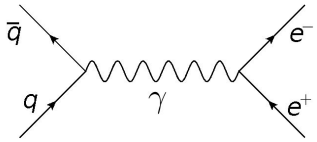
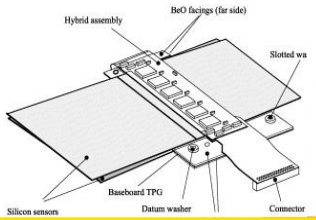
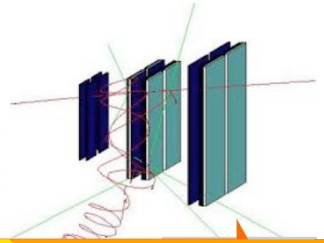
Simulation: Experience on ATLAS

JOHN CHAPMAN (CAMBRIDGE)
ON BEHALF OF THE ATLAS COLLABORATION

ATLAS Monte Carlo Production Workflow







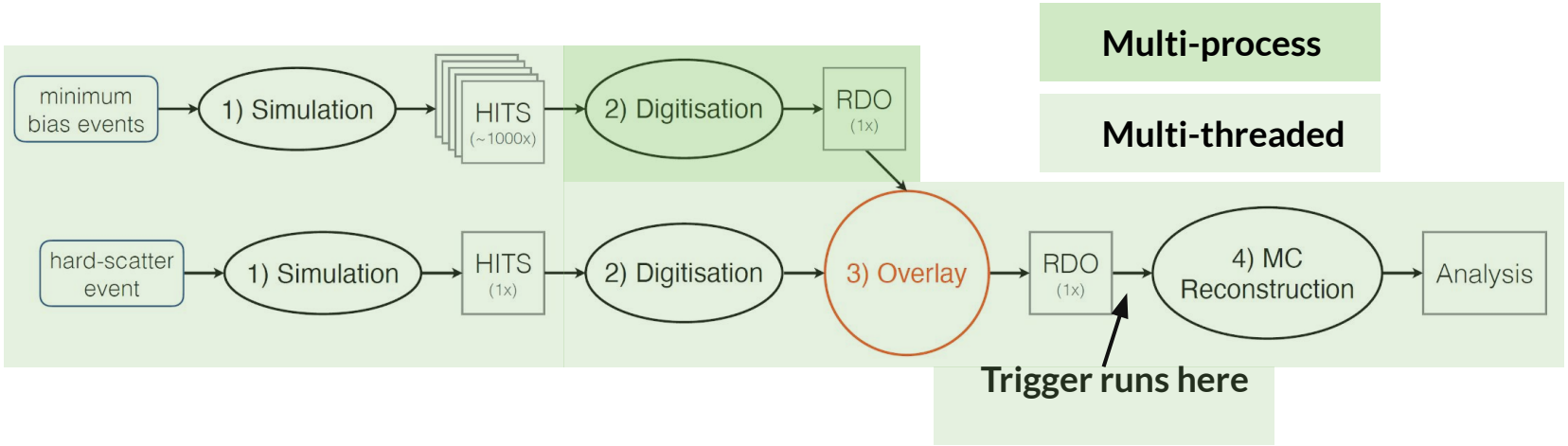
GENERATION → SIMULATION → DIGITIZATION → RECONSTRUCTION → DERIVATIONS

GENERATION	SIMULATION	DIGITIZATION	RECONSTRUCTION	DERIVATIONS
EVNT	HITS	RDO	RDO_TRIG ESD AOD	DAOD

Digitization and Reconstruction are typically run as separate athena jobs in the same production step, so the RDO files are not usually saved on the grid.

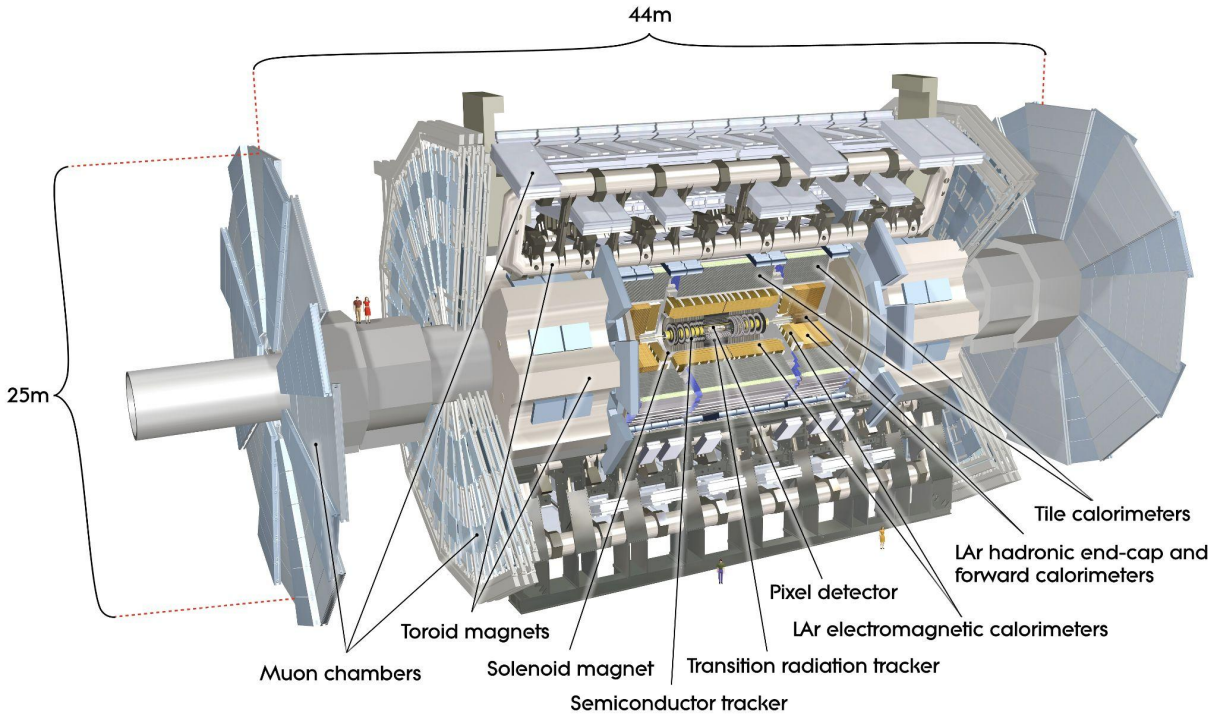
TEMPORARY
SAVED ON GRID

ATLAS Run 3 MC Production Workflow



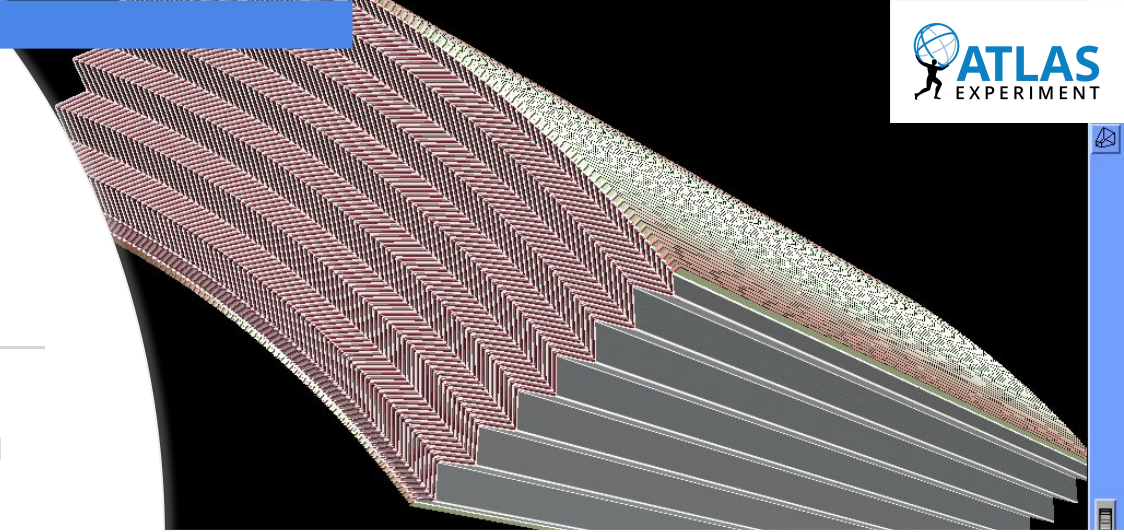
- For Run 3 most of the ATLAS MC production workflow was migrated to be thread-safe to take advantage of the large memory savings possible when running multi-threaded.
- The pile-up presampling step (see later) runs as a multi-process job (see later). This will be migrated to run multi-threaded before Run 4.

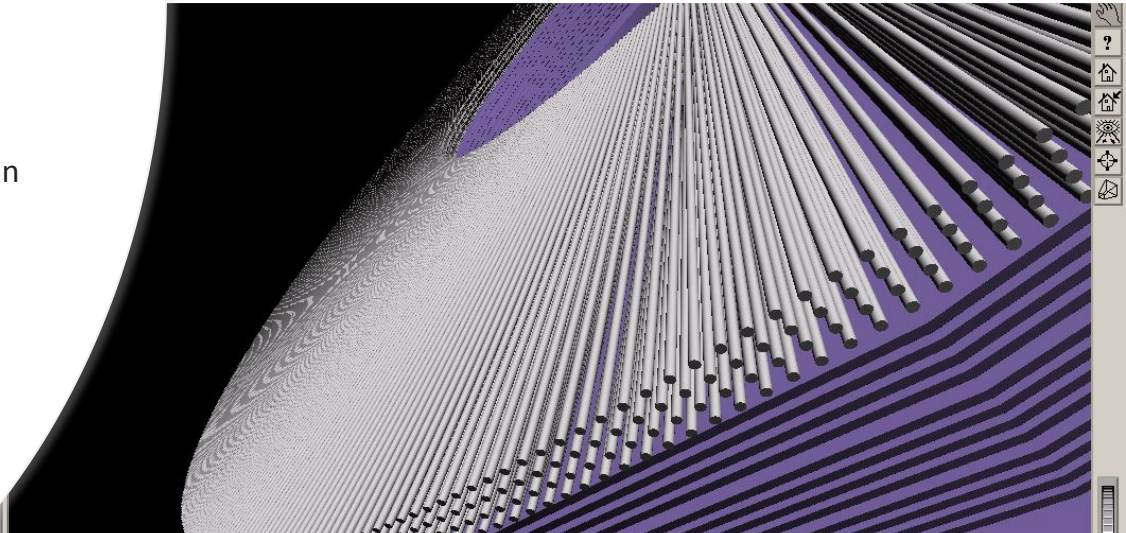
Detector Description: ATLAS Detector



ATLAS Detector Description: History

- The ATLAS detector description is based on a Geometry Kernel (set of geometry primitives) called GeoModel, introduced in 2003
- This is a second-generation geometry description language. Ideas are taken from previous collider experiment (CDF)
- However, ATLAS was an order of magnitude more complicated with many more volumes.
- There was skepticism that GeoModel would work in ATLAS (but then what would?)
- So in the design of GeoModel we were careful to use many tricks to minimize the memory footprint.
- Much of the design of the GeoModel kernel is inspired by the Open Inventor toolkit, which uses scene graphs to describe geometry.

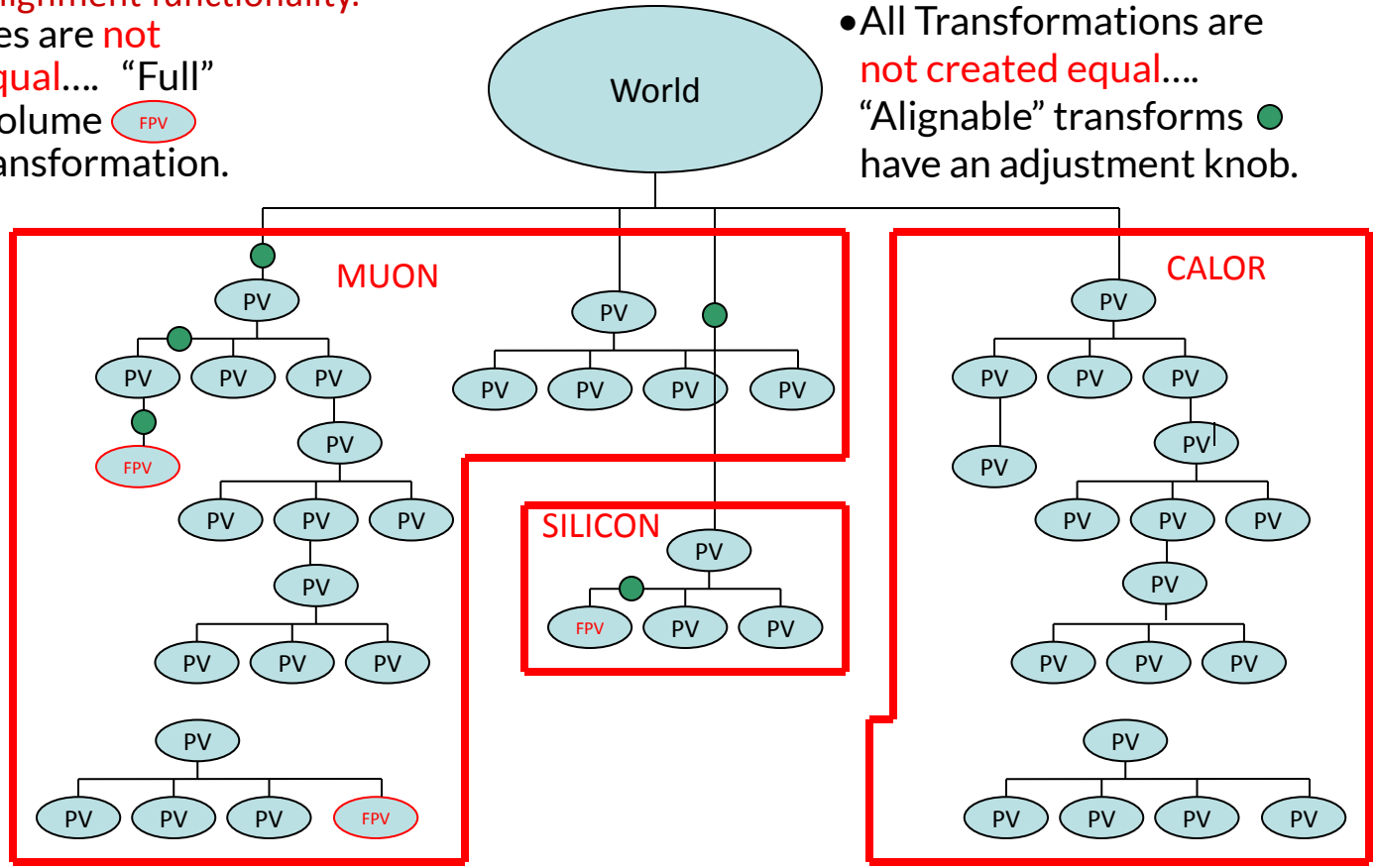




The GeoModel tree is a tree of graph nodes which mimic a tree of volumes. Physical volumes do not hold their transformations, which are otherwise encoded in the tree. The GeoModel system incorporates alignment functionality.

- All Volumes are not created equal…. "Full" Physical volume (FPV) caches transformation.

- All Transformations are not created equal…. "Alignable" transforms (●) have an adjustment knob.



Memory trick: high levels of compression are obtained by embedding symbolic expressions directly into the geometry tree (Serial Transformers)

```
Variable K;
GeoPhysVol  tV;
GENFUNCTION f = tubePitch*K + lstart;

TRANSFUNCTION t =     RotateX3D(90*deg)*
                TranslateX3D(tstart)*
                Pow(TranslateY3D(1.0),f);

GeoSerialTransformer
*s = new GeoSerialTransformer(tV,&t,nrOfTubes);
```

`t` is a mapping from real numbers to affine transformations.

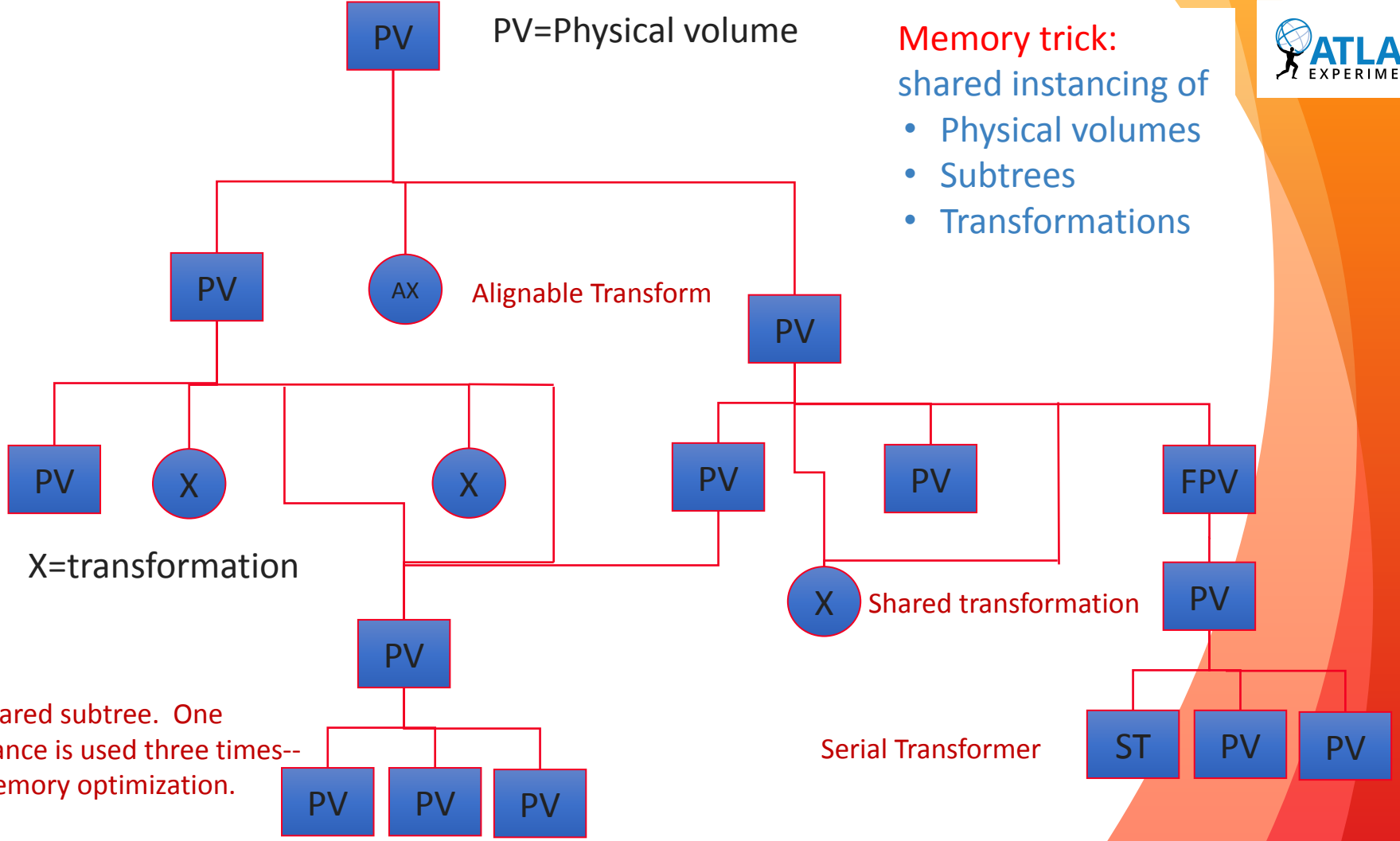
So `t(i)` is an affine transformation.

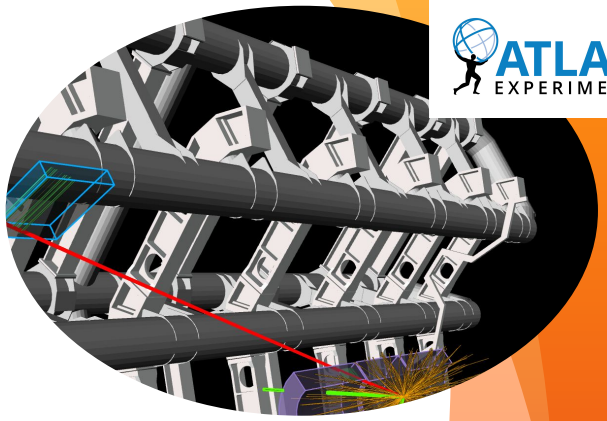
It encodes how volumes are to be placed.

Recipes of almost arbitrary complexity can be stored in a few bytes.

Geo-> G4 Conversion:
Same trick on the GEANT4 side lets us translate parameterizations into G4 parameterizations or parameterizations into placements, at the flip of a switch!

Memory trick: The tagging of geometrical volumes with names and copy numbers uses similar memory-saving techniques (in particular the names and numbers are not properties of the physical volumes, they live (if needed) in the tree).


PV
PV=Physical volume
Memory trick:
shared instancing of
• Physical volumes
• Subtrees
• Transformations
PV
AX
Alignable Transform
PV
PV
X
X
X=transformation
PV
PV
PV
FPV
X
Shared transformation
PV
PV
A shared subtree. One instance is used three times-- a memory optimization.
PV
PV
PV
Serial Transformer
ST
PV
PV



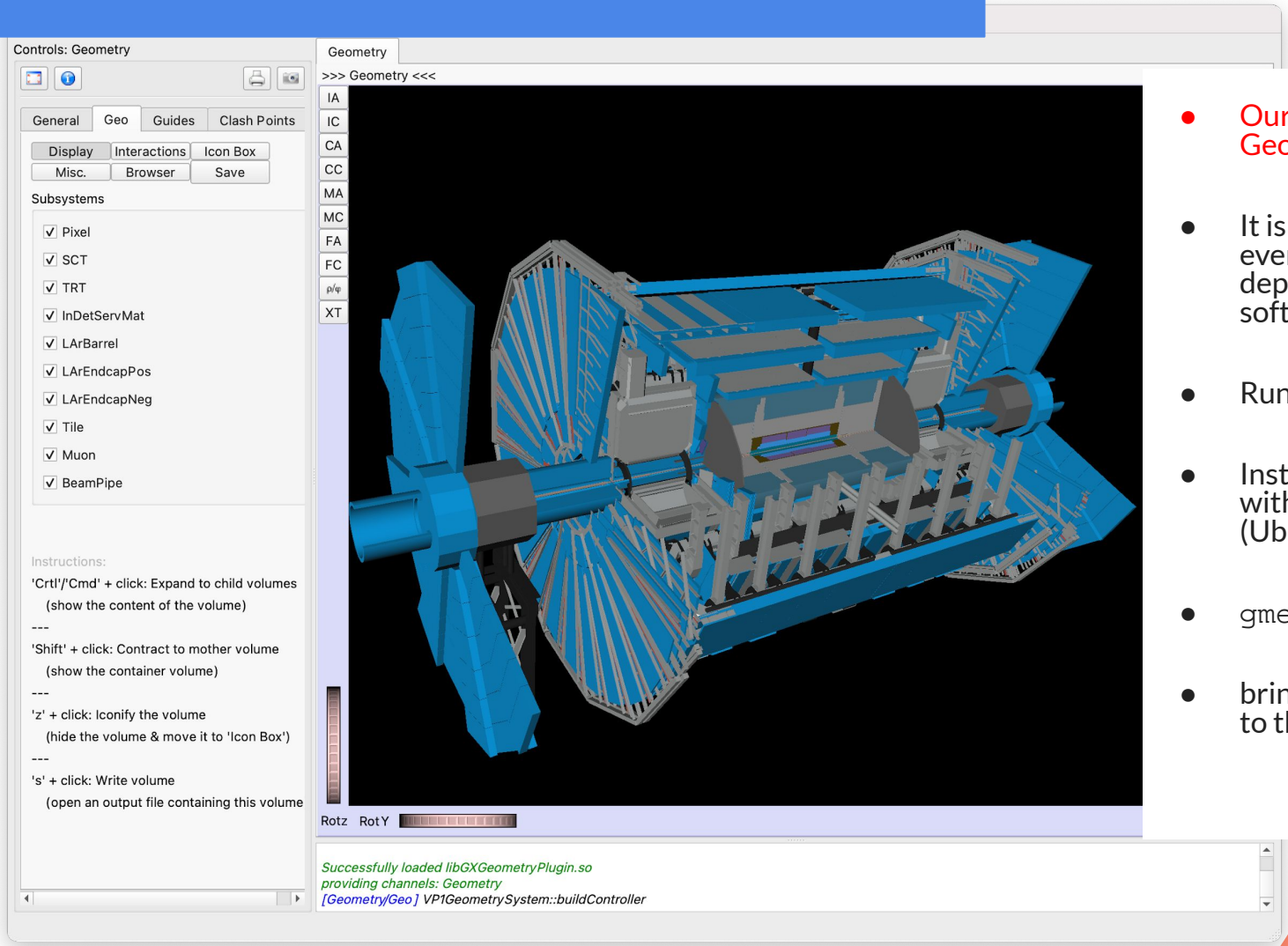
Example of
Boolean shapes

Memory trick: All instances of physical volumes, transformations, logical volumes, shapes, materials, elements Boolean shapes.. may be shared.

New: the GeoModel Toolkit

- The GeoModel tool suite is a set of tools that interoperate and exchange data either through geometry-builder plugins or through sqlite files. (Based on geometry persistification introduced in 2016)
- Basic:
  - gmcat: concatenates the output of plugins or sqlite files
  - gmstatistics: monitors memory consumption
- Visual
  - gmex: the geometry explorer, fast visual debugging from plugins or files
- G4 based
  - fullsimlight, testbench for simulation performance.. *plug in geometry like you plug in event generators*
  - gmmasscalc: mass inventories
  - gmgeantino: geantino scans
  - gmclash: clash detection
- Athena-based
  - dump-geo: writes the GeoModel used in athena to an sqlite file

All of these tools interoperate and can exchange data in sqlite format. Some are evolving, but slowly. These are released together with the GeoModel libraries. Minimal external dependencies.



- Our most popular tool is the Geometry Explorer (gmex).
- It is adapted from VP1 (ATLAS event display) but does not depend on a sprawling ATLAS software environment.
- Runs on Mac or Linux laptops
- Installs on these platforms with with brew (MacOS) or with apt (Ubuntu linux).
- `gmex geometry_atlas.db`
- brings the full ATLAS geometry to the screen in ~ 7 seconds.

Detector Description: Finally…

- We have been quite successful with geometry description for ~ 18 years.
- A recent review of other systems did not identify any advantages, only limitations and unwanted external dependencies.
- We are sometimes asked why we do not use “standard” root-based geometry description.
- We think that the question should be turned around: why don’t the other experiments (now and in the future) switch to GeoModel?
- Website https://geomodel.web.cern.ch

Simulation

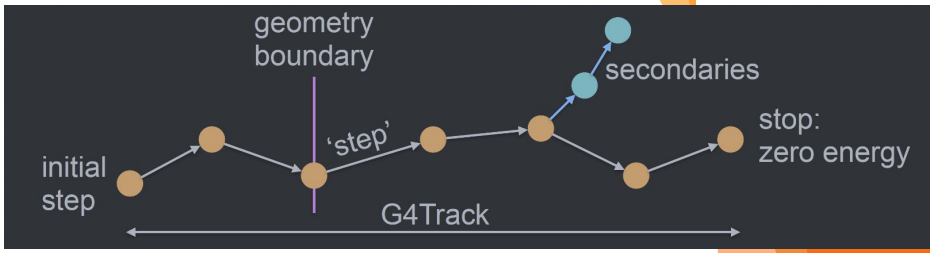


Figure by Miha Muškinja.

- ATLAS Simulation runs in Athena (based on Gaudi). This is interfaced to Geant4.
- G4Steps are key to optimizing the simulation. Two approaches:
  - **Make each step faster.**
  - **Reduce the number of steps taken.**
- Fast Simulations are likely to be very experiment-specific. Off the shelf solutions are unlikely to be optimal
- ML is not a magic bullet. Pick specific aspects of the simulation to start with.
- Know your detector! Ensure that the detailed simulation is being done in the parts of the detector where it is most useful for physics.
- Strike a balance between sticking with stable versions and early adoption of new ones.

Collaboration with Geant4

- Most likely any experiment will make extensive use of Geant4.
- Making example geometries or even whole test beam analyses available to Geant4 to use when validating new/improved models will help ensure that future G4 physics developments work well for your detector.
- Consider discussing your test beam program with Geant4, ahead of time.
  - It may be possible to do some simple things that would help the future of simulation.

Calorimeter Test Beam Integration to Geant Val

4

[Geant4 validation program]

Automatically validate Geant4 using hadronic and electromagnetic calorimeters test-beam data

Lorenzo Pezzotti & Alberto Ribon

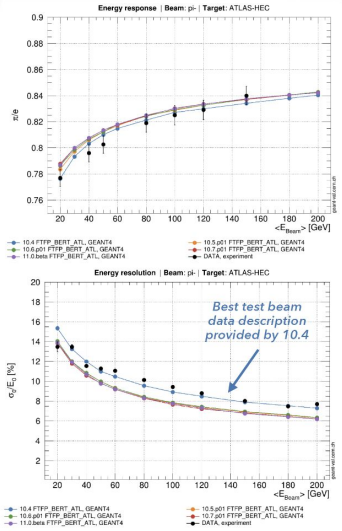
ATLAS data considered:

- Hadronic Endcap Calorimeter (HEC)
- Tile Calorimeter (TileCal)

Workflow:

1. Port the ATLAS HEC simulation into a new standalone Geant4 simulation
2. Perform Geant4 validation against the ATLAS HEC test-beam data
3. Porting the application into the Geant Val testing suite

Excellent example of collaboration between ATLAS and Geant4!



Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements

Gamma General Process

SteppingManager sees only 1 physics process for photons → reduced number of instructions

-4.3%

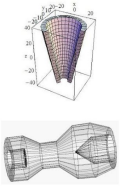
measured on 100 ttbar events in Athena

VecGeom

speed up using internal vectorisation for CPU – just for `G4Cons` & `G4Polycone`, no speed up measure considering all shapes

-1.5%

measured on 500 ttbar events in Athena



Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements 7

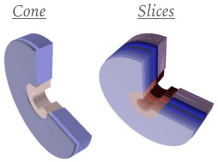
Simplifying Geometries *(aka reducing G4Polycone usage)* 10

EMEC

Described by a custom Geant4 solid using `G4Polycone` for internal calculations (Bounding Shape). Re-Implemented custom solid variants:

- Wheel: the default with G4Polycone
- Cone: improved shape using G4ShiftedCone – outer wheel divided into two conical-shaped sections
- Slices: new LArWheelSliceSolid – each wheel is divided into many thick slices along Z axis

Chosen variant Slices provided 5-6% speed up



Based on slides by E Kourlitis (Argonne)

Optimize!

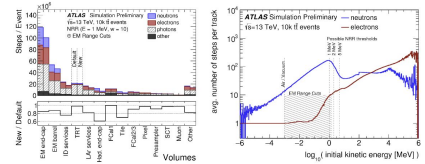
Intrinsic Geant4 Improvements 7

Simplifying Geometries *(aka reducing G4Polycone usage)* 10

Russian Roulettes & EM Range Cuts 9

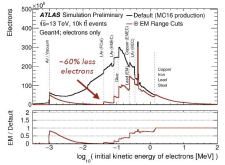
Russian Roulette

- Neutrons and photons take majority of CPU time
  EMEC most resource intensive
- Photon/Neutron Russian Roulette (PRR/NRR): randomly discard particles below energy threshold and weight the energy deposits of remaining particles accordingly
- NRR performance: 10% speed up with 2 MeV threshold for neutrons



EM Range Cuts

- OFF by default for three processes: **Compton, conversion, photo-electric effect**
- Turning them on provide ~6-7% speed up with negligible impact on physics



SIM-2019-001

Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements 7

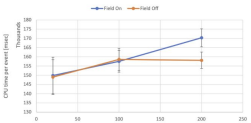
Simplifying Geometries *(aka reducing G4Polycone usage)* 10

Russian Roulettes & EM Range Cuts 9

Reducing Operations 8

Magnetic Field Tailored Switch-OFF

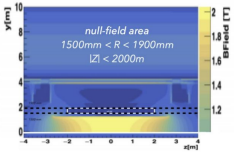
Speed up observed when switching-off magnetic field in LAr calorimeter (except for muons) without affecting shower shapes



Detailed studies showed smaller null-field area needed

- ~3% speed up for full ttbar events
- ~7% speed up for 1GeV e- on 0<n<0.17

Possibility to extend solution to other detector regions too



Vectorized sin/cos calculation in EMEC

calculates both sine and cosine in ElectroMagnetic EndCap geometry for a given radius, vectorization reduces operations needed

Both stand-alone and Athena timing shows a ~20% speed up in `LArWheelCalculator`.

Difficult to assess overall speed up

`LArWheelCalculator::parameterized_sincos` *takes only ~1.5% of total CPU time*

Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements 7

Simplifying Geometries *(aka reducing G4Polycone usage)* 10

Russian Roulettes & EM Range Cuts 9

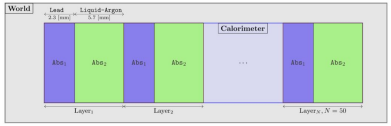
Reducing Operations 8

WOODCOCK TRACKING 16

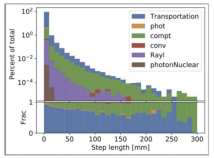
Reducing CPU time without approximations

Idea proposed by John Apostolakis

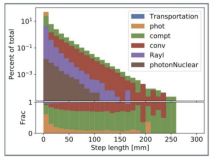
- Especially powerful in highly granular detectors (e.g, the EMEC) where geometric boundaries limit steps, rather than interactions
- Performs tracking in geometry with one material: the densest (Pb)
- Interaction probability is proportional to the cross section ratio between the real material and Pb
- Avoids many steps caused by geometric boundaries (*Transportation*) since there are no boundaries
- Up to 10% computational speed improvement for simplified layered Pb/LAr calorimeter (FullSimLight example by Mihaly Novak – image)
- Implementation for ATLAS EMEC ongoing



EMEC Total



EMEC Lead Only



Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements 7

Simplifying Geometries *(aka reducing G4Polycone usage)* 10

Russian Roulettes & EM Range Cuts 9

Reducing Operations 8

WOODCOCK TRACKING 16

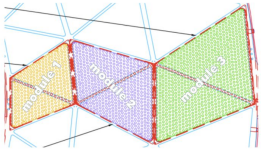
TRT Geometry Optimization 14

Currently the TRT geometry is described using **Boolean operations**

This approach is not optimal as Boolean operations are slow and they can cause tracking issues especially in presence of coincident surfaces

Describe these volumes using alternative shapes:

1. **arbitrary trapezoid (Arb8)**
   requires a total of 8 points to be specified – 4 vertices belonging to the −h/2 plane and 4 points belonging to the +h/2 plane
2. **the Boundary REPresentation (BRep)**
   requires the 4 vertices describing the trapezoid cross-section to be specified



96 trapezoidal modules grouped in 3 types characterized by an increasingly larger cross sectional area

Module shapes	Execution time (s)	Improvement
Boolean solids	1663	Reference
Arb8	1638	1.5%
BRep	1675	−0.7%

A speed up of 1.5% is observed for the Arb8 representation, whereas the BRep solid exhibits a minor slowdown with respect to the reference boolean solids

Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements 7

Simplifying Geometries *(aka reducing G4Polycone usage)* 10

Russian Roulettes & EM Range Cuts 9

Reducing Operations 8

WOODCOCK TRACKING 1

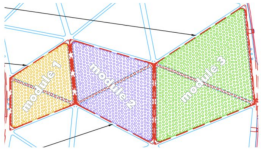
TRT Geometry Optimization

Currently the TRT geometry is described using **Boolean operations**

This approach is not optimal as Boolean operations are slow and they can cause tracking issues especially in presence of coincident surfaces

Describe these volumes using alternative shapes:

1. **arbitrary trapezoid (Arb8)**
   requires a total of 8 points to be specified – 4 vertices belonging to the −h/2 plane and 4 points belonging to the +h/2 plane
2. **the Boundary REPresentation (BRep)**
   requires the 4 vertices describing the trapezoid cross-section to be specified



96 trapezoidal modules grouped in 3 types characterized by an increasingly larger cross sectional area

Module shapes	Execution time (s)	Improvement
Boolean solids	1663	Reference
Arb8	1638	1.5%
BRep	1675	−0.7%

A speed up of 1.5% is observed for the Arb8 representation, whereas the BRep solid exhibits a minor slowdown with respect to the reference boolean solids

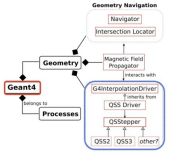
Quantized State System Stepper 17

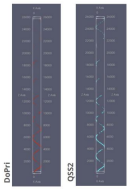
Background

- Quantized State System (QSS) numerical methods to solve the ordinary differential equations that govern the movement of particles in a field.
- QSS methods discretize the system state variables as opposed to traditional methods that discretize the time.
- Very efficient handling of discontinuities in the simulation of continuous systems.
- Based on: Efficient discrete-event based particle tracking simulation for high energy physics

Status

- Successfully ported QSS stepper from Geant4 v10.5 to v10.7.2
  to be added in G4 release
- Results using the N02 model qualitatively indistinguishable compared to those using the G4DormandPrince745
- Testing using FullSimLight
  ATLAS geometry & magnetic field map
- Performance profiling ongoing





Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements 7

Simplifying Geometries *(aka reducing G4Polycone usage)* 10

Russian Roulettes & EM Range Cuts 9

Reducing Operations 8

WOODCOCK TRACKING 1

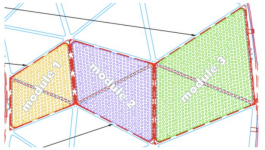
TRT Geometry Optimization

Currently the TRT geometry is described using **Boolean operations**

This approach is not optimal as Boolean operations are slow and they can cause tracking issues especially in presence of coincident surfaces

Describe these volumes using alternative shapes:

1. **arbitrary trapezoid (Arb8)**
   requires a total of 8 points to be specified – 4 vertices belonging to the −h/2 plane and 4 points belonging to the +h/2 plane
2. **the Boundary REPresentation (BRep)**
   requires the 4 vertices describing the trapezoid cross-section to be specified



96 trapezoidal modules grouped in 3 types characterized by an increasingly larger cross sectional area

Module shapes	Execution time (s)	Improvement
Boolean solids	1663	Reference
Arb8	1638	1.5%
BRep	1675	−0.7%

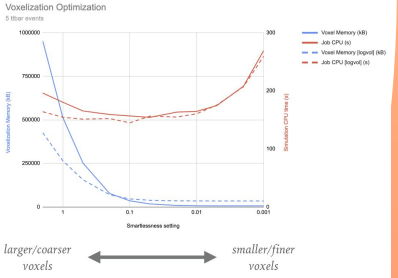
A speed up of 1.5% is observed for the Arb8 representation, whereas the BRep solid exhibits a minor slowdown with respect to the reference boolean solids

Quantized State System Stepper 17

Voxel Density Tuning 22

Tracking can be optimized by voxelization, the size/granularity of the voxels can be tuned by the *Smartless* parameter

- **Goal:** Optimize the values of *Smartless* parameter for a balance between memory used for the detector description and CPU time for simulation
- Simulation accuracy should also be checked – although no effect is expected
- Initial studies targeting the Run4 ATLAS ITk sub-detector with many tracking elements Will also investigate for Run3 detector



larger/coarser voxels ⟷ *smaller/finer voxels*

Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements 7

Simplifying Geometries *(aka reducing G4Polycone usage)* 10

Russian Roulettes & EM Range Cuts 9

Reducing Operations 8

WOODCOCK TRACKING 1

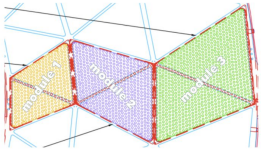
TRT Geometry Optimization

Currently the TRT geometry is described using **Boolean operations**

This approach is not optimal as Boolean operations are slow and they can cause tracking issues especially in presence of coincident surfaces

Describe these volumes using alternative shapes:

1. **arbitrary trapezoid (Arb8)**
   requires a total of 8 points to be specified – 4 vertices belonging to the −h/2 plane and 4 points belonging to the +h/2 plane
2. **the Boundary REPresentation (BRep)**
   requires the 4 vertices describing the trapezoid cross-section to be specified



96 trapezoidal modules grouped in 3 types characterized by an increasingly larger cross sectional area

Module shapes	Execution time (s)	Improvement
Boolean solids	1663	Reference
Arb8	1638	1.5%
BRep	1675	−0.7%

A speed up of 1.5% is observed for the Arb8 representation, whereas the BRep solid exhibits a minor slowdown with respect to the reference boolean solids

Quantized State System Stepper 17

Voxel Density Tuning 22

GPU-Friendly EMEC 15

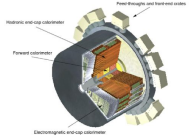
Description of the EMEC with Geant4/VecGeom standard shapes

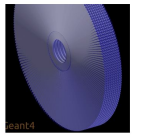
no accordion shape within the GEANT4 standard geometry shapes, defined a custom solid

- Possible speed up in VecGeom on CPU making use of internal vectorisation
- Possibility for the ATLAS geometry to be standard and GPU-friendly (see AdePT project)

Status

- Repeated accordion volume implementations using:
  1. `G4GenericTrap` (converted from `G4TwistedTrap`)
  2. `Arb8` & `G4Trap`
- Good progress overall
- Repository: https://gitlab.cern.ch/avishwak/atlas_emec_g4





Benchmark run in FullsimLight for G4Trap and G4GenericTrap using 1000 events with 10 GeV electrons

Geometry	Time (s)
G4Trap	111.67
G4GenericTrap	72.07

Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements 7

Simplifying Geometries *(aka reducing G4Polycone usage)* 10

Russian Roulettes & EM Range Cuts 9

Reducing Operations 8

WOODCOCK TRACKING

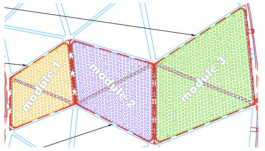
TRT Geometry Optimization

Currently the TRT geometry is described using **Boolean operations**

This approach is not optimal as Boolean operations are slow and they can cause tracking issues especially in presence of coincident surfaces

Describe these volumes using alternative shapes:

1. **arbitrary trapezoid (Arb8)**
   requires a total of 8 points to be specified – 4 vertices belonging to the −h/2 plane and 4 points belonging to the +h/2 plane
2. **the Boundary REPresentation (BRep)**
   requires the 4 vertices describing the trapezoid cross-section to be specified



96 trapezoidal modules grouped in 3 types characterized by an increasingly larger cross sectional area

Module shapes	Execution time (s)	Improvement
Boolean solids	1663	Reference
Arb8	1638	1.5%
BRep	1675	−0.7%

A speed up of 1.5% is observed for the Arb8 representation, whereas the BRep solid exhibits a minor slowdown with respect to the reference boolean solids

Quantized State System Stepper 17

Voxel Density Tuning 22

GPU-Friendly EMEC 15

G4HepEM Library Integration 18

G4HepEM library is a new compact Geant4 EM library
Jonas Hahnfeld, Benjamin Morgan, Mihaly Novak

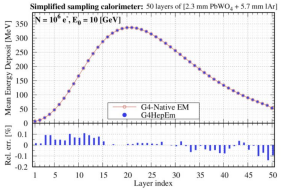
Optimized to be used for HEP electromagnetic showers development and transport

- more compact and GPU-friendly
- provides significant speed up with *Specialized Tracking*

Ongoing work of integration and benchmark, first in FullSimLight and then in Athena

CMS detector configuration simulating ttbar events	Physics List	Specialised Tracking	difference
	G4NativeEm	2889 s	2747 s	-4.9 %
	G4HepEm	2847 s	2660 s	-6.6 %
	difference	-1.5 %	-3.2 %	-7.9 %

Note: significant performance gain due to the specialised tracking of e^-/e^+ and γ even already using GEANT4 native processes that is boosted further with G4HepEm (even in its current, preliminary phase)



max 0.1% change in simplified calorimeter observables

More info & data here

Based on slides by E Kourlitis (Argonne)

Optimize!

Intrinsic Geant4 Improvements

Simplifying Geometries *(aka reducing G4Polycone usage)*

Russian Roulettes & EM Range Cuts

Reducing Operations

WOODCOCK TRACKING

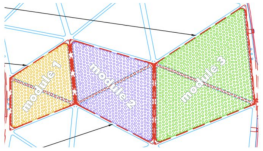
TRT Geometry Optimization

Currently the TRT geometry is described using **Boolean operations**

This approach is not optimal as Boolean operations are slow and they can cause tracking issues especially in presence of coincident surfaces

Describe these volumes using alternative shapes:

1. **arbitrary trapezoid (Arb8)**
   requires a total of 8 points to be specified – 4 vertices belonging to the −h/2 plane and 4 points belonging to the +h/2 plane
2. **the Boundary REPresentation (BRep)**
   requires the 4 vertices describing the trapezoid cross-section to be specified



96 trapezoidal modules grouped in 3 types characterized by an increasingly larger cross sectional area

Module shapes	Execution time (s)	Improvement
Boolean solids	1663	Reference
Arb8	1638	1.5%
BRep	1675	−0.7%

A speed up of 1.5% is observed for the Arb8 representation, whereas the BRep solid exhibits a minor slowdown with respect to the reference boolean solids

Quantized State System Stepper

Voxel Density Tuning

GPU-Friendly EMEC

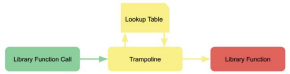
G4HepEM Library Integration

Non-Physics Improvements

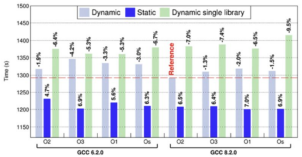
Big (static) Library

converging!

- Use Geant4 as static library(ies) to avoid "trampolines"



- Define a `BigSimulation` SHARED library, as a grouping of all libraries from packages that use Geant4



HepExpMT benchmark (Geant4 10.5.1) show 6-7% speed up

integration/testing into Athena ongoing ***no validation needed!***

Thread Local Storage (TLS)

- Athena profiling showed bottlenecks from usage of TLS
  - Going to MT in Athena/G4 cost ~5-10% due to TLS
- Both Athena & Geant4 are using TLS:
  - Athena → magnetic field
  - Geant4 → geometry data
- Work on reducing TLS usage is on-going from both sides
  - Athena → performance bug fix
  - Geant4 → investigating code restructure

Summary of most recent Geant4 Technical Forum update from ATLAS.

Quasi-stable Particle Simulation

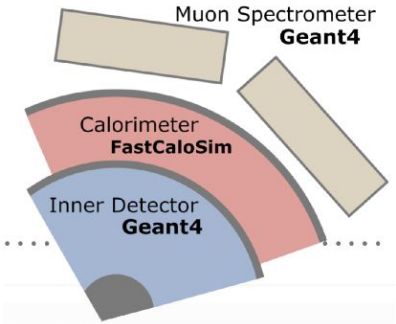
- Ordinarily only particles considered stable by the generator are passed to Geant4. As collision energies increase the boundary between evgen and simulation becomes blurred.
- Working definition of a **quasi-stable particle**:

  "***a particle which propagates outside the beam-pipe, but has already been decayed by the generator***"

- In such cases *hits may be missed* causing problems when apply b- and τ-tagging algorithms tuned on MC to data.
- Geant4 is working on adding hadronic interactions of b-hadrons, but even now it can handle generator defined decays, propagation and ionisation energy losses for such particles, allowing some "missing" hits to be recovered, if such particles are passed to Geant4.

Based on slides by J Beirer (Goettingen)

Fast Simulation in ATLAS: Overview



- ~80-90% of detector simulation spent on calorimeter simulation due to the large number of particles which need to be tracked.
- Fast and accurate shower simulation crucial.
- AtlFast3 (AF3) is the successor of the Atlfast-II (AFII) simulator.
- Full simulation of the ID and parameterized simulation of the calorimeter.
- AF3 implements two distinct approaches of shower generation:
  - **FastCaloSimV2:** parameterized modelling
  - **FastCaloGAN:** Generative Adversarial Network
- Dedicated parameterization for punch through particles from calorimeter showers.

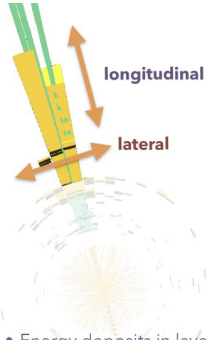
AF3 already used for reprocessing Run-2 events!

arXiv:2109.02551

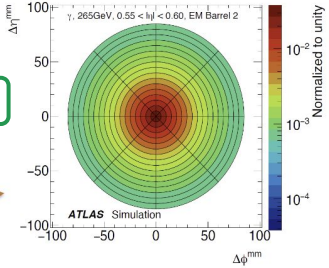
SCAN ME!

Based on slides by J Beirer (Goettingen)

FastCaloSimV2+FastCaloGAN: Overview



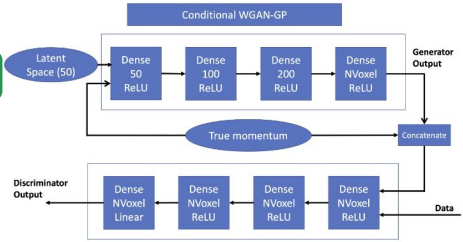
- Parametrization using Geant4 single photon, electron and pion samples
- Separate parametrisation of **longitudinal** and **lateral** shower development
- 17 bins of energy (64MeV - 4TeV) and 100 bins of |η| (0 - 5.0)



- Energy deposits in layers are **highly correlated**, difficult to model
- Classify showers based on depth on the interaction point, i.e. depth of shower initiation
- **Principal Component Analysis** (PCA) to de-correlate layers

- **Average energy distribution** in lateral direction parametrised over radial distance containing 99.5% of total energy and 8-bins in angular direction
- Parametrization for each particle, energy, eta, calorimeter layers and bins of 1st PCA
- During simulation, randomly generate quantised energy deposits (hits) from **2D shape histograms** (PDFs)

- FastCaloGAN based on **WGAN-GP** algorithm which offers more stable training compared to conventional GANs
- Electrons, photons and pions used to train the network
- **One GAN is trained for each of the 100 bins** in |η| (0 - 5.0) and conditioned on **truth momentum**
- Total of 300 GANs to cover full detector region
- GAN **trained to reproduce voxels and energies in the layer as well as total energy** in one single step
- Each GAN trained for 1M epochs with a checkpoint saved every 1K epochs



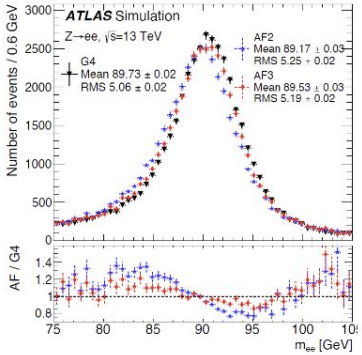
NVoxel	Number of voxels
Generator nodes	50, 50, 100, 200, NVoxel
Discriminator nodes	NVoxel, NVoxel, NVoxel, NVoxel, 1
Activation function	ReLU
Optimizer	Adam [50]
Learning rate	10^{-4}
$\beta 1$	0.5
$\beta 2$	0.999
Batch size	128
Training ratio (D/G)	5
Gradient penalty (λ)	10

WGAN-GP parameters

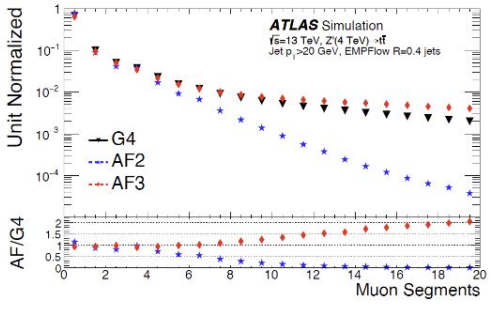
Based on slides by J Beirer (Goettingen)

Fast Simulation in ATLAS: Performance

Reconstructed Z mass from $Z \to ee$ decays

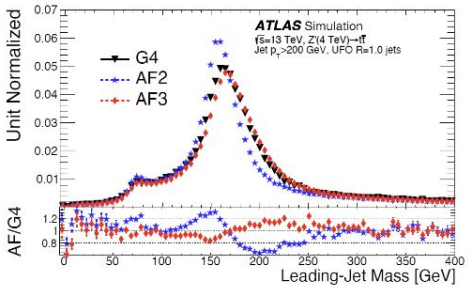


Associated muon segments in $Z' \to t\bar{t}$ decays

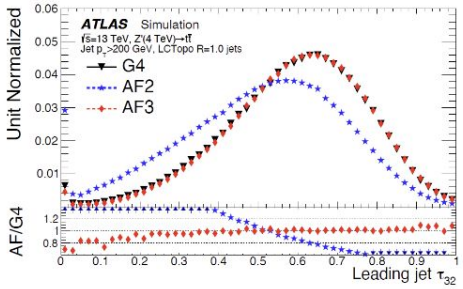


- AtlFast3 significantly improves in physics performance over the previous version AFII in almost all aspects
- AFII could not be used by many boosted analysis, while with AtlFast3 this is expected to be possible

Leading jet mass in $Z' \to t\bar{t}$ decays

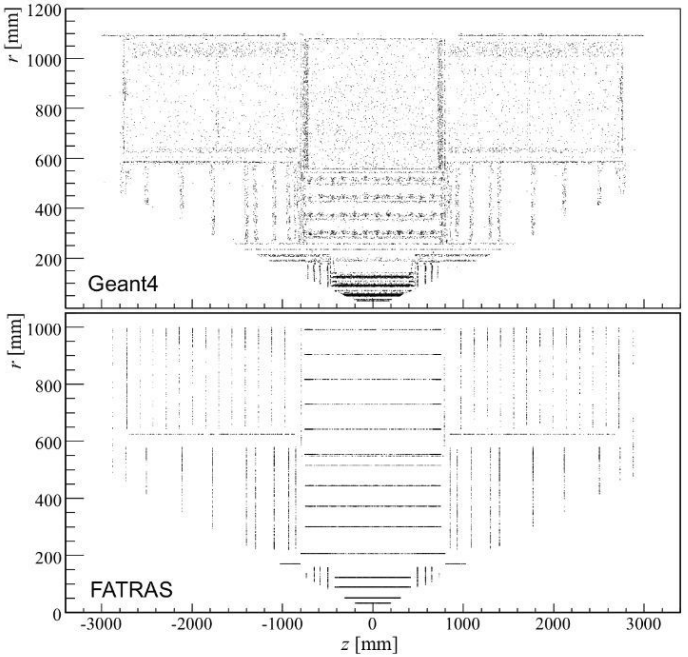


Leading jet τ_{32}



Future: Fast Chain

- For Run 4, the amount of data will be such that even ATLFAST3 will not be fast enough to keep up.
- The next step will be a fast simulation for the ATLAS Tracker. This will be (ACTS) FATRAS (see earlier talk).
  - FATRAS+FastCaloSim is ~100 times faster than pure Geant4.
  - MC production time will then be reconstruction dominated.
- At this point we will stop saving simulation output (HITS) as an intermediate format and go straight from EVNT to AOD in a single production step on the grid.
- Aiming for production-readiness before the end of Run 3.

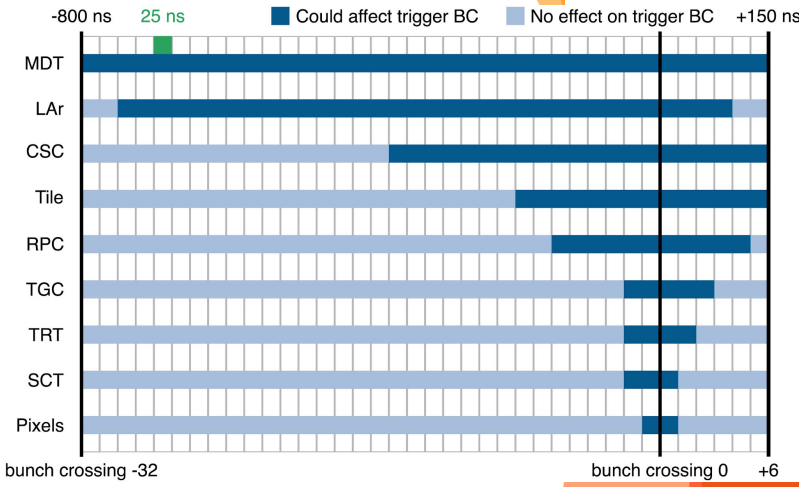


Visualization of the simplified geometry used by the standard ATLAS track reconstruction and FATRAS, derived from photon conversion vertices. (ref.)

Digitization

- ATLAS Digitization code is implemented by the software developers from each detector subsystem project.
  - All using a common interface.
- Core code for handling pile-up implemented by the Simulation group.
- Run within the Athena framework.
- Once again the balance is between accuracy and speed.
  - This should ideally be taken into account during the detector design phase.
- Key issue is how to deal with pile-up (soft collisions in the current and surrounding bunch crossings).

Pile-up Digitization (I)



- The readout of many subsystems is sensitive to multiple LHC bunch-crossings (BCs) around the trigger BC.
  - Collisions from 39 BCs must be taken into account.
- The average number of interactions that must be included is quite large:
  - E.g. 1560 interactions for 40 average interactions per bunch-crossing.
- Simulating this many extra interactions for each hard-scatter event would be prohibitive.
- Instead ATLAS decided to include the effect of pile-up collisions during the digitization production step.

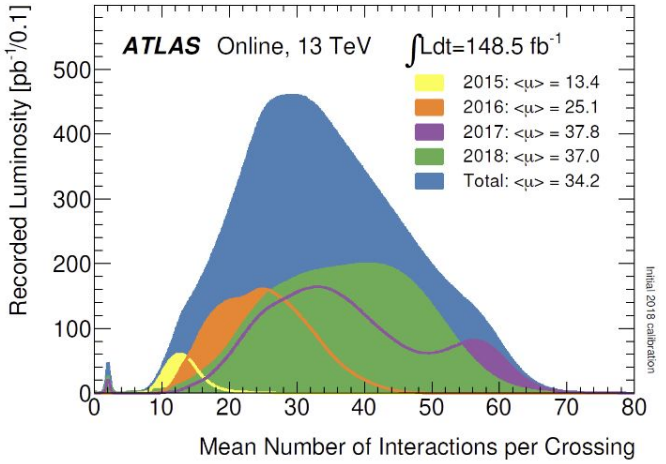
Pile-up Digitization (II)

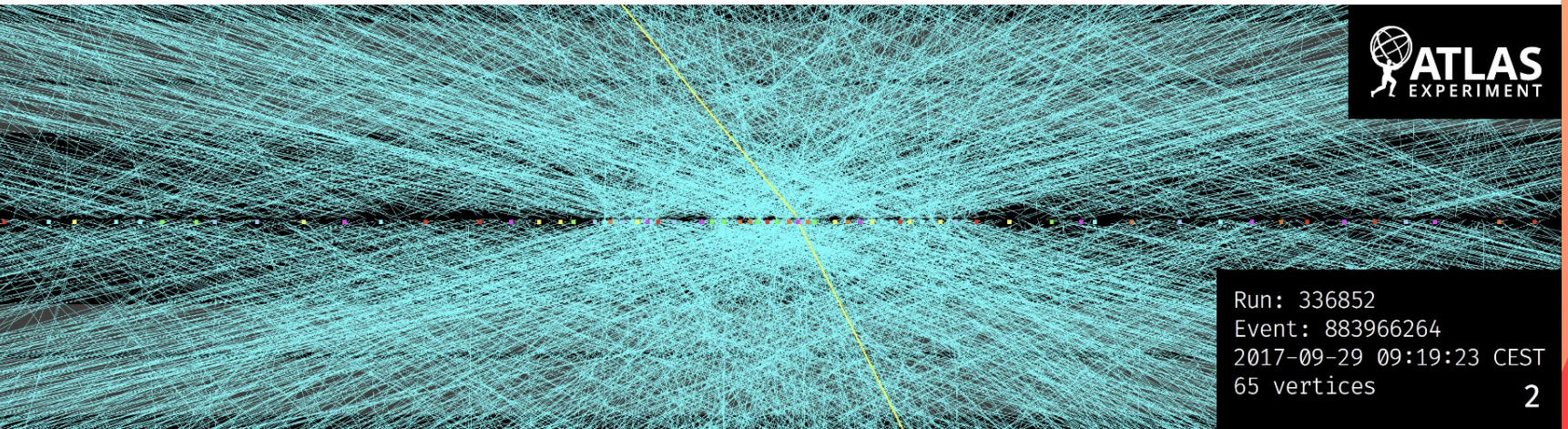
- **How best to include pile-up interactions during the digitization step?**
- This has undergone multiple interactions in ATLAS:
  - **Initial approach (~2008):** simulate individual pileup interactions, build a cache of these events in the digitization job and sample N events at random to use with each hard-scatter. Works fine for low numbers of interactions per BC, but memory does not scale well.
  - **Revised approach (~2012):** similar, but read pileup interactions one BC at a time, use, then drop from memory. Memory scales better, but I/O load higher.
- **New approach required…**

Based on slides by T Novak (DESY)

MC Overlay: Motivation

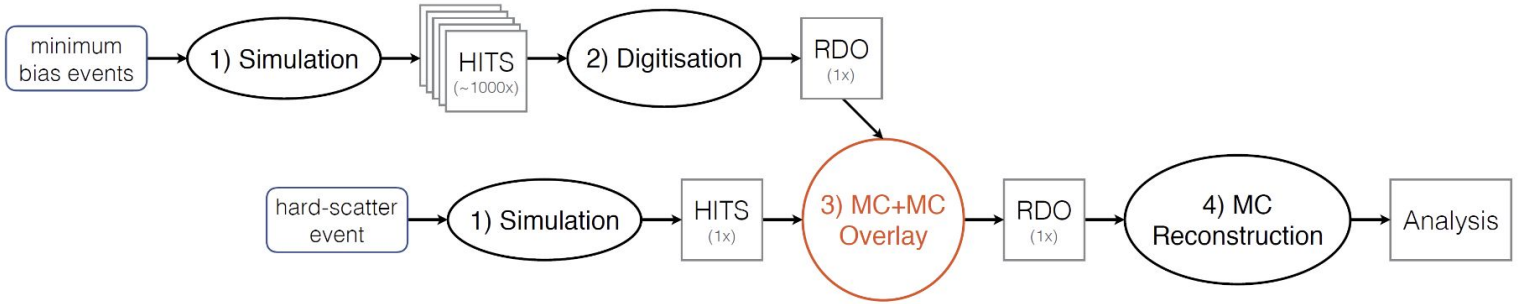
- Pile-up: soft collisions in current and surrounding bunch crossings.
- The average pile-up in HL-LHC expected to increase 4-5 times from 34 to 140 reaching values up to 200.
- High simulation and digitisation CPU requirements directly proportional to number of interactions μ, depending on the strategy.



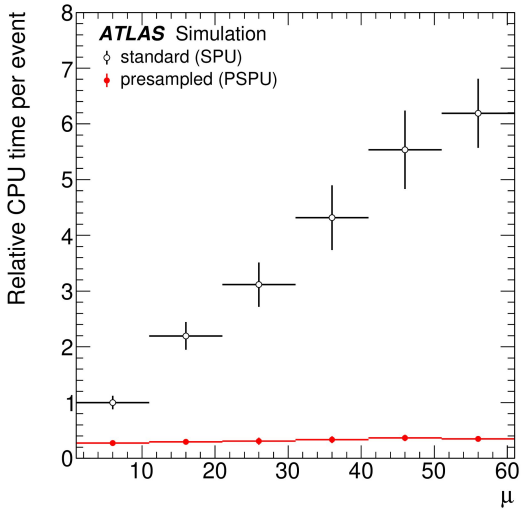


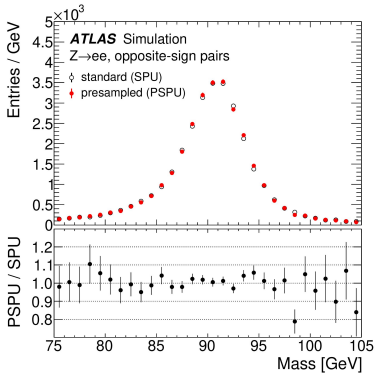
MC Overlay: Method Overview

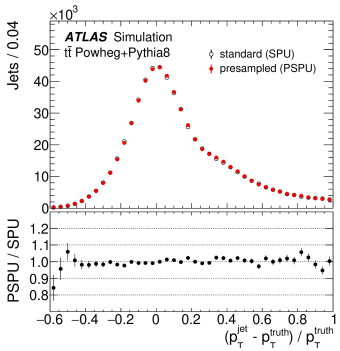
- Simulate hard-scatter and minimum bias events with GEANT4 as usual.
- Presampling: A large sample of combined pile-up events is produced from simulated minimum bias events during a separate digitisation step.
- Each simulated hard-scatter event is then digitised and combined with an event sampled from these pileup datasets.
- The main benefit of the new method is that the CPU and I/O requirements of the digitisation are significantly lower and have almost no dependence on μ.
- Pre-mixed pile-up events can be reused for different hard-scatter samples.



MC Overlay: Physics and Computational Performance





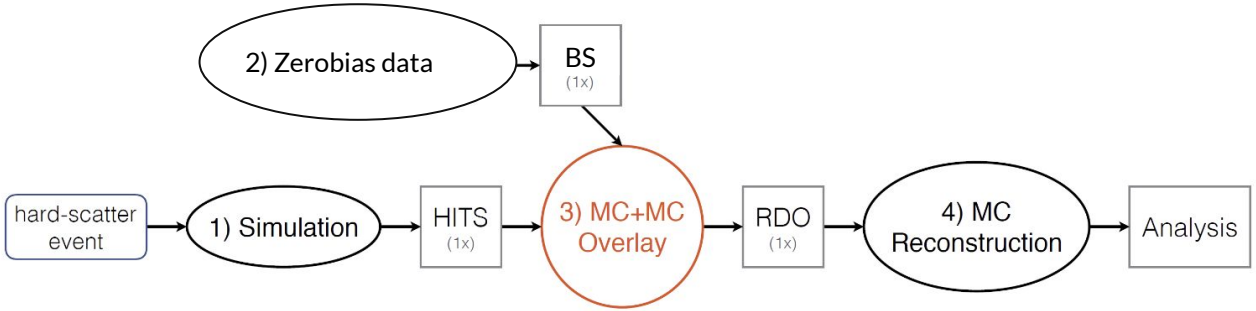


- No significant reduction in tracking resolution and calorimeter performance.
- Reconstruction and trigger efficiencies very similar.
- Analysis-level observables very close to standard digi.
- CPU gains come from re-using presampled RDO files.

Method now published in Computing and Software for Big Science

Data Overlay: The future?

- Record zerobias data events (trigger readout one LHC turn after normal trigger with some probability).
- Combine this information with a simulated hard-scatter event using the Overlay machinery.
- The main benefit is that we now include the exact backgrounds from data.
- Issues to work around: real detector components change shape with temperature and under gravity, this is not included in the Geant4 geometry. Applying data alignments to Geant4 geometry leads to volume overlaps and missing SimHits.
- Combining with FATRAS may be easier as this uses a simplified geometry.



Things to consider: Cost in Personpower

- Optimizations which change physics output take longer to validate than those that don't.
  - Sometimes this can be an argument for targeting smaller gains first.
- Physics objects produced by each simulation "flavour" will need to be separately calibrated.
  - If considering using full simulation and fast simulation together this may double the calibration workload.
  - Often personpower is the rarest commodity on an experiment.
- Document how to do things. Especially things that only have to be done every few years!
  - This one is really tough.
  - Recently we have been trialling recording workshops where procedures are discussed.

Summary

- ATLAS Software has been evolving for over 20 years, so the collaboration has had time to try out a number of different approaches.
- Consider using GeoModel for detector description, especially for complex geometries.
- Work closely with Geant4 to ensure new versions can easily be compared to your data.
- Investigate all possible Geant4 optimizations!
- Consider quasi-stable particles.
- ATLFAST3 makes calorimeter simulation time negligible without sacrificing physics output.
- Optimize storage usage (Fast Chain).
- Consider how to deal with pile-up carefully.
- Don’t forget the cost in personpower of decisions made when optimizing the software?