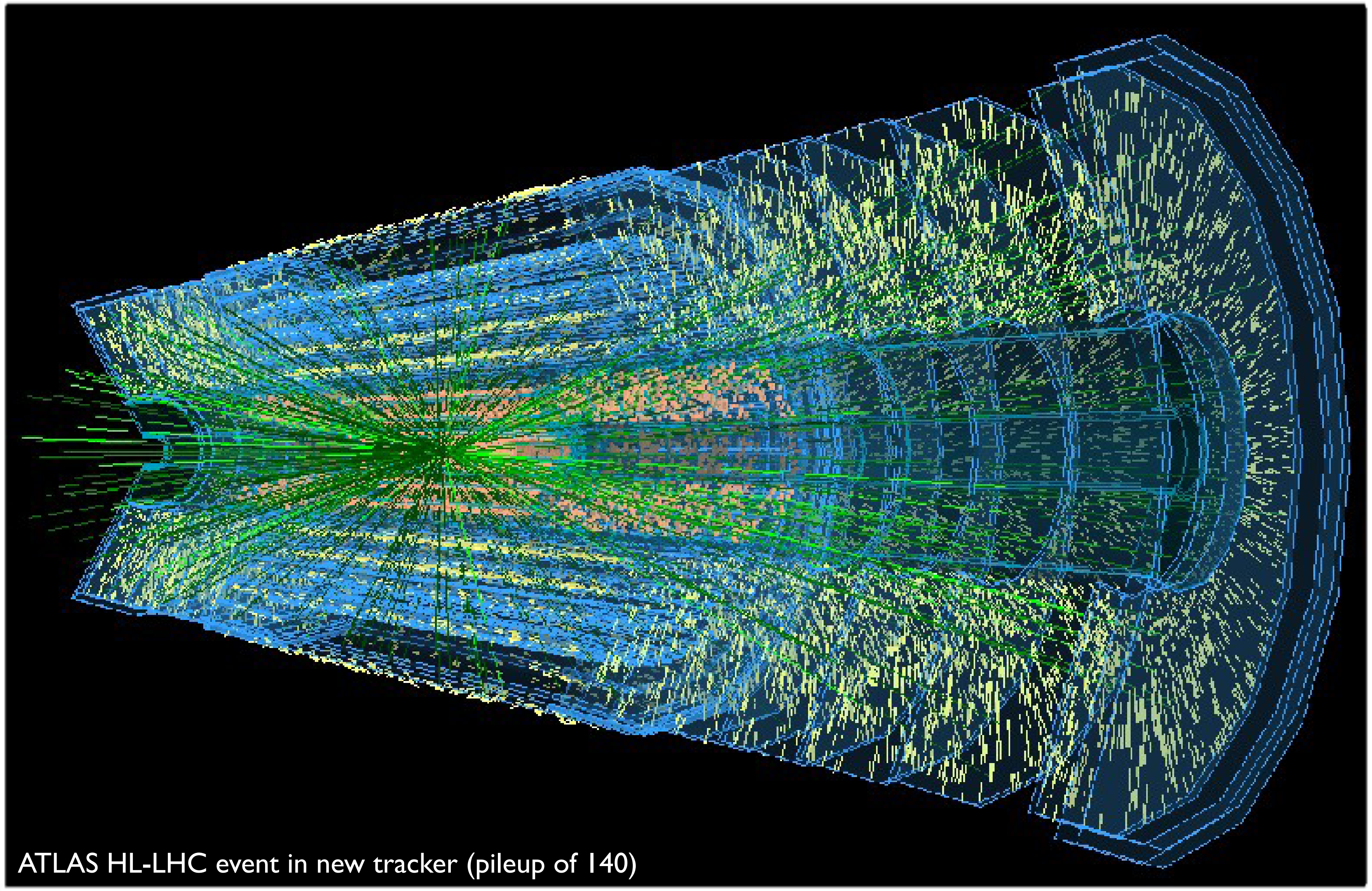
Tracking for High Pileup

Markus Elsing

ECFA Preparation Workshop, September 4-5th

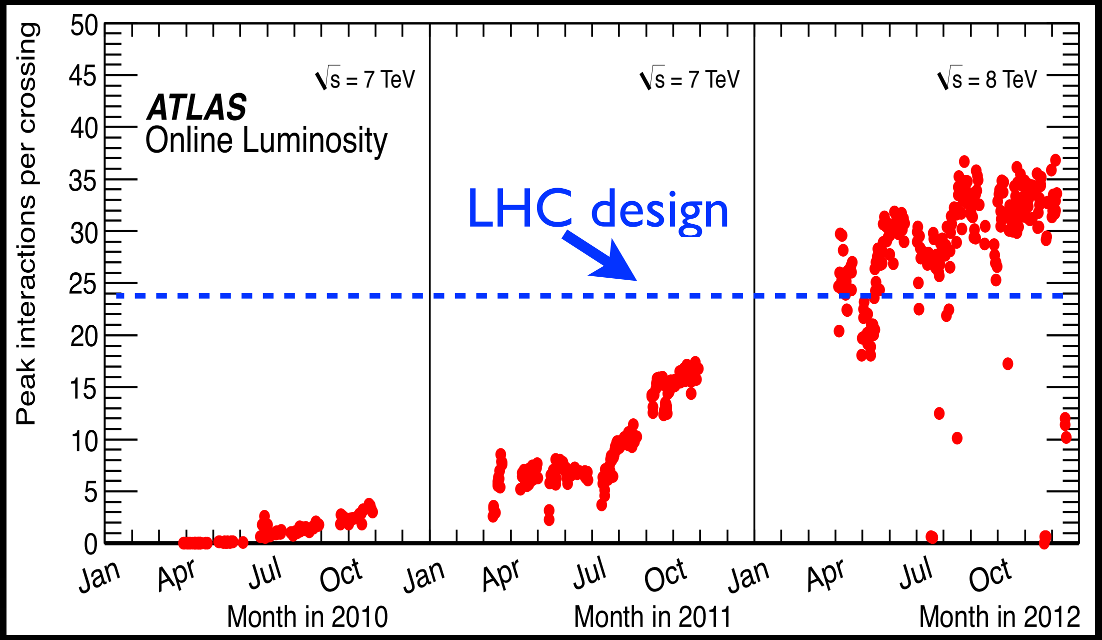


ATLAS HL-LHC event in new tracker (pileup of 140)

Introduction: the Challenge

Pileup during Run-1and Future Expectations

- pileup in 2012 exceeded design
  - ➡ average pileup up to 35 (1.5 × design)
  - ➡ due 50 *ns* operation during Run-1

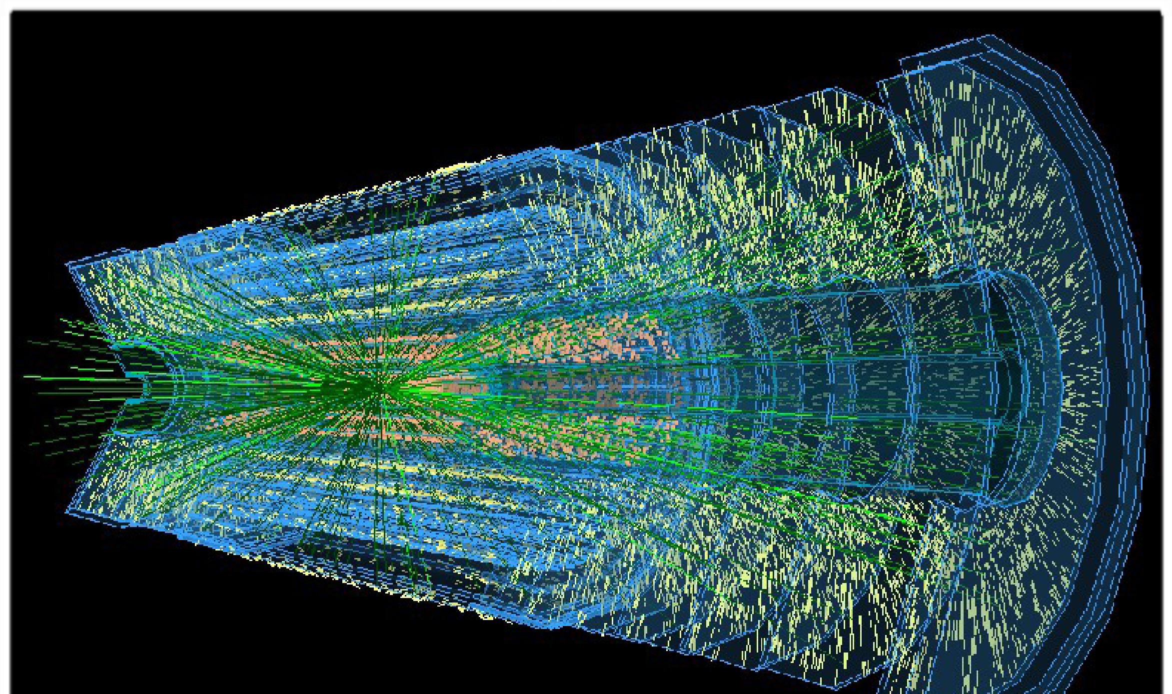


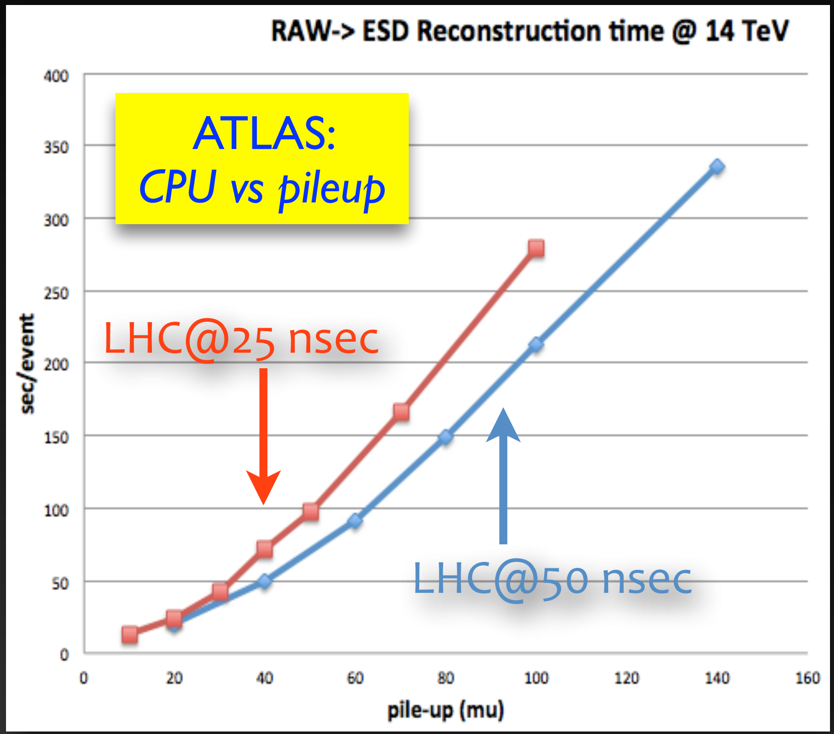
- Run-1: good stability of tracking performance vs pileup (ATLAS, CMS)
  - ➡ test with high pileup runs show limitations when going much further

- expectation for Run-2 and Run-3
  - ➡ luminosity up to 2-3 × 10^{34} $cm^{-2}s^{-1}$
    - pileup of 40 up to 80 (at 25 *ns*)
  - ➡ ATLAS and CMS aim for ~1 *kHz* data taking rate
    - allows to keep especially single lepton triggers
  - ➡ challenge for physics performance and resource needs for reconstruction, especially for tracking

Tracking at High Pileup ?

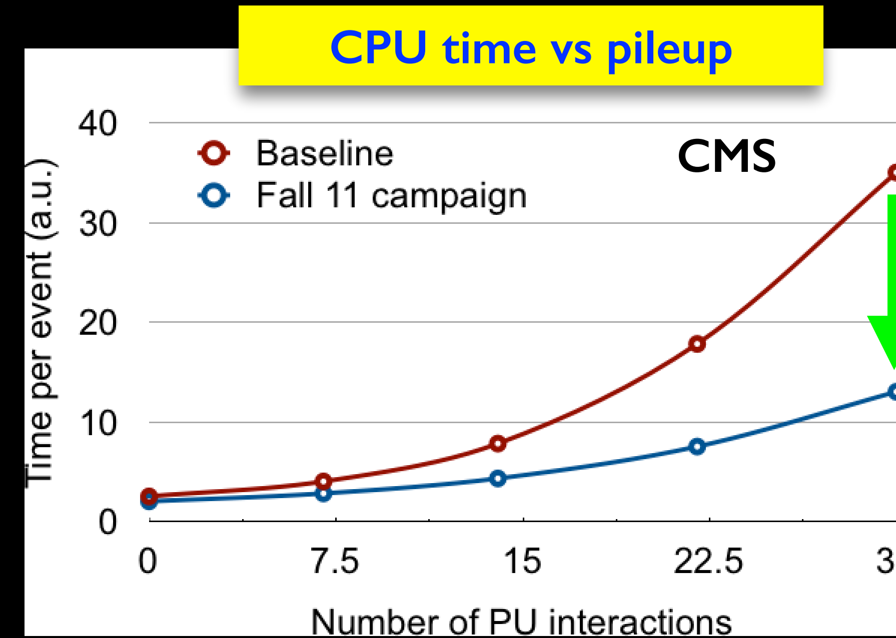
- looking even further: HL-LHC
  - ➡ luminosity $5 \times 10^{34}\ cm^{-2}s^{-1}$ with leveling
  - ➡ pileup levels ~140-200
  - ➡ major tracker upgrades in shutdown 2023
- the million dollar question:
  - ➡ how to reconstruct HL-LHC events within resources ?
  - ➡ tracking naturally resource driver for reconstruction (CPU/memory)
- this is not a new question !
  - ➡ we knew that tracking at the LHC is going to be a problem
    - hence: we aim at improving over something that has already been highly optimised
  - ➡ processor technologies are going to change as well
    - need to rethink some of the design decisions we did
    - will require vectorisation and multi-threading
    - improve data locality (avoid cache misses), etc.

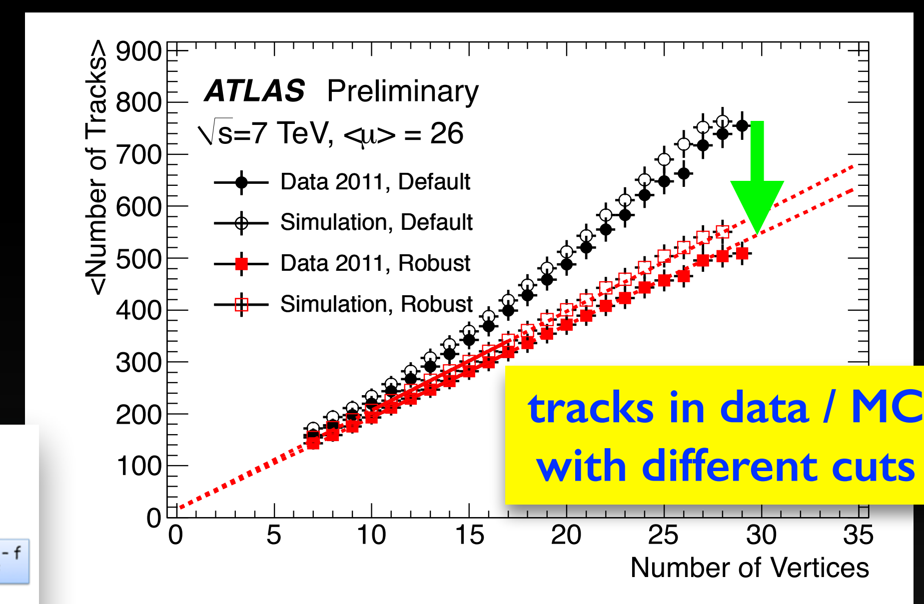

ATLAS HL-LHC event in new tracker

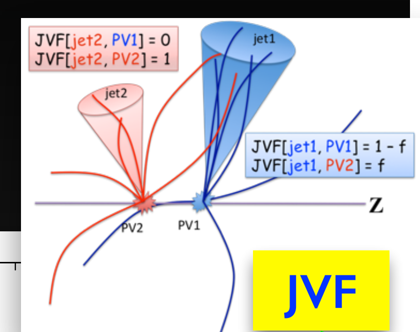


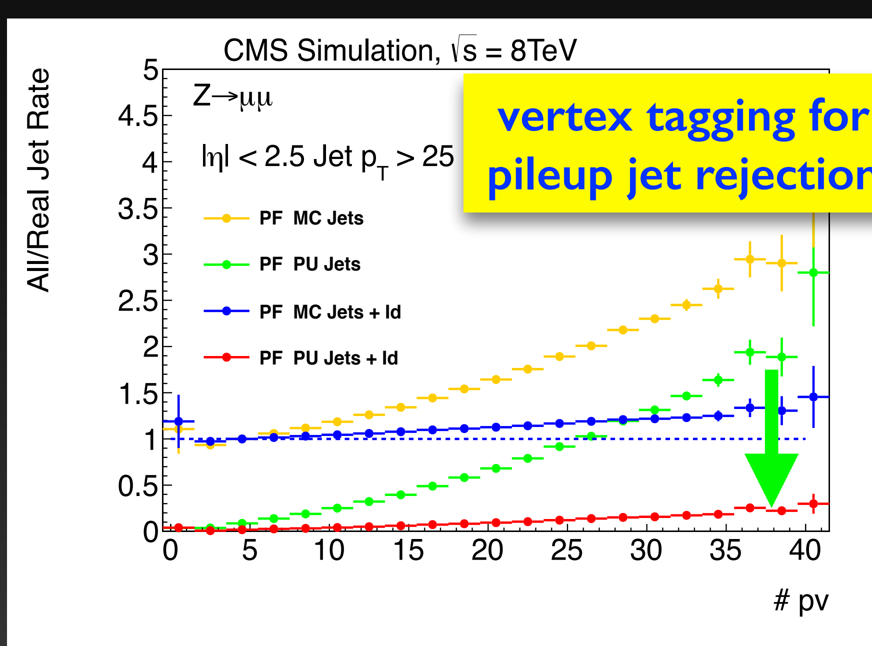
Run-1 Experience with Pileup

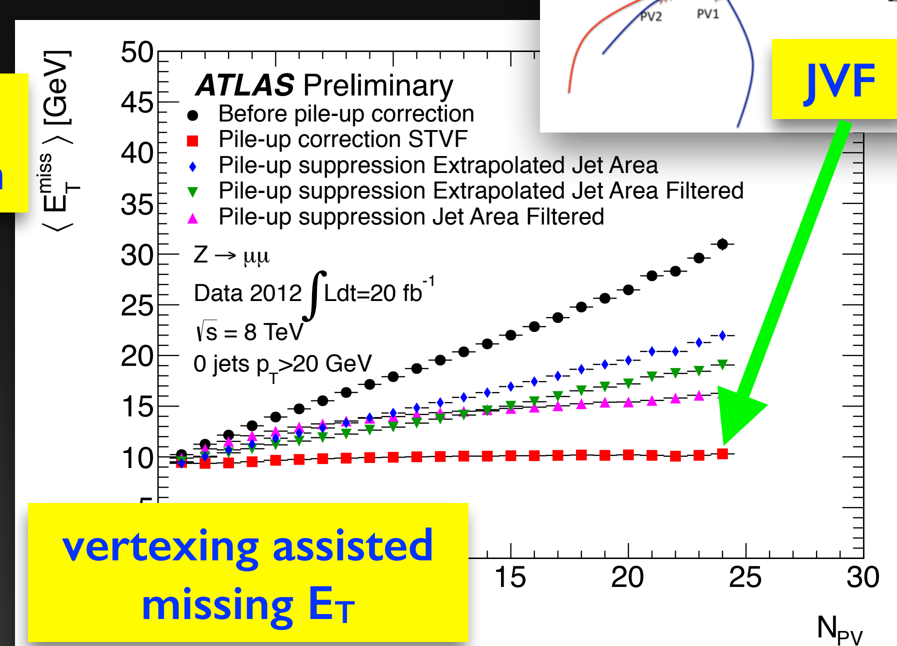
- **tracking** performance as expected
  - ➡ both experiments use similar tracking strategy (in silicon)
    - CPU increases rapidly with μ (combinatorial explosion)
    - big improvements with tracking updates during Run-1
  - ➡ more robust tracking cuts controls fakes
- primary **vertexing**
  - ➡ visible effects of vertex merging at high μ
  - ➡ Σp_T based vertex tagging less and less optimal (see MC)
- tracking as a **tool for pileup control**
  - ➡ e.g. pileup jet tagging (JVF and variants of it)
  - ➡ CMS jets, $\not{E}_T$ and τ based on particle flow

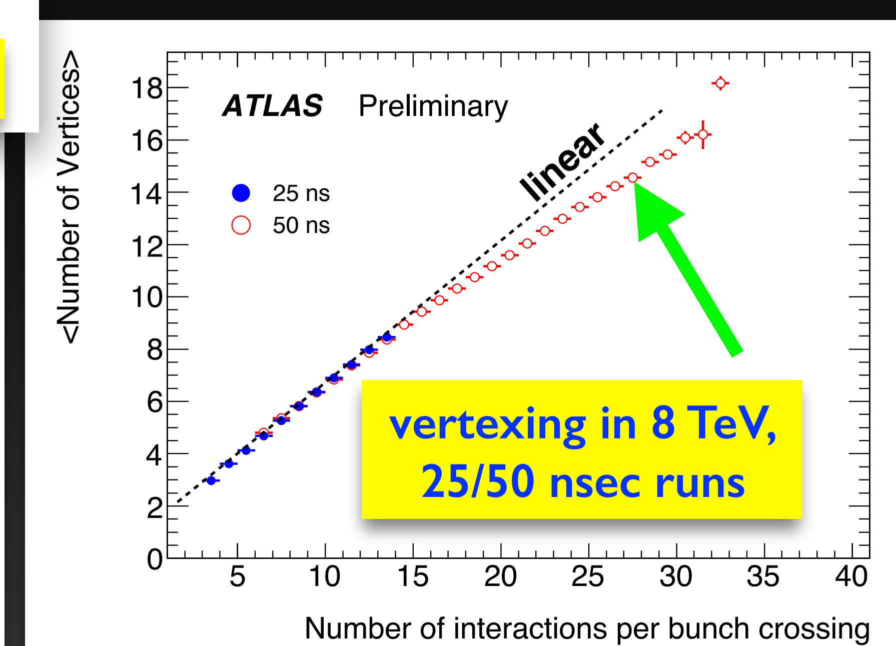








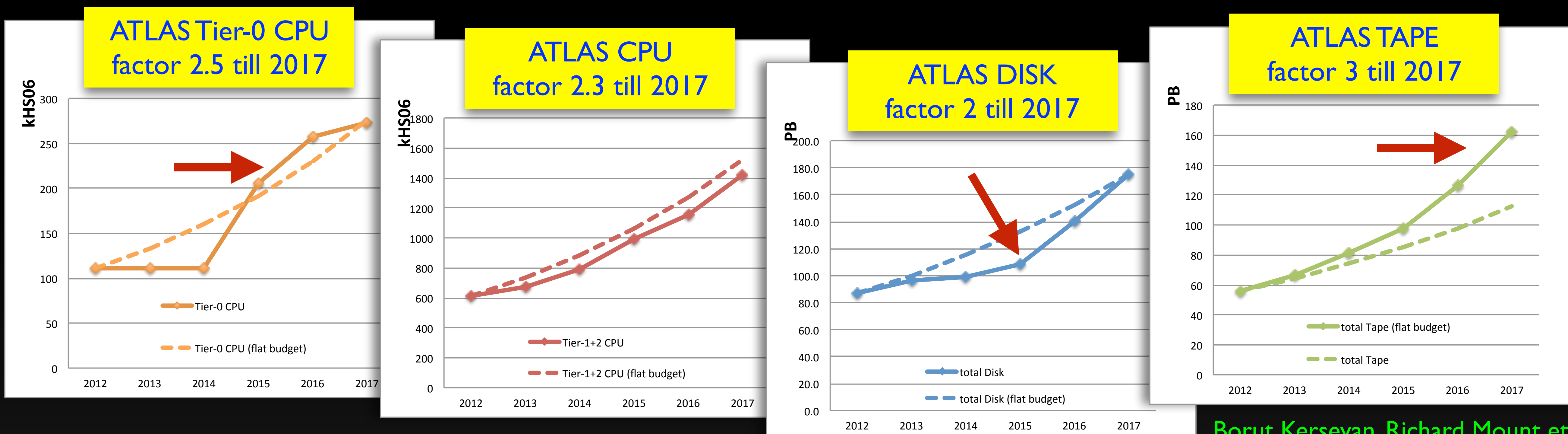




Preparing for Run-2 in current Long Shutdown (LS-1)

Computing Constraints for Run-2

- unlike Run-1, our computing resources will be limited !
  - ➡ assumption is we stay with a constant computing budget
  - ➡ interplay of technology advancement, market price and needed replacements

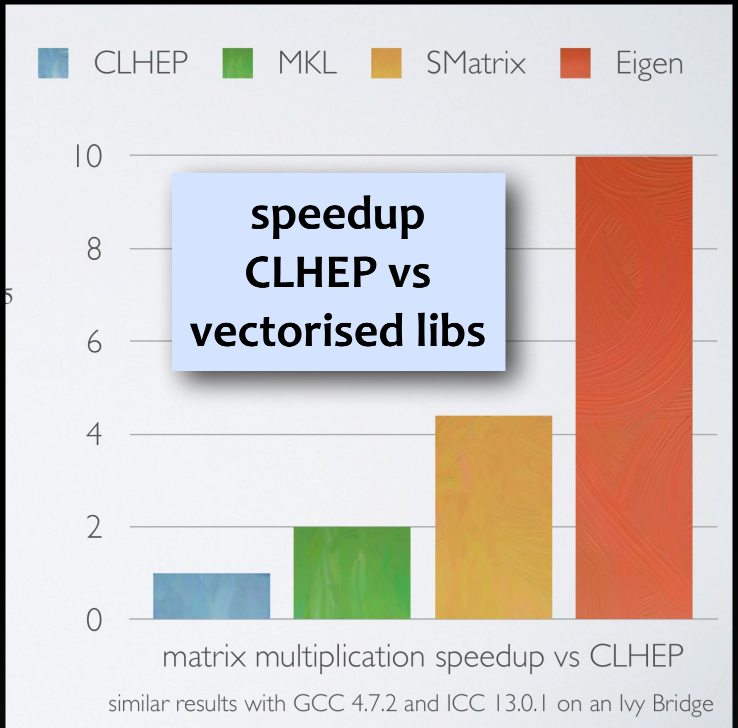


Borut Kersevan, Richard Mount et al.

- motivation for LS1 software upgrades
  - ➡ ensure Tier-0 can process 1kHz trigger rate, required to keep single lepton triggers
  - ➡ optimise disk usage (see new Analysis Model)
  - ➡ "soften" disk and CPU limits on Monte Carlo statistics
- focus here on preparation of tracking for 40 pileup

Tracking Developments towards Run-2

- ATLAS and CMS focus on technology and strategy to improve CURRENT algorithms
  - ➡ improve software technology, including:
    - simplify EDM design to be less OO (“hip” 10 years ago)
    - ATLAS migrated to Eigen - faster vector+matrix algebra (CMS was already using SMatrix)
    - vectorised trigonometric functions (CMS: VDT or ATLAS: intel math lib)
    - work on CPU hot spots (e.g. ATLAS replaced F90 by C++ for B-field service)
  - ➡ tune reconstruction strategy (very similar in ATLAS and CMS):
    - optimise iterative track finding strategy for 40 pileup
    - ATLAS modified track seeding to explore 4th Pixel layer
    - CMS added cluster-shape filter against out-of-time pileup



- hence, mix of SIMD and algorithm tuning
  - ➡ CMS made their tracking as well thread-safe

Tuning the Tracking Strategy

- optimal seeding strategy depends on level of pileup (ATLAS)
  - ➡ fraction of seeds to give a good track candidate:

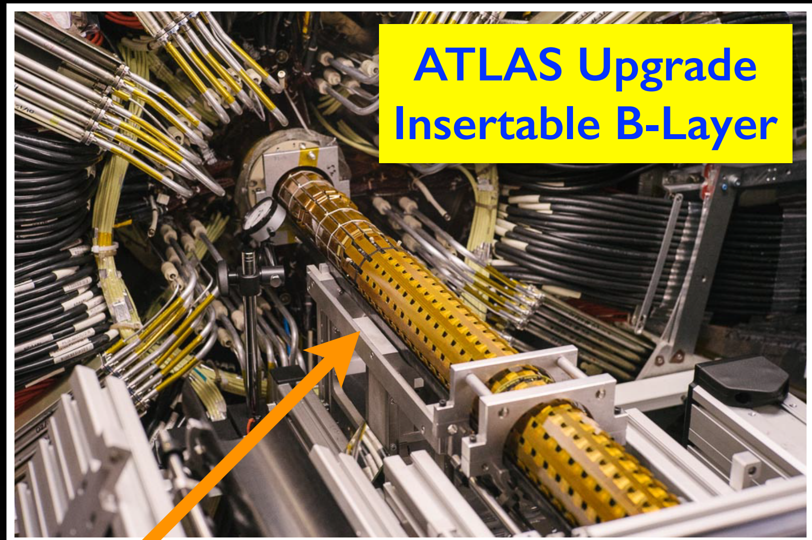
Seed-Triplets:
P = Pixel
S = Strips

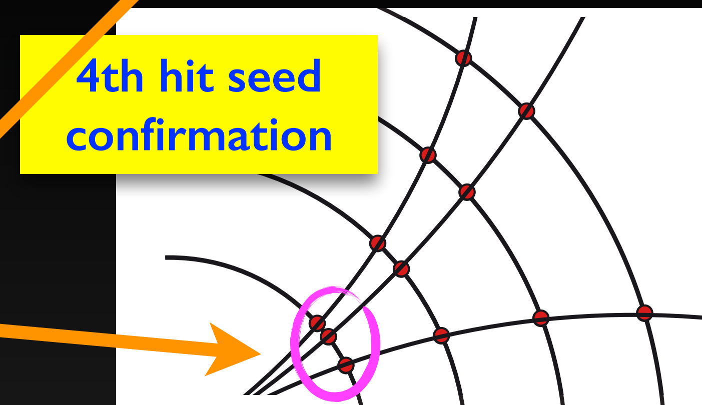
pileup	"PPP"	"PPS"	"PSS"	"SSS"
0	57%	26%	29%	66%
40	17%	6%	5%	35%

  - hence start with SSS at 40 pileup !
  - ➡ further increase good seed fraction using 4th hit

pileup	"PPP+1"	"PPS+1"	"PSS+1"	"SSS+1"
0	79%	53%	52%	86%
40	39%	8%	16%	70%

  - takes benefit from new Insertable B-Layer (IBL)

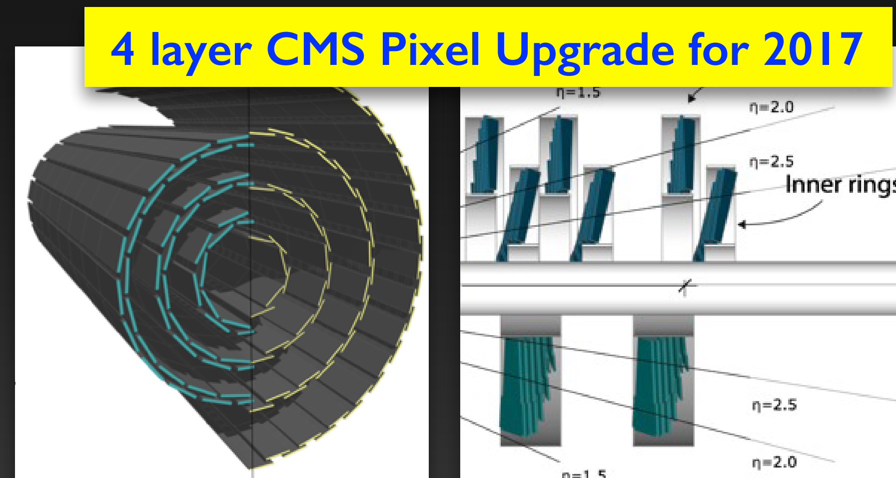




- final ATLAS Run-2 seeding strategy
  - ➡ significant speedup at 40 pileup (and 25 *nsec*)

seeding	efficiency	CPU*
"Run-1"	94.0%	9.5 sec
"Run-2"	94.2%	4.7 sec

*on local machine



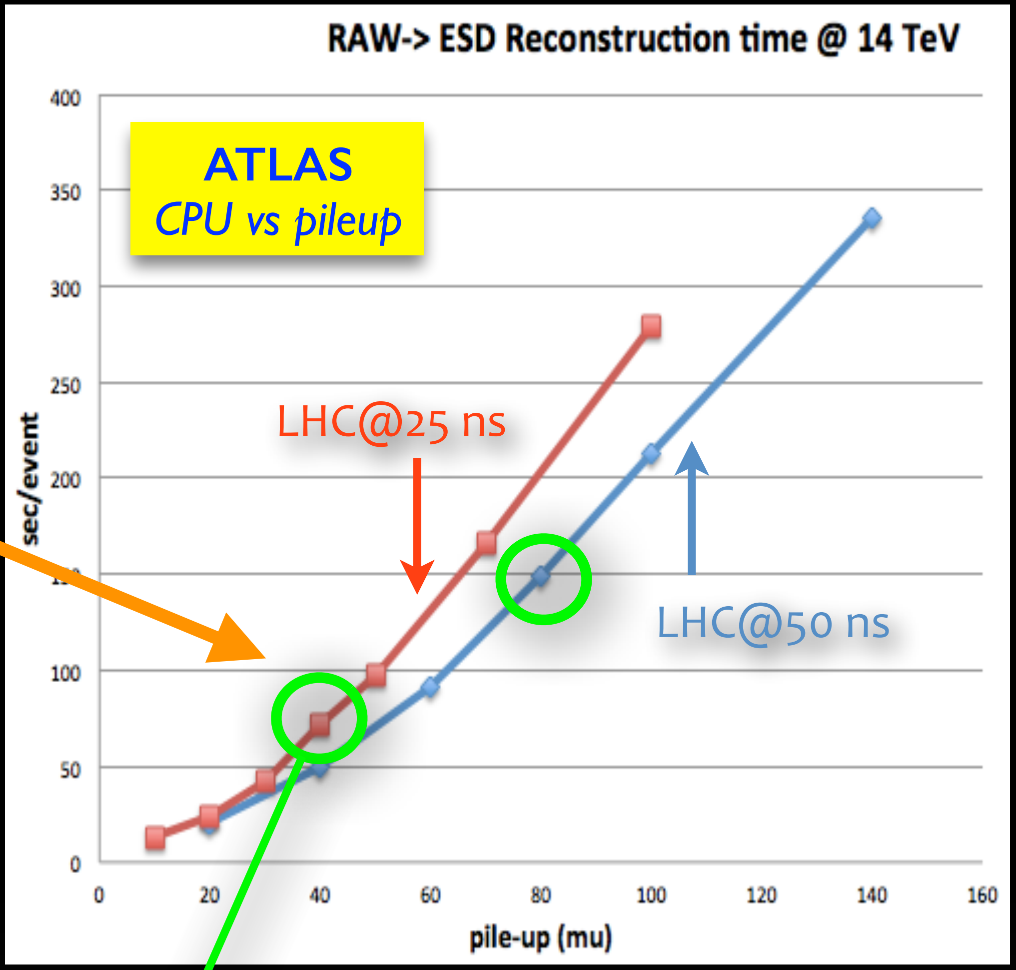
Overall CPU Improvements

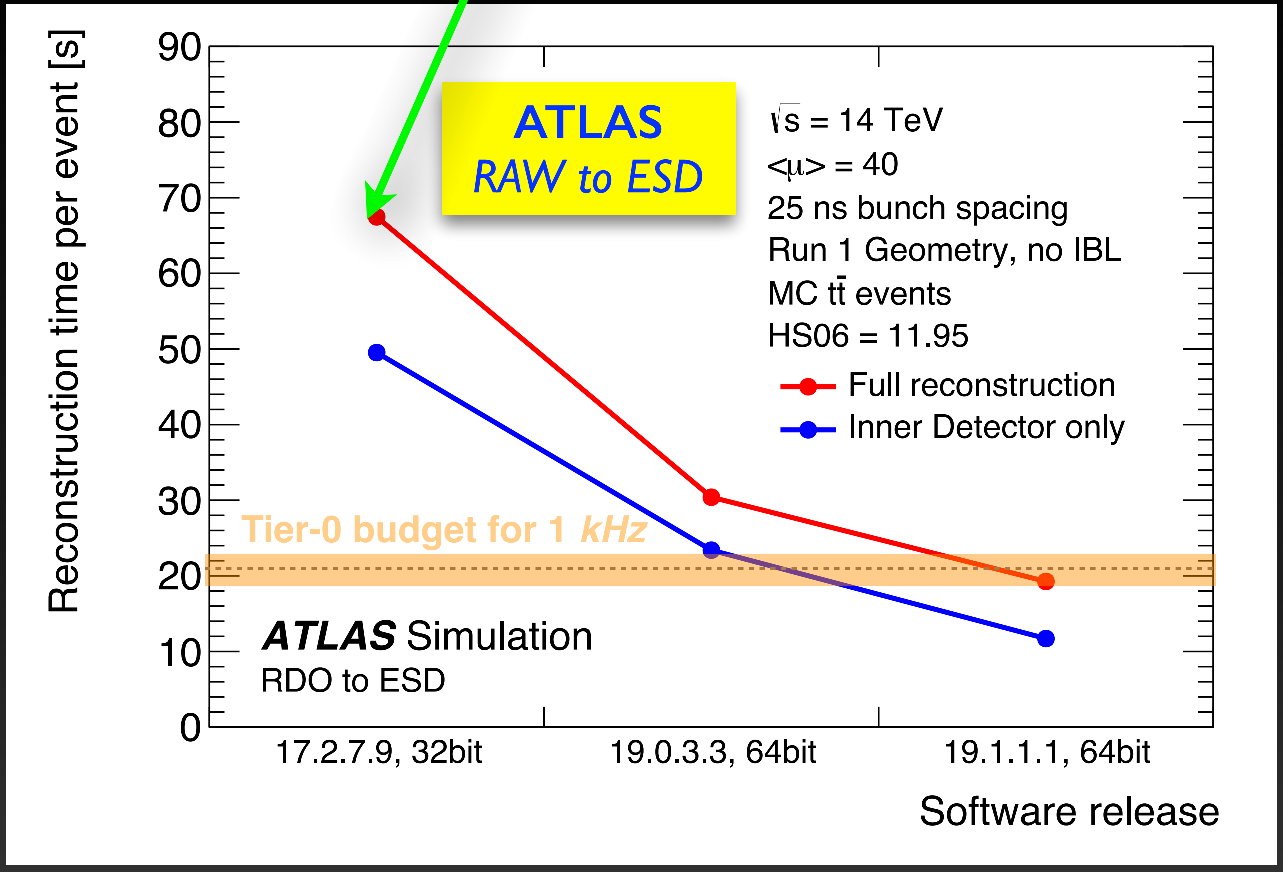
- result of ATLAS LS1 tracking upgrade
  - ➡ compare to Run-1 behaviour shown before
  - ➡ touched more than 1000 packages !
  - ➡ technical and strategy improvements for 40 pileup

- ATLAS reports factor 3 in CPU time

  (for tracking a factor 4)
  - ➡ benchmark releases using $t\bar{t}$ (14 *TeV*, μ=40):
    - 17.2.7.9-32bit is the 2012 Tier-0 release
    - 19.0.3.3 fully optimised for 8 *TeV*
    - 19.1.1.1. has setup for 13 *TeV* @ 40 pileup
  - ➡ 250 HS06/event within reach
    (CPU budget for 1 *kHz* @ Tier-0)

- CMS reports factor 2 in CPU
  - ➡ on top of what was achieved 2011/12
  - ➡ as well within 1 *kHz* Tier-0 budget

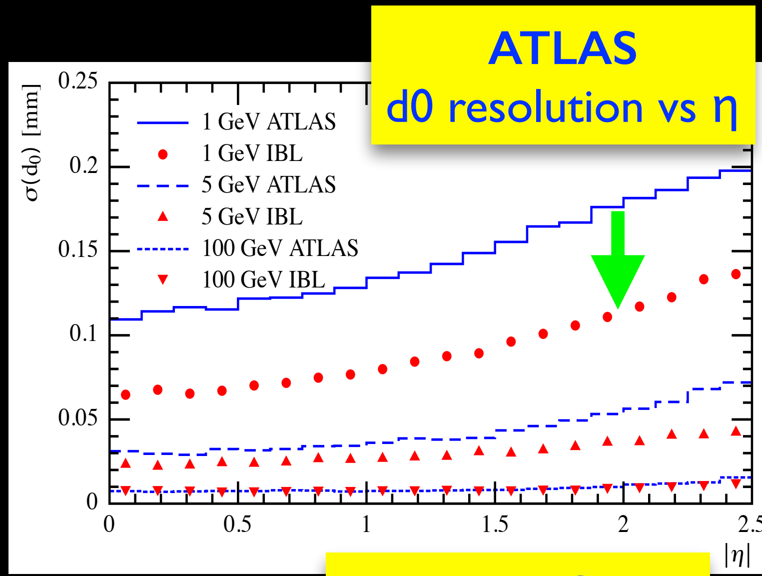


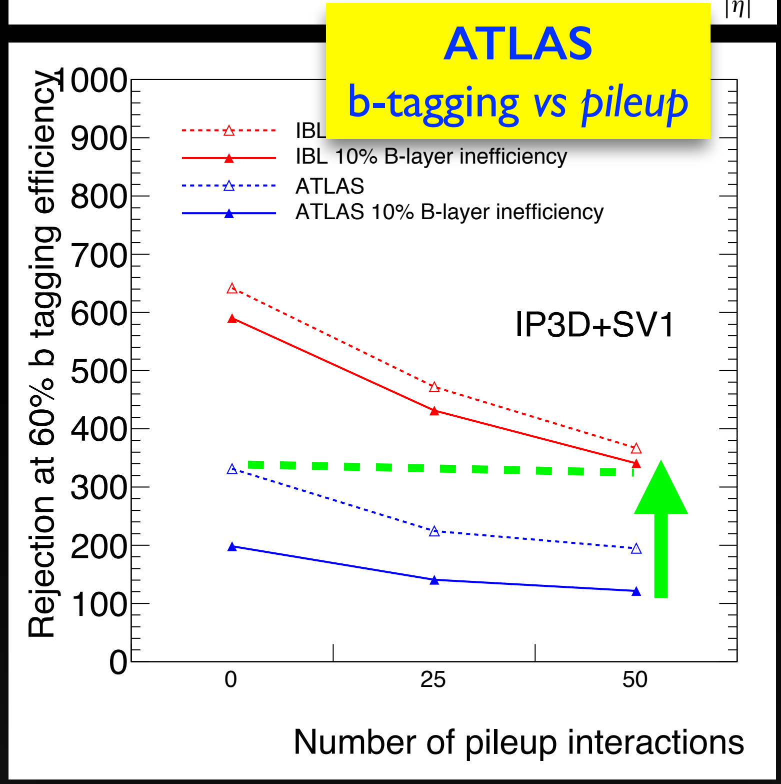


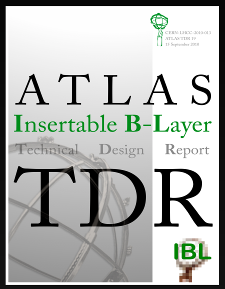
What is coming next ?

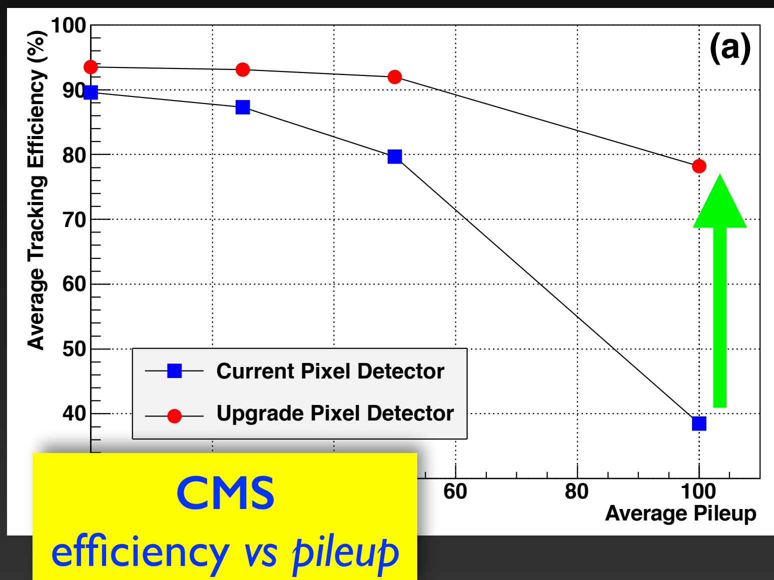
Pixel Upgrades - Performance

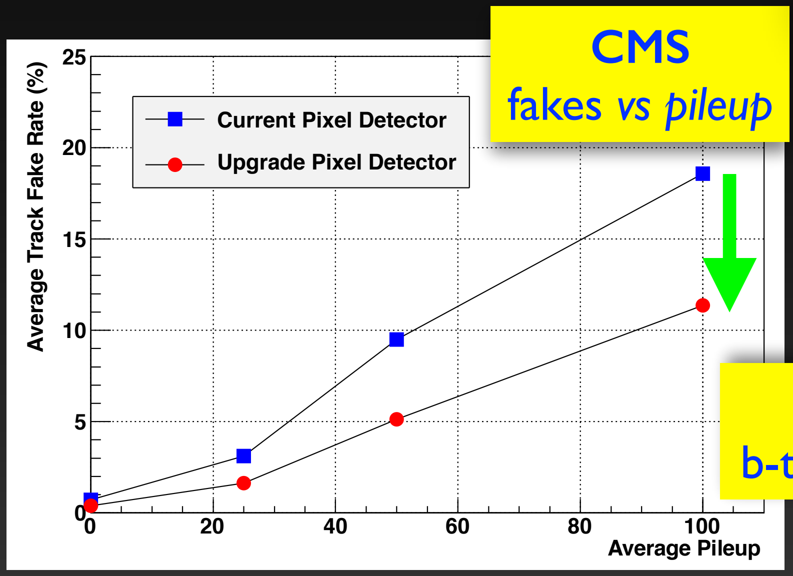
- aim is to mitigate effects of Run-2/3 pileup
  - ➡ ATLAS: IBL for 2015, CMS: new 4 layer Pixels for 2017
  - ➡ both experiments add low mass Pixel layer close to beam
    - improves impact parameter resolution
  - ➡ additional hit to reduce fakes and/or improve efficiency
    - and use 4th layer in seeding to reduce CPU
- significant improvements on b-tagging
  - ➡ at 50 pileup both experiments recover b-tagging performance like without pileup, or even improve upon it

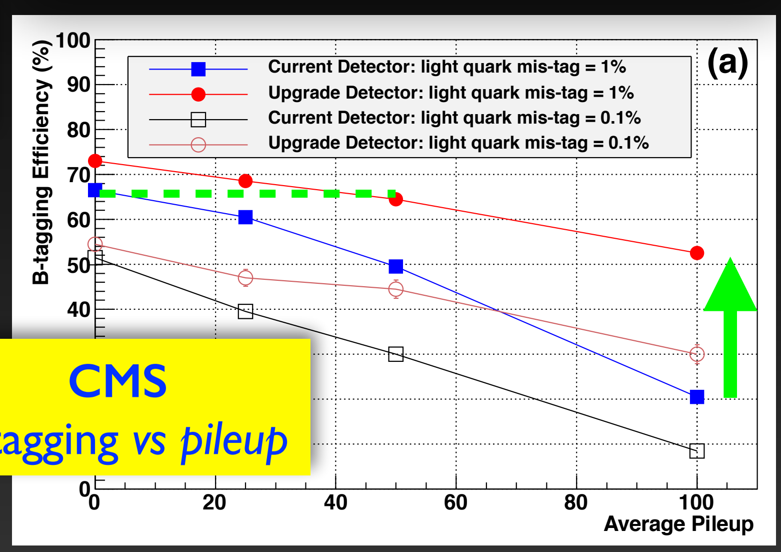










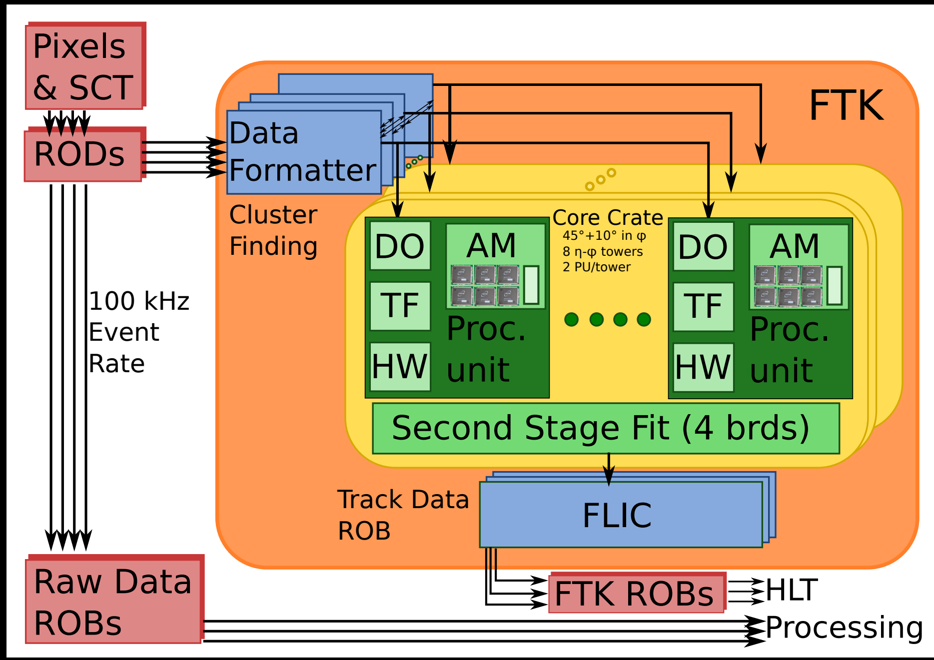


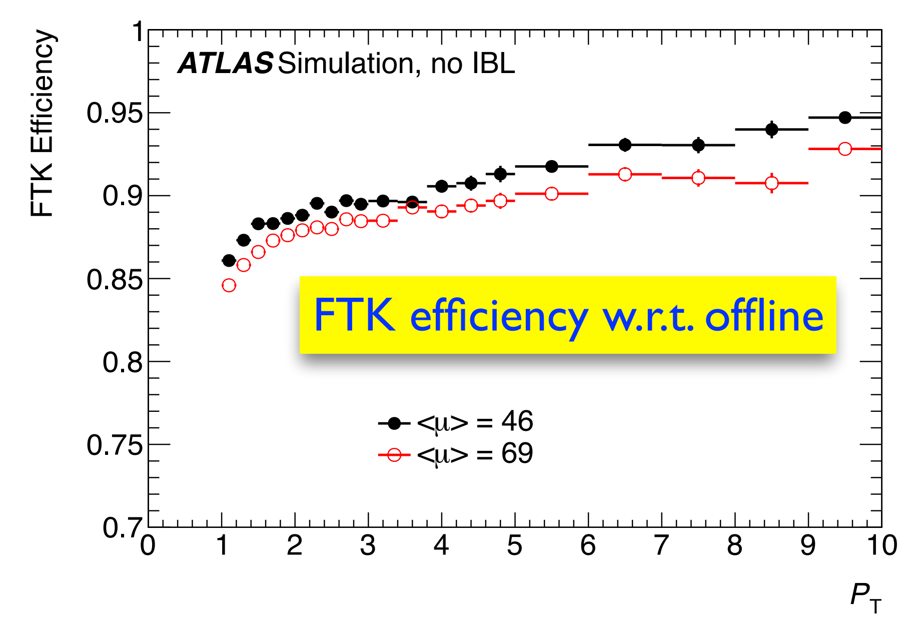
Hardware based Tracking ?

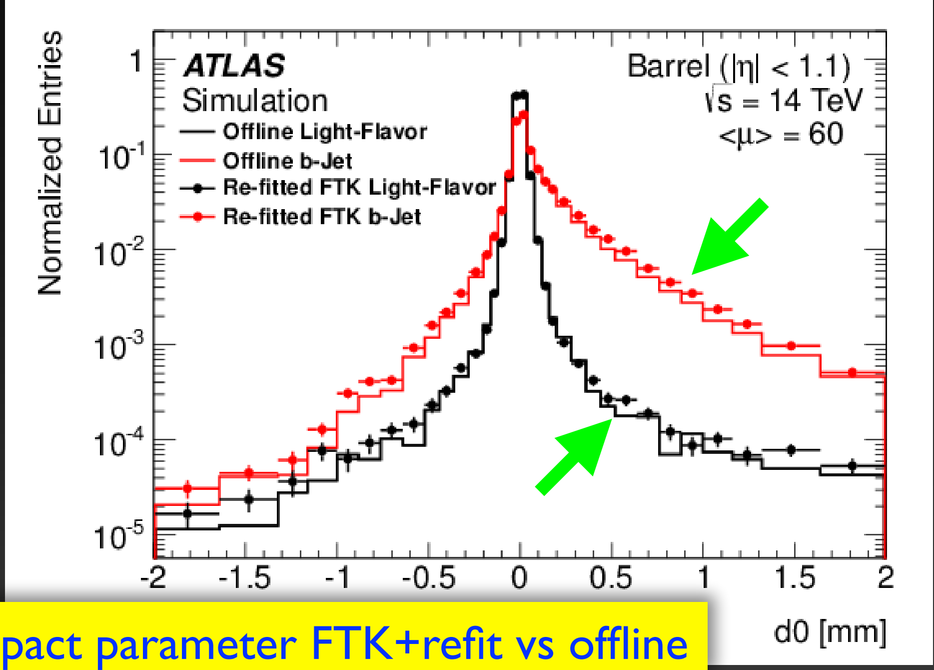
- current ATLAS trigger chain
  - ➡ Level-1: hardware based (~50 *kHz*)
  - ➡ Level-2: software based with RoI access to data (~5 *kHz*)
  - ➡ Event Filter: software trigger (~500 *Hz*)

tracking enters here

- ATLAS installs FTK during Run-2
  - ➡ hardware track reconstruction for Level-2 Trigger
    - associative memory (AM) chips to find patterns
    - FPGA based track parameter estimation
    - "Hit Worrier" (HW) to remove fakes
  - ➡ slice installed for 2015, full coverage in 2016
    - will replace software based Level-2 tracking in ATLAS
  - ➡ full event track reconstruction at latency of ~ 100 *μs*
    - fast track confirmation of Level-1 triggers
    - particle flow like tau tagging
    - fast b-jet tagging
    - pileup corrections for jets and missing ET
  - ➡ excellent performance for Level-2 purposes
    - track efficiency is 90-95% w.r.t. offline
    - track refit using full fitter recovers offline resolution

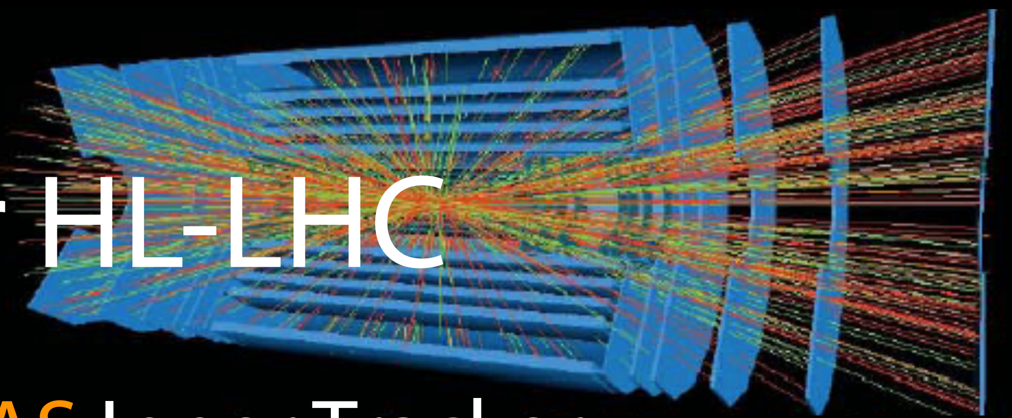






impact parameter FTK+refit vs offline

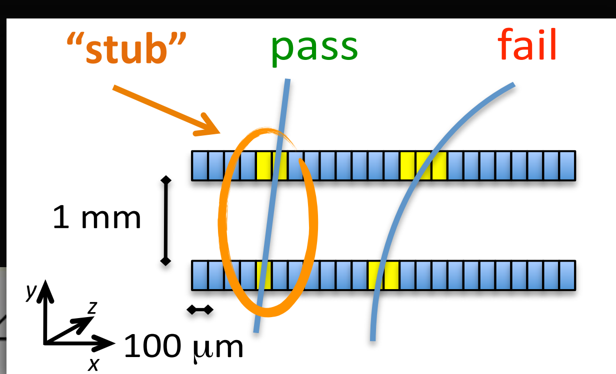
Inner Tracker Upgrades for HL-LHC

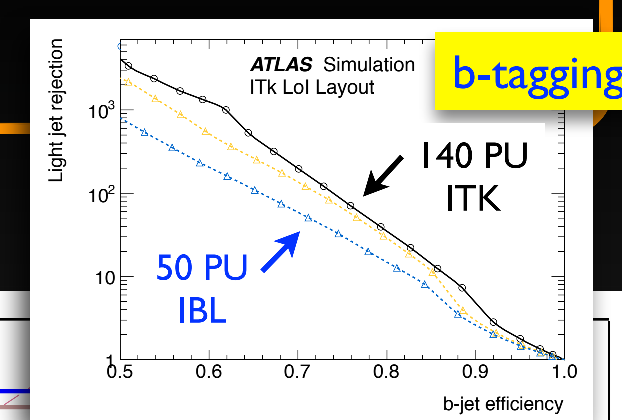


- CMS Inner Tracker
  - ➡ Strip tracker replacement
    - several layouts under consideration
    - short strips in $R\phi$, macro-pixels in z
  - ➡ Level-1 track trigger with high p_T stubs
    - correlate 2 sensors, threshold ~ *2 GeV*
    - pattern in FPGA or AM chips, FPGA fit
  - ➡ Pixels: extend η coverage to 4 (!)

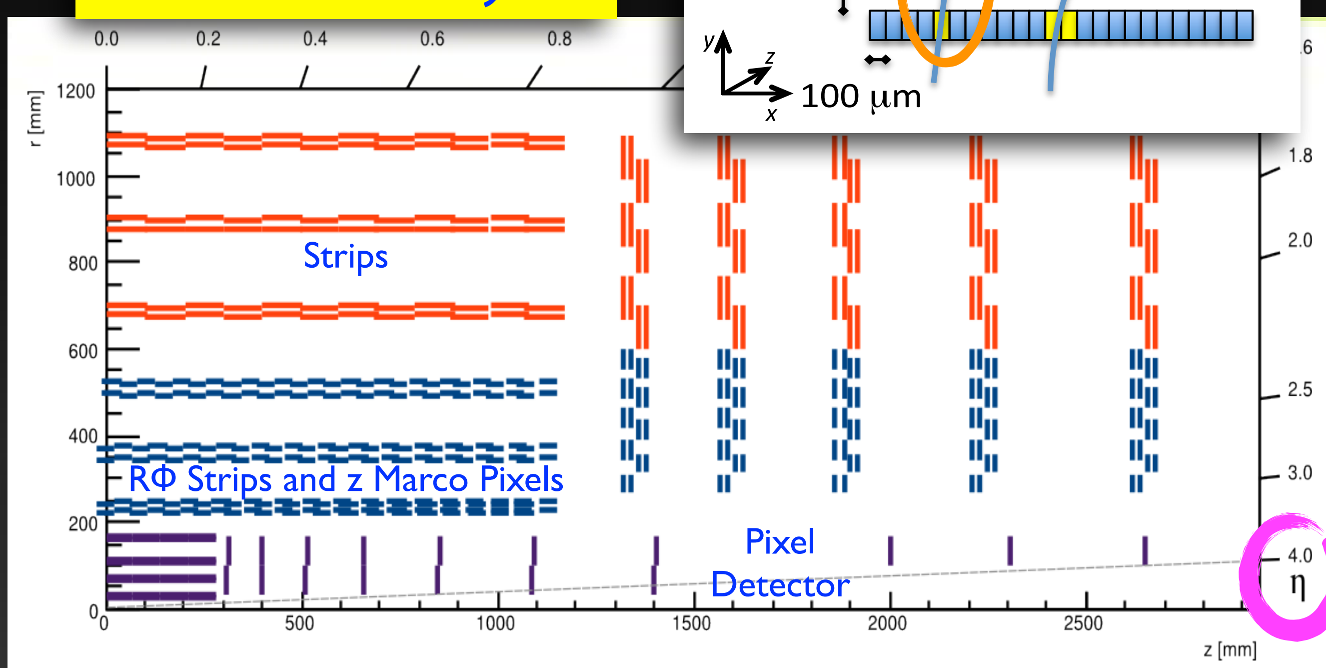
optimised for fast (HW) tracking

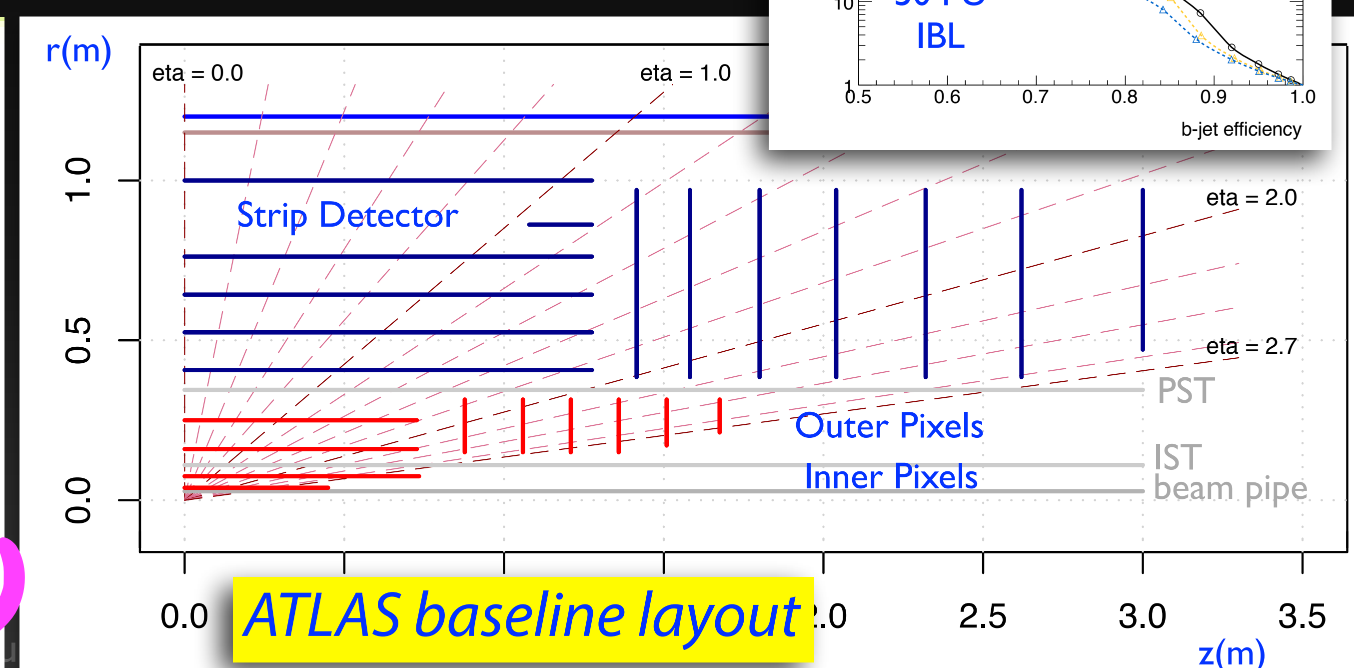
- ATLAS Inner Tracker
  - ➡ baseline: all silicon tracker, 14 hits
    - robust tracking @140 PU for η<2.5
  - ➡ Strip tracker with short strips + stereo
  - ➡ Pixels cover η<2.7 (Muons)
    - inner Pixels replaceable, reduced pitch
    - alternative layouts ("Alpine", conical)
  - ➡ Level-1 track trigger seeded by Level-0
    - FTK inspired, reduced latency





CMS baseline layout

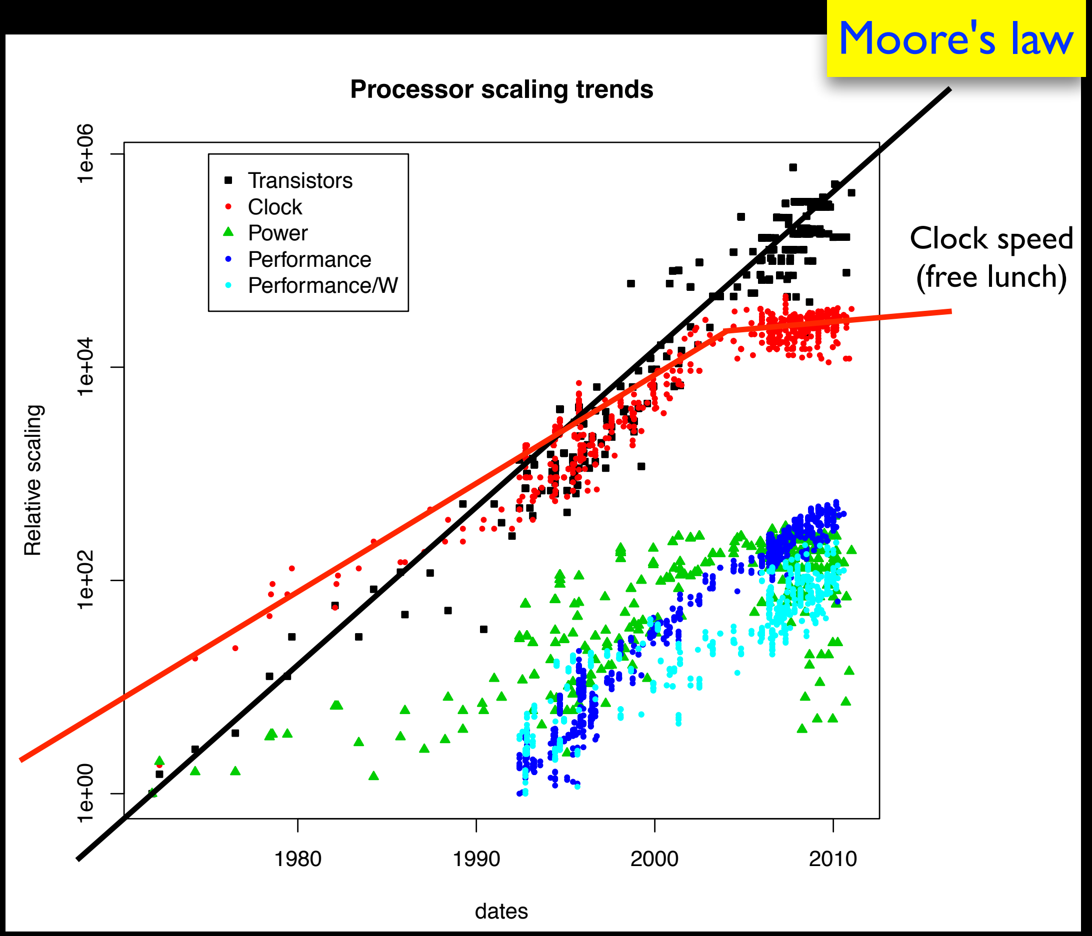




ATLAS baseline layout

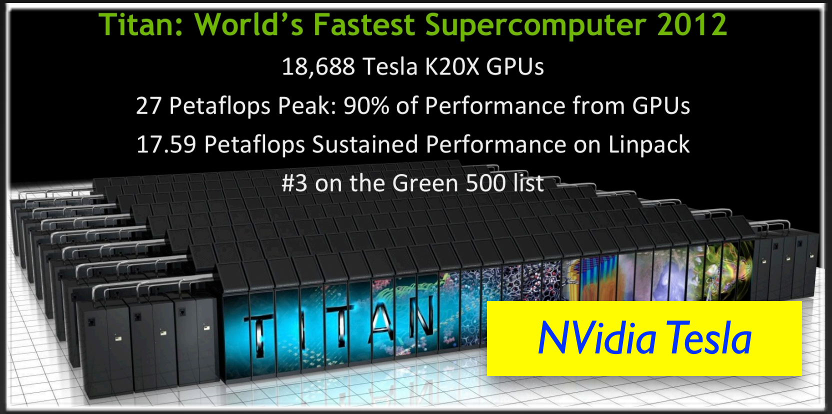
Processor Technology

- Moore's law is still alive
  - ➡ number of transistors doubles every 2 years
  - ➡ lots of transistors looking for something to do:
    - vector registers
    - out of order execution
    - multiple cores
    - hyper threading
  - ➡ increase theoretical performance of processors
    - hard to achieve this performance with HEP applications

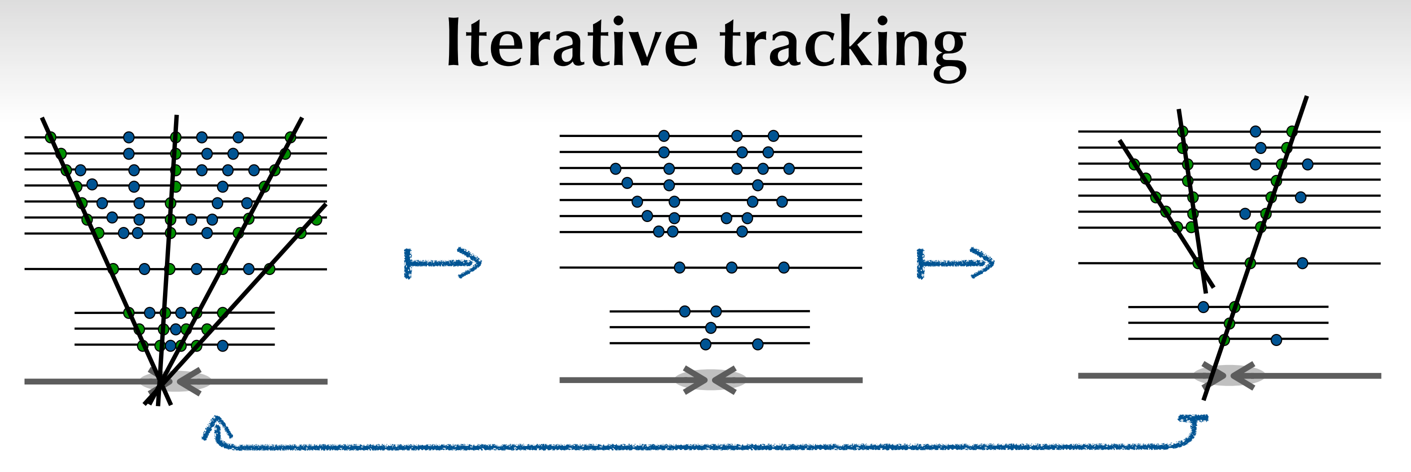


- taking benefit from vector registers (SIMD)
  - ➡ LS1: Eigen and Intel math lib used in ATLAS, VDT in CMS

- many-core processors, including GPGPUs
  - ➡ e.g. NVidia Tesla, Intel Xenon Phi
    - one sees them in High Performance Computing (HPC)
  - ➡ lots of cores with less memory
    - same for ARM or ATOM processors with small memory
    - need to parallelise applications (multi-threading)



Massively parallel Tracking ?



- ATLAS/CMS tracking strategy is for early rejection
  - ➡ iterative tracking: avoid combinatorial overhead as much as possible !
    - early rejection requires strategic candidate processing and hit removal
  - ➡ not a heavily parallel approach, it is a SEQUENTIAL approach !
- implications for making it massively parallel ?
  - ➡ Armdahl's law at work:

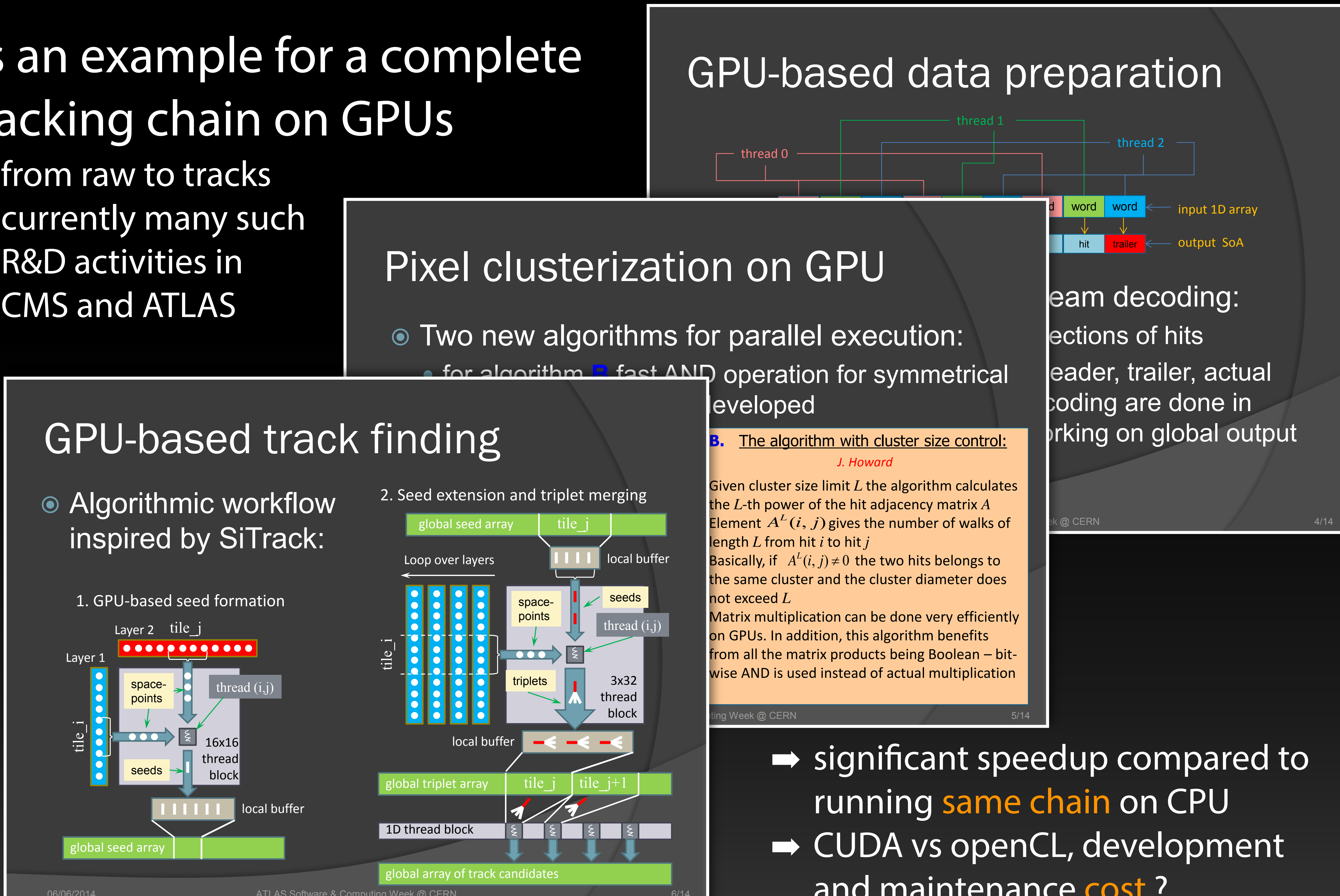
$$\text{Time}_{||} = \text{Para} / N + \text{Seq}$$

  - ➡ iterative tracking: small parallel part Para, heavy on sequential Seq
    - hence, if we want to gain by a large N threads, we need to reduce Seq
- CMS study: run combinatorial filter in parallel for seeds
  - ➡ find compromise on early rejection, but still limit combinatorial overhead
    - as a result, one spends somewhat more CPU, main gain is in memory
  - ➡ promising if one uses additional processing power that otherwise would not be usable (many core processors) or if latency is the main issue (trigger)

ATLAS Level-2 GPU Tracking Prototype

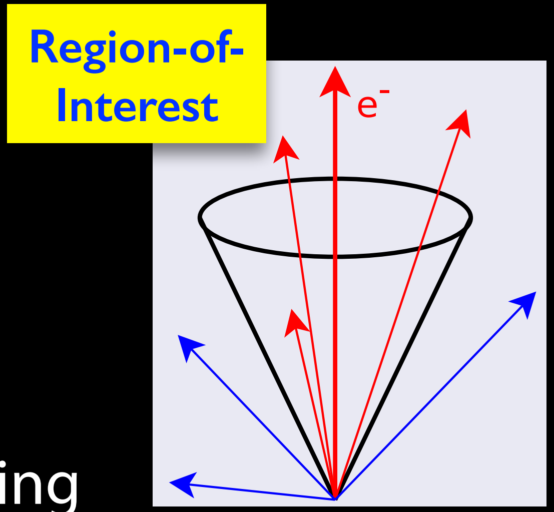
for completeness

- as an example for a complete tracking chain on GPUs
  - ➡ from raw to tracks
  - ➡ currently many such R&D activities in CMS and ATLAS

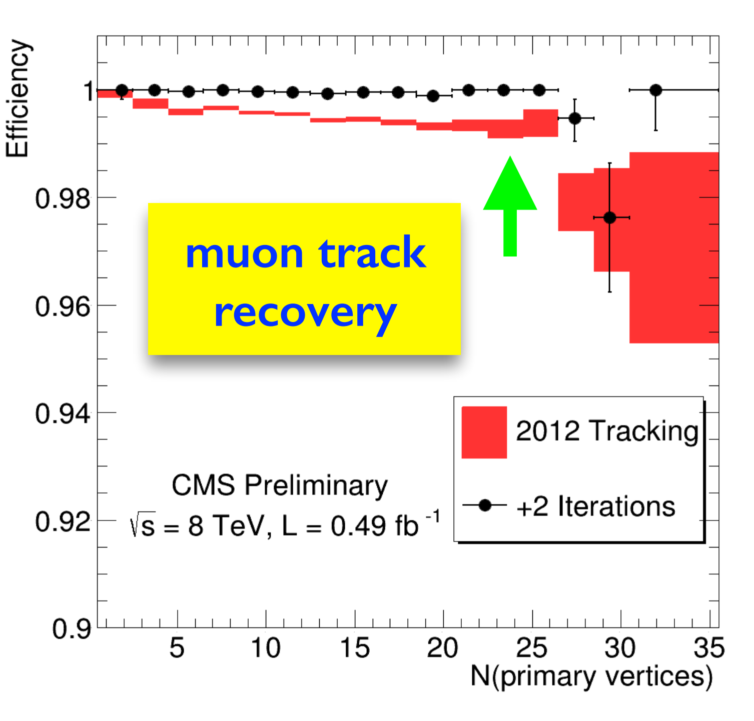


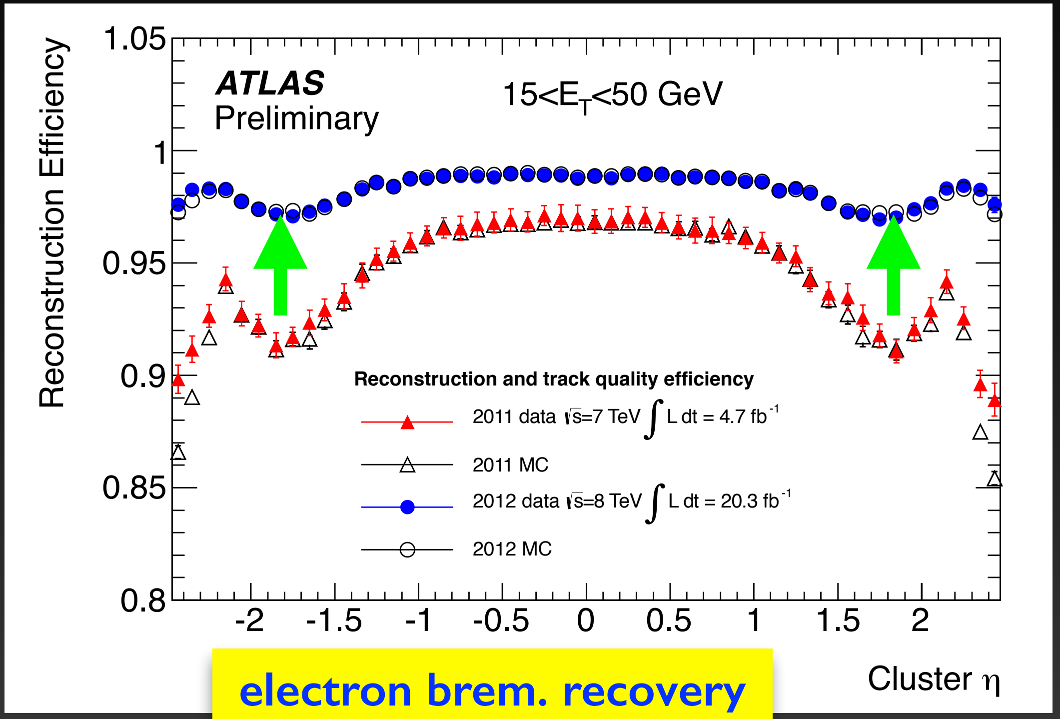
- ➡ significant speedup compared to running same chain on CPU
- ➡ CUDA vs openCL, development and maintenance cost ?

Tracking Algorithms for High Pileup



- alternative tracking techniques for parallelisation ?
  - ➡ CMS investigated using Hough Transforms, limited by multiple scattering
- tracking according to physics needs ?
  - ➡ idea: run different tracking inside/outside Region-of-Interest
    - best possible tracking for signal event or region
    - faster, approximate tracking on pileup and underlying event (extreme: truth guided tracking on MC to avoid pattern overhead)
  - ➡ experiments already started doing this in Run-1 !
    - CMS runs tracking passes to recover efficiency for muons
    - ATLAS runs brem. recovery for tracks pointing to EM clusters
  - ➡ and for Run-2
    - ATLAS regional tracking for photon conversions
    - both experiments are working on dedicated tracking in jets
- future ATLAS simulation
  - ➡ Integrated Simulation Framework (ISF)
    - fast and full simulation for different parts of an event
    - matches tracking in regions
    - huge potential for CPU savings



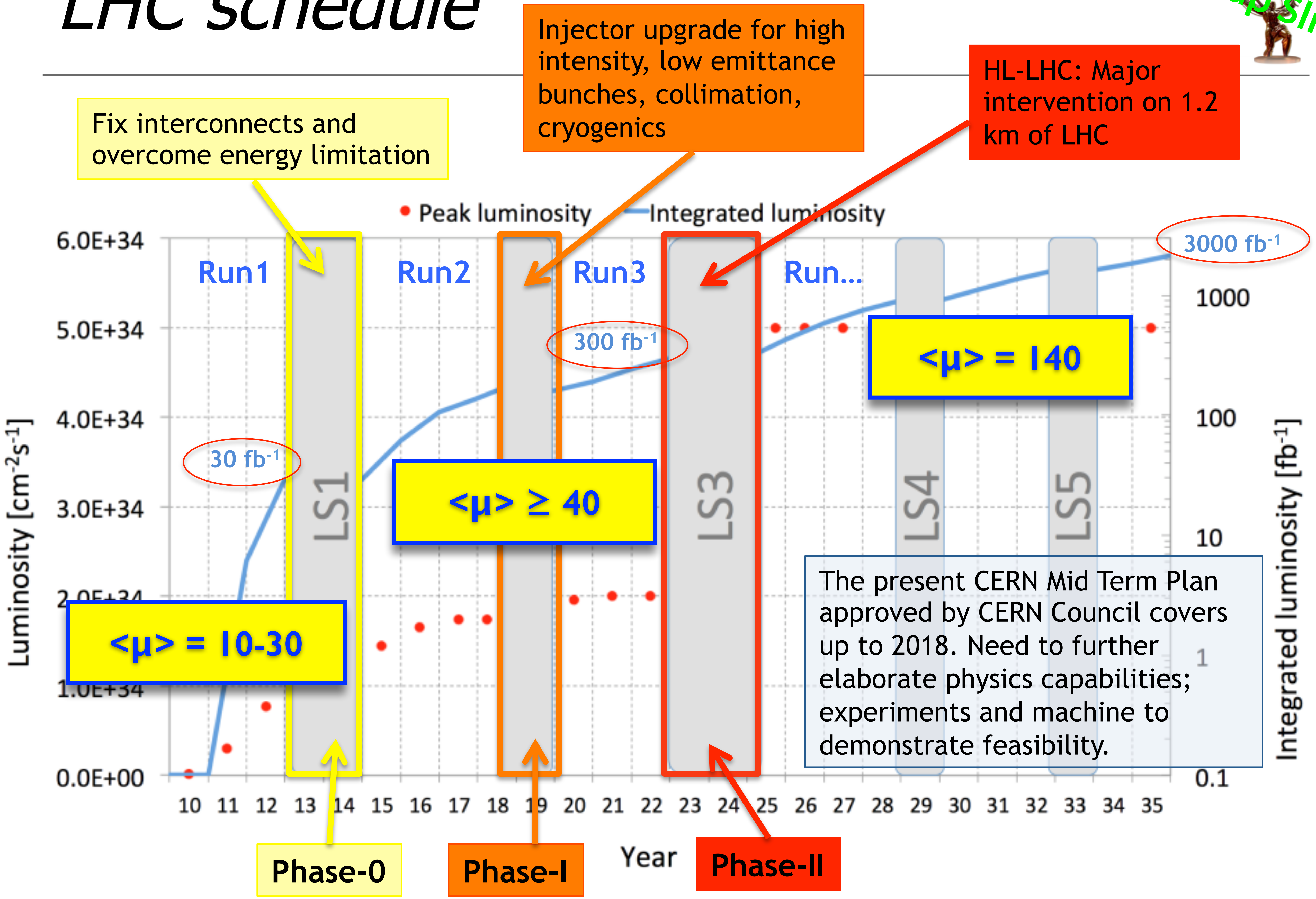


Summary and Outlook

- excellent tracking performance during Run-1
  - ➡ ATLAS and CMS use very similar (silicon) tracking techniques
    - both experiments optimised technical performance and strategy in LS1
  - ➡ experiments ready to meet performance and CPU requirements for Run-2
- Pixel upgrades will further mitigate effects of pileup
  - ➡ ATLAS will as well deploy FTK as hardware tracking for Level-2 Trigger
- evolution of processor technology towards many-core
  - ➡ need to parallelise tracking to take benefit
  - ➡ R&D on algorithms, especially on tracking on GPUs
- algorithm developments for very high pileup
  - ➡ experiments introduced already specialised tracking in Regions-of-Interest
  - ➡ forum to discuss algorithm developments across experiments is lacking
- proposal to organise dedicated workshop(s) in Vienna (?)
- biggest concern (as usual in software) is manpower

LHC schedule

Backup Slide

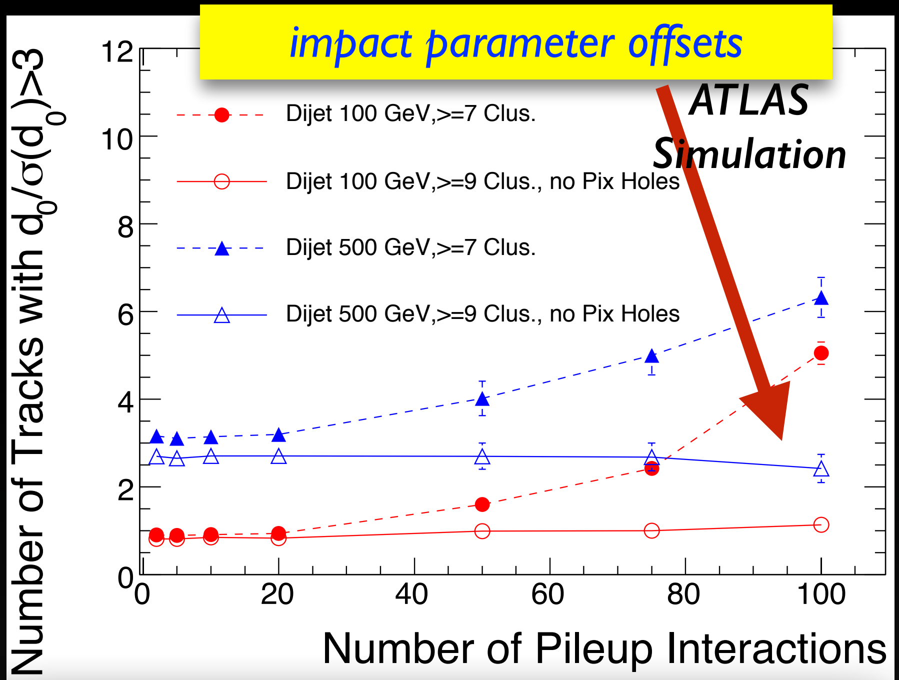


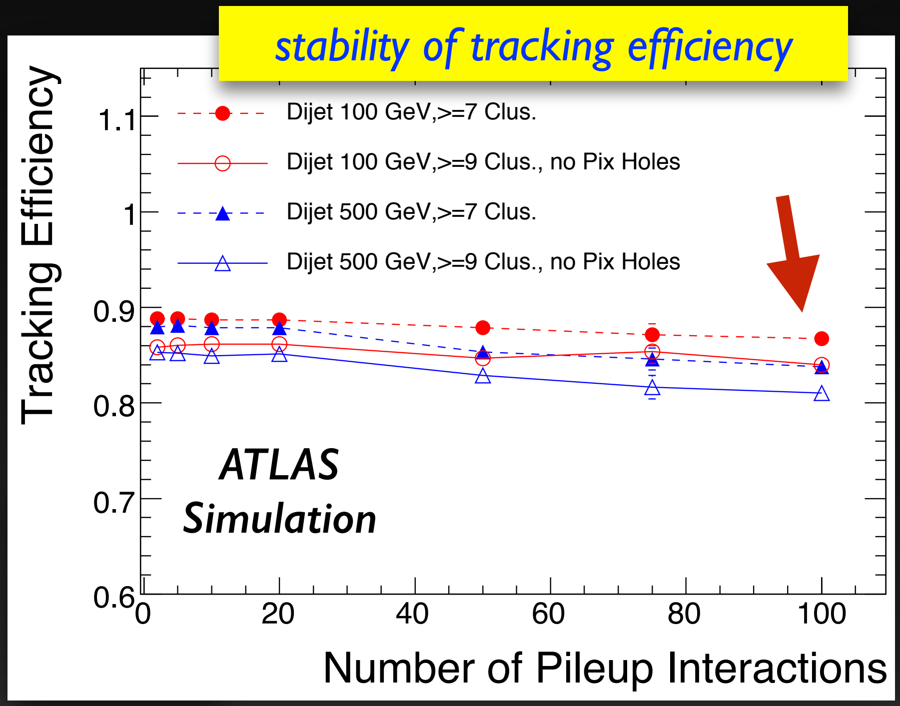
LHC schedule approved by CERN management and LHC experiments spokespersons and technical coordinators (December 2013)

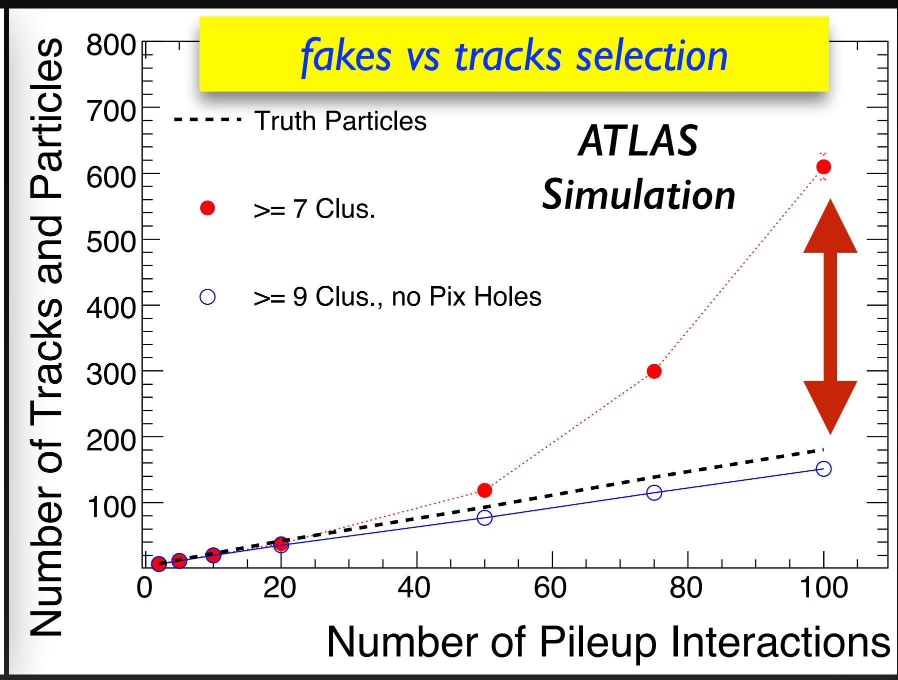
C.Gemme, LHCP

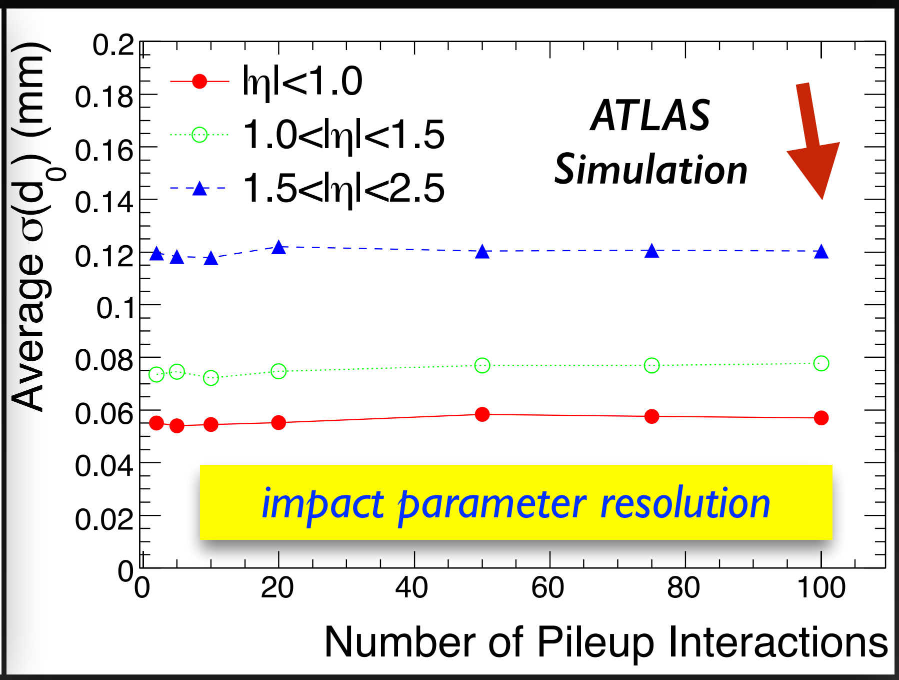
Expected Tracking Performance vs Pileup

- affects on tracking in current detector
  - ➡ pileup affects physics performance if reconstruction unchanged
    - adjusting track selection allows to mitigate effects
  - ➡ studied extensively even pre-data taking (see plots)
- current tracker ok until ~100 pileup
  - ➡ no effects on efficiencies or resolutions
  - ➡ control fakes and fake impact offsets with tracking cuts



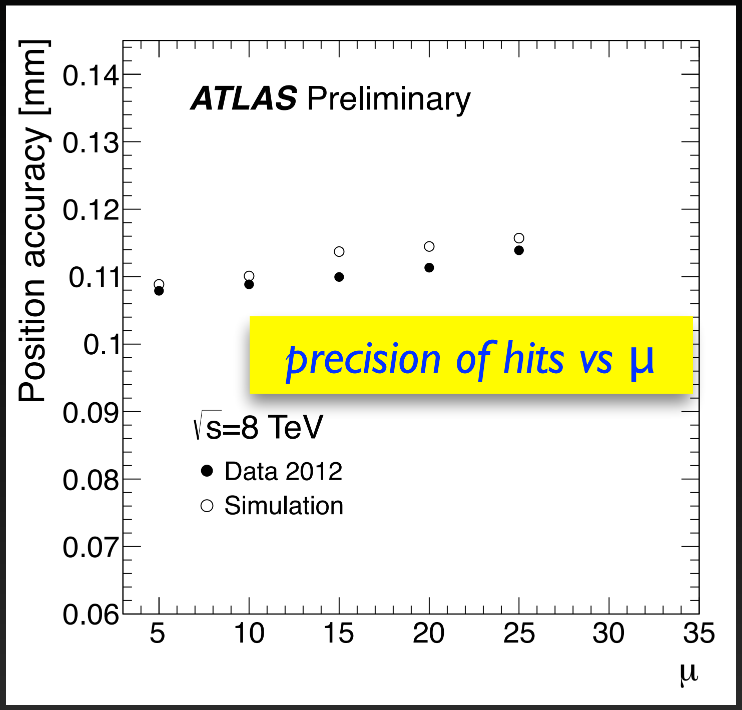


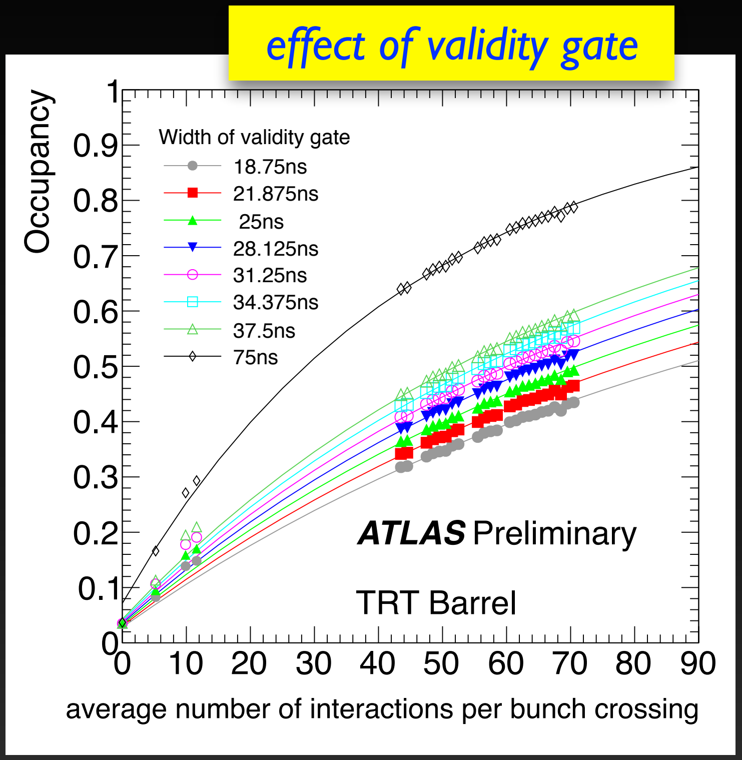


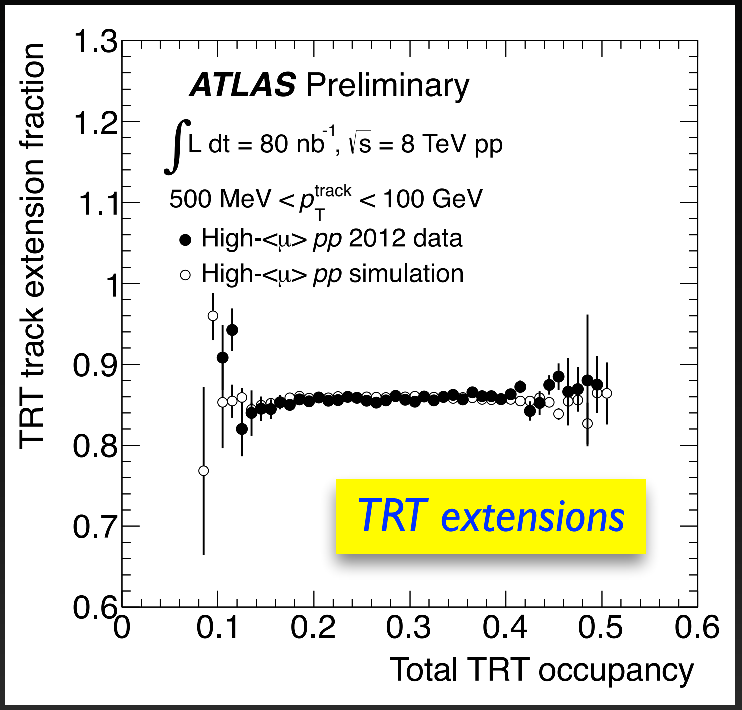


ATLAS TRT Performance at High Pileup

- TRT is designed for high occupancy
  - ➡ tracking uses precision hits (leading edge)
    - hit precision not much affected by pileup
    - some shadowing of at very high <μ>
  - ➡ use trailing edge to establish validity gate against out of time pileup
- fraction of silicon tracks extended into TRT quite stable







Introduction: NewTracking in ATLAS

Backup Slide

pre-precessing
- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation

combinatorial track finder
- ➡ iterative :
  1. Pixel seeds
  2. Pixel+SCT seeds
  3. SCT seeds
- ➡ restricted to roads
- ➡ bookkeeping to avoid duplicate candidates

ambiguity solution
- ➡ precise least square fit with full geometry
- ➡ selection of best silicon tracks using:
  1. hit content, holes
  2. number of shared hits
  3. fit quality...

extension into TRT
- ➡ progressive finder
- ➡ refit of track and selection

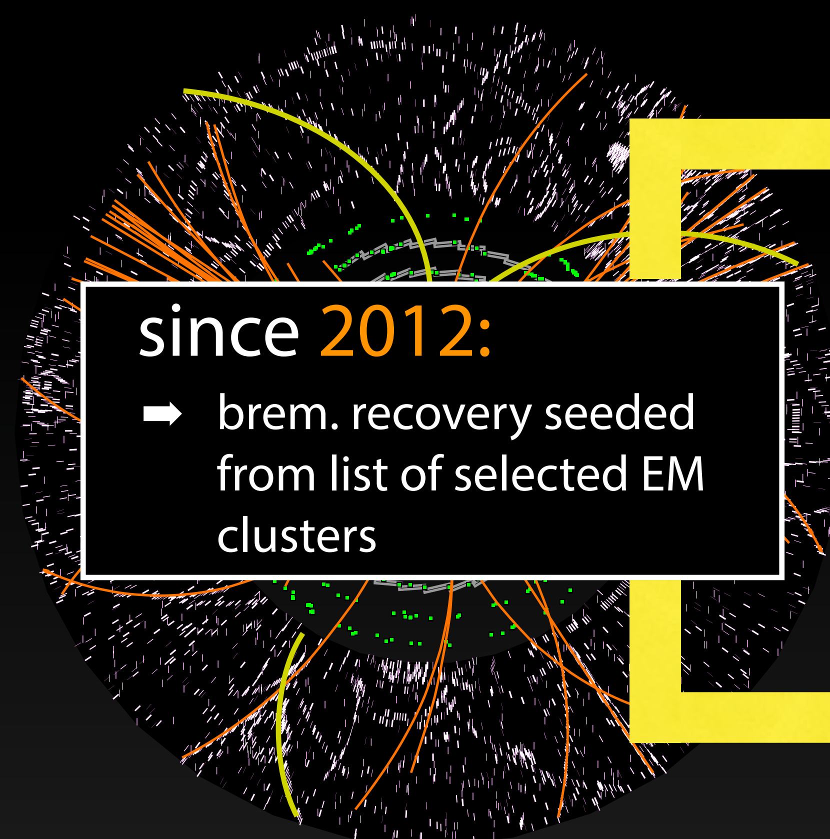
TRT segment finder
- ➡ on remaining drift circles
- ➡ uses Hough transform

TRT seeded finder
- ➡ from TRT into SCT+Pixels
- ➡ combinatorial finder

ambiguity solution
- ➡ precise fit and selection
- ➡ TRT seeded tracks

standalone TRT
- ➡ unused TRT segments

vertexing
- ➡ primary vertexing
- ➡ conversion and V0 search

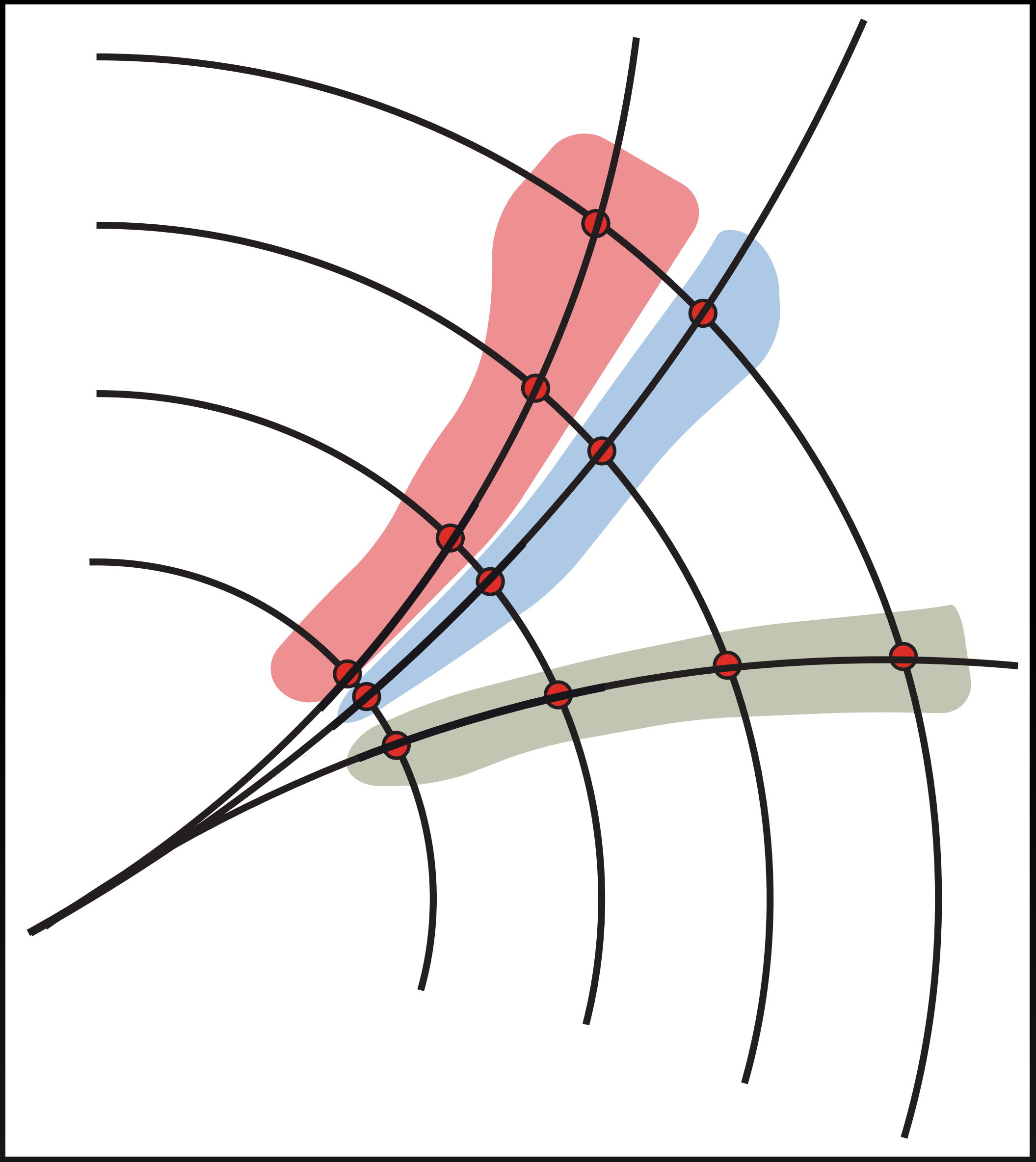


since 2012:
- ➡ brem. recovery seeded from list of selected EM clusters

Backup Slide

Tuning the Seeding Strategy

- the track finding algorithm
  - ➡ find **seed** from combination of 3 hits
    - search using hough transform
  - ➡ build **road** along the likely trajectory
  - ➡ run **combinatorial Kalman Filter** for a seed
    - full **exploration** of all possible candidates
    - update trajectory with hits at each layer
    - take material effects into account



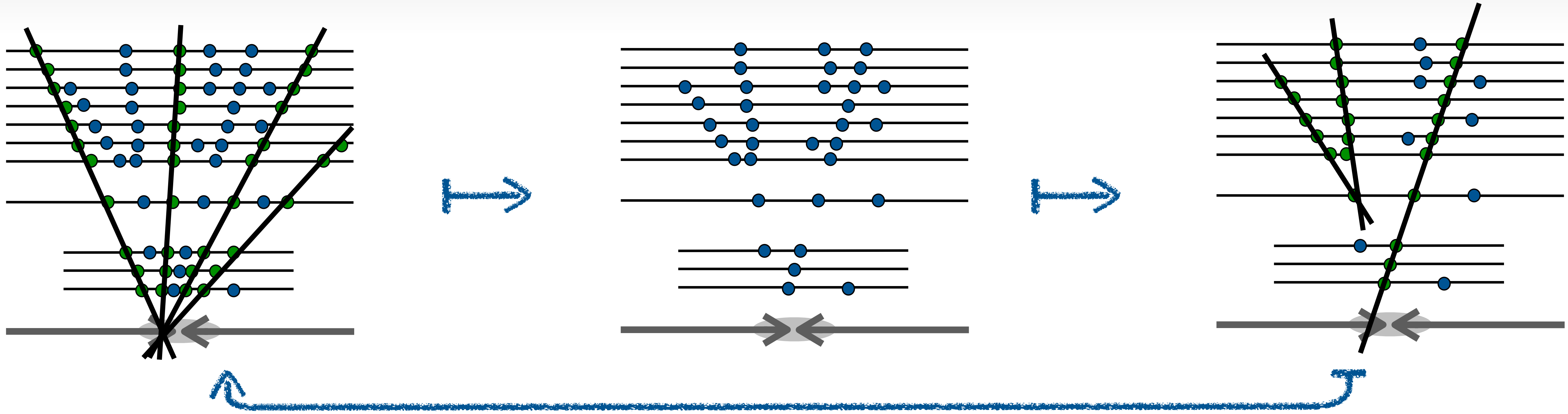
- iterative seeding approach (Run-1)
  - ➡ seeds are worked on in an ordered list
    - start with 3 Pixels, 2 Pixel+Strip, 3 Strips
  - ➡ bookkeeping layer:
    - hits from good candidates removed
    - build next seed ONLY from left over hits
  - ➡ sequential seed finding to avoid combinatorial explosion
    - unlike in the animation, tracks are found for one-after-the-other
    - hence, the ordering matters !!! (especially sorting in p_T bins)

Sguazzoni et al.,

GSI Tracking Workshop 2012

Iterative tracking

Backup Slide



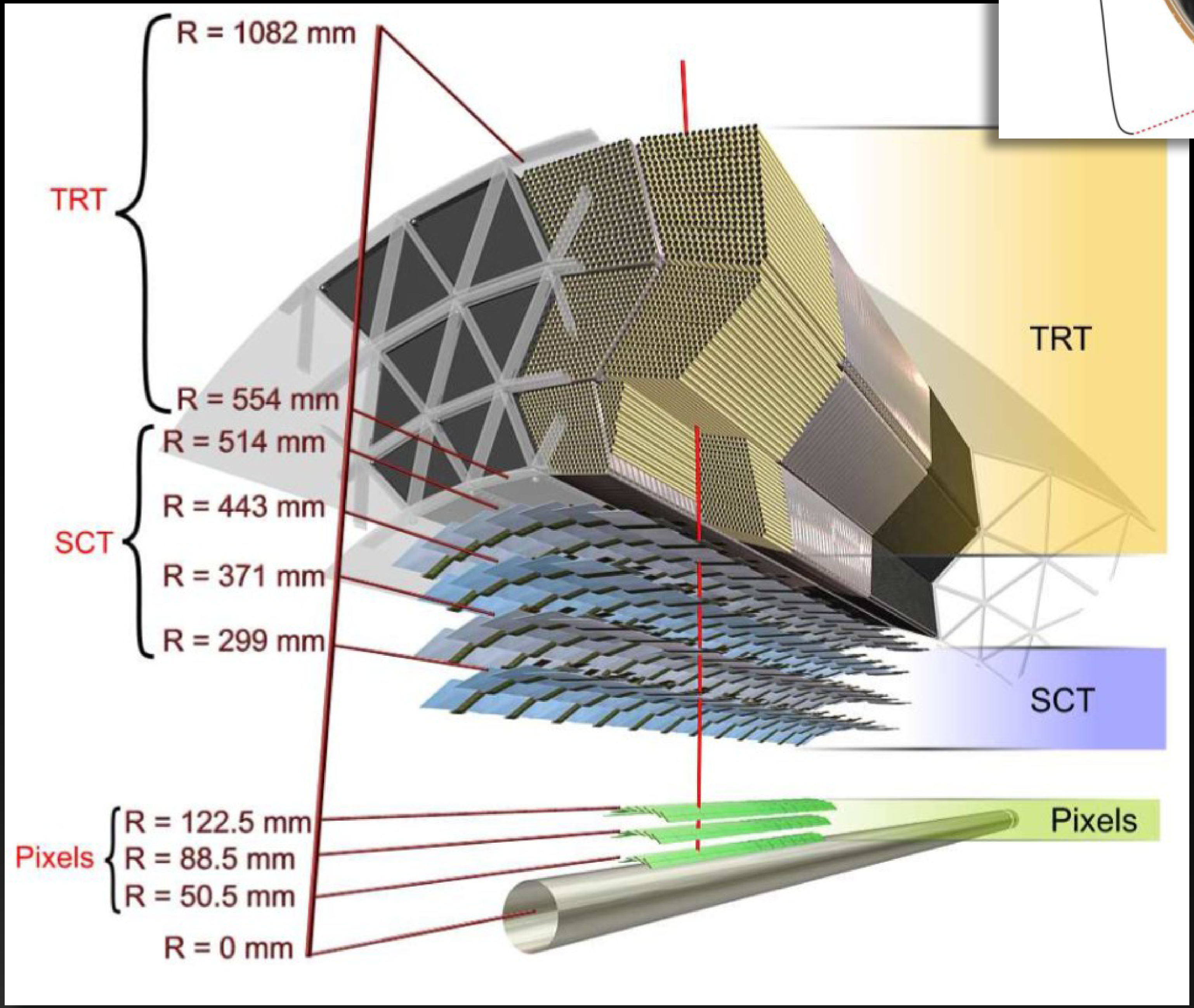
The CMS tracking relies on iterations (*steps*) of the tracking procedure; each step works on the remaining not-yet-associated hits and is optimized with respect to the seeding topology and to the final quality cuts.

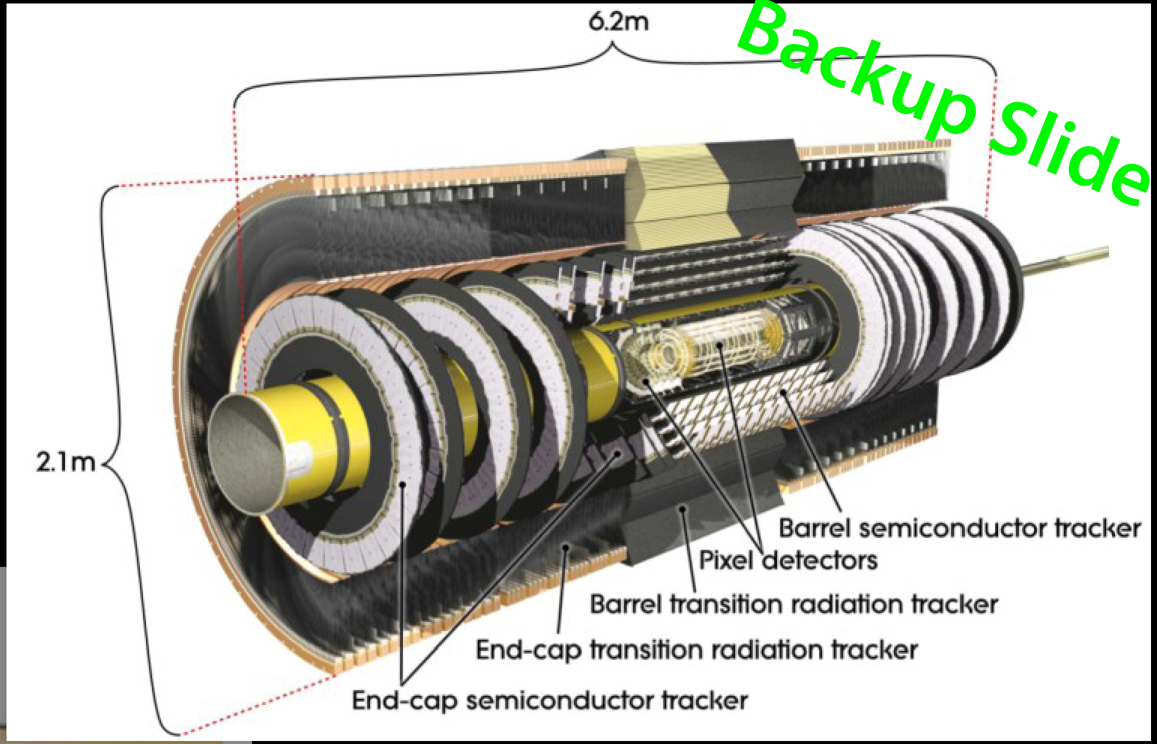
#step	seed type	seed subdetectors	P_T^{min} [GeV/c]	d_0 cut	z_0 cut
0	triplet	pixel	0.6	**0.02** cm	4.0σ
1	triplet	pixel	0.2	**0.02** cm	4.0σ
2	pair	pixel	0.6	**0.015** cm	0.09 cm
3	triplet	pixel	**0.3**	**1.5** cm	**2.5**σ
4	triplet	pixel/TIB/TID/TEC	**0.5-0.6**	**1.5** cm	10.0 cm
5	pair	TIB/TID/TEC	0.6	2.0 cm	10.0 cm
6	pair	TOB/TEC	0.6	2.0 cm	30.0 cm

Iterative tracking in 2012 (CMSSW 52x) / In **bold** the changes with respect to 44x

ATLAS Inner Detector

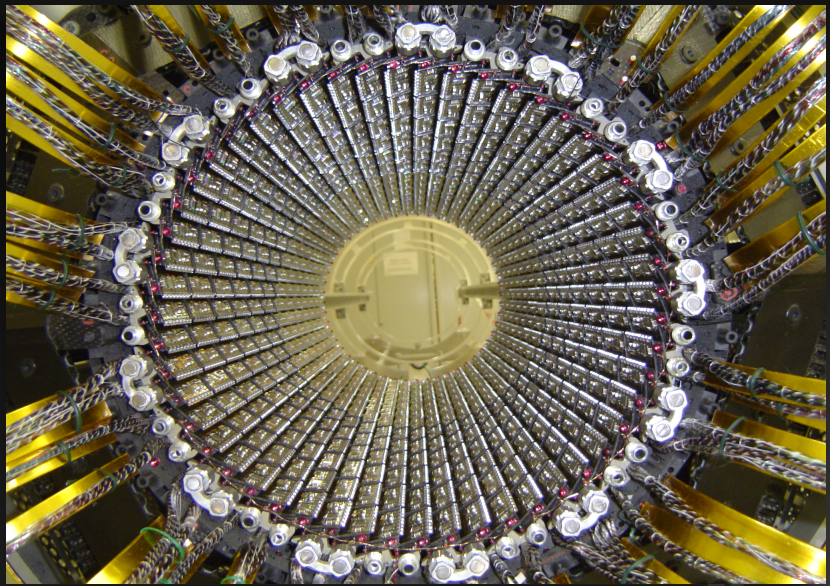
- optimised for 24 pileup events



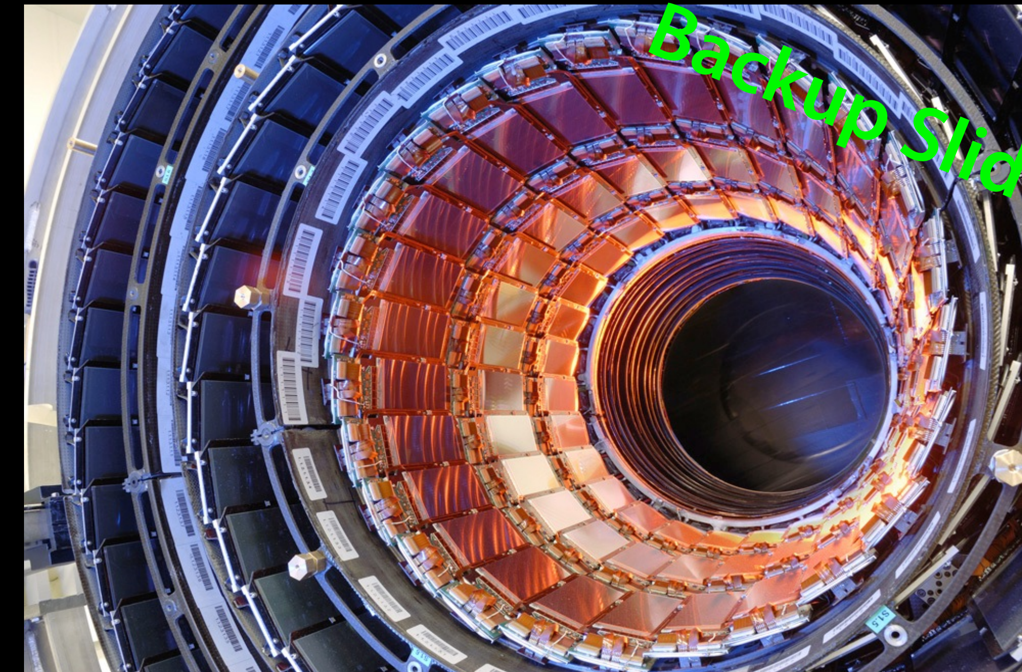


Backup Slide

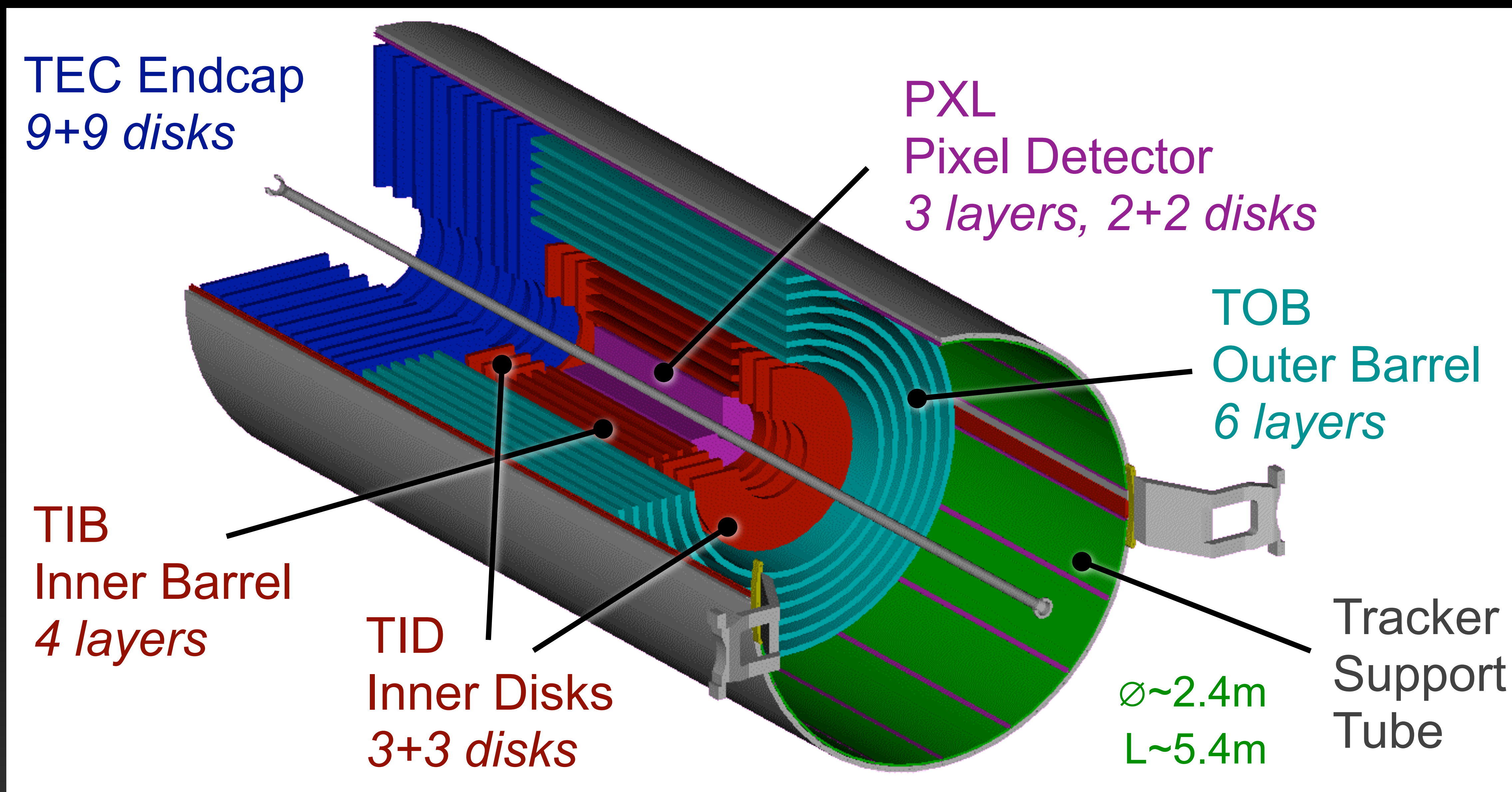
- barrel track passes:
  - ➡ ~36 TRT 4mm straws
  - ➡ 4x2 Si strips on stereo modules12cm x 80 mm, 285mm thick
  - ➡ 3 pixel layers, 250mm thick



CMS Tracker



- largest silicon tracker ever built
  - ➡ **Pixels:** 66M channels, 100x150 μm² Pixel
  - ➡ **Si-Strip detector:** ~23m³, 210m² of Si area, 10.7M channels



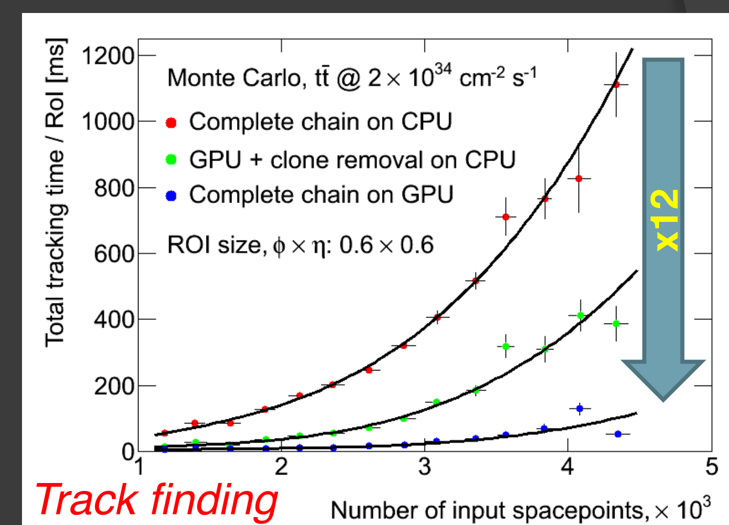
Backup Slide

Level-2 GPU Tracking Prototype

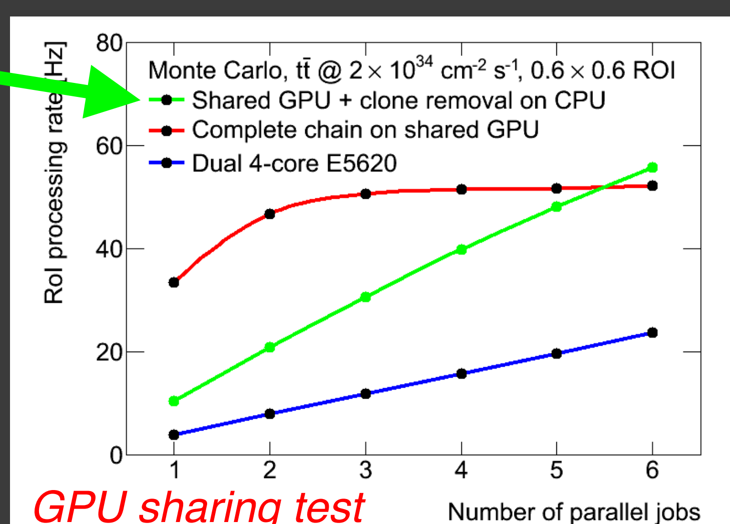
Summary of the results

- GPU-based code vs. 32-bit Athena (17.1.0)

RoI type	Data prep. speed-up
Tau 0.6x0.6	9
B-phys, 1.5x1.5	12
FullScan	26



sequential part on CPU



- x12 speed-up was obtained for the full LVL2 ID tracking chain on large RoIs
- “Client-server” architecture for GPU sharing seems to be feasible

06/06/2014 ATLAS Software & Computing Week @ CERN 7/14