

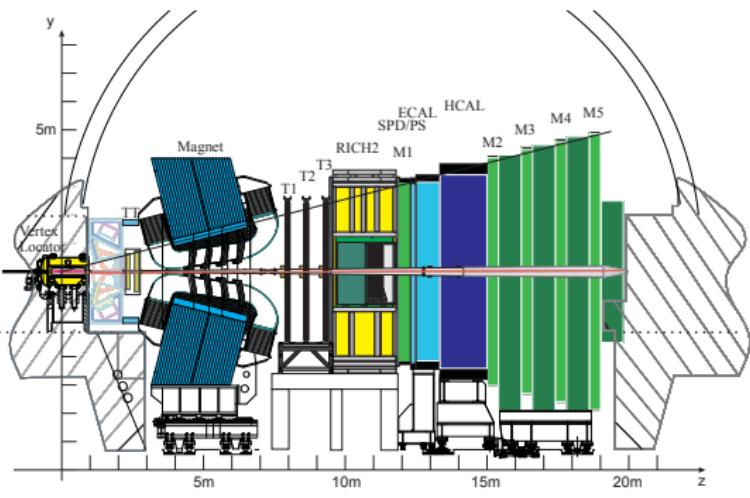
Tracking, Vertexing and data handling strategy for the LHCb upgrade

Paul Seyfert
on behalf of the LHCb collaboration

CERN

VERTEX 2017





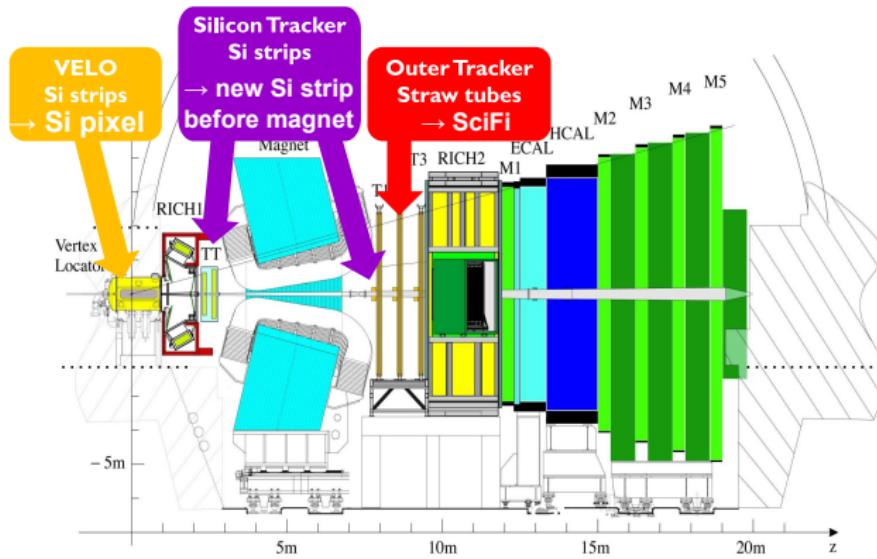
- Fully equipped forward detector at the LHC
- Approaching 400 papers
- exceeding our own expectations:
 - online calibration and alignment
 - exceeding design pile-up

Type	Observable	Current precision	LHCb 2018 (8 fb ⁻¹)	Upgrade (50 fb ⁻¹)	Theory uncertainty
B_s^0 mixing	$2\beta_s(B_s^0 \rightarrow J/\psi \phi)$	0.10	0.025	0.008	~ 0.003
	$2\beta_s(B_s^0 \rightarrow J/\psi f_0(980))$	0.17	0.045	0.014	~ 0.01
Higgs penguins	$\mathcal{B}(B_s^0 \rightarrow \mu^+ \mu^-)$	1.5×10^{-9}	0.5×10^{-9}	0.15×10^{-9}	0.3×10^{-9}
Gluonic penguins	$2\beta_s^{\text{eff}}(B_s^0 \rightarrow \phi \phi)$	–	0.17	0.03	0.02
Unitarity triangle angles	$\gamma(B \rightarrow D^{(*)} K^{(*)})$	$\sim 10\text{--}12^\circ$	4°	0.9°	negligible
	$\gamma(B_s^0 \rightarrow D_s K)$	–	11°	2.0°	negligible
	$\beta(B^0 \rightarrow J/\psi K_S^0)$	0.8°	0.6°	0.2°	negligible

Eur. Phys. Journal C (2013) 73:2373

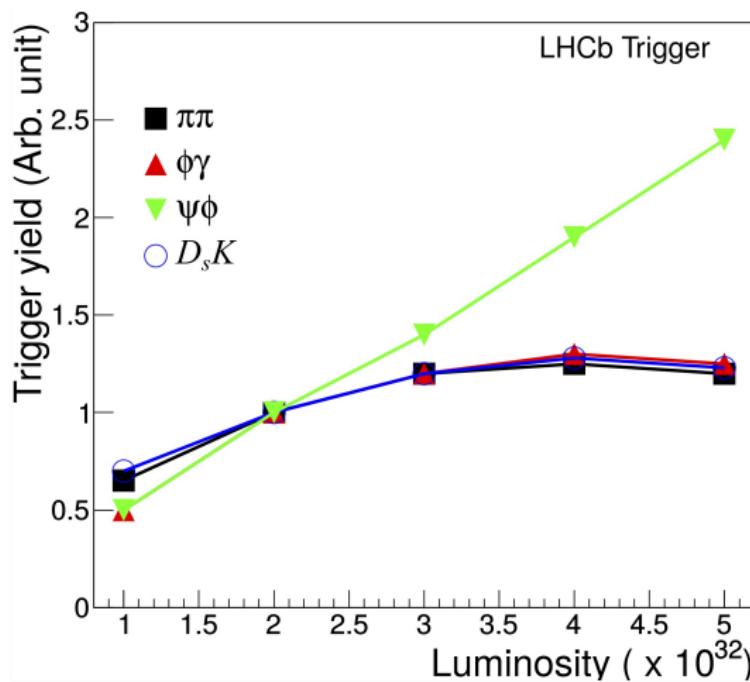
- By 2018 important analyses will still be statistically limited
- Theoretical uncertainty smaller than experimental
 - Significantly more statistics needed
 - ⇒ Go to higher luminosity

Upgrade of the tracking system



- Vertex pixel detector
see talk by Edgar Lemos Cid
- silicon strip detector
see talk by Marco Petruzzo
- scintilating fiber tracker

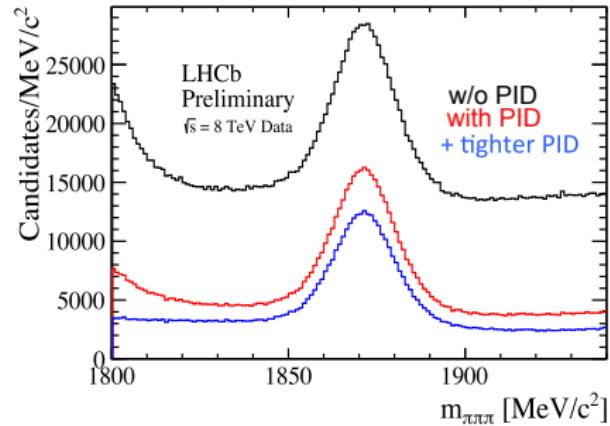
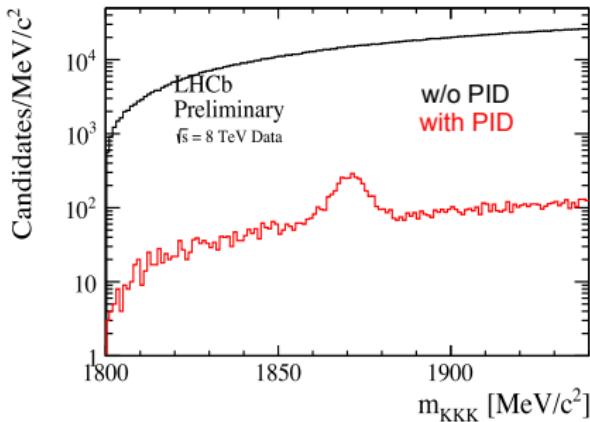
removal of hardware trigger I



what doesn't work

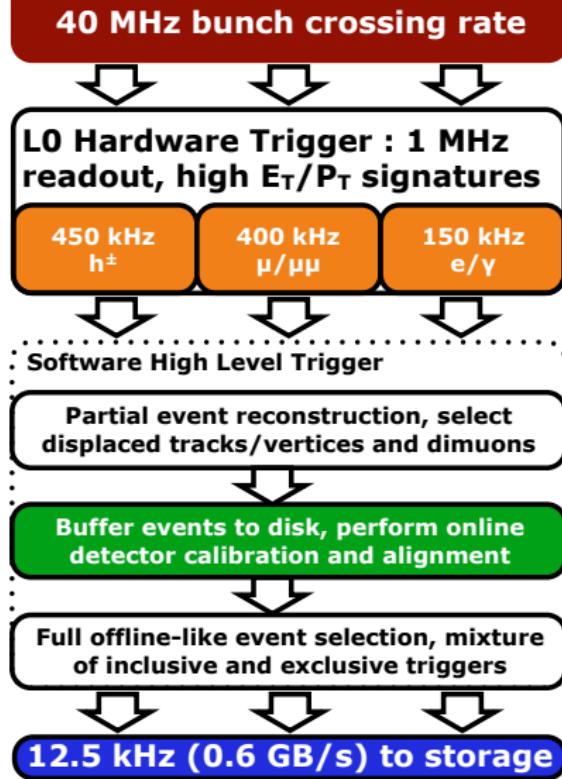
- increased luminosity
- events passing hardware trigger
- saturating bandwidth
- tighten thresholds
- loss in efficiency
- ⇒ no increase in statistics for analyses
(depending on the decay channel)

removal of hardware trigger II

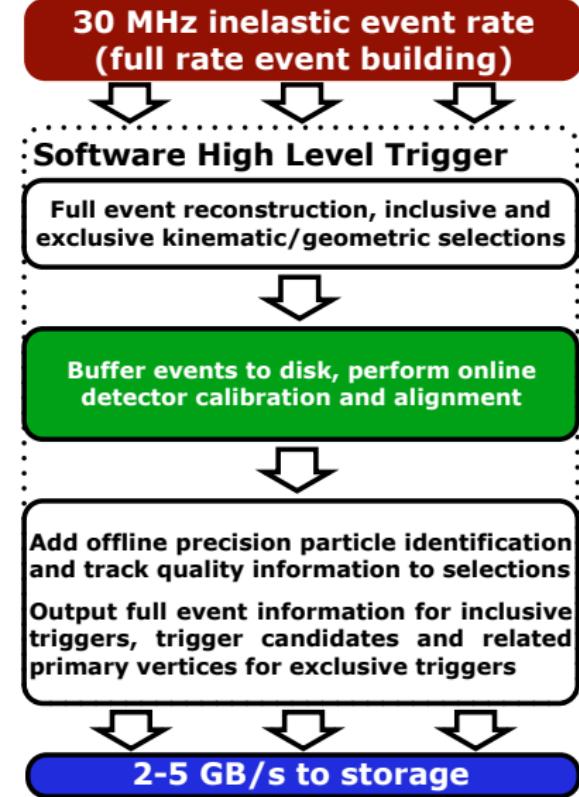


- backgrounds from real physics events
- cannot distinguish signal from background w/o RICH PID
- ⇒ even selection in software

LHCb 2015 Trigger Diagram



LHCb Upgrade Trigger Diagram



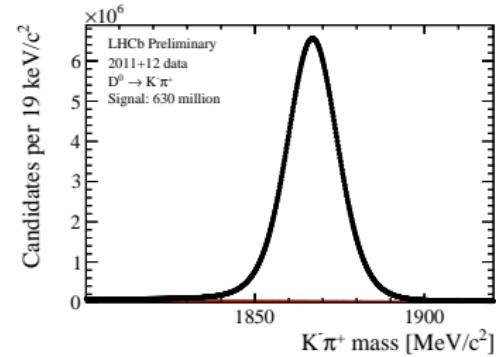
Luxury problem: MHz signals



**Triggers
today**



**Real-time data
analysis tomorrow**



5

- Selecting and storing full events could work for rare signal
- When dealing with “millions” of good signal events, rejecting background isn’t enough to stay within processing bandwidths

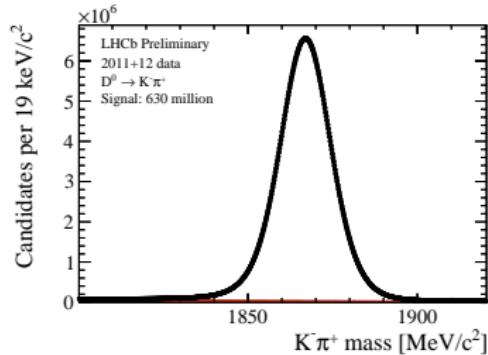
Luxury problem: MHz signals



**Triggers
today**



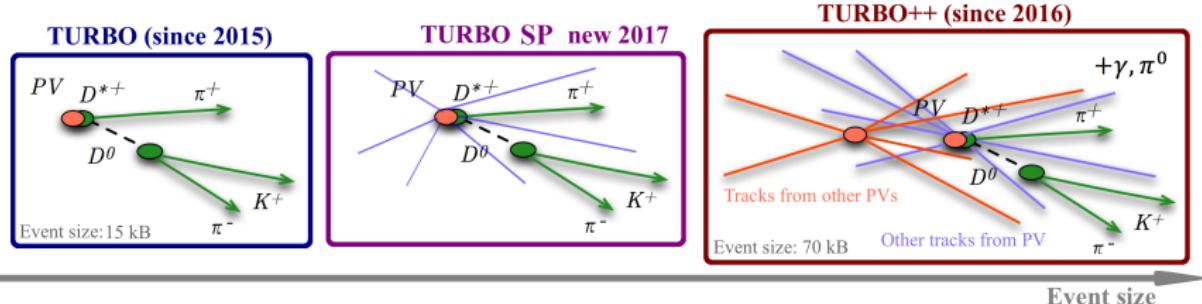
**Real-time data
analysis tomorrow**



5

The TURBO approach

- once a decay is reconstructed (mass, decay time, Dalitz plot)
no need to access raw data for analysts
- once a decay is reconstructed in the trigger
no need to re-reconstruct offline
- (unaffordable to study raw data for millions of events anyway)



10.1016/j.cpc.2016.07.022

per trigger line storage definition

- only decay and nothing else
- decay and selected reconstructed objects
- all *reconstructed* objects (no raw data)
- full raw event

TURBO triggers must be a default for many analyses

Bandwidth division I

- There's always an efficiency vs. event rate tradeoff
- assume: every analysis could max out the full data bandwidth to maximise their *efficiency*
- compromises need to be made
- ideally with little *sensitivity* loss

- Genetic algorithm approach

- Minimise the χ^2 by varying the MVA response for each decay

- w_i channel weight (= 1.0 here)

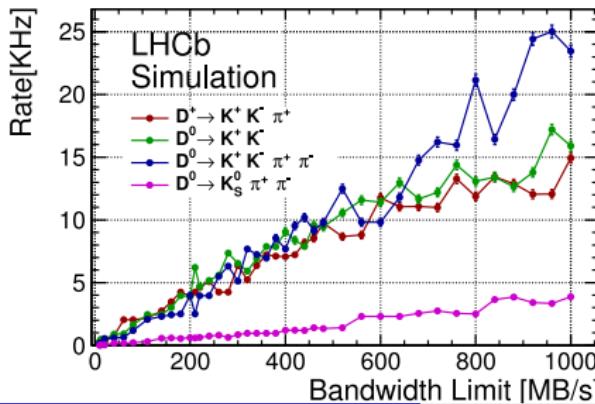
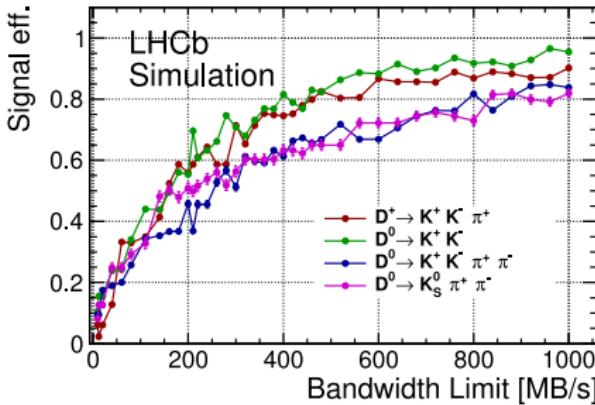
- ε_i channel efficiency

- ε_i^{\max} maximum channel efficiency
when given the full output BW

$$\chi^2 = \sum_i^{\text{channels}} w_i \times \left(1 - \frac{\varepsilon_i}{\varepsilon_i^{\max}}\right)^2$$

- if sum of all channels exceeds total bandwidth
→ assume random dropping of events

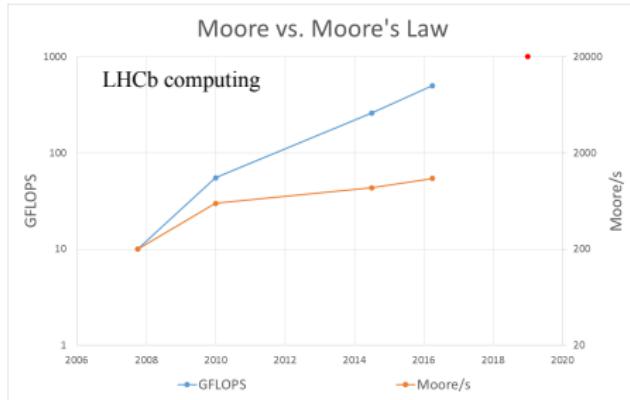
Bandwidth division II



going from maximal bandwidth to restricted bandwidth

- only small efficiency decrease
- “90 % of the data holds 95 % of the statistical power”

“Moore doesn’t obey Moore’s law”

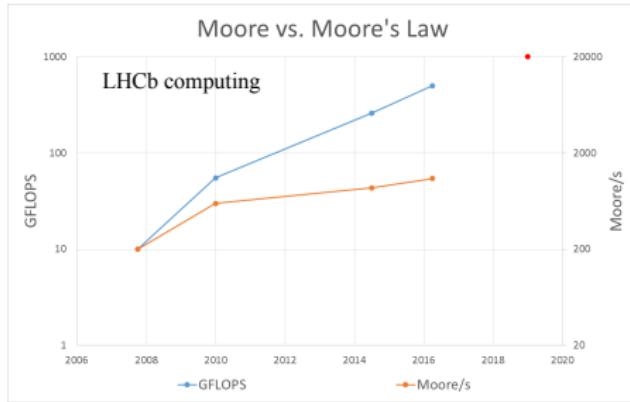


- theoretical computing power of CPUs increases (per second, per Watt, per CHF)
- observed computed trigger decisions does not follow that increase

reasons from a CPU's point of view I/II

- modern vector units process 2, 4, or 8 inputs at a time
 - ~~ our software often didn't use these
 - 7/8 of the silicon unused!

"Moore doesn't obey Moore's law"

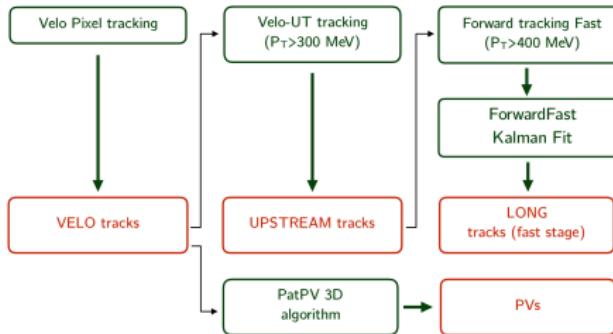


- theoretical computing power of CPUs increases (per second, per Watt, per CHF)
- observed computed trigger decisions does not follow that increase

reasons from a CPU's point of view II/II

- software not parallelised (just start multiple processes on a multicore machine)
 - ~~ processes compete for memory
 - ~~ even multiple instances of the same data (detector geometry)
 - CPU waits for data instead of computing

tracking sequence



full sequence 6.0 ms/evt

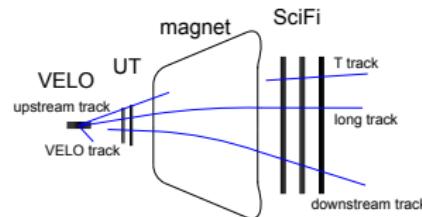
VELO tracking 2.0 ms/evt

VELO-UT tracking 0.5 ms/evt

forward tracking 2.3 ms/evt

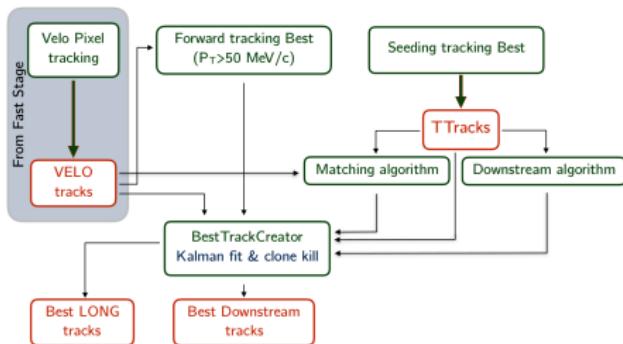
PV finding 1.1 ms/evt

(present HLT1: 35 ms)

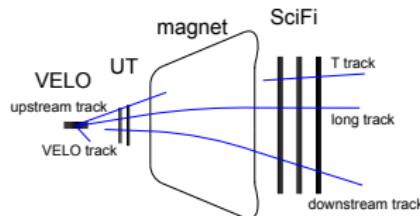


- similar to current software trigger
- single track and two track selections for displaced objects (“easy” combinatorics, limited reconstruction)

tracking sequence



(present HLT2: 650 ms)



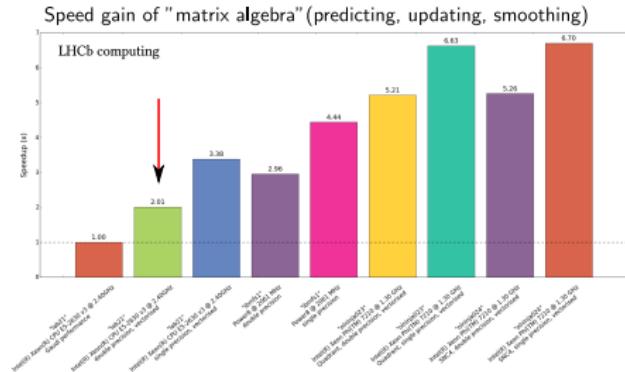
- similar to current software trigger
- single track and two track selections for displaced objects (“easy” combinatorics, limited reconstruction)
- reconstruct remaining tracks in the “full stage”
- also reconstruct decay products of strange decays outside the VELO

Kalman filter track fit

- track fit one of the big CPU time consumers
- written for sequential adding of hits
- but different tracks can be fitted independent of each other (thread parallelisable)
- matrix operations are always the same (vectorisable)



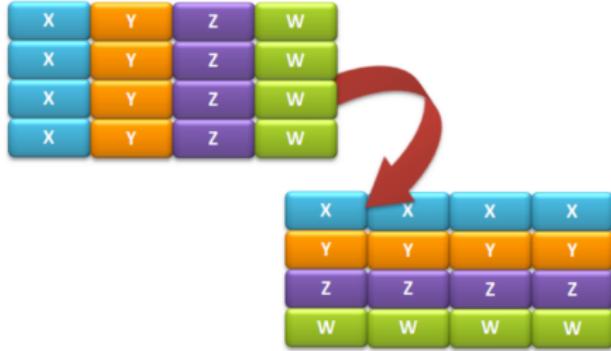
LHCb-TALK-2016-372



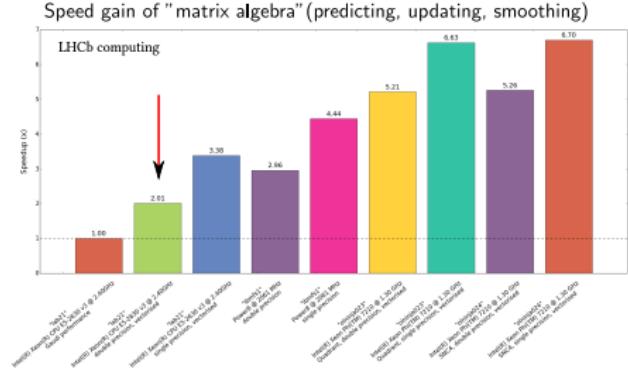
Kalman filter track fit

grain of salt

- only speeds up the matrix algebra
 - material lookup remains
 - now requires back-and-forth conversion of memory layout
- ⇒ to be consequent need to adapt underlying event model

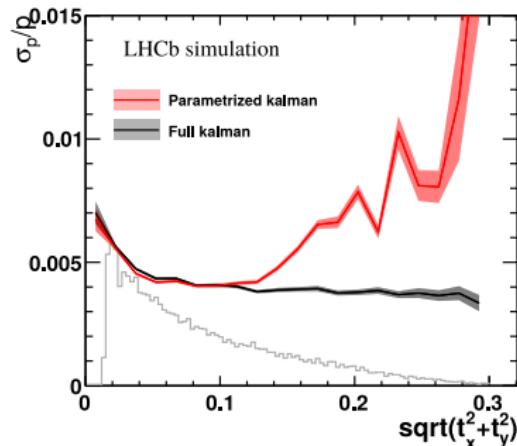
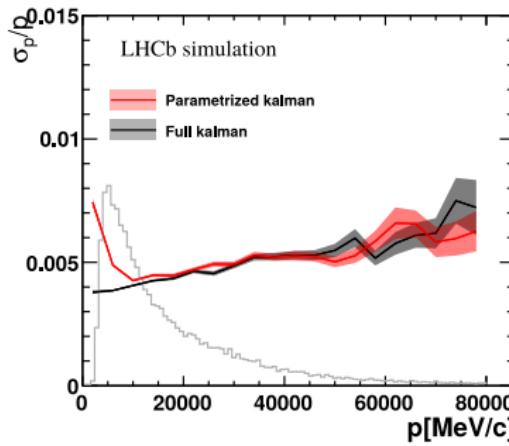


LHCb-TALK-2016-372



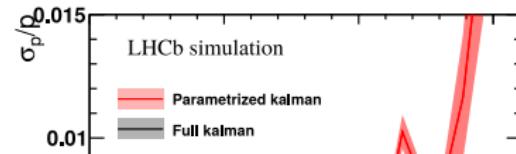
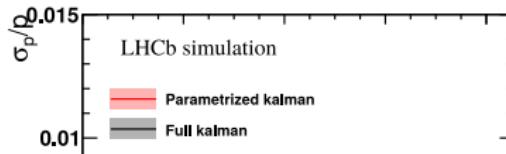
parametrised Kalman fit

- avoid first-principles math for every track
 - ~~ parametrisations can be equally accurate
 - reduce complicated B field propagation and material lookup to $\mathcal{O}(20)$ parameters



parametrised Kalman fit

- avoid first-principles math for every track
 - ~~ parametrisations can be equally accurate
 - reduce complicated B field propagation and material lookup to $\mathcal{O}(20)$ parameters



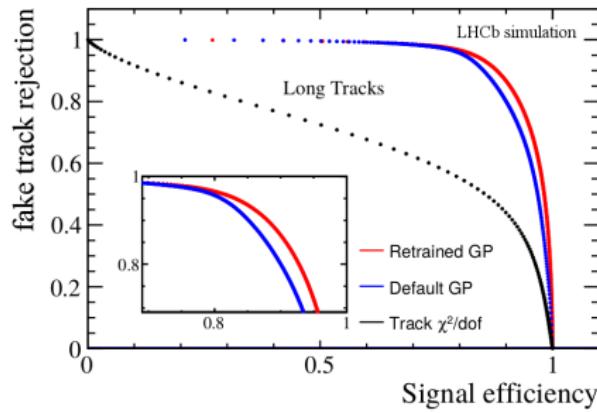
work in progress

- resolution close to reference
- potentially use full fit for tracks with large $\sqrt{t_x^2 + t_y^2}$
- find alternative parametrisations

fake track identification

- fake tracks a big contribution to computing budget in run I
- identification of fakes w/ neural network after track fit more powerful than track fit χ^2 alone

upgrade fake rejection:



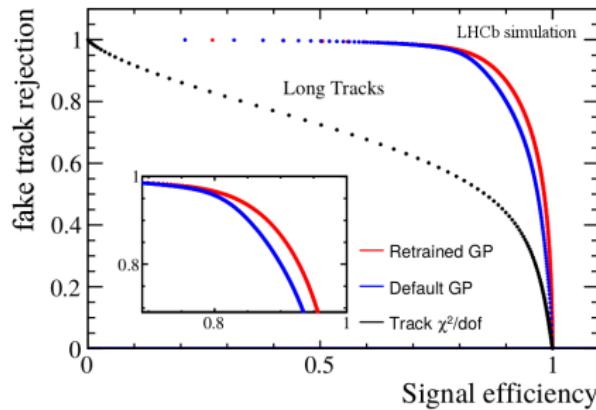
impact on run II

- RICH PID $- \mathcal{O}(20\%)$ CPU
- combinatorics $- \mathcal{O}(60\%)$ CPU
- trigger $- \mathcal{O}(30\%)$ rate

fake track identification

- fake tracks a big contribution to computing budget in run I
- identification of fakes w/ neural network after track fit more powerful than track fit χ^2 alone
- As more and more ML goes into earlier stages of the track reconstruction, there are less fakes to remove after the track fit
→ looking forward for this to become less important

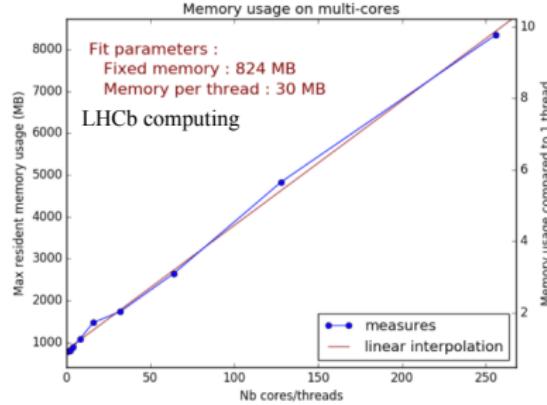
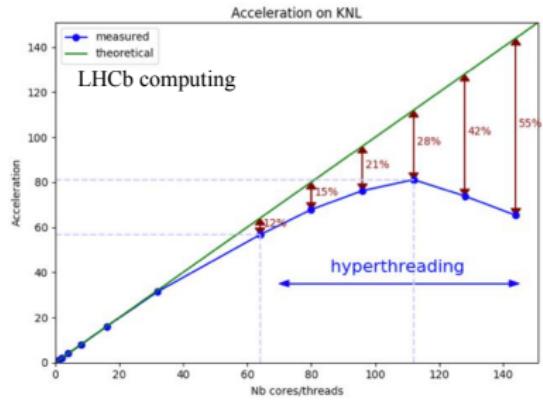
upgrade fake rejection:



impact on run II

- RICH PID $- \mathcal{O}(20\%)$ CPU
- combinatorics $- \mathcal{O}(60\%)$ CPU
- trigger $- \mathcal{O}(30\%)$ rate

multi threaded processing framework



- introduce harder framework constrains
(functional programming)
- observe near optimal speedup when increasing number of threads
- observe little memory increase when increasing number of threads

Conclusion

- LHCb physics program relies on software trigger at 30 *MHz*
- Fast tracking *without performance loss* crucial for LHCb upgrade
- Needs reconstruction software close to computer hardware to optimally use