

The LHCb Software and Computing Upgrade Towards LHC Run 3

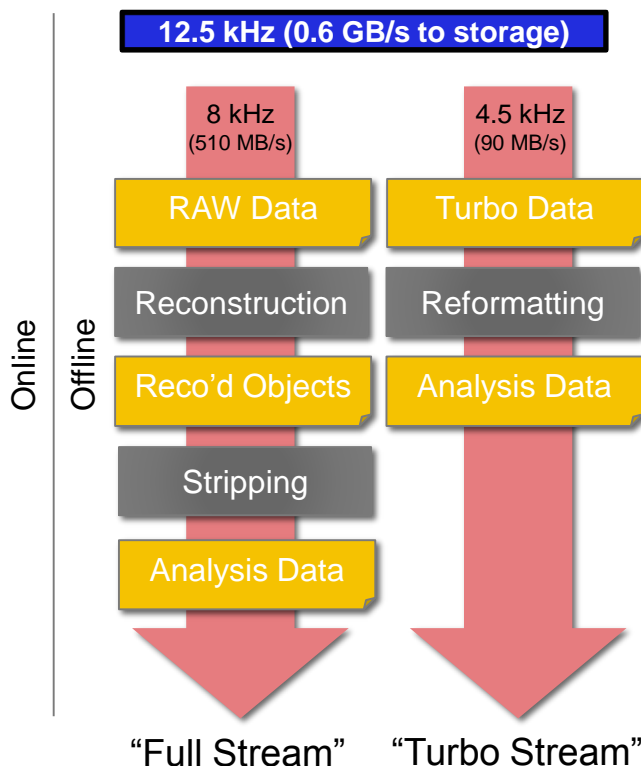
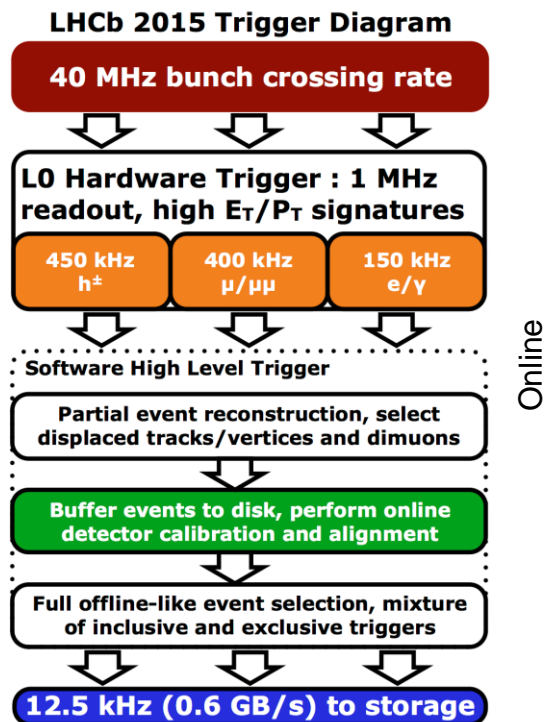
Concezio Bozzi^{1,2} and Stefan Roiser¹
On behalf of the LHCb Collaboration

¹ CERN, ² INFN Ferrara

ACAT 2017, Seattle, 22 August 2017



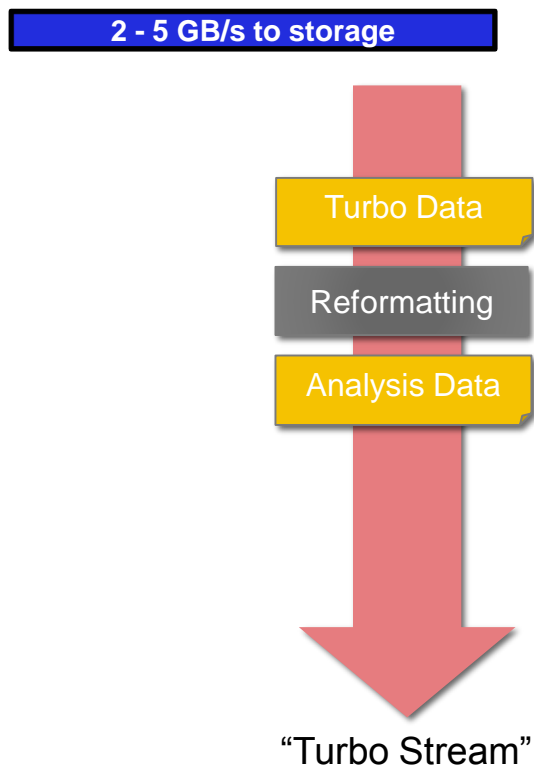
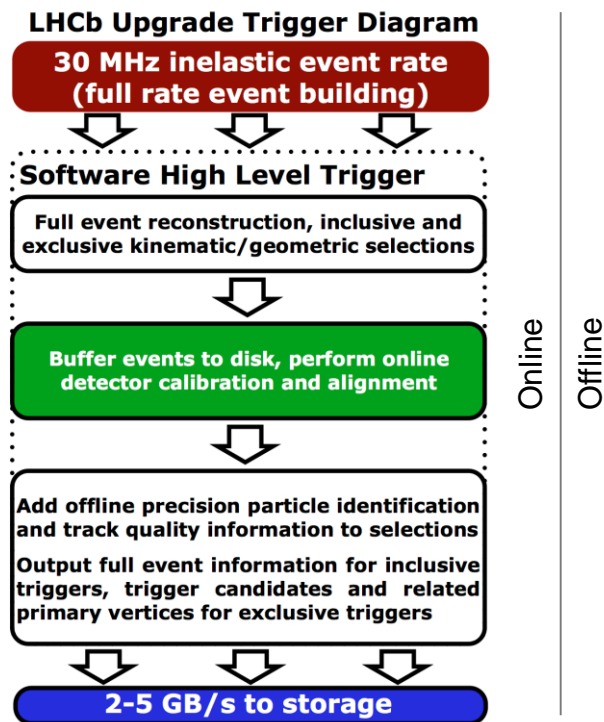
Run 2 Current Data Processing Model



- Final detector calibration and alignment done between HLT steps
 - Better signal selection in the trigger
 - No re-processing of data needed offline
- New concept of “Turbo”
 - Full event reconstruction in the trigger identical to offline
 - Turbo has smaller event size
 - Every event is used for physics
 - Offline reformatting goes fast

Data Processing Model Towards Run 3

Monday, Rosen Matev, The upgrade of the LHCb trigger for Run 3

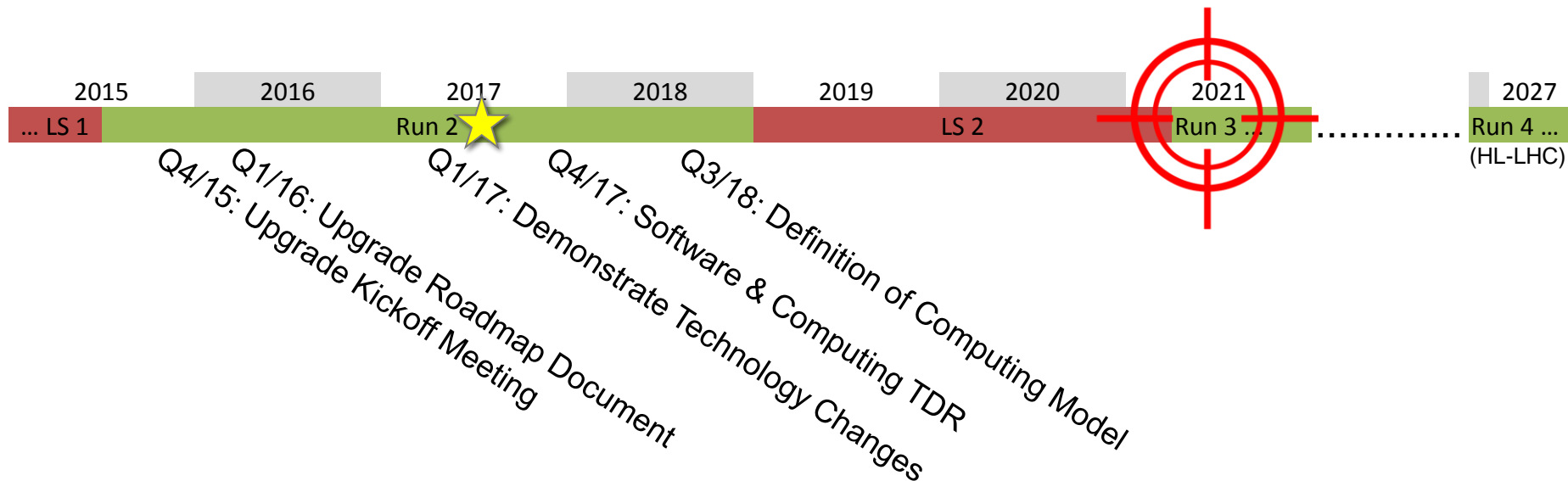


- Remove hardware trigger
 - Only selection done via software trigger farm
 - More flexibility possible
- Further extension of "Turbo Stream" concept
 - Vast majority of data after HLT will be Turbo
 - Higher rate of only signal events, almost no offline data processing needed
- Smaller event sizes with more events

Main Software and Computing Challenges

- Removal of the hardware trigger produces strain on the software framework
 - → Need to improve the performance of the software framework drastically
- Number of signal events will increase dramatically, due factor 5 increase in luminosity, smaller event sizes and no further offline selection
 - → Simulation work explodes
 - Scales with the number of signal events
 - → User analysis needs to scale with the number of signal events
 - New ways of processing data on distributed computing infrastructure
 - → Distributed Computing infrastructure needs to scale
- How to re-organize the computing model to cope with the data increase

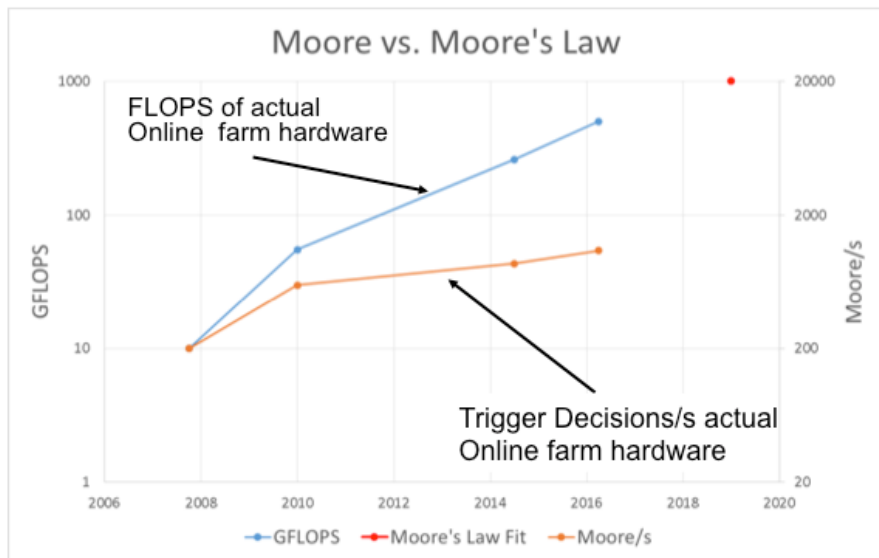
Roadmap towards Software and Computing for Run 3



- So far concentrated on changes in the framework
 - Technical feasibility demonstrated end of Q1/17
- Very tight schedule for Software and Computing “technical design report (TDR)” by end of 2017

Framework

- Current trigger is not exploiting available hardware capabilities

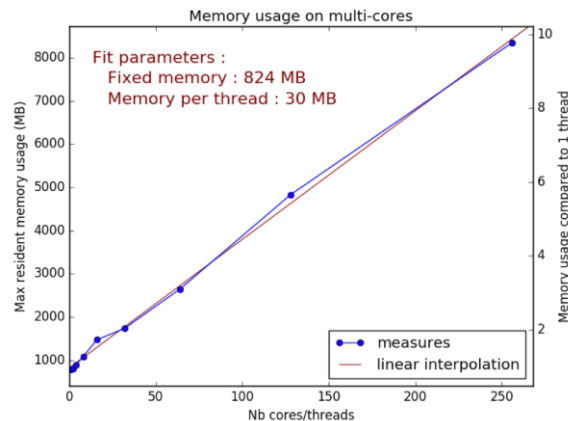
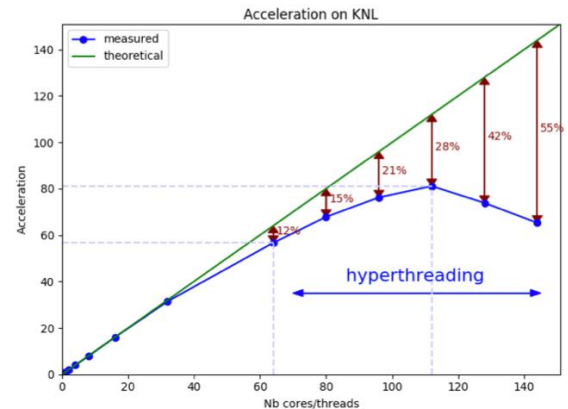
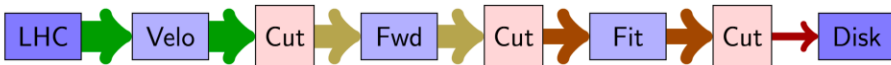


© Rainer Schwemmer

- Gaudi, the LHCb software framework was developed 15 years ago as single-threaded, sequential event processing framework
- Re-design Gaudi along
 - Introducing task parallelism
 - Data becomes immutable
 - Allow better exploitation of SIMD instructions
 - Reduce cache misses

Framework Status

- New framework is functional
 - ~ 100 algorithms have been ported including a full HLT1 sequence
 - Physics output has been validated
- Initial very good scaling of framework
 - Currently looking into lock contentions b/c of too many small memory allocations
- Major memory gains with multi-threaded framework
- Next step: Introduce filtering between reconstruction steps



More Changes

- Event Model
 - Read-only, composable, no inheritance
 - Allow different representations in AoS/SoA
 - Move to single precision whenever possible
- Detector description and conditions database need to be re-designed
 - Packages not threadsafe and partly not maintained anymore
 - E.g. use DD4HEP for geometry description
- Use of alternative architectures
 - Porting stack to ARM, Openpower

Thursday, Niko Neufeld, Cross-architecture Kalman
filter benchmarks on modern hardware platforms

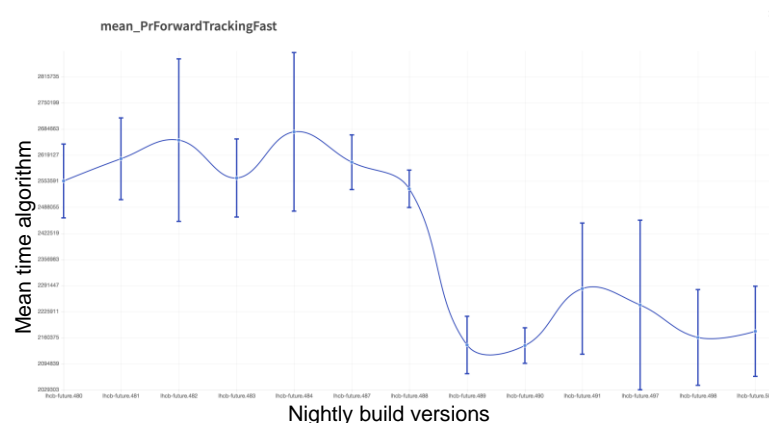
Code Optimisations

		SSE4		AVX		AVX2	
double	SISD	233.462		228.793		228.752	
	SIMD	122.259	(1.90)	58.277	(3.93)	58.243	(3.93)
float	SISD	214.451		209.745		209.756	
	SIMD	55.707	(3.85)	26.582	(7.89)	26.539	(7.90)

Timing and (speedup) of SIMD instructions © Chris R Jones

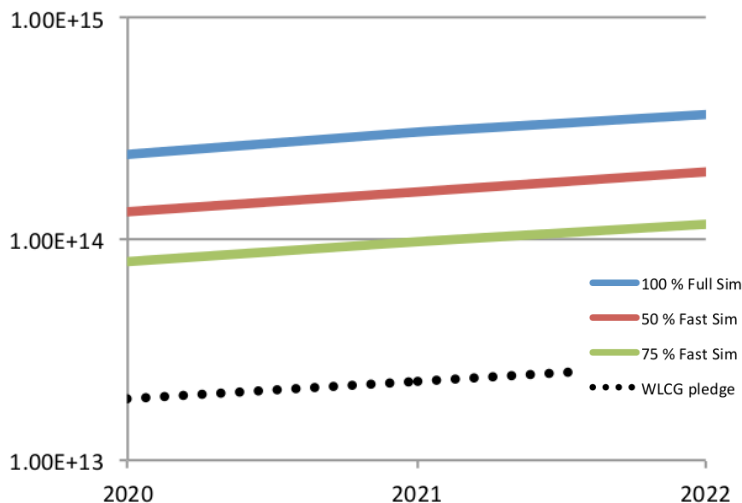
- Example: Vectorization of Rich Photon reconstruction shows perfect speedup using VC library
 - NB: AVX2 behavior to be understood

- Metrics extracted from “nightly builds”, show evolution of the software
 - Timing, cache misses, SIMD instructions, branch miss predictions, ...



Simulation

- WLCG forecasted pledges will not fit needed simulation work
 - Distributed computing will be almost exclusively used for simulation work
 - Aiming to simulate 100 % of signal (currently ~ 12 %)



22 Aug '17

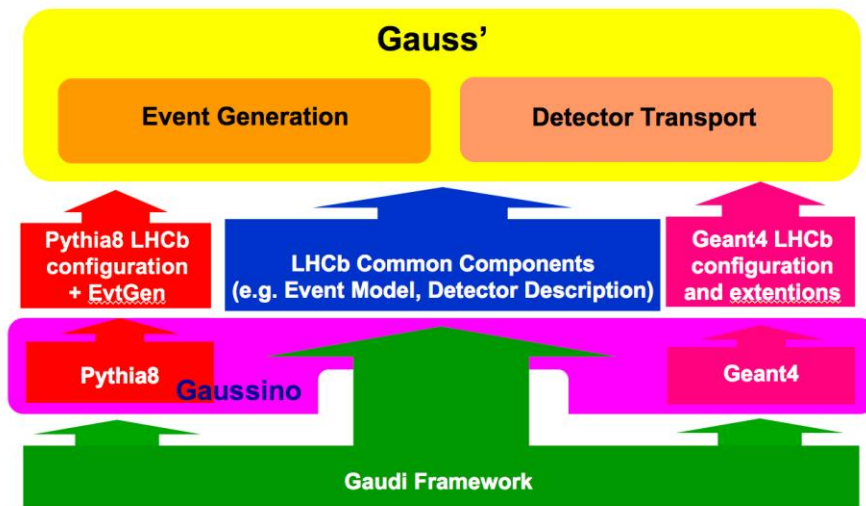
Stefan Roiser - ACAT

- Develop a palette of different “fast simulations” options:
 - Re-decay signal N times, re-using same underlying event
 - Fully simulate only part of the detector, e.g. tracker only
 - Signal ParticleGun
 - Calo-shower libraries
 - Delphes, fully parametric simulation



Simulation – Status and Outlook

- How to integrate Geant4 multi-threaded into the new framework
 - Geant4 MT and Gaudi use different concurrency models
 - Geant4 MT thread based \leftrightarrow Gaudi task based
 - Geant4 MT event level \leftrightarrow Gaudi event and algorithm level
- Integrate Gaussino into the LHCb software stack
 - Gaussino is an experiment agnostic version of Gauss
 - Developed together with FCC and CERN/SFT
- Keep compatibility with GeantV
 - R&D for vectorized simulation



More Areas of Development

- Distributed (Offline) Computing Framework
 - Evolutionary changes to allow scaling with increased data volume
 - Workflow management, data management, monitoring, ...
- Distributed Computing Analysis Model
 - Deal with increased data volumes. Sparse selection of data
 - Looking into centralized productions to increase I/O efficiency
 - Introduce multiple streaming layers to avoid sparse selection
- Computing Model
 - How many event data replicas can be made available?
 - Models for data processing and event simulation on distributed computing resources.
 - Working towards a plan by mid 2018.

Involving the Whole Experiment



Hackathon



Advanced C++ course

- Computing Project is working on the engineering of the “core aspects” of the upgrade
 - Several Mio LOC code base throughout the collaboration needs to be optimized and adapted with the sub-systems
- Training is essential
 - C++ '17, Python 3, ...
- Concentration on upgrade in dedicated hackathons to get sub-system developers up to speed

Summary

- Major changes in the LHCb Run 3 event processing
 - Software only trigger at 30 MHz bunch crossing rate
 - Full exploitation of Turbo concept leading to vast increase of signal events
- Need to cope with repercussions on the software and computing infrastructure
 - Re-design of the Gaudi needed towards a multi-threaded task-parallel framework
 - Adopt simulation software to new framework and develop different fast simulation options
- Working on a technical design report, due by the end of 2017

Thank You for Your Attention !!!!

Do you have any questions?