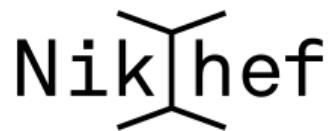


Allen and HLT1

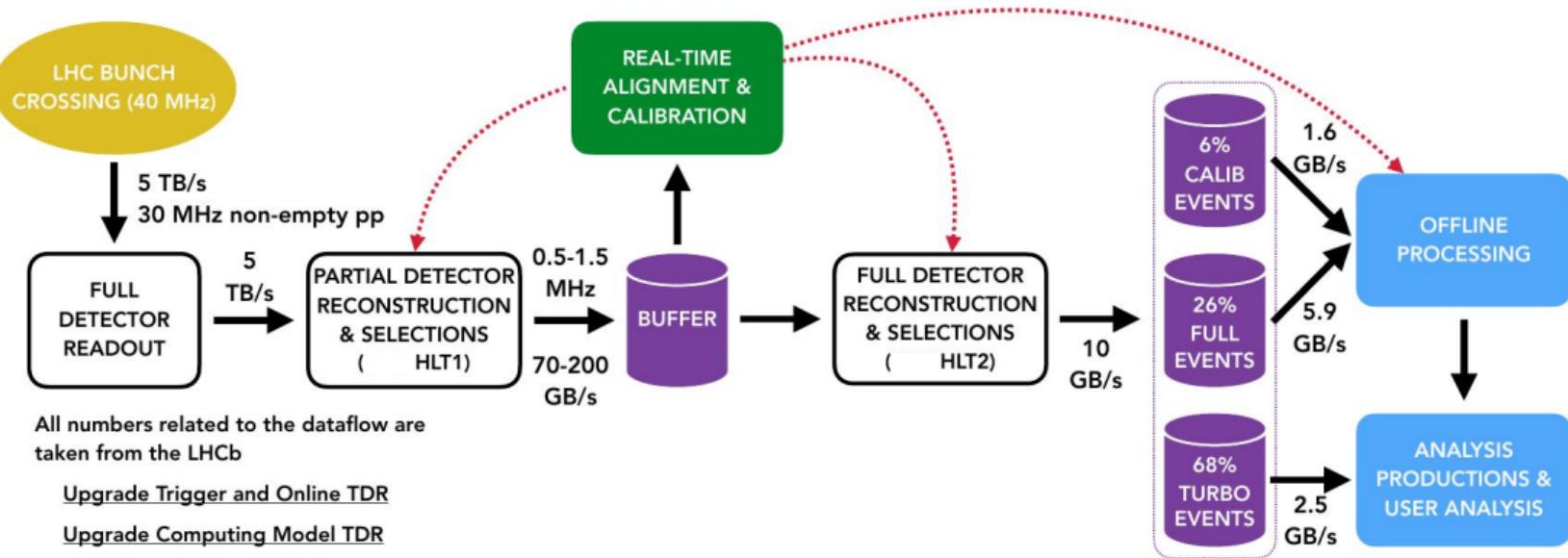
Run3 starterkit-for-all

Roel Aaij

March 18th, 2022

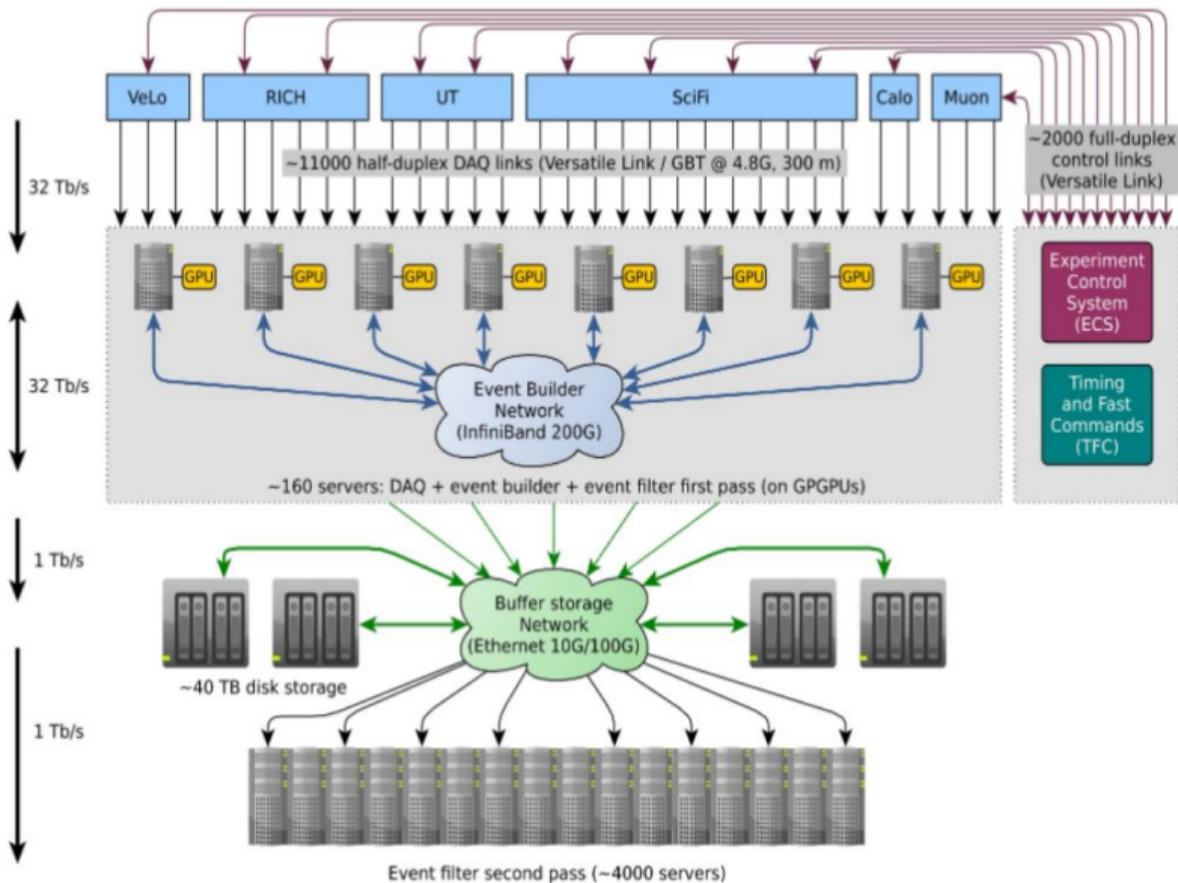


LHCb Upgrade Dataflow

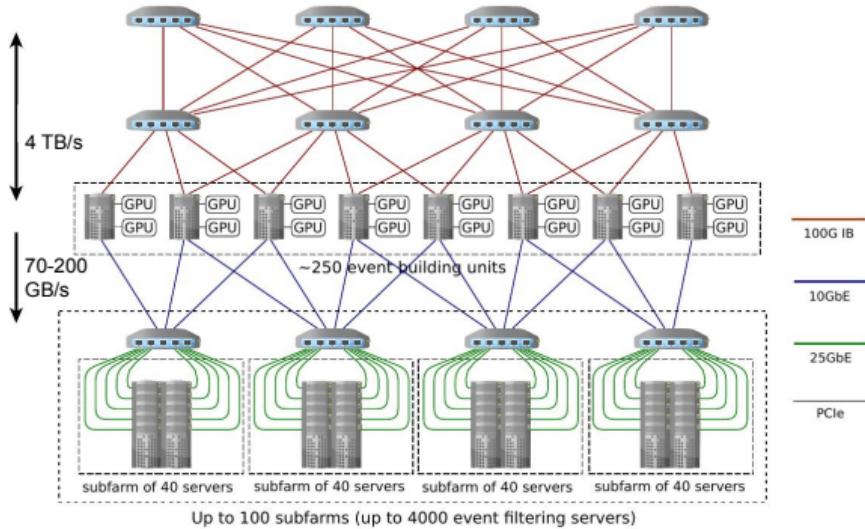


HLT1 challenge: reduce 4 TB/s to 70-200 GB/s in real-time with high physics efficiency

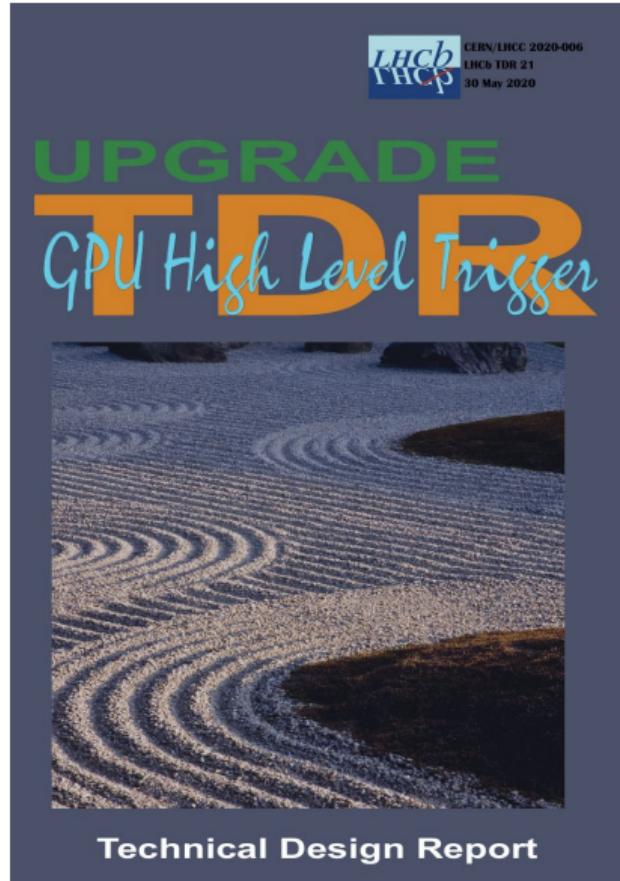
LHCb Upgrade Trigger and DAQ



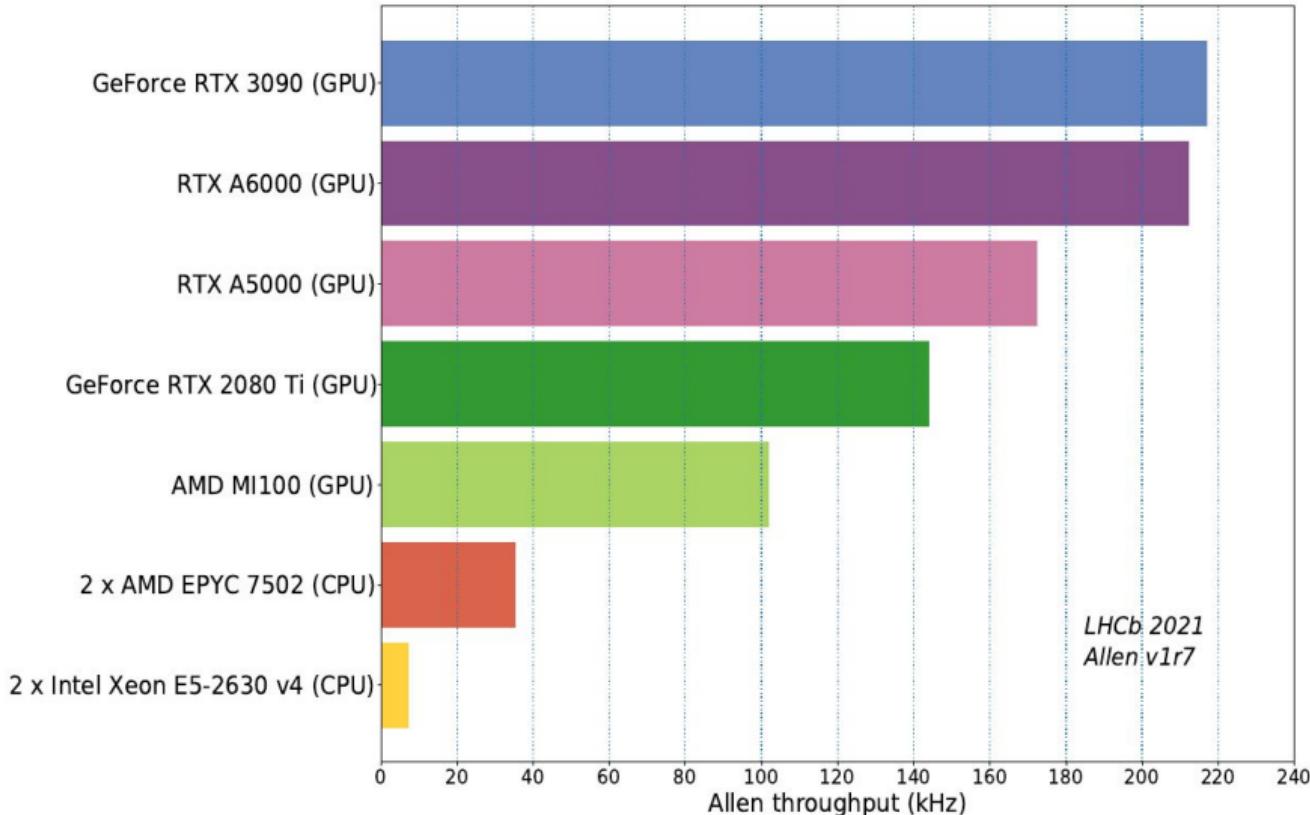
HLT1 on GPUs: Allen



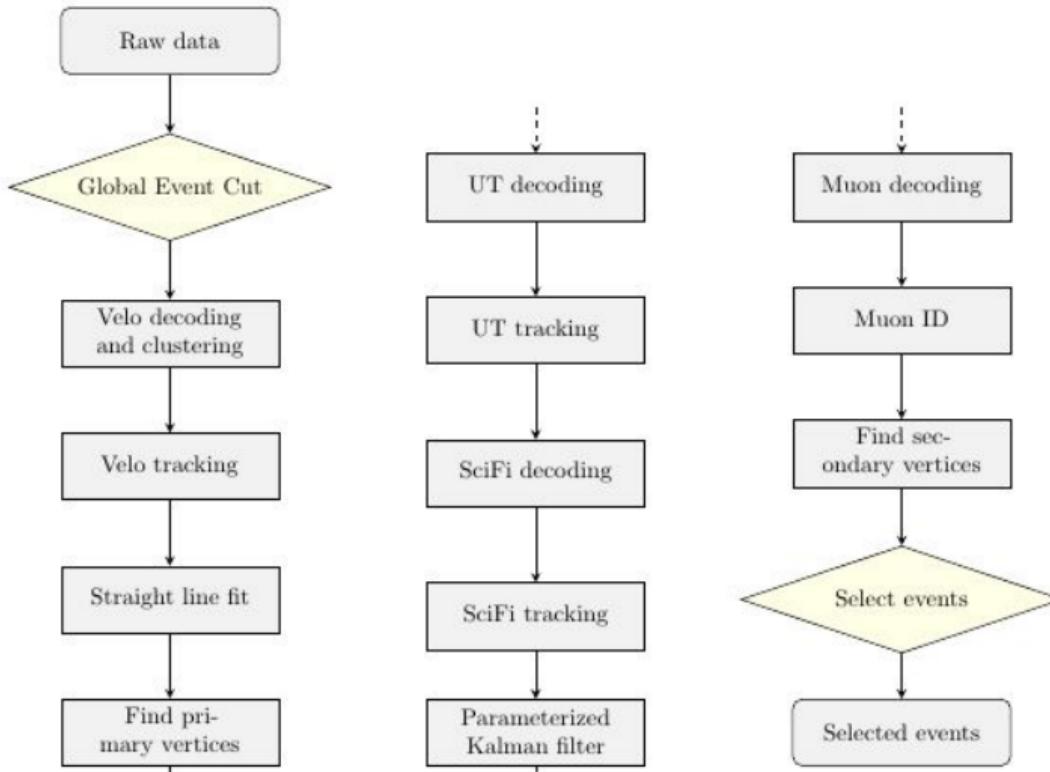
Allen implements HLT1 as a GPU application; up to 3 GPUs installed in each event builder server



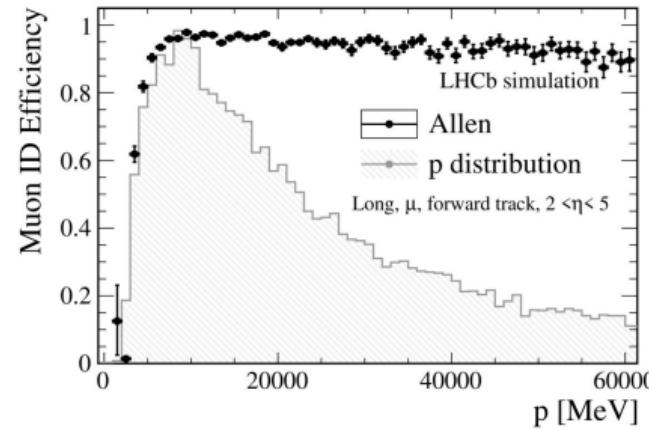
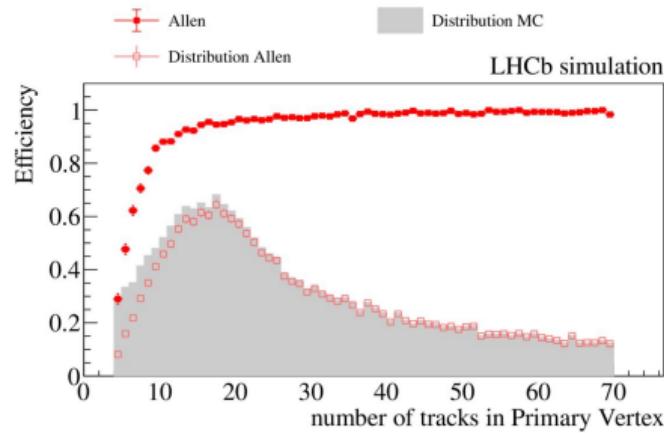
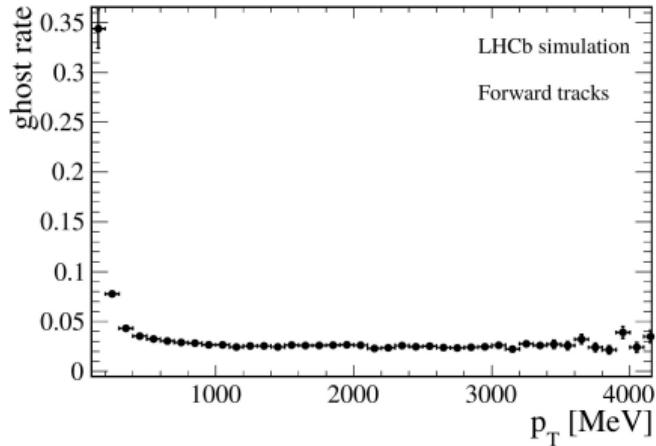
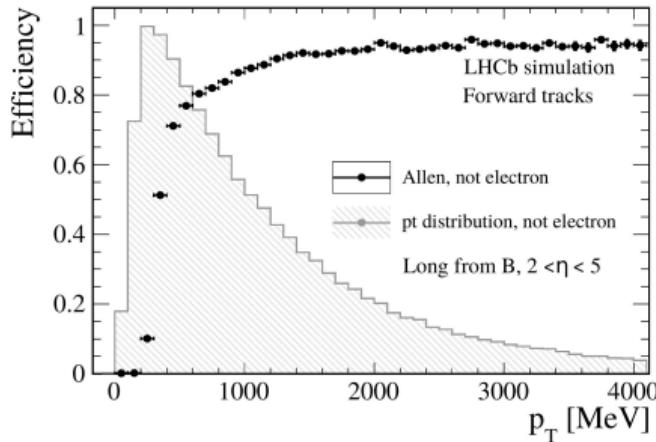
Throughput



Reconstruction Sequence

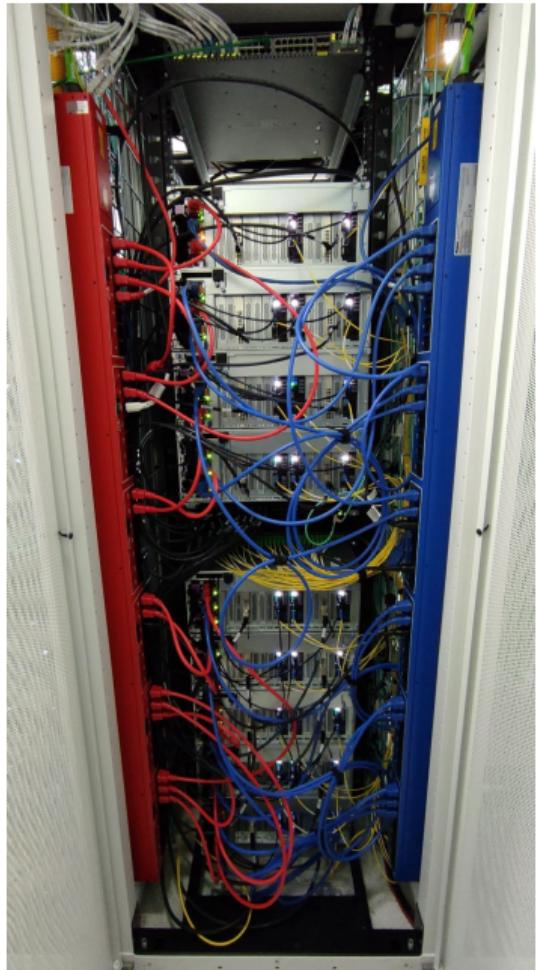


Reconstruction Performance

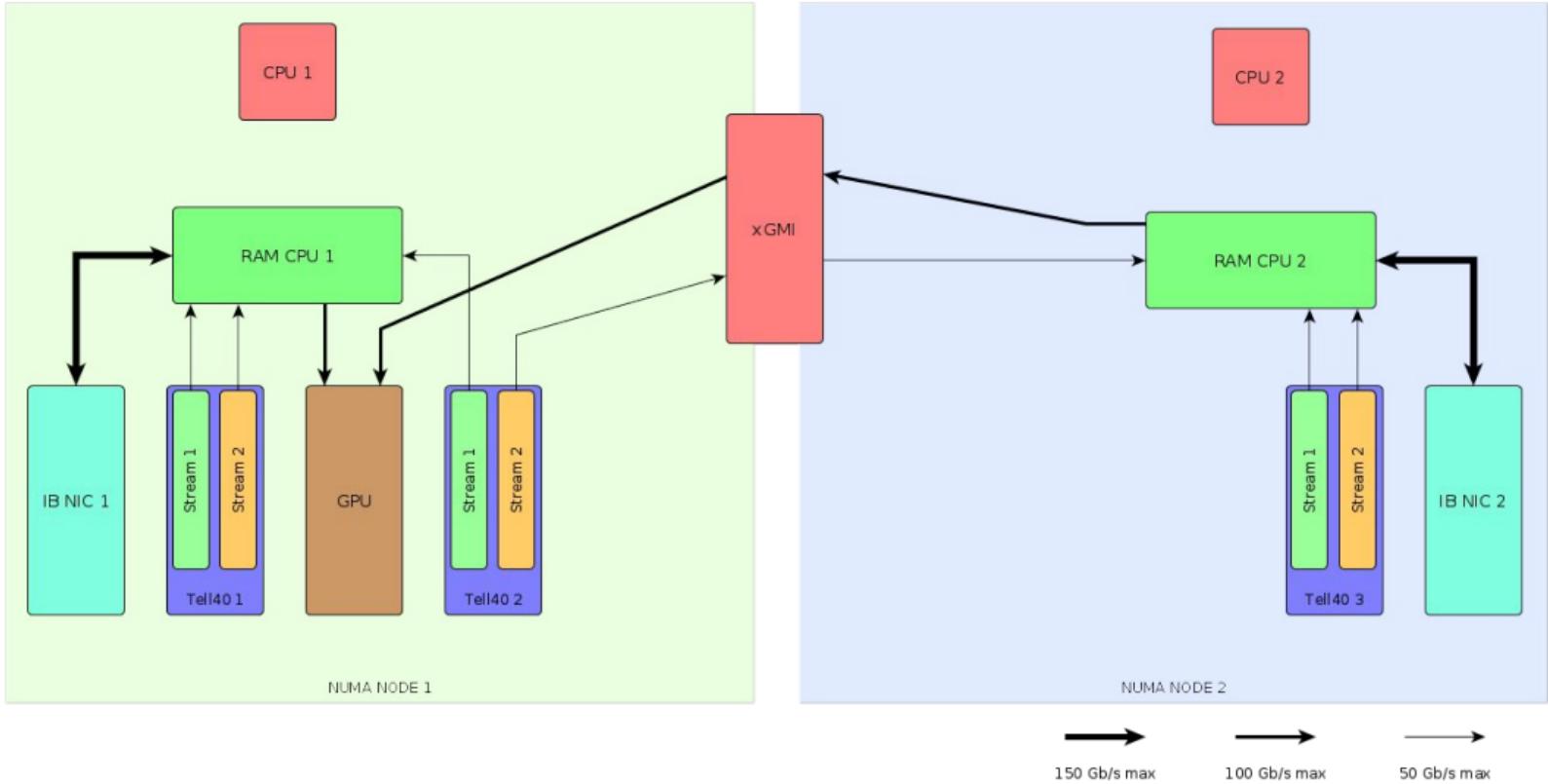


Integration

- 200 GPUs installed in EB servers
- Input data copied to GPUs in EB format:
 $O(1000)$ multi-fragment-packets in
multi-event-packets of $O(30000)$ events
- $O(24)$ GB/s per GPU
- Event data memory layout “transposed” with
respect to event-by-event
- Input data directly from shared memory
- Output in MDF format to DAQ
- Experiment Control System steers HLT1
- Obtain geometry and conditions from LHCb
software on-the-fly



Layout of an EB server



Allen

- gitlab: gitlab.cern.ch/lhcb/Allen
- C++17, CUDA, HIP
- Builds and runs for CPU and GPU (NVIDIA and AMD)
- Standalone builds and “stack” builds
- Batches of $O(1000)$ events; $O(10)$ batches in parallel
- GPUs have their own memory
- All data must be copied to and from the GPU
 - input data
 - geometry and conditions
 - output data
- Allen manages that for you
- No dynamic allocations, e.g. `v.push_back(3.14)` or `new`
- Count first, write later
- All algorithms written from scratch to parallelise better
- documentation: <https://allen-doc.docs.cern.ch/>

Demo

Hlt1KsToPiPi:	1955/ 97799, (599.70 +/-	13.43) kHz
Hlt1TrackMVA:	1383/ 97799, (424.24 +/-	11.33) kHz
Hlt1TwoTrackMVA:	2285/ 97799, (700.93 +/-	14.49) kHz
Hlt1TwoTrackKs:	209/ 97799, (64.11 +/-	4.43) kHz
Hlt1SingleHighPtMuon:	29/ 97799, (8.90 +/-	1.65) kHz
Hlt1LowPtMuon:	7163/ 97799, (2197.26 +/-	24.99) kHz
Hlt1D2KK:	376/ 97799, (115.34 +/-	5.94) kHz
Hlt1D2KPi:	494/ 97799, (151.54 +/-	6.80) kHz
Hlt1D2PiPi:	254/ 97799, (77.91 +/-	4.88) kHz
Hlt1DiMuonHighMass:	402/ 97799, (123.31 +/-	6.14) kHz
Hlt1DiMuonLowMass:	889/ 97799, (272.70 +/-	9.10) kHz
Hlt1DiMuonSoft:	15/ 97799, (4.60 +/-	1.19) kHz
Hlt1LowPtDiMuon:	1091/ 97799, (334.67 +/-	10.08) kHz
Hlt1TrackMuonMVA:	82/ 97799, (25.15 +/-	2.78) kHz
Hlt1TrackElectronMVA:	420/ 97799, (128.84 +/-	6.27) kHz
Hlt1SingleHighPtElectron:	75/ 97799, (23.01 +/-	2.66) kHz
Hlt1DisplacedDielectron:	428/ 97799, (131.29 +/-	6.33) kHz
Hlt1DisplacedLeptons:	441/ 97799, (135.28 +/-	6.43) kHz
Hlt1SingleHighEt:	2/ 97799, (0.61 +/-	0.43) kHz
Hlt1GECPassthrough:	90837/ 97799, (27864.40 +/-	24.67) kHz
Hlt1NoBeam:	0/ 97799, (0.00 +/-	0.00) kHz
Hlt1BeamOne:	0/ 97799, (0.00 +/-	0.00) kHz
Hlt1BeamTwo:	0/ 97799, (0.00 +/-	0.00) kHz
Hlt1BothBeams:	99/ 97799, (30.37 +/-	3.05) kHz
Hlt1VeloMicroBias:	90/ 97799, (27.61 +/-	2.91) kHz
Hlt10DINLumi:	0/ 97799, (0.00 +/-	0.00) kHz
Hlt10DINNoBias:	0/ 97799, (0.00 +/-	0.00) kHz
Hlt1Passthrough:	97799/ 97799, (30000.00 +/-	0.00) kHz
Hlt1RICH1Alignment:	43/ 97799, (13.19 +/-	2.01) kHz
Hlt1RICH2Alignment:	13/ 97799, (3.99 +/-	1.11) kHz
Inclusive:	97799/ 97799, (30000.00 +/-	0.00) kHz

36003.281133 events/s

Hands-on: Write your own trigger line

- Open the documentation: <https://allen-doc.docs.cern.ch/>
 - Writing Selection -> Adding a new Selection
- Let's try Z->ee
- Use the stack you already have or:
`tar -xzf /eos/lhcb/user/a/ascarabo/stack_starterkitRun3.tar.gz -C .; cd stack`
- device/selections/lines/electron
- Take inspiration from: Hlt1DisplacedDielectron (aka copy-paste)
- Name your line
- Create the .cuh in include, .cu in src
- Change/update get_input and select
- Update properties/cuts
- Update
`configuration/python/AllenConf/hlt1_electron_lines.py`
- Update `configuration/python/AllenConf/HLT1.py`
- Build
- Run

Tricks for building

- make fast/Allen/configure
- Sequences are not automatically regenerated:
`rm -f Allen/build.x86_64_v2-centos7-gcc11-opt/*.json`
- The stack builds automatically generates wrappers for Gaudi integration
These are not automatically regenerated:
- `cd Allen/build.x86_64_v2-centos7-gcc11-opt; \`
`rm code_generation/parsed_algorithms.pickle; \`
`rm code_generation/algorithm_wrappers/alg_t_gaudi.cpp`

Running

- Copy options from `/afs/cern.ch/user/r/raaij/public/run3-starterkit`
- `./Moore/run bash`
- To run on signal MC:
 - `gaudirun.py Zee_input_and_conds.py allen_in_moore.py`
- To run on minbias:
 - `gaudirun.py default_input_and_conds_hlt1.py \ allen_in_moore.py`
- Retina clusters are the default Velo clusters, but most extant MC does not contain them. `allen_in_moore.py` uses the `hlt1_pp_veloSP` sequence instead of `hlt1_pp_default`
- Solution on branches `run3_starterkit`
`HeavyDielectron.{cuh,cu}`

Bonus: Standalone Allen

- new shell on lxplus-gpu.cern.ch
- mkdir stack/Allen-build; cd stack/Allen-build
- source /cvmfs/sft.cern.ch/lcg/views/setupViews.sh \
LCG_101 x86_64-centos7-clang12-opt
- cmake -GNinja -DTARGET_DEVICE=CUDA -DCUDA_ARCH=MAX
-DSTANDALONE=ON -DUSE_ROOT=ON -DBUILD_TESTING=ON
-DSEQUENCES=all \
-DCMAKE_CUDA_COMPILER="/cvmfs/sft.cern.ch/lcg/contrib" \
"/cuda/11.4/x86_64-centos7/bin/nvcc" ../Allen
- ninja
- ./Allen --mdf /eos/lhcb/user/r/raaij/Upgrade/mdf/Zee.mdf \
--param ../Allen/input/parameters -g \
"/eos/lhcb/user/r/raaij/Allen" \
"/geometry_ddb-20201211_sim-20201218-vc-md100" \
--sequence hlt1_pp.veloSP \
--run-from-json=1 --events-per-slice=100