LHCb Computing

Computing for the LHCb Upgrade

LHCb Upgrade: goal and timescale

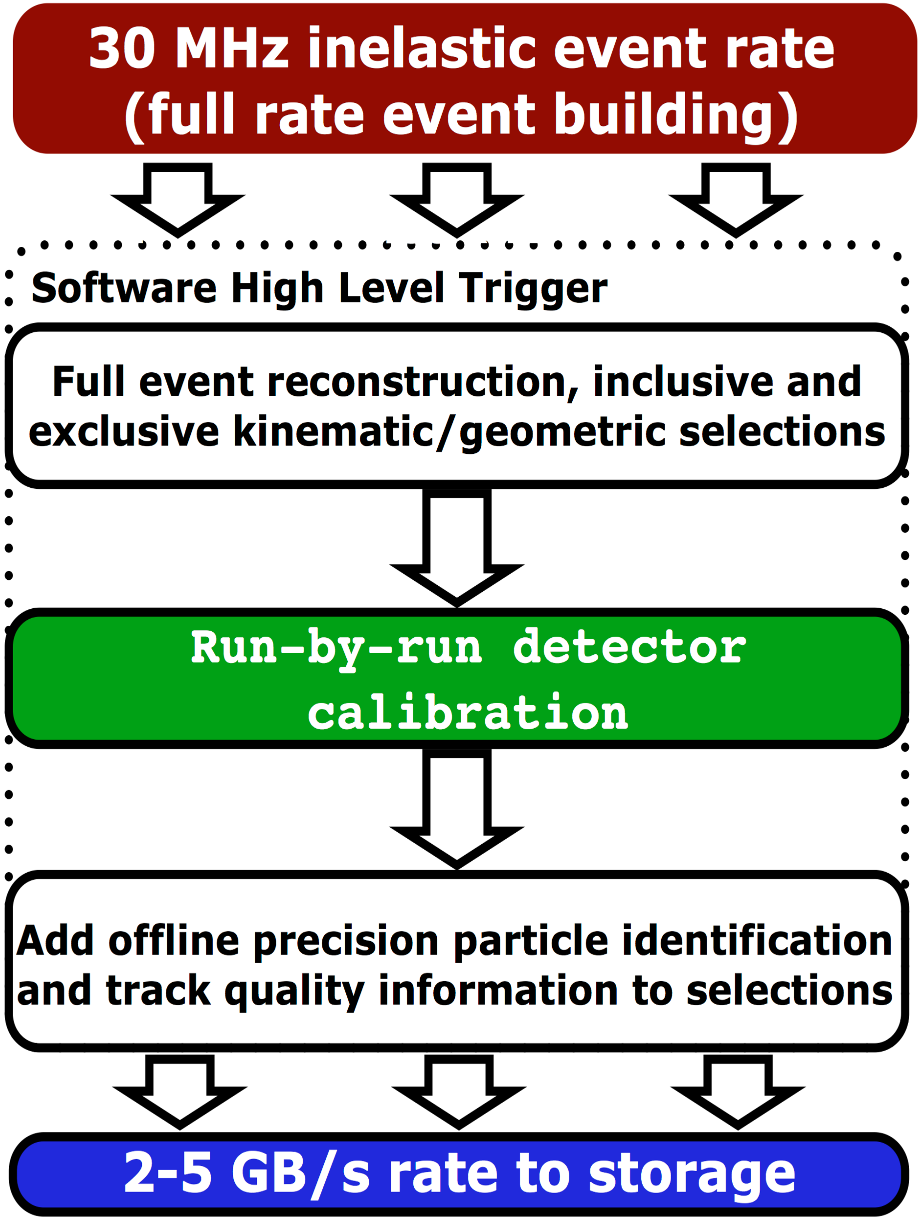
- **LHCb upgrade will be operational after LS2 (~2021)**
- Increase significantly the statistics collected by the experiment, keeping the present excellent performance
  - Raise operational luminosity to a levelled 2×10^{33} $cm^{-2}s^{-1}$
    - Necessitates redesign of several sub-detectors
    - Does not require HL-LHC
  - Full software trigger at 40 MHz bunch crossing rate
    - Allows effective operation at higher luminosity
    - Improved efficiency in hadronic modes
    - Necessitates upgrade of the DAQ and HLT
- The gain is a huge increase in precision, in many cases to the theoretical limit, and the ability to perform studies beyond the reach of the current experiment
  - Flexible trigger and unique acceptance also opens up opportunities for other topics apart from flavour

LHCb Upgrade Computing Timeline

- 2021 is tomorrow
  - No time (or effort) for major changes in technology
  - Focused R&D based on existing experience
  - Possibility to use Run 2 as a test bed for new ideas
- Roadmap document for TDR published 31st March 2016
  - Specifies R&D required for informed decisions in TDR
- All R&D reports ready Q2 2017
- Software and Computing TDR scheduled for Q4 2017
  - Baseline technology choices made
- Computing model finalized Q4 2018

- Trigger-less readout at full LHC crossing rate
  - No hardware (L0) trigger
- First and second level software trigger (HLT1/2) running on Event Filter Farm
  - Full HLT2 deferral
    - (as in Run 2)
    - Offline quality detector calibration and reconstruction
- Event Size:
  - 100 kB maximum (constraint from readout system)
    - 10-20 times smaller for channels going to Turbo stream

LHCb Upgrade Trigger Diagram



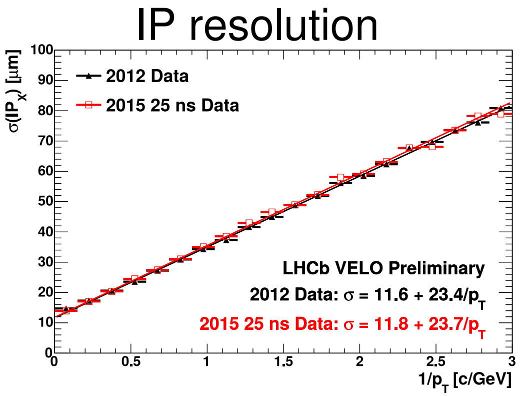
Online reconstruction

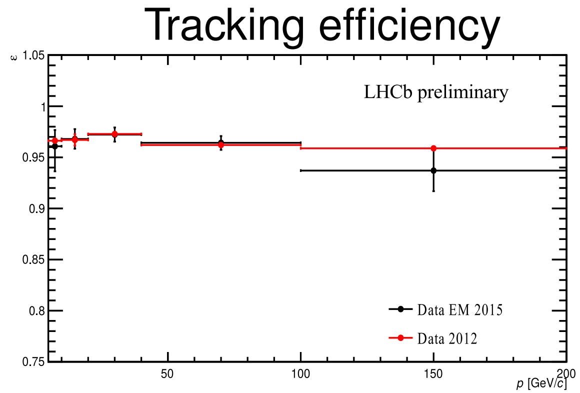
- "Offline quality" online calibration commissioned in 2015
  - Same calibration online and offline
  - Sufficient quality for offline analysis
    - No need for "end of year" reprocessing
- Opens up possibility to do full offline reconstruction online
  - Has been a goal for Run 2, limited only by CPU budget
  - Has been achieved for 2016, thanks to CPU optimisation and improvements to reconstruction algorithms
- If reconstruction is identical to offline, is there a need to run it again offline?
  - Commissioning in 2016 an online reconstruction format to be transmitted to offline, for direct analysis
- Baseline for the upgrade: all reconstruction done online in HLT farm
  - No offline reconstruction for real data

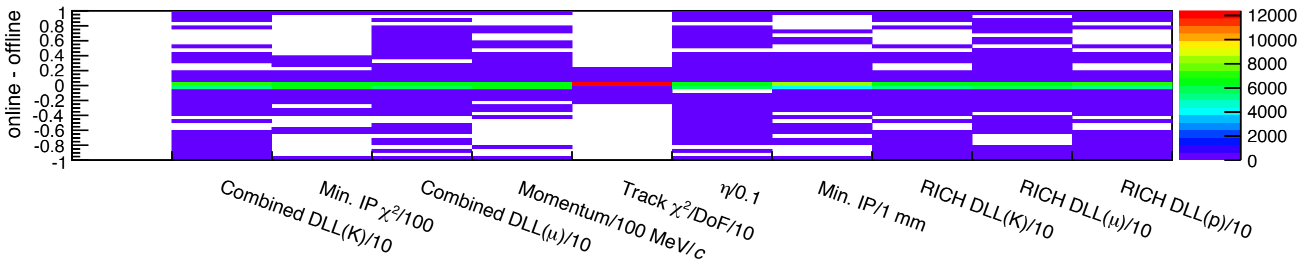
Results

LHCb-PROC-2015-011

- Every subdetector now has an alignment/calibration procedure in place
- Run 2 online reconstruction performance is now equal to that of Run1 offline:







- The alignment and calibration procedure is an unqualified success

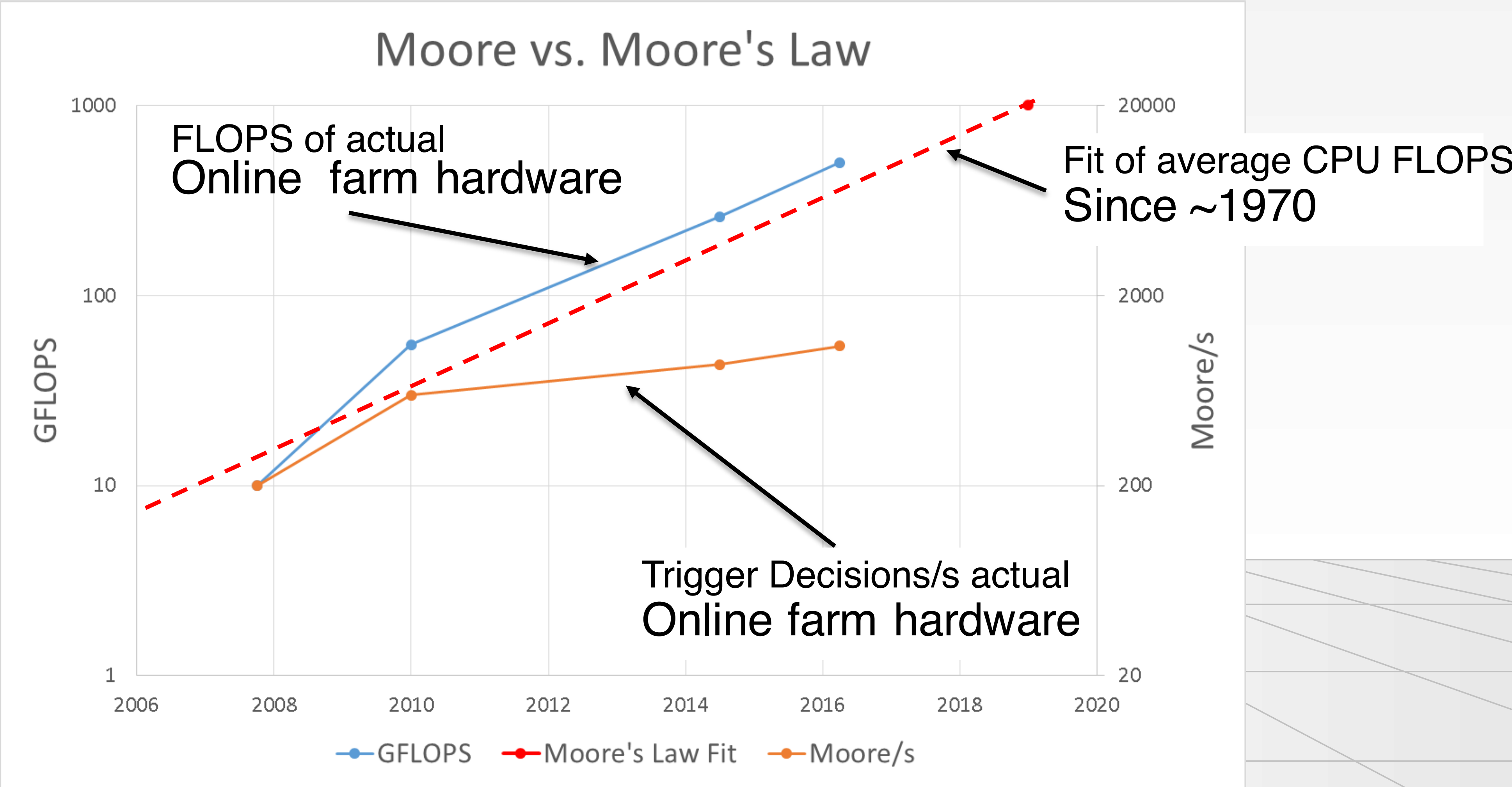
Event selection
Introduction
Run 2
Calibration in real time
Analysis in real time
Planning + Schedule

C. Fitzpatrick
May 12, 2016
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Online reconstruction at the upgrade

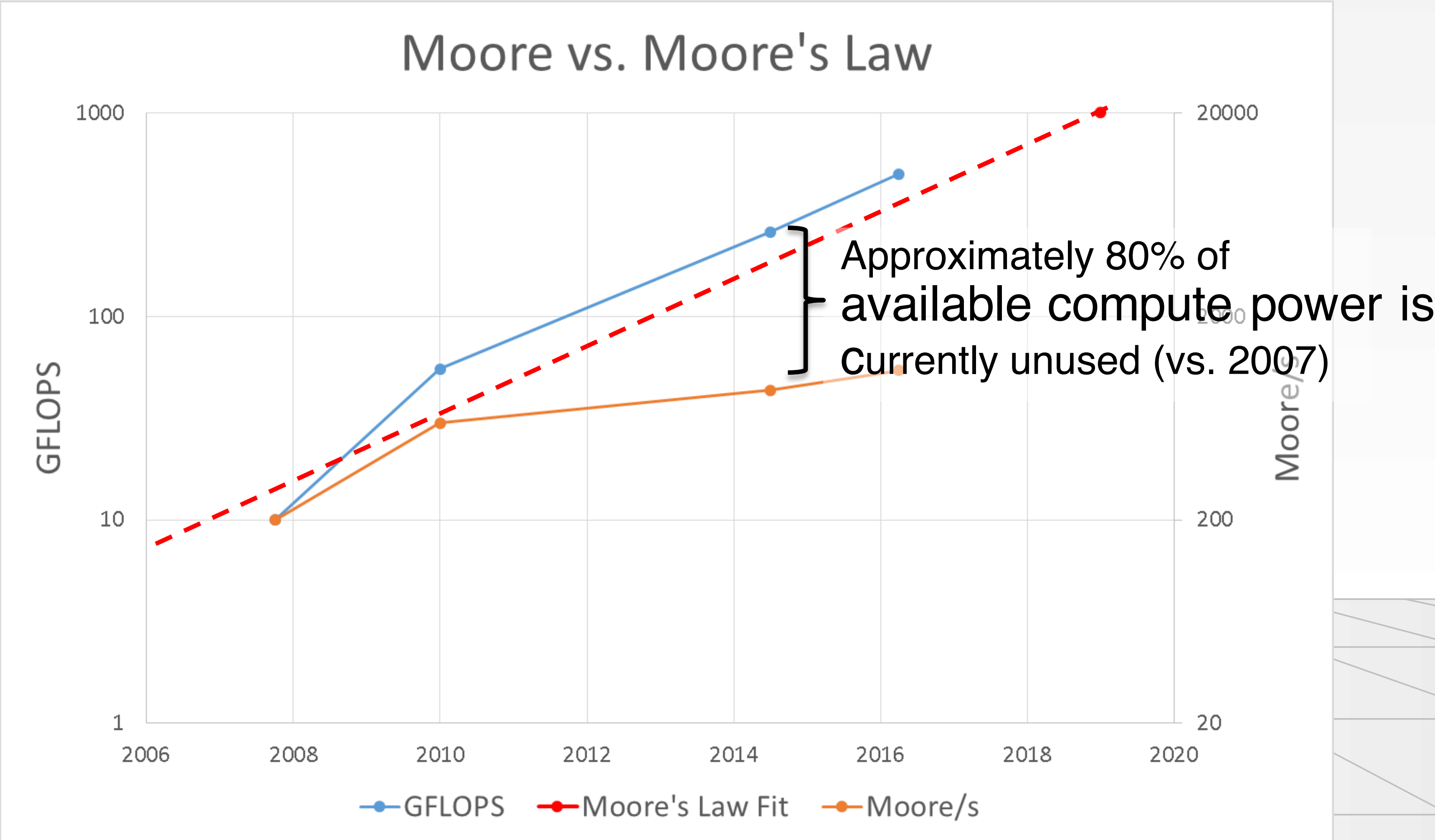
- In upgrade:
  - HLT1 input rate 30 MHz (c.f. 1 MHz now)
  - HLT2 input rate 2 MHz (c.f. 150 kHz now)
- All these events have to be reconstructed
  - Partial track reconstruction at HLT1
  - Full offline quality reconstruction at HLT2
- CPU budget for reconstruction is crucial
  - Optimise for the HLT farm hardware
  - R&D for alternative architectures
    - Since reconstruction runs only in one place (HLT farm), can optimise hardware and software together
      - x86, but also GPU, Xeon Phi, ARM, OpenPower…
    - Metric is events reconstructed per €
      - But remember code must also run on standard CPU (e.g. for offline simulation)

Moore vs. Moore's Law (© Rainer Schwemmer)



- Tests done with 2014 Farm tender benchmark
- Dates are release dates of CPU

Moore vs. Moore's Law (© Rainer Schwemmer)



- Tests done with 2014 Farm tender benchmark
- Dates are release dates of CPU

Core software R&D

- In order to fully exploit the hardware:
  - Need to evolve framework from single threaded, sequential processing of events
    - Cache misses are increasingly a problem
    - Not taking advantage of wide processing units
    - Use multiple cores to work on closely related data
  - Industry addresses these issues with "task-based" systems
    - Task-based prototype of Gaudi exists (Gaudi-Hive)
      - Rethink our algorithms to make them "stateless" and execute them as independent tasks
      - Requires also declaration of input and output data that must be immutable (changes to Event Model)
  - Current event model makes exploitation of SIMD features difficult (due to AoS design)
    - Also, not composable, makes copies expensive
    - Re-develop according to new guidelines
      - Read only
      - Composable
      - Allow choice of SoA or AoS
      - Single precision wherever possible

Hardware and Data Flow R&D

- Build a set of demonstrators to study hardware architectures and languages
  - Proofs-of-principle that well designed algorithms can significantly improve performance of many- and multi-core architectures
  - Guide hardware technology choices for the TDR next year
  - Architectures considered:
    - x86_64
    - Knights Landing (KNL)
    - GPU accelerators
    - OpenPower Foundation
    - ARM64
  - Algorithms considered:
    - Kalman Filter
    - Forward Tracking
    - RICH reconstruction

Map different possibilities in e.g.

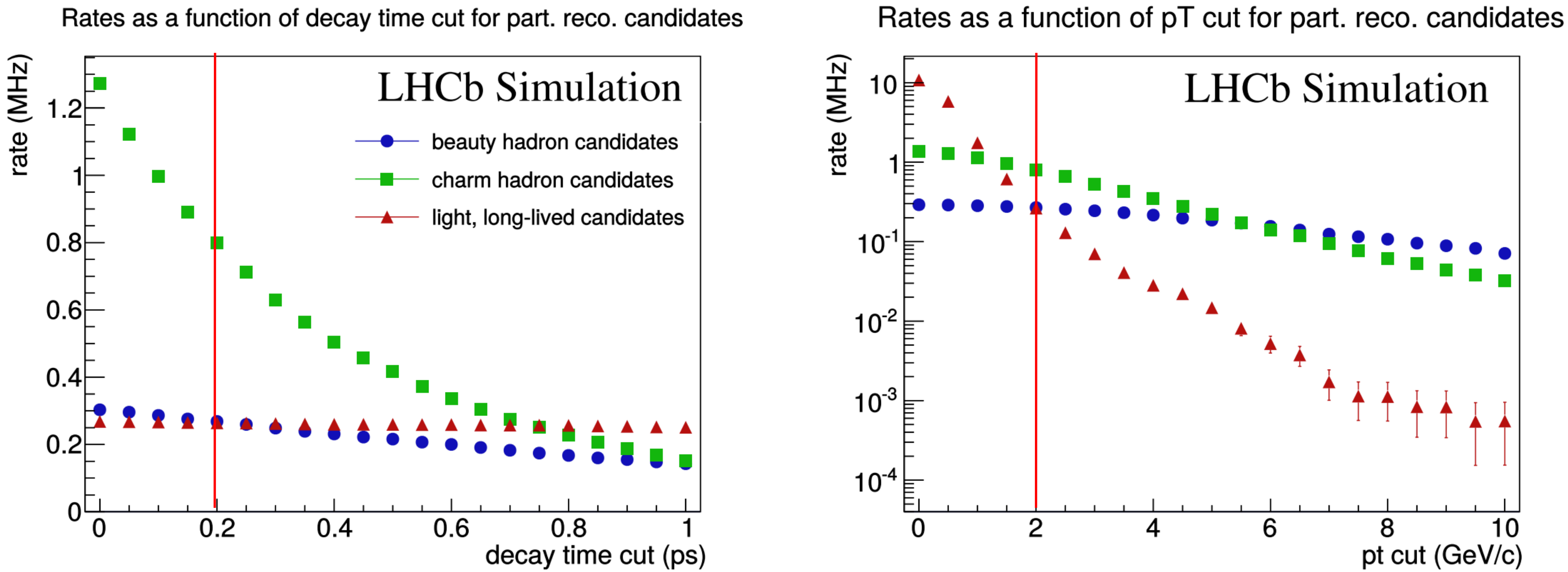
- Programming languages
- Parallel execution
- Power consumption

Parts of codebase

- With dominant execution time
- More likely to be parallelizable

Event selection: The Game Has Changed (© Tim Head)

- In the upgrade area there are no "boring" events, HLT is about classifying signal events!



  - 800 kHz of reconstructible charm hadrons,
    270 kHz of reconstructible beauty hadrons

- Offline storage cost driven by HLT output bandwidth (GB/s)
  - Optimisation of event size crucial for selection efficiency

The Game Has Changed

Triggers today

Triggers in the future

83

(© Vava Gligorov)

The Game Has Changed

- Trigger is no longer selecting events, but classifying them
  - Write out what bandwidth and offline resources allow, but everything written out will be analysed

- In many exclusive analyses, interested only in the decay tree of the triggering signal
  - So write out only the interesting part of the event, not the whole event
  - Turbo Stream idea, see next slide

- If all events are interesting offline, current model of stripping no longer applies
  - Streaming more relevant, but how many streams?
  - Is direct access to individual events more relevant?
    - Needs event index, and R&D on efficient access to single events

The Turbo stream (© Sean Benson)

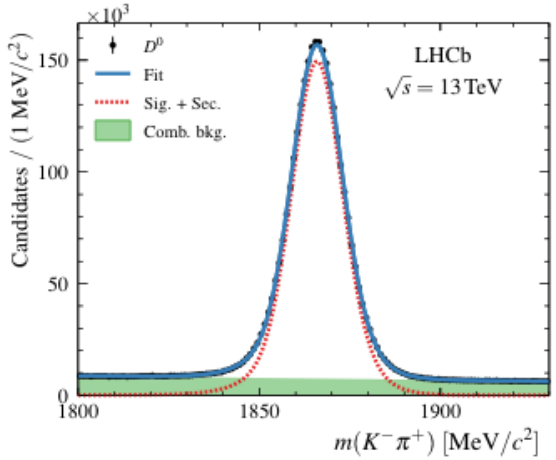
- Offline quality reconstruction and PID in HLT2 allows to do physics selection at HLT2
- For many analyses, sufficient to store tracks participating in decay of interest -> "Turbo" stream
  - 10-20 times saving in event size stored (5-10kB/evt)
  - Does not need offline reconstruction
  - Can be used directly for offline analysis
    - Fast turnaround
    - In 2015, of 374 HLT2 trigger lines, 185 chose Turbo
- Opens up possibility to record an order of magnitude more of interesting events (kHz) for a given cost
  - Clearly some analysis techniques will always require full event
    - For a given bandwidth, adjust the HLT processing strategy to exploit it (e.g. adjust the ratio of full events at ~100kB and turbo events at ~10kB).

Turbo analyses

- 2015 early measurements performed exclusively with Turbo:

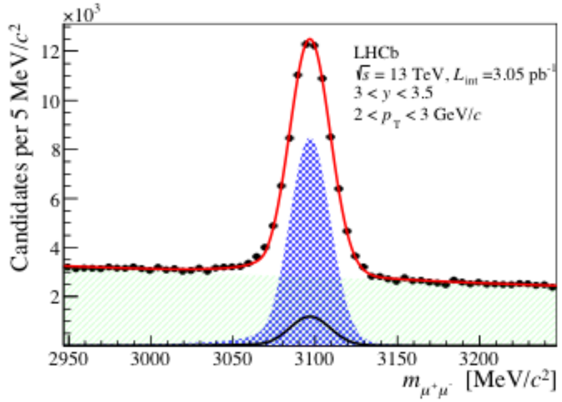
CERN-PH-EP-2015-272
LHCb-PAPER-2015-041
6th October 2015

Measurements of prompt charm production cross-sections in pp collisions at $\sqrt{s} = 13$ TeV



CERN-PH-EP-2015-222
LHCb-PAPER-2015-037
September 2, 2015

Measurement of forward J/ψ production cross-sections in pp collisions at $\sqrt{s} = 13$ TeV



- Several more involved analyses underway

C. Fitzpatrick
May 12, 2016
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Output event rate at the upgrade

- Design range for offline storage is HLT output rate of 2-5 GB/s
  - This represents huge range of event rates, depending on mix of Full events and Turbo events:
    - 2 GB/s of Full events (100kB) -> 20kHz
    - 5 GB/s of Turbo events (5kB) -> 1MHz
  - Reality will be somewhere in between, physics optimisation of bandwidth to be done
- All real data reconstruction done online: main implication for offline CPU resources is CPU for simulation
  - Factor 50 in events to be simulated between two extremes above
    - This will be important ingredient of physics optimisation
  - Clear that major development effort in Fast MC techniques is needed
    - Already started for Run 2 physics, focusing on reducing needs for full simulation
      - Parametric approaches, partial event simulation

Towards a computing model for the LHCb upgrade

- We do not plan a revolution for LHCb upgrade computing
- Rather an evolution to fit the following boundary conditions:
  - Luminosity levelling at $2x10^{32}$
    - Factor 5 more than in run 2
  - 100kHz HLT output rate for full physics programme
    - Factor 8-10 more than in Run 2
    - With tunable event size to fit in 5 GB/s
  - Flat funding for offline computing resources

Assumption →

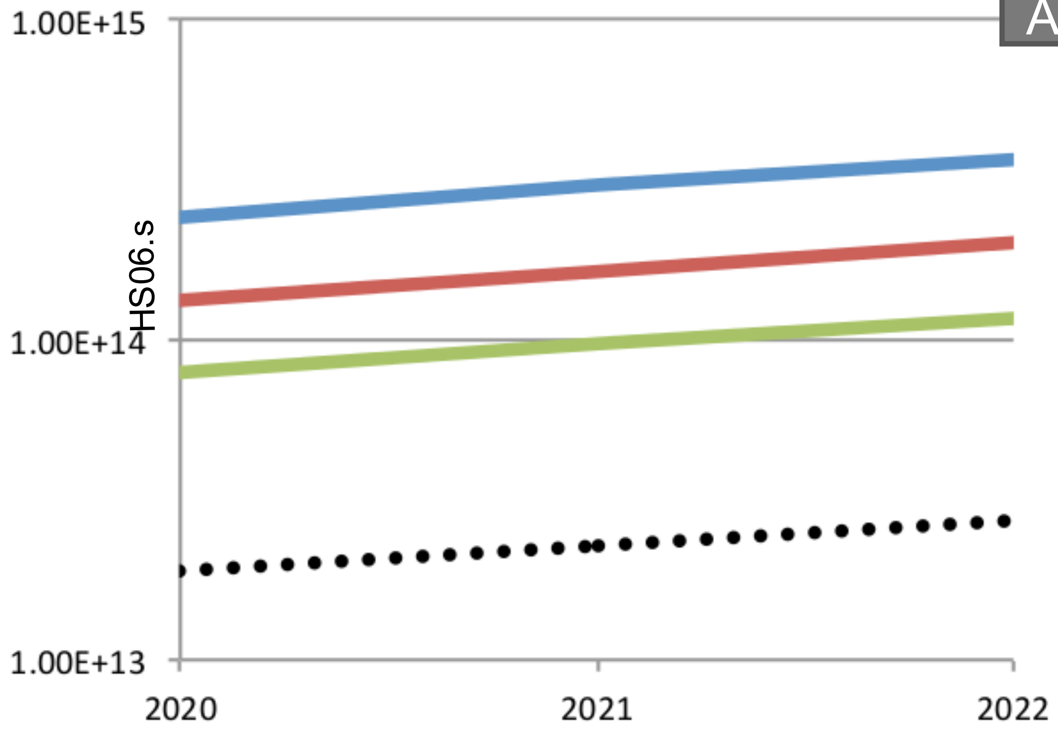
20%CPU/year
15% disk/year
25% tape/year

Monte Carlo Simulation

- Run 1 simulated events ~ $4.5*10^9$ (spring '15)
  - ~ 12 % of recorded
- Aim to simulate 100 % of recorded

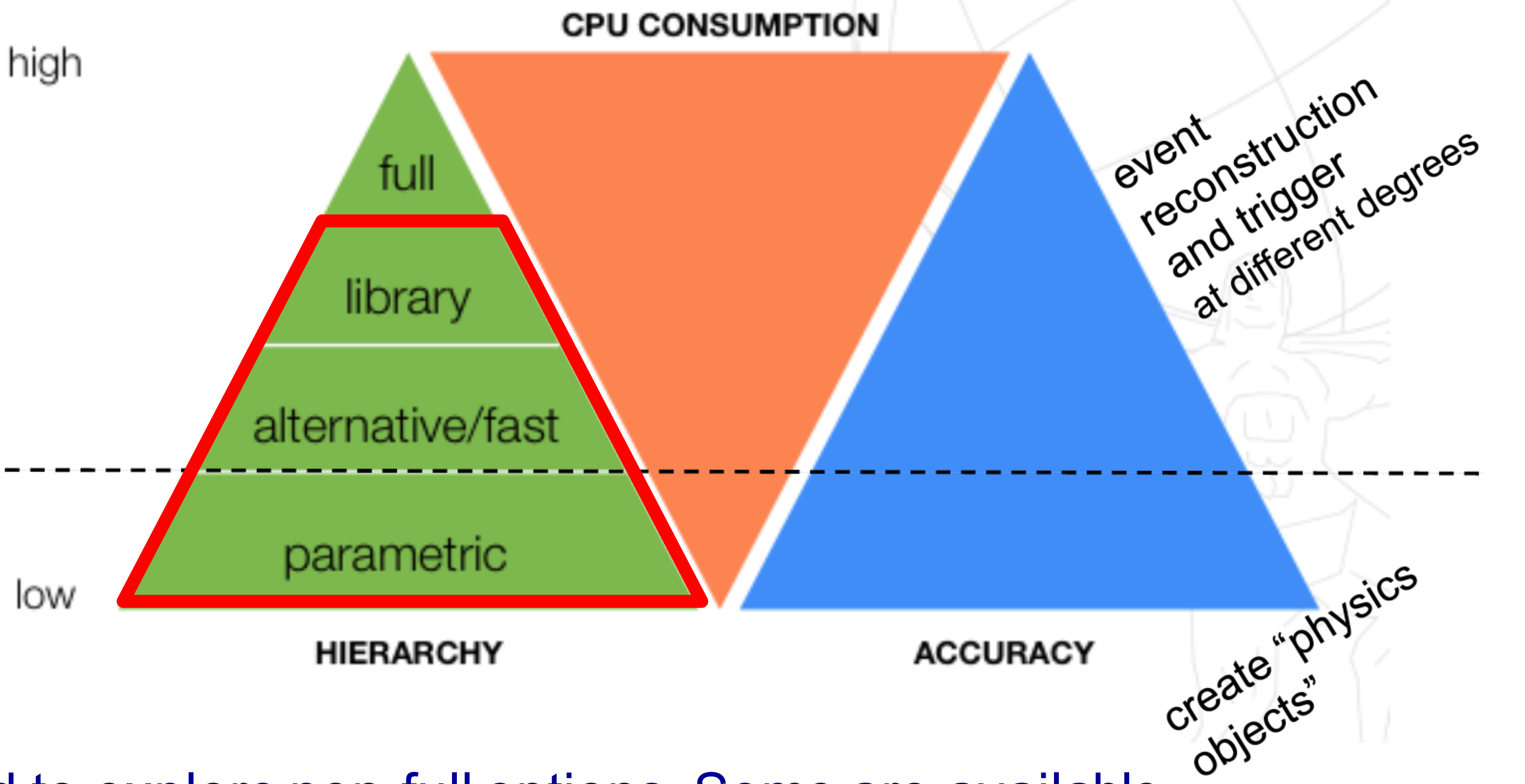
Assumption

- Full Sim 600 HS06.s (curr 3-5 times that)
- Fast Sim 10% of Full Sim



MC won't fit (by far) into the pledged resources :-(

How do we do it?



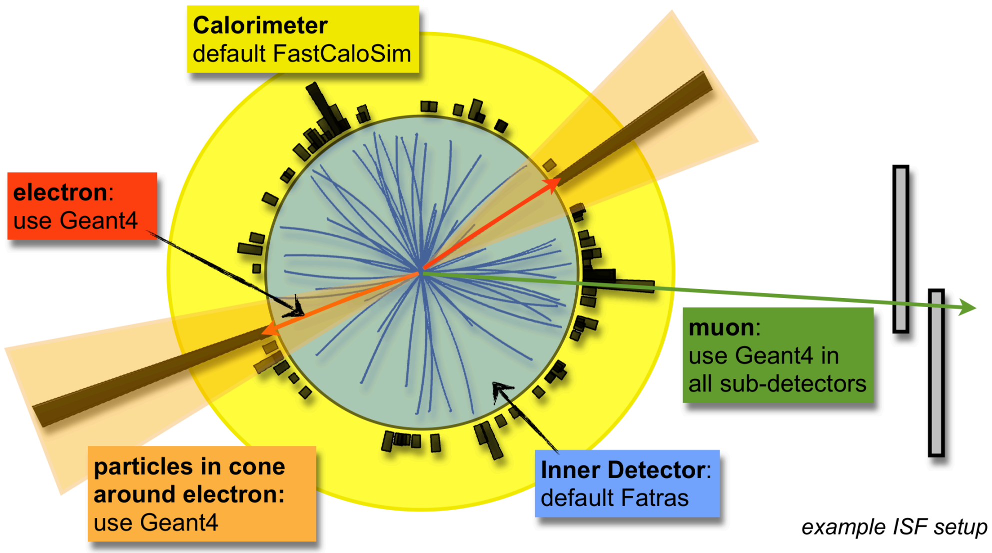
Need to explore non-full options. Some are available or require 'little' work, others more, others ??

© ATLAS

G. Corti, A&S Week, Nov. 2015 & Z. Marshall, Paris Computing Workshop Nov 2015

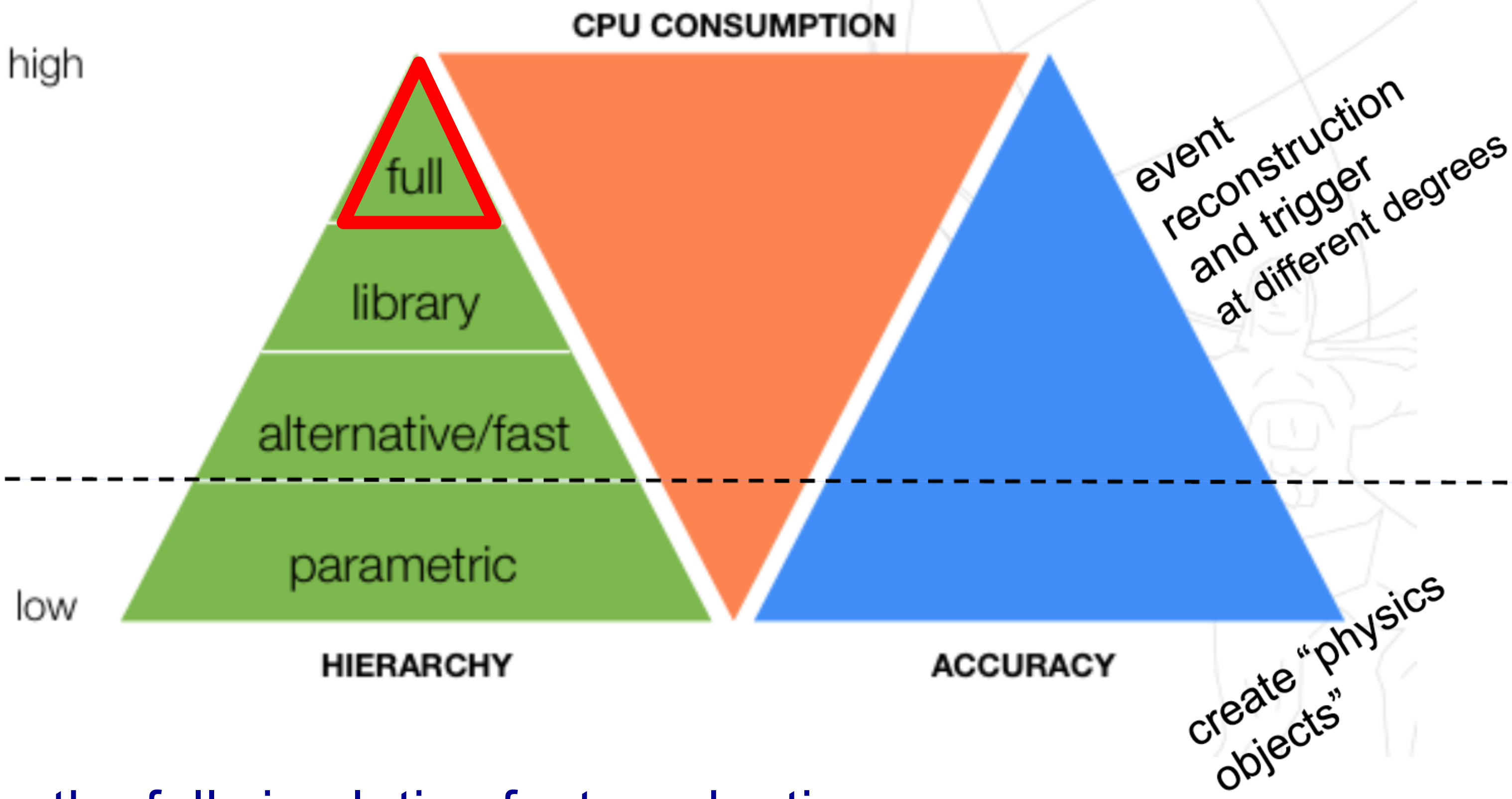
Simulation Framework

- ATLAS has recently introduced an **Integrated Simulation Framework**
  - Allows to mix simulation flavors for different particles in the same event
- In LHCb framework (Gauss) it is possible to replace simulation flavours for the whole event and with some changes in the implementation of its interaction with G4 it should be able to provide the same functionality
  - We need to prove it!
  - Make it easy
  - And then we choose the best mix



"We want FASTER simulationS"

How do we do it?



Make the full simulation faster adopting new technologies and optimizing code

© ATLAS

Adopting new technologies for simulation

- Migration to Geant4 10 ongoing
  - Some simple speed up expected
    - Faster simulation with Geant4 10 in sequential mode (ATLAS – 15%)
    - Faster simulation with Geant4 static libraries (CMS – 10%)
  - Reduced memory footprint with multi-threading, more cache friendly
- Investigate modern geometry packages
  - Usolid library (will become G4 default)
  - VecGeom (developed for GeantV)
    - Targets use of SIMD vectorization
    - Library support for GPUs
- Geant 4.10 (and GeantV) are enabling technologies
  - Not LHCb specific
  - LHCb will directly benefit from common developments and hardware optimisations

Comments on CPU

- Run 3 CPU dominated by simulation
  - More so than in Run 2

- Simulation CPU can continue to be anywhere
  - Current WLCG distributed model
  - Leverage on opportunistic resources for simulation
    - Including e.g. commercial clouds spot market
  - GPU for simulation could be used if supported by GEANT – not LHCb specific

- Analysis CPU should remain "close to the data"
  - Depends on analysis model, see next slides

Data processing/access and analysis models

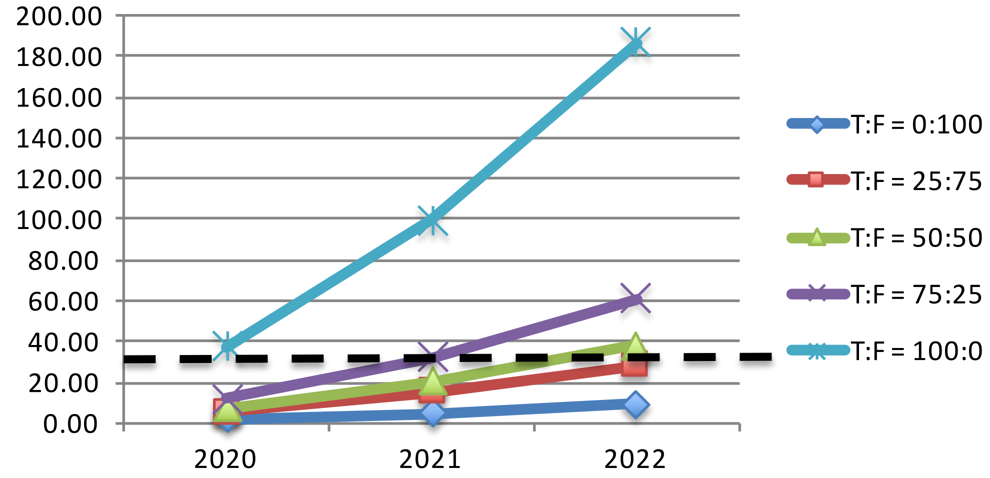
- Successful run 1 dataflow does not scale to Run 3
  - RAW event storage too expensive, stripping does not scale
- Run 3 concepts to be addressed in 2016, using current framework:
  - Turbo stream by default in Run 3
    - Flexible data formats for saving reconstructed event and not the RAW data
      - Varying level of detail depending on the triggering analysis
      - E.g. storing cone of tracks around selected candidate
  - Event Index with random access
    - Stripping would flag events rather than copy them.
    - Important to understand I/O performance
    - Follow Atlas developments in this area
  - Centralised Ntuple production
    - Investigate organising "trains" of Ntuple production
  - Evolution of distributed computing
    - E.g. handling of random access to events
    - More dynamic replication of data using data popularity

Brainstorming: storage model

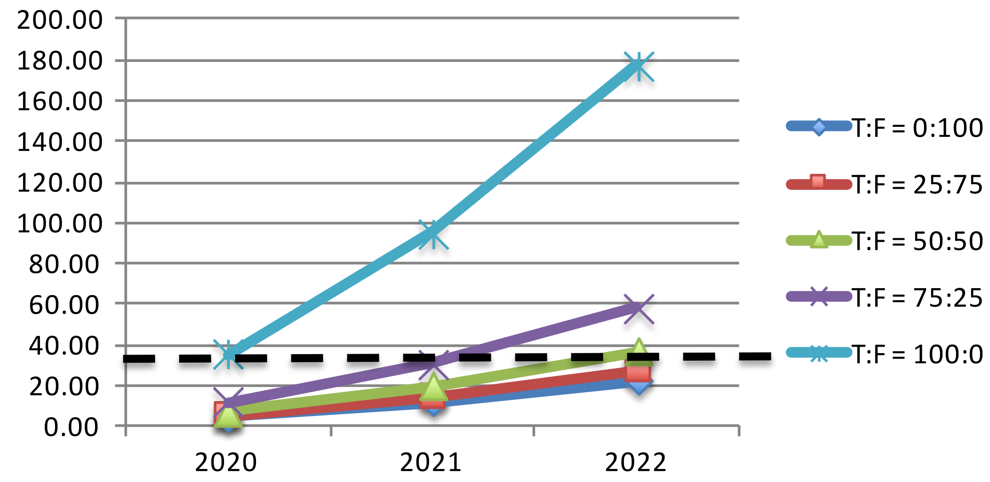
- Three classes of storage
  - Disk for active data analysis
    - Real data, frequently accessed simulation
  - Active Tape
    - Less frequently accessed simulation
    - Migration between disk and tape based on popularity predictions
  - Archive Tape
    - Only for data, simulation and analysis preservation
      - No need for large disk cache
      - No I/O latency constraint, can be outsourced?
- A few sites for disk, even fewer for tape
  - ~3 sites with active tape sufficient
  - Sufficient disk sites to provide low latency and high availability for analysis jobs
    - No technical need for many small disk pools
      - (but recognise it as important funding/sociological issue)
- Investigate (with other experiments) role of specialised databases – e.g. for event index, conditions database
  - Cannot afford to make them LHCb specific developments

Disk (5GB/s)

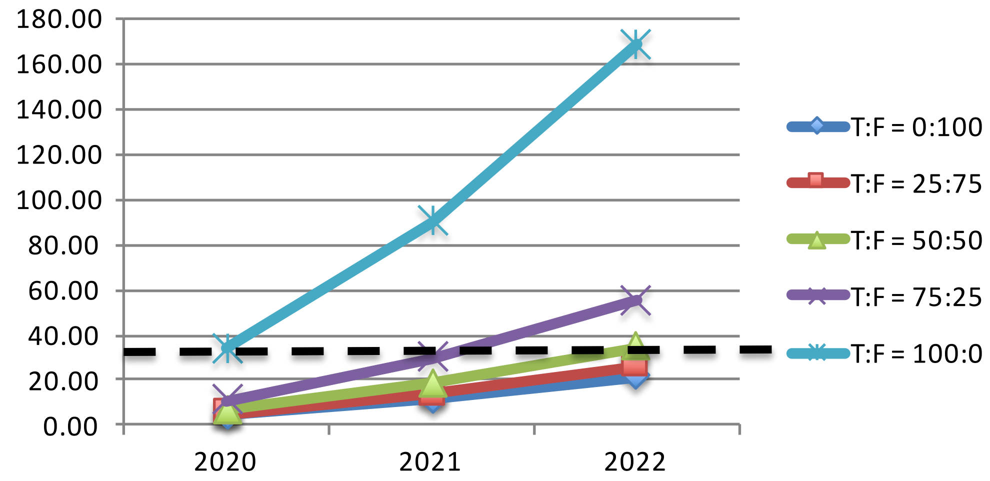
Total DISK (PB), 5GB/s, NO FAST MC



Total DISK (PB), 5GB/s, 50% FAST MC



Total DISK (PB), 5GB/s, FAST MC only

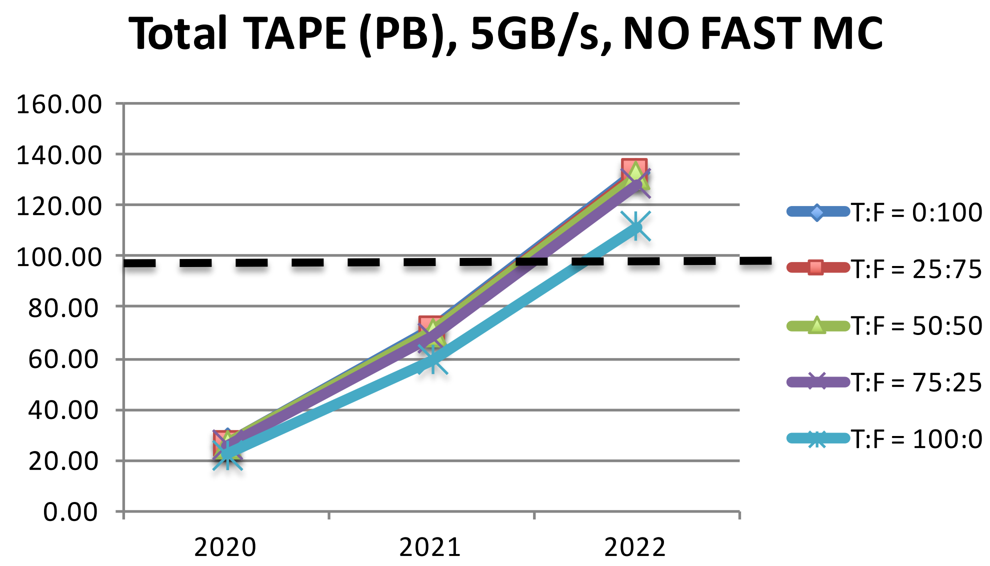


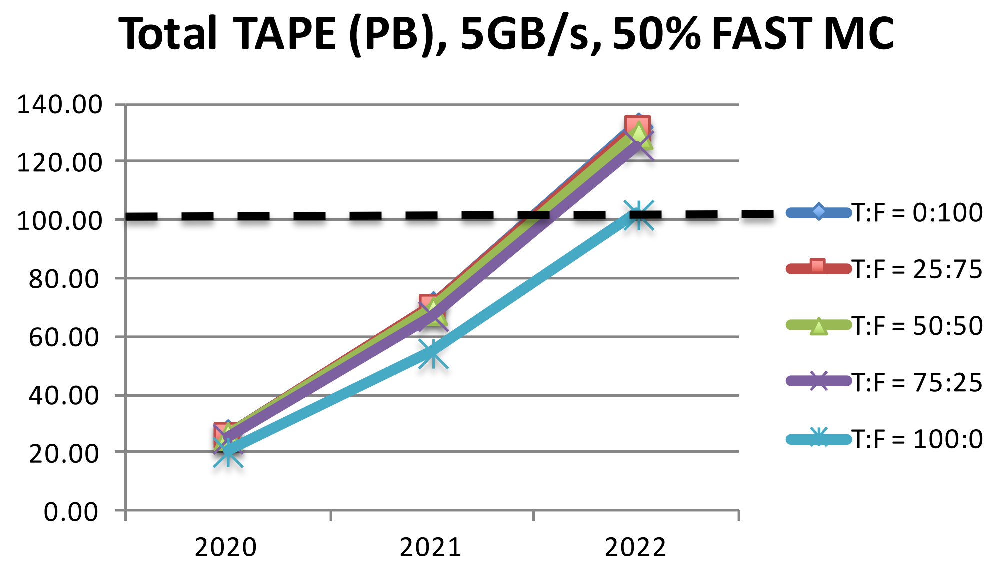
Dashed line = Expected resources at end of Run2

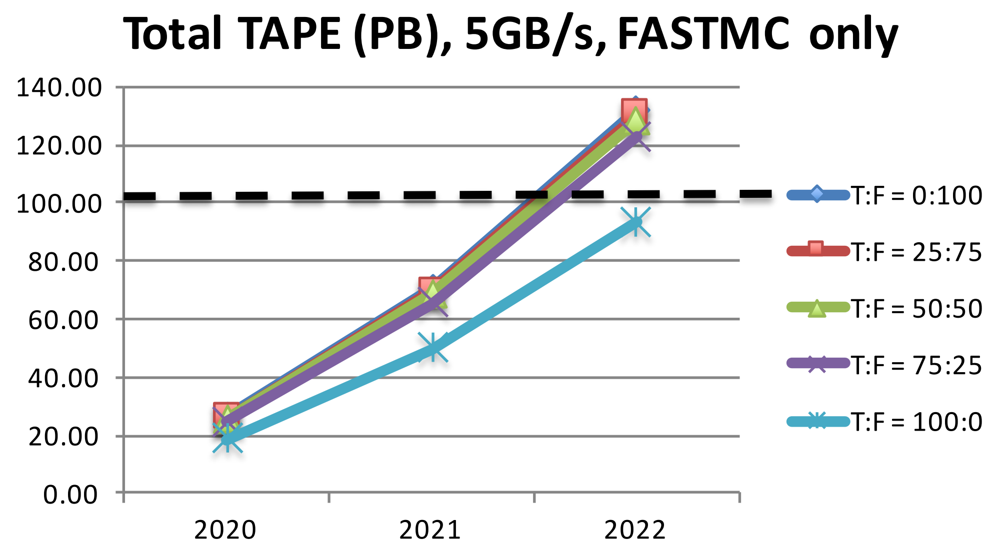
Comments on disk

- Disk 2018 – 2019 ~35PB
  - 20PB data + 15PB MC
  - 4(3) copies of most recent data(MC) processing, 2 for previous one
- Run3 disk kept at manageable level by
  - Reducing number of copies: 2 for data, 1 for MC
  - Having a large fraction of mDST MC
  - Keeping only most recent processing on disk (implicit in the current model)
- Please note that plots refer only to disk needed for Run3. Keeping Run2 data on disk will add significant overhead

Tape (5GB/s)







Dashed line
=
Expected resources at end of Run2

Comments on tape

- Run1+2 tape: ~100PB, dominated by RAW
- Run3 tape: in the ballpark of Run2. Increasing TURBO rate
  - decreases data due to smaller event size
  - Increases sim data due to larger number of events to be produced
- Please note that plots refers to tape needed for Run3. Tape space needed for Run1+2 increases tape by a factor ~2

Conclusions

- LHCb upgrade is around the corner

- Major (r)evolution of computing model to take advantage of full reconstruction online
  - Most concepts already proved in Run 2
    - But need 1-2 orders of magnitude more efficient computing to make it reality
    - Tape and Disk requirements under control

- Major challenges are software
  - Modernisation of reconstruction and simulation to take full advantage of modern hardware
  - R&D on analysis models for sparse access to 1-2 orders of magnitude more (smaller) events

- Role of accelerators will become clearer in 12-18 months

LHCb Computing

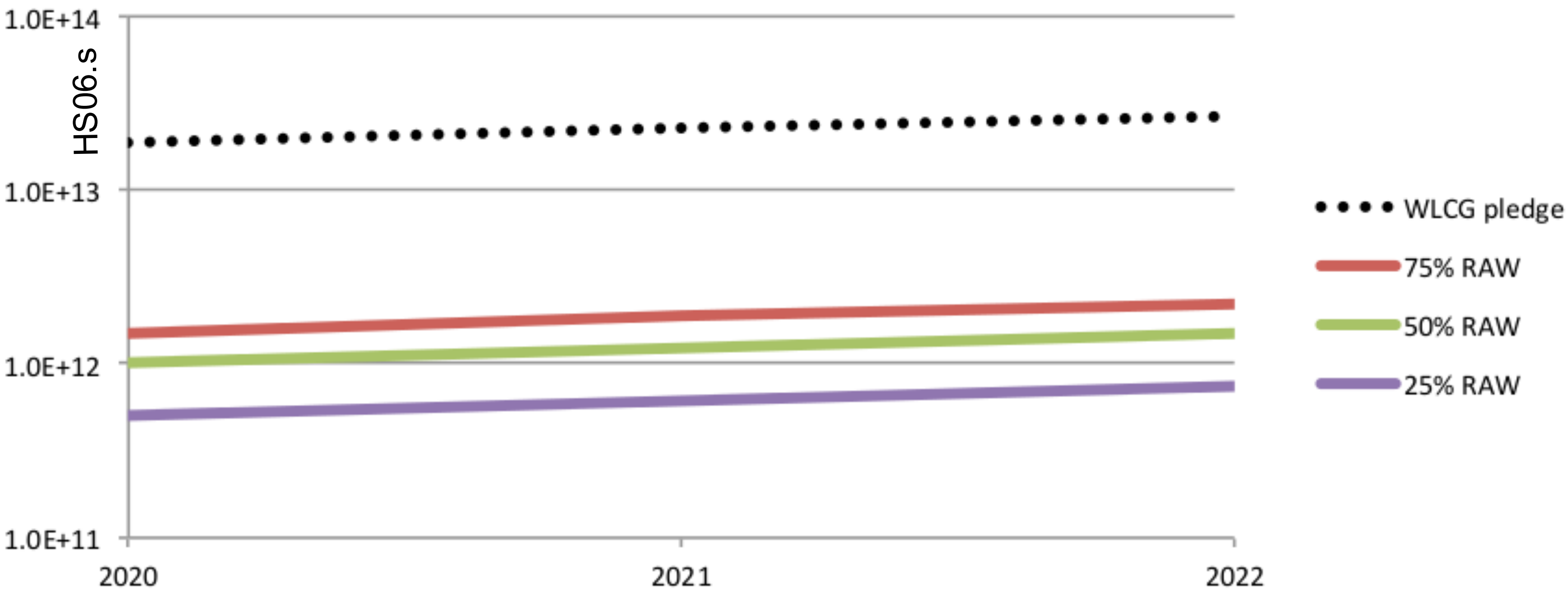
Backups

Data Processing

- Run 3 yield will be $1.5*10^{12}$ events
  - Run 1: $3.8*10^{10}$ events

Assumption

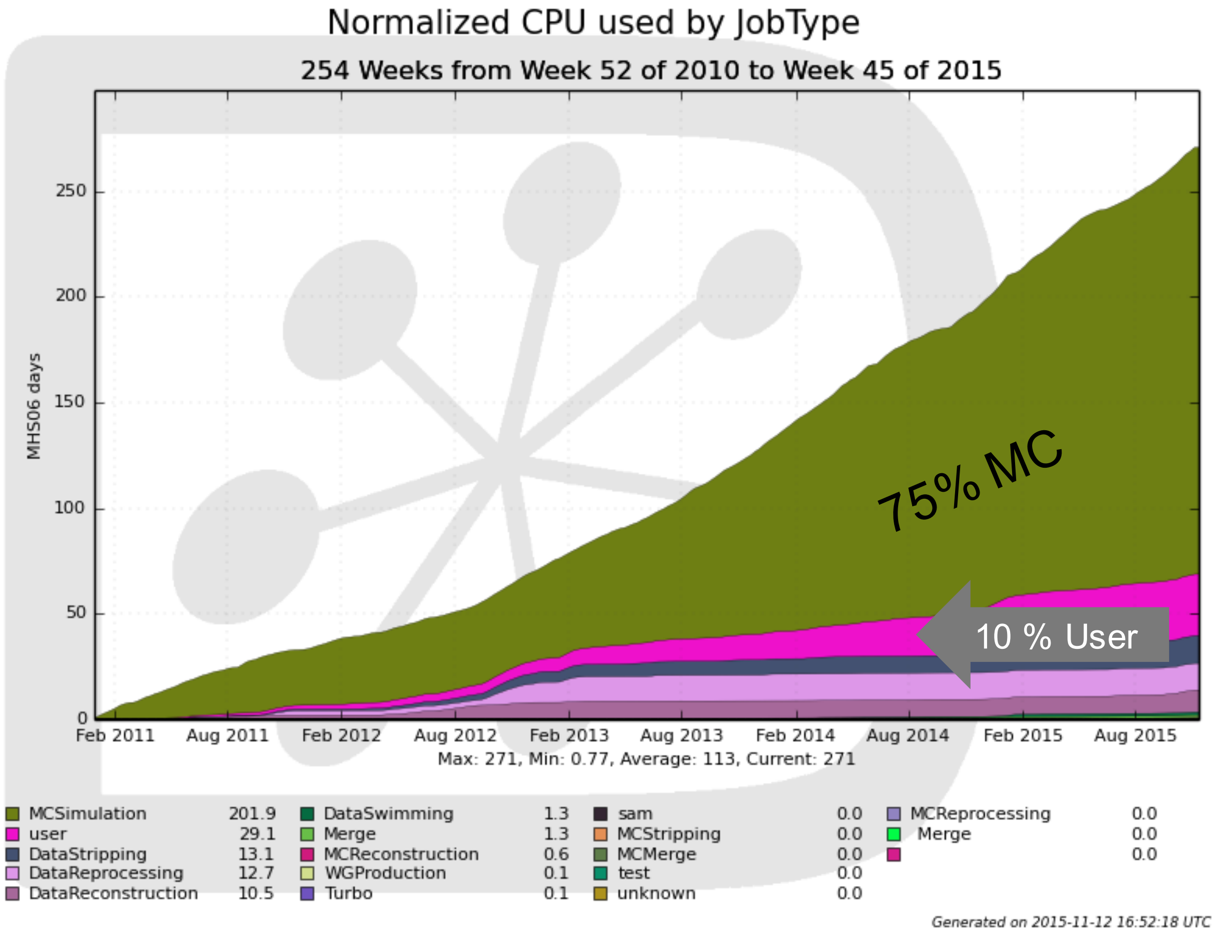
- 4,5,6 Mio sec pp collisions / year
- Reco .5 sec / event (current)

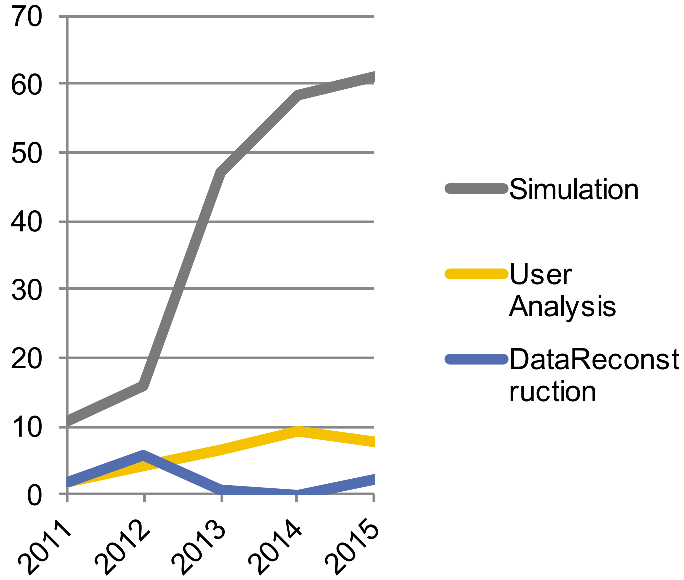


Data processing will fit into the pledged resources ;-)

OK

Already now, simulation using 75% of CPU resources



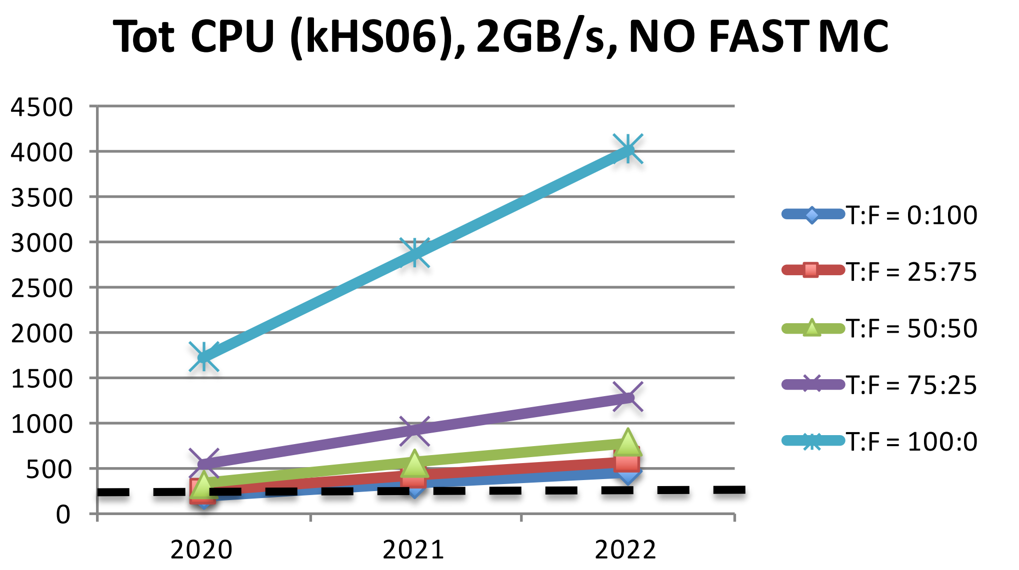


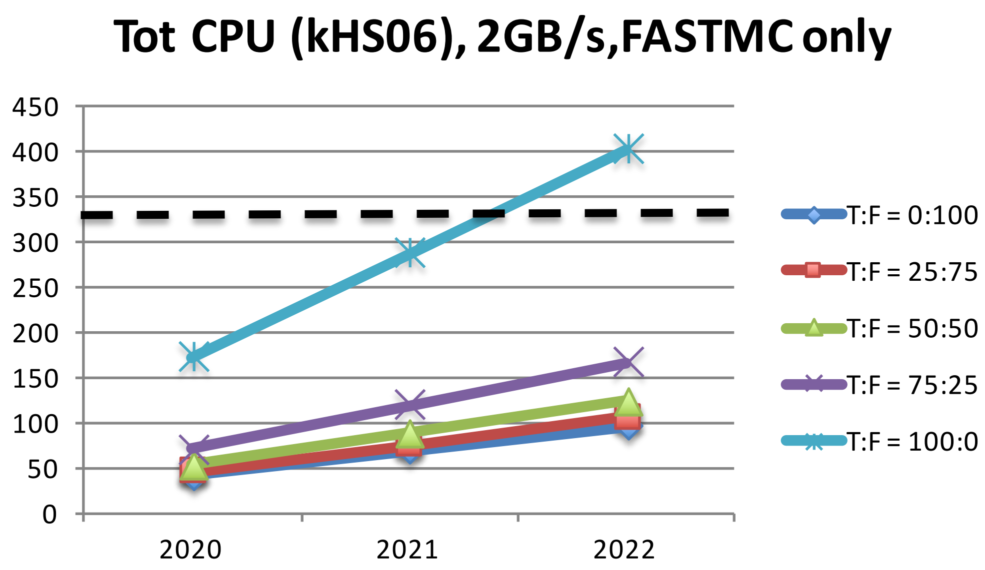
- Simulation: Bulk of work
- DataReconstruction
- User Analysis
  - 2012-2014 steady increase by ~ 50 %

Resources estimates

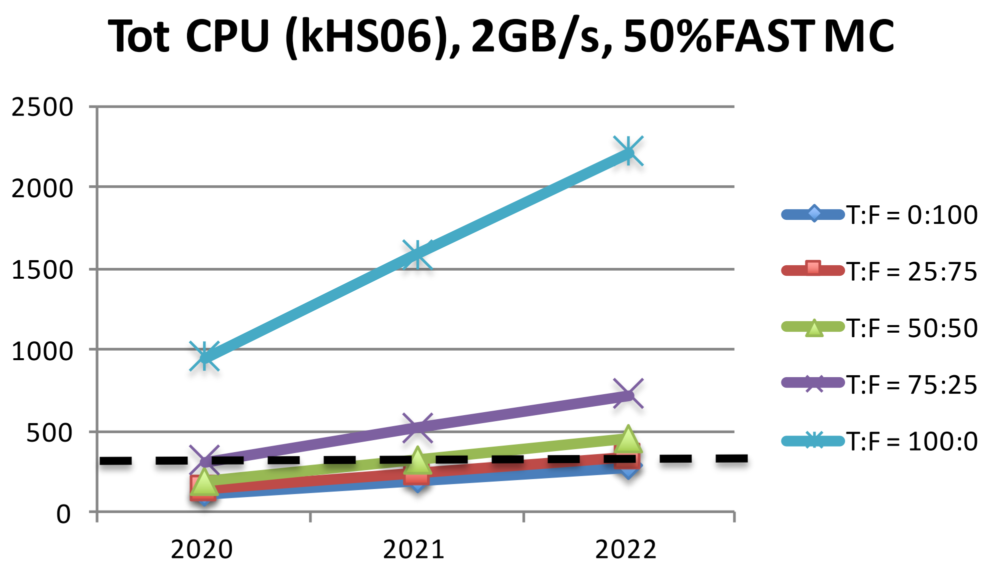
- Consider two scenarios:
  - “minimal” 2GB/s, corresponding to a B-physics only experiment at 20kHz)
  - “maximal” 5GB/s, corresponding to a full physics program at a rate to be defined
- Adjust the ratio of full events at ~100kB and turbo events at ~10kB to evaluate various total rate scenarios
  - T:F = 0:100, 25:75, 50:50, 75:25, 100:0
  - In terms of RATES (==kHz)

CPU plots (1): total 2GB/s

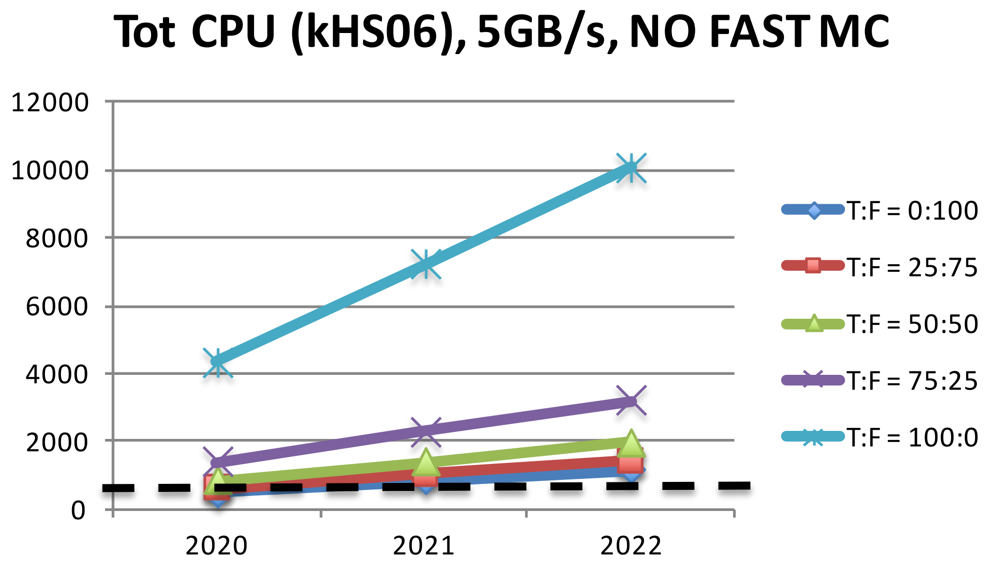


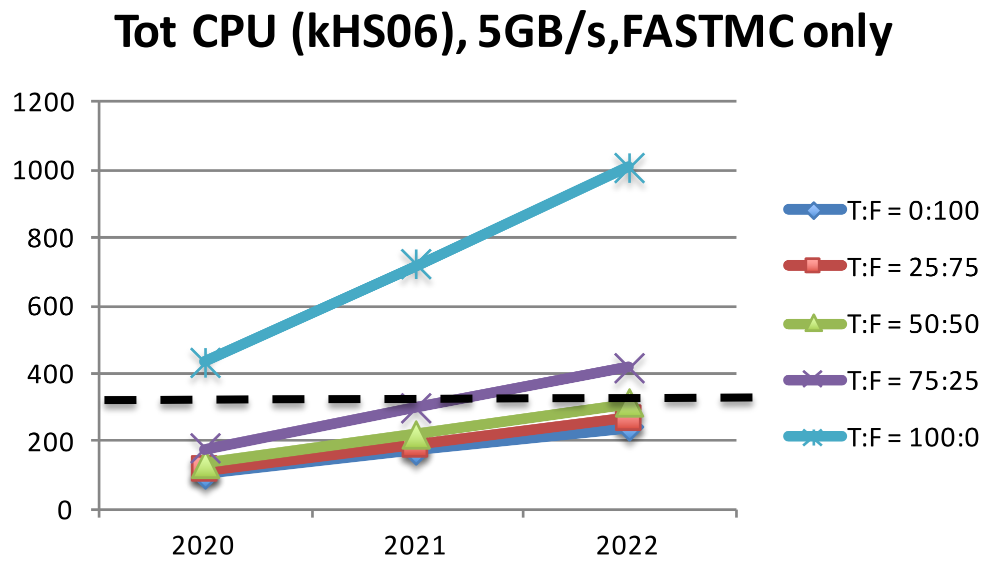


Dashed line = Expected resources at end of Run2

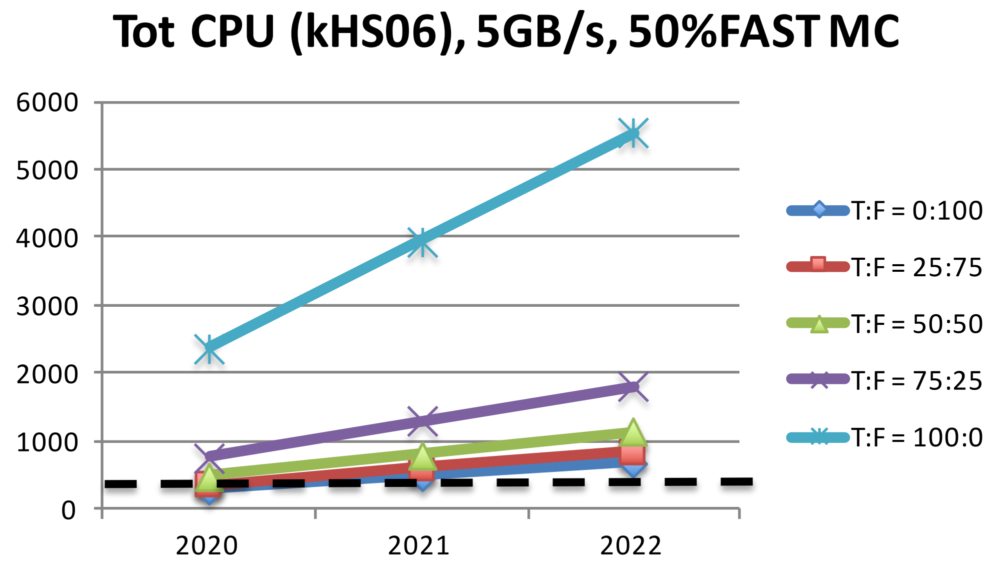


CPU plots (2): total 5GB/s

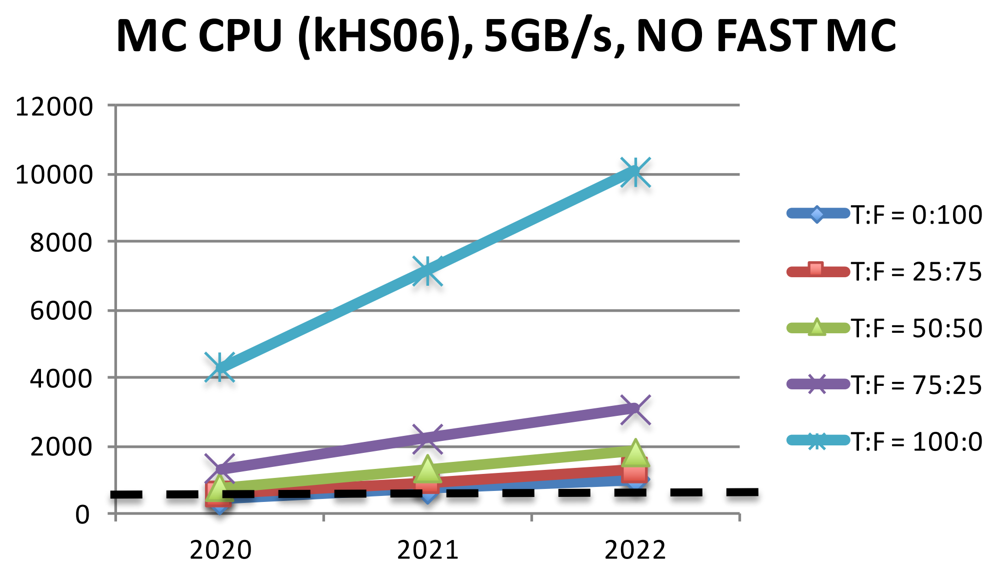


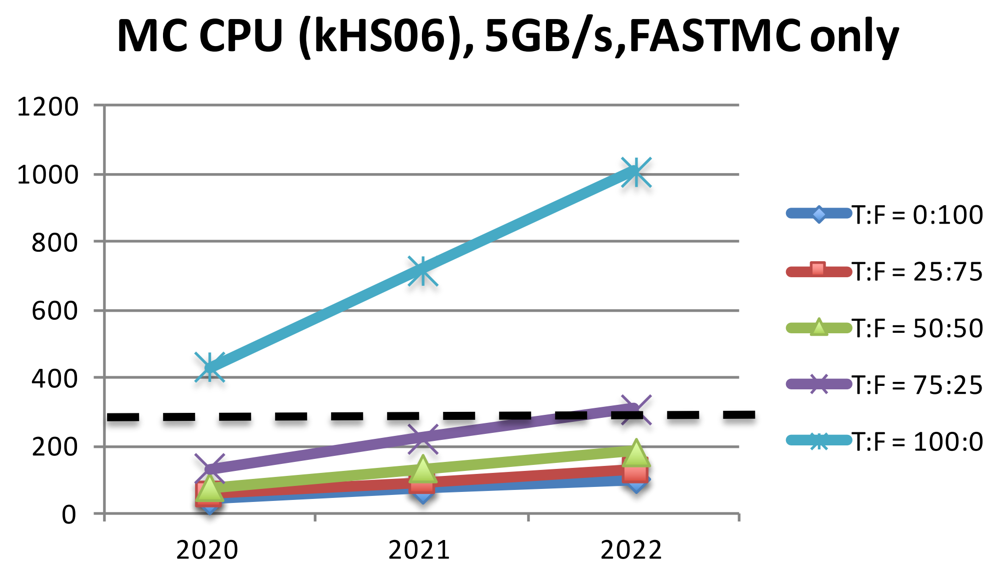


Dashed line = Expected resources at end of Run2

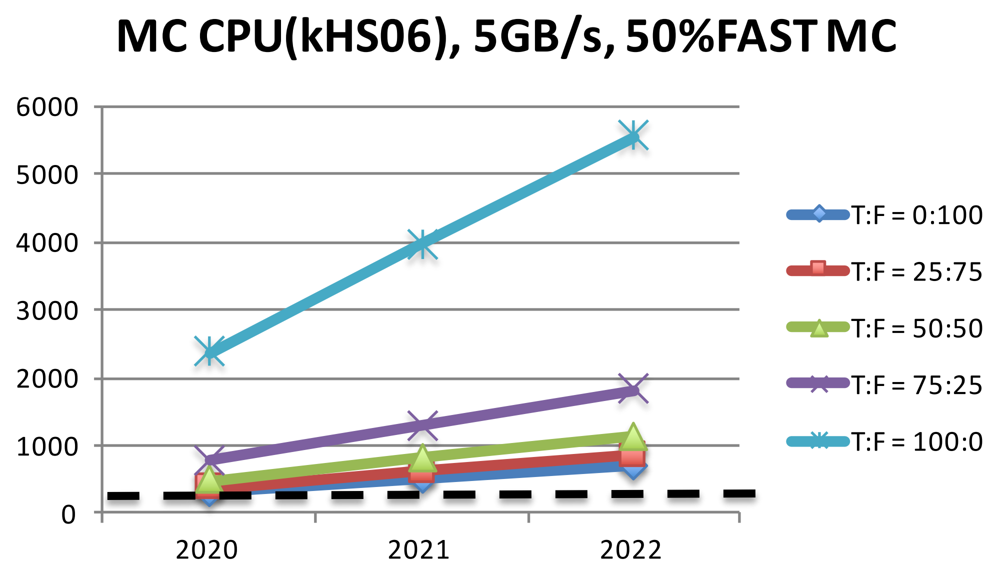


CPU plots (3): MC 5GB/s

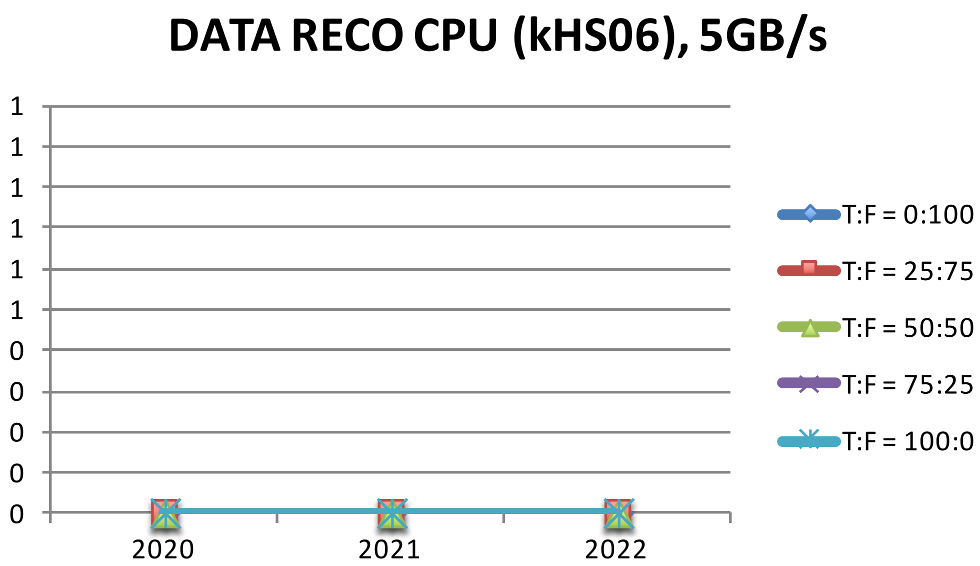


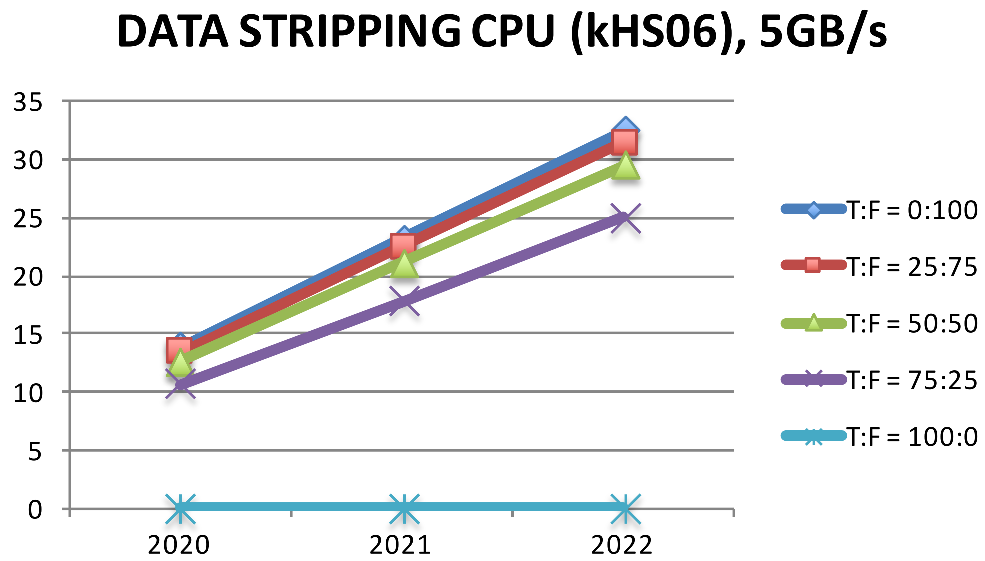


Dashed line = Expected resources at end of Run2

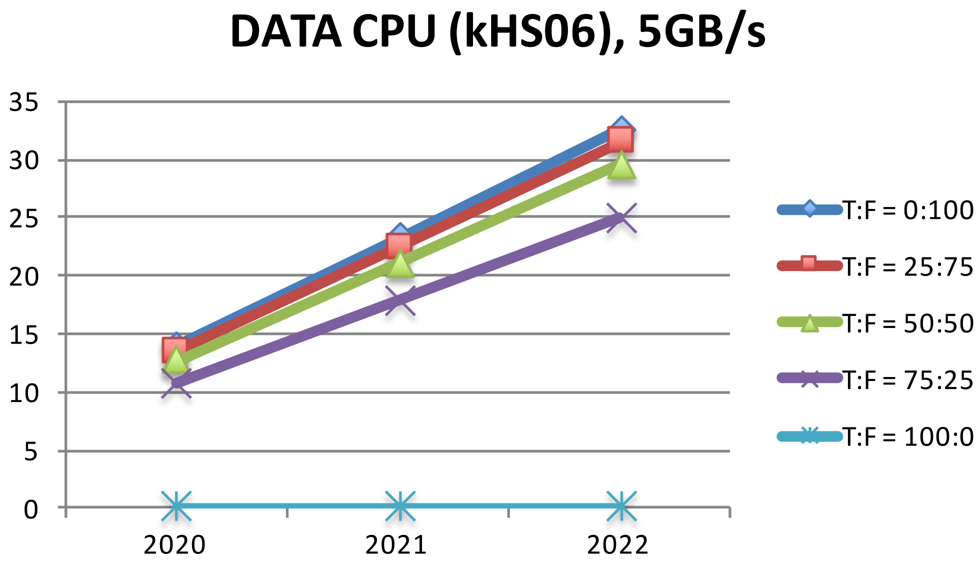


CPU plots (4): data 5GB/s

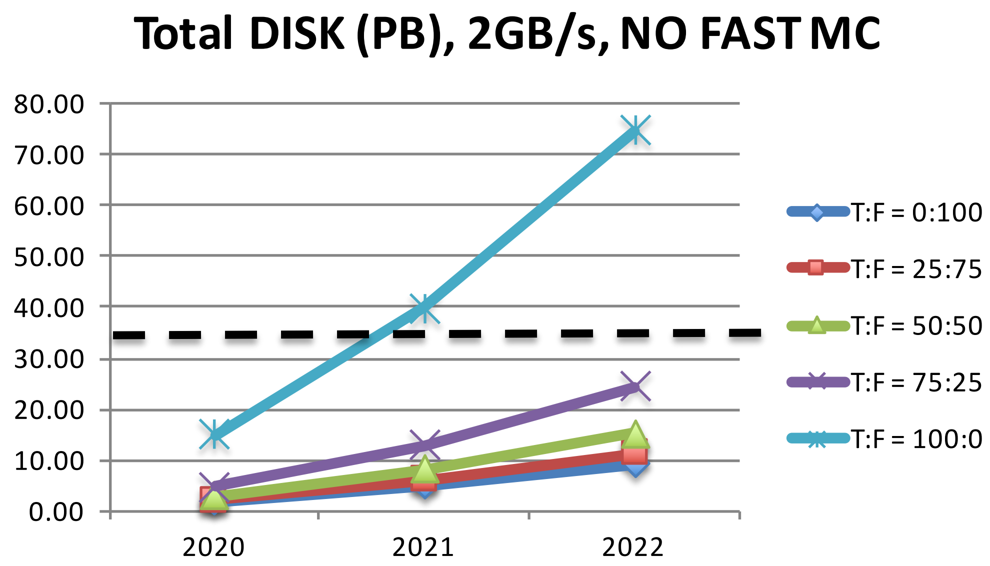


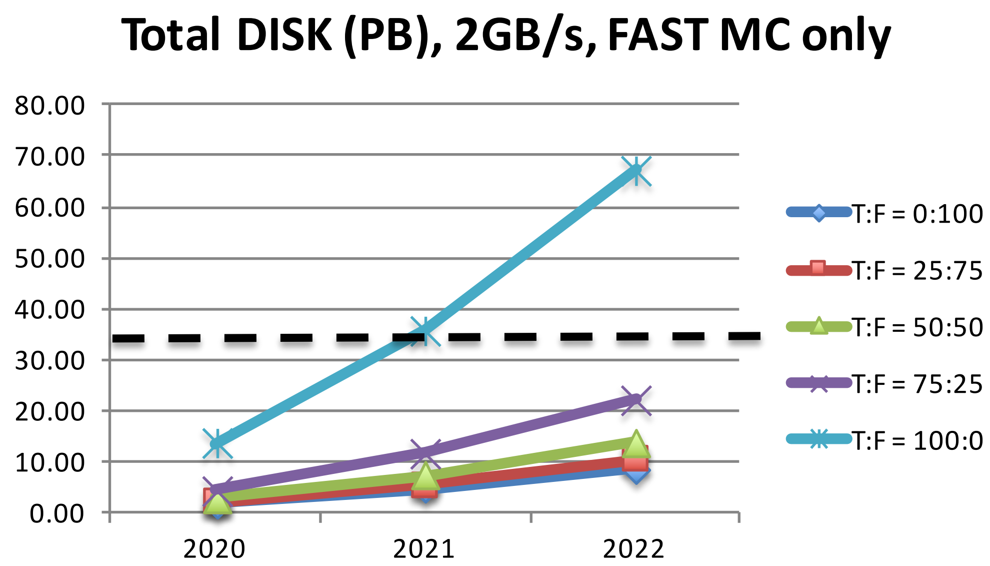


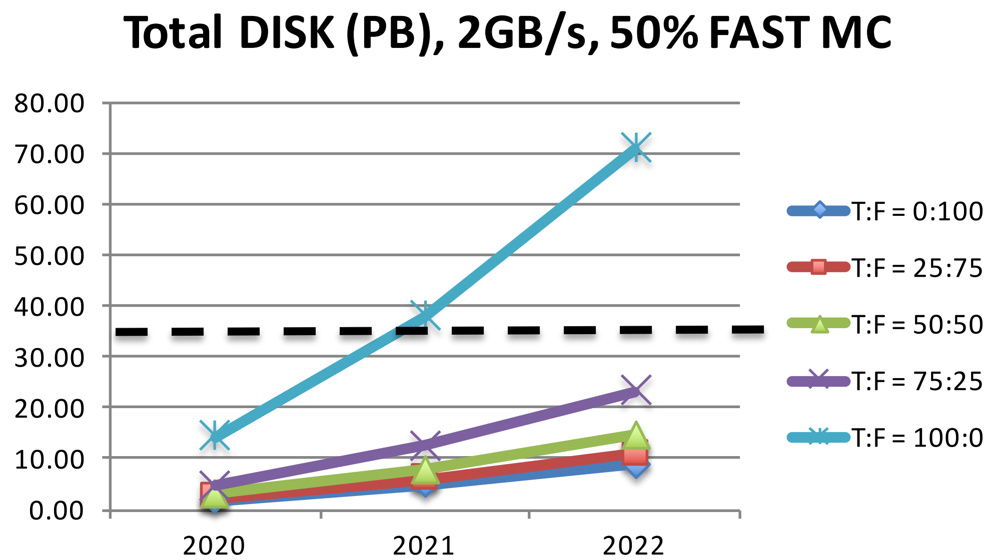
Well below expected usage at end of Run2 (50kHS06)



Disk (2GB/s)

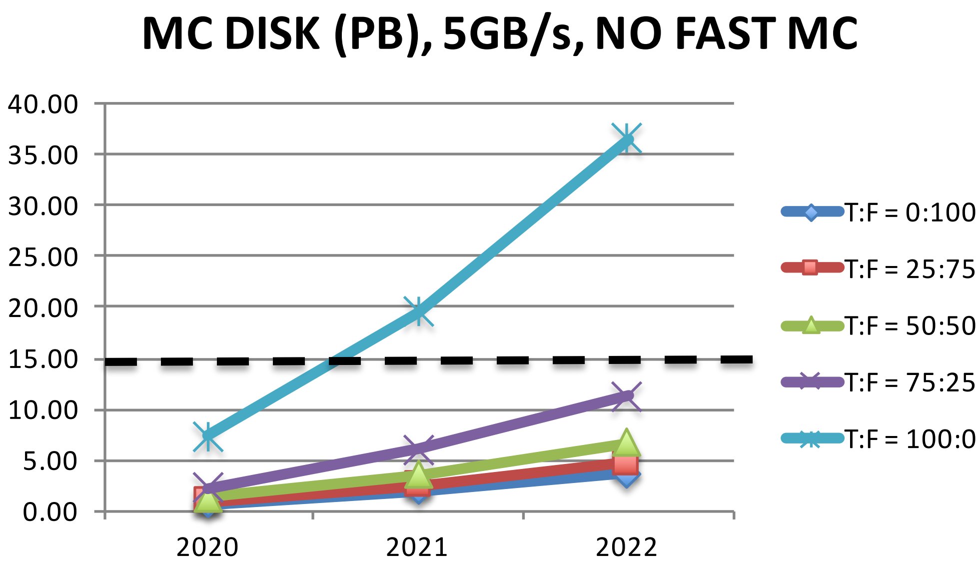


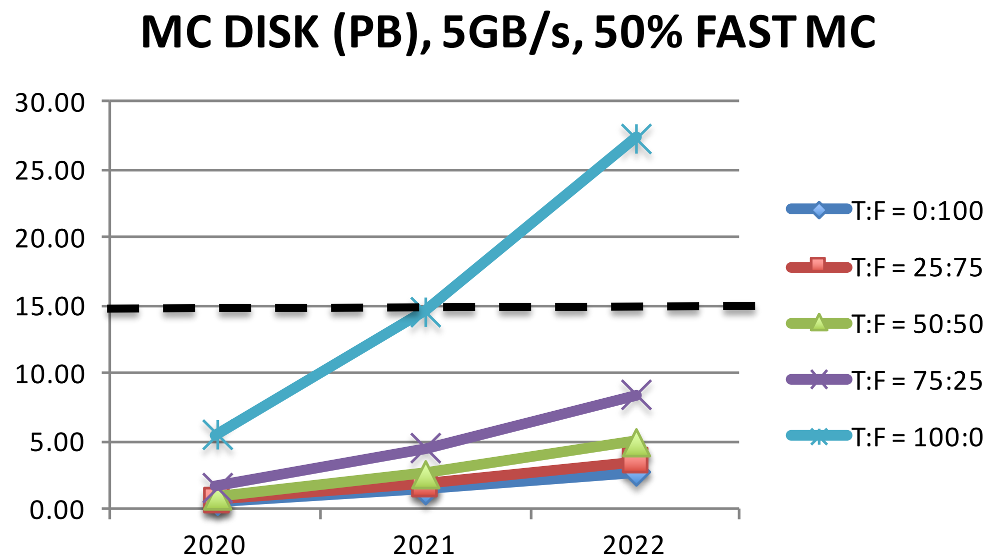


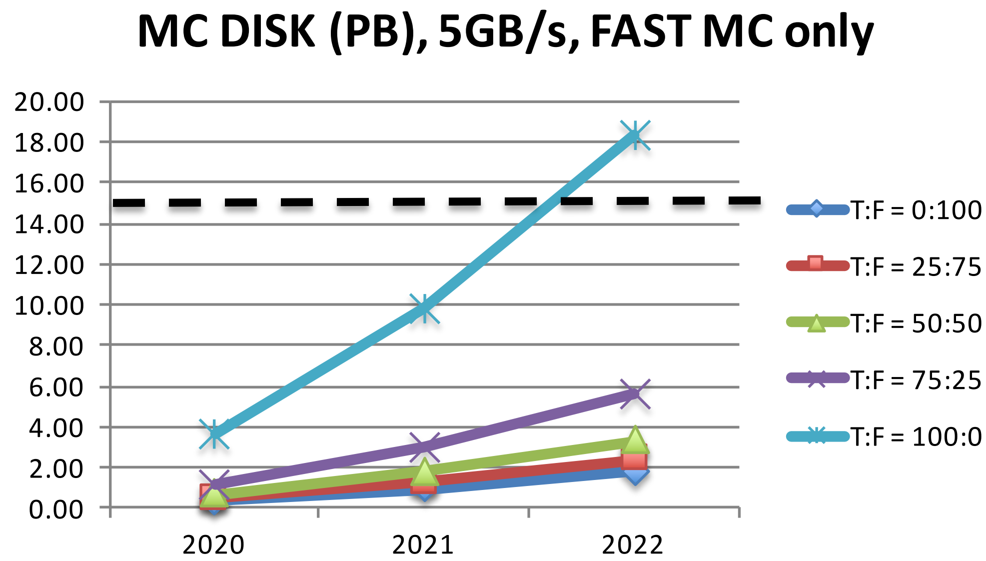


Dashed line = Expected resources at end of Run2

MC Disk (5GB/s)

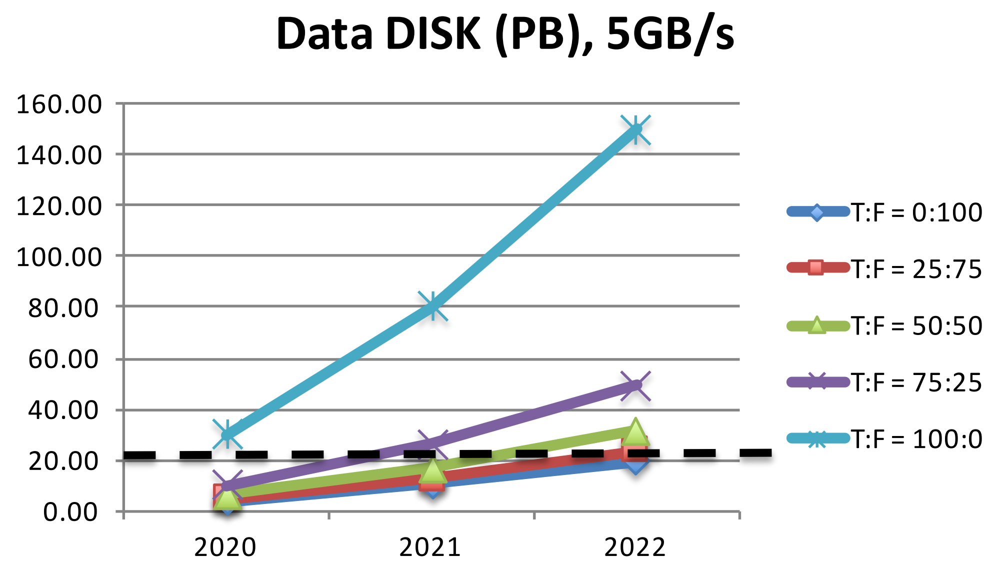


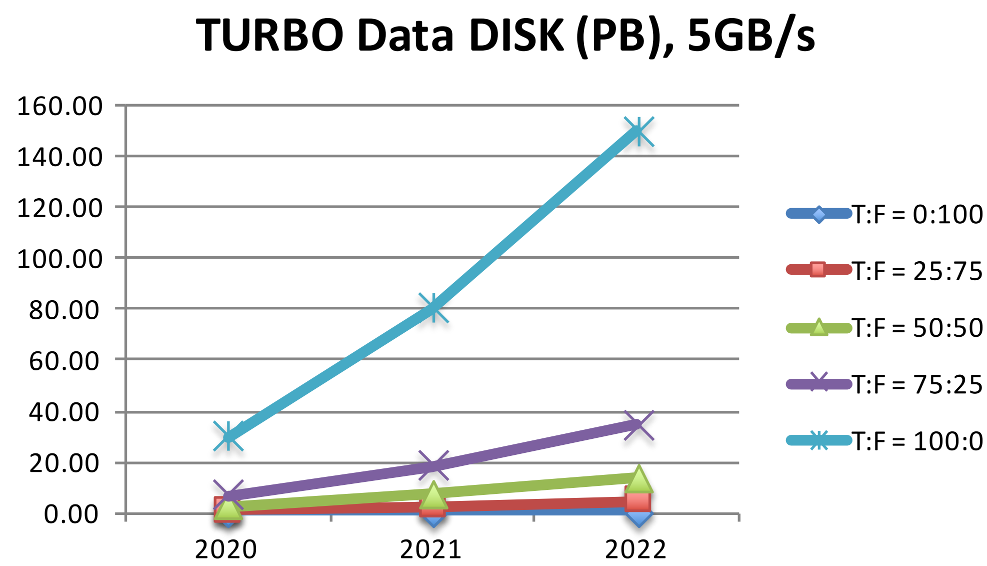


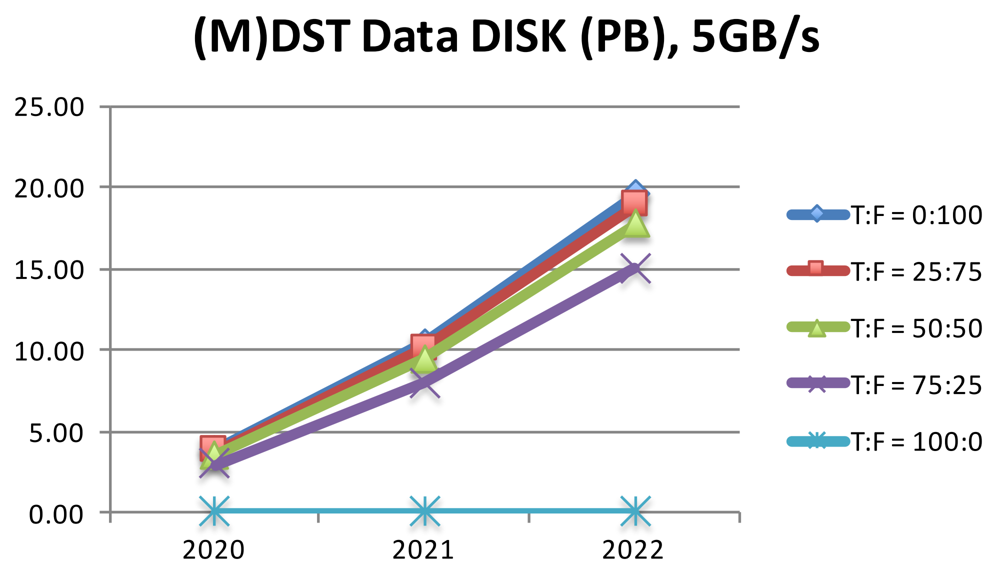


Dashed line
=
Expected resources at end of Run2

Data Disk (5GB/s)

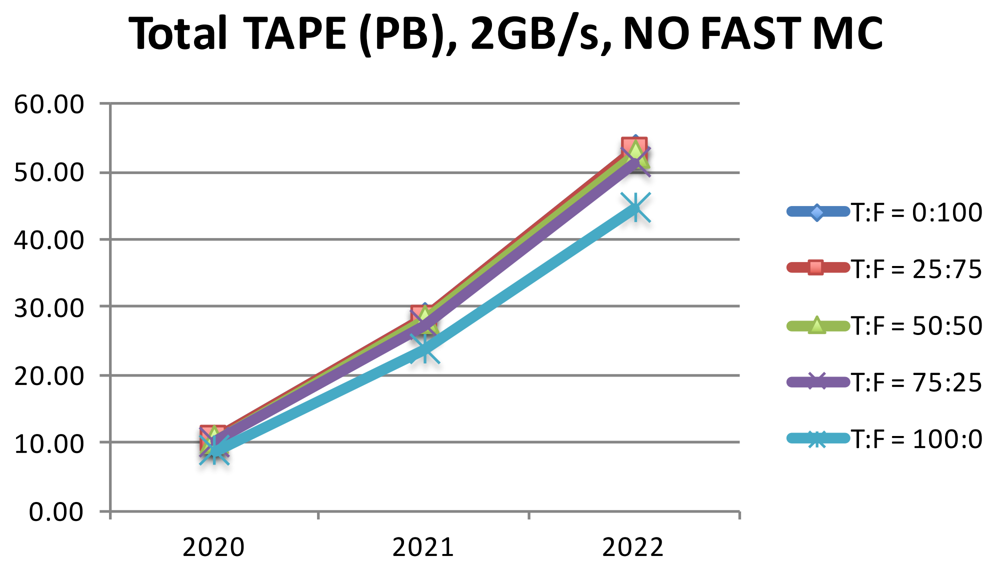


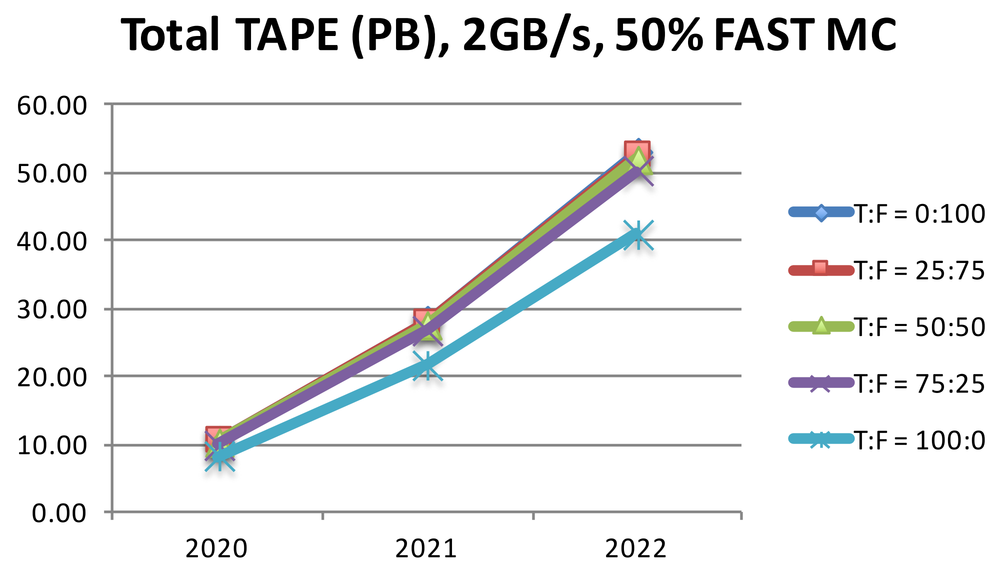


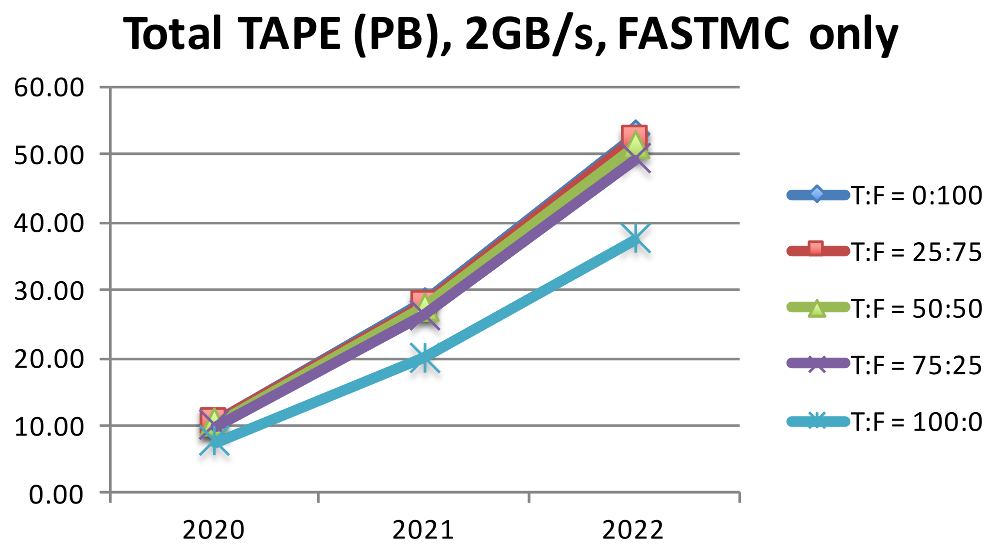


Dashed line = Expected resources at end of Run2

Tape (2GB/s)

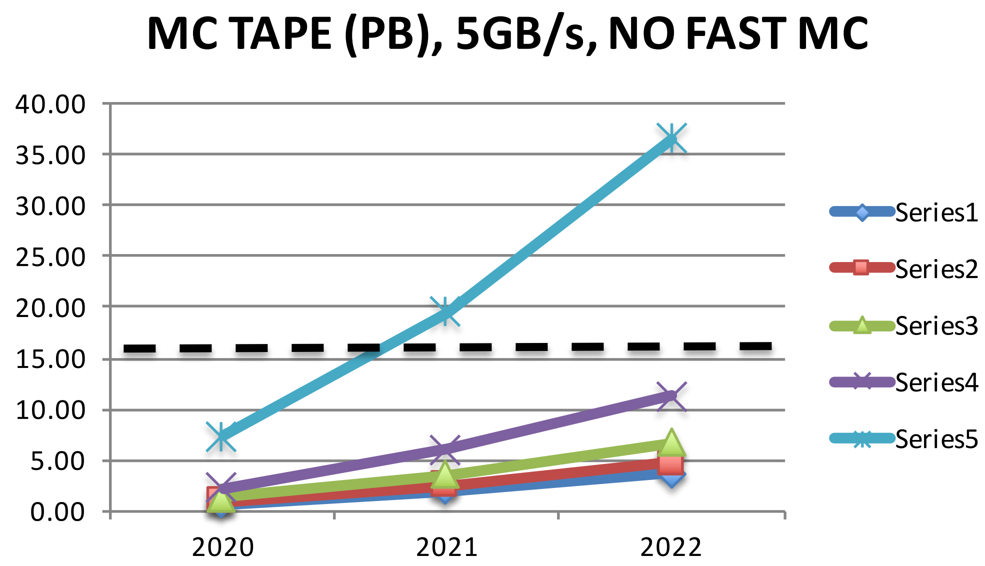


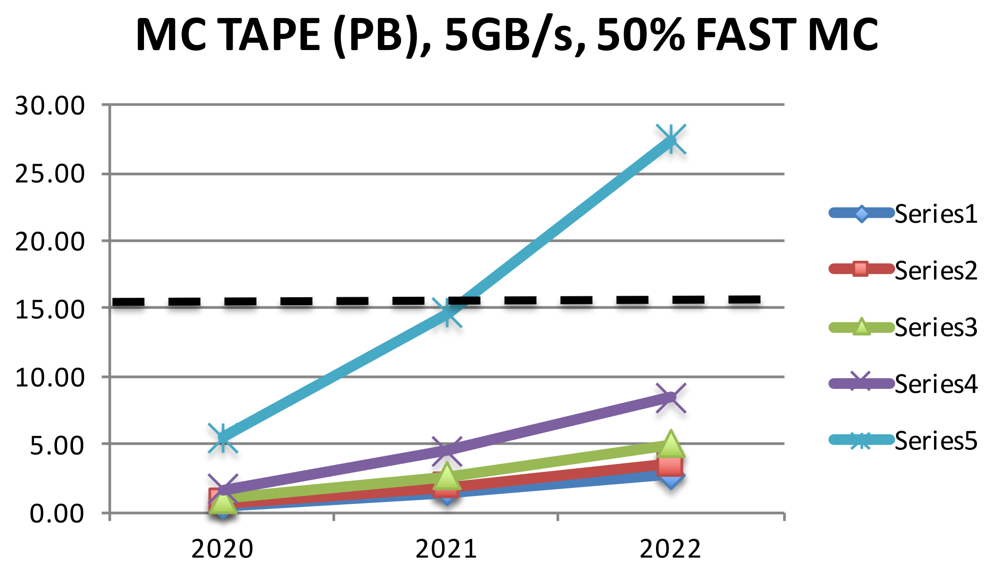


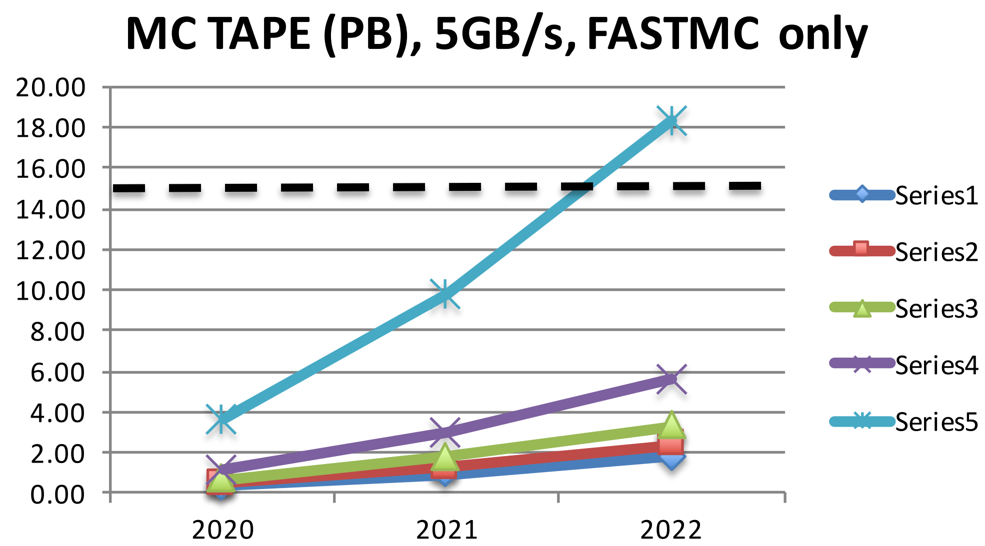


Dashed line
=
Expected resources at end of Run2

MC Tape (5GB/s)

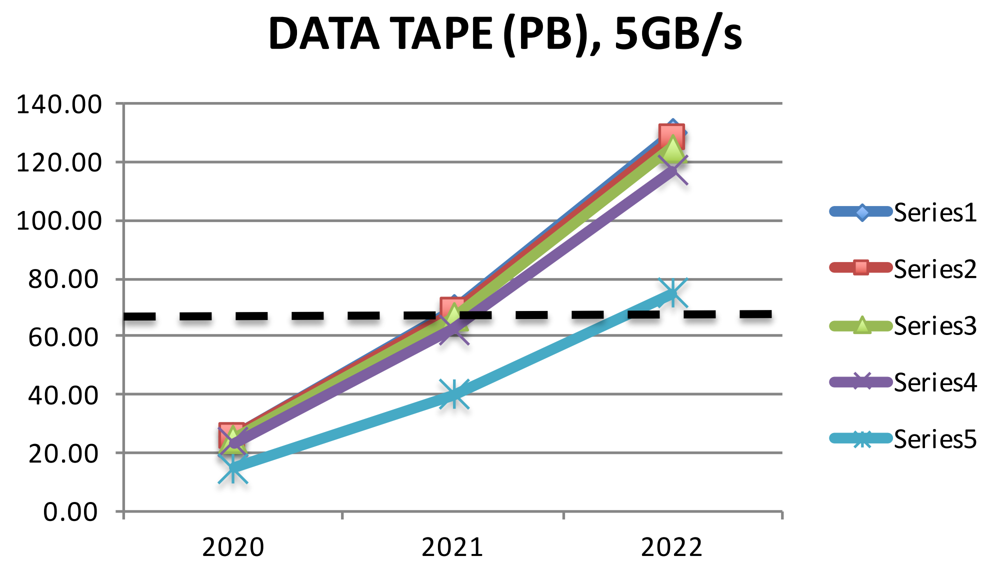


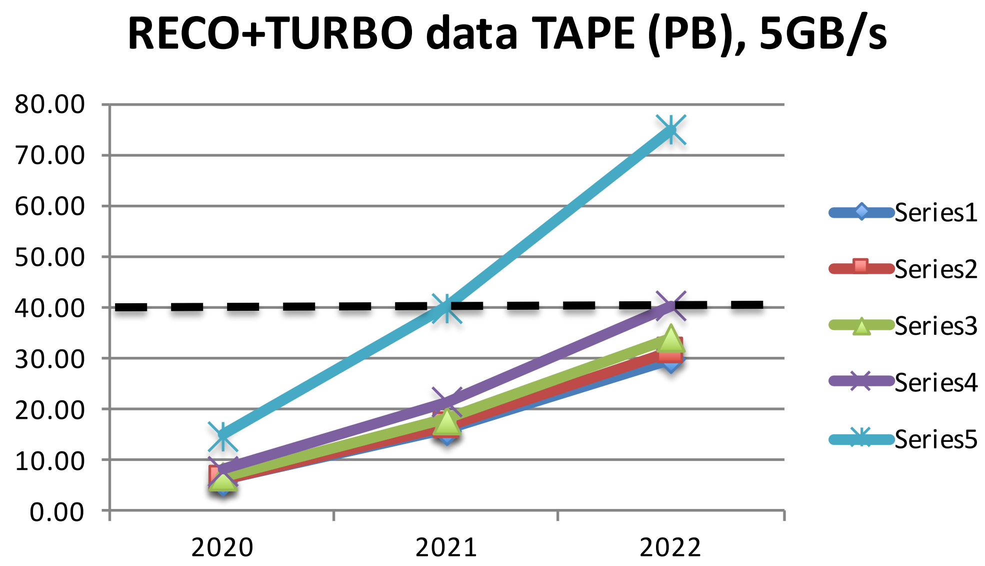


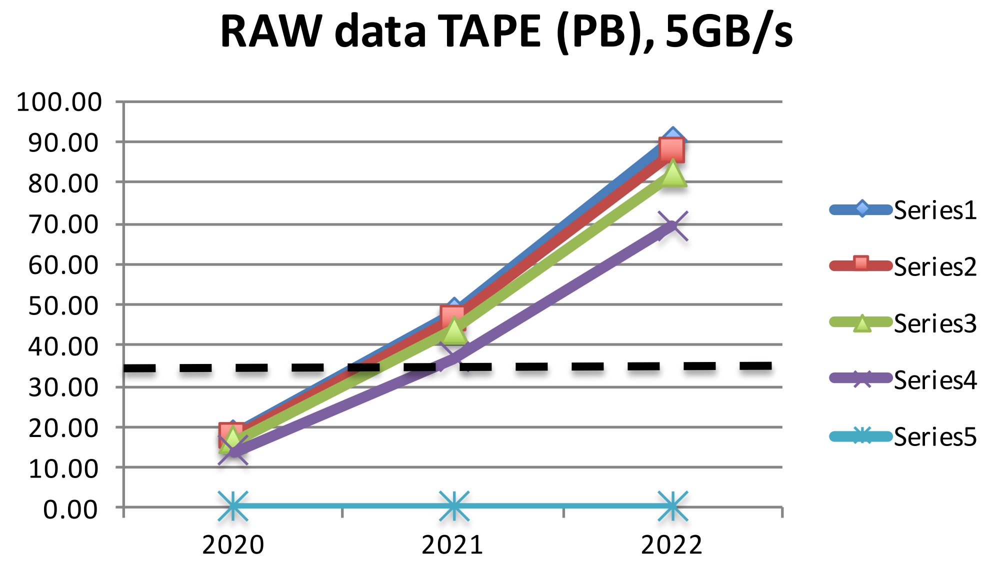


Dashed line
=
Expected resources at end of Run2

Data Tape (5GB/s)







Dashed line
=
Expected resources at end of Run2