Computing Models: Lessons Learned

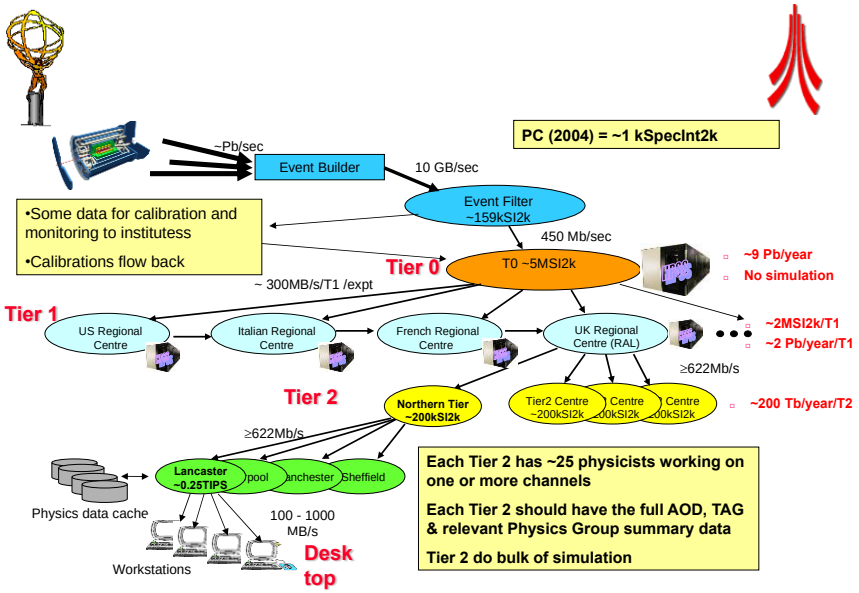
Lancaster University

- Thanks to many people, especially LHC experiment computing co-ordinators, and input from the many submissions
- Opinions are my own!

- The overall picture is an evolution from rather strict hierarchies to looser arrangements
- The overall lessons learned are
  - Plan for change
  - Aim to be agile

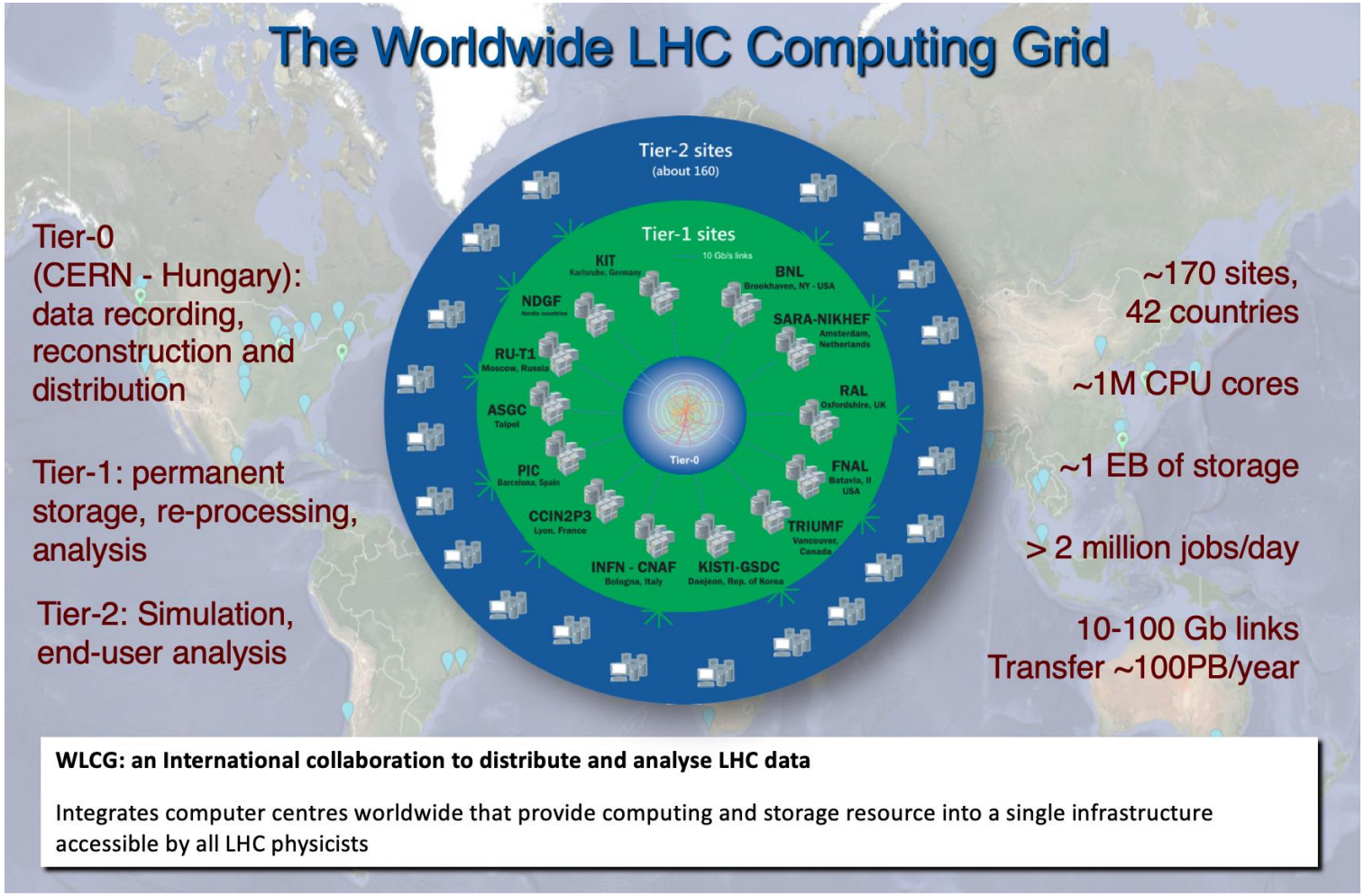
The evolving model….. From Hierarchy to a Nucleus Model

Very much started from the MONARC project hierarchical design



- Strict hierarchy of regional ‘clouds’ meant
  - Inefficient pairing of cpu with data
  - Tier 2 storage only use for secondary replicas
  - High priority tasks got stuck in small regional clouds.

The overall LHC model



The evolving model….. From Hierarchy to a Nucleus Model

- Boundaries between the Tier 2s and the Tier 1s blurring, as are regional boundaries
  - The levels of assured service remain different
  - Making a world cloud, with Tier 1 and large Tier 2 nucleus sites to aggregate data solved the hierarchy issues/improved resource usage.
  - Task outputs aggregate to a designated nucleus site

- Tier 1s retain a special custodial data role
  - This special role seems likely to remain in future models
  - The requirement for data preservation (and analysis preservation) has grown through the life of the LHC

- Tier 2 sites are differentiating: large sites with large datastores, smaller cpu-intensive sites
- Tier 2 sites can now execute most tasks

- ‘Tier 3’ computing (local to institute) now contributes to simulation load

- And ‘other’ resources are an important feature for opportunistic use– HPCs, commercial clouds, other private clouds and varying OS

The GPD experiences

Lancaster University

- Overall computing models now mature since ‘nucleus’ model after Run 1
- The main drives are robustness and efficiency – with Run 4 a big driver
  - Doubling the size of the CPU and disk in a few years has not sacrificed efficiency.
- Reprocessing becomes less frequent as you understand the detector better
- Simulation is also stable, allowing a focus on the appropriate level of detail in each aspect and an optimal use of resources.

Opportunistic Resources

Lancaster University

- There are increasing drivers to work with other communities
  - This means you access more and diverse resources
- Opportunistic computing is really important!
  - A trigger farm is a powerful computing resource – use it between runs!
  - We remain High Throughput Computing, but HPC and cloud resources are often available and can be at low or no up front cost.
- But the workflows you put on them matter
  - E.g. Simulation is good for HPCs (cpu bound, little IO)
- Match the workflow to the resource

PRACE

HELIX NEBULA THESCIENCECLOUD

Software and production systems

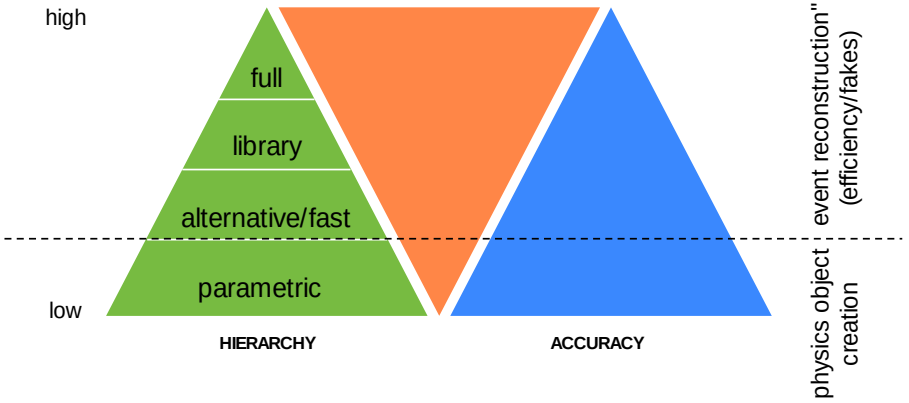
Lancaster University

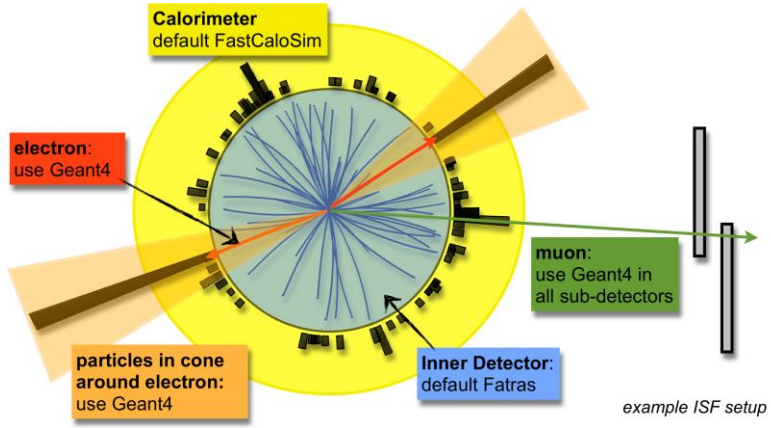
- Consider "esoteric" capabilities from the start. You will need special architectures at some point, and rewriting and redesigning is harder.
  - Examples: use of GPUs and accelerators, multithreading, multiprocesses
  - Example: Consider the use of HPCs from the start
  - Example: build in remote connection ability

- Optimize from the start. ‘Good enough’ code for a specific task can limit the total work you can do for the totality of tasks, leading to ‘not good enough’ physics outcomes.
  - Know the bottlenecks early.
  - Profile

Simulation

- Simulation is vital for LHC physics, and a big CPU driver
  - Plan it carefully, and check what you are producing
  - The analyses are now multi-year, and the simulation planning needs to be on the same timescale
  - Do enough detail for the task in hand, but no more!





- Initial analyses were hampered by the available MC statistics
  - As experiments mature, this has improved (optimization again)

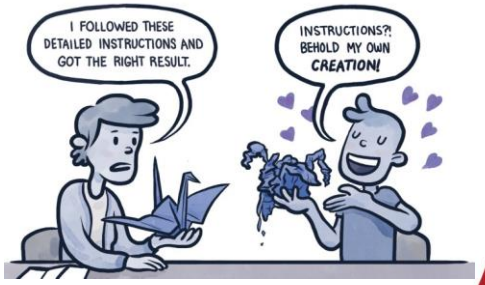
Production systems

- Building your own tools on standard deployments seems like a good idea at the start, but
  - Need to be maintained
  - Limit your use of opportunistic sites
  - Use them judiciously
  - Wrap them in containers

Intention

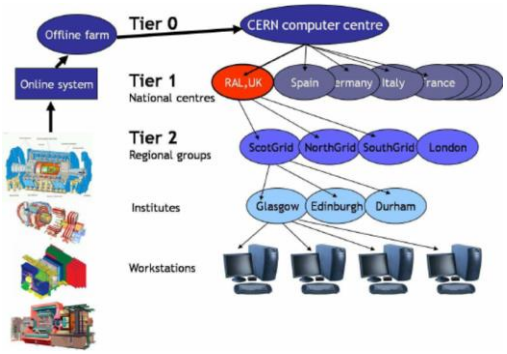
Reality

- In general, co-development is the future
- Avoid ‘NIH’ (Not Invented Here)
  - Early development and funding can stimulate parallel development
  - As things mature, finance and effort encourage standard tools



Data distribution and replication

- In the beginning ….
  - Strict regional distribution
  - Strict demarcation of Tiers
  - Fixed number of replicas
  - Little data movement



- Now moving to federated data, data lakes and trans-regional data movement
  - Enhanced flexibility, better use of CPU-heavy sites with caches
  - Concentration of data in fewer sites (eases management?)
  - Reduced disk use and better replication using popularity measures and on-demand replication

- Rucio emerging as a common large scale data management tool across HEP and into other science areas.
  - Policy driven management

Analysis tiers

- Every bit is not sacred!
  - Demands at start up that every event needed to be online for analysis in fully reconstructed mode, and even for the raw data
    - In practice, and a small subset of events are used

- Assess what data time critical – tape is still a live technology that can be used for less time critical datasets

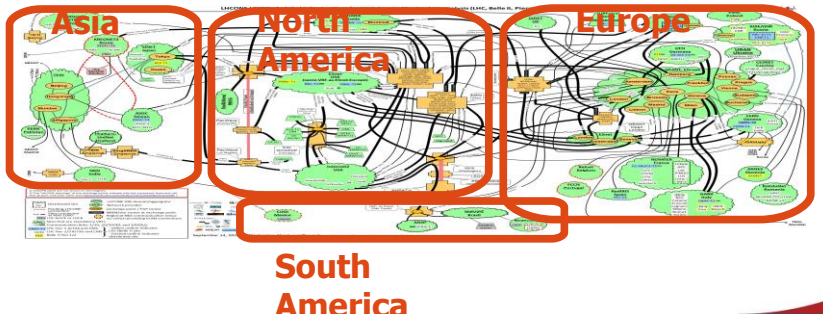
Storage – data models and replication

- Disk usage is driven by the analysis model.
  - As soon as you can, *share* formats between analysis groups
  - This improves access and reduces disk demand
- Push to smaller analysis (tiered) formats (seen at previous colliders)
  - Store only what you need for common tasks, in sufficient precision
  - Compact shared root-able formats allow for more analysis to be done
    - CMS NanoAOD - ultra-compact
    - ATLAS xAOD – thinning, slimming and strimming, cut *all* the fat


- Revise, revise revise. Excellent review, with great recommendations – but reality will diverge from expectation so keep sanity checking
- Plan for data preservation from the start

Networking

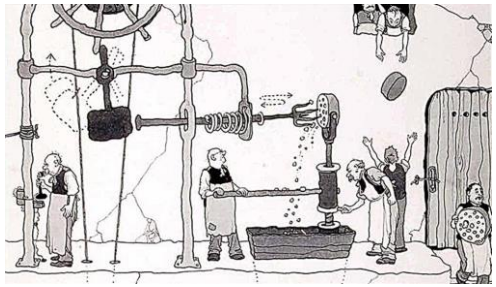
- Networking really matters. Much of the evolution of the models (and the tools underpinning them, Xrootd federations, data lakes, premixing libraries) rely on good networking
  - But recognise local difference
  - In many regions, there is no up front cost to high bandwidth networking or ‘special networks’
  - In others, it is part of the total computing budget and so tensioned against disk/cpu etc
- Design your model to accommodate the politics/finance of the regions



Plan for Diminishing Human Resource

- The effort for computing (and software) has never been great, but seems to be an early sacrifice when cuts come
- There seems an external belief that once running, the required effort becomes smaller
  - This might be true in a static environment
  - But the environment and experiment ambitions always change and grow!

- Invest as much as you can on automatic systems / good monitoring / documentation from first moment



- Career paths for the required experts are a problem
  - Often undervalued in academic physics positions
  - Often not highly regarded as ‘academic’ computer scientists
- Plan a career path for ‘research software engineers’

Development vrs more capacity

- There are trade-offs between hardware and development (both development of the payload software and of the production system)

- Both hardware and development have costs, but often from different lines; try to view them holistically
  - Example: The cost of RAM is a the main driver for the GPD work on multi threaded frameworks.
  - Example: Work on a simulation framework with the ability to select full/fast/faster simulation modes at a granular level can reduce the CPU costs
  - Example: Work on effective reduction in analysis formats can lead to large savings in storage costs.

Software Delivery

- Software delivery has evolved beyond recognition from package mangers and shared software areas
- We started by running ‘installation jobs’ running tools largely developed ‘in house’ in the experiments

- We have now moved to a content delivery approach
- CERN VM File System, CVMFS – widely adopted common tool
  - Aggressive caching and reduction of latency
- Also used for (small) datasets, conditions data etc
- Seen as class leading, being considered for use by other communities – biomedical, space & earth science, industry etc

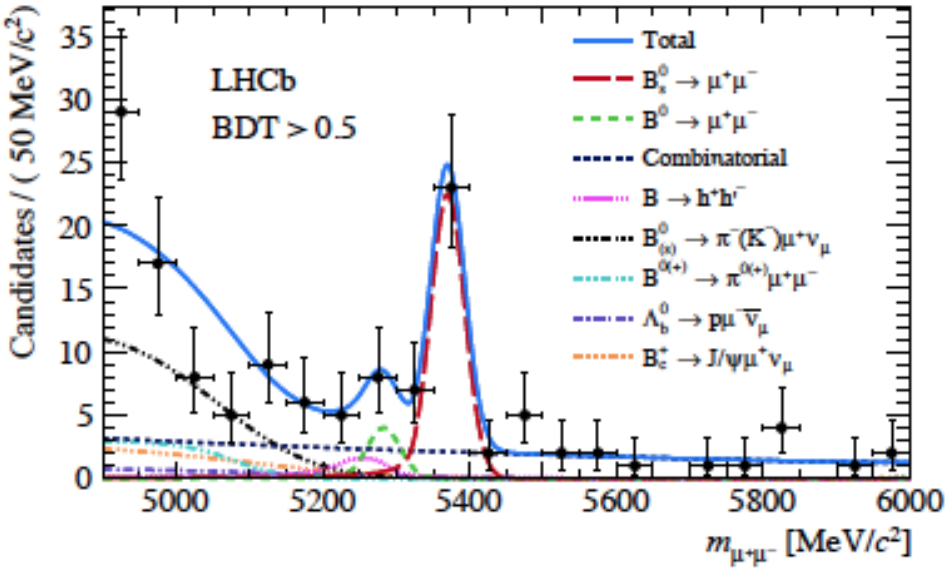
An Interesting Evolution in the Offline/Online hierarchy - LHCb

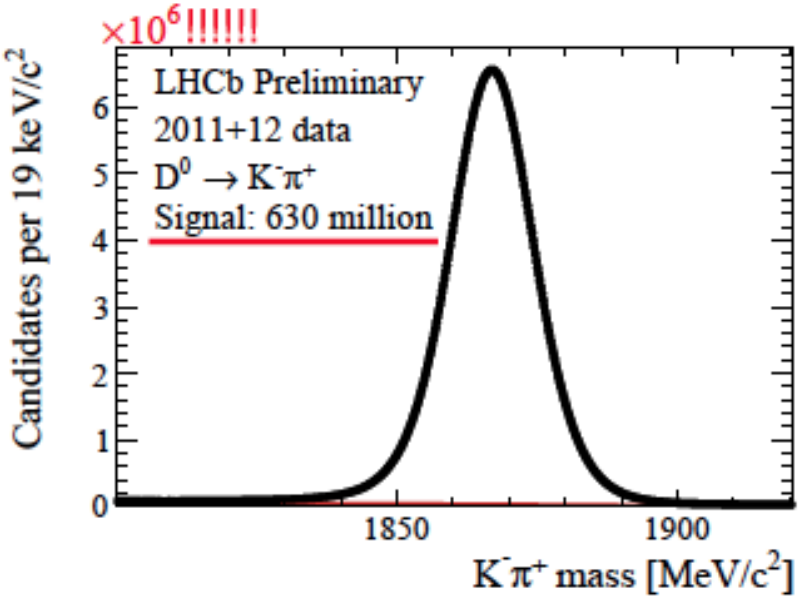
- LHCb began with a now traditional design of levels of online trigger, but large volumes written for offline processing and reduction
- It is now on an upgrade path to move more an more ‘offline’ activity into the online (including reconstruction, but even analysis)
- Fully software trigger
  - The gains are in the good data volume you can store and process for analysis
  - By doing this as an evolution, they have mitigated the risks to the physics
  - The quality of the triggering is much improved

LHCb Run 2 Trigger

Lancaster University

- The LHCb trigger has to cover extremes of data taking:



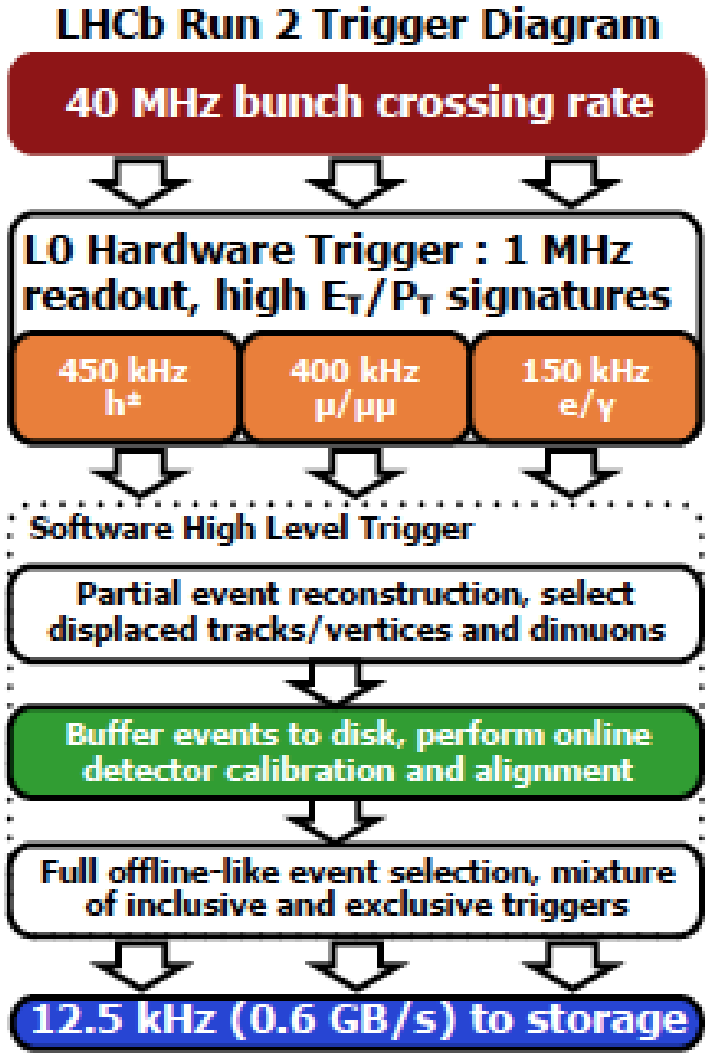


- High efficiency to collect rare decays like $B_s^0 \to \mu\mu$[1]
- High purity for enormous charm signals like $D^0 \to K\pi$[2]
- Must be flexible to operate in both extremes simultaneously: After readout, HLT has access to 100% of event in software

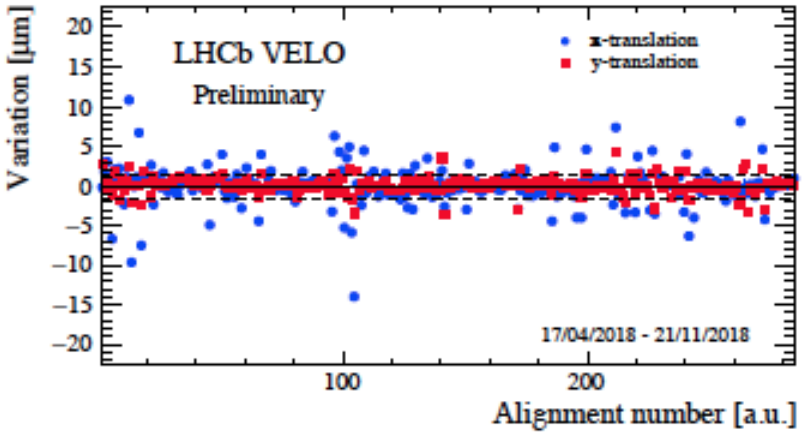
[1] Phys. Rev. Lett. 118, 191801 (2017)
[2] LHCb-CONF-2016-005

LHCb Run 2 Trigger: Trialing Real Time Analysis

Lancaster University



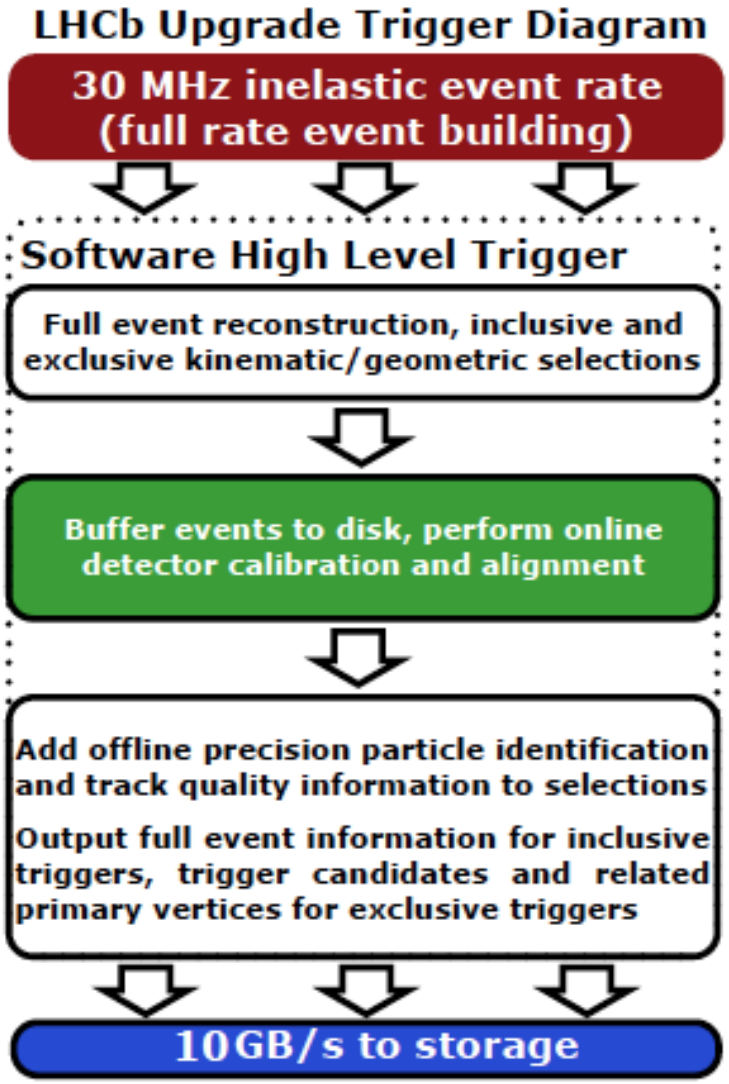
- Enabled by online calibration & alignment



- e.g. VELO alignment performed online in 7mins in Run2
- Run full offline reconstruction online
  - No loss of physics
  - Required **lots** of software optimization

LHCb Upgrade I – Fully Software Trigger

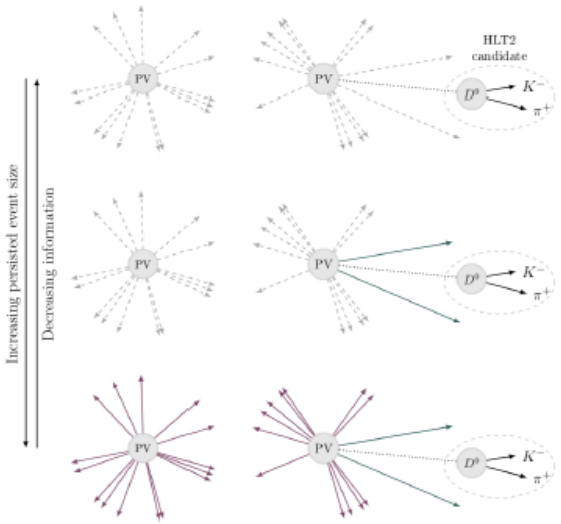
Lancaster University



- No hardware – LHC collision rate directly into software trigger
- Directly using the output of the trigger will be the main analysis model in Run 3
  - No re-reconstruction for these events
- ***enormously expand the physics programme*** in areas where the signal is too abundant to be analyzed saving all the raw detector data for each bunch crossing

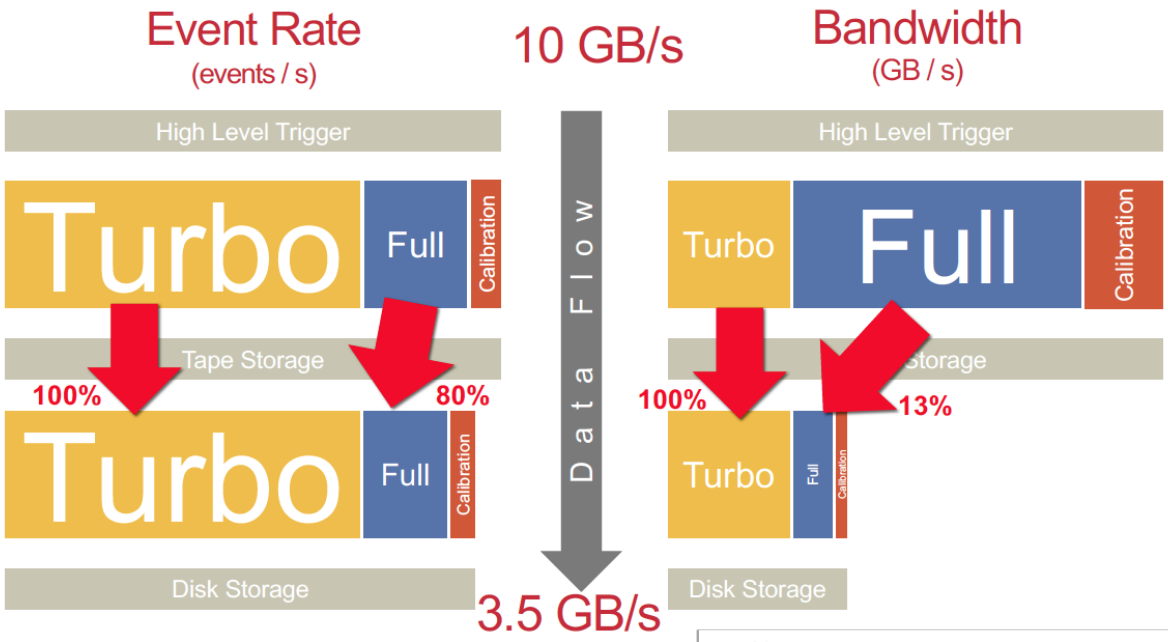
Real Time Analysis - Turbo

- Online Align and Calib means…
- Optimal quality reconstruction online in trigger
  - No need for re-reconstrrction
  - No need to keep raw data
- Benefits:
  - Expansion of physics programme
  - Large reduction in computing resources (raw data 200kB, triggered objects 15kB)
- Risks:
  - Reprocessing not possible in case of errors



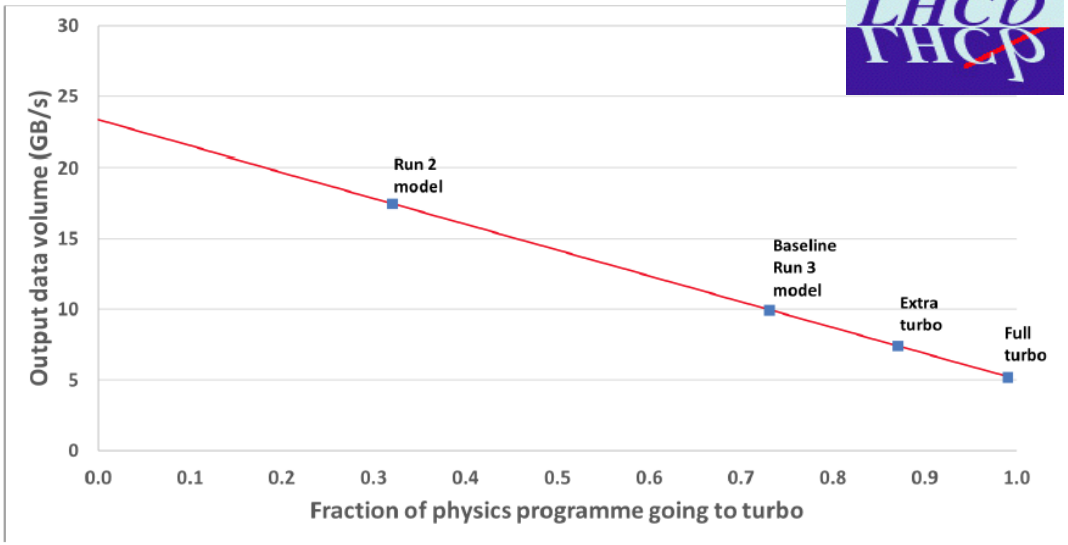
- Selective persistence – per analysis choice
  - Only signal decay tracks….
  - those in cone around...
  - those from same PV....
  - All tracks in event….
  - All ECAL clusters....

Real time Analysis - Computing Resources



- Run 2 has paved way for this model in Upgrade I
- Efficient use of computing resources
- Focus on bandwidth not event rate
- Minimise expensive disk resource

- Turbo extensively used in Run 2
- >70% of events in Upgrade I will use Turbo



ALICE in Run 3

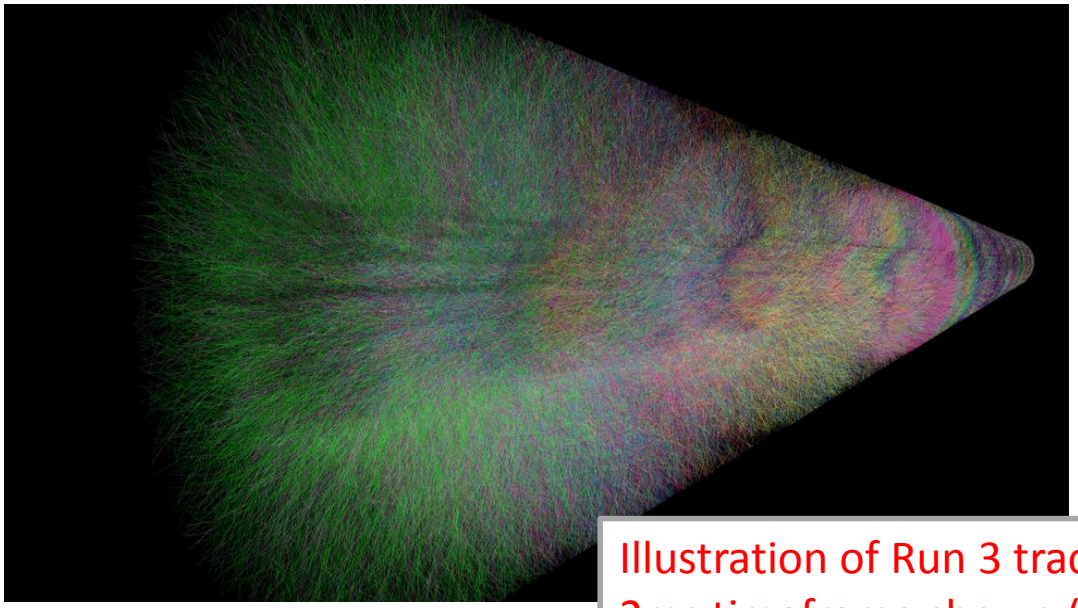
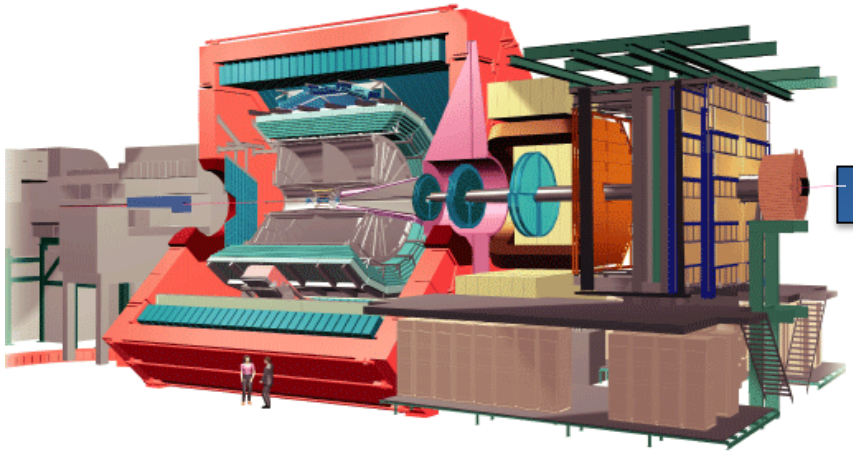


Illustration of Run 3 tracking problem:
2ms timeframe shown (10% of total)

- ALICE is upgrading its detectors in Run 3 aiming to handle x100 increase in PbPb readout rate compared to Run 1&2.

- Moving to continuous readout.
- One data chunk (timeframe) will contain up to 1000 PbPb collisions seen during 20ms in ALICE TPC.
- No rejection at trigger level possible, all events will be recorded.
- Requires online reconstruction to achieve x20 data reduction.

Online data compression



3.5 TB/s

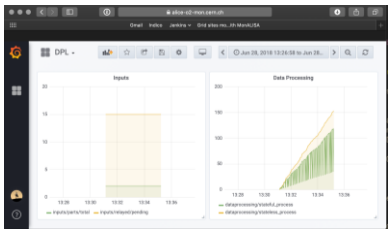
O2 Facility

90 GB/s

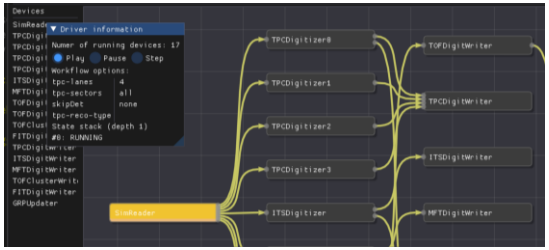
Storage

- O2 Computing Facility at P2
  - 15000 CPU cores
  - 1500 GPUs
  - 60 PB disk buffer
  - Synchronous (online) reconstruction and data compression
  - Asynchronous (offline) calibration and reconstruction

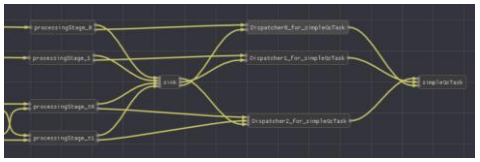
Data Processing Layer (DPL)



O2 Monitoring and InfoLogger integration

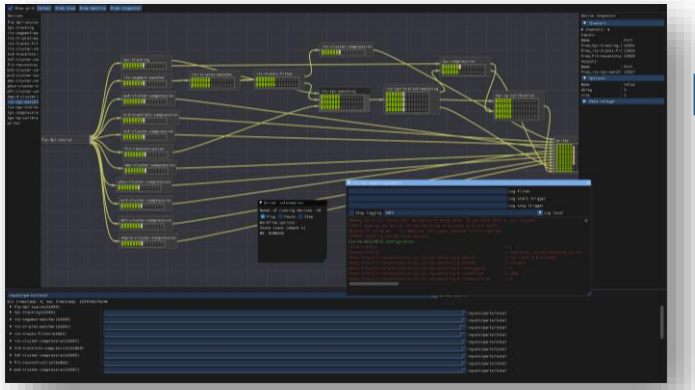


Digitization example using DPL



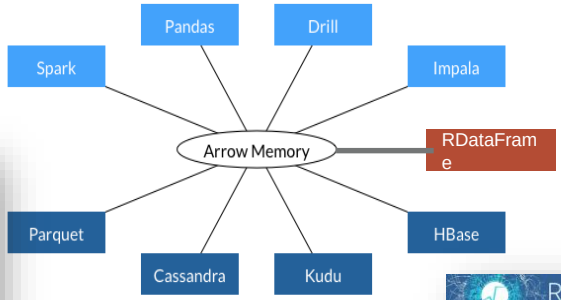
DataSampling example using DPL

Multi process, concurrent, message driven software framework

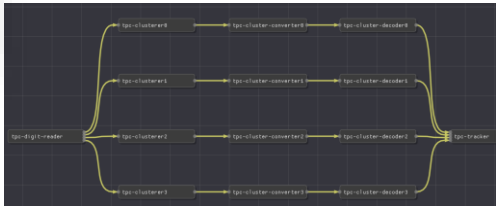


Based on ALFA, common data transport and messaging components developed by ALICE and FAIR/GSI

Allows for gradual optimization the critical parts such as using GPUs for synchronous reconstruction

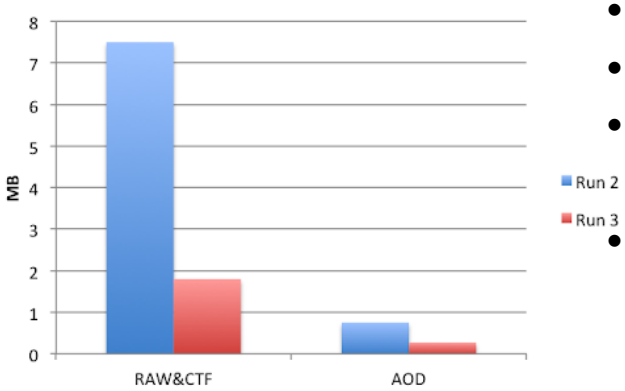


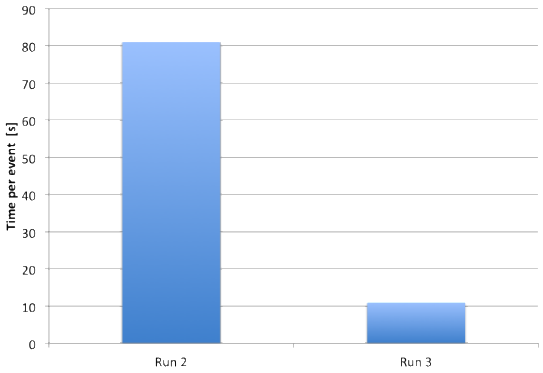
Apache Arrow integration for ROOT allows simplifeied data model



TPC reconstruction example using DPL

Run 3 expectations





- The average raw event is expected to be 4x smaller.
- AOD size per event will be reduced by factor 2.5.
- Data formats must be optimized for reading
  - Prefer simple data model for fast serialization.
- Use of hardware accelerators (GPUs) is essential to make O2 Facility affordable
  - Even more significant is the gain from re-thinking the reconstruction approach needed to make the code effectively run on GPU which made it run faster on CPU as well.
  - Performance of the reconstruction on CPU is expected to improve by factor 8 and additional factor 40 when running on GPU.
- Multi-process message driven framework opens the door for the use of HPC facilities in particular for simulation.

Conclusions

Lancaster University

The LHC has provided many lessons

- Computing solutions did not exist until we created them
- Many lessons learned

- The code and systems are produced by academics and so they have a tendency to resist standardisation
- But the realities of computing at large scale mean common tools are needed for cost and efficiency

- At the very least, learn from the mistakes!

- Timescales are long – recognize this from the start

- Plan to be agile – technologies will change

- Plan for less (hardware, effort)!