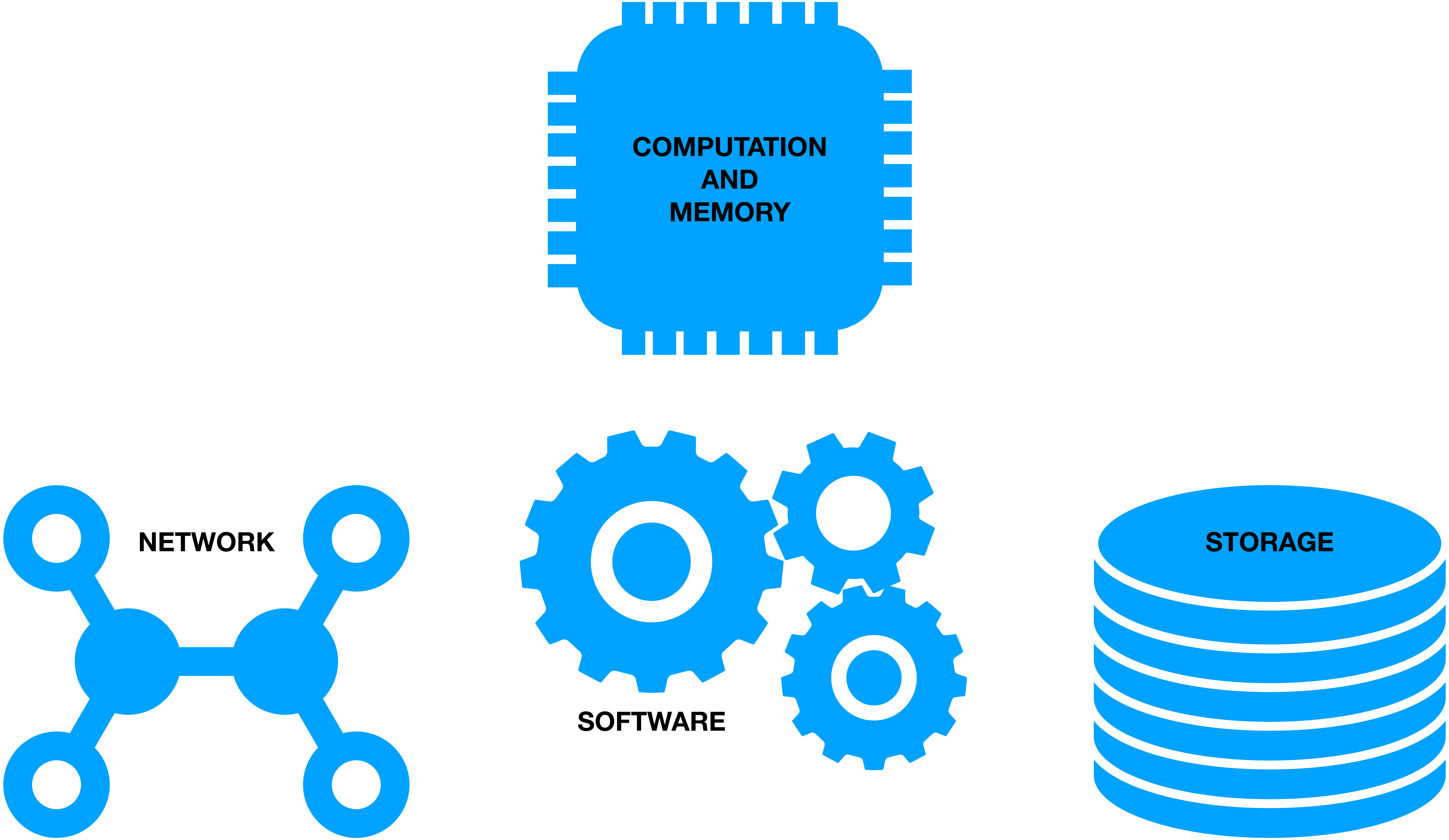
UiO : University of Oslo

Computing and Data Handling: Summary @ ICHEP 2020

James Catmore, University of Oslo
Computing Coordinator for the ATLAS Collaboration



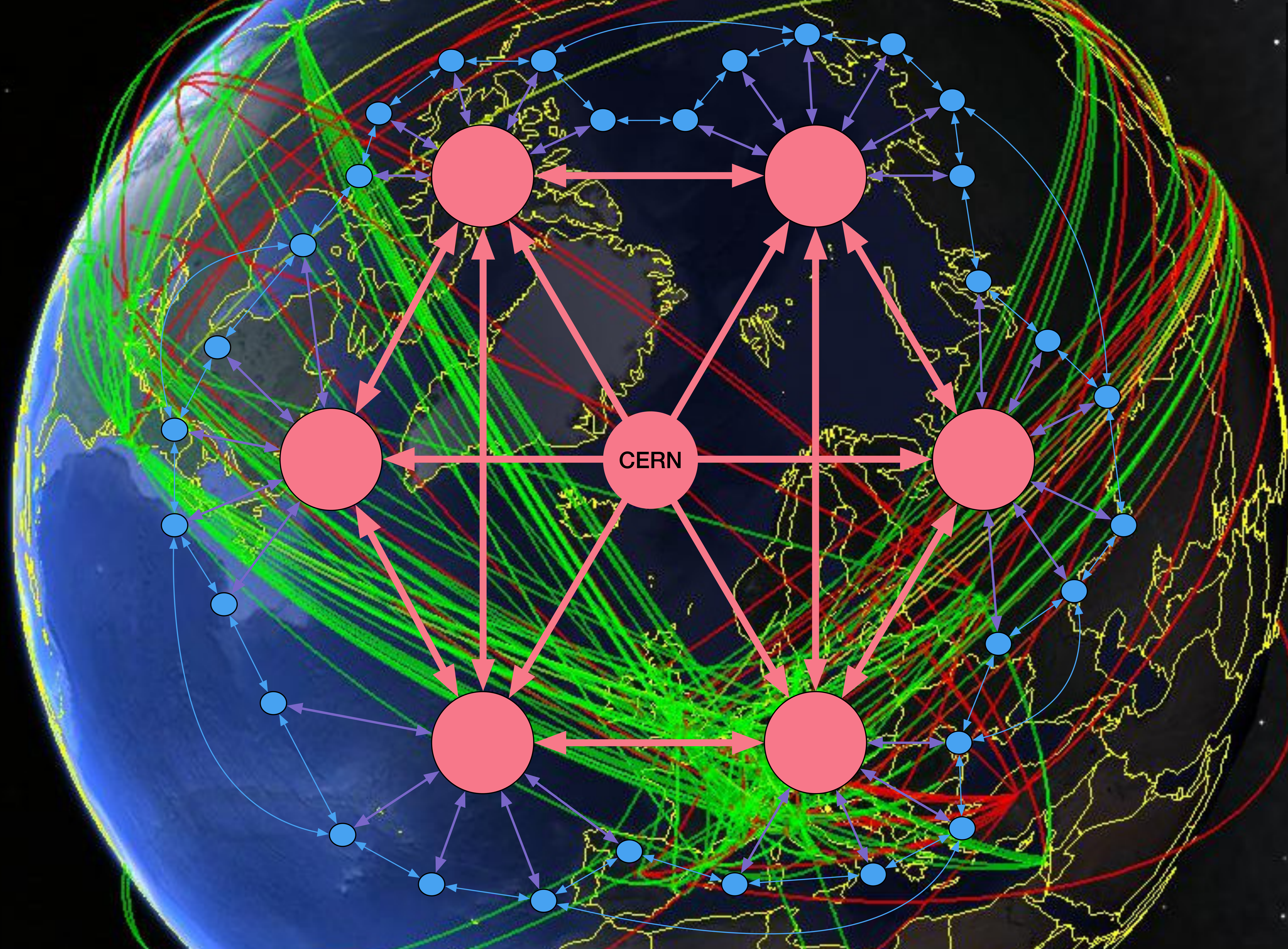
COMPUTATION
AND
MEMORY

NETWORK

SOFTWARE

STORAGE

WLCG
Worldwide LHC Computing Grid

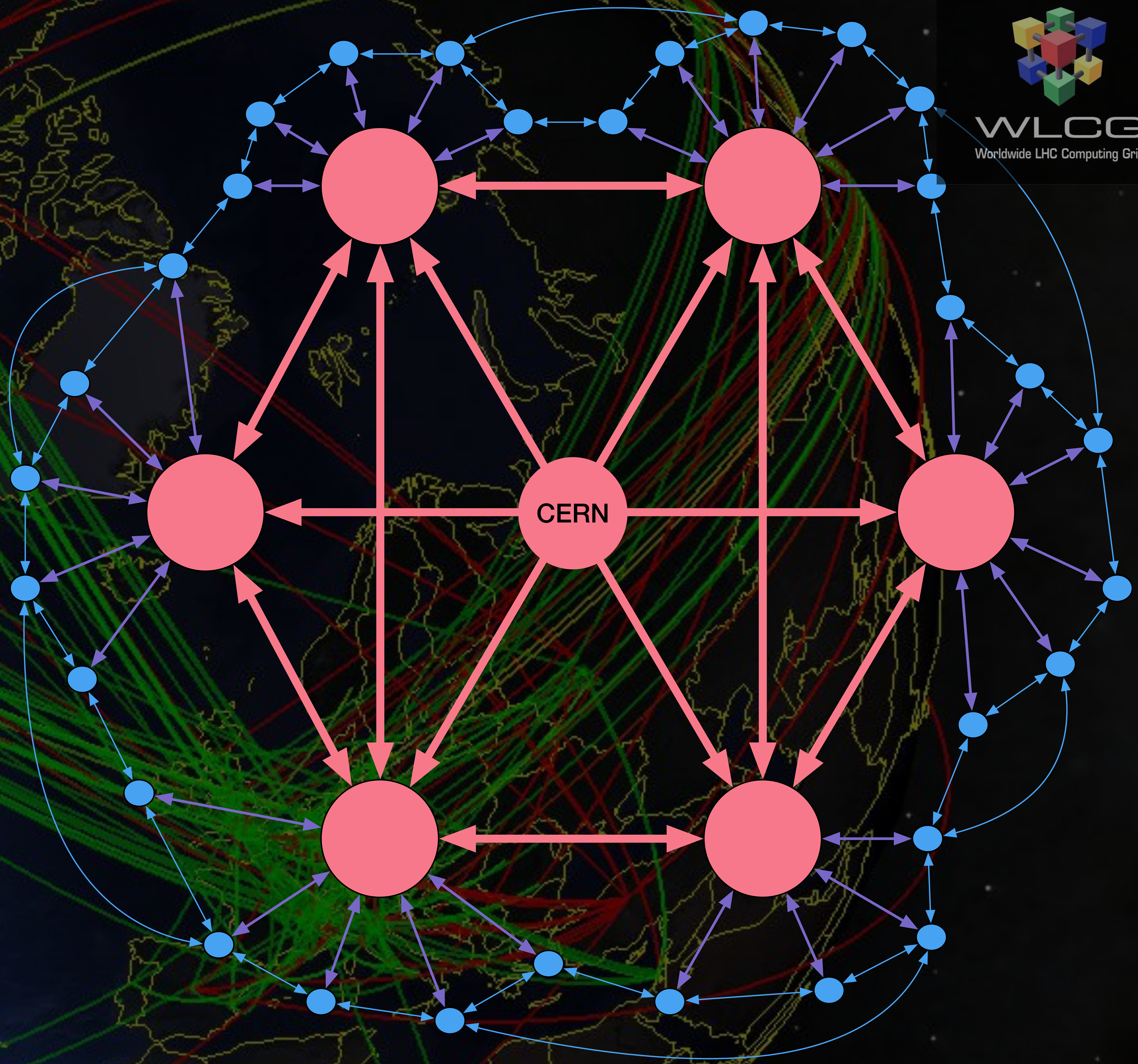


WLCG
Worldwide LHC Computing Grid

50 GB/second
50 million files/week

> 6 billion HS06-hours/month

> 1 exabyte on disk and tape



WLCG
Worldwide LHC Computing Grid

Compute: mostly Intel or AMD CPUs with x86 instruction sets. Usually multi-core. Some GPUs now available at the main Grid sites and (particularly) at HPC centres.

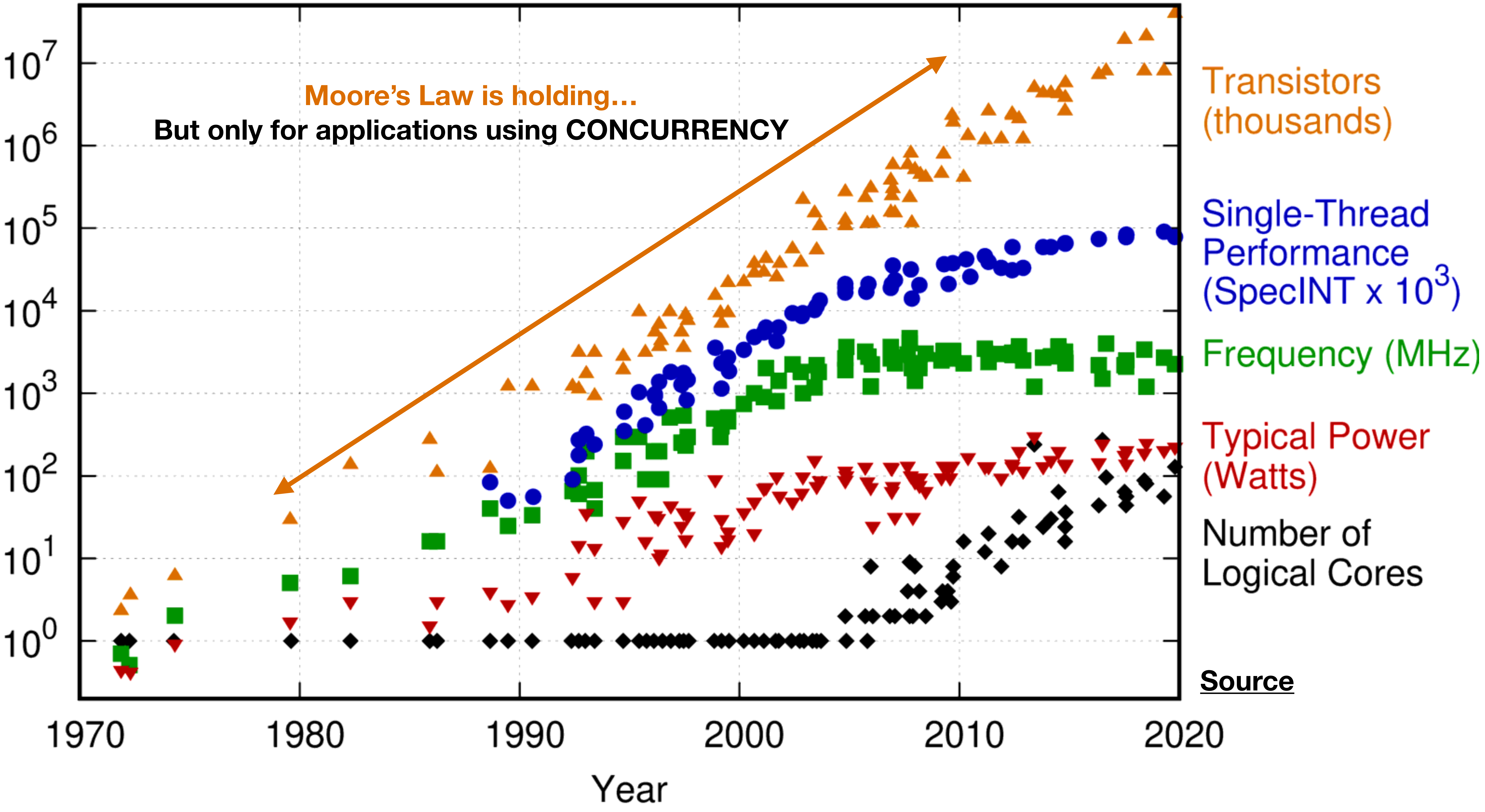
Storage: mixture of tape (for long term data archival) and disk (for fast and regular data access). Very little solid state technology in use.

Network: CERN is connected to each of the major grid sites around the world on a dedicated, private, high-bandwidth network called the LHC Optical Private Network (LHCOPN). Links can sustain between 10-100 gigabits/second, leading to average data movement of 50 GB/second

Software: complex patchwork. Most experiments have dedicated frameworks written in C++ and configured in Python. Rely on many external packages from within and outside the field. Many millions of lines of code. Generally written for x86, originally single threaded but increasingly multi-threaded.

Analysis software as varied as the user community, but strong movement towards the Python ecosystem and particularly notebooks.

48 Years of Microprocessor Trend Data



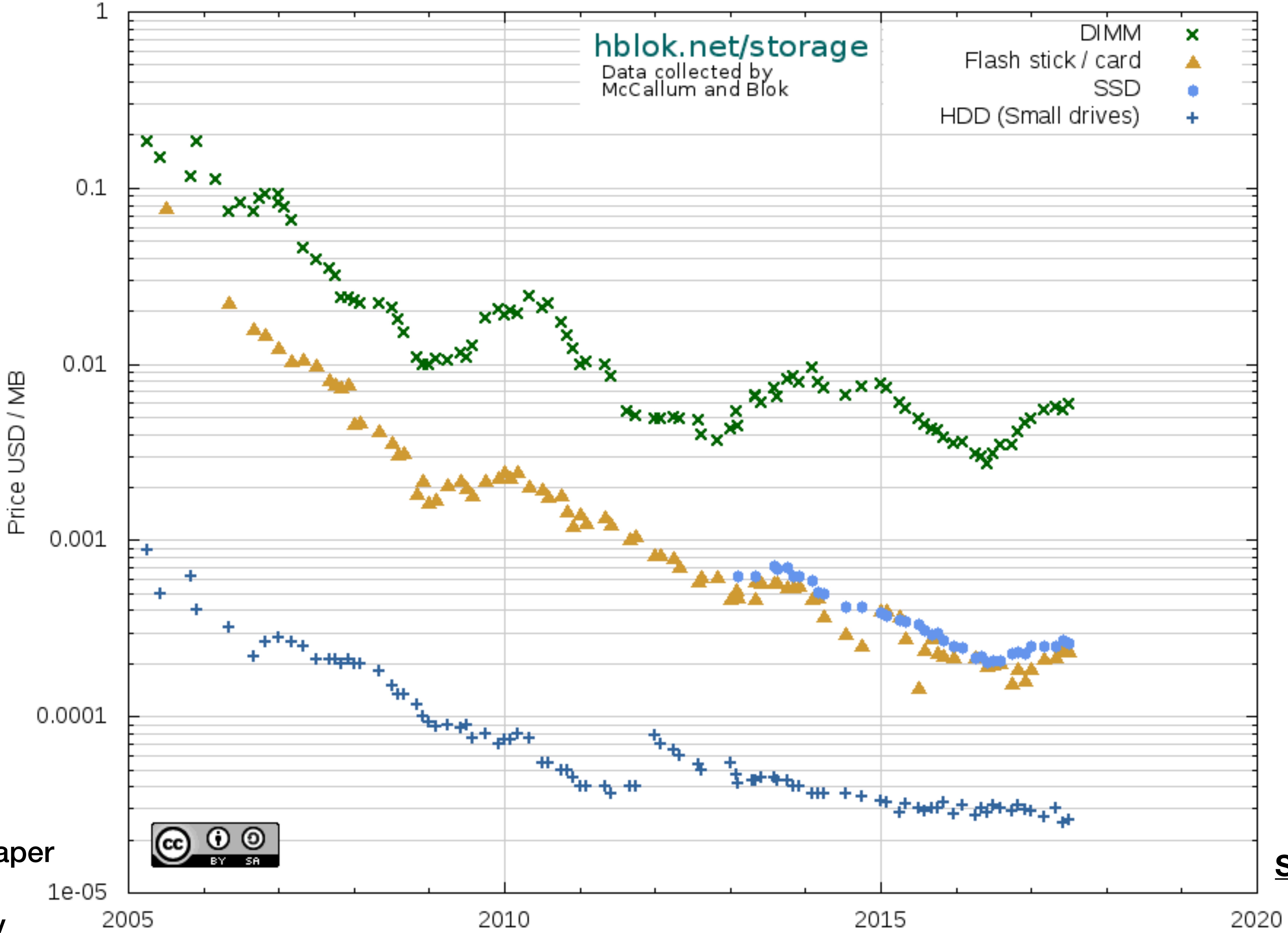
Moore's Law is holding…

But only for applications using CONCURRENCY

Source

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2019 by K. Rupp

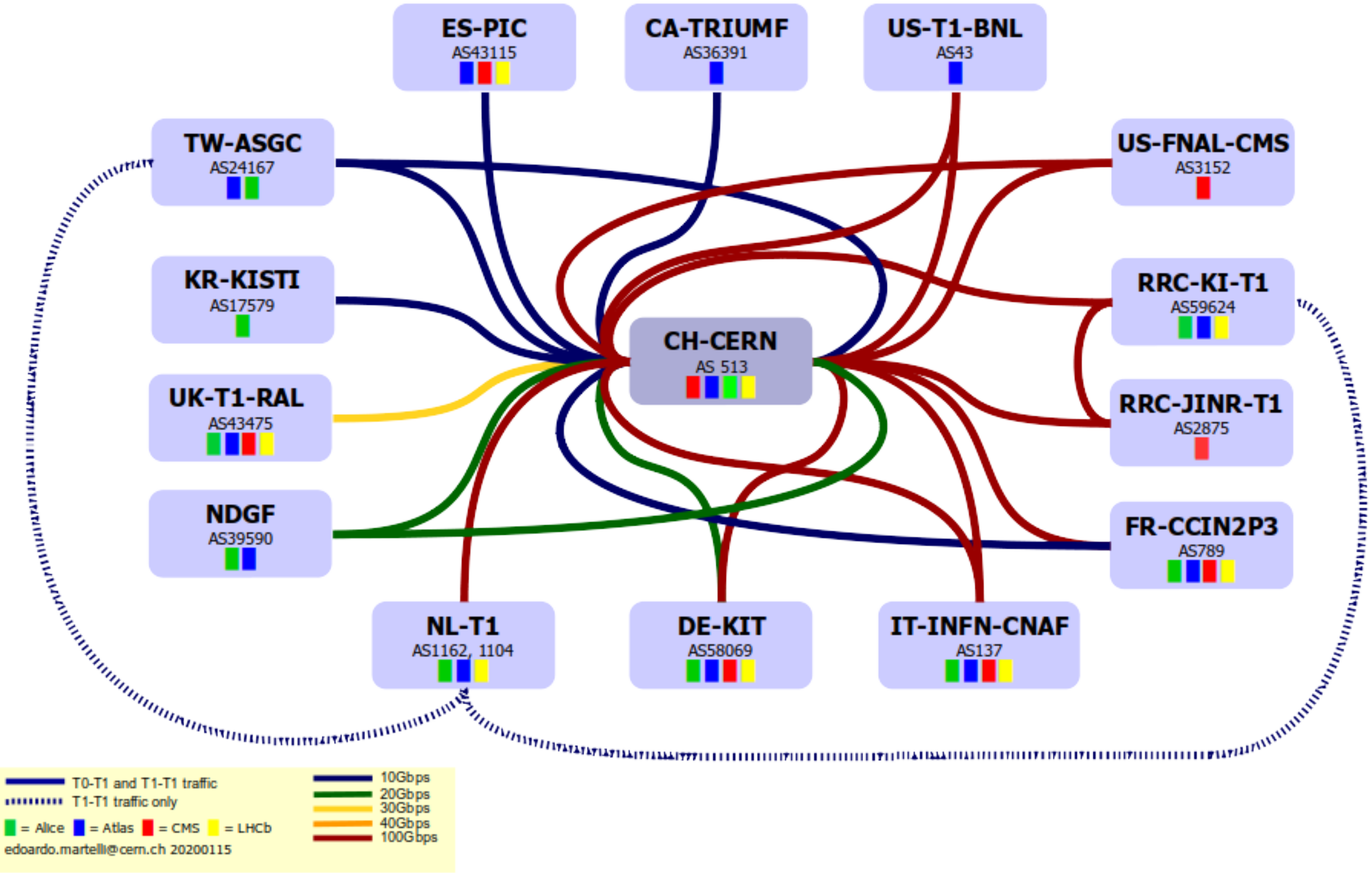
Historical Cost of Computer Memory and Storage



Tape significantly cheaper than disk…
but near-monopoly

Source

LHCOPN



ES-PIC
AS43115
🟦🟥🟨

CA-TRIUMF
AS36391
🟦

US-T1-BNL
AS43
🟦

TW-ASGC
AS24167
🟦🟩

US-FNAL-CMS
AS3152
🟥

KR-KISTI
AS17579
🟩

RRC-KI-T1
AS59624
🟩🟦🟨

CH-CERN
AS 513
🟥🟦🟩🟨

UK-T1-RAL
AS43475
🟩🟦🟥🟨

RRC-JINR-T1
AS2875
🟥

NDGF
AS39590
🟩🟦

FR-CCIN2P3
AS789
🟩🟦🟥🟨

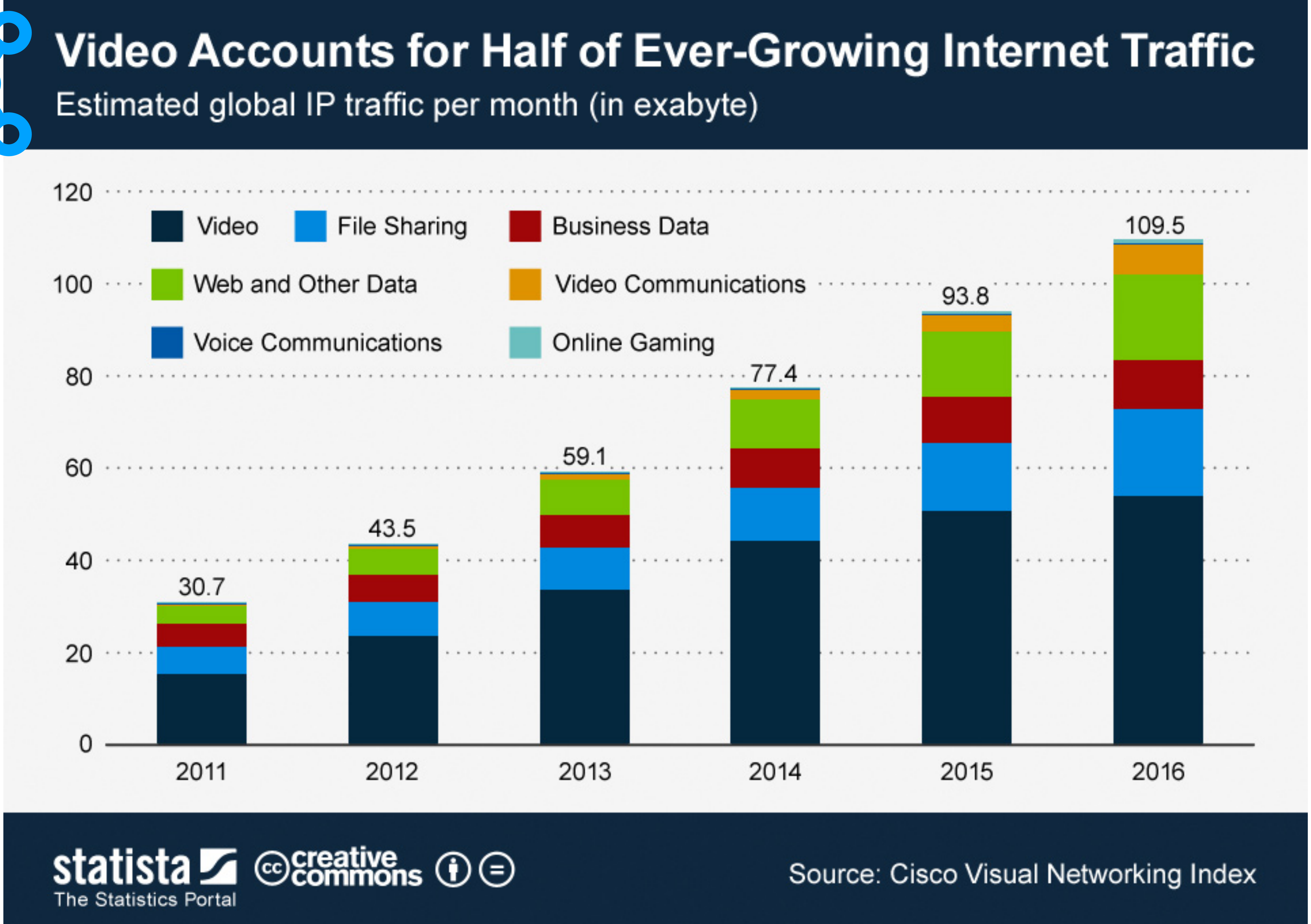
NL-T1
AS1162, 1104
🟩🟦🟨

DE-KIT
AS58069
🟩🟦🟥🟨

IT-INFN-CNAF
AS137
🟩🟦🟥🟨

T0-T1 and T1-T1 traffic
T1-T1 traffic only
🟩 = Alice 🟦 = Atlas 🟥 = CMS 🟨 = LHCb
edoardo.martelli@cern.ch 20200115

10Gbps
20Gbps
30Gbps
40Gbps
100Gbps



Video Accounts for Half of Ever-Growing Internet Traffic

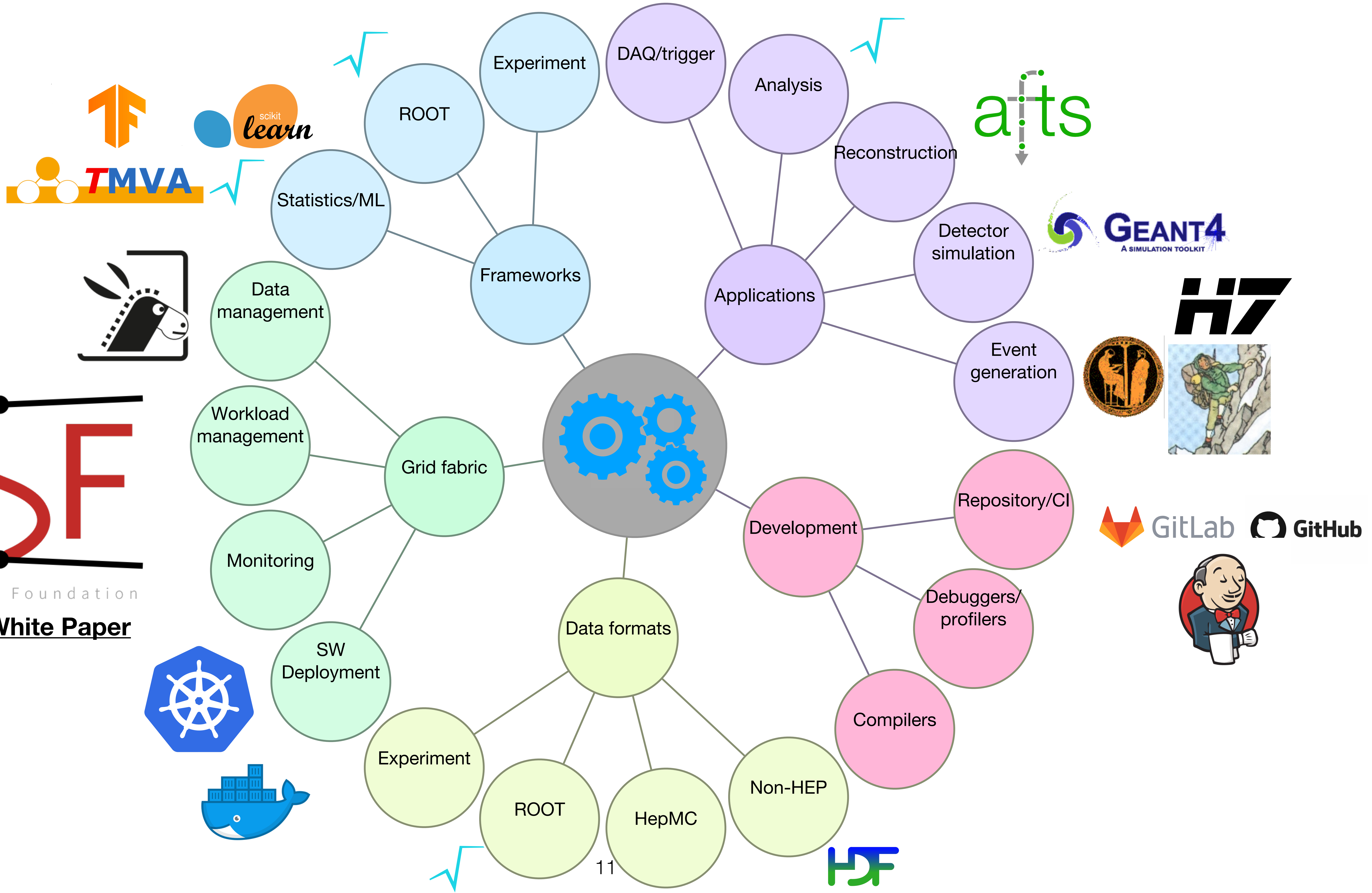
Estimated global IP traffic per month (in exabyte)

Source: Cisco Visual Networking Index

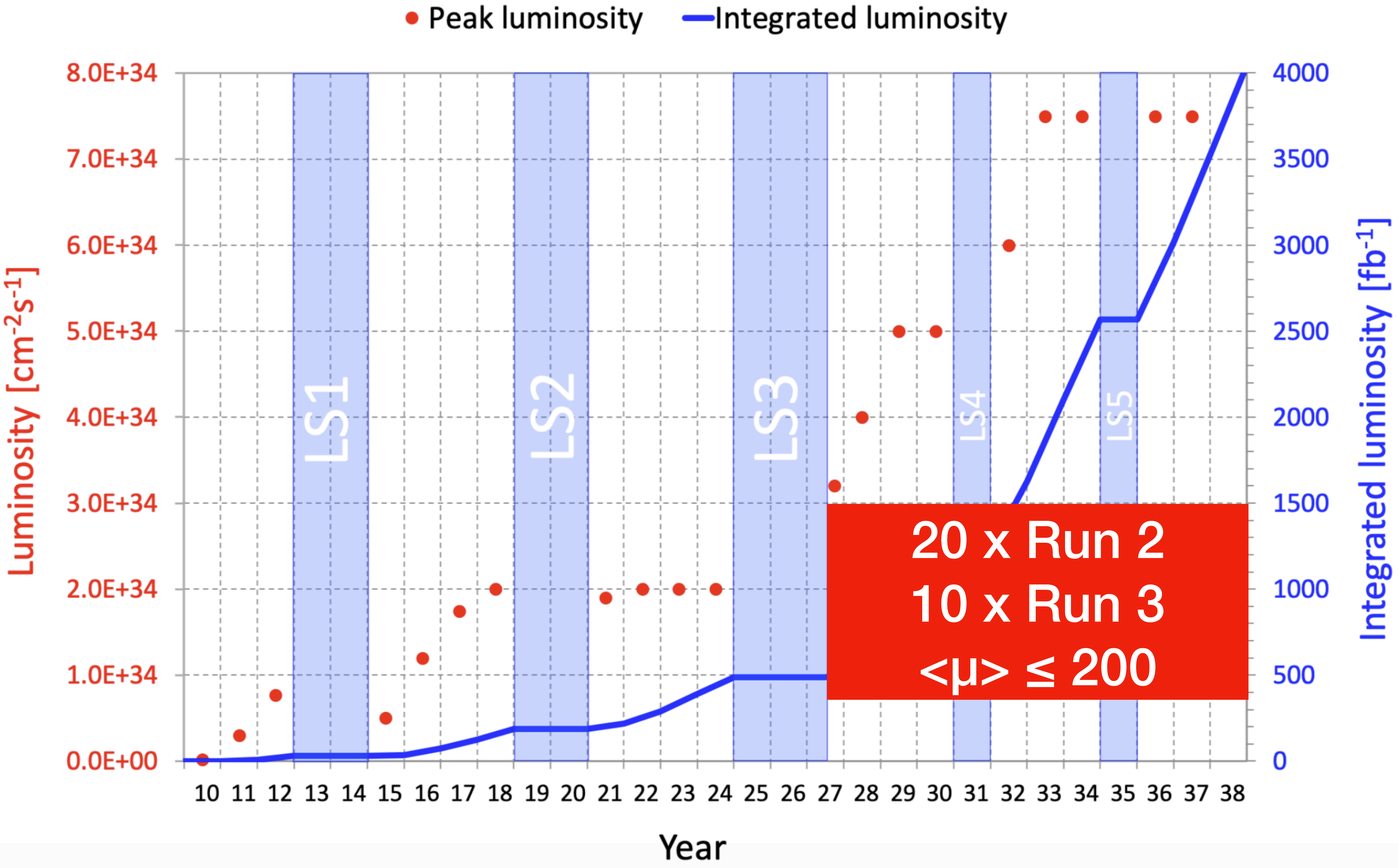
Source

HEP Software Foundation

Community White Paper



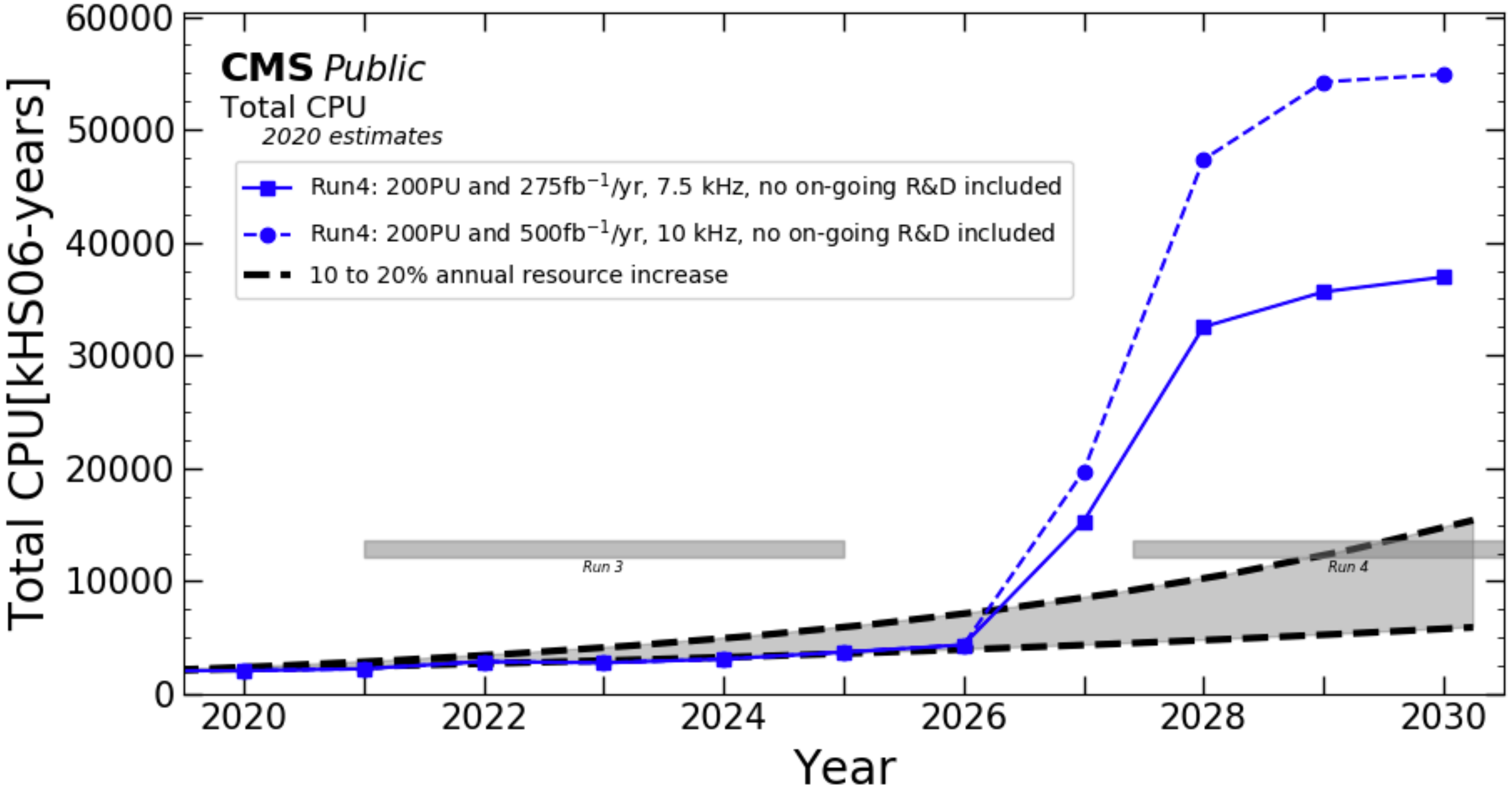
- Frameworks
  - Statistics/ML
  - ROOT
  - Experiment
- Applications
  - DAQ/trigger
  - Analysis
  - Reconstruction
  - Detector simulation
  - Event generation
- Development
  - Repository/CI
  - Debuggers/profilers
  - Compilers
- Data formats
  - Experiment
  - ROOT
  - HepMC
  - Non-HEP
- Grid fabric
  - Data management
  - Workload management
  - Monitoring
  - SW Deployment

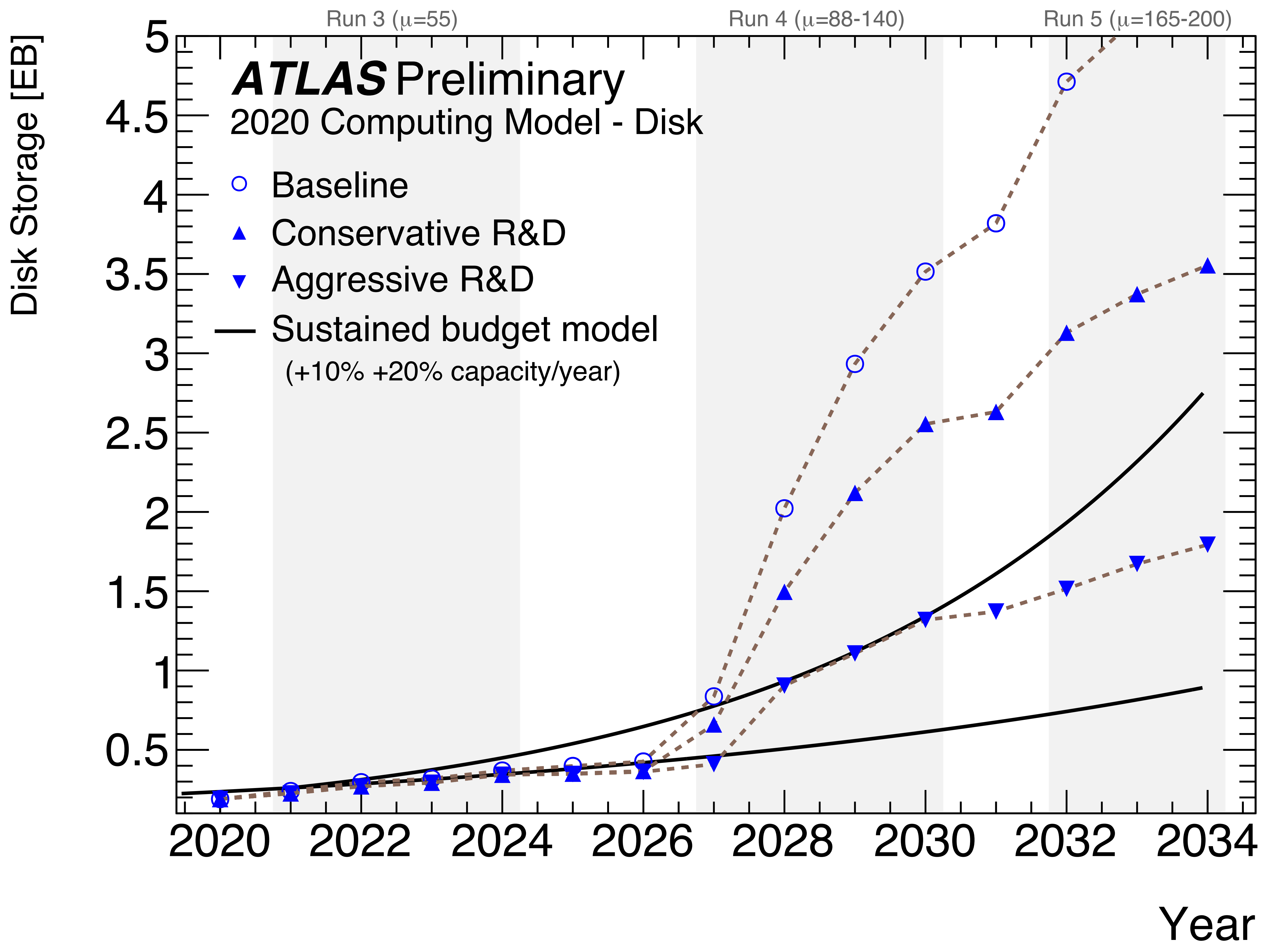


20 x Run 2

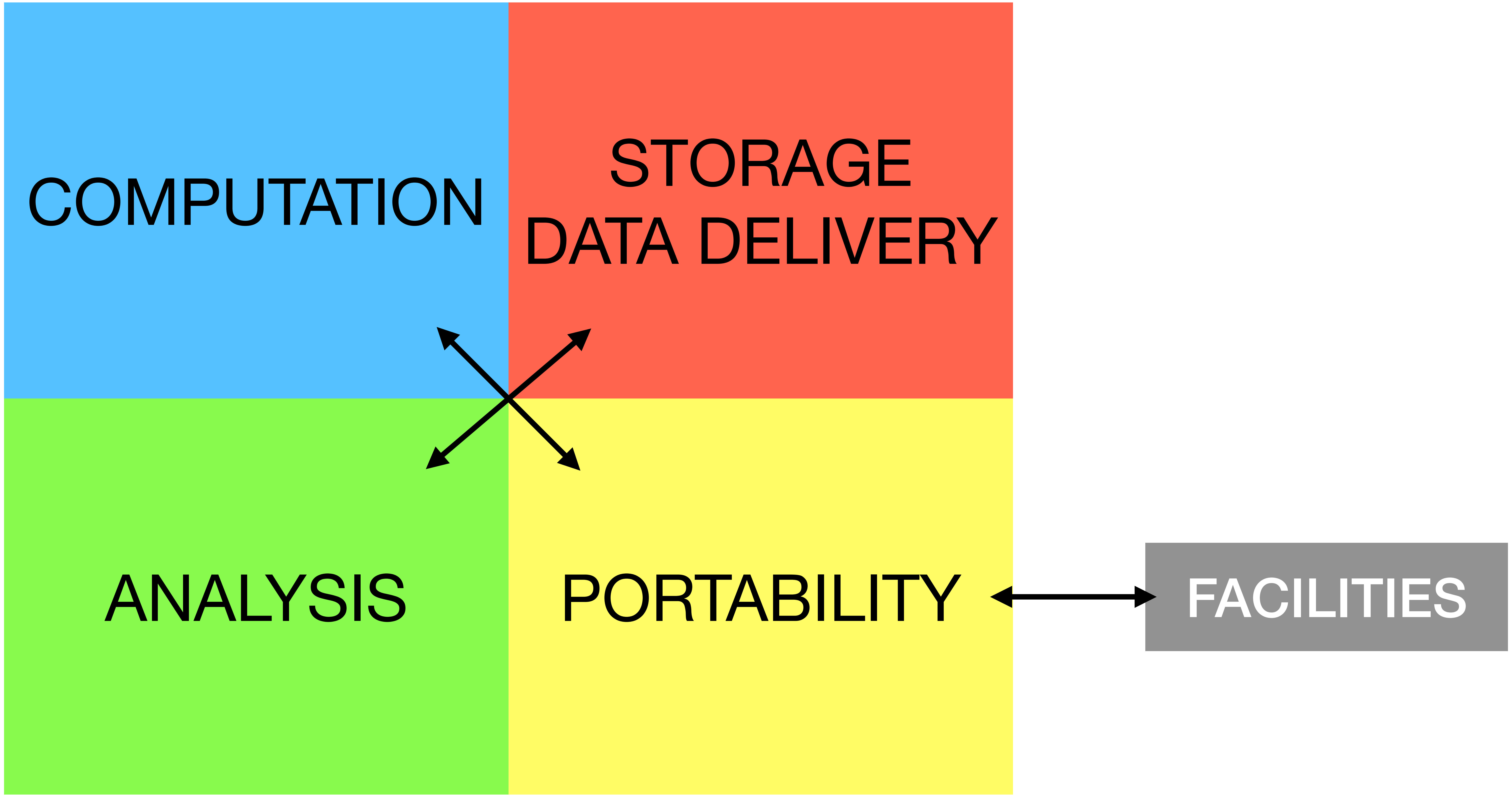
10 x Run 3

<μ> ≤ 200


CMS Public
Total CPU
2020 estimates
Run4: 200PU and 275fb^{-1}/yr, 7.5 kHz, no on-going R&D included
Run4: 200PU and 500fb^{-1}/yr, 10 kHz, no on-going R&D included
10 to 20% annual resource increase
Run 3
Run 4
Total CPU[kHS06-years]
0
10000
20000
30000
40000
50000
60000
2020
2022
2024
2026
2028
2030
Year


Run 3 (μ=55)
Run 4 (μ=88-140)
Run 5 (μ=165-200)
ATLAS Preliminary
2020 Computing Model - Disk
○ Baseline
▲ Conservative R&D
▼ Aggressive R&D
Sustained budget model
(+10% +20% capacity/year)
Disk Storage [EB]
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
2020
2022
2024
2026
2028
2030
2032
2034
Year

Challenges for HL-LHC



Computing and Data Handling session @ ICHEP 2020

- 33 talks
  - Data acquisition: 1
  - Event generation: 3
  - Simulation: 4
  - Reconstruction: 4
  - Analysis techniques: 3
  - Analysis tools: 4
  - Data and workload management: 3
  - General experiment summaries: 7
  - Software management and distribution: 1
  - Monitoring and anomaly detection: 1
  - Quantum (inspired) computing: 2

Excellent talks throughout the sessions!

Covered the full range of activities and issues related to computing and software in the 2020s

I highlight the talks that most directly address the challenges described in the previous slide.

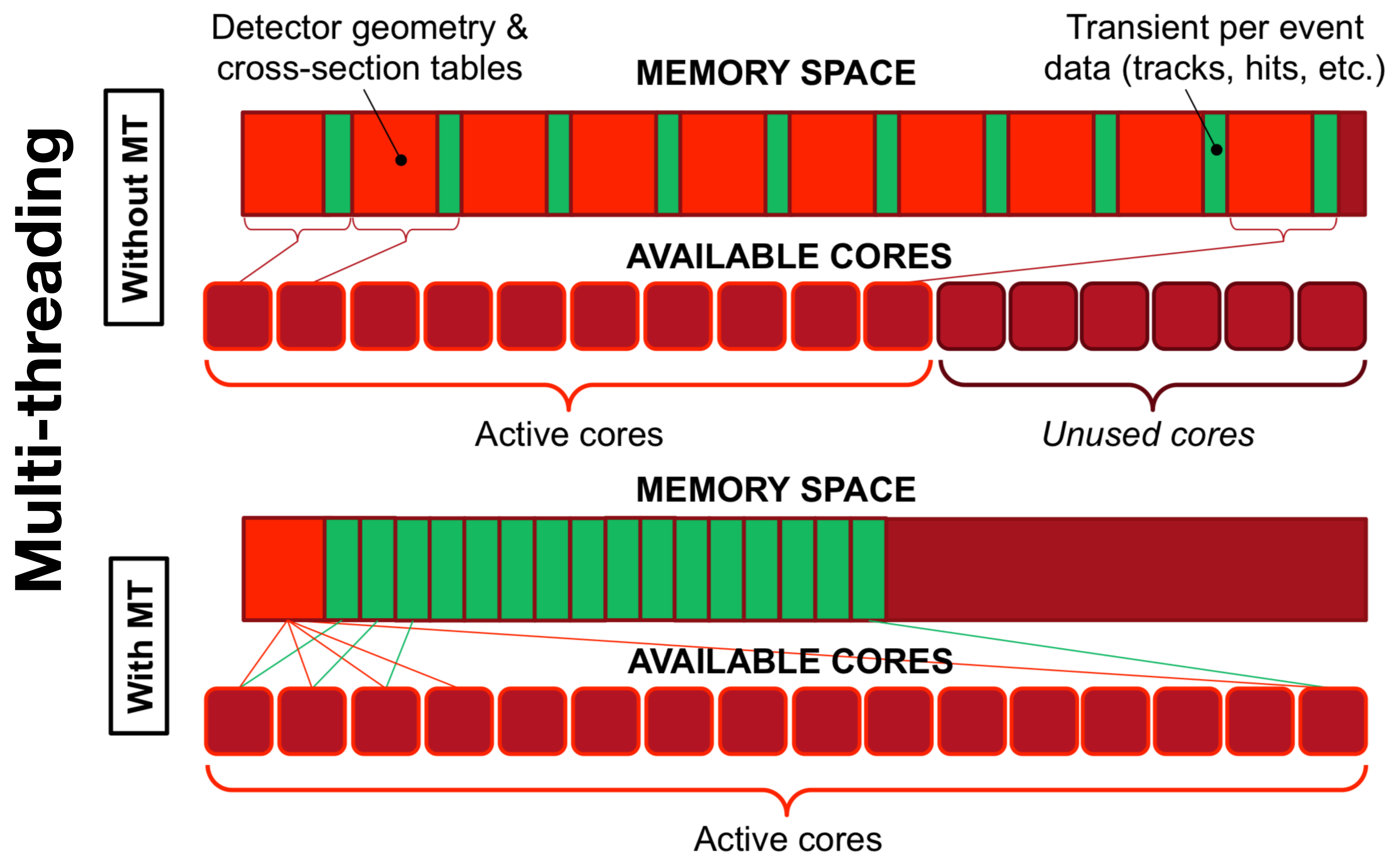
Apologies to speakers whose material is not included.

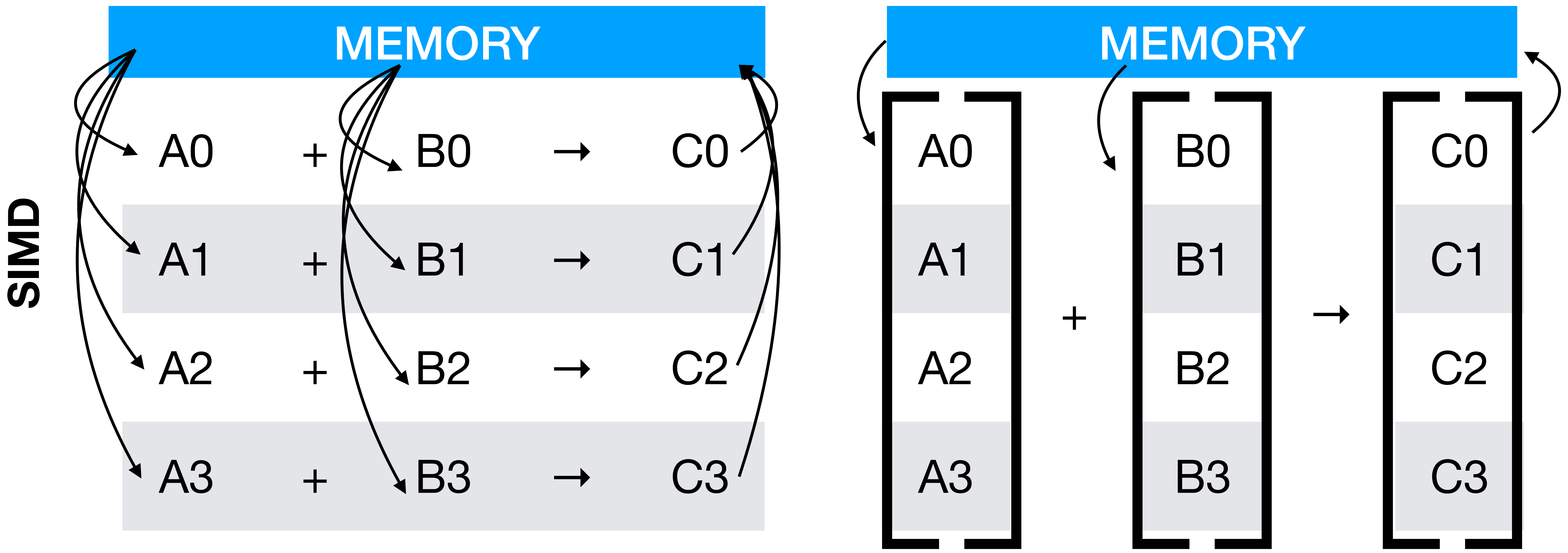
Disclaimer: these are my own interpretations and any errors are mine alone

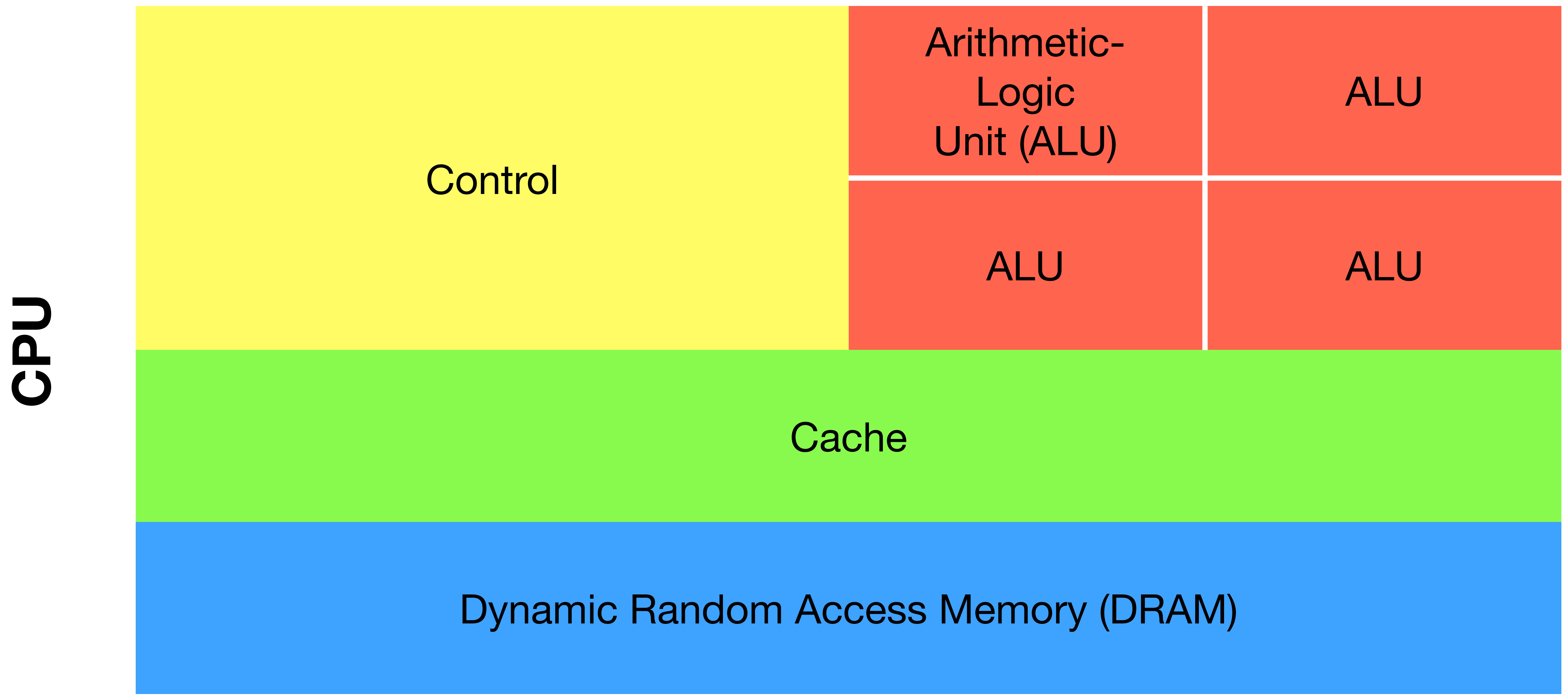
Compute challenge
Portability challenge

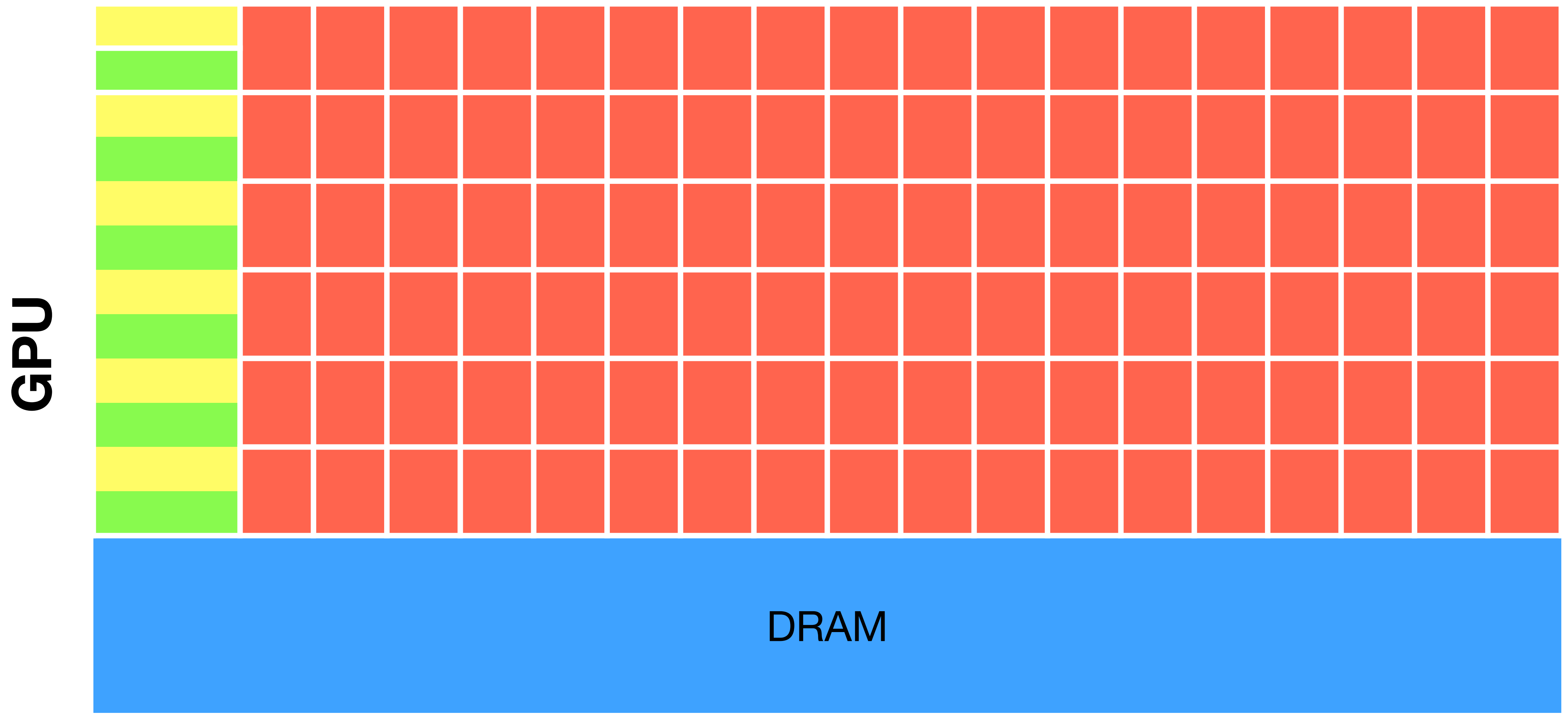
Addressing the Compute Challenge

Can either optimise existing code… or make use of concurrency…






CPU
Control
Arithmetic-Logic Unit (ALU)
ALU
ALU
ALU
Cache
Dynamic Random Access Memory (DRAM)



This is where the Portability Challenge comes in… will portability libraries save us?

Role of machine learning

- **Machine learning is not new in our field** - simple neural networks were in use in LEP days
- **Boosted Decision Trees** have been instrumental in many measurements and discoveries, including single top quark production and some Higgs channels
  - How many physics results presented at this conference *do not* use some form of multi-variate technique?
- In recent years there have been huge advances in **deep learning,** powered by very large and artfully constructed neural networks
- Easy availability of powerful software for building complex neural networks
- Training deep neural networks is particularly well suited to GPUs → possible solution to the portability problem
- Deep learning is already making significant contributions to **analysis** and **simulation**: can be both *better* and *faster*
  - Less obvious for **reconstruction**

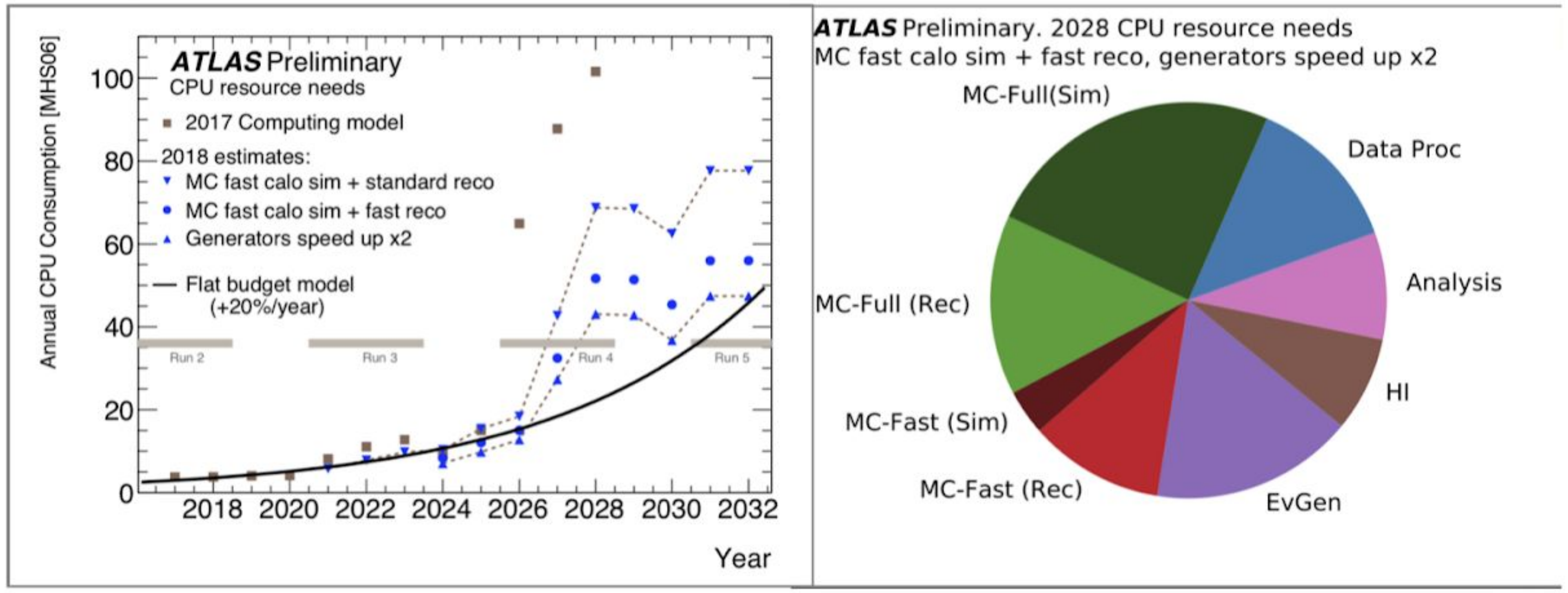
G. Stewart, Wednesday

Anticipated Simulation Needs

Many physics and performance studies require large datasets of simulated events

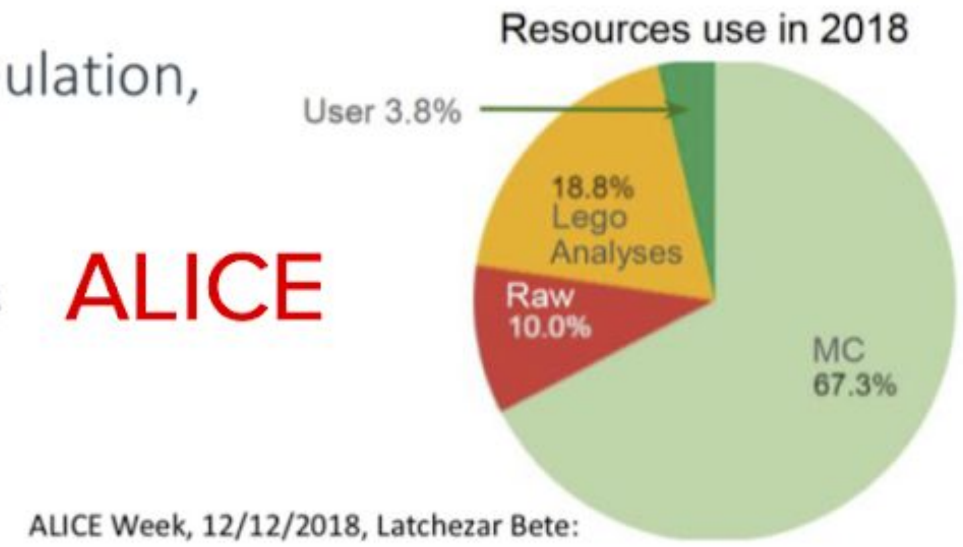
ATLAS

- Geant4 is highly CPU-intensive
- Already lacking statistics -- increasing luminosity poses greater challenges

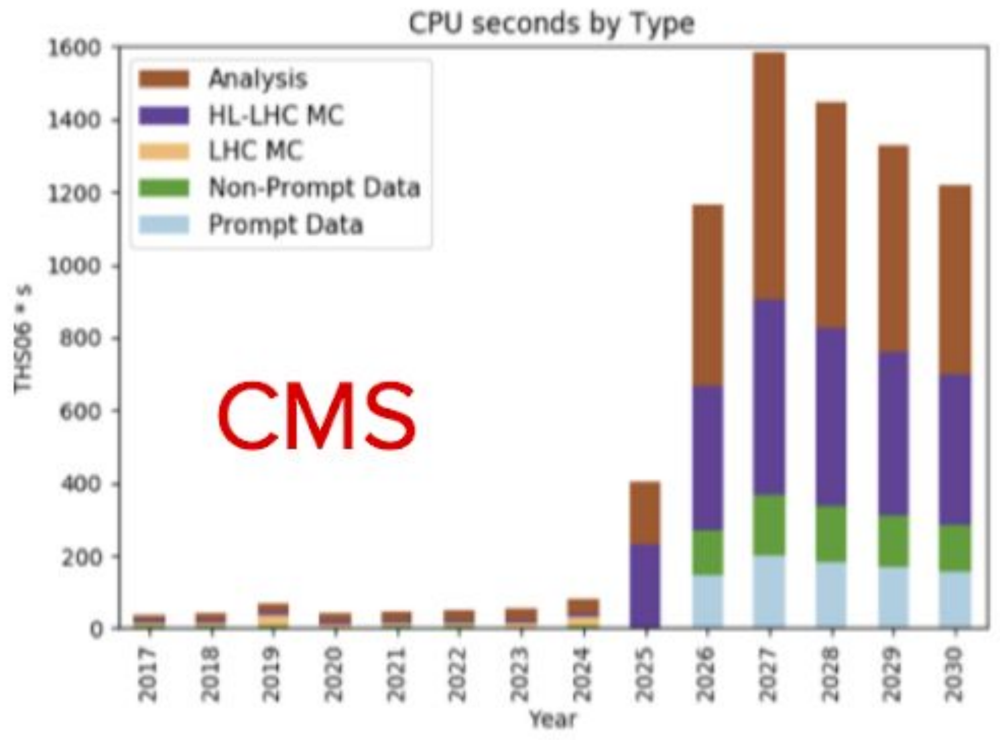


- 2/3 of the computing resources are dedicated to MC simulation, all full sim
  - fast sim not used in production yet
  - fully parametrised fast simulation approach for upgrade studies
- expected 10-100 times more data in Runs 3 and 4
  - cannot cover that with current usage of full sim

ALICE



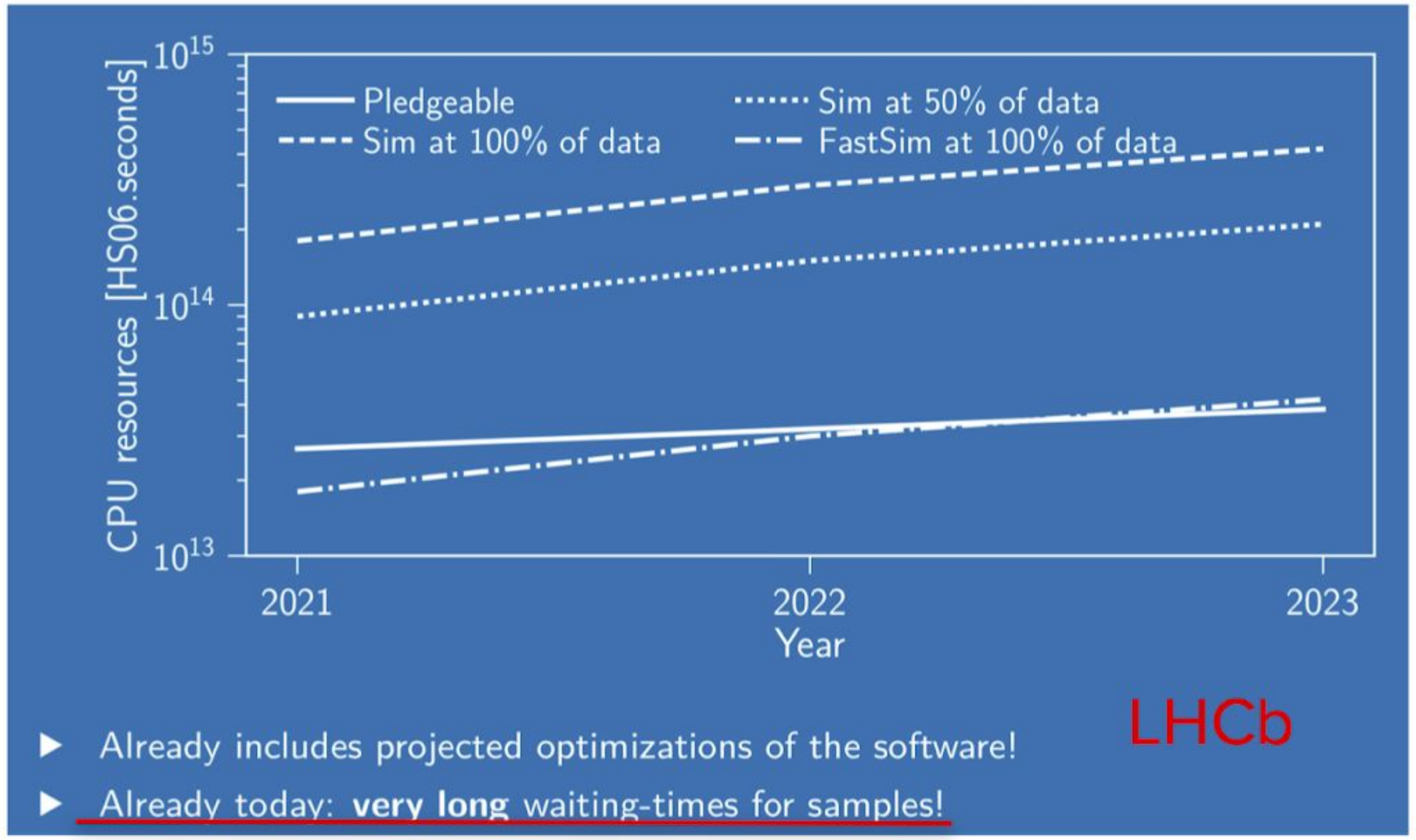
ALICE Week, 12/12/2018, Latchezar Bete:



CMS

- Simulate more events to keep up with HL-LHC data volumes: 10×(Phase1)
- May also need to improve accuracy of physics lists to simulate HGCal
- Reconstruction will take longer due to high pileup and granular detectors

➢ Need more events, more accuracy, in more complicated geometry… w/ relatively smaller fraction of total CPU usage



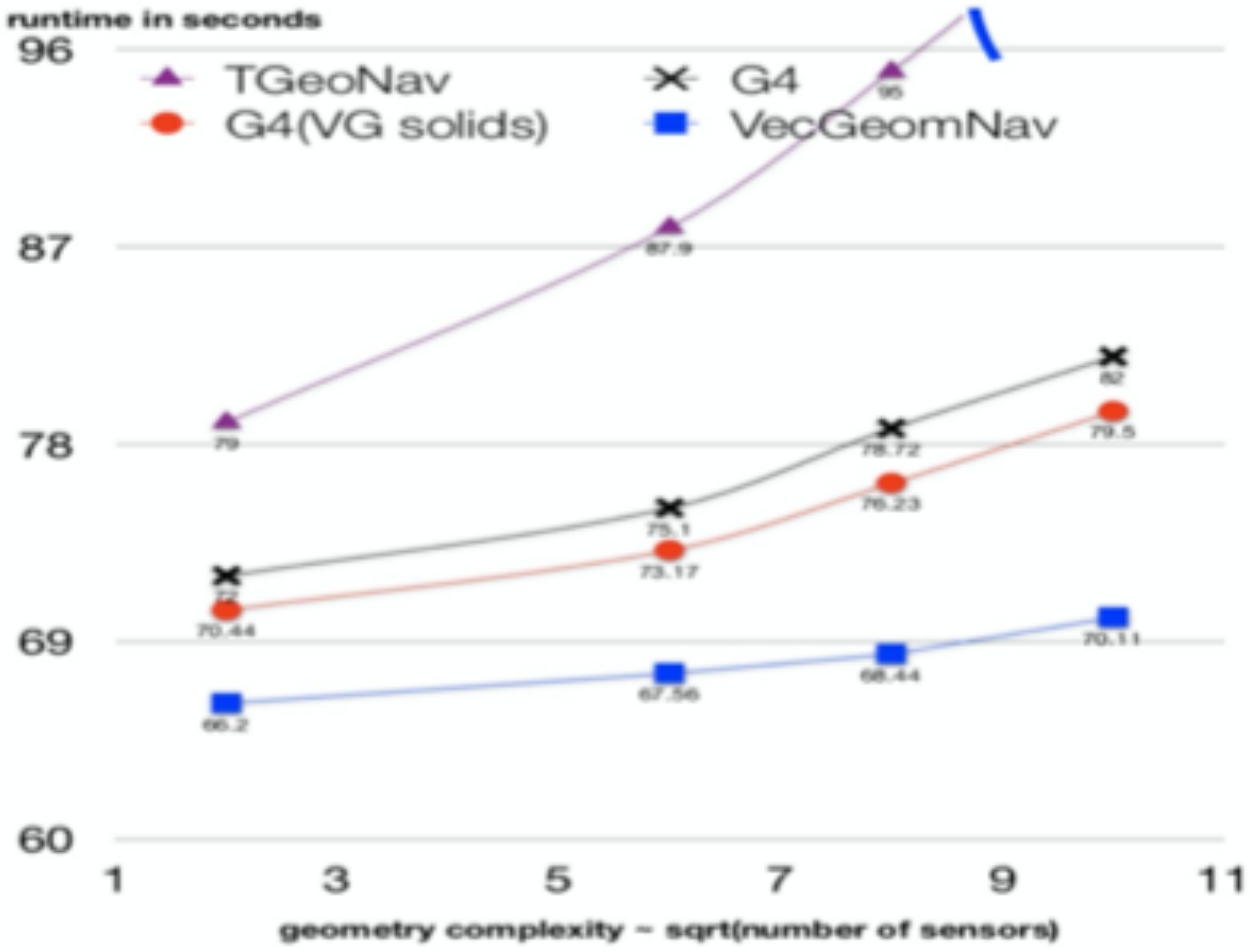
- ▶ Already includes projected optimizations of the software!
- ▶ Already today: **very long** waiting-times for samples!

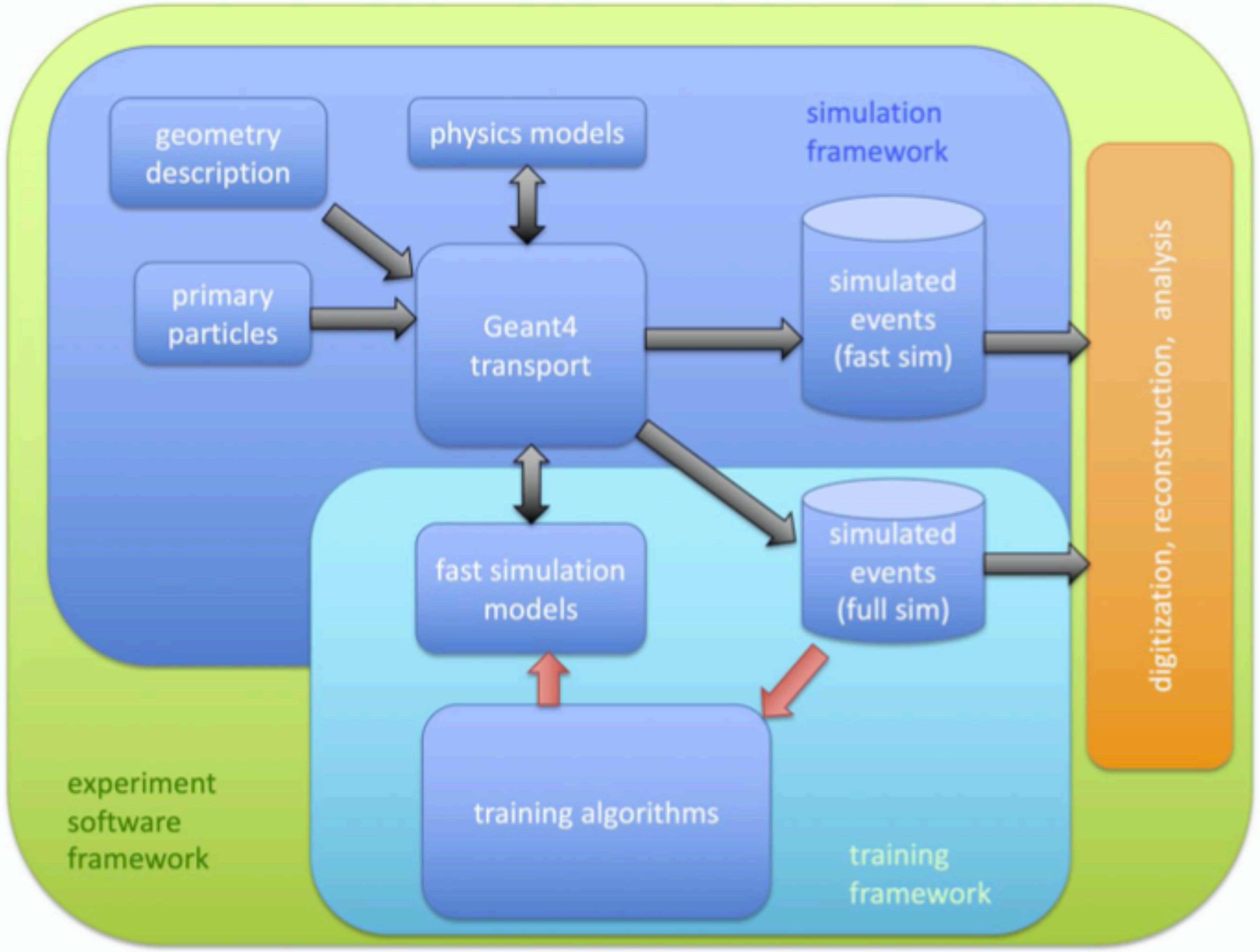
LHCb

G. Stewart, Wednesday

Three Tracks for Improvement

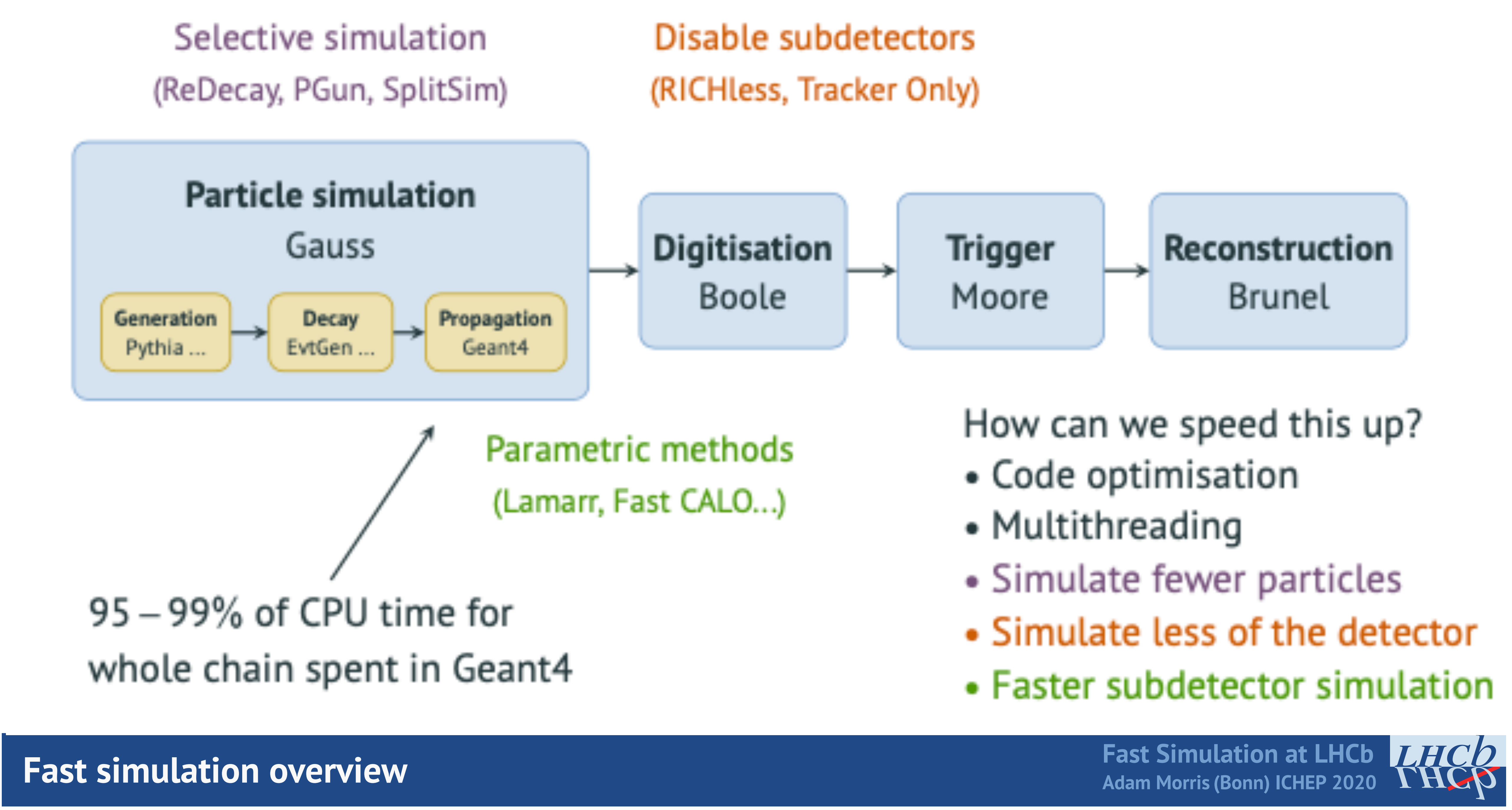
- Refactoring and internal improvements
  - Optimisation of current Geant4 code to run faster
  - Mostly work that is internal to Geant4, little direct impact for user code
- Fast Simulation
  - Replace detailed particle tracking models with different methods
  - Long tradition of parametric response implementations
  - Machine Learning is the hot topic here
- Hardware (R)Evolutions
  - Increasing trend away from purely CPU based machines
  - In particular GPUs become more and more common
  - So we have to start looking at how we could use these machines for detector simulation





- Conclude that rewriting and modernising parts of Geant4 could bring tens of % speed-up, depending on CPU and caches
  - Compact code, use better data layouts, reduce virtual function calls

A. Morris, Wednesday



Selective simulation (ReDecay, PGun, SplitSim)

Disable subdetectors (RICHless, Tracker Only)

Parametric methods (Lamarr, Fast CALO...)

95 – 99% of CPU time for whole chain spent in Geant4

How can we speed this up?
- Code optimisation
- Multithreading
- Simulate fewer particles
- Simulate less of the detector
- Faster subdetector simulation

Fast simulation overview

Fast Simulation at LHCb
Adam Morris (Bonn) ICHEP 2020

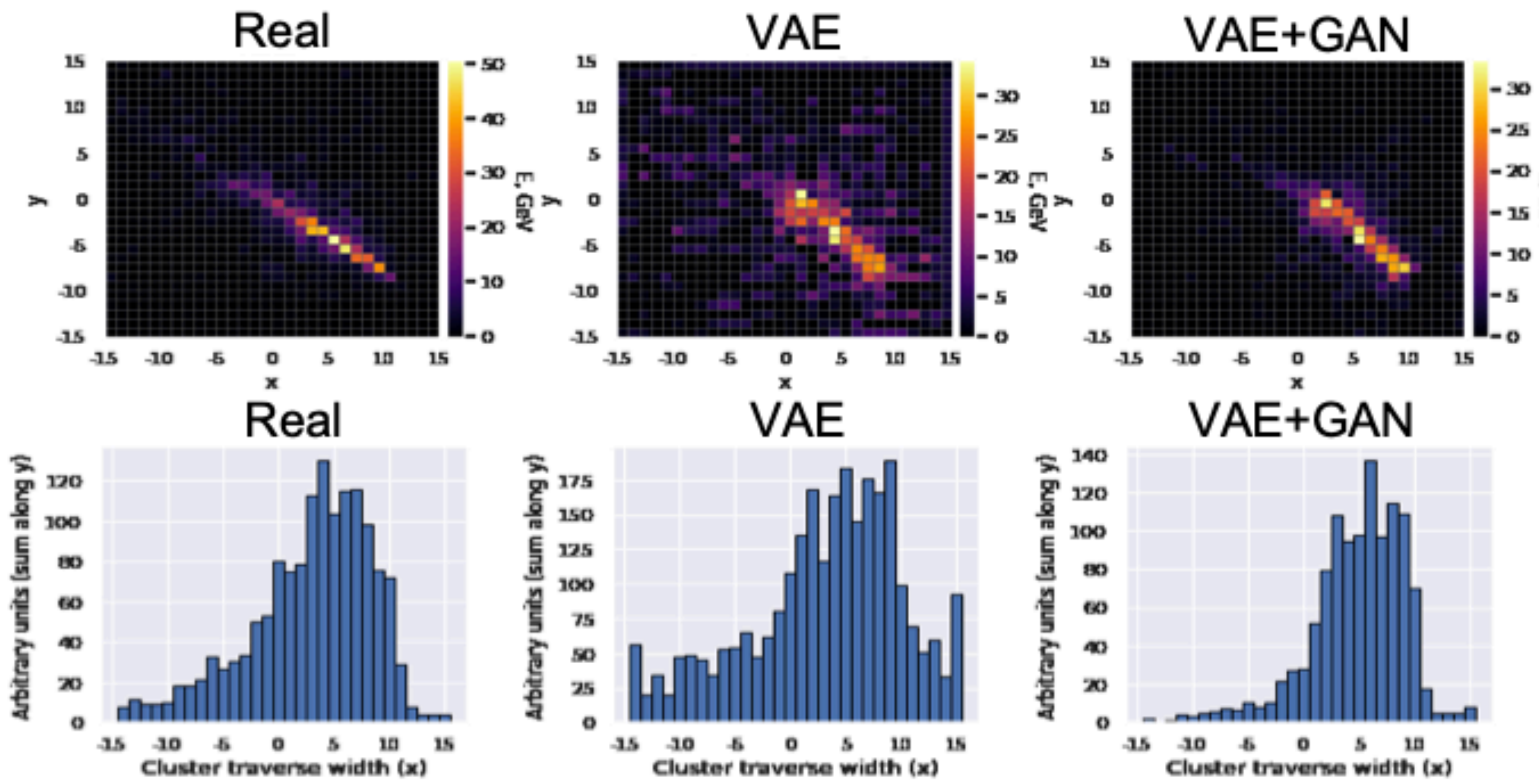
Growing menu of fast simulation options

	Step sped-up					
Method	Generation	Decay	Propagation	Digitisation	Trigger	Reconstruction
ReDecay	✓	✓	✓			
PGun	✓	✓	✓			
SplitSim	✓		✓			
RICHless			✓			
TrackerOnly			✓			
Lamarr			✓	✓	✓	✓
FastCALO*			✓			

* [Separate talk by M. Rama]

M. Rama, Wednesday

- Two lines of development
  - Machine learning techniques
  - Library of energy deposits (shortened as *hit library* in the following)



Under Development

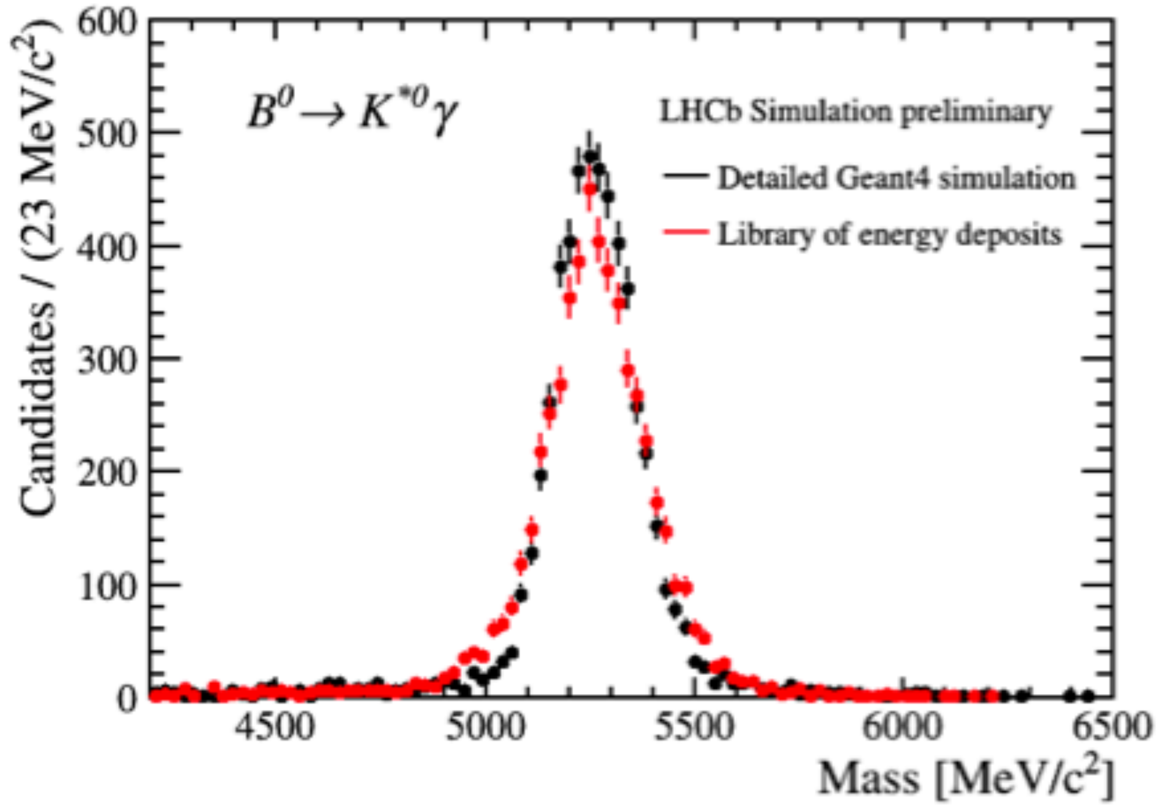
- VAE trains well but the results are often "blurry"
- GAN does not reach good accuracy
- VAE+GAN provides better results

link

- Combined VAE+GAN model performs better than the two separately
- Nonetheless, more work still needed for further improvement

Comparison with Geant4-based simulation using $B^0 \to K^{*0}\gamma$

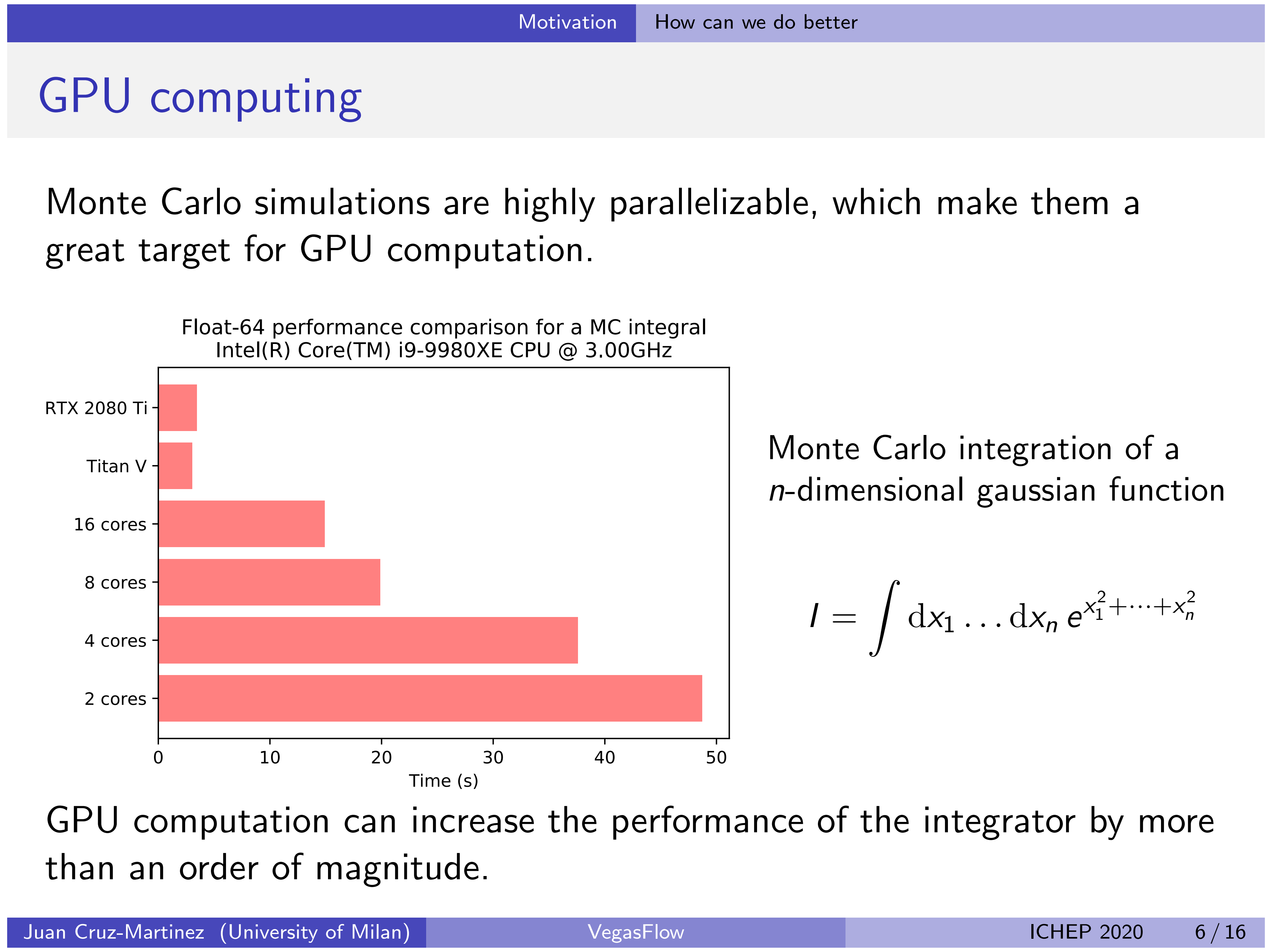
Under Development



- Reconstruction efficiencies consistent within 1% rel. uncertainty
- Some residual differences in the mass shape, should be fixed by
  - building the library with photons entering the calorimeter from different positions
  - possible additional calibrations
- The overall CPU time spent with the library is negligible in Gauss

B mass dependency on E_γ calibration: ~ 17 MeV shift for each % of E_γ bias

J. Cruz-Martinez, Thursday



Motivation | How can we do better

GPU computing

Monte Carlo simulations are highly parallelizable, which make them a great target for GPU computation.

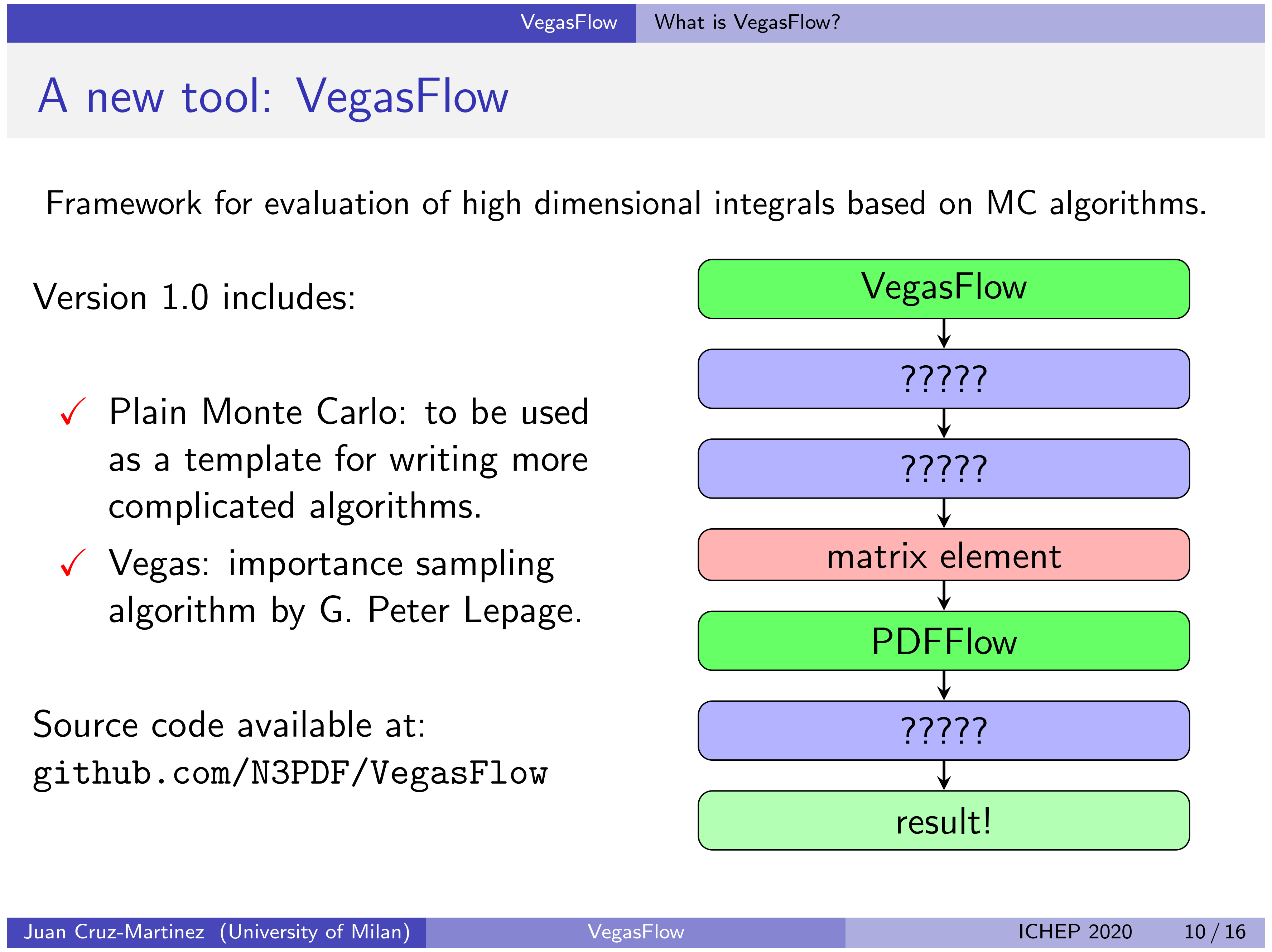
Float-64 performance comparison for a MC integral
Intel(R) Core(TM) i9-9980XE CPU @ 3.00GHz
RTX 2080 Ti
Titan V
16 cores
8 cores
4 cores
2 cores
0
10
20
30
40
50
Time (s)

Monte Carlo integration of a *n*-dimensional gaussian function

$$I = \int \mathrm{d}x_1 \ldots \mathrm{d}x_n \, e^{x_1^2 + \cdots + x_n^2}$$

GPU computation can increase the performance of the integrator by more than an order of magnitude.

Juan Cruz-Martinez (University of Milan) | VegasFlow | ICHEP 2020 | 6 / 16



VegasFlow | What is VegasFlow?

A new tool: VegasFlow

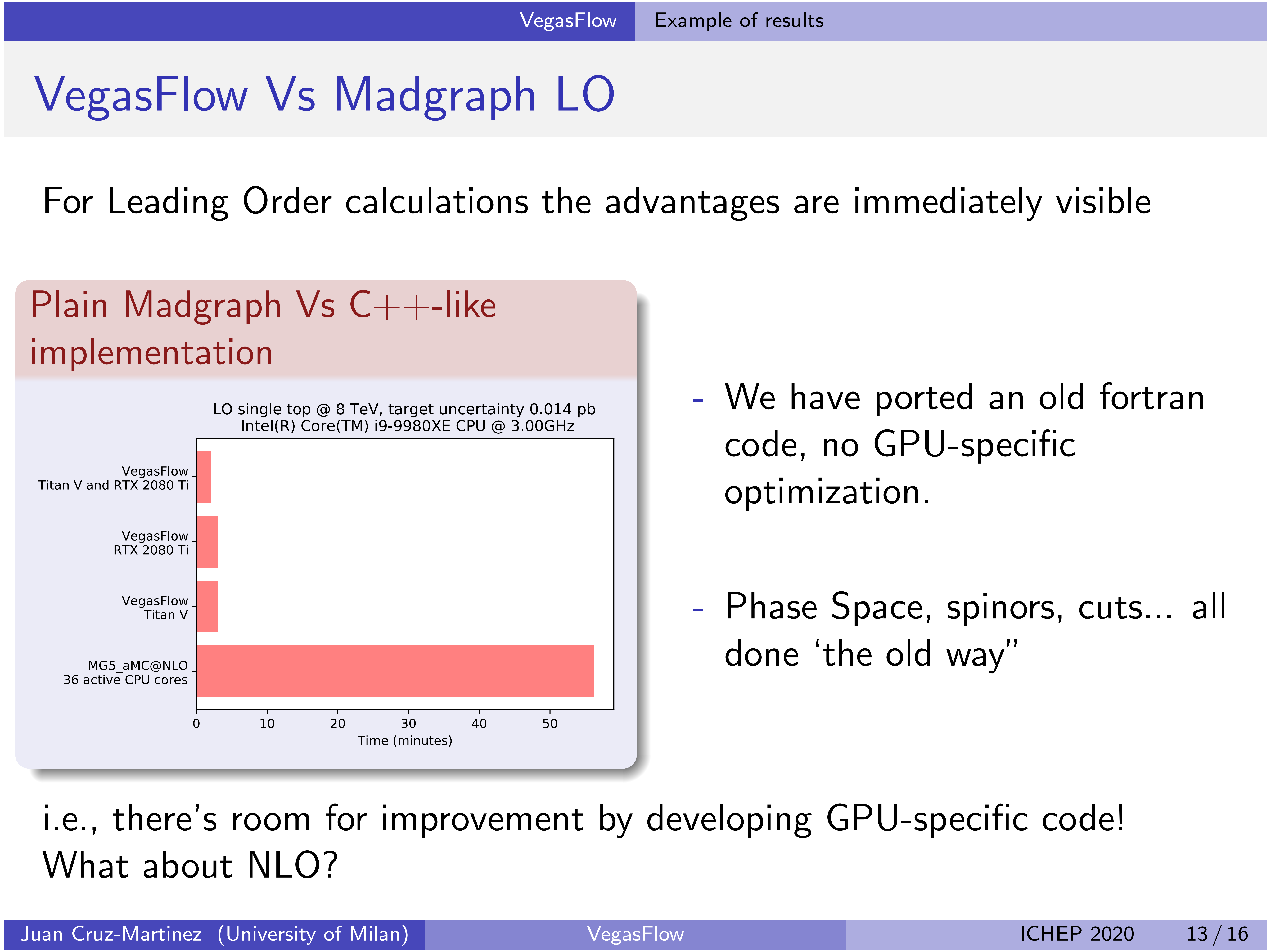
Framework for evaluation of high dimensional integrals based on MC algorithms.

Version 1.0 includes:

- ✓ Plain Monte Carlo: to be used as a template for writing more complicated algorithms.
- ✓ Vegas: importance sampling algorithm by G. Peter Lepage.

Source code available at:
`github.com/N3PDF/VegasFlow`

Juan Cruz-Martinez (University of Milan) | VegasFlow | ICHEP 2020 | 10 / 16



VegasFlow | Example of results

VegasFlow Vs Madgraph LO

For Leading Order calculations the advantages are immediately visible

Plain Madgraph Vs C++-like implementation

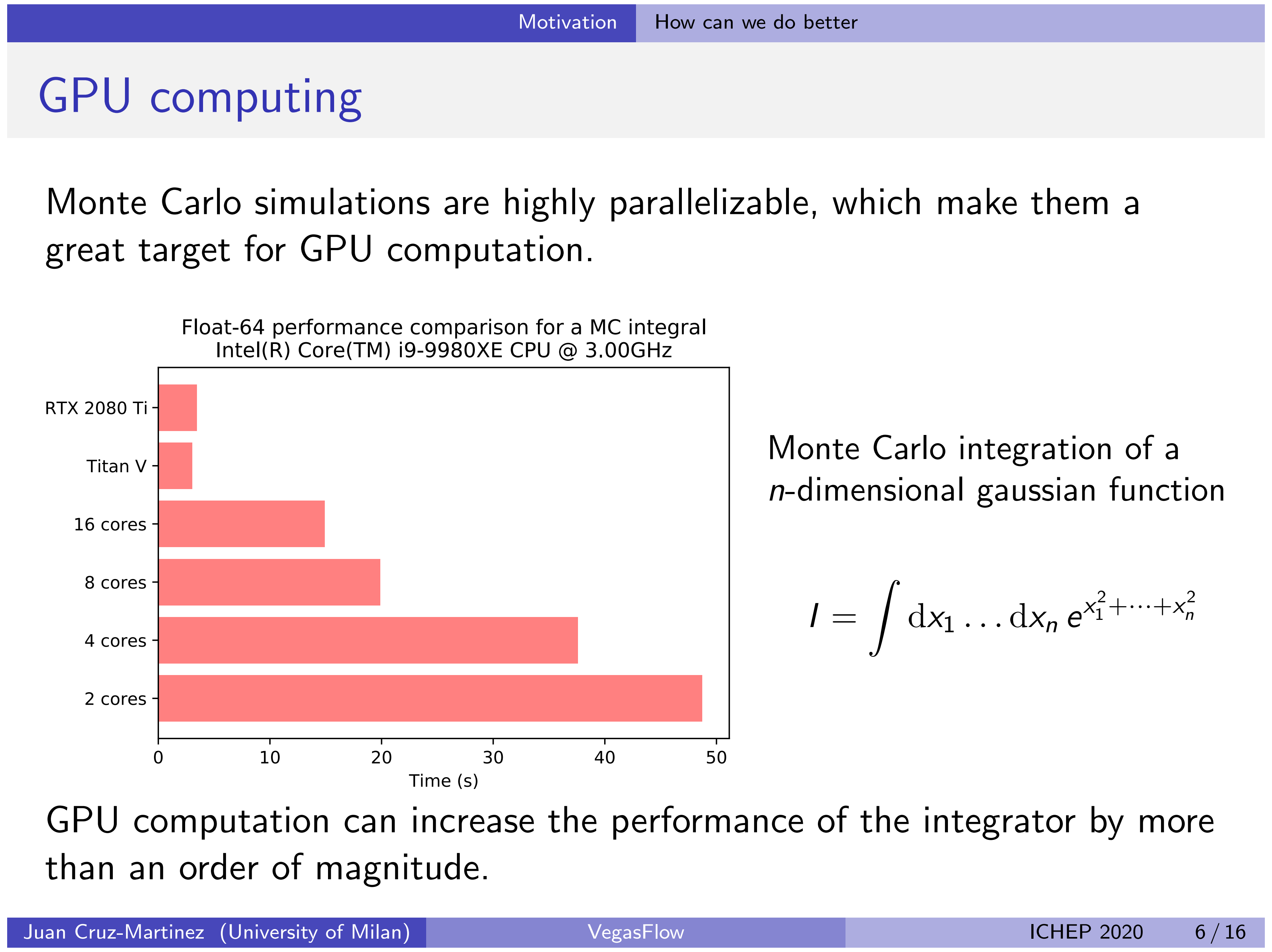
LO single top @ 8 TeV, target uncertainty 0.014 pb
Intel(R) Core(TM) i9-9980XE CPU @ 3.00GHz
VegasFlow Titan V and RTX 2080 Ti
VegasFlow RTX 2080 Ti
VegasFlow Titan V
MG5_aMC@NLO 36 active CPU cores
0
10
20
30
40
50
Time (minutes)

- We have ported an old fortran code, no GPU-specific optimization.
- Phase Space, spinors, cuts... all done ‘the old way”

i.e., there’s room for improvement by developing GPU-specific code!
What about NLO?

Juan Cruz-Martinez (University of Milan) | VegasFlow | ICHEP 2020 | 13 / 16

J. Cruz-Martinez, Thursday



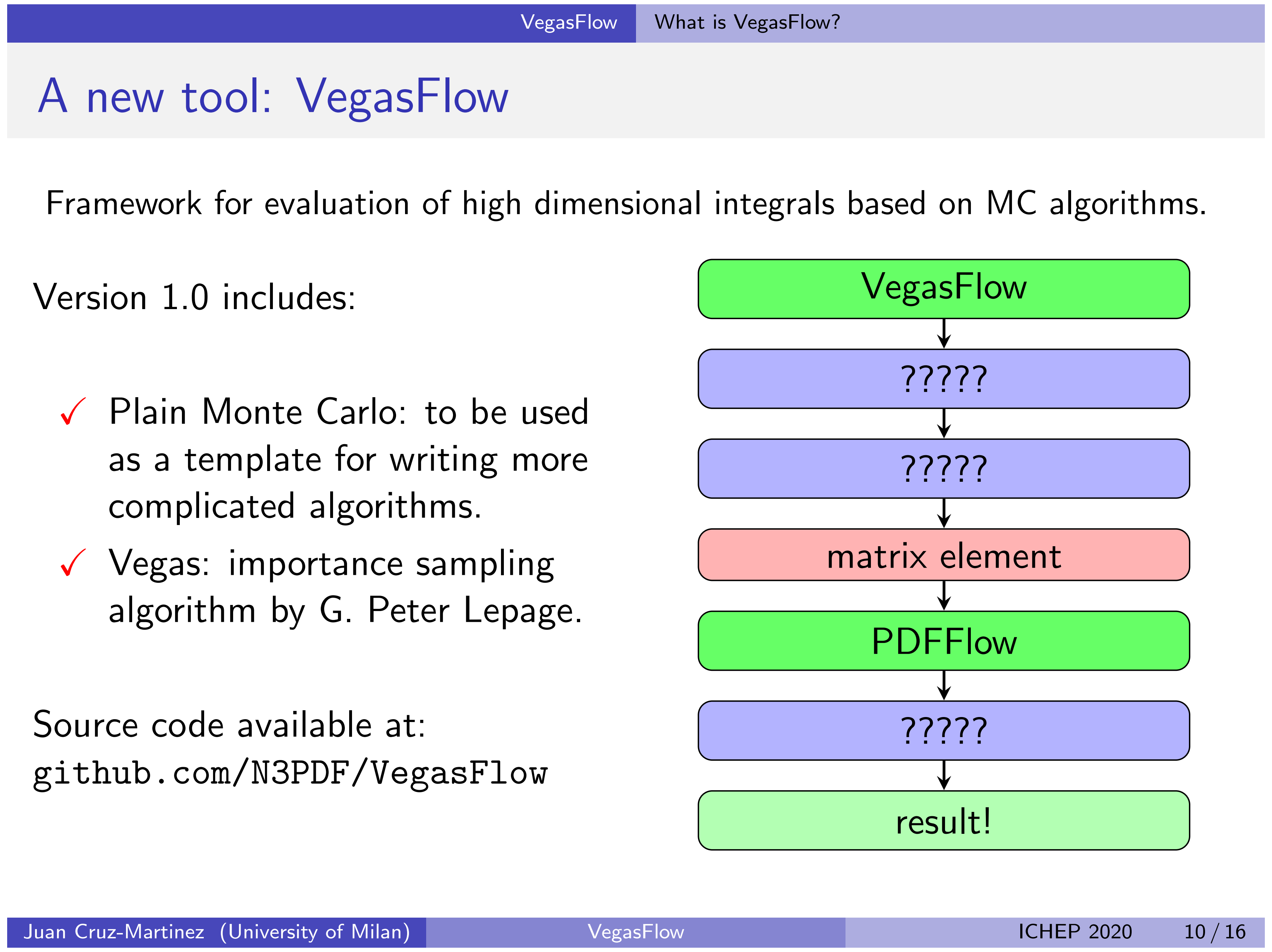
GPU computing

Monte Carlo simulations are highly parallelizable, which make them a great target for GPU computation.

Monte Carlo integration of a *n*-dimensional gaussian function

$$I = \int \mathrm{d}x_1 \dots \mathrm{d}x_n \, e^{x_1^2+\dots+x_n^2}$$

GPU computation can increase the performance of the integrator by more than an order of magnitude.



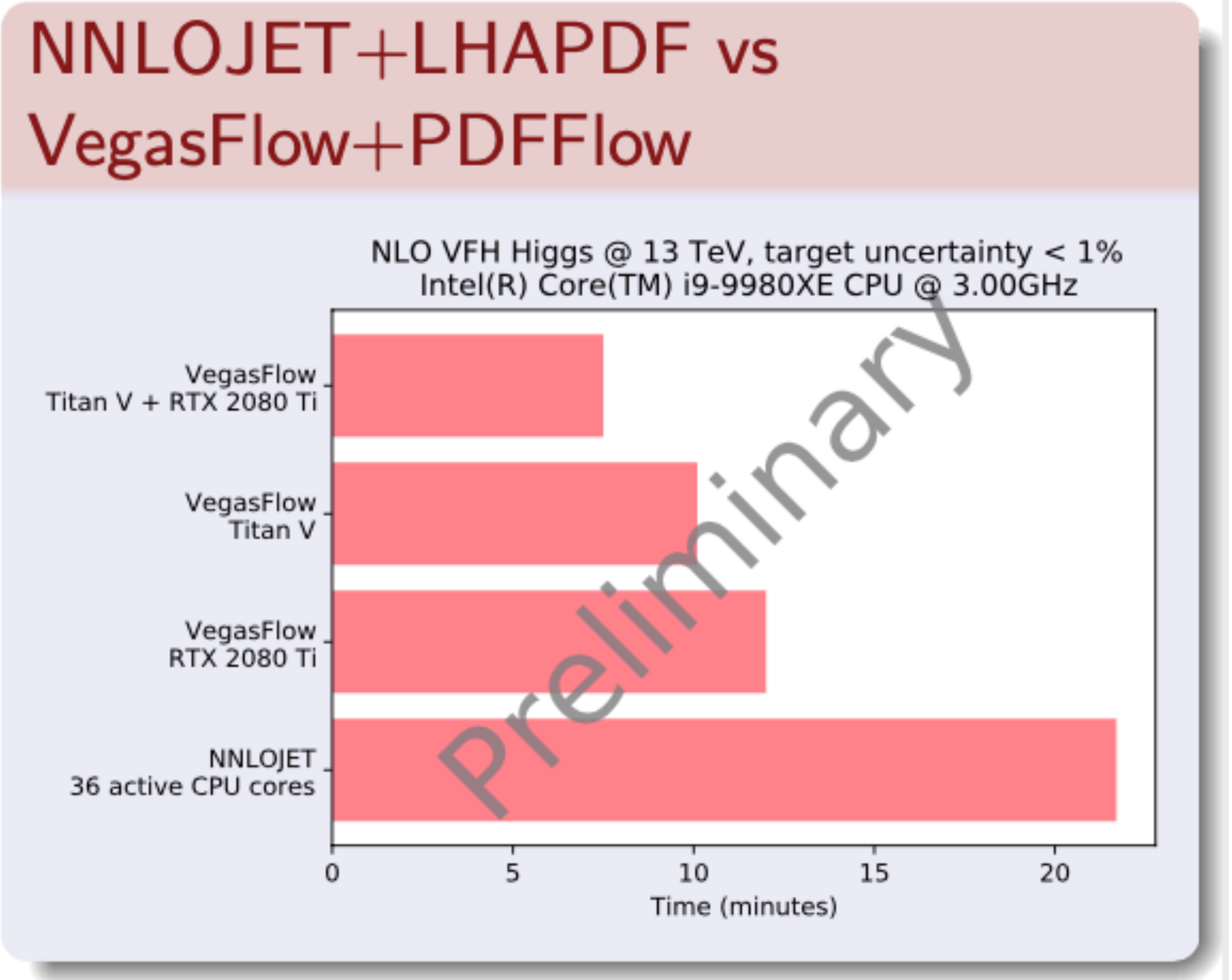
A new tool: VegasFlow

Framework for evaluation of high dimensional integrals based on MC algorithms.

Version 1.0 includes:

- ✓ Plain Monte Carlo: to be used as a template for writing more complicated algorithms.
- ✓ Vegas: importance sampling algorithm by G. Peter Lepage.

Source code available at:
`github.com/N3PDF/VegasFlow`



NNLOJET+LHAPDF vs VegasFlow+PDFFlow

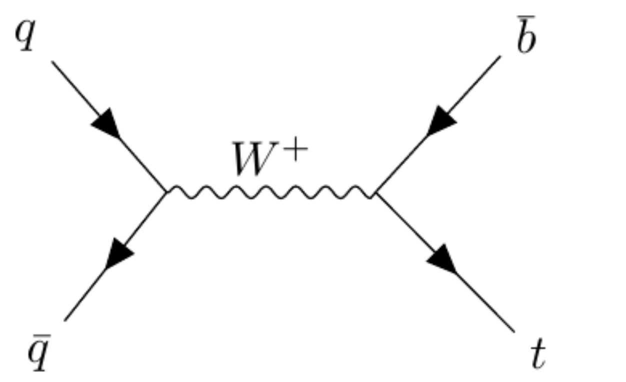
M. Rossi, Friday

PDFFlow: PDF interpolation

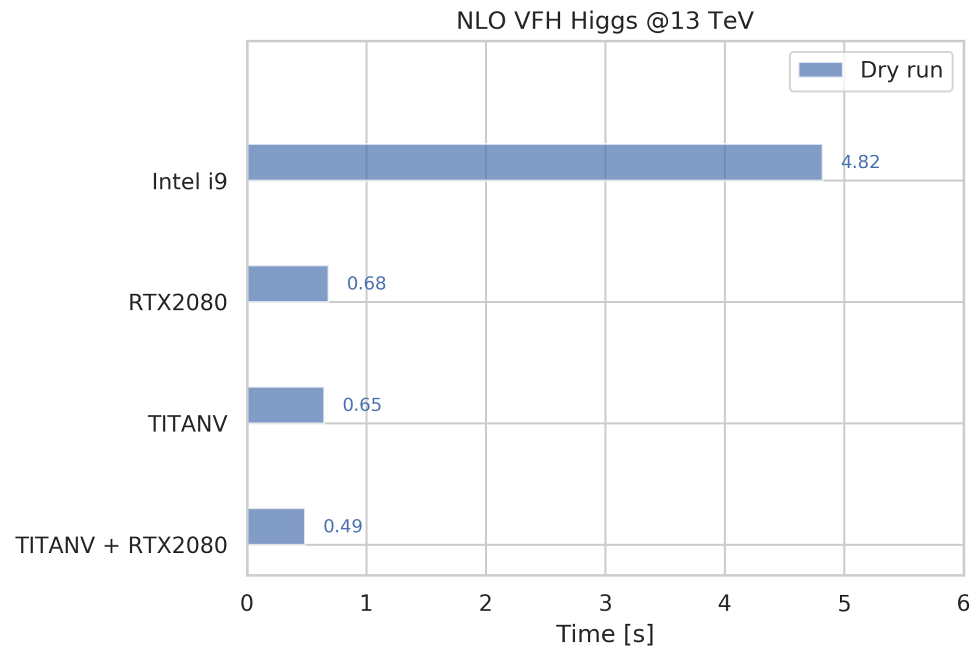
Physical example - Single top production at LO

Combine `PDFFlow` and `VegasFlow` (MC integrator, 10.1016/j.cpc.2020.107376)

Speed comparison CPU-GPU for `PDFFlow` + `VegasFlow`



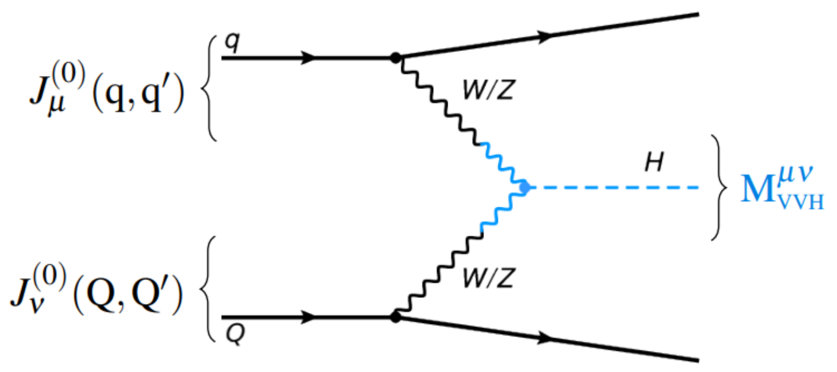
Born diagram for $q\bar{q} \to \bar{b}t$



Single top dry run

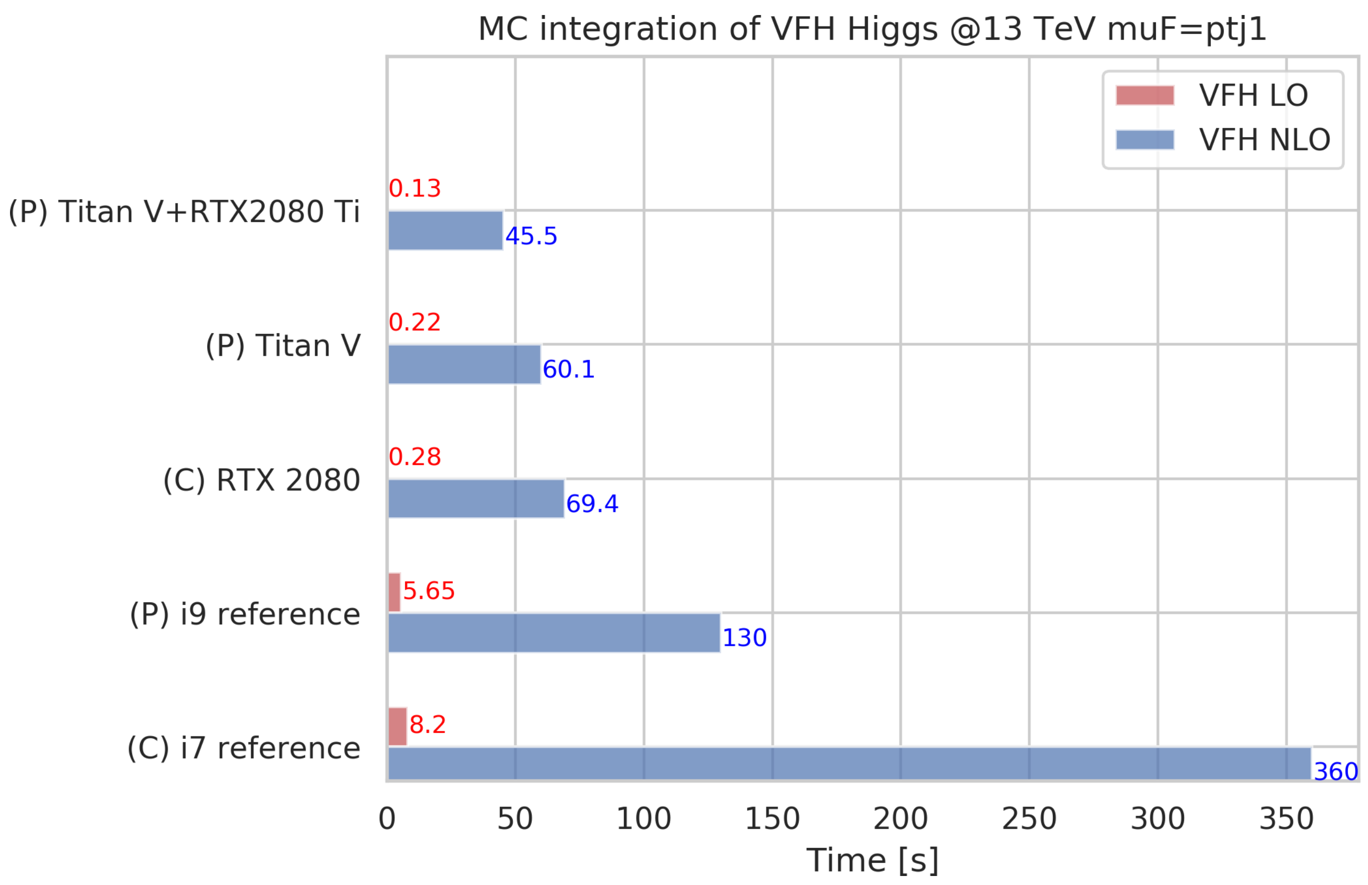
✓ Speed-up range: $[7.0, 9.9]x$

Physical example - VBF Higgs production at NLO



Born diagram for $qQ \to qQH$

hep-ph/arXiv:1807.07908



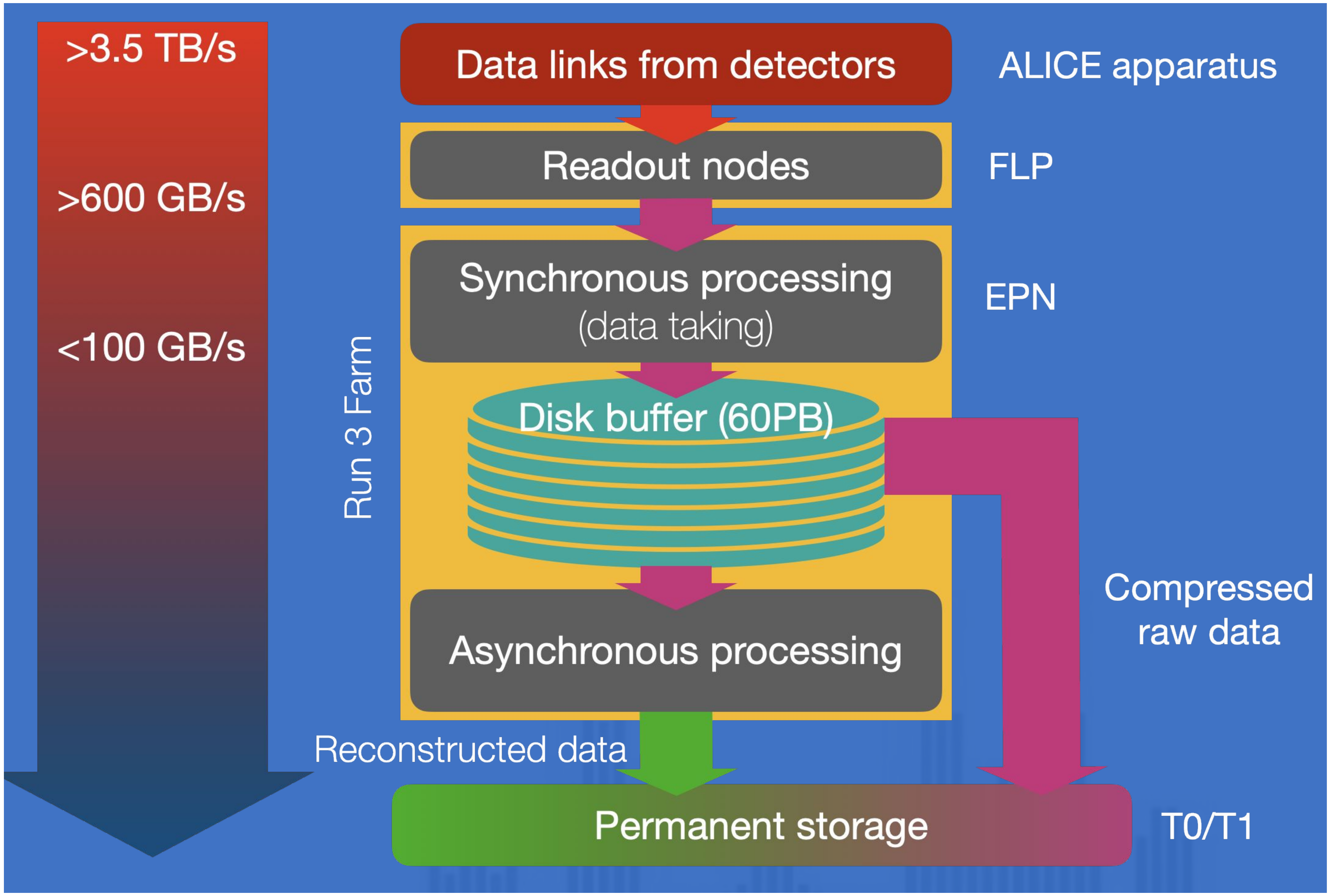
(C) consumer-grade
(P) professional-grade hardware

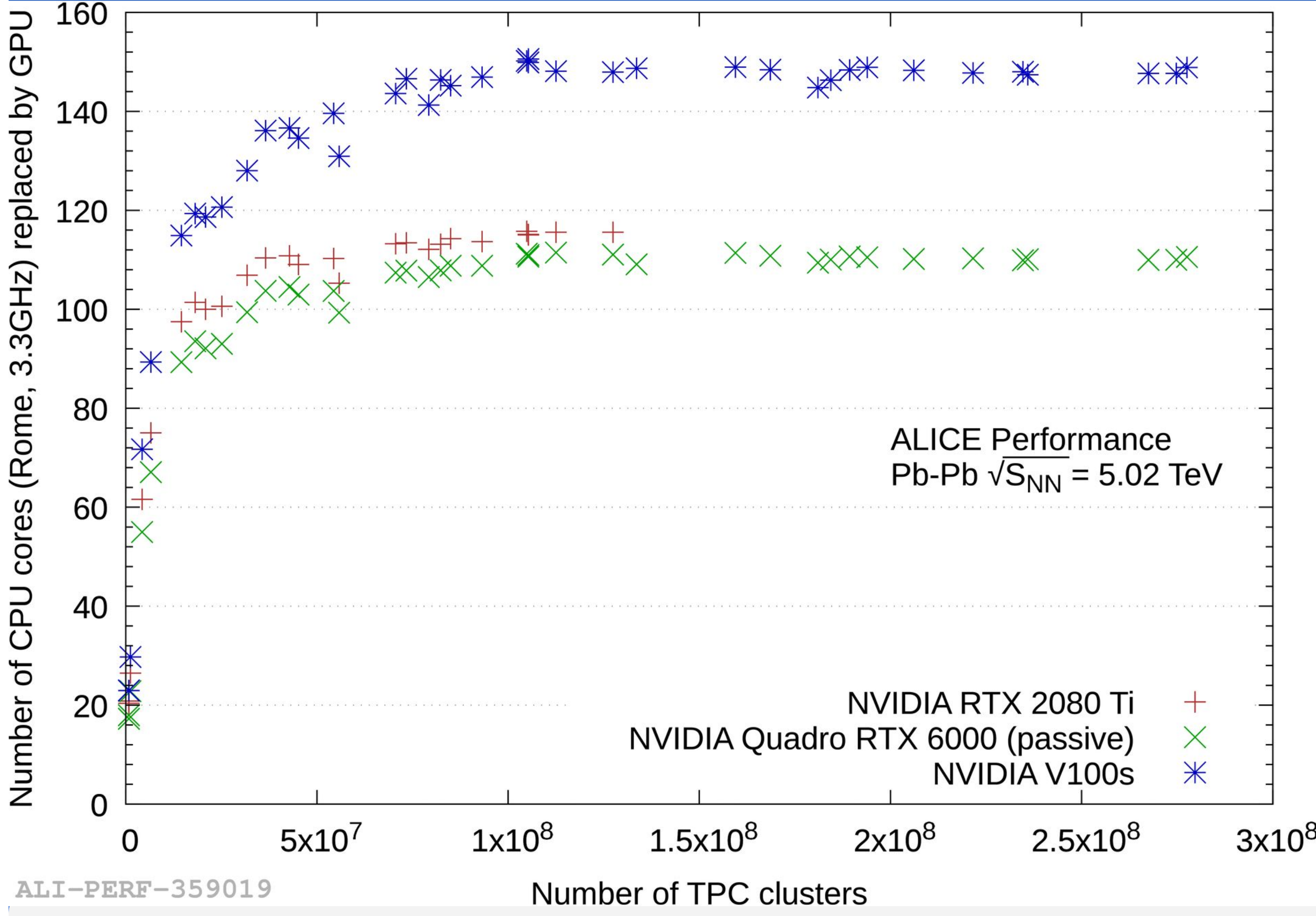
✓ Best speed-up at LO: $63x$(C), $43x$(P)

✓ Best speed-up at NLO: $7.9x$(C), $2.9x$(P)

CPU implementation: `LHAPDF` + Fortran code
GPU implementation: `PDFFlow` + `VegasFlow`

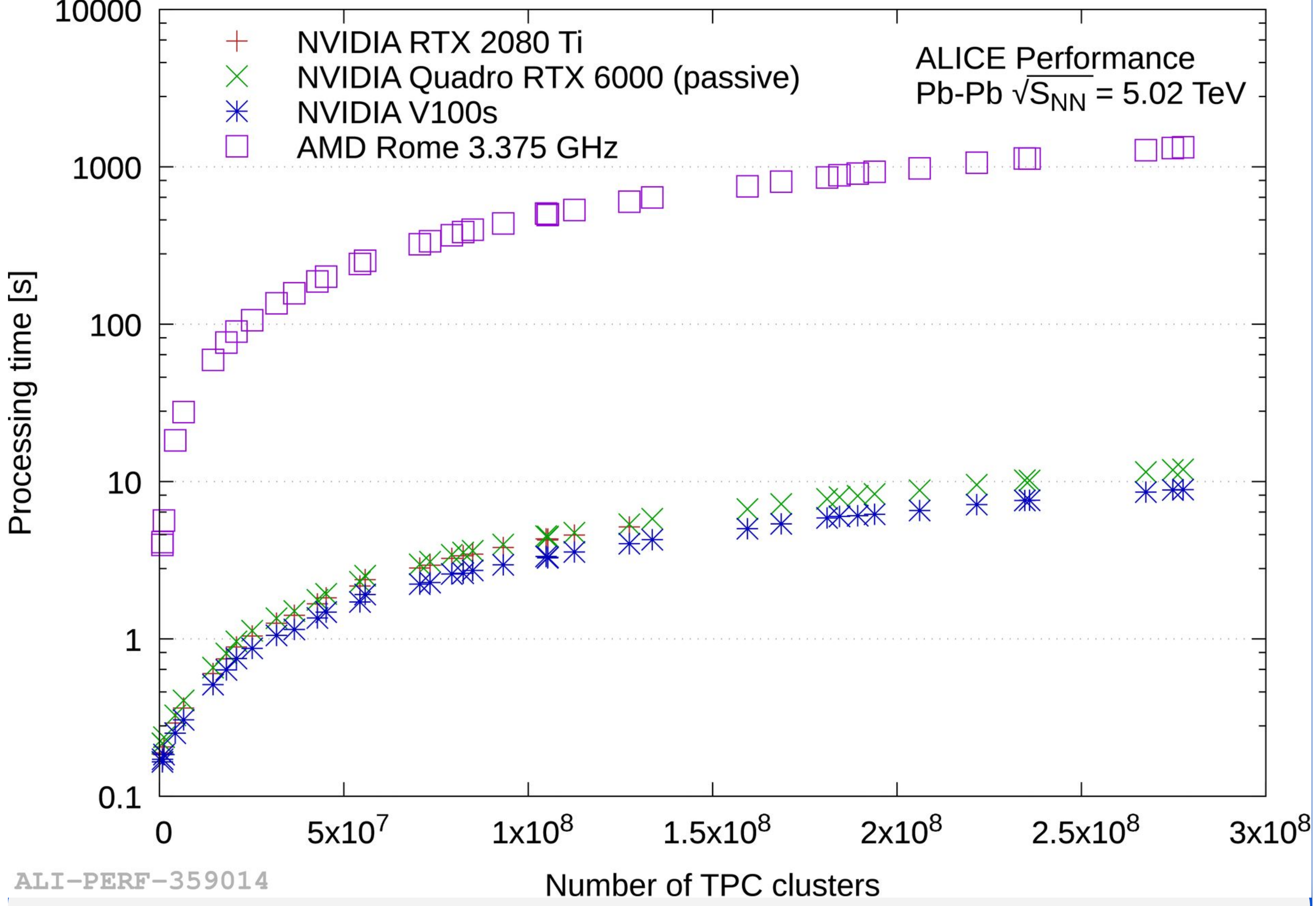
C. Zampolli, Thursday | M. Concas, Thursday





Average speedup normalised to one core for different GPU cards and a comparison with former offline tracking vs HLT tracking.

- 40-150 CPUs replaced by one GPU
- TPC tracking speeded up by factor of 50-100



Comparisons for the computing time spent between recent CPU and GPUs, two orders of magnitude faster

Synchronous processing

Goal of synchronous reconstruction is to reach factor 35 of compression.

Most relevant detector is TPC: from 3.4 TB/s to 70 GB/s

TPC data compression will consist of:

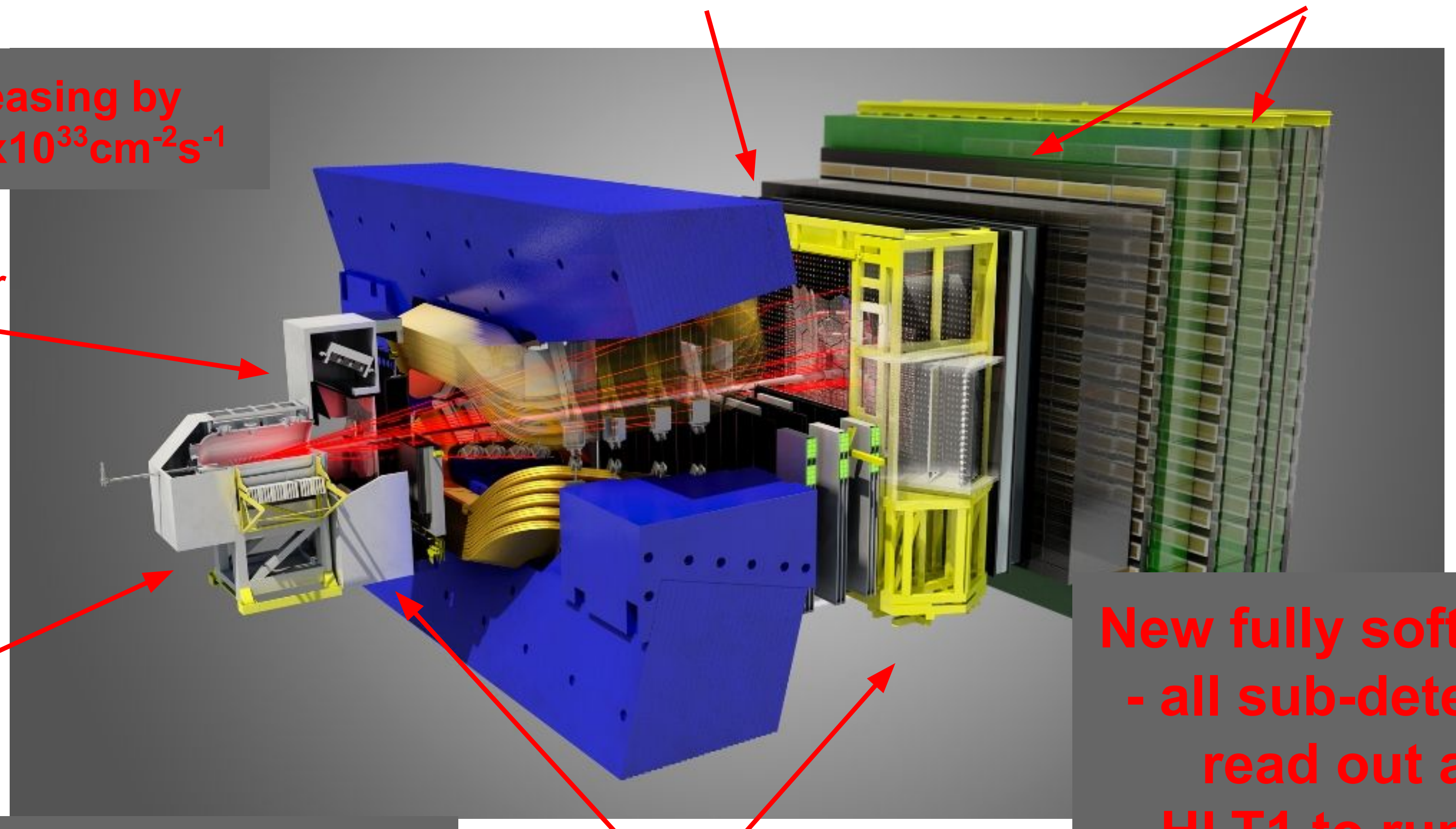
- **Clusterization**
- **TPC tracking**

USE OF GPUs MANDATORY

> 40x faster than CPU but only 4x more expensive

N. Skidmore, Friday

LHCb Upgrade I



Inst. Lumi. increasing by factor 5 up to $2 \times 10^{33} cm^{-2} s^{-1}$

New scintillating fibre tracker (SciFi)

New electronics (CALO, MUON)

New silicon upstream tracker (UT)

New pixel vertex detector (VELO)

New optics and photodetectors (RICH)

Target Int. Lumi. $50 fb^{-1}$

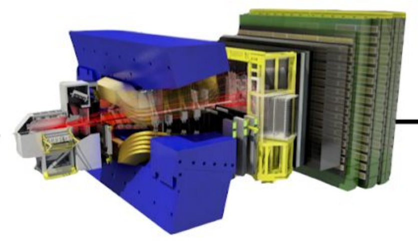
New fully software trigger - all sub-detectors must read out at 40MHz HLT1 to run on GPUs

LHCB-TDR-12
LHCB-TDR-13
LHCB-TDR-14
LHCB-TDR-15
LHCB-TDR-16
LHCB-TDR-17
LHCB-TDR-18

Data flow evolution Upgrade

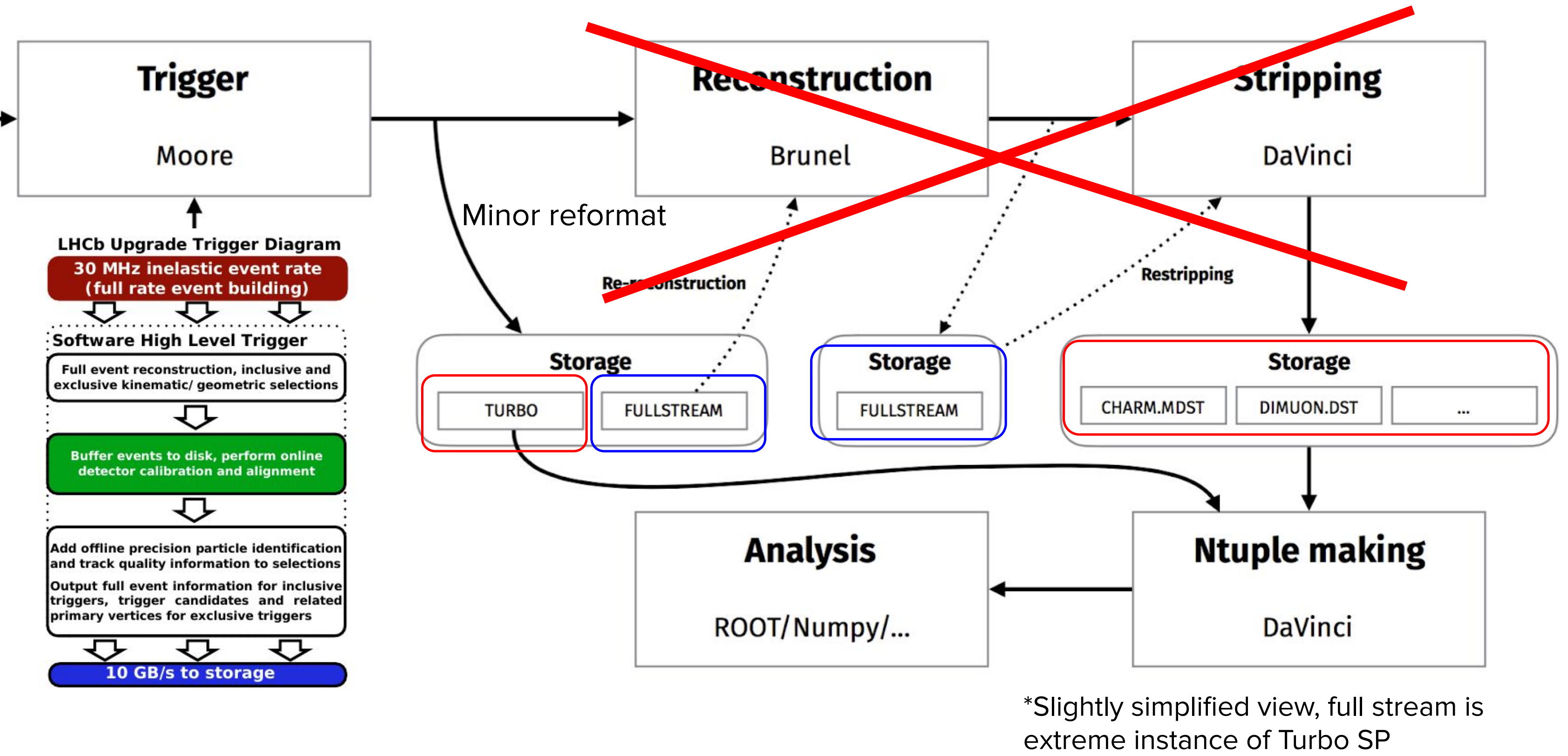
All reconstruction, alignment and calibration performed **online**

- TURBO (+SP) stream ~immediately available on disk
- FULL stream saves **full reconstructed event*** - RAW data can be removed
- TURCAL stream for calibration saves FULL+RAW info



Tape storage - not accessible to users
Disk storage - available to users

For LHCb upgrade trigger see talk #520 and #521


Trigger
Moore
LHCb Upgrade Trigger Diagram
30 MHz inelastic event rate (full rate event building)
Software High Level Trigger
Full event reconstruction, inclusive and exclusive kinematic/ geometric selections
Buffer events to disk, perform online detector calibration and alignment
Add offline precision particle identification and track quality information to selections
Output full event information for inclusive triggers, trigger candidates and related primary vertices for exclusive triggers
10 GB/s to storage
Minor reformat
~~Reconstruction~~
Brunel
~~Stripping~~
DaVinci
Re-reconstruction
Restripping
Storage
TURBO
FULLSTREAM
Storage
FULLSTREAM
Storage
CHARM.MDST
DIMUON.DST
...
Analysis
ROOT/Numpy/...
Ntuple making
DaVinci

*Slightly simplified view, full stream is extreme instance of Turbo SP

Online

Offline

Heterogenous resources

Worldwide LHC Computing Grid (WLCG) consists of ~ 1M CPU cores over 170 sites

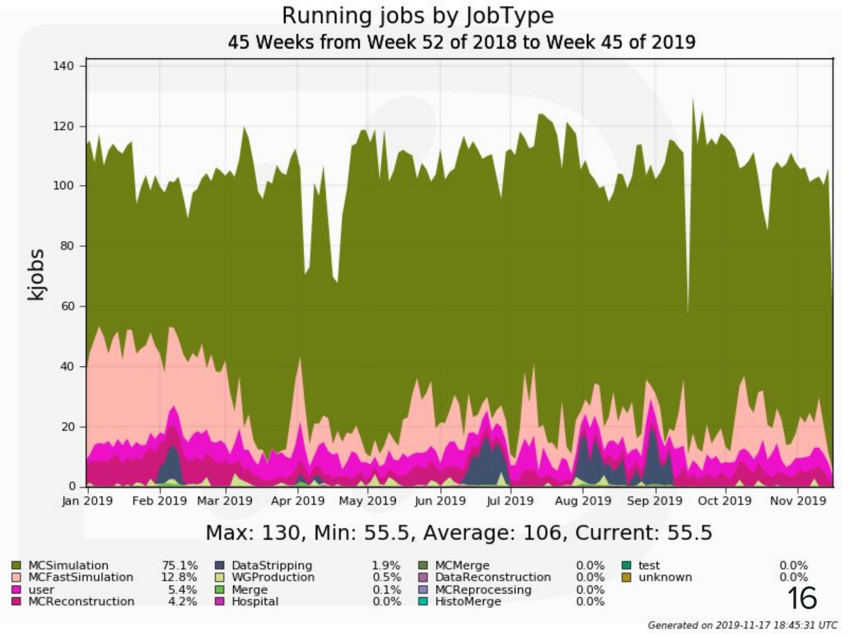
Most sites have no GPUs yet - push towards High Performance Computing (HPC) centers providing **large GPU resources**

Potential to **utilise HLT1 GPU farm** like current HLT CPU farm during detector downtime

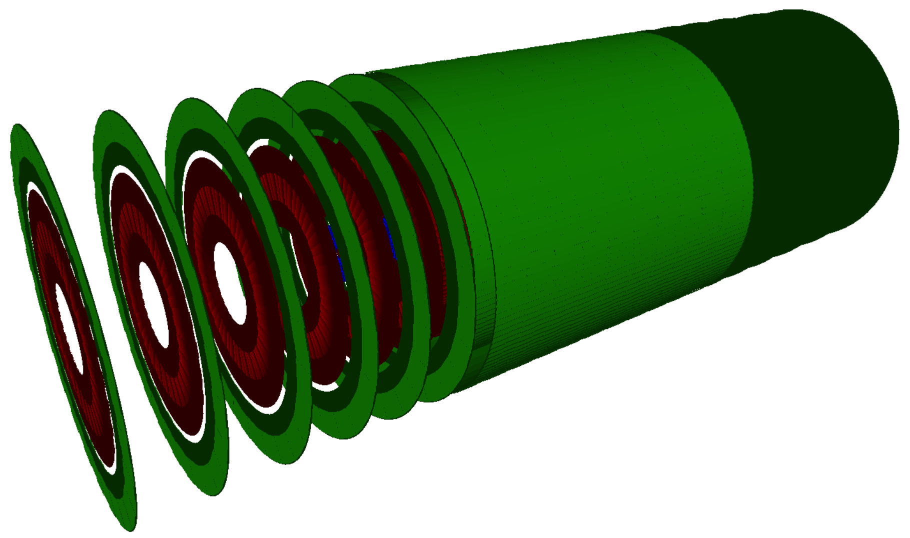
Need development such that **significant LHCb payloads can run on GPUs**

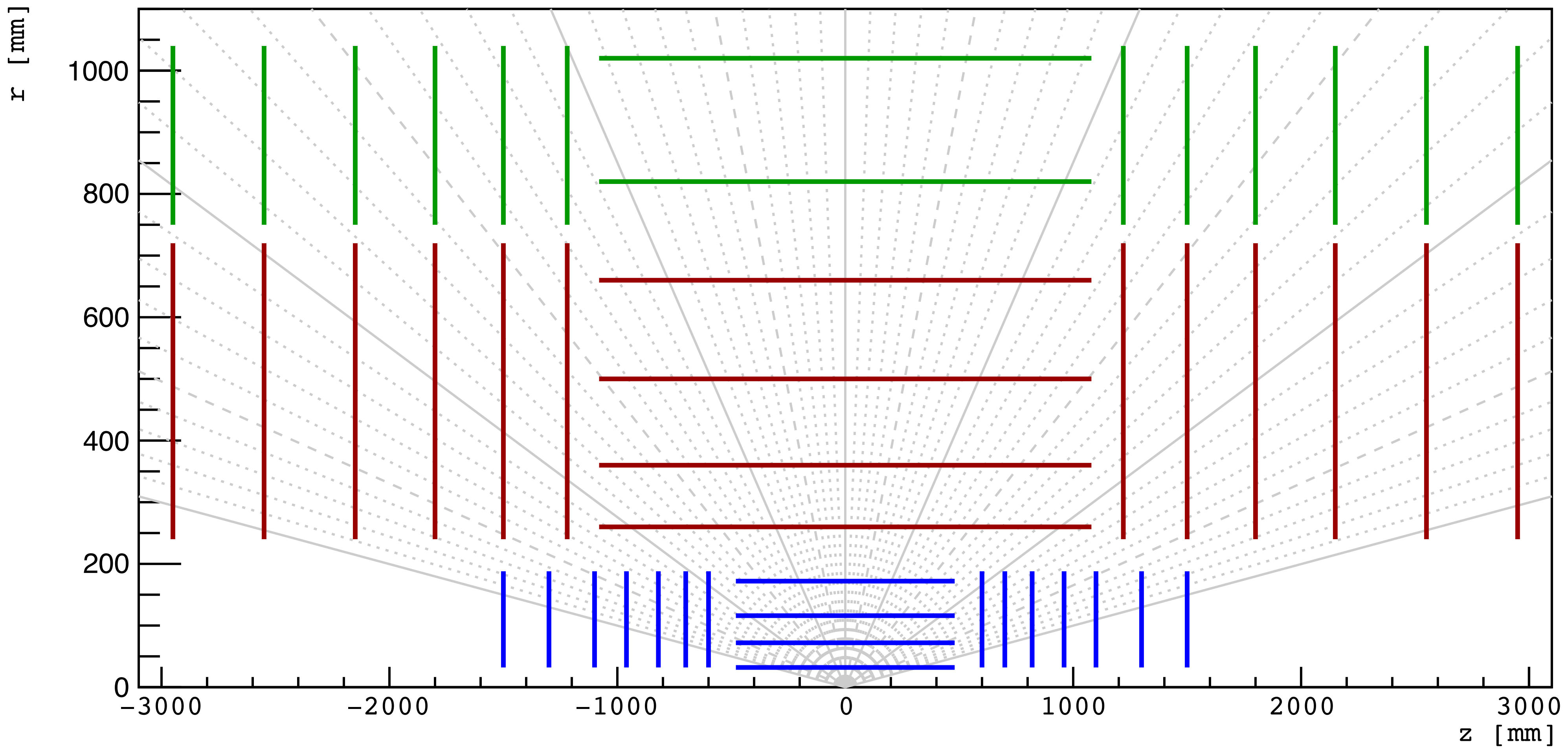
- User analysis utilising eg. TensorFlow for ML and fitting but small share of LHCb's CPU
- Full detector simulation main payload but Geant4 has no GPU compatibility yet (work ongoing outside LHCb)

GPU batch cluster at CERN to develop/run GPU workflows

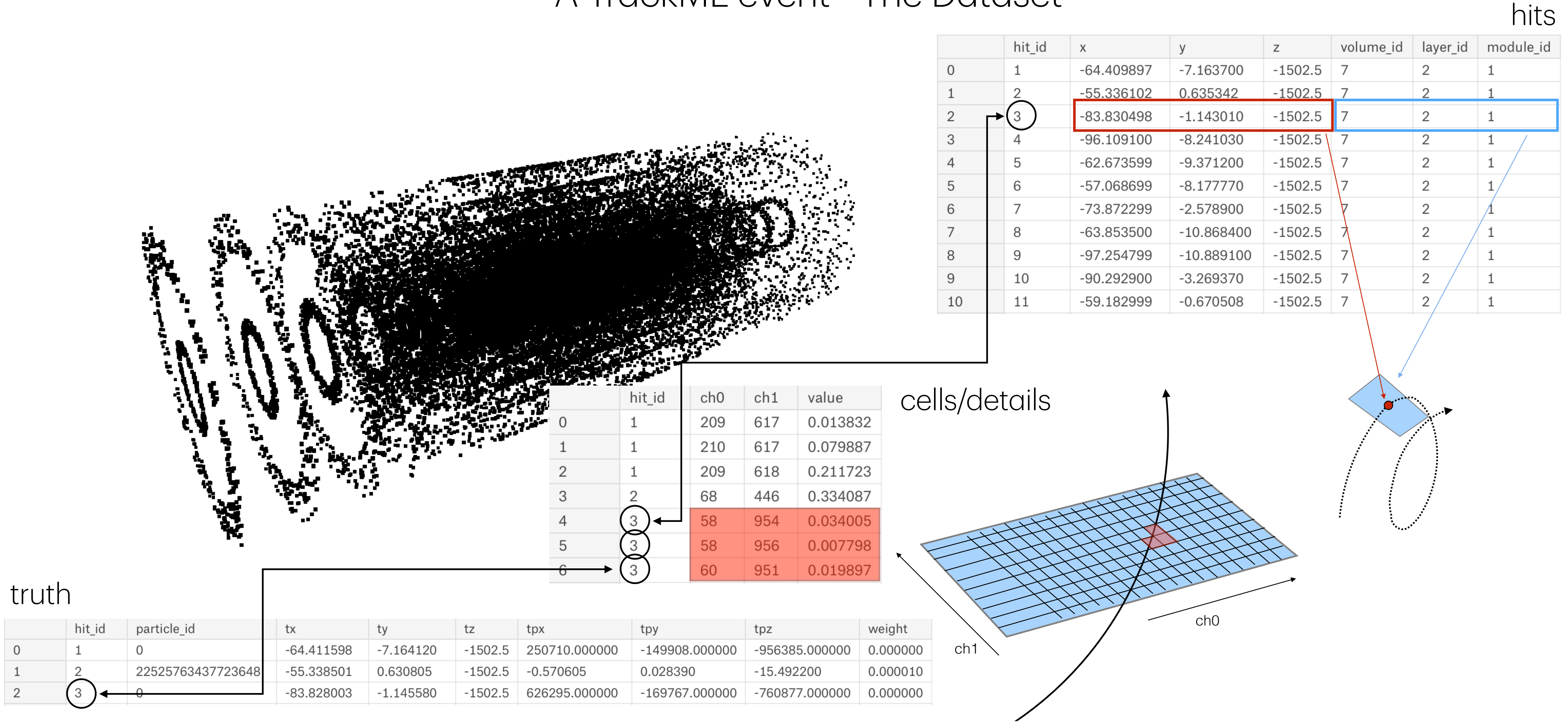


A. Salzburger, Friday





A TrackML event - The Dataset



hits

	hit_id	x	y	z	volume_id	layer_id	module_id
0	1	-64.409897	-7.163700	-1502.5	7	2	1
1	2	-55.336102	0.635342	-1502.5	7	2	1
2	3	-83.830498	-1.143010	-1502.5	7	2	1
3	4	-96.109100	-8.241030	-1502.5	7	2	1
4	5	-62.673599	-9.371200	-1502.5	7	2	1
5	6	-57.068699	-8.177770	-1502.5	7	2	1
6	7	-73.872299	-2.578900	-1502.5	7	2	1
7	8	-63.853500	-10.868400	-1502.5	7	2	1
8	9	-97.254799	-10.889100	-1502.5	7	2	1
9	10	-90.292900	-3.269370	-1502.5	7	2	1
10	11	-59.182999	-0.670508	-1502.5	7	2	1

cells/details

	hit_id	ch0	ch1	value
0	1	209	617	0.013832
1	1	210	617	0.079887
2	1	209	618	0.211723
3	2	68	446	0.334087
4	3	58	954	0.034005
5	3	58	956	0.007798
6	3	60	951	0.019897

truth

	hit_id	particle_id	tx	ty	tz	tpx	tpy	tpz	weight
0	1	0	-64.411598	-7.164120	-1502.5	250710.000000	-149908.000000	-956385.000000	0.000000
1	2	22525763437723648	-55.338501	0.630805	-1502.5	-0.570605	0.028390	-15.492200	0.000010
2	3	0	-83.828003	-1.145580	-1502.5	626295.000000	-169767.000000	-760877.000000	0.000000

13

Phase 1

focus on accuracy

Apr 13, 2018 → Aug 14, 2018

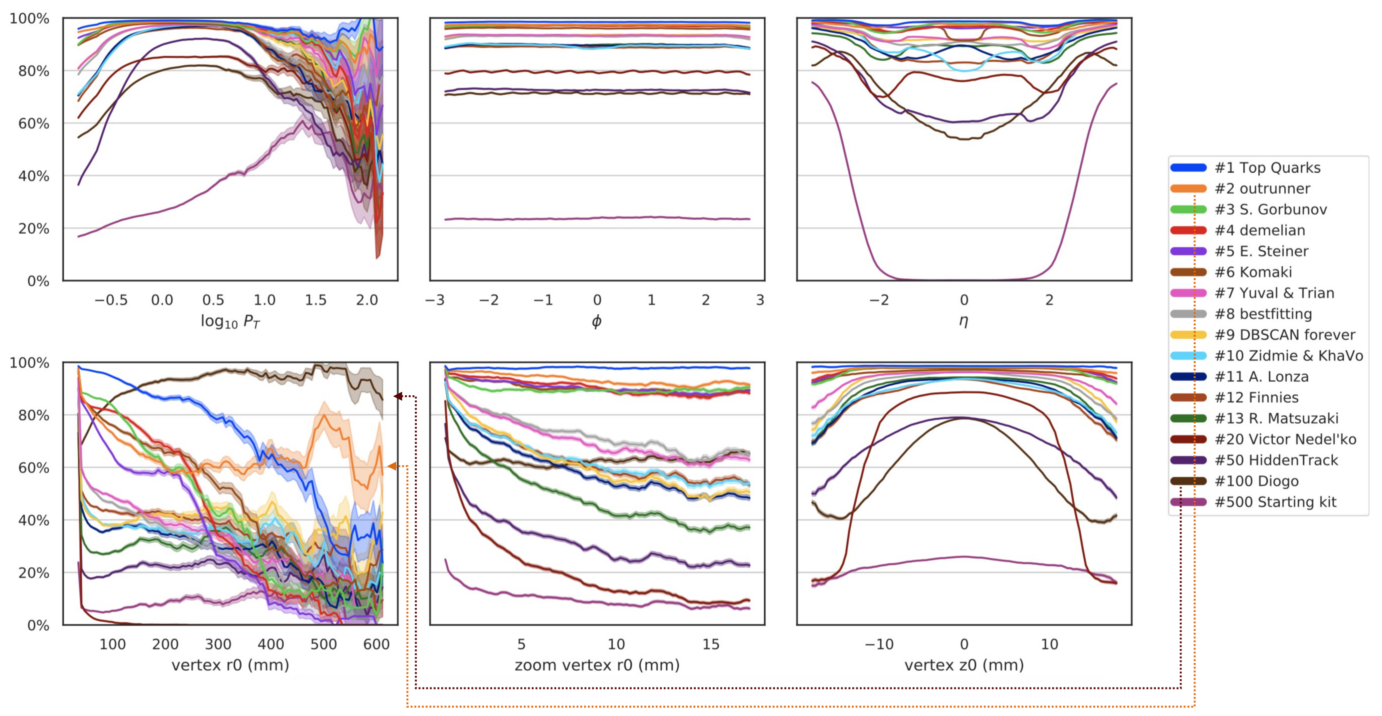
Phase 2

accuracy & speed

Sep 07, 2018 → Nov 12, 2018

Phase 1 Aftermath - Tracking Efficiency

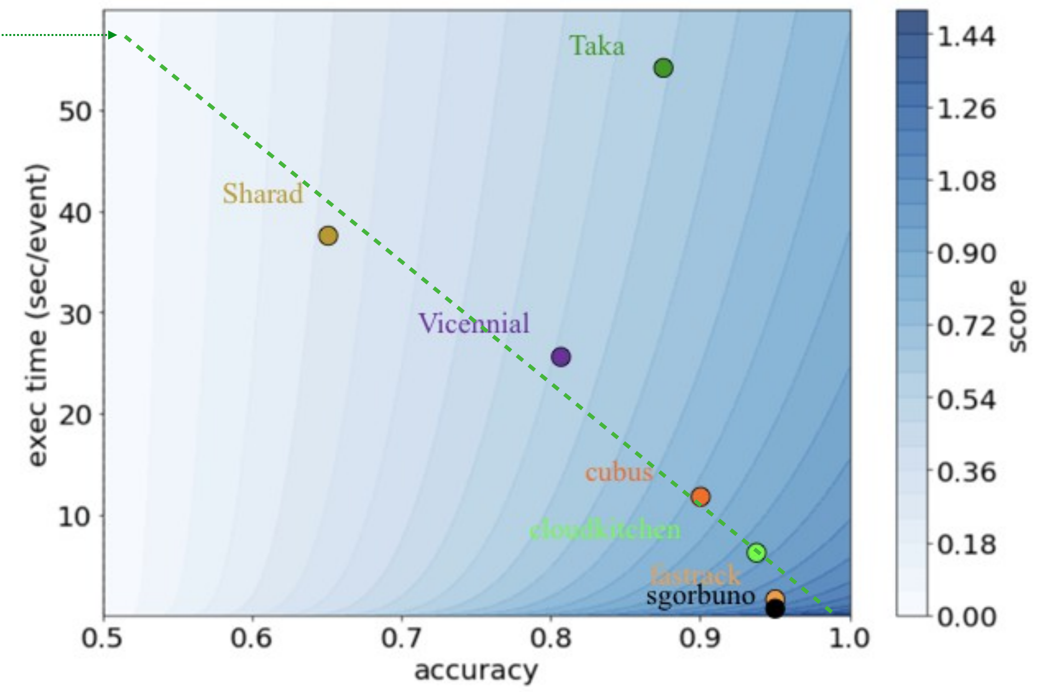
High score means High tracking efficiency



Phase 2 - Final Score Map

Correlation accuracy / speed

Fastest solution are in general also the most accurate one!

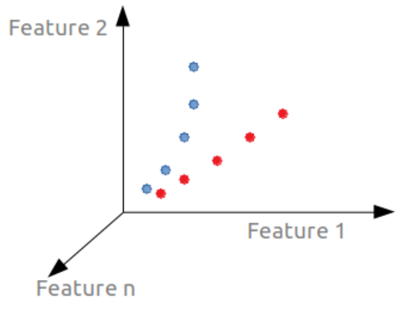


Phase 1 DL Prize

Seeding → Inference & Assembling → Fitting

Seeding

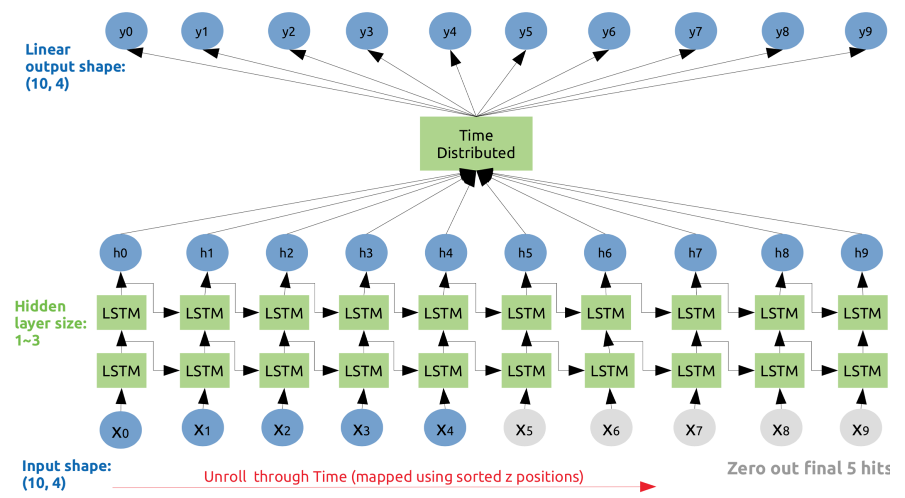
DBScan



Inference & Assembling

Recurrent NN

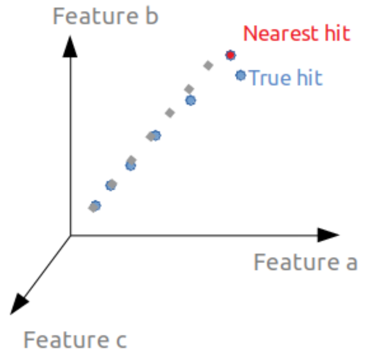
10 output sets of hit quadruplets



$(\phi, r, z, z/r)$ 5 sets of hit quadruplets

Fitting

KNN-tree



Authors L. & N. Finnie

25

A. Yzquierdo, Friday

Motivation for the use of HPC resources

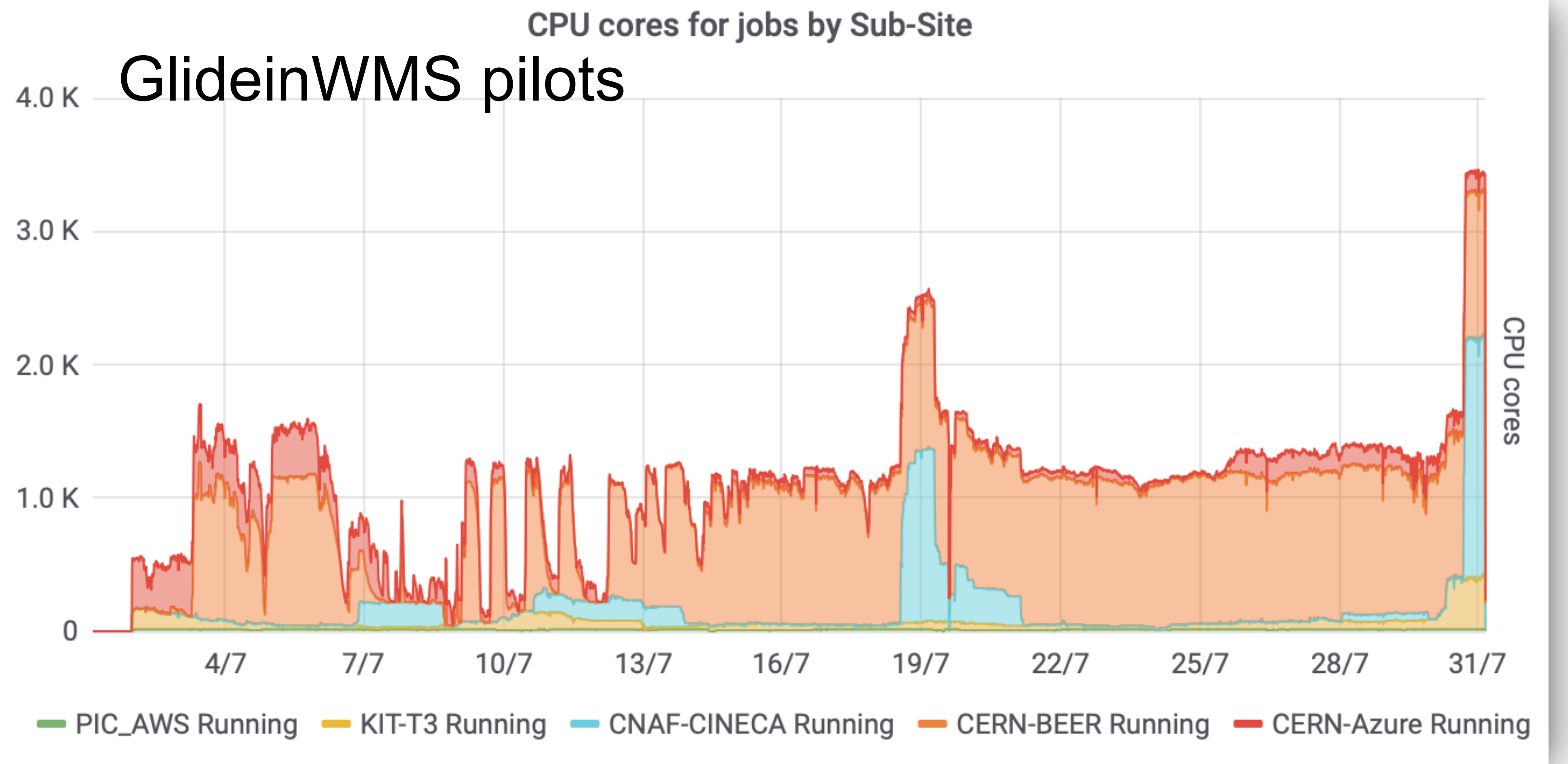
CMS aims towards **increasing the usage of HPC resources** in the mid to long term future (Run3 & HL-LHC):

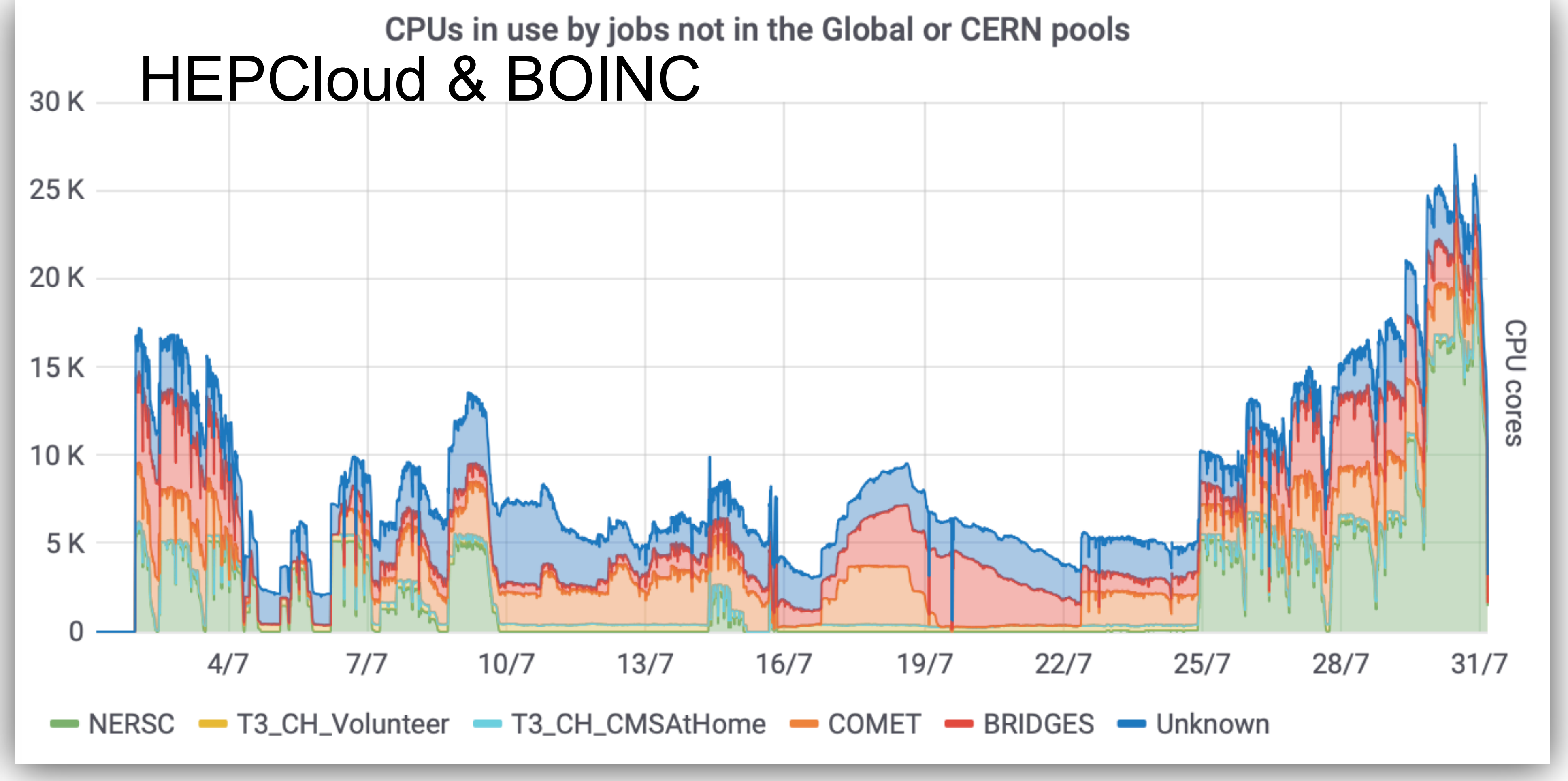
- **Growing funding** in HPC infrastructures looking onwards to **deploying Exascale machines**
- **Countries/Funding agencies** pushing HEP communities to **make use of these resources**
- Interest in HEP experiments to **access best technologies available**, usually employed at HPC sites
- HPC contribution in the future regarded as integral part of **WLCG strategy towards HL-LHC**

Recent progress in the integration of **new resources** into the CMS Global Pool and for CMS use:

- **HPC**: via GlideinWMS pilot submission (CINECA) or integrated to HEPCloud (NERSC, etc)
- **Cloud**: as extension of Grid sites (CERN_Azure and PIC_AWS)
- **Opportunistic use of clusters**: CERN_BEER and extended Research or University campus (e.g. KIT_T3 and at Purdue)
- **CMS@Home** jobs in a separated **Volunteer pool**

Non-standard resources require **enhanced workload-to-resource matchmaking**: working on an **expanded description** of jobs and resources for flexible and efficient scheduling (e.g. select no input data tasks, suitable job processing time in KNL nodes, etc.)


GlideinWMS pilots


HEPCloud & BOINC

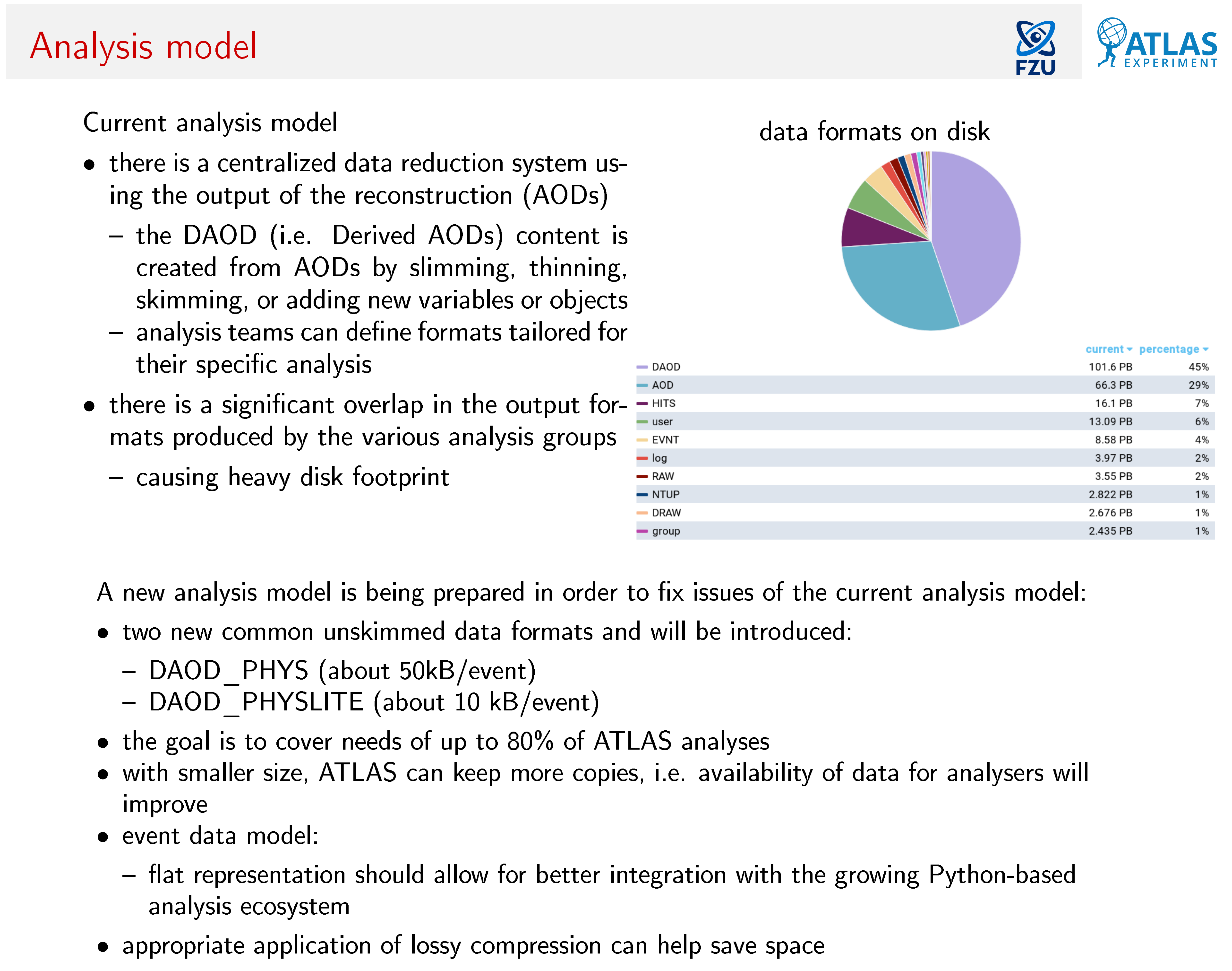
Analysis challenge

Storage and Data Delivery challenge

Analysis models

- Evolution of analysis models towards HL-LHC is to smaller and flatter data formats
  - e.g. nanoAOD for CMS, DAOD_PHYSLITE for ATLAS
- Increasing interest in high-speed delivery of columnar data to physicists in real time via Spark or similar technologies, with analysis in workbooks
- Challenge: integrating this with our distributed computing infrastructure
- Concept of Analysis Facilities within existing grid sites is gaining interest

M. Svatoš, Friday



Analysis model

FZU ATLAS EXPERIMENT

Current analysis model

- there is a centralized data reduction system using the output of the reconstruction (AODs)
  - the DAOD (i.e. Derived AODs) content is created from AODs by slimming, thinning, skimming, or adding new variables or objects
  - analysis teams can define formats tailored for their specific analysis
- there is a significant overlap in the output formats produced by the various analysis groups
  - causing heavy disk footprint

data formats on disk

	current ▾	percentage ▾
DAOD	101.6 PB	45%
AOD	66.3 PB	29%
HITS	16.1 PB	7%
user	13.09 PB	6%
EVNT	8.58 PB	4%
log	3.97 PB	2%
RAW	3.55 PB	2%
NTUP	2.822 PB	1%
DRAW	2.676 PB	1%
group	2.435 PB	1%

A new analysis model is being prepared in order to fix issues of the current analysis model:

- two new common unskimmed data formats and will be introduced:
  - DAOD_PHYS (about 50kB/event)
  - DAOD_PHYSLITE (about 10 kB/event)
- the goal is to cover needs of up to 80% of ATLAS analyses
- with smaller size, ATLAS can keep more copies, i.e. availability of data for analysers will improve
- event data model:
  - flat representation should allow for better integration with the growing Python-based analysis ecosystem
- appropriate application of lossy compression can help save space

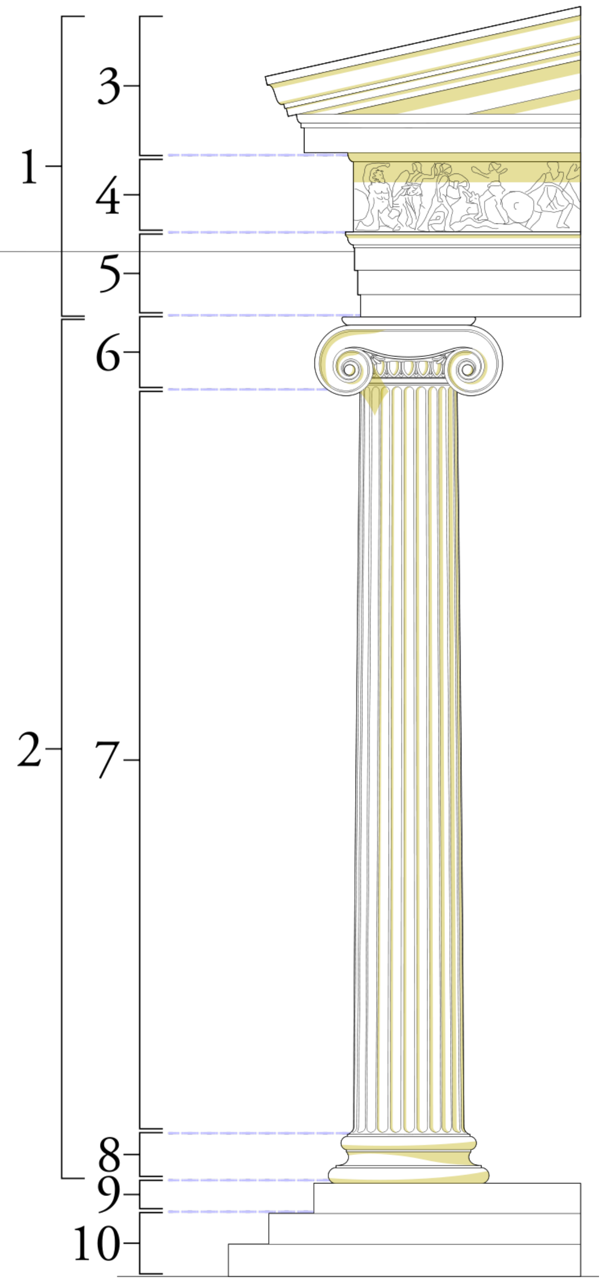
A. Naumann, Tuesday

Introducing ROOT

- ROOT is a centerpiece of HEP, virtually every HEPicist uses ROOT for analysis, > 1 exabyte of data in ROOT format
- Common (also graphics) language, common data format, common grounds
- Coherently designed, integrated solution with optimized interplay
- Core in C++, with dynamic Python bindings

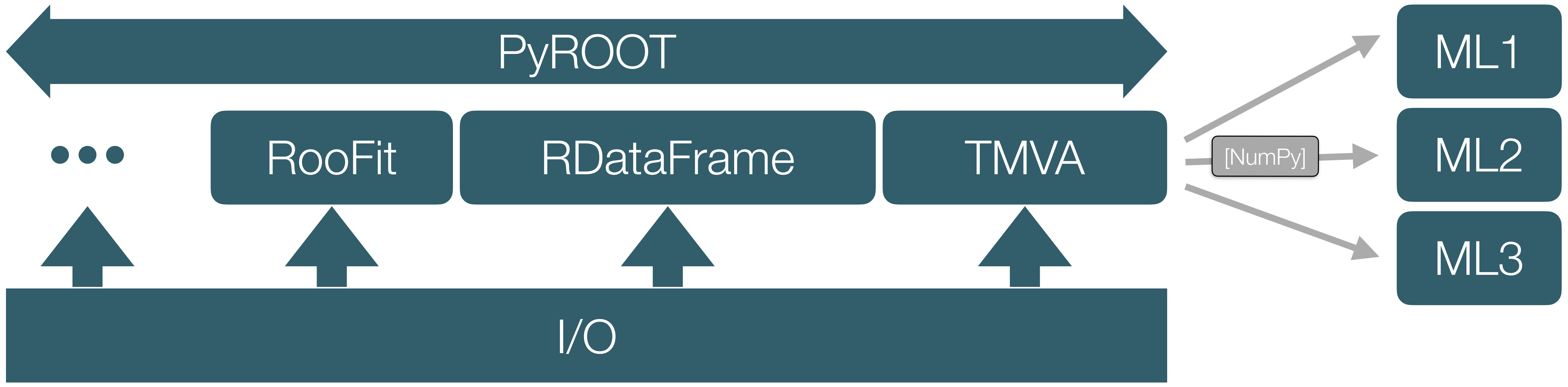
"ROOT7"

- Massive, multi-year development effort
- Focused on main ROOT columns:
  - Analysis: parallelism, Python, RDataFrame, RooFit, TMVA
  - I/O: TTree successor RNtuple
  - Graphics: web-based graphics, GUI, event display
  - Foundational math: histograms

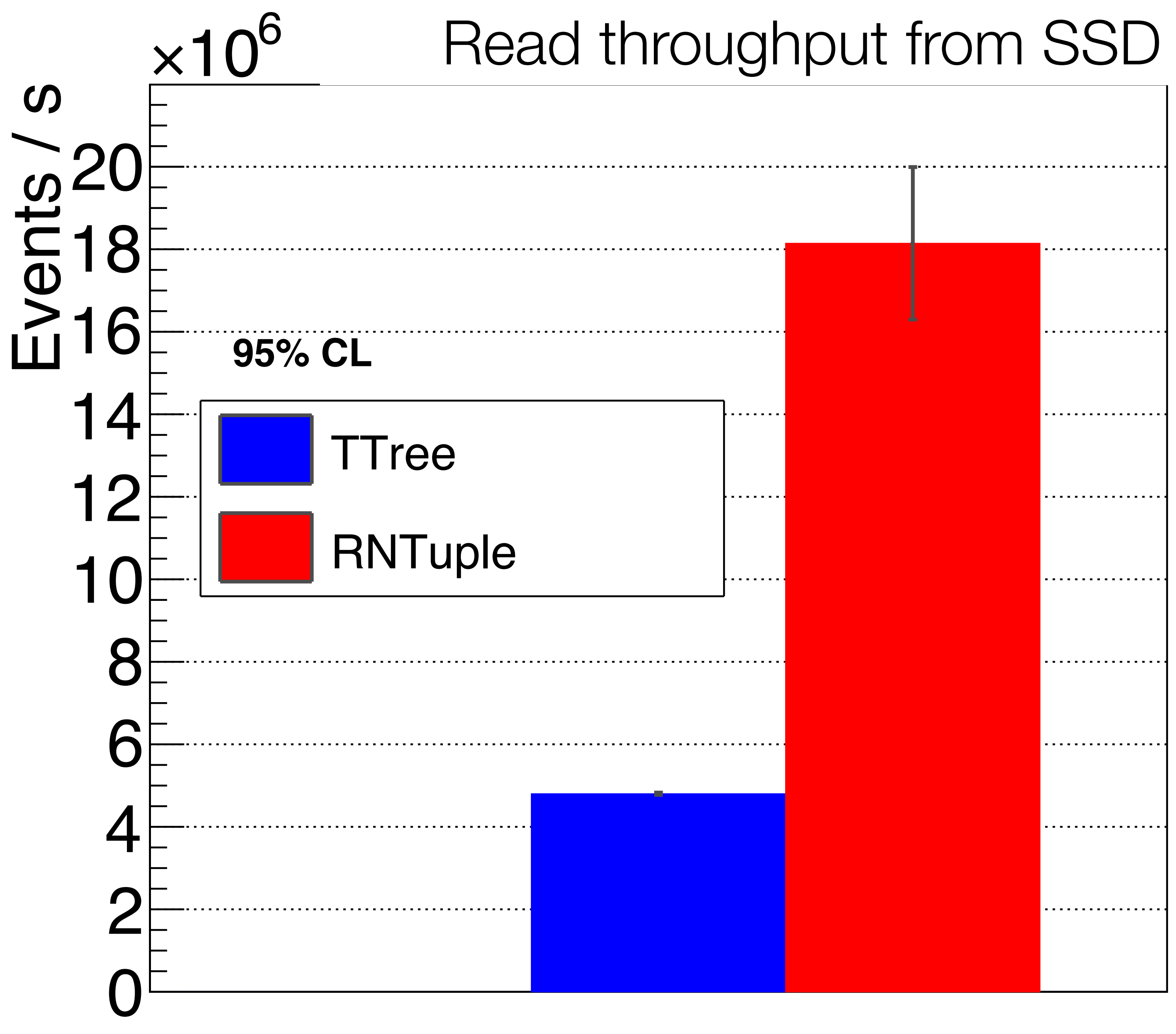


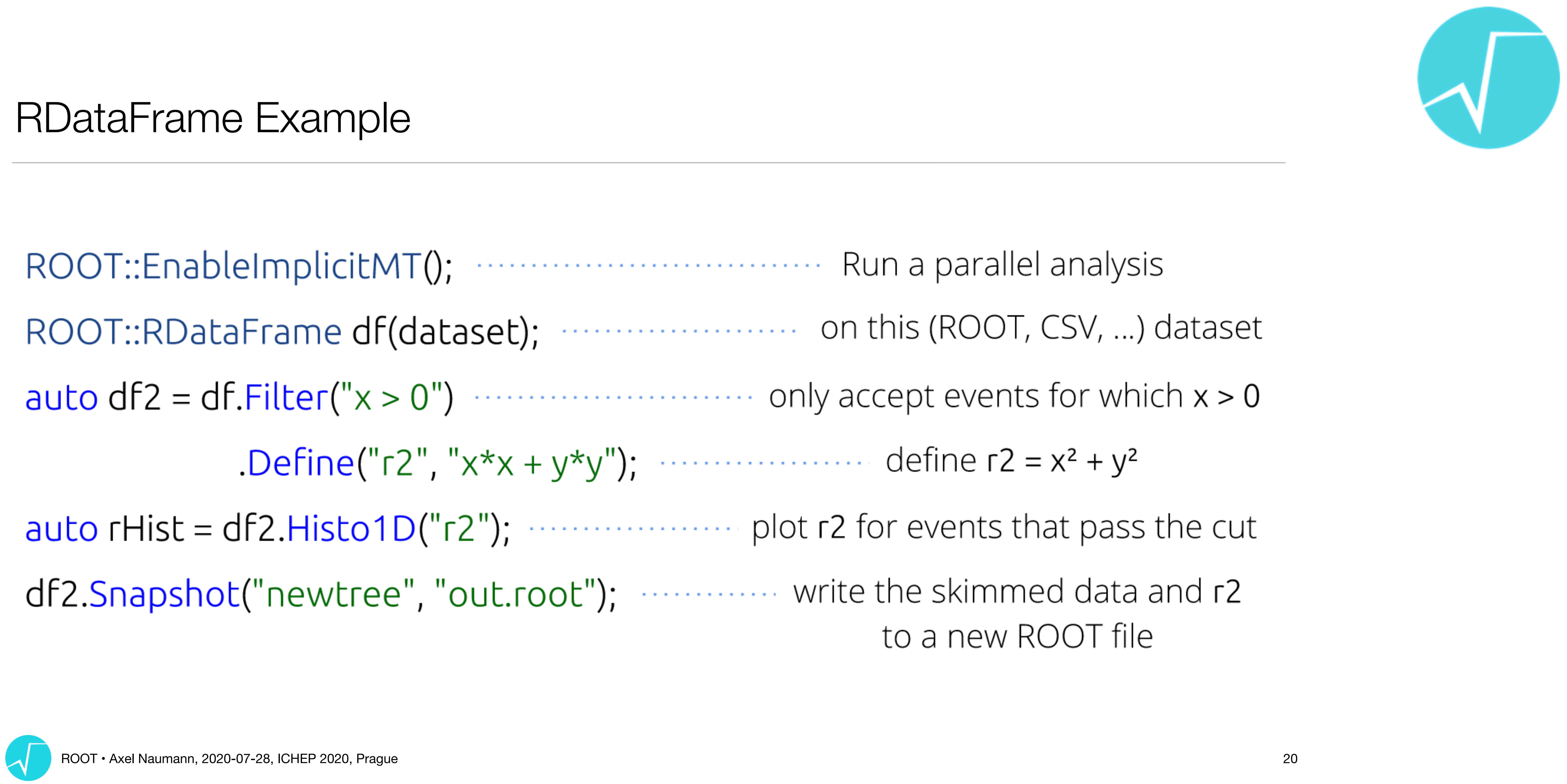
Why to bet on ROOT

- Targeted for HEP: simplicity, efficiency, support
- Allows to predict changes, adapt and benefit: solutions and R&D tailored to our very own problems
- Interface with and learn from other tools
- Single point of improvement: contribute here to have an impact, coherency and synergies (experiment vs analysis etc) guaranteed
- Advantage: community knows its challenges; gets a coherent, reliable, performant and agreed solution



A. Naumann, Tuesday





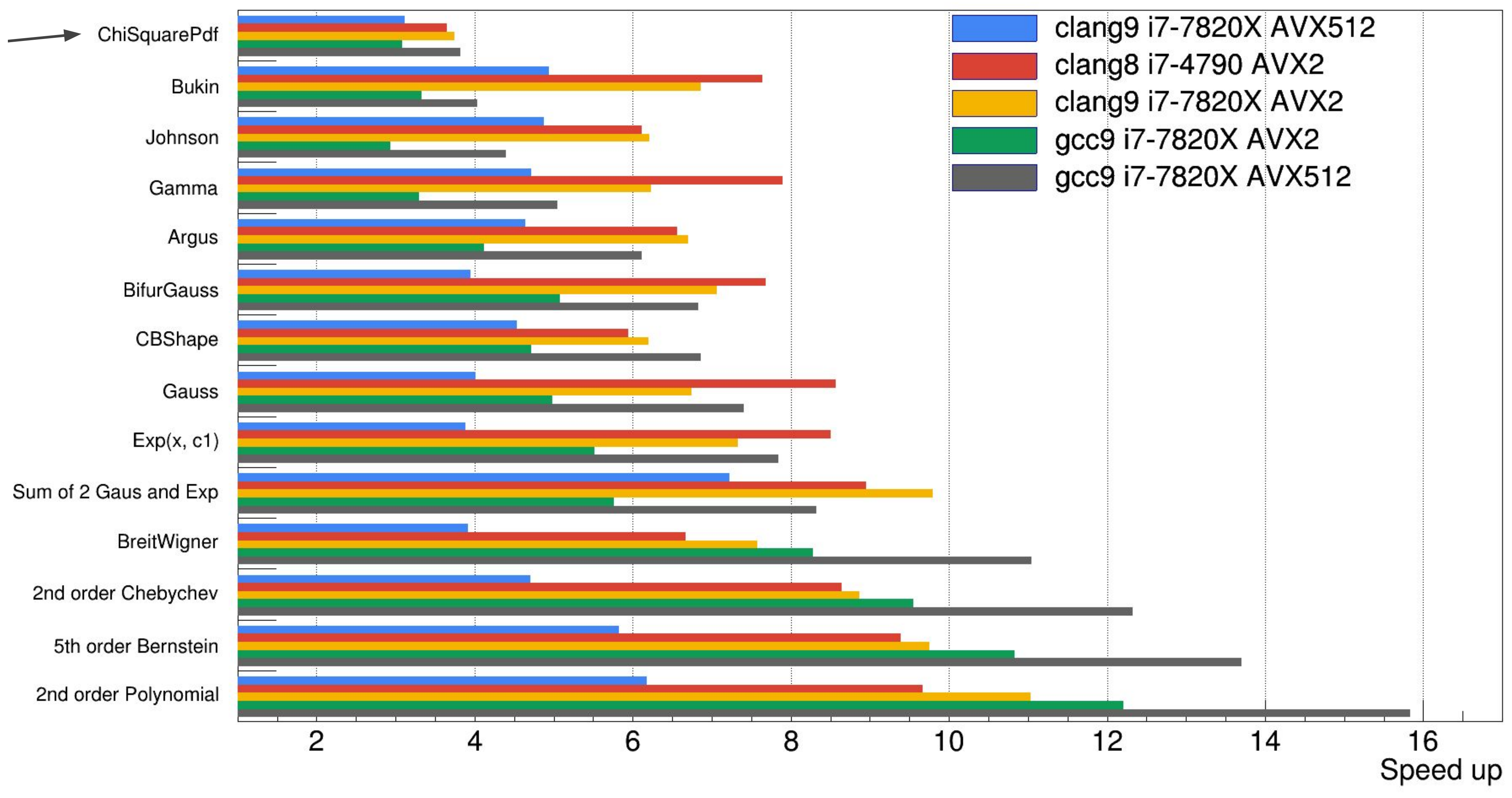
RDataFrame Example

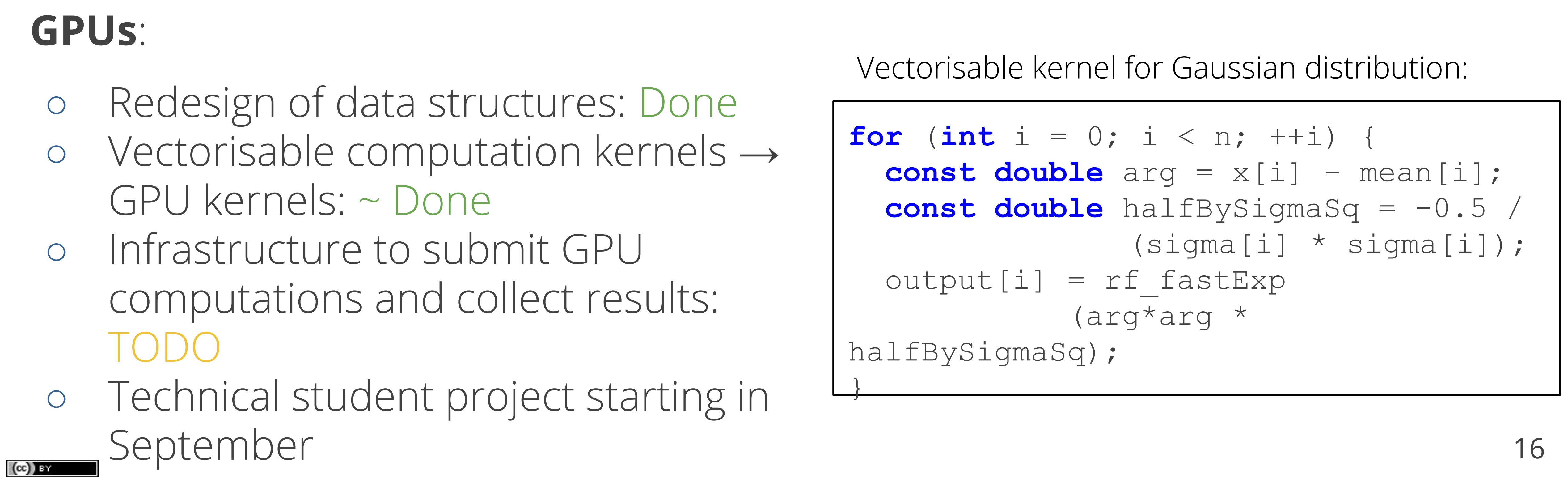
```
ROOT::EnableImplicitMT();  ··········  Run a parallel analysis
ROOT::RDataFrame df(dataset);  ········  on this (ROOT, CSV, ...) dataset
auto df2 = df.Filter("x > 0")  ········  only accept events for which x > 0
              .Define("r2", "x*x + y*y");  ·····  define r2 = x² + y²
auto rHist = df2.Histo1D("r2");  ······  plot r2 for events that pass the cut
df2.Snapshot("newtree", "out.root");  ····  write the skimmed data and r2
                                              to a new ROOT file
```

ROOT • Axel Naumann, 2020-07-28, ICHEP 2020, Prague 20

S. Hageboeck, Tuesday

Speed up using vectorisation





GPUs:

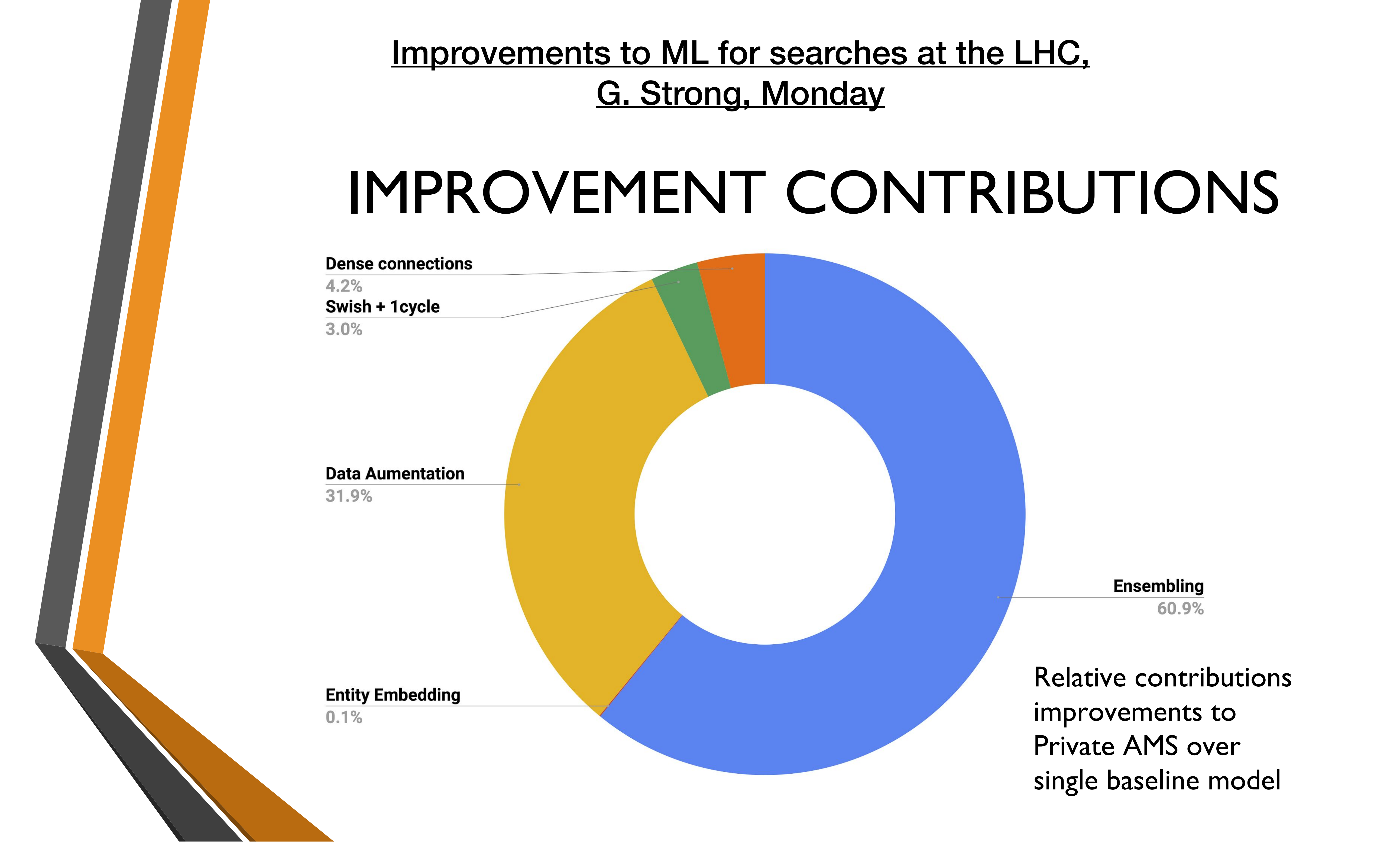
- ○ Redesign of data structures: Done
- ○ Vectorisable computation kernels → GPU kernels: ~ Done
- ○ Infrastructure to submit GPU computations and collect results: TODO
- ○ Technical student project starting in September

Vectorisable kernel for Gaussian distribution:

```
for (int i = 0; i < n; ++i) {
  const double arg = x[i] - mean[i];
  const double halfBySigmaSq = -0.5 /
              (sigma[i] * sigma[i]);
  output[i] = rf_fastExp
           (arg*arg *
halfBySigmaSq);
}
```

16

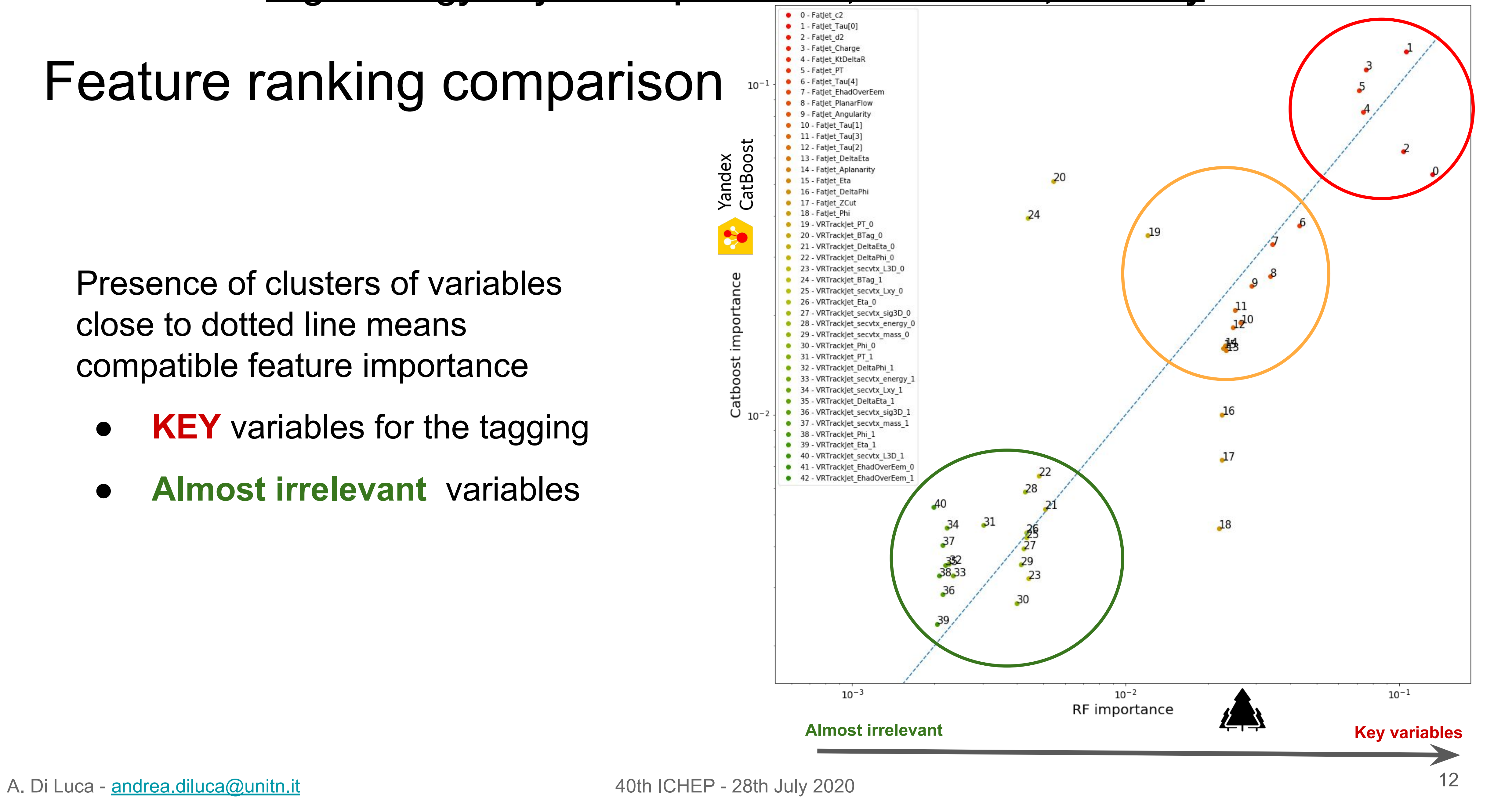
Improvements to ML for searches at the LHC, G. Strong, Monday



IMPROVEMENT CONTRIBUTIONS

Relative contributions improvements to Private AMS over single baseline model

Automated selection of particle-jet features for data analysis in High Energy Physics experiments, A. Di Luca, Monday



Feature ranking comparison

Presence of clusters of variables close to dotted line means compatible feature importance

- **KEY** variables for the tagging
- **Almost irrelevant** variables

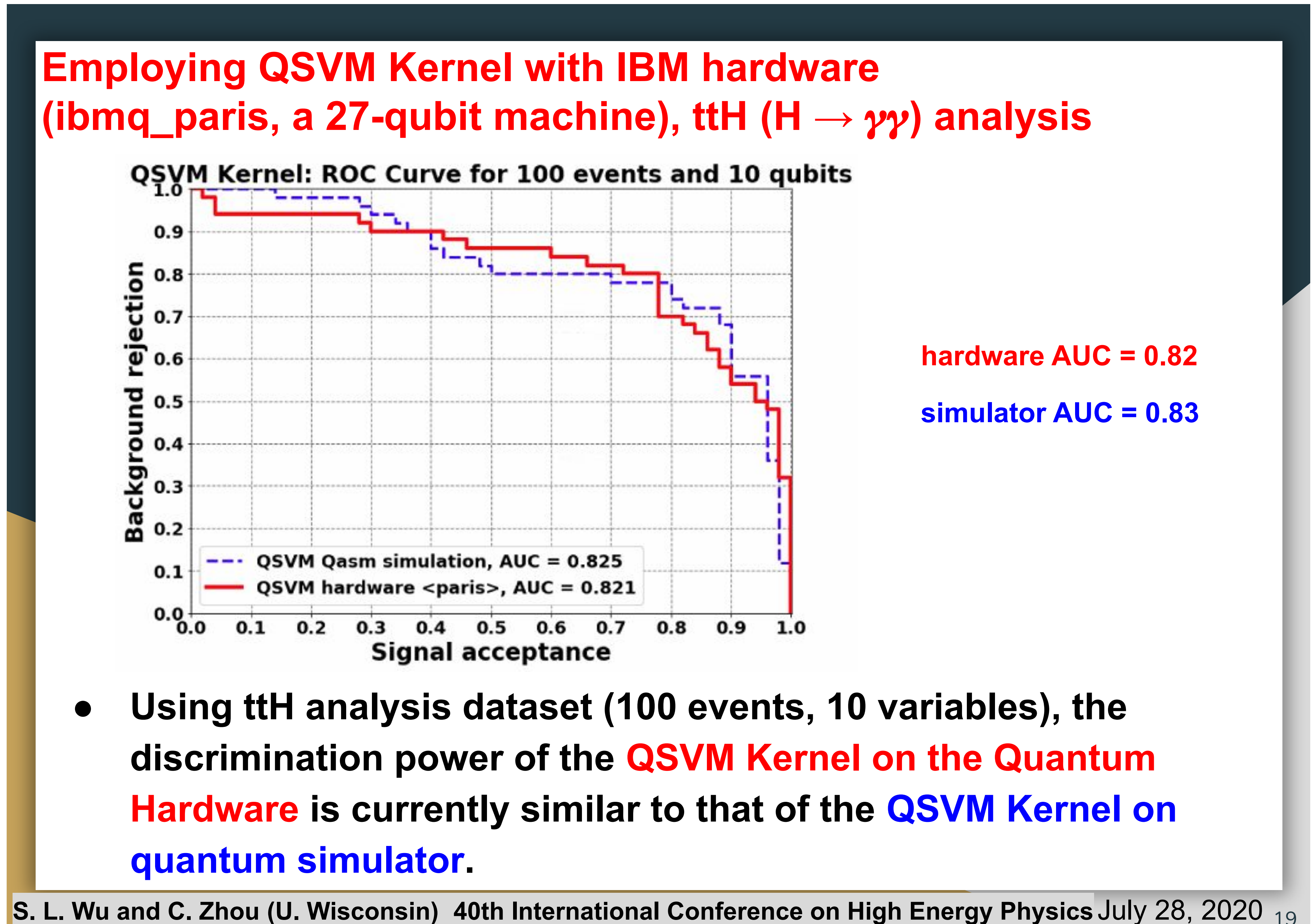
A. Di Luca - andrea.diluca@unitn.it

40th ICHEP - 28th July 2020

12

Application of Quantum Machine Learning to HEP Analysis at LHC using IBM Quantum Computer Simulators and Hardware,

C. Zhou, Tuesday



Employing QSVM Kernel with IBM hardware (ibmq_paris, a 27-qubit machine), ttH (H → $\gamma\gamma$) analysis

hardware AUC = 0.82

simulator AUC = 0.83

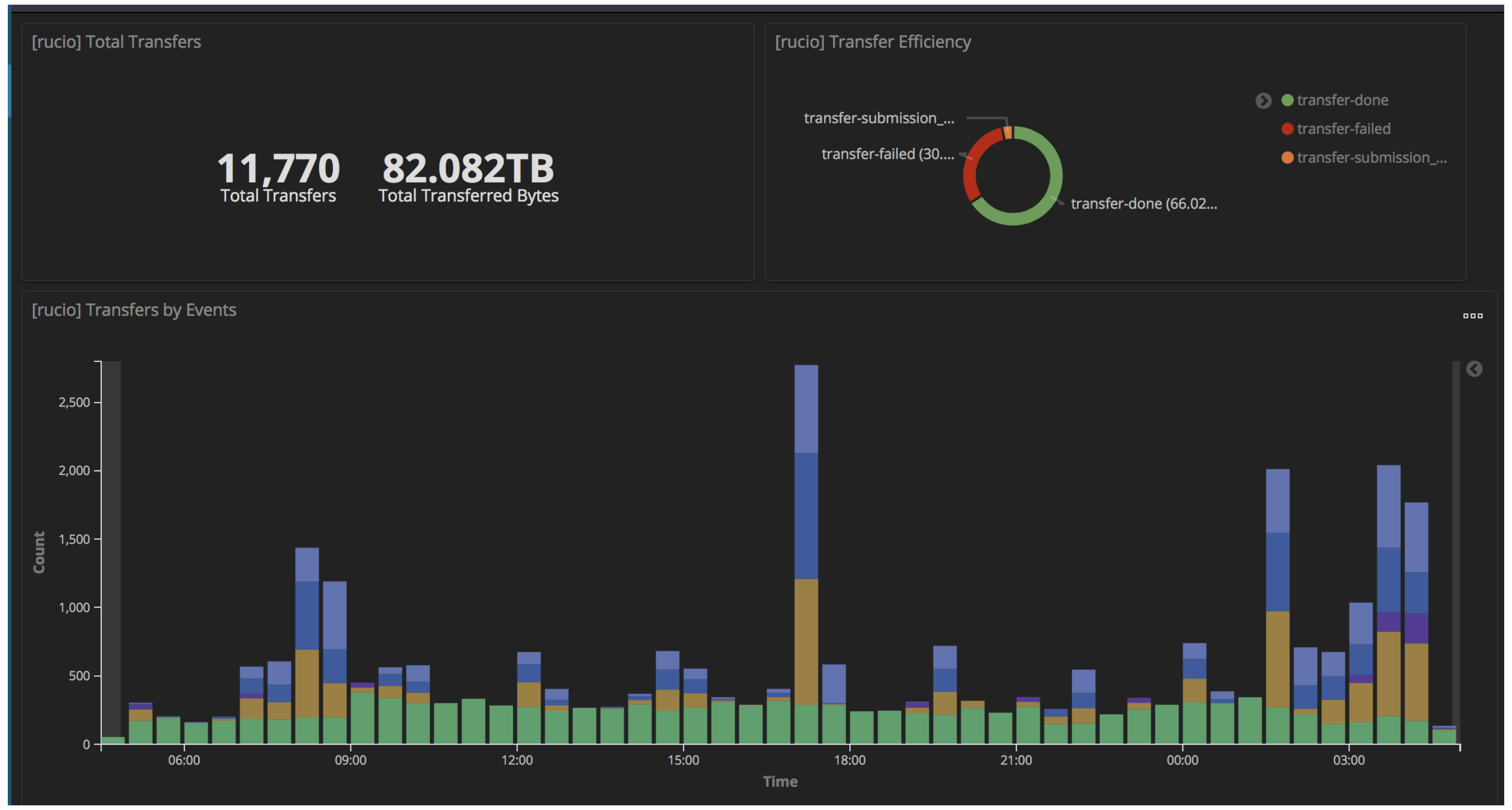
- Using ttH analysis dataset (100 events, 10 variables), the discrimination power of the QSVM Kernel on the Quantum Hardware is currently similar to that of the QSVM Kernel on quantum simulator.

S. L. Wu and C. Zhou (U. Wisconsin) 40th International Conference on High Energy Physics July 28, 2020 19

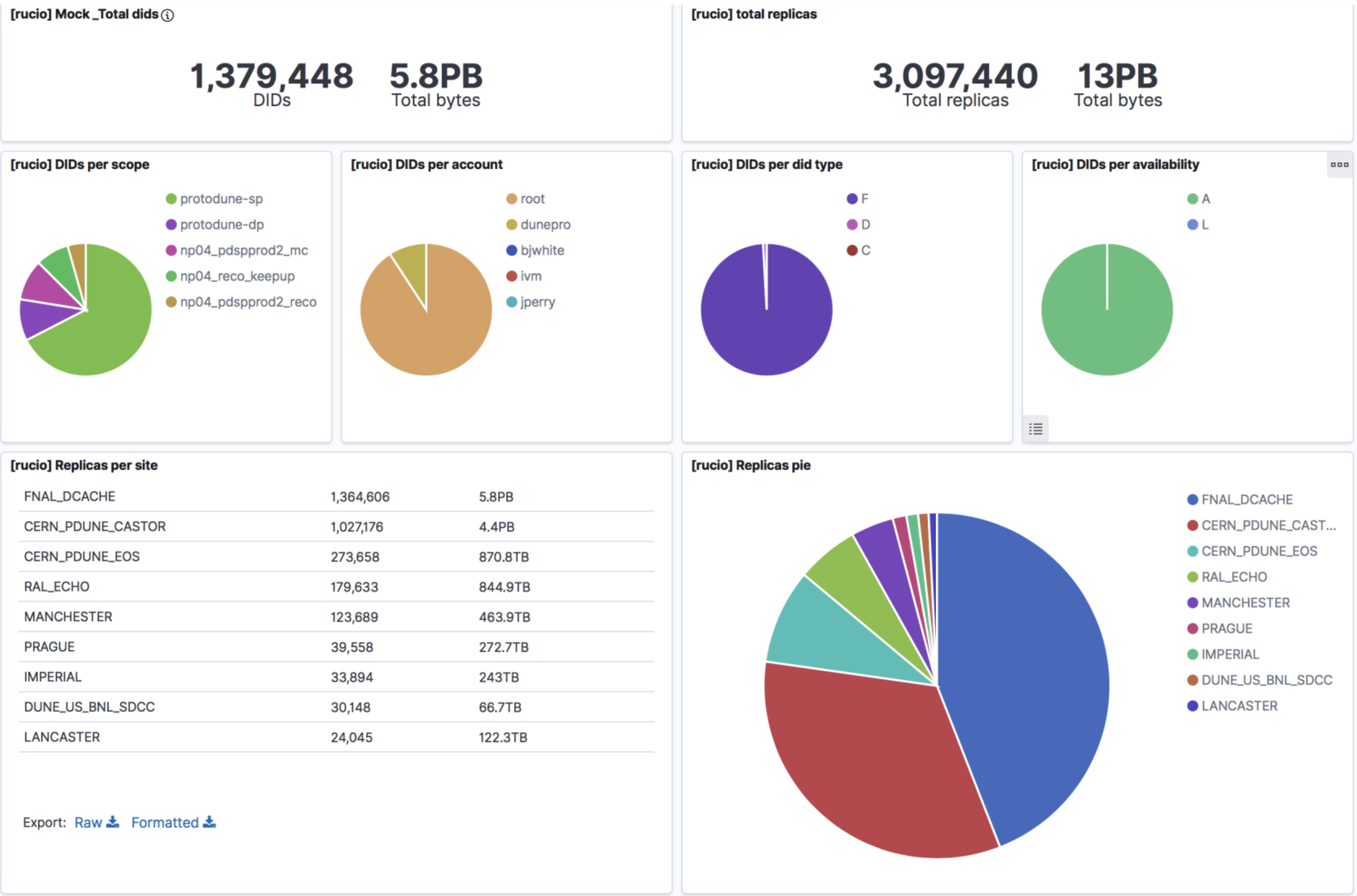
S. Timm, Wednesday

Monitoring: Transfers

- Rucio Kibana Monitoring. Shows queued, failed, submitted, done.



Monitoring: File size and location



[rucio] Replicas per site

FNAL_DCACHE	1,364,606	5.8PB
CERN_PDUNE_CASTOR	1,027,176	4.4PB
CERN_PDUNE_EOS	273,658	870.8TB
RAL_ECHO	179,633	844.9TB
MANCHESTER	123,689	463.9TB
PRAGUE	39,558	272.7TB
IMPERIAL	33,894	243TB
DUNE_US_BNL_SDCC	30,148	66.7TB
LANCASTER	24,045	122.3TB

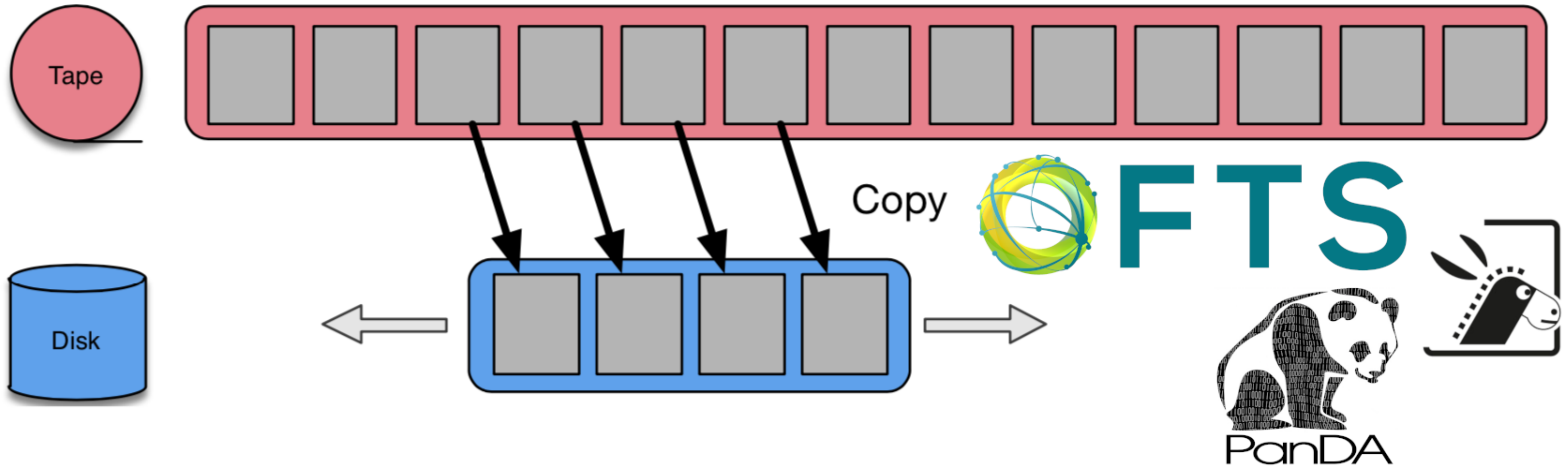
Export: Raw Formatted

10 29 July 2020 S. Timm I DUNE Data Movement Experience with Rucio

M. Svatoš, Friday

Data Carousel

- is a sliding window approach to orchestrate data processing with the majority of data resident on tape storage
- The processing is executed by staging the data onto disk storage and promptly processing them
  - only the minimum required input data are located on disk at any time
  - tested on full Run2 RAW data reprocessing (18 PB staged over several weeks rather than all at once)



Conclusions

- The HEP community has a number of challenges to address with regards to computing and software before the HL-LHC era
    - Computation, Portability, Storage & Data Delivery, Analysis
- The good news is we have tools to deal with them, as has been shown in the Computing and Data Handling track of this conference
- We also need people to do the work
    - Funding agencies and institutes must realise that computing and software is as important for physics as detector development and construction
    - The days when software grows organically with the detectors are over - writing software and building computing systems for HEP now requires detailed project planning and management, and significant person power sustained over many years
    - Stable career paths need to be defined for those who wish to stay in HEP and work on computing