Models and Algorithms for (the future of) High-Energy Physics

Maurizio Pierini

CERN

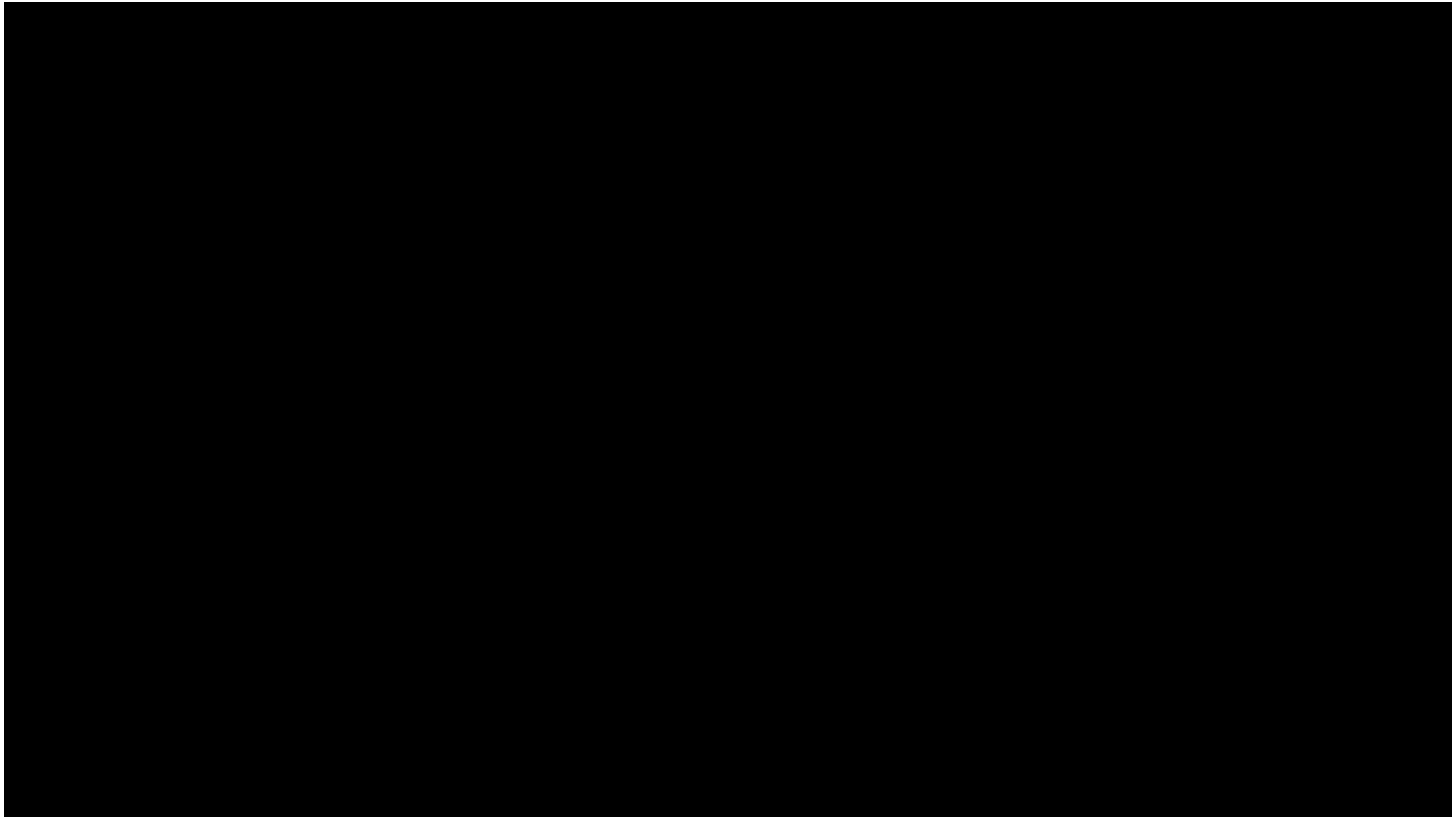
European Research Council

SOSC 2018, Perugia

In this talk

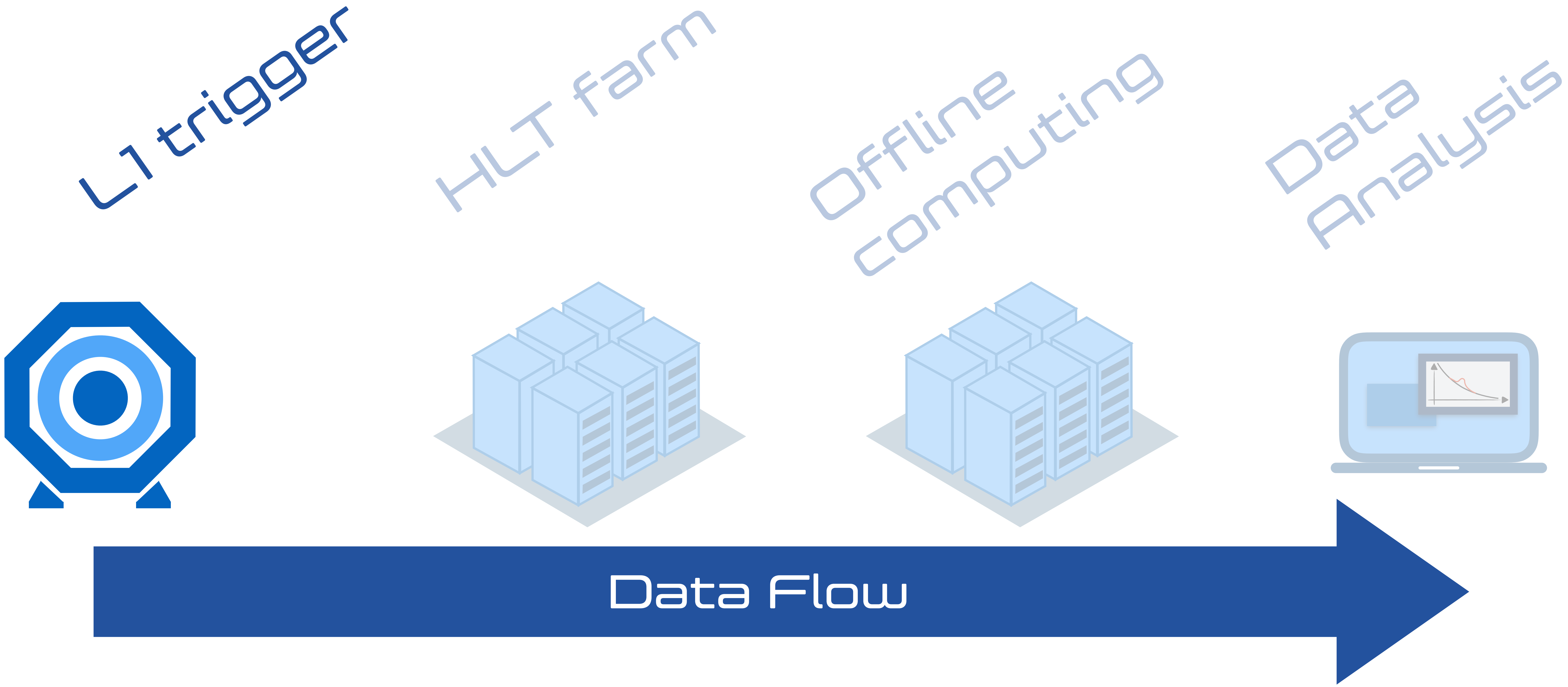
- *Big-data in real time*
  - *“data scouting” with trigger-level analysis*
- *Big-Data tools and High-Energy physics (HEP) workflows*
- *HPC centres & HEP computing workflows*
  - *opportunistic processing*
  - *distributed training for Machine Learning*

I am replacing M. Zanetti here. I am trying to follow his initial idea about this talk, but with some personal point of view.
Not sure this matches 100% what this talk was about and your expectations.
I hope this will be useful nevertheless.



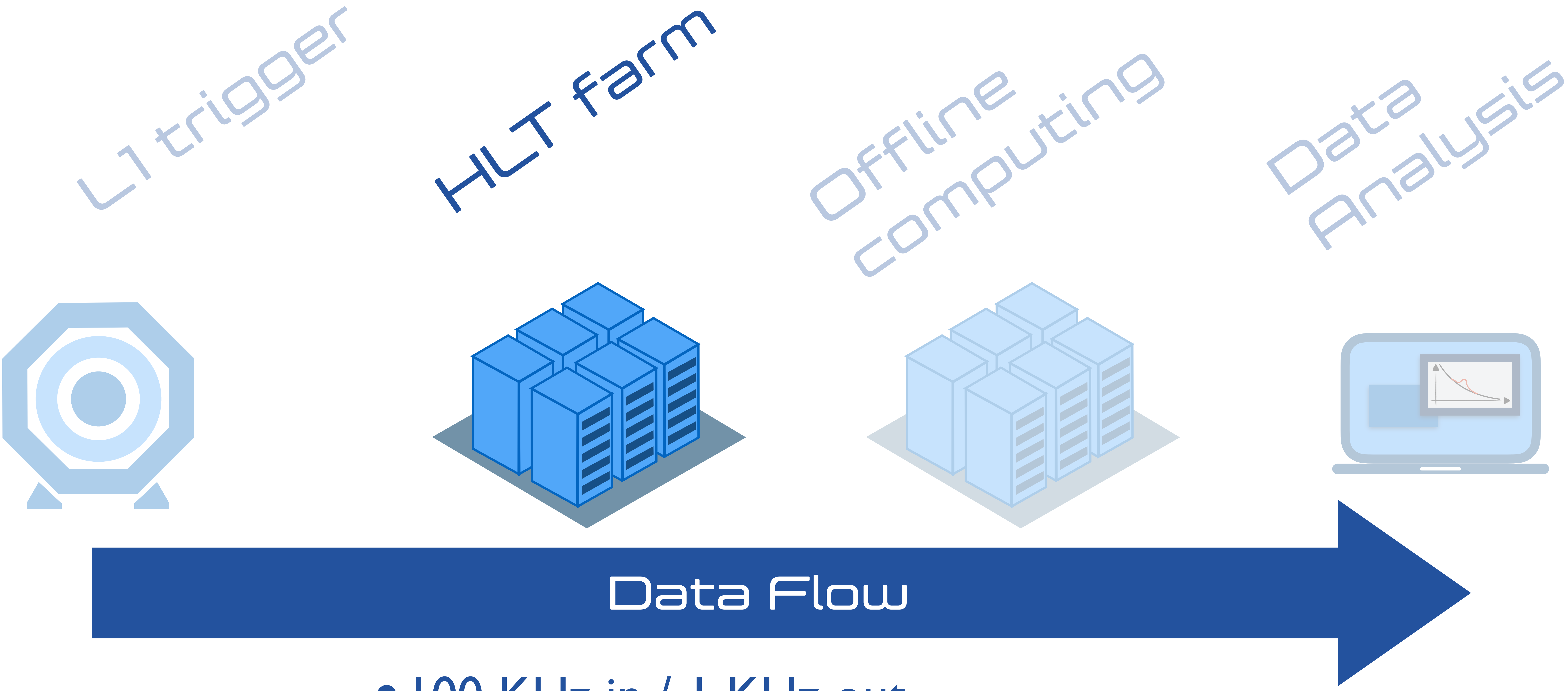
https://www.youtube.com/watch?v=jDC3-QSiLB4

The LHC Big Data problem



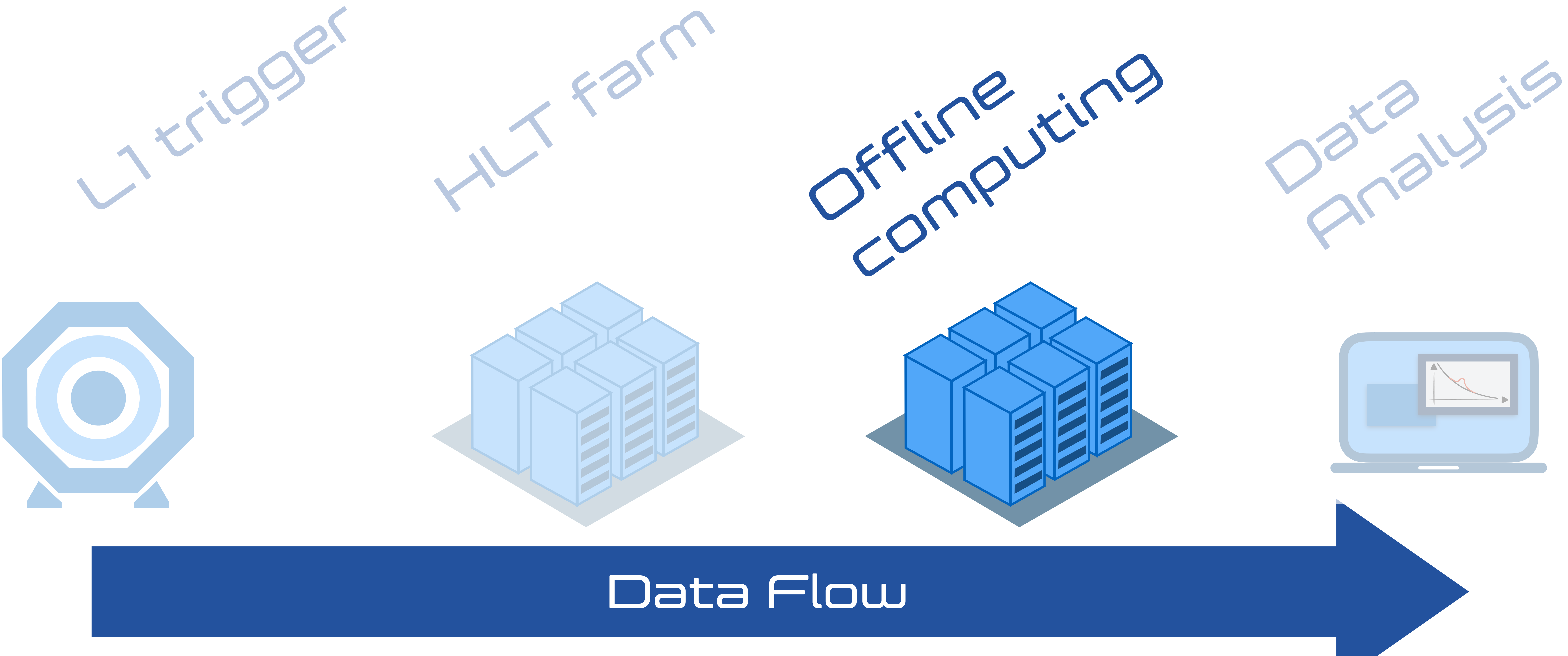
- 40 MHz in / 100 KHz out
- ~ 500 KB / event
- Processing time: ~10 μs
- Based on coarse local reconstructions
- FPGAs / Hardware implemented

The LHC Big Data problem



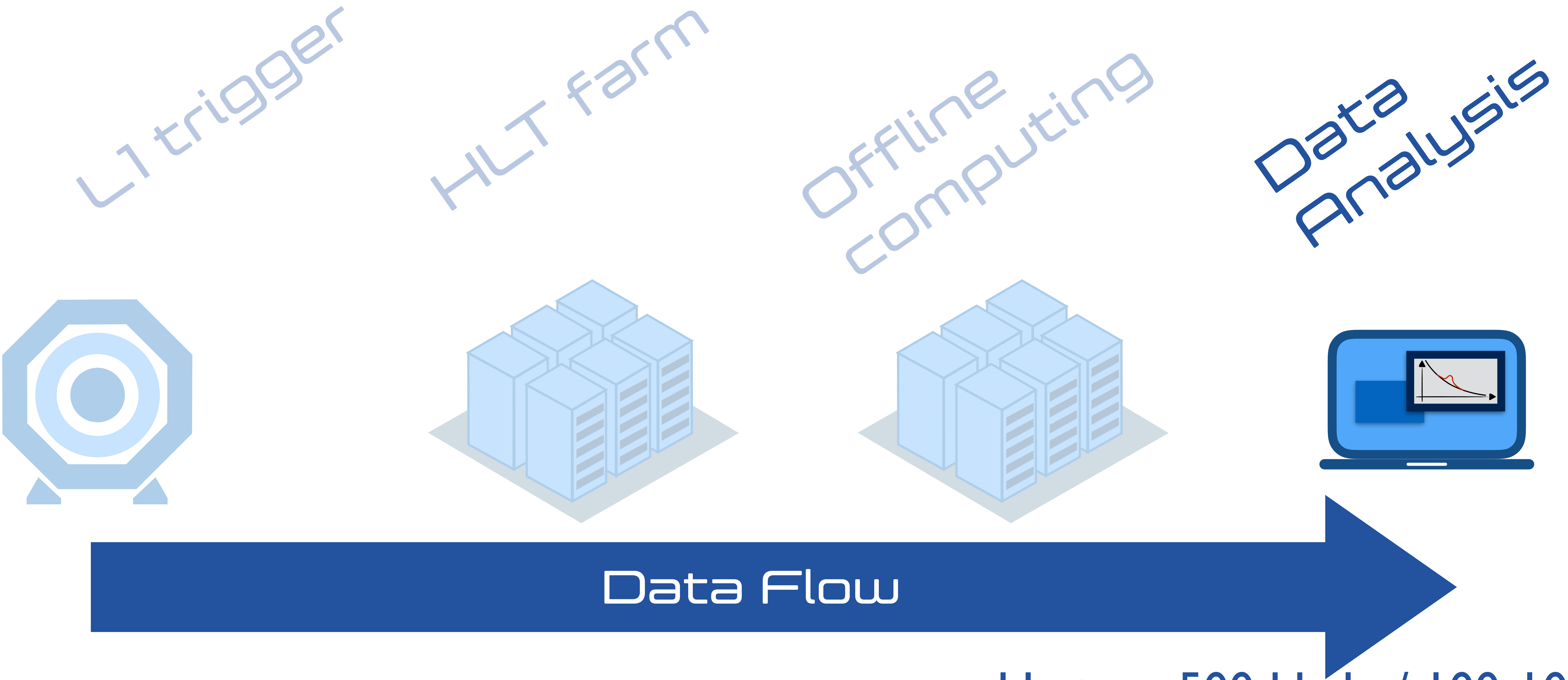
- 100 KHz in / 1 KHz out
- ~ 500 KB / event
- Processing time: ~30 ms
- Based on simplified global reconstructions
- Software implemented on CPUs

The LHC Big Data problem



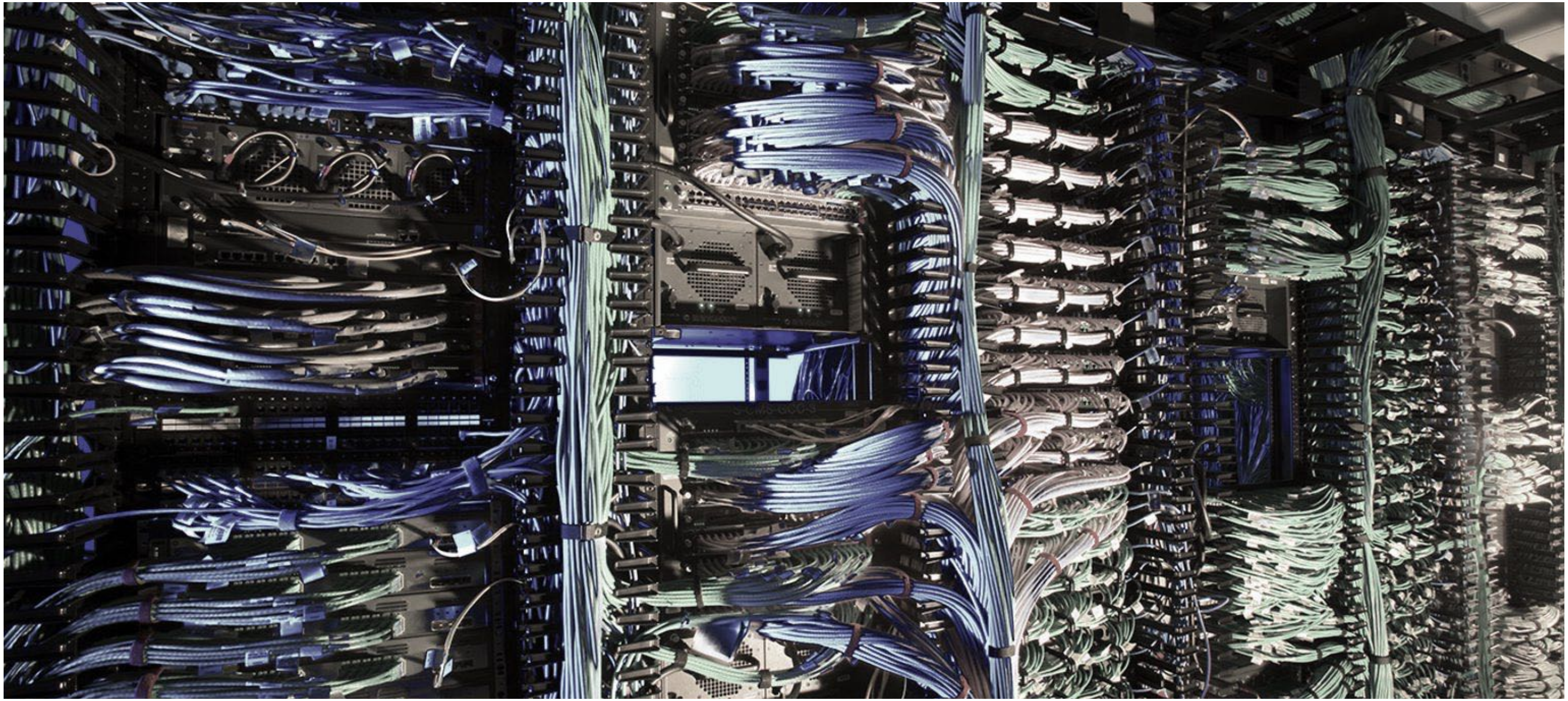
- 1 KHz in / 1.2 kHz out
- ~ 1 MB / 200 kB / 30 kB per event
- Processing time: ~20 s
- Based on accurate global reconstructions
- Software implemented on CPUs

The LHC Big Data problem



- Up to ~ 500 Hz In / 100-1000 events out

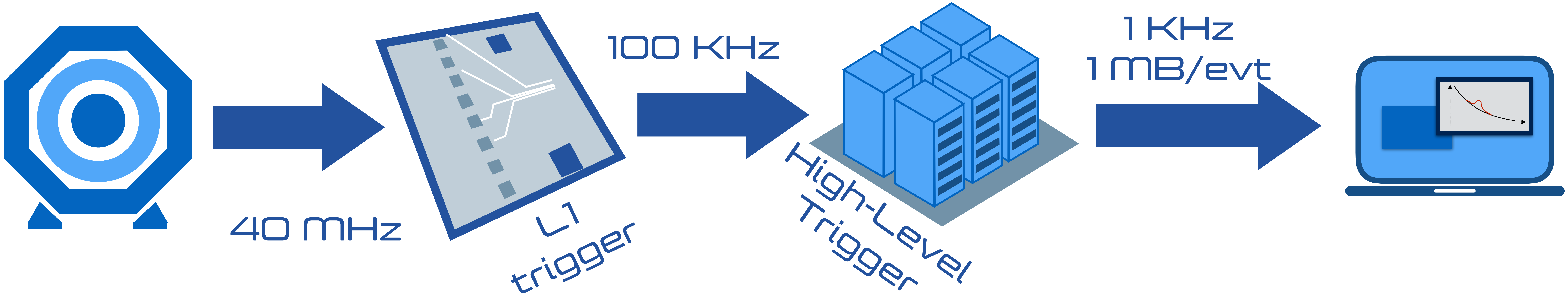
- <30 KB per event
- Processing time irrelevant
- User-written code + centrally produced selection algorithms



HEP, Big Data & Real Time Processing

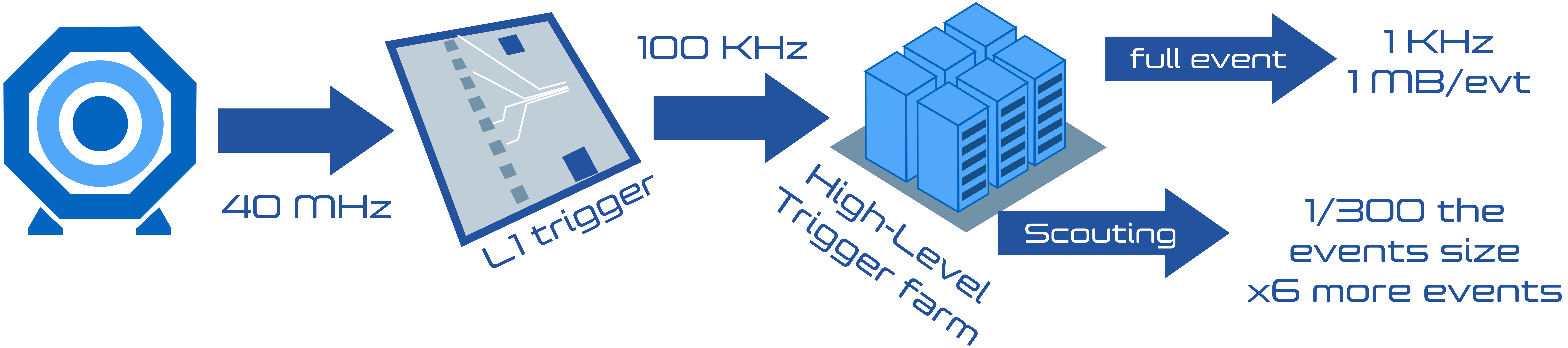
The Trigger Problem

- Too many data, too large data → need to filter online
- Filters based on pheno bias: **we might be loosing good events**



- L1 trigger: local, hardware based, on FPGA, @experiment site
- HLT: local/global, software based, on CPU, @experiment site
- Offline: global, software based, on CPU, @CERN T0
- Analysis: user-specific applications running on the grid

Doing more with less



Real-time new physics search with large datasets

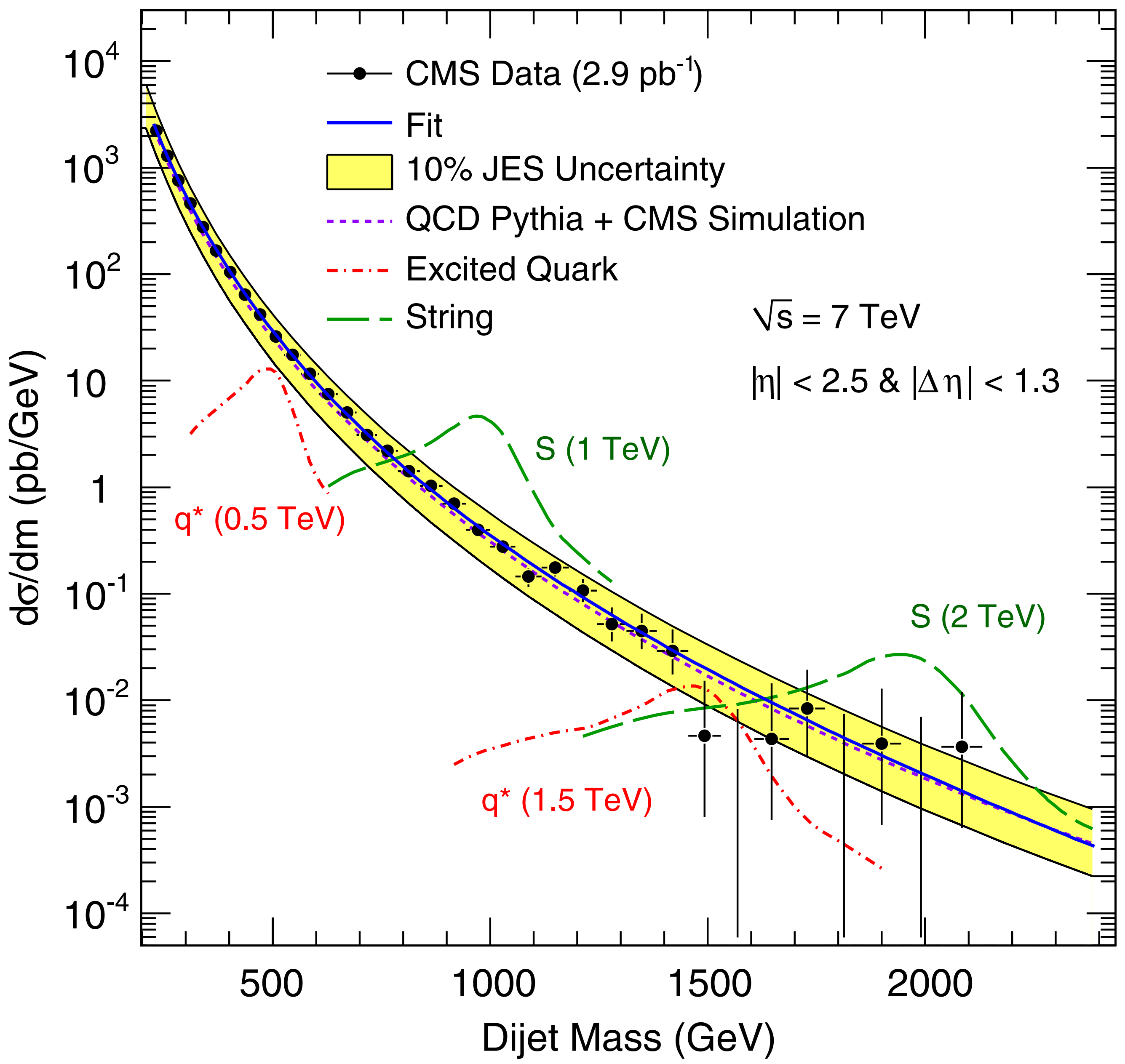
- *Run reconstruction in the trigger farm*
- *Avoid resource limitations: write less information (a few floats) for more events*
- *Probes unexplored territory, previously left behind*

Problem: practical (so far) only for specific topologies

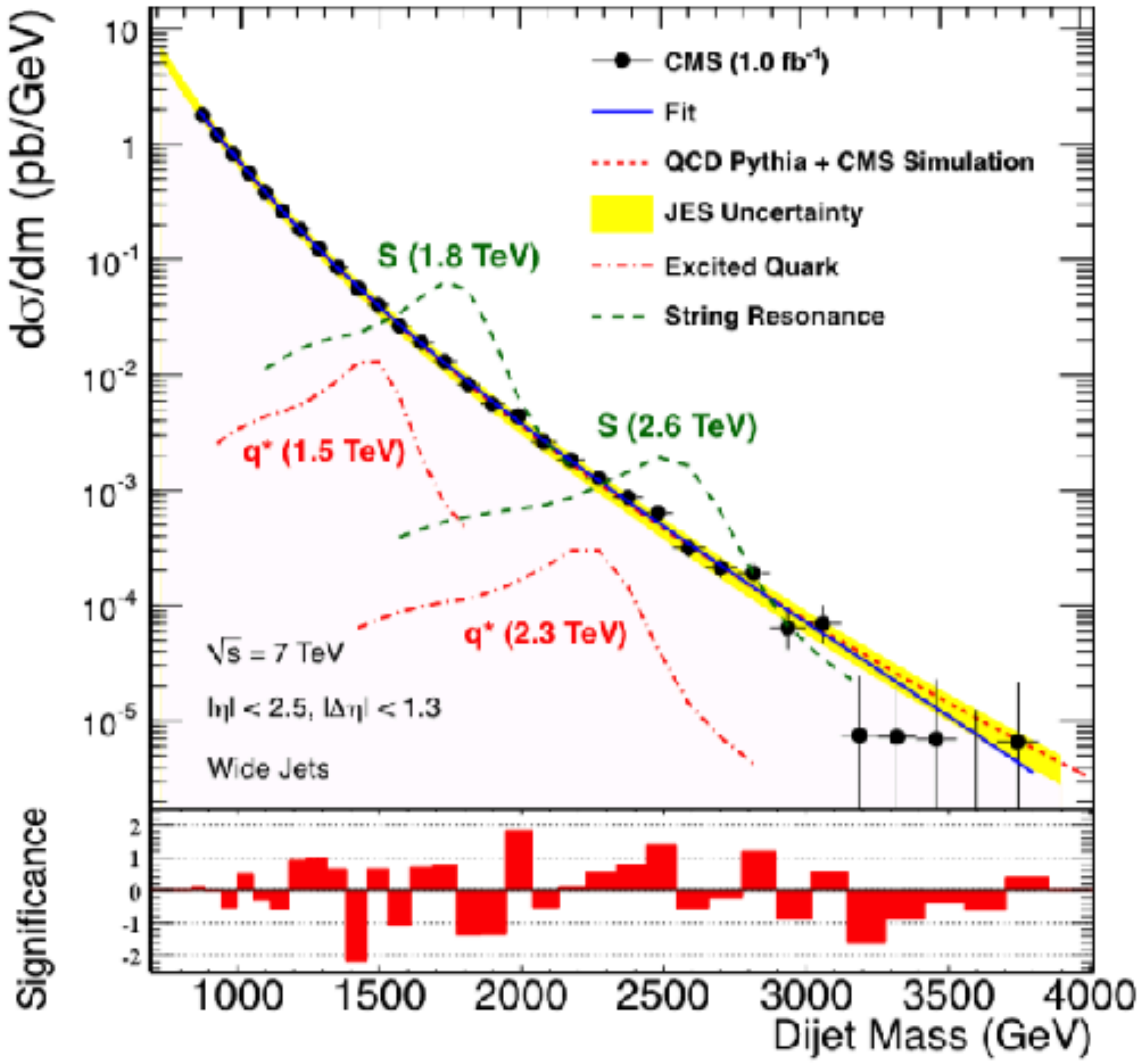
Why did we do this?

- In Run I, dijet search was the first BSM analysis published by CMS
  - Quick improved results from Tevatron in a wide range of mass spectra
  - Quickly forced to reduce mass range under investigation, due to increasing trigger rates vs limited resources
- Scouting was introduced to recover the lost territory (500 to 1100 GeV)

3 pb^{-1} @7 TeV in 2010

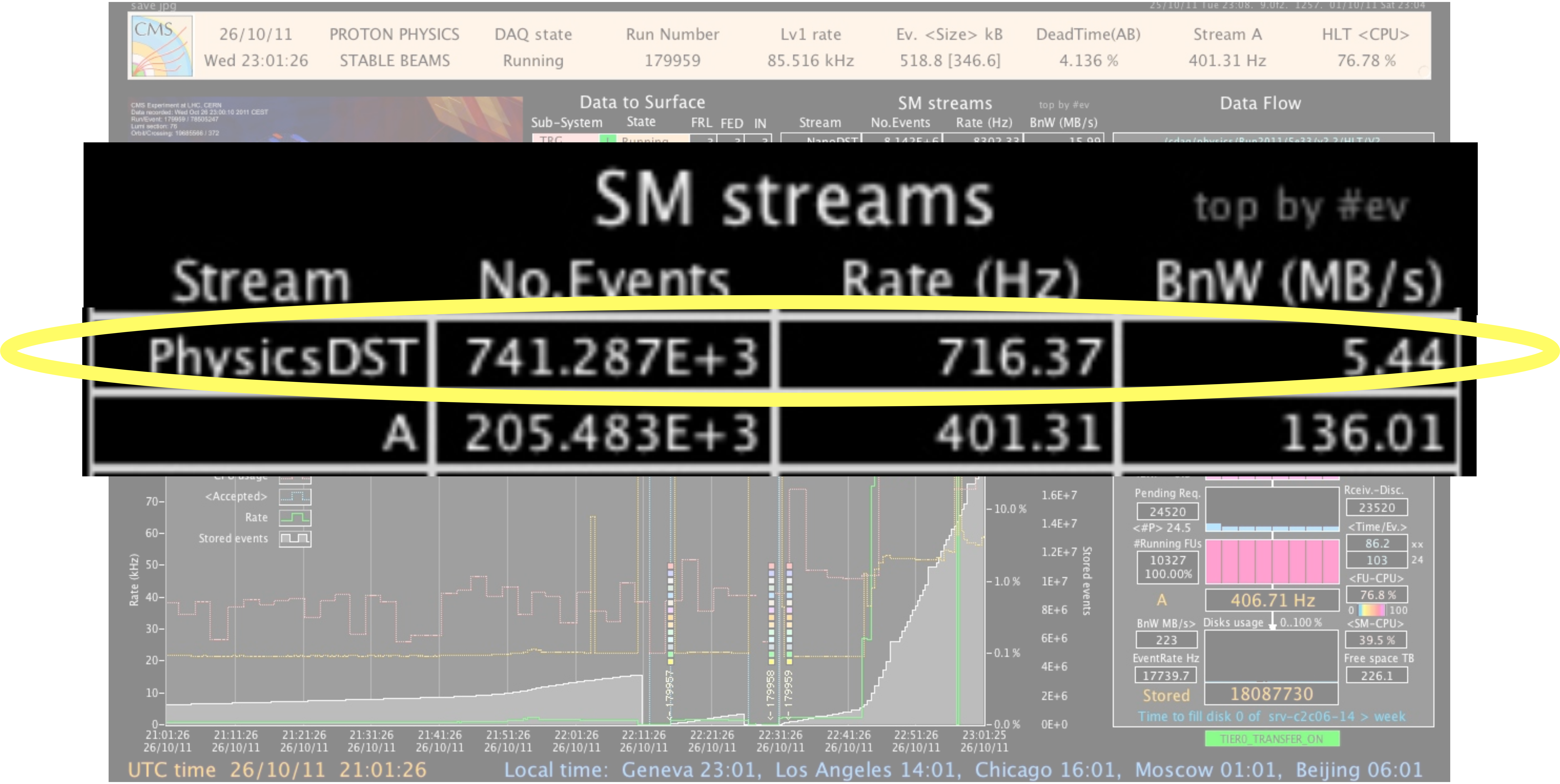


1 fb^{-1} @7 TeV in 2011



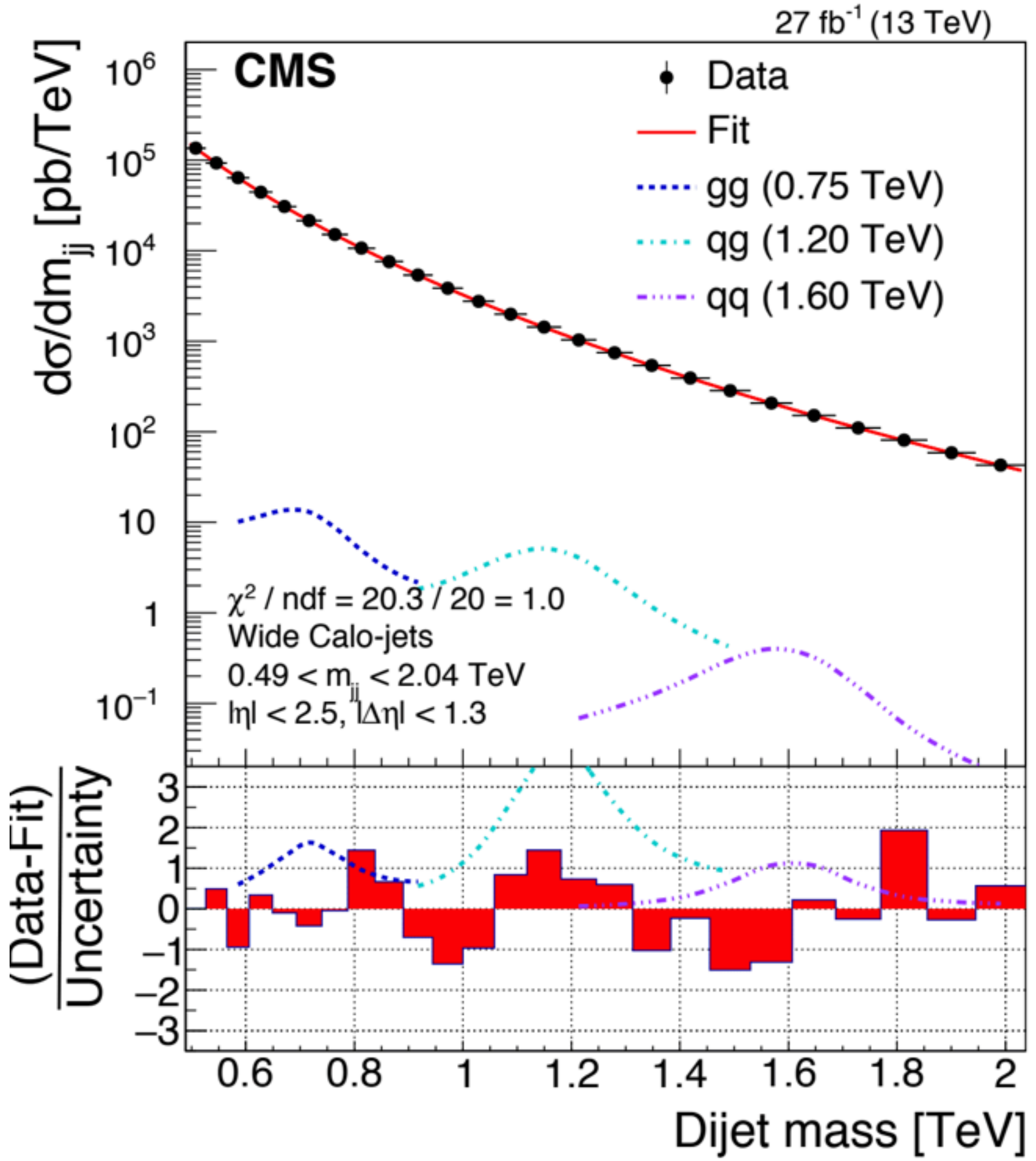
The first attempt

The first attempt



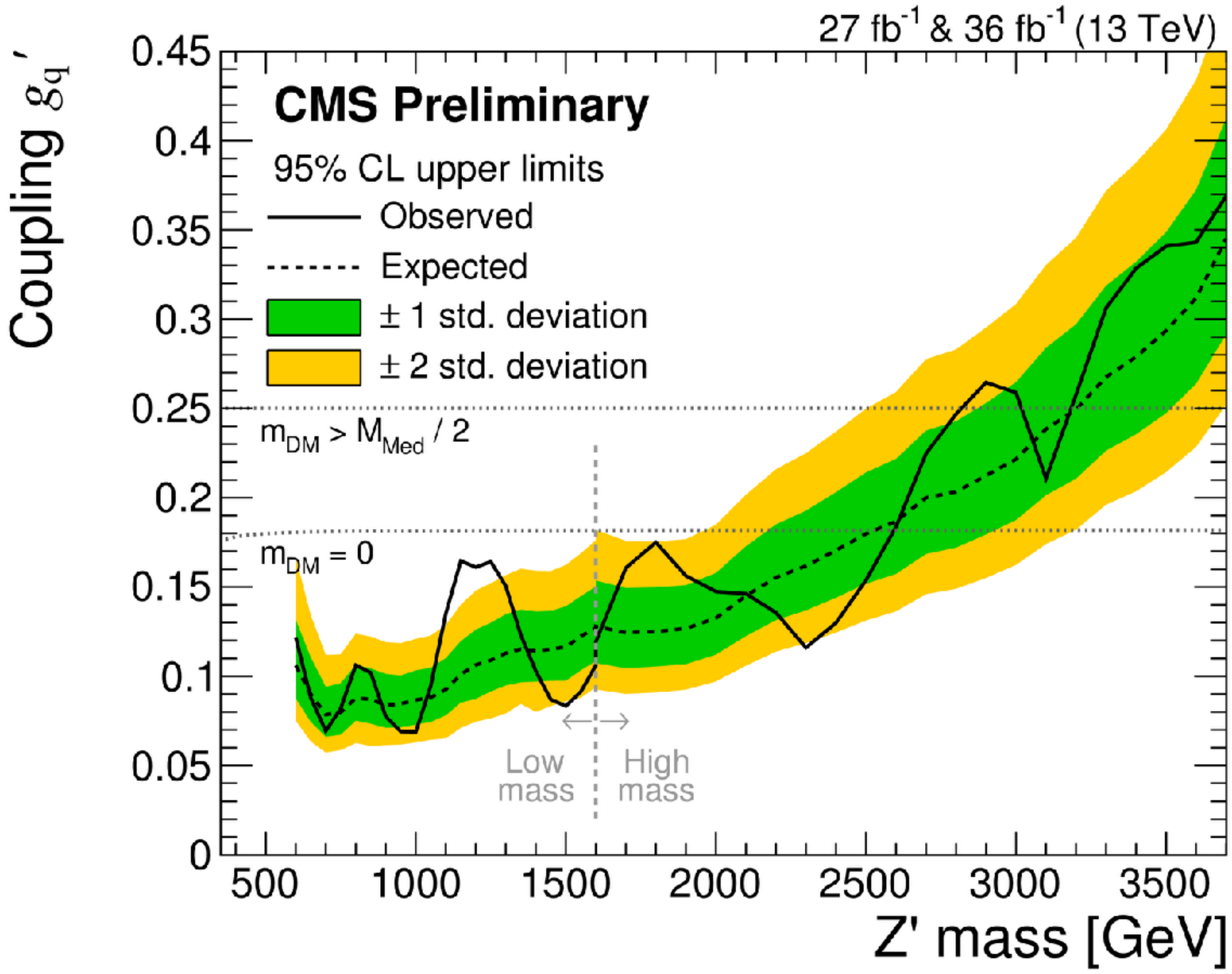
What we accomplished

- Recovered sensitivity to 500 GeV resonances
- Reached limitation of L1 seed-> need to improve our hardware trigger (more on this later)
- Now extending the method to more final states (collected x3 more data than the rest of CMS in 2017)

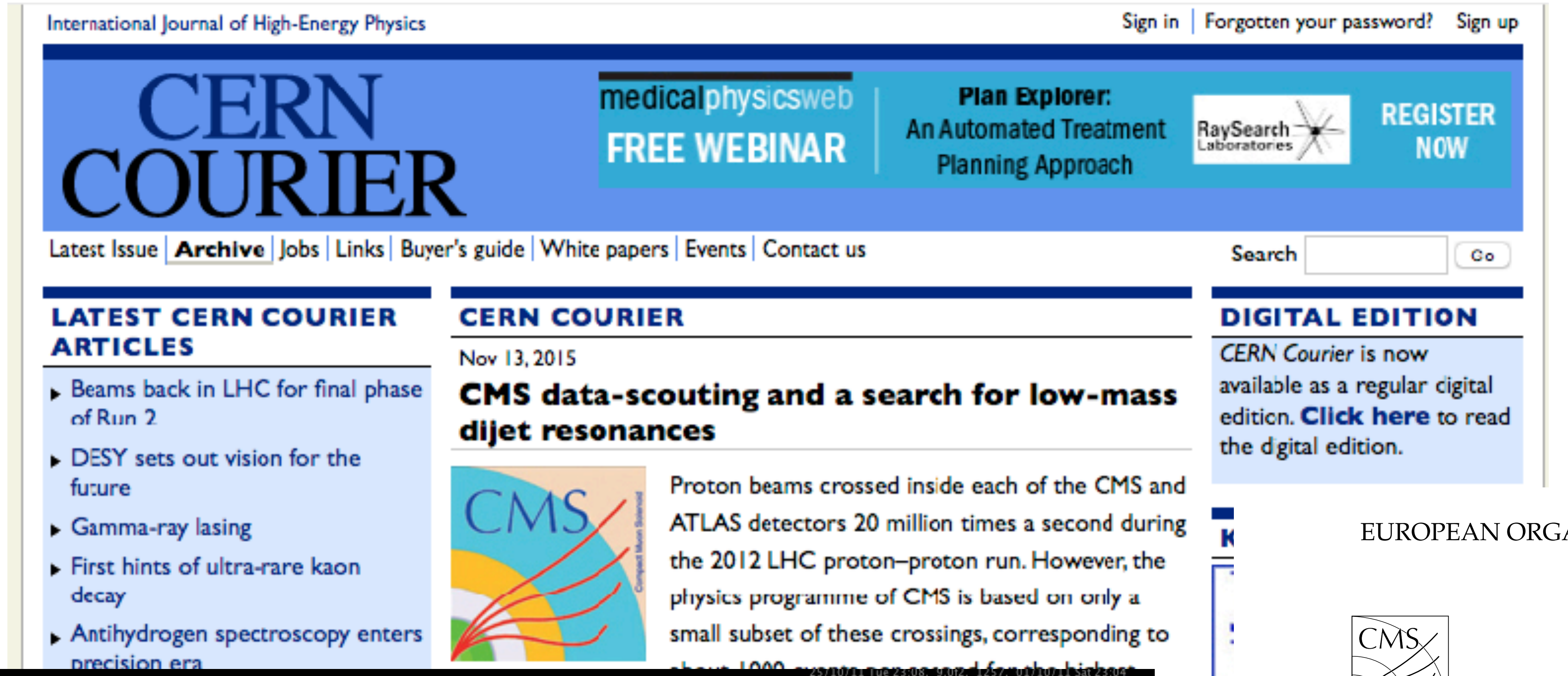


What we accomplished

- *Kept sensitivity to 500-1500 GeV resonances*
- *Current limitation is L1 efficiency*
- *Can probe lower couplings by collecting more data*



An established approach



International Journal of High-Energy Physics

Sign in | Forgotten your password? | Sign up

CERN COURIER

medicalphysicsweb FREE WEBINAR

Plan Explorer: An Automated Treatment Planning Approach

RaySearch Laboratories

REGISTER NOW

Latest Issue | Archive | Jobs | Links | Buyer's guide | White papers | Events | Contact us

Search Go

LATEST CERN COURIER ARTICLES

- Beams back in LHC for final phase of Run 2
- DESY sets out vision for the future
- Gamma-ray lasing
- First hints of ultra-rare kaon decay
- Antihydrogen spectroscopy enters precision era

CERN COURIER

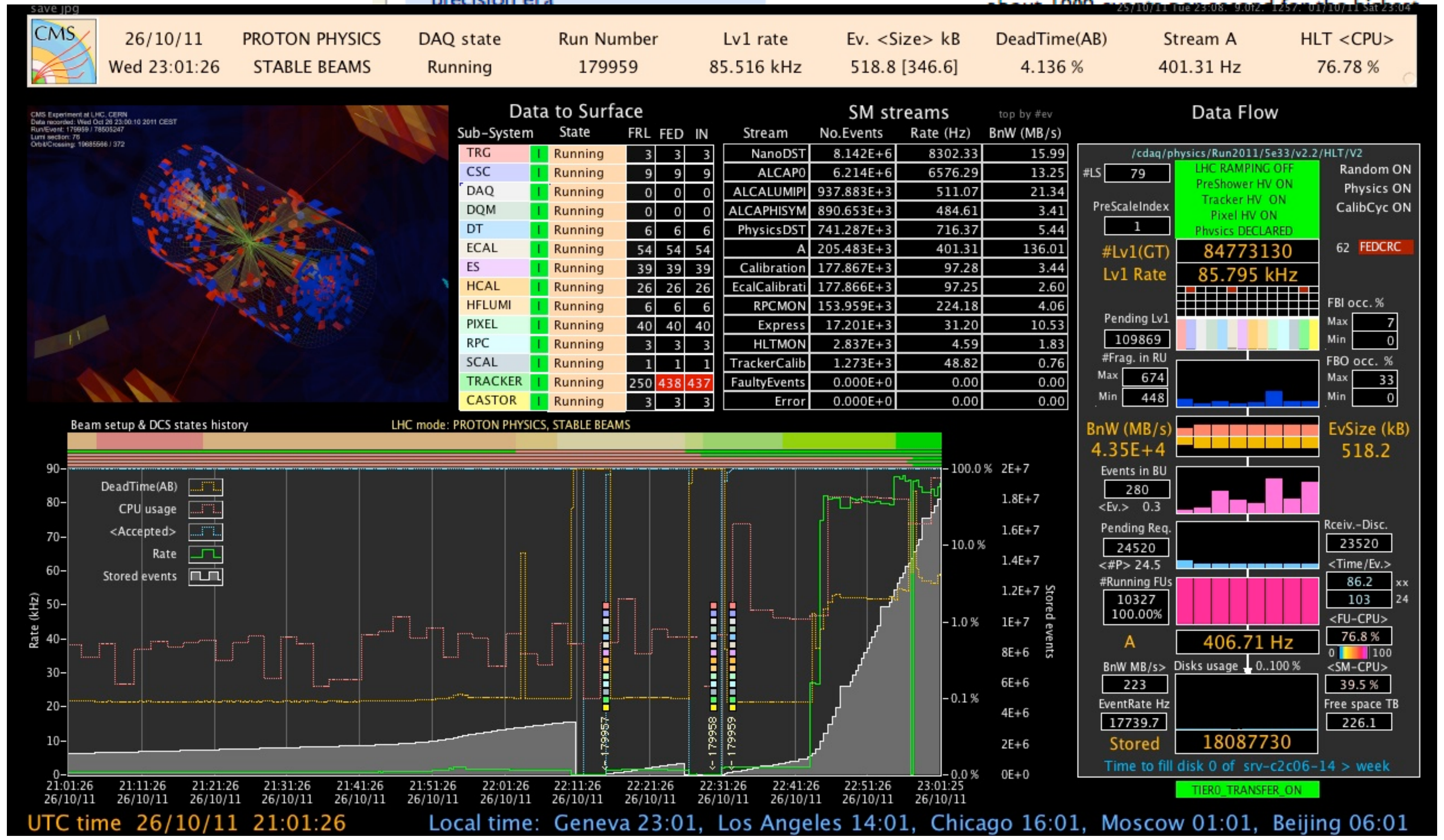
Nov 13, 2015

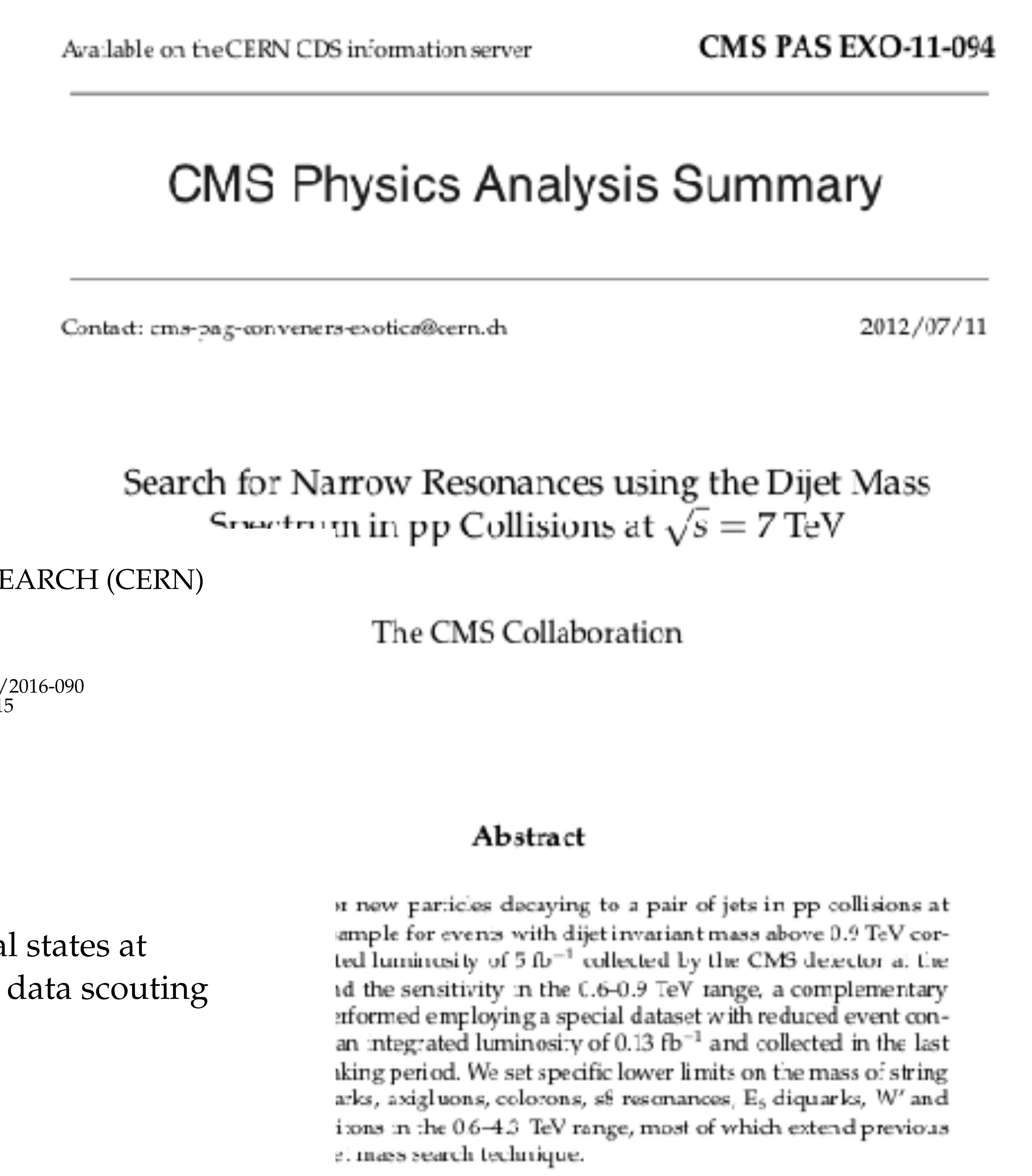
CMS data-scouting and a search for low-mass dijet resonances

Proton beams crossed inside each of the CMS and ATLAS detectors 20 million times a second during the 2012 LHC proton–proton run. However, the physics programme of CMS is based on only a small subset of these crossings, corresponding to

DIGITAL EDITION

CERN Courier is now available as a regular digital edition. Click here to read the digital edition.





Available on the CERN CDS information server

CMS PAS EXO-11-094

CMS Physics Analysis Summary

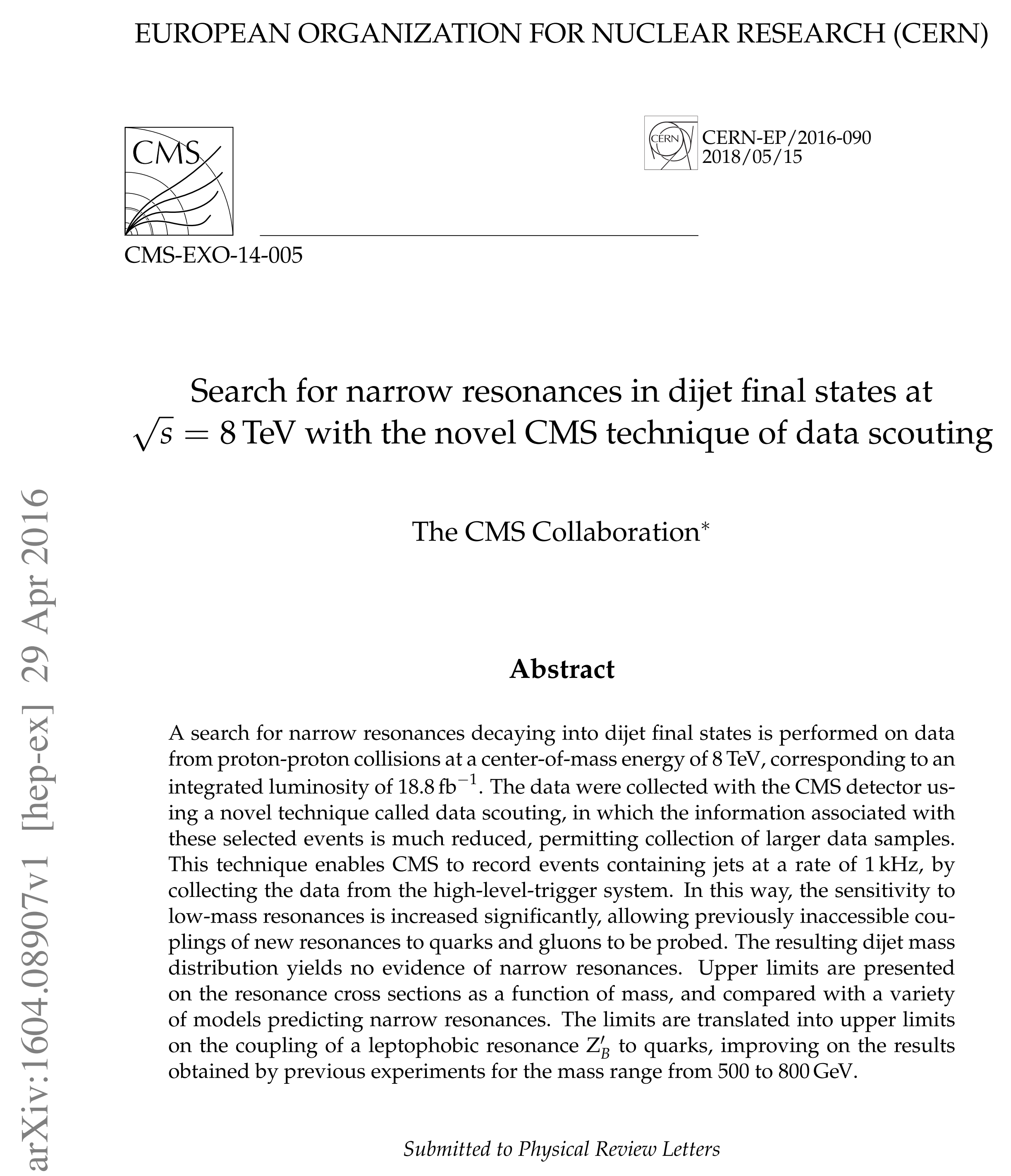
Contact: cms-pag-conveners-exotica@cern.ch

2012/07/11

Search for Narrow Resonances using the Dijet Mass Spectrum in pp Collisions at $\sqrt{s} = 7$ TeV

The CMS Collaboration

Abstract



EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH (CERN)

CMS-EXO-14-005

CERN-EP/2016-090
2018/05/15

arXiv:1604.08907v1 [hep-ex] 29 Apr 2016

Search for narrow resonances in dijet final states at $\sqrt{s} = 8$ TeV with the novel CMS technique of data scouting

The CMS Collaboration*

Abstract

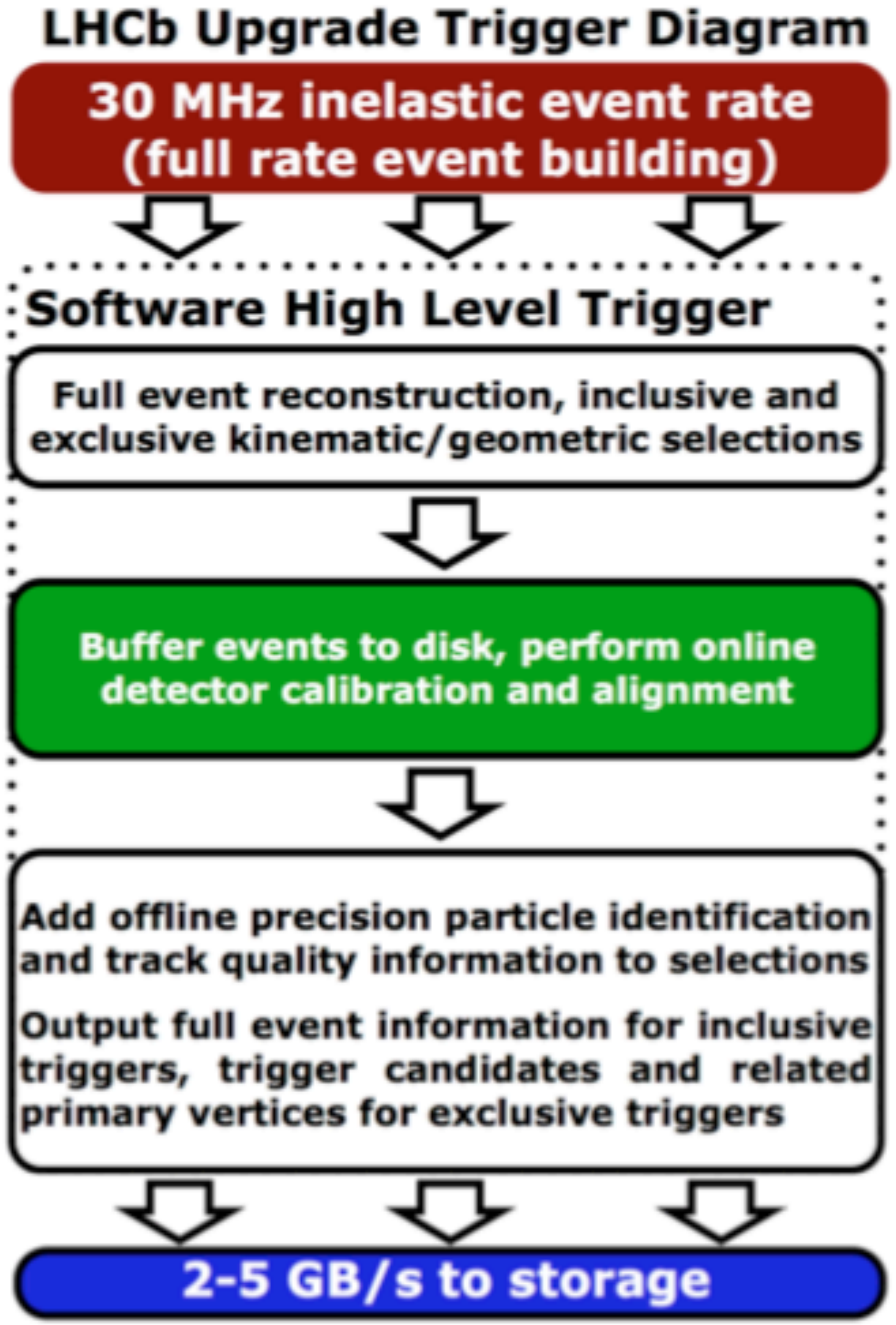
A search for narrow resonances decaying into dijet final states is performed on data from proton-proton collisions at a center-of-mass energy of 8 TeV, corresponding to an integrated luminosity of 18.8 fb^{-1}. The data were collected with the CMS detector using a novel technique called data scouting, in which the information associated with these selected events is much reduced, permitting collection of larger data samples. This technique enables CMS to record events containing jets at a rate of 1 kHz, by collecting the data from the high-level-trigger system. In this way, the sensitivity to low-mass resonances is increased significantly, allowing previously inaccessible couplings of new resonances to quarks and gluons to be probed. The resulting dijet mass distribution yields no evidence of narrow resonances. Upper limits are presented on the resonance cross sections as a function of mass, and compared with a variety of models predicting narrow resonances. The limits are translated into upper limits on the coupling of a leptophobic resonance Z'_B to quarks, improving on the results obtained by previous experiments for the mass range from 500 to 800 GeV.

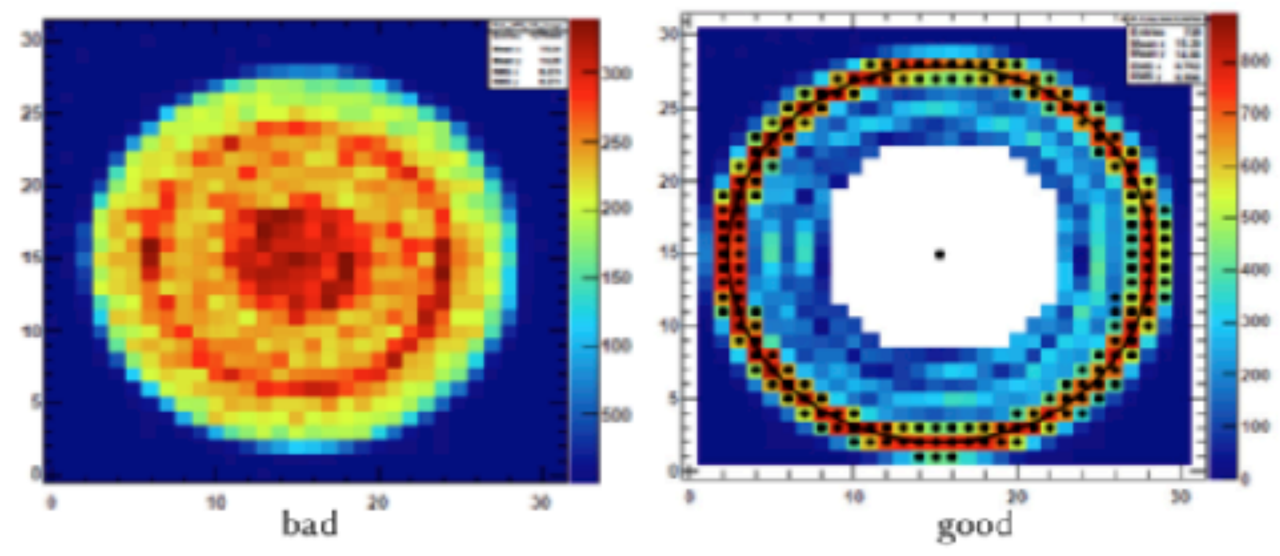
Submitted to Physical Review Letters

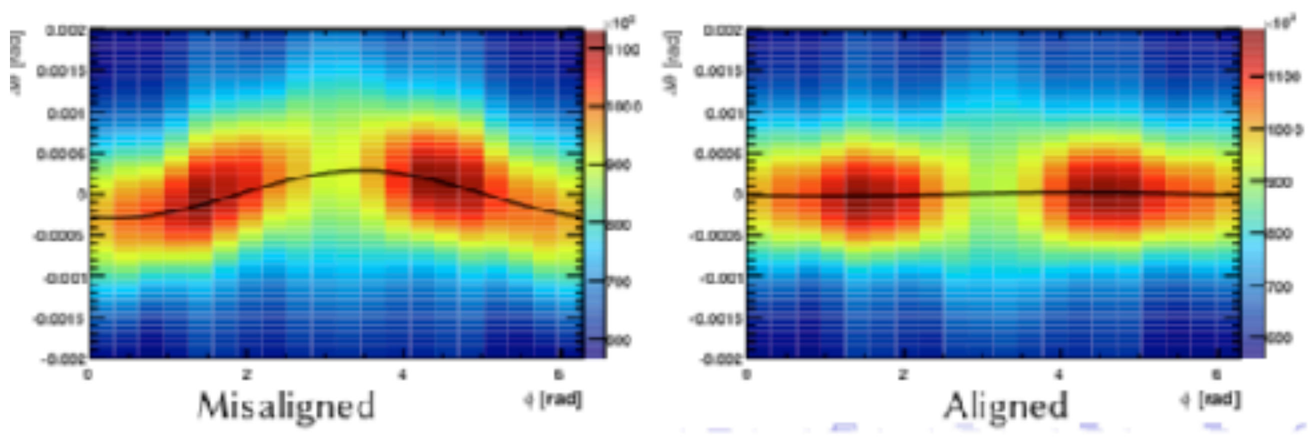
European Research Council

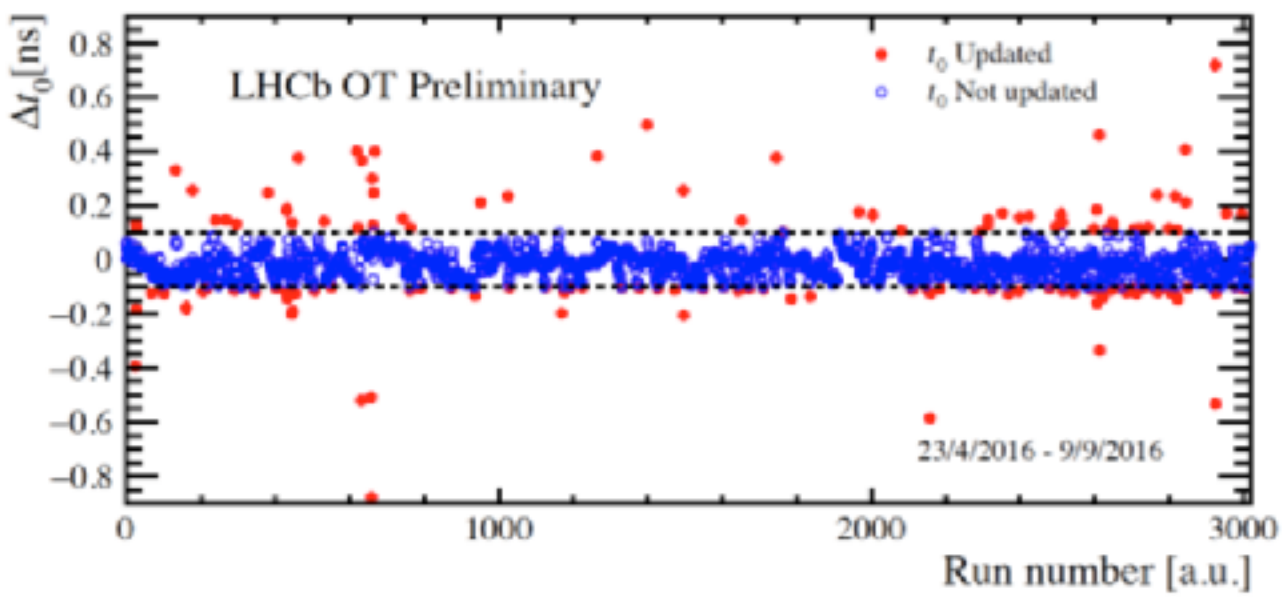
Next-step: Trigger-less

- *LHCb & ALICE soon to start a detector & online-infrastructure upgrade. Final goal is to*
  - *Read ALL collisions*
  - *Process them in real time*
  - *Align & calibrate detector at the same time*
- *The ultimate extrapolation of the scouting paradigm: Can take more data -> increase detector precision*

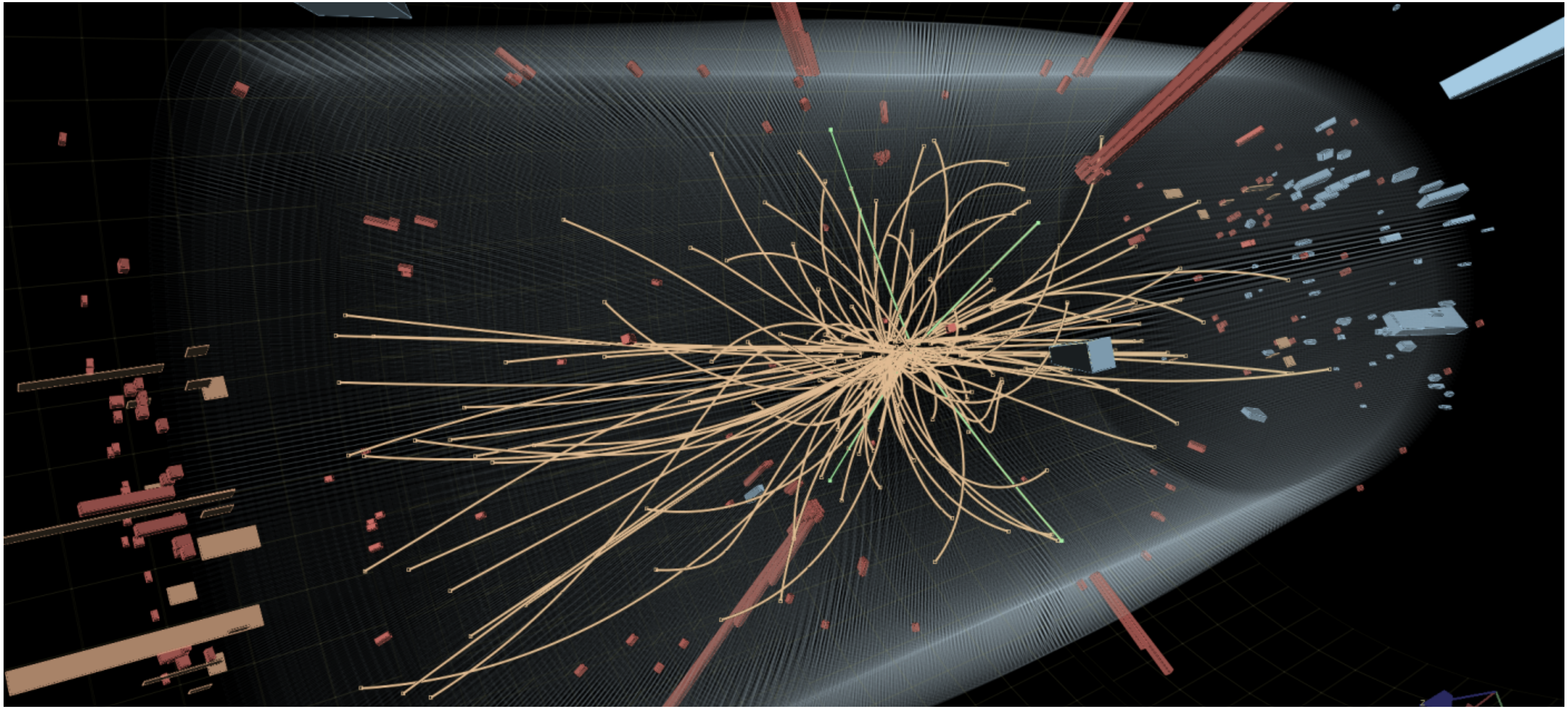








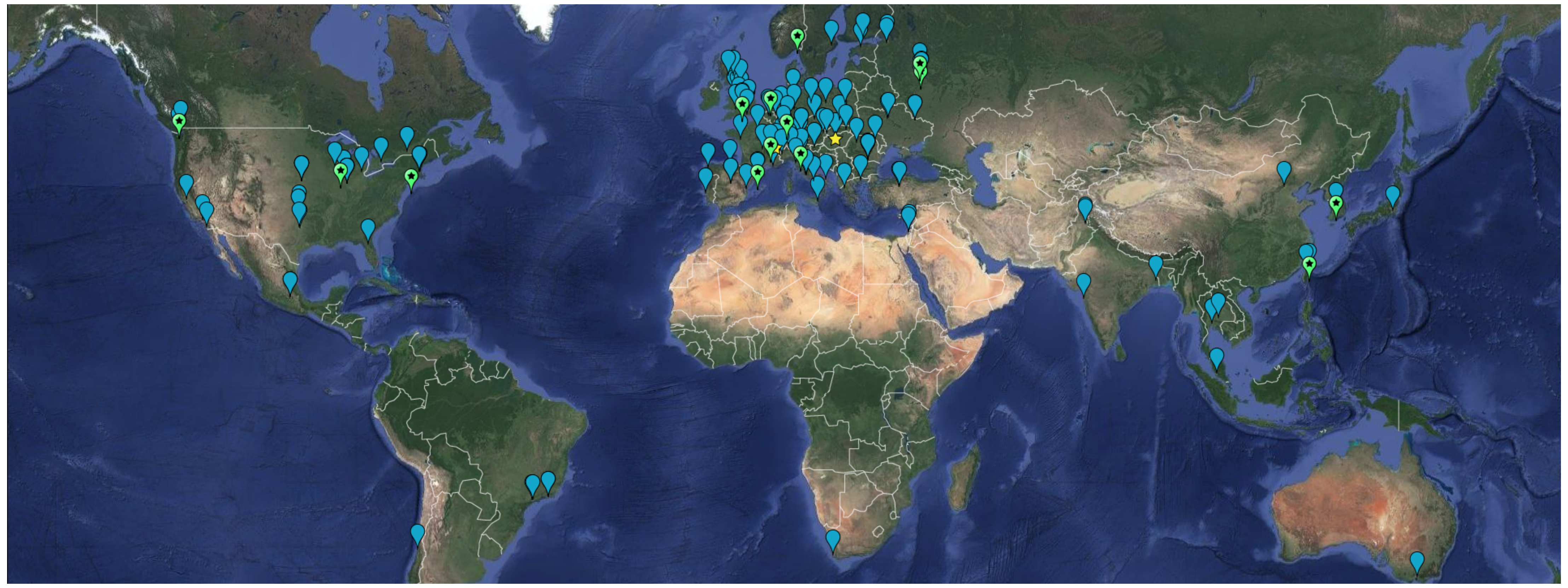
European Research Council

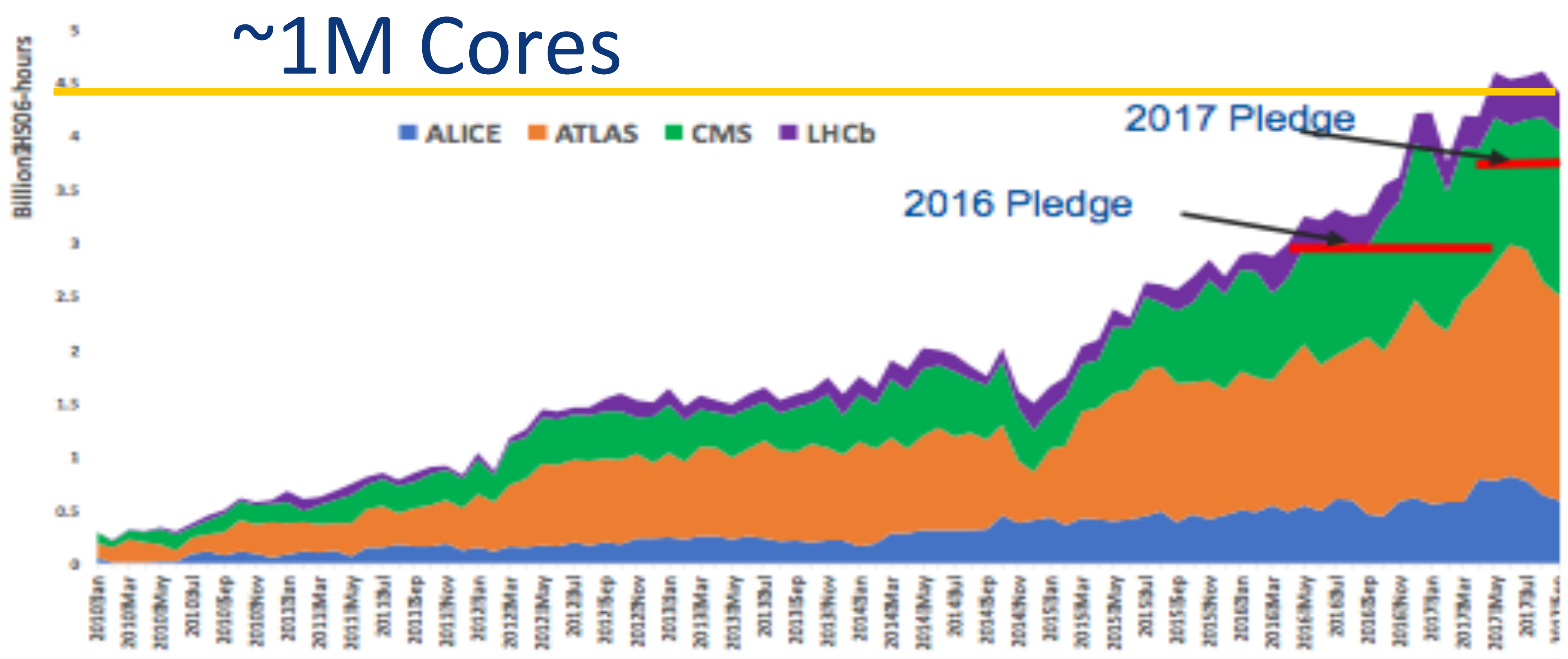


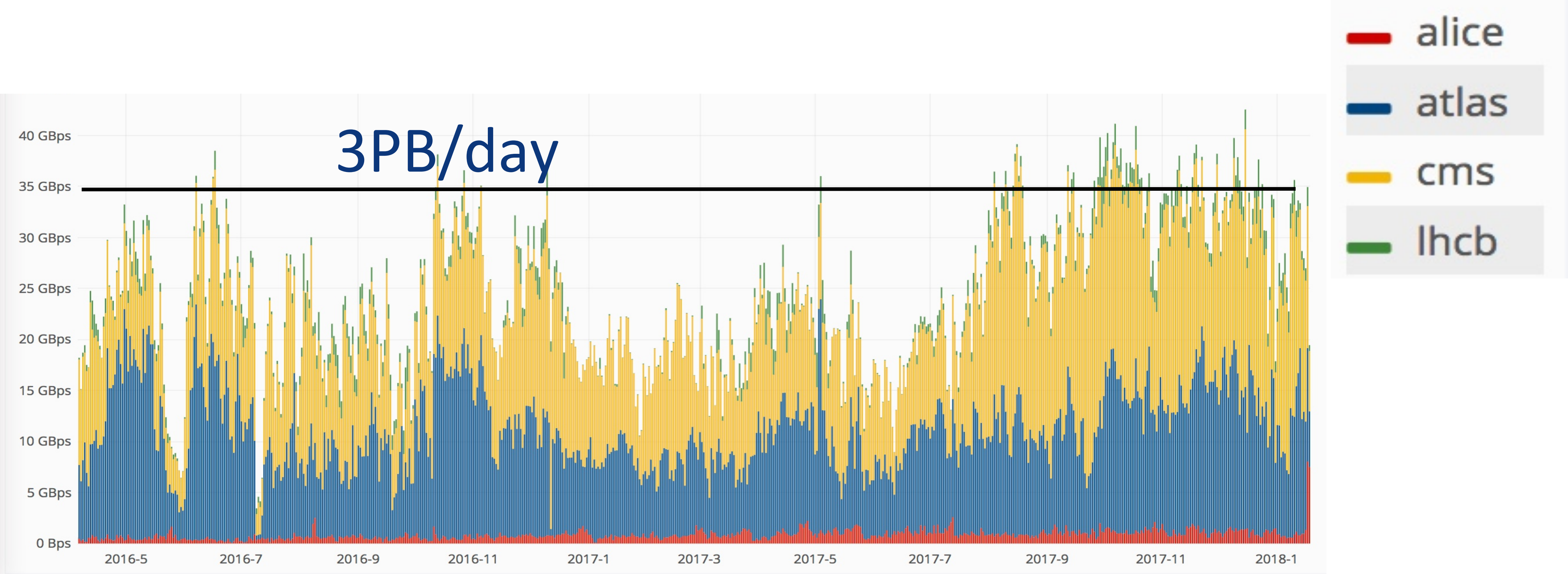
HEP, Cloud & HPCs

The WLCG Grid

- 170 centres in 42 countries, for central processing and analysis-related user jobs
  - 1M cores
  - 1 EB storage
  - >2M jobs & 3 PB moved /day



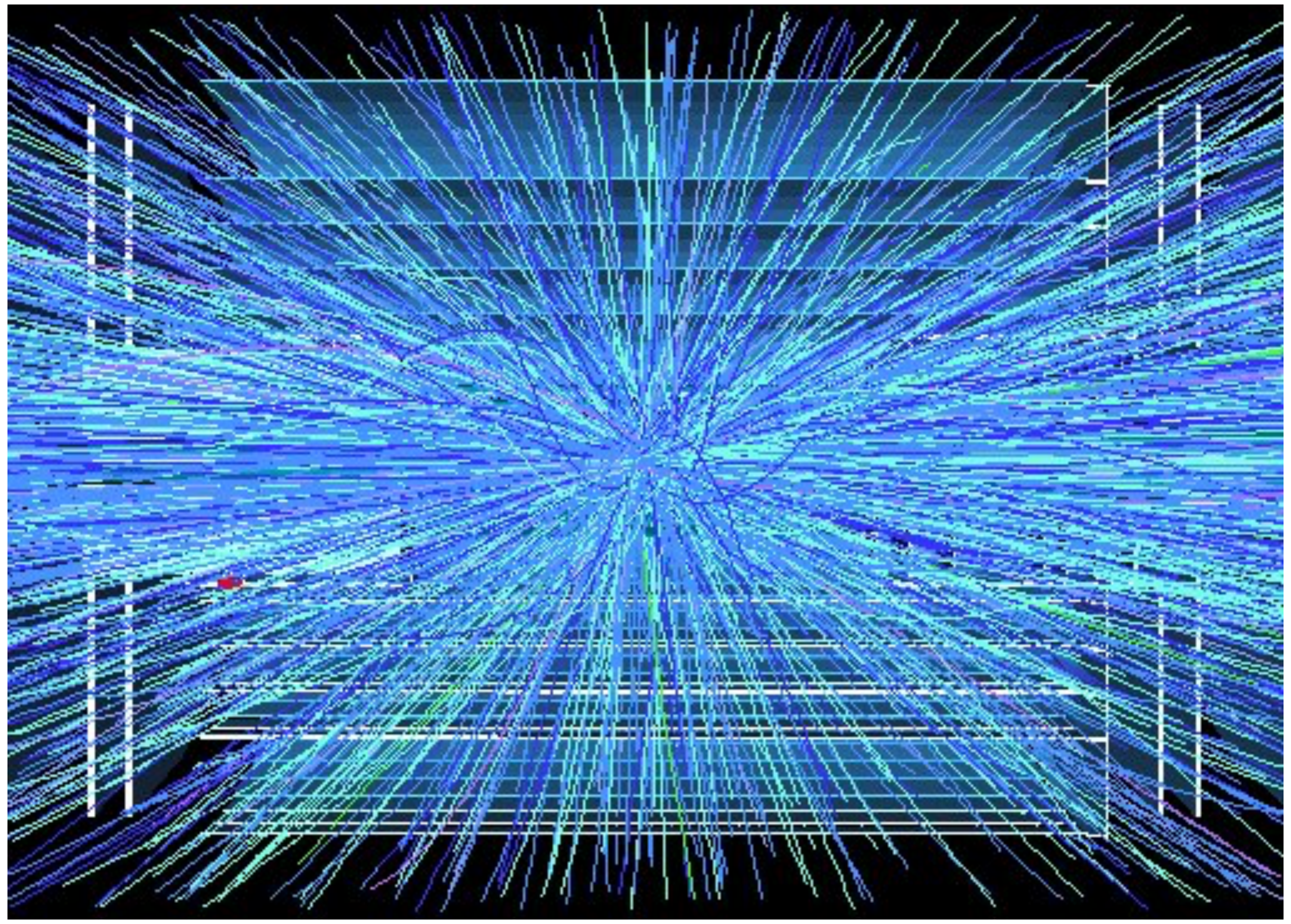


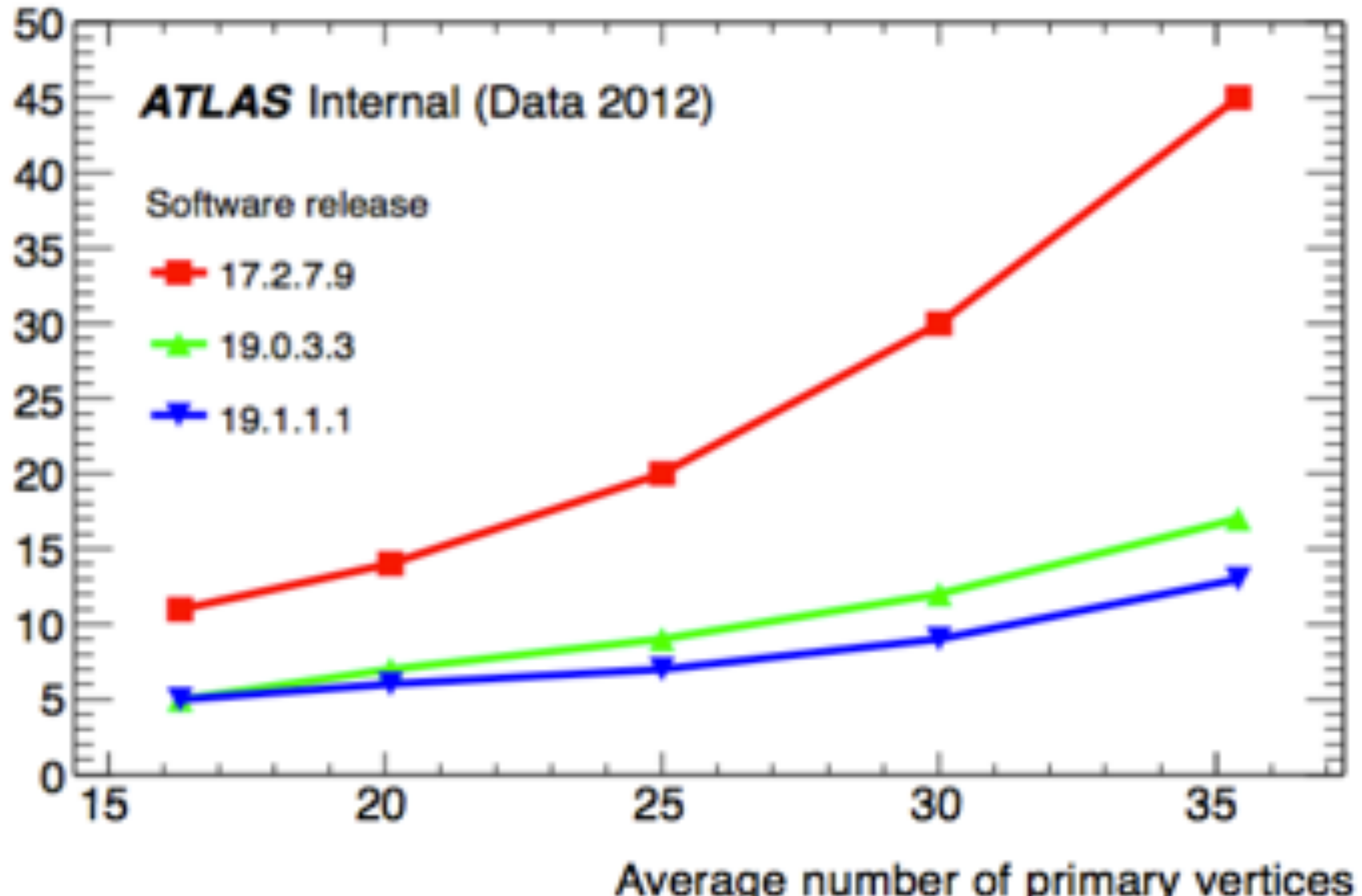


European Research Council

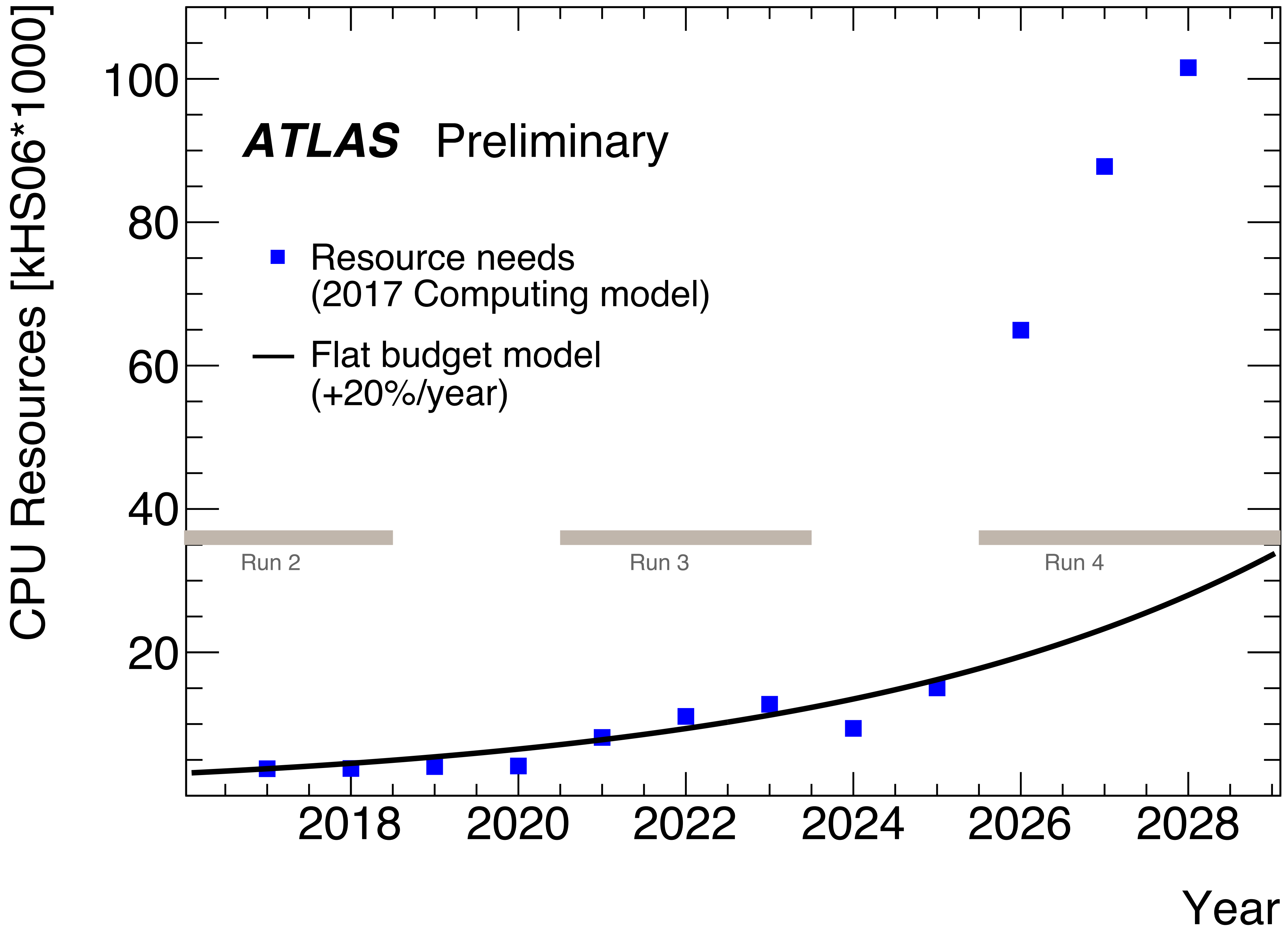
The challenge ahead

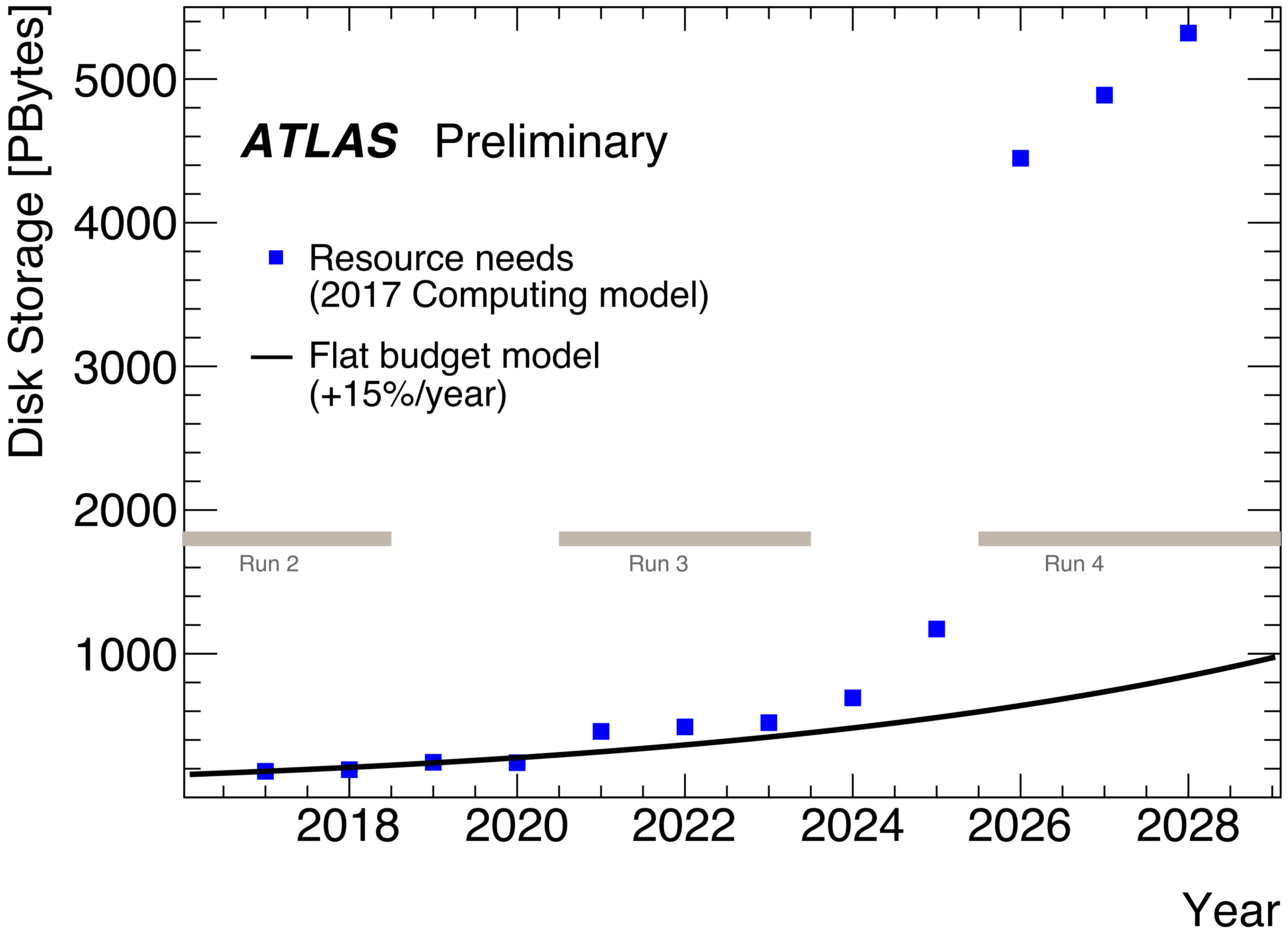
- The evolving conditions of the machine are drifting the experiments to more prohibitive environments (luminosity comes with a cost)
- More (& bigger) events to handle
- More noise from pileup interactions
- Increase in resources will not scale with needs
- Flat (or decreasing?) budget
- (Non linearly) increasing demand
- Need to find better ways to do things





The challenge ahead





- *Event complexity, volume, and number will challenge the current paradigm*
- *Assuming flat budget, we simply cannot keep doing things as we do now*

More CPU: Cloud

- The growing complexity of LHC events is forcing us to look for more resources, particularly for the computation-heavy central reconstruction
  - CERN extended the T0 center by adding a site in Wigner (Hungary)
  - Similar approach used by T1s (e.g., CNAF T1 extended with CPUs in Bari)
- Paradigm extended opportunistically to Cloud services and HPC sites

amazon web services

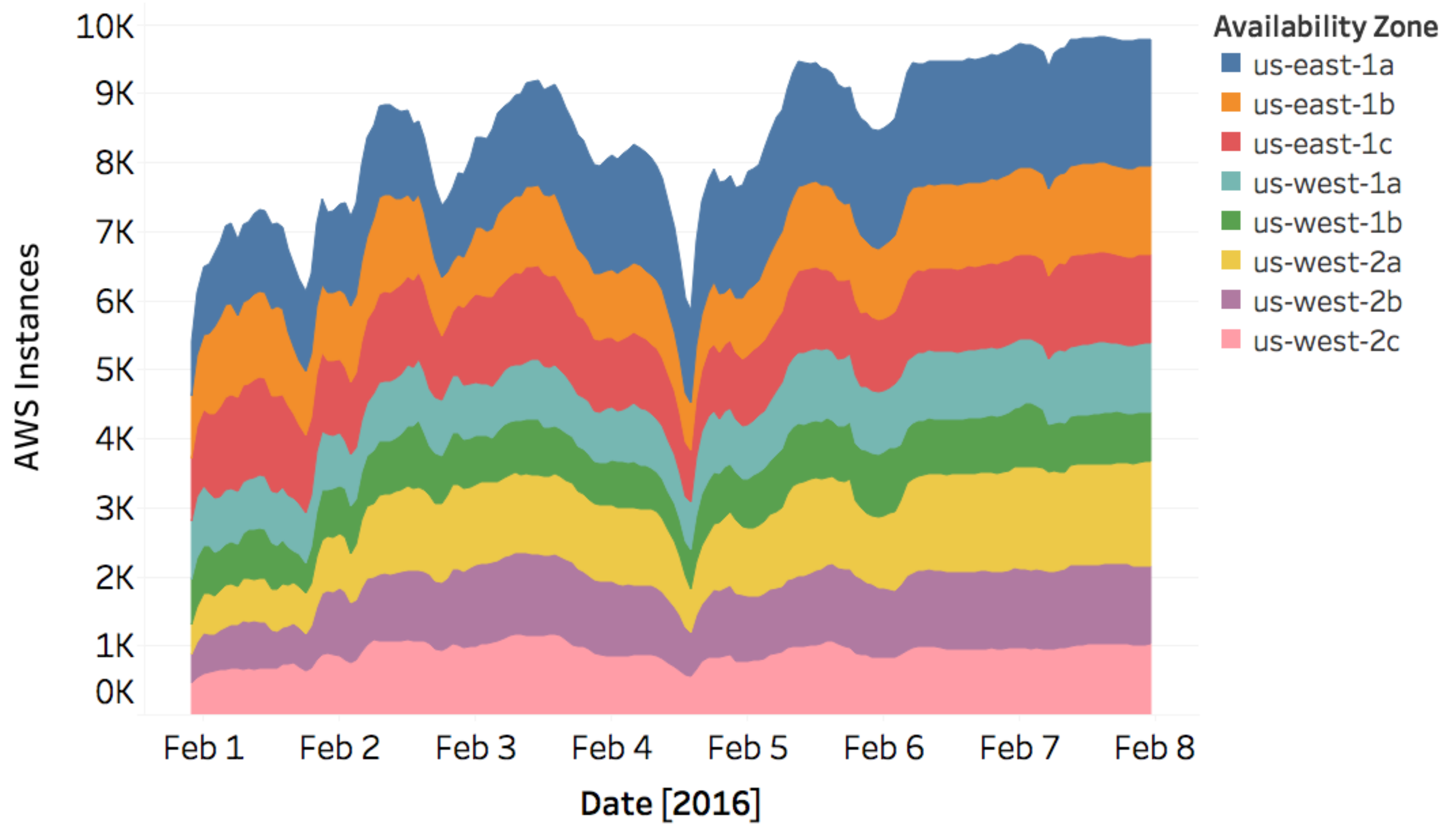
aruba CLOUD

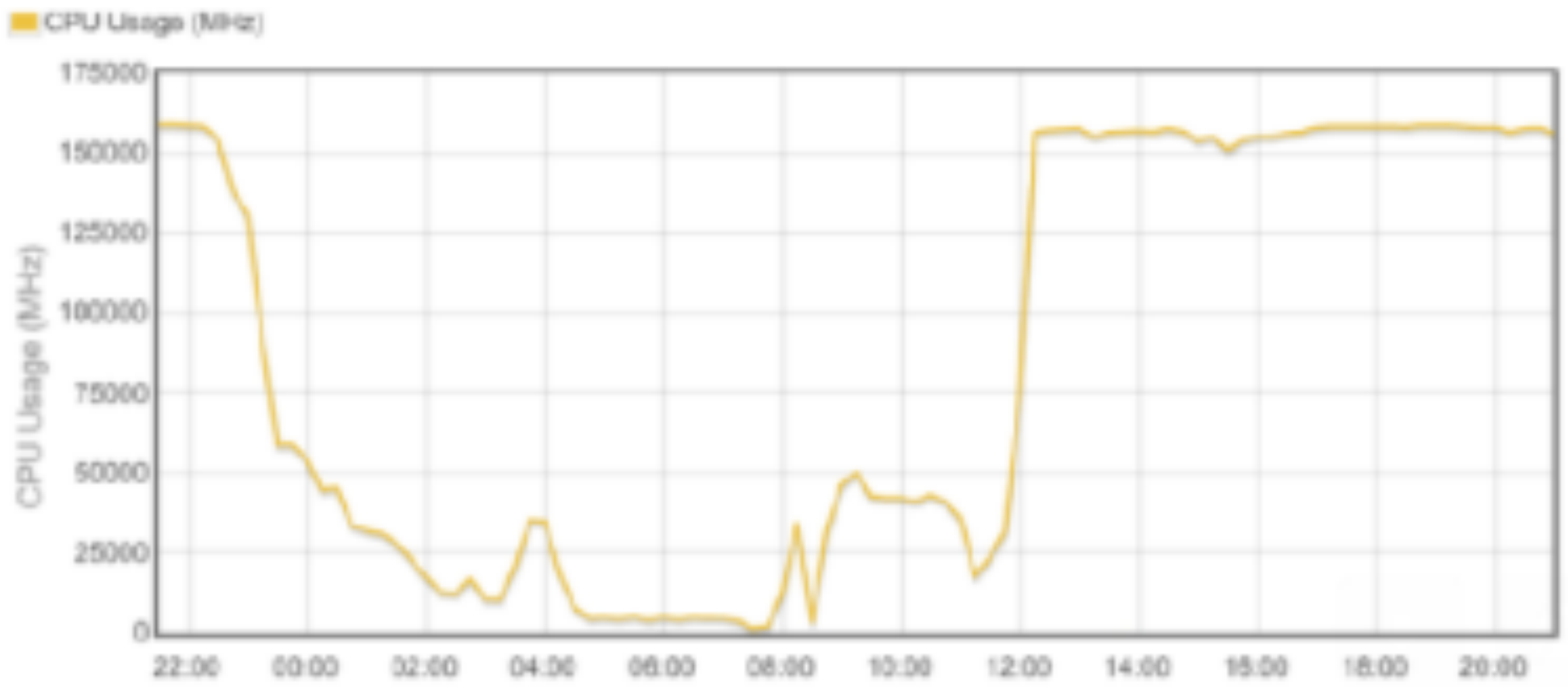
Microsoft Azure

erc European Research Council

More CPU: Cloud

- *The growing complexity of LHC events is forcing us to look for more resources, particularly for the computation-heavy central reconstruction*
  - *CERN extended the T0 center by adding a site in Wigner (Hungary)*
  - *Similar approach used by T1s (e.g., CNAF T1 extended with CPUs in Bari)*
- *Paradigm extended opportunistically to Cloud services and HPC sites*

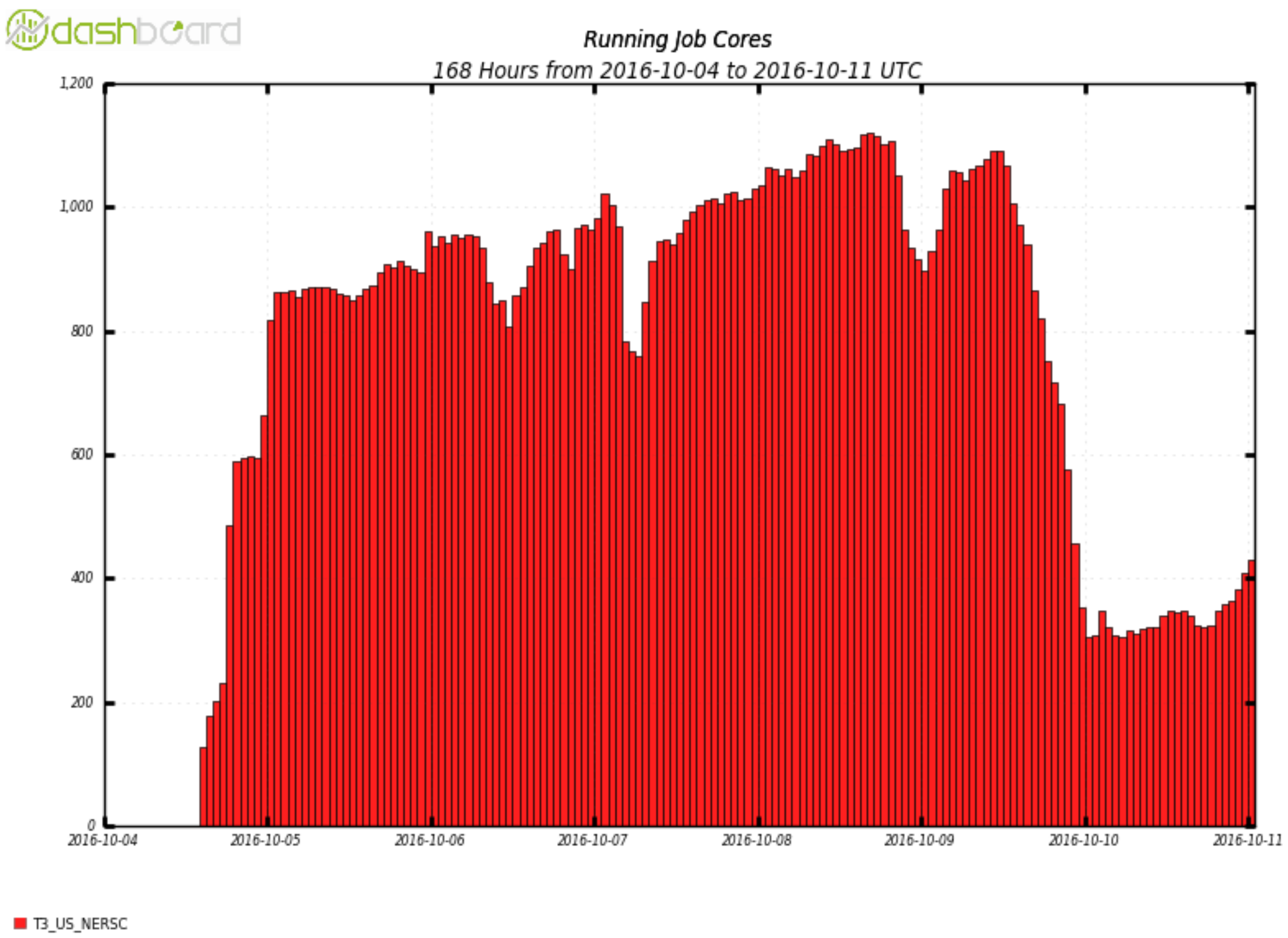




European Research Council

More CPU: HPCs

- Similar tests done on HPC sites (NERSC Cori)
- x86 machines, in very different setup than T0/T1/T2/T3 sites
- Challenge stands in working out all details and finding workarounds to incompatible setups (e.g., not-supported components)
- Result: storage-less site added to the CMS grid as yet another Tx



European Research Council

A convenient new Paradigm?

Grid

- Virtual Organizations (VOs) of users trusted by Grid sites
- VOs get allocations ➜ **Pledges**
  - –Unused allocations: opportunistic resources

“Things you borrow”

Trust Federation

Cloud

- Community Clouds - Similar trust federation to Grids
- Commercial Clouds - **Pay-As-You-Go** model
  - Strongly accounted
  - Near-infinite capacity ➜ **Elasticity**
  - Spot price market

“Things you rent”

Economic Model

HPC

- Researchers granted access to HPC installations
- Peer review committees award **Allocations**
  - Awards model designed for individual PIs rather than large collaborations

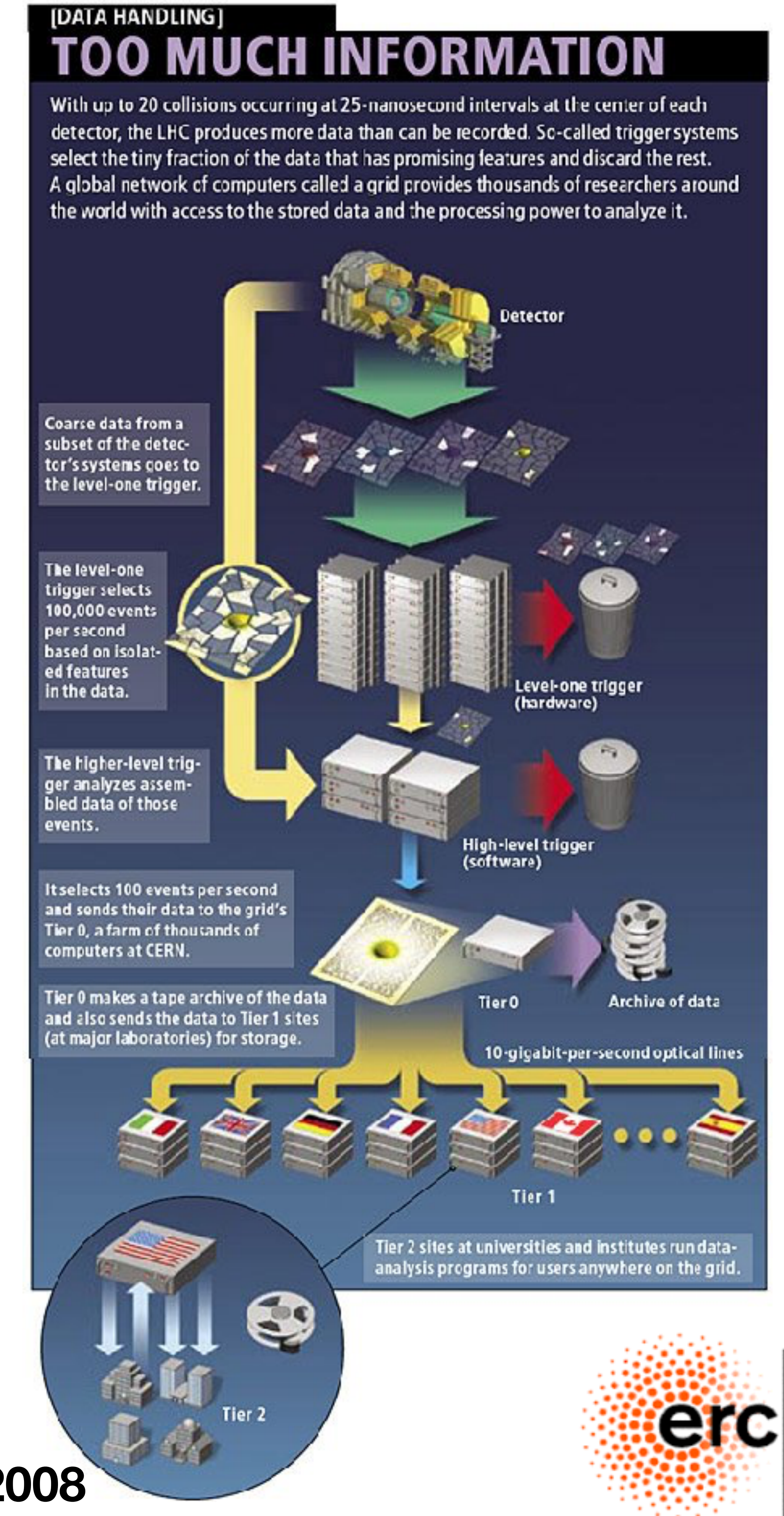
“Things you are given”

Grant Allocation

HEP, Big Data & “offline” Processing

The foreseen analysis workflow

- ***Central processing:*** *Runs @T0. Start from RAW data and creates a collection of "Primary" Datasets then distributed to T1s*
- ***Data skimming:*** *Runs @T0 or T1s. From the Primary Datasets, produce "Secondary datasets" by removing events (so why did you take them to start with?) or reducing the information (data compression)*
- ***Data analysis:*** *runs on Secondary Datasets, applying analysis specific selection, reconstructing high-level objects on which signal-to-background discriminating quantities are computed. Runs on T3s, on the Grid, etc*
- ***Result extraction:*** *typically a ML fit, based on data distributions in signal region and control region + prediction from MC simulation (runs on laptops)*



Scientific America, Sep 2008

It didn't really go like that

- *Disk issue is less (but still quite) serious than anticipated:*
  - *We (all) introduced **AODs** (500-1000 kB/evt) compressed version of RECO data format. We saved disk, so we just distributed Primary Datasets rather than using the (very bad) Secondary datasets*
  - *With gain detector understanding, we (CMS) then moved forward to **miniAODs** (30 kB/evt) and **nanoAOD** (3 kB/evt), compressed data formats with top-bottom object definition, serving >80% of the analysis use cases*
- *Large **demand of CPU** faced breaking the paradigm rigidity:*
  - *T1s and T2s interconnection was improved. Now one runs a job somewhere accessing data somewhere else*
- ***Still, we would use more disk & CPU if we had it …***

Big Data tools & HEP

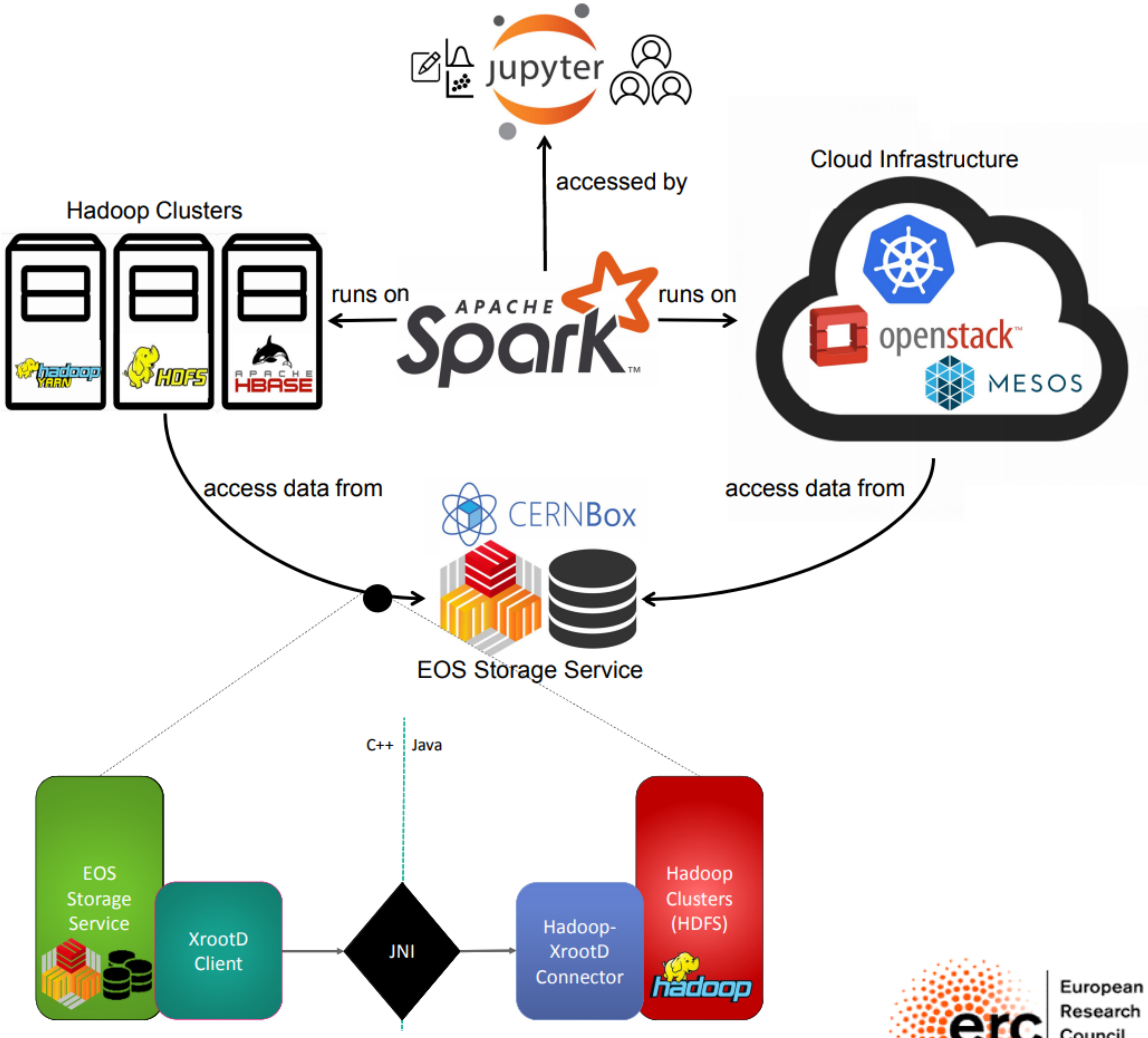
- A lot is happening outside HEP
  - full data-scientist echo-system
  - big-data handling tools
- But we have specific tools (ROOT)
  - optimized on our use cases
  - very competitive on I/O point of view
  - long-term future guaranteed (we develop it for ourselves)
- A big effort to integrate ROOT & outside-world big-data tools is ongoing, with promising results

BigData tools integration

- *Effort to modernise approach to data analysis by integrating/creating data-analytics tools for physics analyses*
- *Goals:*
  - *Reduce number of intermediate processing+storage steps*
  - *Allow analyses to run on (mini)AODs way down to the publication-ready plots in a data-science framework*

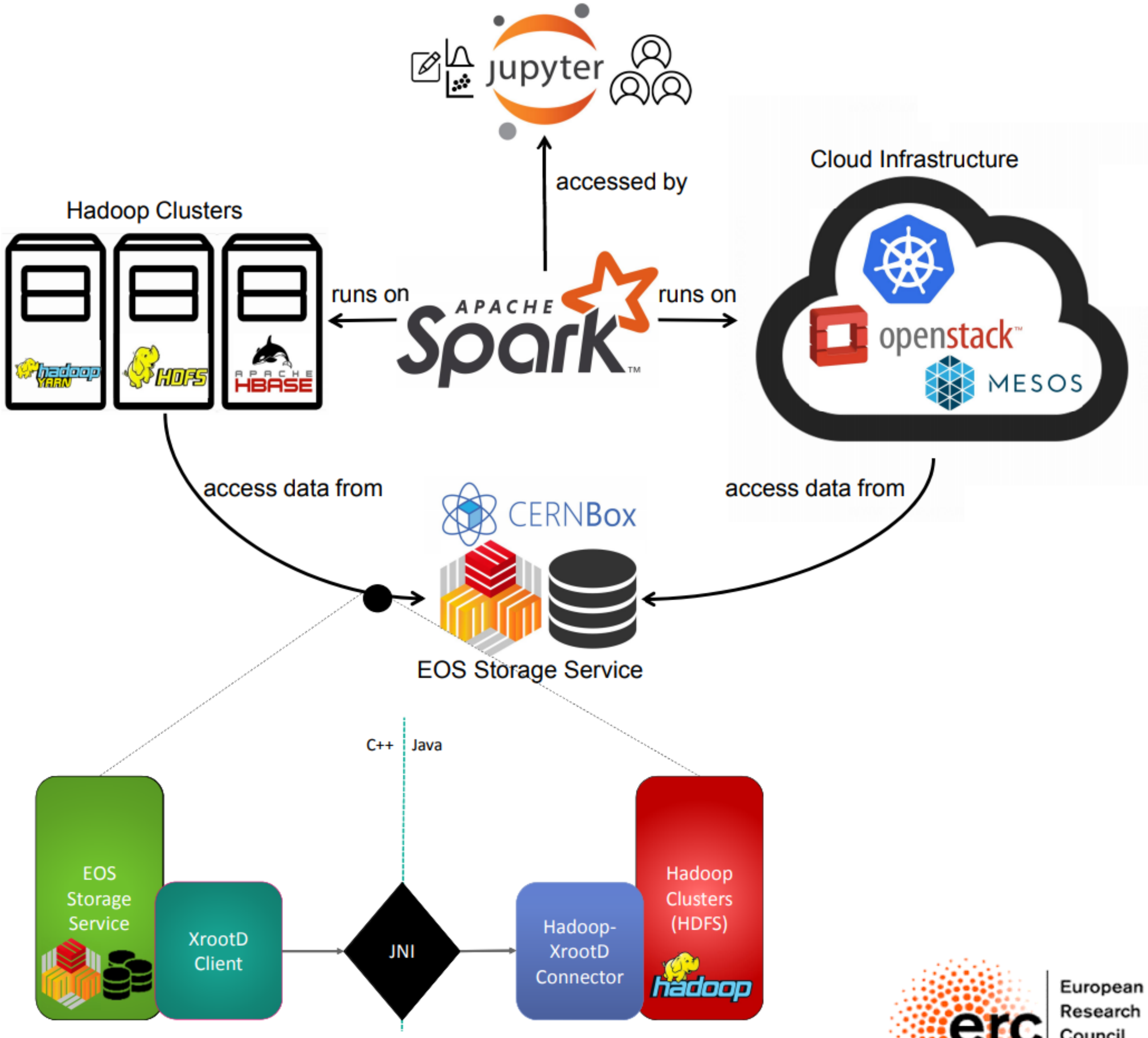
The Final Goal

- *Develop a CMS analysis workflow in Apache Spark:*
- *Full ROOT -> Spark analysis workflow with*
  - *Event selections*
  - *Data-Simulation comparison*
  - *Data reduction scheme in Spark*
- *Provided services*
  - *Machine-Learning toolkit*
  - *Data in memory for fast training*
- ***Benchmarking all that and compare performance/results with standard workflow***



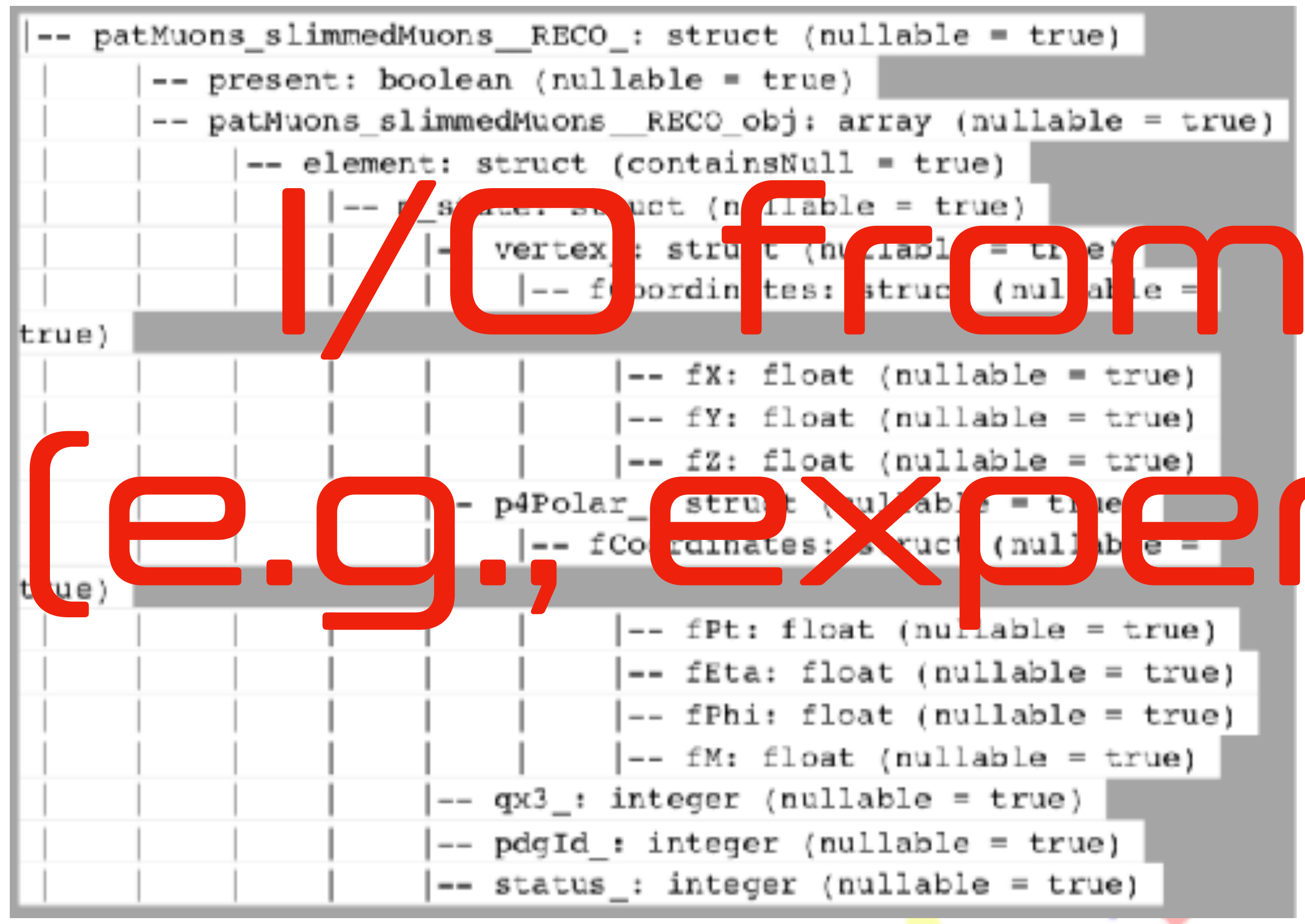
How It works

- Dedicated libraries to implement the workflow:
  - XrootD connector to access files on CERN EOS filesystem: (So far) from public area. Authentication via certificate is under developement
  - Spark-root: Read ROOT object collections and automatically infer their class schema
  - Histogrammar (by DIANA-HEP): To fill histograms passing lambda functions and use them in the same way as transformations are used in Apache Spark
- 100% data-science echosystem, compatible with ROOT I/O but w/o ROOT installation



European Research Council

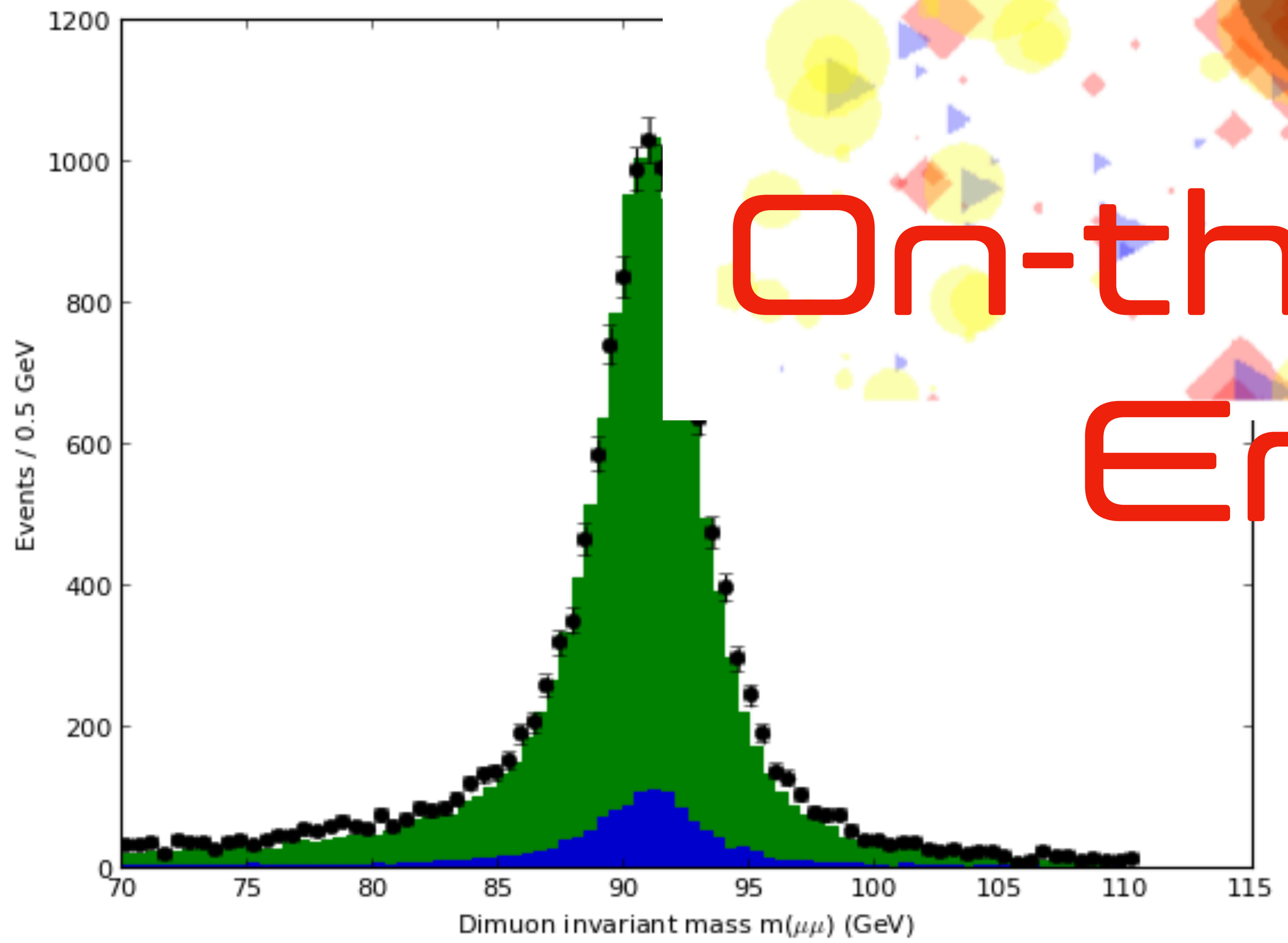
What Can It Do?

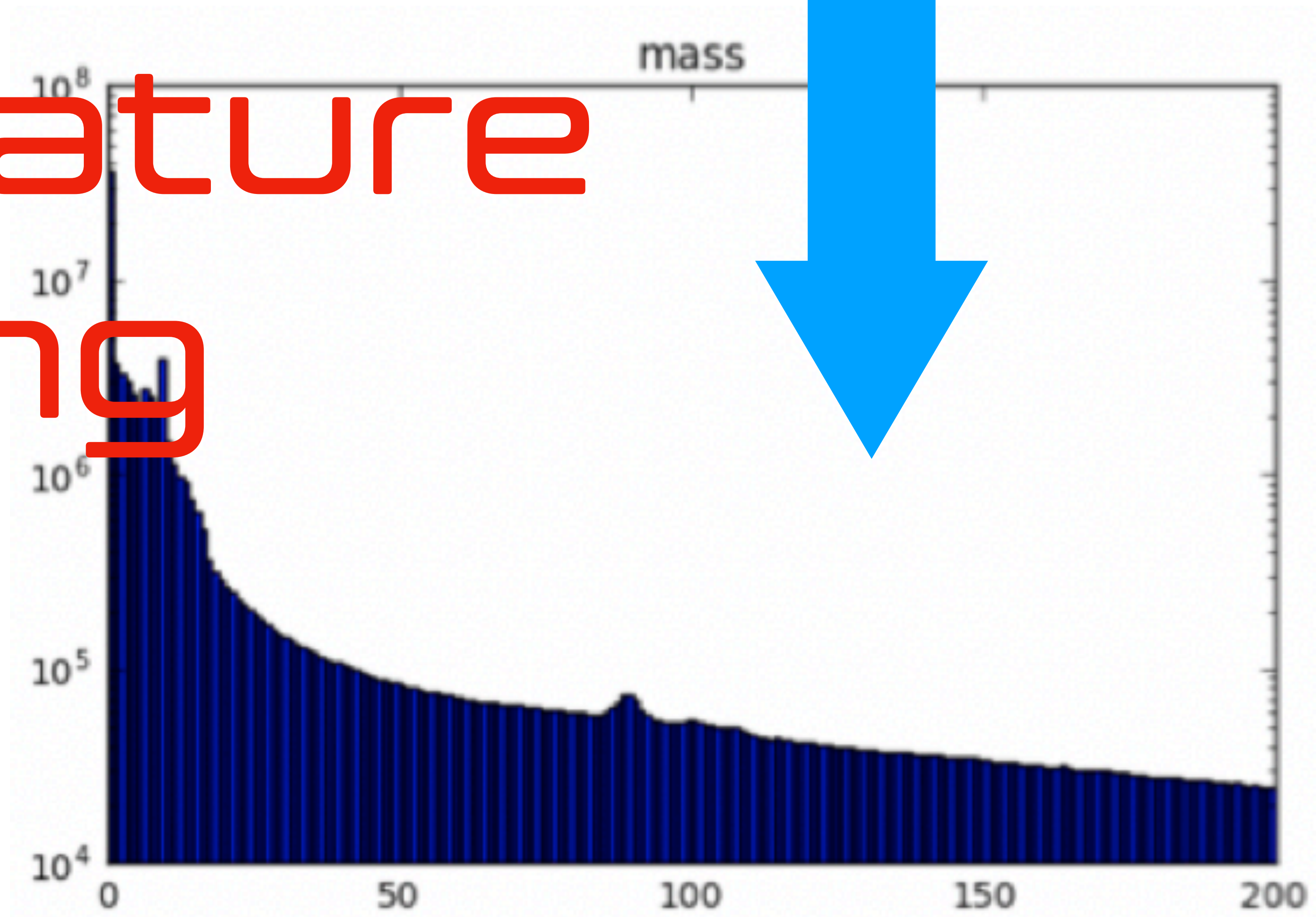


I/O from structured ROOT files (e.g., experiment-specific data files)

```
# read in the data
df = sqlContext.read\
    .format("org.dianahep.sparkroot.experimental")\
    .load("hdfs://path/to/files/*.root")

# count the number of rows:
df.count()

# select only muons
muons = df.select("patMuons_slimmedMuons__RECO_.patMuons_slimmedMuons__RECO_obj.m_state").toDF("muons")

# map each event to an invariant mass
inv_masses = muons.rdd.map(toInvMass)

# Use histogrammar to perform aggregations
empty = histogrammar.Bin(200, 0, 200, lambda row: row.mass)
h_inv_masses = inv_masses.aggregate(empty,
    histogrammar.increment,
    histogrammar.combine)
```

On-the-fly Feature Engineering

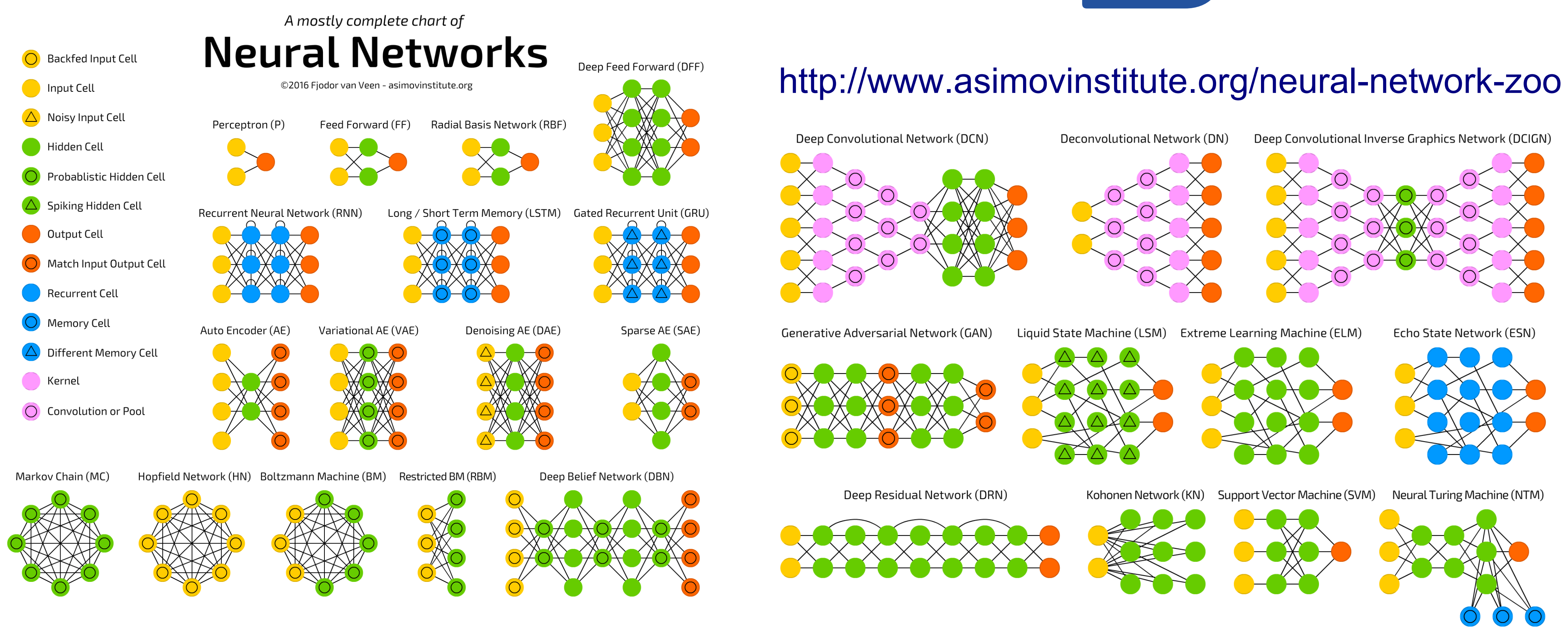


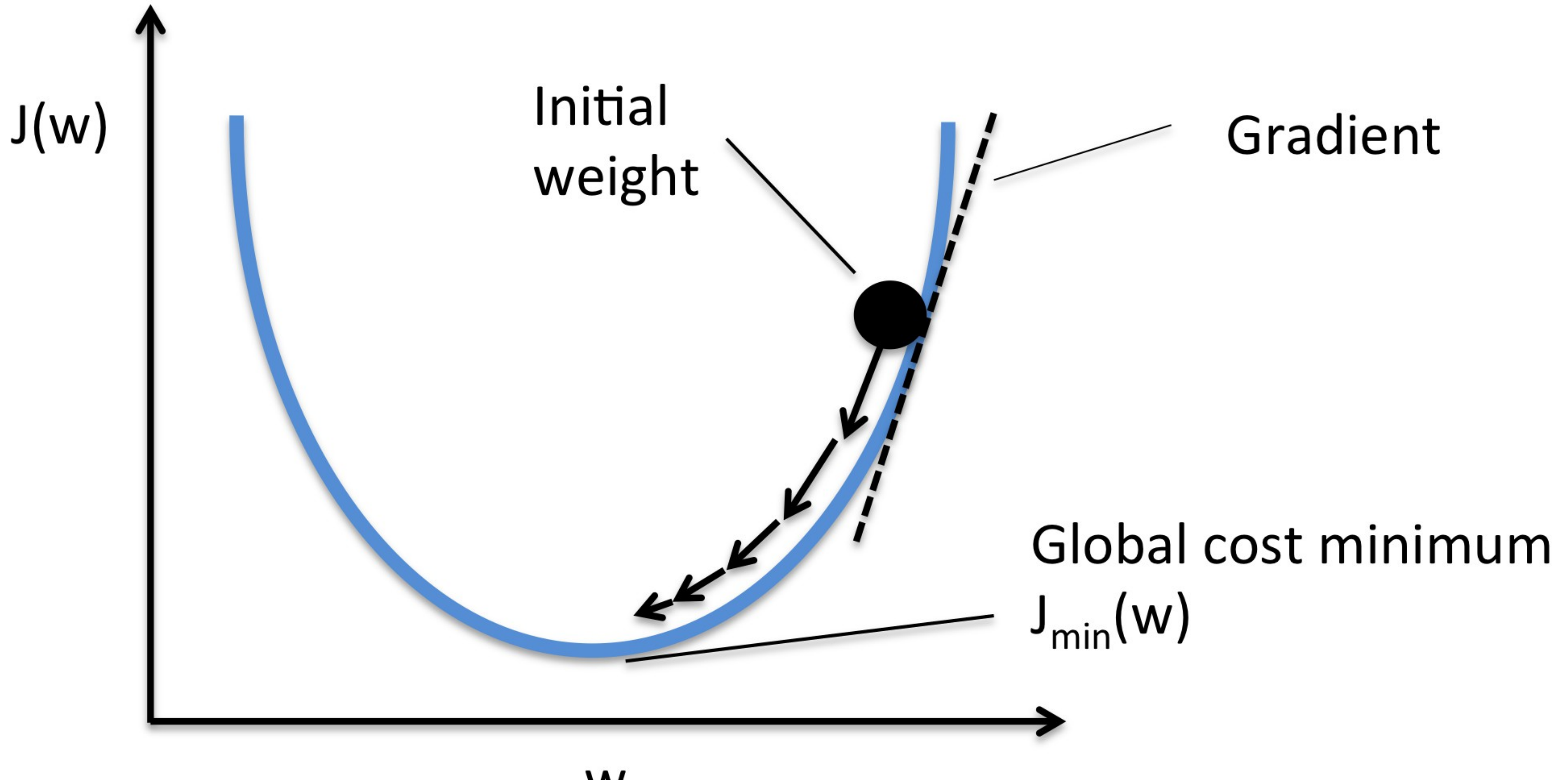


HPC, HEP & Deep Learning

Large-scale training

- With Deep Learning gaining territory in HEP(*), NN training will become soon a new workflow for large HEP experiments
  - Experiments will want to maximise performances
  - Fast turn-around for new trainings, as long as new data are collected
  - Need dedicated hardware to be effective (TPU, GPU, etc)
- Ideal use case to integrate HEP workflows into network of HPC sites

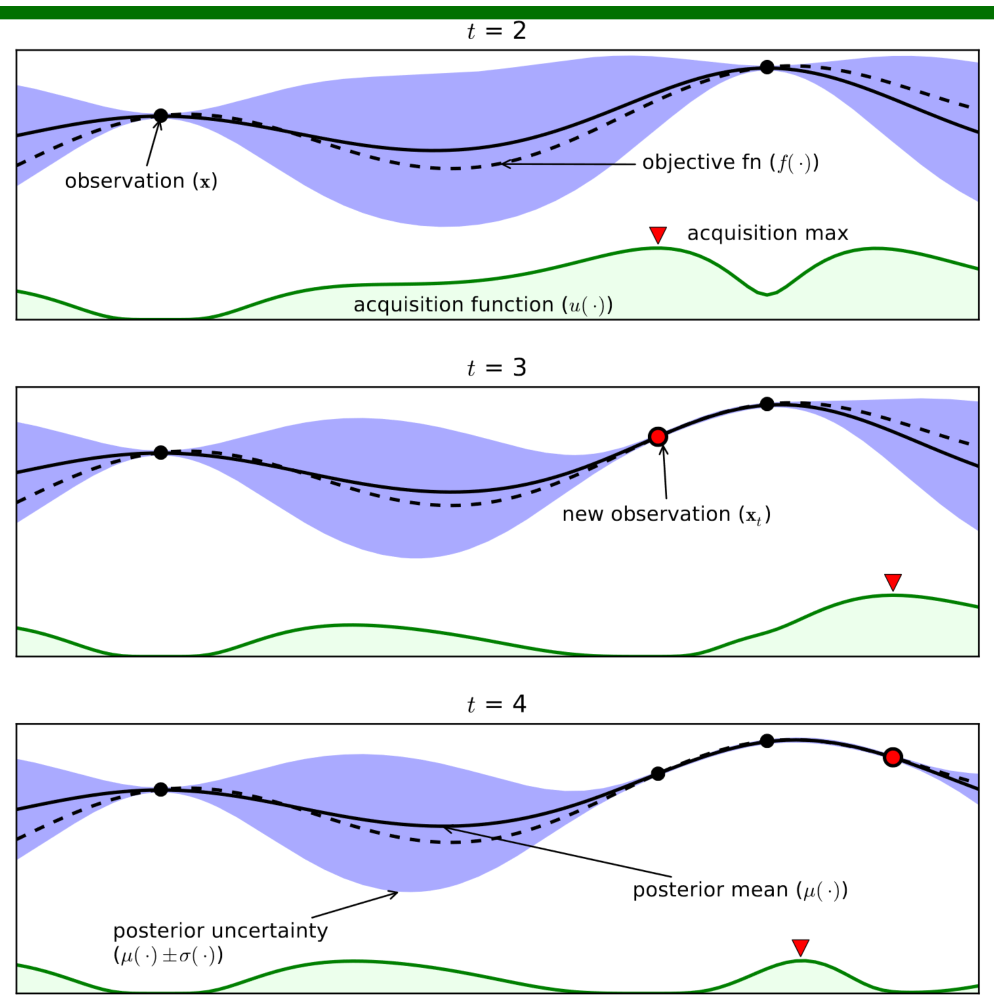




(*) more on this in afternoon seminar

Hyper-parameter optimization

- *Not only the best set of parameters, but also the best network overall:*
  - *how many layers?*
  - *how many nodes/layers?*
  - *which activation function?*
- *Answers to be find by Optimization algorithm*
  - ***Bayesian Optimization***
  - ***Evolutionary Algorithms***
  - *…*
- ***One extra reason to train production-ready algorithms @HPC sites***



https://tinyurl.com/yc2phuaj

- Objective function is approximated as a multivariate gaussian
- Measurements provided one by one to improve knowledge of the objective function
- Next best parameter to test is determined from the acquisition function
- Using the python implementation from https://scikit-optimize.github.io

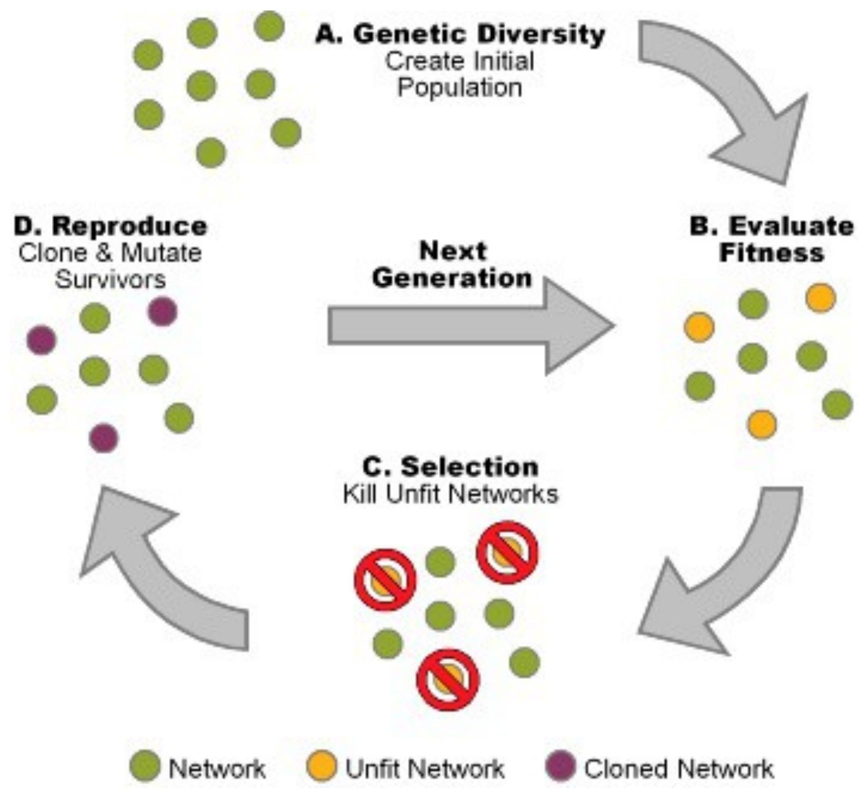
- Chromosome crossover:
  - Let Parent A be more fit than Parent B
  - For each parameter p, generate a random number r in (0, 1) to find p_{child}

$$p_{child} = (r)(p_{Parent\ A} - p_{Parent\ B}) + p_{Parent\ A}$$

- Non-uniform mutation (Michalewicz):
  - In generation g out of a total G generations, for each parameter p in a child, generate random numbers $r_1, r_2 \in (0,1)$ to define a mutation m:

$$m = \left(1 - r_1^{\left(1-\frac{g}{G}\right)^3}\right) * \begin{cases} (p_{MAX} - p_{child}) & IF\ \ r_2 > 0.5 \\ (p_{LOW} - p_{child}) & IF\ \ r_2 \leq 0.5 \end{cases}$$

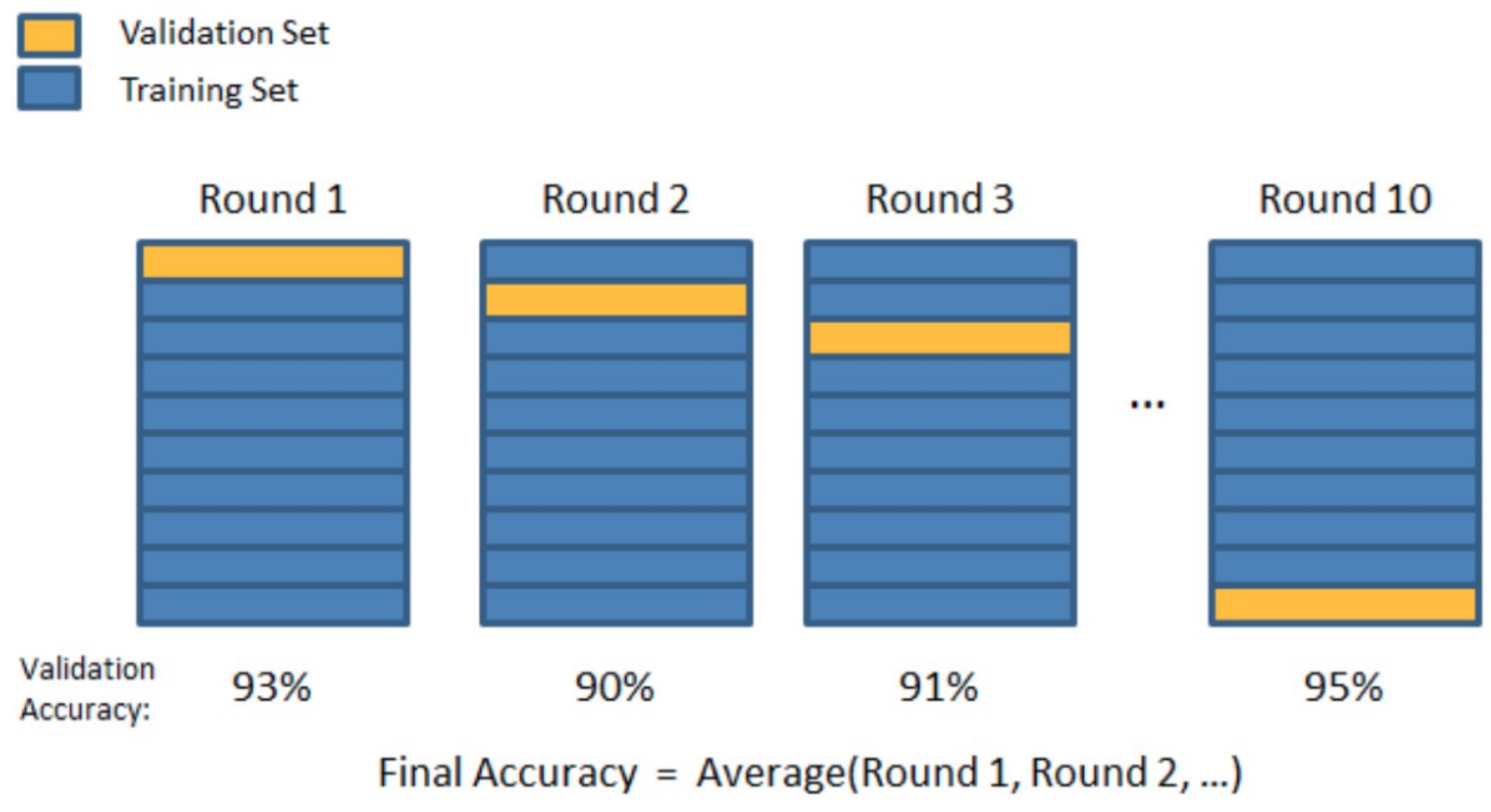
$$p_{child} = p_{child} + m$$



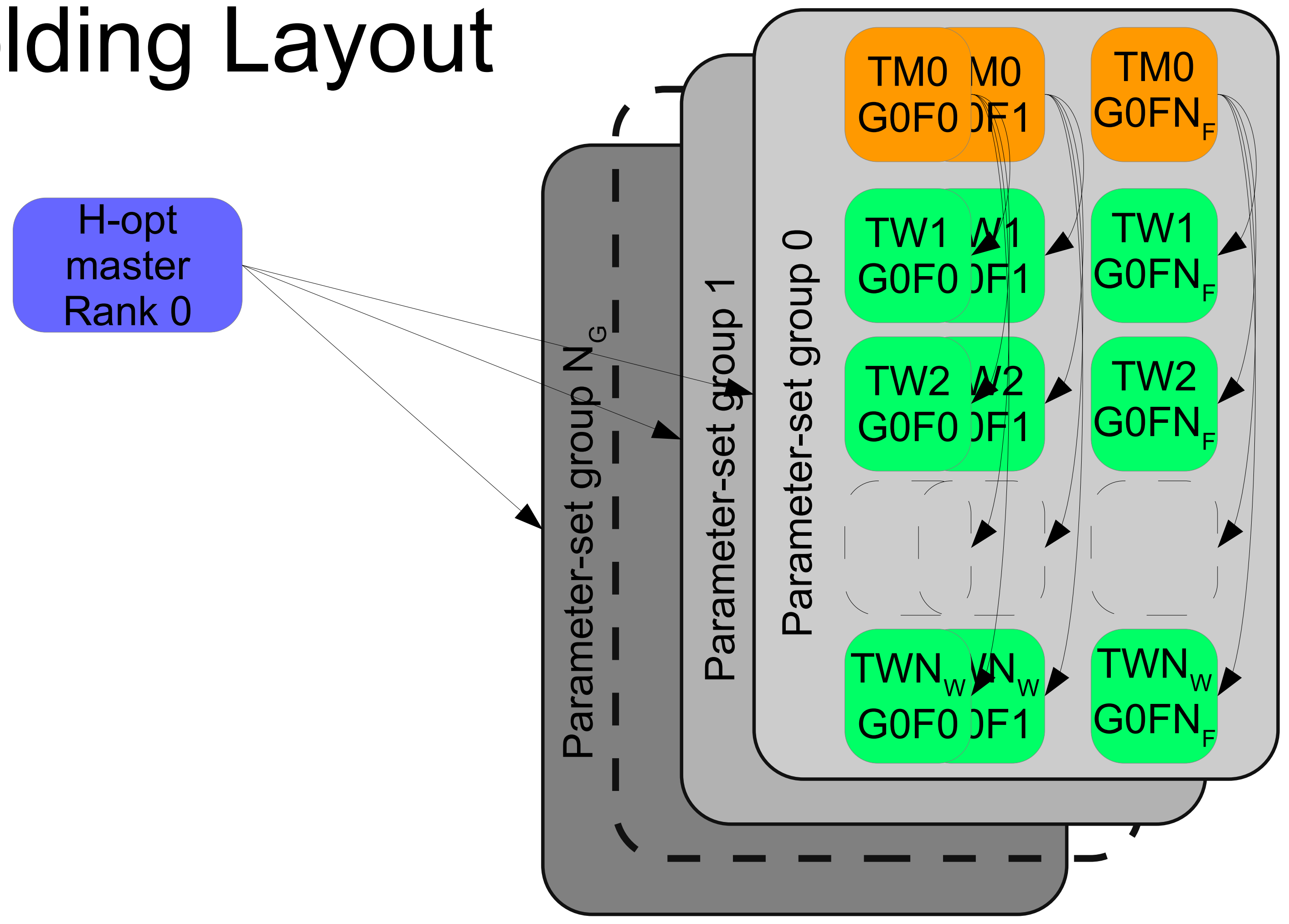
European Research Council

K-folding cross validation

- Assigning uncertainties to training-goodness figures of merit to establish ACTUAL improvements
  - Are ROC AUCs 98.754 and 97.998 actually different?
- Done by different training vs validation dataset splits
- Average performances and performance dispersion allow to "measure" mean and variance
- Multiply workflow computing needs by K



K-folding Layout

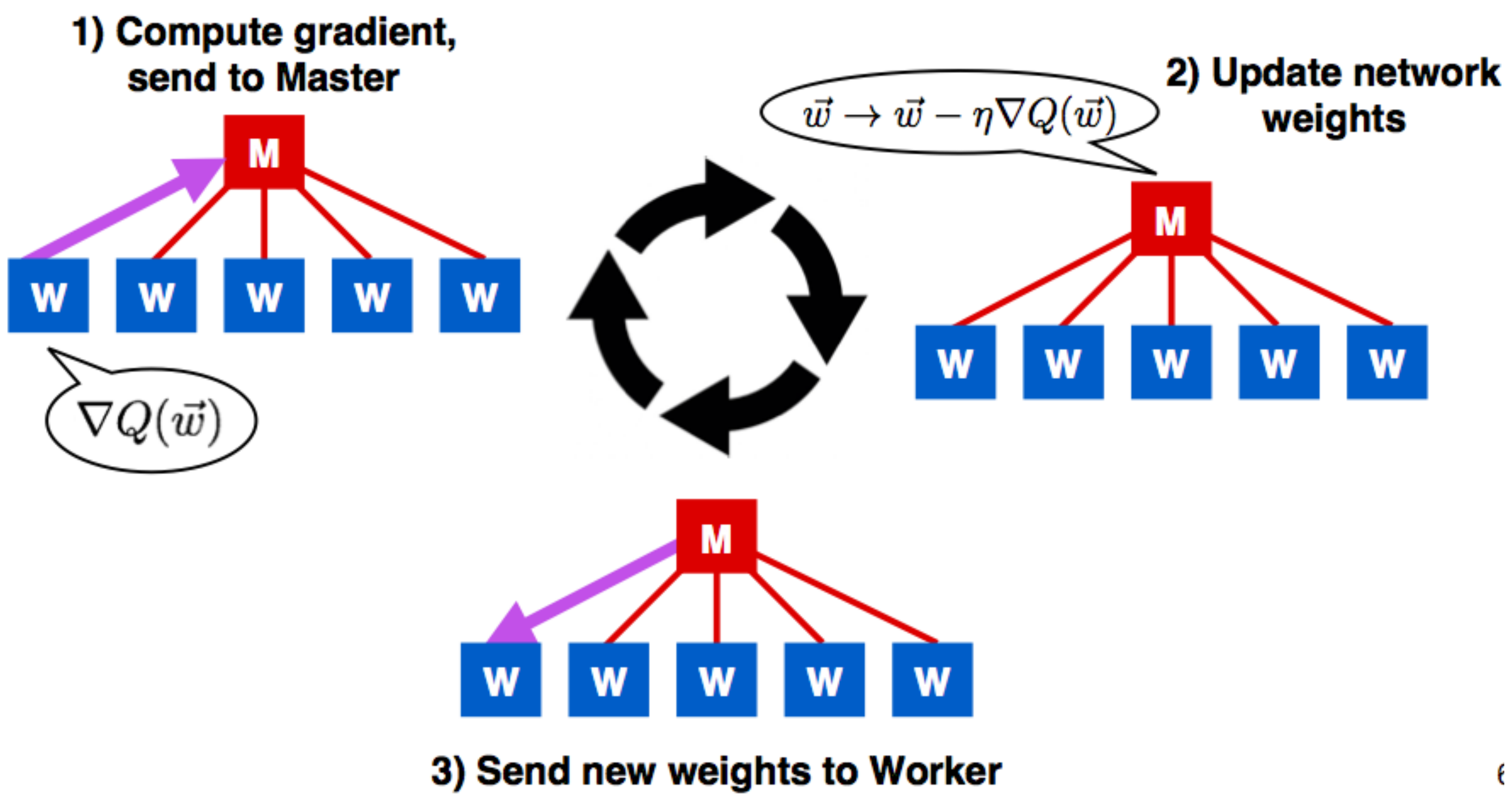


- One master running the optimization. Receiving the average figure of merit over N_F folds of the data
  - N_G groups of nodes training on a parameter-set on simultaneously
  - N_F groups of nodes running one fold each

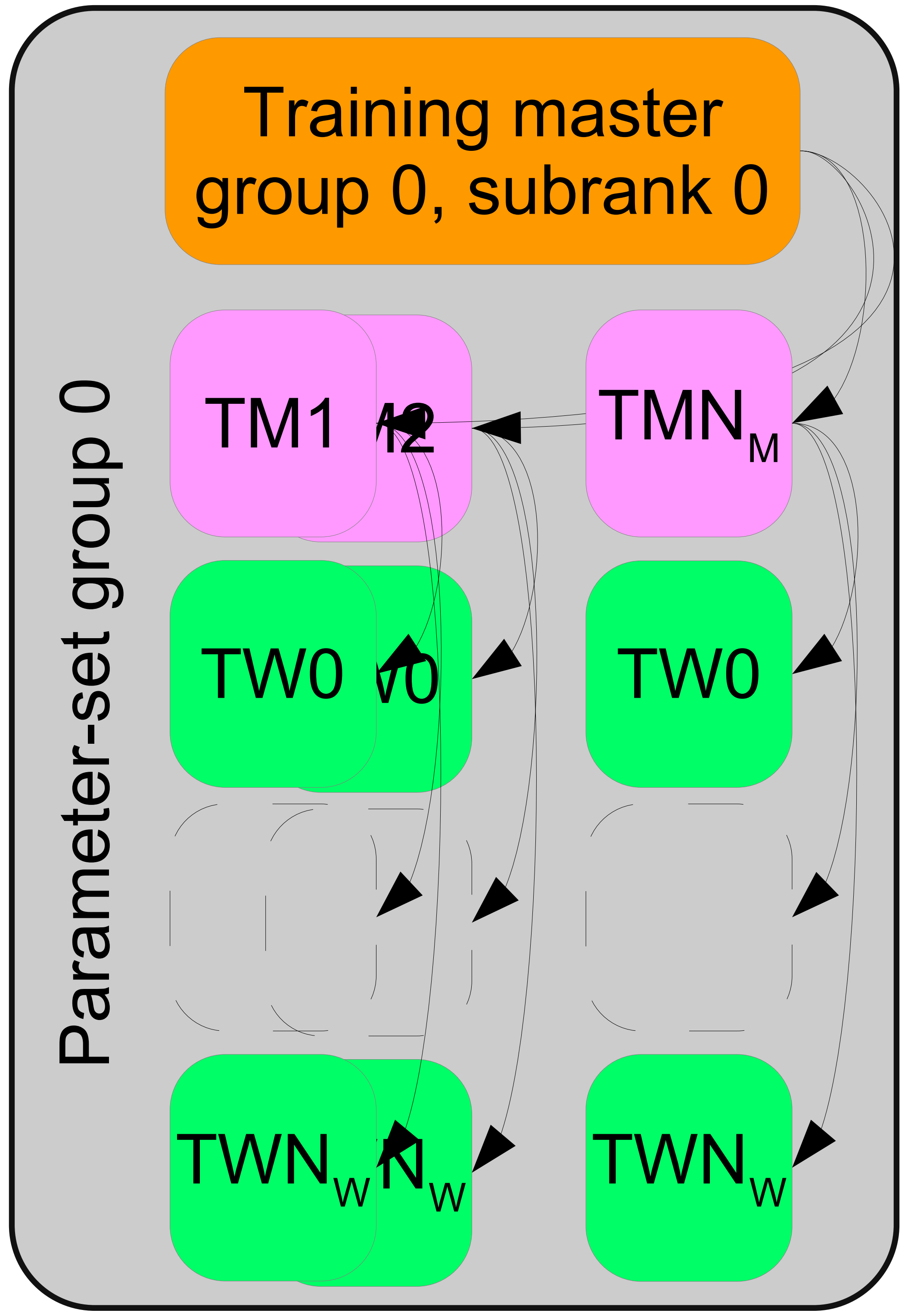
European Research Council

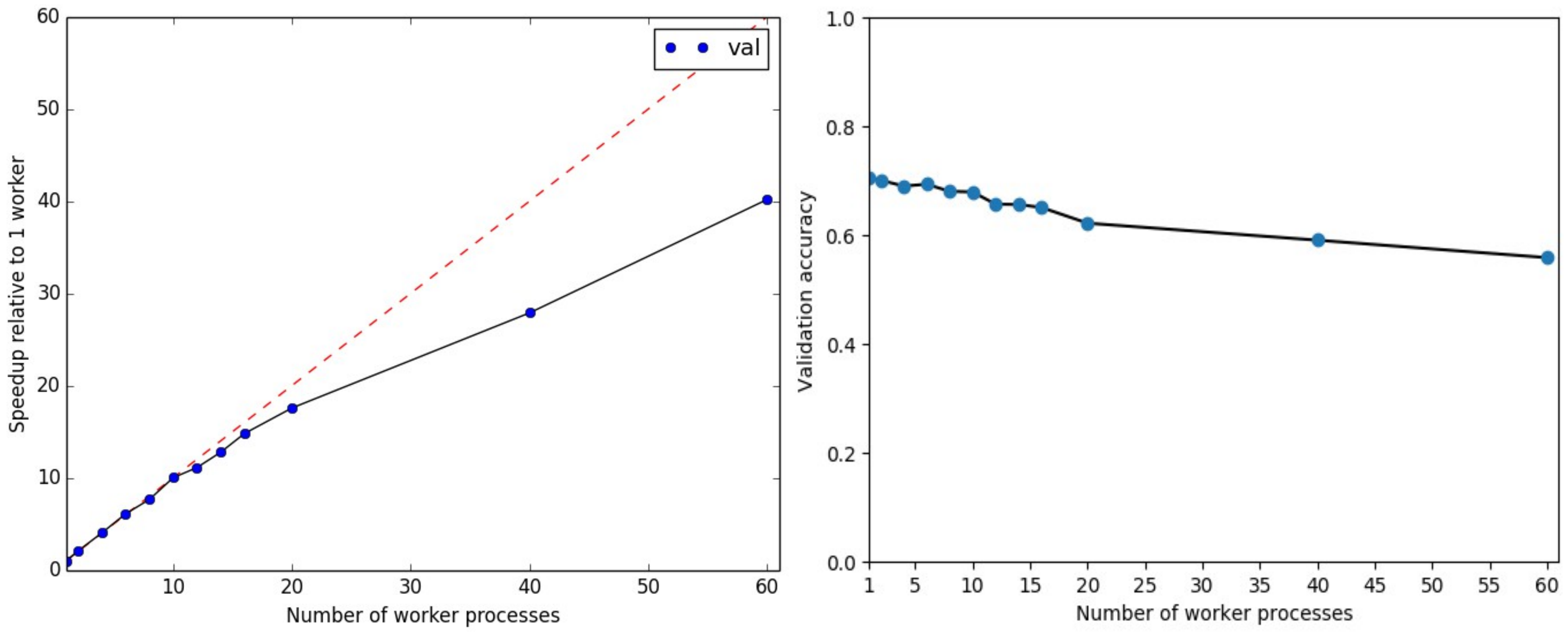
Parallelisms

- ***Data Parallelism:*** *master nodes handle parameter setting, receives gradients from workers and distribute new parameter values. Good for datasets with many events*



https://arxiv.org/abs/1712.05878

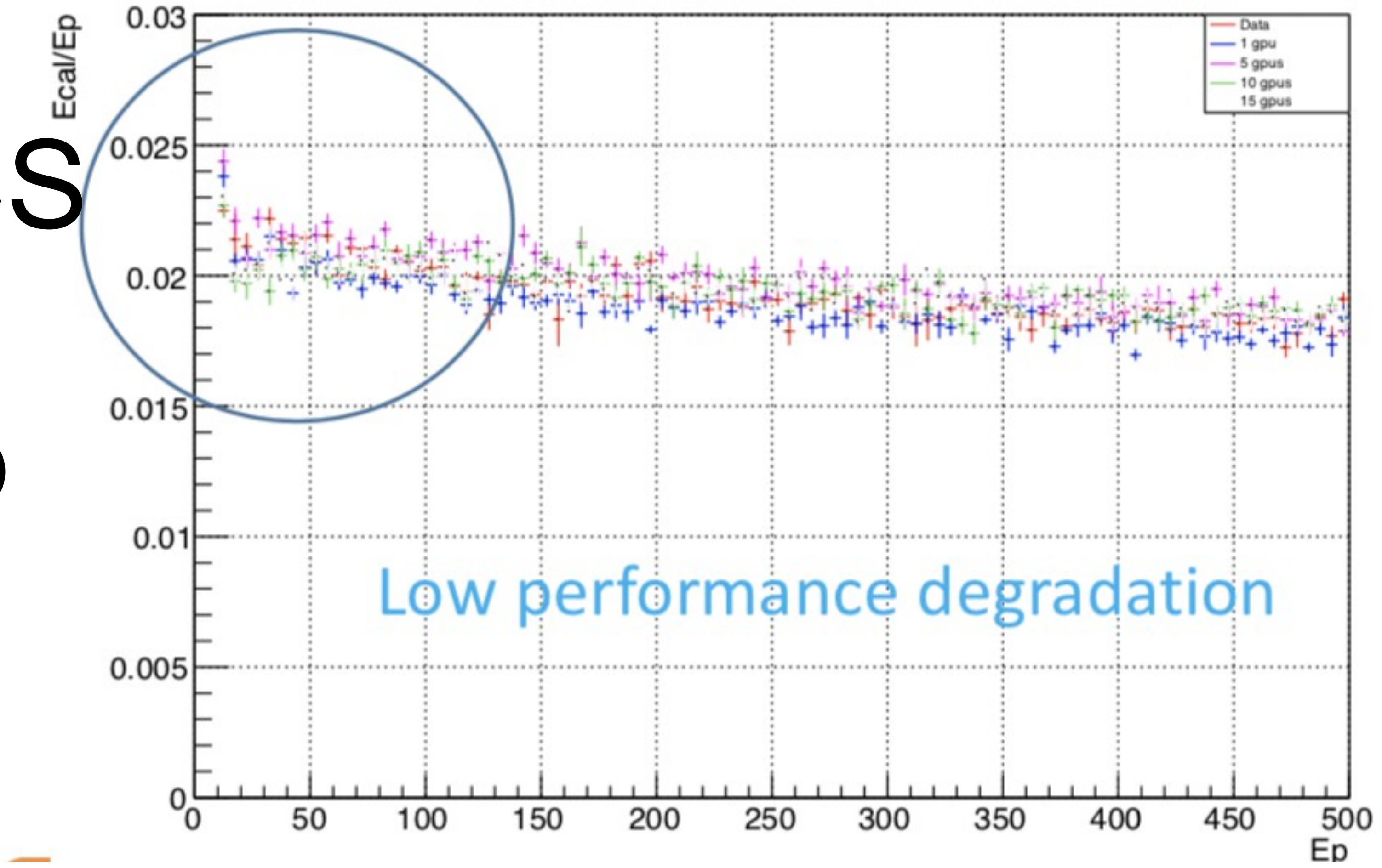


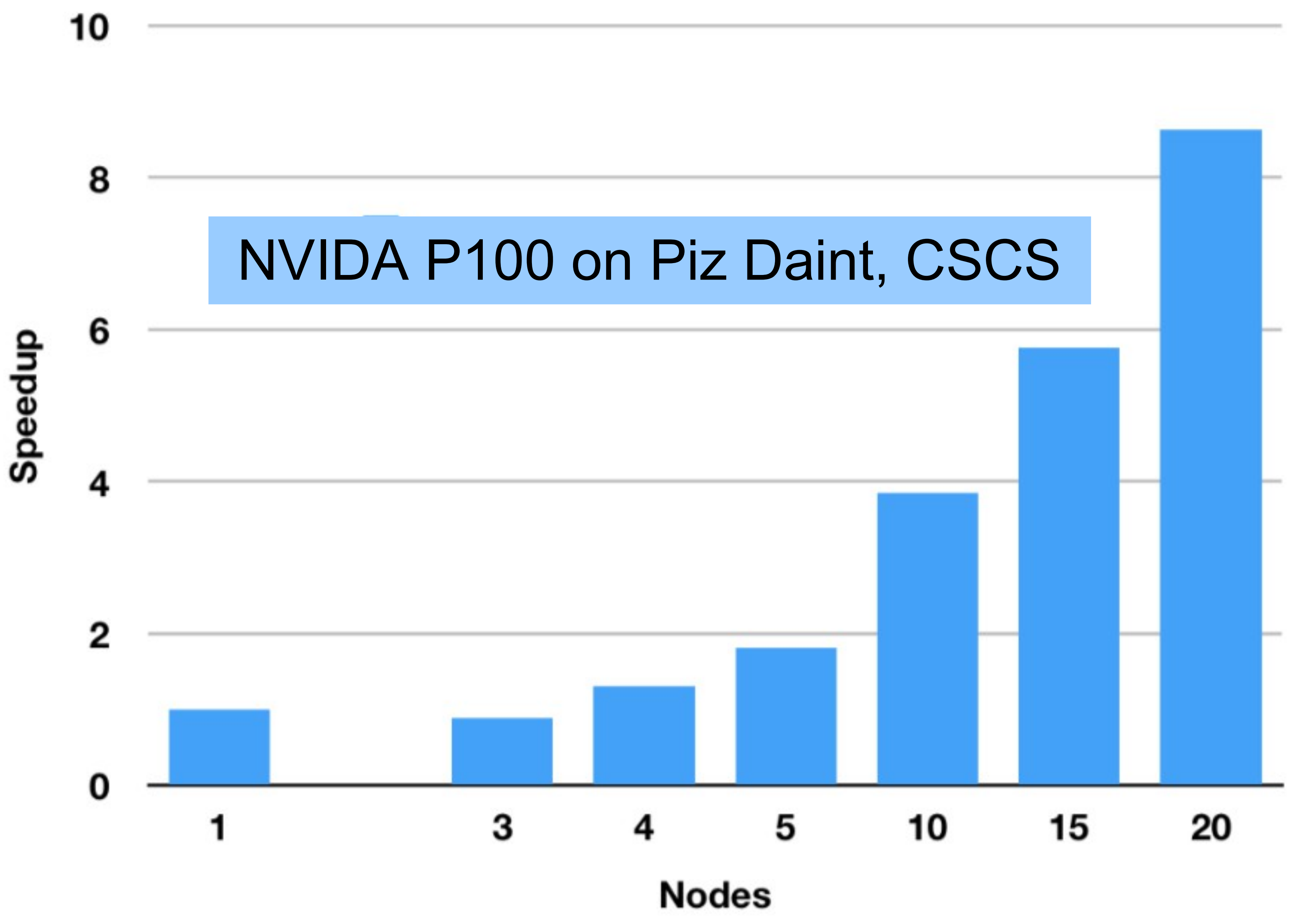


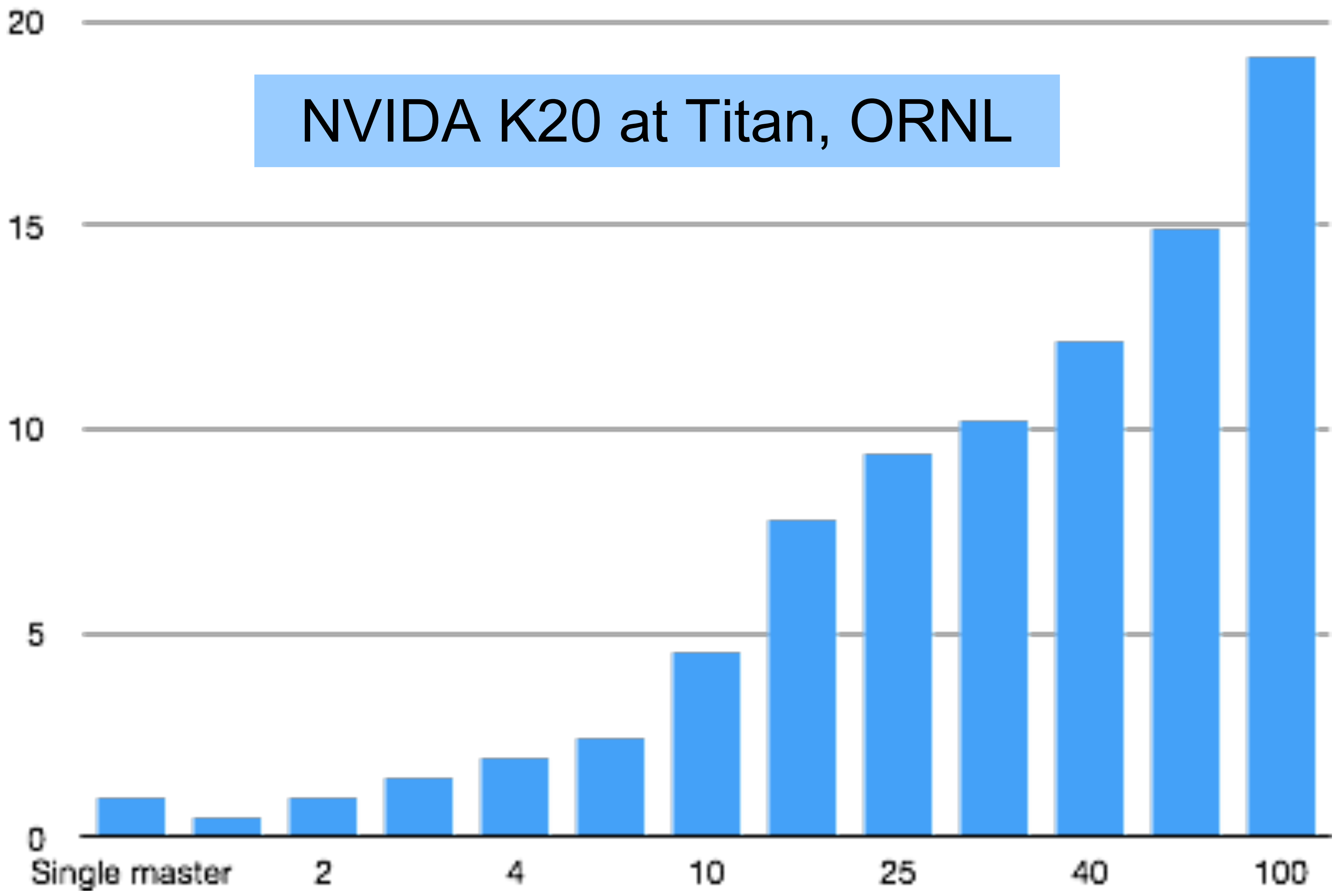
https://github.com/duanders/mpi_learn

Deployed @HPC centres

- Speed up in training generative adversarial networks on Piz Daint CSCS and Titan ORNL supercomputers
  - ➔ Using easgd algorithm with rmsprop
  - ➔ Speed up is not fully efficient. Bottlenecks to be identified

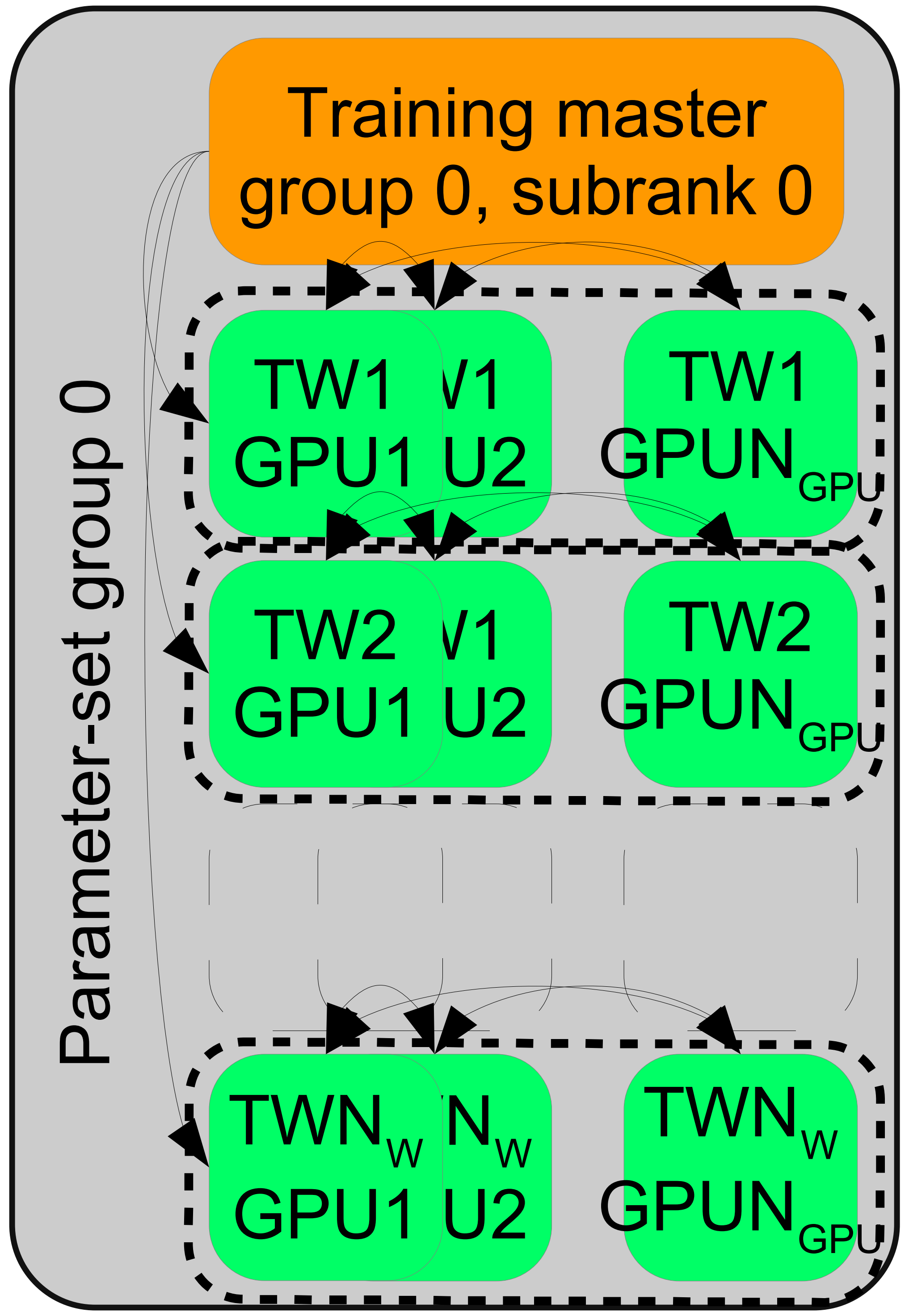






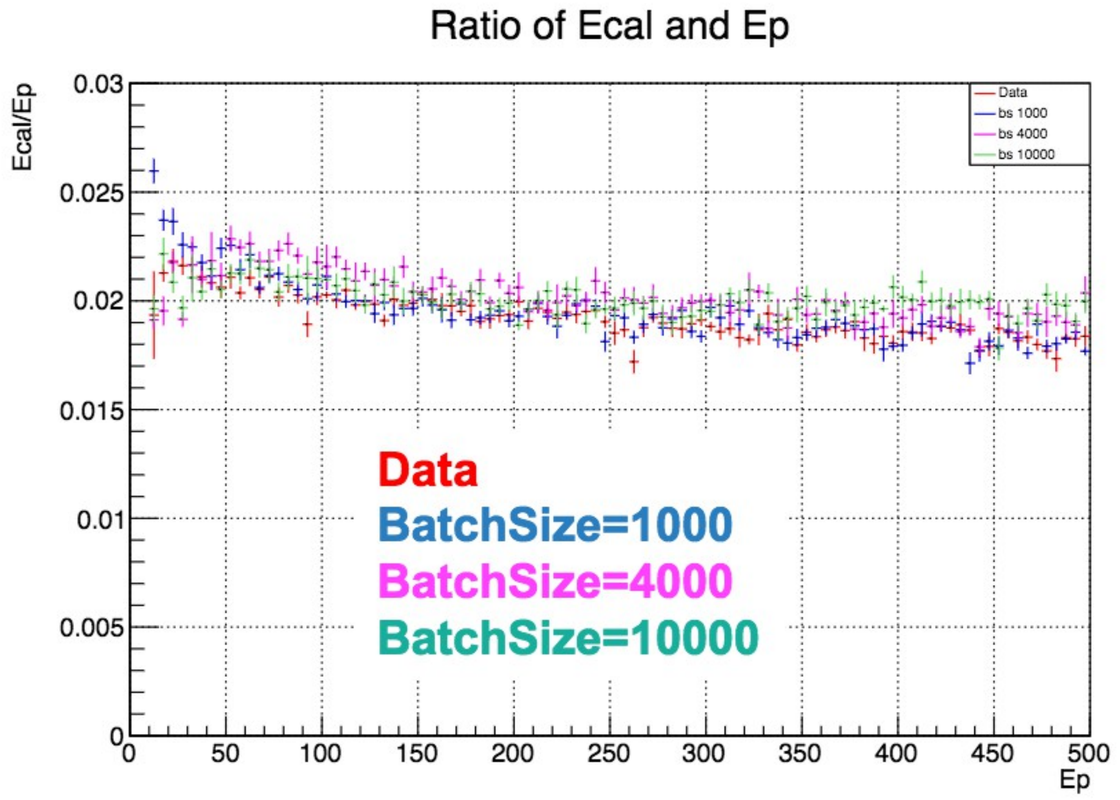
Parallelisms

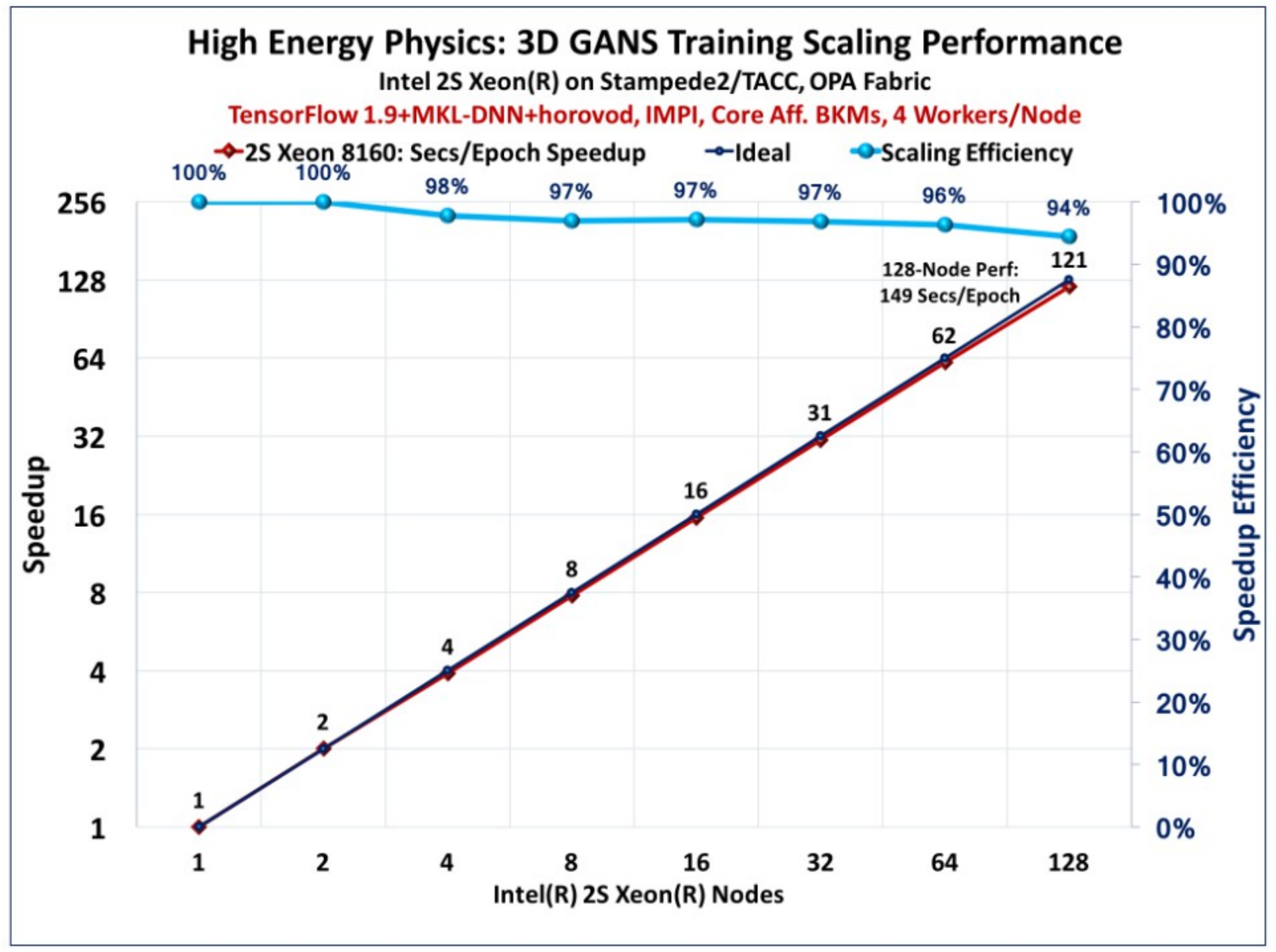
- *Gradient Parallelism: parallelise gradient computation of a single batch on multiple workers. Good for datasets with large-size examples*



Communication through Horovod with fast GPU-GPU communication (nVidia NCCL)

Some performance degradation
Mostly in the low energy regions for large batchsize

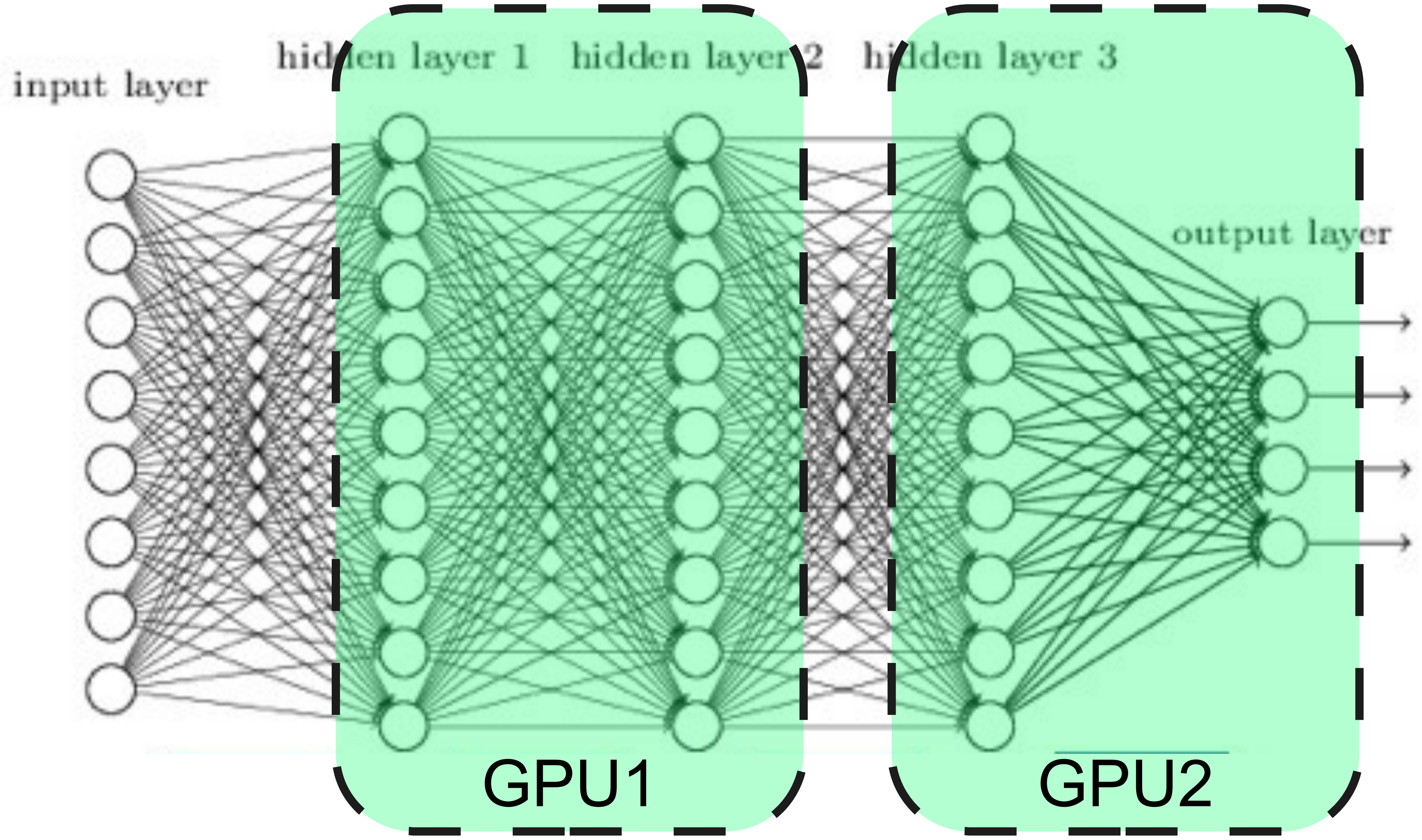




Sofia V. @ https://sites.google.com/nvidia.com/ai-hpc

Parallelisms

- *Model Parallelism: compute gradients for different parts of the networks on different workers. Good for large models*



Requires good devide-to-device communication
Used TensorFlow native multi-device manager
Aiming to test this on machines with multi-gpu nodes (Summit)

Conclusions

- *Large-scale computing is a consolidated tradition in HEP*
- *Things didn’t go as planned, since new developments made us more competitive at fixed/decreasing resource budget (e.g., scouting & real-time processing to do more with less)*
- *New challenges ahead call for qualitative and quantitative (more CPU + GPU + …) and qualitative (Big data tools integration) improvements*
- *Exploiting existing sites (HPC) rather than building dedicated facilities*
  - *Opportunistic cloud computing*
  - *Large-scale training as a service on GPU clusters*
- ***Challenges ahead, time for brave people to come out with ideas***

Backup

Not mpi-opt

Distributed training

Use keras 2.13 /Tensorflow 1.9 (Intel optimised)

- AVX512 –FMA-XLA support
- Intel® MKL-DNN (with 3D convolution support)

Optimised multicore utilisation

- inter_op_paralellism_threads/intra_op_paralellism threads

Horovod 0.13.4

- Synchronous SGD approach
- MPI_AllReduce

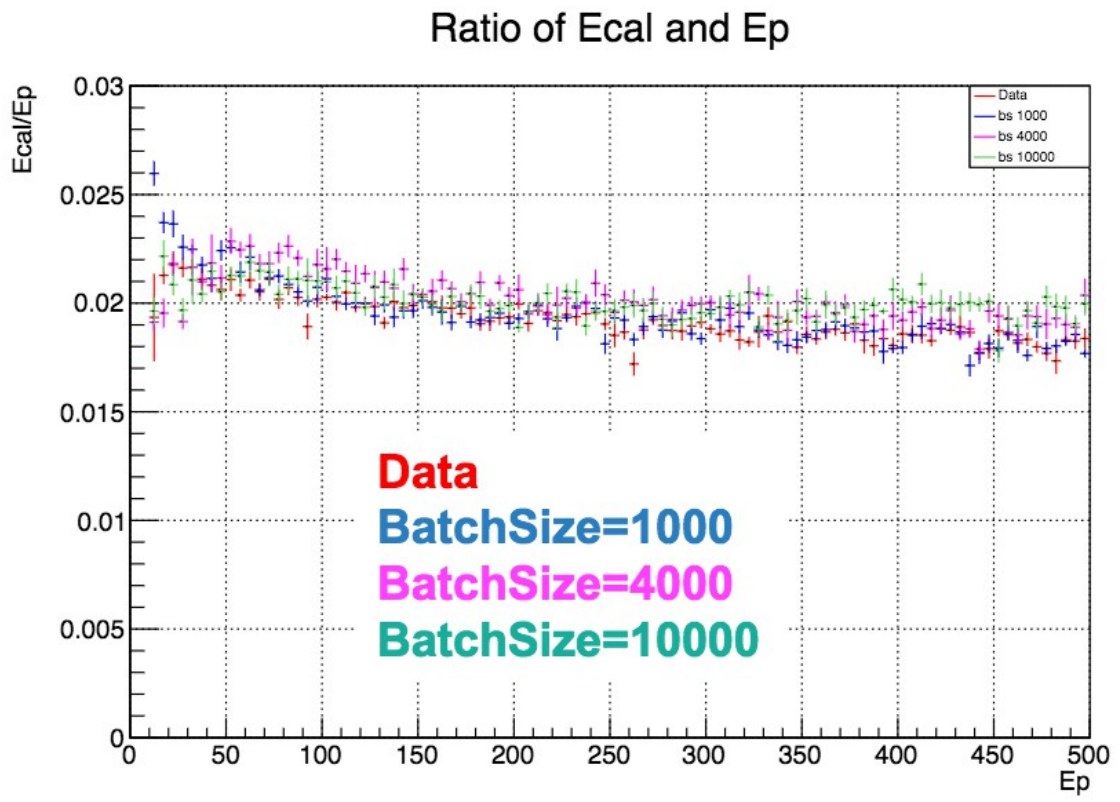
Run on TACC Stampede2 cluster:

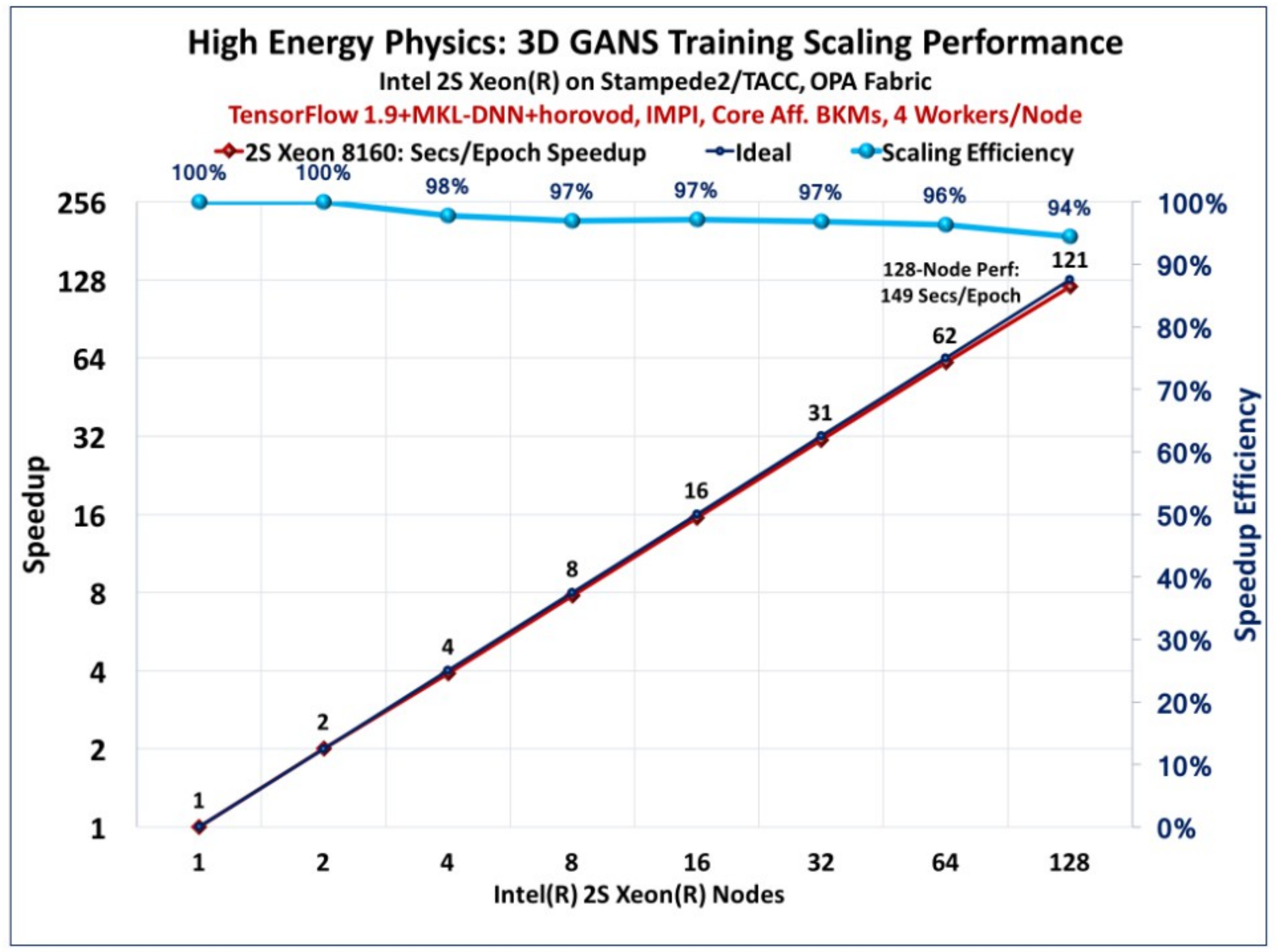
- Dual socket Intel Xeon 8160
- 2x 24 cores per node, 192 GB RAM
- Intel® Omni-Path Architecture

Test several MPI scheduling configurations

- 2,4, 8 processes per nodes.
- Best machine efficiency with 4 processes/node

Some performance degradation
Mostly in the low energy regions for large batchsize





14

Sofia V. @ https://sites.google.com/nvidia.com/ai-hpc