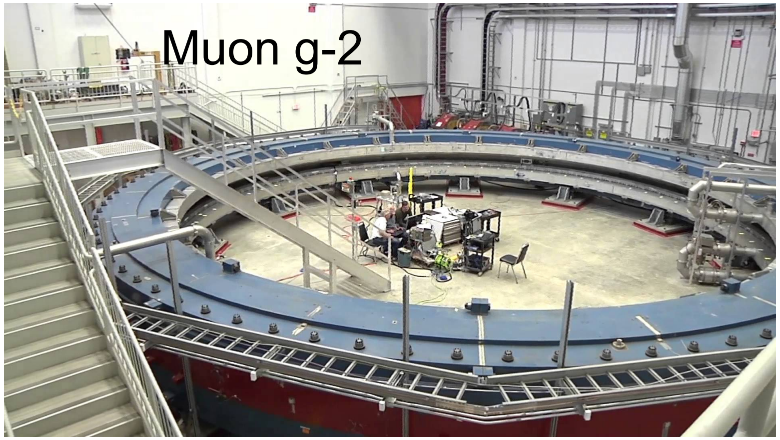
Scientific computing at Fermilab

Erica Snider
Fermilab

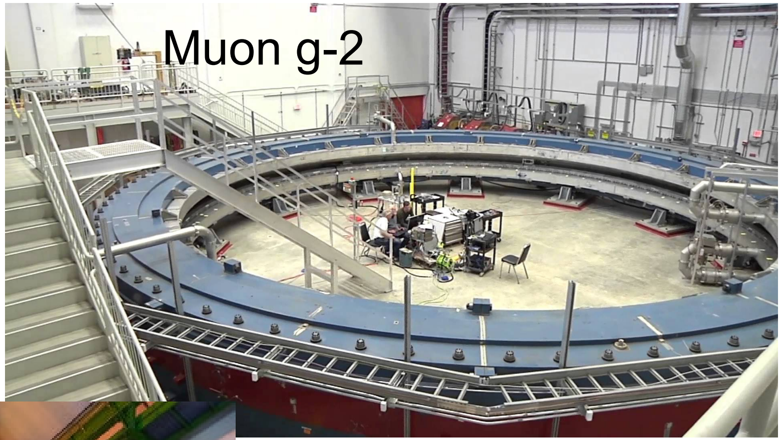
Fermilab Undergraduate Lecture Series
July 7, 2016

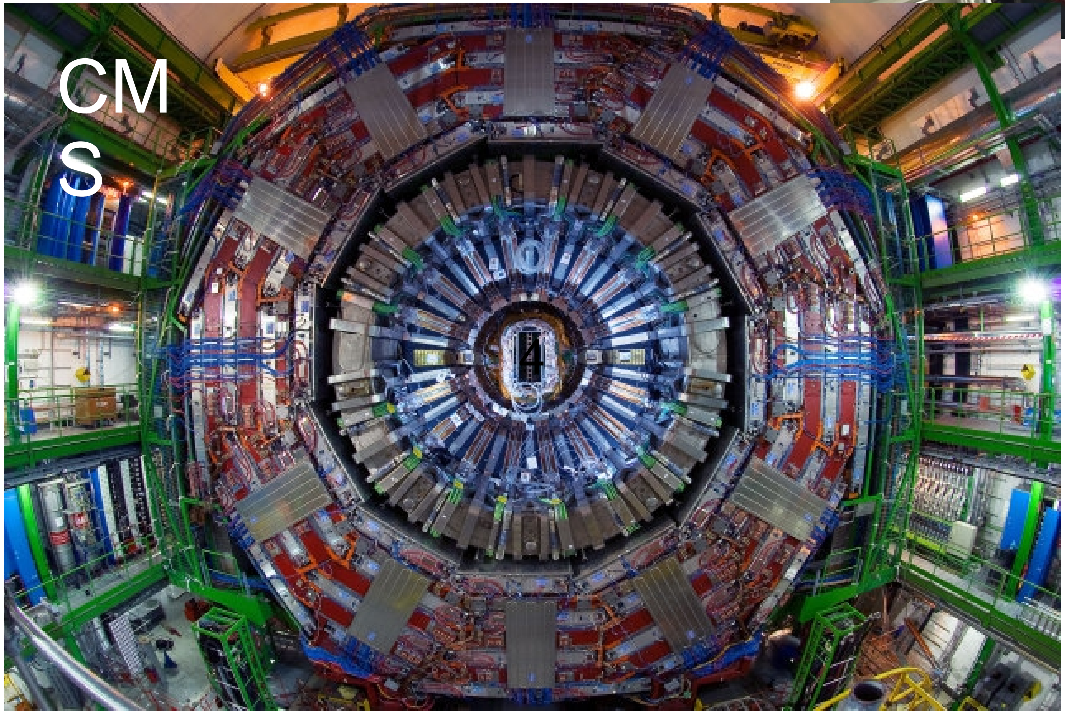

MicroBooNE
detector on the
move

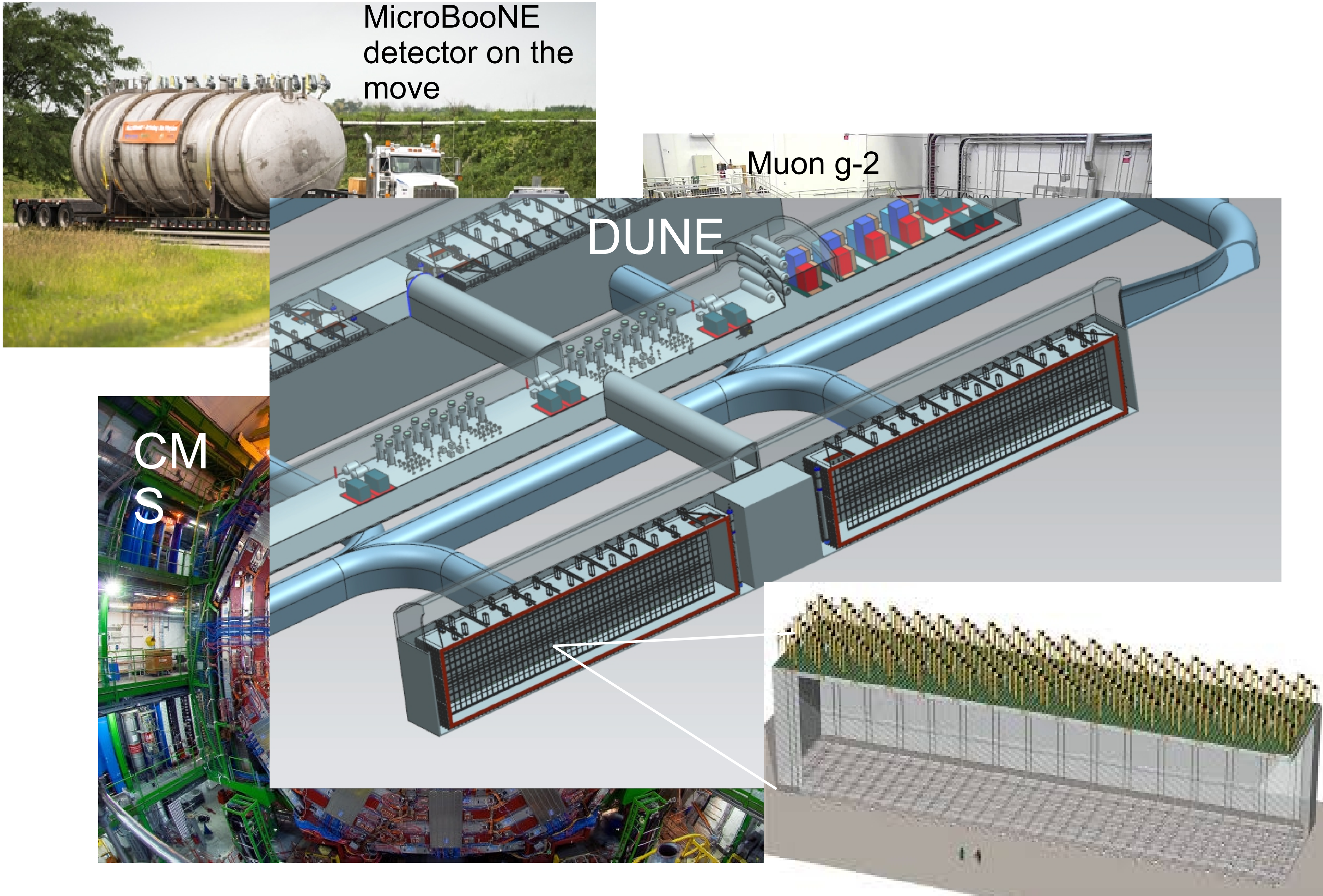

MicroBooNE
detector on the
move


Muon g-2

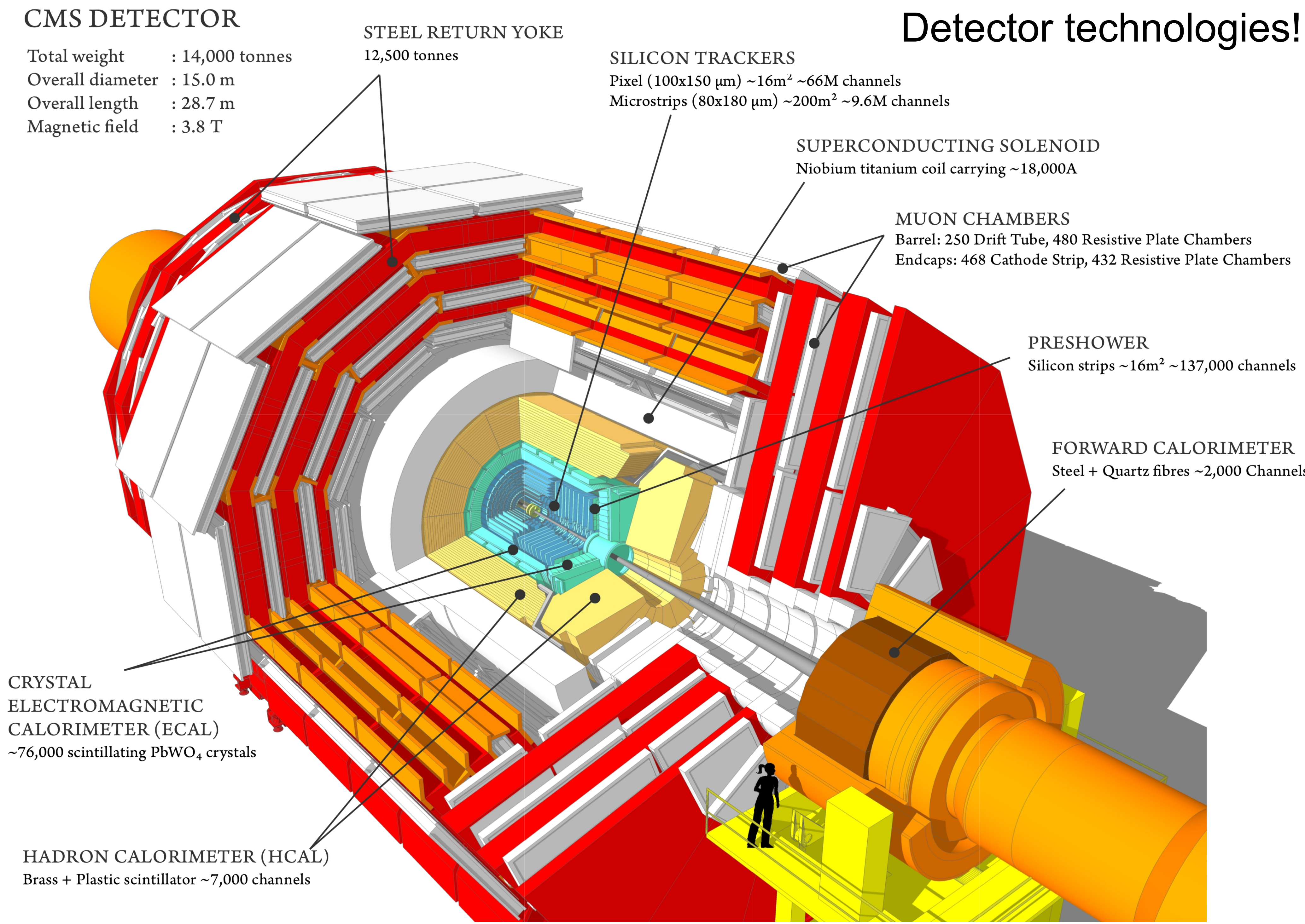

MicroBooNE detector on the move

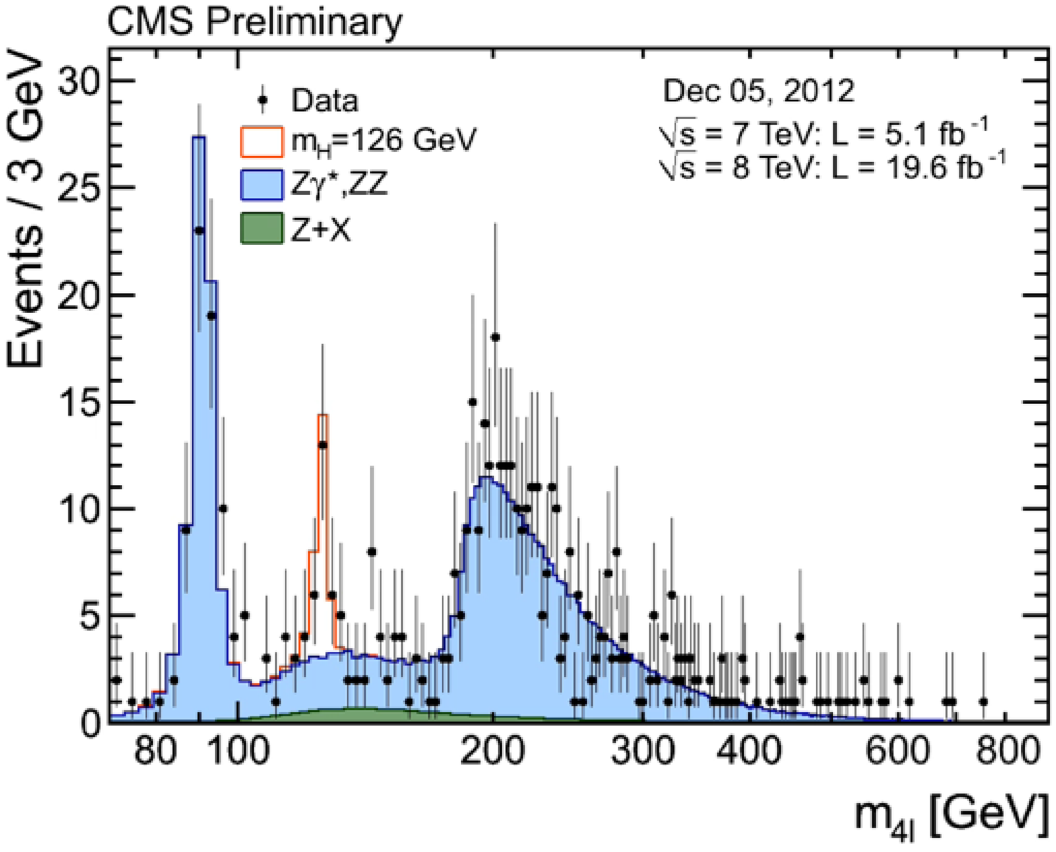

Muon g-2

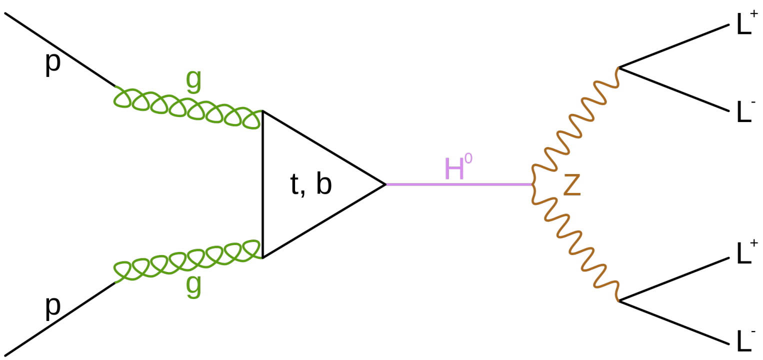

CMS


MicroBooNE detector on the move
Muon g-2
DUNE
CMS

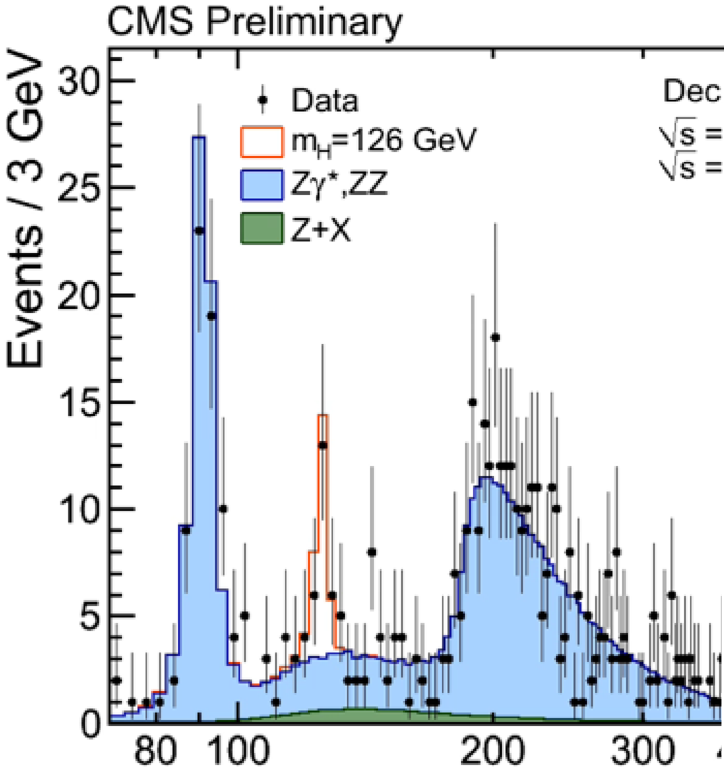
Detector technologies!

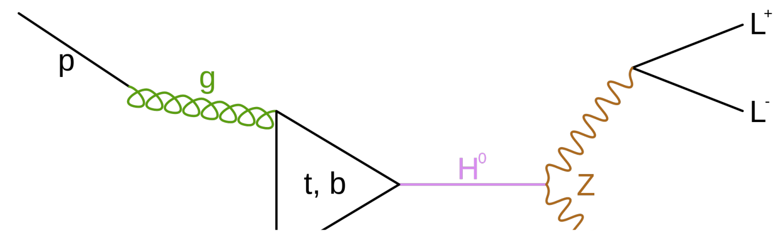




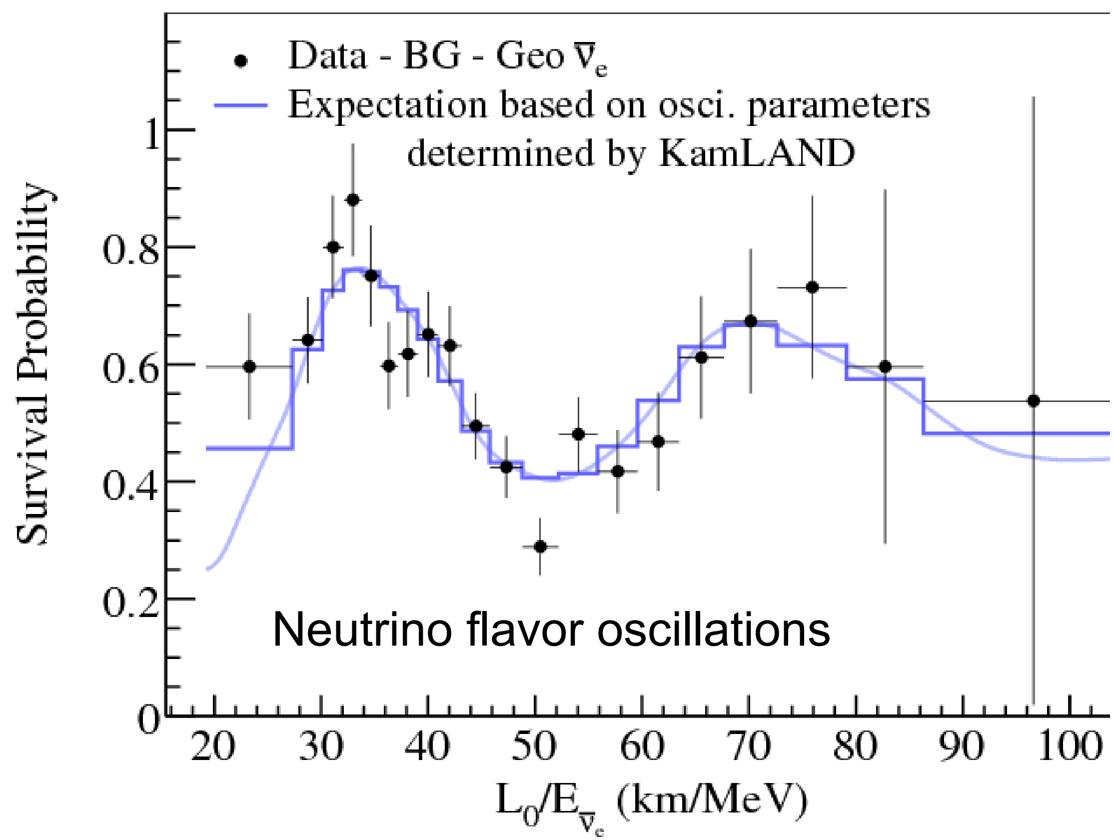


Higgs explains mass
of elementary particles

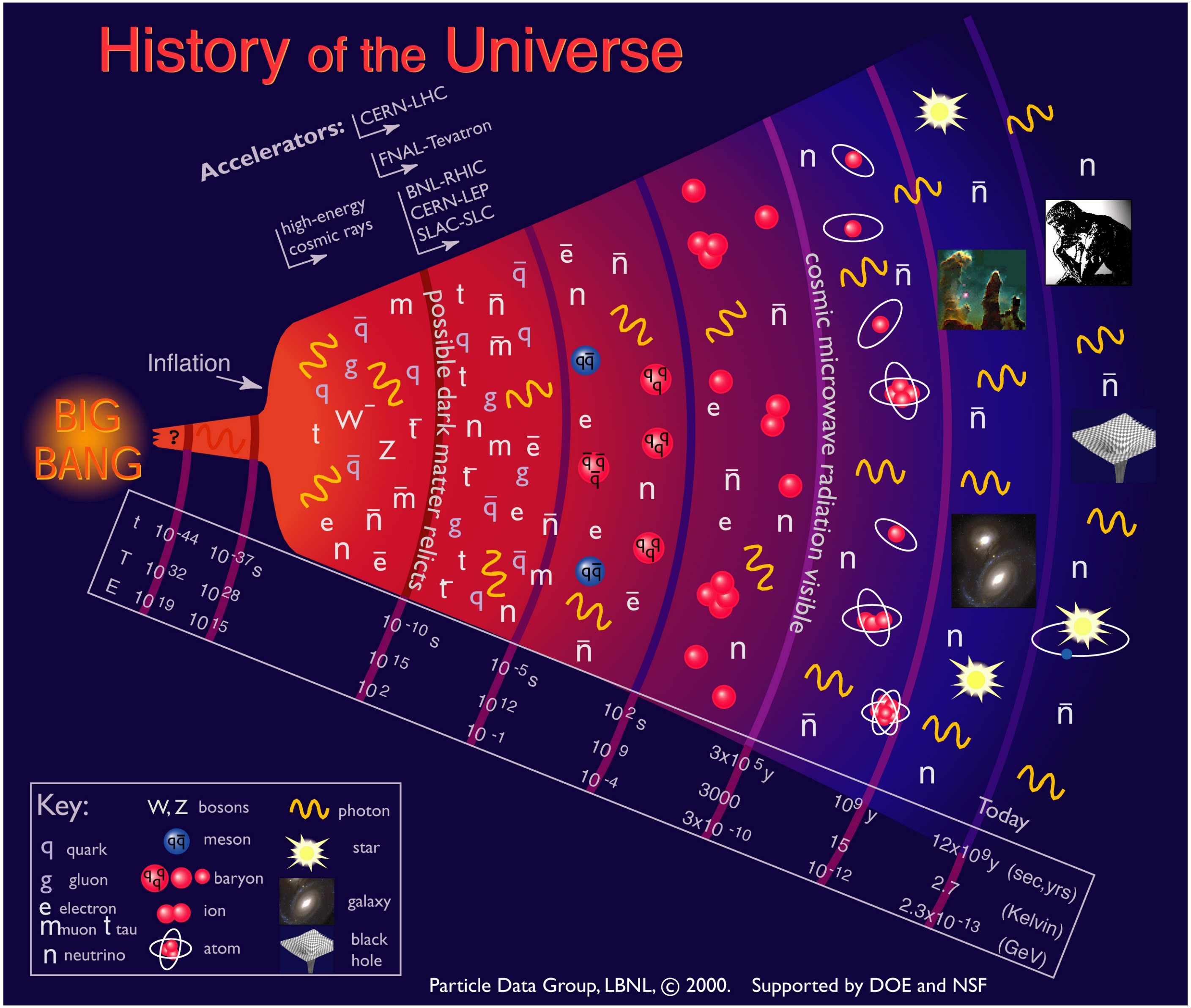




Neutrinos have mass!

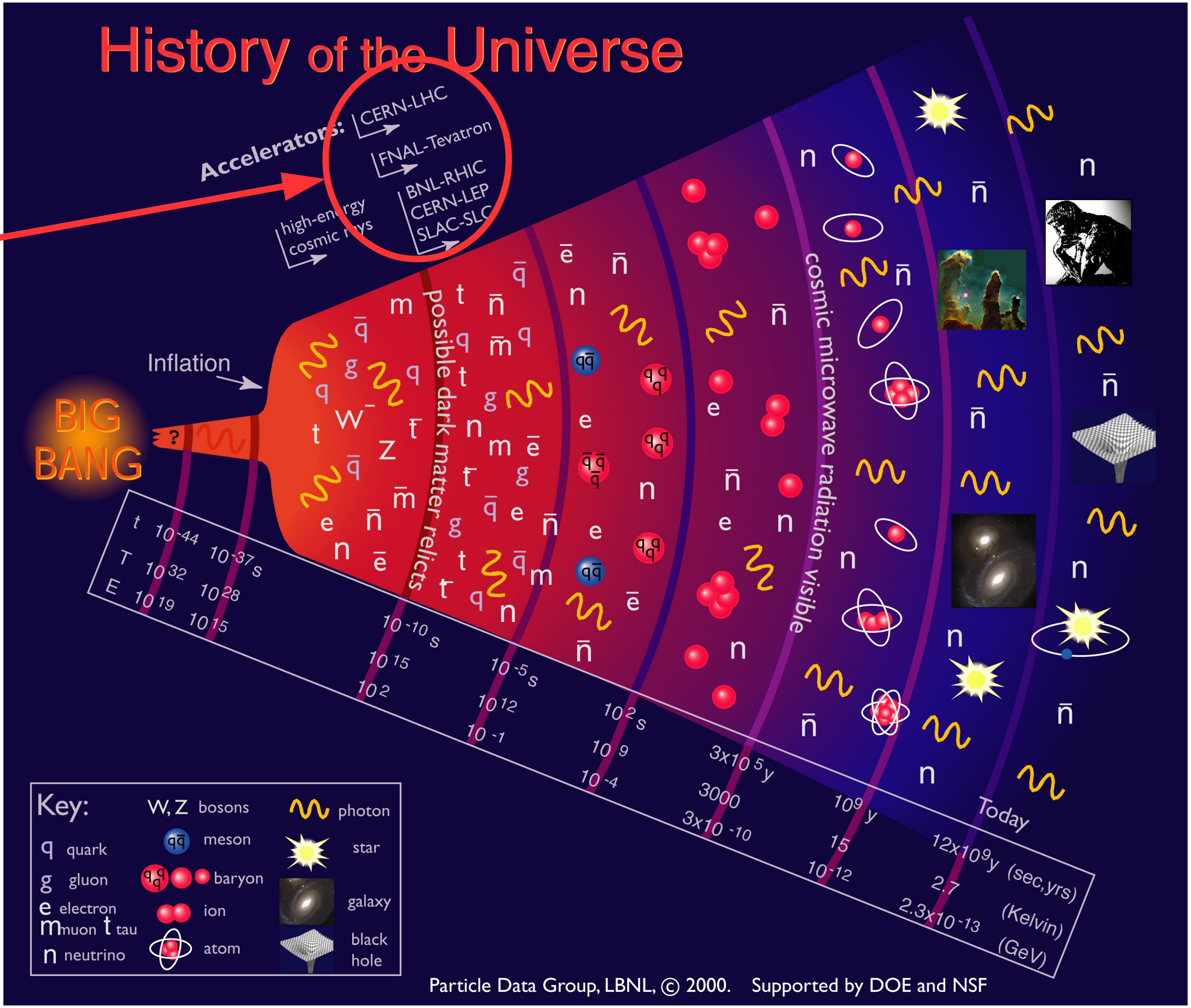


The physics of the smallest scales are intimately connected to the largest scales in the universe

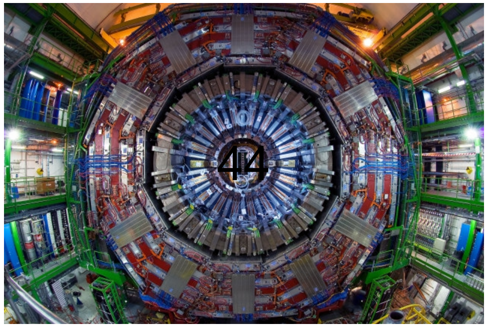


The physics of the smallest scales are intimately connected to the largest scales in the universe

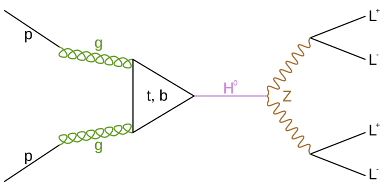
The machines
we work with
are here!!



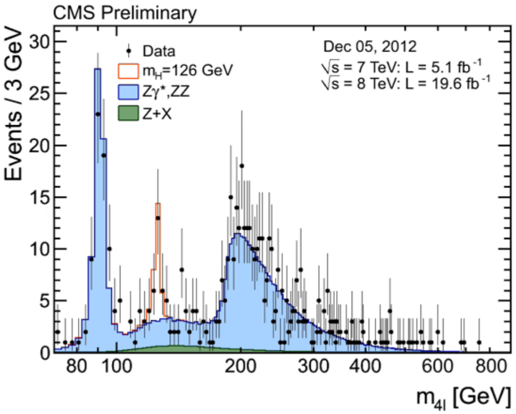
Grad school



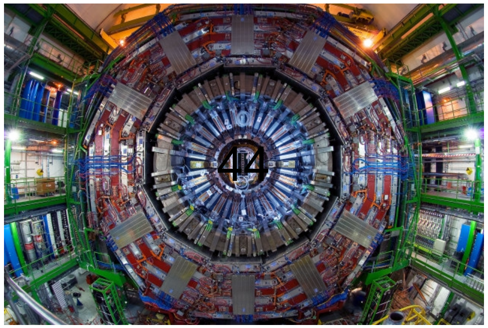
I learned a lot about this...



...and this...



Grad school

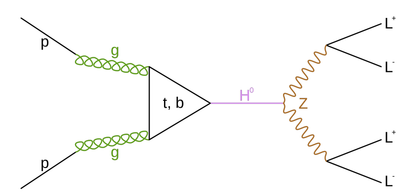


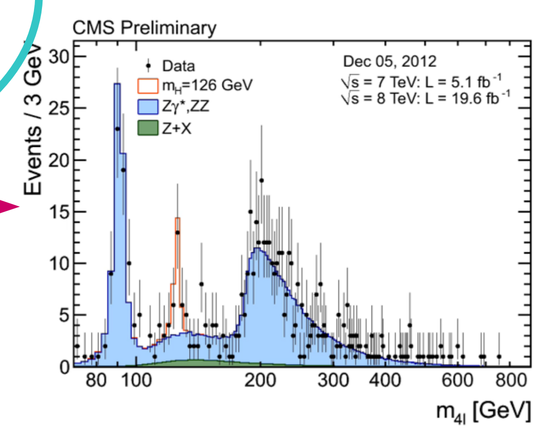
I learned a lot about this...

...but really nothing about all this

Basically what it takes to get from here to there

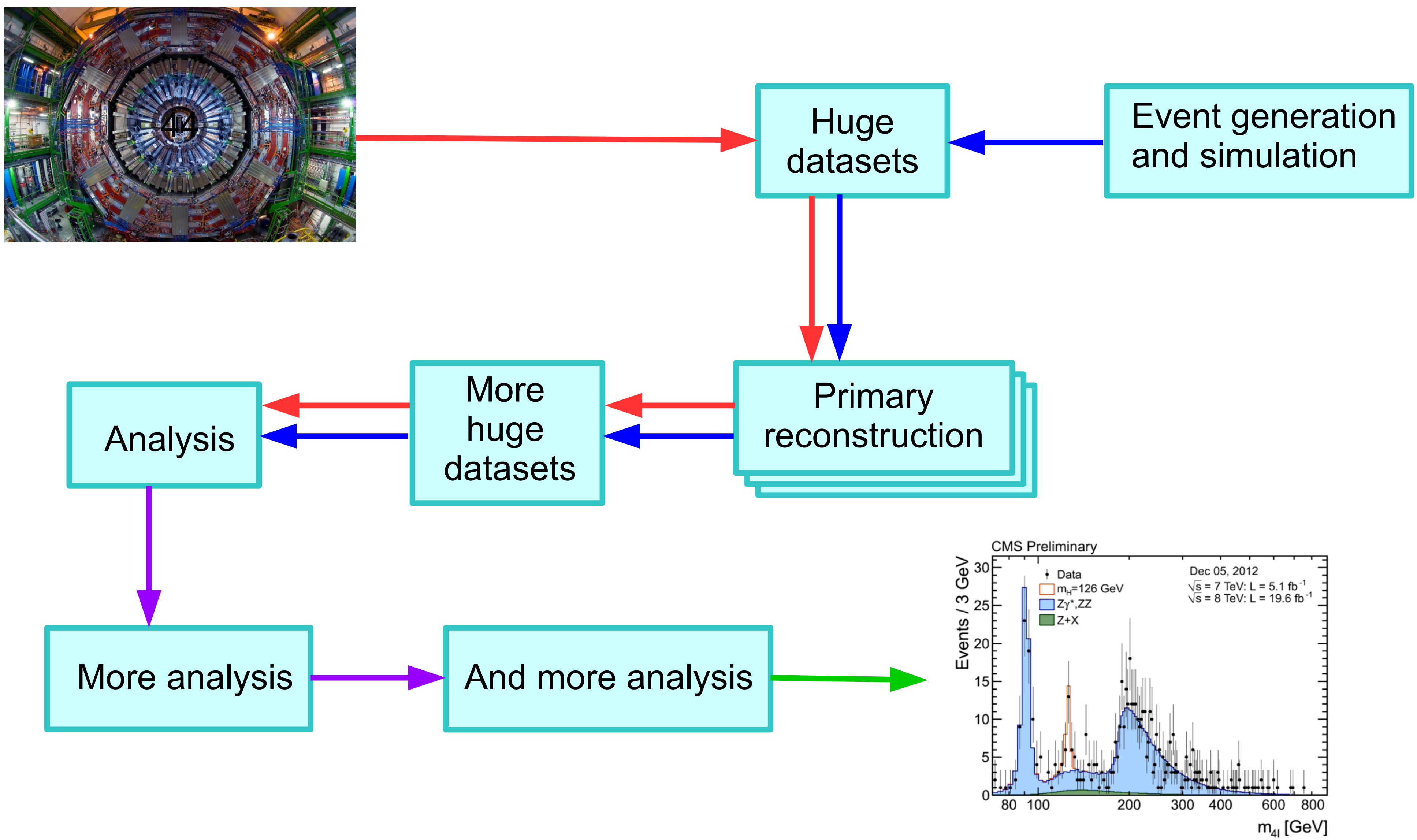
This is scientific computing!!





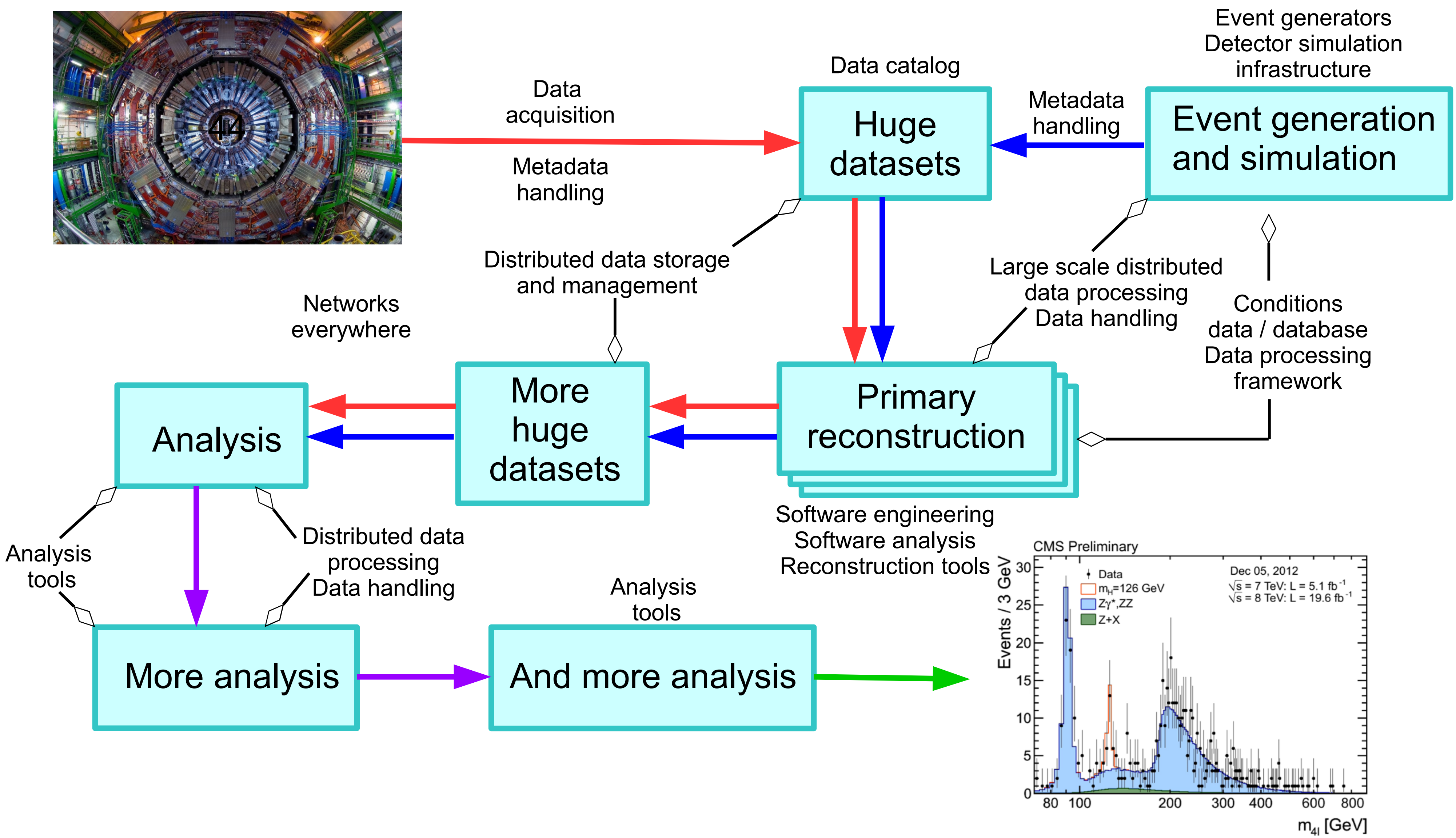
...and this...

A typical workflow



A typical workflow

with some computing service areas



Experiments supported by Fermilab computing

- **Collider experiments**
  - **CMS**
  - CDF
  - D0
- **Flavor experiments**
  - MIPP
  - **Mu2e**
  - **Muon g-2**
  - SeaQuest
- **Neutrino experiments**
  - ArgoNeut
  - LAr1-ND
  - LArIAT
  - DUNE
  - **MicroBooNE**
  - **MINERvA**
  - MiniBoone
  - MINOS
  - **NOvA**
  - NUMI
  - SNO+
- **Astroparticle/ Cosmology Experiments**
  - CDMS
  - COUPP
  - DAMIC
  - **Dark Energy Survey**
  - DarkSide-50
  - DESI
  - Holometer
  - LSST
  - SDSS

Fermilab's Scientific Computing Division (SCD) attempts to provide **common solutions** and **standard interfaces** to computing resources, and a **toolkit of of services and applications** that span their computing needs, integrating everything into a seamless model.

My goal today

- Touch on some major areas of scientific computing at Fermilab
- To give a flavor for some of the challenges and trade-offs

Caveat:

- Cannot cover everything (not even close!)
- Will have bias towards HEP, LHC, LAr neutrino experiments
  - The major drivers
  - That is my background

Outline

- What do we mean by "scientific computing"?
- Data storage and management
- Software: extracting the physics from the data
- Data processing
- New directions in software
- Summary

A common theme: the challenge of scale

The principal challenge in much of scientific computing

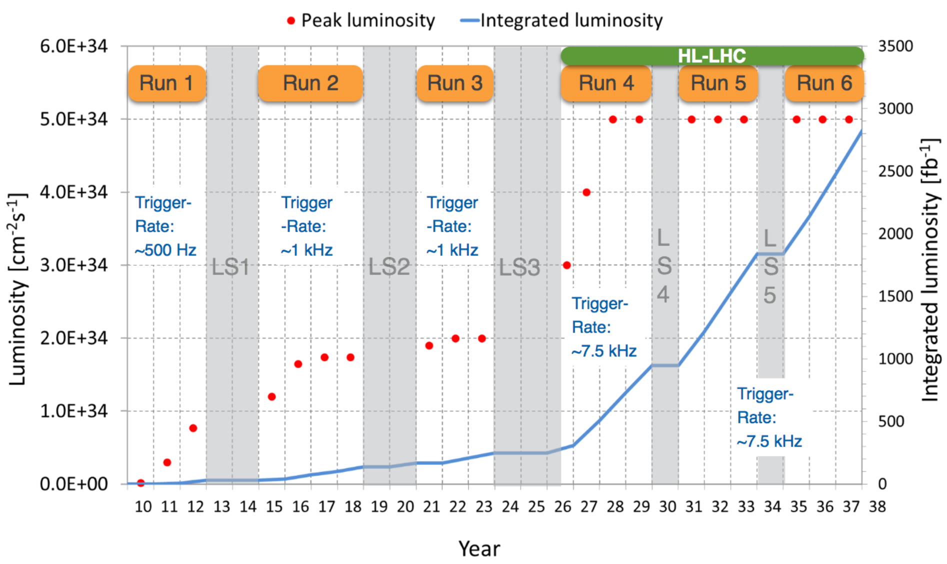
- Data volume
  - Already dealing with datasets of 100 PB (PetaBytes – will come back to this...)
    - LHC experiments now generate 20—30 PB of data per year
    - This will increase in the future
- Complexity of the calculation required
  - Difficult pattern recognition problems in interpreting data from detectors
  - Some theoretical calculations / simulations
    - Lattice QCD
    - Cosmological simulations

Require different computing strategies, trade-offs

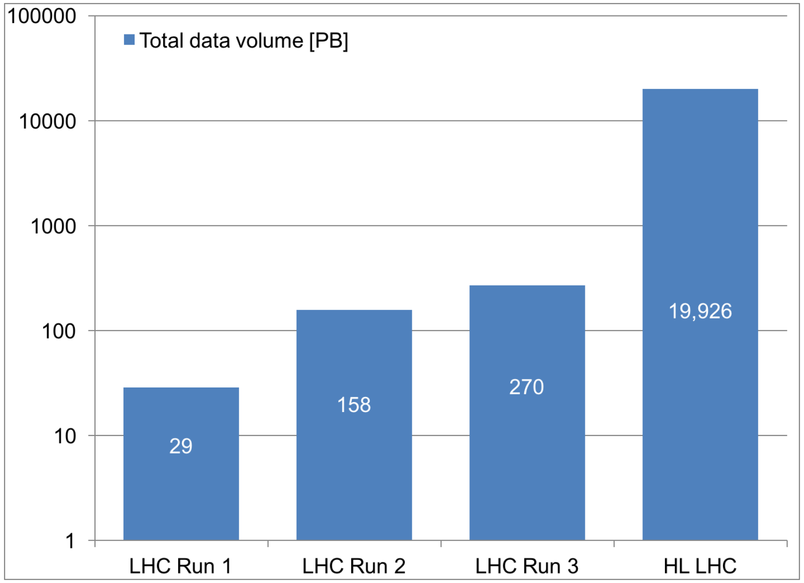
Data storage and management

Data storage and management

- “Big data” – think CMS (and other LHC experiments)
  - Scale is 100's of petabytes (PB) in the near term
  - 10,000's of PB (exabytes) in the long term

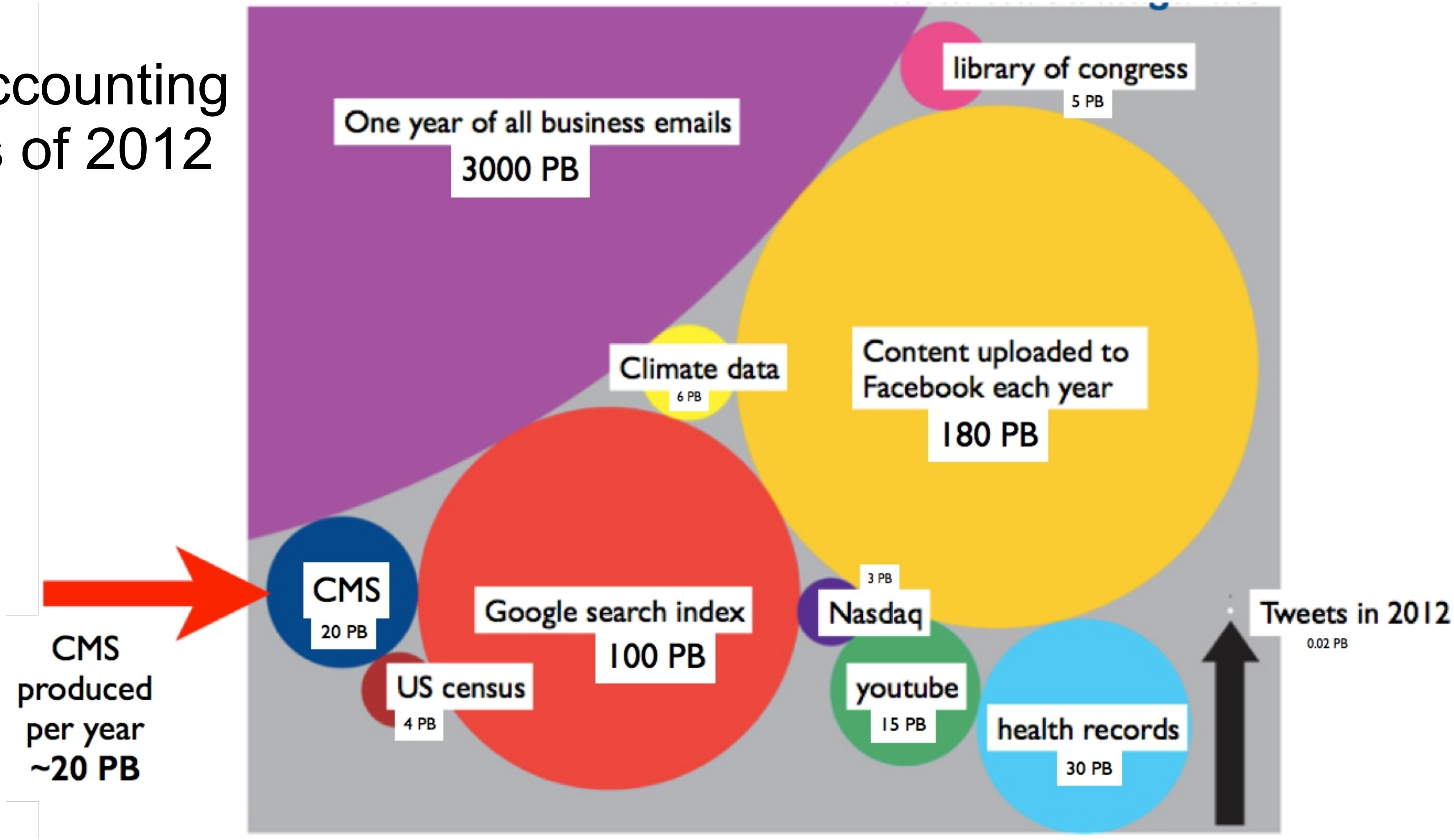


CMS data volume by LHC run



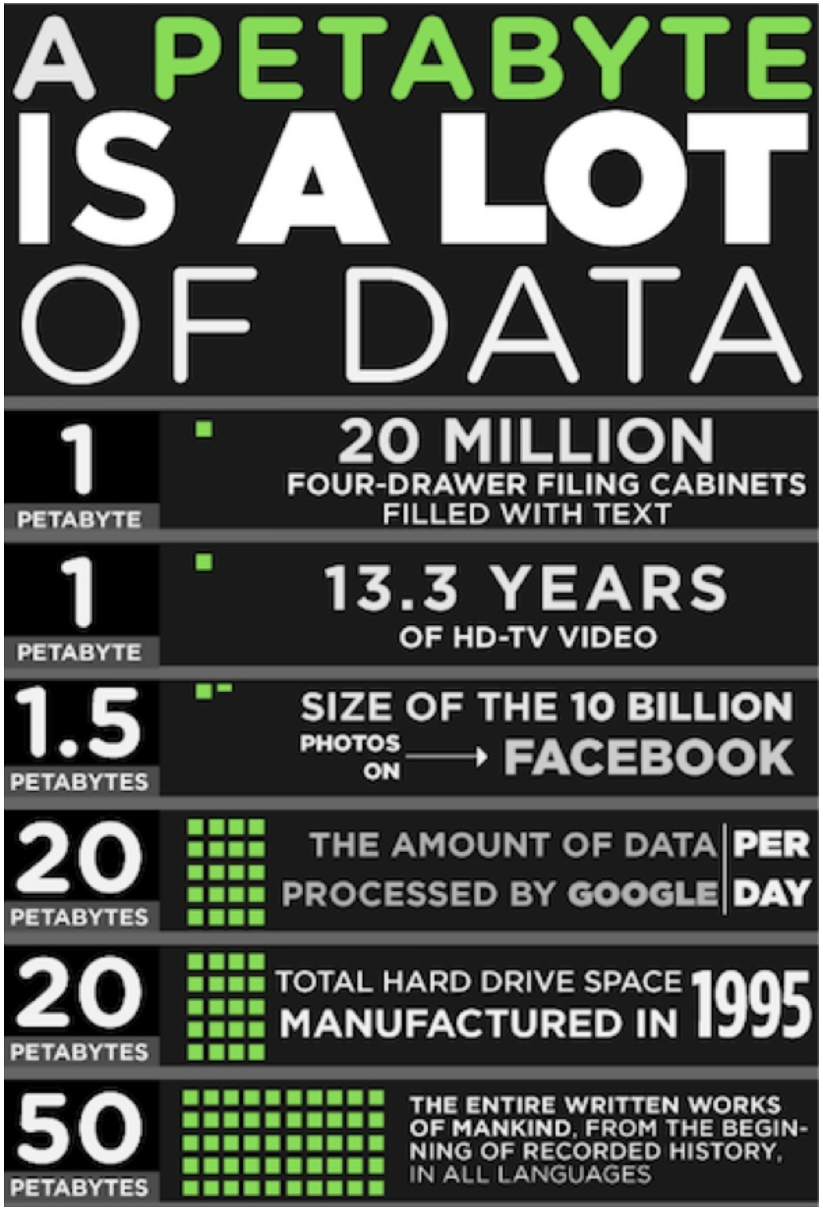
Not the only big-data kids on the block...

Accounting as of 2012



Adapted from Wired: http://www.wired.com/magazine/2013/04/bigdata/

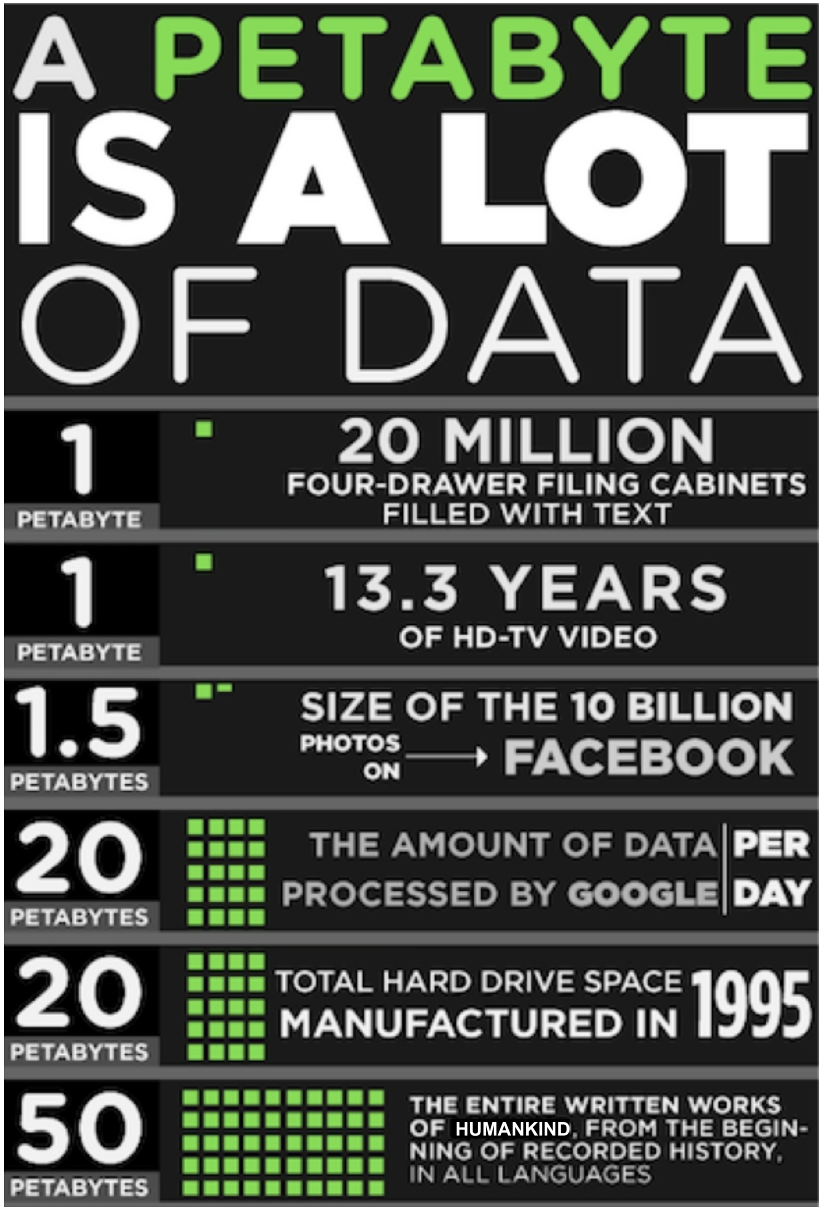
How big is a petabyte??



PB = 10^{15} bytes

10^{15} AA batteries end-to-end is about 3 light-years, so most of the way to nearest star (4.2 ly)

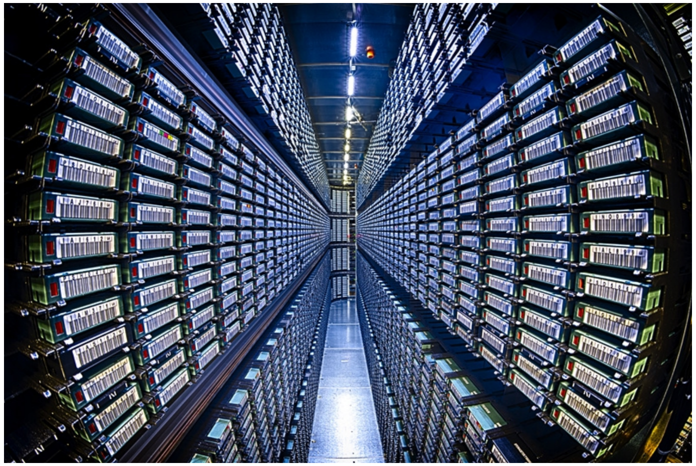
How big is a petabyte??



Binging all 6 seasons of Game of Thrones ~2000 times

How is it stored?

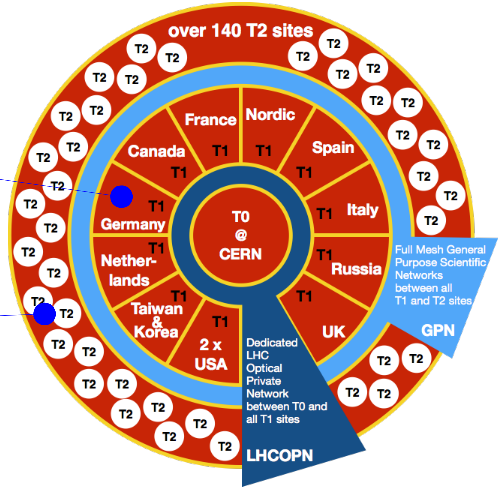
- Tape robots
  - Over 100 PB stored in 7 robots
  - Up to 10k tapes per robot
  - About 320 PB total capacity with current tapes
  - Inexpensive, but high latency access



- Disk
  - About 30 PB of capacity managed by mass storage system
    - “dCache”, “pnfs” all refer to this system
    - Software is a joint project of Fermilab / DESY (accelerator lab in Germany)
  - Most serves as front-end to tape system
    - Files transparently moved between dCache and tape
  - dCache replicates files across servers based on demand
    - There spreads the load for serving “popular” files

Data placement strategy

- Large data stores typically only at a few locations
  - For CMS: CERN, Fermilab, other “Tier 1” facilities
- Processing, selected datasets widely distributed
  - For CMS: the 50+ “Tier 2” facilities
- Jobs routed to where the relevant data is stored
- The CMS example
  - Initially managed placement manually, which requires
    - Matching dataset popularity with site capacity
    - Reacting to changes in popularity
  - Does not scale well
    - Periods when number of file replicas mis-matched to demand



The worldwide
LHC Grid (WLCG)

Data placement strategy

- CMS “dynamic data management”
  - Track dataset popularity based on requests from jobs
  - Manage file replication and placement based on the results

  So data placement becomes automated
  - Data movement more responsive to demand, but still not “on demand”

- Another approach: data federations

Data federations

Definition

"A collection of disparate storage resources that are transparently accessible across a wide area via a common namespace*"
K. Bloom for CMS Collab, J. Phys. Conf. Ser. 513 (2014) 042005

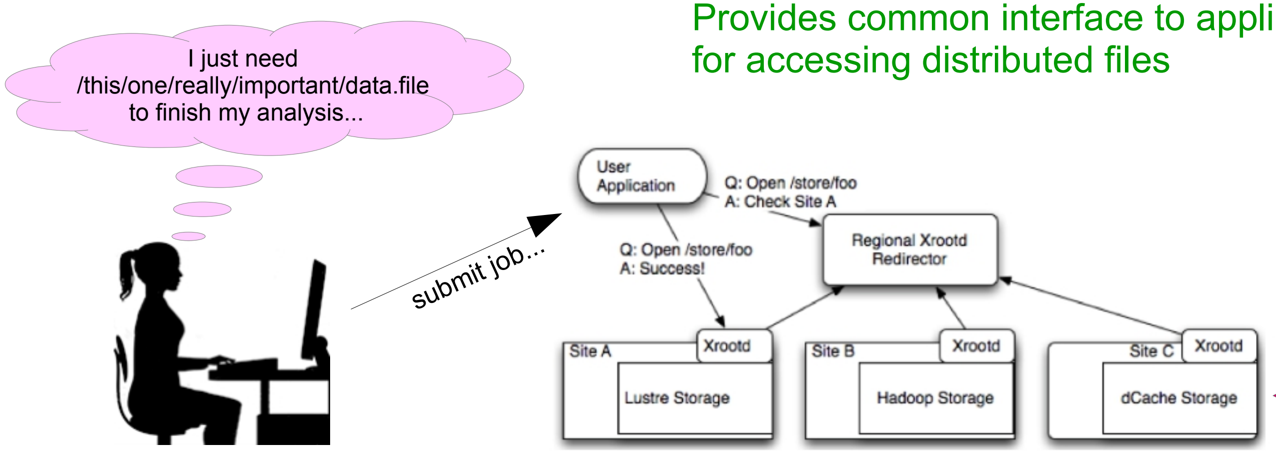
* "Common namespace": the path to files is independent of physical location
For example, might look like /pnfs/uboone/data/.../my_dataset_file1.root
This is what dCache does

- Allows a strategy of on-demand data access from any site to any site

Data federations

- Many experiments implement using xrootd
  http://xrootd.org/

Provides common interface to applications for accessing distributed files

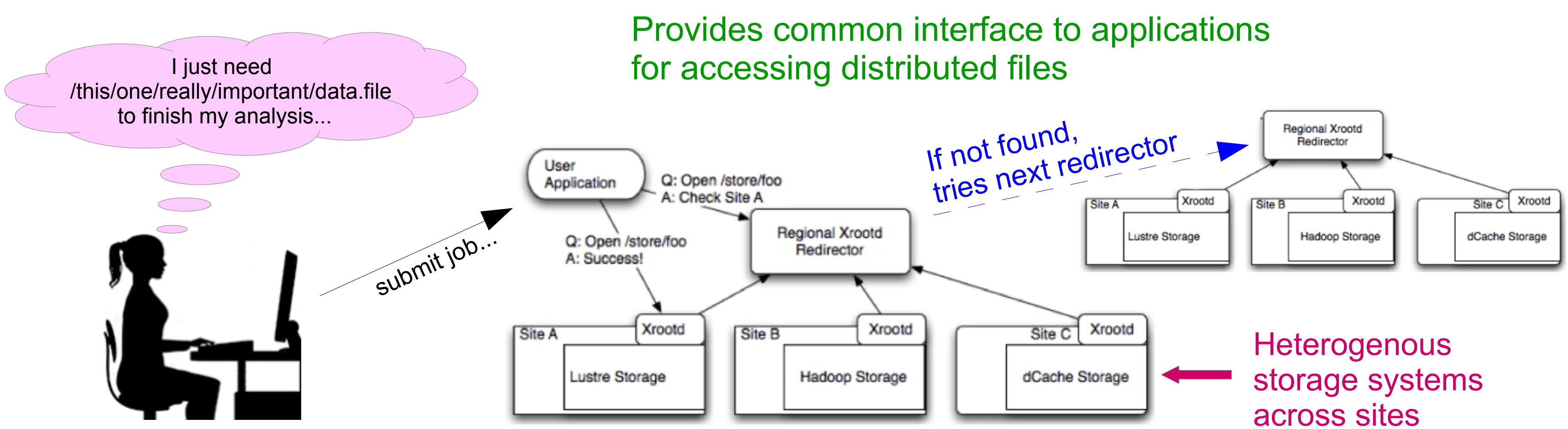


Heterogenous storage systems across sites

- A hierarchy of re-directors provides efficient location service
- Once found, **data is streamed to the job** over the network
  - Some performance cost, but no local storage needed

Data federations

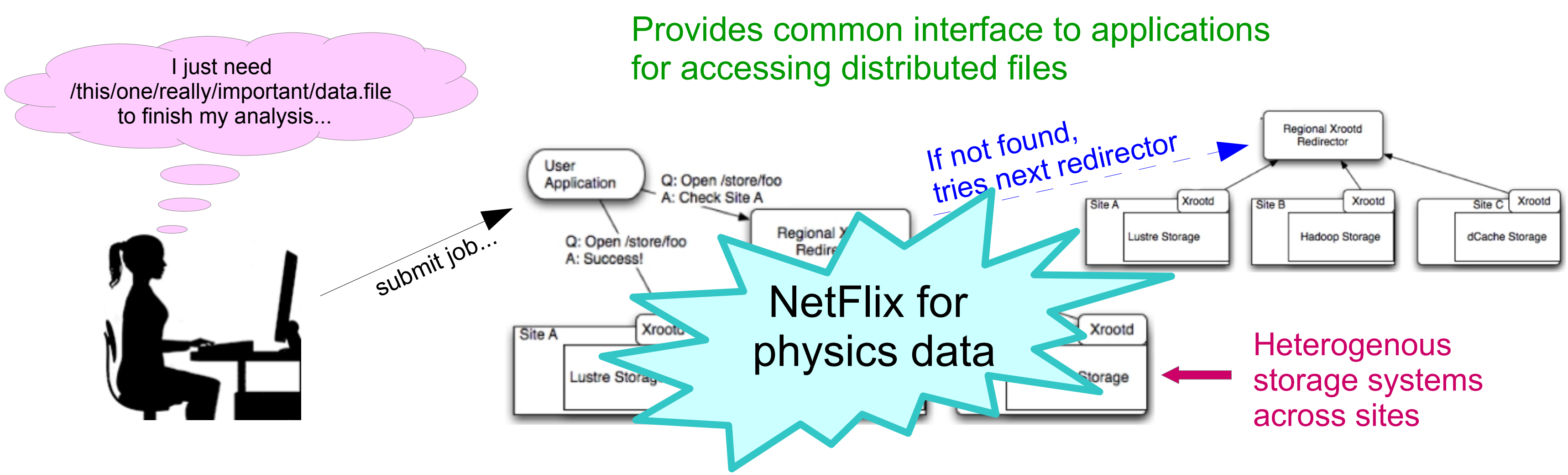
- Many experiments implement using xrootd
  http://xrootd.org/



  - A hierarchy of re-directors provides efficient location service
  - Once found, **data is streamed to the job** over the network
    - Some performance cost, but no local storage needed

Data federations

- Many experiments implement using xrootd
  http://xrootd.org/



  - A hierarchy of re-directors provides efficient location service
  - Once found, **data is streamed to the job** over the network
    - Some performance cost, but no local storage needed

Data placement for everyone!

- StashCache
  - Supported by OSG (Open Science Grid)*
  - “Opportunistic” storage for OSG users
    - OSG collaborators provide local disk
    - Other OSG users can unused space via StashCache
  - Based on xrootd
    - OSG runs xrootd cache servers
    - Dynamically populate caches
      - Very efficient distributed data access

Stash origin: ☆

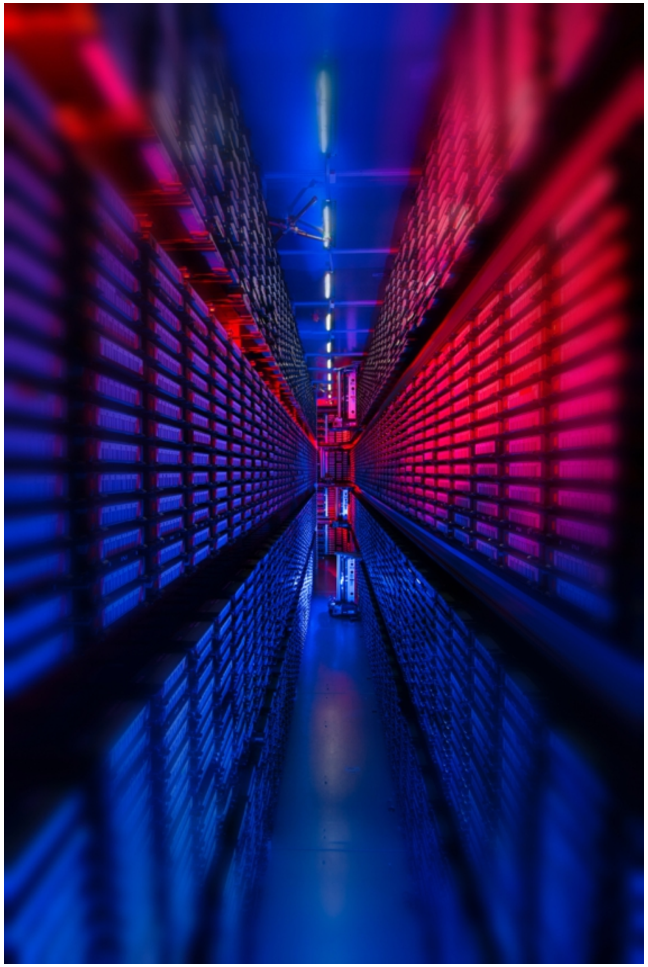
OSG Caches: ●

An **enabling technology** for users who don't have / can't support the infrastructures like LHC experiments

* Will discuss more about OSG a bit later...

Active Archival Facility

- Lending HEP data management expertise
  - Working with broader science community to enable everyone to manage, distribute, access their data globally
- Fermilab's Active Archival Facility (AAF)
  - Provide services to other science activities to preserve integrity and availability of important and irreplaceable scientific data
  - Projects
    - Genomic research community
      - Archiving datasets at Fermilab's AAF
      - Providing access through Fermilab services to ~300 researchers worldwide
    - University of Nebraska and University of Wisconsin are setting up archival efforts with Fermilab's AAF



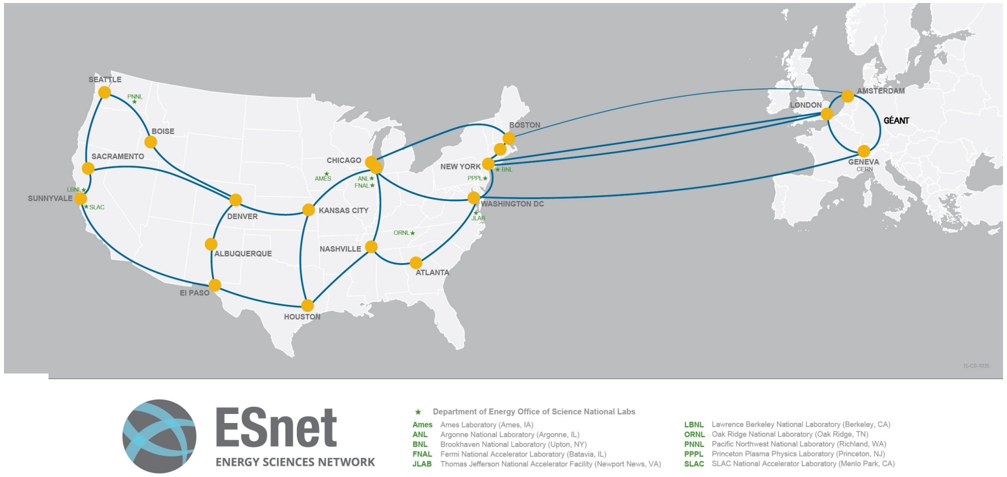
Data movement

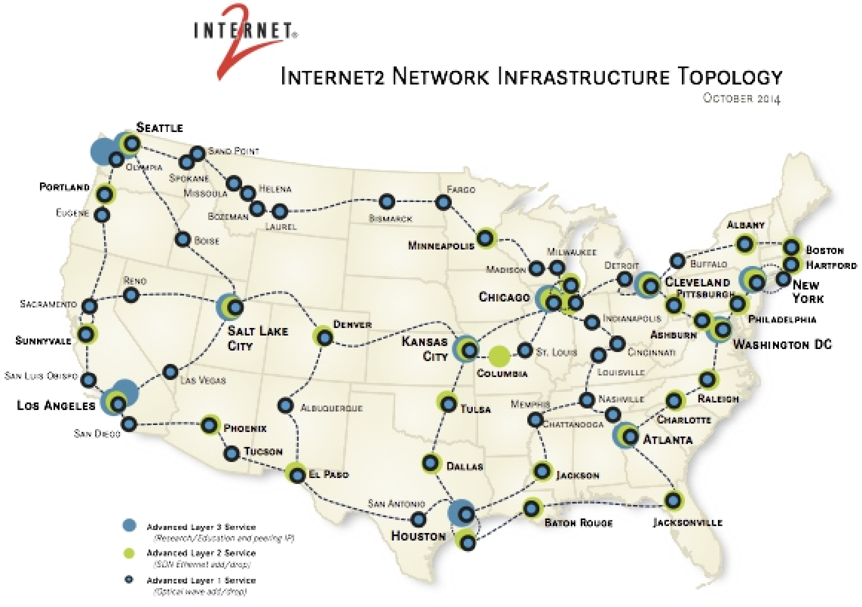
- Supporting all of this requires very fast, highly reliable networks

ESnet

(DOE Energy Sciences Network)

100 Gbps optical fiber backbone





Existing infrastructure proven to meet the demanding requirements enabling these data management strategies

Would have been unthinkable only a few years ago

Software: extracting the physics from the data

First, some basic software tools

- Event processing and analysis frameworks
  - Underlying infrastructure, the core of the software by providing
    - I/O handling, integration with lab data handling systems
    - Event loop
    - Configuration
    - Metadata handling / generation
    - Provenance tracking
    - Dynamic library loading
    - etc., etc....
- Fermilab supports two for the community
  - cmssw for CMS experiment
  - *art* for neutrino and muon experiments
    - Shared development and support across experiments
    - Allows experiments to focus on physics

First, some basic software tools

- LArSoft
  - "An integrated, *art*-based, experiment independent set of software tools used by multiple experiments to perform simulation, reconstruction and analysis of LAr TPC data" *– LArSoft project http://larsoft.org/*
  - Used by all LAr-based experiments at Fermilab
    - Each experiment contributes algorithm code
    - Algorithms shared among all experiments
  - Shared code base
    - Lowers development cost for larger experiments
    - An enabling technology for smaller experiments
  - Fermilab SCD scientists, computing professionals involved at every level
    - Software engineering and development
    - Algorithm development and testing
    - Project support

Event reconstruction

Definition

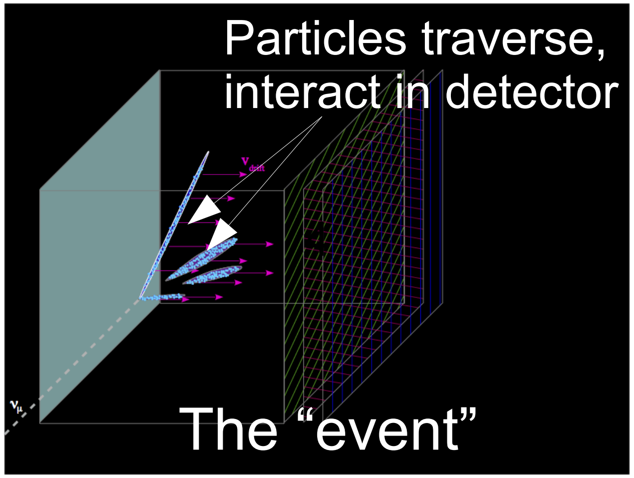
The process of identifying and aggregating the signals left by individual particles or groups of particles that traversed (or interacted in) the detector, measuring their paths, kinematic properties, and production points, and identifying their particles species (when possible)

- This is a **HUGE** topic

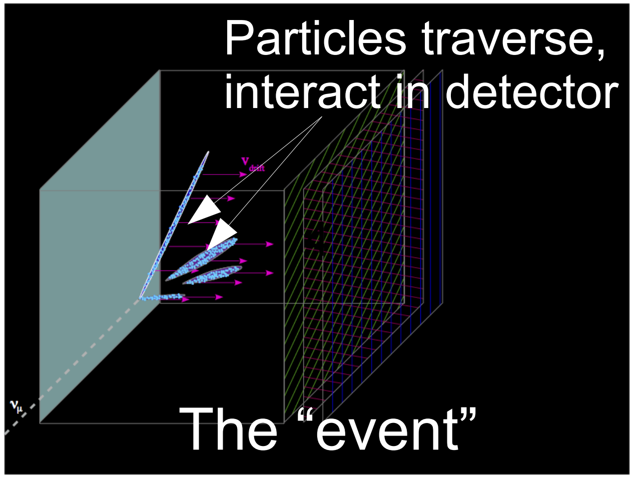
  ...and very interesting

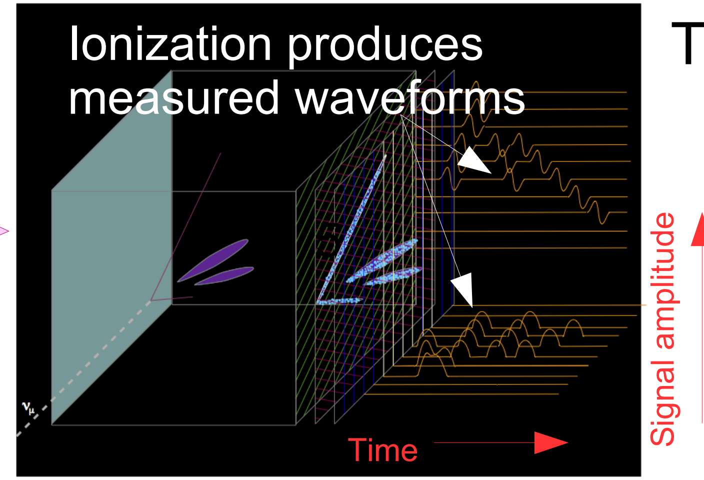
- Involves many steps
- For many experiments, it defines "the computing problem"
  - The issue: "data production"
    - The process of reconstructing **all** of the raw data to produce data ready to be analyzed for physics content
    - Typically repeated throughout the experiment as algorithms improve

Consider the example of a liquid argon (LAr) Time Projection Chamber



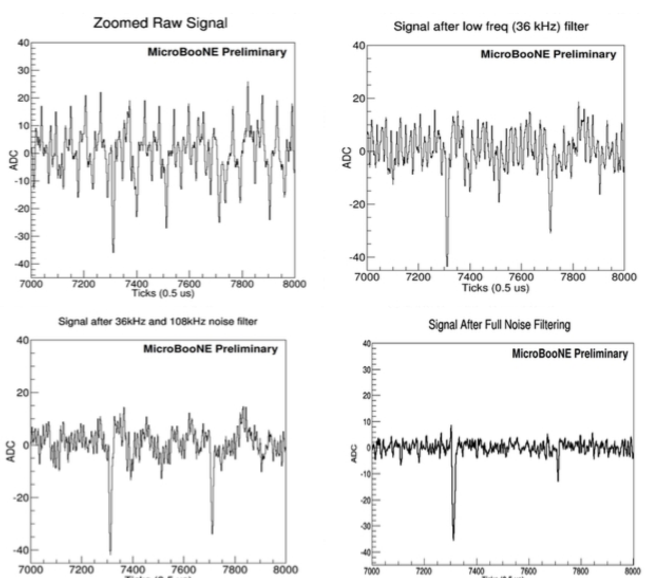
Consider the example of a liquid argon (LAr) Time Projection Chamber

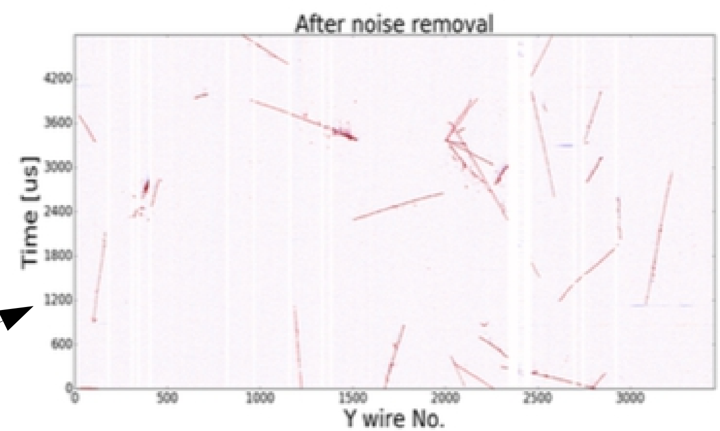




The “raw data”

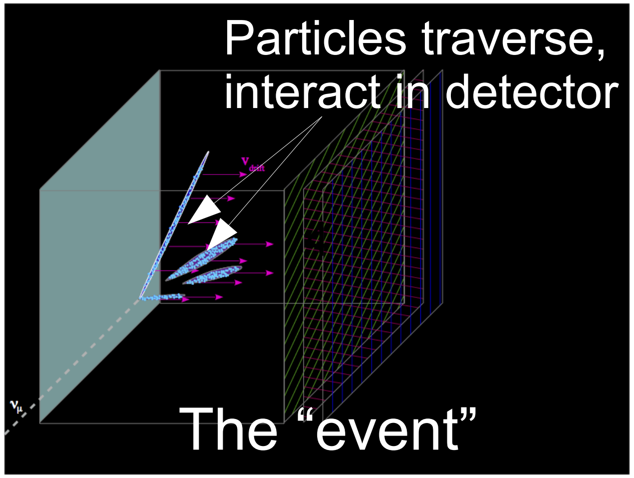
Example from MicroBooNE: raw data waveforms on one wire + signal processing

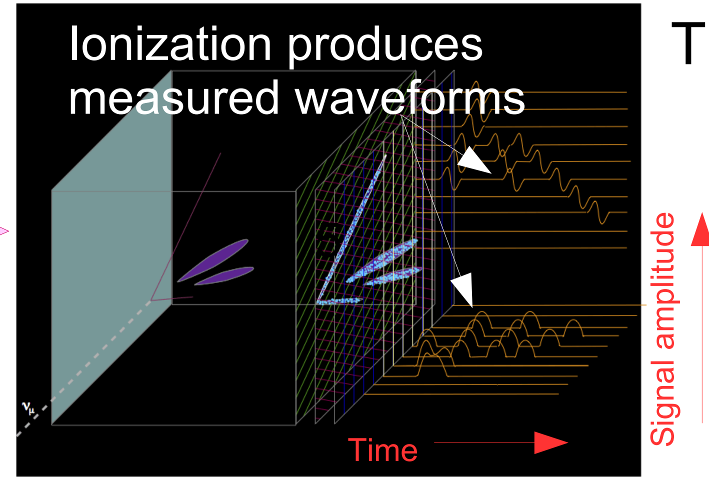




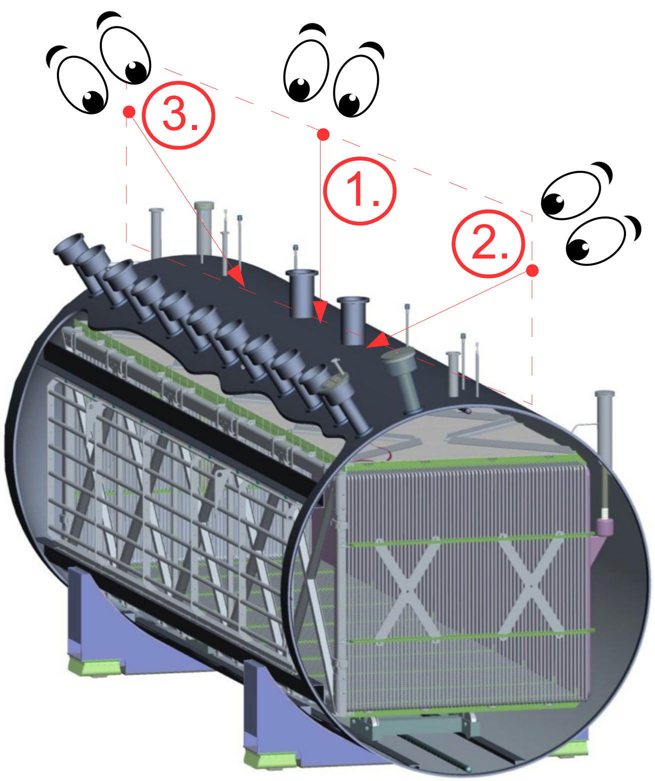
Signals on all wires in a plane create a 2D image of the event

Consider the example of a liquid argon (LAr) Time Projection Chamber

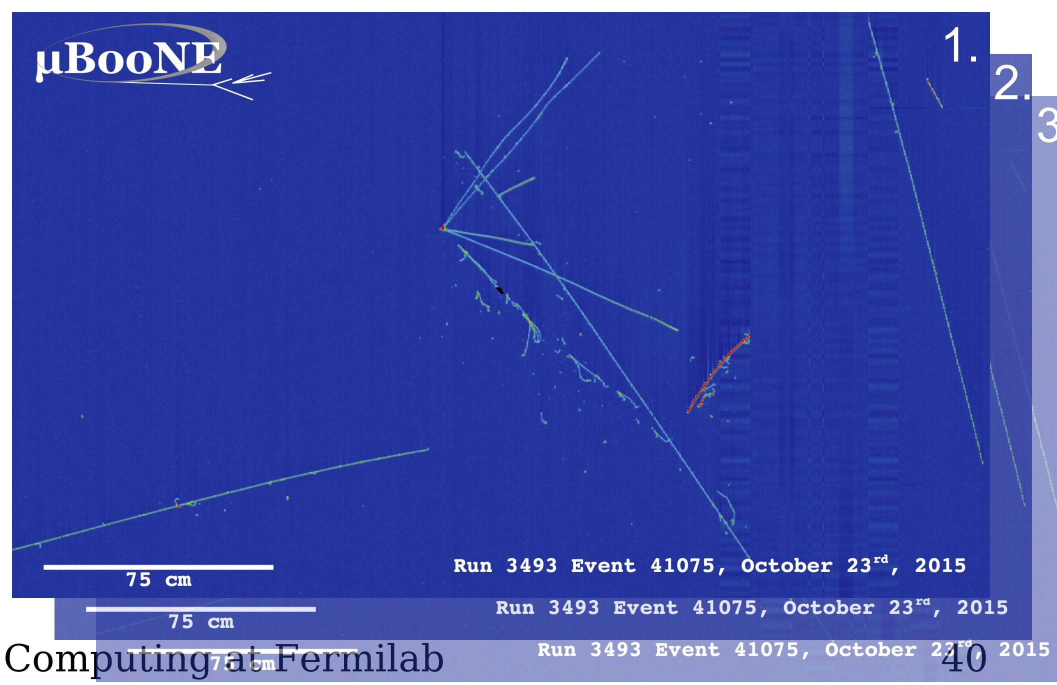




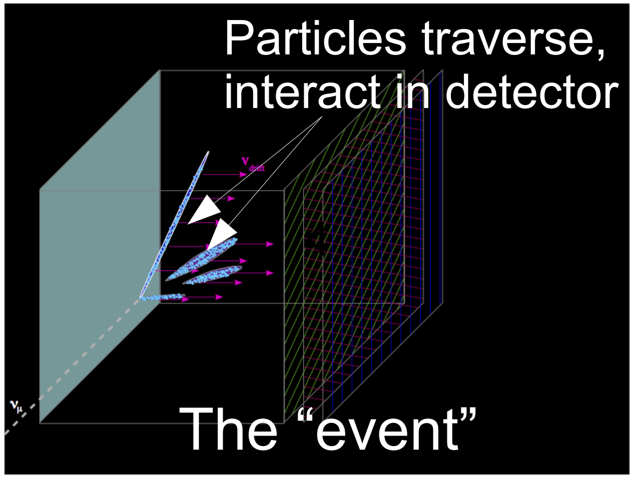
The “raw data”

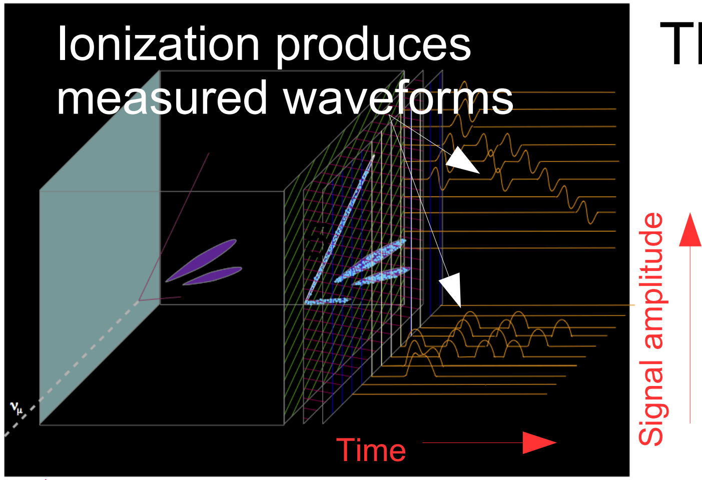


Have three 2D views of the event, so three images



Consider the example of a liquid argon (LAr) Time Projection Chamber





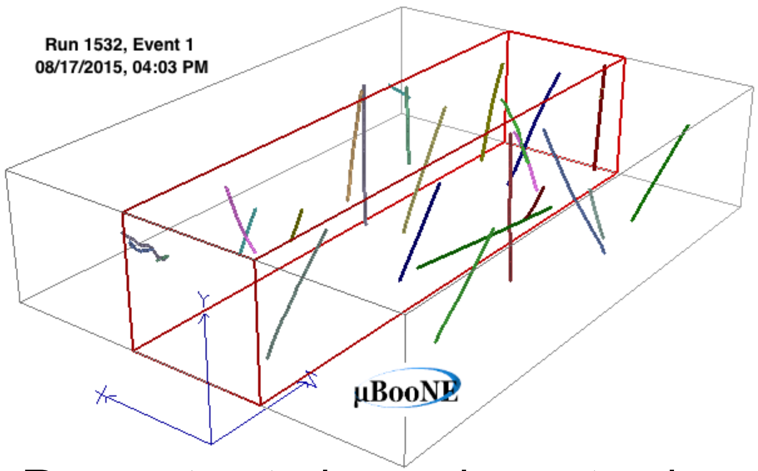
The "raw data"

Reconstruction algorithms

An example workflow

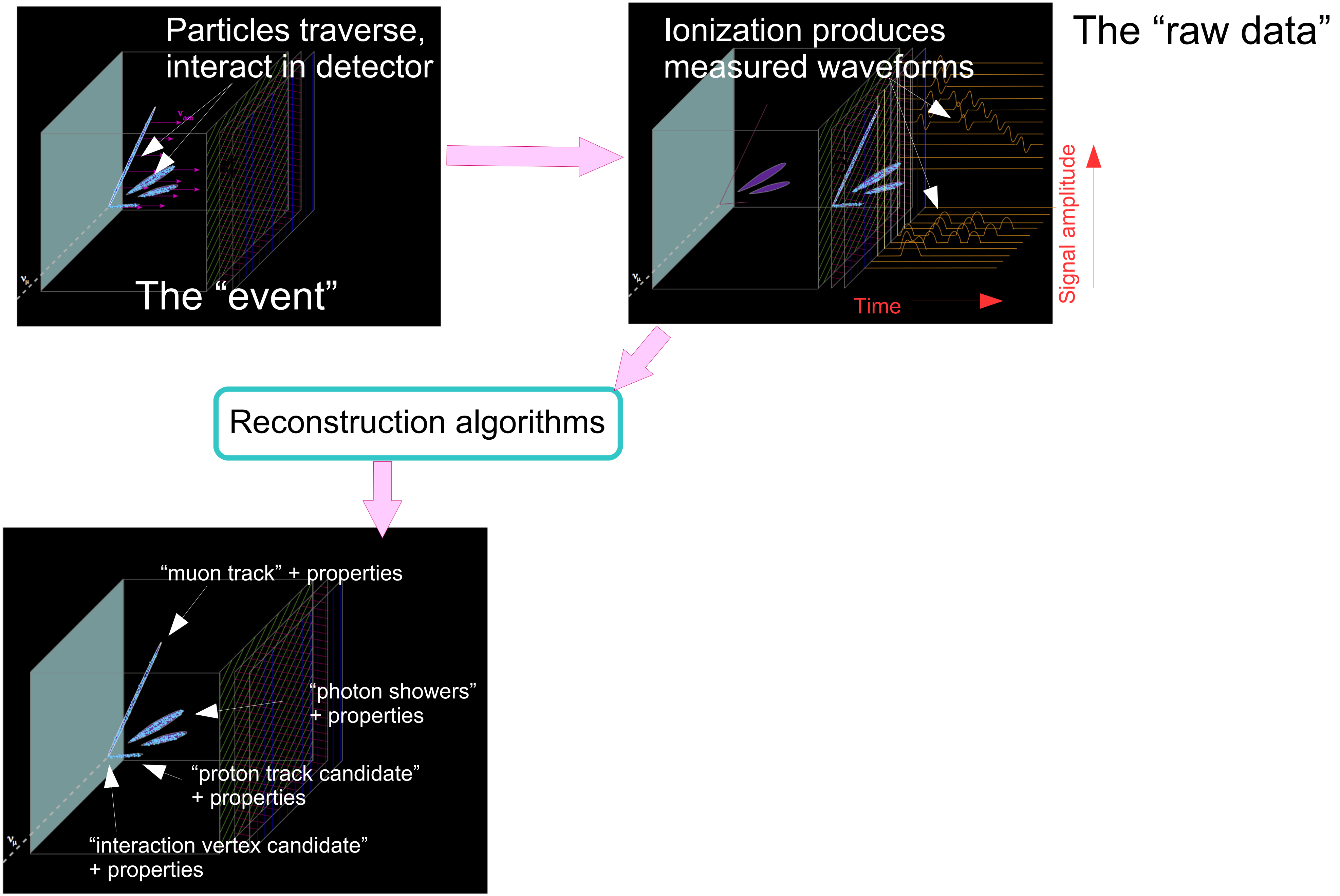
- Find "hits" (peaks) in individual waveforms
- "Cluster" all hits that appear associated with a single particle
- Match clusters between views: 3D objects
- Identify tracks and EM showers
- Find vertices
- Perform particle ID

Usually multiple algorithms at each step



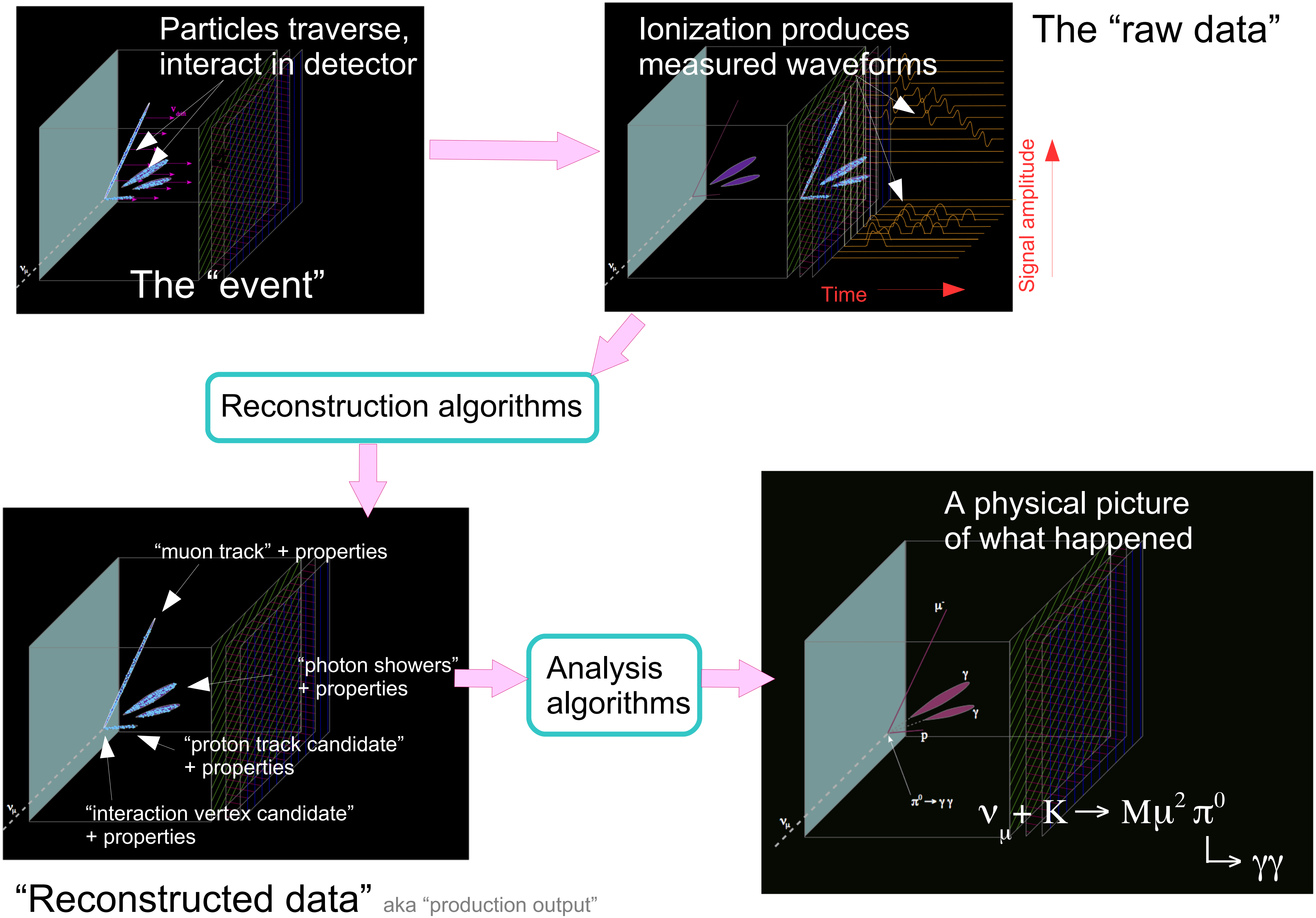
Reconstructed cosmic ray tracks

Consider the example of a liquid argon (LAr) Time Projection Chamber

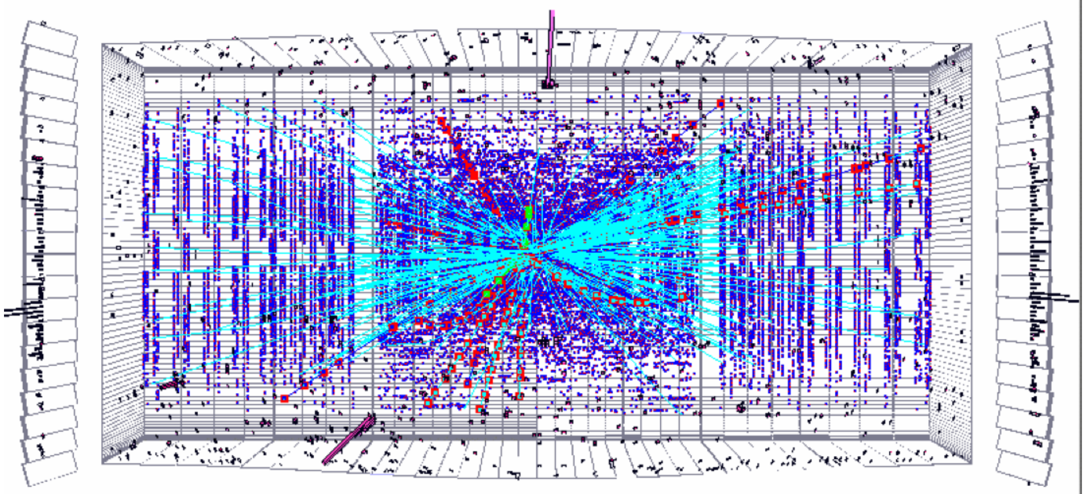


"Reconstructed data" aka "production output"

Consider the example of a liquid argon (LAr) Time Projection Chamber

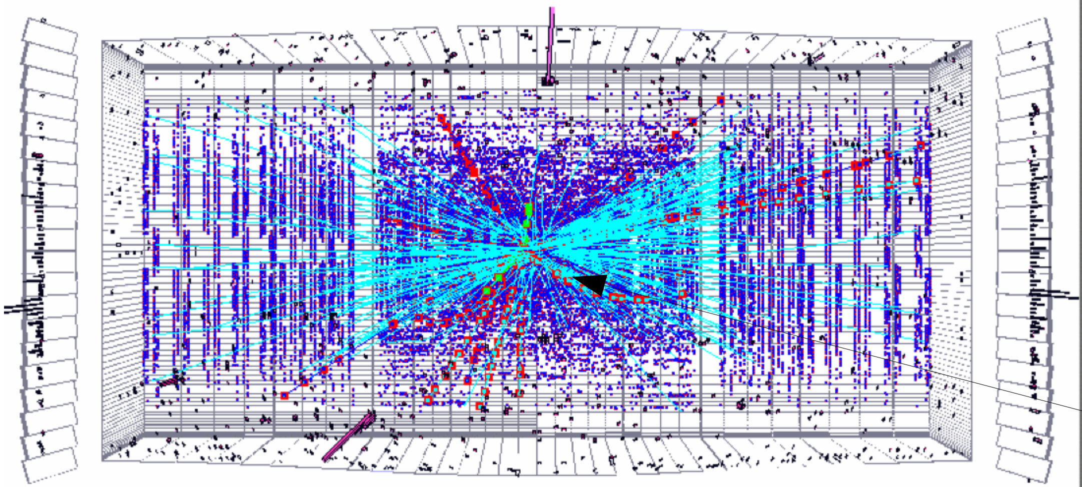


CMS example: similar story, but a bit more complicated events...

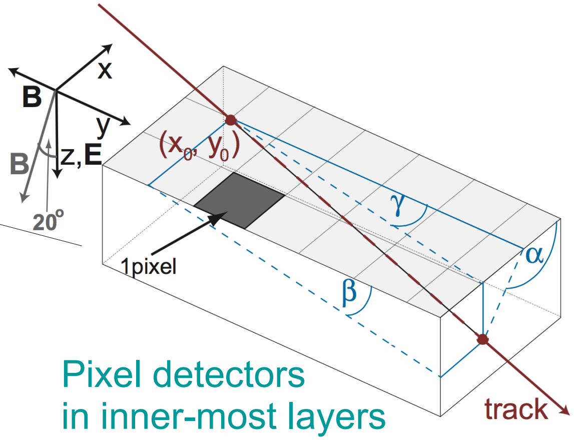


Side view of tracking detector in simulated event

CMS example: similar story, but a bit more complicated events...

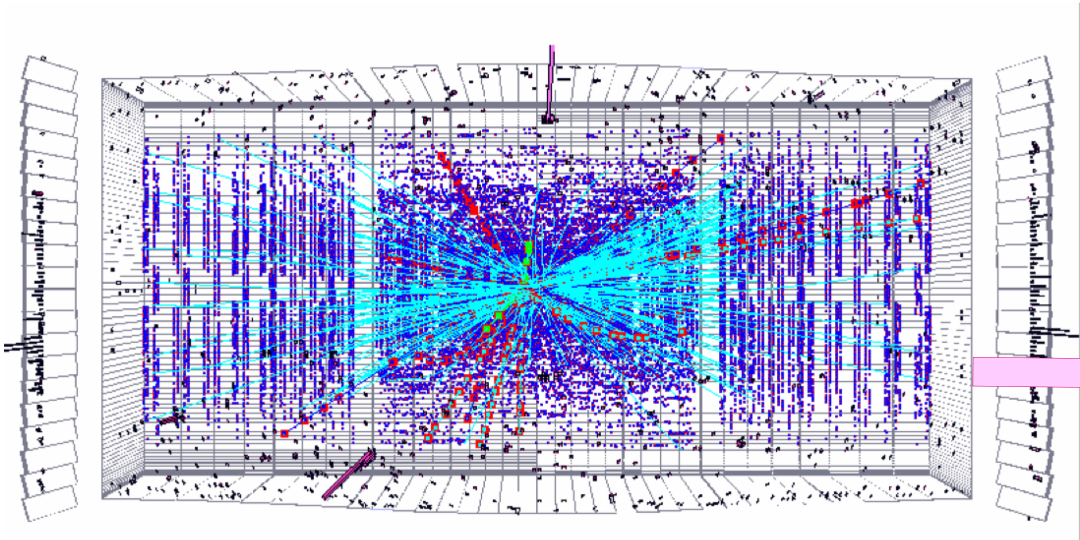


Side view of tracking detector in simulated event

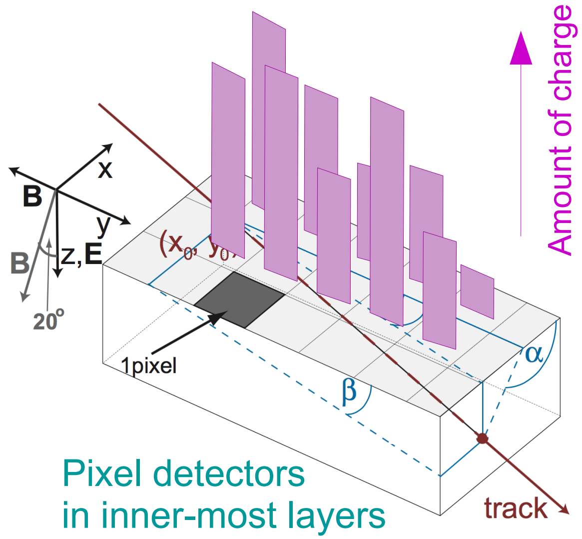


Pixel detectors
in inner-most layers

CMS example: similar story, but a bit more complicated events...



Side view of tracking detector in simulated event



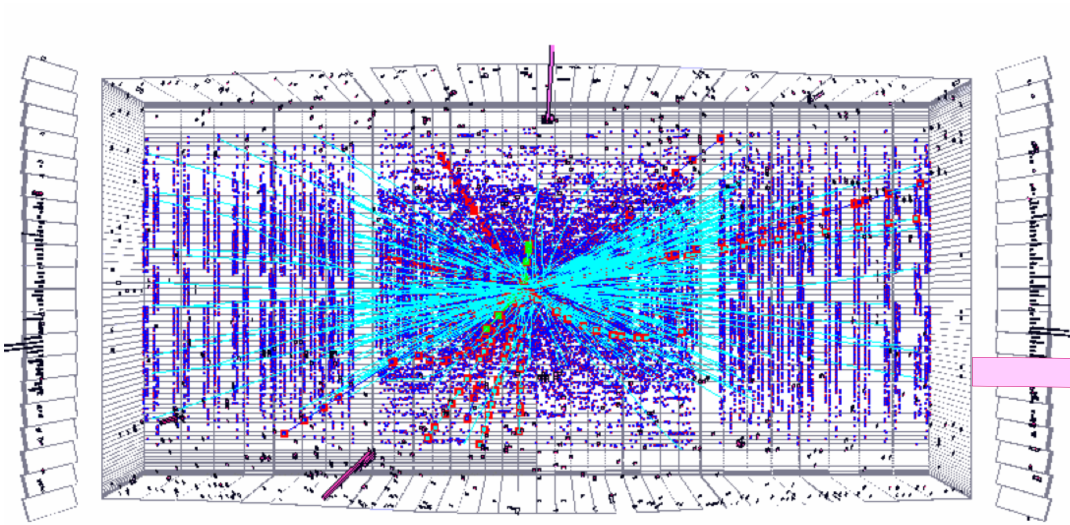
Pixel detectors
in inner-most layers

Raw data for silicon detectors:
charge deposited in each channel

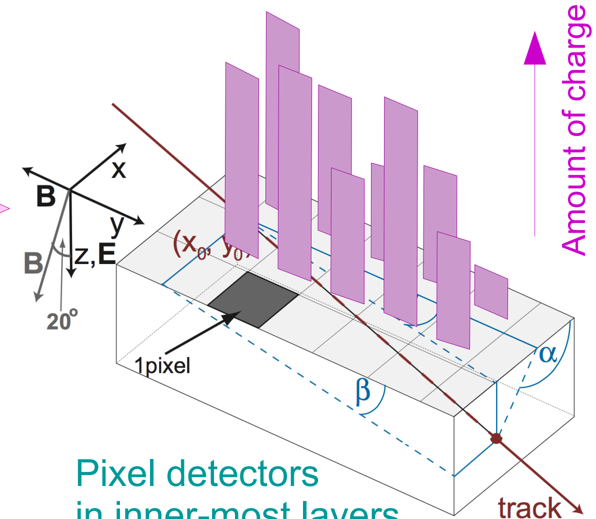
First identify groups of channels associated with the passage of a track

The event display shows these “hits”

CMS example: similar story, but a bit more complicated events...



Side view of tracking detector in simulated event



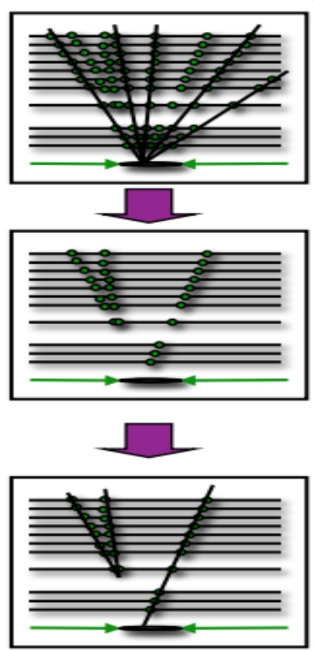
Raw data for silicon detectors: charge deposited in each channel

First identify groups of channels associated with the passage of a track

The event display shows these “hits”

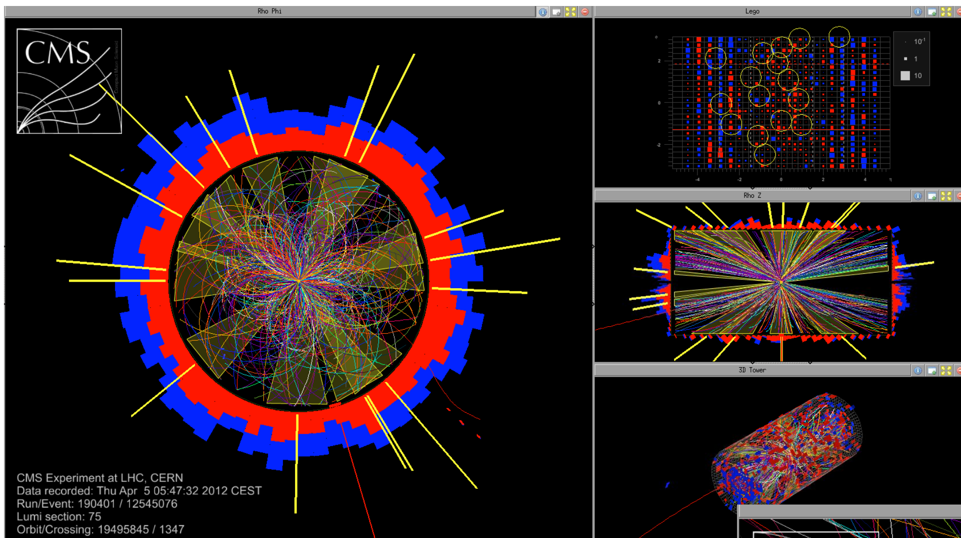
Then reconstruct tracks by connecting hits across layers to form track trajectories

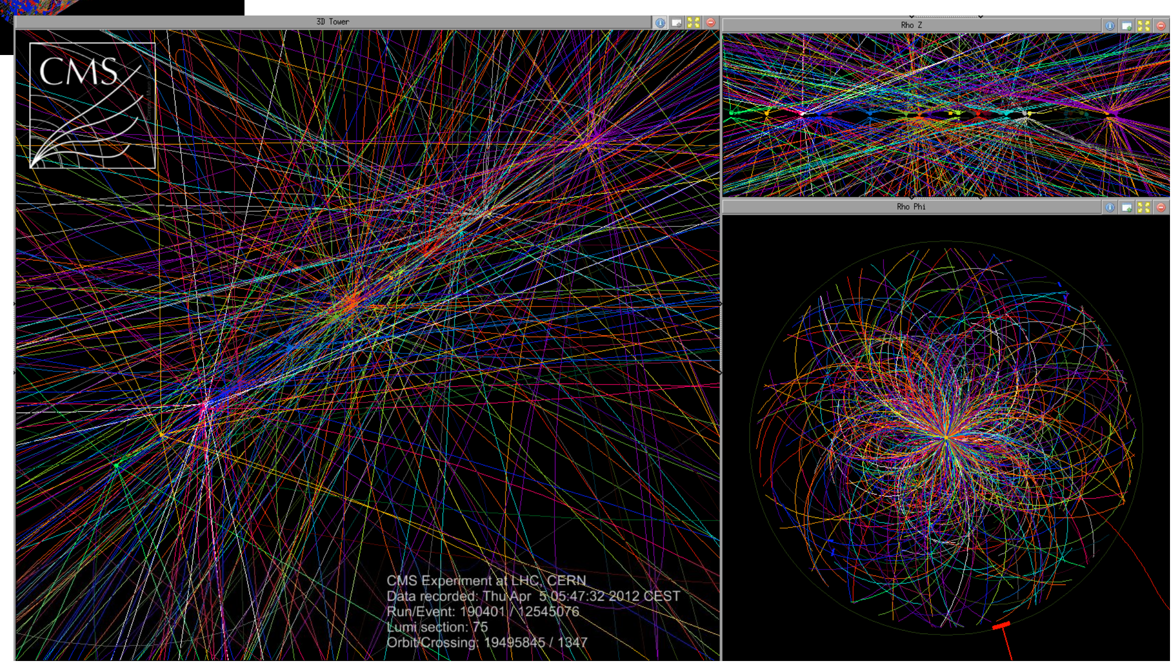
Performed iteratively by finding good tracks, removing hits, looking again



CMS example: similar story, but a bit more complicated events...

Looks intimidating! But can do well:





29 separate pp interactions identified via tracks found in this event

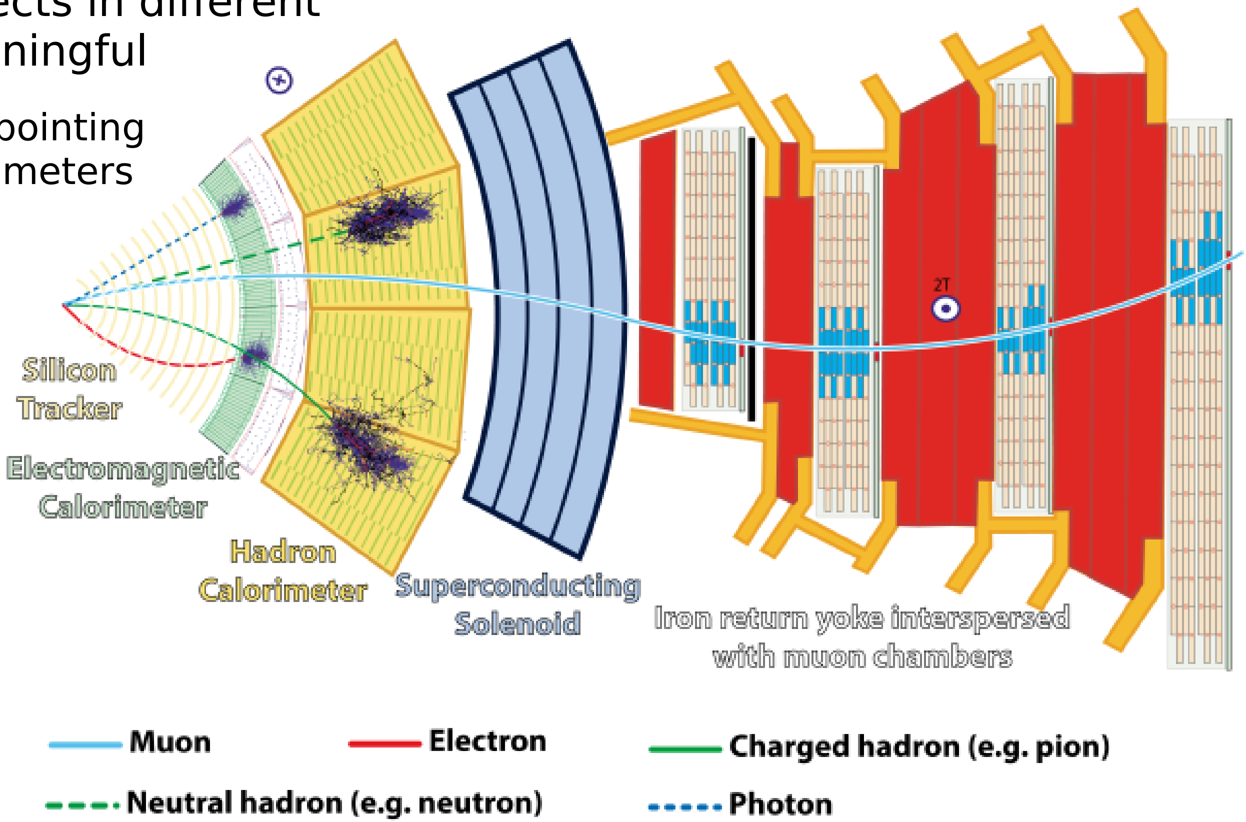
CMS example: similar story, but a bit more complicated events...

Treat each sub-detector in a similar way

- Low-level reconstruction of particle signals in sub-detector

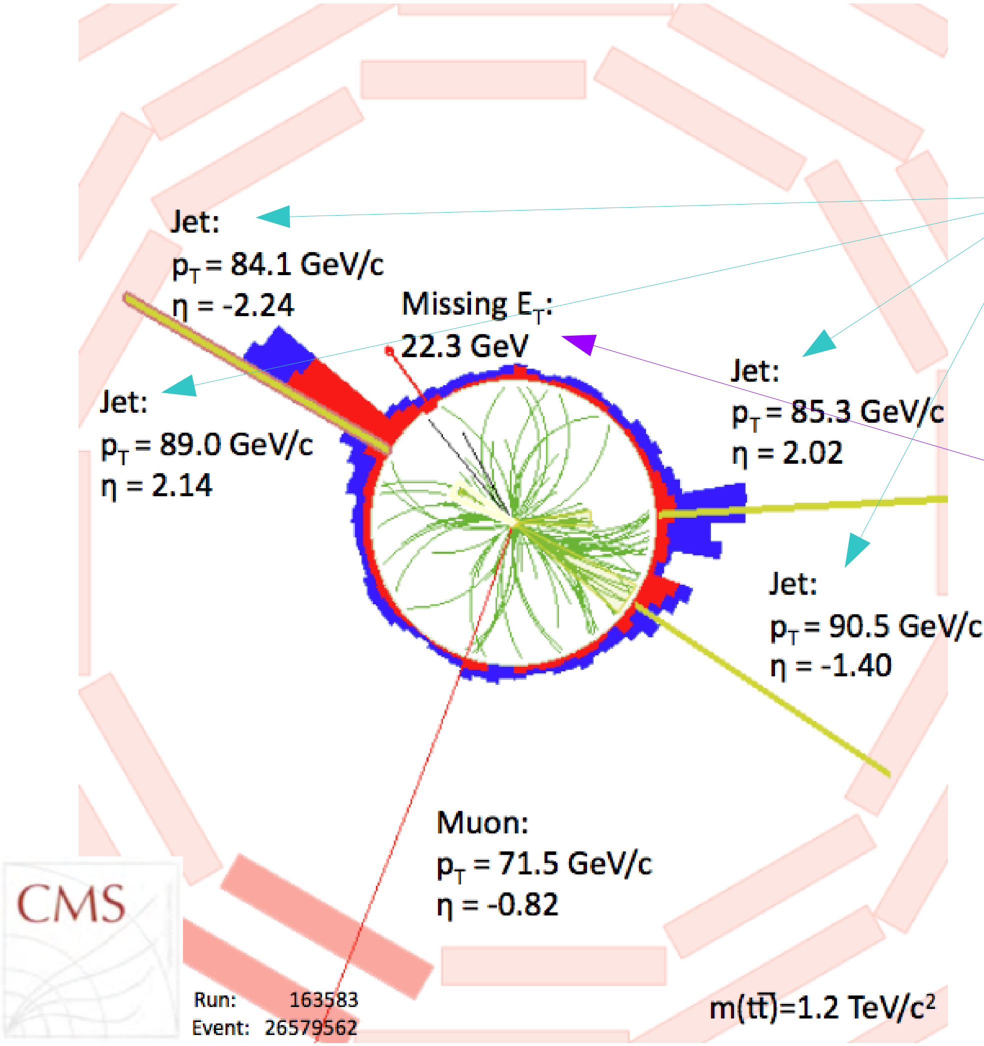
Then integrate information across sub-detectors

- Can use objects in one to seed reconstruction in others
- Associations between objects in different sub-detectors can be meaningful
  - E.g. tracks vs. no tracks pointing at a shower in the calorimeters



Step-by-step, build as complete a picture of the various particles, interactions in each event

CMS example: putting it all together



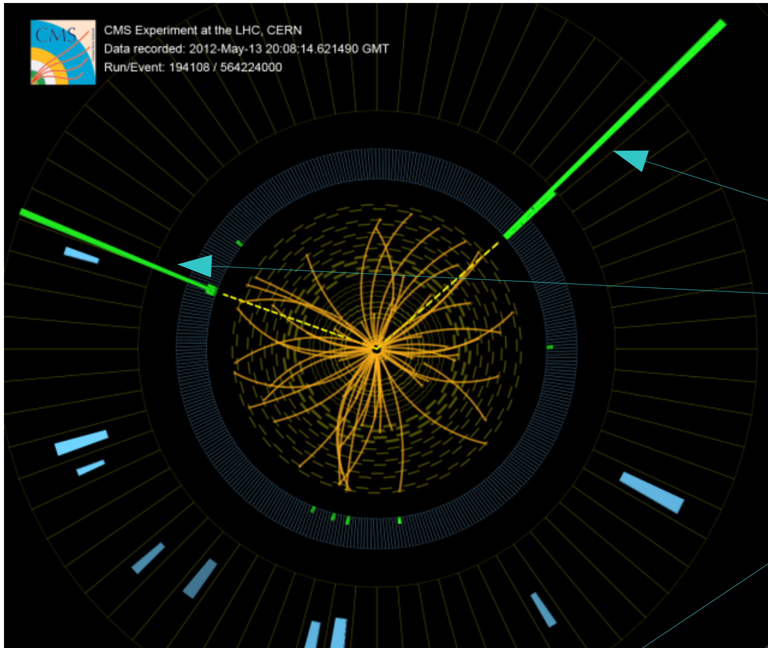
"Jet":
A spray of particles associated with hadronic or EM showers in the calorimeters

"Missing Et":
the imbalance in momentum transverse to the beam after summing over all observed particles

Non-zero means something escaped the detector undetected. (Or that we mis-measured something...)

– from Lukasz Kreczko, "The CMS Experiment at the LCH", seminar Feb 12, 2015

CMS example: putting it all together



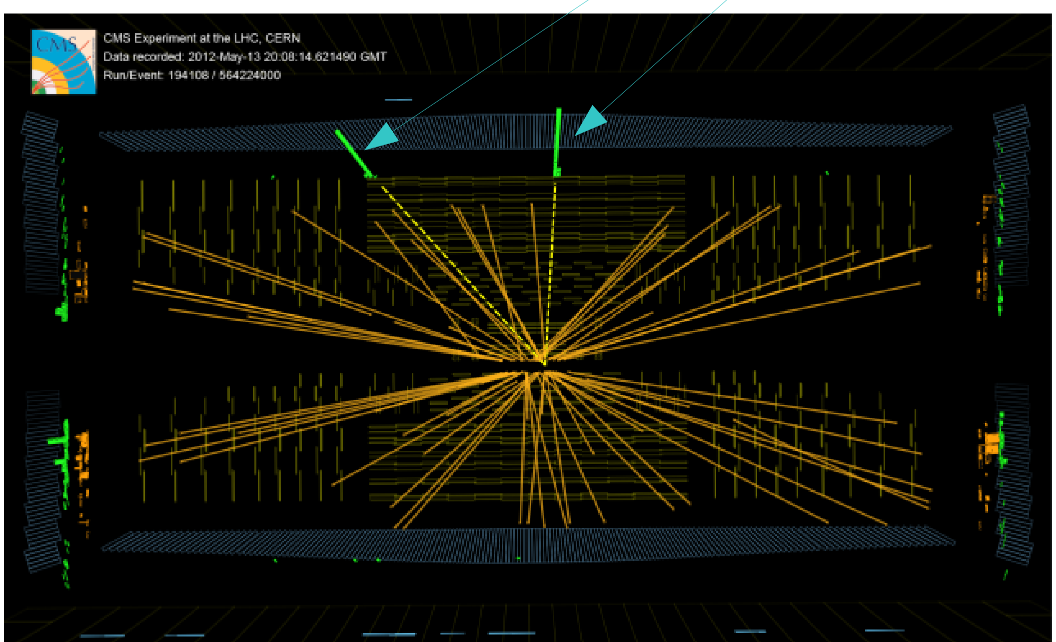
Photons:

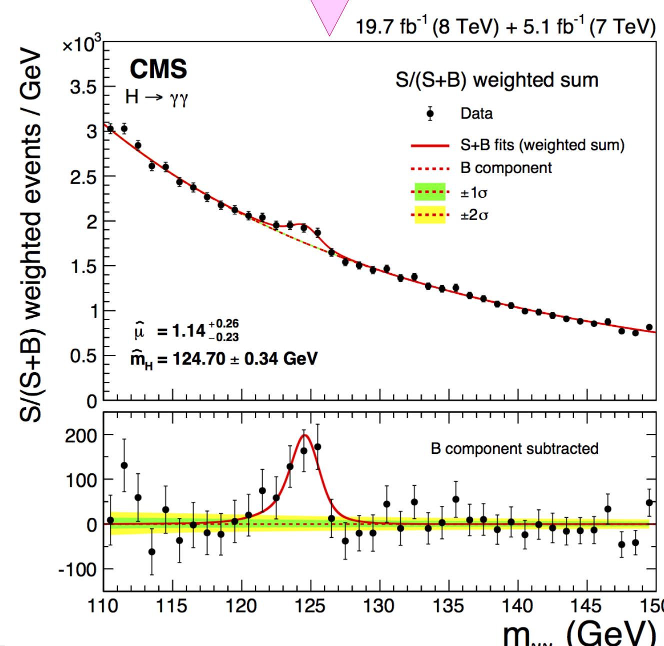
Large EM shower, no track pointing at it

Higgs to two photons candidate:

Calculate invariant mass of the two photons

Plot it along with all other two-photon

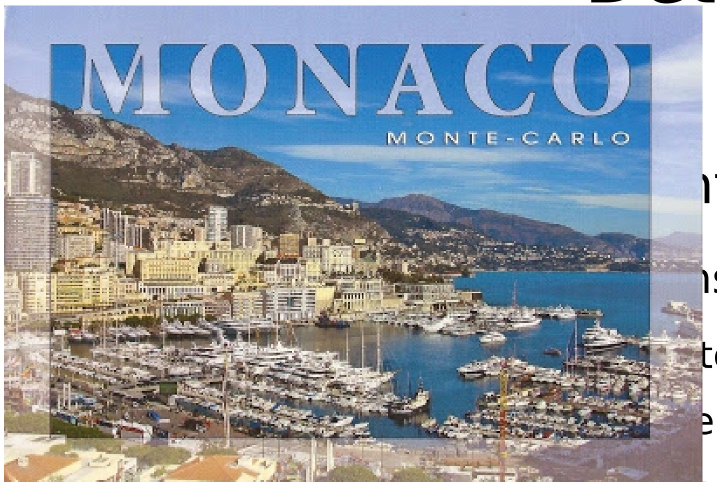




Detector simulation

- The technique: "Monte Carlo" simulation
  - Model detector response based on:
    - physics of the detector, known response of the electronics
    - test beam data, other experiments, etc.
  - Use random number generators to model various random processes
    - The details of particles created in primary interactions
    - Noise in the electronics
    - Physical effects in detector material (e.g., multiple Coulomb scattering, secondary interactions, fluctuations in ionization, fluctuation in showers)
- Need to simulate events, detector response in order to:
  - help tune, understand reconstruction performance
  - understand how a physics signal appears in the detector
  - help understand processes that mimic signal
- Importance to the computing problem: usually even slower than reconstruction!

Det



tor, know electroni

er experi

– use random number generatc ious ran

- The details of particles create actions
- Noise in the electronics
- Physical effects in detector m iple Coulomb scattering, secondary interactions, fluctu n, fluctuation in showers)

- Need to simulate events, detector response in order to:
  - help tune, understand reconstruction performance
  - understand how a physics signal appears in the detector
  - help understand processes that mimic signal
- Importance to the computing problem: usually even slower than reconstruction!

Detector simulation

- The technique: "Monte Carlo" simulation
  - Model detector response based on:
    - physics of the detector, known response of the electronics
    - test beam data, other experiments, etc.
  - Use random number generators to model various random processes
    - The details of particles created in primary interactions
    - Noise in the electronics
    - Physical effects in detector material (e.g., multiple Coulomb scattering, secondary interactions, fluctuations in ionization, fluctuation in showers)
- Need to simulate events, detector response in order to:
  - help tune, understand reconstruction performance
  - understand how a physics signal appears in the detector
  - help understand processes that mimic signal
- Importance to the computing problem: usually even slower than reconstruction!

The basic process

- Run the reconstruction on all events in relevant data sample
- Generate and reconstruct simulated data
- Analyze, re-analyze everything until you understand the data
- Publish papers
- Become famous!

The basic process

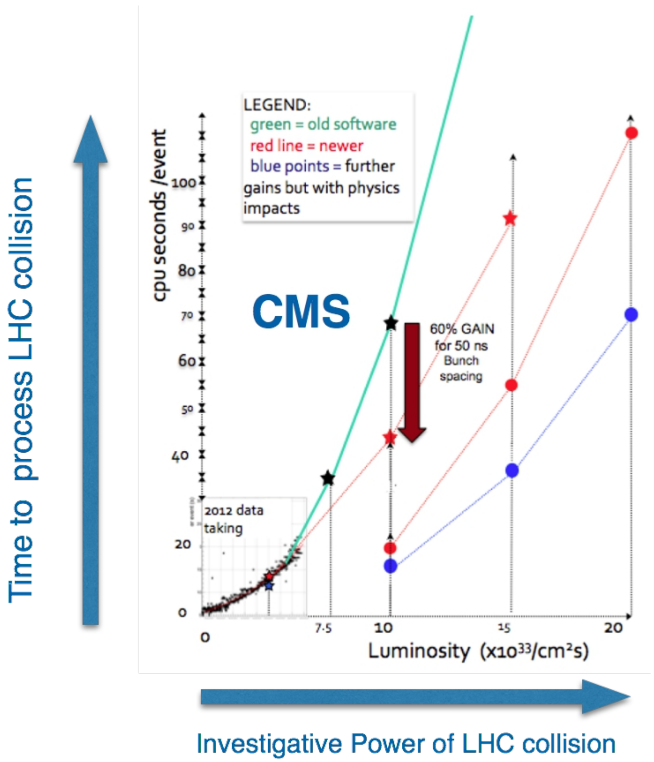
- Run the reconstruction on all events in relevant data sample
- Generate and reconstruct simulated data
- Analyze, re-analyze everything until you understand the data
- Publish papers
- Become famous!

So what's the problem??

The computing problem

How long does it take to do all that?

- Consider CMS reconstruction
  - 10 – 30 sec in Run 1,
    - Upwards of 1 minute for Run 2 and beyond
  - Adding ~ billion events per year
- MicroBooNE reconstruction
  - Currently about 5 min
  - Millions of events



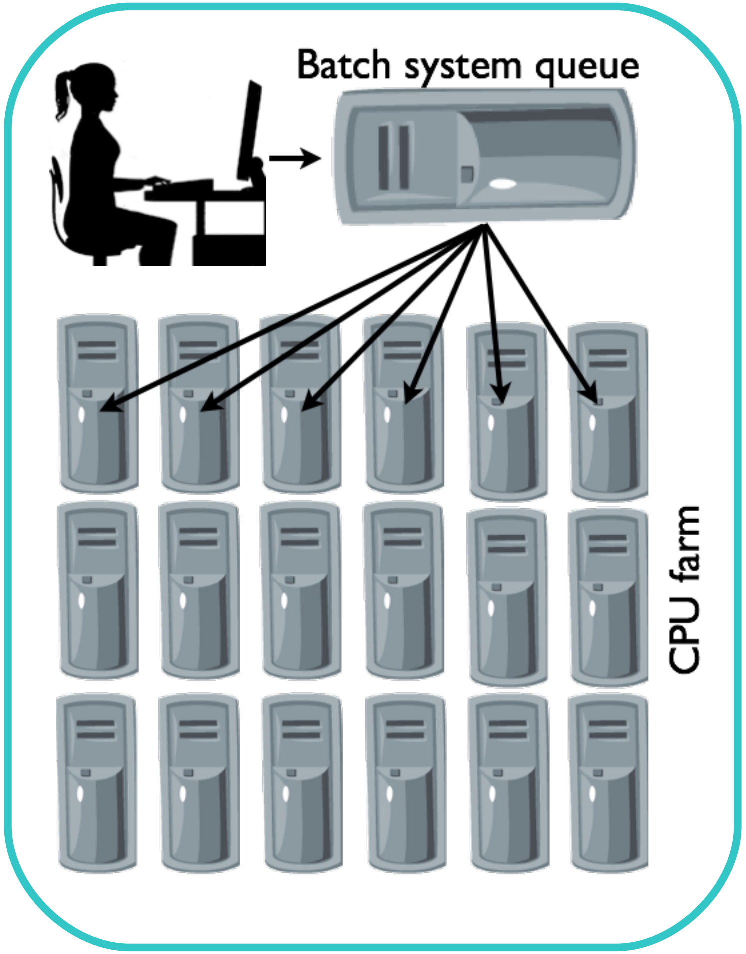
- In both cases, can analyze each event individually
  - Use lots of computers!!
  - Run jobs in parallel

Data processing

Batch farms

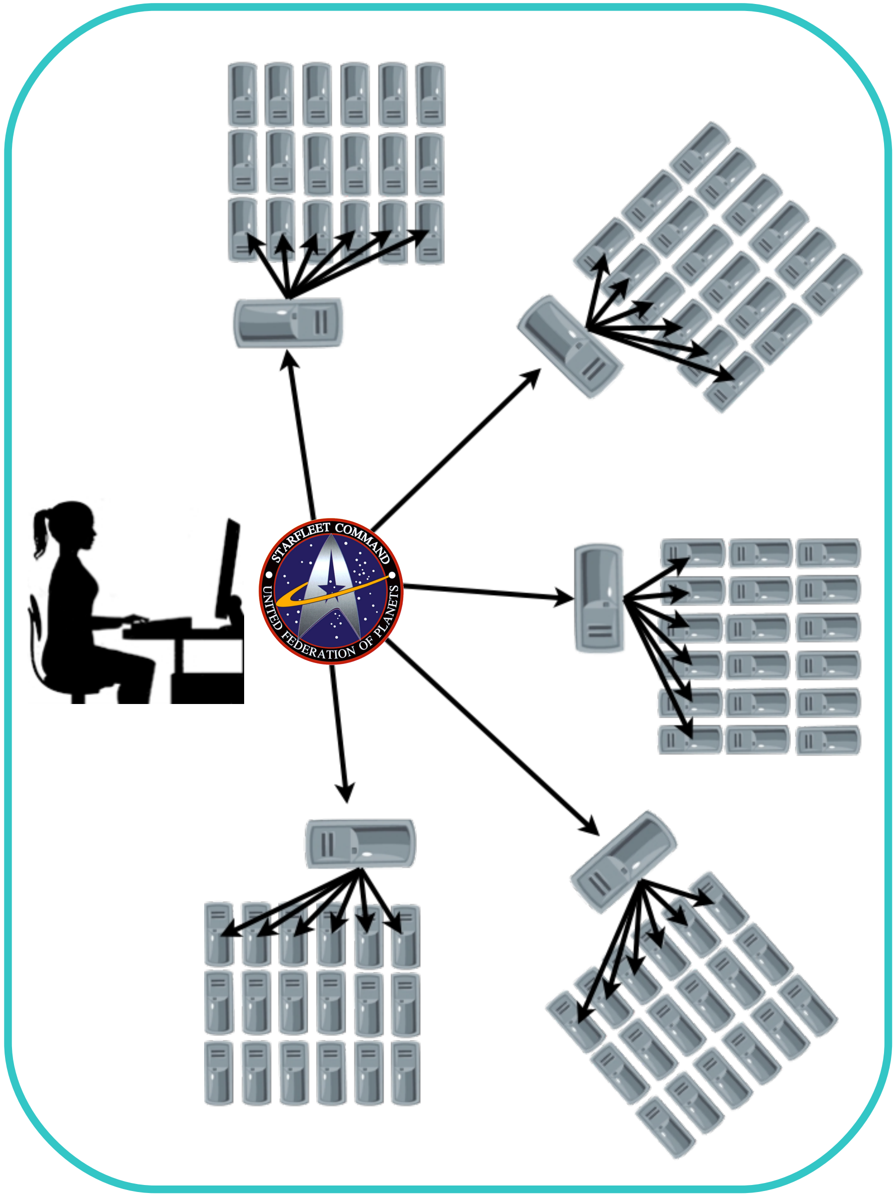
- Manage computers in CPU farms with batch systems
  - Many thousands of cores in the farm
  - Submit individual jobs to a queue
  - Jobs are removed from the queue and run on free cores
  - Fills the farm with running jobs
- Great, but...
  - Difficult to host all the computing needed for large experiments (e.g., CMS) at one location
  - Solution: distribute computing resources across multiple sites

The grid!



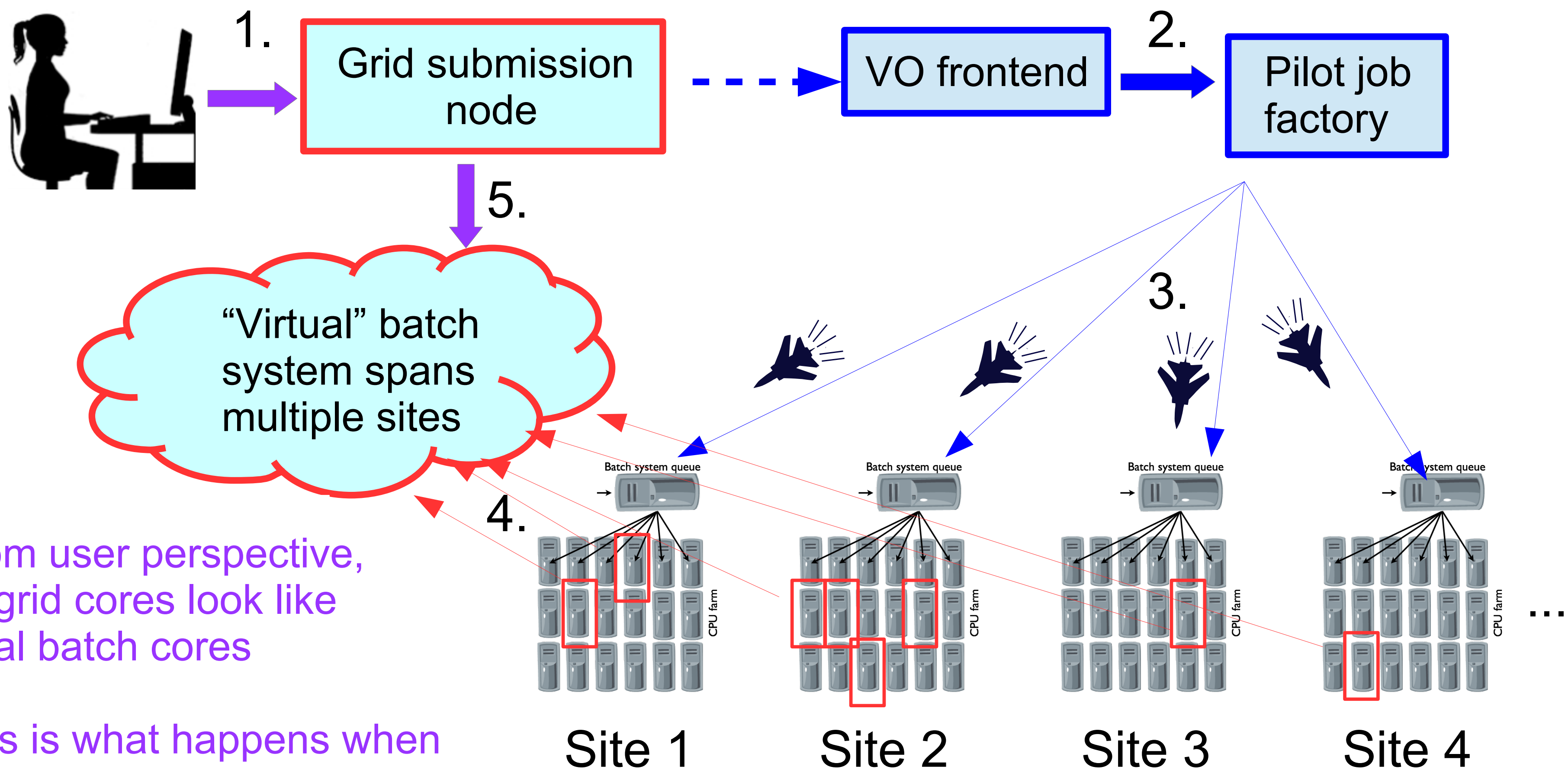
Grid computing

- A distributed collection of farms in a “trust federation”
  - “Virtual organizations” of users
  - VOs own / have allocations at sites within grid
  - Can also share with other VOs
    - Provide allocations
    - Allow “opportunistic” access
- Common grid “middleware”
  - To manage flow of jobs across sites
  - To provide support for authentication, data movement,workflows
  - To present a single interface to users



Grid computing

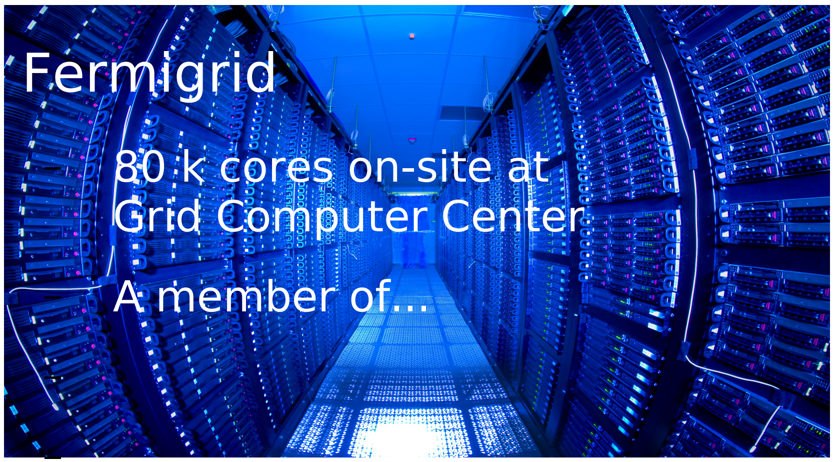
- For example, job submission



From user perspective, all grid cores look like local batch cores

This is what happens when you submit to Fermigrid

Grid computing

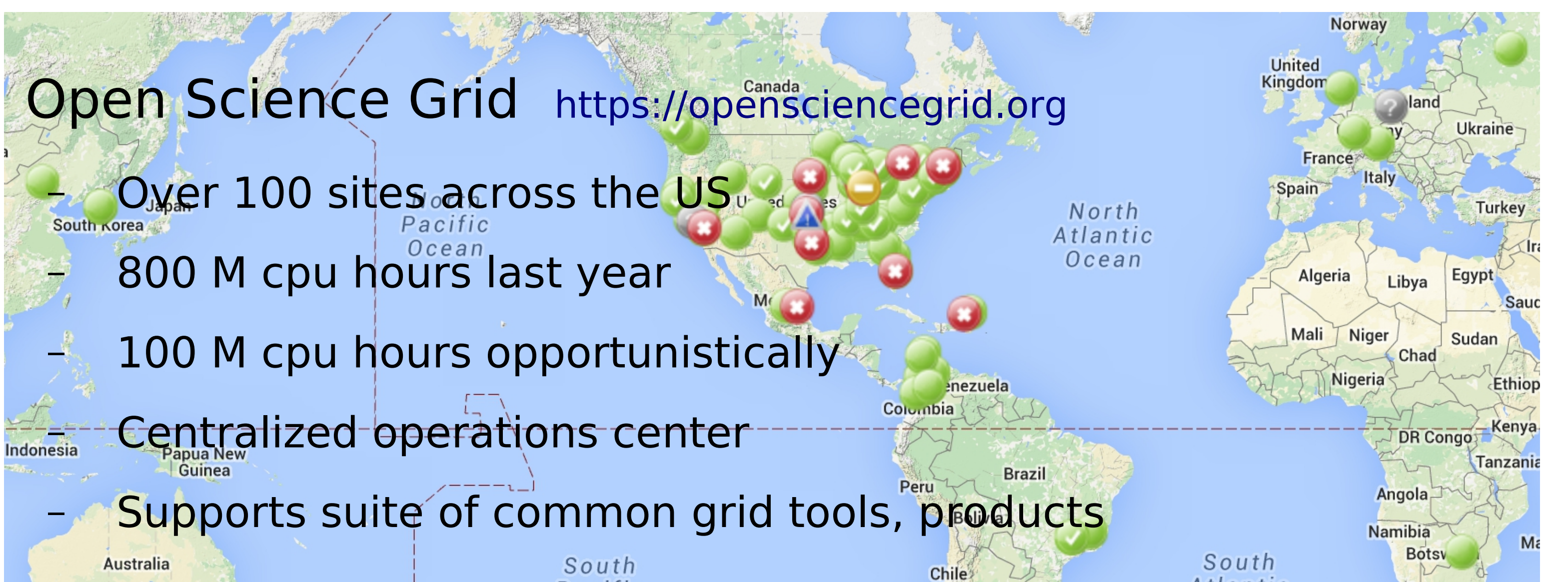


- Fermigrid
  - 80 k cores on-site at Grid Computer Center
  - A member of…
  -
  -

The grid has dramatically **improved efficiency** of support

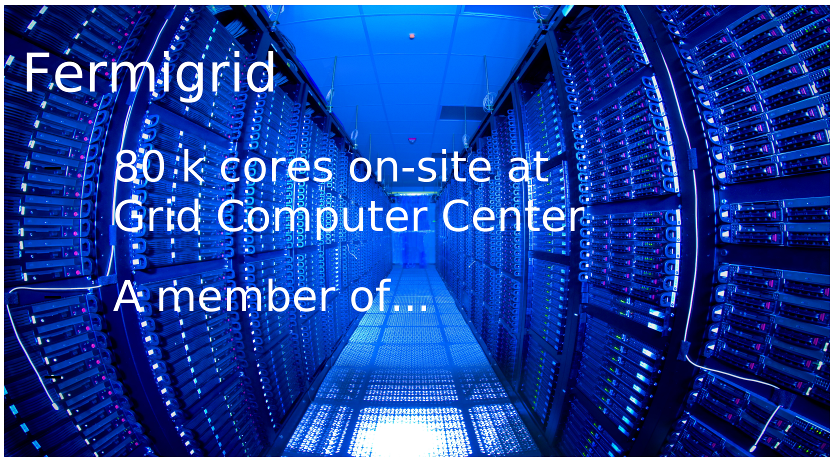
Opportunistic access to spare cycles in one VOs allocation provides flexibility in capacity.

A boon to handling peak demand within an otherwise fixed allocation



- Open Science Grid https://opensciencegrid.org
  - Over 100 sites across the US
  - 800 M cpu hours last year
  - 100 M cpu hours opportunistically
  - Centralized operations center
  - Supports suite of common grid tools, products

Grid computing

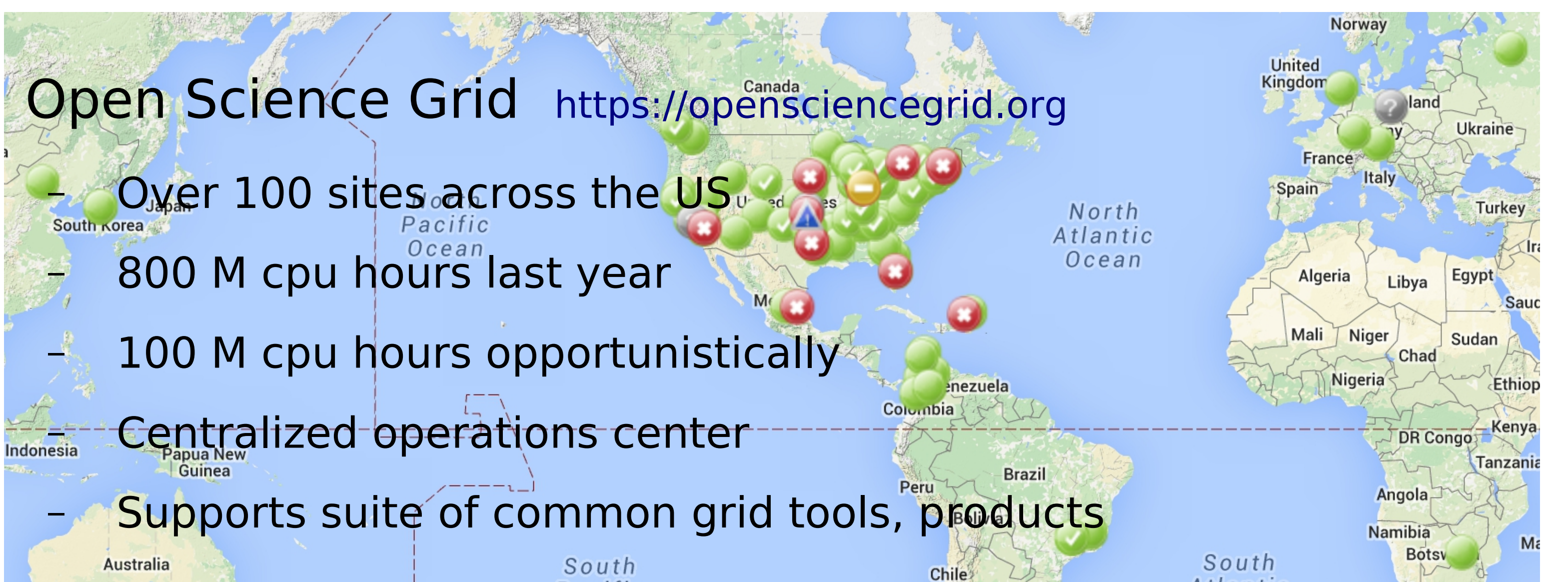


- Fermigrid
  - 80 k cores on-site at Grid Computer Center
  - A member of…

The grid has also been an important, **enabling resource**

Allows small experiments, institutions to work with large-scale computing with a minimum of overhead

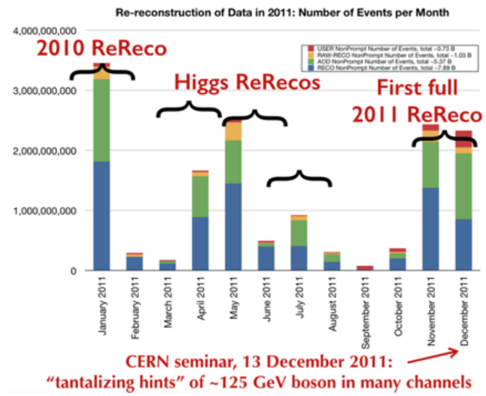
Fermilab plays a key role in support and development



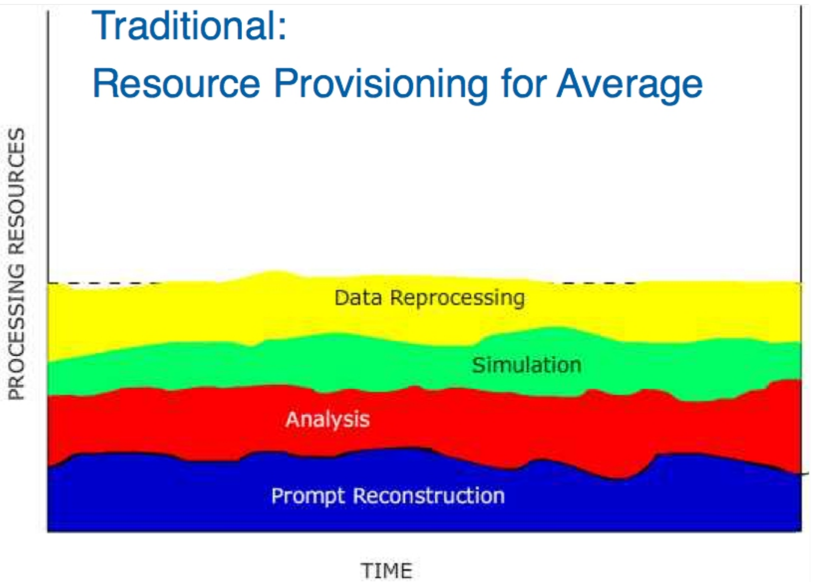
- Open Science Grid https://opensciencegrid.org
  - Over 100 sites across the US
  - 800 M cpu hours last year
  - 100 M cpu hours opportunistically
  - Centralized operations center
  - Supports suite of common grid tools, products

Evolving the grid model

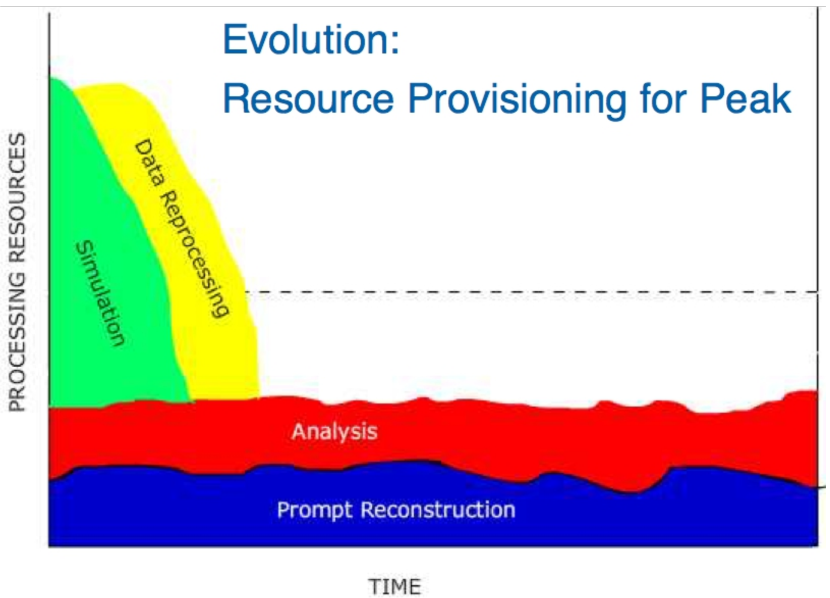
- Computing demand is not flat
  - Conference cycles, accelerator schedules, holidays, etc.



- Not well matched to budgeting, resource provisioning model:



Typically planning, provisioning performed annually



Would be more efficient to match provisioned resources to demand profile

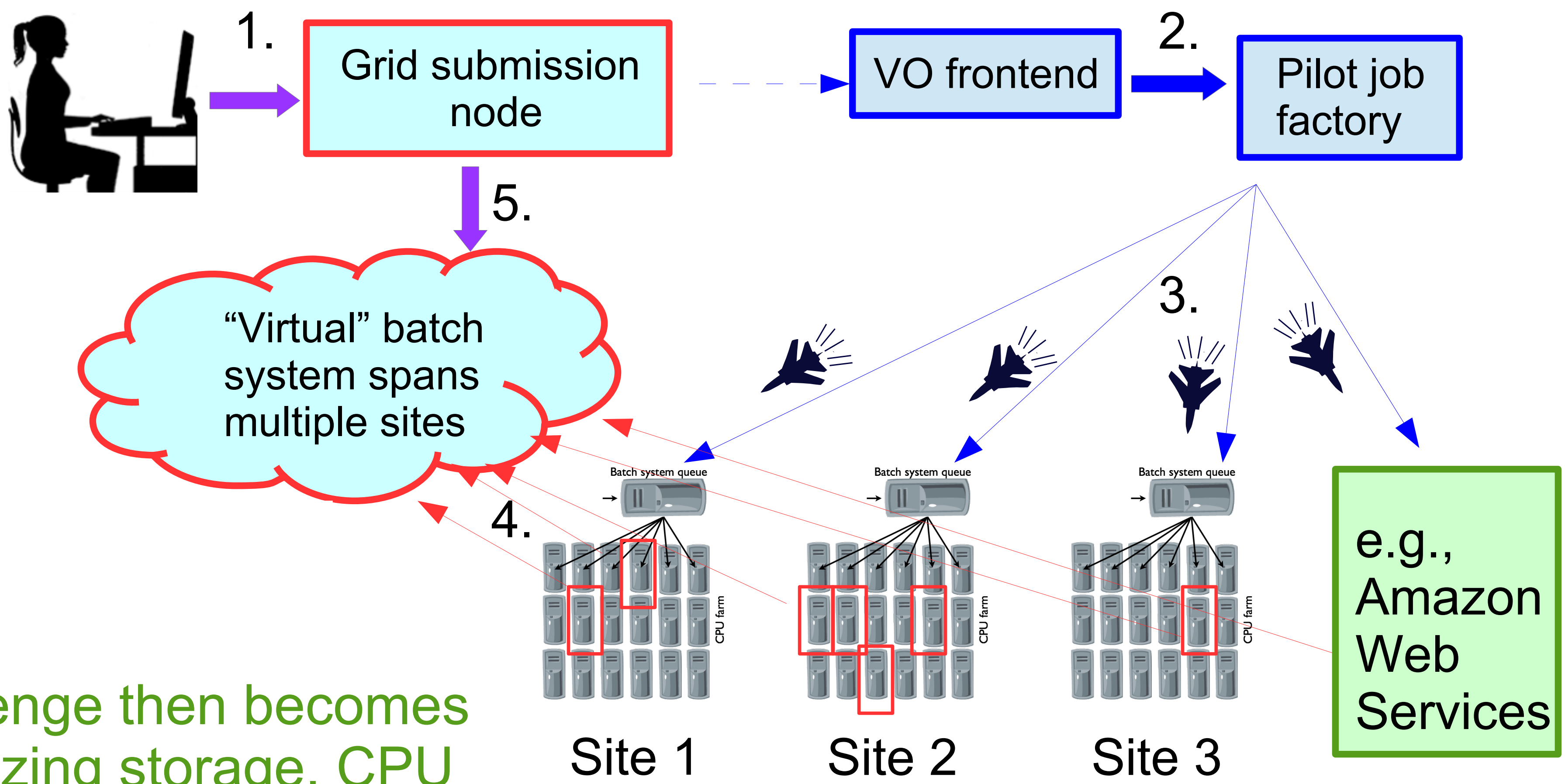
Cloud computing

Definition

> "The practice of using a network of remote servers hosted on the Internet to store, manage and process data, rather than a local server or personal computer."
>
> *– from the Internet...*

- Commercial clouds: an "economic model"
  - Pay for CPU as needed
  - Allows elasticity in resources available to experiments on short time scales
  - Subject to a spot market price
- Strategy: integrate into existing grids
  - Provision nodes
  - Then send pilot jobs as if regular grid nodes
  - Brings them into the "trust federation"

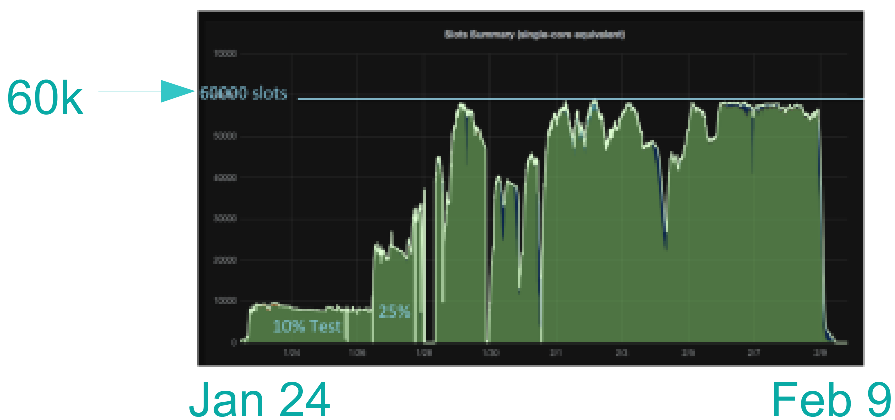
Cloud computing



Challenge then becomes optimizing storage, CPU usage to minimize cost

Cloud computing

- A number of experiments are currently working with AWS
  - Fermilab, CMS, NOvA among others



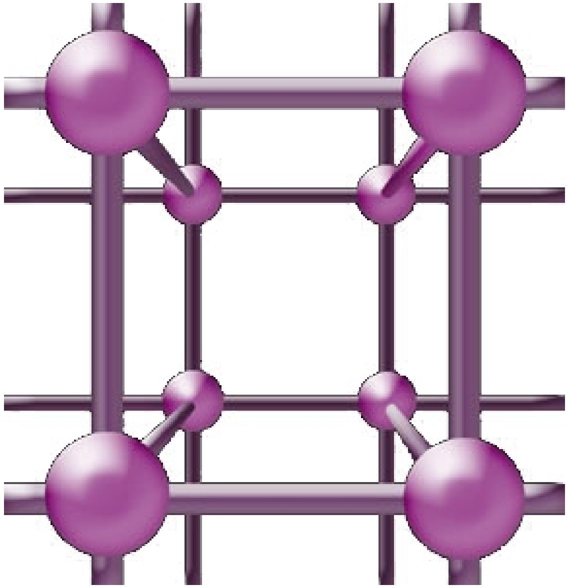
- HEPCloud project
  - Add 50k cores to CMS for 1 month
  - Have demonstrated 58k
  - Largest cloud project in HEP

- Other on-demand cloud services
  - Storage
  - Networking

HPC

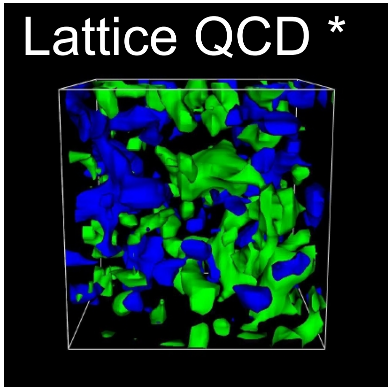
- Until now, talking about “high throughput computing” (HTC)
  - Independent, sequential jobs that can be individually scheduled on many different computing resources across multiple administrative boundaries
  - Has been the primary computing model in HEP

- HPC: “high performance computing”
  - Tightly coupled parallel jobs
  - Execute within a particular site with low-latency interconnects – jobs talk to each other!
  - Ubiquitous in modeling of phenomena in 3D/4D
    - Partial differential equations!

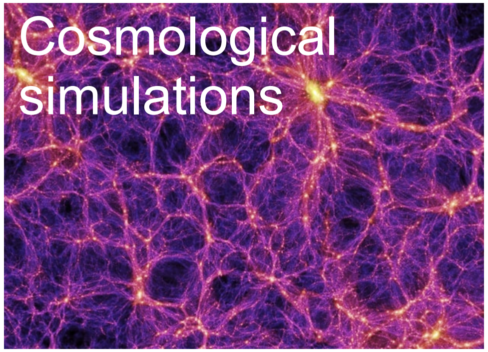


- Based on a “grant model”
  - Competitive, peer reviewed allocation requests

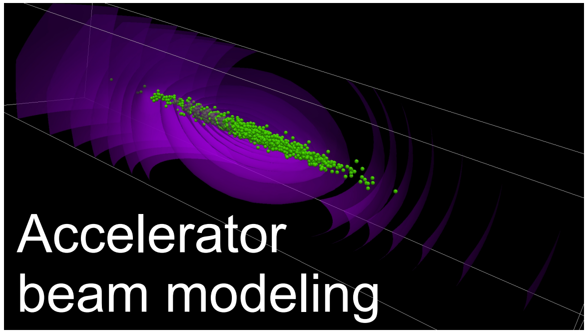
History of HPC in HEP / cosmology at Fermilab



Calculations of strong force interactions that cannot be calculated using traditional methods



Used extensively by DES to model observed data



Collective beam dynamics

* LQCD clusters at Fermilab

HPC

- HPC facilities at Fermilab
  - For accelerator modeling:

    Wilson cluster used for development of accelerator modeling applications

  - For lattice QCD:

    Operate at hundreds of trillions of operations per second (TFLOPS)

    Fastest available machines about 100x bigger:

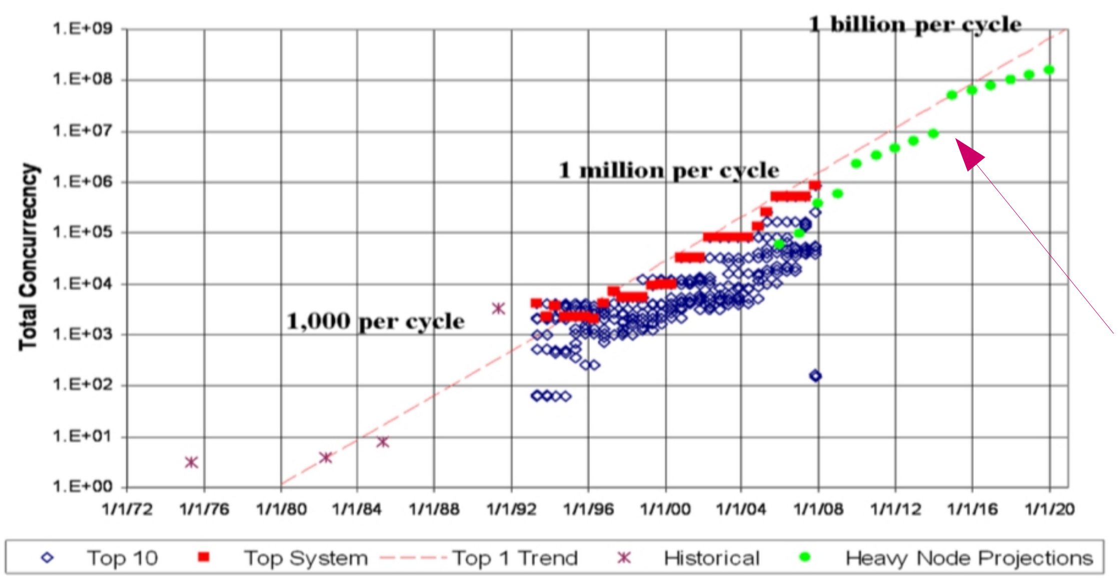
    10^{15} operations per second (Peta-FLOPS)

The next bigger thing in HPC

- Exascale

  (~10^{18} operations per second)

Department of Energy: Advanced Scientific Computing Research (ASCR) plans for exascale computing



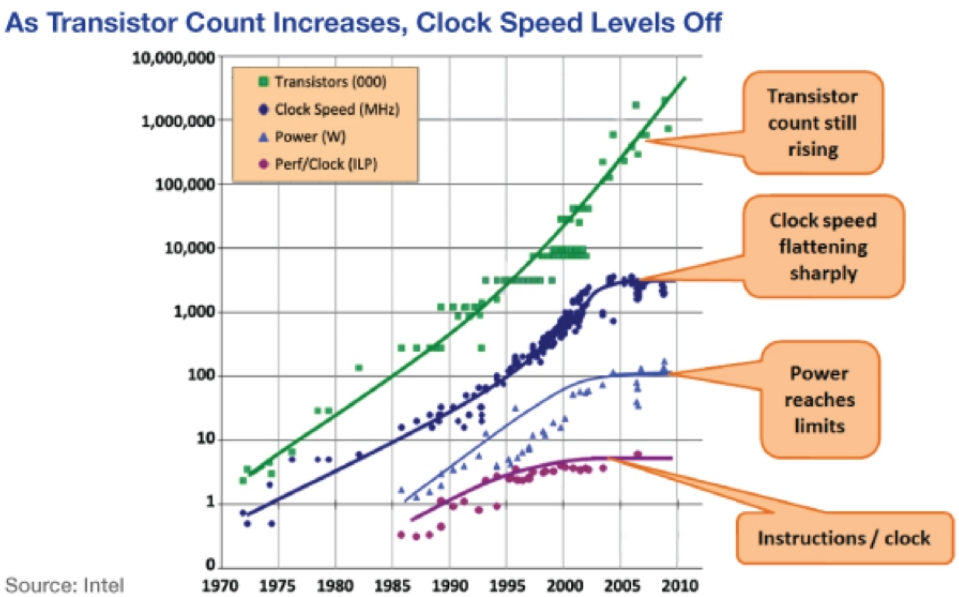
Total concurrency = value on plot x a few billion cycles per second

Exascale opens exciting possibilities for HEP / Fermilab science program

New directions in software

Software / hardware trends

- Traditionally, HEP software optimized to run on “simple” architectures
  - i.e., single CPU



- Growth in computing power now appearing in the form of more rather than faster CPUs

- A new technology: general purpose computing on graphics processing units (GPGPU)

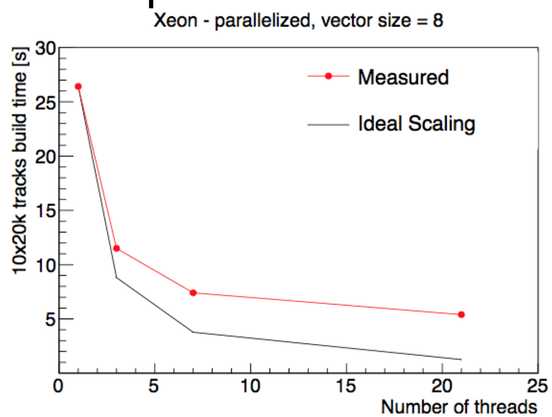
Need a new approach to get the full benefits of these developments

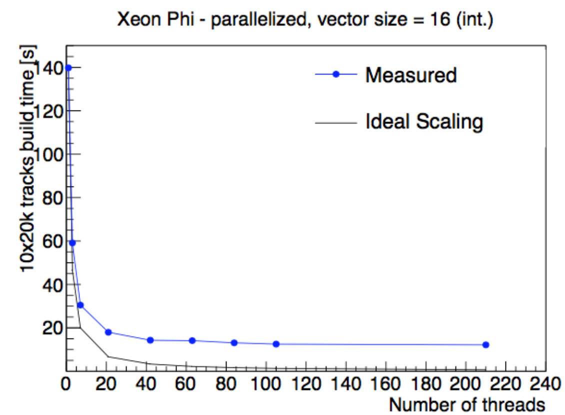
Multi-threading

- Running multiple execution paths within a single program
- Different schemes possible
  - Run different events in different threads
    - Saves on memory
  - Break single events into different threads
    - Better optimization results with even less memory
- GPUs and newer CPUs/co-processors
  - Increased speed effectively by performing more calculations per cycle
  - “Vectorize” programs to allow parallel execution of loops, other workflows

Multi-threading

Vectorized CMS track reconstruction on multi-core co-processor architecture *– Cerati, et al., arXiv:1505.04540v1 [physics.ins-det]*

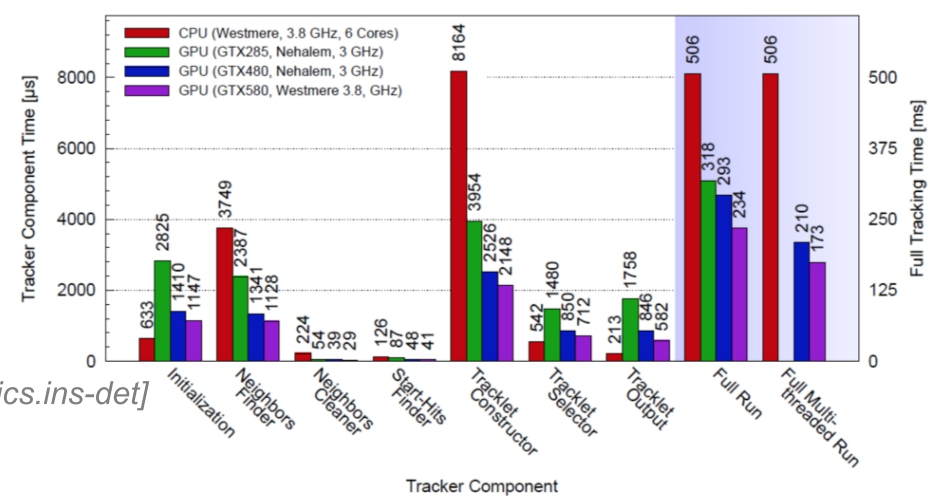




LARGE gains possible

Requires new programming skills

ALICE track reconstruction on GPUs



– Lujan and Halyo, arXiv:1505.03137v2 [physics.ins-det]

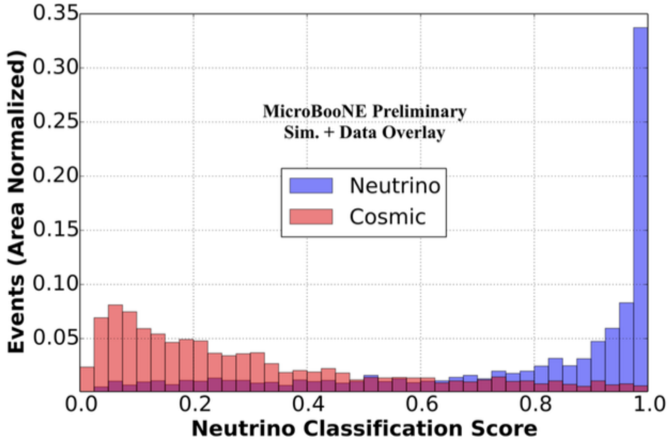
Machine learning

- Another **HUGE** area with a long history in HEP
  - Boosted decision trees
  - Artificial neural networks (feed-forward NN)

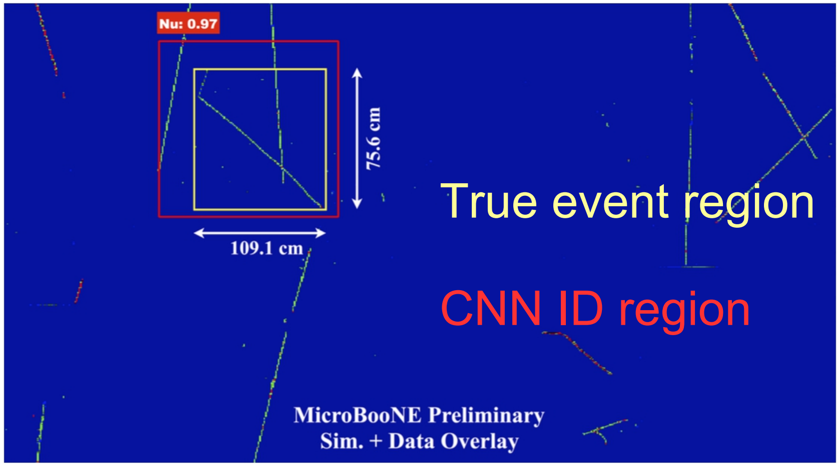
- A recent development: “convolutional neural networks” (CNN)
  - An advance on previous generation of neural networks
    - Much more complex network architecture
    - Yet still trainable
  - Ideally suited to image processing
  - Some exciting early tests in LAr TPC data (images!!)

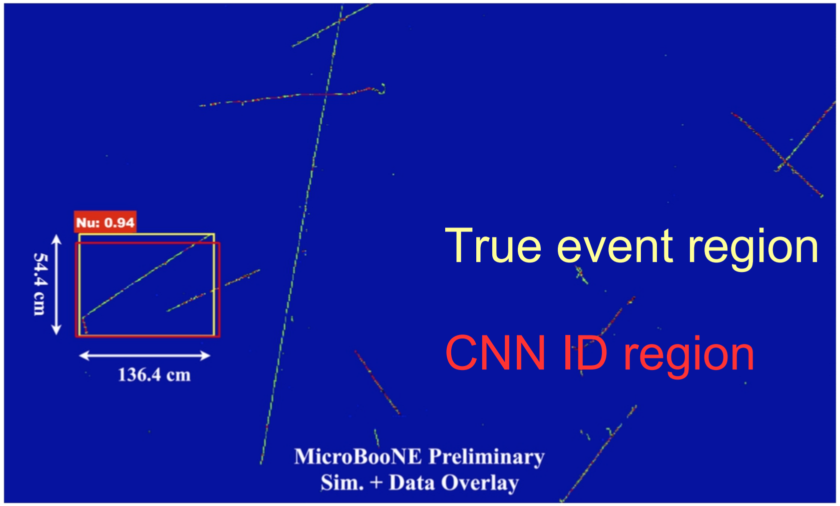
CNN tests at MicroBooNE

Neutrino / cosmic identification after training



Neutrino event ID in LAr TPC images





A promising early result

Training CNNs requires a lot of computing!
– GPUs, HPC...

Summary

- Computing is an essential enabling technology in HEP
  - Fermilab has a broad program in scientific computing
  - Succeeds via dedicated collaboration between physicists, computer professionals
- Vastly greater resources will be needed to meet future demand
- Meeting this challenge will require
  - innovative new ways of processing the data
  - new and better software algorithms for extracting the physics
- Scientific computing ecosystem is evolving rapidly
- An exciting time to be involved!

Acknowledgements

- Thank you to:
  - Jim Amundson, Burt Holzman, Rob Kutschke, Adam Lyon, Tia Miceli, Ruth Pordes, Anne Schukraft, Eric Vaandering for helpful comments and material
  - Numerous unnamed members of
    - SCD
    - CMS
    - MicroBooNE
    - DUNE

    for content
- Special “thank you” to Oli Gutsche for allowing massive theft of presentation materials
- 2016 Undergraduate Lecture Series organizers for this opportunity!

Backup slides

CMS simulated events vs. instantaneous luminosity