



# Computing at Fermilab

**Marco Mambelli**

Scientific Computing Division

*Fermilab Summer Students School*

**July 23, 2024**



# What is scientific computing?

## Computational science

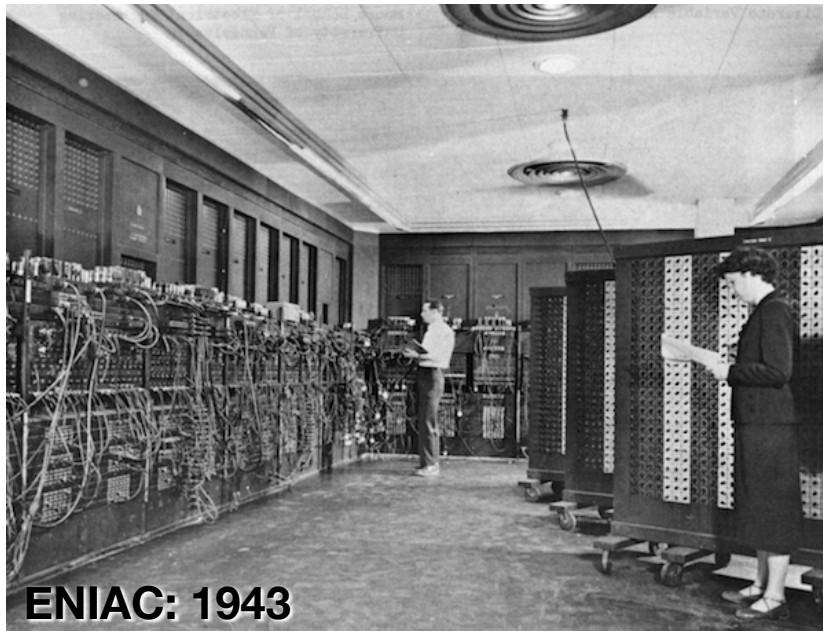
From Wikipedia, the free encyclopedia

*Not to be confused with computer science.*

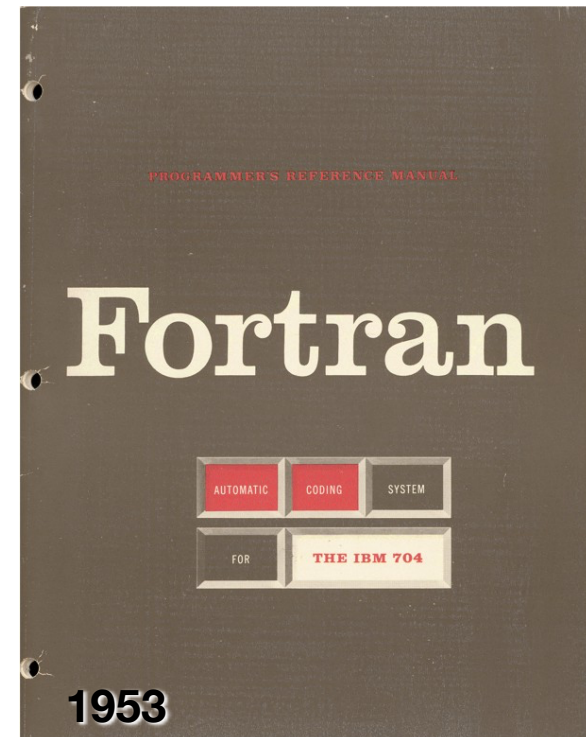
**Computational science** (also **scientific computing** or **scientific computation (SC)**) is a rapidly growing multidisciplinary field that uses advanced computing capabilities to understand and solve complex problems. It is an area of science which spans many disciplines, but at its core it involves the development of models and simulations to understand natural systems.

- **Algorithms** (numerical and non-numerical): **mathematical models**, **computational models**, and **computer simulations** developed to solve **science** (e.g., **biological**, **physical**, and **social**), **engineering**, and **humanities** problems
- **Computer and information science** that develops and optimizes the advanced system **hardware**, **software**, **networking**, and **data management** components needed to solve computationally demanding problems
- The computing infrastructure that supports both the science and engineering problem solving and the developmental computer and information science

In practical use, it is typically the application of **computer simulation** and other forms of **computation** from **numerical analysis** and **theoretical computer science** to solve problems in various scientific disciplines. The field is different from theory and laboratory experiment which are the traditional forms of science and **engineering**. The scientific

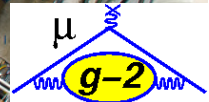
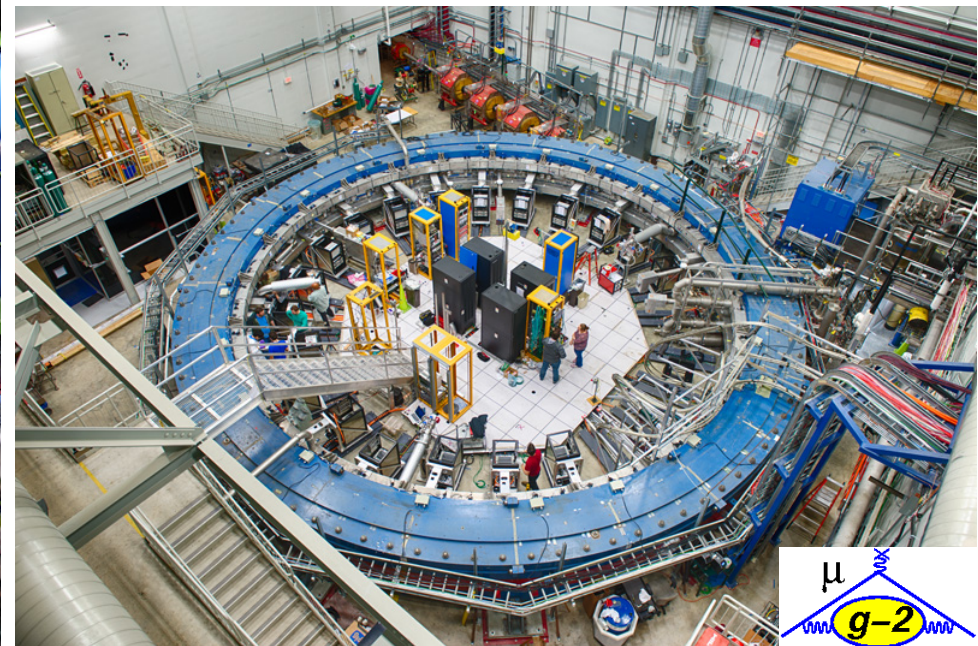
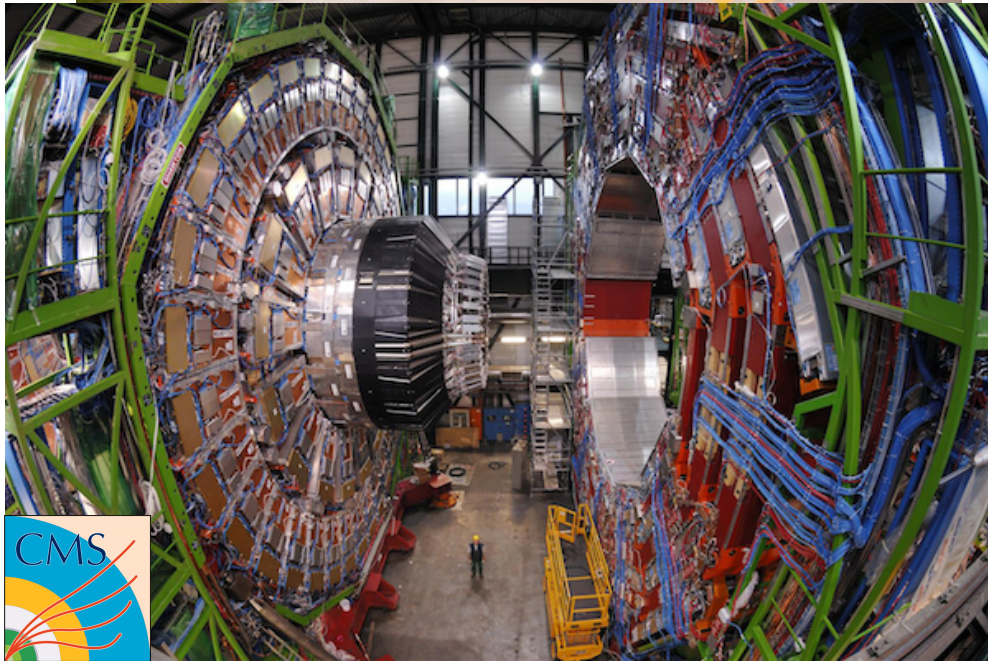
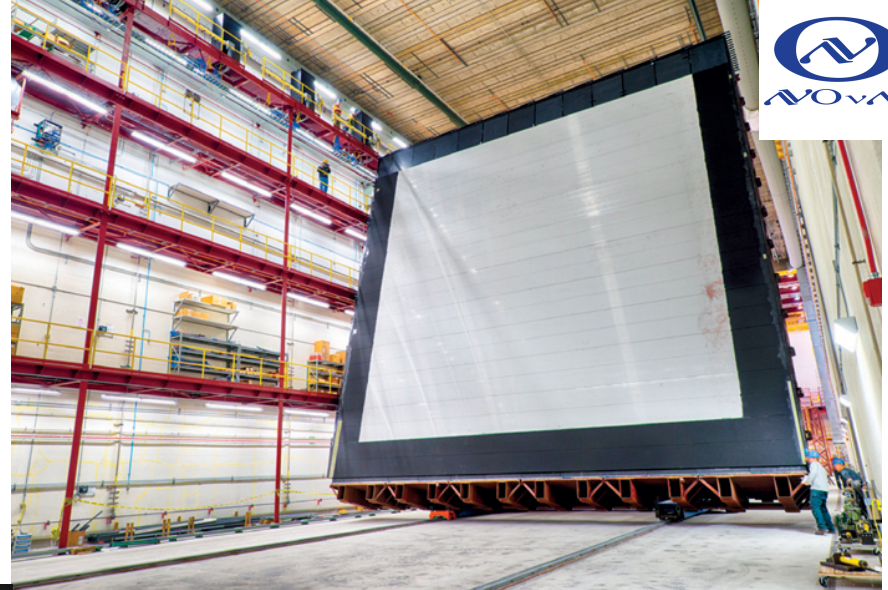


**ENIAC: 1943**





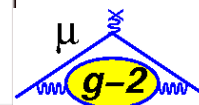
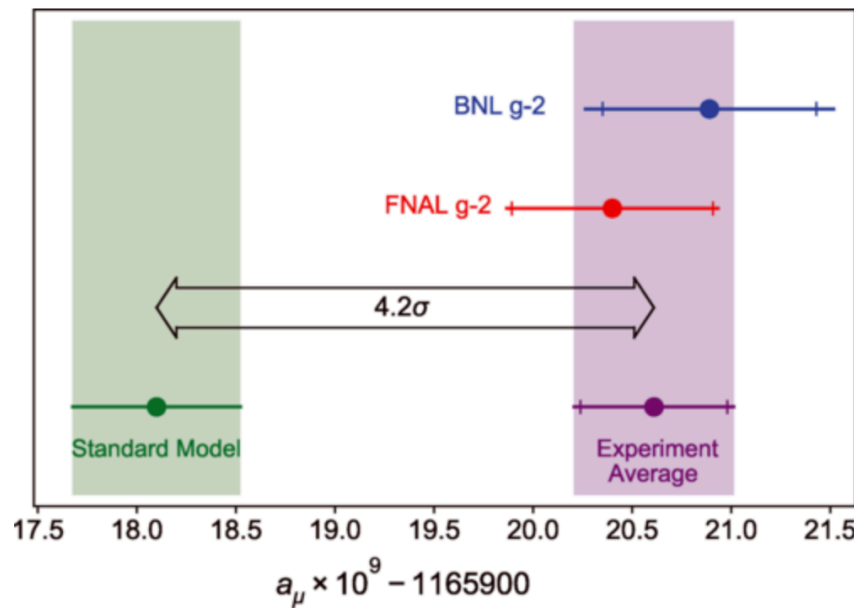
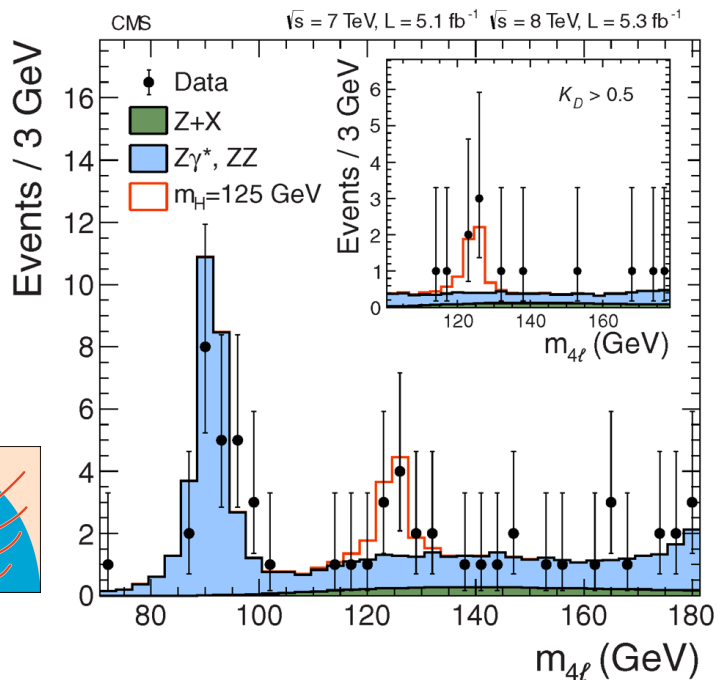
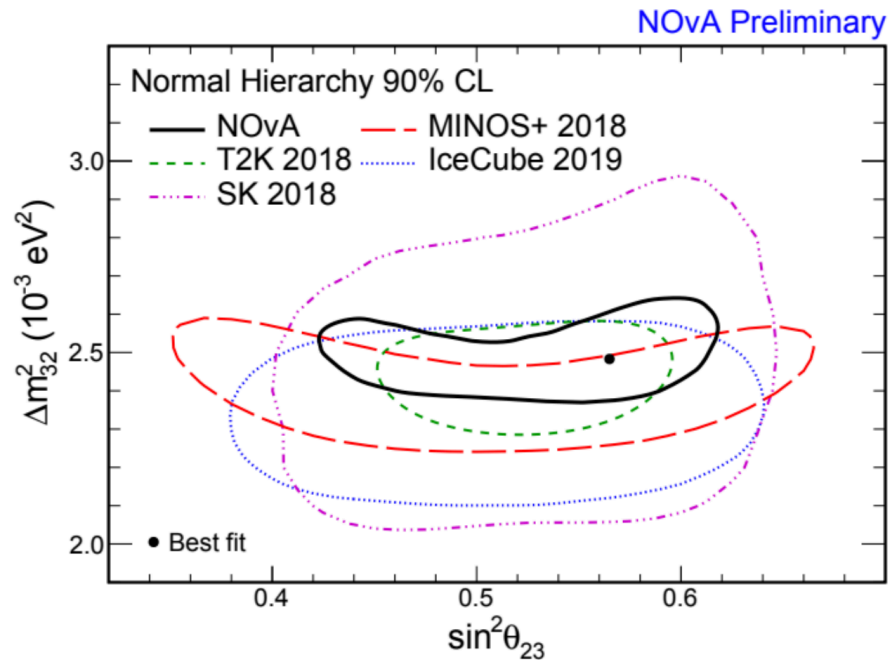
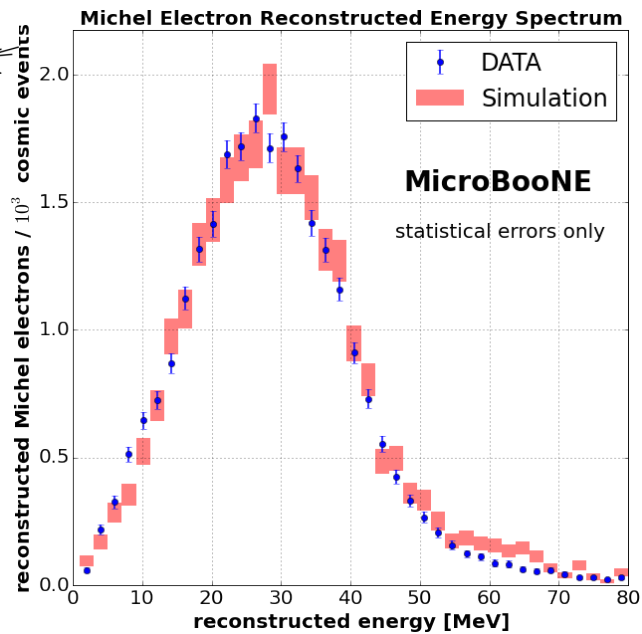
# Scientific Computing in HEP: Getting from here...





...to here

**μBooNE**

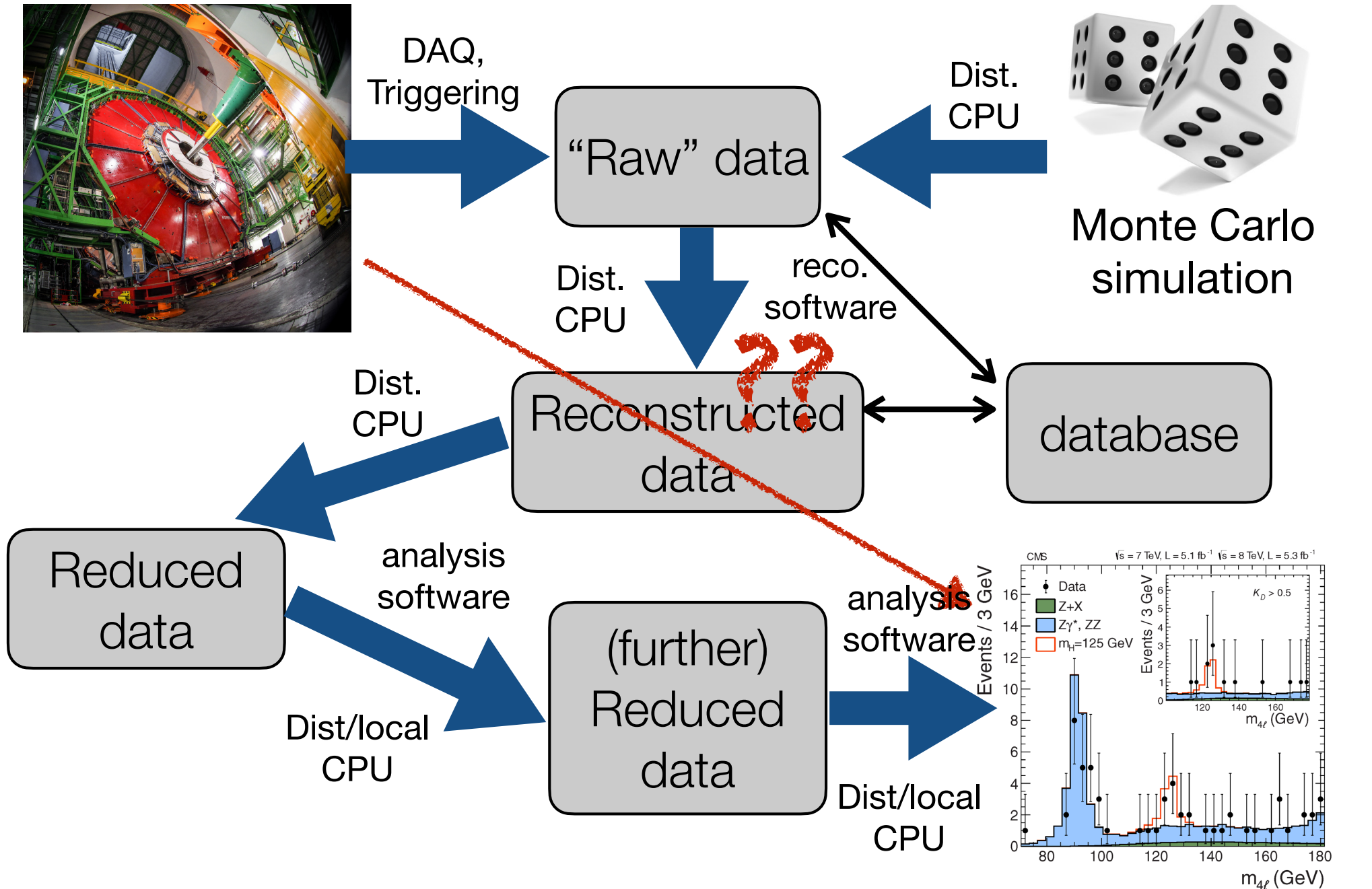




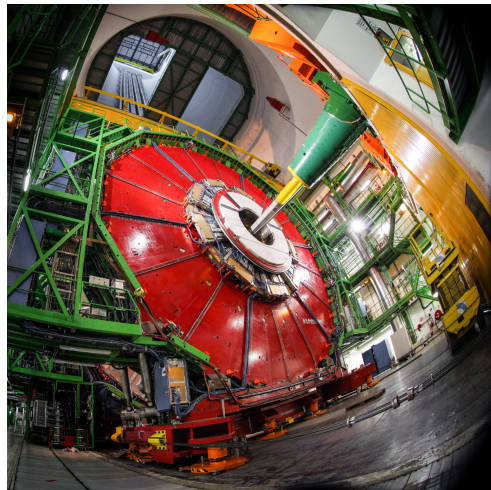
Software



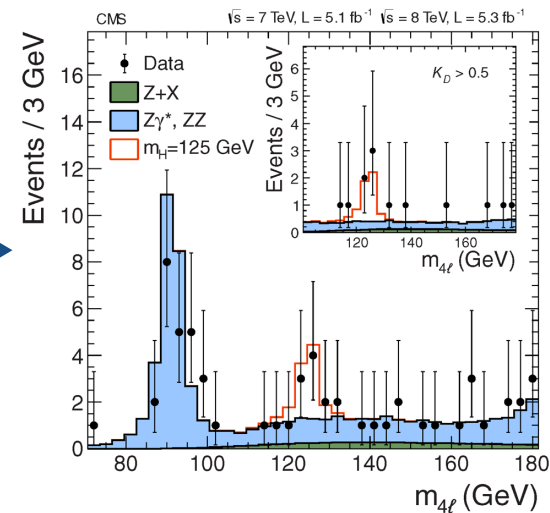
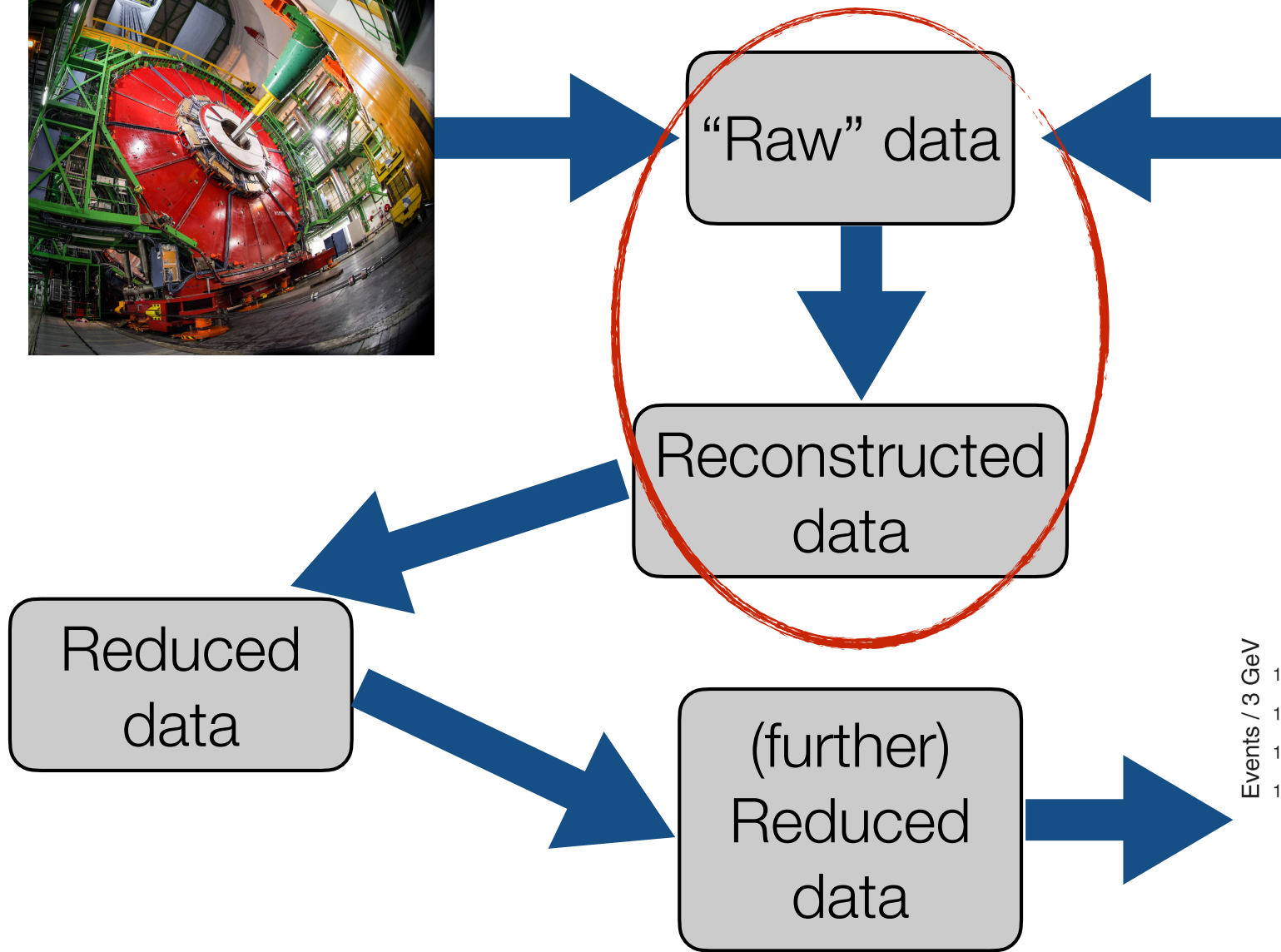
# The process



# Reconstruction



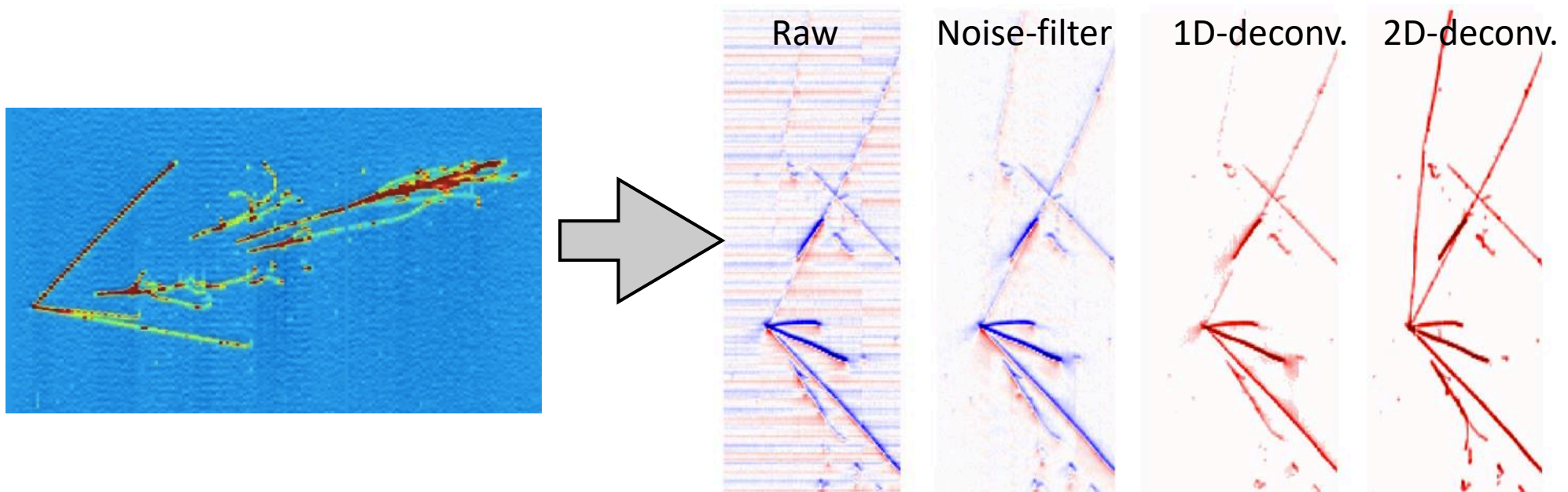
Monte Carlo  
simulation



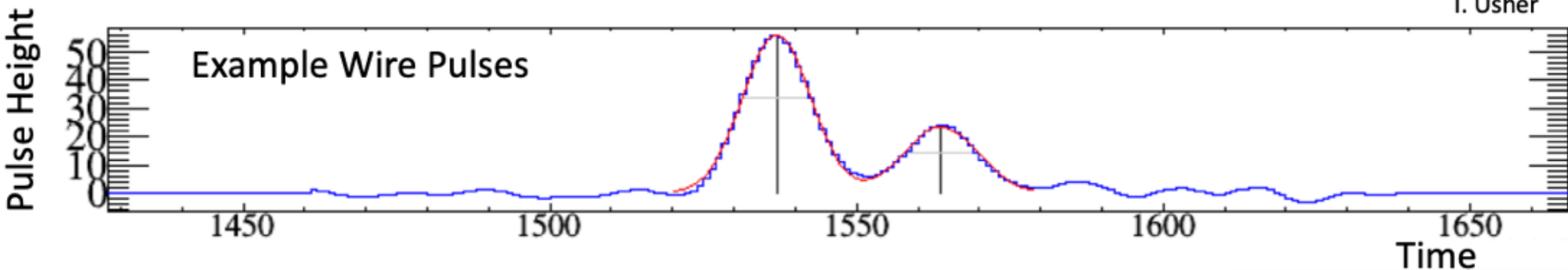


# Reconstruction: LArSoft (MicroBooNE)

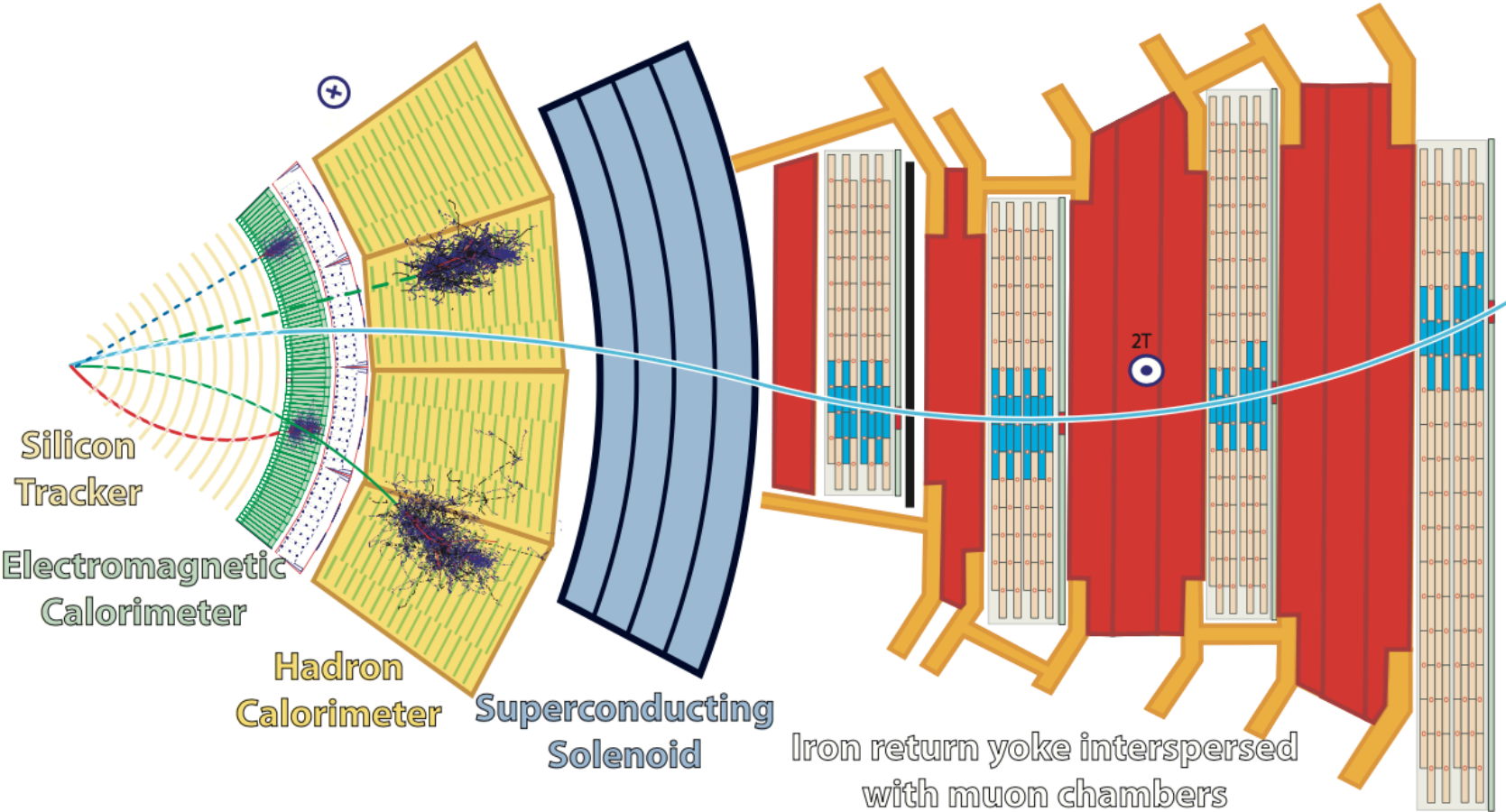
- **LArSoft**, developed at Fermilab, is a common code base for LAr neutrino detectors
  - Used by MicroBooNE, SBND, ICARUS, DUNE
  - Includes tools for both reconstruction and data analysis



T. Usher



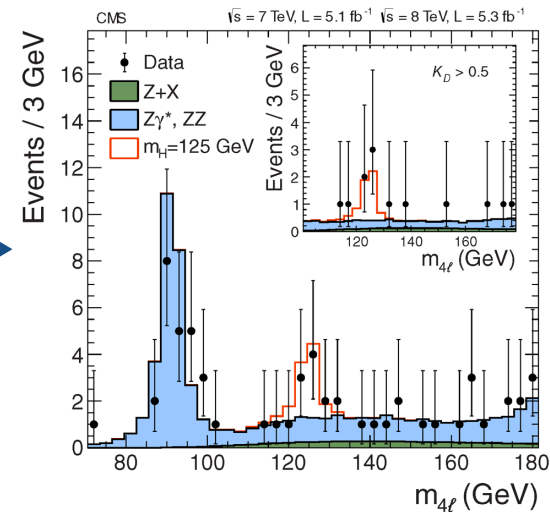
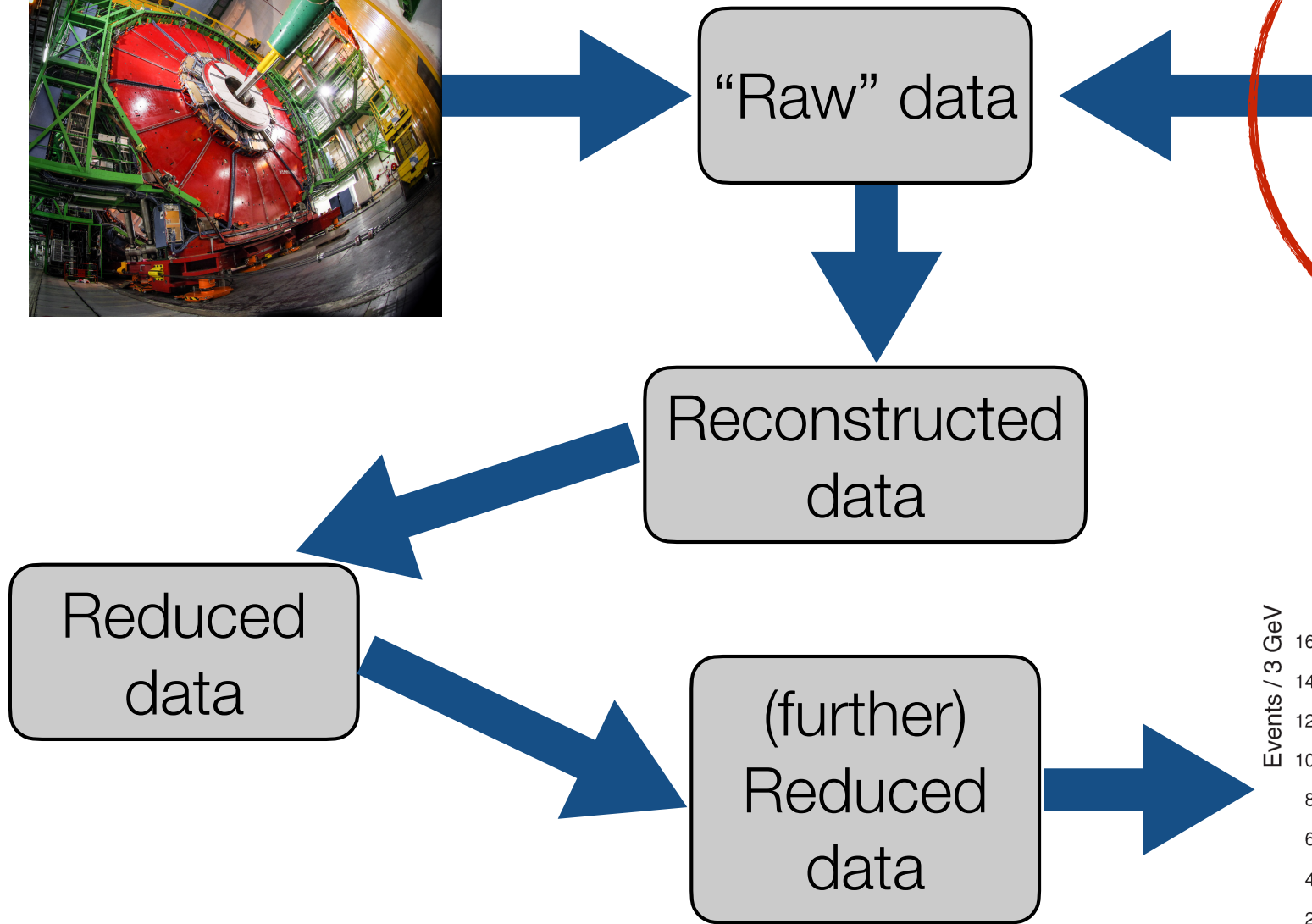
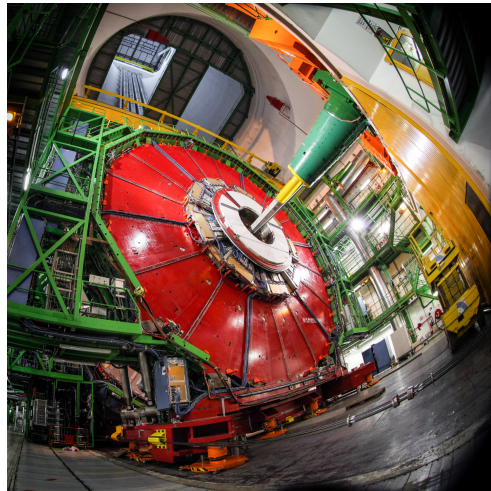
# Reconstruction: CMS



- Muon**
- Electron**
- Charged hadron (e.g. pion)**
- - - Neutral hadron (e.g. neutron)**
- - - Photon**

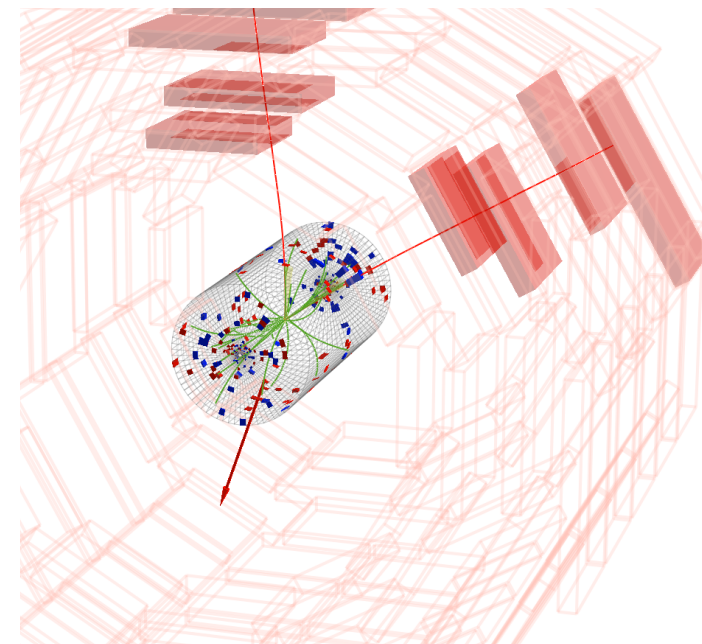
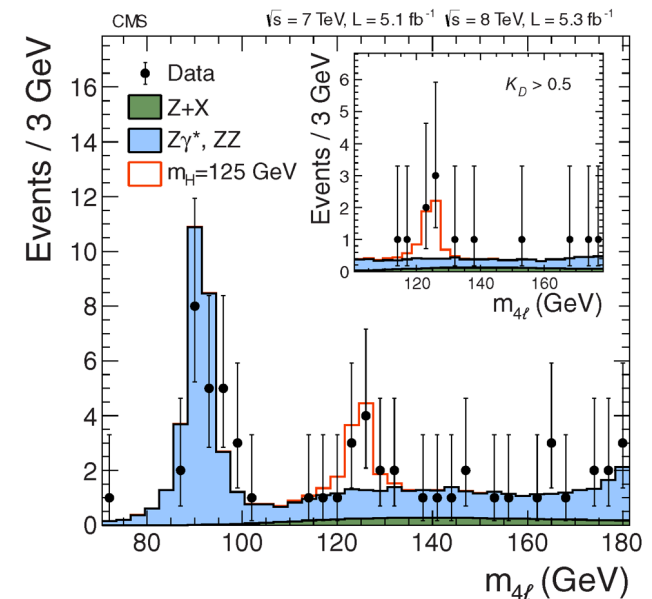


# Simulation



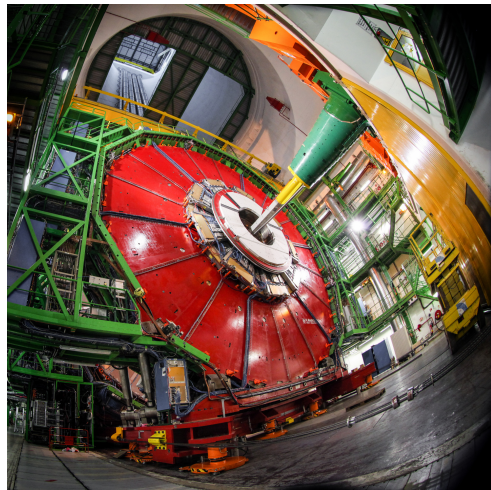
# Simulation

- **Simulated events** are crucial to science at HEP experiments
  - Rely on **Monte Carlo Simulation** to produce these events
- **Event generators** simulate the underlying particle interaction of interest
- Resulting interaction event is then fed to a **detector simulator**
  - Consider the **material** and **geometry** of every part of the detector
  - Simulate how particles from interaction and decay would propagate
  - Most detector simulations use **GEANT**
    - Also used in nuclear and accelerator physics as well as medical and space science

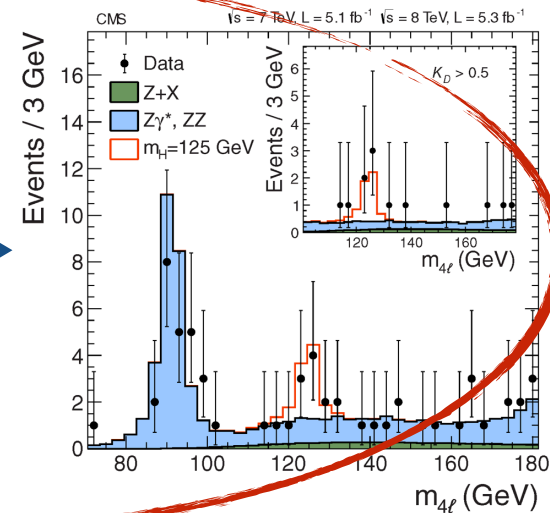
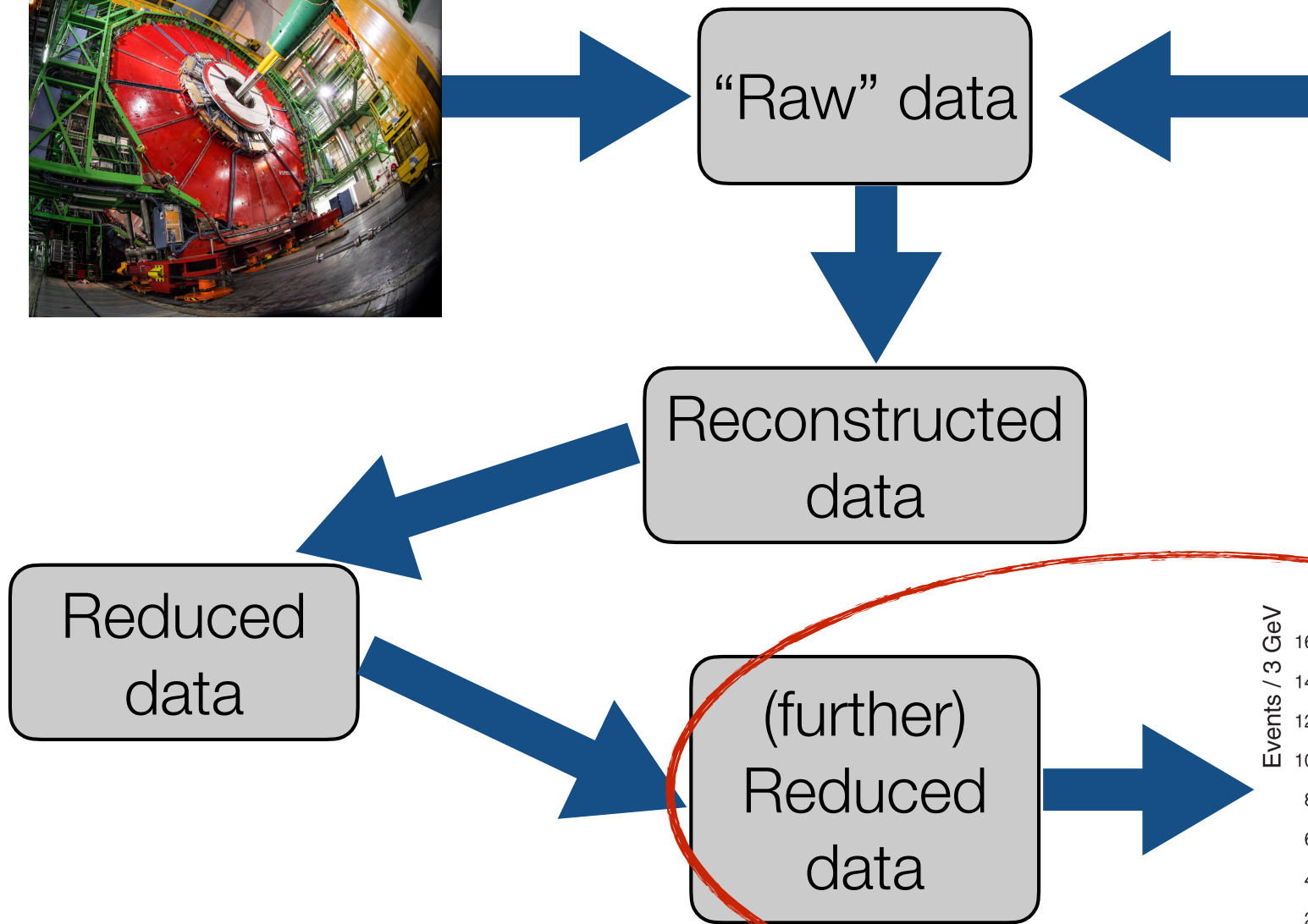




# Analysis

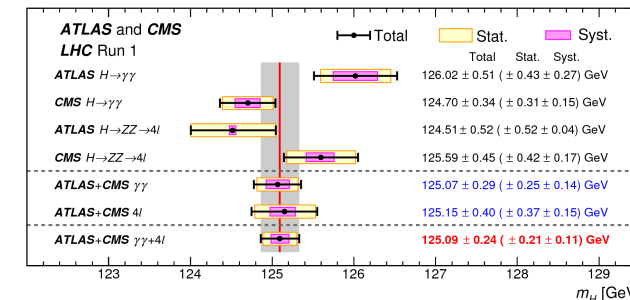
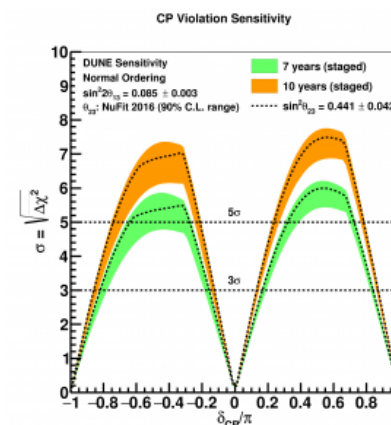
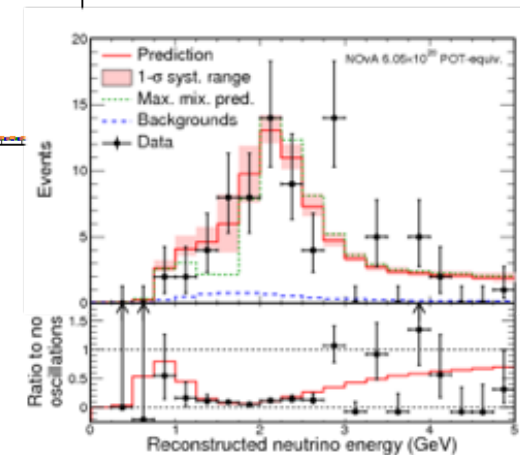
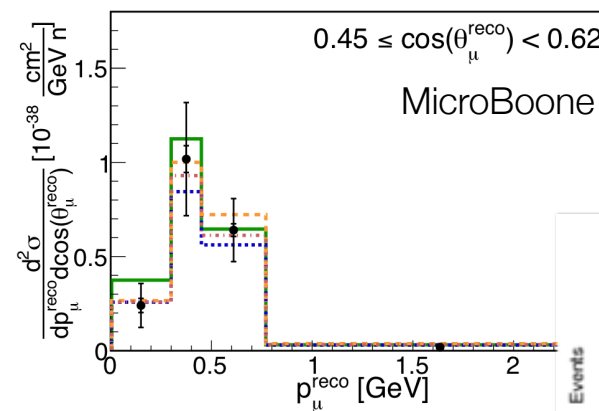


Monte Carlo simulation



# Analysis

- Getting from data events of interest to plots, tables, and numbers
  - This is the computing step nearly all HEP experimentalists are familiar with
- Common tools are needed
  - Mathematical functions
  - Statistical analysis
  - Plotting/histogramming
- Nearly all HEP experiments use the **ROOT** framework
  - Developed by CERN and Fermilab
  - C++** (object oriented)
  - Couples with code written in other languages (e.g. Python)
  - <https://pos.sissa.it/093/002/pdf>





# Preservation and Portability

---

- Experiments and data lifetime
  - Can stretch over OS upgrades and hardware evolution
  - Survives developers or software projects
  - Limited upgrade possibilities during data taking
  - Limited effort at the end of lifetime
- Use of standard tools
  - C++, Python, scikit
  - CI/CD
- Capturing the whole environment
  - Containers: Docker, Apptainer
  - REANA



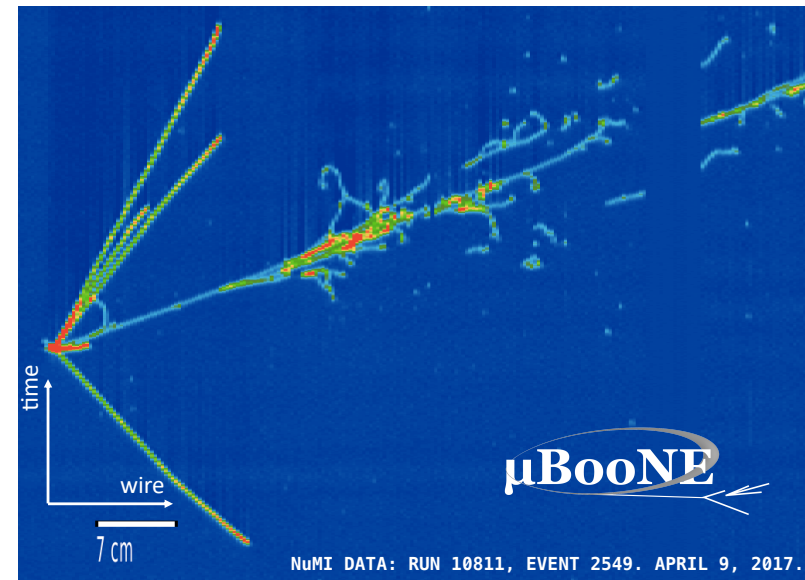
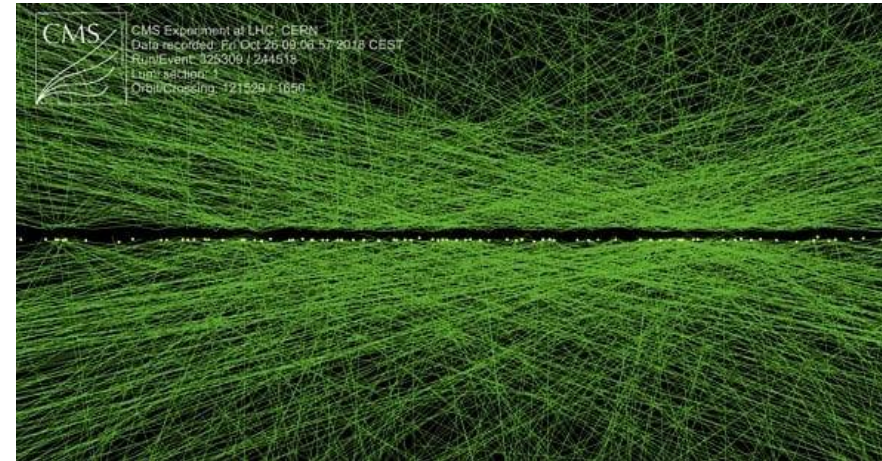
**reana**

Hardware (CPUs, Storage and Networks)



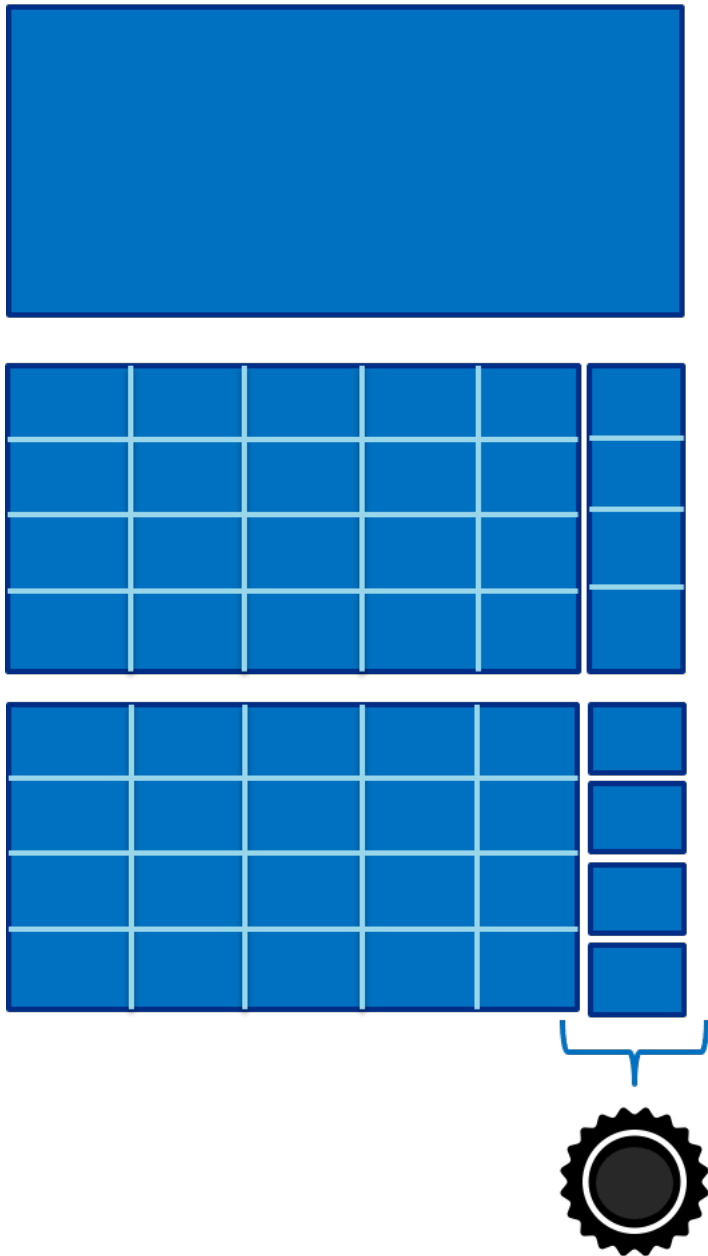
# CPU

- **Reconstruction** and **simulation** of events are biggest CPU drivers
- Such computing is “**pleasantly parallel**”
  - Processing one event is completely independent of processing any other
  - Relatively short processing times but **many events** and **growing complexity**
- **CMS** - typical collider experiment
  - ~30 s/event (~30x more in a few years!)
  - ~billions events (simulated+collision)/month
- **MicroBooNE** - liquid Argon (LAr) neutrino experiment
  - ~1-2 min/event
  - ~million events (simulated+beam)/month
  - 1 event in DUNE will have ~50x more channels (!)

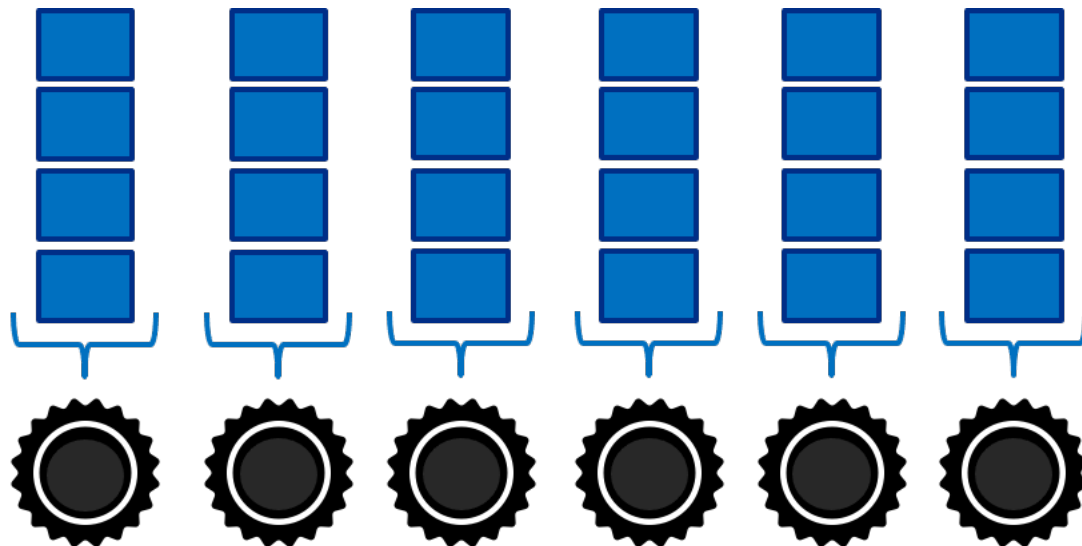
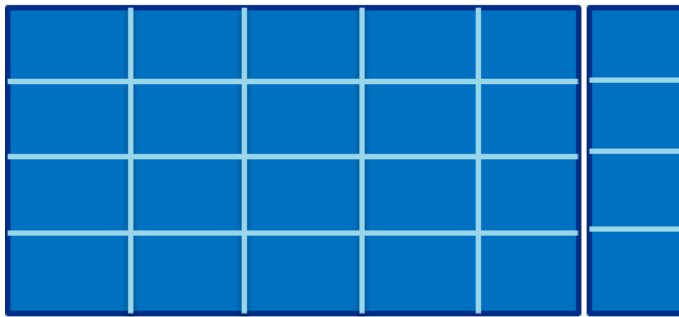


# Divide et Impera

- Split the complex problem
- Solve the parts (jobs)
- Get the overall solution



# Divide et Impera



- Split the complex problem
- Solve the parts (jobs)
- Get the overall solution



Get it fast!

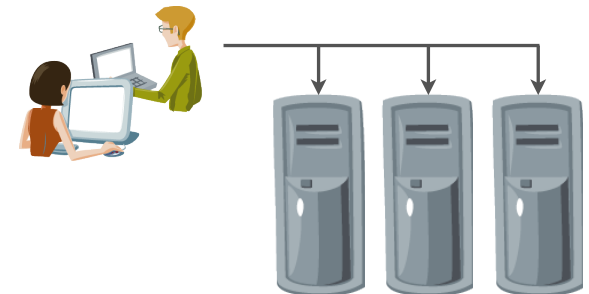
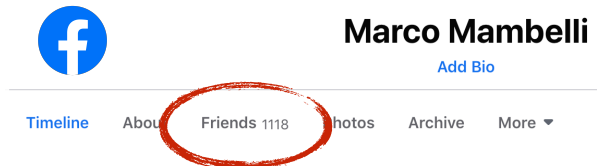
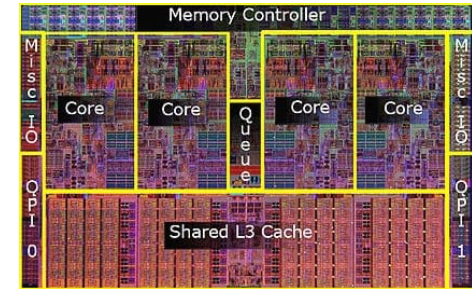
High Throughput Computing  
(HTC)



# Divide et Impera: 1 Billion Events



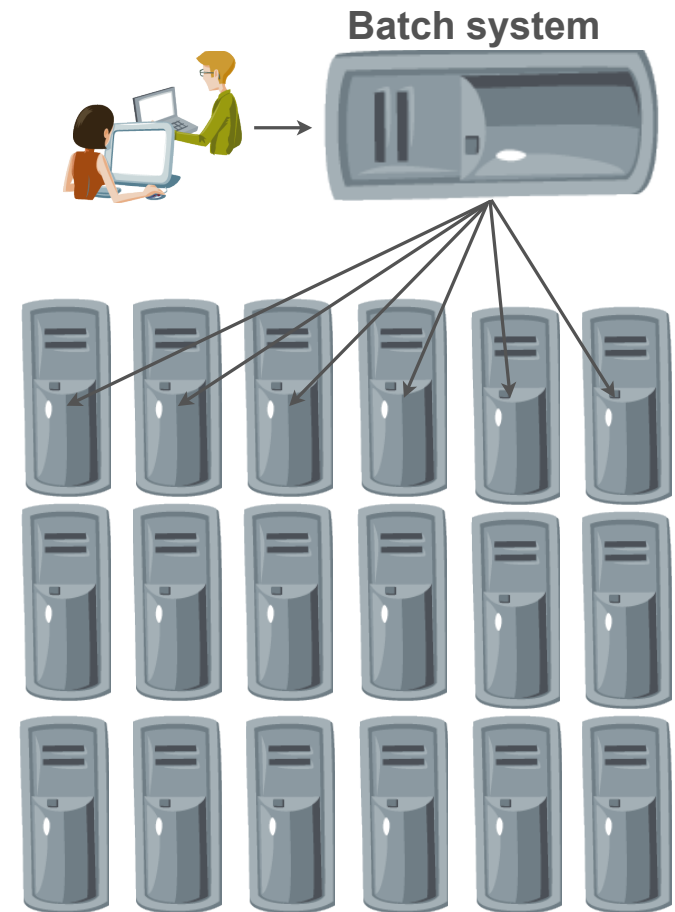
- Use one computer for the CMS events?
    - CPU: Central Processing Unit (typically whole chip)
      - Between 2 and 64 individual cores
      - Each core can process one instruction at a time
- 30s/8 cores\*1B events ~120 years*
- Use your friends' computers as well
    - To get 1B events in one month, we require **1,440 8-core computers**
      - We are almost there with the friends!
    - Your **software and data** would need to get to each of those computers
    - You'd need to **collect output** from each
    - And you'd need **user accounts** on all of them
  - Need to find an easier way to get from one computer to many!



# Divide et Impera: batch systems



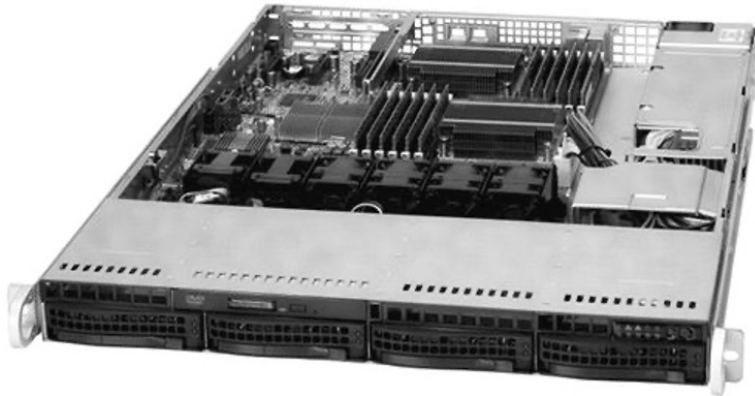
- Batch systems
  - Single entry point (queue for many users)
  - Jobs to any available slot
  - **Output** from each job handled in the same way
  - Batch system can handle user authentication on each individual computer



Some batch systems at Fermilab:  
**Fermigrid** (~20k CPU cores)  
**US CMS Tier1** (~20k CPU cores)  
**CMS LPC** (~5k CPU cores)

*30s/20,000 cores\*1B events ~17 days*

# Inside a Data Center: Computing Racks



Rack-mounted units "pizza boxes"  
dual multi-core processors, large amount  
of memory, system and small local disk  
Newest units with 64-core processors,  
256 GB of memory, 100 Gb/s network  
All running Linux!

Hot and Cold rows  
Limited by power and cooling  
GCC has a power capacity of  $\sim 2.5$   
Megawatts





# The Two Fermilab Data Centers

---



## Feynman Computing Center (FCC)

Two “logical” data centers (FCC2, FCC3)  
High Availability, each has own generator / UPS system  
Each a “corner” of a redundant network

## GRID Computing Center (GCC)

Two “logical” data centers:

- Room A, Tape Room, Network Room A
- Rooms B, C, Network Room B

“Lights out” facility, with UPS systems,  
only good for minutes

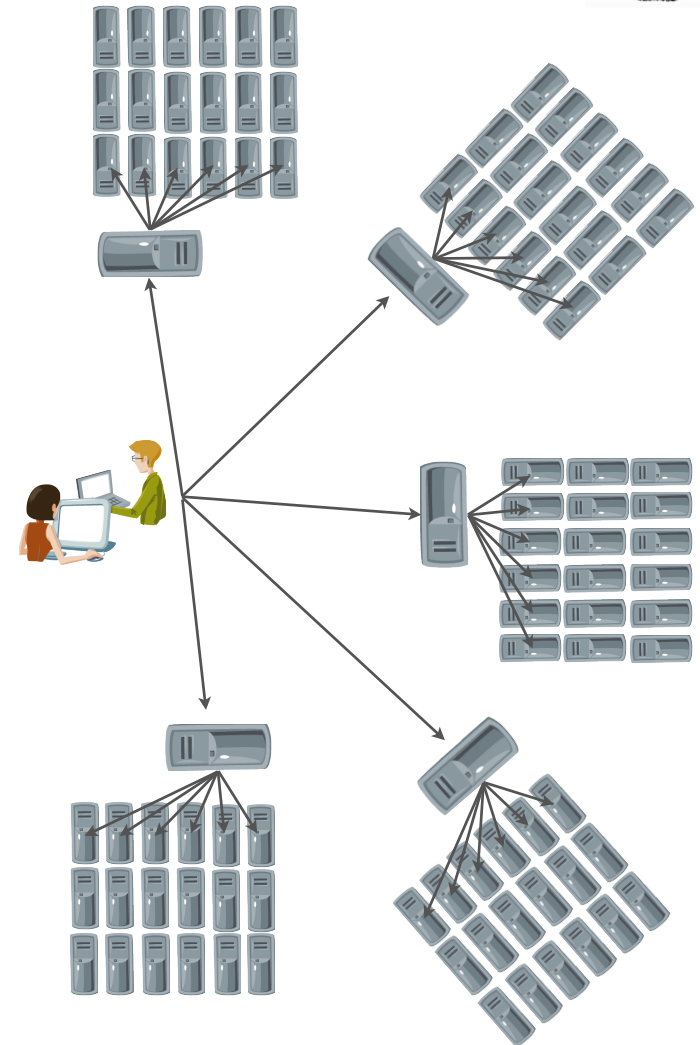
Each a “corner” of a redundant network



# Divide et Impera: the Grid



- Many batch systems accessible from one point: a computing **grid**
  - **grid sites** (batch systems) from universities and labs across the world into one grid
  - **Distributed high-throughput computing (DHTC)**
- Analogy: utility grids
- Delegated and Federated Trust Model
  - Use **tokens (JWT)** or **grid certificates (x509)** and **Virtual Organization (VO)**
  - A certificate is an encrypted “signature” that verifies you belong to an organization (e.g., a collaboration like NOvA)
  - Each site decides which VOs to trust





# Overlapping grids

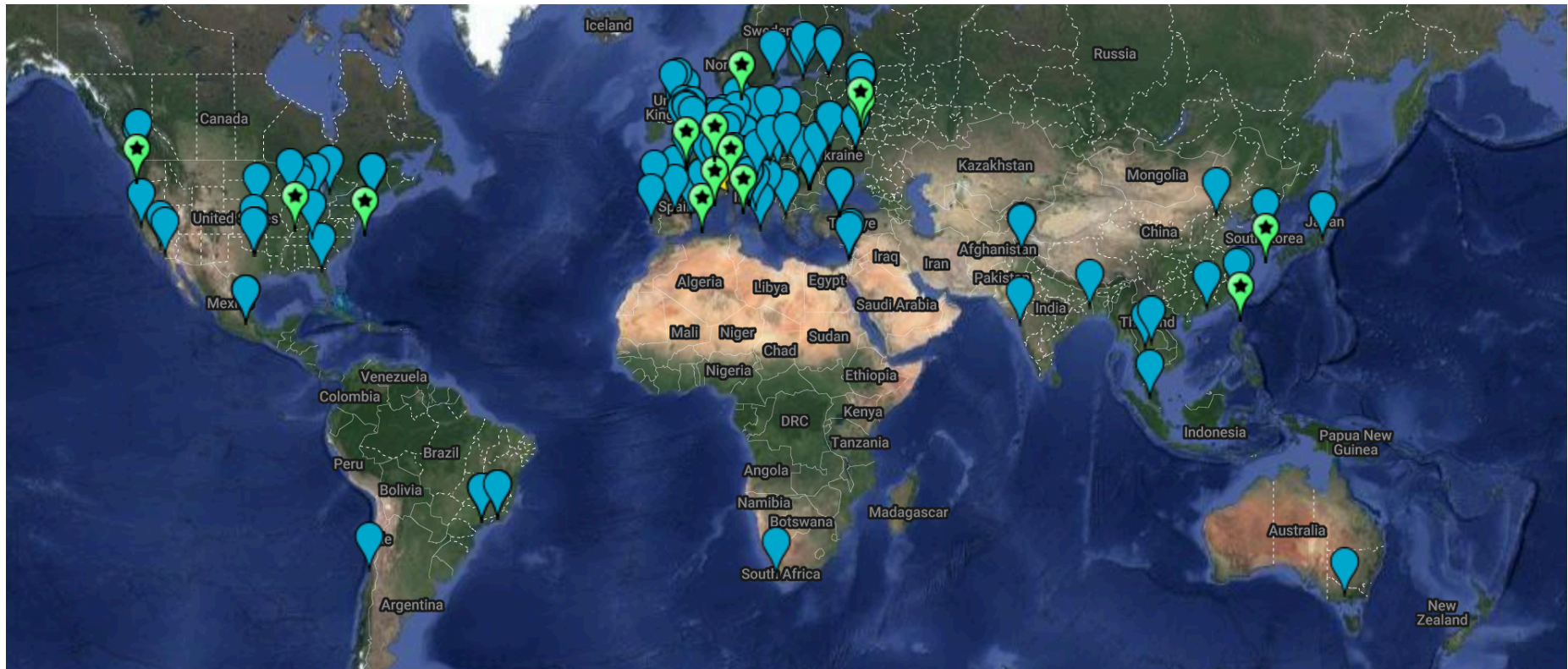




# Overlapping grids

---

## World LHC Computing Grid

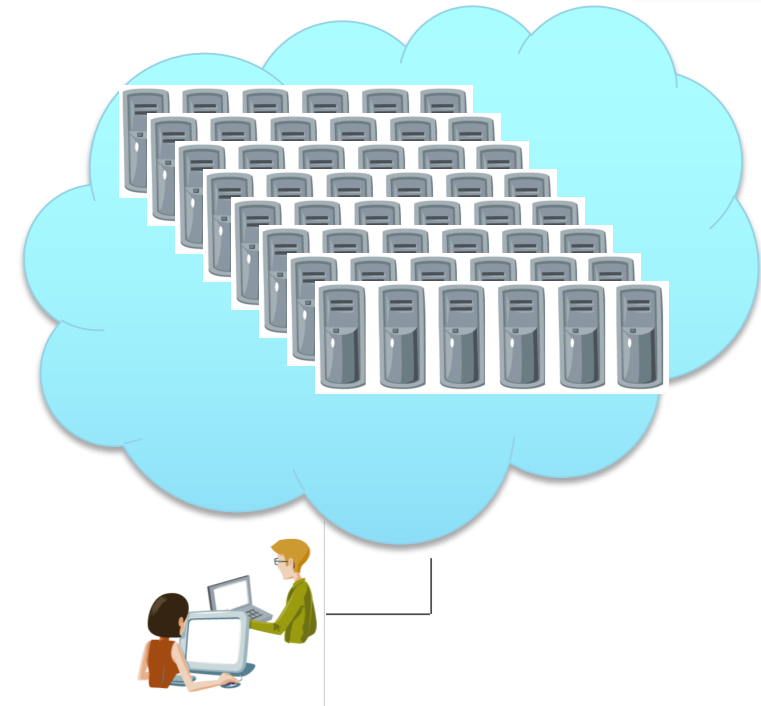


200+ sites, 2M+ CPU cores

# Divide et Impera: the Cloud



- And when you still don't have enough, then you can rent it
  - **Commercial Clouds** like AWS (Amazon), GCE (Google) and Azure (Microsoft) can rent you a seemingly endless amount of computing power
  - **Elastic computing:** expands at will
- Problem: irregular use pattern
- Solution: **burst-out** by **renting** resources on the Cloud for peak usage
- More expensive than local resources
- Difficult to justify non-capital expenditures

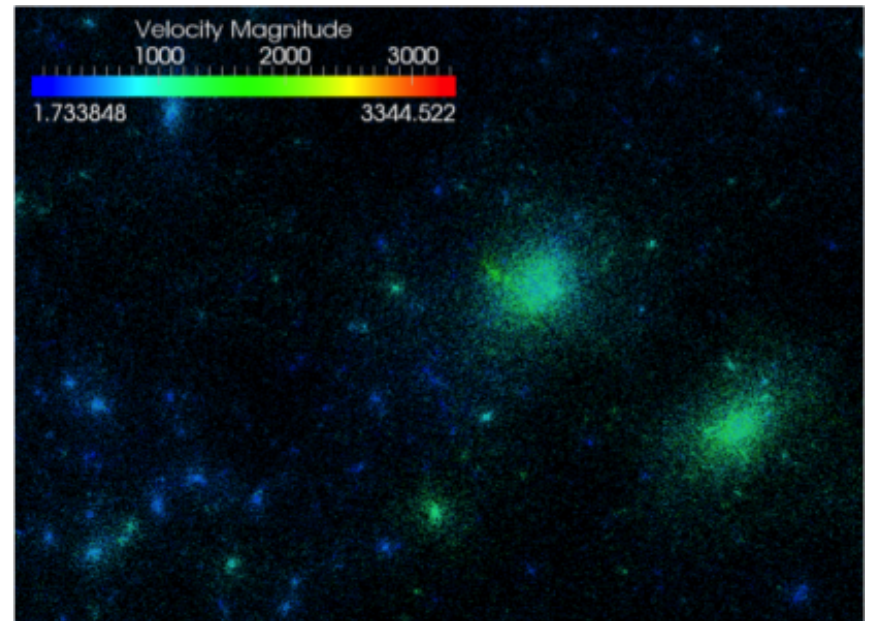


# High Performance Computing

- Much scientific computing outside of experimental HEP is not “pleasantly parallel”
- Better platform: **High Performance Computing (HPC)**
  - Batch systems where individual computers are **interconnected** via high-speed links
  - Large HPC systems are often called “supercomputers”
- Fermilab has 5 HPC clusters
  - Total of 18.5k CPU cores
  - Used for Lattice QCD calculations, accelerator modeling, large-scale astrophysical simulations and testing
- Used also for event processing by splitting it into pieces



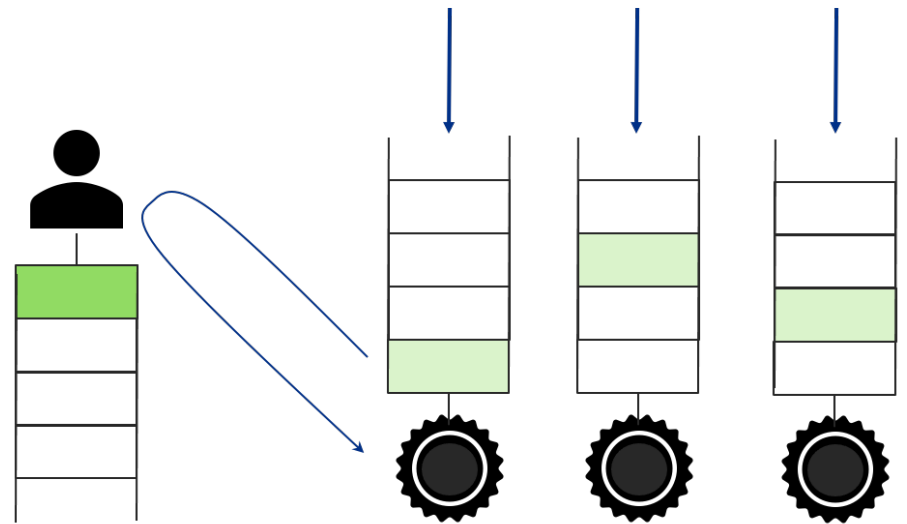
Aurora@Argonne: 20k CPUs 60k GPUs, 10PB memory, 2 DP exaflops



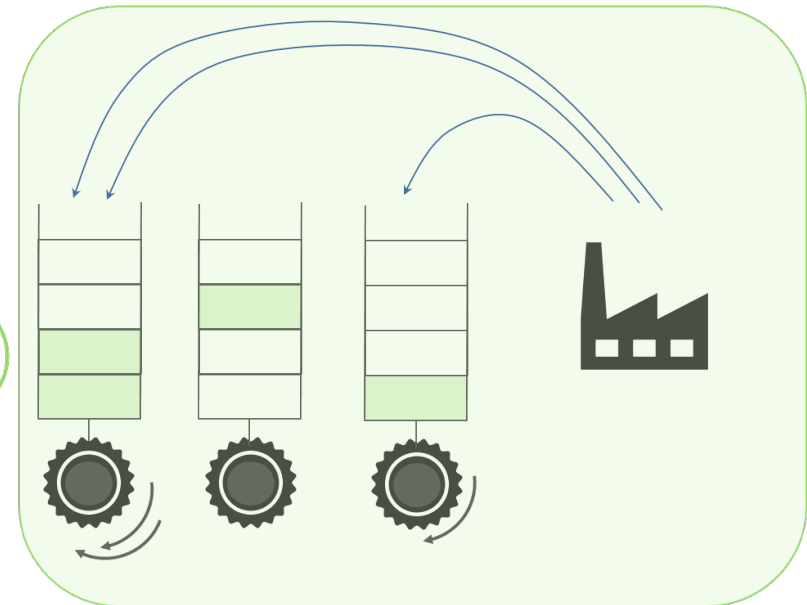


# Simplify access: Pilot jobs

- Late binding
- Separation of tasks
  - Pilot jobs
    - Test
    - Set up
    - Wait in the queue
  - User jobs
    - Science

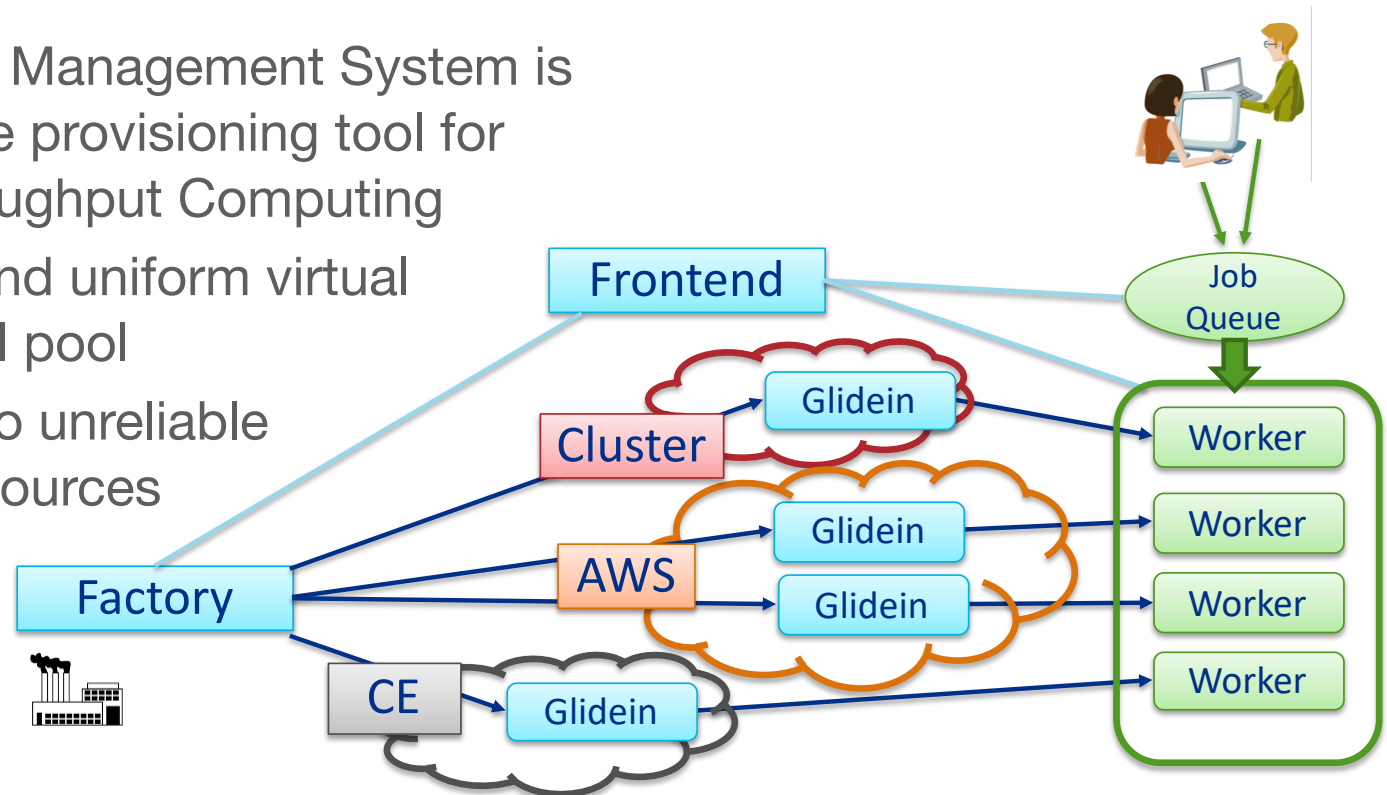


- A Factory will submit pilot jobs for you
  - Use resources as available (pressure)
  - Separation of knowledge

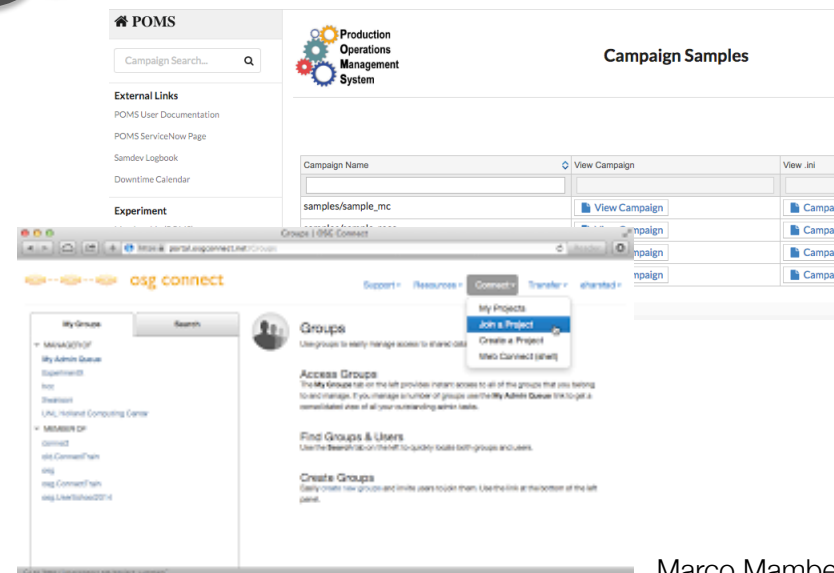


# Putting it all together: GlideinWMS

- The Glidein Workload Management System is a pilot based resource provisioning tool for Distributed High Throughput Computing
  - Provides reliable and uniform virtual clusters, the global pool
  - Submits Glideins to unreliable heterogeneous resources

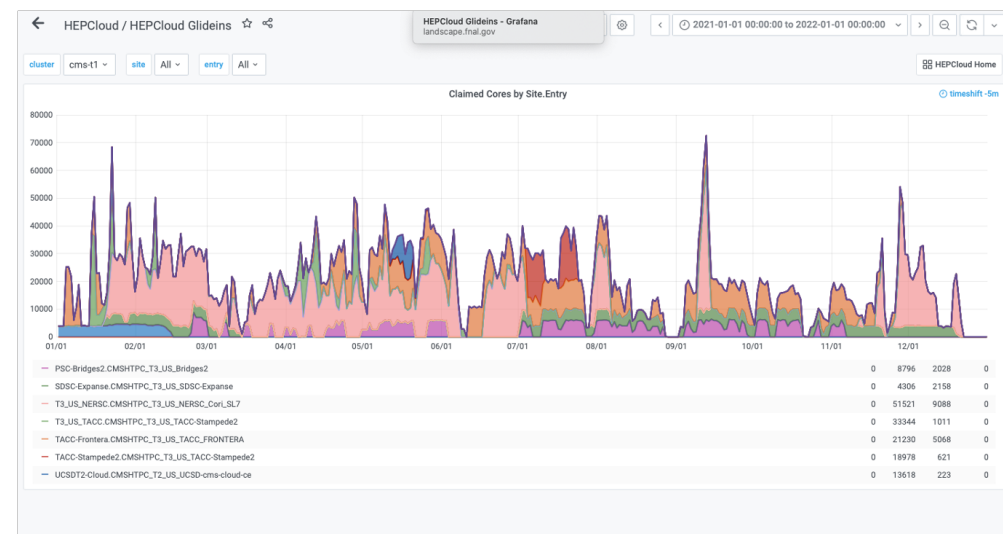
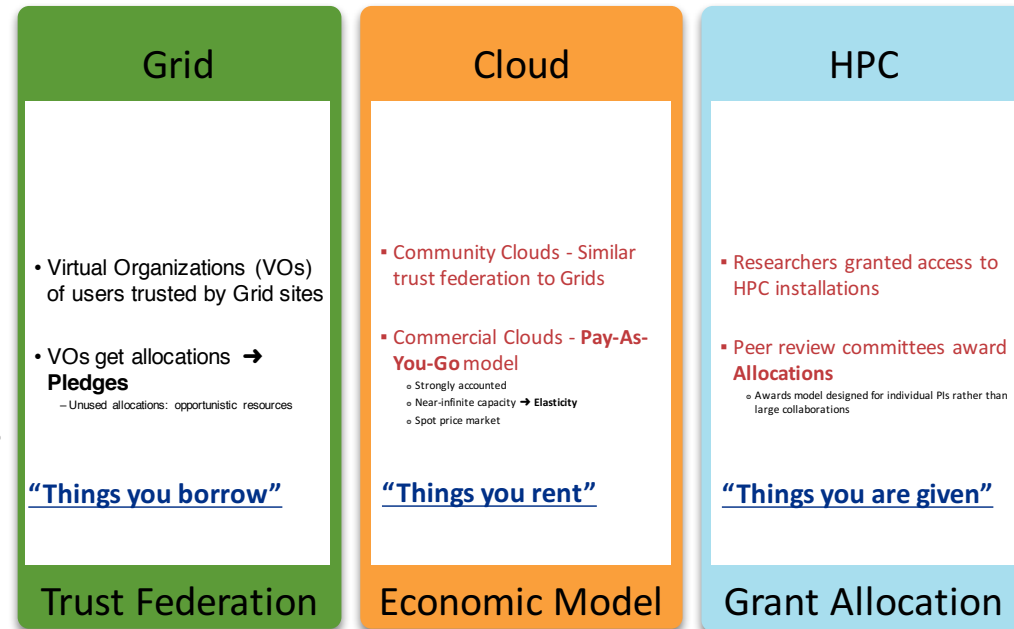


- Knows "how to talk" to all the different systems
- Multiple Frontends and Factories work together to provide High Availability
- Used by: CMS, DUNE, FIFE, OSG, (POMS, Jobsub, OSG-Connect)



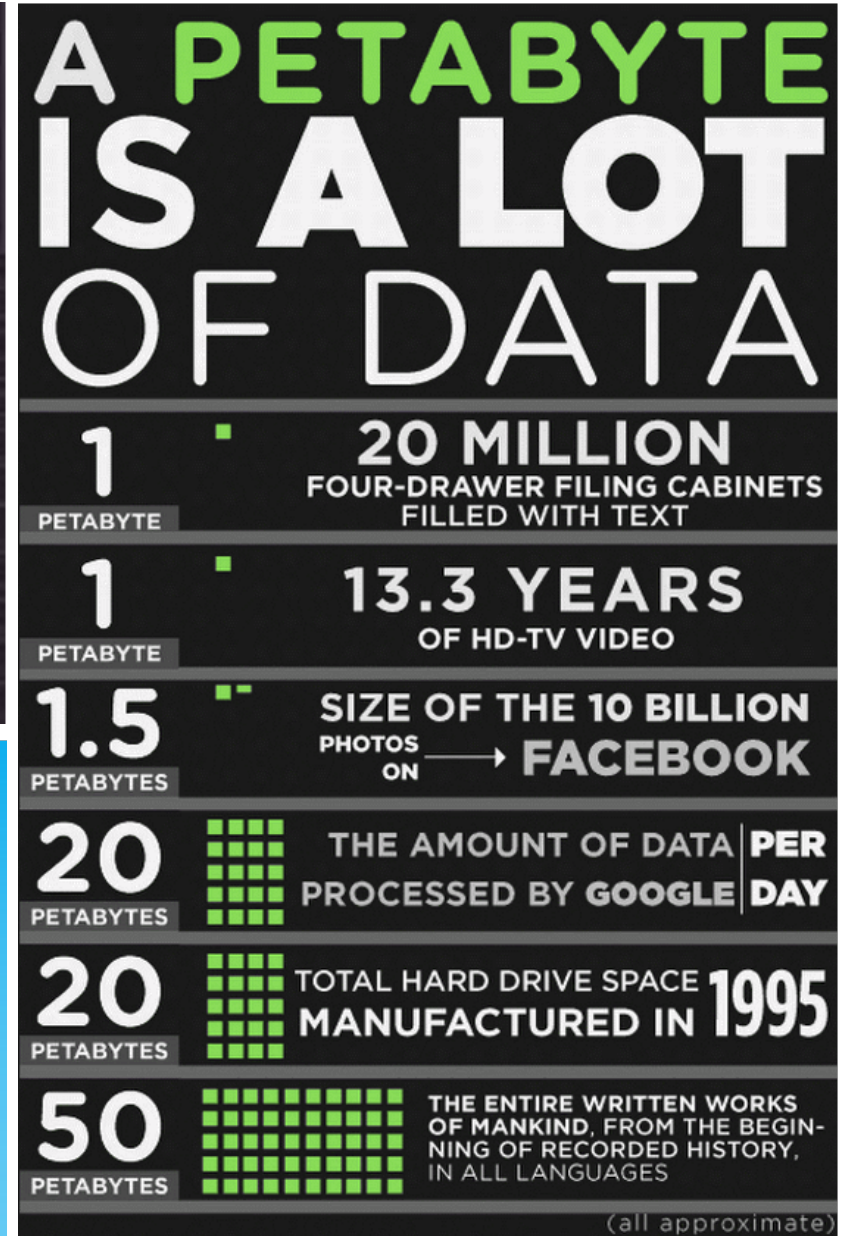
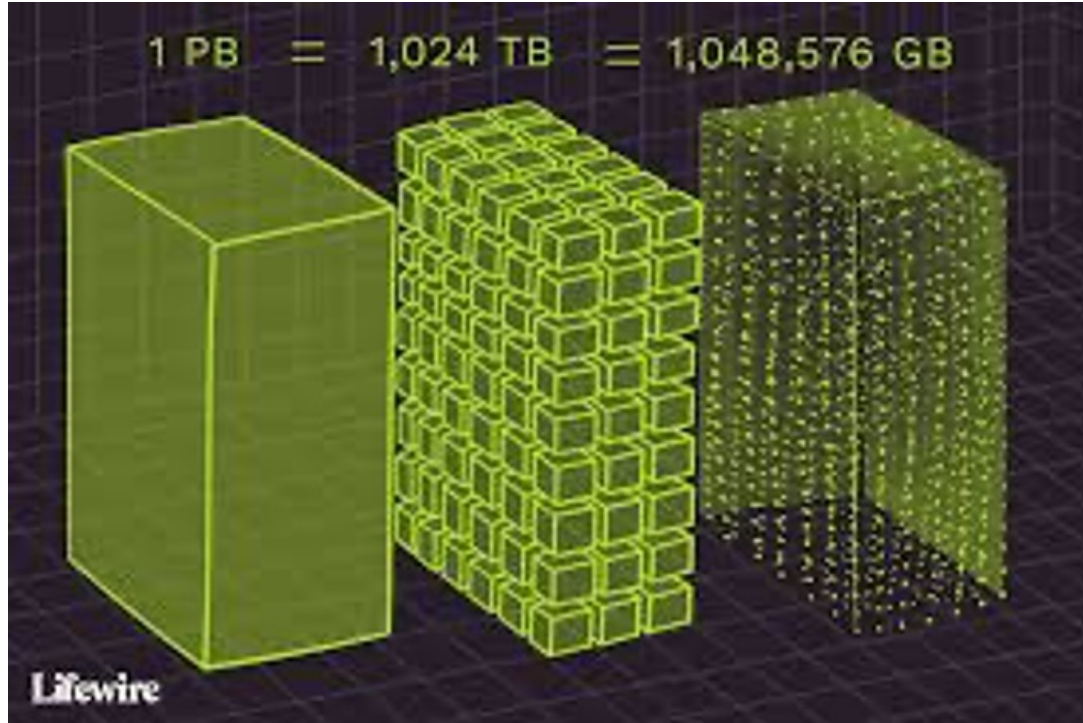
# HEPCloud

- Making it all possible from one place: **HEPCloud**
- **Unified interface** to Grid, Cloud, and HPC resources
- Currently used mainly to run CMS workflows on NERSC supercomputers
- Optimized bidding for AWS spot pricing
- Better handling of heterogeneous resources (GPUs, QPUs)
- In 2021 doubled CMS Tier 1 capacity using NERSC and other facilities, 160M CoreHrs
- Simulated 1 billion events in 48 hours





# Storage: understanding units





# Storage: understanding units



*Scientific data stored at Fermilab = 380 PB!*

It would take over



**20** YEARS TO TWEET 150 PB

if everyone in the world tweeted once per day.

## A PETABYTE IS A LOT OF DATA

1 MILLION FOUR-DRAWER FILING CABINETS FILLED WITH TEXT

**1** PETABYTE = 13.3 YEARS OF HD-TV VIDEO

**1.5** PETABYTES = SIZE OF THE 10 BILLION PHOTOS ON FACEBOOK

**20** PETABYTES = THE AMOUNT OF DATA PROCESSED BY GOOGLE PER DAY

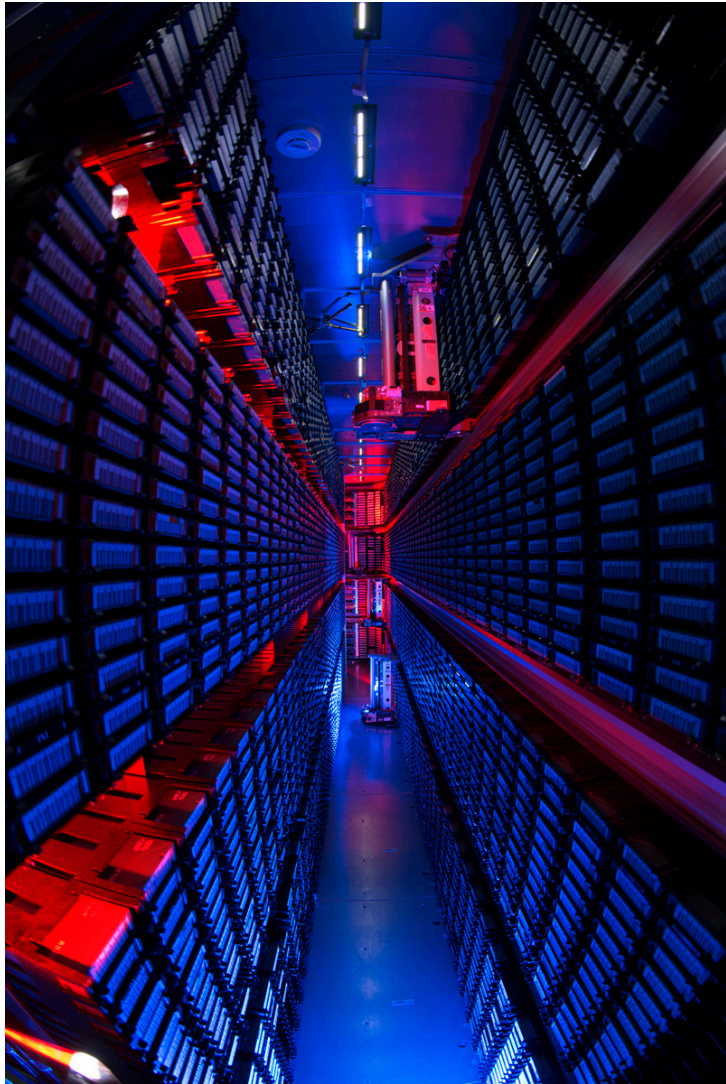
**20** PETABYTES = TOTAL HARD DRIVE SPACE MANUFACTURED IN 1995

**50** PETABYTES = THE ENTIRE WRITTEN WORKS OF MANKIND, FROM THE BEGINNING OF RECORDED HISTORY, IN ALL LANGUAGES

(all approximate)

# Storage: tape

---



- Primary storage medium for scientific data at Fermilab: **magnetic tape**
- Still the most efficient way to store petabytes of data if:
  - Not all of it is accessed at the same time
  - Access patterns are fairly linear
  - Sufficient disk for **staging**
- Fermilab has seven **robotic tape libraries**
  - Each library can hold up to 10,000 tapes
- Current tapes hold ~12TB of data each (1km of tape!)
  
- Total active on tape: 300 PB
  - CMS, 89.69
  - NOVA, 53.65
  - uBoone, 28.48
  - gm2, 17.51
  - DUNE, 15.06



# Storage: tape libraries

- 9000 to 10000 tapes per library
- A few dozen drives
- Fire suppression system



<https://drive.google.com/open?id=1caM9JoUONtflgqxcJBBN5OImGrpnYy5>

# Storage: disk

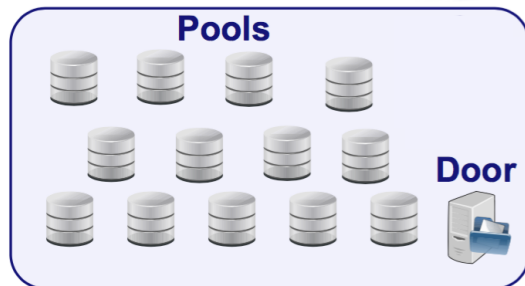
Harddrive



Diskpool



Storage System

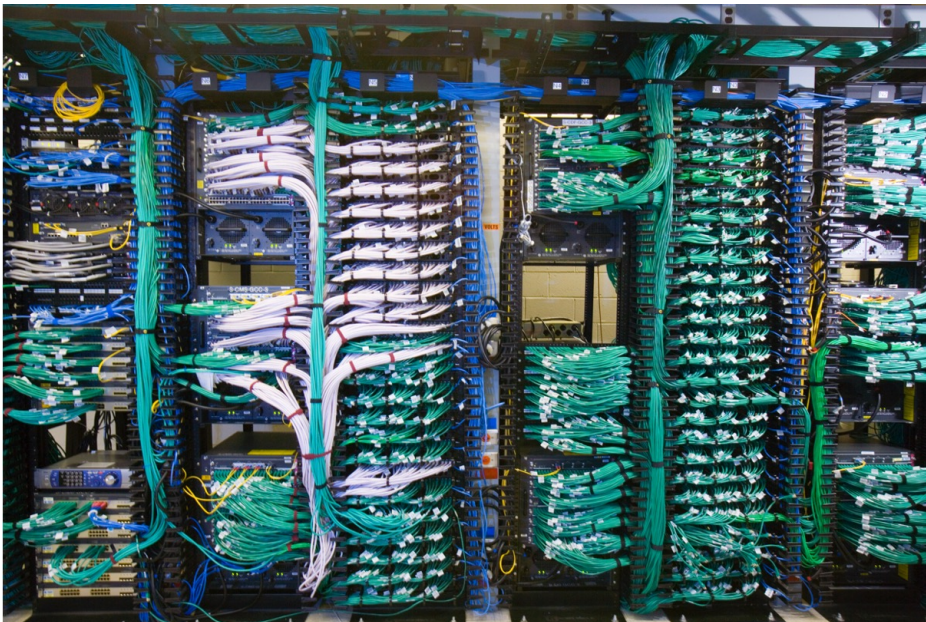
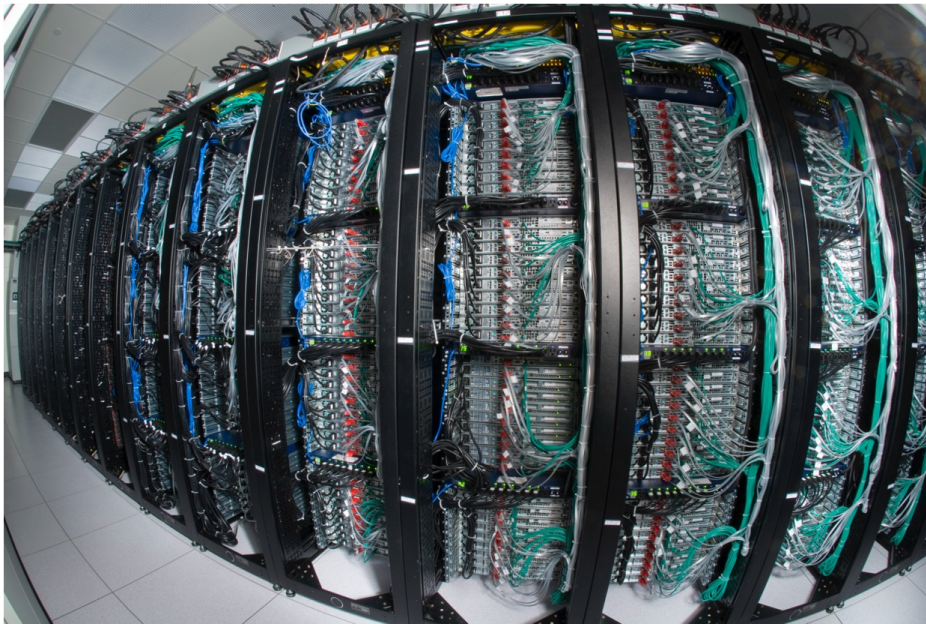


- ~80PB of disk (hard drive) storage
  - Most used as staging area from tape
- Disks are organized into **pools**
- Software allows collections of pools to appear to a user as a **single storage device**
  - Fermilab uses a system called “dCache”
  - In a typical week, data throughput in the Fermilab dCache pools average **30GB/s**





# Networking: Local Networks

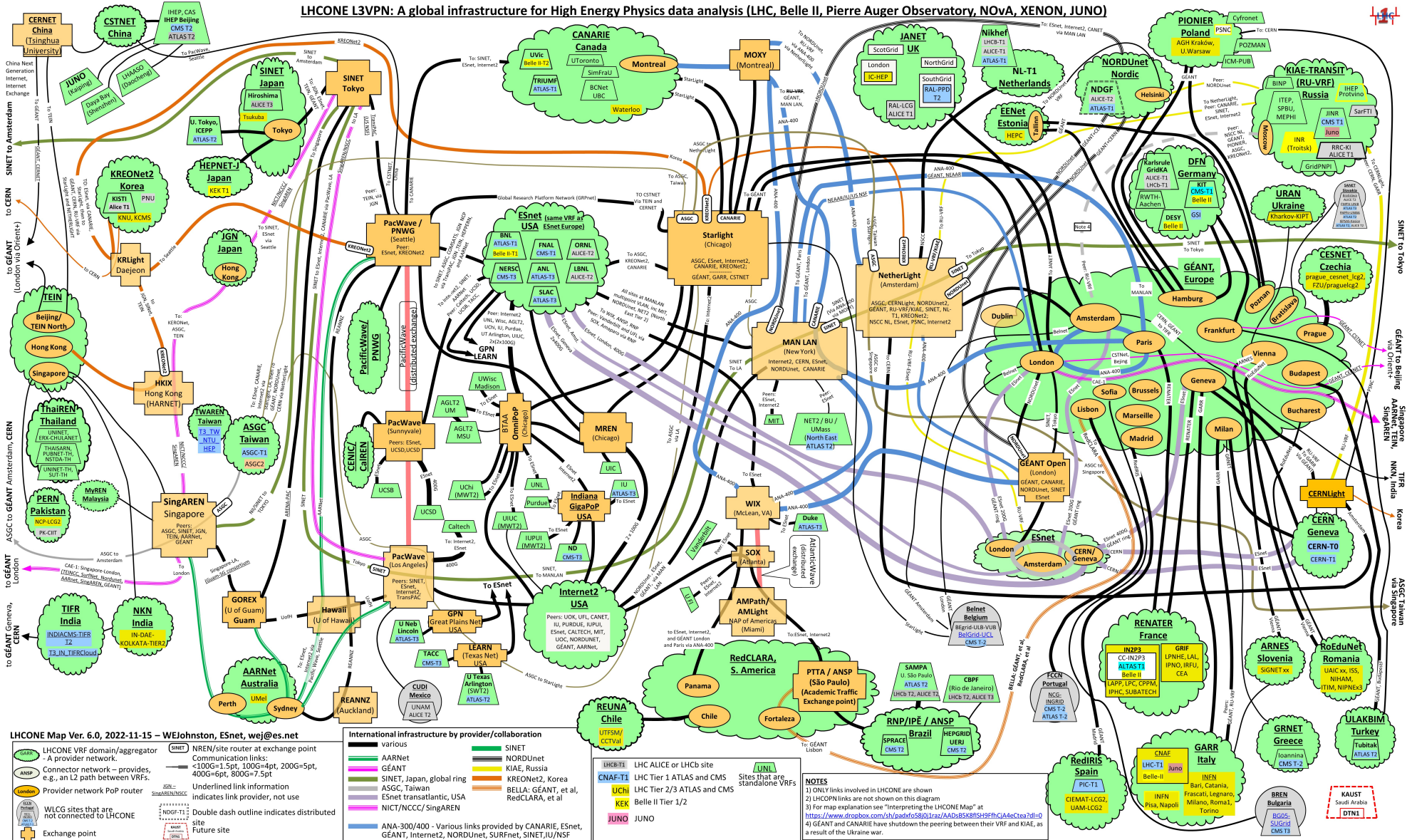






# Networking

LHCONE L3VPN: A global infrastructure for High Energy Physics data analysis (LHC, Belle II, Pierre Auger Observatory, NOvA, XENON, JUNO)



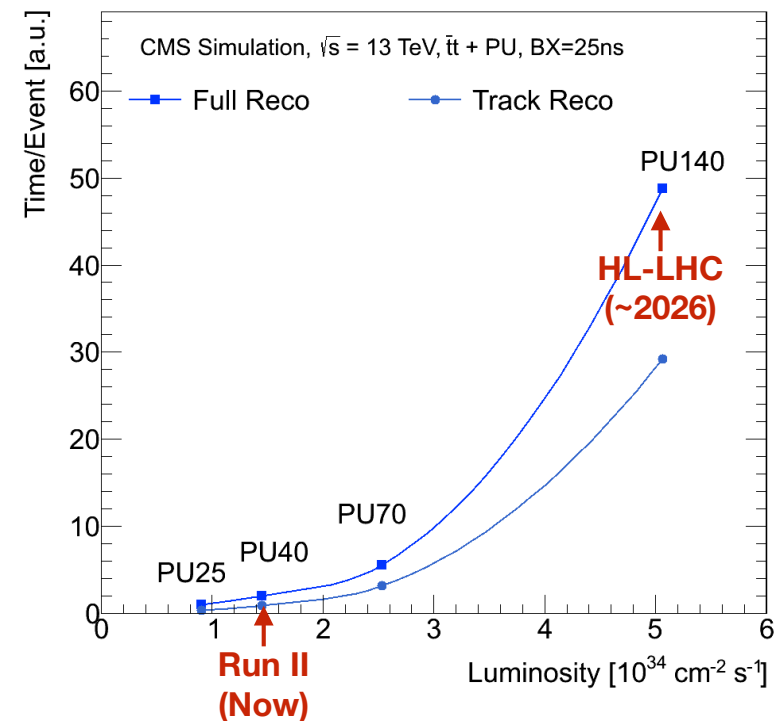
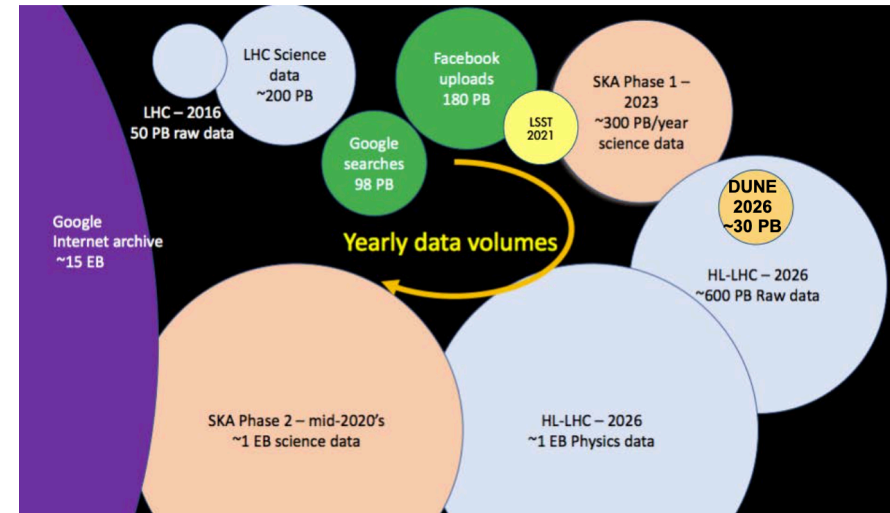
LHCONE - intermediate detail from William Johnston - <https://www.dropbox.com/sh/padxfo58j0j1raz/AADsB5K8f1SH9FfhCjA4eCtea?dl=0>

# The Future



# Explosion of data and ever-growing need for CPU

- HL-LHC, DUNE, LSST, SKA will produce up to **exabytes of data** per year
- More than one order of magnitude above current dataset sizes
- And more **complex** data to allow **precision measurements**
- Growing **dataset size** and **event complexity** = **more computing!**
  - If we scale **current algorithms**, the CPU needs of LHC experiments will grow by a factor of **30** in a decade
  - Similar issues faced by newer, bigger LAr experiments such as DUNE

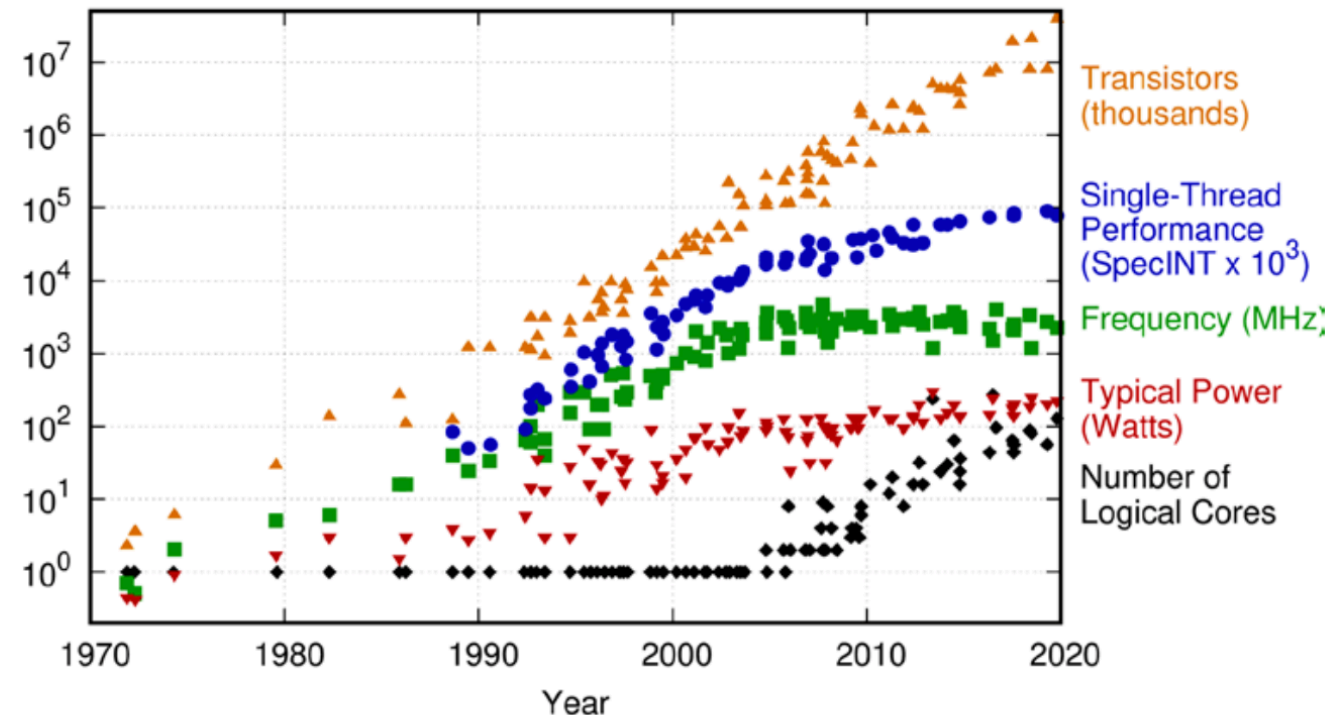


# Moore's Law probably won't help

- Moore's Law: **Number of transistors** on a chip doubles every 2 years
- That's still true, but **single-thread performance** has stopped increasing
- Instead, **number of cores** is now dramatically increasing

To take advantage of these:

48 Years of Microprocessor Trend Data



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



Intel Many Integrate Core (MIC) CPU



NVIDIA GPU

We need to rewrite a lot of software!

# What about supercomputers?

## ASCR\* Computing Upgrades At a Glance

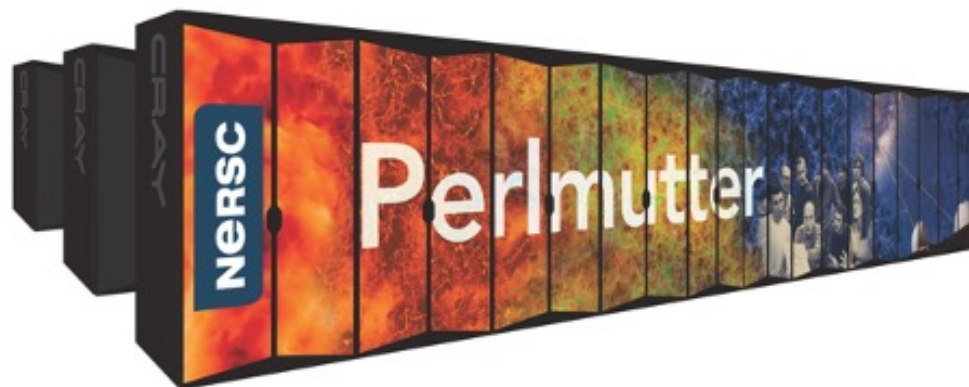
| System attributes                | NERSC Now | OLCF Now | ALCF Now | NERSC Upgrade | ALCF Upgrade | OLCF Upgrade | ALCF Upgrade |
|----------------------------------|-----------|----------|----------|---------------|--------------|--------------|--------------|
| Name                             | Cori      | Summit   | Theta    | Perlmutter    | Polaris      | Frontier     | Aurora       |
| Installation (planned or actual) | 2016      | 2018     | 2017     | 2021          | 2021         | 2021-2       | 2022-3       |
| System peak (PF)                 | 30        | 200      | 12       | >120          | 35-45        | 1500         | > 1000       |
| Peak Power (MW)                  | 3.7       | 10       | <2.1     | 6             | <2           | 29           | 60           |

This is the hard part - getting the computing power without melting the building.

PF = petaflops, floating point operations per second

1,000 PF = 1 exaflops

**“Excascale”**



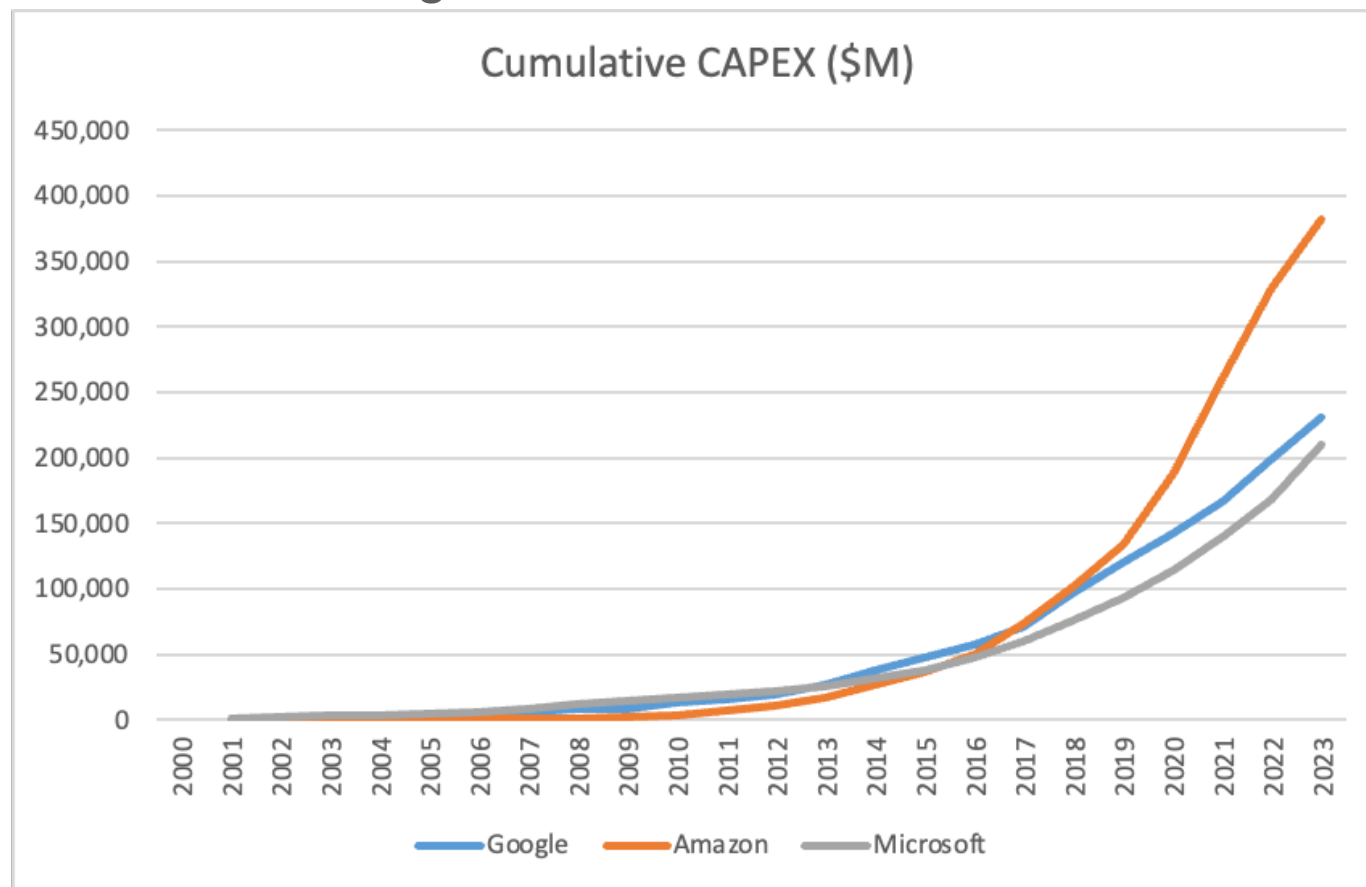
\*Advanced Scientific Computing Research  
(Dept. of Energy)

<http://exascaleproject.org/>



# Commercial cloud computing (>> HEP computing)

- Total spending on cloud computing is now close to \$700 billion per year
- Many huge companies (Netflix, for example) don't buy their own clusters but rely entirely on cloud computing
- HEP experiments are using these resources as well

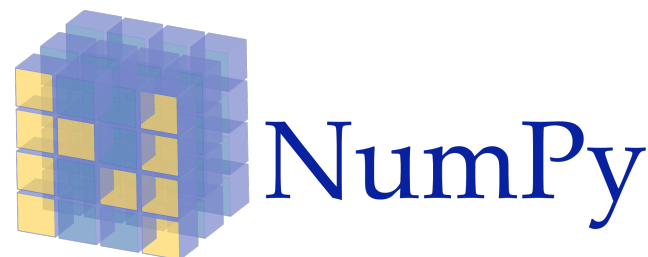


<https://platformonomics.com/2024/02/follow-the-capex-cloud-table-stakes-2023-retrospective>

# Analysis techniques: tools from industry

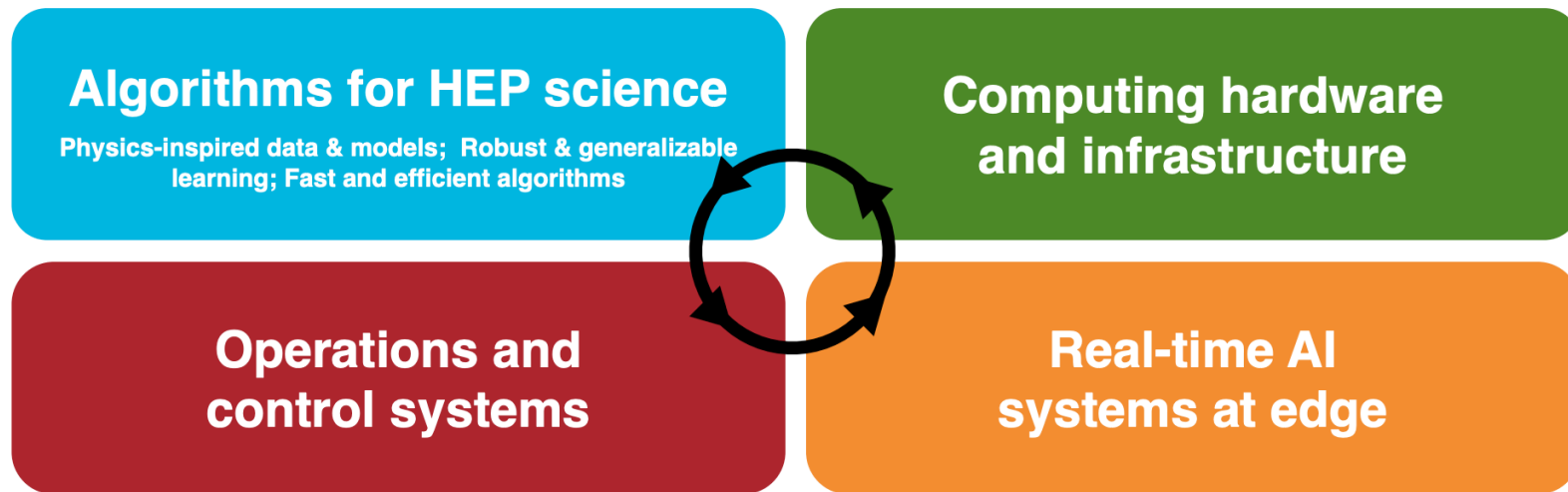
---

- HEP experiments were some of **the first** cases where people had to deal with analyzing really big datasets
  - Had to develop our own tools to get the science done (ROOT, for example)
- Not true anymore. Basically every big company you can think of has huge amounts of data at their fingertips
  - Many tools of been developed outside labs and universities to help store, process, and analyze all this data
- Fermilab's approach for CMS analysis is COFFEA (the COmpact Framework for Elaborate Algorithms)
  - Instead of a **for loop** over events, use **array programming** expressions to process many events simultaneously
  - Uses Apache Spark and tools from the scientific python “ecosystem” based on numpy



# AI at Fermilab

---



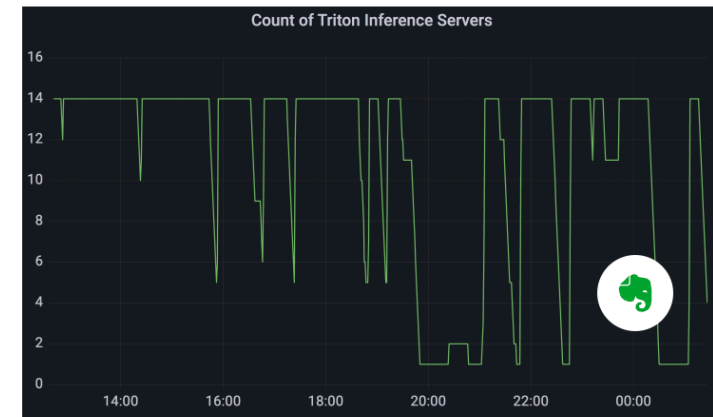
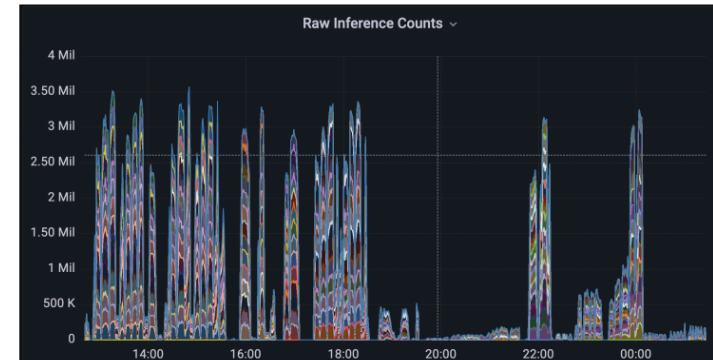
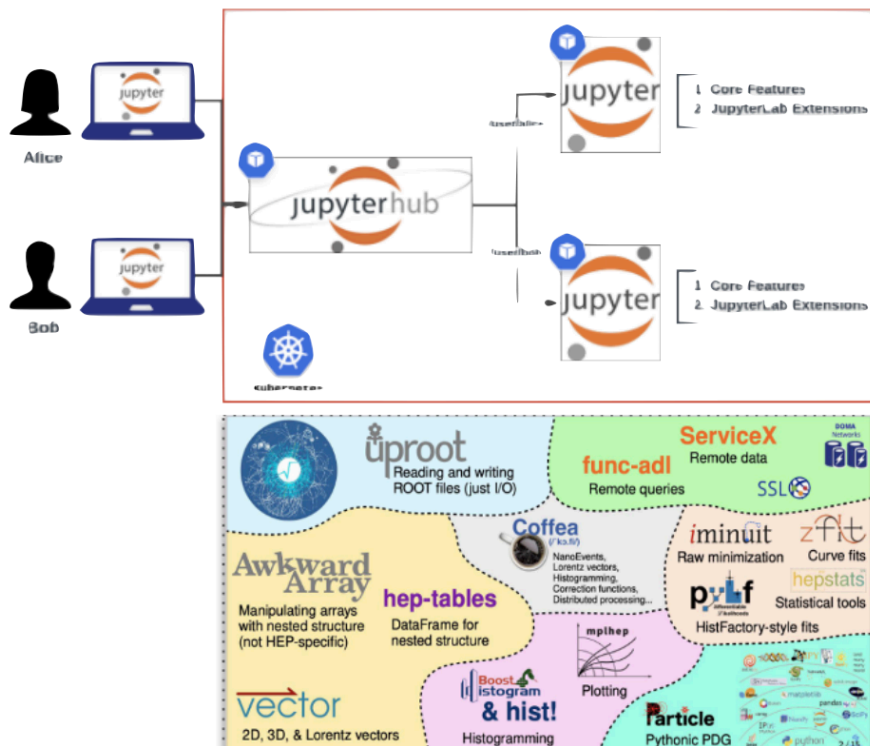
Tia Miceli - AI at Fermilab - [https://indico.fnal.gov/event/59656/contributions/269042/attachments/168498/225755/20230629\\_AI\\_at\\_Fermilab\\_TiaMiceli.pdf](https://indico.fnal.gov/event/59656/contributions/269042/attachments/168498/225755/20230629_AI_at_Fermilab_TiaMiceli.pdf)



# Computing hardware infrastructure

Flechas et al., [arXiv:2203.10161](https://arxiv.org/abs/2203.10161)  
 Benjamin et al., [arXiv:2203.08010](https://arxiv.org/abs/2203.08010)

- **Elastic Analysis Facility** @ Fermilab provides **resources** and **data-science standard industry tools** for AI training and inference
- Additional GPU resources available on CMS LPC, Wilson Cluster
- Capable of **bursting** to O(100k) batch computing CPU cores



Tia Miceli - AI at Fermilab - [https://indico.fnal.gov/event/59656/contributions/269042/attachments/168498/225755/20230629\\_AI\\_at\\_Fermilab\\_TiaMiceli.pdf](https://indico.fnal.gov/event/59656/contributions/269042/attachments/168498/225755/20230629_AI_at_Fermilab_TiaMiceli.pdf)

# Conclusions

---

- The complexity of HEP experiments doesn't stop with the detectors
- Scientific computing permeates every aspect of how we do physics at Fermilab
  
- There are challenges ahead
  - Many that **you** could help solve!
  
- **Thanks to all those who helped with content**
  - Especially: Bo Jayatilaka, Allison Hall, Kevin Pedro, Lorena Lobato, Dmitry Litvintsev, Sophie Berkman, Oliver Gutsche, Ken Herner, Burt Holzman, Michael Kirby, Anne Schukraft, Erica Snider, Alexander Radovic, Stuart Fuess

Questions?

<http://computing.fnal.gov>