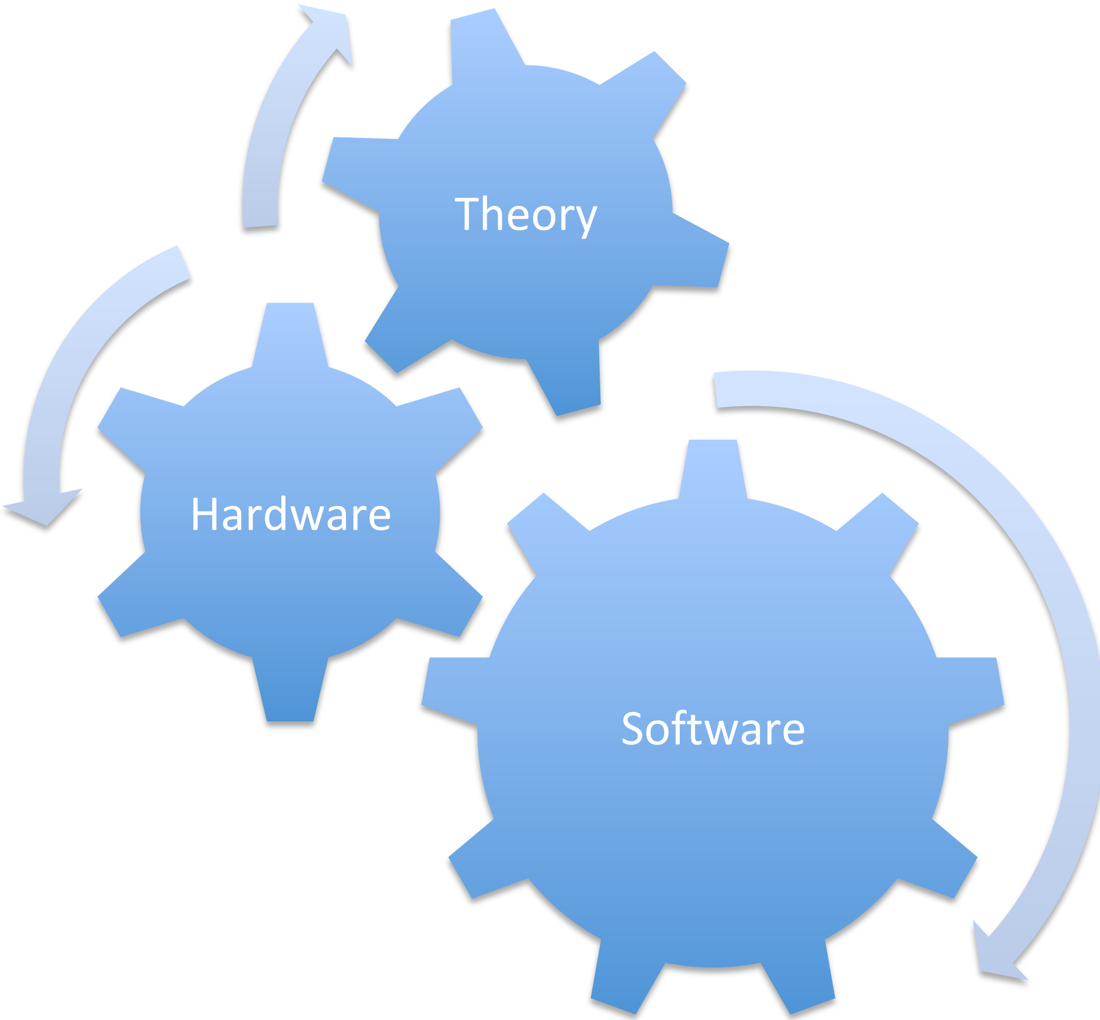
Future Software Challenges in High-Energy Physics

HEP Software Foundation

Luiza Adelina Ciucu
summarizing the DPNC seminar talk
by Graeme Stewart (CERN)

The quantities of data only keep growing in particle physics.
Software's role becomes crucial, besides **theory** & **hardware**.

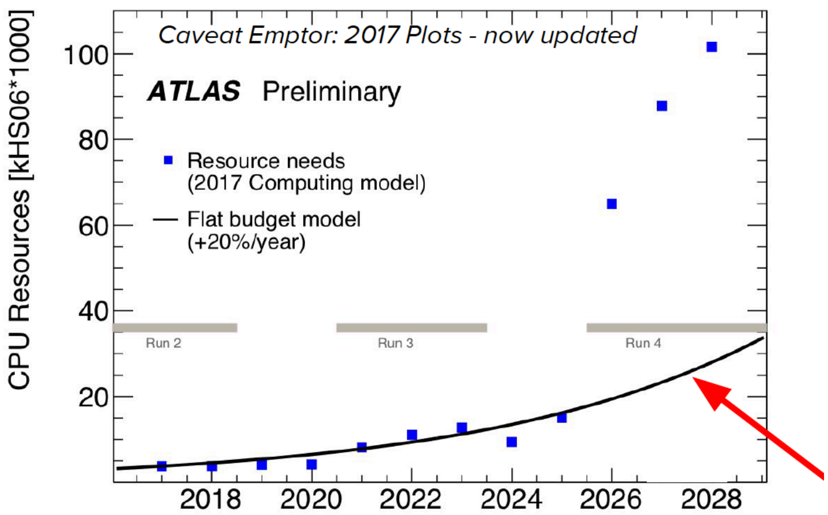


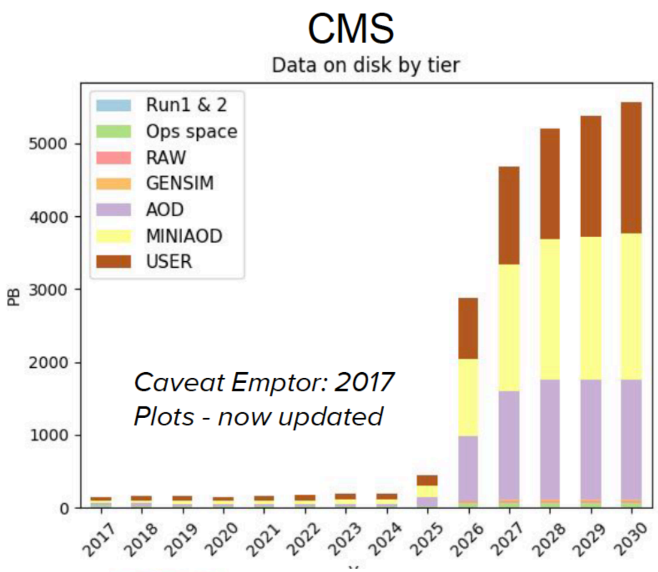
Theory:
SM, BSM
(SUSY, Dark Matter)

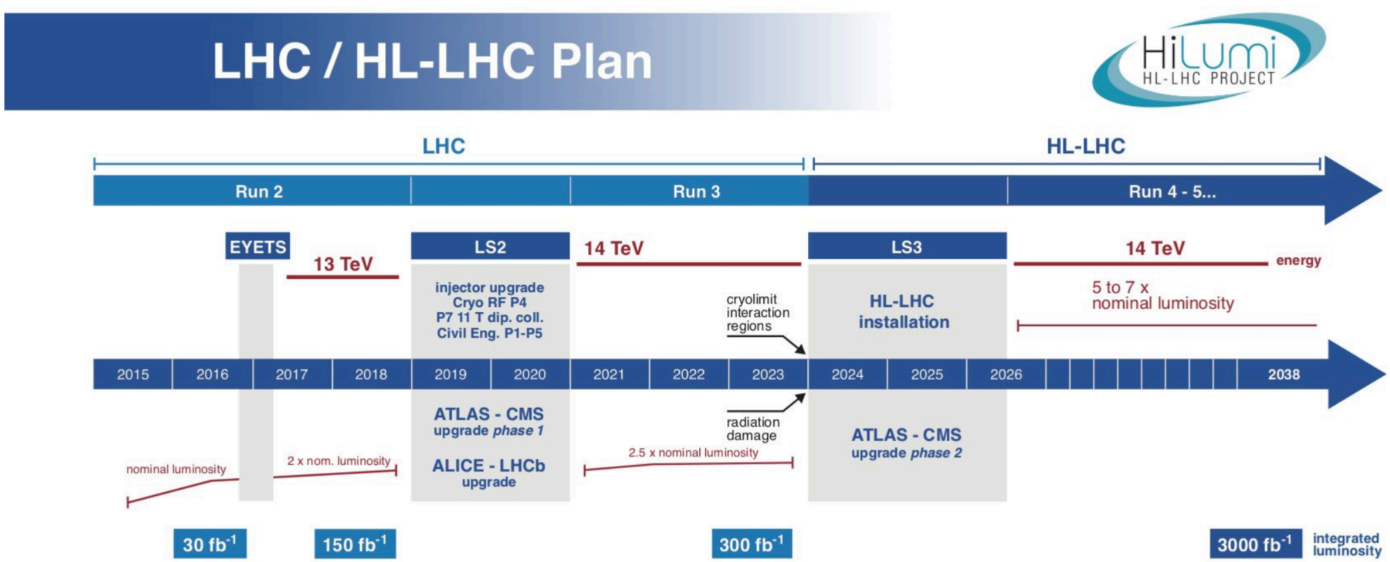
Hardware:
Detectors
(ATLAS, CMS, DUNE)

Software:
data acquisition,
triggering,
data analysis,
statistical analysis

Only seven years from now (2026) CERN's computing needs will outgrow massively the hardware resources available, when the HL-LHC will starts with the Run-4 data taking.

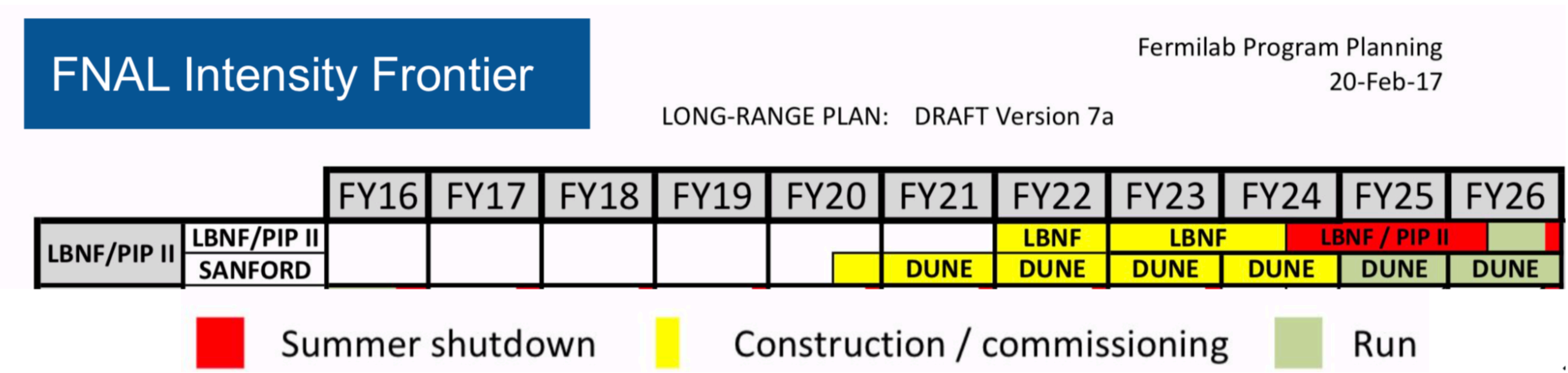




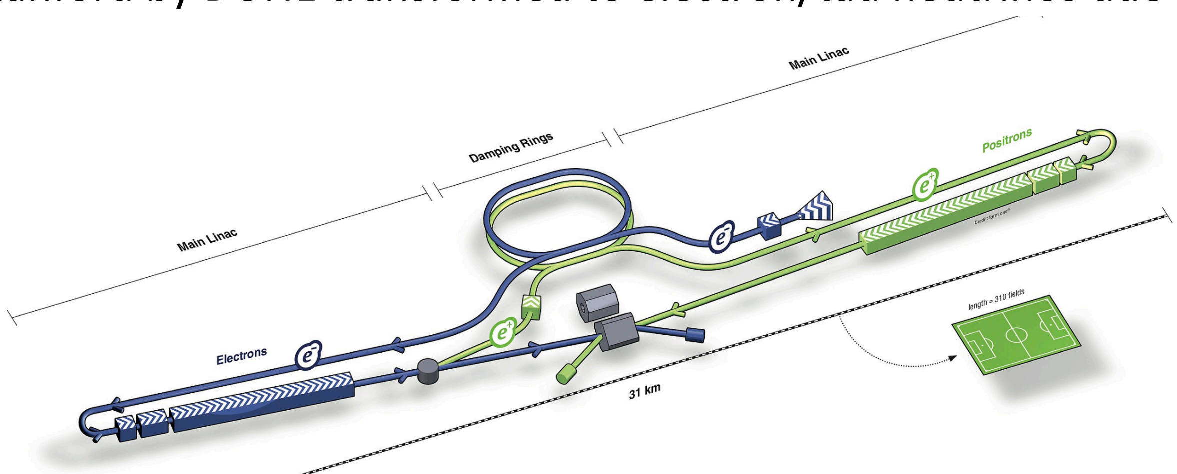


Measure precisely SM parameters.
Two Higgs bosons.
CP violation.
Dark Matter.
A surprise BSM?

Six years from now (2025) also Fermilab will face huge computing challenges with the start of the new high intensity neutrino beam experiments.



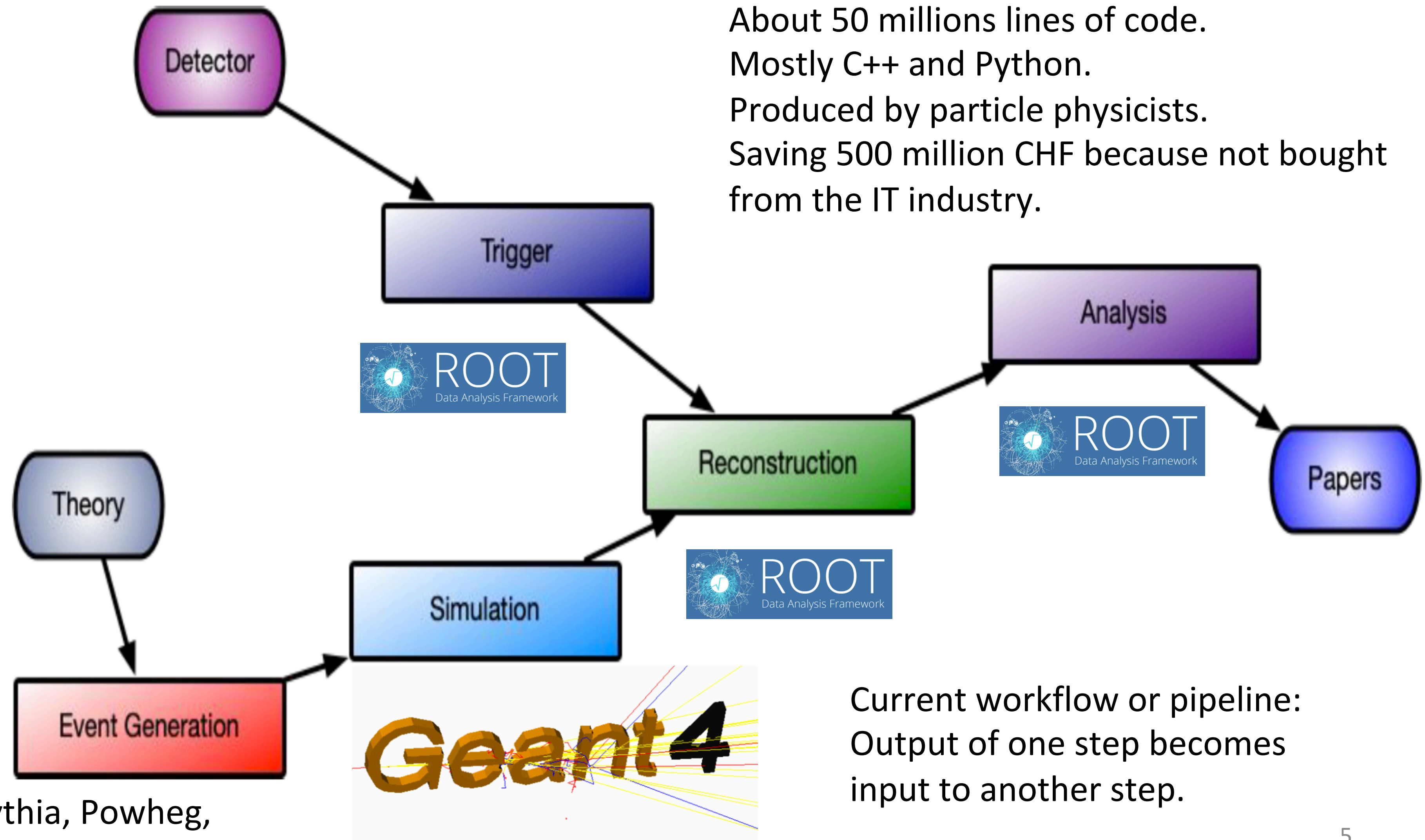
Study the nature of **neutrino mass**, via **neutrino oscillations**.
At the Long Baseline Neutrino Facility (LBNF), muon neutrinos produced at Fermilab are detected at Stanford by DUNE transformed to electron/tau neutrinos due to neutrino oscillation.



On a longer time-scale (2030-2040), there may be several accelerators with their detectors:
(ILC, left), CEPC, etc.

Computing & software are key in all steps of a particle physics data analysis, both for real data (detector -> trigger) & for theoretical predictions (event generations, simulation).

About 50 millions lines of code.
Mostly C++ and Python.
Produced by particle physicists.
Saving 500 million CHF because not bought from the IT industry.



Current workflow or pipeline:
Output of one step becomes input to another step.

Pythia, Powheg,
MadGraph, Sherpa, aMC@NLO

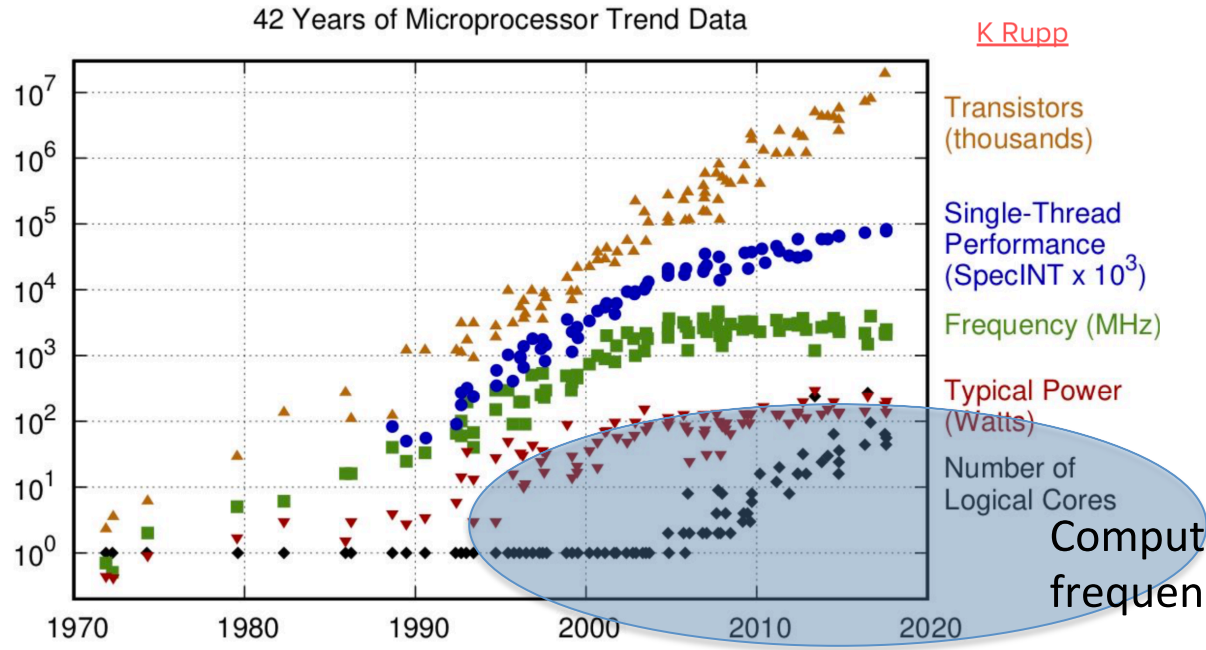
The particularity of particle physics is that each collision (event) is independent of another => parallelization & Grid.

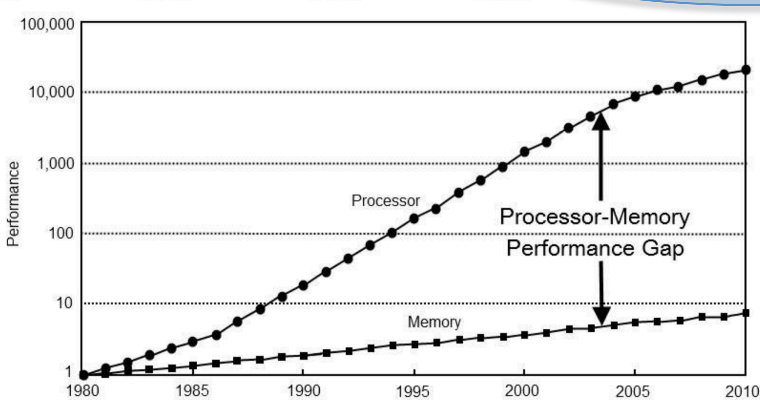
Task: You are asked to generate 1M Higgs events.
Jobs: You divide the task into 100 jobs, each running on 10k events,
each on various super-computer centers from around the world (including CERN, left),
forming the world wide LHC Computing Grid (LCG, right).
Event: Each event can run in a different core or thread on a supercomputer.

1M cores always running worldwide for the LHC research.
Already 1 ExaByte (EB) data produce and stored at CERN.
Every year 0.1 EB is moved around the world via the Grid.
(1 ExaB = 1024 PetaB; 1 PetaB = 1024 TerraB).

Although the number of transistors per chip continues to grow as per Moore's law, the clock frequency is constant since 2006 (about 3 GHz), due to the power limit wall.

The processor-memory gap widens.





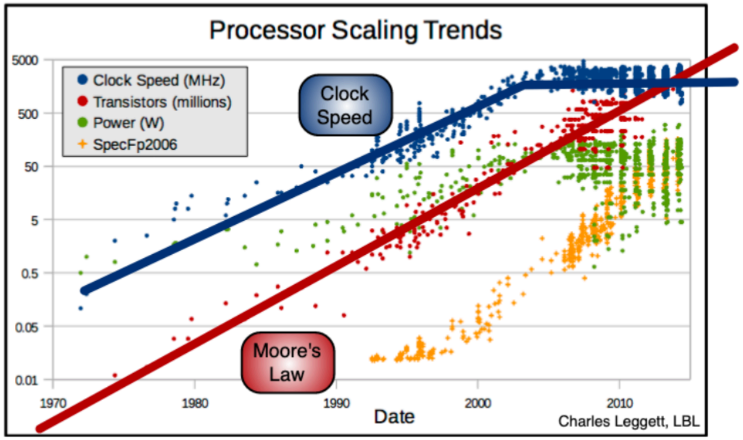
The new era

Computers don't become faster (clock frequency) every year as they used to.

Instead, they add **more and more logical cores**, meaning more independent threads that can be used in parallel.

Key to design algorithms that can run efficiently in **parallel computing**!

Big change in each event analysis: sequentially to concurrency. Run different algorithms for the same event, at the same time, in different cores (threads).



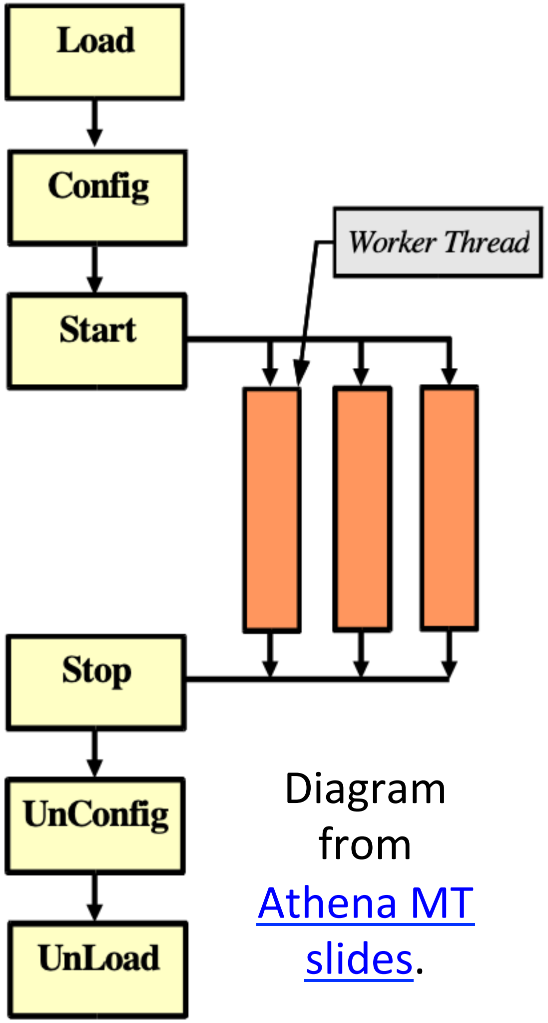


Diagram from [Athena MT slides](#).

Recall **while the clock speed has plateaued**, **the number of transistors keep growing (Moore's law)**.

The following can be used in a particle physics analysis:

- laptops usually have 2-4 cores
- big work stations have 50-100 cores -> batch computing (e.g. Condor)
- High Performance Computing (HPC) have even 2000 cores.

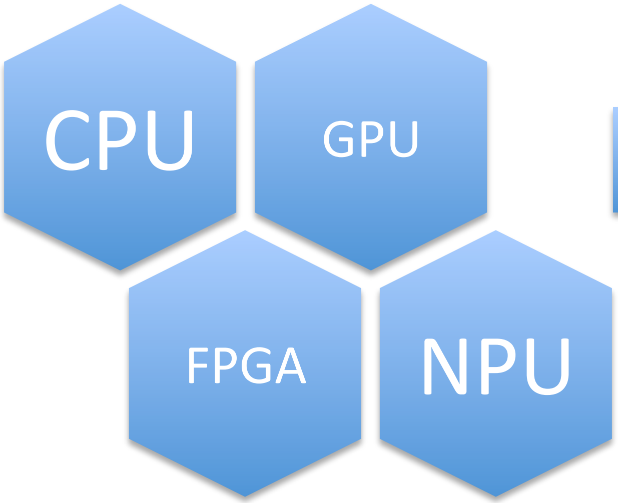
Example:

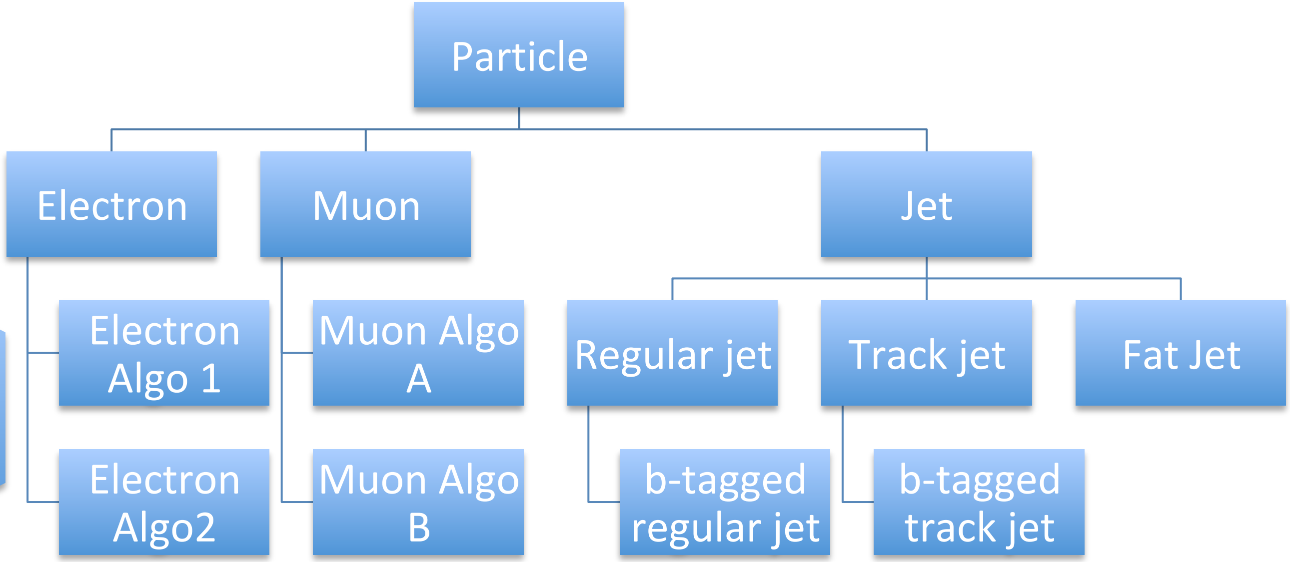
reconstruct tracks from the inner detector ionization hits in one core, while calorimeter clusters from calorimeter cells in another core.

But ideally re-write the current algorithms so that one event can run very fast on about 2000 cores at the same time.

ATLAS Athena Multi-Threading will be used in 2021 in Run-3 data taking.
https://cds.cern.ch/record/2142775/files/ATL-SOFT-PROC-2016-005.pdf

Use new hardware architectures available besides CPU. Create more efficient event data model (EDM).





The event data model (EDM) describes the physical information of an event using C++ inheritance, like in the diagram above.

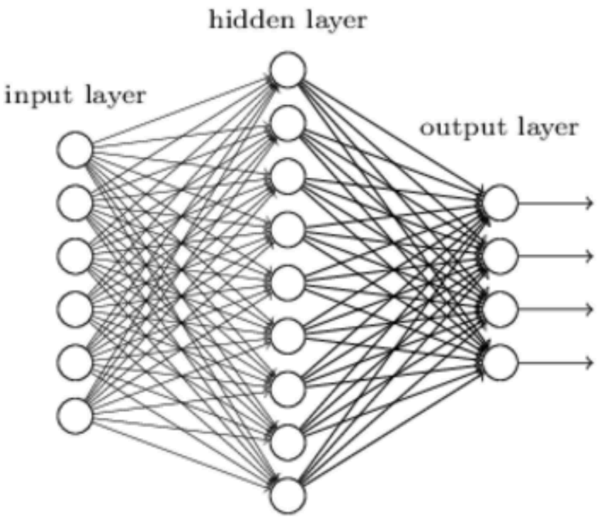
Very inefficient in terms of code, but this was hidden due to faster processors appearing all the time.

With the clock limit, and era of many cores, EDM must also be improved. Maybe learn from from CMS's nano-AOD, with a reduced data format.

CPU – central processing unit.
GPU – graphical processing unit.
FPGA – field-programmable gate array, used for example in the ATLAS trigger (slides, public note).
NPU – neural (network) processing unit

Machine learning (ML) is already widely used, like shallow neural networks (left). In future, deep neural networks will dominate, but they need more data and computing power.

"Non-deep" feedforward neural network



Deep neural network

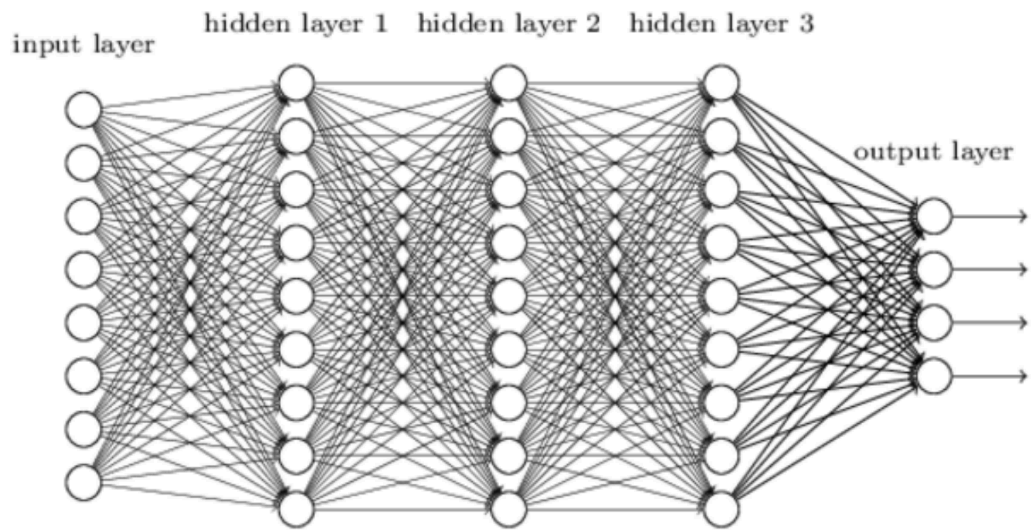


Table 1 | Effect of machine learning on the discovery and study of the Higgs boson

Analysis	Years of data collection	Sensitivity without machine learning	Sensitivity with machine learning	Ratio of P values	Additional data required
CMS[24] $H \to \gamma\gamma$	2011–2012	2.2σ, $P = 0.014$	2.7σ, $P = 0.0035$	4.0	51%
ATLAS[43] $H \to \tau^+\tau^-$	2011–2012	2.5σ, $P = 0.0062$	3.4σ, $P = 0.00034$	18	85%
ATLAS[99] $VH \to bb$	2011–2012	1.9σ, $P = 0.029$	2.5σ, $P = 0.0062$	4.7	73%
ATLAS[41] $VH \to bb$	2015–2016	2.8σ, $P = 0.0026$	3.0σ, $P = 0.00135$	1.9	15%
CMS[100] $VH \to bb$	2011–2012	1.4σ, $P = 0.081$	2.1σ, $P = 0.018$	4.5	125%

Shallow learning (left) uses a few engineered variables with one hidden layer to separate the Higgs signal from SM background, allowing the discovery with less data (and a few years faster), than without a neural network.

Deep learning (right) uses more input data (advantage: one can use raw data, no need of human engineering), more hidden layers (disadvantage: need a higher computing power, so a GPU or NPU) but provides better output. Example: tracking with DL in HL-LHC.

See excelelnt Nature review article of Machine learning used in HEP!

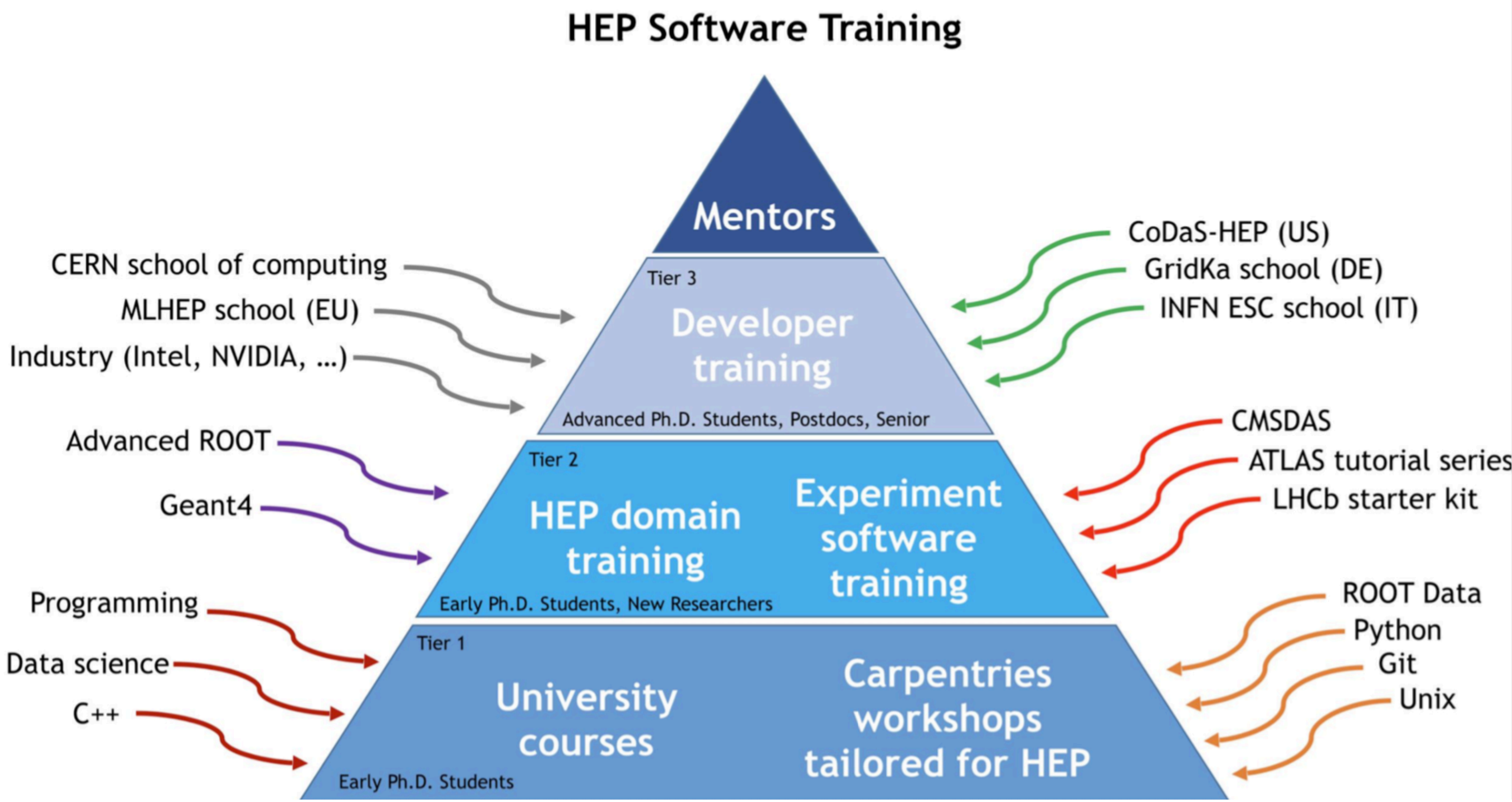
These are only a few examples of how computing is needed in HEP. The computing needs will only grow with more data but hardware resources that do not grow as much.

HEP Software Foundation

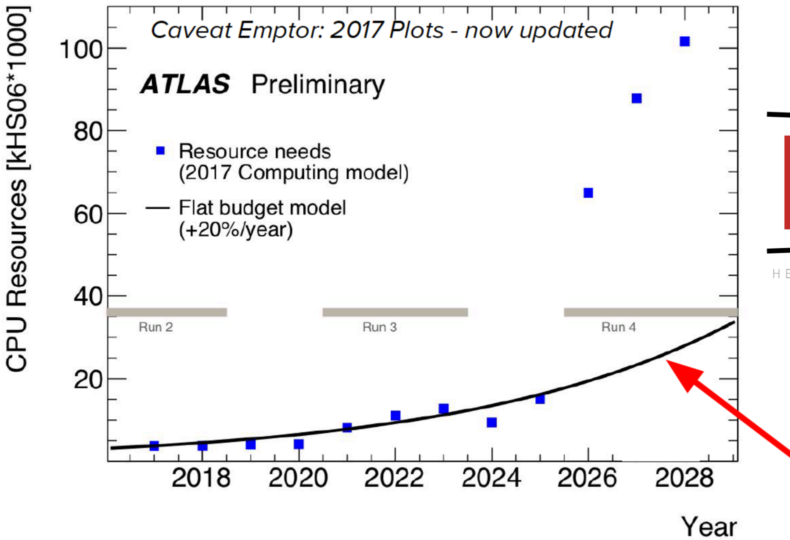
Created to bring together the entire particle physics community for the computing challenges and solutions for the next decades: HL-LHC, LBNF, ILC, etc.

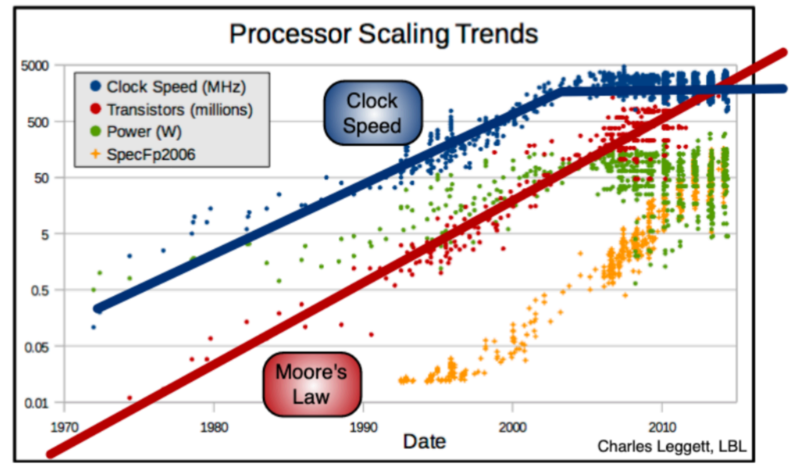
Currently every experiment develops their own software solutions.
But problems are very similar, and become ever larger.
Solution is to create general software usable for all experiments,
like is the case already for ROOT (data analysis) and Geant4 (detector simulation).

HSF has published a roadmap for 2020 and white papers. One of the main messages for universities is to train their students in programming (C++) and data science (statistics).



Conclusion: HEP Software Foundation has been created to bring together the HEP community to solve the huge computing and software challenges of the next decades.





7 years from now, LHC Run-4 will need computing beyond the current resources.
Need to apply new techniques, for example:

Concurrency (parallelization at sub-event level)

New architectures (GPU, PFGA, NPU)

Deep neural networks

Student programming training