Computing in High Energy Physics

G. Bagliesi
INFN - Pisa

Credits to Tommaso Boccali for most of the material shown here

Outline

1. High Energy Physics (HEP) experiments
    - Why do we build them?
    - How are they built (with a focus on LHC)?
    - Rough estimate of necessary resources
        - Some words on typical units of measurement
2. How to handle lots (lots!) of data
    - Possible solutions
        - The GRID solution
        - The Cloud solution(s)
        - HPC / ML / DL / AI / QC / …
3. How will they evolve?
        - 2020
            - 2025
                - 2035
                    - 2045
4. Analysis strategies

High Energy Physics / Particle Physics / HEP

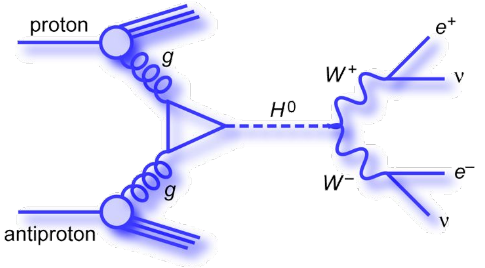
Particle physics

From Wikipedia, the free encyclopedia

For other uses of "particle", see Particle (disambiguation).

Particle physics (also known as **high energy physics**) is a branch of physics that studies the nature of the particles that constitute matter and radiation.

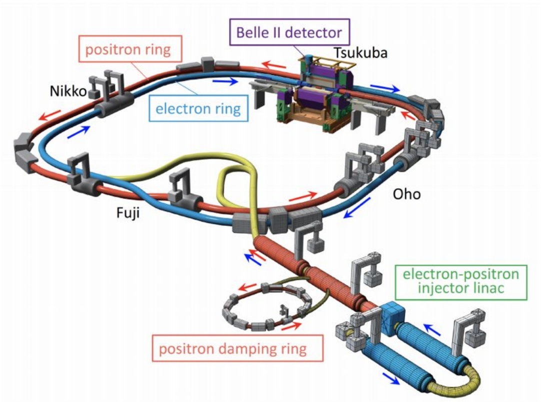
- The main mean of exploration without looking to astronomical phenomena, is to probe short distances / high energy / short time scales by preparing high energy systems
- Via Einstein's **E = mc²**, these systems can evolve into stable / unstable particles we can then probe and study



The highest the energy, the biggest the technical problems

- Bigger infrastructures
- More precise detectors
- .. And more data collected!

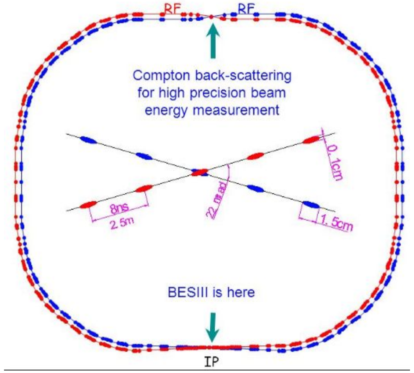
Which are today's (main) operational colliders?



SuperKEKB (JP)

- Collides electrons and positrons
- Center of mass energy 10.6 GeV
- ~3 km «circumference»

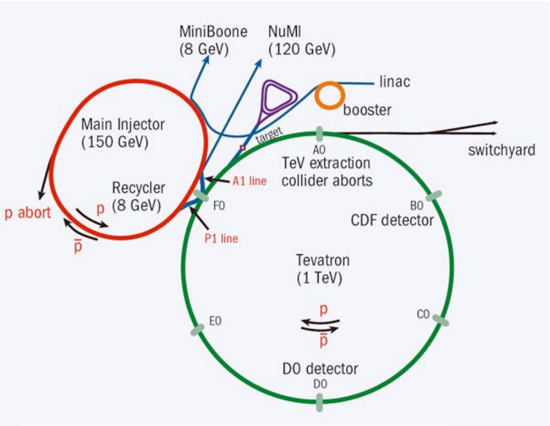
- (1 GeV = 1.6 10^{-10} J)

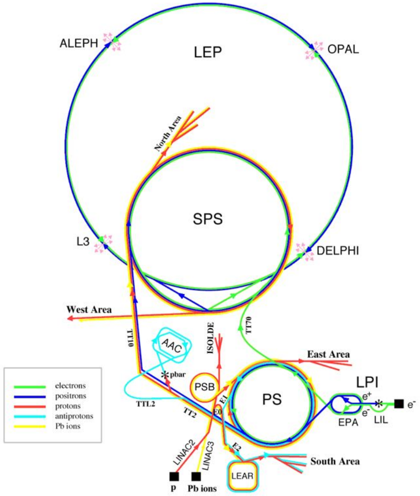


BEPC II (CN)

- Collides electrons and positrons
- Center of mass energy up to 4.6 GeV
- ~0.2 km «circumference»

Not operational anymore …





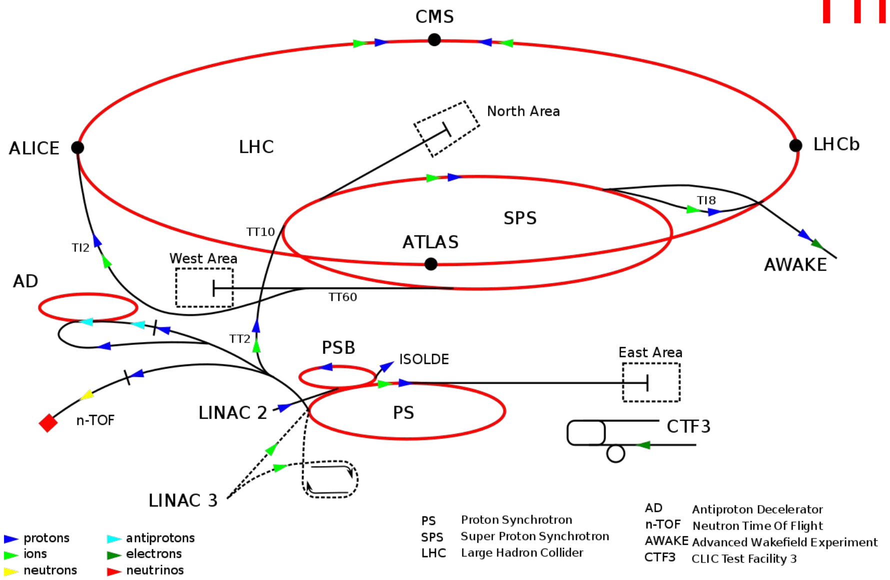
Tevatron (US)

- Collided protons and antiprotons
- 1983-2011
- Center of mass energy up to 2000 GeV (2 TeV)
- ~6.3 km circumference

LEP (CH)

- Collided electrons and positrons
- 1989-2000
- Center of mass energy up to 209 GeV
- 27 km circumference

The highest energy: LHC

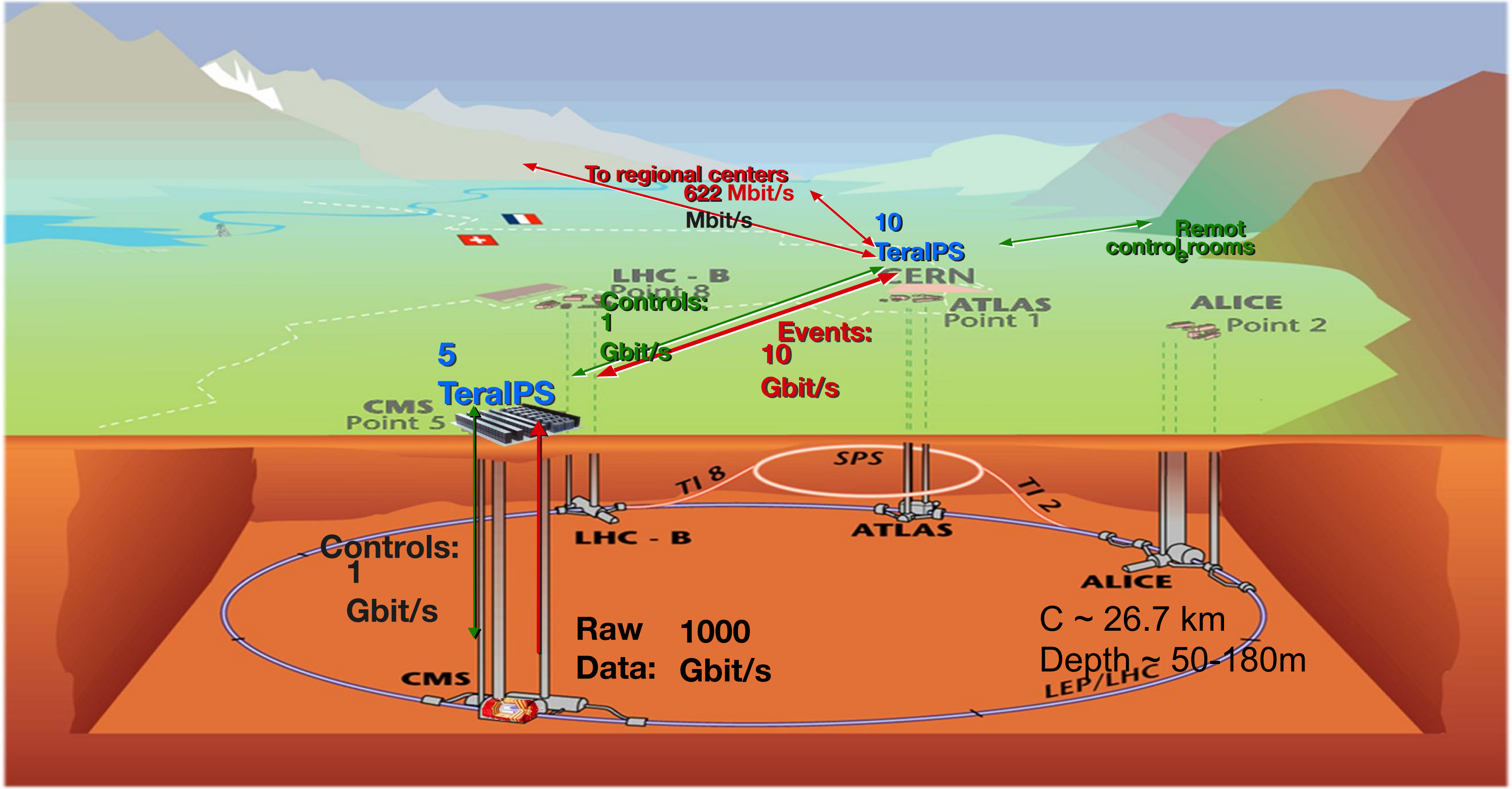


LHC (CH)

- Collides protons
- Since 2010
- Center of mass energy 13000 GeV (13 TeV)
- 27 km circumference
- In the picture you also see the various rings needed for pre-acceleration

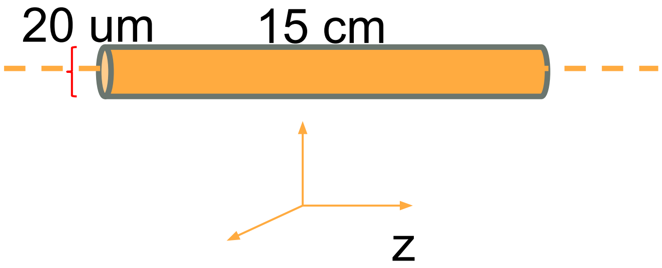
For some of these parameters (and others), the data needs from LHC are much larger than the previous experiments. **HEP has computing needs comparable or larger than more usual Big Data examples**

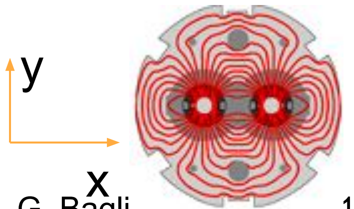
Experiments at LHC



How does it work?

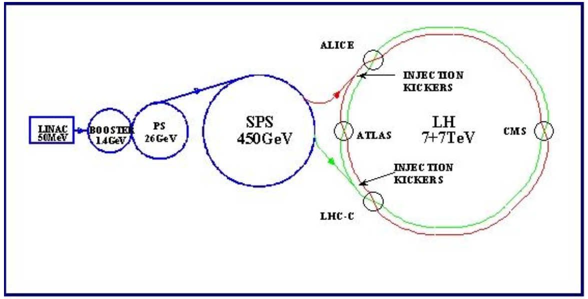
- You prepare “bunches” of protons
  - 1.4 10^{11} p per bunch
    - A bunch is at collision ~ 15 cm long, ~20 um in diameter ... A long “tube”
  - You put as many bunches (“*n*”) as you can on a 27 km circumference
    - @25 ns spacing means 7.5 m spacing at *c*
    - 27km/7.5 m = 3600 possible bunches
    - Only 28xx are available, the others are needed empty for safety reason (a time with no protons long enough is needed to dump the beam in a safe place)
  - At every turn, each bunch ideally crosses all the others (*nxn*) but only *n* such collisions happen in a given position where a detector is located





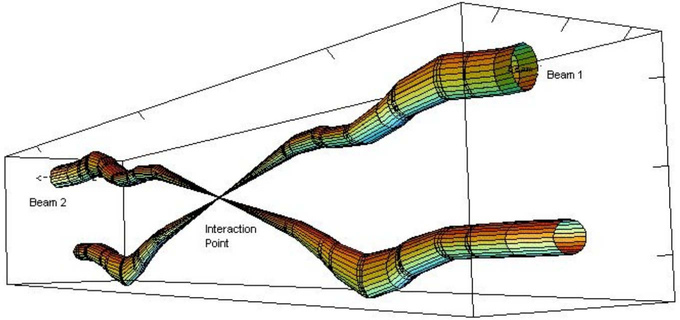
How does it work? #2

- Bunches are injected in LHC from the chain (source)-LINAC-BOOSTER-PS-SPS-LHC, which increase energy at every step
- They are “squeezed” in (x,y) and z by quadrupoles, and forced in a circular orbit by dipoles; they are accelerated using radio frequency cavities
- At some specific locations, the two p beams are collapsed and put into collision



Quantity	number
Circumference	26 659 m
Dipole operating temperature	1.9 K (-271.3°C)
Number of magnets	9593
Number of main dipoles	1232
Number of main quadrupoles	392
Number of RF cavities	8 per direction
Energy, protons*	6.5 TeV
Energy, ions	2.56 TeV/u (**)
Peak magnetic dipole field	7.74 T
Distance between bunches	~7.5 m
Luminosity (protons)	Peak Luminosity: ~ 1.2 x 10^{34} cm^{-2} s^{-1}
No. of bunches per proton beam (design value)	2808
No. of protons per bunch (at start)	1.2 x 10^{11}
Number of turns per second	11 245
Number of collisions per second	1 billion

(*) Design value: 7 TeV
(**) Energy per nucleon



Relative beam sizes around IP1 (Atlas) in collision

Numbers of LHC...

How much energy are we talking about?

7 TeV = $7 \cdot 10^{12}$ eV · $1{,}6 \cdot 10^{-19}$ J/eV = **$1{,}12 \cdot 10^{-6}$ J**

It doesn´t look like a lot of energy

For the **ALICE** experiment, **each ion of Pb-208 reaches** 1150/2 = **575 TeV.**

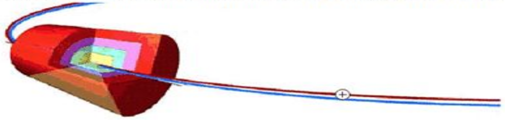
So, the energy per nucleon is: 575/208 = 2,76 TeV

Let´s calculate the kinetic energy of an insect of 60 mg flying at 20 cm/s:

$$E_k = \frac{1}{2} m \cdot v^2 \Rightarrow E_k = \frac{1}{2}\, 6 \cdot 10^{-5} \cdot 0{,}2^{\,2} \sim 7 \text{ TeV}$$

That is, in LHC each proton will reach an energy similar to that of an annoying ... **MOSQUITO!**

But we have to keep in mind that this mosquito has 36 thousand trillion nucleons, whereas the **7 TeV in the LHC will be concentrate in one sole proton.**



Maybe this comparison is not very convincing so let´s look at it from another point of view.

Let´s calculate the energy present in each bunch:

7 TeV/protón$\cdot 1{,}15 \cdot 10^{11}$ protons/bunch ~**$1{,}29 \cdot 10^5$ J/bunch**

A powerful motorbike 150 kg travelling at 150 km/h_

$$E_k = \frac{1}{2} \cdot 150 \cdot 41{,}7^2 \sim \mathbf{1{,}29 \cdot 10^5 \text{ J}}$$

So if a bunch of protons collides with you the impact is similar to that produced by a powerful motorbike travelling at 150 km/h.

If you are lucky to avoid that "**0,2 picogram motorbike**", don´t worry, there are 2807 following it. And if you decide to change lanes, the equivalent is coming in the opposite direction.

Another calculation which can show the enormous amount of energy reached is:
$1{,}29 \cdot 10^5$ J / bunch x 2808 bunches ~ **360 MJ**

-Stored beam energy-

And that is equivalent to

77,4 kg of TNT

The energy content of TNT is 4.68MJ/kg (Beveridge 1998).

The Heat of Fusion of Gold is ΔH_F = 63,71 kJ/kg and the Molar Heat Capacity is 25,42 J/mol·K

So, 360 MJ are enough to take 1500 kg of Gold from 25°C to total fusion ⇒ 1,5 Tonnes of Gold.

Obviously **such an amount of energy can not be supplied instantly**. In fact **the process lasts over 20 min** through a chain of different accelerators.

Why do we need such extreme parameters?

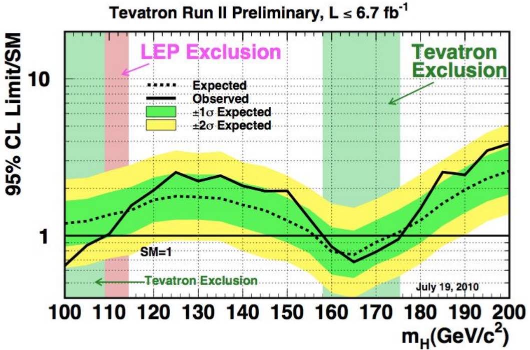
- LHC was built having in mind a very rich physics program, but with a clear focus on two possible fields
  - **Higgs' boson discovery & physics**
  - Search for physics beyond the Standard Model
  - Look for the unexpected

- The fields are by no means "new", and has already been attempted at least it the last two "discovery machines": **LEP** (CERN, ~1989-2000) and **Tevatron** (Fermilab, ~1985-2011)

- So we knew in advance where that physics was NOT to be found, and LHC was thought and built mostly in order to explore the same physics in new energy regions.

Let's just focus on Higgs Boson: where to search for it

- After LEP and Tevatron, we knew quite well where NOT so search for it
  - LEP: lower mass limit ~115 GeV (direct exclusion)
  - LEP: most probably below 200 GeV (indirect limits, depending on many theoretical assumptions)
  - Tevatron: not in the range between ~160 and ~175
- Strong theoretical arguments against a Higgs boson higher than 1 TeV



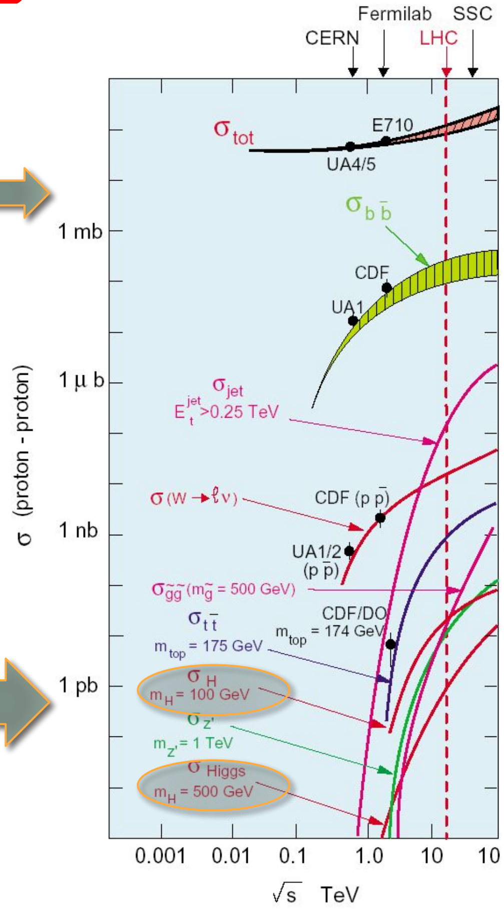
The nice feature of standard Higgs searches is that once you have (postulate) the mass, all the other parameters like couplings, production, decay rates are known (its mass is the **last unknown parameter in the standard model**), hence one can plan on Higgs characteristics

Higgs boson production: to the problem's root

- Higgs production cross section (how probable to create one) increases very sharply with collider energy
- The actual number of produced events in a given process is proportional to its cross section, and the collider luminosity
- **$N = \sigma \times L_{int}$**

  How probable the process is "per collision" (1 m^2 = 10^{28} barn)

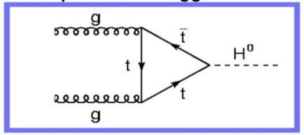
  How many collisions we are trying m^{-2}
- Where L_{int} is the integrated luminosity an experiment has been given
- Quite varying with the mass, but the typical Higgs production cross section is ~1-100 pb @ a 13 TeV collider
  - @ 1 TeV collider it would be ~ 100-1000 times lower, this is the reason why a direct positive discovery at Tevatron was basically hopeless



Higgs production and decays at LHC

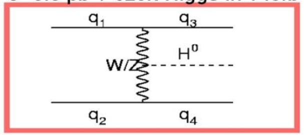
σ=49 pb / 6.9M Higgs in $140 fb^{-1}$

~88%



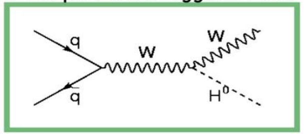
σ=3.8 pb / 520k Higgs in $140 fb^{-1}$

~7%



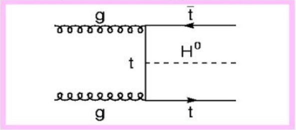
σ=2.3 pb / 320k Higgs in $140 fb^{-1}$

~4%

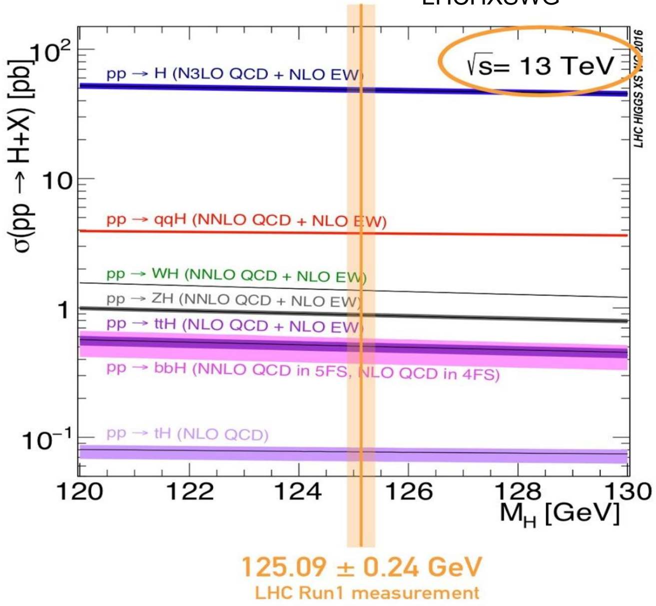


σ=0.5 pb / 70k Higgs in $140 fb^{-1}$

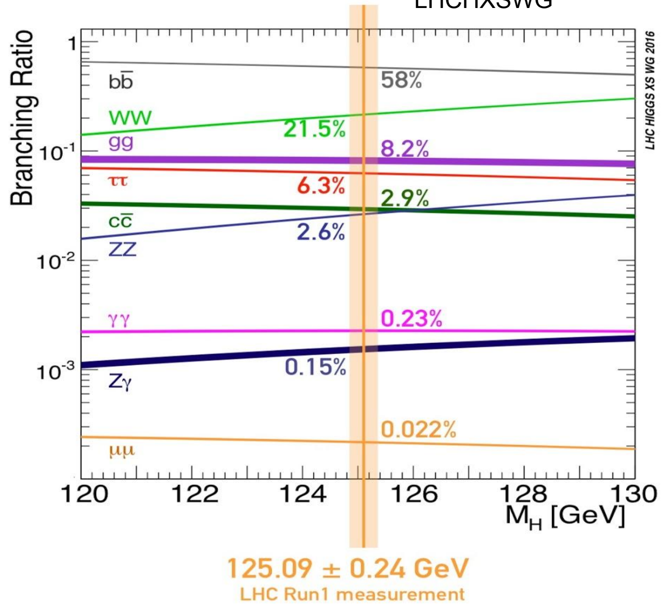
~1%



LHCHXSWG



LHCHXSWG



And then, which collider parameters do we need?

- Total "integrated" luminosity is the time integral of the instantaneous luminosity
- $L_{int} = L_{inst\_average}$ x (data taking seconds)
- And again, L_{inst} is

$$L = \frac{f \sum_{i=1}^{k_b} N_{1i} N_{2i}}{4\pi \sigma_x^* \sigma_y^*}$$

LHC
3564 (2808) bunches
10^{11} p/bunch
Δt =25 ns
$\sigma^*_{x/y}$ = 375.2(16.7) μm [IP1]

f = revolution frequency (c/27 km)
N_{1i}, N_{2i} = number of protons in i-th bunch
k_b = number of bunches
σ_x, σ_y = transversal dimension of bunches in the colliding area

Putting all together ...

- If your goal is to have 10.000.000 produced Higgs in 5 years (per experiment)
- L_{int} = 100 fb^{-1} (10^7/(10000fb)) and then, scaling to the instantaneous lumi (assuming an efficiency factor ~5 for shutdown periods, vacations, repairs, etc)
- L_{int_max} = 100 fb^{-1}
- If you remember that 1 b = 10^{-24} cm^2 → L_{int} = 10^{41} cm^{-2}
- 1 y of data taking → 10^7 s

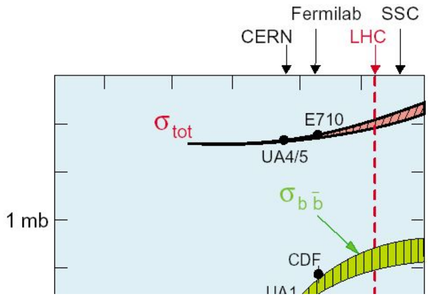
$$\mathbf{L_{INST} = 5 * 10^{41}\ cm^{-2} / (5\ y * 10^7 s/y) =\sim 10^{34}\ cm^{-2}\ s^{-1}}$$

- This is what you get in the previous page formula with LHC parameters

- **SO: the extreme LHC parameters are the only way to "guarantee" LHC would have been able to discover / exclude the Higgs boson in the energy range where we were searching for him.**
- **Any machine with lower parameters could have not been able to close the issue on the Higgs (if you want, not well spent money)**

Summary on LHC

- It collides bunches of $1.x10^{11}$ protons every 25 ns
- At each beams' collision, O(<u>25</u>-50) hadronic events are generated
- Total = **1 billion** hadronic collisions per second
- Each collision ~ 50 primary particles on average
- **<u>50</u>-100 billion primary particles per seconds are generated into each experiment**



Total pp cross section ~ 100 mb

100 mb * 10^{34} $cm^{-2}s^{-1}$ = 10^9 /s = 1 hadronic event per ns = 25 hadronic events per bunch crossing

.. But in reality the machine has been able to reach 2 10^{34} → 50!

This is one of the most important scaling parameters also when considering computing needs computing

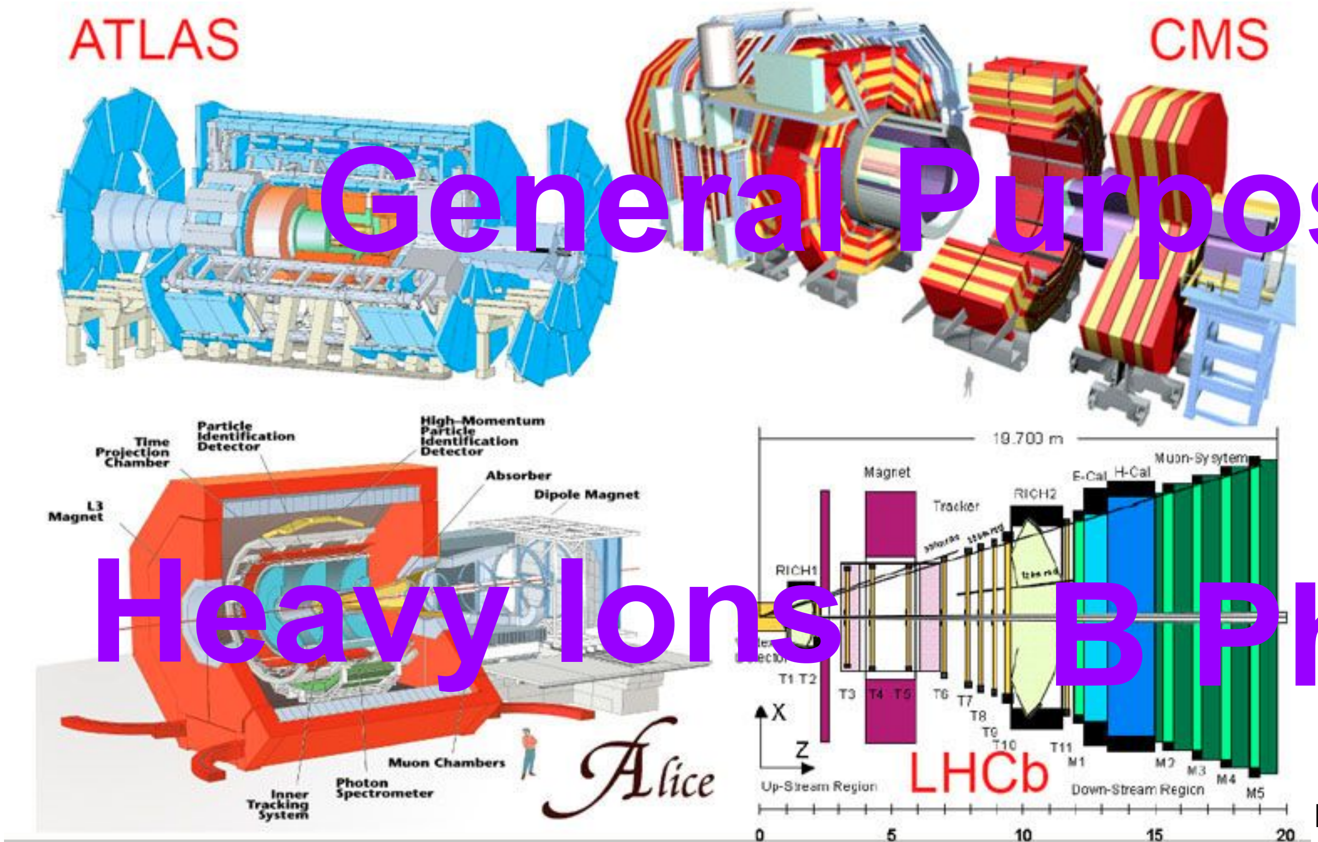
Requirements...

- Build detectors ("experiments") able to sustain and use such a particle rate, and extract "physics knowledge" from the collisions

- The same detectors have to survive for at least 5 years to the particle flux, while being able to identify/select the 10000000 Higgs which are produced, among ~10^{15} collision events

- **Selection factor = 10000000/10^{15} $\rightarrow$ 1 "interesting" events every 100 million interactions**

LHC Experiments (the major ones)



General Purpose

Heavy Ions

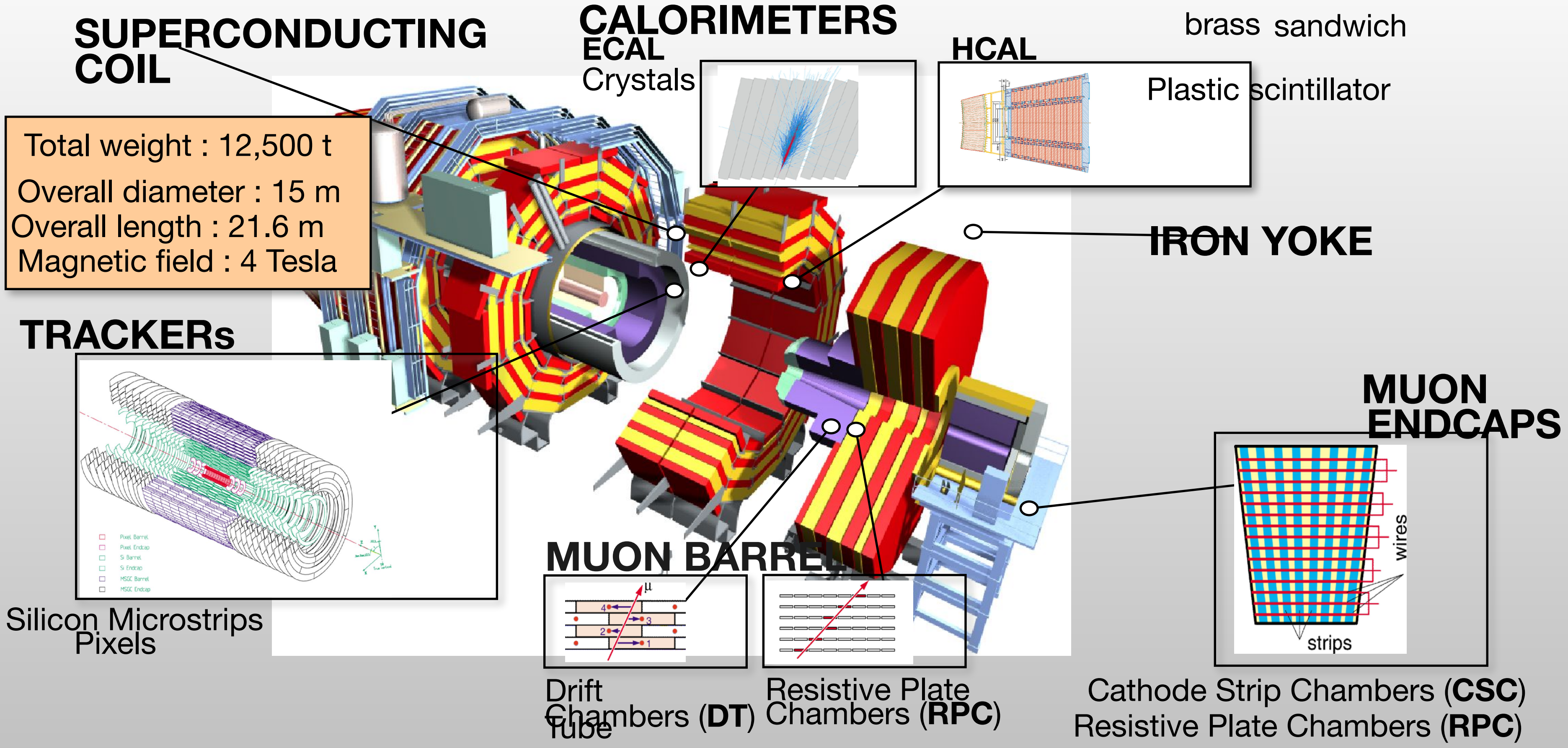
B Physics

Detectors

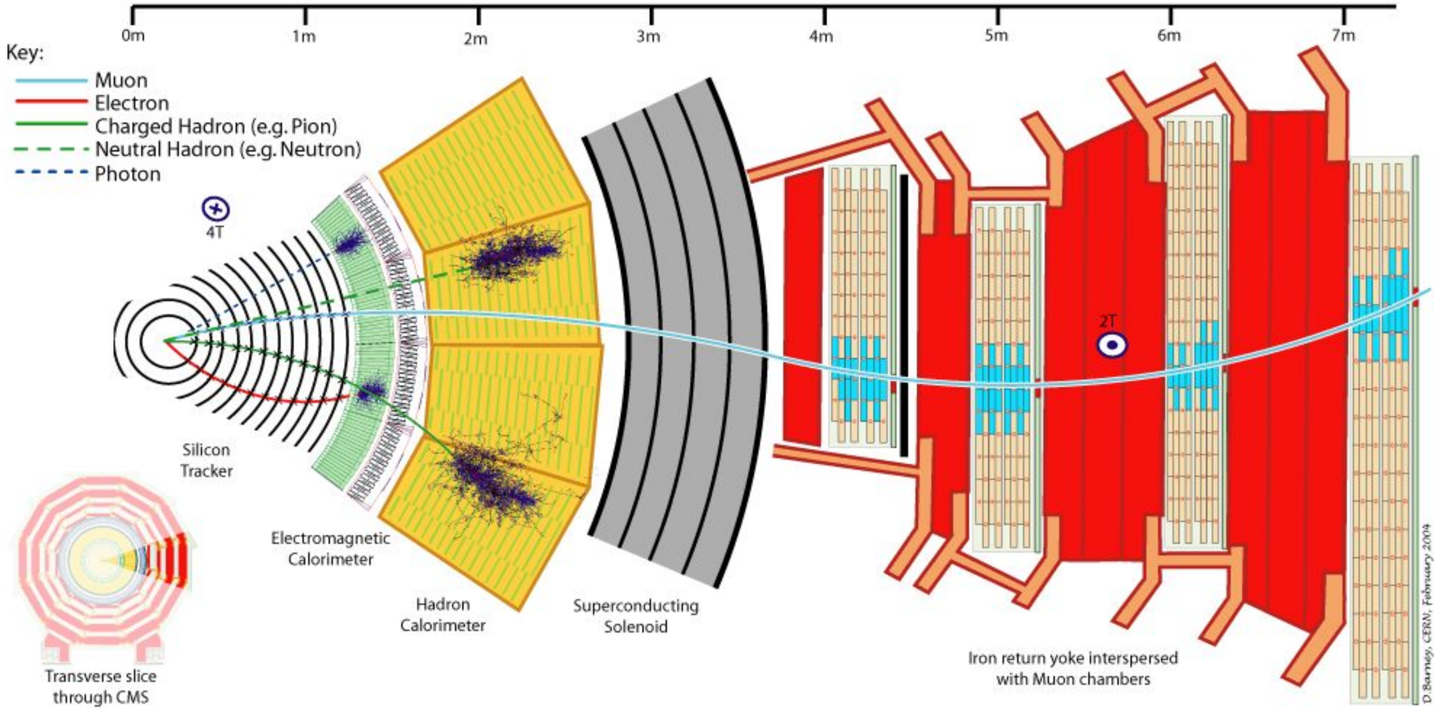
- There is no time to describe here LHC detectors, and it is not even the scope of this seminar, but
  - The extreme event selection capability requires a strong precision on basic physics quantity measurements (like **momentum, energy, position**) for all the particles produced in the collisions
  - The only way we know to achieve this is via complex detectors, with many measuring channels (“**acquisition channels**”)

- Without distinguishing between the experiments, the average number of **DISTINCT acquisition channels** (“wires” going into a computer) is **about 100 Million**
  - And we can suppose each of these will produce 1 Byte per reading (naïve but not too unrealistic)

The Compact Muon Solenoid (CMS)



How particles are seen in the experiment



Units of Measurements in HEP Computing

- Storage
  - 1 byte (B)= [0…255]
  - 1 GB = 10^9 B
  - 1 TB = 10^{12} B
  - 1 PB = 10^{15} B
  - 1 EB = 10^{18} B
  - 1 ZB = 10^{21} B
- today= 1 HardDisk ~ 8 TB
- Network:
  - 1 Gbit/s = 2^{30} bit/s ~ 100 MB/s
- Today = National REsearch Networks (NREN) ~ 10-100 Gbit/s

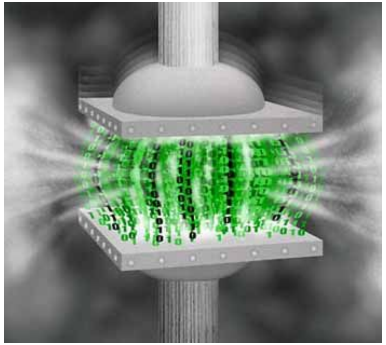
- CPU:
  - 1 HepSpec06 (HS06) = unit specifically thought for HEP

  - Today = 1 computing core ~ 10 HS06
  - Today = 1 CPU (~16 cores) ~ <200 HS06

Which is the expected data rate?

- 40 Million collisions per second * 100 Million acquisition channels * 1 Byte per channels per collision = **4 PB/s**
  - **In 5 y, usual factor 5 = 4 PB/s * 5y * 3 10^7 s/y /5 = 96 ZettaBytes**
    - 1 ZB = 10^{21} B = 1000000 PB
- Here we enter directly Computing Models realm: how to
  - Reduce 4 PB/s to something manageable
  - Analyze such a data flow and produce something human readable (a physics paper, for example)
    - Like: “Higgs Mass is 125 GeV”
  - **Taking to the extreme, Computing Models are the means to reduce 96 ZB to one Byte**

96 ZB



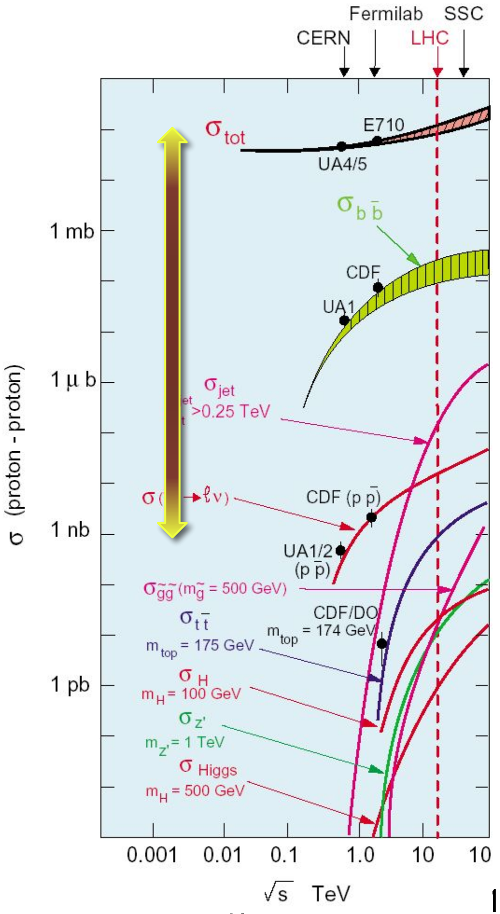
1 byte

But we need much less, luckily !

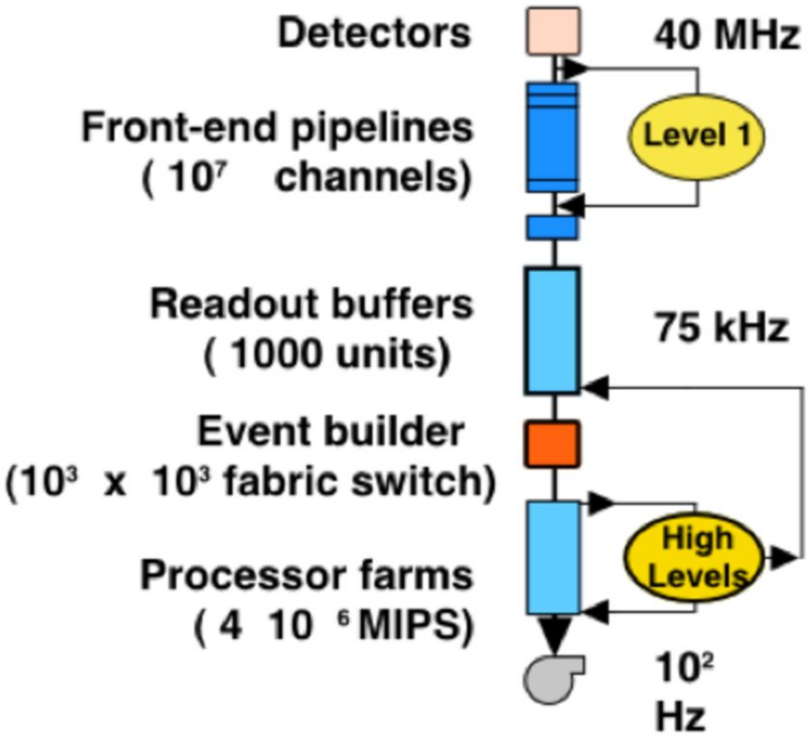
- It is absolutely clear no one will be able in the near future to handle 4 PB/s with IT systems, **by many orders of magnitude**

- It is also clear that the very bulk of this rate consists not so interesting events (like low energy QCD): **there are 5+ orders of magnitude between total cross section and interesting phenomena**

- The largest part of the events, if correctly identified, can be just thrown away
  - "if correctly identified"



The trigger: select a subset of interesting events



- Input = 40 MHz (1/(25ns))
  - LHC bunch crossing rate
- Custom electronics select and reduce down to ~100 kHz (selection factor ~1/400)
- A second system, **based on commodity CPUs**, which works on semi-optimal quantities, goes down by another ~100 to O(1000 Hz)

Decrease in data rate: not only trigger

- We said we work under the assumptions that each detector has ~ 100 Million acquisition channels, 1 Byte each per event
- Reading all of them is impossible, but also useless: most will not have values resulting from having been hit by a particle, but some form of **noise**

- **Zero Suppression** is the process with which on board detector electronics is able to detect null results (only due to noise), and transmit only real results
- Final event dimensions scale down by a factor 100 thanks do this for proton-proton collisions, 10 for Heavy Ions collisions
  - In what follows we will assume that event size is ~ 1 MB in pp, ~ 10 MB in Ion collisions

Realistic numbers to deal with...

- Fast recap of parameters
  - Rate of selected (triggered) events: O(1000) Hz
  - Typical dimension of each event: O(1) MB
  - Seconds of data taking per year: O(7 10^6) s
- Amounts to :
  - 7 Billion events per year
  - 7 PB per year of “RAW” data
- Much lower than the initial figures, manageable ….
- **Now what?**

Typical data workflow

- A physicist is not able to interpret directly the RAW data form the detector
- He is used to think in terms of Particles, Jets, Decay Chains, ..
- The process which allows for the interpretation of RAW data in terms of physical objects is called “reconstruction”, and it is usually CPU intensive.
- **So: we do not have only the too-much-data problem, but also the too-much-cpu …**

And also simulations

- Up to now we spoke just about Data from the experiments

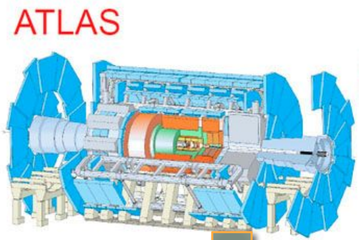
- In reality, this is not all of it. HEP dynamics, while in theory quite well known, in practice does not provide an analytical solution from the initial high energy collision to hadronization, decays, and finally stable particles.

- The only viable method is to generate statistically distributions via a **Monte Carlo method**, and compare these with the data

- In practice: events are “generated” sampling theoretical models with high statistics, and the events are then formatted to look as close as possible identical to the data events. **In this way, a 1-to-1 comparison can be cast between data and simulated events**

Data workflow

LHC collisions → Decay of unstable particles →

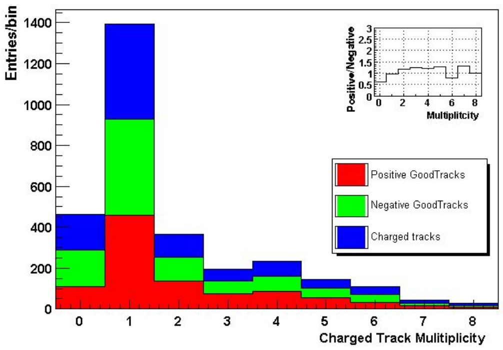


↓ Detector electronics

↓ Trigger

↓ Reconstruction

↓ Analysis



Simulation workflow

Theoretical model

Simulation of decays of unstable particles

Simulation of interactio particle-detector

Geant4

Madgraph, Sherpa, …

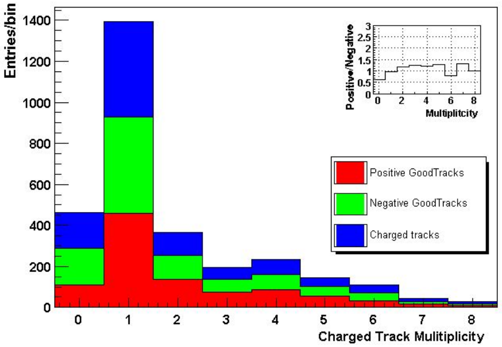
Pythia,…

Simulation of detector electronics

Trigger

Reconstruction

Analysis



Simulations

- As a consequence, **theoretical estimates** are not given to the experimental physicist as equations or such, but as simulated events which
  - as number
  - as size/content
- Are as close as possible to real data

- An accurate description of the models (due to its sampling) requires that the number of simulated events **cannot be too small**; they are typically at least matching the real data events (more realistically, at least 2x more).

- Storage and CPU needs to store and analyze simulated events is not smaller than the one for data
  - Our approximation: **we need to scale by at least 3x all the computing figures we have given up to now**

CPU needs in HEP

- The most important use cases are
  - **Event reconstruction**: CPU need varies per experiment, but a reasonable estimate is 30 sec/event on today's CPU
    - 300 sec x HS06/ev
  - **Event simulations**: simulation of interaction of particles with matter (Geant4, mostly)
    - 500 sec x HS06/ev
  - **Final data analysis** (fits, final selections, result extraction, etc etc)
    - 1-10 sec x HS06/ev

Official experiment figures (2018)

Experiment	Size RAW (MB)	Size RECO (MB)	Reduced size (analysis)	Reconstruction (sec.HS06/ev)	Simulation (sec.HS06/ev)	Analysis (sec.HS06/ev)
ALICE	1	0.04	0.004	25	150	2-64 pp
ALICE	12	2.5	0.25	3000	70000	30-1000 HI
ATLAS	1	0.5	.1	300	3000	2
CMS	1	0.5	0.05	300	500	1
LHCB	0.025	0.075	0.025	10		1

A single data taking year ….

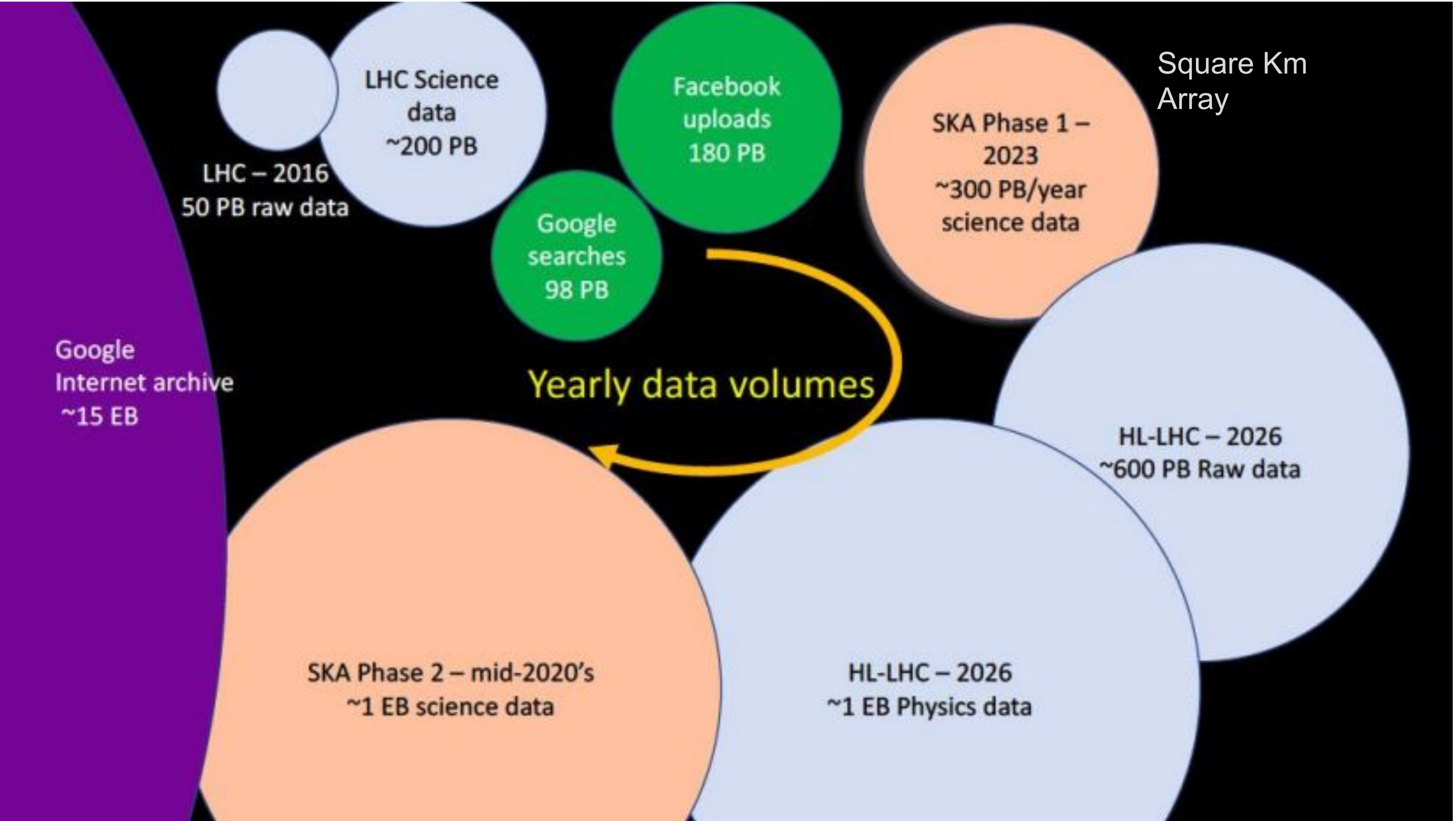
- Storage
  - Data:
    - 7 PB RAW (x2 for a backup copy)
    - 3.5 PB reconstructed data
  - MonteCarlo
    - 14 PB RAW
    - 7 PB reconstructed simulation
- ~30 PB/year
- CPU
  - Data:
    - 7 10^9 ev*300 sec*HS06/ev = 2 10^{12} sec*HS06 = 70000 HS06 for the entire year (--> 7000 CPU cores)
  - MC
    - 2x70000 HS06 reconstruction
    - 2x110000 HS06 simulation
  - Analysis (MC + DT):
    - 7 10^9ev*2*10 sec*HS06/sec *N = 1.4 10^{11} sec*HS06 *N = 4500*N HS06
    - Where N is the number of independent analyses,can be very high (~100)
  - TOTAL: 70000+140000+220000+450000 ~ 1M HS06
  - **With current hardware:**
    - **3000 HDD/y**
    - **100000 computing cores**
  - .. And these are per experiment!

Situation today (2019, after 7 years of data taking)?

Experiment	CPU (kHS06)	Disk (PB)	Tape (PB)
ALICE	1000	100	85
ATLAS	2800	230	310
CMS	2000	160	280
LHCB	450	45	90
TOTAL	**6250**	**535**	**765**

Resources experiments have online in 2019

Factor ~2-3x wrt previous estimates (many details, more MC, more intense analysis activities, …)


LHC Science data ~200 PB
LHC – 2016 50 PB raw data
Facebook uploads 180 PB
Google searches 98 PB
SKA Phase 1 – 2023 ~300 PB/year science data
Square Km Array
Google Internet archive ~15 EB
Yearly data volumes
HL-LHC – 2026 ~600 PB Raw data
SKA Phase 2 – mid-2020's ~1 EB science data
HL-LHC – 2026 ~1 EB Physics data

Summary of computing needs

- Even if we try and
  - Discard all the non interesting events
  - Pack our detector data
  - Limit the number of simulated events to the bare minimum
- We still have a data / computing problem which by today standards is matched only by a few other fields

How to build on paper a Computing model in ~ 1995?

- When LHC computing models started to be sketched, a typical computer had
  - ~ 10 GB HDD (1/1000 of today's HD)
  - ~ 0.1 HS06 single core CPU (1/2000 of today's multi-core CPU)
- You can understand what **leap of faith in technology** is needed to think that in 10 years you will be able to handle resources which, in 1995, were of the same size of the entire world IT resource
- That said, how to handle this amount of resources?

Possibilities (in 1995, but also now…)

1. **A BIG data center**

2. **Many small data centers**

A big data center

- A large building with ~1.000.000 computing cores, and 200.000 HDD
  - Probably it would work; **Google** apparently has facilities much larger than that; **NSA** for sure has them
- But, the solution was considered not interesting, due to various reasons

1. **A single point of failure** (if CERN goes offline, LHC computing follows…)
2. **Political problems**: Member States were not so happy to finance “cash” computing at CERN (and in general, out of national boundaries)
3. **Manpower**: difficult to find locally the large amount needed
4. **(other) political problems**: member states wanted to increase their national expertise, not to finance Swiss ones …

Go distributed!

- During the ‘90s, as a pure IT concept, an alternative was born; the GRID

- In 5 minutes
  - Key concepts
  - Philosophy
  - implementations

GRID basic idea

The Grid Vision (by Ian Foster)

"Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations"

- On-demand, ubiquitous access to computing, data, and services
- New capabilities constructed dynamically and transparently from distributed services

"When the network is as fast as the computer's internal links, the machine disintegrates across the net into a set of special purpose appliances"

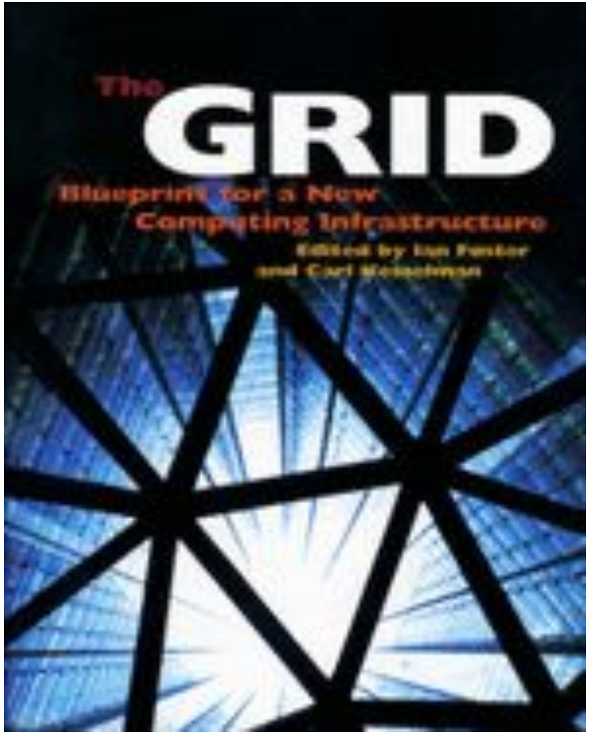
(George Gilder)

More simply …

- **Give access to heterogeneous and geographically distributed computing, without being (too) aware of this**
- GRID: they are named after the “power grid”

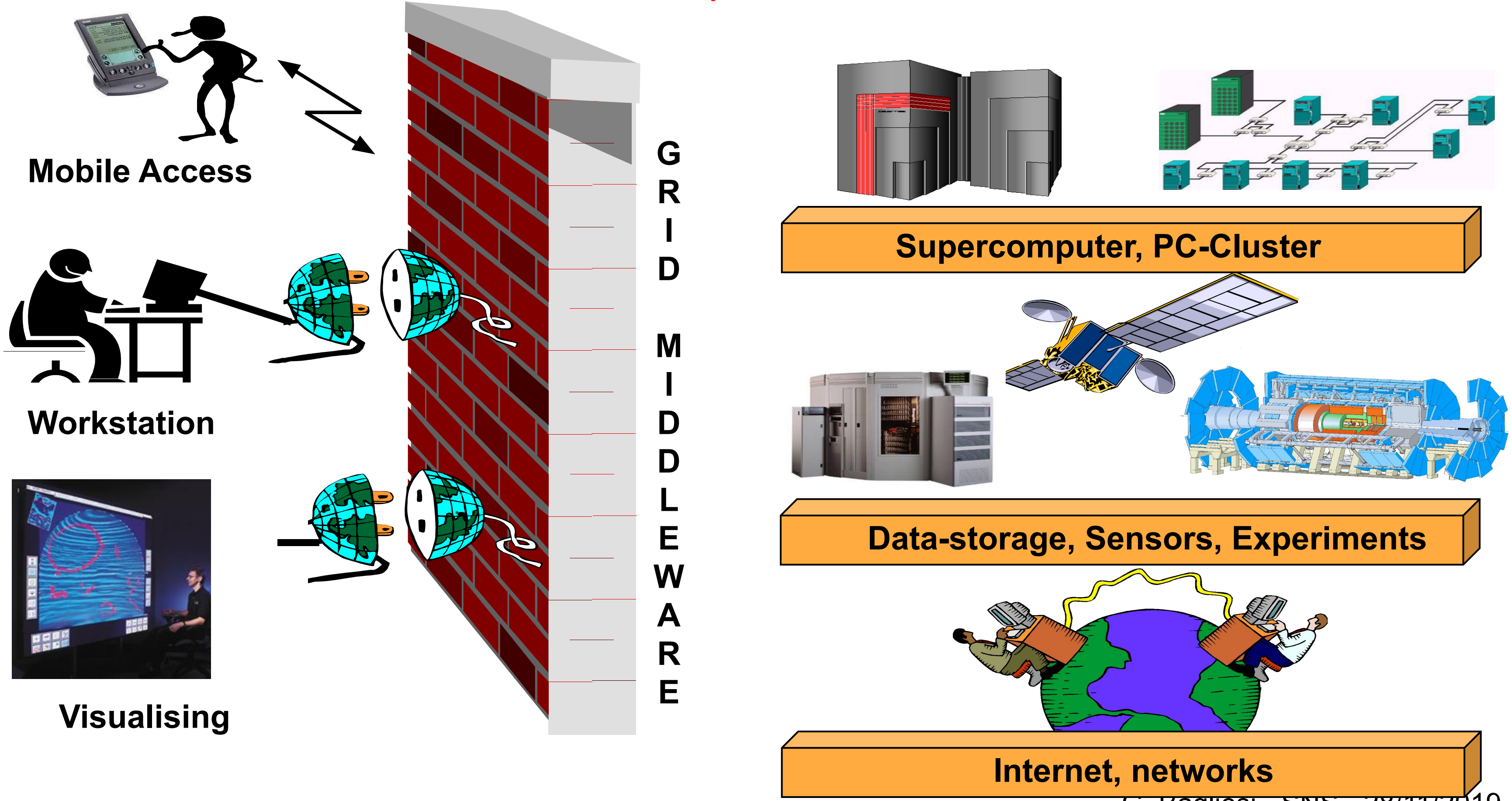
- **For example**: Italy produces idro-electric and thermal power, moreover Italy buys power from outside (France, …)
- But, when you need to use a blender, you do not need to care about
  - Which is the power source
  - Where was it produced
- You simply want and can access the power you have been given (== you decided to pay)

Formalization …



1999:
The GRID
Blueprint for a new
Computing Infrastructure

The Grid metaphor



GRID dreams...

- And, at least in some GRID implementations, some “resource brokering”
  - Given a computational task, **find the “best place”** where to execute it (on a planetary scale)
  - Given a filename, access it **wherever it is** (without explicitly knowing it)

- **GRID ambition was to have geographically distributed computing not different from local one, from a user point of view**

GRID and LHC experiments

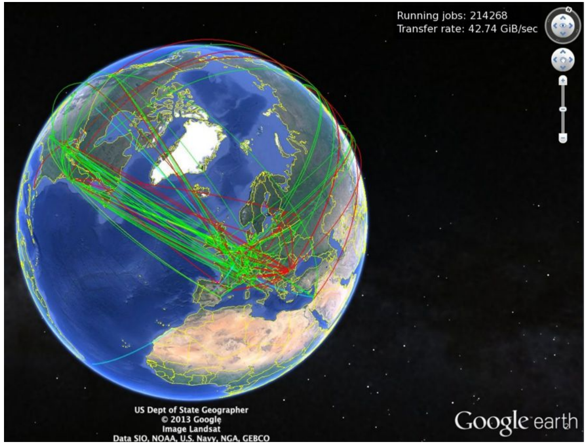
- So, distributed computing was chosen as the solution
- That given, how to organize LHC computing on it?
- **It turns out it is NOT as simple as to divide the resources in 50 sites and use them (regardless the GRID)**
- There is a nasty aspect we did not cover for the moment: the **Network**!
- Again some rough HEP estimates, this time on the networking

A single experiment networking needs

- RAW data = 1000 Hz * 1 MB/s = **1 GB/s**
- Reconstructed data = at least **2x** (including reprocessing)
- MonteCarlo = as data, so factor **2x**
- Analysis = a rough estimate gives 1 Mbit/s/HS06, so **10 GB/s**

- **Overall per experiment ~ 15 GB/s or O(150 Gbit/s)**

- In an ideal GRID environment, chaotically distributed among 50 sites (each of them should support a large fraction of this)



Indeed today's LHC traffic is O(500) Gbit/s, for the 4 experiments

~2000: which networks were expected to exist?

- In many states it was before network deregulation: single actor, semi-monopoly
  - **No Netflix, no Spotify, no bit torrents**
- Expected increase (also due to monopoly) less than a factor 2 per year, at a given price
- **Pisa INFN as example**: in 2000 it had a WAN connections via GARR (Italian research network) topping at 8 Mbit/s. In the 5-6 years to the LHC start no way to get to 10 Gbit/s, right? (ehm…)

- Result:
  - **It turns out it is possible to guarantee (== pay) only a small number of network connections, and require on these high performance**

We need to be Data Driven!

- Even if GRID is used, **if we do so we are not really "location independent"**: and **not all the sites are equal** (since they are served with different connections)

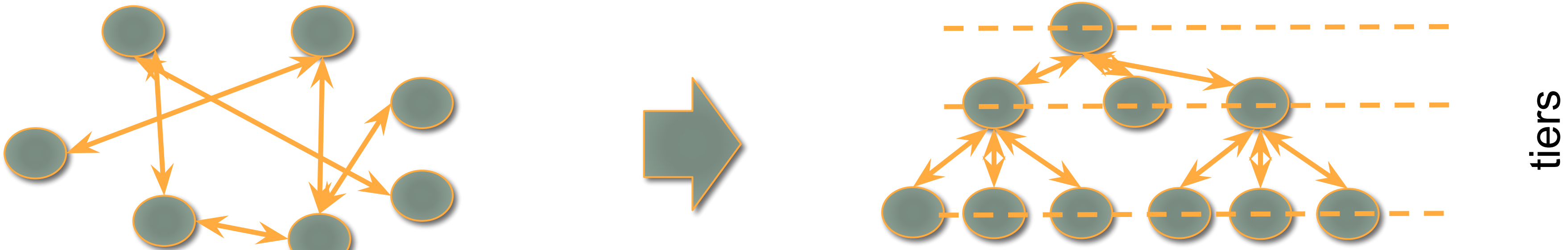
- LHC Computing model becomes <u>Data Driven</u>
  - The activity a single site can carry on depends on the data it can access "locally"
    - A local LAN activity, with no geographical WAN consequences
  - Local data depends on its turn on how easy is to move data locally

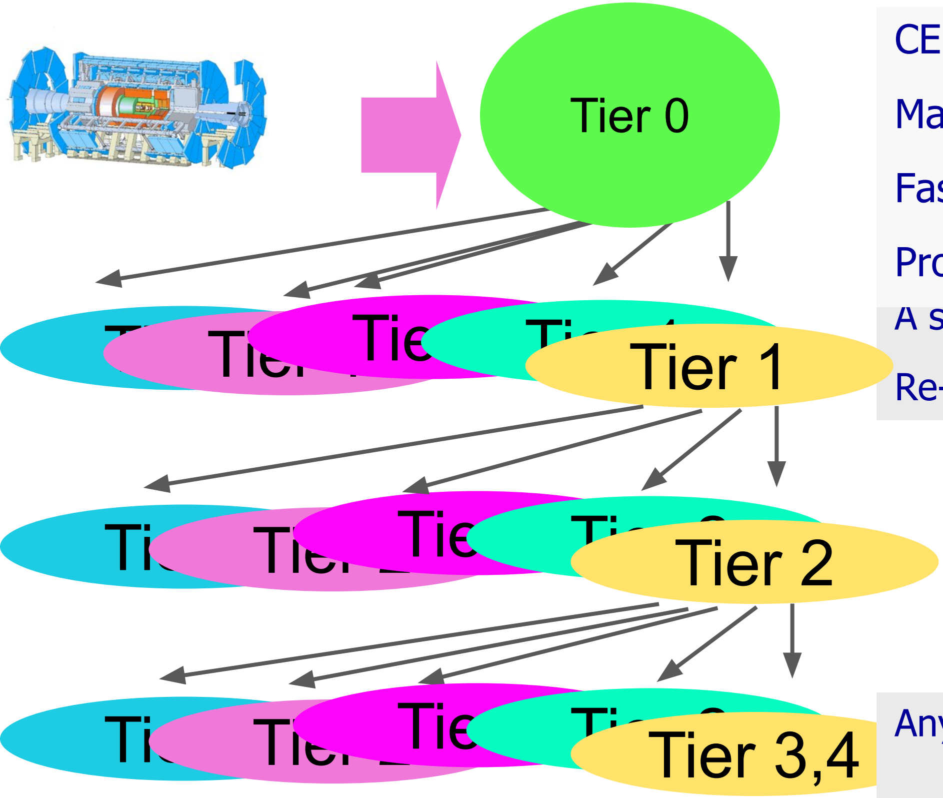
  - **MONARC Study group**

Outcome (early 2000s)…

- Distributed computing model, but in a hierarchic structure: hierarchy via "computing tiers"

- Hierarchical model: since (real) data originates at CERN, it must have a central role. Data will flow from it to the other sites, in a pyramidal structure
    - MC can in principle be generated in any place, but it will still need a central place for consolidation and traffic management





CERN

Master copy of RAW data

Fast calibrations

Prompt Reconstruction

A second copy of RAW data (Backup)

Re-reconstructions with better calibrations

Analysis Activity

They are dimensioned to help ~ 50 physicists in their analysis activities

Anything smaller, from University clusters to your laptop

Other effects of being Data Driven

- **Ideal GRID**: if I need to process computational tasks (“jobs”), I will do it on sites where there are some available CPUs. They will access data transparently via the network
  - This is BAD: this makes data paths not predictable. We cannot do it

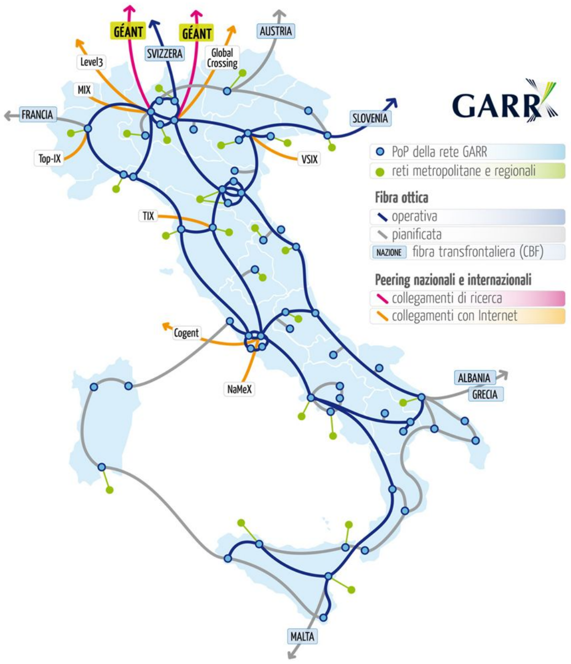
- Hierarchical GRID model (“DataGRID”)
  - Jobs **just access local data** (local = already present in the same site/ cluster/ building)
  - … but someone must have **preplaced** the data there!

What about T1-T1 and T1-T2?

- Nothing guaranteed, just based on what National Research Networks (NREN) were providing
  - no network provisioning: LHC traffic is just like any other research traffic

- For Example, in Italy our NREN is called GARR (Gruppo Armonizzazione Reti della Ricerca)

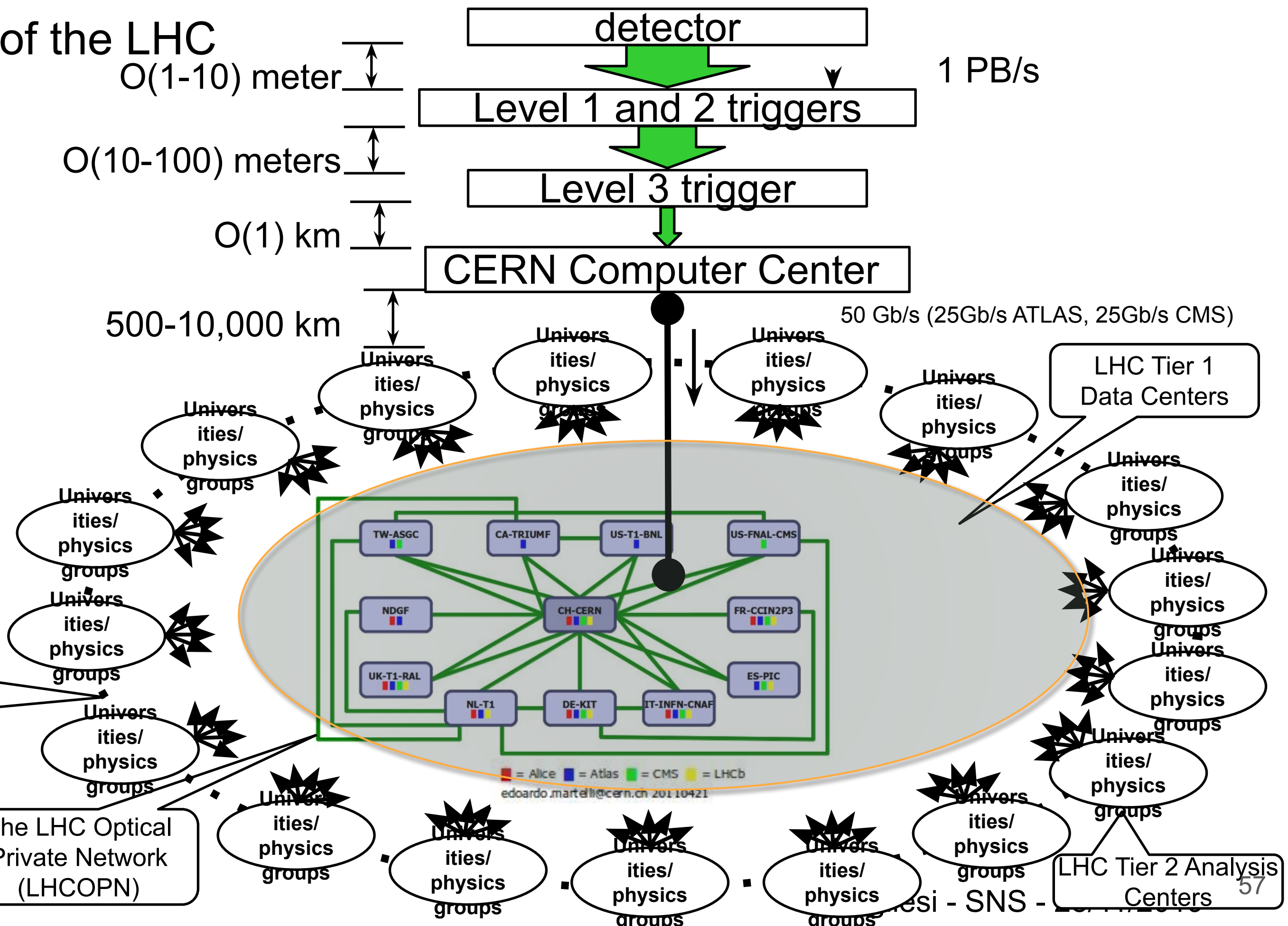
- The full LHC network topology is the following:



Enabling this scale of data-intensive system requires a sophisticated network infrastructure

A Network Centric View of the LHC

CERN →T1	miles	kms
France	350	565
Italy	570	920
UK	625	1000
Netherlands	625	1000
Germany	700	1185
Spain	850	1400
Nordic	1300	2100
USA – New York	3900	6300
USA - Chicago	4400	7100
Canada – BC	5200	8400
Taiwan	6100	9850



This is intended to indicate that the physics groups now get their data wherever it is most readily available

So we have the Computing Model infrastructure

- We have GRID(s), we defined MONARC
- We have ~50x4 Computing Centres (the “Sites”)
- What defines a working system, which needs to have
  - Uniformity in the computing environment
  - Uniformity in the access protocols
  - Support for operations…

- We need a Worldwide coordination

For example, GRID projects

- Are more than a few, in principle each with a different interface, Middleware …

CrossGrid

the globus project™
www.globus.org

Condor
High Throughput Computing

GriPhyN
Data Intensive Science

Nordic Testbed for Wide Area Computing and Data Handling

egee
Enabling Grids for E-science in Europe

iVDgL

PPDG

egee
Enabling Grids for E-sciencE

Open Science Grid

DataTAG

GEANT

WLCG as the orchestrator

- "GRID" is a computing paradigm
- WLCG governs the interoperation since 2002 between the number of "concrete GRID implementations" (a number of, the main ones being OSG, LCG, NorduGrid, …)
- WLCG was crucial in planning, deploying, and testing the infrastructure before 2010, and is crucial for operations now

As of today, from REBUS

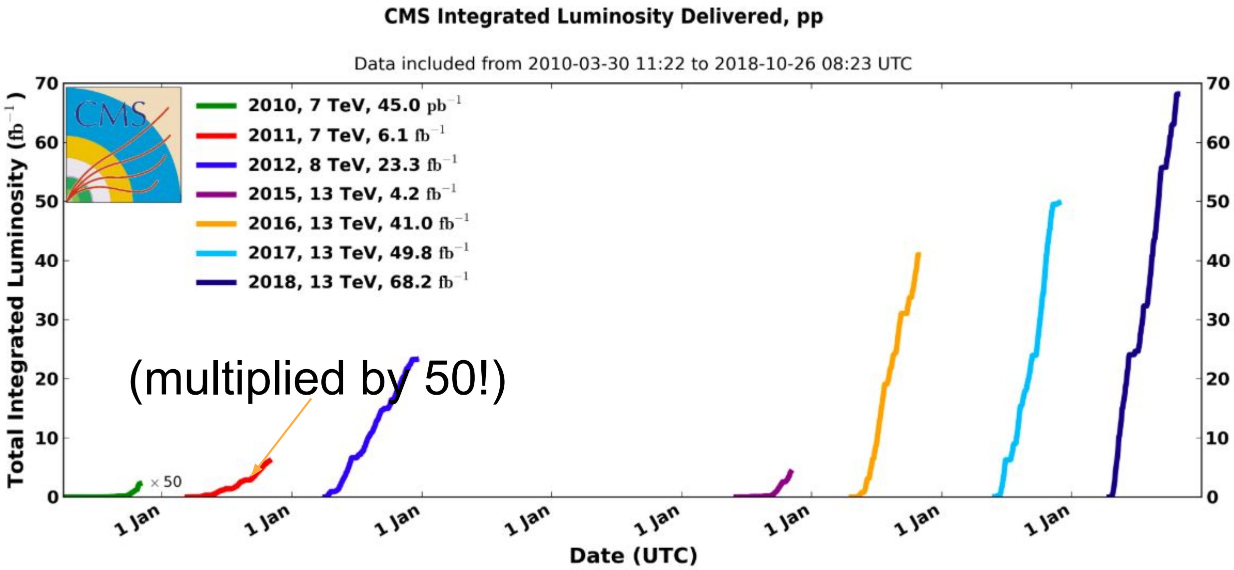
- **CPU 6 MHS06 (~600k computing cores)**
- **DISK 550PB (~80k HDDs)**
- **TAPE 800 PB (80k tapes)**
- **# Sites exceeding 200**

Summary on computing models

- We defined the amount of resources needed for LHC computing
- We decided where to deploy them, with which structure
- We have computational activities, and we defined where in the structure to perform them
- This needs organized data moving activities

- That is the 1995-2005 model, where are we now?

LHC Runs since then

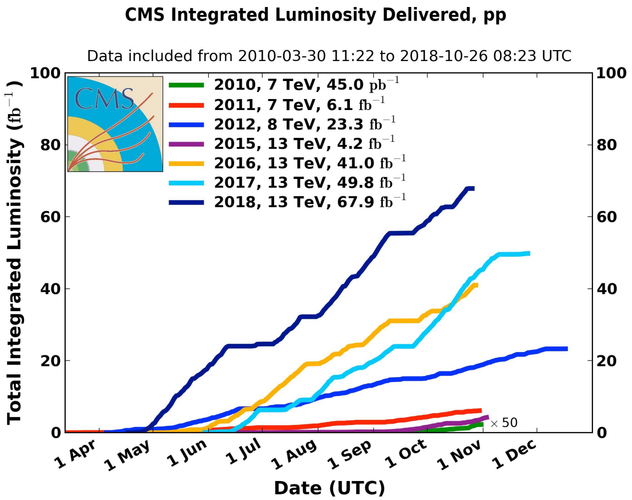


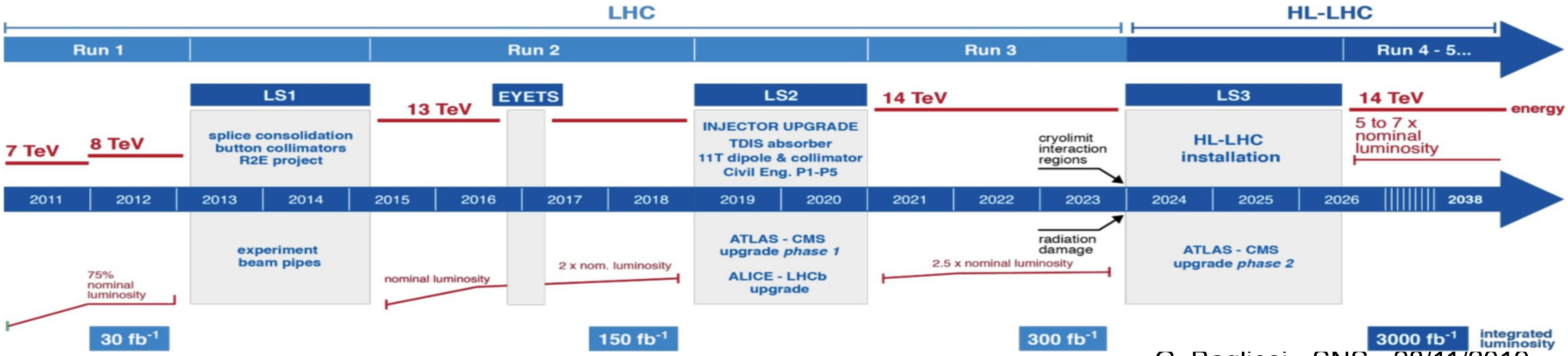
(multiplied by 50!)

“RunI” “RunII”

Current status of LHC (and CMS experiment)

- A peak lumi of $2x10^{34}cm^{-2}s^{-1}$ was achieved
- Run1 =~30/fb, Run2 =~163/fb
- Run2: ~35 pileup interaction per beam crossing
- Many analyses based on the full Run2 dataset are still ongoing

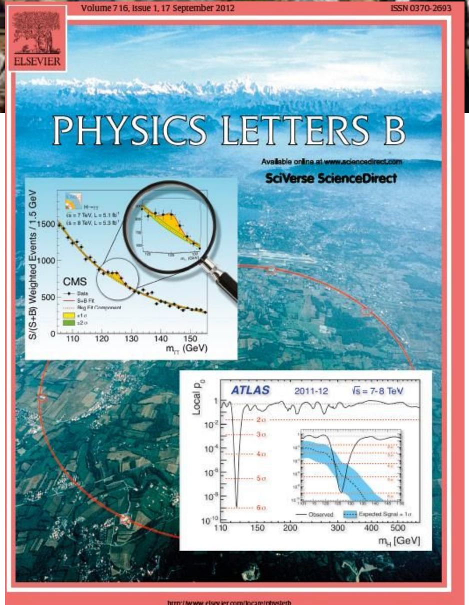




How did it work so far ?

- Handling LHC computing has surely been in these 9 years
  - **Fatiguing** (lots of manpower needed for services, support, data movement, job handling …)
  - **Complicated** (the system has a huge number of degrees of freedom, it is hard to optimize)
  - **Expensive** (200+ sites, XX Meur/y)
- … but it has lived up to Physicists' expectations
  - Jul 2010: first ttbar events shown in Paris, 72 hours after having been collected
  - Jul 2012: "Higgs discovery day", with data shown collected up the previous week
- By now, the LHC (4 exps) paper production rate is 1 paper/day!

Results…



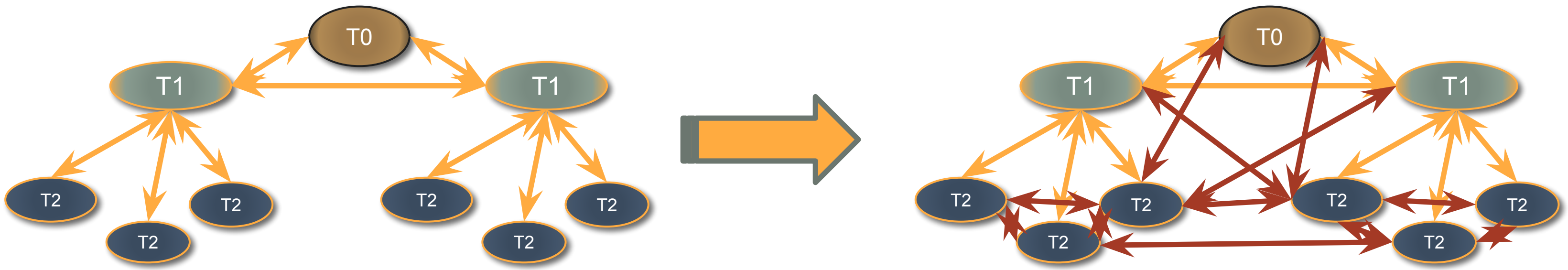
Volume 716, Issue 1, 17 September 2012
ISSN 0370-2693
ELSEVIER
PHYSICS LETTERS B
Available online at www.sciencedirect.com
SciVerse ScienceDirect
CMS
S/(S+B) Weighted Events / 1.5 GeV
$m_{\gamma\gamma}$ (GeV)
ATLAS
2011-12
$\sqrt{s}$ = 7-8 TeV
Local p_0
m_H [GeV]

In summary
We have observed a new
boson with a mass of
125.3 ± 0.6 GeV
at
4.9 σ significance

TIME
Nº5
PARTICLE
PHYSICIST
FABIOLA
GIANOTTI

Real operation mode today

- **Netflix, Spotify**, … → commercial commodity networks available at a lower price / larger bandwidth than expected (and yes, Pisa got that 1 Gbit/s by 2005!, >20 Gbits/s now...)
- No need to have strict hierarchical network paths, → full mesh: every site can transfer from any other



How to use the new network capabilities?

- **Direct Remote data access (a.k.a Streaming!)**

- You remember the problem with Data Driven: jobs go where data is
  - If a site has spare CPUs, but no data → not used
  - If a site has data, but no spare CPUs → jobs kept waiting

- If we remove the constraint of Data locality, match-making becomes very easy + efficient
  - Direct Remote Data Access: think of Youtube/Netflix!
  - You do not download the file, you access it over the network

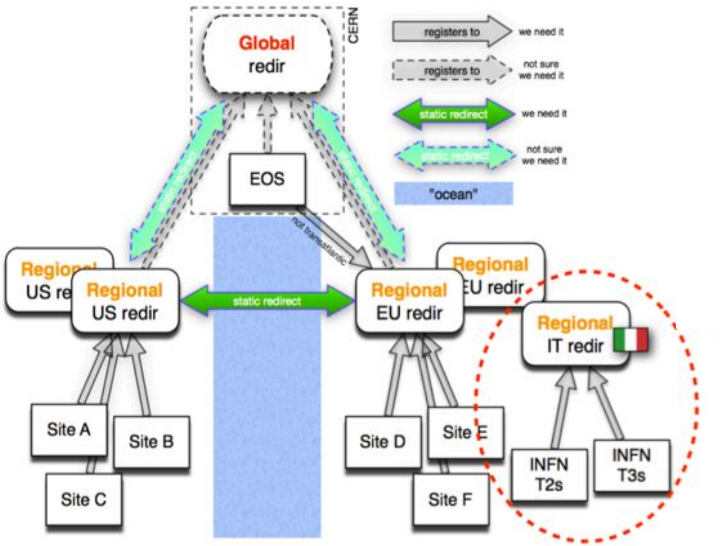
Storage Federations

- Imagine the scenario:
    - You put data anywhere (on any of the Sites serving the experiment)
    - Jobs go anywhere CPU is available
    - Jobs have to access data:
        - How? Via a remote access protocol
        - Where from? It would be better from a close place

- Storage Federations are a way to fake the existence of a single global storage system, and to implement priorities of access

Idea: hierarchic federations

- **(Examples: AAA in CMS...)**
  - **Any Data, Anytime, Anywhere**
- When a file is opened (POSIX *fopen*)
  - If the file is local (local storage), open it; otherwise
    - Ask your national redirector. If the file is found in your country, open it; otherwise
      - Ask you regional redirector. If the file is found in EU, open it; otherwise
        - Reach the top level redirector; if the file is found, open it, otherwise -> ERROR
- While all the files are accessible in this way, “cheap” transfers are tried at first
- It is NOT different than Netflix distribution model, after all…



The software (a small parenthesis)

- For the moment we focused on **HOW** to handle LHC computing at large scale
- We did not really clarify **WHAT** needs to be executed

- Small outline
  - Basic software workflows
  - Overall organization
  - performance is money! The eternal fight for performance

Basic SW workflows

- By workflow:
  - If you take today's share of Computing resources, you roughly get

1. ~40% spent on Monte Carlo simulation
2. ~30% reconstruction time (including Data and MC, and including the several reconstruction passes)
3. ~30% analysis activities

- While the first bullet is mostly **Geant4** processing time, on which we have not too many handles, the rest is software directly written by the Experiment

- How big/complex is it?

A case study: CMSSW (CMS Offline SW)

- **CMSSW on GitHub**
- **Started development** = early 2005 (superseding an older sw)
- **Core algorithms in C++;** some **Fortran** in externally provided routines, now gone for good; a lot of **Python** for steering and analyses
- A **single solution** for all the use cases
  - **Trigger (!)**
  - Reconstruction
  - Simulation
  - Analysis

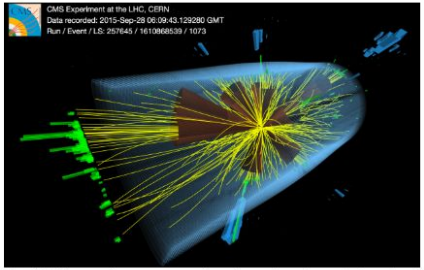
- Current size is 1120 packages, divided into 120 Subsystems

CMS Offline Software http://cms-sw.github.io/

hep | cern | cms-experiment | c-plus-plus

197,478 commits | 77 branches | 1,830 releases | 768 contributors | Apache-2.0

Welcome to CMS and CMSSW

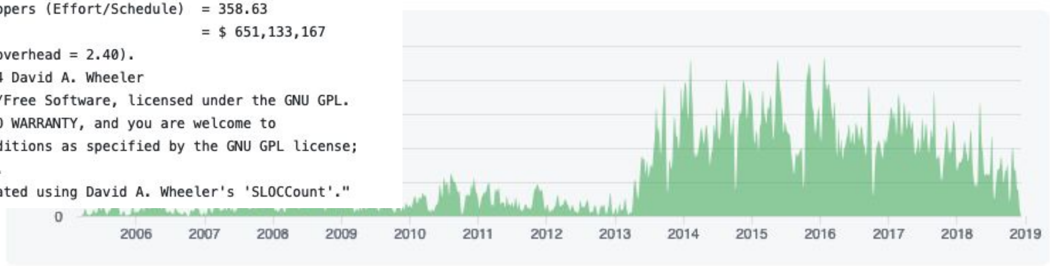



```
Total Physical Source Lines of Code (SLOC)                = 4,878,616
Development Effort Estimate, Person-Years (Person-Months) = 1,491.91 (17,902.87)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 8.61 (103.29)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 173.33
Total Estimated Cost to Develop                           = $ 201,536,212
```

The LHC smashes groups of protons together at close to the speed of light: 40 million times per second and with

```
Total Physical Source Lines of Code (SLOC)                = 14,905,968
Development Effort Estimate, Person-Years (Person-Months) = 4,820.12 (57,841.49)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 13.44 (161.28)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 358.63
Total Estimated Cost to Develop                           = $ 651,133,167
 (average salary = $56,286/year, overhead = 2.40).
SLOCCount, Copyright (C) 2001-2004 David A. Wheeler
SLOCCount is Open Source Software/Free Software, licensed under the GNU GPL.
SLOCCount comes with ABSOLUTELY NO WARRANTY, and you are welcome to
redistribute it under certain conditions as specified by the GNU GPL license;
see the documentation for details.
Please credit this data as "generated using David A. Wheeler's 'SLOCCount'."
```

ommits



Activity exploded for RunII!

Evolution of LHC Computing Models → Future

If it works (as we claim), why change it at all?

- LHC conditions are changing … faster than technology can absorb
- We have updated priorities now (we found the Higgs!)
- Run1+2 Experiments had limits (due to technology being not mature)
  - We can change it now!

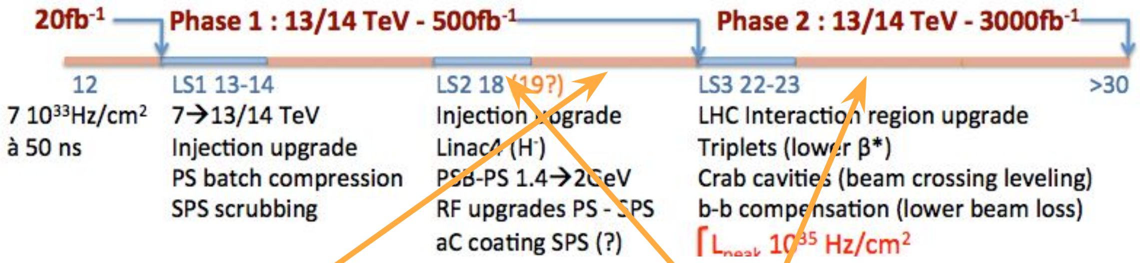
- BUT not to be forgotten: economical situation is **Much Different** now with respect to early 2000x

LHC 2013+

shutdown

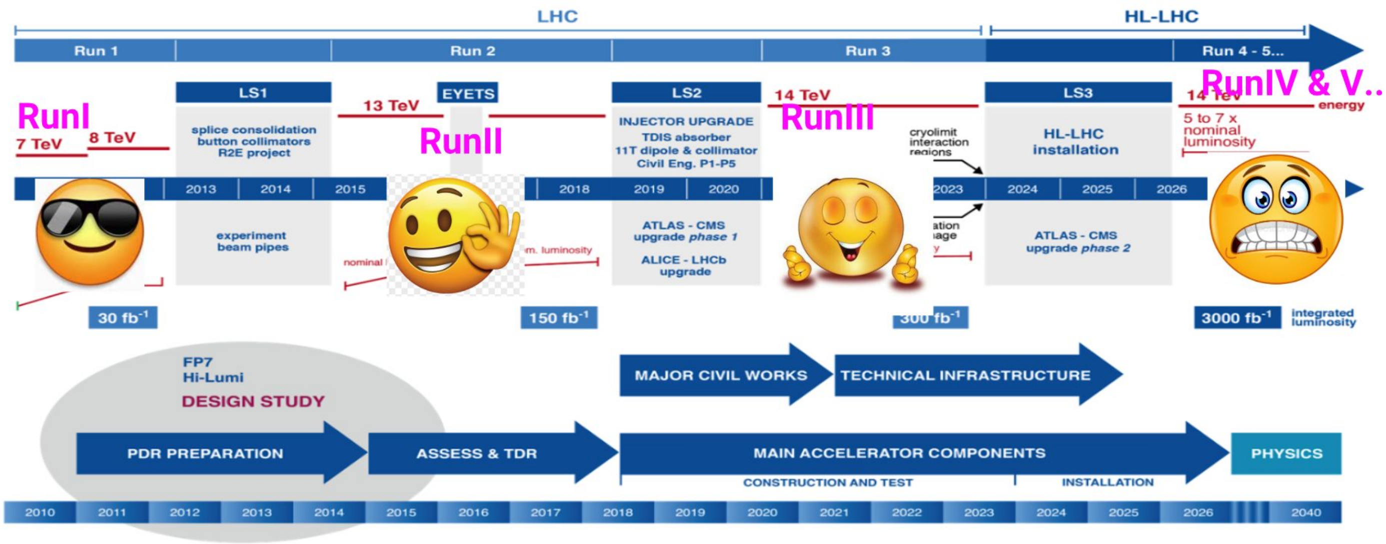
run

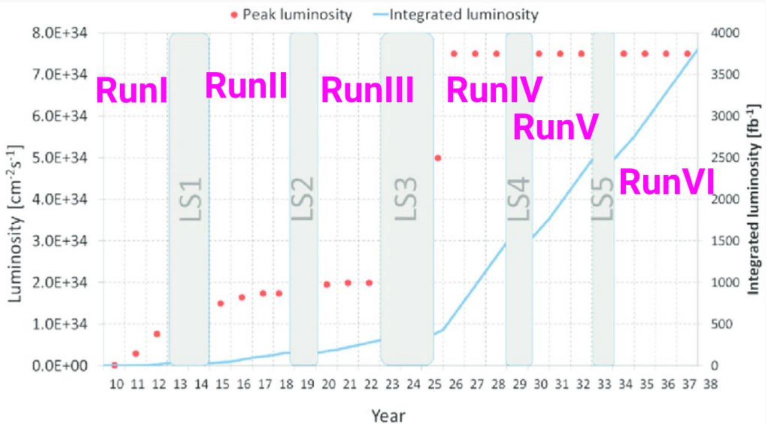
Run 1 Run 2 Run 3 Unknown territory…



We are here

- **2015-2018: 13 TeV, ~2.5x in luminosity, up to 3x in hadronic events per collision**
- **2021-2023: 14 TeV, again 2.5x in luminosity**
- **2026+: the so-called HL-LHC (or SLHC)**
- **2035+: still under discussion whether we will use LHC (improbable) or go for a completely new thing**





Some true but amazing statements:

- “We collected 5% of LHC foreseen integrated luminosity”
- “We are at 1/5th of the LHC machine capabilities”

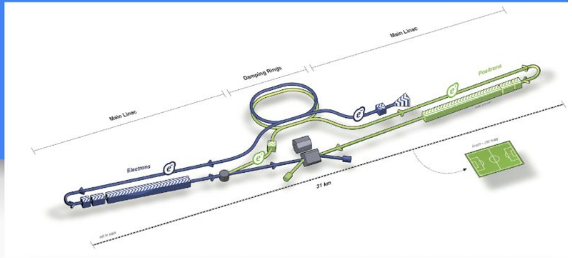
(to be clear: I am not even considering RunIII, it is just a “simple” extension of RunII for ATLAS and CMS - no tension)

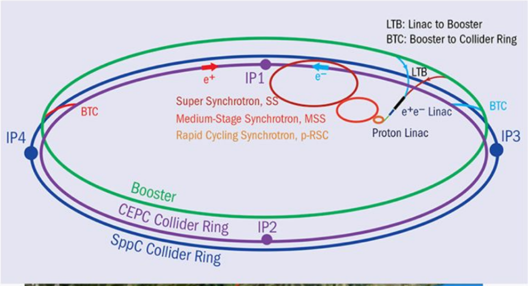
HL-LHC is not the end of the Story …

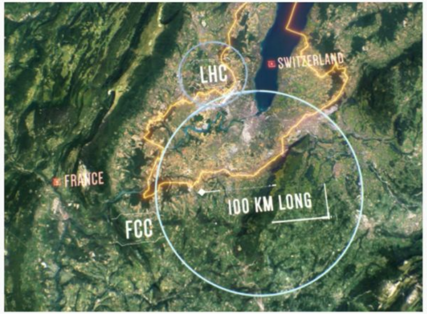
2035?

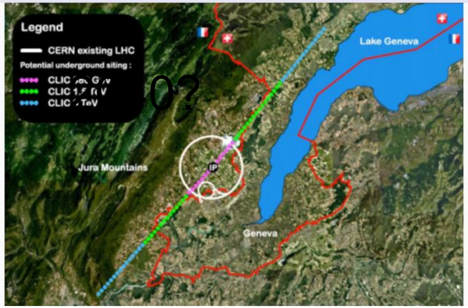
Beyond #1?

- ee machines (CLIC, ILC, FCC-ee,CepC ….)
  - **No major computing problem expected**
  - FCC-ee initial event size estimates are 0.01 - 0.1 the current LHC-pp, and 20 years later
  - Even a huge increase in DAQ channels / interaction rate can hardly be a problem





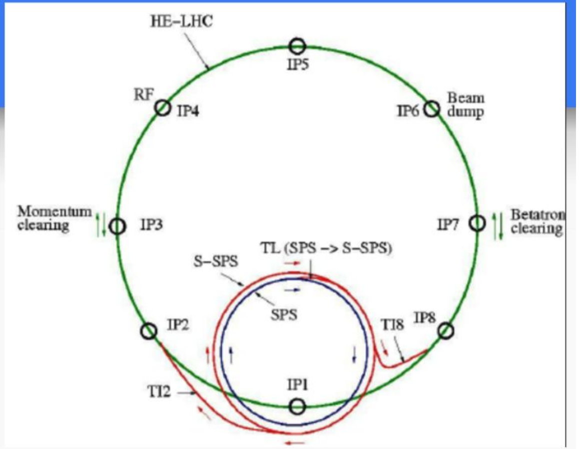




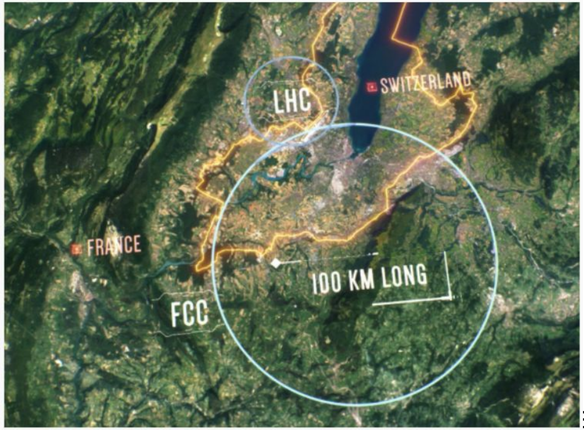
Beyond #2?

- hh machines (FCC-hh, HE-LHC, ...)
  - ...go as high as you want: FCC-hh has (wrt to current LHC)
    - <PU> ~30x (and 5x HL-LHC)
    - Similar collision rate
    - Event sizes not yet known atm
  - But: there is at least a +20y between them, which reduces the problem
  - **HE-LHC** parameters are intermediate between HL-LHC and FCC-hh, but time scale is still at least 2035

- **My thoughts: the step LHC→ HL-LHC in 2026 is the biggest; if we can make HL-LHC computing work, we have a clear path**



2035?

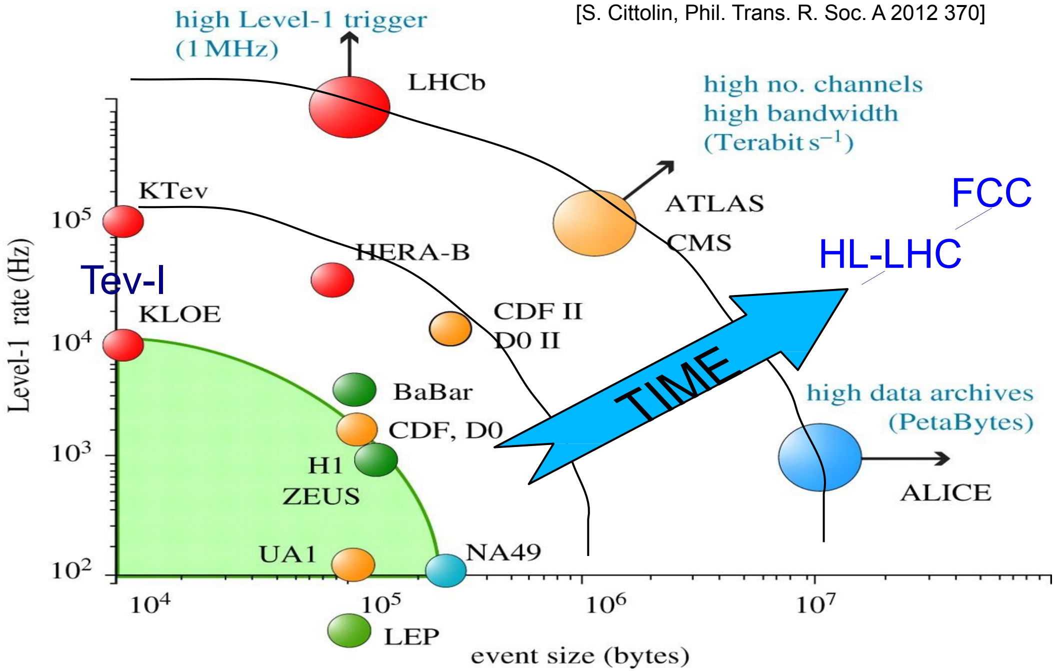


2045?

History of HEP Data-Processing in a plot

Complexity and computational load kept increasing, in pace with electronics advancements
→ Data processing still a major cost item of experiments
→ Often a major technical constraint

[S. Cittolin, Phil. Trans. R. Soc. A 2012 370]



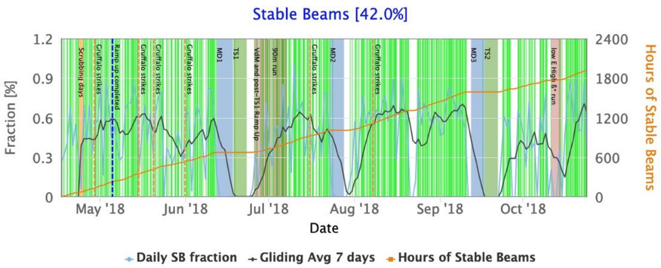
But: - Physics landscape now asking for more ***precision***

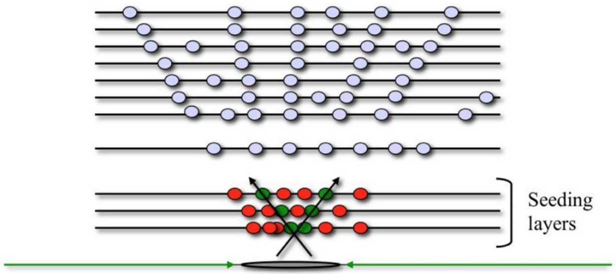
- Moore's law ***slowing down***

… symptoms that HEP might face a computing roadblock

Scaling LHC → HL-LHC

- Main Evolution of important computing parameters
  - Live time cannot change much; if anything can go much below
  - <PU> goes from 35 to 200
  - Trigger rate 1 kHz → 7.5 kHz
- **HL-LHC / LHC = (7.5/1) * (200/35) = 42**
- This is optimistic!
  - Triggers have to remain clean
  - Assumes all is linear with <PU>, while reconstruction has at least a superlinear component
  - Upgraded detectors, more DAQ channels
- **A more realistic educated guess is 50-100x**

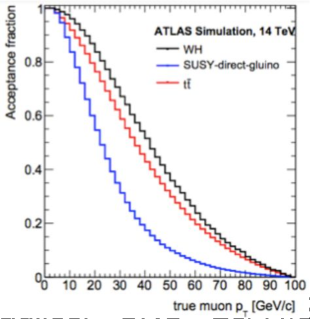




Trigger rate scales at best with $\mathcal{L}$ for

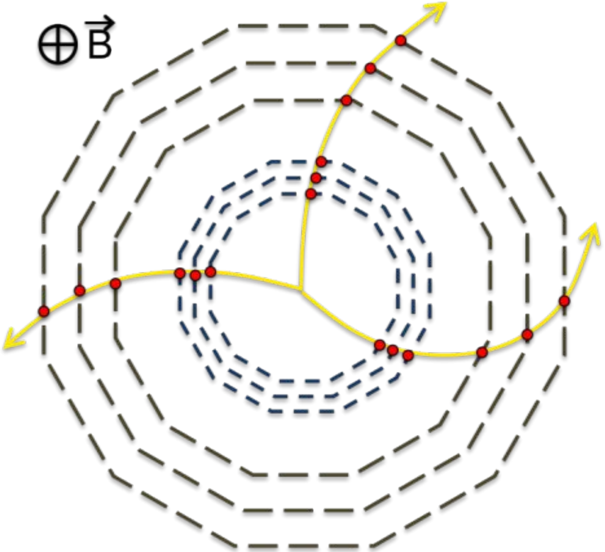
- Same physics
- Clean triggers

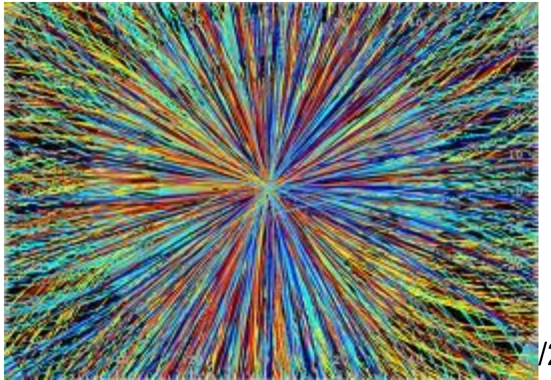
Difficult to do better than this



(parenthesis: why CPU more than linear?)

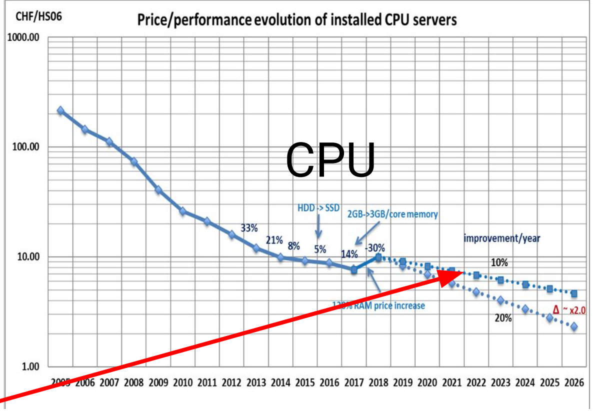
- It is simply a combinatorial effect, which enters in the most CPU consuming reconstruction algorithm: TRACKING
- In a quite naïve view Tracking is: “link the dots”
  - But we do have many “dots”!
- Strategy: find 2 hits which are compatible with forming an arc together with the interaction point, and a given momentum range
  - Propagate them and see if external links are found
- Just saying “**find 2 hits**” means it will scale quadratically:
  - 2x the hits → 4x processing time
  - This is called “**combinatorial explosion**”

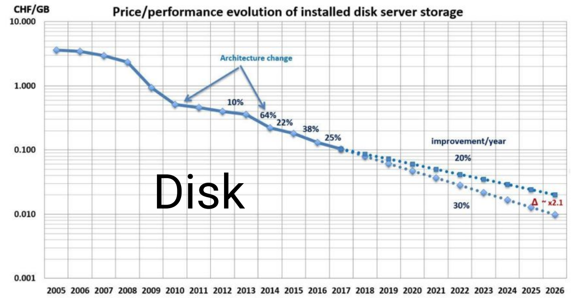


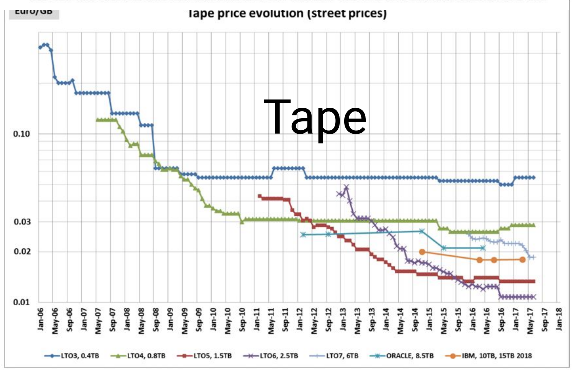


And in the meantime .

- The days of a +50% value per year from Moore (and similar) law are gone
- A +20%/y seems already optimistic, and there is even some indication of inversion of trend
- Even if we stick to +20%/y, 1.2^7 = 3.6:
  → natural technology evolution (also known as the "sit-and-wait" approach) is not going to help us.
- **50-100x → 14-30x taking into account technology**
- **We need real and furious R&D**
- **7 years are not that much!**







The once trusted "sit-and-wait" approach: do nothing, Intel will solve your problems

Empirical laws

- A few empirical laws are common when trying to predict the costs of resources with time:
  - **Moore's law**: The number of transistors on integrated circuits doubles approximately every two years". This can be translated into "every two years, for the same money, you get a computer twice as fast";
  - **Kryder's law**: "the capacity of Hard Drives doubles approximately every two years";
  - **Butter's law of photonics**: "The amount of data coming out of an optical fiber doubles every nine months";
  - **Nielsen's law**: "Bandwidth available to users increases by 50% every year.

  - **.. All not realistic any more …**

Summary on future experiments

- Future (today +10y) HEP experiments do not have an easy path to computing
  - A simple extrapolation of today's models diverges financially by a factor >10x in the next 10 years
- If this is to remain true, the computing would cost more than the accelerator and the experiments
  - A no-go from funding agencies
- **Which are the solutions / paths we can try to follow towards a mitigation of the problem?**

A non final list of improvements to pursue

1. **Infrastructure** changes
2. **Technological** changes
3. **Physics** #1: change analysis model
4. **Physics** #2: reduce the physics reach (for example increasing trigger thresholds)
   - Not even considered here … it is the “desperation move” if we fail with everything else
5. Use “modern weapons”
   - Big Data, Machine Learning, …
7. Something **unexpected**...

Infrastructure changes

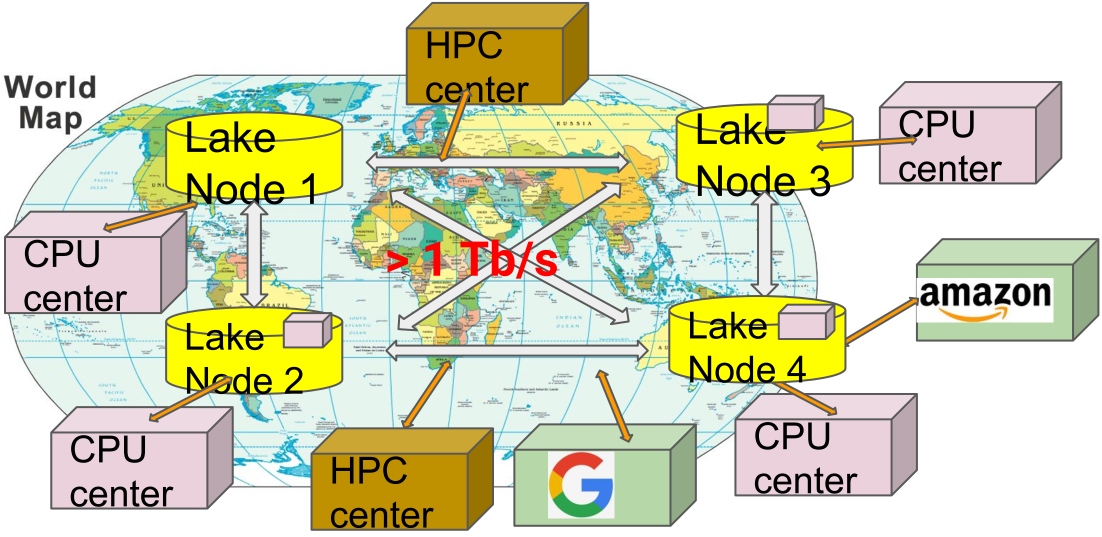
- Today's HEP computing
  - Owned centers, long lifetime (10+ y)
  - Well balanced in storage vs CPU
  - FAs pay for resources + infrastructure + personnel

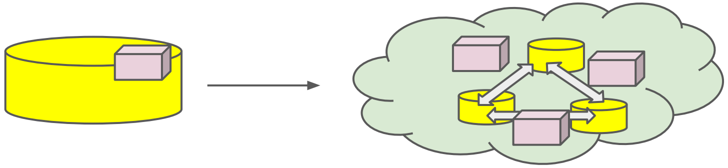
Is it the most economic computing you can buy today?

- **YES**, if you care about your data safety (and your capability to access it)
- **NO**, if you can use opportunistic resources
  - They come and go fast
  - You can hire them (from a commercial provider, ...)
  - You can use "someone else" resources

The “data lake” model

- Keep the real value from the experiments safe
  - (RAW) **data** and a solid baseline of **CPU** in owned and stable sites
  - Allow for multiple CPU resources to join, even temporarily
    - Eventually choosing the cheapest at any moment
  - Solid networking: use caches / streaming to access data
- Reduce requirements for Computing resources
  - Commercial Clouds
  - Other sciences’ resources
    - SKA, CTA, Dune, Genomics, ...
  - HPC systems





DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

ProtoDune 2-3 GB/s (like CMS); Real Dune 80x

SKA SQUARE KILOMETRE ARRAY

SKA up to 2 PB/day

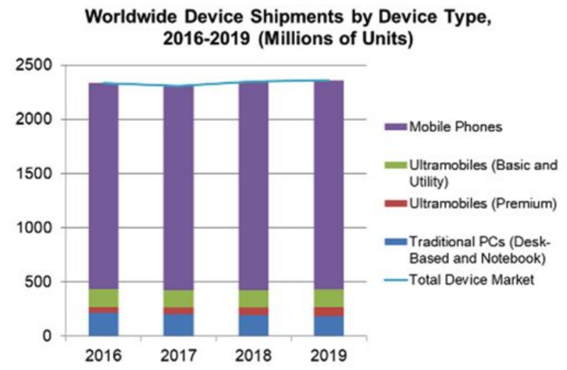
A single genome ~ 100 GB. A 1M survey = 100 PB

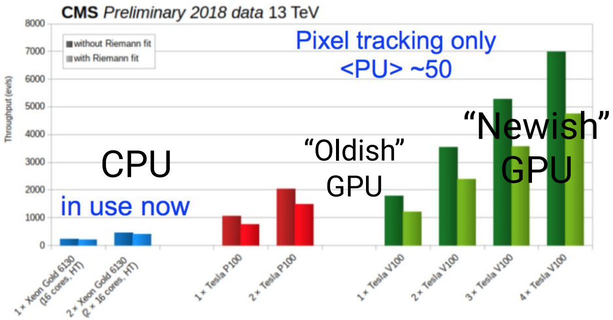
cta

CTA projects to 10 PB/y

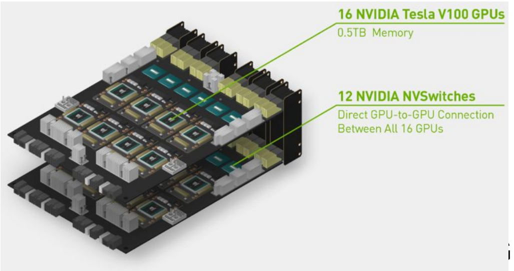
Technology changes

- Use the **cheapest technology per $.** It used to be Linux PCs, now it is
  - Mobile (low power) processors
  - Vector processors (“GPGPUs”, “TPUs”)
  - Code-in-hardware (“FPGA”, “ASIC”, …)
- Can we use them?
  - Not easily - limited to mission critical algorithms
  - We need a way not to write the code once per platform
  - We need frameworks to embrace <u>Heterogeneous Computing</u>





Low power (running cost /4)



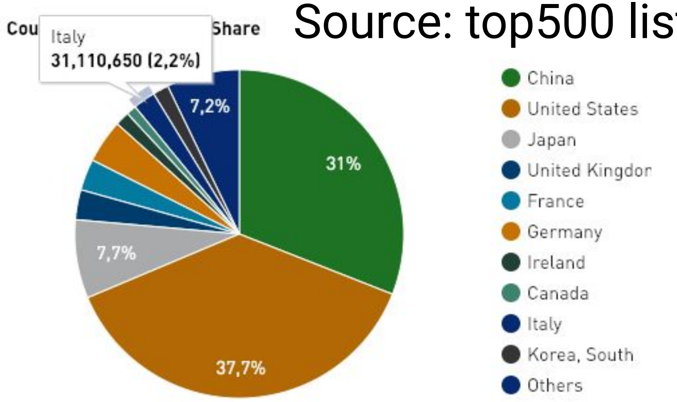
High performance

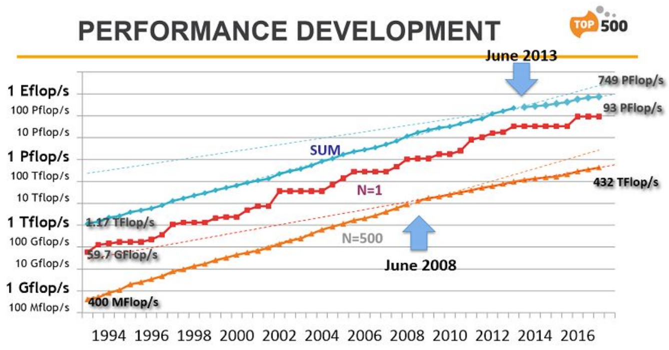
Supercomputing (HPC)

- The world is literally full of Supercomputers. Why ?
  - Real scientific use cases
    - Lattice QCD, Meteo, ...
  - Industrial showcase
    - And hence not 100% utilized, opportunities for smart users. Can we be one of them?
- Many not trivial problems to solve:
  - **Data access** (access, bandwidth, ...)
  - **Accelerator** Technology (KNL, GPU, FPGA, TPU, ???, …)
  - **Submission of tasks** (MPI vs Batch systems vs proprietary systems)
  - **Node configuration** (low RAM/Disk, …)
  - **Not-too-open environment** (OS, …)

- Some hint of global slowing down, but not for top systems where the “war” is on

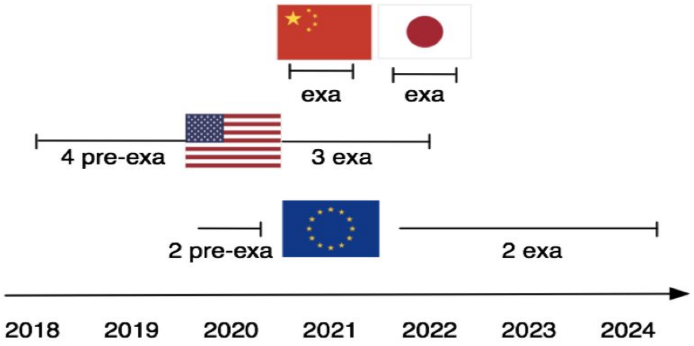
Source: top500 list

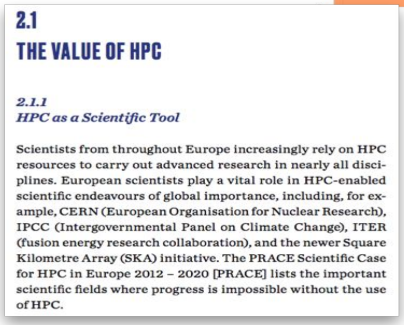




Supercomputing - the expected future

- The race will go on, at least between major players
- EU wants to enter the game - never a the top in the last 25y
- Next big thing is **ExaScale** (10^{18} Flops - operations per second)
  - Should be well available by HL-LHC
- Somehow difficult to compare, technologies / benchmarks, but
  - LHC needs today the equivalent of ~30 PFlops
  - A single Exascale system is ok to process 30 “today” LHC
  - **Scaling: a single Exascale system could process the whole HL-LHC with no R&D or model change**
- Some FAs/countries are explicitly requesting HEP to use the HPC infrastructure as ~ only funding; **it is generally ok IF we are allowed to be part in the planning (to make sure they are usable for us)**

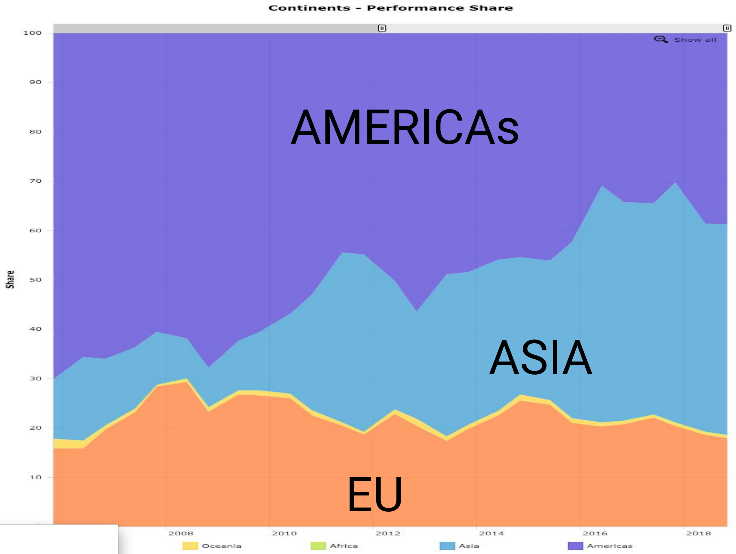




2.1
THE VALUE OF HPC

2.1.1
HPC as a Scientific Tool

Scientists from throughout Europe increasingly rely on HPC resources to carry out advanced research in nearly all disciplines. European scientists play a vital role in HPC-enabled scientific endeavours of global importance, including, for example, CERN (European Organisation for Nuclear Research), IPCC (Intergovernmental Panel on Climate Change), ITER (fusion energy research collaboration), and the newer Square Kilometre Array (SKA) initiative. The PRACE Scientific Case for HPC in Europe 2012 – 2020 [PRACE] lists the important scientific fields where progress is impossible without the use of HPC.



US: apparently no way to have a say
EU: ETP4HPC has at least “asked for HEP position”
China: (no way)^2

Our computers up to now

- We use pretty standard out of the shelf computers
- Today you can buy for ~5000 Euros
  - 96 computing cores (x86_64)
  - 256 GB RAM
  - 2-4 TB SSD disk
- On this, we use to run 96 single processes

A “thing” like this is
- ~1000 HS06
- Consumes 1 kW + 500 W for cooling
- Has a lifetime of 3 years
- It costs ~4 kEuro on power in these 3 years

What’s the current direction of high performance computing?

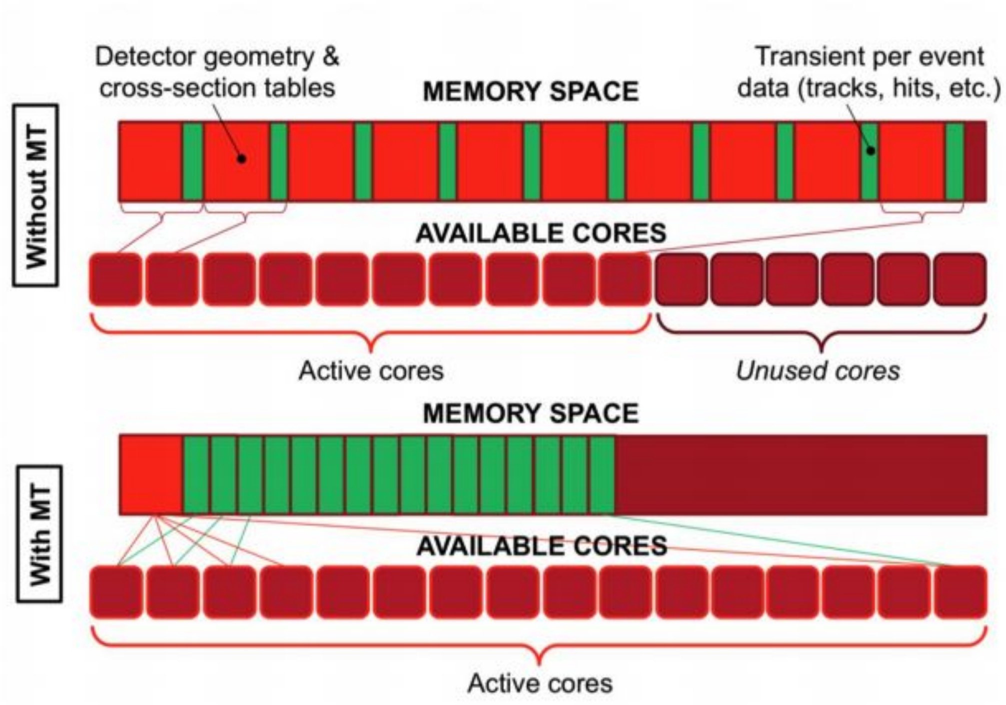
1. Multicore processing: treat one such machine as a single job instead of 64 distinct machines

2. High performance vector units: Xeon Phi, GPGPU, FPGA, …

3. Low power architectures (ARM…)

- Let’s say a few words on them

Multithreading: general concept

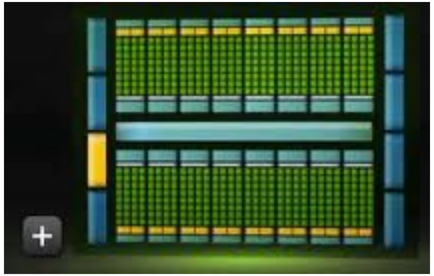


Geometry, calibrations,… (usually valid for many contigous events)

One event in memory (the DAQ channels)

New Architectures

- Massively parallel CPUs are with us since at least 5 years
  1. General Purposes Graphical Processing Units (GPGPU)
     - Video games oriented Graphics Cards recycled as Vector machines
     - Up to 5000 cores per board
     - Vector processing = they are only able to repeat the same operation on multiple data (Single Instruction Multiple Data = SIMD)
- Very powerful, but SIMD is limited to very specific applications (matrix multiplication … and eventually particle propagation)

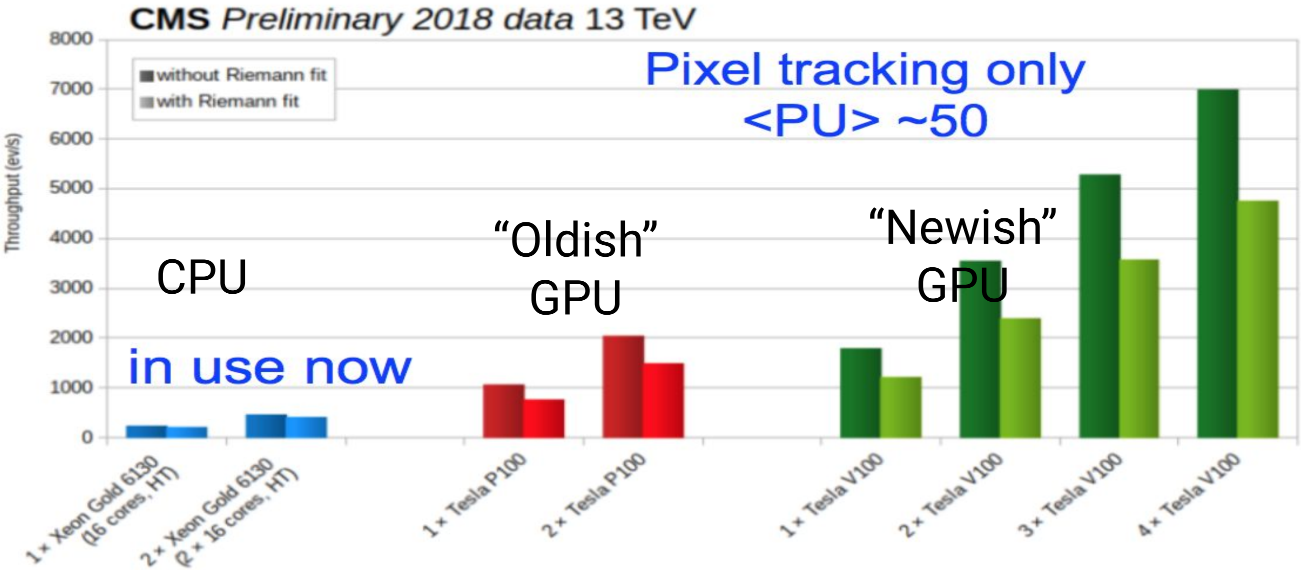


But beware:

- Very power hungry
- This kind of performance just for very specific use cases
- Very difficult to program

A more realistic estimate

- CMS Tracking in silicon tracker



Xeon Phi - KNL

- Concept:
  - put many low power, low dissipation cores together
  - Put a good interconnect
  - Put memory close

Essentials Export specifications

Product Collection	Intel® Xeon Phi™ x200 Product Family
Code Name	Products formerly Knights Landing
Vertical Segment	Server
Processor Number	7250
Status	Launched
Launch Date ?	Q2'16
Lithography ?	14 nm

Performance

# of Cores ?	68
Processor Base Frequency ?	1.40 GHz
Max Turbo Frequency ?	1.60 GHz
Cache ?	34 MB L2
TDP ?	215 W
VID Voltage Range ?	0.550-1.125V

ARM

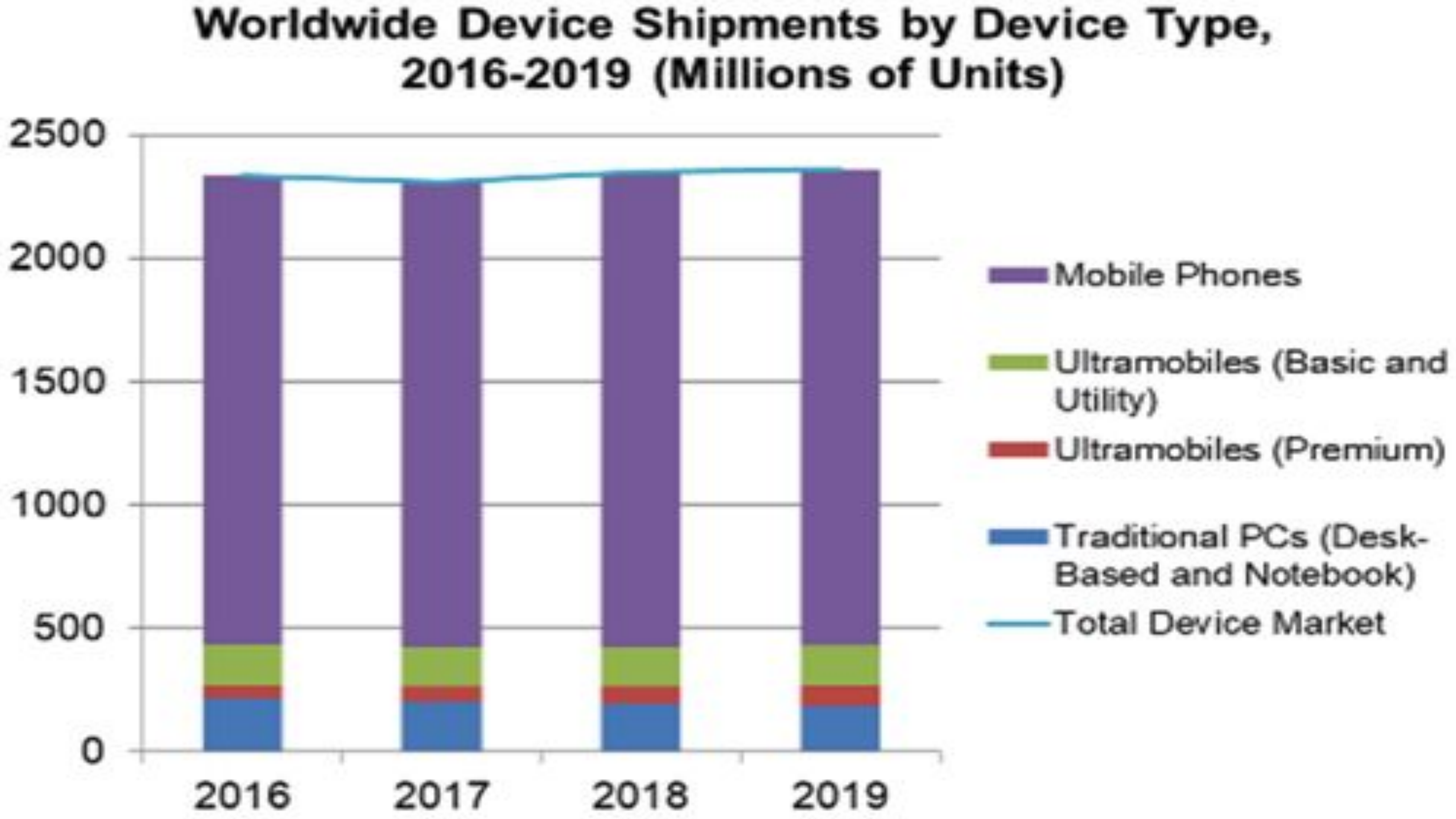
- A low power architecture (so attacks the price problem from another side)
- Still much less performing than x86_64 (at least a factor 4 less)
- But per Watt, a factor 4 better!

	Type	Cores	Power	Events/ min/core	Events/ min/Watt
ARM	Exynos4412 Prime @ 1.704GHz	4	4W?	1.14	1.14
x86	Xeon L5520 @ 2.27GHz	2x4	120W?	3.50	0.23
x86	Xeon E5-2630L @ 2.0GHz	2x6	190W?	3.33	0.21

CMS test (ARM vs x86) with simulation (Geant4)

- Events/core/min still worse
- But Events/min/Watt largely better
  - Ev/min/W ~ Ev/Joule!

- Would allow construction of much cheaper computing centres
  - Much less in $$ per power bill
  - Much less cooling infrastructure


Worldwide Device Shipments by Device Type, 2016-2019 (Millions of Units)
2500
2000
1500
1000
500
0
2016
2017
2018
2019
Mobile Phones
Ultramobiles (Basic and Utility)
Ultramobiles (Premium)
Traditional PCs (Desk-Based and Notebook)
Total Device Market

But what about Algorithms!

- A large impulse to a viable computing can come from better algorithms
  - Better: essentially faster either due to the use of new tools (Map&Reduce, Spark) or to the new of new concepts (Machine Learning)
  - Better: with also better physics performance, but less relevant here
- How?
  - Physicists already spent 20+ y to optimize their algorithms, no new ground breaking idea ..
  - We need something completely new

Reduction facilities / analysis farms

- Up to now our code was essentially sequential, with user writing stuff like
- This is (on purpose!) very fortran like; there are new technologies available which move from «describe how to do stuff» to «describe what you want to do»
- Examples: Map&Reduce, Apache Spark, Pig, …

```
events = load_events()
for ievent in events:
        do_something(ievent)
        do_something_else(ievent)
        accumulate_results(ievent)
Do_final_stuff()
Show_results()
```

Idea is…

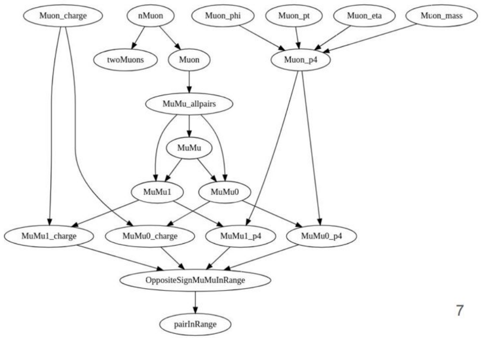
- Write an high level description of what you want to do (even in the form of graphs)
- Let the «compiler» understand which is the best technology to process a given data in a given place
  - An Hadoop enabled site → use Apache
  - A GPU enabled site → use tensorflow implementation
  - Scale out on the GRID if there are 10000 cores available

```
object muonsVeto
  take Muon
  select pt > 5
  select |eta| < 2.4
  select softId == 1
  select miniPFRelIso_all < 0.2
  select |dxy| < 0.2
  select |dz| < 0.5

# jets - no photon
object AK4jetsNopho
  take AK4jets j
  reject dR(j, photons) < 0.4 and
           photons.pt/j.pt [] 0.5 2.0
```

```
# EVENT SELECTION

cut preselection
# Pre-selection cuts
  select MET.pt > 200
  reject cleanmuons.size > 0
  reject verycleanelectrons.size > 0
  select jetsSR.size >= 2
```



What is the difference

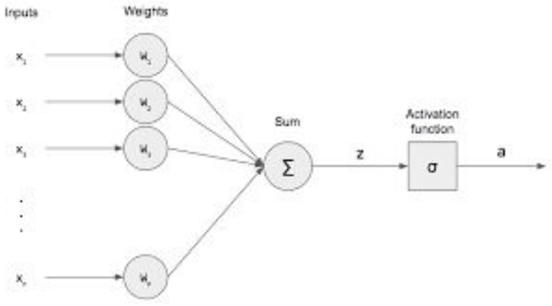
- Clearly this is not finding new resources, it is just trying to use better what we have
  - Matches better the underlying hardware, which can be very different – without users needing to know
  - Can change the percepted behaviour of the system
- Grid/Cloud: it is a container ship
  - Process many items at the same time, but the shipping time for a given item cannot be made faster
- Reduction facilities: easier to steer more resources to a single use case
  - High priority tasks can overtake a large fraction of the system

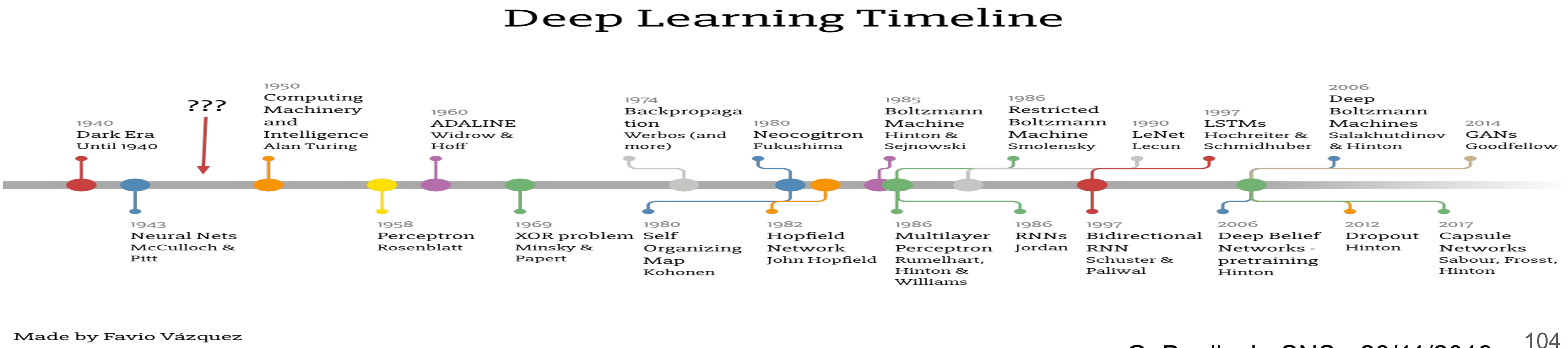
«These 3000 analysis tasks will be done in 5 days»

«In the next 5 days you will get an analysis done every 30 sec»

Machine Learning: it is not a new idea

- Overall:
  - Idea from the 40s (Turing, Pitt)
  - Perceptron (1957) as the building bloch, mimics a neuron
  - Explosion → 1990



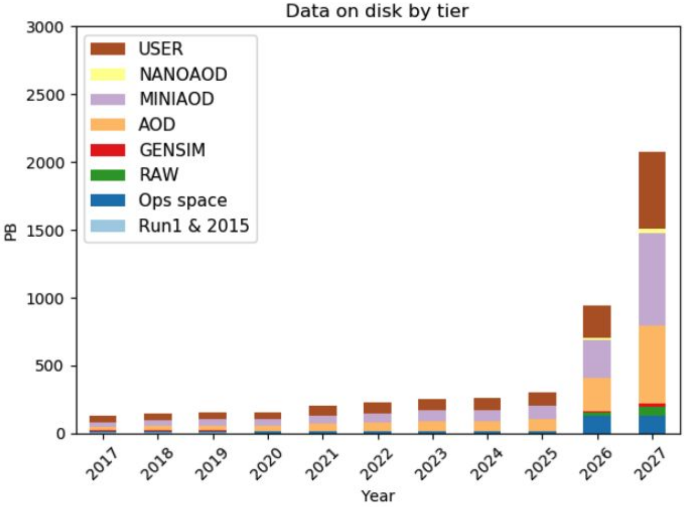


Summary: developments

- In order to cope with the Computing needs of the next decade(s) – at the ExaScale, we will need to abandon our comfortable model with GRID + Intel CPU + «fortran like» code + «physicist written» algorithms
- The adoption of these new technologies can be painful, and requires training on physicists' side
    - Fortran → C++ was not an easy task …
- **Still, there is confidence that the solutions can bring to an affordable HL-LHC Computing, and pave the way for later experiments**

Some more recent extrapolations (already better than the 50-100x !!!)

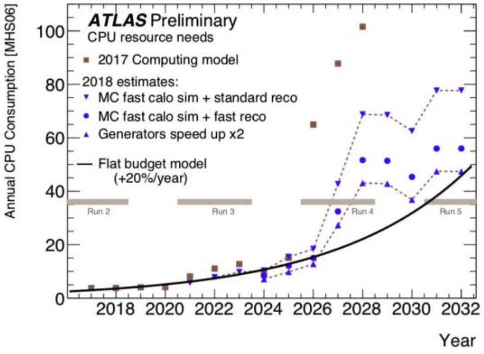
CMS preliminary



- **CMS needs @ 2027:**
  - **CPU: 44 MHS06**
  - **Disk: 2.2 EB**
  - **Tape: 3 EB**
- **(with respect to 2019 pledges, these are 22x, 13x and 15x)**
- **If you factor in ~4x from Moore's law, we are ~ 3x off**
- **Very recent CMS extrapolations → factor 2x**

And these do not include yet any extrapolation on the use of new stuff (GPU, ML, …)

- **ATLAS CPU needs @ 2027:**
  - **CPU: 30-70 MHS06**
  - **Disk: 1-4 EB**
  - **Tape: 3 EB**
- **ATLAS starts higher than CMS in 2019: so it is easier**

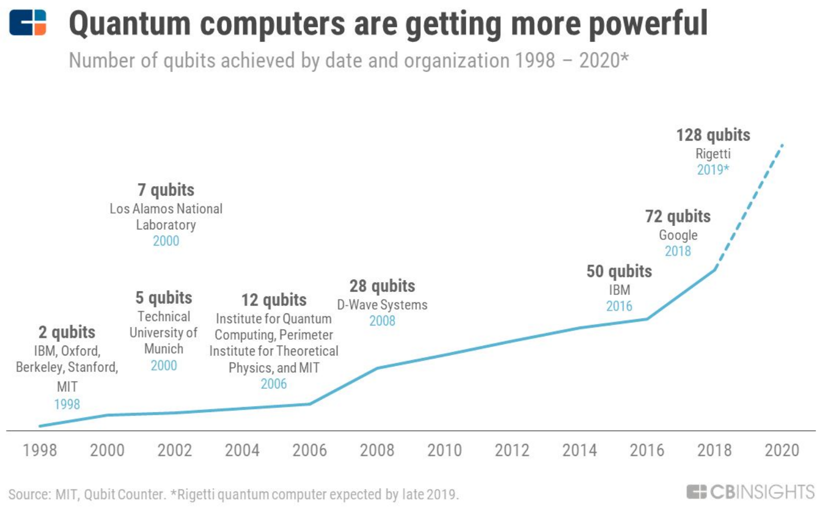


Solutions for future ?

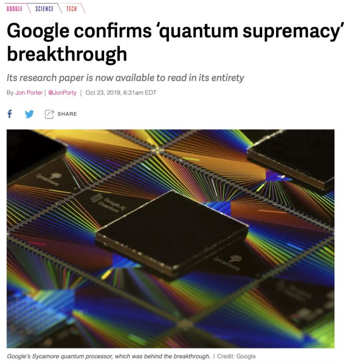
- We do not have an handy solution for 2026+ LHC computing
- But R&D is furious in all the directions
  - Modelling needs
  - Looking into new hardware solutions
  - Looking into new programming paradigms
- Today (T_0 – 7y) we see clear paths to the solution
  - If we would update our figures including what we assume we will be able to do with GPU, the compute problem could even be solved
  - Still work to be done on storage

Something completely new?

- Up to now
    - Evolution with optimizations (do the same things slightly better)
    - Some more radical changes (use GPUs, HPCs, special processors, …)
- Isn't anything completely different on the market?
    - → Quantum Computing!
    - Uses superposition of states to allow for multiple transformations at the same time (very very naively, a N qubit QC can explore the same phase space of a classical 2^N bit computer)
    - Is it real today? **No** (apart from the labs and for some specifically designed tests)
    - Is it coming? **Most probably yes**



The situation is slightly worse than what these numbers show: usually the qubits stay coherent for a very small amount of time, and errors are not negligible



But: "quantum supremacy" announced recently by Google...

But you cannot underestimate the trend (which come from technology improvements) to reach the ~1000 qubits in ~10 y

Quantum supremacy

From Wikipedia, the free encyclopedia

Quantum supremacy is the potential ability of quantum computing devices to solve problems that classical computers practically cannot.[1] Quantum advantage is the potential to solve problems faster. In computational-complexity-theoretic terms, this generally means providing a superpolynomial speedup over the best known or possible classical algorithm.[2] The term was originally popularized by John Preskill[1] but the concept of a quantum computational advantage, specifically for simulating quantum systems, dates back to Yuri Manin's (1980)[3] and Richard Feynman's (1981) proposals of quantum computing.[4]

QC for HEP …

- We cannot currently count on it to solve our problems…. But we can keep our **eyes open for opportunities**!
- Quantum Computing **could** become relevant for the next experiment after HL-LHC; we are the perfect users (we have a use case not easily solvable with standard means)

CERN openlab

Quantum Computing for High Energy Physics workshop

5 Nov 2018, 08:30 → 6 Nov 2018, 18:50 Europe/Zurich

500-1-001 - Main Auditorium (CERN)

Federico Carminati (CERN)

nature
International journal of science

Solving a Higgs optimization problem with quantum annealing for machine learning

Alex Mott, Joshua Job, Jean-Roch Vlimant, Daniel Lidar & Maria Spiropulu

"We show that the resulting quantum and classical annealing-based classifier systems perform comparably to the state-of-the-art machine learning methods that are currently used in particle physics[9,10]. However, in contrast to these methods, the annealing-based classifiers are simple functions of directly interpretable experimental parameters with clear physical meaning…"

DAILY NEWS 8 January 2019

IBM unveils its first commercial quantum computer

Supervised learning with quantum enhanced feature spaces

Vojtech Havlicek[1,*] Antonio D. Córcoles[1], Kristan Temme[1], Aram W. Harrow[2], Abhinav Kandala[1], Jerry M. Chow[1], and Jay M. Gambetta[1]
[1]*IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA and*
[2]*Center for Theoretical Physics, Massachusetts Institute of Technology, USA*
(Dated: June 7, 2018)

QC and HEP

- Three possible interaction domains we are working on

1. Quantum Simulators replacing part of the MC generators
    - Impose the QCD / SM hamiltonian to a quantum system, and let it evolve → get events to be used in simulation
2. A generic minima finding tool
    - On paper much faster as the # of dimensions increase
    - Most of our algorithms could be rewritten as a likelihood / chi square minimization, if needed (also ML!)
3. Combinatorial unrolling
    1. Linearize combinatorial steps, like tracking, and make them linear in time and not (super) quadratic

- Difficult to see QC impacting the next 10 years, difficult to see QC NOT impacting in the next 30

Build a controlled quantum state which behaves like the one you want to study

Build an universal minimization engine

Explore all the phase space at the same time

Conclusions

- HEP has been the first scientific field to have Big Data and distributed computing needs (LHC is already at the Exascale, was studying the Petascale when floppy disks were still common!)

- Our needs are still increasing with time, in such a way that simple technology improvements cannot cope with
  - → we need new ideas, and much much R&D

- A solution for next LHC runs is (hopefully) coming from changes at various level, including
  - Embracing new technologies
  - (super) optimizing our procedures
  - Using Data Science ideas linked to the Big Data and online media revolution, with a clear focus on Artificial Intelligence

- There is an (almost) infinite space for young data scientists joining us!

Backup

How good is a Trigger? Metrics:

1. Must be able to decrease the actual data rate from 4 PB/s to something manageable (today at most 2-3 GB/s if sustained for months)
2. Must have decent efficiency on (like 10% or more) on events of physics interest
3. Must have high rejection (like ~1e(5-6)) on not interesting events
4. Must work in real time or close (CMS = 300 ms at most)

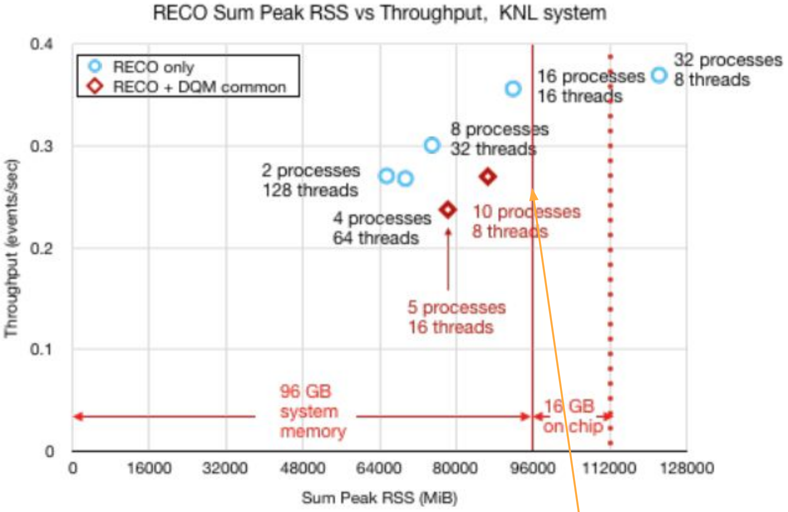
Process	$\mathcal{E}$ L1	$\mathcal{E}$ HLT	$\mathcal{E}_{\mathcal{R}}$ HLT
$Z/\gamma \rightarrow l^+l^-$	0.49	0.39	0.81
$t\bar{t}$	0.70	0.39	0.56
$W + jets$	0.57	0.42	0.72
wtb	0.61	0.36	0.59
$Zb\bar{b} \rightarrow b\bar{b}\tau^+\tau^-$	0.44	0.19	0.43
signal $m_A = 200, \tan\beta = 20$	0.60	0.42	0.70
signal $m_A = 300, \tan\beta = 20$	0.78	0.63	0.81
signal $m_A = 400, \tan\beta = 20$	0.86	0.75	0.86

Table 4.4: Selection efficiency at L1 and HLT. The last column contains the HLT trigger efficiency relative to the L1 accepted events. $\mathcal{E}_{\mathcal{R}} = N_{HLT}/N_{L1}$ where N_{HLT} is the number of events passing the HLT and N_{L1} the number of events selected at Level-1.

A Tau lepton trigger
Typical efficiencies on selected channels

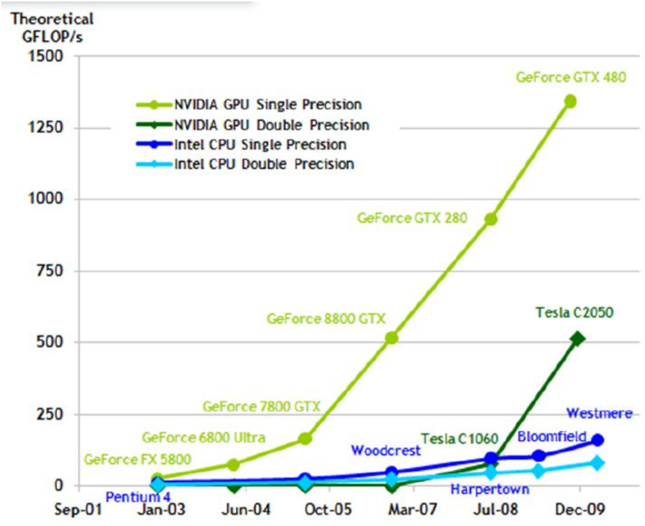
A concrete example: running on KNL

- Intel KNL is a very nice architecture:
  - Think of many ~ Pentium II in the same silicon, with some good interconnect
  - Many: 68 cores, 4-way hyperthreading → 272 cores per machine
  - On the other hand, just 96 GB of RAM
    - 0.5 GB/core – to be compared with the standard 2 GB/core needed by our sequential code
    - → you cannot run 272 jobs on a KNL, you would miss a factor 4 RAM
- Multi threading saves RAM with respect to N sequential as in previous slide



Extreme case: use 2 processes @ 128 threads each (256 cores used) → fits in 64 GB!
Some 20% decrease in overall performance (synchronizations, Amdahl law, ..)

GPGPU – relative performance



A high end GPGPU

Intel x86 cores

But beware:

- Very power hungry
- This kind of performance just for very specific use cases
- Very difficult to program