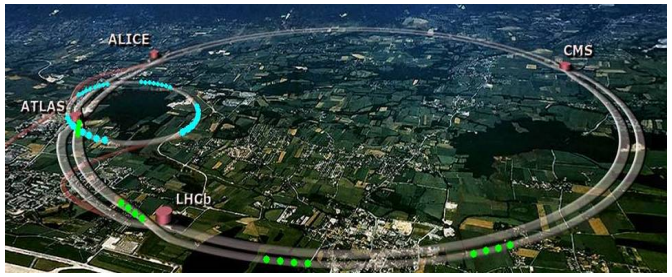
COMPUTING AND DATA MANAGEMENT FOR PARTICLE PHYSICS IN 2020

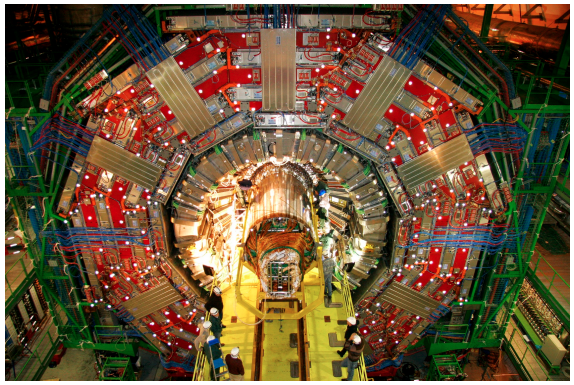
Tommaso Boccali – INFN Pisa

High collision rate +
high luminosity

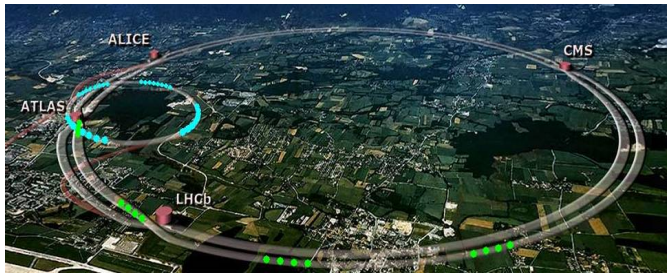


+

Millions of acquisition
channels

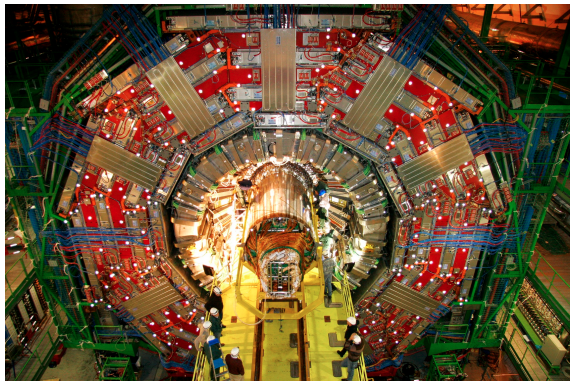


High collision rate + high luminosity



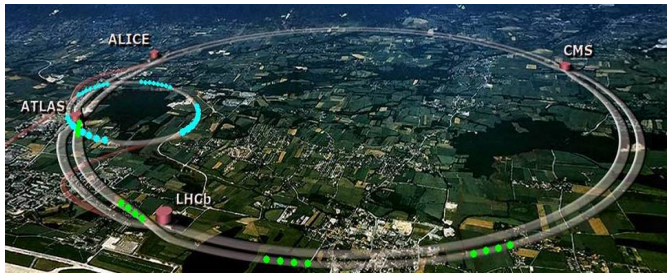
+

Millions of acquisition channels



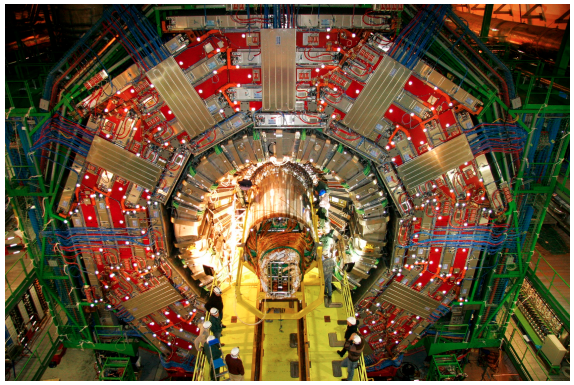
= Big Data!

High collision rate + high luminosity



+

Millions of acquisition channels



= Big Data!

What is “Big” today: RRB recommendation for ATLAS/2013, Apr 2012 (the highest of the 4)

resource	pledge
Disk (PB)	88 (~40k HDD)
Tape (PB)	53 (~10-50k tapes)
CPU (kHS06)	727 (~ 70k CPU cores)

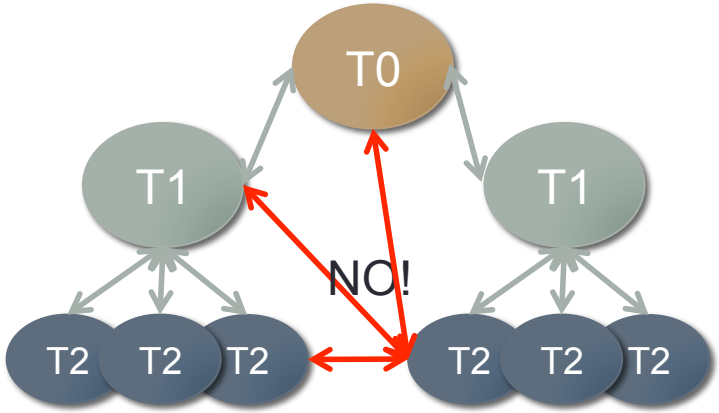
How to handle this?

How … ?

MONARC report (circa 2000)

- **"Do not centralize resources":**
  - Space + Power limited at CERN
  - Political interest in distributing resources and competences
- **"GRID as the glue between sites"**
- **"Network is the expensive piece"**
  - Only a few routes can be prepared / validated / guaranteed

- **Create a strict hierarchy of sites, different by functionalities: T0s, T1s, T2s**
- **Go for a "DataGRID" paradigm:**
  - Pre-planned data distribution
  - Computing tasks ("jobs") go where data is



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, NurduGrid, …)
- WLCG was crucial in planning, deploying, and testing the infrastructure before 2010, and is crucial for operations now

As of today, from REBUS

- **CPU 1.8 MHS06 (~180k computing cores)**
- **DISK 175 PB (~80k HDDs)**
- **TAPE 170 PB (20-80k tapes)**
- **# Sites exceeding 200**

ATLAS > 100k jobs/day
CMS > 100 TB moved /day

Still increasing …

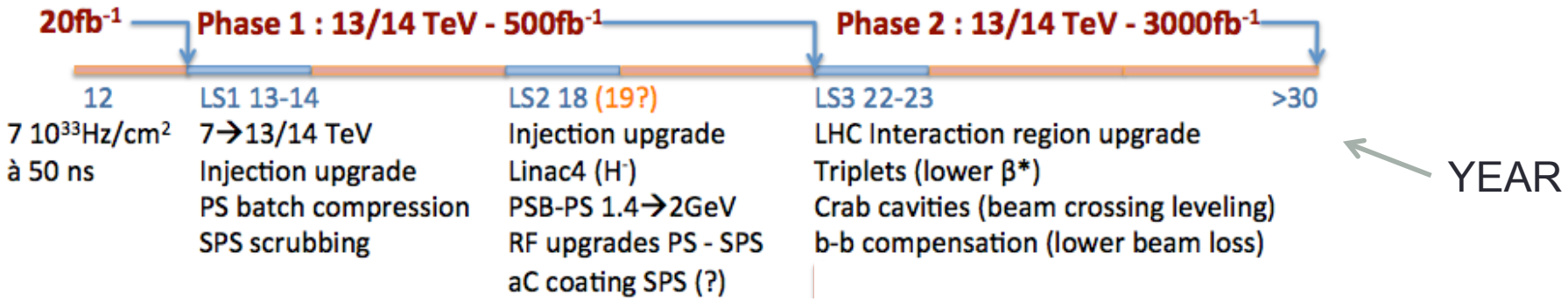
A few words about things we learned in the process

- **Software and Computing** have been considered @ LHC Experiments on par with Subdetectors
  - Not something you can arrange at last minute
  - Something on which it is a good idea to invest resources
  - Something which deserves years of testing, up to full scale

- Did it pay? For the first time in HEP, from day 1
  - We had an accurate simulation (thanks, Geant4, and thanks to years of delay). There are experiments which in ten years of running did not reach this level
  - Data processing chain working at 100% scale
  - **First physics papers in 2 months**
    - **Common to see at major conferences results using data from previous week**

Today + 10 : What is this about?

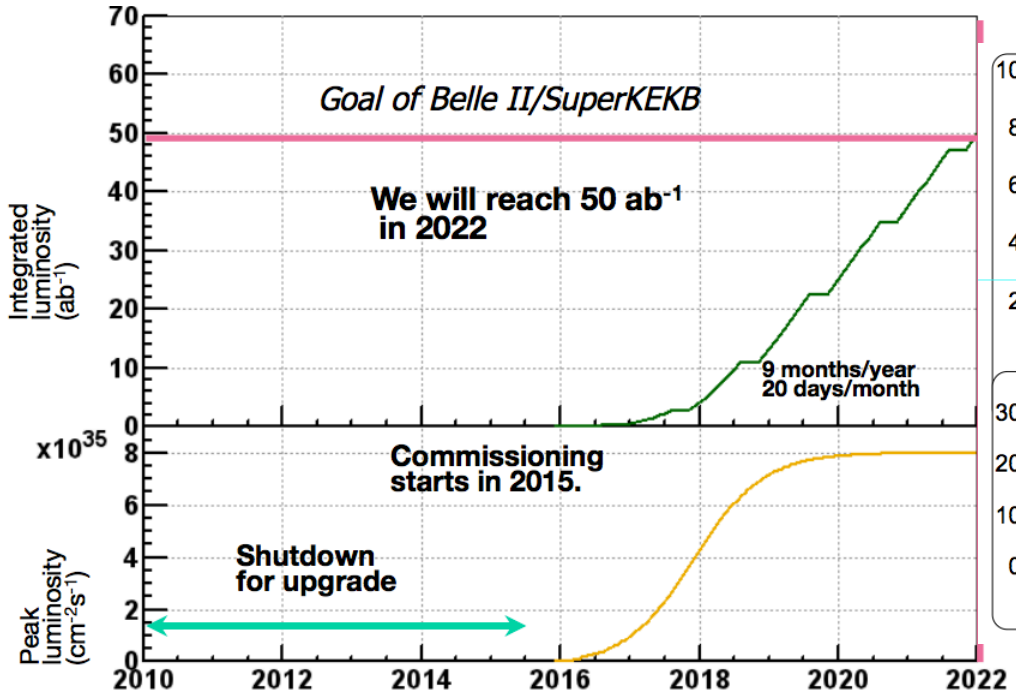
- **LHC: towards / into the HL-LHC**

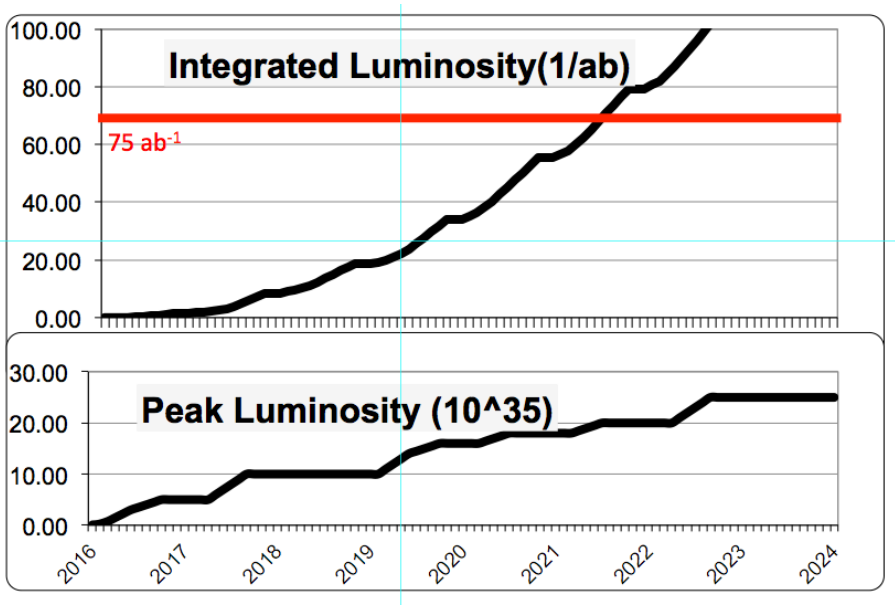


7×10^{33} Hz/cm^2 Up to 2×10^{34} Hz/cm^2

$> 5 \times 10^{34}$ Hz/cm^2
Detectors heavily refurbished

- **Super b factories:**





Belle II

SuperB

Computing “needs” #1

- **Computing models for HL-LHC not finalized**
  - They depend strongly on trigger decisions, which can still vary by factors
- **An back-of-the-envelope PRIVATE estimate (pp):**
  - Event size ~ PU driven (~3x) + some detector complexity increased (< 2x, could even decrease)
  - Filter farm output ~ 1-5 kHz (1-5x)
  - Let’s say O(10-15x) current data
  - Which translates in roughly the same increase CPU needed
  - **When no PU driven ALICE/LHCb**: factors different, end up in **5x**

	2012
Disk (PB)	80 (~40k HDD)
Tape (PB)	50 (~10-50k tapes)
CPU (kHS06)	800 (~ 70k CPU cores)

As order of magnitude!

	2022
Disk (PB)	1000
Tape (PB)	1000
CPU (kHS06)	10000

Computing “needs” #2

- **Super b factories**
  - Essentially trigger-less, so easier to extrapolate from last generation
  - Belle II: storage resources **x50** resources wrt Belle
  - SuperB: **1 EB tape, 100 PB disk, 10 MHS06 CPU**

~202X

	Year	1	2	3	4	5	
Luminosity	Peak	**0.25**	**0.7**	**1.0**	**1.0**	**1.0**	10^{36} cm^{-2}sec^{-1}
	per Year	3.75	10.51	15.01	15.01	15.01	ab^{-1}
	integrated	3.75	14.26	29.26	44.27	59.28	ab^{-1}
Data Sets	Raw Data	39.4	149.7	307.3	464.9	622.4	Pbyte
	Mini	1.2	4.5	9.2	14.0	18.7	Pbyte
	Micro	2.8	10.6	21.8	33.0	44.1	Pbyte
	User	0.4	1.3	2.8	4.2	5.6	Pbyte
Storage	Disk	9.4	31.4	51.3	68.5	77.1	Pbyte
	Tape	43.7	168.0	348.2	530.7	713.1	Pbyte
CPU	Reconstruction	0.09	0.34	0.69	1.04	1.40	MHS06
	Simulation	0.28	1.06	2.18	3.30	4.42	MHS06
	Skimming	0.06	0.26	0.59	0.95	1.32	MHS06
	Analysis	0.11	0.43	0.88	1.33	1.78	MHS06
	Total	**0.54**	**2.09**	**4.34**	**6.63**	**8.91**	MHS06

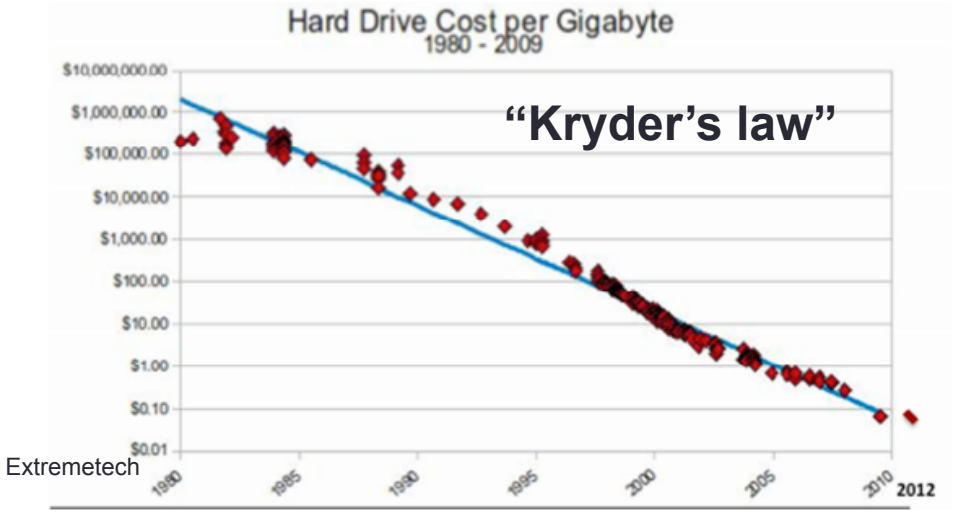
Features:

- **Big unbalance Tape/Disk (for LHC tape ~ disk)**
- **Many many many very simple events. Processing time of the single event is short**
- **Disk small since reco size goes as ~ # physics objects, and e+e- is a bless**

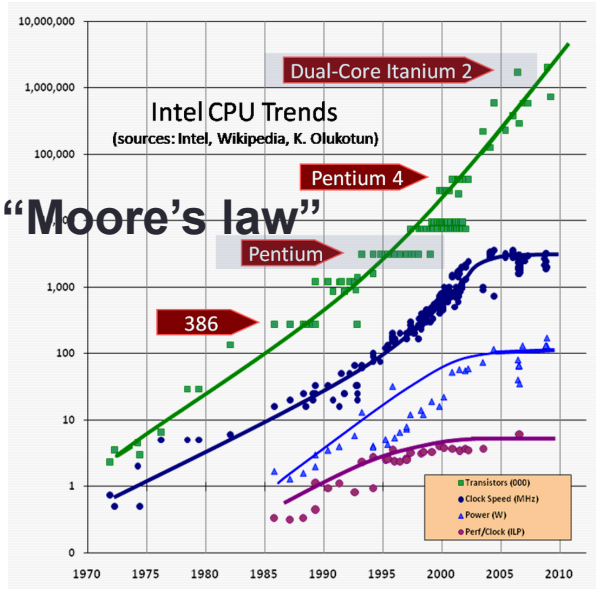
Technology trends

- **Last ~20 years have been luxury years for computing**

  - In many cases, we just had to "sit&wait"
  - Year to year up to 2x increase in capacity/computing power/#of transistors per chip/network capacity essentially for free for HEP
  - For free: the **same** program that in year 2002 was needing 10 seconds, would need 1 seconds in 2009. Same for disk capacity, memory capacity etc



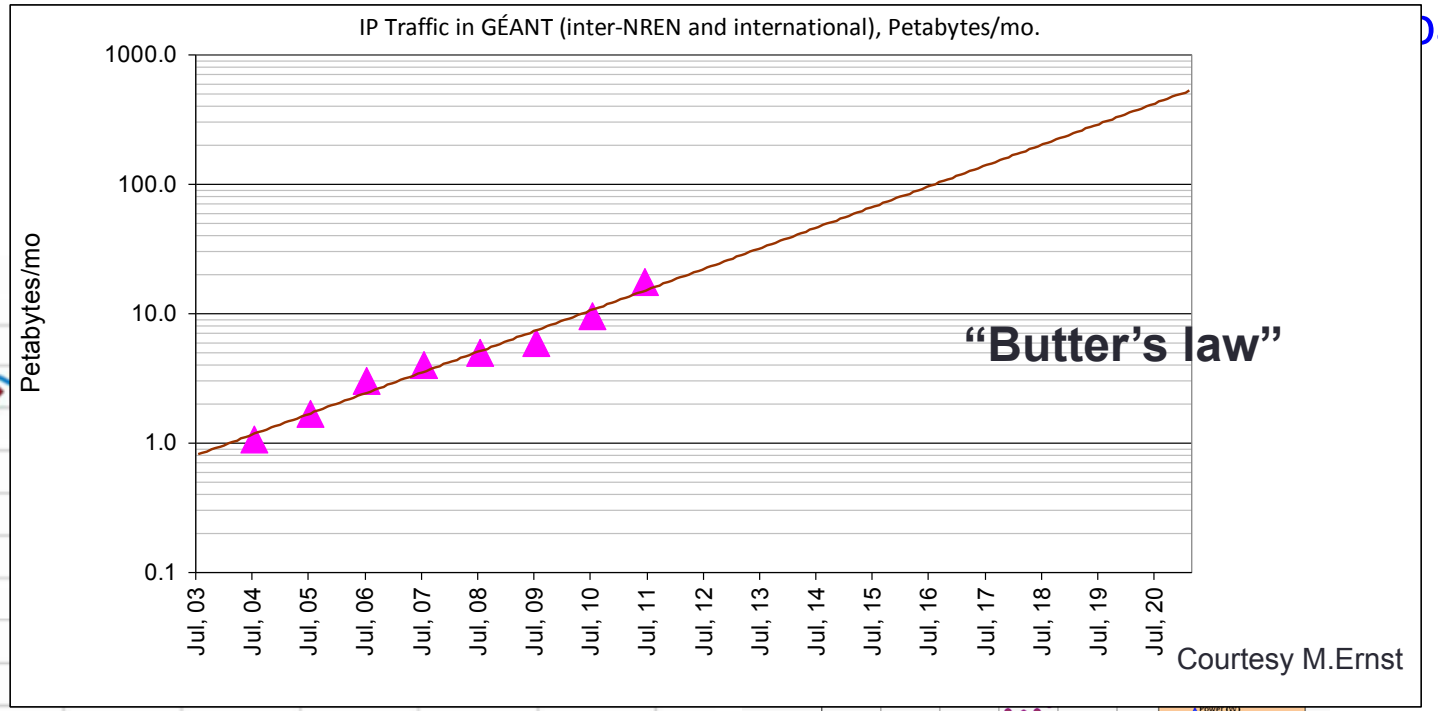
Extremetech

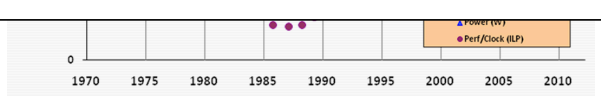


Technology trends

- **Last ~20 years have been luxury years for computing**

  - In many cases, we just had to "sit&wait"
  - Year to year up to 2x increase in capacity/computing power/#of transistors per chip/network capacity essentially for free for HEP
  - For free: the **same** program that in year 2002 was needing 10 [illegible]pacity,





If this still holds …

- In 10 years from now, you should get for the same yearly budget (and same rack space, maintenance effort, …)
  - ✓ **>~20x computing power**
  - ✓ **>~20x disk/tape capacity**
  - ✓ **>~20x network bandwidth**
- **… so, are we done with computing models for 2020+?**

If this still holds …

- In 10 years from now, you should get for the same yearly budget (and same rack space, maintenance effort, …)
  - ✓ **>~20x computing power**
  - ✓ **>~20x disk/tape capacity**
  - ✓ **>~20x network bandwidth**
- **… so, are we done with computing models for 2020+?**

If this still holds …

- In 10 years from now, you should get for the same yearly budget (and same rack space, maintenance effort, …)
  - ✓ **>~20x computing power**
  - ✓ **>~20x disk/tape capacity**
  - ✓ **>~20x network bandwidth**
- **… so, are we done with computing models for 2020+?**

…unfortunately not

- **CPU:**
  - While transistor count per chip is still increasing, **clock speed is not**. What you are going to have is many processing units (“cores”) with constant today-like speed
    - **Even at transistor count, increase is expected to slow down with smaller CMOS processes;**
    - **Power consumption can become problematic.**
- **Disk:**
  - While capacity/$ still seems on the exponential curve, I/O speed is not. A PB system made up with 10 disks you need weeks to be read is not that useful. SSD disks have an initial penalty of 10x in size (or cost). Also, already today, storage budget is an increasing fraction of the total.

- **Network:**
  - Indeed seems to keep its “good behavior” for at least some years. 100 Gbit/s geographical networks are “**close even today**”.

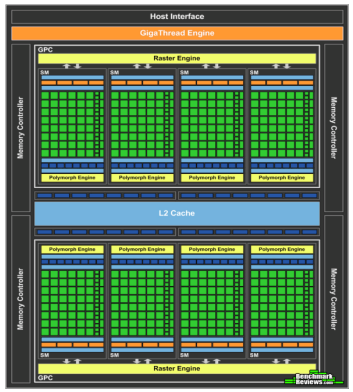
  - Let’s just assume for the sake of the talk you will have the total capacities in the previous slides, at today’s budget – then, how to use them

1. What will our software look like in 2020+?

CPUs: from increasing clock speed to increasing number of cores

- We need to learn how to use many cores per machine
  - How much is "**many**"? Today ~24
  - **Intel MIC**, future: potentially hundreds or thousands of core. Each core's performance not dissimilar from today's cores. Use of graphics cards (**GPU**) already in O(1000) cores today. If we ever move to new low power architectures (**ARM**) it will not be different.

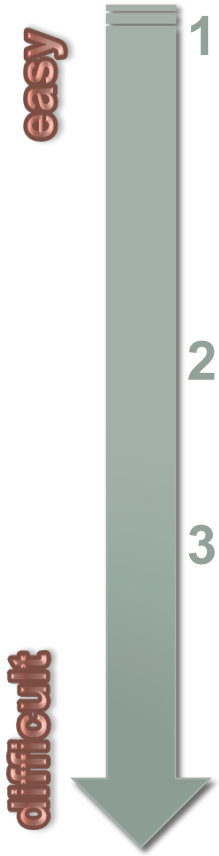
Intel MIC



NVidia Fermi GPU

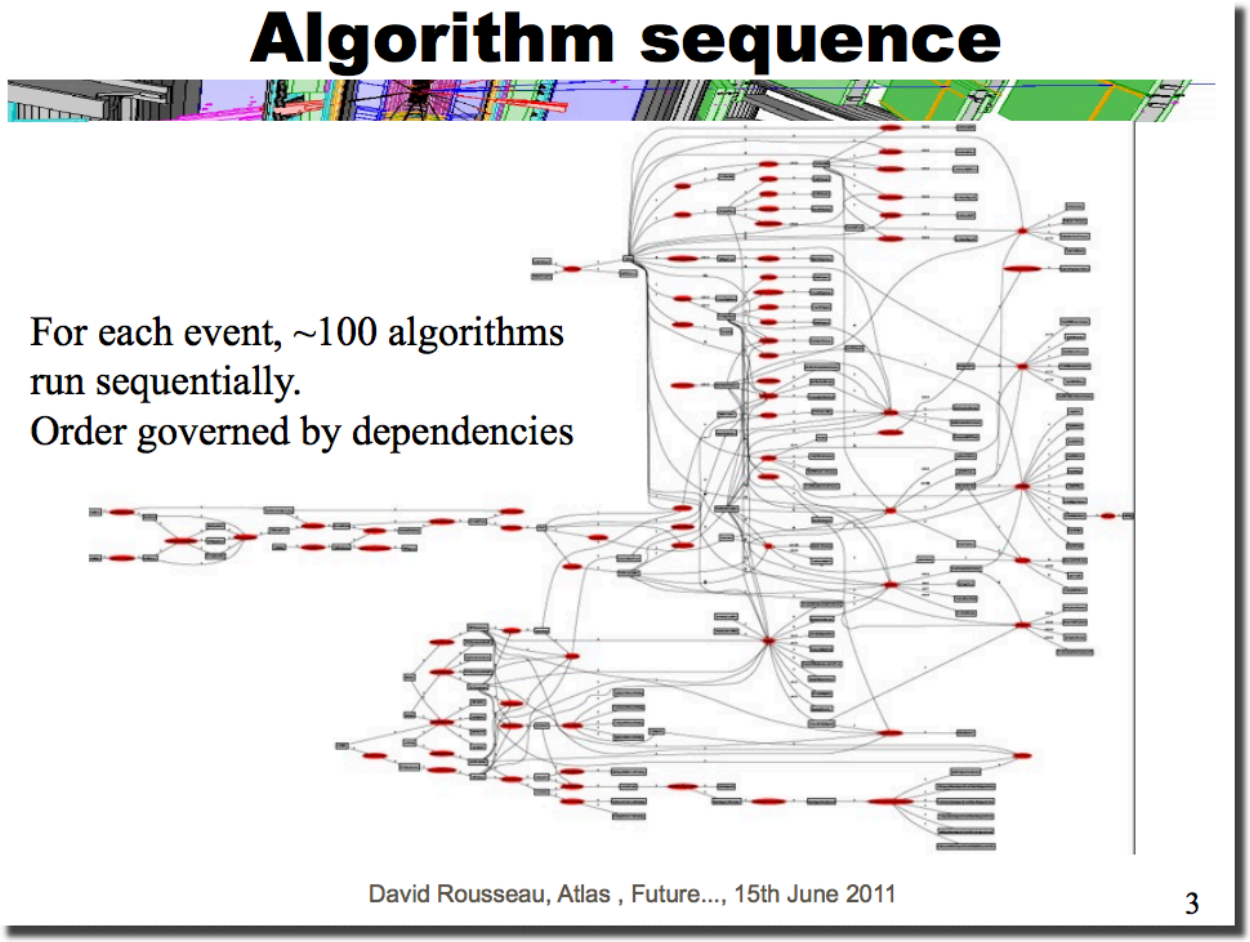
NVidia Tegra3 ARM

How to use many (>100) core machines?



1. **As batch nodes (1 job per core)**
   - It will not scale due to memory issues when dealing with 1 TB memory machines
2. **…**
3. **Implementing parallelism at algorithmic level (i.e. tracking and calorimetry at the same time for the same event)**
   - The gain depends on the **entanglement** in our software (i.e module interdependence)
     - And the ability to unroll the sequence…

Are we entangled?



- **You can improve parallelism by unrolling sequences**
- **But on the other hand global event reconstruction techniques a-la Energy Flow make everything more interdependent**

Lines are input/output to modules

… and even worse

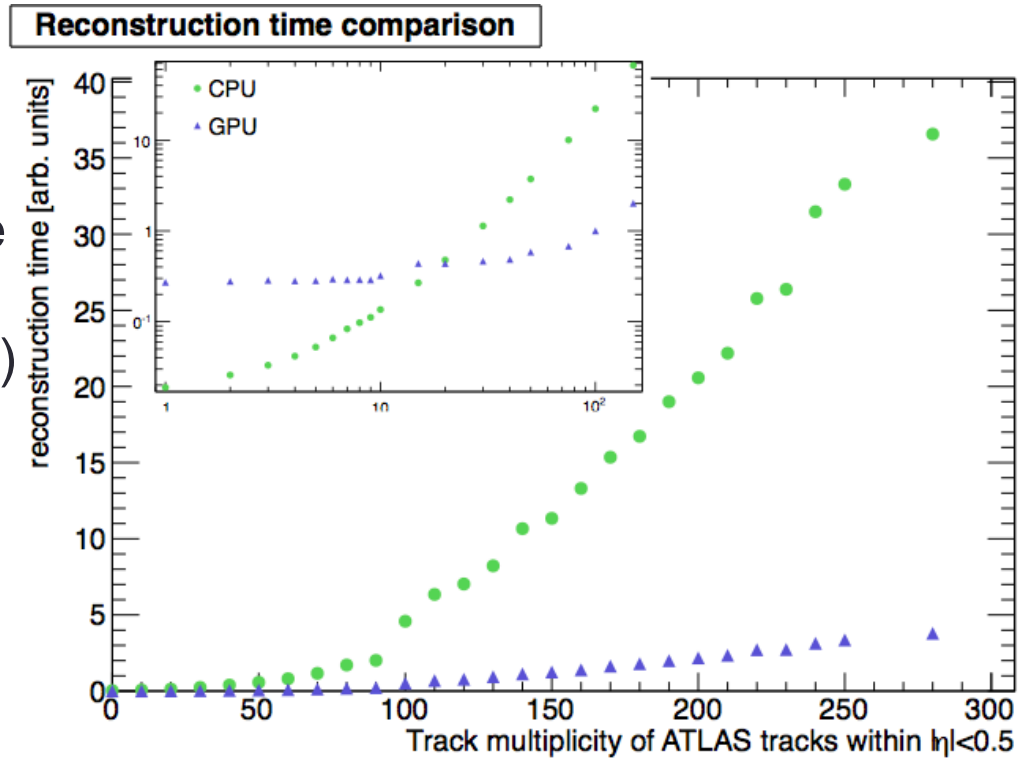
More difficult

This works well if all the modules take ~ the same time, and there is **no timeout** imposed (you are not at trigger level)

Tracking @ LHC: single module taking > 50% of the reconstruction time

You need to parallelize the single algorithm (one box in previous slide)

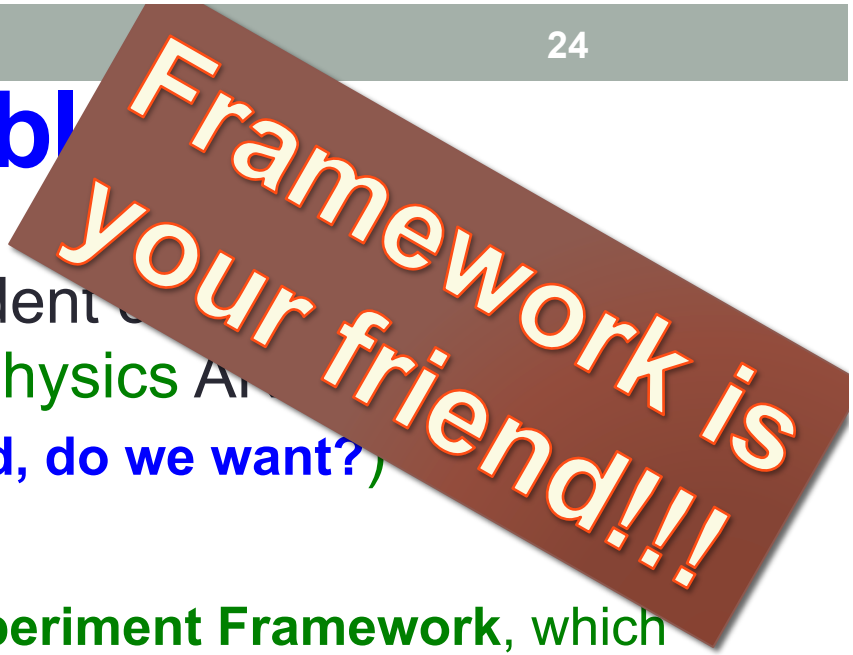
- Either multi threaded algorithms, or GPUs (from hard to harder …)



Are we (physicists) able?

- **Personally** I am skeptical a graduate student can in a PhD time master parallel programming AND physics AND possibly a detector calibration service task (**and, do we want?**)
- The best we can do is
  1. Have a well done parallel capable **Experiment Framework**, which masks modules' parallelism from users ("physicists can still think in FORTRAN")
     - **Start Now**: you need to learn new architectures as they appear, you need to explore coding outside general HEP knowledge, you need to teach users how to operate with the FW.
     - **Start Together**: we are not that different after all.
       - **Concurrency Forum** an opportunity in this direction, trying to bring together the whole (not only CERN-centric) community.
  2. Have some exceptions when otherwise impossible (tracking….); there (few) physicists and parallel programming experts work together in mission critical algorithms

Are we (physicists) abl



- **Personally** I am skeptical a graduate student
  master parallel programming AND physics A
  detector calibration service task (**and, do we want?**)
- The best we can do is
  1. Have a well done parallel capable **Experiment Framework**, which masks modules' parallelism from users ("physicists can still think in FORTRAN")
     - **Start Now**: you need to learn new architectures as they appear, you need to explore coding outside general HEP knowledge, you need to teach users how to operate with the FW.
     - **Start Together**: we are not that different after all.
       - **Concurrency Forum** an opportunity in this direction, trying to bring together the whole (not only CERN-centric) community.
  2. Have some exceptions when otherwise impossible (tracking….); there (few) physicists and parallel programming experts work together in mission critical algorithms

Event Simulation

- Particle-matter simulation toolkits, born in HEP for HEP, are a huge gift to non-HEP world, and are widely used in science and industry. Citations' count for Geant4, for example, exceeds 3.2k and still growing fast (and, most of these are not references from HEP!)

- **In the close future, we need to:**
  - Have an improved descriptions of microscopic interactions. This is needed to cope with next generation detector R&D, where new materials are beyond the scope of current approximations.
    - Do not assume that since today simulations are usable, they have reached end of development phase

- **Ensure a better use of resources:**
  - **Geant4 is the single software component which takes more CPU in HEP processing**
  - If you want, the toolkit is responsible for most $$$ spent on computing, OR
  - A 1% performance improvement here is more $$$ than elsewhere
    - **Be it via use of new architectures or by rewriting critical parts of the code**

2. How to run our software in 2020+?

Evolution, not revolution

1. Good world-wide coordination by WLCG
2. Grid model **is** able to sustain the flux of incoming data

- We are in continuous operations in the the next 10+ years; no opportunity to “start from scratch”

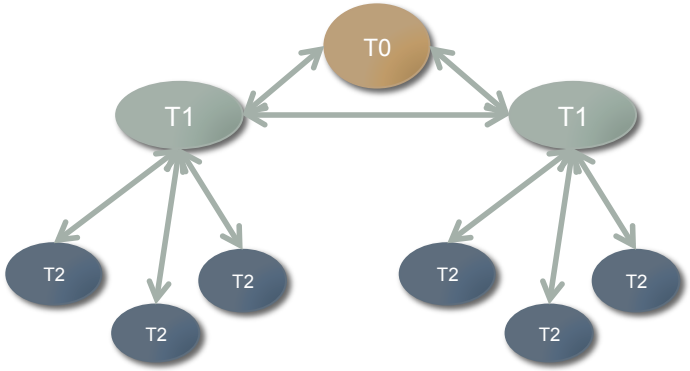
- Tier1s and Tier2s are serving for the design purpose, and form a very solid base for any development
  - Given the LHC schedule, no time for revolution, just evolution
  - WLCG’s **T**echnical **E**volution **G**roups (**TEG**) a step in this direction

- **Keyword: use better what you have (remove inefficiencies)**

Trends – data management

- **work to improve the efficiency of what you (can) have**

- **Network is your friend (the commodity resource for the next 10+ years)**
- **Relax MONARC**: flatten hierarchy of transfers



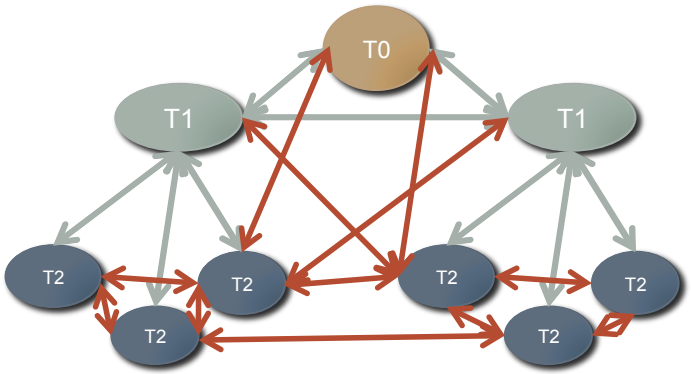
- Use network to place facilities where more convenient,
  Distributed Tier0 @ CERN and SuperB

- **Allow for remote data access, decoupling CPU from Storage and thus using both at best** (today we store multiple copies also because the access pattern is not trivial)
  - **Can be a life saver if disk/cpue increase less than expected**
  - **Expected to happen via "Storage Federation", defined by "network vicinity"**

Trends – data management

- **work to improve the efficiency of what you (can) have**

- **Network is your friend (the commodity resource for the next 10+ years)**
- **Relax MONARC**: flatten hierarchy of transfers



- Use network to place facilities where more convenient,
  Distributed Tier0 @ CERN and SuperB

- **Allow for remote data access, decoupling CPU from Storage and thus using both at best** (today we store multiple copies also because the access pattern is not trivial)
  - **Can be a life saver if disk/cpu increase less than expected**
  - **Expected to happen via "Storage Federation", defined by "network vicinity"**

Trends - facilities

- **Grid** is named after the Power Grid: **transparent use of resources, "without users having to care"**

- …But eventually, "**site admins have to care**":
  - Difficult to run jobs on sites where your experiment does not have local support
  - Difficult to absorb peak usages using resources "external to your experiment"
  - **A limit to efficient resource usage for HEP Expts**
  - **A huge limit for the exploitation by other sciences**

Clouds?

- **What is a Cloud? Can be many things, but let's focus on ability to use resources in a more flexible way (ComputingAsAService)**
- CMS-2004 Computing Model: local CMS support in a site is badly needed to run CMS tasks

You need	Who provides it (as in 2004 CM)
A computer connected to network, with conditioning and power	**Local site staff**
An operating system "compatible" with the application	**Local site staff**, after negotiation with experiments
A local installation of the experiment software (and a local area where to store it)	**Local site** staff provides area, **Experiment support** installs software
Machines for local experiment facilities (voboxes etc)	**Local site staff** provide them.
A local storage containing the input data	**Local site staff** needs to have bought storage for the experiment
A configuration to be executed	**User**!

Clouds?

- **What is a Cloud? Can be many things, but let's focus on ability to use resources in a more flexible way (ComputingAsAService)**
- CMS-2004 Computing Model: local CMS support in a site is badly needed to run CMS tasks

You need	Who provides it (as in 2004 CM)
A computer connected to network, with conditioning and power	**Local site staff**
An operating system "compatible" with the application	**Local** …fter nec…
A local installation of the experiment software … local area where …	…ovides …ment support …tware
Machines fo… experime… (voboxes e…	**…l site staff** provide …em.
A local storag… …ing the input data	**Local site staff** needs to have bought storage for the experiment
A configuration to be executed	**User**!

Site intervention needed at all levels to support the experiment

In a cloud model …

You need	Who provides it (cloud)
A computer connected to network, with conditioning and power	**Local site staff**
An operating system "compatible" with the application	Comes as a virtual image from the **experiment central infrastructure**
A local installation of the experiment software (and a local area where to store it)	Downloaded on demand from the **experiment central infrastructure**
Machines for local experiment facilities (voboxes etc)	They are also virtual images / not needed locally
A local storage containing the input data	Data can be accessed remotely
A configuration to be executed	**User**!

Much easier to use non pledged resources
Better treatment of load spikes / low activity moments

Already today, as test, HEP sw runs on HelixNebula

In a cloud model …

You need	Who provides it (cloud)
A computer connected to network, with conditioning and power	**Local site staff**
An operating system "compatible" with the application	Comes as virtual image from … **…eriment …ructure**
A local installation of th… experiment softw… local area w…	…demand **…eriment …infrastructure**
Machines for lo… experiment facili… (voboxes etc)	They are also virtual images / not needed locally
A local storage containing the input data	Data can be accessed remotely
A configuration to be executed	**User**!

Site intervention just needed to provide the fabric

Much easier to use non pledged resources
Better treatment of load spikes / low activity moments

Already today, as test, HEP sw runs on HelixNebula

Extreme consequences?

- Your site becomes a “proxy” to Amazon Cloud Services (EC2)
- All your site has to do is communicate a Visa/Mastercard number to Amazon …

- **Forget it …**
- Studies (here) and experiments’ estimates still say Commercial Clouds are **factors more expensive** than academic in-house (**and are going to stay for long**)
- Better keep our T1s and T2s operative and funded for the future …

Data Preservation / Open Access

- **Our data is valuable (both as physics content, as $$ to get it and as human effort)**
  - It's lifetime-of-interest is not predefined (new TH models: we may need to look back!)
  - **Data Preservation** (as the ability to retain the possibility to look into it for long) is becoming crucial to our field
  - Better plan **in advance** (before hundreds of PBs are stored in unfriendly ways)
  - **DPHEP project well on track, large HEP labs need to be in the game**
- **We are publicly funded: not OUR data (or at least not forever)**
  - **Open access** to HEP data is essential to our "public image"
  - Open access naturally **drives experiment to common structures / tools / software**
  - Some fields are more advanced than us:
    - NASA: data public after 1 year, **FITS** format for all the data

Get these built in from the start in future computing models

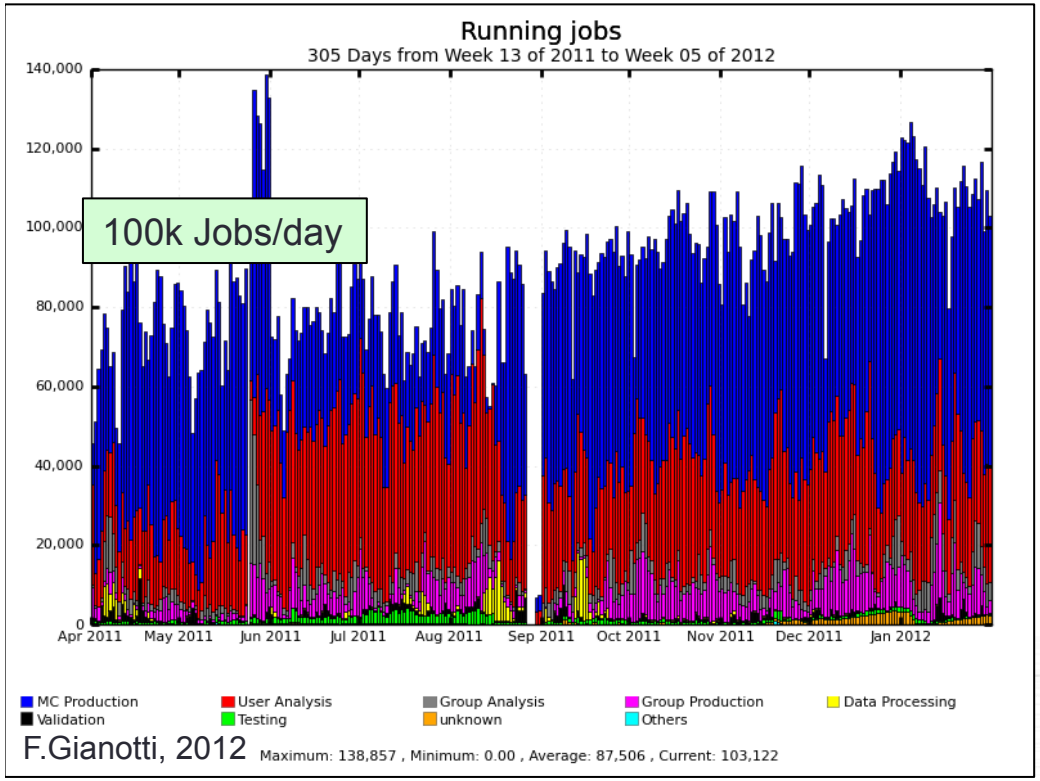
"Vision"

- **LHC Expt used to be the scientific costumers with highest data needs – the "Big data" ones, served by WLCG**
  - No reason to have special efforts for LHC (WLCG mandate) experiments: already now next-gen HEP experiments are evaluating WLCG products. Difficult to see how needs could be too different and they should reinvent all from scratch. **Be HEP-wide, not LHC-wide**
- **HEP: It is not clear we will remain the biggest data/CPU consumers in science**
  - Square Kilometer Array telescope (circa 2025: 2900 antennas, spread into multi million squared kilometers): **multi terabit** signal delivery on O(1000) km, final data for analysis **exceeding 100 Gb/s**
  - Human brain project (>2020): **10^{18} flops**, ~10M todays cores (or 1M GPUs)
- **Still, we are the ones with more "expertise" on the field today and can/should drive next years' development and planning – and eventually a science-wide eInfrastructure**

In one page, my personal summary

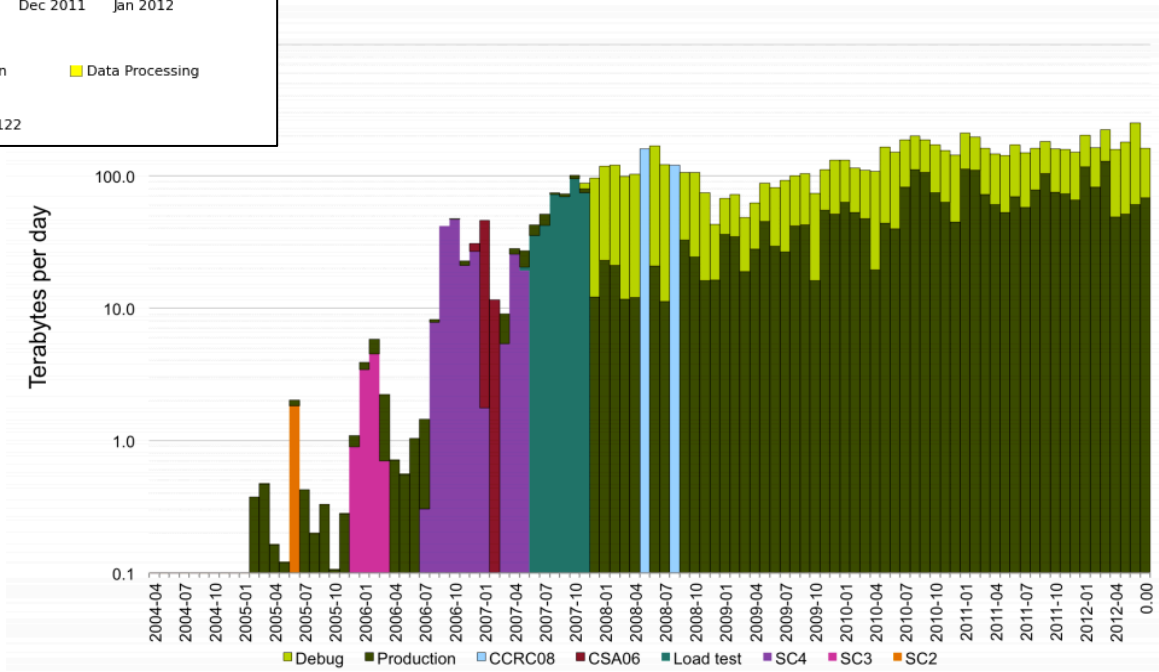
1. Whatever software we will run in 10 years, it will
   - Be on a complex architecture
   - Need new competences and a parallel framework to shield users: **start now**, **start together**
2. On infrastructure side, we will not have periods of playtime, we need evolution, not revolution
   - Build your model on the current solid base of T1s and T2s – keep them **funded and working**, evolve from these
3. Increase efficiency of resource utilization: **network is your friend, clouds (…)**
4. Whatever model will come out
   - Will need a central coordination and support
   - Will need 100% scale testing before day 0

Backup …



Jobs/day exceeding 100k (ATLAS)

Transfers: >100 TB/day (CMS)



Why so much Data/CPU?

- **Data:**
  - Accelerator + Detector (but, in the end, physics reach driving these)
- **CPU:**
  - Detector complexity (order of 100M acquisition channels)
  - Event complexity (with ultimate pileup, hundreds of charged tracks per event) – combinatorial explosion
  - Very accurate simulations in fast changing magnetic gradients, with μm accuracy in sensitive regions

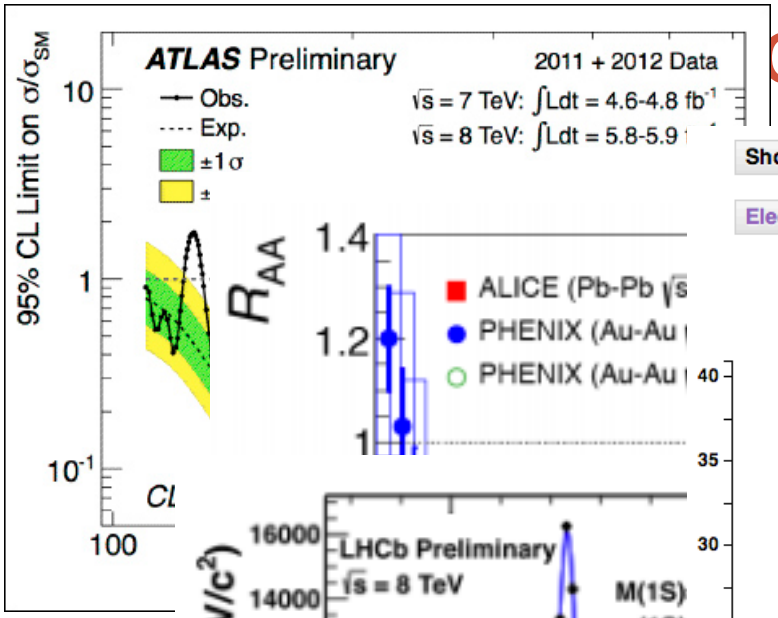
	CMS or ATLAS 2012
Inst lumi	Up to ~8 Hz/nb (8e33 cm-2s-1)
<PU>	1-40
Output to "tape" (after triggers)	400-1000 Hz soon
Event size	O(1 MB/Ev)
Data Events/y	5-10 BEvents
MC Events/y	10-20 BEvents

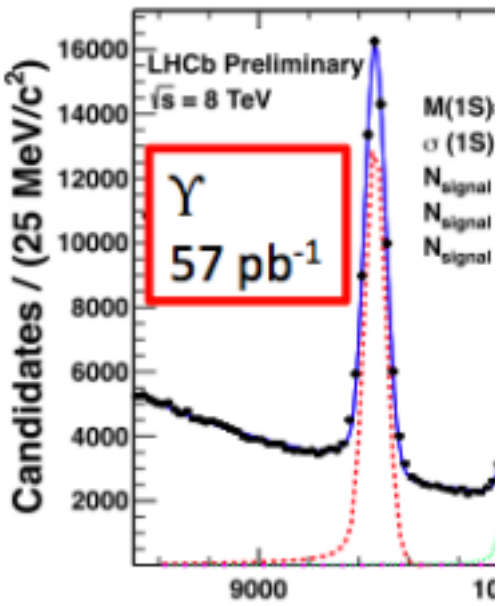
~600 s

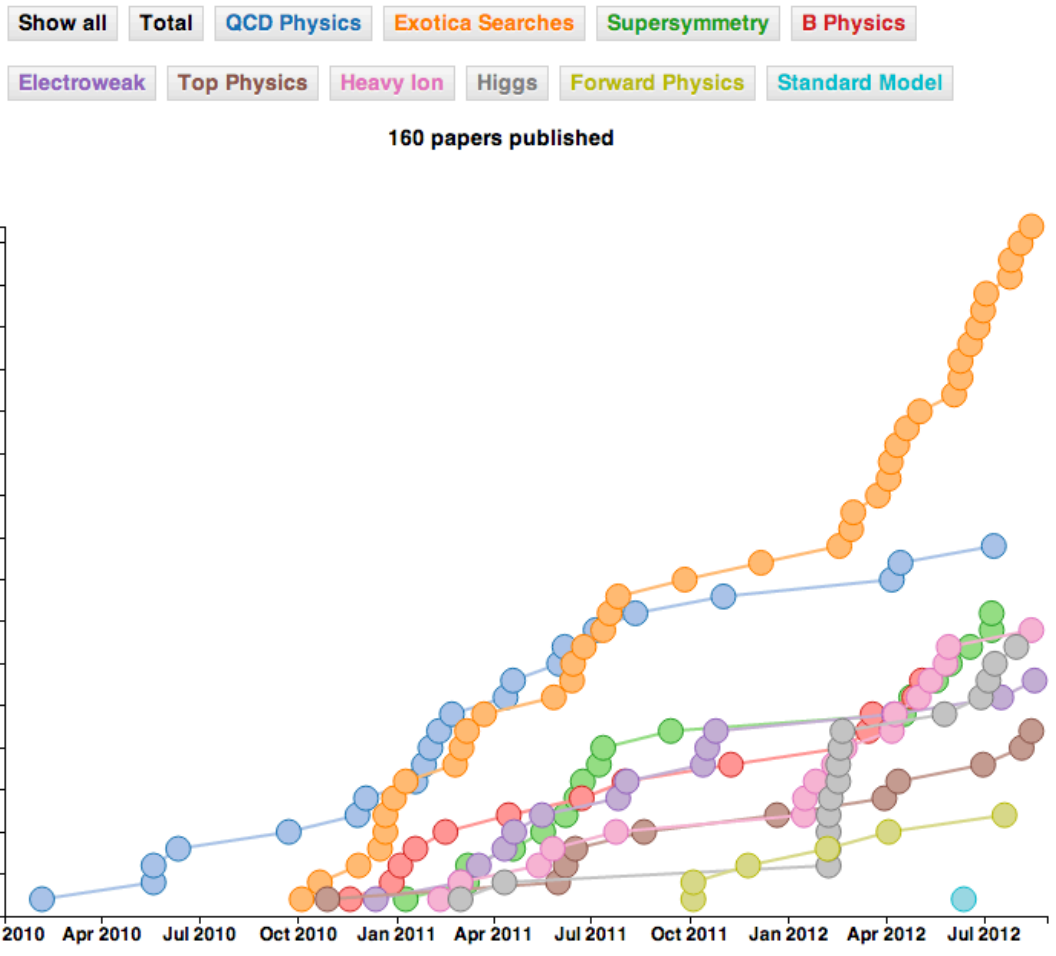
CPU times per event			
	Full sim CPU	HS06-s	6000
	Fast sim CPU	HS06-s	400
	Real recon CPU	HS06-s	80
	Sim recon CPU	HS06-s	135
	Group analysis CPU	HS06-s	20
	User analysis CPU	HS06-s	0.2

ATLAS - K.Bos, 2011

ore physics oriented way

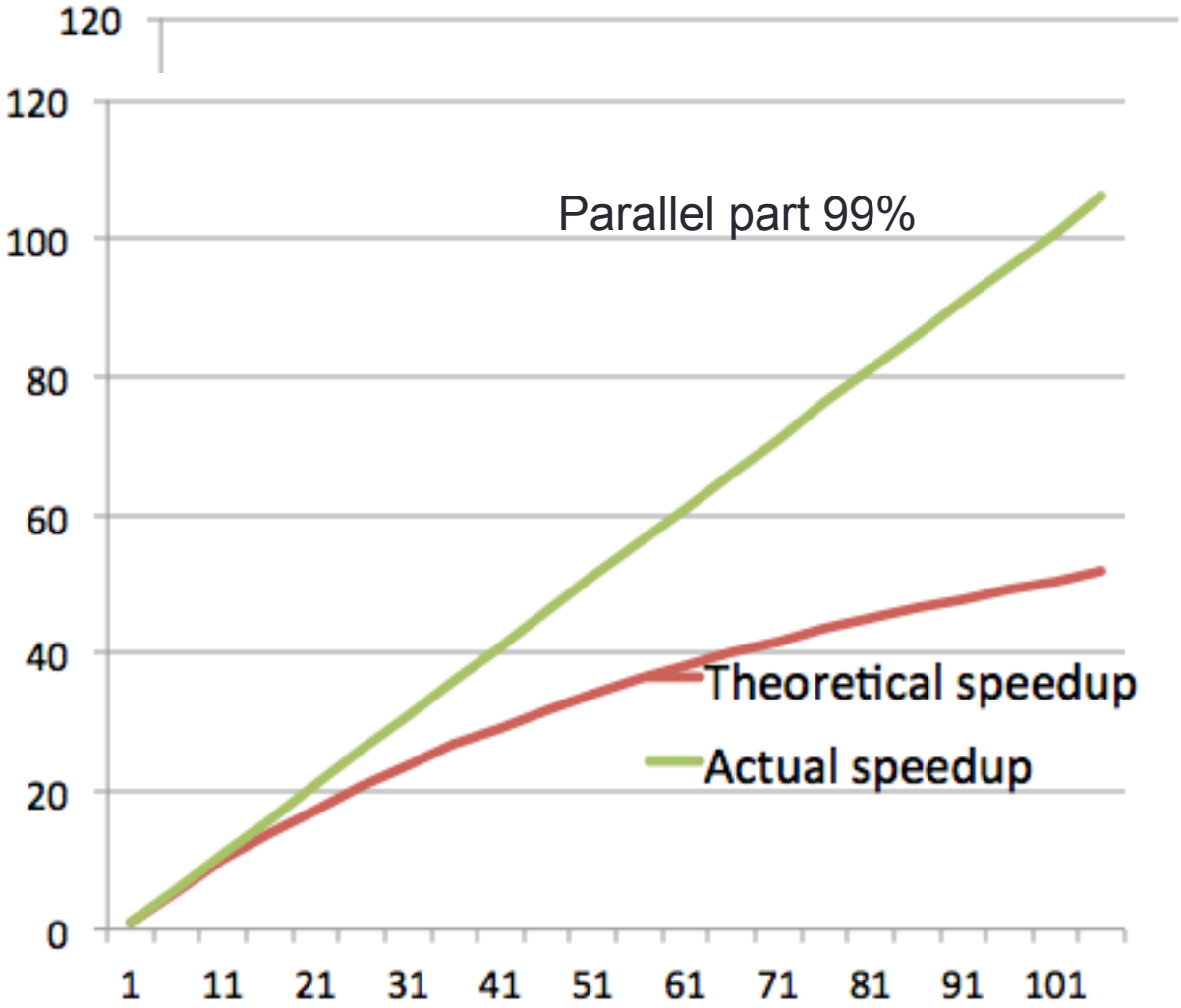






$$S(N) = \frac{1}{(1-P) + \frac{P}{N}}.$$

When number of computing cores (N) increases, the maximum speedup (S) depends on the parallel fraction (P) So, if activities before the forking (load geometry, open I/O connections) , and after (wait for slowest process) can be a killer



With 10% of non parallel part, you simply cannot gain using 10+ core

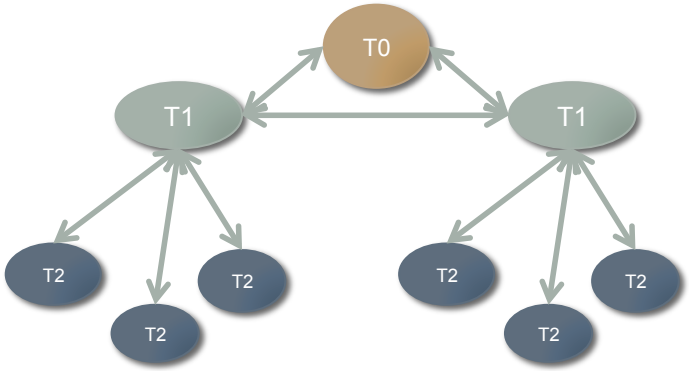
With 1% of non parallel part, you underuse by 50% 100+ cores machines

A different battle ….

- We **can** certainly improve our current software, on current hardware, by improving quality in coding
  - The caveat is the same: it costs a lot of **time** and **education** if one wants to increase the **average** level
    - And most probably does not even pay off
  - It is practical only in very specific use cases
    - Mission critical algorithms (tracking, again)
    - Toolkits, where changes are confined and do not impact external usage
- This is already happening today, mostly driven by necessity
  - For example, battle with increasing pile up at LHC

Evolution - Already now

- Network “not the limiting factor” as foreseen
  - Thanks to commercial increase of traffic, forcing down prices and boosting technology
    - BBC: during Olympics2012, peaked at 700 Gbit/s in streaming
  - No real need to limit traffic to some routes
  - Full mesh of transfers possible (LHCONE there to allow T2 to T2 flat traffic)

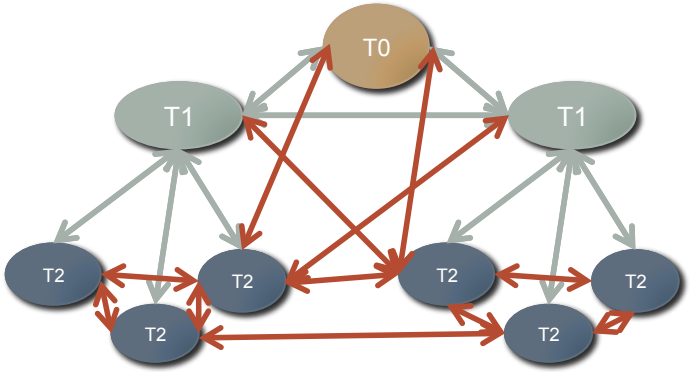


- Analysis not only at T2s
- MC not only at T2s
- (initial) possibility to access remote files

In the direction of removing the limits we had self imposed

Evolution - Already now

- Network “not the limiting factor” as foreseen
  - Thanks to commercial increase of traffic, forcing down prices and boosting technology
    - BBC: during Olympics2012, peaked at 700 Gbit/s in streaming
  - No real need to limit traffic to some routes
  - Full mesh of transfers possible (LHCONE there to allow T2 to T2 flat traffic)



- Analysis not only at T2s
- MC not only at T2s
- (initial) possibility to access remote files

In the direction of removing the limits we had self imposed

Trends – tools and software

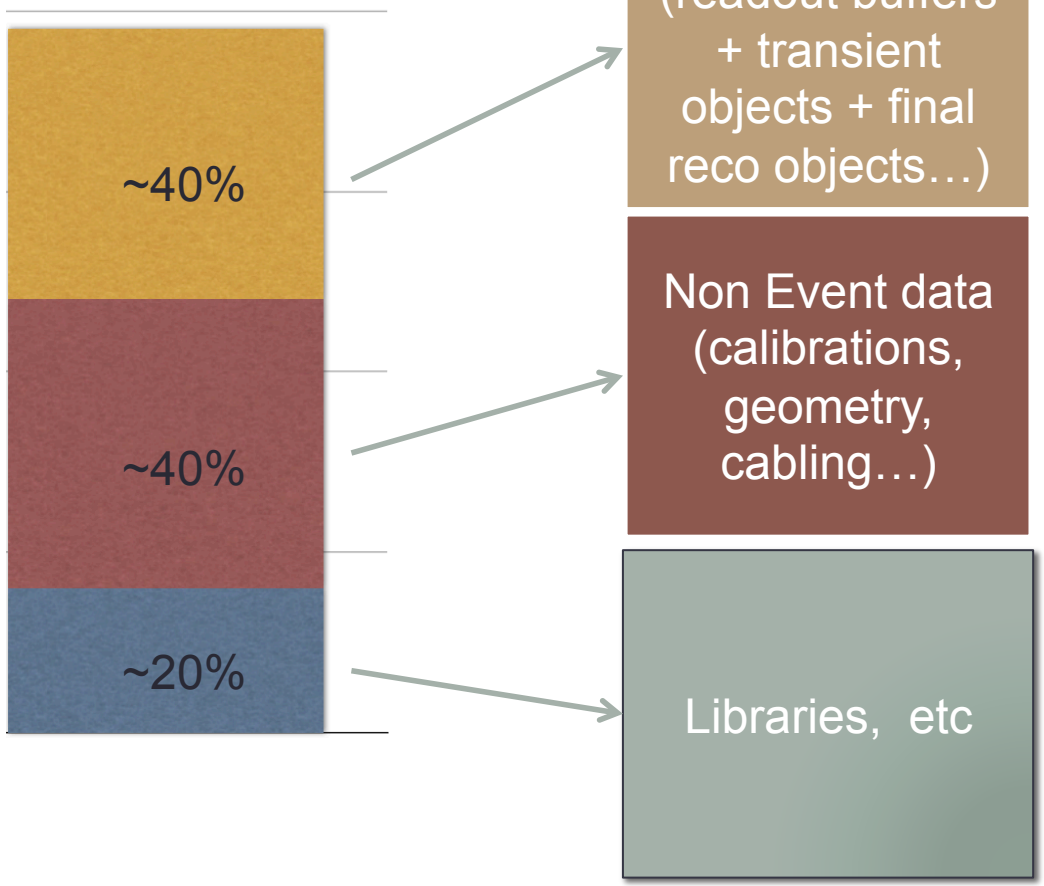
- We (HEP) are generally good at writing software but
  - We are not that many (and maintaining is worse than writing)
    - Use **common solutions** between experiments – differences are small
  - Use open standards / open software whenever available
    - Open source community is soooo larger, and long term maintenance of open source software is usually good
  - Involve industries (CERN OpenLab is a good example)

- Planning for support, maintenance, testing is a MUST
  - Don't take for granted what we have now, it would fade fast in a not WLCG supported environment
  - Plan to test everything @ 100% scale before real data arrives (lesson from the past…)
  - **Remember that support is not for free, and cannot be a last minute addition**

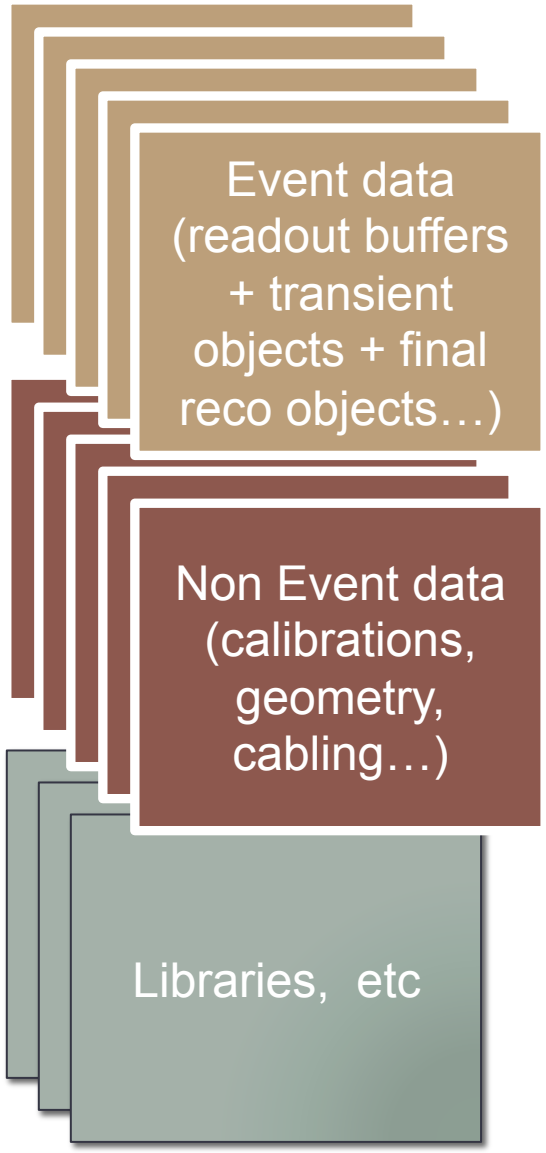
Many (many!) cores – how to handle them

- How are our tasks using system memory?

CMS Reco
Total ~ 2 GB



Memory needed while processing reconstruction



With a **batch-like** approach (use N jobs on a N core machine) , you end up using N times the memory of the single task.

Apart from cost (>1TB memory/ machine) there are issue of

- Memory to CPU communications not remaining "flat"
- Frequent Cache miss (CPU remaining idle)