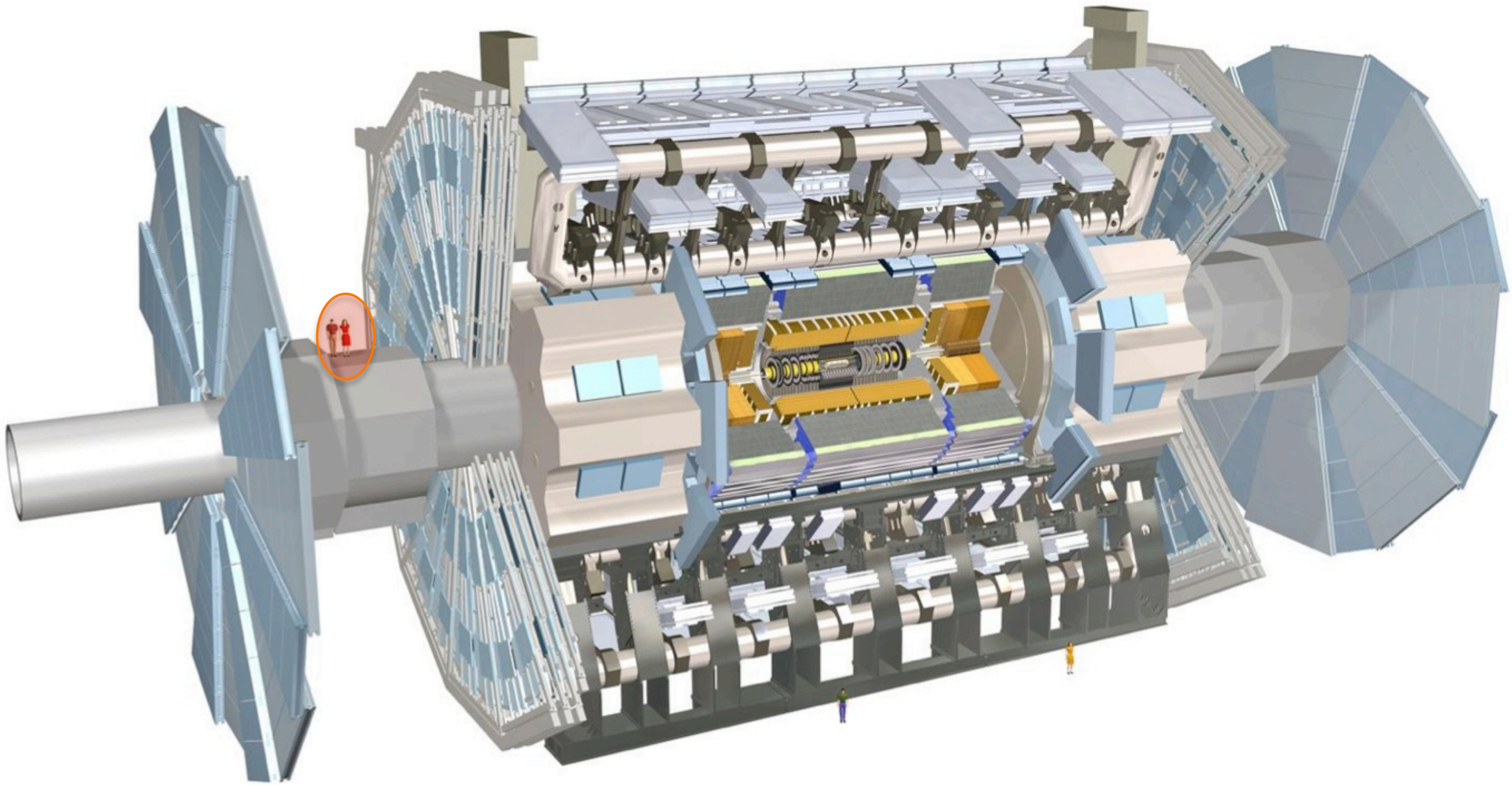
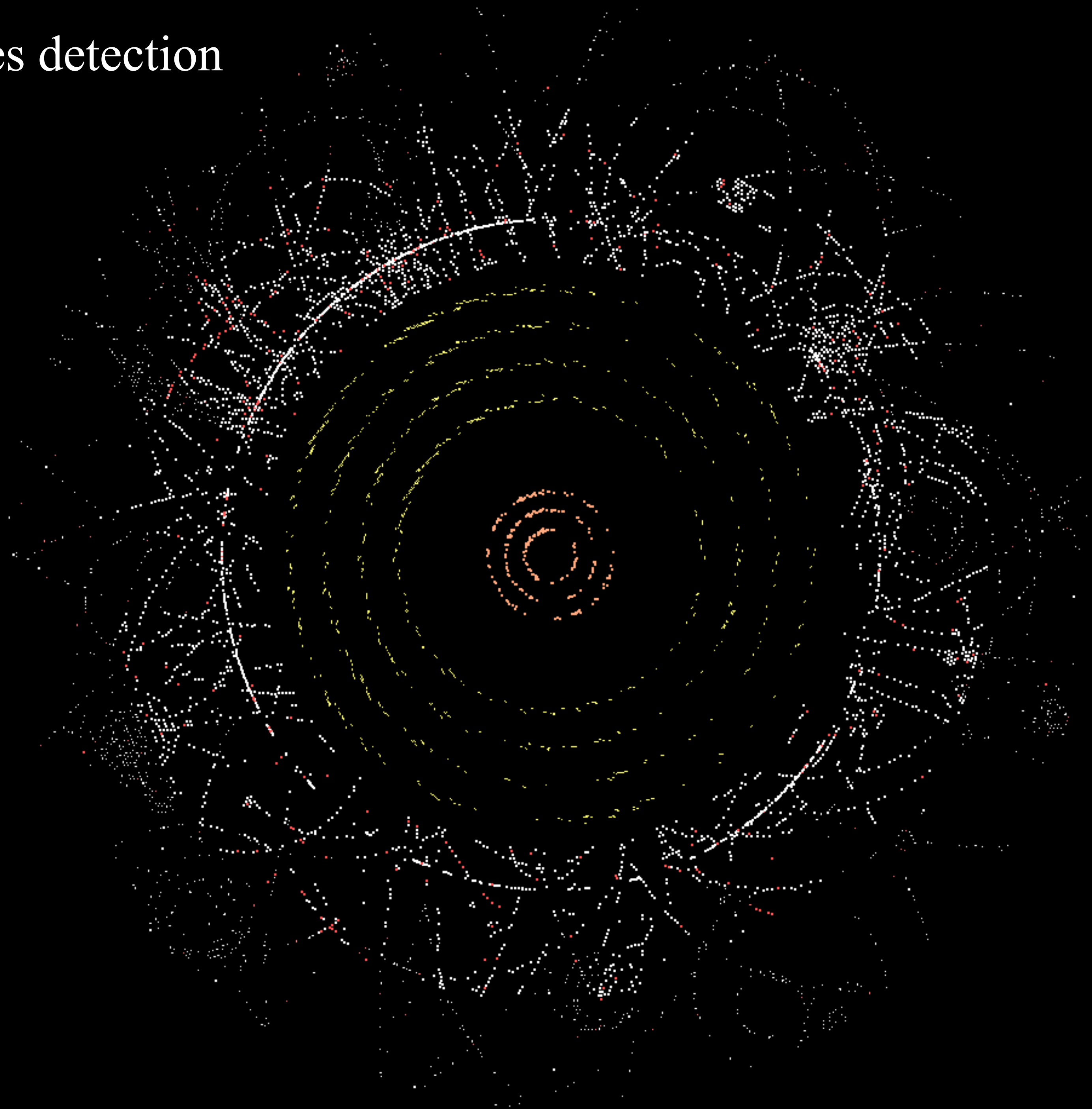
HEP software: the return of parallelism

David Rousseau (LAL-Orsay)

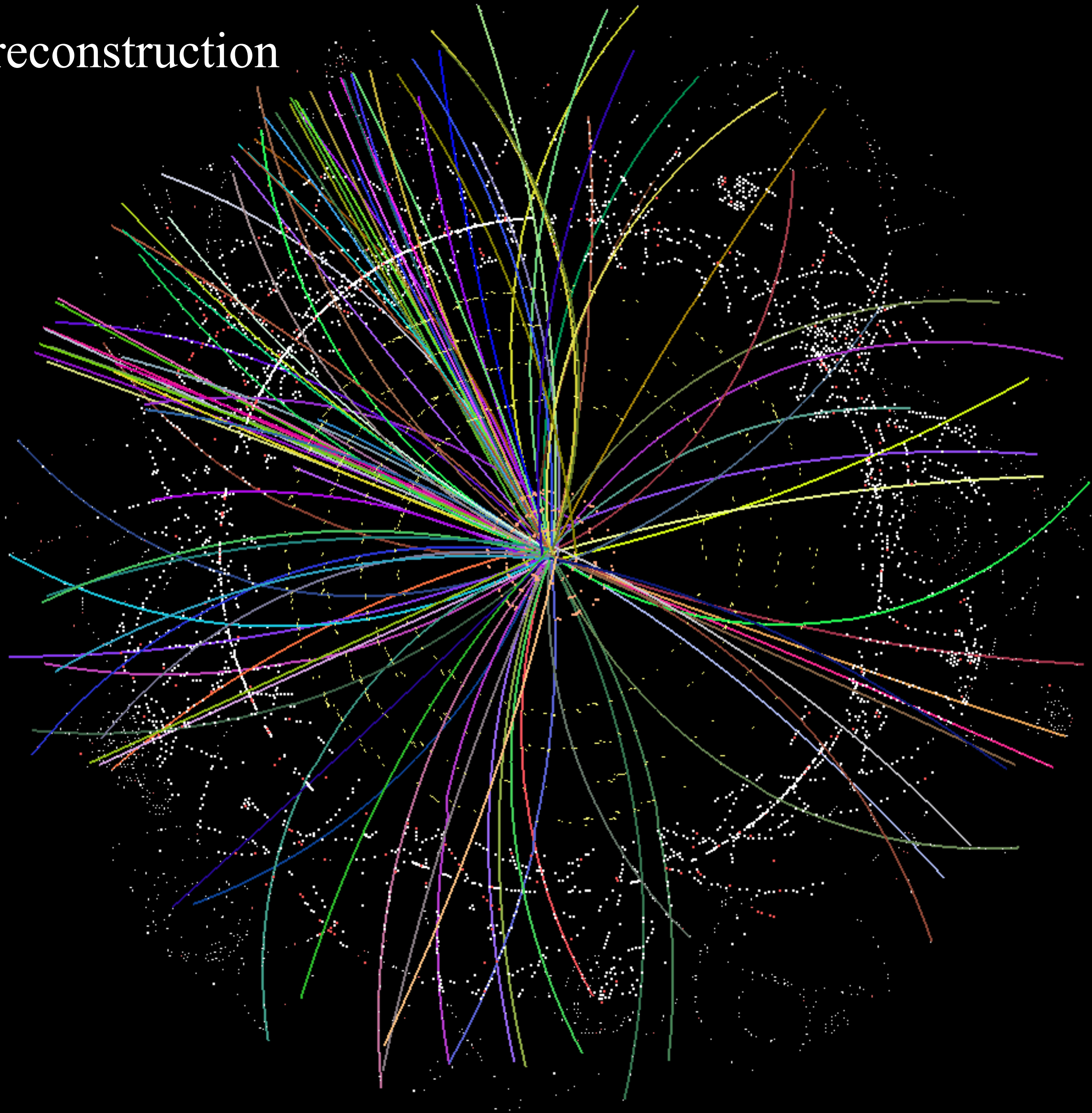
ATLAS detector



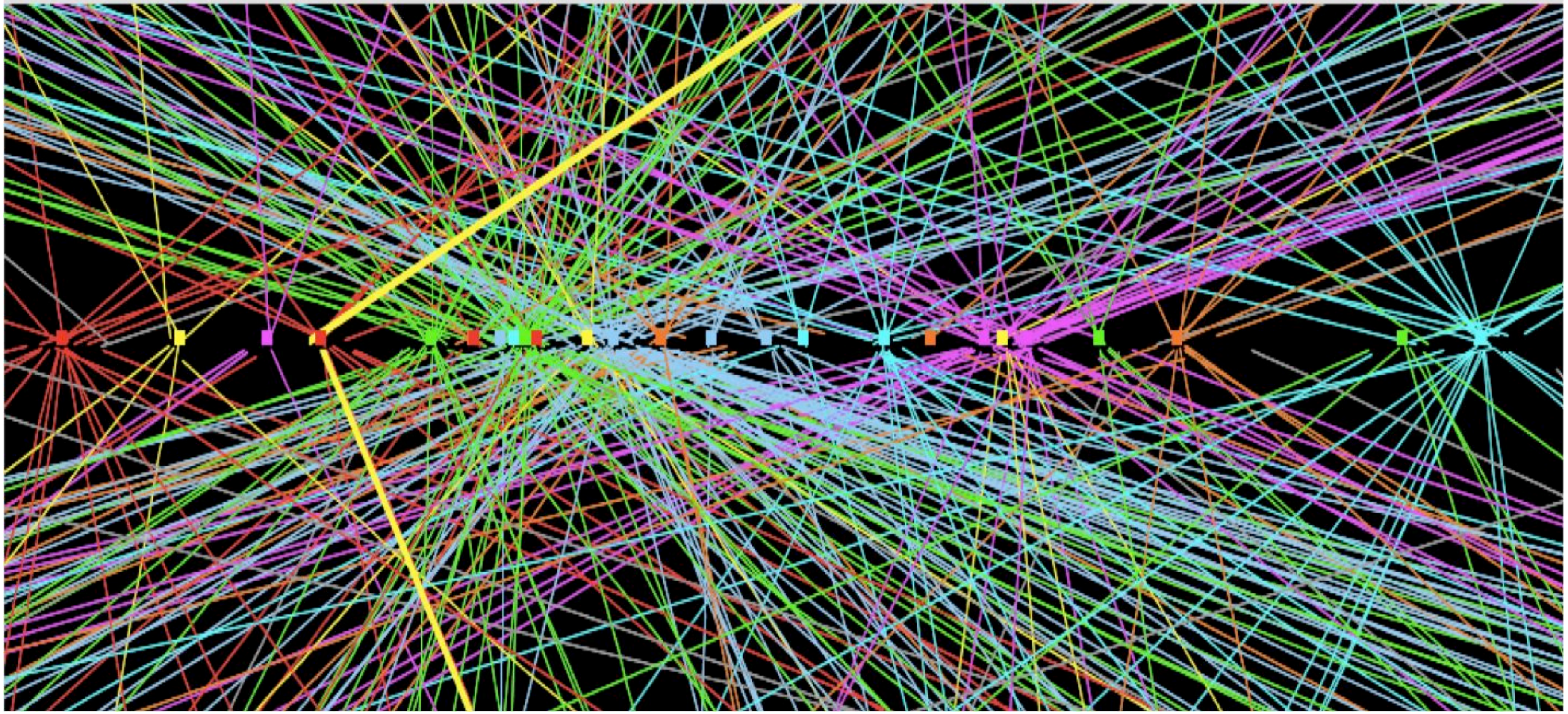
Particles detection



Track reconstruction



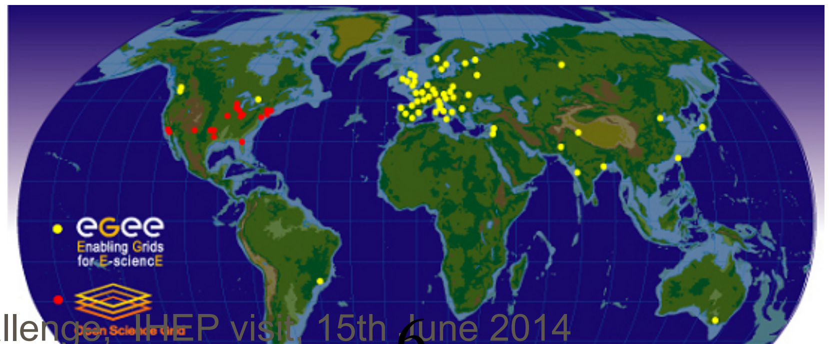
Event



The detector precision allows to separate the interesting collision from ~20 additional ones (pile-up)

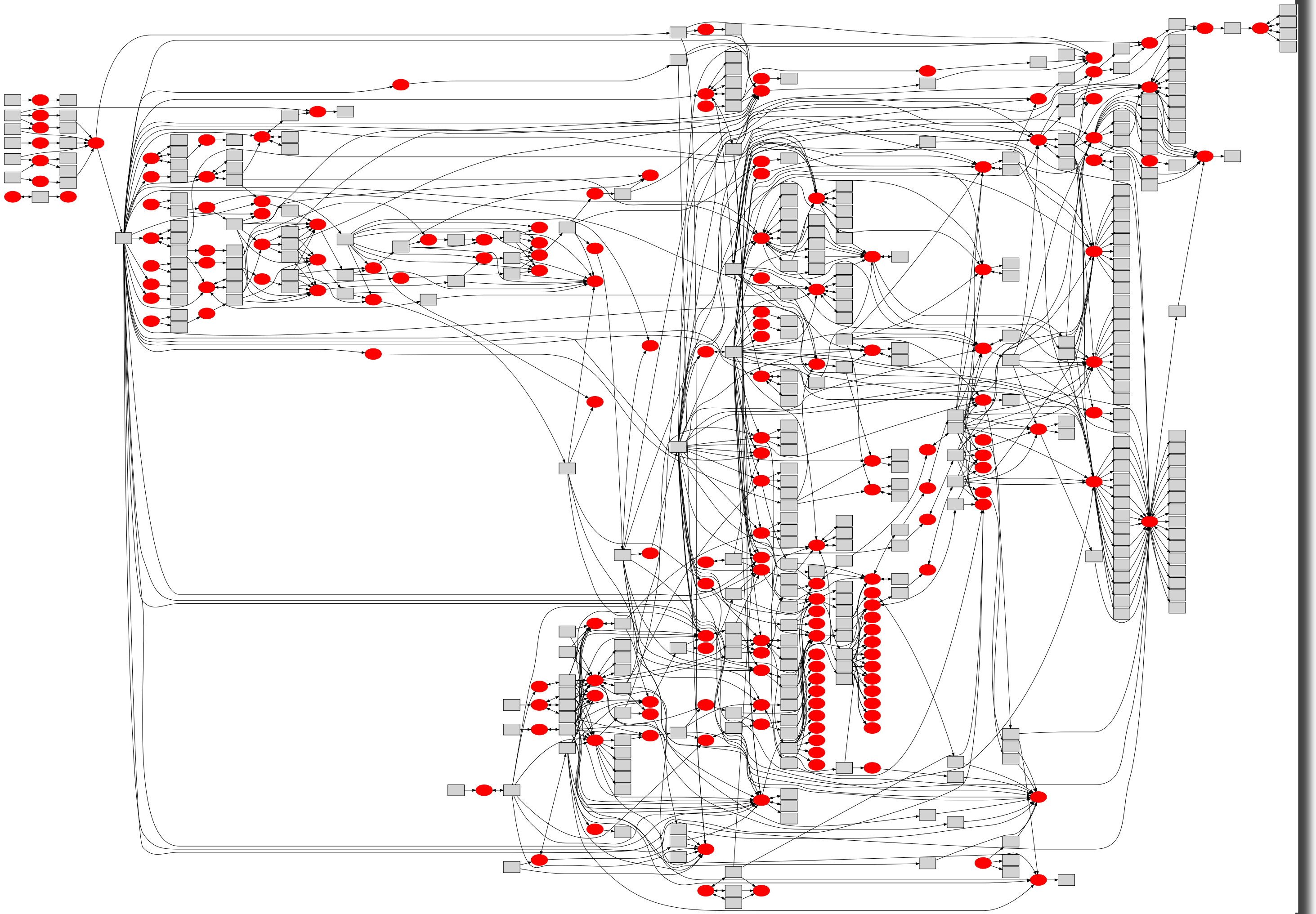
Few numbers on data handline

•several PetaBytes of data each year for one experiment =>
•prompt reconstruction on ~6000 cores at CERN
•…world wide distribution on the grid
•Reduction to a few GigaBytes on physicists' laptops
•Meanwhile, about 150.000 cores running permanently to produce about a few billion events per year (~same number as real data)



Real life

- Direct Acyclic Graph extracted from real reco job
- Today, algorithms run sequentially



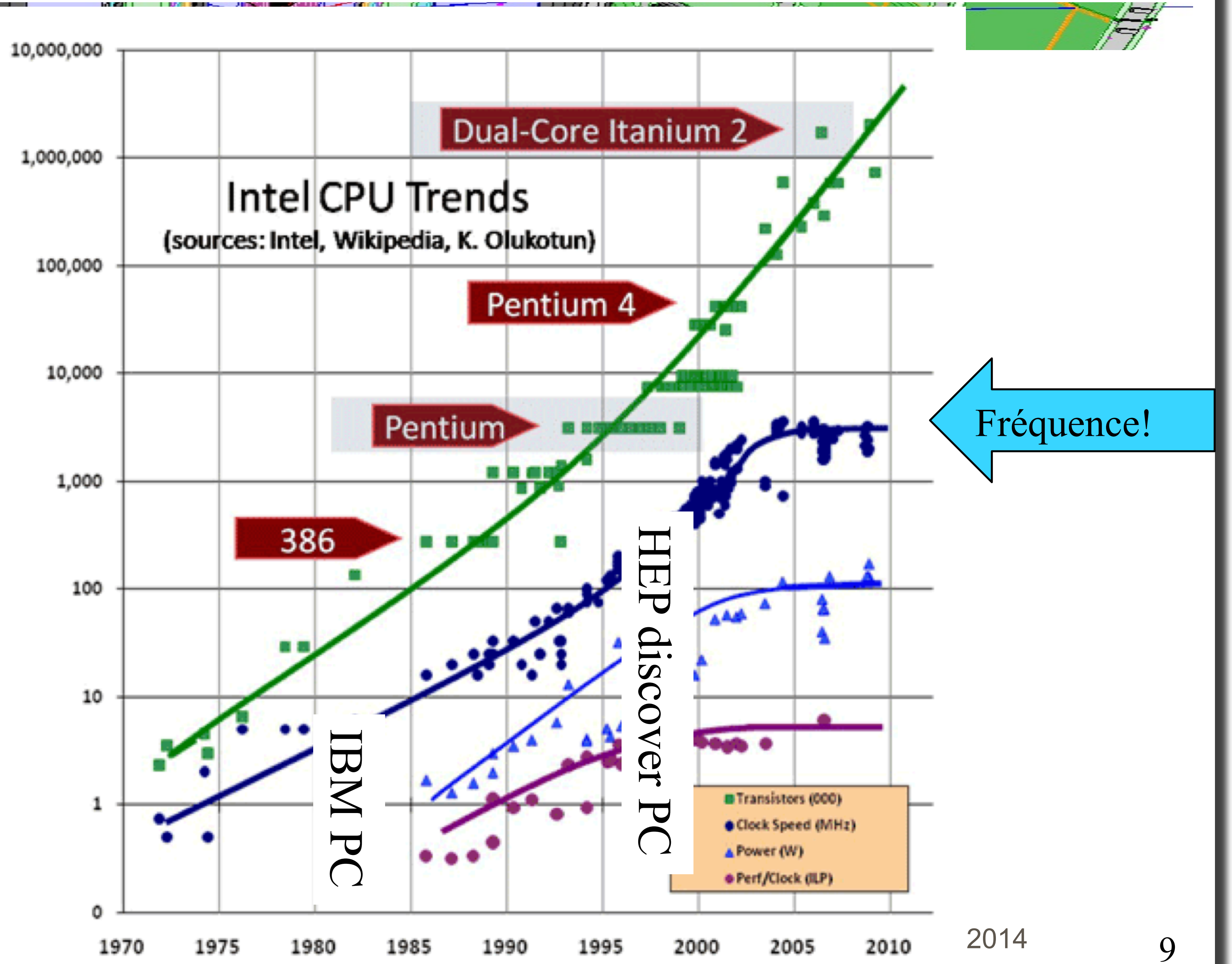
A very brief history of HEP computing

- From 50ies to beginning of 90ies: mainframe IBM, or (here at CERN) Cray XMP

- During the 90ies : risc processors farm:

(here SHIFT at CERN)

Processor technology



1997@CERN

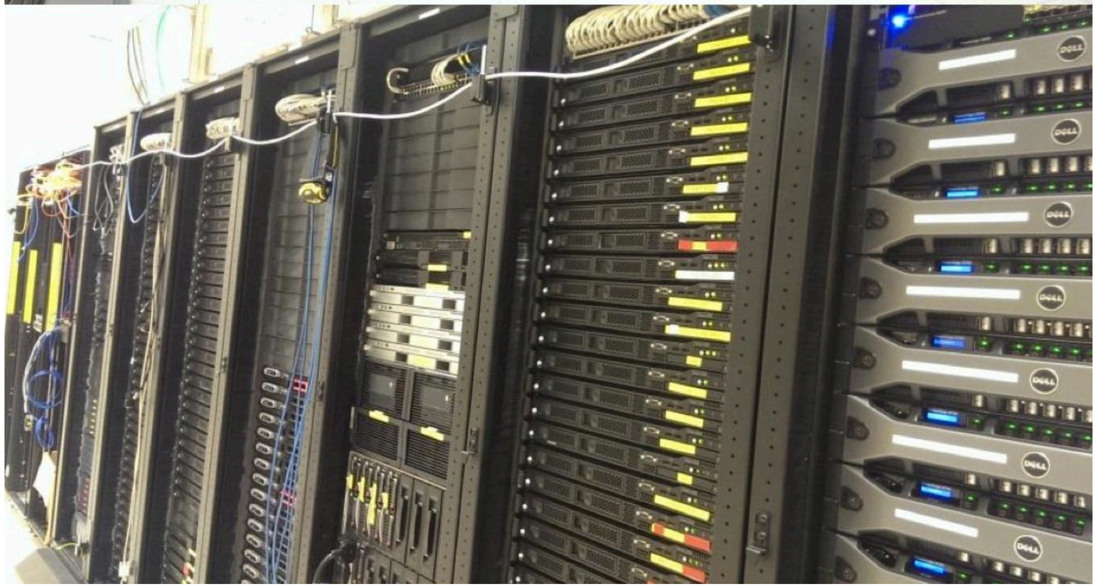
PC farm

Pentium 200MHz

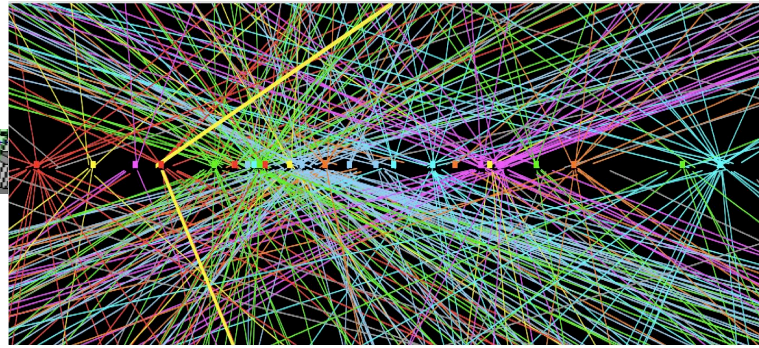
A bit later

Virtual data

- 2014 @ Orsay
- Salle Vallée
- Still intel processors, in racks and multicores



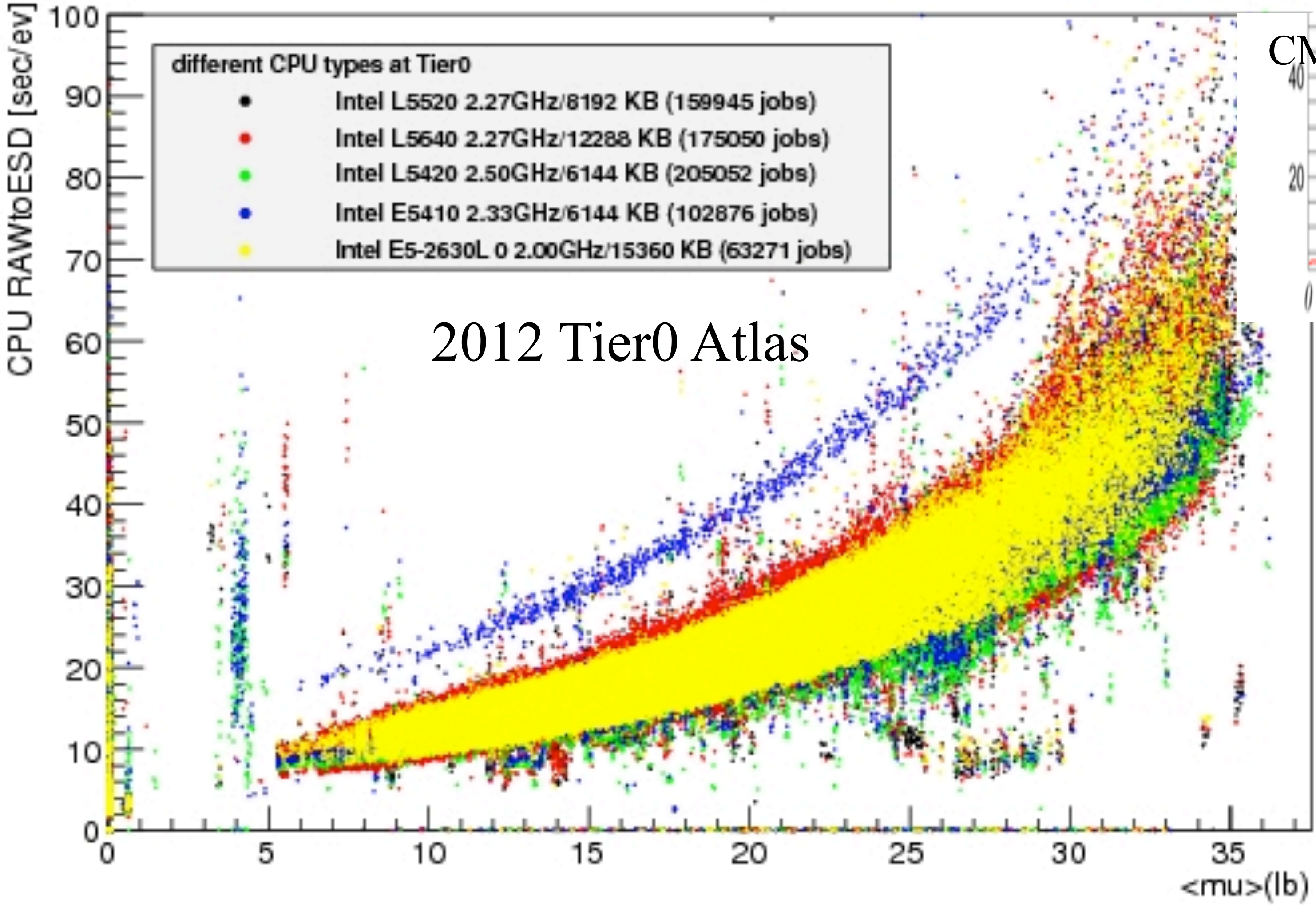
LHC Context

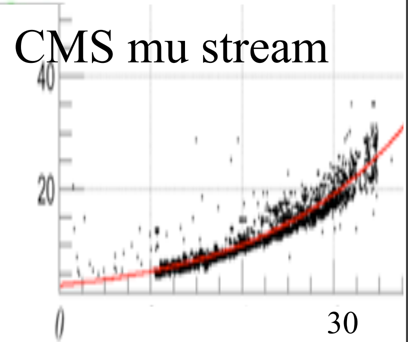


- Run 1 : 2010-2012
- Run 2 : 2014-2018 : energy x 2, pile-up ~50 instead ~25, N événements x2
- HL-LHC >2025 : pile-up ˜150, N events x 10
- Flat resources (in euros) and Moore's law give us a factor 10 in CPU power (if and only if we can use the processors as efficiently as today!)
- ➔handling HL-LHC event added complexity, and maintenance/improvement of processor efficiency rely on software improvements. If not, impact on physics.

Impact of pileup on reco

Prompt reconstruction CPU time vs pileup

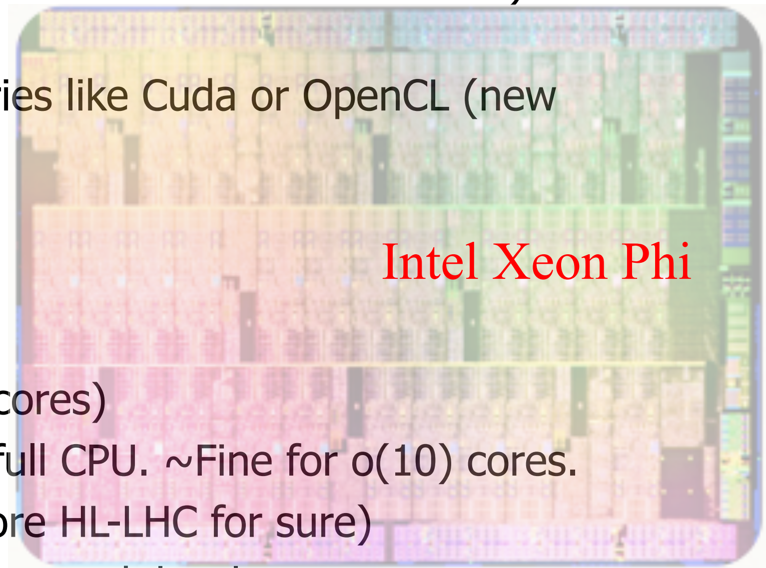




150→

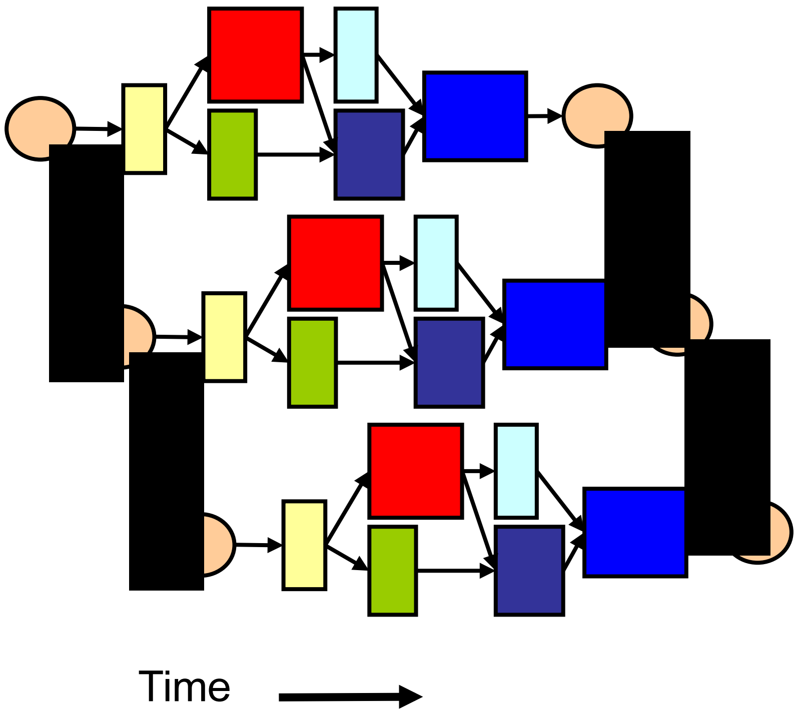
CPU Context

- Remember: no more CPU frequency gain since ~2005
- Two orthogonal avenues (in addition to traditional algorithm improvement):
- Micro-parallelism
  - Modern cores are not used efficiently by HEP software. Many cache misses. SIMD (Single Instruction Multiple Data), ILP (Instruction Level Parallelism) etc… to be used
  - Specialised cores like GPU require use of libraries like Cuda or OpenCL (new languages effectively)
  - Expert task. Focus on hot spots.
  - Immediate benefit on performance
- Macro-parallelism
  - More cores per CPU (and possibly specialised cores)
  - So far largely ignored : treat one core as one full CPU. ~Fine for o(10) cores.
  - Will break down for o(100) cores (when ? before HL-LHC for sure)
  - ➔calling for macro-parallelism, handled at framework level
  - Mitigates impact on sw developers if framework is smart enough
  - However does not help throughput if already enough I/O and memory



Event level concurrent event processing

a.k.a the Holy Grail



- The framework process several events simultaneously…
- …distributes intelligently algorithms to cores
- can allocate more cores to slowest algorithms
- can optimise use of specialised cores

- In addition to algorithm scheduling, the framework provides services to pipeline access to resources (I/O, conditions, message logging…)
- Algorithms should be thread safe : no global object (except through the framework), only use thread safe services and libraries
- Algorithms do not need to handle threads themselves
- ➔regular software physicist with proper training can (re)write algorithms

Analysis chain

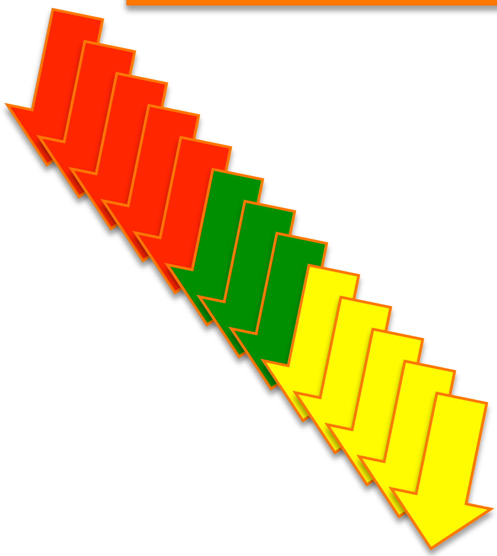
10^{14} collisions

Trigger

10^{9} events on disk

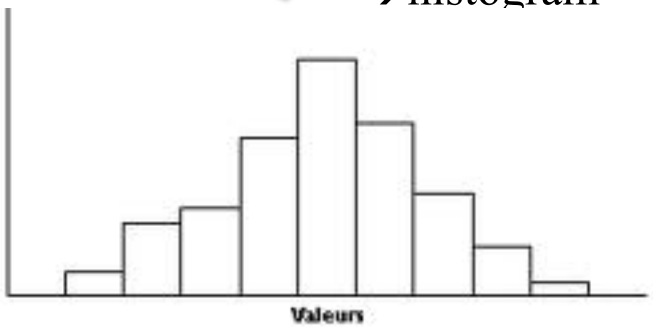
Selection

10^{5} two gamma events



gamma-gamma
mass calculation
→histogram

Hundreds different analyses in ATLAS



Analysis cycle

x Petabytes

Creativity in physics analysis : having a lot of ideas…and…being able to test them quickly

KiloBytes

RAW

How much time does it take to redo a plot ? (new cut, new variable,…)
How much time does it take to redo a full analysis ? (proper reweighted plots and values after all corrections)

plot

1-3 months
Every
4-12 months

Few seconds
every minute
root

Petabytes

Analysis objects

Difficulty : ~200 analysis streams/teams, 2000 users sharing same resources

dedicated Ntuple

Giga/MegaBytes

One or more dedicated intermediate datasets with event selection and information reduction. Balance between completeness, **resource** usage, speed to use and speed to reproduce the dataset.

Summary

- High Luminosity LHC (>2025) : complexity x10, N events x 10 ➔ needs x 100
- Flat budget : resource gain x 10
- ➔a factor 10 to win
- Need to introduce micro and macro parallelism, in 5 millions lines of code (per experiment), maintained by hundred of physicists (part time developers)
- More and more common (=across experiments) software developments (not just Root and Geant4)

The Higgs Machine Learning Challenge

David Rousseau (LAL-Orsay)

The Higgs Machine Learning Challenge

- Why not put some Atlas simulated data on the web and ask data scientists to find the best machine learning algorithm to find the Higgs ?
  - Instead of HEP people browsing machine learning papers, coding possibly interesting algorithm, trying and seeing whether it can work for our problems
- Challenge for us : make a full ATLAS Higgs analysis simple for non physicists, but not too simple so that it remains useful

Machine Learning?

- Machine Learning is the part of computer science which, in particular, deals among other with automatic classification:
  - E.g. neural nets to read handwritten digits (~30 years ago)
  - In HEP we call it MVA (Multi Variate Analysis)
- Developing rapidly
  - Lots of data to deal with
  - Lots of CPU power too
  - Big money involved:
    - google advertisements based on your searches or your gmail messages
    - amazon : "we recommend for you" based on what you already bought
  - "Big data" buzz word
  - New field "Data Science"

Goals

- 1) Favor collaboration between HEP and Machine Learning :
    - HEP people become aware that there might be a friendly data scientist on their campus who would love to apply new ML techniques to HEP problems
- 2) New classification algorithms might perform significantly better than TMVA BDT➔they will eventually be integrated in TMVA and used routinely in ATLAS instead of the current BDT
- 10) Discovery of a magic variable allowing better signal/ background separation in Htautau analysis

How does it work ?

- From may to september 2014 (opening Tuesday 13th May (date finalised after this meeting)
- People register to Kaggle web site : https://www.kaggle.com/c/higgs-boson. Open to almost any one
  - Data scientist
  - HEP physicists
  - Students, geeks, high school teachers with their class etc… (maybe a bit difficult for high school, this will not be the new 2048)
  - Except LAL-Orsay employees (for legal reasons)
- …download training dataset (with label) with 250k events
- …train their own algorithm to optimise the significance (à la s/sqrt(b))
- …download test dataset (without labels)
- …upload their own classification
- The site automatically calculates significance. Public (100k events) and private (450k events) leaderboards update instantly.
- Competition closes mid september 2014. People are asked to provide their code and methods. Best 1 2 3 from private leaderboard win 7k€ 4k€ 2k€
- The most interesting one gets the "HEP meets ML award"

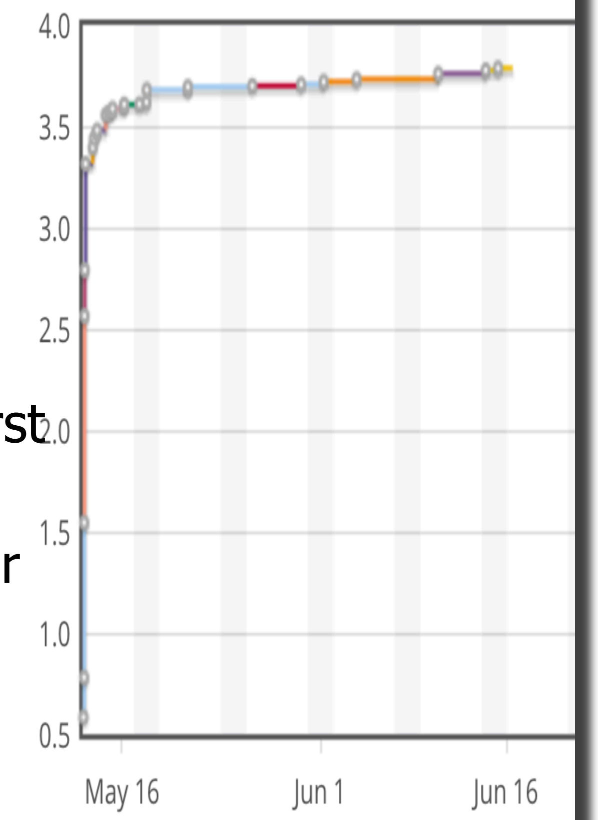
Funded by: Paris Saclay Center for Data Science, Google, INRIA

What data did we release ?

- From ATLAS full sim Geant4 MC12 production
- 30 variables
- Events from Htautau signal, and backgrounds : Ztautau, tt, W
- Preselection as in released ATLAS note for lep-had topology : single lepton trigger, one lepton identified, one hadronic tau identified
- →800.000 events:
  - 250.000 training data set
  - 550.000 test data set without label and weight
- The real analysis uses manipulated data events : embedded (real Zmumu with G4 tau desintegration), anti-tau

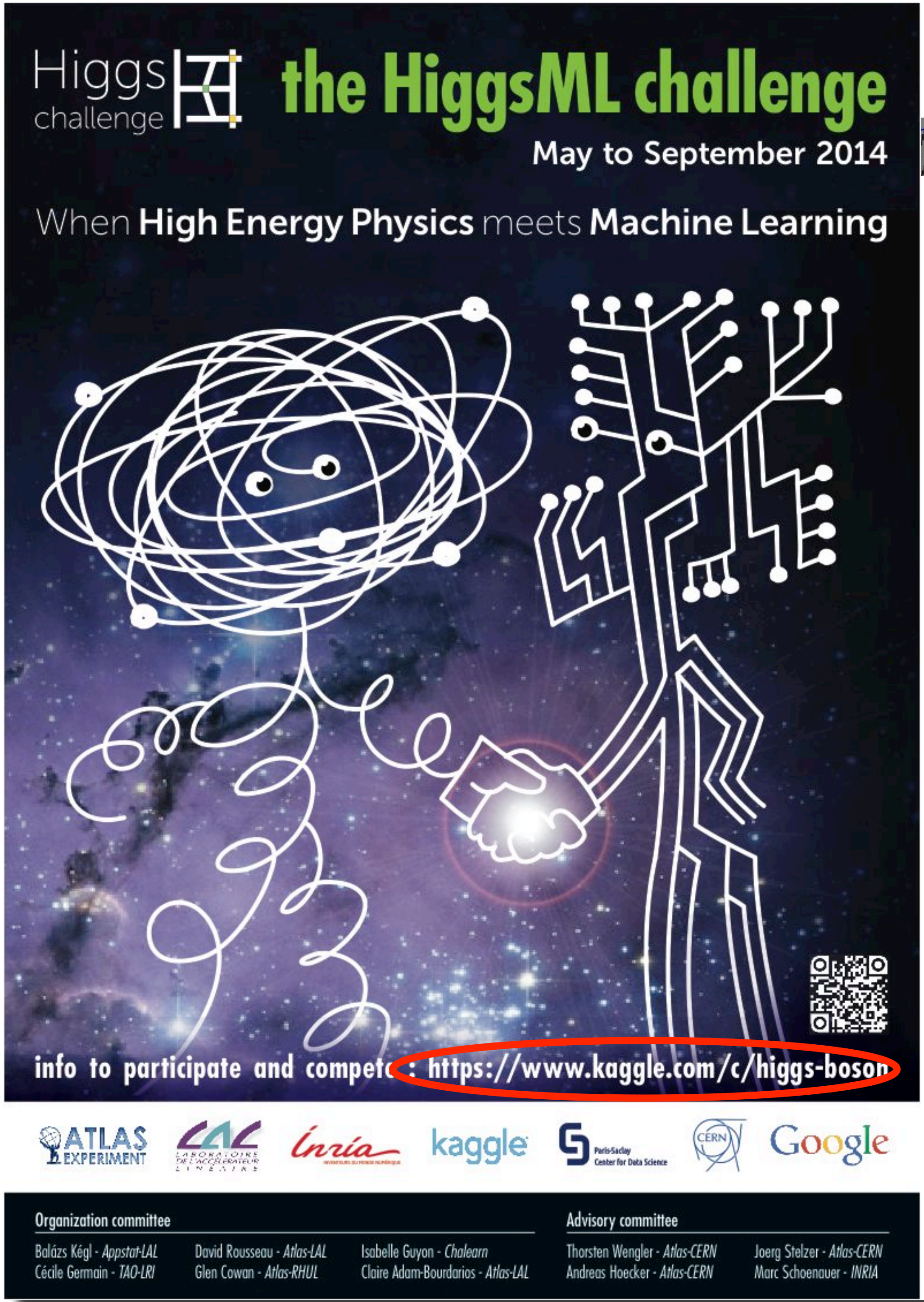
Status today

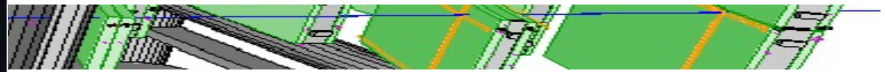
- 717 players in 671 teams registered (have submitted at least one solution)
- The training kit has been downloaded by 3455 users.
- 30k unique user hits on our web page
- Score (significance ~s/sqrt(b)):
  - TMVA benchmark 3.2 (TMVA BDT out of the box, no attempt of any tuning), beaten the first day
  - Multiboost benchmark 3.4 (ditto), beaten after two days
  - 3.6 was reached after a few days
  - Now at 3.78



Future steps

- Challenge closes 15th September
- Participants have two weeks to submit their software
- Prize announcements 15th October
- Need to prepare the examination of the submissions
- We are applying for an organisation of a satellite workshop in NIPS 2014 (December at Montreal)
- Thinking of organising an HEP meets ML event at CERN when the award winner comes

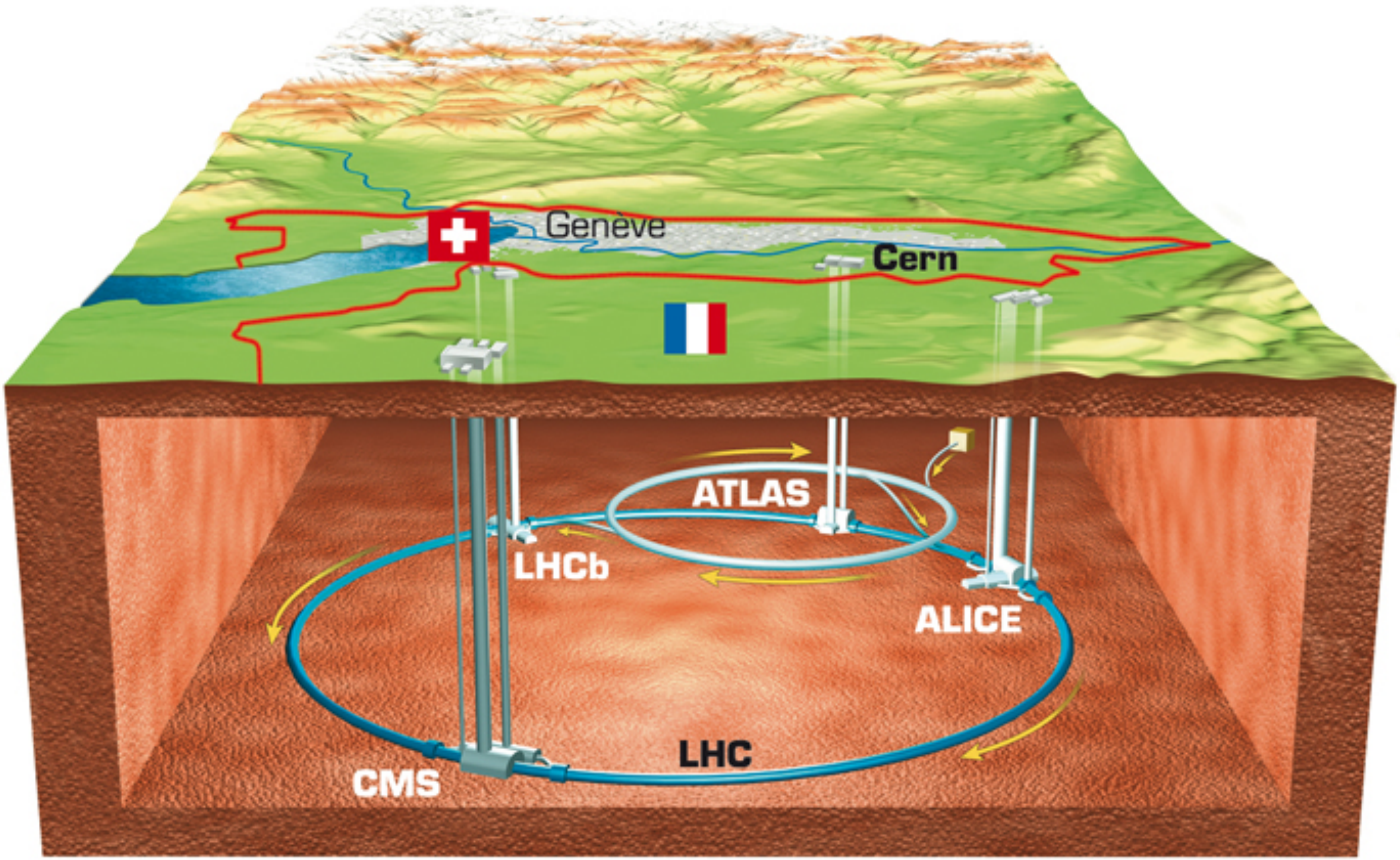

Higgs challenge
the HiggsML challenge
May to September 2014
When High Energy Physics meets Machine Learning
info to participate and compete : https://www.kaggle.com/c/higgs-boson
ATLAS EXPERIMENT
kaggle
Google
Organization committee
Balázs Kégl - Appstat-LAL
Cécile Germain - TAO-LRI
David Rousseau - Atlas-LAL
Glen Cowan - Atlas-RHUL
Isabelle Guyon - Chalearn
Claire Adam-Bourdarios - Atlas-LAL
Advisory committee
Thorsten Wengler - Atlas-CERN
Andreas Hoecker - Atlas-CERN
Joerg Stelzer - Atlas-CERN
Marc Schoenauer - INRIA



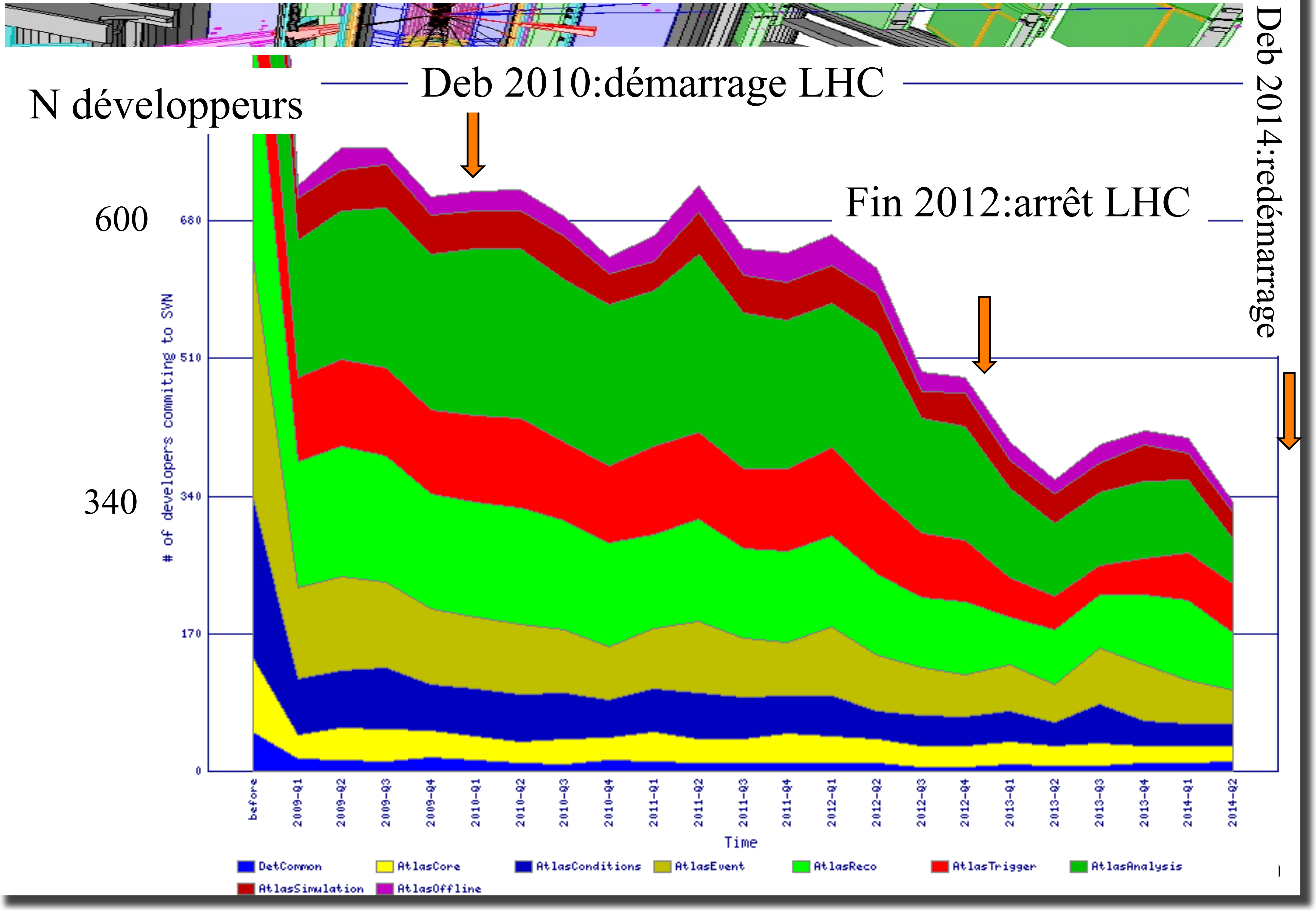
Thank you !

Back up + some slides from Weekly meeting talk

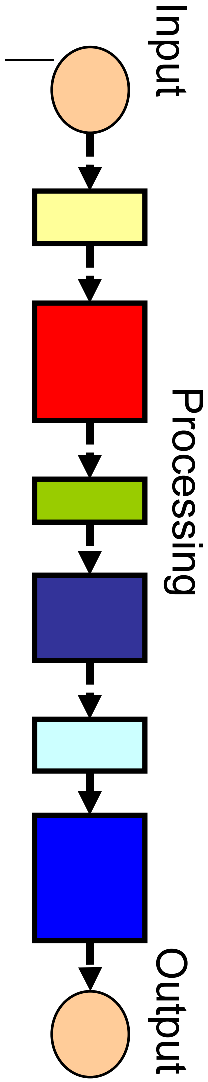
Le LHC

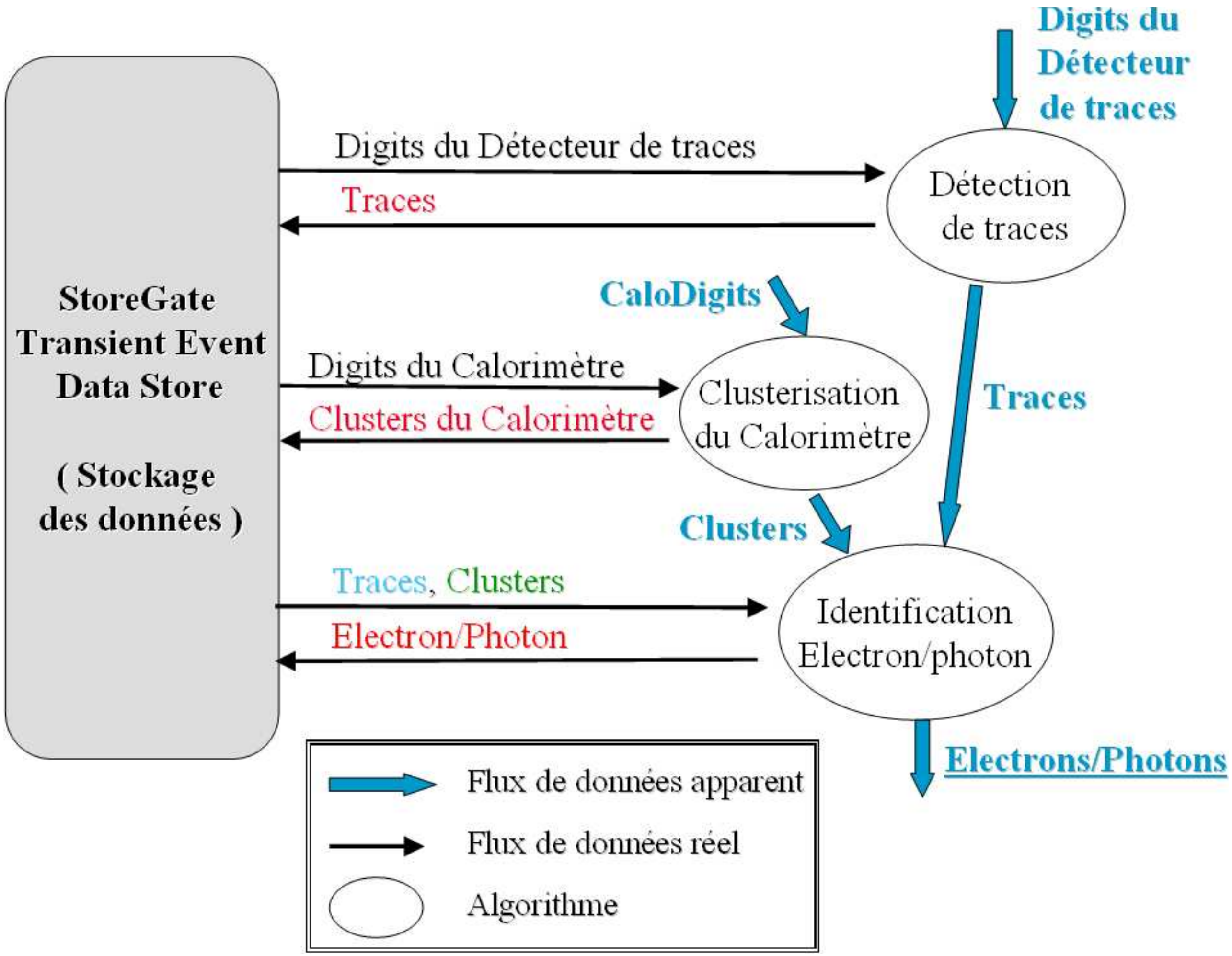


Développement durable ?



Architecture tableau noir

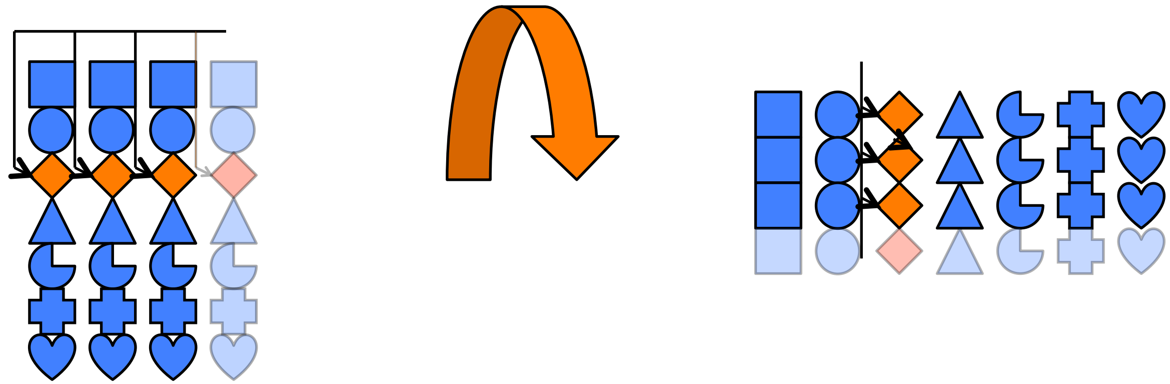




Note on data organisation

Array of objects

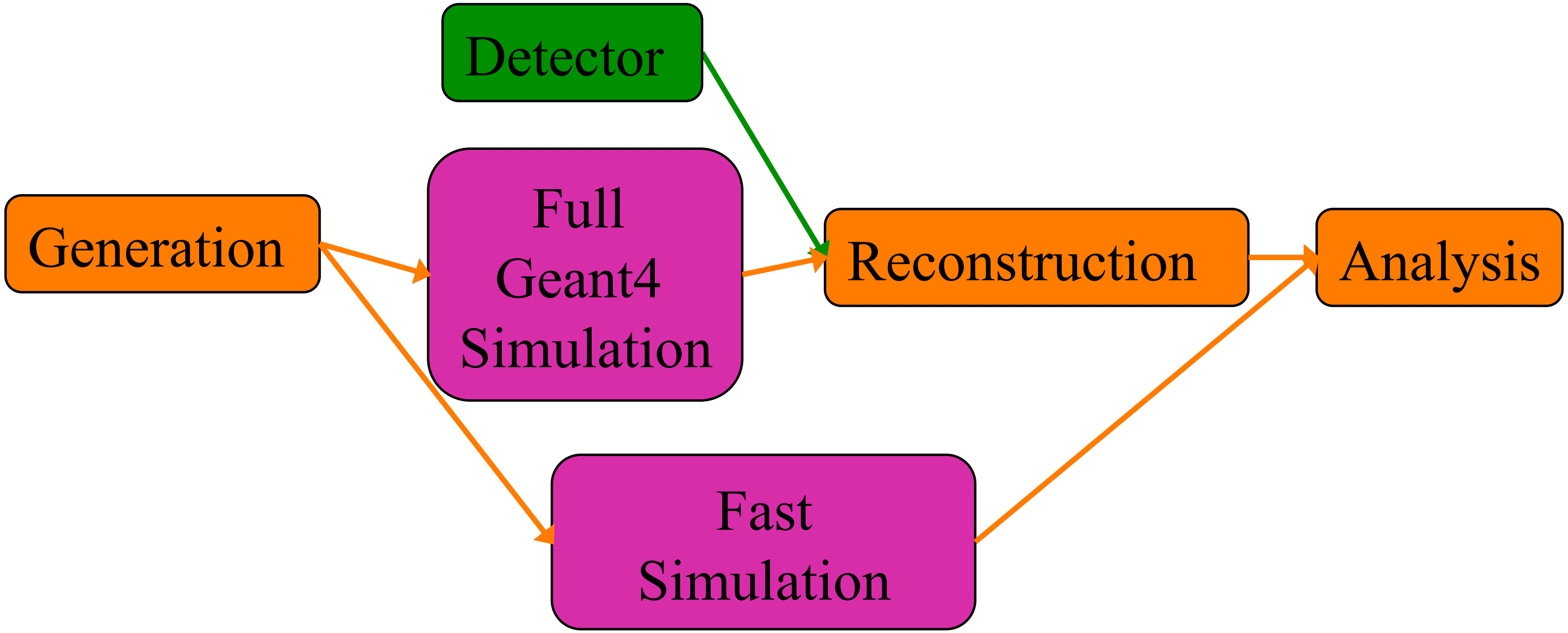
Struct of arrays



➔ More suitable for vectorisation

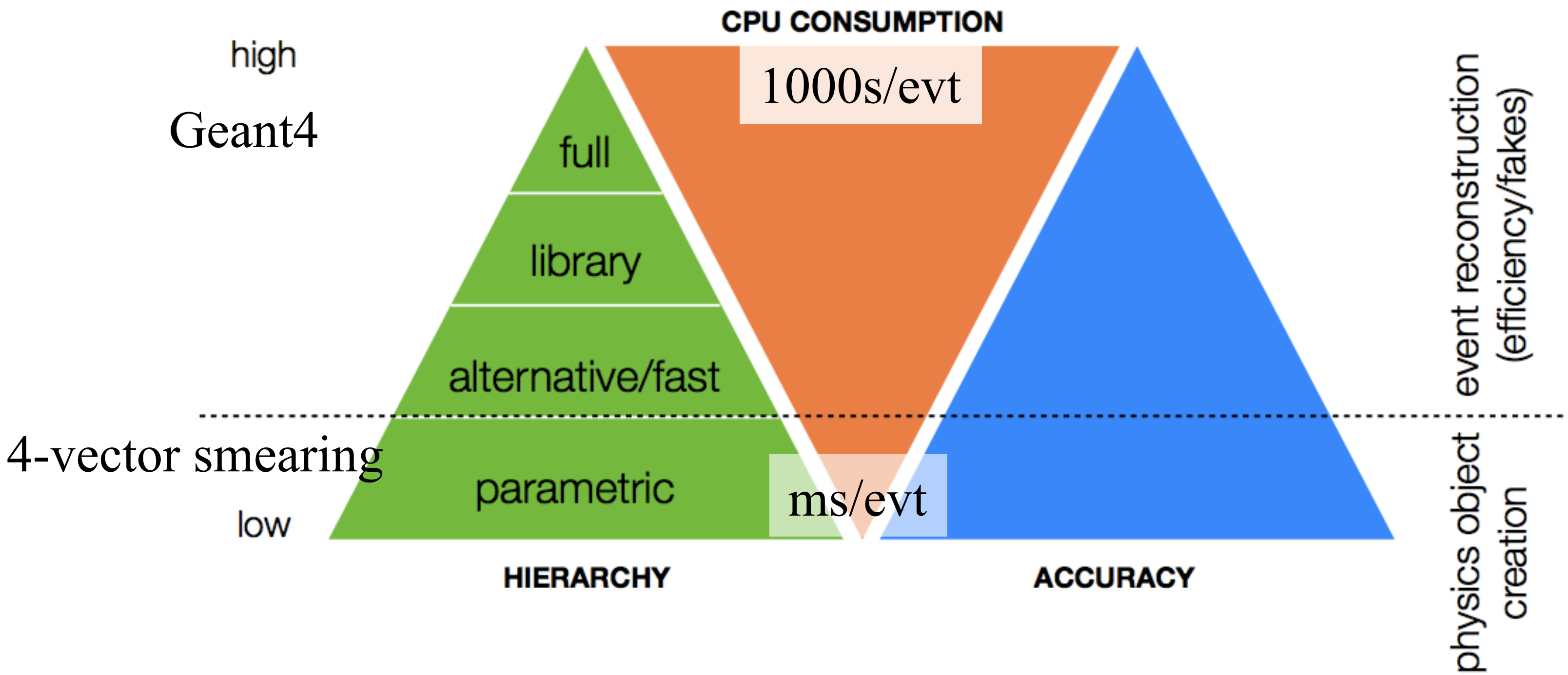
- Data organisation often need to be completely revisited prior to algorithm vectorisation
- (may improve performance even without vectorisation due to better locality (less cache misses))

Processing steps



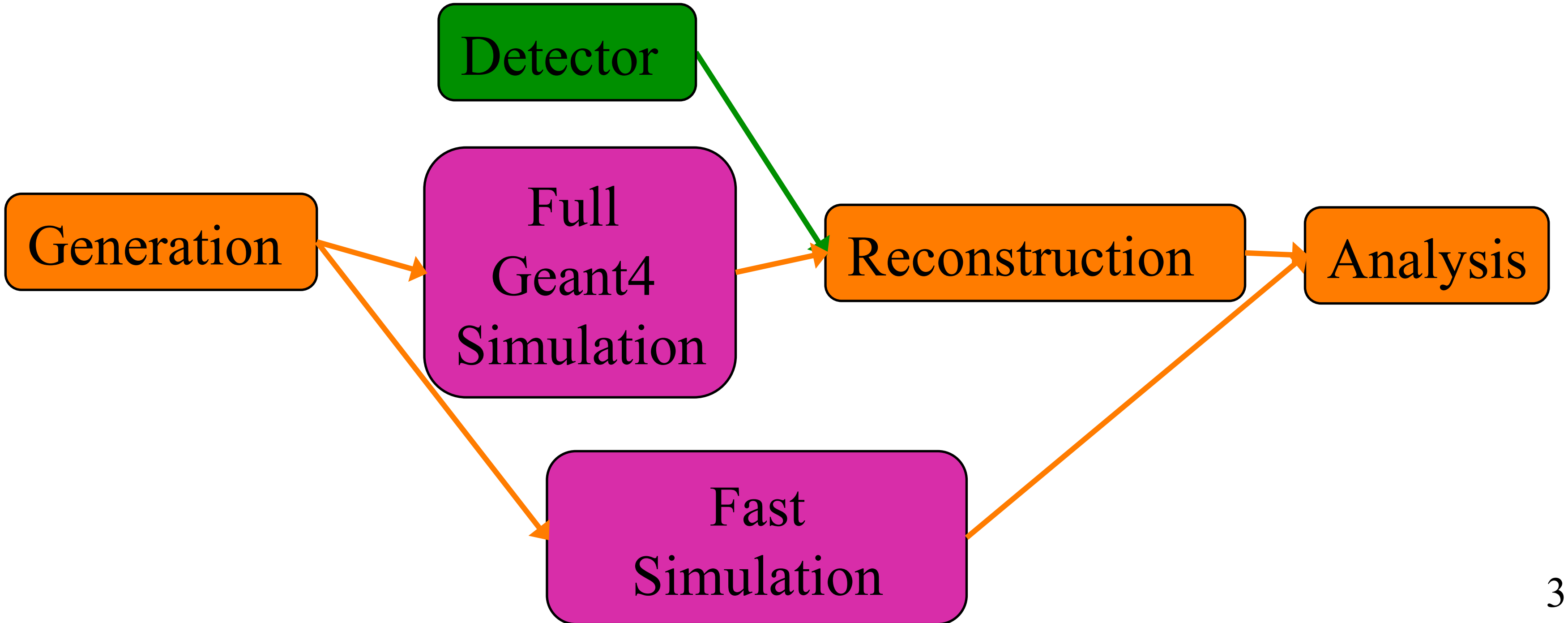
Simulation

- Dominates CPU consumption on the grid
- HL-LHC : 10x read-out rate➔10x n events simulated ? Even more due to increased requirement on precision
- Continue effort on Geant4 optimisation:
  - G4 10.0 multi-threaded released Dec 2013
  - Re-thinking core algorithms with vectorisation in mind
- Rely on blend of G4/Fast sim/Parametric. Challenge : the optimal blend is very analysis dependent. But only one pot of resources.



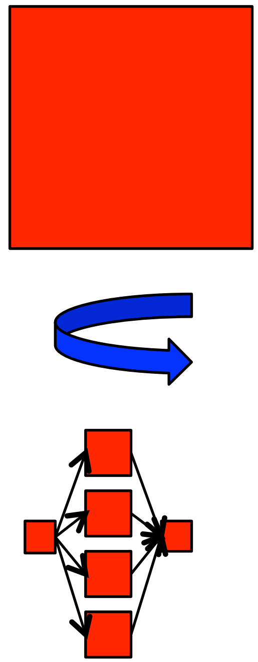
Atlas sw in a nutshell

- Generators: :Generation of true particle from fondamental physics first principles=>not easy, but no sw challenge
- Full simulation :Tracking of all stable particles in magnetic field through the detector simulating interaction, recording energy deposition. (CPU intensive)
- Reconstruction : for real data as it comes out of the detector, or Monte-Carlo simulation data as above
- Fast simulation : parametric simulation, faster, coarser
- Analysis : Daily work of physicists, running on output of reconstruction to derive analysis specific information (I/O intensive)
- All in same framework (Gaudi/Athena), except last analysis step in Root. All C++.



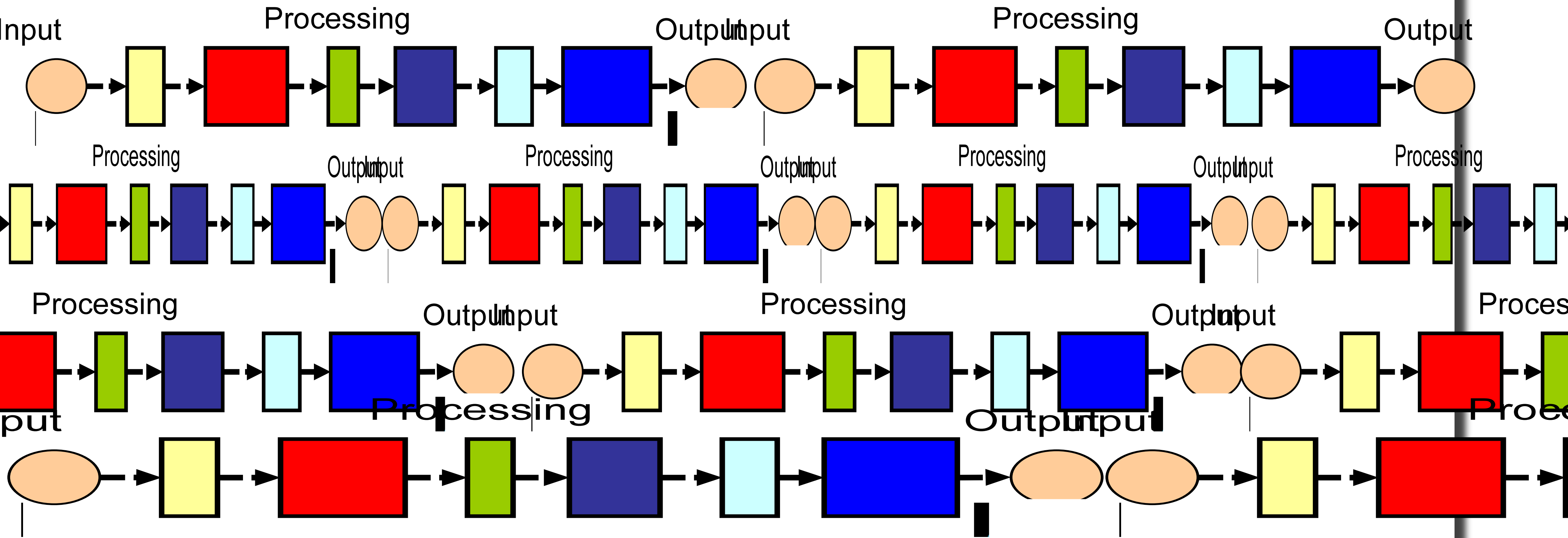
Note on multi-threading

- One possible answer is to use parallel multi-threading within algorithms
  - E.g. the tracking algorithm spawns multiple threads, each one reconstructing tracks in an eta-phi region
  - Test jobs on empty processor will effectively run (much) faster
  - However, in a grid environment, with one job per core, the multiple threads will compete with the other jobs running on the same processor➔no good!



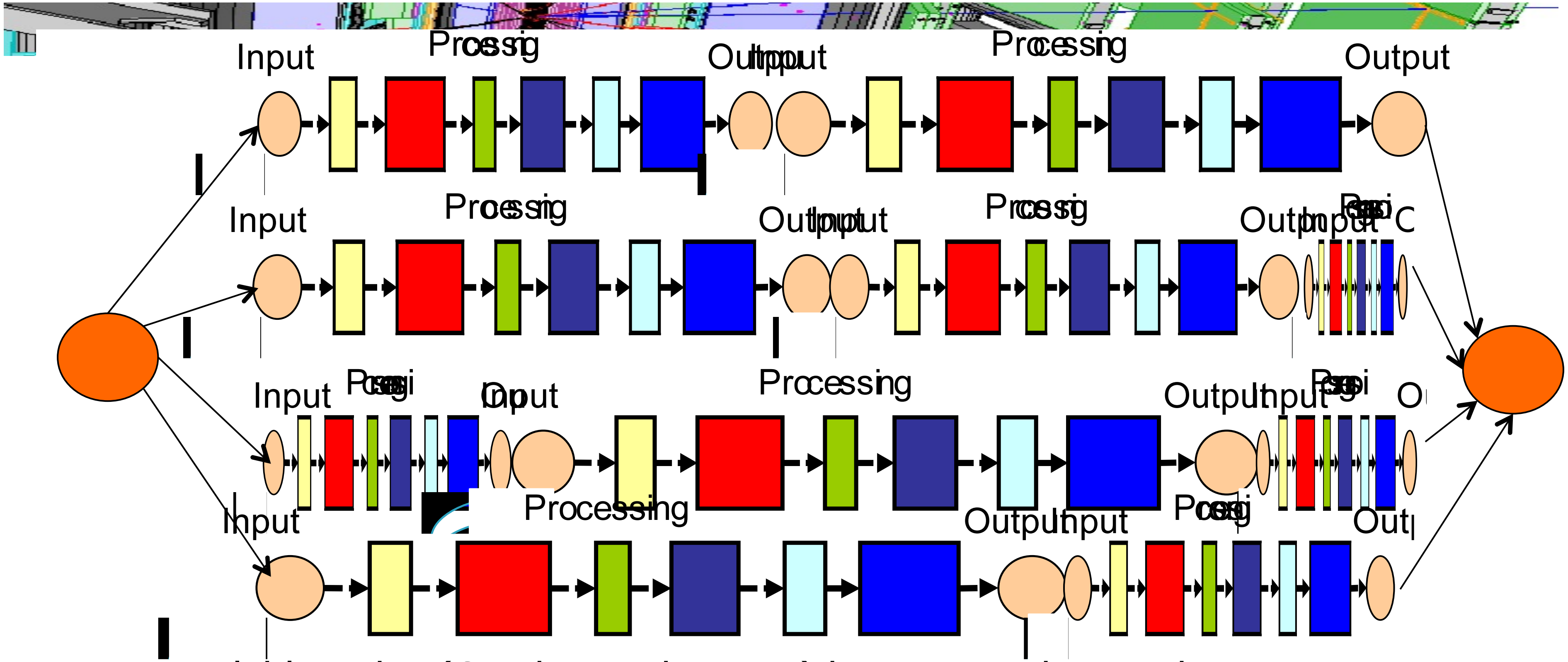
- Multi-threading useful if done at framework level, in an organised way (➔)

One core one job



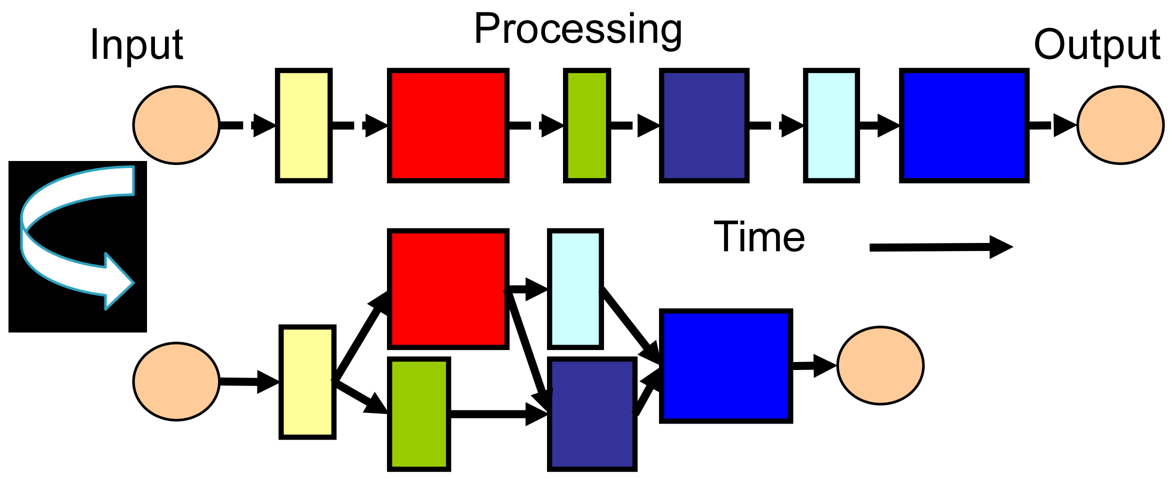
- Today, typical grid workhorse is a 16GB memory, 8 core CPU (2GB/core) (3GB/core is now common, but not sustainable in the future)
- Each core is adressed by the batch system as a separate processor
- Each job process event one by one, running one by one a finite number of algorithms
- One processor may handle simultaneously e.g. one Atlas reco job, 3 CMS simulation job, and 4 LHCb analysis jobs
- This works (today), however disorganised competition for resources like memory, I/O

One processor one job



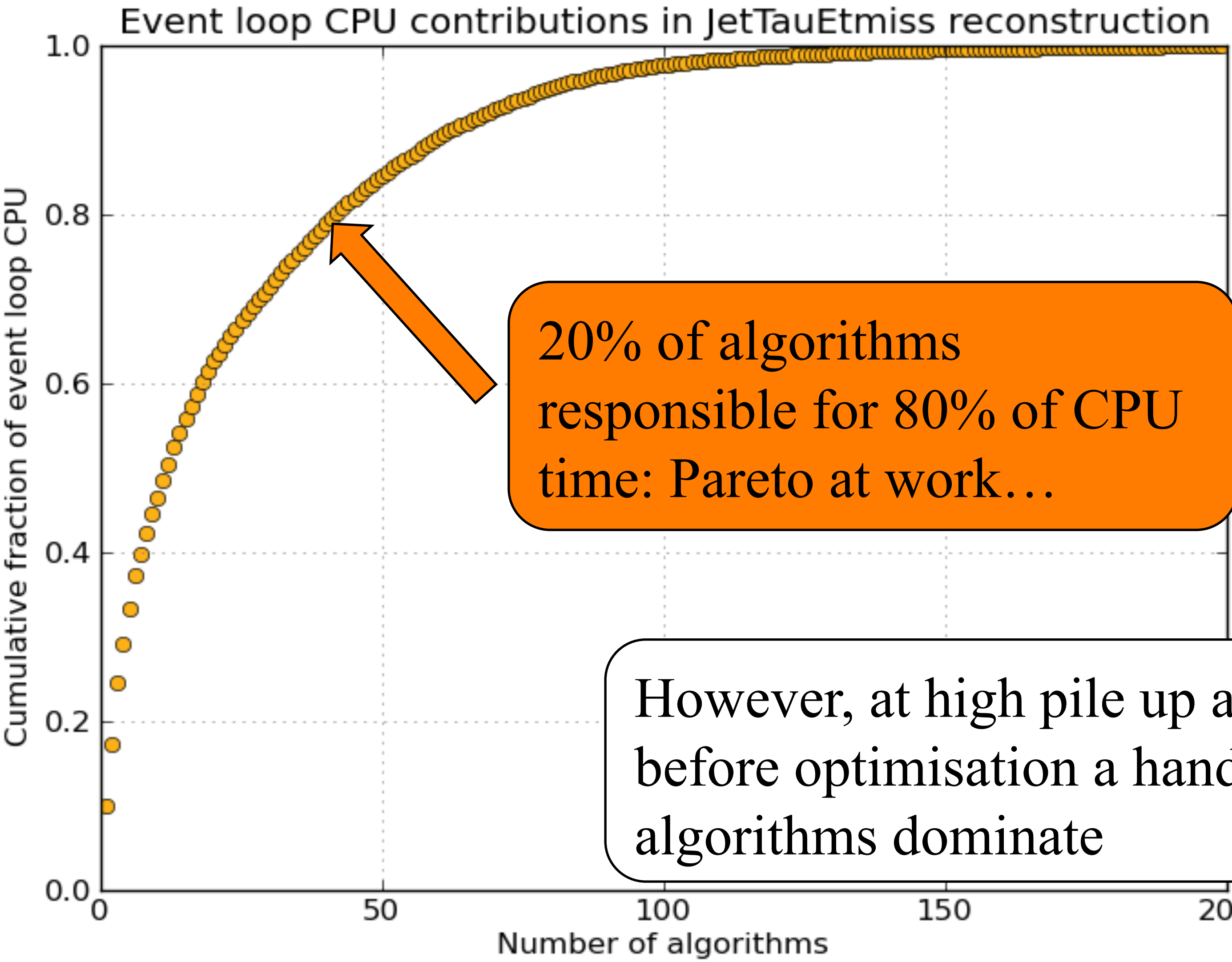
- Available today (GaudiMP, AthenaMP) but not used in production yet
- One job goes to one processor (which is completely free)
- The framework distributes event processing to all cores, while sharing common memory (code, conditions,…) using Copy-on-Write
- No change to algorithmic code required (in principle)
- ~50% reduction of memory achieved (w.r.t. independent jobs)

Event level parallelism



- framework schedules intelligently the algorithms from their dependency graph
- e.g. run tracking in parallel with calorimeter, then electron ID
- in practice too few algorithms can run in parallel (amdahl's law)
- ➔most cores remain idle

CPU per algorithm (reco)



20% of algorithms responsible for 80% of CPU time: Pareto at work…

However, at high pile up and before optimisation a handful of algorithms dominate

LHC experiments code base

- LHC experiments code base
  - ˜5 millions line of code per experiment
  - written by ˜ 1000 people per experiment since ˜15 years

- Who are they ?
  - Very few software engineers
  - Few physicists with very strong software expertise
  - Many physicists with ad-hoc software experience
- All these people need take part to the new transition

Software for GPU

- Graphic co processor massively parallel, up to x100 speed-up on paper
- In practice, task must be prepared by traditional CPU and transferred to GPU
- Successfully used in HEP for very focussed usage e.g. Alice trigger tracking (gain factor 3 in farm size), now also in other experiments
- Code need to be written from scratch using libraries such as Cuda, etc…
- …and largely rewritten/retuned again for different processors, generation
- Physics performance not as good as original code
- Usage on the grid unlikely/difficult due to the variety of hardware
- ➔Need to maintain a second, traditional, version of the code
- In the future, expect progress in generic libraries (e.g. OpenCL) which would ease maintenance (one code for all processors) at an acceptable loss in performance

GridCL

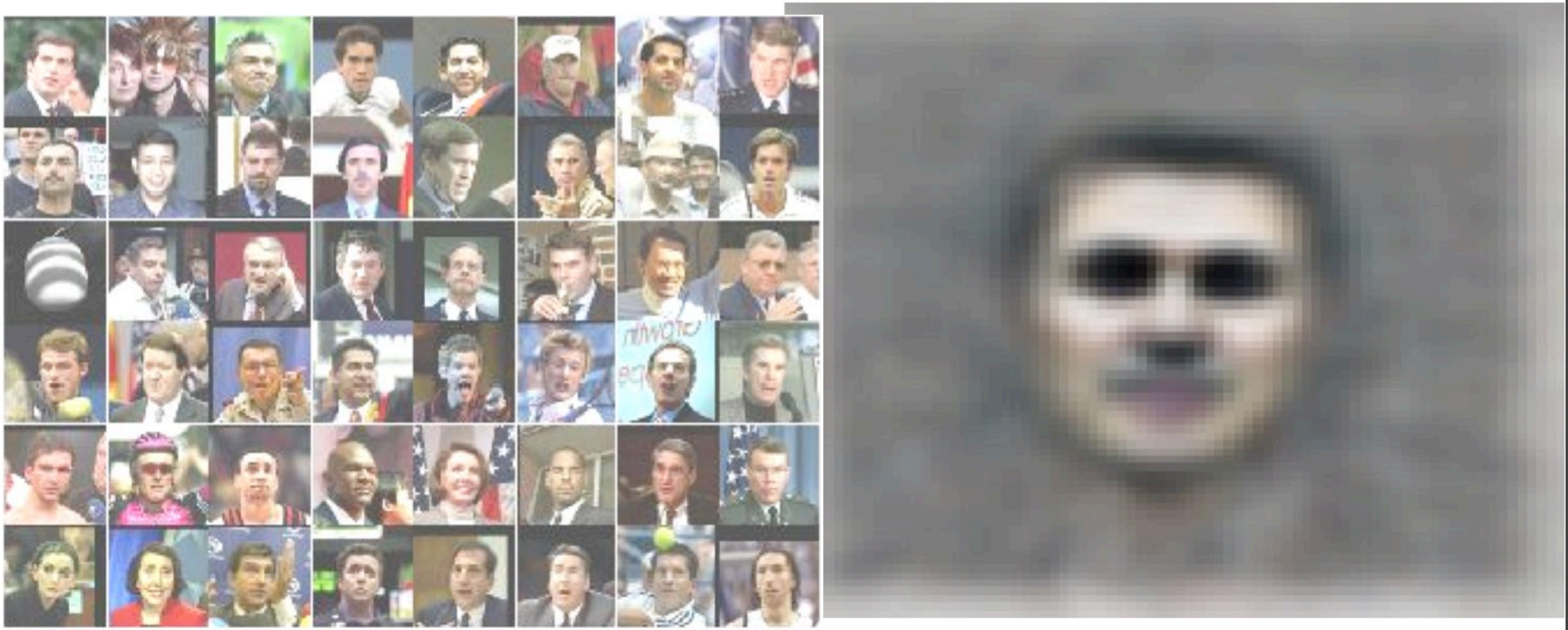
- GridCL : local project
- Objectif premier : Evaluer l'utilisation de matériel many-core hétérogène, via OpenCL, au sein d'une grille.
- Financé 2012 par P2IO (labex Physique des 2 Infinis et des Origines)
- Partenaires : LLR, IAS, LAL, IMNC, IRFU, IPNO, LPT...
  - La porte est ouverte, contacter David Chamont (LLR)
- Ressources partagées:
  - 2 noeuds sandy-bridge, dotés chacun de 2 NVidia K20, connectés en Infiniband.
  - 2 noeuds sandy-bridge, dotés chacun de 2 Intel 5110P, connectés en Infiniband.
  - Bientôt : 1 noeud ivy-bridge, doté de 6 NVidia Titan.
  - Logiciel : OpenCL, CUDA 5, Intel Cluster Studio XE, CAPS OpenACC.
- Expérimentations
  - Pour CMS : reconstruction des traces dans les collisions d'ions lourds. o Pour CTA : traitement de signaux de télescope.
  - Pour SDO : traitement d'images satellitaires sur matériel hétérogène. o Evaluation et enrichissement du banc d'essai SHOC.
- Dissémination
  - Atelier OpenMP/MPI/OpenCL aux JDEVs 2013
  - A venir : formation sur les outils de profilage Intel

LPaSo

- Candidature ANR « défi de tous les savoirs », préselectionnée 2014
- Objectif : Gagner en performance en exploitant les cœurs multiples, les instructions vectorielles, les accélérateurs, ..., afin de pouvoir absorber la montée en luminosité du LHC.
- Partenaires:
  - LAL (ATLAS, LHCb), LLR (CMS), IRFU/SPP (ATLAS)
  - LRI, LIMOS
  - LPNHE (LHCb), LPC-Clermont (ATLAS)
  - LPTHE
- Thématiques
  - Parallélisme de tâches avec GaudiHive (ATLAS, LHCb).
  - Déclenchement haut-niveau avec GPUs (LHCb).
  - Traitement de données avec accélérateurs (CMS).
  - Parallélisation/vectorisation des outils d'analyse statistique.
  - Parallélisation/vectorisation de FastJet.

BDT and Machine Learning

- BDT (Boosted Decision Tree) which is by far the most used technique in Atlas is actually an old technique (Adaboost 1997)
- More recent techniques, just an example:
  - Unsupervised neural network (Example: The Google cat: deep learning technique running on 16K cores for three days, watching 10M random YouTube video stills [Le et al., ICML’12]



Challenge ?

- Challenges have become in the last 10 years a common way of working for the machine learning community
- Machine learning scientists are eager to test their pet algorithms on real life problems
- Company or academics want to outsource a problem to machine learning scientist, but also geeks etc. The company sets up a challenge like:
  - Netflix : predict movie preference from past movie selection
  - Gesture recognition
  - Separating pictures of cats from pictures of dogs
  - NASA/JPL mapping dark matter through (simulated) galaxy distortion
  - …
- Some companies makes a business from organising challenges: datascience.net, kaggle

Higgs meet ML award

- The most interesting method for us is not necessarily the absolute best in significance
- Hence we will evaluate the best submissions on criteria:
  - Usability in typical ATLAS analysis
  - Significance
  - simplicity (how many lines to explain in details)
  - CPU and memory demands
  - robustness with respect to lack of training statistics
- The winner will be invited to CERN at ATLAS expenses in winter 2014 (we might think organising "an ATLAS/CERN meets machine learning" event)

Why Higgs to tau tau and how

- The challenge has to have "Higgs" in the name to surf the HiggsMania
- Higgs to tau tau is difficult:
  - Large Z to tau tau background with similar topology (except VBF)
  - Other non neligible backgrounds ttbar, Z/W…
  - Poor mass resolution (>=2 neutrino, poor hadronic tau energy resolution, poor MET)
  - Different categories VBF, boosted,…
- In practice focus on lep-had channel, and use same Geant4 simulation events as in conf note ATLAS-CONF-2013-108 :
  - Use lep-had Common Ntuple (validated for Conf Note)
  - Higgs (125) signal (all production modes)
  - $Z\rightarrow\tau\tau$ background
  - Top and W+jet background
  - No embedded, no fakes from anti-tau : this would be real data, and tricky to explain
- Preselection as in conf note, except:
  - Single lepton trigger
  - B jet veto already applied
- Reproduce reasonably well (~20%) content of 3 highest sensitivity bins (x 2 categories) in backup note
- A baseline running TMVA (v4.1.3 in Root 5.34/03) BDT out of the box using all the variables (next slide) is provided. (no attempt to do any tuning of BDT configuration, choice of better variables…)

Significance

- Need to have one robust estimator of the quality of the classification algorithm
- Decided to use the well known (in ATLAS) "Asimov" formula (G. Cowan, K. Cranmer, E. Gross, and O. Vitells, "Asymptotic formulae for likelihood-based tests of new physics", *EPJCC*, vol. 71, pp. 1–19, 2011.) with regularization on top
  - $\sqrt{(2*((s+b')*\log(1+s/b')-s))}$
  - with s and $b'=b+10$ normalised to 2012 luminosity:
  - $s=\Sigma$(selected signal) weights_i
  - $b=\Sigma$(selected background) weights_i
- Why $b'=b+10$ ("regularisation") : practical way to avoid large significance fluctuation when small phase space region with very few background events is chosen. Do not want to pick winners on their luck.
- Note that normalisation already done in the weights : no need to explain integrated luminosity and cross-section
- Glen Cowan has derived a new version of Asimov formula including a sigma_b (to be shown at coming Statistics Forum) from systematics or statistics
  - However not robust enough in our case

Time line

- Sep 2012 : first idea, discussions in Orsay
- April 2013 : DR presentation at HSG4 workshop in Corfu
- Summer 2013 : work on significance and dataset, understand each other background and vocabulary
- October 2013 : start serious legal discussion with Kaggle, ATLAS management, CERN (who legally owns ATLAS simulated events) management , LAL (who formally organizes the challenge) management
- April 2014 : contract finalised
- 29th April 2014 : DR talk at Atlas weekly

Now

- 13th May 2014 (date finalised after this meeting): beginning of the challenge
- 13th May 2014 11AM : Balazs Kegl, chair of the new Center for Data Science Paris Saclay, CERN-EP seminar "Learning to discover: machine learning in high-energy physics »
- June 2014 : LHCP @ New York poster
- July 2014 : abstract in ICHEP (outreach session)
- 15th Sep 2014 : end of the challenge
- Fall 2014 : ATLAS price winner comes to CERN
- December 2014 : possible NIPS talk/workshop (the major machine learning conference)

Committees

- Organization committee:

  ATLAS

  - David Rousseau : Atlas-LAL
  - Claire Adam-Bourdarios : Atlas-LAL (outreach, legal matter)
  - Glen Cowan : Atlas-RHUL (statistics)

  Machine Learning

  - Balazs Kegl : Appstat-LAL
  - Cécile Germain : TAO-LRI
  - Isabelle Guyon : Chalearn (challenges organisation)

- Advisory committee:
  - Andreas Hoecker : Atlas-CERN (PC,TMVA)
  - Joerg Stelzer : Atlas-CERN (TMVA)
  - Thorsten Wengler : Atlas-CERN (ATLAS management)
  - Marc Schoenauer : INRIA (french computer science institute)

Licensing issues

- Anyone participating to the challenge had to agree to the rules (https://www.kaggle.com/c/higgs-boson/rules) in particular:
- Software
  - Participants can use whatever software they like, but to win a price, they have to release it under an OS license (so that we can 1) verify it 2) use it)
- Simulated data:
  - All ATLAS real and simulated data belongs to CERN
  - We've had the signature of Bertolucci, CERN director of research to release the data (Thorsten W handled this negociation)
  - Data is made available **only** for the duration of the challenge and **only** for the challenge ("Can I use the data for a master thesis ?" Sorry no.)
  - Note that we are not naïve : copies of the data will not autodestroy…

Real analysis vs challenge

1. Systematics
2. 2 categories x n BDT score bins
3. Background estimated from data (embedded, anti tau, control region) and some MC
4. Weights include all corrections. Some negative weights (tt)
5. Potentially use any information from all 2012 data and MC events
6. Few variables fed in two BDT
7. Significance from complete fit with NP etc…
8. MVA with TMVA BDT

1. No systematics
2. No categories, one signal region
3. Straight use of ATLAS G4 MC
4. Weights only include normalisation and pythia weight. Neg. weight events rejected.
5. Only use variables and events preselected by the real analysis
6. All BDT variables + categorisation variables + primitives 3-vector
7. Significance from "regularised Asimov"
8. MVA "no-limit"

Simpler, but not too simple!