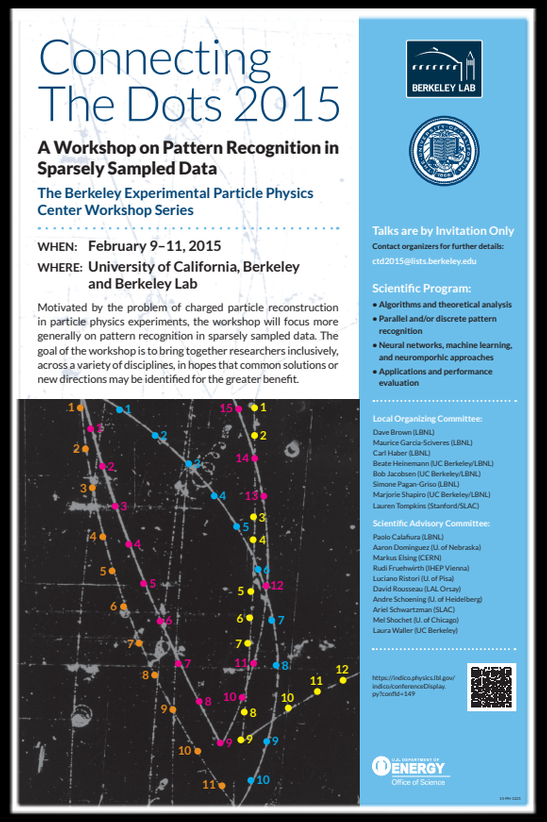
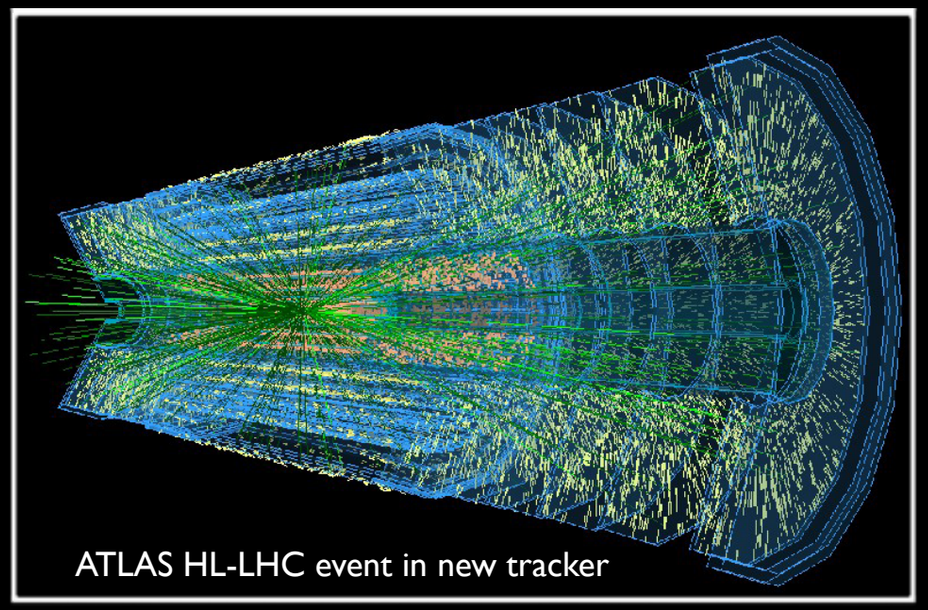




On the "HSF Algorithms Forum"

Markus Elsing

An introduction to creating a "forum" across experiments to discuss and promote the development of reconstruction algorithms

CERN

Introduction: Why an Algorithm Forum ?

- HSF is an initiative to foster collaboration and common developments
  - ➡ builds on several well established common projects (Root, Geant4...)
  - ➡ Concurrency Forum is well established to cover technical software development discussions

- need to as well address the algorithmic problem
  - ➡ especially in view of the resource problems we will face in the future
  - ➡ see processing technology and experiment's software challenges we have to face...

Agreed HSF Goals

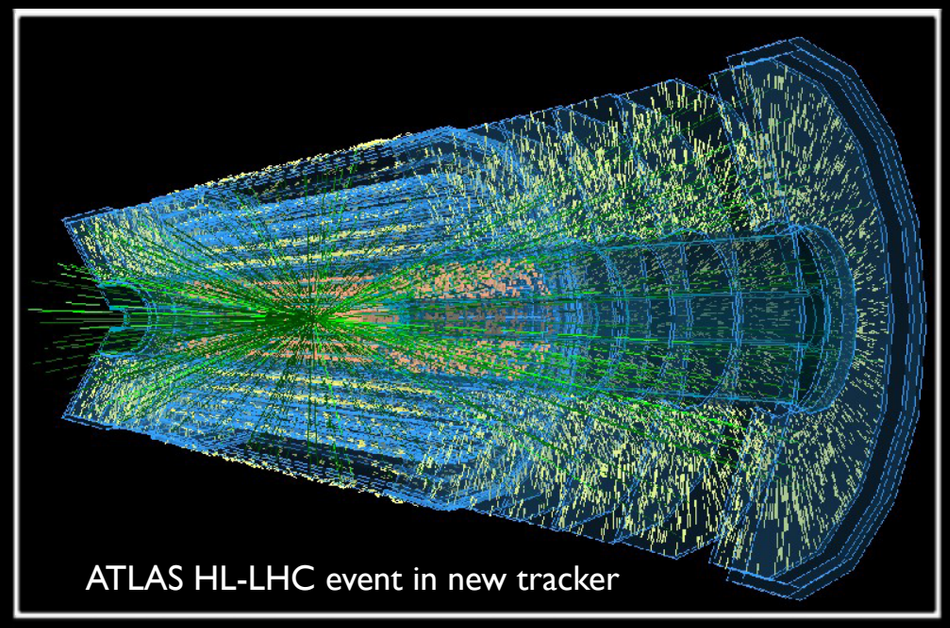
- Share expertise
- Raise awareness of existing software and solutions
- Catalyze new common projects, create an incubator
- Promote commonality and collaboration in new developments to make the most of limited resources
- Aid developers and users in discovering, using and sustaining common software
- Support training career development for software and computing specialists
- Provide a framework for attracting effort and support to S&C common projects
- Provide a structure for the community to set priorities and goals for the work
- Facilitate wider connections; while the HSF is a HEP community effort, it should be open enough to form the basis for collaboration with other sciences

12

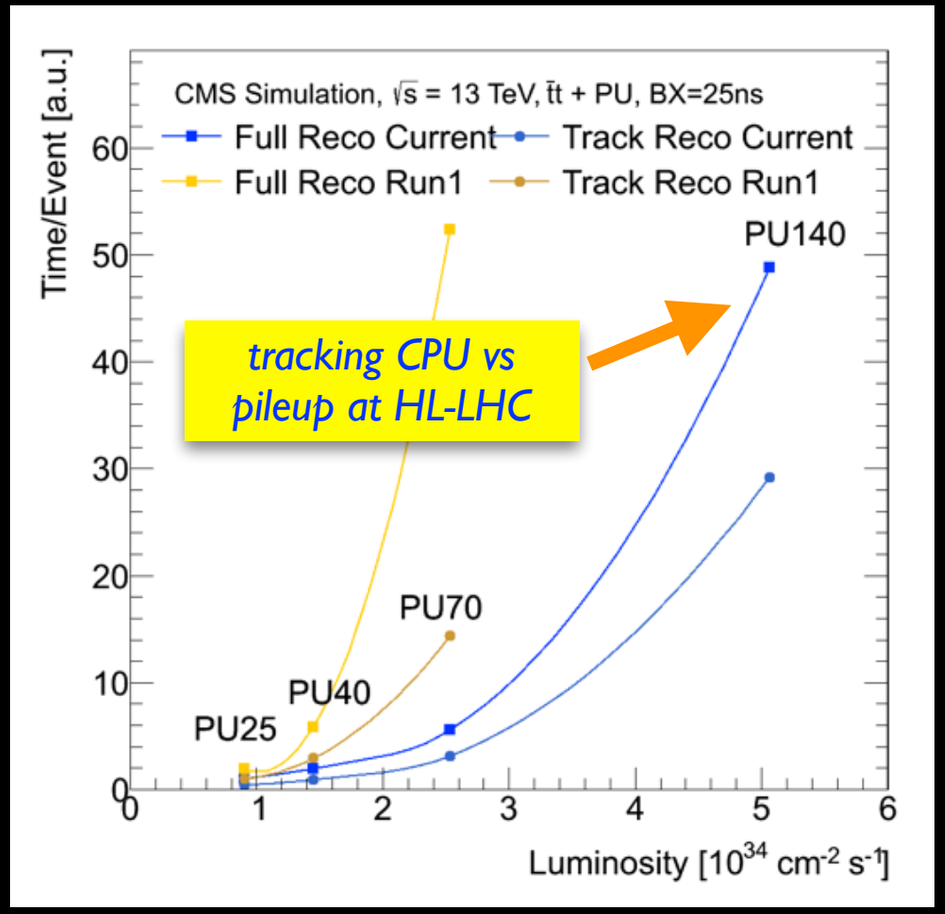
see L.Sexton-Kennedy, WLCG-WS 2015

The Experiments' Software Challenges

- ATLAS/CMS - million dollar question:
  - ➡ how to reconstruct HL-LHC events with 200 pileup
  - ➡ how to keep the physics performance up
  - ➡ and do it within the computing resources we'll have...
- tracking is reconstruction CPU driver
  - ➡ not new, we knew this would be the problem
  - ➡ will aim to improve on already highly optimised code


ATLAS HL-LHC event in new tracker

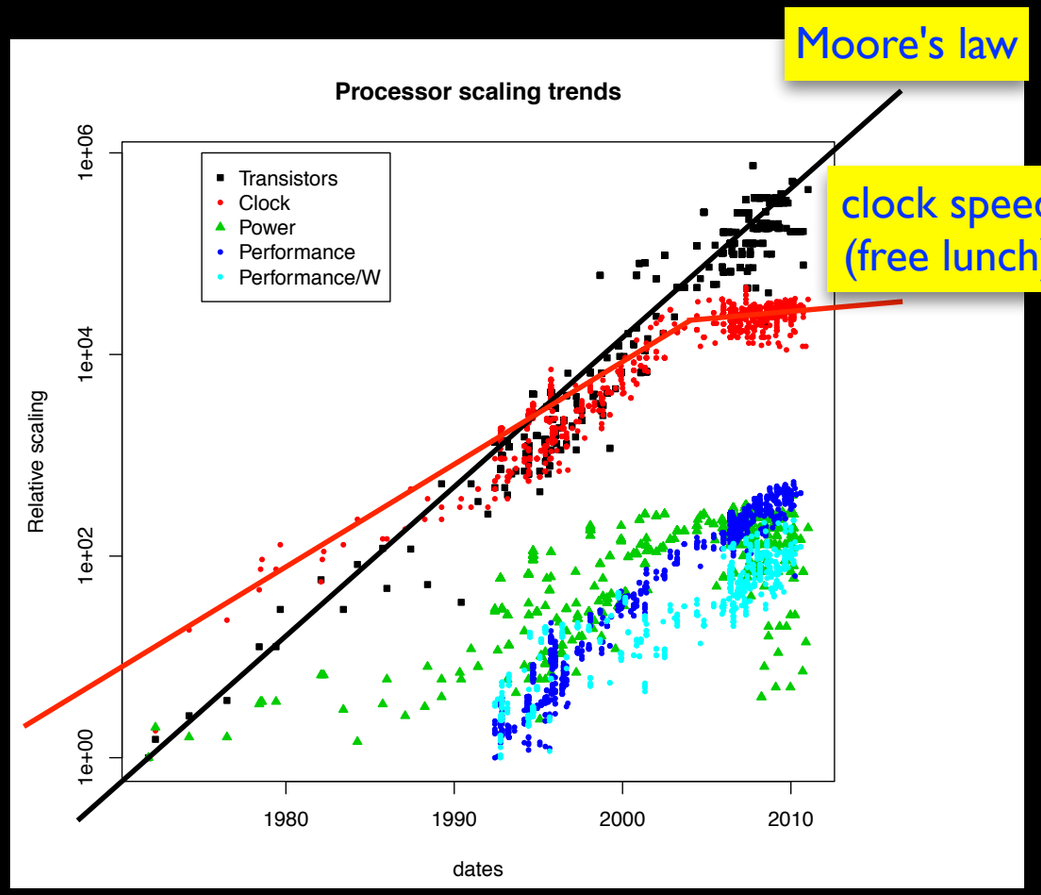
- LHCb and ALICE trigger-less readout
  - ➡ processing/filtering done in online trigger farms
  - ➡ offline quality reconstruction online to achieve needed data reductions
- Belle-II is about to start data taking
  - ➡ raw data volumes comparable to LHC
- Future Collider studies (ILC, CLIC, FCC)



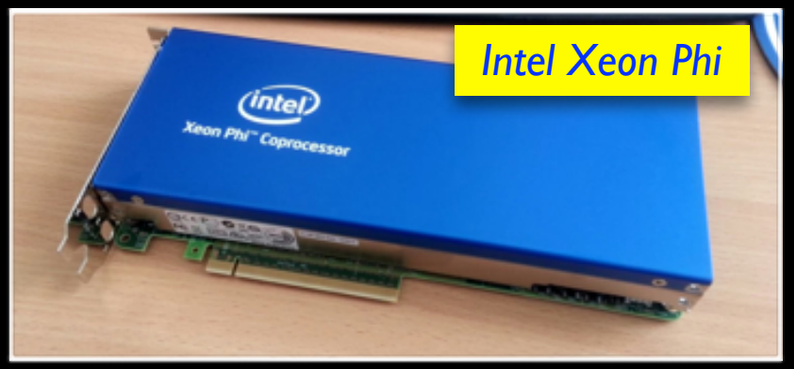
Technology Challenges

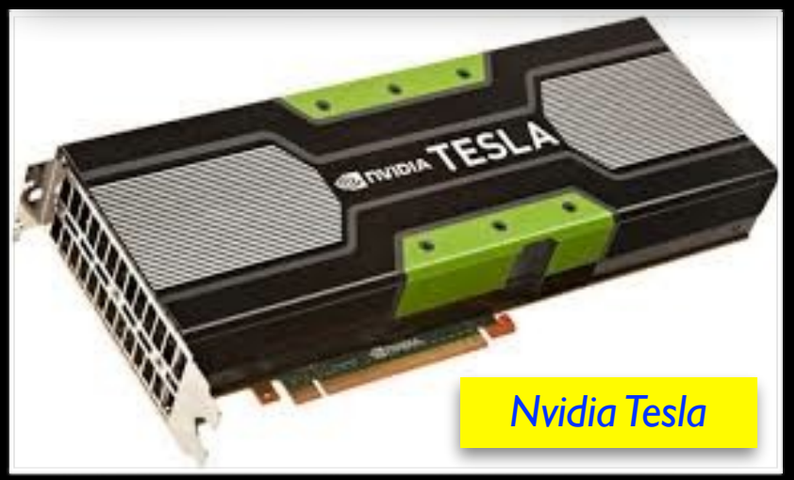
- Moore's law is still alive
  - ➡ number of transistors still doubles every 2 years
    - no free lunch, clock speed no longer increasing
  - ➡ lots of transistors looking for something to do:
    - vector registers
    - out of order execution
    - hyper threading
    - multiple cores
  - ➡ many-core processors, including GPGPUs
    - lots of cores with less memory
  - ➡ increase theoretical performance of processors

- challenge will be to adapt HEP software
  - ➡ hard to exploit theoretical processor performance
    - many of our algorithm strategies are sequential
  - ➡ need to parallelise applications (multi-threading)
    (GAUDI-HIVE and CMSSW multi-threading a step in this direction)
    - change memory model for objects, more vectorisation, ...



see G.Stewart, CHEP 2015

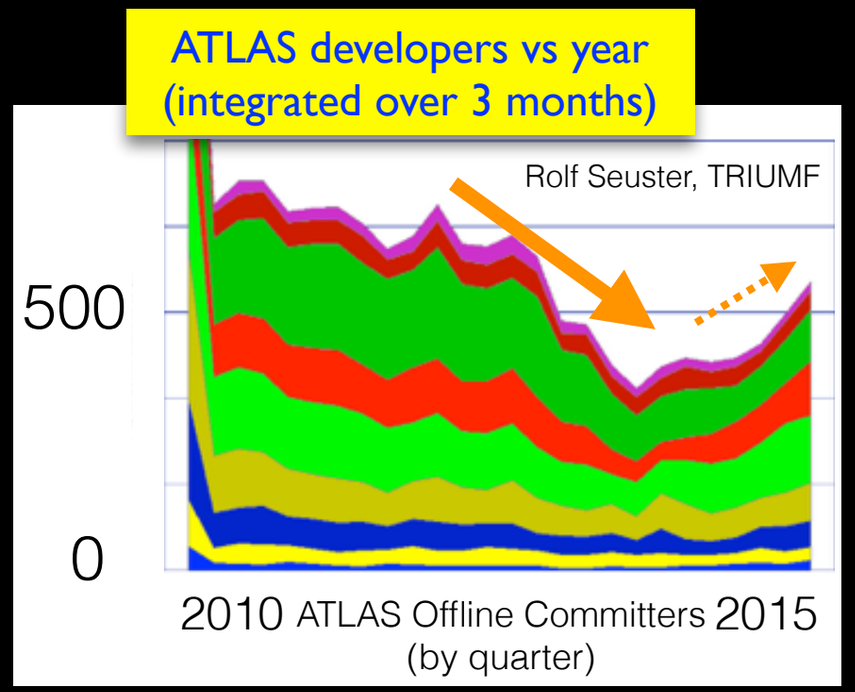


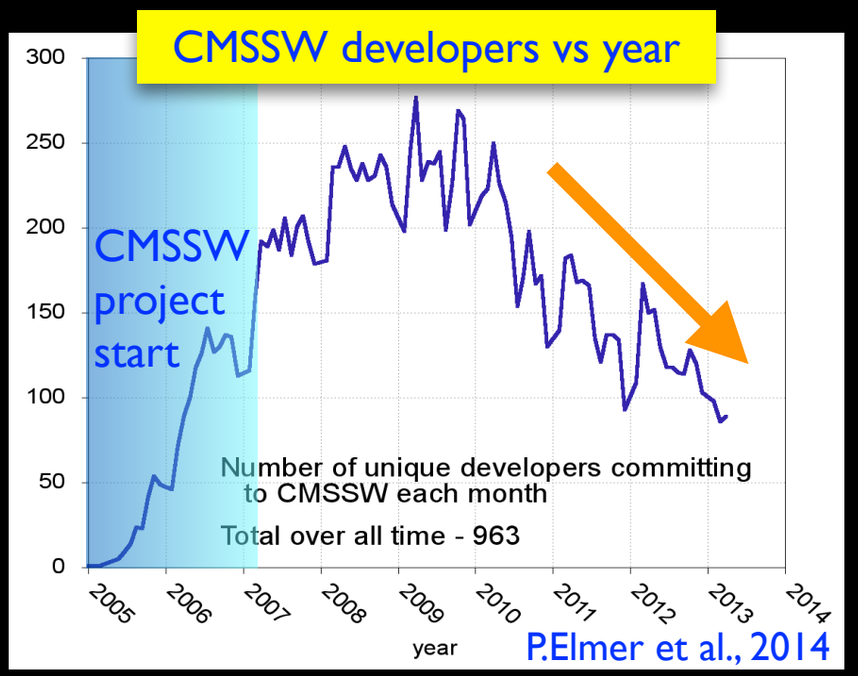


Software and Manpower

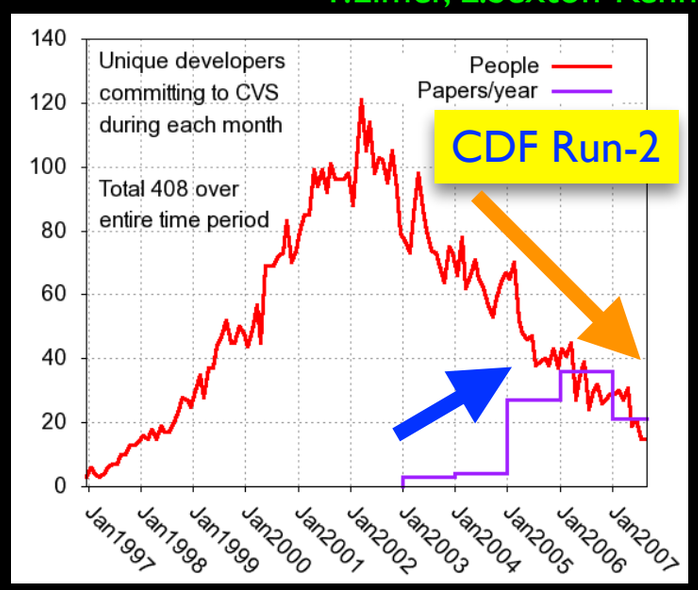
- software follows a natural life cycle
  - ➡ building up the software for an experiment
  - ➡ start of experiment operations and data taking
  - ➡ data analysis and detector upgrades

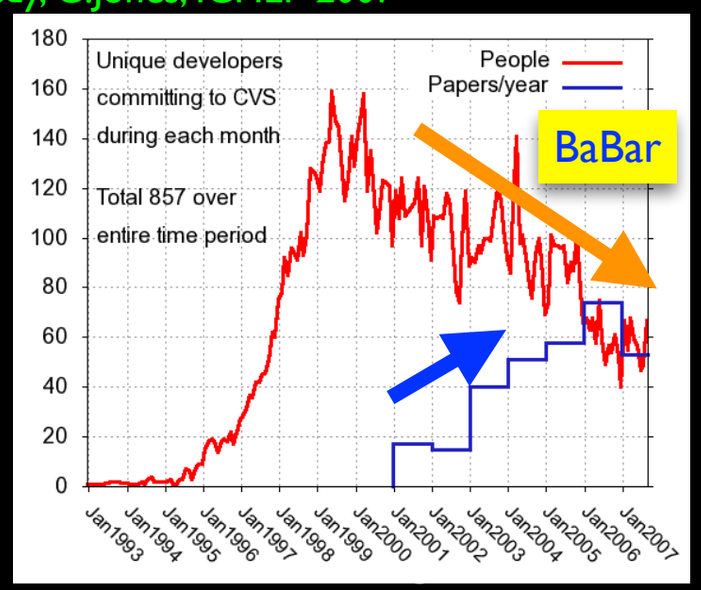
- loss of software manpower in ATLAS/CMS
  - ➡ (mostly) students and postdocs moved on to do physics
    - same trend like in previous experiments
  - ➡ like CDF/D0 Run-2, LHC upgrade program is ambitious
    - need to find sufficient manpower to develop the software for the upgrade (some positive trend in ATLAS)





P.Elmer, L.Sexton-Kennedy, C.Jones, ICHEP 2007

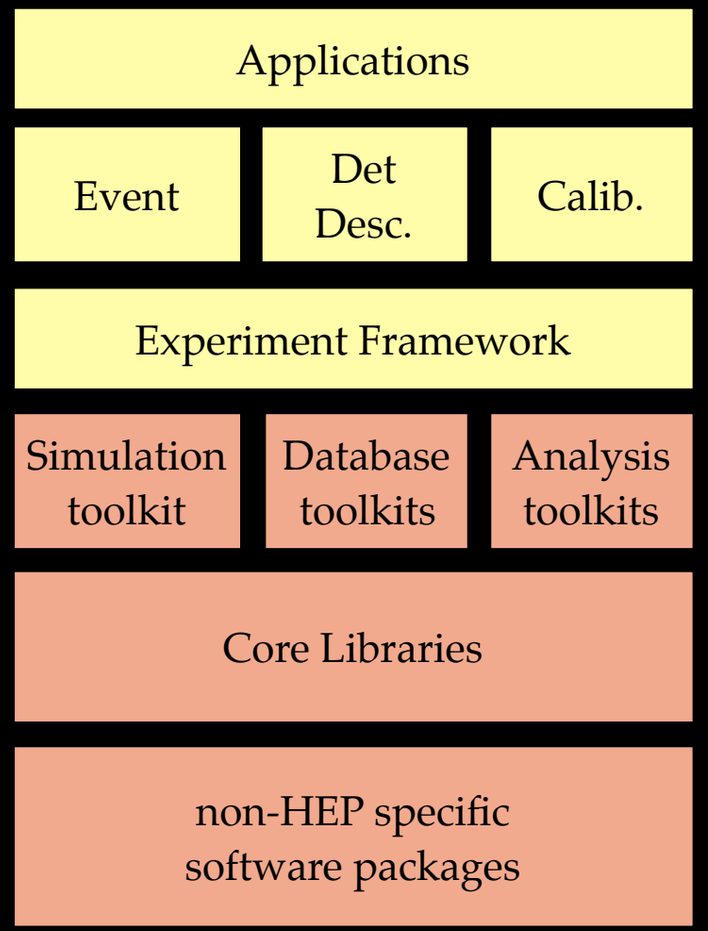




Present Software of Experiments

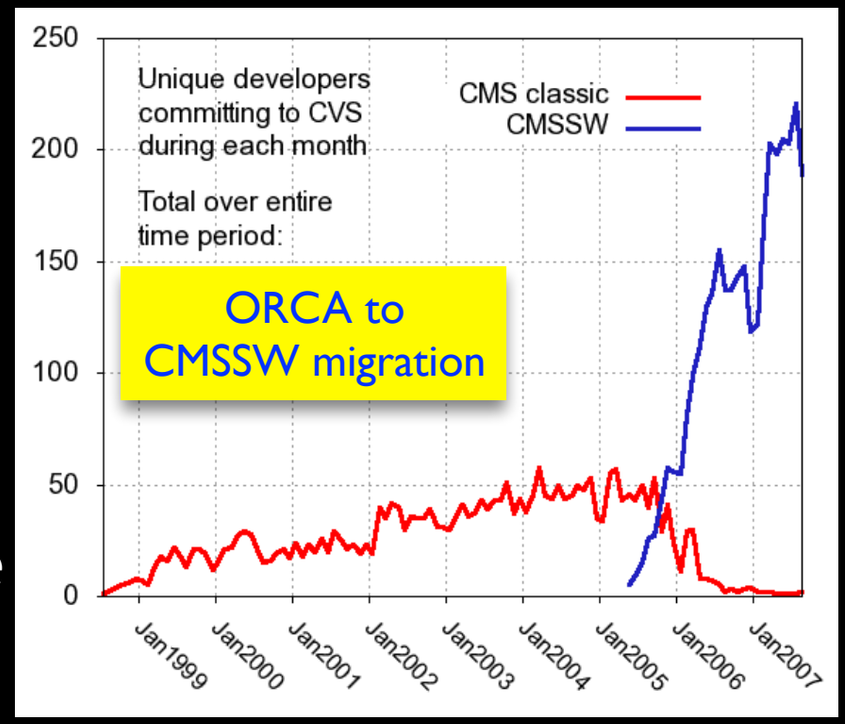
- software stacks of the experiments
  - ➡ applications (algorithms) implemented in framework
    - detector simulation, trigger, reconstruction, ...
  - ➡ based on common software toolkits
    - things like Geant4, Root, Pool, Cool, Coral, ...
  - ➡ today a full CMSSW release has 7.5 *million* lines of code
    - OpenHUB "estimated cost" is **125 M$**

      https://www.openhub.net/p/cms-sw-cmssw/estimated_cost
    - similar for all experiments, framework is only a fraction of this



- scale of software stacks and ability to do large scale migrations ?
  - ➡ CMS started 2005 CMSSW to replace ORCA
    - huge effort, took >3 years
  - ➡ existing experiments able to do such "disruptive" changes in the future ? - given the (manpower) investment
    - still we have to address the future challenges
    - need to find a way to do this adiabatically where possible

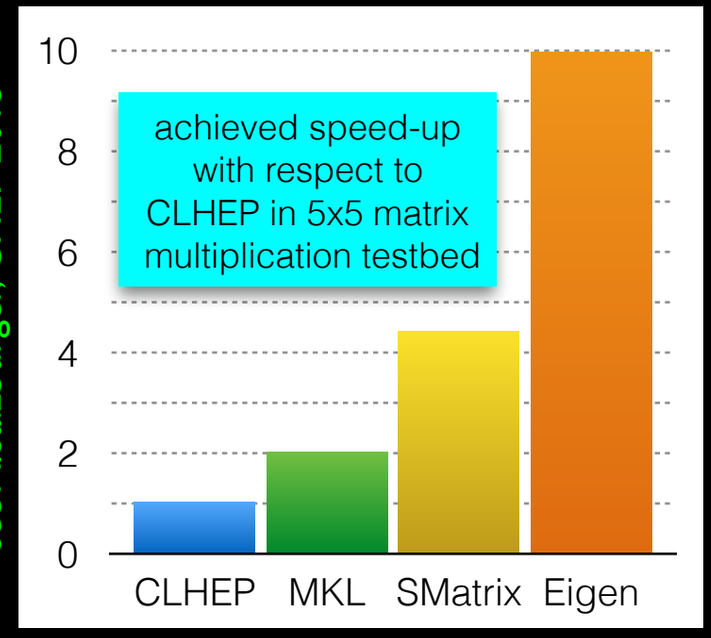
P.Elmer et al.



Lessons from Tracking Work for Run-2 ?

- ATLAS and CMS focus on technology and strategy to improve CURRENT algorithms
  - ➡ improve software technology, including:
    - simplify EDM design to be less OO ("hip" 10 years ago)
    - ATLAS migrated to Eigen - faster vector+matrix algebra
      (CMS was already using SMatrix)
    - vectorised trigonometric functions
      (CMS: VDT or ATLAS: intel math lib)
    - work on CPU hot spots
      (e.g. ATLAS replaced F90 by C++ for B-field service)
  - ➡ tune reconstruction strategy (very similar in ATLAS and CMS):
    - optimise iterative track finding strategy for 40 pileup
    - ATLAS modified track seeding to explore 4th Pixel layer
    - CMS added cluster-shape filter against out-of-time pileup

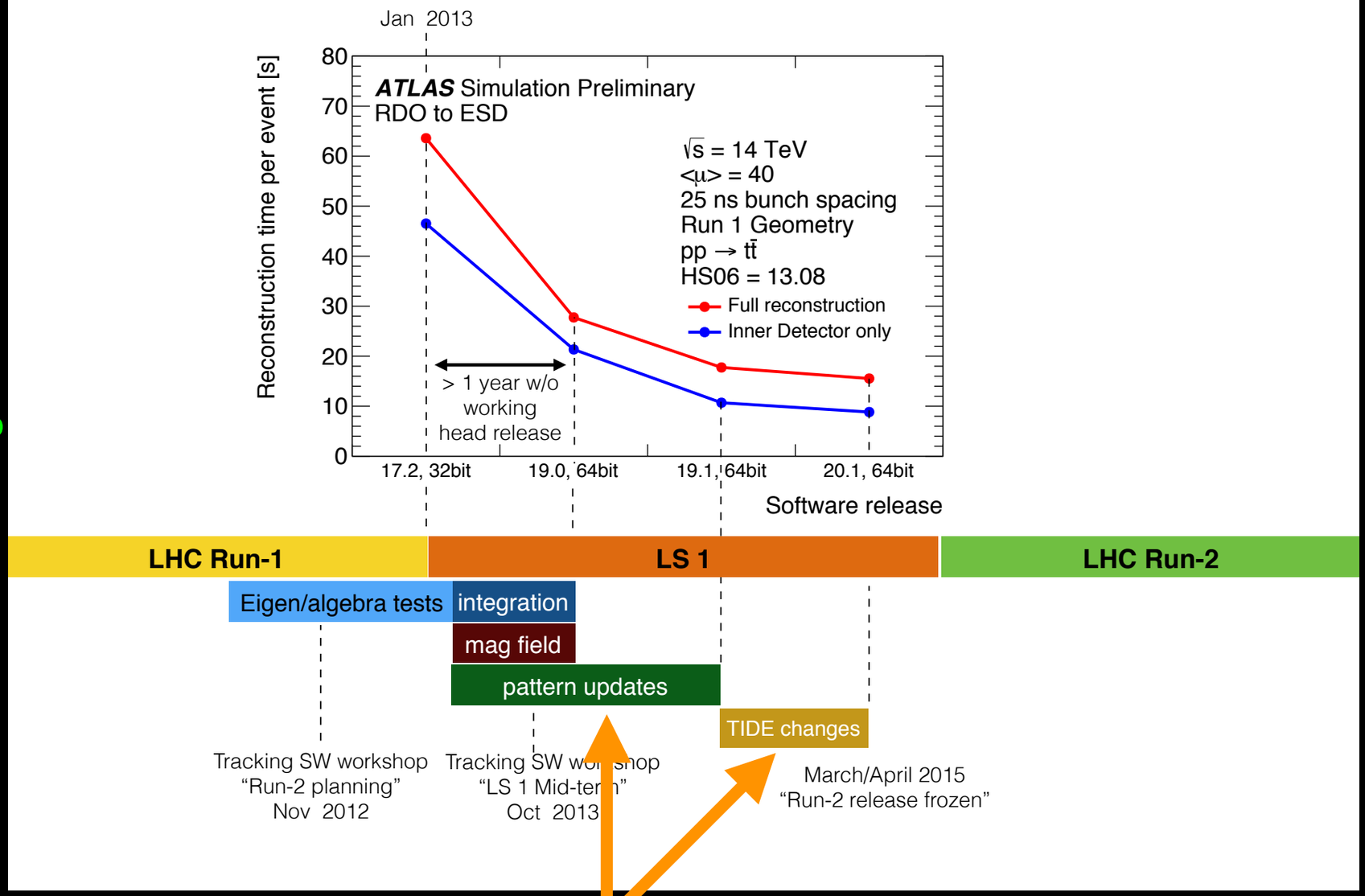
- hence, mix of SIMD and algorithm tuning
  - ➡ CMS made their tracking as well thread-safe



Huge Improvements e.g. in ATLAS

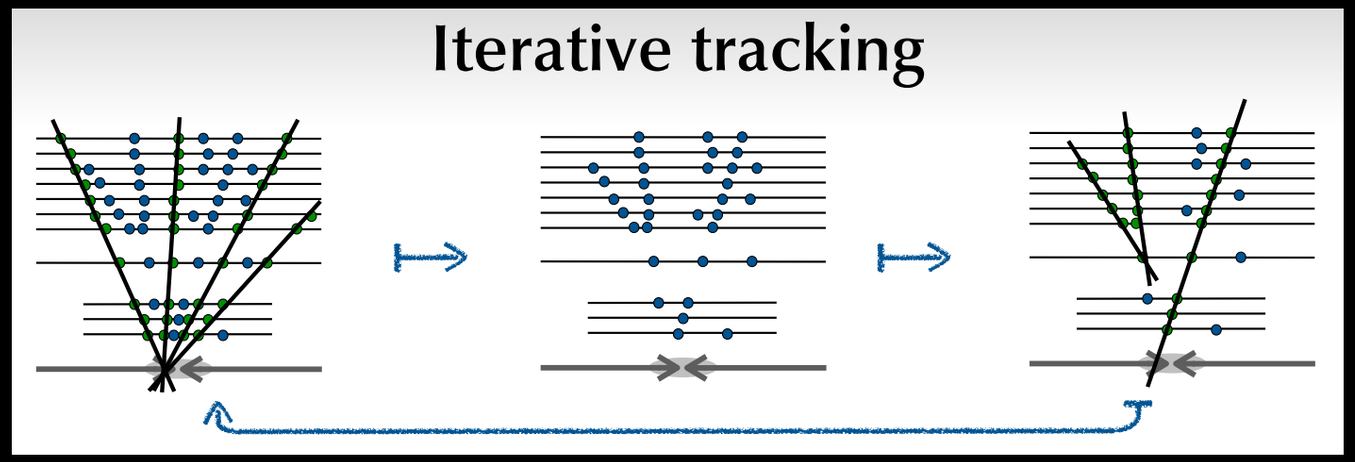
A.Salzburger, CHEP 2015

The factor 4 - a planning & deployment exercise



- biggest gain is in algorithms
  - ➡ similar gains achieved by CMS

Massively parallel Tracking ?



- ATLAS/CMS tracking strategy is for early rejection
  - ➡ iterative tracking: avoid combinatorial overhead as much as possible !
    - early rejection requires strategic candidate processing and hit removal
  - ➡ not a heavily parallel approach, it is a SEQUENTIAL approach !

- implications for making it massively parallel ?
  - ➡ Armdahl's law at work:

$$\text{Time}_{||} = \text{Para} / N + \text{Seq}$$

  - ➡ iterative tracking: small parallel part Para, heavy on sequential Seq
    - hence, if we want to gain by a large N threads, we need to reduce Seq

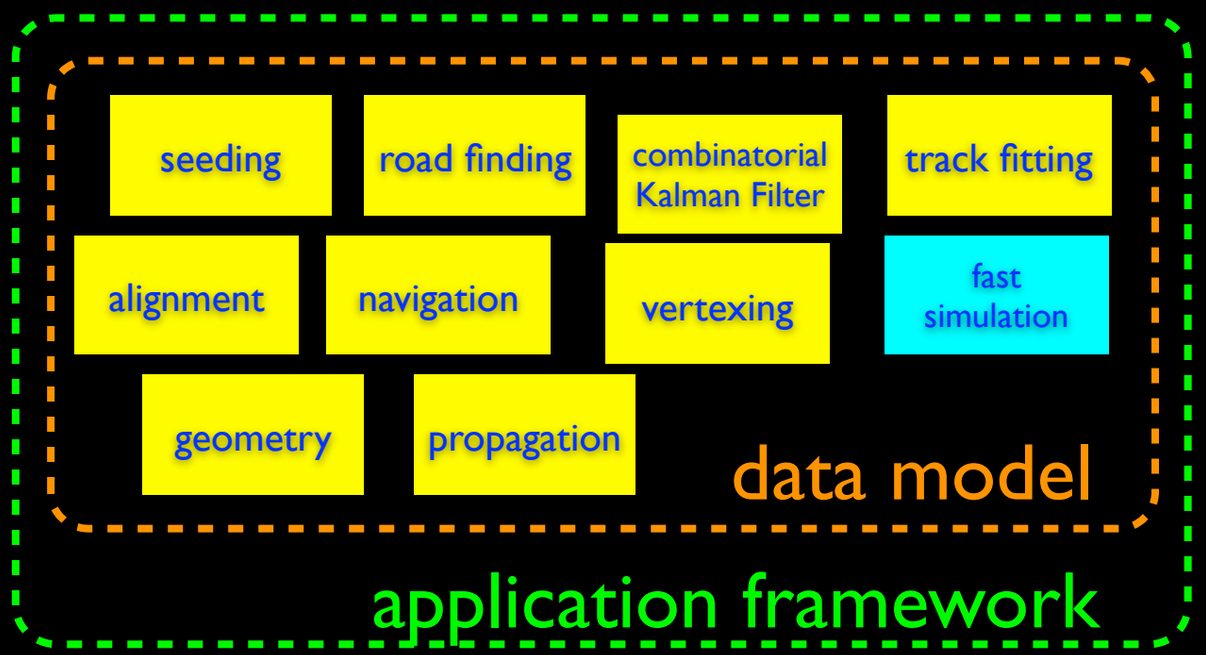
- hence we need to re-think the algorithmic strategy
  - ➡ having concurrency in mind from the very start

Common Algorithmic Software ?

from Graeme's talk

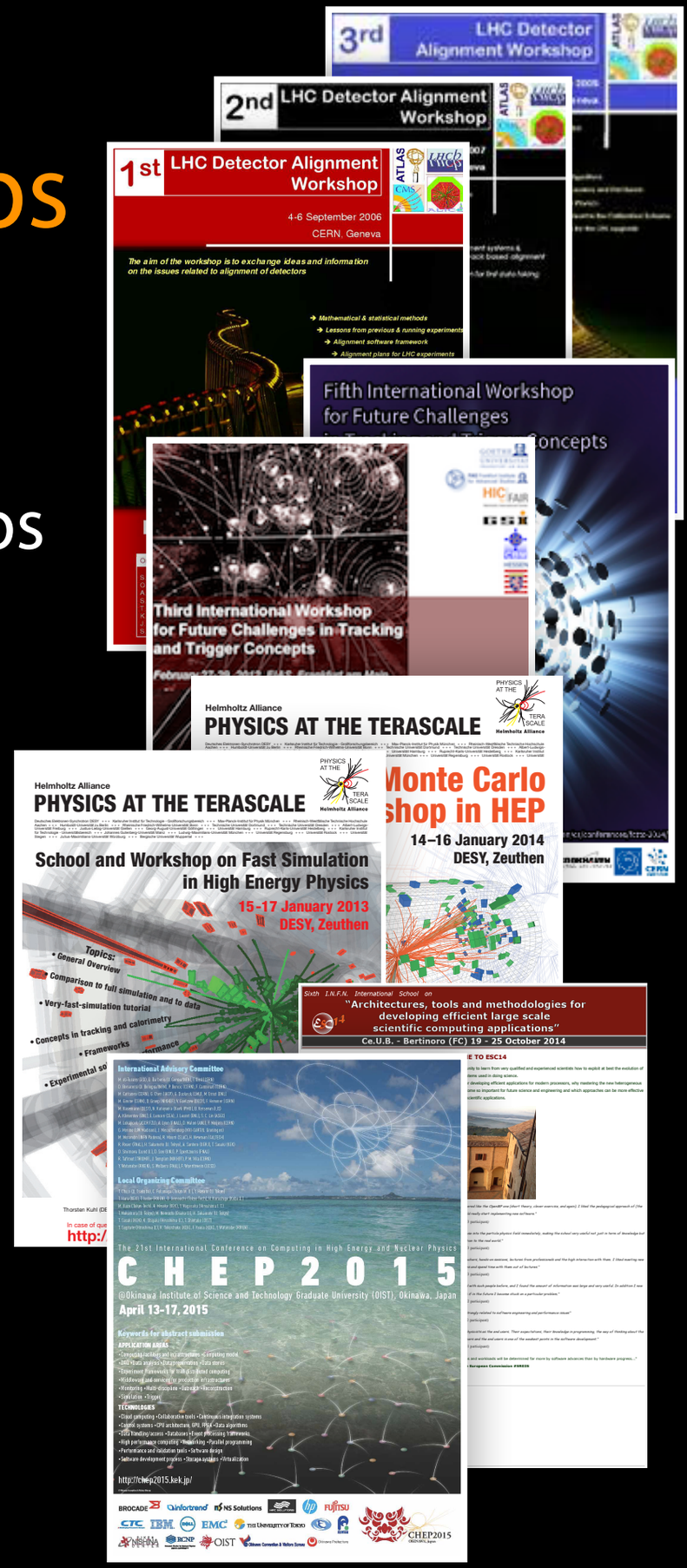
- examples for common algorithmic software
  - ➡ FastJet - de-facto standard for jet finding, distribution as part of LCG externals
  - ➡ TMVA, RooFit/RooStat, HistFitter, BAT - statistics and multivariate analysis
  - ➡ AIDA tracking - primarily targeting ILC / FCC
  - ➡ genfit - an implementation of standard track fitting techniques (Belle-II)
  - ➡ CMS vertexing suite - package of standard vertexing codes (CMS, Belle-II,...)
  - ➡ VDT, SMatrix, Eigen - vector algebra and math libs

- a real integrated common tracking implementation ?
  - ➡ AIDA is the one aiming at this ...
  - ➡ integration means picking a data model
    - determines Jacobians in math formulars
  - ➡ integration means framework interfaces
  - ➡ best physics performance ?
    - pattern strategy depends on experiment
  - ➡ manpower on AIDA vs (e.g.) CMS/ATLAS ?
  - ➡ discussion in ATLAS:
    - make tracking/vertexing suite public ? (for FCC)



Examples for Tracking Workshops

- in the past e.g. LHC alignment workshops
  - ➡ alignment algorithms and strategies in 2006-2008
- GSI/FAIR future tracking and trigger workshops
  - ➡ served as well as broader forum for algorithm discussion
    - recently focus shifting towards FAIRRoot
- related fast simulation workshops
  - ➡ fast simulation engines reuse tracking codes...

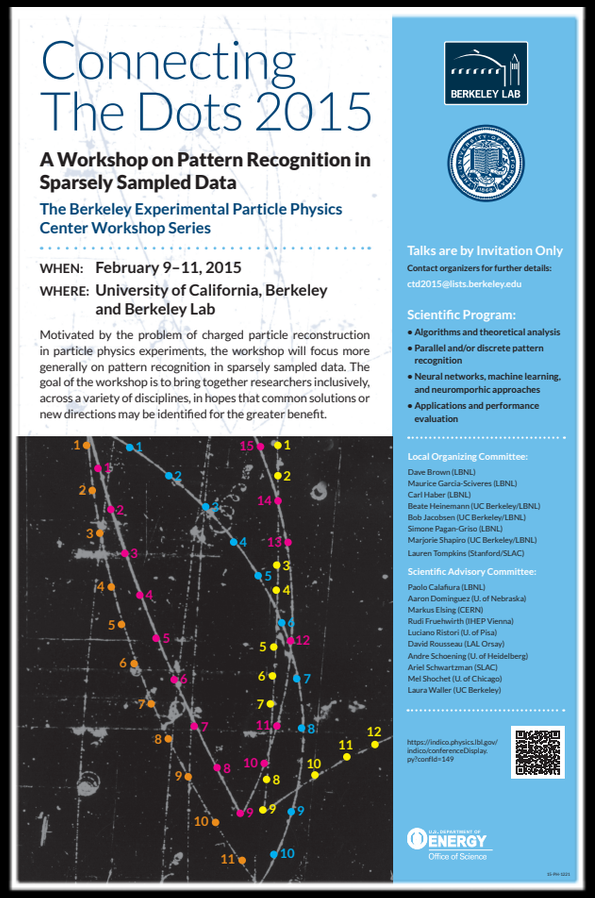


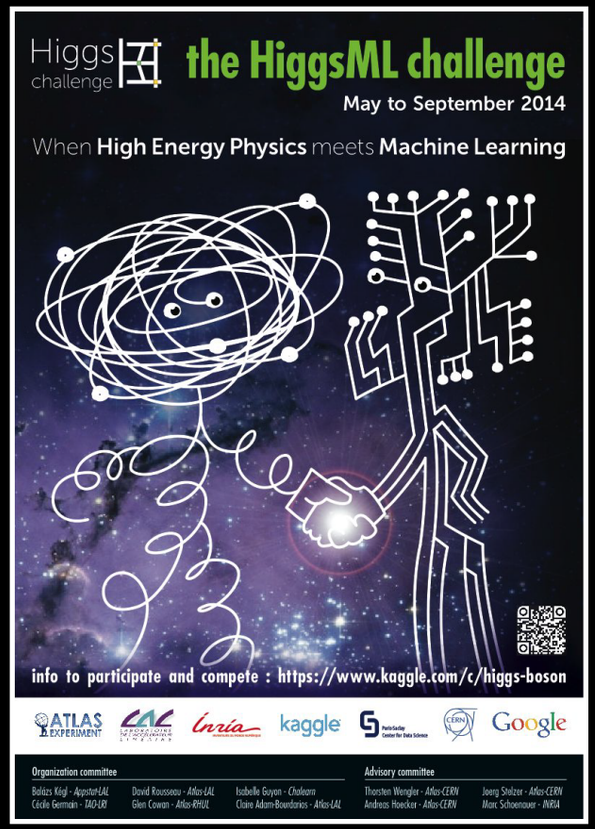
- examples for more general conferences:
  - ➡ ACAT and of course CHEP
  - ➡ Vertex - silicon hardware oriented with some software talks
- examples for more general schools:
  - ➡ CERN schools of computing
  - ➡ others like the ESC INFN computing schools

... this list is of course far from complete...

Connecting the Dots Workshop

- dedicated to pattern recognition techniques
  - ➡ organised at Berkeley in February 2015
- well received by community
  - ➡ 55 participants across all LHC/Belle-2/Future Collider experiments, as well contributions from theory and even non-HEP
- 4 main workshop subjects
  - ➡ mathematical algorithms and theoretical analysis
  - ➡ parallel and/or discrete pattern recognition techniques
  - ➡ neural networks, machine learning, neuromorphic approaches
  - ➡ applications and performance evaluation of existing applications
- follow up initiatives
  - ➡ created mailing list: Detector-Technology-Pattern-Recognition@cern.ch
    - we recently had our first post-workshop phone meeting
    - work started on deep learning tracking project for Kaggle
  - ➡ Vienna offered to host (one of) the next workshops in 2016





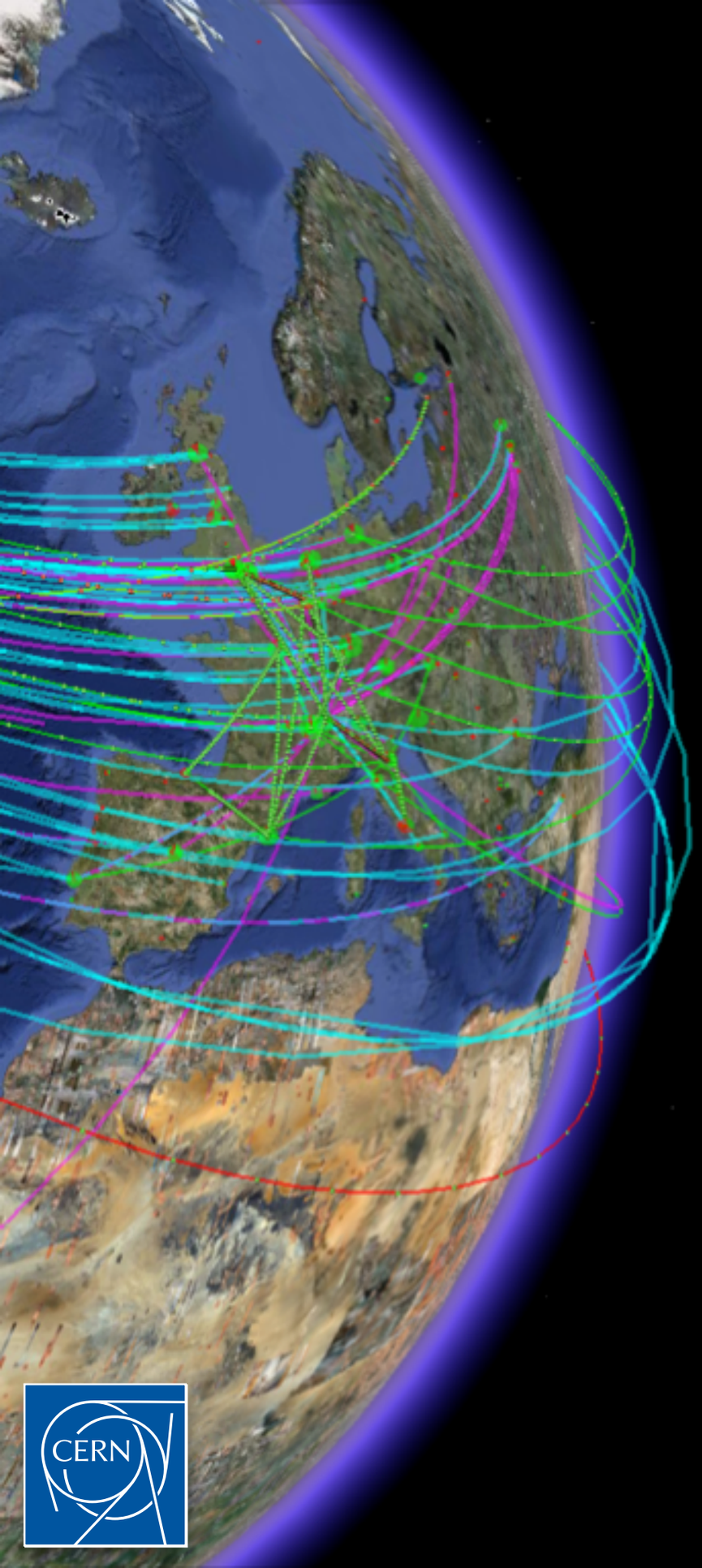
Building a "Forum" and a Community ?

- some obvious observations:
  - ➡ we need to make workshops like Connecting the Dots more regular
    - yearly like BOOST workshops ? every 18 months like CHEP and ACAT ?
  - ➡ we need to think about dedicated schools to teach algorithms to students
    - we need to invest in future experts (and give them career perspectives)
  - ➡ do we need some more regular forum alongside the Concurrency Forum ?
    - need will grow once we have more common developments to discuss
    - how often shall we do such a meeting initially ?

- focus on exchange of ideas, techniques, best practices ... ?
  - ➡ at Connecting the Dots meeting, not much enthusiasm across all experiments (but maybe FCC) to migrate to something like a common algorithm stack
  - ➡ common software projects may grow naturally out of needs we may identify

- created as well a generic HSF mailing list:

  http://hepsoftwarefoundation.org/content/reconstruction-algorithms-forum
  - ➡ to be used to bring together initiatives like Connecting the Dots for tracking and the communities working on boosted object reconstruction and alike



Discussion…