Data Reconstruction in Modern Particle Physics

Daniel Saunders, University of Bristol

About me…

- Particle Physics student, final year.
  - CSC 2014, tCSC 2015, iCSC 2016
- Main research interests.
  - Detector upgrades for LHC - pixel detectors for LHCb.
  - Neutrino experiments at nuclear reactors - data reconstruction and analysis.
- But day to day…
  - Professional ROOT and git complainer.
  - Developing C++ and python projects to perform above.

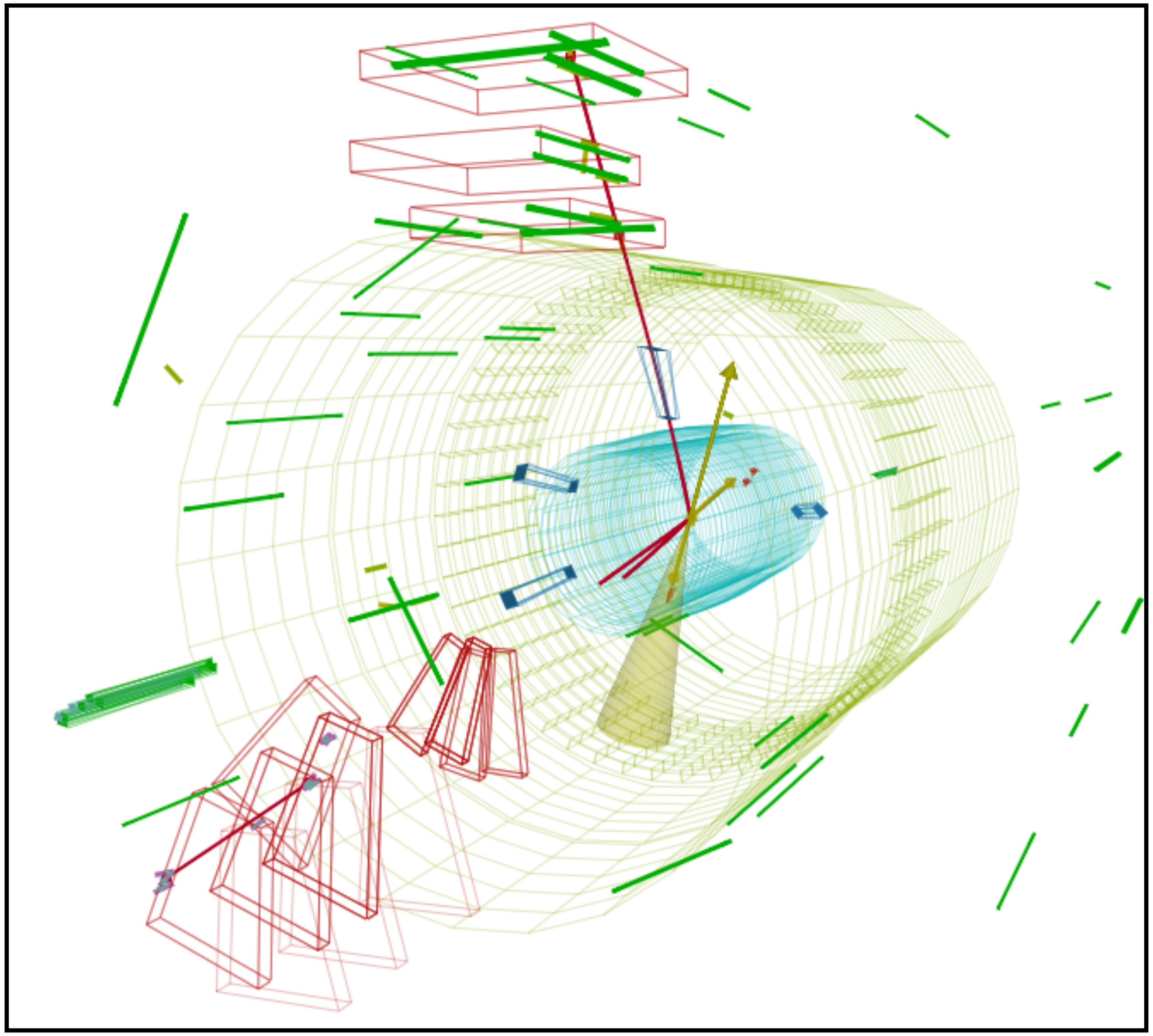
But it’s not all about me…

- I assume you know a bit about:
  - The LHC.
  - LHC-like particle physics detectors, and that they give information like:
    - Particle energies.
    - Particle paths.

The Tardis prompts questions for the audience.

Aim of these lectures

What just happened?

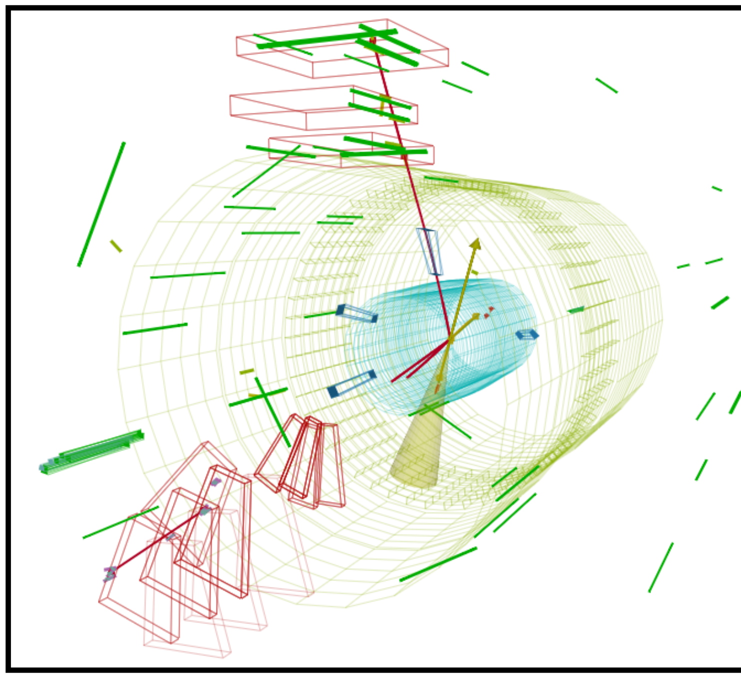


Example collision event from CMS.

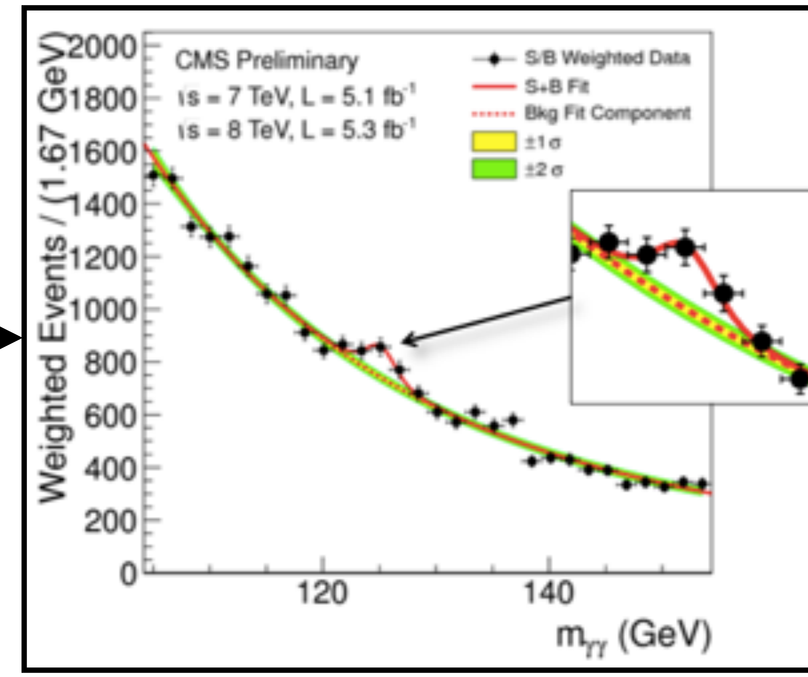
Aim of these lectures

What just happened?

- LHC detectors produces O(10) petabytes of data per year[1].
- Data is processed to the stage of physics papers → measurements and discoveries.



Example collision event from CMS.



Higgs discovery at CMS.

- Many steps involved.
  - Each step has **computing** costs, varying **inefficiencies**, often in large **backgrounds**.
- We'll consider some steps in detail, looking at tradeoffs between these three factors.

Lecture Outline - Scope

Aim of these lectures: 'How to take detector output and make physics measurements.'

Lecture 1

- Introduction and context:
  - Elements of LHC detectors (CMS & LHCb).
  - Data rates and formats.
- Data taking strategies:
  - Triggers.

Lecture 2

- Event reconstruction principles:
  - Tracking.
  - Particle identification.
  - Vertexing.
- Optimisations:
  - Parallelism.
  - The Grid.

Lecture Outline - Beyond Scope

- Particle Physics detectors exist:
  - Able to measure the position of a crossing particle and energy.
  - Measurements are to a limited resolution.
  - Detectors aren't perfect: can be inefficient, impure and have bad resolution.

- The exact workings of detectors **is not** considered.
  - Lecture concepts are general and apply to all particle physics experiments.
  - For detector concepts, see:

- Many physics measurements are made by comparing reconstructed data to simulation.
  - A large topic! Many sophisticated data analysis techniques used.
    - We will stop at this point.

Many of these topics are discussed in other lectures of this school.

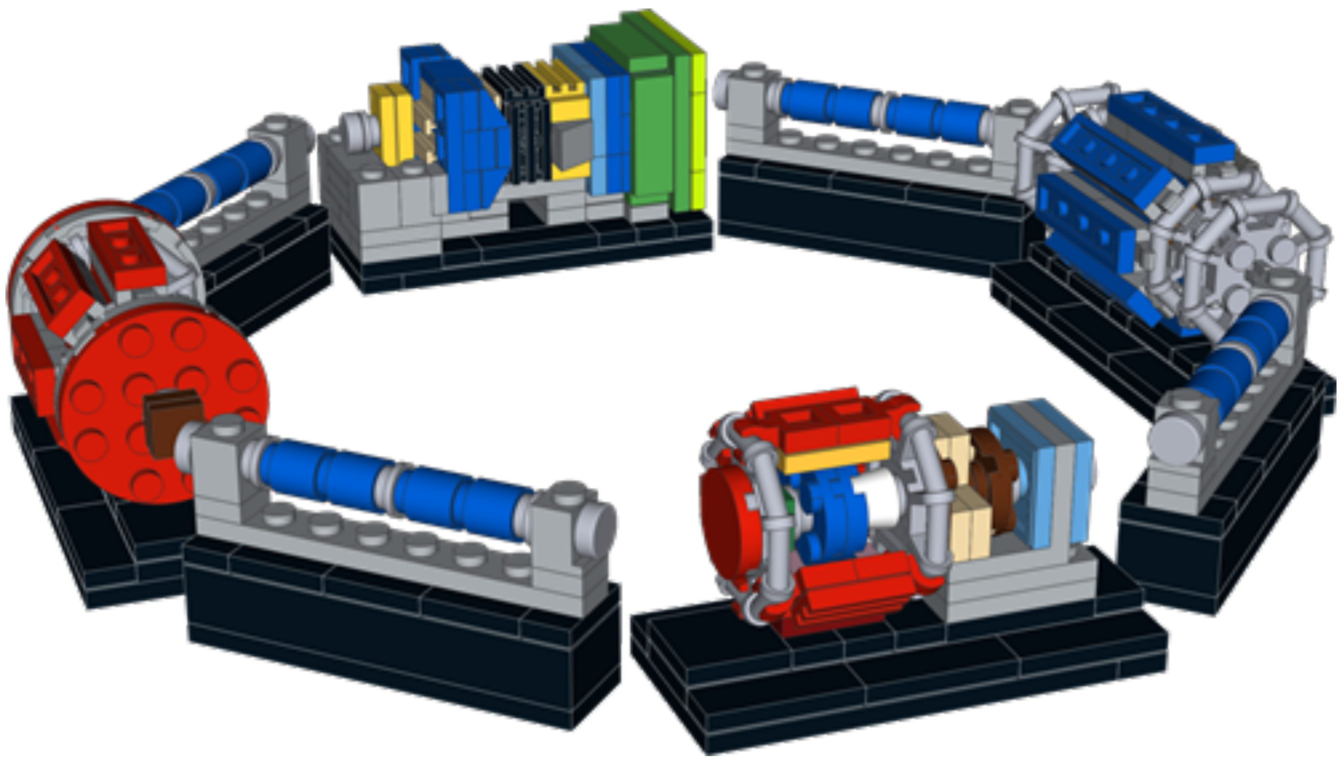
Lecture 1

Introduction and Data Taking

Introduction and Motivation

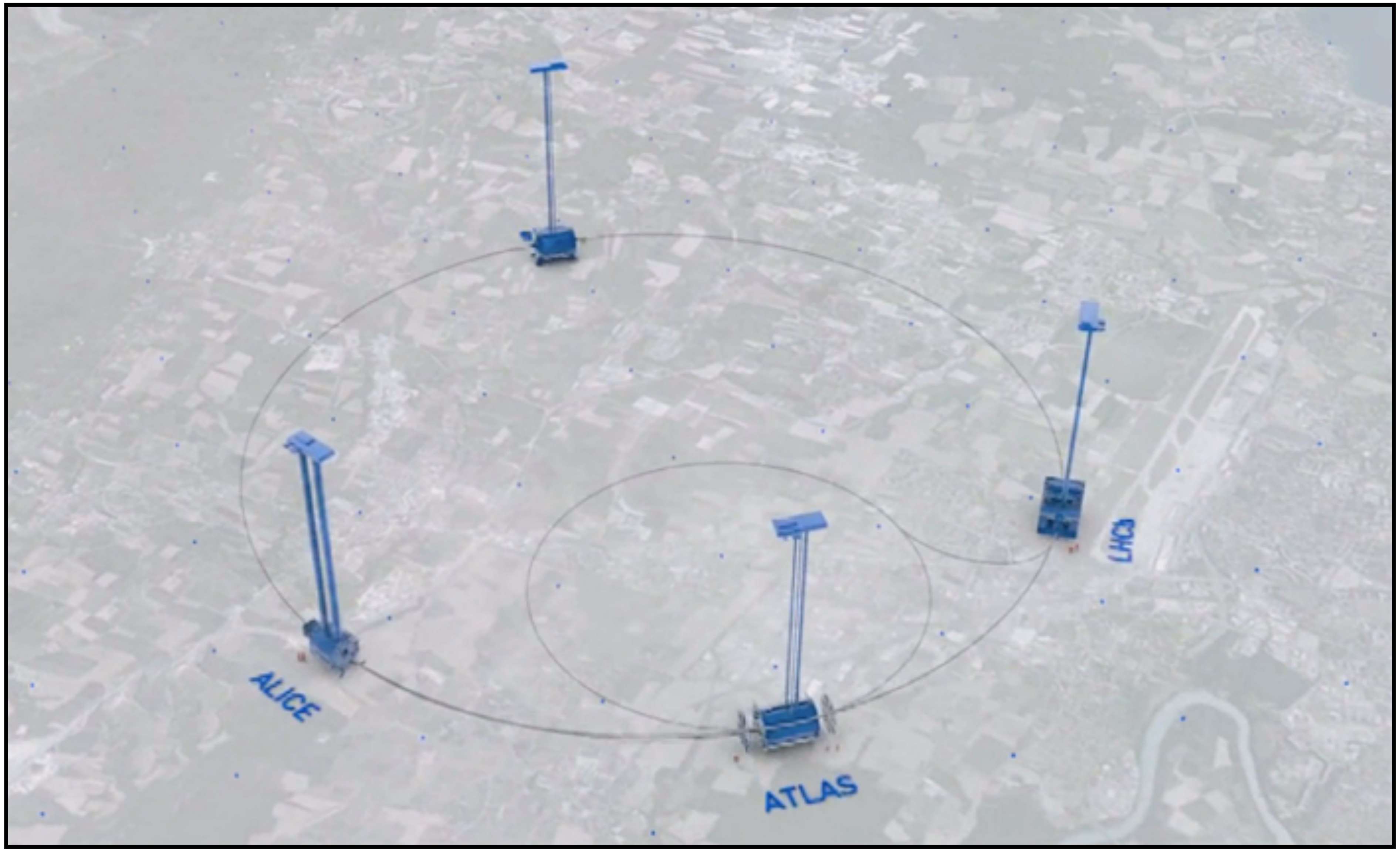
- LHC computing is an enormous task:
  - Data rate of typical LHC experiments in run two (i.e. current) ~1GB/s[2].
  - In extreme cases (CMS[3] & ATLAS), this is post 1 in ~10,000 filter by electronics.
  - Still large background of uninteresting events.

- Collaborations of thousands of scientists:
  - Many kinds of analysis requiring different data selection.
  - Competitive spirit (e.g. ATLAS vs CMS).
  - Many kinds of people (Computer Scientists, Engineers, Physicists etc.) with varying computing skills.



LHC Reminder

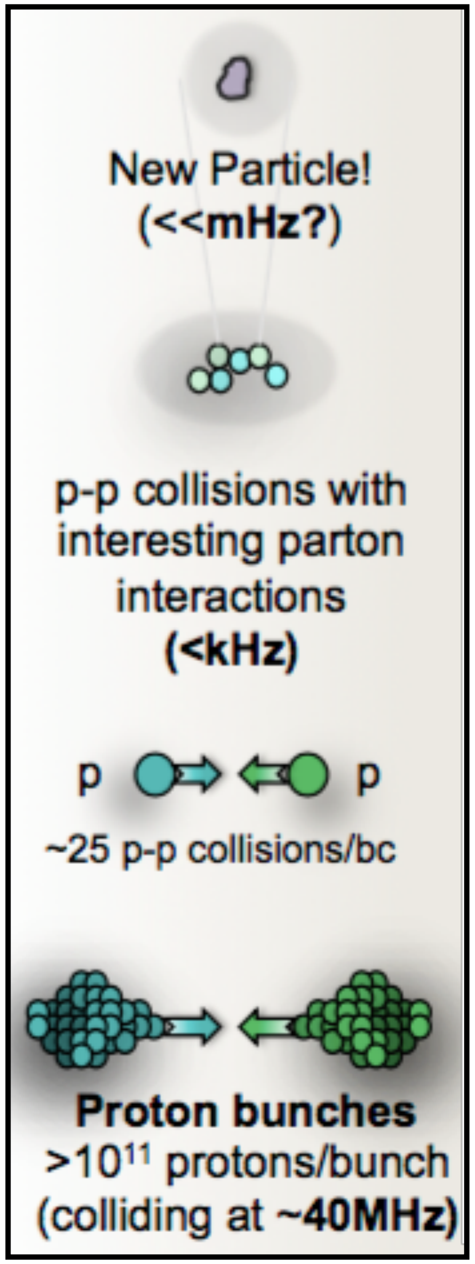
- LHC is a proton - proton (or Pb - Pb) collider.
- **Each collision is called an event.**



- LHC examples used in these lectures, but many concepts apply to other big data experiments in particle physics and beyond.

LHC Collisions

- Particles are grouped into bunches of ~ 100 billion protons.
- LHC collides two proton beams, circling in opposite directions.
- A collision between two bunches happens once every 25ns (so 40MHz).
- In each collision, ~25 pairs of proton collisions. About 1 per million is used for physics.
  - The rest is well understood background of un-interesting events.



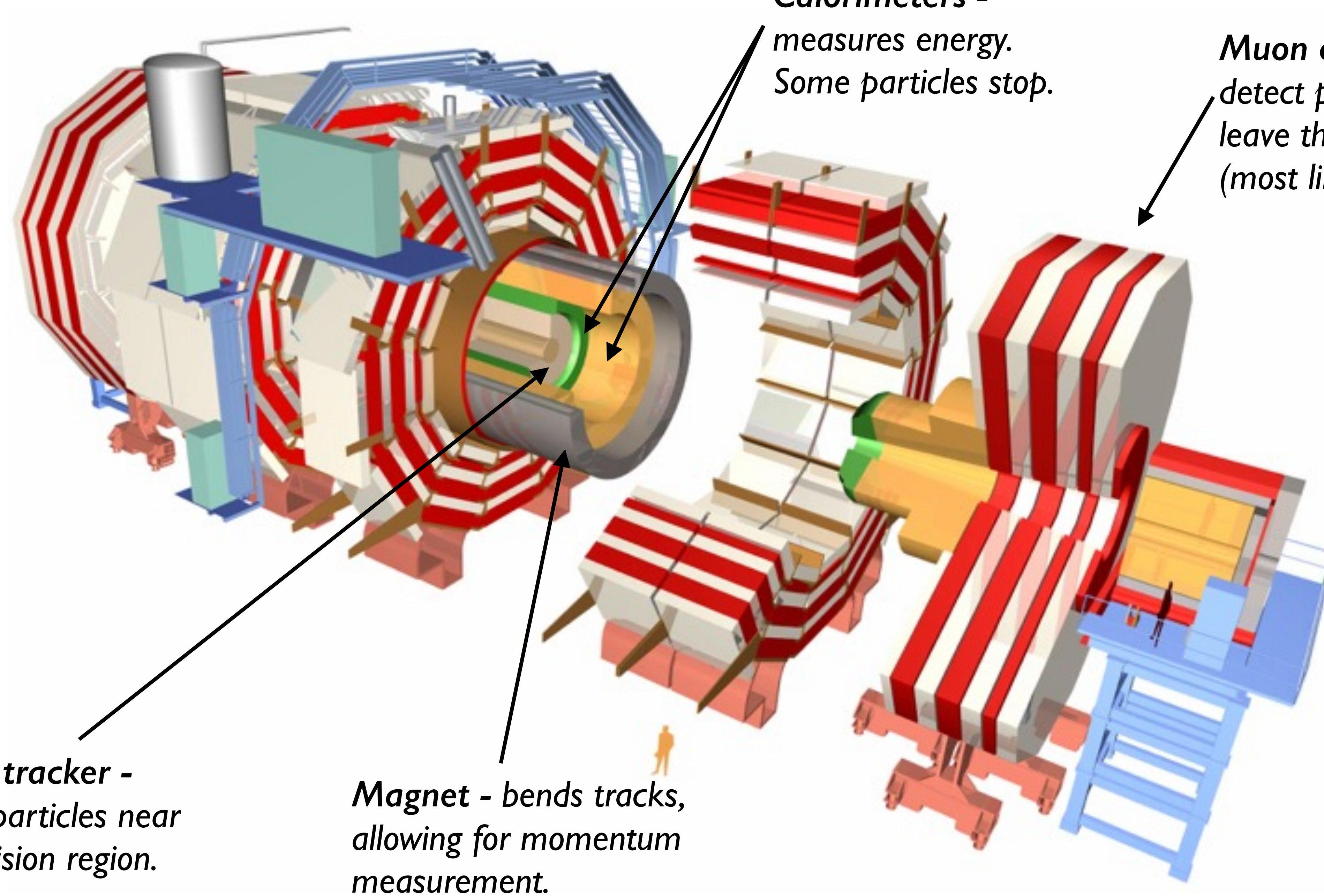
LHC Detectors

- Large dedicated detectors built surrounding four collision points:
  - ATLAS & CMS: general purpose detectors, discovery machines.
  - ALICE: high energy plasma studies, conditions approx 10^{-6} seconds post big bang.
  - LHCb: precision antimatter studies.
  - Many other smaller experiments.

- Detectors generally either perform:
  - Direct searches (e.g. CMS) - looking for new phenomena, such as the Higgs Boson.
  - Indirect searches (e.g. LHCb) - compare precision measurements to theoretical predictions. Differences can be a sign of new physics that has been unaccounted.

- Choose CMS and LHCb to look at in more detail.

LHC Detectors - CMS



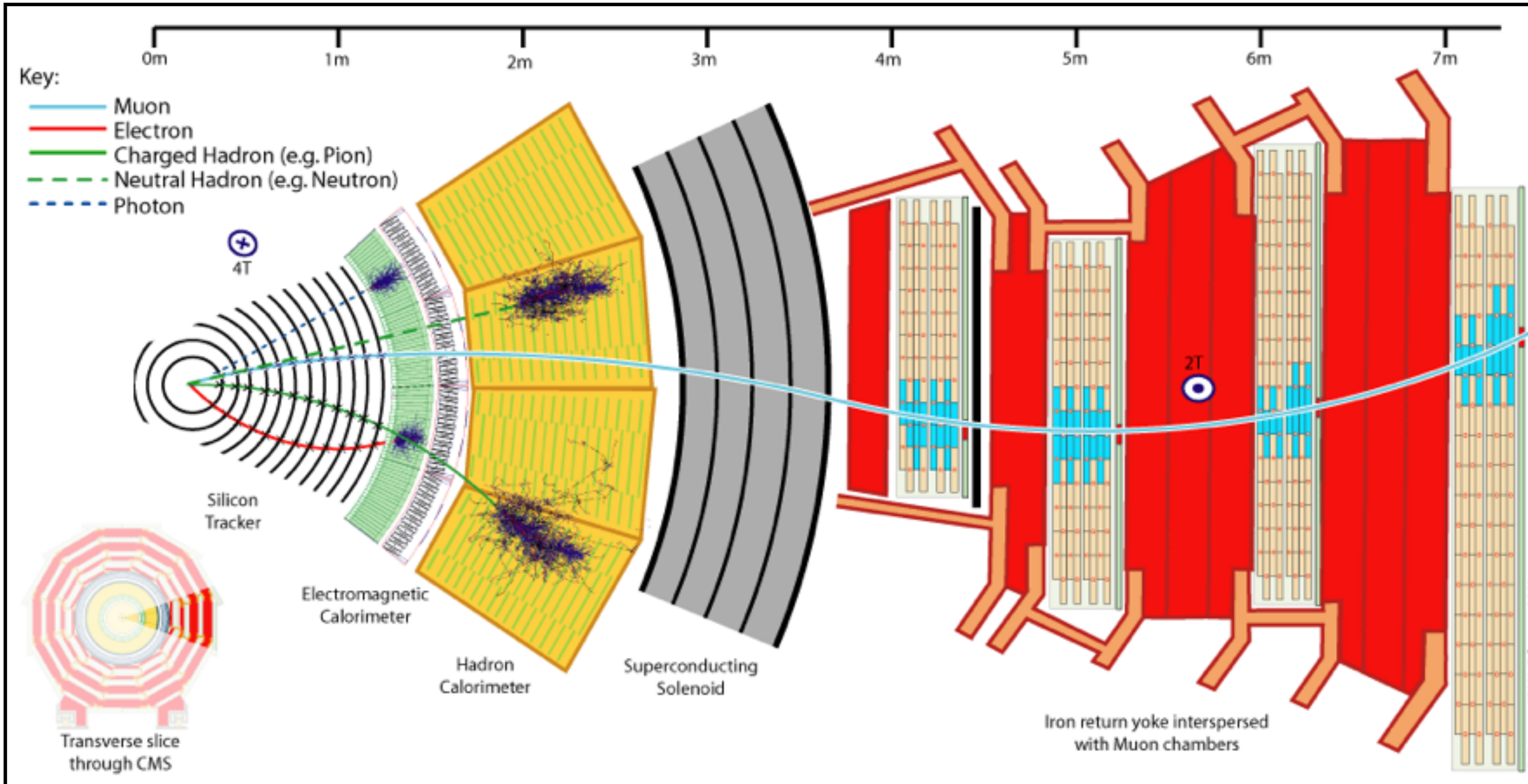
Calorimeters - measures energy. Some particles stop.

Muon chambers - detect particles that leave the detector (most likely muons).

Silicon tracker - tracks particles near the collision region.

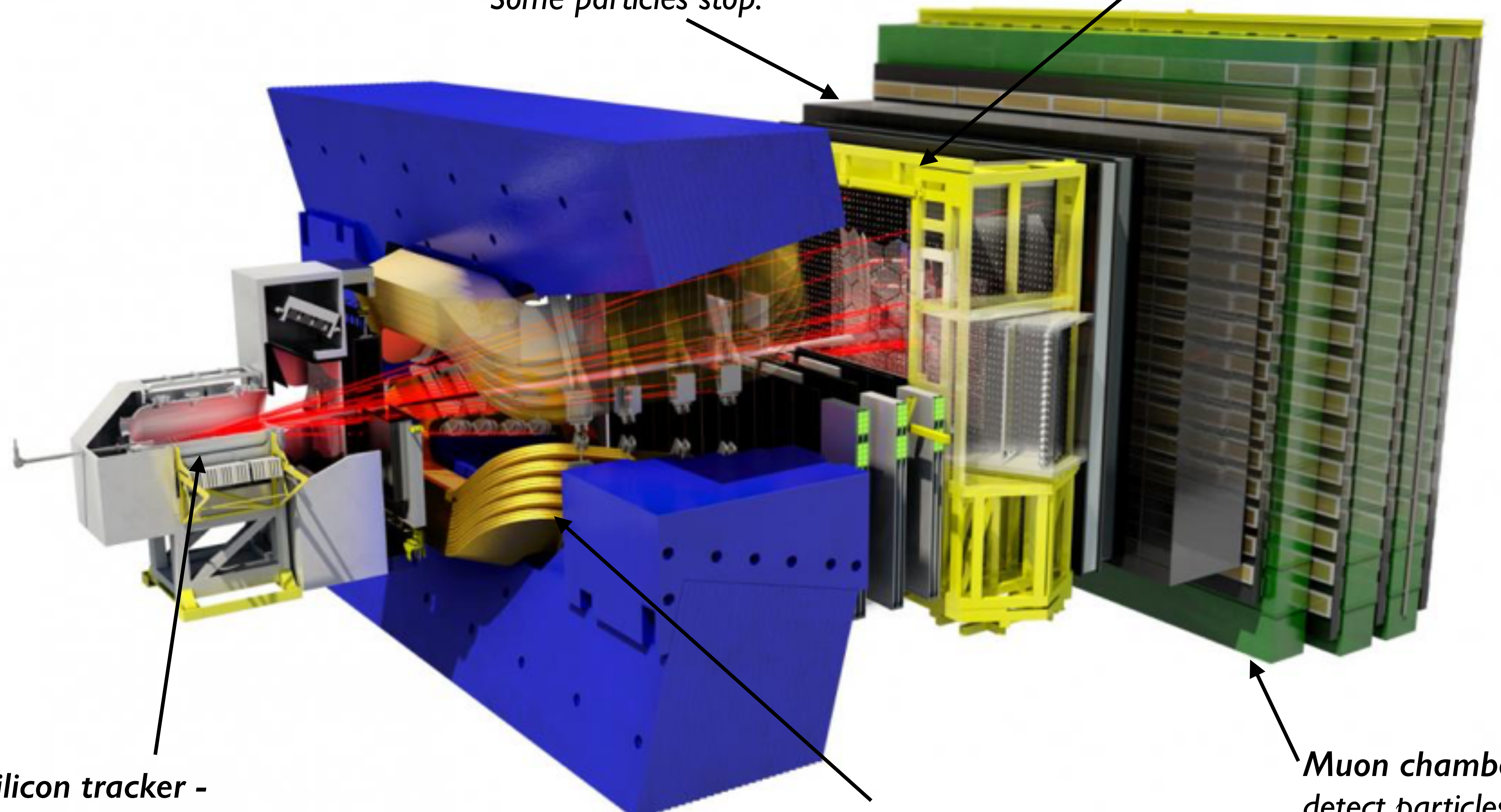
Magnet - bends tracks, allowing for momentum measurement.

LHC Detectors - CMS



CMS slice, with different particle examples.

LHC Detectors - LHCb



Calorimeters - *measures energy. Some particles stop.*

RICH *(**R**ing **I**maging **C**herenkov Detector) - particle ID from radiation induced by crossing particles.*

Silicon tracker - *tracks particles near the collision region.*

Magnet - *bends charged tracks, allowing for momentum measurement.*

Muon chambers - *detect particles that leave the detector (most likely muons).*

The Task

What just happened?

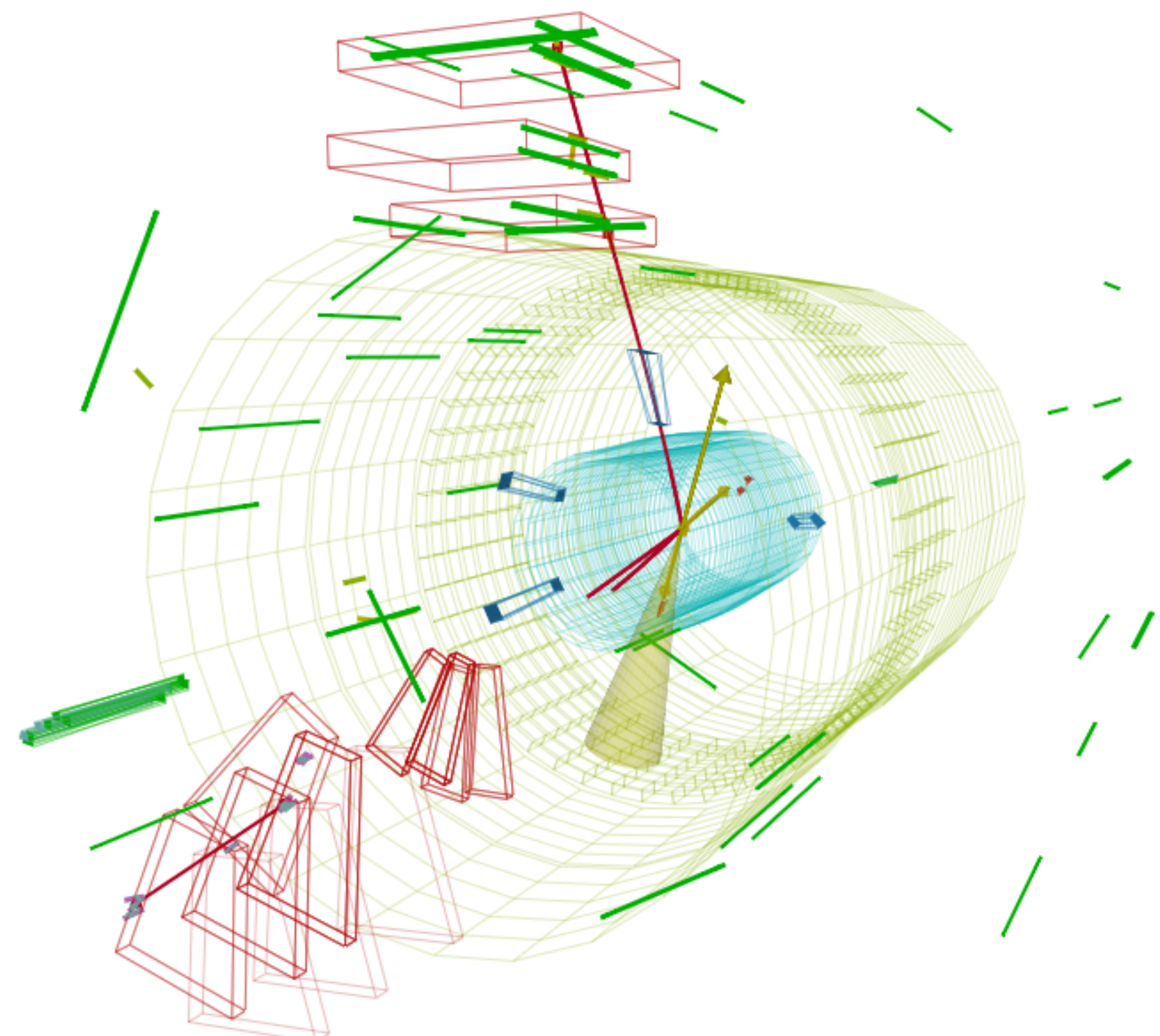
- All LHC computing is connected to data processing.
- Need to take detector output (~1GB/s of electrical signals) and perform:

Needed live.

- Detector related studies:
  - Online data quality monitoring.
  - Calibrations.

Processed similar rate to data taking.

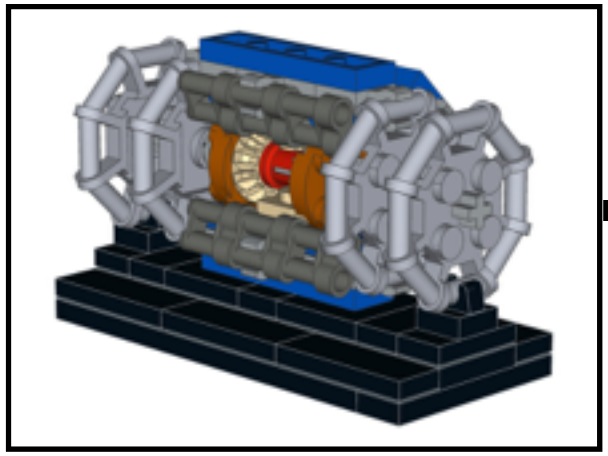
- Physics analysis:
  - Discovery of new particles.
  - Comparisons with simulation.



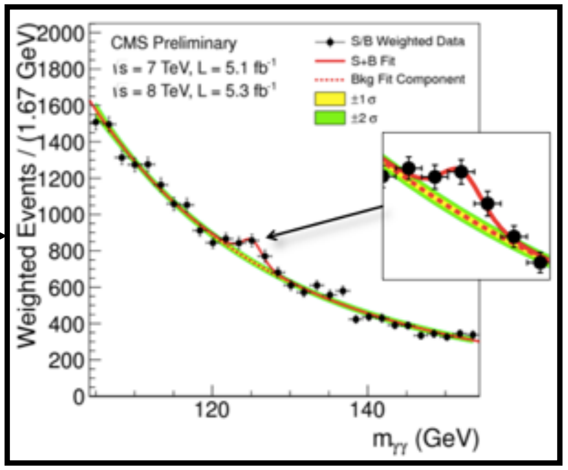
Example collision event from CMS.

Data Flow

- Data reconstruction **generally** involves several steps of processing and reduction:



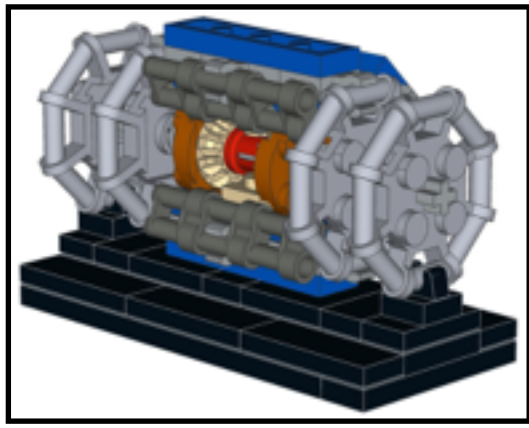
Detector output

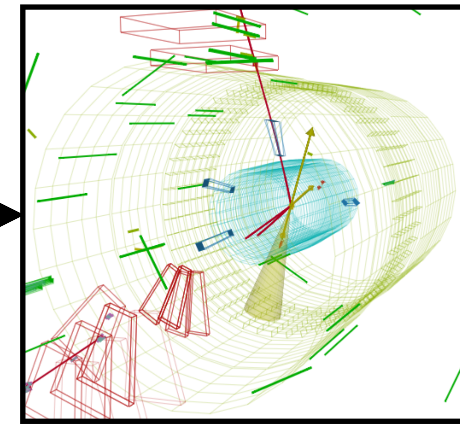


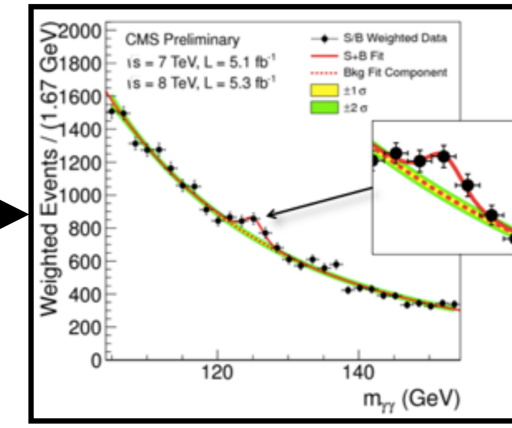
Physics measurements

Data Flow

- Data reconstruction **generally** involves several steps of processing and reduction:







Stage	Trigger	Event Reconstruction	Stripping (AKA Skimming)
Description	Initial selection for finding interesting events.	Reconstruct triggered data into list of particles.	Signature selection trained by prior physics knowledge.
Hardware Implemented	Local electronics or CPU/ GPU processing farm.	Inside trigger and/or the Grid (see later).	The Grid.
Timescale	Live.	Almost live (requires detector calibration). Repeated ~yearly.	Any point, ~monthly turn around.
Data reduction factor	10^6* (permanent loss).	10x (used for Physics).	Analysis dependant.

Lecture 1   Lecture 2

*CMS example.

Triggers

Triggers

- The trigger decides what data to save in real time.
  - Typically just 1 in a million events considered interesting by trigger.
  - Can be formed of several levels, data reduction and increasing complexity at each step.
  - Executed on FPGAs and/or local CPU/GPU processing farms.

FPGAs: Use simple algorithms that perform quickly.

CPU/GPU: Perform partial/full event reconstruction (see later).

- Efficient (aim >80%) to maximise storage and reconstruction resources.
  - All other events permanently discarded.

- Many O(50) configurations used (a few examples shown later):
  - Tuned to physics analysis of the experiment.
  - Try to be least biased whilst recording interesting events.

Triggers

- The trigger decides what data to save in real time.
  - Typically just 1 in a million events considered interesting by trigger.
  - Can be formed of several levels, data reduction and increasing complexity at each step.
  - Executed on FPGAs and/or local CPU/GPU processing farms.

Use simple algorithms that perform quickly. (FPGAs)

Perform partial/full event reconstruction (see later). (CPU/GPU)

- Efficient (aim >80%) to maximise storage and reconstruction resources.
  - All other events permanently discarded.

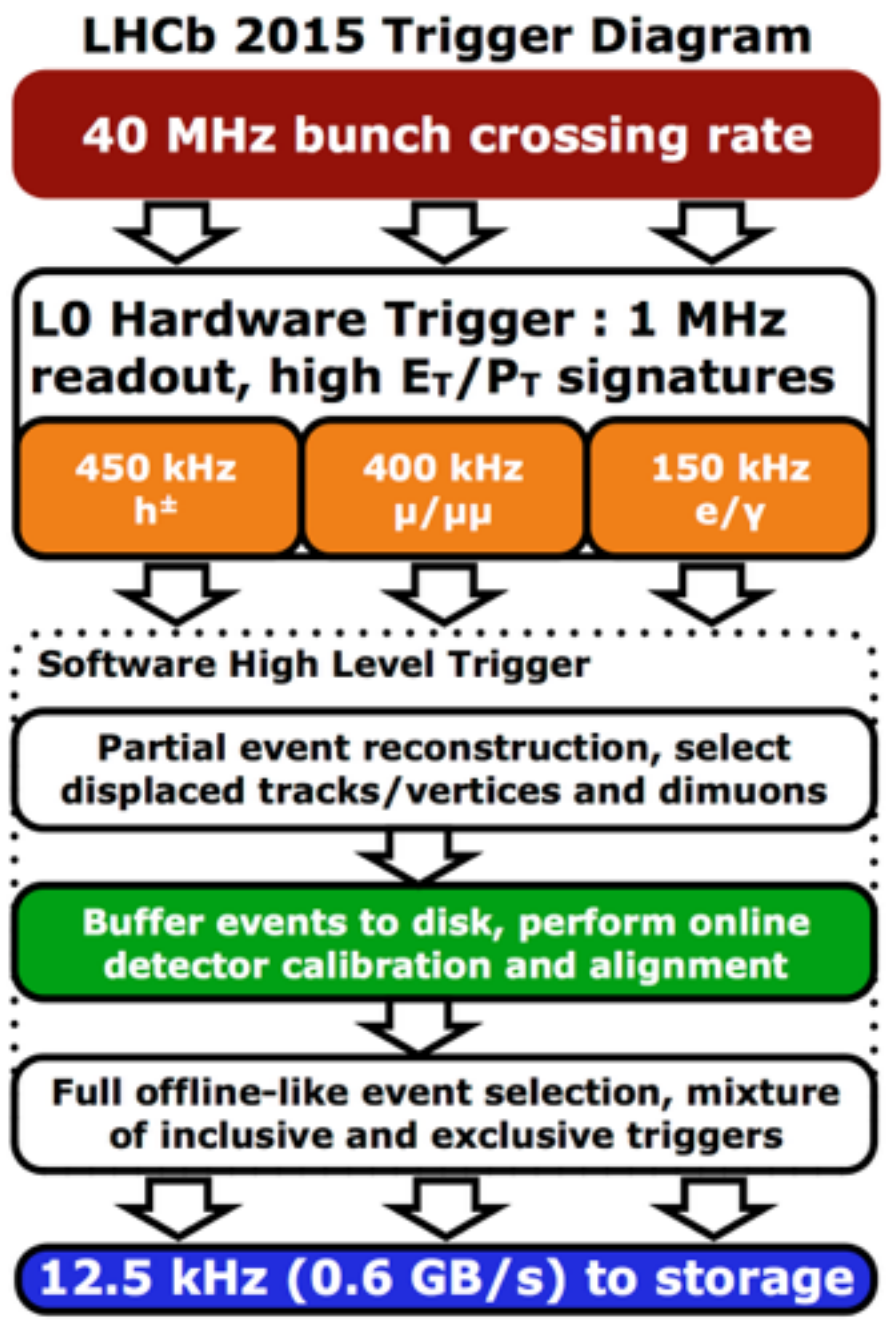
- Many O(50) configurations used (a few examples shown later):
  - Tuned to phy
  - Try to be leas

A theorist has just predicted a new physics particle, produced as frequently as the Higgs Bosons in LHC, but with significantly different topology and energy distributions: what are the chances it's discovered?

- *Almost zero!*

Triggers Worked Example - LHCb

- Data rate of entire detector too high for all to be used in trigger:
  - Use a subset of information.
  - Introduce multiple levels of triggers → use more information in higher levels.
- May deliberately reduce resolution of detectors to reduce data size.
  - E.g. combine cells in tracker, or use a less precise data type.
- LHCb model is very efficient, allowing for physics analysis immediately after trigger - not always possible.



Particles cross each other every 25ns.

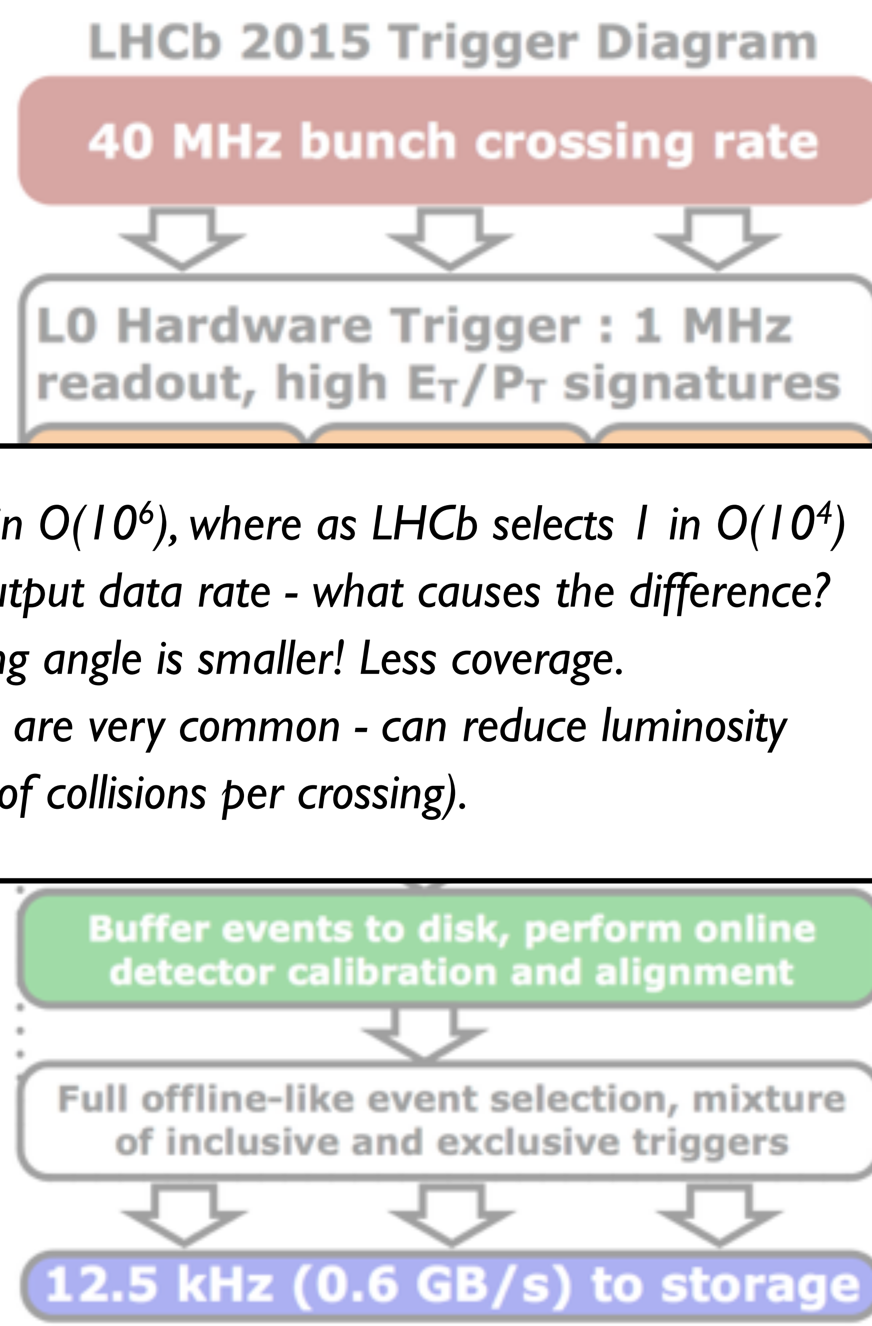
First trigger selects 1 in 40 events. Performed in hardware, using subset of detector data.

*Software trigger selects 1 in 100 events. Since called less frequently, can use **full** detector information for full event reconstruction.*

Combined trigger selects 1 in 40k events for physics analysis.

Triggers Worked Example - LHCb

- Data rate of entire detector too high for all to be used in trigger:
  - Use a subset of information.
  - Introduce multiple levels of triggers → use more information in higher levels.
- May deliberately r resolution of dete reduce data size.
  - E.g. combine cells in tracker, or use a less precise data type.
- LHCb model is very efficient, allowing for physics analysis immediately after trigger - not always possible.



Particles cross each other every 25ns.

First trigger selects 1 in 40 events. Performed in hardware, using subset of detector data.

*Software trigger selects 1 in 100 events. Since called less frequently, can use **full** detector information for full event reconstruction.*

Combined trigger selects 1 in 40k events for physics analysis.

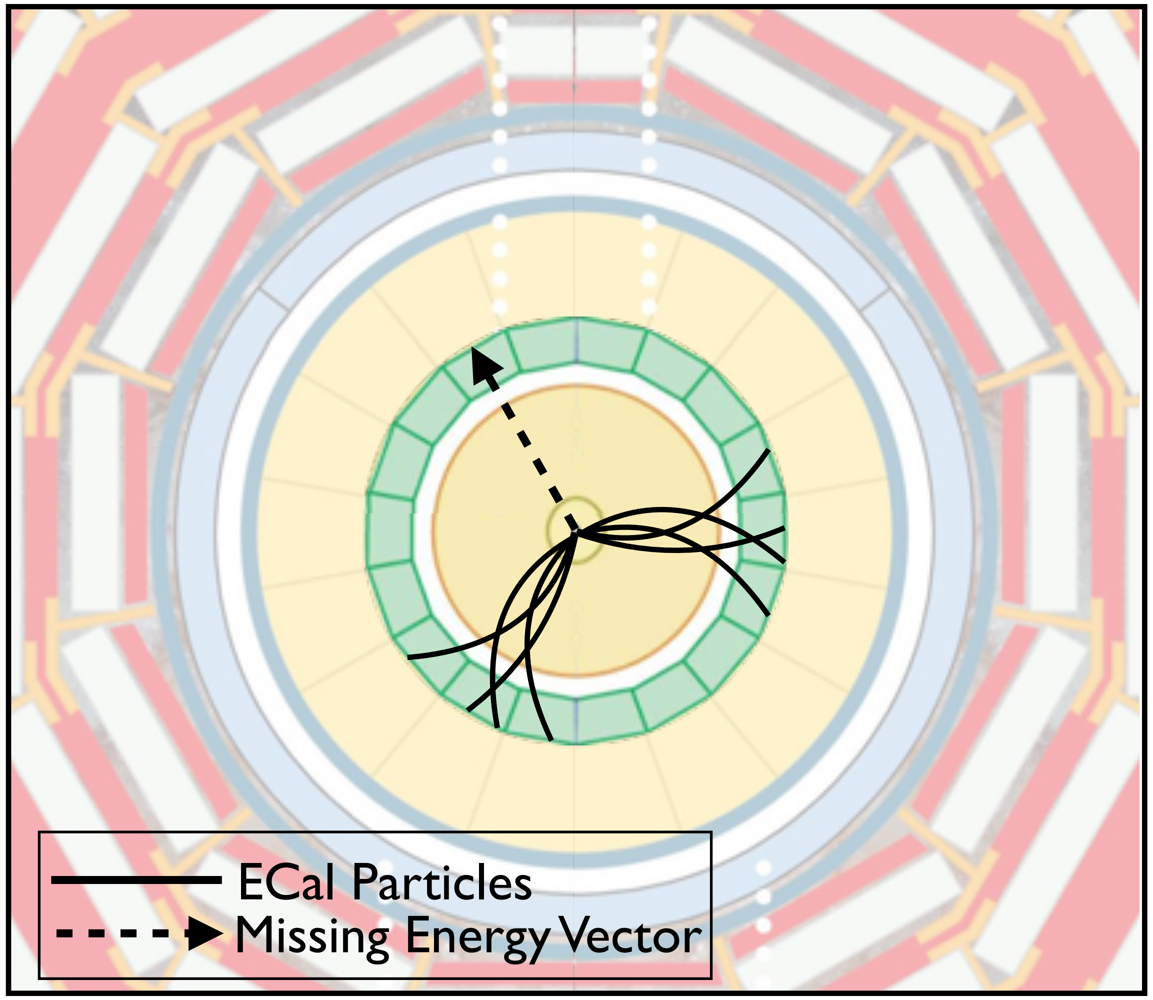
CMS selects 1 in $O(10^6)$, where as LHCb selects 1 in $O(10^4)$ for the same output data rate - what causes the difference?

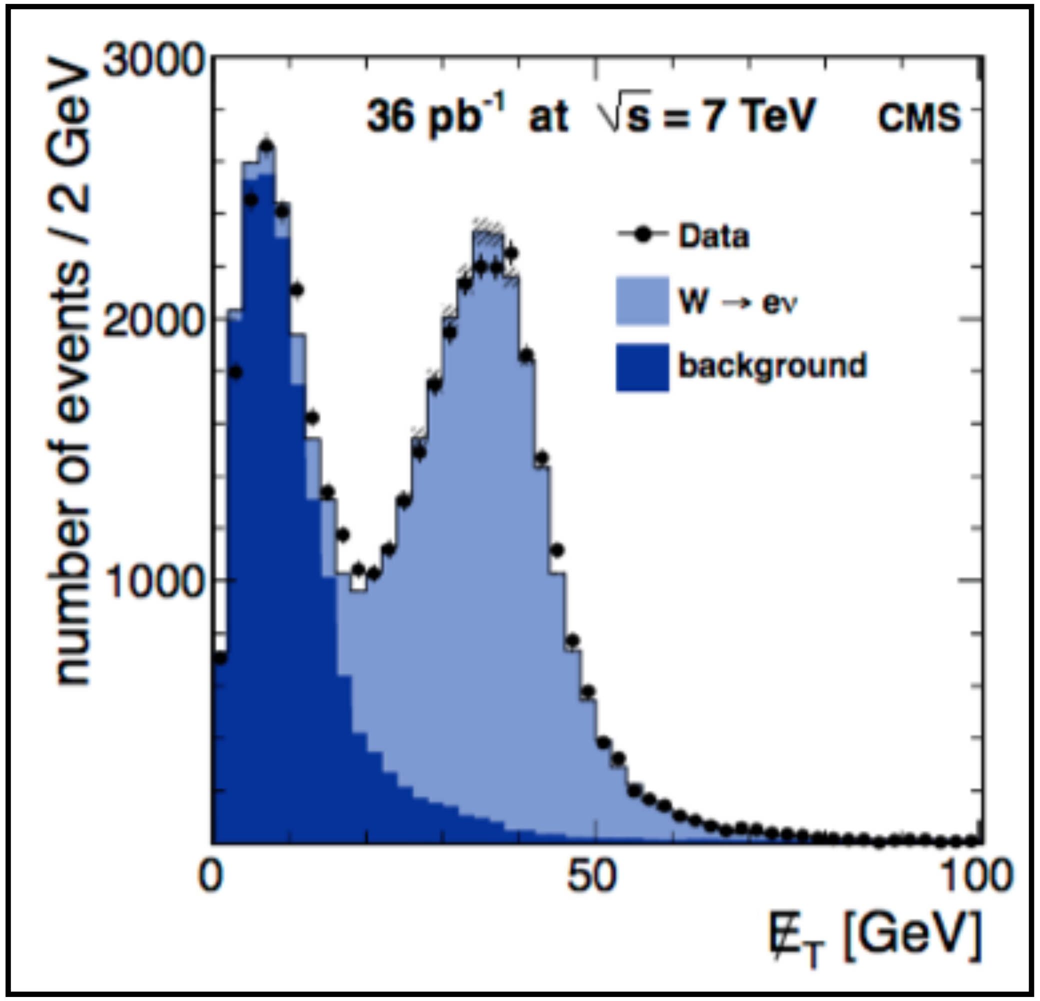
- *LHCb opening angle is smaller! Less coverage.*
- *LHCb events are very common - can reduce luminosity (i.e. number of collisions per crossing).*

Trigger Example 1 - ECal at CMS (L0)

*Filtering for events we **don't** know about.*

- Typical algorithms identify data variables that distinguish signals from backgrounds.
- Difficult to define signal and background for discoveries - try to keep general:





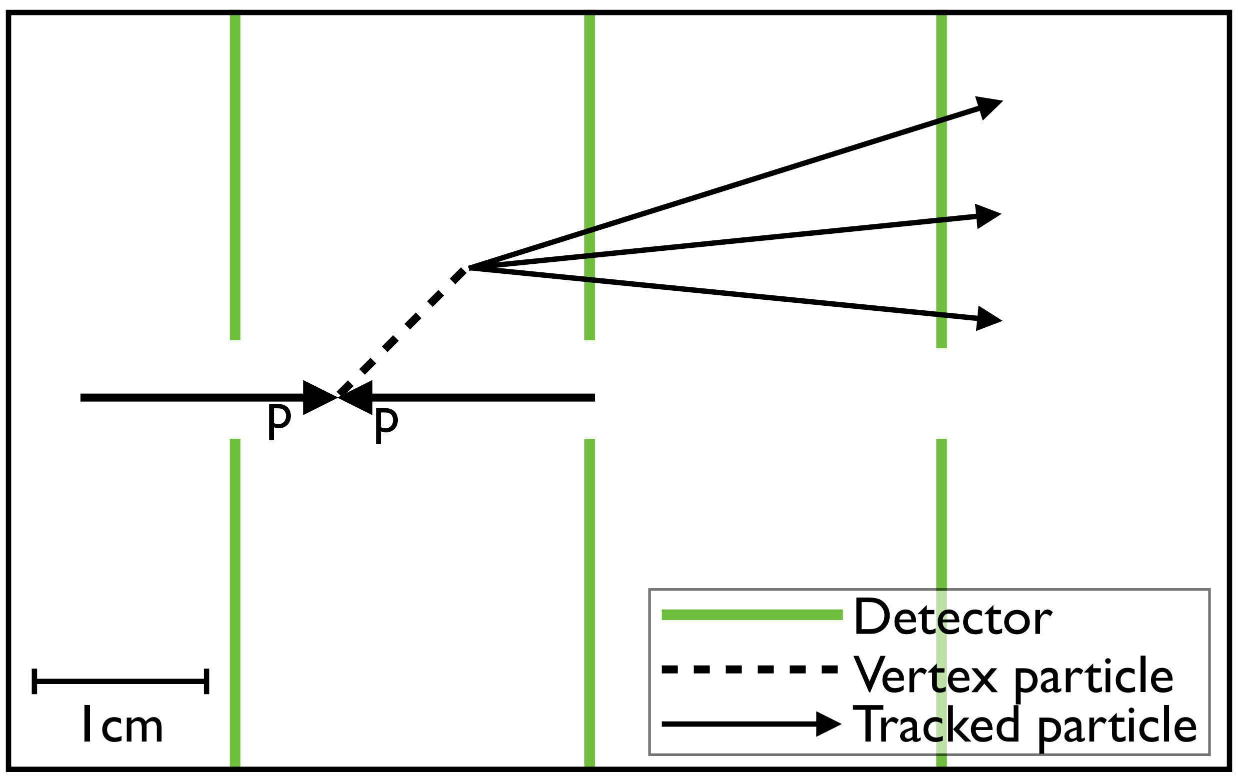
- Known processes are approx symmetrically distributed in energy. Missing energy could be a sign of something new produced - **dark matter**.

- Separation between background and W→eν events as missing energy in the traversal direction.

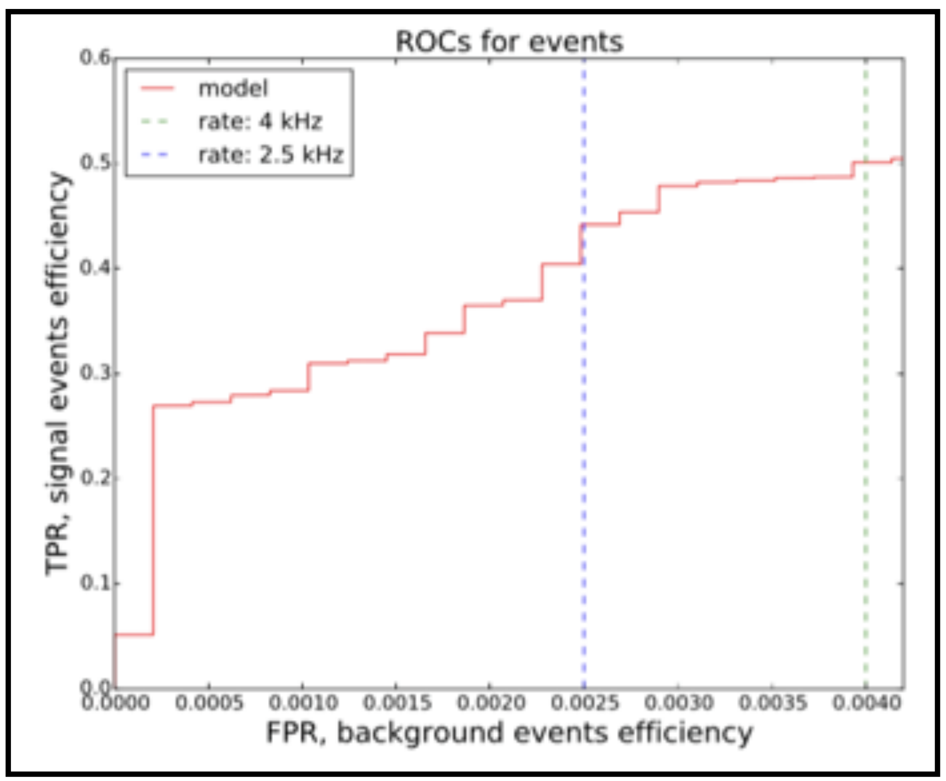
Trigger Example 2 - Vertexing at LHCb

*Filtering for events we **do** know about.*

- LHCb is designed for precision studies of collisions involving b quarks:
  - Complimentary to direct searches (e.g. discovering new particles).
  - Perform indirect searches by comparing experimental rates to theoretical predictions.
  - Trigger can be designed for specific physics: typical signature involves a displaced vertex.
    - Not easy to reconstruct - performed in software farm at higher level trigger.
    - (Reconstruction of these objects discussed in lecture 2).



Displaced vertex in LHCb - signature of interesting event.



LHCb VELO HLT ROC.

Trigger Bias

- Data sets from triggers inevitably biased by trigger.
  - E.g. experiment finds deficit Higgs candidates with $E_T < 5$ GeV (unsurprising if $E_T^{Trig} = 5$ GeV).
- Can be accounted for:
  - Comparisons with simulation, although many factors (detector performance, collider conditions).
  - Comparison with non-triggered data:
    - Far lower rate! Have to extrapolate.

Trigger Hardware Example 1 - LHCb

Frequent small events.

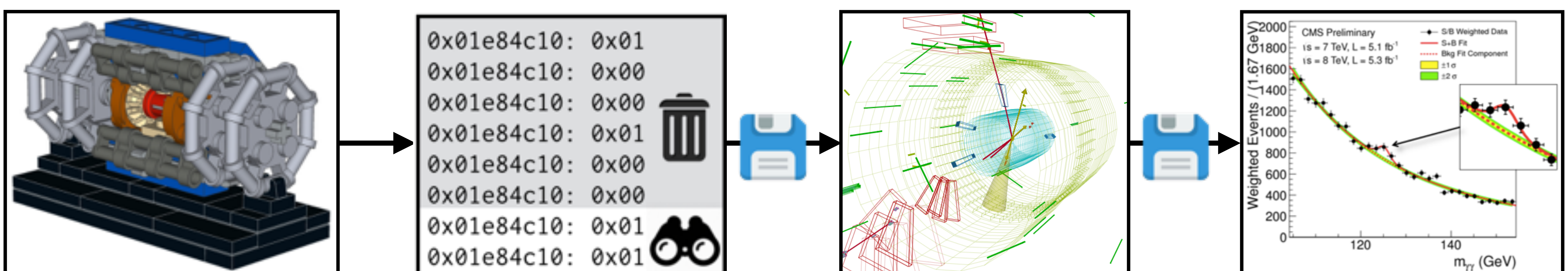
Level	Level0*	High Level Trigger*
Input rate	40 MHz	1MHz, 70GB/s
Hardware	FPGAs.	Local cluster using 20k[7] CPUs.
Output rate	1MHz	10kHz, 700MB/s[2]
Event filter factor	40x	100x
Notes	Uses subset of detector data (ECal and muons only).	Full reconstruction performed.

*Values accurate to ±50%.

Lecture 1 Summary

- LHC detectors output a large amount of data.
  - Four major experiments, each outputting O(1) GB/s physics data (post trigger).
  - Many results required live (detector monitoring, calibrations etc.).
  - Reconstruction performed on the GRID (Lecture 2).

- Triggers are used to filter for interesting events in real time.
  - Can be multilayered, using electronics, CPUs and GPUs.
  - Complexity ranges from simple algorithms through to full event reconstruction.
  - Designs heavily influenced by the physics aims of the experiment.



Lecture 2

Event Reconstruction

Event Reconstruction

- Triggered detector collision data → particle interactions.
- Seek the following information as input for physics analysis:
  - What particles were created?
  - Where were they produced?
  - What were the parent particles?

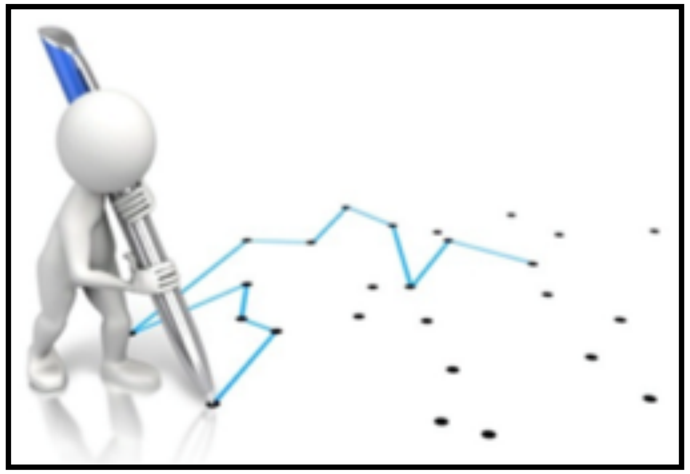
- To find this, perform (at least):
  - **Tracking**: Reconstruct particle trajectories into tracks.
  - **Vertexing**: Group particles into vertices.
  - **Particle ID**: Find the particle identification of each track (e.g. a muon, electron etc.).

- Requirements:
  - Fast.
  - Good quality for physics analysis.

Usually anti correlated - a fast algorithm often leads to inefficiency and impurities (see later).

Tracking Algorithms

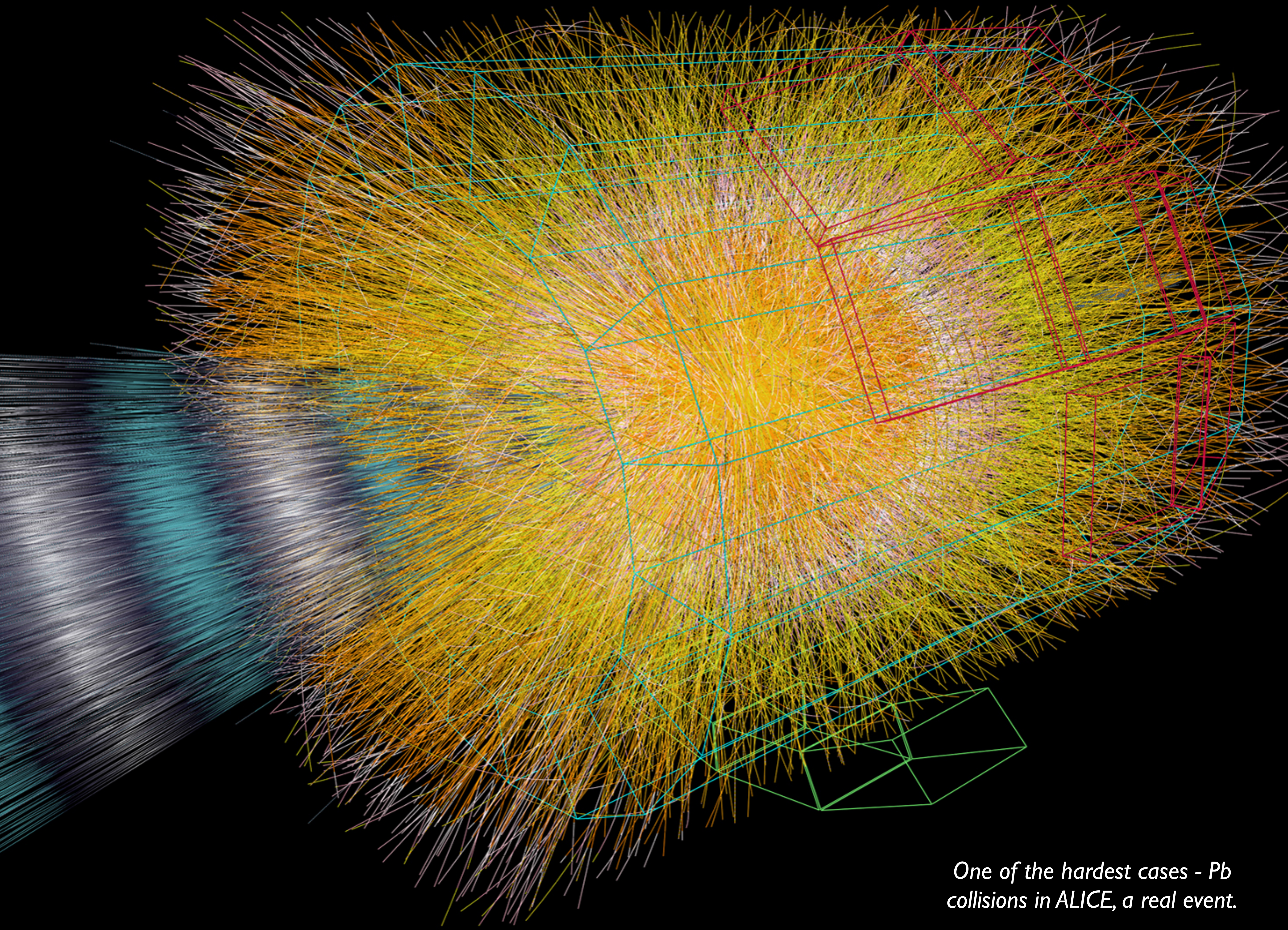
Aim: to play a game join the dots at 1kHz with many fake dots.



- Tracking particles through detectors involves two step.
  - Pattern recognition: identifying which detector hits for a track.
  - Track fit: approximate the path of the particle with an equation.

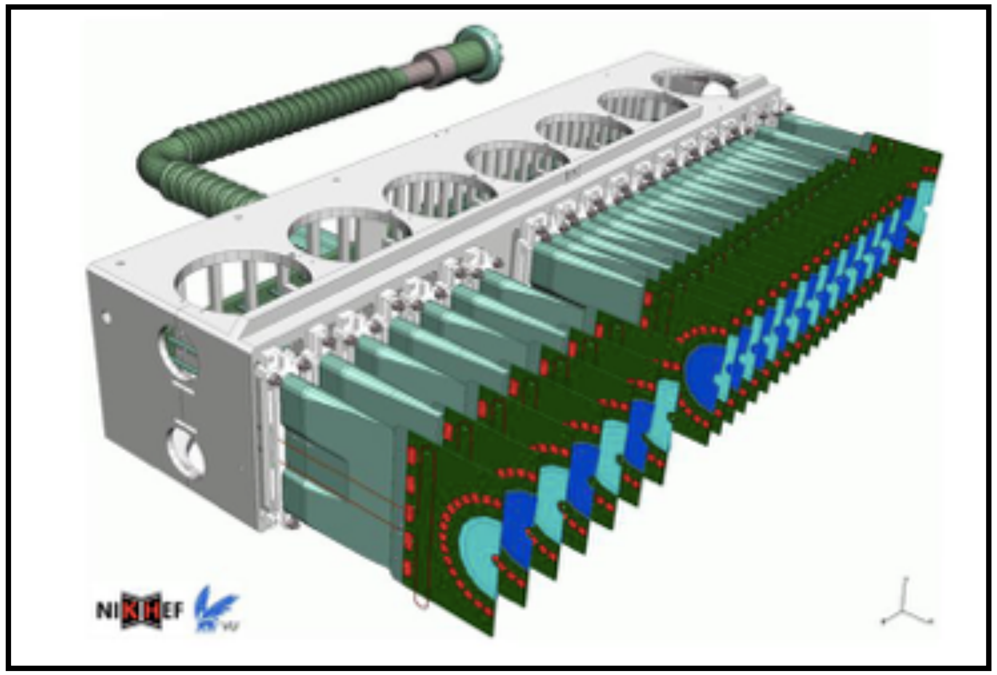
- No one size fits all solution.
  - Many detectors use different combinations of algorithms (e.g. LHCb uses 4 different algorithms for difference combinations of sub detectors, but basic ideas are the same).
  - Usually a trade off between:
    - Efficiency: **fraction of real tracks found**
    - Purity: **fraction of tracks that are real**
    - Computational speed.

Typically these two are anti correlated: a good efficiency typically has a bad purity, and vice versa. Both good efficiency and purity is usually computationally expensive - see later.

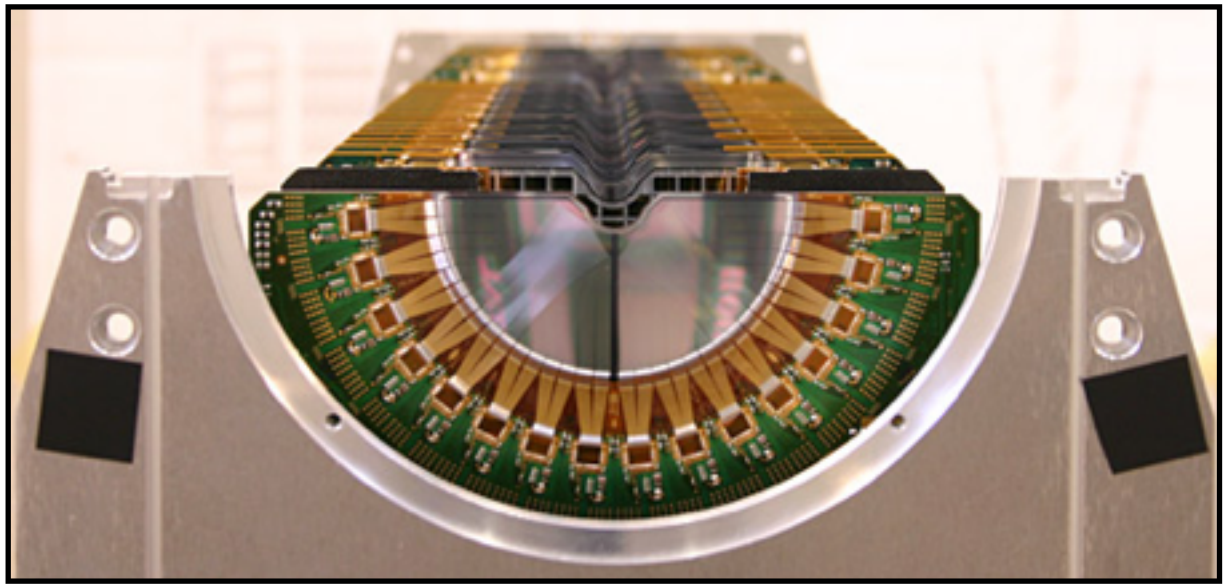


One of the hardest cases - Pb collisions in ALICE, a real event.

Tracking - Pattern Recognition Example



- Consider tracking in the Vertex Locator (VELO) of LHCb.
  - The VELO is a silicon strip detector, consisting of 42 layers.
  - Gives an x, y and z position.



LHCb VELO

Tracking - Pattern Recognition Example

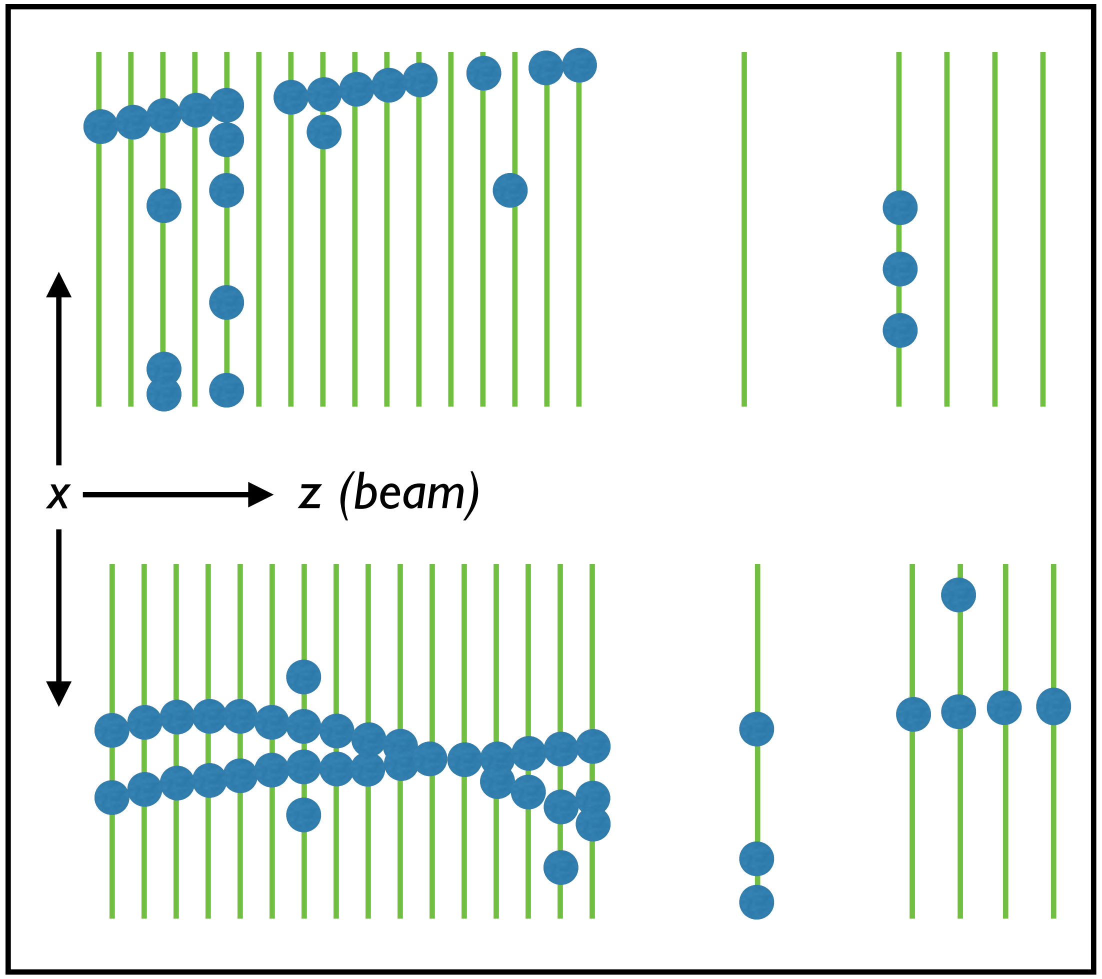
- Looking side on:
  - Particle tracks clearly visible to eye.
  - Extra hits present, typically electrical noise or secondary short tracks.

- Recall data points in the format:

  (x, y, z, ~~time~~)

  - Time resolution only accurate to which collision the particles come from (25ns, sometimes worse…).

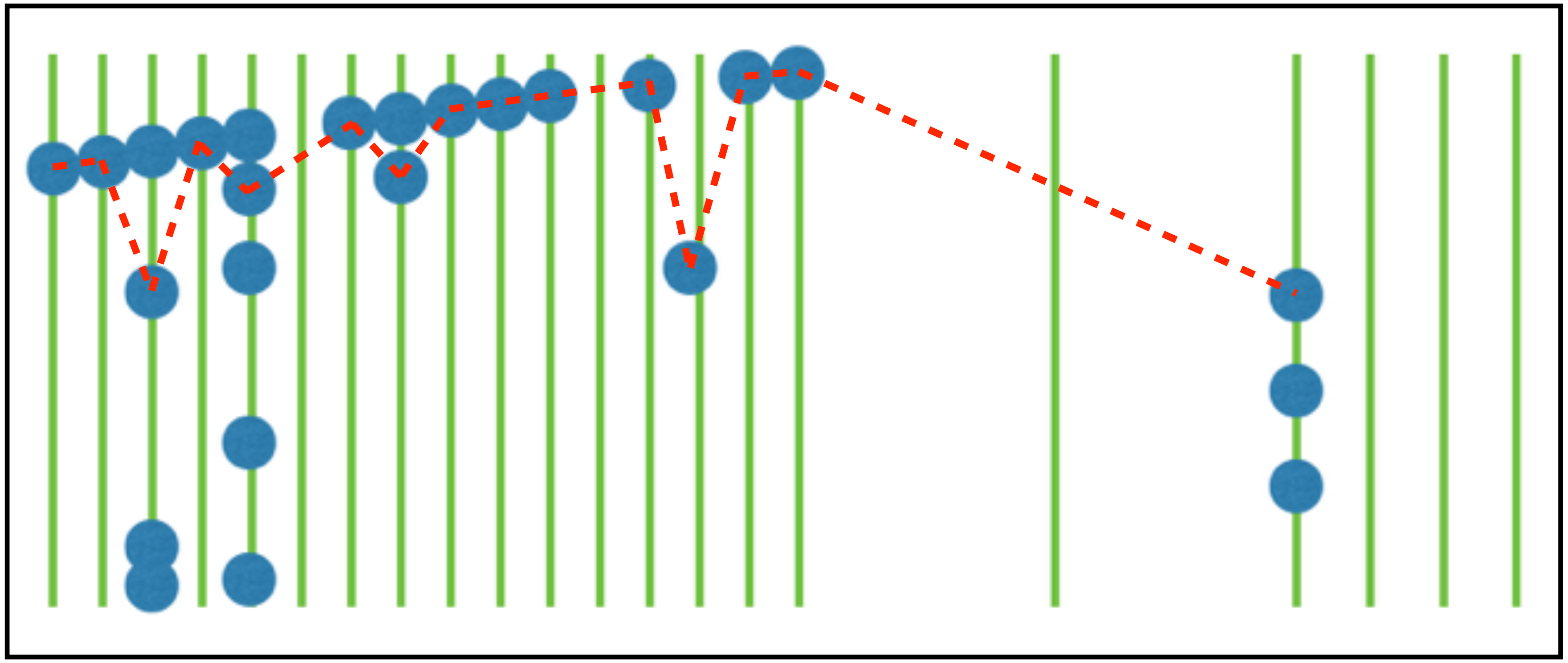
- Have to find an algorithm to track using this information and in these conditions. Many choices - consider the following (LHC) examples…



LHCb VELO data event (2d projection)

Tracking - Pattern Recognition Example

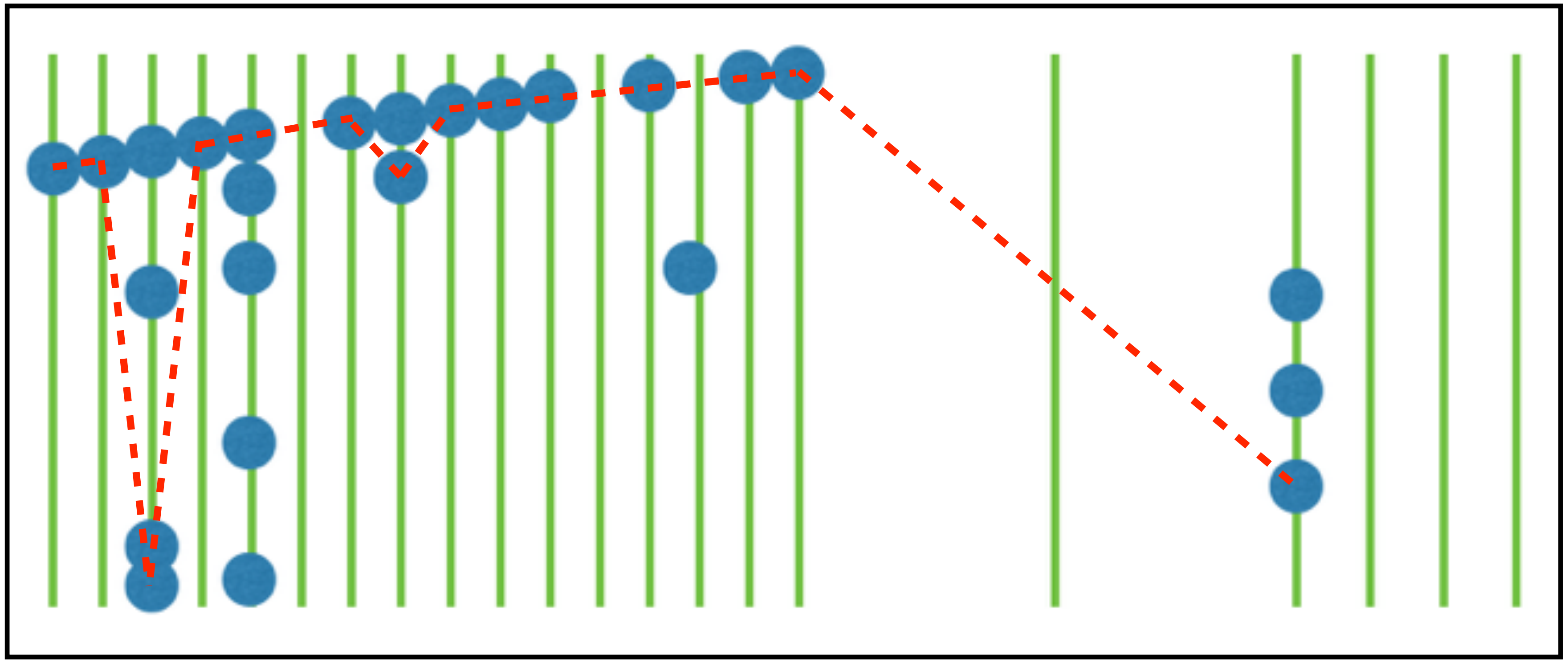
Name	Description	Scalability
Combinatorial	• Form every track from each possible combination of hits. • Access each track by quality (e.g. χ^2) and tag.	$n_{Tracks}!$



LHCb VELO data event (2d projection, top half)

Tracking - Pattern Recognition Example

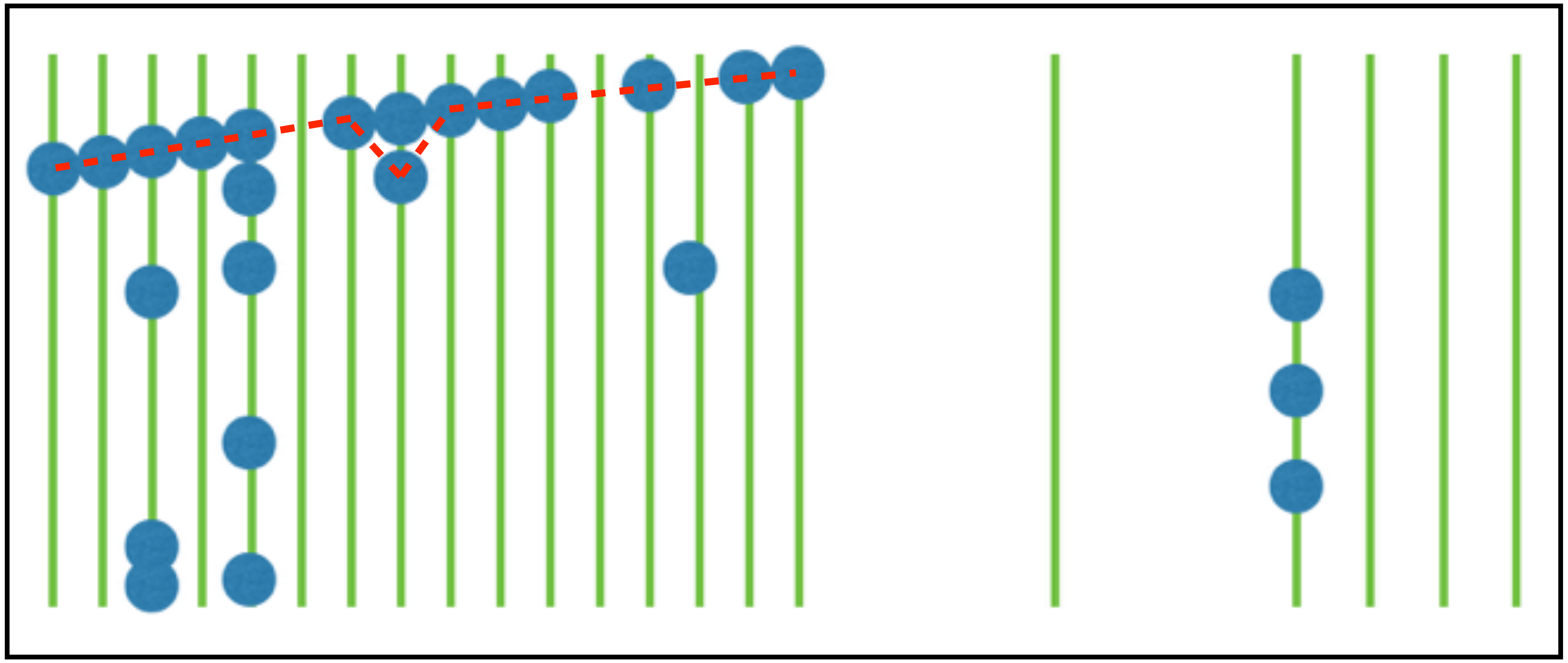
Name	Description	Scalability
Combinatorial	• Form every track from each possible combination of hits. • Access each track by quality (e.g. χ^2) and tag.	$n_{Tracks}!$



LHCb VELO data event (2d projection, top half)

Tracking - Pattern Recognition Example

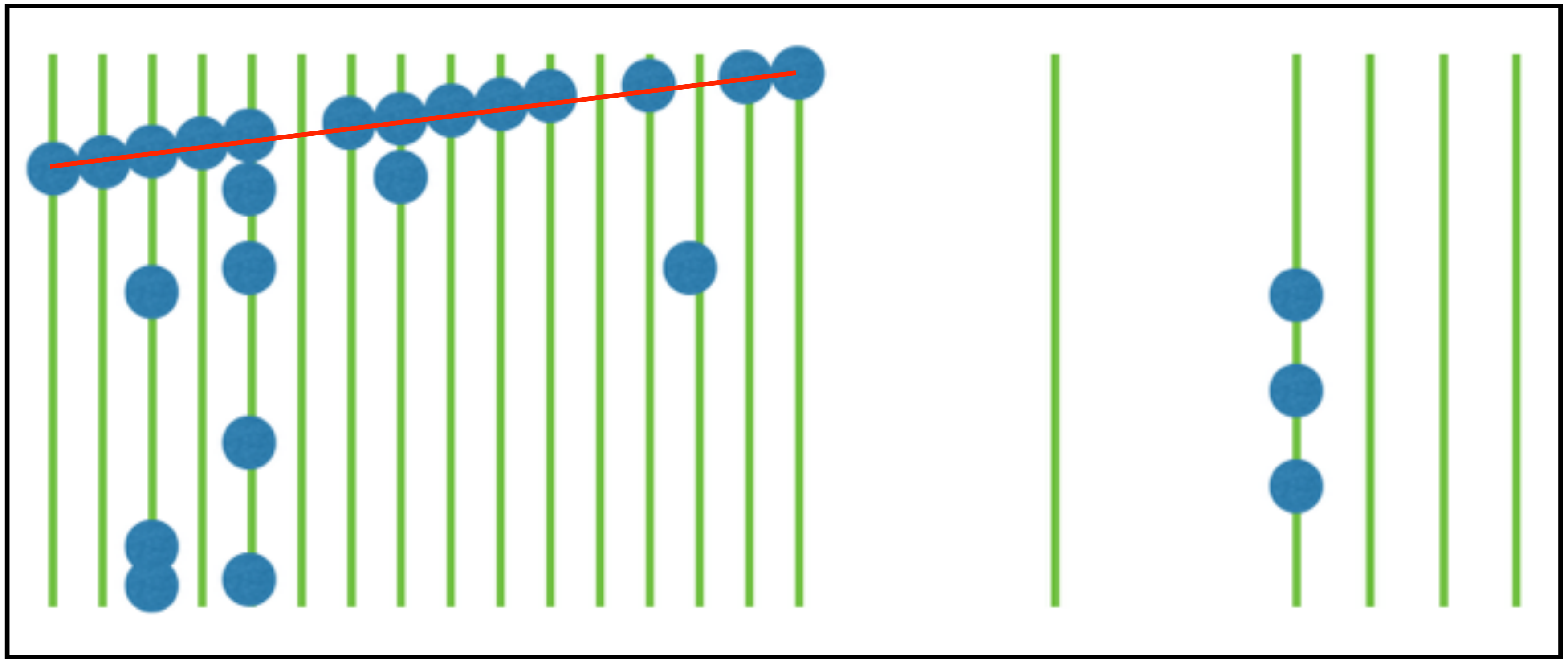
Name	Description	Scalability
Combinatorial	• Form every track from each possible combination of hits. • Access each track by quality (e.g. χ^2) and tag.	$n_{Tracks}!$



LHCb VELO data event (2d projection, top half)

Tracking - Pattern Recognition Example

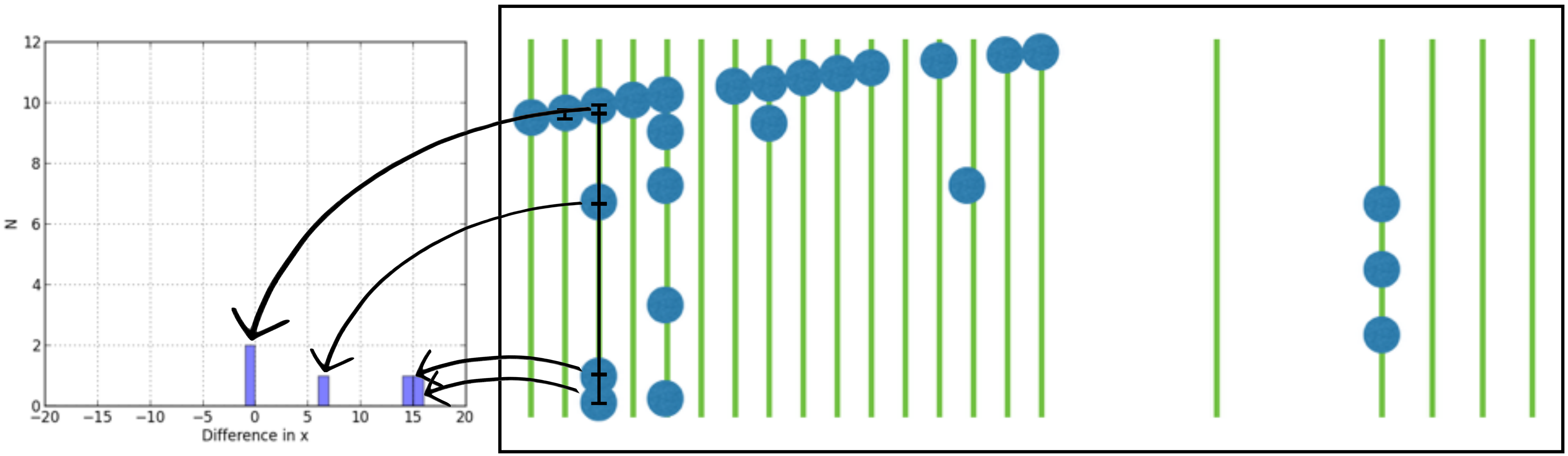
Name	Description	Scalability
Combinatorial	• Form every track from each possible combination of hits. • Access each track by quality (e.g. χ^2) and tag.	$n_{Tracks}!$



LHCb VELO data event (2d projection, top half)

Tracking - Pattern Recognition Example

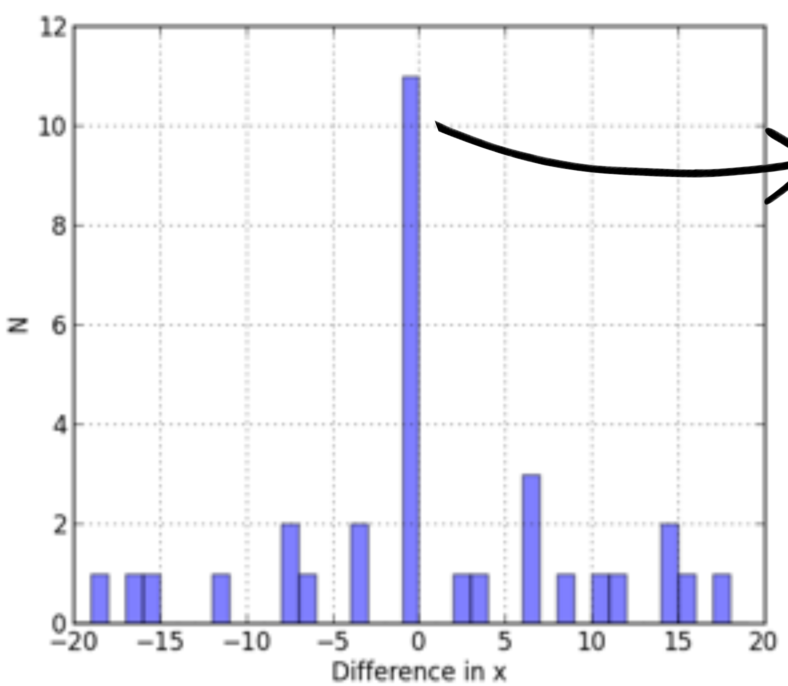
Name	Description	Scalability
Combinatorial	• Form every track from each possible combination of hits. • Access each track by quality (e.g. χ^2) and tag.	$n_{Tracks}!$
Hough Transform	• Transform points into a system where clusters form. • If straight tracks, take the difference between consecutive hits. • Group (e.g. in a histogram) and tag peaks.	x

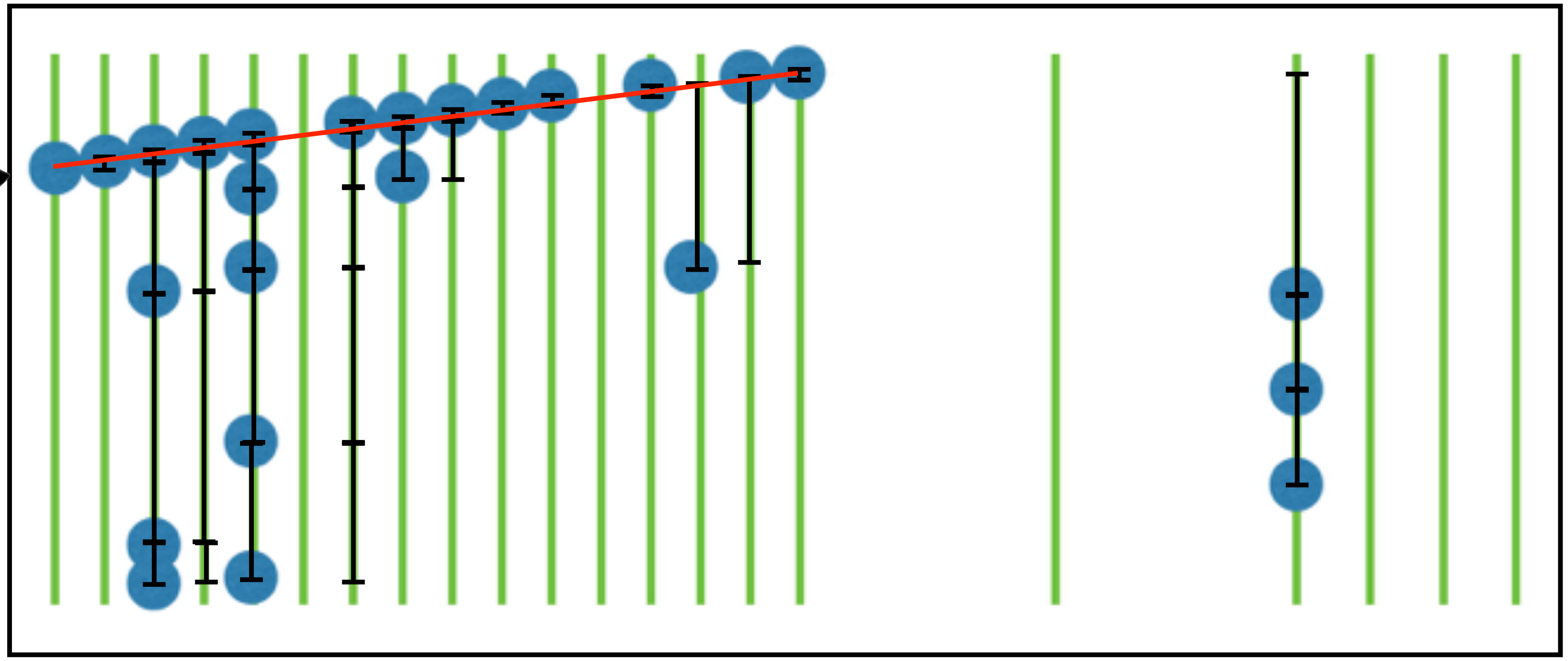


LHCb VELO data event (2d projection, top half)

Tracking - Pattern Recognition Example

Name	Description	Scalability
Combinatorial	• Form every track from each possible combination of hits. • Access each track by quality (e.g. χ^2) and tag.	$n_{Tracks}!$
Hough Transform	• Transform points into a system where clusters form. • If straight tracks, take the difference between consecutive hits. • Group (e.g. in a histogram) and tag peaks.	x

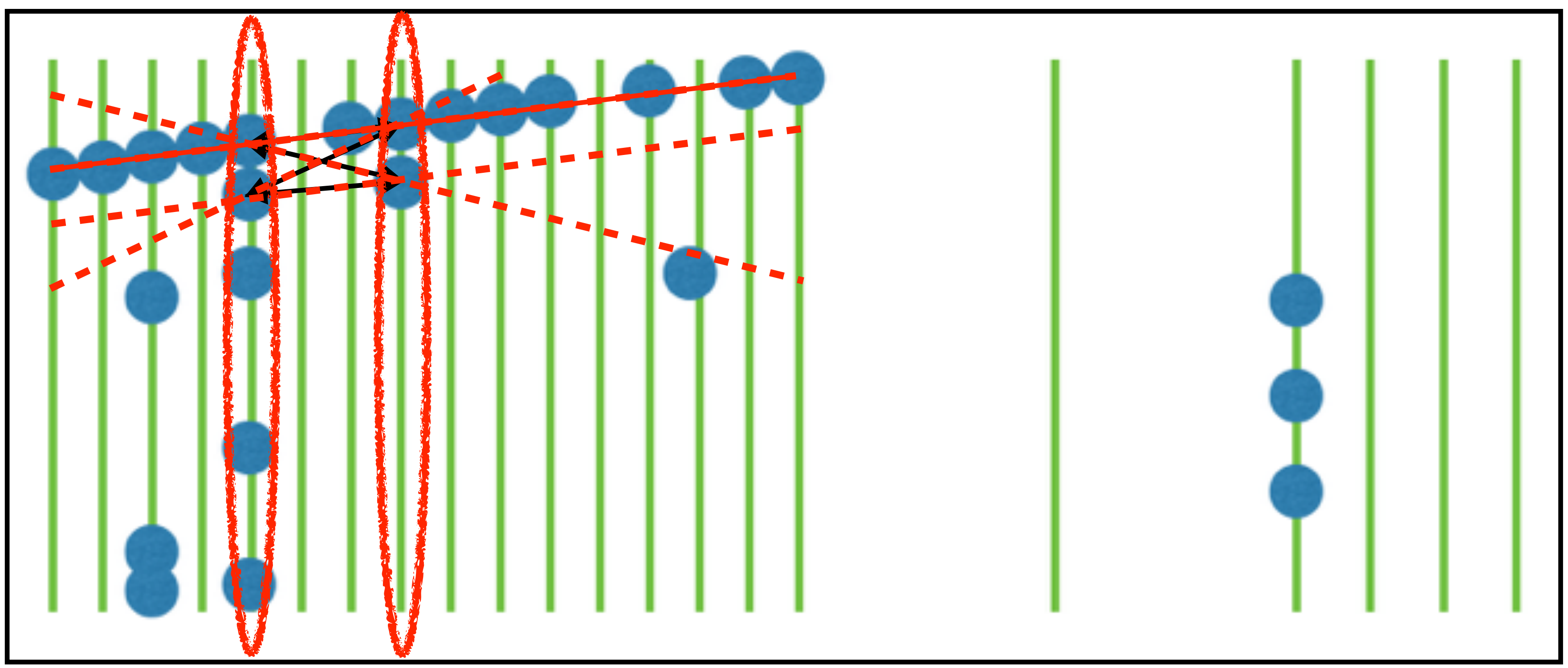




LHCb VELO data event (2d projection, top half)

Tracking - Pattern Recognition Example

Name	Description	Scalability
Combinatorial	• Form every track from each possible combination. • Access each track by quality (e.g. χ^2) and tag.	$n_{Tracks}!$
Hough Transform	• Transform points into a system where clusters form. • E.g. for straight tracks, take the difference between consecutive hits. • Group (e.g. in a histogram) and tag peaks.	x
Seeding	• Form seeds from pairs of hits on a sub set of the detector. • Extrapolate the seed and count hits intercepted. • Tag if sufficient number of hits.	nlog(n)



LHCb VELO data event (2d projection, top half)

Tracking - Pattern Recognition Example

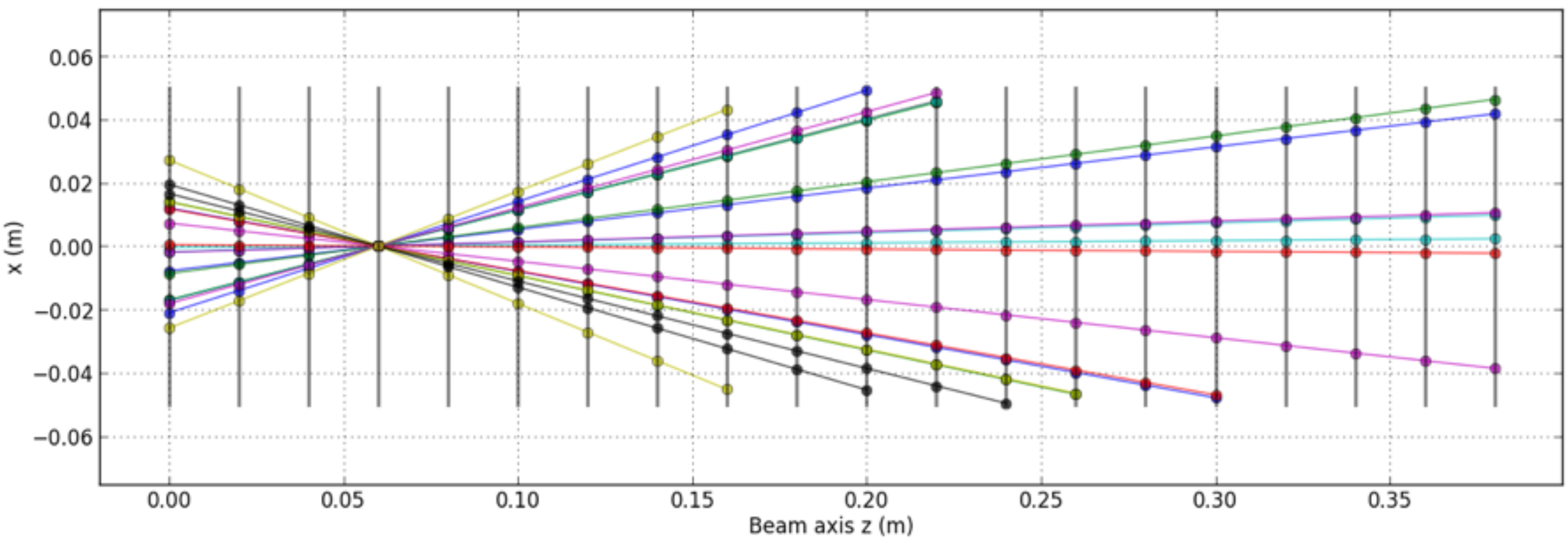
Name	Description	Scalability
Combinatorial	• Form every track from each possible combination. • Access each track by quality (e.g. χ^2) and tag.	$n_{Tracks}!$
Hough Transform	• Transform points into a system where clusters form. • E.g. for straight tracks, take the difference between consecutive hits. • Group (e.g. in a histogram) and tag peaks.	x
Seeding	• Form seeds from pairs of hits on a sub set of the detector. • Extrapolate the seed and count hits intercepted. • Tag if sufficient number of hits.	nlog(n)

Question: which algorithm to pick?
Even in the case where efficiency and purity are constant?
Still not enough information…

Tracking - Pattern Recognition Algorithms

- Recall three main factors in choosing such algorithms:
  - Efficiency: **fraction of real tracks found**
  - Purity: **fraction of tracks that are real**
  - Computational speed.

- Toy simulation for LHCb VELO:



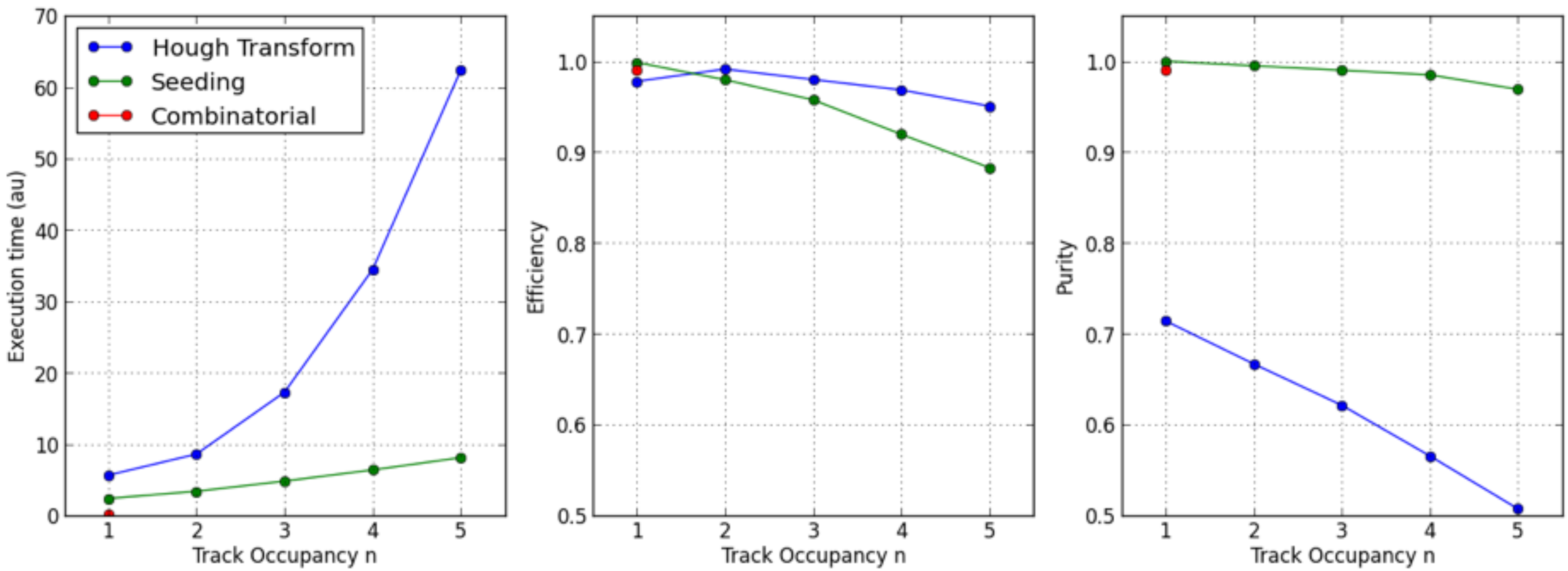
LHCb VELO toy event (2d projection)

Tracking - Pattern Recognition Algorithms

- Recall three main factors in choosing such algorithms:
  - Efficiency: **fraction of real tracks found**
  - Purity: **fraction of tracks that are real**
  - Computational speed.

Any use case for green? Curves!

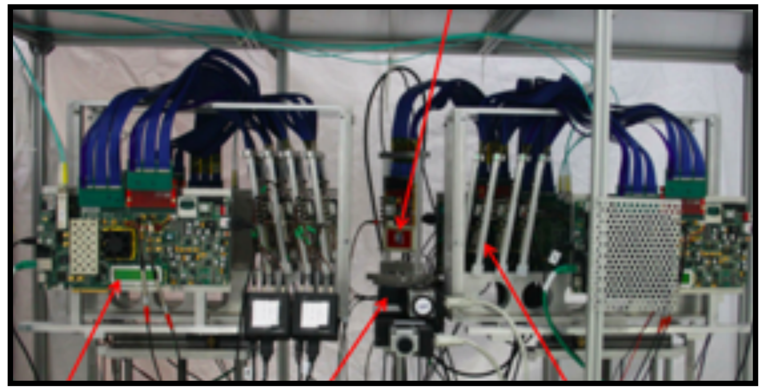
- Toy simulation for LHCb VELO:



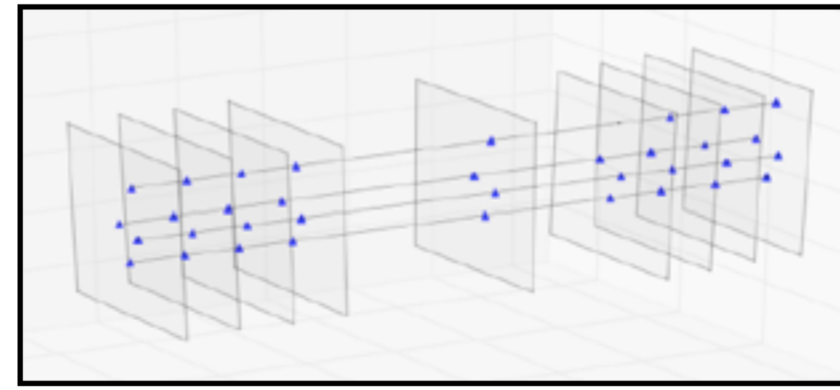
Tracking - Pattern Recognition Algorithms

- Typically use a combination of these algorithms. Each example taken from LHC activities:

- Combinatorial often used at testbeams:
  - Low occupancy, so fast.
  - Efficient and pure.



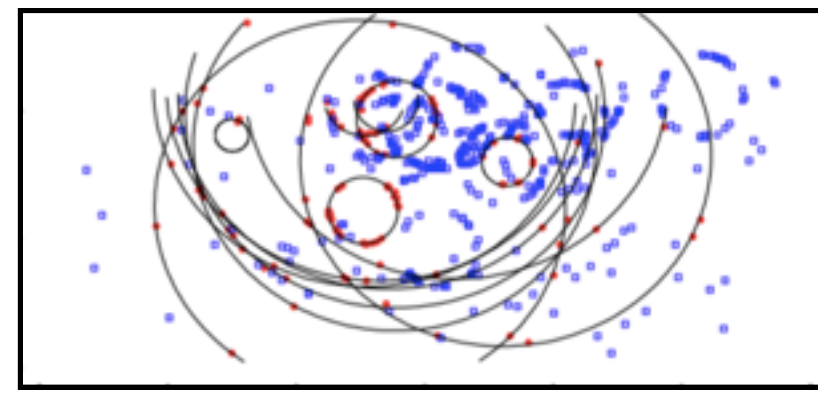
Timepix3 Tracking Telescope



Testbeam Data

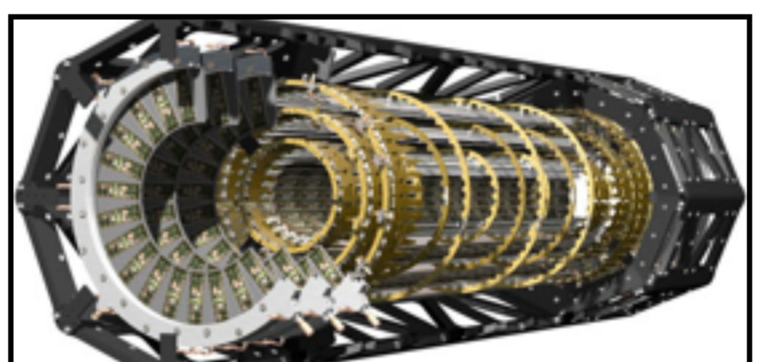
- Hough transforms used for more complicated shapes (e.g. rings in LHCb RICH*).

LHCb RICH Subdetector

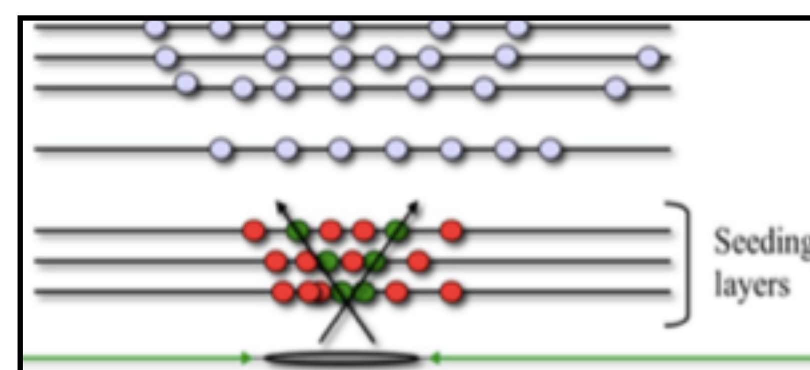


RICH Data

- All LHC experiments use seeding extensively (highest occupancy).



ATLAS Tracker



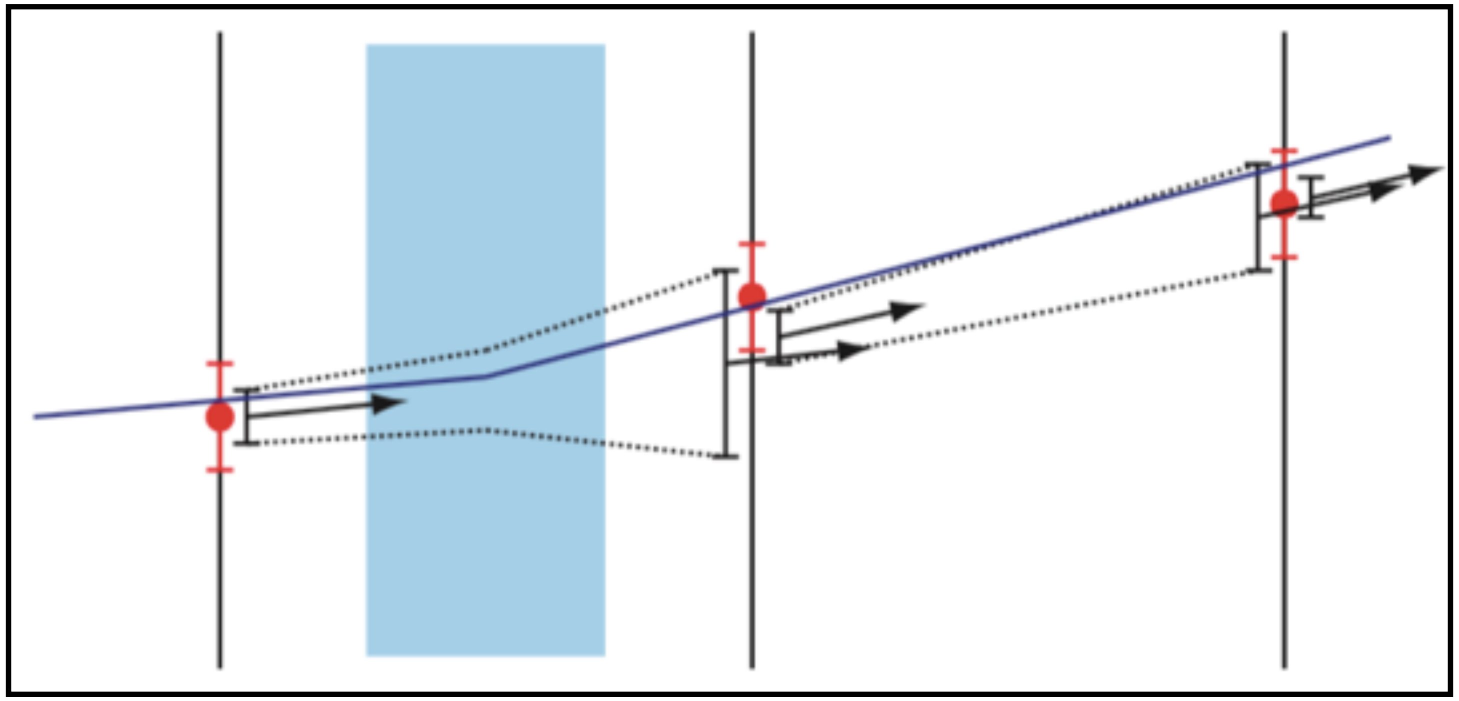
ATLAS Inner Layers

*Often not needed to actually reconstruct rings.

Track Fitting

- Tracking particles through detectors involves two step.
  - Pattern recognition: identifying which detector hits for a track.
  - Track fit: approximate the path of the particle with an equation.

- Typically use a Kalman filter. Basic steps:
  - Track is approximated as a 'zig-zag' (fewer free parameters than co-ordinates!).
  - Start with seed or estimate of track parameters (e.g. straight line fit).
  - Propagate to the next plane (approximating B field, account for scattering in material).
  - Predict position of next particle, weighting by closest hits (needs too be tuned).



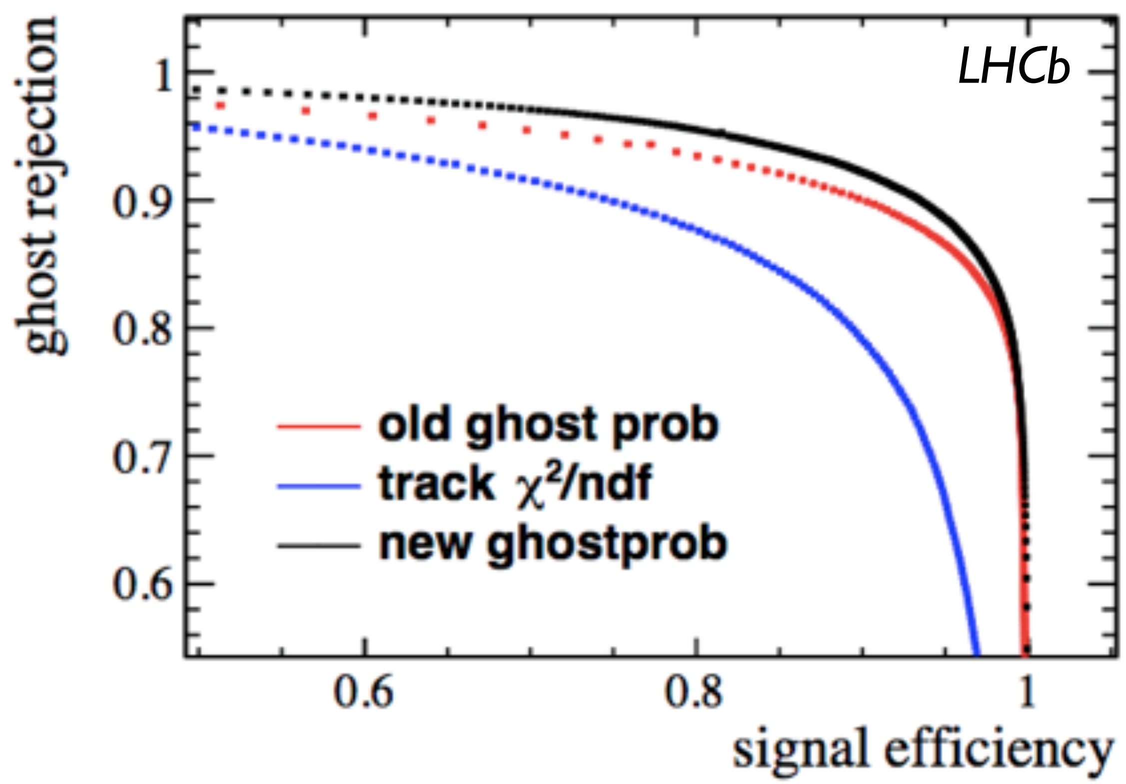
Kalman Filter Example

Track Refinement

- Common to tune pattern recognition to be efficient and impure → refine selection later using full particle information.

  - Can use χ^2 to find well fitting tracks.
  - Can also use/combine with other parameters:
    - Number of hits (complimentary information to χ^2).
    - Fits from different sub detectors

  - Typically build an MVA out of different quality parameters - LHCb uses a neutral net.



- Caution: if fake/ghost tracks are formed from parts of real tracks, they may be lost.

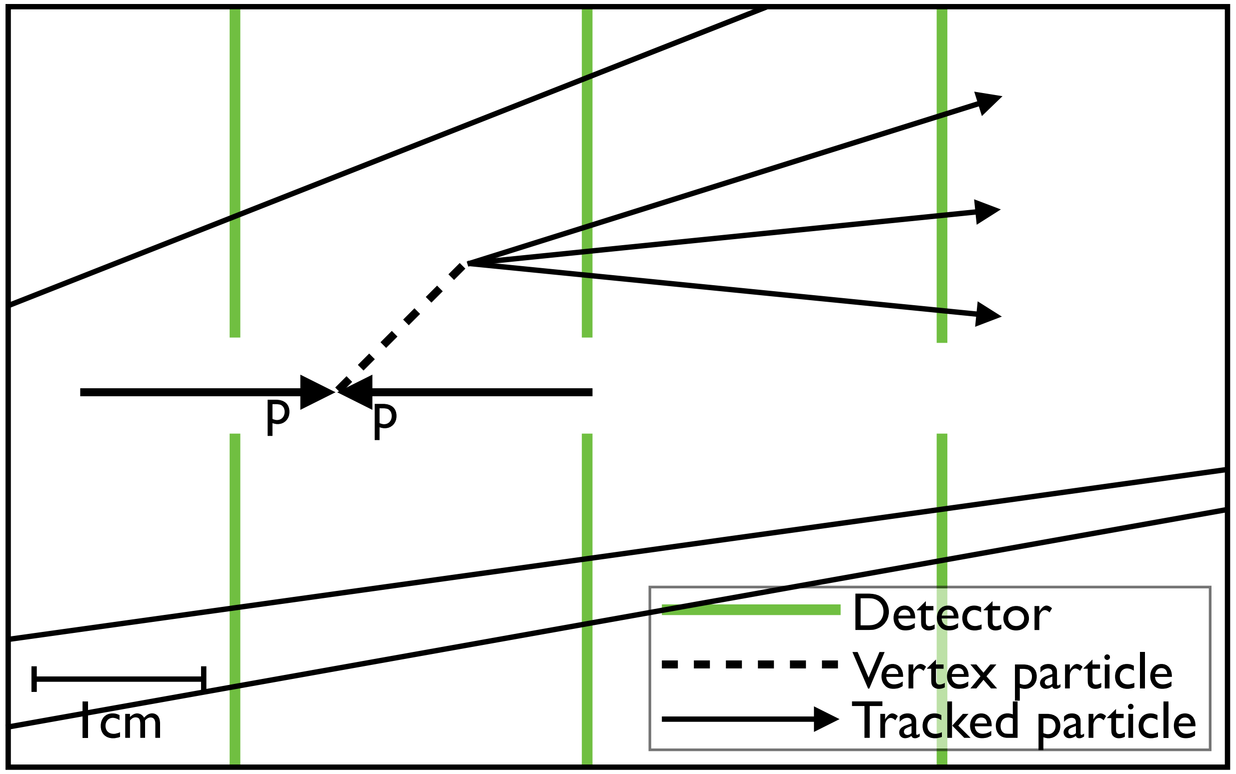
Tracking Notes

- Kalman filter and pattern recognition can be merged into a single step → computationally more efficient (see later).

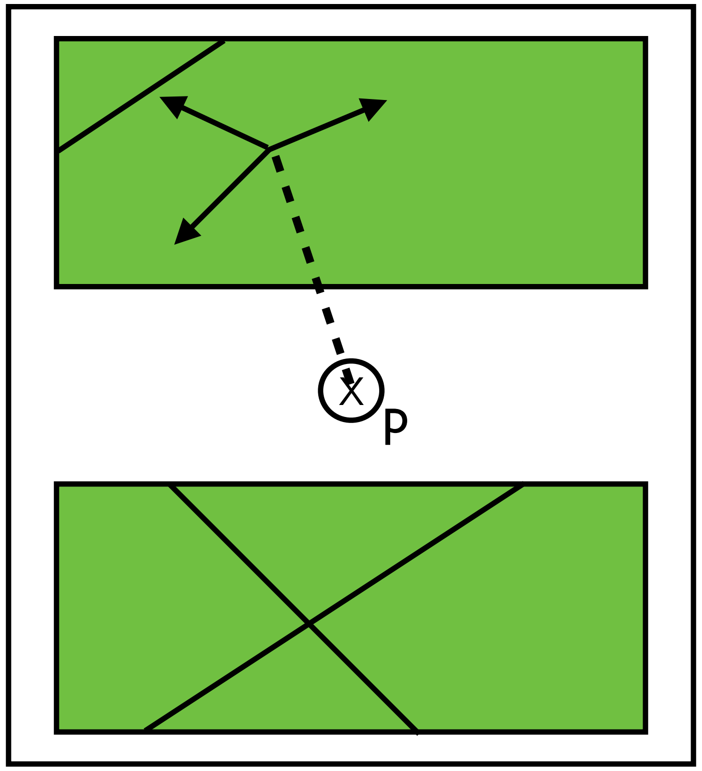
- Detector hits can sometimes form part of multiple tracks.
  - Detector spatial resolution too low to separate near tracks.
  - Secondary tracks produced by interactions with the detector material.

Vertexing (Briefly)

- Vertexing involves clustering tracks that originate from the same point.
  - Easy in cases where vertex location is known - extrapolate all tracks and apply selection criteria.
  - Else, Physics input can narrow search region significantly.
  - Can use analytic methods (e.g. distance of closest approach) to seed search.
  - Common to seed by projecting into 2D plane and searching for point of high "track density" (essentially a peak finding/clustering problem).



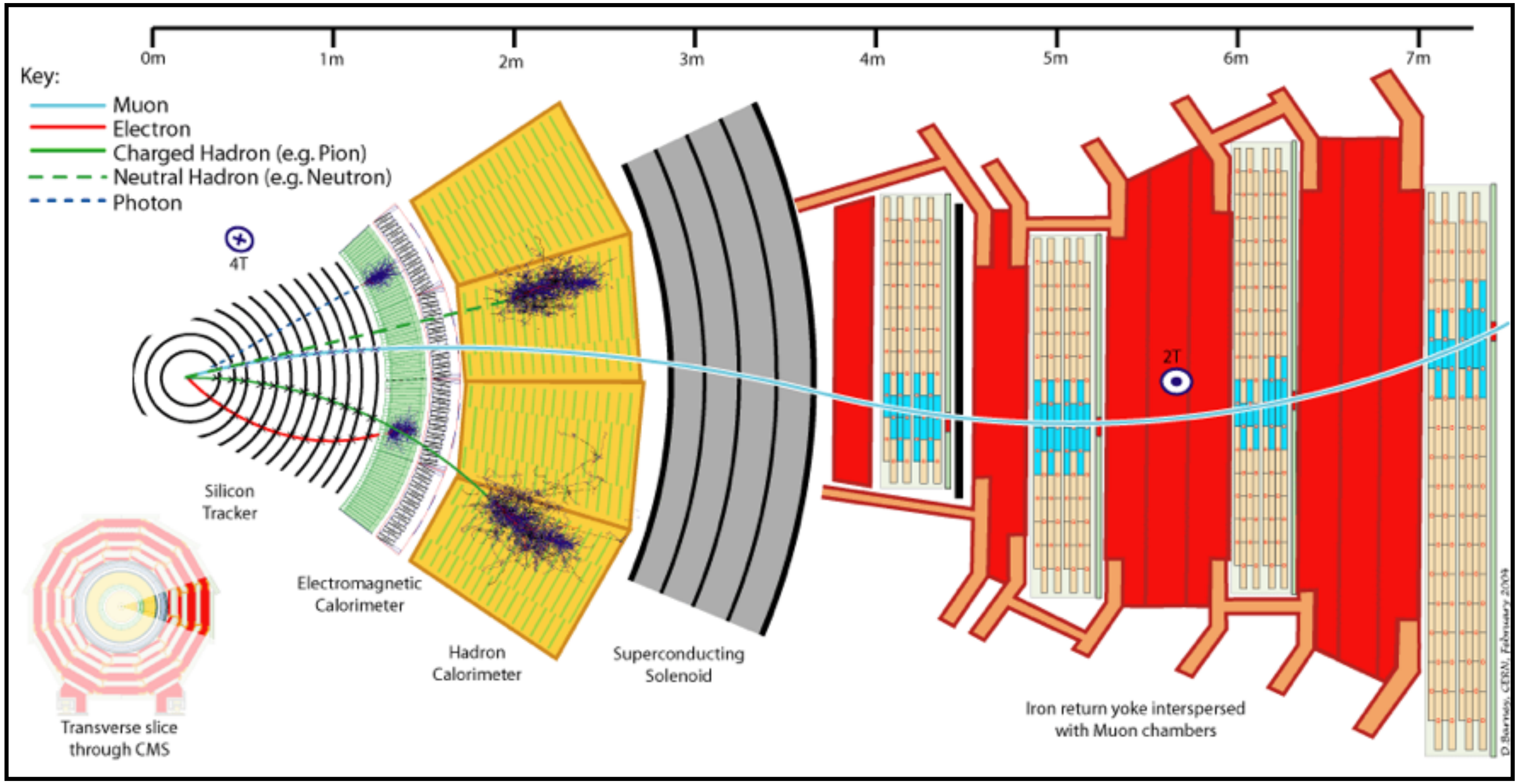
Vertex example in LHCb Velo.



End on projection.

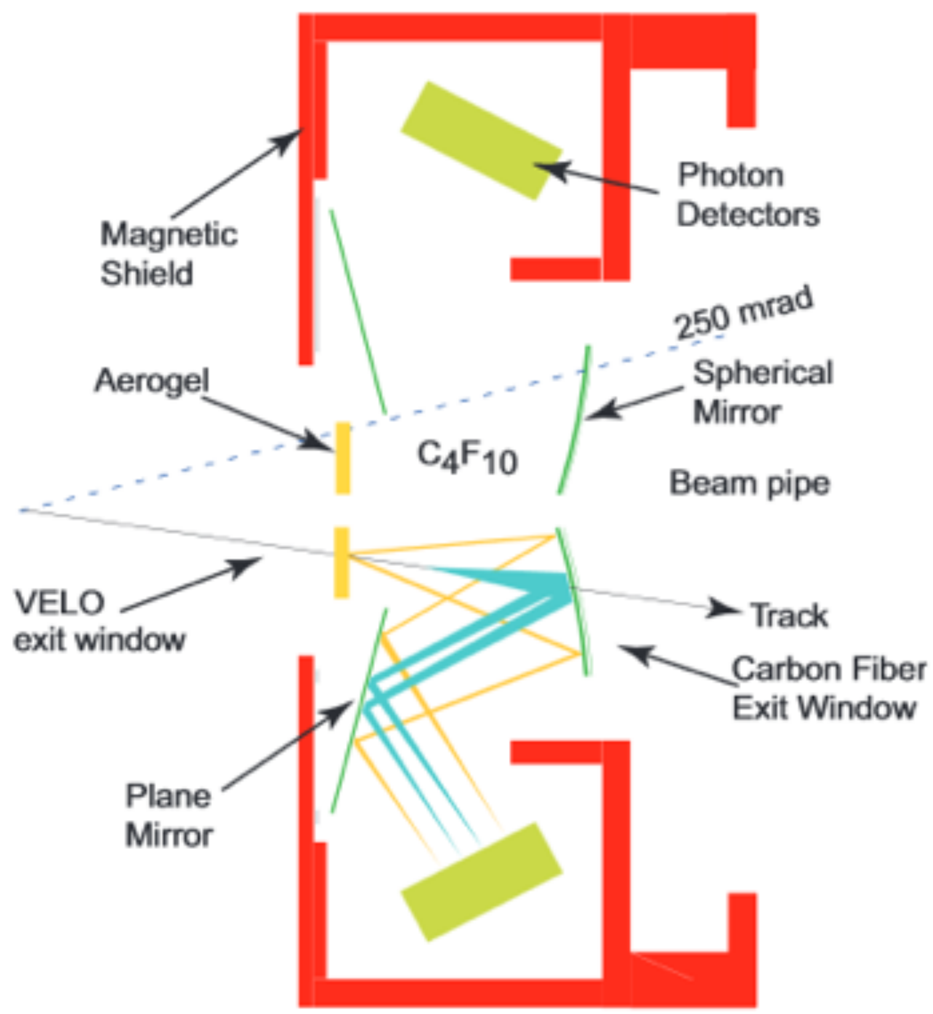
Particle ID (Briefly)

- Classify each track as a type of particle event by event:
  - Needed to refine selections for offline analysis (remove background).
- Many kinds of particle:
  - Not just fundamental particles, also composite hadrons (e.g. Pion, Kaon).
- Some easy cases:



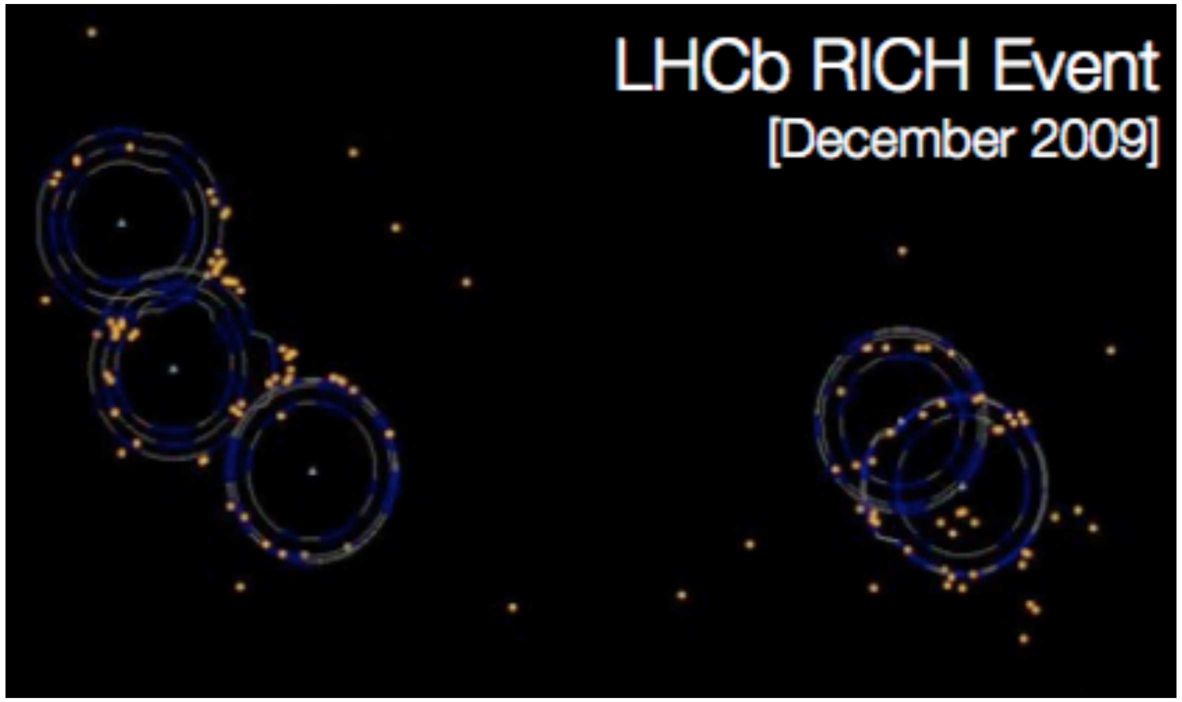
Particle ID (Briefly)

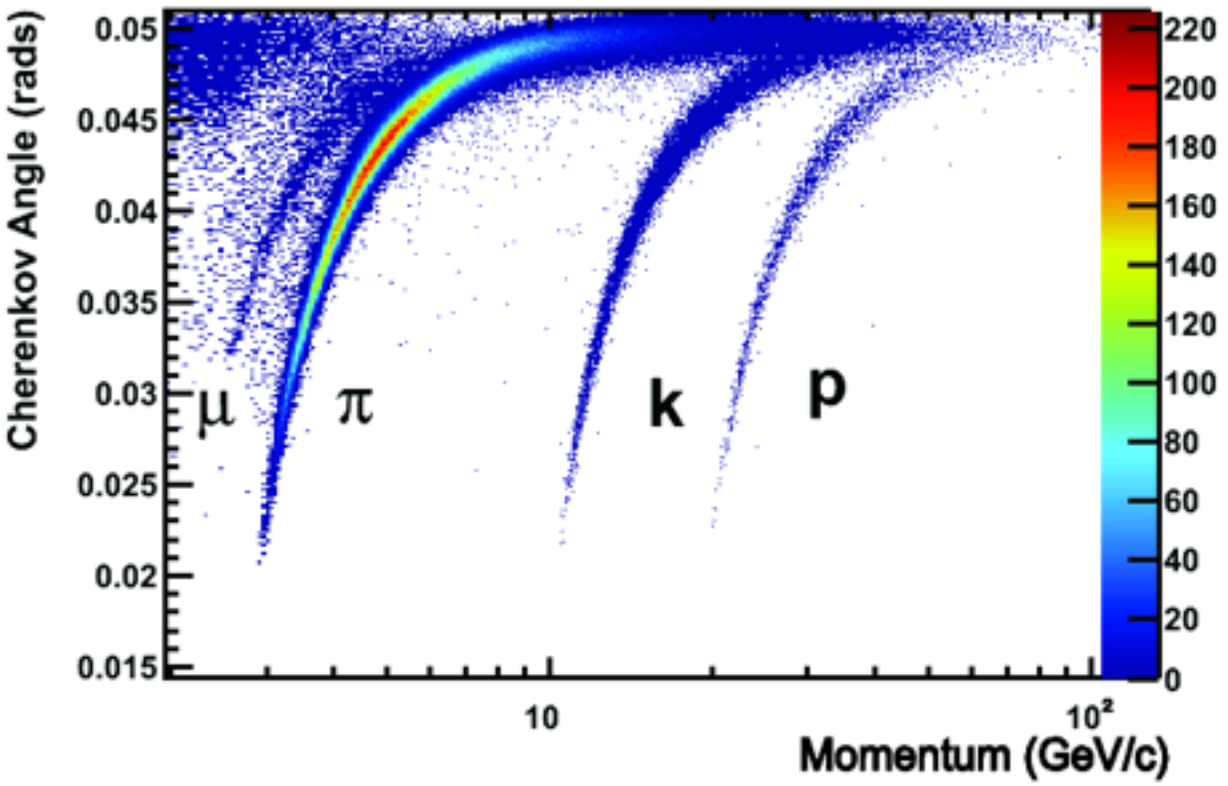
- Many techniques needed for other cases.
- RICH detector at LHCb uses Cherenkov radiation:
  - Light emitted when a particle slows passing through a material.
  - Emission is isotropic, and forms rings on detectors.
  - Often not required to reconstruct the ring itself - instead, test different hypothesis.
  - Sometimes multiple solutions (e.g. high momentum) - can still assign probabilities.



Particle ID (Briefly)

- Many techniques needed for other cases.
- RICH detector at LHCb uses Cherenkov radiation:
  - Light emitted when a particle slows passing through a material.
  - Emission is isotropic, and forms rings on detectors.
  - Often not required to reconstruct the ring itself - instead, test different hypothesis.
  - Sometimes multiple solutions (e.g. high momentum) - can still assign probabilities.

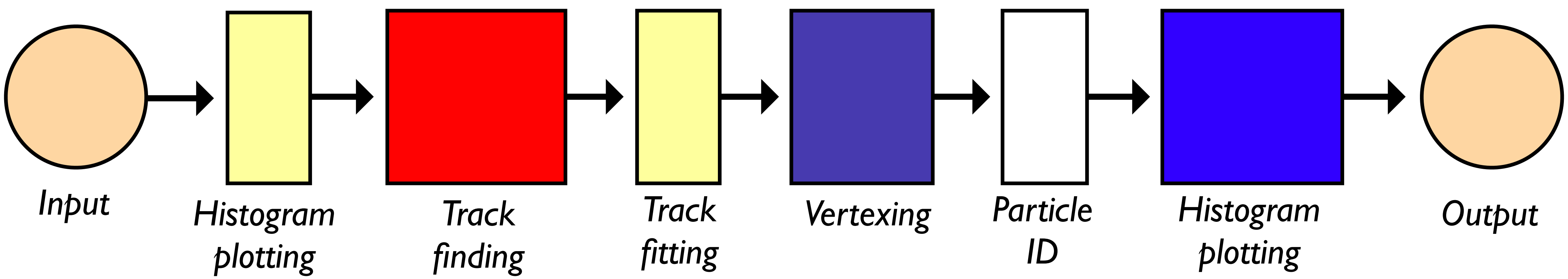




Event Reconstruction Implementation

Event Reconstruction Implementation

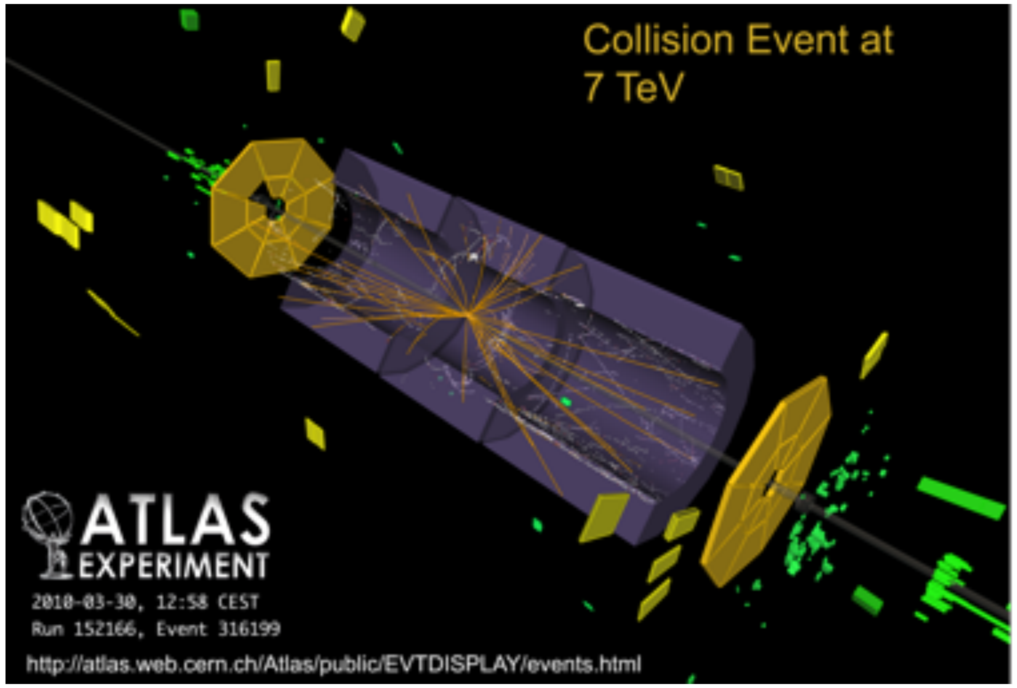
- Each reconstruction stage typically (sometimes by necessity) follows sequentially, e.g:

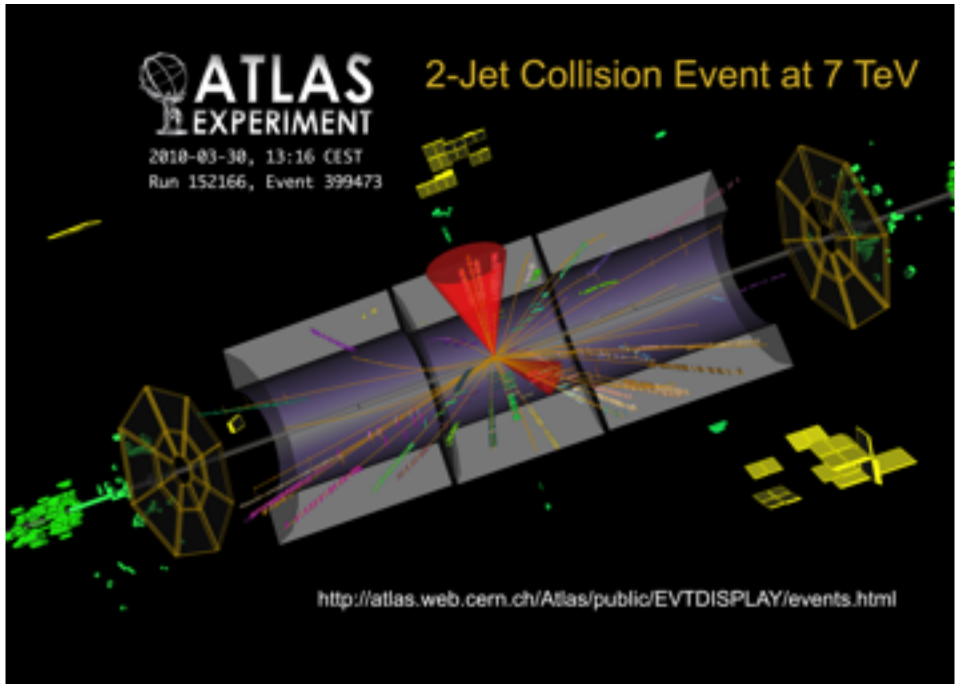


- Such a chain can be performed for a single event, or large set of events.
  - Reminder: each event is (usually) statistically independent of each-other.

- Strategy for single core is obvious, but for multi core, not so much.
- Nowadays, reconstruction involves tens of thousands of CPUs worldwide - need efficient strategy.
- Currently limited by memory:
  - E.g. CMS end of 2011 could only 6 out of 8 cores on average.

Event Reconstruction Optimisations





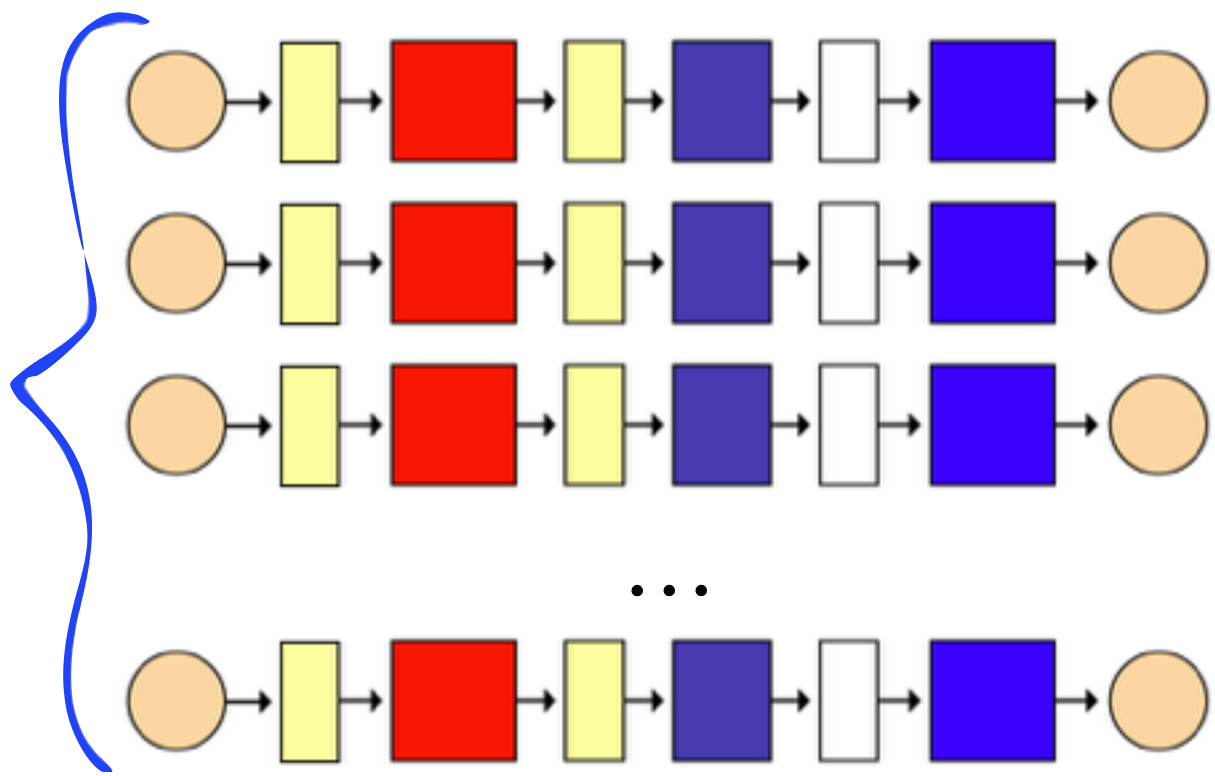
- Some advantages:
  - Data is essentially a list of collision data.
  - Each collision is (mostly) separate from the next.

- Some disadvantages:
  - Many packages needed, not all thread safe.
  - Large overhead of memory to run even a small job, can quickly reach O(1)GB:
    - (e.g. calibration and alignment values).
    - External libraries.

Event Reconstruction Optimisations - Parallelism

- Many possible optimisation strategies:
  - Parallelism - many possible options:

- Run multiple jobs on the same multi core machine.
- Very common strategy currently.
- Split input file (easy), and merge output file (easy).

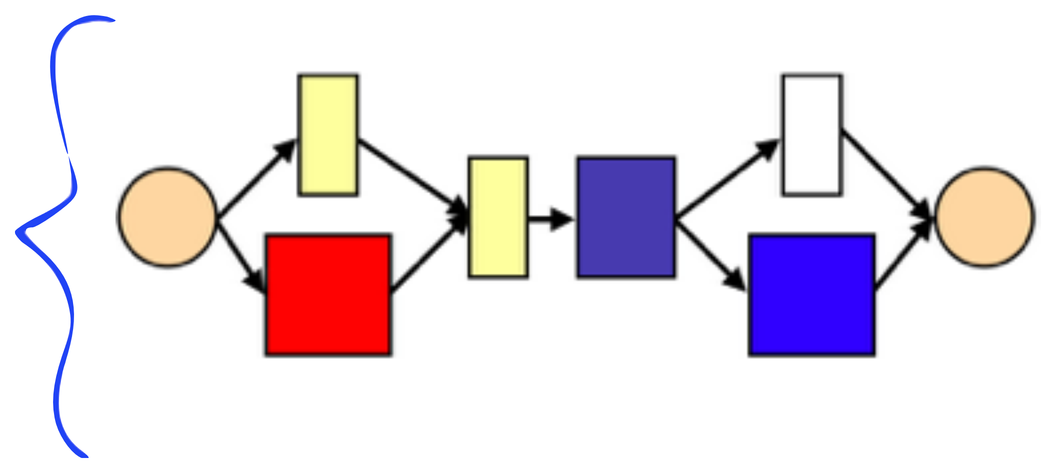


Potential pitfalls? Memory!

Event Reconstruction Optimisations - Parallelism

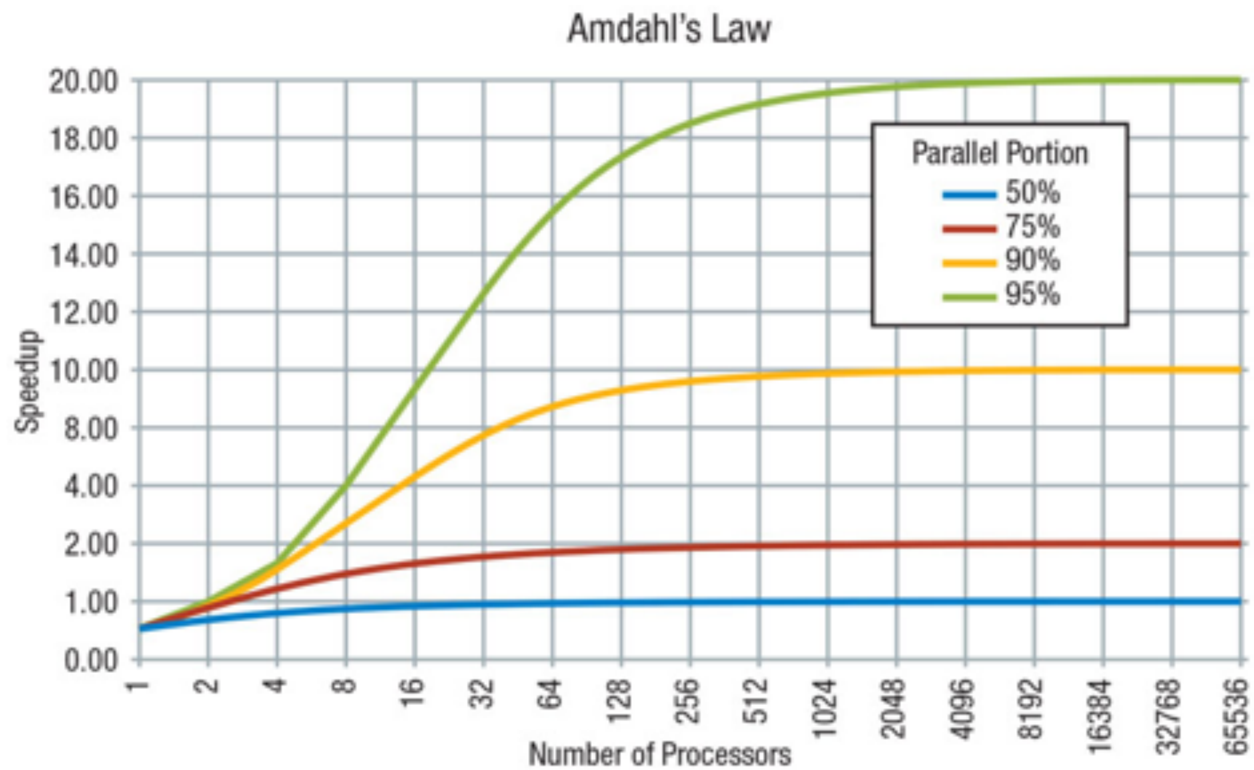
- Many possible optimisation strategies:
  - Parallelism - many possible options:

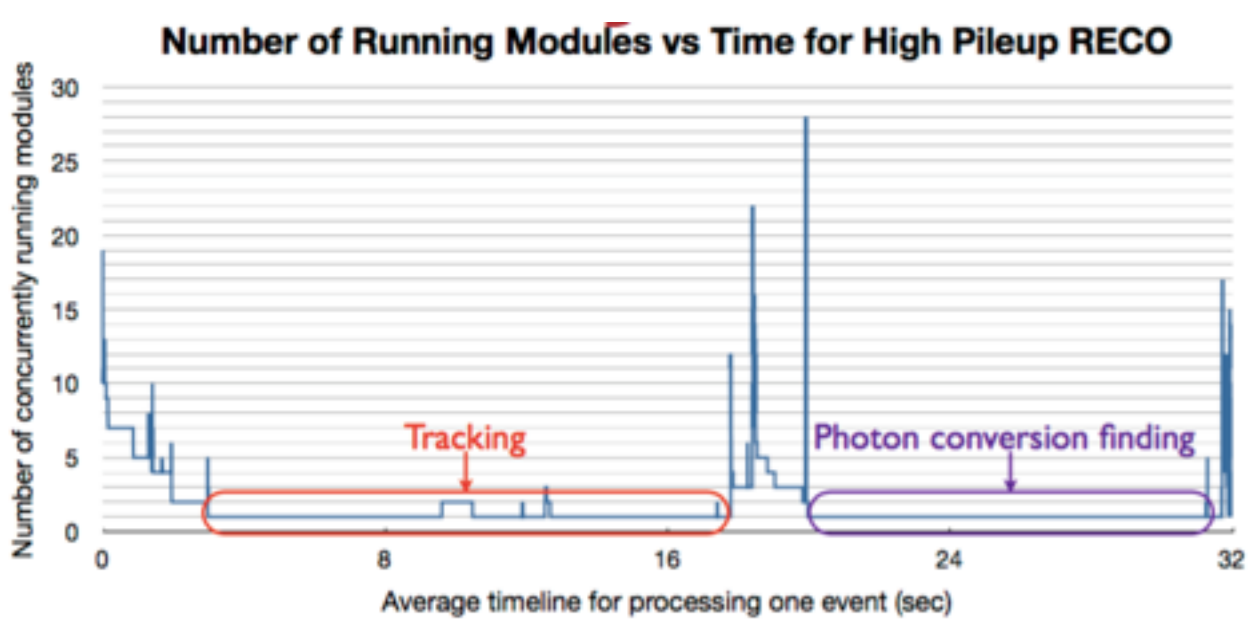
- Many segments can be performed in parallel.
- Many segments (e.g. tracking) depend on those previous, which can give long serial sections.



Potential pitfalls? Amdahl's Law! Typically limited at factor of 5[9] speedup.

Event Reconstruction Optimisations - Parallelism



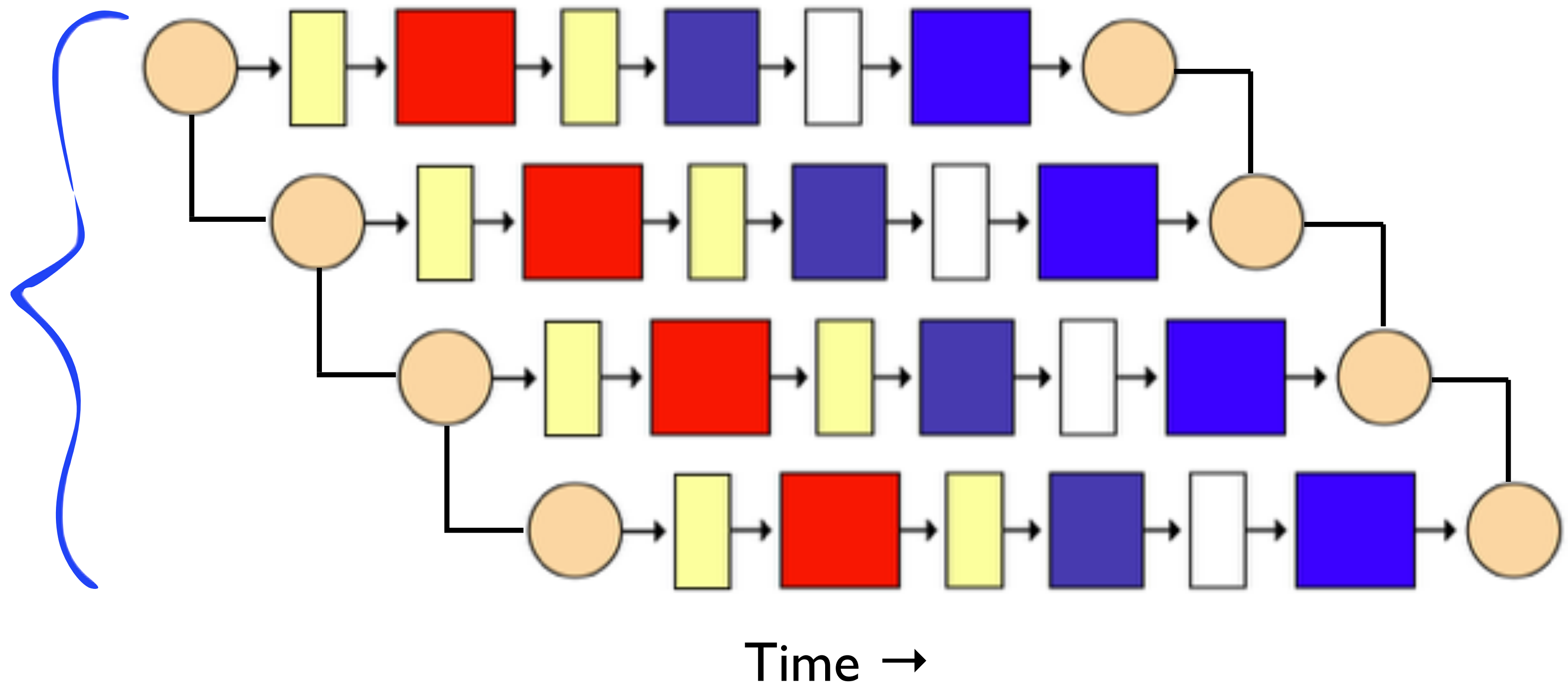


- Large segments of sequential code limit theoretical speedup:
  - E.g. tracking, expected to get harder.
  - Typically limited to factor of 5 speedup (~80% parallel portion).
- Successfully implemented at CMS (see later).

Event Reconstruction Optimisations - Parallelism

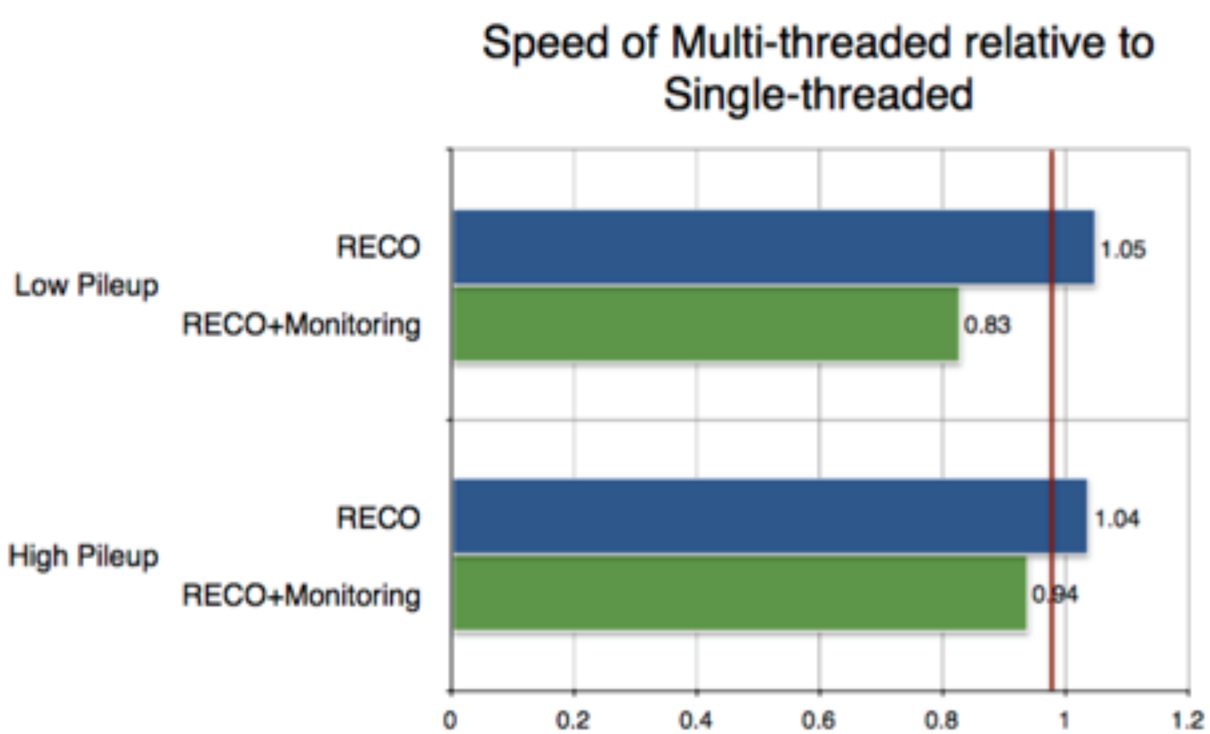
- Many possible optimisation strategies:
  - Parallelism - many possible options:

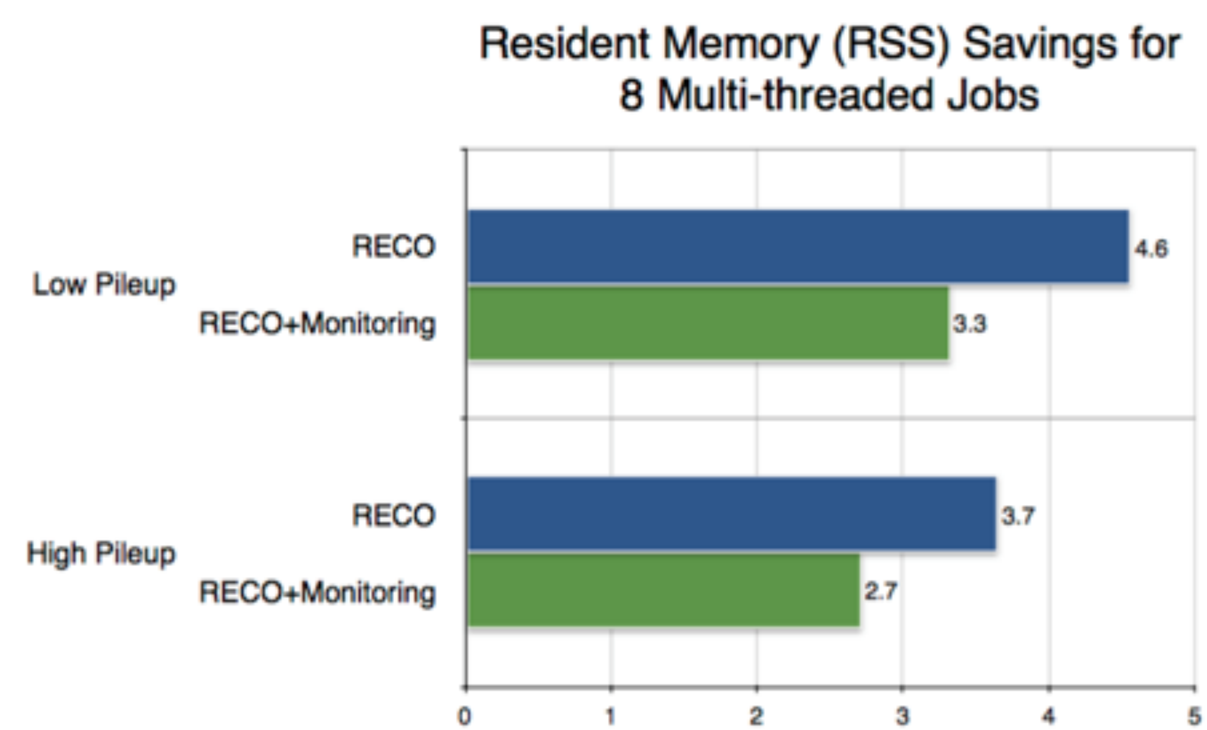
    - Process multiple events concurrently in one big job.
    - Large overhead of memory (e.g. calibration constants) can be **shared** between threads.



Event Reconstruction Optimisations - Parallelism

- Current limited by memory and not CPU power.
  - Manageable at run 1, but not for higher luminosity.
- Parallelism strategy aims for similar performance whilst saving memory.

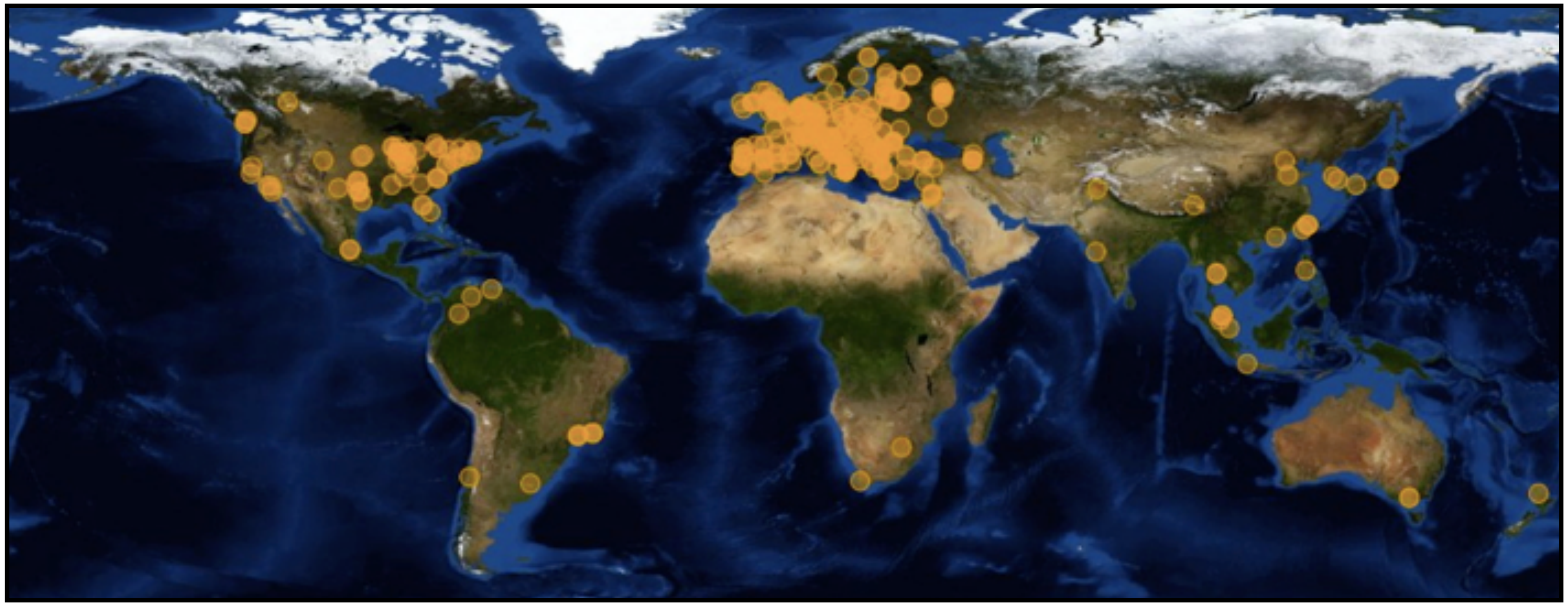




The Grid

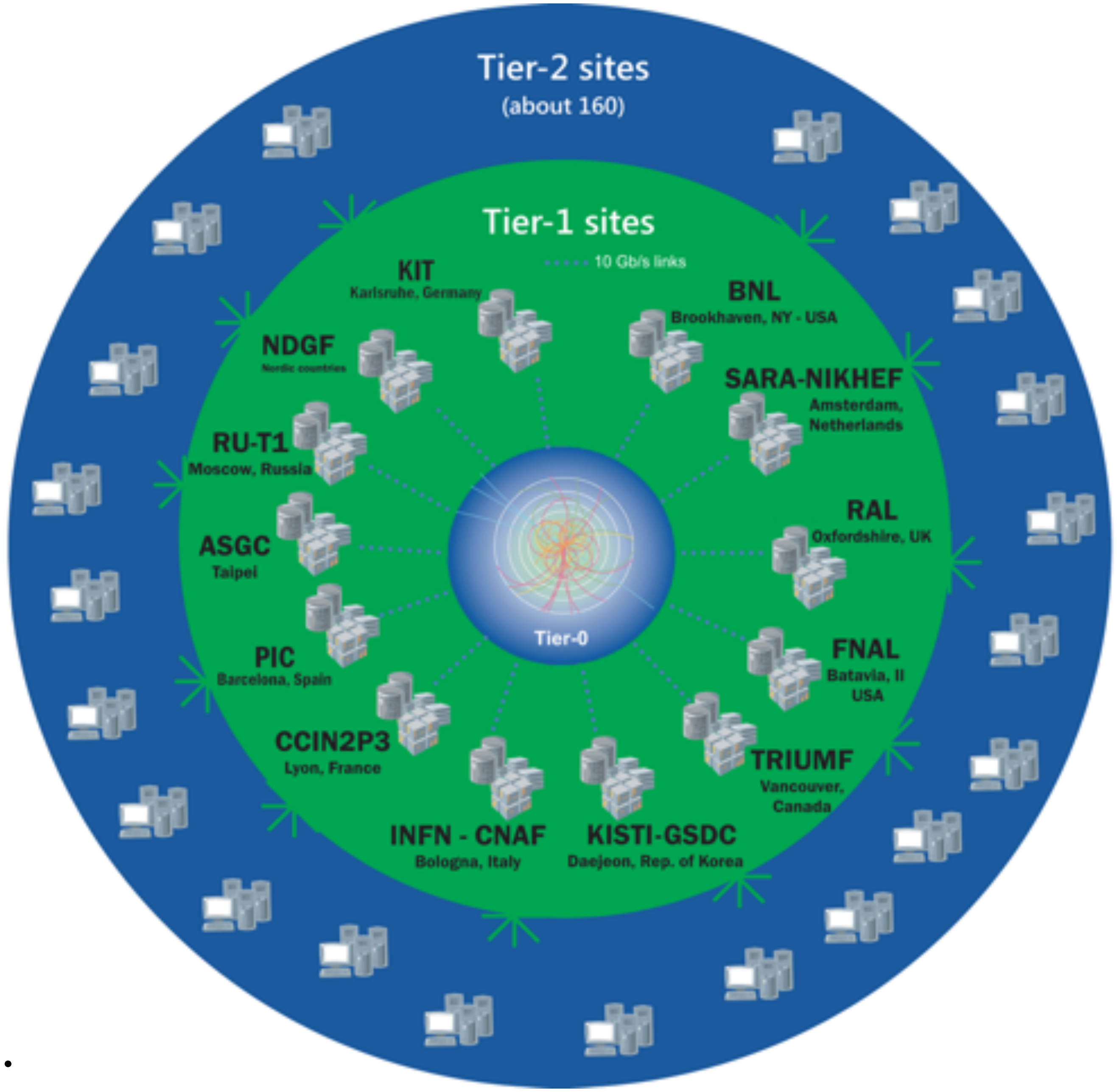
- Majority of reconstruction jobs performed on the Grid (excluding HLTs):
  - ~2 million jobs processed each day.
  - 170 computing centres in 42 different countries.

- Managed by each institute:
  - Large variation of technology and software used.



The Grid

- Centres are organised in a tier system.
- Tier 0 (CERN):
  - Receives data from experiments.
  - First copy of data to tape.
  - Distribute reconstruction in Tier1.
  - Re-reconsuction when LHC is not on.
- Tier 1 (distributed, 10GB/s connection):
  - Offline event reconstruction performed here (&storage).
  - Distribute jobs on Tier 2.
- Tier 2 (distributed):
  - University sites.
  - Ideal for analysis jobs performed on reconsulted data (e.g Higgs Searches).



Lecture 2 Summary

- Event reconstruction typically involves a set of algorithms executed in a particular order (often by necessity):
  - Each algorithm offers varying efficiency, purity and speed.
  - Tracking in particular often a large sequential portion of event reconstruction.

- Parallelism required to utilise available computing resources:
  - No obvious methods to parallelise.
  - Until recently, ok to have multiple processes on multi core machines:
    - Now memory issues - multi threading required to reduce large overhead.
    - In process of multi threading reconstruction packages.

- Many computing resources available, not just relying on CERN:
  - The Grid is used for processing many reconstruction and analysis jobs.
  - Spread worldwide, using a variety of different technologies.

- With flat computing budgets and increased luminosities, reconstruction is challenging:
  - Clever ideas welcome!

References

[1] Introduction to Physics Computing, CSC 2014.

[2] CERN computing website.

[3] CMS upgrade trigger TDR.

[4] LHCb trigger TDR.

[6] LHCb Starter Kit.

[7] Sascha Stahl, LHCb High Level Trigger, EPS-HEP 2015.

[8] ALICE HLT High Speed Tracking on GPU, IEEE Transactions on Nuclear Science, Vol. 58, No. 4, August 2011.

[9] Software Design in the Many-Cores era, CSC 2014.

[10] Study of a Fine Grained Threaded Framework Design, Christopher Jones,
http://indico.cern.ch/event/149557/session/3/contribution/194/attachments/150160/212729/Threaded_Framework_talk.pdf

[11] Using the CMS Threaded Application in a Production Environment, Christopher Jones, CHEP 2015,
http://indico.cern.ch/event/304944/session/2/contribution/120/attachments/578690/796858/Threading_Production_CHEP2015.pdf

+ many more…

[12] LHCb Topological Trigger Reoptimization, https://cdsweb.cern.ch/record/2019813/files/LHCb-PROC-2015-018.pdf