DyTER: A framework for Dynamic Track and Event Reconstruction for PANDA at FAIR

Michael Papenbrock

Department for Physics and Astronomy
Uppsala University

for the PANDA collaboration

October 18th, 2017
SFAIR/SFS-KF meeting
Stockholm, Sweden

DyTER - Dynamic Track and Event Reconstruction

Idea

- Focus on hyperons (displaced vertices)
- Break away from traditional event-based reconstruction
- Generate tracks and events dynamically from continuous data stream
- Use track and vertex information in event building

→ Track reconstruction and event building as an interdependent process

- Write highly modularised code

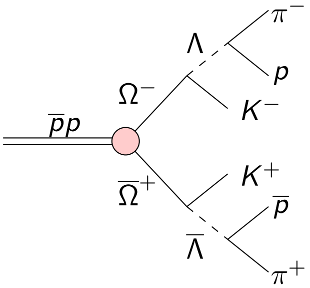
DyTER - Dynamic Track and Event Reconstruction

Our work so far

- Starting point: Track finder based on Cellular Automaton (J. Schumann, FZ Jülich)
- Implementation of longitudinal momentum reconstruction (W. Ikegami Andersson)
- Detailed investigation of detector signatures of hyperons to guide development (J. Regina)
- Prototype of a highly parallelised reconstruction scheme (B. Andersson, J. Nordström)
- Implementation of algorithms for complete time-based simulation/reconstruction chain (D. Steinschaden)
- Development of a pattern matching algorithm (M. Papenbrock)

Hyperons

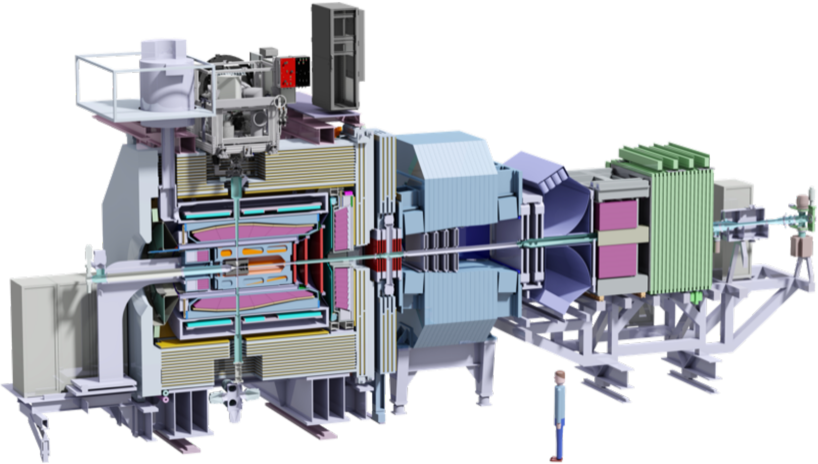
Example channel:
$\overline{p}p \rightarrow \overline{\Omega}\Omega \rightarrow \overline{\Lambda}K^+\Lambda K^- \rightarrow \overline{p}\pi^+K^+p\pi^-K^-$



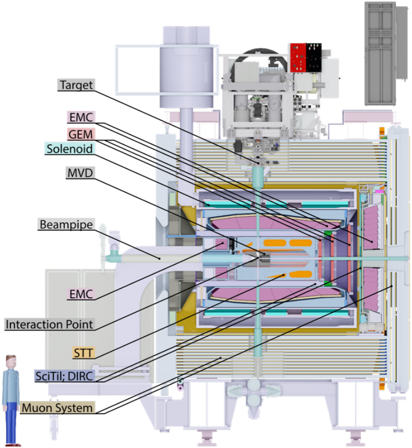
Why special attention?

- Complex topology
- Displaced vertices
- Intersecting tracks
- Different subdetectors for different tracks

The PANDA detector



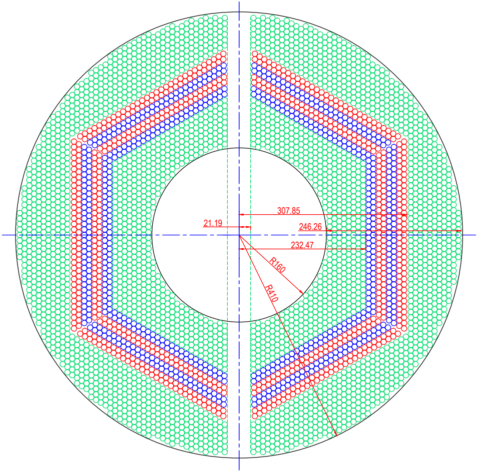
PANDA target spectrometer



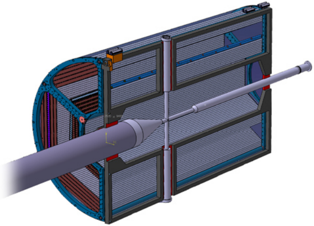
Charged track reconstruction

- Straw Tube Tracker (STT)
- Micro Vertex Detector (MVD)
- Gas Electron Multiplier (GEM)
- Scintillator Tile Hodoscope (SciTil / Barrel TOF)

Initial focus: Straw Tube Tracker (STT)

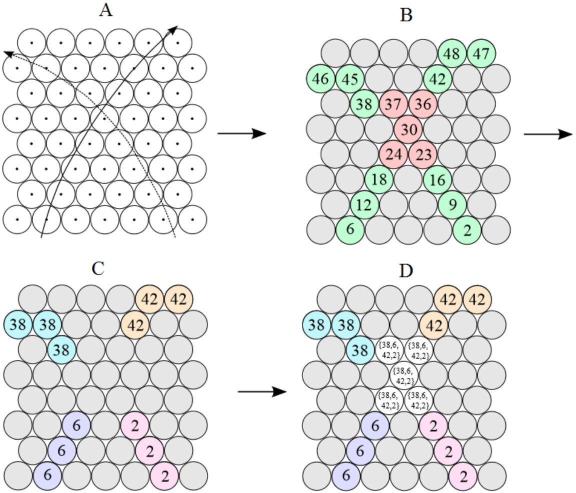


- 4224 straws
- 19 axial layers (green)
- 8 stereo layers (±3° blue/red)



SttCellTrackFinder

J. Schumann

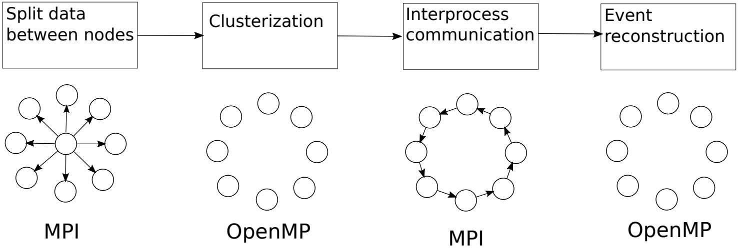


- Mark cell as active if it corresponds to hit
- Assign unique ID to unambiguous cells (i.e. ≤ 2 neighbours)
- Set ID of cells to minimum of itself and neighbours
- Ambiguous cells: Include all IDs of neighbours

Prototype of an online reconstruction scheme

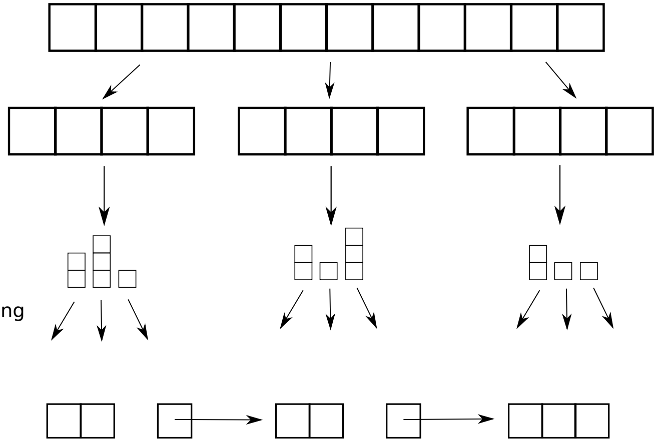
B. Andersson and J. Nordström

- Framework using two models for parallelisation
  - MPI: Distribution of workload on several nodes
  - OpenMP: Local parallelisation of clusterisation on multi-core CPUs



Prototype of an online reconstruction scheme

B. Andersson and J. Nordström



Data stream splitting (MPI)

Time-based bin stacking (seq.)

Parallel clustering (OpenMP)

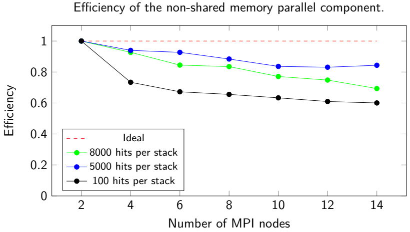
Merging of edge clusters (MPI)

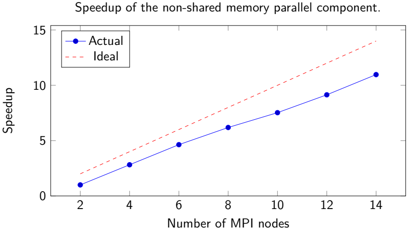
Prototype of an online reconstruction scheme

B. Andersson and J. Nordström

Performance analysis: MPI

- Non-shared memory environment
- Fixed problem size: 250000 STT hits
- Good scaling with number of nodes





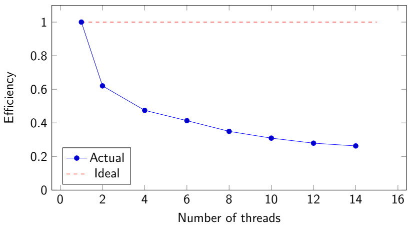
Prototype of an online reconstruction scheme

B. Andersson and J. Nordström

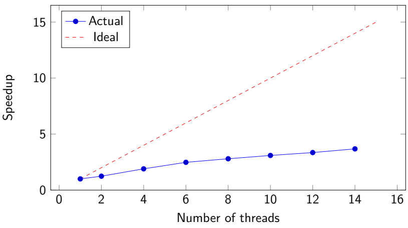
Performance analysis: OpenMP

- Shared memory environment
- Fixed problem size: 250000 STT hits; stack size: 5000 hits
- Slight speedup with increasing number of CPU cores

Efficiency of the shared memory parallel components.



Speedup of the shared memory parallel components.

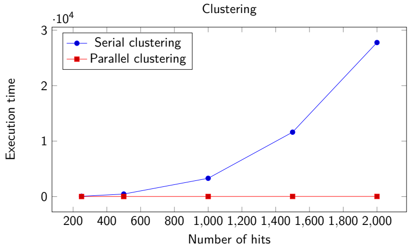


Prototype of an online reconstruction scheme

B. Andersson and J. Nordström

Performance analysis: Parallel hit clustering

- Shared memory environment, 16 CPU cores
- Dynamic problem size
- Parallel clustering algorithm incorporating hit time stamps
- Substantial gain with increasing problem size



Pattern Matcher: Concept

Ideas

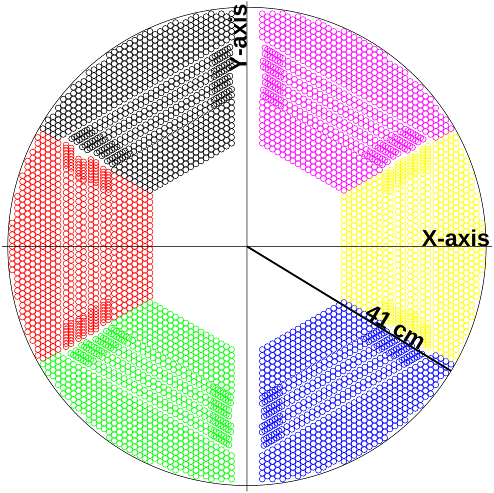
- Pre-clustering (procedure suitable for FPGAs)
- Augment SttCellTrackFinder with pattern matching algorithms or vice versa
- Stand-alone track finder using machine learning

- Divide STT into 6 sectors
- Simulate desired channel (here: $\Lambda\bar{\Lambda}$)
- Store pattern as set of tube IDs
- Determine and store complementary information
- Merge duplicate/similar patterns
- Start matching

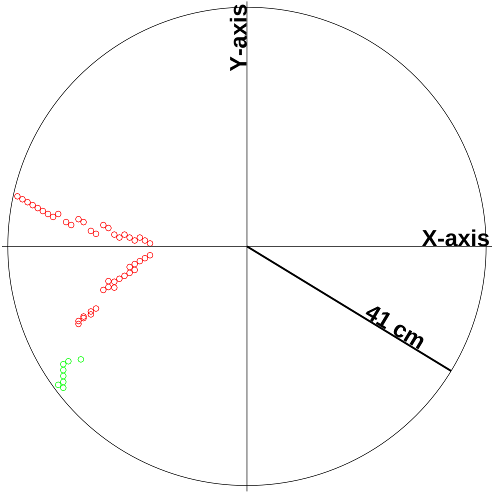
Closer look: Pattern

- tubeIDs
- momenta
- timeStamps
- sectorID
- count

Pattern Matcher: Concept



Pattern Matcher: Concept



Pattern Matcher: Database Generator

- Generate events for desired channel (use ideal track finder)
- Identify patterns as tubeIDs for hits corresponding to a track
- Extract complementary information (e.g. momentum, sectorID, etc.)
- Store items in database

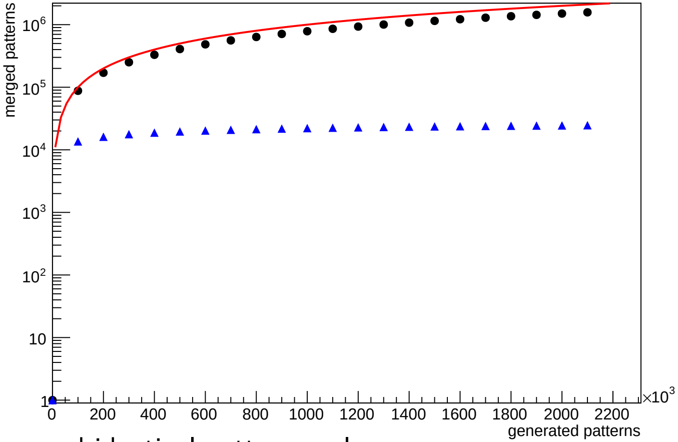
Attention

- Database will be filled with duplicate patterns!

→ Identify and merge identical patterns

→ Bonus: Identify and count "similar" patterns (e.g. 90 % match)

Pattern Matcher: Merging



black: merged identical patterns only

blue: merged 90% similar patterns

- similar patterns saturate $\lesssim$ 100000 ($<$ 100 MB)

Summary & Outlook

Summary

- Highly parallelised framework for reconstruction algorithms
- Further development of track finder based on cellular automaton
- First prototype of pattern matching algortihm

Outlook

- Develop shared memory parallelisation based on FairMQ
- Implement time-based processing for track finders
- Explore machine learning possibilities

Thank you for your attention!