CONTINUOUS ONLINE TRACKING FRAMEWORK

Sean Dobbs

(Northwestern U.)

M. Mertens, J. Ritman, P. Wintz

(FZ Jülich)

GSI, 11 Dec. 2012

Motivation

- PANDA is planned to run with a quasi-continuous beam and triggerless readout at high rate (up to 2×10^7 events/s)
- "Interesting" events are many orders of magnitude more rare than the "uninteresting" events, but often have similar topologies.
- Software trigger needs well reconstructed tracks
- This creates a demanding situation for online tracking
  - Needs to be robust against low-p_T tracks, displaced vertices, decays-in-flight, etc.
  - Need to determine which particles come from which event.

Setting the Scale of the Problem

- LHCb
  - pp collisions, $\sqrt{s} = 7(8)$ TeV @ 20 MHz
  - Hardware L0 triggers on muons, calorimeter energy, reduces event rate to 1 MHz (limited by FEE, upgrade plans to read out at full rate)
  - HLT runs offline algorithms (or slightly simplified versions) on >15000 processors (>25000 instances), reducing event rate to 3 kHz

How does PANDA compare with LHCb?

- PANDA: $p\bar{p}$ collisions, $\sqrt{s} = 2.5 - 5$ GeV
  - Event rate from FEE is an order of magnitude higher
    - PANDA: 20 MHz, LHCb 1 MHz
    - Individual channels in PANDA have < 1 MHz
  - Event complexity is an order of magnitude lower
    - Average number of tracks/event: PANDA ~5, LHCb ~70
- Data rate for both experiments is comparable
- PANDA has more complicated geometry
  - Target spectrometer in addition to forward spectrometer
- No a priori knowledge of event timing
- Comparable online processing resources

Continuous Online Tracking

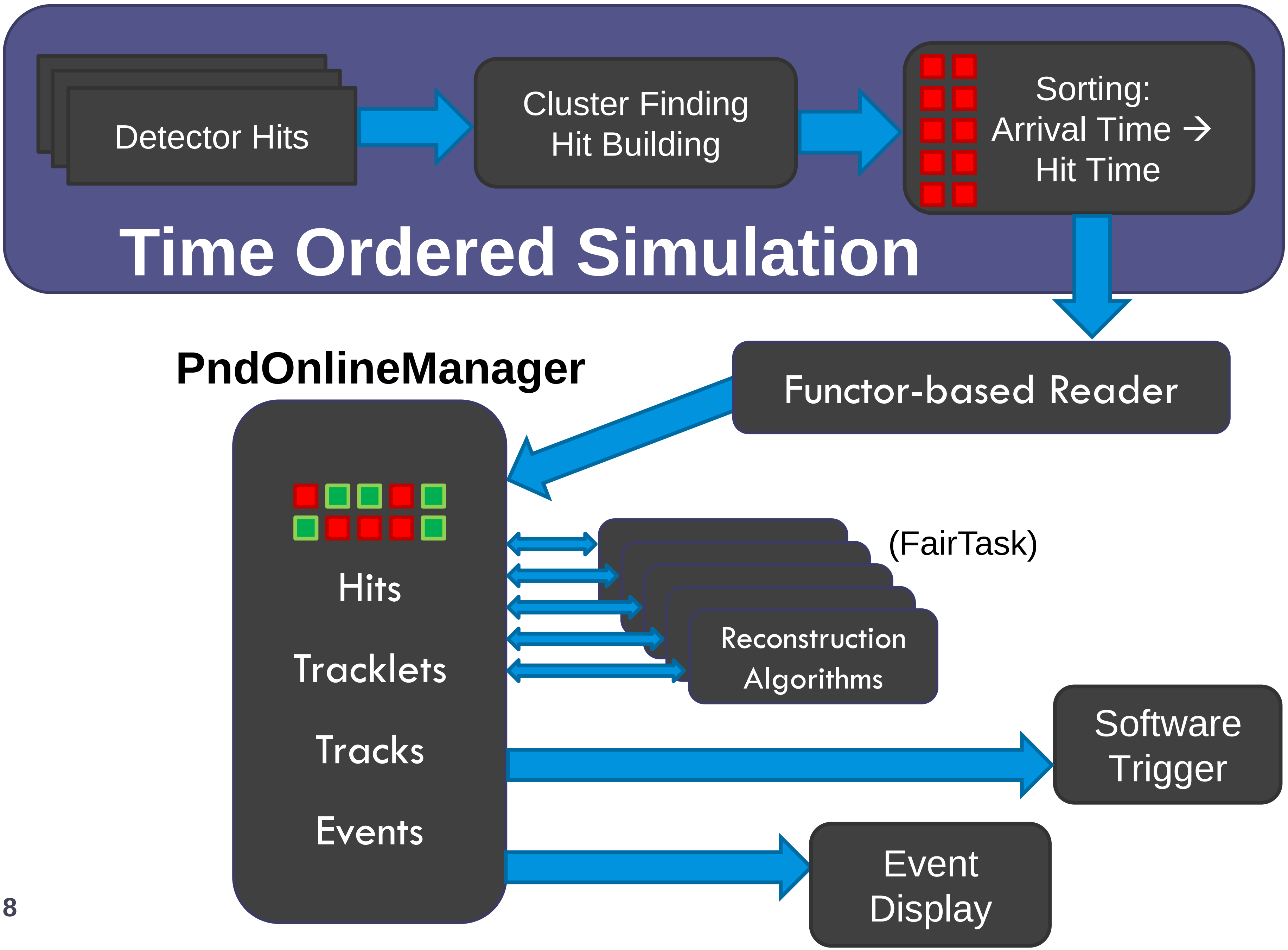
- The constraints on online tracking are:
  - Triggerless readout
  - High event rate
  - Continuous beam
  - Different track topologies
- Various types of tracks must be reconstructed on a non-event-based, “continuous”, basis
- Algorithms should be selected which maximize speed and reconstruction efficiency while using a reasonable amount of computing resources.

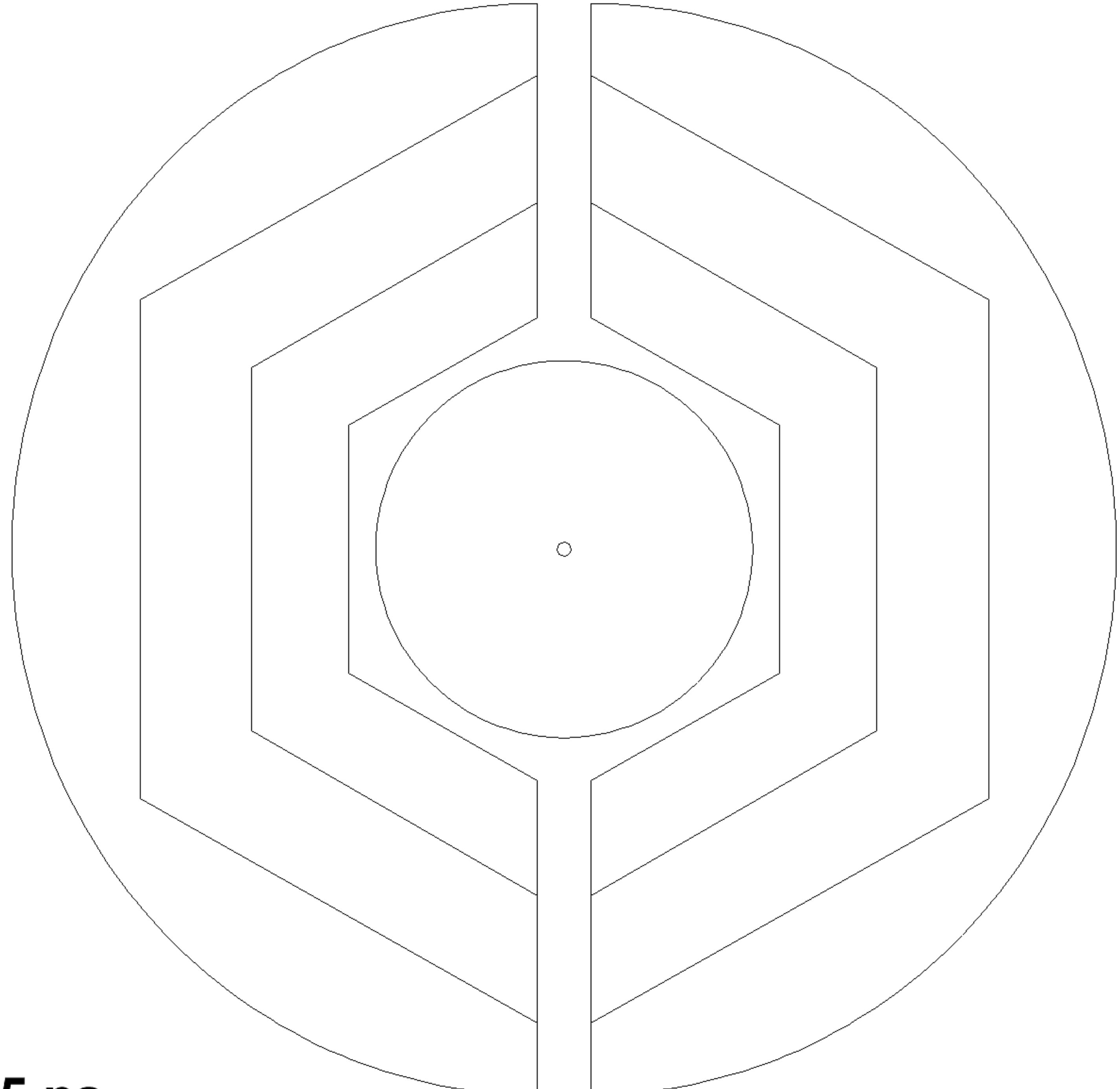
Continuous Online Tracking Framework

- Algorithms must be tested with time-based simulations and benchmarked against key physics channels
- To this end, we have developed a prototype framework for running and evaluating tracking algorithms
  - Tests continuous tracking data flow
  - Development has focused on STT & MVD detectors

Framework for Algorithm Development

- Reads in hits from time-based simulation
- Runs series of algorithms and keeps running track of results
  - Hits → Tracks → Events
  - Standard classes are wrapped or extended to store online-specific information, e.g., t_0
- FairRunAna handles ROOT I/O, geometry
  - Most detectors do local clustering, simple geometry needed for straw tubes due to long drift times
- Simple event display to facilitate development
- Example: Triplet finder


Detector Hits
Cluster Finding
Hit Building
Sorting:
Arrival Time → Hit Time
Time Ordered Simulation
PndOnlineManager
Functor-based Reader
Hits
Tracklets
Tracks
Events
(FairTask)
Reconstruction Algorithms
Software Trigger
Event Display



5 ns

Summary

- PANDA online tracking needs to reconstruct a variety of different track topologies in a demanding environment.
- Algorithms must be tested with realistic, time-based simulations
- A framework for running and evaluating these algorithms is under development
  - Short term: Standardize and make available in SVN
  - Future: Integration with other infrastructure (Event Dispatcher), execution on GPUs & Compute Nodes
  - Other detectors easily integrated (forward tracking?)
- We look forward to contributions from many others!