Mitglied der Helmholtz-Gemeinschaft

Thoughts on Online Tracking/Triggering

Darmstadt, 5.3. 2012 Marius C. Mertens

Aims and Parameters

- Inbound event rate: $2 \cdot 10^7$/s
- Detector data does not arrive event sorted
  → Processing of all detector data required to build events
  - *Must be done quickly in a large buffer/pipeline*
  - *Data here is only temporary*
- After event building, still temporary „storage“
  - *Filter rules to decide which data goes to persistant storage*
  - *Filtering must be quick and efficient*
- Filter properties
  - *Simplicity of algorithm*
  - *Efficiency: Discarded good events, false positives*
  - ***Event reduction factor***
- Two ways to address this question:
  - *How many events can we write out?*
  - *How much reduction can physics afford?*

One year earlier . . .

„Traditional“ Approach

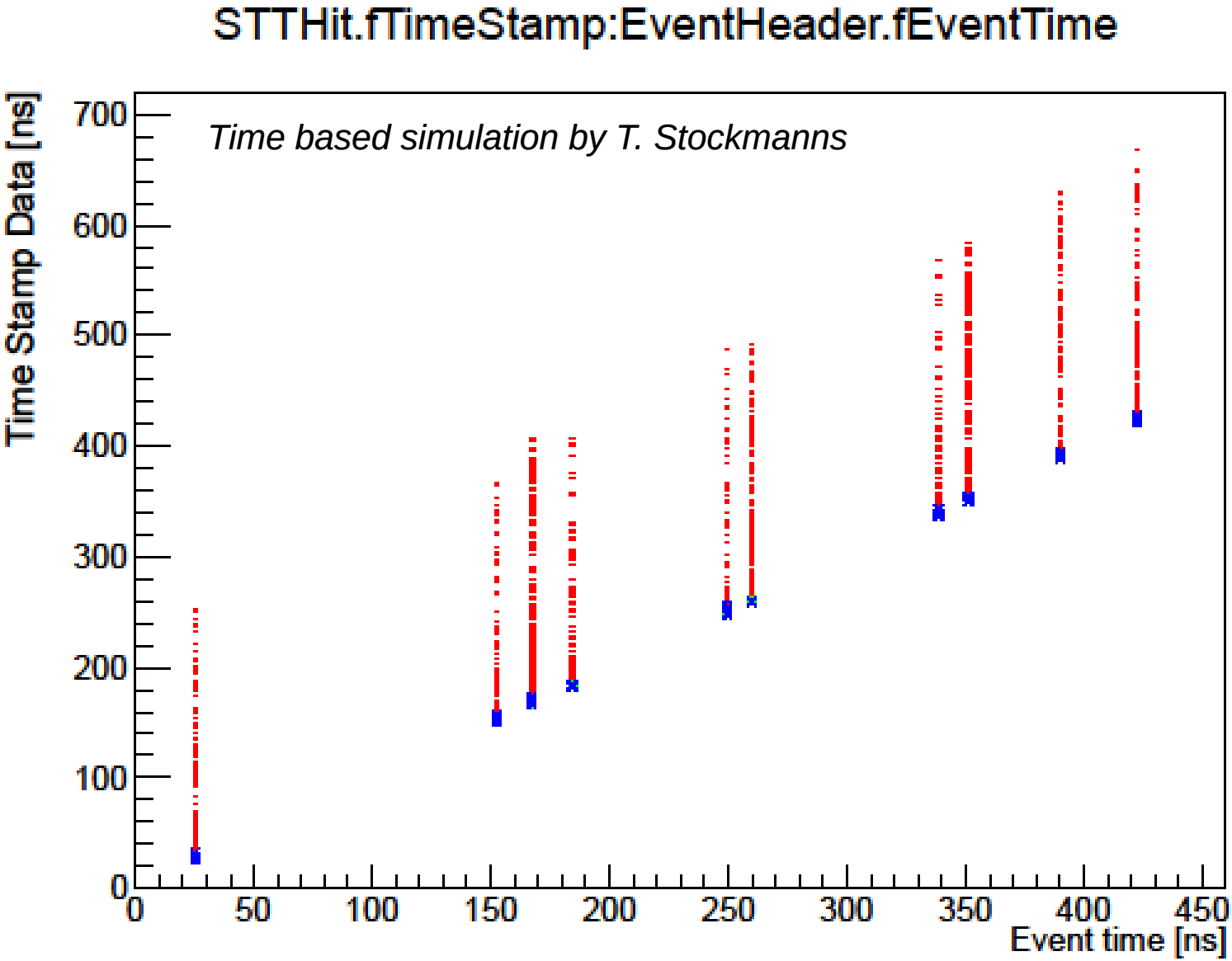
- Search for a characteristic (simple) pattern
  - *Hit count in detector X*
  - *Tracklet in certain area*
  - *Energy deposition in calorimeter*
  - *…*
- Once found process the surrounding block of data

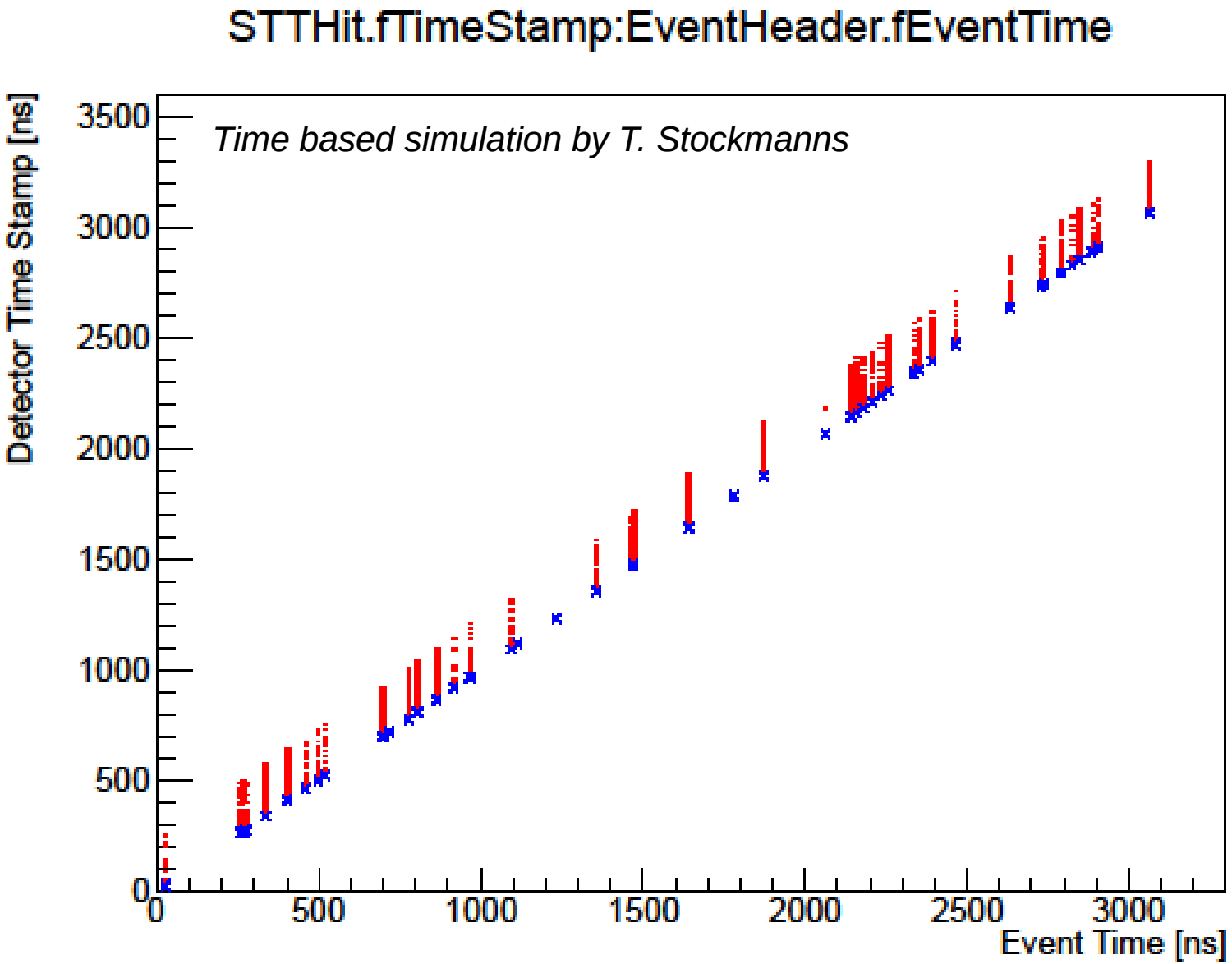
- Challenges of PANDA
  - ***Large overlaps, quasi-continuous operation***
    → *Makes pattern recognition more difficult*
    → *Requires computing power for sorting*
    → *May lead to redundant work in stateless pattern reco*

Panda Processing Stages

- Detector raw data collection
- Cluster finding, hit building
- Track finding
- Track feature extraction: Momentum, mass, …
- Event building

- Filter decision
- Write to disk


STTHit.fTimeStamp:EventHeader.fEventTime
Time based simulation by T. Stockmanns
Time Stamp Data [ns]
Event time [ns]
0
100
200
300
400
500
600
700
0
50
100
150
200
250
300
350
400
450


STTHit.fTimeStamp:EventHeader.fEventTime
Time based simulation by T. Stockmanns
Detector Time Stamp [ns]
Event Time [ns]
0
500
1000
1500
2000
2500
3000
3500

Possible(?) PANDA Procedure

- Continuous tracking to remove hits otherwise found in later events
- Take information from fast detectors as t0 seeds
  → Fit results can indicate if track belongs to „this“ t0
- Remove uniquely identified hits (tracks) from further processing
- Search for simple patterns first, then complex (different algorithms!)
  → Later processing stages operate on less hits
- Reset at gaps (overlap free)
  or after certain times (with overlap penalty)
- Process data in a pipelined manner
- Distribute blocks of data to different compute nodes
  → Makes use of highly parallel architecture
- **Needs to be checked whether this is actually feasible**

Possible(?) PANDA Procedure