

# A PandaRoot Interface for Binary Data in the PANDA Prototype DAQ System

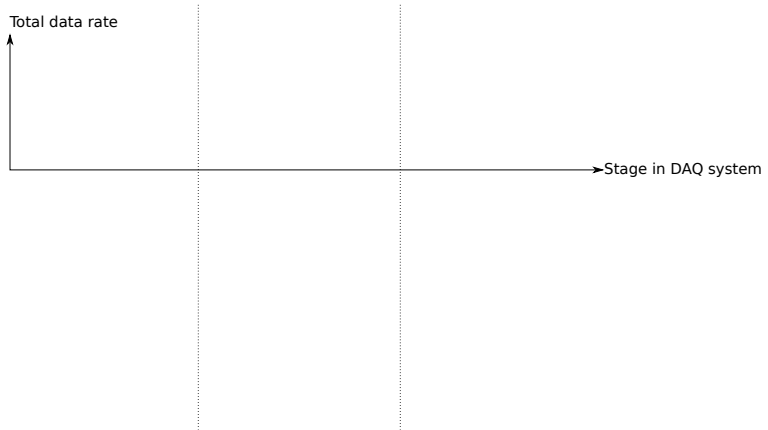
Sören Fleischer, Christopher Hahn, Milan Wagner, Sören Lange

December 09, 2014

# Inhaltsverzeichnis

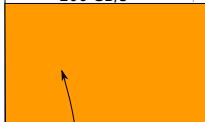
- 1 Motivation
  - Data Flow through the DAQ
- 2 Implementation
  - Implementation Scheme
  - Data Format
  - State Machine
  - Details/Summary
  - Outlook
- 3 Thank you

# Motivation



# Motivation

Total data rate  
200 GB/s



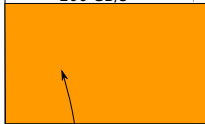
Detector front ends



→ Stage in DAQ system

# Motivation

Total data rate  
200 GB/s



Detector front ends



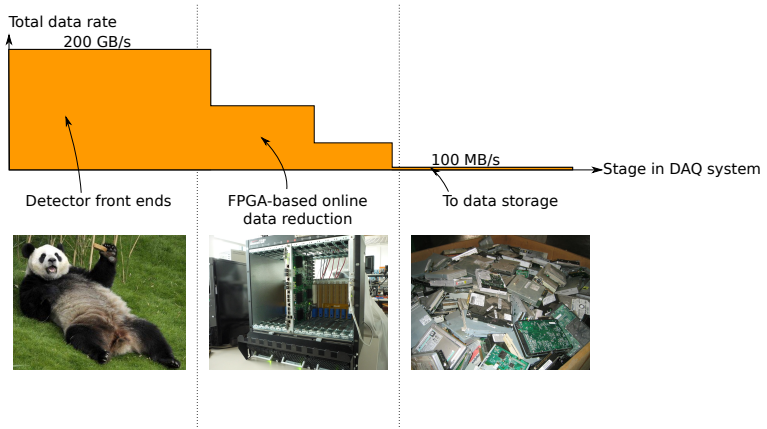
100 MB/s

→ Stage in DAQ system

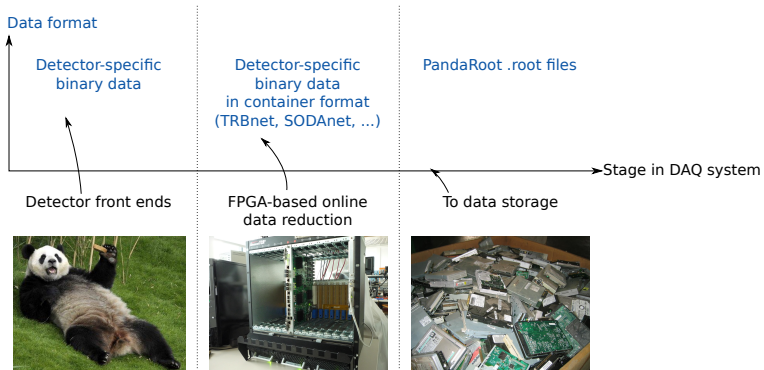
To data storage



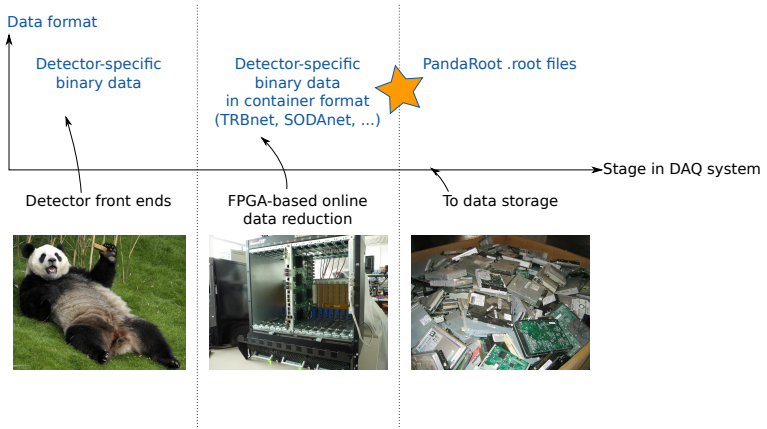
# Motivation



# Motivation



# Motivation



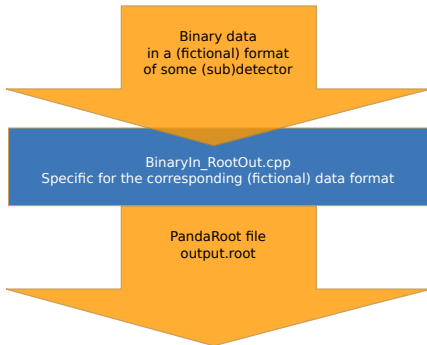


# Implementation Scheme

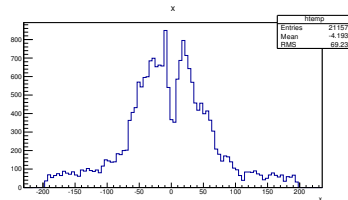
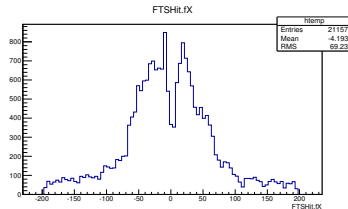
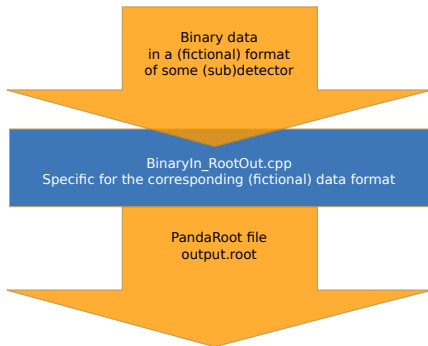
BinaryIn\_RootOut.cpp

Specific for the corresponding (fictional) data format

# Implementation Scheme



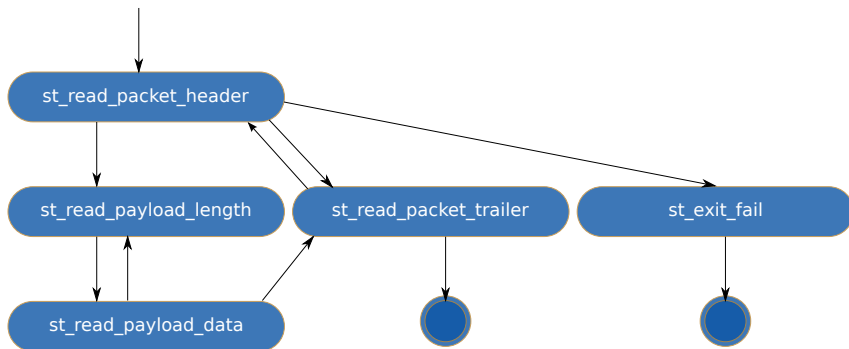
# Implementation Scheme



# Data Format



# State Machine



## Implementation Details / Summary

- State-machine-based binary parser with PandaRoot output
- Implemented in PandaRoot/C++

## Implementation Details / Summary

- State-machine-based binary parser with PandaRoot output
- Implemented in PandaRoot/C++
- Start using `$ root -l BinaryIn_RootOut.cpp`

## Implementation Details / Summary

- State-machine-based binary parser with PandaRoot output
- Implemented in PandaRoot/C++
- Start using `$ root -l BinaryIn_RootOut.cpp`
- $\approx$  300 LOC



## Implementation Details / Summary

- State-machine-based binary parser with PandaRoot output
- Implemented in PandaRoot/C++
- Start using `$ root -l BinaryIn_RootOut.cpp`
- $\approx 300$  LOC
- Successfully tested against input files with  $\approx 20k$  records

## Implementation Details / Summary

- State-machine-based binary parser with PandaRoot output
- Implemented in PandaRoot/C++
- Start using `$ root -l BinaryIn_RootOut.cpp`
- $\approx 300$  LOC
- Successfully tested against input files with  $\approx 20k$  records
- Tiny deviations between input and output data are the result of "lossy" encoding of the binary input files, where some LSB digits may be cut off and are unrelated to this implementation.

## Implementation Details / Summary

- State-machine-based binary parser with PandaRoot output
- Implemented in PandaRoot/C++
- Start using `$ root -l BinaryIn_RootOut.cpp`
- $\approx 300$  LOC
- Successfully tested against input files with  $\approx 20k$  records
- Tiny deviations between input and output data are the result of "lossy" encoding of the binary input files, where some LSB digits may be cut off and are unrelated to this implementation.
- Very inefficient in terms of IO (needs input buffer)

## Implementation Details / Summary

- State-machine-based binary parser with PandaRoot output
- Implemented in PandaRoot/C++
- Start using `$ root -l BinaryIn_RootOut.cpp`
- $\approx 300$  LOC
- Successfully tested against input files with  $\approx 20k$  records
- Tiny deviations between input and output data are the result of "lossy" encoding of the binary input files, where some LSB digits may be cut off and are unrelated to this implementation.
- Very inefficient in terms of IO (needs input buffer)

## Implementation Details / Summary

- State-machine-based binary parser with PandaRoot output
- Implemented in PandaRoot/C++
- Start using `$ root -l BinaryIn_RootOut.cpp`
- $\approx 300$  LOC
- Successfully tested against input files with  $\approx 20k$  records
- Tiny deviations between input and output data are the result of "lossy" encoding of the binary input files, where some LSB digits may be cut off and are unrelated to this implementation.
- Very inefficient in terms of IO (needs input buffer)

# Outlook

- Adaption to different (sub)detector and/or container data formats
- Speed improvement (needs input buffer)

# Outlook

- Adaption to different (sub)detector and/or container data formats
- Speed improvement (needs input buffer)
- Verification again

# Outlook

- Adaption to different (sub)detector and/or container data formats
- Speed improvement (needs input buffer)
- Verification again



# Outlook

- Adaption to different (sub)detector and/or container data formats
- Speed improvement (needs input buffer)
- Verification again

Thank you for your attention