Status of Raw Data

R. Brun, C. Cheshkov, M. Mager,
L. Musa, A. Ur Rehman

22/06/2009
ALICE Offline Week

Contents

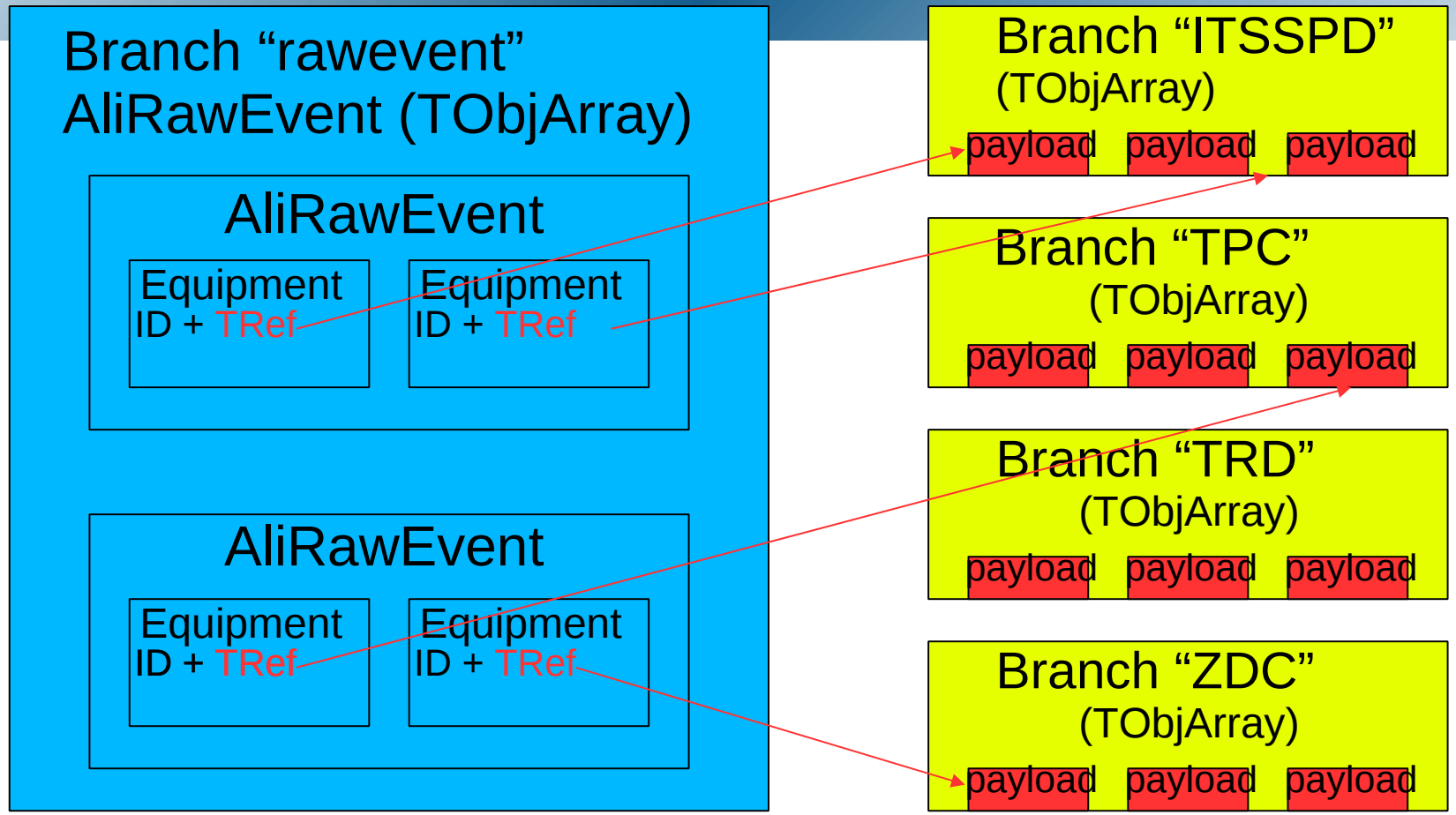
- New format of ROOT raw data
- Raw-data compression
- New RCU ALTRO format
- Conclusions

New Raw-Data Format - Motivation

Present situation:

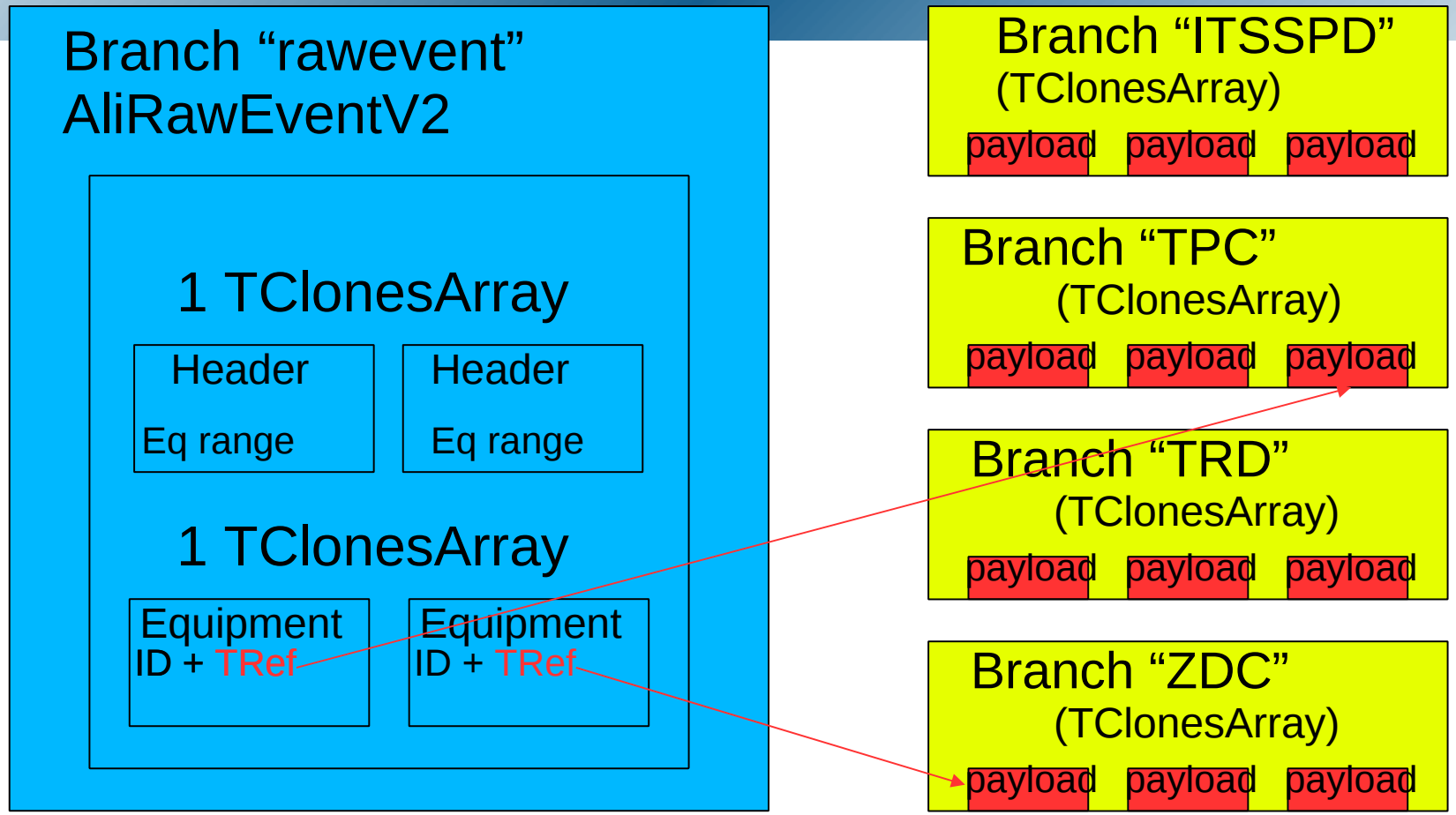
- From DAQ point of view:
  - High CPU load on GDCs, limitation on data taking rate
  - Overhead in raw-data file-size
  - Both effects are especially pronounced in case of higher event rate and smaller event size
- From Offline point of view:
  - Raw-data format optimized for time performance within DAQ framework -> not necessarily the most optimal format for offline raw-data read access
  - Not optimal compression, data written in non-split mode

Present format



- Event (AliRawEvent): Header + TObjArray of sub-events (AliRawEvent)
  - Each sub-event – header + TObjArray of equipments (AliRawEquipment)
    - Each equipment – header + TRef to payload
- Payloads written in separate detector branches
- Data written in non-split mode

New Raw-Data Format – Flat Structure



- Event – 1 TClonesArray of event/sub-event header + 1 TClonesArray of equipments
  - Each sub-event header contains indexes of first and last associated equipments
  - Each equipment – header + TRef to payload

New Raw-Data format (cont.)

- Event header stored in sub-event headers' TClonesArray
- Data written in full split mode
- All pointers to objects replaced by objects
- Headers' TClonesArray and tree created on the first event (and not on file opening) – after the DAQ header version has been identified
- Size of TClonesArray's correspond to the actual event content
  - Significant gain in case of runs with several trigger detector clusters
- Ignore TObject streamer for all classes except referenced raw-data payloads

New Raw-Data Format - Results

- On local machine:
  - ~2 times in processing rate
  - Twice smaller overhead in file-size (in case of no compression)
  - Better compression factor (for example 2.8 -> 3.5)
- On DAQ setup:
  - Up to 50% gain in the processing rate in case of no compression (therefore bottleneck should be elsewhere now)
  - Overhead reduction to be confirmed over wider range of event sizes
  - Full scale bench-marking is being done

Backward compatibility

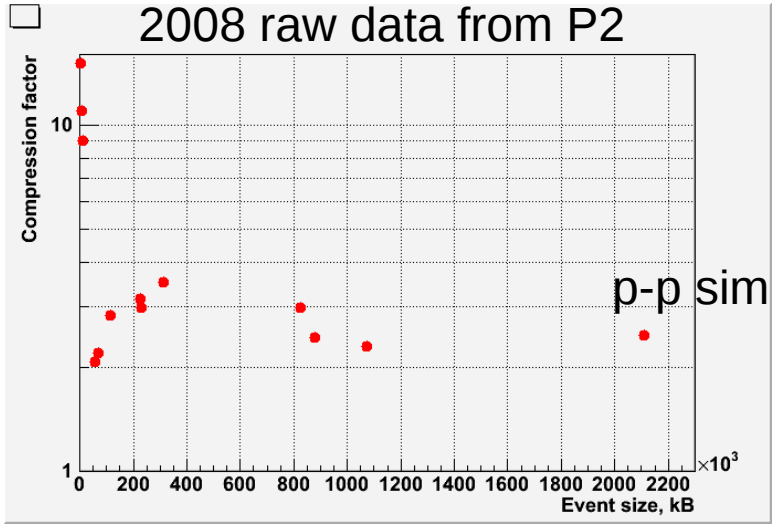
- New format implemented so that the reading interface is fully preserved
  - Base abstract classes AliRawVEvent and AliRawVEquipment
  - Both old and new classes derive from them and implement the corresponding read access methods
- No fixes in aliroot raw-data reader classes or elsewhere
  - Note: AliRawReader has to be changed in order to cache the pointer to event-header (to be 100% that it can't be mixed with sub-event headers)

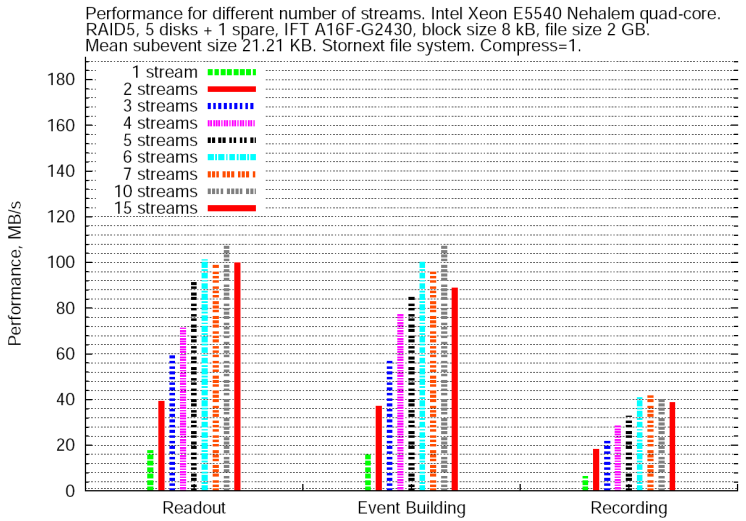
Backward compatibility - details

- Schema evolution:
  - Enabled for all new classes
  - Not used for:
    - Event/sub-event header – headers defined by DAQ and explicitly introduced as aliroot classes deriving from one base class
    - Equipment header – header fixed by DAQ, no need for any evolution
    - Raw-data payload contained – size + pointer only
- Custom streamers:
  - Old AliRawEvent and AliRawEquipment
    - In order to avoid problems with class map (different inheritance)
  - Event/sub-event header base class
    - schema evolution is enabled from now on
    - first/last equipment index is accommodated
- With all these backward compatibility is full preserved
- No time performance deterioration is observed in old raw-data reading (to be rechecked on various raw-data data)

ROOT Raw-Data Compression on GDCs

- New ROOT raw-data format gives better compression factor with less CPU usage
- DAQ is studying new GDC machines (dual quad-core CPUs):
  - Possibility to run up to 6-7 parallel recording processes
  - For the first time there is almost no impact on the data taking rate due to the compression





New RCU ALTRO raw-data format

- Motivation:
  - Common problems with data corruption due mainly to faulty ALTRO memory (single bits or full words are stuck)
    - Allow for fast and robust raw-data reading in case of corruption
    - Properly identification and logging of readout errors
  - Allow for HLT (H-RORC) raw-data processing
    - The data order had to be reversed

New RCU format

31 30 29		20 19	10 9	0
01	E	Altro channel header 1		
00	DW9	DW8	DW7	
00	DW6	DW5	DW4	
00	DW3	DW2	DW1	
00	DW0	0x00	0x00	
01	E	Altro channel header 2		
00	DW4	DW3	DW2	
00	DW1	DW2	0x00	
01	E	Altro channel header 3		
00	DW3	DW2	DW1	

Table 2.3 Order of 10 bit words in 32 bit data packet

Altro channel header [11:0]: [Channel Address from Altro]

Altro channel header [25:16]: [Block Length Number of 10 bit words from Altro]

Altro channel header [29]: E: Channel Error Bit, it is set to "1" if there has been a mismatch for channel address or block length received from ALTRO trailer. RCU will add a corrected Altro channel header followed by the data received from corresponding channel.

Altro channel header [31:30]: Word ID = "01" to mark ALTRAO Header : "00" for payload

Altro channel header [15:12] – [28:26]: Reserved = "0000"

DW: 10 bit data word.

0x00: Padding

- ALTRO data code in
  3x10 + 2 (reserved) bits = 32-bit word
  (old – 16x10 = 5 32-bit words)
- 2 reserved bit used as tags (ALTRO data, channel header, RCU trailer, end of trailer)
- Reversed order (first header then data)
  (old – first data then 'header')
- New 'error bit' in the channel header

New RCU-Trailer Format

31 30	29 26	25 0	
Word ID	**Parameter**		**Value**
10	Pay Load Length	0000	Number of 32 bit words
10	Error Register 1	0001	Error Registers for Branch A and B
10	Error Register 2	0010	Read out Errors
10	Error Register 3	0011	Number of Altro Trailer Errors
10	Act FEC A	0100	Active Front End Cards for Branch A
10	Act FEC B	0101	Active Front End Cards for Branch B
10	RDO CONFIG 1	0110	Readout configuration Register1 of Altro
10	RDO CONFIG 2	0111	Readout configuration Register2 of Altro
11	RCU ID	1000	RCU FW version / RCU address

Table 2.4 RCU Trailer

[31:30]: Word ID: two most significant bits are used to identify the start and end of RCU trailer. Word ID= "10" shows that the word belong to RCU Trailer where as "11" shows that this is the last word of the RCU trailer.

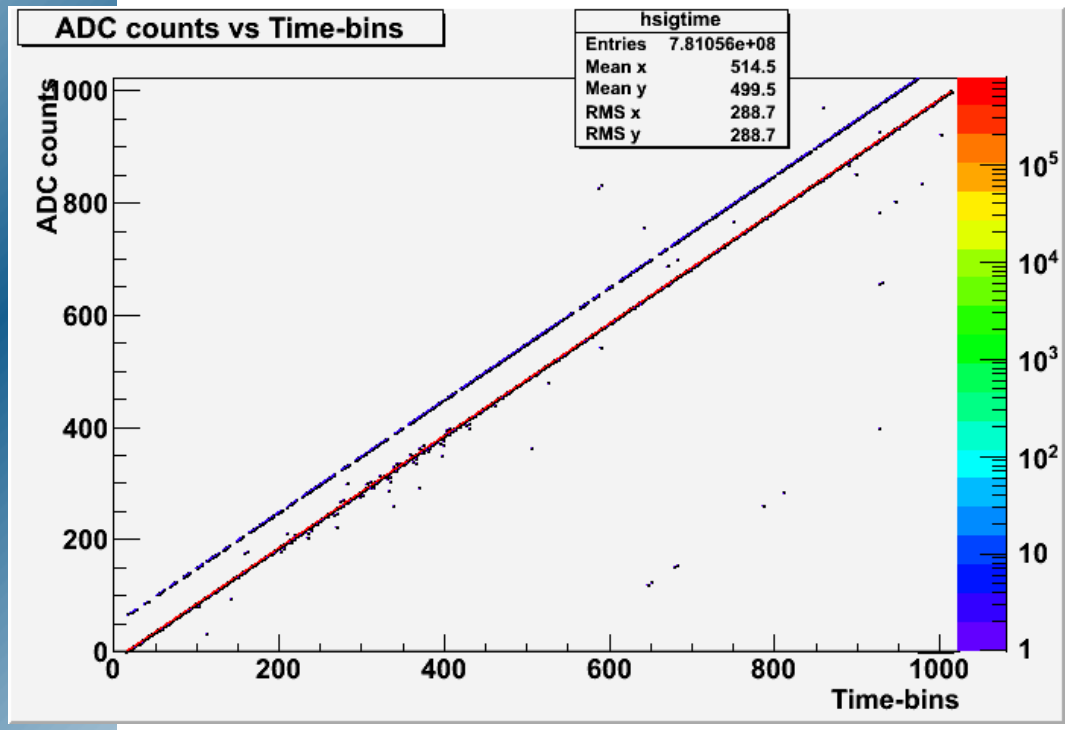
[29:26]: Parameter: four bits field is used to mark different words in the RCU trailer

[25:0]: Value: These 26 bits are used to record the status of the parameters listed in the table for the event.

New ALTRO Decoder

- One decoder for TPC, PHOS, EMCAL, FMD and HLT
- Decoder exploits the format properties so that to provide fast and robust data decoding
  - Uses the 2 dedicated bits for navigation in the data stream
- User interface follows the one used in the old fast decoder developed within HLT
  - “Bunch-by-bunch” instead of “sample-by-sample” reading
- Memory allocation is minimal
  - Limited to max bunch size
- Backward compatibility is 100% preserved
  - Automatically identifies and reads the old data format
  - Uses the old offline decoder with updated interface (also much faster now)

New ALTRO Decoder - Results with ramp data



- No decoding errors observed
- List of channels with 'wrong' contents fully coincides with the one from identified channels with 'faulty' ALTRO memory
- Combined with the results from other test runs (zero-suppressed data or 'black' events) give us confidence that all the data decoded correctly

New ALTRO Decoder - Code

- The code has been committed to aliroot svn trunk and is ready to be used
  - $ALICE_ROOT/RAW/AliAltroRawStreamV3.*
  - Example of its usage can be found in $ALICE_ROOT/RAW/readAltroV3.C macro
  - Detectors should derive AliDetAltroRawStream from AliAltroRawStreamV3, put the mapping and interface all raw-data reading tasks with the new decoder

- Time performance of the decoder:
  - 70-80 MB/s (TPC zero-suppressed data) decoding rate on a quite moderate desktop machine
  - Kenneth performed some bench-marking on the HLT cluster
    - Results are fully identical on old raw data transformed into the new format - OK
    - ~4-5% slower than the fast decoder of Per Thomas – OK

Conclusions

- New raw-data format:
  - Code is ready and has been committed to trunk and v4-16-Release
  - DAQ is checking and bench-marking it
  - Plans are to use it for the forthcoming data taking
- Raw-data tags:
  - Not in split mode
  - Do we really use them? If yes, do we adapt the format?
- Raw-data compression can offer great reduction of the needed storage resources and is under evaluation by DAQ
- New RCU format:
  - Format defined, implemented and finalized
  - Decoder implemented and tested – allows for fast and more robust data decoding
  - Detectors have to move to the new decoder (and new format)
  - Simulation of the new format is being implemented