Level 3 Status

Compression and Clustering

Nikola Whallon

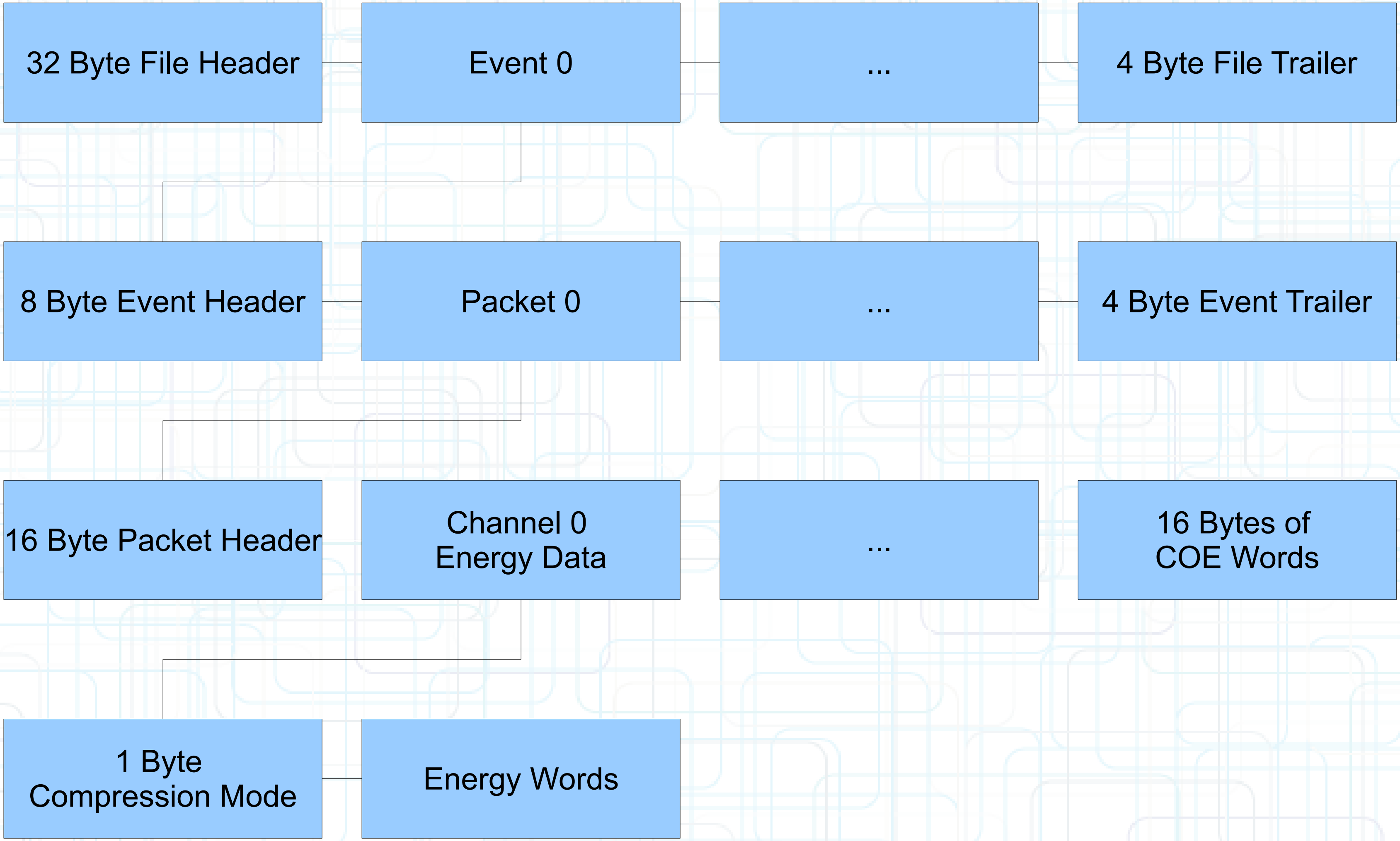
L3 Status Overview

- Updated L3 code was used in December and was able to capture all data coming into mandolin with plenty of resources to spare.
- L3 code using 8-bit compression was tested in December – it seems to be working with compression ratios of around **~60%**.
- Variable bit (3-8 bit) compression has been added to the L3 code, and needs to be tested for the March run.
- I have started to work on clustering, and believe that it can be used to perform an online L3 cut.

Variable Bit Compression

- Similar to how 8-bit compression works, if the range of energy word values is < 2^n, the the energy words can be stored as n-bit data (with extra minimum and offset words).
- However, because computer memory is allocated in bytes (8-bits), generic n-bit data has to be "packed" into 8-bit words.
- 8 n-bit words thus can become n 8-bit words, and this works out perfectly because the number of energy words is a multiple of 8.
- Currently, I have to test a version of the L3 code with 3, 4, 5, 6, and 7-bit compression added. These would result in compression ratios of 12, 12, 12, 12, and 12 percent, respectively.

Data Format



32 Byte File Header

Event 0

...

4 Byte File Trailer

8 Byte Event Header

Packet 0

...

4 Byte Event Trailer

16 Byte Packet Header

Channel 0
Energy Data

...

16 Bytes of
COE Words

1 Byte
Compression Mode

Energy Words

Clustering

- Clustering would be one way of implementing a L3 cut.
- One algorithm for clustering would be to sum the samples of each channel, make a cut on the value of each channel sum, and then recursively group channels into clusters based on their distance to othe channels in a cluster.
- If such an algorithm were fast enough to work online, it would be most efficient to have it run on another thread of the main L3 code, right before the data is compressed.

slide with picture of the crystal array and clusters

Clustering Test

- To test if the algorithm would work fast enough, I generated a 2D array of 3000 x 64 values (not included when timing).
- I then summed the respective 64 “channels,” ran through the resulting 3000 sums and made an arbitrary cut.
- I then went through the remaining elements, and implemented the clustering algorithm.
- Doing this 256 times (roughly representing the number of events each node receives data for per spill) only took **~10-20 ms**.

Conclusion

- 8-bit compression should be used for the majority of the data during the March run.
- Some data should be taken with variable bit compression during the March run to test, and if all is good, more data can be taken with variable bit compression.
- I will spend time working on integrating clustering into L3, and this can be tested during the March run as well.

Thanks!