INTT weekly meeting

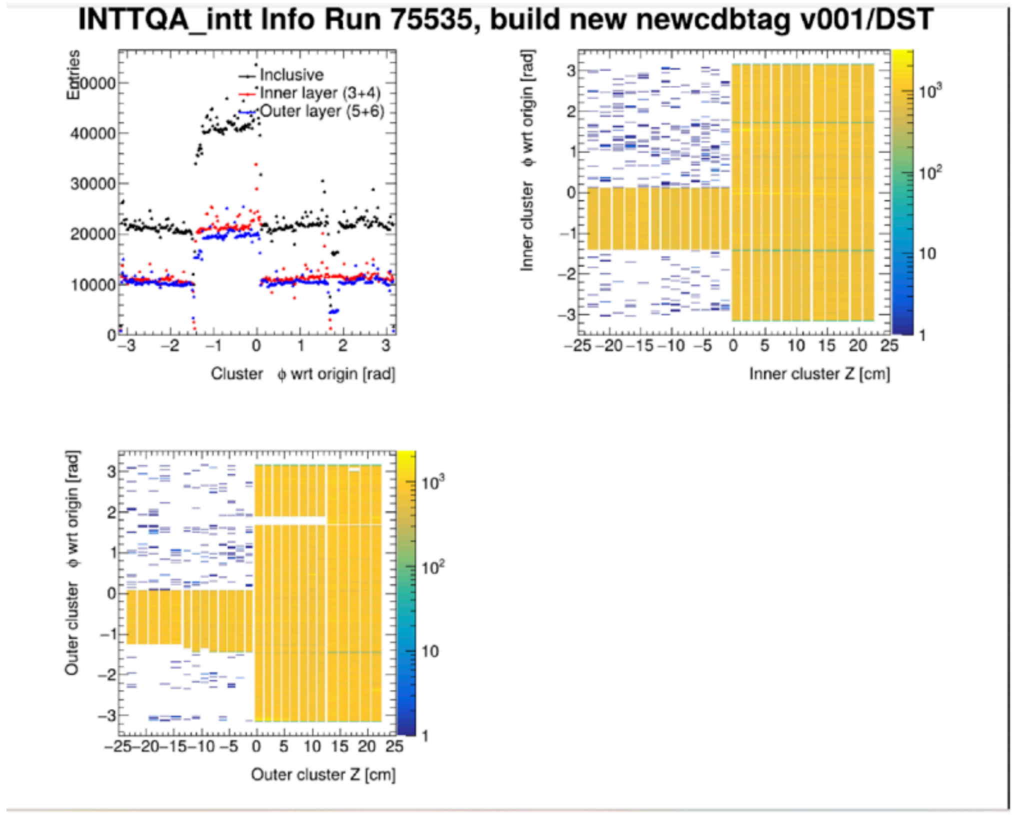
INTT large acceptance missing issue & BCO QA update

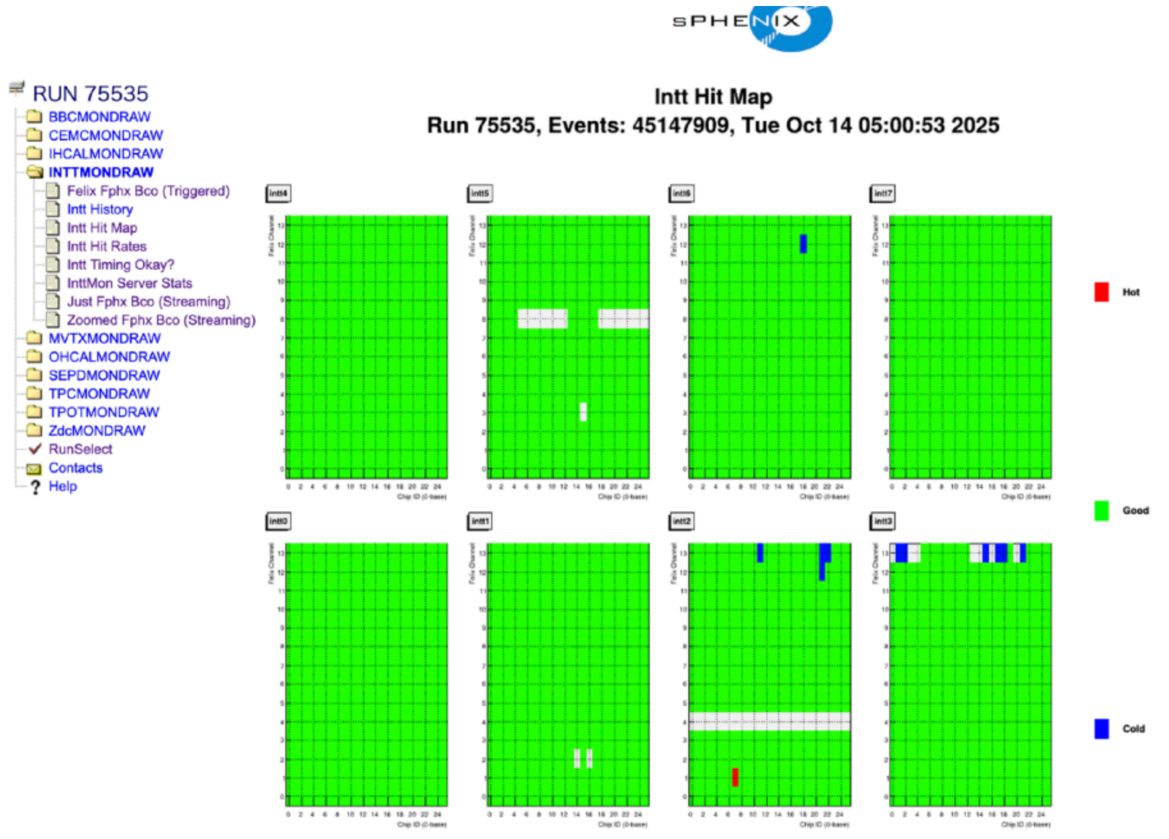
Jaein Hwang (Korea Univ.)

Oct. 30 2025

Large acceptance missing(cont.)

Yuko presented Offline shifter reported several time that we have **large acceptance missing**

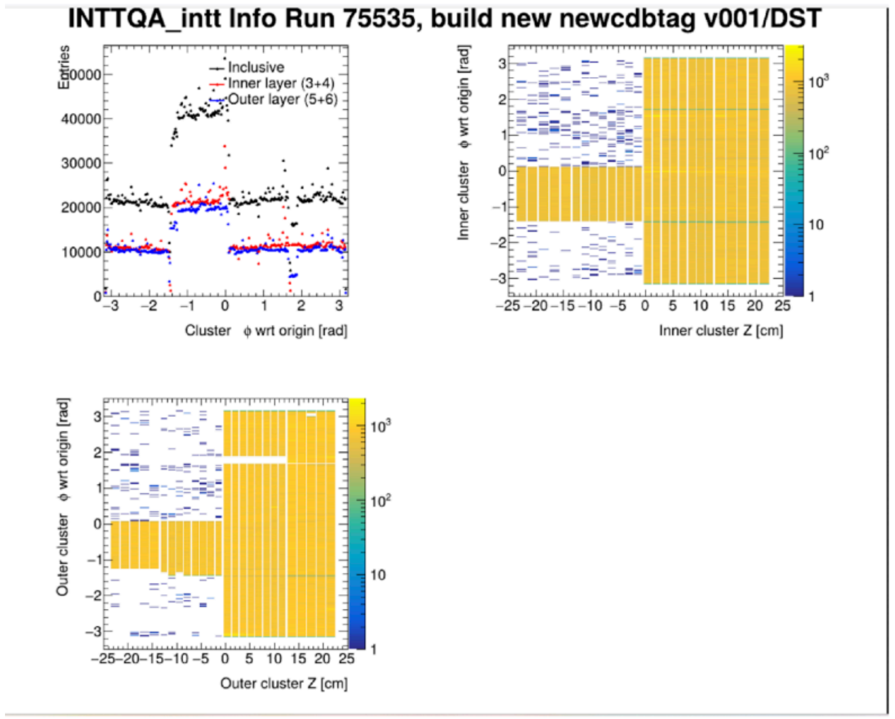


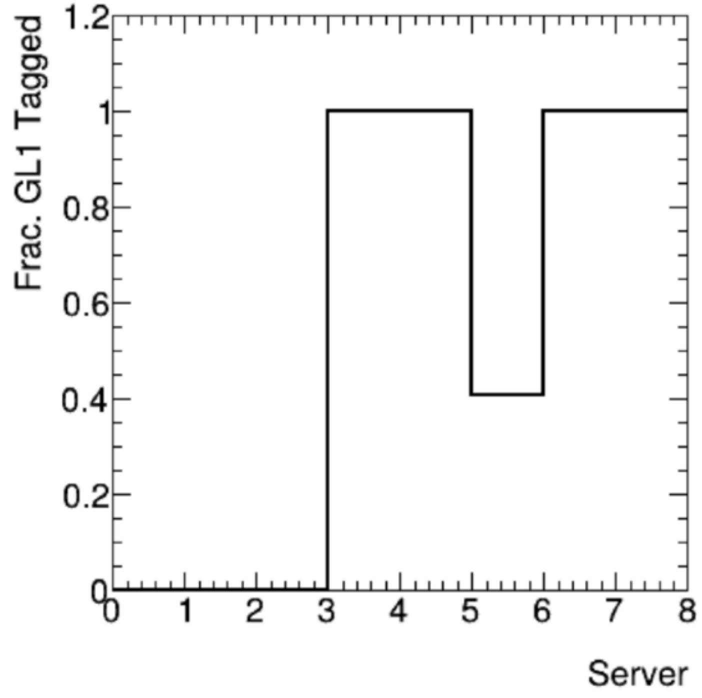


But Yuko confirmed that the Online monitor for these runs look OK.
And Akitmomo checked the raw-data size, but no significant issue found from the raw data size.

Large acceptance missing

Yuko presented Offline shifter reported several time that we have large acceptance missing





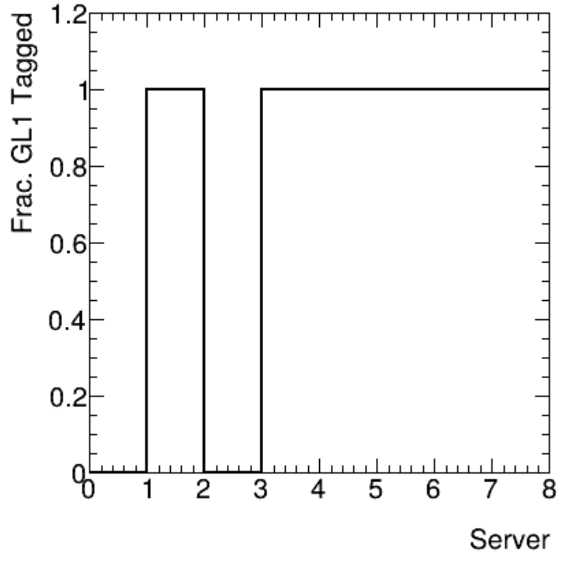
$$\text{Frac.GL1 tagged} = \frac{\text{(Number of GL1 BCO received by FELIX)}}{\text{(Number of Gl1 BCO sent by GTM to FELIX)}}$$

And Joe also mentioned that **BCO QA shows significant data lost from missing server**
It was getting serious issue it's not just few acceptance missing, something happened which has to be addressed

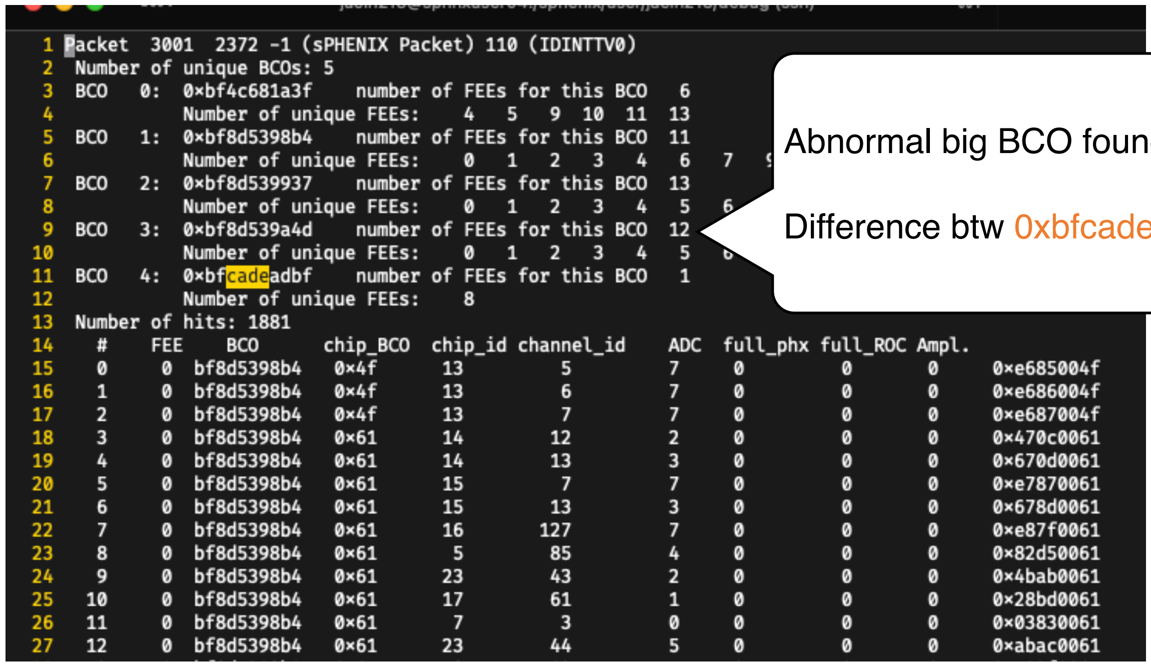
Rawdata check Run 75935 INTT0

Checked raw data file(.evt file) before event combining by ddump commend
ex) ddump /sphenix/lustre01/sphnxpro/physics/INTT/physics/physics_intt0-00075905-0000.evt

Run 75935



Raw data from Run 75935 INTT0



```
Packet  3001  2372 -1 (sPHENIX Packet) 110 (IDINTTV0)
Number of unique BCOs: 5
BCO   0:  0×bf4c681a3f    number of FEEs for this BCO   6
              Number of unique FEEs:    4   5   9  10  11  13
BCO   1:  0×bf8d5398b4    number of FEEs for this BCO  11
              Number of unique FEEs:    0   1   2   3   4   6   7
BCO   2:  0×bf8d539937    number of FEEs for this BCO  13
              Number of unique FEEs:    0   1   2   3   4   5   6
BCO   3:  0×bf8d539a4d    number of FEEs for this BCO  12
              Number of unique FEEs:    0   1   2   3   4   5
BCO   4:  0×bfcadeadbf    number of FEEs for this BCO   1
              Number of unique FEEs:    8
Number of hits: 1881
  #    FEE     BCO       chip_BCO  chip_id channel_id   ADC  full_phx full_ROC Ampl.
  0      0  bf8d5398b4   0×4f      13          5        7     0         0       0     0×e685004f
  1      0  bf8d5398b4   0×4f      13          6        7     0         0       0     0×e686004f
  2      0  bf8d5398b4   0×4f      13          7        7     0         0       0     0×e687004f
  3      0  bf8d5398b4   0×61      14         12        2     0         0       0     0×470c0061
  4      0  bf8d5398b4   0×61      14         13        3     0         0       0     0×670d0061
  5      0  bf8d5398b4   0×61      15          7        7     0         0       0     0×e7870061
  6      0  bf8d5398b4   0×61      15         13        3     0         0       0     0×678d0061
  7      0  bf8d5398b4   0×61      16        127        7     0         0       0     0×e87f0061
  8      0  bf8d5398b4   0×61       5         85        4     0         0       0     0×82d50061
  9      0  bf8d5398b4   0×61      23         43        2     0         0       0     0×4bab0061
 10      0  bf8d5398b4   0×61      17         61        1     0         0       0     0×28bd0061
 11      0  bf8d5398b4   0×61       7          3        0     0         0       0     0×03830061
 12      0  bf8d5398b4   0×61      23         44        5     0         0       0     0×abac0061
```

Abnormal big BCO found 0xbfcadeadbf

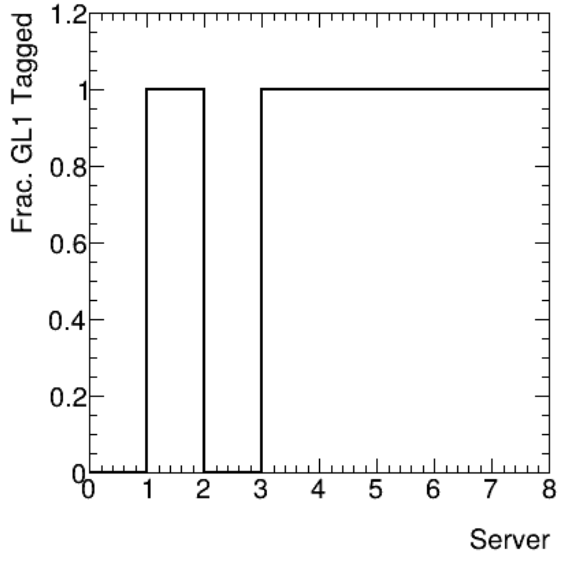
Difference btw 0xbfcadeadbf and 0xbf8d539a4d

Found that at the very beginning of the data taking, we have 0xdfcadeadbf BCO
As far as I know, 0xad##cade type is the header word, which shouldn't be in the list of hits
This abnormal extremely big BCO misread from header makes stop decoding INTT data

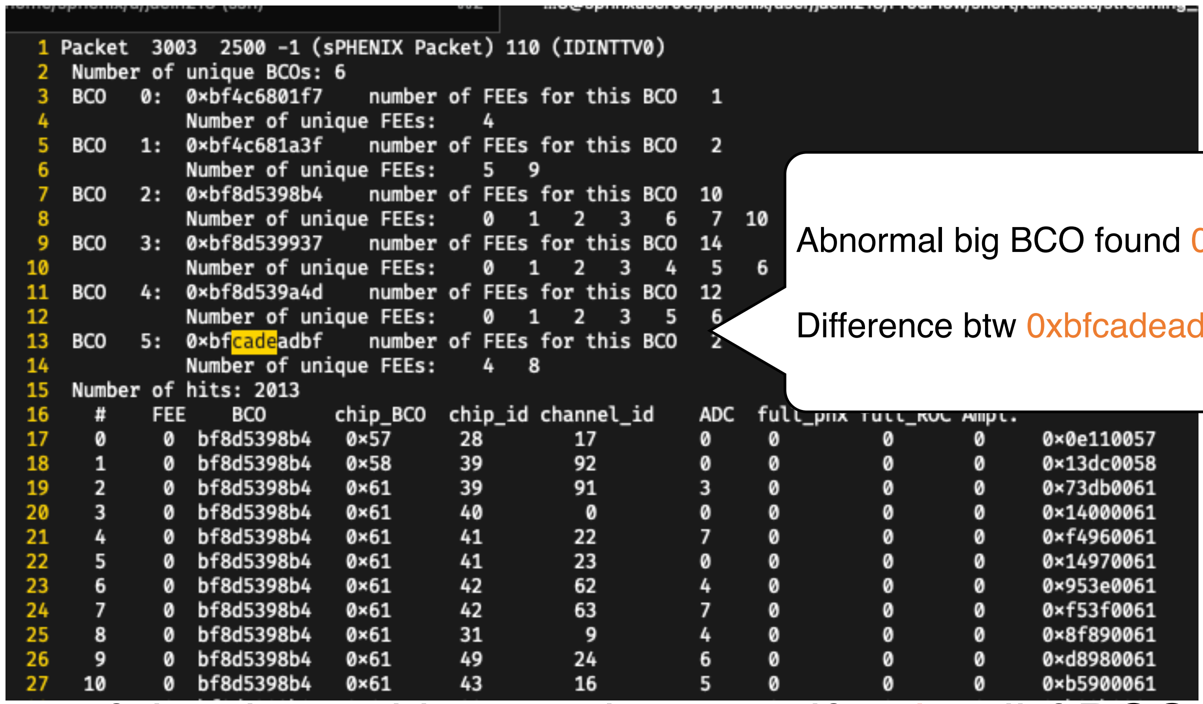
Rawdata check Run 75935 INTT2

Checked raw data file(.evt file) before event combining by ddump commend
ex) ddump /sphenix/lustre01/sphnxpro/physics/INTT/physics/physics_intt0-00075905-0000.evt

Run 75935



Raw data from Run 75935 INTT2



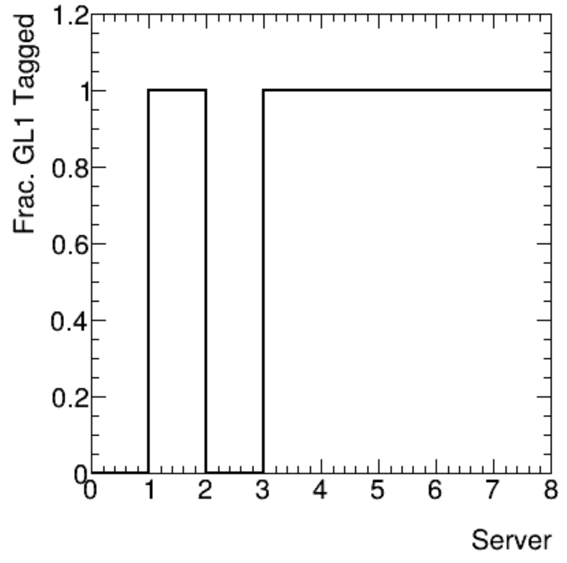
```
Packet  3003  2500 -1 (sPHENIX Packet) 110 (IDINTTV0)
 Number of unique BCOs: 6
 BCO   0:  0×bf4c6801f7     number of FEEs for this BCO   1
           Number of unique FEEs:    4
 BCO   1:  0×bf4c681a3f     number of FEEs for this BCO   2
           Number of unique FEEs:    5   9
 BCO   2:  0×bf8d5398b4     number of FEEs for this BCO  10
           Number of unique FEEs:    0   1   2   3   6   7  10
 BCO   3:  0×bf8d539937     number of FEEs for this BCO  14
           Number of unique FEEs:    0   1   2   3   4   5   6
 BCO   4:  0×bf8d539a4d     number of FEEs for this BCO  12
           Number of unique FEEs:    0   1   2   3   5   6
 BCO   5:  0×bfcadeadbf     number of FEEs for this BCO   2
           Number of unique FEEs:    4   8
 Number of hits: 2013
   #    FEE    BCO       chip_BCO  chip_id channel_id    ADC  full_phx full_ROC Ampl.
   0      0  bf8d5398b4   0×57      28         17        0    0          0        0       0×0e110057
   1      0  bf8d5398b4   0×58      39         92        0    0          0        0       0×13dc0058
   2      0  bf8d5398b4   0×61      39         91        3    0          0        0       0×73db0061
   3      0  bf8d5398b4   0×61      40          0        0    0          0        0       0×14000061
   4      0  bf8d5398b4   0×61      41         22        7    0          0        0       0×f4960061
   5      0  bf8d5398b4   0×61      41         23        0    0          0        0       0×14970061
   6      0  bf8d5398b4   0×61      42         62        4    0          0        0       0×953e0061
   7      0  bf8d5398b4   0×61      42         63        7    0          0        0       0×f53f0061
   8      0  bf8d5398b4   0×61      31          9        4    0          0        0       0×8f890061
   9      0  bf8d5398b4   0×61      49         24        6    0          0        0       0×d8980061
  10      0  bf8d5398b4   0×61      43         16        5    0          0        0       0×b5900061
```

Abnormal big BCO found 0xbfcadeadbf

Difference btw 0xbfcadeadbf and 0xbf8d539a4d

Found that at the very beginning of the data taking, we have 0xdfcadeadbf BCO
As far as I know, 0xad##cade type is the header word, which shouldn't be in the list of hits
This abnormal extremely big BCO misread from header makes stop decoding INTT data

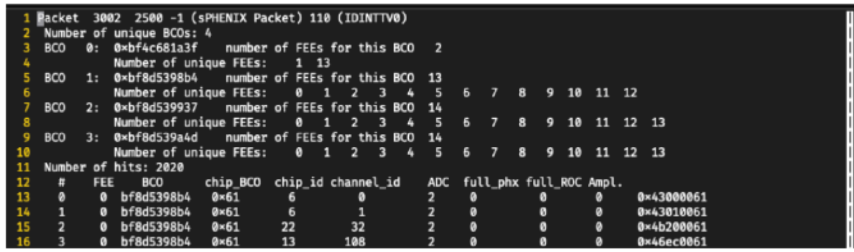
Rawdata check Run 75935 INTT2



Then, why big BCO makes stop decoding INTT raw data?
Let's try to understand with very simplified example

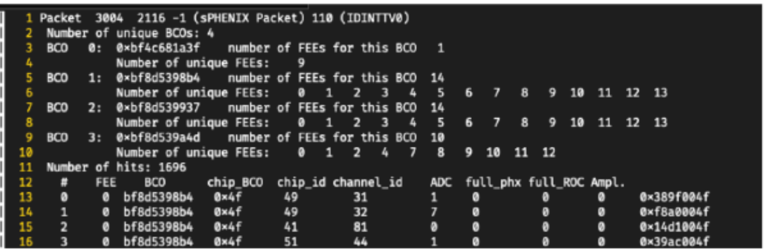
We don't have abnormal BCO from healthy servers

RUN 75905 INTT1



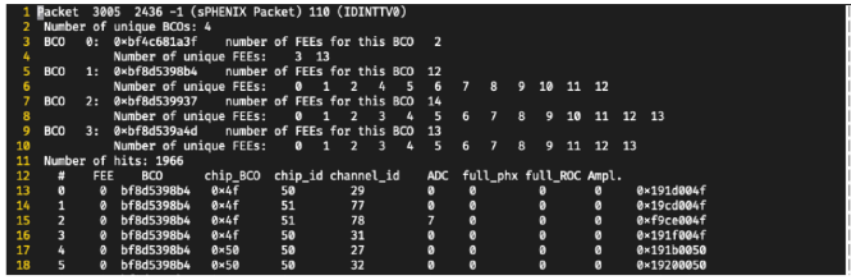
```
Packet  3002  2500 -1 (sPHENIX Packet) 110 (IDINTTV0)
 Number of unique BCOs: 4
 BCO   0:  0×bf4c681a3f    number of FEEs for this BCO   2
           Number of unique FEEs:    1  13
 BCO   1:  0×bf8d5398b4    number of FEEs for this BCO  13
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12
 BCO   2:  0×bf8d539937    number of FEEs for this BCO  14
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12  13
 BCO   3:  0×bf8d539a4d    number of FEEs for this BCO  14
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12  13
 Number of hits: 2020
   #    FEE    BCO       chip_BCO  chip_id channel_id    ADC  full_phx full_ROC Ampl.
   0     0  bf8d5398b4   0×61       6         0         2    0         0       0     0×43000061
   1     0  bf8d5398b4   0×61       6         1         2    0         0       0     0×43010061
   2     0  bf8d5398b4   0×61      22        32         2    0         0       0     0×4b200061
   3     0  bf8d5398b4   0×61      13       108         2    0         0       0     0×46ec0061
```

RUN 75905 INTT3



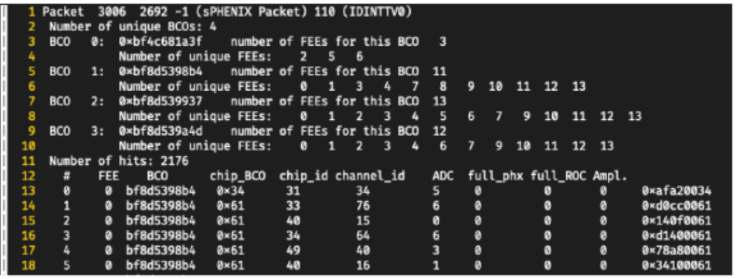
```
Packet  3004  2116 -1 (sPHENIX Packet) 110 (IDINTTV0)
 Number of unique BCOs: 4
 BCO   0:  0×bf4c681a3f    number of FEEs for this BCO   1
           Number of unique FEEs:    9
 BCO   1:  0×bf8d5398b4    number of FEEs for this BCO  14
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12  13
 BCO   2:  0×bf8d539937    number of FEEs for this BCO  14
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12  13
 BCO   3:  0×bf8d539a4d    number of FEEs for this BCO  10
           Number of unique FEEs:    0   1   2   4   7   8   9  10  11  12
 Number of hits: 1696
   #    FEE    BCO       chip_BCO  chip_id channel_id    ADC  full_phx full_ROC Ampl.
   0     0  bf8d5398b4   0×4f      49        31         1    0         0       0     0×389f004f
   1     0  bf8d5398b4   0×4f      49        32         7    0         0       0     0×f8a0004f
   2     0  bf8d5398b4   0×4f      41        81         0    0         0       0     0×14d1004f
   3     0  bf8d5398b4   0×4f      51        44         1    0         0       0     0×39ac004f
```

RUN 75905 INTT4



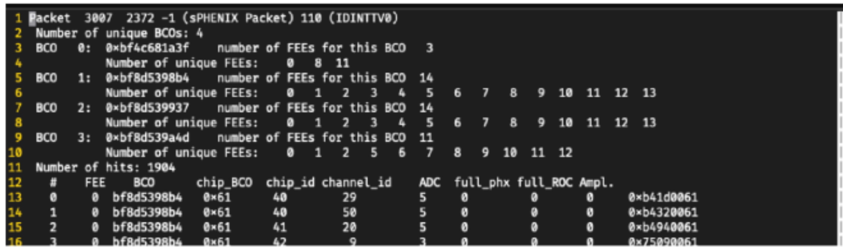
```
Packet  3005  2436 -1 (sPHENIX Packet) 110 (IDINTTV0)
 Number of unique BCOs: 4
 BCO   0:  0×bf4c681a3f    number of FEEs for this BCO   2
           Number of unique FEEs:    3  13
 BCO   1:  0×bf8d5398b4    number of FEEs for this BCO  12
           Number of unique FEEs:    0   1   2   4   5   6   7   8   9  10  11  12
 BCO   2:  0×bf8d539937    number of FEEs for this BCO  14
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12  13
 BCO   3:  0×bf8d539a4d    number of FEEs for this BCO  13
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  11  12  13
 Number of hits: 1966
   #    FEE    BCO       chip_BCO  chip_id channel_id    ADC  full_phx full_ROC Ampl.
   0     0  bf8d5398b4   0×4f      50        29         0    0         0       0     0×191d004f
   1     0  bf8d5398b4   0×4f      51        77         0    0         0       0     0×19cd004f
   2     0  bf8d5398b4   0×4f      51        78         7    0         0       0     0×f9ce004f
   3     0  bf8d5398b4   0×4f      50        31         0    0         0       0     0×191f004f
   4     0  bf8d5398b4   0×50      50        27         0    0         0       0     0×191b0050
   5     0  bf8d5398b4   0×50      50        32         0    0         0       0     0×19200050
```

RUN 75905 INTT5



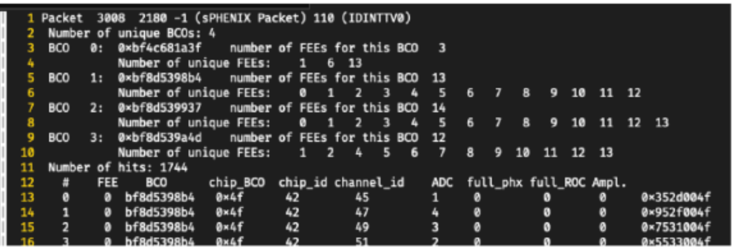
```
Packet  3006  2692 -1 (sPHENIX Packet) 110 (IDINTTV0)
 Number of unique BCOs: 4
 BCO   0:  0×bf4c681a3f    number of FEEs for this BCO   3
           Number of unique FEEs:    2   5   6
 BCO   1:  0×bf8d5398b4    number of FEEs for this BCO  11
           Number of unique FEEs:    0   1   3   4   7   8   9  10  11  12  13
 BCO   2:  0×bf8d539937    number of FEEs for this BCO  13
           Number of unique FEEs:    0   1   2   3   4   5   6   7   9  10  11  12  13
 BCO   3:  0×bf8d539a4d    number of FEEs for this BCO  12
           Number of unique FEEs:    0   1   2   3   4   6   7   9  10  11  12  13
 Number of hits: 2176
   #    FEE    BCO       chip_BCO  chip_id channel_id    ADC  full_phx full_ROC Ampl.
   0     0  bf8d5398b4   0×34      31        34         5    0         0       0     0×afa20034
   1     0  bf8d5398b4   0×61      33        76         6    0         0       0     0×d0cc0061
   2     0  bf8d5398b4   0×61      40        15         0    0         0       0     0×140f0061
   3     0  bf8d5398b4   0×61      34        64         6    0         0       0     0×d1400061
   4     0  bf8d5398b4   0×61      49        40         3    0         0       0     0×78a80061
   5     0  bf8d5398b4   0×61      40        16         1    0         0       0     0×34100061
```

RUN 75905 INTT6



```
Packet  3007  2372 -1 (sPHENIX Packet) 110 (IDINTTV0)
 Number of unique BCOs: 4
 BCO   0:  0×bf4c681a3f    number of FEEs for this BCO   3
           Number of unique FEEs:    0   8  11
 BCO   1:  0×bf8d5398b4    number of FEEs for this BCO  14
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12  13
 BCO   2:  0×bf8d539937    number of FEEs for this BCO  14
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12  13
 BCO   3:  0×bf8d539a4d    number of FEEs for this BCO  11
           Number of unique FEEs:    0   1   2   5   6   7   8   9  10  11  12
 Number of hits: 1904
   #    FEE    BCO       chip_BCO  chip_id channel_id    ADC  full_phx full_ROC Ampl.
   0     0  bf8d5398b4   0×61      40        29         5    0         0       0     0×b41d0061
   1     0  bf8d5398b4   0×61      40        50         5    0         0       0     0×b4320061
   2     0  bf8d5398b4   0×61      41        20         5    0         0       0     0×b4940061
   3     0  bf8d5398b4   0×61      42         9         3    0         0       0     0×75090061
```

RUN 75905 INTT7



```
Packet  3008  2180 -1 (sPHENIX Packet) 110 (IDINTTV0)
 Number of unique BCOs: 4
 BCO   0:  0×bf4c681a3f    number of FEEs for this BCO   3
           Number of unique FEEs:    1   6  13
 BCO   1:  0×bf8d5398b4    number of FEEs for this BCO  13
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12
 BCO   2:  0×bf8d539937    number of FEEs for this BCO  14
           Number of unique FEEs:    0   1   2   3   4   5   6   7   8   9  10  11  12  13
 BCO   3:  0×bf8d539a4d    number of FEEs for this BCO  12
           Number of unique FEEs:    1   2   4   5   6   7   8   9  10  11  12  13
 Number of hits: 1744
   #    FEE    BCO       chip_BCO  chip_id channel_id    ADC  full_phx full_ROC Ampl.
   0     0  bf8d5398b4   0×4f      42        45         1    0         0       0     0×352d004f
   1     0  bf8d5398b4   0×4f      42        47         4    0         0       0     0×952f004f
   2     0  bf8d5398b4   0×4f      42        49         3    0         0       0     0×7531004f
   3     0  bf8d5398b4   0×4f      42        51         2    0         0       0     0×5533004f
```

Mechanism of event combining-Simple version (cont.)

How inttSinglePoolInput works(combining data with decoder)
- Check GL1 BCO from gl1 raw data.
- Decode INTT until we can find biggest BCO from INTT is greater than GL1 BCO
- Stop INTT decoding until we can find GL1 BCO equal or greater than biggest BCO from INTT

Not easy to understand.. Let me put very simple version as an example

Mechanism of event combining-Simple version (cont.)

How inttSinglePoolInput works(combining data with decoder)
- Check GL1 BCO from gl1 raw data.
- Decode INTT until we can find biggest BCO from INTT is greater than GL1 BCO
- Stop INTT decoding until we can find GL1 BCO equal or greater than biggest BCO from INTT

let's say If the decoding depth is 4, we decode 4 blocks(packets) by once

1st decoding

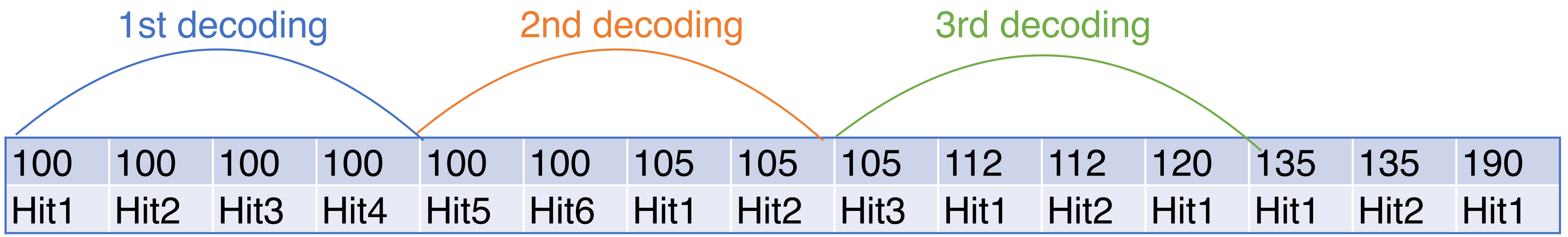
BCO from our FELIX (Unique BCO)	100	100	100	100	100	100	105	105	105	112	112	120	135	135	190
HIT index	Hit1	Hit2	Hit3	Hit4	Hit5	Hit6	Hit1	Hit2	Hit3	Hit1	Hit2	Hit1	Hit1	Hit2	Hit1

1) Decode gl1 raw data, and found that 1st GL1 BCO is 100 -> our reference BCO
2) Decode INTT raw data, we only decode 4 blocks since our decoding depth is 4.(1st decoding)
3) Check maximum BCO from INTT evt.

Mechanism of event combining-Simple version (cont.)

How inttSinglePoolInput works(combining data with decoder)
- Check GL1 BCO from gl1 raw data.
- Decode INTT until we can find biggest BCO from INTT is greater than GL1 BCO
- Stop INTT decoding until we can find GL1 BCO equal or greater than biggest BCO from INTT

let's say If the decoding depth is 4, **we decode 4 blocks(packets) by once**



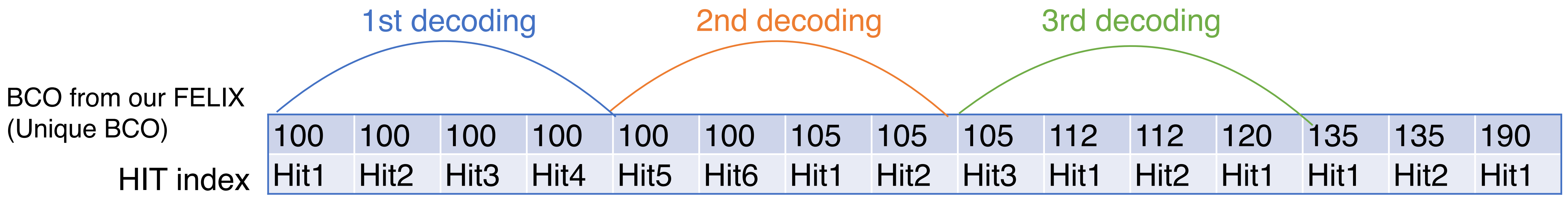
BCO from our FELIX (Unique BCO)	100	100	100	100	100	100	105	105	105	112	112	120	135	135	190
HIT index	Hit1	Hit2	Hit3	Hit4	Hit5	Hit6	Hit1	Hit2	Hit3	Hit1	Hit2	Hit1	Hit1	Hit2	Hit1

1) Decode gl1 raw data, and found that 1st GL1 BCO is 100 -> our reference BCO
2) Decode INTT raw data, we only decode 4 blocks since our decoding depth is 4.(1st decoding)
3) Check maximum BCO from INTT evt. It's still 100! Let's decode more! (2nd decoding)
4) Check maximum BCO from INTT evt. It's now 105! Let's stop decoding until we can find corresponding BCO from gl1 raw data
5) Decode gl1 raw data, and found that 2nd GL1 BCO is 105.

Mechanism of event combining-Simple version (cont.)

How inttSinglePoolInput works(combining data with decoder)
- Check GL1 BCO from gl1 raw data.
- Decode INTT until we can find biggest BCO from INTT is greater than GL1 BCO
- Stop INTT decoding until we can find GL1 BCO equal or greater than biggest BCO from INTT

let's say If the decoding depth is 4, **we decode 4 blocks(packets) by once**



BCO from our FELIX (Unique BCO)	100	100	100	100	100	100	105	105	105	112	112	120	135	135	190
HIT index	Hit1	Hit2	Hit3	Hit4	Hit5	Hit6	Hit1	Hit2	Hit3	Hit1	Hit2	Hit1	Hit1	Hit2	Hit1

1) Decode gl1 raw data, and found that 1st GL1 BCO is 100 -> our reference BCO
2) Decode INTT raw data, we only decode 4 blocks since our decoding depth is 4.(1st decoding)
3) Check maximum BCO from INTT evt. It's still 100! Let's decode more! (2nd decoding)
4) Check maximum BCO from INTT evt. It's now 105! Let's stop decoding until we can find corresponding BCO from gl1 raw data
5) Decode gl1 raw data, and found that 2nd GL1 BCO is 105.
6) GL1 BCO is 105 and our maximum BCO from INTT is 105. Let's decode more.(3rd decoding)
7).. keep going..

Case for the problem(cont.)

How inttSinglePoolInput works(combining data with decoder)
- Check GL1 BCO from gl1 raw data.
- Decode INTT until we can find biggest BCO from INTT is greater than GL1 BCO
- Stop INTT decoding until we can find GL1 BCO equal or greater than biggest BCO from INTT

let's say If the decoding depth is 4, **we decode 4 blocks(packets) by once**

1st decoding

BCO from our FELIX (Unique BCO)	9999	100	100	100	100	100	105	105	105	112	112	120	135	135	190
HIT index	Hit1	Hit1	Hit2	Hit3	Hit4	Hit5	Hit1	Hit2	Hit3	Hit1	Hit2	Hit1	Hit1	Hit2	Hit1

But if we have extremely big BCO at beginning of raw data, what happened?
1) Decode gl1 raw data, and found that 1st GL1 BCO is 100 -> our reference BCO
2) Decode INTT raw data, we only decode 4 blocks since our decoding depth is 4.(1st decoding)

Case for the problem

How inttSinglePoolInput works(combining data with decoder)
- Check GL1 BCO from gl1 raw data.
- Decode INTT until we can find biggest BCO from INTT is greater than GL1 BCO
- Stop INTT decoding until we can find GL1 BCO equal or greater than biggest BCO from INTT

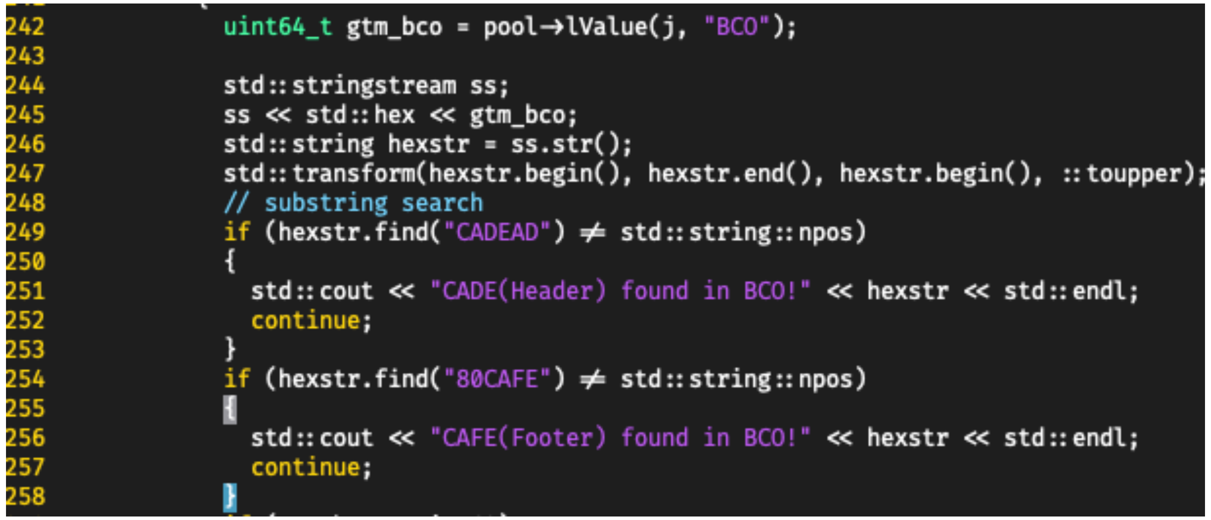
let's say If the decoding depth is 4, **we decode 4 blocks(packets) by once**

1st decoding

BCO from our FELIX (Unique BCO)	9999	100	100	100	100	100	105	105	105	112	112	120	135	135	190
HIT index	Hit1	Hit1	Hit2	Hit3	Hit4	Hit5	Hit1	Hit2	Hit3	Hit1	Hit2	Hit1	Hit1	Hit2	Hit1

But if we have extremely big BCO at beginning of raw data, what happened?
1) Decode gl1 raw data, and found that 1st GL1 BCO is 100 -> our reference BCO
2) Decode INTT raw data, we only decode 4 blocks since our decoding depth is 4.(1st decoding)
3) Check maximum BCO from INTT evt. It's still 9999! **Let's STOP INTT decoding until GL1 BCO is equal or greater than 9999!**
4) **Never decode INTT raw data(Looks like INTT data is empty because nothing more decoded!!)**

Solution on software level



```
uint64_t gtm_bco = pool->lValue(j, "BCO");

std::stringstream ss;
ss << std::hex << gtm_bco;
std::string hexstr = ss.str();
std::transform(hexstr.begin(), hexstr.end(), hexstr.begin(), ::toupper);
// substring search
if (hexstr.find("CADEAD") != std::string::npos)
{
  std::cout << "CADE(Header) found in BCO!" << hexstr << std::endl;
  continue;
}
if (hexstr.find("80CAFE") != std::string::npos)
{
  std::cout << "CAFE(Footer) found in BCO!" << hexstr << std::endl;
  continue;
}
```

I do not mention today's meeting, but we rarely have the case reading footer, which also gives us an abnormally large BCO. See backup slide for the details.

Suggestion to address this issue

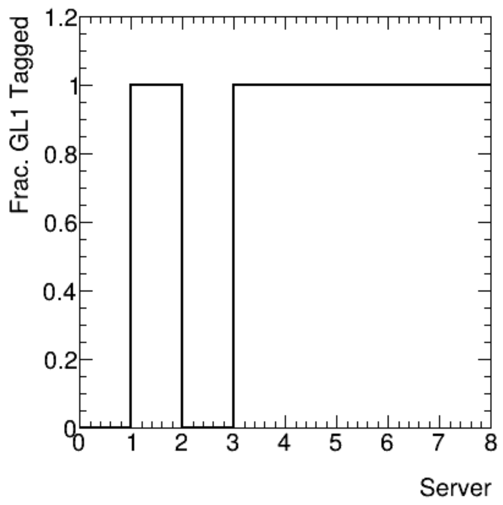
Check INTT BCO, if BCO has patterns 'CADE' (header) or 'CAFE' (footer) are excluded as abnormal BCO.

Just adding very few lines!

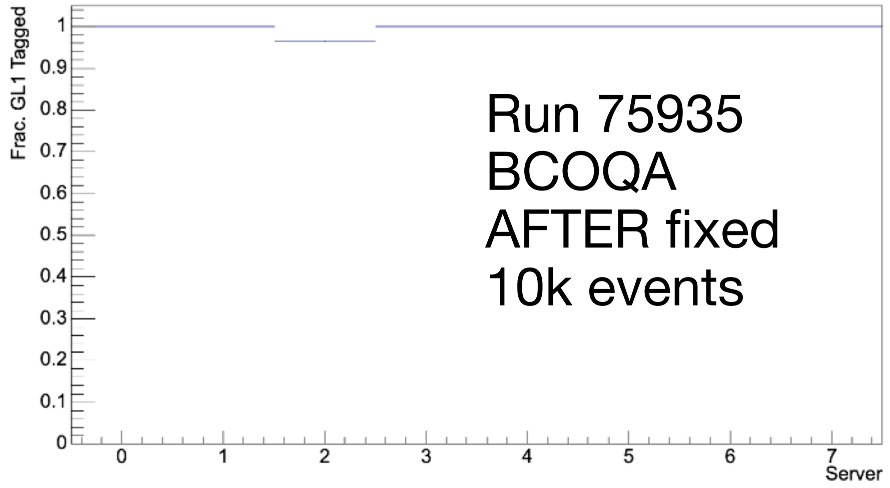
Why it happened? I don't know. 🤷

But we should try to address the issue without touching FELIX server. Since we are mostly stable better not to touch firmware itself! (Of course, we should understand why it happened. Discussion with DAQ expert needed)

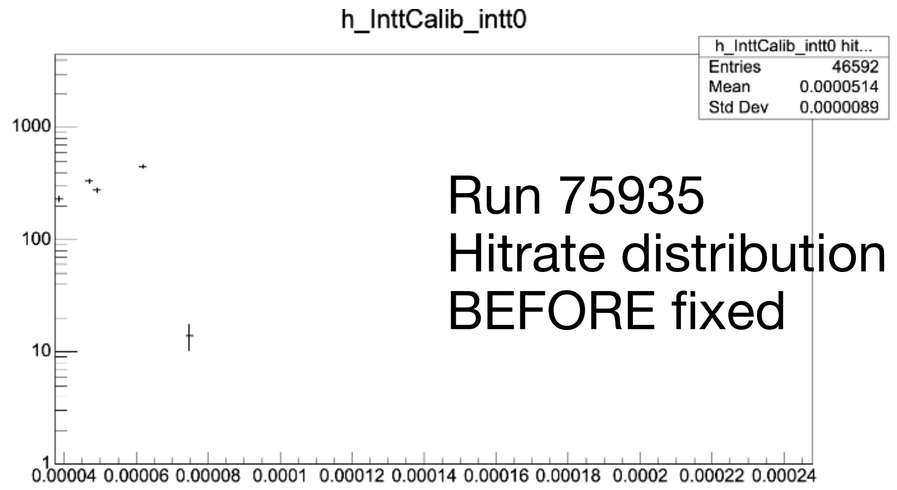
Before/After fix Run 75935



Run 75935
BCOQA
BEFORE fixed

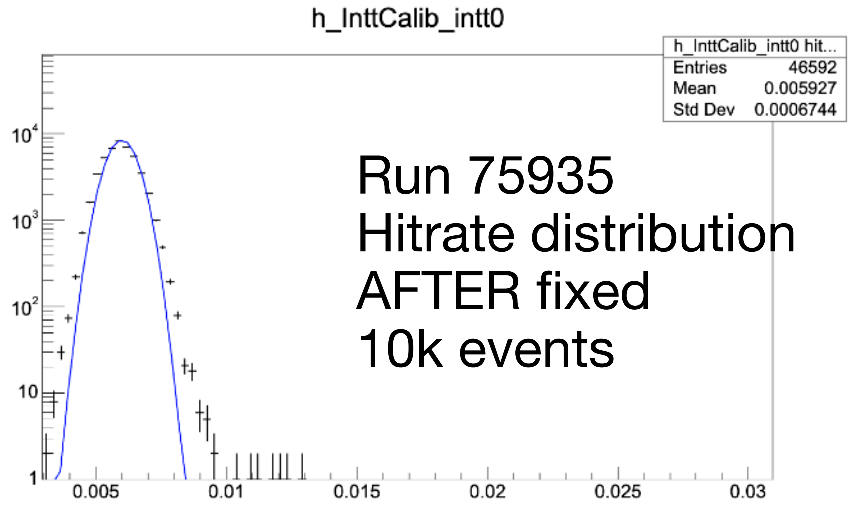


Run 75935
BCOQA
AFTER fixed
10k events



```
h_InttCalib_intt0 hit...
Entries          46592
Mean         0.0000514
Std Dev      0.0000089
```

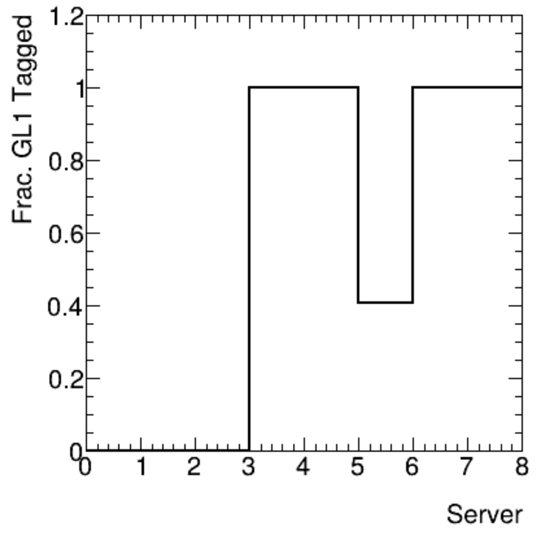
Run 75935
Hitrate distribution
BEFORE fixed



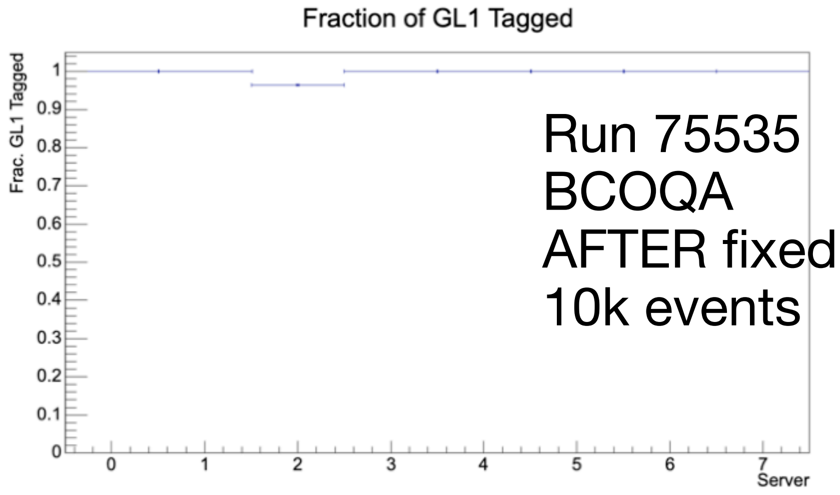
```
h_InttCalib_intt0 hit...
Entries          46592
Mean          0.005927
Std Dev      0.0006744
```

Run 75935
Hitrate distribution
AFTER fixed
10k events

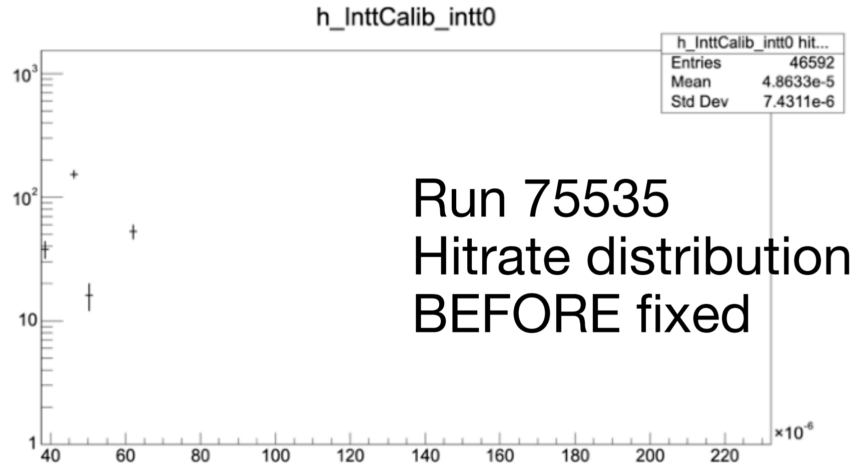
Before/After fix Run 75535



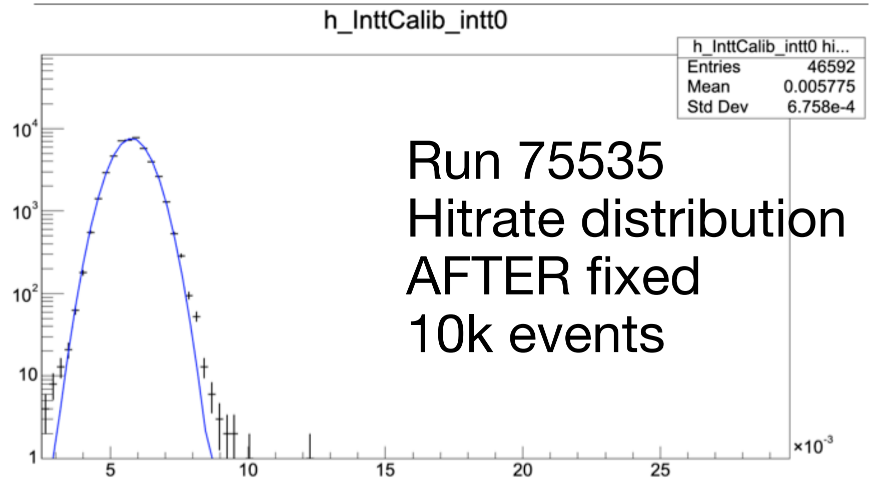
Run 75535
BCOQA
BEFORE fixed



Run 75535
BCOQA
AFTER fixed
10k events

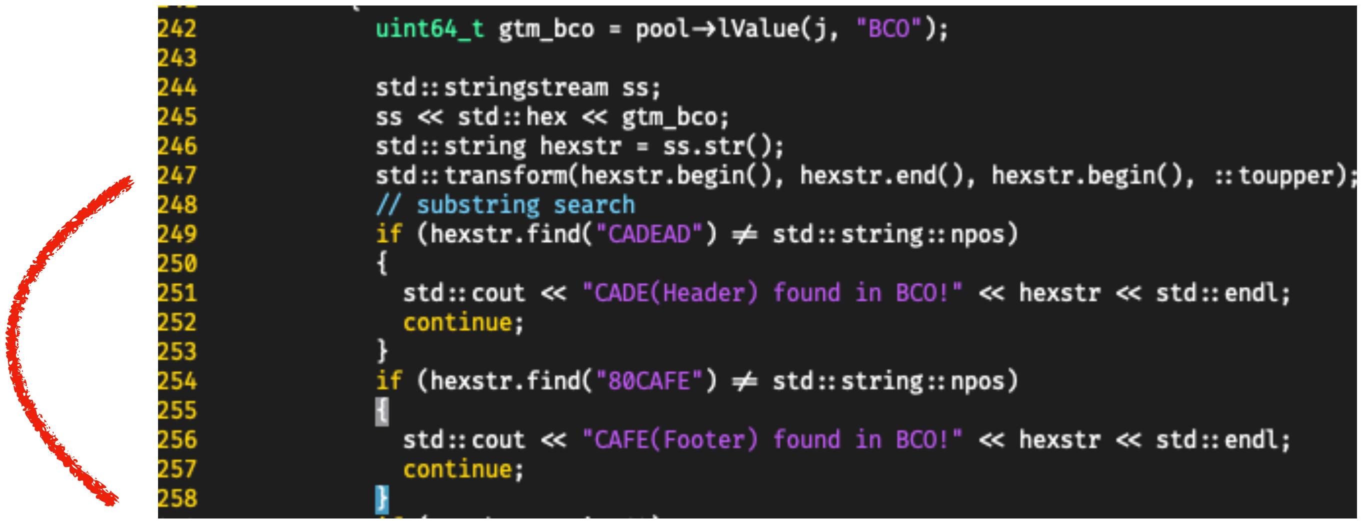


Run 75535
Hitrate distribution
BEFORE fixed



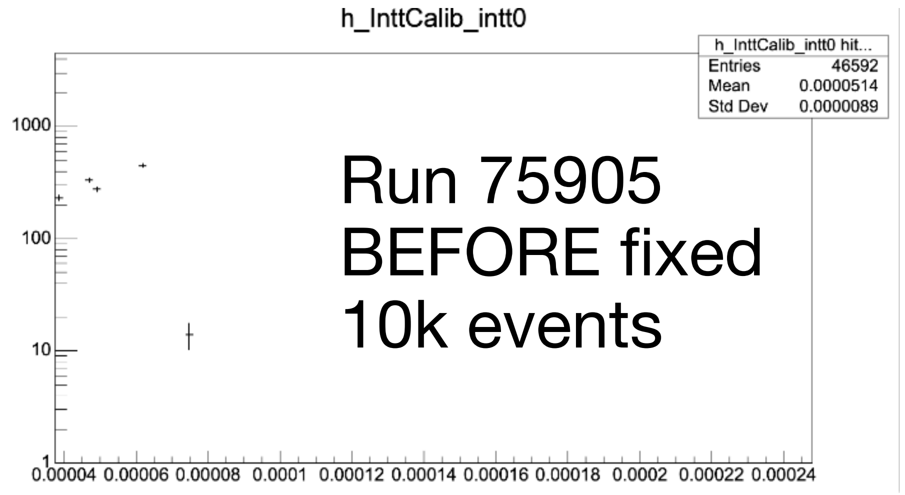
Run 75535
Hitrate distribution
AFTER fixed
10k events

Before/After fix Run 75935; Hitrate(after event combining)

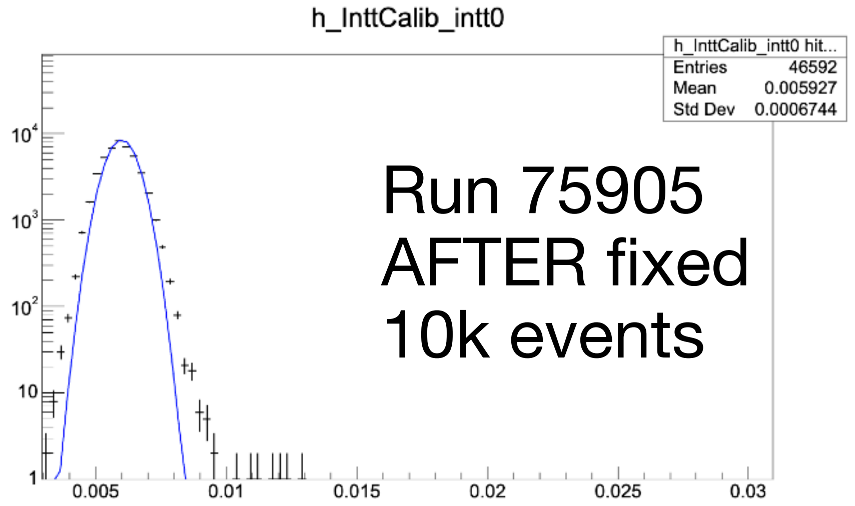


```
            uint64_t gtm_bco = pool->lValue(j, "BCO");

            std::stringstream ss;
            ss << std::hex << gtm_bco;
            std::string hexstr = ss.str();
            std::transform(hexstr.begin(), hexstr.end(), hexstr.begin(), ::toupper);
            // substring search
            if (hexstr.find("CADEAD") != std::string::npos)
            {
              std::cout << "CADE(Header) found in BCO!" << hexstr << std::endl;
              continue;
            }
            if (hexstr.find("80CAFE") != std::string::npos)
            {
              std::cout << "CAFE(Footer) found in BCO!" << hexstr << std::endl;
              continue;
            }
```

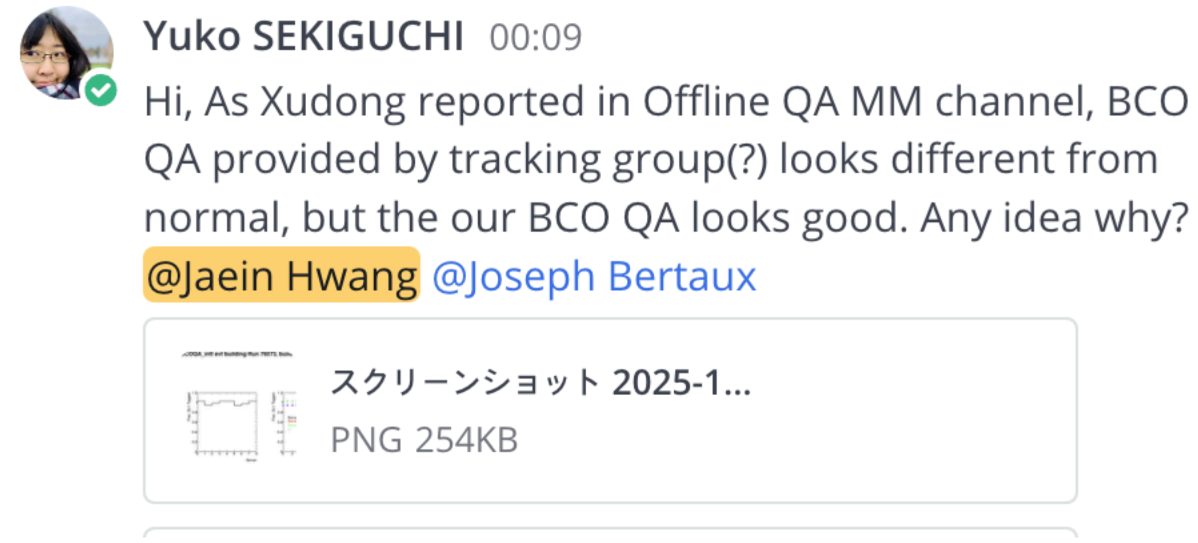


Run 75905
BEFORE fixed
10k events



Run 75905
AFTER fixed
10k events

Another issue..?

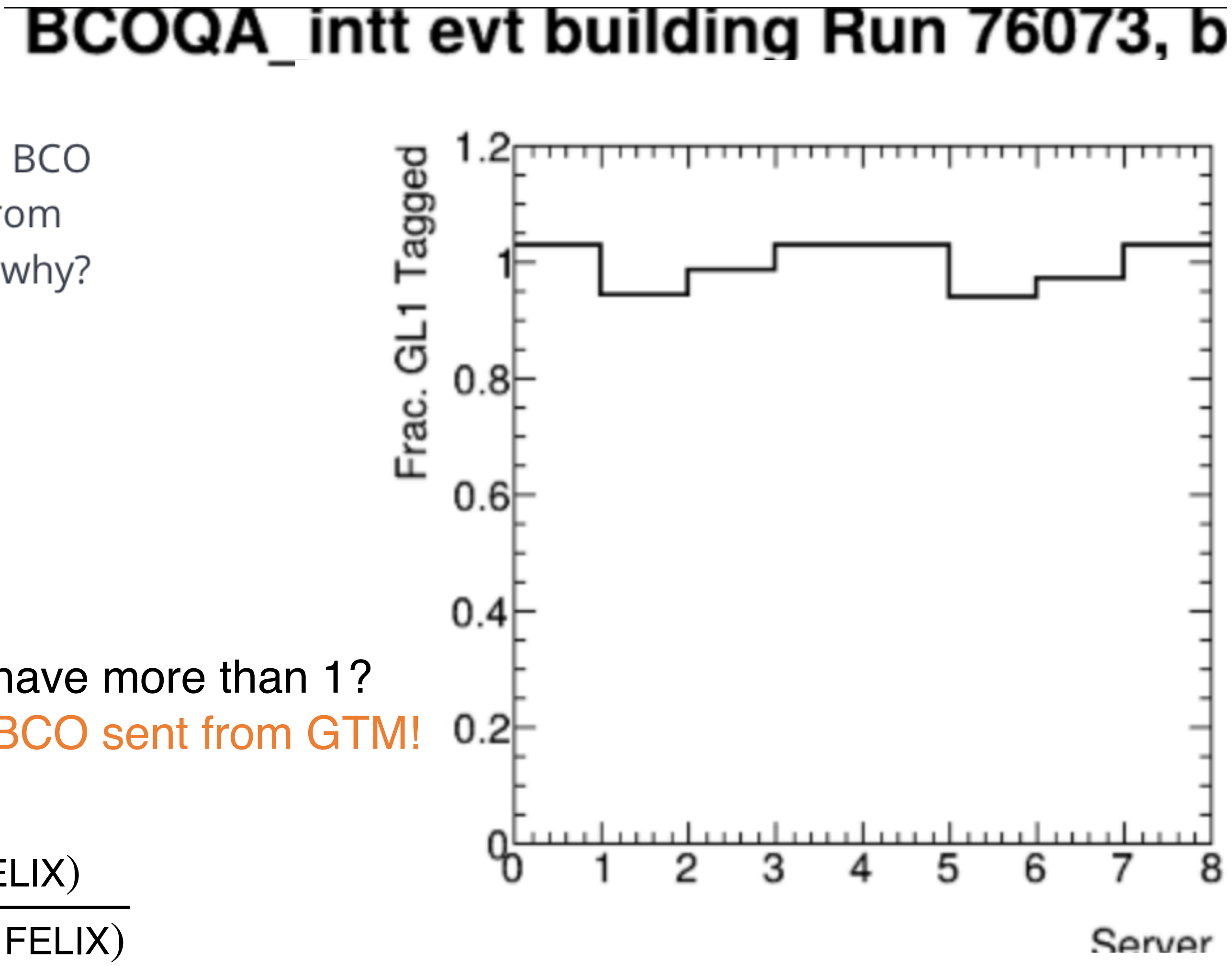


Yuko SEKIGUCHI 00:09

Hi, As Xudong reported in Offline QA MM channel, BCO QA provided by tracking group(?) looks different from normal, but the our BCO QA looks good. Any idea why? @Jaein Hwang @Joseph Bertaux

スクリーンショット 2025-1...

PNG 254KB



We have BCO drop at server 1,2,5 and 6, and other have more than 1?

We cannot have more BCO received by FELIX than BCO sent from GTM!

$$\text{Frac.GL1 tagged} = \frac{\text{(Number of GL1 BCO received by FELIX)}}{\text{(Number of Gl1 BCO sent from GTM to FELIX)}}$$

How BCO QA works?

Frac.GL1 tagged[server]

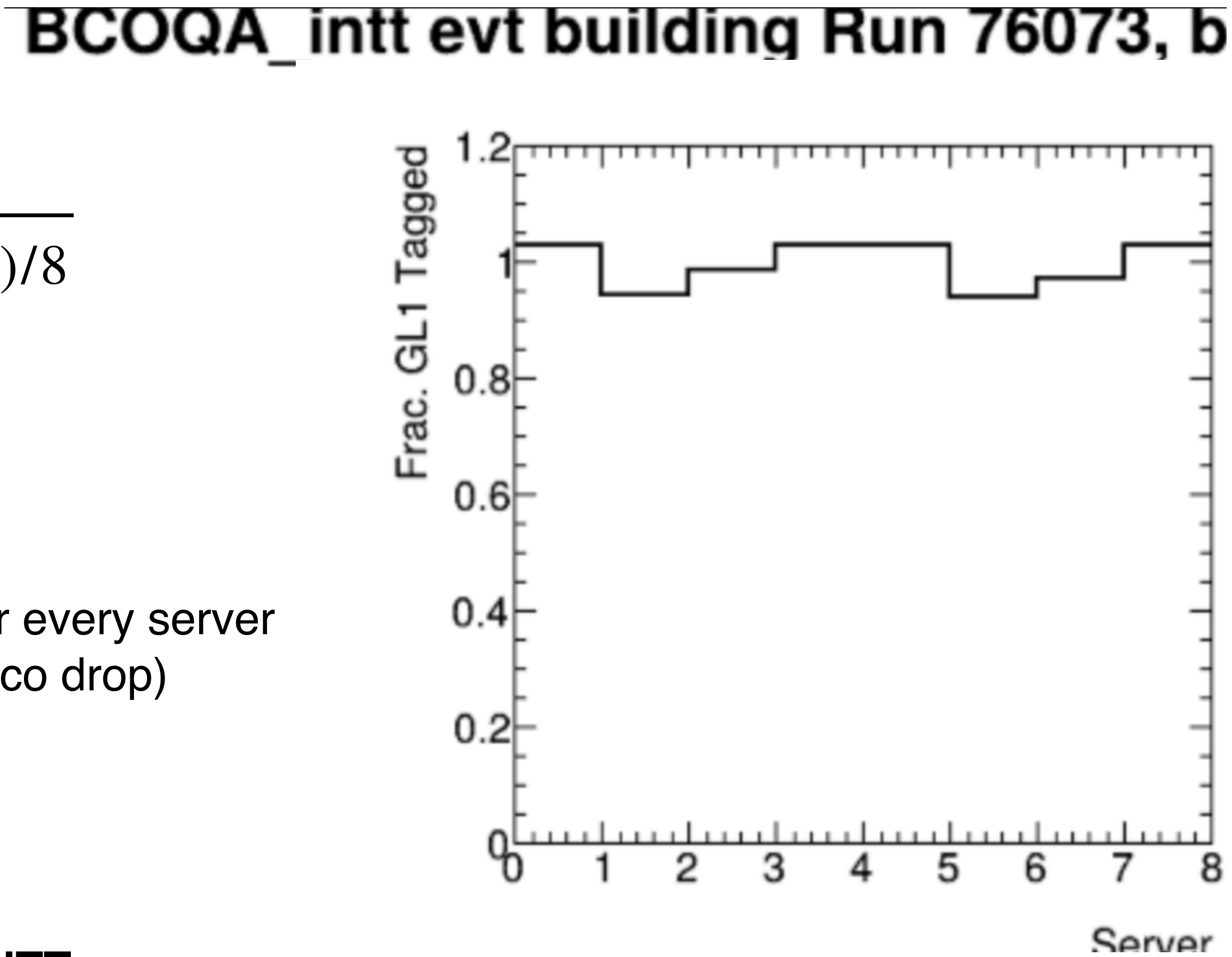
$$= \frac{\text{(Number of BCO received by FELIX[server])}}{(\sum_{i=0}^{7} \text{Number of BCO sent from GTM to FELIX[i]})/8}$$

If 1, GOOD! Else, something is happening..

If we have 10,000 events,
Number of BCO sent from GTM to INTT[i] = 10,000 for every server
Number of BCO received by FELIX[i] = 10,000 (if no bco drop)

$$= \frac{\text{(Number of BCO received by FELIX[server])}}{(80000)/8}$$

We must have same number of BCO for each INTT servers if we analyze FULL data



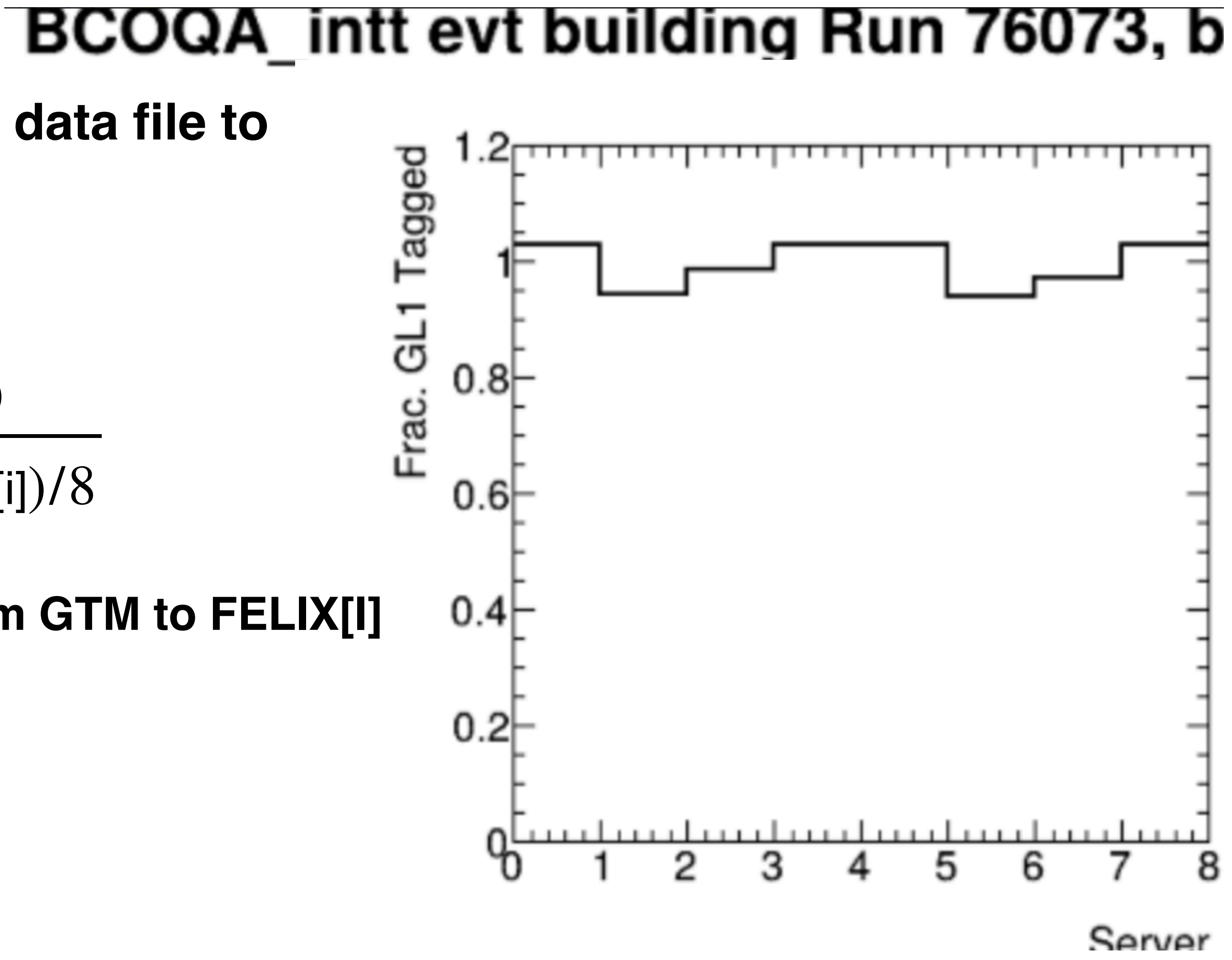
Issue on BCOQA module

But current auto-productions
We uses only 20GB for each subsystem raw data file to have quick offline QA plot available.

Frac.GL1 tagged[server]

$$= \frac{(\text{Number of BCO received by FELIX[server]})}{(\sum_{i=0}^{7} \text{Number of BCO sent from GTM to FELIX[i]})/8}$$

If we don't use full data, number of BCO sent from GTM to FELIX[I] cannot be identical for all FELIX[0-7]



Modification

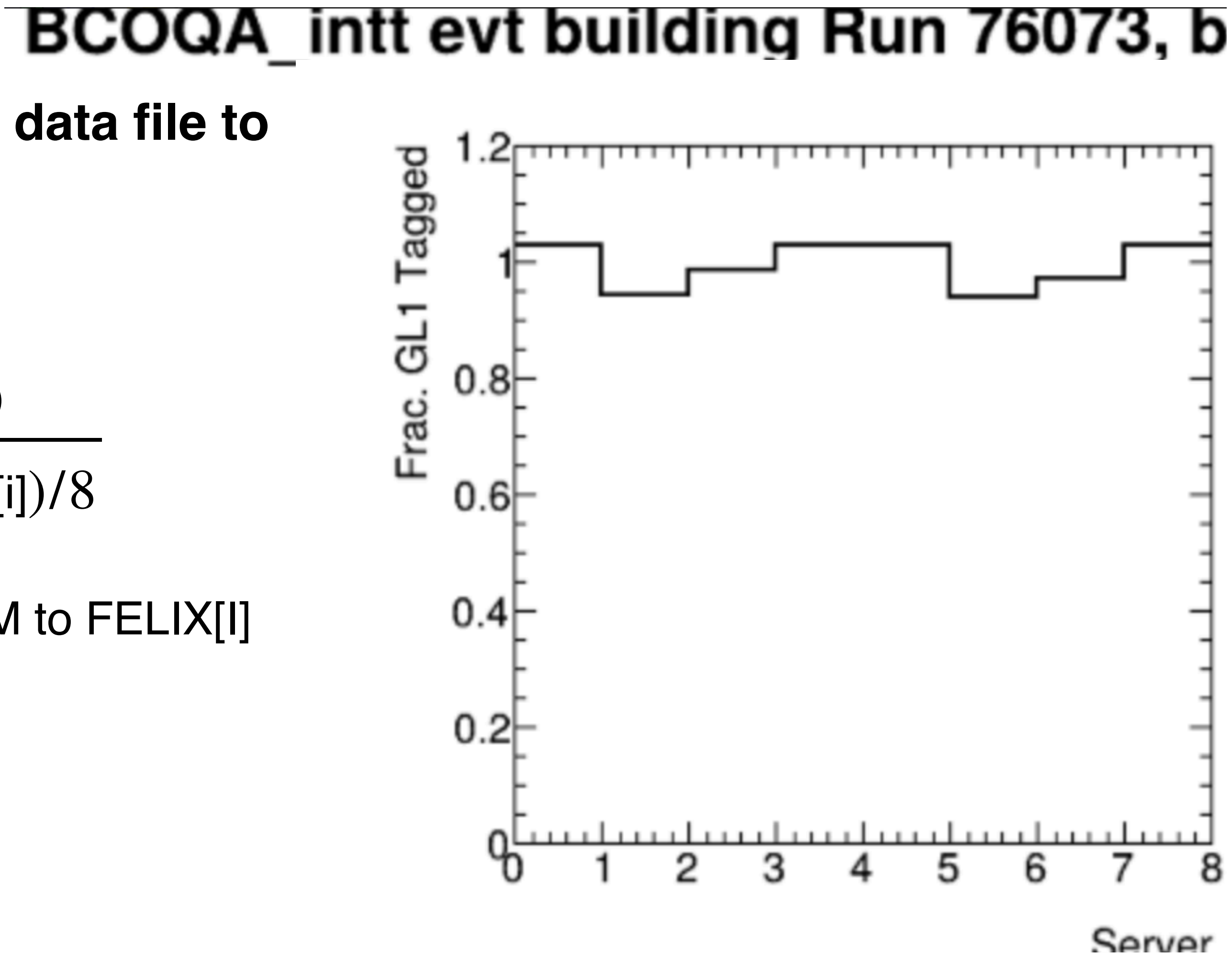
But current auto-productions
We uses only 20GB for each subsystem raw data file to have quick offline QA plot available.

Frac.GL1 tagged[server]

$$= \frac{(\text{Number of BCO received by FELIX[server]})}{(\sum_{i=0}^{7} \text{Number of BCO sent from GTM to FELIX[i]})/8}$$

If we don’t use full data, number of BCO sent by GTM to FELIX[I] cannot be identical for all FELIX[0-7]

Joe and myself detected that issue



Bug fixed

Frac.GL1 tagged[server]

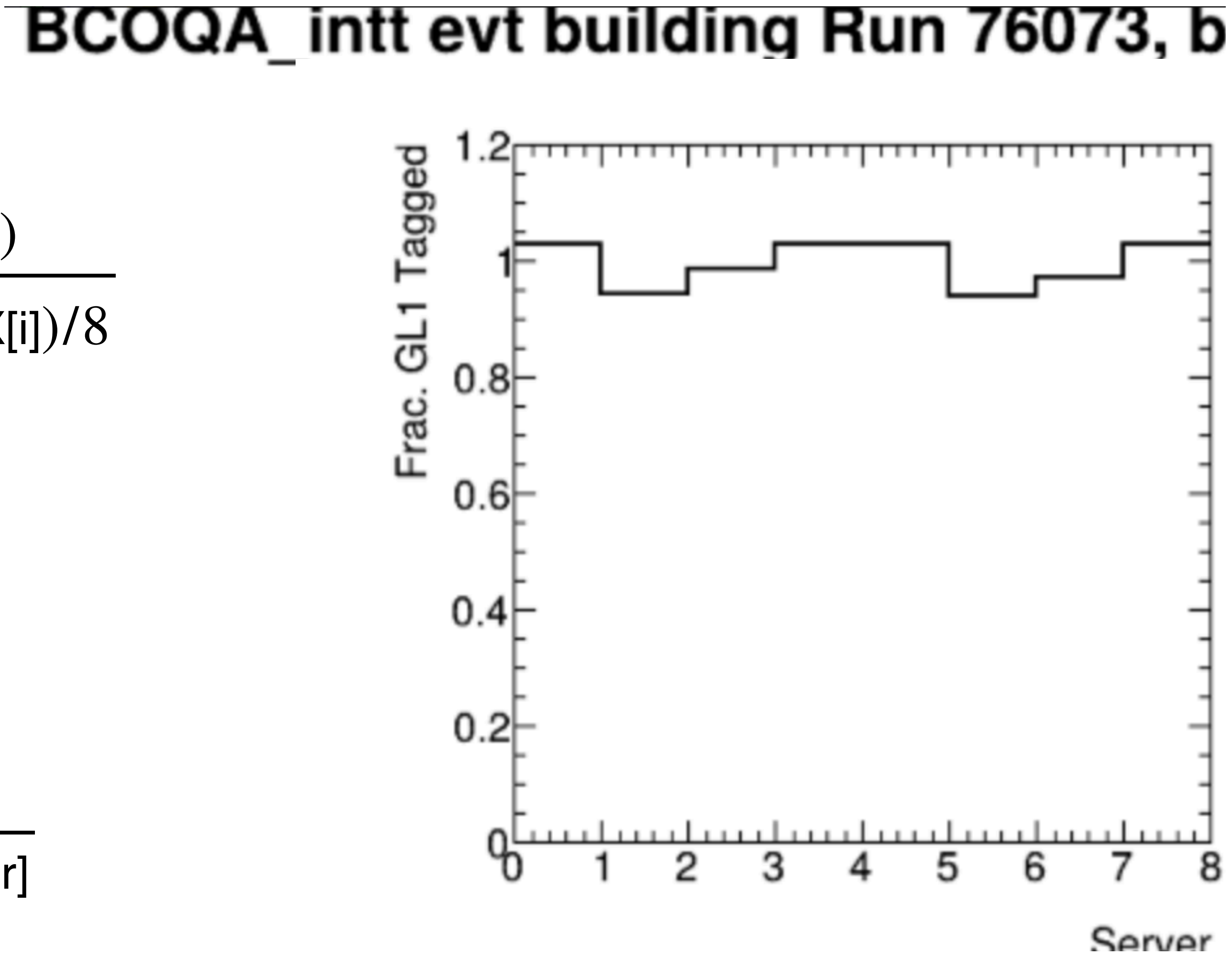
$$= \frac{(\text{Number of BCO received by FELIX[server]})}{(\sum_{i=0}^{7} \text{Number of BCO sent from GTM to FELIX[i]})/8}$$

Bug Fixed by Joe(Thanks!)
Doing QA FELIX server by server

Frac.GL1 tagged[server]

$$= \frac{(\text{Number of BCO received by FELIX[server]})}{(\text{Number of BCO sent from GTM to FELIX[server]}}$$

Now, QA code arrangement is ongoing. You may want to know how to check the BCO QA by yourself



How to Check Auto-production BCO QA

We (especially onsite crew) need to check QA files before QA webpage is available time-to-time

Location of QA files
/sphenix/data/data02/sphnxpro/production/run3auau/physics/new_nocdbtag_v001/
DST_STREAMING_EVENT_intt[0-7]/run_[lower_index]_[upper_index]/
HIST_DST_STREAMING_EVENT_intt[0-7]_run3auau_new_nocdbtag_v001-000{runner}-00000.root

For example, INTT7, renumber 76269
/sphenix/data/data02/sphnxpro/production/run3auau/physics/new_nocdbtag_v001/
DST_STREAMING_EVENT_intt7/run_00076200_00076300/hist/
HIST_DST_STREAMING_EVENT_intt7_run3auau_new_nocdbtag_v001-00076269-00000.root

How to Check Auto-production BCO QA

We (especially onsite crew) need to check QA files before QA webpage is available time-to-time

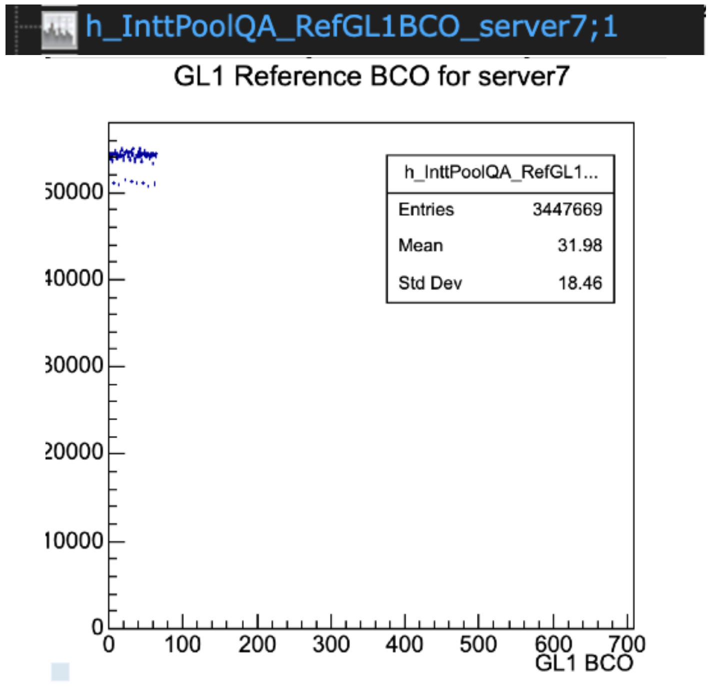
Location of QA files
/sphenix/data/data02/sphnxpro/production/run3auau/physics/new_nocdbtag_v001/
DST_STREAMING_EVENT_intt[0-7]/run_[lower_index]_[upper_index]/
HIST_DST_STREAMING_EVENT_intt[0-7]_run3auau_new_nocdbtag_v001-000{runner}-00000.root

For example, INTT7, renumber 76269
/sphenix/data/data02/sphnxpro/production/run3auau/physics/new_nocdbtag_v001/
DST_STREAMING_EVENT_intt7/run_00076200_00076300/hist/
HIST_DST_STREAMING_EVENT_intt7_run3auau_new_nocdbtag_v001-00076269-00000.root

How to Check Auto-production BCO QA

Example : HIST_DST_STREAMING_EVENT_intt7_run3auau_new_nocdbtag_v001-00076417-00000.root

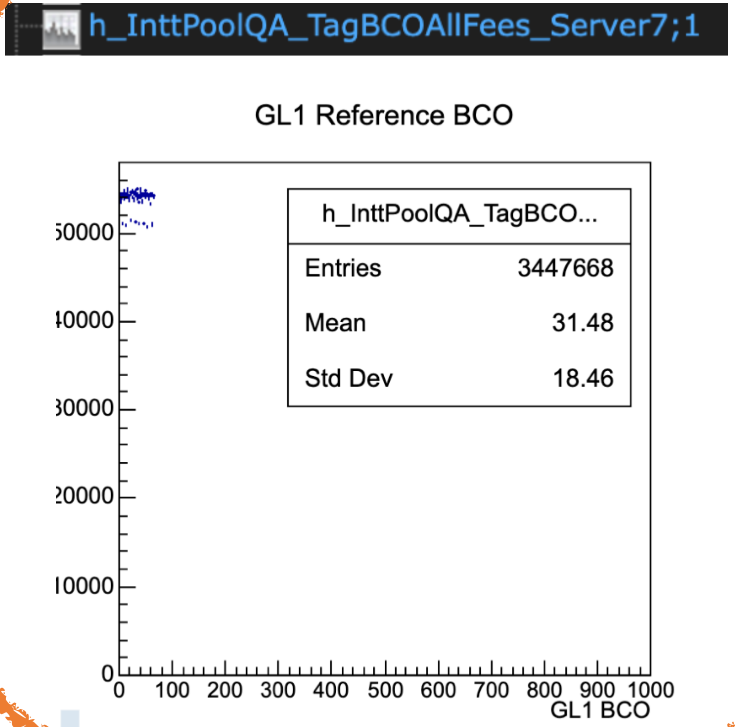
# of BCO sent from GTM to FELIX7


h_InttPoolQA_RefGL1BCO_server7;1

GL1 Reference BCO for server7

h_InttPoolQA_RefGL1...	
Entries	3447669
Mean	31.98
Std Dev	18.46

# of BCO received by FELIX7


h_InttPoolQA_TagBCOAllFees_Server7;1

GL1 Reference BCO

h_InttPoolQA_TagBCO...	
Entries	3447668
Mean	31.48
Std Dev	18.46

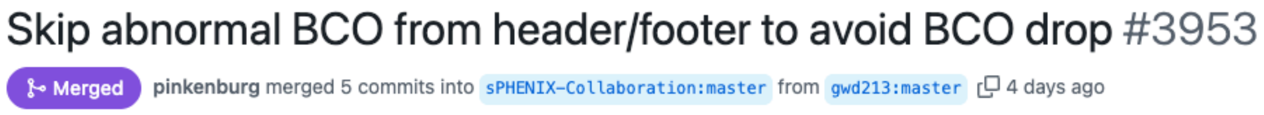
$$\text{Frac.GL1 tagged[server]} = \frac{\text{(Number of BCO received by FELIX[server])}}{\text{(Number of BCO sent from GTM to FELIX[server]}} = \frac{3{,}447{,}668}{3{,}447{,}669} > 0.99999$$

Summary

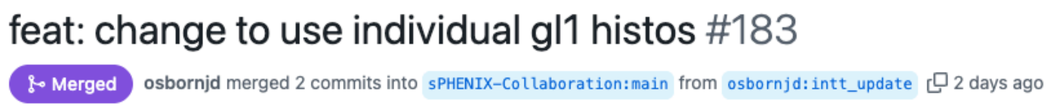
Both PR merged.
Due to the urgency of the issue, PR has been merged before presenting at the INTT meeting, but I mentioned it on Mattermost in advance, I hope it's fine :)

LINK

LINK


Skip abnormal BCO from header/footer to avoid BCO drop #3953

Merged pinkenburg merged 5 commits into `sPHENIX-Collaboration:master` from `gwd213:master` 4 days ago


feat: change to use individual gl1 histos #183

Merged osbornjd merged 2 commits into `sPHENIX-Collaboration:main` from `osbornjd:intt_update` 2 days ago

Thanks to Joseph for helping make the code run much faster
Thnaks to Joe for figuring out the problem and updating the BCOQA

- Issues have been addressed.

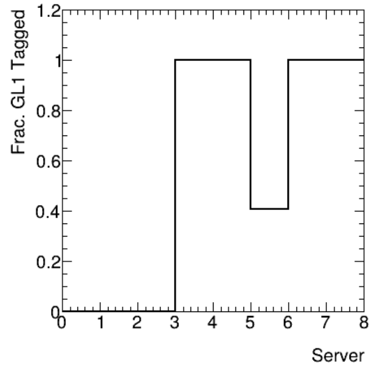
- Good to understand why it happened.(discussion with DAQ expert needed)

- If you are onsite and need to check Offline QA in advance, please use the way mentioned today's meeting

BACKUP

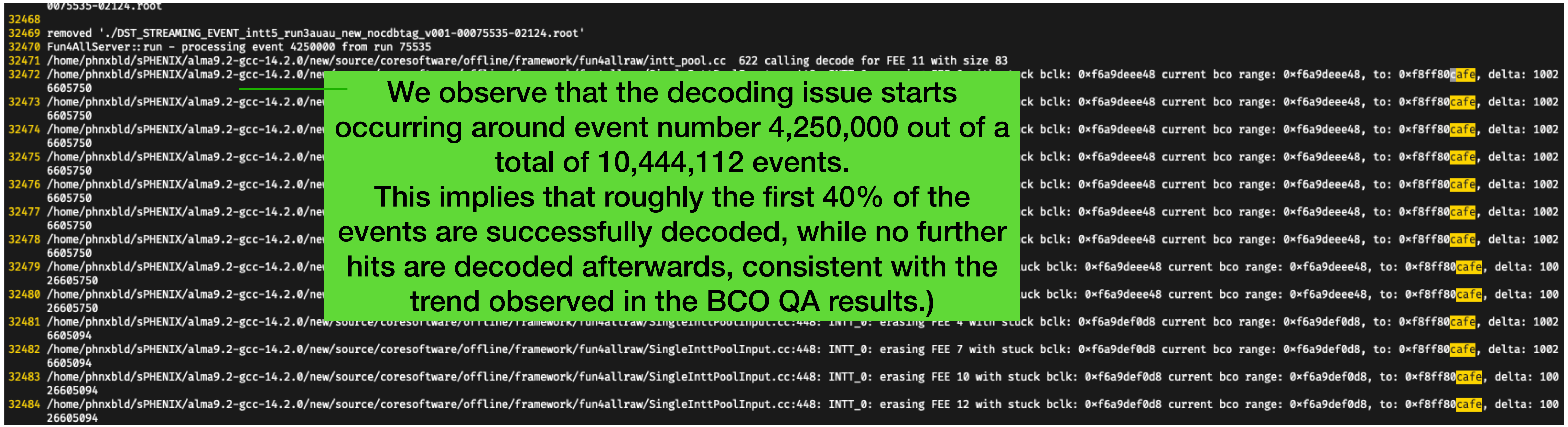
BACKUP

Case for having footer at the middle of the run RUN 75535 INTT5



Check production log,
/sphenix/data/data02/sphnxpro/production/run3auau/physics/new_nocdbtag_v001/DST_STREAMING_EVENT_intt5/run_00075500_00075600/log/DST_STREAMING_DST_STREAMING_EVENT_intt5_run3auau_new_nocdbtag_v001-00075535-00000.out

RUN 75535 INTT5



```
     0075535-02124.root

removed './DST_STREAMING_EVENT_intt5_run3auau_new_nocdbtag_v001-00075535-02124.root'
Fun4AllServer::run - processing event 4250000 from run 75535
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new/source/coresoftware/offline/framework/fun4allraw/intt_pool.cc  622 calling decode for FEE 11 with size 83
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new                                        ck bclk: 0xf6a9deee48 current bco range: 0xf6a9deee48, to: 0xf8ff80cafe, delta: 1002
6605750
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new                                        ck bclk: 0xf6a9deee48 current bco range: 0xf6a9deee48, to: 0xf8ff80cafe, delta: 1002
6605750
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new                                        ck bclk: 0xf6a9deee48 current bco range: 0xf6a9deee48, to: 0xf8ff80cafe, delta: 1002
6605750
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new                                        ck bclk: 0xf6a9deee48 current bco range: 0xf6a9deee48, to: 0xf8ff80cafe, delta: 1002
6605750
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new                                        ck bclk: 0xf6a9deee48 current bco range: 0xf6a9deee48, to: 0xf8ff80cafe, delta: 1002
6605750
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new                                        ck bclk: 0xf6a9deee48 current bco range: 0xf6a9deee48, to: 0xf8ff80cafe, delta: 1002
6605750
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new                                        ck bclk: 0xf6a9deee48 current bco range: 0xf6a9deee48, to: 0xf8ff80cafe, delta: 1002
6605750
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new                                        uck bclk: 0xf6a9deee48 current bco range: 0xf6a9deee48, to: 0xf8ff80cafe, delta: 100
26605750
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new                                        uck bclk: 0xf6a9deee48 current bco range: 0xf6a9deee48, to: 0xf8ff80cafe, delta: 100
26605750
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new/source/coresoftware/offline/framework/fun4allraw/SingleInttPoolInput.cc:448: INTT_0: erasing FEE 4 with stuck bclk: 0xf6a9def0d8 current bco range: 0xf6a9def0d8, to: 0xf8ff80cafe, delta: 1002
6605094
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new/source/coresoftware/offline/framework/fun4allraw/SingleInttPoolInput.cc:448: INTT_0: erasing FEE 7 with stuck bclk: 0xf6a9def0d8 current bco range: 0xf6a9def0d8, to: 0xf8ff80cafe, delta: 1002
6605094
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new/source/coresoftware/offline/framework/fun4allraw/SingleInttPoolInput.cc:448: INTT_0: erasing FEE 10 with stuck bclk: 0xf6a9def0d8 current bco range: 0xf6a9def0d8, to: 0xf8ff80cafe, delta: 100
26605094
/home/phnxbld/sPHENIX/alma9.2-gcc-14.2.0/new/source/coresoftware/offline/framework/fun4allraw/SingleInttPoolInput.cc:448: INTT_0: erasing FEE 12 with stuck bclk: 0xf6a9def0d8 current bco range: 0xf6a9def0d8, to: 0xf8ff80cafe, delta: 100
26605094
```

We observe that the decoding issue starts occurring around event number 4,250,000 out of a total of 10,444,112 events.
This implies that roughly the first 40% of the events are successfully decoded, while no further hits are decoded afterwards, consistent with the trend observed in the BCO QA results.)

Found that at the middle of the data taking, we have 0xf8ff80cafe BCO
As far as I know, 0xcafeff80 type is the footer word, which shouldn't be in the list of hits