

Update for BCO calibration

Jaein Hwang
on behalf of INTT students
2024/12/13



BCO calibration files

BCO CDBTTree are placed in CDB Database for both trigger and streaming mode

Problem : We have used BCO Diff cut [peak+1,peak-1] **whenever we find BCO CDBTTree**



afrawley 6일 전

As of yesterday I see very few INTT clusters in the TrackResiduals clustertree, and hence most silicon seeds have only 3 or 4 clusters. I do not have any local modifications to clustering or silicon seeding. Is anyone else seeing this?

@trakingsoftware mattermost : Problem detected right after CDBTTree placed in.

We don't want to use BCO filter for extended readout / streaming readout analysis at least it requires more study on it.

Update on software



Problem : We have used BCO Diff cut [peak+1,peak-1] **whenever we find BCO CDBTTree**

1) BCODiff cut is off by default. Needed one additional function to turn it on.

2) Timebucket offset is loaded from BCO CDBTTree for trigger mode.

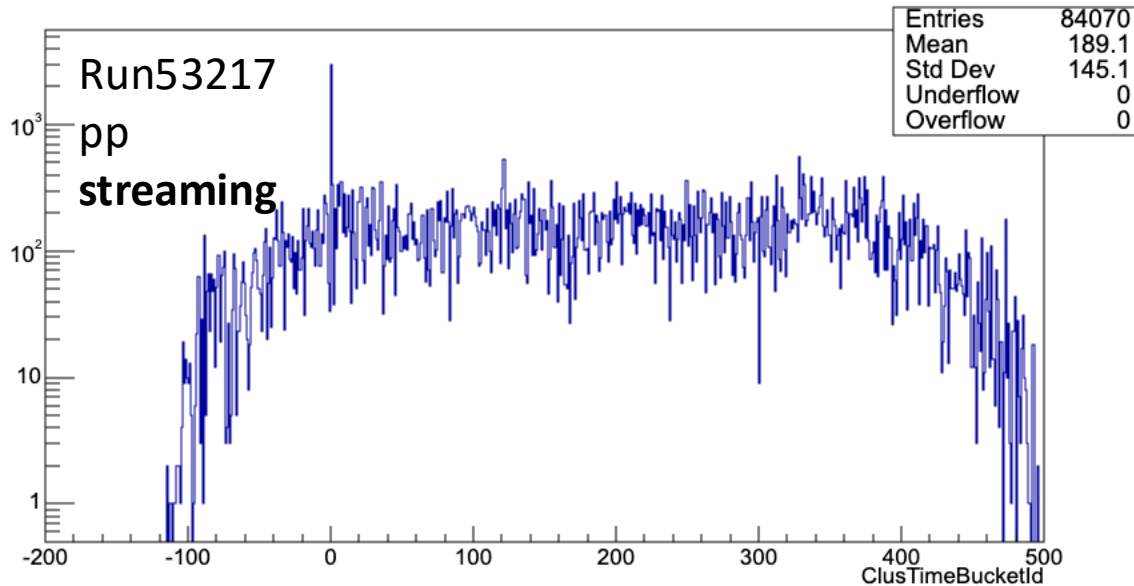
(streaming mode always need 23 offset value for whole runs, loading CDB is not needed)

inttunpacker / without PR update , without BCO CDB

```

149 auto inttunpacker = new InttCombinedRawDataDecoder;
150 inttunpacker->Verbosity(0);
151 InttOdbcQuery query;
152 query.Query(runnumber);
153 bool isStreaming = query.IsStreaming();
154 if(isStreaming) {std::cout<<"HERE ! : Streaming mode"<<std::endl;}
155 else {std::cout<<"HERE ! : Non-Streaming mode"<<std::endl;}
156 inttunpacker->runInttStandalone(false);
157 inttunpacker->writeInttEventHeader(true);
158 inttunpacker->set_triggeredMode(!isStreaming);
159 inttunpacker->SetCalibBCO("", InttCombinedRawDataDecoder::FILE);
160 inttunpacker->SetCalibDAC(dac_file, InttCombinedRawDataDecoder::FILE);
161 // inttunpacker->set_bcoFilter(true);
162 // inttunpacker->LoadHotChannelMapLocal(hotchannel_file);
163 // inttunpacker->runInttStandalone(true);
164 se->registerSubsystem(inttunpacker);
    
```

INTT cluster TimeBucket distribution



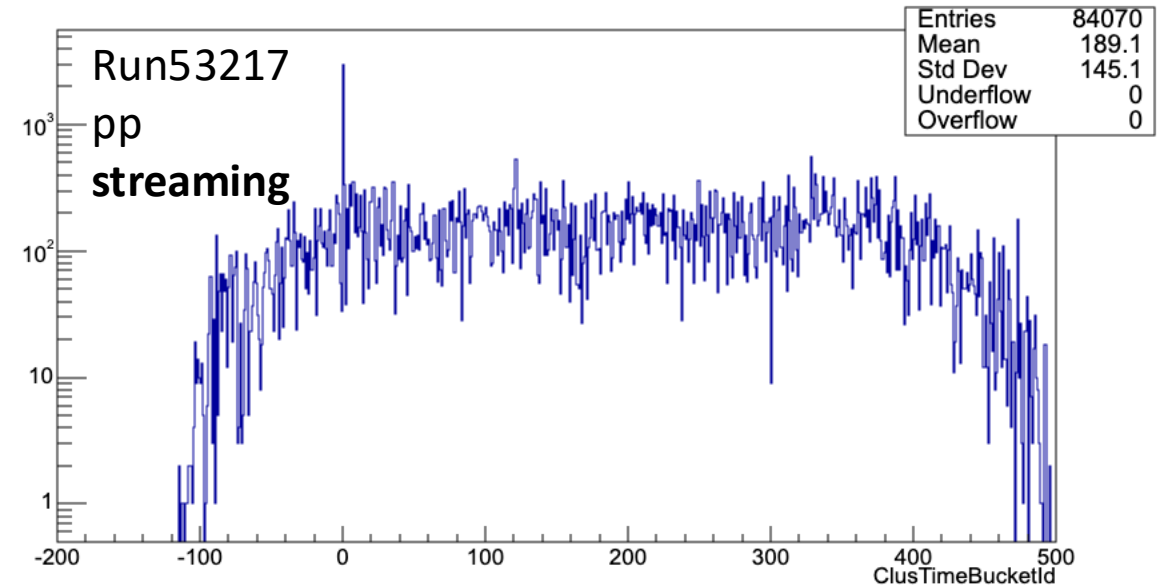
Inttunpacker / with PR update , with BCO CDB



```

149 auto inttunpacker = new InttCombinedRawDataDecoder;
150 inttunpacker->Verbosity(0);
151 InttOdbcQuery query;
152 query.Query(runnumber);
153 bool isStreaming = query.IsStreaming();
154 if(isStreaming) {std::cout<<"HERE ! : Streaming mode"<<std::endl;}
155 else {std::cout<<"HERE ! : Non-Streaming mode"<<std::endl;}
156 inttunpacker->runInttStandalone(false);
157 inttunpacker->writeInttEventHeader(true);
158 inttunpacker->set_triggeredMode(!isStreaming);
159 // inttunpacker->SetCalibBCO("", InttCombinedRawDataDecoder::FILE);
160 inttunpacker->SetCalibDAC(dac_file, InttCombinedRawDataDecoder::FILE);
161 // inttunpacker->set_bcoFilter(true);
162 // inttunpacker->LoadHotChannelMapLocal(hotchannel_file);
163 // inttunpacker->runInttStandalone(true);
164 se->registerSubsystem(inttunpacker);
    
```

INTT cluster TimeBucket distribution



Offset 23 has been set as default BCO offset in INTT unpacker before new PR published
streaming readout TimeBucket is **identical** as expected(without/with new PR)

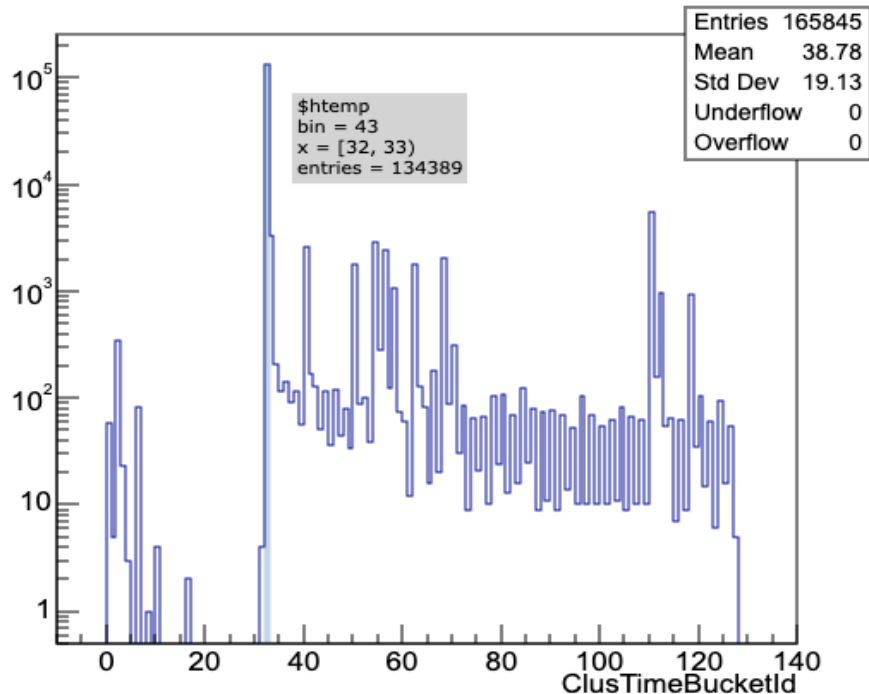
inttunpacker / without PR update , without BCO CDB

```

149 auto inttunpacker = new InttCombinedRawDataDecoder;
150 inttunpacker->Verbosity(0);
151 InttOdbcQuery query;
152 query.Query(runnumber);
153 bool isStreaming = query.IsStreaming();
154 if(isStreaming) {std::cout<<"HERE ! : Streaming mode"<<std::endl;}
155 else {std::cout<<"HERE ! : Non-Streaming mode"<<std::endl;}
156 inttunpacker->runInttStandalone(false);
157 inttunpacker->writeInttEventHeader(true);
158 inttunpacker->set_triggeredMode(!isStreaming);
159 inttunpacker->SetCalibBCO("", InttCombinedRawDataDecoder::FILE);
160 inttunpacker->SetCalibDAC(dac_file, InttCombinedRawDataDecoder::FILE);
161 // inttunpacker->set_bcoFilter(true);
162 // inttunpacker->LoadHotChannelMapLocal(hotchannel_file);
163 // inttunpacker->runInttStandalone(true);
164 se->registerSubsystem(inttunpacker);
    
```

INTT cluster TimeBucket distribution

Run54280
AuAu
Trigger



With the BCO calibrations Timebucket of gl1 crossing associated hit properly set to 0

Inttunpacker / with PR update , with BCO CDB

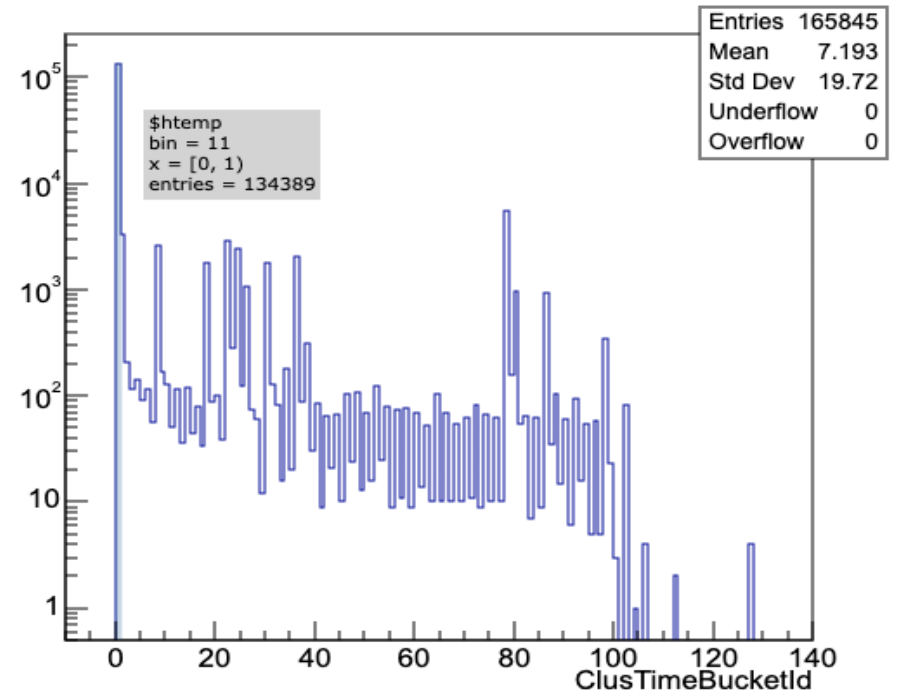


```

149 auto inttunpacker = new InttCombinedRawDataDecoder;
150 inttunpacker->Verbosity(0);
151 InttOdbcQuery query;
152 query.Query(runnumber);
153 bool isStreaming = query.IsStreaming();
154 if(isStreaming) {std::cout<<"HERE ! : Streaming mode"<<std::endl;}
155 else {std::cout<<"HERE ! : Non-Streaming mode"<<std::endl;}
156 inttunpacker->runInttStandalone(false);
157 inttunpacker->writeInttEventHeader(true);
158 inttunpacker->set_triggeredMode(!isStreaming);
159 // inttunpacker->SetCalibBCO("", InttCombinedRawDataDecoder::FILE);
160 inttunpacker->SetCalibDAC(dac_file, InttCombinedRawDataDecoder::FILE);
161 // inttunpacker->set_bcoFilter(true);
162 // inttunpacker->LoadHotChannelMapLocal(hotchannel_file);
163 // inttunpacker->runInttStandalone(true);
164 se->registerSubsystem(inttunpacker);
    
```

Run54280 INTT cluster TimeBucket distribution

AuAu
Trigger



One additional step needed



Since InttRawDataDecorder need to process different steps depending on data taking mode(streaming / trigger)

```
149 auto inttunpacker = new InttCombinedRawDataDecoder;
150 inttunpacker->Verbosity(0);
151 InttOdbcQuery query;
152 query.Query(runnumber);
153 bool isStreaming = query.IsStreaming();
154 if(isStreaming) {std::cout<<"HERE ! : Streaming mode"<<std::endl;}
155 else {std::cout<<"HERE ! : Non-Streaming mode"<<std::endl;}
156 inttunpacker->runInttStandalone(false);
157 inttunpacker->writeInttEventHeader(true);
158 inttunpacker->set_triggeredMode(!isStreaming);
159 inttunpacker->SetCalibBCO("", InttCombinedRawDataDecoder::FILE);
160 inttunpacker->SetCalibDAC(dac_file, InttCombinedRawDataDecoder::FILE);
161 // inttunpacker->set_bcoFilter(true);
162 // inttunpacker->LoadHotChannelMapLocal(hotchannel_file);
163 // inttunpacker->runInttStandalone(true);
164 se->registerSubsystem(inttunpacker);
```

We have function to check our runmode from database

Want to discuss about : Where is the BEST place to apply mode selection in our production stream? Probably Intt_HitUnpacking() function? https://github.com/SPHENIX-Collaboration/macros/blob/master/common/Trkr_Clustering.C#L71-L85

```
71 void Intt_HitUnpacking(const std::string& server="")
72 {
73     int verbosity = std::max(Enable::VERBOSITY, Enable::INTT_VERBOSITY);
74     Fun4AllServer* se = Fun4AllServer::instance();
75
76     auto inttunpacker = new InttCombinedRawDataDecoder("InttCombinedRawDataDecoder"+server);
77     inttunpacker->Verbosity(verbosity);
78     inttunpacker->LoadHotChannelMapRemote("INTT_HotMap");
79     if(server.length() > 0)
80     {
81         inttunpacker->useRawHitNodeName("INTTRAWHIT_" + server);
82     }
83     se->registerSubsystem(inttunpacker);
84 }
```

inttunpacker / without PR update , without BCO CDB

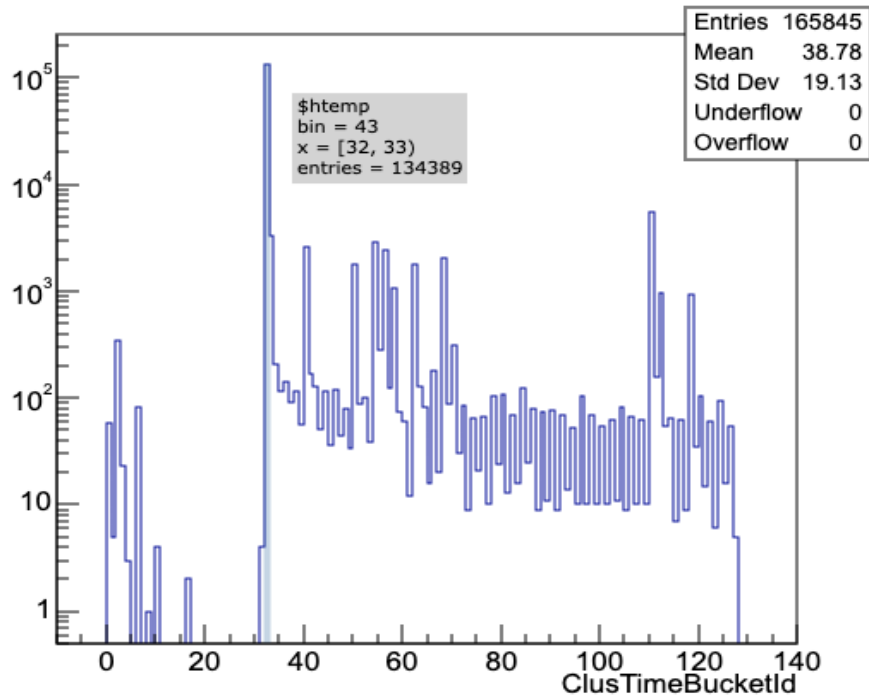
```

149 auto inttunpacker = new InttCombinedRawDataDecoder;
150 inttunpacker->Verbosity(0);
151 InttOdbcQuery query;
152 query.Query(runnumber);
153 bool isStreaming = query.IsStreaming();
154 if(isStreaming) {std::cout<<"HERE ! : Streaming mode"<<std::endl;}
155 else {std::cout<<"HERE ! : Non-Streaming mode"<<std::endl;}
156 inttunpacker->runInttStandalone(false);
157 inttunpacker->writeInttEventHeader(true);
158 inttunpacker->set_triggeredMode(!isStreaming);
159 inttunpacker->SetCalibBCO("", InttCombinedRawDataDecoder::FILE);
160 inttunpacker->SetCalibDAC(dac_file, InttCombinedRawDataDecoder::FILE);
161 // inttunpacker->set_bcoFilter(true);
162 // inttunpacker->LoadHotChannelMapLocal(hotchannel_file);
163 // inttunpacker->runInttStandalone(true);
164 se->registerSubsystem(inttunpacker);

```

INTT cluster TimeBucket distribution

Run54280
AuAu
Trigger



With the BCO calibrations Timebucket of gl1 crossing associated hit properly set to 0

Inttunpacker / with PR update , with BCO CDB



```

149 auto inttunpacker = new InttCombinedRawDataDecoder;
150 inttunpacker->Verbosity(0);
151 InttOdbcQuery query;
152 query.Query(runnumber);
153 bool isStreaming = query.IsStreaming();
154 if(isStreaming) {std::cout<<"HERE ! : Streaming mode"<<std::endl;}
155 else {std::cout<<"HERE ! : Non-Streaming mode"<<std::endl;}
156 inttunpacker->runInttStandalone(false);
157 inttunpacker->writeInttEventHeader(true);
158 inttunpacker->set_triggeredMode(!isStreaming);
159 // inttunpacker->SetCalibBCO("", InttCombinedRawDataDecoder::FILE);
160 inttunpacker->SetCalibDAC(dac_file, InttCombinedRawDataDecoder::FILE);
161 // inttunpacker->set_bcoFilter(true);
162 // inttunpacker->LoadHotChannelMapLocal(hotchannel_file);
163 // inttunpacker->runInttStandalone(true);
164 se->registerSubsystem(inttunpacker);

```

Run54280 INTT cluster TimeBucket distribution

Run54280
AuAu
Trigger

