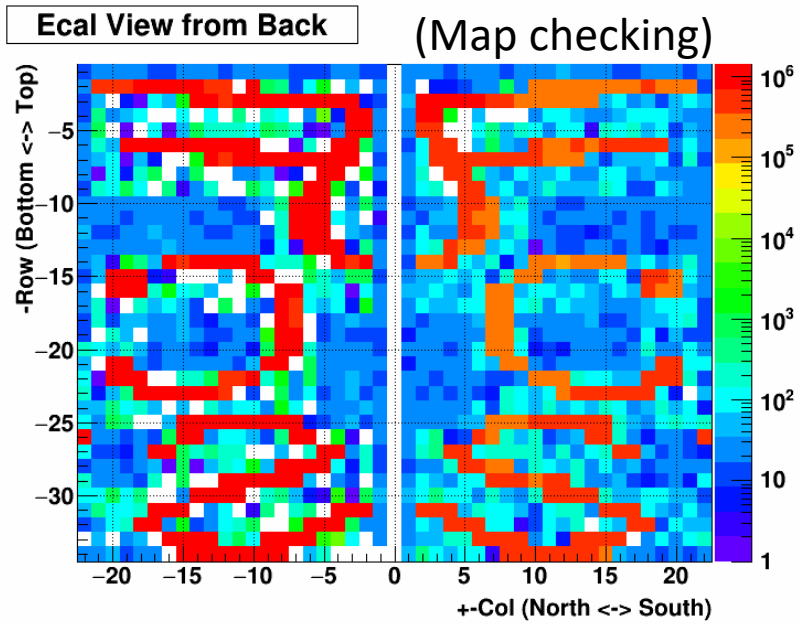
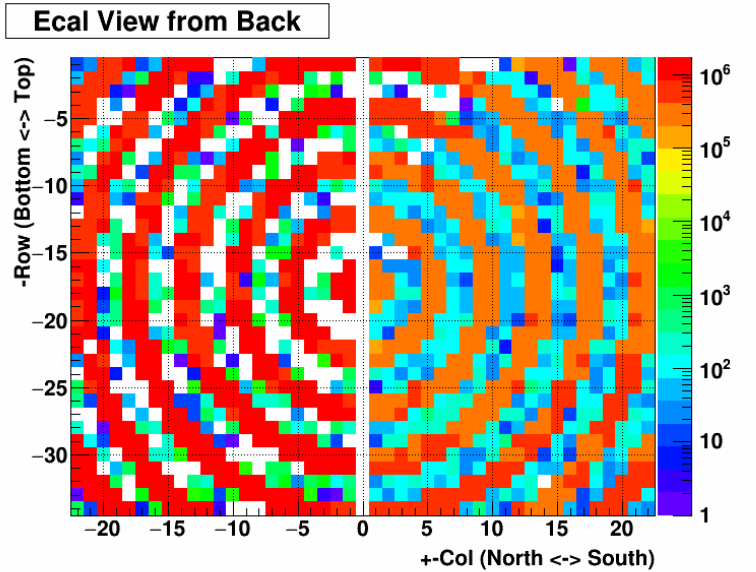
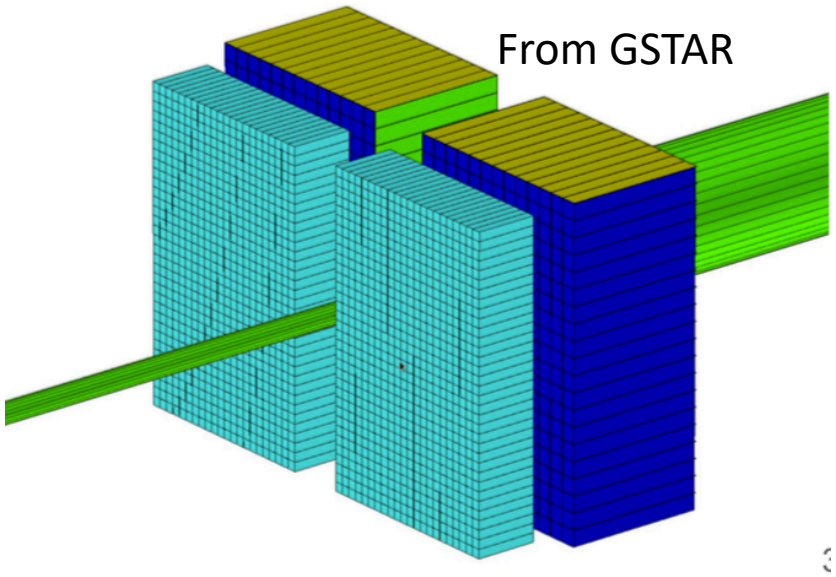
FCS Software

From Online QA
(Map checking)





From GSTAR

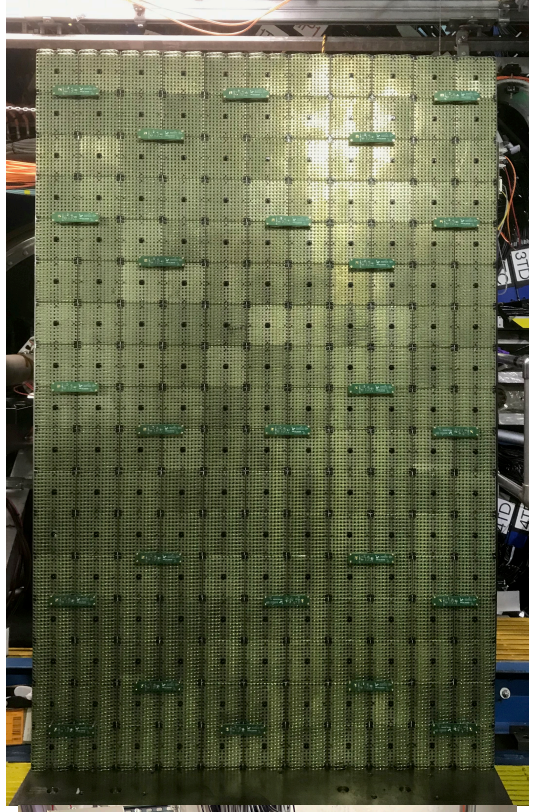


Ecal 34x22x2 = 1496ch
Hcal 20x13x2 = 520ch
EPD-W as PreShower (384ch of separate readout)

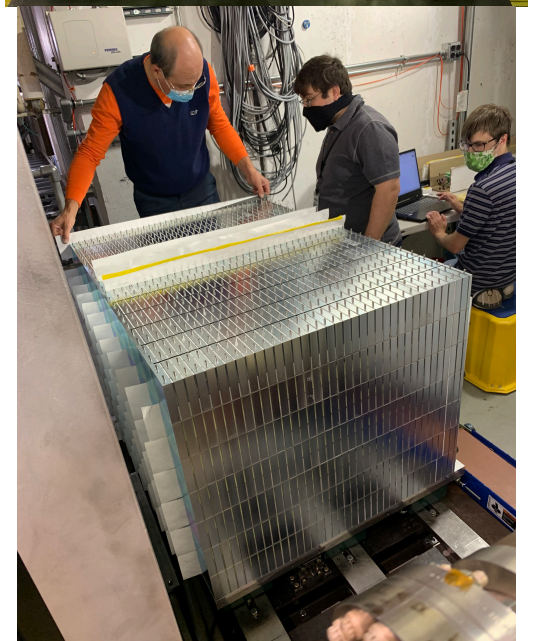
Akio Ogawa

2021 02 12

Ecal



Hcall

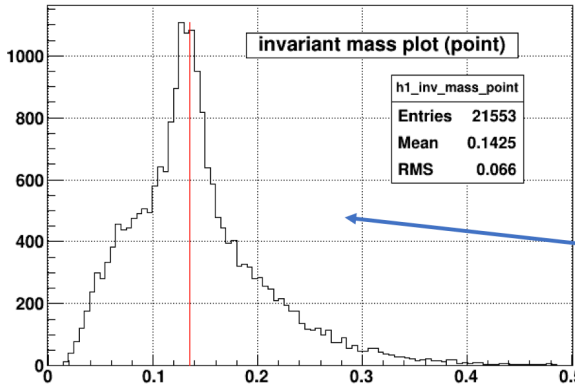


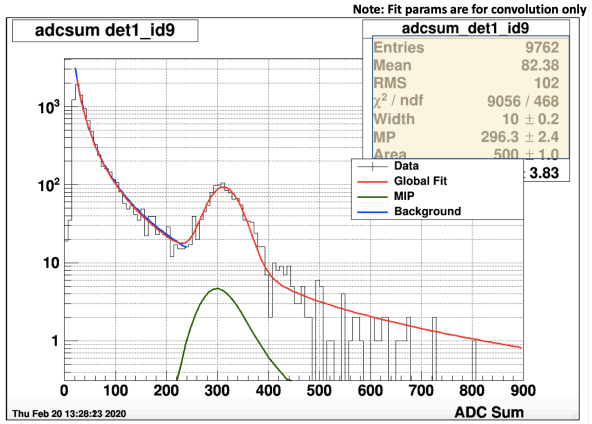
FCS in STAR software

My webpage https://www.star.bnl.gov/protected/spin/akio/fcs/index.html
Code location https://www.star.bnl.gov/cgi-bin/protected/cvsweb.cgi/**offline/upgrades/akio/**

- Geometry xml (WcalGeo0.g, HcalGeo0.g, PlatGeo0.g)
- G2T
- StEvent (StFcsCollection, StFcsHit, StFcsCluster, StFcsPoint)
- **Offline Raw data reader (StFcsRawHitMaker)**
- **Fast Simulator (StFcsFastSimulatorMaker)**
- **Pulse Fitter (StFcsWaveformFitMaker)**
- **Cluster Finder (StFcsClusterMaker)**
- **Photon fitting (StFcsPointMaker)**
- **DB for constants/calibration/utilities (StFcsDbMaker)**
- Pythia Filter for DY & Jet (FcsDYFilter, FCcsDYBGFilter, FcsJetFilter)
- New BFC chain options (StBFChain)
- Online Raw data reader (StRoot/StSpinPool/StFcsRawDaqReader)
- 2D Event Display (StFcsEventDisplay)
- Trigger Simulator & Bit checker (StFcsTriggerSimMaker)
- Online QA (StRoot/StSpinPool/StFcsQaMaker)
- Pi0 finder (StRoot/StSpinPool/StFcsPi0ReconstuctionMaker)
- MIP peak finder (StRoot/StSpinPool/StFcsMIPMaker)
- MuDst & PicoDst

Pi0 from Run19 AuAu200





Ecal MIP peak (charged hadrons) from Run19 AuAu200

Already in STAR library
Currently under STAR code review
To be merged to existing code
Analysis code
To be done

FCS in StEvent

- StEvent is STAR's (1st) data model used during "Production" BFC
- https://drupal.star.bnl.gov/STAR/comp/sofi/tutorials/stevent-special-documentation
- StFcsCollection
  - Holds all FCS related data in StSPtrVecXxx container
  - StFcsHit = tower Id, adc vs timebin, energy, waveform fit results
  - StFcsCluster = collection of hits in close proximity
  - StFcsPoint = EM shower shape fit to a cluster

StFcsHit https://www.star.bnl.gov/cgi-bin/protected/viewvc.cgi/cvsroot/StRoot/StEvent/StFcsHit.h?view=markup

```
UShort_t mDetId=0; // 1 bit ZS, 3 bits DetectorId, 12 bits id
UShort_t mDepCh=0; // 1 bit for NS, 2 bits for EHP, 5 bits for DEP, 8 bits for channel
UInt_t mAdcSum=0; // ADC sum
Float_t mFitPeak=0.0; // fit peak position
Float_t mFitSigma=0.0; // fit sigma
Float_t mFitChi2=0.0; // fit chi2
UInt_t mNPeak=0; // number of peaks found
Float_t mEnergy=0.0; // corrected energy
StFcsCluster* mCluster=0; // pointer to cluster this hit belongs
TArrayS* mData=0; // 12bit ADC values + flag at highest 4 bits, array of time bin
```

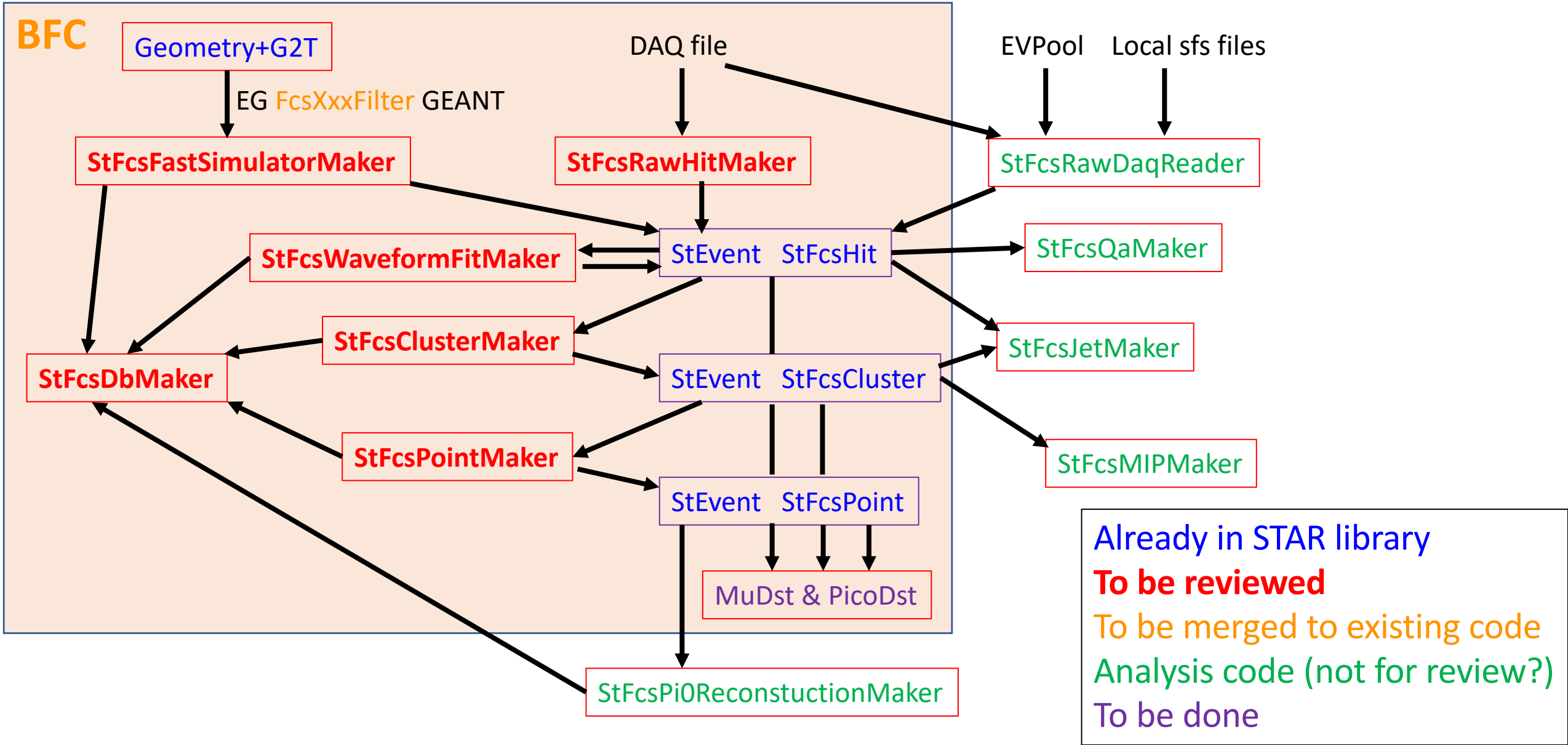
StFcsCluster https://www.star.bnl.gov/cgi-bin/protected/viewvc.cgi/cvsroot/StRoot/StEvent/StFcsCluster.h?view=markup

```
Int_t mId=-1; // Eventwise cluster ID
UShort_t mDetectorId=0; // Detector starts from 1
Int_t mCategory=0; // Category of cluster
Int_t mNTowers=0; // Number of non-zero-energy tower hits in the cluster
Float_t mEnergy=0.0; // Total energy contained in this cluster (0th moment)
Float_t mX=0.0; // Mean x ("center of gravity") in local grid coordinate (1st moment)
Float_t mY=0.0; // Mean y ("center of gravity") in local grid coordinate (1st moment)
Float_t mSigmaMin=0.0; // Minimum 2nd moment
Float_t mSigmaMax=0.0; // Maximum 2nd moment (along major axis)
Float_t mTheta=0.0; //Angle in x-y plane that defines the direction of least-2nd-sigma
Float_t mChi2Ndf1Photon=0.0; //  chi2/ ndf for 1-photon fit
Float_t mChi2Ndf2Photon=0.0; // chi2/ ndf for 2-photon fit
StLorentzVectorD mFourMomentum; // Cluster four momentum
StPtrVecFcsHit mHits; // Tower hits of the current cluster
StPtrVecFcsCluster mNeighbor; // Neighbor clusters
StPtrVecFcsPoint mPoints; // Fitted points (photons) in the cluster
```

StFcsPoint https://www.star.bnl.gov/cgi-bin/protected/viewvc.cgi/cvsroot/StRoot/StEvent/StFcsPoint.h?view=markup

```
UShort_t mDetectorId=0; // North=0, South=1
Float_t mEnergy=0; // Fitted energy
Float_t mX=0.0; // Fitted x-position in local coordinate
Float_t mY=0.0; // Fitted y-position in local coordinate
Int_t mNParentClusterPhotons=0; // Number of photons in the parent cluster
StFcsCluster* mCluster=0; // Pointer to parent cluster
StLorentzVectorD mFourMomentum; // Photon 4-momentum
StThreeVectorD mXYZ; // Photon position in STAR coordinate
```

FCS in STAR Software Flaw Chart



Mapping https://www.star.bnl.gov/protected/spin/akio/fcs/index.html#mapping

- Detector Map Det(0~5),Id(0~747),Row(1~34), Col(1~22)
- Readout Map EHP(0~3),NS(0~1),DEP(0~23),CH(0~31), Crt(0~4), Slt(0~19)
- Slow Control Map EHP(0~3),NS(0~1),DEP(0~23),Branch(0~1),Address(0~12), SiPM#(0~3),PatchPanel, CableColor
- Name combines all 3 like "EN023_r02c02_Dep00Ch00_F20/0/00/3"
- StRoot/StEvent/StEnumerations.h has basic some constants

```
enum StFcsDetectorId{
        kFcsEcalNorthDetId=0,
        …
}
enum StFcsConstants {
    kFcsNDet=6,
    …
}
```

- StFcsDbMaker converts between maps:

```
Int_t nRow(Int_t det)
Int_t nColumn(Int_t det)
Int_t maxId(Int_t det)
getName(Int_t det, Int_t id, char name[])
getDepfromId (Int_t detectorId, Int_t id, Int_t &ehp, Int_t &ns, Int_t &crt, Int_t &slt, Int_t &dep, Int_t &ch)
getIdfromDep (Int_t ehp, Int_t ns, Int_t dep, Int_t ch, Int_t &detectorId, Int_t &id, Int_t &crt, Int_t &slt)
getSCmap(Int_t det, Int_t id, Int_t &ehp, Int_t &ns, Int_t &scdep, Int_t &branch, Int_t &fee_i2c, Int_t &sipm, Int_t &pp, Int_t &jacket)
getFromName(const char name[], Int_t& det, Int_t& id)
```

DataBase https://drupal.star.bnl.gov/STAR/subsys/fcs/fcs-db

Offline Database, Geometry for FCS

- fcsDetectorPosition.idl (1 row) : Detector Position for each [det=0~3]

```
struct fcsDetectorPosition {
float xoff[4]; /* x offset [cm] from beam to front of near beam corner */
float yoff[4]; /* y offset [cm] from beam to detector center */
float zoff[4]; /* z offset [cm] from IR to front of near beam corner */
};
```

Offline Database, Calibration for FCS

- fcsEcalGain.idl (1 row) : Ecal Gain [GeV/ch]

```
struct fcsEcalGain {
float gain[1496]; /* gain 2*748 */
};
```

- fcsEcalGainCorr.idl (1 row) : Ecal Gain Correction[unitless]

```
struct fcsEcalGainCorr {
float gaincorr[1496]; /* gain correction 2*748 */
};
```

- fcsHcalGain.idl (1 row) : Hcal Gain [GeV/ch]

```
struct fcsHcalGain {
float gain[520]; /* gain 2*260 */
};
```

- fcsHcalGainCorr.idl (1 row) : Hcal Gain Correction [unitless]

```
struct fcsHcalGainCorr {
float gaincorr[520]; /* gain correction 2*260 */
};
```

- fcsPresGain.idl (1 row) : Pres Gain [MIP/ch]

```
struct fcsPresGain {
float gain[384]; /* gain 2*192 */
};
```

- fcsPresValley.idl (1 row) : Pres Valley Position [MIP]

```
struct fcsPresValley {
float valley[384]; /* valley 2*192 */
};
```

Online DB

FEE Bd temperature and current (Just monitor for radiation damage)
MPOD voltage & current (Alarm)
DEP temperature (Alarm)
Voltage setting for each run?
Gain correction used for triggering?
Attenuator setting for each run? We may need this @ offline

Offline DB

XYZ offset (MC geometry may also pick up this)
Gain [GeV/ADC count]
GainCorrection [no unit]

Ecal/Hcal. Energy = ADC Integral * Gain * GainCorrection
PRES(EPDW) # of MIP = ADC Integral * Gain
& Valley Position for Hit/NoHit cut (~0.5 MIP)

Mappings are currently generated from StFcsDbMaker code

StFcsFastSimulatorMaker

Reading G2T in FZD file, making StFcsHit in StEvent

https://www.star.bnl.gov/cgi-bin/protected/viewvc.cgi/cvsroot/offline/upgrades/akio/StRoot/StFcsFastSimulatorMaker/

- Reading FCS (WCAL and HCAL) G2T table
- It also read EPD G2T table since we'll have EPD W signals in DEP as separate readout
- Handles a couple of different light collection models for Hcal
- Handles different z dependent light collection efficiency models for Hcal
- Access StFcsDbMaker for Mapping
- Fill StFcsHitCollection in StEvent
- Currently there is no simulation of pulse shape vs time bin – putting all dE in 1 time bin

StFcsRawHitMaker

Reading DAQ file, making StFcsHit in StEvent

https://www.star.bnl.gov/cgi-bin/protected/viewvc.cgi/cvsroot/offline/upgrades/akio/StRoot/StFcsRawHitMaker/

- Inherit from StRTSBaseMaker
- Reading DAQ file
- Can switch reading
  - “ADC” bank - None pedestal subtracted, none zero suppressed data for pedestal & LED runs
  - “ZS” bank - For physics runs
- Access StFcsDbMaker for Mapping
- Fill StFcsHitCollection in StEvent with
  - encoded channel ids & adc/zs flag
  - ADC vs time bin data

StFcsDbMaker – DB interface & Utilities

https://www.star.bnl.gov/cgi-bin/protected/viewvc.cgi/cvsroot/offline/upgrades/akio/StRoot/StFcsDbMaker/

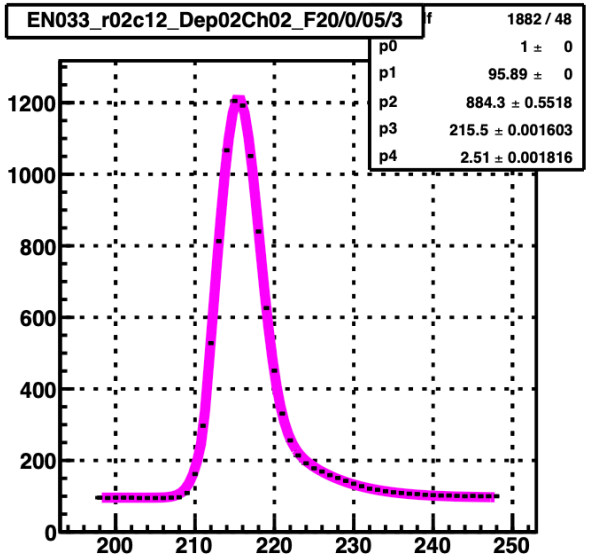
- Interface to offline DB
  - Offline DB tables : https://drupal.star.bnl.gov/STAR/subsys/fcs/fcs-db already in DEV
  - Geometry : Detector Position (xyz offsets)
  - Calibration : Gain [GeV/ch] & Gain Correction [no unit] & Valley location(Pres)
- Many other utility functions
  - Generating Maps. This is used for detector construction & cabling
    - https://www.star.bnl.gov/protected/spin/akio/fcs/index.html#mapping
  - Converting between detector, readout and slow control maps
  - Local coordinate & STAR coordinate positions
  - We anticipate many of those are not changing at all
    - Mapping for example, cannot be changed because tied with trigger algorithm
    - Their logic (not simple numbers) are “hard coded”
    - If we decide to change it, we can switch to DB without changing interface & user codes

StFcsWaveformFitMaker – Pileup proof pulse fit

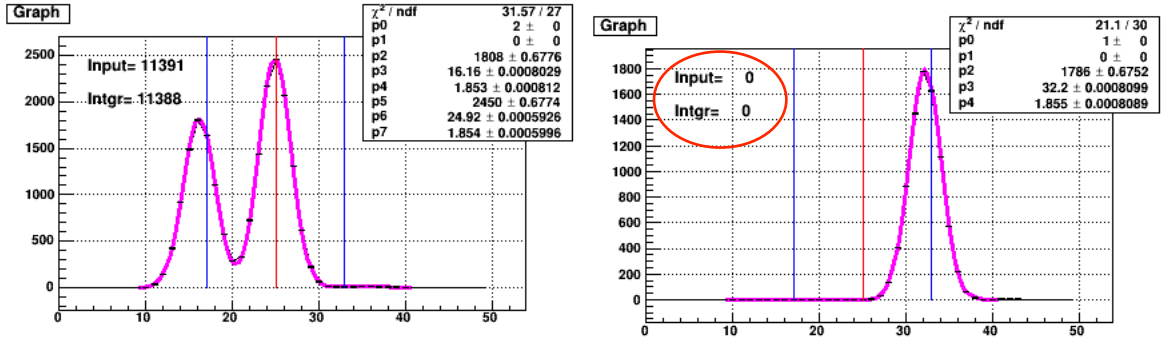
https://www.star.bnl.gov/cgi-bin/protected/viewvc.cgi/cvsroot/offline/upgrades/akio/StRoot/StFcsWaveformFitMaker/

- Readout (DEP board) gives 12bit ADC for every 1/8 of RHIC clock (~*13.5nsec*)
- Locate peaks in ADC vs time bin data from a StFcsHit
- Fit pulse shape vs time bin using gaussian + pre-determined tail function
- Apply gain & gain correction from StFcsDbMaker
- Set resulted integral of pulse in StFcsHit as energy
- Fit with multiple pulses when multiple peaks found
- Only report a pulse found in ”triggered” time bin range
- Initial study with test pulse:
  - https://www.star.bnl.gov/protected/spin/akio/fcs/pulse/index.html
- Updated study with real detector and real LED pulse:
  - https://www.star.bnl.gov/protected/spin/akio/fcs/pulse_led/index.html
- Choice of different fitter and various ”sum” methods without fit
- Currently it takes 2~3msec per fit …. Too slow???

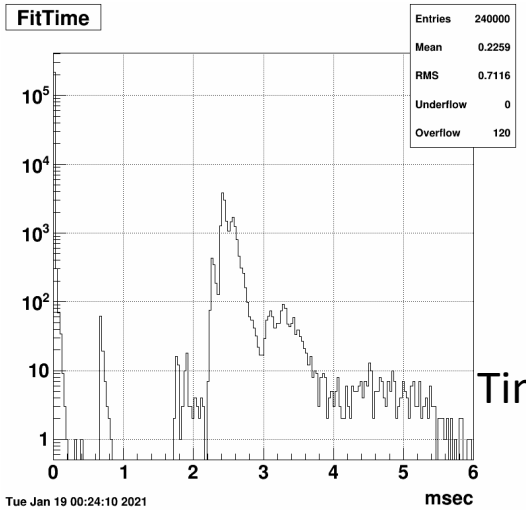
$$xexp(x)=\frac{A}{\tau^2}(x-x0)^p e^{\frac{-(x-x0)}{\tau}}$$



Example fit to LED data



Fir to waveform simulator with pileups



Time to do a fit

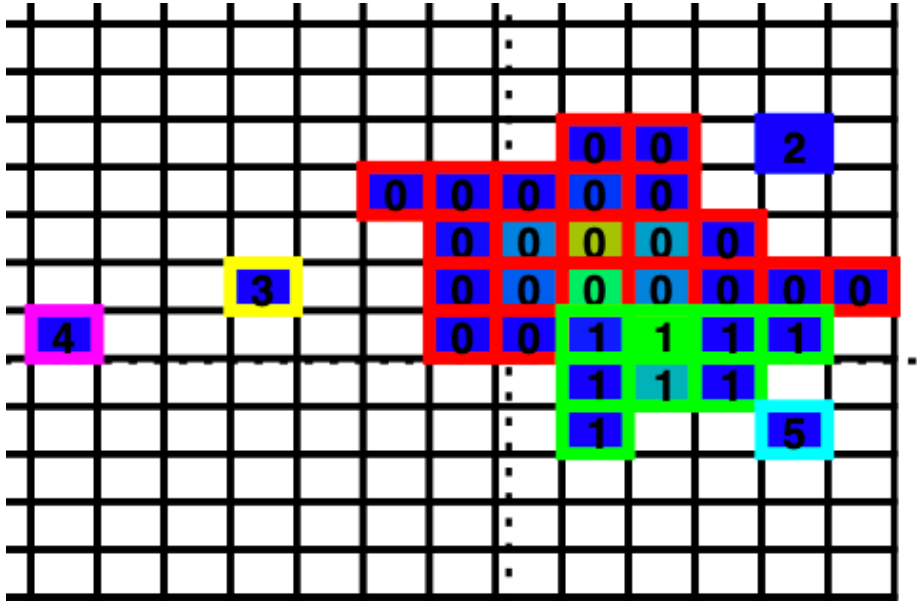
StFcsClusterMaker

Cluster finder for FCS

https://www.star.bnl.gov/cgi-bin/protected/viewvc.cgi/cvsroot/offline/upgrades/akio/StRoot/StFcsClusterMaker/

- Re-write of cluster finder for FMS
- Read StFcsHitCollection
- Sort towers by energy
- Different parameters for Ecal & Hcal
- Looping all towers in descending order
  - If it is neighbor to existing cluster, add to the cluster
  - If not, create new cluster
- Perform cluster moment analysis
- Store resulting clusters in to StFcsClusters
  - Total energy
  - Center position (local coordinate)
  - Set Category, SigmaMax & SigmaMin from moment analysis
  - Pointers to all StFcsHits
  - Pointers to neighbor clusters

Example pi0 event from MC



By StFcsEventDisplayMaker

Color of cell represent energy
Color of border line (and number) shows which cluster it belongs

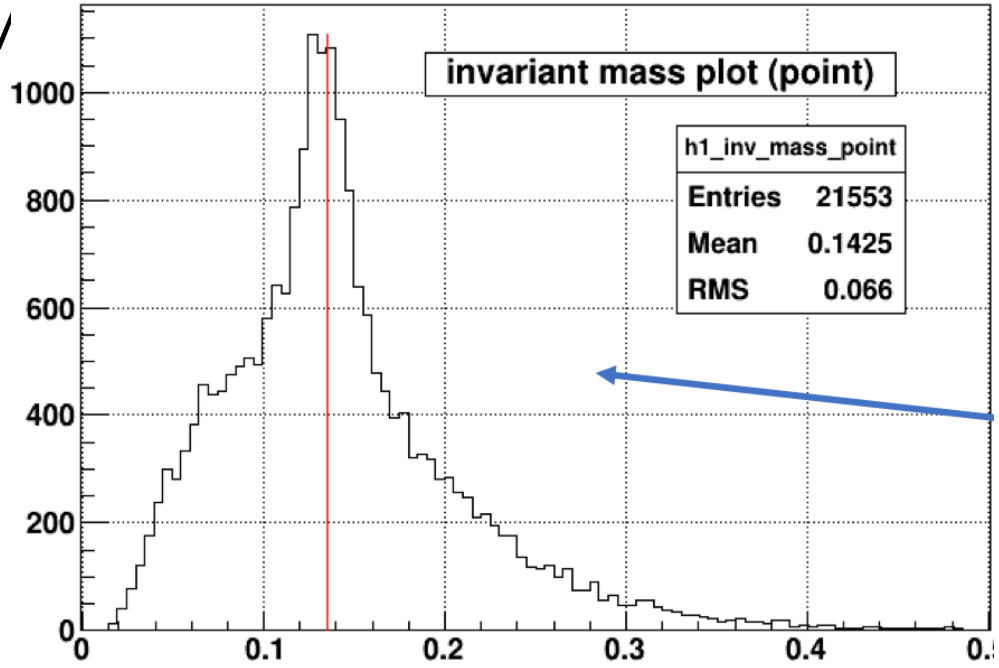
StFcsPointMaker

EM shower fit for FCS Ecal

https://www.star.bnl.gov/cgi-bin/protected/viewvc.cgi/cvsroot/offline/upgrades/akio/StRoot/StFcPointMaker/

- Re-write of Photon Fitter for FMS
- Read StFcsClusterCollection
- Few choice of 2D/3D EM shower shapes scaled from FM
- Depending on cluster category
  - 1 photon fit
  - 2 photon fit
  - Try both and pick better one
- Store result in to StFcsPoints
  - Fit energy
  - Center position (2D local coordinate)
  - Center position in STAR coordinate
  - Pointers to parent cluster
  - Also put chi2 and pointer to StFcsPoints in StFcsCluster

Pi0 from Run19 AuAu200



Summary

- Many of FCS software are written and already being used for
  - For MC study like trigger algorithm design, HCal hardware design, etc
  - For Run19 AuAu200GeV test data
    - We see pi0 and MIP peak!!!
  - For commissioning with full detector right now (Run21)
- Geometry, G2T and StEvent are already in STAR library
- 6 Makers which will be in BFC are under STAR code reviews
- Event Filters and BFC changes should be merged to existing codes
- Other "user analysis code" are in StRoot/StSpinPool
- We need uDST and re-create StEvent on memory for re-running(and picoDST?)
- Codes are in CVS at $CVSROOT/offline/upgrades/akio
- Howto : https://www.star.bnl.gov/protected/spin/akio/fcs/index.html#howto