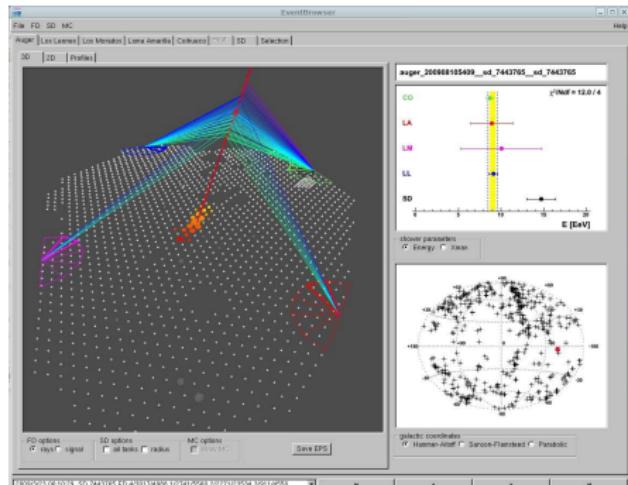


NDST: A first attempt of unifying NA61 software

X. Garrido^a, I. Mariš^b, M. Szuba^c, R. Ulrich^{c,d}, M. Unger^c

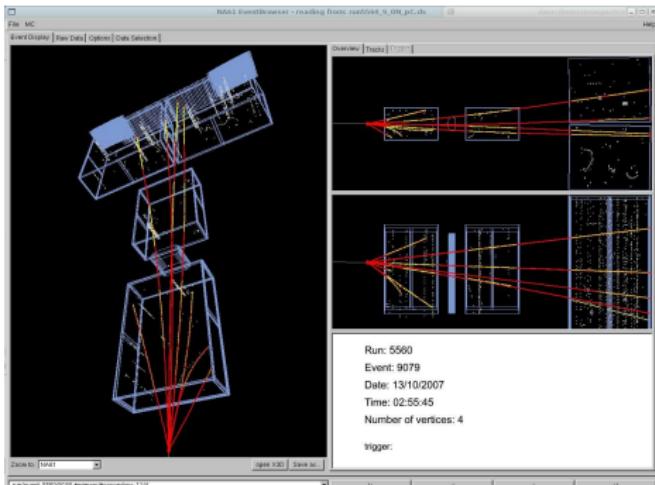
^aLAL, ^bLPNHE, ^cKIT, ^dPSU

from 100 EeV...



(Pierre Auger Observatory, quadruple event, $E \sim 10^{19}$ eV)

...to 31 GeV/c



(NA61, p-C at 31 GeV/c, 2007)

New Data Summary Trees (NDST)

state of the art at NA61 startup (NA49 software)

- ▶ event formats
 - ▶ raw data (BOS/DSPACK)
 - ▶ full reconstruction output (DSPACK)
 - ▶ miniDSTs (ROOT)
- ▶ event visualization
 - ▶ OM (online monitor, BOS, Motif)
 - ▶ TopView (2D, DSPACK, HIGZ&ZEBRA)
 - ▶ x3d (DSPACK, X11 wireframe viewer)
 - ▶ qcdisplay (DSPACK, Motif)

goal of NDST development

unification of data structures and event visualization

NDST features

- ▶ based on highly successful Auger 'ADST'-package
- ▶ uses ROOT for streaming, schema evolution
- ▶ data structures: classes and STL containers
- ▶ visualization using ROOTs GUI features
- ▶ 'shrinkable' design to allow same structures to hold full detail or only miniDST

NDST features

NDST supports

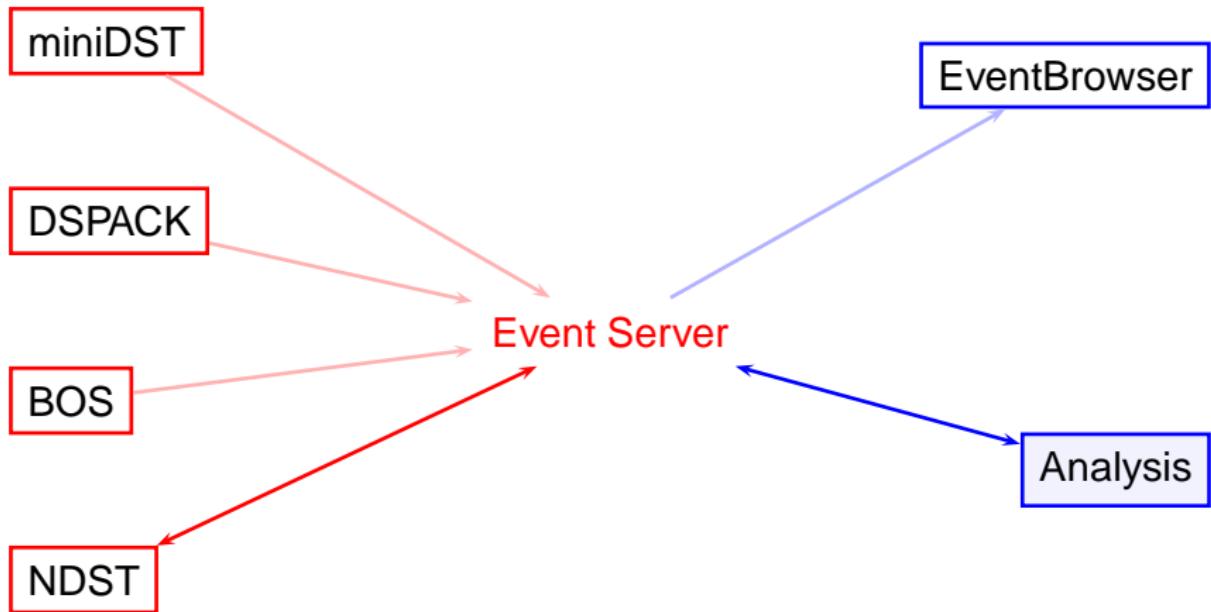
- ▶ high level reconstruction variables (vertices, tracks)
- ▶ MC input
- ▶ intermediate results (clusters charge, unused tracks)
- ▶ raw data (BOS)

hence 'data summary tree' is a misnomer, but file can be compressed via

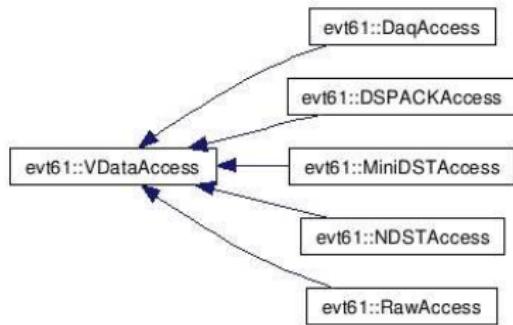
```
EventFile/EventServer::SetMicroDST()
```

after which data corresponds to T61 DSTs

Data Formats and NDST Framework



Unified data access



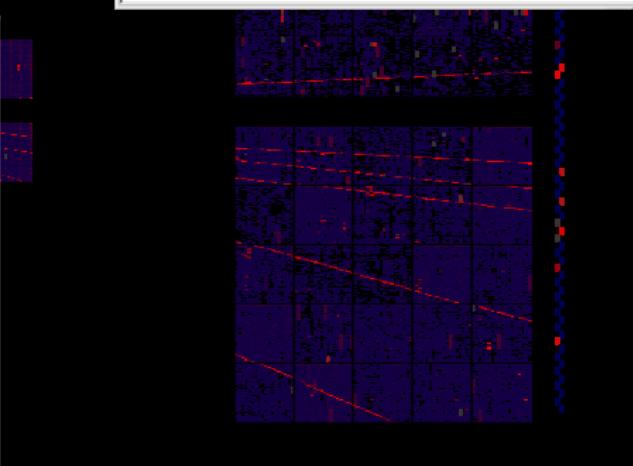
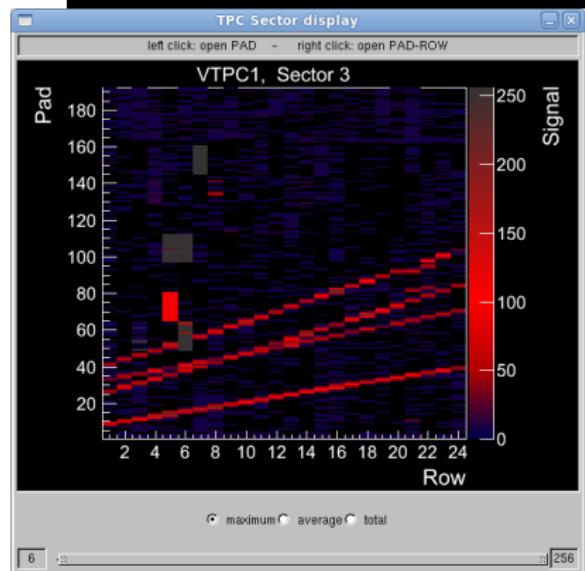
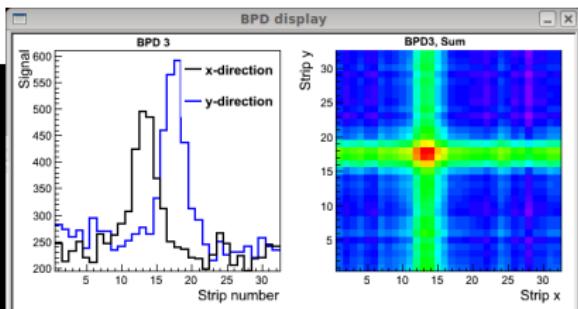
(doxygen class hierarchy)

all common NA61 data formats
can be mapped to NDST

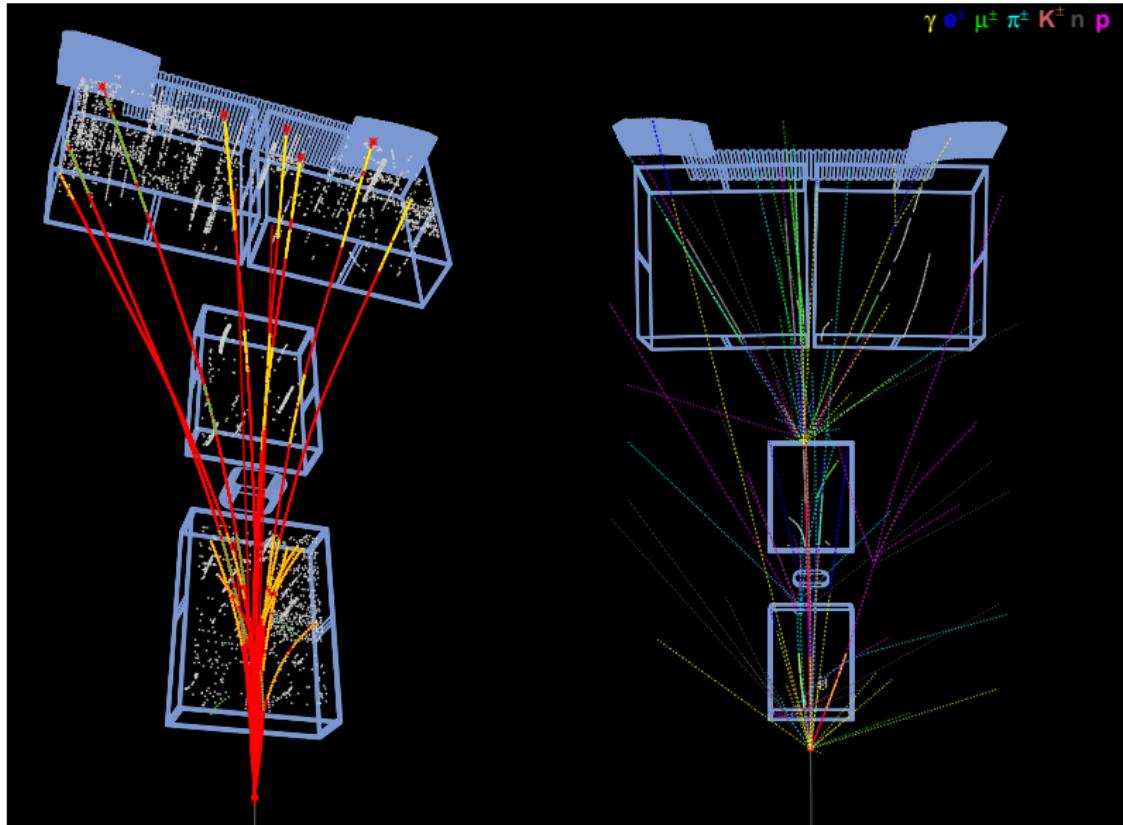
- ▶ raw data (BOS)
- ▶ DSPACK
- ▶ T61 ROOT miniDSTs
- ▶ and (of course) NDST

Example: Online monitoring

Run 9891 Event 691 02/11/2009 17:13:13



Example: EventBrowser (data/MC)



Simple event loop

```
EventServer eventServer;
eventServer.Open(fileNames);

while( eventServer.NextEvent() ) {

    const Event& na61Event = eventServer.GetEvent();

    const int run = na61Event->GetHeader().GetRunNumber();
    const int event = na61Event->GetHeader().GetEventNumber();
    const RecBeam& beam = na61Event->GetRecEvent().GetBeam();
    const int trigger = beam.GetPatUnit1();

    cout << " run " << run
        << " event " << event <<
        << " trigger: " << trigger << endl;

}
```

same code for BOS, DSPACK, miniDST, NDST!

Looping over tracks

```
const Event& event = eventServer.GetEvent();
const RecEvent& recEvt = event.GetRecEvent();

for (RecEvent::VertexIterator vtxIter = recEvt.VerticesBegin();
     vtxIter != recEvent.VerticesEnd(); ++vtxIter) {

    for (RecVertex::TrackIterator trackIter = vtxIter.TracksBegin();
         trackIter != vtxIter->TracksEnd(); ++trackIter) {

        const TVector3& momentum    = iRecTrack->GetMomentum();
        cout << " momentum is " << momentum.Mag() << endl;
    }
}
```

same code for (BOS), DSPACK, miniDST, NDST!

NDST Todo

general overhaul needed to serve as the backbone of new framework:

- ▶ naming conventions
(hits→clusters,...)
- ▶ better linking between vertices and tracks
- ▶ not all DSPACK data implemented yet
- ▶ ...

IT TOOK A LOT OF WORK, BUT THIS LATEST LINUX PATCH ENABLES SUPPORT FOR MACHINES WITH 4,096 CPUs, UP FROM THE OLD LIMIT OF 1,024.

DO YOU HAVE SUPPORT FOR SMOOTH FULL-SCREEN FLASH VIDEO YET?
NO, BUT WHO USES THAT?

