Introduction to PandaRoot

Panda CM II 2018 Stockholm

Ralf Kliemt

PandaRoot Overview/Structure

PandaRoot Overview/Structure

Event Generator

Transport

Detector Simulation

Digitization

Local

Global

Analysis

Particle gun: PndBoxGenerator

```
TStringinputGenerator = "box:type(211,1):p(1):tht(10,120)";
PndMasterRunSim *fRun = new PndMasterRunSim();
fRun->SetInput(inputGenerator);
```

Signal with EvtGen

```
TString inputGenerator = "psi2s_Jpsi2pi_Jpsi_mumu.dec";
PndMasterRunSim *fRun = new PndMasterRunSim();
fRun->SetInput(inputGenerator);
```

```
Find out which Models are there:
pgenerators/EvtGen/EvtGen/Private/EvtGenModels$ grep -A 2 "getName()" *cpp | grep return
```

Background (generic pions, etc.)

- DPM & FTF

```
TStringinputGenerator = "ftf";
//TString inputGenerator = "dpm";
PndMasterRunSim *fRun = new PndMasterRunSim();
fRun->SetInput(inputGenerator);
```

PandaRoot Overview/Structure

Event Generator

Transport

Detector Simulation

Digitization

Local

Global

Analysis

GEANT3 & 4 transport particles through geometry volumes

- magnetic field
- energy loss
- scattering
- Bremsstrahlung

"Active" and "passive" materials

Detector code called when particle is passing sensitive volumes

- Positions, momenta, time, energy loss

PandaRoot Overview/Structure

Event Generator

Transport

Detector Simulation

Digitization

Local

Global

Analysis

Detector options:

- day1: Reduced setup from autumn 2017:
  - FTS1234
  - …
- Additionally:
  - gem2/gem3: GEM with 2 or 3 planes
  - fts1256: remove FTS layers 3&4, not 5&6
  - strips: No MVD pixels

```
TString parAsciiFile= "all.par";
TString prefix= "evtcomplete"; // prefix string for output files
TString options= ""; //"day1+gem2+strip+fts1256"
// TString inputGenerator =
// EvtGen -> "xxxxxxxx.dec"
// DPM -> "dpm_xxxxx"
// FTF -> "ftf_xxxxx"
// BOX -> "box:type(pdgcode,mult):p(min,max):tht(min,max):phi(min,max)"
// PIPI-> "pipi:cosTheta(min,max)"
// LEP -> "leplep:pid(value):gegm(value):cosTheta(min,max)"

TString inputGenerator = "psi2s_Jpsi2pi_Jpsi_mumu.dec";
//TString inputGenerator = "dpm";
//TString inputGenerator = "ftf";
//TString inputGenerator = "box:type(211,1):p(1,1):tht(10,120):phi(0,360)";

PndMasterRunSim *fRun = new PndMasterRunSim();
fRun->SetInput(inputGenerator);
fRun->SetName(SimEngine);
fRun->SetOptions(options);
fRun->SetParamAsciiFile(parAsciiFile);
fRun->SetNumberOfEvents(nEvents);
fRun->SetBeamMom(BeamMomentum);
fRun->SetStoreTraj(kTRUE);

fRun->Setup(prefix);
fRun->CreateGeometry();
fRun->SetGenerator();
```

e.g. in macro/master

Adding detectors by hand, e.g. instead of

fRun->CreateGeometry();

```
FairModule *Cave= new PndCave("CAVE");
Cave->SetGeometryFileName("pndcave.geo");
fRun->AddModule(Cave);
//-------------------------Magnet-----------------
FairModule *Dipole= new PndMagnet("MAGNET");
Dipole->SetGeometryFileName("dipole.geo");
fRun->AddModule(Dipole);
//-------------------------Pipe-----------------
FairModule *Pipe= new PndPipe("PIPE");
Pipe->SetGeometryFileName("beampipe_201309.root");
fRun->AddModule(Pipe);
//-------------------------STT ------------------
FairDetector *Stt= new PndStt("STT", kTRUE);
Stt->SetGeometryFileName(
“straws_skewed_blocks_35cm_pipe.geo");
fRun->AddModule(Stt);
//-------------------------MVD ------------------
FairDetector *Mvd = new PndMvdDetector("MVD", kTRUE);
Mvd->SetGeometryFileName("Mvd-2.1_FullVersion.root");
fRun->AddModule(Mvd);
//-------------------------GEM ------------------
FairDetector *Gem = new PndGemDetector("GEM", kTRUE);
Gem->SetGeometryFileName(
"gem_3Stations_realistic_v2.root");
fRun->AddModule(Gem);
//-------------------------EMC ------------------
PndEmc *Emc = new PndEmc("EMC",kTRUE);
Emc->SetGeometryVersion(1);
Emc->SetStorageOfData(kFALSE);
fRun->AddModule(Emc);
//-------------------------SCITIL ------------------
FairDetector *SciT = new PndSciT("SCIT",kTRUE);
SciT->SetGeometryFileName("SciTil_201601.root");
fRun->AddModule(SciT);
//-------------------------DRC ------------------
PndDrc *Drc = new PndDrc("DIRC", kTRUE);
Drc->SetGeometryFileName("dirc_e3_b3_l6_m40.root");
Drc->SetRunCherenkov(kFALSE);
fRun->AddModule(Drc);
//-------------------------DISC------------------
PndDsk* Dsk = new PndDsk("DSK", kTRUE);
Dsk->SetStoreCerenkovs(kFALSE);
Dsk->SetStoreTrackPoints(kFALSE);
fRun->AddModule(Dsk);
//-------------------------MDT ------------------
PndMdt *Muo = new PndMdt("MDT",kTRUE);
Muo->SetBarrel("fast");
Muo->SetEndcap("fast");
Muo->SetMuonFilter("fast");
Muo->SetForward("fast");
Muo->SetMdtMagnet(kTRUE);
Muo->SetMdtCoil(kTRUE);
Muo->SetMdtMFIron(kTRUE);
fRun->AddModule(Muo);
//-------------------------FTS ------------------
FairDetector *Fts= new PndFts("FTS", kTRUE);
Fts->SetGeometryFileName("fts.geo");
fRun->AddModule(Fts);
//-------------------------FTOF------------------
FairDetector *FTof = new PndFtof("FTOF",kTRUE);
FTof->SetGeometryFileName("ftofwall.root");
fRun->AddModule(FTof);
//-------------------------RICH ------------------
PndRich *Rich= new PndRich("RICH",kTRUE);
Rich->SetGeometryFileName("rich_v313.root");
fRun->AddModule(Rich);
```

see macro/run

PandaRoot Overview/Structure

- Event Generator
- Transport
- Detector Simulation
- **Digitization**
- Local
- Global
- Analysis

All detector effects including

- segmentation
- electronics emulation
- noise & thresholds
- timing

Data format looks like real detector answer

```
// -----   Add tasks ------
fRun->AddDigiTasks();
```

Explicit Digi Tasks

```
PndPersistencyTask *task;

// -----Add tasks
task = new PndMvdDigiTask();
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndMvdClusterTask();
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndSttHitProducerRealFast();
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndGemDigitize("GEM Digitizer", 0);
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndGemFindHits("GEM Hit Finder", 0);
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndDrcHitProducerReal();
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndSciTDigiTask();
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndEmcHitsToWaveform();
task->SetPersistency(kFALSE);
task->SetVerbose(0);
fRun->AddTask(task);
```

```
task = new PndEmcWaveformToDigi();
task->SetPersistency(kTRUE);
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndEmcMakeCluster();
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndEmcMakeBump();
task->SetVerbose(0);
fRun->AddTask(task);

PndMdtHitProducerIdeal* mdt = new
                    PndMdtHitProducerIdeal();
mdt->SetPositionSmearing(.3);
mdt->SetVerbose(0);
fRun->AddTask(mdt);

task = new PndMdtTrkProducer();
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndFtsHitProducerRealFast();
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndFtofHitProducerIdeal();
task->SetVerbose(0);
fRun->AddTask(task);

task = new PndRichHitProducer();
task->SetVerbose(0);
fRun->AddTask(task);
```

macro/run

PandaRoot Overview/Structure

- Event Generator
- Transport
- Detector Simulation
- Digitization
- **Local**
- **Global**
- Analysis

Reconstruction for each detector part

Cluster finding

Hit mapping

calculating observables

Tracking: MVD, STT, GEM combined

PID: Correlation of detector information

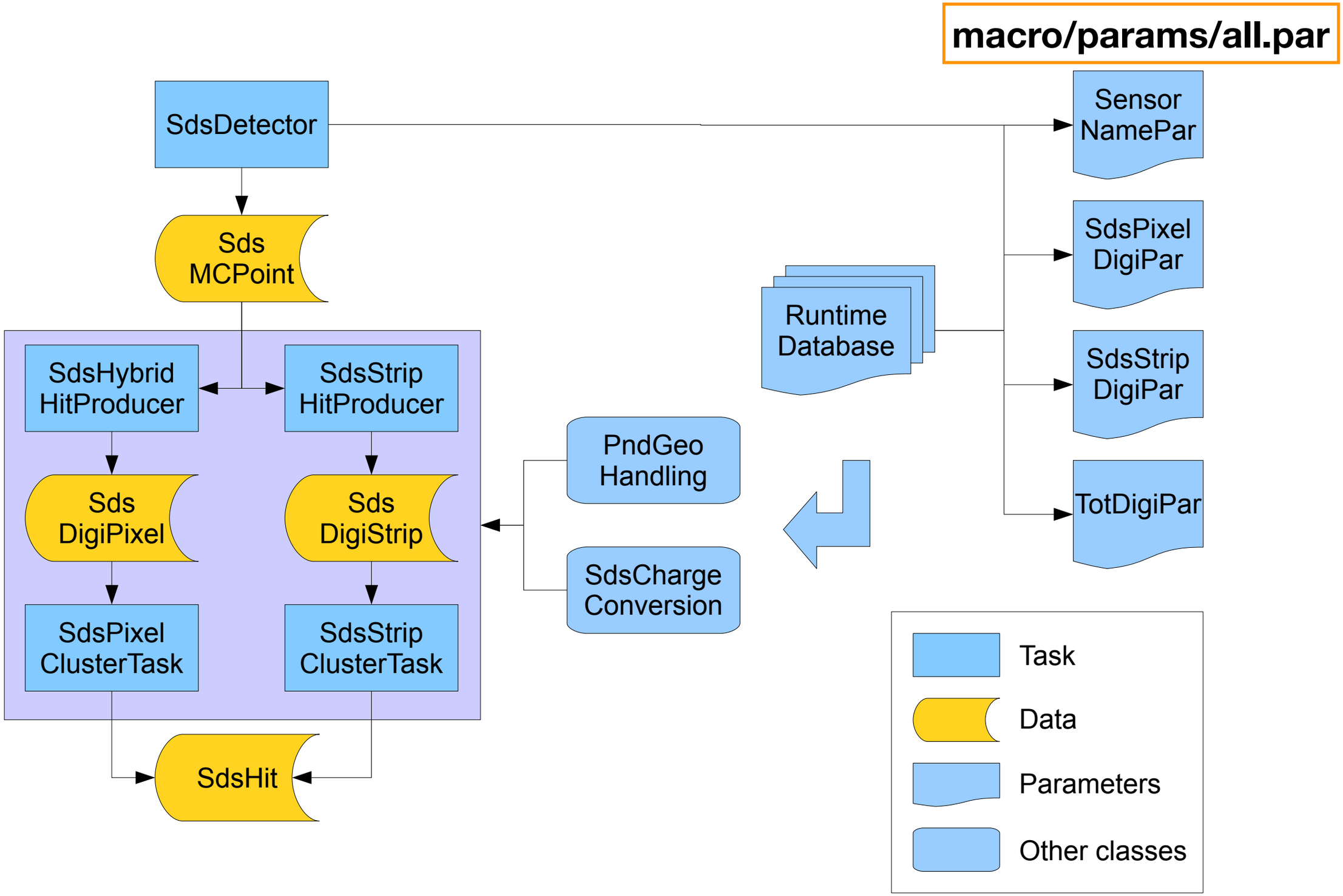
Merging time information, etc.

```
// -----   Add tasks ------
fRun->AddRecoTasks();
```

Explicit Tracking Tasks

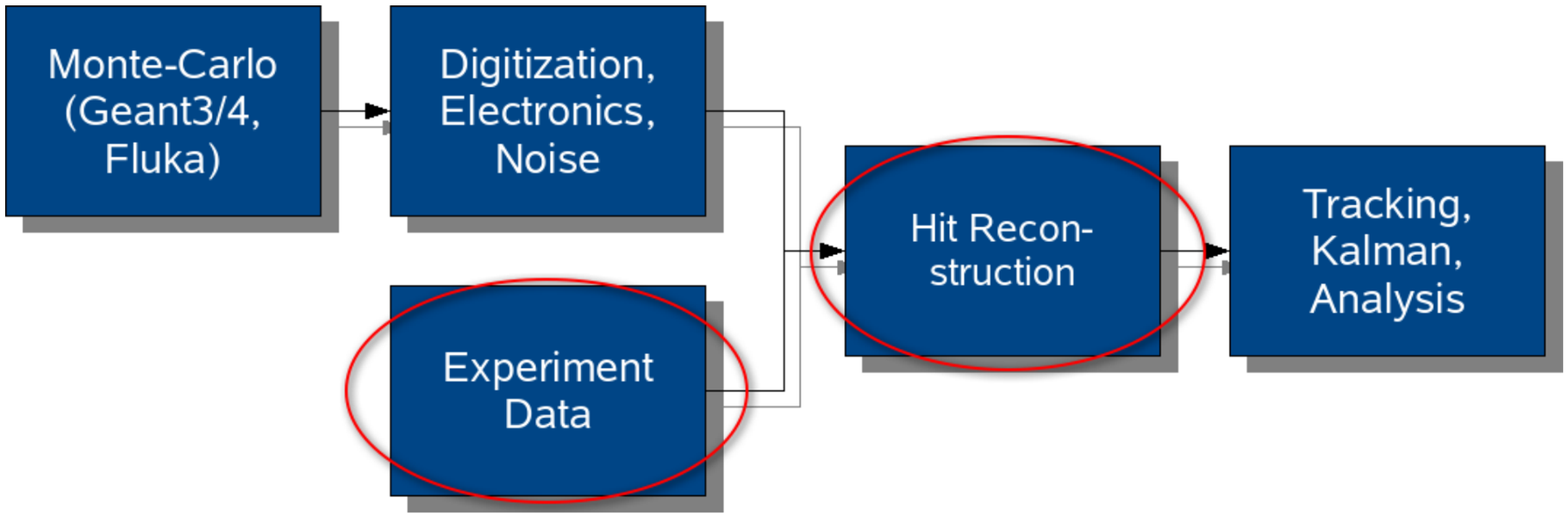
```
FairGeane *Geane = new FairGeane();
fRun->AddTask(Geane);

PndTrkTracking2* tracking = NULL;
fRun->AddTask(tracking = new
PndTrkTracking2(0,false,false,true));
tracking-
>SetInputBranchName("STTHit","MVDHitsPixel","MVDHitsStr
ip");
tracking->NoMvdAloneTracking();
tracking->SetPersistency(kFALSE);

PndSttMvdGemTracking *SttMvdGemTracking = NULL;
fRun->AddTask(SttMvdGemTracking = new
PndSttMvdGemTracking(0));
SttMvdGemTracking->SetPersistency(kFALSE);

PndRecoKalmanTask* recoKalman = NULL;
fRun->AddTask(recoKalman = new PndRecoKalmanTask());
recoKalman->SetTrackInBranchName("SttMvdGemTrack");
recoKalman->SetTrackOutBranchName("SttMvdGemGenTrack");

recoKalman->SetBusyCut(50); // CHECK to be tuned
//recoKalman->SetIdealHyp(kTRUE);
//recoKalman->SetNumIterations(3);
recoKalman->SetTrackRep(0); // 0 Geane (default), 1 RK
//recoKalman->SetPropagateToIP(kFALSE);

if (fOptions.Contains("filtered")){
  PndMissingPzCleanerTask* cleaner = NULL;
  fRun->AddTask(cleaner = new
PndMissingPzCleanerTask()); //4
  cleaner->SetInputTrackBranch("SttMvdGemGenTrack");
  cleaner->SetRemoveTrack(kTRUE);
}

PndIdealTrackFinder* trackFts = NULL;
fRun->AddTask(trackFts = new PndIdealTrackFinder());
trackFts->SetTrackSelector("FtsTrackFunctor");
trackFts->AddBranchName("FTSHit");
trackFts->AddBranchName("MVDHitsPixel");
trackFts->AddBranchName("MVDHitsStrip");
trackFts->SetRelativeMomentumSmearing(0.05);
trackFts->SetVertexSmearing(0.05, 0.05, 0.05);
trackFts->SetTrackingEfficiency(1.);
trackFts->SetOutputBranchName("FtsIdealTrack");
trackFts->SetPersistence(kFALSE);

PndRecoKalmanTask* recoKalmanFwd = NULL;
fRun->AddTask(recoKalmanFwd = new PndRecoKalmanTask());
recoKalmanFwd->SetTrackInBranchName("FtsIdealTrack");
//recoKalmanFwd-
>SetTrackInIDBranchName("FtsIdealTrackID");
recoKalmanFwd-
>SetTrackOutBranchName("FtsIdealGenTrack");
recoKalmanFwd->SetBusyCut(50); // CHECK to be tuned
//recoKalmanFwd->SetIdealHyp(kTRUE);
//recoKalmanFwd->SetNumIterations(3);
recoKalmanFwd->SetTrackRep(0); // 0 Geane (default), 1
RK
//recoKalmanFwd->SetPropagateToIP(kFALSE);
```

Detector Processing Example

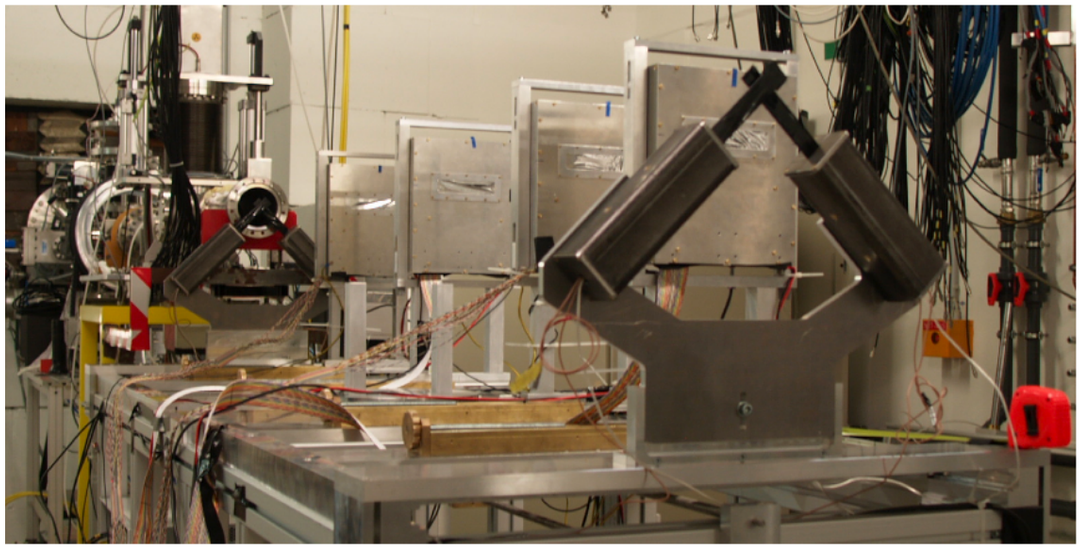


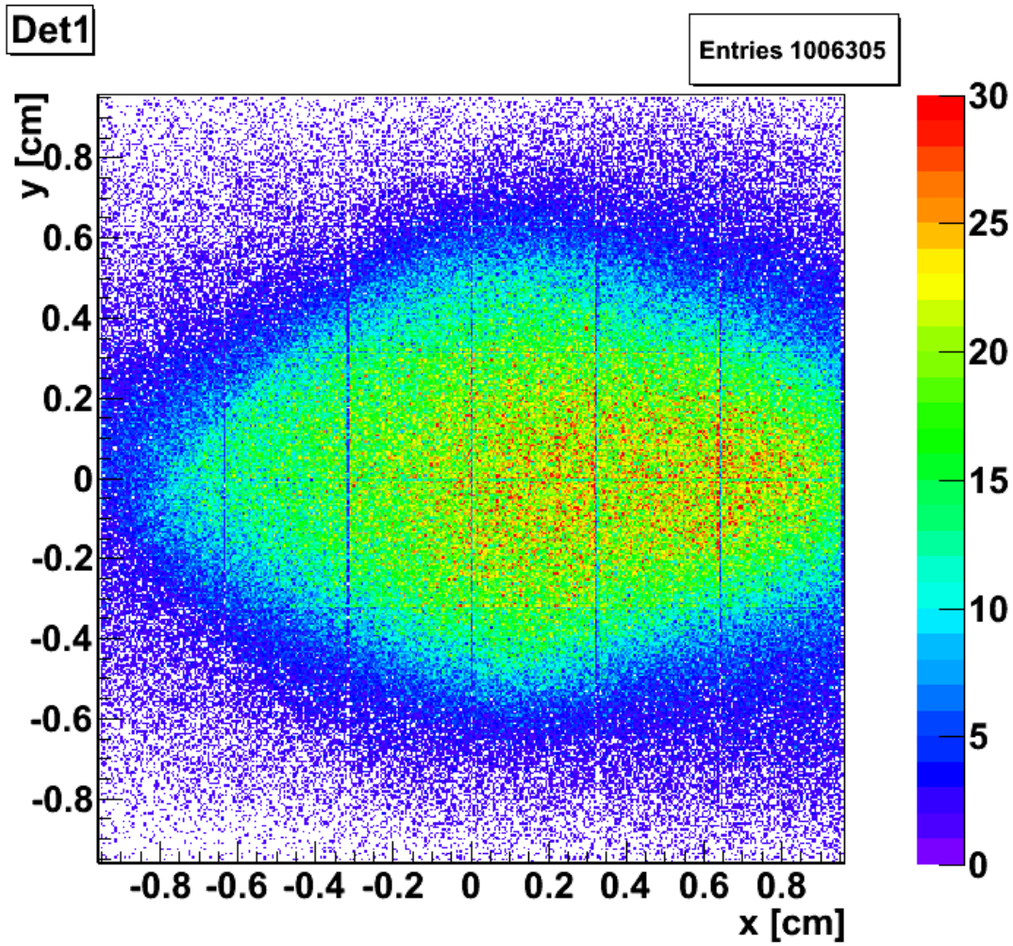
Real Data Reconstruction

since 2013



Bonn Tracking Station





PandaRoot Overview/Structure

Event Generator

Transport

Detector Simulation

Digitization

Local

Global

Analysis

- Extract physics from reconstructed 4-momenta
- Combinatorics
- Geometric & Kinematic fits
- Spectra, fits, angular distributions....

see more: tutorials/thailand2017

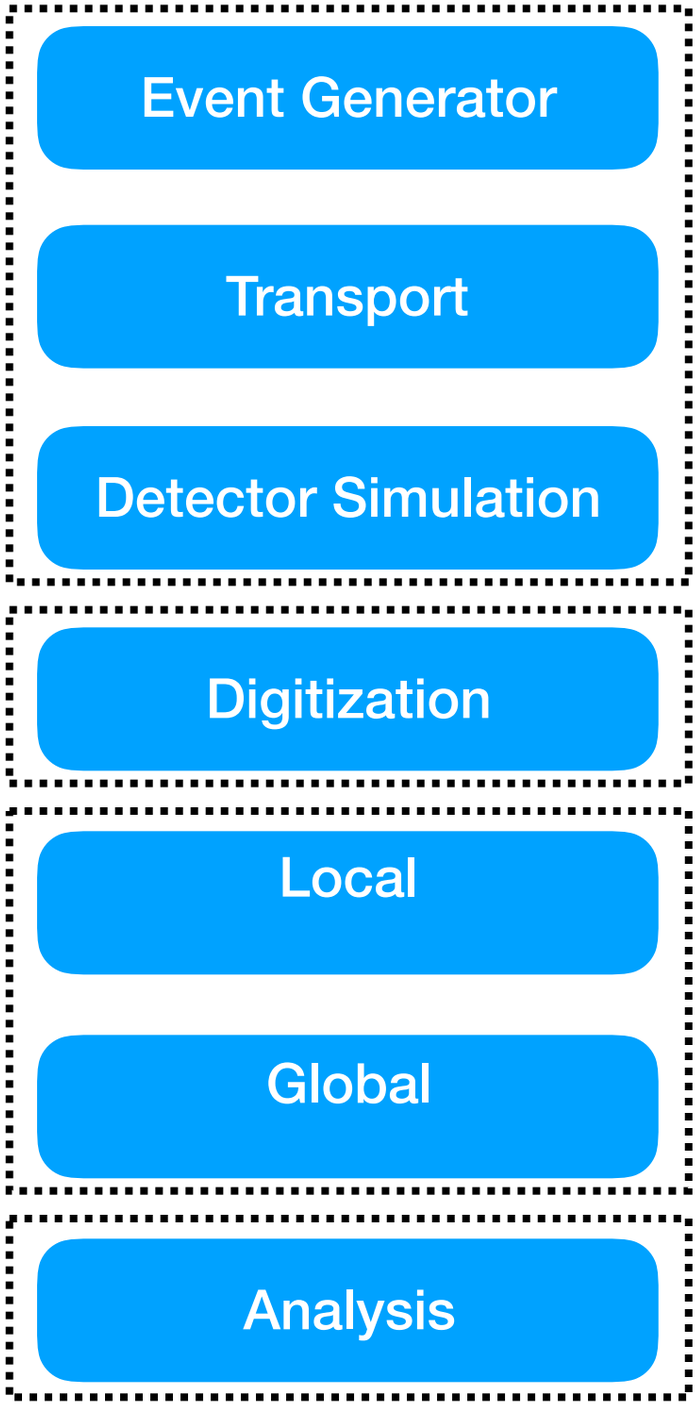
PandaRoot Overview/Structure

Data types

- Event Generator
- Transport
- Detector Simulation
- Digitization
- Local
- Global
- Analysis

- McTrack
- Pnd...Point
- Pnd...Digi
- Pnd...Hit
- PndTrack, PndPidCandidate, PndPidProbability

PandaRoot Overview/Structure



macro/master/sim_complete.C

macro/master/digi_complete.C

macro/master/reco_complete.C

macro/master/pid_complete.C

macro/master/ana_complete.C

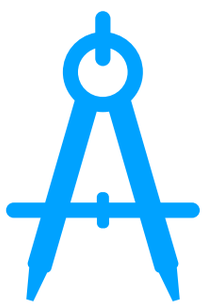
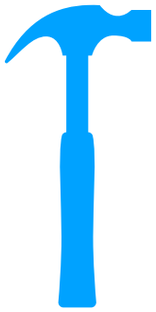
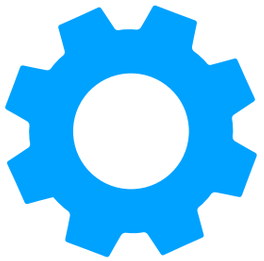
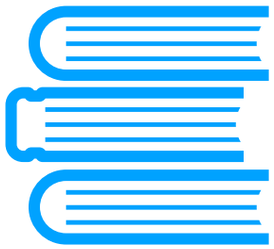
PandaRoot Build Process

- CMake based build:
  - Creates Makefiles (less messy to maintain)
  - Scripts to configure each package: CMakeLists.txt
  - Dependency handling

- Externals found via include & library path shell variables

- build/lib, build/bin house final products

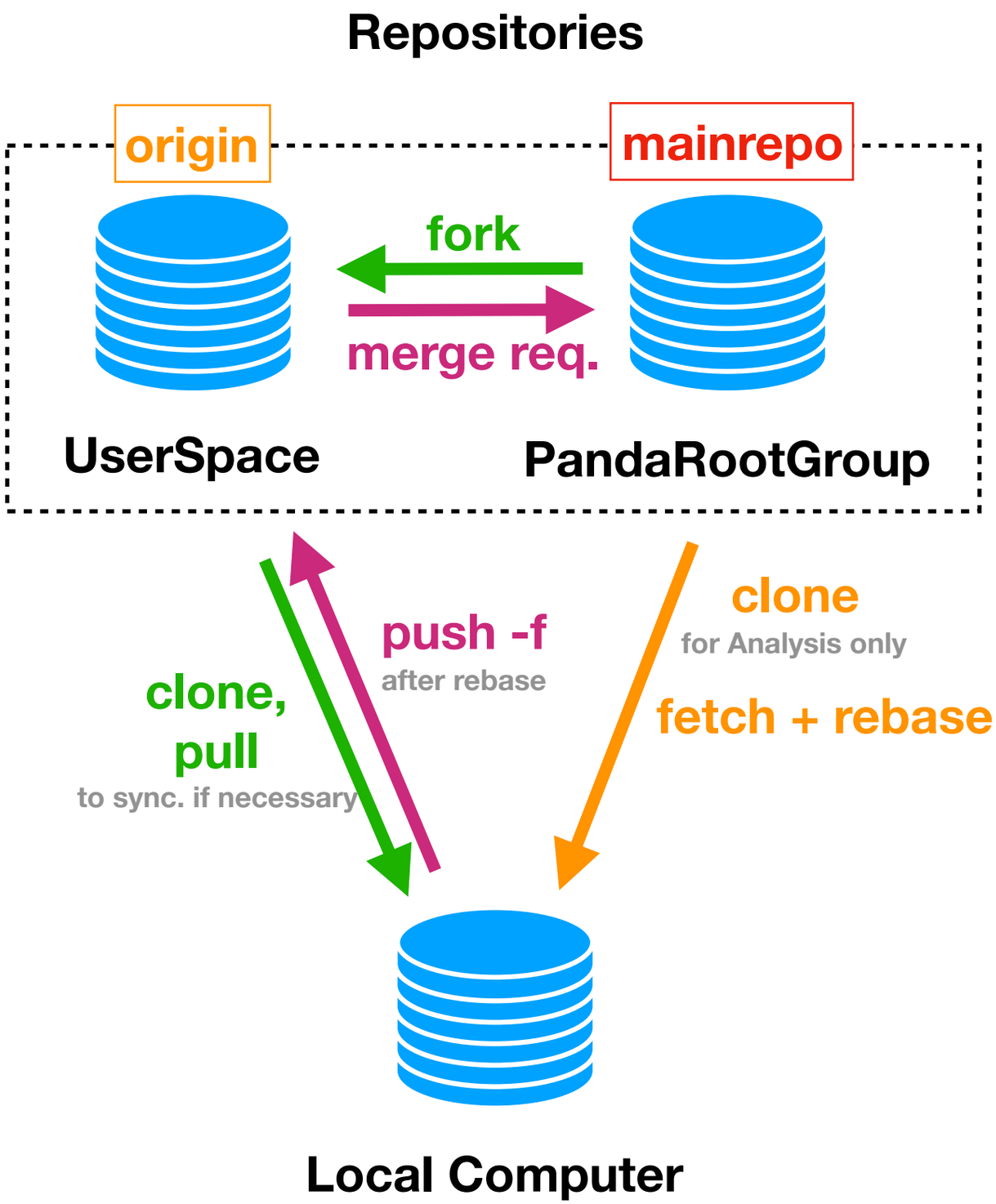
```
export SIMPATH=…
export FAIRROOTPATH=…
cmake <sourcepath>
make -j <#cpu>
```


Example CMakeLists.txt

```
Set(SYSTEM_INCLUDE_DIRECTORIES
  ${SYSTEM_INCLUDE_DIRECTORIES}
  ${ROOT_INCLUDE_DIR}
  ${BASE_INCLUDE_DIRECTORIES}
)

Set(INCLUDE_DIRECTORIES
  ${CMAKE_SOURCE_DIR}/pnddata
  ${CMAKE_SOURCE_DIR}/pnddata/TrackData
  ${CMAKE_SOURCE_DIR}/pnddata/PidData
)

Include_Directories(${INCLUDE_DIRECTORIES})
Include_Directories(SYSTEM ${SYSTEM_INCLUDE_DIRECTORIES})

set(LINK_DIRECTORIES
  ${ROOT_LIBRARY_DIR}
  ${FAIRROOT_LIBRARY_DIR}
)

link_directories( ${LINK_DIRECTORIES})

set(SRCS
  PndMytoolTask.cxx
  PndMytoolCalculate.cxx
  PndMytoolHelpers.cxx
)

set(LINKDEF  MytoolLinkDef.h)
set(LIBRARY_NAME Mytool)

set(DEPENDENCIES Base ParBase PndData generalTools)

GENERATE_LIBRARY()
```

A Small GIT Tutorial



0 **Fork once on the website**

1 **git clone**
https://pandaatfair.githost.io/<user>/PandaRoot.git

2 **git fetch mainrepo**
don't forget to rebase

3 **git push**
after rebase use git push -f

4 **Create Merge Request on website**

Shortcuts

- origin:
  ...githost.io/<user>/PandaRoot.git
- mainrepo:
  ...githost.io/PandaRootGroup/PandaRoot.git

A Small GIT Tutorial

Preparation

Create a Fork on the Website (once)

Setup git on each computer (once)

```
git config --global branch.autosetuprebase always
git config --global user.name "FirsName LastName"
git config --global user.email johndoe@example.com
git config --global core.ignorecase false
```

Download

```
git clone
  https://pandaatfair.githost.io/<user>/PandaRoot.git
git remote add mainrepo
  https://pandaatfair.githost.io/PandaRootGroup/PandaRoot.git
```

Switch to dev (optional)

```
(git fetch mainrepo)
git checkout -b dev mainrepo/dev
(git push -u origin dev)
```

Development Workflow

Create feature branch

```
(git fetch mainrepo)
git checkout -b featureXXX mainrepo/dev
git push -u origin featureXXX
```

Update news to feature branch

```
git fetch mainrepo
git checkout featureXXX
git rebase mainrepo/dev
```

Upload feature branch

```
git push -f origin
```

More details on panda-wiki.gsi.de/foswiki/bin/view/Computing/PandaRoot

PandaRoot Communication

- Code **Repository**: pandaatfair.githost.io
  - **Issue tracker**, including discussions
- **WiKi** page:
  panda-wiki.gsi.de/foswiki/bin/view/Computing/PandaRoot
- Discussion **forums**: forum.gsi.de
- Bi-weekly online **meetings**: Wed. 10-11
- **Dashboard**:

  https://cdash.gsi.de/index.php?project=PandaRoot

Thanks for your attention.