# ANAROOT: new online/offline framework for RIBF data analysis based on ROOT

Tadaaki Isobe for the ANAROOT development collaboration
RIKEN, Nishina Center, isobe@riken.jp
http://ribf.riken.jp/RIBFDAQ/index.php?Tools%2FAnalysis%2FANAROOT

## What is ANAROOT:

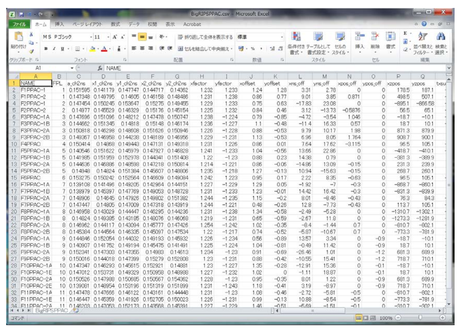
ANAROOT is the ROOT (root.cern.ch) based toolkit for the online/offline analysis at RIBF experiments. This software is optimized for the analysis of RIDF raw data produced by RIBF DAQ system (Babirl).

## Functionalities implemented in ANAROOT:

- RIDF raw data decoding.
- Data input through RIDF file, online shared memory and data streaming are supported.
- RIBF data reconstruction libraries.
  a. particle identification of secondary particles with ZeroDegree and/or BigRIPS.
  b. reconstruction code for SAMURAI.
  c. reconstruction code for EURICA including event building by using time-stamp information.
- CUI base UI
  a. ANAPAW like command lines with Nadeko library.
  b. User routine with AnaLoop.

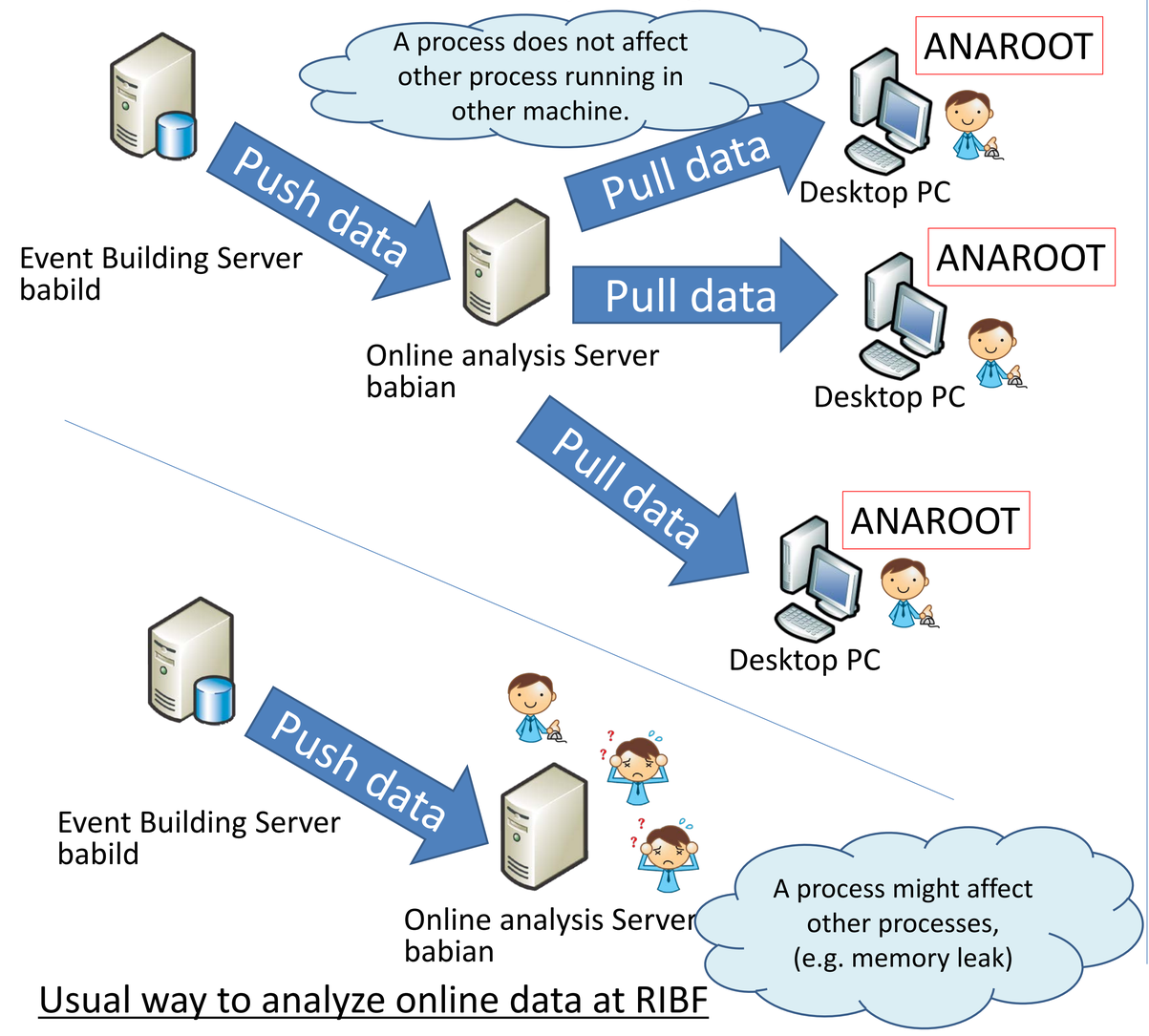
## Database handling in ANAROOT:

Database handling with XML files. Easier to modify, easier to handle, suitable for small/middle size data analysis, but it keeps to extend the database handling with database server, such as postgresql.



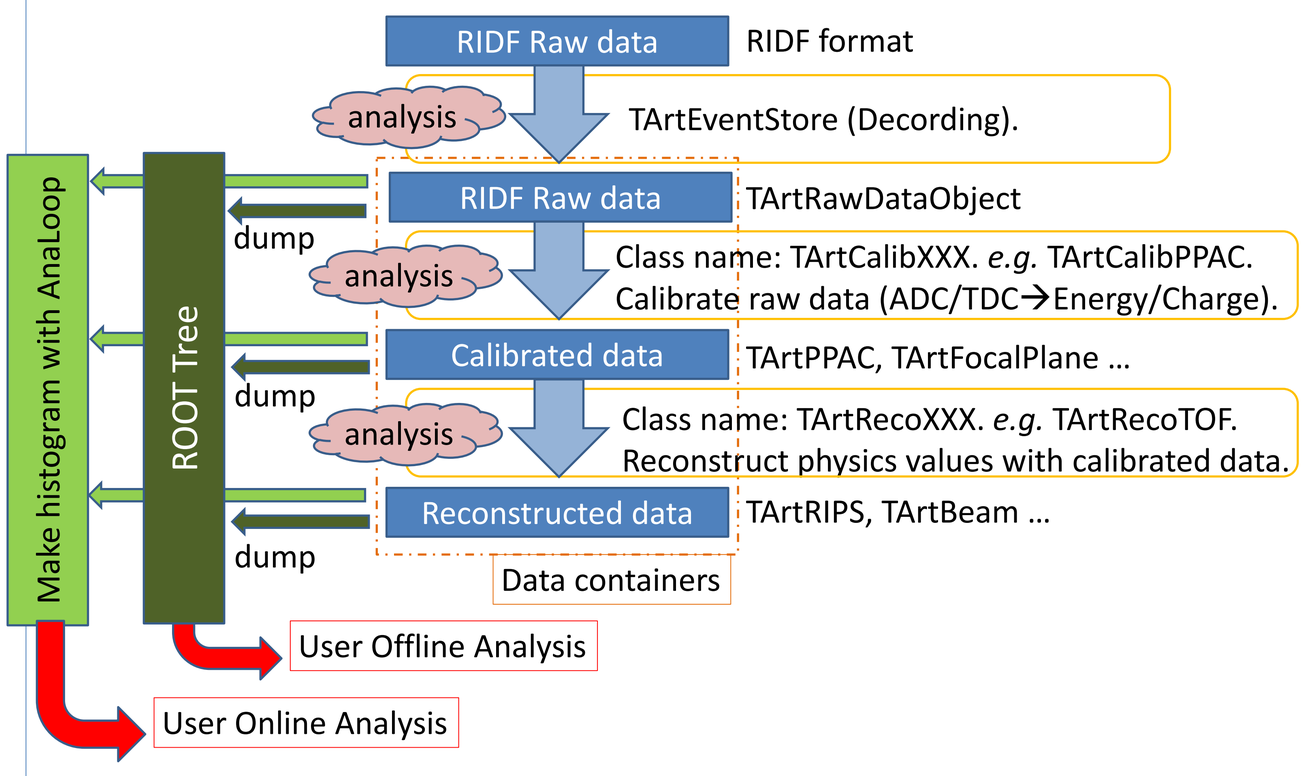
Editing PPAC DB with excel

## Distributed online computing with ANAROOT:

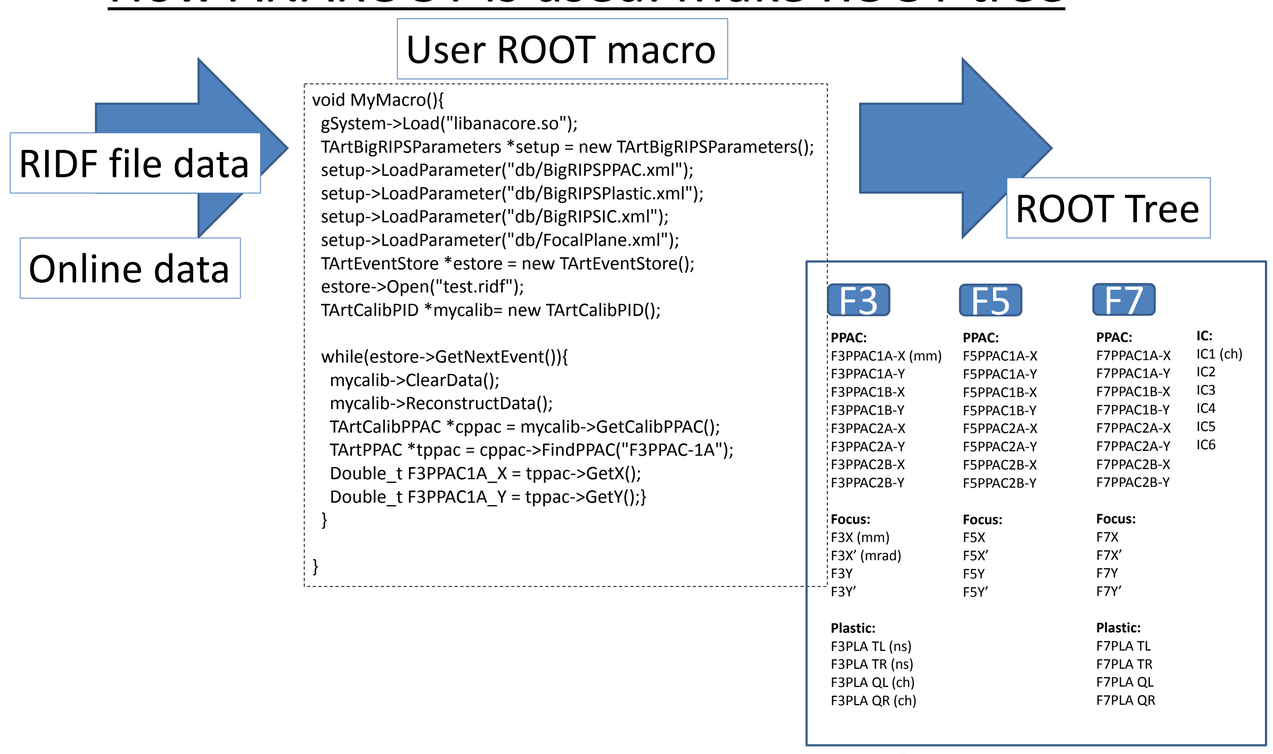



Usual way to analyze online data at RIBF

## ANAROOT Analysis Scheme:

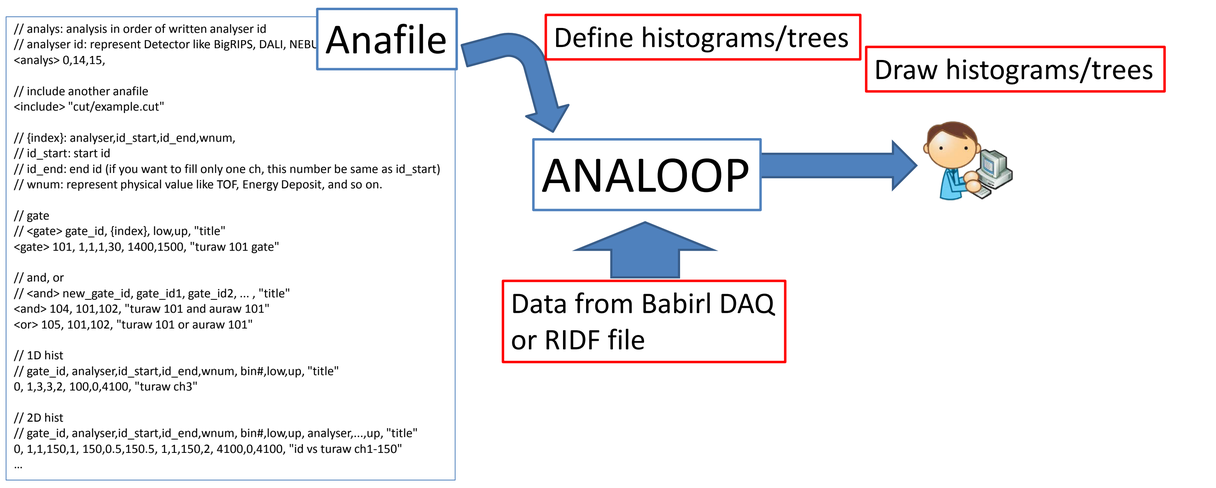



## How ANAROOT is used: Make ROOT tree

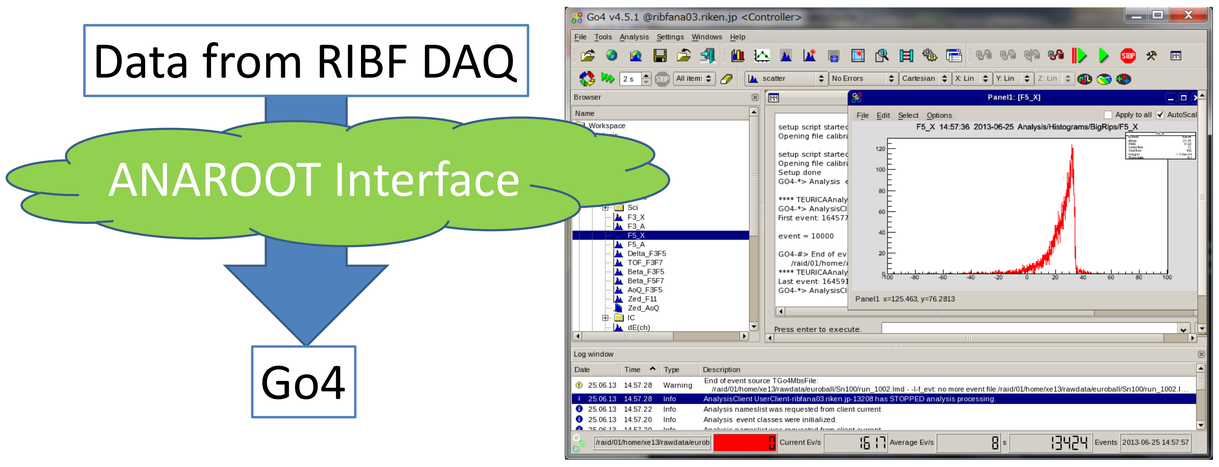



```
void MyMacro(){
  gSystem->Load("libanacore.so");
  TArtBigRIPSParameters *setup = new TArtBigRIPSParameters();
  setup->LoadParameter("db/BigRIPSPPAC.xml");
  setup->LoadParameter("db/BigRIPSPlastic.xml");
  setup->LoadParameter("db/BigRIPSIC.xml");
  setup->LoadParameter("db/FocalPlane.xml");
  TArtEventStore *estore = new TArtEventStore();
  estore->Open("test.ridf");
  TArtCalibPID *mycalib= new TArtCalibPID();

  while(estore->GetNextEvent()){
    mycalib->ClearData();
    mycalib->ReconstructData();
    TArtCalibPPAC *cppac = mycalib->GetCalibPPAC();
    TArtPPAC *tppac = cppac->FindPPAC("F3PPAC-1A");
    Double_t F3PPAC1A_X = tppac->GetX();
    Double_t F3PPAC1A_Y = tppac->GetY();}
  }

}
```

## How ANAROOT is used: AnaLoop/AnaFile CUI




```
// analys: analysis in order of written analyser id
// analyser id: represent Detector like BigRIPS, DALI, NEBU
<analys> 0,14,15,

// include another anafile
<include> "cut/example.cut"

// {index}: analyser,id_start,id_end,wnum,
// id_start: start id
// id_end: end id (if you want to fill only one ch, this number be same as id_start)
// wnum: represent physical value like TOF, Energy Deposit, and so on.

// gate
// <gate> gate_id, {index}, low,up, "title"
<gate> 101, 1,1,1,30, 1400,1500, "turaw 101 gate"

// and, or
// <and> new_gate_id, gate_id1, gate_id2, ... , "title"
<and> 104, 101,102, "turaw 101 and auraw 101"
<or> 105, 101,102, "turaw 101 or auraw 101"

// 1D hist
// gate_id, analyser,id_start,id_end,wnum, bin#,low,up, "title"
0, 1,3,3,2, 100,0,4100, "turaw ch3"

// 2D hist
// gate_id, analyser,id_start,id_end,wnum, bin#,low,up, analyser,...,up, "title"
0, 1,1,150,1, 150,0.5,150.5, 1,1,150,2, 4100,0,4100, "id vs turaw ch1-150"
...
```

## How ANAROOT is used: combination with Go4 (under testing)

http://www-win.gsi.de/go4/