# HIPO4 + CLAS12TOOL + ROOT

Derek Glazier, Gagik Gavalian for the CLAS Collaboration

06/03/19



gavalian / Clas12Tool				O Watch →	2 🕈 Star 0	۲ Fork 3
<> Code	() Issues 0	1) Pull requests 0 Projects 0 E Wi	ki <u>III</u> Insights			
CLAS12 Data Analysis Tools						
59 commits		<pre></pre>			<b>L</b> 3 contributors	
Branch: hipo	4 <del>-</del> New pull re	quest	Create new file	Upload files	Find file Clon	e or download <del>▼</del>
This branch is 6 commits ahead of master.				1) Pull reques	st 🖹 Compare	
<b>gavalian</b> added more examples, and recordbuilder Latest commit 4d57697 19 days				697 19 days ago		
Banks		added method detector position local position				3 months ago
C12AnC	рр	fix memory leak in particle making				3 months ago
ClasBan	ClasBanks added Hipo library, and cmake scripts. Needs revising					7 months ago
Example	xamples added more examples, and recordbuilder					19 days ago
Hipo		added new directory Hipo4 for reading 4th generat	25 days ago			
Hipo4		added more examples, and recordbuilder	19 days ago			
🚡 Lz4 @ e	95781d	Getting Iz4 submodule working				7 months ago
Utils		modified example				3 months ago
.gitignor	е	tuple and histo filling in algs				3 months ago
📄 .gitmodu	lles	Getting Iz4 submodule working				7 months ago
Makefile	)	added new directory Hipo4 for reading 4th generat	ion file format.			25 days ago
	E.md	fixed the README file				3 months ago



#### ReadEvents.cc

```
//open hipo4 file for reading
hipo::reader reader;
reader.open(inputFile);
```

```
//read dictionary ~ bank configuration
hipo::dictionary factory;
reader.readDictionary(factory);
```

```
//event for holding data
hipo::event event;
//particular bank we are interested in
hipo::bank PART(factory.getSchema("REC::Particle"));
```

```
while(reader.next()==true){//loop over all events
    reader.read(event); //read event
    event.getStructure(PART); //get particle data
```

```
int nrows = PART.getRows();
```

```
for(int i = 0; i < nrows; i++){
    int    pid = PART.getInt("pid",i);
    float    px = PART.getFloat("px",i);
    float    py = PART.getFloat("py",i);
    float    pz = PART.getFloat("pz",i);</pre>
```



#### New clas12reader class

```
//reader
hipo::reader
              reader;
hipo::event
               event;
//DST banks
head ptr bhead;
par_ptr _bparts;
mcpar_ptr _bmcparts;
covmat_ptr _bcovmat;
cal ptr bcal;
scint_ptr _bscint;
trck_ptr _btrck;
                                   region particles
traj ptr btraj;
                                   = particle +
cher ptr bcher;
ft ptr bft;
                                   associated detector info
vtp_ptr _bvtp;
scaler_ptr _bscal;
                                   Seperate class for regions:
//Detector region vectors,
//each particle in an event will have
                                   FT, FD, CD
//one associated
std::vector<region fdet ptr> rfdets;
std::vector<region cdet ptr> rcdets;
std::vector<region ft ptr> rfts;
std::vector<region_part_ptr> _detParticles;
```

```
//create the clas12 event reader
clas12reader c12event(inputFile);
```

```
while(c12event.next()==true){//loop over all events
    //loop over particles
    for(auto& p : c12event.getDetParticles()){
        int pid = p->par()->getPid();
        float px = p->par()->getPx();
        float py = p->par()->getPx();
        float pz = p->par()->getPx();
    }
}
```



```
//create the clas12 event reader
clas12reader c12event(inputFile);
while(c12event.next()==true){//loop over all events
  //loop over particles
  for(auto& p : c12event.getDetParticles()){
                                         // get predefined selected information
           pid = p->par()->getPid();
    int
                                         p->getTime();
    float px = p->par()->getPx();
                                         p->getDetEnergy();
    float py = p->par()->getPx();
                                         // get any detector information (if exists for this particle)
                                         // there should be a get function for any entry in the bank
    float pz = p->par()->getPx();
                                         switch(p->getRegion()) {
                                         case FD :
                                           p->cal(PCAL)->getEnergy();
                                           p->cal(ECIN)->getEnergy();
                                           p->cal(ECOUT)->getEnergy();
                                           p->sci(FTOF1A)->getEnergy();
                                           p->sci(FTOF1B)->getEnergy();
                                           p->sci(FTOF2)->getEnergy();
                                           p->trk(DC)->getSector();
                                           p->che(HTCC)->getNphe();
                                           p->che(LTCC)->getNphe();
 Only need to loop over
                                           p->traj(TRAJ_HTCC)->getX();
                                           break:
 Particle!
                                         case FT :
                                           p->ft(FTCAL)->getEnergy();
                                           p->ft(FTHODO)->getEnergy();
                                           break:
                                         case CD:
                                           p->sci(CTOF)->getEnergy();
                                           p->sci(CND)->getEnergy();
                                           break;
```

### Example data analysis (I)

```
// get particles by type
auto electrons=event.getByID(11);
auto gammas=event.getByID(22);
auto protons=event.getByID(2212);
auto pips=event.getByID(211);
auto pims=event.getByID(-211);
if(electrons.size()==1 && gammas.size()==2 && protons.size()==1 &&
   pips.size()==1 &&pims.size() == 1){
  // set the particle momentum
 // i.e. auto el = electrons[0]->p4(); etc.
 SetLorentzVector(el,electrons[0]);
  SetLorentzVector(pr,protons[0]);
  SetLorentzVector(g1,gammas[0]);
  SetLorentzVector(g2,gammas[1]);
  SetLorentzVector(pip,pips[0]);
  SetLorentzVector(pim.pims[0]);
  TLorentzVector miss=beam+target-*el-*pr-*g1-*g2-*pip-*pim;
  hmiss->Fill(miss.M2());
  TLorentzVector pi0 = *q1+*q2;
  hm2g->Fill(pi0.M());
  if(TMath::Abs(miss.M2())<0.5)hm2gCut->Fill(pi0.M());
```





# Full Reconstructed data file 1 File 1.7G; 48k events

#### Hipo4 Loops

Just reader.next()	1.4s
+ read(event)	1.6s
+ getStructure(PART)	1.6s
+ all DST banks	1.7s
+sort(particle detectors)	1.8s
+analyse	1.9s

\*previous Hipo3 event gave ~2.3s for sort
\*\* this was for files already in file
buffer cache!



# DST data file 1 File 2.2G = 26 actual files; 1.25M events

#### Hipo4 Loops

Just reader.next()	1.4s
+ read(event)	1.7s
+ getStructure(PART)	1.8s
+ all DST banks	2.7s
+sort(particle-detectors)	3.7s
+analyse	5.9s

\*previous Hipo3 event gave ~16s for sort
\*\* this was for files already in file
buffer cache!



# ROOT Loops on 1.21M events Just tree.GetEntry()(all banks) 38s (2.7s) SetBranchStatus(REC::Particle,1) 6s (1.8s)

#### \*\* this was for files already in file buffer cache!



Read Speed for Run 4152, on my laptop Include pindex sort for Hipo file (fair comparison)

FILE	HDD	SSD	BUFFER CACHE	Nick SSD? Buffer?	Nick SSD? Buffer?
My ROOT NOT 4152	44s	44s	38s		
dst2root conversion 458secs	58s	37s	32s	32s	
C++ hipo4	26s	9s	4s		
Java hipo4					*12s Sort?

For dst2root see Nick Tyler's presentation
https://clas12-docdb.jlab.org/DocDB/0003/000350/001/dst2root.pdf

dst2root missing TRCK, TRAJ,.. banks

Indicitive values, there is a lot of fluctuation in some cases Not accounted for buffer cache on lustre server

		Read /volatile	Read /cache	Read /scratch	Read buffer
	Hipo4 2.2Gb	33s	21s	36s	19s
1000s Conversion time	dst2root 1.8Gb No TRCK, TRAJ	?160?s	60s	75s	50s

Rates vary a lot when doing the same thing on different days Rates very different on different hardware Hipo results relatively consistent Needs redone with ~100Gb of data Parse strings into compiled C++ code

Use clas12reader for histogramming particle info

e.g can draw time versus momentum, calculated mass. Currently includes :

1D, 2D, Cuts, Formula, Chains, Lazy execution

**ParticleHist** Correlated particle - detector information

#### BankHist

Fast draw of individual bank items



#### Compare ROOT::TTree::Draw()

21M events



#### Compare HIPO::ParticleHist::Draw()

ParticleHist hists("clas\_004152.recon.hipo"); 1.25M events
hists.Hist1D("PBANK.Px",1000,-5,5,"PBANK.Px")->Draw("");



NOTE 2s compilation overhead to be added on loop time : Real Time = 4.07 seconds Cpu Time = 4.07 seconds



#### Compare HIPO::ParticleHist::Draw()

1.27M events ParticleHist hists("clas 004152.recon.hipo"); hists.Hist1D("sqrt(PBANK.Px\*PBANK.Px+PBANK.Py\*PBANK.Py+ PBANK.Pz\*PBANK.Pz)",1000,0,5,"PBANK.Px")->Draw("");



NOTE 2s compilation overhead to be added on loop time : Real Time = 4.63 seconds Cpu Time = 4.63 seconds ROOT::Ttree::Draw ~7s



### ParticleHist parser

In interactive ROOT

```
ParticleHist hists("my/hipo/file.hipo");
hists.Hist1D("P.Theta*TMath::RadToDeg()",180,0,180);
hists.Hist1D("P.Phi*TMath::RadToDeg()",180,-180,180);
hists.Hist1D("P.P",100,0.1,12);
hists.Hist1D("P.Time-EVNT.StartTime",1000,-200,200,"P.Time")
                                                        ->Draw("(2x2)");
                    Becomes ...
    While(c12.next()==true){
       for(auto& p : *particles) {
         if((p->getTime()))hists->at(3)->Fill((p->getTime())-
                                       (c12.head()->getStartTime()));
        if(1)hists->at(2)->Fill((p->getP()));
        if(1)hists->at(1)->Fill((p->getPhi())*TMath::RadToDeg());
        if(1)hists->at(0)->Fill((p->getTheta())*TMath::RadToDeg());
```

1.2M events Standard read/sort ~ 3.7s
With 1 histogram Draw no cuts ~ 4s



# Particle hists, 1.27M events Cut on Pid



Time = 5.6s



















# 2D Theta versus phi, 1.27M events Cut on Pid





Time = 4.8s

150

50

-100

-150

-50

-100

-150

# Experimental::HipoTreeMaker Create a Ndim tree rather than histogram

#### hiporoot::ParticleTree treemaker("clas\_004152.recon.hipo","test.root");

```
treemaker.Branch("P.Time/F");
treemaker.Branch("PBANK.Px/F");
treemaker.Branch("PBANK.Py/F");
treemaker.Branch("PBANK.Pz/F");
treemaker.Branch("PBANK.Vx/F");
treemaker.Branch("PBANK.Vy/F");
treemaker.Branch("PBANK.Vz/F");
treemaker.Branch("PBANK.Pid/I");
```

Includes filters on particle conditions Or event condition (Nelectrons etc)

```
// treemaker.AddParticleCut("PBANK.Pid==11");
```



## HipoSelector and PROOF

Run multicore with ROOT-PROOF

Similar to TChain based TSelector

Data access via clas12reader

Splits data on hipo::records

Run in interactive ROOT hiporoot::HipoChain chain; chain.Add("my/hipo/files\*.hipo");

hiporoot::HipoSelector sel(&chain);

}

gProof->Process(&sel,chain.GetNRecords());

void HipoSelector::SlaveBegin(TTree \* /\*tree\*/)

// The SlaveBegin() function is called after the Begin() function.
// When running with PROOF SlaveBegin() is called on each slave server.
// The tree argument is deprecated (on PROOF 0 is passed).

TString option = GetOption(); \_chain=dynamic\_cast<HipoChain\*>(fInput->FindObject("HIPOFILES"));

```
_hist1=new TH1F("Time","Time",600,-100,500);
fOutput->Add(_hist1);
```

Bool\_t HipoSelector::ProcessEvent(){

\_hist1->Fill(\_c12->head()->getStartTime()); return kTRUE;





<pre>@ROOT Ex_Python3 Last Checkpoint: 21 hours ago (unsaved changes)</pre>		Logout Terminal	
File Edit View Insert Cell Kernel Widgets Help	Trusted	Python 3	0
B + ≫ C2 E ↑ ↓ NRun ■ C ≫ Code ▼			
In [ ]: import ROOT			
<pre>In [ ]: ROOT.gROOT.ProcessLine(".x \$CLAS12TOOL/RunRoot/importToROOT.C+")</pre>			
<pre>In [ ]: c12 = R00T.clas12.clas12reader("/home/dglazier/fastdata/hipo4test/filtered/clas_004152.recon.hipo");</pre>			
<pre>In []: el=R00T.TLorentzVector(0,0,1,1) pr=R00T.TLorentzVector(0,0,0,1) R00T.gBenchmark.Start("python")</pre>			
<pre>In []: while cl2.next() :     cl2.head().getStartTime()     electrons=cl2.getByID(11)     if electrons.size()==1 :         el.SetXYZM(electrons[0].par().getPx(),electrons[0].par().getPy(),electrons[0].par().getPz(), 0.0005109989:</pre>	1)		
<pre>In [ ]: R00T.gBenchmark.Stop("python") R00T.gBenchmark.Print("python")</pre>			
But why would you ? Time 20s (python is slower)			
Can be used with HipoHist/Tree with no time penalty	′ <b>C</b>	IQS	

## HSScheme





```
New hipo4 c++ library is working well
```

```
For reading full CLAS12 DSTs (IN BUFFER CACHE)
HIPO4 ~ 4x faster than HIPO3
HIPO4 ~ 2-10(!)x faster than ROOT file
```

Additional clas12reader class correlates particle and detector information (No additional loops required!) Can be embedded in any ROOT based analysis scheme

Experimental HipoDraw can perform operations similar to TTree::Draw and RDataFrame

Scales very well for multiple branch histograms

HipoTreeMaker can skim and filter Hipo file into simple ROOT trees

HipoSelector can be used to run multi-core with PROOF

What else is needed ?

