Subject: [FIXED] Propagate Functions in PndAnalysis
Posted by Andreas Pitka on Thu, 02 May 2013 17:21:18 GMT
View Forum Message <> Reply to Message

Hi,

should it be possible to directly invoke the Propagate functions of PndAnalysis (like PropagateToIp (TCandidate *cand)) in an analysis macro?

When i try something simple like simulating single pions with the particle gun and in the analysis doing:

for (int i=0;i<pip.GetLength();i++)
{theAnalysis->PropagateToIp(&pip[i]);}

(where pip is the filled TCandList and theAnalysis is the PndAnalysis) i get a segmentation violation:

#12 0xb0b3af47 in FairGeanePro::FairGeanePro (this=0x103bea78) at
/home/pitka/uni/Pandaroot/Rev19237/geane/FairGeanePro.cxx:51
#13 0xafee22f7 in PndAnalysis::Propagator (this=0x10313678, mode=1, tStart=...,
cand=0x103b73c0, mypoint=0x103be5f8, skipcov=false) at
/home/pitka/uni/Pandaroot/Rev19237/PndTools/AnalysisTools/PndAnalysis.cx x:481
#14 0xafee1c0c in PndAnalysis::PropagateToPoint (this=0x10313678, cand=0x103b73c0,
mypoint=0x103be5f8) at
/home/pitka/uni/Pandaroot/Rev19237/PndTools/AnalysisTools/PndAnalysis.cx x:420
#15 0xafee1998 in PndAnalysis::PropagateToIp (this=0x10313678, cand=0x103b73c0) at
/home/pitka/uni/Pandaroot/Rev19237/PndTools/AnalysisTools/PndAnalysis.cx x:385

Best regards and thanks a lot

Andreas

Subject: Re: Propagate Functions in PndAnalysis
Posted by StefanoSpataro on Thu, 02 May 2013 17:43:30 GMT
View Forum Message <> Reply to Message

In theory the candidates are already propagated to the IP.
However, could you please write the full error backtrace?

Subject: Re: Propagate Functions in PndAnalysis
Posted by Andreas Pitka on Fri, 03 May 2013 06:36:54 GMT
View Forum Message <> Reply to Message

Actually i am more interested in the PropagateToPoint (TCandidate *cand, TVector3 *mypoint) function to get the momenta of pions from a K short decay at the correct point, but there the error is exactly the same.

The full backtrace is:

==========
There was a crash (#7 0xb721c65b in SigHandler(ESignals) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34).
This is the entire stack trace of all threads:
==========
#0 0xb77ae424 in __kernel_vsyscall ()
#1 0xb6dcc9e3 in __waitpid_nocancel () at ../sysdeps/unix/syscall-template.S:82
#2 0xb6d53ff3 in do_system (line=0xffe9258
"/home/pitka/uni/fairroot/apr13_src/../apr13/etc/gdb-backtrace.sh 2442 1>&2") at
../sysdeps/posix/system.c:149
#3 0xb6ecce2b in system (line=0xffe9258
"/home/pitka/uni/fairroot/apr13_src/../apr13/etc/gdb-backtrace.sh 2442 1>&2") at
pt-system.c:29
#4 0xb721575b in TUnixSystem::Exec(char const*) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#5 0xb7219a40 in TUnixSystem::StackTrace() () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#6 0xb721c547 in TUnixSystem::DispatchSignals(ESignals) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#7 0xb721c65b in SigHandler(ESignals) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#8 0xb7213832 in sighandler(int) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#9 0xb7246aa5 in textinput::TerminalConfigUnix::HandleSignal(int) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#10 0xb7246ae4 in (anonymous namespace)::TerminalConfigUnix__handleSignal(int) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#11 <signal handler called>
#12 0xb0bbef47 in FairGeanePro::FairGeanePro (this=0xffe8f90) at
/home/pitka/uni/Pandaroot/Rev19237/geane/FairGeanePro.cxx:51
#13 0xaff662f7 in PndAnalysis::Propagator (this=0xff29f08, mode=1, tStart=..., cand=0xffcfee0,
mypoint=0xffe8b10, skipcov=false) at
/home/pitka/uni/Pandaroot/Rev19237/PndTools/AnalysisTools/PndAnalysis.cx x:481
#14 0xaff65c0c in PndAnalysis::PropagateToPoint (this=0xff29f08, cand=0xffcfee0,
mypoint=0xffe8b10) at
/home/pitka/uni/Pandaroot/Rev19237/PndTools/AnalysisTools/PndAnalysis.cx x:420
#15 0xafff16b9 in G__AnalysisToolsDict_898_0_11 (result7=0xbffe7158, funcname=0xff28560
"", libp=0xbffdcb14, hash=0) at
/home/pitka/uni/Pandaroot/Rev19237/build/PndTools/AnalysisTools/Analysis
ToolsDict.cxx:7283
#16 0xb675a50c in Cint::G__ExceptionWrapper(int (*)(G__value*, char const*, G__param*,
int), G__value*, char*, G__param*, int) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#17 0xb680b969 in G__execute_call () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#18 0xb680bd2c in G__call_cppfunc () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#19 0xb67e882f in G__interpret_func () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#20 0xb67d6562 in G__getfunction () from

/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#21 0xb68d0291 in G__getstructmem(int, G__FastAllocString&, char*, int, char*, int*, G__var_array*, int) () from /home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#22 0xb68c84b3 in G__getvariable () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#23 0xb67ac98a in G__getitem () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#24 0xb67b3d38 in G__getexpr () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#25 0xb6840e38 in G__exec_statement () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#26 0xb684964a in G__exec_loop(char const*, char*, std::list<G__FastAllocString, std::allocator<G__FastAllocString> > const&) [clone .constprop.69] () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#27 0xb6846716 in G__exec_statement () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#28 0xb684964a in G__exec_loop(char const*, char*, std::list<G__FastAllocString, std::allocator<G__FastAllocString> > const&) [clone .constprop.69] () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#29 0xb6841d6e in G__exec_statement () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#30 0xb67e9cc9 in G__interpret_func () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#31 0xb67d65d8 in G__getfunction () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#32 0xb67ad215 in G__getitem () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#33 0xb67b3d38 in G__getexpr () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#34 0xb67bfd23 in G__calc_internal () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#35 0xb68507c4 in G__process_cmd () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCint.so.5.34
#36 0xb71e071a in TCint::ProcessLine(char const*, TInterpreter::EErrorCode*) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#37 0xb71d604f in TCint::ProcessLineSynch(char const*, TInterpreter::EErrorCode*) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#38 0xb7133d97 in TApplication::ExecuteFile(char const*, int*, bool) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#39 0xb713416c in TApplication::ProcessFile(char const*, int*, bool) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#40 0xb71328ff in TApplication::ProcessLine(char const*, bool, int*) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libCore.so.5.34
#41 0xb7004b1f in TRint::Run(bool) () from
/home/pitka/uni/fairroot/apr13_src/../apr13/lib/root/libRint.so.5.34
#42 0x08048c8f in main ()
==========

Subject: Re: Propagate Functions in PndAnalysis

Posted by StefanoSpataro on Fri, 03 May 2013 14:48:30 GMT
View Forum Message <> Reply to Message

Hi,
I suppose you are not using neither the trunk neither the release.
Could you please write what is in the following line:

/home/pitka/uni/Pandaroot/Rev19237/geane/FairGeanePro.cxx:51

To understand what is crashing. In my feairgeanepro version the line 51 cannot produce crash.

Subject: Re: Propagate Functions in PndAnalysis
Posted by Andreas Pitka on Mon, 06 May 2013 15:18:02 GMT
View Forum Message <> Reply to Message

Hi,

i also tried it with the apr13 release, there its the same.

the line is:

fApp(FairGeaneApplication::Instance())

Subject: Re: Propagate Functions in PndAnalysis
Posted by StefanoSpataro on Mon, 06 May 2013 15:36:10 GMT
View Forum Message <> Reply to Message

Let's see with the new rho/pid interface.

Subject: Re: Propagate Functions in PndAnalysis
Posted by StefanoSpataro on Tue, 02 Jul 2013 14:20:07 GMT
View Forum Message <> Reply to Message

Could you please try the new trunk, to see if the problem is still persistent?

Subject: Re: Propagate Functions in PndAnalysis
Posted by Ralf Kliemt on Thu, 04 Jul 2013 14:36:30 GMT
View Forum Message <> Reply to Message

I think I found the issue:

you need these two lines in the analysis macro before calling the Init:

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

Cheers
Ralf

Subject: Re: Propagate Functions in PndAnalysis
Posted by Andreas Pitka on Thu, 04 Jul 2013 15:28:06 GMT
View Forum Message <> Reply to Message

There is a segmentation violation if i try to call the default constructor (the most simple macro to reproduce this is):

```
void test()
{

  gROOT->LoadMacro("$VMCWORKDIR/gconfig/rootlogon.C");

  FairGeanePro* Geane = new FairGeanePro();

}
```

I tried it with Rev 20624

Subject: Re: Propagate Functions in PndAnalysis
Posted by StefanoSpataro on Thu, 04 Jul 2013 15:38:58 GMT
View Forum Message <> Reply to Message

FairGeane, not FairGeanePro.

Subject: Re: Propagate Functions in PndAnalysis
Posted by Andreas Pitka on Thu, 04 Jul 2013 16:35:15 GMT
View Forum Message <> Reply to Message

doh.. you are right, now its working.

thanks a lot

Subject: Re: Propagate Functions in PndAnalysis
Posted by Ralf Kliemt on Fri, 05 Jul 2013 06:39:52 GMT
View Forum Message <> Reply to Message

Hi Andreas,

I suppose it is important where to put that line in the initialization. For reference I post the head of my macro in all it's glory. (We don't need rootlogon.C anymore and Tools.C moved to macro/run)

```
void testParticles ( int nevts=0 )
{
  // *** some variables
  int i=0,j=0, k=0, l=0;
  TString plotfile="datakin/testParticles.root";
  TString OutFile="datakin/dummyOut.root";

  // *** the files coming from the simulation
  TString inPidFile  = "datakin/pid_complete.root";    // this file contains the PndPidCandidates
  TString inRecoFile = "datakin/reco_complete.root";
  TString inDigiFile = "datakin/digi_complete.root";
  TString inSimFile  = "datakin/sim_complete.root";  // this file contains the MC truth
  TString inParFile  = "datakin/simparams.root";

  gStyle->SetOptFit ( 1011 );
  gROOT->LoadMacro ( "$VMCWORKDIR/macro/run/Tools.C" );
  ImproveDefaultStyle();

  FairLogger::GetLogger()->SetLogToFile ( kFALSE );

  // *** initialization
  FairRunAna* fRun = new FairRunAna();
  FairRuntimeDb* rtdb = fRun->GetRuntimeDb();
  fRun->SetInputFile ( inSimFile );
  fRun->AddFriend ( inDigiFile );
  fRun->AddFriend ( inRecoFile );
  fRun->AddFriend ( inPidFile );

  FairParRootFileIo* parIO = new FairParRootFileIo();
  parIO->open ( inParFile );
  rtdb->setFirstInput ( parIO );
  rtdb->setOutput ( parIO );

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

As I see it you want to create a FairGeanePro object without initializing FairGeane, the thing which interfaces to the Field, Geo and Framework.

Cheers
Ralf
