Subject: crush at the end of pid macro
Posted by Dmitry Khaneft on Mon, 12 Mar 2012 15:02:00 GMT
View Forum Message <> Reply to Message

Dear all,

I use trunk 14823 and many simulations are crushes. Here is a log:

Quote:
...
===== PndPidCorrelator - Event: 998 - Number of tracks for pid 2 - Number of EMC Cluster for pid 7
-W- PndPidCorrelator::GetMvdInfo: Track perpendicular to MVD strip/pixel! Not added to MVD eloss
 ===== PndPidCorrelator - Event: 999 - Number of tracks for pid 2 - Number of EMC Cluster for pid 8
 ===== PndPidCorrelator - Event: 1000 - Number of tracks for pid 2 - Number of EMC Cluster for pid 5
-W- PndPidCorrelator::GetMvdInfo: Track perpendicular to MVD strip/pixel! Not added to MVD eloss

 *** Break *** segmentation violation
 Generating stack trace...
 0x00007f694b1977eb in FairTask::FinishTasks() at /data/work/kpha4/khaneftd/trunk_14823/base/FairTask.cxx:153 from /data/work/kpha4/khaneftd/build_14823/lib/libBase.so
 0x00007f694b197496 in FairTask::FinishTask() at /data/work/kpha4/khaneftd/trunk_14823/base/FairTask.cxx:96 from /data/work/kpha4/khaneftd/build_14823/lib/libBase.so
 0x00007f694b1925de in FairRunAna::Run(int, int) at /data/work/kpha4/khaneftd/trunk_14823/base/FairRunAna.cxx:403 from /data/work/kpha4/khaneftd/build_14823/lib/libBase.so
 0x00007f694b20455b in <unknown> from /data/work/kpha4/khaneftd/build_14823/lib/libBase.so
 0x00007f69560fc069 in Cint::G__ExceptionWrapper(int (*)(G__value*, char const*, G__param*, int), G__value*, char*, G__param*, int) + 0x39 from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f695619e7c1 in G__execute_call + 0x61 from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f695619f612 in G__call_cppfunc + 0x292 from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f695617cdb8 in G__interpret_func + 0x1d98 from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f695616ab0c in G__getfunction + 0x18cc from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f695625b738 in G__getstructmem(int, G__FastAllocString&, char*, int, char*, int*, G__var_array*, int) + 0x6e8 from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f6956251ef8 in G__getvariable + 0x1ea8 from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f69561480a1 in G__getitem + 0x81 from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32

```
 0x00007f695614d368 in G__getexpr + 0x4058 from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f69561cc76f in G__exec_statement + 0x638f from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f695617e195 in G__interpret_func + 0x3175 from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f695616ab6b in G__getfunction + 0x192b from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f695614881a in G__getitem + 0x7fa from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f695614d368 in G__getexpr + 0x4058 from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f6956156868 in G__calc_internal + 0x3f8 from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f69561dbc46 in G__process_cmd + 0x4ec6 from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCint.so.5.32
 0x00007f6956c3dbf6 in TCint::ProcessLine(char const*, TInterpreter::EErrorCode*) + 0x536
from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCore.so.5.32
 0x00007f6956c3b463 in TCint::ProcessLineSynch(char const*, TInterpreter::EErrorCode*) +
0x103 from /cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCore.so.5.32
 0x00007f6956babed8 in TApplication::ExecuteFile(char const*, int*, bool) + 0x848 from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCore.so.5.32
 0x00007f6956ba9f93 in TApplication::ProcessLine(char const*, bool, int*) + 0x7e3 from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libCore.so.5.32
 0x00007f6955e8e438 in TRint::Run(bool) + 0x498 from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/lib/libRint.so.5.32
 0x000000000040106c in main + 0x4c from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/bin/root.exe
 0x00007f6954f58c5d in __libc_start_main + 0xfd from /lib64/libc.so.6
 0x0000000000400f39 in <unknown> from
/cluster/gsi/fairsoft/gcc/fairsoft_jan12/tools/root/bin/root.exe
Function run_pid_stt_evt() busy flag cleared

 MZSTOR.  ZEBRA table base TAB(0) in /MZCC/ at adr   259696927     F7AA91F HEX

 MZSTOR.  Initialize Store  0  in /GCBANK/
        with Store/Table at absolute adrs   259719965   259696927
                                  HEX     F7B031D     F7AA91F
                                  HEX        560A           0
                        relative adrs       22026           0
        with     1 Str. in     2 Links in   5300 Low words in 4999970 words.
        This store has a fence of   16 words.

 MZLOGL.  Set Log Level 0 for store  0
1*****  GEANT Version  3.21/11 Released on 100298
0*****  Correction Cradle Version  0.1100

 MZDIV.   Initialize Division  Constant  in Store  0
        NW/NWMAX=   20004000000,  MODE/KIND=  1  2
        Division 20 initialized.

 MZLINK.  Initialize Link Area  /GCLINK/  for Store  0 NL/NS=    20    20
```

MZLINK. Initialize Link Area /GCSLNK/ for Store 0 NL/NS=   100   100

Calculating cross section tables, see gphysi.dat for more information

Cross section calculation concluded successfully

At this point the macro is almost finished, all events are processed (1000 in total) and have to be saved in a file.

Does anybody know what is wrong or experienced the same problem?

Cheers,
Dmitry

Subject: Re: crush at the end of pid macro
Posted by StefanoSpataro on Mon, 12 Mar 2012 15:12:03 GMT
View Forum Message <> Reply to Message

How are you launching such macro? Do you use a script?

Subject: Re: crush at the end of pid macro
Posted by Ralf Kliemt on Mon, 12 Mar 2012 15:25:56 GMT
View Forum Message <> Reply to Message

Hi,

The backtrace goes to FairTask:153. There you find:

```
149 void FairTask::FinishTasks()
150 {
151   TIter next(GetListOfTasks());
152   FairTask* task;
153   while( ( task=dynamic_cast<FairTask*>(next()) ) ) { task->FinishTask(); }
154   while( ( task=dynamic_cast<FairTask*>(next()) ) ) { task->Finish(); }
155 }
```

I ask myself why Finish() (to be done after each event) is not called before FinishTask().
A check if the pointer is healthy wouldn't hurt, too.

Cheers.
 Ralf

Subject: Re: crush at the end of pid macro
Posted by Dmitry Khaneft on Mon, 12 Mar 2012 15:33:59 GMT
View Forum Message <> Reply to Message

Stefano Spataro wrote on Mon, 12 March 2012 16:12How are you launching such macro? Do you use a script?

I use a bash script which basically says

root -l -q -b run_pid_stt_evt.C'("epem_mom3.3_gegm3.0_seed1000")'

I did't met this problem in nov11 release or in any previous versions.

Subject: Re: crush at the end of pid macro
Posted by StefanoSpataro on Mon, 12 Mar 2012 15:37:27 GMT
View Forum Message <> Reply to Message

Take out such "-l" and it should work. I had the same problem.

Subject: Re: crush at the end of pid macro
Posted by Dmitry Khaneft on Tue, 13 Mar 2012 00:00:50 GMT
View Forum Message <> Reply to Message

Stefano Spataro wrote on Mon, 12 March 2012 16:37Take out such "-l" and it should work. I had the same problem.
Still have the same problem
