

art - Feature #5217

File opening slow due to parsing of saved configuration.

01/17/2014 10:49 AM - Andrei Gaponenko

Status:	Closed	Start date:	01/17/2014
Priority:	Low	Due date:	
Assignee:	Christopher Green	% Done:	100%
Category:	Metadata	Estimated time:	120.00 hours
Target version:	1.11.00	Spent time:	0.00 hour
Scope:	Internal	SSI Package:	-
Experiment:	Mu2e		

Description

Hello,

art spends 4 CPU-minutes just to open a file. I observe it with any fcl and both on my laptop and on detsim. For example:

```
detsim $ /usr/bin/time mu2e -c Analyses/test/dumpDataProducts.fcl /mu2e/data/users/gandr/redmine/20140117a/detectorDetectorBeamFlash.root
```

....

Art has completed and will exit with status 0.

```
220.81user 4.98system 3:54.79elapsed 96%CPU (0avgtext+0avgdata 3365584maxresident)k
0inputs+0outputs (65major+210581minor)pagefaults 0swaps
```

A peek with gdb suggests it spends all the time dealing with ParameterSet-s. I suspect art does a lot of unnecessary moves in this case. Lazy evaluation would help.

Andrei

```
#0 0x00007f6d21a7eabf in polarssl::shal_process (ctx=ctx@entry=0x7fff8e9e5830,
data=data@entry=0xa4956f8 "\", \"keep mu2e::StatusG4_*_*_*\", \"keep *_g4run_*_*\", \"drop uintm
u2e::PhysicalVolumeInfovstd::pairs_g4run_*_*\", \"keep *_compressPVFull_*_*\"]}")
at /home/garren/scratch/p/art_suite/v1_08_09/source/cetlib-v1_03_25/polarssl/shal.cc:213
#1 0x00007f6d21a7f0bf in polarssl::shal_update (ctx=0x7fff8e9e5830, input=<optimized out>, ilen=<
optimized out>)
at /home/garren/scratch/p/art_suite/v1_08_09/source/cetlib-v1_03_25/polarssl/shal.cc:271
#2 0x00007f6d21eeb4c5 in operator<< (mesg=..., this=0x7fff8e9e5830) at /home/garren/scratch/p/cet
lib/v1_03_25/include/cetlib/shal.h:42
#3 shal (mesg=..., this=0x7fff8e9e5830) at /home/garren/scratch/p/cetlib/v1_03_25/include/cetlib/
shal.h:30
#4 fhicl::ParameterSetID::reset (this=this@entry=0x7fff8e9e5b10, ps=...) at /home/garren/scratch/
p/art_suite/v1_08_09/source/fhicl-cpp-v2_17_12/fhiclcpp/ParameterSetID.cc:66
#5 0x00007f6d21ee761e in fhicl::ParameterSet::id (this=this@entry=0x7fff8e9e5ae0) at /home/garren
/scratch/p/art_suite/v1_08_09/source/fhicl-cpp-v2_17_12/fhiclcpp/ParameterSet.cc:66
#6 0x00007f6d29fef5c3 in cet::registry_via_id<fhicl::ParameterSetID, fhicl::ParameterSet>::put (v
alue=...)
at /home/garren/scratch/p/cetlib/v1_03_25/include/cetlib/registry_via_id.h:94
#7 0x00007f6d21f01613 in encode (xval=...) at /home/garren/scratch/p/art_suite/v1_08_09/source/fh
icl-cpp-v2_17_12/fhiclcpp/make_ParameterSet.cc:60
#8 0x00007f6d21f0232e in fhicl::make_ParameterSet (tbl=..., ps=...) at /home/garren/scratch/p/art
_suite/v1_08_09/source/fhicl-cpp-v2_17_12/fhiclcpp/make_ParameterSet.cc:79
#9 0x00007f6d21f023c5 in fhicl::make_ParameterSet (str=..., ps=...) at /home/garren/scratch/p/art
_suite/v1_08_09/source/fhicl-cpp-v2_17_12/fhiclcpp/make_ParameterSet.cc:109
#10 0x00007f6d25ae35ed in art::RootInputFile::RootInputFile (this=0x2c2ee40, fileName=..., catalog
Name=..., processConfiguration=..., logicalFileName=..., filePtr=...,
origEventID=..., eventsToSkip=0, whichSubRunsToSkip=..., fcip=..., treeCacheSize=0, treeMaxVir
tualSize=-1, processingMode=art::InputSource::RunsSubRunsAndEvents,
forcedRunOffset=0, whichEventsToProcess=..., noEventSort=false, groupSelectorRules=..., dropMe
rgeable=false, duplicateChecker=..., dropDescendants=true)
at /home/garren/scratch/p/art_suite/v1_08_09/source/art-v1_08_09/art/Framework/IO/Root/RootInp
```

utFile.cc:192

```
#11 0x00007f6d25aeb3 in art::RootInputFileSequence::initFile (this=this@entry=0x2b398f0, skipBadFiles=<optimized out>)
```

```
    at /home/garren/scratch/p/art_suite/v1_08_09/source/art-v1_08_09/art/Framework/IO/Root/RootInputFileSequence.cc:238
```

Related issues:

Related to art - Bug #5805: Attacking parameter set bloat

Closed

04/01/2014

History

#1 - 02/10/2014 12:00 PM - Christopher Green

- Subject changed from *lazier file opening?* to *File opening slow due to parsing of saved configuration.*
- Category set to *Metadata*
- Status changed from *New* to *Accepted*
- Target version set to *1.09.00*
- Estimated time set to *120.00 h*
- SSI Package - *added*
- SSI Package *deleted ()*

As reflected by the change in subject, the slowness is likely to be the parsing of the saved configuration information. In order to speed this up, we would need to revise the way that parameter sets are saved, which is currently by string-ifying the entire expanded configuration of each parameter set. When they are read back in, each parameter set is parsed independently, leading to duplication of effort.

Given that checksums would change, this would present a backward-compatibility problem, also, which would need to be addressed.

#2 - 02/17/2014 12:27 PM - Christopher Green

- Target version changed from *1.09.00* to *521*

#3 - 03/12/2014 05:09 PM - Christopher Backhouse

I have a similar problem.

The file at `/nova/app/users/bckhouse/summed.root` takes forever to do anything with (20 minutes to run `config_dumper`).

This file contains about 10,000 subruns, but with very little information remaining. These files are naturally produced by our calibration procedure. You could also imagine, for example, collections of data events for scanning to be constructed in a similar way.

I've discussed this before, but it doesn't look like it ever made it into an issue.

Issue 4498 would mitigate my particular case, but there will still be significant overhead from the Runs remaining.

#4 - 03/12/2014 05:09 PM - Christopher Backhouse

In fact, the job that I'm trying to run doesn't have a `RootOutput` module. Is it possible to skip all this work in that case?

#5 - 09/02/2014 03:07 PM - Christopher Green

- Status changed from *Accepted* to *Resolved*
- Assignee set to *Christopher Green*

This issue was resolved with the fix for issue [#5805](#).

#6 - 09/02/2014 03:11 PM - Christopher Green

- Target version changed from *521* to *1.11.00*

#7 - 09/02/2014 03:11 PM - Christopher Green

- % Done changed from *0* to *100*

#8 - 09/03/2014 09:26 AM - Christopher Green

- Status changed from *Resolved* to *Closed*