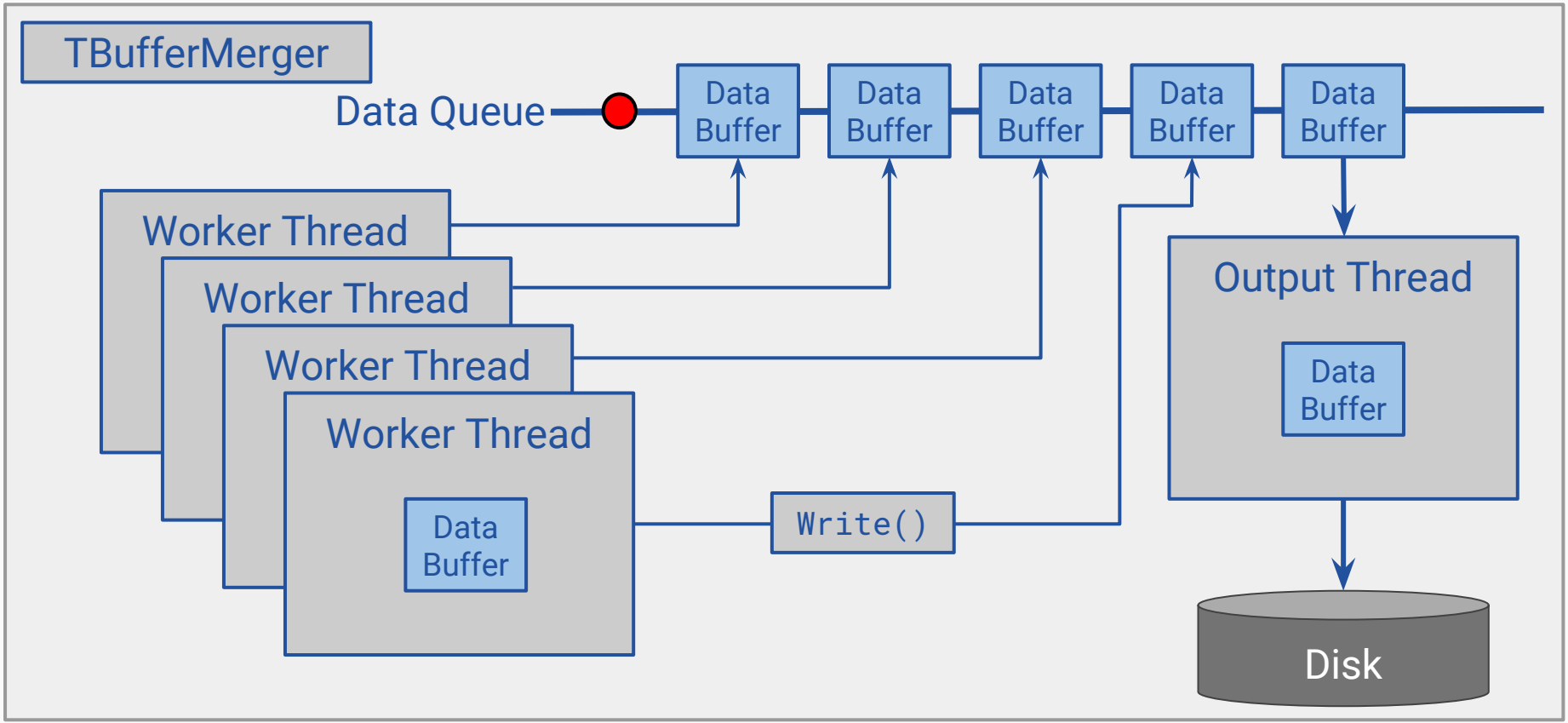

# Performance measurements of Parallel TTree writing in ROOT

Guilherme Amadio

# Performance Optimization of ROOT I/O

- Used VTune to analyse performance of parallel I/O
- Found bottleneck in TClass:LoadClassInfo()
- Changes under testing are in PR 666 on GitHub

# Data Flow with TBufferMerger



# Programming Model: Pythia event generation

```
#include <thread>
#include <vector>

#include "Pythia8/Pythia.h"
#include "ROOT/TBufferMerger.hxx"
#include "TROOT.h"
#include "TTree.h"

using namespace ROOT::Experimental;

int main() {

   size_t nWorkers = 4;
   size_t nEvents  = 1000;
   size_t nEventsPerWorker = nEvents/nWorkers;

   std::string filename("pythia-tbm.root");

   gROOT->SetBatch();
   ROOT::EnableThreadSafety();

   TBufferMerger merger(filename.c_str(), "recreate");
```

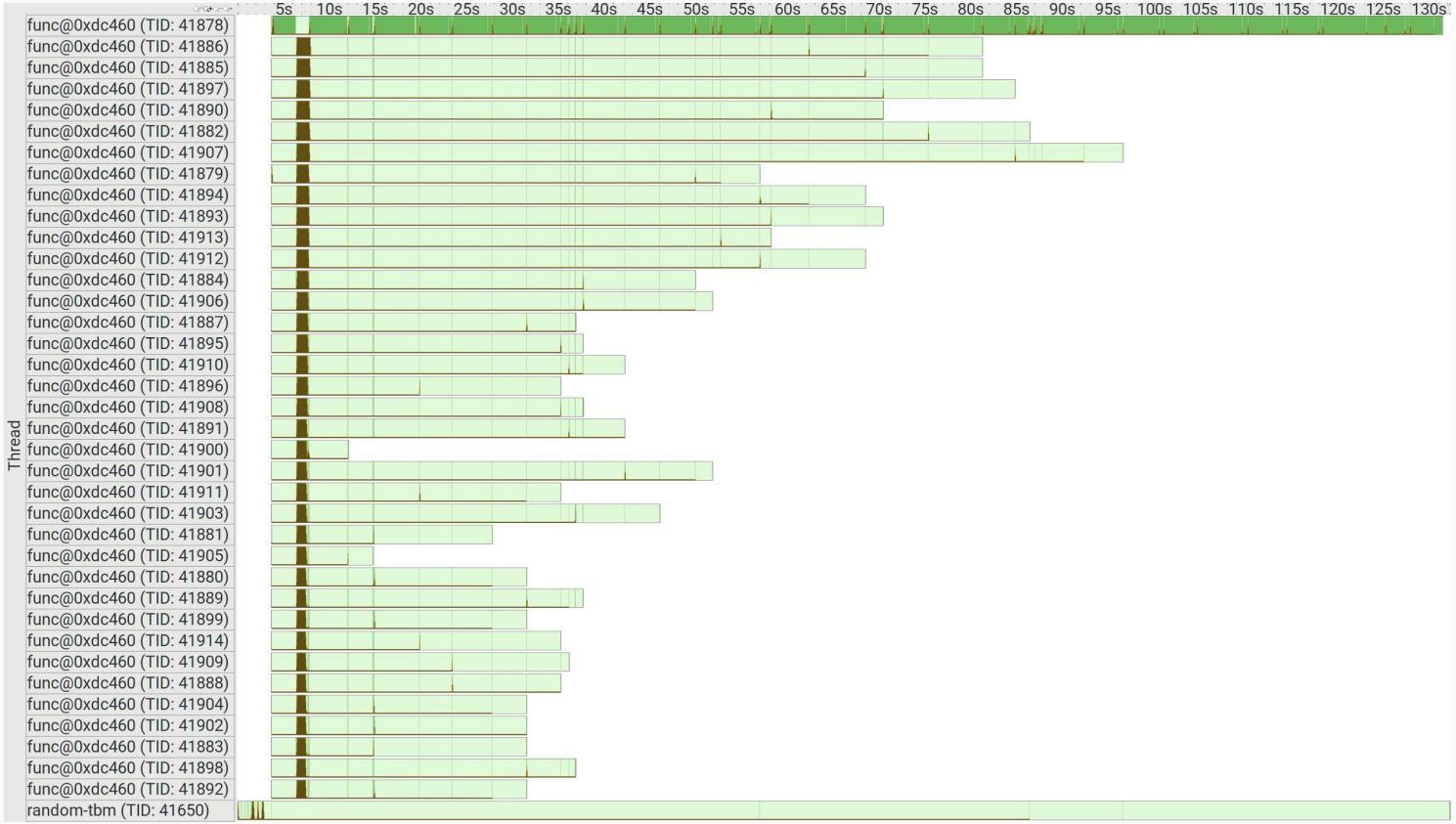
```
   auto pythia_gen = [=, &merger]() {
      Pythia8::Pythia pythia;
      pythia.readString("HardQCD:all = on");
      pythia.readString("PhaseSpace:pTHatMin = 20.");
      pythia.readString("Beams:eCM = 14000.");
      pythia.init();

      auto f = merger.GetFile();
      Pythia8::Event *e = &pythia.event;
      auto t = new TTree("pythia", "pythia");
      t->ResetBit(kMustCleanup);
      t->Branch("event", &e);

      for (size_t n = 0; n < nEventsPerWorker; ++n) {
         while (!pythia.next());
         t->Fill();
      }
      f->Write();
   };

   std::vector<std::thread> workers;
   for (size_t i = 0; i < nWorkers; ++i)
      workers.emplace_back(pythia_gen);

   for (auto&& worker : workers)  worker.join();

   return 0;
}
```

# TBufferMerger Benchmarks

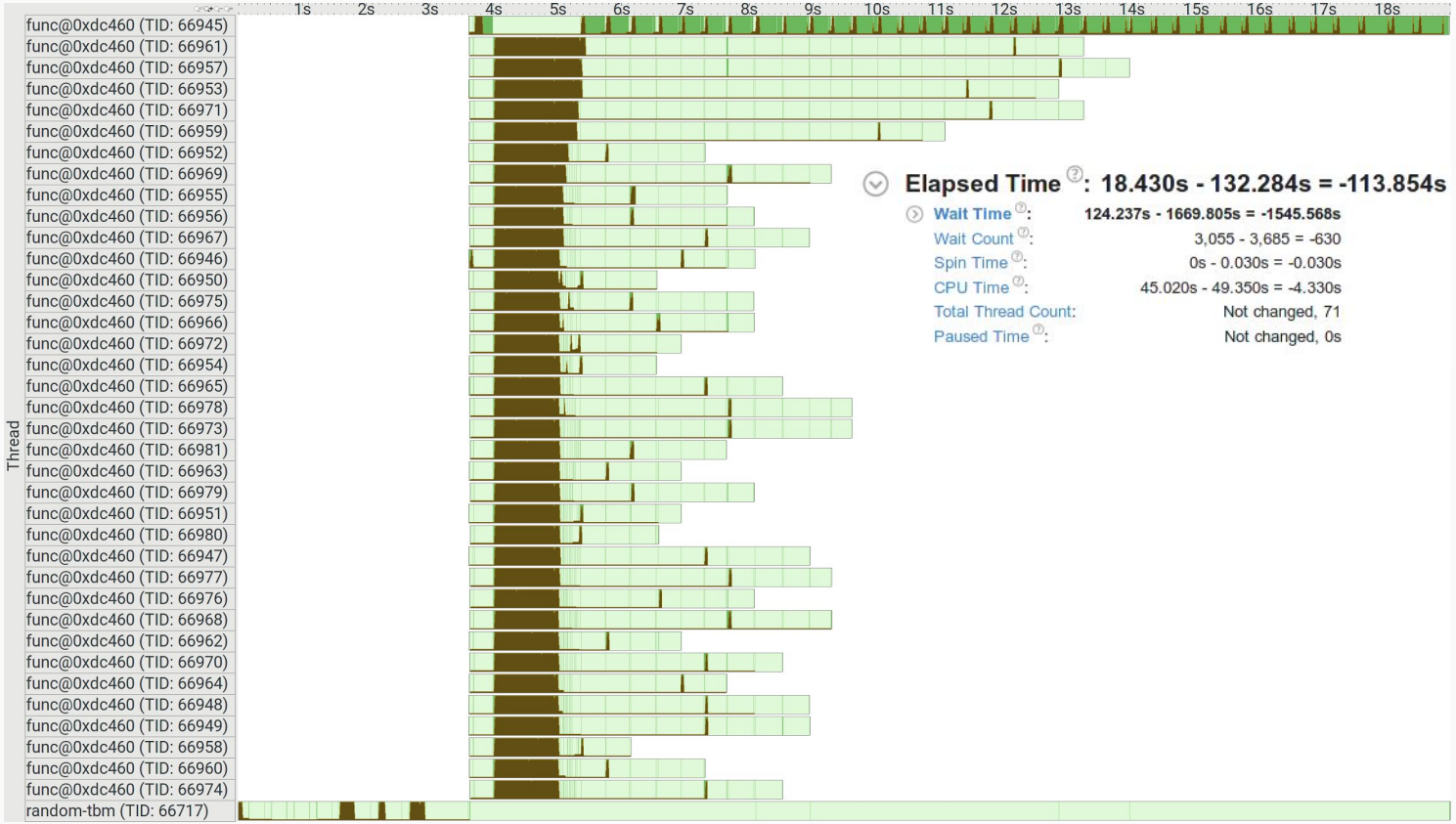
Hardware configuration:

- Partnership with São Paulo State University (UNESP)
- Server: 2x Intel Xeon E5-2699 v3 (2.30GHz, 128GB RAM)
- Intel Xeon Phi KNL (68 cores, 1.4GHz, 210GB RAM)

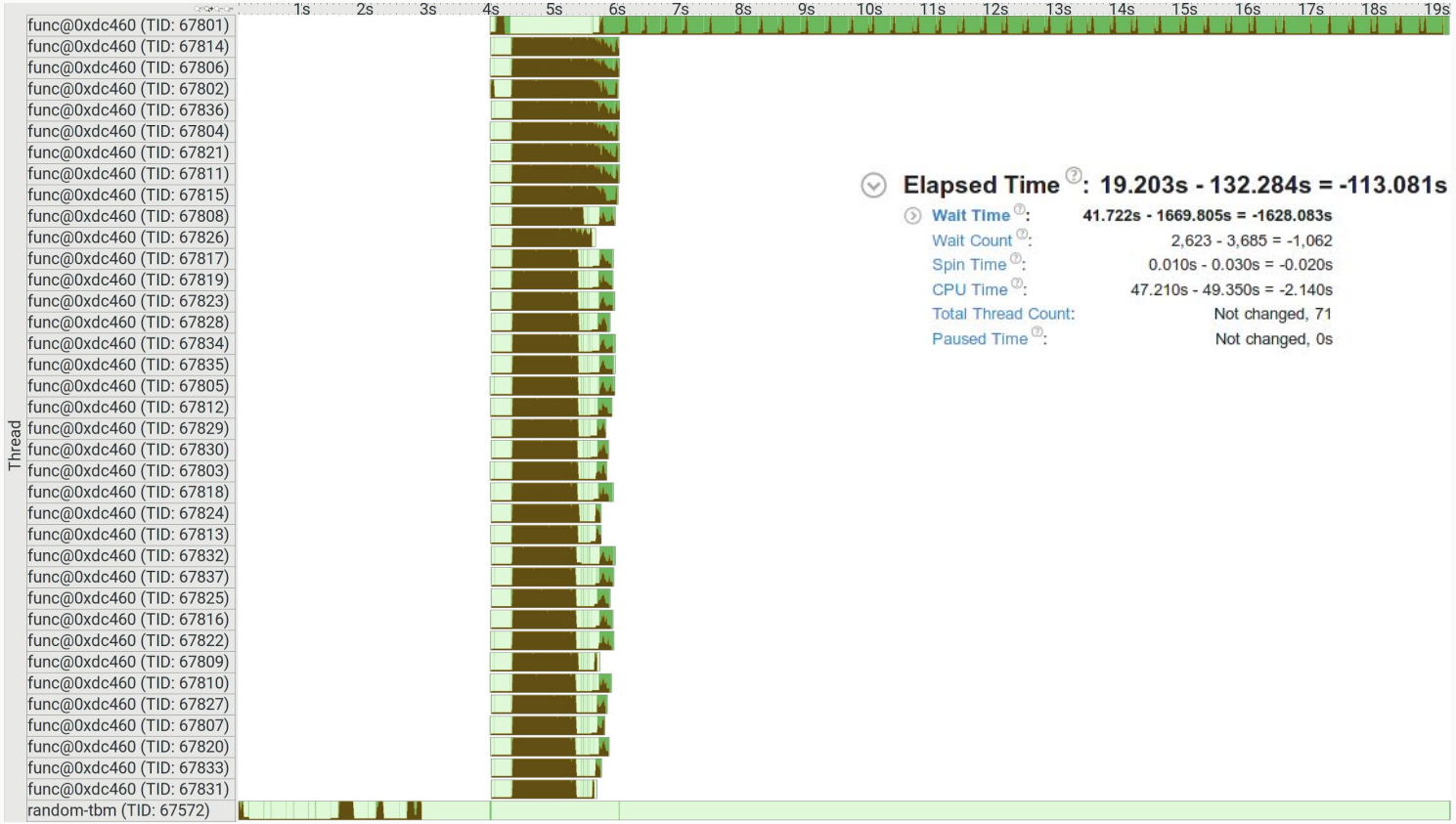
# VTune Analysis: Locks and Waits



# VTune Analysis: Locks and Waits (PR666)



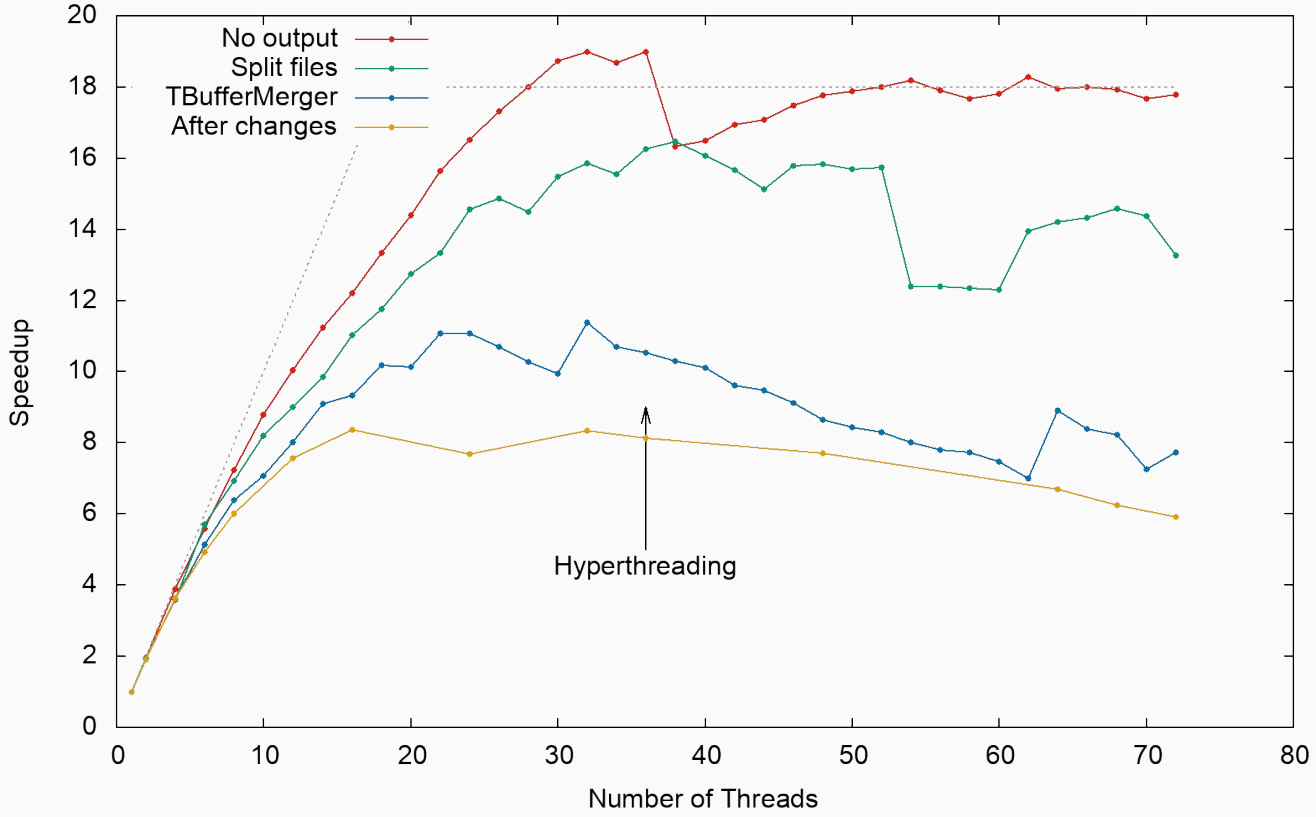
# VTune Analysis: Locks and Waits (Extras)



# VTune Analysis: Remaining Long Waits

| Sync Object / Function / Call Stack | Wait Time by Thread Concurrency ▼ | | | | | Wait Count | Spin Time | Module |
|---|---|---|---|---|---|---|---|---|
| | Idle | Poor | Ok | Ideal | Over | | | |
| Mutex 0xc97d2486 | 0.123s | 139.316s | 0.209s | 0.006s | 0s | 1,787 | 0s | |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.017s | 31.792s | 0s | 0s | 0s | 371 | 0s | libCling.so |
| TClingClassInfo::GetBaseOffset | 0.017s | 31.792s | 0s | 0s | 0s | 371 | 0s | libCling.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.030s | 31.091s | 0s | 0s | 0s | 521 | 0s | libRIO.so |
| TStreamerInfo::Streamer | 0.030s | 31.091s | 0s | 0s | 0s | 521 | 0s | libRIO.so |
| TClass::Streamer | 0.030s | 31.091s | 0s | 0s | 0s | 521 | 0s | libRIO.so |
| TBufferFile::WriteObjectClass | 0.030s | 31.091s | 0s | 0s | 0s | 521 | 0s | libRIO.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.014s | 11.911s | 0s | 0s | 0s | 145 | 0s | libCore.so |
| TDirectory::UnregisterContext | 0.014s | 11.911s | 0s | 0s | 0s | 145 | 0s | libCore.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.004s | 11.861s | 0s | 0s | 0s | 36 | 0s | libRIO.so |
| TMemFile::ResetAfterMerge | 0.004s | 11.861s | 0s | 0s | 0s | 36 | 0s | libRIO.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.018s | 10.311s | 0s | 0s | 0s | 35 | 0s | libRIO.so |
| TFile::~TFile | 0.018s | 10.311s | 0s | 0s | 0s | 35 | 0s | libRIO.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.001s | 8.080s | 0s | 0s | 0s | 274 | 0s | libCore.so |
| TClass::GetStreamerInfo | 0.001s | 8.080s | 0s | 0s | 0s | 274 | 0s | libCore.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.008s | 7.067s | 0.190s | 0.006s | 0s | 130 | 0s | libCore.so |
| TDirectory::RegisterContext | 0.008s | 7.067s | 0.190s | 0.006s | 0s | 130 | 0s | libCore.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.004s | 6.038s | 0.001s | 0s | 0s | 23 | 0s | libCore.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.013s | 5.732s | 0s | 0s | 0s | 29 | 0s | libRIO.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.001s | 4.160s | 0s | 0s | 0s | 32 | 0s | libRIO.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.007s | 3.506s | 0s | 0s | 0s | 29 | 0s | libRIO.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.002s | 3.082s | 0.009s | 0.000s | 0s | 30 | 0s | libCore.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.002s | 2.203s | 0s | 0s | 0s | 35 | 0s | libCore.so |
| _ZN10TLockGuardC4EP13TVirtualMutex | 0.000s | 1.806s | 0s | 0s | 0s | 35 | 0s | libRIO.so |

# Pythia Event Generation: Speedup



# Summary and Conclusion

- Good performance compared with writing to multiple files
- Fixed an important performance issue that affects ROOT I/O as a whole. More improvements to come
- Parallel snapshot action now available without changes in user code other than calling `ROOT::EnableImplicitMT()`
- However, some scaling issues remain for large numbers of worker threads, currently under investigation