

Google Summer Of Code 2022 1st Evaluation

Report:(CERN-HSF):RNTuple

By : Sayandeep Ghosh , Undergraduate Engineering Pre Final Year
Student(Electronics & Instrumentation)
Jadavpur University, Kolkata, West bengal, India
Mentors: Javier Lopez Gomez, Jacob Blomer

Software Environment Used: Ubuntu(WSL-Windows Linux Subsystem)

Task 1 : Building ROOT from Sources :

```
$ git clone --branch latest-stable https://github.com/root-project/root.git root_src
$ mkdir root_build root_install && cd root_build
$ cmake -DCMAKE_INSTALL_PREFIX=../root_install ../root_src # && check cmake configurat
$ cmake --build . -- install -j4 # if you have 4 cores available for compilation
```

I used these lines of code for building ROOT from sources.
After building here is my result:

```
sayang@DESKTOP-IFS80HM:~$ source root_install/bin/thisroot.sh
sayang@DESKTOP-IFS80HM:~$ root

-----
| Welcome to ROOT 6.27/01                               https://root.cern |
| (c) 1995-2021, The ROOT Team; conception: R. Brun, F. Rademakers |
| Built for linuxx86_64gcc on Mar 14 2022, 18:22:00 |
| From heads/master@v6-25-02-721-gb92cbb9ea9 |
| With c++ (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0 |
| Try '.help', '.demo', '.license', '.credits', '.quit'/'.q' |
-----

root [0]
```

I also familiarized myself with ROOT with the help of:
<https://root.cern/primer/>

I also checked whether RNTuple was build or not.

```
sayang@DESKTOP-IFS80HM:~$ root-config --features
cxx17 asimage builtin_afterimage builtin_clang builtin_cling builtin_llvm builtin_lz4 builtin_lzma builtin_nlohmannjson
builtin_openui5 builtin_pcre builtin_xxhash builtin_zstd dataframe exceptions gdm1 http mlp minuit2 pyroot roofit root7
rpath runtime_cxxmodules shared ssl tmva tmva-pymva spectrum x11 xml
sayang@DESKTOP-IFS80HM:~$ root-config --has-root7
yes
sayang@DESKTOP-IFS80HM:~$
```

Task 2: RNTuple Specific Task :

For the input file I took 3 csv files which I saved in the directory mentioned below:

root_src/tutorials/tree/

The files can be found on my github repository :

(https://github.com/sayang14/GSOC_2022_Evaluation_RNTuple)

- 1) test_sample100.csv
- 2) test_sample84006.csv
- 3) input_file.csv(given by the respected mentors)

I mainly used 3 databases but any database(.csv) will work provided that the user will have to mention the name(filename.csv after saving it in the tutorials/tree directory of the installation path) within the command line. I created the following RNTuple ROOT File: gsoc_eval_RNTuple.C(Source Code below which I tried to make it a little self explanatory):

```
R_LOAD_LIBRARY(ROOTNTuple)
#include <ROOT/RNTuple.hxx>
#include <ROOT/RNTupleModel.hxx>

#include <TCanvas.h>
#include <TH1I.h>
#include <TROOT.h>
#include <TString.h>

#include <cstring>
#include <cassert>
#include <cstdio>
#include <fstream>
#include <iostream>
#include <memory>
#include <string>
#include <sstream>
#include <utility>

using RNTupleModel = ROOT::Experimental::RNTupleModel;
using RNTupleReader = ROOT::Experimental::RNTupleReader;
using RNTupleWriter = ROOT::Experimental::RNTupleWriter;

constexpr char const* kNTupleFileName = "gsoc_eval_RNTuple.root";

std::vector<vector<string>> h;
std::pair<std::shared_ptr<int>,std::shared_ptr<float>> p;
std::map<int,std::shared_ptr<int>> mp1;
std::map<int,std::shared_ptr<float>> mp2;

void Ingest() {
    char filename[100];
    //I specifically mentioned the tutorials directory since I found most of the datasets are stored there
    std::cout << "Enter the csv filename path(no more than 100 characters) present in the installation/tutorials directory of
ROOT):" << std::endl;
    std::cin >> filename;
    ifstream fin(gROOT->GetTutorialDir() + "/tree/" + filename);
```

```

assert(fin.is_open());
//creating unique pointer to an empty data model
auto model = RNTupleModel::Create();

std::string record_header, word;
getline(fin, record_header);
std::cout<<record_header<<std::endl;
std::istringstream iss(record_header);
int i = 0;
while(getline(iss, word, ','))
{
    if(word.find("int")!= string::npos){
        mp1[i] = model->MakeField<int>(word);
        //mp1[i] = fld;
        mp2[i] = nullptr;
        i++;
    }
    else if(word.find("float")!= string::npos) {
        mp2[i] = model->MakeField<float>(word);
        //mp2[i] = fld;
        mp1[i] = nullptr;
        i++;
    } else continue;
}

// We hand-over the data model to a newly created ntuple of name "new_ntuple", stored in kNTupleFileName
auto ntuple = RNTupleWriter::Recreate(std::move(model), "new_ntuple", kNTupleFileName);

std::string record;
//std::cout<<v1.size()<< " "<<v2.size()<<std::endl;
while(std::getline(fin, record)) {
    std::istringstream iss(record);
    //iss>>*fld1>>*fld2>>*fld3>>*fld4;
    for(int j=0; j<i; j++)
    {
        if(mp1[j]==nullptr)iss>>*mp2[j];
        else iss>>*mp1[j];
        if(iss.peek()=='\n')iss.ignore();
    }
    ntuple->Fill();
}

void Analyze() {
    // Get a unique pointer to empty RNTuple models
    auto model = RNTupleModel::Create();

    std::string entry;
    std::cout<<"Enter the entry whose distribution you wish to see in the following way(Name:Type)(Avoid any spaces)!"<<std::endl;
    std::cin>>entry;

    //defining field(fld from Ingest function where it was declared) that is needed for reading
    if(entry.find("int")!=string::npos){auto fld = model->MakeField<int>(entry);
        p.first = fld;
    }
    else if(entry.find("float")!=string::npos){auto fld = model->MakeField<float>(entry);
        p.second = fld;
    }

    // Quick overview of the ntuple and list of fields.
    auto ntuple = RNTupleReader::Open(std::move(model), "new_ntuple", kNTupleFileName);
    ntuple->PrintInfo();

    std::cout << "The first entry in JSON format:" << std::endl;
    ntuple->Show(0);

    auto c = new TCanvas("c", "", 200, 10, 700, 500);
    TH1 h("h", " Distribution for your entry", 100, -100, 100);
    h.SetFillColor(40);

    for (auto entryId : *ntuple) {
        ntuple->LoadEntry(entryId);
        if(entry.find("int")!=string::npos)h.Fill(*p.first);
        else if(entry.find("float")!=string::npos)h.Fill(*p.second);
    }
}

```

```

    }

    h.DrawCopy();
}

void gsoc_eval_RNTuple() {
    Ingest();
    Analyze();
}

```

After Running the file, the user will have to enter the csv file path as well as the parameter(Name:type) whose distribution he/she wishes to see in a specific way. For simplicity as mentioned types should be int or float.

It shows the following :

Result 1: input_file.csv

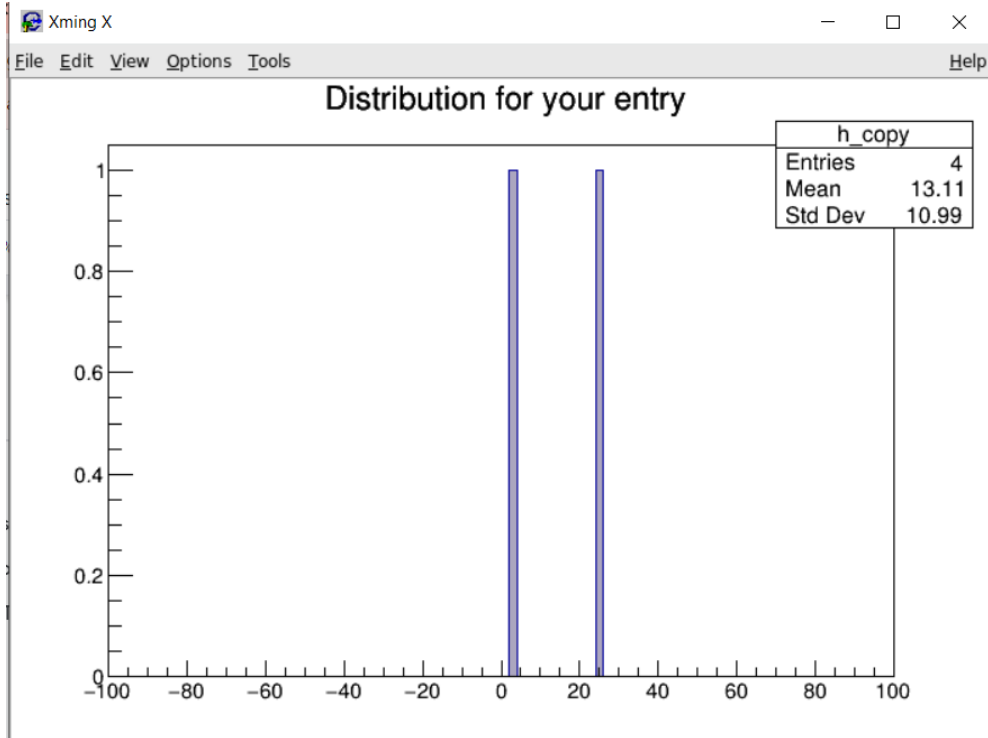
```

sayang@DESKTOP-IFS80HM:~/root_src/tutorials/v7/ntuple$ root gsoc_eval_RNTuple.C
-----
| Welcome to ROOT 6.27/01                               https://root.cern |
| (c) 1995-2021, The ROOT Team; conception: R. Brun, F. Rademakers |
| Built for linuxx86_64gcc on Mar 18 2022, 19:30:00 |
| From heads/master@v6-25-02-721-gb92cbb9ea9 |
| With c++ (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0 |
| Try '.help', '.demo', '.license', '.credits', '.quit'/'.' |
|-----|

root [0]
Processing gsoc_eval_RNTuple.C...
Enter the csv filename path(no more than 100 characters) present in the installation/tutorials directory of ROOT):
input_file.csv
An_Integer:int,pX:float,pY:float,pZ:float,Integer_2:int,ProbK:float

Enter the entry whose distribution you wish to see in the following way(Name:Type)(Avoid any spaces)!
pX:float
Warning in <[ROOT.NTuple] Warning /home/sayang/root_src2/tree/ntuple/v7/src/RNTupleSerialize.cxx:1113 in static ROOT::Experimental::RResult<void> ROOT::Experimental::Internal::RNTupleSerializer::DeserializeHeaderV1(const void*, uint32_t, ROOT::Experimental::RNTupleDescriptorBuilder&)>: Pre-release format version: RC 1
***** NTUPLE *****
* N-Tuple : new_ntuple *
* Entries : 4 *
*****
* Field 1 : An_Integer:int (std::int32_t) *
* Field 2 : pX:float (float) *
* Field 3 : pY:float (float) *
* Field 4 : pZ:float (float) *
* Field 5 : Integer_2:int (std::int32_t) *
* Field 6 : ProbK:float (float) *
*****
The first entry in JSON format:
{
  "pX:float": 2.123
}
root [1] .q

```



Result 2: test_sample100.csv

```
sayang@DESKTOP-IFS80HM:~/root_src/tutorials/v7/ntuple$ root gsoc_eval_RNTuple.C
```

```
-----
| Welcome to ROOT 6.27/01                               https://root.cern |
| (c) 1995-2021, The ROOT Team; conception: R. Brun, F. Rademakers         |
| Built for linuxx86_64gcc on Mar 18 2022, 19:30:00                       |
| From heads/master@v6-25-02-721-gb92cbb9ea9                             |
| With c++ (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0                         |
| Try '.help', '.demo', '.license', '.credits', '.quit'/'.'q'           |
|-----
```

```
root [0]
Processing gsoc_eval_RNTuple.C...
Enter the csv filename path(no more than 100 characters) present in the installation/tutorials directory of ROOT):
test_sample100.csv
Emp_ID:int,Age1:float,Weight:int,Year:int,Month:int,Day:int,Age2:float,Salary:int
```

```
Enter the entry whose distribution you wish to see in the following way(Name:Type)(Avoid any spaces)!
```

```
Age1:float
```

```
Warning in <[ROOT.NTuple] Warning /home/sayang/root_src2/tree/ntuple/v7/src/RNTupleSerialize.cxx:1113 in static ROOT::Experimental::RResult<void> ROOT::Experimental::Internal::RNTupleSerializer::DeserializeHeaderV1(const void*, uint32_t, ROOT::Experimental::RNTupleDescriptorBuilder&)>: Pre-release format version: RC 1
```

```
***** NTUPLE *****
* N-Tuple : new_ntuple *
* Entries : 100 *
*****
* Field 1 : Emp_ID:int (std::int32_t) *
* Field 2 : Age1:float (float) *
* Field 3 : Weight:int (std::int32_t) *
* Field 4 : Year:int (std::int32_t) *
* Field 5 : Month:int (std::int32_t) *
* Field 6 : Day:int (std::int32_t) *
* Field 7 : Age2:float (float) *
* Field 8 : Salary:int (std::int32_t) *
```

```
The first entry in JSON format:
```

```
{
  "Age1:float": 36.36
}
```



Result 3: test_sample84006.csv

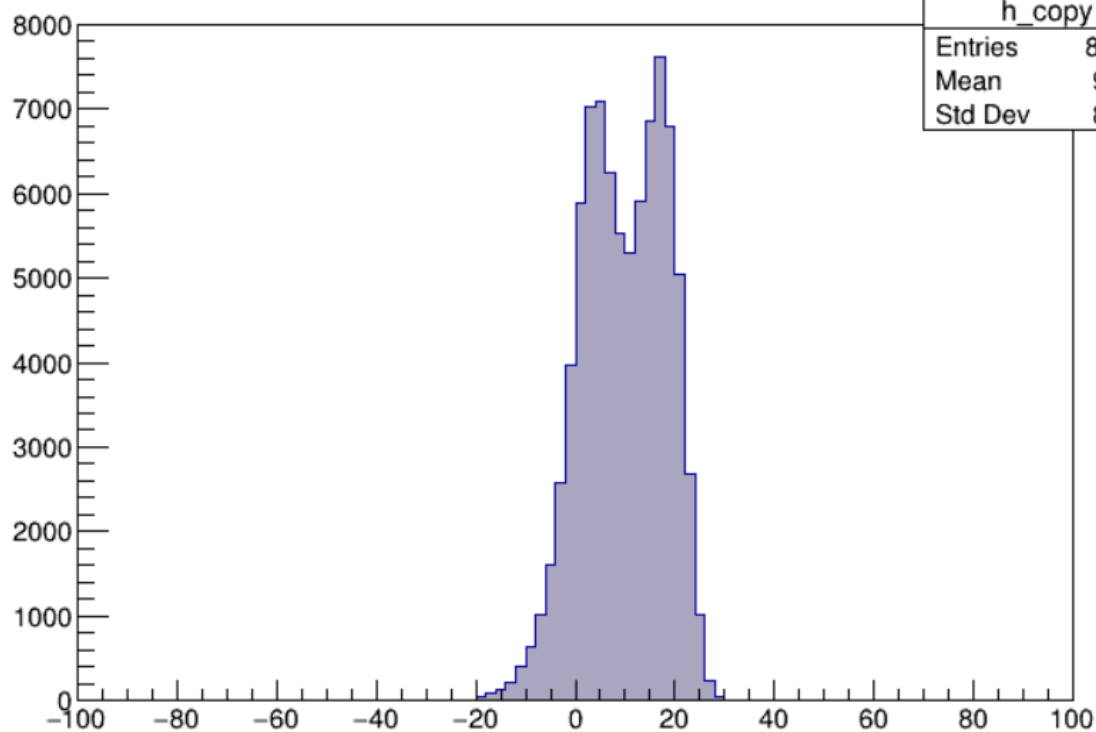
```
sayang@DESKTOP-IFS80HM:~/root_src/tutorials/v7/ntuple$ root gsoc_eval_RNTuple.C

-----
| Welcome to ROOT 6.27/01                               https://root.cern |
| (c) 1995-2021, The ROOT Team; conception: R. Brun, F. Rademakers |
| Built for linuxx8664gcc on Mar 18 2022, 19:30:00 |
| From heads/master@v6-25-02-721-gb92cbb9ea9 |
| With c++ (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0 |
| Try '.help', '.demo', '.license', '.credits', '.quit'/'.'q' |
-----

root [0]
Processing gsoc_eval_RNTuple.C...
Enter the csv filename path(no more than 100 characters) present in the installation/tutorials directory of ROOT):
test_sample84006.csv
Year:int,Month:int,Day:int,Temperature:float

Enter the entry whose distribution you wish to see in the following way(Name:Type)(Avoid any spaces)!
Temperature:float
Warning in <[ROOT.NTuple] Warning /home/sayang/root_src2/tree/ntuple/v7/src/RNTupleSerialize.cxx:1113 in static ROOT::Experimental::RResult<void> ROOT::Experimental::Internal::RNTupleSerializer::DeserializeHeaderV1(const void*, uint32_t, ROOT::Experimental::RNTupleDescriptorBuilder&): Pre-release format version: RC 1
***** NTUPLE *****
* N-Tuple : new_ntuple *
* Entries : 84006 *
*****
* Field 1 : Year:int (std::int32_t) *
* Field 2 : Month:int (std::int32_t) *
* Field 3 : Day:int (std::int32_t) *
* Field 4 : Temperature:float (float) *
*****
The first entry in JSON format:
{
  "Temperature:float": -7.4
}
```

Distribution for your entry



h_copy	
Entries	84006
Mean	9.585
Std Dev	8.345