

Application of PROOF system to optimization processes of physical analysis for the MPD experiment

Kacper Skelnik

Outline

1. What Is PROOF?
2. How to write Selectors
3. How we run physical analysis
4. Results
5. Plans for the future
6. Summary

What Is PROOF?

“PROOF, is an extension of ROOT enabling interactive analysis of large sets of ROOT files in parallel on clusters of computers or many-core machines.” (<https://root.cern.ch/proof>)

Selector

To analyze data using PROOF, user has to write macro called selector. It contains files:

- Selector.C
- Selector.h

Selector.h

```
// MpddstSelector.h  
  
class MpddstSelector : public TSelector{  
    /* Place to define list of TTree branches,  
    and functions */  
}
```

TSelector functions

- Begin() - executed on master at the beginning
- SlaveBegin() - executed on workers at the beginning
- Init() - executed on worker when getting new Tree
- Process() - executed on worker for every entry
- SlaveTerminate() - executed on worker at the end
- Terminate() - executed on master at the end

Selector.h

```
void MpddstSelector::Init(TTree *tree) {  
    ...  
}  
  
Bool_t MpddstSelector::Notify() {  
    return kTRUE;  
}
```

Selector.C

```
void MpddstSelector::Begin(TTree *) {}
```

```
void MpddstSelector::SlaveBegin(TTree *) {}
```

```
Bool_t MpddstSelector::Process(Long64_t entry) {}
```

```
void MpddstSelector::Terminate() {}
```


How we run physical analysis

```
root [ 0 ] TProof : : Open ( " " )
```

```
root [ 1 ] TChain* myChain = new TChain ( " cbmsim " )
```

```
root [ 2 ] myChain->AddFile ( " file.root " )
```

..... you can add more file to myChain

```
root [ 3 ] myChain->SetProof()
```

```
root [ 4 ] myChain->Process("MpddstSelector.C")
```

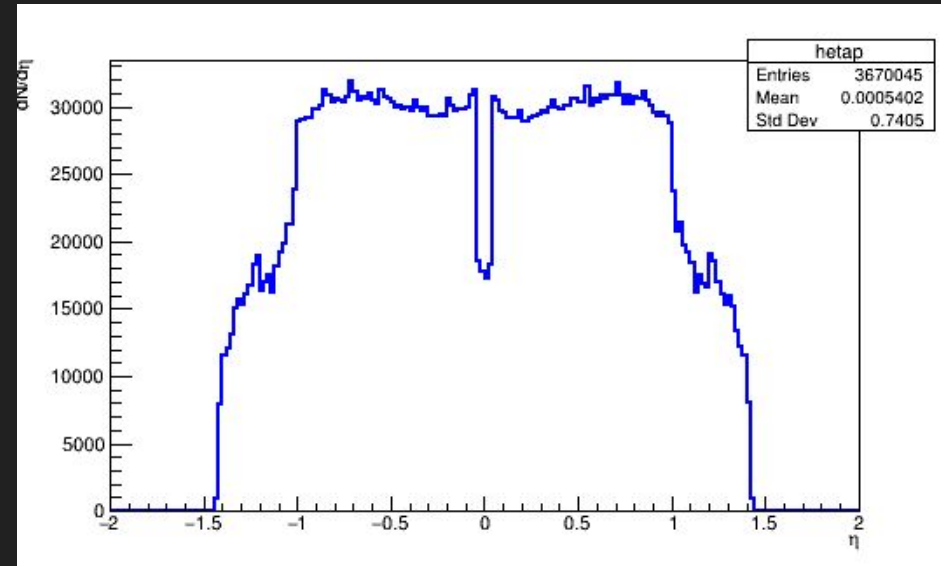
But that's a lot of code to write

So I needed to:

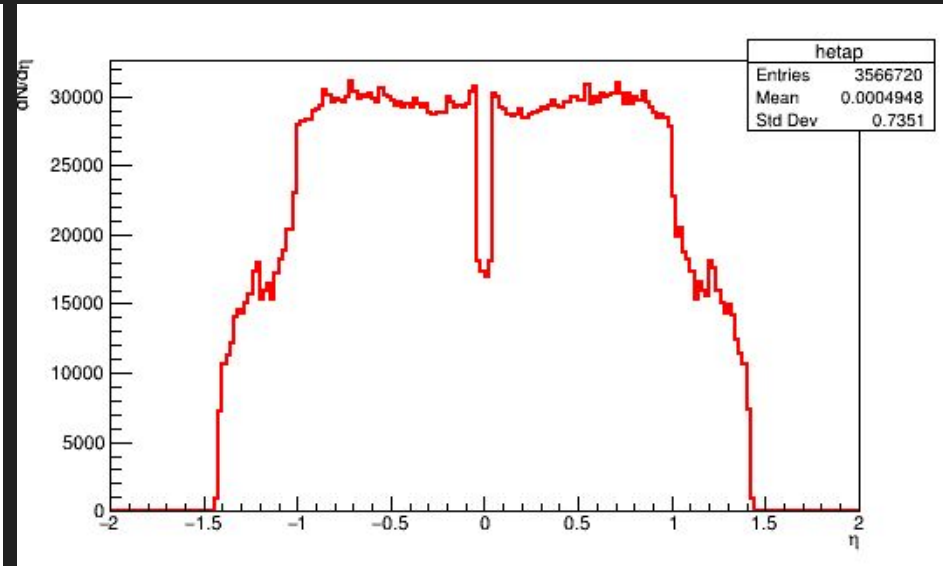
- write macro which can read any quantity of data, and load it into TChain object
- write script which starts ROOT, run PROOF session implement variable, run macros, and so on....

Now it is in just one command (`./runAnalyze.sh`).

Results - data quality

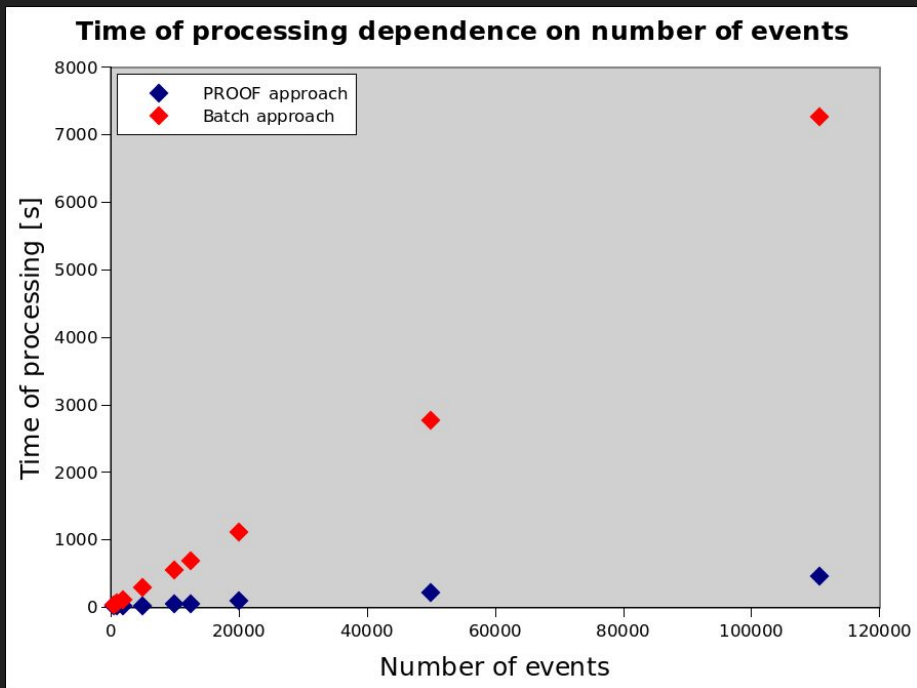


Analysis result
using PROOF

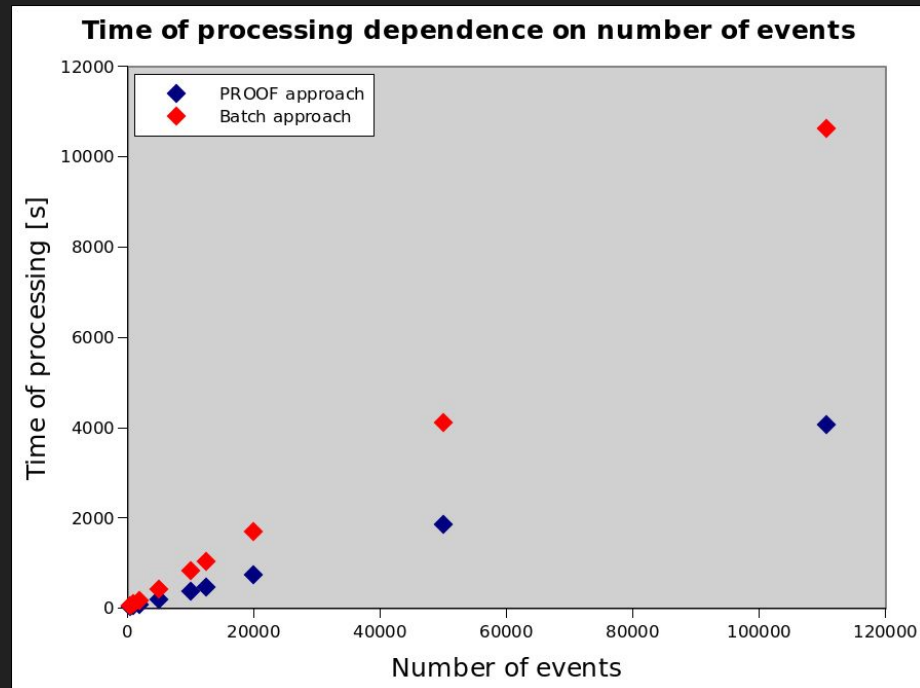


Analysis result
using classic macro

Results

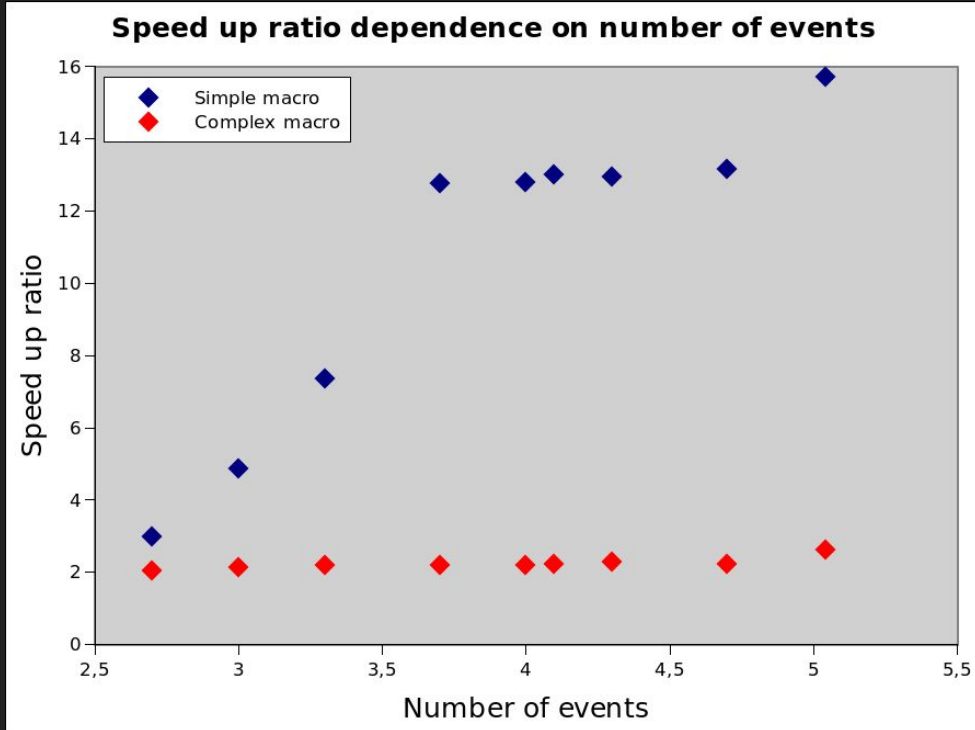


simple macro



complex macro

Results



Speed up ratio = time of
processing in classic
branch approach / time
of processing using
PROOF

Results

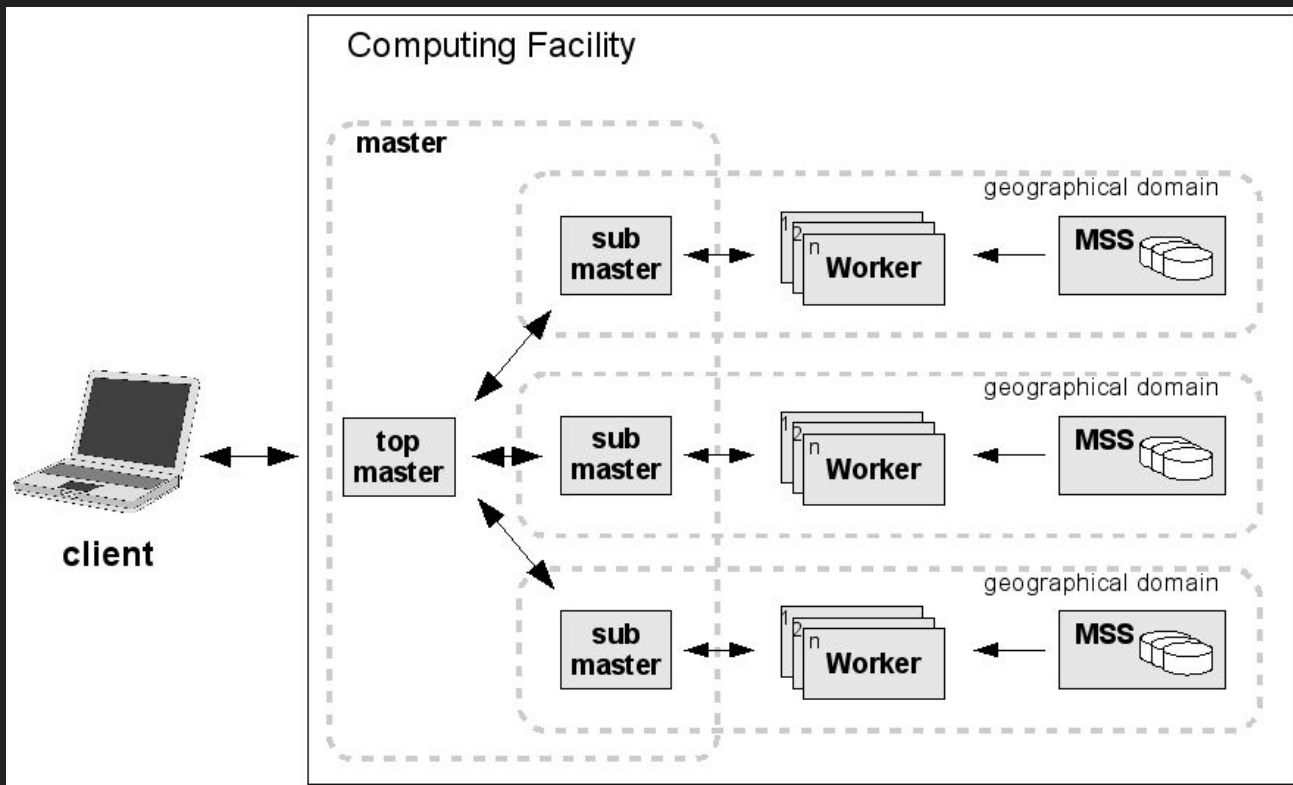


Efficiency improvement is clearly dependence on number of workers, but it seem to be limited

Plans for the future

- Developing system adjusting number of workers
- Creating Multi-Tier Master-Worker Architecture

Multi-Tier Master-Worker Architecture



source: (<https://root.cern.ch/multi-tier-master-worker-architecture>)

Summary

- Processing of physical data is now much faster (from 2 to almost 16 times faster).
- It can be even faster.
- It can take less computer resources.
- It is still a lot of to do.

Thank you for your
attention!