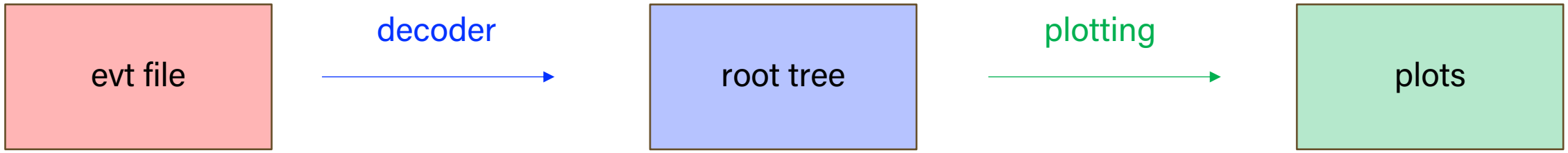
PARALLELIZATION OF RAW DATA DECODING

Milan Stojanovic

07/06/2023

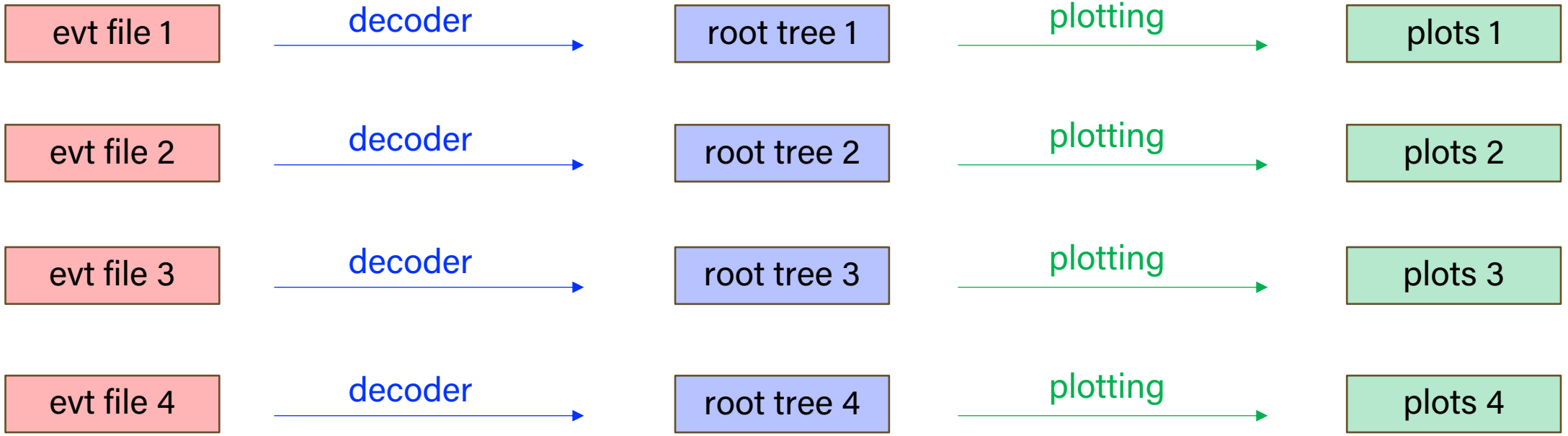
PURDUE UNIVERSITY

Overview



Current workflow:
Decoder: /sphenix/tg/tg01/commissioning/INTT/work/jbertaux_sPHENIX_home/*
Plots: /sphenix/tg/tg01/commissioning/INTT/work/misaki/FelixQuickViewer_1Felix.C

Overview



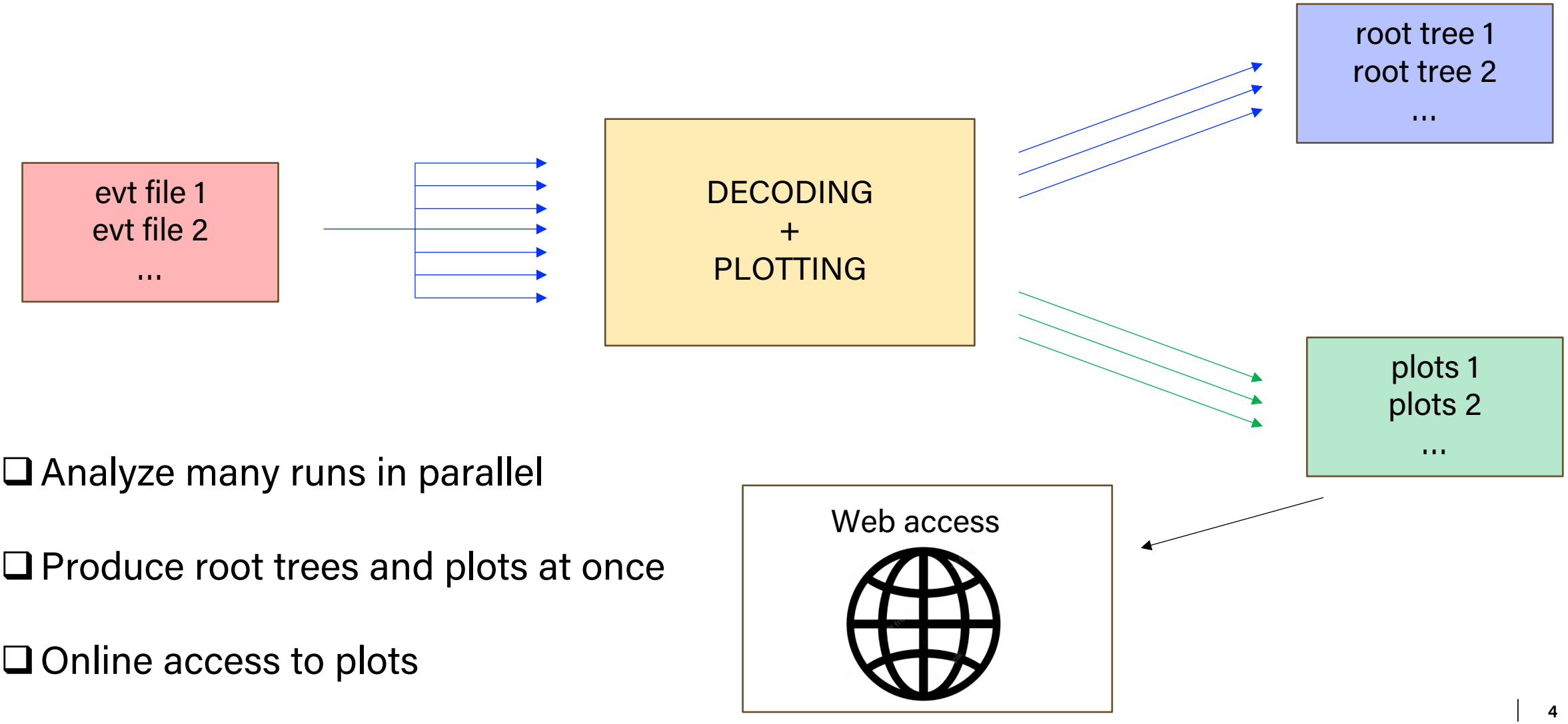
Current workflow:

Decoder: /sphenix/tg/tg01/commissioning/INTT/work/jbertaux_sPHENIX_home/*

Plots: /sphenix/tg/tg01/commissioning/INTT/work/misaki/FelixQuickViewer_1Felix.C

NEEDS TO BE DONE FOR EACH FILE SEPARATELY!

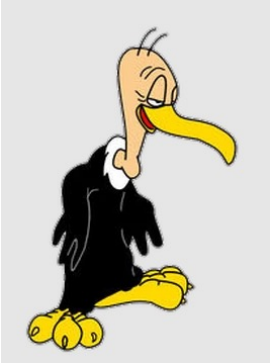
Parallelization



- ❑ Analyze many runs in parallel
- ❑ Produce root trees and plots at once
- ❑ Online access to plots

Parallelization

Runing many files in parallel → Condor jobs



- Scripts:
  `/sphenix/u/mstojano/rawdecoder/condorjobsintt/`

- Step1: Copy everything in your local folder and compile:

```
cp -r /sphenix/u/mstojano/rawdecoder/condorjobsintt /local/path/
cd /local/path/condorjobsintt/
make clean
make
```

- Step2: create list of input (*.evt) files:
  `infiles.list`

```
/sphenix/lustre01/sphnxpro/commissioning/INTT/beam/beam_intt0-00000001-0000.evt
/sphenix/lustre01/sphnxpro/commissioning/INTT/beam/beam_intt0-00007420-0000.evt
/sphenix/lustre01/sphnxpro/commissioning/INTT/beam/beam_intt0-00007421-0000.evt
/sphenix/lustre01/sphnxpro/commissioning/INTT/beam/beam_intt0-00007422-0000.evt
/sphenix/lustre01/sphnxpro/commissioning/INTT/beam/beam_intt0-00007423-0000.evt
```

Parallelization

❑ Step 3: submit jobs:

```
bash submitjobs.sh
```

❖ The script `submitjobs.sh` submits one job for one file
❖ Converts evt to root file
❖ Produces adc & entry vs channel distributions plots
❖ The both root and plots are stored in the:
`/sphenix/tg/tg01/commissioning/INTT/root_files/` (~1500 jobs done so far)

❖ The condor submission script :
`start.sub`

- ➢ "longlunch" = 2 hrs (max runtime)
- ➢ It can be increased if needed
- ➢ Err, out, and log files are all being stored in local submission folder

```
executable     = executable.sh
arguments      = $(Process)
error          = logfiles/err.$(Process)
output         = logfiles/out.$(Process)
log            = logfiles/foo_$(Process).log
+JobFlavour    = "longlunch"
```

Parallelization

- To add a custom script to analyze root file add it in the `macros.list`
- Currently `macros.list` has one macro:

```
FelixQuickViewer_1Felix_v2.C
~
~
```

- New macros need to satisfy two conditions:

1) Needs to take root file as input argument
2) Needs to specify output name (and path)

  - Output file of each job should have new name, otherwise output will be overwritten by other jobs.

Example:	
Input file:	beam_intt0-00007437-0000.root
Output file:	beam_intt0-00007437-0000_adc.jpg

Available function to do that in: `plotname.h`

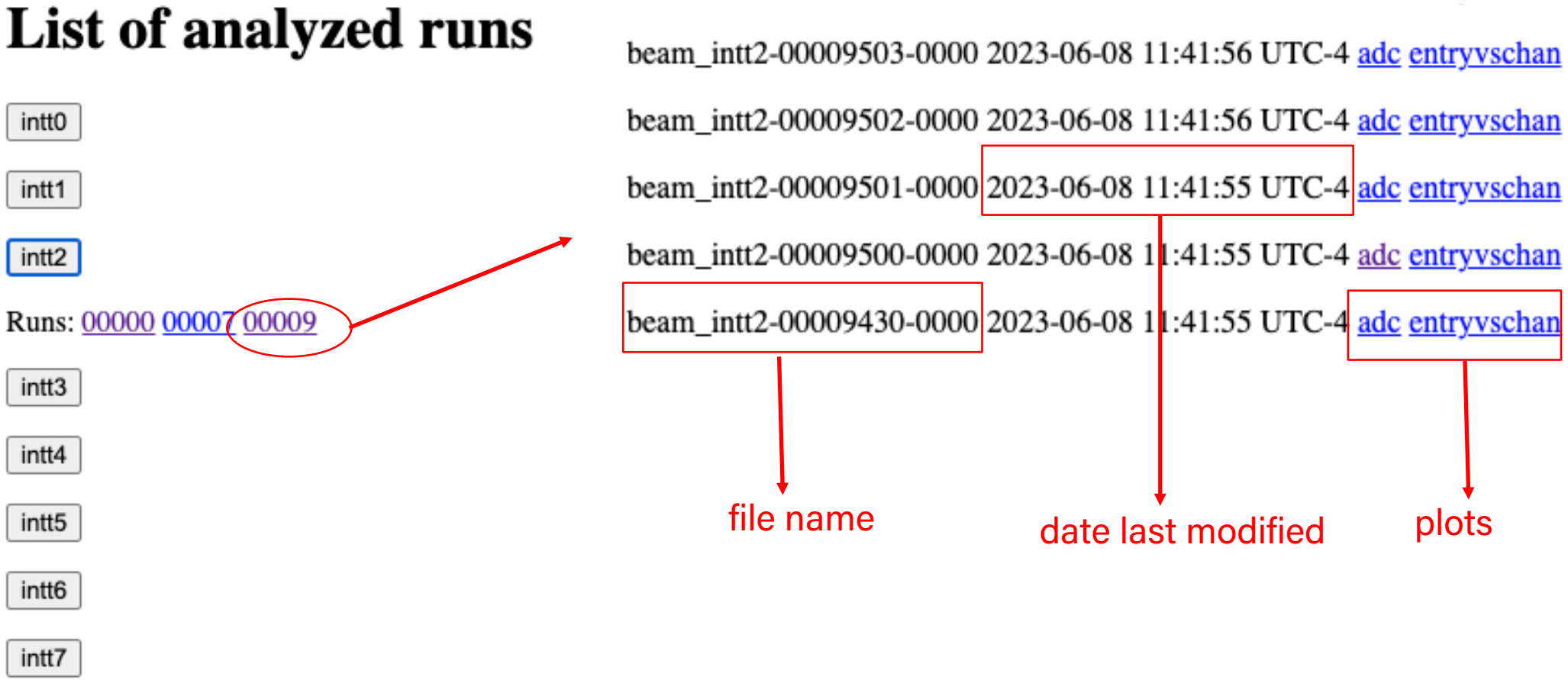
```
#include "plotname.h"
```

```
std::string inputname = "root_files/beam_intt0-00007437-0000.root";
142ms/step - loss: 0.5624 - auc: 0.9576
std::string outname = plotname(inputname, "adc.jpg");
```


outname = “/sphenix/tg/tg01/commissioning/INTT/root_files/beam_intt0-00007437-0000_adc.jpg”

Online access

- The web page: https://sphenix-intra.sdcc.bnl.gov/WWW/subsystem/intt/



To do

- ❑ Polishing the codes
- ❑ Macros name conventions
- ❑ Classification of the files?
- ❑ More/less automation?

Feel free to use and give feedback!