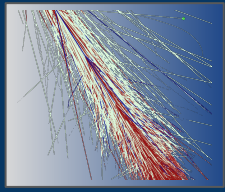
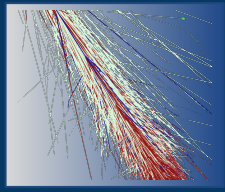


CORSIKA Output Format

Ralf Ulrich, 30.9.2020

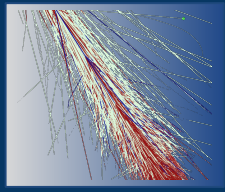
Note



playground branch: **output_format_testing**

- Added: **ASCII, ROOT, parquet, HDF5** output writers based on **ObservationPlane**
- removed all std::cout and std::cerr
- extended **vertical_EAS** to generate multiple showers

Overview

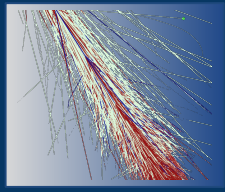


1. **HPC friendly**
2. **low maintenance**
3. **Cleanliness and simplicity of format/solution (requirement: multi-file solution to easily split of ground particles etc.)**
5. **Disk space consumption (aka sharing data)**
4. **Read performance (aka analyzing data)**
6. **Write performance (aka producing data)**

Testing setup on branch “output_format_testing”:

- single p shower at 1e19eV with seed 12345 (./vertical_EAS 1 1 1e10 1 1 12345)
- 1e5 p showers at 1e13eV with seed 12345 (./vertical_EAS 1 1 1e4 1e5 1 12345)

Basic results



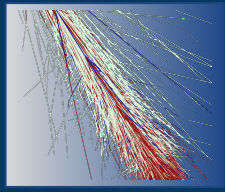
1e19eV p shower:

- 1 shower
- 1’244’553 particles
- About 2 h
  - root: 116min
  - Parquet: 123min
- Size:
  - Hdf5 24M
  - Parquet-plain 49M+0M
  - Parquet-gzip 53M+0M
  - ROOT 23M

1e13eV p showers:

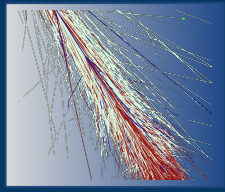
- 1e5 showers
- 708’474 particles
- About 2 h
  - root: 118min
  - Parquet: 121min
- Size
  - Hdf5 356M
  - Parquet-plain 24M+3.2M
  - Parquet-gzip 25M+3.2M
  - ROOT 65M

Analysis, parquet



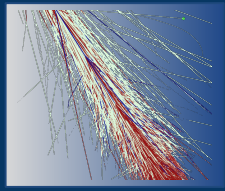
```
%matplotlib inline
import pyarrow.parquet as pq
import matplotlib.pyplot as plt
import timeit
import math
import numpy as np

name='1e19/particles.plain-parquet'
f = pq.ParquetFile(name)

t=f.read(columns=["PID","energy","distance"], use_threads=1)
def to_list():
    return t.column(1)

%timeit to_list()
e_data = to_list()

plt.figure()
plt.yscale('log')
#%timeit plt.hist(list(math.log10(e.as_py()) for e in t.column(1)), bins=100, range=(9,14))
%timeit plt.hist(np.log10(np.asarray(e_data)), bins=100, range=(9,14))
plt.show()
```

Analysis, hdf5



```
import h5py
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import timeit
import math

name = '1e13/particles.hd5'
file = h5py.File(name, 'r')

def to_list():
    e_data = []
    for n,ev in file.items():
        for dsetname,dset in ev.items():
            e_data += list(dset['energy'])
    return e_data

%timeit e_data = to_list()
print (len(e_data))



plt.figure()
plt.yscale('log')
%timeit plt.hist(np.log10(np.asarray(e_data)), bins=100, range=(9,14))
plt.show()
```

Analysis, ROOT



- … not yet… trouble with PyROOT