LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Overview about analysis software in HEP

David Koch

Ludwig-Maximilians-Universität München

22. 9. 2022

# ROOT::RDataFrame

ROOT is the jack of all trades in HEP for basically everything from IO to statistical analysis to plotting

RDataFrame is ROOT's recommended way for working with root files
modern and easy to use declarative API

```
tree = ROOT.RDataFrame(
    "mini",  # treename
    "path/to/ntuples"
)
tree = (tree
    .Define("HT", calc_HT)
    .Filter("HT > 500")
)
hist = tree.Histo1D("x")
```

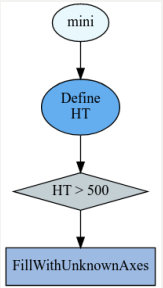
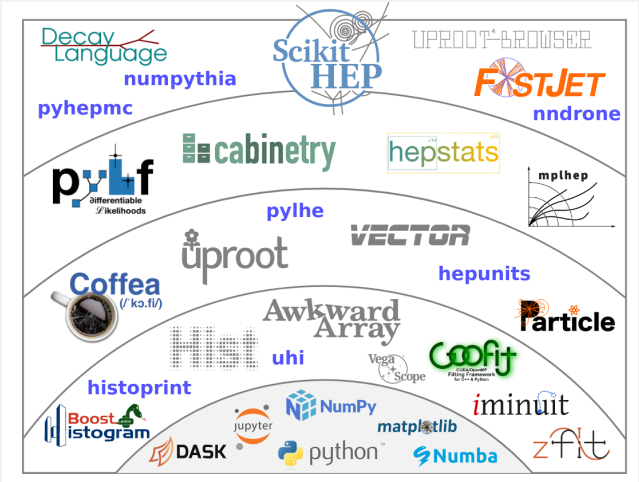



Figure: automatically generated computational graph

# Python ecosystem for analyses



# Python ecosystem for analyses

**numpy** for fast computations with array-like data

```
np.mean(np.random.rand(1000))
```

**matplotlib** for creating plots

```
plt.hist(distribution, n_bins=20)
plt.title("$p_T$")
```

**numba** to compile python functions into machine code

```
@numba.njit
def square(x):
  return x**2
```

**dask** for distributed computing

```
cluster = dask_jobqueue.SLURMCluster(cores=1, queue="mycluster")
cluster.scale(60)
client = dask.Client(cluster)
results = client.map(myfunction, data)
```

# Python ecosystem for analyses

**awkward** for numpy-like operations with jagged arrays

```
x = ak.Array([[1., 1.2], [2.], [], [1., 2., 3.]])
x[x**2 > 4.0]  # select entries from x whose square is larger than 4
```

**hist** and **boost-histogram** as high-level wrappers for `boost::histogram`
**zfit** for fitting

# Python ecosystem for analyses

**uproot** for IO with ROOT files

```
with uproot.open(filename+":"+treename) as tree:
  for data in tree.iterate(library="ak"):
    # process data ...
```

**vector** for computations with 4-vectors

```
lepton_p4 = vector.array(dict(pt=lepton_pt, phi=lepton_phi, eta=lepton_eta, M=lepton_m))
```

**coffea** framework for columnar data analysis in HEP

```
class MyAnalysis(coffea.processor.ProcessorABC):
  def process(self, events):
    selected_electrons = events.electron[events.electron.pt > 25]
    selected_muons = events.muon[events.muon.pt > 25]
    event_filters = ((
      ak.count(selected_electrons.pt, axis=1)
      + ak.count(selected_muons.pt, axis=1)) == 1
    )
    selected_events = events[event_filters]  # ...
```

# Workflow management tools



## Extensive comparison upcoming...

| | Criteria | Luigi | Snakemake | Reana |
|---|---|---|---|---|
| Project | Learning curve | intermediate | simple | simple |
| | Programming languages | single | multiple | multiple |
| | Relation to analysis code | integrated | complete factorization | complete factorization |
| | Workflow language | Python | custom Python-based | Python-based |
| | Boilerplate code | minimal | minimal | intermediate |
| | Data formats | any | any | any |
| | Dependency management | automatic | automatic | automatic |
| | State management | target based | target based | target based |
| Interface | Visualization and monitoring | extensive | no dynamic DAG | no dynamic DAG |
| | Execution control | extensive | extensive | limited |
| | Error handling and debugging | failed output remains | failed output deleted | single-use clean environment |
| | Architecture | single application | single application | server |
| Features | Scalability | easy | easy | easy |
| | Portability | easy | easy | easy |
| | Environment management | minimal | extensive | limited |
| | Storage systems support | extensive | extensive | extensive |
| | Remote execution support | extensive | extensive | extensive |
| | Authentication mechanism | environment variables | environment variables | access tokens |
| | Version control | external | external and internal | external and internal |
| Integration | Installation | pip, non-root | conda, non-root | pip, non-root |
| | Documentation | extensive | extensive | incomplete |
| | Support | extensive | extensive | satisfactory |
| | System developers | Spotify Group | academic team | CERN |
| | History and activity | very active | very active | less active |
| | User community | large | large | significant |
| | Long term perspective | good | good | acceptable |
| | Lock-in | yes | no | no |
| | License | Spotify Group, free use | MIT, free use | CERN |
| | Use in Punch | Belle 2 | LHCb, Radioastronomy | CERN |

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Thank you

for your attention