Data Management Package for the novel data delivery system, ServiceX, and Applications to various physics analysis workflows

KyungEon Choi, Peter Onyisi

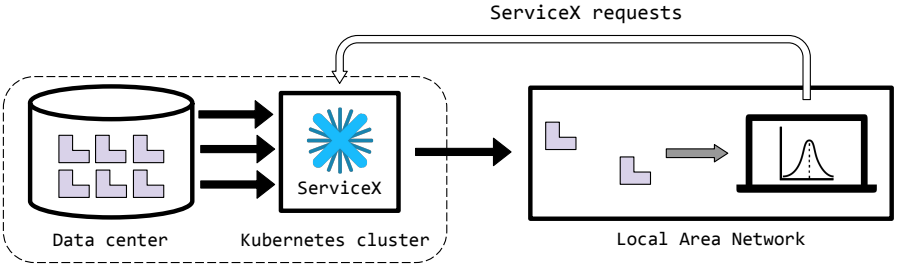
University of Texas at Austin
Department of Physics

CHEP 2023 @ Norfolk, VA
May 11, 2023

TEXAS
The University of Texas at Austin

ATLAS
EXPERIMENT

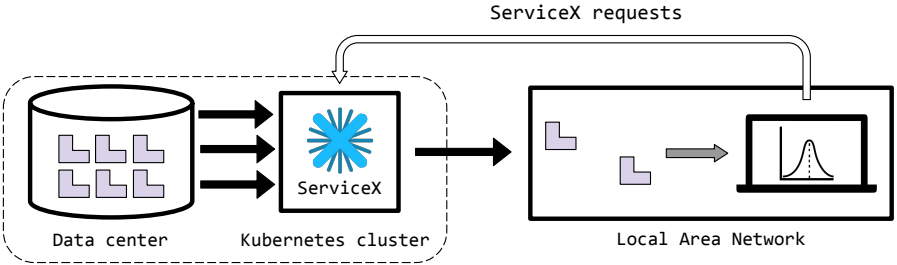
What's ServiceX?



- A service to **easily** and **quickly** access large data at remote with **transformation**
- Deployed in a Kubernetes cluster and usually co-located with the data center to allow a wide network bandwidth.
- See more at Ben Galewsky's talk on the ServiceX

ServiceX readthedocs

What's ServiceX?



- A service to **easily** and **quickly** access large data at remote with **transformation**
- Deployed in a Kubernetes cluster and usually co-located with the data center to allow a wide network bandwidth.
- See more at Ben Galewsky's talk on the ServiceX

Why ServiceX?

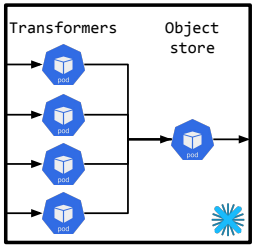
- ✓ A user wants few columns from datasets in the grid for ML study. Grid jobs?
- ✓ What about full-scale analysis?

ServiceX readthedocs

What's ServiceX?

Under the hood

1. ServiceX spawns the so-called **Transformer** pods (Docker container) to read columns and filter events or do whatever it's designed for
2. Transformer pods are also scaled automatically using Kubernetes auto-scaling feature
3. Outputs from each transformer written to object store then delivered to a destination as soon as it becomes available or consumed later



Transformers today

Transformer	Input data format
Uproot (`uproot`)	Flat ROOT ntuple
ATLAS R21 (`atlasr21`)	ATLAS R21 xAOD
CMS Run1 (`cmssw-5-3-32`)	CMS Run-1 AOD
Python (`python`)	*Runs user-provided python function

What is ServiceX DataBinder?

Python library (GitHub , PyPI)

for **easy** configurations of ServiceX delivery requests

and handling of delivered data using a single configuration file

and more

Use-case: Fast column extraction from remote data for ML study

```
General:
  ServiceXName: servicex-uc-af
  Transformer: uproot
  OutputFormat: root
  OutputDirectory: /Users/kchoi/data_for_MLstudy
  WriteOutputDict: fileset_ml_study

Sample:
  - Name: Signal
    RucioDID: user.kchoi:user.kchoi.fcnc_tHq_ML.ttH.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttW.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttZ.v8
    Tree: nominal
    FuncADL: DEF_ttH_nominal_query
  - Name: Background1
    XRootDFiles: DEF_ggH_input
    Tree: mini
    Filter: lep_n>2
    Columns: lep_pt, lep_eta
  - Name: Background2
    Transformer: atlasr21
    RucioDID: DEF_Zee_input
    FuncADL: DEF_Zee_query
  - Name: Background3
    LocalPath: /Users/kchoi/Work/data/fcnc

Definition:
  DEF_ttH_nominal_query: "Where(lambda e: e.met_met>150e3). \
              Select(lambda event: {'el_pt': event.el_pt, 'jet_e': event.jet_e, \
              'jet_pt': event.jet_pt, 'met_met': event.met_met})"
  DEF_ggH_input: "root://eospublic.cern.ch//eos/opendata/atlas/OutreachDatasets\
                  /2020-01-22/4lep/MC/mc_345060.ggH125_ZZ4lep.4lep.root"
  DEF_Zee_input: "mc15_13TeV:mc15_13TeV.361106.PowhegPythia8EvtGen_AZNLOCTEQ6L1_Zee.\
                  merge.DAOD_STDM3.e3601_s2576_s2132_r6630_r6264_p2363_tid05630052_00"
  DEF_Zee_query: "SelectMany('lambda e: e.Jets(\"AntiKt4EMTopoJets\")'). \
              Where('lambda j: (j.pt() / 1000) > 30'). \
              Select('lambda j: j.pt() / 1000.0'). \
              AsROOTTTree('junk.root', 'my_tree', [\"JetPt\"])"
```

Base Transformer in this config

File paths of delivered files written in a yaml

4 Samples are defined

Keep configuration tidy! → `Definition` block

OutputFormat can be root or parquet

Path where data will be delivered

Remote data via Rucio (RSE)

Multiple inputs separated by comma

TCut syntax for Uproot (`Filter` only for scalar type variables)

Background2 is only available as DAOD, thus extract using xAOD transformer

Files for Background3 is already at local storage → Just bind it together

Use-case: Fast column extraction from remote data for ML study

Base Transformer in this config

File paths of delivered files written in a yaml

OutputFormat can be root or parquet

Path where data will be delivered

Remote data via Rucio (RSE)

Multiple inputs separated by comma

4 Samples are defined

TCut syntax for Uproot
(`Filter` only for scalar type variables)

Background2 is only available as DAOD, thus extract using xAOD transformer

Files for Background3 is already at local storage → Just bind it together

Keep configuration tidy!
→ `Definition` block

```yaml
General:
  ServiceXName: servicex-uc-af
  Transformer: uproot
  OutputFormat: root
  OutputDirectory: /Users/kchoi/data_for_MLstudy
  WriteOutputDict: fileset_ml_study

Sample:
  - Name: Signal
    RucioDID: user.kchoi:user.kchoi.fcnc_tHq_ML.ttH.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttW.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttZ.v8
    Tree: nominal
    FuncADL: DEF_ttH_nominal_query
  - Name: Background1
    XRootDFiles: DEF_ggH_input
    Tree: mini
    Filter: lep_n>2
    Columns: lep_pt, lep_eta
  - Name: Background2
    Transformer: atlasr21
    RucioDID: DEF_Zee_input
    FuncADL: DEF_Zee_query
  - Name: Background3
    LocalPath: /Users/kchoi/Work/data/fcnc

Definition:
  DEF_ttH_nominal_query: "Where(lambda e: e.met_met>150e3). \
              Select(lambda event: {'el_pt': event.el_pt, 'jet_e': event.jet_e, \
              'jet_pt': event.jet_pt, 'met_met': event.met_met})"
  DEF_ggH_input: "root://eospublic.cern.ch//eos/opendata/atlas/OutreachDatasets\
                  /2020-01-22/4lep/MC/mc_345060.ggH125_ZZ4lep.4lep.root"
  DEF_Zee_input: "mc15_13TeV:mc15_13TeV.361106.PowhegPythia8EvtGen_AZNLOCTEQ6L1_Zee.\
                  merge.DAOD_STDM3.e3601_s2576_s2132_r6630_r6264_p2363_tid05630052_00"
  DEF_Zee_query: "SelectMany('lambda e: e.Jets(\"AntiKt4EMTopoJets\")'). \
              Where('lambda j: (j.pt() / 1000) > 30'). \
              Select('lambda j: j.pt() / 1000.0'). \
              AsROOTTTree('junk.root', 'my_tree', [\"JetPt\"])"
```

Use-case: Fast column extraction from remote data for ML study

Base Transformer in this config

File paths of delivered files written in a yaml

4 Samples are defined

OutputFormat can be root or parquet

Path where data will be delivered

Remote data via Rucio (RSE)

Multiple inputs separated by comma

TCut syntax for Uproot
(`Filter` only for scalar type variables)

Background2 is only available as DAOD, thus extract using xAOD transformer

Files for Background3 is already at local storage → Just bind it together

Keep configuration tidy!
→ `Definition` block

```yaml
General:
  ServiceXName: servicex-uc-af
  Transformer: uproot
  OutputFormat: root
  OutputDirectory: /Users/kchoi/data_for_MLstudy
  WriteOutputDict: fileset_ml_study

Sample:
  - Name: Signal
    RucioDID: user.kchoi:user.kchoi.fcnc_tHq_ML.ttH.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttW.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttZ.v8
    Tree: nominal
    FuncADL: DEF_ttH_nominal_query
  - Name: Background1
    XRootDFiles: DEF_ggH_input
    Tree: mini
    Filter: lep_n>2
    Columns: lep_pt, lep_eta
  - Name: Background2
    Transformer: atlasr21
    RucioDID: DEF_Zee_input
    FuncADL: DEF_Zee_query
  - Name: Background3
    LocalPath: /Users/kchoi/Work/data/fcnc

Definition:
  DEF_ttH_nominal_query: "Where(lambda e: e.met_met>150e3). \
              Select(lambda event: {'el_pt': event.el_pt, 'jet_e': event.jet_e, \
              'jet_pt': event.jet_pt, 'met_met': event.met_met})"
  DEF_ggH_input: "root://eospublic.cern.ch//eos/opendata/atlas/OutreachDatasets\
                  /2020-01-22/4lep/MC/mc_345060.ggH125_ZZ4lep.4lep.root"
  DEF_Zee_input: "mc15_13TeV:mc15_13TeV.361106.PowhegPythia8EvtGen_AZNLOCTEQ6L1_Zee.\
                  merge.DAOD_STDM3.e3601_s2576_s2132_r6630_r6264_p2363_tid05630052_00"
  DEF_Zee_query: "SelectMany('lambda e: e.Jets(\"AntiKt4EMTopoJets\")'). \
              Where('lambda j: (j.pt() / 1000) > 30'). \
              Select('lambda j: j.pt() / 1000.0'). \
              AsROOTTTree('junk.root', 'my_tree', [\"JetPt\"])"
```

Use-case: Fast column extraction from remote data for ML study

Base Transformer in this config

File paths of delivered files written in a yaml

4 Samples are defined

```yaml
General:
  ServiceXName: servicex-uc-af
  Transformer: uproot
  OutputFormat: root
  OutputDirectory: /Users/kchoi/data_for_MLstudy
  WriteOutputDict: fileset_ml_study

Sample:
  - Name: Signal
    RucioDID: user.kchoi:user.kchoi.fcnc_tHq_ML.ttH.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttW.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttZ.v8
    Tree: nominal
    FuncADL: DEF_ttH_nominal_query
  - Name: Background1
    XRootDFiles: DEF_ggH_input
    Tree: mini
    Filter: lep_n>2
    Columns: lep_pt, lep_eta
  - Name: Background2
    Transformer: atlasr21
    RucioDID: DEF_Zee_input
    FuncADL: DEF_Zee_query
  - Name: Background3
    LocalPath: /Users/kchoi/Work/data/fcnc

Definition:
  DEF_ttH_nominal_query: "Where(lambda e: e.met_met>150e3). \
            Select(lambda event: {'el_pt': event.el_pt, 'jet_e': event.jet_e, \
             'jet_pt': event.jet_pt, 'met_met': event.met_met})"
  DEF_ggH_input: "root://eospublic.cern.ch//eos/opendata/atlas/OutreachDatasets\
              /2020-01-22/4lep/MC/mc_345060.ggH125_ZZ4lep.4lep.root"
  DEF_Zee_input: "mc15_13TeV:mc15_13TeV.361106.PowhegPythia8EvtGen_AZNLOCTEQ6L1_Zee.\
               merge.DAOD_STDM3.e3601_s2576_s2132_r6630_r6264_p2363_tid05630052_00"
  DEF_Zee_query: "SelectMany('lambda e: e.Jets(\"AntiKt4EMTopoJets\")'). \
            Where('lambda j: (j.pt() / 1000) > 30'). \
            Select('lambda j: j.pt() / 1000.0'). \
            AsROOTTTree('junk.root', 'my_tree', [\"JetPt\"])"
```

Keep configuration tidy!
→ `Definition` block

OutputFormat can be root or parquet

Path where data will be delivered

Remote data via Rucio (RSE)

Multiple inputs separated by comma

TCut syntax for Uproot
(`Filter` only for scalar type variables)

Background2 is only available as DAOD, thus extract using xAOD transformer

Files for Background3 is already at local storage → Just bind it together

Use-case: Fast column extraction from remote data for ML study

Base Transformer in this config

File paths of delivered files written in a yaml

4 Samples are defined

Keep configuration tidy!
→ `Definition` block

```yaml
General:
  ServiceXName: servicex-uc-af
  Transformer: uproot
  OutputFormat: root
  OutputDirectory: /Users/kchoi/data_for_MLstudy
  WriteOutputDict: fileset_ml_study

Sample:
  - Name: Signal
    RucioDID: user.kchoi:user.kchoi.fcnc_tHq_ML.ttH.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttW.v8,
              user.kchoi:user.kchoi.fcnc_tHq_ML.ttZ.v8
    Tree: nominal
    FuncADL: DEF_ttH_nominal_query
  - Name: Background1
    XRootDFiles: DEF_ggH_input
    Tree: mini
    Filter: lep_n>2
    Columns: lep_pt, lep_eta
  - Name: Background2
    Transformer: atlasr21
    RucioDID: DEF_Zee_input
    FuncADL: DEF_Zee_query
  - Name: Background3
    LocalPath: /Users/kchoi/Work/data/fcnc

Definition:
  DEF_ttH_nominal_query: "Where(lambda e: e.met_met>150e3). \
            Select(lambda event: {'el_pt': event.el_pt, 'jet_e': event.jet_e, \
             'jet_pt': event.jet_pt, 'met_met': event.met_met})"
  DEF_ggH_input: "root://eospublic.cern.ch//eos/opendata/atlas/OutreachDatasets\
                  /2020-01-22/4lep/MC/mc_345060.ggH125_ZZ4lep.4lep.root"
  DEF_Zee_input: "mc15_13TeV:mc15_13TeV.361106.PowhegPythia8EvtGen_AZNLOCTEQ6L1_Zee.\
                  merge.DAOD_STDM3.e3601_s2576_s2132_r6630_r6264_p2363_tid05630052_00"
  DEF_Zee_query: "SelectMany('lambda e: e.Jets(\"AntiKt4EMTopoJets\")'). \
            Where('lambda j: (j.pt() / 1000) > 30'). \
            Select('lambda j: j.pt() / 1000.0'). \
            AsROOTTTree('junk.root', 'my_tree', [\"JetPt\"])"
```

OutputFormat can be root or parquet

Path where data will be delivered

Remote data via Rucio (RSE)

Multiple inputs separated by comma

TCut syntax for Uproot
(`Filter` only for scalar type variables)

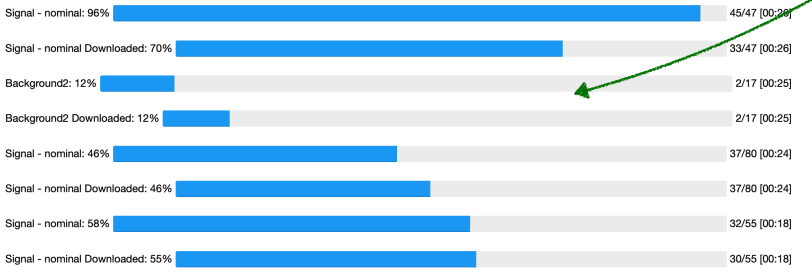
Background2 is only available as DAOD, thus extract using xAOD transformer

Files for Background3 is already at local storage → Just bind it together

FuncADL for Uproot
FuncADL for xAOD

Use-case: Fast column extraction from remote data for ML study

```
In [1]: from servicex_databinder import DataBinder

In [2]: sxdb = DataBinder('config_MLstudy.yaml')

INFO - Loading DataBinder config file: config_MLstudy.yaml
INFO -   4 Samples and 5 ServiceX requests

In [*]: out = sxdb.deliver()

INFO - Deliver via ServiceX endpoint: https://servicex.af.uchicago.edu/
```




```
INFO -   Background1 | mini | ['root://eospublic.cern.ch//eos/opendata/atlas/OutreachDatasets/2020-01-22/4lep/MC/mc_345060.ggH125_ is delivered
```

5 ServiceX requests running in parallel

- Rucio datasets for **`Signal`** in ROOT Ntuple ~200 GB
- Rucio dataset for **`Background2`** in ATLAS xAOD ~75 GB
- Deliver 1-4 columns to my laptop in Norfolk
- **Wall time: 1min 6s**

How to run ServiceX DataBinder?

1. Prepare ServiceX access file ⟹
   - YAML file containing ServiceX endpoint information

```yaml
api_endpoints:
  - name: servicex-uc-af
    endpoint: https://servicex.af.uchicago.edu/
    token: <TOKEN>
    type: uproot
```

2. Prepare DataBinder configuration file ⟹

```yaml
General:
  ServiceXName: servicex-uc-af
  OutputFormat: root

Sample:
  - Name: ggH125_ZZ4lep
    XRootDFiles: "root://eospublic.cern.ch//eos/opendata/atlas/OutreachDatasets\
                  /2020-01-22/4lep/MC/mc_345060.ggH125_ZZ4lep.4lep.root"
    Tree: mini
    FuncADL: "Select(lambda event: {'lep_pt': event.lep_pt, 'lep_eta': event.lep_eta})"
```

3. Deliver!! ⟹

```python
from servicex_databinder import DataBinder
sx_db = DataBinder('config_minimum.yaml')
out = sx_db.deliver()
```

How to run ServiceX DataBinder?

In the Jupyter notebook,

```
In [1]: from servicex_databinder import DataBinder

In [2]: sxdb = DataBinder('config_minimum.yaml')

INFO - Loading DataBinder config file: config_minimum.yaml
INFO -   1 Samples and 1 ServiceX requests
```

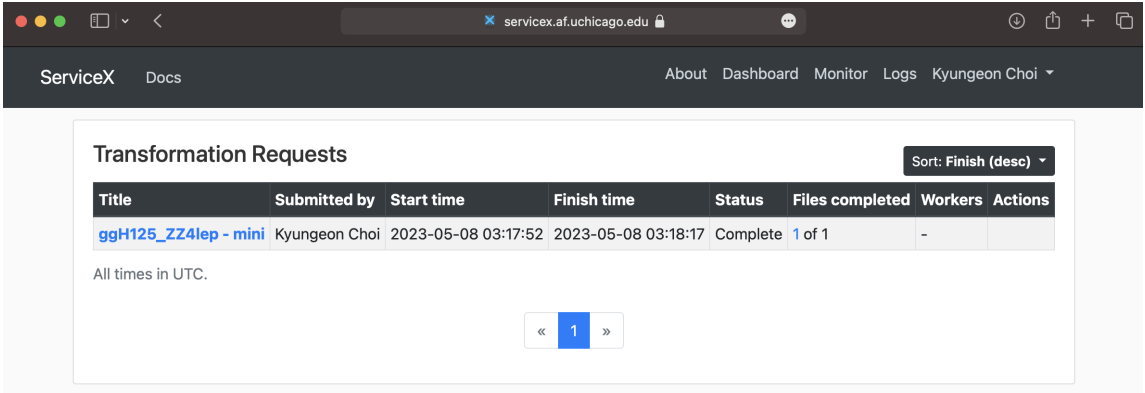
How to run ServiceX DataBinder?

In the Jupyter notebook,

```
In [1]: from servicex_databinder import DataBinder

In [2]: sxdb = DataBinder('config_minimum.yaml')

INFO - Loading DataBinder config file: config_minimum.yaml
INFO -   1 Samples and 1 ServiceX requests

In [*]: out = sxdb.deliver()

INFO - Deliver via ServiceX endpoint: https://servicex.af.uchicago.edu/

ggH125_ZZ4lep - mini: 0%                                    0/1 [00:05]

ggH125_ZZ4lep - mini Downloaded: 0%                         0/1 [00:05]
```

How to run ServiceX DataBinder?

Also from ServiceX Dashboard,



How to run ServiceX DataBinder?

In the Jupyter notebook,

```
In [1]: from servicex_databinder import DataBinder

In [2]: sxdb = DataBinder('config_minimum.yaml')

INFO - Loading DataBinder config file: config_minimum.yaml
INFO -   1 Samples and 1 ServiceX requests

In [3]: out = sxdb.deliver()

INFO - Deliver via ServiceX endpoint: https://servicex.af.uchicago.edu/
INFO -   ggH125_ZZ4lep | mini | ['root://eospublic.cern.ch//eos/opendata/atlas/OutreachDatasets/2020-01-22/4lep/MC/
mc_345060.ggH125_ is delivered
INFO - Delivered at /Users/kchoi/Work/UTAustin/Computing/ServiceX/ServiceXDataBinder/ServiceXData
```

How to run ServiceX DataBinder?

In the Jupyter notebook,

In [1]:
```
from servicex_databinder import DataBinder
```

In [2]:
```
sxdb = DataBinder('config_minimum.yaml')
```

```
INFO - Loading DataBinder config file: config_minimum.yaml
INFO -   1 Samples and 1 ServiceX requests
```

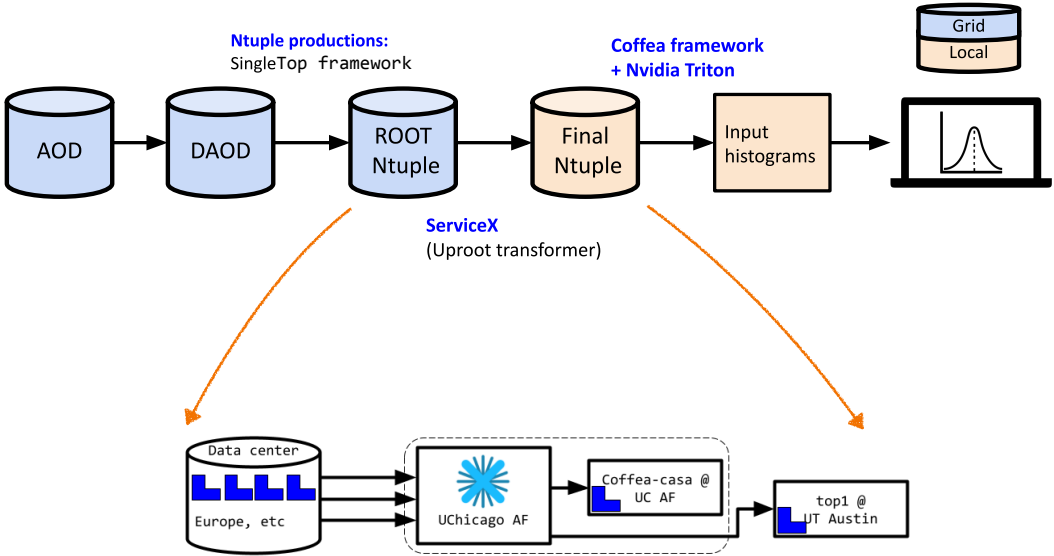
In [3]:
```
out = sxdb.deliver()
```

```
INFO - Deliver via ServiceX endpoint: https://servicex.af.uchicago.edu/
INFO -   ggH125_ZZ4lep | mini | ['root://eospublic.cern.ch//eos/opendata/atlas/OutreachDatasets/2020-01-22/4lep/MC/mc_345060.ggH125_ is delivered
INFO - Delivered at /Users/kchoi/Work/UTAustin/Computing/ServiceX/ServiceXDataBinder/ServiceXData
```

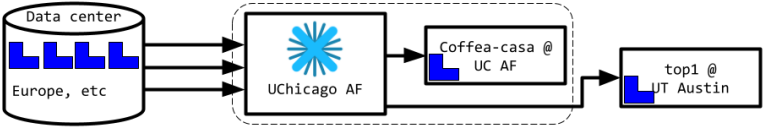
In [4]:
```
out
```

Out[4]:
```
{'ggH125_ZZ4lep': {'mini': ['/Users/kchoi/Work/UTAustin/Computing/ServiceX/ServiceXDataBinder/ServiceXData/ggH125_ZZ4lep/mini/root___192.170.240.18_1094__root___eospublic.cern.ch__eos_opendata_atlas_OutreachDatasets_2020-01-22_4lep_MC_mc_345060.ggH125_ZZ4lep.4lep.root.root']}}
```

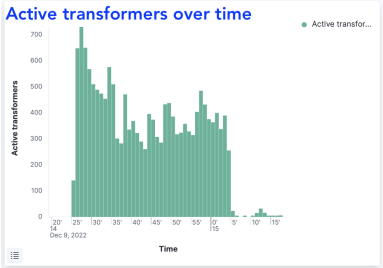
Use-case: ATLAS Run-2 Physics Analysis



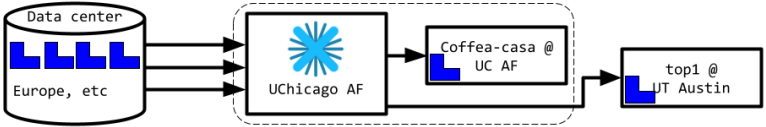
Use-case: ATLAS Run-2 Physics Analysis



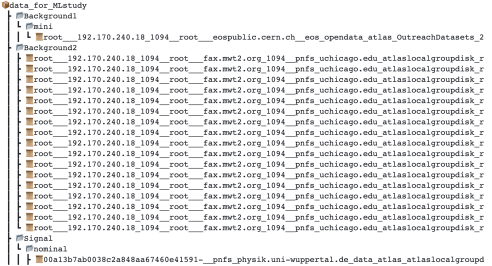
- ServiceX spawns 140 Uproot transformer pods and scaled up/down (up to ∼800 pods) to extract 30-70 columns from 130 trees out of ∼1.1TB ROOT ntuples (>600 Rucio datasets)
- Wall time: ∼53mins to UC Analysis Facility (AF)
- DataBinder configuration file only about 400 lines
- Returned fileset from DataBinder is directly passed to `coffea` as input
- Accommodate various needs
  - ✓ Delivered data can be also processed with ROOT
  - ✓ Data can be delivered to AF, or University cluster



Use-case: ATLAS Run-2 Physics Analysis

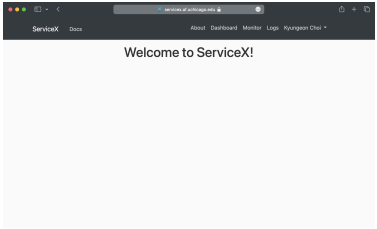


- ServiceX spawns 140 Uproot transformer pods and scaled up/down (up to ∼800 pods) to extract 30-70 columns from 130 trees out of ∼1.1TB ROOT ntuples (>600 Rucio datasets)
- Wall time: ∼53mins to UC Analysis Facility (AF)
- DataBinder configuration file only about 400 lines
- Returned fileset from DataBinder is directly passed to `coffea` as input
- Accommodate various needs
  - ✓ Delivered data can be also processed with ROOT
  - ✓ Data can be delivered to AF, or University cluster



Where is ServiceX?

https://servicex.af.uchicago.edu



https://coffea.casa

https://coffea-opendata.casa

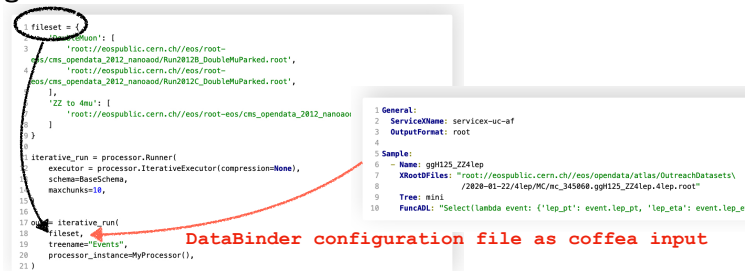
- https://servicex.af.uchicago.edu at University of Chicago Analysis Facility for ATLAS users
- https://coffea.casa at University of Nebraska-Lincoln (UNL) for CMS users
- https://coffea-opendata.casa at UNL for OpenData access

  $\Rightarrow$ ServiceX endpoint accessible only inside UNL coffea-casa facilities but ServiceX access file is pre-generated for users

More about coffea-casa at Oksana's Plenary talk

Future plans

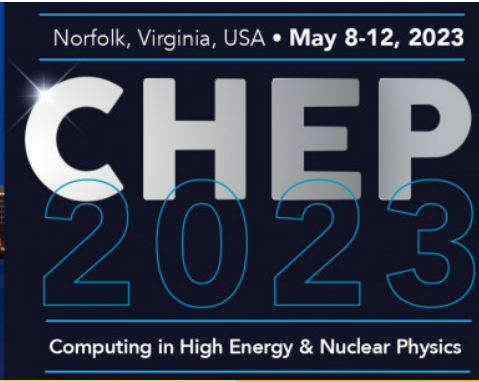
- ✓ **Support other ServiceX data delivery options**
  - Streaming from ServiceX object-store or from local path
- ✓ **Better integration into `coffea` framework**



- ✓ **Support (upcoming) ServiceX updates**
  - multi-tree Uproot transformer, Python transformer, ATLAS R22 transformer, ...
- ✓ **Towards more generic data management package**

Questions?

Special thanks to the ServiceX team!



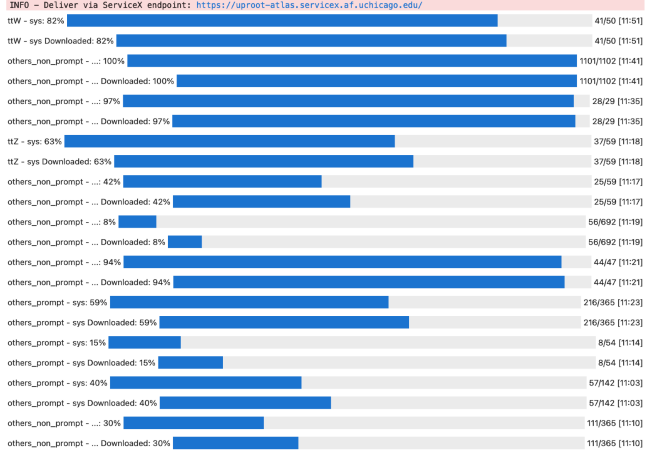
Backup

Delivery of full datasets - a snippet of DataBinder config

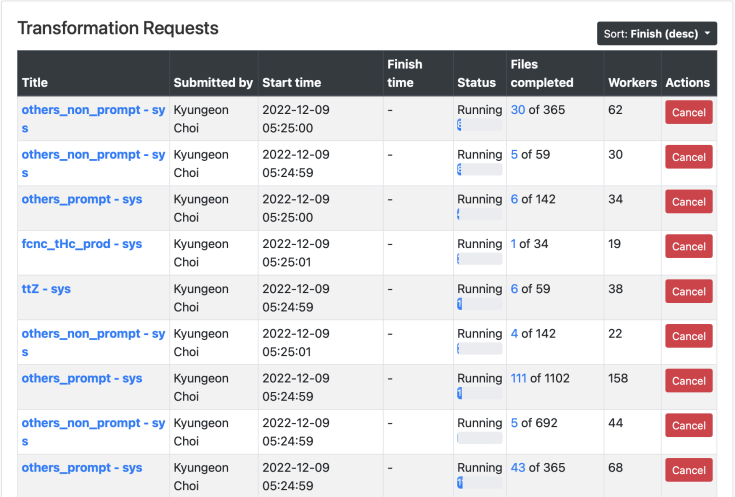
```yaml
General:
    ServiceXBackendName: uproot_uc_af
    OutputDirectory: /data_ceph/kyungeon/fcnc_tHq_ML/ServiceXData_v7
    OutputFormat: root
    WriteOutputDict: out_paths_v7

Sample:
  - Name: ttH
    RucioDID: user.kchoi:user.kchoi.fcnc_tHq_ML.ttH.v7
    TransformerImage: kyungeonchoi/servicex_func_adl_uproot_transformer:fcnc_nominal
    Tree: nominal
    FuncADL: DEF_funcadl_prompt_NOMINAL

  - Name: ttH
    RucioDID: user.kchoi:user.kchoi.fcnc_tHq_ML.ttH.v7
    TransformerImage: kyungeonchoi/servicex_func_adl_uproot_transformer:fcnc_fullsim
    Tree: sys_fullsim
    FuncADL: DEF_funcadl_prompt_SYS

    ...

Definition:
  DEF_funcadl_prompt_NOMINAL: "Where(lambda e: e.el_truthIFFClass.Where(lambda i: i==2).Count() == e.el_truthIFFCl
  Where(lambda e: e.mu_truthIFFClass.Where(lambda i: i==4).Count() == e.mu_truthIFFClass.Count()).
  Select(lambda e: {'mu_pt': e.mu_pt, 'mu_eta': e.mu_eta, 'mu_phi': e.mu_phi, 'mu_e': e.mu_e, 'mu_charge': e.mu_ch
                    'el_pt': e.el_pt, 'el_eta': e.el_eta, 'el_phi': e.el_phi, 'el_e': e.el_e, 'el_charge': e.el_ch
                    'met_met': e.met_met, 'met_phi': e.met_phi,
                    'jet_pt': e.jet_pt, 'jet_eta': e.jet_eta, 'jet_phi': e.jet_phi, 'jet_e': e.jet_e, 'jet_tagWeig
                    'weights': e.weight_mc*e.weight_pileup*e.weight_leptonSF*e.weight_bTagSF_DL1r_Continuous*e.wei
                    'mcChannelNumber': e.mcChannelNumber, 'runNumber': e.runNumber, 'eventNumber': e.eventNumber,
                    'mc_generator_weights': e.mc_generator_weights,
```

Delivery of full datasets - Run DataBinder in Jupyter Notebook

```
[2]: from servicex_databinder import DataBinder

[9]: sx_db = DataBinder('config_fcnc_v7.yml')

INFO – Loading DataBinder config file: config_fcnc_v7.yml
INFO –   21 Samples and 61 ServiceX requests

[*]: %%time
out = sx_db.deliver()

INFO – Deliver via ServiceX endpoint: https://uproot-atlas.servicex.af.uchicago.edu/
```



Delivery of full datasets - Dashboard



Transformation Requests

Sort: **Finish (desc)**

Title	Submitted by	Start time	Finish time	Status	Files completed	Workers	Actions
others_non_prompt - sys	Kyungeon Choi	2022-12-09 05:25:00	-	Running	30 of 365	62	Cancel
others_non_prompt - sys	Kyungeon Choi	2022-12-09 05:24:59	-	Running	5 of 59	30	Cancel
others_prompt - sys	Kyungeon Choi	2022-12-09 05:25:00	-	Running	6 of 142	34	Cancel
fcnc_tHc_prod - sys	Kyungeon Choi	2022-12-09 05:25:01	-	Running	1 of 34	19	Cancel
ttZ - sys	Kyungeon Choi	2022-12-09 05:24:59	-	Running	6 of 59	38	Cancel
others_non_prompt - sys	Kyungeon Choi	2022-12-09 05:25:01	-	Running	4 of 142	22	Cancel
others_prompt - sys	Kyungeon Choi	2022-12-09 05:24:59	-	Running	111 of 1102	158	Cancel
others_non_prompt - sys	Kyungeon Choi	2022-12-09 05:24:59	-	Running	5 of 692	44	Cancel
others_prompt - sys	Kyungeon Choi	2022-12-09 05:24:59	-	Running	43 of 365	68	Cancel