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
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 📅
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?
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

<table>
<thead>
<tr>
<th>Transformer</th>
<th>Input data format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uproot (uproot)</td>
<td>Flat ROOT ntuple</td>
</tr>
<tr>
<td>ATLAS R21 (atlasr21)</td>
<td>ATLAS R21 xAOD</td>
</tr>
<tr>
<td>CMS Run1 (cmssw-5-3-32)</td>
<td>CMS Run-1 AOD</td>
</tr>
<tr>
<td>Python (python)</td>
<td>*Runs user-provided python function</td>
</tr>
</tbody>
</table>
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

Base Transformer in this configuration

File paths of delivered files written in a yaml

4 Samples are defined

Remote data via Rucio (RSE)

Multiple inputs separated by comma

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

OutputFormat can be root or parquet

Path where data will be delivered

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

TCut syntax for Uproot

OutputFormat can be root or parquet

Base Transformer in this config

File paths of delivered files written in a yaml

4 Samples are defined

Remote data via Rucio (RSE)

Multiple inputs separated by comma

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

OutputFormat can be root or parquet

Path where data will be delivered

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
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

Multiple inputs separated by comma

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

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

OutputFormat can be root or parquet

Path where data will be delivered

Remote data via Rucio (RSE)
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

TCut syntax for Uproot

Definition block

OutputFormat can be root or parquet

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

TCut syntax for Uproot

Definition block
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

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

OutputFormat can be root or parquet

Path where data will be delivered

Remote data via Rucio (RSE)

Multiple inputs separated by comma

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

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

5/12

ServiceX DataBinder: Data Management Package for ServiceX @ CHEP 2023

KyungEon Choi (UT Austin)
Use-case: Fast column extraction from remote data for ML study

- 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

2. Prepare DataBinder configuration file

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,

```python
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,

```python
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/

<table>
<thead>
<tr>
<th>ggH125_ZZ4lep - mini: 0%</th>
<th>0/1 [00:05]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ggH125_ZZ4lep - mini Downloaded: 0%</td>
<td>0/1 [00:05]</td>
</tr>
</tbody>
</table>
```
How to run ServiceX DataBinder?

Also from ServiceX Dashboard,

Transformation Requests

<table>
<thead>
<tr>
<th>Title</th>
<th>Submitted by</th>
<th>Start time</th>
<th>Finish time</th>
<th>Status</th>
<th>Files completed</th>
<th>Workers</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ggH125_ZZ4lep - mini</td>
<td>Kyungeon Choi</td>
<td>2023-05-08 03:17:52</td>
<td>2023-05-08 03:18:17</td>
<td>Complete</td>
<td>1 of 1</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

All times in UTC.
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/mc345060.ggh125_is delivered
INFO - Delivered at /Users/kchoi/Work/UTAustin/Computing/ServiceX/ServiceXDataBinder/ServiceXData
```
How to run ServiceX DataBinder?

In the Jupyter notebook,

```python
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

```
Use-case: ATLAS Run-2 Physics Analysis

- **ServiceX DataBinder**: Data Management Package for ServiceX @ CHEP 2023

- **Top framework**
  - ServiceX spawns 140 Uproot transformer pods and scaled up/down (up to \( \sim 800 \) pods) to extract 30-70 columns from 130 trees out of \( \sim 1.1 \text{TB} \) ROOT ntuples (\( > 600 \) Rucio datasets)

- **Wall time**: \( \sim 53 \text{mins} \) 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

---

**Diagram: ServiceX Data Management Flow**

- **Ntuple productions**: SingleTop framework
- **Coffea framework + Nvidia Triton**
- **ServiceX (Uproot transformer)**
- **Data center**: Europe, etc
- **UC AF**: top1 @ UT Austin
Use-case: ATLAS Run-2 Physics Analysis

- ServiceX spawns 140 Uproot transformer pods and scaled up/down (up to \( \sim 800 \) pods) to extract 30-70 columns from 130 trees out of \( \sim 1.1 \)TB ROOT ntuples (>600 Rucio datasets)
- Wall time: \( \sim 53 \)mins 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
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** 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

⇒ 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

General:
- ServiceXBackendName: uproot_uc_sf
- OutputDirectory: /data_ceph/kyungeon/fcnc_thQ_ML/ServiceXData_v7
- OutputFormat: root
- WriteOutputDict: out_paths_v7

Sample:
- Name: tth
  - RucioID: 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
  - RucioID: 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

...
Delivery of full datasets - Run DataBinder in Jupyter Notebook

```python
from serviceX.databinder import DataBinder

sx_db = DataBinder('config_fcnv_v7.yml')

INFO - Loading DataBinder config file: config_fcnv_v7.yml
INFO - 21 Samples and 81 ServiceX requests

INFO - Deliver via ServiceX endpoint: https://uproot-atlas.serviceX.uchicago.edu/
```
### Transformation Requests

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