ServiceX
@AS Workshop
The ServiceX Team

- University of Chicago
  - R. Gardner,
  - I. Vukotic,
  - M. Weinberg
- University of Illinois at Urbana-Champaign
  - M. Neubauer,
  - B. Galewsky
- Princeton University
  - J. Pivarski
- University of Washington
  - G. Watts
  - M. Proffit
What is ServiceX?

A distributed, caching columnar datasource for HEP event data
What is ServiceX?

A distributed, caching columnar datasource for HEP event data

• Distributed:
  • Easily deployed on Kubernetes Cluster
  • Collocated with the Data Lake
  • Federated ServiceX instances
What is ServiceX?

A distributed, caching columnar datasource for HEP event data

• Distributed:
  • Easily deployed on Kubernetes Cluster
  • Collocated with the Data Lake
  • Federated ServiceX instances

• Caching:
  • Extracted n-tuples and simple transformations
  • Data for specific analysis
What is ServiceX?

A distributed, caching columnar datasource for HEP event data

• Distributed:
  • Easily deployed on Kubernetes Cluster
  • Collocated with the Data Lake
  • Federated ServiceX instances

• Caching:
  • Extracted n-tuples and simple transformations
  • Data for specific analysis

• Columnar
  • Awkward Arrays
ServiceX Contract

- **Inputs:**
  - Dataset Identifier (DID)
  - Func DL select
  - Data transformer image name

- **Outputs**
  - Stream of Awkward Arrays
Architecture
Component Details

- **API Gateway**
  - Generic API Gateway handles authentication
  - Routes requests to appropriate service
- **DID Finder**
  - Communicates with Rucio to resolve DID to ROOT file paths
- **C++ Code Generator**
  - Generates ROOT C++ code from FuncDL select
  - Caches generated code
Component Details

- C++ Transformer
  - Based on specific ATLAS or CMS Docker Image
  - Compiles C++ code
  - Caches compiled binary
  - Executes code to:
    - Extract requested columns
    - Perform simple transformation
    - Perform simple filters
  - Outputs stream of awkward arrays
  - Can be scaled horizontally to improve throughput
Component Details

• Columnar Cache
  • Saves output from transformer
  • Additional columns can be added to existing results to satisfy related requests

• Cache Manager
  • Efficiently perform selections and projections on transformed dataset
  • Stream awkward arrays to analysis system
Component Details

• Kafka
  • Popular big data streaming service
  • Data is published to topics
  • Topics are sharded to allow easy scaling
  • Messages are compressed awkward arrays
  • Topics can be replayed to re-run analysis
  • Message processing can be transactional to handle analysis job failure
Analysis Use Cases

- Grid with “Pull” semantics
- *Coffea* on Spark using Spark structured streaming
Current Status

1. Basic REST service
2. Connection to Rucio
3. Transformer container works with xAOD files
   a. Slow - based on pyRoot
   b. Only understands list of columns, not FuncDL selects
4. Streaming service
5. No Columnar Cache

Runs in Kubernetes