Server Side Data Delivery using FAB @ CERN

MANIAC LAB

HTC 2023

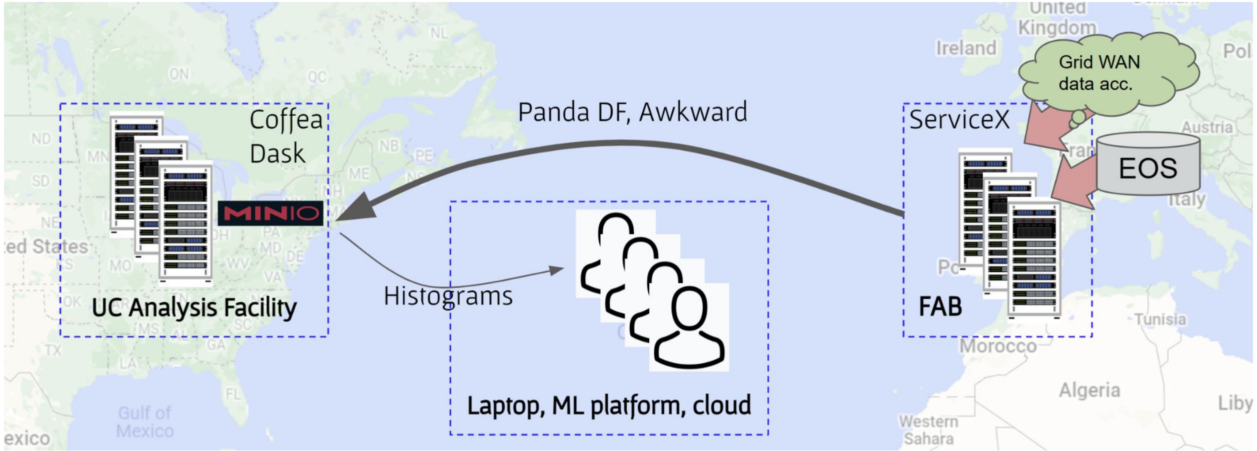
Madison, July 12, 2023

Ilija Vukotic, University of Chicago

ATLAS EXPERIMENT

A demonstrator to inform future LHC computing models

- Deploy ServiceX at CERN (to filter and reformat data on the Tier0)
- Deliver only columnar data directly to analysis facilities, e.g. in the US but potentially elsewhere too
- Examine resulting 1) turn around time and 2) transatlantic bandwidth reduction
- Details here.

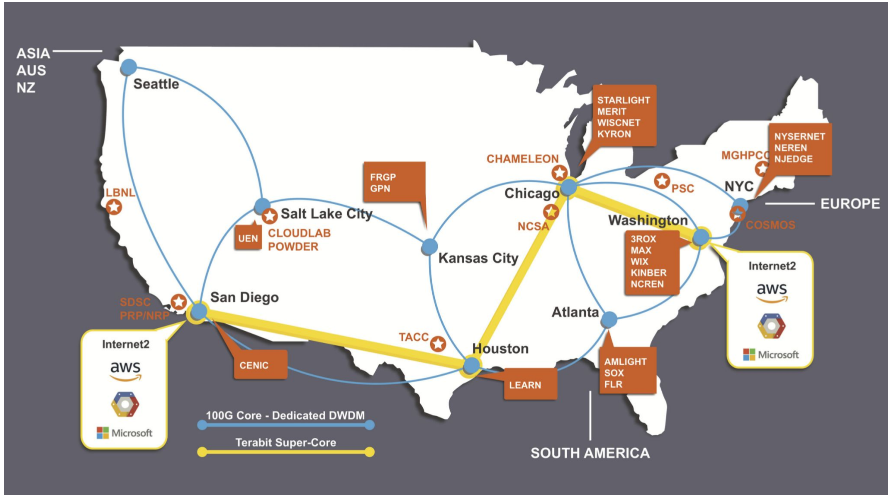


A bit of background

First there was FABRIC. It is an NSF funded network testbed operated by ESnet where one can run experiments in areas of networking, distributed computing, storage, ML, etc.

Main components:

- an everywhere programmable network interconnected by dedicated optical links
- cutting-edge infrastructure for computer science, AI, data-intensive research
- software and support



A bit of background II

Then there is FAB (FABRIC Across Borders). It added five international sites to the FABRIC testbed, including CERN:

Details on hardware and slice setup in Fengping's talk.

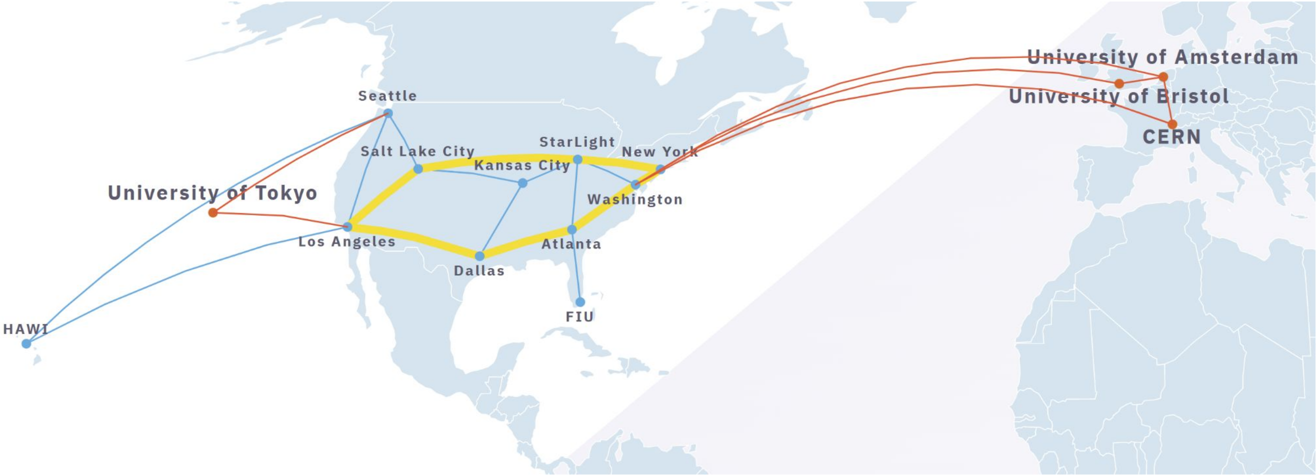
University of Tokyo
Japan

CERN, the European Organization for Nuclear Research
Switzerland

University of Bristol
U.K.

University of Amsterdam
The Netherlands

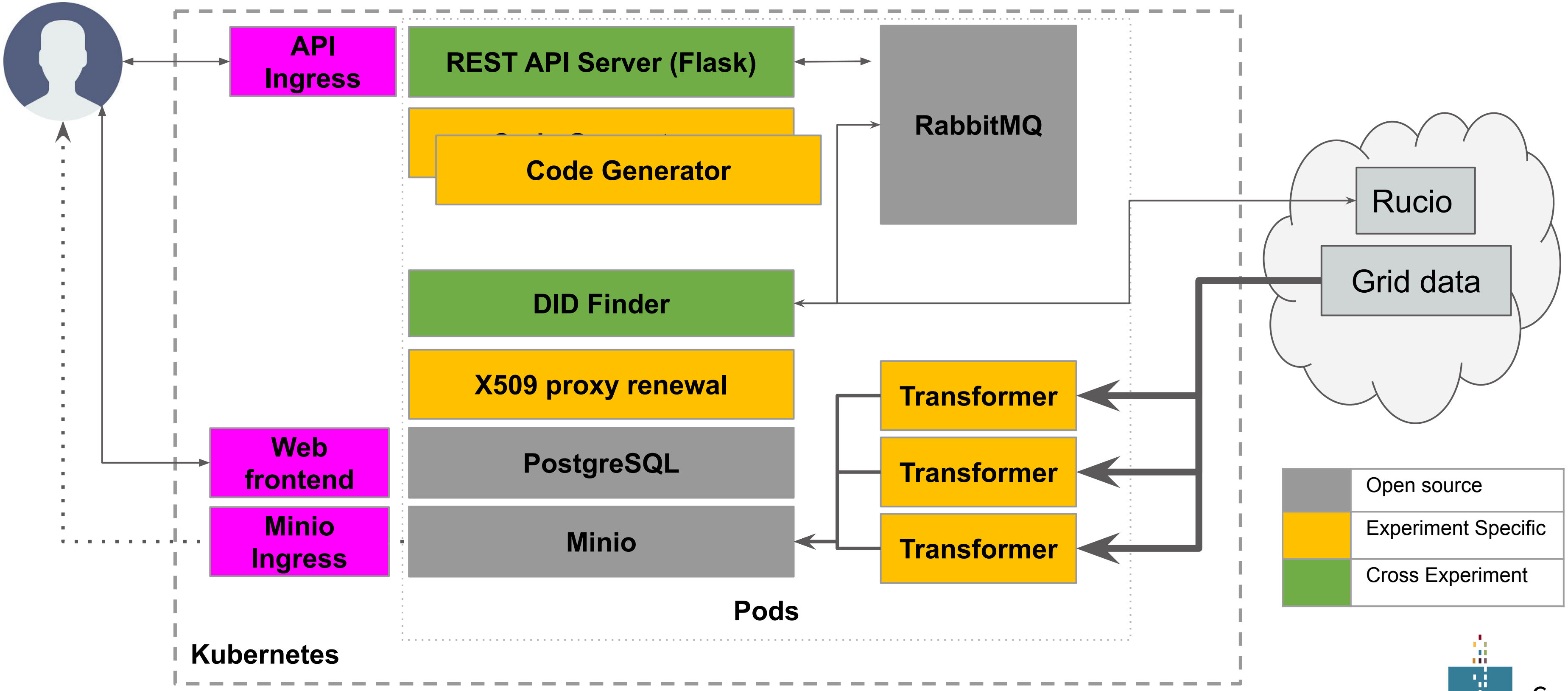
CPTEC/INPE
Brazil



What is ServiceX ?

- A service that quickly **filters** and **delivers** data.
- **Filtering** here means skimming, slimming and augmenting input data. Input data can be xAODs or flat ROOT files.
- Resulting data can be **delivered** as PyArrow Awkward arrays or flat ROOT files.
- By default data gets delivered to a Minio instance (or any S3 storage) and also gets cached client-side.

ServiceX components - all on K8s



API Ingress

REST API Server (Flask)

RabbitMQ

Code Generator

DID Finder

X509 proxy renewal

PostgreSQL

Minio

Transformer

Transformer

Transformer

Web frontend

Minio Ingress

Rucio

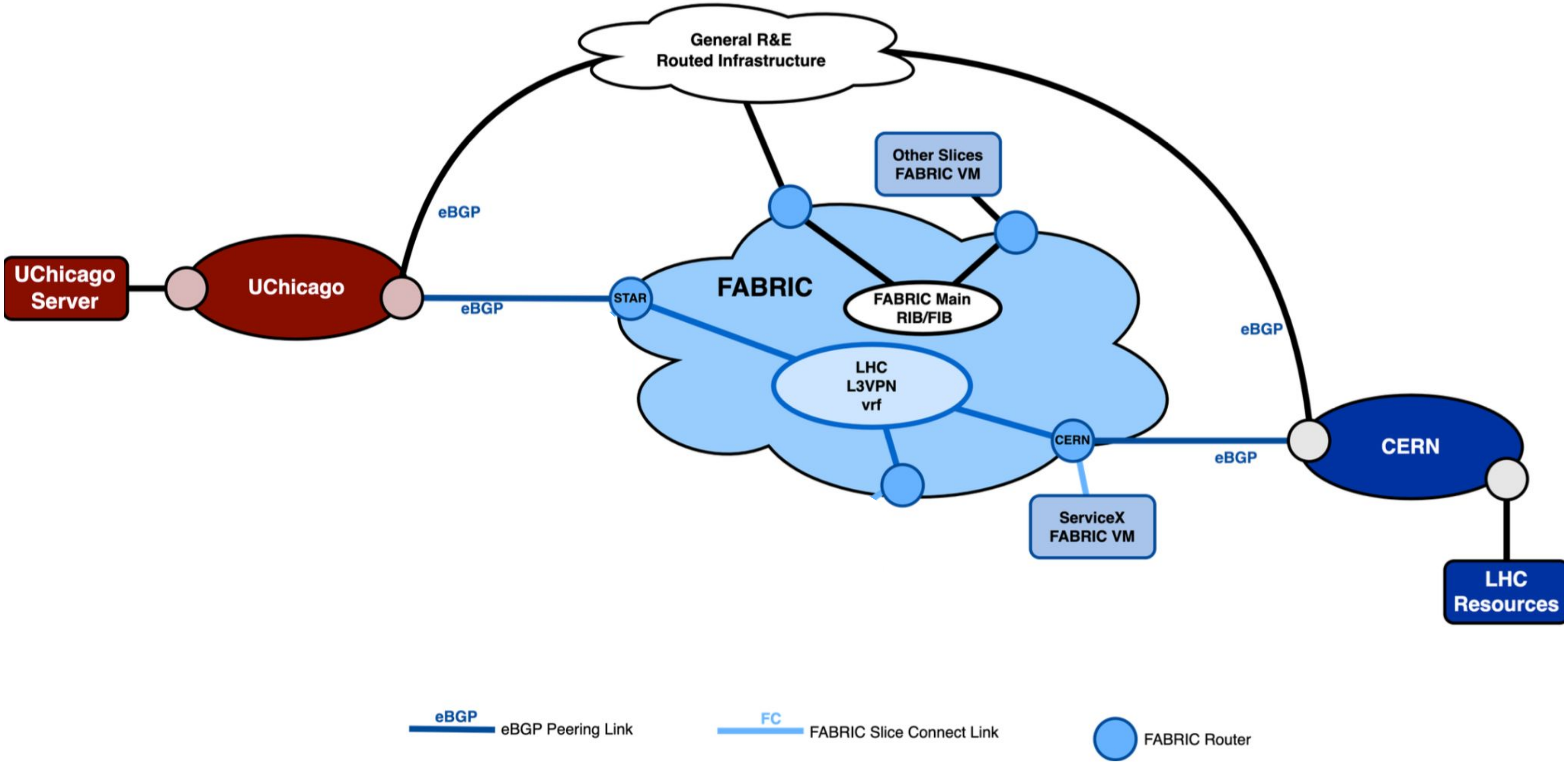
Grid data

Pods

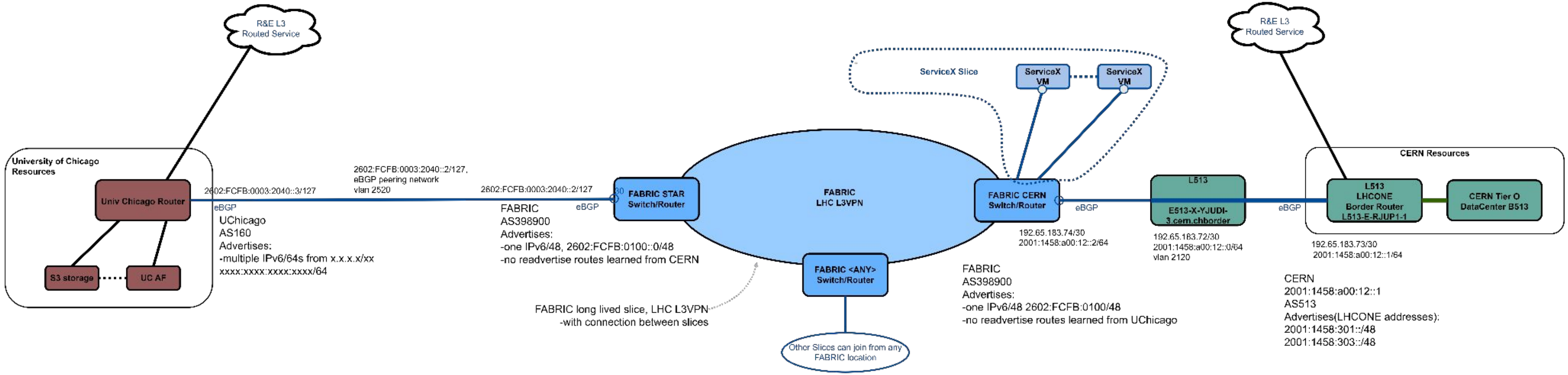
Kubernetes

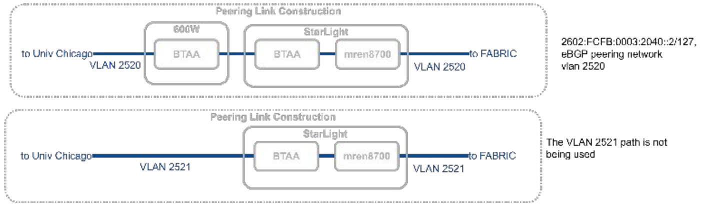
▨	Open source
🟨	Experiment Specific
🟩	Cross Experiment

Network layout (big picture)



Network peering for optimal data flow

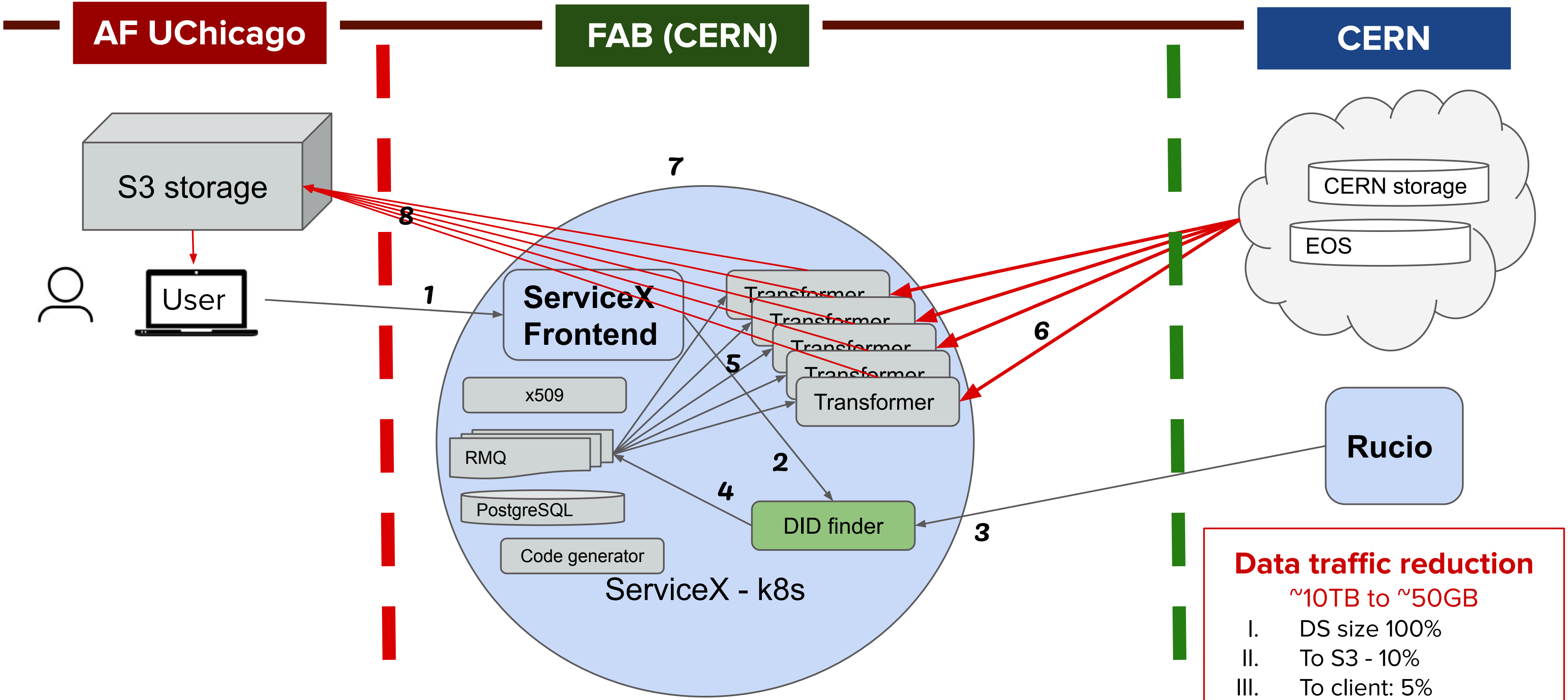




Several meetings with ESnet, UC networking, and Fengping established specific peering for the demonstrator

- We should have very fast connection to CERN EOS
- Use Fabric routes all the way to Starlight at Chicago
- IPv6 only

Services: CERN, FAB & Chicago AF



Data traffic reduction

~10TB to ~50GB

I. DS size 100%
II. To S3 - 10%
III. To client: 5%

Deploying into FABRIC

- A Slice is created (using the Fabric web fronted).
- Peering Network attached.
- Use Fabric Jupyter notebook to create:
  - Kubernetes cluster
  - Setup ingress controller
  - Cert manager
  - Storage
  - Flux CD
  - Sealed secrets created in an S3 bucket to automate deployment.
- Verify network connectivity
- ServiceX deployed via github kustomization

FAB ServiceX customizations

RMQ: Cluster does not allow pods to change ulimits. Solved by adding: `ulimitNofiles: ""`

Postgresql: no storage, but not needed. Turned off.

Transformer: xcache disabled.

S3: Added s3 on AF, serving on IPv6.

GlobusAuth: it does not support IPv6. Removed.

Cluster changes

By default linux prefers IPv4, it retries it 6 times and a single timeout is 10 seconds. This makes Nginx timeout before even trying IPv6. We disable DNS resolving A records (in coredns):

- hooks.slack.com
- s3.af.uchicago.edu
- hub.opensciencegrid.org
- hub.docker.com
- voatlasrucio-server-prod.cern.ch
- voatlasrucio-auth-prod.cern.ch

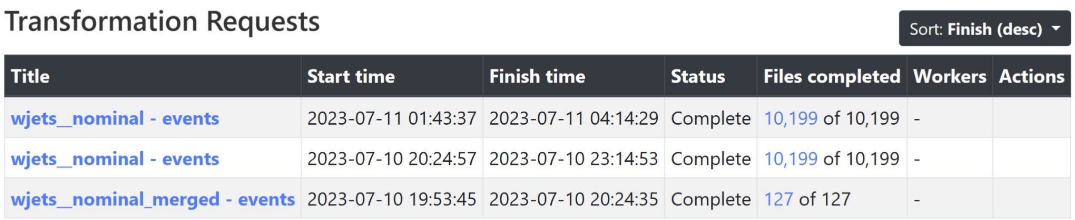
For github access we use nat64.

Functionality Testing

Simplest test - a single file on EOS with given full path, delivery to S3 at AF, works fine.

Larger tests - contacting Rucio to get paths to files. Works but way too slow.

- Exposed issue in Rucio replica sorting. Doesn’t return CERN replica first. Fix in a PR.
- CERN IP addresses according to MaxMind are in lake next to Bern. Which made CSCS computing centar appear closest. Fixed by making a request to MaxMind.
- Exposed issues in ServiceX (processing RMQ messages in main thread).



Transformation Requests

Sort: Finish (desc)

Title	Start time	Finish time	Status	Files completed	Workers	Actions
wjets_nominal - events	2023-07-11 01:43:37	2023-07-11 04:14:29	Complete	10,199 of 10,199	-	
wjets_nominal - events	2023-07-10 20:24:57	2023-07-10 23:14:53	Complete	10,199 of 10,199	-	
wjets_nominal_merged - events	2023-07-10 19:53:45	2023-07-10 20:24:35	Complete	127 of 127	-	

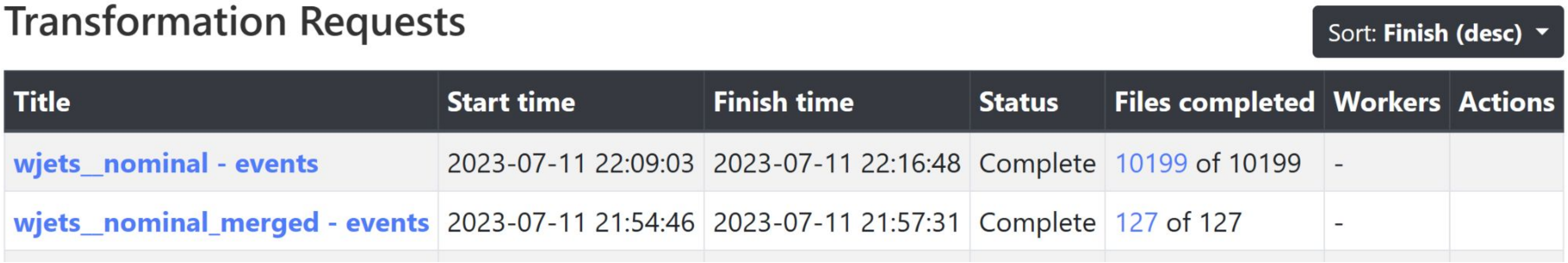
ServiceX changes

- special server version for easy debugging
- adding annotations to pods to enable peering network k8s.v1.cni.cncf.io/networks: macvlan-conf

Performance - current state

With manually tuned ServiceX on FAB, we run 1.2 TB sample in 10199 files in around 7 minutes, from ~450 transformers.

We process it in 3 min from ~50 transformers if the same data is in 127 larger files. That means we still have at least a factor of 2 to improve.



Transformation Requests

Title	Start time	Finish time	Status	Files completed	Workers	Actions
wjets_nominal - events	2023-07-11 22:09:03	2023-07-11 22:16:48	Complete	10199 of 10199	-	
wjets_nominal_merged - events	2023-07-11 21:54:46	2023-07-11 21:57:31	Complete	127 of 127	-	

Performance testing plan

- Have a sample of around 100TB of xAOD data at CERN (EOS ATLAS data disk).
- Run an Analysis Grand Challenge like workflow.
  - On CERN side use up to 750 transformers.
  - On AF side use a Dask cluster large enough so not it be a bottleneck.
  - Measure bandwidth used, time to completion.
- Move all of the data to MWT2, measure time to move. Repeat analysis with same transformer scaling settings.
- We expect factor 10 reduction in bandwidth used compared to moving all the data over first.
- Factor 2 reduction in bandwidth compared to remote access from AF going through xcache. While xcache transfers only things that are requested, block size makes you still transfer more than needed.

Future

- **FABRIC**
  - Automatic attaching of the peering network
  - Way to modify existing slice
- **Rucio** - fix replica sorting.
- **ServiceX** - fix RMQ use, reduce per file overhead.
- **Measure** - potential impacts (analysis latency, WAN bandwidth reduction).