Energy Efficient Climate Science Workflows with *PyActiveStorage*

Varsiha Sothilingam[1], David Hassell[1], Bryan Lawrence[1], Max Norton[2] , Valeriu Predoi[1]
[1] NCAS-CMS, University of Reading
[2] StackHPC

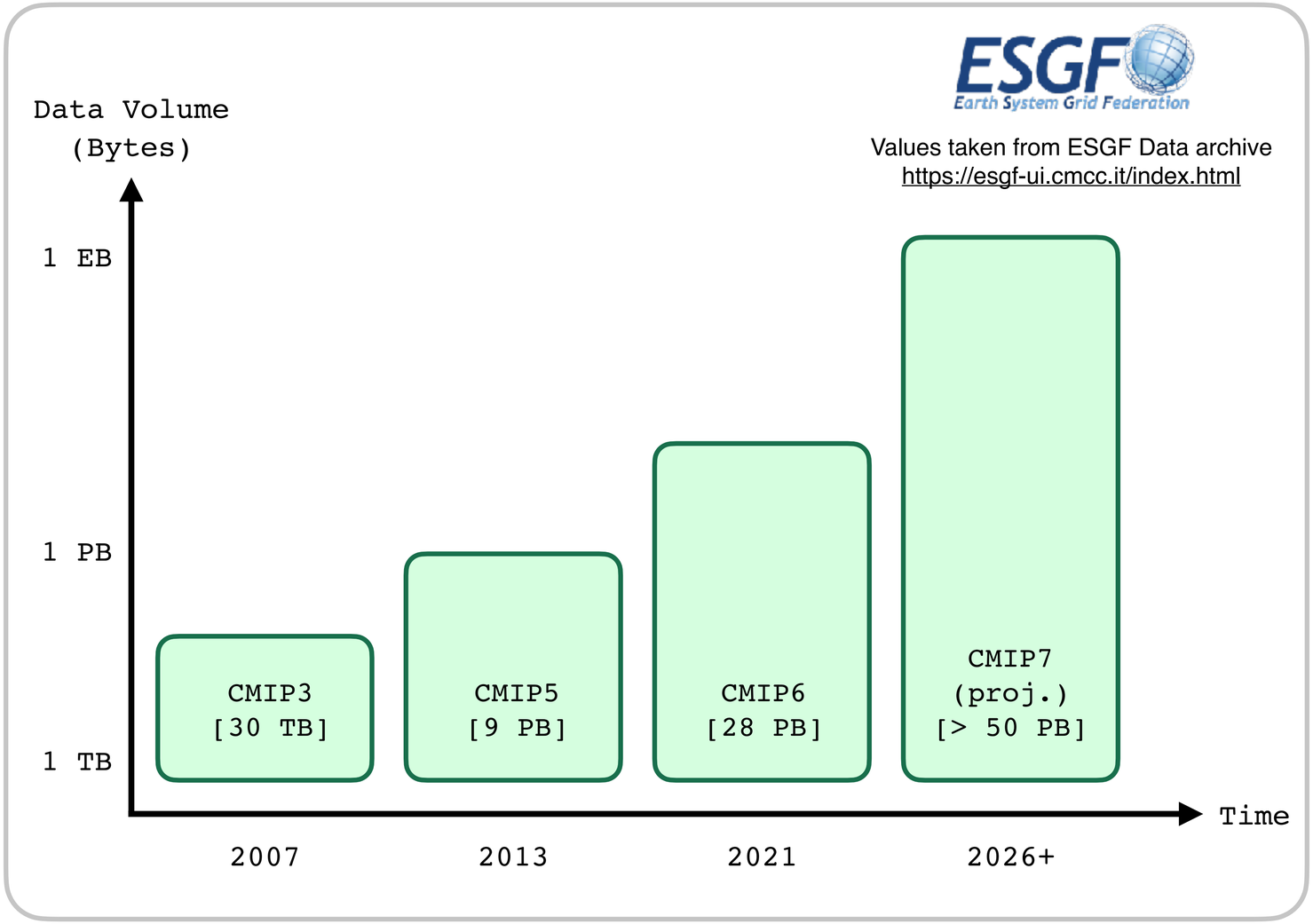
varsiha.sothilingam@ncas.ac.uk
STEP-UP RSLondon2026, *29th June 2026*
doi: 10.5281/zenodo.21217495

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

StackHPC

Exascale Era of Climate Modelling



- Climate models increase in complexity over time, leading to larger datasets.
- The **relation between data storage and computation** must be considered in future workflows

Beyond Climate Modelling

Image 1: "Green rocky hills", National Centre for Atmospheric Science, CC BY-NC-SA 2.0

Image 2: Credit: NASA/CXC/CfA/M. Markevitch; NASA/STScI; Magellan/U. Arizona/D. Clowe; ESO WFI

Image 3: Source: Genomics Education Programme, NHS England

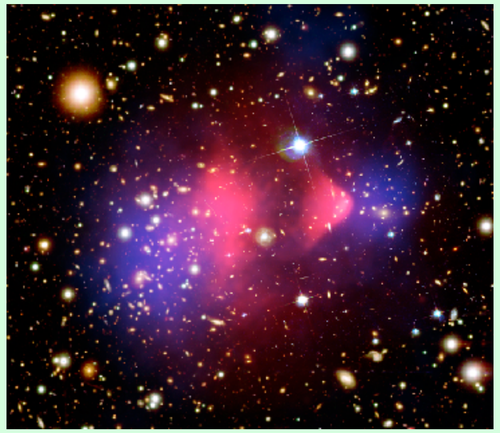
Values taken from the paper: Data adapted from Stephens et al. (2015)

The challenge is not unique.
Many data centric fields are entering exabyte scales and will need to adapt.

Climate Modelling

Future climate and weather models require exabyte storage

Astronomy



Order EB of data stored per year by experiments

Genomics

Complexity of genomic data produces 2-40 EB of data stored per year

Exascale Challenges

High Latency

- Larger wait times for streaming data
- Bottleneck for analysis workflows

Unsustainable Energy Costs

- Moving data is energy intensive
- CPUs sit idle while waiting for disk I/O

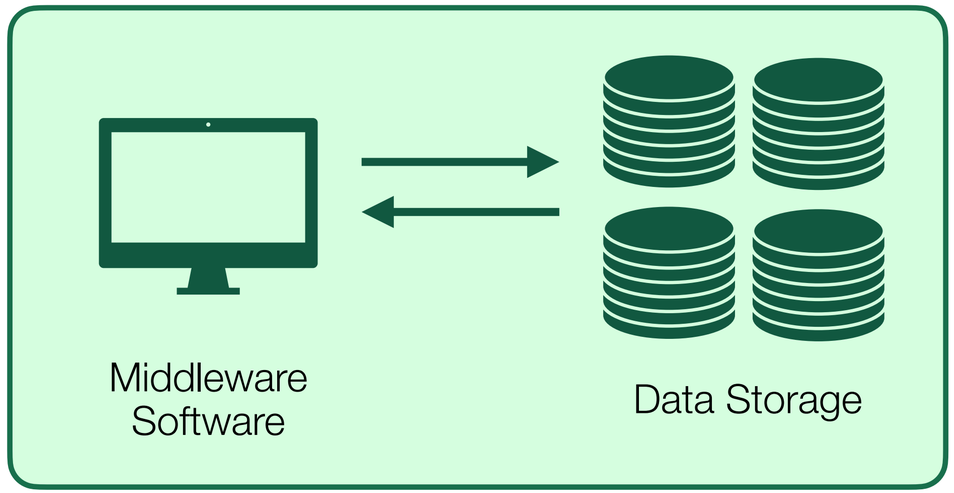
Limited Storage Resources

- Storage of extra copies is expensive
- Not all scientists have access to resources

Solution: Active Storage

→ *More efficient to move the computation than the data*

Active Storage System



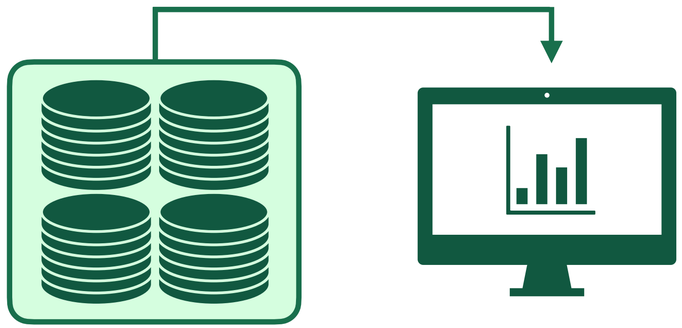
- Modern storage systems have **under utilised compute**
- Perform computation in available computation inside or near storage.

  → Custom middleware software
- **WAN traffic mitigated** as data and middleware are within the same firewall.
- Only results of computation will be read out to users

Active Storage in Workflows

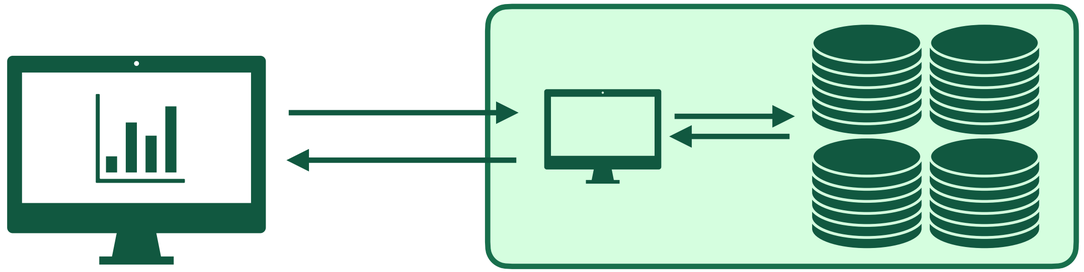
Restructure of workflows to use active storage

Local Workflow



Streaming large amounts of data to a machine to perform computation

Active Workflow



Send computation request to remote **middleware software** located **inside the storage** network

Only result of computation sent from storage back to the machine

Active Storage Live @ CEDA-JASMIN

pyactivestorage.readthedocs.io
stackhpc.github.io/reductionist-rs

JASMIN

Didcot, Oxford

JASMIN is a data analysis facility providing storage and compute facilities for data-intensive environmental science research.

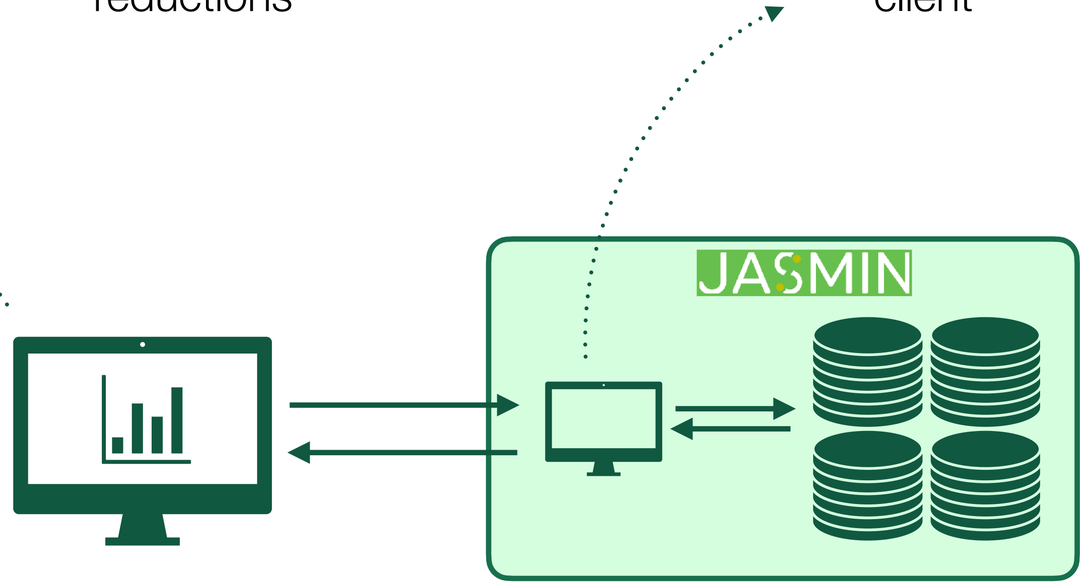
It is designed, integrated and operated by the Science & Technology Facilities Council (STFC) on behalf of the Natural Environment Research Council (NERC).

PyActiveStorage

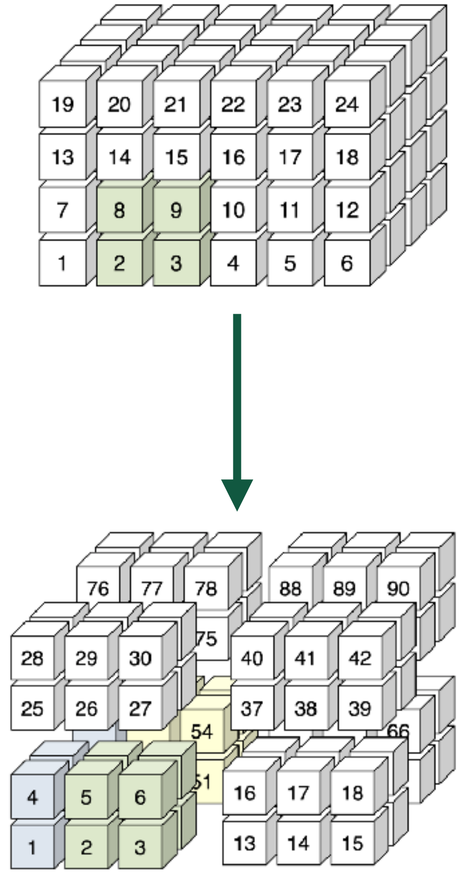
Python client for users can use to communicate with the **Reductionist middleware** requesting data reductions

Reductionist

Receives and **computes** data **reduction requests**, sending results back to PyActiveStorage client



Data Chunking



Data chunking schematic

Climate data is chunked, with a chunking index, which allowing for:

- **Processing efficiency:** handling required subsets instead of the entire dataset
- **Reduce memory usage:** loading only relevant chunks
- **Enables parallel processing:** allowing multiple chunks to be processed simultaneously.

Specialised tools for reading chunked data such as Pyfive which is a highly optimised pure-python HDF5/netCDF4 reader.

Py[five]

pyfive.readthedocs.io

PyActiveStorage : Python Client

Users choice of **dataset stored** remotely, and perform **operation/reduction method** on data

Compatible with different data storage backends:

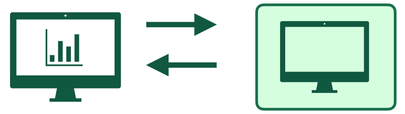
- `s3` buckets
- `https` servers

Reductionist middleware can perform computes, inspired by MPI *data reductions*.

Methods:

minimum, maximum, count, sum, mean

Loop over chunk index, requesting *Reductionist* middleware to **return reduction of each chunk**



Return result to user

PyActiveStorage: User Code

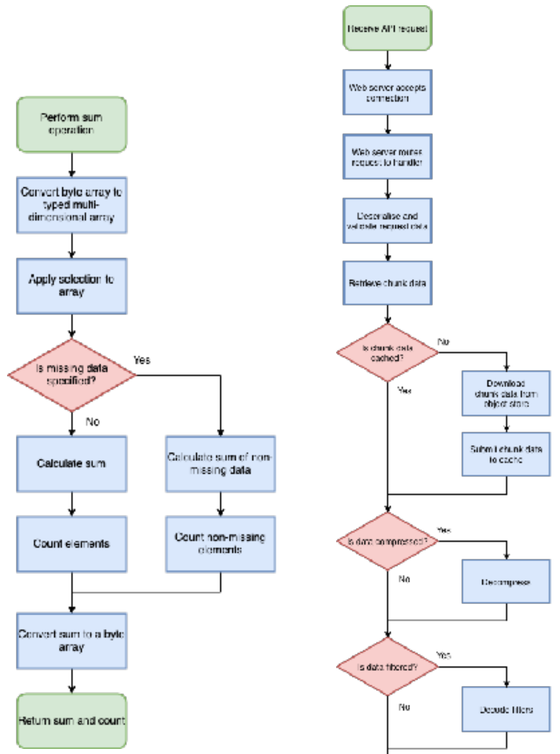
S3 Protocol

```
import os

from activestorage.active import Active

def s3_file():
    """Run a simple active storage instance test with S3 file."""

    S3_BUCKET = "bucket"
    storage_options = {
        'key': "KEY",
        'secret': "SECRET",
        'client_kwargs': {'endpoint_url': "https://uor-aces-o.s3-
ext.jc.rl.ac.uk"},
    }

    active_storage_url = "https://reductionist.jasmin.ac.uk/"
    a_file = "ch330a.pc19790301-def.nc"

    test_file_uri = os.path.join(
        S3_BUCKET,
        a_file
    )

    active = Active(test_file_uri, ncvar='UM_m01s16i202_vn1106',
                    storage_options=storage_options,
                    active_storage_url=active_storage_url)
    active._version = 2
    result = active.min[0:3, 4:6, 7:9]

    return result
```

HTTPS Protocol

```
from activestorage.active import Active

def https_file():
    """Run a simple active storage instance test with https file."""

    test_file_uri = "https://esgf.ceda.ac.uk/thredds/fileServer/..."

    active = Active(test_file_uri, ncvar="cl")
    active._version = 2

    result = active.min[0:3, 4:6, 7:9]

    return result
```

Reductionist

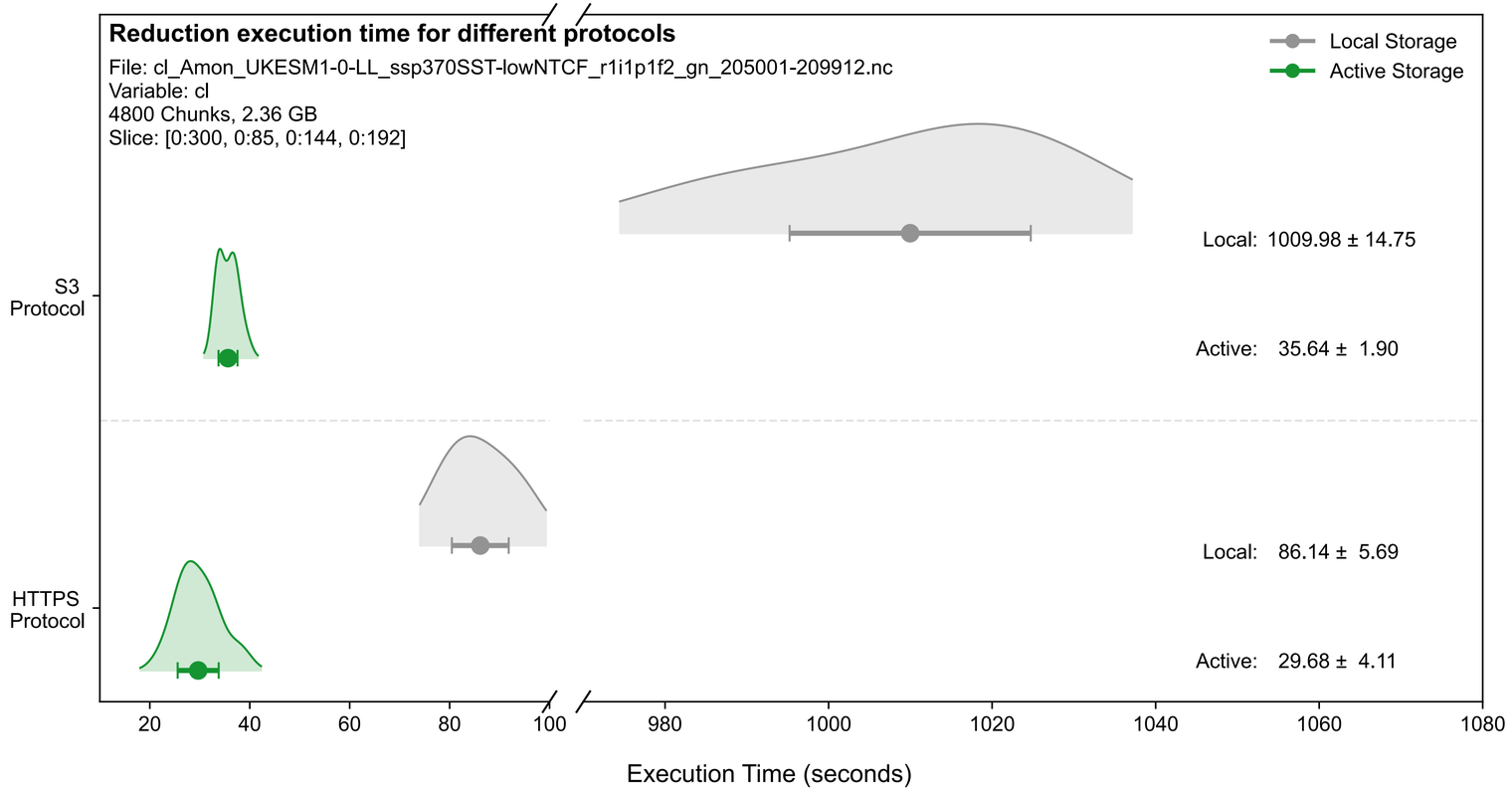


Architecture and implementation of Reductionist

- It is an **ansible deployed container**, which can easily scale and be deployed to more servers as the data storage grows.
- Receives request for chunk reduction and applies reduction, returning result to PyActiveStorage Client
- Features:
  - Numerical operations to perform on data chunks
  - Load balancing from HAProxy server
  - Disk caching of chunk information
  - Decompress and decode filters

stackhpc.github.io/reductionist-rs

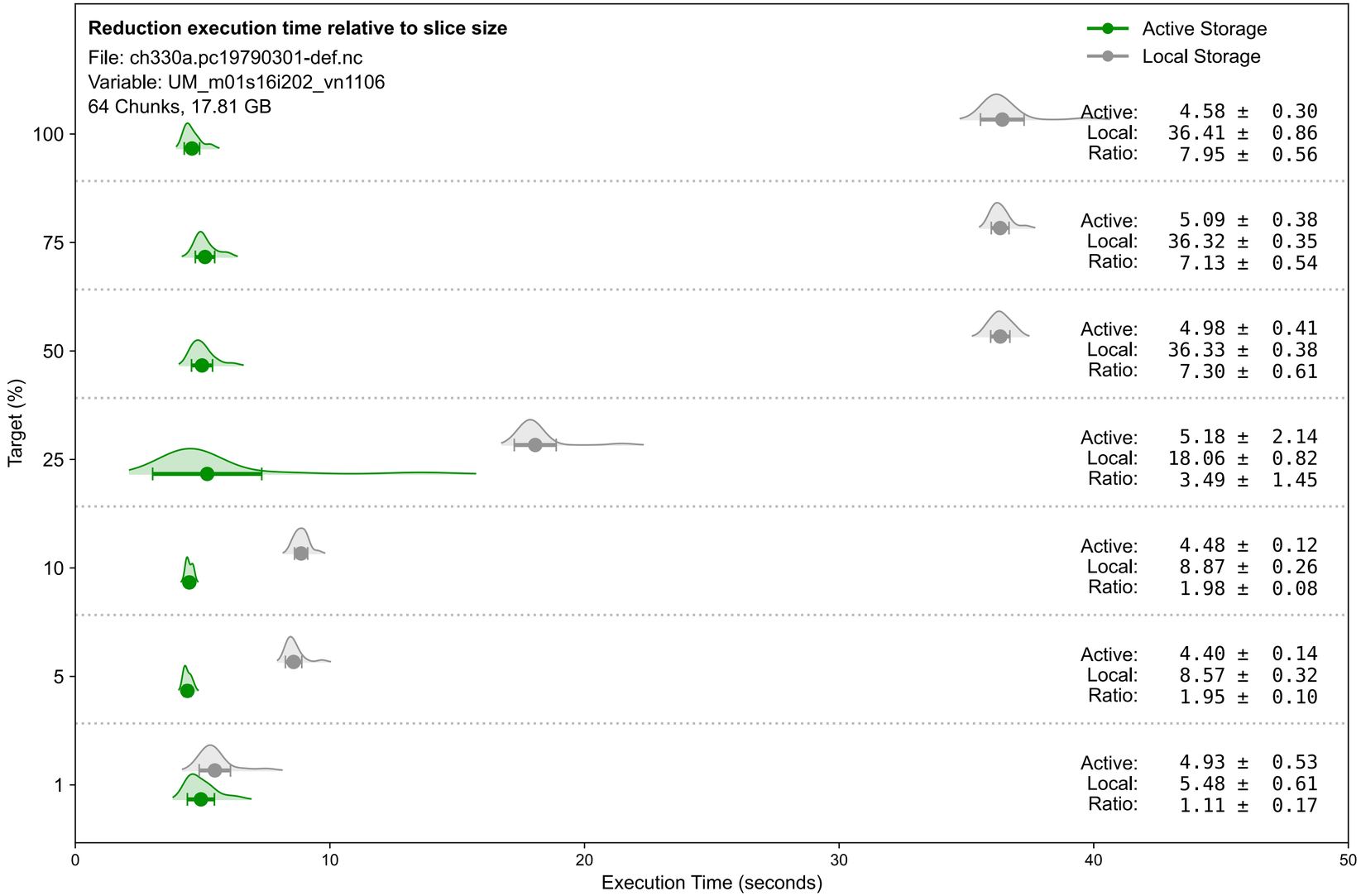
Benchmarking Performance



- Comparing different remote storage backends with and without active storage.
- Results obtained running on university network of 1Gbps (30km away from server location)

→ for both protocols, active storage provides faster execution of computes

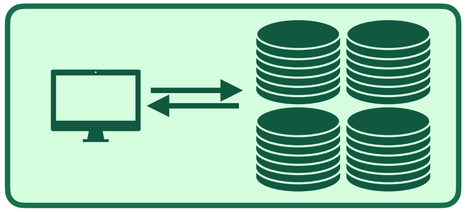
Benchmarking Performance



- Comparing execution time as the data set is sliced to a larger target size
- Impact of active storage is more pronounced

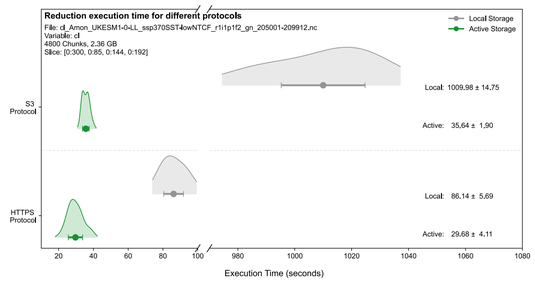
→ processing time with active storage does not scale with dataset size

Summary



Active storage provides a solution to **reducing lower network traffic** and payload, providing more **efficient workflows**

This is essential as data scales into the exabyte realm.



PyActiveStorage provides users an accessible tool to implement active storage in the workflows, **demonstrating significantly faster execution time** of data computes

Reductionist is a deployable ansible container allowing for future deployment, scaling with data storage requirements

Thank you !

References

Software tool sources:

PyActiveStorage Documentation. Available at: https://pyactivestorage.readthedocs.io/

Reductionist-rs Documentation. Available at: https://stackhpc.github.io/reductionist-rs/

PyFive Documentation. Available at: https://pyfive.readthedocs.io/

Lawrence et al., (2026). pyfive: A pure-Python HDF5 reader. Journal of Open Source Software, 11(118), 9688, https://doi.org/10.21105/joss.09688

Lawrence, B. N., Bennett, V. L., Churchill, J., Juckes, M., Kershaw, P., Pascoe, S., Pepler, S., Pritchard, M. and Stephens, A. (2013) Storing and manipulating environmental big data with JASMIN. In: IEEE Big Data, October 6-9, 2013, San Francisco.

Exascale images and sources:

Genomics Education Programme What is genomics? NHS England. Available at: https://www.genomicseducation.hee.nhs.uk/education/core-concepts/what-is-genomics/

Stephens ZD, Lee SY, Faghri F, Campbell RH, Zhai C, et al. (2015) Big Data: Astronomical or Genomical?. PLOS Biology 13(7): e1002195. https://doi.org/10.1371/journal.pbio.1002195

National Centre for Atmospheric Science Green rocky hills [Image]. Licensed under CC BY-NC-SA 2.0.

NASA, Chandra X-ray Center (CXC), Harvard-Smithsonian Center for Astrophysics (CfA), Space Telescope Science Institute (STScI), Magellan/U. Arizona and ESO WFI Bullet Cluster image [Image]. Credit: M. Markevitch and D. Clowe.

About CEDA-JASMIN

JASMIN is a data analysis facility that provides storage and computing resources for data-intensive environmental science research, serving over 1,500 users. The Centre for Environmental Data Analysis (CEDA) operates the data archive and analysis services hosted on JASMIN.

There are 92 compute nodes with 48 cores providing a total of 4416 cores. It designed, integrated and operated by the Science & Technology Facilities Council (STFC) on behalf of the Natural Environment Research Council (NERC).