Hewlett Packard
Enterprise

Arkouda and XArray

A new backend for working with Distributed Arrays

Jeremiah Corrado

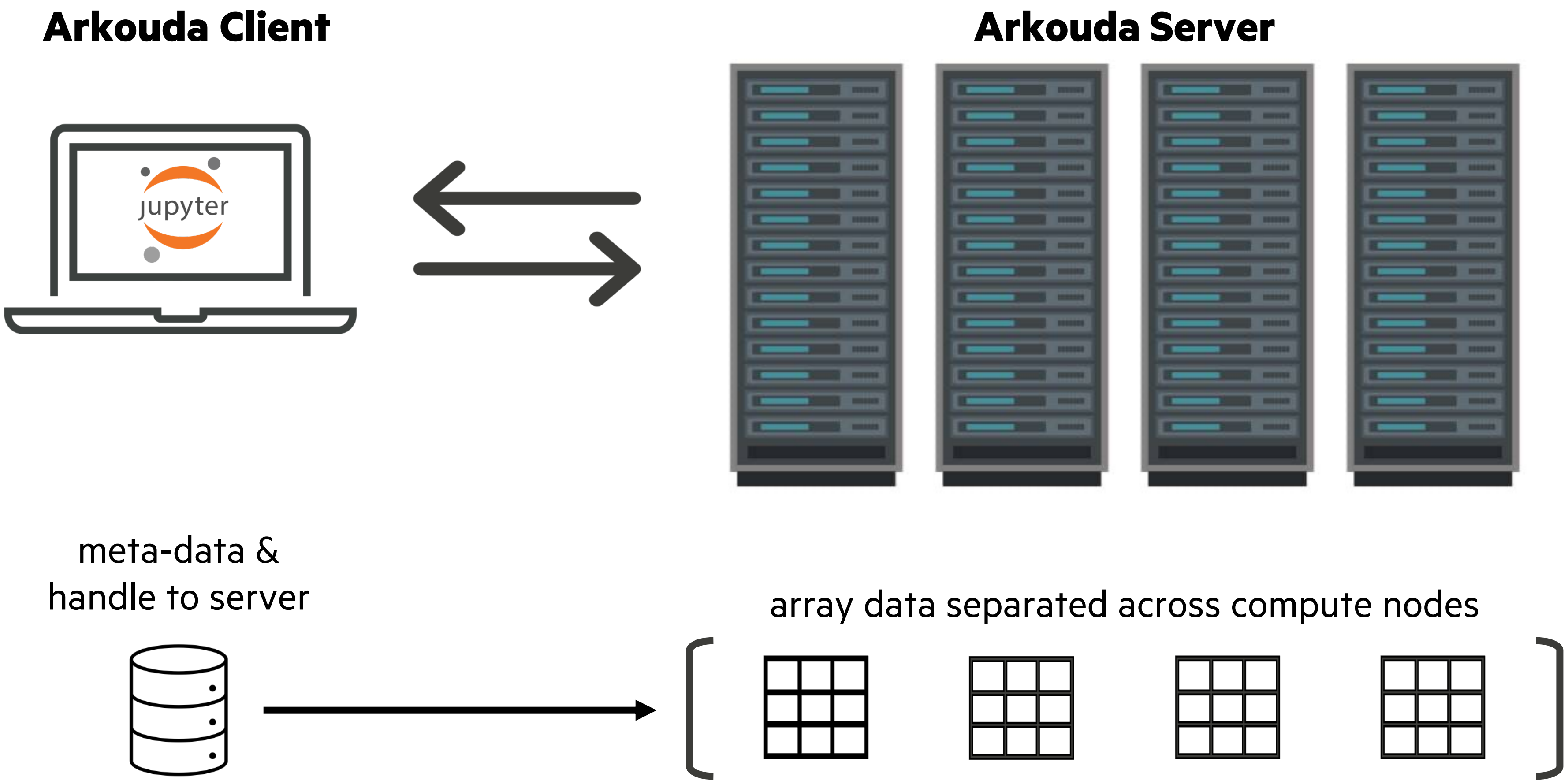
November 20, 2024

CHAPEL

αρκούδα
massive scale
data science

Background: What is Arkouda?

Arkouda is a Python library that provides massive-scale distributed arrays with numpy-like functionality



Arkouda Client

Arkouda Server

meta-data & handle to server

array data separated across compute nodes

Background: Arkouda Performance on HPC

Radix sort scaling to ~9TB/s on >8K Nodes

HPE Apollo (May 2021) ✕—✕

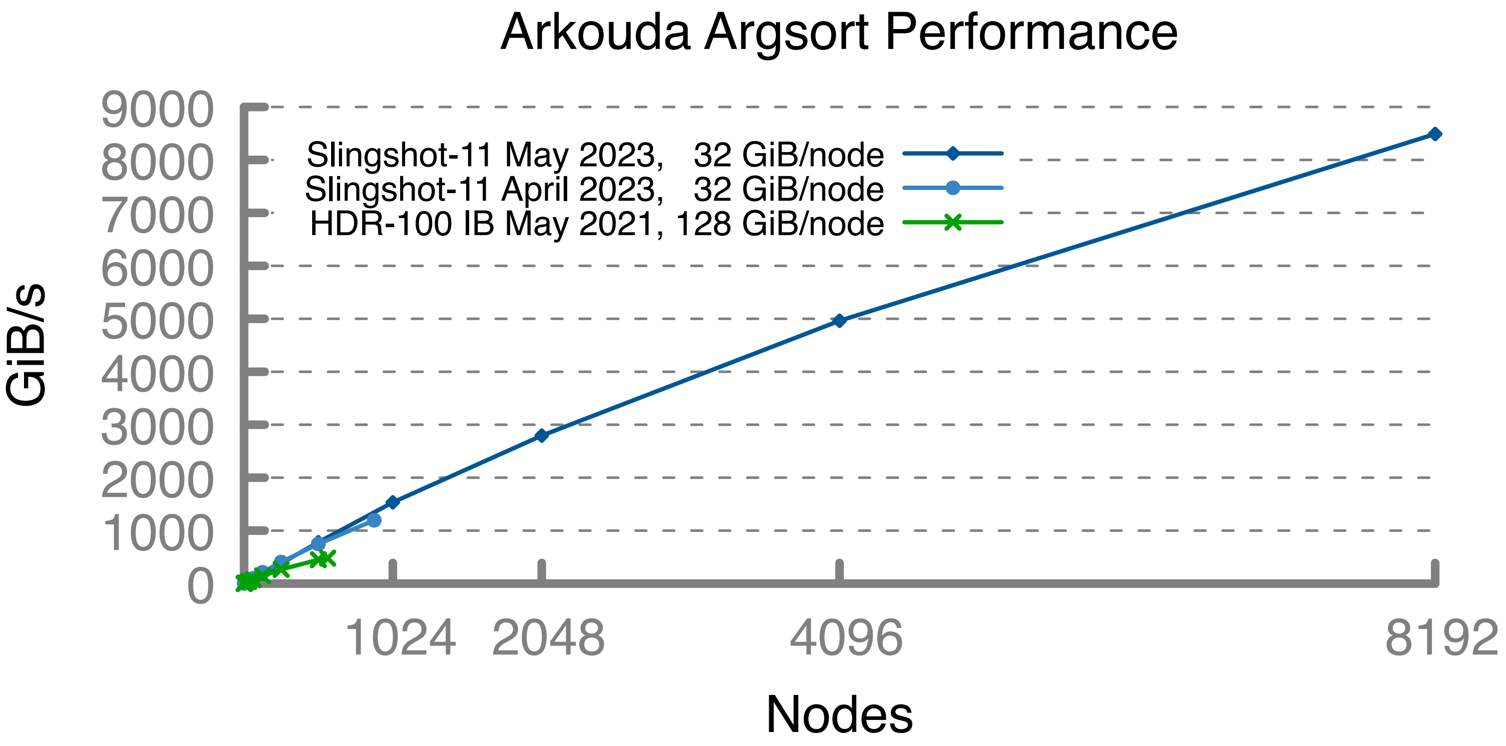
- HDR-100 Infiniband network (100 Gb/s)
- 576 compute nodes
- 72 TiB of 8-byte values
- ~480 GiB/s (~150 seconds)

HPE Cray EX (April 2023) ●—●

- Slingshot-11 network (200 Gb/s)
- 896 compute nodes
- 28 TiB of 8-byte values
- ~1200 GiB/s (~24 seconds)

HPE Cray EX (May 2023) ◆—◆

- Slingshot-11 network (200 Gb/s)
- 8192 compute nodes
- 256 TiB of 8-byte values
- ~8500 GiB/s (~31 seconds)



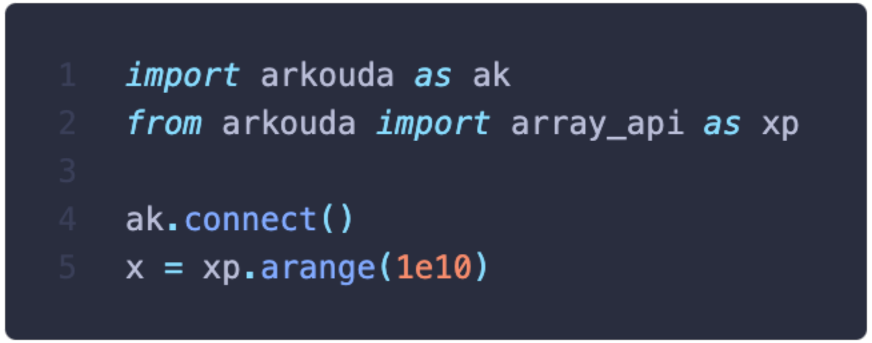
Recent Improvements to Arkouda

- Historically, Arkouda has only supported 1D arrays
  - its features were primarily focused on supporting exploratory data analysis at scale
  - Arkouda includes scalable algorithms for sorting, group-by, statistical operations, etc.
  - It also supports I/O with common data-science file formats: Parquet, HDF5, CSV

- Recent focus on making Arkouda a more general framework for any HPC-scale array computation
  - support for multidimensional arrays
  - implementation of the Array API
  - support for Zarr IO

Arkouda's Array API Implementation

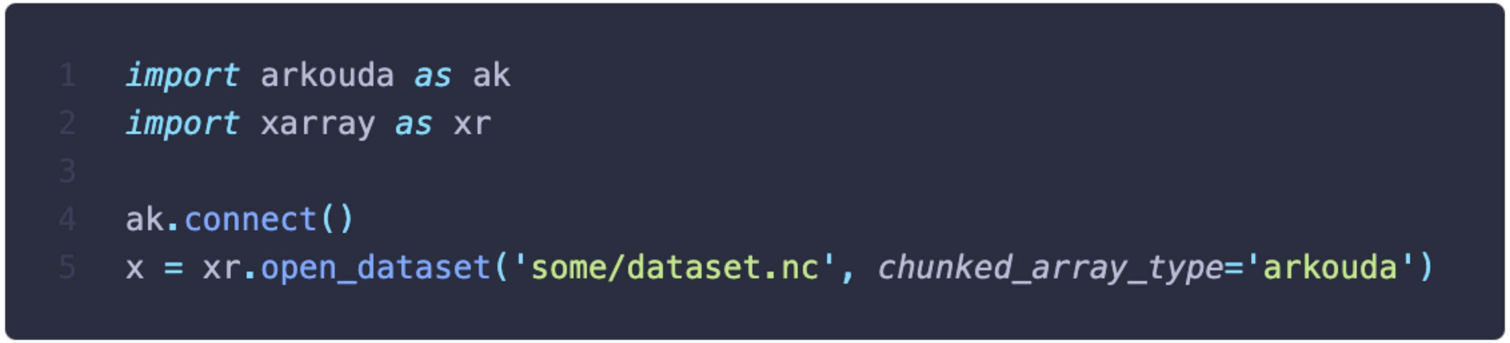
- Recent work has added an Array API module to Arkouda:



```python
import arkouda as ak
from arkouda import array_api as xp

ak.connect()
x = xp.arange(1e10)
```

- The 'array_api' module can be registered as a "chunk-manager" in XArray by installing [arkouda-xarray]()
  - Allows for creation of XArray objects backed by Arkouda arrays:

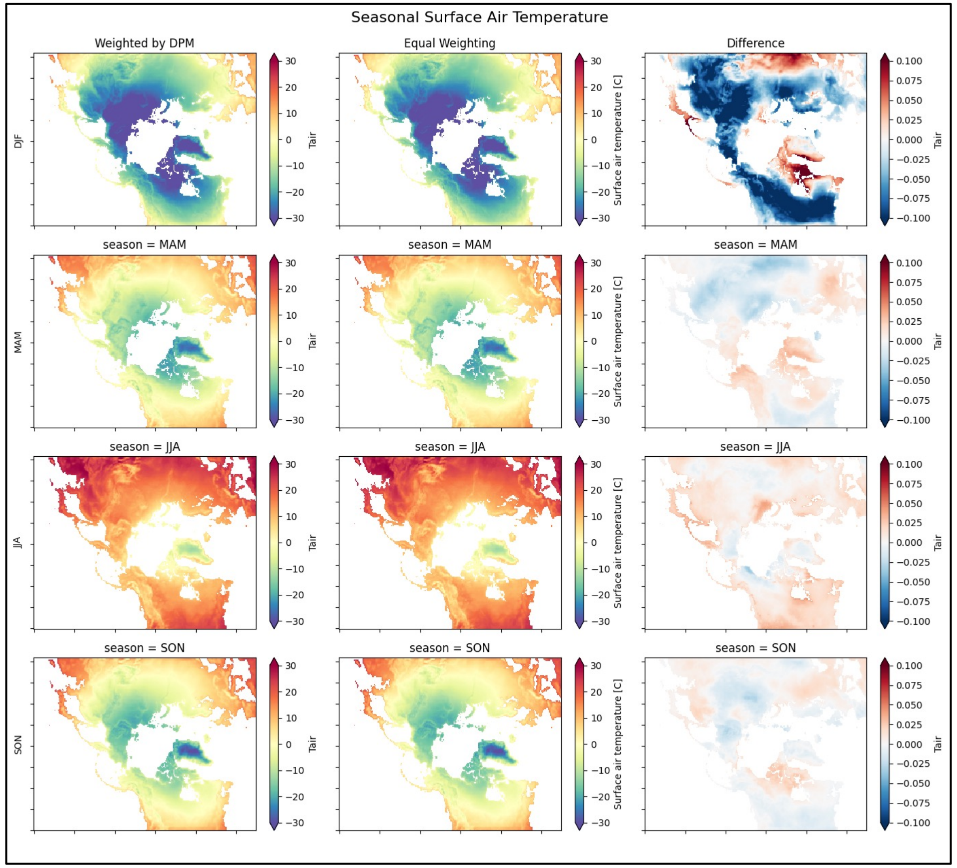


```python
import arkouda as ak
import xarray as xr

ak.connect()
x = xr.open_dataset('some/dataset.nc', chunked_array_type='arkouda')
```

Demo

Seasonal Temperature example from XArray Gallery…



https://github.com/jeremiah-corrado/arkouda-xarray-seasonal-avgs-example

How does Arkouda differ from Dask?

Eager Evaluation

- Arkouda commands are executed by the server as they are received
- No task-graph is generated, meaning calls to '.compute( )' are unnecessary

Opaque Array Chunking

- Arkouda arrays are split evenly across server's compute nodes
- Chunking is opaque to the client / doesn't need to be specified explicitly

Server Execution Model

- Server is a single cohesive program that can be launched on a laptop, cluster, or supercomputer
- No need for multiple Python interpreters running on each compute node

Future Work

- Performance improvements + bug fixes
  - plumb Arkouda's native group-by support directly into XArray (using Flox)
  - array API implementation is still nascent, bug reports would be extremely helpful

- Chapel team would be interested in hearing about workflows that could benefit from Arkouda+Xarray
  - real-world use cases could help us drive/expand performance improvements and feature expansion

Chapel Blog Article: *Doing science in Python? Wishing for more speed or scalability?*

chapel-lang.org/blog/posts/python-science-collabs/

Thank you

Chapel Team Contact for Arkouda:

Engin Kayraklioglu
engin@hpe.com

github.com/Bears-R-Us/arkouda

chapel-lang.org/