



Hewlett Packard
Enterprise



Interactive Exploratory Data Analytics (EDA) on Petabytes with Python and Arkouda, Powered by Chapel

Michelle Strout, Michael Ferguson,
Engin Kayraklioglu, Ben McDonald

November 18-21, 2024

Arkouda Enables HPC from Python

**Performant
&
Scalable**

Interactive Rates

Operations run in seconds

Massive Scales

100s of TBs using 1000s of nodes

**Use a
Supercomputer
from your
Laptop**

Python Interface for Client

Familiar, interactive, Jupyter-ready

Chapel-powered Server

Runs on supercomputer, cluster, cloud

**Modular
&
Extensible**

Modules for Different Uses

Graphs, visualizations, and more

Open-Source

Developed under MIT License

No other tool provides Exploratory Data Analysis (EDA) at these scales

Arkouda Demo

Client

- Jupyter on laptop

Server

- Running on 8 nodes of Cray XC

Dataset

- Too large for a laptop

Operations

- Histogram
- Group-by
- Visualization w/ matplotlib

The video for this slide can be viewed online at <https://youtu.be/0AIO1mhVr7M>

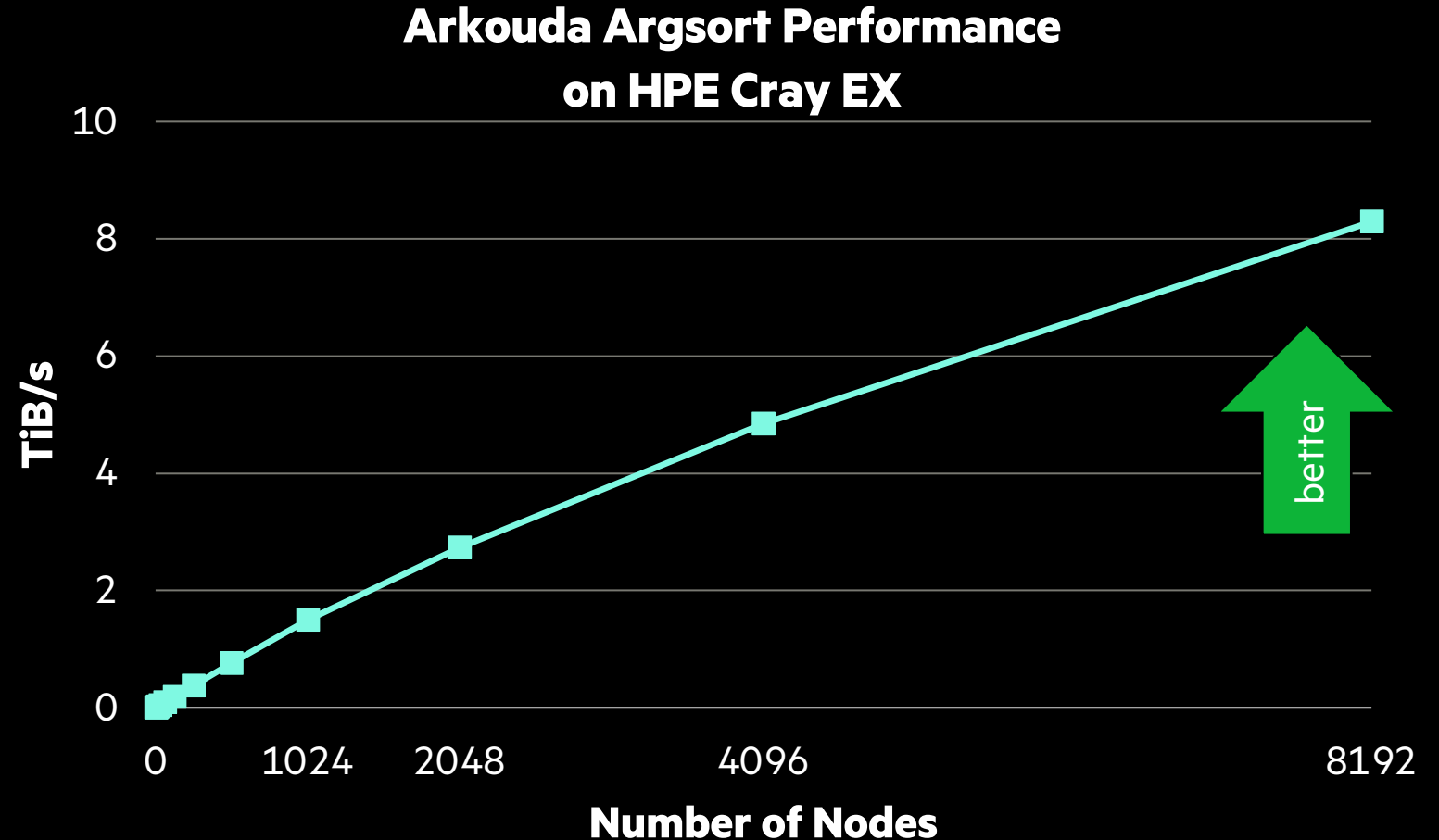
Radix Sort in Arkouda Scaling to 8 TiB/s on 8K Nodes

Slingshot-11 network (200 Gb/s)

8192 compute nodes

256 TiB of 8-byte values

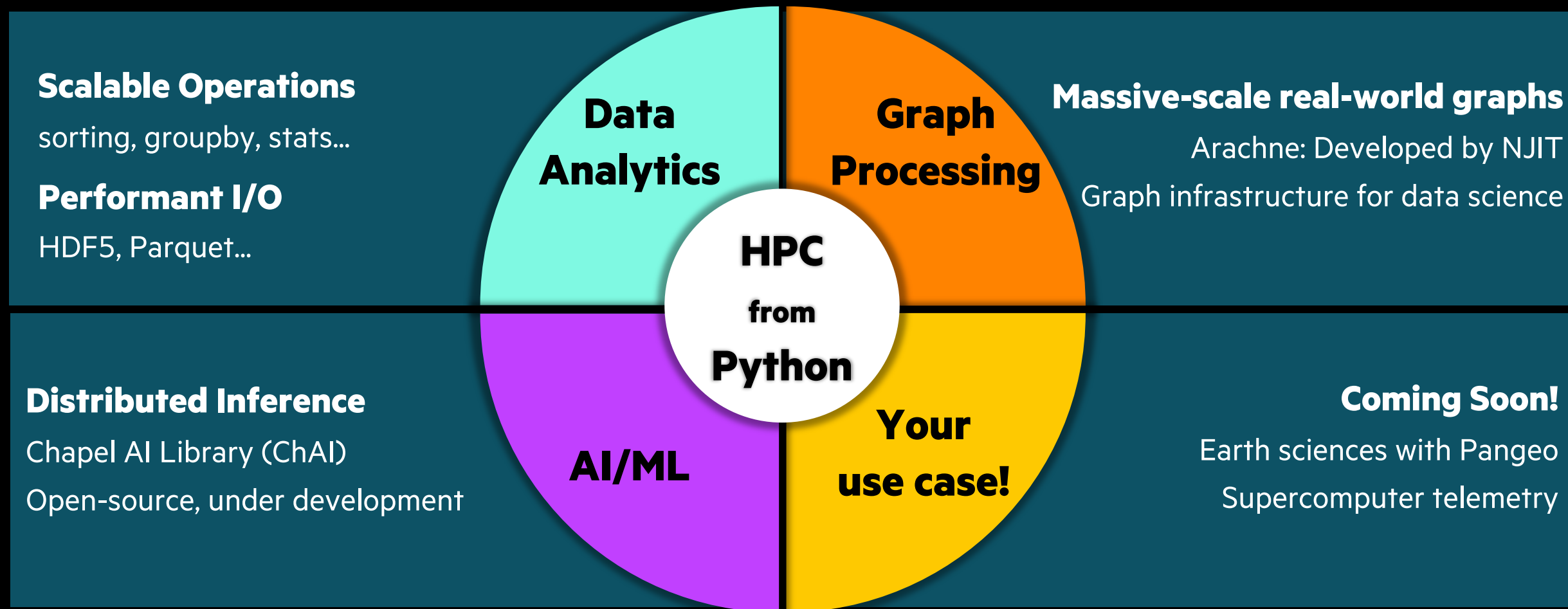
~8500 GiB/s (~31 seconds)



A notable performance achievement in ~100 lines of Chapel



What can Arkouda do?



Powered by the Chapel Parallel Programming Language



Chapel is a language designed for productive parallel programming, particularly on large-scale systems. Chapel is ...

Easy to Use

"We ask students at the master's degree to do stuff that would take 2 years and they do it in 3 months." Eric Laurendeau, Professor of Mechanical Engineering

Portable

HPE Cray EX, HPE Apollo, Cray XC, *nix systems, Mac, NVIDIA and AMD GPUs

Fast & Scalable

Achieved 8,500 GiB/s when sorting 256 TiB in 31 seconds on 8192 HPE Cray EX Nodes

GPU-Ready

Real-world applications were ported on GPUs with few changes, and run on leadership-class systems such as Frontier and Perlmutter

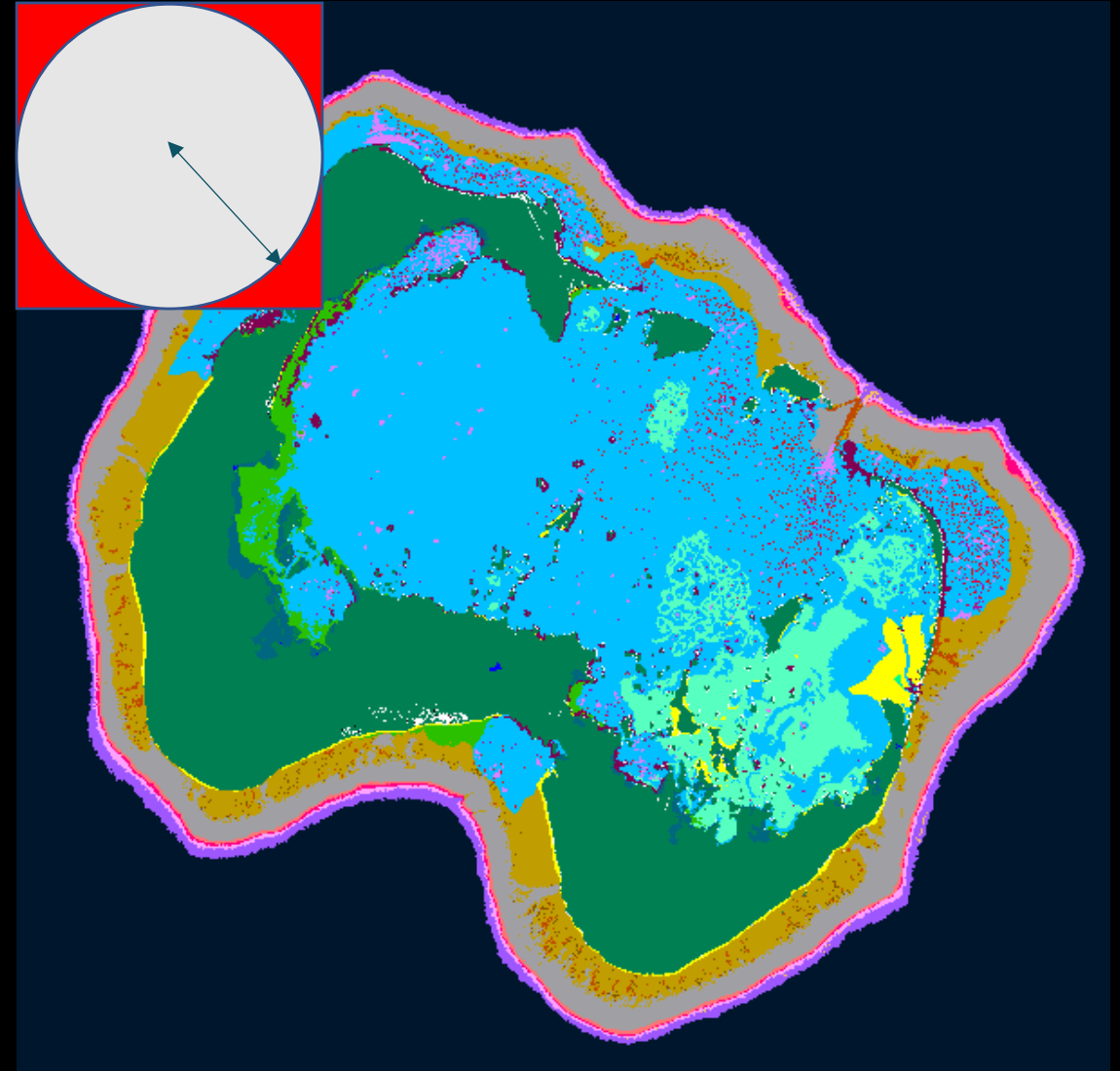
Open source

Team at HPE actively interacts with Chapel community at chapel-lang.org



Another Chapel Use Case: Image Processing for Coral Reef Biodiversity

- **Analyzing images for coral reef biodiversity**
- **Algorithm implemented productively**
 - Add up weighted values of all points in a neighborhood
 - Developed by Scott Bachman, NCAR scientist
 - Scott ...
 - started learning Chapel in Sept 2022,
 - started Coral Reef app in Dec 2022, and
 - had collaborators presenting results in Feb 2023.
 - In July, changed ~5 lines in a variant to run on GPU
- **Performance**
 - Less than 300 lines of Chapel code scales out to 100s of processors on Cheyenne (NCAR)
 - Full maps calculated in **seconds**, rather than days
 - **10,000 times faster than the MATLAB version**



Calling the Coral Reef Chapel Code from Within Python

Runs on a Laptop

- Same code ran on supercomputer

Runs Interactively

- From within Python interpreter
- Calling the fast Chapel code

Calling Chapel Code from Python

For Coral Reef Beta Diversity Analysis

The video for this slide can be viewed online at <https://youtu.be/PvGNwyYK24g>

Thank you

<https://chapel-lang.org>

