

# User Experiences Leveraging Chapel's Features for Productive Parallel Programming

Engin Kayraklioglu, HPE

HPSFCon 2026

March 18, 2026

# What is Chapel?

**Chapel is a programming language that is:**

- **parallel & scalable:**
  - users benefit from multicore parallelism, GPUs, supercomputers...
- **portable:**
  - runs on Linux, MacOS, WSL, Cloud, InfiniBand, Slingshot, ARM, GPUs, Raspberry Pi...
- **general-purpose:**
  - used in fields like aerodynamics, data and graph analytics, quantum physics, earth sciences...
- **modern:**
  - object-orientation, memory and type safety, error handling, tool/IDE support...
- **open-source:**
  - a member of the High-Performance Software Foundation & Linux Foundation...



# Meanwhile at HPSFCon 2026

Daniel Fedorin has discussed challenges for **supporting a community** for an open-source parallel programming language:

**HPE**

## **From Engineering To Community: Challenges for HPC Programming Language Adoption**

Daniel Fedorin, HPE

March 16, 2026

HPE internal use only 1



# What this talk is about is ...

what Chapel users experience, including  
**successes & pains**



# There is no "typical" Chapel User... and we take pride in it!

Our users:

- span diverse fields like earth sciences, engineering disciplines, data analytics, computer science
- use workstations, commodity clusters, leadership-class systems, CPUs, GPUs
- like Chapel for the syntax, easy parallelism & scalability, portability
- have different gripes and wishes, too!

## Read Chapel users' stories on Chapel Blog:



[chapel-lang.org/blog/series/7-questions-for-chapel-users/](https://chapel-lang.org/blog/series/7-questions-for-chapel-users/)



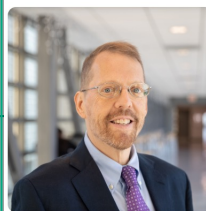
**7 Questions for Éric Laurendeau: Computing Aircraft Aerodynamics in Chapel**



**7 Questions for Nelson Luís Dias: Atmospheric Turbulence in Chapel**



**7 Questions for Scott Bachman: Analyzing Coral Reefs with Chapel**



**7 Questions for David Bader: Graph Analytics at Scale with Arkouda and Chapel**

Posted on November 6, 2024.

Tags: [User Experiences](#) [Interviews](#) [Graph Analytics](#) [Arkouda](#)

By: [Engin Kayraklioglu](#), [Brad Chamberlain](#)



**7 Questions for Bill Reus: Interactive Supercomputing with Chapel for Cybersecurity**



**7 Questions for Tiago Carneiro and Guillaume Helbecque: Combinatorial Optimization in Chapel**



**7 Questions for Marjan Asgari: Optimizing Hydrological Models with Chapel**

Posted on September 15, 2025.



**7 Questions for Oliver Alvarado Rodriguez: Exploiting Chapel's Distributed Arrays for Graph Analysis through Arachne**

Posted on January 21, 2026.

Tags: [User Experiences](#) [Interviews](#) [Graph Analytics](#) [Arkouda](#)

[Sparse Arrays](#)

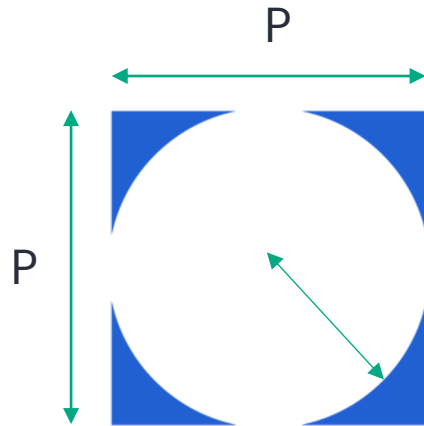
By: [Engin Kayraklioglu](#), [Brad Chamberlain](#)

# Understanding Biodiversity: Going from CPU to Frontier



# Coral Reef Biodiversity Detection w/ Spectral Image Analysis

1. Read in a  $(M \times N)$  raster image of habitat data
2. Create a  $(P \times P)$  mask to find all points within a given radius.
3. Convolve this mask over the entire domain and perform a weighted reduce at each location.

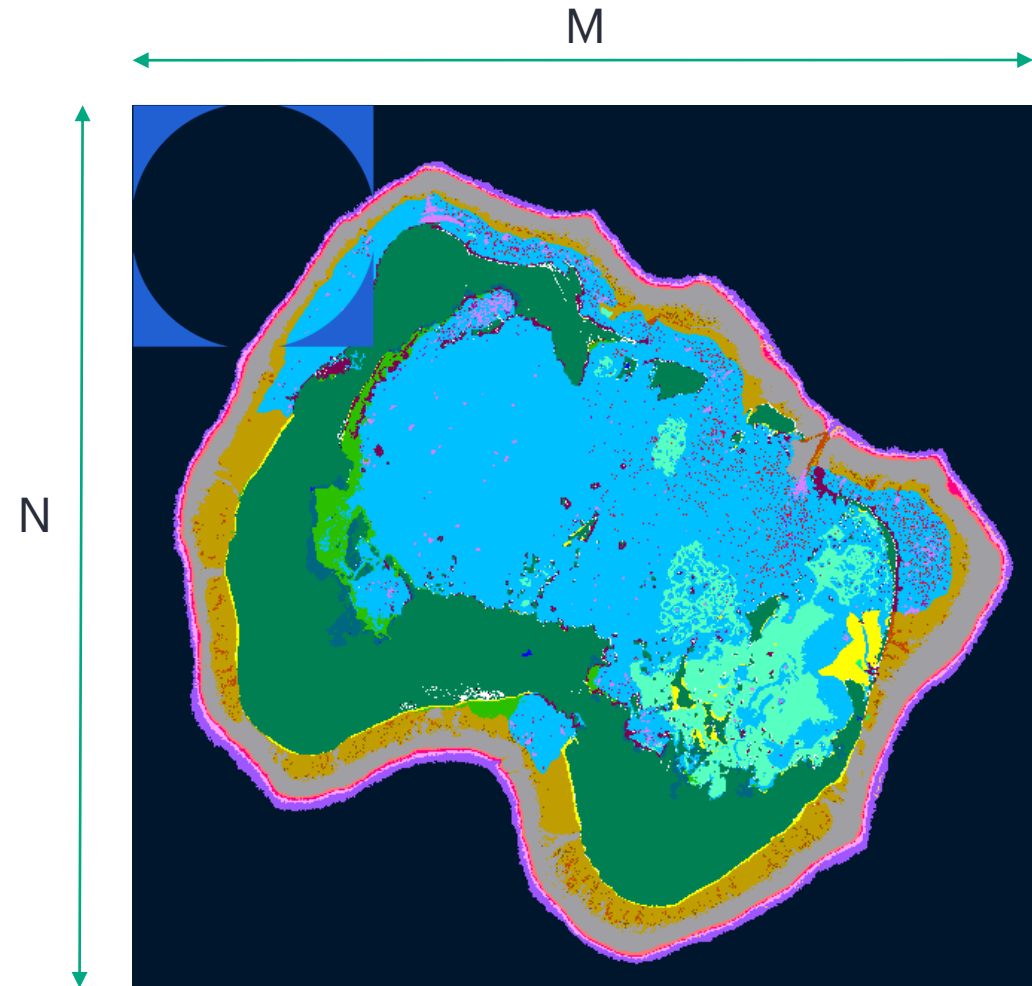


Algorithmic complexity:  $O(MNP^3)$

Typically:

-  $M, N > 10,000$

-  $P \sim 400$



# Coral Reef Biodiversity Detection w/ Spectral Image Analysis

## Background

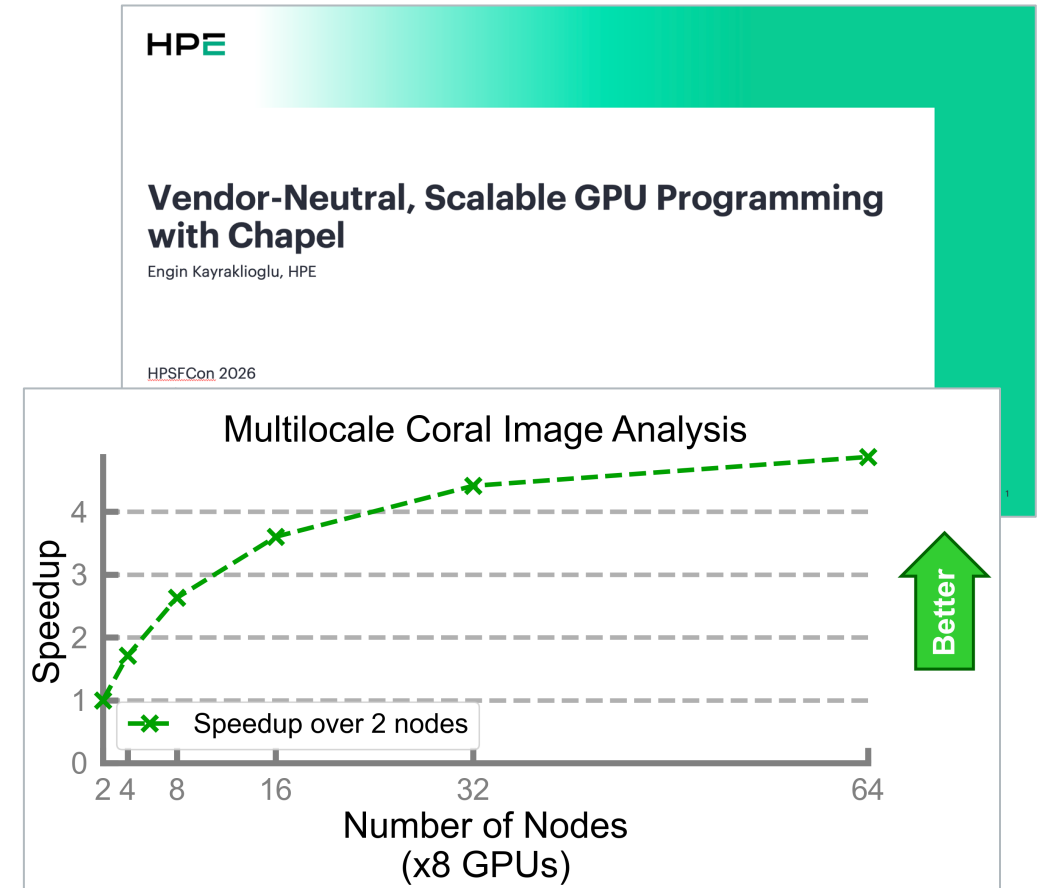
- Teamed up w/ Scott Bachman (NCAR, then), who implemented a CPU-only algorithm
  - For the problem sizes he cared about, we estimated the execution would take **27 days**
  - But he had access to some GPU nodes
- This was during the early days of Chapel's native GPU support

## The Effort

- We ported to Chapel's then-prototypical native GPU support
  - Lines of code went from ~250 to ~300, a very modest change
  - We didn't use any new language construct that he hadn't used
- Straightforward code changes:
  - from sequential Chapel code
  - to GPU-enabled one
  - to multi-node, multi-GPU, multi-thread

## Results

- On 2 nodes, we ran the realistic problem size in **87 minutes**
  - On 64 nodes, **17 minutes**
- 5x improvement going from 2 to 64 nodes
  - (from 16 to 512 GPUs)



# Scott Bachman's Interview

“ *With the coral reef program, I was able to speed it up by a factor of, like 10,000. I would say some of that was algorithmic... but again, Chapel had the features in the language that allowed me to do it pretty succinctly.* ”



## Chapel Language Blog

About Chapel Website Featured Series Tags Authors All Posts

### 7 Questions for Scott Bachman: Analyzing Coral Reefs with Chapel

Posted on October 1, 2024.



Tags: Earth Sciences Image Analysis GPUs

User Experiences Interviews

By: [Brad Chamberlain](#), [Engin Kayraklioglu](#)

Part of a series: [7 Questions for Chapel Users](#)

<https://chapel-lang.org/blog/posts/7qs-bachman/>

## Q: If you could improve Chapel with a finger snap, what would you do?

- Single data-parallel loop mapping to multiple GPUs
  - Dev's note: This requires extending Chapel's distributed array support to GPUs
- Better support for installing on clusters
  - Dev's note: We expanded the formats in which we release versions
    - Linux packages, and a Spack package!



World-class CFD Solver:  
>150k Lines of Chapel



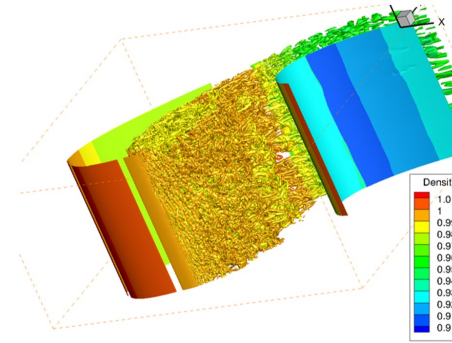
# Overcoming the Inertia

- In 2019, a group of grad students from Polytechnique Montreal reaches out to Brad Chamberlain
  - They say they want to write a CFD solver in Chapel for aerodynamics
  - Their homegrown CFD solver is in C+OpenMP, and getting unwieldy to maintain and expand
  - They also want to run multinode simulations for larger problems
  - Similarly, they want to go from 2D to 3D, from structured to unstructured meshes
- Meanwhile, their PI, Eric Laurendeau is hesitant
  - Rewriting in C++, MPI, and OpenMP feels safer, as Chapel is an unknown technology compared to those
- The grad students, Simon, Matthieu, and Frederick, write a prototype in Chapel anyway
  - They do it surprisingly quickly, achieve fidelity, and can run on multinode
- That convinces Eric to have the team rewrite their code in Chapel
- Turns out, Eric had also pulled a very similar "trick" to switch his institution from Fortran to C in the past!

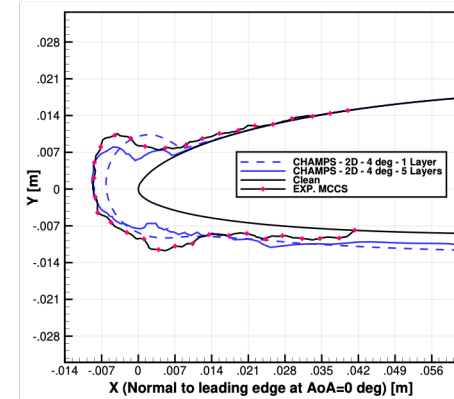


# CHAMPS

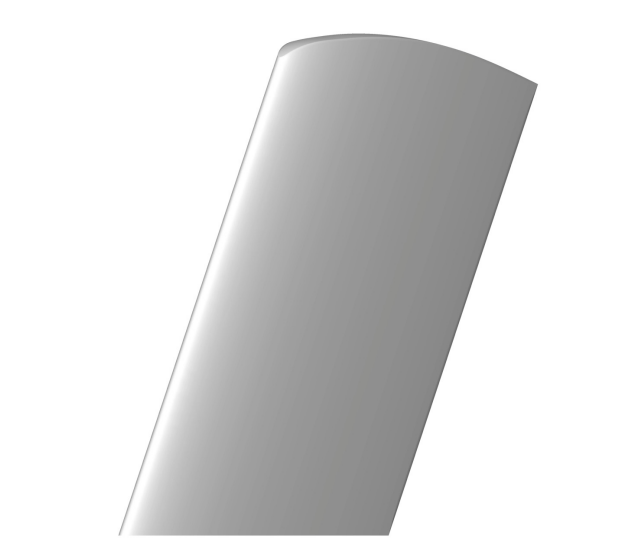
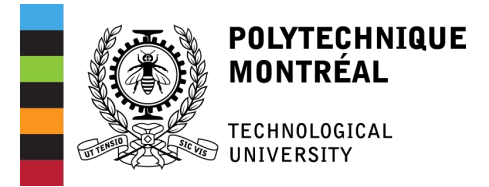
- **CHA**pel **Mu**lti**Ph**ysics **S**imulation
- Designed for aerodynamic design
- Can simulate multiple physical phenomena
  - Flow, droplets, icing, turbulence...
  
- **> 150,000 lines** of Chapel code
- Developed and maintained by mechanical engineering grad students, postdocs
  - 22 contributors to the code so far
  
- Achieves similar fidelity to commercially developed software



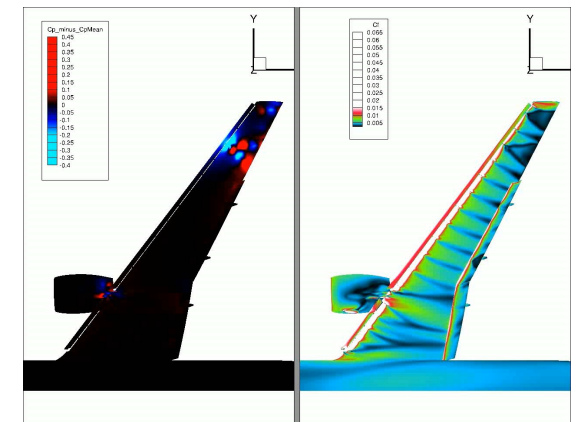
**High-fidelity  
Multiphysics Simulation**



**Ice Accretion Cross-section**



**Ice Accretion Over Time**



**High-lift Simulation**

# More Information on CHAMPS

“ CHAMPS thus stands on par with similar software developed by national research centers such as NASA (USA), ONERA (France), DLR (Germany), JAXA (Japan), etc., as well as commercial vendors (Ansys, STAR-CCM+, etc.). Of these, it is the only one not using OpenMP/MPI protocols...”

## Q: If you could improve Chapel with a finger snap, what would you do?

- Profilers, faster compilation, better error messages, package manager

## Chapel Language Blog

About Chapel Website Featured Series Tags Authors All Posts



### 7 Questions for Éric Laurendeau: Computing Aircraft Aerodynamics in Chapel

Posted on September 17, 2024.

Tags: Computational Fluid Dynamics User Experiences Interviews

By: Engin Kayraklioglu, Brad Chamberlain

Part of a series: [7 Questions for Chapel Users](#)

[Eric's Interview](#)

**Sequel with his current students is coming soon!**



**Hewlett Packard Enterprise** **POLYTECHNIQUE MONTRÉAL** **TECHNOLOGICAL UNIVERSITY**

### High-Performance, Productive Programming using Chapel with Examples from the CFD Solver CHAMPS

Engin Kayraklioglu, Hewlett Packard Enterprise  
Éric Laurendeau, Polytechnique Montréal  
Karim Zayni, Polytechnique Montréal

Advanced Modeling & Simulation (AMS) Seminar Series  
NASA Ames Research Center, February 20th, 2025

For more technical content, check out our [joint talk at NASA Ames](#)

# Greenhouse Gases in the Amazon: From Classic Fortran to GPUs



# Analyzing Greenhouse Gases in the Amazon



There is a tower here

Amazon Tall Tower Observatory (ATTO)  
image credit Sebastian Brill



# Syntax Matters

- A civil engineering group picked up Chapel, primarily because they liked the syntax
  - Expressive array operations, domains (i.e. index sets) as first-class citizens in the language
  - Meanwhile, they were not really happy with Python's performance
    - Fortran, Pascal, and C were other languages being used within the group
    - **Had never done any parallel programming before**
- Then, they realized that they could **write parallel code** as easily as serial code in Chapel
  - Which inevitably resulted in significantly better performance
- They now write everything from simpler scripts to numerical analysis code in Chapel
- Recently they acquired their first **GPU workstation** and use it exclusively with Chapel



# More on Nelson Dias' Research and Experiences



“ *Chapel allows me to use the available CPU and GPU power efficiently, and reduces the time involved in most analyses significantly. This can be done without low-level programming of message passing, data synchronization, managing threads, etc.* ”



## Chapel Language Blog

About Chapel Website Featured Series Tags Authors All Posts

### 7 Questions for Nelson Luís Dias: Atmospheric Turbulence in Chapel

Posted on October 15, 2024.

Tags: [User Experiences](#) [Interviews](#) [Data Analysis](#)

[Earth Sciences](#) [Computational Fluid Dynamics](#)

By: [Engin Kayraklioglu](#), [Brad Chamberlain](#)

Part of a series: [7 Questions for Chapel Users](#)



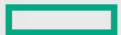
## Q: If you could improve Chapel with a finger snap, what would you do?

- Smaller executables
  - Dev's note: We have never looked at reducing our executable size, to my knowledge
- First-class subroutines that can accept a generic array argument
  - Dev's note: First-class subroutines are still relatively new in the language

## Nelson's CHIOW 2022 Talk *From C and Python to Chapel as My Main Programming Language*



# Juggling Petabytes of Data: Python Powered by Chapel



# Data Science In Python at scale?

**Motivation:** Imagine you work with...

- ...Python-based data scientists
- ...HPC-scale data science problems to solve
- ...access to HPC systems



How will you leverage your Python programmers to get your work done?



# What is Arkouda?

Q: "What is Arkouda?"



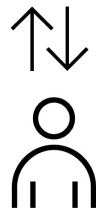
**Arkouda Client**  
(written in Python)

```
jupyter big_add_sum Last Checkpoint: 16 minutes ago (auto saved)
File Edit View Insert Cell Kernel Help Python 3.0
In [1]: import arkouda as ak
In [2]: ak.V = False
         ak.setLogServer("localhost", port=5555)
         4.2.5
         pip = topi://localhost:5555
In [3]: ak.V = False
         N = 1024 * 1024 * 1024 * 1024 * 8 == 8388608 * 2**23 * 8 == 256812
         A = ak.arange(0, N, 1)
         B = ak.arange(1, N, 1)
         C = A+B
         print(ak.info(C, C))
         name: "4_3" dtype: "uint64" size: 10000000 ndim: 1 shape: (10000000,) itemsize: 8
         (0, 2, 4, ..., 9999999) 10000000 10000000 10000000
In [4]: B = (B*B**1)/2
         print(B)
         print(ak.sum(C))
         99999990000000.0
         99999990000000.0
In [5]: ak.shutdown()
```

```
import arkouda as ak

def ak_argSORT(N, seed):
    a = ak.randint(0, 2**64, N, dtype=ak.uint64, seed=seed)
    perm = ak.argsort(a)

    assert ak.is_sorted(a[perm])
```



**User writes Python code**  
**making familiar NumPy/Pandas calls**



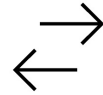
# What is Arkouda?

**Q:** “What is Arkouda?”



**Arkouda Client**  
(written in Python)

```
jupyter big_add_sum Last Checkpoint: 16 minutes ago (autosaved)
File Edit View Insert Cell Kernel Windows Help Python 3.0
In [1]: import arkouda as ak
In [2]: ak.N = False
         ak.setLogPrefix("localhost", port=5555)
         4.2.5
         pip = topi://localhost:5555
In [3]: ak.N = False
         N = 1024 # 2**10 = 1024 * 8 == 8192 # 2**13 = 8 == 256KB
         A = ak.arange(0, N, 1)
         B = ak.arange(1, N, 1)
         C = A+B
         print(ak.info(C, C))
         name="4_3" dtype="uint64" size=1000000 ndim=1 shape=(1000000) itemsize=8
         [0 2 4 ... 1999996 1999998 1999999]
In [4]: B = (B*(B-1))
         print(B)
         9999990000000.0
         9999990000000.0
In [5]: ak.shutdown()
```



**Arkouda Server**  
(written in Chapel)



**User writes Python code**  
**making familiar NumPy/Pandas calls**

```
import arkouda as ak

def ak_argsort(N, seed):
    a = ak.randint(0, 2**64, N, dtype=ak.uint64, seed=seed)
    perm = ak.argsort(a)

    assert ak.is_sorted(a[perm])
```

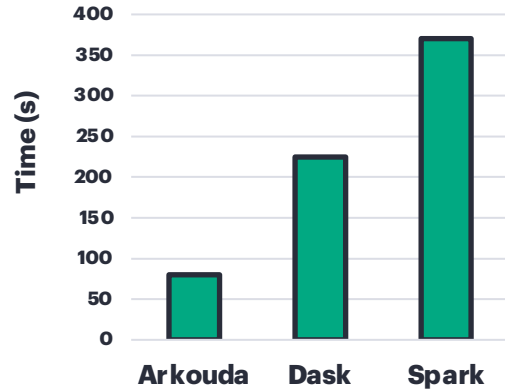
**A1:** “A scalable version of NumPy / Pandas routines for data scientists”

**A2:** “A framework for driving supercomputers interactively from Python”



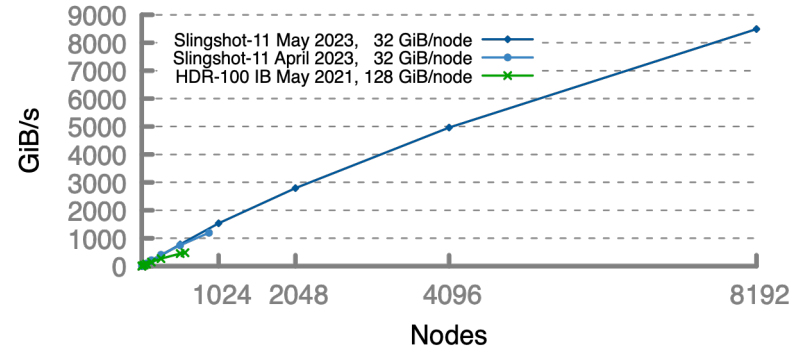
# More on Arkouda

**Execution Time**  
(on 16 HPE Cray EX Nodes)



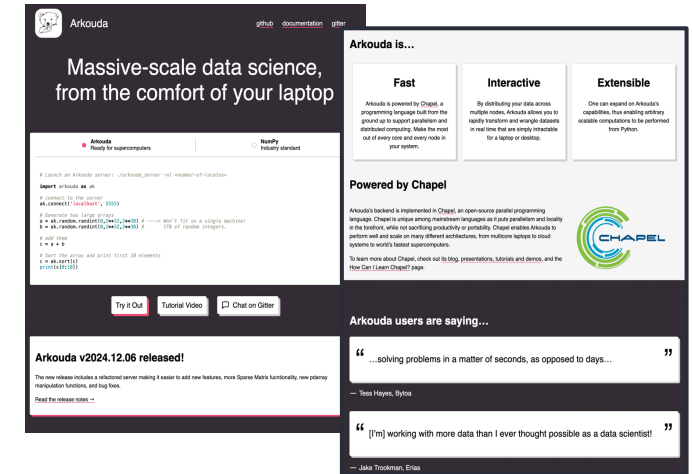
**Arkouda outperforming Dask and Spark in a Server Telemetry Analysis Workflow**

**Arkouda Argsort Performance**



**Arkouda sorting 256 TiB of data in ~30 seconds using over a million cores**

**Website:** <https://arkouda-www.github.io/>



## Chapel Language Blog

About Chapel Website Featured Series Tags Authors All Posts



### 7 Questions for Bill Reus: Interactive Supercomputing with Chapel for Cybersecurity

Posted on February 12, 2025.

Tags: [User Experiences](#) [Interviews](#) [Data Analysis](#) [Arkouda](#)

By: [Engin Kayraklioglu](#), [Brad Chamberlain](#)

Part of a series: [7 Questions for Chapel Users](#)

*"It typically took a skilled programmer with no background in Chapel mere weeks or months to begin contributing performant, scalable code."*



<https://chapel-lang.org/blog/posts/7qs-reus/>

# Closing Thought


## What do your users think about the technology you are working on?




### Read Chapel users' stories on Chapel Blog:

 [chapel-lang.org/blog/series/7-questions-for-chapel-users/](https://chapel-lang.org/blog/series/7-questions-for-chapel-users/)


**Contact Info**  
**Me:** [linkedin.com/in/engink](https://linkedin.com/in/engink)  
**Chapel:** [chapel-lang.org](https://chapel-lang.org)




**7 Questions for Éric Laurendeau: Computing Aircraft Aerodynamics in Chapel**



**7 Questions for Nelson Luís Dias: Atmospheric Turbulence in Chapel**




**7 Questions for Scott Bachman: Analyzing Coral Reefs with Chapel**



**7 Questions for David Bader: Graph Analytics at Scale with Arkouda and Chapel**  
Posted on November 6, 2024.  
Tags: [User Experiences](#) [Interviews](#) [Graph Analytics](#) [Arkouda](#)  
By: [Engin Kayraklioglu](#), [Brad Chamberlain](#)




**7 Questions for Bill Reus: Interactive Supercomputing with Chapel for Cybersecurity**



**7 Questions for Tiago Carneiro and Guillaume Helbecque: Combinatorial Optimization in Chapel**



**7 Questions for Marjan Asgari: Optimizing Hydrological Models with Chapel**  
Posted on September 15, 2025.



**7 Questions for Oliver Alvarado Rodriguez: Exploiting Chapel's Distributed Arrays for Graph Analysis through Arachne**  
Posted on January 21, 2026.  
Tags: [User Experiences](#) [Interviews](#) [Graph Analytics](#) [Arkouda](#) [Sparse Arrays](#)  
By: [Engin Kayraklioglu](#), [Brad Chamberlain](#)

# Thank You

