Hewlett Packard Enterprise

Consider an Applications-First Approach for PDC

Michelle Mills Strout (Presented by Engin Kayraklioglu)

EduHPC Lightning Talk at SC24
November 17, 2024

Problem

We aren't educating enough HPC users and developers

Are PDC course prerequisites really needed?

- Do students really need to know OS concepts?
- How about parallel architecture?

How much do students learn by writing parallel programs from scratch?

- Getting to the point where parallelism can be observed can take too long
- The process can be unnecessarily discouraging

Applications-First Approach

Introduce students to PDC concepts through lens of practical applications

Can we use/develop parallel applications on day 1?

- See the performance and scaling advantages they provide
- Observe those advantages on applications that students care/know about

Can we adjust key outcomes of PDC courses?

- How to extend and maintain existing parallel codes
  - including performance and scalability analysis
- How to port their Python/R/MATLAB etc codes for HPC

Real-World Experiences

CHAMPS: A 3D, Unstructured-Grid CFD Solver for Aerodynamics: >100k lines of Chapel code

- Developed at Polytechnique Montreal, led by Eric Laurendeau

We ask students ... to do stuff that would take 2 years, and they do it in 3 months. ... So, if you want to take a summer internship and you say: 'program a new turbulence model', well they manage.

Eric's CHIUW '21 Keynote:

youtube.com/watch?v=wD-a_KyB8aI

Chapel enables undergraduate students to contribute to CHAMPS' development, something almost impossible to think of when using very complex software.

Eric's Interview on Chapel Blog:

chapel-lang.org/blog/posts/7qs-laurendeau/

Eric's Distinguished Talk at PAW-ATM

A Case Study for using Chapel within the Global Aerospace Industry

Sunday, 2:00 PM, B306

Real-World Experiences

Arachne: Graph analytics extension for Arkouda

- Arkouda: a Python frontend for HPC
- Arachne is developed by David Bader's group at NJIT

> *Students in my group, without prior experience in parallel programming, were able to write and implement scalable graph algorithms in Chapel within just a few weeks.*
>
> **David's Interview on Chapel Blog:**
>
> chapel-lang.org/blog/posts/7qs-bader/

RapidQ: Satellite image analysis for coral reefs

- 100s of lines of code, developed by Scott Bachman

> *I was able to speed it up by a factor of 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.*
>
> **Scott's Int apel Blog:**
>
> chapel-lang.org/blog/posts/7qs-bachman/

Closing Thoughts

5. What would an ”Applications variant” of student cluster competitions specifically for non-CS students look like?

Meet us at the HPE Booth (2219)

Join us at CHUG
Chapel Users Group

Happy hour, Sunday 6:30PM
Der Biergarten

Chapel Educator Meetup
2nd Wednesday of Every Month
Noon, PT

Check out the Community Calendar in the Events page

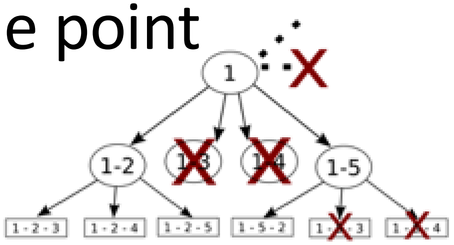
chapel-lang.org/events.html

Thank you

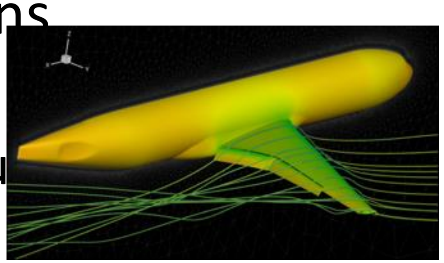
https://chapel-lang.org
@ChapelLanguage

Problem and Proposed Solution of an Applications-First Approach

- **Problem**: We aren’t educating enough HPC users and developers
  - Even for CS majors, PDC courses can be daunting
  - PDC courses have a lot of prerequisites
  - Current focus on building parallel programs from scratch misses the point



- **Proposed Solution**: Applications-First Approach
  - Introduce students to PDC concepts through lens of practical applications
    - –See the performance and scaling advantages they provide
    - –Observe those advantages being applied to problems the students care abou



  - PDC courses should be teaching students…
    - –How to extend and maintain existing parallel codes, including performance and scaling analysis
    - –How to port or user parallel libraries to leverage HPC for Python codes

Chapel Exemplifies an Applications-First Approach

- Eric Laurendeau (PI) gave our CHIUW 2021 keynote
  - title: *HPC Lessons From 30 Years of Practice in CFD Towards Aircraft Design and Analysis*
  - quote:

    "So CHAMPS, that's the new solver that has been made, and all made by the students... So, [Chapel] promotes the programming efficiency. It was easy for them to learn. ...I see the end result. We ask students at the master's degree to do stuff that would take 2 years and they do it in 3 months. And I'm not joking, this is from 2 years to 3 months. So if you want to take a summer internship and you say 'program a new turbulance model', well they manage. And before, it was impossible to do."

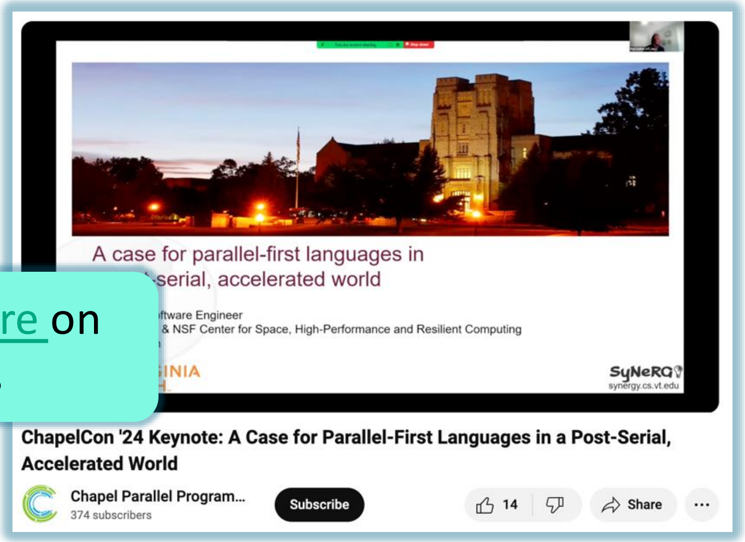
- HPC access from within Python with Arkouda, https://arkouda-www.github.io/

- Monthly meetings with educators to incorporate real-world Chapel applications and examples into teaching materials

Chapel's Approach for Applications-First Approach

- Chapel is an open-source, parallel programming language, where parallelism concepts are baked in
  - Higher-level than C, C++ and Fortran which dominate HPC programming, but more difficult than Python et al.

Watch the ChapelCon '24 Keynote by Paul Sathre on the importance of Parallel-First Languages


A case for parallel-first languages in
serial, accelerated world
ftware Engineer
& NSF Center for Space, High-Performance and Resilient Computing
INIA
SyNeRG
synergy.cs.vt.edu
Chapel Parallel Program...
374 subscribers
Subscribe
14
Share
...

ChapelCon '24 Keynote: A Case for Parallel-First Languages in a Post-Serial, Accelerated World

- Prof Eric Laurendeau is the PI of the team that develops CHAMPS