Chapel
Open-Source Productive Parallel Programming at Scale

Ben Albrecht and Brad Chamberlain
Chapel Team, Cray Inc.
OpenSuCo 2017, SC17, November 12, 2017
What is Chapel?

Chapel: A productive parallel programming language

- portable
- open-source
- a collaborative effort

Goals:

- Support general parallel programming
  - “any parallel algorithm on any parallel hardware”
- Make parallel programming at scale far more productive
Chapel and Productivity

Chapel strives to be…

…as programmable as Python

…as fast as Fortran

…as scalable as MPI (or SHMEM or UPC)

…as portable as C

…as flexible as C++

…as fun as [your favorite programming language]
The Chapel Team at Cray (May 2017)

14 full-time employees + 2 summer interns
Chapel Community R&D Partners

(and several others…)

https://chapel-lang.org/collaborations.html
CLBG Language Cross-Language Summary
(Oct 2017 standings, zoomed in)
CLBG Language Cross-Language Summary
(Oct 2017 standings, zoomed in)
CLBG Language Cross-Language Summary
(Oct 2017 standings)

Compressed Code Size (normalized to smallest entry)

Execution Time (normalized to fastest entry)

smaller

faster

Copyright 2017 Cray Inc.
Chapel Performance: Competitive for HPC

**LCALS:** Chapel vs. C + OpenMP
Shared memory performance competitive with hand-coded

**HPCC RA:**
Performance of RA (atomics)

**STREAM Triad:**
Chapel vs. MPI+OpenMP

**ISx Peformance:** Chapel vs. MPI, SHMEM

**PRK Stencil:**
Stencil PRK (weak scaling)

Chapelvs. C + OpenMP
Shared memory performance competitive with hand-coded

Chapel vs. MPI+OpenMP
Performance of STREAM

Chapel vs. MPI, SHMEM
ISx weakISO Total Time

Stencil PRK (weak scaling)
Nightly performance graphs online at: https://chapel-lang.org/perf

Copyright 2017 Cray Inc.
“My opinion as an outsider...is that Chapel is important, Chapel is mature, and Chapel is just getting started. “If the scientific community is going to have frameworks for solving scientific problems that are actually designed for our problems, they’re going to come from a project like Chapel. “And the thing about Chapel is that the set of all things that are ‘projects like Chapel’ is ‘Chapel.’”

–Jonathan Dursi

Chapel’s Home in the New Landscape of Scientific Frameworks (and what it can learn from the neighbours)
CHIUW 2017 keynote

Demo Time

pre-recorded terminal sessions available online:

● installing via homebrew: https://asciinema.org/a/147072
● basics and task parallelism: https://asciinema.org/a/147073
● locality and task parallelism: https://asciinema.org/a/147135
● data parallelism: https://asciinema.org/a/147082
Chapel Resources
Chapel Central: [https://chapel-lang.org/](https://chapel-lang.org/)

The Chapel Parallel Programming Language

What is Chapel?

Chapel is a modern programming language that is...
- **parallel**: contains first-class concepts for concurrent and parallel computation
- **productive**: designed with programmability and performance in mind
- **portable**: runs on laptops, clusters, the cloud, and HPC systems
- **scalable**: supports locality-oriented features for distributed memory systems
- **open-source**: hosted on GitHub, permissively licensed

New to Chapel?

As an introduction to Chapel, you may want to...
- read a blog article or book chapter
- watch an overview talk or browse its slides
- download the release
- browse sample programs
- view other resources to learn how to trivially write distributed programs like this:

```plaintext
use CyclicDist; // use the Cyclic distribution library
config const n = 100; // use 100 nodes to override this default
forall i in [1..n].damped(y(i),start=1) do
    writeln(“hello from iteration i, “, i, “ of n “, n, “ running on node “, h[i], “”);
```

What’s Hot?

- Chapel 1.16 is now available—download a copy today!
- The CHIUW 2018 call for participation is now available!
- A recent Cray blog post reports on highlights from CHIUW 2017.
- Chapel is now one of the supported languages on Try It Online!
- Watch talks from ACCU 2017, CHIUW 2017, and ATPESC 2016 on YouTube.
- Browse slides from PADAL, EAGE, EMBRACE, ACCU, and other recent talks.
- See also: Chapel’s New Features.
How to Stalk Chapel

http://facebook.com/ChapelLanguage
http://twitter.com/ChapelLanguage
https://www.youtube.com/channel/UCHmm27bYjhknK5mU7ZzPGsQ/chapel-announce@lists.sourceforge.net
Suggested Reading (healthy attention spans)

Chapel chapter from *Programming Models for Parallel Computing*

- a detailed overview of Chapel’s history, motivating themes, features
- published by MIT Press, November 2015
- edited by Pavan Balaji (Argonne)
- chapter is now also available online

Other Chapel papers/publications available at [https://chapel-lang.org/papers.html](https://chapel-lang.org/papers.html)
Suggested Reading (short attention spans)

- a run-down of recent events

- a short-and-sweet introduction to Chapel

**Six Ways to Say “Hello” in Chapel** (parts 1, 2, 3), Cray Blog, Sep-Oct 2015.
- a series of articles illustrating the basics of parallelism and locality in Chapel

**Why Chapel?** (parts 1, 2, 3), Cray Blog, Jun-Oct 2014.
- a series of articles answering common questions about why we are pursuing Chapel in spite of the inherent challenges

- a series of technical opinion pieces designed to argue against standard reasons given for not developing high-level parallel languages
Chapel StackOverflow and GitHub Issues
Where to..

Submit bug reports:
GitHub issues for chapel-lang/chapel: public bug forum
chapel_bugs@cray.com: for reporting non-public bugs

Ask User-Oriented Questions:
StackOverflow: when appropriate / other users might care
#chapel-users (irc.freenode.net): user-oriented IRC channel
chapel-users@lists.sourceforge.net: user discussions

Discuss Chapel development
chapel-developers@lists.sourceforge.net: developer discussions
#chapel-developers (irc.freenode.net): developer-oriented IRC channel

Discuss Chapel’s use in education
chapel-education@lists.sourceforge.net: educator discussions

Directly contact Chapel team at Cray: chapel_info@cray.com
Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Cray uses codenames internally to identify products that are in development and not yet publically announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, and URIKA. The following are trademarks of Cray Inc.: ACE, APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, THREADSTORM. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.