

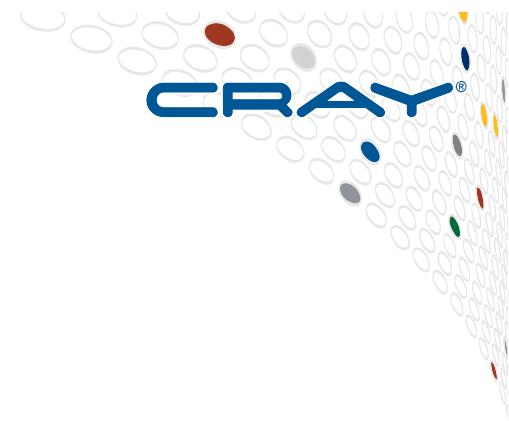
Q & A, Project Status, and Wrap-up



COMPUTE

| STORE

| ANALYZE



Any Questions About the Language?



COMPUTE

|

STORE

|

ANALYZE

Copyright 2018 Cray Inc.



How can I...

...ask questions about the language and its usage?

Stack Overflow: [chapel] tags followed by core developers

chapel-users@lists.sourceforge.net: user-oriented discussion list

#chapel (irc.freenode.net): user-oriented IRC channel

The screenshot shows the Stack Overflow homepage with a search bar containing '[chapel]'. Below the search bar, a section titled 'Tagged Questions' displays three results:

- Get Non-primitive Variables from within a Cobegin - Chapel**
2 votes, 1 answer, 41 views. Asked by xSooDx on Apr 18 at 11:16.
Description: I want to compute some information in parallel and use the result outside the cobegin. To be more precise, my requirement is to retrieve a domain (and other non primitive types) like this var a,b; ...
Tags: chapel
- Is there a default String conversion method in Chapel?**
3 votes, 1 answer, 33 views. Asked by Kyle on Apr 16 at 19:25.
Description: Is there a default method that gets called when I try to cast an object into a string? (E.g. `toString` in Java or `__str__` in Python.) I want to be able to do the following with an array of Objects, ...
Tags: chapel
- Filling eltType with nil values**
0 votes, 1 answer, 112 views. Asked by insta catering on Apr 16 at 1:22.
Description: So I have a chapel issue i can't seem to figure out. I have a queue that one can set size. The only thing is is that it's setting size and filling the queue with a bunch of 0s (which make's sense). I'...
Tags: constructor, parallel-processing, queue, semaphore, chapel



COMPUTE

STORE

ANALYZE

Copyright 2018 Cray Inc.

How can I...

...track the project?

chapel-announce@lists.sourceforge.net: low-frequency announcements

<http://facebook.com/ChapelLanguage>: high-frequency announcements

<http://twitter.com/ChapelLanguage>: high-frequency announcements

<https://www.youtube.com/channel/UCHmm27bYjhknK5mU7ZzPGsQ/> : Chapel videos



The collage illustrates the多样的跟踪方法 for the Chapel project:

- Facebook Page:** Shows the official Chapel Programming Language Facebook page with a green and blue circular logo, a post about the benchmarks game, and navigation links for Page, Messages, Notifications, Insights, and Publishing Tools.
- Twitter Profile:** Shows the official Chapel Language Twitter account (@ChapelLanguage) with a green and blue circular logo, 222 tweets, 12 following, 129 followers, and 32 likes.
- YouTube Channel:** Shows the Chapel Parallel Programming Language YouTube channel with a green and blue circular logo, 270 people reached, and a video titled "SC16 Chapel Tutorial Promo".
- Github Repository:** Shows the official GitHub repository for the Chapel project, featuring the green and blue circular logo, 115 photos and videos, and a link to the "Annual PGAS Applications Workshop".



COMPUTE

STORE

ANALYZE



Where to..

Submit bug reports:

[GitHub issues for chapel-lang/chapel](#): public bug forum
chapel_bugs@cray.com: for reporting non-public bugs

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



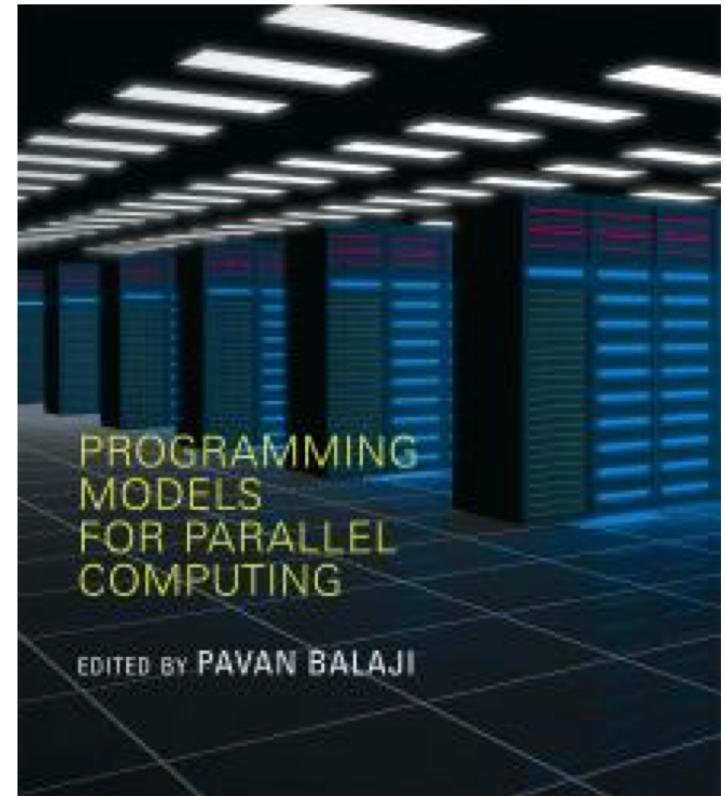
COMPUTE | STORE | ANALYZE

Copyright 2018 Cray Inc.

Suggested Reading

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 <http://chapel.cray.com/papers.html>



Chapel Blog Articles

Chapel: Productive Parallel Programming, Cray Blog, May 2013.

- *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*

[Ten] Myths About Scalable Programming Languages, IEEE TCSC Blog

(index available on chapel.cray.com “blog articles” page), Apr-Nov 2012.

- *a series of technical opinion pieces designed to argue against standard reasons given for not developing high-level parallel languages*





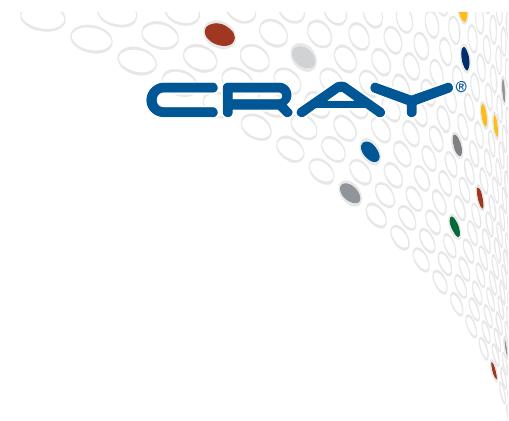
What's Next?

- **Complete new initializer, error-handling features**
- **Continue to improve multi-locale performance & scaling**
 - particularly in the context of application codes
- **Improve support for vectorization and GPUs**
- **Revamp the compiler architecture**
 - make it more approachable to developers
 - improve compilation time
 - support separate compilation and/or incremental recompilation
 - support interactive Chapel programming (e.g., REPL / interpreter)
- **Continue growing set of libraries**
- **Deploy a package manager for Chapel**



COMPUTE | STORE | ANALYZE

Copyright 2018 Cray Inc.



Any Final Questions For Today?



COMPUTE | STORE | ANALYZE

Copyright 2018 Cray Inc.



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.

