CHIUW 2017: Welcome and State of the Project

Brad Chamberlain
Chapel Team, Cray Inc.
June 2, 2017

Safe Harbor Statement

This presentation may contain forward-looking statements that are based on our current expectations. Forward looking statements may include statements about our financial guidance and expected operating results, our opportunities and future potential, our product development and new product introduction plans, our ability to expand and penetrate our addressable markets and other statements that are not historical facts. These statements are only predictions and actual results may materially vary from those projected. Please refer to Cray's documents filed with the SEC from time to time concerning factors that could affect the Company and these forward-looking statements.

Chapel, briefly

What is Chapel?

Chapel: A productive parallel programming language

- portable
- open-source
- a collaborative effort

Goals:

- Support general parallel programming
- Make parallel programming at scale far more productive

Motivation for Chapel

Q: Can a single language be…

…as programmable as Python?
…as fast as Fortran?
…as portable as C?
…as scalable as MPI?
…as generic and meta- as C++? (but using simpler notation?)
…as fun as <your favorite language here>?

A: We believe so.

Q: So why don't we have such languages already?

A: Due to a lack of…

…long-term efforts
…resources
…community will
…developer/user co-design
…patience

Chapel is our attempt to change this

A Year in the Life of Chapel

- **Two major releases per year** (April / October)
  - **~a month later:** detailed release notes
  - **latest release:** Chapel 1.15, released April 6th 2017
- **CHIUW:** Chapel Implementers and Users Workshop (~June)
- **SC** (November)
  - tutorials, panels, BoFs, posters, educator sessions, exhibits, …
  - annual **CHUG (Chapel Users Group) happy hour**

- **Talks, tutorials, research visits, blog posts, …** (year-round)

A Year in the Life of Chapel

- **Two major releases per year** (April / October)
  - **~a month later:** detailed release notes
  - **latest release:** Chapel 1.15, released April 6th 2017
- **CHIUW:** Chapel Implementers and Users Workshop (~June)
- **SC** (November)
  - tutorials, panels, BoFs, posters, educator sessions, exhibits, …
  - annual **CHUG (Chapel Users Group) happy hour**

- **Talks, tutorials, research visits, blog posts, …** (year-round)

Welcome to CHIUW 2017!

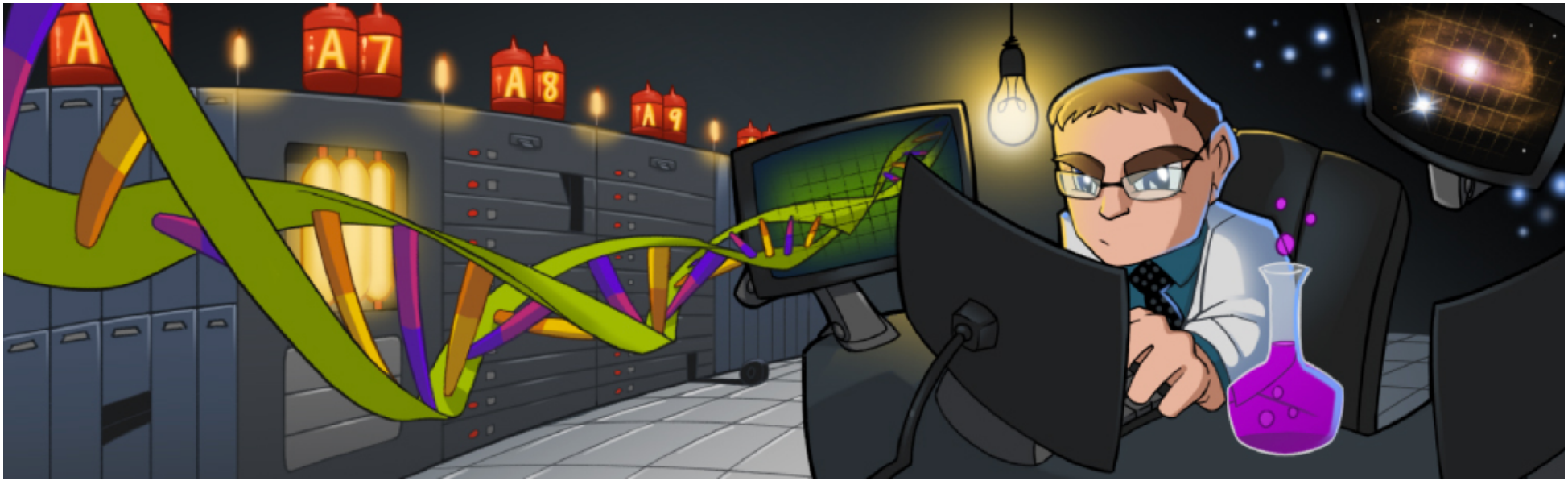
the 4th annual Chapel Implementers and Users Workshop

Chapel's Home in the Landscape of New Scientific Computing Languages

(and what it can learn from the neighbours)

Jonathan Dursi

The Hospital for Sick Children, Toronto



CHIUW 2017: Keynote ("Why do I know that name…?")

Jonathan Dursi

Toronto

canada io mpi mpi-io r astronomy bioinformatics c chapel data fortran gcc hadoop hpc ieee754 julia machine-learning performance presentation python software spark tensorflow tutorial whitepaper

Pictured: The HPC community bravely holds off the incoming tide of new technologies and applications. Via the BBC.

This should be a golden age for High Performance Computing.

For decades, the work of developing algorithms and implementations for tackling simulation and data analysis problems at the largest possible scales was obscure if important work. Then, suddenly, in the mid-2000s, two problems — analyzing internet-scale data, and interpreting an incoming flood of genomics data — arrived on the scene with data volumes and performance requirements which seemed quite familiar to HPCers, but with a size of audience unlike anything that had come before.

Suddenly discussions of topics of scalability, accuracy, large-scale data storage, and distributed matrix arithmetic all became mainstream and urgent. The number of projects and workshops addressing these topics exploded, and new energy went into implementing solutions problems faced in these domains.

In that environment, one might expect that programmers with HPC experience – who have dealt routinely with terabytes and now petabytes of data, and have years or decades of experience with designing and optimizing distributed memory algorithms – would be in high demand.

CHIUW 2017: Keynote ("Why do I know that name…?")

Pictured: The HPC community bravely holds off the incoming tide of new technologies and applications. Via the BBC

CHIUW 2017: Research Papers

Identifying Use-After-Free Variables in Fire-and-Forget Tasks

Jyothi Krishna V S (*IIT Madras*) and Vassily Litvinov (*Cray Inc.*)

Towards a GraphBLAS Library in Chapel

Ariful Azad and Aydin Buluc (*LBNL*)

Comparative Performance and Optimization of Chapel in Modern Manycore Architectures

Engin Kayraklıoğlu, Wo Chang, and Tarek El-Ghazawi (*The George Washington University*)

CHIUW 2017: Technical Talks

Improving Chapel and Array Memory Management
Michael Ferguson, Vassily Litvinov, and Brad Chamberlain (*Cray Inc.*)

Try, Not Halt: An Error Handling Strategy for Chapel
Preston Sahabu, Michael Ferguson, Greg Titus, and Kyle Brady (*Cray Inc.*)

GPGPU support in Chapel with the Radeon Open Compute Platform
Michael Chu, Ashwin Aji, Daniel Lowell, and Khaled Hamidouche (*AMD*)

An OFI libfabric Communication Layer for the Chapel Runtime
Sung-Eun Choi (*Cray Inc.*)

Sketching Streams with Chapel
Christopher Taylor (*DoD*)

Entering the Fray: Chapel's Computer Language Benchmarks Game Entry
Brad Chamberlain, Ben Albrecht, Lydia Duncan, Ben Harshbarger, Elliot Ronaghan, Preston Sahabu, Michael Noakes (*Cray Inc.*), and Laura Delaney (*Whitworth University*)

CHIUW 2017: Planning Committee

General Chairs:

- **Tom MacDonald,** *Cray Inc.*
- **Michael Ferguson,** *Cray Inc.*

Program Committee:

- **Brad Chamberlain** (chair), *Cray Inc.*
- **Nikhil Padmanabhan** (co-chair), *Yale University*
- **Richard Barrett,** *Sandia National Laboratories*
- **Mike Chu,** *AMD*
- **Mary Hall,** *University of Utah*
- **Jeff Hammond,** *Intel*
- **Jeff Hollingsworth,** *University of Maryland*
- **Cosmin Oancea,** *University of Copenhagen*
- **David Richards,** *Lawrence Livermore National Laboratory*
- **Michelle Strout,** *University of Arizona*
- **Kenjiro Taura,** *University of Tokyo*

CHIUIW 2017: Agenda (chapel.cray.com/CHIUW2017.html)

8:30:	Chapel Boot Camp (optional)
9:00:	Welcome, State of the Project
9:30:	Break
10:00:	Talks: Chapel Design and Implementation
11:10:	Quick Break
11:20:	Talks: Targeting New Architectures
12:00:	Lunch
1:30:	Keynote Talk: Jonathan Dursi
2:30:	Talks: Uses of Chapel
3:20:	Break
3:50:	Talks: Benchmarking and Performance
4:40:	Lightning Talks and Flash Discussions
5:30:	Wrap-up / Head to Dinner

CHIUW 2017: Lightning Talks & Flash Discussions

- New this year!
- Last session of the day!
- Goal: high-energy hot topics for low attention spans!
- Format: Short talks, Q&A, war stories, …whatever!
- Sign up for a slot!

CHIUW 2017: Lightning Talks & Flash Discussions

- New this year!
- Last session of the day!
- Goal: high-energy hot topics for low attention spans!
- Format: Short talks, Q&A, war stories, …whatever!
- Sign up for a slot!

CHIUIW 2017: Code Camp (Half) Day

- **Tomorrow morning: 8:30-noon (room: Tangerine 5)**
- **Proposed topics (so far):**
  - write a user-defined domain map
  - Chapel on AWS
  - work on GPU support
  - introduction to Chapel code generation & optimizations
  - network atomics in Chapel's runtime libraries
  - re-architect launcher scripts
  - unified communication diagnostics hooks
  - storage technologies
  - beef up Linear Algebra module
  - port gravitational n-body code to Chapel
  - pyChapel improvements
  - …
  - **your idea here?**

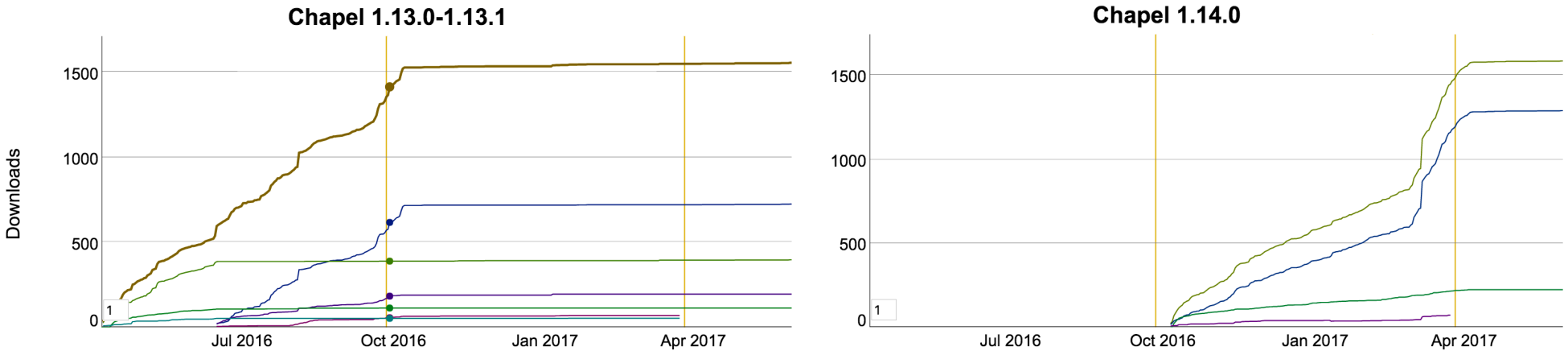
SWAG and Surveys

- **We have a few giveaways today:**
  - Chapel stickers
  - Chapel microfiber wipes for screens / glasses
- **We also have a CHIUW survey**
  - available in paper or online form—please fill one out

State of the Chapel Project 2017

Releases since CHIUW 2016

- **Since last year, two new major versions of Chapel:**
  - **1.14:** October 6, 2016
  - **1.15:** April 6, 2017 ← our most significant release ever!
- **Significant progress in all areas of the release**
  - performance, memory leaks, libraries, documentation, portability, …
- **Achieving 1500+ downloads per release**



Contributors to the Past Year's Releases

Contributors to 1.14 / 1.15:

- Ben Albrecht, Cray Inc.
- Matthew Baker, ORNL
- Paul Cassella, Cray Inc.
- Brad Chamberlain, Cray Inc.
- Sung-Eun Choi, Cray Inc.
- Marcos Cleison Silva Santana, individual contributor
- Laura Delaney, Whitworth University / Cray
- Lydia Duncan, Cray Inc.
- Michael Ferguson, Cray Inc.
- Ben Harshbarger, Cray Inc.
- Andrea Francesco Iuorio, Università degli Studi di Milano / GSoC
- David Iten, Cray Inc.
- David Keaton, Cray Inc.
- Engin Kayraklioglu, George Washington University / Cray Inc.
- Sagar Khatri, individual contributor
- Przemysław Leśniak, individual contributor
- Vassily Litvinov, Cray Inc.
- Tom MacDonald, Cray Inc.
- Deepak Majeti, individual contributor
- Phil Nelson, Western Washington University / Cray
- Michael Noakes, Cray Inc.
- Nikhil Padmanabhan, Yale University
- Nicholas Park, DOD
- Sriraj Paul, Rice University
- Kumar Prasun, individual contributor
- Elliot Ronaghan, Cray Inc.
- Preston Sahabu, Cray Inc.
- Kushal Singh, Int'l Institute of Information Technology, Hyderabad / GSoC
- Kenjiro Taura, University of Tokyo
- Chris Taylor, DOD
- Greg Titus, Cray Inc.
- Rob Upcraft, individual contributor
- Tony Wallace, Cray Inc.
- Hui Zhang, [University of Maryland]

This year saw a record number of contributors to the releases

Contributors to the Past Year's Releases

Contributors to 1.14 / 1.15:

- Ben Albrecht, Cray Inc.
- Matthew Baker, ORNL
- Paul Cassella, Cray Inc.
- Brad Chamberlain, Cray Inc.
- Sung-Eun Choi, Cray Inc.
- Marcos Cleison Silva Santana, individual contributor
- Laura Delaney, Whitworth University / Cray
- Lydia Duncan, Cray Inc.
- Michael Ferguson, Cray Inc.
- Ben Harshbarger, Cray Inc.
- Andrea Francesco Iuorio, Università degli Studi di Milano / GSoC
- David Iten, Cray Inc.
- David Keaton, Cray Inc.
- Engin Kayraklioglu, George Washington University / Cray Inc.
- Sagar Khatri, individual contributor
- Przemysław Leśniak, individual contributor
- Vassily Litvinov, Cray Inc.
- Tom MacDonald, Cray Inc.
- Deepak Majeti, individual contributor
- Phil Nelson, Western Washington University / Cray
- Michael Noakes, Cray Inc.
- Nikhil Padmanabhan, Yale University
- Nicholas Park, DOD
- Sriraj Paul, Rice University
- Kumar Prasun, individual contributor
- Elliot Ronaghan, Cray Inc.
- Preston Sahabu, Cray Inc.
- Kushal Singh, Int'l Institute of Information Technology, Hyderabad / GSoC
- Kenjiro Taura, University of Tokyo
- Chris Taylor, DOD
- Greg Titus, Cray Inc.
- Rob Upcraft, individual contributor
- Tony Wallace, Cray Inc.
- Hui Zhang, [University of Maryland]

17 Cray employees, 3 Cray summer interns/contractors, 15 external contributors

The Chapel Team at Cray (May 2017)

Chapel R&D Organizations

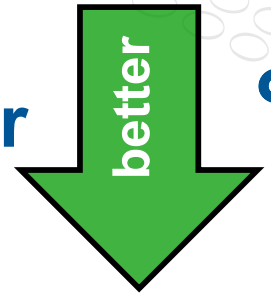
Lawrence Berkeley
National Laboratory

(and several others…)

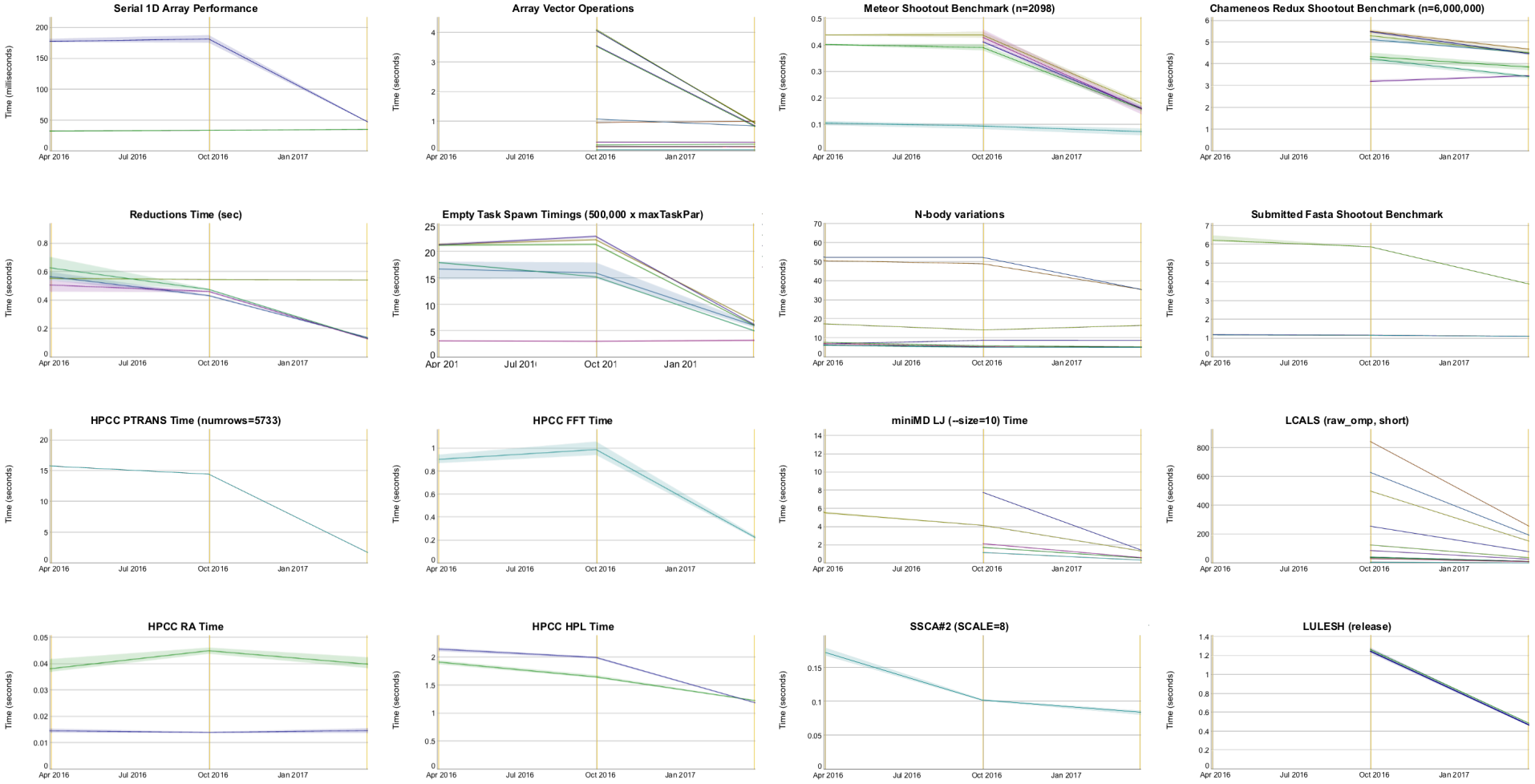
http://chapel.cray.com/collaborations.html

Single-Locale Performance

Single-Locale Performance: the past year



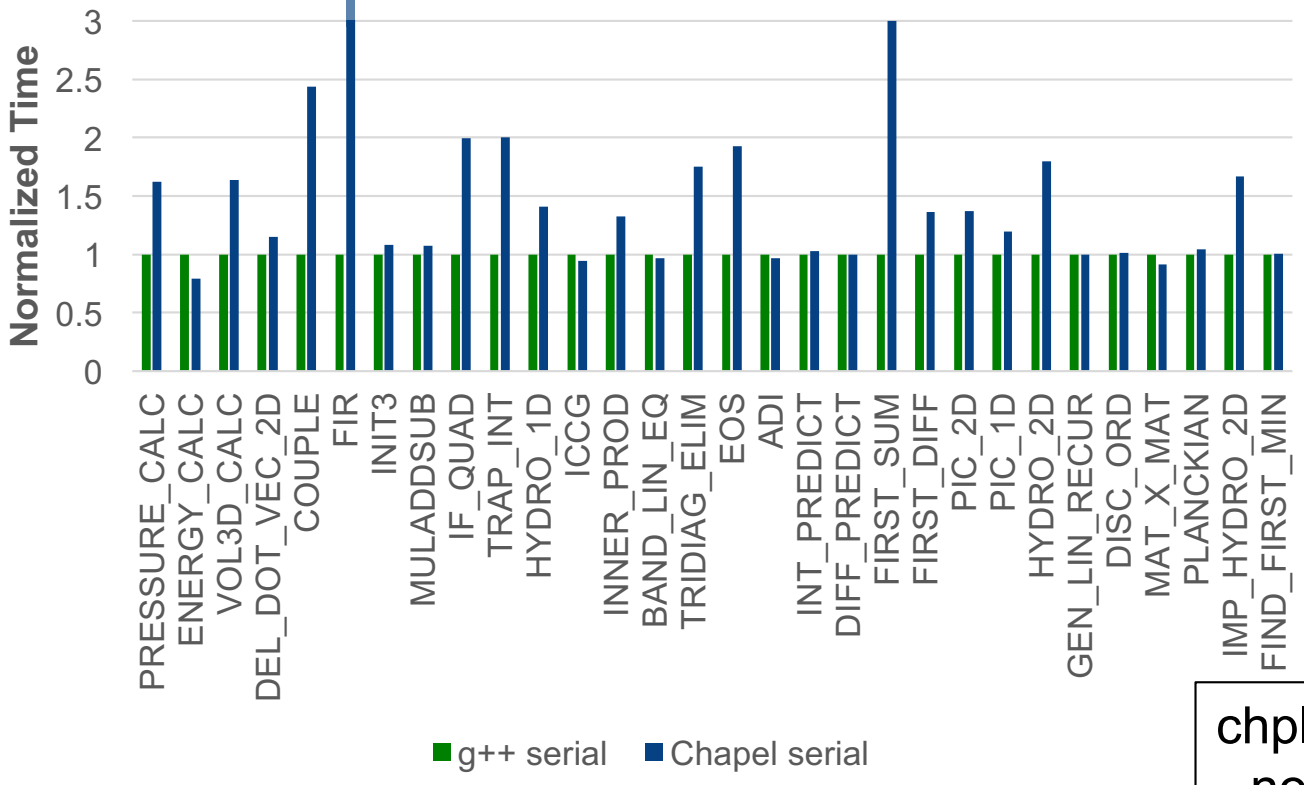
- Overall, single-locale performance improved dramatically



LCALS: Serial Timings, Chapel 1.13.0

Long problem size

(Similar results for medium and short problem sizes)



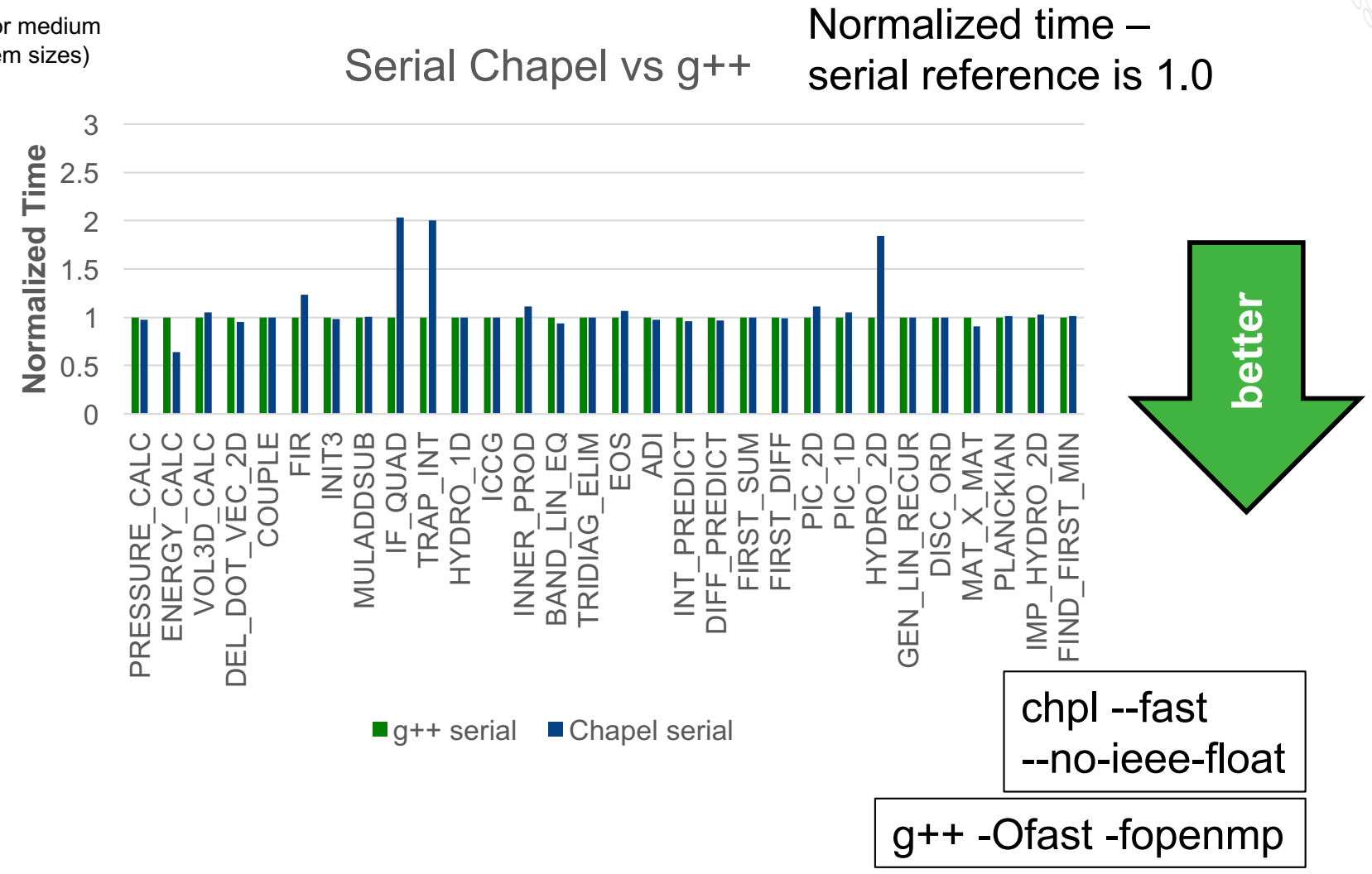
chpl --fast --no-ieee-float

g++ -Ofast -fopenmp

LCALS: Serial Timings, Chapel 1.14.0

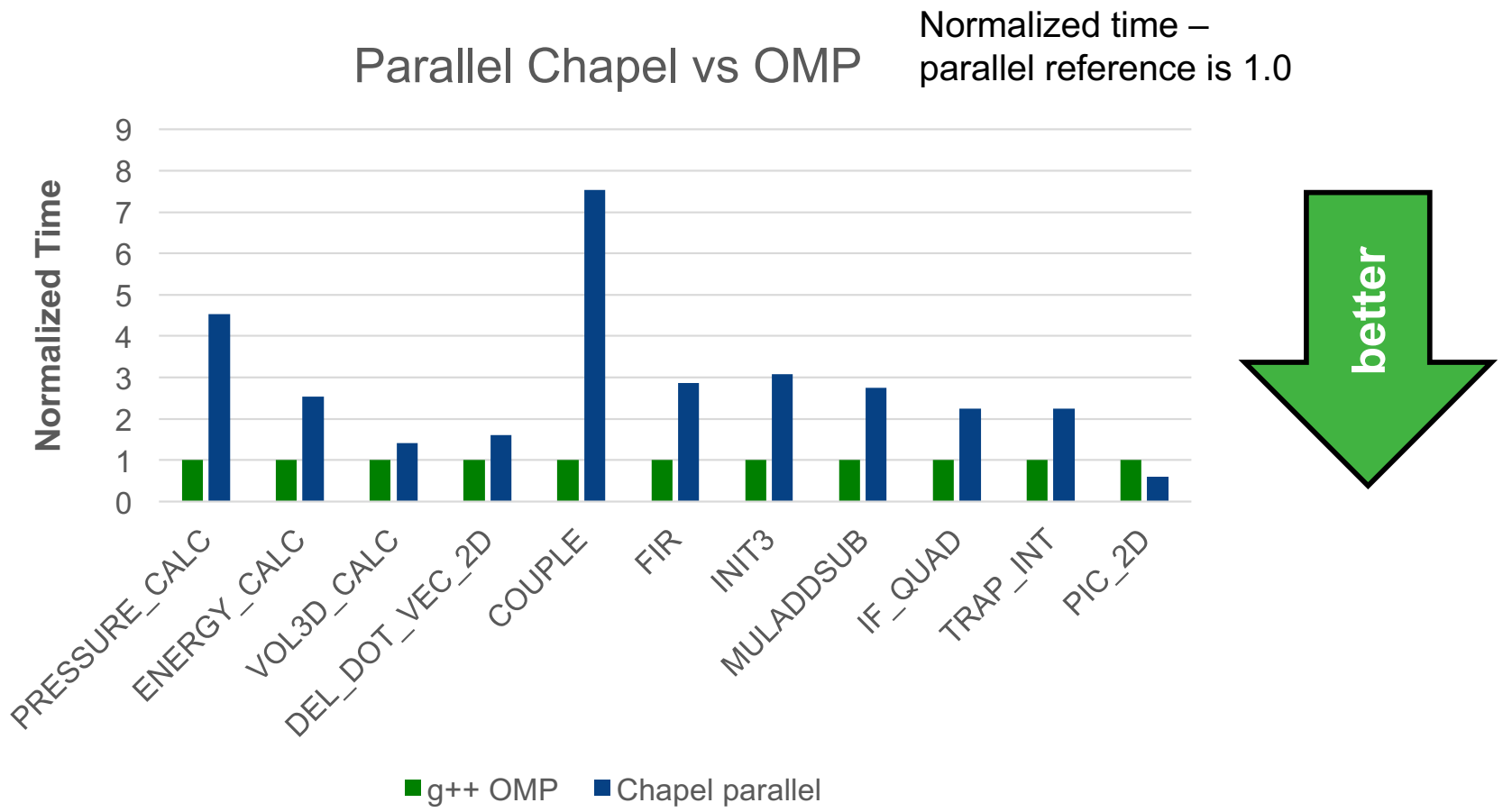
Long problem size

(Similar results for medium and short problem sizes)



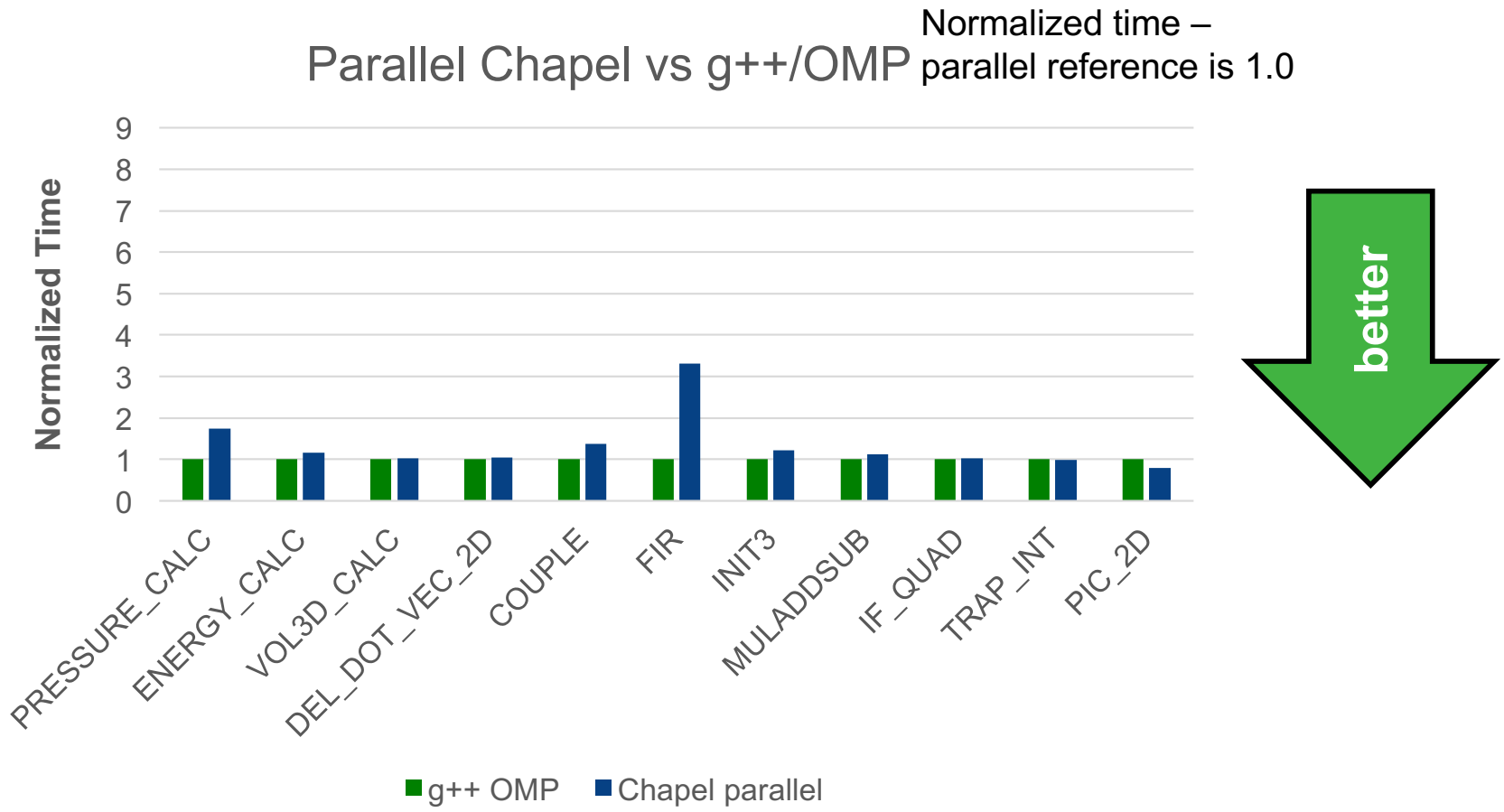
LCALS: Parallel Timings, Chapel 1.14.0

- **Parallel variants still lagged behind the reference in 1.14**
  - between 1.5X and 8X slower for long problem size

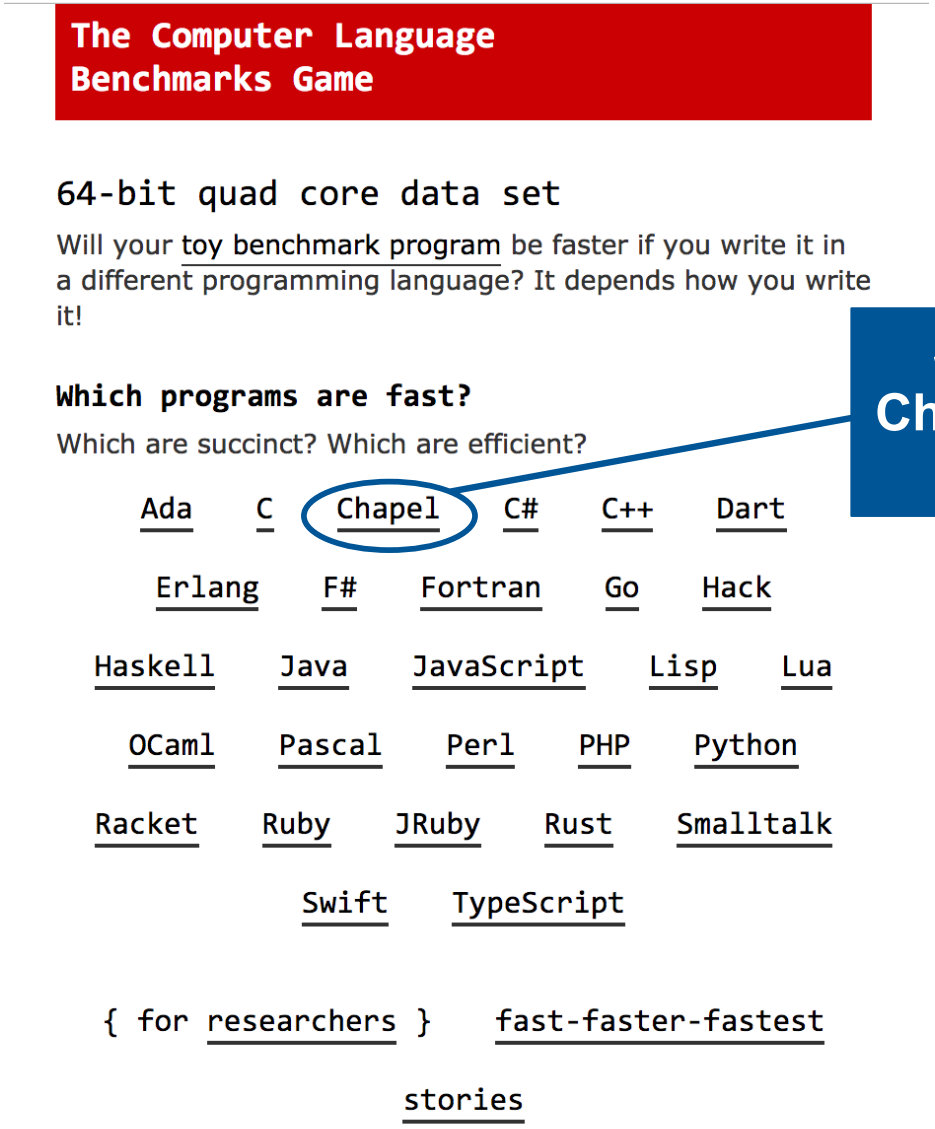


LCALS: Parallel Timings, Chapel 1.15.0

- **Chapel 1.15 closed the gap significantly**
  - ~3-4x speedup: on par or very close to reference for most kernels



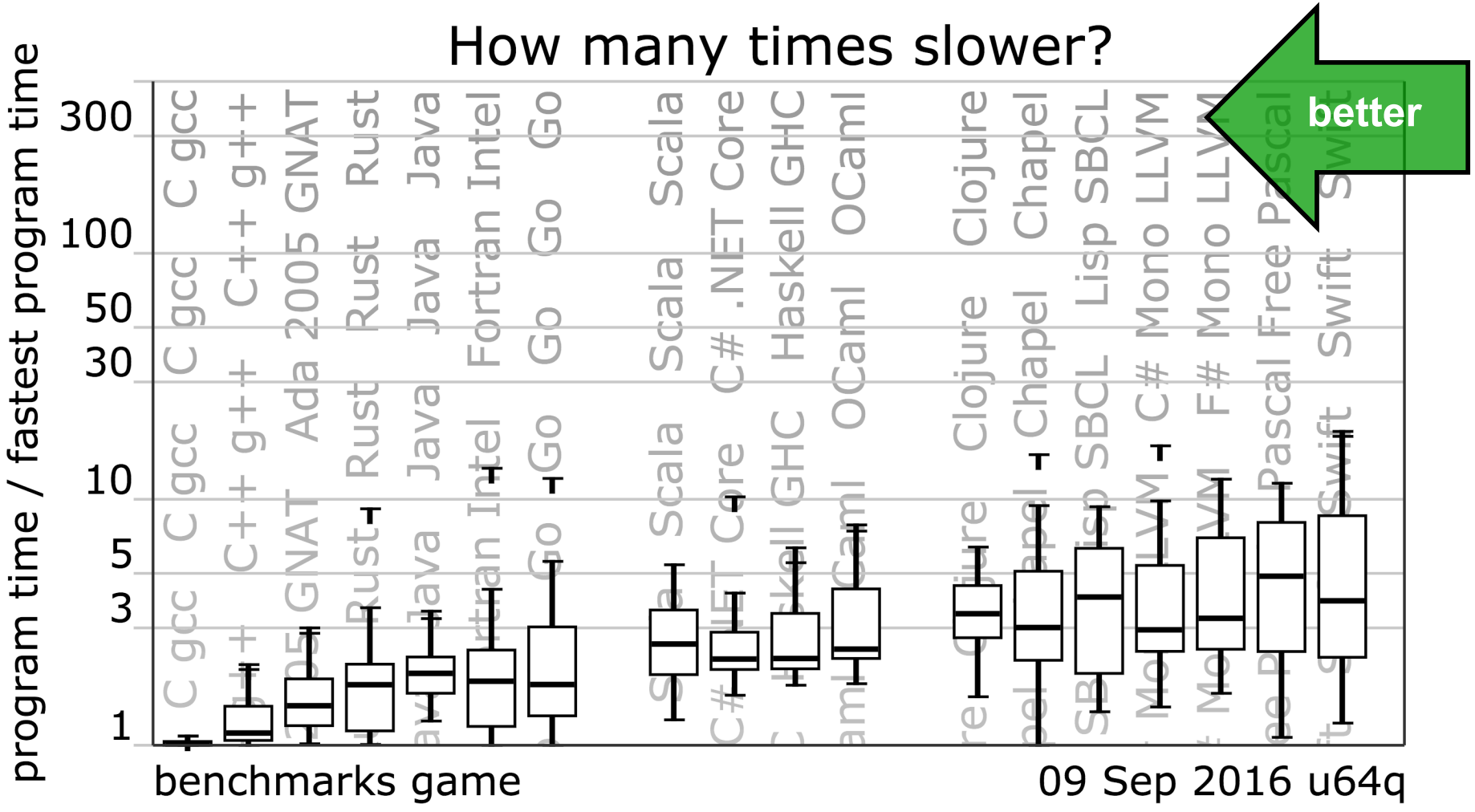
The Computer Language Benchmarks Game



Since CHIUW 2016, Chapel entry completed and listed on site

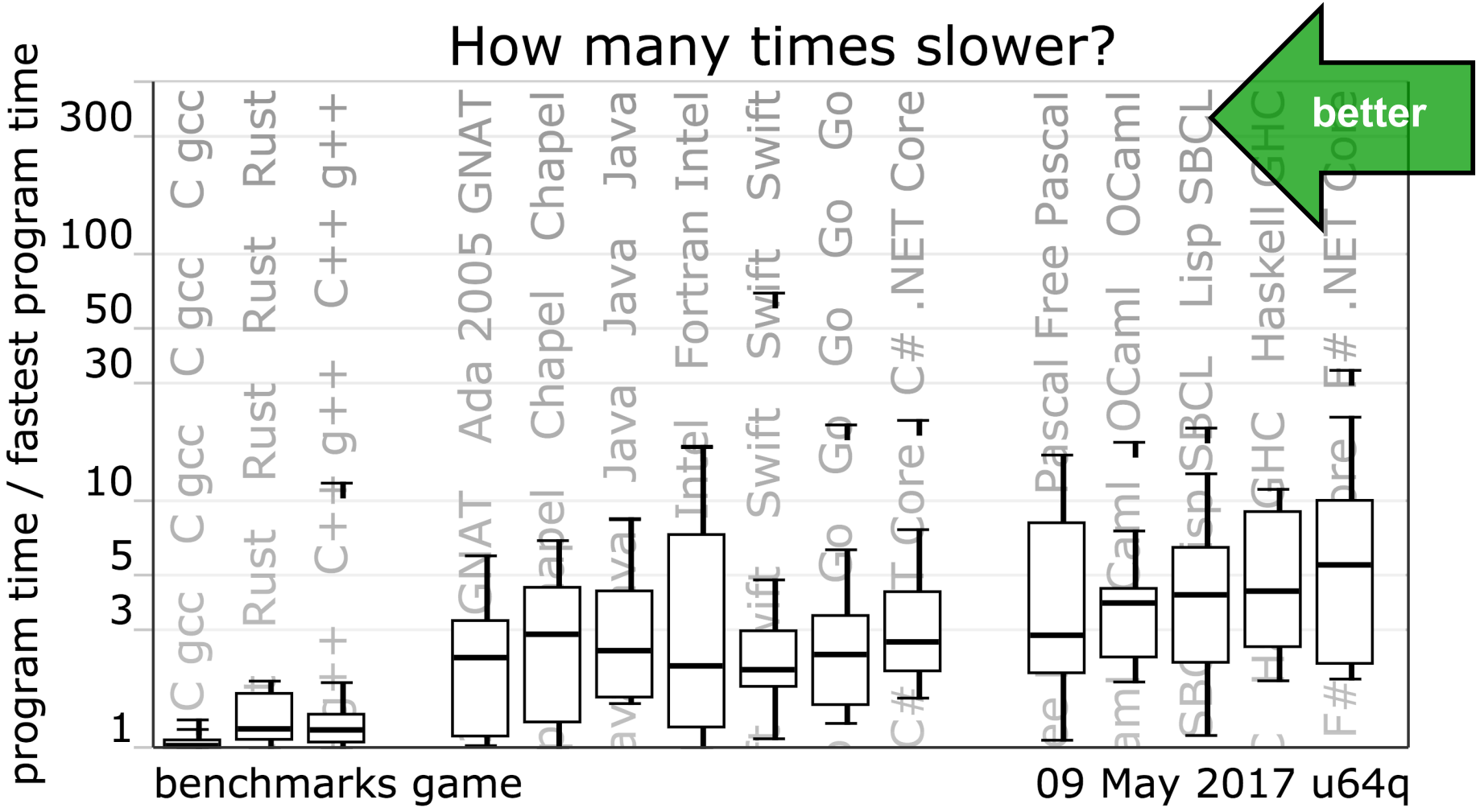
CLBG: Fast-faster-fastest graph (Sep 2016)

Relative performance, sorted by geometric mean



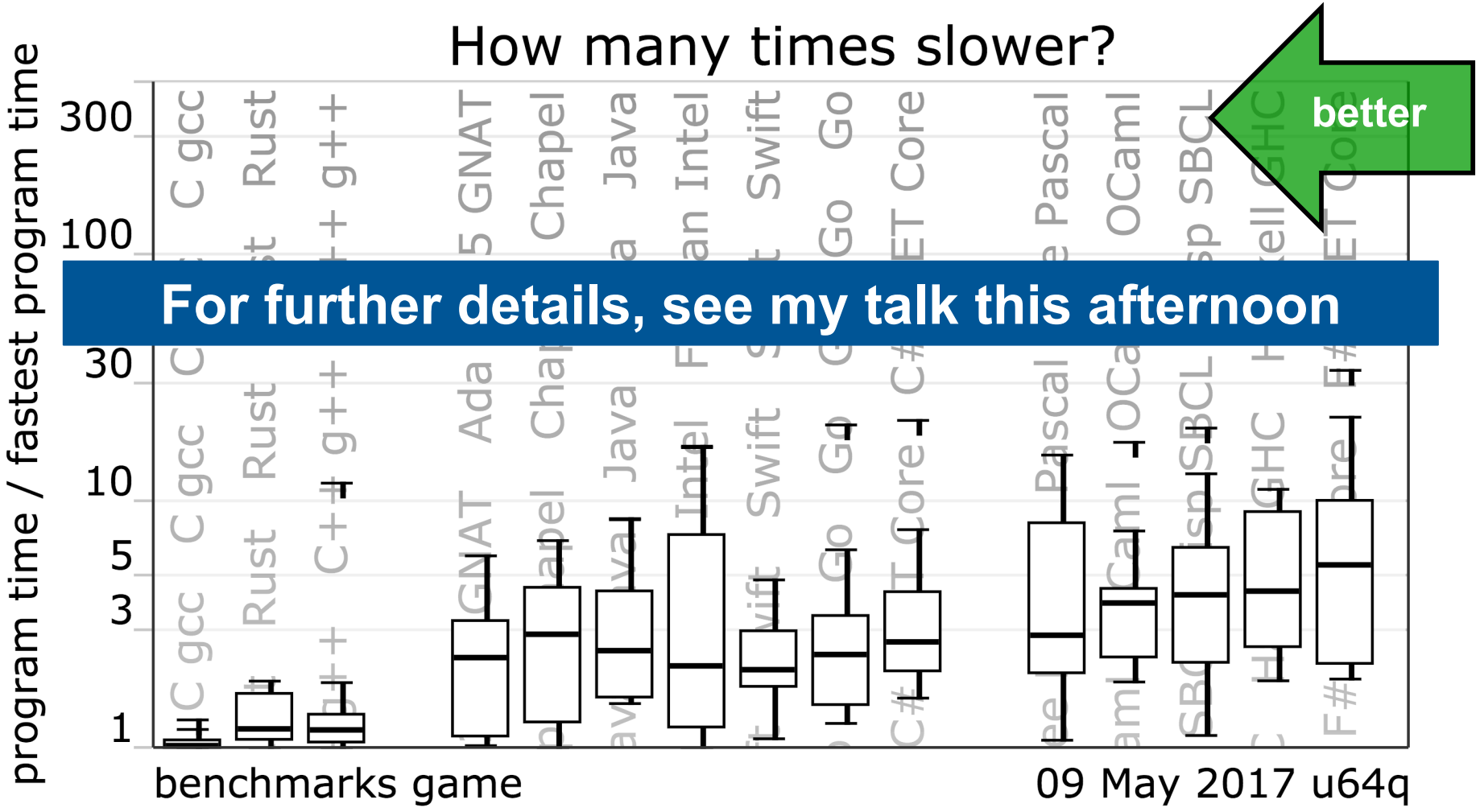
CLBG: Fast-faster-fastest graph (May 2017)

Relative performance, sorted by geometric mean



CLBG: Fast-faster-fastest graph (May 2017)

Relative performance, sorted by geometric mean



Multi-Locale Performance

PGAS Applications Workshop

Monday, November 14th, 2016

Held in conjunction with SC16

In cooperation with:

Optimizing PGAS overhead in a multi-locale Chapel implementation of CoMD

Riyaz Haque and David F. Richards

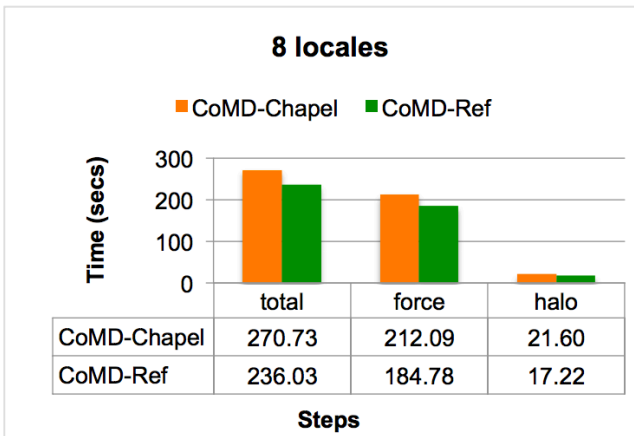
LLNL-PRES-708978

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

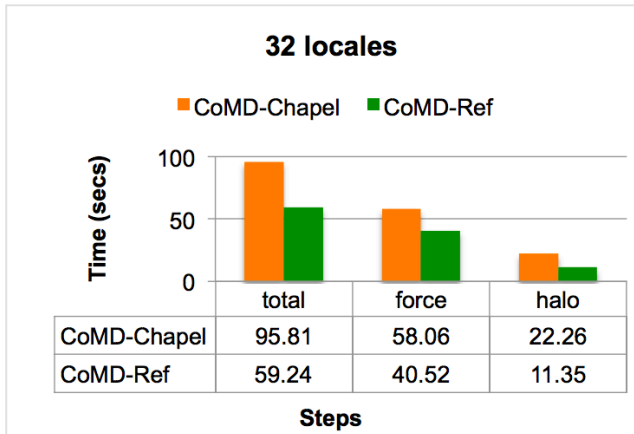
Performance is comparable to the reference implementation

- Code compiled using Chapel v1.13
  - The compiler itself was compiled with gcc-4.9.2, ibv on gasnet and qthreads as the threading framework
- Executed on 1-32 nodes of 64-bit Intel Xeon processors
  - 12 cores, 24 GB RAM, Infiniband high-speed interconnect



	total	force	halo
CoMD-Chapel	270.73	212.09	21.60
CoMD-Ref	236.03	184.78	17.22

Lower is better!



	total	force	halo
CoMD-Chapel	95.81	58.06	22.26
CoMD-Ref	59.24	40.52	11.35

CoMD-Chapel vs. CoMD-Ref, $4x10^6$ atoms

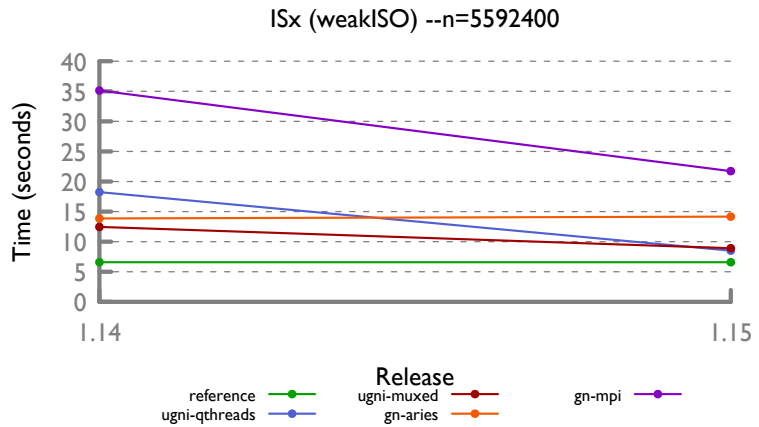
CoMD-Chapel performs to within 87% (8 locales) to 67% (32 locales) of the reference

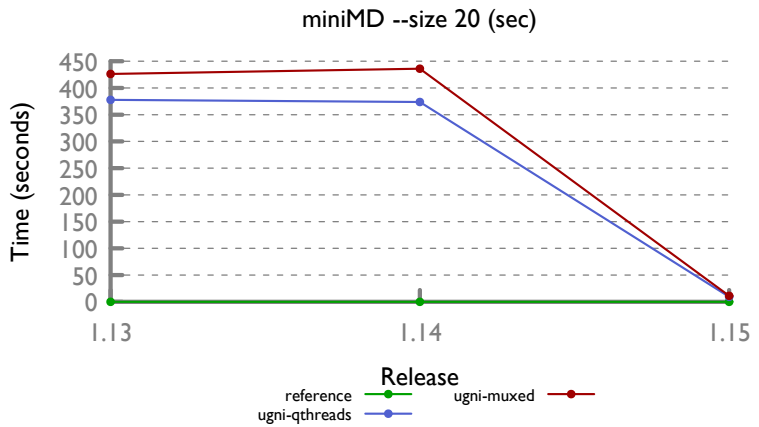
LLNL-PRES-708978

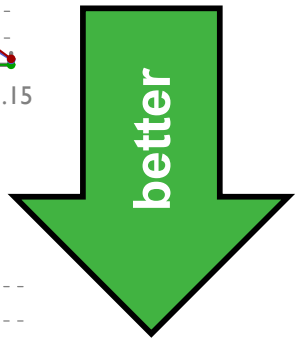
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344

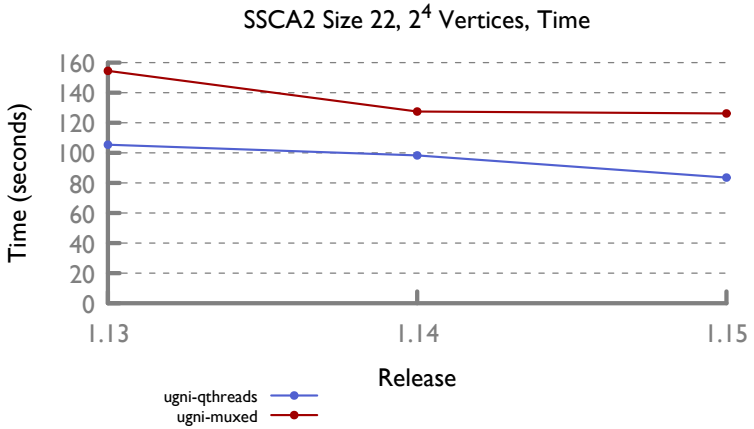
Multi-locale Performance

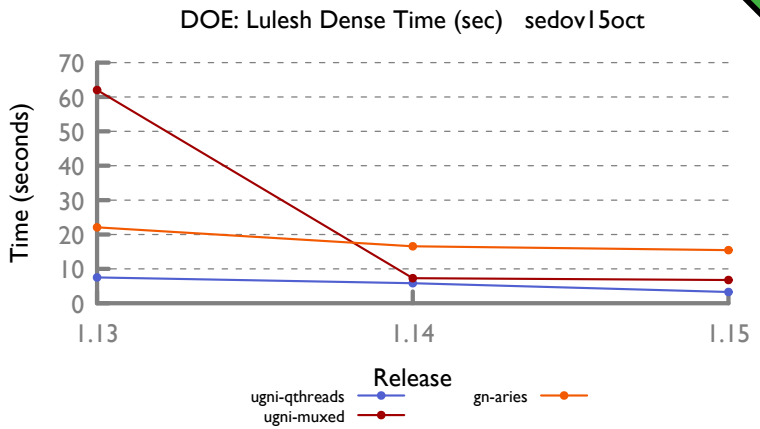
- **Significant multi-locale performance improvements**
  - no known regressions





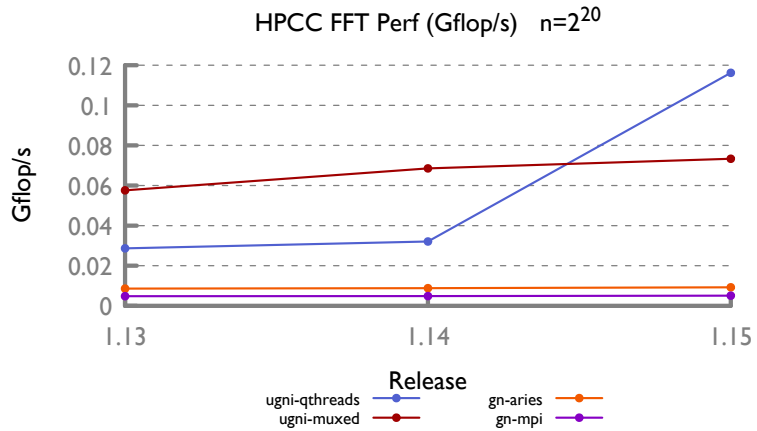


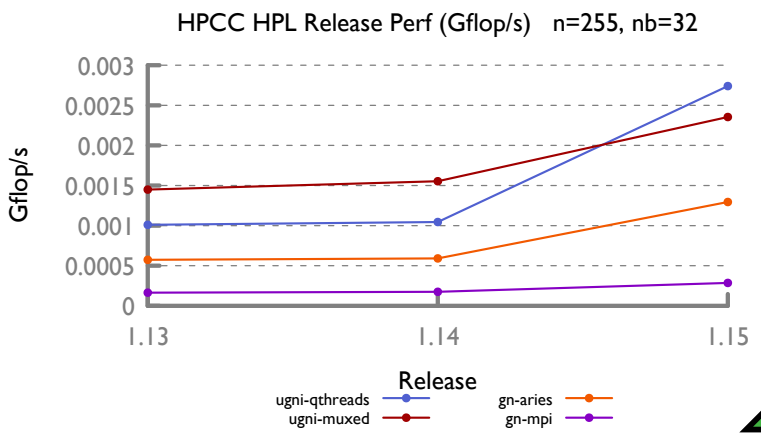


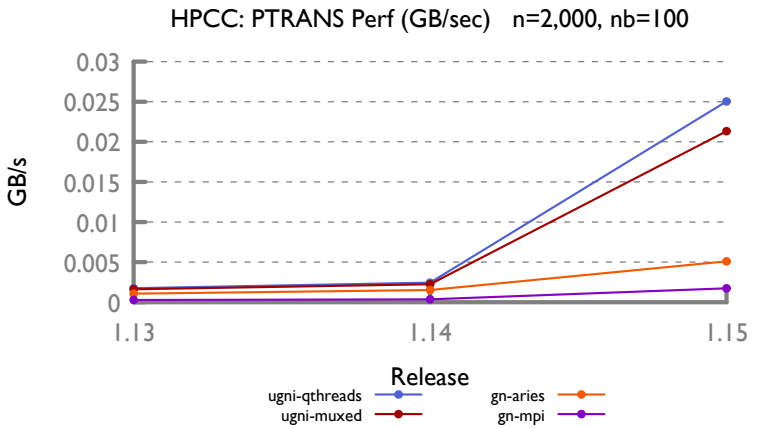


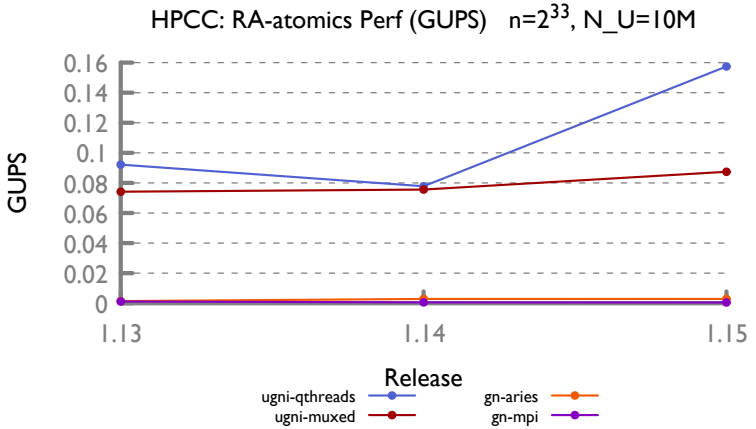
Multi-locale Performance

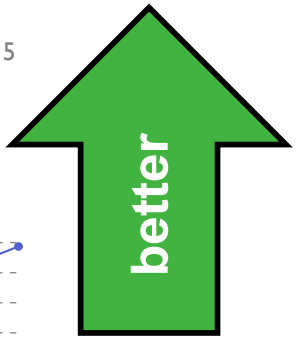
- **Significant multi-locale performance improvements**
  - no known regressions (qthreads now outperforms muxed even more)





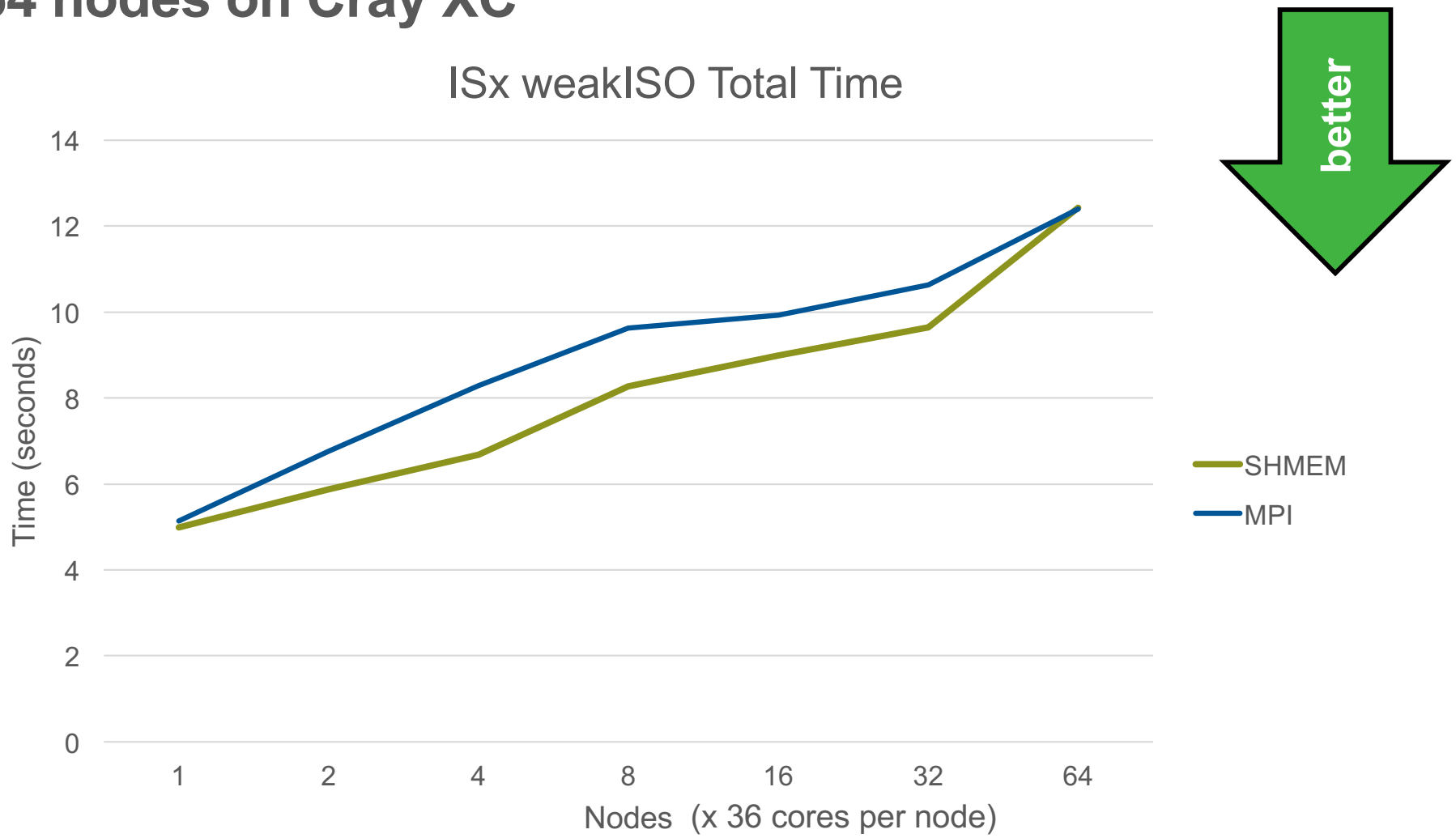






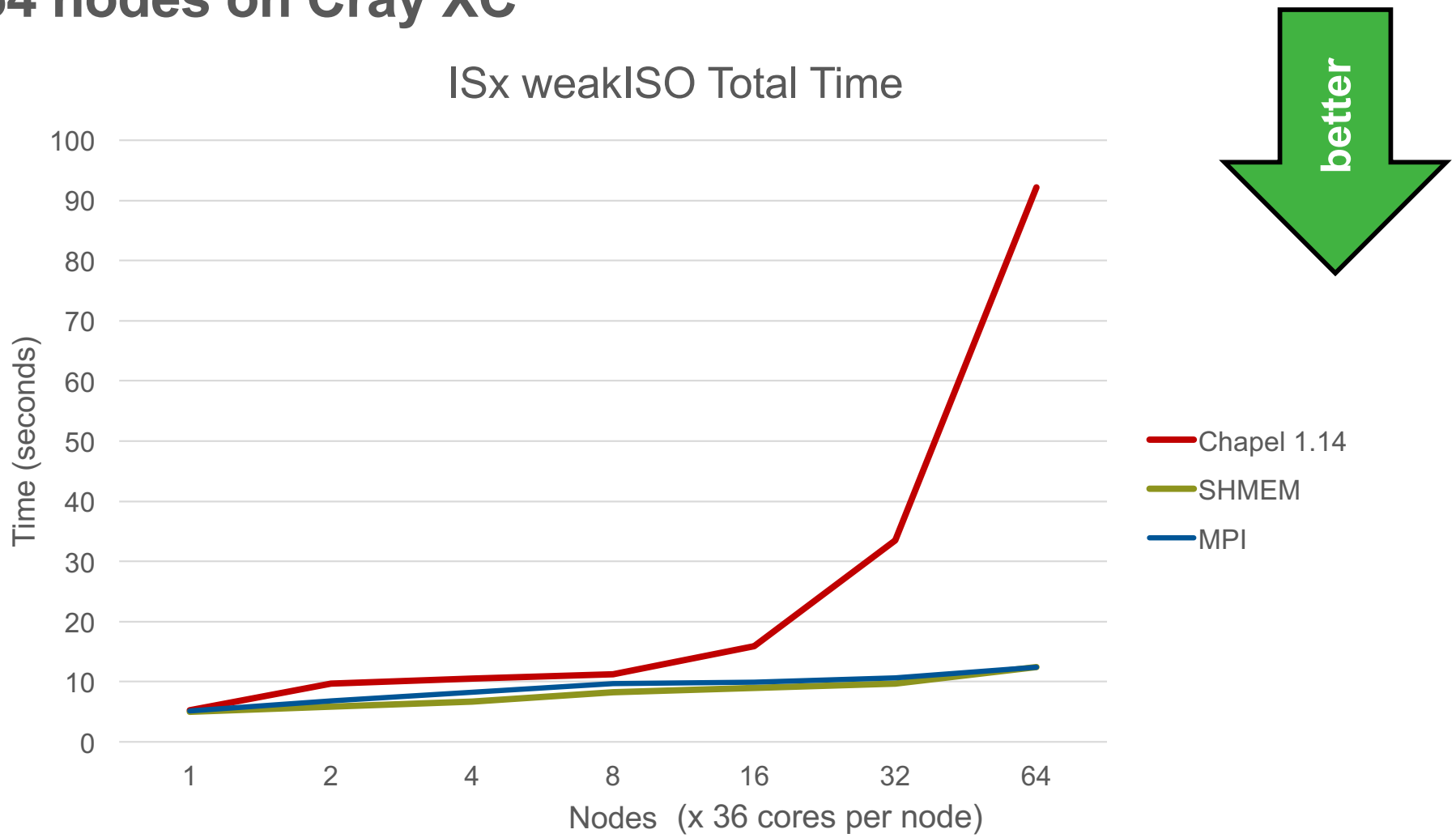
ISx Execution Time: MPI, SHMEM

- 64 nodes on Cray XC



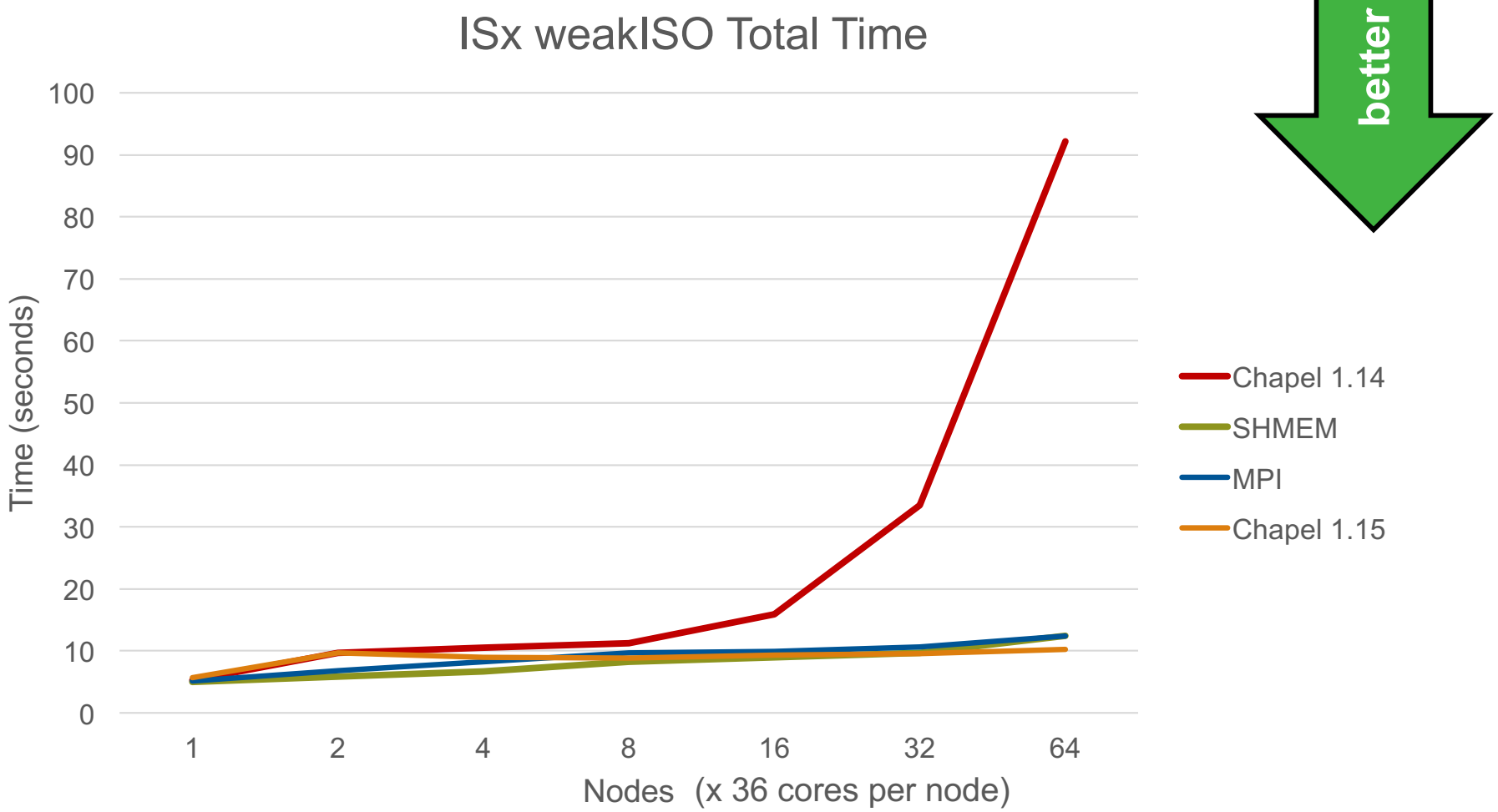
ISx Execution Time: MPI, SHMEM, Chapel 1.14

- 64 nodes on Cray XC



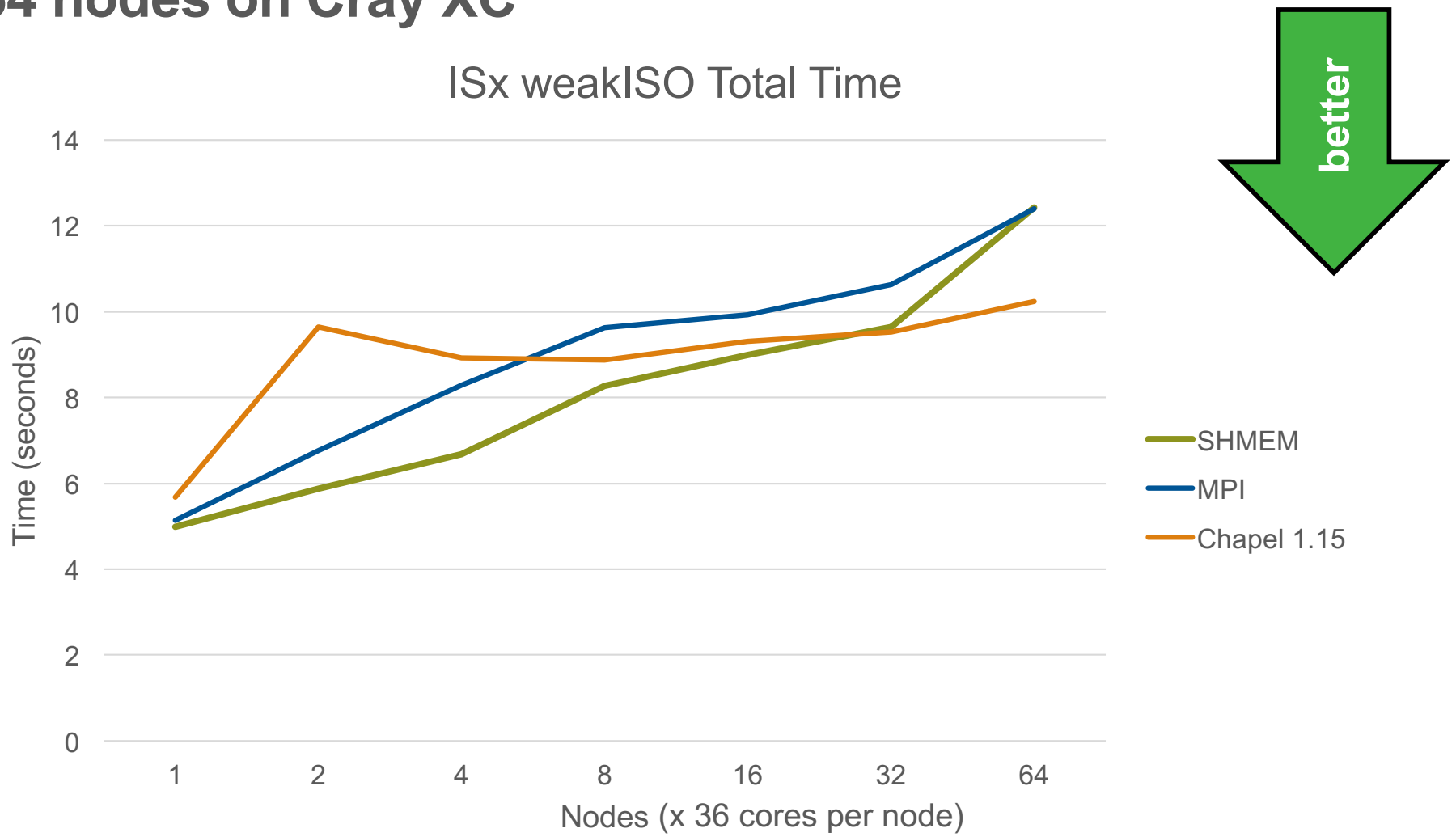
ISx Execution Time: MPI, SHMEM, Chapel

- 64 nodes on Cray XC

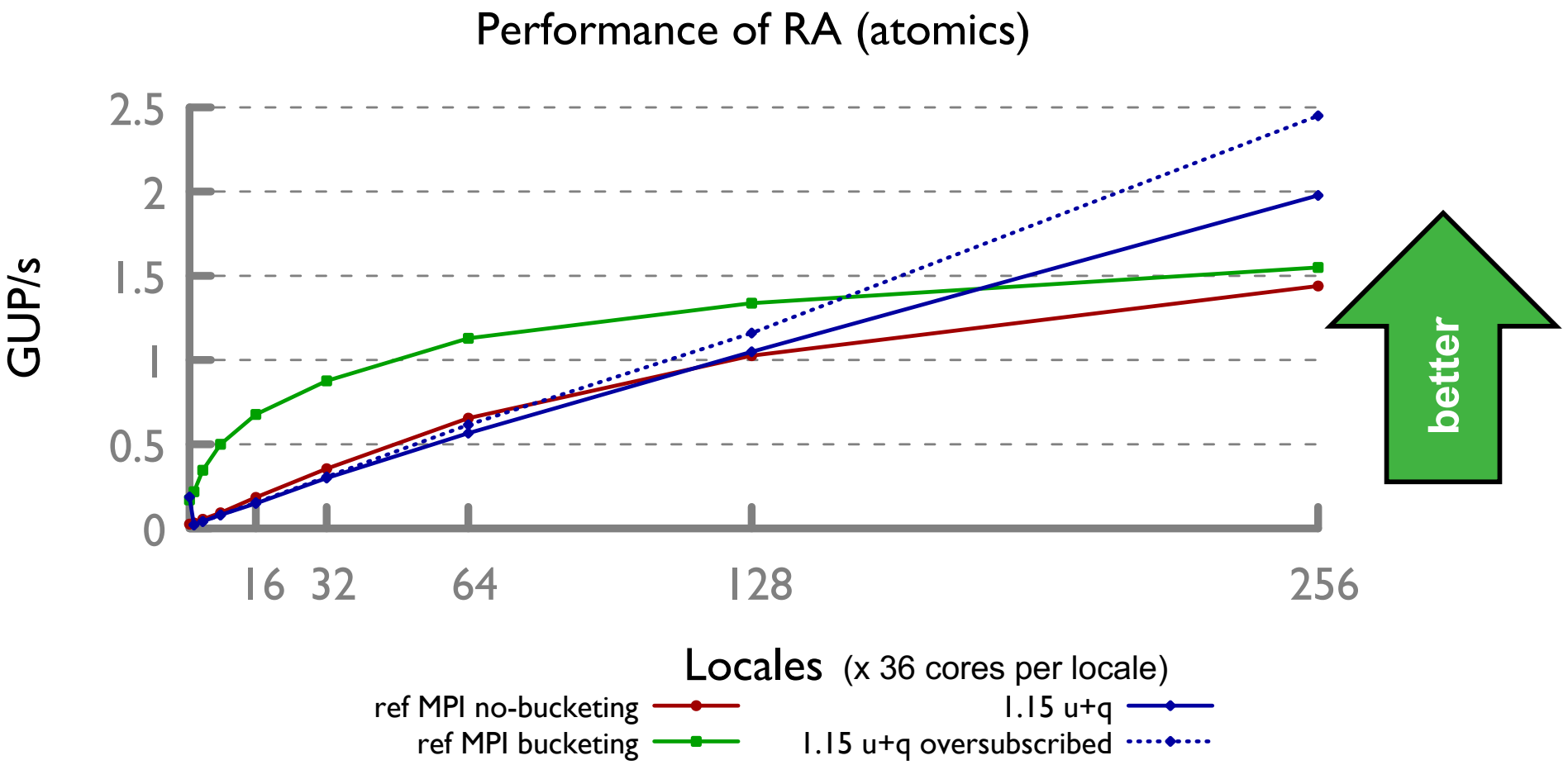


ISx Execution Time: MPI, SHMEM, Chapel 1.15

- 64 nodes on Cray XC



RA Performance: Chapel vs. MPI



Performance: Summary

Summary:

- Chapel has achieved dramatic performance improvements this year

Next steps:

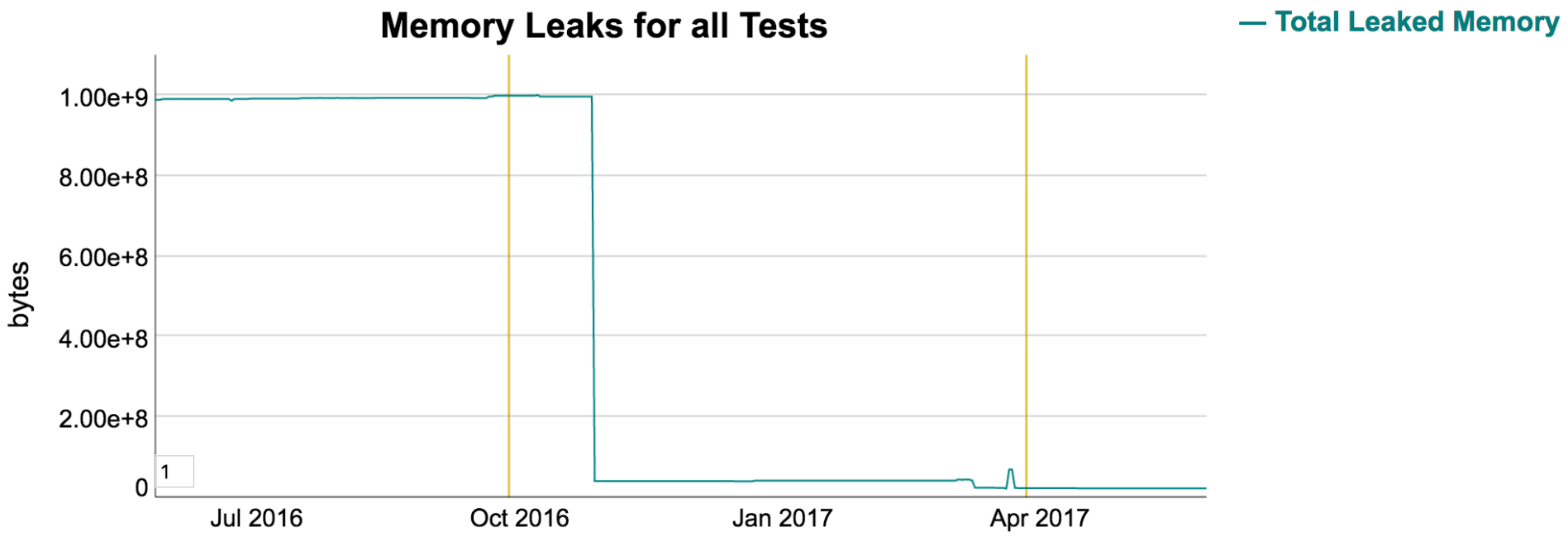
- **Multi-locale:**
  - benchmark-driven performance and scalability improvements
    - particularly for stencils, PRKs, motivating applications
- **Single-locale:**
  - vectorization

Memory Improvements

Reduction in Memory Leaks

Summary:

- We've closed the last major source of compiler-introduced leaks:



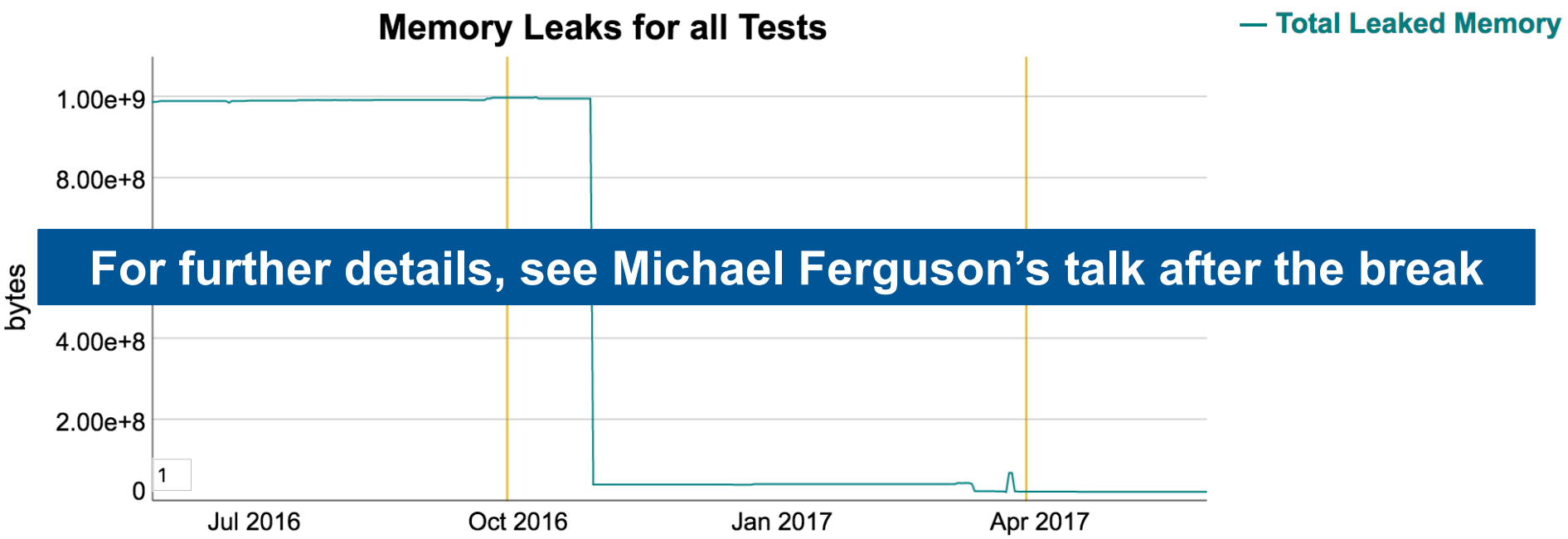
Next Steps:

- Close user-introduced leaks in tests themselves

Reduction in Memory Leaks

Summary:

- We've closed the last major source of compiler-introduced leaks:



For further details, see Michael Ferguson's talk after the break

Next Steps:

- Close user-introduced leaks in tests themselves

Library Improvements

Library Improvements in 1.14–1.15

New Libraries:

- Date / Time
- Owned / Shared for delete-free class objects
- Futures
- BLAS
- MPI
- ZeroMQ
- BigInteger
- MatrixMarket
- RangeChunk
- LinearAlgebra (first draft)

Improved Libraries:

- FFTW
- Sort/Search

Library Improvements in 1.14–1.15

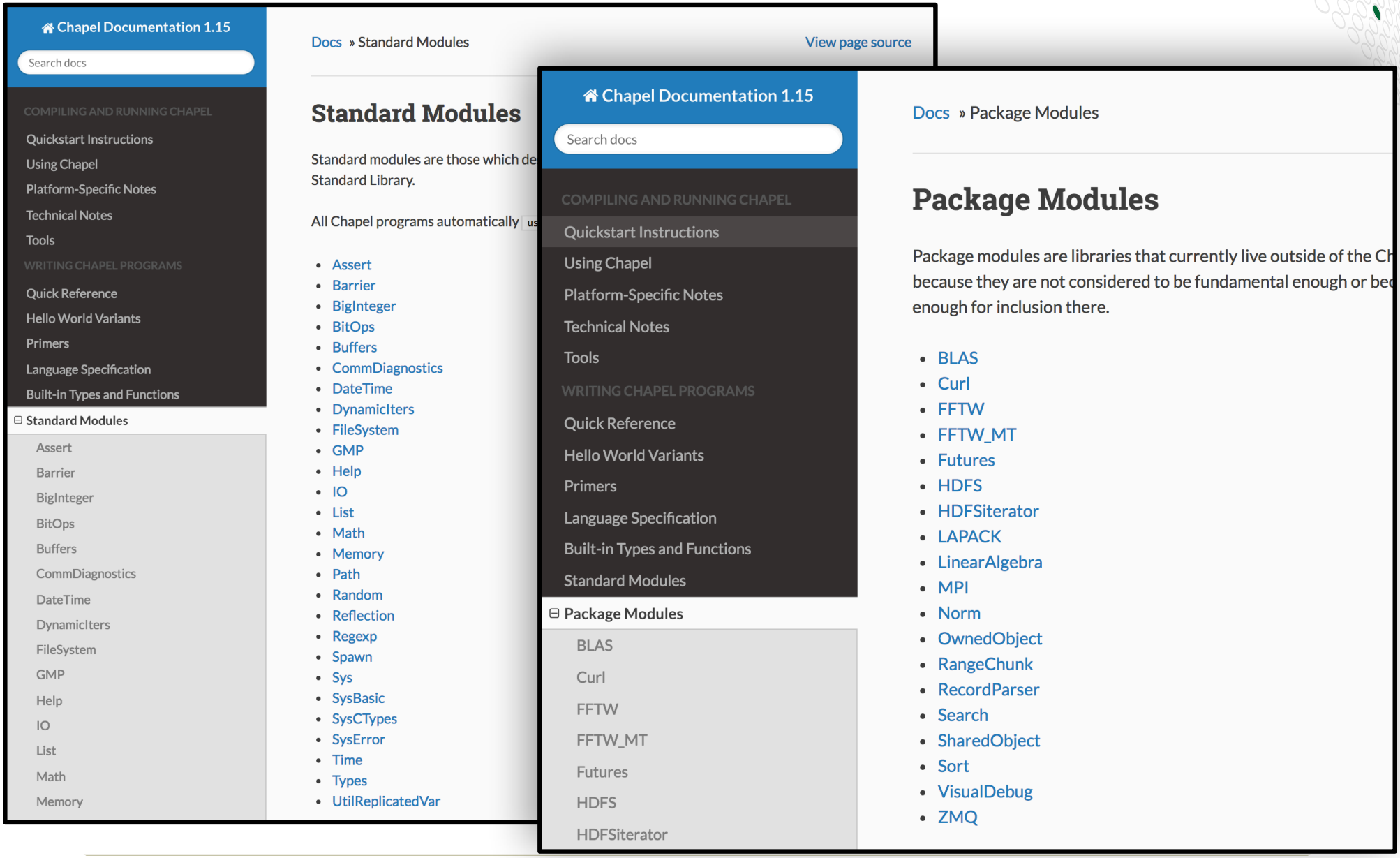
New Libraries:

- Date / Time
- Owned / Shared for delete-free class objects
- Futures
- BLAS
- MPI
- ZeroMQ (contributed by non-Cray developers)
- BigInteger
- MatrixMarket
- RangeChunk
- LinearAlgebra (first draft)

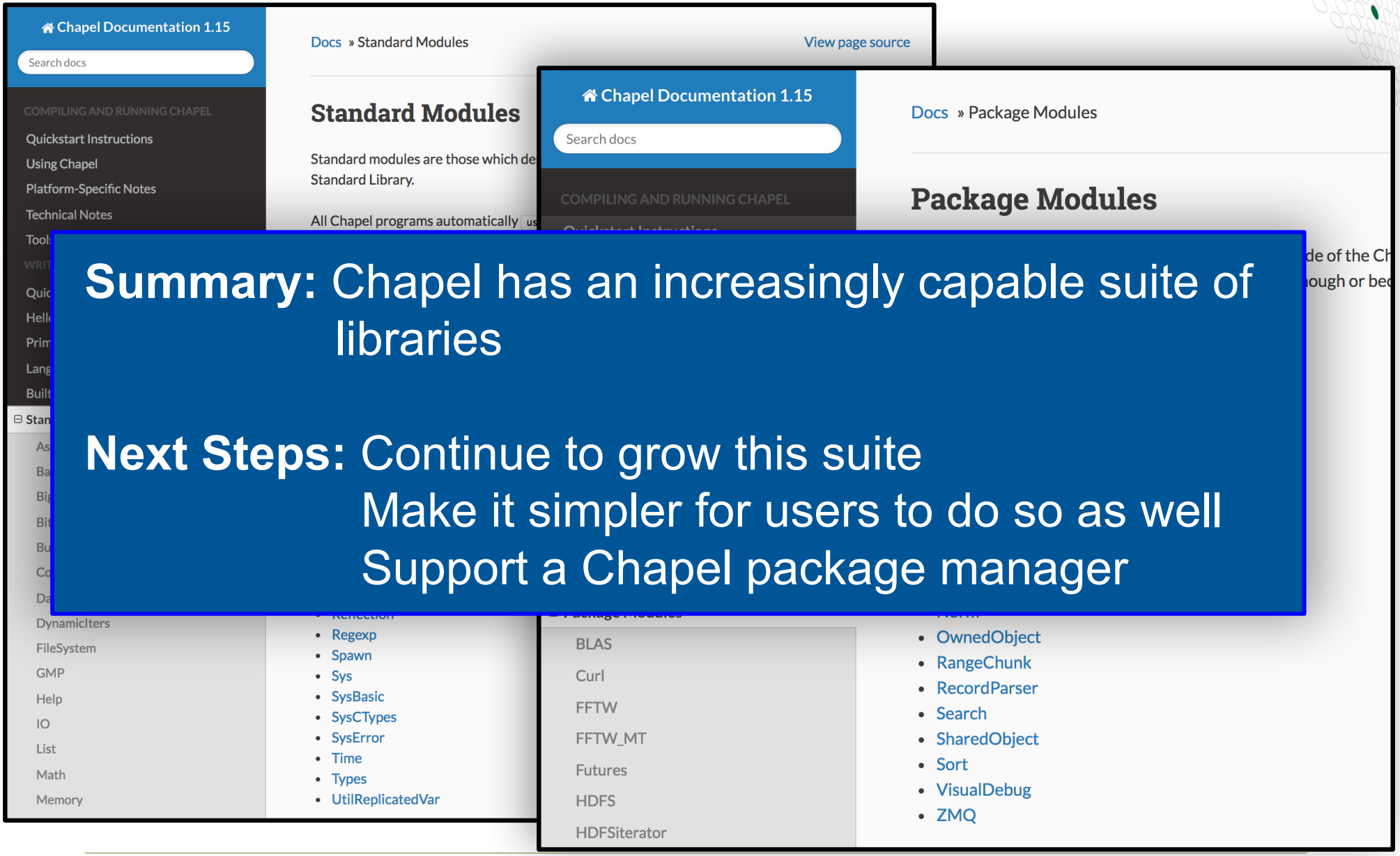
Improved Libraries:

- FFTW
- Sort/Search

Libraries: Summary



Libraries: Summary



Summary: Chapel has an increasingly capable suite of libraries

Next Steps: Continue to grow this suite
Make it simpler for users to do so as well
Support a Chapel package manager

Interoperability Improvements

- Users can now pass Chapel function pointers to C
- Extern block support is far more robust
  - Sample import of GSL routines from C:

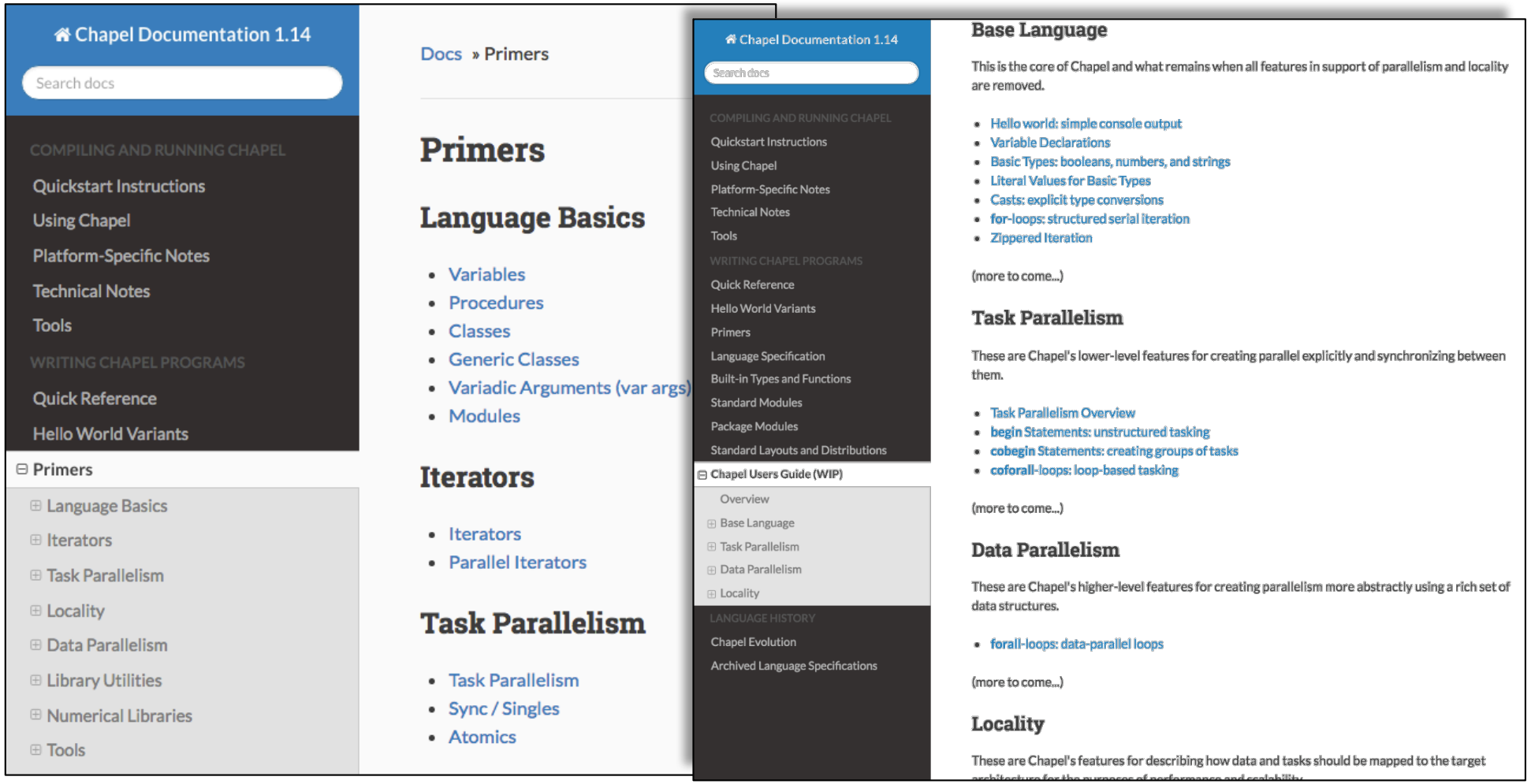
```
extern {
  // Special functions
  #include "gsl/gsl_sf.h"
  // Constants
  #include "gsl/gsl_const.h"
  // Integration
  #include "gsl/gsl_integration.h"
  // Random numbers and distributions
  #include "gsl/gsl_rng.h"
  #include "gsl/gsl_randist.h"
  #include "gsl/gsl_cdf.h"
  // Interpolation
  #include "gsl/gsl_interp.h"
  #include "gsl/gsl_spline.h"
}
```

- Improvements to c2chapel script (see version on master)

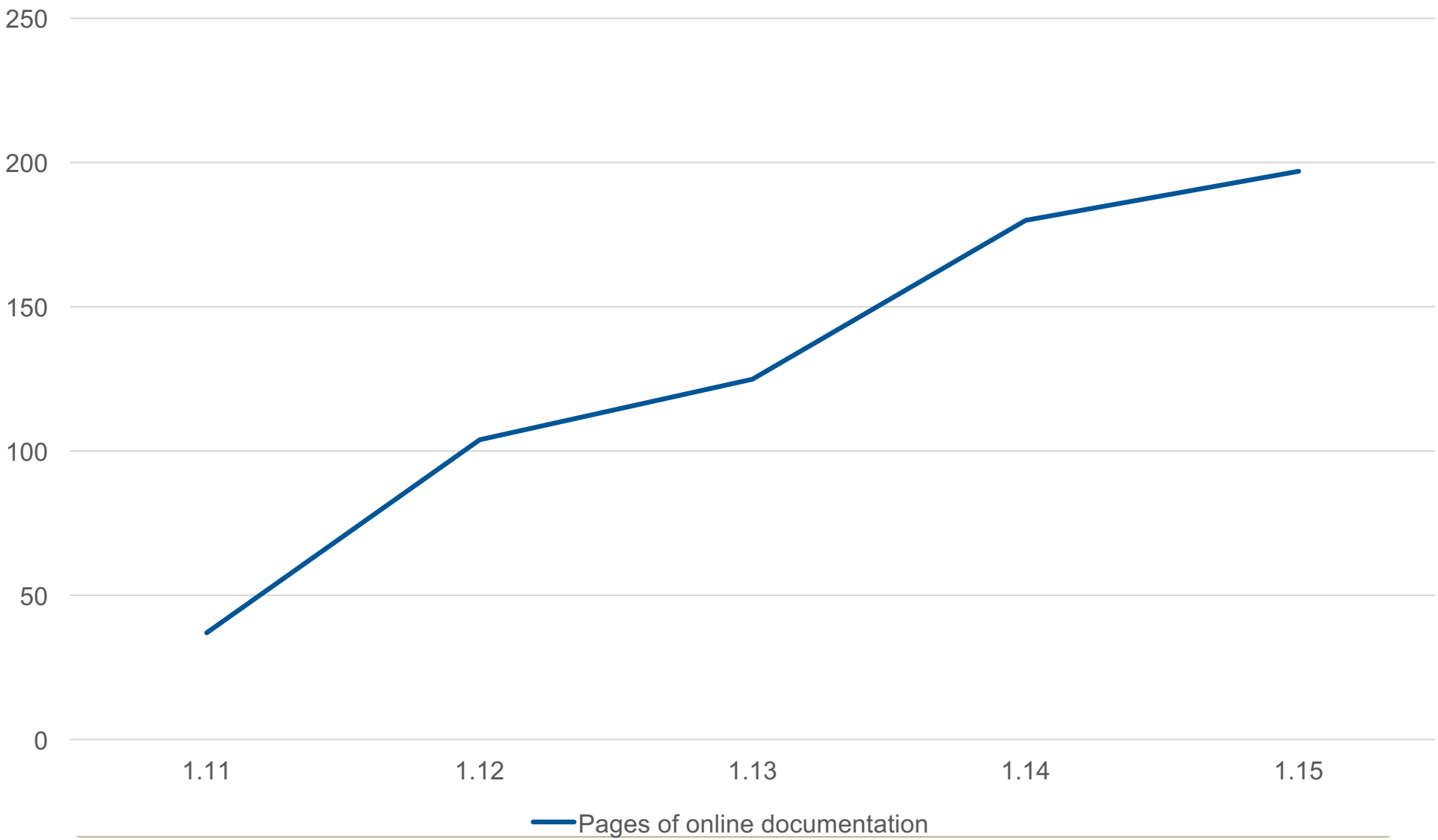
Documentation Improvements

Documentation Improvements

- Significant Expansions / Improvements to online docs:



Pages of Online Docs across Releases



Portability / Packaging Improvements

Intel Xeon Phi ("KNL") locale model

- Chapel can target KNL's MCDRAM via on-clauses

```
on here.highBandwidthMemory() {
  x = new myClass();       // placed in MCDRAM
  ...
  on here.defaultMemory() {
    y = new myClass();     // placed in DDR
    ...
  }
}

on y.locale.highBandwidthMemory() {
  z = new myClass();       // same locale as y, but using MCDRAM
  ...
}
```

Other Portability / Packaging Improvements

- AWS EC2
- Windows 10 bash shell
- Docker package now available
- ARM64 support
- Support for Chapel configuration .dotfiles
- Improved portability across various *nix flavors
- in-progress:
  - Debian
  - AMD
  - OFI / libfabric

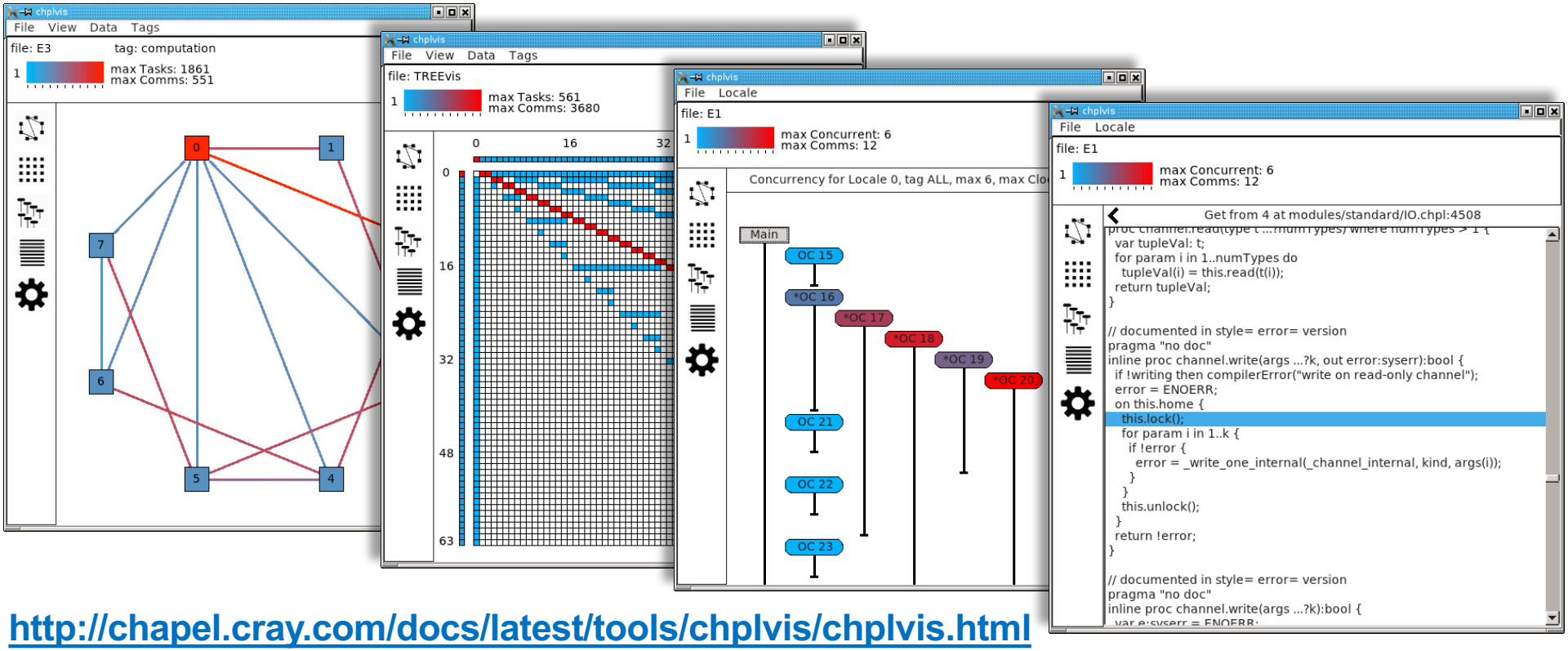
Other Portability / Packaging Improvements

- AWS EC2
- Windows 10 bash shell
- Docker package now available
- ARM64 support
- Support for Chapel configuration .dotfiles
- Improved portability across various *nix flavors
- in-progress:
  - Debian
  - AMD
  - OFI / libfabric

For further details on the AMD and OFI / libfabric efforts, see the portability talks by Mike Chu and Sung-Eun Choi before lunch

Tool Improvements

chplvis: Chapel Execution Visualization Tool



http://chapel.cray.com/docs/latest/tools/chplvis/chplvis.html

Core Improvements

Language Improvements

- **Improvements:**
  - improved array semantics
  - improved support for generic objects
  - first-class ‘void’ type
  - module-level de-initializers
  - forwarding fields in objects
  - fixed where-clause support
- **In-Progress:**
  - initializers (constructor replacement)
  - error-handling

See Preston Sahabu’s talk this morning

Domain Map Improvements

New distributions:

- Stencil distribution
- Sparse Block distribution (needs further tuning)

See early block sparse results in Ariful Azad's talk this afternoon

Domain map improvements:

- Added locality queries to distributed domains/arrays
- Simplified domain map standard interface

Misc Improvements

- Open-sourced ‘ugni’ communication layer
- Support for stack traces on program halt()s (GSoC project)

Meta Stuff

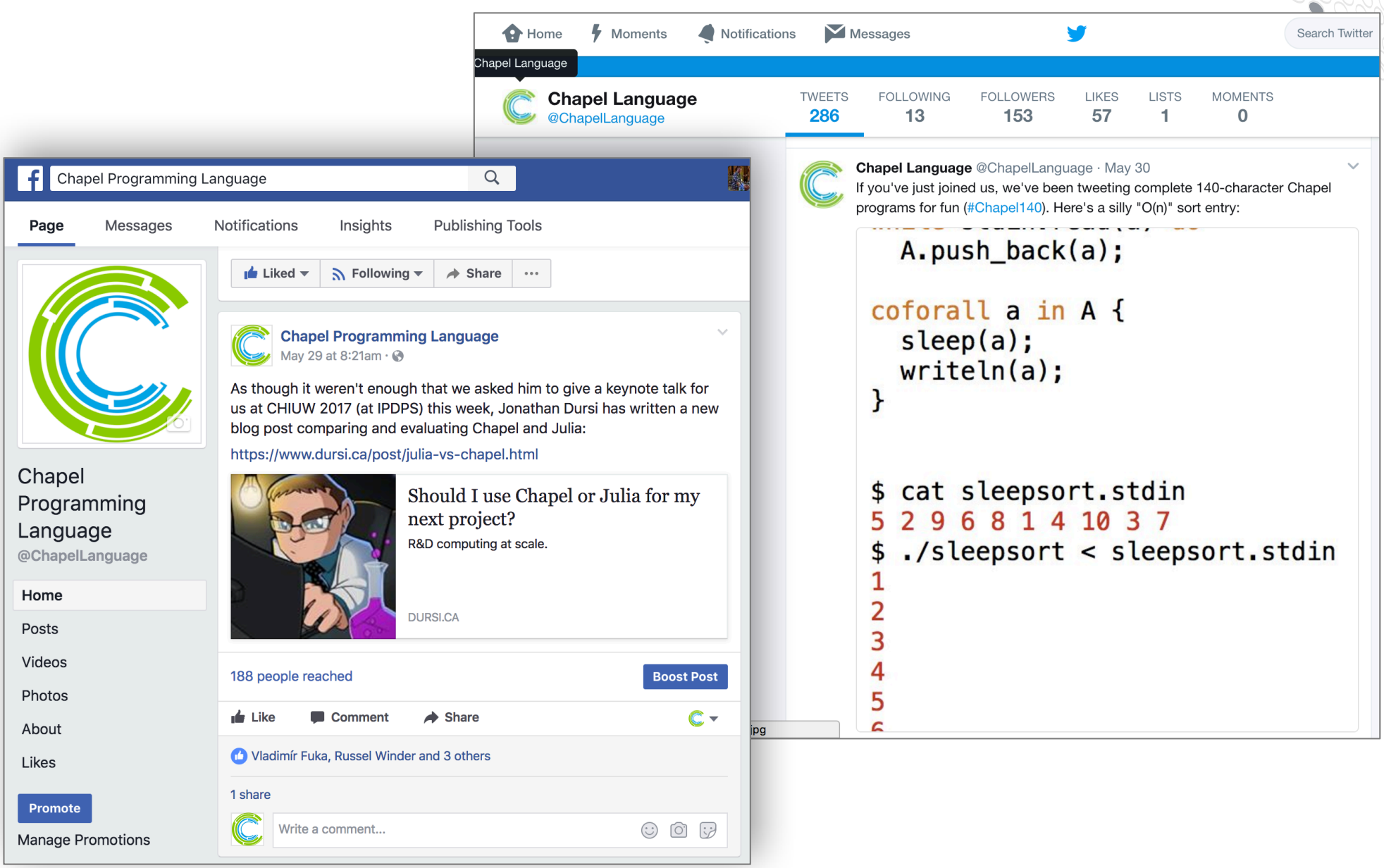
Google Summer of Code 2017

- **Google Summer of Code**
  - Google's way of supporting the open source community
  - Chapel had 2 successful students in GSoC 2016

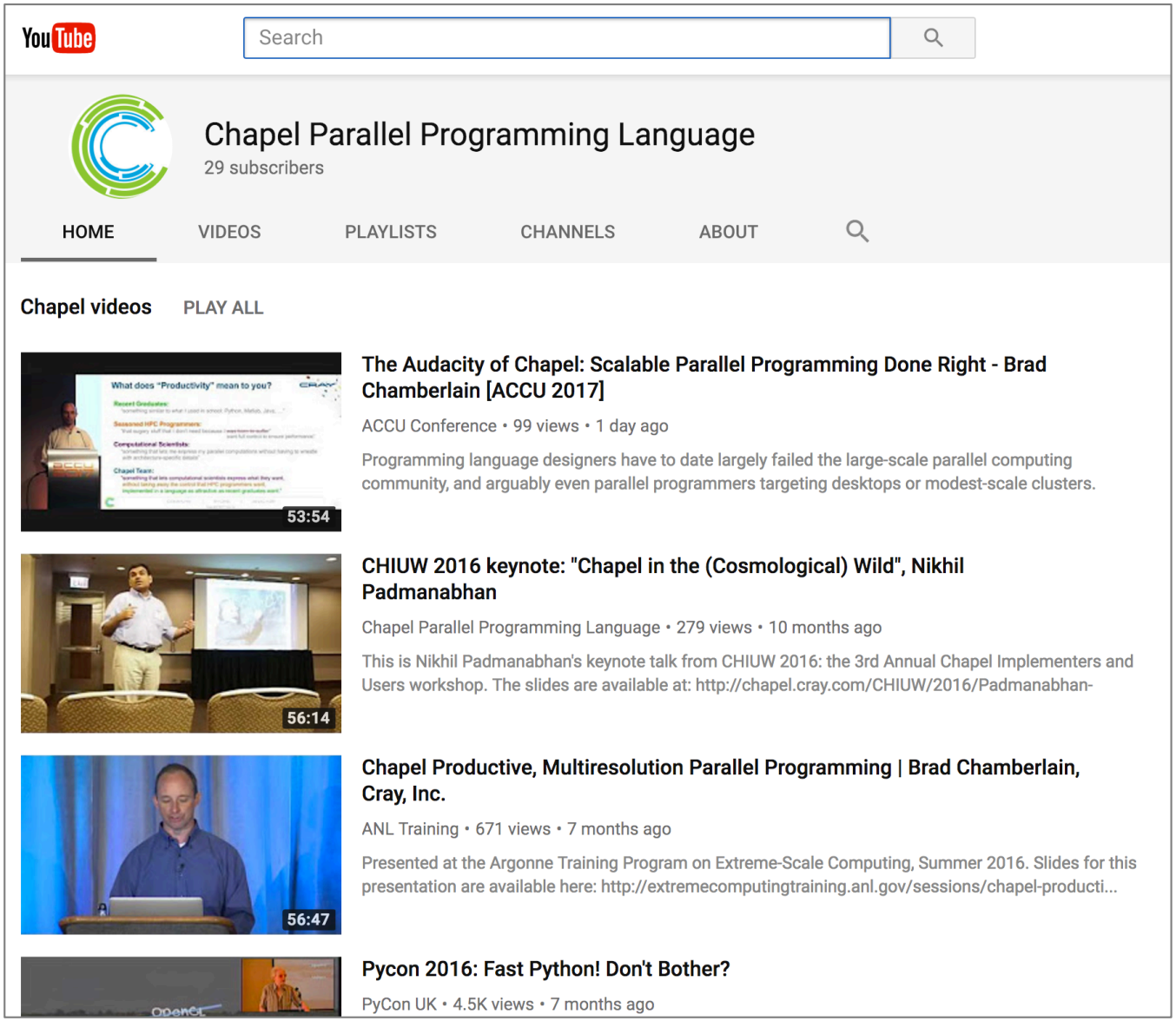
Google Summer of Code 2017:

- Przemyslaw Lesniak – **LLVM backend**
  - Mentor: Michael Ferguson
- Sarthak Munshi – **Cryptography module**
  - Mentor: Andrea Francesco Iurio
- Louis Jenkins - **Distributed data structures**
  - Mentor: Engin Kayraklioglu
- Nikhil Mahendran - **Chapel online**
  - Mentors: Ashish Chaudhary and Ben Albrecht

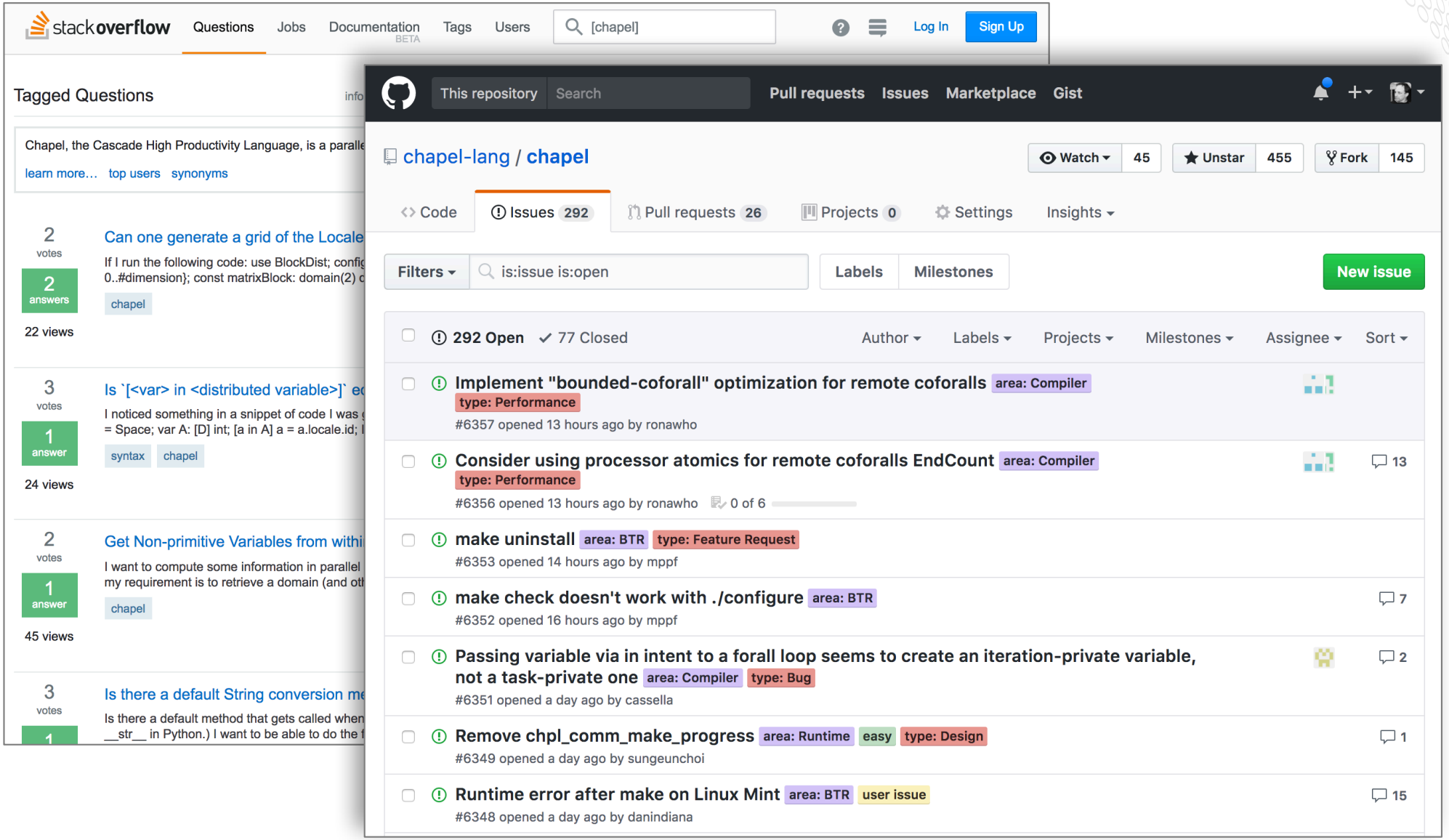
Chapel on Facebook and Twitter



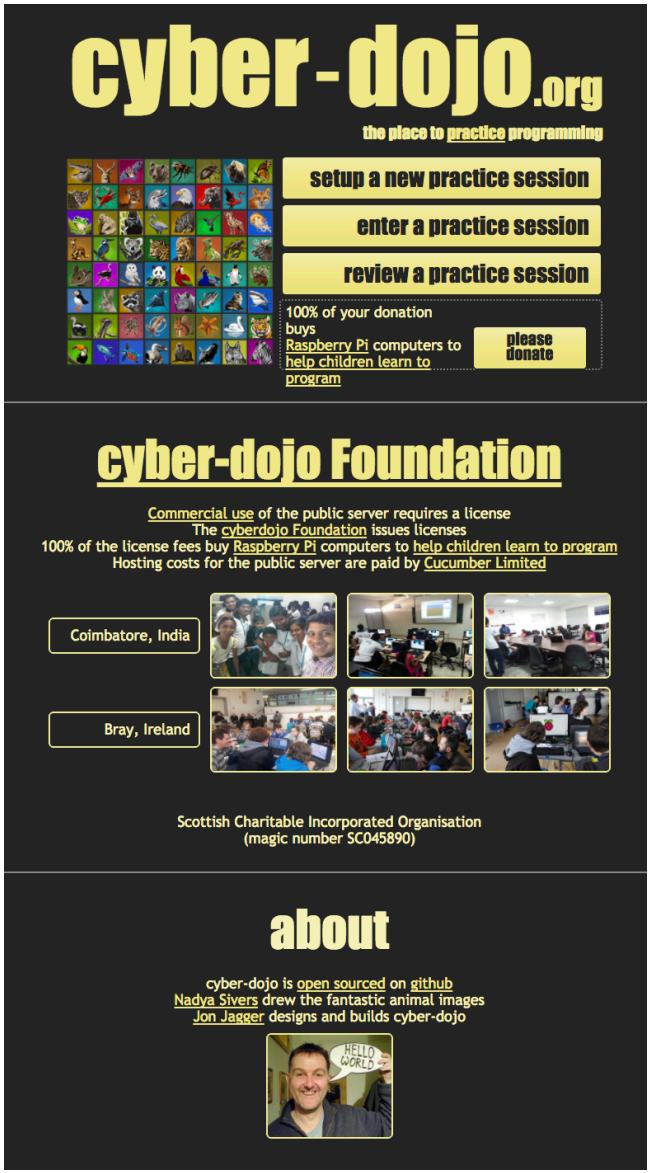
Chapel YouTube Channel

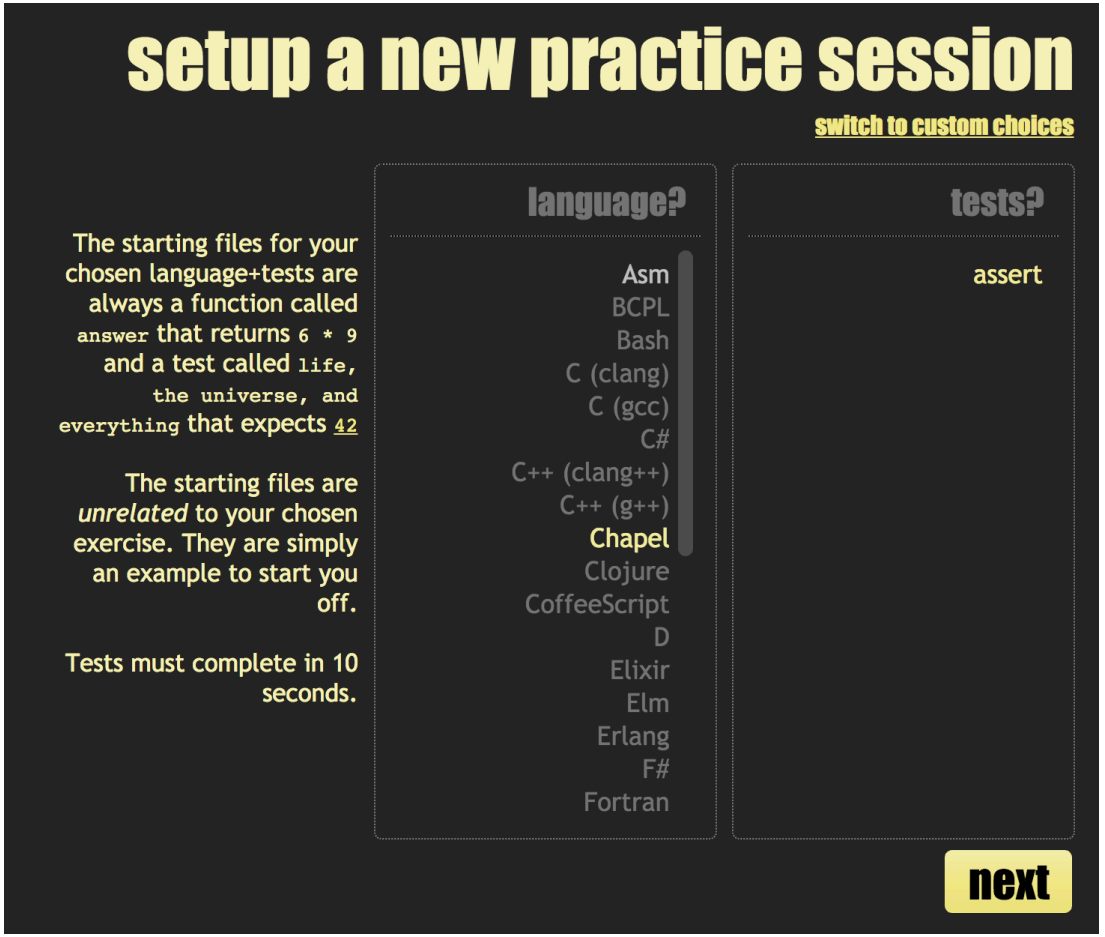


Chapel StackOverflow and GitHub Issues



Chapel on cyber-dojo





What's Next?

Top Chapel Priorities for version 1.16

- **Wrap up key language features:**
  - initializers
  - error-handling
  - delete-free class idioms (Shared, Owned)
- **Package Manager**
- **Benchmark- / App- / User-driven…**

  …performance tuning

  …library expansions

Our #1 Challenge

- **How to grow the user and developer communities?**
- **How to encourage people to look at Chapel again?**
  - overcome impressions made in our young, awkward years…

'Scientific computing communities are very wary of new technologies (it took 10+ years for Python to start getting any traction), with the usual, self-fulfulling, fear being "what if it goes away?"'

- Jonathan Dursi, from *Should I Use Chapel or Julia for my next project?*

CHIUIW 2017: Agenda (chapel.cray.com/CHIUW2017.html)

8:30:	Chapel Boot Camp (optional)
9:00:	Welcome, State of the Project
9:30:	Break
10:00:	Talks: Chapel Design and Implementation
11:10:	Quick Break
11:20:	Talks: Targeting New Architectures
12:00:	Lunch
1:30:	Keynote Talk: Jonathan Dursi
2:30:	Talks: Uses of Chapel
3:20:	Break
3:50:	Talks: Benchmarking and Performance
4:40:	Lightning Talks and Flash Discussions
5:30:	Wrap-up / Head to Dinner

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.

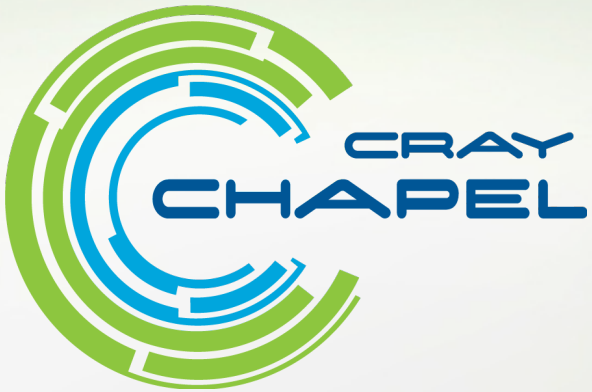
All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

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.



CRAY

THE SUPERCOMPUTER COMPANY